OMRON

머신 자동화 컨트롤러
NJ 시리즈

명령 참조 설명서
기본편

NJ501-15□□
NJ501-14□□
NJ501-13□□
NJ301-12□□
NJ301-11□□

SYSMAC
always in control

SBCA-360J-K6

부탁의 말

(1) 본 설명서 내용의 일부 또는 전체를 무단으로 복사, 복제, 전재하는 것을 금지합니다.

(2) 본 설명서의 내용은 개선을 위해 예고 없이 사양 등이 변경되는 경우가 있으므로 미리 양지해 주시길 바랍니다.

(3) 본 설명서 내용에 만전을 기하였으나 의문이 가는 점이나 잘못된 점을 발견하면 번거로우시겠지만 당사 지점 또는 영업소로 문의해 주십시오.
문의하실 때는 설명서 말미에 기재된 설명서 번호도 함께 알려주십시오.

상표

- Sysmac은 일본 및 기타 국가에서 오므론 주식회사의 FA 기기 제품의 상표 또는 등록 상표입니다.
- Windows, Windows XP, Windows Vista, Windows 7은 미국 및 기타 국가에서 미국 Microsoft Corporation의 등록 상표입니다.
- EtherCAT®은 독일의 Beckhoff Automation GmbH가 라이센스를 제공하고 특허 취득이 완료된 기술이자 등록 상표입니다.
- ODVA, CIP, CompoNet, DeviceNet, EtherNet/IP는 ODVA의 상표입니다.
- SD, SDHC 로고는 SD-3C, LLC의 상표입니다.

본문에 게재된 기타 시스템 이름 및 제품명은 각 회사의 상표 또는 등록 상표입니다.

머리말

NJ 시리즈 CPU 유니트를 구입해 주셔서 감사합니다.
이 설명서는 NJ 시리즈 CPU 유니트를 사용하는 데 필요한 정보를 소개합니다. 이 설명서를 잘 읽고 기능과 성능 등을 충분히 숙지한 후 시스템 구축에 활용하십시오.
또한 읽은 후에도 이 설명서를 가까운 곳에 잘 보관해 두고 사용하십시오.

대상 독자

이 설명서는 다음 사용자를 대상으로 합니다.
전기 지식(전기 공사 또는 이와 동등한 지식)이 있는
- FA 기기 도입 담당자
- FA 시스템 설계자
- FA 기기를 설치, 연결하는 기술자
- FA 현장 관리자

또한 프로그래밍 언어에 관해 국제 표준 규격 IEC61131-3 또는 일본 표준 규격 JIS B3503에 규정된 내용을 숙지하고 있는 사람을 대상으로 기술하였습니다.

대상 제품

이 설명서는 다음 제품을 대상으로 합니다.
- NJ 시리즈 CPU 유니트
  - NJ501-15 □□
  - NJ501-14 □□
  - NJ501-13 □□
  - NJ301-12 □□
  - NJ301-11 □□

각 제품의 사양 중 일부는 다른 설명서에 기재되어 있는 경우가 있습니다. 🕮 '설명서의 구성(2페이지)' 및 🕮 '관련 설명서(25페이지)'를 확인하십시오.

설명서의 구성

본 제품에서는 'NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편', 'NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편', 'NJ 시리즈 명령 참조 설명서 기본편'의 3가지 설명서에 NJ 시리즈를 사용하기 위한 기본 정보가 기재되어 있습니다.
각종 조작에는 주로 자동화 소프트웨어 Sysmac Studio를 사용합니다. Sysmac Studio에 대해서는
🕮 "Sysmac Studio Version 1 조작 설명서"(SBCA-362)를 참조하십시오.
그 밖의 설명서는 시스템 구성이나 용도에 맞춰 참조할 수 있게 구성되었습니다.
목적에 따라 해당 설명서를 읽은 후 본 제품을 이용하시기 바랍니다 .

사용 목적	설명서															
	NJ 시리즈										NX 시리즈					
	기본 정보															
	NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편	NJ 시리즈 명령 참조 설명서 기본편	NJ 시리즈 CPU 유니트 사용 설명서 모션제어편	NJ 시리즈 CPU 유니트 Ether CAT 포트 사용설명서	NJ 시리즈 CPU 유니트 명령 참조 설명서 모션편	NJ 시리즈 내장 EtherNet/IP 포트 사용설명서	NJ 시리즈 데이터 베이스 연결 CPU 유니트 사용설명서	NJ 시리즈 트러블 슈팅 설명서	NX 시리즈 EtherCAT 커플러 유니트 사용설명서	NX 시리즈 각 NX 유니트 사용설명서	NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	NX 시리즈 데이터 참조 설명서	각 CJ 시리즈 고성능 유니트 사용설명서 NJ 시리즈 연결편
NJ 시리즈의 개요 파악	●															
장치, 설치, 하드웨어를 설정																
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
데이터베이스 연결 서비스를 사용하는 경우	●							●								
NX 시리즈를 사용하는 경우										●	●			●		
세이프티 컨트롤을 실행하는 경우												●		●		
CJ 시리즈 유니트를 사용하는 경우																●
소프트웨어 설정																
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우		●					●									
데이터베이스 연결 서비스를 사용하는 경우								●								
NX 시리즈를 사용하는 경우										●	●			●		
세이프티 컨트롤을 실행하는 경우										●		●		●		
사용자 프로그램 작성																
모션 제어를 실행하는 경우				●		●					●					
EtherCAT을 사용하는 경우					●											
데이터베이스 연결 서비스를 사용하는 경우		●	●					●								
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우												●	●			
CJ 시리즈 유니트를 사용하는 경우																●
이상 처리를 포함하는 경우									●							

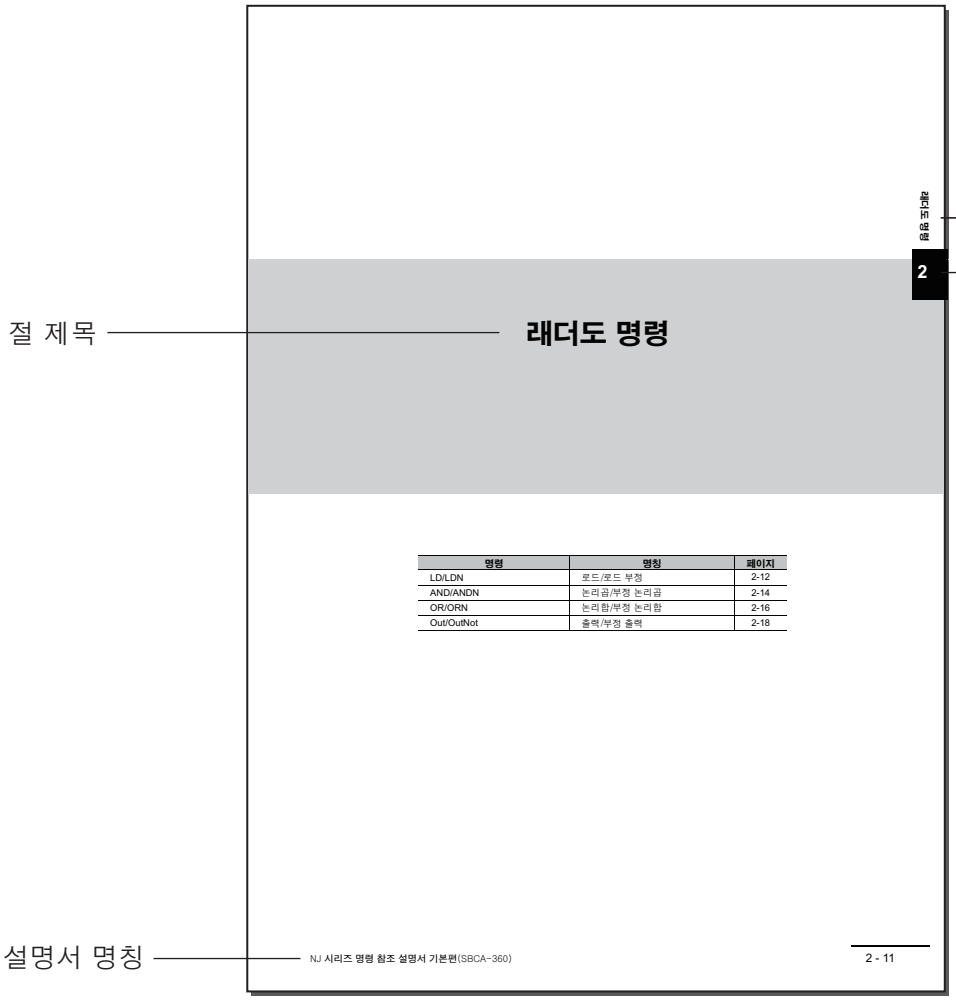
사용 목적	설명서															
	NJ 시리즈										NX 시리즈					
	기본 정보															
	NJ 시리즈 CPU 유니트 하드웨어편	NJ 시리즈 CPU 유니트 소프트웨어편	NJ 시리즈 명령 참조 설명서 기본편	NJ 시리즈 CPU 유니트 모션제어편	NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 설명서	NJ 시리즈 CPU 유니트 명령 참조 설명서 모션편	NJ 시리즈 내장 EtherNet/IP 포트 설명서	NJ 시리즈 데이터 베이스 연결 CPU 유니트 설명서	NJ 시리즈 트러블 슈팅 설명서	NX 시리즈 EtherCAT 커플러 유니트 설명서	NX 시리즈 각 NX 유니트 설명서	NX 시리즈 세이프티 컨트롤 유니트 설명서	NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	NX 시리즈 데이터 참조 설명서	각 CJ 시리즈 고성능 유니트 설명서 NJ 시리즈 연결편
동작 확인이나 디버그 실행		●														
모션 제어를 실행하는 경우				●							●					
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
데이터베이스 연결 서비스를 사용하는 경우								●								
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우												●				
이상 관리 기능이나 문제 발생 시 대처 방법 파악*1	△	△		△	△		△	△	●	△	△	△				△
보수 작업 파악	●															
모션 제어를 실행하는 경우				●							●					
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우												●				
CJ 시리즈 유니트를 사용하는 경우																●

*1. 이상 관리의 개념 및 이상 항목의 개요에 대해서는 "NJ 시리즈 트러블 슈팅 설명서"(SBCA-361)를 참조하십시오. 이상에 대한 자세한 내용은 이상의 내용에 따라 △ 표시된 설명서를 참조하십시오.

설명서 읽는 방법

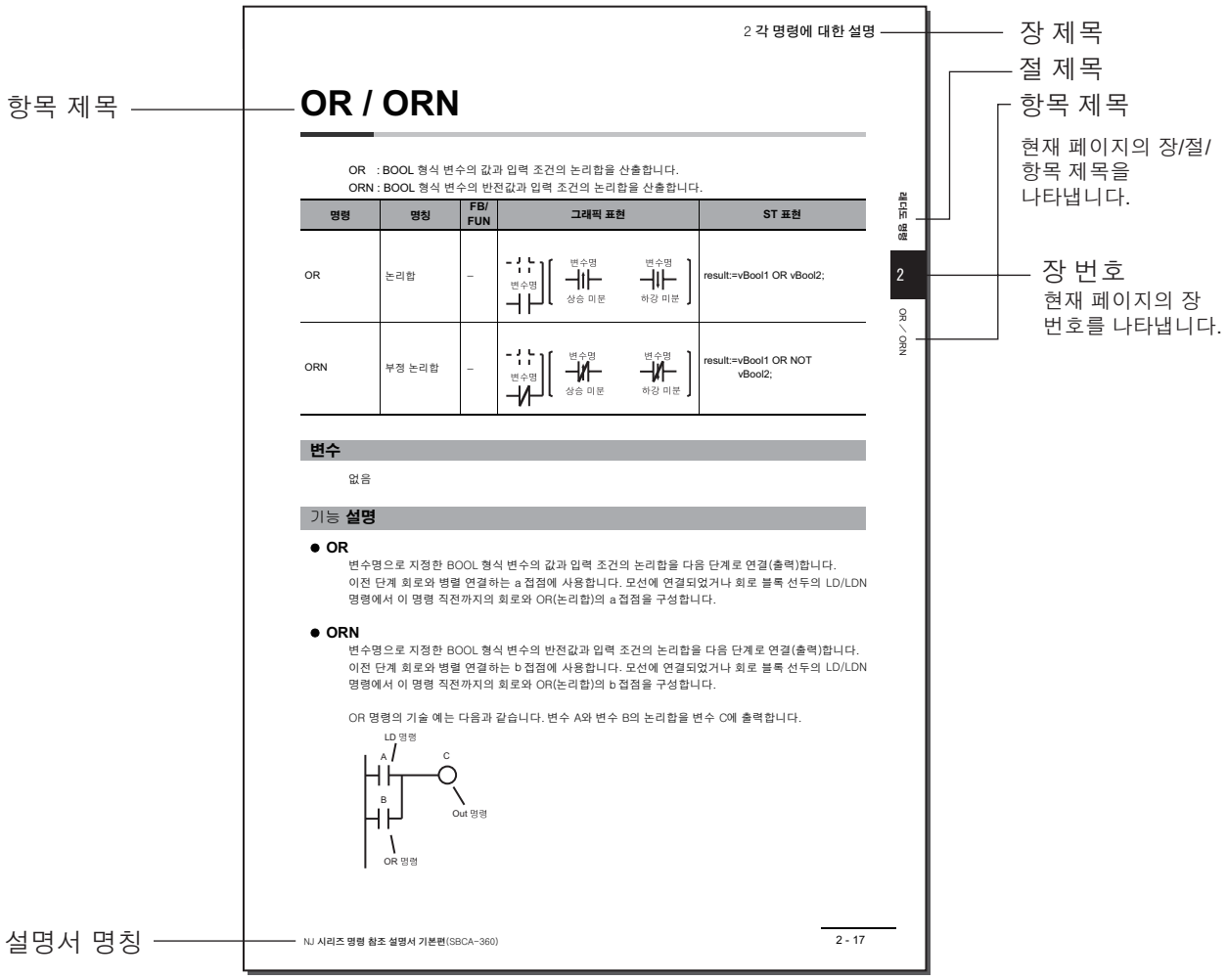
페이지 구성

본 설명서의 각 페이지 구성은 다음과 같습니다.



명령	명칭	페이지
LD/LDN	로드/로드 부정	2-12
AND/ANDN	논리곱/부정 논리곱	2-14
OR/ORN	논리합/부정 논리합	2-16
Out/OutNot	출력/부정 출력	2-18

이 페이지는 설명용 예이므로 실제 내용과 다릅니다.



이 페이지는 설명용 예이므로 실제 내용과 다릅니다.

아이콘

본 설명서에서 사용하는 아이콘에는 다음과 같은 의미가 있습니다.



자세한 정보 및 관련 정보가 기재되어 있는 위치를 나타냅니다.



버전 관련 정보

CPU 유니트 또는 Sysmac Studio의 버전에 따른 성능이나 기능의 차이에 대해 설명합니다.

목차 구성

1 명령 목록

2 각 명령에 대한 설명

A 부록

I 색인

목차

머리말 **1**
설명서의 구성 **2**
설명서 읽는 방법 **4**
목차 구성 **7**
동의 사항 **15**
안전상의 주의 사항 **17**
안전상의 중요 사항 **18**
사용상의 주의 사항 **19**
법규와 규격 **20**
유니트 버전 **22**
관련 설명서 **25**
설명서 개정 이력 **27**

제1장 명령 목록

명령 목록 1-2

제2장 각 명령에 대한 설명

이 장을 보는 방법 2-3
래더도 명령 **2-15**
LD / LDN 2-16
AND / ANDN 2-18
OR / ORN 2-21
Out / OutNot 2-24
ST 구문 명령 **2-27**
IF 2-28
CASE 2-32
WHILE 2-36
REPEAT 2-38
RETURN 2-40
FOR 2-41
EXIT 2-42
시퀀스 입력 명령 **2-43**
R_TRIG, Up / F_TRIG, Down 2-44
TestABit / TestABitN 2-47
시퀀스 출력 명령 **2-49**
RS 2-50
SR 2-52
Set / Reset 2-54
SetBits / ResetBits 2-57
SetABit / ResetABit 2-59
OutABit 2-61

시퀀스 제어 명령 2-63
End 2-64
RETURN 2-65
MC / MCR 2-66
JMP 2-81
FOR / NEXT 2-83
BREAK 2-90

비교 명령 2-93
EQ, = ... 2-94
NE,<> 2-96
LT, < / LE, <= / GT, > / GE, >= 2-98
EQascii 2-101
NEascii 2-103
LTascii / LEascii / GTascii / GEascii 2-105
Cmp 2-108
ZoneCmp 2-110
TableCmp 2-112
AryCmpEQ / AryCmpNE 2-115
AryCmpLT / AryCmpLE / AryCmpGT / AryCmpGE 2-117
AryCmpEQV / AryCmpNEV 2-119
AryCmpLTV / AryCmpLEV / AryCmpGTV / AryCmpGEV 2-121

타이머 명령 2-123
TON 2-124
TOF 2-129
TP 2-132
AccumulationTimer 2-135
Timer 2-138

카운터 명령 2-141
CTD 2-142
CTD_** 2-144
CTU 2-146
CTU_** 2-148
CTUD 2-151
CTUD_** 2-155

산술 명령 2-161
ADD,+ 2-162
AddOU, +OU 2-165
SUB, - 2-168
SubOU, -OU 2-171
MUL, * 2-174
MulOU, *OU 2-177
DIV, / 2-180
MOD 2-183
ABS 2-185
RadToDeg / DegToRad 2-187
SIN / COS / TAN 2-189
ASIN / ACOS / ATAN 2-192
SQRT 2-196
LN / LOG 2-198
EXP 2-201
EXPT, ** 2-203
Inc / Dec 2-205
Rand 2-207
AryAdd 2-209
AryAddV 2-211
ArySub 2-213
ArySubV 2-215
AryMean 2-217
ArySD 2-219
ModReal 2-221
Fraction 2-223
CheckReal 2-225

BCD 변환 명령 2-229
_BCD_TO_* 2-230
_TO_BCD_* 2-233
BCD_TO_** 2-236
BCDsToBin 2-239
BinToBCDs_**..... 2-242
AryToBCD 2-245
AryToBin 2-247

데이터 형식 변환 명령 2-249
_TO_*(정수→정수 변환 그룹)..... 2-250
_TO_*(정수→비트 열 변환 그룹) 2-253
_TO_*(정수→실수 변환 그룹)..... 2-255
_TO_*(비트 열→정수 변환 그룹) 2-257
_TO_*(비트 열→비트 열 변환 그룹)..... 2-259
_TO_*(비트 열→실수 변환 그룹) 2-261
_TO_*(실수→정수 변환 그룹)..... 2-263
_TO_*(실수→비트 열 변환 그룹) 2-266
_TO_*(실수→실수 변환 그룹)..... 2-268
**_TO_STRING(정수→문자열 변환 그룹)..... 2-270
**_TO_STRING(비트 열→문자열 변환 그룹) 2-272
**_TO_STRING(실수→문자열 변환 그룹)..... 2-274
RealToFormatString..... 2-276
LrealToFormatString 2-281
STRING_TO_**(문자열→정수 변환 그룹)..... 2-286
STRING_TO_**(문자열→비트 열 변환 그룹) 2-288
STRING_TO_**(문자열→실수 변환 그룹)..... 2-290
TO_**(정수 변환 그룹)..... 2-293
TO_**(비트 열 변환 그룹) 2-295
TO_**(실수 변환 그룹)..... 2-297
EnumToNum..... 2-299
NumToEnum..... 2-301
TRUNC / Round / RoundUp 2-303

비트 열 연산 명령 2-305
AND, & / OR / XOR..... 2-306
XORN 2-309
NOT 2-311
AryAnd / AryOr / AryXor / AryXorN..... 2-312

선택 명령..... 2-315
SEL 2-316
MUX 2-318
LIMIT 2-321
Band 2-323
Zone 2-325
MAX / MIN 2-327
AryMax / AryMin 2-329
ArySearch..... 2-332

데이터 전송 명령 2-335
MOVE 2-336
MoveBit 2-339
MoveDigit..... 2-341
TransBits..... 2-343
MemCopy 2-345
SetBlock..... 2-347
Exchange..... 2-349
AryExchange 2-351
AryMove..... 2-353
Clear 2-355
Copy**ToNum(비트 열→부호가 있는 정수)..... 2-357
Copy**To***(비트 열→실수)..... 2-359
CopyNumTo**(부호가 있는 정수→비트 열)..... 2-361
CopyNumTo**(부호가 있는 정수→실수)..... 2-363

Copy**To***(실수→비트 열) 2-365
Copy**ToNum(실수→부호가 있는 정수) 2-366

이동 명령 2-369
AryShiftReg 2-370
AryShiftRegLR 2-372
ArySHL / ArySHR 2-375
SHL / SHR 2-378
NSHLC / NSHRC 2-380
ROL / ROR 2-382

데이터 변환 명령 2-385
Swap 2-386
Neg 2-387
Decoder 2-389
Encoder 2-392
BitCnt 2-394
ColmToLine_** 2-395
LineToColm 2-397
Gray 2-399
UTF8ToSJIS 2-404
SJISToUTF8 2-406
PWLApprox / PWLApproxNoLineChk 2-408
PWLLineChk 2-413
MovingAverage 2-416
DispartReal 2-422
UniteReal 2-425
NumToDecString / NumToHexString 2-427
HexStringToNum_** 2-430
FixNumToString 2-432
StringToFixNum 2-434
DtToString 2-437
DateToString 2-439
TodToString 2-440
GrayToBin_** / BinToGray_** 2-442
StringToAry 2-445
AryToString 2-447
DispartDigit 2-449
UniteDigit_** 2-451
Dispart8Bit 2-453
Unite8Bit_** 2-455
ToAryByte 2-457
AryByteTo 2-462
SizeOfAry 2-467

스택/테이블 명령 2-469
StackPush 2-470
StackFIFO / StackLIFO 2-480
StackIns 2-483
StackDel 2-485
RecSearch 2-487
RecRangeSearch 2-492
RecSort 2-497
RecNum 2-502
RecMax / RecMin 2-504

FCS 명령 2-509
StringSum 2-510
StringLRC 2-512
StringCRCCCITT 2-514
StringCRC16 2-516
AryLRC_** 2-518
AryCRCCCITT 2-520
AryCRC16 2-522

문자열 명령 2-525
CONCAT 2-526

LEFT / RIGHT 2-528
MID 2-530
FIND 2-532
LEN 2-534
REPLACE 2-535
DELETE 2-537
INSERT 2-539
GetByteLen 2-541
ClearString 2-542
ToUCase/ToLCase 2-543
TrimL/TrimR 2-545
AddDelimiter 2-547
SubDelimiter 2-558

시간/시각 명령 2-569
ADD_TIME 2-570
ADD_TOD_TIME 2-572
ADD_DT_TIME 2-574
SUB_TIME 2-576
SUB_TOD_TIME 2-578
SUB_TOD_TOD 2-580
SUB_DATE_DATE 2-581
SUB_DT_DT 2-582
SUB_DT_TIME 2-584
MULTIME 2-586
DIVTIME 2-588
CONCAT_DATE_TOD 2-590
DT_TO_TOD 2-592
DT_TO_DATE 2-594
SetTime 2-596
GetTime 2-598
DtToSec 2-600
DateToSec 2-602
TodToSec 2-603
SecToDt 2-604
SecToDate 2-606
SecToTod 2-608
TimeToNanoSec 2-610
TimeToSec 2-611
NanoSecToTime 2-612
SecToTime 2-613
ChkLeapYear 2-615
GetDaysOfMonth 2-616
DaysToMonth 2-618
GetDayOfWeek 2-620
GetWeekOfYear 2-622
DtToDateStruct 2-624
DateStructToDt 2-626
TruncTime 2-628
TruncDt 2-632
TruncTod 2-636

아날로그 제어 명령 2-641
PIDAT 2-642
TimeProportionalOut 2-670
LimitAlarm_** 2-686
LimitAlarmDv_** 2-690
LimitAlarmDvStbySeq_** 2-694
ScaleTrans 2-709
AC_StepProgram 2-712

시스템 제어 명령 2-737
TraceSamp 2-738
TraceTrig 2-741
GetTraceStatus 2-744
SetAlarm 2-747

ResetAlarm ... 2-752
GetAlarm ... 2-754
ResetPLCError ... 2-756
GetPLCError ... 2-760
ResetCJBError ... 2-762
GetCJBError ... 2-764
GetEIPError ... 2-766
ResetMCError ... 2-768
GetMCError ... 2-773
ResetECError ... 2-775
GetECError ... 2-777
SetInfo ... 2-780
ResetUnit ... 2-782
GetNTPStatus ... 2-786
RestartNXUnit ... 2-788
NX_ChangeWriteMode ... 2-793
NX_SaveParam ... 2-797

EtherCAT 통신 명령 ... 2-803
EC_CoESDOWrite ... 2-804
EC_CoESDORead ... 2-807
EC_StartMon ... 2-813
EC_StopMon ... 2-819
EC_SaveMon ... 2-821
EC_CopyMon ... 2-823
EC_DisconnectSlave ... 2-825
EC_ConnectSlave ... 2-833
EC_ChangeEnableSetting ... 2-835
NX_WriteObj ... 2-854
NX_ReadObj ... 2-869

EtherNet/IP 통신 명령 ... 2-877
CIPOpen ... 2-878
CIPOpenWithDataSize ... 2-887
CIPRead ... 2-890
CIPWrite ... 2-895
CIPSend ... 2-901
CIPClose ... 2-905
CIPUCMMRead ... 2-907
CIPUCMMWrite ... 2-912
CIPUCMMSend ... 2-919
SktUDPCreate ... 2-929
SktUDPRcv ... 2-936
SktUDPSend ... 2-939
SktTCPAccept ... 2-942
SktTCPConnect ... 2-945
SktTCPRcv ... 2-953
SktTCPSend ... 2-956
SktGetTCPStatus ... 2-959
SktClose ... 2-962
SktClearBuf ... 2-965
ChangeIPAdr ... 2-968
ChangeFTPAccount ... 2-977
ChangeNTPServerAdr ... 2-980

시리얼 통신 명령 ... 2-985
ExecPMCR ... 2-986
SerialSend ... 2-1000
SerialRcv / SerialRcvNoClear ... 2-1011
SendCmd ... 2-1026

SD 메모리 카드 명령 ... 2-1039
FileWriteVar ... 2-1040
FileReadVar ... 2-1046
FileOpen ... 2-1051
FileClose ... 2-1055

FileSeek 2-1058
FileRead 2-1061
FileWrite 2-1069
FileGets 2-1077
FilePuts 2-1084
FileCopy 2-1092
FileRemove 2-1100
FileRename 2-1105
DirCreate 2-1110
DirRemove 2-1113

타임 스탬프 명령 2-1117
NX_DOutTimeStamp 2-1118
NX_AryDOutTimeStamp 2-1124

기타 명령 2-1131
ReadNbit_** 2-1132
WriteNbit_** 2-1134
ChkRange 2-1136
GetMyTaskStatus 2-1138
Task_IsActive 2-1141
Lock / Unlock 2-1143
ActEventTask 2-1148
Get**Clk 2-1154
Get**Cnt 2-1155

부록

A-1 ErrorID로 확인할 수 있는 에러 코드 목록 A-2
A-2 에러 코드의 개요 A-14
A-3 에러 코드에 대한 자세한 내용 A-41
A-4 이벤트 태스크에서 사용할 수 없는 명령 A-162
A-5 NX 메시지 통신 이상 관련 명령 A-164
A-6 SDO Abort 코드 목록 A-165
A-7 버전 관련 정보 A-166

색인

동의 사항

'당사 제품'에 대해 특별한 합의 사항이 없는 경우, 고객의 구입처와 관계 없이 이 동의 사항에 기재된 조건을 적용합니다.

● 정의

동의 사항에 나오는 용어의 정의는 다음과 같습니다.

- '당사 제품': '당사'의 FA 시스템 기기, 범용 제어 기기, 센싱 기기, 전자 · 기구 부품
- '카탈로그 등': '당사 제품'에 관한 베스트 제어 기기 오므론, 전자 · 기구 부품 종합 카탈로그, 기타 카탈로그, 사양서, 사용 설명서, 설명서 등이며, 전자적 방법으로 제공되는 것도 포함됩니다.
- '사용 조건 등': '카탈로그 등'에 기재된 '당사 제품'의 사용 조건, 정격, 성능, 동작 환경, 취급 방법, 사용상의 주의 사항, 금지 사항 및 기타 사항
- '고객 용도': 고객이 '당사 제품'을 사용하는 방법에는 고객이 제조하는 부품, 전자 기판, 기기, 설비 또는 시스템 등에 '당사 제품'을 조립 또는 사용하는 방법이 포함됩니다.
- '적합성 등': '고객 용도'에서 '당사 제품'의 (a) 적합성, (b) 동작, (c) 제3자의 지적 재산 비침해, (d) 법령 준수 및 (e) 각종 규격의 준수

● 기재 사항에 대한 주의 사항

'카탈로그 등'에 기재된 내용에 대해서는 다음 사항을 숙지하십시오.

- 정격값 및 성능값은 단독 시험에서 각 조건을 기초로 얻은 값이며, 각 정격값 및 성능값의 복합 조건에서 얻은 값을 보증하는 것은 아닙니다.
- 참고 데이터는 참고용으로 제공하는 것으로, 해당 범위에서 항상 정상적으로 동작한다는 것을 보증하는 것은 아닙니다.
- 사용 사례는 참고이므로 '당사'는 '적합성 등'에 대해 보증하지 않습니다.
- '당사'는 개선이나 당사 사정 등에 따라 '당사 제품'의 생산을 중단하거나 '당사 제품'의 사양을 변경할 수 있습니다.

● 사용상의 주의 사항

채택 및 사용할 때는 다음 사항에 주의하십시오.

- 정격 · 성능 이외에 '사용 조건 등'을 준수하여 사용하십시오.
- 고객 자신이 '적합성 등'을 확인한 후 '당사 제품'의 사용 가능 여부를 판단하십시오. '당사'는 '적합성 등'을 일체 보증하지 않습니다.
- '당사 제품'이 고객 시스템 전체적으로 의도한 용도에 적합하게 배선 및 설치되었는지 반드시 고객이 직접 사전에 확인하십시오.
- '당사 제품'을 사용할 때는 (i) 정격 및 성능에 대해 여유가 있는 '당사 제품'의 사용, 여유 있는 안전한 설계, (ii) '당사 제품'이 고장나도 '고객 용도'의 위험을 최소화하는 안전 설계, (iiii) 사용자에게 위험을 알리기 위한 안전 대책의 시스템 전체적인 구축, (iv) '당사 제품' 및 '고객 용도'의 정기적인 보수의 각 사항을 실시하십시오.
- '당사 제품'은 일반 공업 제품용 범용 제품으로 설계 및 제조되었습니다. 따라서 다음과 같은 용도로 사용하는 것은 의도하지 않았으며, 고객이 '당사 제품'을 이러한 용도로 사용하는 경우, '당사'는 '당사 제품'에 대해 어떠한 보증도 하지 않습니다. 단, 아래의 용도라도 '당사'가 의도한 제품 용도인 경우나 특별한 합의가 있는 경우는 예외로 합니다.
  (a) 높은 안전성이 요구되는 용도(예: 원자력 제어 설비, 연소 설비, 항공 · 우주 설비, 철도 설비, 승강 설비, 오락 설비, 의료용 기기, 안전 장치, 기타 생명 · 인체에 위험이 미치는 용도)

(b) 높은 신뢰성이 요구되는 용도(예: 가스 · 수도 · 전기 등의 공급 시스템이나 24시간 연속 운전 시스템, 결제 시스템 외에 권리 · 재산을 취급하는 용도 등)
(c) 악조건 또는 열악한 환경에서 사용하는 용도(예: 옥외에 설치하는 설비, 화학적 오염이 발생하는 설비, 전자적 방해를 받는 설비, 진동 · 충격을 받는 설비 등)
(d) '카탈로그 등'에 기재되지 않은 조건이나 환경에서의 용도

- 상기 (a)부터 (d)에 기재된 내용 이외에 이 '카탈로그 등에 기재된 제품'은 자동차(이륜차 포함. 이하 동일)용 제품이 아닙니다. 자동차에 탑재하는 용도로는 사용하지 마십시오. 자동차 탑재용 제품에 대해서는 당사 영업 담당자에게 문의하십시오.

● 보증 조건

'당사 제품'의 보증 조건은 다음과 같습니다.

- 보증 기간 구입 후 1년입니다.
  (단, '카탈로그 등'에 별도로 기재된 경우는 제외됩니다.)
- 보증 내용 고장난 '당사 제'에 대해 다음 중 하나를 '당사' 임의의 판단으로 실시합니다.
  (a) 당사 보수 서비스 센터에서 고장난 '당사 제품'을 무상 수리
  (단, 전자 · 기구 부품에 대해서는 수리하지 않습니다.)
  (b) 고장난 '당사 제품'과 동일한 수량의 대체품 무상 제공
- 보증 대상 외 고장의 원인이 다음 항목에 해당하는 경우는 보증 대상에서 제외됩니다.
  (a) '당사 제품' 본래의 사용 방법 이외의 방법으로 사용
  (b) '사용 조건 등'에서 벗어난 사용
  (c) 이 동의 사항의 '사용상의 주의 사항'을 위반한 사용
  (d) '당사' 직원 이외의 사람에 의한 개조 또는 수리에 의한 경우
  (e) '당사' 직원 이외의 사람에 의한 소프트웨어 프로그램에 의한 경우
  (f) '당사'에서 출하 당시의 과학 기술 수준으로는 예견할 수 없었던 원인
  (g) 상기 이외의 '당사' 또는 '당사 제품' 이외의 원인(천재지변 등의 불가항력 포함)

● 책임의 제한

이 동의 사항에 기재된 보증이 '당사 제품'에 관한 보증의 전부입니다.
'당사 제품'과 관련되어 발생한 손해에 대해 '당사' 및 '당사 제품'의 판매점은 책임을 지지 않습니다.

● 수출 관리

'당사 제품' 또는 기술 자료를 수출 또는 비거주자에게 제공하는 경우, 안전 보장 무역 관리에 관한 일본 및 관계 각국의 법령 및 법규를 준수하십시오. 고객이 법령 및 규칙을 위반하는 경우에는 '당사 제품' 또는 기술 자료를 제공할 수 없는 경우가 있습니다.

안전상의 주의 사항

안전상의 주의 사항에 관해서는 다음 설명서를 참조하시기 바랍니다.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

안전상의 중요 사항

안전상의 중요 사항에 관해서는 다음 설명서를 참조하시기 바랍니다.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

사용상의 주의 사항

사용상의 주의 사항에 관해서는 다음 설명서를 참조하시기 바랍니다.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

법규와 규격

해외에서의 사용

본 제품 중 외환 및 외국무역법에서 정한 수출 허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우, 해당 법에 근거한 수출 허가, 승인(또는 역무 거래 허가)이 필요합니다.

EC 지령에 대한 적합성

적합 지령

- EMC 지령
- 저전압 지령

적합성의 개념

● EMC 지령

오므론의 제품은 각종 기계, 제조 장치에 조립하여 사용하는 전기 기기이므로 조립한 기계 및 장치가 보다 더 EMC 규격에 적합한 상태가 되도록 제품 자체에 관한 EMC 규격(*)을 준수하기 위해 노력하고 있습니다.

그러나 고객의 기계 및 장치가 다양하며 EMC의 성능은 EC 지령 적합 제품을 조립한 기기 및 제어반의 구성, 배선 상태, 배치 상태 등에 따라 달라지므로 고객이 사용하는 상태에서의 적합성은 확인할 수 없습니다. 따라서 기계 및 장치 전체의 최종적인 EMC 적합성을 고객 스스로 확인하시기 바랍니다.

* EMC (Electro-Magnetic Compatibility : 전자 환경 적합성) 관련 규격 중
EMS (Electro-Magnetic Susceptibility : 전자파 내성)에 관해서는 EN61131-2 또는 EN61000-6-2,
EMI (Electro-Magnetic Interference : 전자 방해)에 관해서는 EN61131-2 또는 EN61000-6-4.
또한 EN61000-6-4 Radiated emission은 10m법에 따릅니다.

● 저전압 지령

전원 전압 50V AC~1000V AC 및 75V DC~1500V DC에서 작동하는 기기에 대해 필요한 안전성을 확보해야 합니다. 적용 규격은 EN61131-2입니다.

● EC 지령에 대한 적합성

NJ 시리즈는 EC 지령에 적합합니다. 그러나 고객의 기계 및 장치가 EC 지령에 적합하도록 하기 위해서는 다음 사항에 주의해야 합니다.

- NJ 시리즈는 반드시 제어반 내에 설치하십시오.
- DC 전원 유니트 및 I/O 유니트에 연결하는 DC 전원은 강화 절연 또는 이중 절연된 것을 사용하십시오.
- NJ 시리즈의 EC 지령 적합 제품은 EMI에 관한 공통 방출 규격에 적합하지만, 특히 Radiated emission(10m법)에 있어서는 사용하는 제어반의 구성, 연결된 다른 기기와의 관계, 배선 등에 따라 달라질 수 있습니다.
  따라서 EC 지령 적합품인 NJ 시리즈를 사용하는 경우에도 고객이 기계 및 장비 전체에 걸쳐 EC 지령의 적합성을 확인하여 대처해야 합니다.

KC 규격에 대한 적합성

본 제품을 한국에서 사용하는 경우는 다음 주의 사항을 준수하십시오.

A 급 기기 (업무용 방송통신기자재)
이 기기는 업무용(A 급) 전자파적합기기로서 판매자
또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

ClassA 기기(업무용 방송 통신 기기)
이 기기는 업무용 전자파 발생 기기(ClassA)로, 가정 외의 장소에서 사용하기 위한 용도로 제작되었습니다. 판매자 또는 사용자는 이 점에 주의하십시오.

선급 규격에 대한 적합성

본 제품은 각종 선급 규격에 적합합니다. 각 선급 규격에 적용하는 경우 사용 조건이 설정되어 있어서 설치 장소에 따라서는 사용할 수 없는 경우도 있으므로 당사 영업 담당자에게 문의하십시오.

각 선급 규격의 사용 조건(NK, LR)

- 본 제품은 반드시 제어반에 설치하십시오.
- 제어반의 개폐구 등의 간격은 개스킷 등으로 완전히 차폐하십시오.
- 전원 라인에 다음과 같은 노이즈 필터를 연결하십시오.

노이즈 필터

제조업체	형식
Cosel(코셀)사	TAH-06-683

소프트웨어의 라이센스와 저작권

이 제품에는 타사의 소프트웨어가 조립되어 있습니다. 이 소프트웨어와 관련된 라이센스와 저작권은 http://www.ia.omron.co.kr/를 참조하십시오.

유니트 버전

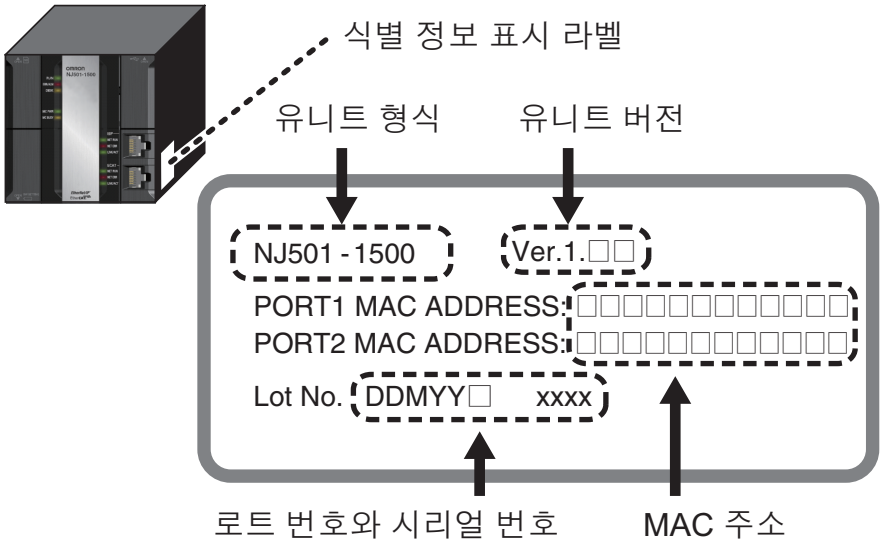
유니트 버전이란

유니트 버전이란, NJ 시리즈의 각 유니트의 개정 번호를 의미합니다. 동일 형식의 유니트에서도 유니트 버전이 다르면 탑재 기능에 몇 가지 차이점이 있습니다.

제품 상의 표기

제품 측면의 식별 정보 표시 라벨로 '유니트 버전'을 확인할 수 있습니다.

NJ 시리즈 NJ501-□□□□의 경우:



식별 정보 표시 라벨에는 다음과 같은 정보가 기재되어 있습니다.

명칭	기능
유니트 형식	본 유니트의 형식을 표시합니다.
유니트 버전	본 유니트의 유니트 버전을 표시합니다.
로트 번호와 시리얼 번호	본 유니트의 로트 번호와 시리얼 번호를 표시합니다. DDMYY: 로트 번호, □: 당사 사용, xxxx: 시리얼 번호 M은 월을 표시, X: 10월, Y: 11월, Z: 12월을 의미합니다.
MAC 주소	본 유니트의 내장 포트의 MAC 주소를 표시합니다.

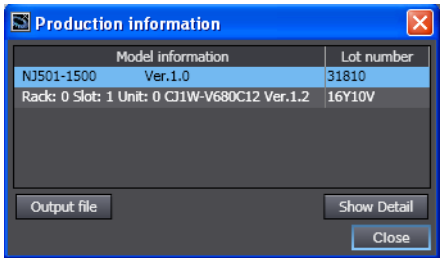
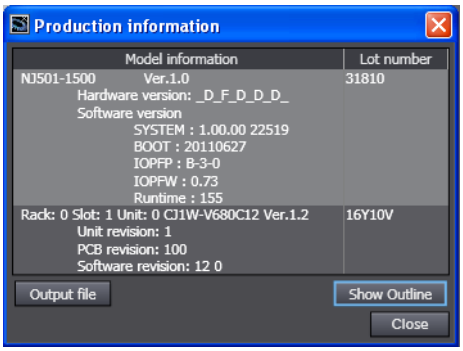
Sysmac Studio에 의한 확인 방법

Sysmac Studio의 [유니트 생산 정보]에서 CPU 유니트 및 CJ 시리즈의 고기능 I/O 유니트/CPU 고기능 유니트, EtherCAT 슬레이브의 유니트 버전을 확인할 수 있습니다.
CJ 시리즈의 기본 I/O 유니트의 유니트 버전은 Sysmac Studio에서는 확인할 수 없습니다.

● CPU 유니트 및 CJ 시리즈의 유니트인 경우

1 **멀티 뷰 탐색기에서 [구성 · 설정] | [CPU · 증설 랙]을 두 번 클릭합니다. 또는 [구성 · 설정] | [CPU · 증설 랙]을 마우스 오른쪽 버튼으로 클릭해서 [편집]을 선택합니다.**
구성 · 설정 레이어에 유니트 편집기가 표시됩니다.

2 **유니트 편집기의 공란에서 마우스 오른쪽 버튼으로 클릭해서 [생산 정보 표시]를 선택합니다.**
[생산 정보] 대화 상자가 표시됩니다.

간이 표시	상세 표시
	

유니트 형식 우측에 예) 'Ver.1.00'으로 표시됩니다.

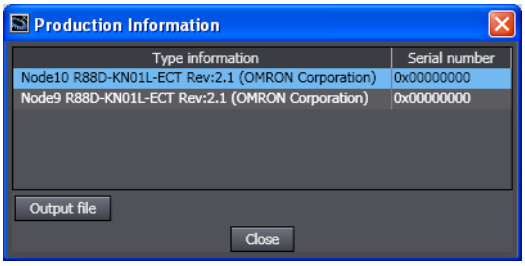
표시 내용은 다음과 같습니다.

CPU 유니트	CJ 시리즈 유니트
유니트 형식 유니트 버전 LOT 번호	유니트 형식 유니트 버전 LOT 번호 랙 번호 슬롯 번호 호기 번호/유니트 번호

● EtherCAT 슬레이브의 경우

1 **멀티 뷰 탐색기에서 [구성 · 설정] | [EtherCAT]을 두 번 클릭합니다. 또는 [구성 · 설정] | [EtherCAT]을 마우스 오른쪽 버튼으로 클릭해서 [편집]을 선택합니다.**
구성 · 설정 레이어에 EtherCAT 구성 편집 화면이 표시됩니다.

2 **EtherCAT 구성 편집 화면에서 마스터를 마우스 오른쪽으로 클릭해서 [생산 정보 표시]를 선택합니다.**
생산 정보 대화 상자가 표시됩니다.



표시 내용은 다음과 같습니다.
노드 주소
형식 번호(*)
시리얼 번호

* 형식을 확인할 수 없는 경우(ESI 파일이 없는 경우 등)에는 벤더 ID, 제품 코드, 개정 번호가 표시됩니다.

참고

데이터베이스 연결 서비스 등의 기능이 추가된 CPU 유니트, 통신 커플러 유니트, NX 유니트, 세이프티 컨트롤 유니트의 유니트 버전에 대해서는 각 유니트의 설명서를 참조하십시오.

유니트 버전과 Sysmac Studio의 버전

NJ 시리즈 CPU 유니트의 유니트 버전에 따라 호환되는 기능이 달라집니다. 버전 업그레이드를 통해 추가된 기능을 사용할 때는 호환되는 버전의 Sysmac Studio를 사용해야 합니다.
CPU 유니트의 유니트 버전 종류와 Sysmac Studio 버전의 관계 및 유니트 버전에 따른 지원 기능 목록은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359B 이상)"을 참조하십시오.

유니트 버전의 표기

본 설명서에서는 유니트 버전에 대해 다음과 같이 기재합니다.

제품의 명판	본 설명서에서의 기재 방법	특별 사항
Lot No.의 우측에, 'Ver.1.□□' 이상	유니트 Ver.1.0 이상	본 설명서에서 유니트 버전을 특별히 명기하지 않은 경우는 모든 유니트 버전을 대상으로 설명한 것입니다.

관련 설명서

관련된 NJ 시리즈 설명서는 다음 표에 기재되어 있으므로 참조하시기 바랍니다.

설명서 명칭	설명서 번호	형식	용도	내용
NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	SBCA-358	NJ501-□□□□ NJ301-□□□□	NJ 시리즈 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶은 경우 주로 하드웨어에 관한 정보	NJ 시리즈의 시스템 전체 개요 및 CPU 유니트에 관해 다음과 같은 내용을 설명합니다. • 특징 및 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검 사용 설명서 소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편	SBCA-359	NJ501-□□□□ NJ301-□□□□	NJ 시리즈 CPU 유니트의 프로그래밍/시스템 기동에 대해 알고 싶은 경우 주로 소프트웨어에 관한 정보	NJ 시리즈 CPU 유니트에 관해 다음과 같은 내용을 설명합니다. • CPU 유니트의 동작 • CPU 유니트의 기능 • 초기 설정 • IEC61131-3 기반의 언어 사양과 프로그래밍 사용 설명서 하드웨어편(SBCA-358)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편	SBCE-363	NJ501-□□□□ NJ301-□□□□	모션 제어의 설정 및 프로그래밍 개념에 대해 알고 싶은 경우	모션 제어를 위한 CPU 유니트의 설정 및 동작, 프로그래밍 개념에 대해 설명합니다. 프로그래밍할 때 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 명령 참조 설명서 기본편	SBCA-360	NJ501-□□□□ NJ301-□□□□	NJ 시리즈의 기본 명령 사양의 자세한 내용을 알고 싶은 경우	각 명령(IEC61131-3 사양)의 자세한 내용을 설명합니다. 프로그래밍할 때 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 명령 참조 설명서 모션편	SBCE-364	NJ501-□□□□ NJ301-□□□□	모션 명령 사양의 자세한 내용에 대해 알고 싶은 경우	각 모션 명령의 자세한 내용을 설명합니다. 프로그래밍할 때 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359), 사용 설명서 모션 제어편(SBCE-363)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 내장 EtherCAT® 포트 사용 설명서	SBCD-358	NJ501-□□□□ NJ301-□□□□	NJ 시리즈 CPU 유니트의 내장 EtherCAT 포트를 사용하는 경우	내장 EtherCAT 포트에 관해 설명합니다. 개요, 구성, 기능, 설정에 대해 설명합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 내장 EtherNet/IP™ 포트 사용 설명서	SBCD-359	NJ501-□□□□ NJ301-□□□□	NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트를 사용하는 경우	내장 EtherNet/IP 포트에 관해 설명합니다. 기본 설정, 태그 데이터 링크, 기타 기능에 대해 설명합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 트러블 슈팅 설명서	SBCA-361	NJ501-□□□□ NJ301-□□□□	NJ 시리즈에서 감지하는 이상의 자세한 내용에 대해 알고 싶은 경우	NJ 시리즈 시스템에서 감지하는 이상 관리의 개념과 각 이상 항목에 대해 설명합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 데이터베이스 연결 CPU 유니트 사용 설명서	SBCA-411	NJ501-1□20	NJ 시리즈에서 데이터베이스 연결 서비스 기능을 사용하는 경우	데이터베이스 연결 서비스 기능에 대해 설명합니다.

설명서 명칭	설명서 번호	형식	용도	내용
NX 시리즈 통신 커플러 유니트 사용 설명서	SBCD-361	NX-ECC□□□	통신 커플러 유니트 및 슬레이브 터미널의 사용 방법에 대해 알고 싶은 경우	통신 커플러 유니트와 NX 유니트로 구성된 슬레이브 터미널의 시스템 개요 및 구성 방법, 각 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다. 다음 유니트에 대해 설명서를 준비합니다. EtherCAT 커플러 유니트
NX 시리즈 NX 유니트 사용 설명서	SBCA-407 SBCA-408 SBCA-409 SBCE-374 SBCA-410	NX-ID□□□□ NX-OC□□□□ NX-OD□□□□ NX-AD□□□□ NX-DA□□□□ NX-TS□□□□ NX-PD1□□□ NX-PF0□□□ NX-TBX□□ NX-EC0□□□ NX-ECS□□□ NX-PG0□□□ NX-□□□□□□	NX 유니트의 사용 방법에 대해 알고 싶은 경우	NX 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다. 다음 유니트에 대해 개별적으로 설명서를 준비합니다. 디지털 I/O 유니트, 아날로그 I/O 유니트, 시스템 유니트, 위치 인터페이스 유니트
NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	Z930	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	NX 시리즈 세이프티 컨트롤 유니트의 사용 방법에 대해 알고 싶은 경우	NX 시리즈 세이프티 컨트롤 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다.
NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	Z931	NX-SL□□□□	세이프티 CPU 유니트용 명령의 사양을 자세히 알고 싶은 경우	세이프티 CPU 유니트용 명령에 대해 자세히 설명합니다. 프로그래밍할 때 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)과 함께 사용하십시오.
CJ 시리즈 고기능 유니트 사용 설명서 (NJ 시리즈 연결편)	SBCC-846 SBCC-849 SBCC-847 SDGR-703 SBCC-848 SBCD-354 SBCD-357 SBCD-355 SBCD-353	CJ1W-□□□□	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트로 사용하는 방법을 알고 싶은 경우	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트로 사용하는 방법(액세스 방법, 사용자 프로그램 I/F 등)이나 주의 사항에 대해 설명합니다. 다음 유니트에 대해 개별적으로 설명서를 준비합니다. 아날로그 I/O 유니트, 절연형 아날로그 유니트, 온도 조절 유니트, ID 센서 유니트, 고속 카운터 유니트, 시리얼 커뮤니케이션 유니트, DeviceNet 유니트, EtherNet/IP 유니트, CompoNet 마스터 유니트 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
Sysmac Studio Version 1 조작 설명서	SBCA-362	SYSMAC -SE2□□□	Sysmac Studio의 조작 방법, 기능에 대해 알고 싶은 경우	Sysmac Studio의 조작 방법에 대해 설명합니다.
CX-Integrator CS/CJCP/NSJ/NJ 시리즈용 네트워크 컨피그레이션 툴 조작 설명서	SBCA-347		네트워크 구축(데이터 링크, 라우팅 테이블, 통신 유니트 설정 등)에 대해 알고 싶은 경우	CX-Integrator의 조작 방법에 대해 설명합니다.
CX-Designer 사용 설명서	SBSA-532		프로그래머블 터미널 NS 시리즈의 화면 데이터를 작성하고 싶은 경우	CX-Designer의 조작 방법에 대해 설명합니다.
CX-Protocol 조작 설명서	SBCA-307		CJ 시리즈 시리얼 커뮤니케이션 유니트와 연결된 범용 외부 기기와의 데이터 송수신 순서(프로토콜)를 작성하고 싶을 때	CX-Protocol의 조작 방법에 대해 설명합니다.

설명서 개정 이력

설명서 개정 기호는 앞표지와 뒷표지에 기재된 설명서 번호 끝에 표기하였습니다.

예



개정 기호	개정 연월	개정 내용
A	2011년 7월	초판
B	2011년 9월	잘못 기재된 내용 수정에 따른 개정
C	2012년 3월	CPU 유니트 Ver.1.01, Sysmac Studio Ver.1.02 대응에 의한 개정
D	2012년 5월	CPU 유니트 Ver.1.02, Sysmac Studio Ver.1.03 대응에 의한 개정
E	2012년 8월	CPU 유니트 Ver.1.03, Sysmac Studio Ver.1.04 대응에 의한 개정
F	2013년 2월	CPU 유니트 Ver.1.04, Sysmac Studio Ver.1.05 대응에 의한 개정
G	2013년 4월	CPU 유니트 Ver.1.05, Sysmac Studio Ver.1.06 대응에 의한 개정
H	2013년 6월	CPU 유니트 Ver.1.06, Sysmac Studio Ver.1.07 대응에 의한 개정
J	2013년 9월	CPU 유니트 Ver.1.07, Sysmac Studio Ver.1.08 대응에 의한 개정

1

명령 목록

이 장에서는 NJ 시리즈에서 사용할 수 있는 명령의 목록을 제공합니다.

명령 목록 **1-2**

명령 목록

종류	명령	명칭	기능 개요	페이지
래더도 명령	LD	로드	BOOL 형식 변수의 값을 읽어옵니다.	2-16
	LDN	로드 부정	BOOL 형식 변수의 반전값을 읽어옵니다.	2-16
	AND	논리곱	BOOL 형식 변수의 값과 입력값의 논리곱을 산출합니다.	2-18
	ANDN	부정 논리곱	BOOL 형식 변수의 반전값과 입력값의 논리곱을 산출합니다.	2-18
	OR	논리합	BOOL 형식 변수의 값과 입력 조건의 논리합을 산출합니다.	2-21
	ORN	부정 논리합	BOOL 형식 변수의 반전값과 입력 조건의 논리합을 산출합니다.	2-21
	Out	출력	이전 단계까지의 논리 연산 처리 결과를 BOOL 형식 변수로 출력합니다.	2-24
	OutNot	부정 출력	이전 단계까지의 논리 연산 처리 결과의 반전값을 BOOL 형식 변수로 출력합니다.	2-24
ST 구문 명령	IF	선택	지정한 조건식의 평가 결과에 따라 실행될 구문을 이지선다 방식으로 선택합니다.	2-28
	CASE	다방향 분기	지정한 정수식의 값에 따라 복수의 구문 중에서 실행될 구문을 선택합니다.	2-32
	WHILE	반복	지정한 조건식의 평가 결과가 TRUE인 동안, 특정 구문을 반복 실행합니다.	2-36
	REPEAT	반복	특정 구문을 실행하여, 지정한 조건식이 TRUE가 될 때까지 그 구문을 반복 실행합니다.	2-38
	RETURN	복귀	펑션 또는 펑션 블록을 종료하고 처리를 호출원으로 되돌립니다.	2-40
	FOR	반복	반복 조건을 지정하여 FOR 구문에서 END_FOR 구문까지의 특정 구문을 반복 실행합니다.	2-41
	EXIT	루프 중단	가장 안쪽의 FOR 구문에서 END_FOR 구문까지의 반복 처리를 중단합니다.	2-42
시퀀스 입력 명령	R_TRIG, Up	상승 미분	입력 신호의 상승(FALSE→TRUE) 시, 1태스크 주기 동안만 TRUE를 출력합니다.	2-44
	F_TRIG, Down	하강 미분	입력 신호의 하강(TRUE→FALSE) 시, 1태스크 주기 동안만 TRUE를 출력합니다.	2-44
	TestABit	비트 테스트	비트 열의 지정 비트값을 출력합니다.	2-47
	TestABitN	비트 테스트 부정	비트 열의 지정 비트의 반전값을 출력합니다.	2-47
시퀀스 출력 명령	RS	리셋 우선 유지	BOOL 형식 변수의 값을 유지합니다. 세트 입력과 리셋 입력이 동시에 TRUE인 경우 리셋 입력이 우선합니다.	2-50
	SR	세트 우선 유지	BOOL 형식 변수의 값을 유지합니다. 세트 입력과 리셋 입력이 동시에 TRUE인 경우 세트 입력이 우선합니다.	2-52
	Set	세트	BOOL 형식 변수에 TRUE를 세트합니다.	2-54
	Reset	리셋	BOOL 형식 변수에 FALSE를 세트합니다.	2-54
	SetBits	다중 비트 세트	비트 열 데이터의 연속 복수 비트에 TRUE를 세트합니다.	2-57
	ResetBits	다중 비트 리셋	비트 열 데이터의 연속 복수 비트에 FALSE를 세트합니다.	2-57
	SetABit	1비트 세트	비트 열 데이터의 지정 비트에 TRUE를 세트합니다.	2-59
	ResetABit	1비트 리셋	비트 열 데이터의 지정 비트에 FALSE를 세트합니다.	2-59

종류	명령	명칭	기능 개요	페이지
시퀀스 출력 명령	OutABit	1비트 출력	비트 열 데이터의 지정 비트에 TRUE/FALSE를 세트합니다.	2-61
시퀀스 제어 명령	End	END	프로그램의 현재 태스크 주기의 실행을 종료합니다.	2-64
	RETURN	복귀	펑션 또는 펑션 블록을 종료하고 처리를 호출원으로 되돌립니다.	2-65
	MC	마스터 컨트롤 시작	마스터 컨트롤 영역의 시작점을 표시하고 마스터 컨트롤에 의한 리셋을 실행합니다.	2-66
	MCR	마스터 컨트롤 종료	마스터 컨트롤 영역의 종료점을 표시합니다.	2-66
	JMP	점프	지정한 점프 지점으로 처리가 이동됩니다.	2-81
	FOR	반복 시작	반복 처리의 시작 위치를 표시하고 반복 조건을 지정합니다.	2-83
	NEXT	반복 종료	반복 처리의 종료 위치를 표시합니다.	2-83
	BREAK	루프 중단	가장 안쪽의 FOR 명령에서 NEXT 명령까지의 반복 처리를 중단합니다.	2-90
비교 명령	EQ,=	비교 EQ	복수의 데이터가 모두 동일한지 판정합니다.	2-94
	NE,<>	비교 NE	2개의 데이터가 상이한지 판단합니다.	2-96
	LT,<	비교 LT	복수의 수치의 크기 비교(<)를 수행합니다.	2-98
	LE,<=	비교 LE	복수의 수치의 크기 비교(<=)를 수행합니다.	2-98
	GT,>	비교 GT	복수의 수치의 크기 비교(>)를 수행합니다.	2-98
	GE,>=	비교 GE	복수의 수치의 크기 비교(>=)를 수행합니다.	2-98
	EQascii	문자열 비교 EQ	복수의 문자열이 모두 동일한지 판정합니다.	2-101
	NEascii	문자열 비교 NE	2개의 문자열이 상이한지 판단합니다.	2-103
	LTascii	문자열 비교 LT	복수의 문자열의 크기 비교(<)를 수행합니다.	2-105
	LEascii	문자열 비교 LE	복수의 문자열의 크기 비교(<=)를 수행합니다.	2-105
	GTascii	문자열 비교 GT	복수의 문자열의 크기 비교(>)를 수행합니다.	2-105
	GEascii	문자열 비교 GE	복수의 문자열의 크기 비교(>=)를 수행합니다.	2-105
	Cmp	비교	2개의 수치를 비교합니다.	2-108
	ZoneCmp	영역 비교	비교 데이터가 지정한 상한값과 하한값 사이에 있는지 판정합니다.	2-110
	TableCmp	테이블 비교	비교 데이터와 비교 테이블로 지정된 복수의 정의 구간을 비교합니다.	2-112
	AryCmpEQ	배열 일괄 비교 EQ	2개의 배열의 각 요소가 동일한지 판정합니다.	2-115
	AryCmpNE	배열 일괄 비교 NE	2개의 배열의 각 요소가 상이한지 판정합니다.	2-115
	AryCmpLT	배열 일괄 비교 LT	2개의 배열의 각 요소의 크기 비교(<)를 수행합니다.	2-117
	AryCmpLE	배열 일괄 비교 LE	2개의 배열의 각 요소의 크기 비교(<=)를 수행합니다.	2-117
	AryCmpGT	배열 일괄 비교 GT	2개의 배열의 각 요소의 크기 비교(>)를 수행합니다.	2-117
	AryCmpGE	배열 일괄 비교 GE	2개의 배열의 각 요소의 크기 비교(>=)를 수행합니다.	2-117
	AryCmpEQV	배열 요소 비교 EQ	배열의 각 요소와 수치가 동일한지 판정합니다.	2-119
	AryCmpNEV	배열 요소 비교 NE	배열의 각 요소와 수치가 상이한지 판정합니다.	2-119
	AryCmpLTV	배열 요소 비교 LT	배열의 각 요소와 수치의 크기 비교(<)를 수행합니다.	2-121
	AryCmpLEV	배열 요소 비교 LE	배열의 각 요소와 수치의 크기 비교(<=)를 수행합니다.	2-121
	AryCmpGTV	배열 요소 비교 GT	배열의 각 요소와 수치의 크기 비교(>)를 수행합니다.	2-121
	AryCmpGEV	배열 요소 비교 GE	배열의 각 요소와 수치의 크기 비교(>=)를 수행합니다.	2-121
타이머 명령	TON	온 딜레이 타이머	기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다.	2-124

종류	명령	명칭	기능 개요	페이지
타이머 명령	TOF	오프 딜레이 타이머	기동 시점부터 설정 시간 경과 후에 FALSE를 출력하는 타이머입니다.	2-129
	TP	펄스 출력	기동 시점부터 설정 시간 만큼 TRUE를 출력하는 타이머입니다.	2-132
	Accumulation Timer	적산 타이머	타이머 입력이 TRUE인 시간을 적산하는 타이머입니다.	2-135
	Timer	100ms 타이머	기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 100ms입니다.	2-138
카운터 명령	CTD	감산 카운터	카운터 입력 신호가 입력될 때마다 감산하는 카운터입니다. 프리 세트값, 카운터값의 데이터 형식은 INT입니다.	2-142
	CTD_**	감산 카운터 그룹	카운터 입력 신호가 입력될 때마다 감산하는 카운터입니다. 프리 세트값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.	2-144
	CTU	가산 카운터	카운터 입력 신호가 입력될 때마다 가산하는 카운터입니다.프리 세트값, 카운터값의 데이터 형식은 INT입니다.	2-146
	CTU_**	가산 카운터 그룹	카운터 입력 신호가 입력될 때마다 가산하는 카운터입니다.프리 세트값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.	2-148
	CTUD	가역 카운터	가산 카운터 입력 및 감산 카운터 입력에 따라 가감산하는 카운터입니다.프리 세트값, 카운터값의 데이터 형식은 INT입니다.	2-151
	CTUD_**	가역 카운터 그룹	가산 카운터 입력 및 감산 카운터 입력에 따라 가감산하는 카운터입니다.프리 세트값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.	2-155
산술 명령	ADD,+	가산	정수, 실수를 가산합니다.또는 문자열을 결합합니다.	2-162
	AddOU,+OU	가산(오버 플로우 확인 있음)	정수, 실수를 가산합니다. 또한 정수의 가산 결과에 대해 오버 플로우 확인을 수행합니다.	2-165
	SUB,-	감산	정수, 실수를 감산합니다.	2-168
	SubOU,-OU	감산(오버 플로우 확인 있음)	정수, 실수를 감산합니다. 또한 정수의 감산 결과에 대해 오버 플로우 확인을 수행합니다.	2-171
	MUL,*	승산	정수, 실수를 승산합니다.	2-174
	MulOU,*OU	승산(오버 플로우 확인 있음)	정수, 실수를 승산한 결과를 출력합니다. 또한 정수의 승산 결과에 대해 오버 플로우 확인을 수행합니다.	2-177
	DIV,/	제산	정수, 실수를 제산합니다.	2-180
	MOD	나머지	정수를 제산했을 때의 나머지를 구합니다.	2-183
	ABS	절대값	정수, 실수의 절대값을 구합니다.	2-185
	RadToDeg	라디안→각도 변환	실수를 라디안(rad) 단위에서 도(°)단위로 변환합니다.	2-187
	DegToRad	각도→라디안 변환	실수를 도(°) 단위에서 라디안(rad) 단위로 변환합니다.	2-187
	SIN	SIN 연산	실수의 sin(사인값) 연산을 수행합니다.	2-185
	COS	COS 연산	실수의 cos(코사인값) 연산을 수행합니다.	2-185
	TAN	TAN 연산	실수의 tan(탄젠트값) 연산을 수행합니다.	2-185
	ASIN	SIN^{-1} 연산	실수의 sin^{-1}(역 사인값) 연산을 수행합니다.	2-192
	ACOS	COS^{-1} 연산	실수의 cos^{-1}(역 코사인값) 연산을 수행합니다.	2-192

종류	명령	명칭	기능 개요	페이지
산술 명령	ATAN	TAN^{-1} 연산	실수의 tan^{-1}(역 탄젠트값) 연산을 수행합니다.	2-192
	SQRT	제곱근 연산	실수의 제곱근(√)을 구합니다.	2-196
	LN	자연 로그 연산	실수의 자연 로그를 구합니다.	2-198
	LOG	상용 로그 연산	실수의 상용 로그를 구합니다.	2-198
	EXP	자연 지수 연산	자연 로그의 밑수 e의 지수 함수를 구합니다.	2-201
	EXPT,**	거듭제곱 연산	2개의 실수의 거듭제곱을 구합니다.	2-203
	Inc	증가	정수값을 증가시킵니다.	2-205
	Dec	감소	정수값을 감소시킵니다.	2-205
	Rand	난수 발생	의사 난수를 발생시킵니다.	2-207
	AryAdd	배열 일괄 가산	2개의 배열의 요소별 가산을 수행합니다.	2-209
	AryAddV	배열 요소 가산	배열의 각 요소에 동일 수치를 가산합니다.	2-211
	ArySub	배열 일괄 감산	2개의 배열의 요소별 감산을 수행합니다.	2-213
	ArySubV	배열 요소 감산	배열의 각 요소에서 동일 수치를 감산합니다.	2-215
	AryMean	배열 요소의 평균값 산출	배열 요소의 평균값을 산출합니다.	2-217
	ArySD	배열 요소의 표준 편차	배열 요소의 표준 편차를 산출합니다.	2-219
	ModReal	실수 나머지	실수를 제산했을 때의 나머지를 산출합니다.	2-221
	Fraction	실수 소수부 추출	실수의 소수부를 추출합니다.	2-223
	CheckReal	실수 확인	실수가 무한대 또는 비수인지 판정합니다.	2-225
BCD 변환 명령	**_BCD_TO_***	BCD→부호가 없는 정수 변환 그룹	BCD의 비트 열을 부호가 없는 정수로 변환합니다.	2-230
	_TO_BCD_*	부호가 없는 정수→BCD 변환 그룹	부호가 없는 정수를 BCD의 비트 열로 변환합니다.	2-233
	BCD_TO_**	BCD군→부호가 없는 정수 변환 그룹	BCD의 비트 열을 부호가 없는 정수로 변환합니다.	2-236
	BCDsToBin	부호가 있는 BCD→부호가 있는 정수 변환	부호가 있는 BCD의 비트 열을 부호가 있는 정수로 변환합니다.	2-239
	BinToBCDs_**	부호가 있는 정수→BCD 변환 그룹	부호가 있는 정수를 부호가 있는 BCD의 비트 열로 변환합니다.	2-242
	AryToBCD	배열 일괄 BCD 변환	부호가 없는 정수 배열의 각 요소를 BCD의 비트 열로 변환합니다.	2-245
	AryToBin	배열 일괄 부호가 없는 정수 변환	BCD의 비트 열 배열의 각 요소를 부호가 없는 정수로 변환합니다.	2-247
데이터 형식 변환 명령	**_TO_***(정수→정수 변환 그룹)	정수→정수 변환 그룹	정수를 다른 데이터 형식의 정수로 변환합니다.	2-250
	_TO_*(정수→비트 열 변환 그룹)	정수→비트 열 변환 그룹	정수를 비트 열로 변환합니다.	2-253
	_TO_*(정수→실수 변환 그룹)	정수→실수 변환 그룹	정수를 실수로 변환합니다.	2-255
	_TO_*(비트 열→정수 변환 그룹)	비트 열→정수 변환 그룹	비트 열을 정수로 변환합니다.	2-257
	_TO_*(비트 열→비트 열 변환 그룹)	비트 열→비트 열 변환 그룹	비트 열을 다른 데이터 형식의 비트 열로 변환합니다.	2-259

종류	명령	명칭	기능 개요	페이지
데이터 형식 변환 명령	**_TO_***(비트 열→실수 변환 그룹)	비트 열→실수 변환 그룹	비트 열을 실수로 변환합니다.	2-261
	_TO_*(실수→정수 변환 그룹)	실수→정수 변환 그룹	실수를 정수로 변환합니다.	2-263
	_TO_*(실수→비트 열 변환 그룹)	실수→비트 열 변환 그룹	실수를 비트 열로 변환합니다.	2-266
	_TO_*(실수→실수 변환 그룹)	실수→실수 변환 그룹	실수를 다른 데이터 형식의 실수로 변환합니다.	2-268
	**_TO_STRING (정수→문자열 변환 그룹)	정수→문자열 변환 그룹	정수를 문자열로 변환합니다.	2-270
	**_TO_STRING (비트 열→문자열 변환 그룹)	비트 열→문자열 변환 그룹	비트 열을 문자열로 변환합니다.	2-272
	**_TO_STRING (실수→문자열 변환 그룹)	실수→문자열 변환 그룹	실수를 문자열로 변환합니다.	2-274
	RealToFormatString	실수(REAL)→서식 있는 문자열 변환	REAL 형식 변수를 서식 지정이 가능한 문자열로 변환합니다.	2-276
	LrealToFormatString	실수(LREAL)→서식 있는 문자열 변환	LREAL 형식 변수를 서식 지정이 가능한 문자열로 변환합니다.	2-281
	STRING_TO_** (문자열→정수 변환 그룹)	문자열→정수 변환 그룹	문자열을 정수로 변환합니다.	2-286
	STRING_TO_** (문자열→비트 열 변환 그룹)	문자열→비트 열 변환 그룹	문자열을 비트 열로 변환합니다.	2-288
	STRING_TO_** (문자열→실수 변환 그룹)	문자열→실수 변환 그룹	문자열을 실수로 변환합니다.	2-290
	TO_**(정수 변환 그룹)	정수 변환 그룹	정수, 비트 열, 실수, 문자열을 정수로 변환합니다.	2-293
	TO_**(비트 열 변환 그룹)	비트 열 변환 그룹	정수, 비트 열, 실수, 문자열을 비트 열로 변환합니다.	2-295
	TO_**(실수 변환 그룹)	실수 변환 그룹	정수, 비트 열, 실수, 문자열을 실수로 변환합니다.	2-297
	EnumToNum	열거 형식→정수 변환	열거 형식을 DINT 형식으로 변환합니다.	2-299
	NumToEnum	정수→열거 형식 변환	DINT 형식을 열거 형식으로 변환합니다.	2-301
	TRUNC	실수 버림	실수의 소수점 이하 1자리를 버리고 정수화합니다.	2-303
	Round	실수 반올림	실수의 소수점 이하 1자리를 반올림해서 정수화합니다.	2-303
	RoundUp	실수 올림	실수의 소수점 이하 1자리를 올려 정수화합니다.	2-303
비트 열 연산 명령	AND, &	논리곱	복수의 부울, 비트 열의 1비트별로 논리곱 연산을 수행합니다.	2-306
	OR	논리합	복수의 부울, 비트 열의 1비트별로 논리합 연산을 수행합니다.	2-306
	XOR	배타적 논리합	복수의 부울, 비트 열의 1비트별로 배타적 논리합 연산을 수행합니다.	2-306
	XORN	배타적 논리합 부정	복수의 부울, 비트 열의 1비트별로 배타적 논리합 부정 연산을 수행합니다.	2-309
	NOT	비트 반전	부울, 비트 열의 각 비트를 반전합니다.	2-311

종류	명령	명칭	기능 개요	페이지
비트 열 연산 명령	AryAnd	배열 논리곱	배열 사이의 각 요소의 부울, 비트 열의 1비트별 논리곱 연산을 수행합니다.	2-312
	AryOr	배열 논리합	배열 사이의 각 요소의 부울, 비트 열의 1비트별 논리합 연산을 수행합니다.	2-312
	AryXor	배열 배타적 논리합	배열 사이의 각 요소의 부울, 비트 열의 1비트별 배타적 논리합 연산을 수행합니다.	2-312
	AryXorN	배열 배타적 논리합 부정	배열 사이의 각 요소의 부울, 비트 열의 1비트별 배타적 논리합 부정 연산을 수행합니다.	2-312
선택 명령	SEL	비트 선택	2개의 선택지 가운데 1개를 선택합니다.	2-316
	MUX	멀티플렉서	2~5개의 선택지 가운데 1개를 선택합니다.	2-318
	LIMIT	상하한 리미트 제한	설정된 상하한값으로 입력 변수의 값을 제한합니다.	2-321
	Band	데드 밴드 제어	데드 밴드(불감대)를 제어합니다.	2-323
	Zone	데드 존 제어	입력값에 바이어스값을 부가합니다.	2-325
	MAX	최대값 검색	2~5개의 수치 가운데 최대값을 검색합니다.	2-327
	MIN	최소값 검색	2~5개의 수치 가운데 최소값을 검색합니다.	2-327
	AryMax	배열 변수의 최대값 검색	1차원 배열 요소의 최대값을 검색합니다.	2-329
	AryMin	배열 변수의 최소값 검색	1차원 배열 요소의 최소값을 검색합니다.	2-329
	ArySearch	배열 검색	1차원 배열에서 지정된 값을 검색합니다.	2-332
데이터 전송 명령	MOVE	전송	변수 및 정수의 값을 다른 변수로 전송합니다.	2-336
	MoveBit	비트 전송	비트 열 내의 1비트를 전송합니다.	2-339
	MoveDigit	자릿수 전송	비트 열 내의 복수 자릿수(1자릿수는 4비트)를 전송합니다.	2-341
	TransBits	다중 비트 전송	비트 열 내의 복수 비트를 전송합니다.	2-343
	MemCopy	메모리 복사	복수의 배열 요소를 전송합니다. 전송원과 전송대상은 동일한 데이터 형식에 한합니다.	2-345
	SetBlock	블록 설정	복수의 배열 요소에 변수 및 정수의 값을 전송합니다.	2-347
	Exchange	데이터 교환	2개의 변수의 값을 교환합니다.	2-349
	AryExchange	배열 데이터 교환	2개의 배열 간의 요소를 교환합니다.	2-351
	AryMove	배열의 전송	복수의 배열 요소를 전송합니다. 전송원과 전송대상의 데이터 형식이 달라도 상관없습니다.	2-353
	Clear	초기화	변수를 초기화합니다.	2-355
	Copy**ToNum(비트 열→부호가 있는 정수)	비트 패턴 복사(비트 열→부호가 있는 정수) 그룹	비트 열의 내용을 그대로 부호가 있는 정수로 복사합니다.	2-357
	Copy**To***(비트 열→실수)	비트 패턴 복사(비트 열→실수) 그룹	비트 열의 내용을 그대로 부호가 있는 실수로 복사합니다.	2-359
	CopyNumTo**(부호가 있는 정수→비트 열)	비트 패턴 복사(부호가 있는 정수→비트 열) 그룹	부호가 있는 정수의 내용을 그대로 비트 열로 복사합니다.	2-361
	CopyNumTo**(부호가 있는 정수→실수)	비트 패턴 복사(부호가 있는 정수→실수) 그룹	부호가 있는 정수의 내용을 그대로 실수로 복사합니다.	2-363
	Copy**To***(실수→비트 열)	비트 패턴 복사(실수→비트 열) 그룹	실수의 내용을 그대로 비트 열로 복사합니다.	2-365

종류	명령	명칭	기능 개요	페이지
데이터 전송 명령	Copy**ToNum (실수→부호가 있는 정수)	비트 패턴 복사 (실수→부호가 있는 정수) 그룹	실수의 내용을 그대로 부호가 있는 정수로 복사합니다.	2-366
이동 명령	AryShiftReg	이동 레지스터	배열 요소로 이루어진 비트 열 전체를 1비트 좌측으로 이동하여 최하위 비트에 입력값을 삽입합니다.	2-370
	AryShiftRegLR	좌우 이동 레지스터	배열 요소로 이루어진 비트 열 전체를 좌측(또는 우측) 방향으로 1비트 이동하여 최하위(또는 최상위) 비트에 입력값을 삽입합니다.	2-372
	ArySHL	배열의 N요소 좌측 이동	배열 요소를 복수 요소 만큼 좌측 방향(상위 방향)으로 이동합니다.	2-375
	ArySHR	배열의 N요소 우측 이동	배열 요소를 복수 요소 만큼 우측 방향(하위 방향)으로 이동합니다.	2-375
	SHL	N비트 좌측 이동	비트 열을 복수 비트 만큼 좌측 방향(상위 방향)으로 이동합니다.	2-378
	SHR	N비트 우측 이동	비트 열을 복수 비트 만큼 우측 방향(하위 방향)으로 이동합니다.	2-378
	NSHLC	N비트 데이터 N 비트 좌측 이동 CY 있음	비트 열 배열을 캐리(CY) 플래그 있음으로 복수 비트 만큼 좌측 방향(상위 방향)으로 이동합니다.	2-380
	NSHRC	N비트 데이터 N 비트 우측 이동 CY 있음	비트 열 배열을 캐리(CY) 플래그 있음으로 복수 비트 만큼 우측 방향(하위 방향)으로 이동합니다.	2-380
	ROL	N비트 좌회전	비트 열을 복수 비트 만큼 좌측 방향(상위 방향)으로 회전합니다.	2-382
	ROR	N비트 우회전	비트 열을 복수 비트 만큼 우측 방향(하위 방향)으로 회전합니다.	2-382
데이터 변환 명령	Swap	바이트 스왑	16비트 값의 상위 바이트와 하위 바이트를 교환합니다.	2-386
	Neg	부호 반전	수치의 부호를 반전합니다.	2-387
	Decoder	비트 디코더	최대 256비트로 이루어진 배열 요소의 지정된 1비트를 TRUE, 기타 비트를 FALSE로 설정합니다.	2-389
	Encoder	비트 엔코더	최대 256비트로 이루어진 배열 요소 가운데, 값이 TRUE인 비트의 위치를 구합니다.	2-392
	BitCnt	비트 카운터	비트 열 내에서 값이 TRUE인 비트의 총수를 카운트합니다.	2-394
	ColmToLine_**	비트 열→비트 행 변환 그룹	각 배열 요소에서, 지정된 위치에 있는 비트의 값을 추출하여 비트 열로서 출력합니다.	2-395
	LineToColm	비트 행→비트 열 변환	비트 열을 분해하여 배열 요소의 지정된 비트 위치로 출력합니다.	2-397
	Gray	그레이 코드 변환	그레이 코드를 각도로 변환합니다.	2-399
	UTF8ToSJIS	문자 코드 변환 (UTF-8 → SJIS)	문자 코드 UTF-8의 문자열을 문자 코드 SJIS의 BYTE 형식 배열로 변환합니다.	2-404
	SJISToUTF8	문자 코드 변환 (SJIS → UTF-8)	문자 코드 SJIS의 BYTE 형식 배열을 문자 코드 UTF-8의 문자열로 변환합니다.	2-404
	PWLApprox	꺾은선 근사 변환(꺾은선 데이터 확인 있음)	꺾은선 데이터가 유효한지 여부를 확인하고 정수, 실수의 꺾은선 근사 계산을 수행합니다.	2-408
	PWLApproxNoLineChk	꺾은선 근사 변환 (꺾은선 데이터 확인 없음)	꺾은선 데이터가 유효한지 여부를 확인하지 않고 정수, 실수의 꺾은선 근사 계산을 수행합니다.	2-408
	PWLLineChk	꺾은선 데이터 확인	꺾은선 근사 변환(꺾은선 데이터 확인 없음) 명령에서 사용하는 꺾은선 데이터가 X 좌표의 오름차순으로 정렬되어 있는지 여부를 판정합니다.	2-408
	MovingAverage	이동 평균	이동 평균값을 산출합니다.	2-416

종류	명령	명칭	기능 개요	페이지
데이터 변환 명령	DispartReal	실수의 가수 · 지수 분리	실수를 부호가 있는 가수부와 지수부로 분리합니다.	2-422
	UniteReal	가수 · 지수를 실수로 결합	부호가 있는 가수부와 지수부를 결합하여 실수로 만듭니다.	2-425
	NumToDec String	고정 길이 10진수 문자열 변환	정수를 고정 길이 10진수 문자열 표현으로 변환합니다.	2-427
	NumToHex String	고정 길이 16진수 문자열 변환	정수를 고정 길이 16진수 문자열 표현으로 변환합니다.	2-427
	HexStringTo Num_**	16진수 문자열→수치 변환 그룹	16진수 문자열 표현을 정수로 변환합니다.	2-430
	FixNumToString	고정 소수점 수→문자열 변환	부호가 있는 고정 소수점 수를 10진수 문자열 표현으로 변환합니다.	2-432
	StringToFixNum	문자열→고정 소수점 수 변환	10진수 문자열을 부호가 있는 고정 소수점 수 표현으로 변환합니다.	2-434
	DtToString	일시→문자열 변환	일시를 문자열 표현으로 변환합니다.	2-437
	DateToString	날짜→문자열 변환	날짜를 문자열 표현으로 변환합니다.	2-439
	TodToString	시각→문자열 변환	시각을 문자열 표현으로 변환합니다.	2-440
	GrayToBin_**	그레이 코드→BIN 코드 변환 그룹	그레이 코드를 비트 열로 변환합니다.	2-442
	BinToGray_**	BIN 코드→그레이 코드 변환 그룹	비트 열을 그레이 코드로 변환합니다.	2-442
	StringToAry	문자열→배열 변환	문자열을 BYTE 형식의 배열로 변환합니다.	2-445
	AryToString	배열→문자열 변환	BYTE 형식 배열을 문자열로 변환합니다.	2-447
	DispartDigit	4비트 분리	비트 열을 4비트 단위로 분리합니다.	2-449
	UniteDigit_**	4비트 결합 그룹	4비트 단위의 데이터를 비트 열로 결합합니다.	2-451
	Dispart8Bit	바이트 단위 데이터 분리	비트 열을 1바이트 단위로 분리합니다.	2-453
	Unite8Bit_**	바이트 단위 데이터 결합 그룹	1바이트 단위의 데이터를 비트 열로 결합합니다.	2-455
	ToAryByte	바이트 배열로 변환	변수를 1바이트 단위로 분해하여 BYTE 형식 배열에 저장합니다.	2-457
	AryByteTo	바이트 배열에서 변환	BYTE 형식 배열 요소를 결합하여 변수에 저장합니다.	2-462
	SizeOfAry	배열 요소 수 취득	배열 요소 수를 취득합니다.	2-467
스택/테이블 명령	StackPush	스택 데이터 저장	스택에 값을 저장합니다.	2-470
	StackFIFO	선입선출	스택의 최하위값을 가져옵니다.	2-480
	StackLIFO	후입선출	스택의 최상위값을 가져옵니다.	2-480
	StackIns	스택 데이터 삽입	스택의 임의의 위치에 값을 삽입합니다.	2-483
	StackDel	스택 데이터 삭제	스택의 임의의 위치의 값을 삭제합니다.	2-485
	RecSearch	레코드 검색	구조체를 요소로 하는 배열로부터, 검색 키와 일치하는 요소를 지정된 방법으로 검색합니다.	2-487
	RecRange Search	범위 지정 레코드 검색	구조체를 요소로 하는 배열로부터, 검색 조건의 범위와 일치하는 요소를 지정된 방법으로 검색합니다.	2-492
	RecSort	레코드 정렬	구조체를 요소로 하는 배열의 요소를 정렬합니다.	2-497
	RecNum	레코드 수 취득	구조체를 요소로 하는 배열의 종단 데이터까지의 레코드 수를 구합니다.	2-502
	RecMax	레코드 최대값 검색	구조체를 요소로 하는 배열로부터, 지정된 멤버의 최대값을 검색합니다.	2-504

종류	명령	명칭	기능 개요	페이지
스택/테이블 명령	RecMin	레코드 최소값 검색	구조체를 요소로 하는 배열로부터, 지정된 멤버의 최소값을 검색합니다.	2-504
FCS 명령	StringSum	SUM값 산출	문자열의 SUM값을 산출합니다.	2-510
	StringLRC	LRC값 산출(문자열)	문자열의 LRC값(수평 패리티)을 산출합니다.	2-512
	String CRCCCITT	CRC-CCITT값 산출(문자열)	문자열의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.	2-514
	StringCRC16	CRC-16값 산출(문자열)	문자열의 CRC-16값을 MODBUS 방식으로 산출합니다.	2-516
	AryLRC_**	LRC값 산출(배열) 그룹	배열의 LRC값을 산출합니다.	2-518
	AryCRCCCITT	CRC-CCITT값 산출(배열)	배열의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.	2-520
	AryCRC16	CRC-16값 산출(배열)	배열의 CRC-16값을 MODBUS 방식으로 산출합니다.	2-522
문자열 명령	CONCAT	문자열 · 연결	2~5개의 문자열을 연결합니다.	2-526
	LEFT	문자열 · 좌측에서 가져오기	문자열의 좌측(선두)으로부터 지정 수의 문자열을 가져옵니다.	2-528
	RIGHT	문자열 · 우측에서 가져오기	문자열의 우측(끝)으로부터 지정 수의 문자열을 가져옵니다.	2-528
	MID	문자열 · 임의의 위치에서 가져오기	문자열의 임의의 위치로부터 지정 수의 문자열을 가져옵니다.	2-530
	FIND	문자열 · 검색	문자열 중에서 지정 문자열의 위치를 검색합니다.	2-532
	LEN	문자열 · 길이 검출	문자열의 문자 수를 구합니다.	2-534
	REPLACE	문자열 · 치환	문자열의 일부를 다른 문자열로 치환합니다.	2-535
	DELETE	문자열 · 삭제	문자열의 일부 또는 전부를 삭제합니다.	2-537
	INSERT	문자열 · 삽입	문자열에 다른 문자열을 삽입합니다.	2-539
	GetByteLen	문자열 · 바이트 수 취득	문자열의 바이트 수를 카운트합니다.	2-541
	ClearString	문자열 · 클리어	문자열을 클리어합니다.	2-542
	ToUCase	문자열 · 대문자 변환	문자열 중의 반각 알파벳을 모두 대문자로 변환합니다.	2-543
	ToLCase	문자열 · 소문자 변환	문자열 중의 반각 알파벳을 모두 소문자로 변환합니다.	2-543
	TrimL	문자열 · 좌측 트림	문자열 중 선두의 공백을 삭제합니다.	2-545
	TrimR	문자열 · 우측 트림	문자열 중 끝의 공백을 삭제합니다.	2-545
	AddDelimiter	구분 문자 부가 문자열 쓰기	구조체 전체의 값을 구분 문자를 부가한 문자열로 변환합니다.	2-547
	SubDelimiter	문자열 읽기 구분 문자 삭제	문자열에서 구분 문자로 구분된 데이터를 읽어서 구조체의 멤버 값으로 저장합니다.	2-558
시간/시각 명령	ADD_TIME	시간 가산	시간과 시간을 가산합니다.	2-570
	ADD_TOD_ TIME	시각과 시간의 가산	시각과 시간을 가산합니다.	2-572
	ADD_DT_TIME	날짜 시각과 시간의 가산	날짜 시각에 시간을 가산합니다.	2-574
	SUB_TIME	시간 감산	시간에서 시간을 감산합니다.	2-576
	SUB_TOD_ TIME	시각과 시간의 감산	시각에서 시간을 감산합니다.	2-578
	SUB_TOD_TOD	시각 감산	시각에서 시각을 감산합니다.	2-580
	SUB_DATE_ DATE	날짜 감산	날짜에서 날짜를 감산합니다.	2-581
	SUB_DT_DT	날짜 시각 감산	날짜 시각에서 날짜 시각을 감산합니다.	2-582
	SUB_DT_TIME	날짜 시각과 시간의 감산	날짜 시각에서 시간을 감산합니다.	2-584
	MULTIME	시간 승산	시간을 지정 승수로 승산합니다.	2-586

종류	명령	명칭	기능 개요	페이지
시간/시각 명령	DIVTIME	시간 제산	시간을 지정 제수로 제산합니다.	2-588
	CONCAT_DATE_TOD	날짜와 시각의 결합	날짜와 시각을 결합합니다.	2-590
	DT_TO_TOD	날짜 시각으로부터 시각 잘라내기	날짜 시각으로부터 시각을 잘라냅니다.	2-592
	DT_TO_DATE	날짜 시각으로부터 날짜 잘라내기	날짜 시각으로부터 날짜를 잘라냅니다.	2-594
	SetTime	시계 보정	시스템 시각을 보정합니다.	2-596
	GetTime	시각 취득	현재 시각을 읽어옵니다.	2-598
	DtToSec	날짜 시각→초 변환	날짜 시각을 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다.	2-600
	DateToSec	날짜→초 변환	날짜를 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다.	2-602
	TodToSec	시각→초 변환	시각을 0시 0분 0초부터의 초 수로 변환합니다.	2-603
	SecToDt	초→날짜 시각 변환	1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜 시각으로 변환합니다.	2-604
	SecToDate	초→날짜 변환	1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜로 변환합니다.	2-606
	SecToTod	초→시각 변환	0시 0분 0초부터의 초 수를 시각으로 변환합니다.	2-608
	TimeToNanoSec	시간→나노초 변환	시간을 나노초 수로 변환합니다.	2-610
	TimeToSec	시간→초 변환	시간을 초 수로 변환합니다.	2-611
	NanoSecToTime	나노초→시간 변환	나노초 수를 시간으로 변환합니다.	2-612
	SecToTime	초→시간 변환	초 수를 시간으로 변환합니다.	2-613
	ChkLeapYear	윤년 판별	지정된 해가 윤년인지 판정합니다.	2-615
	GetDaysOfMonth	달의 일수 취득	지정된 달의 일수를 취득합니다.	2-616
	DaysToMonth	일수→달 변환	1월 1일부터의 일수로부터, 그 날이 몇 월인지 산출합니다.	2-618
	GetDayOfWeek	요일 취득	지정된 연월일의 요일을 취득합니다.	2-620
	GetWeekOfYear	주 취득	지정된 연월일이 그 해의 몇째 주인지 산출합니다.	2-622
	DtToDateStruct	시각 분해	날짜 시각을 '년''월''일''시''분''초''나노초'로 분해합니다.	2-624
	DateStructToDt	시각 결합	'년''월''일''시''분''초''나노초'로 분해된 날짜 시각을 결합합니다.	2-626
	TruncTime	시간 버림	TIME 형식 변수에서 지정된 단위 미만의 값을 버립니다.	2-628
	TruncDt	날짜 시각 버림	DT 형식 변수에서 지정된 단위 미만의 값을 버립니다.	2-632
	TruncTod	시각 버림	TOD 형식 변수에서 지정된 단위 미만의 값을 버립니다.	2-636
아날로그 제어 명령	PIDAT	자동 조정 가능한 PID 연산	자동 조정 가능한 PID 연산(목표값 필터 형식 2자유도 PID 제어)을 수행합니다.	2-642
	TimeProportionalOut	시분할 비례 출력	조작량을 시분할 비례 출력으로 변환합니다.	2-670
	LimitAlarm_**	상하한 경보 그룹	입력값이 하한 설정값 미만의 값 또는 상한 설정값을 초과한 값인 경우에 경보를 출력합니다.	2-686
	LimitAlarmDv_**	상하한 편차 경보 그룹	기준값에서의 입력값의 편차가 하한 편차 설정값 또는 상한 편차 설정값을 초과하는 값인 경우에 경보를 출력합니다.	2-690
	LimitAlarmDvStbySeq_**	대기 시퀀스 부가 상하한 편차 경보 그룹	대기 시퀀스가 있는 상하한 편차 경보를 출력합니다.	2-694
	ScaleTrans	스케일 변환	입력값을 입력 범위에서 출력 범위로 변환합니다.	2-709

종류	명령	명칭	기능 개요	페이지
아날로그 제어 명령	AC_Step Program	스텝 프로그램	지정된 프로그램 패턴에 따라 태스크 주기마다 현재 목표값과 예측 목표값을 산출합니다.	2-712
시스템 제어 명령	TraceSamp	데이터 트레이스 샘플링	데이터 트레이스 샘플링을 실행합니다.	2-738
	TraceTrig	데이터 트레이스 트리거 조건 성립	데이터 트레이스의 트리거를 발생시킵니다.	2-741
	GetTraceStatus	데이터 트레이스 Status 읽어오기	데이터 트레이스의 실행 Status를 읽어옵니다.	2-744
	SetAlarm	사용자 이상 발생	사용자 이상을 발생시킵니다.	2-747
	ResetAlarm	사용자 이상 해제	사용자 이상을 해제합니다.	2-752
	GetAlarm	사용자 이상 Status 취득	발생 중인 사용자 이상에서 최고 중요 이벤트의 중요도(사용자 이상 레벨 1~8)와 최고 중요 이벤트 코드를 취득합니다.	2-754
	ResetPLCError	PLC 이상 해제	PLC 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.	2-756
	GetPLCError	PLC 이상 Status 취득	PLC 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.	2-760
	ResetCJBError	I/O 버스 이상 해제	I/O 버스에서 발생 중인 컨트롤러 이상을 해제합니다.	2-762
	GetCJBError	I/O 버스 이상 Status 취득	I/O 버스에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.	2-762
	GetEIPError	EtherNet/IP 이상 Status 취득	EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.	2-766
	ResetMCError	모션 제어 이상 해제	모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.	2-768
	GetMCError	모션 제어 이상 Status 취득	모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.	2-773
	ResetECError	EtherCAT 이상 해제	EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.	2-775
	GetECError	EtherCAT 이상 Status 취득	EtherCAT 마스터 기능 모듈에서 발생 중인 이상을 검출합니다.	2-777
	SetInfo	사용자 정보 발생	사용자 정보를 발생시킵니다.	2-780
	ResetUnit	유니트 재시작	CPU 고기능 유니트 또는 고기능 I/O 유니트를 재시작합니다.	2-782
	GetNTPStatus	NTP Status 읽어오기	NTP의 Status를 읽어옵니다.	2-786
	RestartNXUnit	NX 유니트 재시작	통신 커플러 유니트 또는 NX 유니트를 재시작합니다.	2-788
	NX_Change WriteMode	NX 유니트 쓰기 모드 변경	통신 커플러 유니트 또는 NX 유니트를 데이터를 쓸 수 있는 모드로 변경합니다.	2-793
	NX_SaveParam	NX 유니트 파라메터 저장	통신 커플러 유니트 또는 NX 유니트에 기록한 데이터를 저장합니다.	2-797
EtherCAT 통신 명령	EC_CoESDO Write	CoE SDO 쓰기	EtherCAT 네트워크상의 지정된 슬레이브가 가진 CoE 개체에 값을 씁니다.	2-804
	EC_CoESDO Read	CoE SDO 읽기	EtherCAT 네트워크상의 지정된 슬레이브가 가진 CoE 개체에서 값을 읽습니다.	2-807
	EC_StartMon	EtherCAT 패킷 모니터 기능 시작	EtherCAT 통신의 패킷 모니터 기능을 실행합니다.	2-813
	EC_StopMon	EtherCAT 패킷 모니터 기능 정지	EtherCAT 통신의 패킷 모니터 기능을 정지합니다.	2-819

종류	명령	명칭	기능 개요	페이지
EtherCAT 통신 명령	EC_SaveMon	EtherCAT 패킷 저장	EtherCAT 통신의 패킷 데이터를 CPU 유니트의 메인 메모리의 내부 파일에 저장합니다.	2-821
	EC_CopyMon	EtherCAT 패킷 전송	EtherCAT 통신에서 CPU 유니트의 메인 메모리상의 내부 파일에 있는 패킷 데이터를 SD 메모리 카드에 전송합니다.	2-823
	EC_Disconnect Slave	EtherCAT 슬레이브 이탈	EtherCAT 네트워크상의 지정된 슬레이브를 네트워크에서 이탈시킵니다.	2-825
	EC_Connect Slave	EtherCAT 슬레이브 재가입	EtherCAT 네트워크상의 지정된 슬레이브를 네트워크에 재가입시킵니다.	2-833
	EC_Change EnableSetting	EtherCAT 슬레이브의 유효/무효 전환	EtherCAT 슬레이브의 유효/무효를 전환합니다.	2-835
	NX_WriteObj	NX 개체 쓰기	통신 커플러 유니트 또는 NX 유니트의 NX 개체에 데이터를 씁니다.	2-854
	NX_ReadObj	NX 개체 읽기	통신 커플러 유니트 또는 NX 유니트의 NX 개체에서 데이터를 읽습니다.	2-869
EtherNet/IP 통신 명령	CIPOpen	CIPClass3(Large_Forward_Open) 연결 확립	지정된 상대 노드에 대해 CIPClass3(Large_Forward_Open) 연결을 확립합니다.	2-878
	CIPOpenWith DataSize	CIP Class3 연결 확립(크기 지정 있음)	지정된 상대 노드에 대해, 지정된 데이터 길이 이하의 Class3 Explicit 메시지를 송수신할 수 있는 CIP Class3 연결을 확립합니다	2-887
	CIPRead	변수 읽기(Class3 Explicit 메시지)	CIP 네트워크상의 타국 컨트롤러의 변수 값을 읽습니다(Class3 Explicit 메시지).	2-890
	CIPWrite	변수 쓰기(Class3 Explicit 메시지)	CIP 네트워크상의 타국 컨트롤러의 변수에 값을 씁니다(Class3 Explicit 메시지).	2-895
	CIPSend	임의 Explicit 메시지 송신(Class3)	CIP 네트워크상의 지정된 장치에 Class3의 CIP 메시지를 송신합니다.	2-901
	CIPClose	CIP Class3 연결 중단	지정된 핸들의 CIP Class3 연결을 중단합니다.	2-905
	CIPUCMMRead	변수 읽기(UCMM Explicit 메시지)	지정된 CIP 네트워크상의 타국 컨트롤러의 변수 값을 읽습니다(UCMM Explicit 메시지).	2-907
	CIPUCMMWrite	변수 쓰기(UCMM Explicit 메시지)	CIP 네트워크상의 타국 컨트롤러의 변수에 값을 씁니다(UCMM Explicit 메시지).	2-912
	CIPUCMMSend	임의 Explicit 메시지 송신(UCMM)	CIP 네트워크상의 지정된 장치에 UCMM의 CIP 메시지를 송신합니다.	2-919
	SktUDPCreate	UDP 소켓 생성	내장 EtherNet/IP의 UDP 소켓 생성 요구(서버 포트의 오픈)를 수행합니다.	2-929
	SktUDPRcv	UDP 소켓 수신	내장 EtherNet/IP의 UDP 소켓의 수신 버퍼 내의 데이터를 읽습니다.	2-936
	SktUDPSend	UDP 소켓 송신	내장 EtherNet/IP의 UDP 포트에서 데이터를 송신합니다.	2-939
	SktTCPAccept	TCP 소켓 허용	내장 EtherNet/IP의 UDP 소켓 허용 요구를 수행합니다.	2-942
	SktTCPConnect	TCP 소켓 연결	내장 EtherNet/IP에서 상대 TCP 포트에 연결합니다.	2-945
	SktTCPRcv	TCP 소켓 수신	내장 EtherNet/IP의 지정된 TCP 소켓의 수신 버퍼 내의 데이터를 읽습니다.	2-953
	SktTCPSend	TCP 소켓 송신	내장 EtherNet/IP의 지정된 TCP 포트에서 데이터를 송신합니다.	2-956

종류	명령	명칭	기능 개요	페이지
EtherNet/IP 통신 명령	SktGetTCP Status	TCP 소켓의 Status 읽기	TCP 소켓의 Status를 읽어옵니다.	2-959
	SktClose	TCP/UDP 소켓 클로즈	내장 EtherNet/IP의 지정된 TCP 소켓 또는 UDP 소켓을 클로즈합니다.	2-962
	SktClearBuf	TCP/UDP 소켓 수신 버퍼 클리어	내장 EtherNet/IP의 지정된 TCP 소켓 또는 UDP 소켓의 수신 버퍼를 클리어합니다.	2-965
	ChangeIPAdr	IP 주소 변경	내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 IP 주소를 변경합니다.	2-968
	ChangeFTP Account	FTP 계정 변경	내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 FTP 로그인명과 비밀번호를 변경합니다.	2-977
	ChangeNTP ServerAdr	NTP 서버 수신처 변경	내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 NTP 서버 수신처를 변경합니다.	2-980
시리얼 통신 명령	ExecPMCR	프로토콜 매크로	시리얼 커뮤니케이션 유니트에 등록된 송수신 시퀀스(프로토콜 데이터)의 실행을 의뢰합니다.	2-986
	SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	시리얼 커뮤니케이션 유니트의 시리얼 포트에서 무수순으로 데이터를 송신합니다.	2-1000
	SerialRcv	시리얼 커뮤니케이션 유니트 시리얼 포트 입력	시리얼 커뮤니케이션 유니트의 시리얼 포트에서 무수순의 수신 데이터를 읽습니다. 읽은 후, 수신 버퍼는 전체 클리어됩니다.	2-1011
	SerialRcvNo Clear	시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함	시리얼 커뮤니케이션 유니트의 시리얼 포트에서 무수순의 수신 데이터를 읽습니다. 읽은 후, 수신 버퍼는 전체 클리어되지 않습니다.	2-1011
	SendCmd	명령 송신	시리얼 커뮤니케이션 유니트에 대해 시리얼 게이트웨이 기능으로 명령을 송신합니다. 또는 DeviceNet 유니트나 CompoNet 마스터 유니트에 대해 Explicit 명령을 송신합니다.	2-1026
SD 메모리 카드 명령	FileWriteVar	변수 파일 쓰기	SD 메모리 카드 내의 지정된 파일에 1개의 변수값을 바이너리 형식으로 써넣습니다.	2-1040
	FileReadVar	변수 파일 읽기	SD 메모리 카드 내의 지정된 파일의 값을 바이너리 형식으로 읽어 변수에 써넣습니다.	2-1046
	FileOpen	파일 열기	SD 메모리 카드 내의 지정된 파일을 엽니다.	2-1051
	FileClose	파일 닫기	SD 메모리 카드 내의 지정된 파일을 닫습니다.	2-1055
	FileSeek	파일 검색	SD 메모리 카드 내의 지정된 파일에 파일 위치 지시자를 설정합니다.	2-1058
	FileRead	파일 읽기	SD 메모리 카드 내의 지정된 파일의 데이터를 읽어옵니다.	2-1061
	FileWrite	파일 쓰기	SD 메모리 카드 내의 지정된 파일에 데이터를 써넣습니다.	2-1069
	FileGets	문자열 읽기	SD 메모리 카드 내의 지정된 파일로부터 1행의 문자열을 읽어옵니다.	2-1077
	FilePuts	문자열 쓰기	SD 메모리 카드 내의 지정된 파일에 문자열을 써넣습니다.	2-1084
	FileCopy	파일 복사	SD 메모리 카드 내의 지정된 파일을 복사합니다.	2-1092
	FileRemove	파일 삭제	SD 메모리 카드 내의 지정된 파일을 삭제합니다.	2-1100

종류	명령	명칭	기능 개요	페이지
SD 메모리 카드 명령	FileRename	파일명 변경	SD 메모리 카드 내의 지정된 파일 및 디렉토리의 명칭을 변경합니다.	2-1105
	DirCreate	디렉토리 작성	SD 메모리 카드 내에 지정된 명칭의 디렉토리를 작성합니다.	2-1110
	DirRemove	디렉토리 삭제	SD 메모리 카드 내의 지정된 디렉토리를 삭제합니다.	2-1113
타임 스탬프 명령	NX_DOut TimeStamp	타임 스탬프 디지털 출력 쓰기	타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점에 값을 씁니다.	2-1118
	NX_AryDOut TimeStamp	타임 스탬프 디지털 출력 배열 쓰기	타임 스탬프 방식 대응 디지털 출력 유니트에서 펄스를 출력시킵니다.	2-1124
기타 명령	ReadNbit_**	N비트 읽기 그룹	비트 열 내의 복수 비트를 읽어옵니다.	2-1132
	WriteNbit_**	N비트 쓰기 그룹	비트 열 내에 복수 비트를 써넣습니다.	2-1134
	ChkRange	범위형 변수 확인	변수의 값이 범위형 지정의 유효 범위 내에 있는지 판정합니다.	2-1136
	GetMyTask Status	로컬 태스크 Status 읽기	로컬 태스크의 태스크 Status를 읽어옵니다.	2-1138
	Task_IsActive	태스크 실행 중 판정	지정된 태스크가 실행 중인지 판정합니다.	2-1141
	Lock	태스크 간 배타적 잠금	태스크 간 배타적 잠금을 시작합니다. 다른 태스크의 동일한 잠금 번호 구간은 실행할 수 없게 됩니다.	2-1143
	Unlock	태스크 간 배타적 잠금 해제	태스크 간 배타적 잠금을 종료합니다.	2-1143
	ActEventTask	이벤트 태스크 기동	이벤트 태스크를 기동합니다.	2-1148
	Get**Clk	클록 펄스 취득 그룹	지정된 주기의 클록 펄스를 출력합니다.	2-1154
	Get**Cnt	가산 프리 러닝 카운터 취득 그룹	지정된 주기의 프리 러닝 카운터값을 취득합니다.	2-1155

- 모션 제어 명령의 사양은 "NJ 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.
- 시뮬레이션 명령의 사양은 "Sysmac Studio Version1 조작 설명서(SBCA-362)"를 참조하십시오.

2

각 명령에 대한 설명

이 장에서는 NJ 시리즈에서 사용할 수 있는 명령의 사양을 설명합니다.

이 장을 보는 방법 **2-3**
래더도 명령 **2-15**
ST 구문 명령 **2-27**
시퀀스 입력 명령 **2-43**
시퀀스 출력 명령 **2-49**
시퀀스 제어 명령 **2-63**
비교 명령 **2-93**
타이머 명령 **2-123**
카운터 명령 **2-141**
산술 명령 **2-161**
BCD 변환 명령 **2-229**
데이터 형식 변환 명령 **2-249**
비트 열 연산 명령 **2-305**
선택 명령 **2-315**
데이터 전송 명령 **2-335**
이동 명령 **2-369**
데이터 변환 명령 **2-385**
스택/테이블 명령 **2-469**
FCS 명령 **2-509**
문자열 명령 **2-525**
시간/시각 명령 **2-569**
아날로그 제어 명령 **2-641**
시스템 제어 명령 **2-737**
EtherCAT 통신 명령 **2-803**
EtherNet/IP 통신 명령 **2-877**
시리얼 통신 명령 **2-985**

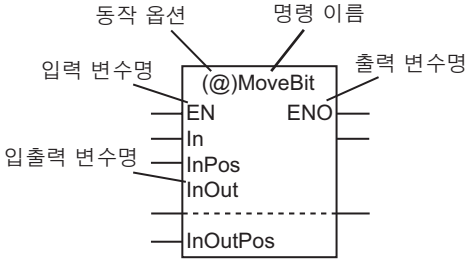
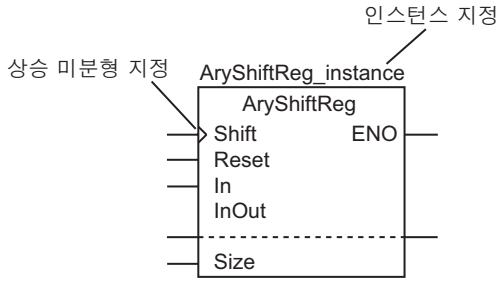
SD 메모리 카드 명령 2-1039
타임 스탬프 명령 2-1117
기타 명령 2-1131

이 장을 보는 방법

여기서는 이 장(각 명령에 대한 설명)의 기재 항목과 보는 방법을 설명합니다.

기재 항목

기재 항목은 다음과 같습니다.

항목	내용
명령	명령어의 명칭을 나타냅니다. 예: MoveBit
명칭	명령의 일본어 명칭을 나타냅니다. 예: 비트 전송
FB/FUN	명령이 펑션 블록(FB) 형식인지 펑션(FUN) 형식인지를 나타냅니다. FB 형식 명령은 프로그램 또는 FB에서만 호출할 수 있습니다. FUN 형식 명령은 프로그램, FB, FUN에서 모두 호출할 수 있습니다.
그래픽 표현	해당 명령을 래더도 프로그램 내에서 표현할 때의 그림을 나타냅니다. ●FUN 형식의 예  ●FB 형식의 예  이 중에서 '동작 옵션', '상승 미분 형식 지정', '인스턴스 지정'의 의미는 다음과 같습니다. 동작 옵션 : FUN 형식의 명령 이름 앞에 붙는 (@)을 의미합니다. 동작 옵션이 있는 명령은 명령 이름 앞에 @을 부가하여 입력 상승 미분 형식으로 만들 수 있습니다. 입력 상승 미분 형식이란, 실행 조건의 입력 변수(EN)의 값이 이전 태스크 주기에서 FALSE, 현재 태스크 주기에서 TRUE인 경우에 실행되는 명령을 의미합니다. 상승 미분 형식 지정: 입력 변수의 핀에 적혀 있는 화살표(->)를 의미합니다.이 지정이 있는 명령은 입력 상승 미분 형식으로 동작합니다. 인스턴스 지정 : FB 형식 명령 위에 적혀 있는 'XX_instance'를 의미합니다. 인스턴스 지정이 적혀 있는 명령에는 임의의 인스턴스명을 붙여야 합니다.
ST 표현	해당 명령을 ST 프로그램 내에서 표현할 때의 기술을 나타냅니다. ST 프로그램 내에서 명령을 기술하는 데는 다음과 같은 2가지 방법이 있습니다. 1. 변수(입력 변수, 출력 변수, 입출력 변수)명과 파라메터명의 대응을 명시하는 방법 예: MoveBit(In: =abc, InPos: =def, InOut: =ghi, InOutPos: =jkl); 2. 변수(입력 변수, 출력 변수, 입출력 변수)명은 생략하고 파라메터명만을 기술하는 방법 예: MoveBit(In, InPos, InOut, InOutPos); 이 장에서는 2.)의 방법으로 기술합니다. XX_instance(변수명)이라고 적혀 있는 명령에는 임의의 인스턴스명을 붙여야 합니다. 예: TON_instance (In, PT, Q, ET);

항목	내용
변수	• 변수명 입력 변수, 출력 변수, 입출력 변수의 명칭을 나타냅니다. 예: In1 단, 복수의 명령에 공통으로 사용되는 변수는 각 명령에 대한 설명 페이지에 기재하지 않았습니다. 공통으로 사용되는 변수란 다음 8가지 변수를 말합니다. 각 변수의 사양은 뒤에 다룹니다. (EN, ENO, Execute, Done, Busy, Error, ErrorID, ErrorIDEx) • 명칭 변수의 일본어 명칭을 나타냅니다. 예: 가산 카운터 • 입/출력 입력 변수, 출력 변수, 입출력 변수 중 어느 것인지를 나타냅니다. • 내용 해당 변수의 의미 및 제한을 나타냅니다. • 유효 범위 해당 변수의 취득값의 범위를 나타냅니다. '데이터 형식에 따름'이라고 적혀 있는 경우, 그 변수의 데이터 형식의 유효 범위에 따릅니다. 각 데이터 형식의 유효 범위는 뒤에 다룹니다. • 단위 해당 변수의 단위를 나타냅니다. '-'라고 적혀 있는 경우는 단위가 없습니다. 예: 바이트 • 초기값 변수에 파라메터를 대입하지 않고 명령을 실행한 경우, 그 변수에 자동적으로 설정되는 값을 의미합니다. '-'라고 적혀 있는 경우는 단위가 없습니다. 입력 변수의 경우 : 해당 입력 변수의 데이터 형식 기본값의 초기값이 대입됩니다. 각 데이터 형식 기본값의 초기값은 뒤에 다룹니다. 입력 변수가 구조체인 경우, 기능 설명 중의 구조체 사양표에 초기값이 게재되어 있습니다. 출력 변수의 경우 : 초기값은 설정되어 있지 않습니다. 입출력 변수의 경우 : 초기값은 설정되어 있지 않습니다. • 데이터 형식 해당 변수의 데이터 형식을 나타냅니다. 열거 형식, 배열, 구조체, 공용체가 적용되는 경우, 해당 사실을 명기했습니다.
기능	해당 명령의 기능을 나타냅니다. 본문 중 변수명은 작은 따옴표 ' '에 넣어 표기합니다. 예: 'In1' 또한 배열명은 대괄호 [ ]에 넣어 표기합니다. 예: InOut[]
관련된 시스템 정의 변수	해당 명령과 관련이 있는 시스템 정의 변수의 명칭과 개요를 나타냅니다. 시스템 정의 변수의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하시기 바랍니다.
관련된 준 사용자 정의 변수	해당 명령과 관련이 있는 준 사용자 정의 변수의 명칭과 개요를 나타냅니다. 준 사용자 정의 변수의 자세한 내용은 해당 페이지에 기재된 참조 설명서를 참조하시기 바랍니다.
참고	해당 명령의 기능에 대한 보충 정보를 나타냅니다. 관련된 명령 및 그 명령의 편리한 사용 방법 등을 기술합니다.
사용상의 주의 사항	해당 명령을 사용할 때의 주의 사항을 나타냅니다. 어떤 경우에 이상이 발생하는지도 아울러 설명합니다.
샘플 프로그램	해당 명령을 포함하는 짧은 응용 프로그램의 예를 나타냅니다. 동일한 처리 내용으로 래더도 프로그램과 ST 프로그램 양쪽을 모두 기술합니다.

공통 변수

복수의 명령에 공통으로 사용되는 변수(EN, ENO, Execute, Done, Busy, Error, ErrorID, ErrorIDEx)의 사양을 설명합니다.
이러한 공통 변수는 각 명령의 변수표에는 기재되어 있지 않습니다. 각 명령이 이러한 공통 변수를 갖는지의 여부는 그래픽 표현 및 ST 표현 항목에서 확인하시기 바랍니다.

EN

EN은 FUN 형식 명령의 실행 조건을 나타내는 입력 변수입니다.
래더도 프로그램에서 FUN 형식 명령을 사용하는 경우, 실행 조건을 EN에 입력(결선)하시기 바랍니다.

	명칭	입/출력	내용	데이터 형식	유효 범위	초기값
EN	ENABLE (실행 조건)	입력	TRUE : 명령을 실행함(*) FALSE : 명령을 실행하지 않음	BOOL	TRUE, FALSE	TRUE

* 동작 옵션에서 입력 상승 미분 형식(@)으로 지정된 경우, EN의 값이 FALSE에서 TRUE로 변화하는 것이 실행 조건입니다.

- FB 형식 명령에는 EN이 없습니다.
- ST 프로그램에서 FUN 형식 명령을 읽어오는 경우, EN의 표기를 생략하시기 바랍니다. 명령 실행 조건은 ST 프로그램의 동작 시퀀스에서 결정하므로 EN은 필요 없습니다.

ENO

ENO는 래더도 프로그램에서 실행 조건을 다음 단계의 명령으로 전달하기 위한 출력 변수입니다.
통상적으로는 명령이 정상 종료된 시점에서 ENO의 값이 TRUE가 됩니다. 그 값을 받아 다음 단계의 명령이 실행되기 시작합니다.

	명칭	입/출력	내용	데이터 형식	유효 범위	초기값
ENO	ENABLE 출력	출력	TRUE : 정상 종료(*) FALSE : 이상 종료, 실행 중, 또는 실행 조건 비성립	BOOL	TRUE, FALSE	–

* 실행 조건이 성립된 경우에 한합니다. 정상 종료 후, 실행 조건이 성립되지 않았다면 그 시점에서 ENO의 값은 FALSE가 됩니다.

- ENO는 FUN 형식 명령과 FB 형식 명령에 모두 존재합니다. 하지만 명령에 따라서는 ENO를 갖지 않는 것도 있습니다.
- ST 프로그램에서는 ENO의 표기를 생략하시기 바랍니다. 다음 단계의 명령 실행 조건은 ST 프로그램의 동작 시퀀스에서 결정하므로 ENO는 필요 없습니다.

Execute, Done, Busy

Execute는 FB 형식 명령의 일부가 갖는, 실행 조건을 나타내는 입력 변수입니다.
Execute의 값이 FALSE에서 TRUE로 변화하면 명령이 실행됩니다. 일단 명령이 실행되면, Execute의 값이 FALSE가 되거나 명령 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다.

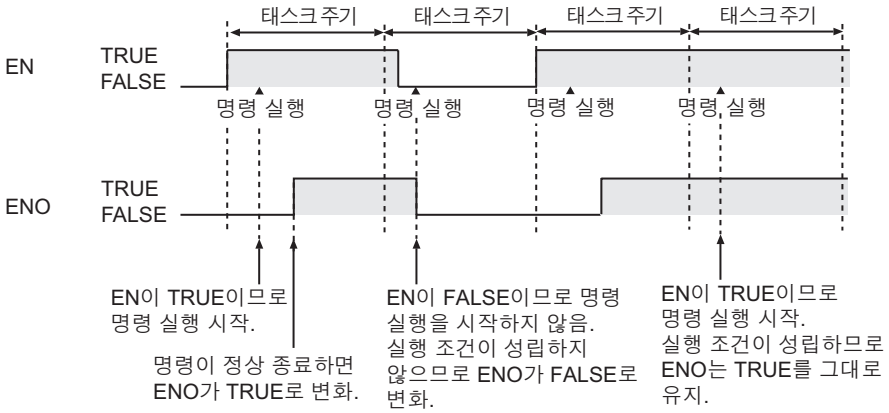
Done은 FB 형식 명령의 일부가 갖는, 명령이 정상 종료되었음을 나타내는 출력 변수입니다.
Busy는 FB 형식 명령의 일부가 갖는, 명령이 실행 중임을 나타내는 출력 변수입니다.

	명칭	입/출력	내용	데이터 형식	유효 범위	초기값
Execute	기동	입력	TRUE : 명령을 실행함(*1) FALSE : 명령을 실행하지 않음 (*2)	BOOL	TRUE, FALSE	FALSE
Done	완료	출력	TRUE : 정상 종료(*3)(*4) FALSE : 이상 종료, 실행 중, 또는 실행 조건 비성립	BOOL	TRUE, FALSE	–
Busy	실행 중		TRUE : 실행 중 FALSE : 실행 중이 아님			

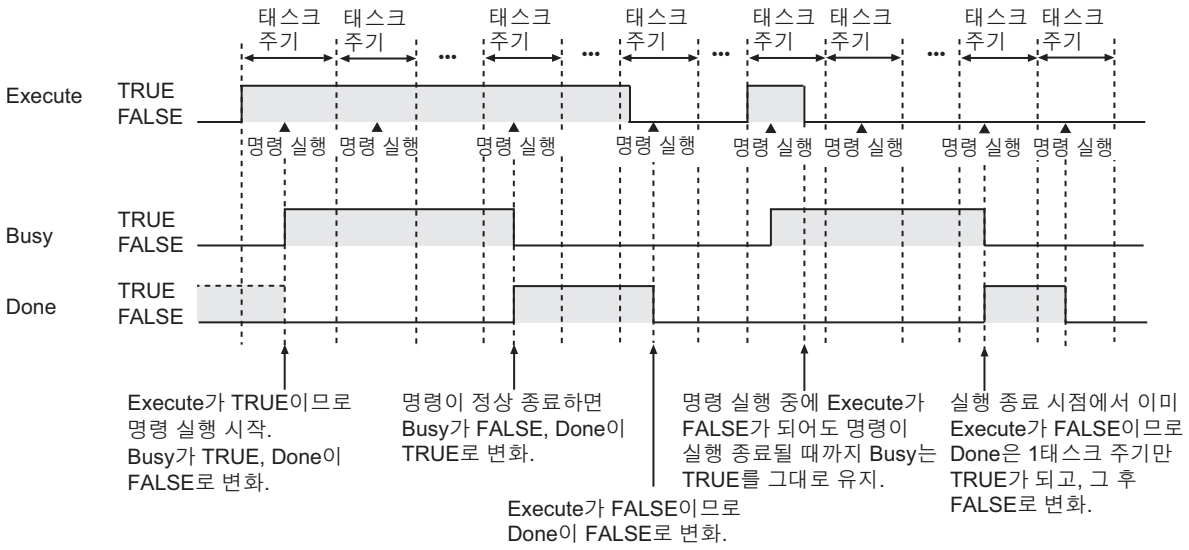
*1 운전을 시작했을 때 이미 Execute의 값이 TRUE인 경우, 명령을 실행하지 않습니다. 명령을 실행시키려면 Execute의 값을 일단 FALSE로 바꾸시기 바립니다.
*2 명령 실행 중에 값이 FALSE가 되더라도 처리를 끝까지 계속합니다.
*3 정상 종료 후, 실행 조건이 성립되지 않았다면 그 시점에서 Done의 값은 FALSE가 됩니다.
*4 정상 종료된 시점에서 이미 실행 조건이 비성립 상태라면, 1태스크 주기 동안 Done의 값이 TRUE가 되며, 그 후에는 FALSE가 됩니다.

EN 또는 ENO가 있는 명령, 즉 1태스크 주기에 처리가 완료되는 명령과 Execute 또는 Busy가 있는 명령, 즉 복수의 태스크 주기에 걸쳐 처리가 이루어지는 명령의 타이밍 차트는 다음과 같습니다.

● 1태스크 주기에 처리가 완료되는 명령



● **복수의 태스크 주기에 걸쳐 처리가 이루어지는 명령**



Error, ErrorID, ErrorIDEx

Error, ErrorID, ErrorIDEx는 FB 형식 명령의 일부가 갖는, 명령이 이상 종료되었음을 나타내는 출력 변수입니다.

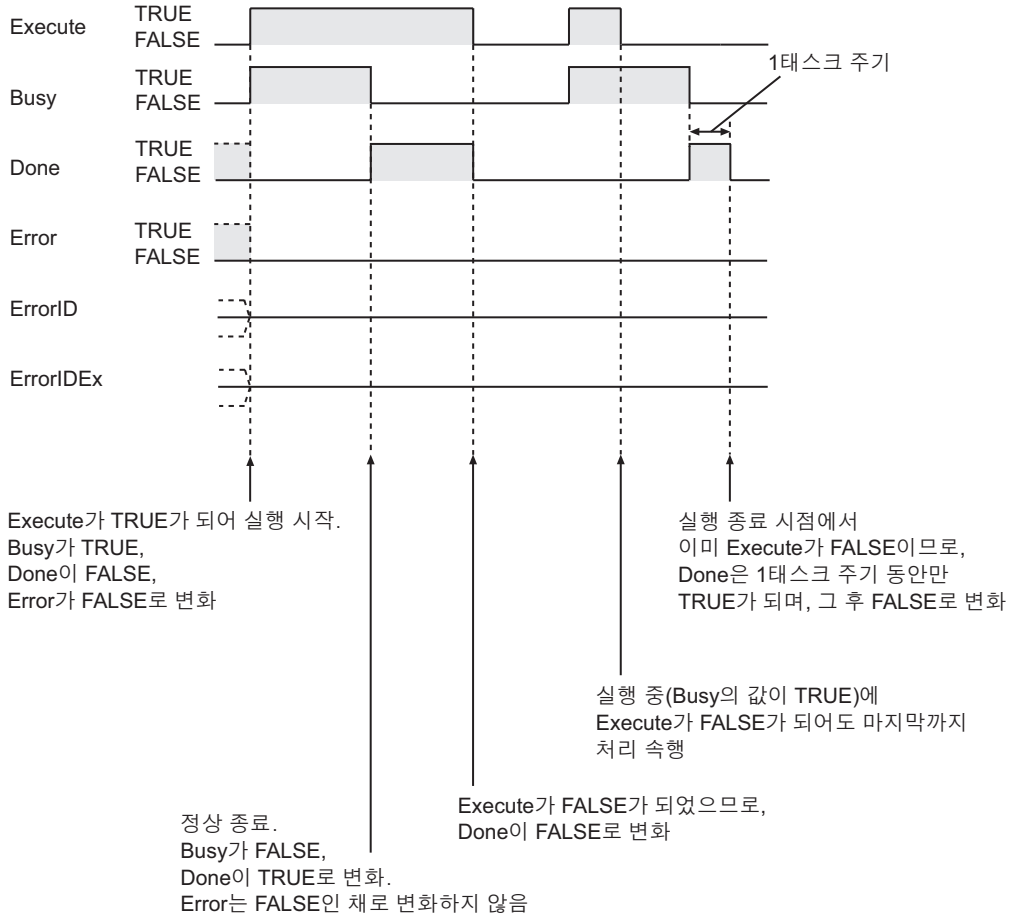
	명칭	입/출력	내용	데이터 형식	유효 범위	초기값
Error	에러	출력	TRUE : 이상 종료(*1) (*2) FALSE : 정상 종료, 실행 중, 또는 실행 조건 비성립	BOOL	TRUE, FALSE	–
ErrorID	에러 코드		이상 종료 시의 이상 ID 정상 종료 시는 WORD#16#0	WORD	명령에 따라 다름	
ErrorIDEx	확장 에러 코드		이상 종료 시의 확장 이상 ID 정상 종료 시는 DWORD#16#0	DWORD		

*1 이상 종료 후, 실행 조건이 성립되지 않았다면 그 시점에서 Error의 값은 FALSE가 됩니다.

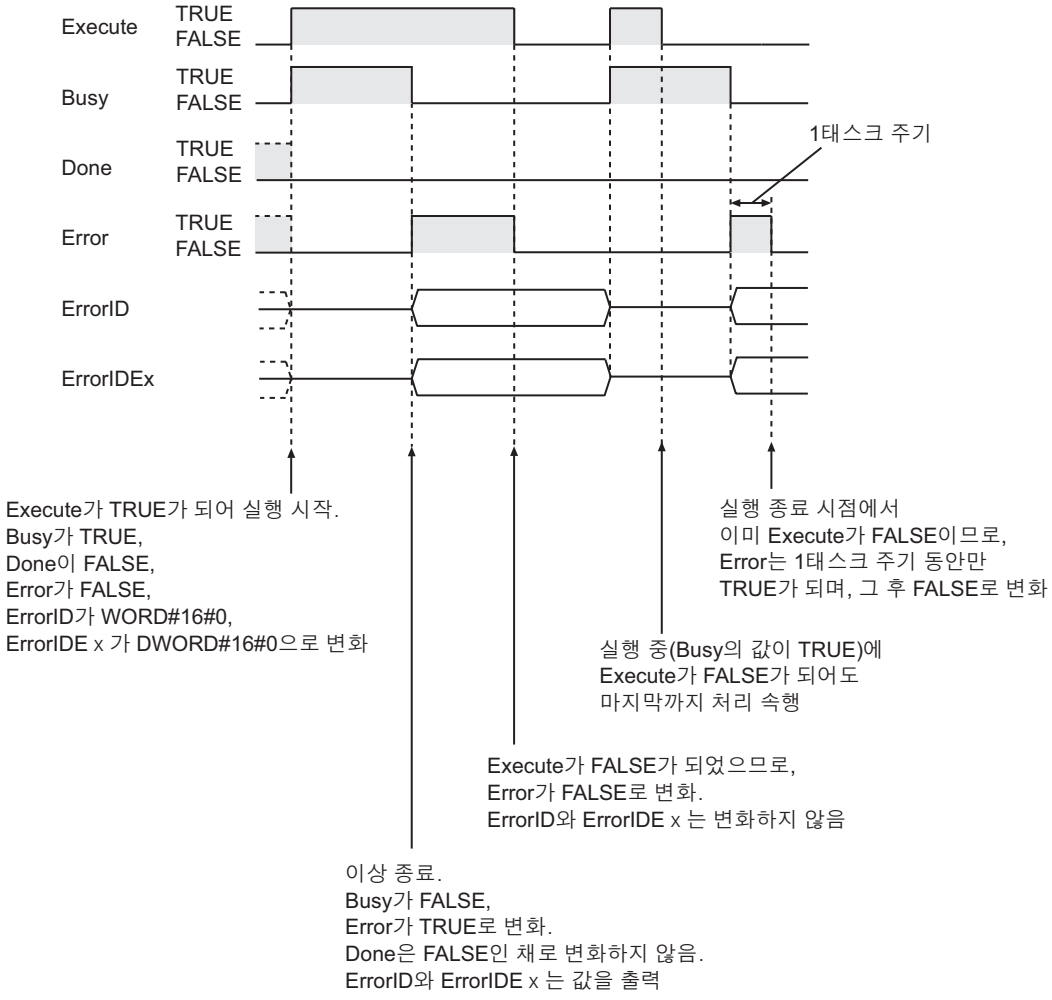
*2 이상 종료된 시점에서 이미 실행 조건이 비성립 상태라면, 1태스크 주기 동안 Error의 값이 TRUE가 되며, 그 후에는 FALSE가 됩니다.

Execute, Done, Busy, Error, ErrorID, ErrorIDEx의 타이밍 차트가 아래에 나와 있습니다.

● 정상 종료인 경우


Execute
TRUE
FALSE
Busy
TRUE
FALSE
Done
TRUE
FALSE
Error
TRUE
FALSE
ErrorID
ErrorIDEx
1태스크 주기
Execute가 TRUE가 되어 실행 시작.
Busy가 TRUE,
Done이 FALSE,
Error가 FALSE로 변화
정상 종료.
Busy가 FALSE,
Done이 TRUE로 변화.
Error는 FALSE인 채로 변화하지 않음
Execute가 FALSE가 되었으므로,
Done이 FALSE로 변화
실행 중(Busy의 값이 TRUE)에
Execute가 FALSE가 되어도 마지막까지
처리 속행
실행 종료 시점에서
이미 Execute가 FALSE이므로,
Done은 1태스크 주기 동안만
TRUE가 되며, 그 후 FALSE로 변화

● **이상 종료인 경우**



변수의 유효 범위와 기본값의 초기값

변수의 유효 범위란, 그 변수가 취득할 수 있는 값의 범위를 말합니다. 또한 변수의 기본값의 초기값이란, 입력 변수에 파라메터를 대입하지 않고 명령을 실행한 경우에 그 입력 변수에 대입되는 값을 말합니다. 이 값들은 데이터 형식별로 정해져 있습니다. 명령으로 특별히 지정되지 않은 경우, 변수의 유효 범위와 초기값은 그 값에 따릅니다.

이 장의 변수표에서 유효 범위 항목에 '데이터 형식에 따름', 입력 변수의 초기값 항목에 '-'라고 적혀 있는 것이 이에 해당합니다.

각 데이터 형식의 유효 범위와 초기값이 아래에 나와 있습니다.

분류	데이터 형식	유효 범위	초기값
부울	BOOL	FALSE, TRUE	FALSE
비트 열	BYTE	BYTE#16#00~FF	BYTE#16#00
	WORD	WORD#16#0000~FFFF	WORD#16#0000
	DWORD	DWORD#16#00000000~FFFFFFFF	DWORD#16#00000000
	LWORD	LWORD#16#0000000000000000~ FFFFFFFFFFFFFFFF	LWORD#16#0000000000000000
정수	USINT	USINT#0~+255	USINT#0
	UINT	UINT#0~+65535	UINT#0
	UDINT	UDINT#0~+4294967295	UDINT#0
	ULINT	ULINT#0~+18446744073709551615	ULINT#0
	SINT	SINT#-128~+127	SINT#0
	INT	INT#-32768~+32767	INT#0
	DINT	DINT#-2147483648~+2147483647	DINT#0
	LINT	LINT#-9223372036854775808~ +9223372036854775807	LINT#0
실수	REAL	REAL#-3.402823e+38~-1.175495e-38, 0, +1.175495e-38~+3.402823e+38, +∞/-∞	REAL#0
	LREAL	LREAL#-1.79769313486231e+308~ -2.22507385850721e-308, 0, +2.22507385850721e-308~ +1.79769313486231e+308, +∞/-∞	LREAL#0
시각, 지속 시간, 날짜, 문자열	TIME	T#-9223372036854.775808ms (T#-106751d_23h_47m_16s_854.775808ms)~ T#9223372036854.775807ms (T#+106751d_23h_47m_16s_854.775807ms)	T#0s
	DATE	D#1970-01-01~D#2106-02-06 (1970년 1월 1일~2106년 2월 6일)	D#1970-01-01
	TOD	TOD#00:00:00.000000000~ TOD#23:59:59.999999999 (0시 0분 0.000000000초~ 23시 59분 59.999999999초)	TOD#00:00:00.000000000
	DT	DT#1970-01-01-00:00:00.000000000~ DT#2106-02-06-23:59:59.999999999 (1970년 1월 1일 0시 0분 0.000000000초~ 2106년 2월 6일 23시 59분 59.999999999초)	DT#1970-01-01-00:00:00.000000000
	STRING	문자 코드: UTF-8 0~1986바이트 (반각 문자로 0~1985문자+종단 NULL 문자)	''

파생 데이터 형식(열거 형식, 구조체, 공용체)

파생 데이터 형식(열거 형식, 구조체, 공용체)을 사용하는 변수에 대해서는, 해당 사실을 변수 데이터 형식표에 명기했습니다. 기술 방법에 대해 설명합니다.

열거 형식

열거 형식 변수는 표 안에 열거 데이터 형식을 기재했습니다.
다음은 그 예를 나타낸 것입니다. 이 경우, 변수 'Out'의 데이터 형식은 열거 형식 _eDAYOFWEEK입니다. 열거자에 대한 자세한 내용은 기능 설명에서 다룹니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																	○		○	
Out	열거 형식 _eDAYOFWEEK 열거자는 기능 설명 참조																			

구조체, 공용체

구조체 변수, 공용체 변수는 표 안에 구조체, 공용체의 데이터 형식을 기재했습니다.
다음은 그 예를 나타낸 것입니다. 이 경우, 변수 'In1'의 데이터 형식은 구조체 _sPORT입니다. 구조체, 공용체의 멤버에 대한 자세한 내용은 기능 설명에서 다룹니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	구조체 _sPORT에 대한 자세한 내용은 기능 설명 참조																			

구조체 전체, 구조체의 1개 멤버, 공용체 전체, 공용체의 1개 멤버를 파라메터로 지정할 수 있는 변수는 표 안에 해당 사실을 명기했습니다.
다음 예의 경우, 변수 'In1'에는 기본 데이터 형식 외에도 구조체 전체, 구조체의 1개 멤버, 공용체 전체, 공용체의 1개 멤버를 지정할 수 있습니다. 구조체 전체, 공용체 전체를 지정하는 경우에는 구조체명 또는 공용체명만을 파라메터로 사용하시기 바랍니다. 구조체의 1개 멤버, 공용체의 1개 멤버를 지정하는 경우에는 멤버명까지 명기한 것을 파라메터로 사용하시기 바랍니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	구조체 전체, 구조체의 1개 멤버, 공용체 전체, 공용체의 1개 멤버도 지정 가능																			

배열 지정

배열 지정 변수에는 변수명 뒤에 대괄호 [ ]를 붙여서 '배열'로 기재했습니다. 이 표시가 있는 경우, 첨자가 있는 배열의 요소를 파라메터로 사용하시기 바랍니다.
다음은 그 예를 나타낸 것입니다. 이 경우, In1[]은 BYTE 형식 배열임을 나타냅니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열		○																		

구조체 및 공용체를 요소로 하는 배열은 다음 예와 같이 표 안에 해당 사실을 명기했습니다. 이 표기가 있는 경우에도, 첨자가 있는 배열의 요소를 파라메터로 사용하시기 바랍니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열	구조체를 요소로 하는 배열, 공용체를 요소로 하는 배열을 지정																			

배열 전체, 배열의 1개 요소를 파라메터로 지정할 수 있는 변수는 표 안에 해당 사실을 명기했습니다.
다음 예의 경우, 변수 'In1'에는 기본 데이터 형식 외에도 배열 전체, 배열의 1개 요소를 지정할 수 있습니다. 배열 전체를 지정하는 경우에는 배열명만을 파라메터로 사용하시기 바랍니다. 배열의 1개 요소를 지정하는 경우, 첨자로 요소 번호까지 명기한 것을 파라메터로 사용하시기 바랍니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	배열 전체, 배열의 1개 요소도 지정 가능																			

기타

모든 명령 공통의 이상 감지

사용상의 주의 사항에는 각 명령에서 발생할 가능성이 있는 이상의 종류가 게재되어 있습니다. 하지만 다음과 같은 이상은 모든 명령 공통으로 감지되기 때문에 사용상의 주의 사항에는 게재하지 않았으므로 주의하시기 바랍니다.

- 배열 지정 변수의, 배열 영역을 넘어선 요소를 읽어온 경우
  (예) 배열 지정 변수a[0..3]에 대해 a[4]를 입력 변수로 설정
- 입력 변수, 출력 변수, 입출력 변수로서 배열 지정 변수가 정의되어 있는 명령에 대해, 배열이 아닌 파라메터를 보낸 경우
- STRING 형식 변수에, 정의된 바이트 수보다 긴 문자열을 대입한 경우
- STRING 형식 변수에 NULL 문자로 종단되지 않은 문자열을 대입한 경우
- 정수 형식 변수를 0으로 제산한 경우

모든 명령 공통의 주의 사항

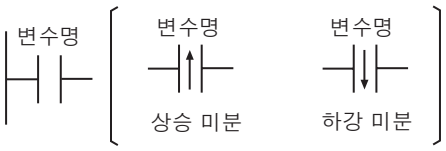
명령 중에는 연결하는 파라메터에 따라 처리량이 가변적인 것이 있습니다. 처리량이 너무 크면 명령 실행 시간이 길어져서 태스크 주기를 초과하게 되는 경우가 있습니다. 그 경우 태스크 주기 초과가 되므로 처리량을 적절히 조정하시기 바랍니다.

래더도 명령

명령	명칭	페이지
LD/LDN	로드/로드 부정	2-16
AND/ANDN	논리곱/부정 논리곱	2-18
OR/ORN	논리합/부정 논리합	2-21
Out/OutNot	출력/부정 출력	2-24

LD / LDN

LD : BOOL형식 변수의 값을 읽어옵니다.
LDN : BOOL형식 변수의 반전값을 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LD	로드	–		없음
LDN	로드 부정	–		없음

변수

없음

기능

● LD

변수명으로 지정한 BOOL 형식 변수의 값을 읽어서 다음 단계로 연결(출력)합니다.
지정한 변수의 값이 TRUE라면 TRUE를 출력, FALSE라면 FALSE를 출력합니다.
모선에서 시작되는 최초의 a 접점 또는 회로 블록의 최초의 a 접점에 사용합니다.

● LDN

변수명으로 지정한 BOOL 형식 변수의 반전값을 읽어서 다음 단계로 연결(출력)합니다.
지정한 변수의 값이 TRUE라면 FALSE를 출력, FALSE라면 TRUE를 출력합니다.
모선에서 시작되는 최초의 b 접점 또는 회로 블록의 최초의 b 접점에 사용합니다.

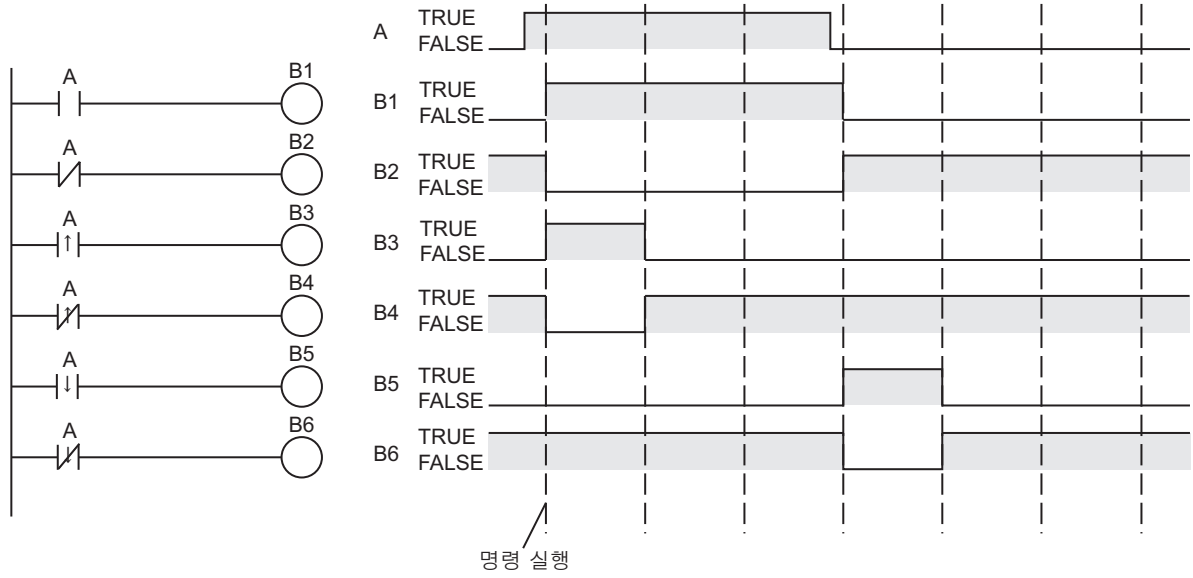
상승 미분, 하강 미분이 지정되지 않은 경우의 동작은 다음과 같습니다.

명령	변수의 값	출력
LD	TRUE	TRUE
	FALSE	FALSE
LDN	TRUE	FALSE
	FALSE	TRUE

상승 미분, 하강 미분이 지정되는 경우, 이전 실행 시의 변수값과 현재 변수값에 따라 다음과 같이 동작합니다.

명령	미분 지정	이전 실행 시의 변수값과 현재 변수값	출력
LD	상승 미분	이전 실행 시 FALSE → 현재 TRUE	TRUE
		상기 이외	FALSE
	하강 미분	이전 실행 시 TRUE → 현재 FALSE	TRUE
		상기 이외	FALSE
LDN	상승 미분	이전 실행 시 FALSE → 현재 TRUE	FALSE
		상기 이외	TRUE
	하강 미분	이전 실행 시 TRUE → 현재 FALSE	FALSE
		상기 이외	TRUE

기술 예 및 타이밍 차트가 아래에 나와 있습니다.

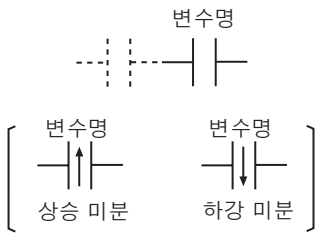
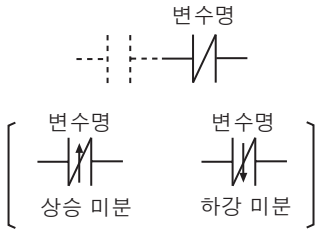


사용상의 주의 사항

- 다음과 같은 경우, 이상으로 간주되어 이전 실행 시의 출력값을 유지합니다.
  - 변수의 값에 배열 요소를 지정했으나 그 요소가 존재하지 않는 경우
    (예) BOOL 형식 배열 a[0..5]가 정의되어 있으며, 변수 a[10]을 지정해서 이 명령을 실행하려고 한 경우
- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.

AND / ANDN

AND : BOOL 형식 변수의 값과 입력 조건의 논리곱을 산출합니다.
ANDN : BOOL 형식 변수의 반전값과 입력 조건의 논리곱을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AND	논리곱	–		result:=vBool1 AND vBool2; result:=vBool1 & vBool2;
ANDN	부정 논리곱	–		result:=vBool1 AND NOT vBool2;

변수

없음

기능

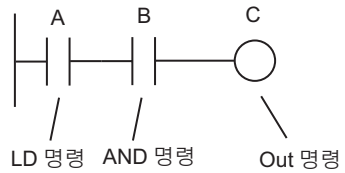
● AND

변수명으로 지정한 BOOL 형식 변수의 값과 입력 조건의 논리곱을 다음 단계로 연결(출력)합니다.
이전 단계 회로와 직렬 연결하는 a 접점에 사용합니다.

● ANDN

변수명으로 지정한 BOOL 형식 변수의 반전값과 입력 조건의 논리곱을 다음 단계로 연결(출력)합니다.
이전 단계 회로와 직렬 연결하는 b 접점에 사용합니다.

AND 명령의 기술 예는 다음과 같습니다. 변수 A와 변수 B의 논리곱을 변수 C에 출력합니다.





변수 A와 변수 B의 논리곱을 변수 C에 출력합니다.

상승 미분, 하강 미분이 지정되지 않은 경우의 동작은 다음과 같습니다.

명령	변수의 값과 입력 조건의 조합	출력
AND	변수의 값 : TRUE 입력 조건 : TRUE	TRUE
	상기 이외	FALSE
ANDN	변수의 값 : FALSE 입력 조건 : TRUE	TRUE
	상기 이외	FALSE

상승 미분, 하강 미분이 지정되지 않는 경우, 이전 실행 시의 변수값과 현재 변수값 및 입력 조건에 따라 다음과 같이 동작합니다.

명령	미분 지정	이전 실행 시의 변수값과 현재 변수값 및 입력 조건의 조합	출력
AND	상승 미분	변수의 값 : 이전 실행 시 FALSE → 현재 TRUE 입력 조건 : TRUE	TRUE
		상기 이외	FALSE
	하강 미분	변수의 값 : 이전 실행 시 TRUE → 현재 FALSE 입력 조건 : TRUE	TRUE
		상기 이외	FALSE
ANDN	상승 미분	변수의 값 : 이전 실행 시 FALSE → 현재 TRUE 입력 조건 : TRUE	FALSE
		변수의 값 : 관계 없음 입력 조건 : FALSE	
		상기 이외	TRUE
	하강 미분	변수의 값 : 이전 실행 시 TRUE → 현재 FALSE 입력 조건 : TRUE	FALSE
		변수의 값 : 관계 없음 입력 조건 : FALSE	
		상기 이외	TRUE

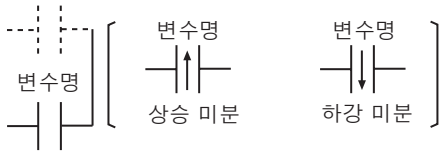
사용상의 주의 사항

- 다음과 같은 경우, 이상으로 간주되어 이전 실행 시의 출력값을 유지합니다.
  - 변수의 값에 배열 요소를 지정했으나 그 요소가 존재하지 않는 경우
    (예) BOOL 형식 배열 a[0..5]가 정의되어 있으며, 변수 a[10]을 지정해서 이 명령을 실행하려고 한 경우

- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.
- 이 명령은 모션에 직접 연결할 수 없습니다.

OR / ORN

OR : BOOL 형식 변수의 값과 입력 조건의 논리합을 산출합니다.
ORN : BOOL 형식 변수의 반전값과 입력 조건의 논리합을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
OR	논리합	–		result:=vBool1 OR vBool2;
ORN	부정 논리합	–		result:=vBool1 OR NOT vBool2;

변수

없음

기능 설명

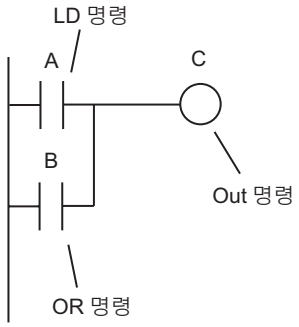
● OR

변수명으로 지정한 BOOL 형식 변수의 값과 입력 조건의 논리합을 다음 단계로 연결(출력)합니다.
이전 단계 회로와 병렬 연결하는 a 접점에 사용합니다. 모선에 연결되었거나 회로 블록 선두의 LD/LDN 명령에서 이 명령 직전까지의 회로와 OR(논리합)의 a 접점을 구성합니다.

● ORN

변수명으로 지정한 BOOL 형식 변수의 반전값과 입력 조건의 논리합을 다음 단계로 연결(출력)합니다.
이전 단계 회로와 병렬 연결하는 b 접점에 사용합니다. 모선에 연결되었거나 회로 블록 선두의 LD/LDN 명령에서 이 명령 직전까지의 회로와 OR(논리합)의 b 접점을 구성합니다.

OR 명령의 기술 예는 다음과 같습니다. 변수 A와 변수 B의 논리합을 변수 C에 출력합니다.





변수 A와 변수 B의 논리합을 변수 C에 출력합니다.

상승 미분, 하강 미분이 지정되지 않은 경우의 동작은 다음과 같습니다.

명령	변수의 값과 입력 조건의 조합	출력
OR	변수의 값 : FALSE 입력 조건 : FALSE	FALSE
	상기 이외	TRUE
ORN	변수의 값 : TRUE 입력 조건 : FALSE	FALSE
	상기 이외	TRUE

상승 미분, 하강 미분이 지정되지 않는 경우, 이전 실행 시의 변수값과 현재 변수값 및 입력 조건에 따라 다음과 같이 동작합니다.

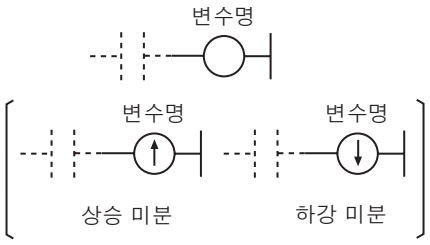
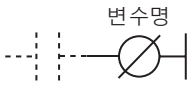
명령	미분 지정	이전 실행 시의 변수값과 현재 변수값 및 입력 조건의 조합	출력
OR	상승 미분	변수의 값 : 이전 실행 시 FALSE → 현재 TRUE 입력 조건 : 관계 없음	TRUE
		변수의 값 : 관계 없음 입력 조건 : TRUE	
		상기 이외	FALSE
	하강 미분	변수의 값 : 이전 실행 시 TRUE → 현재 FALSE 입력 조건 : 관계 없음	TRUE
		변수의 값 : 관계 없음 입력 조건 : TRUE	
		상기 이외	FALSE
ORN	상승 미분	변수의 값 : 이전 실행 시 FALSE → 현재 TRUE 입력 조건 : FALSE	FALSE
		상기 이외	TRUE
	하강 미분	변수의 값 : 이전 실행 시 TRUE → 현재 FALSE 입력 조건 : FALSE	FALSE
		상기 이외	TRUE

사용상의 주의 사항

- 다음과 같은 경우, 이상으로 간주되어 이전의 본 명령 실행 시의 출력값을 유지합니다.
  - 변수의 값에 배열 요소를 지정했으나 그 요소가 존재하지 않는 경우
    (예) BOOL 형식 배열 a[0..5]가 정의되어 있으며, 변수 a[10]을 지정해서 이 명령을 실행하려고 한 경우
- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.

Out / OutNot

Out : 이전 단계까지의 논리 연산 처리 결과를 BOOL 형식 변수로 출력합니다.
OutNot : 이전 단계까지의 논리 연산 처리 결과의 반전값을 BOOL 형식 변수로 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Out	출력	–		변수명: =(이전 단계까지의 논리 연산식);
OutNot	부정 출력	–		변수명: =NOT(이전 단계까지의 논리 연산식);

변수

없음

기능

● Out

이전 단계까지의 논리 연산 처리 결과를 변수명으로 지정한 BOOL 형식 변수로 출력합니다.

상승 미분, 하강 미분이 지정되지 않은 경우의 동작은 다음과 같습니다.

이전 단계까지의 논리 연산 처리 결과	출력
TRUE	TRUE
FALSE	FALSE

Out 명령에는 상승 미분, 하강 미분의 동작 지정을 할 수 있습니다. 그 경우, 이전 단계까지의 논리 연산 처리 결과가 이전 실행 시와 현재 어떻게 변화했는지에 따라 출력값이 결정됩니다. 현재를 기준으로 이전 단계까지의 논리 연산 처리 결과에 따라, 다음과 같이 동작합니다.

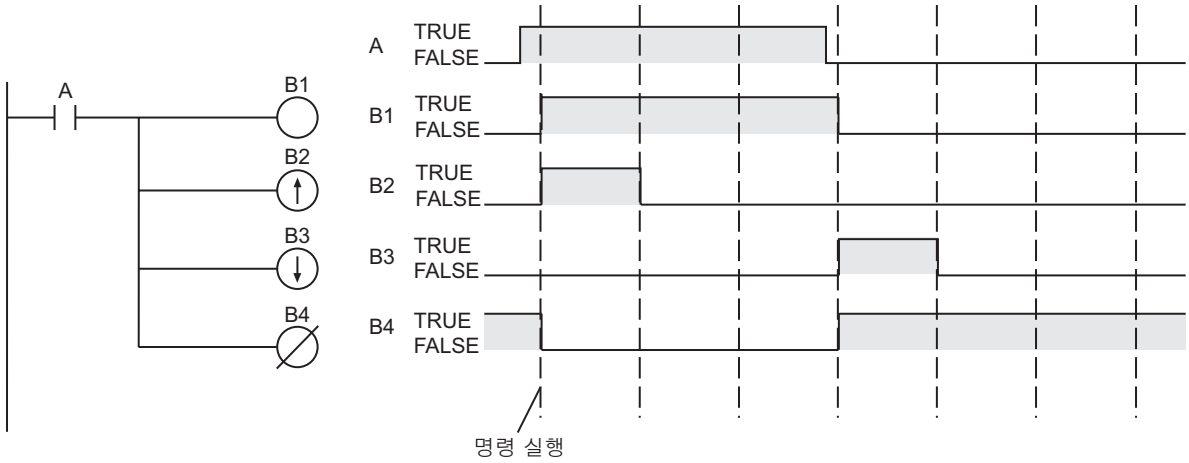
미분 지정	이전 실행 시 이전 단계까지의 논리 연산 처리 결과와, 현재를 기준으로 이전 단계까지의 논리 연산 처리 결과	출력
상승 미분	이전 실행 시 FALSE → 현재 TRUE	TRUE
	상기 이외	FALSE
하강 미분	이전 실행 시 TRUE → 현재 FALSE	TRUE
	상기 이외	FALSE

● OutNot

이전 단계까지의 논리 연산 처리 결과의 반전값을 변수명으로 지정한 BOOL 형식 변수로 출력합니다.

이전 단계까지의 논리 연산 처리 결과	출력
TRUE	FALSE
FALSE	TRUE

기술 예 및 타이밍 차트가 아래에 나와 있습니다.



참고

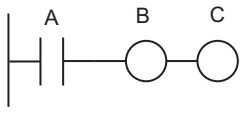
Set/Reset 명령과 Out/OutNot 명령의 차이

- Set/Reset 명령은 입력이 True가 되었을 때 동작하며, 입력이 FALSE일 때는 동작하지 않습니다. 즉, 입력이 FALSE일 때는 출력에 변화가 없습니다.
- Out/OutNot 명령은 이전 단계까지의 논리 연산 처리 결과가 TRUE이건 FALSE이건, 그에 따른 출력을 수행합니다.

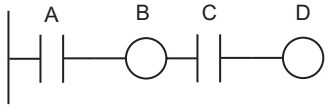
사용상의 주의 사항

- 다음과 같은 경우, 이상으로 간주되어 출력을 수행하지 않습니다.
  - 변수의 값에 배열 요소를 지정했으나 그 요소가 존재하지 않는 경우

  (예) BOOL 형식 배열 a[0..5]가 정의되어 있으며, 변수 a[10]을 지정해서 이 명령을 실행하려고 한 경우
- 다음과 같은 연결은 가능합니다.
  - Out 명령의 다음 단계에 다른 Out 명령을 연결

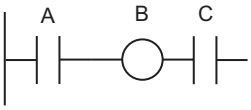


  - Out 명령의 다음 단계에 LD 명령과 Out 명령을 연결

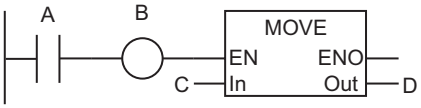


- 다음과 같은 연결은 할 수 없습니다.

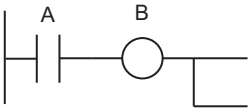
- Out 명령의 다음 단계에 LD 명령만을 연결

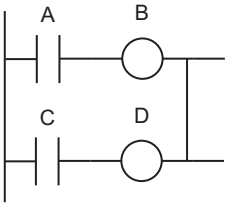


- Out 명령의 다음 단계에 펑션, 펑션 블록을 연결



- Out 명령의 다음 단계에서의 분기 및 합류





ST 구문 명령

명령	명칭	페이지
IF	선택	2-28
CASE	다방향 분기	2-32
WHILE	반복	2-36
REPEAT	반복	2-38
RETURN	복귀	2-65
FOR	반복	2-83
EXIT	루프 중단	2-90

IF

지정한 조건식의 평가 결과에 따라 실행될 구문을 이지선다 방식으로 선택합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
IF	선택	–	없음	IF 조건식 THEN 구문; ELSIF 조건식 THEN 구문; ELSE 구문; END_IF;

변수

없음

기능

지정한 조건식의 평가 결과에 따라 실행될 구문을 이지선다 방식으로 선택합니다. 조건식에는 TRUE 또는 FALSE의 값을 평가 결과로 갖는 다음과 같은 것을 기술합니다.

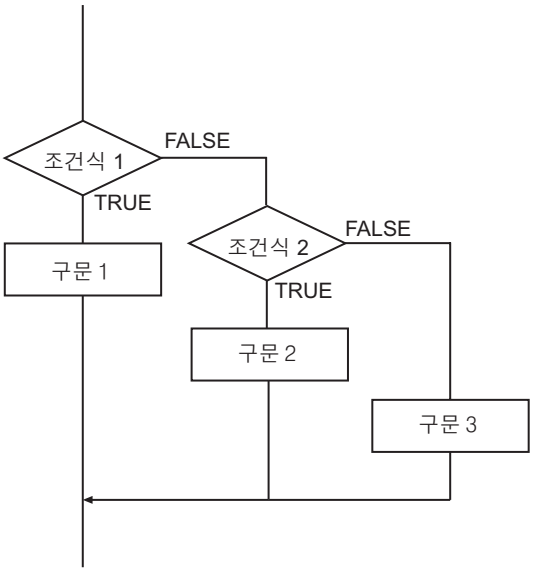
조건식으로 기술할 수 있는 것	기술 예	평가 결과
논리식	a>3	변수 a의 값이 3보다 크면 TRUE, 그 이외에는 FALSE
	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
BOOL 형식 변수	abc	abc의 값이 TRUE라면 TRUE, FALSE라면 FALSE
BOOL 형식 정수	TRUE	TRUE
BOOL 형식의 반환값을 갖는 펑션	FUN명	해당 펑션의 반환값이 TRUE라면 TRUE, FALSE라면 FALSE

논리식에는 다음과 같은 연산자를 사용할 수 있습니다.

연산자	의미	기술 예	평가 결과
=	일치	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
<>	불일치	a<>b	변수 a, b의 값이 동일하지 않으면 TRUE, 그 이외에는 FALSE
<	비교	a<b	변수 a의 값이 b보다 작으면 TRUE, 그 이외에는 FALSE
< =		a<=b	변수 a의 값이 b 이하라면 TRUE, 그 이외에는 FALSE
>		a>b	변수 a의 값이 b보다 크면 TRUE, 그 이외에는 FALSE
>=		a>=b	변수 a의 값이 b 이상이라면 TRUE, 그 이외에는 FALSE
AND, &	논리곱	a AND b a & b	BOOL 형식 변수 a, b의 논리곱
OR	논리합	a OR b	BOOL 형식 변수 a, b의 논리합
XOR	배타적 논리합	a XOR b	BOOL 형식 변수 a, b의 배타적 논리합
NOT	부정	NOT a	BOOL 형식 변수 a의 부정

다음 기술은 조건식 1, 조건식 2의 평가 결과에 따라 그림과 같은 처리 플로우를 갖게 됩니다. 구문 n에는 복수의 구문을 기술할 수 있습니다.

```
IF 조건식 1 THEN
  구문 1;
ELSIF 조건식 2 THEN
  구문 2;
ELSE
  구문 3;
END_IF;
```



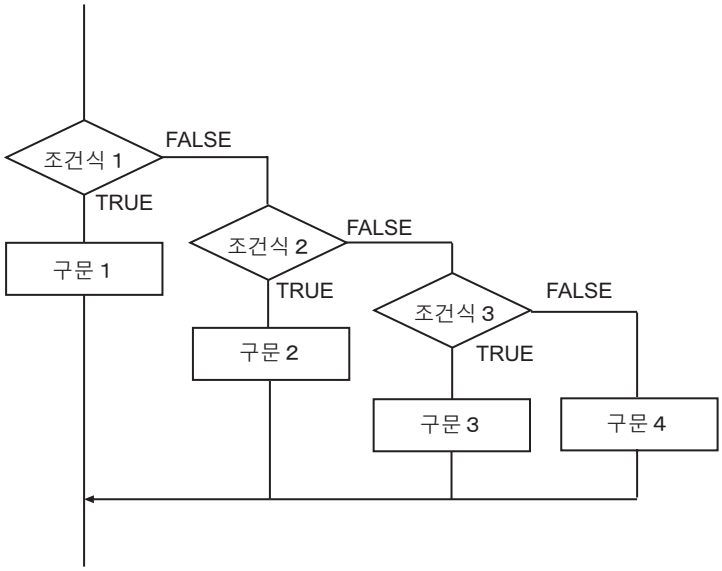
참고

- 이 명령은 계층 구조로 만들 수 있습니다. 다음과 같은 기술의 경우, 구문 11은 조건식 1의 평가 결과가 TRUE이며 조건식 11의 평가 결과도 TRUE일 때 실행됩니다.

```
IF 조건식 1 THEN
    IF 조건식 11 THEN
      구문 11;
    ELSIF 조건식 12 THEN
      구문 12;
    ELSE
      구문 13;
    END_IF;
ELSIF 조건식 2 THEN
    구문 2;
ELSE
    구문 3;
END_IF;
```

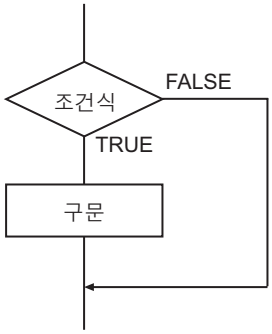
ELSIF는 복수 기술할 수 있습니다. 다음과 같은 기술의 경우, 처리 플로우는 그림과 같습니다.

```
IF 조건식 1 THEN
  구문 1;
ELSIF  조건식 2 THEN
  구문 2;
ELSIF  조건식 3 THEN
  구문 3;
ELSE
  구문 4;
END_IF;
```



- 조건식이 1개뿐인 경우, ELSIF는 필요 없습니다. 또한, 어느 조건식에도 맞지 않으면 아무 처리도 하지 않는 경우에도 ELSE가 필요 없습니다. 다음과 같은 기술의 경우, 처리 플로우는 그림과 같습니다.

```
IF 조건식 THEN
  구문;
END_IF;
```



- 구문에 쓸 수 있는 내용에 특별한 제한이 없습니다. 펑션 블록 호출 및 FOR 명령 등, IF 명령 이외에는 무엇이든 쓸 수 있습니다.

사용상의 주의 사항

- IF와 END_IF는 생략할 수 없습니다. 또한, 이 둘은 반드시 짝을 이루도록 함께 사용해야 합니다.
- IF 명령, CASE 명령, FOR 명령, WHILE 명령, REPEAT 명령을 포함해서, 계층의 깊이는 15단계까지입니다.

샘플 프로그램

변수 abc의 값이 INT#0보다 작으면 변수 def에 INT#0을 대입합니다. abc의 값이 INT#0이라면 변수 def에 INT#1을, 변수 ghi에 INT#2를 대입합니다. abc의 값이 어느 쪽에도 해당하지 않으면, 변수 def에 INT#3을 대입합니다.

명칭	데이터 형식	초기값
abc	INT	0
def	INT	0
ghi	INT	0

```
IF (abc<INT#0) THEN
   def:=INT#0;
ELSIF (abc=INT#0) THEN
   def:=INT#1;
   ghi:=INT#2;
ELSE
   def:=INT#3;
END_IF;
```

CASE

지정한 정수식의 값에 따라 복수의 구문 중에서 실행될 구문을 선택합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CASE	다방향 분기	–	없음	CASE 정수식 OF 선택값: 구문; 선택값 : 구문; : ELSE 구문; END_CASE;

변수

없음

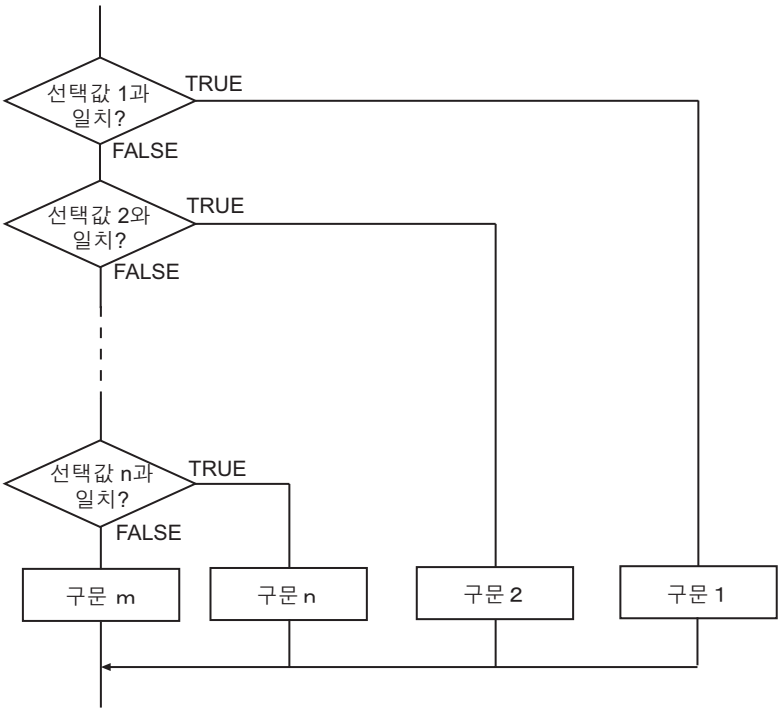
기능

지정한 정수식의 값에 따라 복수의 구문 중에서 실행될 구문을 선택합니다.
정수식과 선택값에는 다음과 같은 것을 기술합니다.

	기술할 수 있는 것
정수식	정수 형식 변수, 정수 형식 정수, 정수 형식의 식, 정수 형식의 반환값을 갖는 펑션, 열거 형식 변수, 열거 형식의 식, 열거 형식의 반환값을 갖는 펑션
선택값	정수 형식 정수

다음 기술은 정수식의 값에 따라 그림과 같은 처리 플로우를 갖게 됩니다. 구문이라고 적힌 곳에는 복수의 구문을 기술할 수 있습니다.

```
CASE 정수식 OF
    1 :
      구문 1;
    2 :
      구문 2;
    :
    n :
      구문 n;
    ELSE
      구문 m;
END_CASE;
```



참고

- 이 명령은 계층 구조로 만들 수 있습니다. 다음과 같은 기술의 경우, 구문 12는 조건식 1의 값이 1이고 정수식 11의 값이 2일 때 실행됩니다.

```
CASE 정수식 1 OF
    1 :
        CASE 정수식 11 OF
            1 :
                구문 11;
            2 :
                구문 12;
            ELSE
                구문 1m;
        END_CASE;
    2 :
        구문 2;
    3 :
        구문 3;
     ELSE
        구문 m;
END_CASE;
```

- 선택값에는 복수의 값을 기술할 수 있습니다. 복수의 값은 ','(쉼표)로 구분해서 기술합니다. 다음과 같은 기술의 경우, 정수식의 값이 1 또는 2일 때 구문 1이 실행됩니다.

```
CASE 정수식 1 OF
    1,2 :
      구문 1;
    3 :
      구문 2;
    4 :
      구문 3;
     ELSE
      구문 m;
END_CASE;
```

- 선택값에 연속값을 기술할 수도 있습니다. 연속값은 '..'(마침표 2개)로 구분해서 기술합니다. 다음과 같은 기술의 경우, 정수식의 값이 10 이상 15 이하일 때 구문 1이 실행됩니다.

```
CASE 정수식 1 OF
    10..15 :
          구문 1;
    16 :
          구문 2;
    17 :
          구문 3;
     ELSE
          구문 m;
END_CASE;
```

- ELSE는 생략할 수 있습니다. 그 경우, 정수식의 값이 어떤 선택값과도 동일하지 않다면 어느 구문도 실행되지 않습니다.
- 구문에 쓸 수 있는 내용에 특별한 제한이 없습니다. 펑션 블록 호출 및 FOR 명령 등, CASE 명령 이외에는 무엇이든 쓸 수 있습니다.
- C언어의 switch 구문과는 다음과 같은 사양이 다릅니다. C언어의 switch 구문은, break 구문을 사용하지 않는 이상 정수식과 같은 선택값 이후의 구문이 모두 실행됩니다. 이와는 달리 CASE 구문은 정수식과 같은 선택값에 해당하는 구문만이 실행됩니다. 예를 들면 다음과 같은 예의 경우, C언어의 switch 구문은 구문 1에서 구문 3까지가 모두 실행됩니다. 반면 CASE 명령에서는 구문 1만 실행됩니다.

C언어의 switch 구문

```
val=1;
switch val
{
    case 1:
        구문1;
    case 2:
        구문2;
    case 3:
        구문3;
}
```

CASE 명령

```
val:=1;
CASE val OF
    1:
        구문1;
    2:
        구문2;
    3:
        구문3;
END_CASE;
```

사용상의 주의 사항

- CASE와 END_CASE는 생략할 수 없습니다. 또한, 이 둘은 반드시 짝을 이루도록 함께 사용해야 합니다.
- 정수식과 선택값은 데이터 형식이 다르더라도 상관없습니다.
- 선택값에는 같은 값이 중복되도록 기술할 수 없습니다.
- IF 명령, CASE 명령, FOR 명령, WHILE 명령, REPEAT 명령을 포함해서, 계층의 깊이는 15단계까지입니다.

샘플 프로그램

변수 abc의 값이 INT#1이라면 변수 def에 INT#10을 대입, INT#2라면 INT#20을 대입, INT#3이라면 INT#30을 대입합니다. 어느 것에도 해당하지 않는다면 변수 ghi의 값을 대입합니다.

명칭	데이터 형식	초기값
abc	INT	0
def	INT	0
ghi	INT	0

```
CASE abc OF
   INT#1:
      def:=INT#10;
   INT#2:
      def:=INT#20;
   INT#3:
      def:=INT#30;
   ELSE
      def:=ghi;
END_CASE;
```

변수 abc의 값이 INT#1이라면 변수 def에 INT#10을, INT#2 또는 INT#5라면 INT#20을, INT#6 이상 INT#10 이하라면 INT#30을 대입합니다. 어느 것에도 해당하지 않는 경우, 아무 처리도 하지 않습니다.

명칭	데이터 형식	초기값
abc	INT	0
def	INT	0

```
CASE abc OF
   INT#1:
      def:=INT#10;
   INT#2,INT#5:
      def:=INT#20;
   INT#6..INT#10:
      def:=INT#30;
END_CASE;
```

WHILE

지정한 조건식의 평가 결과가 TRUE인 동안, 특정 구문을 반복 실행합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
WHILE	반복	–	없음	WHILE 조건식 DO 구문; END_WHILE;

변수

없음

기능

지정한 조건식의 평가 결과가 TRUE인 동안, 특정 구문을 반복 실행합니다. 조건식에는 TRUE 또는 FALSE의 값을 평가 결과로 갖는 다음과 같은 것을 기술합니다.

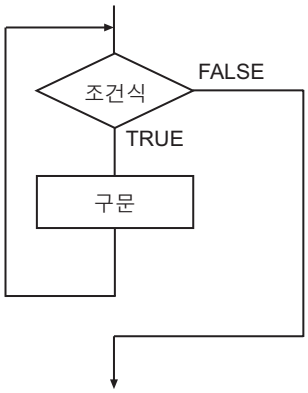
조건식으로 기술할 수 있는 것	기술 예	평가 결과
논리식	a>3	변수 a의 값이 3보다 크면 TRUE, 그 이외에는 FALSE
	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
BOOL 형식 변수	abc	abc의 값이 TRUE라면 TRUE, FALSE라면 FALSE
BOOL 형식 정수	TRUE	TRUE
BOOL 형식의 반환값을 갖는 펑션	FUN명	해당 펑션의 반환값이 TRUE라면 TRUE, FALSE라면 FALSE

논리식에는 다음과 같은 연산자를 사용할 수 있습니다.

연산자	의미	기술 예	평가 결과
=	일치	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
<>	불일치	a<>b	변수 a, b의 값이 동일하지 않으면 TRUE, 그 이외에는 FALSE
<	비교	a<b	변수 a의 값이 b보다 작으면 TRUE, 그 이외에는 FALSE
< =		a<=b	변수 a의 값이 b 이하라면 TRUE, 그 이외에는 FALSE
>		a>b	변수 a의 값이 b보다 크면 TRUE, 그 이외에는 FALSE
>=		a>=b	변수 a의 값이 b 이상이라면 TRUE, 그 이외에는 FALSE
AND, &	논리곱	a AND b a & b	BOOL 형식 변수 a, b의 논리곱
OR	논리합	a OR b	BOOL 형식 변수 a, b의 논리합
XOR	배타적 논리합	a XOR b	BOOL 형식 변수 a, b의 배타적 논리합
NOT	부정	NOT a	BOOL 형식 변수 a의 부정

다음과 같은 기술의 경우, 처리 플로우는 그림과 같습니다. 그림에서 구문이라고 적힌 곳에는 복수의 구문을 기술할 수 있습니다.

```
WHILE 조건식 DO
    구문;
END_WHILE;
```



참고

- 조건식을 최초로 평가한 결과가 FALSE인 경우, 구문은 한 번도 실행되지 않습니다.
- 구문에 쓸 수 있는 내용에 특별한 제한이 없습니다. 펑션 블록 호출 및 FOR 명령 등, WHILE 명령 이외에는 무엇이든 쓸 수 있습니다.

사용상의 주의 사항

- WHILE과 END_WHILE은 생략할 수 없습니다. 또한, 이 둘은 반드시 짝을 이루도록 함께 사용해야 합니다.
- IF 명령, CASE 명령, FOR 명령, WHILE 명령, REPEAT 명령을 포함해서, 계층의 깊이는 15단계까지입니다.

샘플 프로그램

변수 abc의 값이 INT#1000 이하일 동안, abc에 INT#7을 계속 더합니다.

명칭	데이터 형식	초기값
abc	INT	0

```
abc:=INT#0;
WHILE abc<=INT#1000 DO
   abc:=abc+INT#7;
END_WHILE;
```

REPEAT

특정 구문을 실행하여, 지정한 조건식이 TRUE가 될 때까지 그 구문을 반복 실행합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
REPEAT	반복	–	없음	REPEAT 구문; UNTIL 조건식 END_REPEAT;

변수

없음

기능

특정 구문을 실행하여, 지정한 조건식이 TRUE가 될 때까지 그 구문을 반복 실행합니다. 조건식에는 TRUE 또는 FALSE의 값을 평가 결과로 갖는 다음과 같은 것을 기술합니다.

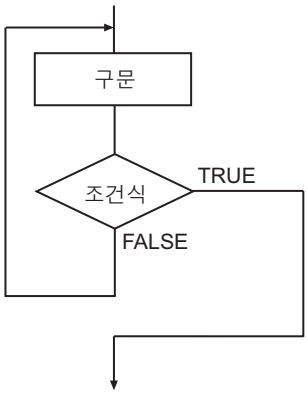
조건식으로 기술할 수 있는 것	기술 예	평가 결과
논리식	a>3	변수 a의 값이 3보다 크면 TRUE, 그 이외에는 FALSE
	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
BOOL 형식 변수	abc	abc의 값이 TRUE라면 TRUE, FALSE라면 FALSE
BOOL 형식 정수	TRUE	TRUE
BOOL 형식의 반환값을 갖는 펑션	FUN명	해당 펑션의 반환값이 TRUE라면 TRUE, FALSE라면 FALSE

논리식에는 다음과 같은 연산자를 사용할 수 있습니다.

연산자	의미	기술 예	평가 결과
=	일치	a=b	변수 a, b의 값이 동일하면 TRUE, 그 이외에는 FALSE
<>	불일치	a<>b	변수 a, b의 값이 동일하지 않으면 TRUE, 그 이외에는 FALSE
<	비교	a<b	변수 a의 값이 b보다 작으면 TRUE, 그 이외에는 FALSE
< =		a<=b	변수 a의 값이 b 이하라면 TRUE, 그 이외에는 FALSE
>		a>b	변수 a의 값이 b보다 크면 TRUE, 그 이외에는 FALSE
>=		a>=b	변수 a의 값이 b 이상이라면 TRUE, 그 이외에는 FALSE
AND, &	논리곱	a AND b a & b	BOOL 형식 변수 a, b의 논리곱
OR	논리합	a OR b	BOOL 형식 변수 a, b의 논리합
XOR	배타적 논리합	a XOR b	BOOL 형식 변수 a, b의 배타적 논리합
NOT	부정	NOT a	BOOL 형식 변수 a의 부정

다음과 같은 기술의 경우, 처리 플로우는 그림과 같습니다. 그림에서 구문이라고 적힌 곳에는 복수의 구문을 기술할 수 있습니다.

```
REPEAT
    구문;
UNTIL 조건식
END_REPEAT;
```



참고

- 구문을 한 번 실행한 후 조건식을 평가합니다. 따라서, 구문은 반드시 한 번은 실행됩니다.
- 구문에 쓸 수 있는 내용에 특별한 제한이 없습니다. 펑션 블록 호출 및 FOR 명령 등, REPEAT 명령 이외에는 무엇이든 쓸 수 있습니다.

사용상의 주의 사항

- REPEAT, UNTIL, END_REPEAT는 모두 생략할 수 없습니다. 또한, 이 명령들은 반드시 짝을 이루도록 함께 사용해야 합니다.
- IF 명령, CASE 명령, FOR 명령, WHILE 명령, REPEAT 명령을 포함해서, 계층의 깊이는 15단계까지입니다.

샘플 프로그램

변수 abc의 값이 INT#10보다 커질 때까지, abc에 INT#1을 계속 더합니다.

명칭	데이터 형식	초기값
abc	INT	0

```
abc:=INT#0;
REPEAT
   abc:=abc+INT#1;
UNTIL abc>INT#10
END_REPEAT;
```

RETURN

RETURN 명령의 설명은 시퀀스 제어 명령의 'RETURN 명령(2-65페이지)'을 참조하시기 바랍니다.

FOR

FOR 명령의 설명은 시퀀스 제어 명령의 'FOR / NEXT 명령(2-83페이지)'을 참조하시기 바랍니다.

EXIT

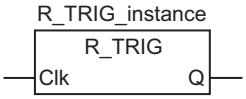
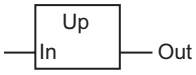
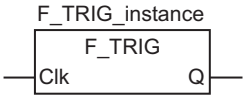
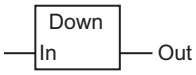
EXIT 명령의 설명은 시퀀스 제어 명령의 'BREAK 명령(2-90페이지)'을 참조하시기 바랍니다.
래더도의 BREAK 명령과 ST의 EXIT 명령은 같은 의미입니다.

시퀀스 입력 명령

명령	명칭	페이지
R_TRIG, Up/ F_TRIG, Down	상승 미분/ 하강 미분	2-44
TestABit/TestABitN	비트 테스트/ 비트 테스트 부정	2-47

R_TRIG, Up / F_TRIG, Down

R_TRIG, Up : 입력 신호의 상승(FALSE→TRUE) 시, 1태스크 주기 동안만 TRUE를 출력합니다.
F_TRIG, Down : 입력 신호의 하강(TRUE→FALSE) 시, 1태스크 주기 동안만 TRUE를 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
R_TRIG	상승 미분	FB		R_TRIG_instance(Clk, Q);
Up		FUN		없음
F_TRIG	하강 미분	FB		F_TRIG_instance(Clk, Q);
Down		FUN		없음

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Clk, In	입력 신호	입력	입력 신호	데이터 형식에 따름	–	–
Q, Out	출력 신호	출력	출력 신호	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Clk, In	○																			
Q, Out	○																			

기능

● R_TRIG

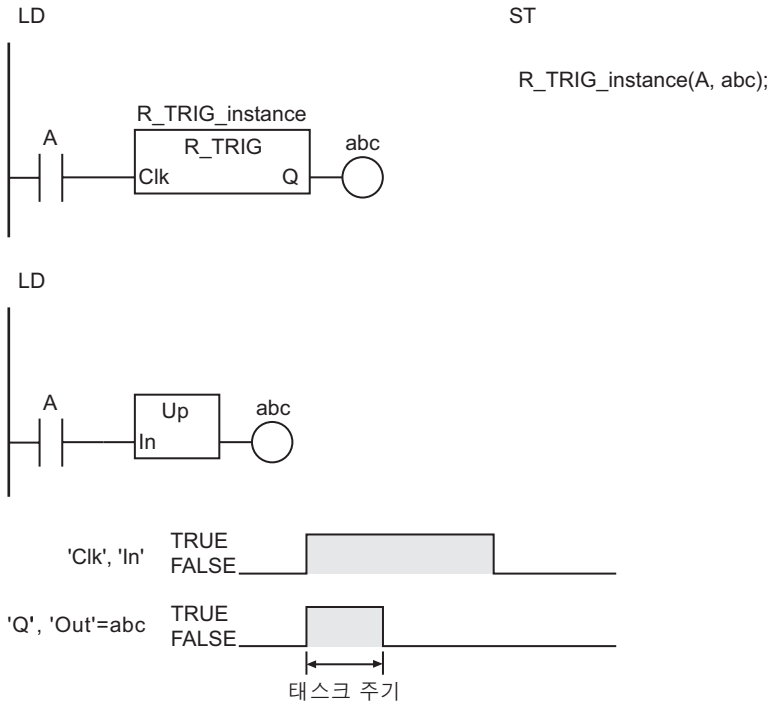
입력 신호 'Clk'의 상승(FALSE→TRUE) 시, 1태스크 주기 동안만 출력 신호 'Q'에 TRUE를 대입합니다. 그 이외에는 'Q'의 값이 FALSE가 됩니다.

● Up

R_TRIG 명령과 Up 명령의 기능은 완전히 동일합니다. F_TRIG 명령의 기능 설명문에서 변수명 'Clk'를 'In'으로, 'Q'를 'Out'로 각각 바꿔 읽으십시오.

단, 이 명령을 최초로 실행하는 태스크 주기에서의 동작은 R_TRIG 명령의 동작과 다릅니다. 이 명령을 최초로 실행하는 태스크 주기에서의 동작은 🕮 사용상의 주의 사항을 참조하시기 바랍니다.

기술 예 및 타이밍 차트가 아래에 나와 있습니다.



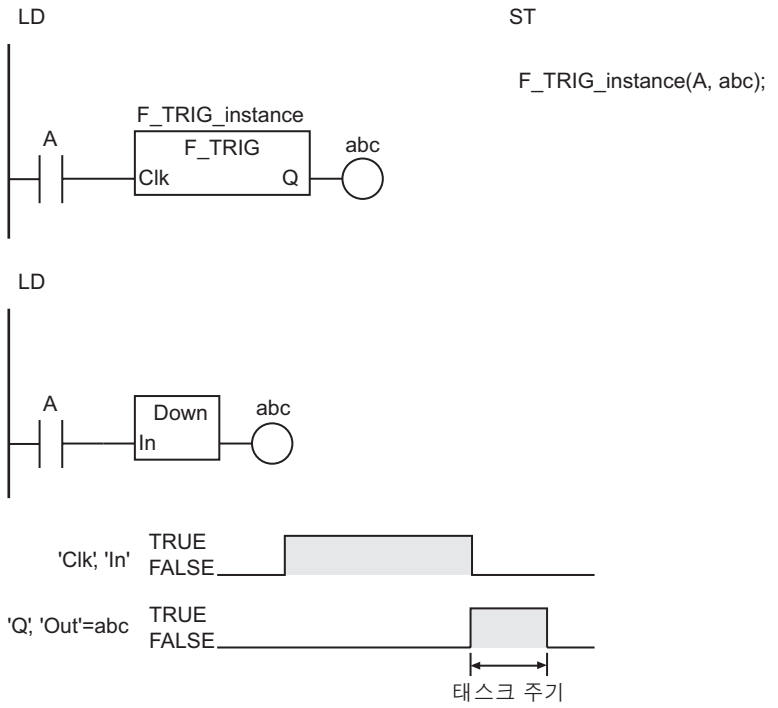
● F_TRIG

입력 신호 'Clk'의 상승(FALSE→TRUE) 시, 1태스크 주기 동안만 출력 신호 'Q'에 TRUE를 대입합니다. 그 이외에는 'Q'의 값이 FALSE가 됩니다.

● Down

F_TRIG 명령과 Down 명령의 기능은 완전히 동일합니다. F_TRIG 명령의 기능 설명문에서 변수명 'Clk'를 'In'으로, 'Q'를 'Out'로 각각 바꿔 읽으십시오.

기술 예 및 타이밍 차트가 아래에 나와 있습니다.



사용상의 주의 사항

- 이전의 이 명령 실행 시와 현재의 'Clk' 또는 'In'의 값의 차이로 상승/하강을 감지합니다. 그러므로 JMP 명령 등을 사용해서 이 명령을 매회 태스크 주기마다 실행하지 않는 경우 주의하시기 바랍니다.
- 전원 OFF가 발생하더라도, 'Clk' 또는 'In'의 값이 FALSE가 된 것으로 인식하지 않습니다. 'Clk', 'In'의 값이 FALSE가 된 것으로 인식하는 것은, 'Clk', 'In'의 값이 FALSE인 상태에서 이 명령이 'Clk', 'In'의 값을 평가했을 때뿐입니다.
- Up 명령을 최초로 실행한 태스크 주기에는 'In'의 값에 관계 없이 'Out'의 값이 반드시 FALSE가 됩니다.
- 전원 ON 시, Up 명령의 'In'의 값이 TRUE인 경우는 'In'의 값이 일단 FALSE가 된 후 TRUE가 될 때까지 'Out'의 값은 FALSE가 됩니다.
- F_TRIG 명령을 최초로 실행한 태스크 주기에는 'Clk'의 값에 관계 없이 'Q'의 값이 반드시 FALSE가 됩니다.
- 전원 ON 시, F_TRIG 명령의 'Clk'의 값이 FALSE인 경우는 'Clk'의 값이 일단 TRUE가 된 후 FALSE가 될 때까지 'Q'의 값은 FALSE가 됩니다.
- Down 명령을 최초로 실행한 태스크 주기에는 'In'의 값에 관계 없이 'Out'의 값이 반드시 FALSE가 됩니다.
- 전원 ON 시, Down 명령의 'In'의 값이 FALSE인 경우는 'In'의 값이 일단 TRUE가 된 후 FALSE가 될 때까지 'Out'의 값은 FALSE가 됩니다.

버전 관련 정보

CPU 유니트의 유니트 버전에 따라 'Clk'의 값이 TRUE일 때 다음 타이밍에 R_TRIG 명령이 실행된 경우의 'Q'의 값이 달라집니다.

'Clk'의 값이 TRUE일 때 이 명령이 실행되는 타이밍	'Q'의 값	
	CPU 유니트 Ver.1.02 이상	CPU 유니트 Ver.1.01 이하
R_TRIG 명령을 최초로 실행하는 태스크 주기	TRUE가 됩니다.	반드시 FALSE가 됩니다.
전원 ON 시	TRUE가 됩니다.	'Clk'의 값이 일단 FALSE가 된 후 TRUE가 될 때까지, 'Q'의 값은 FALSE가 됩니다.

TestABit / TestABitN

TestABit : 비트 열의 지정 비트값을 출력합니다.
TestABitN : 비트 열의 지정 비트의 반전값을 출력합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TestABit	비트 테스트	FUN	(@)TestABit EN In Pos Out	Out:=TestABit (In, Pos);
TestABitN	비트 테스트 부정	FUN	(@)TestABitN EN In Pos Out	Out:=TestABitN (In, Pos);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	비트 열	입력	비트 열	데이터 형식에 따름	–	(*)
Pos	비트 위치		지정 비트 위치	0~'In'의 비트 수 -1		0
Out	비트값	출력	• TestABit 지정 비트값 • TestABitN 지정 비트의 반전값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Pos						○														
Out	○																			

기능

● TestABit

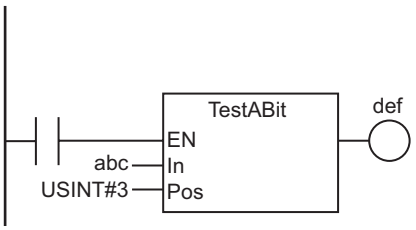
EN이 TRUE일 때, 비트 열 'In'의 비트 위치 'Pos'의 값을 비트값 'Out'에 대입합니다. EN이 FALSE일 때, 'Out'의 값은 FALSE가 됩니다.

● TestABitN

EN이 TRUE일 때, 비트 열 'In'의 비트 위치 'Pos'의 반전값을 비트값 'Out'에 대입합니다. EN이 FALSE일 때, 'Out'의 값은 FALSE가 됩니다.

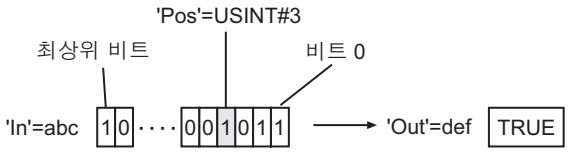
TestABit 명령에서, 'Pos'=USINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=TestABit(abc, USINT#3);
```



사용상의 주의 사항

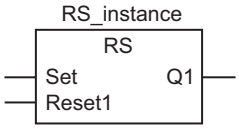
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 다음과 같은 경우 이상 상태가 되어 'Out'의 값이 FALSE가 됩니다.
  - 'Pos'의 값이 'In'의 비트 수-1보다 큰 경우

시퀀스 출력 명령

명령	명칭	페이지
RS	리셋 우선 유지	2-50
SR	세트 우선 유지	2-52
Set/Reset	세트/리셋	2-54
SetBits/ResetBits	다중 비트 세트/ 다중 비트 리셋	2-57
SetABit/ResetABit	1비트 세트/ 1비트 리셋	2-59
OutABit	1비트 출력	2-61

RS

BOOL 형식 변수의 값을 유지합니다.세트 입력과 리셋 입력이 동시에 TRUE인 경우, 리셋 입력이 우선합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
RS	리셋 우선 유지	FB		RS_instance(Set, Reset1, Q1);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Set*1	세트	입력	세트 입력	데이터 형식에 따름	–	0
Reset1*1	리셋		리셋 입력			
Q1	유지	출력	유지 출력	데이터 형식에 따름	–	–

*1 Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Set를 S, Reset1을 R1로 기술할 수 있습니다. 예를 들어, RS_instance(S:=A, R1:=B, Q1=>abc);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Set	○																			
Reset1	○																			
Q1	○																			

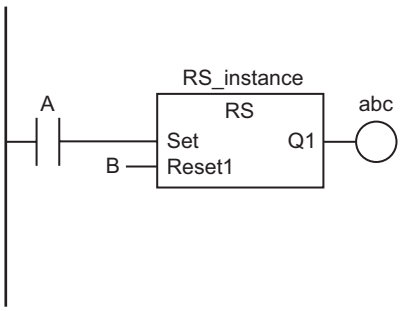
기능

리셋 우선의 자체 유지 회로입니다. 입출력의 관계가 아래에 나와 있습니다.

'Set'의 값	'Reset1'의 값	'Q1'의 값
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	유지

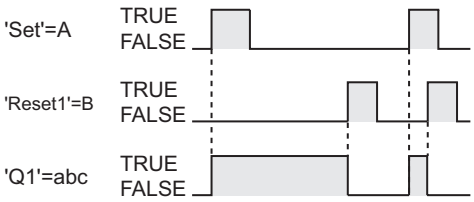
기술 예 및 타이밍 차트가 아래에 나와 있습니다.

LD



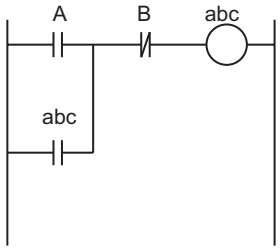
ST

```
RS_instance(A, B, abc);
```



참고

- RS 명령은 아래에 나와 있는 자체 유지 회로와 동일한 동작을 수행합니다.



- 단, RS 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 수행된 경우, 상기 자체 유지 회로와 다른 동작을 수행합니다.

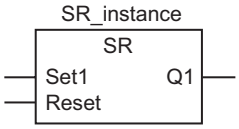
회로	B의 값	abc의 값
RS 명령	TRUE	유지
	FALSE	FALSE
자체 유지 회로	TRUE	FALSE
	FALSE	

사용상의 주의 사항

- 'Reset1' 입력을 b 접점의 외부 기기에서 직접 전송하지 마십시오. AC 전원 OFF 및 순간 정전 시에는 컨트롤러의 내부 전원이 곧바로 OFF되지 않고 입력 유니트의 입력이 먼저 ON이 되는 경우가 있습니다. 그 결과, 'Reset1' 입력의 값이 TRUE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q1'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'Q1'은 이전 실행 시의 값을 유지합니다.
- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 다음과 같이 동작합니다.
  - 'Reset1'의 값이 TRUE라면 'Q1'의 값을 유지하며, FALSE라면 'Q1'의 값은 FALSE가 됩니다.
  - 'Q1'의 다음 단계에 연결된 명령에는 'Q1'의 값이 TRUE라도 FALSE가 입력됩니다.

SR

BOOL 형식 변수의 값을 유지합니다. 세트 입력과 리셋 입력이 동시에 TRUE인 경우 세트 입력이 우선합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SR	세트 우선 유지	FB		SR_instance(Set1, Reset, Q1);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Set1*1	세트	입력	세트 입력	데이터 형식에 따름	–	0
Reset	리셋		리셋 입력			
Q1	유지	출력	유지 출력	데이터 형식에 따름	–	–

*1 Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Set1을 S1, Reset을 R로 기술할 수 있습니다. 예를 들어, RS_instance(S1:=A, R:=B, Q1=>abc);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Set1	○																			
Reset	○																			
Q1	○																			

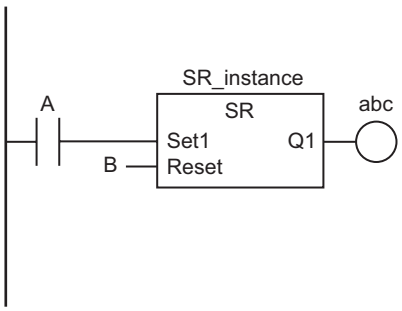
기능

세트 우선의 자체 유지 회로입니다. 입출력의 관계가 아래에 나와 있습니다.

'Set1'의 값	'Reset'의 값	'Q1'의 값
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	유지

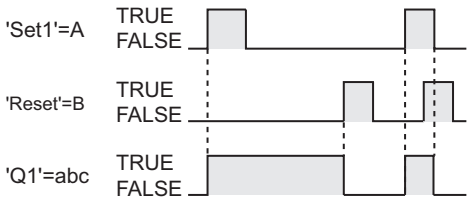
기술 예 및 타이밍 차트가 아래에 나와 있습니다.

LD



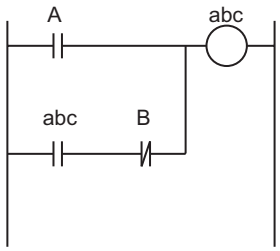
ST

```
SR_instance(A, B, abc);
```



참고

- SR 명령은 아래 게재한 자체 유지 회로와 동일한 동작을 수행합니다.



- 단, SR 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 수행된 경우, 상기 자체 유지 회로와 다른 동작을 수행합니다.

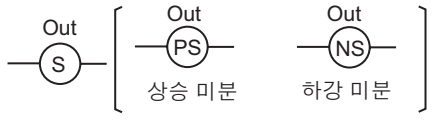
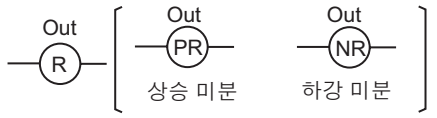
회로	B의 값	abc의 값
SR 명령	TRUE	유지
	FALSE	FALSE
자체 유지 회로	TRUE	FALSE
	FALSE	

사용상의 주의 사항

- 'Reset' 입력을 b 접점의 외부 기기에서 직접 전송하지 마십시오. AC 전원 OFF 및 순간 정전 시에는 컨트롤러의 내부 전원이 곧바로 OFF되지 않고 입력 유니트의 입력이 먼저 ON이 되는 경우가 있습니다. 그 결과, 'Reset' 입력의 값이 TRUE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q1'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'Q1'은 이전 실행 시의 값을 유지합니다.
- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 다음과 같이 동작합니다.
  - 'Reset'의 값이 TRUE라면 'Q1'의 값을 유지하며, FALSE라면 'Q1'의 값은 FALSE가 됩니다.
  - 'Q1'의 다음 단계에 연결된 명령에는 'Q1'의 값이 TRUE라도 FALSE가 입력됩니다.

Set / Reset

Set : BOOL 형식 변수에 TRUE를 세트합니다.
Reset : BOOL 형식 변수에 FALSE를 세트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Set	세트	–		없음
Reset	리셋	–		없음

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	출력	출력	출력	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			

기능

● Set

이 명령으로의 입력이 TRUE인 경우, 'Out'에 TRUE를 세트합니다.
일단 'Out'을 TRUE로 하면, 입력이 FALSE가 되더라도 'Out'은 FALSE가 되지 않습니다.
'Out'을 FALSE로 되돌리려면 Reset 명령을 사용합니다.

● Reset

이 명령으로의 입력이 TRUE인 경우, 'Out'에 FALSE를 세트합니다.
일단 'Out'을 FALSE로 하면, 입력이 FALSE가 되더라도 'Out'은 TRUE가 되지 않습니다.
'Out'을 TRUE로 되돌리려면 Set 명령을 사용합니다.

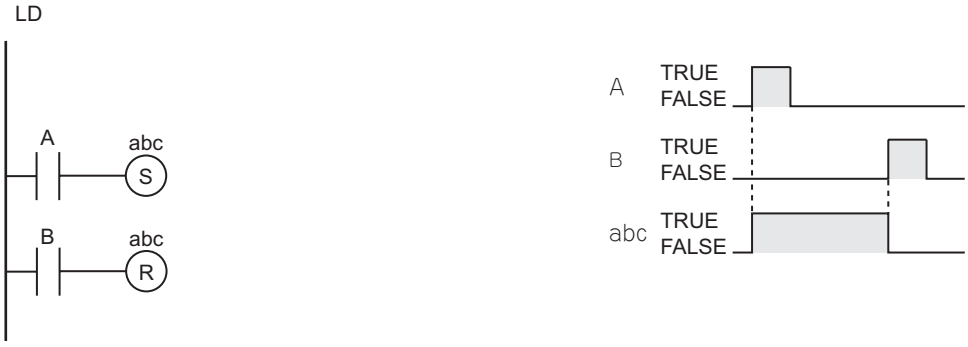
상승 미분, 하강 미분이 지정되지 않은 경우의 동작은 다음과 같습니다.

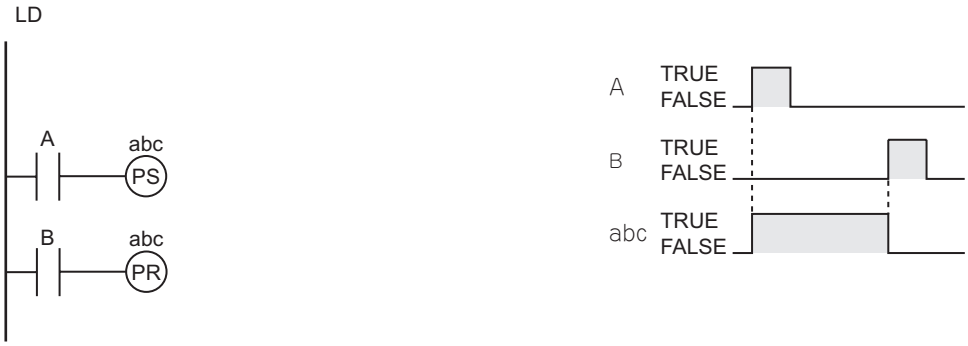
명령	입력	출력
Set	TRUE	TRUE
	FALSE	유지
Reset	TRUE	FALSE
	FALSE	유지

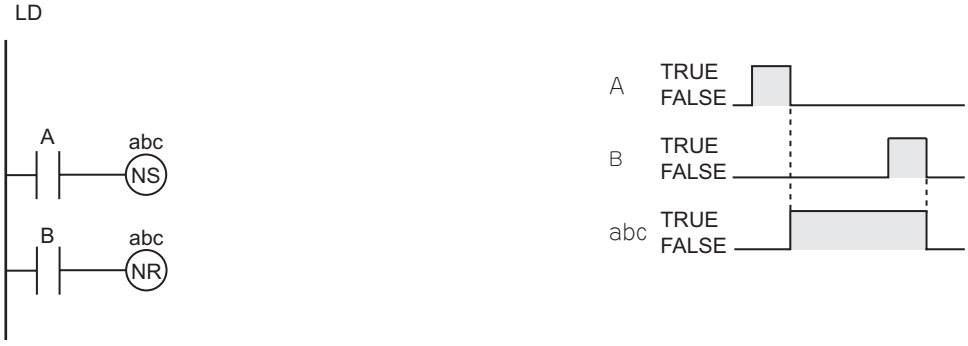
상승 미분, 하강 미분이 지정되는 경우, 이전 실행 시와 현재의 입력값에 따라 다음과 같이 동작합니다.

명령	미분 지정	이전 실행 시와 현재의 입력값	출력
Set	상승 미분	이전 실행 시 FALSE → 현재 TRUE	TRUE
		상기 이외	유지
	하강 미분	이전 실행 시 TRUE → 현재 FALSE	TRUE
		상기 이외	유지
Reset	상승 미분	이전 실행 시 FALSE → 현재 TRUE	FALSE
		상기 이외	유지
	하강 미분	이전 실행 시 TRUE → 현재 FALSE	FALSE
		상기 이외	유지

기술 예 및 타이밍 차트가 아래에 나와 있습니다.







참고

Set/Reset 명령과 Out 명령의 차이

- Set/Reset 명령은 입력이 True가 되었을 때 동작하며, 입력이 FALSE일 때는 동작하지 않습니다. 즉, 입력이 FALSE일 때는 출력에 변화가 없습니다.
- Out 명령은 이전 단계까지의 결과가 TRUE일 때 지정 변수를 TRUE로 하고, 이전 단계까지의 결과가 FALSE일 때는 지정 변수를 FALSE로 합니다. 즉, 입력이 TRUE거나 FALSE거나 모두 동작합니다.

Set/Reset 명령과 SR/RS 명령의 차이

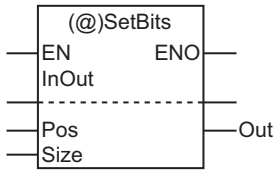
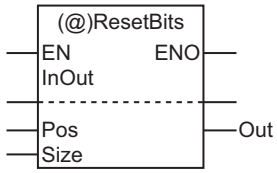
- SR/RS 명령은 'Set' 입력과 'Reset' 입력을 프로그램 내의 같은 위치에 기술해야 하지만, Set/Reset 명령은 각각 다른 위치에 기술할 수 있습니다.

사용상의 주의 사항

- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 'Out'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'Out'의 값을 유지합니다.
- 상승 미분이 지정되고 전원 기동 시에 입력이 TRUE인 경우, 그대로는 이 명령이 실행되지 않습니다. 입력이 일단 FALSE가 된 후, 다시 TRUE가 되었을 때 실행됩니다.
- 상승 미분이 지정되지 않고 전원 기동 시에 입력이 TRUE인 경우, 입력을 일단 FALSE로 하지 않더라도 이 명령은 실행됩니다.

SetBits / ResetBits

SetBits : 비트 열 데이터의 연속 복수 비트에 TRUE를 세트합니다.
ResetBits : 비트 열 데이터의 연속 복수 비트에 FALSE를 세트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetBits	다중 비트 세트	FUN		SetBits(InOut, Pos, Size);
ResetBits	다중 비트 리셋	FUN		ResetBits(InOut, Pos, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	비트 열	입출력	비트 열	데이터 형식에 따름	–	–
Pos	비트 위치	입력	지정 비트 위치	0~'InOut'의 비트 수-1	–	0
Size	비트 수		비트 수	0~'InOut'의 비트 수		1
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut		○	○	○	○															
Pos						○														
Size						○														
Out	○																			

기능

● SetBits

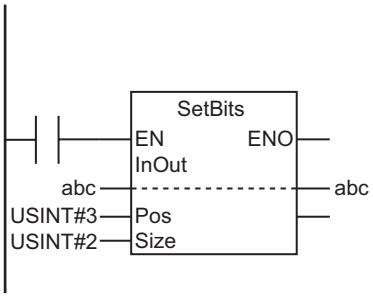
비트 열 'InOut'의 비트 위치 'Pos'에서 'Size'개의 비트값을 TRUE로 세트합니다. 그 이외의 비트는 변화하지 않습니다.

● ResetBits

비트 열 'InOut'의 비트 위치 'Pos'에서 'Size'개의 비트값을 FALSE로 세트합니다. 그 이외의 비트는 변화하지 않습니다.

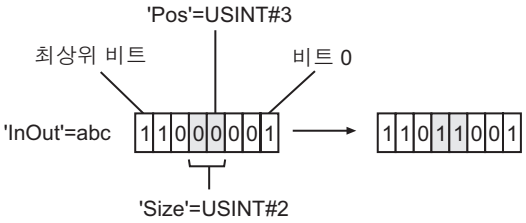
SetBits 명령에서 'Pos'=USINT#3, 'Size'=USINT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
SetBits(abc, USINT#3, USINT#2);
```



참고

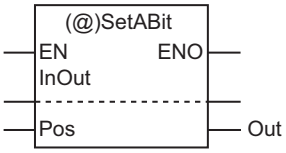
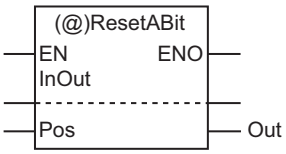
이 명령을 사용하면 데이터 메모리 등의 채널(워드) 단위로 데이터를 다루는 메모리 영역에 AT 지정을 한 변수에 대해, 일괄적으로 TRUE/FALSE를 세트/리셋할 수 있습니다.

사용상의 주의 사항

- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 'InOut'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'InOut'의 값을 유지합니다.
- 'Size'의 값이 0인 경우, 'InOut'의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', 'InOut'은 변화하지 않습니다.
  - 'Pos'의 값이 'InOut'의 비트 수-1보다 큰 경우
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Pos'와 'Size'의 지정이 'InOut'의 비트 수를 초과하는 경우

SetABit / ResetABit

SetABit : 비트 열 데이터의 지정 비트에 TRUE를 세트합니다.
ResetABit : 비트 열 데이터의 지정 비트에 FALSE를 세트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetABit	1비트 세트	FUN		SetABit (InOut, Pos);
ResetABit	1비트 리셋	FUN		ResetABit (InOut, Pos);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	비트 열	입출력	비트 열	데이터 형식에 따름	–	–
Pos	비트 위치	입력	지정 비트 위치	0~'InOut'의 비트 수-1	–	0
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut		○	○	○	○															
Pos						○														
Out	○																			

기능

● SetABit

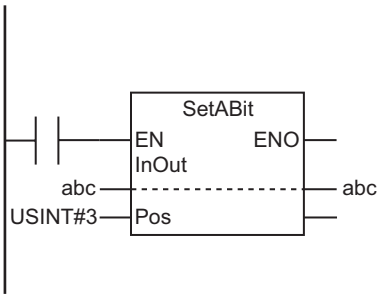
비트 열 'InOut'의 비트 위치 'Pos'의 값을 TRUE로 세트합니다. 그 이외의 비트는 변화하지 않습니다.
실행 후에 EN이 FALSE가 되더라도 'InOut'의 'Pos' 비트는 변화하지 않습니다.

● ResetABit

비트 열 'InOut'의 비트 위치 'Pos'의 값을 FALSE로 세트합니다.그 이외의 비트는 변화하지 않습니다.
실행 후에 EN이 FALSE가 되더라도 'InOut'의 'Pos' 비트는 변화하지 않습니다.

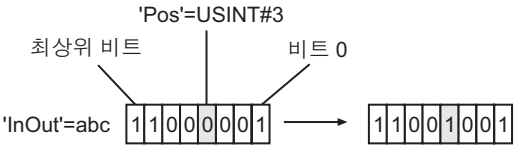
SetABit 명령에서 'Pos'=USINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
SetABit(abc, USINT#3);
```



참고

SetABit/ResetABit 명령과 OutABit 명령의 차이

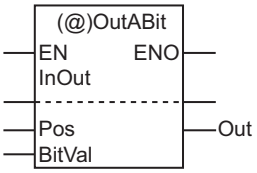
- SetABit/ResetABit 명령은 지정 비트에 세트하는 값이 TRUE 또는 FALSE로 고정되어 있습니다.
- OutABit 명령은 지정 비트에 세트하는 값을 동적으로 바꿀 수 있습니다.

사용상의 주의 사항

- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 'InOut'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'InOut'의 값을 유지합니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', 'InOut'은 변화하지 않습니다.
  - 'Pos'의 값이 'In'의 비트 수-1보다 큰 경우

OutABit

비트 열 데이터의 지정 비트에 TRUE/FALSE를 세트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
OutABit	1비트 출력	FUN		OutABit (InOut, Pos, BitVal);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	비트 열	입출력	비트 열	데이터 형식에 따름	–	–
Pos	비트 위치	입력	지정 비트 위치	0~'InOut'의 비트 수-1	–	0
BitVal	세트값		세트하는 값	데이터 형식에 따름		TRUE
Out	반환값	출력	항상 TRUE	TRUE만	–	–

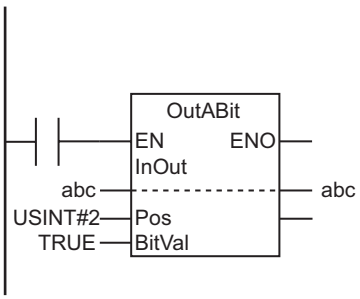
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut		○	○	○	○															
Pos						○														
BitVal	○																			
Out	○																			

기능

비트 열 'InOut'의 비트 위치 'Pos'에 세트값 'BitVal'의 값을 저장합니다. 'Pos' 이외의 비트 위치의 값은 변화하지 않습니다.

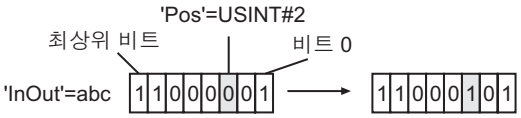
'Pos'=USINT#2, 'BitVal'=TRUE인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
OutABit(abc, USINT#2, TRUE);
```



참고

SetABit/ResetABit 명령과 OutABit 명령의 차이

- SetABit/ResetABit 명령은 지정 비트에 세트하는 값이 TRUE 또는 FALSE로 고정되어 있습니다.
- OutABit 명령은, 'BitVal'의 값을 변화시킴으로써 지정 비트에 세트하는 값을 동적으로 바꿀 수 있습니다.

사용상의 주의 사항

- 이 명령이 마스터 컨트롤 영역에 있으며 마스터 컨트롤에 의한 리셋이 이루어진 경우, 'InOut'의 값을 유지합니다.
- 점프 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'InOut'의 값을 유지합니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', 'InOut'은 변화하지 않습니다.
  - 'Pos'의 값이 'InOut'의 비트 수-1보다 큰 경우

시퀀스 제어 명령

명령	명칭	페이지
End	종료	2-64
RETURN	복귀	2-65
MC/MCR	마스터 컨트롤 시작/ 마스터 컨트롤 종료	2-66
JMP	점프	2-81
FOR/NEXT	반복 시작/반복 종료	2-83
BREAK	루프 중단	2-90

End

프로그램의 현재 태스크 주기의 실행을 종료합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
End	종료	FUN	End EN ENO	없음

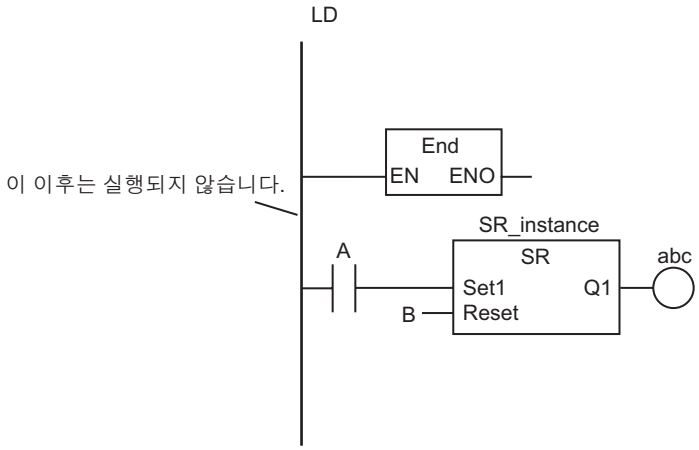
변수

없음

기능

프로그램의 현재 태스크 주기의 실행을 종료합니다.

기술 예가 아래에 나와 있습니다. 이 예의 경우, End 명령이 실행되면 그 다음의 SR 명령은 실행되지 않습니다.



사용상의 주의 사항

- 이 명령은 프로그램 내에서만 사용하십시오.
- 이 명령을 펑션, 펑션 블록, 인라인 ST 내에서 사용한 경우, 빌드 시에 이상 상태가 됩니다.
- 이 명령은 반드시 좌측 모선에 직접 연결하십시오.

RETURN

펑션 또는 펑션 블록을 종료하고 처리를 호출원으로 되돌립니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RETURN	복귀	FUN	RETURN EN ENO	RETURN

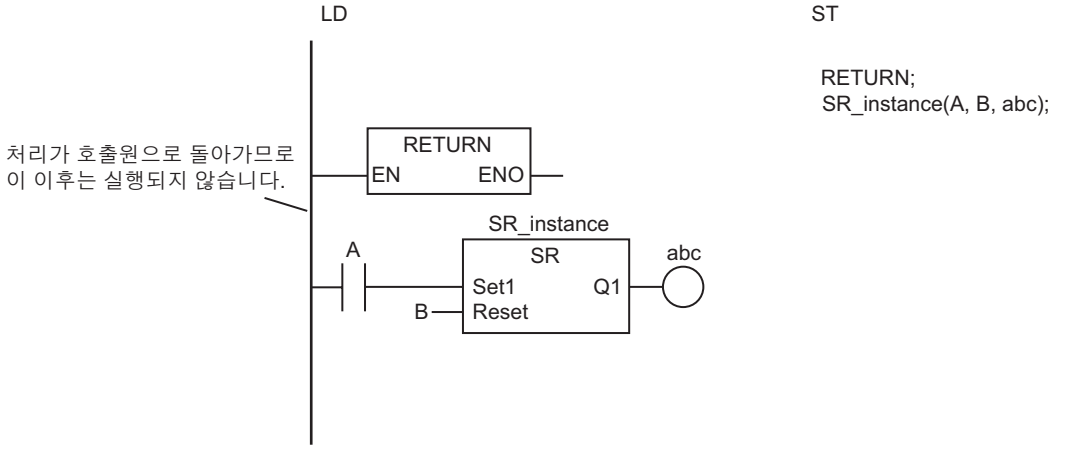
변수

없음

기능

펑션 또는 펑션 블록을 종료하고 처리를 호출원으로 되돌립니다.

기술 예가 아래에 나와 있습니다. 이 예의 경우, RETURN 명령이 실행되면 그 다음의 SR 명령은 실행되지 않습니다.



사용상의 주의 사항

- 이 명령을 래더도 프로그램에서 사용하는 경우, 다음 사항에 주의하십시오.
  - 이 명령은 펑션 또는 펑션 블록 내에서 사용하십시오. 프로그램 내에서 사용하면 빌드 시에 이상 상태가 됩니다.
  - 이 명령은 반드시 좌측 모선에 직접 연결하십시오.
- 이 명령을 실행하기 전에, 해당 POU의 반환값, 출력 변수, ENO의 값을 세트하십시오.
- 이 명령을 너무 자주 사용하면 처리의 흐름을 파악하기 어려워집니다. 주의하십시오.

MC / MCR

MC : 마스터 컨트롤 영역의 시작점을 표시하고 마스터 컨트롤에 의한 리셋을 실행합니다.
MCR : 마스터 컨트롤 영역의 종료점을 표시합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC	마스터 컨트롤 시작	–	MC In ENO MCNo	없음
MCR	마스터 컨트롤 종료	–	MCR EN ENO MCNo	없음

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In (MC 명령만)	마스터 컨트롤 입력	입력	FALSE: 마스터 컨트롤에 의한 리셋 실행	데이터 형식에 따름	–	–
MCNo	마스터 컨트롤 번호		마스터 컨트롤 번호	0~14(*)		0

* Sysmac Studio에 의해 자동 등록됩니다. 사용자는 설정할 필요가 없습니다.

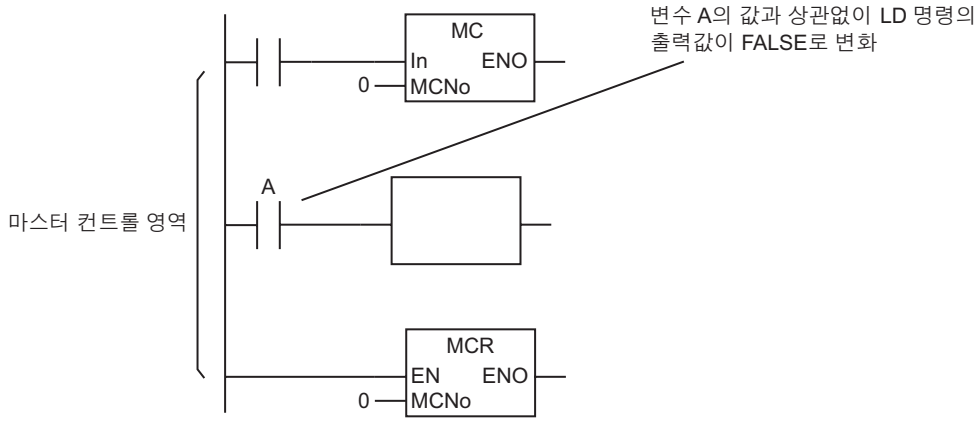
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In (MC 명령만)	○																			
MCNo							○													

기능

마스터 컨트롤이란, 사용자 프로그램 내의 지정된 영역에 존재하는 모든 POU를 처리 정지 또는 그에 준하는 상태로 만드는 제어를 의미합니다. 마스터 컨트롤 기능을 사용하면 비교적 긴 일련의 처리의 실행 조건을 간단히 제어할 수 있습니다.

마스터 컨트롤로 제어하는 사용자 프로그램 내의 영역을 마스터 컨트롤 영역이라고 부릅니다. 마스터 컨트롤 영역의 시작 위치에 MC 명령을, 종료 위치에 MCR 명령을 배치합니다.
MC 명령의 마스터 컨트롤 입력 'In'의 값이 FALSE가 되면, 다음 그림과 같이 마스터 컨트롤 영역의 좌측 모선에 연결된 LD 명령의 출력이 강제적으로 FALSE가 됩니다. 이 상태를 마스터 컨트롤에 의한 리셋이라고 부릅니다.

마스터 컨트롤에 의한 리셋이 실행되면 LD 명령 다음 단계의 POU는 원칙적으로 실행 조건이 FALSE가 된 경우의 동작을 수행합니다. 단, POU에 따라서는 원칙대로 동작하지 않는 것도 있습니다. 이에 대해서는 뒤에 다룹니다.



'In'의 값이 TRUE인 경우, 마스터 컨트롤에 의한 리셋은 발생하지 않습니다. 마스터 컨트롤 영역에 있는 POU는 통상 동작을 수행합니다.

마스터 컨트롤에 의한 리셋을 실행 중인 각 POU의 동작

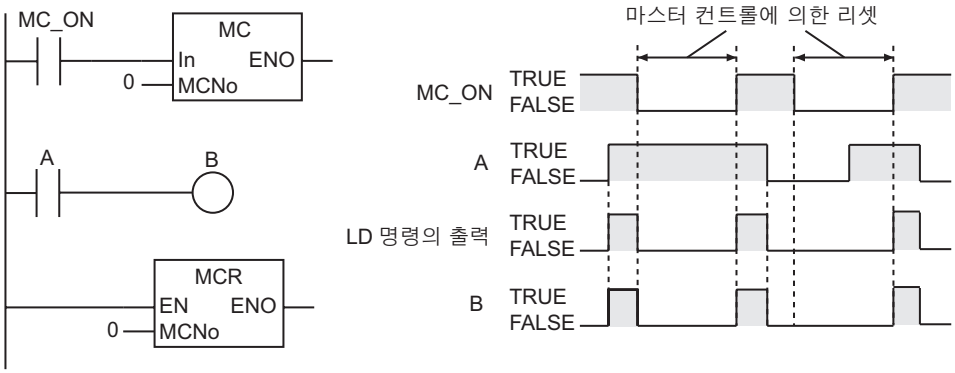
마스터 컨트롤에 의한 리셋 실행 중의 동작은 POU에 따라 다음과 같이 달라집니다.

POU	동작
Out/OutABit 명령	지정된 변수에 FALSE를 출력합니다.
OutNot 명령	지정된 변수에 FALSE를 출력합니다.
Set/Reset 명령	마스터 컨트롤에 의한 리셋 실행 전의 출력을 유지합니다.
TON 명령	타이머 입력 'In'의 값이 FALSE인 경우의 동작을 수행합니다. 즉, 타이머가 리셋됩니다. 경과 시간 'ET'의 값이 0, 타이머 출력 'Q'의 값이 FALSE가 됩니다.
TOF 명령	타이머 입력 'In'의 값이 TRUE인 경우의 동작을 수행합니다. 즉, 타이머가 리셋됩니다. 경과 시간 'ET'의 값이 0, 타이머 출력 'Q'의 값이 TRUE가 됩니다. 단, 'Q'를 다음 단계의 Out 명령에 연결한 경우, Out 명령에는 FALSE가 입력됩니다.
TP 명령	타이머 입력 'In'의 값이 FALSE인 경우의 동작을 수행합니다. 즉, 타이머가 리셋됩니다. 계시 중 : 경과 시간 'ET'의 값은 계시 종료까지 증가하며, 종료 후에 0이 됩니다. 타이머 출력 'Q'의 값은 계시 종료까지 TRUE 종료 후에 FALSE가 됩니다. 계시 중 이외: 'ET'의 값은 0, 'Q'의 값은 FALSE가 됩니다. 단, 'Q'를 다음 단계의 Out 명령에 연결한 경우, 계시 중이라도 Out 명령에는 FALSE가 입력됩니다.
AccumulationTimer 명령	타이머 입력 'In'의 값이 FALSE인 경우의 동작을 수행합니다. 즉, 타이머가 정지됩니다. 적산 시간 'ET'와 타이머 출력 'Q'도 모두 값을 유지합니다. 단, 'Q'를 다음 단계의 Out 명령에 연결한 경우, 'Q'의 값이 TRUE라도 Out 명령에는 FALSE가 입력됩니다. 또한, 리셋 'Reset'은 유효합니다.
Timer 명령	타이머 입력 'In'의 값이 FALSE인 경우의 동작을 수행합니다. 즉, 타이머가 리셋됩니다. 남은 시간 'ET'에 설정 시간 'PT'의 값이 세트되며, 타이머 출력 'Q'의 값이 FALSE가 됩니다.
CTU/CTD/CTUD 명령	실행되지 않습니다. 마스터 컨트롤에 의한 리셋 실행 전에 이러한 명령이 실행 중이었을 경우, 카운트값은 리셋 실행 전의 값을 유지합니다. 카운트 업 플래그 'Q'를 이후 단계의 Out 명령에 연결한 경우, 그 Out 명령에는 FALSE가 입력됩니다.
JMP 명령	실행되지 않습니다.
FOR/NEXT 명령	실행되지 않습니다.
BREAK 명령	실행되지 않습니다.
복수의 태스크 주기에 걸쳐 실행되는 펑션 블록 (출력 변수로 'Done', 'Busy', 'Error'를 갖는 명령)	좌측 모선에 연결된 실행 조건의 값이 FALSE가 됩니다. 마스터 컨트롤에 의한 리셋 실행 전에 이 명령이 실행 중이었을 경우, 처리가 완료되기까지 계속해서 실행됩니다. 그 경우, 'Done', 'Busy', 'Error'의 출력은 수행하지 않지만, 그 다음 단계에 코일을 연결한 경우 반드시 FALSE를 출력합니다. 'Done', 'Busy', 'Error'에 변수를 직접 연결한 경우, 그 변수에는 명령의 사양과 같은 값이 대입됩니다. 'Done', 'Busy', 'Error'의 값을 '인스턴스명.출력 변수'의 형식으로 취득할 수도 있습니다.
상기 이외의 펑션	실행되지 않습니다.
상기 이외의 펑션 블록	좌측 모선에 연결된 실행 조건의 값이 FALSE가 됩니다.

대표적인 명령의 동작에 대해 자세히 설명합니다.

● Out

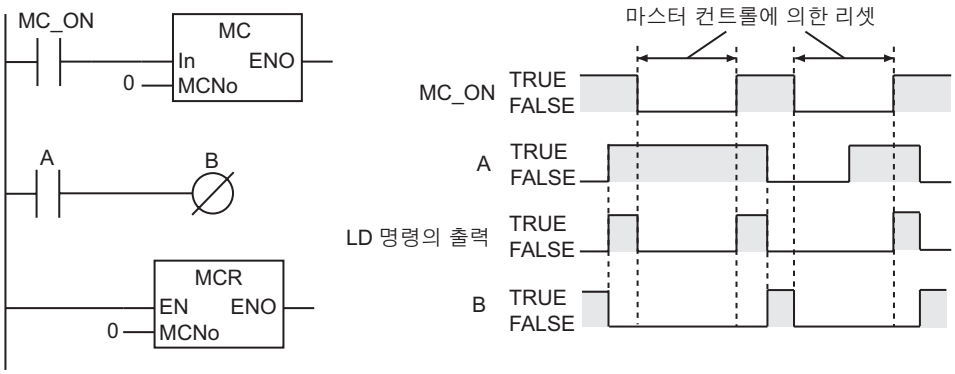
마스터 컨트롤에 의한 리셋 실행 중에는 FALSE를 출력합니다.



● OutNot

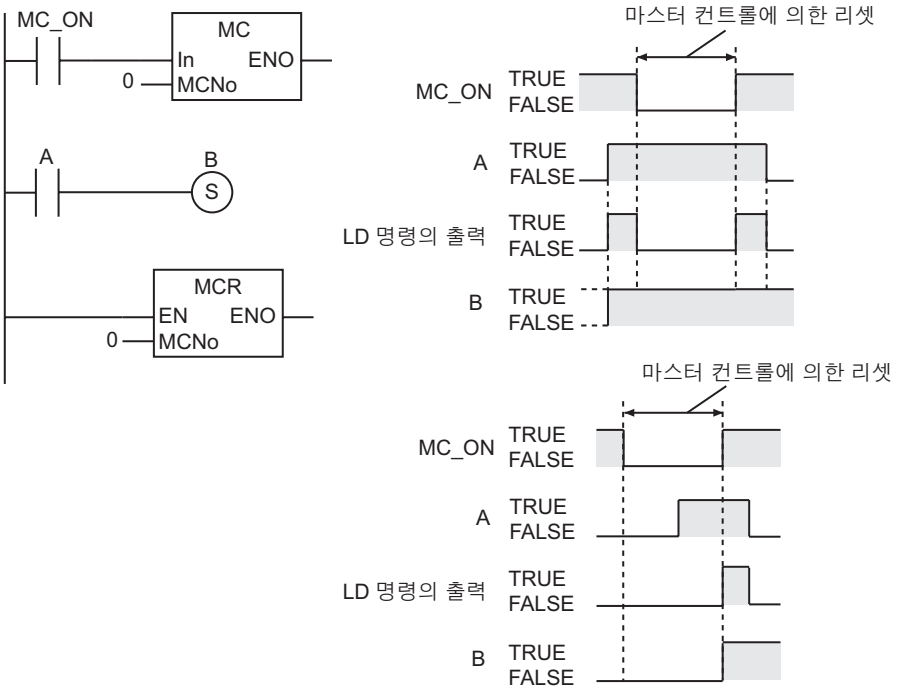
마스터 컨트롤에 의한 리셋 실행 중에는 FALSE를 출력합니다.

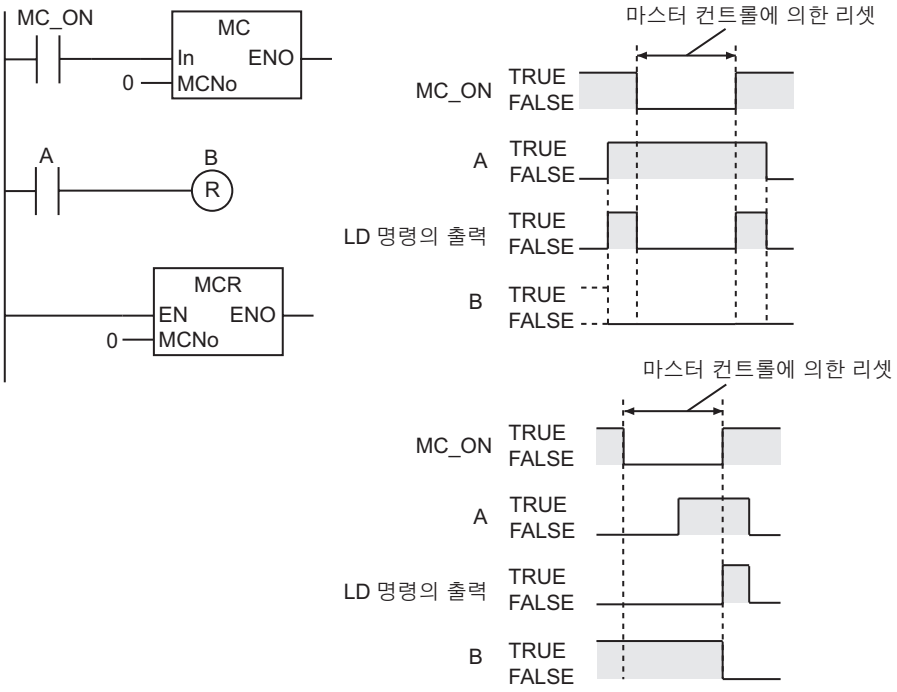
이전 단계의 LD 명령의 출력이 FALSE가 된 경우의 통상적인 OutNot 명령 동작과는 다르므로 주의하십시오.



● Set/Reset

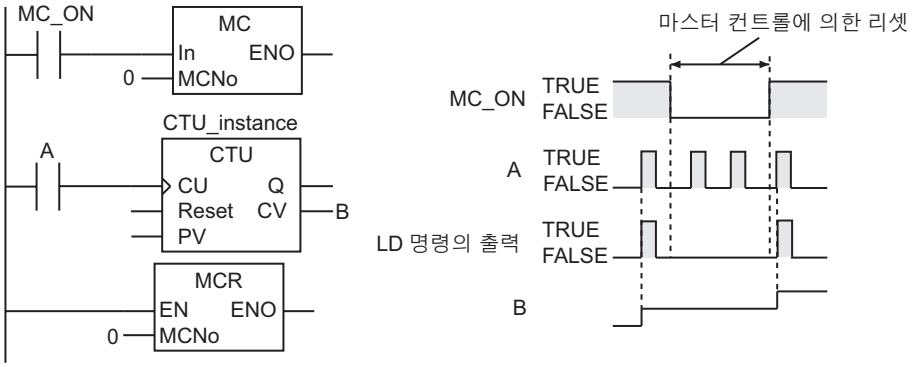
마스터 컨트롤에 의한 리셋 실행 중에는 그 이전의 출력값을 유지합니다.





● CTU/CTD/CTUD

마스터 컨트롤에 의한 리셋 실행 중에는 그 이전의 카운터값을 유지합니다. 마스터 컨트롤에 의한 리셋 해제 후에는, 유지하던 카운터값에서 카운트를 속행합니다.



입력 상승 미분, 입력 하강 미분 지정이 있는 POU의 동작

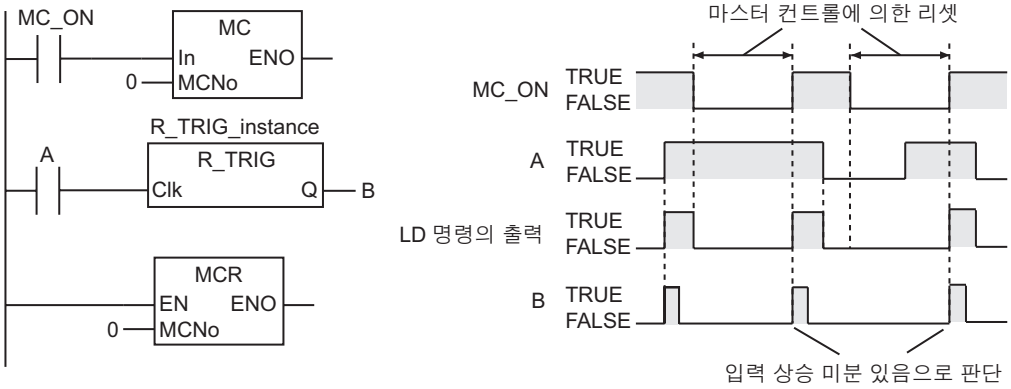
입력 상승 미분 지정, 입력 하강 미분 지정이 있는 POU는 다음과 같습니다.

미분 형식	명령
입력 상승 미분 형식	• 상승 미분 지정이 있는 LD, LDN, AND, ANDN, OR, ORN, Out • R_TRIG, Up • 입력 상승 미분 형식의 동작 옵션(@)이 있는 펑션 • 입력 상승 미분 형식 지정이 있는 펑션 블록(카운터 명령 등)
입력 하강 미분 형식	• 하강 미분 지정이 있는 LD, LDN, AND, ANDN, OR, ORN, Out • F_TRIG, Down

마스터 컨트롤에 의한 리셋이 실행 또는 해제되면 이러한 POU의 실행 조건의 값이 변화합니다. 그로 인해 이러한 POU의 실행 조건의 상승 미분 및 하강 미분 조건이 성립되는 경우가 있습니다. 이렇게 발생하는 실행 조건의 상승 미분 및 하강 미분도 유효하게 기능합니다.
대표적인 명령의 동작에 대해 자세히 설명합니다.

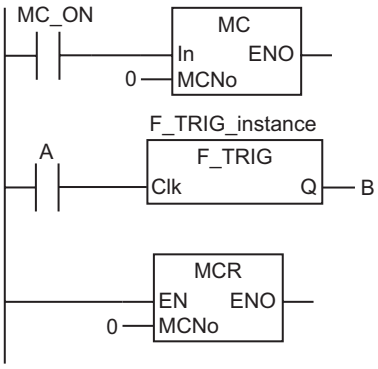
● R_TRIG, Up

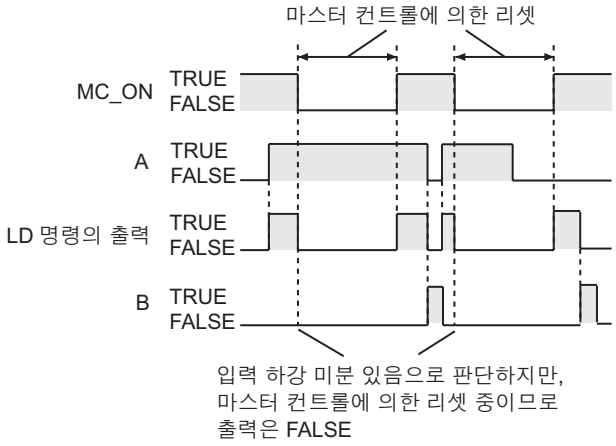
마스터 컨트롤에 의한 리셋이 실행되면 실행 조건의 값이 FALSE가 됩니다. 그 후, 마스터 컨트롤에 의한 리셋 해제 시에 실행 조건의 값이 TRUE라면 입력 상승 미분이 발생한 것으로 판단하여 동작을 수행합니다.



● F_TRIG, Down

마스터 컨트롤에 의한 리셋이 실행되면 실행 조건의 값이 FALSE가 됩니다. 그 시점까지의 실행 조건의 값이 TRUE라면, 그로 인해 입력 하강 미분이 발생한 것으로 판단합니다. 하지만 마스터 컨트롤에 의한 리셋이 실행 중인 F_TRIG, Down의 출력값은 강제적으로 FALSE가 되므로, 결과적으로 출력값은 FALSE가 됩니다.

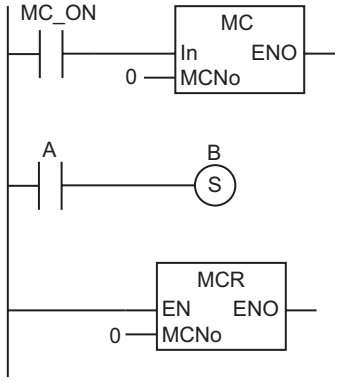


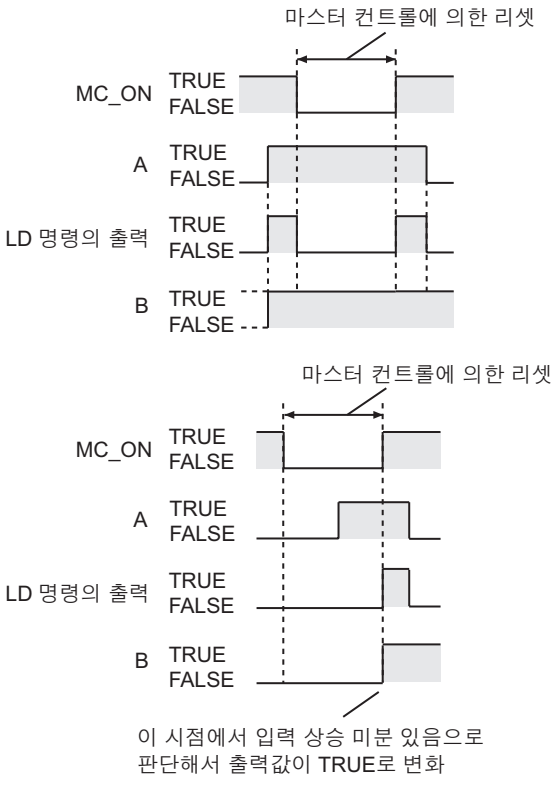


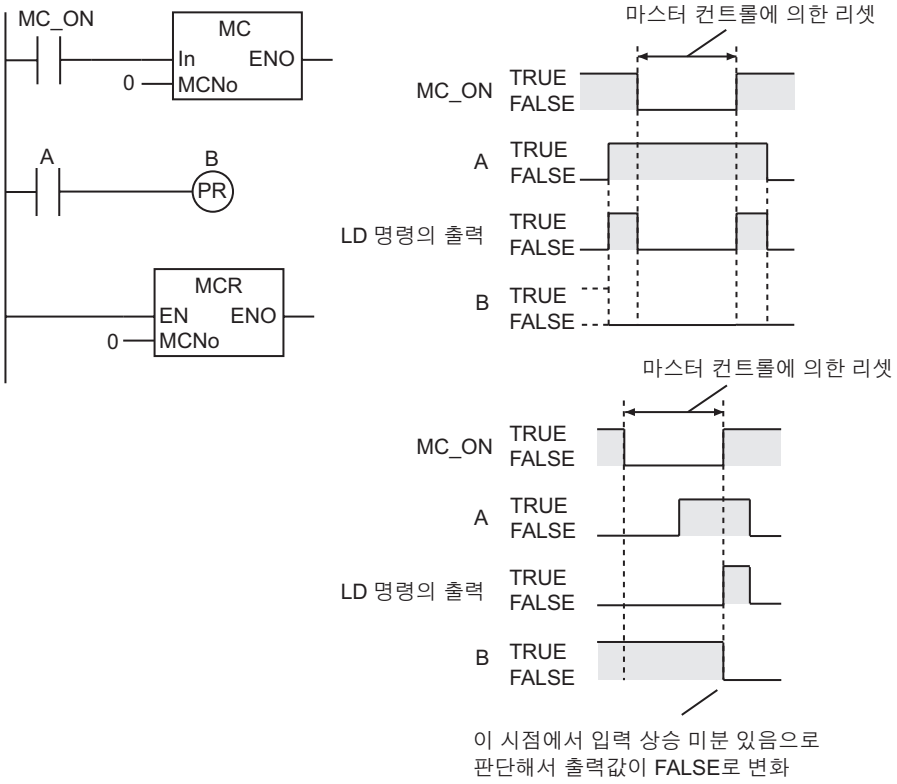
● 입력 상승 미분 지정이 있는 Set/Reset

마스터 컨트롤에 의한 리셋 실행 중에는 그 이전의 출력값을 유지합니다.

마스터 컨트롤에 의한 리셋 해제 시에, 실행 조건이 FALSE→TRUE가 되면 그 시점에서 동작합니다.

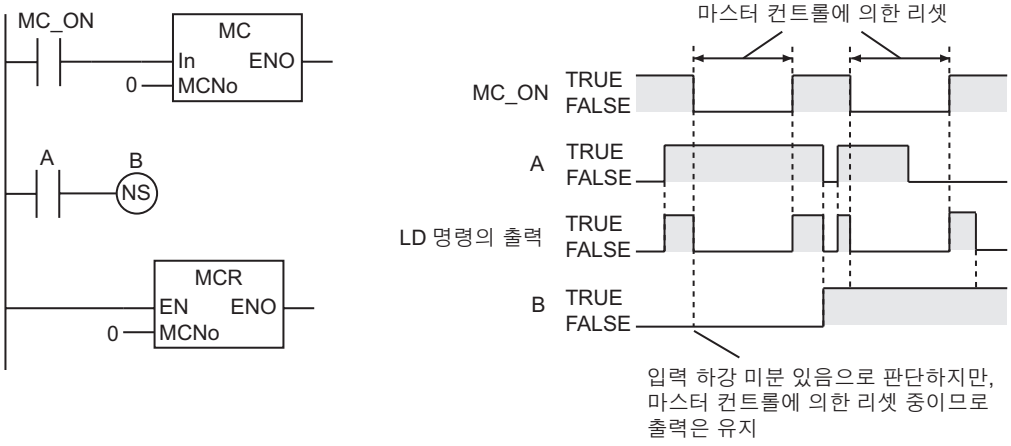


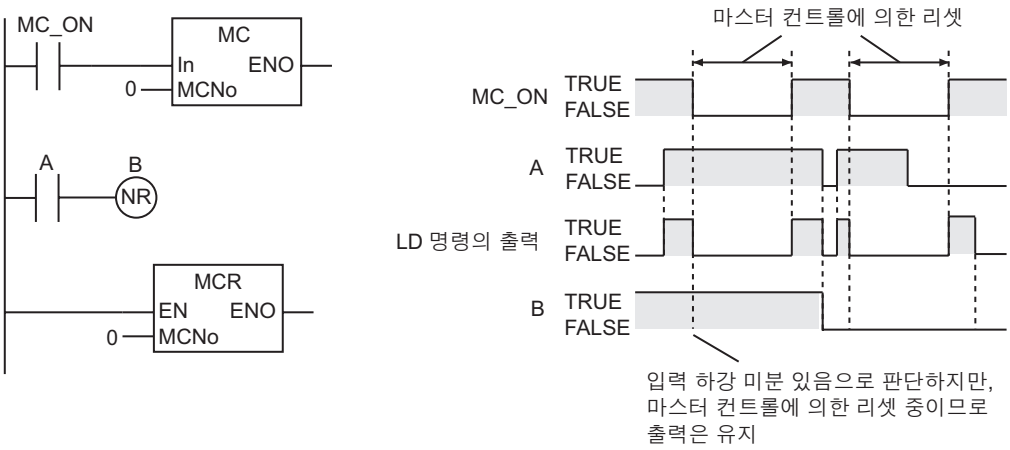




● 입력 하강 미분 지정이 있는 Set/Reset

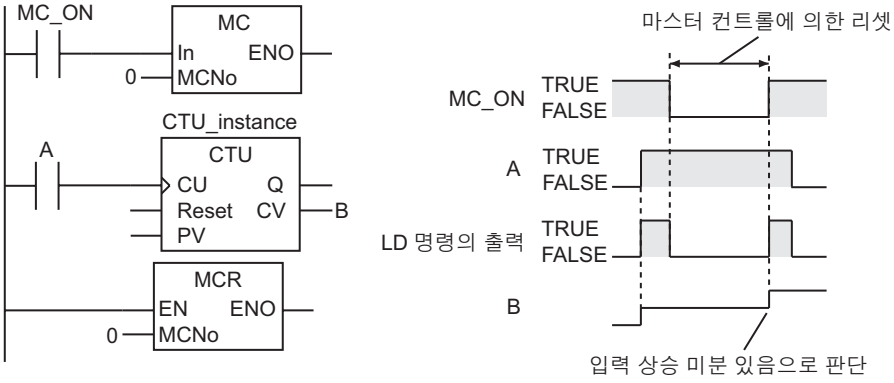
마스터 컨트롤에 의한 리셋이 실행되면 실행 조건의 값이 FALSE가 됩니다. 그 시점까지의 실행 조건의 값이 TRUE라면, 그로 인해 입력 하강 미분이 발생한 것으로 판단합니다. 하지만 마스터 컨트롤에 의한 리셋이 실행 중인 출력값은 그 이전의 값을 유지하므로, 결과적으로 출력값은 유지됩니다.





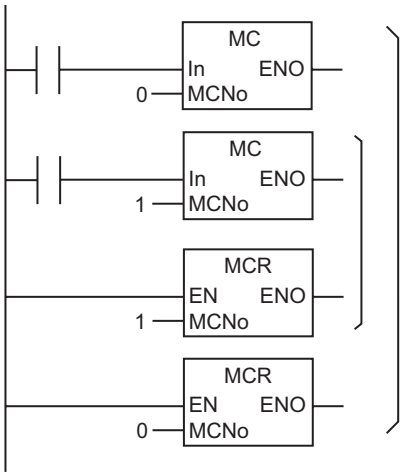
● CTU/CTD/CTUD

마스터 컨트롤에 의한 리셋이 실행되면 카운터 입력값이 FALSE가 됩니다. 그 후, 마스터 컨트롤에 의한 리셋 해제 시에 카운터 입력값이 TRUE라면 입력 상승 미분이 발생한 것으로 판단하여 카운트합니다.

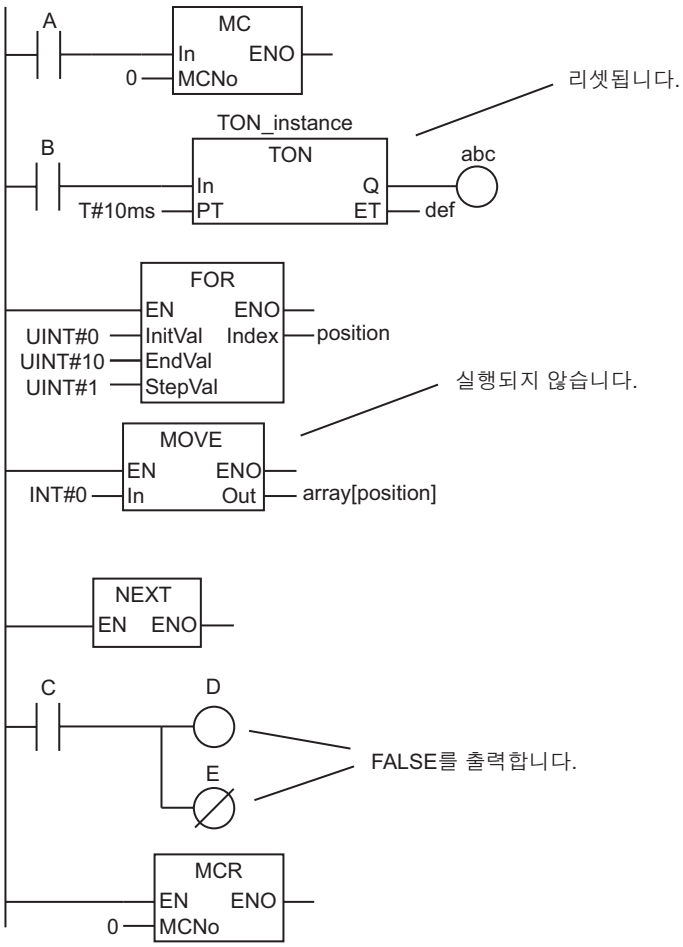


MC 명령과 MCR 명령은 반드시 동일한 POU 내에서 짝을 지어 사용합니다. 짝을 이루는 MC 명령과 MCR 명령에는 마스터 컨트롤 번호 'MCNo'에 같은 값이 할당됩니다. 'MCNo'의 값은 사용자가 임의로 결정하는 것이 아니라, Sysmac Studio에 의해 자동 등록됩니다.

MC 명령과 MCR 명령은 15단계까지 중첩할 수 있습니다.



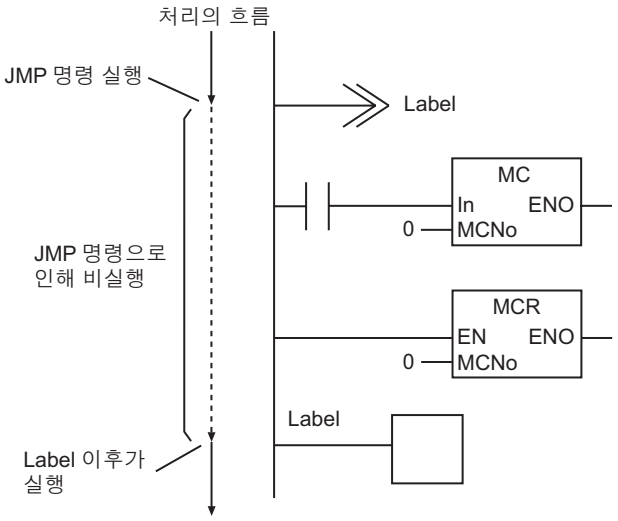
기술 예가 아래에 나와 있습니다.
변수 A의 값이 FALSE가 되면, 마스터 컨트롤에 의한 리셋을 실행합니다. 그 동안 TON 명령은 리셋됩니다. MOVE 명령은 실행되지 않습니다. 또한, Out 명령과 OutNot 명령도 변수 D와 E에 FALSE를 출력합니다.



사용상의 주의 사항

- 이 명령은 래더도 프로그램에서만 사용 가능합니다. ST 프로그램에서는 사용할 수 없습니다. 래더도 프로그램의 인라인 ST 내에서도 사용할 수 없습니다.
- 'In'은 반드시 좌측 모선에 연결하십시오. 'In'에 변수나 정수를 전달할 수는 없습니다.
- MC 명령과 MCR 명령은 반드시 동일한 POU 내에서 짝을 지어 사용하십시오.
- MCR 명령은 반드시 MC 명령 뒤에 배치하십시오.
- MC 명령과 MCR 명령은 16단 이상 중첩할 수 없습니다.
- 마스터 컨트롤 영역에 인라인 ST가 포함된 경우, 마스터 컨트롤에 의한 리셋이 실행되면 해당 인라인 ST의 기술은 실행되지 않습니다.

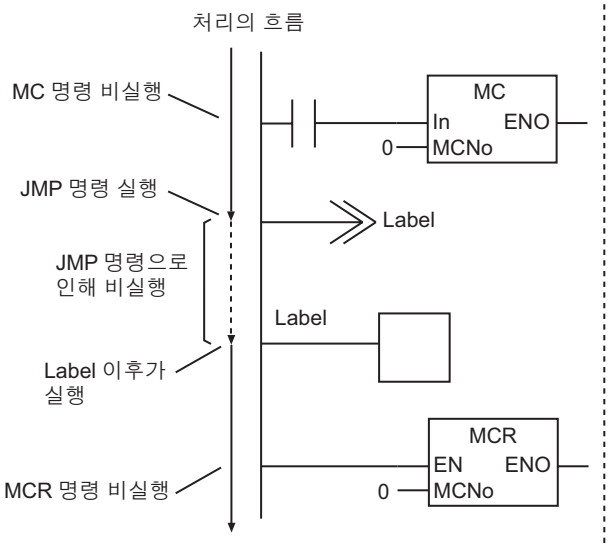
- MC/MCR 명령과 JMP 명령을 조합하여 사용하는 경우, 다음과 같이 동작합니다.
  - 아래 그림은 JMP-Label의 안쪽에 MC-MCR이 있습니다. 따라서 'In'의 값에 관계 없이 점프가 실행됩니다.



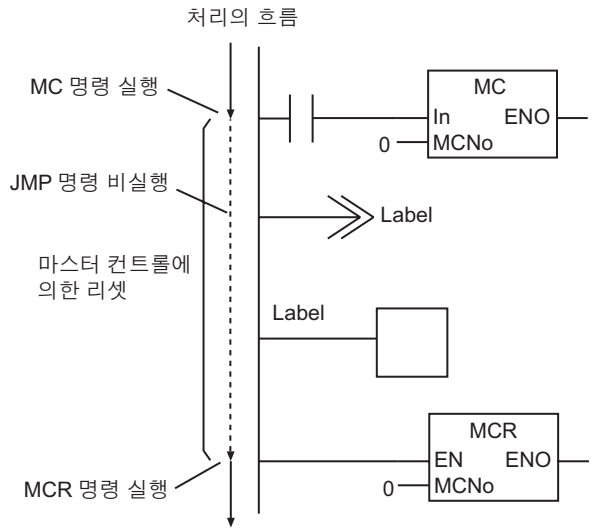
- 아래 그림은 MC-MCR의 안쪽에 JMP-Label이 있습니다. 따라서 다음 표와 같이 동작합니다.

'In'의 값	동작
TRUE	마스터 컨트롤에 의한 리셋은 실행되지 않습니다. 점프는 실행됩니다.
FALSE	마스터 컨트롤에 의한 리셋이 실행됩니다. 점프는 실행되지 않습니다.

● 'In'=TRUE인 경우

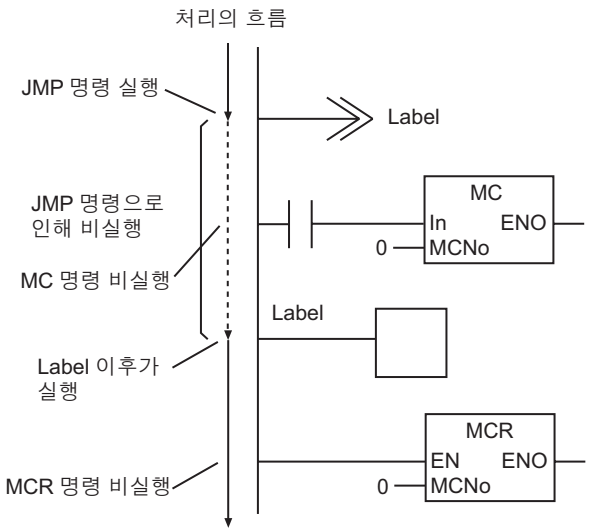


● 'In'=FALSE인 경우

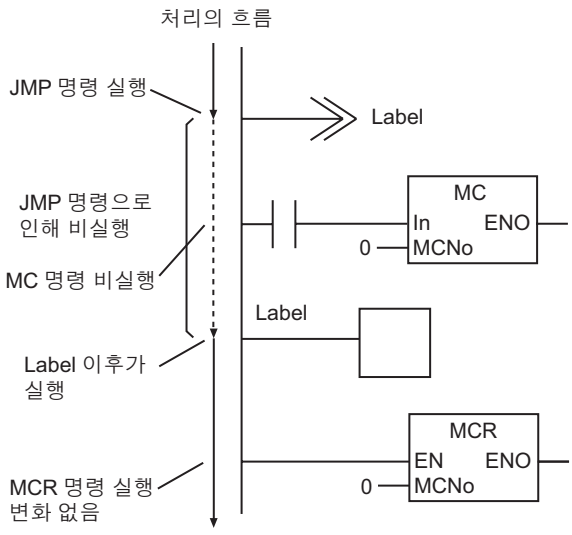


- 아래 그림은 JMP 명령, MC 명령, Label, MCR 명령 순으로 나열되어 있습니다. 따라서, 먼저 점프가 실행됩니다. 그 결과, MC 명령은 실행되지 않습니다. 그러므로 Label 이후는 실행됩니다. 'In'의 값이 FALSE라면 MCR 명령은 실행되지만, 변화는 없습니다.

● 'In'=TRUE인 경우



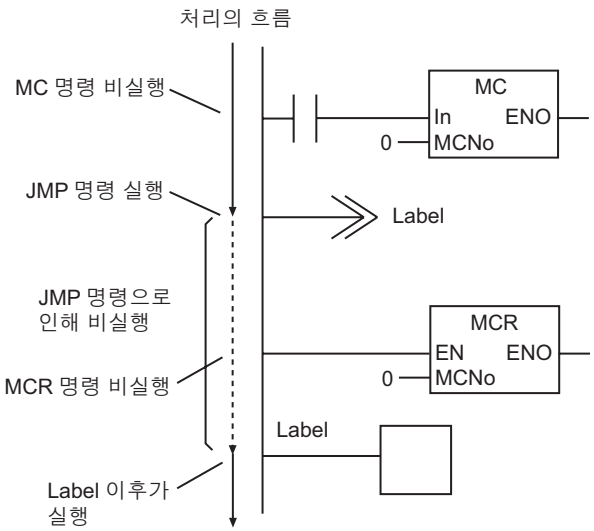
● 'In'=FALSE인 경우



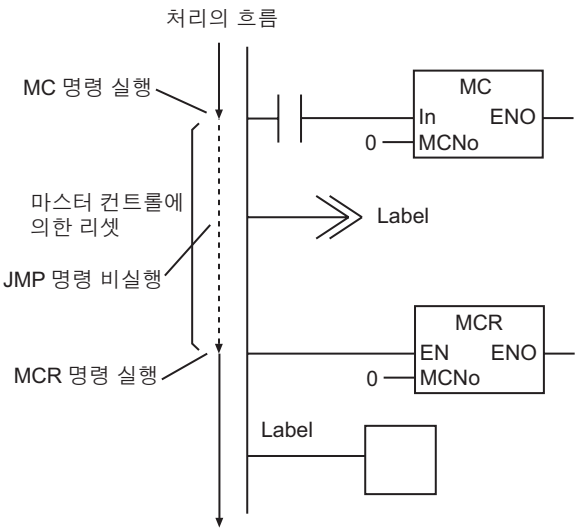
- 아래 그림은 MC 명령, JMP 명령, MCR 명령, Label 순으로 나열되어 있습니다. 따라서 다음 표와 같이 동작합니다.

'In'의 값	동작
TRUE	마스터 컨트롤에 의한 리셋은 실행되지 않습니다. 점프는 실행됩니다.
FALSE	마스터 컨트롤에 의한 리셋이 실행됩니다. 점프는 실행되지 않습니다.

● 'In'=TRUE인 경우



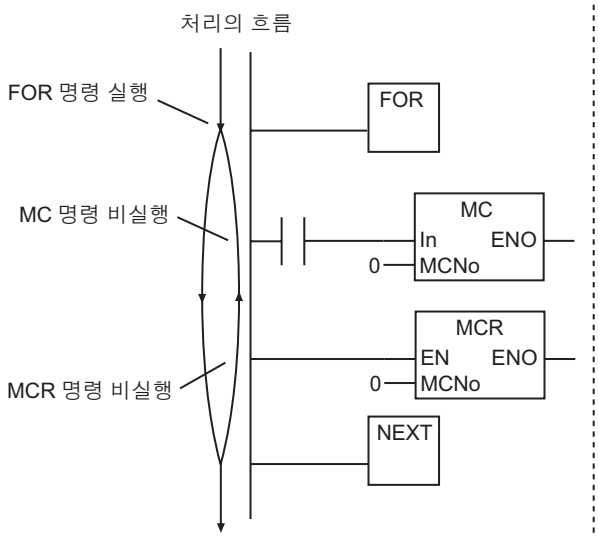
● 'In'=FALSE인 경우



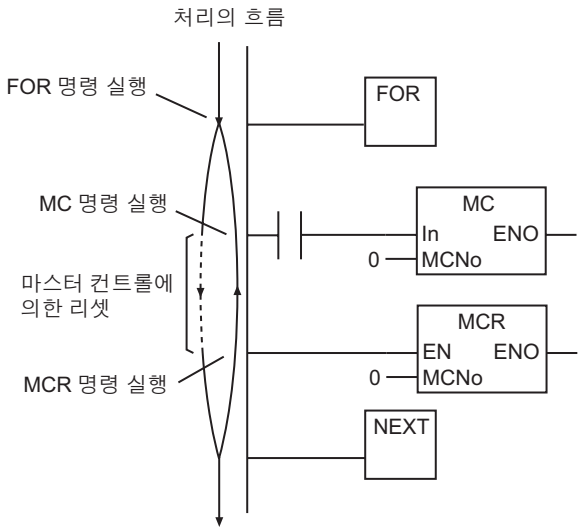
- MC/MCR 명령과 FOR/NEXT 명령을 조합하여 사용하는 경우, 다음과 같이 동작합니다.
  - 아래 그림은 FOR-NEXT의 안쪽에 MC-MCR이 있습니다. 따라서 다음 표와 같이 동작합니다.

'In'의 값	동작
TRUE	마스터 컨트롤에 의한 리셋은 실행되지 않습니다. FOR 루프는 실행됩니다.
FALSE	마스터 컨트롤에 의한 리셋이 실행됩니다. FOR 루프는 실행되지만, MC-MCR의 안쪽은 실행되지 않습니다.

● 'In'=TRUE인 경우



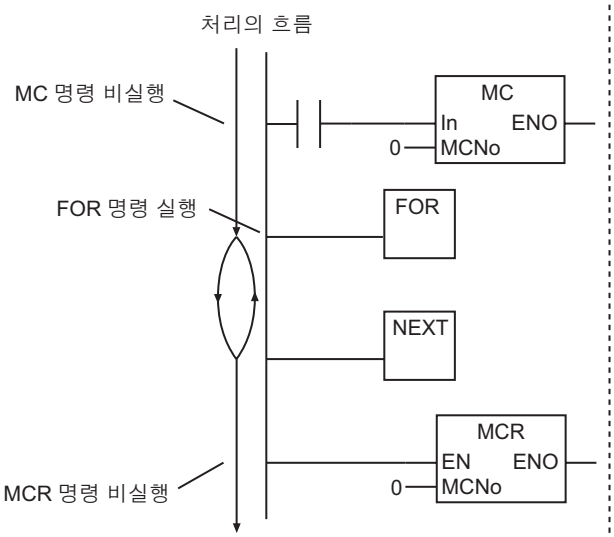
● 'In'=FALSE인 경우



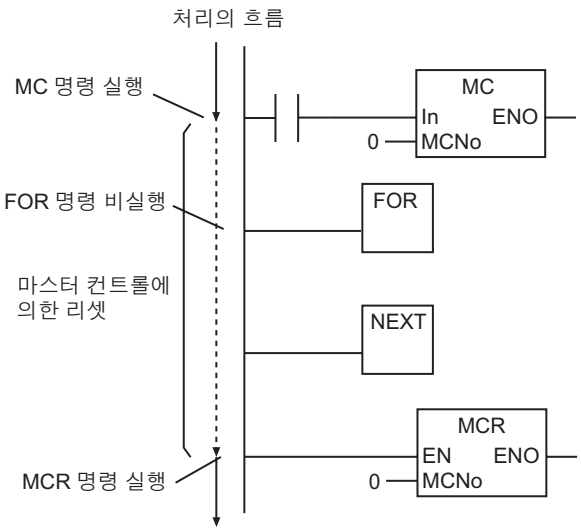
  - 아래 그림은 MC-MCR의 안쪽에 FOR-NEXT가 있습니다. 따라서 다음 표와 같이 동작합니다.

'In'의 값	동작
TRUE	마스터 컨트롤에 의한 리셋은 실행되지 않습니다. FOR 루프는 실행됩니다.
FALSE	마스터 컨트롤에 의한 리셋이 실행됩니다. FOR 루프는 실행되지 않습니다.

● 'In'=TRUE인 경우



● 'In'=FALSE인 경우



- FOR 명령, NEXT 명령, MC 명령, MCR 명령이 다음과 같이 서로 다르게 배치되어 있으면, 빌드 시에 이상 상태가 됩니다.
  FOR, MC, NEXT, MCR 순, 또는 MC, FOR, MCR, NEXT 순

JMP

지정한 점프대상으로 처리가 이동됩니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
JMP	점프	FUN	Label	없음

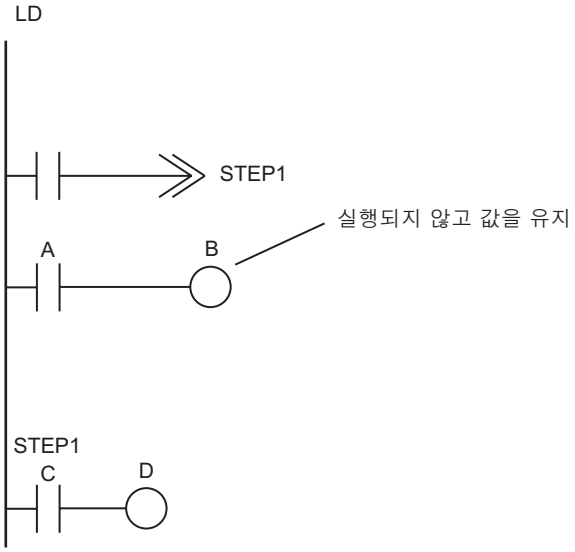
변수

없음

기능

입력 조건이 TRUE인 경우, 래더도 프로그램상에 Label로 지정한 점프대상으로 처리를 이동합니다. Label은 임의의 문자열입니다.

기술 예가 아래에 나와 있습니다. 이 예에서는 'STEP1'이라는 문자열을 Label로 사용했습니다. JMP 명령이 실행되면 처리가 'STEP1'이라고 적힌 위치로 이동합니다. 이 경우, JMP 명령과 Label 사이에 있는 Out 명령은 실행되지 않으며, 변수 B는 값을 유지합니다.



참고

- 섹션의 위 방향으로 점프할 수도 있습니다.
- 복수의 JMP 명령이 동일한 Label을 점프대상으로 지정할 수도 있습니다.

사용상의 주의 사항

- Label은 생략할 수 없습니다. 생략하면 빌드 시에 이상 상태가 됩니다.
- JMP 명령과 Label은 같은 섹션 내에 설정하십시오.

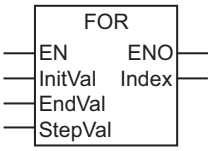
- 같은 섹션 내에 동일한 Label을 복수 설정할 수는 없습니다.
- FOR-NEXT 루프 밖에서 안으로 점프할 수는 없습니다.
- Label로 사용 가능한 문자의 제한은 다음과 같습니다.

항목	사양
최대 바이트 수	127바이트 ASCII 코드로 환산하면 127문자 일본어 문자(반각 가나 포함)로 환산하면 31문자
문자 코드	UTF-8
사용 가능한 문자	CASE 인센시티브 영숫자, 각국 언어 문자 기호: _(밑줄), ~(물결 문자)
조합해서 사용할 수 없는 문자	• ASCII 코드의 숫자 0~9(문자 코드 16#30~16#39)로 시작되는 문자열 • ASCII 코드의 '_'(밑줄) 1문자만으로 이루어진 문자열 • ASCII 코드의 '_'(밑줄)을 2개 이상 연속으로 포함하는 문자열 • 첫 번째 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 마지막 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 첫 2개의 문자가 'P_'인 문자열
금칙 문자	공백, !(느낌표), "(큰따옴표), #, $, &, '(작은따옴표), (,), *, +, ','(쉼표), -, '.'(마침표), /, :(쌍점), ;(쌍반점), <, =, >, ?, @, [,], ^, '`'(닫힌 작은따옴표), %

- 변수명을 Label로 지정할 수는 없습니다.

FOR / NEXT

FOR : 반복 처리의 시작 위치를 표시하고 반복 조건을 지정합니다.
NEXT : 반복 처리의 종료 위치를 표시합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
FOR	반복 시작	FUN		FOR Index:=InitVal TO EndVal BY StepVal DO 식; END_FOR*;
NEXT	반복 종료	FUN		*ST 언어에서는 반복 처리 종료 위치를 표시하는 데 'NEXT'를 사용하지 않고 'END_FOR'를 사용합니다.

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InitVal	초기값	입력	반복 시작 시에 'Index'에 세트되는 값	데이터 형식에 따름(*1)	–	(*2)
EndVal	최종값		반복 종료 시의 'Index'의 값			
StepVal	증감값		처리가 반복될 때마다 'Index'에 가산되는 값	데이터 형식에 따름(*3)		(*4)
Index	제어 변수	출력	루프 인덱스	데이터 형식에 따름	–	–

*1 래더도 프로그램의 경우 'InitVal'＜'EndVal'이 되도록 설정하십시오.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*3 래더도 프로그램의 경우 0 및 음수는 포함되지 않습니다. ST 프로그램의 경우 0은 포함되지 않습니다.
*4 래더도 프로그램의 경우, 입력 파라메터가 생략되어도 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
ST 프로그램의 경우, 입력 파라메터가 생략되면 초기값으로 1이 적용됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InitVal						○	○	○	○	○	○	○	○							
	열거 형식, 배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능(*)																			
EndVal						○	○	○	○	○	○	○	○							
	열거 형식, 배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능(*)																			
StepVal						○	○	○	○	○	○	○	○							
	열거 형식, 배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능(*)																			
Index						○	○	○	○	○	○	○	○							

* 래더도 언어인 경우, 열거 형식은 지정할 수 없습니다.

기능

FOR에서 NEXT까지(ST 언어의 경우는 FOR에서 END_FOR까지)의 처리를 반복해서 수행합니다.
FOR/NEXT의 처리 순서는 다음과 같습니다.

1 **제어 변수 'Index'에 'InitVal'의 값을 설정합니다.**

2 **'StepVal', 'InitVal', 'EndVal'의 값이 다음 표의 조건을 충족하는지 여부를 판단합니다. 조건을 충족하면 3으로 진행합니다. 충족하지 않으면 반복 처리를 수행하지 않고 NEXT 명령(ST 언어의 경우는 END_FOR)의 다음 처리로 이동합니다.**

프로그램 언어	반복 처리 계속 조건
래더도	'StepVal'≧0이고 'Index'<'EndVal'
ST	'StepVal'≧0이고 'Index'≦'EndVal'
	'StepVal'＜0이고 'Index'≧'EndVal'

3 **'Index', 'EndVal'의 값이 다음 표의 조건을 충족하는지 여부를 판단합니다. 조건을 충족하면 4로 진행합니다. 충족하지 않으면 반복 처리를 종료하고 NEXT 명령(ST 언어의 경우는 END_FOR)의 다음 처리로 이동합니다.**

프로그램 언어	반복 처리 계속 조건
래더도	'Index'≦'EndVal'
ST	'StepVal'≧0인 경우 'Index'≦'EndVal'
	'StepVal'＜0인 경우 'Index'≧'EndVal'

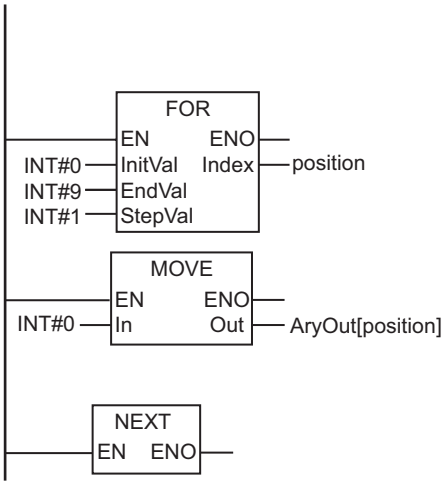
4 **FOR 명령과 NEXT 명령(ST 언어의 경우 END_FOR) 사이에 있는 처리를 1회 실행합니다.**

5 **'Index'에 'StepVal'의 값을 가산합니다.**

6 **3으로 되돌아갑니다.**

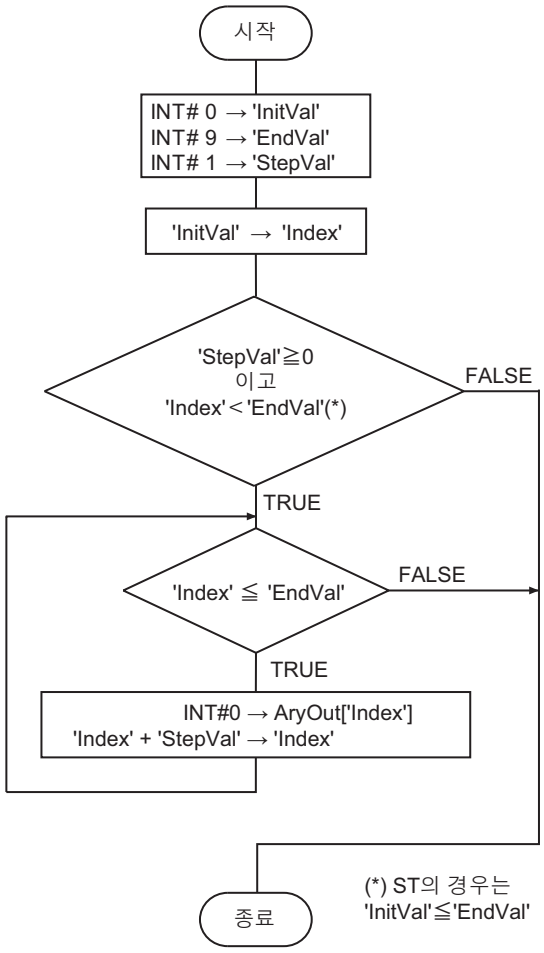
'InitVal'=INT#0, 'EndVal'=INT#9, 'StepVal'=INT#1인 경우의 예가 아래에 나와 있습니다. MOVE 명령이 10회 실행되며, 배열 변수 AryOut [0]~[9]에 값 INT#0이 대입됩니다.

LD



ST

```
FOR position:=INT#0 TO INT#9 BY INT#1 DO
    AryOut[position]:=INT#0;
END_FOR;
```



AryOut[0]부터 AryOut[9]에
순서대로 INT#0이 대입됩니다.

AryOut[0]	INT#0
AryOut[1]	INT#0
AryOut[2]	INT#0
AryOut[3]	INT#0
AryOut[4]	INT#0
AryOut[5]	INT#0
AryOut[6]	INT#0
AryOut[7]	INT#0
AryOut[8]	INT#0
AryOut[9]	INT#0

입력 변수에 연산식이나 펑션을 사용한 ST 프로그램의 기술 예

이 명령을 ST 프로그램에서 사용하는 경우, 'InitVal', 'EndVal', 'StepVal'의 입력 변수에는 다음을 기술할 수 있습니다.

- 연산 결과가 정수값이 되는 연산식
- 정수값을 반환하는 펑션
- 열거자를 반환하는 펑션

다음은 'EndVal'을 펑션, 'StepVal'을 연산식으로 기술한 예입니다.

```
A := DINT#1;
B := DINT#2;
C := REAL#9.6;
FOR i := 0 TO RoundUp(C) BY A+B DO
  DINTArray[i] := i;
END_FOR;
```

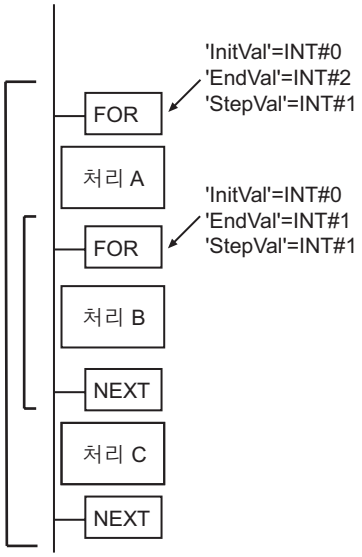
버전 관련 정보

'EndVal', 'StepVal'에 연산식을 기술할 수 있는 것은 Sysmac Studio Ver.1.08 이상입니다.
'InitVal'은 Sysmac Studio Ver.1.07 이하에서도 연산식을 기술할 수 있습니다.

참고

- 반복 처리를 중단하려면 BREAK 명령(ST 언어의 경우 EXIT)을 실행하십시오. 그 경우, BREAK 명령에서 NEXT 명령까지의 처리는 실행되지 않습니다.
- FOR에서 NEXT(ST 언어의 경우 FOR에서 END_FOR) 사이에 다시 FOR/NEXT 명령을 배치(중첩)할 수도 있습니다. 아래 그림의 경우, 처리 순서는 다음과 같습니다.

처리 A→처리 B→처리 B→처리 C→처리 A→처리 B→처리 B→처리 C→
처리 A→처리 B→처리 B→처리 C



사용상의 주의 사항

- 이 명령을 래더도 프로그램에서 사용하는 경우, FOR 명령과 NEXT 명령은 모두 반드시 좌측 모선에 직접 연결하십시오.
- FOR 명령과 NEXT 명령(ST 언어의 경우 FOR와 END_FOR)은 반드시 짝을 지어 사용하십시오. 양쪽 명령의 수가 같지 않으면 프로그램 이상이 발생합니다.
- 짝을 이루는 FOR 명령과 NEXT 명령은 같은 섹션 내에 설정하십시오.
- 처리의 반복이 무한히 수행(무한 루프)되지 않도록, 충분한 주의를 기울여 반복 종료 조건을 설정하십시오. 무한 루프가 되면 태스크 실행 타임아웃이 발생합니다.

  (예) 다음 표에서 각 변수의 입력 파라메터 값을 보면, SINT 형식의 최대값은 127이므로 'Index'의 값이 'EndVal'의 값보다 커지지는 않습니다. 따라서 무한 루프가 됩니다.
  'EndVal'에는 데이터 형식의 최대값을 설정하지 않도록 하십시오.

변수	입력 파라메터 값
'InitVal'	SINT#0
'EndVal'	SINT#127
'StepVal'	SINT#1
'Index'	–

- 'StepVal', 'InitVal', 'EndVal'의 값에 따라 다음과 같이 동작합니다.

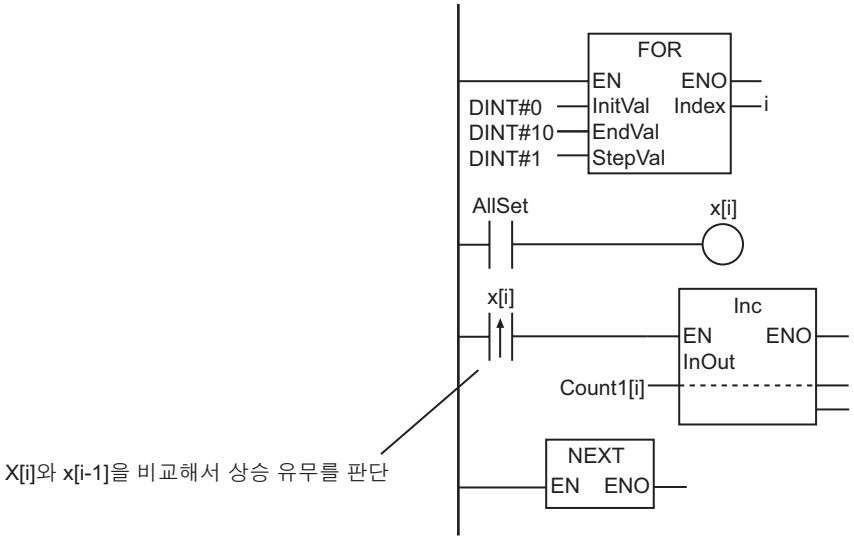
프로그램 언어	'StepVal'의 값	'InitVal', 'EndVal'의 값	동작
래더도	'StepVal'>0	'InitVal'＜'EndVal'	정상적으로 동작합니다.
		'InitVal'≧'EndVal'	FOR 명령과 NEXT 명령 사이의 처리가 한 번도 실행되지 않습니다. 이상은 발생하지 않습니다.
	'StepVal'<0	'InitVal'＜'EndVal'	FOR 명령과 NEXT 명령 사이의 처리가 일정하지 않은 횟수로 실행됩니다. 이렇게 설정하지 마십시오. 이상은 발생하지 않습니다.
		'InitVal'≧'EndVal'	FOR 명령과 NEXT 명령 사이의 처리가 한 번도 실행되지 않습니다. 이상은 발생하지 않습니다.
	'StepVal'=0	'InitVal'＜'EndVal'	무한 루프가 되어 태스크 실행 타임아웃이 발생합니다.
		'InitVal'≧'EndVal'	FOR 명령과 NEXT 명령 사이의 처리가 한 번도 실행되지 않습니다. 이상은 발생하지 않습니다.
ST	'StepVal'>0	'InitVal'≦'EndVal'	정상적으로 동작합니다.
		'InitVal'＞'EndVal'	FOR 명령과 END_FOR 명령 사이의 처리가 한번도 실행되지 않습니다. 이상은 발생하지 않습니다.
	'StepVal'<0	'InitVal'＜'EndVal'	FOR 명령과 END_FOR 명령 사이의 처리가 한번도 실행되지 않습니다. 이상은 발생하지 않습니다.
		'InitVal'≧'EndVal'	정상적으로 동작합니다.
	'StepVal'=0	'InitVal'≦'EndVal'	무한 루프가 되어 태스크 실행 타임아웃이 발생합니다.
		'InitVal'＞'EndVal'	FOR 명령과 END_FOR 명령 사이의 처리가 한번도 실행되지 않습니다. 이상은 발생하지 않습니다.

- 중첩(FOR/NEXT 명령의 계층 구조)의 깊이는 IF 명령, CASE 명령, FOR 명령, WHILE 명령, REPEAT 명령을 포함하여 15계층까지입니다.
- 중첩된 경우, 반복 처리를 중단하려면 계층의 수만큼 BREAK 명령(ST 언어의 경우 EXIT)이 필요합니다.
- 반복 처리를 중단하기 위해 점프 계통 명령(JMP 명령 등)을 사용하지 마십시오. 반복 처리를 중단하려면 반드시 BREAK 명령(ST 언어의 경우 EXIT)을 사용해야 합니다.
- 반복 처리 중에 'InitVal', 'EndVal', 'StepVal'의 값을 변경한 경우의 동작이 래더도 언어와 ST 언어에서 서로 다릅니다.

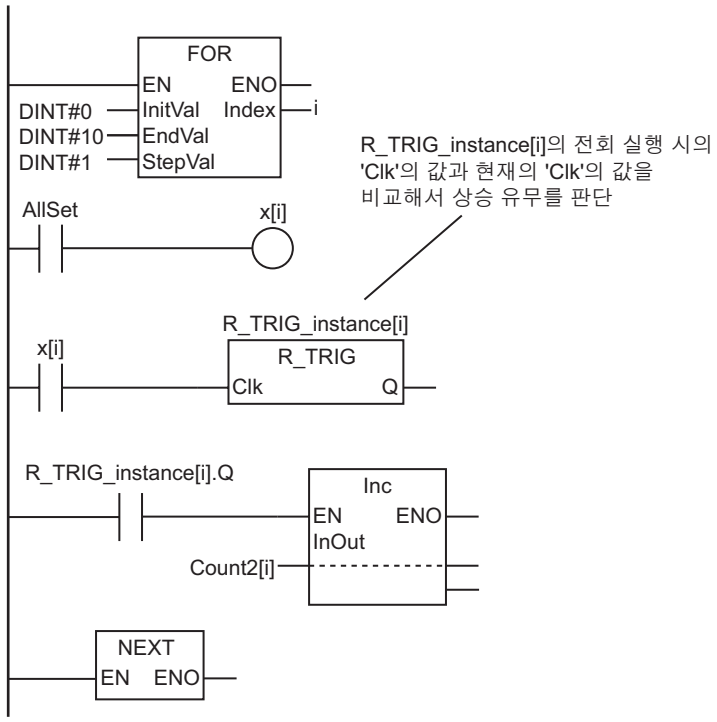
변수	동작	
	래더도 언어	ST 언어
'InitVal'	반복 처리가 종료될 때까지 변경 후의 값이 반영되지 않습니다.	반복 처리가 종료될 때까지 변경 후의 값이 반영되지 않습니다.
'EndVal'	반복 처리 도중에도 변경 후의 값이 반영됩니다.	
'StepVal'	의도한 대로 동작하지 않는 경우가 있습니다. 반복 처리 중에는 값을 변경하지 마십시오.	

- 래더도 언어인 경우 'InitVal', 'EndVal', 'StepVal', 'Index'는 모두 같은 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 'Index'의 데이터 형식은 'InitVal', 'EndVal', 'StepVal'의 유효 범위를 포함하도록 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.

- 반복 처리 종료 후의 'Index'의 값은, 래더도 언어와 ST 언어에서 서로 다릅니다. 래더도 언어는, 마지막에 'Index'에 'Stepval'을 더하지 않고 종료합니다. ST 언어는, 마지막에 'Index'에 'Stepval'을 더하고 종료합니다. 반복 횟수에는 차이가 없습니다.
  (예) 'InitVal'=1, 'EndVal'=100, 'StepVal=1인 경우
  래더도 언어 : 반복 횟수는 100회, 반복 종료 후의 'Index의 값은 100
  ST 언어 : 반복 횟수는 100회, 반복 종료 후의 'Index의 값은 101
- LD 프로그램의 FOR 루프 중에서 LD 명령(또는 AND 명령, OR 명령)을 상승 미분 지정(또는 하강 미분 지정) 있음으로 배열과 함께 사용하는 경우에는 주의가 필요합니다.
  상승 미분 지정은, 그 명령의 이전 실행 시의 지정 변수값과 이번 실행 시의 지정 변수값을 비교해서 상승의 유무를 판단합니다. 통상적으로 명령을 실행할 때마다 지정 변수가 변하지는 않지만, FOR 루프 중에서 배열을 지정한 경우에는 명령을 실행할 때마다 배열의 요소가 바뀝니다. 따라서 배열 요소의 값을 서로 비교해서 상승의 유무를 판단하게 됩니다.
  아래 사용자 프로그램의 경우, FOR 루프의 3주째(x=2)의 LD 명령은 현재의 x[2]의 값과 이전 LD 명령 실행 시의 지정 변수 x[1]의 값을 비교해서 상승 없음으로 판단됩니다. 그 결과, Count1[2]의 증가는 실행되지 않습니다.

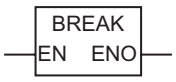


한편, 아래 사용자 프로그램의 경우에는 x[i]의 상승 유무를 R_TRIG 명령으로 감지합니다. x[i]의 각 요소에 대응하는 R_TRIG 인스턴스가 마련되어 있으므로, x[i]의 요소별 값의 변화를 감지할 수 있습니다. 그 결과, Count2[0]~Count2[10]은 모두 증가합니다.



BREAK

가장 안쪽의 FOR 명령에서 NEXT 명령까지의 반복 처리를 중단합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
BREAK	루프 중단	FUN		FOR Index:=0 TO 9 BY 1 DO IF Error[Index] THEN EXIT*; END_IF END_FOR; *ST 언어에서는 루프 중단 명령으로 "BREAK"가 아닌 "EXIT"를 사용합니다. 의미는 같습니다.

변수

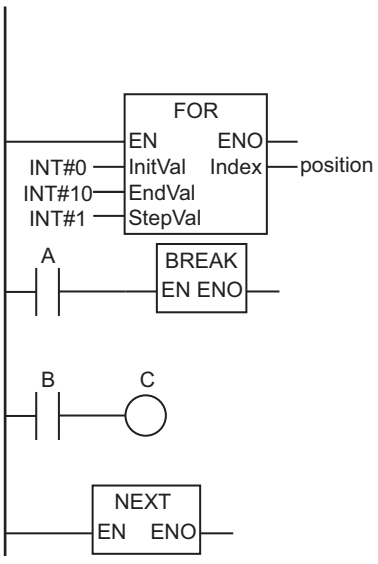
없음

기능

입력 조건이 TRUE인 경우, 가장 안쪽의 FOR 명령에서 NEXT 명령(ST 언어의 경우 END_FOR)까지의 반복 처리를 강제 종료하고, NEXT 명령 다음의 처리로 이동합니다.
BREAK 명령(ST 언어의 경우 EXIT)에서 NEXT 명령까지의 처리는 실행되지 않습니다.

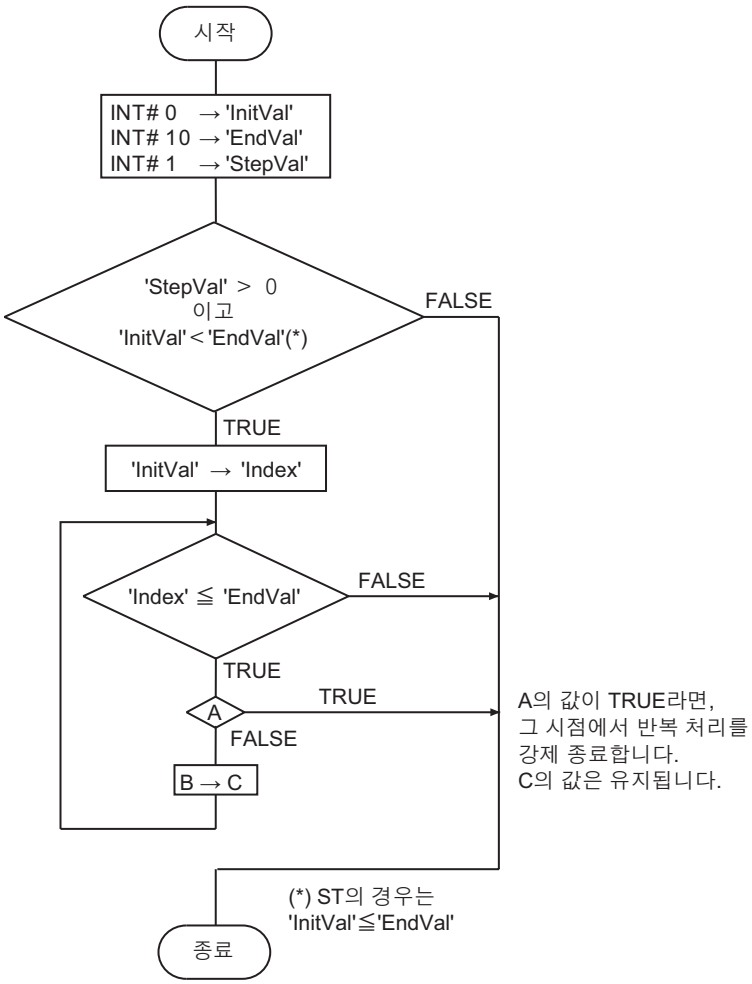
기술 예가 아래에 나와 있습니다. FOR 루프 실행 중에 매번 변수 A의 값을 확인합니다. 변수 A의 값이 TRUE라면, 그 시점에서 반복 처리를 강제 종료합니다.
그 경우, BREAK 명령(ST 언어의 경우 EXIT) 뒤에 있는 Out 명령은 실행되지 않으며 변수 C는 값을 유지합니다.

LD



ST

```
FOR position:=INT#0 TO INT#10 BY INT#1 DO
    IF (A=TRUE) THEN
        EXIT;
    END_IF;
    C:=B;
END_FOR;
```



사용상의 주의 사항

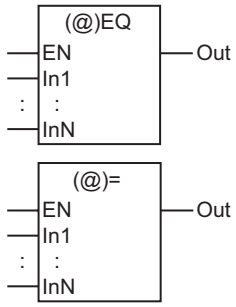
- 이 명령은 반드시 FOR 명령과 NEXT 명령(ST 언어의 경우 FOR와 END_FOR) 사이에 사용하십시오.
- FOR 명령과 NEXT 명령(ST 언어의 경우 FOR와 END_FOR)을 계층적으로 사용하는(중첩) 경우, 반복 처리를 완전히 중단하려면 계층의 수만큼 BREAK 명령(ST 언어의 경우 EXIT)이 필요합니다.
- 반복 처리를 중단하기 위해 점프 계통 명령(JMP 명령 등)을 사용하지 마십시오. 반복 처리를 중단하려면 반드시 BREAK 명령(ST 언어의 경우 EXIT)을 사용해야 합니다.

비교 명령

명령	명칭	페이지
EQ, =	비교 EQ	2-94
NE,<>	비교 NE	2-96
LT, </LE, <=/GT, >/GE, >=	비교 LT/비교 LE/비교 GT/비교 GE	2-98
EQascii	문자열 비교 EQ	2-101
NEascii	문자열 비교 NE	2-103
LTascii/LEascii/ GTascii/GEascii	문자열 비교 LT/ 문자열 비교 LE/ 문자열 비교 GT/ 문자열 비교 GE	2-105
Cmp	비교	2-108
ZoneCmp	영역 비교	2-110
TableCmp	테이블 비교	2-112
AryCmpEQ/AryCmpNE	배열 일괄 비교 EQ/ 배열 일괄 비교 NE	2-115
AryCmpLT/AryCmpLE/ AryCmpGT/AryCmpGE	배열 일괄 비교 LT/ 배열 일괄 비교 LE/ 배열 일괄 비교 GT/ 배열 일괄 비교 GE	2-117
AryCmpEQV/ AryCmpNEV	배열 요소 비교 EQ/ 배열 요소 비교 NE	2-119
AryCmpLTV/ AryCmpLEV/ AryCmpGTV/ AryCmpGEV	배열 요소 비교 LT/ 배열 요소 비교 LE/ 배열 요소 비교 GT/ 배열 요소 비교 GE	2-121

EQ, =

복수의 데이터가 모두 동일한지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EQ, =	비교 EQ	FUN		Out: =(In1=In2) & (In2=In3) & • • • & (InN-1=InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	비교 대상	입력	비교하는 수치 N은 2~5	데이터 형식에 따름	–	0*1
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

*1 InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

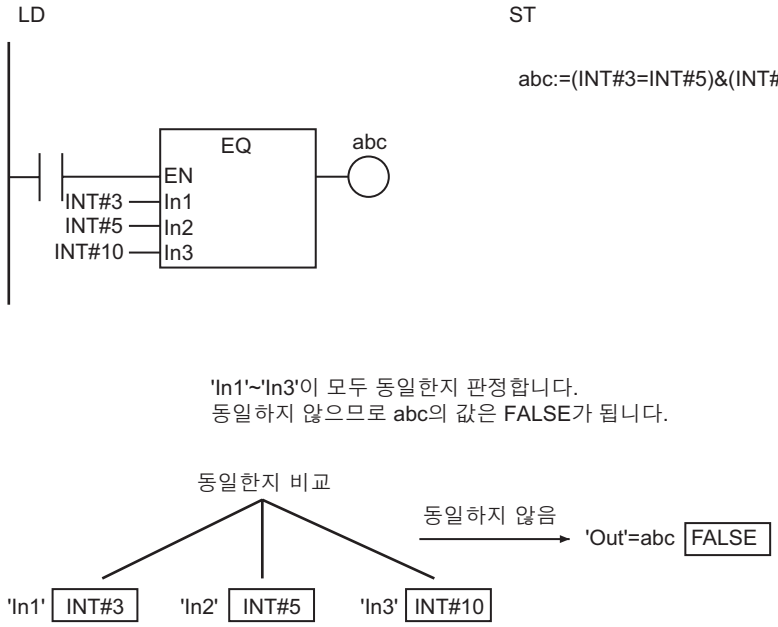
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능															*1				
Out	○																			

*1 TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다. SysmacStudio Ver.1.01 이하에서 작성한 프로젝트를 Sysmac Studio Ver.1.02 이상에서 열고 이들 데이터 형식을 사용하는 경우는 표시를 갱신하십시오. 표시를 갱신하려면 편집 창의 명령을 마우스 오른쪽 버튼으로 클릭하여 [업데이트]를 선택하십시오. 표시를 갱신하지 않으면 빌드 시에 이상 상태가 됩니다.

기능

2개 이상 5개 이하의 데이터 'In1'~'InN'이 모두 같은지 판정합니다.
모두 같은 경우에만, 비교 결과 'Out'의 값이 TRUE가 됩니다. 그 이외에는 'Out'의 값이 FALSE가 됩니다.
STRING 형식을 비교하는 경우, 동일하다는 것은 문자열의 길이와 내용이 모두 동일하다는 의미입니다.

'In1'=INT#3, 'In2'=INT#5, 'In3'＝INT#10인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 FALSE가 됩니다.



참고

- EQ와 =은 완전히 동일한 기능의 명령입니다. 사용하기 편한 쪽을 사용하시기 바랍니다.
- TIME 형식, DT 형식, TOD 형식을 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'In1'~'InN'의 데이터 형식이 다른 경우, 모든 데이터 형식의 유효 범위를 포함하는 데이터 형식으로 형식을 확장한 후 비교를 수행합니다.
- 비트 열 형식(BYTE, WORD, DWORD, LWORD)과 정수 형식(SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다. 또한, 비트 열 형식과 실수 형식(REAL, LREAL)의 비교도 할 수 없습니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 동일한 데이터 형식끼리만 비교할 수 있습니다. 서로 다른 데이터 형식을 지정하면 빌드 시에 이상 상태가 됩니다.
- 열거 형식과 열거 형식 이외의 형식은 비교할 수 없습니다. 또한 열거 형식끼리도 형식이 일치하지 않으면 비교할 수 없습니다.
- 'In1'~'InN'이 실수이고 나누어 떨어지지 않는 제산 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- +∞끼리, 또는 -∞끼리는 서로 같은 것으로 판단합니다.
- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 'Out'의 값은 FALSE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

NE,<>

2개의 데이터가 상이한지 판단합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NE,<>	비교 NE	FUN	(@)NE EN In1 In2 Out (@)<> EN In1 In2 Out	Out: =(In1 <> In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1, In2	비교 대상	입력	비교하는 수치	데이터 형식에 따름	–	*1(*)
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

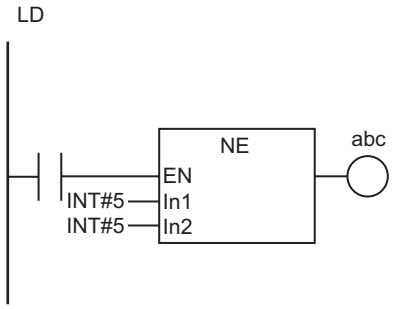
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1, In2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능															*1				
Out	○																			

*1. TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다. SysmacStudio Ver.1.01 이하에서 작성한 프로젝트를 Sysmac Studio Ver.1.02 이상에서 열고 이들 데이터 형식을 사용하는 경우는 표시를 갱신하십시오. 표시를 갱신하려면 편집 창의 명령을 마우스 오른쪽 버튼으로 클릭하여 [업데이트]를 선택하십시오. 표시를 갱신하지 않으면 빌드 시에 이상 상태가 됩니다.

기능

2개의 데이터 'In1'과 'In2'가 상이한지 판정합니다.
상이한 경우, 비교 결과 'Out'은 TRUE, 동일하다면 FALSE가 됩니다.
STRING 형식을 비교하는 경우, 동일하다는 것은 문자열의 길이와 내용이 모두 동일하다는 의미입니다.

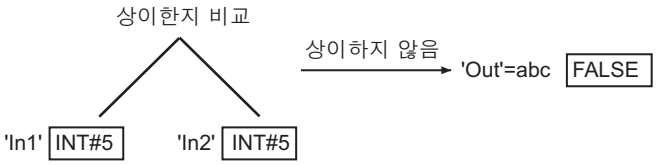
'In1'='In2'=INT#5인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 FALSE가 됩니다.



ST

```
abc:=(INT#5<>INT#5);
```

'In1'과 'In2'가 상이한지 여부를 판정합니다.
상이하지 않으므로 abc의 값은 FALSE가 됩니다.



참고

- NE 명령과<> 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.
- TIME 형식, DT 형식, TOD 형식을 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)' 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'In1'과 'InN'의 데이터 형식이 다른 경우, 두 데이터 형식의 유효 범위를 포함하는 데이터 형식으로 형식을 확장한 후 비교를 수행합니다.
- 비트 열 형식(BYTE, WORD, DWORD, LWORD)과 정수 형식(SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다. 또한, 비트 열 형식과 실수 형식(REAL, LREAL)의 비교도 할 수 없습니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 동일한 데이터 형식끼리만 비교할 수 있습니다. 서로 다른 데이터 형식을 지정하면 빌드 시에 이상 상태가 됩니다.
- 열거 형식과 열거 형식 이외의 형식은 비교할 수 없습니다. 또한 열거 형식끼리도 형식이 일치하지 않으면 비교할 수 없습니다.
- 'In1', 'InN'이 실수이고 나누어 떨어지지 않는 제산 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- +∞끼리, 또는 -∞끼리는 서로 같은 것으로 판단합니다.
- 'In1', 'InN' 중 어느 것의 값이 비수인 경우, 'Out'의 값은 TRUE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

LT, < / LE, <= / GT, > / GE, >=

복수의 수치의 크기 비교를 수행합니다.
LT,< : 크기 비교(<)를 수행합니다.
LE, <= : 크기 비교(<=)를 수행합니다.
GT,> : 크기 비교(>)를 수행합니다.
GE, >= : 크기 비교(>=)를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LT,<	비교 LT	FUN	(@)LT EN, In1, :, InN → Out (@)< EN, In1, :, InN → Out	Out: =(In1<In2) & (In2<In3) & • • • & (InN-1<InN);
LE, <=	비교 LE	FUN	(@)LE EN, In1, :, InN → Out (@)<= EN, In1, :, InN → Out	Out: =(In1<=In2) & (In2<=In3) &• • • & (InN-1<=InN);
GT,>	비교 GT	FUN	(@)GT EN, In1, :, InN → Out (@)> EN, In1, :, InN → Out	Out: =(In1>In2) & (In2>In3) & • • • & (InN-1>InN);
GE, >=	비교 GE	FUN	(@)GE EN, In1, :, InN → Out (@)>= EN, In1, :, InN → Out	Out: =(In1>=In2) & (In2>=In3) & • • • & (InN-1>=InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	비교 대상	입력	비교하는 수치 N은 2~5	데이터 형식에 따름	–	*1
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

*1 InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
		*1														*1				
Out	○																			

*1 BYTE 형식, WORD 형식, DWORD 형식, LWORD 형식, TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다. Sysmac Studio Ver.1.01 이하에서 작성한 프로젝트를 Sysmac Studio Ver.1.02 이상에서 열고 이들 데이터 형식을 사용하는 경우는 표시를 갱신하십시오. 표시를 갱신하려면 편집 창의 명령을 마우스 오른쪽 버튼으로 클릭하여 [업데이트]를 선택하십시오. 표시를 갱신하지 않으면 빌드 시에 이상 상태가 됩니다.

기능

2개 이상 5개 이하의 데이터 'In1'~'InN'의 크기를 비교합니다.

각 명령의 출력 'Out'의 값은 다음과 같습니다.

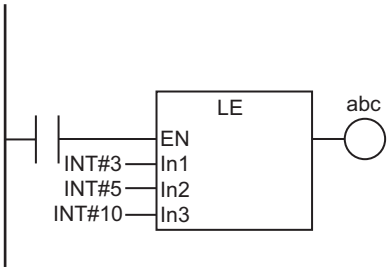
명령	'Out'의 값
LT, <	'In1' < 'In2' < · · < 'InN' 일 때 TRUE, 그 이외일 때 FALSE
LE, <=	'In1' <= 'In2' <= · · <= 'InN' 일 때 TRUE, 그 이외일 때 FALSE
GT, >	'In1' > 'In2' > · · > 'InN' 일 때 TRUE, 그 이외일 때 FALSE
GE, >=	'In1' >= 'In2' >= · · >= 'InN' 일 때 TRUE, 그 이외일 때 FALSE

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
BYTE, WORD, DWORD, LWORD	부호가 없는 정수로 판단합니다.
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.
STRING	🕮 'LTascii / LEascii / GTascii / GEascii 명령(2-105페이지)'과 동일한 사양입니다. 해당 부분을 참조하십시오.

LE 명령에서 'In1'=INT#3, 'In2'=INT#5, 'In3'＝INT#10인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 TRUE가 됩니다.

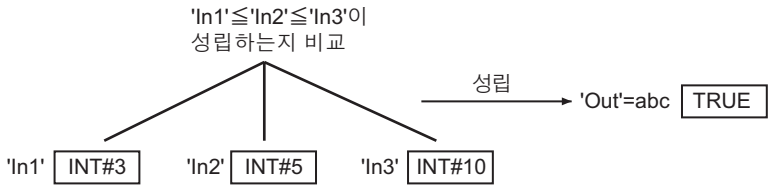
LD



ST

```
abc:=(INT#3<= INT#5)&(INT#5<=INT#10);
```

'In1'≦'In2'≦'In3'이 성립하는지 비교합니다.
성립하므로 abc의 값은 TRUE가 됩니다.



참고

- LT와<, LE와<=, GT와>, GE와>=은 각각 완전히 동일한 기능의 명령입니다. 사용하기 편한 쪽을 사용하시기 바랍니다.
- TIME 형식, DT 형식, TOD 형식의 크기를 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'In1'~'InN'의 데이터 형식이 다른 경우, 모든 데이터 형식의 유효 범위를 포함하는 데이터 형식으로 형식을 확장한 후 비교를 수행합니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- 비트 열 형식(BYTE, WORD, DWORD, LWORD)과 정수 형식(SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다. 또한 비트 열 형식과 실수 형식(REAL, LREAL)의 비교도 할 수 없습니다.
- TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 동일한 데이터 형식끼리만 비교할 수 있습니다. 서로 다른 데이터 형식을 지정하면 빌드 시에 이상 상태가 됩니다.
- 'In1'~'InN'이 실수이고 나누어 떨어지지 않는 제산 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- +∞끼리, 또는 -∞끼리는 서로 같은 것으로 판단합니다.
- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 'Out'의 값은 FALSE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

EQascii

복수의 문자열이 모두 동일한지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EQascii	문자열 비교 EQ	FUN	(@)EQascii EN In1 : InN Out	Out: =EQascii(In1, • • , InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	비교 대상 문자열	입력	비교하는 문자열 N은 2~5	데이터 형식에 따름	–	"(*)
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

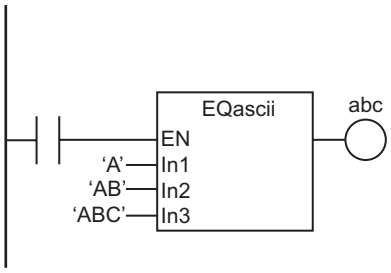
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN																				○
Out	○																			

기능

2개 이상 5개 이하의 문자열 'In1'~'InN'이 모두 같은지 판정합니다.
모두 같은 경우에만, 비교 결과 'Out'의 값이 TRUE가 됩니다. 그 이외에는 'Out'의 값이 FALSE가 됩니다.
동일하다는 것은, 길이와 내용이 모두 동일한 것을 가리킵니다.

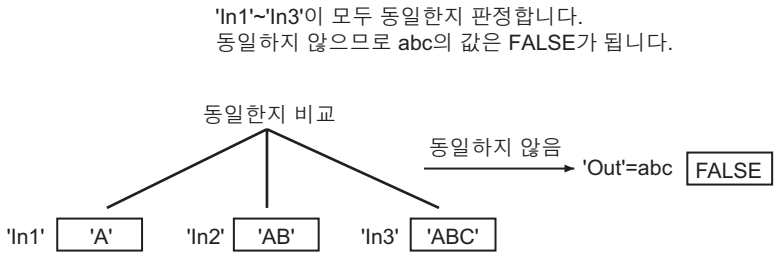
'In1'='A', 'In2'='AB', 'In3'＝'ABC'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 FALSE가 됩니다.

LD



ST

```
abc:=EQascii('A', 'AB', 'ABC');
```



참고

문자열 비교 명령은 복수의 문자열을 문자 코드 순으로 재배열하는 경우 등에 편리합니다. 예를 들면, 영문자는 문자 코드의 순서가 알파벳 순서와 같으므로, 알파벳 순으로 재배열할 수 있습니다.

버전 관련 정보

Sysmac Studio Ver.1.02 이상에서는 'EQ, = 명령(2-94페이지)'의 문자열 비교를 사용하는 방법도 있습니다. EQ,= 명령의 문자열 비교와 이 명령의 사양은 완전히 동일합니다.

사용상의 주의 사항

- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 'In1'~'InN'에는 ASCII 코드만으로 구성된 문자열을 지정하십시오.

NEascii

2개의 문자열이 상이한지 판단합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NEascii	문자열 비교 NE	FUN	(@)NEascii EN In1 In2 Out	Out: =NEascii(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1, In2	비교 대상 문자열	입력	비교하는 문자열	데이터 형식에 따름	–	(*)
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

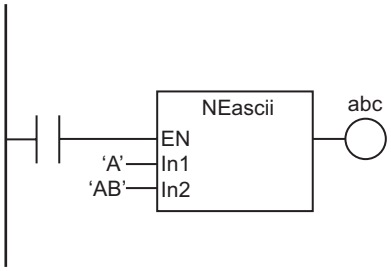
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1, In2																				○
Out	○																			

기능

2개의 문자열 'In1'과 'In2'가 상이한지 판정합니다.
상이한 경우, 비교 결과 'Out'은 TRUE, 동일하다면 FALSE가 됩니다.
동일하다는 것은, 길이와 내용이 모두 동일한 것을 가리킵니다.

'In1'='A', 'In2'='AB'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 TRUE가 됩니다.

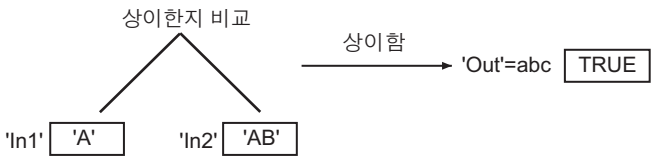
LD



ST

```
abc:=NEascii('A', 'AB');
```

'In1'과 'In2'가 상이한지 여부를 판정합니다.
상이하므로 abc의 값은 TRUE가 됩니다.



참고

문자열 비교 명령은 복수의 문자열을 문자 코드 순으로 재배열하는 경우 등에 편리합니다. 예를 들면, 영문자는 문자 코드의 순서가 알파벳 순서와 같으므로, 알파벳 순으로 재배열할 수 있습니다.

버전 관련 정보

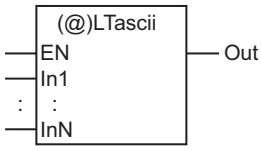
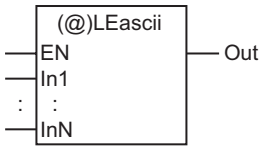
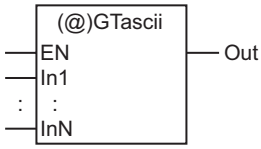
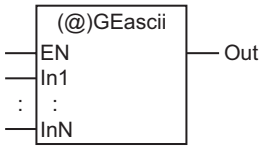
Sysmac Studio Ver.1.02 이상에서는 'NE,<> 명령(2-96페이지)'의 문자열 비교를 사용하는 방법도 있습니다. NE,<> 명령의 문자열 비교와 이 명령의 사양은 완전히 동일합니다.

사용상의 주의 사항

- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 'In1', 'In2'에는 ASCII 코드만으로 구성된 문자열을 지정하십시오.

LTascii / LEascii / GTascii / GEascii

복수의 문자열의 크기 비교를 수행합니다.
LTascii : 크기 비교(<)를 수행합니다.
LEascii : 크기 비교(<=)를 수행합니다.
GTascii : 크기 비교(>)를 수행합니다.
GEascii : 크기 비교(>=)를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LTascii	문자열 비교 LT	FUN		Out: =LTascii(In1, • • • , InN);
LEascii	문자열 비교 LE	FUN		Out: =LEascii(In1, • • • , InN);
GTascii	문자열 비교 GT	FUN		Out: =GTascii(In1, • • • , InN);
GEascii	문자열 비교 GE	FUN		Out: =GEascii(In1, • • • , InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	비교 대상 문자열	입력	비교하는 문자열 N은 2~5	데이터 형식에 따름	–	"(*)
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN																				○
Out	○																			

기능

2개 이상 5개 이하의 문자열 'In1'~'InN'의 크기를 비교합니다.
각 명령의 출력 'Out'의 값은 다음과 같습니다.

명령	'Out'의 값
LTascii	'In1' < 'In2' < · · < 'InN' 일 때 TRUE, 그 이외일 때 FALSE
LEascii	'In1' <= 'In2' <= · · <= 'InN' 일 때 TRUE, 그 이외일 때 FALSE
GTascii	'In1' > 'In2' > · · > 'InN' 일 때 TRUE, 그 이외일 때 FALSE
GEascii	'In1' >= 'In2' >= · · >= 'InN' 일 때 TRUE, 그 이외일 때 FALSE

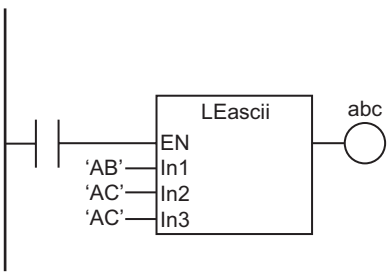
크기 비교는 문자 코드를 이용해 수행합니다. 비교 순서는 다음과 같습니다.
먼저 각 문자열의 1번째 문자의 문자 코드를 비교합니다. 문자 코드가 상이하면, 그 문자 코드의 크기가 문자열의 크기 관계가 됩니다.
만일 1번째 문자의 문자 코드가 동일한 경우, 상이한 문자 코드가 발견될 때까지 2번째 문자, 3번째 문자를 순서대로 비교해 나갑니다.
각 문자열의 길이가 다른 경우, 긴 쪽의 문자열에 맞춰서 짧은 쪽의 문자열 끝에 NULL 문자(16#00)를 넣어 비교를 수행합니다.

몇 개의 문자열의 크기 관계를 나타냅니다.
'AD' (16#414400) < 'BC'(16#424400)
'ADC' (16#41444300) < 'B'(16#42000000)
'ABC' (16#41424300) < 'ABD'(16#41424400)
'ABC' (16#41424300) > 'AB'(16#41420000)
'AB' (16#414200) = 'AB'(16#414200)

문자열 중에 다중 바이트 문자가 사용되는 경우, 그 문자를 1바이트씩 분할해서 해석합니다. 예를 들면 2바이트 문자 16#C281은, 16#C2와 16#81이라는 2개의 1바이트 문자가 있는 것으로 해석합니다.

LEascii 명령에서, 'In1'='AB', 'In2'='AC', 'In3'＝'AC'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 TRUE가 됩니다.

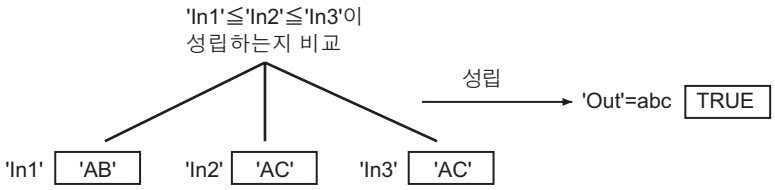
LD



ST

```
abc:=LEascii('AB', 'AC', 'AC');
```

'In1'≦'In2'≦'In3'이 성립하는지 비교합니다.
성립하므로 abc의 값은 TRUE가 됩니다.



참고

문자열 비교 명령은 복수의 문자열을 문자 코드 순으로 재배열하는 경우 등에 편리합니다. 예를 들면, 영문자는 문자 코드의 순서가 알파벳 순서와 같으므로, 알파벳 순으로 재배열할 수 있습니다.

버전 관련 정보

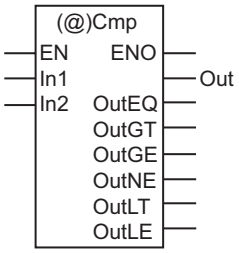
Sysmac Studio Ver.1.02 이상에서는 'LT, < / LE, <= / GT, > / GE, >= 명령(2-98페이지)'의 문자열 비교를 사용하는 방법도 있습니다. LT,< / LE,<= / GT,> / GE,>= 명령의 문자열 비교와 이 명령의 사양은 완전히 동일합니다.

사용상의 주의 사항

- 이 명령은 회로의 최종 단계에는 사용할 수 없습니다. 사용한 경우, Sysmac Studio에서 이상이 발생하여 컨트롤러로 전송할 수 없습니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 'In1'~'InN'에는 ASCII 코드만으로 구성된 문자열을 지정하십시오.

Cmp

2개의 수치를 비교합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Cmp	비교	FUN		Out: =Cmp(In1, In2, OutEQ, OutGT, OutGE, OutNE, OutLT, OutLE); Out은 생략 가능

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1, In2	비교 대상	입력	비교하는 수치	데이터 형식에 따름	–	(*)
Out	반환값	출력	항상 TRUE	TRUE만	–	–
OutEQ	=플래그		=플래그	데이터 형식에 따름		
OutGT	>플래그		>플래그			
OutGE	>=플래그		>=플래그			
OutNE	<>플래그		<>플래그			
OutLT	<플래그		<플래그			
OutLE	<=플래그		<=플래그			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1, In2						○	○	○	○	○	○	○	○	○	○					
Out	○																			
OutEQ	○																			
OutGT	○																			
OutGE	○																			
OutNE	○																			
OutLT	○																			
OutLE	○																			

기능

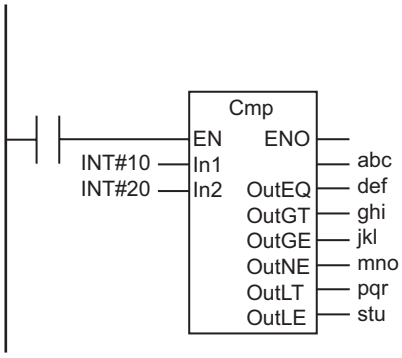
2개의 수치 'In1'과 'In2'를 비교해서 각 플래그의 값을 출력합니다.
각 플래그의 값은 다음과 같습니다.

플래그	값
'OutEQ'	'In1' = 'In2' 일 때 TRUE, 그 이외일 때 FALSE
'OutGT'	'In1' > 'In2' 일 때 TRUE, 그 이외일 때 FALSE
'OutGE'	'In1' >= 'In2' 일 때 TRUE, 그 이외일 때 FALSE

플래그	값
'OutNE'	'In1' <> 'In2' 일 때 TRUE, 그 이외일 때 FALSE
'OutLT'	'In1' < 'In2' 일 때 TRUE, 그 이외일 때 FALSE
'OutLE'	'In1' <= 'In2' 일 때 TRUE, 그 이외일 때 FALSE

'In1'=INT#10, 'In2'=INT#20인 경우의 예가 아래에 나와 있습니다. 변수 def, ghi, jkl의 값은 FALSE, abc, mno, pqr, stu의 값은 TRUE가 됩니다.

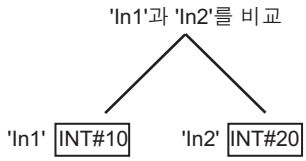
LD



ST

```
abc:=Cmp(INT#10, INT#20, def, ghi, jkl, mno, pqr, stu);
```

'In1'과 'In2'를 비교합니다.
각 판정 조건에 따라 결과는 다음과 같습니다.



'Out'	항상 TRUE
'OutEQ'	'In1'='In2'가 아니므로 FALSE
'OutGT'	'In1'>'In2'가 아니므로 FALSE
'OutGE'	'In1'≧'In2'가 아니므로 FALSE
'OutNE'	'In1'≠'In2'이므로 TRUE
'OutLT'	'In1'<'In2'이므로 TRUE
'OutLE'	'In1'≦'In2'이므로 TRUE

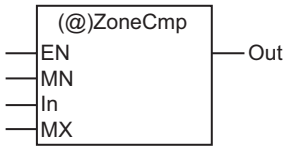
'Out' =abc	TRUE
'OutEQ'=def	FALSE
'OutGT'=ghi	FALSE
'OutGE'=jkl	FALSE
'OutNE'=mno	TRUE
'OutLT' =pqr	TRUE
'OutLE' =stu	TRUE

사용상의 주의 사항

- 'In1'과 'InN'의 데이터 형식이 다른 경우, 두 데이터 형식의 유효 범위를 포함하는 데이터 형식으로 형식을 확장한 후 비교를 수행합니다.
- 'In1', 'In2'가 실수이고 나누어 떨어지지 않는 제산 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- +∞끼리, 또는 -∞끼리는 서로 같은 것으로 판단합니다.
- 'In1', 'In2' 중 어느 것의 값이 비수인 경우, 'OutEQ', 'OutGT', 'OutGE', 'OutNE', 'OutLT', 'OutLE'의 값은 FALSE가 됩니다.

ZoneCmp

비교 데이터가 지정한 상한값과 하한값 사이에 있는지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ZoneCmp	영역 비교	FUN		Out: =ZoneCmp(MN, In, MX);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
MN	하한값	입력	하한값	데이터 형식에 따름	–	0
In	비교 데이터		비교하는 수치			(*)
MX	상한값		상한값			0
Out	비교 결과	출력	비교 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN						○	○	○	○	○	○	○	○	○	○	○	○	○	○	
																(*)				
In						○	○	○	○	○	○	○	○	○	○	○	○	○	○	
																(*)				
MX						○	○	○	○	○	○	○	○	○	○	○	○	○	○	
																(*)				
Out	○																			

* TIME 형식, DATE 형식, TOD 형식, DT 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

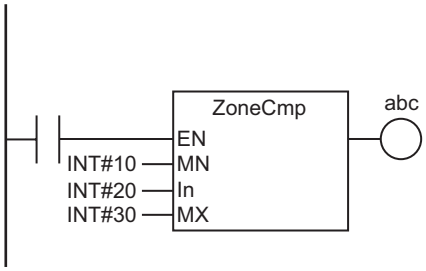
비교 데이터 'In'이 상한값 'MX'와 하한값 'MN' 사이에 있는지 판정합니다.
'MX' ≧ 'In' ≧ 'MN'일 때, 'Out'의 값은 TRUE, 그렇지 않으면 FALSE가 됩니다.

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.

'MN'=INT#10, 'In'=INT#20, 'MX'=INT#30인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 TRUE가 됩니다.

LD

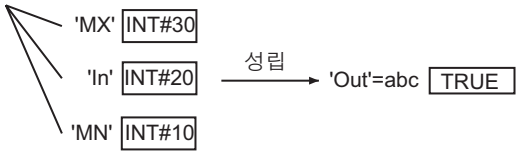


ST

```
abc:=ZoneCmp(INT#10, INT#20, INT#30);
```

'MX'≧'In'≧'MN'이 성립하는지 판정합니다.
성립하므로 abc의 값은 TRUE가 됩니다.

'In'이 'MX'와 'MN' 사이에 있는지 판정



참고

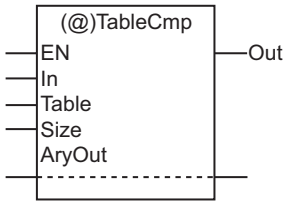
TIME 형식, DT 형식, TOD 형식의 크기를 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'In', 'MX', 'MN'의 데이터 형식이 다른 경우, 모든 데이터 형식의 유효 범위를 포함하는 데이터 형식으로 형식을 확장한 후 비교를 수행합니다.
- 'In', 'MX', 'MN'이 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- TIME 형식, DATE 형식, TOD 형식, DT 형식은 동일한 데이터 형식끼리만 비교할 수 있습니다. 서로 다른 데이터 형식을 지정하면 빌드 시에 이상 상태가 됩니다.
- +∞끼리, 또는 -∞끼리는 서로 같은 것으로 판단합니다.
- 'In'의 값이 비수인 경우, 'Out'의 값은 FALSE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 다음과 같은 경우 이상 상태가 되어 'Out'의 값이 FALSE가 됩니다.
  - 'MN'의 값이 'MX'의 값보다 큰 경우
  - 'MX', 'MN' 중 어느 것이 비수인 경우

TableCmp

비교 데이터와 비교 테이블에서 지정된 복수의 정의 구간을 비교합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TableCmp	테이블 비교	FUN		Out: =TableCmp(In, Table, Size, AryOut);

변수

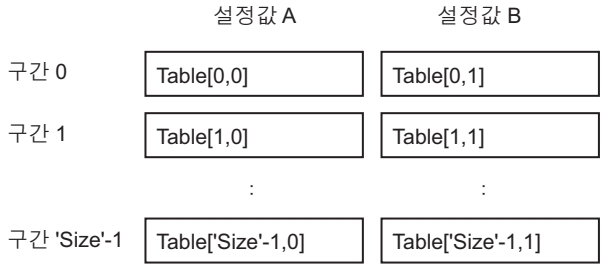
	명칭	입/출력	내용	유효 범위	단위	초기값
In	비교 데이터	입력	비교하는 수치	데이터 형식에 따름	–	(*)
Table[] 2차원 배열	비교 테이블		각 정의 구간을 요소로 하는 2차원 배열			
Size	비교 크기		'In'과 비교하는 Table[]의 요소 수			1
AryOut[] 배열	개별 비교 결과 배열	입출력	Table[]의 각 요소의 비교 결과 TRUE : 일치 FALSE : 불일치	데이터 형식에 따름	–	–
Out	비교 결과	출력	TRUE : Table[]의 모든 요소가 'In'과 일치 FALSE : 일치하지 않는 요소가 1개 이상 있음	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

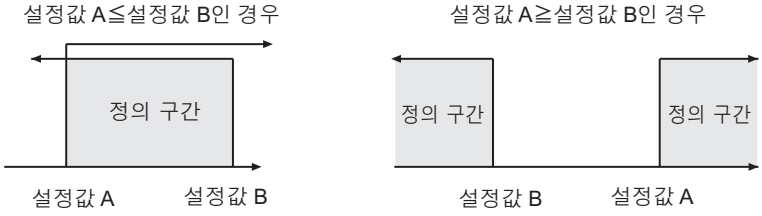
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○	○	○					
Table[] 2차원 배열	'In'과 동일한 데이터 형식의 요소를 갖는 2차원 배열																			
Size							○													
AryOut[] 배열	○																			
Out	○																			

기능

비교 데이터 'In'과 비교 테이블 Table[]로 지정된 'Size' 세트의 정의 구간을 비교합니다.
Table[]은 2차원 배열로서, 제1차원이 정의 구간의 번호, 제2차원의 0번째 요소가 정의 구간의 설정값 A, 1번째 요소가 정의 구간의 설정값 B를 나타냅니다.

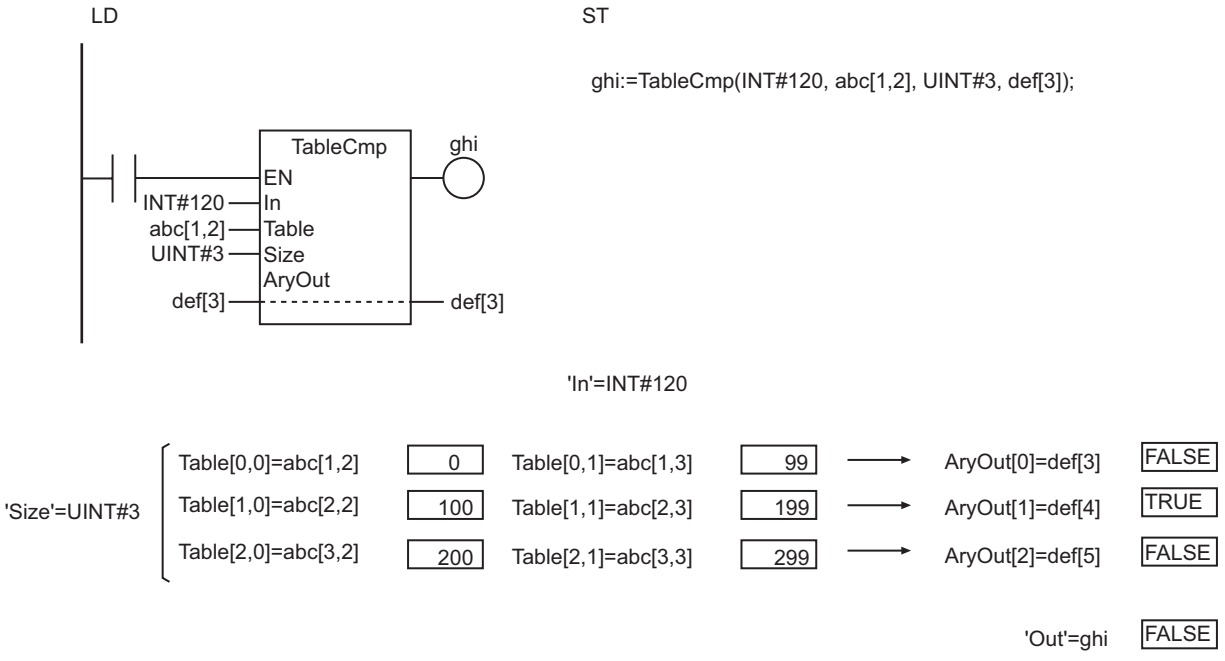


설정값 A와 설정값 B에 의한 정의 구간 지정 방법을 다음 그림에 나타냅니다. 설정값 A 및 설정값 B의 값 자체는 정의 구간 내에 포함됩니다.



'In'과 Table[]의 비교 결과는 개별 비교 결과 배열 AryOut[]에 저장됩니다. 'In'이 번째 정의 구간에 포함되어 있으면 AryOut[i]의 값은 TRUE, 포함되어 있지 않으면 AryOut[i]의 값은 FALSE가 됩니다. 그리고 AryOut[]의 'Size'개의 요소가 모두 TRUE라면, 비교 결과 'Out'의 값은 TRUE, 그렇지 않으면 FALSE가 됩니다.

'In'=INT#120, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

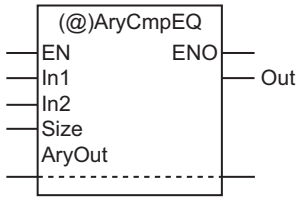
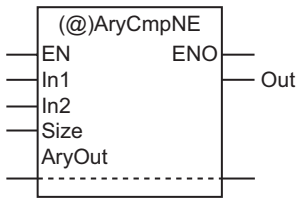


사용상의 주의 사항

- 'In'과 Table[]의 요소의 데이터 형식이 동일하게 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- Table[]은 반드시 2차원 배열로 하십시오.
- Table[]의 제2차원 배열의 크기가 3 이상인 경우, 제2차원의 2번째 이후의 요소는 무시됩니다.
- AryOut[]의 배열의 크기가 'Size' 이상인 경우, AryOut[0]~AryOut['Size'-1]에 비교 결과가 저장됩니다. 그 이외의 배열 요소는 변화하지 않습니다.
- 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)은 비교할 수 없습니다.
- 비교 대상이 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 FALSE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Out'은 FALSE가 됩니다.
  - 'Size'의 값이 AryOut[]의 배열의 크기를 초과하는 경우
  - 'Size'의 값이 Table[]의 제1차원 배열의 크기를 초과하는 경우
  - Table[]의 제2차원 배열의 크기가 1인 경우

AryCmpEQ / AryCmpNE

2개의 배열의 각 요소를 비교합니다.
AryCmpEQ : 동일한지 판정합니다.
AryCmpNE : 상이한지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCmpEQ	배열 일괄 비교 EQ	FUN		AryCmpEQ(In1, In2, Size, AryOut);
AryCmpNE	배열 일괄 비교 NE	FUN		AryCmpNE(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[], In2[] 배열	비교 배열	입력	비교하는 수치를 요소로 하는 배열	데이터 형식에 따름	–	(*)
Size	비교 요소 수		비교하는 요소 수			1
AryOut[] 배열	비교 결과 배열	입출력	비교 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

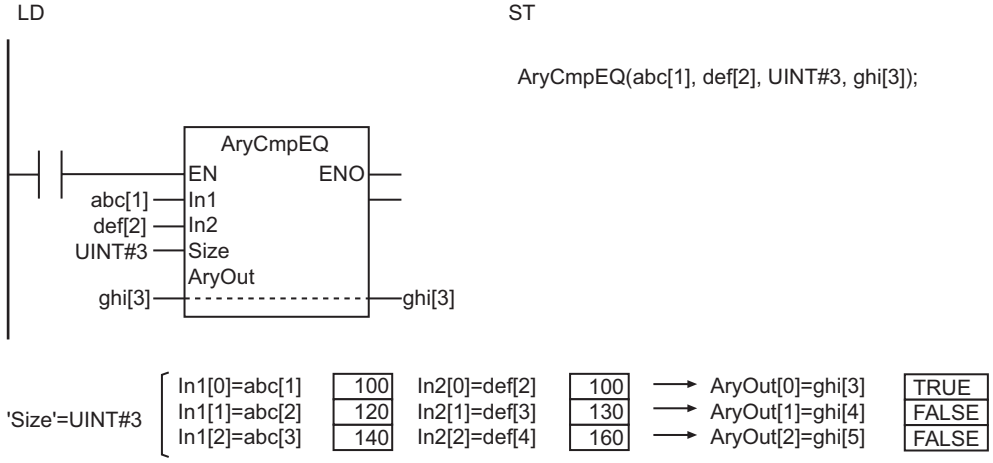
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					
In2[] 배열	In1[]과 동일한 데이터 형식의 배열																			
Size							○													
AryOut[] 배열	○																			
Out	○																			

기능

2개의 배열 In1[0]~In1['Size'-1], In2[0]~In2['Size'-1]의 동일한 요소 번호끼리 비교합니다. 비교 결과는, 비교 결과 배열 AryOut[0]~AryOut['Size'-1]의 대응하는 요소 번호에 저장됩니다.
각 명령의 AryOut[i]의 값은 다음과 같습니다.

명령	AryOut[i]의 값
AryCmpEQ	In1[i]=In2[i] 일 때 TRUE, 그 이외일 때 FALSE
AryCmpNE	In1[i]≠In2[i] 일 때 TRUE, 그 이외일 때 FALSE

AryCmpEQ 명령에서, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

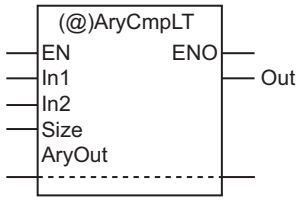
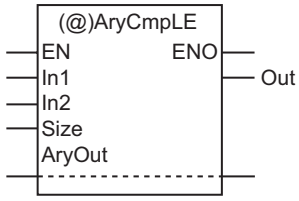
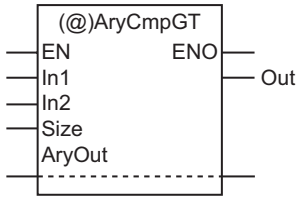
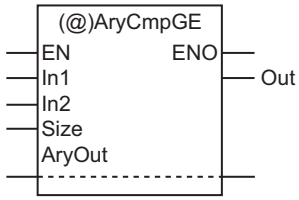


사용상의 주의 사항

- In1[]과 In2[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- AryOut[]의 배열 크기는 'Size' 이상으로 하십시오.
- In1[], In2[]가 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - In1[], In2[], AryOut[] 중 어느 것의 배열 크기가 'Size'보다 작은 경우

AryCmpLT / AryCmpLE / AryCmpGT / AryCmpGE

2개의 배열의 각 요소의 크기 비교를 수행합니다.
AryCmpLT : 크기 비교(<)를 수행합니다.
AryCmpLE : 크기 비교(<=)를 수행합니다.
AryCmpGT : 크기 비교(>)를 수행합니다.
AryCmpGE : 크기 비교(>=)를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCmpLT	배열 일괄 비교 LT	FUN		AryCmpLT(In1, In2, Size, AryOut);
AryCmpLE	배열 일괄 비교 LE	FUN		AryCmpLE(In1, In2, Size, AryOut);
AryCmpGT	배열 일괄 비교 GT	FUN		AryCmpGT(In1, In2, Size, AryOut);
AryCmpGE	배열 일괄 비교 GE	FUN		AryCmpGE(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[], In2[] 배열	비교 배열	입력	비교하는 수치를 요소로 하는 배열	데이터 형식에 따름	–	(*)
Size	비교 요소 수		비교하는 요소 수			1
AryOut[] 배열	비교 결과 배열	입출력	비교 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다

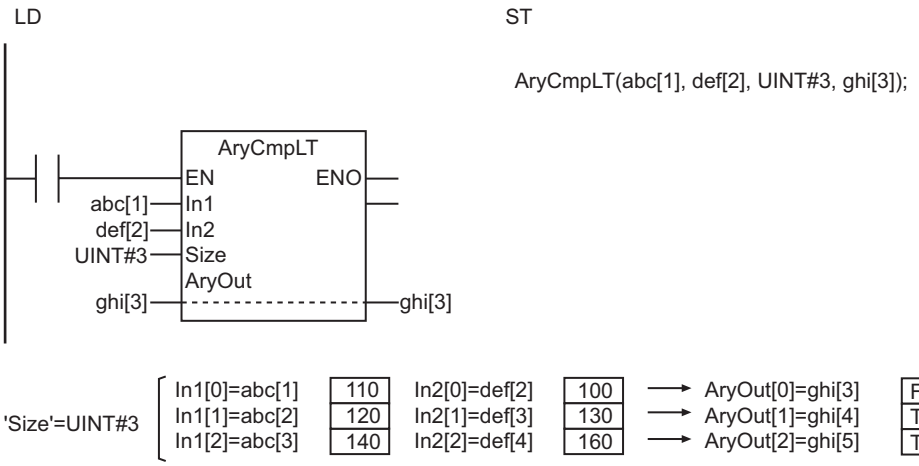
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2[] 배열	In1[]과 동일한 데이터 형식의 배열																			
Size							○													
AryOut[] 배열	○																			
Out	○																			

기능

2개의 배열 In1[0]~In1['Size'-1], In2[0]~In2['Size'-1]의 동일한 요소 번호끼리 크기를 비교합니다. 비교 결과는, 비교 결과 배열 AryOut[0]~AryOut['Size'-1]의 대응하는 요소 번호에 저장됩니다.
각 명령의 AryOut[i]의 값은 다음과 같습니다.

명령	AryOut[i]의 값
AryCmpLT	In1[i]<In2[i] 일 때 TRUE, 그 이외일 때 FALSE
AryCmpLE	In1[i]<=In2[i] 일 때 TRUE, 그 이외일 때 FALSE
AryCmpGT	In1[i]>In2[i] 일 때 TRUE, 그 이외일 때 FALSE
AryCmpGE	In1[i]>=In2[i] 일 때 TRUE, 그 이외일 때 FALSE

AryCmpLT 명령에서, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

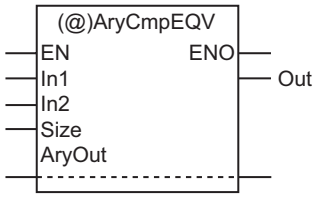
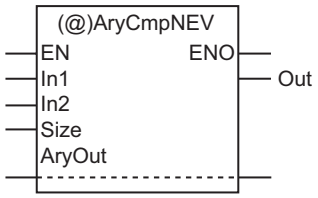


사용상의 주의 사항

- In1[]과 In2[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- AryOut[]의 배열 크기는 'Size' 이상으로 하십시오.
- In1[], In2[]가 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - In1[], In2[], AryOut[] 중 어느 것의 배열 크기가 'Size'보다 작은 경우

AryCmpEQV / AryCmpNEV

배열의 각 요소와 수치의 크기 비교를 수행합니다.
AryCmpEQV : 동일한지 판정합니다.
AryCmpNEV : 상이한지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCmpEQV	배열 요소 비교 EQ	FUN		AryCmpEQV(In1, In2, Size, AryOut);
AryCmpNEV	배열 요소 비교 NE	FUN		AryCmpNEV(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열	비교 배열	입력	비교하는 수치를 요소로 하는 배열	데이터 형식에 따름	–	(*)
In2	비교 수치		비교하는 수치			
Size	비교 요소 수		비교하는 요소 수			1
AryOut[] 배열	비교 결과 배열	입출력	비교 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○					
In2	In1[]의 요소와 동일한 데이터 형식																			
Size							○													
AryOut[] 배열	○																			
Out	○																			

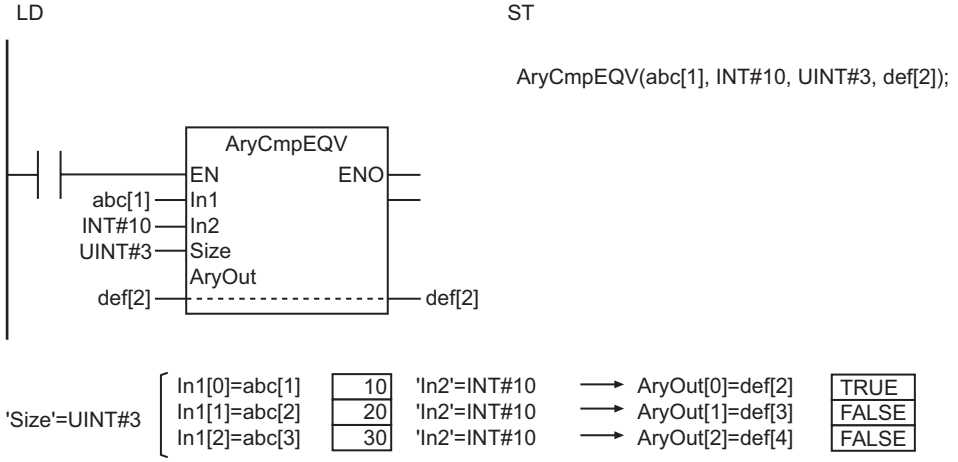
기능

비교 배열 In1[0]~In1['Size'-1]과 비교 수치 'In2'를 비교합니다.
비교 결과는, 비교 결과 배열 AryOut[0]~AryOut['Size'-1]의 대응하는 요소 번호에 저장됩니다.

각 명령의 AryOut[i]의 값은 다음과 같습니다.

명령	AryOut[i]의 값
AryCmpEQV	In1[i]='In2' 일 때 TRUE, 그 이외일 때 FALSE
AryCmpNEV	In1[i]≠'In2' 일 때 TRUE, 그 이외일 때 FALSE

AryCmpEQV 명령에서 'In2'=INT#10, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

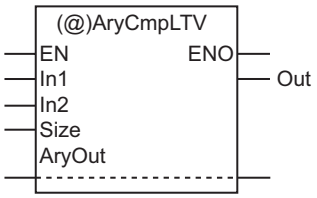
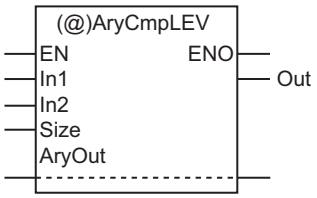
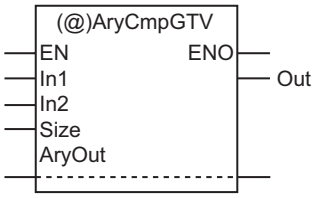
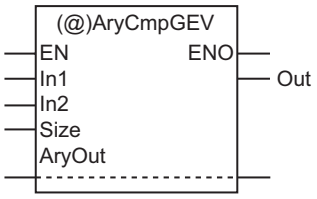


사용상의 주의 사항

- In1[]과 'In2'의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- AryOut[]의 배열 크기는 'Size' 이상으로 하십시오.
- In1[], 'In2'가 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - In1[], AryOut[] 중 어느 것의 배열 크기가 'Size'보다 작은 경우

AryCmpLTV / AryCmpLEV / AryCmpGTV / AryCmpGEV

배열의 각 요소와 수치의 크기 비교를 수행합니다.
AryCmpLTV : 크기 비교(<)를 수행합니다.
AryCmpLEV : 크기 비교(<=)를 수행합니다.
AryCmpGTV : 크기 비교(>)를 수행합니다.
AryCmpGEV : 크기 비교(>=)를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCmpLTV	배열 요소 비교 LT	FUN		AryCmpLTV(In1, In2, Size, AryOut);
AryCmpLEV	배열 요소 비교 LE	FUN		AryCmpLEV(In1, In2, Size, AryOut);
AryCmpGTV	배열 요소 비교 GT	FUN		AryCmpGTV(In1, In2, Size, AryOut);
AryCmpGEV	배열 요소 비교 GE	FUN		AryCmpGEV(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열	비교 배열	입력	비교하는 수치를 요소로 하는 배열	데이터 형식에 따름	–	(*)
In2	비교 수치		비교하는 수치			
Size	비교 요소 수		비교하는 요소 수			1
AryOut[] 배열	비교 결과 배열	입출력	비교 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

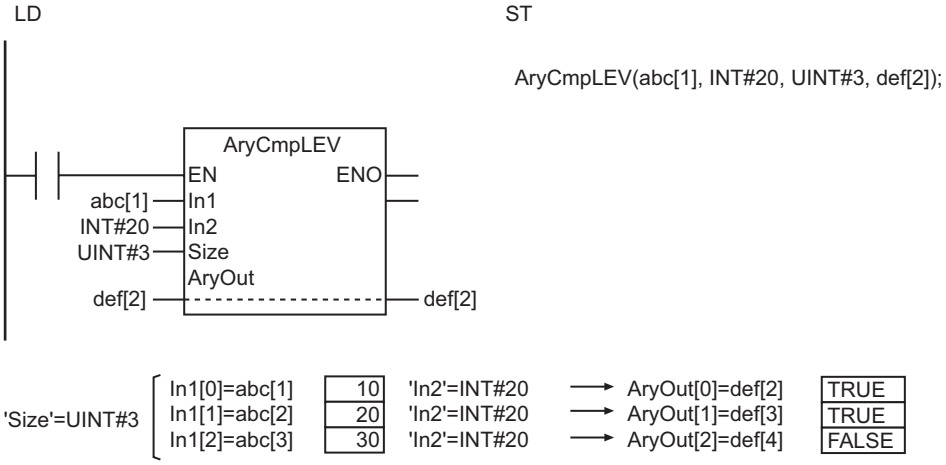
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2	In1[]의 요소와 동일한 데이터 형식																			
Size							○													
AryOut[] 배열	○																			
Out	○																			

기능

비교 배열 In1[0]~In1['Size'-1]과 비교 수치 'In2'의 크기를 비교합니다.
비교 결과는, 비교 결과 배열 AryOut[0]~AryOut['Size'-1]의 대응하는 요소 번호에 저장됩니다.
각 명령의 AryOut[i]의 값은 다음과 같습니다.

명령	AryOut[i]의 값
AryCmpLTV	In1[i]<'In2' 일 때 TRUE, 그 이외일 때 FALSE
AryCmpLEV	In1[i]<='In2' 일 때 TRUE, 그 이외일 때 FALSE
AryCmpGTV	In1[i]>'In2' 일 때 TRUE, 그 이외일 때 FALSE
AryCmpGEV	In1[i]>='In2' 일 때 TRUE, 그 이외일 때 FALSE

AryCmpLEV 명령에서, 'In2'=INT#20, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

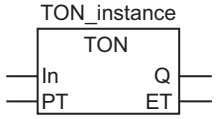
- In1[]과 'In2'의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- AryOut[]의 배열 크기는 'Size' 이상으로 하십시오.
- In1[], 'In2'가 실수이고 나누어 떨어지지 않는 제산 결과 등을 포함하고 있으면, 오차로 인해 처리 결과가 기대한 것과 다를 수 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - In1[], AryOut[] 중 어느 것의 배열 크기가 'Size'보다 작은 경우

타이머 명령

명령	명칭	페이지
TON	온 딜레이 타이머	2-124
TOF	오프 딜레이 타이머	2-129
TP	펄스 출력	2-132
AccumulationTimer	적산 타이머	2-135
Timer	100ms 타이머	2-138

TON

기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TON	온 딜레이 타이머	FB		TON_instance (In, PT, Q, ET);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	타이머 입력	입력	TRUE : 타이머 기동 지시 FALSE : 타이머 리셋 지시	데이터 형식에 따름	–	FALSE
PT	설정 시간		타이머 기동부터 'Q'가 TRUE로 바뀔 때까지의 시간	(*)	ms	0
Q	타이머 출력	출력	TRUE : 타이머 출력 ON FALSE : 타이머 출력 OFF	데이터 형식에 따름	–	–
ET	경과 시간		타이머 기동 이후의 경과 시간	(*)	ms	

* T#0ms~T#106751d_23h_47m_16s_854.775807ms

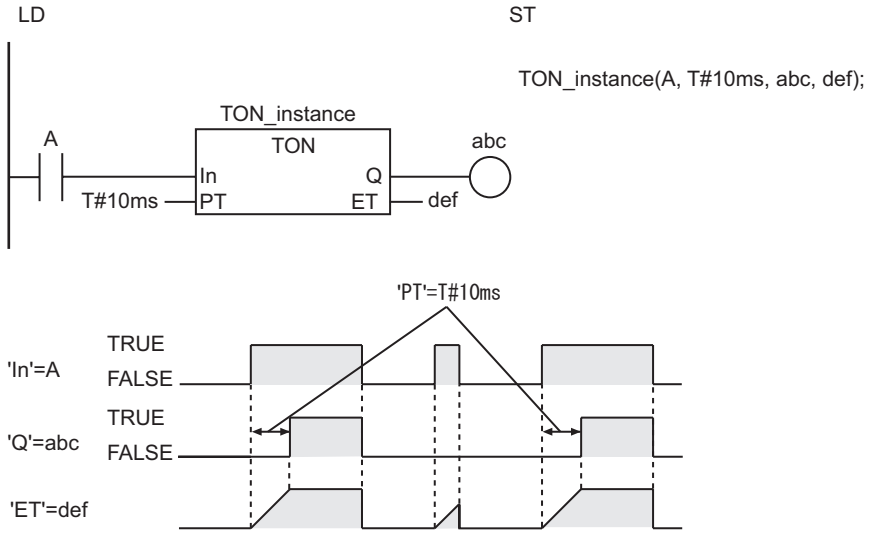
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○																			
PT																○				
Q	○																			
ET																○				

기능

기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 ns입니다. 타이머 입력 'In'이 FALSE에서 TRUE로 변화하면 타이머가 기동하며, 경과 시간 'ET'가 시간이 흐름에 따라 증가합니다.
'ET'가 설정 시간 'PT'에 도달하면 타이머 출력 'Q'가 TRUE가 됩니다. 'ET'의 증가는 그 시점에서 멈춥니다. 'In'이 FALSE가 되면 타이머가 리셋됩니다. 'ET'는 0이 되며, 'Q'는 FALSE가 됩니다.

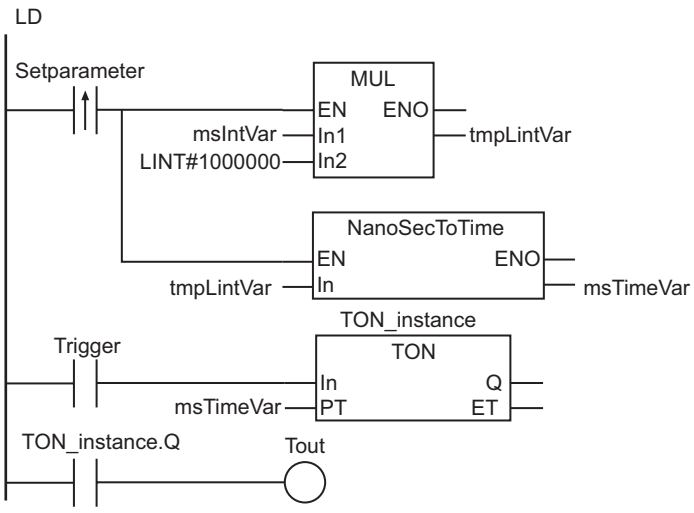
타이머 기동 후 'ET'가 'PT'에 도달하기 전이라도 'In'이 FALSE가 되면 타이머는 리셋됩니다.

'PT'=T#10ms인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다. 변수 A가 TRUE로 변화한 지 10ms 후에 변수 abc의 값이 TRUE가 됩니다.



참고

- 타이머 기동 후 곧바로 타이머 출력이 TRUE가 되며 설정 시간 경과 후에 FALSE가 되는 타이머가 필요한 경우에는 'TP 명령(2-132페이지)'을 사용하십시오.
- 'In'이 FALSE가 되었을 때 타이머가 기동하며 경과 시간이 설정 시간에 도달하면 타이머 출력이 FALSE가 되는 타이머가 필요한 경우에는 'TOF 명령(2-129페이지)'을 사용하십시오.
- 명령 실행 시간을 단축하고자 하는 경우에는 100ms 단위로 계시를 수행 'Timer 명령(2-138페이지)'을 사용하십시오.
- TIME 형식과 호환되지 않는 터치 패널 등과 연결한 경우는 정수 형식으로 표시된 설정 시간을 TIME 형식으로 일단 변환한 후 이 명령에 입력해야 합니다. 정수 형식에서 TIME 형식으로 변환하려면 'NanoSecToTime 명령(2-612페이지)'을 사용하십시오. 또한 TIME 형식에서 정수 형식으로 변환하려면 'TimeToNanoSec 명령(2-610페이지)'을 사용하십시오. 어느 명령이든 시간의 단위는 나노초입니다. INT 형식 변수 msIntVar이 ms 단위의 설정 시간인 경우의 사용자 프로그램은 다음과 같습니다.




```
ST
tmpLintVar:=msIntVar*LINT#1000000;
msTimeVar:=NanoSecToTime(tmpLintVar);
TON_instance(In:=Trigger, PT:=msTimeVar, Q=>Tout);
```

사용상의 주의 사항

- 'PT'에 대해서 'Q'가 TRUE가 되는 계시 오차는 -100ns ~ (100ns+1 태스크 주기)입니다.
  상기 범위의 내역은 다음과 같습니다.
  - ±100ns는 'ET'의 계시 오차.
  - 태스크 주기마다 'ET'의 계시가 'PT'에 도달했는지 판단하며, 판단 직후에 'ET'의 계시가 'PT'에 도달한 경우에 +1 태스크 주기 지연됩니다.
- Sysmac Studio에서는 시간값이 0.001ms 단위로 표시되지만 계시 정밀도는 1ns입니다.
- 'PT', 'ET'는 ns 단위로 설정하며 계시 정밀도는 100ns입니다.
- 운전 시작 시 이미 'In'의 값이 TRUE인 경우, 그 시점부터 계시를 시작합니다.
- 'PT'에 T#0ms 또는 음수가 설정된 경우, 'In'의 값이 TRUE가 된 직후에 'Q'가 TRUE가 됩니다.
- 'In'의 값이 TRUE인 동안 'PT'의 값을 변경할 수 있습니다. 그 경우의 동작은 다음과 같습니다.

타이머의 상태	'Q'의 값	변경 후의 'PT'의 값	동작
계시 종료 후	TRUE	–	'Q'의 값은 그대로 TRUE입니다. 'ET'의 값도 변화하지 않습니다(변경 전의 'PT'의 값 그대로).
계시 중	FALSE	'PT' ≧ 'ET'	계시를 속행합니다. 'ET'의 값이 변경 후의 'PT'의 값에 도달한 시점에서, 'Q'의 값은 TRUE가 되며 'ET'의 증가가 멈춥니다.
		'PT' < 'ET'	즉시 'Q'의 값이 TRUE가 됩니다. 'ET'의 증가도 즉시 멈춥니다.

- 이 명령이 마스터 컨트롤 영역에 존재하며 마스터 컨트롤에 의한 리셋이 실행된 경우, 타이머가 리셋됩니다. 'ET'의 값이 0이 되며 'Q'의 값은 FALSE가 됩니다.
- JMP 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'ET'의 값은 갱신되지 않습니다. 하지만 그 동안에도 계시는 계속됩니다. 따라서, 다음으로 이 명령이 실행될 때 'ET'는 올바른 값으로 갱신됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

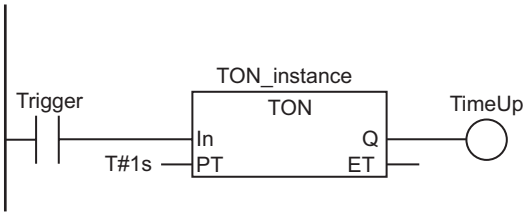
샘플 프로그램

● 1개의 온 딜레이 타이머로 시간을 측정하는 예

Trigger의 값이 TRUE가 된 지 1s 후에 TimeUp의 값이 TRUE가 됩니다.

LD

명칭	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
TimeUp	BOOL	FALSE	타이머 출력
TON_instance	TON		



ST

명칭	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
TimeUp	BOOL	FALSE	타이머 출력
TON_instance	TON		

```
IF (Trigger=TRUE) THEN
   TON_instance(In:=TRUE, PT:=T#1s, Q=>TimeUp);
ELSE
   TON_instance(In:=FALSE, Q=>TimeUp);
END_IF;
```

ST

명칭	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
TimeUp	BOOL	FALSE	타이머 출력
TON_instance	TON		

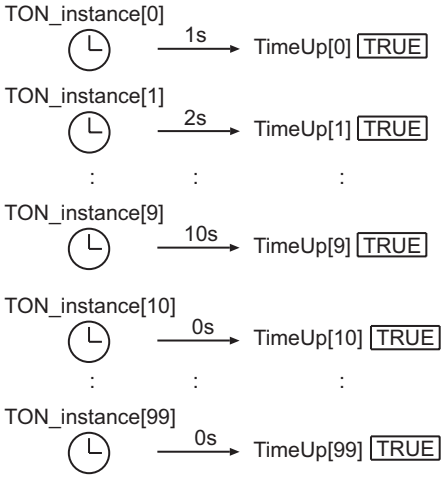
```
TON_instance(In:=Trigger, PT:=T#1s, Q=>TimeUp);
```

● 복수의 온 딜레이 타이머로 시간을 측정하는 예

TON_instance[0]~TON_instance[99]의 100개의 온 딜레이 타이머를 준비합니다. 타이머 입력 Input[0]~Input[99]의 값을 TRUE로 해서 각각의 타이머를 기동합니다.

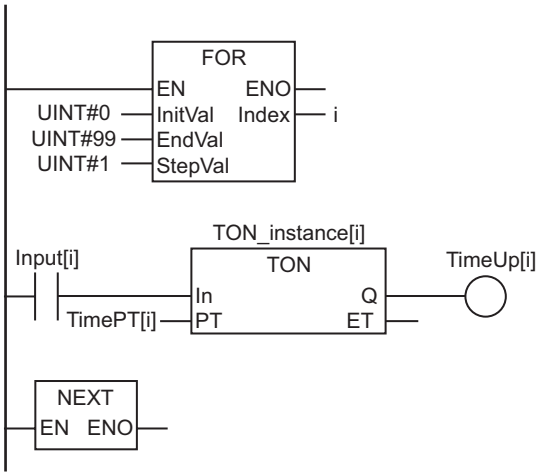
첫 10개의 타이머 TON_instance[0]~TON_instance[9]는 각각이 기동한 지 i+1초 후(i=0~9)에 TimeUp[i](i=0~9)의 값을 TRUE로 합니다.

남은 90개의 타이머 TON_instance[10]~TON_instance[99]는, 기동 후 즉시 TimeUp[i](i=10~99)의 값을 TRUE로 합니다.



LD

명칭	데이터 형식	초기값	코멘트
Input	ARRAY[0..99] OF BOOL	[100(FALSE)]	타이머 입력
TimeUp	ARRAY[0..99] OF BOOL	[100(FALSE)]	타이머 출력
TimePT	ARRAY[0..99] OF TIME	[T#1s, T#2s, T#3s, T#4s, T#5s, T#6s, T#7s, T#8s, T#9s, T#10s, 90(T#0s)]	설정 시간
TON_instance	ARRAY[0..99] OF TON		
i	UINT	0	인텍스



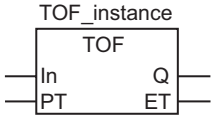
ST

명칭	데이터 형식	초기값	코멘트
Input	ARRAY[0..99] OF BOOL	[100(FALSE)]	타이머 입력
TimeUp	ARRAY[0..99] OF BOOL	[100(FALSE)]	타이머 출력
TimePT	ARRAY[0..99] OF TIME	[T#1s, T#2s, T#3s, T#4s, T#5s, T#6s, T#7s, T#8s, T#9s, T#10s, 90(T#0s)]	설정 시간
TON_instance	ARRAY[0..99] OF TON		
i	UINT	0	인텍스

```
FOR i:=UINT#0 TO UINT#99 DO
   TON_instance[i](
      In   := Input[i],
      PT  := TimePT[i],
      Q   =>TimeUp[i]);
END_FOR;
```

TOF

기동 시점부터 설정 시간 경과 후에 FALSE를 출력하는 타이머입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TOF	오프 딜레이 타이머	FB		TOF_instance (In, PT, Q, ET);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	타이머 입력	입력	TRUE : 타이머 리셋 지시 FALSE : 타이머 기동 지시	데이터 형식에 따름	–	FALSE
PT	설정 시간		타이머 기동부터 'Q'가 FALSE로 바뀔 때까지의 시간	(*)	ms	0
Q	타이머 출력	출력	TRUE : 타이머 출력 ON FALSE : 타이머 출력 OFF	데이터 형식에 따름	–	–
ET	경과 시간		타이머 기동 이후의 경과 시간	(*)	ms	

* T#0ms~T#106751d_23h_47m_16s_854.775807ms

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○																			
PT																○				
Q	○																			
ET																○				

기능

기동 시점부터 설정 시간 경과 후에 FALSE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 ns입니다.

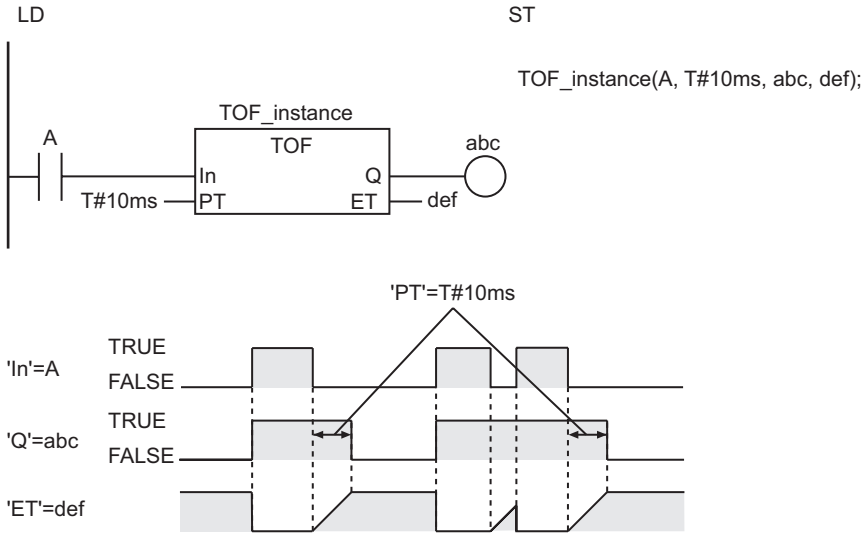
타이머 입력 'In'이 TRUE에서 FALSE로 변화하면 타이머가 기동하며, 경과 시간 'ET'가 시간이 흐름에 따라 증가합니다.

'ET'가 설정 시간 'PT'에 도달하면 타이머 출력 'Q'가 FALSE가 됩니다. 'ET'의 증가는 그 시점에서 멈춥니다.

'In'이 TRUE가 되면 타이머가 리셋됩니다. 'ET'는 0이 되며, 'Q'는 TRUE가 됩니다.

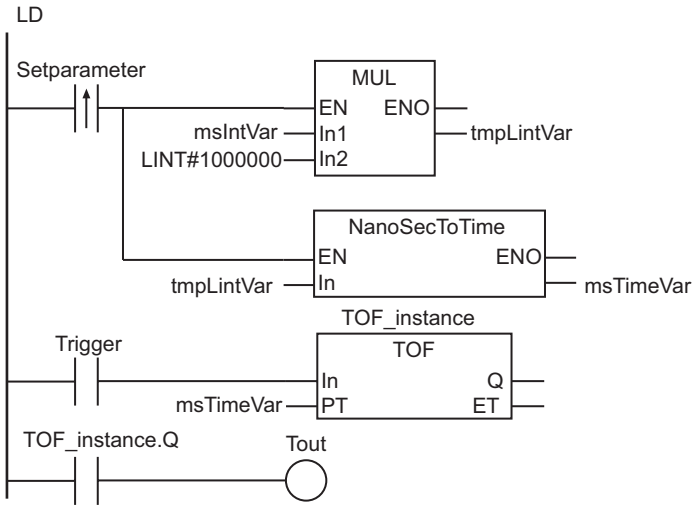
타이머 기동 후 'ET'가 'PT'에 도달하기 전이라도 'In'이 FALSE가 되면 타이머는 리셋됩니다.

'PT'=T#10ms인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다. 변수 A가 FALSE로 변화한 지 10ms 후에 변수 abc의 값이 FALSE가 됩니다.



참고

- 타이머 기동 후 즉시 타이머 출력이 TRUE가 되며 설정 시간 경과 후에 FALSE가 되는 타이머가 필요한 경우에는 'TP 명령(2-132페이지)'을 사용하십시오.
- 'In'이 TRUE가 되었을 때 타이머가 기동하며 경과 시간이 설정 시간에 도달하면 타이머 출력이 TRUE가 되는 타이머가 필요한 경우에는 'TON 명령(2-124페이지)'을 사용하십시오.
- TIME 형식과 호환되지 않는 터치 패널 등과 연결한 경우는 정수 형식으로 표시된 설정 시간을 TIME 형식으로 일단 변환한 후 이 명령에 입력해야 합니다. 정수 형식에서 TIME 형식으로 변환하려면 'NanoSecToTime 명령(2-612페이지)'을 사용하십시오. 또한 TIME 형식에서 정수 형식으로 변환하려면 'TimeToNanoSec 명령(2-610페이지)"을 사용하십시오. 어느 명령이든 시간의 단위는 나노초입니다.
  INT 형식 변수 msIntVar이 ms 단위의 설정 시간인 경우의 사용자 프로그램은 다음과 같습니다.




```
ST
tmpLintVar:=msIntVar*LINT#1000000;
msTimeVar:=NanoSecToTime(tmpLintVar);
TOF_instance(In:=Trigger, PT:=msTimeVar, Q=>Tout);
```

사용상의 주의 사항

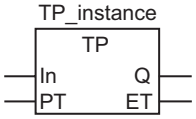
- 'PT'에 대해서 'Q'가 TRUE가 되는 계시 오차는 -100ns ~ (100ns+1 태스크 주기)입니다.
  상기 범위의 내역은 다음과 같습니다.
  - ±100ns는 'ET'의 계시 오차.
  - 태스크 주기마다 'ET'의 계시가 'PT'에 도달했는지 판단하며, 판단 직후에 'ET'의 계시가 'PT'에 도달한 경우에 +1 태스크 주기 지연됩니다.
- Sysmac Studio에서는 시간값이 0.001ms 단위로 표시되지만 계시 정밀도는 1ns입니다.
- 'PT'에 T#0ms 또는 음수가 설정된 경우, 'In'의 값이 FALSE가 된 직후에 'Q'가 FALSE가 됩니다.
- 이 명령을 실행한 직후 'In'의 값이 TRUE라면, 'Q'의 값은 TRUE가 됩니다. 'Q'의 값이 FALSE가 되는 것은, 타이머 기동 후 'PT'만큼 시간이 경과했을 때입니다.
- 'In'의 값이 FALSE인 동안 'PT'의 값을 변경할 수 있습니다. 그 경우의 동작은 다음과 같습니다.

타이머의 상태	'Q'의 값	변경 후의 'PT'의 값	동작
계시 종료 후	FALSE	–	'Q'의 값은 그대로 FALSE입니다. 'ET'의 값도 변화하지 않습니다(변경 전의 'PT'의 값 그대로).
계시 중	TRUE	'PT' ≧ 'ET'	계시를 속행합니다. 'ET'의 값이 변경 후의 'PT'의 값에 도달한 시점에서, 'Q'의 값은 FALSE가 되며 'ET'의 증가가 멈춥니다.
		'PT' < 'ET'	즉시 'Q'의 값이 FALSE가 됩니다. 'ET'의 증가도 즉시 멈춥니다.

- 이 명령이 마스터 컨트롤 영역에 존재하며 마스터 컨트롤에 의한 리셋이 실행된 경우, 다음과 같이 동작합니다.
  - 'ET'의 값이 0이 되며 'Q'의 값은 TRUE가 됩니다.
  - 'Q'를 다음 단계의 Out 명령에 연결한 경우, 그 Out 명령에는 FALSE가 입력됩니다.
  - 리셋 해제 후에 계시를 시작합니다.
- JMP 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'ET'의 값은 갱신되지 않습니다. 하지만 그 동안에도 계시는 계속됩니다. 따라서, 다음으로 이 명령이 실행될 때 'ET'는 올바른 값으로 갱신됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

TP

기동 시점부터 설정 시간만큼 TRUE를 출력하는 타이머입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TP	펄스 출력	FB		TP_instance (In, PT, Q, ET);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	타이머 입력	입력	TRUE : 타이머 기동 지시 FALSE : 타이머 리셋 지시	데이터 형식에 따름	–	FALSE
PT	설정 시간		'Q'가 TRUE를 유지하는 시간	(*)	ms	0
Q	타이머 출력	출력	TRUE : 타이머 출력 ON FALSE : 타이머 출력 OFF	데이터 형식에 따름	–	–
ET	경과 시간		타이머 기동 이후의 경과 시간	(*)	ms	

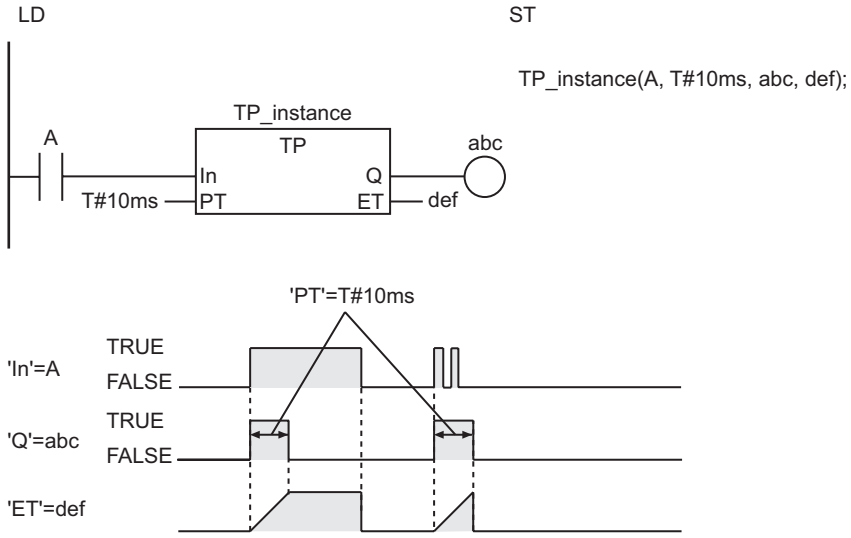
* T#0ms~T#106751d_23h_47m_16s_854.775807ms

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○																			
PT																○				
Q	○																			
ET																○				

기능

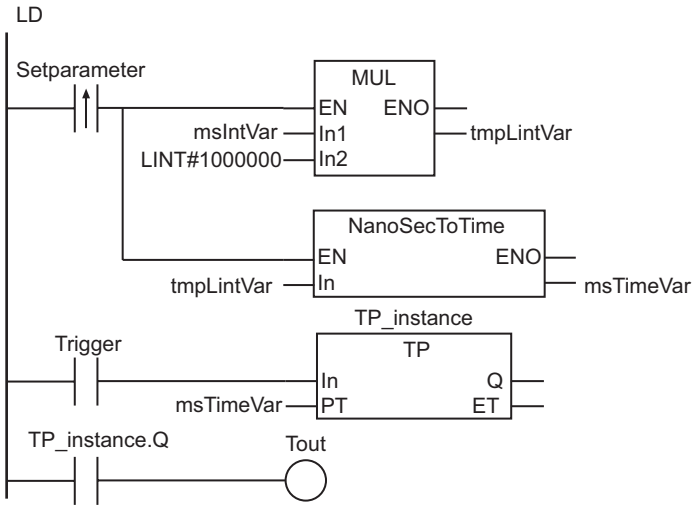
기동 시점부터 설정 시간만큼 TRUE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 ns입니다.
타이머 입력 'In'이 FALSE에서 TRUE로 변화하면 타이머가 기동하며, 타이머 출력 'Q'가 TRUE가 됩니다.
그리고 경과 시간 'ET'가 시간이 흐름에 따라 증가합니다.
'ET'가 설정 시간 'PT'에 도달하면 타이머 출력 'Q'가 FALSE가 됩니다. 'ET'의 증가는 그 시점에서 멈춥니다.
'In'이 FALSE가 되면 타이머가 리셋됩니다. 'ET'는 0이 됩니다.
단, 기동 후 'ET'가 'PT'에 도달하기 전에 'In'이 FALSE가 되면 타이머는 리셋되지 않습니다.

'PT'=T#10ms인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다. 변수 A가 TRUE로 변화하면 변수 abc의 값이 TRUE가 되며, 10ms 후에 FALSE가 됩니다.



참고

- 'In'이 TRUE가 되었을 때 타이머가 기동하며 경과 시간이 설정 시간에 도달하면 타이머 출력이 TRUE가 되는 타이머가 필요한 경우에는 'TON 명령(2-124페이지)'을 사용하십시오.
- 'In'이 FALSE가 되었을 때 타이머가 기동하며 경과 시간이 설정 시간에 도달하면 타이머 출력이 FALSE가 되는 타이머가 필요한 경우에는 'TOF 명령(2-129페이지)'을 사용하십시오.
- TIME 형식과 호환되지 않는 터치 패널 등과 연결한 경우는 정수 형식으로 표시된 설정 시간을 TIME 형식으로 일단 변환한 후 이 명령에 입력해야 합니다. 정수 형식에서 TIME 형식으로 변환하려면 'NanoSecToTime 명령(2-612페이지)'을 사용하십시오. 또한 TIME 형식에서 정수 형식으로 변환하려면 'TimeToNanoSec 명령(2-610페이지)'을 사용하십시오. 어느 명령이든 시간의 단위는 나노초입니다. INT 형식 변수 msIntVar이 ms 단위의 설정 시간인 경우의 사용자 프로그램은 다음과 같습니다.




```
ST
tmpLintVar:=msIntVar*LINT#1000000;
msTimeVar:=NanoSecToTime(tmpLintVar);
TP_instance(In:=Trigger, PT:=msTimeVar, Q=>Tout);
```

사용상의 주의 사항

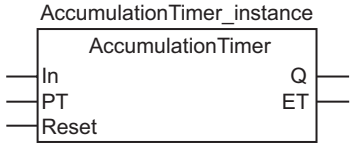
- 'PT'에 대해서 'Q'가 TRUE가 되는 계시 오차는 -100ns ~ (100ns+1 태스크 주기)입니다.
  상기 범위의 내역은 다음과 같습니다.
  - ±100ns는 'ET'의 계시 오차.
  - 태스크 주기마다 'ET'의 계시가 'PT'에 도달했는지 판단하며, 판단 직후에 'ET'의 계시가 'PT'에 도달한 경우에 +1 태스크 주기 지연됩니다.
- Sysmac Studio에서는 시간값이 0.001ms 단위로 표시되지만 계시 정밀도는 1ns입니다.
- 운전 시작 시 이미 'In'의 값이 TRUE인 경우, 그 시점부터 계시를 시작합니다.
- 'PT'에 T#0ms 또는 음수가 설정된 경우, 'In'의 값이 TRUE가 되더라도 'Q'는 TRUE가 되지 않습니다.
- 'In'의 값이 TRUE인 동안 'PT'의 값을 변경할 수 있습니다. 그 경우의 동작은 다음과 같습니다.

타이머의 상태	'Q'의 값	변경 후의 'PT'의 값	동작
계시 종료 후	FALSE	–	'Q'의 값은 그대로 FALSE입니다. 'ET'의 값도 변화하지 않습니다(변경 전의 'PT'의 값 그대로).
계시 중	TRUE	'PT' ≧'ET'	계시를 속행합니다. 'ET'의 값이 변경 후의 'PT'의 값에 도달한 시점에서, 'Q'의 값은 FALSE가 되며 'ET'의 증가가 멈춥니다.
		'PT' < 'ET'	즉시 'Q'의 값이 FALSE가 됩니다. 'ET'의 증가도 즉시 멈춥니다.

- 이 명령이 마스터 컨트롤 영역에 존재하며 마스터 컨트롤에 의한 리셋이 실행된 경우, 계시 중이라면 완료될 때까지 계속 계시합니다. 그 후 'ET'의 값은 0이 되며 'Q'의 값은 FALSE가 됩니다. 단, 'Q'를 다음 단계의 Out 명령에 연결한 경우, 'Q'의 값이 TRUE라도 Out 명령에는 FALSE가 입력됩니다.
- JMP 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'ET'의 값은 갱신되지 않으며 계시를 수행하지 않습니다. 다음으로 이 명령이 실행될 때 계시를 재개합니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

AccumulationTimer

타이머 입력이 TRUE인 시간을 적산하는 타이머입니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Accumulation Timer	적산 타이머	FB		AccumulationTimer_instance(In, PT, Reset, Q, ET);

변수

	명칭	입/출력	내용	유효 범위	단위	초기
In	타이머 입력	입력	TRUE : 타이머 작동 FALSE : 타이머 정지	데이터 형식에 따름	–	FALSE
PT	설정 시간		계시의 최대값	(*)	ms	0
Reset	리셋		TRUE : 타이머 리셋 FALSE : 타이머 리셋 안 함	데이터 형식에 따름	–	FALSE
Q	타이머 출력	출력	TRUE : 'ET'가 'PT'에 도달함 FALSE : 'ET'가 'PT'에 도달하지 않음	데이터 형식에 따름	–	–
ET	적산 시간		적산 시간	(*)	ms	

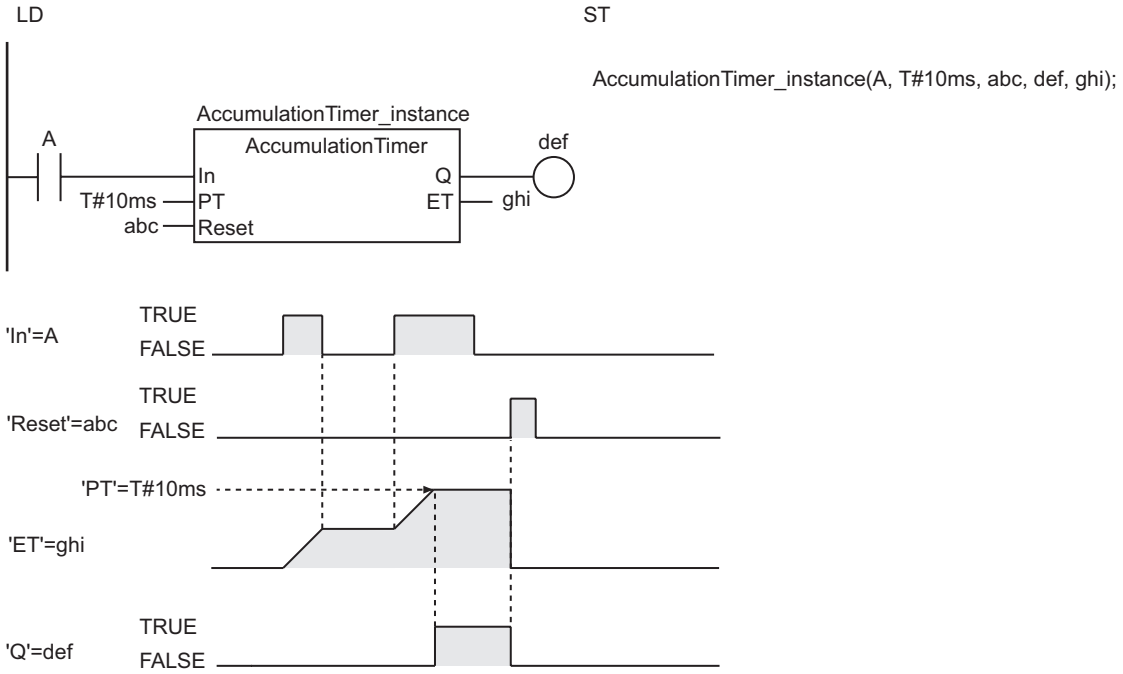
* T#0ms~T#106751d_23h_47m_16s_854.775807ms

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○																			
PT																○				
Reset	○																			
Q	○																			
ET																○				

기능

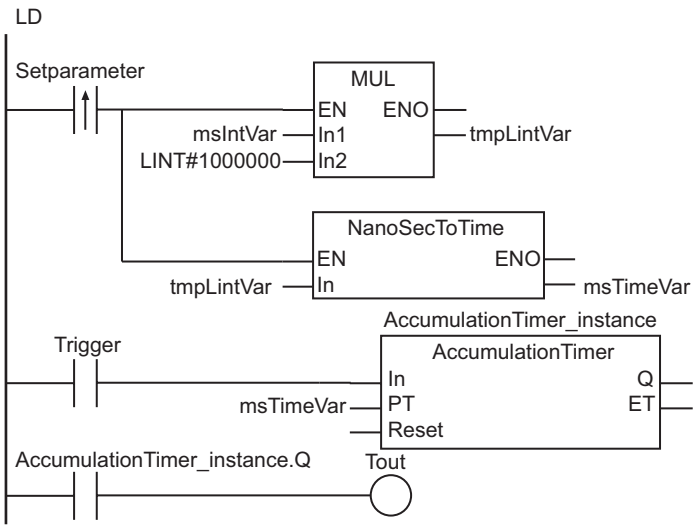
타이머 입력 'In'이 TRUE인 시간을 적산하는 타이머입니다. 설정 시간의 최소 단위는 ns입니다.
리셋 'Reset'이 FALSE일 때 'In'이 FALSE에서 TRUE로 변화하면 타이머가 기동하며, 적산 시간 'ET'가 시간의 흐름에 따라 증가합니다.
'In'이 FALSE가 되면 타이머가 정지합니다. 하지만 그 시점의 'ET'의 값은 유지됩니다.
'In'이 다시 TRUE가 되면 타이머가 재기동합니다. 그리고 'ET'는 유지되던 값에서 증가합니다.
'ET'가 설정 시간 'PT'에 도달하면 타이머 출력 'Q'가 TRUE가 됩니다. 'ET'의 증가는 그 시점에서 멈춥니다.
'Reset'이 TRUE가 되면 타이머가 리셋됩니다. 'ET'의 값이 0이 되며 'Q'는 FALSE가 됩니다.

'PT'=T#10ms인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다. 변수 A가 TRUE인 적산 시간이 10ms가 되면, 변수 abc의 값이 TRUE가 됩니다.



참고

- 'In'이 FALSE가 되었을 때 타이머 출력 및 경과 시간을 리셋하려는 경우에는 'TON 명령(2-124페이지)'을 사용하십시오.
- TIME 형식과 호환되지 않는 터치 패널 등과 연결한 경우는 정수 형식으로 표시된 설정 시간을 TIME 형식으로 일단 변환한 후 이 명령에 입력해야 합니다. 정수 형식에서 TIME 형식으로 변환하려면 'NanoSecToTime 명령(2-612페이지)'을 사용하십시오. 또한 TIME 형식에서 정수 형식으로 변환하려면 'TimeToNanoSec 명령(2-610페이지)'을 사용하십시오. 어느 명령이든 시간의 단위는 나노초입니다. INT 형식 변수 msIntVar이 ms 단위의 설정 시간인 경우의 사용자 프로그램은 다음과 같습니다.




```
ST
tmpLintVar:=msIntVar*LINT#1000000;
msTimeVar:=NanoSecToTime(tmpLintVar);
AccumulationTimer_instance(In:=Trigger, PT:=msTimeVar, Q=>Tout);
```

사용상의 주의 사항

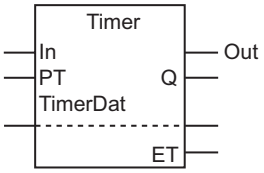
- 'PT'에 대해서 'Q'가 TRUE가 되는 계시 오차는 -100ns ~ (100ns+1 태스크 주기)입니다.
  상기 범위의 내역은 다음과 같습니다.
  - ±100ns는 'ET'의 계시 오차.
  - 태스크 주기마다 'ET'의 계시가 'PT'에 도달했는지 판단하며, 판단 직후에 'ET'의 계시가 'PT'에 도달한 경우에 +1 태스크 주기 지연됩니다.
- Sysmac Studio에서는 시간값이 0.001ms 단위로 표시되지만 계시 정밀도는 1ns입니다.
- 'In'과 'Reset'이 모두 TRUE인 경우, 'Reset'이 우선합니다. 즉, 'ET'의 값은 0이 되며 'Q'의 값은 FALSE가 됩니다.
- 운전 시작 시 이미 'In'의 값이 TRUE인 경우, 그 시점부터 계시를 시작합니다.
- 'PT'에 T#0ms 또는 음수가 설정된 경우, 'In'의 값이 TRUE가 된 직후에 'Q'가 TRUE가 됩니다.
- 'ET'의 값이 'PT'의 값에 도달하기 전에 'PT'의 값을 변경할 수 있습니다. 그 경우의 동작은 다음과 같습니다.

타이머의 상태	'Q'의 값	변경 후의 'PT'의 값	동작
계시 종료 후	TRUE	–	'Q'의 값은 그대로 TRUE입니다. 'ET'의 값도 변화하지 않습니다(변경 전의 'PT'의 값 그대로).
계시 중	FALSE	'PT' ≧ 'ET'	'In'의 값이 TRUE가 되면 계시를 속행합니다. 'ET'의 값이 변경 후의 'PT'의 값에 도달한 시점에서, 'Q'의 값은 TRUE가 되며 'ET'의 증가가 멈춥니다.
		'PT' < 'ET'	'In'의 값이 TRUE가 되면, 즉시 'Q'의 값이 TRUE가 됩니다. 'ET'의 증가도 즉시 멈춥니다.

- 이 명령이 마스터 컨트롤 영역에 존재하며 마스터 컨트롤에 의한 리셋이 실행된 경우, 다음과 같이 동작합니다.
  - 타이머가 정지됩니다. 'ET' 및 'Q'는 그 시점의 값을 유지합니다.
  - 마스터 컨트롤에 의한 리셋이 해제되면 'ET'는 유지하던 값에서 다시 증가되기 시작합니다.
  - 'Q'를 다음 단계의 Out 명령에 연결한 경우, 'Q'의 값이 TRUE라도 Out 명령에는 FALSE가 입력됩니다.
  - 'Reset'은 유효합니다.
- JMP 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'ET'의 값은 갱신되지 않습니다. 하지만 그 동안에도 계시는 계속됩니다. 따라서, 다음으로 이 명령이 실행될 때 'ET'는 올바른 값으로 갱신됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

Timer

기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 100ms입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Timer	100ms 타이머	FUN		Out: =Timer (In, PT, TimerDat, Q, ET);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	타이머 입력	입력	TRUE : 타이머 기동 지시 FALSE : 타이머 리셋 지시	데이터 형식에 따름	–	FALSE
PT	설정 시간		타이머 기동부터 'Q'가 TRUE로 바뀔 때까지의 시간		ms	(*)
TimerDat	타이머 상태	입출력	타이머의 현재 상태	–	–	–
Out	반환값	출력	TRUE : 타이머 출력 ON FALSE : 타이머 출력 OFF	데이터 형식에 따름	–	–
Q	타이머 출력		'Out'과 같은 의미			
ET	남은 시간		남은 시간		ms	

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○																			
PT							○													
TimerDat	구조체 _sTimer																			
Out	○																			
Q	○																			
ET							○													

기능

기동 시점부터 설정 시간 경과 후에 TRUE를 출력하는 타이머입니다. 설정 시간의 최소 단위는 100ms입니다.

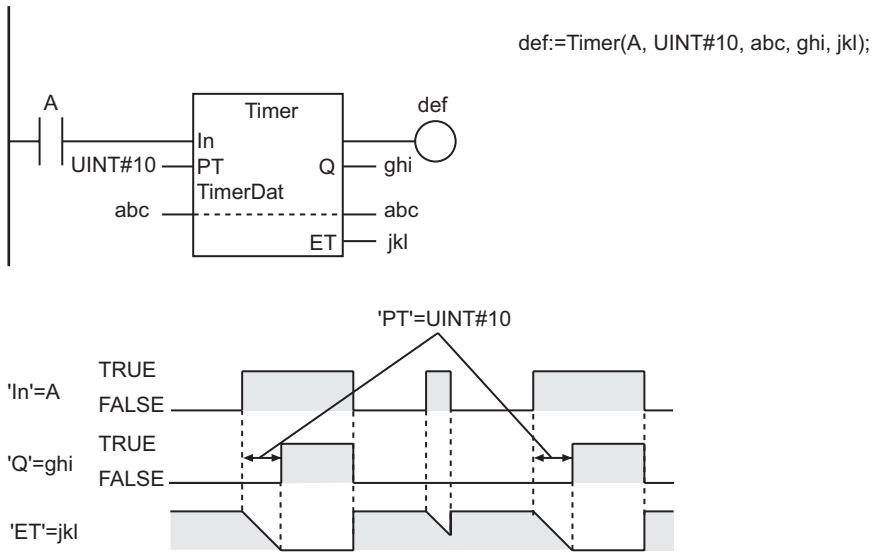
타이머 입력 'In'의 값이 FALSE가 되면 타이머가 리셋됩니다. 남은 시간 'ET'에 설정 시간 'PT'가 세트되며, 타이머 출력 'Q'의 값이 FALSE가 됩니다.
'In'이 FALSE에서 TRUE로 변화하면 타이머가 기동합니다. 'ET'의 값이 시간이 흐름에 따라 감소합니다.

'ET'의 값이 0에 도달하면 타이머 출력 'Q'가 TRUE가 됩니다. 'ET'의 값의 감소는 그 시점에서 멈춥니다.

타이머 기동 후 'ET'가 0에 도달하기 전이라도 'In'이 FALSE가 되면 타이머는 리셋됩니다.

타이머 상태 'TimerDat'의 데이터 형식은 구조체 _sTimer입니다.

'PT'=UINT#10인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다. 변수 A가 TRUE로 변화한 지 1,000ms(1s) 후에 변수 ghi의 값이 TRUE가 됩니다.



참고

보다 정확한 계시를 수행하는 경우, 1ns 단위의 계시를 수행함 🕮 'TON 명령(2-124페이지)'을 사용하십시오. TON 명령은 명령 실행 시에 1ns 단위의 계시를 수행하므로 Timer 명령보다 정확한 계시가 가능합니다. 단, 명령 실행 시간은 Timer 명령이 더 짧습니다.

사용상의 주의 사항

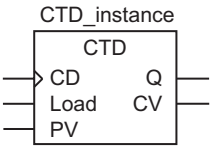
- 계시는 이 명령을 기술한 POU의 시작 부분에서 수행합니다. 따라서 동일한 POU 내의 어느 곳에서 이 명령을 실행하더라도 'ET'의 값은 동일합니다.
- 'PT'에 대해서 'Q'가 TRUE가 되는 계시 오차는 +1 태스크 주기입니다.
  상기 범위의 내역은 다음과 같습니다.
  - 태스크 주기마다 'ET'의 계시가 'PT'에 도달했는지 판단하며, 판단 직후에 'ET'의 계시가 'PT'에 도달한 경우에 +1 태스크 주기 지연됩니다.
- 'TimerDat'는 입출력 변수이지만 값을 전달할 필요는 없습니다. 구조체 _sTimer 분량의 메모리 영역을 확보하여 이 명령에 전달하십시오.
- 'TimerDat'의 내용은 변경하지 마십시오.
- 운전 시작 시 이미 'In'의 값이 TRUE인 경우, 그 시점부터 계시를 시작합니다.
- 'PT'의 값을 변경한 경우, 그것이 반영되는 것은 다음 타이머 리셋 시점입니다. 계시 중에는 반영되지 않습니다.
- 이 명령이 마스터 컨트롤 영역에 존재하며 마스터 컨트롤에 의한 리셋이 실행된 경우, 타이머가 리셋됩니다. 'ET'에는 'PT'의 값이 세트되며, 'Q'의 값은 FALSE가 됩니다.
- JMP 계통 명령(JMP 명령 등)의 실행에 의해 이 명령이 실행되지 않은 경우, 'ET'의 값은 갱신되지 않습니다. 하지만 그 동안에도 계시는 계속됩니다. 따라서, 다음으로 이 명령이 실행될 때 'ET'는 올바른 값으로 갱신됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q' 및 'Out'의 값은 FALSE가 됩니다.

카운터 명령

명령	명칭	페이지
CTD	감산 카운터	2-142
CTD_**	감산 카운터 그룹	2-144
CTU	가산 카운터	2-146
CTU_**	가산 카운터 그룹	2-148
CTUD	가역 카운터	2-151
CTUD_**	가역 카운터 그룹	2-155

CTD

카운터 입력 신호가 입력될 때마다 감산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 INT 입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CTD	감산 카운터	FB		CTD_instance(CD, Load, PV, Q, CV);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CD	카운터 입력	입력	카운터 입력	데이터 형식에 따름	–	FALSE
Load*1	로드 신호		TRUE: 'CV'에 'PV'를 로드			
PV	사전 설정값		카운터의 사전 설정값	0~32767		0
Q	카운터 출력	출력	TRUE : 카운터 출력 ON FALSE : 카운터 출력 OFF	데이터 형식에 따름	–	–
CV	카운터값		카운터의 현재값	0~32767		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Load를 LD로 기술할 수 있습니다. 예를 들어 CTD_instance(CD:=A, LD:=abc, PV:=INT#5, Q=>def, CV=>ghi);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CD	○																			
Load	○																			
PV											○									
Q	○																			
CV											○									

기능

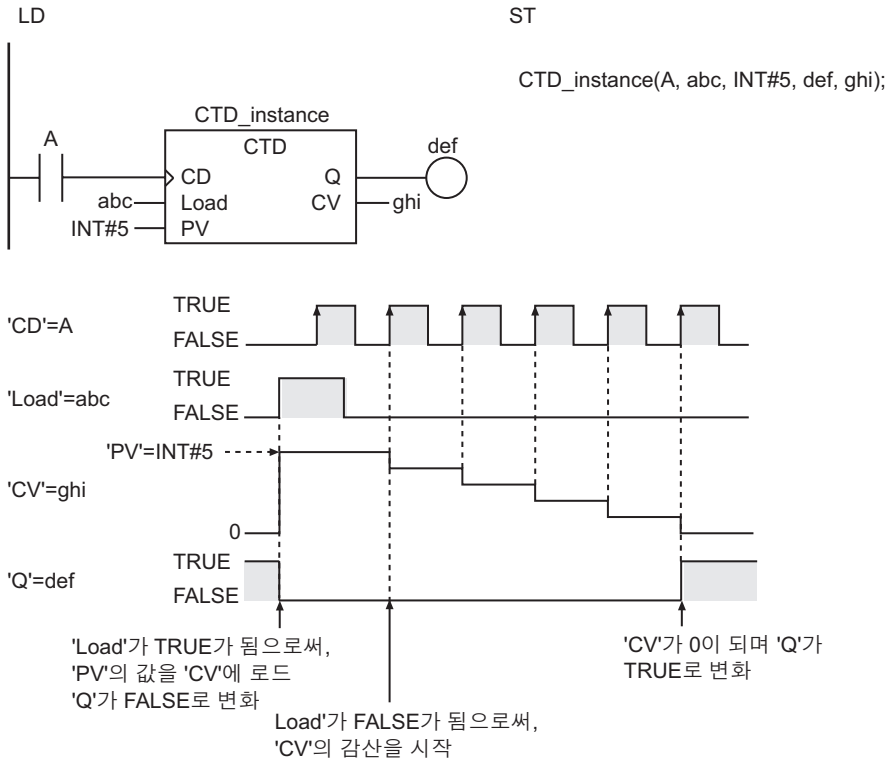
감산 카운터입니다.사전 설정값, 카운터값의 데이터 형식은 INT입니다.

로드 신호 'Load'가 TRUE로 설정되면, 사전 설정값 'PV'의 값을 카운터값 'CV'에 로드해서 카운터 출력 'Q'가 FALSE가 됩니다.
카운터 입력 신호 'CD'가 상승할 때 'CV'를 감산해 나갑니다. 'CV'의 값이 0 이하에 도달하면 'Q'의 값이 TRUE로 바뀝니다.

'CV'의 값이 0 이하가 되면, 그 이상 'CD'가 상승하더라도 'CV'는 변화하지 않습니다.

'Load'가 TRUE일 동안은 'CD'를 무시합니다. 'CV'의 감산은 수행하지 않습니다.

'PV'=INT#5인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

- 카운터 입력 신호가 입력될 때마다 가산되는 카운터가 필요한 경우에는 'CTU 명령(2-146페이지)'을 사용하십시오.
- 가산과 감산이 모두 가능한 카운터가 필요한 경우에는 'CTUD 명령(2-151페이지)'을 사용하십시오.

사용상의 주의 사항

- 카운트 다운 종료 후에 카운터를 재기동하려면 'Load'의 값을 일단 TRUE로 한 후 FALSE로 하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Load'의 값이 TRUE가 된 시점에서 'PV'의 값을 'CV'에 로드합니다. 'CV'의 값이 0 이하이므로, 즉시 'Q'의 값이 TRUE가 됩니다. 그 후, 'CD'가 변화하더라도 'CV'의 감산은 수행하지 않습니다.
- 'CD'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CD'의 값이 TRUE라면 'CV'의 감산이 1회 수행됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

CTD_**

카운터 입력 신호가 입력될 때마다 감산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CTD_**	감산 카운터 그룹	FB	CTD_**_instance CTD_** CD Q Load CV PV 단, **는 DINT, LINT, UDINT, ULINT 중 하나	CTD_**_instance (CD, Load, PV, Q, CV); 단, **는 DINT, LINT, UDINT, ULINT 중 어느 것

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CD	카운터 입력	입력	카운터 입력	데이터 형식에 따름	–	FALSE
Load*1	로드 신호		TRUE : 'CV'에 'PV'를 로드			
PV	사전 설정값		카운터의 사전 설정값	데이터 형식에 따름 *2		0
Q	카운터 출력	출력	TRUE : 카운터 출력 ON FALSE : 카운터 출력 OFF	데이터 형식에 따름	–	–
CV	카운터값		카운터의 현재값	데이터 형식에 따름		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Load를 LD로 기술할 수 있습니다. 예를 들어 CTD_LINT_instance(CD:=A, LD:=abc, PV:=LINT#5, Q=>def, CV=>ghi);로 기술할 수 있습니다.

*2. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CD	○																			
Load	○																			
PV								○	○			○	○							
Q	○																			
CV	'PV'와 같은 데이터 형식																			

기능

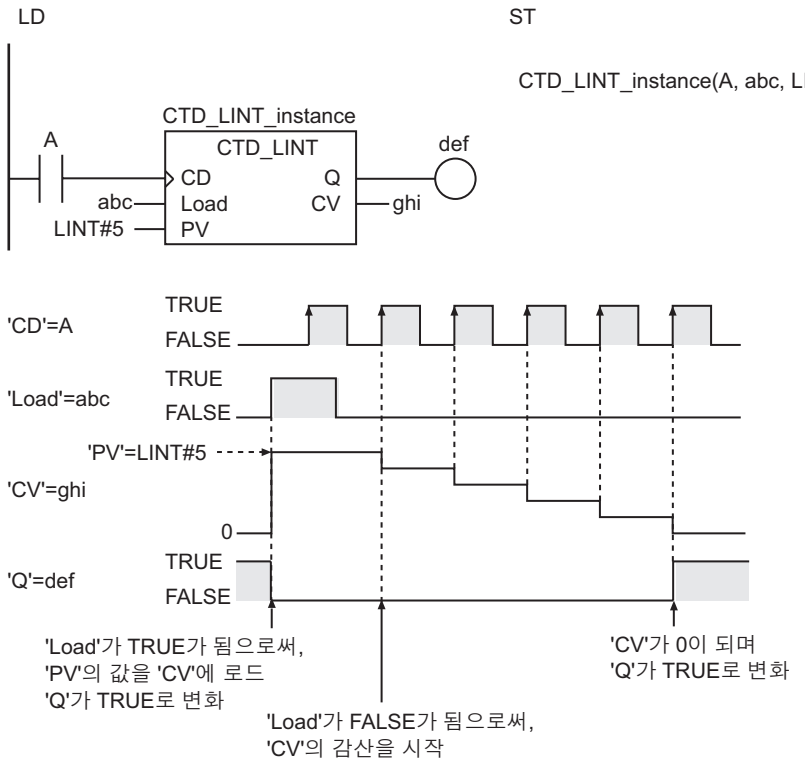
감산 카운터입니다.사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다. 명령 이름은 'PV'와 'CV'의 데이터 형식에 따라 다릅니다. 예를 들면, 양쪽이 LINT 형식인 경우, 명령 이름은 CTD_LINT가 됩니다.

로드 신호 'Load'가 TRUE로 설정되면, 사전 설정값 'PV'의 값을 카운터값 'CV'에 로드해서 카운터 출력 'Q'가 FALSE가 됩니다.
카운터 입력 신호 'CD'가 상승할 때 'CV'를 감산해 나갑니다. 'CV'의 값이 0 이하에 도달하면 'Q'의 값이 TRUE로 바뀝니다.

'CV'의 값이 0 이하가 되면, 그 이상 'CD'가 상승하더라도 'CV'는 변화하지 않습니다.

'Load'가 TRUE일 동안은 'CD'를 무시합니다. 'CV'의 감산은 수행하지 않습니다.

CTD_LINT 명령으로 'PV'=LINT#5인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

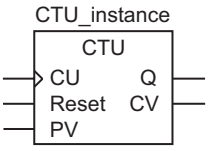
- 카운터 입력 신호가 입력될 때마다 가산되는 카운터가 필요한 경우에는 'CTU 명령(2-146페이지)'을 사용하십시오.
- 가산과 감산이 모두 가능한 카운터가 필요한 경우에는 'CTUD 명령(2-151페이지)'을 사용하십시오.

사용상의 주의 사항

- 카운트 다운 종료 후에 카운터를 재기동하려면 'Load'의 값을 일단 TRUE로 한 후 FALSE로 하십시오.
- 'PV'와 'CV'의 데이터 형식을 동일하게 하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Load'의 값이 TRUE가 된 시점에서 'PV'의 값을 'CV'에 로드합니다. 'CV'의 값이 0 이하이므로, 즉시 'Q'의 값이 TRUE가 됩니다. 그 후, 'CD'가 변화하더라도 'CV'의 감산은 수행하지 않습니다.
- 'CD'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CD'의 값이 TRUE라면 'CV'의 감산이 1회 수행됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

CTU

카운터 입력 신호가 입력될 때마다 가산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 INT 입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CTU	가산 카운터	FB		CTU_instance (CU, Reset, PV, Q, CV);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CU	카운터 입력	입력	카운터 입력	데이터 형식에 따름	–	FALSE
Reset*1	리셋 신호		TRUE : 'CV'를 0으로 리셋			
PV	사전 설정값		카운터의 사전 설정값	0~32767		0
Q	카운터 출력	출력	TRUE : 카운터 출력 ON FALSE : 카운터 출력 OFF	데이터 형식에 따름	–	–
CV	카운터값		카운터의 현재값	0~32767		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Reset을 R로 기술할 수 있습니다. 예를 들어 CTU_instance(CU:=A, R:=abc, PV:=INT#5, Q=>def, CV=>ghi);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	○																			
Reset	○																			
PV											○									
Q	○																			
CV											○									

기능

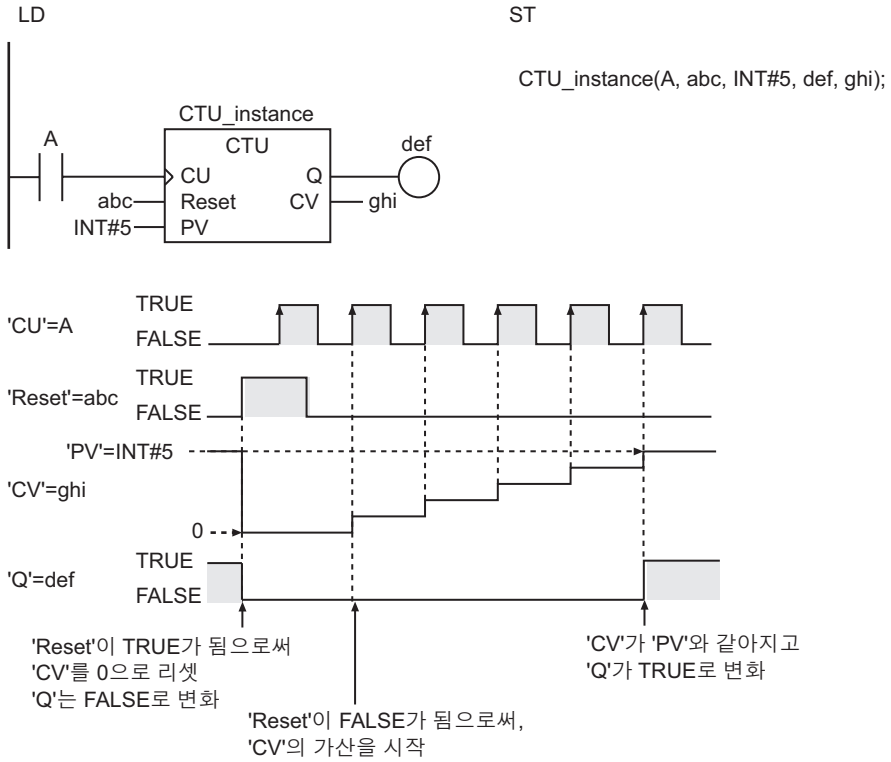
가산 카운터입니다.사전 설정값, 카운터값의 데이터 형식은 INT입니다.

리셋 신호 'Reset'이 TRUE로 설정되면, 카운터값 'CV'의 값이 0이 되어 카운터 출력 'Q'가 FALSE가 됩니다.
카운터 입력 신호 'CU'가 상승할 때 'CV'를 가산해 나갑니다. 'CV'의 값이 사전 설정값 'PV'의 값 이상이 되면 'Q'의 값이 TRUE로 바뀝니다.

'CV'의 값이 'PV'의 값 이상이 되면, 그 이상 'CU'가 입력되더라도 'CV'는 변화하지 않습니다.

'Reset'이 TRUE인 동안은 'CU'를 무시합니다. 'CV'의 가산은 수행하지 않습니다.

'PV'=INT#5인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

- 카운터 입력 신호가 입력될 때마다 감산되는 카운터가 필요한 경우에는 'CTD 명령(2-142페이지)'을 사용하십시오.
- 가산과 감산이 모두 가능한 카운터가 필요한 경우에는 'CTUD 명령(2-151페이지)'을 사용하십시오.

사용상의 주의 사항

- 카운트 업 종료 후에 카운터를 재기동하려면 'Reset'의 값을 일단 TRUE로 한 후 FALSE로 하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Reset'의 값이 TRUE가 된 시점에서 'CV'의 값은 0이 됩니다. 'CV'의 값이 'PV'의 값 이상이므로, 즉시 'Q'의 값이 TRUE가 됩니다. 그 후, 'CU'가 변화하더라도 'CV'의 가산은 수행하지 않습니다.
- 'Reset'의 값이 FALSE인 상태에서 'PV'의 값이 변경된 경우, 다음과 같이 동작합니다.

'PV'의 값	의미
그 시점에서의 'CV' 보다 큼	카운트 동작을 계속합니다.
그 시점에서의 'CV' 이하	카운트 동작을 종료합니다. 'Q'의 값이 TRUE가 됩니다. 'CV'의 값은 그 시점의 값을 유지한 채 변화하지 않습니다.

- 'CU'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CU'의 값이 TRUE라면 'CV'의 가산이 1회 수행됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

CTU_**

카운터 입력 신호가 입력될 때마다 가산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CTU_**	가산 카운터 그룹	FB	CTU_**_instance CTU_** CU Q Reset CV PV 단, **는 DINT, LINT, UDINT, ULINT 중 하나	CTU_**__instance (CU, Reset, PV, Q, CV); 단, **는 DINT, LINT, UDINT, ULINT 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CU	카운터 입력	입력	카운터 입력	데이터 형식에 따름	–	FALSE
Reset*1	리셋 신호		TRUE: 'CV'를 0으로 리셋			
PV	사전 설정값		카운터의 사전 설정값	데이터 형식에 따름 *2		0
Q	카운터 출력	출력	TRUE : 카운터 출력 ON FALSE : 카운터 출력 OFF	데이터 형식에 따름	–	–
CV	카운터값		카운터의 현재값	데이터 형식에 따름 *2		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Reset을 R로 기술할 수 있습니다. 예를 들어 CTU_LINT_instance(CU:=A, R:=abc, PV:=LINT#5, Q=>def, CV=>ghi);로 기술할 수 있습니다.

*2. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	○																			
Reset	○																			
PV								○	○			○	○							
Q	○																			
CV	'PV'와 같은 데이터 형식																			

기능

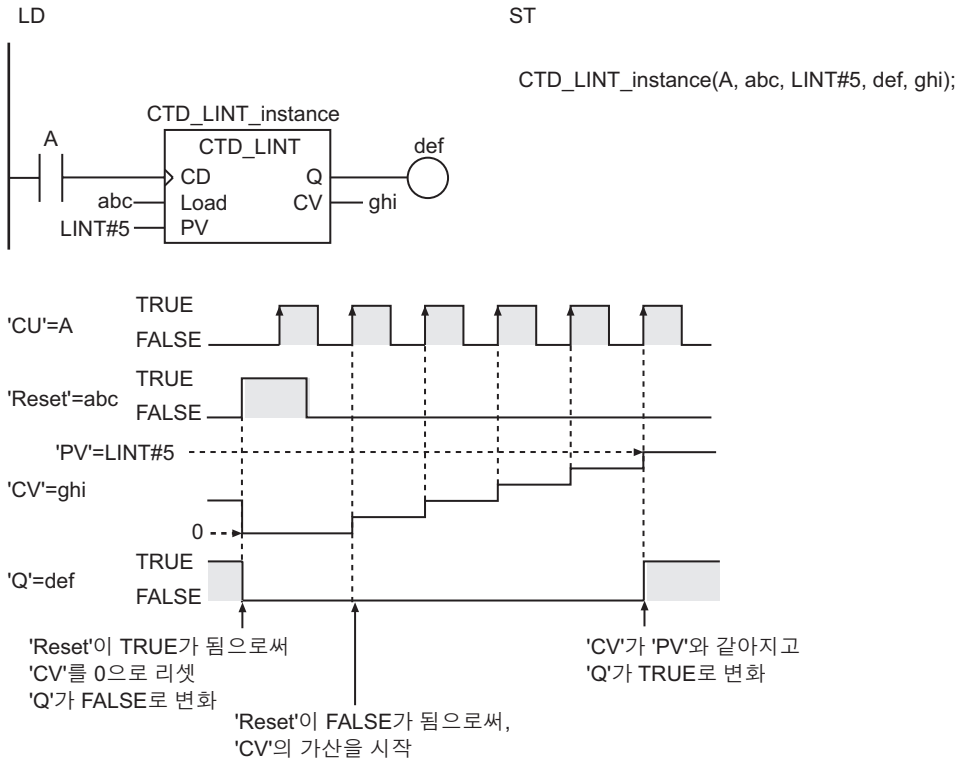
가산 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다. 명령 이름은 'PV'와 'CV'의 데이터 형식에 따라 다릅니다. 예를 들면, 양쪽이 LINT 형식인 경우, 명령 이름은 CTU_LINT가 됩니다.

리셋 신호 'Reset'이 TRUE로 설정되면, 카운터값 'CV'의 값이 0이 되어 카운터 출력 'Q'가 FALSE가 됩니다.
카운터 입력 신호 'CU'가 상승할 때 'CV'를 가산해 나갑니다. 'CV'의 값이 사전 설정값 'PV'의 값 이상이 되면 'Q'의 값이 TRUE로 바뀝니다.

'CV'의 값이 'PV'의 값 이상이 되면, 그 이상 'CU'가 입력되더라도 'CV'는 변화하지 않습니다.

'Reset'이 TRUE인 동안은 'CU'를 무시합니다. 'CV'의 가산은 수행하지 않습니다.

CTU_LINT 명령으로 'PV'=LINT#5인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

- 카운터 입력 신호가 입력될 때마다 감산되는 카운터가 필요한 경우에는 'CTD 명령(2-142페이지)'을 사용하십시오.
- 가산과 감산이 모두 가능한 카운터가 필요한 경우에는 'CTUD 명령(2-151페이지)'을 사용하십시오.

사용상의 주의 사항

- 카운트 업 종료 후에 카운터를 재기동하려면 'Reset'의 값을 일단 TRUE로 한 후 FALSE로 하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Reset'의 값이 TRUE가 된 시점에서 'CV'의 값은 0이 됩니다. 'CV'의 값이 'PV'의 값 이상이므로, 즉시 'Q'의 값이 TRUE가 됩니다. 그 후, 'CU'가 변화하더라도 'CV'의 가산은 수행하지 않습니다.
- 'PV'와 'CV'의 데이터 형식을 동일하게 하십시오.

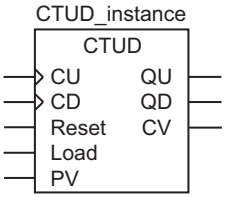
- 'Reset'의 값이 FALSE인 상태에서 'PV'의 값이 변경된 경우, 다음과 같이 동작합니다.

'PV'의 값	의미
그 시점에서의 'CV' 보다 큼	카운트 동작을 계속합니다.
그 시점에서의 'CV' 이하	카운트 동작을 종료합니다. 'Q'의 값이 TRUE가 됩니다. 'CV'의 값은 그 시점의 값을 유지한 채 변화하지 않습니다.

- 'CU'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CU'의 값이 TRUE라면 'CV'의 가산이 1회 수행됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Q'의 값은 FALSE가 됩니다.

CTUD

가산 카운터 입력 및 감산 카운터 입력에 따라 가감산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 INT입니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CTUD	가역 카운터	FB		CTUD_instance (CU, CD, Reset, Load, PV, QU, QD, CV);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CU	가산 카운터 입력	입력	가산 카운터 입력	데이터 형식에 따름	–	FALSE
CD	감산 카운터 입력		감산 카운터 입력			
Reset*1	리셋 신호		TRUE : 'CV'가 0으로 리셋			
Load*1	로드 신호		TRUE : 'CV'에 'PV'가 로드됨			
PV	사전 설정값		가산 카운터에 있어서의 카운트업 종료값 감산 카운터에 있어서의 초기값	0~32767		0
QU	가산 카운터 출력	출력	TRUE : 가산 카운터 출력 ON FALSE : 가산 카운터 출력 OFF	데이터 형식에 따름	–	–
QD	감산 카운터 출력		TRUE : 감산 카운터 출력 ON FALSE : 감산 카운터 출력 OFF			
CV	카운터값		카운터의 현재값	0~32767		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Reset을 R, Load를 LD로 기술할 수 있습니다. 예를 들어 CTUD_instance(CU:=A, CD:=B, R:=abc, LD:=def, PV:=INT#3, QU=>ghi, QD=>jkl, CV=>mno);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	○																			
CD	○																			
Reset	○																			
Load	○																			
PV											○									
QU	○																			
QD	○																			

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CV											○									

기능

가산 카운터 입력 신호 및 감산 카운터 입력 신호의 입력에 따라 가감산하는 카운터입니다.
가산 카운터와 감산 카운터 양쪽의 기능을 모두 갖고 있습니다.
사전 설정값, 카운터값의 데이터 형식은 INT입니다.

가산 카운터로서의 기능

리셋 신호 'Reset'이 TRUE로 설정되면, 카운터값 'CV'의 값이 0이 되어 가산 카운터 출력 'QU'가 FALSE가 됩니다.
가산 카운터 입력 신호 'CU'가 상승할 때 'CV'를 가산해 나갑니다. 'CV'의 값이 사전 설정값 'PV'의 값 이상이 되면 'QU'의 값이 TRUE로 바뀝니다.
'CV'의 값이 'PV'의 값 이상이 되면, 그 이상 'CU'가 입력되더라도 'CV'는 변화하지 않습니다.

감산 카운터로서의 기능

로드 신호 'Load'가 TRUE로 설정되면, 카운터값 'CV'에 사전 설정값 'PV'의 값이 로드되어, 감산 카운터 출력 'QD'가 FALSE가 됩니다.
감산 카운터 입력 신호 'CD'가 상승할 때 'CV'를 감산해 나갑니다. 'CV'의 값이 0 이하가 되면 'QD'의 값이 TRUE로 바뀝니다.
'CV'의 값이 0 이하가 되면, 그 이상 'CD'가 입력되더라도 'CV'는 변화하지 않습니다.

가산 카운터, 감산 카운터 공통의 기능

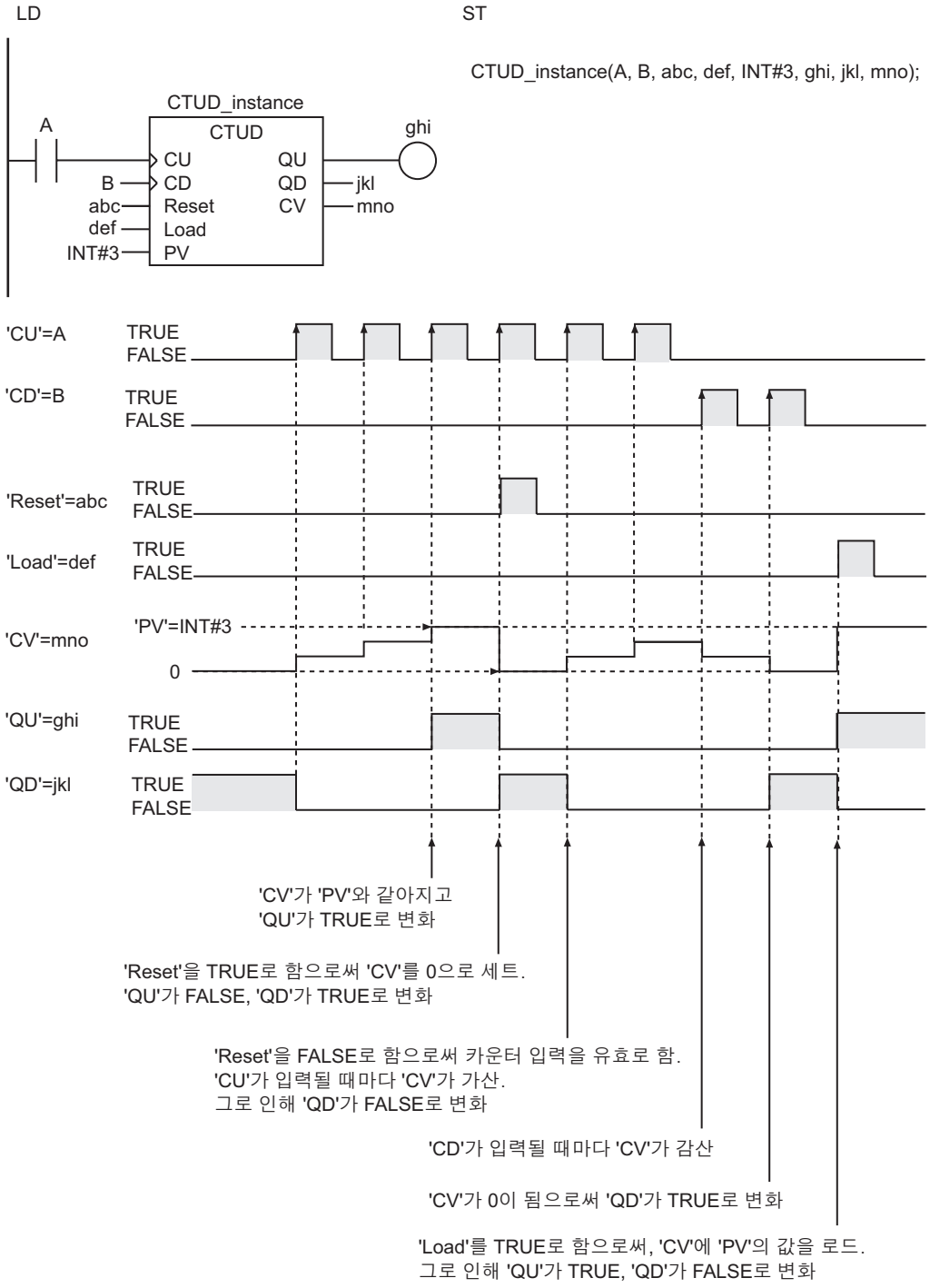
'Load' 또는 'Reset'이 TRUE인 동안은 'CU' 및 'CD'를 무시합니다. 'CV'의 가감산은 수행하지 않습니다.
'CU'와 'CD'가 동시에 상승한 경우, 'CV'는 변화하지 않습니다.
'Reset'과 'Load'가 모두 TRUE인 경우, 'Reset'이 우선시되어 'CV'의 값은 0이 됩니다.
'Reset'이 TRUE로 설정되더라도, 'CV'의 값이 0이 되므로 'QD'의 값은 TRUE가 됩니다.
'Load'가 TRUE로 설정되더라도, 'CV'의 값이 'PV'와 같아지므로 'QU'의 값은 TRUE가 됩니다.

'Reset', 'Load', 'CV', 'QU', 'QD'의 관계가 아래에 나와 있습니다. 단, 'PV'의 값은 0보다 큰 것으로 간주합니다.

'Reset'	'Load'	'CV'	'QU'	'QD'	동작
FALSE	FALSE	0 이하	FALSE	TRUE	가산 카운터로서만 동작합니다. • 'CV'는 'CU'가 상승하면 가산됩니다. 'CD'가 상승해도 감산되지 않습니다.
		0과 'PV' 사이	FALSE	FALSE	가산 카운터, 감산 카운터로서 동작합니다. • 'CV'는 'CU'가 상승하면 가산, 'CD'가 상승하면 감산됩니다.
		'PV' 이상	TRUE	FALSE	감산 카운터로서만 동작합니다. • 'CV'는 'CD'가 상승하면 감산됩니다. 'CU'가 상승해도 가산되지 않습니다.

'Reset'	'Load'	'CV'	'QU'	'QD'	동작
TRUE	FALSE	0	FALSE	TRUE	가산 카운터 리셋 상태입니다. • 'CV'의 값은 0으로 설정됩니다.
FALSE	TRUE	'PV'	TRUE	FALSE	감산 카운터 리셋 상태입니다. • 'CV'의 값은 'PV'로 설정됩니다.
TRUE	TRUE	0	FALSE	TRUE	가산 카운터 리셋 상태입니다. 'Load'보다 'Reset'이 우선시됩니다. • 'CV'의 값은 0으로 설정됩니다.

'PV'=INT#3인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

감산 카운터만으로, 또는 가산 카운터만으로 충분한 경우에는 'CTD 명령(2-142페이지)', 'CTU 명령(2-146페이지)'도 있습니다.

사용상의 주의 사항

- 가산 카운터를 리셋하기 위해 'Reset'의 값을 TRUE로 하면, 'QU'의 값이 FALSE가 됨과 동시에 'QD'의 값이 TRUE가 되는 것에 주의하십시오.
- 감산 카운터를 리셋하기 위해 'Load'의 값을 TRUE로 하면, 'QD'의 값이 FALSE가 됨과 동시에 'QU'의 값이 TRUE가 되는 것에 주의하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Load'의 값이 TRUE가 된 시점에서 'PV'의 값을 'CV'에 로드합니다. 'CV'의 값이 0 이하이므로, 즉시 'QD'의 값이 TRUE가 됩니다. 그 후, 'CD'가 변화하더라도 'CV'의 감산은 수행하지 않습니다.또한, 'Reset'의 값이 TRUE가 된 시점에서 'CV'의 값은 0이 됩니다. 'CV'의 값이 'PV'의 값 이상이므로, 즉시 'QU'의 값이 TRUE가 됩니다. 그 후, 'CU'가 변화하더라도 'CV'의 가산은 수행하지 않습니다.
- 이 명령 실행 중에 'PV'의 값을 변경할 수 있습니다. 그 경우, 변경 후의 'PV'의 값이 그 시점의 'CV' 미만이라면, 즉시 'QU'의 값이 TRUE가 됩니다.
- 'CU' 또는 'CD'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CU' 또는 'CD'의 값이 TRUE라면 'CV'의 가산 또는 감산이 1회 수행됩니다.

CTUD_**

가산 카운터 입력 및 감산 카운터 입력에 따라 가감산하는 카운터입니다. 사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CTUD_**	가역 카운터 그룹	FB	CTUD_**_instance CTUD_** CU QU CD QD Reset CV Load PV 단, **는 DINT, LINT, UDINT, ULINT 중 하나	CTUD_**_instance (CU, CD, Reset, Load, PV, QU, QD, CV); 단, *는 DINT, LINT, UDINT, ULINT 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
CU	가산 카운터 입력	입력	가산 카운터 입력	데이터 형식에 따름	–	FALSE
CD	감산 카운터 입력		감산 카운터 입력			
Reset*1	리셋 신호		TRUE : 'CV'가 0으로 리셋			
Load*1	로드 신호		TRUE : 'CV'에 'PV'가 로드됨			
PV	사전 설정값		가산 카운터에 있어서의 카운트업 종료값 감산 카운터에 있어서의 초기값	데이터 형식에 따름*2		0
QU	가산 카운터 출력	출력	TRUE : 가산 카운터 출력 ON FALSE : 가산 카운터 출력 OFF	데이터 형식에 따름	–	–
QD	감산 카운터 출력		TRUE : 감산 카운터 출력 ON FALSE : 감산 카운터 출력 OFF			
CV	카운터값		카운터의 현재값	데이터 형식에 따름*2		

*1. Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Reset을 R, Load를 LD로 기술할 수 있습니다. 예를 들어 CTUD_LINT_instance(CU:=A, CD:=B, R:=abc, LD:=def, PV:=LINT#3, QU=>ghi, QD=>jkl, CV=>mno);로 기술할 수 있습니다.

*2. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	○																			
CD	○																			
Reset	○																			

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Load	○																			
PV								○	○			○	○							
QU	○																			
QD	○																			
CV	'PV'와 같은 데이터 형식																			

기능

가산 카운터 입력 신호 및 감산 카운터 입력 신호의 입력에 따라 가감산하는 카운터입니다.
가산 카운터와 감산 카운터 양쪽의 기능을 모두 갖고 있습니다.
사전 설정값, 카운터값의 데이터 형식은 DINT, LINT, UDINT, ULINT 중 하나입니다.
명령 이름은 'PV'와 'CV'의 데이터 형식에 따라 다릅니다. 예를 들어 양쪽이 LINT 형식인 경우, 명령 이름은 CTUD_LINT가 됩니다.

가산 카운터로서의 기능

리셋 신호 'Reset'이 TRUE로 설정되면, 카운터값 'CV'의 값이 0이 되어 가산 카운터 출력 'QU'가 FALSE가 됩니다.
가산 카운터 입력 신호 'CU'가 상승할 때 'CV'를 가산해 나갑니다. 'CV'의 값이 사전 설정값 'PV'의 값 이상이 되면 'QU'의 값이 TRUE로 바뀝니다.
'CV'의 값이 'PV'의 값 이상이 되면, 그 이상 'CU'가 입력되더라도 'CV'는 변화하지 않습니다.

감산 카운터로서의 기능

로드 신호 'Load'가 TRUE로 설정되면, 카운터값 'CV'에 사전 설정값 'PV'의 값이 로드되어, 감산 카운터 출력 'QD'가 FALSE가 됩니다.
감산 카운터 입력 신호 'CD'가 상승할 때 'CV'를 감산해 나갑니다. 'CV'의 값이 0 이하가 되면 'QD'의 값이 TRUE로 바뀝니다.
'CV'의 값이 0 이하가 되면, 그 이상 'CD'가 입력되더라도 'CV'는 변화하지 않습니다.

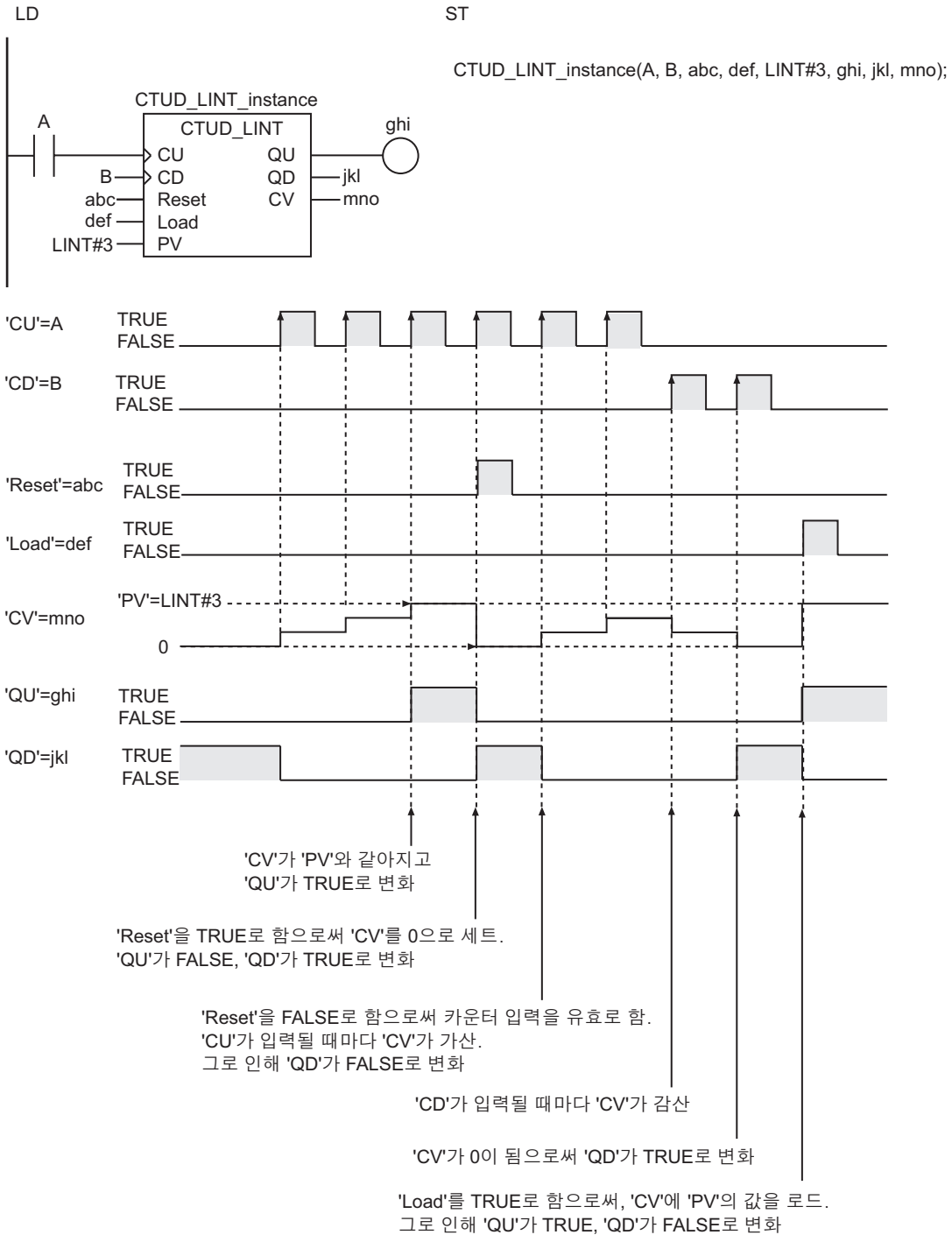
가산 카운터, 감산 카운터 공통의 기능

'Load' 또는 'Reset'이 TRUE인 동안은 'CU' 및 'CD'를 무시합니다. 'CV'의 가감산은 수행하지 않습니다.
'CU'와 'CD'가 동시에 상승한 경우, 'CV'는 변화하지 않습니다.
'Reset'과 'Load'가 모두 TRUE인 경우, 'Reset'이 우선시되어 'CV'의 값은 0이 됩니다.
'Reset'이 TRUE로 설정되더라도, 'CV'의 값이 0이 되므로 'QD'의 값은 TRUE가 됩니다.
'Load'가 TRUE로 설정되더라도, 'CV'의 값이 'PV'와 같아지므로 'QU'의 값은 TRUE가 됩니다.

'Reset', 'Load', 'CV', 'QU', 'QD'의 관계가 아래에 나와 있습니다. 단, 'PV'의 값은 0보다 큰 것으로 간주합니다.

'Reset'	'Load'	'CV'	'QU'	'QD'	동작
FALSE	FALSE	0 이하	FALSE	TRUE	가산 카운터로서만 동작합니다. • 'CV'는 'CU'가 상승하면 가산됩니다. 'CD'가 상승해도 감산되지 않습니다.
		0과 'PV' 사이	FALSE	FALSE	가산 카운터, 감산 카운터로서 동작합니다. • 'CV'는 'CU'가 상승하면 가산, 'CD'가 상승하면 감산됩니다.
		'PV' 이상	TRUE	FALSE	감산 카운터로서만 동작합니다. • 'CV'는 'CD'가 상승하면 감산됩니다. 'CU'가 상승해도 가산되지 않습니다.
TRUE	FALSE	0	FALSE	TRUE	가산 카운터 리셋 상태입니다. • 'CV'의 값은 0으로 설정됩니다.
FALSE	TRUE	'PV'	TRUE	FALSE	감산 카운터 리셋 상태입니다. • 'CV'의 값은 'PV'로 설정됩니다.
TRUE	TRUE	0	FALSE	TRUE	가산 카운터 리셋 상태입니다. 'Load'보다 'Reset'이 우선시됩니다. • 'CV'의 값은 0으로 설정됩니다.

CTUD_LINT 명령에서 'PV'=LINT#3인 경우의 기술 예와 타이밍 차트가 아래에 나와 있습니다.



참고

감산 카운터만으로, 또는 가산 카운터만으로 충분한 경우에는 📖 'CTD_** 명령(2-144페이지)', 📖 'CTU_** 명령(2-148페이지)'도 있습니다.

사용상의 주의 사항

- 가산 카운터를 리셋하기 위해 'Reset'의 값을 TRUE로 하면, 'QU'의 값이 FALSE가 됨과 동시에 'QD'의 값이 TRUE가 되는 것에 주의하십시오.

- 감산 카운터를 리셋하기 위해 'Load'의 값을 TRUE로 하면, 'QD'의 값이 FALSE가 됨과 동시에 'QU'의 값이 TRUE가 되는 것에 주의하십시오.
- 'PV'에 음수가 설정된 경우에도, 'Load'의 값이 TRUE가 된 시점에서 'PV'의 값을 'CV'에 로드합니다. 'CV'의 값이 0 이하이므로, 즉시 'QD'의 값이 TRUE가 됩니다. 그 후, 'CD'가 변화하더라도 'CV'의 감산은 수행하지 않습니다. 또한, 'Reset'의 값이 TRUE가 된 시점에서 'CV'의 값은 0이 됩니다. 'CV'의 값이 'PV'의 값 이상이므로, 즉시 'QU'의 값이 TRUE가 됩니다. 그 후, 'CU'가 변화하더라도 'CV'의 가산은 수행하지 않습니다.
- 이 명령 실행 중에 'PV'의 값을 변경할 수 있습니다. 그 경우, 변경 후의 'PV'의 값이 그 시점의 'CV' 미만이라면, 즉시 'QU'의 값이 TRUE가 됩니다.
- 'PV'와 'CV'의 데이터 형식을 동일하게 하십시오.
- 'CU' 또는 'CD'의 값이 FALSE인 상태에서 전원 OFF가 발생하거나 동작 모드를 프로그램으로 설정한 경우, 이 명령의 실행을 재개했을 때 'CU' 또는 'CD'의 값이 TRUE라면 'CV'의 가산 또는 감산이 1회 수행됩니다.

산술 명령

명령	명칭	페이지	명령	명칭	페이지
ADD, +	가산	2-162	EXP	자연 지수 연산	2-201
AddOU, +OU	가산(오버 플로우 확인 있음)	2-165	EXPT, **	거듭제곱 연산	2-203
SUB, -	감산	2-168	Inc/Dec	증가/ 감소	2-205
SubOU, -OU	감산(오버 플로우 확인 있음)	2-171	Rand	난수 발생	2-207
MUL, *	승산	2-174	AryAdd	배열 일괄 가산	2-209
MulOU, *OU	승산(오버 플로우 확인 있음)	2-177	AryAddV	배열 요소 가산	2-211
DIV, /	제산	2-180	ArySub	배열 일괄 감산	2-213
MOD	나머지	2-183	ArySubV	배열 요소 감산	2-215
ABS	절대값	2-185	AryMean	배열 요소의 평균값 산출	2-217
RadToDeg/ DegToRad	라디안→각도 변환/ 각도→라디안 변환	2-187	ArySD	배열 요소의 표준 편차	2-219
SIN/COS/TAN	SIN 연산/COS 연산/TAN 연산	2-189	ModReal	실수 나머지	2-221
ASIN/ACOS/ATAN	SIN^{-1}연산/COS^{-1}연산/TAN^{-1} 연산	2-192	Fraction	실수 소수부 추출	2-223
SQRT	제곱근 연산	2-196	CheckReal	실수 확인	2-225
LN/LOG	자연 로그 연산/상용 로그 연산	2-198			

ADD,+

정수, 실수를 가산합니다. 또는 문자열을 결합합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ADD, +	가산	FUN	(@)ADD EN ENO In1 Out : : InN (@)+ EN ENO In1 Out : : InN	Out:=In1 + ⋯ + InN;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	가산값	입력	가산하는 수 N은 2~5	데이터 형식에 따름	–	0(*)
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

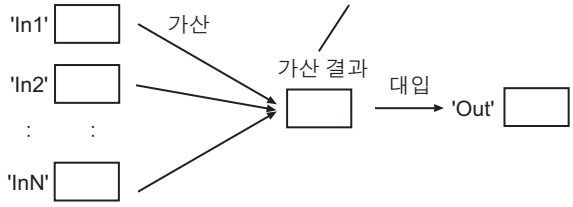
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN						○	○	○	○	○	○	○	○	○	○					○
Out						○	○	○	○	○	○	○	○	○	○					○

기능

2개 이상 5개 이하의 정수 또는 실수의 가산 결과를 출력값 'Out'에 출력합니다.

가산값 'In1'~'InN'은 상이한 데이터 형식이라도 상관없습니다.
그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 가산 결과는 DINT 형식이 됩니다.

'In1'~'InN'을 모두 포함하는 데이터 형식
(예) INT 간의 가산인 경우는 INT
INT와 DINT의 가산인 경우는 DINT



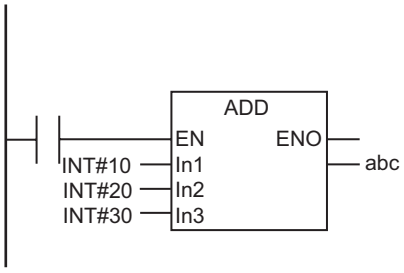
'In1'~'InN'의 합이 가산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
오버 플로우가 발생한 경우의 'In1'~'InN'의 데이터 형식, 가산 결과의 데이터 형식, 가산 결과의 값 간의 관계는 다음과 같습니다.

'In1'~'InN'의 데이터 형식	가산 결과의 데이터 형식	가산 결과의 값
모두 정수 형식	정수 형식	'In1'~'InN'의 합 중, 가산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)
실수 형식이 있는 경우	실수 형식	± ∞ (*2)

*1 예를 들어, 'In1'의 값이 INT#32767, 'In2'의 값이 INT#3인 경우, 가산 결과의 데이터 형식은 INT 형식이 됩니다.
가산 결과의 값은 양쪽의 합인 32770의 하위 16비트로, INT#-32766이 됩니다.
*2 'In1'~'InN'의 합이 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=INT#10, 'In2'=INT#20, 'In3'=INT#30인 경우의 예가 아래에 나와 있습니다. INT 형식 변수 abc의 값은 INT#60이 됩니다.

LD



ST

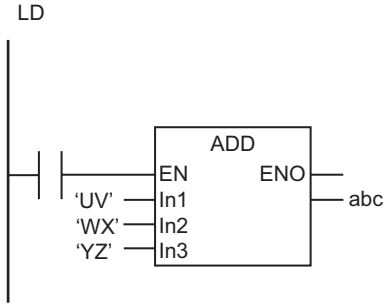
```
abc:=INT#10+INT#20+INT#30;
```

'In1'~'InN'을 가산합니다.
10＋20＋30＝60이므로 abc의 값은 INT#60이 됩니다.

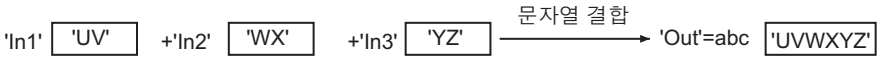


'In1'~'InN'이 STRING 형식인 경우, 그 문자열들을 결합합니다.
단, 'In1'~'InN'에 STRING 형식을 지정할 수 있는 것은 래더도 프로그램의 경우뿐입니다.

'In1'='UV', 'In2'='WX', 'In3'='YZ'인 경우의 예가 아래에 나와 있습니다. STRING 형식 변수 abc의 값은 'UVWXYZ'가 됩니다.



'In1'~'InN'을 문자열 결합합니다.
'UV'＋'WX'＋'YZ'＝'UVWXYZ'이므로 abc의 값은 'UVWXYZ'가 됩니다.



ADD 명령과 + 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

- 실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.
- ST 프로그램에서 문자열 결합을 수행하는 경우, 'CONCAT 명령(2-526페이지)'을 사용하십시오.

사용상의 주의 사항

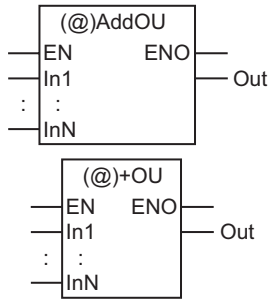
- 'Out'은 가산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 가산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 문자열의 결합인 경우, 'In1'~'InN'과 'Out'을 모두 STRING 형식으로 하십시오.
- 가산에 의해 오버 플로우 또는 언더 플로우가 발생해도 이상 상태가 되지 않습니다.
- 가산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.
- 실수의 +∞/-∞에 대한 가산 결과의 값은 다음과 같습니다.

가산 내용	가산 결과의 값
+∞와 수치의 가산	+∞
-∞와 수치의 가산	-∞
+∞와 +∞의 가산	+∞
-∞와 -∞의 가산	-∞
+∞와 -∞의 가산	비수

- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 가산 결과의 값은 비수가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 문자열의 결합에서, 결합한 문자열의 크기가 1986바이트를 초과하는 경우.

AddOU, +OU

정수, 실수를 가산합니다. 또한 결과에 대해 오버 플로우 확인을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AddOU, +OU	가산 (오버 플로우 확인 있음)	FUN		Out:=AddOU(In1, • • •, InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	가산값	입력	가산하는 수 N은 2~5	데이터 형식에 따름	–	0(*)
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN						○	○	○	○	○	○	○	○	○ (*)	○ (*)					
Out						○	○	○	○	○	○	○	○	○	○					

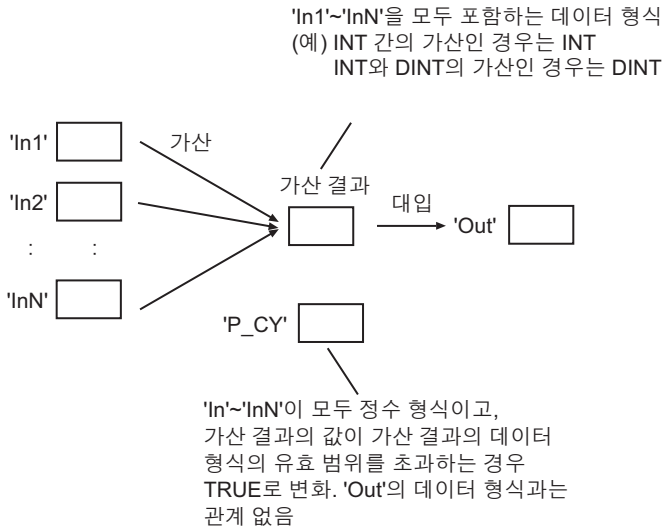
* 'In1'~'InN' 중 하나의 데이터 형식이 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다.

기능

2개 이상 5개 이하의 정수 또는 실수의 가산 결과를 출력값 'Out'에 출력합니다.

가산값 'In1'~'InN'은 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 가산 결과는 DINT 형식이 됩니다.

'In1'~'InN'의 합이 가산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
'In1'~'InN'이 모두 정수 형식인 경우 오버 플로우가 발생하면 시스템 정의 변수 'P_CY'(캐리 플래그)의 값이 TRUE가 됩니다.
'In1'~'InN' 중 하나가 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다. 따라서 'P_CY'의 값은 변화하지 않습니다.



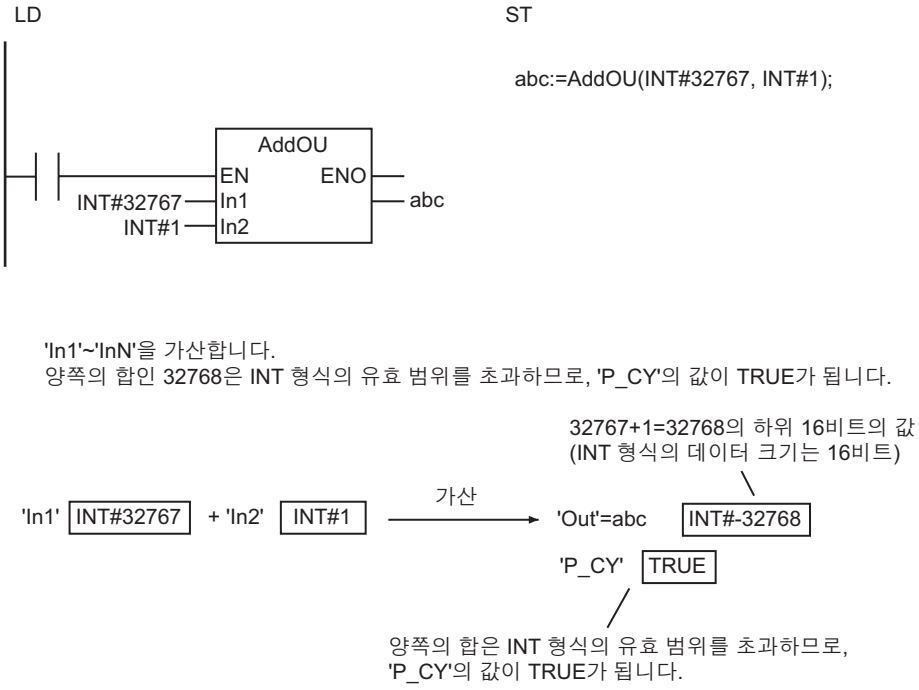
오버 플로우가 발생한 경우의 'In1'~'InN'의 데이터 형식, 가산 결과의 데이터 형식, 가산 결과의 값, 'P_CY'의 값 간의 관계는 다음과 같습니다.

'In1'~'InN'의 데이터 형식	가산 결과의 데이터 형식	가산 결과의 값	'P_CY'의 값
모두 정수 형식	정수 형식	'In1'~'InN'의 합 중, 가산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)	TRUE
실수 형식이 있는 경우	실수 형식	± ∞ (*2)	변화 없음

*1 예를 들어, 'In1'의 값이 INT#32767, 'In2'의 값이 INT#3인 경우, 가산 결과의 데이터 형식은 INT 형식이 됩니다. 가산 결과의 값은 양쪽의 합인 32770의 하위 16비트로, INT#-32766이 됩니다.

*2 'In1'~'InN'의 합이 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=INT#32767, 'In2'=INT#1이며 변수 abc가 INT 형식인 경우의 예가 아래에 나와 있습니다. 'In1'과 'In2'는 모두 INT 형식이므로 가산 결과의 데이터 형식은 INT 형식이 됩니다. 양쪽의 합인 32768은 INT 형식의 유효 범위를 초과하므로, 'P_CY'의 값이 TRUE가 됩니다. 또한 INT 형식 변수 abc의 값은 INT#-32768이 됩니다(32768의 하위 16비트의 값).



AddOU 명령과 +OU 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	TRUE : 정수 연산에서 오버 플로우 있음 FALSE : 정수 연산에서 오버 플로우 없음

참고

- 'Out'이 실수 형식인 경우, 값이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.
- 오버 플로우를 확인할 필요가 없는 경우에는 'ADD,+ 명령(2-162페이지)'을 사용하십시오. 그래야 처리 시간이 짧아집니다.

사용상의 주의 사항

- 'Out'은 가산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 가산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 가산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.
- 실수의 +∞/-∞에 대한 가산 결과의 값은 다음과 같습니다.

가산 내용	가산 결과의 값
+∞와 수치의 가산	+∞
-∞와 수치의 가산	-∞
+∞와 +∞의 가산	+∞
-∞와 -∞의 가산	-∞
+∞와 -∞의 가산	비수

- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 가산 결과의 값은 비수가 됩니다.

SUB, -

정수, 실수를 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB, -	감산	FUN	(@)SUB EN ENO In1 Out In2 (@)- EN ENO In1 Out In2	Out:=In1 - In2;

변수

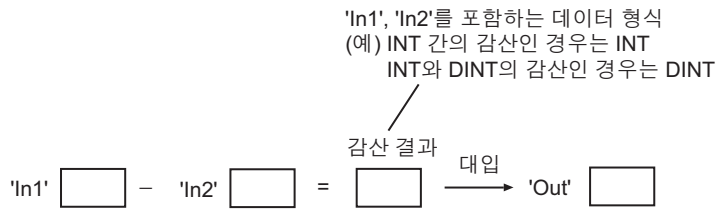
	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산값	입력	피감산값	데이터 형식에 따름	–	0(*)
In2	감산값		감산값			
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1						○	○	○	○	○	○	○	○	○	○					
In2						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

피감산값 'In1'에서 감산값 'In2'를 감산한 결과를 출력값 'Out'에 출력합니다.
'In1'과 'In2'는 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 감산 결과는 DINT 형식이 됩니다.



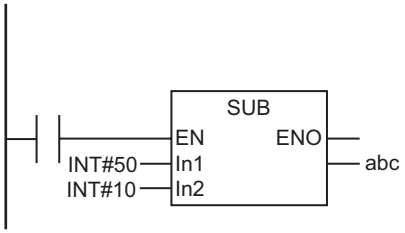
'In1'~'In2'의 차가 감산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
오버 플로우가 발생한 경우의 'In1', 'In2'의 데이터 형식, 감산 결과의 데이터 형식, 감산 결과의 값 간의 관계는 다음과 같습니다.

'In1', 'In2'의 데이터 형식	감산 결과의 데이터 형식	감산 결과의 값
모두 정수 형식	정수 형식	'In1'과 'In2'의 차 중, 감산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)
실수 형식이 있는 경우	실수 형식	± ∞ (*2)

*1 예를 들어, 'In1'의 값이 INT#32767, 'In2'의 값이 INT#-3인 경우, 감산 결과의 데이터 형식은 INT 형식이 됩니다. 감산 결과의 값은 양쪽의 차인 32770의 하위 16비트로, INT#-32766이 됩니다.
*2 'In1'과 'In2'의 차가 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=INT#50, 'In2'=INT#10인 경우의 예가 아래에 나와 있습니다. INT 형식 변수 abc의 값은 INT#40이 됩니다.

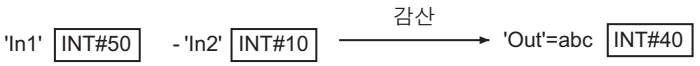
LD



ST

```
abc:=INT#50-INT#10;
```

'In1'에서 'In2'를 감산합니다.
50-10＝40이므로 abc의 값은 INT#40이 됩니다.



SUB 명령과 - 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Out'은 감산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 감산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 감산에 의해 오버 플로우 또는 언더 플로우가 발생해도 이상 상태가 되지 않습니다.
- 감산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.

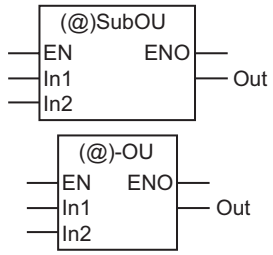
- 실수의 +∞/-∞에 대한 감산 결과는 다음과 같습니다.

감산 내용	감산 결과
(+∞) - (수치)	+∞
(수치) - (+∞)	-∞
(-∞) - (수치)	-∞
(수치) - (-∞)	+∞
(+∞) - (+∞)	비수
(+∞) - (-∞)	+∞
(-∞) - (+∞)	-∞
(-∞) - (-∞)	비수

- 'In1', 'In2' 중 어느 것의 값이 비수인 경우, 감산 결과의 값은 비수가 됩니다.

SubOU, -OU

정수, 실수를 감산합니다. 또한 감산 결과에 대해 오버 플로우 확인을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SubOU, -OU	감산 (오버 플로우 확인 있음)	FUN		Out:=SubOU(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산값	입력	피감산값	데이터 형식에 따름	–	0(*)
In2	감산값		감산값			
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1						○	○	○	○	○	○	○	○	○ (*)	○ (*)					
In2						○	○	○	○	○	○	○	○	○ (*)	○ (*)					
Out						○	○	○	○	○	○	○	○	○	○					

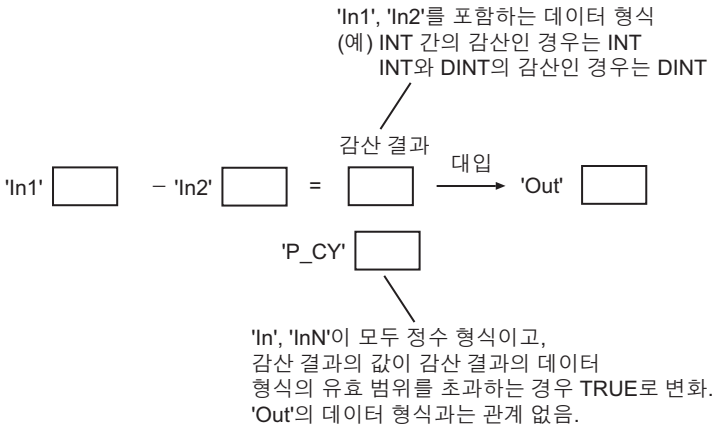
* 'In1', 'In2' 중 하나의 데이터 형식이 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다.

기능

피감산값 'In1'에서 감산값 'In2'를 감산한 결과를 출력값 'Out'에 출력합니다.

'In1'과 'In2'는 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 감산 결과는 DINT 형식이 됩니다.

'In1'~'In2'의 차가 감산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
'In1'과 'In2'가 모두 정수 형식인 경우 오버 플로우가 발생하면 시스템 정의 변수 'P_CY'(캐리 플래그)의 값이 TRUE가 됩니다.
'In1', 'In2' 중 하나가 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다. 따라서 'P_CY'의 값은 변화하지 않습니다.



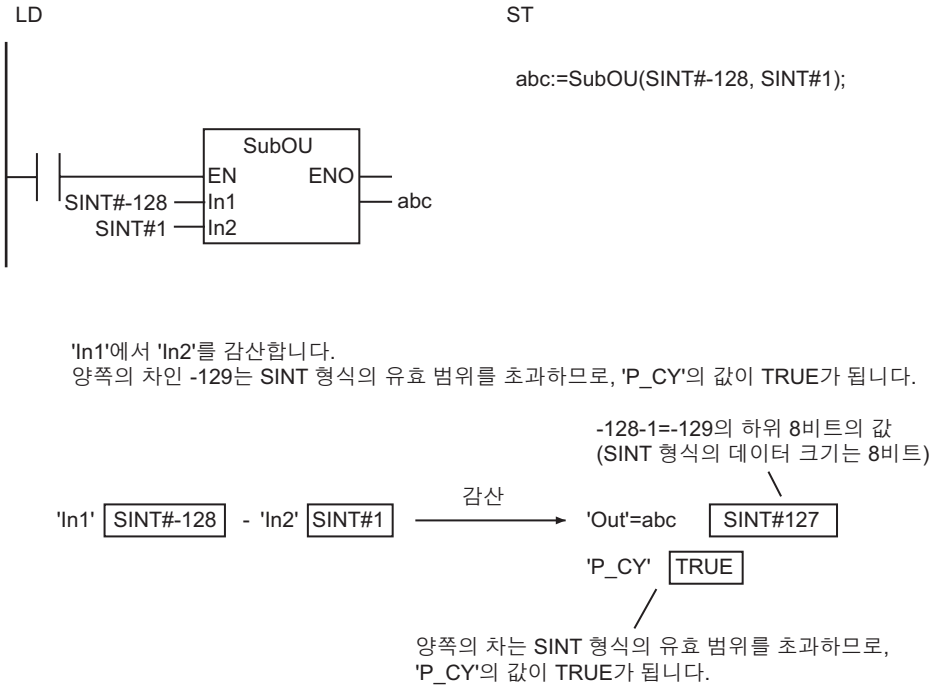
오버 플로우가 발생한 경우의 'In1', 'In2'의 데이터 형식, 감산 결과의 데이터 형식, 감산 결과의 값, 'P_CY'의 값 간의 관계는 다음과 같습니다.

'In1', 'In2'의 데이터 형식	감산 결과의 데이터 형식	감산 결과의 값	'P_CY'의 값
모두 정수 형식	정수 형식	'In1'과 'In2'의 차 중, 감산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)	TRUE
실수 형식이 있는 경우	실수 형식	± ∞ (*2)	변화 없음

*1 예를 들어, 'In1'의 값이 INT#32767, 'In2'의 값이 INT#-3인 경우, 감산 결과의 데이터 형식은 INT 형식이 됩니다. 감산 결과의 값은 양쪽의 차인 32770의 하위 16비트로, INT#-32766이 됩니다.

*2 'In1'과 'In2'의 차가 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=SINT#-128, 'In2'=SINT#1일 때 변수 abc가 SINT 형식인 경우의 예가 아래에 나와 있습니다.
'In1'과 'In2'는 모두 SINT 형식이므로 감산 결과의 데이터 형식은 SINT 형식이 됩니다. 양쪽의 차인 -129는 SINT 형식의 유효 범위를 초과하므로, 'P_CY'의 값이 TRUE가 됩니다. 또한 SINT 형식 변수 abc의 값은 SINT#127이 됩니다(-129의 하위 8비트의 값).



SubOU 명령과 -OU 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	TRUE : 정수 연산에서 오버 플로우 있음 FALSE : 정수 연산에서 오버 플로우 없음

참고

- 실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.
- 오버 플로우를 확인할 필요가 없는 경우에는 'SUB, - 명령(2-168페이지)'을 사용하십시오. 그래야 처리 시간이 짧아집니다.

사용상의 주의 사항

- 'Out'은 감산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 감산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 감산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.
- 실수의 +∞/-∞에 대한 감산 결과는 다음과 같습니다.

감산 내용	감산 결과
(+∞) - (수치)	+∞
(수치) - (+∞)	-∞
(-∞) - (수치)	-∞
(수치) - (-∞)	+∞
(+∞) - (+∞)	비수
(+∞) - (-∞)	+∞
(-∞) - (+∞)	-∞
(-∞) - (-∞)	비수

- 'In1', 'In2' 중 어느 것의 값이 비수인 경우, 감산 결과의 값은 비수가 됩니다.

MUL, *

정수, 실수를 승산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MUL, *	승산	FUN	(@)MUL EN ENO In1 Out : : InN (@)* EN ENO In1 Out : : InN	Out:=In1 * … * InN;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	승산값	입력	승산하는 수 N은 2~5	데이터 형식에 따름	–	1(*)
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

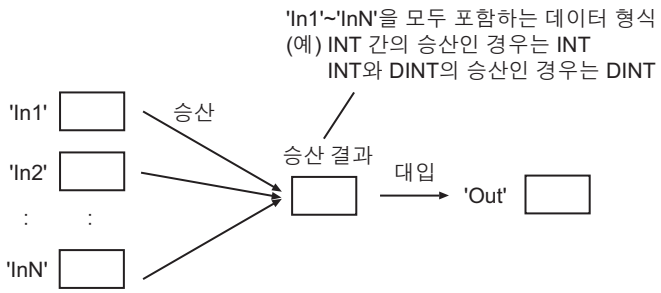
* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

2개 이상 5개 이하의 정수 또는 실수의 승산 결과를 출력값 'Out'에 출력합니다.

승산값 'In1'~'InN'은 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 승산 결과는 DINT 형식이 됩니다.

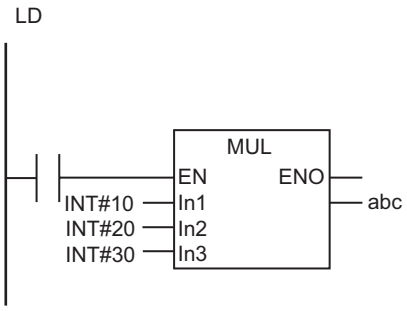


'In1'~'InN'의 곱이 승산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
오버 플로우가 발생한 경우의 'In1'~'InN'의 데이터 형식, 승산 결과의 데이터 형식, 승산 결과의 값 간의 관계는 다음과 같습니다.

'In1'~'InN'의 데이터 형식	승산 결과의 데이터 형식	승산 결과의 값
모두 정수 형식	정수 형식	'In1'~'InN'의 곱 중, 승산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)
실수 형식이 있는 경우	실수 형식	± ∞ (*2)

*1 예를 들어, 'In1'의 값이 INT#16384, 'In2'의 값이 INT#2인 경우, 승산 결과의 데이터 형식은 INT 형식이 됩니다. 승산 결과의 값은 양쪽의 곱인 32768의 하위 16비트로, INT#-32768이 됩니다.
*2 'In1'~'InN'의 곱이 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=INT#10, 'In2'=INT#20, 'In3'=INT#30인 경우의 예가 아래에 나와 있습니다. INT 형식 변수 abc의 값은 INT#6000이 됩니다.



ST

```
abc:=INT#10*INT#20*INT#30;
```

'In1'~'InN'을 승산합니다.
10*20*30＝6000이므로 abc의 값은 INT#6000이 됩니다.



MUL 명령과 * 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 🕮 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

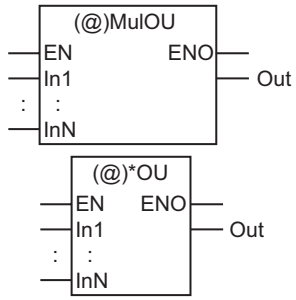
- 'Out'은 승산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 승산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 승산에 의해 오버 플로우 또는 언더 플로우가 발생해도 이상 상태가 되지 않습니다.
- 승산에 의해 오버 플로우 또는 언더 플로우가 발생하는 경우, 래더도와 ST에서 결과가 다른 경우가 있습니다. 래더도는 입력 변수 데이터 형식의 범위에서 연산하는 것에 비해, ST는 숫자의 정밀도를 높여서 연산합니다.
- 승산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.
- 실수의 +∞/-∞에 대한 승산 결과는 다음과 같습니다.

승산 내용	승산 결과
+∞와 정수를 승산	+∞
+∞와 음수를 승산	-∞
-+∞와 정수를 승산	-∞
-+∞와 음수를 승산	+∞
+∞와 +∞의 승산	+∞
-∞와 -∞의 승산	+∞
+∞와 -∞의 승산	-∞
+∞와 0의 승산	비수
-∞와 0의 승산	비수

- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 승산 결과의 값은 비수가 됩니다.

MulOU, *OU

정수, 실수를 승산한 결과를 출력합니다. 또한 정수의 승산 결과에 대해 오버 플로우 확인을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MulOU, *OU	승산 (오버 플로우 확인 있음)	FUN		Out:=MulOU(In1, •••, InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	승산값	입력	승산하는 수 N은 2~5	데이터 형식에 따름	–	1(*)
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN						○	○	○	○	○	○	○	○	○ (*)	○ (*)					
Out						○	○	○	○	○	○	○	○	○	○					

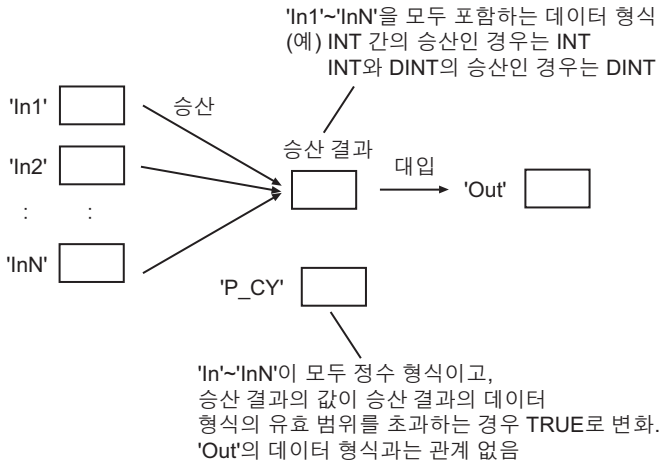
* 'In1'~'InN' 중 하나의 데이터 형식이 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다.

기능

2개 이상 5개 이하의 정수 또는 실수의 승산 결과를 출력값 'Out'에 출력합니다.

승산값 'In1'~'InN'은 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 승산 결과는 DINT 형식이 됩니다.

'In1'~'InN'의 곱이 승산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
'In1'~'InN'이 모두 정수 형식인 경우 오버 플로우가 발생하면 시스템 정의 변수 'P_CY'(캐리 플래그)의 값이 TRUE가 됩니다.
'In1'~'InN' 중 하나가 실수 형식인 경우, 오버 플로우 확인을 수행하지 않습니다. 따라서 'P_CY'의 값은 변화하지 않습니다.



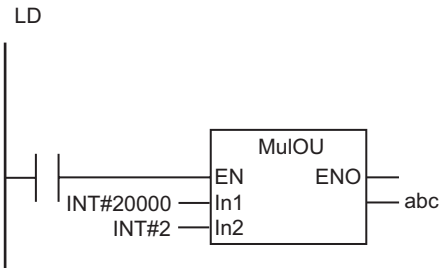
오버 플로우가 발생한 경우의 'In1'~'InN'의 데이터 형식, 승산 결과의 데이터 형식, 승산 결과의 값, 'P_CY'의 값 간의 관계는 다음과 같습니다.

'In1'~'InN'의 데이터 형식	승산 결과의 데이터 형식	승산 결과의 값	'P_CY'의 값
모두 정수 형식	정수 형식	'In1'~'InN'의 곱 중, 승산 결과의 데이터 형식의 비트 수로 표현할 수 있는 값(*1)	TRUE
실수 형식이 있는 경우	실수 형식	± ∞ (*2)	변화 없음

*1 예를 들어, 'In1'의 값이 INT#16384, 'In2'의 값이 INT#2인 경우, 승산 결과의 데이터 형식은 INT 형식이 됩니다. 승산 결과의 값은 양쪽의 곱인 32768의 하위 16비트로, INT#-32768이 됩니다.

*2 'In1'~'InN'의 곱이 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

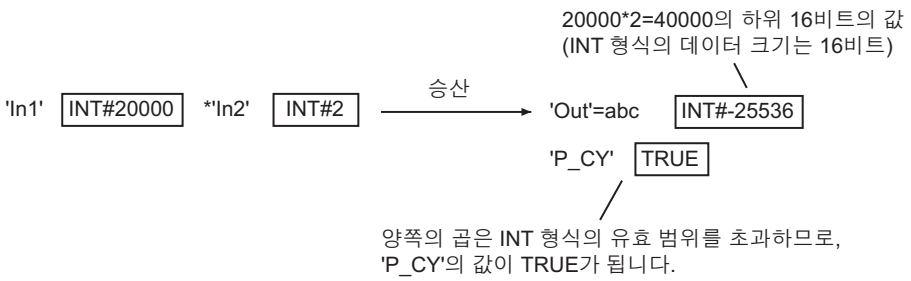
'In1'=INT#20000, 'In2'=INT#2이며 변수 abc가 INT 형식인 경우의 예가 아래에 나와 있습니다.
'In1'과 'In2'는 모두 INT 형식이므로 가산 결과의 데이터 형식은 INT 형식이 됩니다. 양쪽의 곱인 40000은 INT 형식의 유효 범위를 초과하므로, 'P_CY'의 값이 TRUE가 됩니다. 또한 INT 형식 변수 abc의 값은 INT#-25536이 됩니다(40000의 하위 16비트의 값).




```
ST

abc:=MulOU(INT#20000, INT#2);
```

'In1'~'InN'을 승산합니다.
양쪽의 곱인 40000은 INT 형식의 유효 범위를 초과하므로, 'P_CY'의 값이 TRUE가 됩니다.



MulOU 명령과 *OU 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	TRUE : 정수 연산에서 오버 플로우 있음 FALSE : 정수 연산에서 오버 플로우 없음

참고

- 실수를 연산하는 경우 'Out'이 +∞, -∞, 비수인지에 대한 판정에는 'CheckReal 명령(2-225페이지)'을 사용하십시오.
- 오버 플로우를 확인할 필요가 없는 경우에는 'MUL, * 명령(2-174페이지)'을 사용하십시오. 그래야 처리 시간이 짧아집니다.

사용상의 주의 사항

- 'Out'은 승산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 승산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 승산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.
- 실수의 +∞/-∞에 대한 승산 결과는 다음과 같습니다.

승산 내용	승산 결과
+∞와 정수를 승산	+∞
+∞와 음수를 승산	-∞
-∞와 정수를 승산	-∞
-∞와 음수를 승산	+∞
+∞와 +∞의 승산	+∞
-∞와 -∞의 승산	+∞
+∞와 -∞의 승산	-∞
+∞와 0의 승산	비수
-∞와 0의 승산	비수

- 'In1'~'InN' 중 어느 것의 값이 비수인 경우, 승산 결과의 값은 비수가 됩니다.

DIV, /

정수, 실수를 제산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DIV, /	제산	FUN	(@)DIV EN ENO In1 Out In2 (@)/ EN ENO In1 Out In2	Out:=In1/ In2;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피제산값	입력	피제산값	데이터 형식에 따름	–	(*)
In2	제산값		제산값			
Out	출력값	출력	출력값	데이터 형식에 따름	–	–

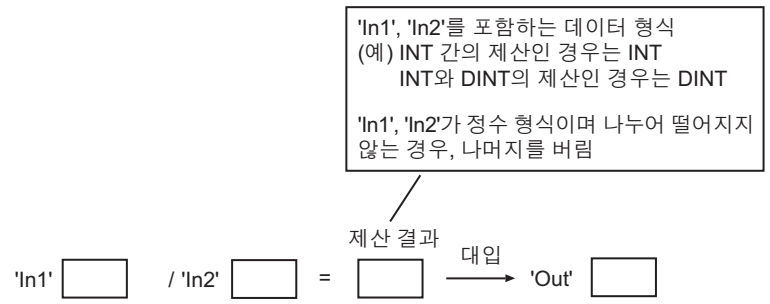
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1						○	○	○	○	○	○	○	○	○	○					
In2						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

피제산값 'In1'을 제산값 'In2'로 제산한 결과를 출력값 'Out'에 출력합니다.

'In1'과 'In2'는 상이한 데이터 형식이라도 상관없습니다. 그 경우, 그 모두를 포함하는 데이터 형식으로 연산 처리를 수행합니다. 예를 들면 'In1'이 INT 형식, 'In2'가 DINT 형식인 경우, 연산 처리를 DINT 형식으로 수행합니다. 따라서 제산 결과는 DINT 형식이 됩니다.
'In1', 'In2'가 모두 정수이며 나누어 떨어지지 않는 경우, 나머지를 버립니다.

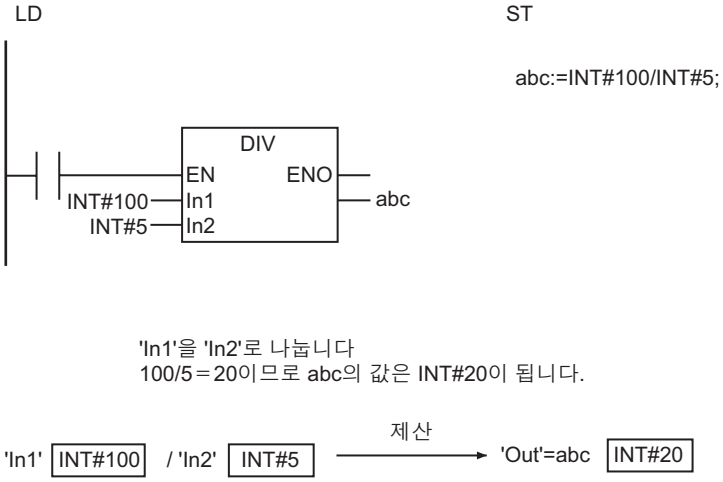


'In1'과 'InN'의 몫이 제산 결과의 데이터 형식의 유효 범위를 초과하는 것을 오버 플로우라고 합니다.
오버 플로우가 발생한 경우의 'In1', 'In2'의 데이터 형식, 제산 결과의 데이터 형식, 제산 결과의 값 간의 관계는 다음과 같습니다.

'In1', 'In2'의 데이터 형식	제산 결과의 데이터 형식	제산 결과의 값
실수 형식이 있는 경우	실수 형식	± ∞ (*)

* 'In1', 'In2'의 몫이 양수인 경우는 +∞, 음수인 경우는 -∞가 됩니다.

'In1'=INT#100, 'In2'=INT#5인 경우의 예가 아래에 나와 있습니다. INT 형식 변수 abc의 값은 INT#20이 됩니다.



DIV 명령과 / 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Out'은 제산 결과와 다른 데이터 형식이라도 상관없습니다. 단, 제산 결과의 데이터 형식을 포함하는 데이터 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 제산에 의해 오버 플로우 또는 언더 플로우가 발생해도 이상 상태가 되지 않습니다.
- 제산에 의해 오버 플로우나 언더 플로우가 발생하는 경우, 연산 결과가 의도한 대로 되지 않을 가능성이 있습니다. 입력 파라메터, 출력 파라메터의 데이터 형식의 크기에는 오버 플로우나 언더 플로우가 발생하지 않도록 충분히 여유 있게 설정하십시오.

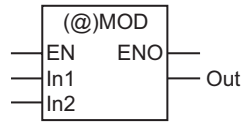
- 실수의 +∞/-∞ 및 0 대한 제산 결과는 다음과 같습니다.

		'In1'				
		+∞	**정수**	**0**	**음수**	**-∞**
'In2'	**+∞**	비수	0	0	0	비수
	정수	+∞	정수	0	음수	-∞
	0	+∞	+∞	비수	-∞	-∞
	음수	-∞	음수	0	정수	+∞
	-∞	비수	0	0	0	비수

- 'In1', 'In2' 중 어느 것의 값이 비수인 경우, 제산 결과의 값은 비수가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In1', 'In2'가 정수 형식이고 'In2'의 값이 0인 경우

MOD

정수를 제산했을 때의 나머지를 구합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MOD	나머지	FUN		Out:=In1 MOD In2;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피제산값	입력	피제산값	데이터 형식에 따름	–	(*)
In2	제산값		제산값			
Out	나머지	출력	나머지	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1						○	○	○	○	○	○	○	○							
In2						○	○	○	○	○	○	○	○							
Out						○	○	○	○	○	○	○	○							

기능

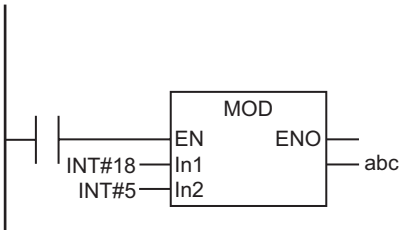
피제산값 'In1'을 제산값 'In2'로 제산했을 때의 나머지를 구합니다.
'In1', 'In2' 및 나머지 'Out'은 상이한 데이터 형식이라도 상관없습니다.

이 명령의 연산은 다음 식으로 수행합니다.
'Out'='In1'-('In1' / 'In2')*'In2' 단, 괄호 안의 제산은 소수점 이하를 버립니다.
따라서 'In1', 'In2'와 'Out'의 값의 예를 나타내면 다음과 같습니다.

'In1'의 값	'In2'의 값	'Out'의 값
5	3	2
5	-3	2
-5	3	-2
-5	-3	-2

'In1'=INT#18, 'In2'=INT#5인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 INT#3이 됩니다.

LD



ST

```
abc:=INT#18 MOD INT#5;
```

'In1'을 'In2'로 나누었을 때의 나머지를 구합니다.
18/5의 나머지는 3이므로 abc의 값은 INT#3이 됩니다.



사용상의 주의 사항

- 'Out'의 데이터 형식은 'In1', 'In2'의 유효 범위를 포함하도록 설정하십시오.

ABS

정수, 실수의 절대값을 구합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ABS	절대값	FUN	(@)ABS EN ENO In Out	Out:=ABS(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름	–	(*1)
Out	절대값	출력	절대값	데이터 형식에 따름 (*2)	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 음수는 포함되지 않습니다.

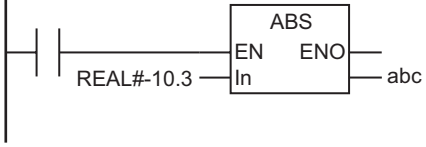
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

연산 대상 'In'의 절대값을 출력합니다.
'In'과 절대값 'Out'은 상이한 데이터 형식이라도 상관없습니다.

'In'=REAL# -10.3인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#10.3이 됩니다.

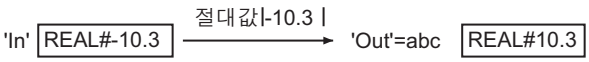
LD



ST

```
abc:=ABS(REAL#-10.3);
```

'In'의 절대값을 출력합니다.
REAL#-10.3의 절대값이므로 abc의 값은 REAL#10.3이 됩니다.



참고

실수를 연산하는 경우, 'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Out'의 데이터 형식은 'In'의 절대값이 포함되도록 설정하십시오.
- 'In'의 값이 +∞, -∞, 비수인 경우 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
+∞	+∞
-∞	+∞
비수	비수

RadToDeg / DegToRad

RadToDeg : 실수를 라디안(rad) 단위에서 도(°) 단위로 변환합니다.
DegToRad : 실수를 도(°) 단위에서 라디안(rad) 단위로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RadToDeg	라디안→각도 변환	FUN	(@)RadToDeg EN ENO In Out	Out:=RadToDeg(In);
DegToRad	각도→라디안 변환	FUN	(@)DegToRad EN ENO In Out	Out:=DegToRad(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환하는 데이터	데이터 형식에 따름	• RadToDeg : rad • DegToRad : °	(*)
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	• RadToDeg : ° • DegToRad : rad	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

● RadToDeg

변환 대상 'In'을 라디안(rad) 단위에서 도(°) 단위로 변환합니다.
변환식은 다음과 같습니다.

'Out'='In'*180/π

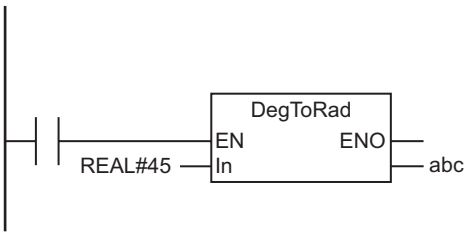
● DegToRad

변환 대상 'In'을 도(°) 단위에서 라디안(rad) 단위로 변환합니다.
변환식은 다음과 같습니다.

'Out'='In'*π/180

DegToRad 명령에서 'In'=REAL#45인 경우의 예가 아래에 나와 있습니다. REAL 형식 변수 abc의 값은 REAL#0.785398이 됩니다.

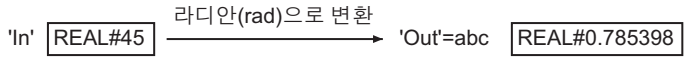
LD



ST

```
abc:=DegToRad(REAL#45);
```

'In'의 값을 도(°)에서 라디안(rad)으로 변환합니다.
45°는 라디안(rad)으로 0.785398이므로 abc의 값은 REAL#0.785398이 됩니다.



참고

'Out'이 +∞, -∞, 비수인지를 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 변환 결과의 절대값이 'Out'의 데이터 형식으로 표현할 수 있는 최대값보다 커지는 경우, 'Out'의 값은 +∞/-∞이 됩니다.
- 변환 결과의 절대값이 'Out'의 데이터 형식으로 표현할 수 있는 최소값보다 작아지는 경우, 'Out'의 값은 0이 됩니다.
- 'Out'의 데이터 형식은 'In'의 데이터 형식 이상의 크기로 하십시오.
- 'In'의 값이 +∞, -∞, 비수인 경우 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
+∞	+∞
-∞	-∞
비수	비수

- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

SIN / COS / TAN

실수의 삼각함수 연산을 수행합니다.
SIN : sin(사인값)
COS : cos(코사인값)
TAN : tan(탄젠트값)

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SIN	SIN 연산	FUN	(@)SIN EN ENO In Out	Out:=SIN(In);
COS	COS 연산	FUN	(@)COS EN ENO In Out	Out:=COS(In);
TAN	TAN 연산	FUN	(@)TAN EN ENO In Out	Out:=TAN(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름	rad	(*1)
Out	연산 결과	출력	연산 결과	• SIN : (*2) • COS : (*2) • TAN : 데이터 형식에 따름	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 REAL의 경우, -1.000000e+0~1.000000e+0입니다. LREAL의 경우, -1.00000000000000e+0~1.00000000000000e+0입니다.

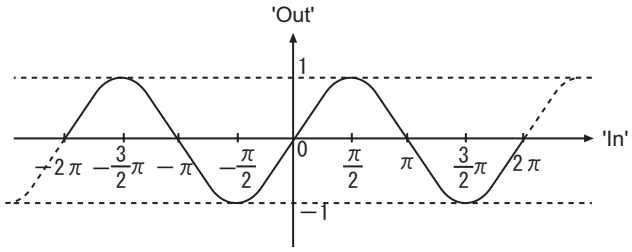
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

실수의 삼각함수 연산을 수행합니다.
연산 대상 'In'은 라디안(rad) 단위의 각도입니다.

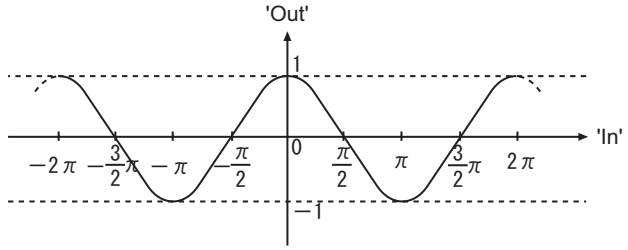
● SIN

'In'의 sin(사인값)을 구합니다.



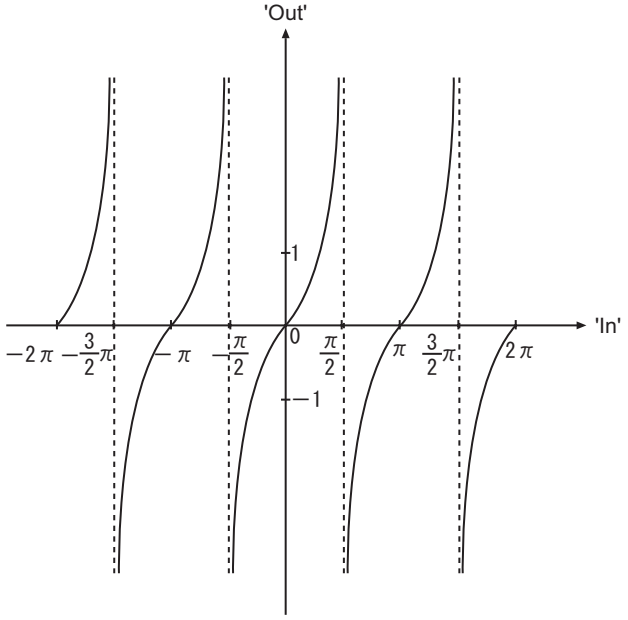
● COS

'In'의 cos(코사인값)을 구합니다.



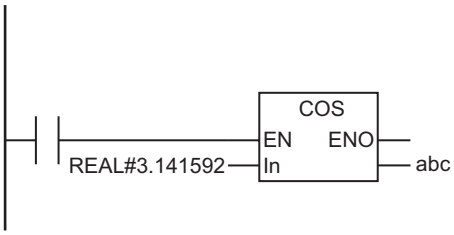
● TAN

'In'의 tan(탄젠트값)을 구합니다.



COS 명령에서 'In'=REAL#3.141592인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#-1.0이 됩니다.

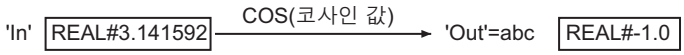
LD



ST

```
abc:=COS(REAL#3.141592);
```

'In'의 COS(코사인 값)을 구합니다.
3.141592의 COS(코사인 값)은 -1.0이므로 abc의 값은 REAL#-1.0이 됩니다.



참고

- 도(°)와 라디안(rad)을 변환하려면 'RadToDeg / DegToRad 명령(2-187페이지)'을 사용하십시오.
- TAN 명령에서는 'In'이 $n\pi/2$(n은 정수)인 경우, 'Out'의 값이 +∞/-∞가 됩니다. 'Out'의 값이 +∞/-∞인지 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 값이 +∞, -∞, 비수인 경우, 'Out'의 값은 비수가 됩니다.
- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

ASIN / ACOS / ATAN

실수의 역 삼각함수 연산을 수행합니다.

ASIN : $\sin^{-1}$(역 사인값)

ACOS : $\cos^{-1}$(역 코사인값)

ATAN : $\tan^{-1}$(역 탄젠트값)

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ASIN	SIN^{-1} 연산	FUN	(@)ASIN EN ENO In Out	Out:=ASIN(In);
ACOS	COS^{-1} 연산	FUN	(@)ACOS EN ENO In Out	Out:=ACOS(In);
ATAN	TAN^{-1} 연산	FUN	(@)ATAN EN ENO In Out	Out:=ATAN(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름	–	(*)
Out	연산 결과	출력	연산 결과	• ASIN : $-\pi/2 \sim \pi/2$ • ACOS : $0 \sim \pi$ • ATAN : $-\pi/2 \sim \pi/2$	rad	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

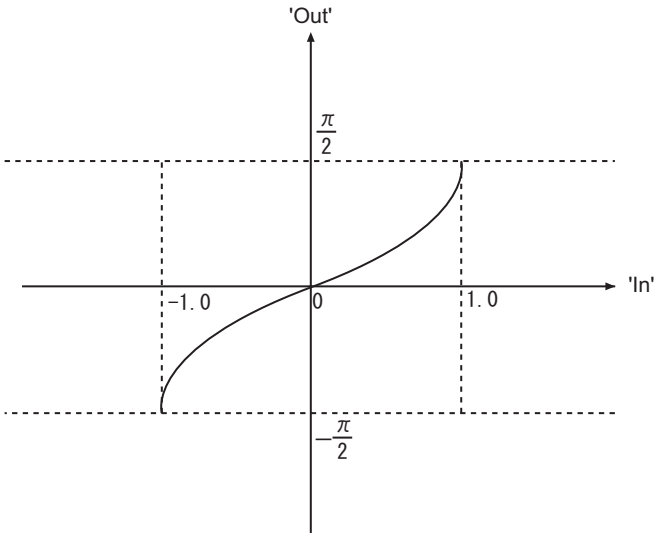
기능

실수의 역 삼각함수 연산을 수행합니다.

연산 결과 'Out'은 라디안 단위의 각도입니다.

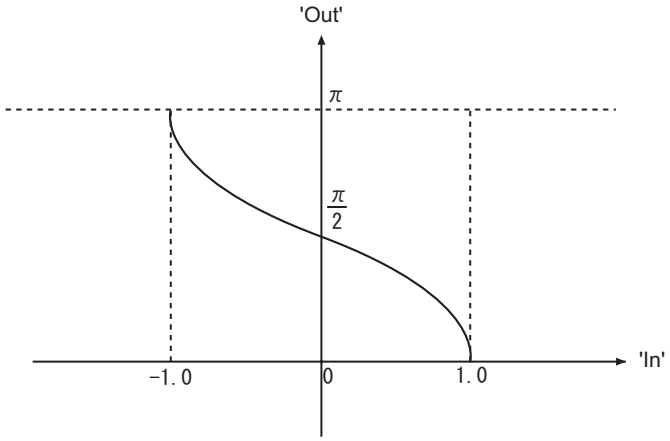
● ASIN

'In'의 $\sin^{-1}$(역 사인값)을 출력합니다. 'Out'은 $-\pi/2 \sim \pi/2$의 범위입니다.



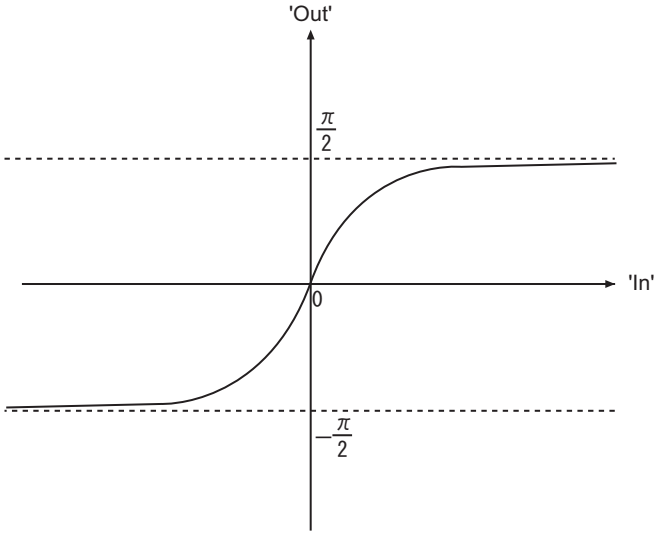
● ACOS

'In'의 $\cos^{-1}$(역 코사인값)을 출력합니다. 'Out'은 $0 \sim \pi$의 범위입니다.



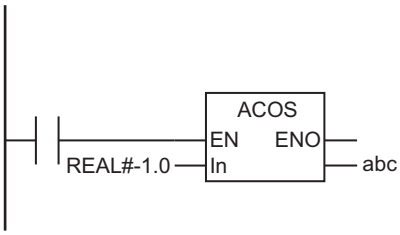
● ATAN

'In'의 $\tan^{-1}$(역 탄젠트값)을 출력합니다. 'Out'은 $-\pi/2 \sim \pi/2$의 범위입니다.
'In'의 값이 +∞인 경우, 'Out'의 값은 $\pi/2$가 됩니다. 'In'의 값이 -∞인 경우, 'Out'의 값은 $-\pi/2$가 됩니다.



ACOS 명령에서 'In'=REAL#-1.0의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#3.141592이 됩니다.

LD



ST

```
abc:=ACOS(REAL#-1.0);
```

'In'의 $\cos^{-1}$(역코사인 값)을 구합니다.
-1.0의 $\cos^{-1}$(역코사인 값)은 3.141592이므로 abc의 값은 REAL#3.141592가 됩니다.

'In' REAL#-1.0 —$\cos^{-1}$(역코사인 값)→ 'Out'=abc REAL#3.141592

참고

도(°)와 라디안(rad)을 변환하려면 📖 'RadToDeg / DegToRad 명령(2-187페이지)'을 사용하십시오.

사용상의 주의 사항

- ASIN 명령, ACOS 명령에서 'In'의 값이 -1.0~1.0 이외인 경우, 'Out'의 값은 비수가 됩니다. 'In'의 값이 +∞, -∞, 비수인 경우도 마찬가지입니다.
- ATAN 명령에서 'In'의 값이 비수인 경우, 'Out'의 값은 비수가 됩니다.

• 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

SQRT

실수의 제곱근(√)을 구합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SQRT	제곱근 연산	FUN	(@)SQRT EN ENO In Out	Out:=SQRT(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름 (*1)	–	(*2)
Out	제곱근	출력	제곱근	(*3)	–	–

*1 음수는 포함되지 않습니다.

*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

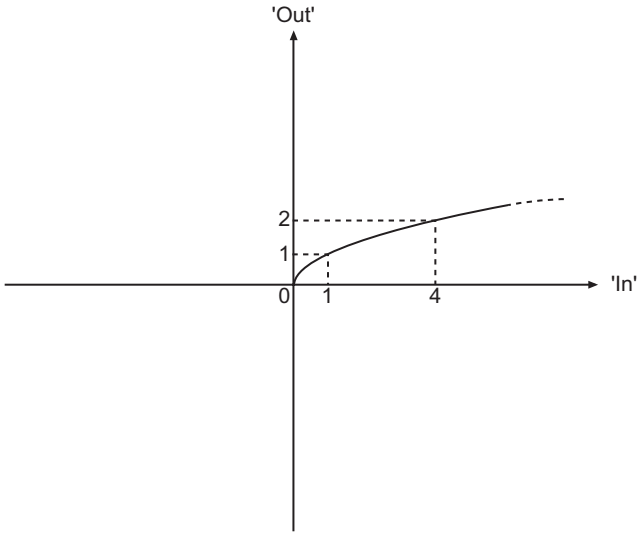
*3 REAL의 경우, 0.000000e+00~1.844674e+19 또는 +∞입니다. LREAL의 경우, 0.00000000000000e+000~1.34078079299425e+154 또는 +∞입니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

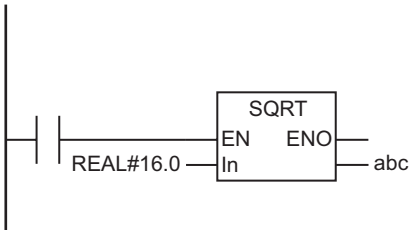
실수의 연산 대상 'In'의 제곱근(√)을 구합니다.

'In'과 제곱근 'Out'은 상이한 데이터 형식이라도 상관없습니다.



'In'=REAL# 16.0인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#4.0이 됩니다.

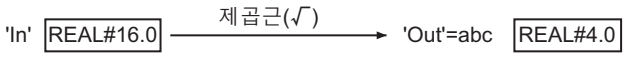
LD



ST

```
abc:=SQRT(REAL#16.0);
```

'In'의 제곱근(√)을 구합니다.
16.0의 제곱근(√)은 4.0이므로 abc의 값은 REAL#4.0이 됩니다.



참고

'Out'의 값이 +∞인지 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 값이 양의 수치가 아닌 경우, 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
음의 수치	비수
0	0
+∞	+∞
-∞	비수
비수	비수

- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

LN / LOG

실수의 로그를 구합니다.
LN : 자연 로그
LOG : 상용 로그

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LN	자연 로그 연산	FUN	(@)LN EN ENO In Out	Out:=LN(In);
LOG	상용 로그 연산	FUN	(@)LOG EN ENO In Out	Out:=LOG(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름 (*1)	–	(*2)
Out	로그	출력	로그	(*3)	–	–

*1 음수는 포함되지 않습니다.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

- LN
  'In', 'Out'이 REAL인 경우 -8.73365448e+1~8.87228390e+1 또는 -∞/+∞
  'In'이 REAL, 'Out'이 LREAL인 경우 -8.7336544750000000e+1~8.8722839050000000e+1 또는 -∞/+∞
  'In'이 LREAL, 'Out'이 REAL인 경우 -7.08384950e+2~7.09782712e+2 또는 -∞/+∞
  'In', 'Out'이 LREAL인 경우 -7.0838495021978327e+1~7.0978271289338399e+2 또는 -∞/+∞
- LOG
  'In', 'Out'이 REAL인 경우 -3.79297795e+1~3.85318394e+1 또는 -∞/+∞
  'In'이 REAL, 'Out'이 LREAL인 경우 -3.7929779453965430e+1~3.8531839419564961e+1 또는 -∞/+∞
  'In'이 LREAL, 'Out'이 REAL인 경우 -3.07652656e+2~3.08254716e+2 또는 -∞/+∞
  'In', 'Out'이 LREAL인 경우 -3.0765265556858878e+2~3.0825471555991674e+2 또는 -∞/+∞

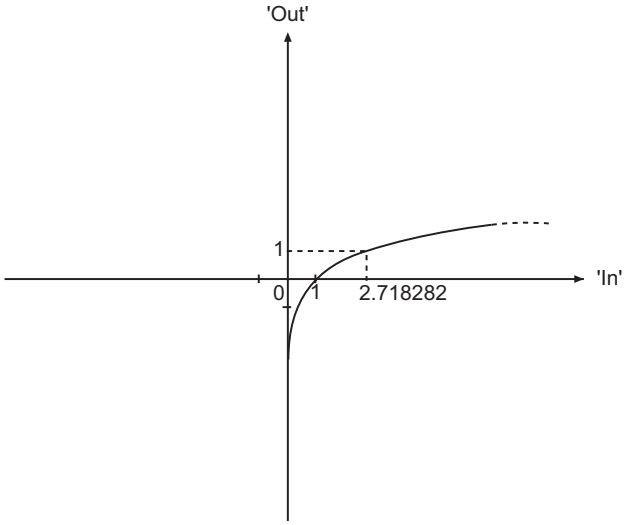
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

실수의 로그를 구합니다.

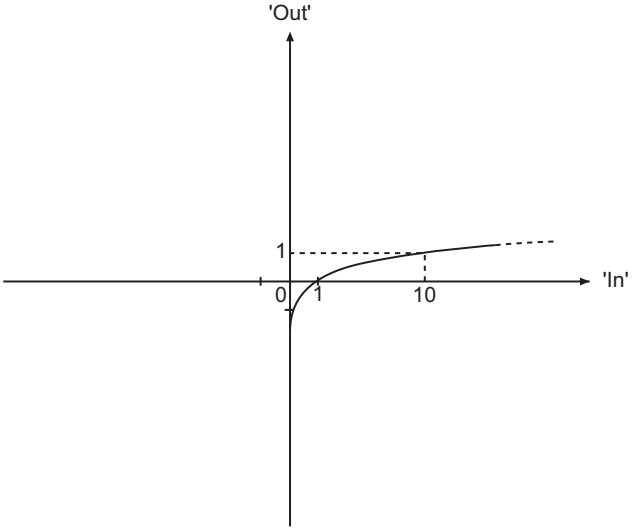
● LN

자연 로그(밑수가 e 2.718282인 로그)를 구합니다.



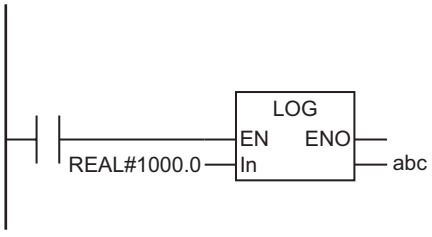
● LOG

상용 로그(밑수가 10인 로그)를 구합니다.



LOG 명령에서 'In'=REAL#1000.0인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#3.0이 됩니다.

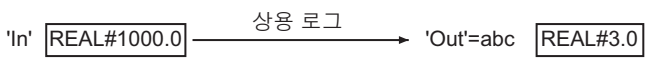
LD



ST

```
abc:=LOG(REAL#1000.0);
```

'In'의 상용 로그를 구합니다.
1000.0의 상용 로그는 3.0이므로 abc의 값은 REAL#3.0이 됩니다.



참고

'Out'의 값이 +∞, -∞, 비수인지 판정하려면 📖 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 값이 양의 수치가 아닌 경우, 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
음의 수치	비수
0	-∞
+∞	+∞
-∞	비수
비수	비수

- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

EXP

자연 로그의 밑수 e의 지수 함수를 구합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EXP	자연 지수 연산	FUN	(@)EXP EN ENO In Out	Out:=EXP(In);

변수

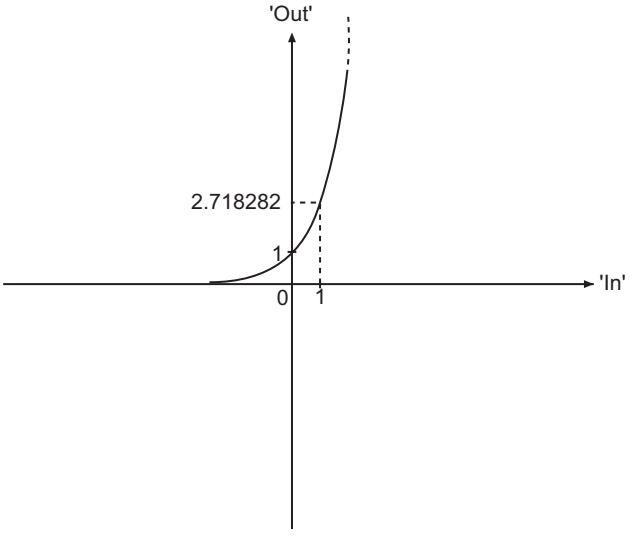
	명칭	입/출력	내용	유효 범위	단위	초기값
In	거듭제곱	입력	거듭제곱	데이터 형식에 따름	–	(*1)
Out	연산 결과	출력	연산 결과	데이터 형식에 따름 (*2)	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

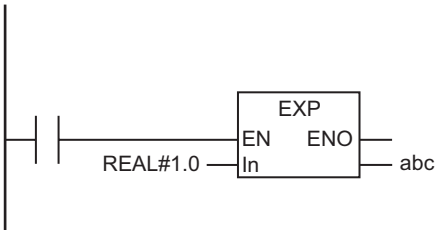
기능

자연 로그의 밑수 e의 'In'승을 구합니다.



'In'=REAL# 1.0인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#2.718282가 됩니다.

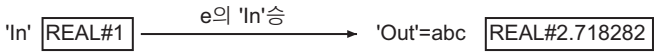
LD



ST

```
abc:=EXP(REAL#1.0);
```

자연 로그의 밑수 e의 'In'승을 구합니다.
e^1은 2.718282이므로 abc의 값은 REAL#2.718282가 됩니다.



참고

- 밑수가 e 이외인 지수를 구하는 경우에는 'EXPT, ** 명령(2-203페이지)'을 사용하십시오.
- 'Out'의 값이 +∞, -∞, 비수인지 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 값이 0.0, +∞, -∞, 비수인 경우 'Out'의 값은 다음과 같습니다.

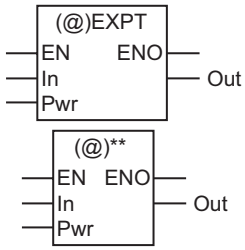
'In'의 값	'Out'의 값
0.0	1.0
+∞	+∞
-∞	0.0
비수	비수

- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

EXPT, **

2개의 실수의 거듭제곱을 구합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
EXPT, **	거듭제곱 연산	FUN		Out:=EXPT(In, Pwr); Out:=In ** Pwr;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	피거듭제곱 수	입력	피거듭제곱 수 5^2인 경우의 5	데이터 형식에 따름	–	(*)
Pwr	거듭제곱		피거듭제곱 수 5^2인 경우의 2			
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

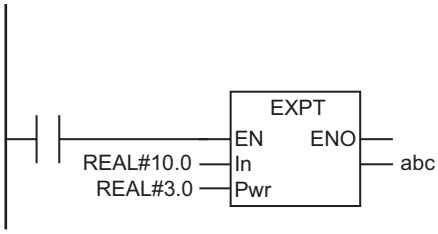
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Pwr														○	○					
Out														○	○					

기능

피거듭제곱 수 'In'과 거듭제곱 수 'Pwr'에 대해, In^{Pwr}의 값을 구합니다.

'In'=REAL#10.0, 'Pwr'=REAL#3.0인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#1000.0이 됩니다.

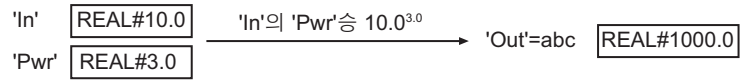
LD



ST

```
abc:=EXPT(REAL#10.0, REAL#3.0);
```

'In'의 'Pwr'승을 구합니다.
$10.0^{3.0}$은 1000.0이므로 abc의 값은 REAL#1000.0이 됩니다.



EXPT 명령과 ** 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

- 밑수가 e 이외인 지수를 구하는 경우에는 'EXP 명령(2-201페이지)'을 사용하십시오.
- 'Out'의 값이 +∞, -∞, 비수인지 판정하려면 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 연산 결과의 절대값이 실수로 표현할 수 있는 최소값보다 작아지는 경우, 'Out'의 값은 0이 됩니다.
  (예) $(1.175494e\text{-}38)^2 \rightarrow 0$
- 'In'과 'Pwr'의 값의 조합에 따라, 'Out'의 값은 다음과 같습니다.

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="10">'In'</th></tr>
<tr><th>+∞</th><th>1~+∞</th><th>1</th><th>0~1</th><th>0</th><th>-1~0</th><th>-1</th><th>-1~-∞</th><th>-∞</th><th>비수</th></tr>
<tr><th rowspan="11">'Pwr'</th><th>+∞</th><td>+∞</td><td>+∞</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>+∞</td><td>+∞</td><td>비수</td></tr>
<tr><th>양의 짝수</th><td rowspan="3">+∞</td><td colspan="3" rowspan="3">수치(*1) (*2)</td><td rowspan="3">0</td><td colspan="3">수치(*1) (*2)</td><td>+∞</td><td rowspan="3">비수</td></tr>
<tr><th>양의 홀수</th><td colspan="3">수치(*2) (*3)</td><td>-∞</td></tr>
<tr><th>양의 소수</th><td colspan="3">비수</td><td>+∞</td></tr>
<tr><th>0</th><td>1</td><td colspan="3">1</td><td>1</td><td colspan="3">1</td><td>1</td><td>1</td></tr>
<tr><th>음의 짝수</th><td rowspan="3">0</td><td colspan="3" rowspan="3">수치(*1) (*2)</td><td rowspan="3">+∞</td><td colspan="3">수치(*1) (*2)</td><td rowspan="3">0</td><td rowspan="3">비수</td></tr>
<tr><th>음의 홀수</th><td colspan="3">수치(*2) (*3)</td></tr>
<tr><th>음의 소수</th><td colspan="3">비수</td></tr>
<tr><th>-∞</th><td>0</td><td>0</td><td>1</td><td>+∞</td><td>+∞</td><td>+∞</td><td>1</td><td>0</td><td>0</td><td>비수</td></tr>
<tr><th>비수</th><td>비수</td><td colspan="3">비수</td><td>비수</td><td colspan="3">비수</td><td>비수</td><td>비수</td></tr>
</table>

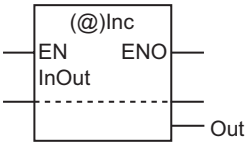
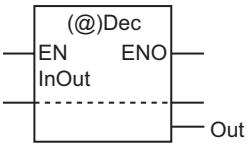
*1 연산 결과가 'Out'의 데이터 형식의 유효 범위를 초과한 경우, 'Out'의 값은 +∞가 됩니다.
*2 연산 결과가 'Out'의 데이터 형식으로는 표현할 수 없을 정도로 0에 가까운 경우, 또는 비정규화 수인 경우 'Out'의 값은 0이 됩니다.
*3 연산 결과가 'Out'의 데이터 형식의 유효 범위를 초과한 경우, 'Out'의 값은 -∞가 됩니다.

- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

Inc / Dec

Inc : 정수값을 증가시킵니다.
Dec : 정수값을 감소시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Inc	증가	FUN		Inc(InOut);
Dec	감소	FUN		Dec(InOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	대상 데이터	입출력	대상 데이터	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut						○	○	○	○	○	○	○	○							
Out	○																			

기능

- **Inc**

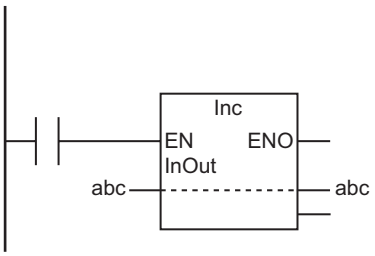
  대상 데이터 'InOut'을 증가시킵니다. 그 결과 'InOut'의 최대값을 초과한 경우에는 최소값으로 되돌립니다.

- **Dec**

  대상 데이터 'InOut'을 감소시킵니다. 그 결과 'InOut'의 최소값을 초과한 경우에는 최대값으로 되돌립니다.

Inc 명령에서 'InOut'에 변수 abc를 전달하는 경우의 기술 예가 아래에 나와 있습니다. abc의 값이 INT#4인 경우, 명령 실행 후 abc의 값은 INT#5가 됩니다.

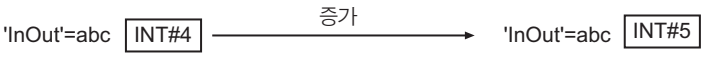
LD



ST

```
Inc(abc);
```

'InOut'을 증가시킵니다.
abc의 값이 INT#4인 경우, 명령 실행 후 abc의 값은 INT#5가 됩니다.



사용상의 주의 사항

이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.

Rand

의사 난수를 발생시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Rand	난수 발생	FB	Rand_instance Rand Execute ENO Seed Rnd	Rand_instance(Execute, Seed, Rnd);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Seed	난수 패턴	입력	난수 계열 지정 0: 지정 없음	데이터 형식에 따름	–	(*1)
Rnd	난수	출력	난수	(*2)	–	–

*1 입력 파라메터가 생략된 경우, 값은 0이 됩니다. 초기값 속성에서 지정된 값이 되지는 않습니다.

*2 0.00000000000000e+0~1.00000000000000e+0

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Seed							○													
Rnd															○					

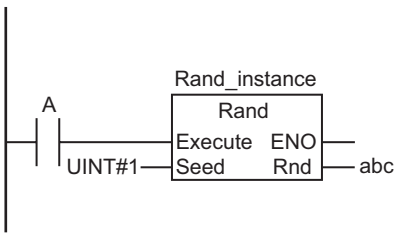
기능

난수 'Rnd'를 발생시킵니다. 'Rnd'의 값은 명령을 실행할 때마다 달라집니다.
난수 패턴 'Seed'는 난수 계열의 지정입니다. 'Seed'의 값이 동일하면, 전원을 ON으로 할 때마다 동일한 난수 계열이 발생합니다. 따라서, 재현성이 있는 난수를 발생시킬 수 있습니다.

'Seed'의 값이 0인 경우에는 재현성이 없는 난수가 발생합니다. 전원을 ON으로 할 때마다 동일한 난수 계열을 발생시키고 싶지 않은 경우에는, 'Seed'의 값을 0으로 하십시오.

'Seed'=UINT#1인 경우의 기술 예가 아래에 나와 있습니다. 'Seed'의 값이 0이 아니므로 재현성이 있는 난수가 발생합니다.

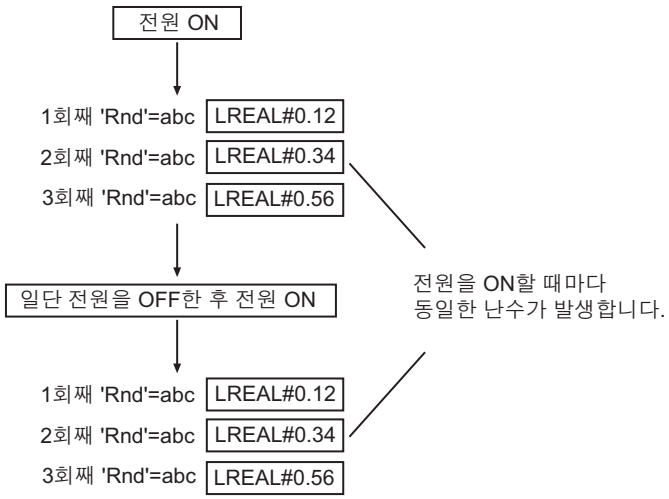
LD



ST

```
Rand_instance(A, UINT#1, abc);
```

재현성이 있는 난수가 발생합니다.



(*) 상기의 난수 값은 예이며, 실제 값과 다릅니다.

참고

'Rnd'의 값은 0~1의 범위의 실수입니다. 따라서, 다음과 같은 처리를 통해 특정한 범위의 난수를 발생시킬 수 있습니다.

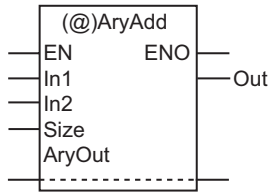
(예) 100~200의 범위의 난수를 발생시키는 경우

Rand_instance(A, UINT#1, abc);

난수: =LREAL_TO_INT((200.0-100.0)*abc)+100;

AryAdd

2개의 배열의 요소별 가산을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryAdd	배열 일괄 가산	FUN		AryAdd(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열 In2[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	연산 대상의 요소 수		연산 대상의 요소 수			1
AryOut[] 배열	연산 결과 배열	입출력	연산 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

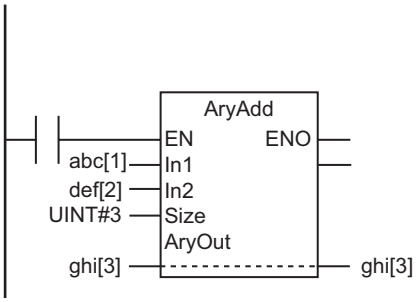
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2[] 배열	In1[]과 동일한 데이터 형식의 배열																			
Size							○													
AryOut[] 배열	In1[]과 동일한 데이터 형식의 배열																			
Out	○																			

기능

연산 대상 배열 In1[], In2[]의 In1[0], In2[0] 이후의 'Size'개의 요소 각각의 가산값을 연산 결과 배열 AryOut[]에 대입합니다.

'Size'=UNIT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
AryAdd(abc[1], def[2], UINT#3, ghi[3]);
```

'Size'=UINT#3

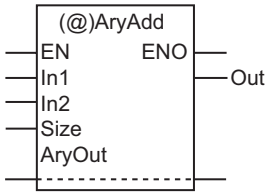
In1[0]=abc[1]	1234	+	In2[0]=def[2]	2345	→	AryOut[0]=ghi[3]	3579
In1[1]=abc[2]	2345	+	In2[1]=def[3]	3456	→	AryOut[1]=ghi[4]	5801
In1[2]=abc[3]	3456	+	In2[2]=def[4]	4567	→	AryOut[2]=ghi[5]	8023

사용상의 주의 사항

- In1[], In2[], AryOut[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- 연산 결과가 AryOut[]의 유효 범위 밖인 경우, 결과는 잘못된 값이 됩니다. 이 경우, 이상 상태가 되지는 않습니다. 또한, 그 요소 주변의 메모리 영역을 파괴하지도 않습니다.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In1[], In2[], AryOut[] 중 어느 것의 배열 영역을 초과한 경우

AryAddV

배열의 각 요소에 동일 수치를 가산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryAddV	배열 요소 가산	FUN		AryAddV(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열	가산 배열	입력	가산 배열	데이터 형식에 따름	–	(*)
In2	가산값		가산값			
Size	요소 수		가산하는 In1[]의 요소 수			1
AryOut[] 배열	가산 결과 배열	입출력	가산 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

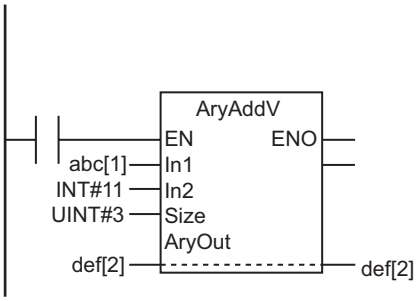
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2	In1[]의 요소와 동일한 데이터 형식																			
Size							○													
AryOut[] 배열	In1[]과 같은 데이터 형식																			
Out	○																			

기능

가산 배열 In1[]의 In1[0]부터 'Size'개까지의 각 요소에, 가산값 'In2'의 값을 가산해서 가산 결과 배열 AryOut[]에 출력합니다.

'In2'=INT#11, 'Size'=UNIT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
AryAddV(abc[1], INT#11, UINT#3, def[2]);
```

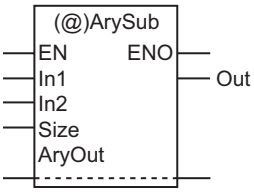
'Size'=UINT#3							
	In1[0]=abc[1]	12	+	'In2'=INT#11	→	AryOut[0]=def[2]	23
	In1[1]=abc[2]	23	+	'In2'=INT#11	→	AryOut[1]=def[3]	34
	In1[2]=abc[3]	34	+	'In2'=INT#11	→	AryOut[2]=def[4]	45

사용상의 주의 사항

- In1[], 'In2', AryOut[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- 가산 결과가 AryOut[]의 유효 범위 밖인 경우, AryOut[]의 요소는 잘못된 값이 됩니다. 이 경우, 이상 상태가 되지는 않습니다. 또한, 그 요소 주변의 메모리 영역을 파괴하지도 않습니다.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In1[] 또는 AryOut[]의 배열 영역을 초과한 경우

ArySub

2개의 배열의 요소별 감산을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ArySub	배열 일괄 감산	FUN		ArySub(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열	피감산 배열	입력	피감산 배열	데이터 형식에 따름	–	(*)
In2[] 배열	감산 배열		감산 배열			
Size	요소 수		감산하는 요소 수			1
AryOut[] 배열	감산 결과 배열	입출력	감산 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

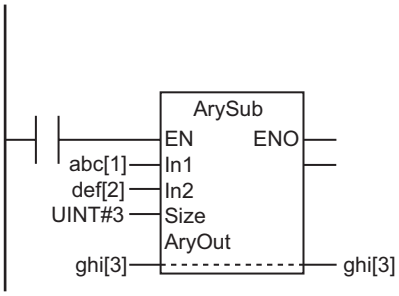
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2[] 배열	In1[]과 같은 데이터 형식																			
Size							○													
AryOut[] 배열	In1[]과 같은 데이터 형식																			
Out	○																			

기능

피감산 배열 In1[]과 감산 배열 In2[]에 대해, 시작부터 'Size'개까지의 각 요소의 감산을 수행합니다. 감산 결과는 감산 결과 배열 AryOut[]에 출력합니다.

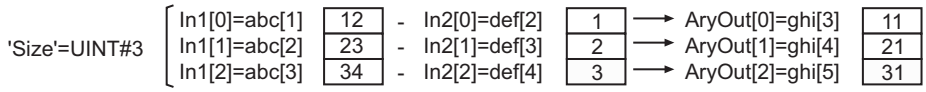
'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
ArySub(abc[1], def[2], UINT#3, ghi[3]);
```

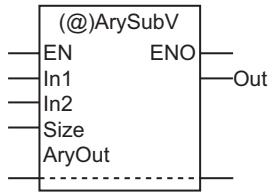


사용상의 주의 사항

- In1[], In2[], AryOut[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- 감산 결과가 AryOut[]의 유효 범위 밖인 경우, AryOut[]의 요소는 잘못된 값이 됩니다. 이 경우, 이상 상태가 되지는 않습니다. 또한, 그 요소 주변의 메모리 영역을 파괴하지도 않습니다.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In1[], In2[], AryOut[] 중 어느 것의 배열 영역을 초과한 경우

ArySubV

배열의 각 요소에서 동일 수치를 감산합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ArySubV	배열 요소 감산	FUN		ArySubV(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[] 배열	피감산 배열	입력	피감산 배열	데이터 형식에 따름	–	(*)
In2	감산값		감산값			
Size	요소 수		감산하는 In1[]의 요소 수			1
AryOut[] 배열	감산 결과 배열	입출력	감산 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

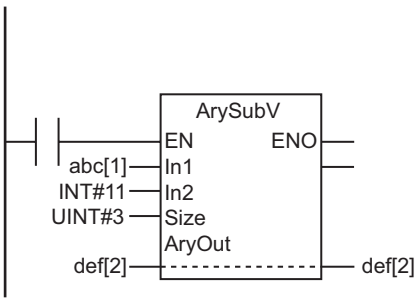
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열						○	○	○	○	○	○	○	○	○	○					
In2	In1[]의 요소와 동일한 데이터 형식																			
Size							○													
AryOut[] 배열	In1[]과 같은 데이터 형식																			
Out	○																			

기능

피감산 배열 In1[]의 In1[0]부터 'Size'개까지의 각 요소에서 감산값 'In2'의 값을 감산해서 감산 결과 배열 AryOut[]에 출력합니다.

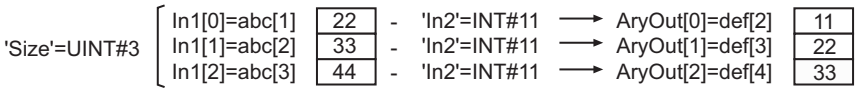
'In2'=INT#11, 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
ArySubV(abc[1], INT#11, UINT#3, def[2]);
```



사용상의 주의 사항

- In1[], 'In2', AryOut[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- 감산 결과가 AryOut[]의 유효 범위 밖인 경우, AryOut[]의 요소는 잘못된 값이 됩니다. 이 경우, 이상 상태가 되지는 않습니다. 또한, 그 요소 주변의 메모리 영역을 파괴하지도 않습니다.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In1[] 또는 AryOut[]의 배열 영역을 초과한 경우

AryMean

배열 요소의 평균값을 산출합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AryMean	배열 요소의 평균값 산출	FUN	(@)AryMean EN ENO In Out Size	Out := AryMean(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	연산 대상의 요소 수		In[]의 요소 수			1
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

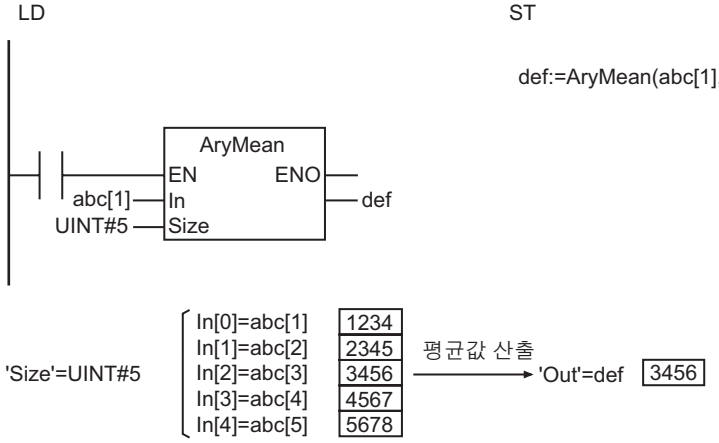
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열						○	○	○	○	○	○	○	○	○	○	○				
Size							○													
Out						○	○	○	○	○	○	○	○	○	○	○				

기능

연산 대상 배열 In[]의, In[0] 이후의 'Size'개의 요소의 평균값을 산출합니다.

'Size'=UINT#5인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- In[]의 값이 +∞/-∞ 및 비수인 경우의 사칙연산 결과는 🕮 'ADD,+ 명령(2-162페이지)', 🕮 'SUB, - 명령(2-168페이지)', 🕮 'MUL, * 명령(2-174페이지)', 🕮 'DIV, / 명령(2-180페이지)'의 기능 설명을 참조하십시오.
- In[], 'Out'이 정수 형식인 경우, 평균값의 소수점 이하를 버립니다.
- In[]과 'Out'의 데이터 형식이 다른 경우에는 'Out'의 유효 범위가 In[]의 유효 범위를 포함하도록 하십시오.
- 연산 결과가 'Out'의 유효 범위를 초과하는 경우, 'Out'의 값은 잘못된 값이 됩니다. 그리고 그 경우, 이상 상태가 되지는 않습니다.
- 연산 과정에서의 중간값이 In[]의 유효 범위를 초과한 경우, 'Out'의 값은 잘못된 값이 됩니다. 그리고 그 경우, 이상 상태가 되지는 않습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

ArySD

배열 요소의 표준 편차를 산출합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ArySD	배열 요소의 표준 편차	FUN	(@)ArySD EN ENO In Out Size	Out:=ArySD(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	요소 수		변환하는 In[]의 요소 수			2
Out	표준 편차	출력	표준 편차	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열														○	○					
Size							○													
Out														○	○					

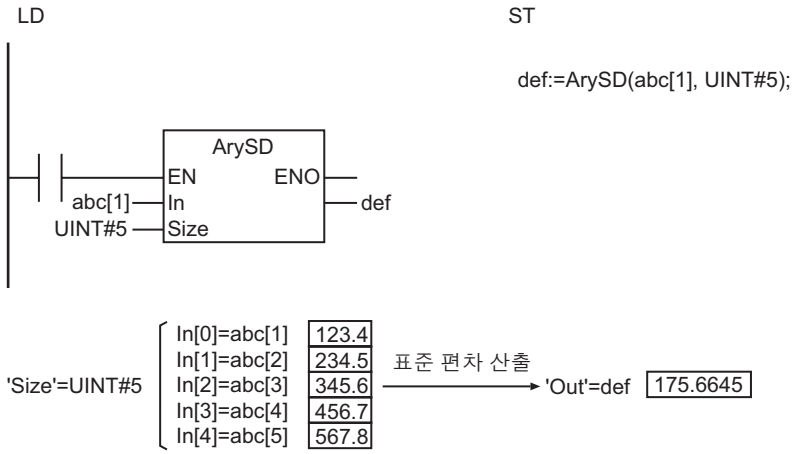
기능

연산 대상 배열 In[]의, In[0]부터 'Size'개의 요소의 표준 편차를 산출합니다.

$$\text{표준 편차} = \sqrt{\frac{\sum_{i} (In[i] - InM)^2}{'Size' - 1}}$$

i : In[]의 첨자 0~'Size'-1
InM : In[0]~In['Size'-1]의 평균값

'Size'=UINT#5인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 'Size'의 값이 0 또는 1인 경우, 'Out'의 값은 0이 됩니다.
- 연산 과정에서의 중간값이 In[]의 유효 범위를 초과한 경우, 'Out'의 값은 잘못된 값이 됩니다. 그리고 그 경우, 이상 상태가 되지는 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

ModReal

실수를 제산했을 때의 나머지를 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ModReal	실수 나머지	FUN	(@)ModReal EN ENO In1 Out In2	Out:=ModReal(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피제산값	입력	피제산값	데이터 형식에 따름	–	(*)
In2	제산값		제산값			
Out	나머지	출력	나머지	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														○	○					
In2														○	○					
Out														○	○					

기능

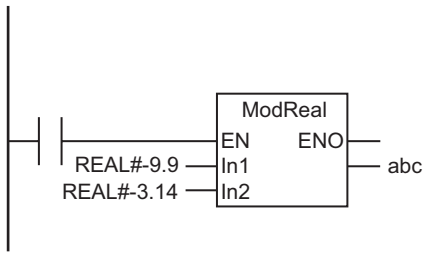
피제산값 'In1'을 제산값 'In2'로 나누었을 때의 나머지를 산출합니다.

이 명령의 연산은 다음 식으로 수행합니다.
'Out'='In1'-('In1'/'In2')*'In2'. 단, 괄호 안의 제산은 소수점 이하를 버립니다.
따라서 'In1', 'In2'와 'Out'의 값의 예는 다음과 같습니다.

'In1'의 값	'In2'의 값	'Out'의 값
9.9	3.14	0.48
9.9	-3.14	0.48
-9.9	3.14	-0.48
-9.9	-3.14	-0.48

'In1'=REAL#-9.9, 'In2'=REAL#-3.14인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#-0.48이 됩니다.

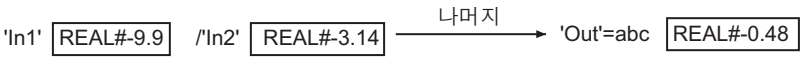
LD



ST

```
abc:=ModReal(REAL#-9.9, REAL#-3.14);
```

'In1'을 'In2'로 나누었을 때의 나머지를 구합니다.
-9.9/(-3.14)의 나머지는 -0.48이므로 abc의 값은 REAL#-0.48이 됩니다.



참고

'Out'의 값이 +∞, -∞, 비수인지 감지하고자 하는 경우에는 📖 'CheckReal 명령(2-225페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'과 'In2'의 값의 조합에 따라, 'Out'의 값은 다음과 같습니다.

		'In1'				
		0	**수치**	**+∞**	**-∞**	**비수**
'In2'	**0**	비수	비수	비수	비수	비수
	수치	0	'In1'/'In2'의 나머지	비수	비수	비수
	+∞	0	'In1'의 값	비수	비수	비수
	-∞	0	'In1'의 값	비수	비수	비수
	비수	비수	비수	비수	비수	비수

- 'In1' 또는 'In2'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In1' 또는 'In2'에 전달하는 파라메터의 데이터 형식	'In1' 또는 'In2'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

Fraction

실수의 소수부를 추출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Fraction	실수 소수부 추출	FUN	(@)Fraction EN ENO In Out	Out:=Fraction(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	실수	입력	실수	데이터 형식에 따름	–	(*)
Out	소수부	출력	소수부	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

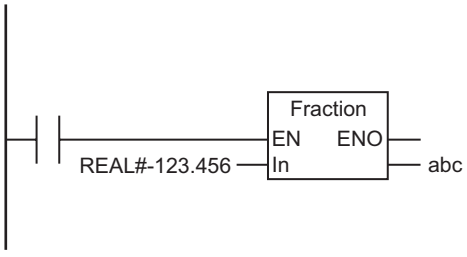
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

실수 'In'의 소수부를 추출합니다.

'In'=REAL#-123.456인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 REAL#-0.456이 됩니다.

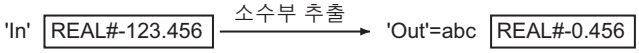
LD



ST

```
abc:=Fraction(REAL#-123.456);
```

'In'의 소수부를 추출합니다.
-123.456의 소수부는 -0.456이므로 abc의 값은 REAL#-0.456이 됩니다.



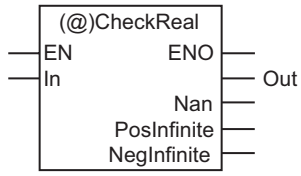
참고

- 'Out'의 값이 +∞, -∞, 비수인지 감지하고자 하는 경우에는 'CheckReal 명령(2-225페이지)'을 사용하십시오.
- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

CheckReal

실수가 무한대 또는 비수인지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CheckReal	실수 확인	FUN		CheckReal(In, Nan, PosInfinite, NegInfinite);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	실수	입력	실수	데이터 형식에 따름	–	(*)
Out	반환값	출력	항상 TRUE	TRUE만	–	–
Nan	비수 판정 결과		TRUE : 비수 FALSE: 비수가 아님	데이터 형식에 따름		
PosInfinite	양의 무한대 판정 결과		TRUE : 양의 무한대 FALSE: 양의 무한대가 아님			
NegInfinite	음의 무한대 판정 결과		TRUE : 음의 무한대 FALSE: 음의 무한대가 아님			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

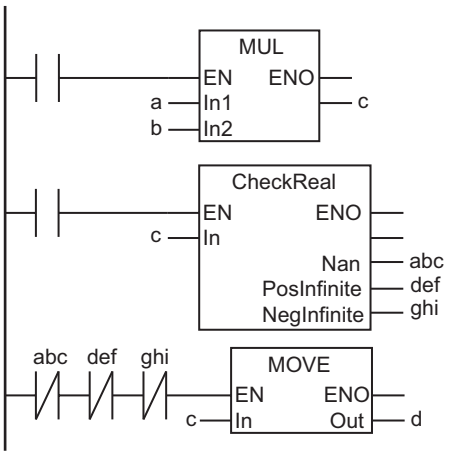
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out	○																			
Nan	○																			
PosInfinite	○																			
NegInfinite	○																			

기능

실수 'In'이 비수, 양의 무한대, 음의 무한대인지 판정해서 각각 'Nan', 'PosInfinite', 'NegInfinite'로 결과를 출력합니다.

기술 예가 아래에 나와 있습니다. REAL 형식의 변수 a, b를 승산한 결과가 유효한 실수인지 조사합니다. 유효한 실수라면, 승산 결과를 변수 d에 대입합니다.

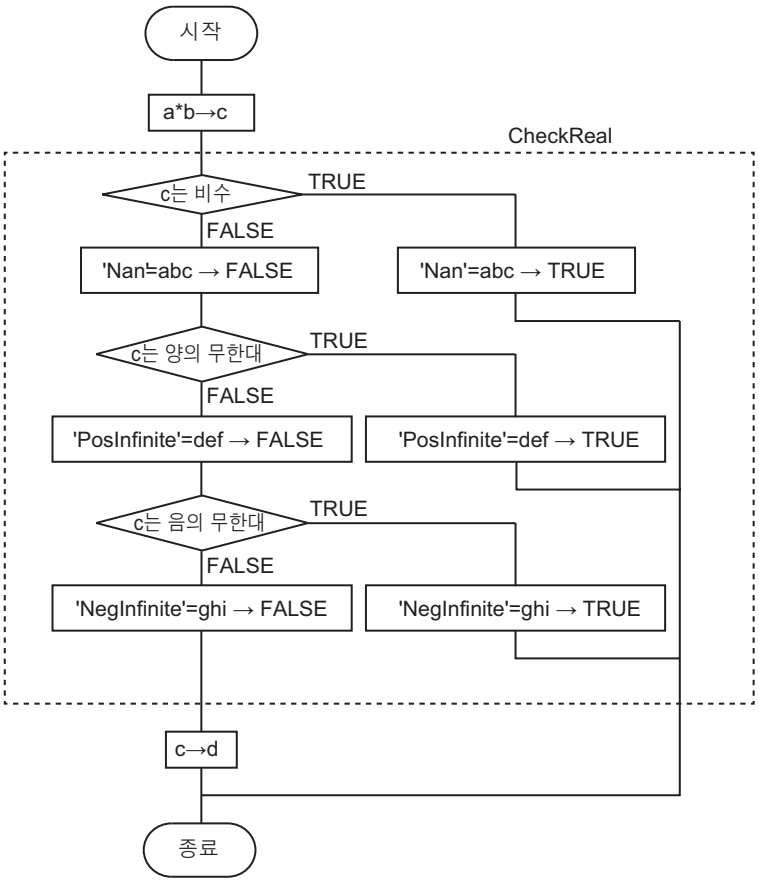
LD



ST

```
c:=a*b;
CheckReal(c, abc, def, ghi);
IF ( (abc=FALSE) AND (def=FALSE) AND (ghi=FALSE) ) THEN
    d:=c;
END_IF;
```

a와 b의 곱 c가 비수, 양의 무한대, 음의 무한대 중 하나가 아니면 c를 d에 대입합니다.



참고

이 명령은 실수를 사용한 연산 명령의 결과가 비수, 양의 무한대, 음의 무한대인지를 감지할 때 사용하십시오.

사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 'In'에 정수 형식의 파라메터를 전달한 경우, 다음과 같이 데이터 형식이 변환되어 입력됩니다.

'In'에 전달하는 파라메터의 데이터 형식	'In'의 데이터 형식
USINT, UINT, SINT, INT	REAL
UDINT, DINT	LREAL
ULINT, LINT	빌드 시에 이상

BCD 변환 명령

명령	명칭	페이지
_BCD_TO_*	BCD→부호가 없는 정수 변환 그룹	2-230
_TO_BCD_*	부호가 없는 정수→BCD 변환 그룹	2-233
BCD_TO_**	BCD군→부호가 없는 정수 변환 그룹	2-236
BCDsToBin	부호가 있는 BCD→부호가 있는 정수 변환	2-239
BinToBCDs_**	부호가 있는 정수→BCD 변환 그룹	2-242
AryToBCD	배열 일괄 BCD 변환	2-245
AryToBin	배열 일괄 부호가 없는 정수 변환	2-247

_BCD_TO_*

BCD의 비트 열을 부호가 없는 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_BCD_TO_*	BCD→부호가 없는 정수 변환 그룹	FUN	(@)**_BCD_TO_*** EN ENO In Out 단, **는 비트 열의 데이터 형식명, ***는 정수의 데이터 형식명	Out:=**_BCD_TO_*** (In); 단, **는 비트 열의 데이터 형식명, ***는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

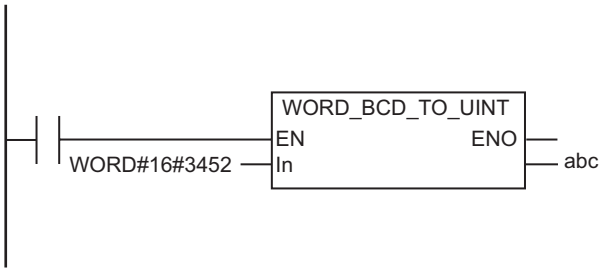
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out						○	○	○	○	○	○	○	○							

기능

BCD의 비트 열인 변환 대상 'In'을 부호가 없는 정수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 WORD 형식, 'Out'이 UINT 형식인 경우, 명령 이름은 WORD_BCD_TO_UINT가 됩니다.

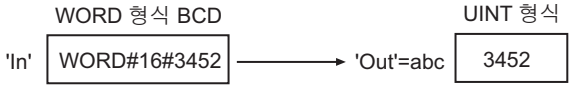
WORD_BCD_TO_UINT 명령에서 'In'=WORD#16#3452인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=WORD_BCD_TO_UINT(WORD#16#3452);
```



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
BYTE	USINT	16#00~16#99(BCD)	0~99
	UINT		
	UDINT		
	ULINT		
	SINT		
	INT		
	DINT		
	LINT		
WORD	USINT	16#0000~16#0255(BCD)	0~255
	UINT	16#0000~16#9999(BCD)	0~9999
	UDINT		
	ULINT		
	SINT	16#0000~16#0127(BCD)	0~127
	INT	16#0000~16#9999(BCD)	0~9999
	DINT		
	LINT		
DWORD	USINT	16#0000_0000~16#0000_0255(BCD)	0~255
	UINT	16#0000_0000~16#0006_5535(BCD)	0~65535
	UDINT	16#0000_0000~16#9999_9999(BCD)	0~99999999
	ULINT		
	SINT	16#0000_0000~16#0000_0127(BCD)	0~127
	INT	16#0000_0000~16#0003_2767(BCD)	0~32767
	DINT	16#0000_0000~16#9999_9999(BCD)	0~99999999
	LINT		
LWORD	USINT	16#0000_0000_0000_0000~ 16#0000_0000_0000_0255(BCD)	0~255
	UINT	16#0000_0000_0000_0000~ 16#0000_0000_0006_5535(BCD)	0~65535
	UDINT	16#0000_0000_0000_0000~ 16#0000_0042_9496_7295(BCD)	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)	0~9999999999999999
	SINT	16#0000_0000_0000_0000~ 16#0000_0000_0000_0127(BCD)	0~127
	INT	16#0000_0000_0000_0000~ 16#0000_0000_0003_2767(BCD)	0~32767
	DINT	16#0000_0000_0000_0000~ 16#0000_0021_4748_3647(BCD)	0~2147483647
	LINT	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)	0~9999999999999999

참고

- 임의의 BCD 비트 열에서 정수로 변환하려면 'BCD_TO_** 명령(2-236페이지)'을 사용하십시오.
- 정수에서 BCD 비트 열로 변환하려면 '**_TO_BCD_*** 명령(2-233페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자릿수에 0이 채워집니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우
  - 'In'의 값이 BCD의 비트 열이 아닌(16진수 표기로 A, B, C, D, E, F 중 어느 것이 포함되는) 경우

_TO_BCD_*

부호가 없는 정수를 BCD의 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_BCD_*	부호가 없는 정수→BCD 변환 그룹	FUN	(@)**_TO_BCD_*** EN ENO In Out 단, **는 정수의 데이터 형식명, ***는 비트 열의 데이터 형식명	Out:=**_TO_BCD_*** (In); 단, **는 정수의 데이터 형식명, ***는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

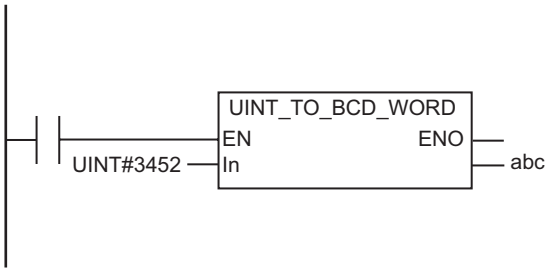
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Out		○	○	○	○															

기능

부호가 없는 정수의 변환 대상 'In'을 BCD의 비트 열로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면, 'In'이 UINT 형식, 'Out'이 WORD 형식인 경우, 명령 이름은 UINT_TO_BCD_WORD가 됩니다.

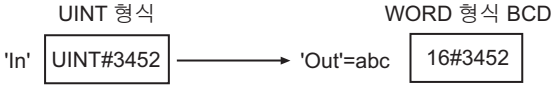
UINT_TO_BCD_WORD 명령에서 'In'=UINT#3452인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=UINT_TO_BCD_WORD(UINT#3452);
```



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
USINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~255	16#0000~16#0255(BCD)
	DWORD		16#0000_0000 ~ 16#0000_0255(BCD)
	LWORD		16#0000_0000_0000_0000~ 16#0000_0000_0000_0255(BCD)
UINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~65535	16#0000_0000 ~ 16#0006_5535(BCD)
	LWORD		16#0000_0000_0000_0000~ 16#0000_0000_0006_5535(BCD)
UDINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~99999999	16#0000_0000~16#9999_9999(BCD)
	LWORD	0~4294967295	16#0000_0000_0000_0000~ 16#0000_0042_9496_7295(BCD)
ULINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~99999999	16#0000_0000~16#9999_9999(BCD)
	LWORD	0~9999999999999999	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)
SINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~127	16#0000~16#0127(BCD)
	DWORD		16#0000_0000 ~ 16#0000_0127(BCD)
	LWORD		16#0000_0000_0000_0000~ 16#0000_0000_0000_0127(BCD)
INT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~32767	16#0000_0000 ~ 16#0003_2767(BCD)
	LWORD		16#0000_0000_0000_0000~ 16#0000_0000_0003_2767(BCD)
DINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~99999999	16#0000_0000~16#9999_9999(BCD)
	LWORD	0~2147483647	16#0000_0000_0000_0000~ 16#0000_0021_4748_3647(BCD)
LINT	BYTE	0~99	16#00~16#99(BCD)
	WORD	0~9999	16#0000~16#9999(BCD)
	DWORD	0~99999999	16#0000_0000~16#9999_9999(BCD)
	LWORD	0~9999999999999999	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)

참고

- 특정한 BCD 비트 열에서 정수로 변환하려면 '**_BCD_TO_*** 명령(2-230페이지)'을 사용하십시오.
- 임의의 BCD 비트 열에서 정수로 변환하려면 'BCD_TO_** 명령(2-236페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자릿수에 0이 채워집니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우

BCD_TO_**

BCD의 비트 열을 부호가 없는 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
BCD_TO_**	BCD군→부호가 없는 정수 변환 그룹	FUN	(@)BCD_TO_** EN ENO In Out 단, **는 정수의 데이터 형식명	Out:=BCD_TO_** (In); 단, **는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*1)	–	(*2)
Out	변환 결과	출력	변환 결과	(*1)	–	–

*1 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

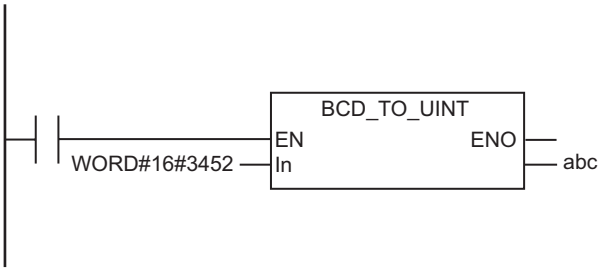
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out						○	○	○	○	○	○	○	○							

기능

BCD의 비트 열인 변환 대상 'In'을 부호가 없는 정수로 변환합니다.
명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 UINT 형식인 경우, 명령 이름은 BCD_TO_UINT가 됩니다.

BCD_TO_UINT 명령에서 'In'=WORD#16#3452인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=BCD_TO_UINT(WORD#16#3452);
```



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
BYTE	USINT	16#00~16#99(BCD)	0~99
	UINT		
	UDINT		
	ULINT		
	SINT		
	INT		
	DINT		
	LINT		
WORD	USINT	16#0000~16#0255(BCD)	0~255
	UINT	16#0000~16#9999(BCD)	0~9999
	UDINT		
	ULINT		
	SINT	16#0000~16#0127(BCD)	0~127
	INT	16#0000~16#9999(BCD)	0~9999
	DINT		
	LINT		
DWORD	USINT	16#0000_0000~16#0000_0255(BCD)	0~255
	UINT	16#0000_0000~16#0006_5535(BCD)	0~65535
	UDINT	16#0000_0000~16#9999_9999(BCD)	0~99999999
	ULINT		
	SINT	16#0000_0000~16#0000_0127(BCD)	0~127
	INT	16#0000_0000~16#0003_2767(BCD)	0~32767
	DINT	16#0000_0000~16#9999_9999(BCD)	0~99999999
	LINT		
LWORD	USINT	16#0000_0000_0000_0000~ 16#0000_0000_0000_0255(BCD)	0~255
	UINT	16#0000_0000_0000_0000~ 16#0000_0000_0006_5535(BCD)	0~65535
	UDINT	16#0000_0000_0000_0000~ 16#0000_0042_9496_7295(BCD)	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)	0~9999999999999999
	SINT	16#0000_0000_0000_0000~ 16#0000_0000_0000_0127(BCD)	0~127
	INT	16#0000_0000_0000_0000~ 16#0000_0000_0003_2767(BCD)	0~32767
	DINT	16#0000_0000_0000_0000~ 16#0000_0021_4748_3647(BCD)	0~2147483647
	LINT	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)	0~9999999999999999

참고

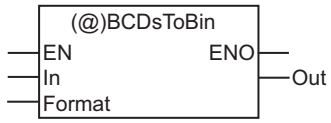
- 특정한 BCD 비트 열에서 정수로 변환하려면 '**_BCD_TO_*** 명령(2-230페이지)'을 사용하십시오.
- 정수에서 BCD 비트 열로 변환하려면 '**_TO_BCD_*** 명령(2-233페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자릿수에 0이 채워집니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우
  - 'In'의 값이 BCD의 비트 열이 아닌(16진수 표기로 A, B, C, D, E, F 중 어느 것이 포함되는) 경우

BCDsToBin

부호가 있는 BCD의 비트 열을 부호가 있는 정수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
BCDsToBin	부호가 있는 BCD→부호가 있는 정수 변환	FUN		Out:=BCDsToBin(In, Format);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*1)	–	(*2)
Format	데이터 형식 지정 번호		BCD의 비트 열의 형식	_BCD0~_BCD3		_BCD0
Out	변환 결과	출력	변환 결과	(*1)	–	–

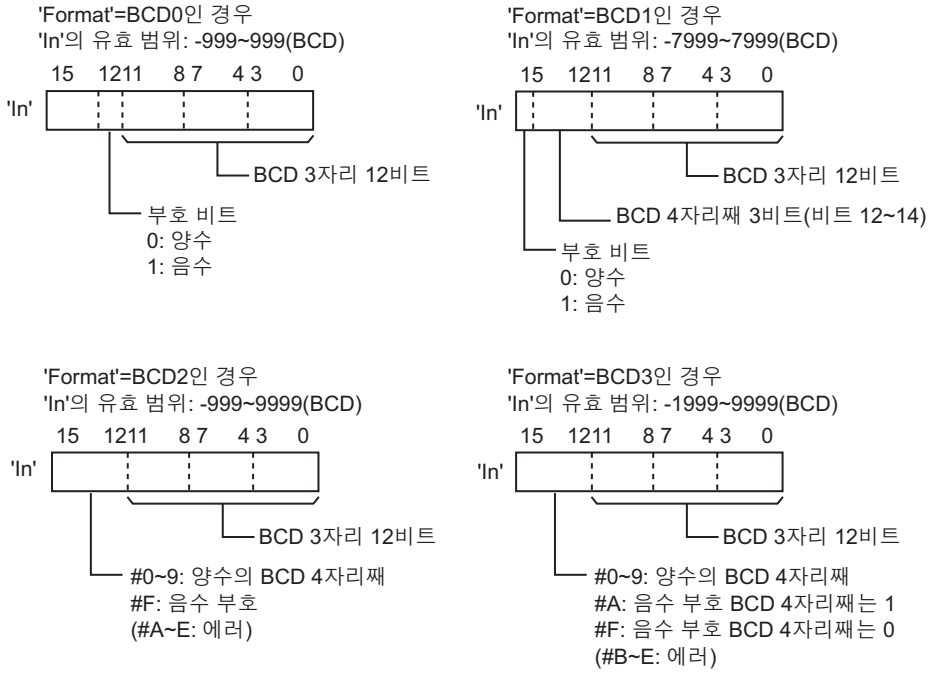
*1 'Format'의 값에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Format	열거 형식 _eBCD_FORMAT 열거자는 기능 설명 참조																			
Out	'In'과 동일한 크기의, 부호가 있는 정수의 데이터 형식																			

기능

부호가 있는 BCD의 비트 열 'In'을 부호가 있는 정수로 변환합니다.

데이터 형식 지정 번호 'Format'의 데이터 형식은 열거 형식 _eBCD_FORMAT입니다. 열거자는 _BCD0, _BCD1, _BCD2, _BCD3의 4가지입니다. 이 값에 따라, 'In'의 최상위 4비트의 부호 표현 방식이 변합니다. 'In'이 WORD 형식인 경우를 예로 들어, 데이터 형식이 아래에 나와 있습니다.

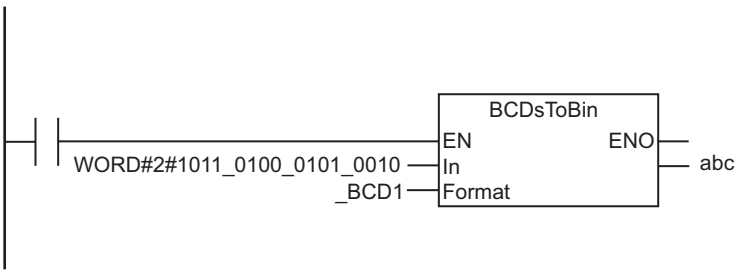


'In'과 'Out' 데이터 형식은 같은 크기로 합니다. 각각의 유효 범위는 'Format'의 값에 따라 다음과 같이 변합니다.

		'Format'의 값			
		_BCD0	**_BCD1**	**_BCD2**	**_BCD3**
'In'의 데이터 형식 ↓ 'Out'의 데이터 형식	**BYTE ↓ SINT**	-9~9	-79~79	-9~99	-19~99
	WORD ↓ INT	-999~999	-7999~7999	-999~9999	-1999~9999
	DWORD ↓ DINT	-9999999~9999999	-79999999~79999999	-9999999~99999999	-19999999~99999999
	LWORD ↓ LINT	-999999999999999 ~ 999999999999999	-7999999999999999 ~ 7999999999999999	-999999999999999 ~ 9999999999999999	-1999999999999999 ~ 9999999999999999

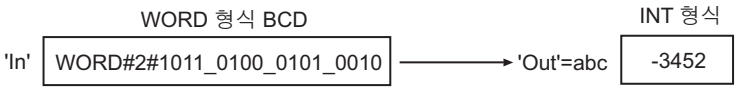
'In'=WORD#2#1011_0100_0101_0010, 'Format'=_BCD1인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=BCDsToBin(WORD#2#1011_0100_0101_0010,
_BCD1);
```



사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식의 크기를 동일하게 하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Format'의 값이 _BCD0이고 'In'의 최상위 자릿수가 2~F인 경우
  - 'Format'의 값이 _BCD2이고 'In'의 최상위 자릿수가 A~E인 경우
  - 'Format'의 값이 _BCD3이고 'In'의 최상위 자릿수가 B~E인 경우
  - 상기 조건을 제외하고 'In' 중 어느 것의 자릿수가 A~F인 경우
  - 'Format'의 값이 유효 범위 밖인 경우

BinToBCDs_**

부호가 있는 정수를 부호가 있는 BCD의 비트 열로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
BinToBCDs_ **	부호가 있는 정수→BCD 변환 그룹	FUN	(@)BinToBCDs_** EN ENO In Out Format 단, **는 비트 열의 데이터 형식명	Out:=BinToBCDs_**(In, Format); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Format	데이터 형식 지정 번호		BCD의 비트 열의 형식	_BCD0~_BCD3		_BCD0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'Format'의 값에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

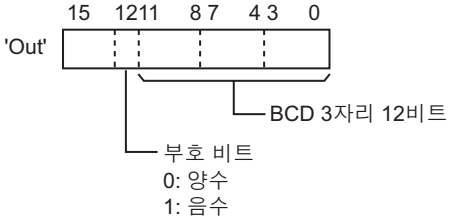
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In										○	○	○	○							
Format	열거 형식 _eBCD_FORMAT 열거자는 기능 설명 참조																			
Out	'In'과 동일한 크기의 비트 열의 데이터 형식																			

기능

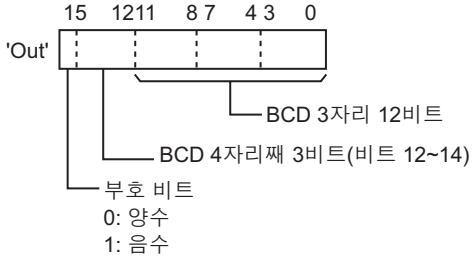
부호가 있는 정수 'In'을 부호가 있는 BCD의 비트 열로 변환합니다.
명령 이름은 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 WORD 형식인 경우, 명령 이름은 BinToBCDs_WORD가 됩니다.

데이터 형식 지정 번호 'Format'의 데이터 형식은 열거 형식 _eBCD_FORMAT입니다. 열거자는 _BCD0, _BCD1, _BCD2, _BCD3의 4가지입니다. 이 값에 따라, 'Out'의 최상위 4비트의 부호 표현 방식이 변합니다. 'Out'이 WORD 형식인 경우를 예로 들어, 데이터 형식이 아래에 나와 있습니다.

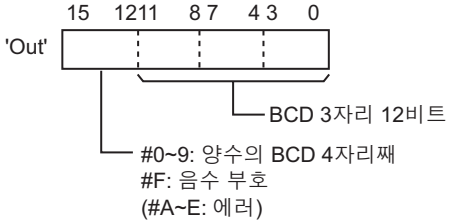
'Format'=BCD0인 경우
'Out'의 유효 범위: -999~999(BCD)



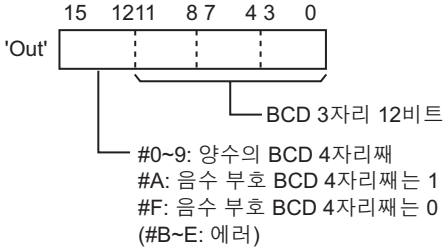
'Format'=BCD1인 경우
'Out'의 유효 범위: -7999~7999(BCD)



'Format'=BCD2인 경우
'Out'의 유효 범위: -999~9999(BCD)



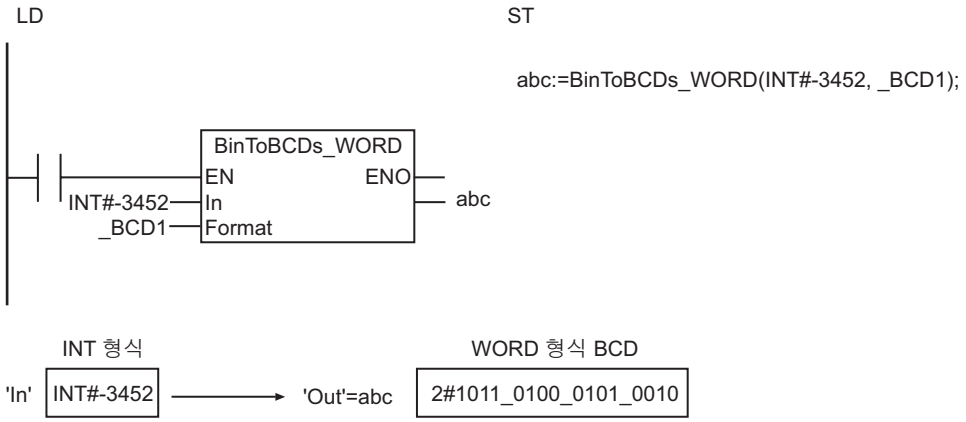
'Format'=BCD3인 경우
'Out'의 유효 범위: -1999~9999(BCD)



'In'과 'Out' 데이터 형식은 같은 크기로 합니다. 각각의 유효 범위는 'Format'의 값에 따라 다음과 같이 변합니다.

		'Format'의 값			
		_BCD0	_BCD1	_BCD2	_BCD3
'In'의 데이터 형식 ↓ 'Out'의 데이터 형식	SINT ↓ BYTE	-9~9	-79~79	-9~99	-19~99
	INT ↓ WORD	-999~999	-7999~7999	-999~9999	-1999~9999
	DINT ↓ DWORD	-9999999~9999999	-79999999~79999999	-9999999~99999999	-19999999~99999999
	LINT ↓ LWORD	-999999999999999 ~ 999999999999999	-7999999999999999 ~ 7999999999999999	-999999999999999 ~ 9999999999999999	-1999999999999999 ~ 9999999999999999

BinToBCDs_WORD 명령에서 'In'=INT#-3452, 'Format'=_BCD1인 경우의 예가 아래에 나와 있습니다.

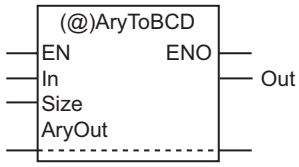


사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우
  - 'Format'의 값이 유효 범위 밖인 경우

AryToBCD

부호가 없는 정수 배열의 각 요소를 BCD의 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AryToBCD	배열 일괄 BCD 변환	FUN		AryToBCD(In, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	부호가 없는 정수 배열	입력	부호가 없는 정수 배열	(*1)	–	(*2)
Size	요소 수		변환하는 In[]의 요소 수	데이터 형식에 따름		1
AryOut[] 배열	BCD 배열	입출력	BCD 배열	(*1)	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

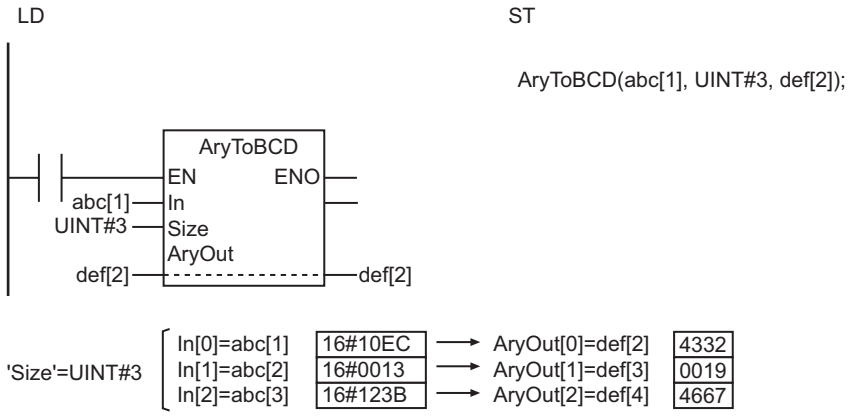
*1 In[], AryOut[]의 요소의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열						○	○	○	○											
Size							○													
AryOut[] 배열	비트 열의 배열 단, In[]의 요소와 같은 크기의 데이터 형식																			
Out	○																			

기능

부호가 없는 정수 배열 In[]의, In[0]부터 'Size'개의 각 요소를 BCD의 비트 열로 변환해서 BCD 배열 AryOut[]에 출력합니다.

'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.



In[]과 AryOut[]의 유효 범위는 요소의 데이터 형식의 조합으로 다음과 같이 변합니다.

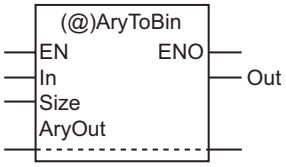
In[]의 요소의 데이터 형식	AryOut[]의 요소의 데이터 형식	In[]의 유효 범위	AryOut[]의 유효 범위
USINT	BYTE	0~99	16#00~16#99(BCD)
UINT	WORD	0~9999	16#0000~16#9999(BCD)
UDINT	DWORD	0~99999999	16#0000_0000~16#9999_9999(BCD)
ULINT	LWORD	0~9999999999999999	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)

사용상의 주의 사항

- In[]의 요소와 AryOut[]의 요소의 데이터 형식은 같은 크기로 하십시오. 예를 들면, In[]의 요소가 UINT 형식인 경우에는 AryOut[]의 요소를 WORD 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 부호가 있는 BCD 변환은 할 수 없습니다. 따라서 In[]의 데이터 형식은 부호가 없는 정수(USINT, UINT, UDINT, ULINT) 중에서 선택하십시오.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - In[]의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[] 또는 AryOut[]의 배열 영역을 초과한 경우

AryToBin

BCD의 비트 열 배열의 각 요소를 부호가 없는 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AryToBin	배열 일괄 부호가 없는 정수 변환	FUN		AryToBin(In, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	BCD의 비트 열 배열	입력	BCD의 비트 열 배열	(*1)	–	(*2)
Size	요소 수		변환하는 In[]의 요소 수	데이터 형식에 따름		1
AryOut[] 배열	부호가 없는 정수 배열	입출력	부호가 없는 정수 배열	(*1)	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

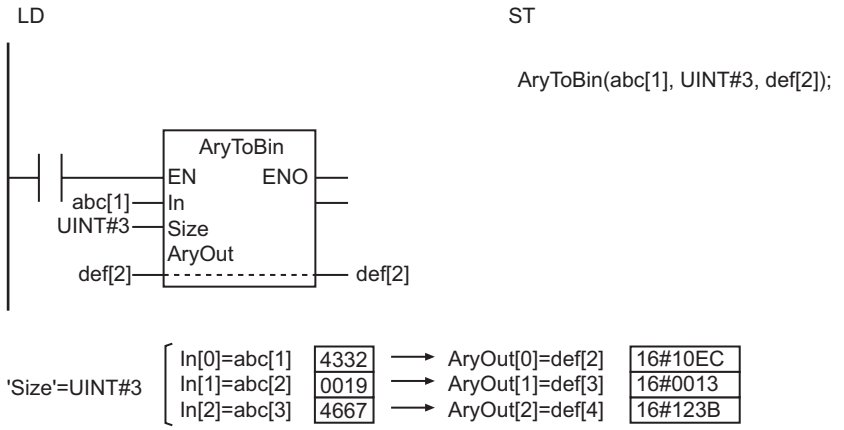
*1 In[], AryOut[]의 요소의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○	○	○	○															
Size							○													
AryOut[] 배열	부호가 없는 정수의 배열 단, In[]의 요소와 같은 크기의 데이터 형식																			
Out	○																			

기능

BCD의 비트 열 배열 In[]의, In[0]부터 'Size'개의 각 요소를 부호가 없는 정수로 변환해서 부호가 없는 정수 배열 AryOut[]에 출력합니다.

'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.



In[]과 AryOut[]의 유효 범위는 요소의 데이터 형식의 조합으로 다음과 같이 변합니다.

In[]의 요소의 데이터 형식	AryOut[]의 요소의 데이터 형식	In[]의 유효 범위	AryOut[]의 유효 범위
BYTE	USINT	16#00~16#99(BCD)	0~99
WORD	UINT	16#0000~16#9999(BCD)	0~9999
DWORD	UDINT	16#0000_0000~ 16#9999_9999(BCD)	0~99999999
LWORD	ULINT	16#0000_0000_0000_0000~ 16#9999_9999_9999_9999(BCD)	0~9999999999999999

사용상의 주의 사항

- In[]과 AryOut[]의 데이터 형식은 같은 크기로 하십시오. 예를 들면, In[]의 요소가 WORD 형식인 경우에는 AryOut[]의 요소를 USINT 형식으로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 부호가 있는 BCD 변환은 할 수 없습니다. 따라서 AryOut[]의 데이터 형식은 부호가 없는 정수(USINT, UINT, UDINT, ULINT) 중에서 선택하십시오.
- 'Size'의 값이 0인 경우, [AryOut]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In[] 또는 AryOut[]의 배열 영역을 초과한 경우
  - In[]의 값이 BCD의 비트 열이 아닌(16진수 표기로 A, B, C, D, E, F 중 어느 것이 포함되는) 경우

데이터 형식 변환 명령

명령	명칭	페이지	명령	명칭	페이지
_TO_*(정수→정수 변환 그룹)	정수→정수 변환 그룹	2-250	RealToFormatString	실수(REAL)→서식 있는 문자열 변환	2-276
_TO_*(정수→비트 열 변환 그룹)	정수→비트 열 변환 그룹	2-253	LrealToFormatString	실수(LREAL)→서식 있는 문자열 변환	2-281
_TO_*(정수→실수 변환 그룹)	정수→실수 변환 그룹	2-255	STRING_TO_**(문자열→정수 변환 그룹)	문자열→정수 변환 그룹	2-286
_TO_*(비트 열→정수 변환 그룹)	비트 열→정수 변환 그룹	2-257	STRING_TO_**(문자열→비트 열 변환 그룹)	문자열→비트 열 변환 그룹	2-288
_TO_*(비트 열→비트 열 변환 그룹)	비트 열→비트 열 변환 그룹	2-259	STRING_TO_**(문자열→실수 변환 그룹)	문자열→실수 변환 그룹	2-290
_TO_*(비트 열→실수 변환 그룹)	비트 열→실수 변환 그룹	2-261	TO_**(정수 변환 그룹)	정수 변환 그룹	2-293
_TO_*(실수→정수 변환 그룹)	실수→정수 변환 그룹	2-263	TO_**(비트 열 변환 그룹)	비트 열 변환 그룹	2-295
_TO_*(실수→비트 열 변환 그룹)	실수→비트 열 변환 그룹	2-266	TO_**(실수 변환 그룹)	실수 변환 그룹	2-297
_TO_*(실수→실수 변환 그룹)	실수→실수 변환 그룹	2-268	EnumToNum	열거 형식→정수 변환	2-299
**_TO_STRING(정수→문자열 변환 그룹)	정수→문자열 변환 그룹	2-270	NumToEnum	정수→열거 형식 변환	2-301
**_TO_STRING(비트 열→문자열 변환 그룹)	비트 열→문자열 변환 그룹	2-272	TRUNC/Round/RoundUp	실수 버림/실수 반올림/실수 올림	2-303
**_TO_STRING(실수→문자열 변환 그룹)	실수→문자열 변환 그룹	2-274			

_TO_*(정수→정수 변환 그룹)

정수를 다른 데이터 형식의 정수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
_TO_*	정수→정수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **, ***는 다른 정수의 데이터 형식명	Out:=**_TO_*** (In); 단, **, ***는 다른 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

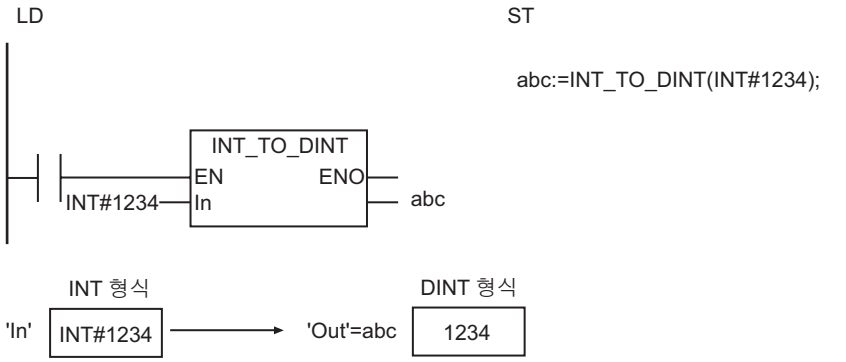
* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Out						○	○	○	○	○	○	○	○							

기능

정수 'In'을 다른 데이터 형식의 정수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 INT 형식이고 'Out'이 DINT 형식인 경우, 명령 이름은 INT_TO_DINT가 됩니다.

INT_TO_DINT 명령에서 'In'=INT#1234인 예가 아래에 나와 있습니다.



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In', 'Out'의 유효 범위
USINT	UINT	0~255
	UDINT	
	ULINT	
	SINT	0~127
	INT	0~255
	DINT	
	LINT	
UINT	USINT	0~255
	UDINT	0~65535
	ULINT	
	SINT	0~127
	INT	0~32767
	DINT	0~65535
	LINT	
UDINT	USINT	0~255
	UINT	0~65535
	ULINT	0~4294967295
	SINT	0~127
	INT	0~32767
	DINT	0~2147483647
	LINT	0~4294967295
ULINT	USINT	0~255
	UINT	0~65535
	UDINT	0~4294967295
	SINT	0~127
	INT	0~32767
	DINT	0~2147483647
	LINT	0~9223372036854775807
SINT	USINT	0~127
	UINT	
	UDINT	
	ULINT	
	INT	-128~127
	DINT	
	LINT	
INT	USINT	0~255
	UINT	0~32767
	UDINT	
	ULINT	
	SINT	-128~127
	DINT	-32768~32767
	LINT	
DINT	USINT	0~255
	UINT	0~65535
	UDINT	0~2147483647
	ULINT	
	SINT	-128~127
	INT	-32768~32767
	LINT	-2147483648~2147483647

'In'의 데이터 형식	'Out'의 데이터 형식	'In', 'Out'의 유효 범위
LINT	USINT	0~255
	UINT	0~65535
	UDINT	0~4294967295
	ULINT	0~9223372036854775807
	SINT	-128~127
	INT	-32768~32767
	DINT	-2147483648~2147483647

참고

임의의 데이터 형식에서 정수 형식으로 변환하려면 📖 'TO_**(정수 변환 그룹) 명령(2-293페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'이 부호가 있는 정수이고 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 부호 확장을 수행합니다.
- 'In'이 부호가 없는 정수이고 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자리에 0이 채워집니다.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 작은 경우, 상위 자리를 버립니다.

_TO_*(정수→비트 열 변환 그룹)

정수를 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	정수→비트 열 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 정수의 데이터 형식명, ***는 비트 열의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 정수의 데이터 형식명, ***는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

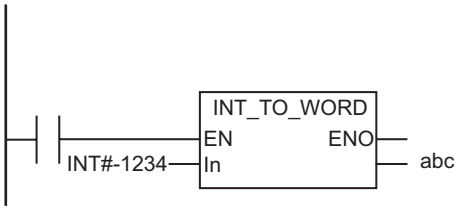
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Out		○	○	○	○															

기능

정수 'In'을 비트 열로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 INT 형식이고 'Out'이 WORD 형식인 경우, 명령 이름은 INT_TO_WORD가 됩니다.

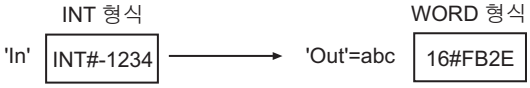
INT_TO_WORD 명령에서 'In'=INT#-1234인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=INT_TO_WORD(INT#-1234);
```



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
USINT	BYTE	0~255	16#00~16#FF
	WORD		
	DWORD		
	LWORD		
UINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD		
	LWORD		
UDINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4294967295	16#0000_0000~16#FFFF_FFFF
	LWORD		
ULINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4294967295	16#0000_0000~16#FFFF_FFFF
	LWORD	0~18446744073709551645	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF
SINT	BYTE	-128~127	16#00~16#FF
	WORD		
	DWORD		
	LWORD		
INT	BYTE	-128~127	16#00~16#FF
	WORD	-32768~32767	16#0000~16#FFFF
	DWORD		
	LWORD		
DINT	BYTE	-128~127	16#00~16#FF
	WORD	-32768~32767	16#0000~16#FFFF
	DWORD	-2147483648~2147483647	16#0000_0000~16#FFFF_FFFF
	LWORD		
LINT	BYTE	-128~127	16#00~16#FF
	WORD	-32768~32767	16#0000~16#FFFF
	DWORD	-2147483648~2147483647	16#0000_0000~16#FFFF_FFFF
	LWORD	-9223372036854775808~ 9223372036854775807	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF

참고

- 비트 열에서 정수로 변환하려면 📖 '**_TO_***(비트 열→정수 변환 그룹) 명령(2-257페이지)'을 사용하십시오.
- 임의의 데이터 형식에서 비트 열로 변환하려면 📖 'TO_**(비트 열 변환 그룹) 명령(2-295페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'이 부호가 있는 정수이고 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 부호 확장을 수행합니다.
- 'In'이 부호가 없는 정수이고 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자리에 0이 채워집니다.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 작은 경우, 상위 자리를 버립니다.

_TO_*(정수→실수 변환 그룹)

정수를 실수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	정수→실수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 정수의 데이터 형식명, ***는 실수의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 정수의 데이터 형식명, ***는 실수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

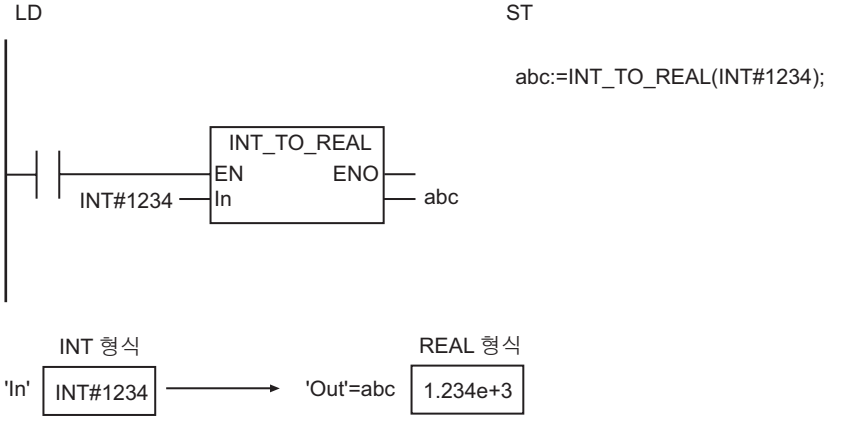
* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Out														○	○					

기능

정수 'In'을 실수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 INT 형식이고 'Out'이 REAL 형식인 경우, 명령 이름은 INT_TO_REAL이 됩니다.

INT_TO_REAL 명령에서 'In'=INT#1234인 경우의 예가 아래에 나와 있습니다.



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
USINT	REAL	0~255	0~2.55e+2
	LREAL		
UINT	REAL	0~65535	0~6.5535e+4
	LREAL		
UDINT	REAL	0~4294967295	0~4.294967e+9
	LREAL		0~4.294967295e+9
ULINT	REAL	0~18446744073709551615	0~1.844674e+19
	LREAL		0~1.84467440737095e+19
SINT	REAL	-128~127	-1.28e+2~1.27e+2
	LREAL		
INT	REAL	-32768~32767	-3.2768e+4~3.2767e+4
	LREAL		
DINT	REAL	-2147483648~2147483647	-2.147483e+9~2.147483e+9
	LREAL		-2.147483648e+9~2.147483647e+9
LINT	REAL	-9223372036854775808~9223372036854775807	-9.223372e+18~9.223372e+18
	LREAL		-9.22337203685477e+18~9.22337203685477e+18

참고

- 실수에서 정수로 변환하려면 '**_TO_***(실수→정수 변환 그룹) 명령(2-263페이지)'을 사용하십시오.
- 임의의 데이터 형식에서 실수로 변환하려면 'TO_**(실수 변환 그룹) 명령(2-297페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'과 'Out'의 데이터 형식의 조합에 따라 실수의 유효 자리에 대응한 반올림이 발생하므로, 변환 전후의 값에 차이가 생길 수 있습니다. 차이가 발생하는 데이터 형식의 조합은 다음과 같습니다.

'In'의 데이터 형식	'Out'의 데이터 형식	차이가 발생하는 값
DINT	REAL	-16777216 이하 또는 16777216 이상
LINT		
UDINT	REAL	16777216 이상
ULINT		
LINT	LREAL	-9007199254740992 이하 또는 9007199254740992 이상
ULINT	LREAL	9007199254740992 이상

_TO_*(비트 열→정수 변환 그룹)

비트 열을 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	비트 열→정수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 비트 열의 데이터 형식명, ***는 정수의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 비트 열의 데이터 형식명, ***는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

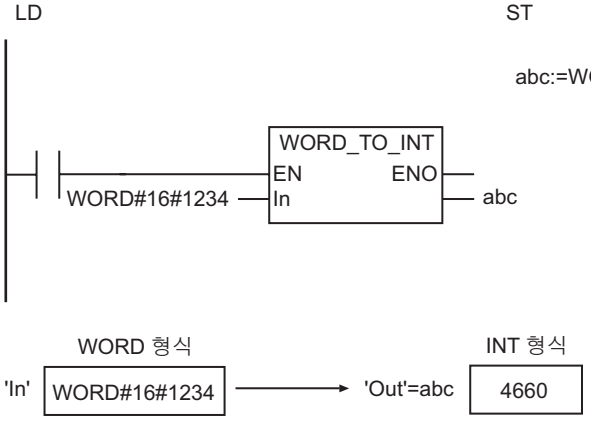
* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out						○	○	○	○	○	○	○	○							

기능

비트 열 'In'을 정수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 WORD 형식, 'Out'이 INT 형식인 경우, 명령 이름은 WORD_TO_INT가 됩니다.

WORD_TO_INT 명령에서 'In'=WORD#16#1234인 경우의 예가 아래에 나와 있습니다.



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
BYTE	USINT	16#00~16#FF	0~255
	UINT		
	UDINT		
	ULINT		
	SINT		-128~127
	INT		
	DINT		
	LINT		
WORD	USINT	16#00~16#FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT		
	ULINT		
	SINT	16#00~16#FF	-128~127
	INT	16#0000~16#FFFF	-32768~32767
	DINT		
	LINT		
DWORD	USINT	16#00~16#FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT		
	SINT	16#00~16#FF	-128~127
	INT	16#0000~16#FFFF	-32768~32767
	DINT	16#0000_0000~16#FFFF_FFFF	-2147483648~2147483647
	LINT		
LWORD	USINT	16#00~16#FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#00~16#FF	-128~127
	INT	16#0000~16#FFFF	-32768~32767
	DINT	16#0000_0000~16#FFFF_FFFF	-2147483648~2147483647
	LINT	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~ 9223372036854775807

참고

- 정수에서 비트 열로 변환하려면 '**_TO_***(정수→비트 열 변환 그룹) 명령(2-253페이지)'을 사용하십시오.
- 임의의 데이터 형식에서 비트 열로 변환하려면 'TO_**(비트 열 변환 그룹) 명령(2-295페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자리에 0이 채워집니다.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 작은 경우, 상위 자리를 버립니다.

_TO_*(비트 열→비트 열 변환 그룹)

비트 열을 다른 데이터 형식의 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	비트 열→비트 열 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **, ***는 다른 비트 열의 데이터 형식명	Out:=**_TO_*** (In); 단, **, ***는 다른 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

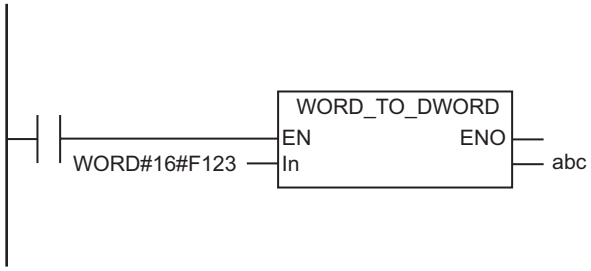
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out		○	○	○	○															

기능

비트 열 'In'을 다른 데이터 형식의 비트 열로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 WORD 형식, 'Out'이 DWORD 형식인 경우, 명령 이름은 WORD_TO_DWORD가 됩니다.

WORD_TO_DWORD 명령에서 'In'=WORD#16#F123인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=WORD_TO_DWORD(WORD#16#F123);
```



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In', 'Out'의 유효 범위
BYTE	WORD	16#00~16#FF
	DWORD	
	LWORD	
WORD	BYTE	16#00~16#FF
	DWORD	16#0000~16#FFFF
	LWORD	
DWORD	BYTE	16#00~16#FF
	WORD	16#0000~16#FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF
LWORD	BYTE	16#00~16#FF
	WORD	16#0000~16#FFFF
	DWORD	16#0000_0000~16#FFFF_FFFF

참고

임의의 데이터 형식에서 비트 열로 변환하려면 🕮 'TO_**(비트 열 변환 그룹) 명령(2-295페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자리에 0이 채워집니다.
- 'In'의 데이터 크기보다 'Out'의 데이터 크기가 작은 경우, 상위 자리를 버립니다.

_TO_*(비트 열→실수 변환 그룹)

비트 열을 실수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	비트 열→실수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 비트 열의 데이터 형식명, ***는 실수의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 비트 열의 데이터 형식명, ***는 실수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

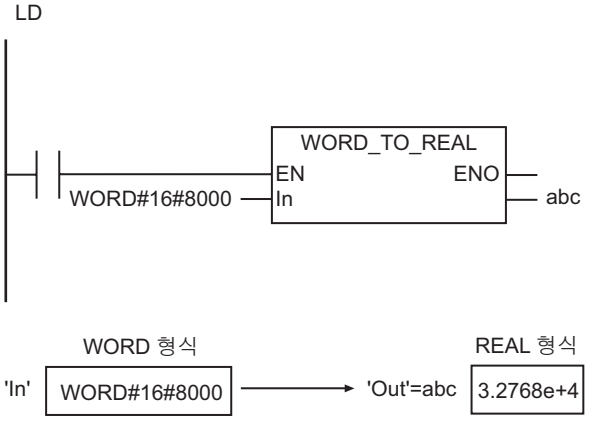
* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out														○	○					

기능

비트 열 'In'을 같은 크기의 부호가 없는 정수로 간주해서 실수 형식으로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 WORD 형식, 'Out'이 REAL 형식인 경우, 명령 이름은 WORD_TO_REAL이 됩니다.

WORD_TO_REAL 명령에서 'In'=WORD#16#8000인 경우의 예가 아래에 나와 있습니다.



ST

```
abc:=WORD_TO_REAL(WORD#16#8000);
```

'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
BYTE	REAL	16#00~16#FF	0~2.55e+2
	LREAL		
WORD	REAL	16#0000~16#FFFF	0~6.5535e+4
	LREAL		
DWORD	REAL	16#0000_0000~16#FFFF_FFFF	0~4.294967e+9
	LREAL		0~4.294967295e+9
LWORD	REAL	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~1.844674e+19
	LREAL		0~1.84467440737095e+19

참고

- 실수에서 비트 열로 변환하려면 📖 '**_TO_***(실수→비트 열 변환 그룹) 명령(2-266페이지)'을 사용하십시오.
- 임의의 데이터 형식에서 실수로 변환하려면 📖 'TO_**(실수 변환 그룹) 명령(2-297페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'과 'Out'의 데이터 형식의 조합에 따라 실수의 유효 자리에 대응한 반올림이 발생하므로, 변환 전후의 값에 차이가 생길 수 있습니다. 차이가 발생하는 데이터 형식의 조합은 다음과 같습니다.

'In'의 데이터 형식	'Out'의 데이터 형식	차이가 발생하는 값
DWORD	REAL	16#0100_0000 이상
LWORD	LREAL	16#0002_0000_0000_0000 이상

_TO_*(실수→정수 변환 그룹)

실수를 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
_TO_*	실수→정수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 실수의 데이터 형식명, ***는 정수의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 실수의 데이터 형식명, ***는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

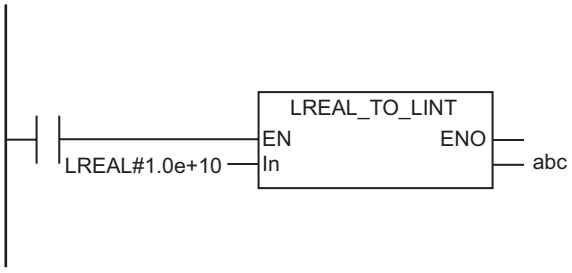
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out						○	○	○	○	○	○	○	○							

기능

실수 'In'을 정수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 LREAL 형식, 'Out'이 LINT 형식인 경우, 명령 이름은 LREAL_TO_LINT가 됩니다.

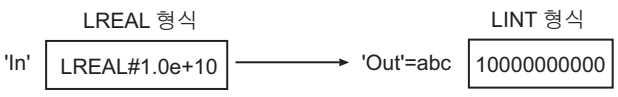
LREAL_TO_LINT 명령에서 'In'＝LREAL#1.0e+10인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=LREAL_TO_LINT(LREAL#1.0e+10);
```



'In'의 값의 소수점 이하는 오사오입해서 정수로 반올림합니다. 오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1 -1.49→-1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2 -1.50→-2 -2.50→-2
0.5보다 큼	올림	1.51→2 -1.51→-2

'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
REAL	USINT	0~2.55e+2	0~255
	UINT	0~6.5535e+4	0~65535
	UDINT	0~4.294967e+9	0~4294967295
	ULINT	0~1.844674e+19	0~18446744073709551615
	SINT	-1.28e+2~1.27e+2	-128~127
	INT	-3.2768e+4~3.2767e+4	-32768~32767
	DINT	-2.147483e+9~2.147483e+9	-2147483648~2147483647
	LINT	-9.223372e+18~9.223372e+18	-9223372036854775808~ -9223372036854775807
LREAL	USINT	0~2.55e+2	0~255
	UINT	0~6.5535e+4	0~65535
	UDINT	0~4.294967295e+9	0~4294967295
	ULINT	0~1.84467440737095e+19	0~18446744073709551615
	SINT	-1.28e+2~1.27e+2	-128~127
	INT	-3.2768e+4~3.2767e+4	-32768~32767
	DINT	-2.147483648e+9~2.147483647e+9	-2147483648~2147483647
	LINT	-9.22337203685477e+18~ 9.22337203685477e+18	-9223372036854775808~ -9223372036854775807

참고

- 정수에서 실수로 변환하려면 '**_TO_***(정수→실수 변환 그룹) 명령(2-255페이지)'을 사용하십시오.
- 임의의 데이터 형식에서 정수로 변환하려면 'TO_**(정수 변환 그룹) 명령(2-293페이지)'을 사용하십시오.
- 이 명령 이외에 실수→정수 변환하는 명령으로는 TRUNC(소수점 이하 버림), Round(소수점 이하 오사오입), RoundUp(소수점 이하 올림)이 있습니다. 모두 입력이 REAL 형식이라면 출력은 DINT 형식, 입력이 LREAL 형식이라면 출력은 LINT 형식입니다. 각 명령의 처리의 차이가 아래에 나와 있습니다.

입력값	출력값			
	REAL_TO_INT	TRUNC	Round	RoundUp
REAL#1.6	INT#2	DINT#1	DINT#2	DINT#2
REAL#1.5	INT#2	DINT#1	DINT#2	DINT#2
REAL#1.4	INT#1	DINT#1	DINT#1	DINT#2
REAL#2.5	INT#2	DINT#2	DINT#2	DINT#3
REAL#-1.6	INT#-2	DINT#-1	DINT#-2	DINT#-2

입력값	출력값			
	REAL_TO_INT	TRUNC	Round	RoundUp
REAL#-1.5	INT#-2	DINT#-1	DINT#-2	DINT#-2
REAL#-1.4	INT#-1	DINT#-1	DINT#-1	DINT#-2
REAL#-2.5	INT#-2	DINT#-2	DINT#-2	DINT#-3

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 변환 결과가 'Out'의 유효 범위를 초과하는 경우, , 'Out'의 값은 잘못된 값이 됩니다.

_TO_*(실수→비트 열 변환 그룹)

실수를 비트 열로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
_TO_*	실수→비트 열 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **는 실수의 데이터 형식명, ***는 비트 열의 데이터 형식명	Out:=**_TO_*** (In); 단, **는 실수의 데이터 형식명, ***는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

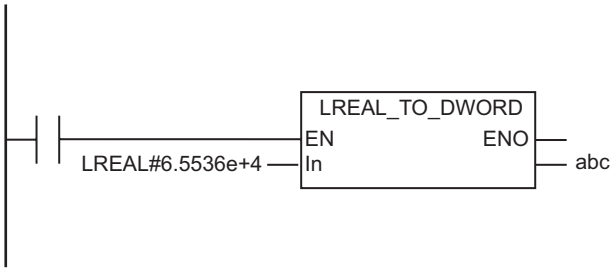
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out		○	○	○	○															

기능

실수 'In'을 비트 열로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 LREAL 형식, 'Out'이 DWORD 형식인 경우, 명령 이름은 LREAL_TO_DWORD가 됩니다.

LREAL_TO_DWORD 명령에서 'In'=LREAL#6.5536e+4인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=LREAL_TO_DWORD(LREAL#6.5536e+4);
```



변환 순서가 아래에 나와 있습니다.

1 **'In'의 값의 소수점 이하는 오사오입(뒤에 설명)해서 정수로 반올림합니다.**

2 **그 정수를, 부호가 없는 정수로서 비트 열로 표현합니다.**

오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1 -1.49→-1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2 -1.50→-2 -2.50→-2
0.5보다 큼	올림	1.51→2 -1.51→-2

몇 가지 변환 예를 게재합니다.

'In'의 값	정수화한 값	'Out'의 값
1.6	2	16#0002
3.5	4	16#0004
-1.6	-2	16#FFFE

'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In'의 유효 범위	'Out'의 유효 범위
REAL	BYTE	-1.285999e+2~1.274999e+2	16#00~16#FF
	WORD	-3.276859e+4~3.276749e+4	16#0000~16#FFFF
	DWORD	-2.147483e+9~2.147483e+9	16#0000_0000~16#FFFF_FFFF
	LWORD	-9.223372e+18~9.223372e+18	16#0000_0000_0000_0000~ 16#FFFF_FFEE_FFFF_FFFF
LREAL	BYTE	-1.28599999999999e+2~ 1.27499999999999e+2	16#00~16#FF
	WORD	-3.27685999999999e+4~ 3.27674999999999e+4	16#0000~16#FFFF
	DWORD	-2.14748364859999e+9~ 2.14748364749999e+9	16#0000_0000~16#FFFF_FFFF
	LWORD	-9.22337203685477e+18~ 9.22337203685477e+18	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF

참고

비트 열에서 실수로 변환하려면 📖 '**_TO_***(비트 열→실수 변환 그룹) 명령(2-261페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 변환 결과가 'Out'의 유효 범위를 초과하는 경우, 'Out'의 값은 잘못된 값이 됩니다.

_TO_*(실수→실수 변환 그룹)

실수를 다른 데이터 형식의 실수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
_TO_*	실수→실수 변환 그룹	FUN	(@)**_TO_*** EN ENO In Out 단, **, ***는 다른 실수의 데이터 형식명	Out:=**_TO_*** (In); 단, **, ***는 다른 실수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

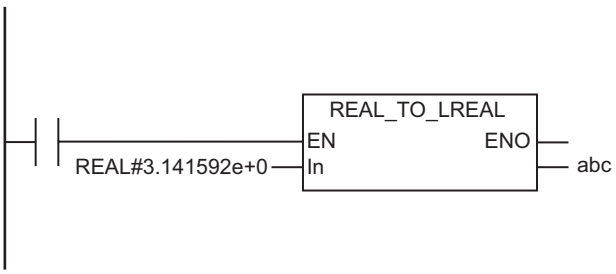
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out														○	○					

기능

실수 'In'을 다른 데이터 형식의 실수로 변환합니다.
명령 이름은 'In'과 변환 결과 'Out'의 데이터 형식의 조합에 따라 다릅니다. 예를 들면 'In'이 REAL 형식, 'Out'이 LREAL 형식인 경우, 명령 이름은 REAL_TO_LREAL이 됩니다.

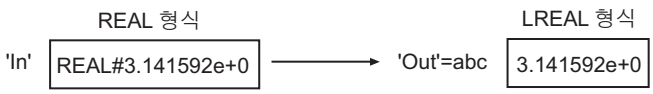
REAL_TO_LREAL 명령에서 'In'=REAL#3.141592e+0인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=REAL_TO_LREAL(REAL#3.141592e+0);
```



'In'과 'Out'의 유효 범위는 다음과 같습니다.

'In'의 데이터 형식	'Out'의 데이터 형식	'In', 'Out'의 유효 범위
REAL	LREAL	-3.402823e+38~3.402823e+38
LREAL	REAL	또는 +∞/-∞

참고

임의의 데이터 형식에서 실수로 변환하려면 'TO_**(실수 변환 그룹) 명령(2-297페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'과 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 값이 +∞/-∞인 경우, 'Out'의 값은 +∞/-∞가 됩니다.
- 'In'의 값이 비수인 경우, 'Out'의 값은 비수가 됩니다.
- 변환 결과가 'Out'의 유효 범위 밖인 경우, 'Out'의 값은 'In'의 값과 같은 부호의 무한대가 됩니다.
- LREAL_TO_REAL 명령의 경우, 'In'의 값이 ±1.175494e-38보다 0에 가까우면 'Out'의 값은 0이 됩니다.

**_TO_STRING(정수→문자열 변환 그룹)

정수를 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
**_TO_STRING	정수→문자열 변환 그룹	FUN	(@)**_TO_STRING EN ENO In Out 단, **는 정수의 데이터 형식명	Out:=**_TO_STRING(In); 단, **는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In'의 데이터 형식에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Out																				○

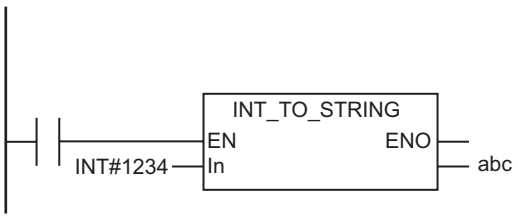
기능

정수 'In'을 문자열로 변환합니다. 'In'을 숫자로 표현한 것이 문자열로서, 변환 결과 'Out'이 됩니다. 'Out'의 종단에는 NULL 문자(16#00)을 붙입니다.

변환은 좌측부터 채웁니다. 'In'의 값이 'In'의 데이터 형식의 최대 자릿수에 채워지지 않는 경우, 상위 자리의 0은 문자열로 출력하지 않습니다. 즉, 제로 억제를 합니다. 'In'이 음수인 경우에는 문자열의 앞에 음수 부호 '-'를 붙입니다.
명령 이름은 'In'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In'이 INT 형식인 경우, 명령 이름은 INT_TO_STRING이 됩니다.

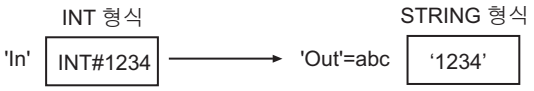
INT_TO_STRING 명령에서 'In'=INT#1234인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=INT_TO_STRING(INT#1234);
```



'Out'의 유효 범위는 'In'의 데이터 형식에 따라 다음과 같이 달라집니다.

'In'의 데이터 형식	'Out'의 유효 범위(최대 바이트 수)
USINT	4바이트(반각 영숫자 3문자+종단 NULL 문자)
UINT	6바이트(반각 영숫자 5문자+종단 NULL 문자)
UDINT	11바이트(반각 영숫자 10문자+종단 NULL 문자)
ULINT	21바이트(반각 영숫자 20문자+종단 NULL 문자)
SINT	5바이트(반각 영숫자 4문자+종단 NULL 문자)
INT	7바이트(반각 영숫자 6문자+종단 NULL 문자)
DINT	12바이트(반각 영숫자 11문자+종단 NULL 문자)
LINT	21바이트(반각 영숫자 20문자+종단 NULL 문자)

참고

문자열에서 정수로 변환하려면 'STRING_TO_**(문자열→정수 변환 그룹) 명령(2-286페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'의 데이터 형식에 맞추십시오.

**_TO_STRING(비트 열→문자열 변환 그룹)

비트 열을 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
**_TO_STRING	비트 열→문자열 변환 그룹	FUN	(@)**_TO_STRING EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out:=**_TO_STRING(In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0
Out	변환 결과	출력	변환 결과	(*)	–	–

* 'In'의 데이터 형식에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

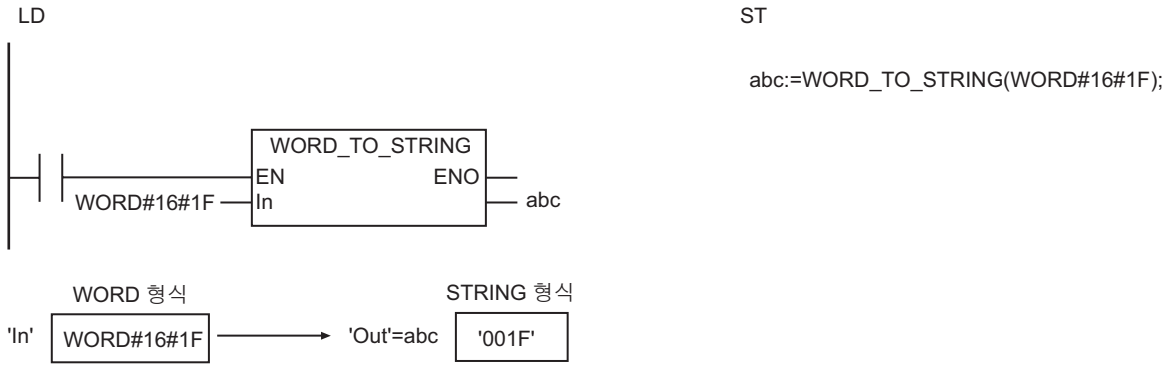
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out																				○

기능

비트 열 'In'을 문자열로 변환합니다. 'In'을 16진수 영숫자로 표현한 것이 문자열로서, 변환 결과 'Out'이 됩니다. 단, 16진수라는 것을 표시하는 접두어 '#16'은 'Out'에 포함되지 않습니다. 또한 'Out'의 종단에는 NULL 문자(16#00)을 붙입니다.

변환은 왼쪽부터 이루어집니다. 'In'의 값이 'In'의 데이터 형식의 최대 자릿수에 채워지지 않는 경우, 상위 자리를 '0'으로 채웁니다. 즉, 제로 패딩을 합니다. 따라서, 'Out'의 바이트 수는 종단의 NULL 문자를 포함해서 항상('In'의 바이트 수×2+1)과 일치합니다.
명령 이름은 'In'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In'이 WORD 형식인 경우, 명령 이름은 WORD_TO_STRING이 됩니다.

WORD_TO_STRING 명령에서 'In'=WORD#16#1F인 경우의 예가 아래에 나와 있습니다.



'Out'의 유효 범위는 'In'의 데이터 형식에 따라 다음과 같이 달라집니다.

'In'의 데이터 형식	'Out'의 유효 범위(최대 바이트 수)
BYTE	3바이트(반각 영숫자 2문자+종단 NULL 문자)
WORD	5바이트(반각 영숫자 4문자+종단 NULL 문자)
DWORD	9바이트(반각 영숫자 8문자+종단 NULL 문자)
LWORD	17바이트(반각 영숫자 16문자+종단 NULL 문자)

참고

'In'을 부호가 있는 문자열로 변환하려는 경우, 우선 '**_TO_***(비트 열→정수 변환 그룹) 명령(2-257 페이지)'으로 부호가 있는 정수로 변환한 후, '**_TO_STRING(정수→문자열 변환 그룹) 명령(2-270 페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'의 데이터 형식에 맞추십시오.

**_TO_STRING(실수→문자열 변환 그룹)

실수를 문자열로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
**_TO_ STRING	실수→문자열 변환 그룹	FUN	(@)**_TO_STRING EN ENO In Out 단, **는 실수의 데이터 형식명	Out:=**_TO_STRING(In); 단, **는 실수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0.0
Out	변환 결과	출력	변환 결과	(*)	–	–

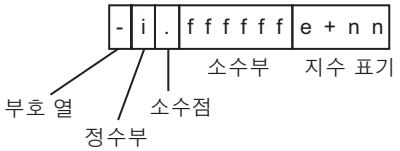
* 'In'의 데이터 형식에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out																				○

기능

실수 'In'을 문자열로 변환합니다. 'In'을 영숫자 문자열로 표현한 것이 변환 결과 'Out'이 됩니다.

'Out'의 구성은 다음과 같습니다.

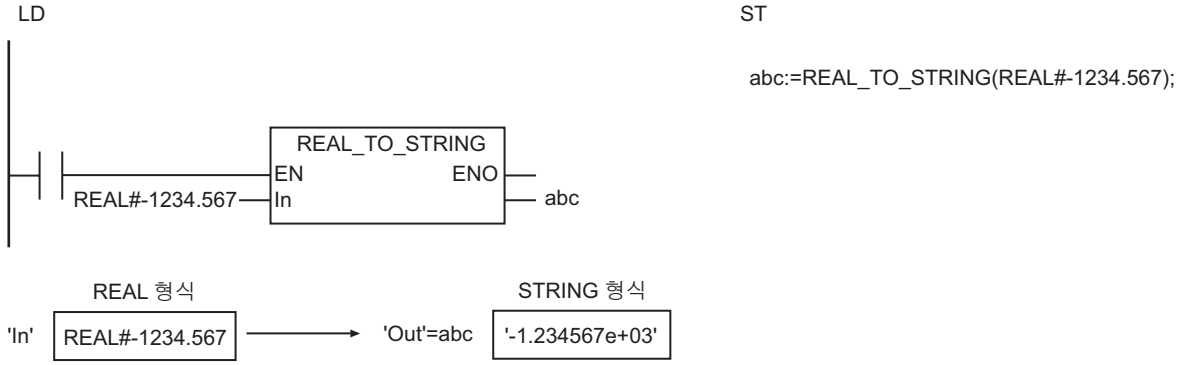


항목	내용
부호 열	'In'이 음수인 경우에만 '-'(마이너스)가 부가됩니다. 'In'이 양수인 경우, '+'(플러스)는 부가되지 않습니다.
정수부	항상 1자리로 표현됩니다.
소수점	'In'이 소수가 아니라도 항상 표기됩니다.
소수부	'In'이 REAL 형식인 경우 6자리, LREAL 형식인 경우 14자리로 표기됩니다.
지수 표기	항상 표기됩니다. nn의 자릿수는 2 또는 3입니다. nn의 부호는 'In'의 절대값이 1.0 이상인 경우 '+'(플러스), 1.0 미만인 경우 '-'(마이너스)입니다.

'Out'의 종단에는 NULL 문자(16#00)을 붙입니다.

명령 이름은 'In'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In'이 REAL 형식인 경우, 명령 이름은 REAL_TO_STRING이 됩니다.

REAL_TO_STRING 명령에서 'In'=REAL#-1234.567인 경우의 예가 아래에 나와 있습니다.



'In'의 값이 0, 무한대, 비수인 경우 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
0	'0'
+∞	'inf'
-∞	'-inf'
비수	'nan' 또는 '-nan'

참고

- 문자열에서 실수로 변환하려면 'STRING_TO_**(문자열→실수 변환 그룹) 명령(2-290페이지)'을 사용하십시오.
- 서석을 지정해서 실수에서 문자열로 변환하는 경우에는 'RealToFormatString 명령(2-276페이지)', 'LrealToFormatString 명령(2-281페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'In'의 데이터 형식에 맞추십시오.

RealToFormatString

REAL 형식 변수를 서식 지정이 있는 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
RealTo FormatString	실수(REAL)→서식 있는 문자열 변환	FUN	(@)RealToFormatString EN In Exponent Sign MinLen DecPlace ENO Out	Out:=RealToFormatString(In, Exponent, Sign, MinLen, DecPlace);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0.0
Exponent	지수 표기		TRUE : 지수 표기 있음 FALSE : 지수 표기 없음			FALSE
Sign	부호 열		TRUE : 부호 열 있음 FALSE : 부호 열 없음			
MinLen	최소 자릿수		'Out'의 최소 자릿수			6
DecPlace	정밀도		'Out'의 소수점 이하의 자릿수	0~15		
Out	변환 결과	출력	변환 결과	최대 327바이트 (반각 영숫자 326문자+종단 NULL 문자)	–	–

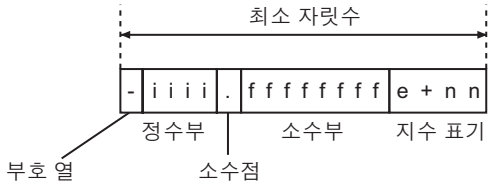
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○						
Exponent	○																			
Sign	○																			
MinLen						○														
DecPlace						○														
Out																				○

기능

REAL 형식 변수 'In'을 문자열로 변환합니다. 'In'을 영숫자 문자열로 표현한 것이 변환 결과 'Out'이 됩니다. 'Out'의 종단에는 NULL 문자(16#00)을 붙입니다.

'In'의 값이 음수인 경우에는 앞에 '-'(마이너스)를 넣지만, 'In'의 값이 양수인 경우에는 앞에 '+'(플러스)를 넣을 수 없습니다.

'Out'의 형식은 지수 표기 'Exponent', 부호 열 'Sign', 최소 자릿수 'MinLen', 정밀도 'DecPlace'에 따라 변합니다.



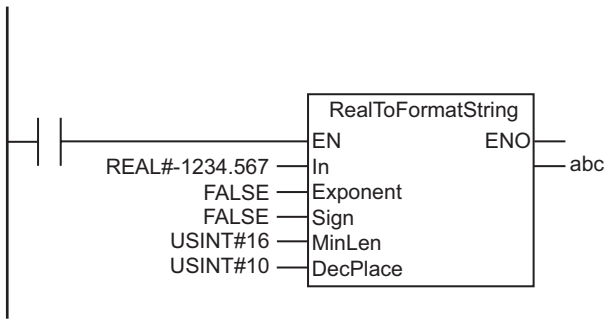
입력 변수	의미
'Exponent'	지수 표기 여부를 지시 TRUE : 지수 표기 있음 FALSE : 지수 표기 없음
'Sign'	부호 열 유무의 여부를 지시 TRUE : 부호 열 있음 FALSE : 부호 열 없음 부호 열에 들어가는 것은 '-'(마이너스)뿐입니다. 양수이며 부호 열 있음인 경우, 부호 열에는 ' '(공백 문자)가 들어갑니다. 음수이며 부호 열 없음인 경우, 정수부의 앞에 '-'(마이너스)가 붙습니다. 단, 변환 결과의 자릿수가 'MinLen'의 값을 초과하는 양수인 경우, 부호 열에 최상위 자리의 숫자가 들어갑니다.
'MinLen'	부호 열, 정수부, 소수점, 소수부, 지수 표기를 합한 전체 자릿수의 최소값 변환 결과의 자릿수가 'MinLen'의 값 미만인 경우, 부호 열을 제외한 문자열을 우측에서부터 배치해서, 빈 자리에는 ' '(공백 문자)를 채워 넣습니다. 변환 결과의 자릿수가 'MinLen'의 값을 초과하는 경우, 문자열을 좌측에서부터 써서 'MinLen'의 값을 초과하는 자릿수의 문자열을 'Out'에 대입합니다.
'DecPlace'	소수부의 자릿수 'DecPlace'의 값을 초과하는 자릿수는 오사오입(뒤에 설명)으로 반올림합니다. 또한, 'DecPlace'의 값이 0인 경우에는 소수부와 함께 소수점 표기도 없어집니다.

'In'의 값이 REAL#-1234.567인 경우의 각 입력 변수의 값과 'Out'의 값의 관계 예를 게재합니다.

(예1) 'Exponent' : FALSE
'Sign' : FALSE
'MinLen' : USINT#16
'DecPlace' : USINT#10

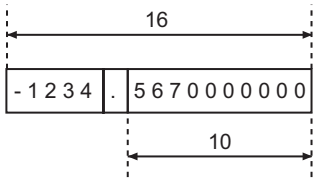
부호 열이 없는 음수이므로, 정수부 앞이 '-'(마이너스)가 됩니다.

LD



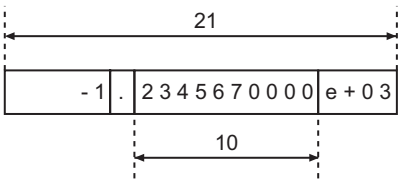
ST

```
abc:=RealToFormatString(REAL#-1234.567, FALSE,
                        FALSE, USINT#16,
                        USINT#10);
```



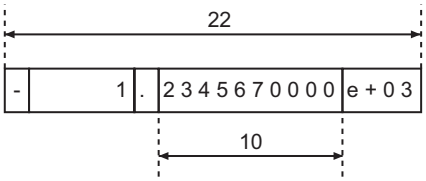
(예2) 'Exponent' : TRUE
'Sign' : FALSE
"MinLen' : USINT#21
"DecPlace' : USINT#10

문자열의 자릿수보다 'MinLen'이 크므로, 문자열을 우측에서부터 써서 앞에 ' '(공백 문자)를 채워 넣습니다.



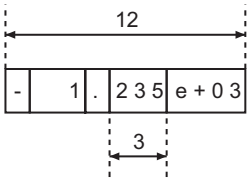
(예3) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#22
'DecPlace' : USINT#10

부호 열은 반드시 좌측 끝에 붙입니다. 정수부 앞에 ' '(공백 문자)를 채워 넣습니다.



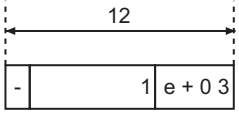
(예4) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#12
'DecPlace' : USINT#3

'DecPlace'=USINT#3이므로 소수점 이하 4번째 자리가 오사오입됩니다.



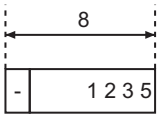
(예5) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#12
'DecPlace' : USINT#0

'DecPlace'=USINT#0이므로 소수점 이하 1번째 자리가 오사오입됩니다. 소수점 표기도 없어집니다.



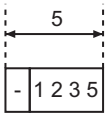
(예6) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#8
'DecPlace' : USINT#0

지수 표기가 없어지고 정수부가 4자리가 됩니다. 또한 소수점 이하 1번째 자리가 오사오입됩니다.



(예7) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#2
'DecPlace' : USINT#0

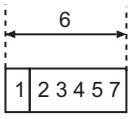
'MinLen'=USINT#2보다 'In'의 정수부 자릿수 4가 더 크므로 정수부는 4자리로 표기됩니다.



'In'의 값이 REAL#123456.7인 경우의 입력 변수의 값과 'Out'의 값의 관계 예를 게재합니다.

(예8) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#4
'DecPlace' : USINT#0

'MinLen'=USINT#4보다 'In'의 정수부 자릿수 6이 더 크므로 정수부는 6자리로 표기됩니다. 또한, 'In'의 값이 양수이므로 부호 열에 최상위 자리의 숫자가 들어갑니다.



'In'의 값이 무한대, 비수인 경우 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
+∞	'inf'
-∞	'-inf'
비수	'nan' 또는 '-nan'

오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2
0.5보다 큼	올림	1.51→2

참고

- 'Exponent', 'Sign', 'MinLen', 'DecPlace'는 각각 생략할 수 있습니다. 생략한 경우 각각의 초기값이 적용됩니다.
- LREAL 형식 변수에서 문자열로 변환하려면 🕮 'LrealToFormatString 명령(2-281페이지)'을 사용하십시오.
- 문자열에서 실수로 변환하려면 🕮 'STRING_TO_**(문자열→실수 변환 그룹) 명령(2-290페이지)'을 사용하십시오.

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'DecPlace'의 값이 유효 범위 밖인 경우
- 'DecPlace'의 값이 'MinLen'의 값보다 큰 경우

LrealToFormatString

LREAL 형식 변수를 서식 지정이 있는 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
LrealTo FormatString	실수(LREAL)→서식 있는 문자열 변환	FUN	(@)LrealToFormatString EN In Exponent Sign MinLen DecPlace ENO Out	Out:=LrealToFormatString(In, Exponent, Sign, MinLen, DecPlace);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0.0
Exponent	지수 표기		TRUE : 지수 표기 있음 FALSE : 지수 표기 없음			FALSE
Sign	부호 열		TRUE : 부호 열 있음 FALSE : 부호 열 없음			
MinLen	최소 자릿수		'Out'의 최소 자릿수			6
DecPlace	정밀도		'Out'의 소수점 이하의 자릿수	0~15		
Out	변환 결과	출력	변환 결과	최대 327바이트 (반각 영숫자 326문자+종단 NULL 문자)	–	–

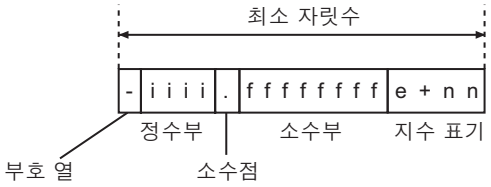
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In															○					
Exponent	○																			
Sign	○																			
MinLen						○														
DecPlace						○														
Out																				○

기능

LREAL 형식 변수 'In'을 문자열로 변환합니다. 'In'을 영숫자 문자열로 표현한 것이 변환 결과 'Out'이 됩니다. 'Out'의 종단에는 NULL 문자(16#00)을 붙입니다.

'In'의 값이 음수인 경우에는 앞에 '-'(마이너스)를 넣지만, 'In'의 값이 양수인 경우에는 앞에 '+'(플러스)를 넣을 수 없습니다.

'Out'의 형식은 지수 표기 'Exponent', 부호 열 'Sign', 최소 자릿수 'MinLen', 정밀도 'DecPlace'에 따라 변합니다.



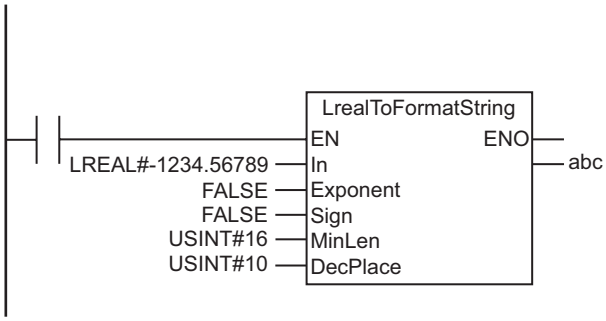
입력 변수	의미
'Exponent'	지수 표기 여부를 지시 TRUE : 지수 표기 있음 FALSE : 지수 표기 없음
'Sign'	부호 열 유무의 여부를 지시 TRUE : 부호 열 있음 FALSE : 부호 열 없음 부호 열에 들어가는 것은 '-'(마이너스)뿐입니다. 양수이며 부호 열 있음인 경우, 부호 열에는 ''(공백 문자)가 들어갑니다. 음수이며 부호 열 없음인 경우, 정수부의 앞에 '-'(마이너스)가 붙습니다. 단, 변환 결과의 자릿수가 'MinLen'의 값을 초과하는 양수인 경우, 부호 열에 최상위 자리의 숫자가 들어갑니다.
'MinLen'	부호 열, 정수부, 소수점, 소수부, 지수 표기를 합한 전체 자릿수의 최소값 변환 결과의 자릿수가 'MinLen'의 값 미만인 경우, 부호 열을 제외한 문자열을 우측에서부터 배치해서, 빈 자리에는 ' '(공백 문자)를 채워 넣습니다. 변환 결과의 자릿수가 'MinLen'의 값을 초과하는 경우, 문자열을 좌측에서부터 써서 'MinLen'의 값을 초과하는 자릿수의 문자열을 'Out'에 대입합니다.
'DecPlace'	소수부의 자릿수 'DecPlace'의 값을 초과하는 자릿수는 오사오입(뒤에 설명)으로 반올림합니다. 또한, 'DecPlace'의 값이 0인 경우에는 소수부와 함께 소수점 표기도 없어집니다.

'In'의 값이 LREAL#-1234.56789인 경우의 각 입력 변수의 값과 'Out'의 값의 관계 예를 게재합니다.

(예1) 'Exponent' : FALSE
'Sign' : FALSE
'MinLen' : USINT#16
'DecPlace' : USINT#10

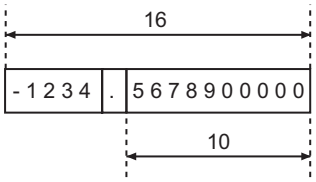
부호 열이 없는 음수이므로, 정수부 앞이 '-'(마이너스)가 됩니다.

LD



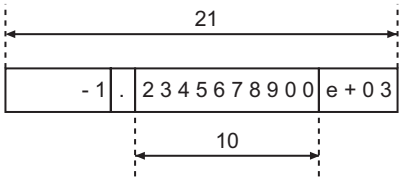
ST

```
abc:=LrealToFormatString(LREAL#-1234.56789, FALSE,
                         FALSE, USINT#16,
                         USINT#10);
```



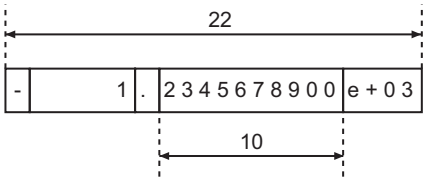
(예2) 'Exponent' : TRUE
'Sign' : FALSE
'MinLen' : USINT#21
'DecPlace' : USINT#10

문자열의 자릿수보다 'MinLen'이 크므로, 문자열을 우측에서부터 써서 앞에 ' '(공백 문자)를 채워 넣습니다.



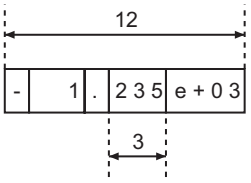
(예3) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#22
'DecPlace' : USINT#10

부호 열은 반드시 좌측 끝에 붙입니다. 정수부 앞에 ' '(공백 문자)를 채워 넣습니다.



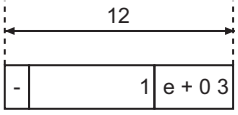
(예4) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#12
'DecPlace' : USINT#3

'DecPlace'=USINT#3이므로, 소수점 이하 4번째 자리가 오사오입됩니다.



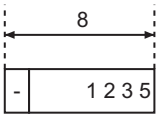
(예5) 'Exponent' : TRUE
'Sign' : TRUE
'MinLen' : USINT#12
'DecPlace' : USINT#0

'DecPlace'=USINT#0이므로 소수점 이하 1번째 자리가 오사오입됩니다. 소수점 표기도 없어집니다.



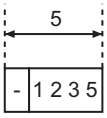
(예6) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#8
'DecPlace' : USINT#0

지수 표기가 없어지고 정수부가 4자리가 됩니다. 또한, 소수점 이하 1번째 자리가 오사오입됩니다.



(예7) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#2
'DecPlace' : USINT#0

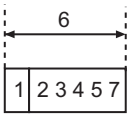
'MinLen'=USINT#2보다 'In'의 정수부 자릿수 4가 더 크므로 정수부는 4자리로 표기됩니다.



'In'의 값이 LREAL#123456.789인 경우의 입력 변수의 값과 'Out'의 값의 관계 예를 게재합니다.

(예8) 'Exponent' : FALSE
'Sign' : TRUE
'MinLen' : USINT#4
'DecPlace' : USINT#0

'MinLen'=USINT#4보다 'In'의 정수부 자릿수 6이 더 크므로 정수부는 6자리로 표기됩니다. 또한, 'In'의 값이 양수이므로 부호 열에 최상위 자리의 숫자가 들어갑니다.



'In'의 값이 무한대, 비수인 경우 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
+∞	'inf'
-∞	'-inf'
비수	'nan' 또는 '-nan'

오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2
0.5보다 큼	올림	1.51→2

참고

- 'Exponent', 'Sign', 'MinLen', 'DecPlace'는 각각 생략할 수 있습니다. 생략한 경우 각각의 초기값이 적용됩니다.
- REAL 형식 변수에서 문자열로 변환하려면 'RealToFormatString 명령(2-276페이지)'을 사용하십시오.
- 문자열에서 실수로 변환하려면 'STRING_TO_**(문자열→실수 변환 그룹) 명령(2-290페이지)'을 사용하십시오.

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'DecPlace'의 값이 유효 범위 밖인 경우
- 'DecPlace'의 값이 'MinLen'의 값보다 큰 경우

STRING_TO_**(문자열→정수 변환 그룹)

문자열을 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
STRING_TO_**	문자열→정수 변환 그룹	FUN	(@)STRING_TO_** EN ENO In Out 단, **는 정수의 데이터 형식명	Out:=STRING_TO_** (In); 단, **는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	''
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 'Out'의 데이터 형식에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out						○	○	○	○	○	○	○	○							

기능

문자열 'In'을 정수 형식으로 변환합니다.

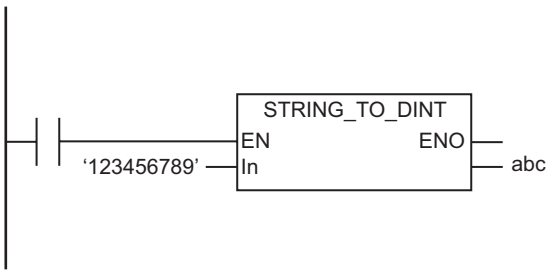
'In'의 문자열은 '0'~'9'의 숫자만으로 구성되어야 합니다. 단, 다음과 같은 예외가 있습니다.

- 앞에 '-'(마이너스) 또는 '+'(플러스)가 하나만 있는 경우, 그것을 부호로 처리합니다.
- 앞에 몇 개의 ' '(공백 문자)가 있는 경우, 그것을 무시합니다.
- 앞의 '-'(마이너스) 또는 '+'(플러스)와 숫자 사이에 몇 개의 ' '(공백 문자)가 있는 경우, 그 ' '(공백 문자)를 무시합니다.
- 임의의 위치에 단독의 '_'(밑줄)을 포함하는 경우, 그것을 무시합니다.
- 임의의 위치에 '_'(밑줄)이 2개 이상 연속으로 있는 경우, 이상 상태가 됩니다.
- 앞과 가장 뒤에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.
- 앞의 '-'(마이너스) 또는 '+'(플러스)와 숫자 사이에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.

명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 DINT 형식인 경우, 명령 이름은 STRING_TO_DINT가 됩니다.

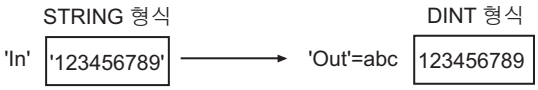
STRING_TO_DINT 명령에서 'In'='123456789'인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=STRING_TO_DINT('123456789');
```



'In'의 유효 범위는 'Out'의 데이터 형식에 따라 다음과 같이 달라집니다.

'Out'의 데이터 형식	'In'의 유효 범위(최대 바이트 수)(*)
USINT	4바이트(반각 영숫자 3문자+종단 NULL 문자)
UINT	6바이트(반각 영숫자 5문자+종단 NULL 문자)
UDINT	11바이트(반각 영숫자 10문자+종단 NULL 문자)
ULINT	21바이트(반각 영숫자 20문자+종단 NULL 문자)
SINT	5바이트(반각 영숫자 4문자+종단 NULL 문자)
INT	7바이트(반각 영숫자 6문자+종단 NULL 문자)
DINT	12바이트(반각 영숫자 11문자+종단 NULL 문자)
LINT	21바이트(반각 영숫자 20문자+종단 NULL 문자)

* 문자열 앞의 복수의 ' '(공백 문자), 문자열 앞의 복수의 '0', 문자열에 포함된 단독의 '_'(밑줄)은 바이트 수에 포함되지 않습니다.

참고

- 문자열을 16진수로 변환하려는 경우, 🕮 'STRING_TO_**(문자열→비트 열 변환 그룹) 명령(2-288페이지)'을 사용하십시오.
- 정수 형식에서 문자열 형식으로 변환하려면 🕮 '**_TO_STRING(정수→문자열 변환 그룹) 명령(2-270페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 값이 '-0'인 경우, 'Out'의 값은 0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 수치를 나타내는 문자열이 아닌 경우
  - 변환 결과가 'Out'인 데이터 형식의 유효 범위 밖인 경우

STRING_TO_**(문자열→비트 열 변환 그룹)

문자열을 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
STRING_TO_**	문자열→비트 열 변환 그룹	FUN	(@)STRING_TO_** EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out:=STRING_TO_** (In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*)	–	''
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 'Out'의 데이터 형식에 따라 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out		○	○	○	○															

기능

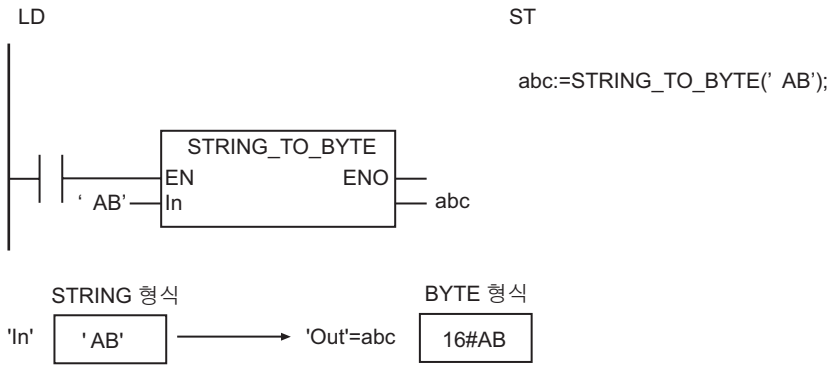
문자열 'In'을 16진수로 간주해서 비트 열로 변환합니다.

'In'의 문자열은 '0'~'9'의 숫자 또는 'a'~'f' 및 'A'~'F'만으로 구성되어야 합니다. 단, 다음과 같은 예외가 있습니다.

- 앞에 ' '(공백 문자) 및 '0'이 연속으로 있는 경우, 그것을 무시합니다.
- 임의의 위치에 단독의 '_'(밑줄)을 포함하는 경우, 그것을 무시합니다.
- 임의의 위치에 '_'(밑줄)이 2개 이상 연속으로 있는 경우, 이상 상태가 됩니다.
- 앞과 가장 뒤에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.
- 앞의 '-'(마이너스) 또는 '+'(플러스)와 숫자 사이에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.

명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 BYTE 형식인 경우, 명령 이름은 STRING_TO_BYTE가 됩니다.

STRING_TO_BYTE 명령에서 'In'='AB'인 경우의 예가 아래에 나와 있습니다. 앞의 ' '(공백 문자)는 무시됩니다.



'In'의 유효 범위는 'Out'의 데이터 형식에 따라 다음과 같이 달라집니다.

'Out'의 데이터 형식	'In'의 유효 범위(최대 바이트 수) (*)
BYTE	3바이트(반각 영숫자 2문자+종단 NULL 문자)
WORD	5바이트(반각 영숫자 4문자+종단 NULL 문자)
DWORD	9바이트(반각 영숫자 8문자+종단 NULL 문자)
LWORD	17바이트(반각 영숫자 16문자+종단 NULL 문자)

* 문자열 앞의 복수의 ' '(공백 문자), 문자열 앞의 복수의 '0', 문자열에 포함된 단독의 '_'(밑줄)은 바이트 수에 포함되지 않습니다.

참고

- 부호가 있는 수치를 표현하는 문자열을 다루는 경우, 🕮 'STRING_TO_**(문자열→정수 변환 그룹) 명령(2-286페이지)'을 사용합니다.
- 비트 열에서 문자열로 변환하려면 🕮 '**_TO_STRING(비트 열→문자열 변환 그룹) 명령(2-272페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 16진수의 수치를 나타내는 문자열이 아닌 경우
  - 변환 결과가 'Out'인 데이터 형식의 유효 범위 밖인 경우

STRING_TO_**(문자열→실수 변환 그룹)

문자열을 실수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
STRING_TO_**	문자열→실수 변환 그룹	FUN	(@)STRING_TO_** EN ENO In Out 단, **는 실수의 데이터 형식명	Out:=STRING_TO_** (In); 단, **는 실수의 데이터 형식명

변수

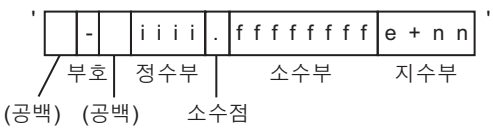
	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	최대 311바이트 (반각 영숫자 310문자+종단 NULL 문자)	–	''
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out														○	○					

기능

문자열 'In'을 실수로 변환합니다.
명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 LREAL 형식인 경우, 명령 이름은 STRING_TO_LREAL이 됩니다.

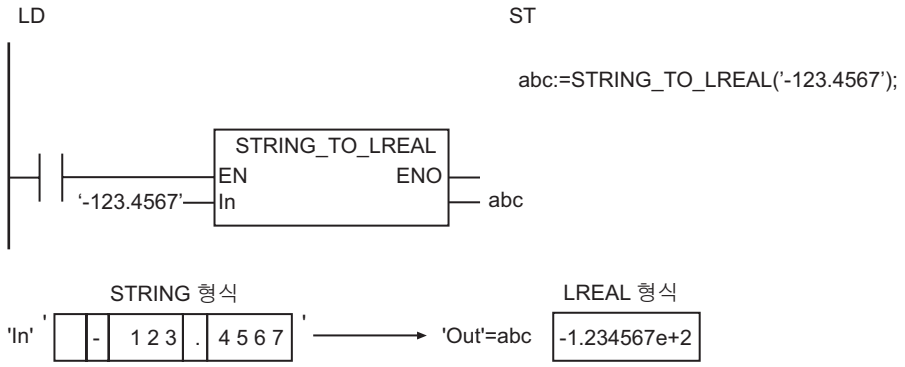
'In'의 문자열 형식은 다음과 같습니다.



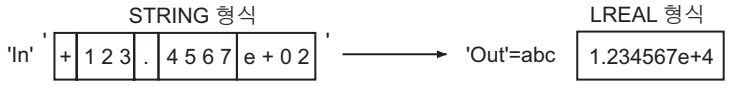
명칭	서식
부호	• 문자열 앞에 나타나는 연속하는 ' '(공백 문자)를 무시한 후에 단독으로 나타나는 '+' 또는 '-'를 부호로 합니다. • '+' 부호는 생략할 수 있습니다. • 부호 뒤의 연속하는 공백 문자도 무시됩니다.

명칭	서식
정수부	• 부호 뒤에서 소수점 앞까지를 정수부라고 합니다. 단, 부호 뒤의 연속하는 공백 문자는 정수부에 포함되지 않습니다. 또한, 부호는 생략되는 경우가 있습니다. • 소수점 및 소수부가 없는 경우, 지수부 앞까지를 정수부로 합니다. • 소수점, 소수부, 지수부가 없는 경우에는 문자열 종단까지를 정수부로 합니다. • 정수부는 '0'~'9'의 숫자로 구성됩니다. • 정수부는 생략할 수 없습니다. • 정수부의 최대 자릿수는 최대 문자열 길이 1985에서 부호, 소수점, 소수부, 지수부 및 부호 앞뒤의 공백 문자의 바이트 수의 합계입니다.
소수점	• 정수부에 이어 단독으로 나오는 '.'(마침표)를 소수점으로 합니다. • 소수부가 존재하지 않는 경우, 소수점은 생략하십시오.
소수부	• 소수점 뒤에서 지수부 앞까지를 소수부라고 합니다. • 지수부가 없는 경우에는 문자열 종단까지를 소수부라고 합니다. • 소수부는 '0'~'9'의 숫자로 구성됩니다. • 소수부는 생략할 수 있습니다. • 소수부의 최대 자릿수는 15입니다. • 소수점이 존재하지 않는 경우, 소수부도 존재하지 않습니다.
지수부	• 소수부에 이어 단독으로 나오는 'e' 또는 'E', 그 뒤에 단독으로 나오는 '+' 또는 '-'에서 문자열 종단까지를 지수부로 합니다. • 소수부가 없는 경우에는 소수점 뒤에 나오는 상기 문자열을 지수부로 합니다. • 소수점 및 소수부가 없는 경우에는 정수부 뒤에 나오는 상기 문자열을 지수부로 합니다. • 지수부의 숫자 부분은 '0'~'9'의 숫자로 구성됩니다. • 지수부는 생략할 수 있습니다. • 지수부의 숫자 부분의 최대 자릿수는 3입니다.

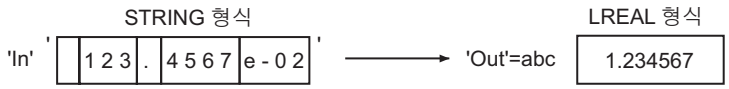
(예1) 부호 있음, 소수점 있음, 소수부 있음, 지수부 없음



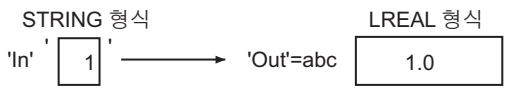
(예2) 부호 있음, 소수점 있음, 소수부 있음, 지수부 있음



(예3) 부호 없음, 소수점 있음, 소수부 있음, 지수부 있음



(예4) 부호 없음, 소수점 없음, 소수부 없음, 지수부 없음



'In'의 값이 +inf인 경우, 'Out'의 값은 +∞가 됩니다. 'In'의 값이 '-Inf'인 경우, 'Out'의 값은 -∞가 됩니다. 어느 경우에도 대소문자의 구별은 없습니다.

참고

실수에서 문자열로 변환하려면 '**_TO_STRING(실수→문자열 변환 그룹) 명령(2-274페이지)'을 사용하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 임의의 위치에 단독의 '_'(밑줄)을 포함하는 경우, 그것을 무시합니다.
- 'In'의 앞과 가장 뒤에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.
- 'In'의 임의의 위치에 '_'(밑줄)이 2개 이상 연속으로 있는 경우, 이상 상태가 됩니다.
- 'In' 앞의 '-'(마이너스) 또는 '+'(플러스)와 숫자 사이에 '_'(밑줄)이 있는 경우, 이상 상태가 됩니다.
- 'In'의 내용이 'Out'의 데이터 형식으로 표현할 수 있는 정밀도를 초과하는 경우, 반올림이 수행됩니다.
- 'In'의 내용이 'Out'의 데이터 형식으로 표현할 수 있는 값보다 0에 가까운 경우, 'Out'의 값은 0이 됩니다.
- 'In'의 내용이 'Out'의 유효 범위를 초과하는 경우, 양수라면 'Out'의 값은 +∞, 음수라면 -∞가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 수치를 나타내는 문자열이 아닌 경우
  - 'In'의 문자열이 소수점은 있고 소수부는 없는 경우

TO_**(정수 변환 그룹)

정수, 비트 열, 실수, 문자열을 정수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TO_**	정수 변환 그룹	FUN	(@)TO_** EN ENO In Out 단, **는 정수의 데이터 형식명	Out:=TO_** (In); 단, **는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*1)	–	(*2)
Out	변환 결과	출력	변환 결과	(*1)	–	–

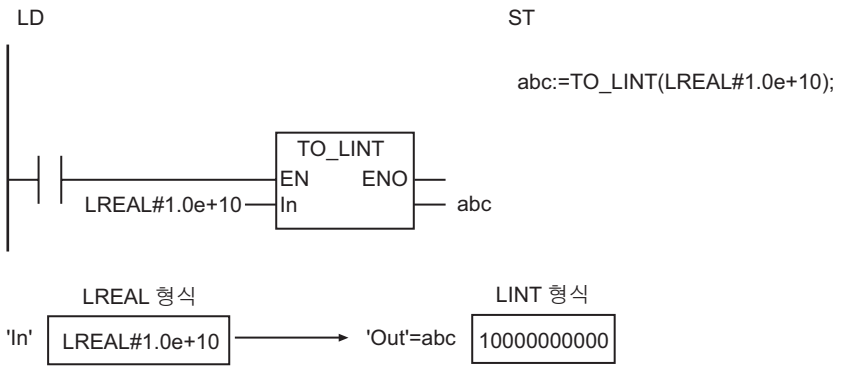
*1 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○	○	○	○	○	○	○	○	○	○	○					○
Out						○	○	○	○	○	○	○	○							

기능

정수, 비트 열, 실수, 문자열 중 하나인 'In'을 정수로 변환합니다.
명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 LINT 형식인 경우, 명령 이름은 TO_LINT가 됩니다.

TO_LINT 명령에서 'In'=LREAL#1.0e+10인 경우의 예가 아래에 나와 있습니다.



- 변환은 'In'의 데이터 형식의 유효 자리의 범위에서 실행합니다. 'In'이 부동 소수점 형식인 경우, 소수점 이하는 오사오입해서 정수로 반올림합니다. 오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2
0.5보다 큼	올림	1.51→2

'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 변합니다. 각각의 유효 범위는 '**_TO_***(정수→정수 변환 그룹) 명령(2-250페이지)', '**_TO_***(비트 열→정수 변환 그룹) 명령(2-257페이지)', '**_TO_***(실수→정수 변환 그룹) 명령(2-263페이지)'의 기능 설명을 참조하십시오.

'In'이 STRING 형식인 경우의 자세한 사양은 'STRING_TO_**(문자열→정수 변환 그룹) 명령(2-286페이지)'의 기능 설명을 참조하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 'In'의 데이터 형식이 비트 열이고 'In'과 'Out'의 데이터 형식의 크기가 다른 경우, 다음과 같은 처리를 수행합니다.
  - 'In'의 데이터 크기보다 'Out'의 데이터 크기가 큰 경우, 'Out'의 상위 자리에 0이 채워집니다.
  - 'In'의 데이터 크기보다 'Out'의 데이터 크기가 작은 경우, 상위 자리를 버립니다.
- 'In'이 STRING 형식인 경우, 다음과 같은 점에 주의하십시오.
  - 'In'의 앞에 '-'(마이너스) 또는 '+'(플러스)가 하나만 존재하는 경우, 부호로 처리됩니다.
  - 'In' 앞의 '-'(마이너스) 또는 '+'(플러스)를 제외하면, 연속하는 '0'~'9'의 숫자로 구성되어야 합니다. 단, 문자열에 포함된 '_'(밑줄)과 '-' 및 '+' 앞뒤의 ' '(공백 문자)는 있어도 상관없습니다.
- 변환 결과가 'Out'의 유효 범위 밖인 경우, 'Out'의 값은 잘못된 값이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'이 STRING 형식이고 그 값이 수치를 나타내는 문자열이 아닌 경우

TO_**(비트 열 변환 그룹)

정수, 비트 열, 실수, 문자열을 비트 열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TO_**	비트 열 변환 그룹	FUN	(@)TO_** EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out:=TO_** (In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*1)	–	(*2)
Out	변환 결과	출력	변환 결과	(*1)	–	–

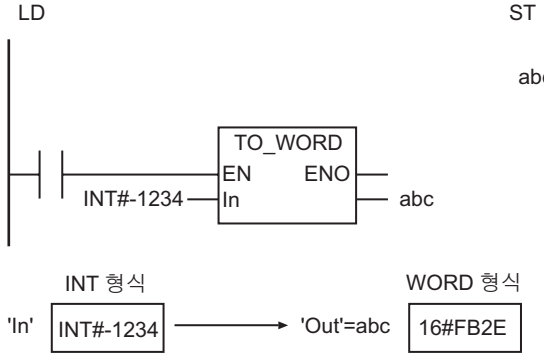
*1 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다.
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○	○	○	○	○	○	○	○	○	○	○					○
Out		○	○	○	○															

기능

정수, 실수, 비트 열, 문자열 중 하나인 'In'을 비트 열로 변환합니다.
명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 WORD 형식인 경우, 명령 이름은 TO_WORD가 됩니다.

TO_WORD 명령에서 'In'=INT#-1234인 경우의 예가 아래에 나와 있습니다.



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 변합니다. 각각의 유효 범위는 '**_TO_***(정수→비트 열 변환 그룹) 명령(2-253페이지)', '**_TO_***(비트 열→비트 열 변환 그룹) 명령(2-259페이지)', '**_TO_***(실수→비트 열 변환 그룹) 명령(2-266페이지)'의 기능 설명을 참조하십시오.

'In'이 STRING 형식인 경우의 자세한 사양은 'STRING_TO_**(문자열→비트 열 변환 그룹) 명령 (2-288페이지)'의 기능 설명을 참조하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 변환 결과가 'Out'의 유효 범위 밖인 경우, 'Out'의 값은 잘못된 값이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'이 STRING 형식이고 그 값이 수치를 나타내는 문자열이 아닌 경우

TO_**(실수 변환 그룹)

정수, 비트 열, 실수, 문자열을 실수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TO_**	실수 변환 그룹	FUN	(@)TO_** EN ENO In Out 단, **는 실수의 데이터 형식명	Out:=TO_** (In); 단, **는 실수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	(*1)(*2)	–	(*3)
Out	변환 결과	출력	변환 결과	(*1)	–	–

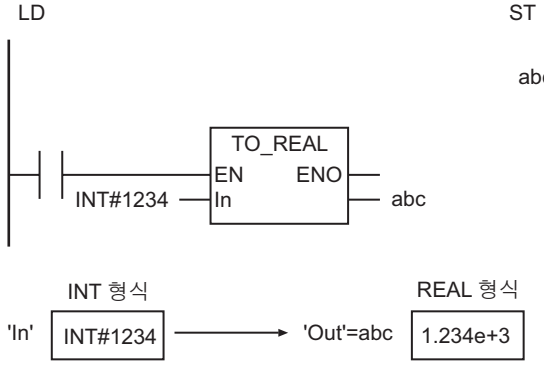
*1 'In', 'Out'의 데이터 형식의 조합으로 유효 범위가 변합니다.
*2 STRING의 경우, 최대 311바이트(반각 영숫자 310문자+종단 NULL 문자)입니다.
*3 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○	○	○	○	○	○	○	○	○	○	○					○
Out														○	○					

기능

정수, 비트 열, 실수, 문자열 중 하나인 변환 대상 'In'을 실수 형식으로 변환합니다.
명령 이름은 변환 결과 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 REAL 형식인 경우, 명령 이름은 TO_REAL이 됩니다.
'In'의 값이 +∞/-∞인 경우, 'Out'의 값은 +∞/-∞가 됩니다.

TO_REAL 명령에서 'In'=INT#1234인 경우의 예가 아래에 나와 있습니다.



'In'과 'Out'의 유효 범위는 데이터 형식의 조합으로 변합니다. 각각의 유효 범위는 '**_TO_***(정수→실수 변환 그룹) 명령(2-255페이지)', '**_TO_***(비트 열→실수 변환 그룹) 명령(2-261페이지)', '**_TO_***(실수→실수 변환 그룹) 명령(2-268페이지)'의 기능 설명을 참조하십시오.

'In'이 STRING 형식인 경우의 자세한 사양은 'STRING_TO_**(문자열→실수 변환 그룹) 명령(2-290페이지)'의 기능 설명을 참조하십시오.

사용상의 주의 사항

- 명령 이름을 반드시 'Out'의 데이터 형식에 맞추십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'이 STRING 형식이고 그 값이 수치를 나타내는 문자열이 아닌 경우

EnumToNum

열거 형식을 DINT 형식으로 변환합니다.

명령	이름	FB/FUN	그래픽 표현	ST 표현
EnumToNum	열거 형식→정수 변환	FUN	(@)EnumToNum EN ENO In Out	Out:=EnumToNum(In);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	–	–	0
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	열거 형식																			
Out												○								

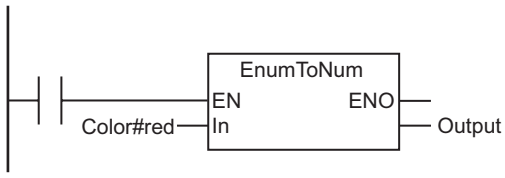
기능

열거 형식의 값인 변환 대상 'In'을 DINT 형식의 값으로 변환하여 변환 결과 'Out'에 출력합니다.

표시기 등과 같이 열거 형식의 변수를 취급하지 않는 기기에서 열거 형식의 변수의 값을 모니터링하려는 경우 등에 이 명령을 사용합니다.

열거 형식 Color의 열거자 red를 DINT 형식의 변수 Output으로 변환하는 경우의 기술 예가 아래에 나와 있습니다. red의 열거값이 0인 경우, Output은 DINT#0이 됩니다.

LD



ST

```
Output:=EnumToNum(Color#red);
```

사용상의 주의 사항

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

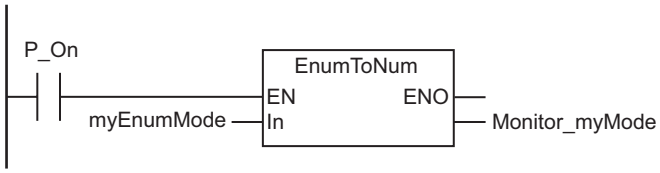
사용자 프로그램의 동작 모드를 열거 형식 EnumMode로 정의한다고 가정합니다. 동작 모드를 표시기에서 모니터링하기 위해 열거 형식 EnumMode의 변수 myEnumMode의 값을 DINT 형식의 변수 Monitor_myMode로 변환합니다. 예를 들어 myEnumMode의 값이 mode2인 경우, Monitor_myMode의 값은 2가 됩니다.

● 데이터 형식의 정의

이름	열거값	코멘트
EnumMode	–	열거 형식
mode0	0	멤버
mode1	1	멤버
mode2	2	멤버

LD

이름	데이터 형식	초기값	코멘트
myEnumMode	EnumMode	mode0	열거 형식 모드의 값
Monitor_myMode	DINT	0	모드의 모니터링 값



ST

이름	데이터 형식	초기값	코멘트
myEnumMode	EnumMode	mode0	열거 형식 모드의 값
Monitor_myMode	DINT	0	모드의 모니터링 값

```
Monitor_myMode:=EnumToNum(myEnumMode);
```

NumToEnum

DINT 형식을 열거 형식으로 변환합니다.

명령	이름	FB/FUN	그래픽 표현	ST 표현
NumToEnum	정수→열거 형식 변환	FUN	(@)NumToEnum EN In InOut Out	NumToEnum(In,InOut);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0
InOut	변환 결과	입출력	변환 결과	–	–	–
Out	반환값	출력	TRUE: 명령을 실행하며 이상 상태가 아님 FALSE: 명령을 실행하지 않거나 이상 상태임	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In												○								
InOut	열거 형식																			
Out	○																			

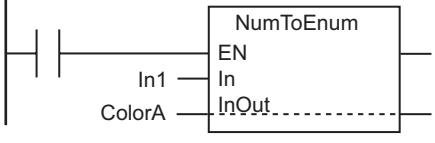
기능

DINT 형식의 값인 변환 대상 'In'을 열거 형식의 값으로 변환하여 변환 결과 'InOut'에 출력합니다.

표시기 등과 같이 열거 형식의 변수를 취급하지 않는 기기에서 열거 형식의 변수의 값을 변경하려는 경우 등에 이 명령을 사용합니다.

DINT 형식의 변수 In1을 열거 형식 Color의 변수 ColorA로 변환하는 경우의 기술 예가 아래에 나와 있습니다. In1의 값이 1, Color의 열거값 1에 대응하는 열거자가 green인 경우, ColorA의 값은 green이 됩니다.

LD



ST

```
NumToEnum(In1,ColorA);
```

참고

이 명령을 래더도 프로그램에서 사용하는 경우, 'In'의 값이 'InOut' 값의 범위 내에 있는지 여부는 'Out'을 통해 확인할 수 있습니다.

사용상의 주의 사항

'In'의 값이 'InOut' 값의 범위 밖인 경우는 이상 상태가 됩니다. 'Out'은 FALSE가 되고 'InOut'의 값은 변화하지 않습니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

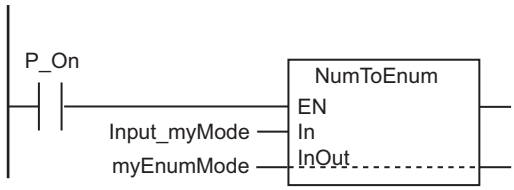
사용자 프로그램의 동작 모드를 열거 형식 EnumMode로 정의한다고 가정합니다. 표시기에서 동작 모드를 변경하기 위해 DINT 형식의 값 Input_myMode를 씁니다. 사용자 프로그램에서 Input_myMode를 열거 형식 EnumMode의 변수 myEnumMode로 변환합니다. 예를 들어 Input_myMode의 값이 1인 경우, myEnumMode의 값은 mode1이 됩니다.

● 데이터 형식의 정의

이름	열거값	코멘트
EnumMode	–	열거 형식
mode0	0	멤버
mode1	1	멤버
mode2	2	멤버

LD

이름	데이터 형식	초기값	코멘트
myEnumMode	EnumMode	mode0	열거 형식 모드의 값
Input_myMode	DINT	0	변경할 모드의 값



ST

이름	데이터 형식	초기값	코멘트
myEnumMode	EnumMode	mode0	열거 형식 모드의 값
Input_myMode	DINT	0	변경할 모드의 값

```
NumToEnum(Input_myMode, myEnumMode);
```

TRUNC / Round / RoundUp

실수를 정수로 합니다.
TRUNC : 소수점 이하 1번째 자리를 버립니다.
Round : 소수점 이하 1번째 자리를 오사오입합니다.
RoundUp : 소수점 이하 1번째 자리를 올립니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TRUNC	실수 버림	FUN	(@)TRUNC EN ENO In Out	Out:=TRUNC(In);
Round	실수 반올림	FUN	(@)Round EN ENO In Out	Out:=Round(In);
RoundUp	실수 올림	FUN	(@)RoundUp EN ENO In Out	Out:=RoundUp(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	(*)
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out												○	○							

기능

실수 'In'의 소수점 이하 1번째 자리를 처리해서 정수로 합니다.

● TRUNC

소수점 이하 1번째 자리를 버립니다.

● Round

소수점 이하 1번째 자리를 오사오입해서 정수로 합니다. 오사오입이란, 다음 표와 같은 처리를 의미합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1 -1.49→-1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2 -1.50→-2 -2.50→-2
0.5보다 큼	올림	1.51→2 -1.51→-2

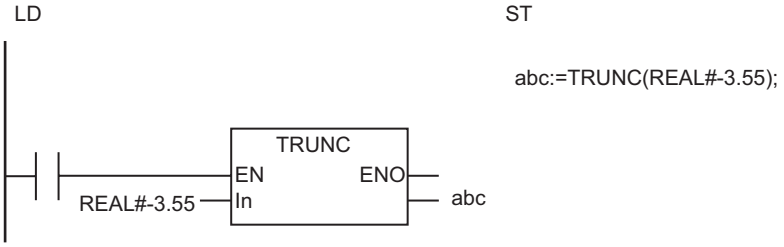
● RoundUp

소수점 이하 1번째 자리를 올립니다.

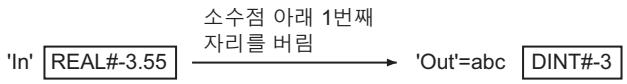
3개의 명령의 동작 차이를, 몇 가지 예를 통해 나타냅니다.

입력값	출력값		
	TRUNC	Round	RoundUp
REAL#1.6	DINT#1	DINT#2	DINT#2
REAL#1.5	DINT#1	DINT#2	DINT#2
REAL#1.4	DINT#1	DINT#1	DINT#2
REAL#2.5	DINT#2	DINT#2	DINT#3
REAL#-1.6	DINT#-1	DINT#-2	DINT#-2
REAL#-1.5	DINT#-1	DINT#-2	DINT#-2
REAL#-1.4	DINT#-1	DINT#-1	DINT#-2
REAL#-2.5	DINT#-2	DINT#-2	DINT#-3

TRUNC 명령에서 'In'= REAL#-3.55인 경우의 기술 예가 아래에 나와 있습니다. 변수 abc의 값은 DINT#-3이 됩니다.



'In'의 소수점 아래 1번째 자리를 버립니다.
'In'의 값이 REAL#-3.55이므로 abc의 값은 DINT#-3이 됩니다.



참고

'In'의 데이터 형식이 REAL이라면, 'Out'의 데이터 형식은 DINT가 됩니다. 'In'의 데이터 형식이 LREAL이라면, 'Out'의 데이터 형식은 LINT가 됩니다.

사용상의 주의 사항

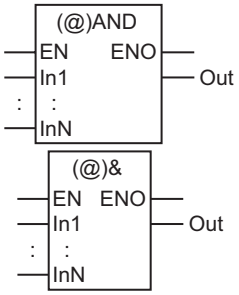
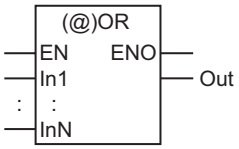
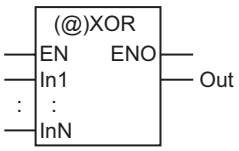
변환 결과가 'Out'의 유효 범위 밖인 경우, 'Out'의 값은 잘못된 값이 됩니다.

비트 열 연산 명령

명령	명칭	페이지
AND, &/OR/XOR	논리곱/논리합/배타적 논리합	2-306
XORN	배타적 논리합 부정	2-309
NOT	비트 반전	2-311
AryAnd/AryOr/ AryXor/AryXorN	배열 논리곱/배열 논리합/ 배열 배타적 논리합/ 배열 배타적 논리합 부정	2-312

AND, & / OR / XOR

복수의 부울, 비트 열의 1비트별로 연산을 수행합니다.
AND, & : 논리곱
OR : 논리합
XOR : 배타적 논리합

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AND, &	논리곱	FUN		Out:=In1 AND ··AND InN; Out:=In1 & ··& InN;
OR	논리합	FUN		Out:=In1 OR ··OR InN;
XOR	배타적 논리합	FUN		Out:=In1 XOR ··XOR InN;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	연산 대상	입력	연산 대상 N은 2~5	데이터 형식에 따름	–	0(*)
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN	○	○	○	○	○															
Out	'In1'~'InN'과 같은 데이터 형식																			

기능

복수의 부울 또는 비트 열의 연산 대상 'In1'~'InN'에 대해, 대응하는 비트별로 연산을 수행합니다.
따라서 'in1'~'InN' 및 'Out'은 같은 데이터 형식이어야 합니다.

연산 대상이 3개 이상인 경우, 처리 순서는 다음과 같습니다.

1 **'In1'과 'In2'를 연산**

2 **1의 결과와 'In3'을 연산**

3 **2의 결과와 'In4'를 연산**

: :

연산의 입출력 관계는 다음과 같습니다.

- **AND, &**

둘 다 TRUE일 때, 연산 결과가 TRUE 그 이외에는 연산 결과가 FALSE

'In1'의 비트	'In2'의 비트	'Out'의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

- **OR**

둘 다 FALSE일 때, 연산 결과는 FALSE 그 이외에는 연산 결과는 TRUE

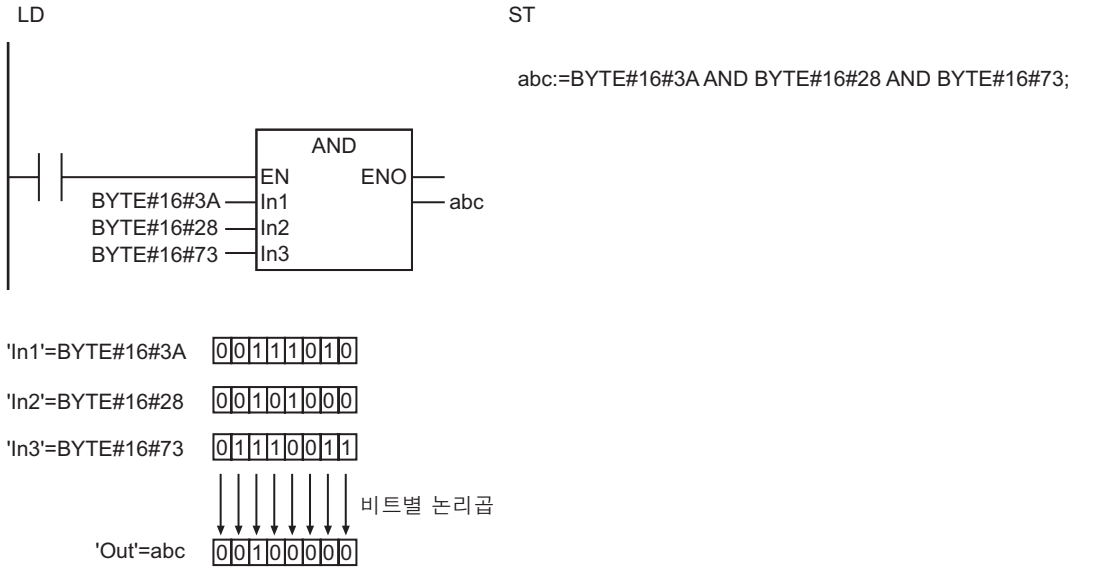
'In1'의 비트	'In2'의 비트	'Out'의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

- **XOR**

두 개의 값이 같은 경우, 연산 결과는 FALSE 다른 경우, 연산 결과는 TRUE

'In1'의 비트	'In2'의 비트	'Out'의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

AND 명령에서 'In1'=BYTE#16#3A, 'In2'＝BYTE#16#28, 'In3'=BYTE#16#73인 경우의 예가 아래에 나와 있습니다.



AND 명령과 & 명령의 기능은 완전히 동일합니다. 사용하기 편한 쪽을 사용하시기 바랍니다.

참고

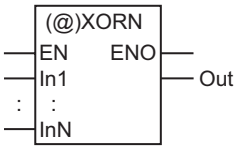
ST 언어에서 다음과 같은 기술을 하는 경우, 입력 변수의 수에는 제한이 없습니다.
Out: =In1 AND In2 AND In3 AND In4 AND In5 AND In6 • • •
Out: =In1 & In2 & In3 & In4 & In5 & In6 • • •
Out: =In1 OR In2 OR In3 OR In4 OR In5 OR In6 • • •
Out: =In1 XOR In2 XOR In3 XOR In4 XOR In5 XOR In6 • • •

사용상의 주의 사항

'In1'~'InN' 및 'Out'의 데이터 형식을 모두 동일하게 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.

XORN

복수의 부울, 비트 열의 1비트별로 배타적 논리합 부정 연산을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
XORN	배타적 논리합 부정	FUN		Out:=In1 XOR NOT ‥ XOR NOT InN;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	연산 대상	입력	연산 대상 N은 2~5	데이터 형식에 따름	–	0(*)
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN	○	○	○	○	○															
Out	'In1'~'InN'과 같은 데이터 형식																			

기능

복수의 부울 또는 비트 열의 연산 대상 'In1'~'InN'에 대해, 대응하는 비트별로 연산을 수행합니다.
따라서 'in1'~'InN' 및 'Out'은 같은 데이터 형식이어야 합니다.

연산 대상이 3개 이상인 경우, 처리 순서는 다음과 같습니다.

1 **'In1'과 'In2'를 연산**

2 **1의 결과와 'In3'을 연산**

3 **2의 결과와 'In4'를 연산**

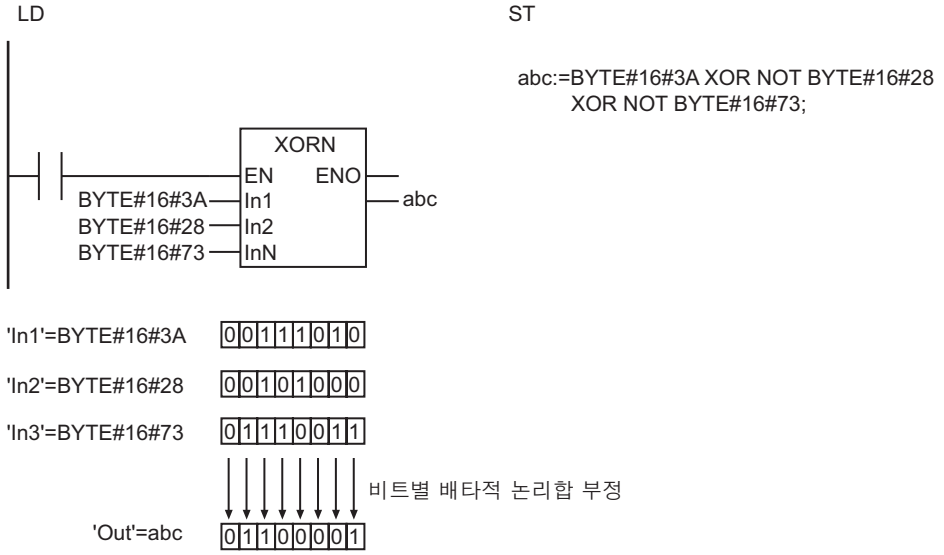
: :

연산의 입출력 관계는 다음과 같습니다. 두 개의 값이 같은 경우에는 연산 결과가 TRUE, 그 이외에는 FALSE가 됩니다.

'In1'의 비트	'In2'의 비트	'Out'의 비트
FALSE	FALSE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE

'In1'의 비트	'In2'의 비트	'Out'의 비트
TRUE	TRUE	TRUE

'In1'=BYTE#16#3A, 'In2'＝BYTE#16#28, 'In3'=BYTE#16#73인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

'In1'~'InN' 및 'Out'의 데이터 형식을 모두 동일하게 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.

NOT

부울, 비트 열의 각 비트를 반전합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
NOT	비트 반전	FUN	(@)NOT EN ENO In Out	Out:=NOT(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연산 대상	입력	연산 대상	데이터 형식에 따름	–	(*)
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

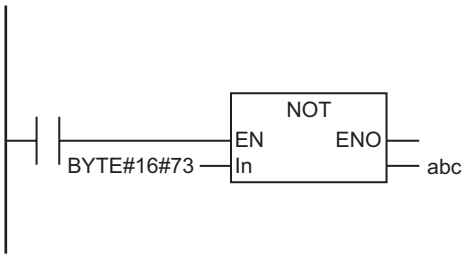
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○															
Out	'In'과 같은 데이터 형식																			

기능

부울 또는 비트 열의 연산 대상 'In'의 각 비트를 반전합니다.
따라서, 'In'과 연산 결과 'Out'은 비트 수가 같아야, 즉 동일한 데이터 형식이어야 합니다.

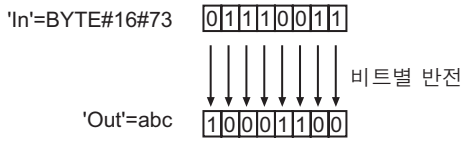
'In'=BYTE#16#73인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=NOT(BYTE#16#73);
```

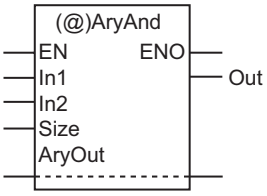
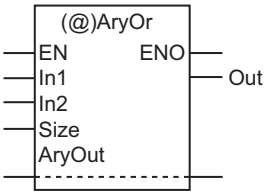
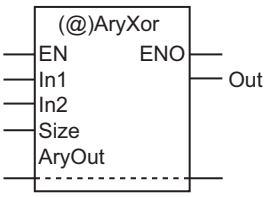
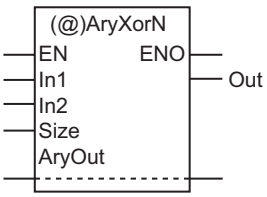


사용상의 주의 사항

'In'과 'Out'의 데이터 형식을 동일하게 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.

AryAnd / AryOr / AryXor / AryXorN

배열 사이의 각 요소의 부울, 비트 열의 1비트별 연산을 수행합니다.
AryAnd : 논리곱
AryOr : 논리합
AryXor : 배타적 논리합
AryXorN : 배타적 논리합 부정

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryAnd	배열 논리곱	FUN		AryAnd(In1, In2, Size, AryOut);
AryOr	배열 논리합	FUN		AryOr(In1, In2, Size, AryOut);
AryXor	배열 배타적 논리합	FUN		AryXor(In1, In2, Size, AryOut);
AryXorN	배열 배타적 논리합 부정	FUN		AryXorN(In1, In2, Size, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1[], In2[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	요소 수		연산 대상 요소 수			1
AryOut[] 배열	연산 결과 배열	입출력	연산 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1[] 배열	○	○	○	○	○															
In2[] 배열	In1[]과 같은 데이터 형식																			
Size							○													
AryOut[] 배열	In1[]과 같은 데이터 형식																			
Out	○																			

기능

연산 대상 배열 In1[], In2[] 앞의 'Size'개의 요소에 대해, 대응하는 요소의 대응 비트별로 논리 연산을 수행합니다. 연산 결과는 AryOut[]의 대응하는 요소에 저장됩니다.
따라서, In1[], In2[] 및 AryOut[]은 같은 데이터 형식이어야 합니다.

연산의 입출력 관계는 다음과 같습니다.

● AryAnd

둘 다 TRUE일 때, 연산 결과가 TRUE 그 이외에는 연산 결과가 FALSE

In1[]의 요소의 비트	In2[]의 요소의 비트	AryOut[]의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

● AryOr

둘 다 FALSE일 때, 연산 결과는 FALSE 그 이외에는 연산 결과는 TRUE

In1[]의 요소의 비트	In2[]의 요소의 비트	AryOut[]의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

● AryXor

두 개의 값이 같은 경우, 연산 결과는 FALSE 다른 경우, 연산 결과는 TRUE

In1[]의 요소의 비트	In2[]의 요소의 비트	AryOut[]의 비트
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

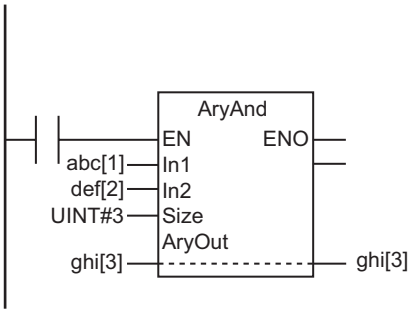
● AryXorN

두 개의 값이 같은 경우, 연산 결과는 TRUE 다른 경우, 연산 결과는 FALSE

In1[]의 요소의 비트	In2[]의 요소의 비트	AryOut[]의 비트
FALSE	FALSE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

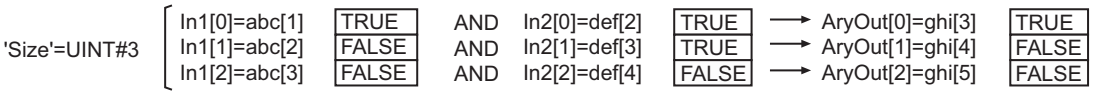
AryOut 명령에서 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
AryAnd(abc[1], def[2], UINT#3, ghi[3]);
```



사용상의 주의 사항

- In1[], In2[], AryOut[]의 데이터 형식은 모두 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- AryOut[]의 배열 요소 수는 'Size'개 이상으로 하십시오.
- 'Size'의 값이 0인 경우, AryOut[]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In1[], In2[], AryOut[] 중 어느 것의 요소 수를 초과한 경우

선택 명령

명령	명칭	페이지
SEL	비트 선택	2-316
MUX	멀티플렉서	2-318
LIMIT	상하한 리미트 제한	2-321
Band	데드 밴드 제어	2-323
Zone	데드 존 제어	2-325
MAX/MIN	최대값 검색/최소값 검색	2-327
AryMax/AryMin	배열 변수의 최대값 검색/ 배열 변수의 최소값 검색	2-329
ArySearch	배열 검색	2-332

SEL

2개의 선택지 가운데 1개를 선택합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SEL	비트 선택	FUN	(@)SEL EN ENO G Out In0 In1	Out: =SEL(G, In0, In1);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
G	게이트	입력	FALSE : 'In0'을 선택 TRUE : 'In1'을 선택	데이터 형식에 따름	–	FALSE
In0, In1	선택지		선택지			(*)
Out	선택 결과	출력	선택 결과	데이터 형식에 따름	–	–

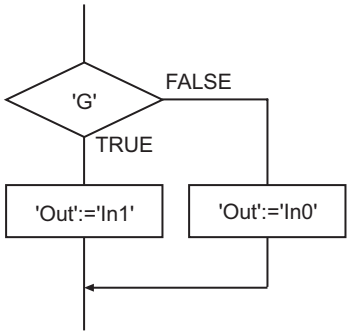
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	○																			
In0, In1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능*1																			
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능*1																			

*1. CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 지정할 수 있습니다.

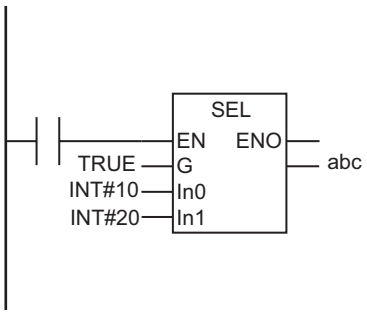
기능

2개의 선택지 'In0', 'In1' 가운데 1개를 선택합니다.
'In0'과 'In1' 중 어느 쪽을 선택할 것인지는 게이트 'G'로 지정합니다.
'G'의 값이 FALSE라면 'In0', TRUE라면 'In1'이 'Out'에 대입됩니다.



'In0'=INT#10, 'In1'=INT#20, 'G'=TRUE인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 INT#20이 됩니다.

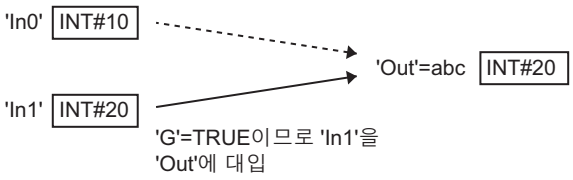
LD



ST

```
abc:=SEL(TRUE, INT#10, INT#20);
```

'In0', 'In1' 중 하나를 선택합니다.
'G'=TRUE이므로 'In1'=INT#20을 선택하여 abc에 대입합니다.



참고

버전 관련 정보

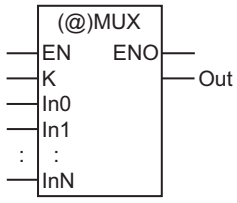
CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서는 'MUX 명령(2-318페이지)'을 사용하는 방법도 있습니다. MUX 명령은 2~5개의 선택지 중 하나를 선택하는 명령입니다.

사용상의 주의 사항

- 'In0', 'In1', 'Out'은 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In0', 'In1'의 유효 범위를 포함하도록 하십시오.
  - 'In0', 'In1', 'Out'은 종류가 다른 데이터 형식(비트 열과 정수, 정수와 문자열 등)으로 하지 마십시오.

MUX

2~5개의 선택지 가운데 1개를 선택합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MUX	멀티플렉서	FUN		Out: =MUX(K, In0, In1, ⋯ , InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
K	셀렉터	입력	0 : 'In0'을 선택 1 : 'In1'을 선택 2 : 'In2'를 선택 3 : 'In3'을 선택 4 : 'In4'를 선택	0~N	–	*1
In0~InN	선택지		선택지 N은 1~4*2	데이터 형식에 따름		0*3
Out	선택 결과	출력	선택 결과	데이터 형식에 따름	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 CPU 유니트 Ver.1.01 이하 또는 Sysmac Studio Ver.1.02 이하에서 N의 범위는 2~4입니다.
*3 InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=2인 경우, In0과 In1에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In2에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
K						○ *1			○ *1											
In0~InN	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능*2																			
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식도 지정 가능*2																			

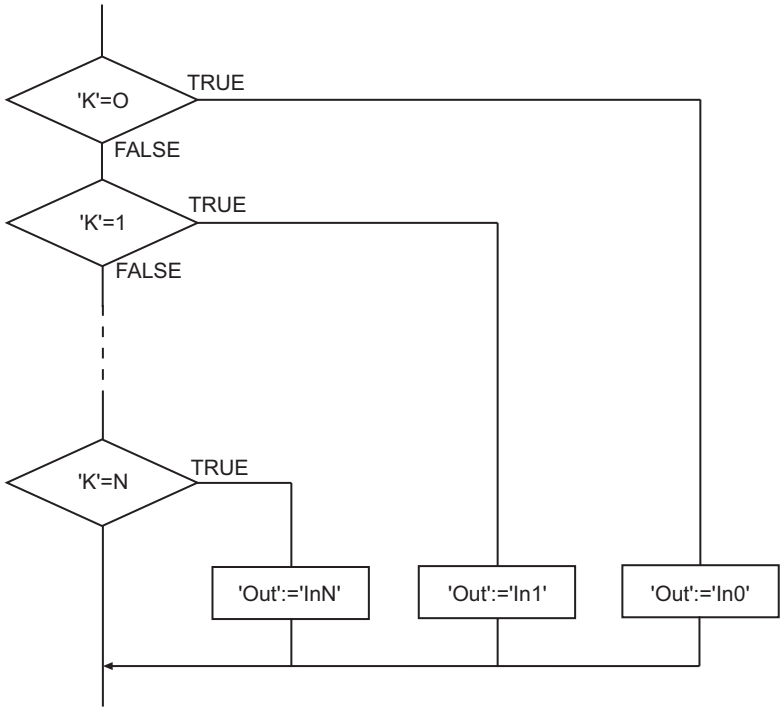
*1 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서는 ULINT 형식입니다. CPU 유니트 Ver.1.01 이하 또는 Sysmac Studio Ver.1.02 이하에서는 USINT 형식입니다.
*2 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 지정할 수 있습니다.

기능

2~5개의 선택지 'In0'~'InN' 가운데 1개를 선택합니다.

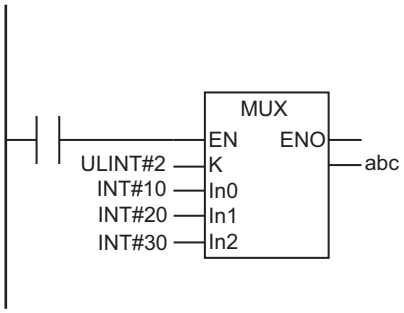
'In0'~'InN' 중 어느 것을 선택할 것인지는 셀렉터 'K'로 지정합니다.

'K'의 값이 0이라면 'In0', 1이라면 'In1', 이하 'K'의 값에 따른 선택지의 값이 'Out'에 대입됩니다.



'In0'=INT#10, 'In1'=INT#20, 'In2'=INT#30, 'K'=ULINT#2인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 INT#30이 됩니다.

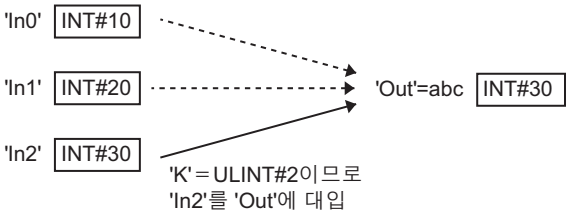
LD



ST

```
abc:=MUX(ULINT#2, INT#10, INT#20, INT#30);
```

'In0'~'InN' 중 하나를 선택합니다.
'K'＝ULINT#2이므로 'In2'=INT#30을 선택하여 abc에 대입합니다.



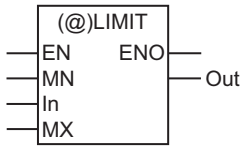
사용상의 주의 사항

- 'In0'~'InN' 및 'Out'은 각각 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In0'~'InN'의 모든 유효 범위를 포함하도록 하십시오.
  - 'In0'~'InN', 'Out'은 종류가 다른 데이터 형식(비트 열과 정수, 정수와 문자열 등)으로 하지 마십시오.

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'K'의 값이 유효 범위 밖(0 미만 또는 N보다 큼)인 경우우

LIMIT

설정된 상하한값으로 입력 변수의 값을 제한합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LIMIT	상하한 리미트 제한	FUN		Out: =LIMIT(MN, In, MX);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
MN	하한값	입력	제한의 하한값	데이터 형식에 따름	–	(*)
In	제한 대상		제한 대상			
MX	상한값		제한의 상한값			
Out	처리 결과	출력	처리 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN						○	○	○	○	○	○	○	○	○	○					
In						○	○	○	○	○	○	○	○	○	○					
MX						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

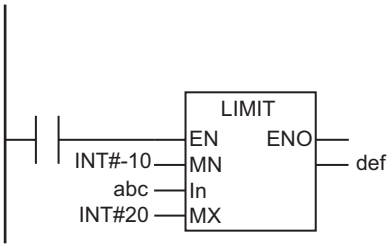
기능

상한값 'MX', 하한값 'MN'에 따라 제한 대상 'In'의 값을 제한합니다.
처리 결과 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
'In' < 'MN'	'MN'
'MN' ≦ 'In' ≦ 'MX'	'In'
'MX' < 'In'	'MX'

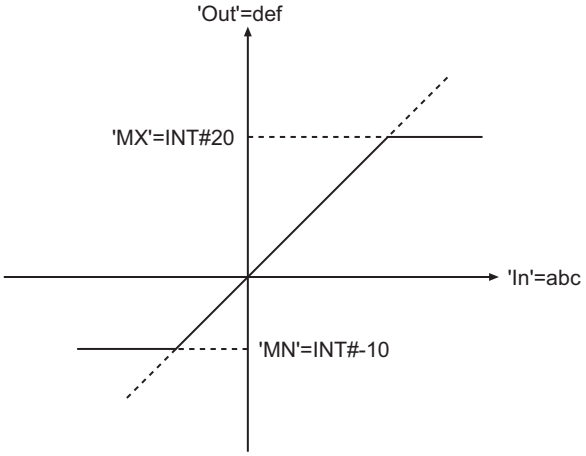
'MN'=INT#-10, 'MX'=INT#20인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=LIMIT(INT#-10, abc, INT#20);
```

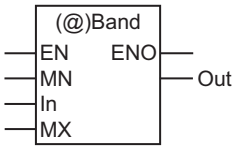


사용상의 주의 사항

- 'In', 'MN', 'MX' 및 'Out'은 각각 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In', 'MN', 'MX'의 모든 유효 범위를 포함하도록 하십시오.
  - 'In', 'MN', 'MX'는 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)이 함께 쓰이지 않도록 하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'MX'의 값이 'MN'의 값보다 작은 경우

Band

데드 밴드(불감대)를 제어합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Band	데드 밴드 제어	FUN		Out: =Band(MN, In, MX);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
MN	하한값	입력	데드 밴드의 하한값	데이터 형식에 따름	–	(*)
In	제어 대상		제어 대상			
MX	상한값		데드 밴드의 상한값			
Out	처리 결과	출력	처리 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN										○	○	○	○	○	○					
In										○	○	○	○	○	○					
MX										○	○	○	○	○	○					
Out										○	○	○	○	○	○					

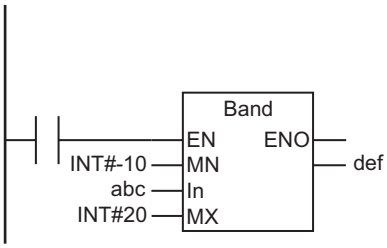
기능

상한값 'MX', 하한값 'MN'에 따라 제어 대상 'In'의 값을 제어합니다.
처리 결과 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
'In' < 'MN'	'In' - 'MN'
'MN' ≦ 'In' ≦ 'MX'	0
'MX' < 'In'	'In' - 'MX'

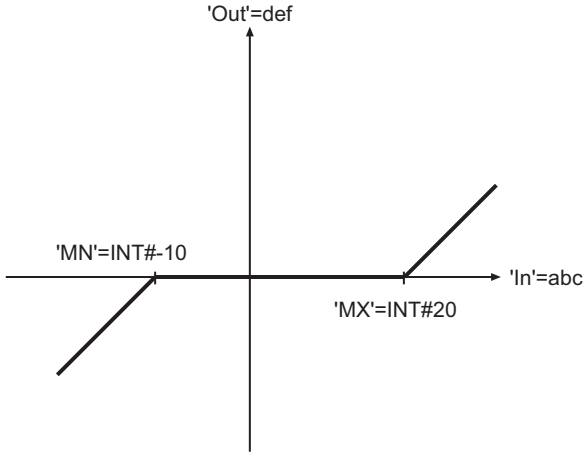
'MN'=INT#-10, 'MX'=INT#20인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=Band(INT#-10, abc, INT#20);
```



사용상의 주의 사항

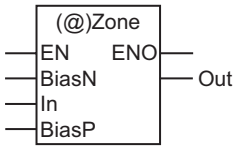
- 'In', 'MN', 'MX' 및 'Out'은 각각 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In', 'MN', 'MX'의 모든 유효 범위를 포함하도록 하십시오.
- 'In'의 값이 비수인 경우, 'Out'의 값은 비수가 됩니다.
- 'In', 'MN', 'MX'의 값이 +∞, -∞인 경우, 'Out'의 값은 다음과 같습니다.

'In'의 값	'MN'의 값	'MX'의 값	'Out'의 값
+∞	+∞	+∞	0
		-∞	이상
	-∞	+∞	0
		-∞	+∞
-∞	+∞	+∞	-∞
		-∞	이상
	-∞	+∞	0
		-∞	0

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'MX'의 값이 'MN'의 값보다 작은 경우
  - 'MX', 'MN' 중 어느 것이 비수인 경우
  - 처리 결과가 'Out'의 유효 범위 밖인 경우

Zone

입력값에 바이어스값을 부가합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Zone	데드 존 제어	FUN		Out: =Zone(BiasN, In, BiasP);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
BiasN	음의 바이어스값	입력	음의 바이어스값	데이터 형식에 따름	–	(*)
In	제어 대상		제어 대상			
BiasP	양의 바이어스값		양의 바이어스값			
Out	처리 결과	출력	처리 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BiasN										○	○	○	○	○	○					
In										○	○	○	○	○	○					
BiasP										○	○	○	○	○	○					
Out										○	○	○	○	○	○					

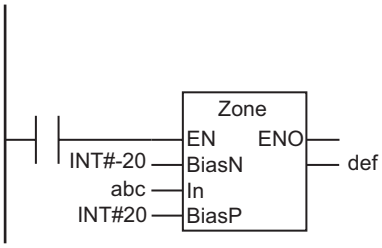
기능

양의 바이어스값 'BiasP', 음의 바이어스값 'BiasN'에 따라 제어 대상 'In'의 값을 제어합니다.
처리 결과 'Out'의 값은 다음과 같습니다.

'In'의 값	'Out'의 값
'In' < 0	'In' + 'BiasN'
'In' = 0	0
0 < 'In'	'In' + 'BiasP'

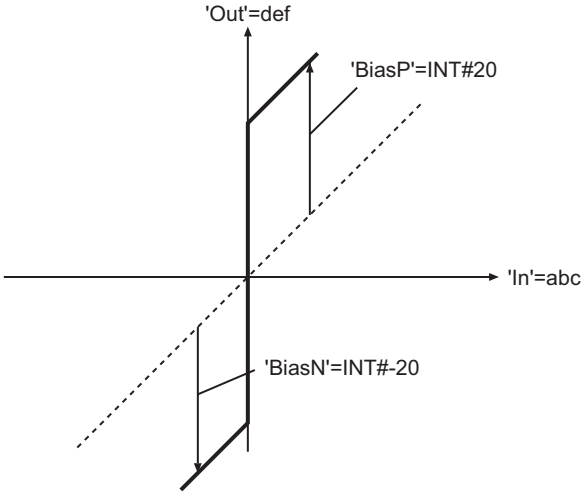
'BiasP'=INT#20, 'BiasN'=INT#-20인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=Zone(INT#-20, abc, INT#20);
```



사용상의 주의 사항

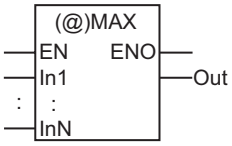
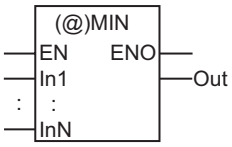
- 'In', 'BiasP', 'BiasN' 및 'Out'은 각각 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In', 'BiasP', 'BiasN'의 모든 유효 범위를 포함하도록 하십시오.
- 'In'의 값이 비수인 경우, 'Out'의 값은 비수가 됩니다.
- 'In', 'BiasP', 'BiasN'의 값이 +∞, -∞인 경우, 'Out'의 값은 다음과 같습니다.

'In'의 값	'BiasP'의 값	'BiasN'의 값	'Out'의 값
+∞	+∞	+∞	+∞
		-∞	+∞
	-∞	+∞	이상
		-∞	0
-∞	+∞	+∞	0
		-∞	-∞
	-∞	+∞	이상
		-∞	-∞

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'BiasP' < 'BiasN'의 경우
  - 'BiasP', 'BiasN' 중 어느 것이 비수인 경우
  - 처리 결과가 'Out'의 유효 범위 밖인 경우

MAX / MIN

MAX: 2~5개의 수치 가운데 최대값을 검색합니다.
MIN : 2~5개의 수치 가운데 최소값을 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MAX	최대값 검색	FUN		Out: =MAX(In1, In2, ⋯ , InN);
MIN	최소값 검색	FUN		Out: =MIN(In1, In2, ⋯ , InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	연산 대상	입력	연산 대상 N은 2~5	데이터 형식에 따름	–	0(*)
Out	검색 결과	출력	검색 결과	데이터 형식에 따름	–	–

* InN에 연결하는 입력 파라메터를 생략한 경우, 초기값은 적용되지 않으며 빌드 시에 이상 상태가 됩니다. 예를 들면 N=3 인 경우, In1과 In2에 연결하는 입력 파라메터를 생략하면 초기값이 적용되지만, In3에 연결하는 입력 파라메터를 생략하면 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

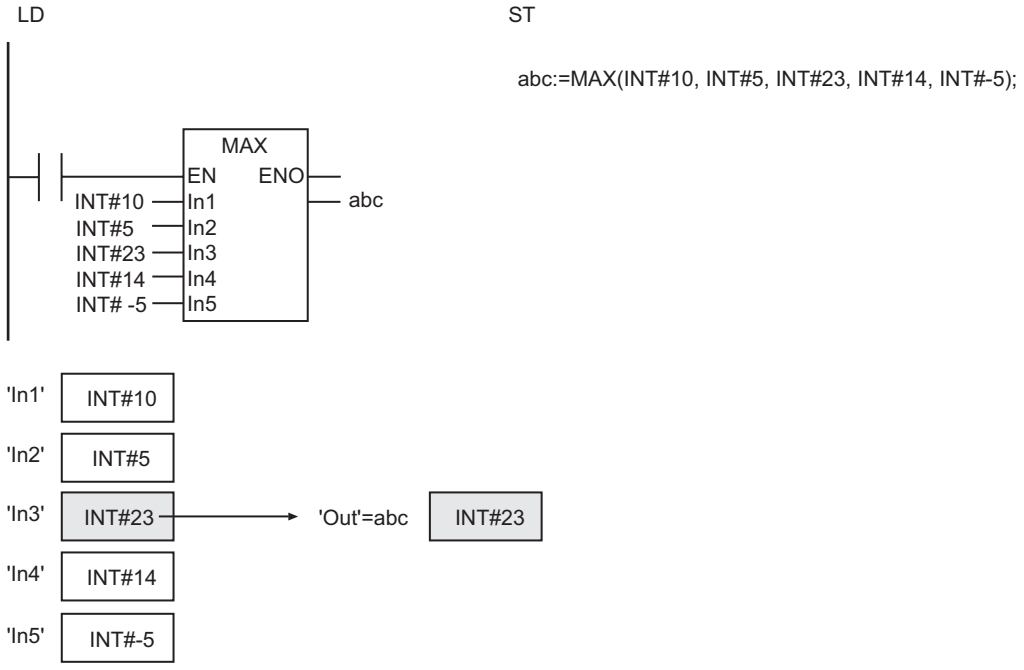
- **MAX**

  2~5개의 연산 대상 'In1'~'InN' 가운데 최대값을 검색합니다.

- **MIN**

  2~5개의 연산 대상 'In1'~'InN' 가운데 최소값을 검색합니다.

MAX 명령에서 'In1'=INT#10, 'In2'=INT#5, 'In3'=INT#23, 'In4'=INT#14, 'In5'=INT#-5의 예가 아래에 나와 있습니다.



참고

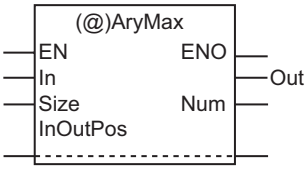
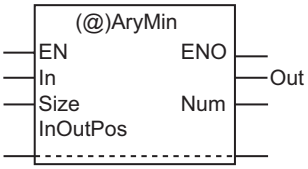
6개 이상의 연산 대상의 최대값/최소값을 검색하려는 경우, 🕮 'AryMax / AryMin 명령(2-329페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In1'~'InN' 및 'Out'은 각각 다른 데이터 형식이어도 상관없지만, 다음 사항에 주의하십시오.
  - 'Out'의 유효 범위가 'In1'~'InN'의 모든 유효 범위를 포함하도록 하십시오.
  - 'In1'~'InN'에는 부호가 있는 정수 형식(SINT, INT, DINT, LINT)과 부호가 없는 정수 형식(USINT, UINT, UDINT, ULINT)이 함께 쓰이지 않도록 하십시오.
- 'In1'~'InN'이 실수인 경우, 오차의 영향으로 기대한 결과를 얻을 수 없는 경우가 있습니다.

AryMax / AryMin

AryMax : 1차원 배열 요소의 최대값을 검색합니다.
AryMin : 1차원 배열 요소의 최소값을 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryMax	배열 변수의 최대값 검색	FUN		Out: =AryMax(In, Size, InOutPos, Num);
AryMin	배열 변수의 최소값 검색	FUN		Out: =AryMin(In, Size, InOutPos, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	검색 대상 배열	입력	검색 대상 배열	데이터 형식에 따름	–	(*)
Size	검색 대상의 요소 수		In[]의 요소 가운데 검색 대상인 것의 수			1
InOutPos	검색 요소 번호	입출력	검색값의 배열 요소 번호	데이터 형식에 따름	–	–
Out	검색 결과	출력	검색 결과	데이터 형식에 따름	–	–
Num	검색값의 수		검색값의 수			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[ ] 배열						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
																(*)				
Size							○													
InOutPos							○													
Out						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
																(*)				
Num							○													

* TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

검색 대상 배열 In[ ]의 In[0]에서 시작되는 'Size'개의 배열 요소에 대해 검색을 수행합니다.

검색값을 'Out'에, 검색 요소 번호를 'InOutPos'에, 검색값의 수를 'Num'에 각각 대입합니다. 'Num'이 1보다 큰 경우, 'InOutPos'의 값은 검색값 중 가장 하위의 요소 번호가 됩니다.

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.
STRING	'LTascii / LEascii / GTascii / GEascii 명령(2-105페이지)' 과 동일한 사양입니다. 해당 부분을 참조하십시오.

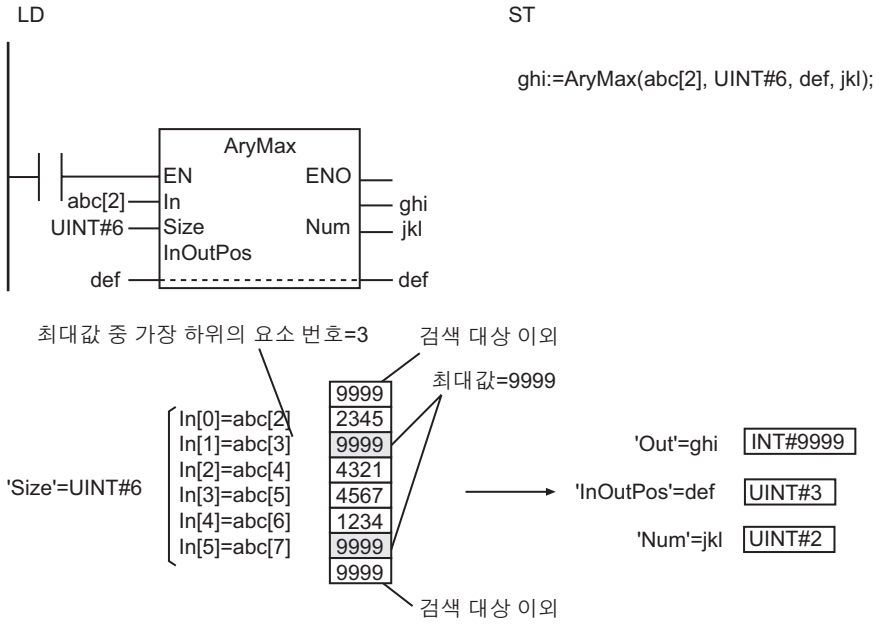
● AryMax

최대값을 검색합니다.

● AryMin

최소값을 검색합니다.

AryMax 명령에서 'Size'=UINT#6인 경우의 예가 아래에 나와 있습니다.
In[]에 전달하는 입력 파라메터는 abc[2]이므로, abc[2] 이후가 검색 대상이 됩니다.



참고

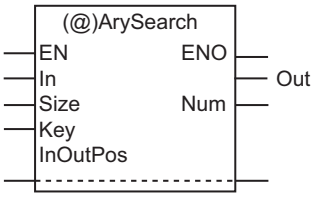
TIME 형식, DT 형식, TOD 형식의 크기를 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 'TruncTime 명령(2-628페이지)', 'TruncDt 명령(2-632페이지)', 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- In[]과 'Out'의 데이터 형식이 다른 경우에는 'Out'의 유효 범위가 In[]의 유효 범위를 포함하도록 하십시오.
- In[]이 실수인 경우, 오차의 영향으로 기대한 결과를 얻을 수 없는 경우가 있습니다.
- In[]은 반드시 1차원 배열로 하십시오.
- 'Size'의 값이 0인 경우, 'Out', 'Num'의 값은 0이 됩니다. 또한 'InOutPos'의 값은 변화하지 않습니다.
- In[]이 STRING 형식이고 'Size'의 값이 0인 경우, 'Out'은 NULL 문자뿐입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Size'가 In[]의 배열 영역을 초과한 경우
  - In[]이 1차원 배열이 아닌 경우
  - In[]이 STRING 형식이고 NULL 문자로 끝나지 않은 경우

ArySearch

1차원 배열에서 지정된 값을 검색합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ArySearch	배열 검색	FUN		Out:=ArySearch(In, Size, Key, InOutPos, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	검색 대상 배열	입력	검색 대상 배열	데이터 형식에 따름	–	(*)
Size	검색 대상의 요소 수		In[]의 요소 가운데 검색 대상인 것의 수			1
Key	검색 키		검색하는 값	데이터 형식에 따름		(*)
InOutPos	검색 요소 번호	입출력	검색값의 배열 요소 번호	데이터 형식에 따름	–	–
Out	검색 결과	출력	검색 결과	데이터 형식에 따름	–	–
Num	검색값의 수		검색값의 수			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○ *1	○	○	○	○
	열거 형식 배열도 지정 가능																			
Size							○													
Key	In[]의 요소와 동일한 데이터 형식																			
InOutPos							○													
Out	○																			
Num							○													

*1. TIME 형식, DATE 형식, TOD 형식, DT 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

1차원의 검색 대상 배열 In[]의 In[0]에서 시작되는 'Size'개의 배열 요소에 대해 검색 키 'Key'와 같은 값의 요소를 검색합니다.

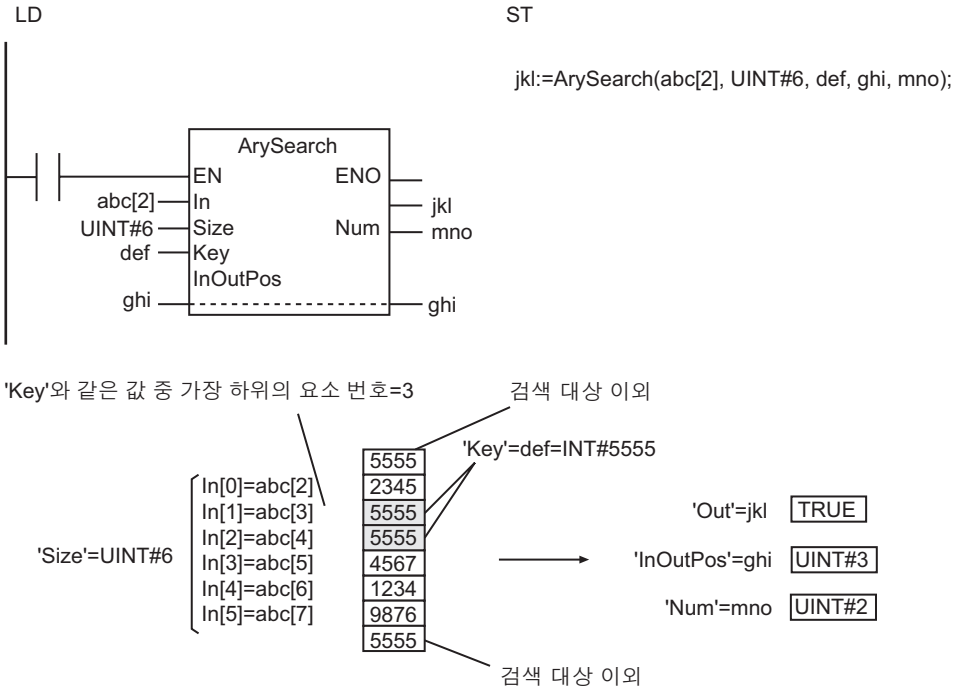
검색 결과 'Out', 검색 요소 번호 'InOutPos', 검색값의 수 'Num'의 값은 다음과 같습니다.

'Key'와 같은 값의 요소의 존재	'Out'	'InOutPos'	'Num'
존재함	TRUE	'Key'와 같은 값의 요소 가운데 가장 하위에 있는 것의 요소 번호	'Key'와 같은 값의 요소의 수
존재하지 않음	FALSE	변화 없음	0

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.

'Size'=UINT#6인 경우의 예가 아래에 나와 있습니다.
In[]에 전달하는 입력 파라메터는 abc[2]이므로, abc[2] 이후가 검색 대상이 됩니다.



참고

TIME 형식, DT 형식, TOD 형식의 크기를 비교하는 경우, 비교할 값 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- In[]은 반드시 1차원 배열로 하십시오.
- 'Key'는 In[]의 요소와 동일한 데이터 형식으로 하십시오.
- 'Size'의 값이 0인 경우, 'Out', 'Num'의 값은 0이 됩니다. 또한 'InOutPos'의 값은 변화하지 않습니다.
- 'Key'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.

- 'Key'가 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', 'Num', 'InOutPos'는 변화하지 않습니다.
  - 'Size'가 In[]의 배열 영역을 초과한 경우
  - In[]이 STRING 형식이고 NULL 문자로 종단되지 않은 경우
  - In[]이 1차원 배열이 아닌 경우

데이터 전송 명령

명령	명칭	페이지
MOVE	전송	2-336
MoveBit	비트 전송	2-339
MoveDigit	자릿수 전송	2-341
TransBits	다중 비트 전송	2-343
MemCopy	메모리 복사	2-345
SetBlock	블록 설정	2-347
Exchange	데이터 교환	2-349
AryExchange	배열 데이터 교환	2-351
AryMove	배열의 전송	2-353
Clear	초기화	2-355
Copy**ToNum (비트 열→부호가 있는 정수)	비트 패턴 복사(비트 열→부호가 있는 정수) 그룹	2-357
Copy**To*** (비트 열→실수)	비트 패턴 복사(비트 열→실수) 그룹	2-359
CopyNumTo** (부호가 있는 정수→비트 열)	비트 패턴 복사(부호가 있는 정수→비트 열) 그룹	2-361
CopyNumTo** (부호가 있는 정수→실수)	비트 패턴 복사(부호가 있는 정수→실수) 그룹	2-363
Copy**To*** (실수→비트 열)	비트 패턴 복사(실수→비트 열) 그룹	2-365
Copy**ToNum (실수→부호가 있는 정수)	비트 패턴 복사(실수→부호가 있는 정수) 그룹	2-366

MOVE

변수 및 정수의 값을 다른 변수로 전송합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MOVE	전송	FUN	(@)MOVE EN ENO In Out	Out:=In;

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	전송원	입력	전송원	데이터 형식에 따름	–	(*)
Out	전송대상	출력	전송대상	데이터 형식에 따름	–	(*)

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

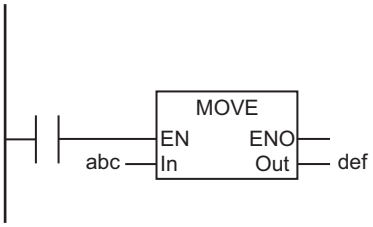
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	'In'이 열거 형식, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버인 경우, 'In'과 같은 데이터 형식을 지정 'In'이 배열 전체인 경우, 'In'과 같은 데이터 형식, 크기, 첨자를 갖는 배열을 지정																			

기능

전송원 'In'의 값을 전송대상 'Out'으로 전송합니다.
'In'에 전달하는 입력 파라메터는 정수이거나 변수이거나 상관없습니다. 또한, 'In'에 열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버를 지정할 수도 있습니다.

기술 예가 아래에 나와 있습니다. 변수 abc의 내용이 변수 def로 전송됩니다.

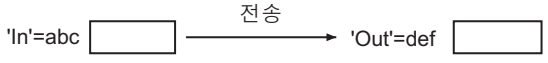
LD



ST

```
def:=abc;
```

'In'의 값을 'Out'으로 전송합니다.

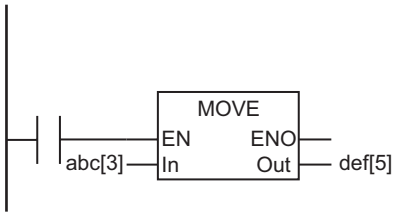


참고

- 배열을 전송하는 경우, 요소를 1개만 전송하거나 배열 전체를 전송할 수 있습니다. 요소를 1개만 전송하는 경우, 배열 변수명에 첨자를 붙입니다. 배열 전체를 전송하는 경우, 배열 변수명에 첨자를 붙이지 않습니다.

  배열 요소를 1개만 전송하는 경우

  LD

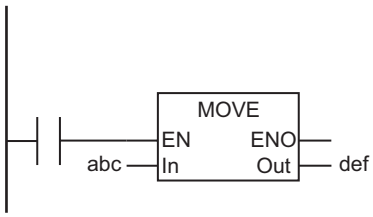
  

  ST

  ```
  def[5]:=abc[3];
  ```

  배열 전체를 전송하는 경우

  LD

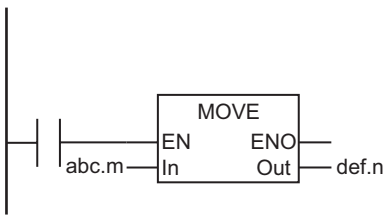
  ST

  ```
  def:=abc;
  ```

- 구조체를 전송하는 경우, 멤버를 1개만 전송하거나 구조체 전체를 전송할 수 있습니다. 멤버를 1개만 전송하는 경우, 멤버를 명시합니다. 구조체 전체를 전송하는 경우, 구조체명만 기술합니다.

  구조체의 멤버를 1개만 전송하는 경우

  LD

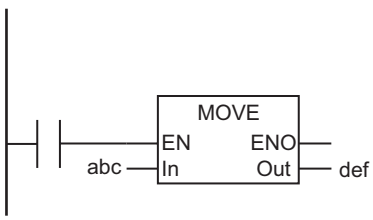
  

  ST

  ```
  def.n:=abc.m;
  ```

  구조체 전체를 전송하는 경우

  LD

  ST

  ```
  def:=abc;
  ```

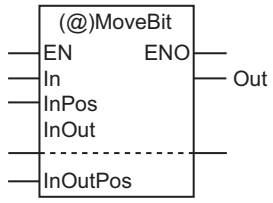
- 배열 전체를 전송하는 경우, 이 명령보다 MemCopy 명령이 더 고속으로 처리할 수 있습니다.

사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식이 다른 경우, 다음 중 하나의 데이터 형식 중에서 'Out'의 유효 범위가 'In'의 유효 범위를 포함하는 조합으로 하십시오.
  - BYTE, WORD, DWORD, LWORD
  - USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL
- 'In'이 열거 형식, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버인 경우, 'Out'은 그와 같은 데이터 형식으로 하십시오.
- 'In'이 배열 전체인 경우, 'Out'은 그와 같은 데이터 형식, 크기, 첨자를 갖는 배열로 하십시오.

MoveBit

비트 열 내의 1비트를 전송합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MoveBit	비트 전송	FUN		MoveBit(In, InPos, InOut, InOutPos);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	전송원	입력	전송원	데이터 형식에 따름	–	(*)
InPos	전송원의 비트 위치		'In' 중에서 전송되는 비트의 위치	0~'In'의 비트 수-1		0
InOutPos	전송대상의 비트 위치		'Out' 중에서 전송대상 비트의 위치	0~'InOut'의 비트 수-1		
InOut	전송대상	입출력	전송대상	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

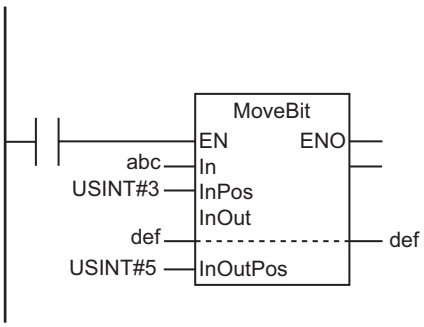
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
InPos						○														
InOutPos						○														
InOut		○	○	○	○															
Out	○																			

기능

전송원 'In'의 비트 위치 'InPos'에 있는 1비트 데이터를 전송대상 'InOut'의 비트 위치 'InOutPos'로 전송합니다.

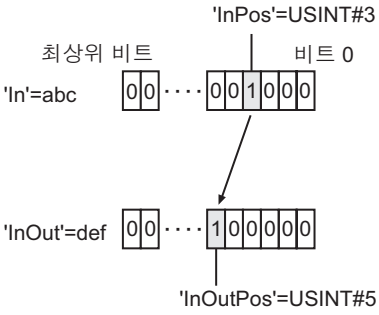
'InPos'=USINT#3, 'InOutPos'=USINT#5인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
MoveBit(abc, USINT#3, def, USINT#5);
```

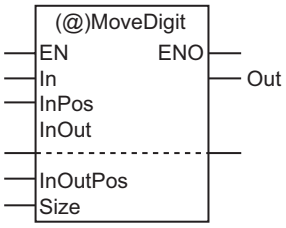


사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'InOut'은 변화하지 않습니다.
  - 'InPos'의 값이 유효 범위 밖인 경우
  - 'InOutPos'의 값이 유효 범위 밖인 경우

MoveDigit

비트 열 내의 복수 자릿수(1자릿수는 4비트)를 전송합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MoveDigit	자릿수 전송	FUN		MoveDigit(In, InPos, InOut, InOutPos, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	전송원	입력	전송원	데이터 형식에 따름	–	(*1)
InPos	전송원의 자리 위치		'In' 중에서 전송되는 자리의 위치	(*2)		0
InOutPos	전송대상의 자리 위치		'Out' 중에서 전송대상 자리의 위치	(*3)		
Size	전송 자릿수		전송하는 자릿수	(*4)		1
InOut	전송대상	입출력	전송대상	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 0~'In'의 비트 수/4- 1
*3 0~'InOut'의 비트 수/4 - 1
*4 0~'In'의 비트 수/4

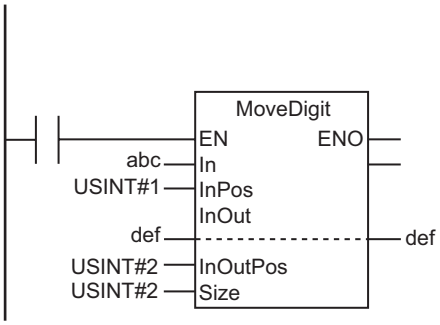
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
InPos						○														
InOutPos						○														
Size						○														
InOut		○	○	○	○															
Out	○																			

기능

전송원 'In'의 자리 위치 'InPos'에서 'Size' 자리만큼을 전송대상 'InOut'의 자리 위치 'InOutPos'로 전송합니다. 1자리는 4비트입니다.

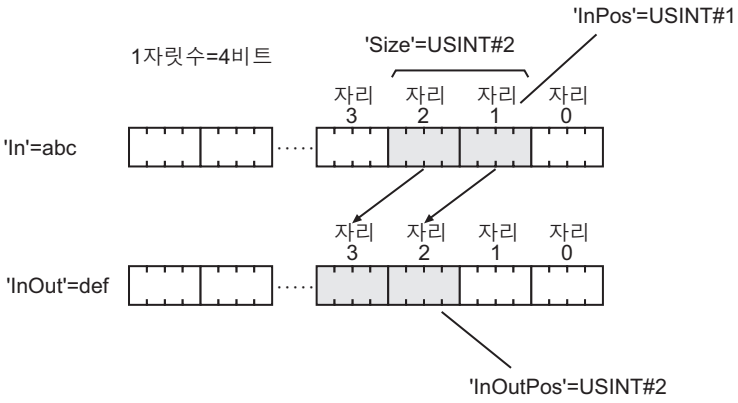
'InPos'=USINT#1, 'InOutPos'=USINT#2, 'Size'=USINT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
MoveDigit(abc, USINT#1, def, USINT#2, USINT#2);
```

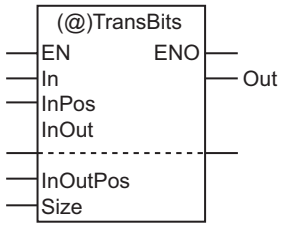


사용상의 주의 사항

- 전송대상의 자리 위치가 'InOut'의 최상위 자리를 초과하도록 지정한 경우, 초과한 데이터는 'InOut'의 최하위 자리로 이동해서 저장됩니다.
- 전송원의 자리 위치가 'In'의 최상위 자리를 초과하도록 지정한 경우, 초과한 데이터는 'In'의 최하위 자리로 이동해서 전송됩니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 'InOut'은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'InOut'은 변화하지 않습니다.
  - 'InPos'의 값이 유효 범위 밖인 경우
  - 'InOutPos'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 유효 범위 밖인 경우

TransBits

비트 열 내의 복수 비트를 전송합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TransBits	다중 비트 전송	FUN		TransBits(In, InPos, InOut, InOutPos, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	전송원	입력	전송원	데이터 형식에 따름	–	(*1)
InPos	전송원의 비트 위치		'In' 중에서 전송되는 비트의 위치	(*2)		0
InOutPos	전송대상의 비트 위치		'Out' 중에서 전송대상 비트의 위치	(*3)		
Size	전송 비트 수		전송하는 비트 수	(*4)		1
InOut	전송대상	입출력	전송대상	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 0~'In'의 비트 수 - 1
*3 0~'InOut'의 비트 수 - 1
*4 0 ~'In'의 비트 수

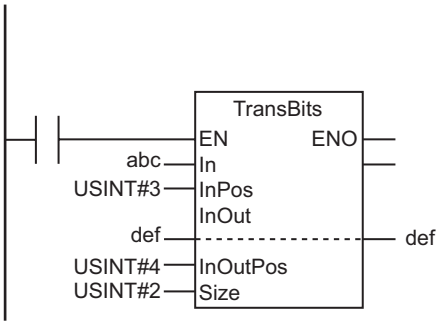
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
InPos						○														
InOutPos						○														
Size						○														
InOut		○	○	○	○															
Out	○																			

기능

전송원 'In'의 비트 위치 'InPos'에서 'Size' 비트만큼을 전송대상 'InOut'의 비트 위치 'InOutPos'로 전송합니다.

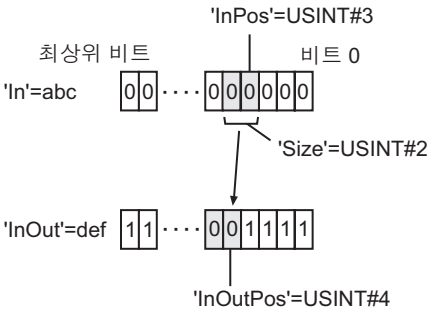
'InPos'=USINT#3, 'InOutPos'=USINT#4, 'Size'=USINT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
TransBits(abc, USINT#3, def, USINT#4, USINT#2);
```



참고

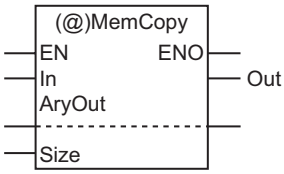
전송원과 전송대상의 데이터 영역이 겹치더라도 상관없습니다.

사용상의 주의 사항

- 전송원 및 전송대상의 비트 위치가 'In' 및 'InOut'의 최상위 비트를 초과하도록 지정하지 마십시오. 그 경우, 이상 상태가 되어 동작하지 않습니다.
- 'Size'의 값이 0인 경우, 전송이 수행되지 않습니다.
- 'InOut'에서 전송과 관계없는 비트는 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'InOut'은 변화하지 않습니다.
  - 'InPos'의 값이 유효 범위 밖인 경우
  - 'InOutPos'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'InPos'와 'Size'의 지정이 'In'의 비트 수를 초과하는 경우
  - 'InOutPos'와 'Size'의 지정이 'InOut'의 비트 수를 초과하는 경우

MemCopy

복수의 배열 요소를 전송합니다. 전송원과 전송대상은 동일한 데이터 형식에 한합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MemCopy	메모리 복사	FUN		MemCopy(In, AryOut, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	전송원 배열	입력	전송원 배열	데이터 형식에 따름	–	(*)
Size	전송 요소 수		전송하는 배열 요소 수			1
AryOut[] 배열	전송대상 배열	입출력	전송대상 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

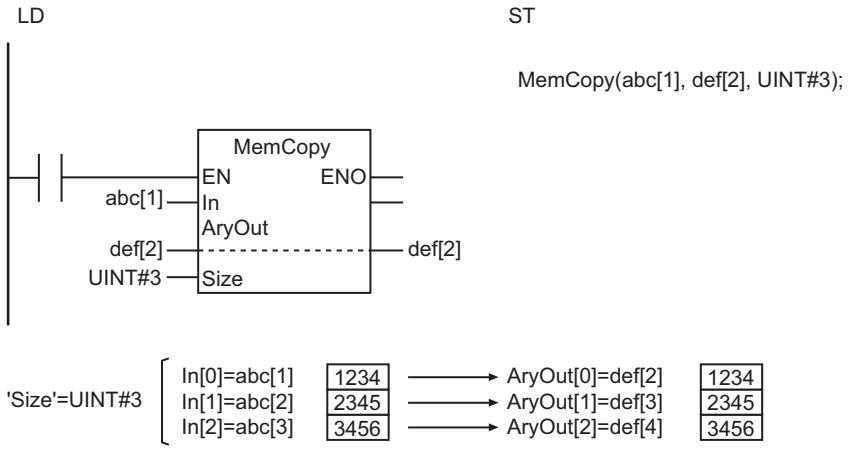
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Size							○													
AryOut[] 배열	In[]과 동일한 데이터 형식의 배열																			
Out	○																			

기능

전송원 배열 In[]의 In[0]에서 시작되는 'Size'개의 요소를 전송대상 배열인 AryOut[]의 AryOut[0] 이후로 전송합니다.

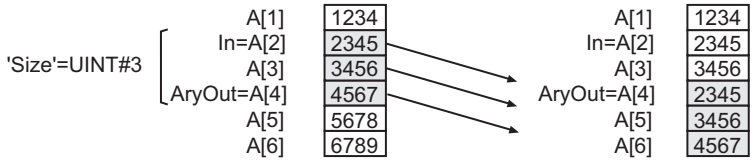
'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.



참고

- In[]과 AryOut[]에, 같은 배열의 다른 위치를 지정할 수도 있습니다. 또한, 위치가 중복되더라도 상관없습니다.

  In=A[2], AryOut=A[4], 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.



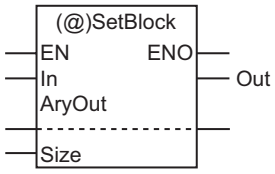
- 전송원과 전송대상의 데이터 형식이 다른 경우, 🕮 'AryMove 명령(2-353페이지)'을 사용하십시오.
- 전송원과 전송대상의 데이터 형식이 같은 경우, AryMove보다 이 명령이 고속으로 처리할 수 있습니다.
- 배열이 아닌 변수를 전송하는 경우, 🕮 'MOVE 명령(2-336페이지)'을 사용하십시오.

사용상의 주의 사항

- In[]과 AryOut[]의 데이터 형식을 동일하게 하십시오. 동일하지 않으면 빌드 시에 이상 상태가 됩니다.
- In[]과 AryOut[]이 STRING 형식의 배열인 경우, 각각의 영역 크기를 일치시키십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'가 In[]의 배열 영역을 초과한 경우
  - 'Size'가 AryOut[]의 배열 영역을 초과한 경우

SetBlock

복수의 배열 요소에 변수 및 정수의 값을 전송합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetBlock	블록 설정	FUN		SetBlock(In, AryOut, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	전송원	입력	전송원	데이터 형식에 따름	–	(*)
Size	전송 요소 수		전송하는 배열 요소 수			1
AryOut[] 배열	전송대상 배열	입출력	전송대상 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

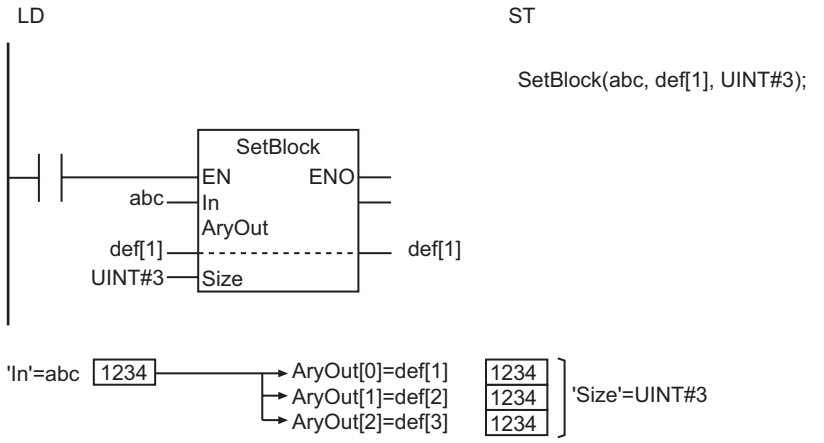
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Size							○													
AryOut[] 배열	'In'과 같은 데이터 형식을 요소로 하는 배열																			
Out	○																			

기능

전송원 'In'의 값을 전송대상 배열 AryOut[]의 AryOut[0]에서 시작되는 'Size'개의 위치로 전송합니다.

'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

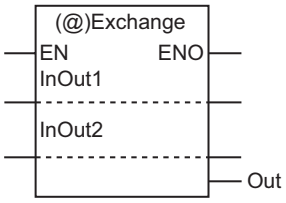


사용상의 주의 사항

- 'In'과 AryOut[]의 데이터 형식을 동일하게 하십시오. 동일하지 않으면 빌드 시에 이상 상태가 됩니다.
- 'In'과 AryOut[]이 STRING 형식인 경우, 각각의 영역 크기를 일치시키십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 AryOut[]의 배열 영역을 초과한 경우

Exchange

2개의 변수의 값을 교환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Exchange	데이터 교환	FUN		Exchange(InOut1, InOut2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut1, InOut2	교환 대상	입출력	교환 대상	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

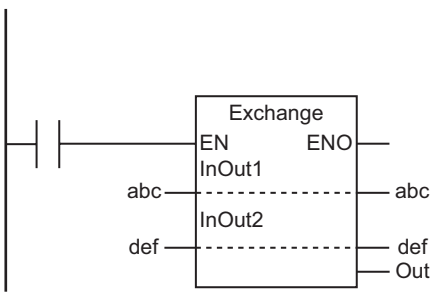
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
InOut2	'InOut1'과 같은 데이터 형식																			
Out	○																			

기능

교환 대상 'InOut1'과 'InOut2'의 값을 교환합니다.
'InOut1', 'InOut2'에는 열거 형식, 구조체 전체, 구조체의 1개 멤버를 지정할 수도 있습니다.

기술 예가 아래에 나와 있습니다. 변수 abc의 값과 변수 def의 값을 교환합니다.

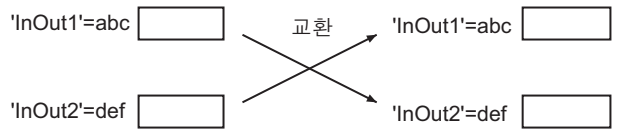
LD



ST

```
Exchange(abc, def);
```

'InOut1'과 'InOut2'의 값을 교환합니다.

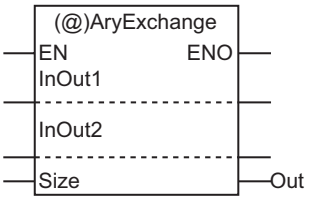


사용상의 주의 사항

- 'InOut1'과 'InOut2'의 데이터 형식을 동일하게 하십시오. 동일하지 않으면 빌드 시에 이상 상태가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'InOut1', 'InOut2'는 변화하지 않습니다.
  - 'InOut1', 'InOut2'가 STRING 형식이고, 어느 하나의 문자열 길이가 다른 하나의 크기에 수용되지 않는 경우

AryExchange

2개의 배열 간의 요소를 교환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryExchange	배열 데이터 교환	FUN		AryExchange(InOut1, InOut2, Size);

변수

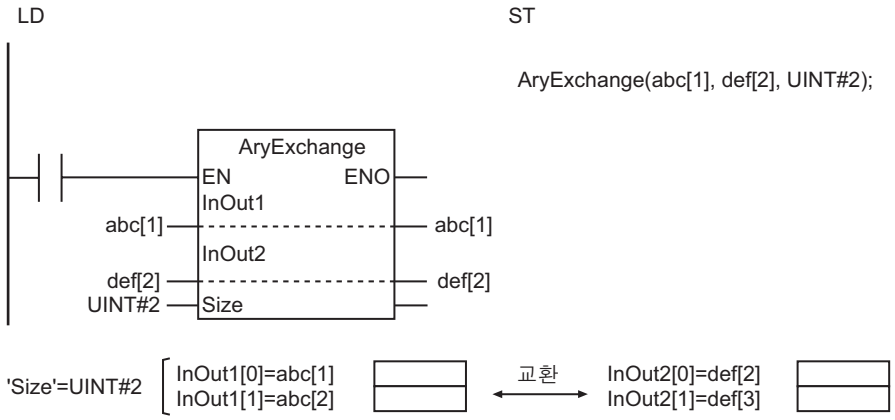
	명칭	입/출력	내용	유효 범위	단위	초기값
Size	교환 요소 수	입력	교환하는 요소 수	데이터 형식에 따름	–	1
InOut1[], InOut2[] 배열	교환 대상 배열	입출력	교환 대상 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size							○													
InOut1[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
InOut2[] 배열	InOut1[]과 동일한 데이터 형식의 배열																			
Out	○																			

기능

교환 대상 배열 InOut1[]의 InOut1[0]에서 시작되는 'Size'개의 요소와, InOut2[]의 InOut2[0]에서 시작되는 'Size'개의 요소를 교환합니다.

'Size'=UINT#2인 경우의 예가 아래에 나와 있습니다.



참고

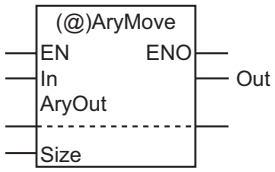
- 정수를 변수에 대입하는 경우, 🕮 'MOVE 명령(2-336페이지)'을 사용하십시오.
- 변수의 값을 다른 변수에 복사하는 경우, 🕮 'MemCopy 명령(2-345페이지)'을 사용하십시오.

사용상의 주의 사항

- InOut1[]과 InOut2[]의 요소의 데이터 형식을 동일하게 하십시오. 동일하지 않으면 빌드 시에 이상 상태가 됩니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 InOut1[]과 InOut2[]는 모두 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut1[]과 InOut2[]는 모두 변화하지 않습니다.
  - 'Size'의 값이 InOut1[] 또는 InOut2[]의 배열 영역을 초과한 경우
  - InOut1[], InOut2[]가 STRING 형식의 배열이며 문자열의 길이가 다른 하나의 크기를 초과하는 요소가 존재하는 경우
  - InOut1[], InOut2[]가 STRING 형식의 배열이며 NULL 문자로 종단되지 않은 요소가 존재하는 경우

AryMove

복수의 배열 요소를 전송합니다. 전송원과 전송대상의 데이터 형식이 달라도 상관없습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryMove	배열의 전송	FUN		AryMove(In, AryOut, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	전송 대상 배열	입력	전송하는 배열	데이터 형식에 따름	–	(*)
Size	전송 요소 수		전송하는 요소 수			1
AryOut[] 배열	전송 결과 배열	입출력	전송 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

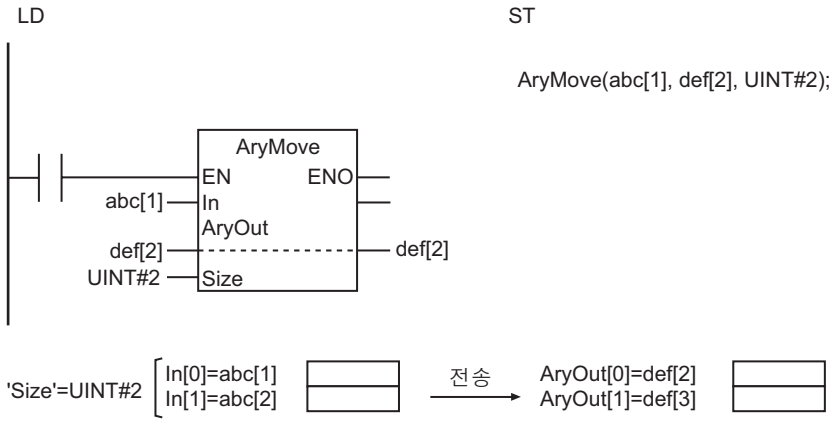
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Size							○													
AryOut[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Out	○																			

기능

전송대상 배열 In[]의 In[0]에서 시작되는 'Size'개의 요소를 전송 결과 배열인 AryOut[]의 AryOut[0]에서 시작되는 요소로 전송합니다.

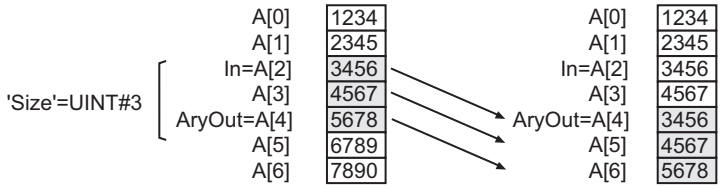
In[]과 AryOut[]의 데이터 형식이 다르더라도 상관없습니다.

'Size'=UINT#2인 경우의 예가 아래에 나와 있습니다.



참고

- In[]과 AryOut[]의 데이터 형식이 같은 경우, MemCopy 명령이 더 고속으로 처리할 수 있습니다.
- In[]과 AryOut[]에 같은 배열을 지정할 수도 있습니다. 또한, 그 경우 전송원 요소와 전송대상 요소가 겹쳐도 상관없습니다. In[0]=A[2], AryOut[0]=A[4], 'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

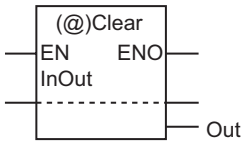


사용상의 주의 사항

- In[]과 AryOut[]의 데이터 형식이 다른 경우, 다음 중 하나의 데이터 형식 중에서 AryOut[]의 유효 범위가 In[]의 유효 범위를 포함하는 조합으로 하십시오.
  - BYTE, WORD, DWORD, LWORD
  - USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL
- In[]이 구조체를 요소로 하는 배열인 경우, In[]과 AryOut[]의 데이터 형식을 동일하게 하십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 AryOut[]은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Size'의 값이 In[] 또는 AryOut[]의 크기를 초과한 경우
  - In[], AryOut[]이 STRING 형식의 배열이며 전송되는 요소의 문자열 크기가 AryOut[]의 요소 크기를 초과하는 경우

Clear

변수를 초기화합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Clear	초기화	FUN		Clear(InOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	초기화 대상	입출력	초기화 대상	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Out	○																			

기능

초기화 대상 'InOut'의 값을 초기화합니다.
변수의 초기값 속성이 설정되어 있는 경우는 설정되어 있는 값으로 초기화됩니다. 초기값 속성이 설정되지 않은 경우는 'InOut'의 데이터 형식의 기본 초기값이 됩니다.
'InOut'이 외부 변수인 경우는 그에 대응하는 글로벌 변수의 초기값 속성과 관계 없이 'InOut' 데이터 형식의 기본 초기값이 됩니다.

각 데이터 형식 기본값의 초기값은 다음과 같습니다.

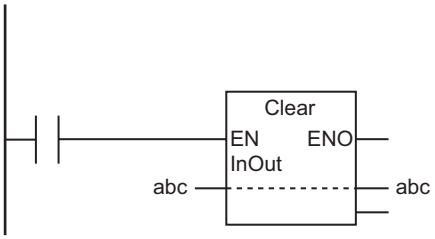
데이터 형식	기본 초기값
BOOL	FALSE
BYTE, WORD, DWORD, LWORD	16#0
USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL	0
TIME	T#0ms
DATE	D#1970-1-1
TOD	TOD#0:0:0
DT	DT#1970-1-1-0:0:0
STRING	''

'InOut'이 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버인 경우에는 다음과 같이 처리합니다.

'InOut'	처리 내용
배열 전체	배열의 모든 요소가 초기화됩니다.
배열의 1개 요소	그 요소만이 초기화됩니다.
구조체 전체	구조체의 모든 멤버가 초기화됩니다.
구조체의 1개 멤버	그 멤버만이 초기화됩니다.

기술 예가 아래에 나와 있습니다. 변수 abc의 값을 초기화합니다. 예를 들어 abc=INT#100인 경우, abc의 값은 INT#0이 됩니다.

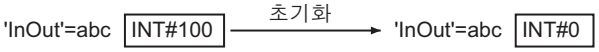
LD



ST

```
Clear(abc);
```

'InOut'의 값을 초기화합니다.
abc의 데이터 형식이 INT이므로 abc의 값은 INT#0이 됩니다.



참고

- 'InOut'이 스택으로 사용되는 배열의 경우, 이 명령을 실행할 때 스택의 저장 수를 관리하는 변수의 값도 0으로 하십시오.
- 캠 데이터 변수를 이 명령으로 초기화한 경우, 값은 MC_SaveCamTable 명령으로 저장된 것이 아닌, 0이 됩니다.

사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 열거 형식의 변수를 초기화하는 경우는 초기값 속성을 설정하십시오. 초기값 속성이 설정되지 않은 경우, 열거 형식 변수의 값은 0이 됩니다.
- 다음의 모든 조건에 해당하는 처리는 실시하지 마십시오. 정확하게 동작하지 않는 경우가 있습니다.
  - FUN 또는 FB에 입출력 변수로 BOOL 형식 배열의 1개 요소를 전달한다.
  - 상기의 FUN 또는 FB 내에서 Clear 명령을 실행한다.
  - Clear 명령에 전달하는 파라메터는 상기 BOOL 형식 배열의 1개 요소를 받은 입출력 변수이다.

Copy**ToNum(비트 열→부호가 있는 정수)

비트 열의 내용을 그대로 부호가 있는 정수로 복사합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Copy**ToNum	비트 패턴 복사(비트 열→부호가 있는 정수) 그룹	FUN	(@)Copy**ToNum EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out:=Copy**ToNum(In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out	'In'의 데이터 형식과 동일한 크기의, 부호가 있는 정수의 데이터 형식																			

기능

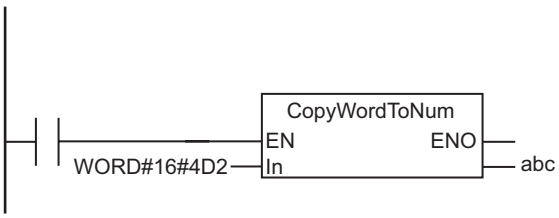
복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 4종류가 있습니다.

'In'	'Out'	명령 이름
BYTE	SINT	CopyByteToNum
WORD	INT	CopyWordToNum
DWORD	DINT	CopyDwordToNum
LWORD	LINT	CopyLwordToNum

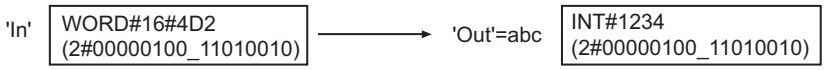
CopyWordToNum 명령에서 'In'=WORD#16#4D2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=CopyWordToNum(WORD#16#4D2);
```



Copy**To***(비트 열→실수)

비트 열의 내용을 그대로 부호가 있는 실수로 복사합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Copy**To***	비트 패턴 복사(비트열→실수) 그룹	FUN	(@)CopyDwordToReal EN ENO In Out (@)CopyLwordToLreal EN ENO In Out	Out:=CopyDwordToReal(In); 또는 Out:=CopyLwordToLreal(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In				○	○															
Out	'In'의 데이터 형식이 DWORD라면 REAL, LWORD라면 LREAL																			

기능

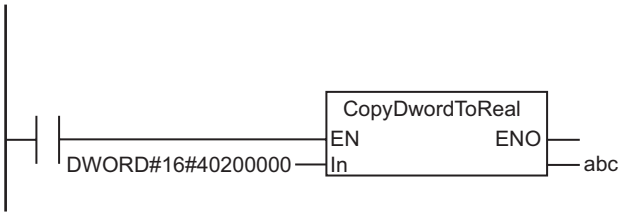
복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

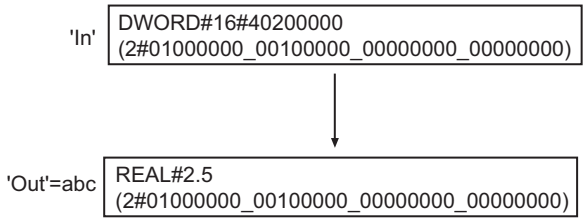
명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 2종류가 있습니다.

'In'	'Out'	명령 이름
DWORD	REAL	CopyDwordToReal
LWORD	LREAL	CopyLwordToLreal

CopyDwordToReal 명령에서 'In'=DWORD#16#40200000인 경우의 예가 아래에 나와 있습니다.

LD





ST

```
abc:=CopyDwordToReal(DWORD#16#40200000);
```

CopyNumTo**(부호가 있는 정수→비트 열)

부호가 있는 정수의 내용을 그대로 비트 열로 복사합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CopyNumTo**	비트 패턴 복사(부호가 있는 정수→비트 열) 그룹	FUN	(@)CopyNumTo** EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out:=CopyNumTo**(In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In										○	○	○	○							
Out	'In'의 데이터 형식과 동일한 크기의, 비트 열의 데이터 형식																			

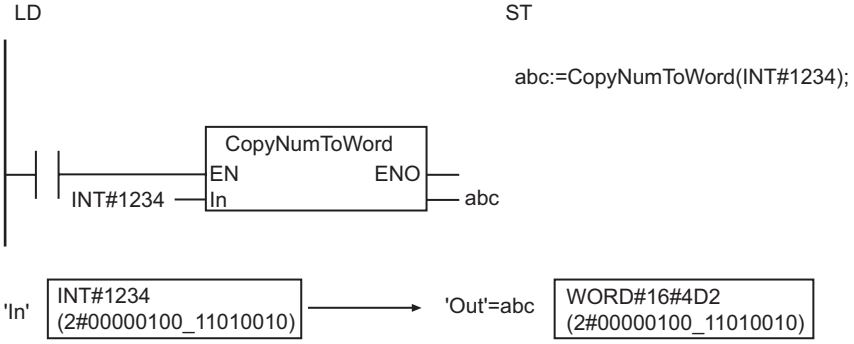
기능

복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 4종류가 있습니다.

'In'	'Out'	명령 이름
SINT	BYTE	CopyNumToByte
INT	WORD	CopyNumToWord
DINT	DWORD	CopyNumToDword
LINT	LWORD	CopyNumToLword

CopyNumToWord 명령에서 'In'=INT#1234인 경우의 예가 아래에 나와 있습니다.



CopyNumTo**(부호가 있는 정수→실수)

부호가 있는 정수의 내용을 그대로 실수로 복사합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CopyNumTo**	비트 패턴 복사(부호가 있는 정수→실수) 그룹	FUN	(@)CopyNumToReal EN ENO In Out (@)CopyNumToLreal EN ENO In Out	Out:=CopyNumToReal(In); 또는 Out:=CopyNumToLreal(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In												○	○							
Out	'In'의 데이터 형식이 DINT라면 REAL, LINT라면 LREAL																			

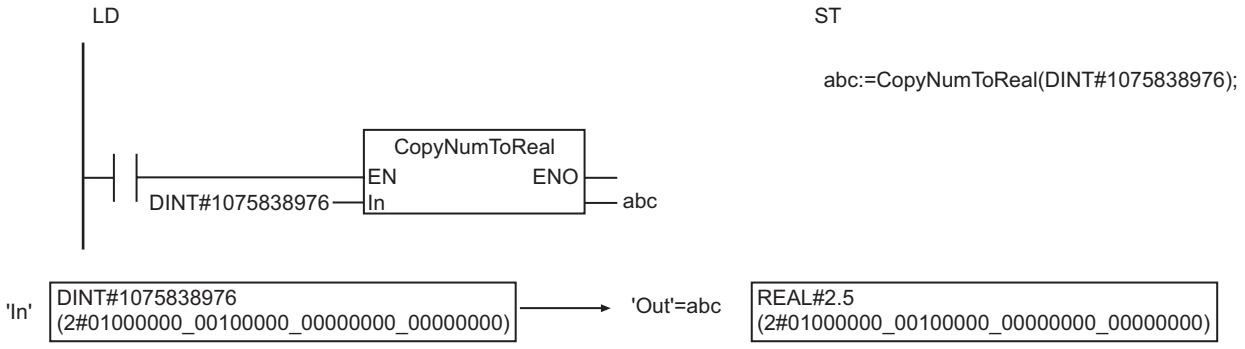
기능

복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 2종류가 있습니다.

'In'	'Out'	명령 이름
DINT	REAL	CopyNumToReal
LINT	LREAL	CopyNumToLreal

CopyNumToReal 명령에서 'In'=DINT#1075838976인 경우의 예가 아래에 나와 있습니다.



Copy**To***(실수→비트 열)

실수의 내용을 그대로 비트 열로 복사합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Copy**To***	비트 패턴 복사(실수→비트 열) 그룹	FUN	(@)CopyRealToDword EN ENO In Out (@)CopyLrealToLword EN ENO In Out	Out:=CopyRealToDword(In); 또는 Out:=CopyLrealToLword(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0.0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out	'In'의 데이터 형식이 REAL이라면 DWORD, LREAL이라면 LWORD																			

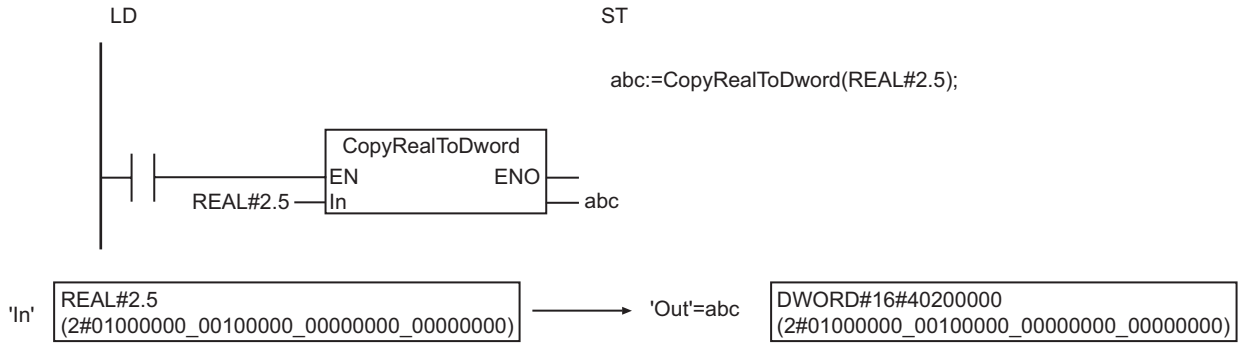
기능

복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 2종류가 있습니다.

'In'	'Out'	명령 이름
REAL	DWORD	CopyRealToDword
LREAL	LWORD	CopyLrealToLword

CopyRealToDword 명령에서 'In'=REAL#2.5인 경우의 예가 아래에 나와 있습니다.



Copy**ToNum(실수→부호가 있는 정수)

실수의 내용을 그대로 부호가 있는 정수로 복사합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Copy**To Num	비트 패턴 복사 (부호가 있는 정수→실수) 그룹	FUN	(@)CopyRealToNum EN ENO In Out (@)CopyLrealToNum EN ENO In Out	Out:=CopyRealToNum(In); 또는 Out:=CopyLrealToNum(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	복사원	입력	복사원	데이터 형식에 따름	–	0.0
Out	복사대상	출력	복사대상	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out	'In'의 데이터 형식이 REAL이라면 DINT, LREAL이라면 LINT																			

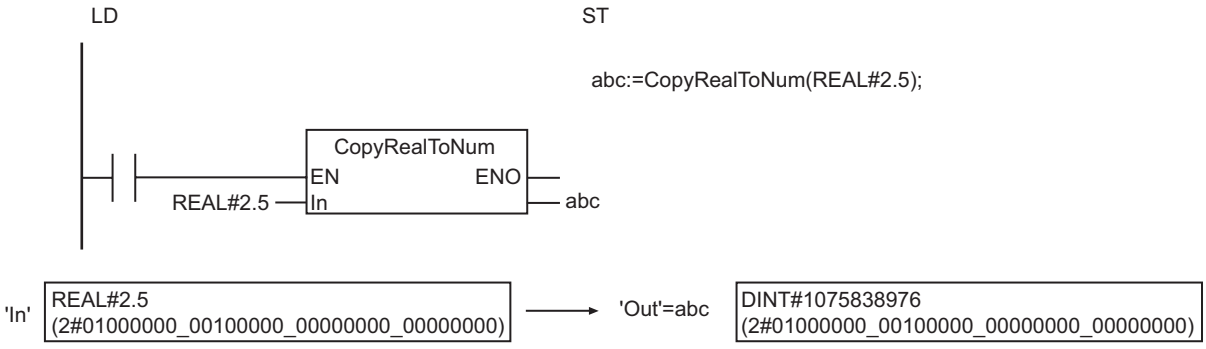
기능

복사원 'In'의 내용을 가공하지 않고 그대로 복사대상 'Out'으로 복사합니다.

명령 이름은 'In'과 'Out'의 데이터 형식의 조합에 따라 2종류가 있습니다.

'In'	'Out'	명령 이름
REAL	DINT	CopyRealToNum
LREAL	LINT	CopyLrealToNum

CopyRealToNum 명령에서 'In'=REAL#2. 5인 경우의 예가 아래에 나와 있습니다.



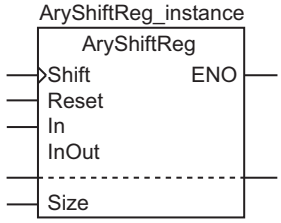
'In' REAL#2.5 (2#01000000_00100000_00000000_00000000) → 'Out'=abc DINT#1075838976 (2#01000000_00100000_00000000_00000000)

이동 명령

명령	명칭	페이지
AryShiftReg	이동 레지스터	2-370
AryShiftRegLR	좌우 이동 레지스터	2-372
ArySHL/ArySHR	배열의 N요소 좌측 이동/ 배열의 N요소 우측 이동	2-375
SHL/SHR	N비트 좌측 이동/ N비트 우측 이동	2-378
NSHLC/NSHRC	N비트 데이터 N비트 좌측 이동 CY 있음/ N비트 데이터 N비트 우측 이동 CY 있음	2-380
ROL/ROR	N비트 좌회전 그룹/ N비트 우회전 그룹	2-382

AryShiftReg

배열 요소로 이루어진 비트 열 전체를 1비트 좌측으로 이동하여 최하위 비트에 입력값을 삽입합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryShiftReg	이동 레지스터	FB		AryShiftReg_instance(Shift, Reset, In, InOut, Size);

변수

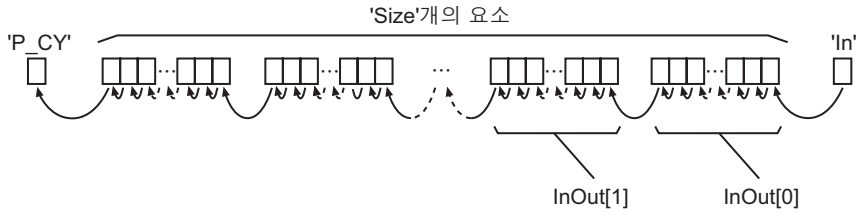
	명칭	입/출력	내용	유효 범위	단위	초기값
Shift	이동	입력	FALSE→TRUE 시 이동 실행	데이터 형식에 따름	–	FALSE
Reset	리셋		TRUE: 리셋 실행			
In	입력값		InOut[]의 최하위 비트에 삽입하는 값			
Size	비트열배열의 요소 수		InOut[] 내의, 이동 레지스터에 사용하는 요소 수			1
InOut[] 배열	비트 열 배열	입출력	비트 열 배열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Shift	○																			
Reset	○																			
In	○																			
Size							○													
InOut[] 배열	○	○	○	○	○															

기능

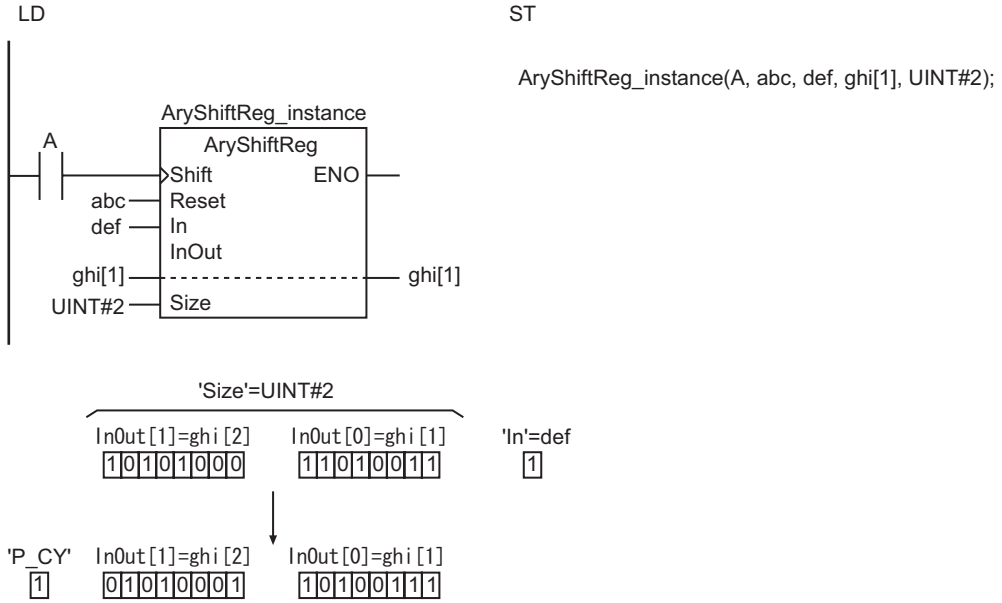
'Shift'가 FALSE→TRUE일 때, 비트 열 배열 InOut[]의 InOut[0]에서 시작되는 'Size'개의 배열 요소를 좌측 방향(상위 방향)으로 1비트 이동합니다.

최하위 비트에는 입력값 'In'을 삽입합니다. 비트 열 배열에서 초과된 최상위 비트는 캐리(CY) 플래그 'P_CY'에 출력합니다.



'Reset'이 TRUE일 때, InOut[0]에서 시작되는 'Size'개의 모든 요소의 비트와 캐리(CY) 플래그에 FALSE를 설정합니다.

InOut[]이 BYTE 형식 배열, 'Size'=UINT#2인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

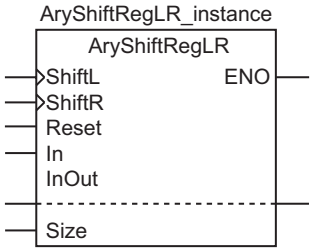
변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	캐리 플래그에 저장되어 있는 값

사용상의 주의 사항

- 'Reset'이 TRUE일 때는 'Shift'가 FALSE→TRUE가 되더라도 이동은 실행되지 않습니다.
- ENO가 TRUE가 되는 것은 'Shift'가 FALSE→TRUE가 되어 이동 동작이 정상적으로 수행되었을 때나, 'Reset'이 TRUE가 되어 리셋 동작이 정상적으로 수행되고 있을 때입니다.
- 'Size'의 값이 0인 경우, InOut[]의 값은 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

AryShiftRegLR

배열 요소로 이루어진 비트 열 전체를 좌측(또는 우측) 방향으로 1비트 이동하여 최하위(또는 최상위) 비트에 입력값을 삽입합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AryShiftRegLR	좌우 이동 레지스터	FB		AryShiftRegLR_instance(ShiftL, ShiftR, Reset, In, InOut, Size);

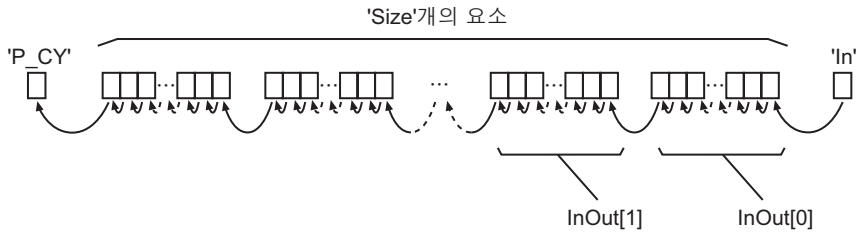
변수

	명칭	입/출력	내용	유효 범위	단위	초기값
ShiftL	좌측 이동	입력	FALSE→TRUE 시 좌측 이동 실행	데이터 형식에 따름	–	FALSE
ShiftR	우측 이동		FALSE→TRUE 시 우측 이동 실행			
Reset	리셋		TRUE: 리셋 실행			
In	입력값		InOut[]의 최하위 비트 또는 최상위 비트에 삽입하는 값			
Size	비트 열 배열의 요소 수		InOut[] 내의, 이동 레지스터에 사용하는 요소 수			1
InOut[] 배열	비트 열 배열	입출력	비트 열 배열	데이터 형식에 따름	–	–

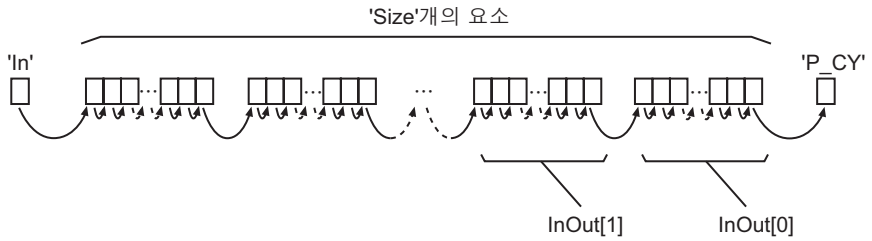
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
ShiftL	○																			
ShiftR	○																			
Reset	○																			
In	○																			
Size							○													
InOut[] 배열	○	○	○	○	○															

기능

'Shift'가 FALSE→TRUE일 때, 비트 열 배열 InOut[]의 InOut[0]에서부터 'Size'개의 배열 요소를 좌측 방향으로 1비트 이동합니다. 최하위 비트에는 입력값 'In'을 삽입합니다. 비트 열 배열에서 초과된 최상위 비트는 캐리(CY) 플래그 'P_CY'에 출력합니다.

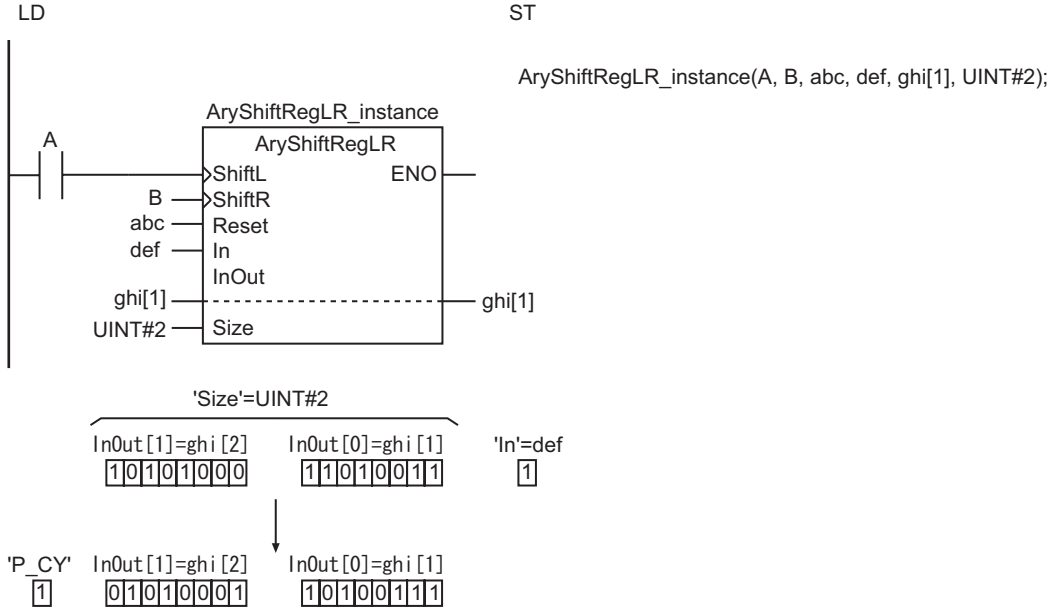


'ShiftR'이 FALSE→TRUE일 때는 우측 방향으로 1비트 이동해서 최상위 비트에 'In'을 삽입합니다. 비트 열 배열에서 초과된 최하위 비트를 캐리(CY) 플래그 'P_CY'에 출력합니다.



'Reset'이 TRUE일 때, InOut[0] 이후의 'Size'개의 모든 요소의 비트와 'P_CY'에 FALSE를 설정합니다.

'InOut'이 BYTE 형식 배열, 'Size'=UINT#2이며 'Shift'가 FALSE→TRUE인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

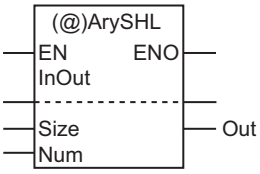
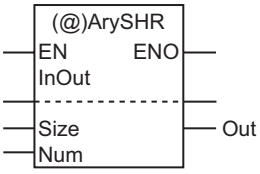
변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	캐리 플래그에 저장되어 있는 값

사용상의 주의 사항

- 'Reset'이 TRUE일 때는 'ShiftL' 및 'ShiftR'이 FALSE→TRUE가 되더라도 이동은 실행되지 않습니다.
- 'ShiftL'과 'ShiftR'이 동시에 FALSE→TRUE가 된 경우, 이동은 실행되지 않습니다.
- ENO가 TRUE가 되는 것은 'Shift'가 FALSE→TRUE가 되어 이동 동작이 정상적으로 수행되었을 때나, 'Reset'이 TRUE가 되어 리셋 동작이 정상적으로 수행되고 있을 때입니다.
- 'Size'의 값이 0인 경우, InOut[]의 값은 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

ArySHL / ArySHR

배열 요소를 복수 요소만큼 이동합니다.
ArySHL : 좌측 방향(상위 방향)으로 이동합니다.
ArySHR : 우측 방향(하위 방향)으로 이동합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ArySHL	배열의 N요소 좌측 이동	FUN		ArySHL(InOut, Size, Num);
ArySHR	배열의 N요소 우측 이동	FUN		ArySHR(InOut, Size, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Size	이동 대상의 요소 수	입력	이동 대상의 요소 수	데이터 형식에 따름	–	1
Num	이동 요소 수		이동시키는 요소 수			
InOut[] 배열	이동 대상 배열	입출력	이동 대상 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size							○													
Num							○													
InOut[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	구조체를 요소로 하는 배열도 지정 가능																			
Out	○																			

기능

이동 대상 배열 InOut[]의 상위 'size'개의 요소에 대해 이동 요소 수 'Num'개만큼의 이동을 수행합니다. 이동으로 인해 초과된 값은 폐기합니다.
또한 빈 요소에는 InOut[]의 데이터 형식의 초기값을 저장합니다. InOut[]에 초기값 속성이 설정되어 있는 경우, 그 값으로 초기화됩니다. 초기값 속성이 설정되어 있지 않은 경우에는 각 데이터 형식의 기본값의 초기값이 됩니다. InOut[]이 구조체를 요소로 하는 배열인 경우, 각 요소의 구조체의 모든 멤버가 초기화됩니다.

각 데이터 형식 기본값의 초기값은 다음과 같습니다.

데이터 형식	초기값
BOOL	FALSE
BYTE, WORD, DWORD, LWORD	16#0
USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL	0
TIME	T#0ms
DATE	D#1970-1-1
TOD	TOD#0:0:0
DT	DT#1970-1-1-0:0:0
STRING	''

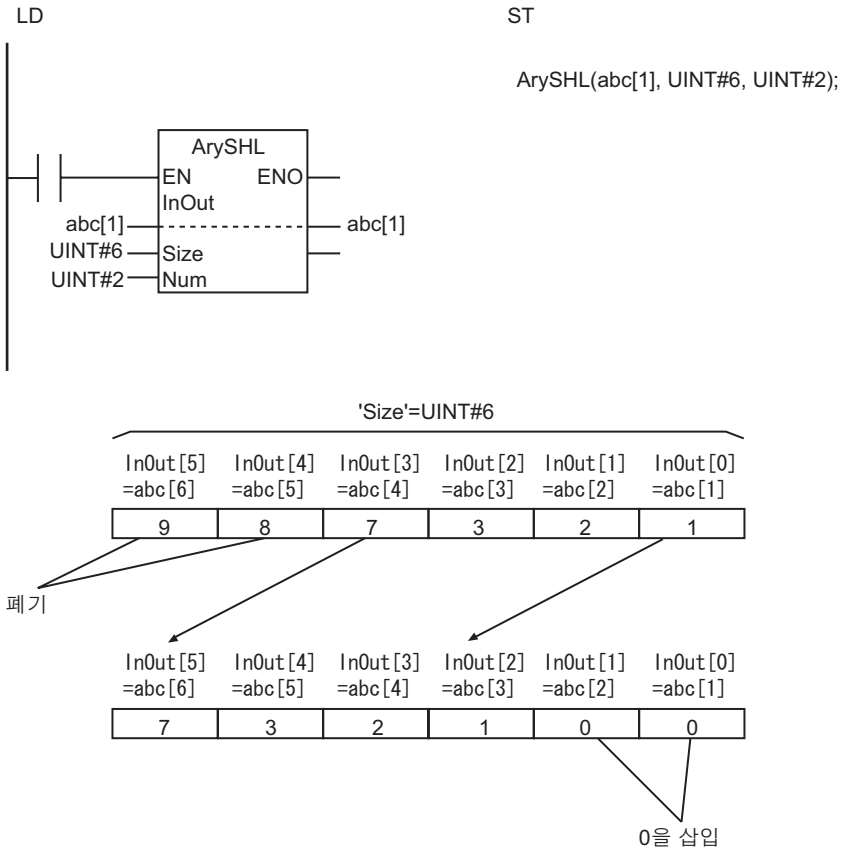
● ArySHL

좌측 방향(배열의 상위 방향)으로 이동합니다.

● ArySHR

우측 방향(배열의 하위 방향)으로 이동합니다.

ArySHL 명령에서 'Size'=UINT#6, 'Num'=UINT#2인 경우의 예가 아래에 나와 있습니다.



참고

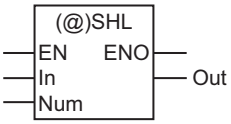
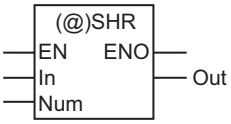
InOut[]이 BOOL 형식인 경우, 'Size' 비트의 비트 열을 'Num' 비트만큼 이동하는 것과 같습니다.

사용상의 주의 사항

- 'Num'의 값이 0인 경우, 이동 동작은 수행되지 않습니다.
- 'Num'의 값이 'Size'보다 큰 경우, InOut[0]에서 InOut['Size'-1]의 모든 값이 초기화됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

SHL / SHR

비트 열을 복수 비트 이동합니다.
SHL : 좌측 방향(상위 방향)으로 이동합니다.
SHR : 우측 방향(하위 방향)으로 이동합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SHL	N비트 좌측 이동	FUN		Out:=SHL(In, Num);
SHR	N비트 우측 이동	FUN		Out:=SHR(In, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	이동 대상	입력	이동 대상	데이터 형식에 따름	–	*1
Num*2	이동 수		이동시키는 비트 수	0~'In'의 비트 수	비트	1
Out	처리 결과	출력	처리 결과	데이터 형식에 따름	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Num을 N으로 기술할 수 있습니다. 예를 들어 Out:=SHL(In:=BYTE#16#89, N:=ULINT#2);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Num						○ *1			○ *1											
Out	'In'과 같은 데이터 형식																			

*1 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서는 ULINT 형식입니다. CPU 유니트 Ver.1.01 이하 또는 Sysmac Studio Ver.1.02 이하에서는 USINT 형식입니다.

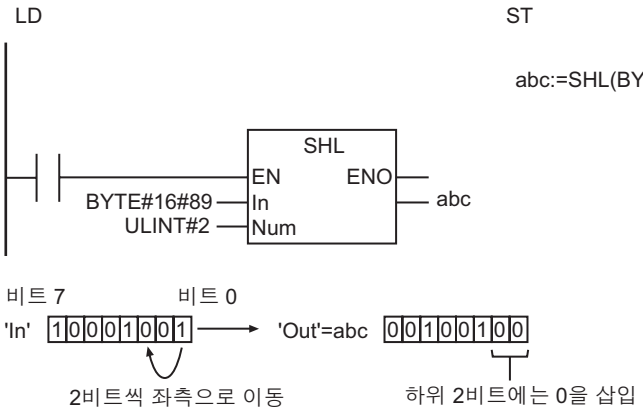
기능

비트 열 형식의 이동 대상 'In'을 이동 수 'Num'의 비트 수만큼 이동합니다. 초과한 비트는 폐기되며, 반대측에서 삽입되는 비트에는 0이 들어갑니다.

● SHL

우측에서 좌측(하위 비트에서 상위 비트) 방향으로 이동합니다.

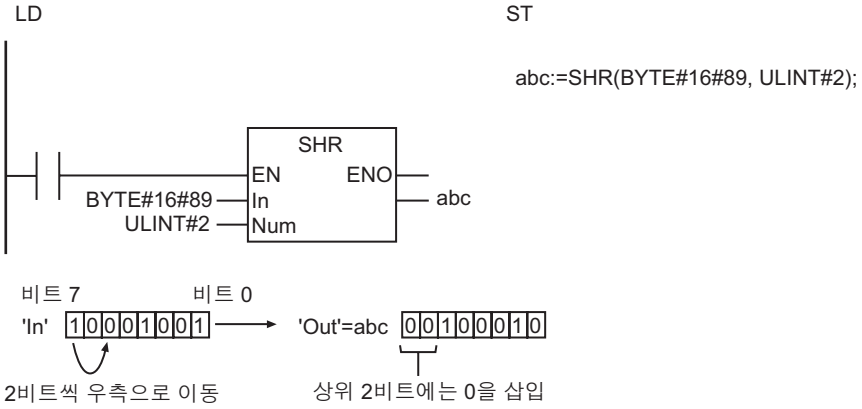
'In'=BYTE#16#89, 'Num'=ULINT#2인 경우의 예가 아래에 나와 있습니다.



● SHR

좌측에서 우측(상위 비트에서 하위 비트) 방향으로 이동합니다.

'In'=BYTE#16#89, 'Num'=ULINT#2인 경우의 예가 아래에 나와 있습니다.



참고

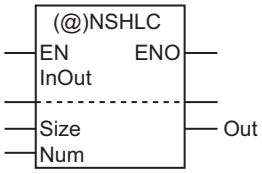
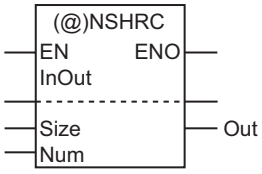
ROL/ROR 명령은, 초과한 비트의 값이 반대측에서 삽입됩니다.

사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식을 동일하게 하십시오.
- 'Num'의 값이 0인 경우, 이상 상태가 되지는 않으며 'In'의 값이 그대로 'Out'에 대입됩니다.
- 'Num'의 값이 'In'의 비트 수보다 큰 경우, 이상 상태가 되지는 않으며 'Out'의 값은 16#0이 됩니다.

NSHLC / NSHRC

비트 열 배열을 캐리(CY) 플래그 있음으로 복수 비트 이동합니다.
NSHLC : 좌측 방향(상위 방향)으로 이동합니다.
NSHRC : 우측 방향(하위 방향)으로 이동합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NSHLC	N비트 데이터 N비트 좌측 이동 CY 있음	FUN		NSHLC(InOut, Size, Num);
NSHRC	N비트 데이터 N비트 우측 이동 CY 있음	FUN		NSHRC(InOut, Size, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Size	이동 대상의 비트 수	입력	이동 대상의 비트 수	데이터 형식에 따름	비트	1
Num	이동 비트 수		이동시키는 비트 수			
InOut[] 배열	이동 대상 배열	입출력	이동 대상의 비트 열 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size						○														
Num						○														
InOut[] 배열	○	○	○	○	○															
Out	○																			

기능

이동 대상 배열 InOut[]의 InOut[0]부터 'Size'개의 배열 요소에 대해, 'Num' 비트만큼 이동합니다. 초과한 비트의 마지막 값은 캐리(CY) 플래그에 출력합니다. 반대측에서 삽입되는 비트에는 0이 들어갑니다.

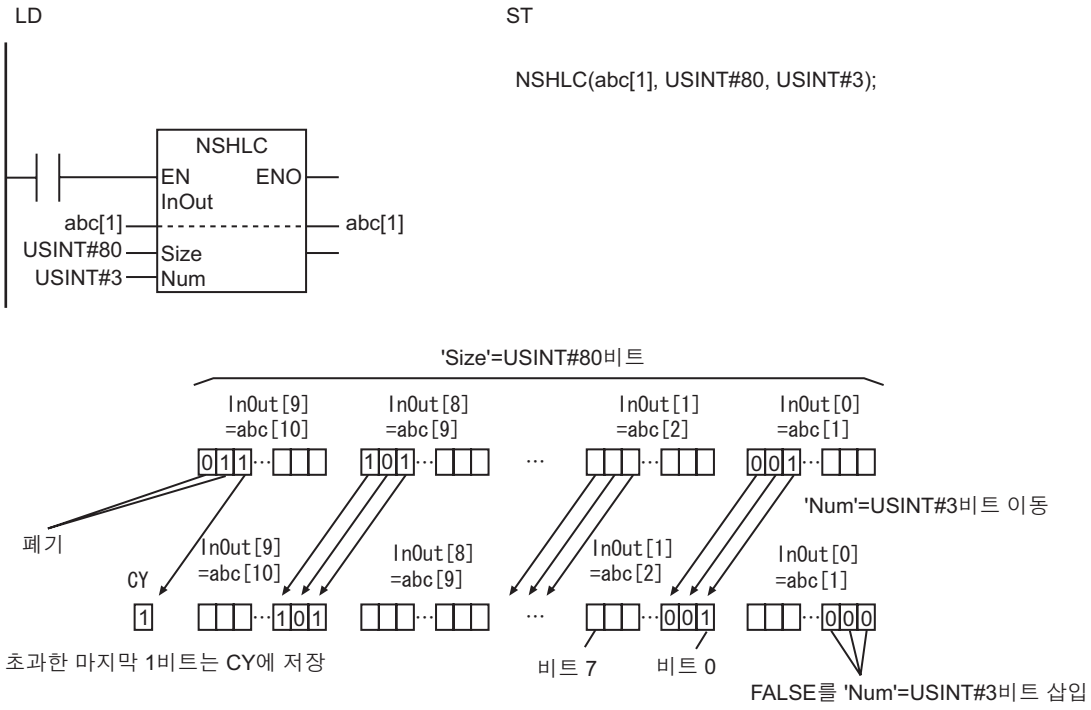
● NSHLC

좌측 방향(배열의 하위에서 상위 및 하위 비트에서 상위 비트)으로 이동합니다.

● NSHRC

우측 방향(배열의 상위에서 하위 및 상위 비트에서 하위 비트)으로 이동합니다.

NSHLC 명령에서 InOut[]이 BYTE 형식 배열, 'Size'=USINT#80, 'Num'=USINT#3인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

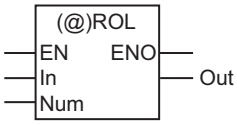
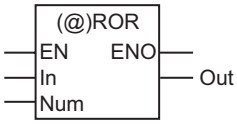
변수명	명칭	데이터 형식	내용
P_CY	캐리(CY) 플래그	BOOL	캐리 플래그에 저장되어 있는 값

사용상의 주의 사항

- 'Num'의 값이 0인 경우, 이동 동작은 수행되지 않습니다.
- 'Num'의 값이 'Size'보다 큰 경우, InOut[0]의 비트 0에서'Size' 비트만큼의 값이 FALSE가 됩니다. 또한, 캐리(CY) 플래그의 값도 FALSE가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

ROL / ROR

비트 열을 복수 비트 회전합니다.
ROL : 좌측 방향(상위 방향)으로 회전합니다.
ROR : 우측 방향(하위 방향)으로 회전합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ROL	N비트 좌회전	FUN		Out:=ROL(In, Num);
ROR	N비트 우회전	FUN		Out:=ROR(In, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	회전 대상	입력	회전 대상	데이터 형식에 따름	–	*1
Num*2	회전 수		회전시키는 비트 수	0~'In'의 비트 수	비트	1
Out	처리 결과	출력	처리 결과	데이터 형식에 따름	–	–

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.
*2 Sysmac Studio Ver.1.03 이상에서 변수명과 파라메터명의 대응을 명시하는 ST 표현을 하는 경우, Num을 N으로 기술할 수 있습니다. 예를 들어 Out:=ROL(In:=BYTE#16#89, N:=ULINT#2);로 기술할 수 있습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Num						○ *1			○ *1											
Out	'In'과 같은 데이터 형식																			

*1 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서는 ULINT 형식입니다. CPU 유니트 Ver.1.01 이하 또는 Sysmac Studio Ver.1.02 이하에서는 USINT 형식입니다.

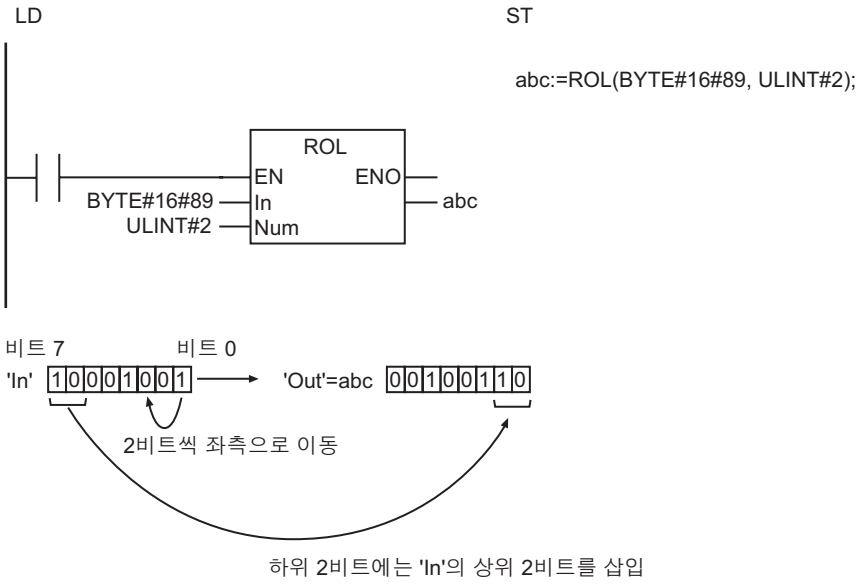
기능

비트 열 형식의 회전 대상 'In'을 회전 수 'Num'의 비트 수만큼 회전합니다. 초과한 비트는 반대측에서 삽입합니다.

● ROL

우측에서 좌측(하위 비트에서 상위 비트) 방향으로 회전합니다.

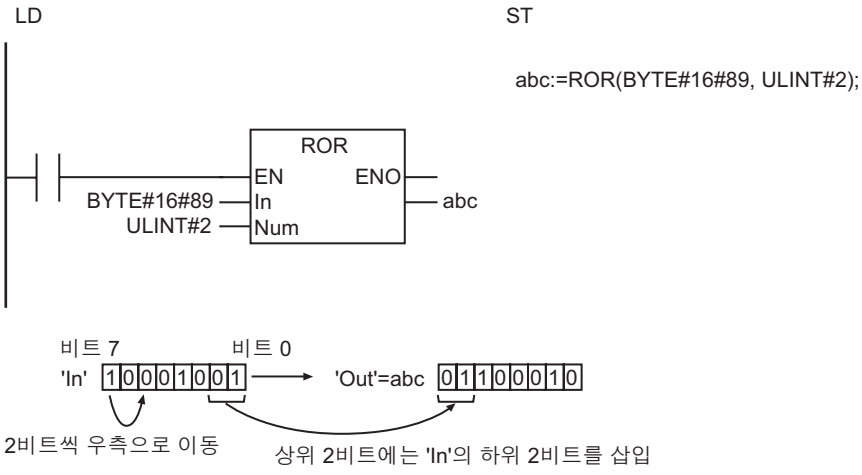
'In'=BYTE#16#89, 'Num'=ULINT#2인 경우의 예가 아래에 나와 있습니다.



● ROR

좌측에서 우측(상위 비트에서 하위 비트) 방향으로 회전합니다.

'In'=BYTE#16#89, 'Num'=ULINT#2인 경우의 예가 아래에 나와 있습니다.



참고

SHL/SHR 명령은, 초과한 비트를 폐기하고 반대측에서 삽입되는 비트에 0을 넣습니다.

사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식을 동일하게 하십시오.
- 'Num'의 값이 0인 경우, 이상 상태가 되지는 않으며 'In'의 값이 그대로 'Out'에 대입됩니다.
- 'Num'의 값이 'In'의 비트 수를 초과하는 경우, 이상 상태가 되지는 않으며 'Num'의 값만큼 회전합니다. 예를 들면, 'In'이 WORD 형식인 경우, 'Num'의 값이 USINT#1인 경우와 USINT#17인 경우 'Out'의 값은 같습니다.

데이터 변환 명령

명령	명칭	페이지	명령	명칭	페이지
Swap	바이트 스왑	2-386	HexStringToNum_**	16진수 문자열→수치 변환 그룹	2-430
Neg	부호 반전	2-387	FixNumToString	고정 소수점 수→문자열 변환	2-432
Decoder	비트 디코더	2-389	StringToFixNum	문자열→고정 소수점 수 변환	2-434
Encoder	비트 엔코더	2-392	DtToString	일시→문자열 변환	2-437
BitCnt	비트 카운터	2-394	DateToString	날짜→문자열 변환	2-439
ColmToLine_**	비트 열→비트 행 변환 그룹	2-395	TodToString	시각→문자열 변환	2-440
LineToColm	비트 행→비트 열 변환	2-397	GrayToBin_**/ BinToGray_**	그레이 코드→BIN 코드 변환 그룹/ BIN 코드→그레이 코드 변환 그룹	2-442
Gray	그레이 코드 변환	2-399	StringToAry	문자열→배열 변환	2-445
UTF8ToSJIS	문자 코드 변환 (UTF-8→SJIS)	2-404	AryToString	배열→문자열 변환	2-447
SJISToUTF8	문자 코드 변환 (SJIS→UTF-8)	2-406	DispartDigit	4비트 분리	2-449
PWLApprox/ PWLApproxNoLine Chk	꺾은선 근사 변환(꺾은선 데이터 확인 있음)/꺾은선 근사 변환(꺾은선 데이터 확인 없음)	2-408	UniteDigit_**	4비트 결합 그룹	2-451
PWLLineChk	꺾은선 데이터 확인	2-413	Dispart8Bit	바이트 단위 데이터 분리	2-453
MovingAverage	이동 평균	2-416	Unite8Bit_**	바이트 단위 데이터 결합 그룹	2-455
DispartReal	실수의 가수 · 지수 분리	2-422	ToAryByte	바이트 배열로 변환	2-457
UniteReal	가수 · 지수를 실수로 결합	2-425	AryByteTo	바이트 배열에서 변환	2-462
NumToDecString/ NumToHexString	고정 길이 10진수 문자열 변환/ 고정 길이 16진수 문자열 변환	2-427	SizeOfAry	배열 요소 수 취득	2-467

Swap

16비트 값의 상위 바이트와 하위 바이트를 교환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Swap	바이트 스왑	FUN	(@)Swap EN ENO In Out	Out:=Swap(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

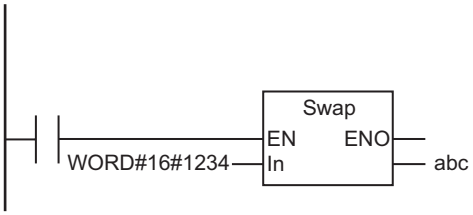
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In			○																	
Out			○																	

기능

변환 대상 'In'의 상위 바이트와 하위 바이트를 교환하여 변환 결과 'Out'에 대입합니다.

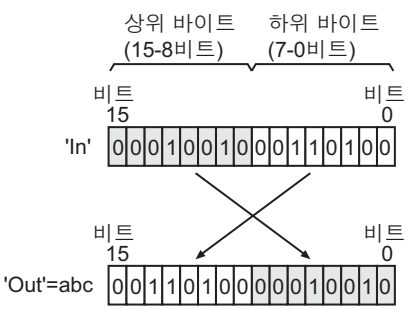
'In'=WORD#16#1234인 경우의 예가 아래에 나와 있습니다.

LD



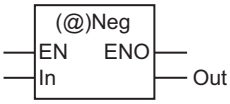
ST

```
abc:=Swap(WORD#16#1234);
```



Neg

수치의 부호를 반전시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Neg	부호 반전	FUN		Out:=Neg(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	(*)
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

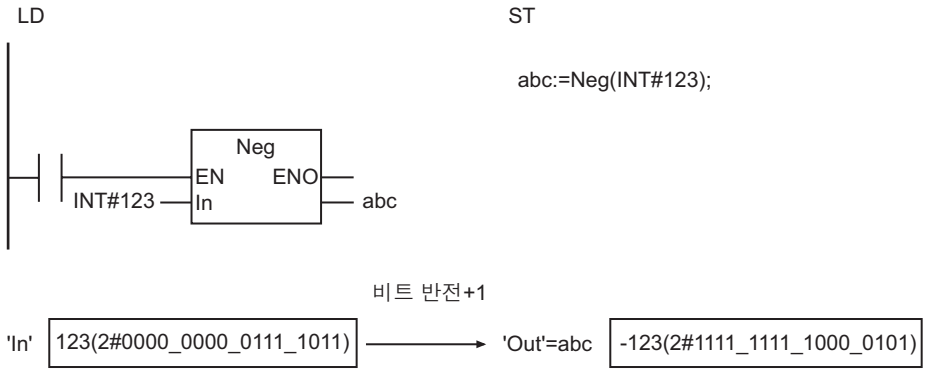
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○	○	○					
Out						○	○	○	○	○	○	○	○	○	○					

기능

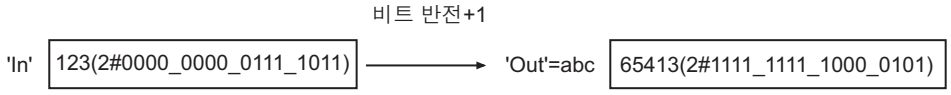
변환 대상 'In'의 부호를 반전시킵니다.
처리 내용은 'In'의 데이터 형식에 따라 다음과 같이 달라집니다.

'In'의 데이터 형식	'Out'의 값
부호가 있는 정수 SINT, INT, DINT, LINT	'In'의 모든 비트를 반전시켜 1을 가산 ('In' x (-1)과 같은 의미)
부호가 없는 정수 USINT, UINT, UDINT, ULINT	'In'의 모든 비트를 반전시켜 1을 가산
실수 REAL, LREAL	'In' x (-1)

'In'=INT#123인 경우의 예가 아래에 나와 있습니다.

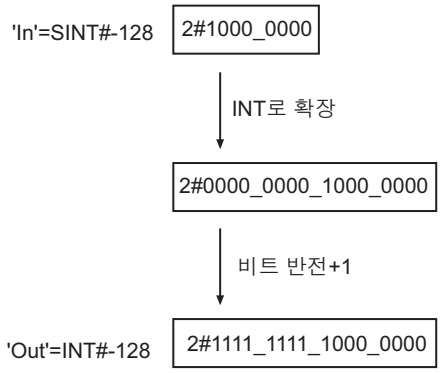


'In'=UINT#123인 경우의 예가 아래에 나와 있습니다.



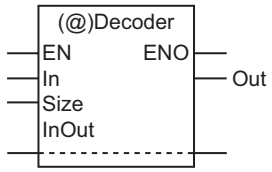
사용상의 주의 사항

'In'과 'Out'의 데이터 형식이 다른 경우에는 'Out'의 유효 범위에 'In'의 유효 범위가 포함되도록 하십시오. 그렇지 않은 경우 이상 상태가 되지 않으며, 'Out'의 값은 잘못된 값이 됩니다.
예를 들면, 'In'의 값이 SINT#-128, 'Out'의 데이터 형식이 INT인 경우, 'Out'의 값은 INT#128이 되지 않고 INT#-128이 됩니다.



Decoder

최대 256비트로 이루어진 배열 요소의 지정된 1비트를 TRUE, 기타 비트를 FALSE로 설정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Decoder	비트 디코더	FUN		Decoder(In, Size, InOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 비트 위치	입력	변환할 비트 위치	데이터 형식에 따름	–	0
Size	변환 비트 수		변환할 비트 수	0~8	비트	1
InOut[] 배열	변환 대상 배열	입출력	변환 대상 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

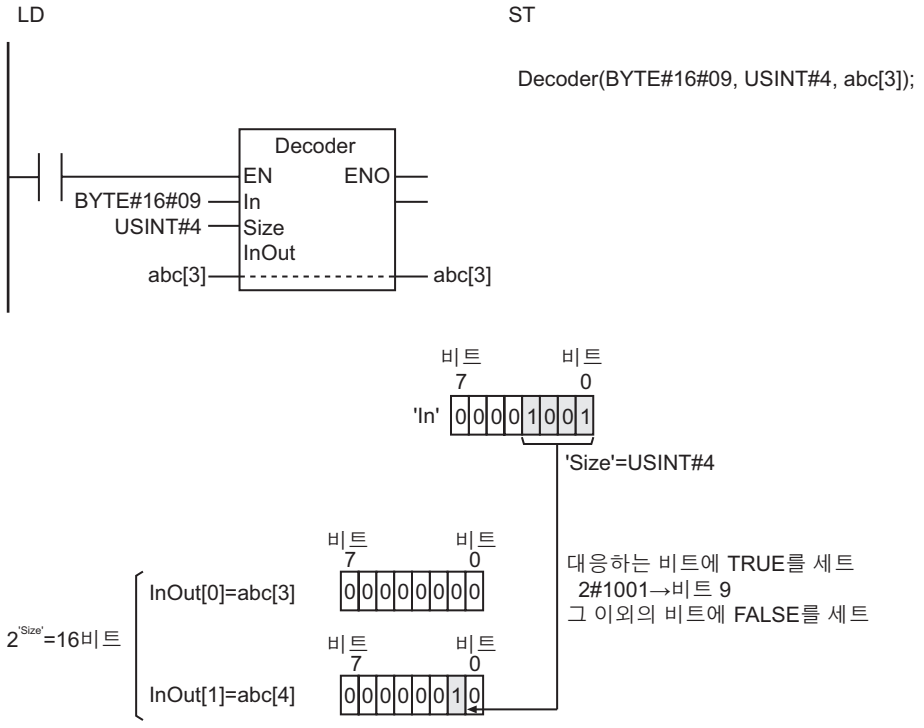
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○																		
Size						○														
InOut[] 배열	○	○	○	○	○															
Out	○																			

기능

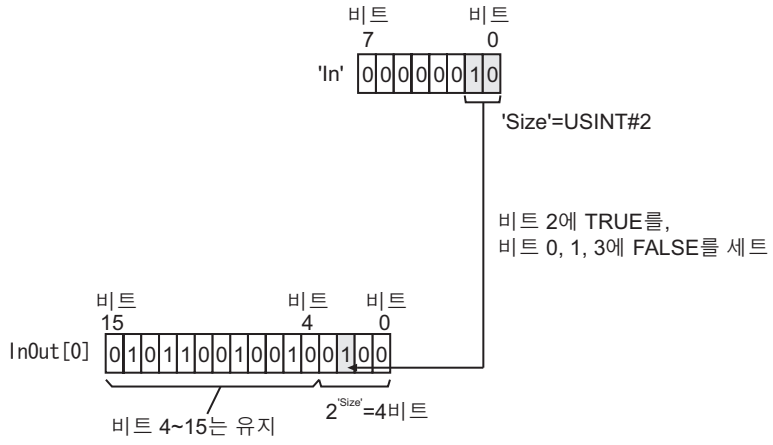
변환 대상 배열 InOut[]의 InOut[0]에서 시작되는 $2^{'Size'}$비트 크기의 배열 요소의 지정 비트를 TRUE로 합니다. 그 이외의 비트는 FALSE로 합니다.
값을 TRUE로 하는 비트 위치는 변환 비트 위치 'In'의 하위 'Size' 비트에서 지정합니다.
InOut[]에 전달하는 입출력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

'In'=BYTE#16#09, 'Size'=USINT#4, InOut[]이 BYTE 형식 배열인 경우를 예로 들어 설명합니다.
'In'의 하위 'Size' 비트의 값 16#9는 10진수로 고치면 9입니다. 따라서 InOut[]의 최하위에서 9번째 비트가 TRUE가 되며, 그 이외의 비트는 FALSE가 됩니다.

InOut[]은 BYTE 형식 배열이므로 최하위에서 9번째 비트는 InOut[1]의 비트 1입니다. 따라서 InOut[1]의 비트 1이 TRUE가 되며, InOut[0]의 모든 비트와 InOut[1]의 비트 1 이외의 비트는 FALSE가 됩니다.



InOut[]의 요소의 비트 수가 'Size'로 표시되는 비트 수보다 큰 경우, 나머지 비트의 값은 유지됩니다. 'In'=BYTE#16#02, 'Size'=USINT#2, InOut[]이 WORD 형식 배열인 경우를 예로 들어 설명합니다. 'Size'=USINT#2이므로 InOut[0]의 하위 4비트에 값이 설정됩니다. 그리고 InOut[0]의 나머지 비트인 비트 4~15의 값은 유지됩니다.



참고

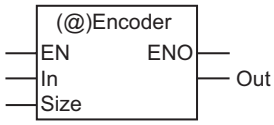
최대 256비트로 이루어진 배열 요소 가운데 TRUE가 된 비트의 위치를 구하는 경우에는 'Encoder 명령(2-392페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Size'의 값이 0인 경우, InOut[]의 모든 비트가 FALSE가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'가 유효 범위 밖인 경우
  - $2^{'Size'}$의 값이 InOut[]의 배열 요소의 비트 수를 초과한 경우

Encoder

최대 256비트로 이루어진 배열 요소 가운데, 값이 TRUE인 비트의 위치를 구합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Encoder	비트 엔코더	FUN		Out:=Encoder(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	변환 대상 배열	입력	변환 대상 배열	데이터 형식에 따름	–	(*)
Size	변환 비트 수		변환할 비트 수	0~8	비트	1
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	○	○	○	○	○															
Size						○														
Out		○																		

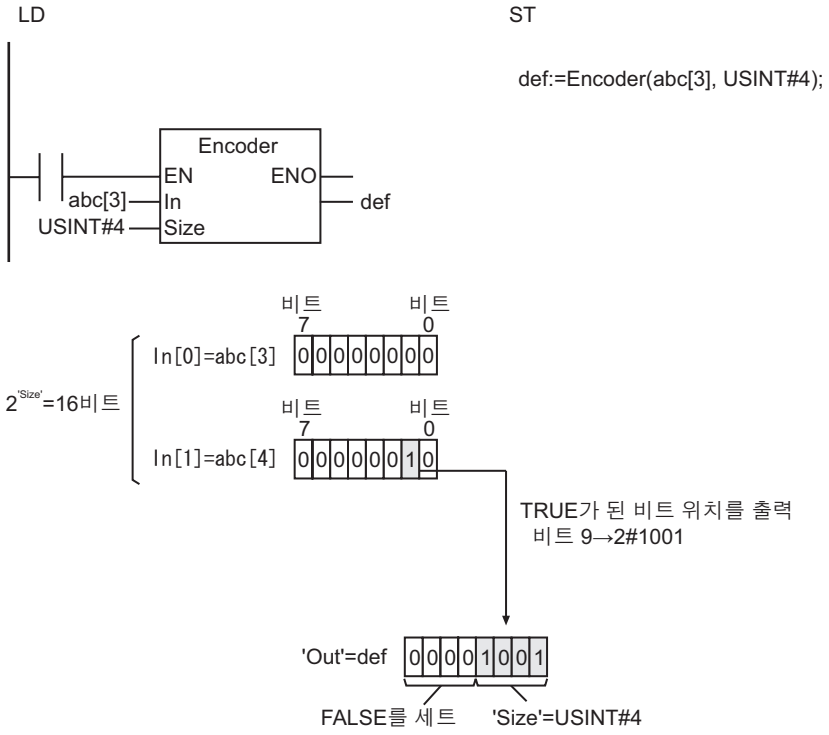
기능

변환 대상 배열 In[]의 임의의 범위에 있는 값이 TRUE인 비트의 위치를 구합니다.
비트 위치는 In[0]부터 $2^{'Size'}$ 비트의 범위에서 구합니다. 이 범위 안에 있는 값이 TRUE인 비트의 위치를 2진수로 표시하고 변환 결과 'Out'의 하위 'Size' 비트에 저장합니다. 그리고 'Out'의 나머지 비트에는 FALSE를 저장합니다.

만일 지정 범위 안에 값이 TRUE인 비트가 여러 개 있는 경우에는 그 가운데 최상위에 있는 비트의 위치를 구합니다.
In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

'Size'=USINT#4, In[]이 BYTE 형식 배열인 경우의 예를 나타냅니다.
'Size'=USINT#4이므로, In[0]부터 시작하여 2^4=16비트가 TRUE인 비트 위치를 구하는 대상 범위가 됩니다. 다음 그림에서는 이 범위의 9비트째에 값이 TRUE인 비트가 있습니다.

'Size'=USINT#4이므로 'Out'의 하위 4비트에 구한 9=2#1001을 저장합니다. 'Out'의 상위 4비트에는 FALSE를 저장합니다.



참고

최대 256비트로 이루어진 배열 요소 가운데 1비트를 TRUE, 그 이외의 비트를 FALSE로 설정하는 경우에는 'Decoder 명령(2-389페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Size'의 값이 0인 경우, 'Out'의 모든 비트가 FALSE가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'가 유효 범위 밖인 경우
  - 2'Size'의 값이 In[]의 배열 요소의 비트 수를 초과한 경우
  - In[]의 비트 가운데 'Size'로 지정된 모든 값이 FALSE인 경우

BitCnt

비트 열 내에서 값이 TRUE인 비트의 총수를 카운트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
BitCnt	비트 카운터	FUN	(@)BitCnt EN ENO In Out	Out:=BitCnt(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	카운트 대상	입력	값이 TRUE인 비트를 카운트할 대상	데이터 형식에 따름	–	(*)
Out	카운트 결과	출력	값이 TRUE인 비트의 수	0~'In'의 비트 수	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

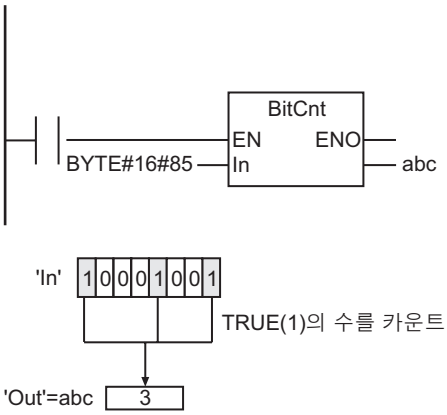
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out						○														

기능

카운트 대상 'In' 내에서 값이 TRUE인 비트의 총수를 카운트합니다.

'In'이 BYTE 형식이고 값이 BYTE#16#85인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=BitCnt(BYTE#16#85);
```

ColmToLine_**

각 배열 요소에서 지정된 위치에 있는 비트의 값을 추출하여 비트 열로 출력합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ColmToLine_**	비트 열→비트 행 변환 그룹	FUN	(@)ColmToLine_** EN ENO In Out Size Pos 단, **는 비트 열의 데이터 형식명	Out:=ColmToLine_**(In, Size, Pos); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	변환 대상 배열	입력	변환 대상 배열	데이터 형식에 따름	–	(*)
Size	변환 대상의 요소 수		In[] 가운데 변환 대상이 되는 요소의 수	0~'Out'의 비트 수		1
Pos	변환 비트 위치		변환할 비트 위치	0~In[]의 비트 수-1		0
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○	○	○	○															
Size						○														
Pos						○														
Out		○	○	○	○															

기능

배열의 각 요소의 지정된 위치에 있는 비트의 값을 추출하여 그것을 순서대로 비트 열로 출력합니다.

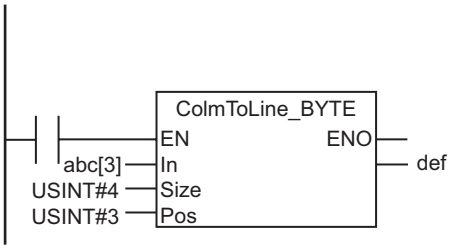
우선 변환 대상 배열 In[]의 In[0]에서 시작되는 'Size'개의 요소를 추출합니다. 그런 다음, 그 중 'Pos' 비트 째의 값만을 추출합니다. 그것들을 순서대로 'Size' 비트의 비트 열로 만들어 변환 결과 'Out'의 최하위 비트 위치에 저장합니다. 그리고 'Out'의 나머지 비트에는 FALSE를 저장합니다.

명령 이름은 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 BYTE 형식인 경우, 명령 이름은 ColmToLine_BYTE가 됩니다.

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함하여 지정합니다.

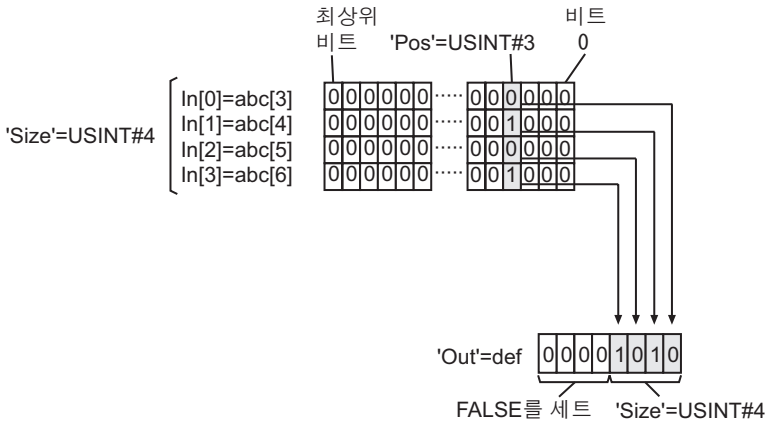
ColmToLine_BYTE 명령에서 'Pos'=USINT#3, 'Size'=USINT#4인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=ColmToLine_BYTE(abc[3], USINT#4, USINT#3);
```



참고

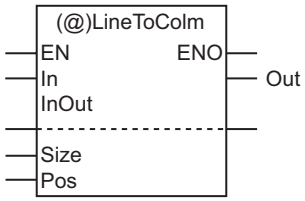
각 배열 요소의 지정된 비트 위치에 비트 열을 출력하는 경우에는 'LineToColm 명령(2-397페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Size'의 값이 0인 경우, 'Out'의 모든 비트가 FALSE가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Pos'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

LineToColm

비트 열을 분해하여 배열 요소의 지정된 비트 위치로 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LineToColm	비트 행→ 비트 열 변환	FUN		LineToColm(In, InOut, Size, Pos);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	(*)
Size	변환 결과의 요소 수		변환 결과의 요소 수	0~'In'의 비트 수		1
Pos	변환 비트 위치		변환할 비트 위치	0~InOut[]의 비트 수-1		0
InOut[] 배열	변환 결과 배열	입출력	변환 결과	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

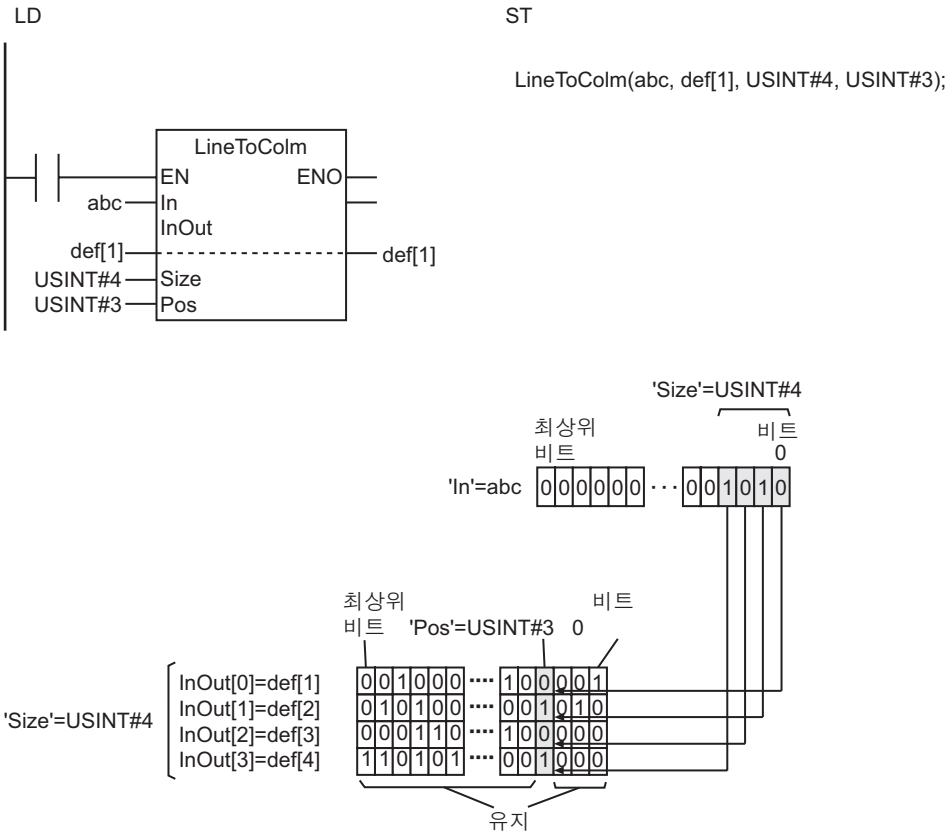
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Size						○														
Pos						○														
InOut[] 배열		○	○	○	○															
Out	○																			

기능

비트 열을 분해하여 배열 요소의 지정된 비트 위치에 출력합니다.

우선 변환 대상 'In'의 최하위 비트에서 'Size' 비트만큼을 추출하여 1비트씩 분해합니다.
그런 다음, 그것들을 변환 결과 배열 InOut[]의 InOut[0]부터 시작하여 각 요소의 'Pos' 비트째에 저장합니다. 저장되는 배열 요소 수는 'Size'개가 됩니다.
값이 저장되는 비트 이외의 비트는 값을 유지합니다.

'Pos'=USINT#3, 'Size'=UINT#4인 경우의 예가 아래에 나와 있습니다.



참고

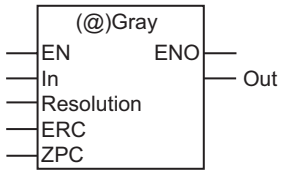
각 배열 요소의 지정된 위치에 있는 비트의 값을 추출하여 비트 열에 출력하는 경우에는 'ColmToLine_** 명령(2-395페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Size'의 값이 0인 경우, InOut[]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Pos'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

Gray

그레이 코드를 각도로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Gray	그레이 코드 변환	FUN		Out:=Gray(In, Resolution, ERC, ZPC);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상의 그레이 코드	데이터 형식에 따름	–	0
Resolution	분해능		분해능	_R256, _R1B~_R15B, _R360, _R720, _R1024		_R256
ERC	엔코더 나머지 보정값		엔코더 나머지 보정값	0~'Resolution'의 분해능		0
ZPC	원점 보정값		원점 보정값			
Out	변환 결과	출력	변환 결과	(*)	°	–

* 0~3.59999999999999e+2

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In			○																	
Resolution	열거 형식_eGRY_RESOLUTION 열거자는 기능 설명 참조																			
ERC							○													
ZPC							○													
Out															○					

기능

그레이 코드로 표현된 로터리 엔코더의 출력값 'In'을 각도값으로 변환합니다. 변환 결과 'Out'의 단위는 도(°)입니다.

'Resolution'의 데이터 형식은 열거 형식 _eGRY_RESOLUTION입니다. 열거자의 의미는 다음과 같습니다.

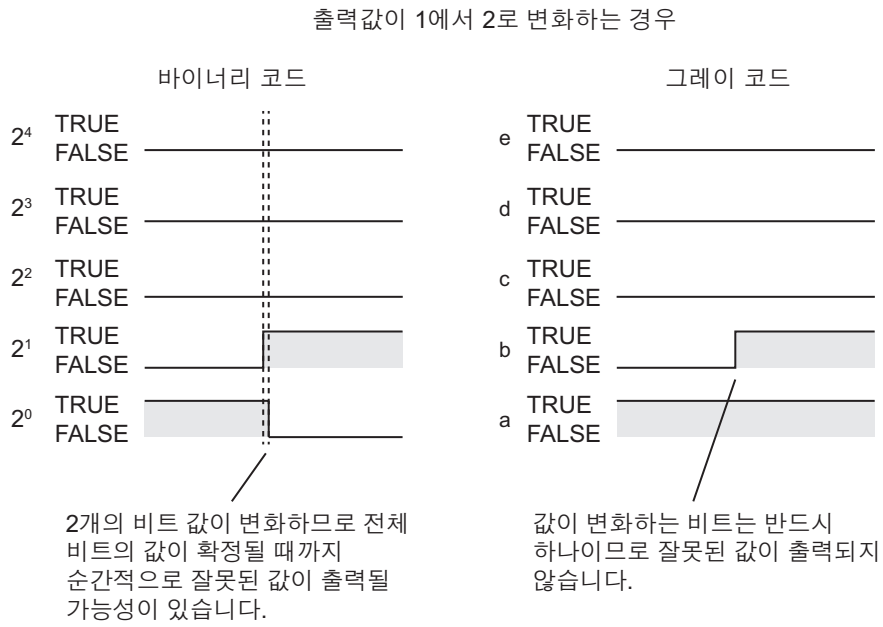
열거자	의미
_R256	256
_R1B	1비트(2)
_R2B	2비트(4)
_R3B	3비트(8)
_R4B	4비트(16)
_R5B	5비트(32)
_R6B	6비트(64)
_R7B	7비트(128)
_R8B	8비트(256)
_R9B	9비트(512)
_R10B	10비트(1024)
_R11B	11비트(2048)
_R12B	12비트(4096)
_R13B	13비트(8192)
_R14B	14비트(16384)
_R15B	15비트(32768)
_R360	360
_R720	720
_R1024	1024

그레이 코드

그레이 코드는 2진수의 일종으로, 교번 이진 부호라고 부릅니다. 0과 1, 1과 2 등과 같이 차가 1인 수치의 코드는 반드시 하나의 비트 값만 다르다는 특징이 있습니다. 그레이 코드는 앱솔루트 엔코더의 출력 등에 사용됩니다.
아래에 4비트의 바이너리 코드와 그레이 코드가 나와 있습니다.

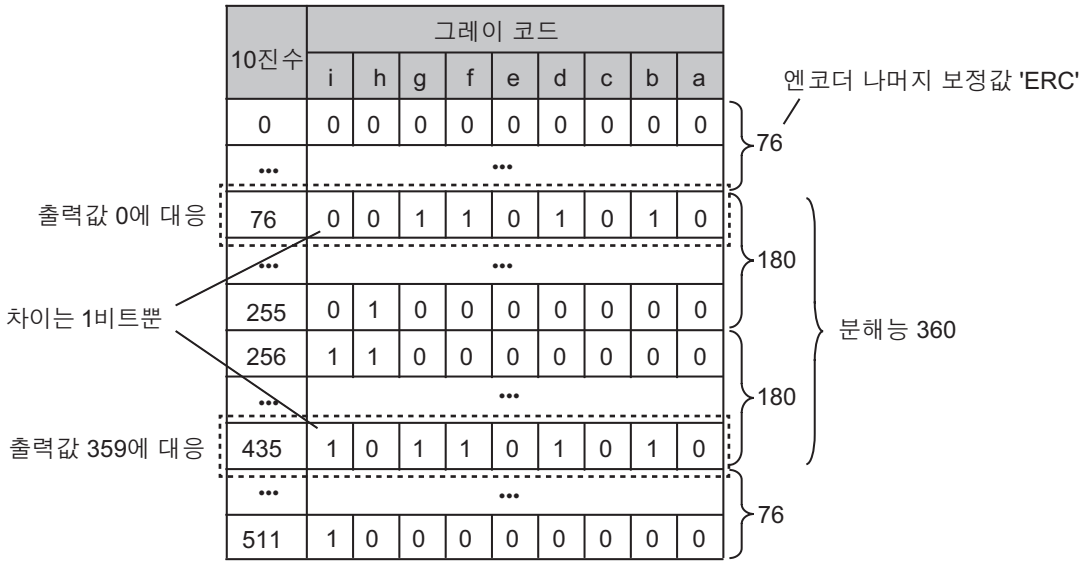
10진수	바이너리 코드				그레이 코드			
	2^3	2^2	2^1	2^0	d	c	b	a
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

그레이 코드를 사용하면 엔코더의 출력값이 1 증가 또는 감소할 때 하나의 비트 값만 변화하므로 순간적으로 잘못된 값이 출력되는 것을 막을 수 있습니다. 그레이 코드와 바이너리 코드의 차이에 따른 엔코더 출력값의 변화 차이는 다음과 같습니다.



엔코더 나머지 보정값 'ERC'

'ERC'는 엔코더의 분해능이 2의 거듭제곱이 아닌 경우에 엔코더 출력의 최대값과 최소값에 대한 그레이 드 코드의 차이가 1비트가 되도록 그레이 코드의 범위를 지정하는 보정값입니다.
예를 들어, 분해능이 360인 앱솔루트 엔코더를 사용한다고 가정하면 그레이 코드에는 9비트가 사용됩니다. 9비트로 표현할 수 있는 값의 범위는 0~511입니다.
이 경우, 0~511의 중간을 기준으로 전후 180씩의 범위, 즉 76~435에 대한 그레이 코드를 사용합니다.
따라서 출력값 0을 나타내는 그레이 코드는 001101010(10진수로 76), 출력값 359를 나타내는 그레이 코드는 101101010(10진수로 435)이 되며, 둘의 차이는 1비트가 됩니다.
이 경우의 엔코더 나머지 보정값 'ERC'의 값은 76이 됩니다.



10진수	그레이 코드 i	h	g	f	e	d	c	b	a
0	0	0	0	0	0	0	0	0	0
...	...								
76	0	0	1	1	0	1	0	1	0
...	...								
255	0	1	0	0	0	0	0	0	0
256	1	1	0	0	0	0	0	0	0
...	...								
435	1	0	1	1	0	1	0	1	0
...	...								
511	1	0	0	0	0	0	0	0	0

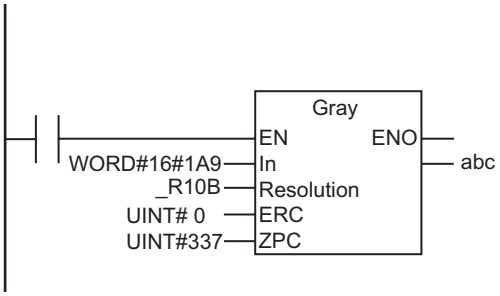
원점 보정값 'ZPC'

'ZPC'는 로터리 엔코더 원점의 각도를 이동하는 경우에 설정합니다. 예를 들어, 분해능이 256인 로터리 엔코더의 원점을 90° 이동하는 경우, 'ZPC'의 값은 256×(90/360)=64로 설정합니다.

기술 예

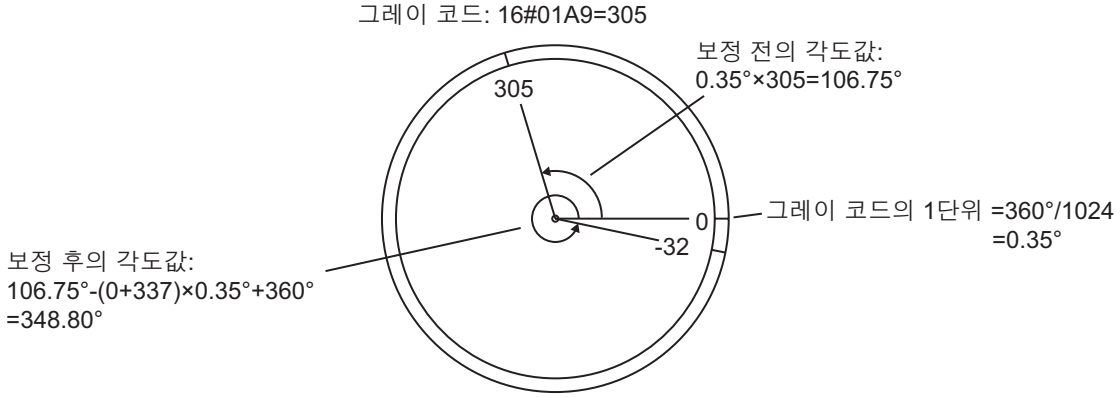
'In'=WORD#16#1A9, 'Resolution'=R10B, 'ERC'=UINT#0, 'ZPC'=UINT#337인 경우의 예가 아래에 나와 있습니다.
먼저, 분해능이 10비트이므로 그레이 코드의 1단위는 360°/1024=0.35°가 됩니다.
그레이 코드 16#01A9에 대응하는 10진수의 값은 305입니다. 따라서 보정 전의 각도값은 0.35°×305=106.75°가 됩니다.
'ERC'의 값이 0, 'ZPC'의 값이 337입니다. 따라서 이를 보정한 각도값은 106.75°-(0+337)×0.35°=-11.20°가 됩니다.
'Out' 값의 범위는 0 이상이므로 -11.20° +360° =348.80° 가 됩니다. 'Out' 값은 LREAL#348.8이 됩니다.

LD



ST

```
abc:=Gray(WORD#16#1A9, _R10B, UINT#0, UINT#337);
```



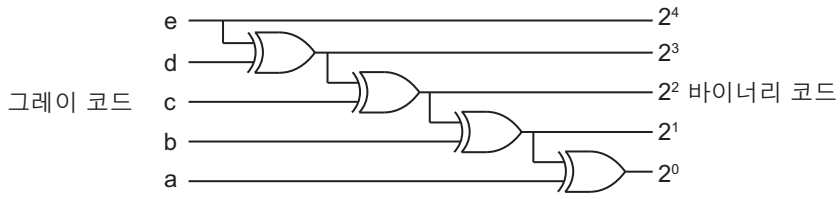
참고

'Resolution' 및 'ERC'에 지정하는 값은 사용하는 로터리 엔코더의 사용 설명서 등을 참조하여 결정하십시오.

그레이 코드에서 바이너리 코드로 변환

다음과 같은 처리를 통해 그레이 코드를 바이너리 코드로 변환할 수 있습니다. 그림의 논리 기호는 배타적 논리합을 나타냅니다.

5비트인 경우의 변환 회로



사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. 'ENO'는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'Resolution'의 값이 유효 범위 밖인 경우
- 'ERC'의 값이 'Resolution'으로 지정된 분해능 이상인 경우
- 'ZPC'의 값이 'Resolution'으로 지정된 분해능 이상인 경우
- 'In'을 비트 열로 변환한 값이 'ERC'의 값보다 작은 경우
- 비트 열을 'ERC'로 보정한 값이 'Resolution'으로 지정된 분해능 이상인 경우

UTF8ToSJIS

문자 코드 UTF-8의 문자열을 문자 코드 SJIS의 BYTE 형식 배열로 변환합니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
UTF8ToSJIS	문자 코드 변환 (UTF-8 → SJIS)	FUN	(@)UTF8ToSJIS EN ENO In Out SJISCode	Out:=UTF8ToSJIS(In, SJISCode);

변수

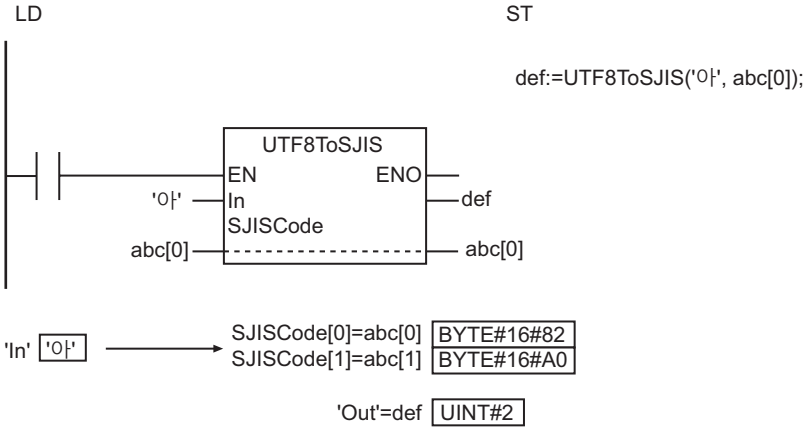
	이름	입/출력	내용	유효 범위	단위	초기값
In	변환 대상 문자열	입력	변환할 문자열	데이터 형식에 따름	–	''
SJIS Code[] 배열	SJIS 배열	입출력	문자 코드 SJIS의 배열	데이터 형식에 따름	–	–
Out	변환 요소 수	출력	SJISCode[]에 저장된 요소 수	0~1985	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
SJIS Code[] 배열		○																		
Out							○													

기능

문자 코드 UTF-8의 변환 대상 문자열 'In'을 문자 코드 SJIS의 BYTE 형식 배열 SJISCode[]로 변환합니다.
변환 후 데이터는 1바이트씩 분할하여 SJISCode[0]부터 순서대로 저장합니다.
변환 요소 수 'Out'에는 SJISCode[]에 저장된 변환 후 데이터의 요소 수를 저장합니다.

'In'='아'인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 'In'의 끝에 있는 NULL 문자는 변환되지 않습니다. 또한 변환 요소 수에도 카운트되지 않습니다.
- 'In'이 NULL 문자만 있는 문자열인 경우, 'Out'의 값은 0이 되며 SJISCode[]는 변화하지 않습니다.
- SJISCode[]의 'Out'번째 이후의 요소는 변화하지 않습니다. 예를 들어 변환 요소 수가 5인 경우, SJISCode[5] 이후의 요소는 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'ENO'는 FALSE가 되고 'Out', SJISCode[]는 변화하지 않습니다.
  - 변환 결과의 요소 수가 SJISCode[]에 연결하는 출력 파라메터의 영역 크기를 초과하는 경우
  - 'In'의 내용이 변환 불가능한 문자인 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 사용할 수 있습니다.

SJISToUTF8

문자 코드 SJIS의 BYTE 형식 배열을 문자 코드 UTF-8의 문자열로 변환합니다.

명령	이름	FB/FUN	그래픽 표현	ST 표현
SJISToUTF8	문자 코드 변환 (SJIS → UTF-8)	FUN	(@)SJISToUTF8 EN ENO In Out Size	Out:=SJISToUTF8(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	변환 대상 SJIS 배열	입력	변환할 문자 코드 SJIS의 배열 (*1)	데이터 형식에 따름	–	(*2)
Size	SJIS 배열 요소 수		변환할 In[]의 요소 수			–
Out	변환 문자열	출력	문자 코드 UTF-8로 변환한 문자열	데이터 형식에 따름	–	–

*1 NULL 문자(BYTE#16#00)를 포함하는 경우의 최대 요소 수는 1986입니다. NULL 문자를 포함하지 않는 경우의 최대 요소 수는 1985입니다.

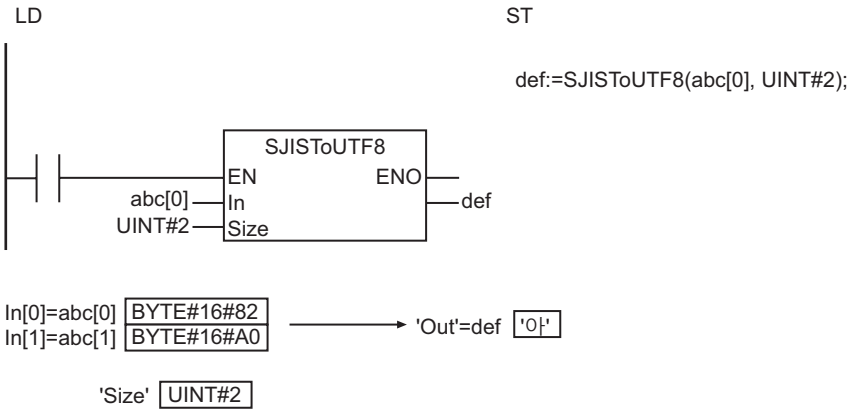
*2 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○																		
Size							○													
Out																				○

기능

문자 코드 SJIS의 BYTE 형식 배열 In[]의 요소를 문자 코드 UTF-8의 문자열로 변환합니다.
변환 대상은 In[0] 이후의 'Size'개의 요소입니다. 단, 도중에 NULL 문자(BYTE#16#00)가 있는 경우는 거기까지만 변환하고 중지합니다.
변환 후의 문자열은 변환 문자열 'Out'에 저장됩니다. 'Out'의 끝에는 NULL 문자를 추가합니다.

In[0]=BYTE#16#82, In[1]=BYTE#16#A0, 'Size'=UINT#2인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

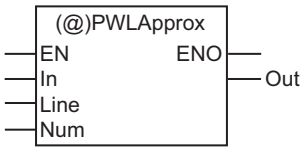
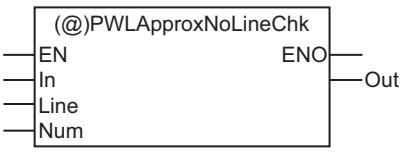
- 'Size'의 값이 0인 경우, 'Out'의 값은 NULL 문자뿐입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'ENO'는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열의 크기를 초과하는 경우
  - In[]의 내용이 변환 불가능한 문자인 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 사용할 수 있습니다.

PWLApprox / PWLApproxNoLineChk

정수, 실수의 꺾은선 근사 계산을 수행합니다.
PWLApprox : 꺾은선 데이터가 유효한지 확인합니다.
PWLApproxNoLineChk : 꺾은선 데이터가 유효한지 확인하지 않습니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
PWLApprox	꺾은선 근사 변환(꺾은선 확인 있음)	FUN		Out:=PWLApprox(In, Line, Num);
PWLApprox NoLineChk	꺾은선 근사 변환(꺾은선 확인 없음)	FUN		Out:=PWLApproxNoLineChk(In, Line, Num);

변수

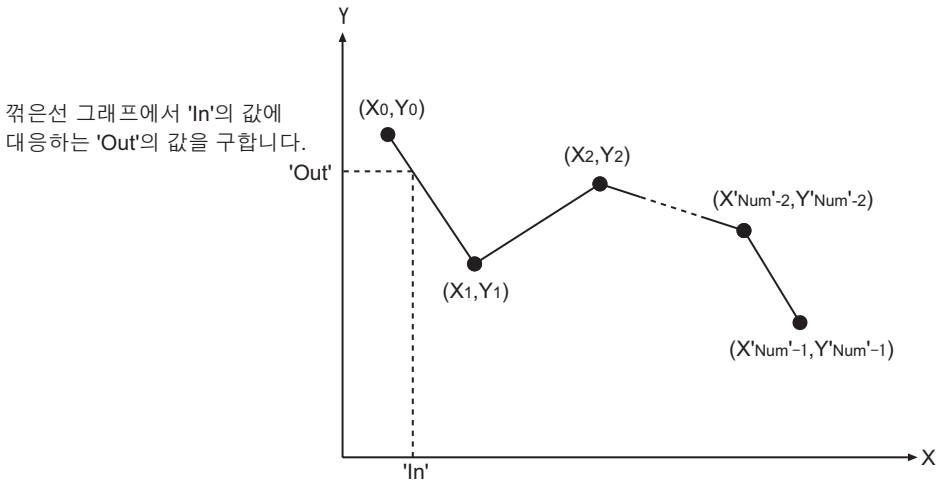
	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	(*)
Line[] 배열	꺾은선 데이터 배열		꺾은선 데이터 배열			
Num	꺾은선 데이터의 수		꺾은선 데이터의 수			1
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○	○	○					
Line[] 배열	'In'과 동일한 데이터 형식을 요소로 하는 배열																			
Num							○													
Out						○	○	○	○	○	○	○	○	○	○					

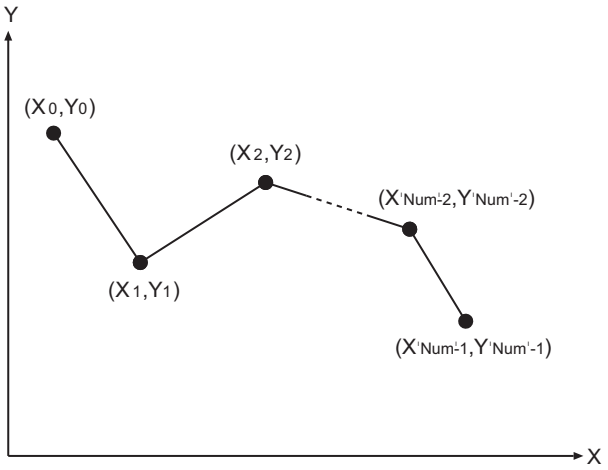
기능

꺾은선 데이터 배열 Line[]에서 Line[0,0] 이후 ('Num'×2)개의 요소로 구성된 꺾은선 데이터에 기초하여 변환 대상 'In'을 근사 변환합니다.
그림과 같이 꺾은선 데이터상의 X 좌표 'In'에 대응하는 Y 좌표가 변환 결과 'Out'에 대입됩니다.



꺾은선 데이터 Line[]의 요소와 꺾은선 데이터의 수 'Num'

Line[]은 2차원 또는 3차원 배열로, 제1차원의 요소 수를 2로 하십시오. 그리고 그림과 같이 꺾은선 데이터의 각 점의 좌표값(X_0,Y_0), (X_1,Y_1) · · · 을 Line[]의 각 요소로 하십시오. 꺾은선 데이터의 수 'Num'은 꺾은선 근사 계산에 사용하는 Line[]의 요소 수의 1/2이 됩니다.



Line[]이 2차원 배열인 경우

Line[0,0]	X_0
Line[0,1]	Y_0
Line[1,0]	X_1
Line[1,1]	Y_1
Line[2,0]	X_2
Line[2,1]	Y_2
:	:
Line['Num'-1,0]	$X_{'Num'-1}$
Line['Num'-1,1]	$Y_{'Num'-1}$

Line[]이 3차원인 경우

Line[0,0,0]	X_0
Line[0,0,1]	Y_0
Line[0,1,0]	X_1
Line[0,1,1]	Y_1
Line[0,2,0]	X_2
Line[0,2,1]	Y_2
:	:
Line[0, 'Num'-1,0]	$X_{'Num'-1}$
Line[0, 'Num'-1,0]	$Y_{'Num'-1}$

기술 예

요소 수 4인 꺾은선 데이터 배열 abc[]에 기초하여 'In'=LREAL#3.0을 근사 변환하는 경우의 예가 아래에 나와 있습니다. 'Num'＝UINT#4일 때 abc[]의 요소 값은 다음과 같습니다.

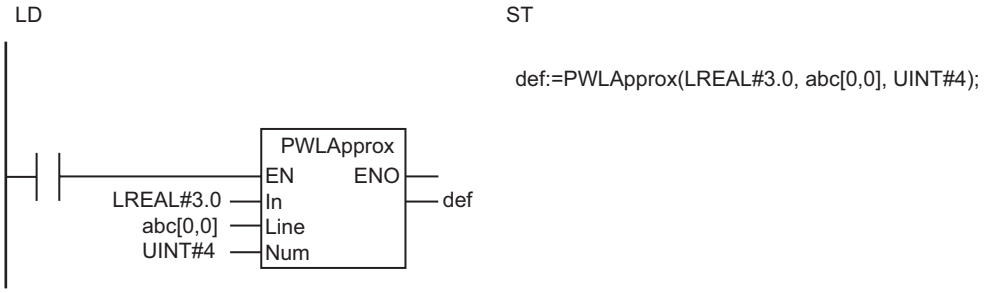
abc[0.0]=X_0=LREAL#1.0, abc[0,1]=Y_0=LREAL#5.0,

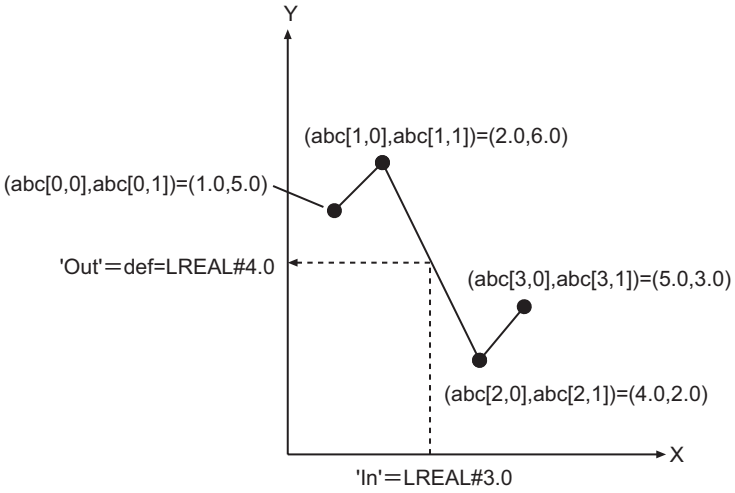
abc[1.0]=X_1=LREAL#2.0, abc[1,1]=Y_1=LREAL#6.0,

abc[2.0]=X_2=LREAL#4.0, abc[2,1]=Y_2=LREAL#2.0,

abc[3.0]=X_3=LREAL#5.0, abc[3,1]=Y_3=LREAL#3.0

변환 결과, 'Out'의 값은 LREAL#4.0이 됩니다.





PWLApprox 명령과 PWLApproxNoLineChk 명령의 차이

PWLApprox 명령과 PWLApproxNoLineChk 명령의 차이는 'In' 또는 Line[]의 유효성 확인 여부와 처리 시간입니다. 각각의 사양은 다음과 같습니다.

명령	확인 내용	데이터가 유효가 아닌 경우의 처리	처리 시간
PWLApprox	• Line[]의 내용이 X 좌표의 오름차순으로 정렬되어 있을 것 • 'In', Line[]이 실수인 경우, 'In'과 Line[]의 요소가 비수, 양의 무한대, 음의 무한대가 아닐 것	• 이상 상태가 됩니다. • ENO의 값이 FALSE가 됩니다. • 'Out'의 값이 변화하지 않습니다.	장시간
PWLApprox NoLineChk	확인 안 함	• 이상 상태가 되지 않습니다. • ENO의 값이 TRUE가 됩니다. • 'Out'에 유효한 값이 출력되지 않는 경우가 있습니다.	단시간

PWLApproxNoLineChk 명령과 PWLLineChk 명령

PWLApproxNoLineChk는 'In' 또는 Line[]의 유효성을 확인하지 않는 대신 처리 시간이 짧다는 특징이 있습니다. 따라서 그러한 입력 변수의 유효성이 보증되는 경우는 PWLApprox 명령보다 PWLApproxNoLineChk 명령을 사용하는 것이 유리합니다.

한편 'PWLLineChk 명령(2-413페이지)'은 Line[]의 내용이 X 좌표의 오름차순으로 정렬되어 있는지 확인하는 명령입니다. 따라서 평상시에는 PWLApproxNoLineChk 명령을 사용하고, Line[]의 내용이 X 좌표의 오름차순으로 정렬되어 있는지 보증되지 않는 경우에만 PWLLineChk 명령을 함께 사용하여 처리 시간을 단축하는 방법도 있습니다.

참고

근사 변환에 사용하는 꺾은선 데이터 배열의 요소 범위를 좁혀도 처리 시간을 단축할 수 있습니다.

앞의 기술 예의 경우, 'In'의 값은 LREAL#3.0이므로 꺾은선 데이터 배열의 요소 중에서 X 좌표값이 3.0에 가까운

(abc[1,0],abc[1,1])=(2.0,6.0)

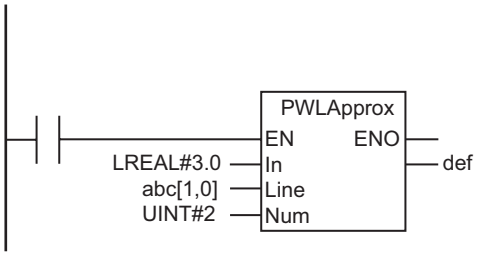
(abc[2,0],abc[2,1])=(4.0,2.0)

의 4개 요소만으로 근사 변환을 하면 처리 시간이 짧아집니다.

이 경우 'Num'＝UINT#2로 하고, Line[]에 전달할 abc[]의 요소는 abc[1,0]로 합니다.

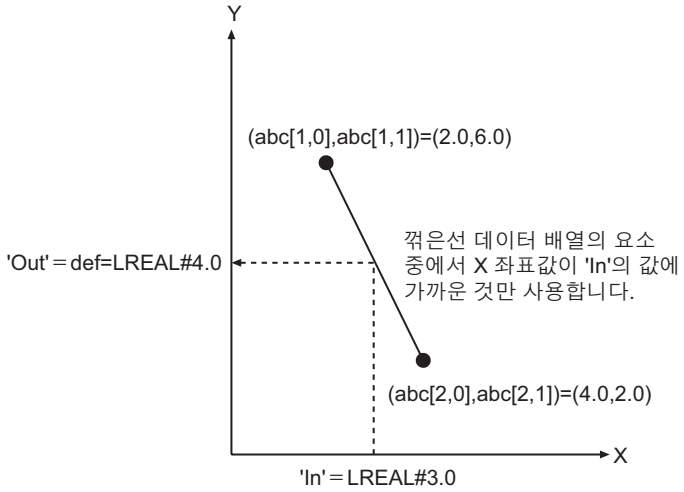
변환 결과는 'Out'＝LREAL#4.0 그대로입니다.

LD



ST

```
def:=PWLApprox(LREAL#3.0, abc[1,0], UINT#2);
```



사용상의 주의 사항

- 'In'의 값이 Line[0,0]의 값(즉, X_1의 값)보다 작은 경우, 'Out'의 값은 Line[0,1](즉, Y_1의 값)이 됩니다.
- 'In'의 값이 Line['Num'-1,0](즉, $X_{'Num'}$의 값)보다 큰 경우, 'Out'의 값은 Line['Num'-1,1](즉, $Y_{'Num'}$의 값)이 됩니다.
- Line[]은 2차원 또는 3차원 배열로, 제1차원의 요소 수를 2로 하십시오.
- 'Num'의 값이 0인 경우, 'Out'의 값은 0이 됩니다.
- 다음과 같은 경우, PWLApprox 명령에서는 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다. PWLApproxNoLineChk 명령에서는 이상 상태가 되지 않습니다.
  - 꺾은선 데이터의 X 좌표가 오름차순, 즉 $X_1<X_2<\cdots<X_{'Num'}$이 되지 않은 경우
  - 'In'과 Line[]이 실수 형식이고 그 값이 비수, 양의 무한대, 음의 무한대인 경우
- 다음과 같은 경우, PWLApprox 명령에서도 PWLApproxNoLineChk 명령에서도 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Num'의 값이 Line[]의 배열 영역을 초과하는 경우
  - 'In'의 값이 Line[]에서 지정된 꺾은선 데이터의 X 좌표의 범위 밖인 경우

버전 관련 정보

PWLApproxNoLineChk 명령은 CPU 유니트 Ver.1.03 이상, Sysmac Studio Ver.1.04 이상에서 사용할 수 있습니다.

PWLLineChk

꺾은선 근사 변환(꺾은선 데이터 확인 없음) 명령에서 사용하는 꺾은선 데이터가 X 좌표의 오름차순으로 정렬되어 있는지 여부를 판정합니다.

명령	이름	FB/FUN	그래픽 표현	ST 표현
PWLLineChk	꺾은선 데이터 확인	FUN	(@)PWLLineChk EN Line Num Out	Out:=PWLLineChk(Line, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Line[] 배열	꺾은선 데이터 배열	입력	꺾은선 데이터 배열	데이터 형식에 따름	–	(*)
Num	꺾은선 데이터의 수		꺾은선 데이터의 수			1
Out	판정 결과	출력	판정 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

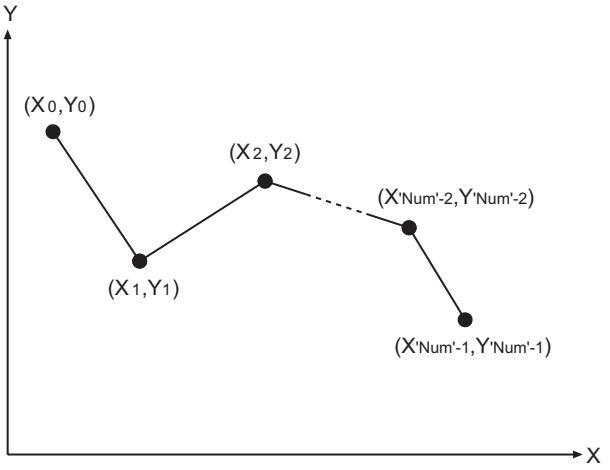
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Line[] 배열						○	○	○	○	○	○	○	○	○	○					
Num							○													
Out	○																			

기능

꺾은선 근사 변환(꺾은선 데이터 확인 없음) PWLApproxNoLineChk 명령에서 사용하는 꺾은선 데이터 배열 Line[]의 내용이 X 좌표의 오름차순으로 정렬되어 있는지 여부를 판정합니다. 오름차순으로 정렬되어 있으면 판정 결과 'Out'의 값이 TRUE, 그렇지 않으면 'Out'의 값이 FALSE가 됩니다.

꺾은선 데이터 Line[]의 요소와 꺾은선 데이터의 수 'Num'

Line[]은 2차원 또는 3차원 배열로, 제1차원의 요소 수를 2로 하십시오. 그리고 그림과 같이 꺾은선 데이터의 각 점의 좌표값(X_0,Y_0), (X_1,Y_1)…을 Line[]의 각 요소로 하십시오. 꺾은선 데이터의 수 'Num'은 꺾은선 근사 계산에 사용하는 Line[]의 요소 수의 1/2이 됩니다.



Line[]이 2차원 배열인 경우

Line[0,0]	X_0
Line[0,1]	Y_0
Line[1,0]	X_1
Line[1,1]	Y_1
Line[2,0]	X_2
Line[2,1]	Y_2
:	:
Line['Num'-1,0]	X'Num'-1
Line['Num'-1,1]	Y'Num'-1

Line[]이 3차원인 경우

Line[0,0,0]	X_0
Line[0,0,1]	Y_0
Line[0,1,0]	X_1
Line[0,1,1]	Y_1
Line[0,2,0]	X_2
Line[0,2,1]	Y_2
:	:
Line[0, 'Num'-1,0]	X'Num'-1
Line[0, 'Num'-1,0]	Y'Num'-1

기술 예

요소 수 4인 꺾은선 데이터 배열 abc[]가 X 좌표의 오름차순으로 정렬되어 있는지 판정하는 경우의 예가 아래에 나와 있습니다. 'Num'＝UINT#4일 때 abc[]의 요소 값은 다음과 같습니다.

abc[0.0]=X_0=LREAL#1.0, abc[0,1]=Y_0=LREAL#5.0,

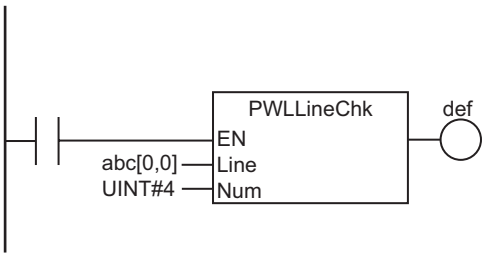
abc[1.0]=X_1=LREAL#6.0, abc[1,1]=Y_1=LREAL#6.0,

abc[2.0]=X_2=LREAL#4.0, abc[2,1]=Y_2=LREAL#2.0,

abc[3.0]=X_3=LREAL#5.0, abc[3,1]=Y_3=LREAL#3.0

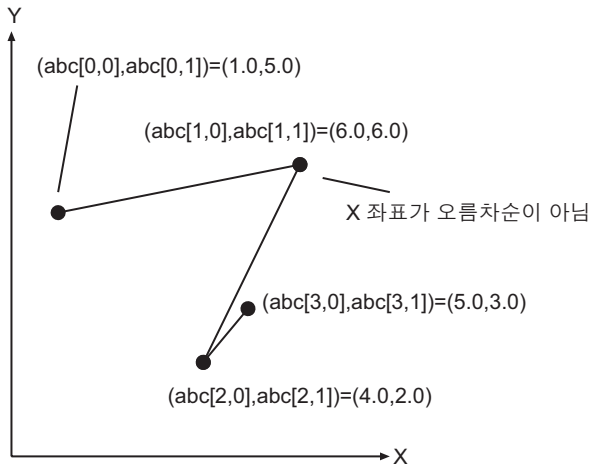
X 좌표가 오름차순이 아니므로 'Out'의 값은 FALSE가 됩니다.

LD



ST

```
def:=PWLLineChk(abc[0,0], UINT#4);
```



참고

- 이 명령은 PWLApproxNoLineChk 명령과 함께 사용하십시오. PWLApproxNoLineChk 명령에 대한 자세한 내용은 'PWLApprox / PWLApproxNoLineChk 명령(2-408페이지)'을 참조하십시오.
- 꺾은선 근사 변환을 할 때마다 꺾은선 데이터 확인을 수행하는 경우는 PWLApprox 명령을 사용하십시오. PWLApprox 명령에 대한 자세한 내용은 'PWLApprox / PWLApproxNoLineChk 명령(2-408페이지)'을 참조하십시오. 단, PWLApprox 명령보다 PWLApproxNoLineChk 명령이 처리 시간이 짧습니다.

사용상의 주의 사항

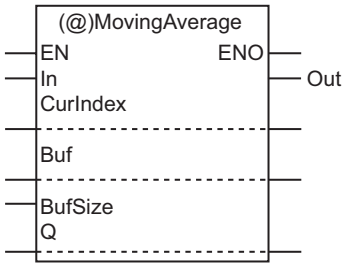
- Line[]은 2차원 또는 3차원 배열로, 제1차원의 요소 수를 2로 하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Out'은 FALSE가 됩니다.
  - 'Num'의 값이 Line[]의 배열 영역을 초과하는 경우
  - Line[]이 실수 형식이고 요소의 값이 비수, 양의 무한대, 음의 무한대인 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.03 이상, Sysmac Studio Ver.1.04 이상에서 사용할 수 있습니다.

MovingAverage

이동 평균값을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Moving Average	이동 평균	FUN		Out:=MovingAverage(In, CurIndex, Buf, BufSize, Q);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	입력값	입력	평균값 연산에 사용할 수치	데이터 형식에 따름	–	(*)
BufSize	최대 요소 수		평균값 연산에 사용할 최대 요소 수			1
CurIndex	입력값 저장 위치	입출력	'In'을 저장할 Buf[]의 위치	데이터 형식에 따름	–	–
Buf[] 배열	입력값 저장 배열		'In'을 저장할 배열			
Q	계산 완료 플래그		TRUE : Buf[]에 저장한 수치 저장 수가 'BufSize' 이상 FALSE : Buf[]에 저장한 수치 저장 수가 'BufSize' 미만			
Out	연산 결과	출력	연산 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

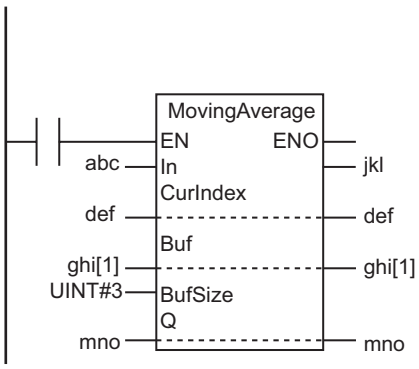
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○	○	○					
BufSize							○													
CurIndex							○													
Buf[] 배열	'In'과 동일한 데이터 형식을 요소로 하는 배열																			
Q	○																			
Out						○	○	○	○	○	○	○	○	○	○					

기능

이 명령이 실행될 때마다 입력값 'In'을 입력값 저장 배열 Buf[]에 저장합니다. 그리고 저장한 값의 평균값을 연산 결과 'Out'에 저장합니다. 평균값 계산에 사용하는 최대 요소 수는 'BufSize'로 지정합니다.

'BufSize'=UINT#3인 경우를 예로 들어 처리 순서를 설명합니다. 기술 방법은 다음과 같습니다.

LD



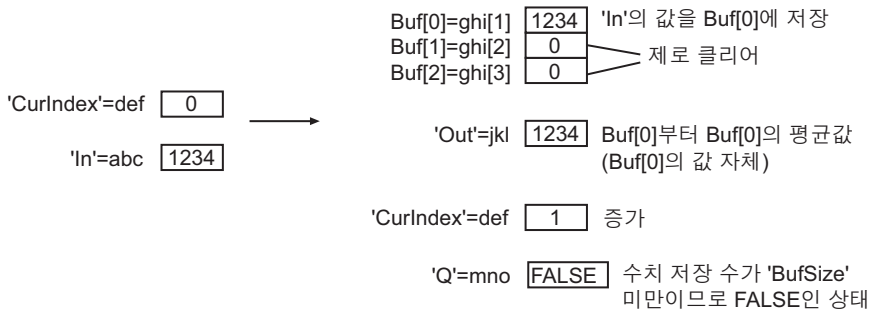
ST

```
jkl:=MovingAverage(abc, def, ghi[1], UINT#3, mno);
```

첫 회의 수치 입력

입력값 저장 위치 'CurIndex'에 0을 설정하여 이 명령을 실행합니다.
입력값 저장 배열 Buf[]의 Buf[0]부터 Buf['BufSize'-1]이 제로 클리어된 후, 최초의 입력값 'In'의 값이 Buf[0]에 저장됩니다.
계산 완료 플래그 'Q'의 값이 FALSE가 됩니다. 이것은 Buf[]에 저장되는 수치 저장 수가 아직 'BufSize'에 도달하지 않았음을 의미합니다.
'Q'의 값이 FALSE인 동안, 평균값 계산은 Buf[0]부터 시작하여 'CurIndex'+1개의 수치로 수행합니다. 그리고 연산 결과가 'Out'에 저장됩니다.
그 후에 'CurIndex'의 값을 증가시킵니다.

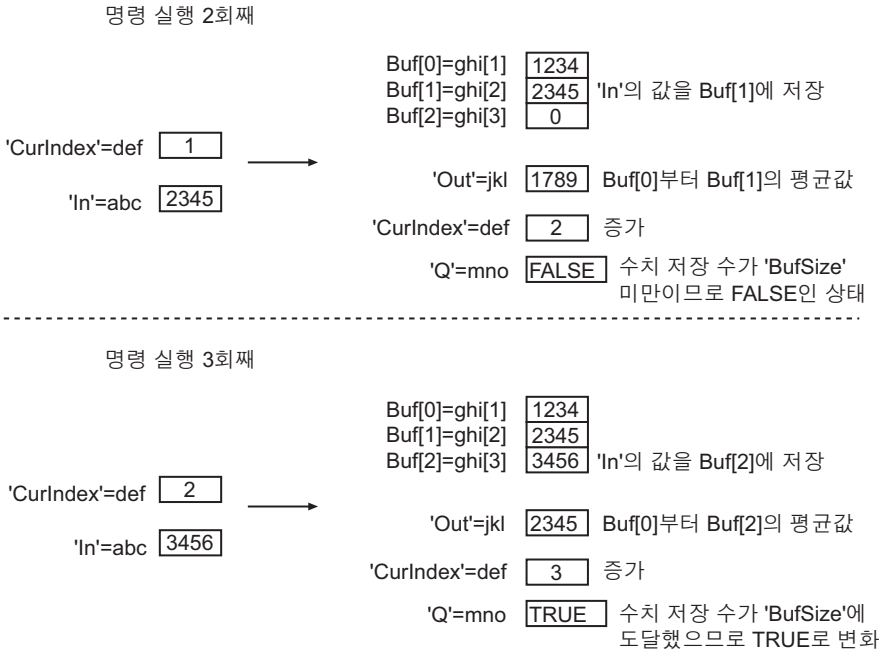
명령 실행 1회째



'BufSize'회째까지의 수치 입력

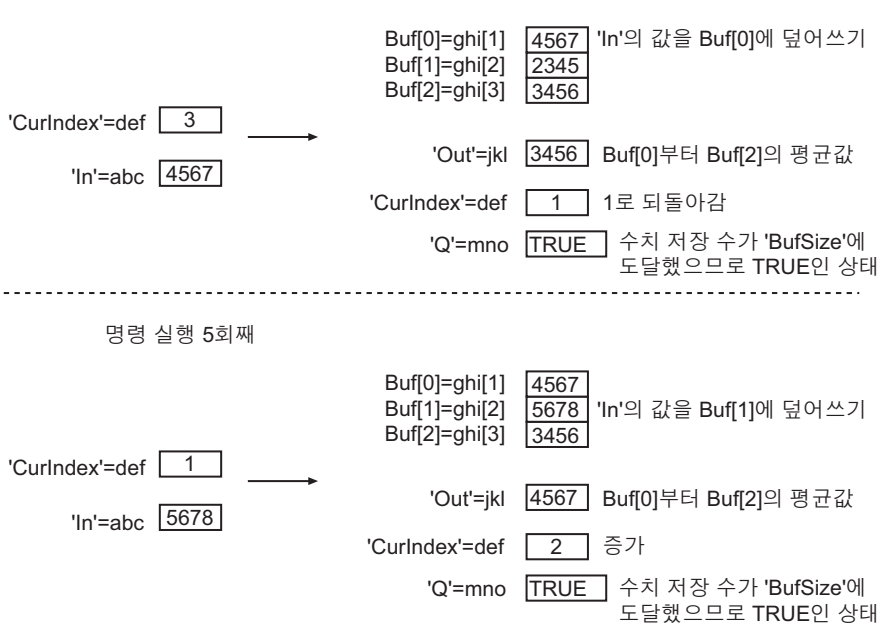
이 명령을 실행할 때마다 'In'의 값이 Buf['CurIndex']에 저장됩니다. 평균값 계산은 Buf[0]에서 시작되는 'CurIndex'+1개의 수치로 수행되며, 연산 결과가 'Out'에 저장됩니다.

실행 횟수가 'BufSize'에 도달하면 'Q'의 값이 TRUE로 변화합니다.



'BufSize'회째후의 수치 입력

이 명령을 실행할 때마다 'In'의 값을 Buf[0]부터 Buf['BufSize'-1] 사이에 주기적으로 덮어씁니다. 평균값 계산은 그 시점에서의 Buf[0]부터 Buf['BufSize'-1]에서 수행되며, 연산 결과가 'Out'에 저장됩니다.
'CurIndex'의 값은 'BufSize'의 다음은 1로 되돌아가며, 거기서 다시 증가됩니다. 'Q'의 값은 그대로 TRUE입니다.

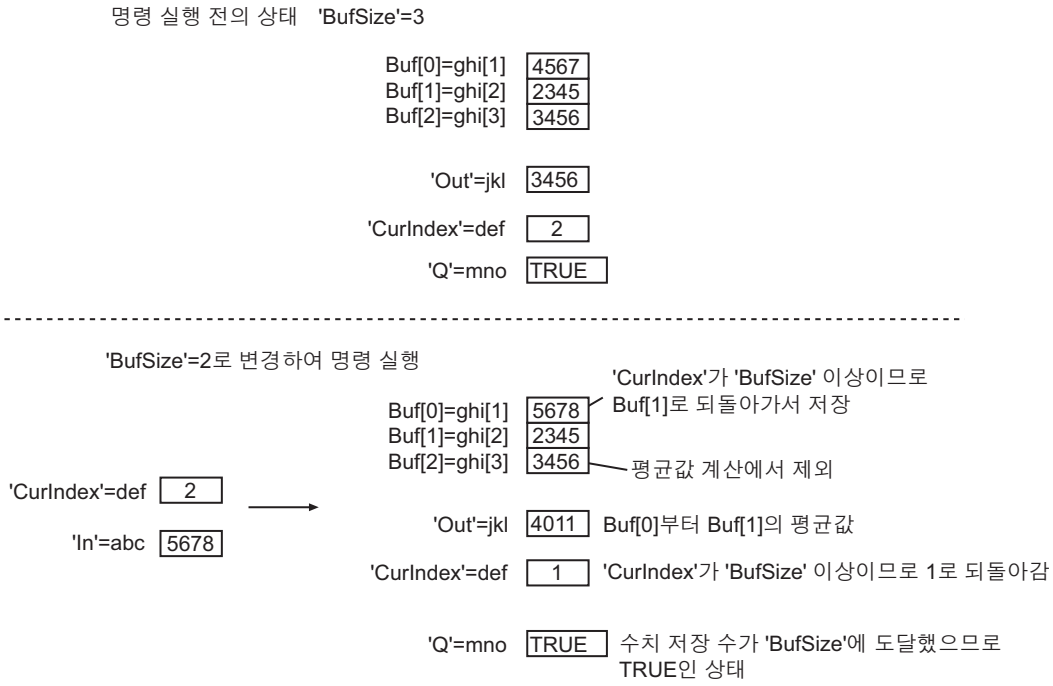


저장값의 초기화

'CurIndex'의 값을 0으로 하여 이 명령을 실행하면, 일단 Buf[0]부터 Buf['BufSize'-1]의 값은 0이 되며, 다시 그 시점의 'In'의 값이 Buf[0]에 저장됩니다.
'CurIndex'의 값은 1이 되고 'Q'의 값은 FALSE가 됩니다.

'BufSize'의 값의 변경

'BufSize'의 값을 변경하여 이 명령을 실행한 경우, 변경 후의 'BufSize'와 그 시점의 'CurIndex'의 값에 따라서 동작합니다.



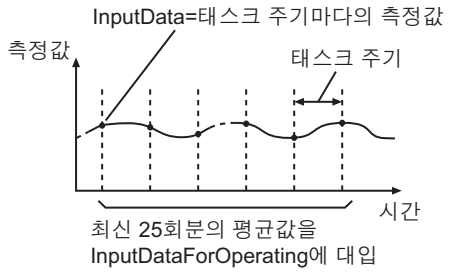
사용상의 주의 사항

- 'In'과 Buf[]의 요소의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- Buf[]의 크기는 반드시 'BufSize' 이상으로 하십시오.
- 연산 결과가 'Out'의 유효 범위 밖이라도 이상 상태가 되지는 않으며, 'Out'에는 잘못된 값이 저장됩니다.
- 'BufSize'의 값이 0인 경우, 'Out', 'CurIndex'의 값은 0이 됩니다. 'Q'의 값은 TRUE가 됩니다.
- 'BufSize'의 값을 변경한 경우, 반드시 'CurIndex'의 값을 0으로 하여 저장값을 초기화하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'BufSize'의 값이 Buf[]의 크기를 초과한 경우

샘플 프로그램

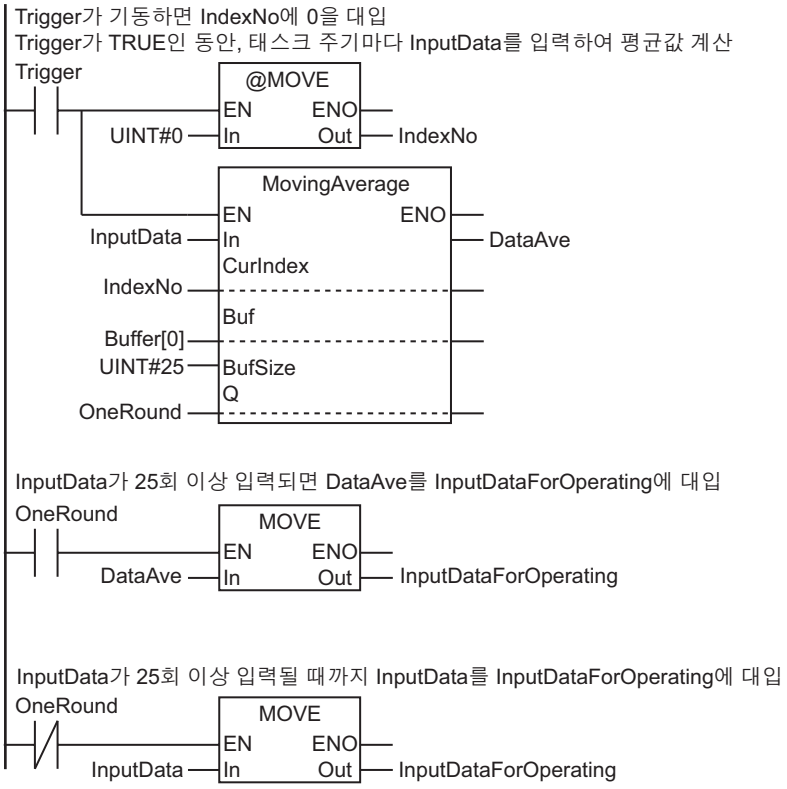
센서 등 아날로그 입력 데이터에 포함되는 노이즈와 같은 외란의 영향을 제거하는 예입니다.
입력 데이터 InputData의 최신 25회분의 평균값 DataAve를 다음 공정의 입력 InputDataForOperating에 대입합니다.
InputData는 실행 조건 Trigger의 값이 TRUE인 동안, 태스크 주기마다 입력됩니다.
InputData가 25회 이상 입력될 때까지는 데이터량이 불충분하므로, 평균값이 아닌 최신 1회의 InputData의 값을 InputDataForOperating에 대입합니다.

Trigger의 값이 FALSE→TRUE가 되면 평균값을 클리어하고 InputData의 입력을 처음부터 다시 수행합니다.



LD

이름	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
InputData	INT	10	입력값
Buffer	ARRAY[0..24] OF INT	[25(0)]	입력값 저장 배열
DataAve	INT	0	평균값
OneRound	BOOL	FALSE	25회 입력 플래그
IndexNo	UINT	0	입력값 저장 위치
InputDataForOperating	INT	0	다음 공정 입력



ST

이름	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
LastTrigger	BOOL	FALSE	이전 테스크 주기의 Trigger의 값
Operating	BOOL	FALSE	처리 중
OperatingStart	BOOL	FALSE	처리 시작
Buffer	ARRAY[0..24] OF INT	[25(0)]	입력값 저장 배열
InputData	INT	10	입력값
DataAve	INT	0	평균값
OneRound	BOOL	FALSE	25회 입력 플래그
IndexNo	UINT	0	입력값 저장 위치
InputDataForOperating	INT	0	다음 공정 입력

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
    OperatingStart:=TRUE;
    Operating:=TRUE;
END_IF;
LastTrigger:=Trigger;

// 평균값 클리어
IF (OperatingStart=TRUE) THEN
    IndexNo:=UINT#0;
    OperatingStart:=FALSE;
END_IF;

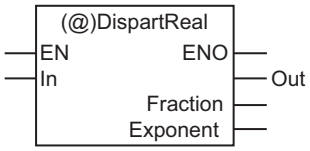
// 이동 평균 처리
IF (Operating=TRUE) THEN
    DataAve:=MovingAverage(
        In          :=InputData,
        CurIndex:=IndexNo,
        Buf         :=Buffer[0],
        BufSize  :=UINT#25,
        Q           :=OneRound);

    IF (OneRound=TRUE) THEN
        // 최신 25회분의 입력값의 평균값을 대입
        InputDataForOperating:=DataAve;
    ELSE
        // 최신 1회분의 입력값을 대입
        InputDataForOperating:=InputData;
    END_IF;
END_IF;

// 이동 평균 처리 종료
IF (Trigger=FALSE) THEN
    Operating:=FALSE;
END_IF;
```

DispartReal

실수를 부호가 있는 가수부와 지수부로 분리합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DispartReal	실수의 가수 · 지수 분리	FUN		Out:=DispartReal(In, Fraction, Exponent);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	실수	입력	분리할 실수	데이터 형식에 따름	–	(*1)
Out	반환값	출력	항상 TRUE	TRUE만	–	–
Fraction	부호가 있는 가수부		부호가 있는 가수부	(*2)		
Exponent	지수부		지수부	(*3)		

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

*2 'In', 'Fraction'의 데이터 형식의 조합으로 유효 범위가 변합니다. 자세한 내용은 기능 설명을 참조하십시오.

*3 'In'이 REAL 형식인 경우, -44~32, LREAL 형식인 경우, -322~294

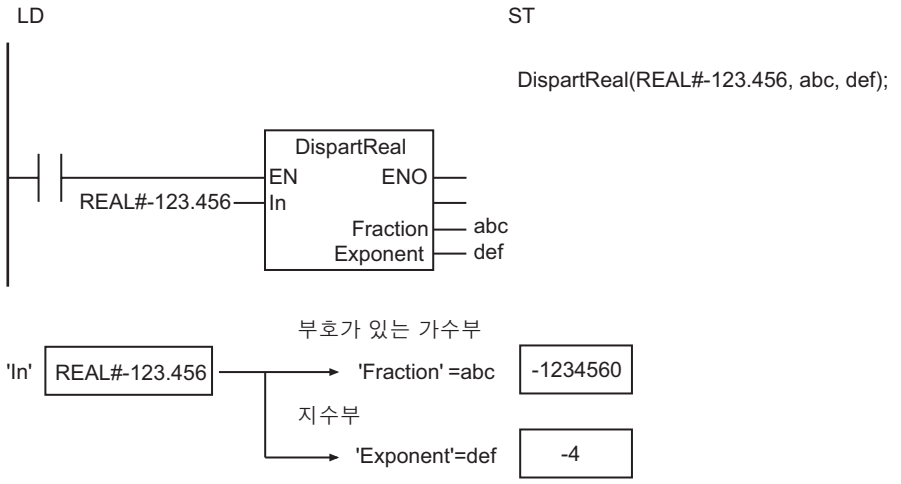
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														○	○					
Out	○																			
Fraction	'In'의 데이터 형식이 REAL이면 DINT, LREAL이면 LINT																			
Exponent											○									

기능

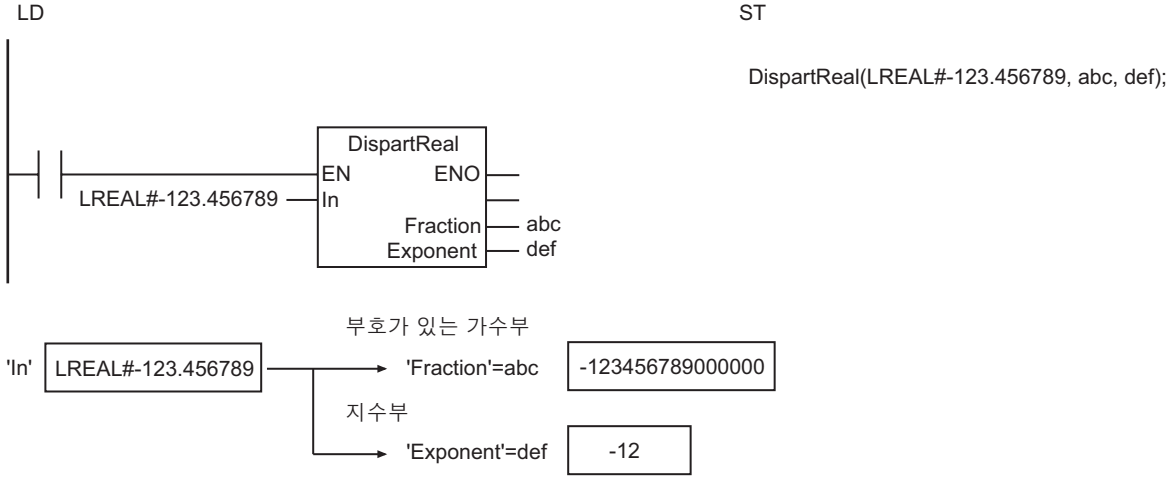
실수 'In'을 부호가 있는 가수부 'Fraction'과 지수부 'Exponent'로 분리합니다.

'In'이 REAL 형식인 경우, 'Fraction'은 7자리 정수가 됩니다. 'In'이 LREAL 형식인 경우, 'Fraction'은 15자리 정수가 됩니다.

'In'이 REAL 형식이고 값이 REAL#-123.456인 경우의 예가 아래에 나와 있습니다.



'In'이 LREAL 형식이고 값이 LREAL#-123.456789인 경우의 예가 아래에 나와 있습니다.



'Fraction'의 유효 범위는 'In'과 'Fraction'의 데이터 형식의 조합으로 다음과 같이 변합니다.

'In'의 데이터 형식	'Fraction'의 데이터 형식	'Fraction'의 유효 범위
REAL	DINT	-9999999~9999999
LREAL	LINT	-999999999999999~999999999999999

참고

부호가 있는 가수부와 지수부를 결합하여 실수를 취득하려는 경우에는 'UniteReal 명령(2-425페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 값에 따라 정수로 변환할 때 오차가 발생하는 경우가 있습니다.
- 'In'의 유효 자릿수가 'Fraction'의 유효 자릿수보다 큰 경우, 'Fraction'의 유효 범위 내에 들어가도록 오사오입합니다. 오사오입이란 다음과 같은 처리를 말합니다.

소수부의 값	처리	예
0.5보다 작음	버림	1.49→1 -1.49→-1

소수부의 값	처리	예
0.5	1의 자리의 값이 짝수이면 버리고 홀수이면 올림	1.50→2 2.50→2 -1.50→-2 -2.50→-2
0.5보다 큼	올림	1.51→2 -1.51→-2

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Fraction', 'Exponent'는 변화하지 않습니다.
  - 'In'의 값이 비수 또는 무한대인 경우

UniteReal

부호가 있는 가수부와 지수부를 결합하여 실수로 만듭니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
UniteReal	가수 · 지수를 실수로 결합	FUN	(@)UniteReal EN ENO Fraction Exponent Out	Out:=UniteReal(Fraction, Exponent);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Fraction	부호가 있는 가수부	입력	부호가 있는 가수부	데이터 형식에 따름	–	(*)
Exponent	지수부		지수부			0
Out	실수	출력	실수	데이터 형식에 따름	–	–

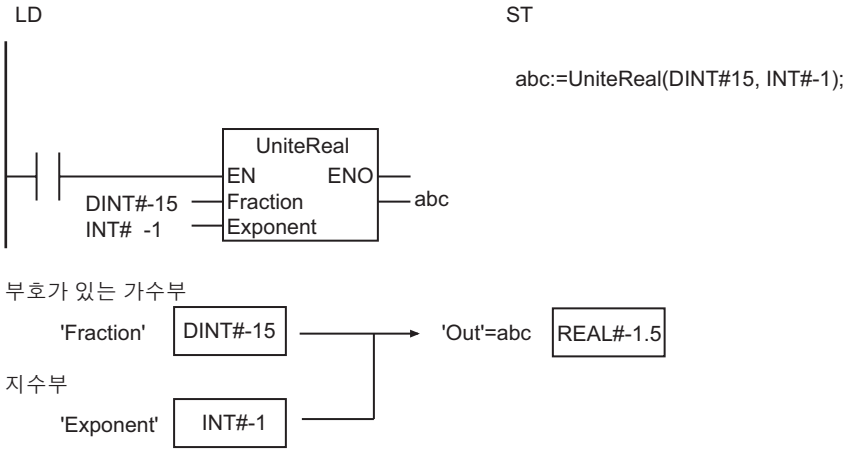
* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Fraction												○	○							
Exponent											○									
Out	'Fraction'의 데이터 형식이 DINT이면 REAL, LINT이면 LREAL																			

기능

부호가 있는 가수부 'Fraction'과 지수부 'Exponent'를 결합하여 실수 'Out'을 취득합니다.

'Fraction'=DINT#-15, 'Exponent'=INT#-1인 경우의 예가 아래에 나와 있습니다.



참고

실수를 부호가 있는 가수부와 지수부로 분리하려는 경우에는 'DispartReal 명령(2-422페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Fraction'과 'Exponent'의 값에 따라 정수에서 실수로 변환할 때 오차가 발생하는 경우가 있습니다.
- 결합 결과가 'Out'의 유효 범위 밖인 경우, 'Exponent'가 양수이면 'Out'의 값은 'Fraction'과 같은 부호를 지닌 무한대가 됩니다. 'Exponent'가 음수이면 'Out'의 값은 0이 됩니다.

NumToDecString / NumToHexString

NumToDecString : 정수를 고정 길이 10진수 문자열 표현으로 변환합니다.
NumToHexString : 정수를 고정 길이 16진수 문자열 표현으로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
NumToDecString	고정 길이 10진수 문자열 변환	FUN	(@)NumToDecString EN ENO In Out L Fill	Out:=NumToDecString(In, L, Fill);
NumToHexString	고정 길이 16진수 문자열 변환	FUN	(@)NumToHexString EN ENO In Out L Fill	Out:=NumToHexString(In, L, Fill);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	정수	입력	정수	데이터 형식에 따름	–	(*)
L	문자 수		'Out'의 문자 수	0~1985		1
Fill	부가 문자		부가 문자	_BLANK, _ZERO		_BLANK
Out	문자열	출력	문자열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
L							○													
Fill	열거 형식 _eFILL_CHR 열거자는 기능 설명 참조																			
Out																				○

기능

● NumToDecString

정수 'In'을 UTF-8 반각 영숫자의 10진수 표기 문자열로 변환합니다. 'In'이 음수인 경우에는 앞에 '-'(마이너스 부호)를 붙입니다.

● NumToHexString

정수 'In'을 UTF-8 반각 영숫자의 16진수 표기 문자열로 변환합니다. 'In'이 음수인 경우에는 2의 보수로 표현(비트 반전+1)합니다.

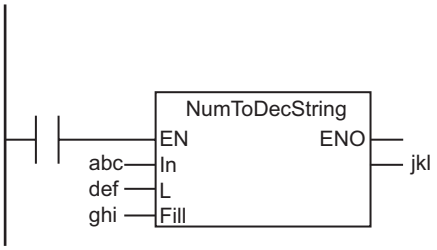
두 명령 모두에서 문자열 'Out'의 문자 수가 문자 수 'L'이 되도록 합니다. 문자 수가 부족한 경우에는 부가 문자 'Fill'로 지정된 문자를 상위 자리에 부가합니다. 변환 결과의 문자 수가 'L'보다 많은 경우, 변환 결과의 최하위 자리에서 'L' 문자만큼을 'Out'에 대입합니다.
'Out'의 끝에는 NULL 문자를 부가합니다. NULL 문자는 문자 수에 포함되지 않습니다.

'Fill'의 데이터 형식은 열거 형식 _eFILL_CHR입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_BLANK	' '(반각 공백 문자)
_ZERO	'0'

NumToDecString 명령의 몇 가지 예가 아래에 나와 있습니다.

LD



ST

```
jkl:=NumToDecString(abc, def, ghi);
```

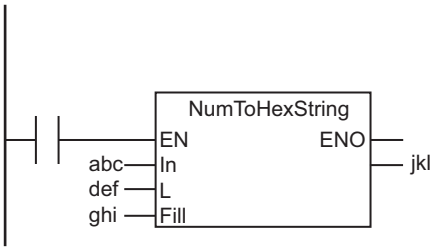
'In'=abc=INT#128, 'L'=def=UINT#8, 'Fill'=ghi=_BLANK
'Out'=jkl □□□□□128

'In'=abc=INT#-128, 'L'=def=UINT#8, 'Fill'=ghi=_BLANK
'Out'=jkl □□□□-128

'In'=abc=INT#-128, 'L'=def=UINT#8, 'Fill'=ghi=_ZERO
'Out'=jkl -0000128

NumToHexString 명령의 몇 가지 예가 아래에 나와 있습니다.

LD



ST

```
jkl:=NumToHexString(abc, def, ghi);
```

'In'=abc=INT#128, 'L'=def=UINT#8, 'Fill'=ghi=_BLANK
'Out'=jkl □□□□□80

'In'=abc=INT#128, 'L'=def=UINT#8, 'Fill'=ghi=_ZERO
'Out'=jkl 00000080

'In'=abc=INT#-128, 'L'=def=UINT#8, 'Fill'=ghi=_BLANK
'Out'=jkl FFFFFF80

사용상의 주의 사항

- 'L'의 값이 0인 경우, 'Out'의 값은 변화하지 않습니다.
- 변환 결과의 문자 수가 'L'의 값보다 많은 경우, 변환 결과의 하위에서 'L'자리만큼을 'Out'에 저장합니다. 다음은 그 예를 나타낸 것입니다.

명령	'In'의 값	'L'의 값	'Out'의 값
NumToDecString	128	2	28
NumToHexString			80

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'L'의 값이 유효 범위 밖인 경우
  - 'Fill'의 값이 유효 범위 밖인 경우

HexStringToNum_**

16진수 문자열 표현을 정수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
HexStringTo Num_**	16진수 문자열→ 수치 변환 그룹	FUN	(@)HexStringToNum_** EN ENO In Out 단, **은 정수의 데이터 형식명	Out: =HexStringToNum_**(In); 단, **는 정수의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	16진수 문자열	입력	16진수 문자열	데이터 형식에 따름	–	"
Out	정수	출력	정수	데이터 형식에 따름	–	–

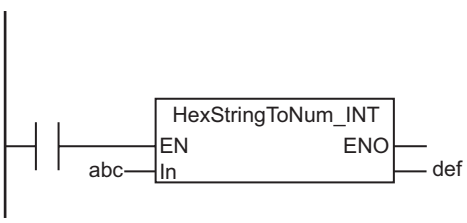
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out						○	○	○	○	○	○	○	○							

기능

16진수 표기의 문자열 'In'을 정수로 변환합니다. 상위 자리의 반각 공백(16#20) 및 '0'(16#30)은 무시합니다. 또한 문자열 중의 밑줄(16#5F)도 무시합니다.
명령 이름은 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 INT 형식인 경우, 명령 이름은 HexStringToNum_INT가 됩니다.

몇 가지 예가 아래에 나와 있습니다.

LD



ST

```
def:=HexStringToNum_INT(abc);
```

'In'=abc 8 0 → 'Out'=def=128

'In'=abc - 8 0 → 'Out'=def=-128

'In'=abc - 0 0 0 0 0 F → 'Out'=def=-15

사용상의 주의 사항

- 변환 결과가 'Out'의 유효 범위 밖이라도 이상 상태가 되지는 않으며, 'Out'에는 잘못된 값이 저장됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 내용이 수치 변환이 불가능한 문자인 경우

FixNumToString

부호가 있는 고정 소수점 수를 10진수 문자열 표현으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FixNumTo String	고정 소수점 수 →문자열 변환	FUN	(@)FixNumToString EN ENO In Out Zero	Out:=FixNumToString(In, Zero);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	고정 소수점 수	입력	부호가 있는 고정 소수점 수	데이터 형식에 따름	–	0
Zero	제로 표시		소수점 이하의 자릿수가 3 미만인 경우의 표시 TRUE : '0'을 채움 FALSE : '0'을 채우지 않음			TRUE
Out	10진수 문자열	출력	10진수 문자열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In				○																
Zero	○																			
Out																				○

기능

부호가 있는 고정 소수점 수 'In'을 10진수 문자열로 변환합니다. 변환 순서는 다음과 같습니다.

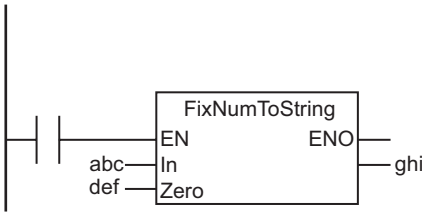
1 **16진수 표현인 'In'을 10진수 표현으로 고칩니다.**

2 **그 값을 1,000으로 나눕니다.**

제로 표시 'Zero'는 'In'의 소수점 이하 자릿수가 3 미만인 경우에 'Out'의 소수점 이하 3번째 자리까지 '0'을 채울지 지정합니다. 'Zero'의 값이 TRUE이면 '0'을 채웁니다.
'Out'의 끝에는 NULL 문자를 부가합니다.

몇 가지 예가 아래에 나와 있습니다.

LD



ST

```
ghi:=FixNumToString(abc, def);
```

'In'=abc	'Out'=ghi	
	'Zero'=def=TRUE	'Zero'=def=FALSE
16#0001462C (10#83500)	'83.500'	'83.5'
16#00051AA4 (10#334500)	'334.500'	'334.5'
16#0003BEFC (10#245500)	'245.500'	'245.5'

참고

고정 소수점 형식은 오므론의 시각 센서 FZ 시리즈의 고정 소수점 출력 형식과 공통입니다.

StringToFixNum

10진수 문자열을 부호가 있는 고정 소수점 수 표현으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StringToFix Num	문자열→고정 소수점 수 변환	FUN	(@)StringToFixNum EN ENO In Out	Out:=StringToFixNum(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	10진수 문자열	입력	10진수 문자열	데이터 형식에 따름	–	''
Out	고정 소수점 수	출력	고정 소수점 수	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out				○																

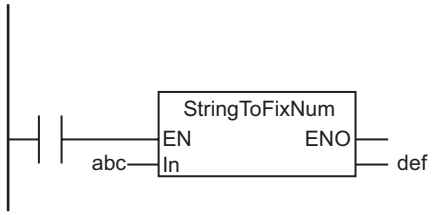
기능

10진수 문자열 'In'을 고정 소수점 수로 변환합니다. 변환 순서는 다음과 같습니다.

1. **'In'이 표현하는 수치를 1,000배합니다.**
2. **그 값의 소수점 이하를 버립니다.**
3. **그 값을 32비트의 16진수 표현(DWORD 형식)으로 합니다.**

몇 가지 예가 아래에 나와 있습니다.

LD



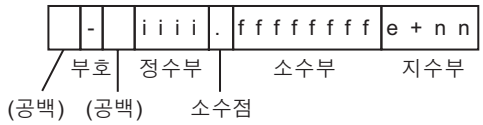
ST

```
def:=StringToFixNum(abc);
```

'In'=abc	'Out'=def
'83.5'	16#0001462C (10#83500)
'334.5'	16#00051AA4 (10#334500)

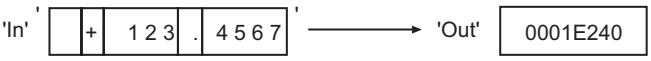
'In'=abc	'Out'=def
'245.5'	16#0003BEFC (10#245500)

'In'의 문자열 형식은 다음과 같습니다.

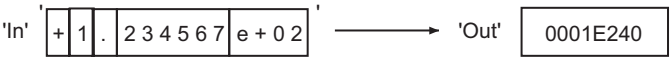


명칭	서식
부호	• 문자열 선두에 나타나는 연속하는 ' '(공백 문자 16#20)를 무시한 후 나타나는 단일 '+' 또는 '-'를 부호로 합니다. • 부호는 생략할 수 있습니다. • 부호 뒤의 연속하는 공백 문자도 무시됩니다.
정수부	• 부호 뒤부터 소수점 앞까지의 연속하는 수치('0'~'9')를 정수부로 합니다. 단, 부호는 생략되는 경우도 있습니다. 또한 부호와 정수부 사이에 연속하는 공백 문자가 들어가는 경우도 있습니다. • 소수점 및 소수부가 없는 경우, 지수부 앞까지를 정수부로 합니다. • 소수점, 소수부, 지수부가 없는 경우에는 문자열 끝까지를 정수부로 합니다. • 정수부는 생략할 수 없습니다. • 정수부의 최대 자릿수는 최대 문자열 길이 1986에서 부호, 소수점, 소수부, 지수부 및 부호 앞뒤의 공백 문자의 바이트 수의 합계를 뺀 수입니다.
소수점	• 정수부에 이어 나오는 단일 '.'(점)을 소수점으로 합니다. • 소수부가 존재하지 않는 경우, 소수점은 생략하십시오.
소수부	• 소수점 뒤부터 지수부 앞까지의 연속하는 수치('0'~'9')를 소수부로 합니다. • 지수부가 없는 경우에는 문자열 끝까지를 소수부로 합니다. • 소수부는 생략할 수 있습니다. 소수점이 없는 경우는 소수부도 존재하지 않습니다. • 소수부의 최대 자릿수는 15입니다.
지수부	• 소수부에 이어 나오는 단일 'e' 또는 'E', 그 뒤에 나오는 단일 '+' 또는 '-'에서 문자열 끝까지의 연속하는 수치('0'~'9')를 지수부로 합니다. • 소수부가 없는 경우에는 소수점 뒤에 나오는 상기 문자열을 지수부로 합니다. • 소수점 및 소수부가 없는 경우에는 정수부 뒤에 나오는 상기 문자열을 지수부로 합니다. • 지수부는 생략할 수 있습니다. • 지수부의 숫자 부분의 최대 자릿수는 3입니다.

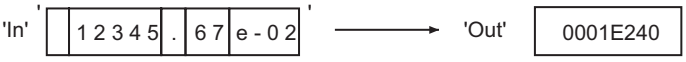
(예 1) 부호 있음, 소수점 있음, 소수부 있음, 지수부 없음



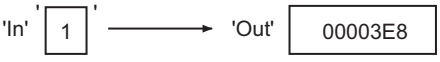
(예 2) 부호 있음, 소수점 있음, 소수부 있음, 지수부 있음



(예 3) 부호 없음, 소수점 있음, 소수부 있음, 지수부 있음



(예 4) 부호 없음, 소수점 없음, 소수부 없음, 지수부 없음



참고

고정 소수점 형식은 오므론의 시각 센서 FZ 시리즈의 고정 소수점 출력 형식과 공통입니다.

사용상의 주의 사항

- 'In'의 소수점 이하 4자리째 다음은 버려집니다.
- 'In'의 문자열 중의 밑줄(16#5F)은 무시합니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 내용이 수치 변환이 불가능한 문자인 경우
  - 'In'의 내용에 소수점은 있고 소수부는 없는 경우

DtToString

일시를 문자열 표현으로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DtToString	일시→문자열 변환	FUN	(@)DtToString EN ENO In Out	Out:=DtToString(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	일시	입력	일시	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	문자열	출력	문자열	30바이트 (반각 영숫자 29문자+끝의 NULL 문자)	–	–

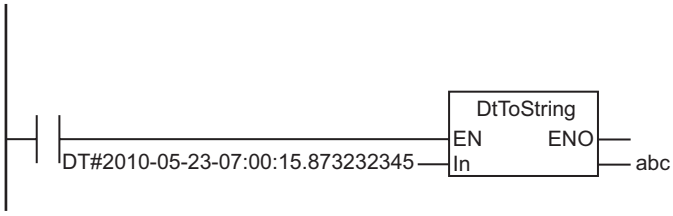
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out																				○

기능

일시 'In'을 문자열로 변환합니다. 문자열 'Out'의 끝에는 NULL 문자를 부가합니다.

'In'의 값이 2010년 5월 23일 7시 0분 15.873232345초인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 '2010-05-23-07:00:15.873232345'가 됩니다.

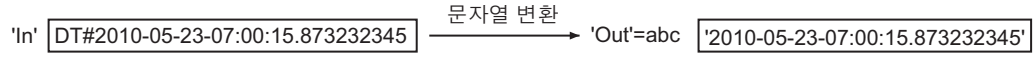
LD



ST

```
abc:=DtToString(DT#2010-05-23-07:00:15.873232345);
```

일시 'In'을 문자열로 변환합니다.
'In'의 값이 2010년 5월 23일 7시 0분 15.873232345초이므로 abc의 값은 '2010-05-23-07:00:15.873232345'가 됩니다.

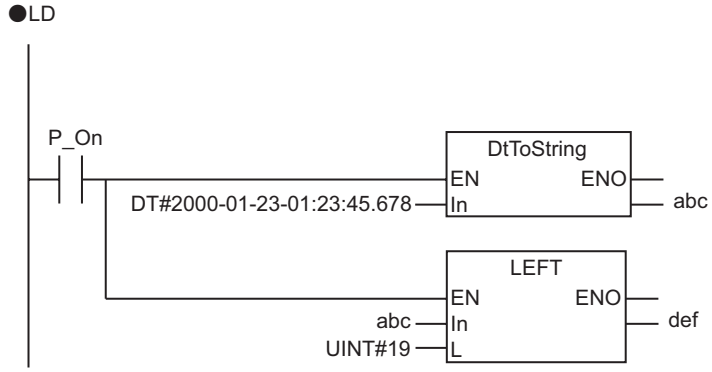


참고

'Out'은 나노초 단위로 표현됩니다. 초 단위 또는 밀리초 단위의 문자열을 취득하려는 경우, 이 명령과 'LEFT / RIGHT 명령(2-528페이지)'을 조합하여 사용하십시오.

초 단위의 문자열을 취득하는 경우의 예가 아래에 나와 있습니다.

●LD



●ST

```
def:=LEFT(DtToString(DT#2000-01-23-01:23:45.678), UINT#19);
```

DateToString

날짜를 문자열 표현으로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DateToString	날짜→문자열 변환	FUN	(@)DateToString EN ENO In Out	Out:=DateToString(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜	입력	날짜	데이터 형식에 따름	연월일	D#1970-1-1
Out	문자열	출력	문자열	11바이트 (반각 영숫자 10문자+끝의 NULL 문자)	–	–

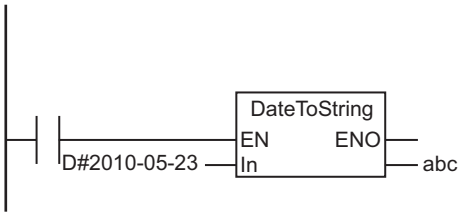
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																	○			
Out																				○

기능

날짜 'In'을 문자열로 변환합니다. 'Out'의 끝에는 NULL 문자를 부가합니다.

'In'의 값이 2010년 5월 23일인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 '2010-05-23'이 됩니다.

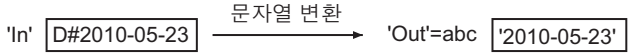
LD



ST

```
abc:=DateToString(D#2010-05-23);
```

날짜 'In'을 문자열로 변환합니다.
'In'의 값이 2010년 5월 23일이므로 abc의 값은 '2010-05-23'이 됩니다.



TodToString

시각을 문자열 표현으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TodToString	시각→문자열 변환	FUN	(@)TodToString EN ENO In Out	Out:=TodToString(In);

변수

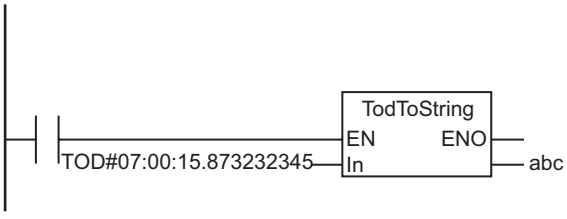
	명칭	입/출력	내용	유효 범위	단위	초기값
In	시각	입력	시각	데이터 형식에 따름	시분초	TOD#0 :0:0
Out	문자열	출력	문자열	19바이트 (반각 영숫자 18 문자+끝의 NULL 문자)	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																		○		
Out																				○

기능

시각 'In'을 문자열로 변환합니다. 'Out'의 끝에는 NULL 문자를 부가합니다.
'In'의 값이 7시 0분 15.873232345초인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 '07:00:15.873232345'가 됩니다.

LD



ST

```
abc:=TodToString(TOD#07:00:15.873232345);
```

시각 'In'을 문자열로 변환합니다.
'In'의 값이 7시 0분 15.873232345초이므로 abc의 값은 '07:00:15.873232345'가 됩니다.

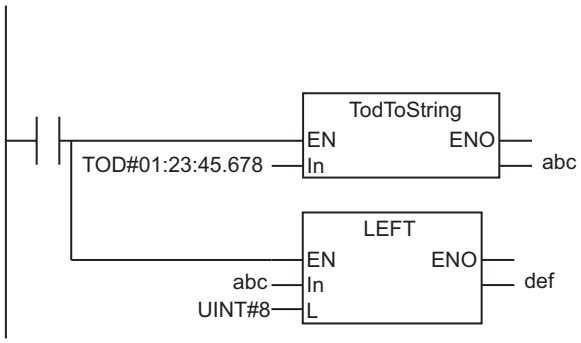
'In' TOD#07:00:15.873232345 → 문자열 변환 → 'Out'=abc '07:00:15.873232345'

참고

'Out'은 나노초 단위로 표현됩니다. 초 단위 또는 밀리초 단위의 문자열을 취득하려는 경우, 이 명령과 'LEFT / RIGHT 명령(2-528페이지)'을 조합하여 사용하십시오.

초 단위의 문자열을 취득하는 경우의 예가 아래에 나와 있습니다.

●LD



●ST

```
def:=LEFT(TodToString(TOD#01:23:45.678), UINT#8);
```

GrayToBin_** / BinToGray_**

GrayToBin_** : 그레이 코드를 비트 열로 변환합니다.
BinToGray_** : 비트 열을 그레이 코드로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GrayToBin_**	그레이 코드→ BIN 코드 변환 그룹	FUN	(@)GrayToBin_** EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out: =GrayToBin_**(In); 단, **는 비트 열의 데이터 형식명
BinToGray_**	BIN 코드→ 그레이 코드 변환 그룹	FUN	(@)BinToGray_** EN ENO In Out 단, **는 비트 열의 데이터 형식명	Out: =BinToGray_**(In); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	0
Out	변환 결과	출력	변환 결과	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Out	'In'과 동일한 데이터 형식																			

기능

● GrayToBin_**

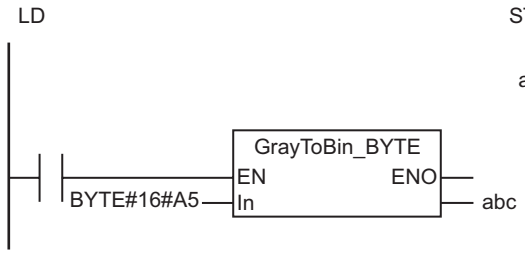
그레이 코드의 변환 대상 'In'을 비트 열로 변환합니다. 변환 순서를 'In', 'Out'이 BYTE 형식인 경우로 설명합니다.

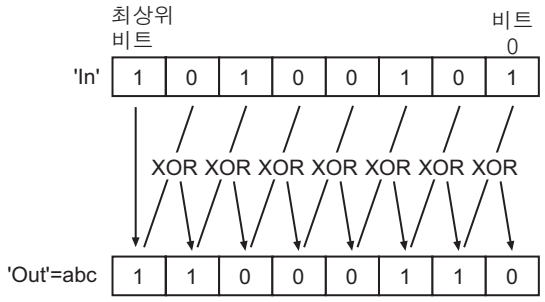
1 **'In'의 최상위 비트(비트 7)를 그대로 'Out'의 최상위 비트(비트 7)로 합니다.**

2 **'In'의 비트 6의 값과 'Out'의 비트 7의 값의 배타적 논리합을 'Out'의 비트 6으로 합니다.**

3 **이후 마찬가지로 'Out'의 최하위 비트(비트 0)까지의 값을 결정합니다.**

GrayToBin_BYTE 명령에서 'In'=BYTE#16#A5인 경우의 예가 아래에 나와 있습니다.





● BinToGray_**

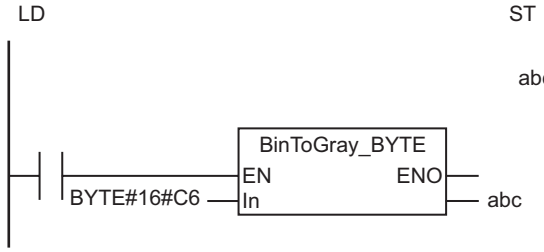
비트 열의 변환 대상 'In'을 그레이 코드로 변환합니다. 변환 순서를 'In', 'Out'이 BYTE 형식인 경우로 설명합니다.

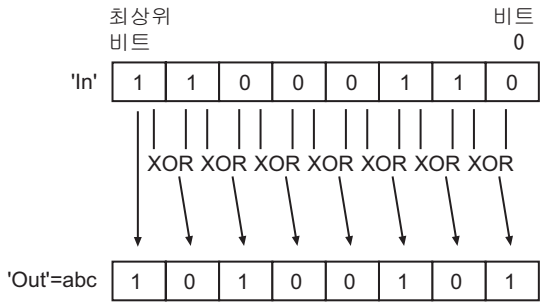
1 **'In'의 최상위 비트(비트 7)를 그대로 'Out'의 최상위 비트(비트 7)로 합니다.**

2 **'In'의 비트 7의 값과 'In'의 비트 6의 값의 배타적 논리합을 'Out'의 비트 6으로 합니다.**

3 **이후 마찬가지로 'Out'의 최하위 비트(비트 0)까지의 값을 결정합니다.**

BinToGray_BYTE 명령에서 'In'=BYTE#16#C6인 경우의 예가 아래에 나와 있습니다.





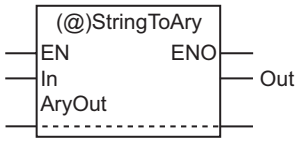
명령 이름은 'In', 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In', 'Out'이 WORD 형식인 경우, 명령 이름은 GrayToBin_WORD 및 BinToGray_WORD가 됩니다.

사용상의 주의 사항

'In'과 'Out'의 데이터 형식을 동일하게 하십시오.

StringToAry

문자열을 BYTE 형식의 배열로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StringToAry	문자열→배열 변환	FUN		Out:=StringToAry(In, AryOut);

변수

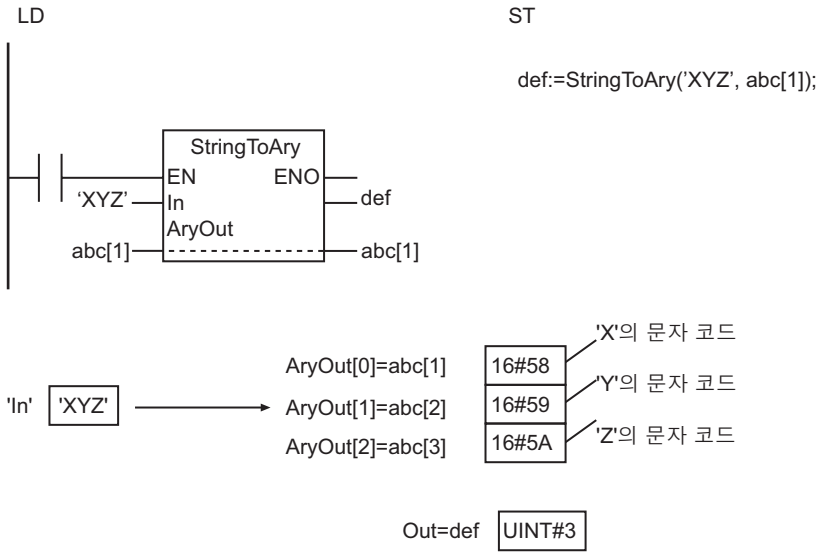
	명칭	입/출력	내용	유효 범위	단위	초기값
In	문자열	입력	문자열	데이터 형식에 따름	–	''
AryOut[] 배열	BYTE 형식 배열	입출력	BYTE 형식 배열	데이터 형식에 따름	–	–
Out	변환한 바이트 수	출력	변환한 바이트 수	0~1985	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
AryOut[] 배열		○																		
Out							○													

기능

문자열 'In'의 문자 코드를 수치로 간주하고 1문자씩 BYTE 형식 배열 AryOut[]에 저장합니다. 'Out'에는 변환한 바이트 수를 저장합니다.

'In'='XYZ'인 경우의 예가 아래에 나와 있습니다.

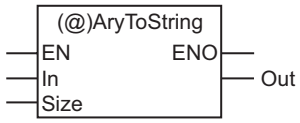


사용상의 주의 사항

- 'In'의 끝에 있는 NULL 문자는 AryOut[]에 저장되지 않습니다.
- 'In'이 NULL 문자만 있는 문자열인 경우, 'Out'의 값은 0이 되며 AryOut[]은 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out' 및 AryOut[]은 변화하지 않습니다.
  - 'In'의 바이트 수가 AryOut[]의 요소 수보다 큰 경우

AryToString

BYTE 형식 배열을 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AryToString	배열→문자열 변환	FUN		Out:=AryToString(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	BYTE 형식 배열	입력	BYTE 형식 배열 요소 수는 최대 1985	데이터 형식에 따름	–	(*)
Size	변환 요소 수		변환할 In[]의 요소 수	0~1985		1
Out	문자열	출력	문자열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

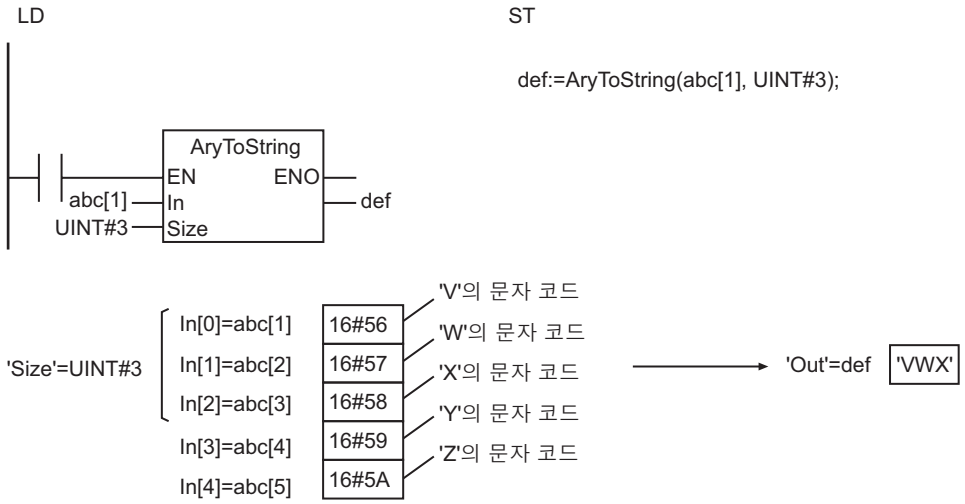
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○																		
Size							○													
Out																				○

기능

BYTE 형식의 배열 In[0] 이후의 각 요소의 값을 문자 코드로 간주하여 문자열 'Out'에 저장합니다. 'Out'의 끝에는 NULL 문자를 부가합니다.

'Size'에는 변환할 In[]의 요소 수를 지정합니다. 단, In[0]에서 In['Size'-1] 사이에 NULL 문자가 있는 경우, 거기까지만 'Out'에 저장합니다.

'Size'=UINT#3인 경우의 예가 아래에 나와 있습니다.

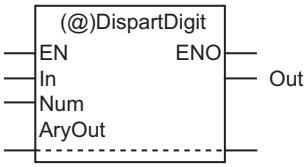


사용상의 주의 사항

- 'Size'의 값이 0인 경우, 'Out'은 NULL 문자만 있는 문자열이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

DispartDigit

비트 열을 4비트 단위로 분리합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DispartDigit	4비트 분리	FUN		DispartDigit(In, Num, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	분리 대상	입력	분리 대상의 비트 열	데이터 형식에 따름	–	(*)
Num	분리 자릿수		분리할 자릿수	0~'In'의 비트 수		1
AryOut[] 배열	분리 결과 배열	입출력	분리 결과 배열	16#00~16#0F	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

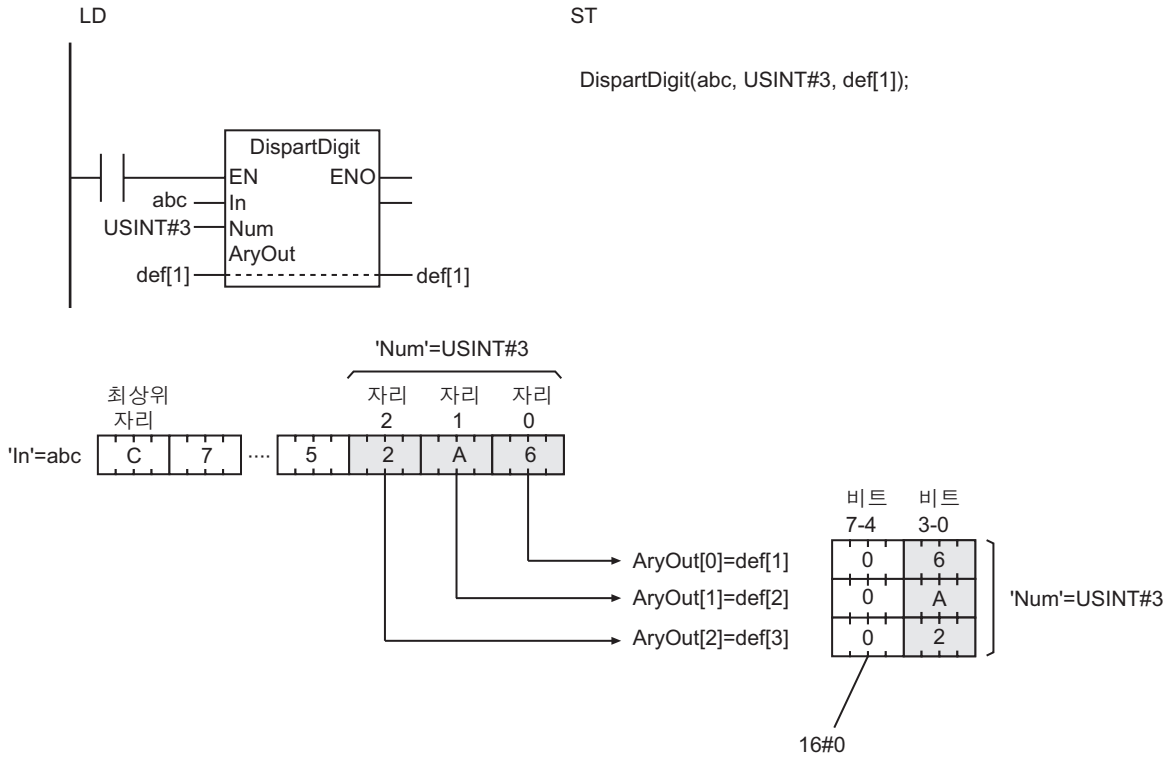
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Num						○														
AryOut[] 배열		○																		
Out	○																			

기능

분리 대상 'In'을 4비트 단위(자리 단위)로 분리하여 분리 결과 배열 AryOut[]에 저장합니다.

우선 'In'을 4비트 단위로 분리합니다. 그리고 최하위 자리의 4비트를 AryOut[0]에 저장합니다. 그때, BYTE 형식인 AryOut[0]의 비트 4에서 비트 7에는 16#0을 부가합니다.
같은 과정을 분리 자릿수 'Num'의 수만큼 수행합니다.

'Num'=USINT#3인 경우의 예가 아래에 나와 있습니다.



참고

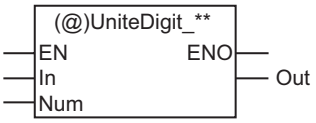
배열의 각 요소를 4비트씩 결합하려는 경우에는 'UniteDigit_** 명령(2-451페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Num'의 값이 0인 경우, AryOut[]의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Num'의 값이 유효 범위 밖인 경우
  - 'Num'의 값이 AryOut[]의 배열 영역을 초과한 경우

UniteDigit_**

4비트 단위의 데이터를 비트 열로 결합합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
UniteDigit_**	4비트 결합 그룹	FUN	 단, **는 비트 열의 데이터 형식명	Out: =UniteDigit_**(In, Num); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	결합 대상 배열	입력	결합 대상의 배열	데이터 형식에 따름	–	(*)
Num	결합 자릿수		결합할 자릿수	0~'Out'의 비트 수		1
Out	결합 결과	출력	결합 결과의 비트 열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○																		
Num						○														
Out		○	○	○	○															

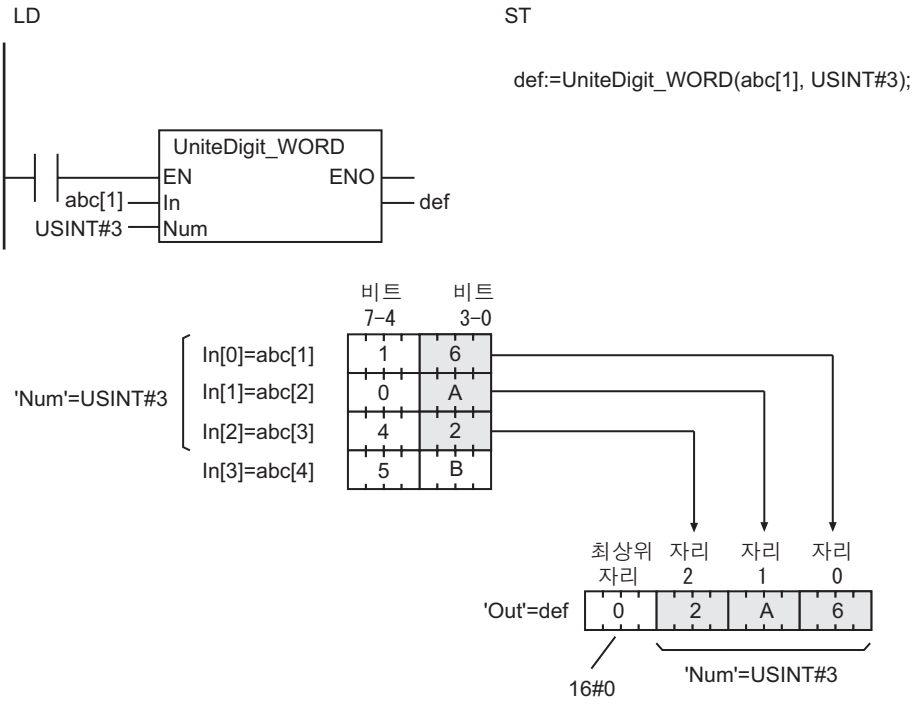
기능

결합 대상 배열 In[]의 각 요소의 하위 4비트씩(1자리=4비트)을 결합하여 결합 결과의 비트 열 'Out'을 작성합니다.

결합할 배열 요소 수는 결합 자릿수 'Num'으로 지정합니다. 우선 In[0]~In['Num'-1]의 하위 4비트씩을 결합하여 'Num'자리의 비트 열을 작성합니다. 그 상위에 {('Out'의 자릿수)-'Num'개}의 16#0을 결합한 것을 'Out'으로 합니다.

명령 이름은 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 WORD 형식인 경우, 명령 이름은 UniteDigit_WORD가 됩니다.

UniteDigit_WORD 명령에서 'Num'=USINT#3인 경우의 예가 아래에 나와 있습니다.



참고

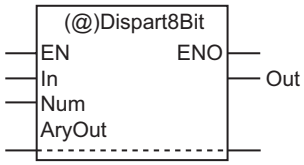
비트 열을 4비트 단위로 분리하려는 경우에는 'DispartDigit 명령(2-449페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Num'의 값이 0인 경우, 'Out'의 값은 0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Num'의 값이 유효 범위 밖인 경우
  - 'Num'의 값이 In[]의 배열 영역을 초과한 경우

Dispart8Bit

비트 열을 1바이트 단위로 분리합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Dispart8Bit	바이트 단위 데이터 분리	FUN		Dispart8Bit(In, Num, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	분리 대상	입력	분리 대상의 비트 열	데이터 형식에 따름	–	(*)
Num	분리 바이트 수		분리할 바이트 수	0~'In'의 바이트 수		1
AryOut[] 배열	분리 결과 배열	입출력	분리 결과 배열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

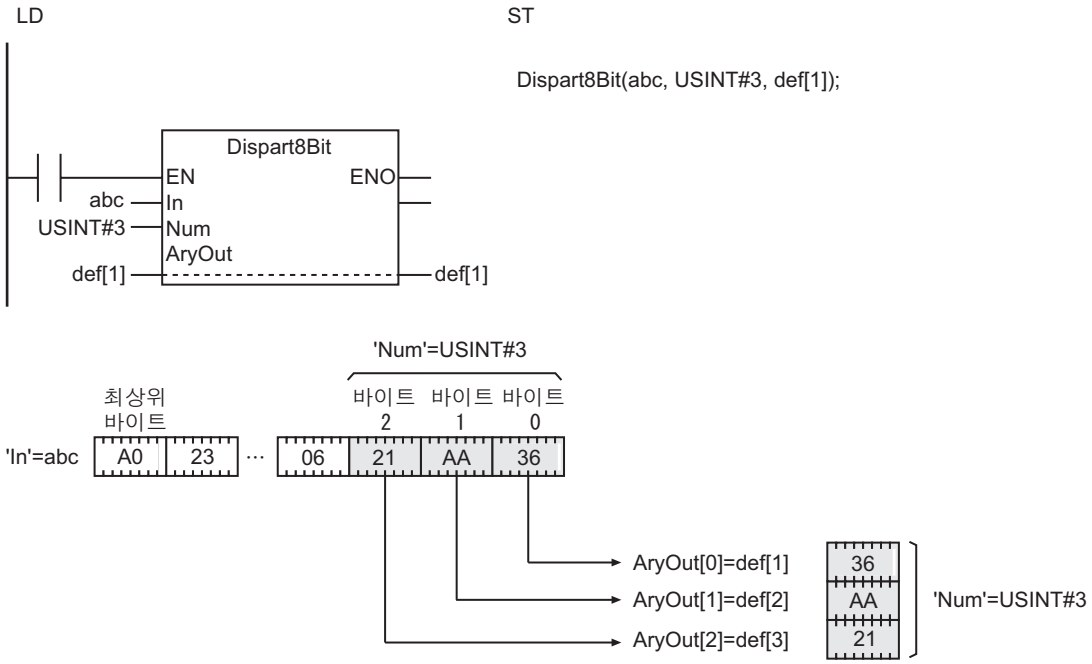
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Num						○														
AryOut[] 배열		○																		
Out	○																			

기능

분리 대상 'In'을 1바이트 단위로 분리하여 분리 결과 배열 AryOut[]에 저장합니다.

우선 'In'을 1바이트 단위로 분리합니다. 그리고 최하위 바이트를 AryOut[0]에 저장합니다. 또한 다음 1바이트를 마찬가지로 AryOut[1]에 저장합니다. 같은 과정을 분리 바이트 수 'Num'의 수만큼 수행합니다.

'Num'=USINT#3인 경우의 예가 아래에 나와 있습니다.



참고

배열의 각 요소를 1바이트씩 결합하려는 경우에는 'Unite8Bit_** 명령(2-455페이지)'을 사용하십시오.

사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 AryOut[]은 변화하지 않습니다.
  - 'Num'의 값이 유효 범위 밖인 경우
  - 'Num'의 값이 AryOut[]의 배열 영역을 초과한 경우

Unite8Bit_**

1바이트 단위의 데이터를 비트 열로 결합합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Unite8Bit_**	바이트 단위 데이터 결합 그룹	FUN	 단, **는 비트 열의 데이터 형식명	Out: =Unite8Bit_**(In, Num); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	결합 대상 배열	입력	결합 대상의 배열	데이터 형식에 따름	–	(*)
Num	결합 바이트 수		결합할 바이트 수	0~'Out'의 바이트 수		1
Out	결합 결과	출력	결합 결과의 비트 열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○																		
Num						○														
Out		○	○	○	○															

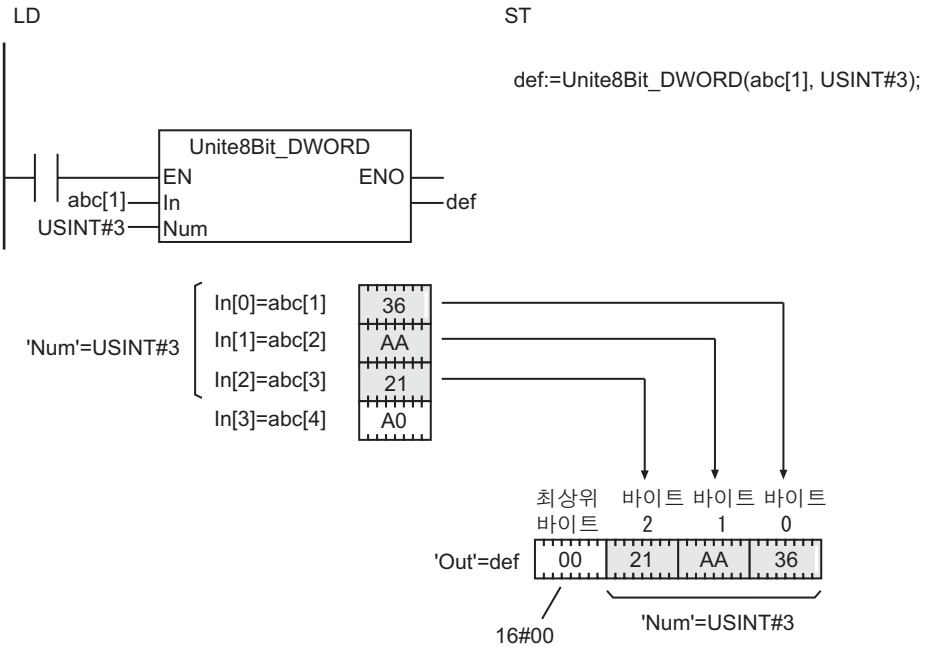
기능

결합 대상 배열 In[]의 각 요소를 결합하여 결합 결과의 비트 열 'Out'을 작성합니다.

결합할 배열 요소 수는 결합 바이트 수 'Num'으로 지정합니다. 우선 In[0]~In['Num'-1]을 결합하여 'Num' 바이트의 비트 열을 작성합니다. 그 상위에 {('Out'의 바이트 수)-'Num'개}의 16#00을 결합한 것을 'Out'으로 합니다.

명령 이름은 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'Out'이 DWORD 형식인 경우, 명령 이름은 Unite8Bit_DWORD가 됩니다.

Unite8Bit_DWORD 명령에서 'Num'=USINT#3인 경우의 예가 아래에 나와 있습니다.



참고

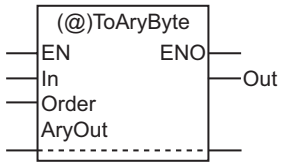
비트 열을 1바이트 단위로 분리하려는 경우에는 'Dispart8Bit 명령(2-453페이지)'을 사용하십시오.

사용상의 주의 사항

- 'Num'의 값이 0인 경우, 'Out'의 값은 0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Num'의 값이 유효 범위 밖인 경우
  - 'Num'의 값이 In[]의 배열 영역을 초과한 경우

ToAryByte

변수를 1바이트 단위로 분할하여 BYTE 형식 배열에 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ToAryByte	바이트 배열로 변환	FUN		Out:=ToAryByte(In, Order, AryOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상	데이터 형식에 따름	–	(*)
Order	변환 순서		변환하는 순서	_LOW_HIGH, _HIGH_LOW		_LOW _HIGH
AryOut[] 배열	변환 결과 배열	입출력	변환 결과 배열	데이터 형식에 따름	–	–
Out	변환 결과 요소 수	출력	변환 결과의 요소 수	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Order	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
AryOut[] 배열		○																		
Out							○													

기능

변환 대상 'In'을 1바이트 단위로 분할하여 변환 결과 배열 AryOut[]의 AryOut[0]부터 순서대로 저장합니다. 그리고 AryOut[]에 저장한 요소 수를 변환 결과 요소 수 'Out'에 저장합니다.

'In'의 값을 1바이트 단위로 변환할 때의 순서는 변환 순서 'Order'로 지정합니다. 'Order'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

'In'의 데이터 형식이 2바이트 이상의 크기인 경우

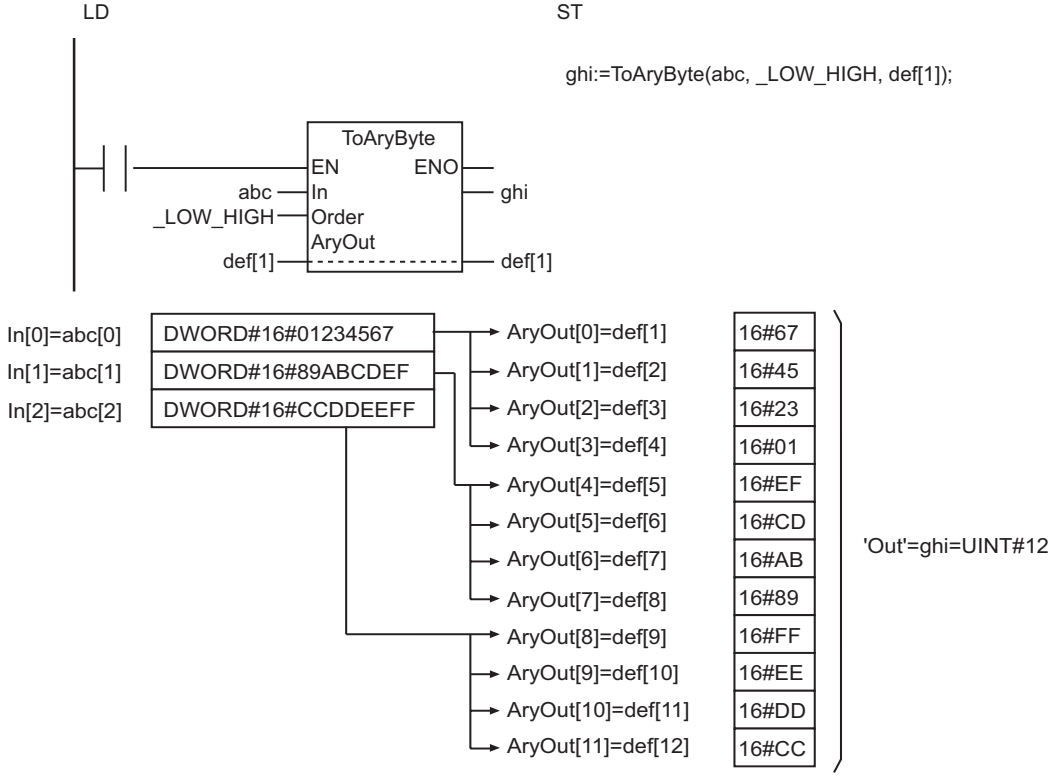
'In'의 데이터 형식이 2바이트 이상의 크기인 경우, 'In'을 1바이트 단위로 분할하여 AryOut[]에 저장합니다.
2바이트 이상 크기의 데이터 형식은 다음과 같습니다.

종류	데이터 형식
비트 열	WORD, DWORD, LWORD
정수	UINT, UDINT, ULINT, INT, DINT, LINT
실수	REAL, LREAL
시각, 지속 시간, 날짜, 문자열	TIME, DATE, TOD, DT, 2바이트 이상의 STRING
기타	열거 형식, 모든 요소의 합계 크기가 2바이트 이상인 배열 전체, 배열에서 2바이트 이상인 1개 요소 모든 멤버의 합계 크기가 2바이트 이상인 구조체 전체, 구조체에서 2바이트 이상인 1개 멤버

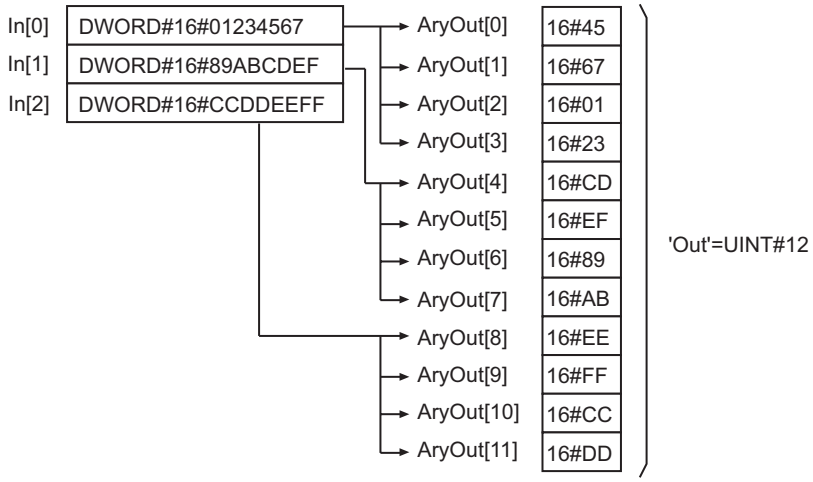
처리 순서는 다음과 같습니다.

1. **'In'의 값을 우선 1워드(2바이트) 단위로 분할합니다.**
2. **그 중 최하위 워드(2바이트)를 다시 1바이트 단위로 분할합니다.**
3. **'Order'의 값이 _LOW_HIGH인 경우 그 중의 하위 바이트를 AryOut[0]에, 상위 바이트를 AryOut[1]에 저장합니다. 'Order'의 값이 _HIGH_LOW인 경우 상위 바이트를 AryOut[0]에, 하위 바이트를 AryOut[1]에 저장합니다.**
4. **마찬가지로, 다음 1워드를 1바이트 단위로 분할하여 AryOut[2]와 AryOut[3]에 저장합니다.**
5. **같은 과정을 'In'의 값의 마지막까지 수행합니다. 'In'이 배열인 경우에는 같은 과정을 'In'의 마지막 요소까지 수행합니다.**

'In'이 DWORD 형식이고 요소 수 3의 배열 전체, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



'In'이 상기와 동일하고 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



'In'의 데이터 형식이 1바이트 크기인 경우

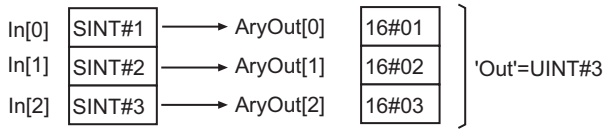
'In'의 데이터 형식이 1바이트 크기인 경우, 'In'을 1바이트 단위 그대로 AryOut[]에 저장합니다.
1바이트 크기의 데이터 형식은 다음과 같습니다.

종류	데이터 형식
비트 열	BYTE
정수	USINT, SINT
실수	없음
시각, 지속 시간, 날짜, 문자열	1바이트의 STRING
기타	모든 요소의 합계 크기가 1바이트인 배열 전체, 배열에서 1바이트인 1개 요소 모든 멤버의 합계 크기가 1바이트인 구조체 전체, 구조체에서 1바이트인 1개 멤버

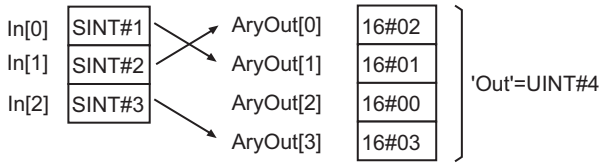
저장 방법은 다음과 같습니다.

'Order'의 값	'In'이 배열인지의 여부	AryOut[]에 저장하는 방법
_LOW_HIGH	배열이 아님	'In'의 값을 AryOut[0]에 저장합니다.
	배열임	In[i]의 값을 AryOut[i]에 저장합니다.
_HIGH_LOW	배열이 아님	'In'의 값을 AryOut[1]에 저장합니다. 16#00을 AryOut[0]에 저장합니다.
	배열임	In[i](i는 짝수)를 AryOut[i+1]에 저장합니다. In[i](i는 홀수)를 AryOut[i-1]에 저장합니다. In[]의 요소 수 n이 홀수이면 마지막에 AryOut[n-1]에 16#00을 저장합니다.

'In'이 SINT 형식이고 요소 수 3의 배열, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



'In'이 상기와 동일하고 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.

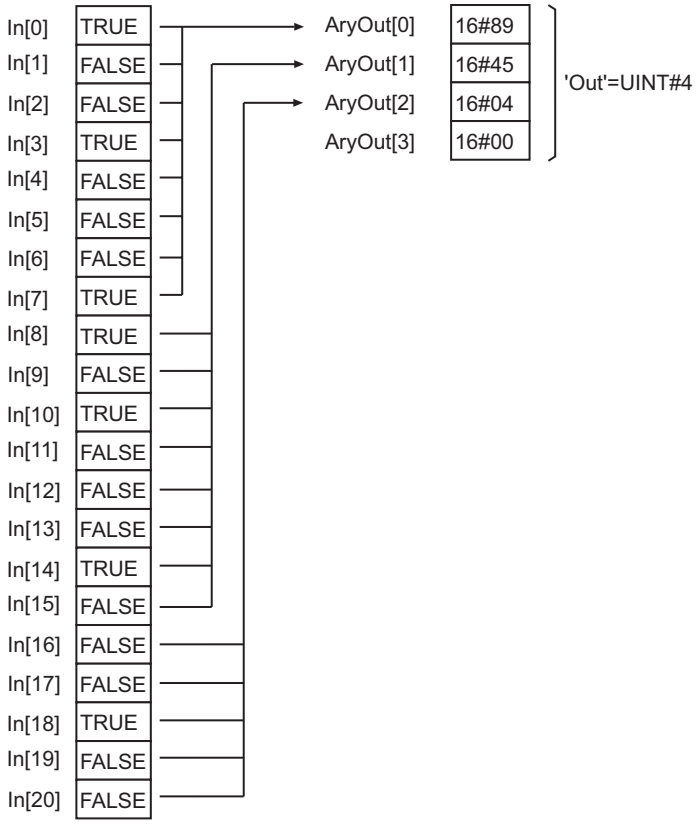


'In'의 데이터 형식이 BOOL인 경우

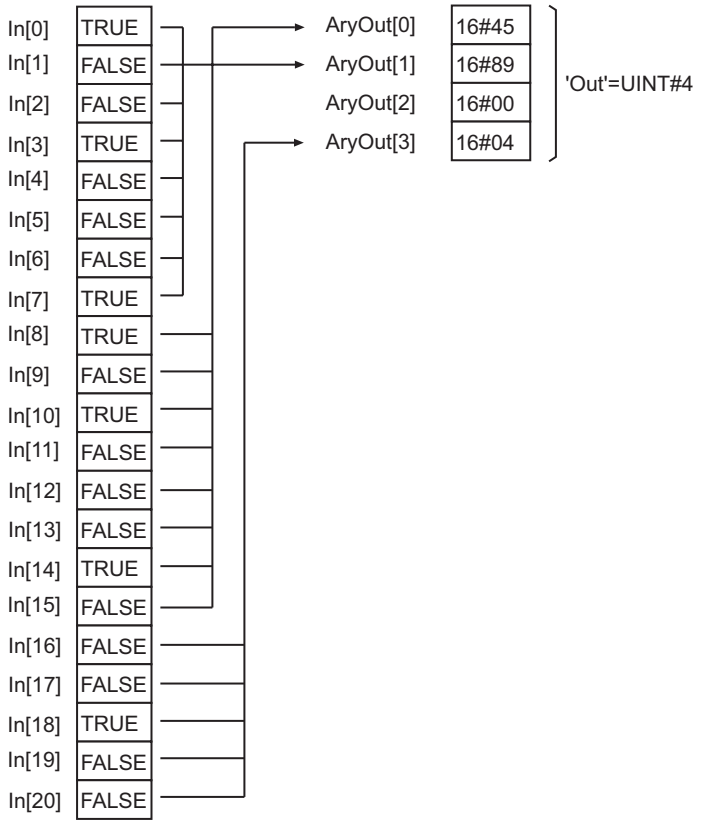
'In'의 데이터 형식이 BOOL(1비트)인 경우, AryOut[]에 저장하는 방법은 다음과 같습니다.

'Order'의 값	'In'이 배열인지의 여부	AryOut[]에 저장하는 방법
_LOW_HIGH	배열이 아님	'In'의 값과 16#00의 논리합을 AryOut[0]에 저장합니다.
	배열임	In[0]~In[7]의 값을 연결하여 AryOut[0]에 저장합니다. In[8]~In[15]의 값을 연결하여 AryOut[1]에 저장합니다. 이후 마찬가지로 저장합니다. In[]이 8개에 못 미치는 경우, 상위 비트에 FALSE를 부가합니다. 'Out'의 값은 반드시 짝수로 합니다. 부족한 비트의 값은 모두 FALSE가 됩니다.
_HIGH_LOW	배열이 아님	'In'의 값과 16#00의 논리합을 AryOut[1]에 저장합니다. 16#00을 AryOut[0]에 저장합니다.
	배열임	In[0]~In[7]의 값을 연결하여 AryOut[1]에 저장합니다. In[8]~In[15]의 값을 연결하여 AryOut[0]에 저장합니다. 이후 마찬가지로 저장합니다. 'Out'의 값은 반드시 짝수가 됩니다. 부족한 비트의 값은 모두 FALSE가 됩니다.

'In'이 BOOL 형식이고 요소 수 21의 배열, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



'In'이 상기와 동일하고 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 'In'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'In'이 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 'In'이 STRING 형식인 경우, 문자열을 수치로 변환하지 않습니다. 변수의 내용을 비트 열로 간주하여 바이트 배열로 변환합니다.
- 'In'이 구조체인 경우, 구성에 따라 각 멤버 사이에 있는 조정용 영역이 AryOut[]에 삽입되는 경우가 있습니다.
- 'Order'의 값이 _HIGH_LOW이고 'In'의 합계 크기가 홀수 바이트인 경우, 'In'의 마지막에 16#00을 부가하여 짝수 바이트로 만든 후 변환을 시작합니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out' 및 AryOut[]은 변화하지 않습니다.
  - 'Order'의 값이 유효 범위 밖인 경우
  - 변환 결과가 AryOut[]의 배열 영역을 초과한 경우

AryByteTo

BYTE 형식 배열 요소를 결합하여 변수에 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryByteTo	바이트 배열에서 변환	FUN	(@)AryByteTo EN ENO In Out Size Order OutVal	AryByteTo(In, Size, Order, OutVal);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	변환 대상 배열	입력	변환 대상의 배열	데이터 형식에 따름	–	(*)
Size	변환 요소 수		In[]의 변환할 요소 수			1
Order	변환 순서		변환하는 순서	_LOW_HIGH, _HIGH_LOW		_LOW _HIGH
OutVal	변환 결과	입출력	변환 결과	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○																		
Size							○													
Order	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
OutVal	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Out	○																			

기능

변환 대상 배열 In[]의 선두의 'Size'개의 요소를 변환 결과 'OutVal'의 데이터 형식 크기에 맞춰 결합하여 'OutVal'에 저장합니다.

In[]의 요소를 결합할 때의 순서는 변환 순서 'Order'로 지정합니다. 'Order'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

'OutVal'의 데이터 형식이 2바이트 이상의 크기인 경우

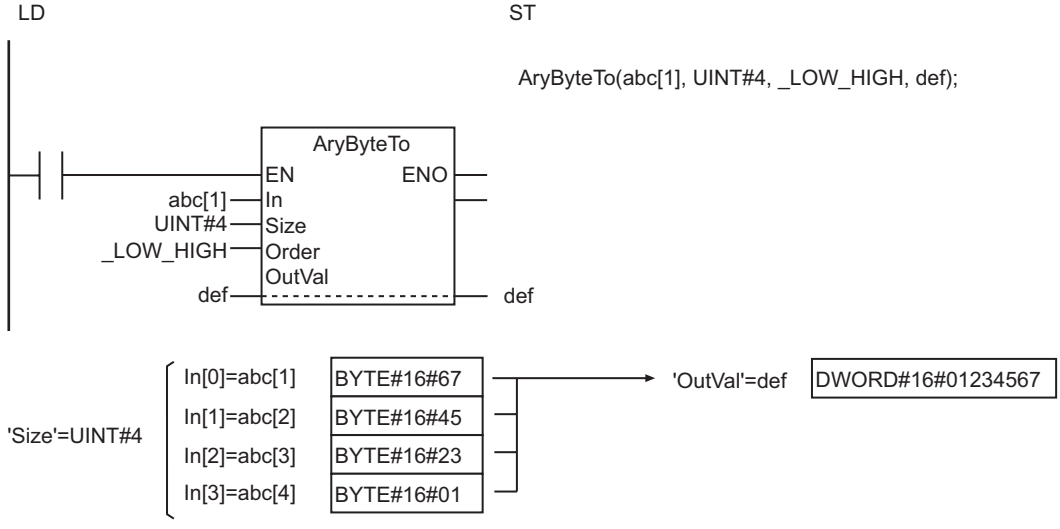
'OutVal'의 데이터 형식이 2바이트 이상의 크기인 경우, In[]을 'OutVal'의 데이터 형식의 크기만큼 결합하여 'OutVal'에 저장합니다.
2바이트 이상 크기의 데이터 형식은 다음과 같습니다.

종류	데이터 형식
비트 열	WORD, DWORD, LWORD
정수	UINT, UDINT, ULINT, INT, DINT, LINT
실수	REAL, LREAL
시각, 지속 시간, 날짜, 문자열	TIME, DATE, TOD, DT, 2바이트 이상의 STRING
기타	열거 형식, 모든 요소의 합계 크기가 2바이트 이상인 배열 전체, 배열에서 2바이트 이상인 1개 요소 모든 멤버의 합계 크기가 2바이트 이상인 구조체 전체, 구조체에서 2바이트 이상인 1개 멤버

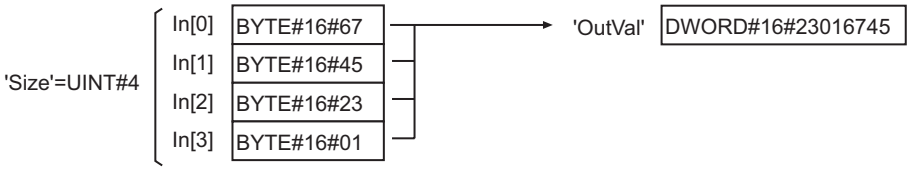
처리 순서는 다음과 같습니다.

1. **In[0]과 In[1]을 'Order'의 값에 따라 결합하여 1워드(2바이트)의 데이터를 만듭니다. 'Order'의 값이 _LOW_HIGH인 경우 상위 바이트가 In[1], 하위 바이트가 In[0]이 됩니다. 'Order'의 값이 _HIGH_LOW인 경우 상위 바이트가 In[0], 하위 바이트가 In[1]이 됩니다.**
2. **마찬가지로 In[2]와 In[3] 이후도 결합하여 1워드의 데이터를 복수 작성합니다.**
3. **작성된 복수의 1워드의 데이터를 'OutVal'의 데이터 형식의 크기에 맞게 다시 결합합니다. 예를 들면, 'OutVal'이 DWORD 형식인 경우, 1워드의 데이터를 4개 결합합니다.**
4. **작성된 데이터를 'OutVal'에 저장합니다.**

'OutVal'이 DWORD 형식, 'Size'=UINT#4, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



'OutVal'이 상기와 동일하고 'Size'=UINT#4, 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



'OutVal'의 데이터 형식이 1바이트 크기인 경우

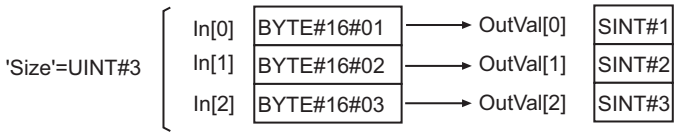
'OutVal'의 데이터 형식이 1바이트 크기인 경우, In[]을 1바이트 단위 그대로 'OutVal'에 저장합니다.
1바이트 크기의 데이터 형식은 다음과 같습니다.

종류	데이터 형식
비트 열	BYTE
정수	USINT, SINT
실수	없음
시각, 지속 시간, 날짜, 문자열	1바이트의 STRING
기타	모든 요소의 합계 크기가 1바이트인 배열 전체, 배열에서 1바이트인 1개 요소 모든 멤버의 합계 크기가 1바이트인 구조체 전체, 구조체에서 1바이트인 1개 멤버

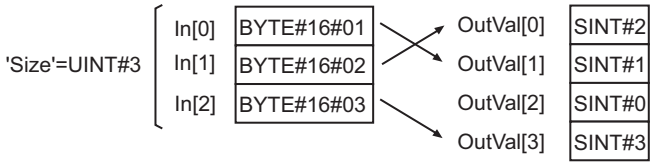
저장 방법은 다음과 같습니다.

'Order'의 값	'OutVal'이 배열인지의 여부	'OutVal'에 저장하는 방법
_LOW_HIGH	배열이 아님	In[0]의 값을 'OutVal'에 저장합니다.
	배열임	In[i]의 값을 OutVal[i]에 저장합니다.
_HIGH_LOW	배열이 아님	In[1]의 값을 'OutVal'에 저장합니다.
	배열임	In[i](i는 짝수)를 OutVal[i+1]에 저장합니다. In[i](i는 홀수)를 OutVal[i-1]에 저장합니다. 'Size'의 값이 홀수이면 OutVal['Size']까지 저장되며, OutVal['Size'-1]에는 16#00을 저장합니다.

'OutVal'이 SINT 형식이고 요소 수 3의 배열, 'Size'=UINT#3, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



'OutVal'과 'Size'가 상기와 동일하고 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



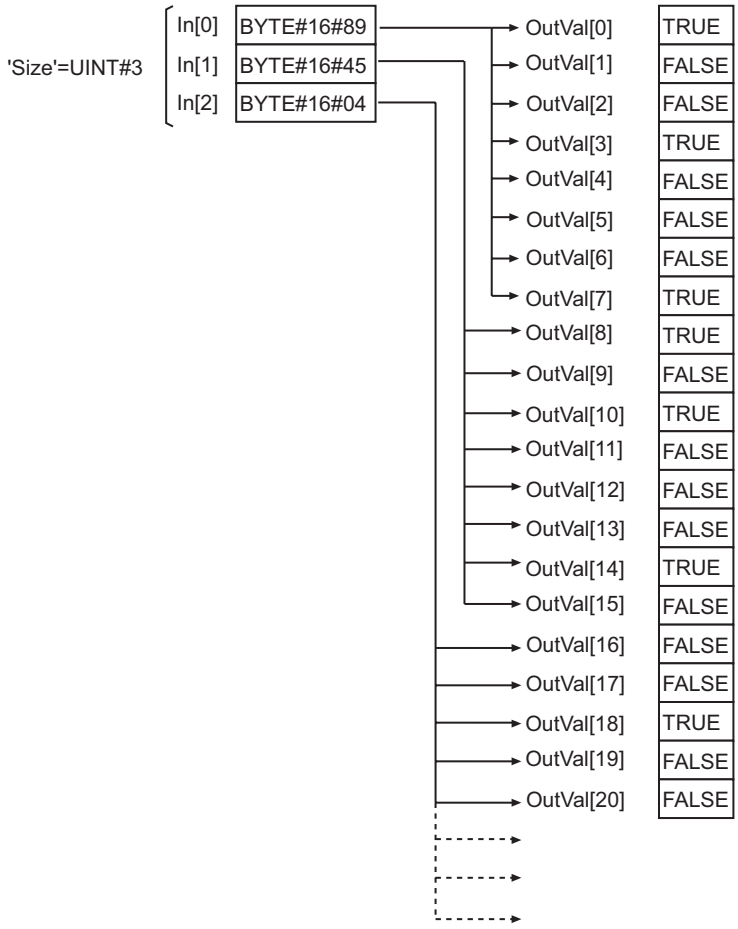
'OutVal'의 데이터 형식이 BOOL인 경우

'OutVal'의 데이터 형식이 BOOL(1비트)인 경우, 'OutVal'에 저장하는 방법은 다음과 같습니다.

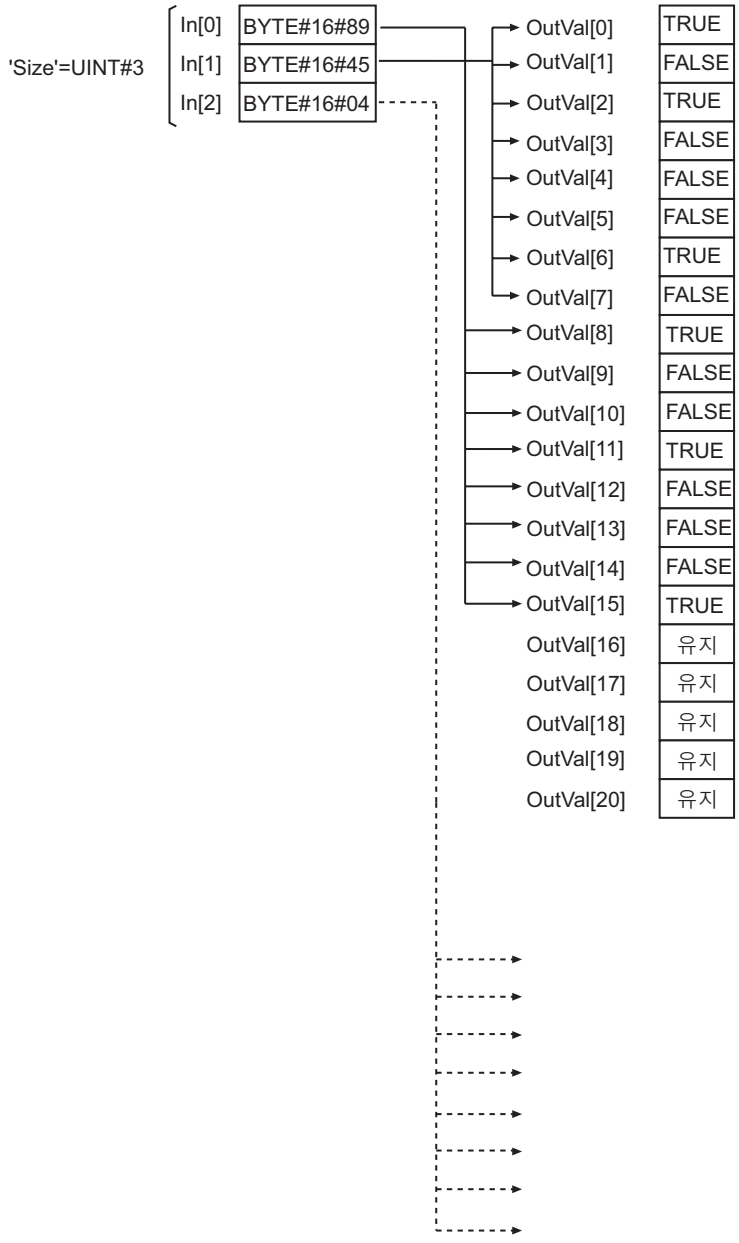
'Order'의 값	'OutVal'이 배열인지의 여부	'OutVal'에 저장하는 방법
_LOW_HIGH	배열이 아님	In[0]의 비트 0의 값을 'OutVal'에 저장합니다.
	배열임	In[0]의 값을 분해하여 OutVal[0]~OutVal[7]에 저장합니다. In[1]의 값을 분해하여 OutVal[8]~OutVal[15]에 저장합니다. 이후 마찬가지로 저장합니다. 남은 비트는 폐기합니다.

'Order'의 값	'OutVal'이 배열인지의 여부	'OutVal'에 저장하는 방법
_HIGH_LOW	배열이 아님	In[1]의 비트 0의 값을 'OutVal'에 저장합니다.
	배열임	In[0]의 값을 분해하여 OutVal[8]~OutVal[15]에 저장합니다. In[1]의 값을 분해하여 OutVal[0]~OutVal[7]에 저장합니다. 이후 마찬가지로 저장합니다. 남은 비트는 폐기합니다.

OutVal[]이 BOOL 형식이고 요소 수 21의 배열, 'Size'=UINT#3, 'Order'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.



OutVal[]과 'Size'가 상기와 동일하고 'Order'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 'OutVal'이 구조체 전체인 경우, 구성에 따라 In[]의 값의 일부가 각 멤버 사이에 있는 조정용 영역에 삽입되는 경우가 있습니다.
- 'Size'의 값이 0인 경우, 'Out'의 값은 TRUE가 되며 'OutVal'은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'OutVal'은 변화하지 않습니다.
  - 'Order'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 요소 수를 초과한 경우

SizeOfAry

배열 요소 수를 취득합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SizeOfAry	배열 요소 수 취득	FUN	(@)SizeOfAry EN ENO In Out	Out:=SizeOfAry(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	배열	입력	배열	데이터 형식에 따름	–	(*)
Out	요소 수	출력	요소 수	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

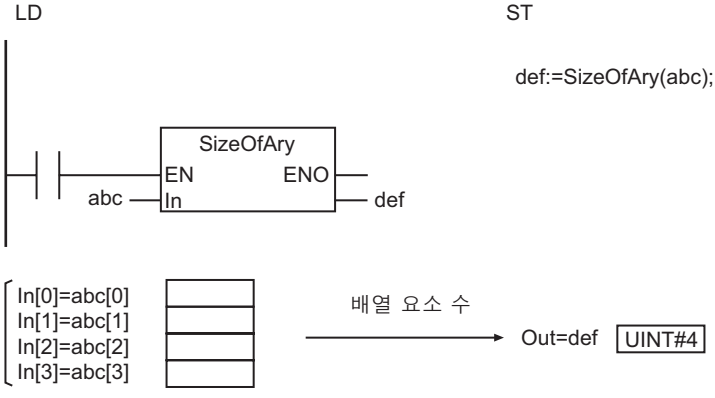
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Out							○													

기능

배열 In[]의 요소 수를 취득합니다.
입력 파라메터에는 array[0]과 같은 배열 요소명이 아닌, array와 같은 배열명을 지정하십시오.

동작 예가 아래에 나와 있습니다.

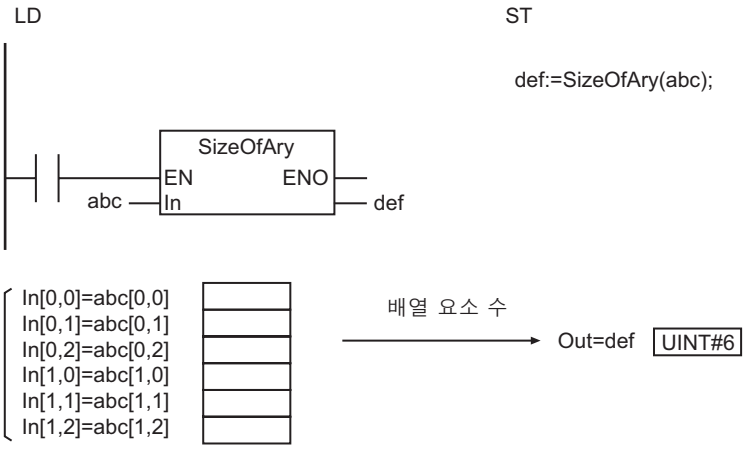
명칭	데이터 형식
abc	ARRAY[0..3] OF INT



참고

In[]은 2차원 이상의 배열이라도 상관없습니다. 그 경우, 'Out'에는 In[]의 모든 요소 수가 저장됩니다. 예를 들면, In[]에 전달하는 입력 파라메터의 배열이 ARRAY[0..1, 0..2]인 경우, 'Out'의 값은 UINT#6이 됩니다.

명칭	데이터 형식
abc	ARRAY[0..1,0..2] OF BOOL

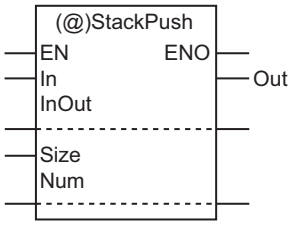


스택/테이블 명령

명령	명칭	페이지
StackPush	스택 데이터 저장	2-470
StackFIFO/StackLIFO	선입선출/후입선출	2-480
StackIns	스택 데이터 삽입	2-483
StackDel	스택 데이터 삭제	2-485
RecSearch	레코드 검색	2-487
RecRangeSearch	범위 지정 레코드 검색	2-492
RecSort	레코드 정렬	2-497
RecNum	레코드 수 취득	2-502
RecMax/RecMin	레코드 최대값 검색/ 레코드 최소값 검색	2-504

StackPush

스택에 값을 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StackPush	스택 데이터 저장	FUN		StackPush(In, InOut, Size, Num);

변수

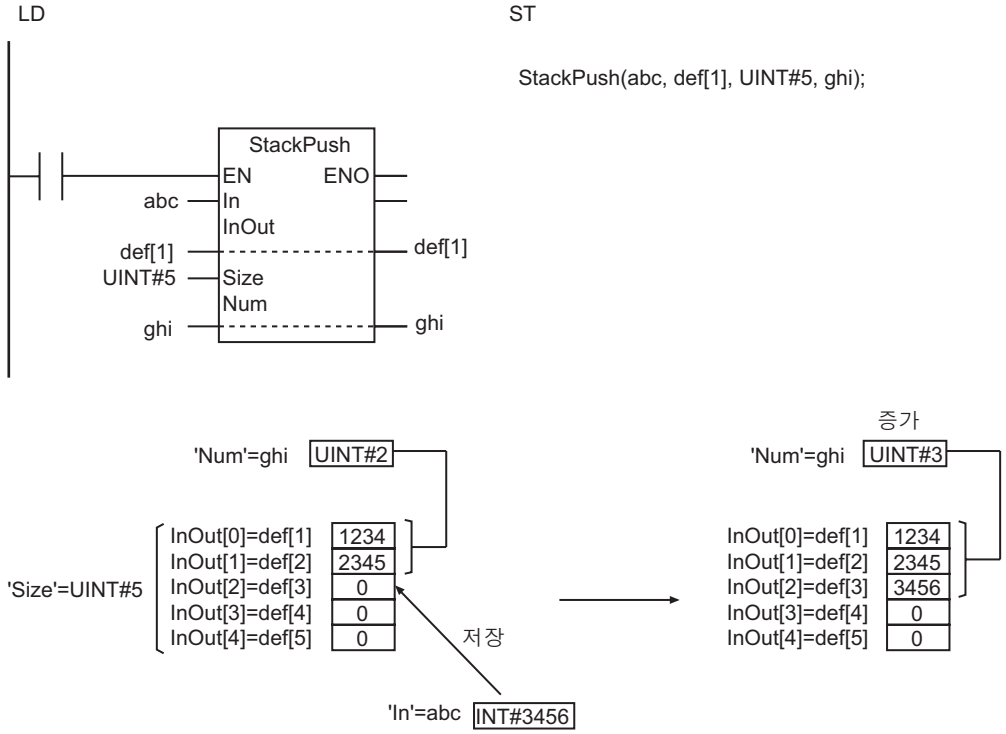
	명칭	입/출력	내용	유효 범위	단위	초기값
In	입력값	입력	스택에 입력할 값, 구조체, 구조체의 1개 멤버	데이터 형식에 따름	–	–
Size	스택의 요소 수		스택의 배열 요소 수			1
InOut[] 배열	스택 배열	입출력	스택을 구성하는 배열	데이터 형식에 따름	–	–
Num	스택의 저장 요소 수		스택에 저장되어 있는 요소 수			
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Size							○													
InOut[] 배열	'In'과 동일한 데이터 형식을 요소로 하는 배열																			
Num							○													
Out	○																			

기능

현재 스택 배열 InOut[]에 저장 요소 수 'Num'개의 요소가 저장되어 있는 것으로 간주합니다. 그리고 입력 값 'In'을 그 다음 요소 InOut['Num']에 덮어씁니다.
그 후에 'Num'을 증가시킵니다.
스택의 요소 수 'Size'에는 InOut[] 중에서 스택으로 사용할 요소 수를 지정합니다.

'Size'=UINT#5, 'Num'=UINT#2 인 경우의 예가 아래에 나와 있습니다 .



참고

스택의 최하위 또는 최상위 값을 가져오는 경우에는 'StackFIFO / StackLIFO 명령(2-480페이지)'을 사용하십시오.

사용상의 주의 사항

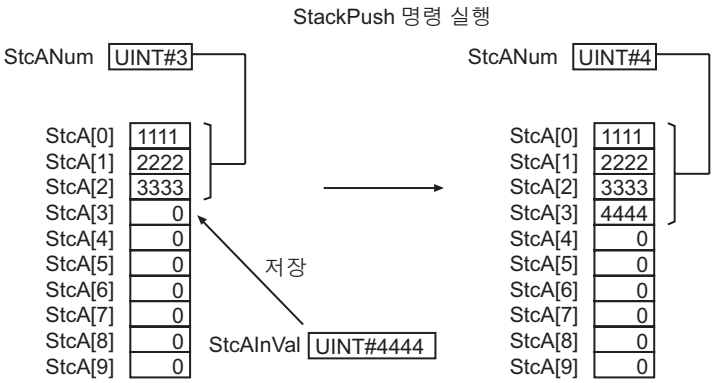
- 'In'과 InOut[]의 요소의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- InOut[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Size'의 값이 0인 경우, InOut[], 'Num'의 값은 변화하지 않습니다.
- 'In'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'In'이 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 0 이외이고 'Num' ≧ 'Size'인 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우
  - 'In', InOut[]이 STRING 형식이고 'In'의 바이트 수가 InOut[]의 크기를 초과한 경우

샘플 프로그램

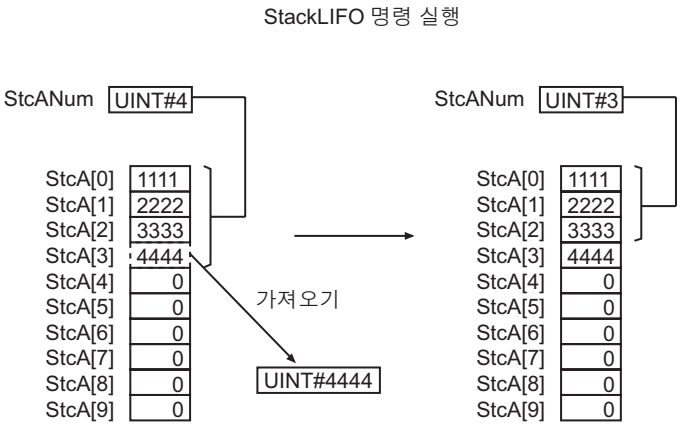
배열 변수 StcA[0..9]를 스택으로 합니다. 먼저, 스택에 3개의 데이터(UINT#1111, UINT#2222, UINT#3333)를 저장해둡니다.

StcA[0]	1111
StcA[1]	2222
StcA[2]	3333
StcA[3]	0
StcA[4]	0
StcA[5]	0
StcA[6]	0
StcA[7]	0
StcA[8]	0
StcA[9]	0

StackPush 명령을 사용하여 스택의 최상위 StcA[3]에 새로운 데이터(UINT#4444)를 1개 저장합니다. 스택 내의 데이터 수는 4개가 됩니다.

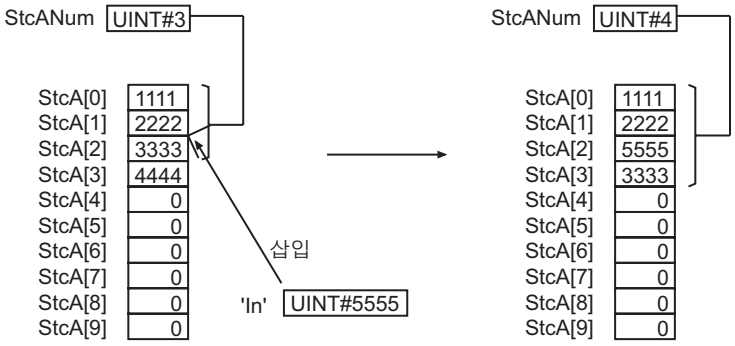


다음으로, StackLIFO 명령을 사용하여 스택의 최상위 StcA[3]의 데이터를 1개 가져옵니다. 스택의 데이터 수는 3개가 됩니다.



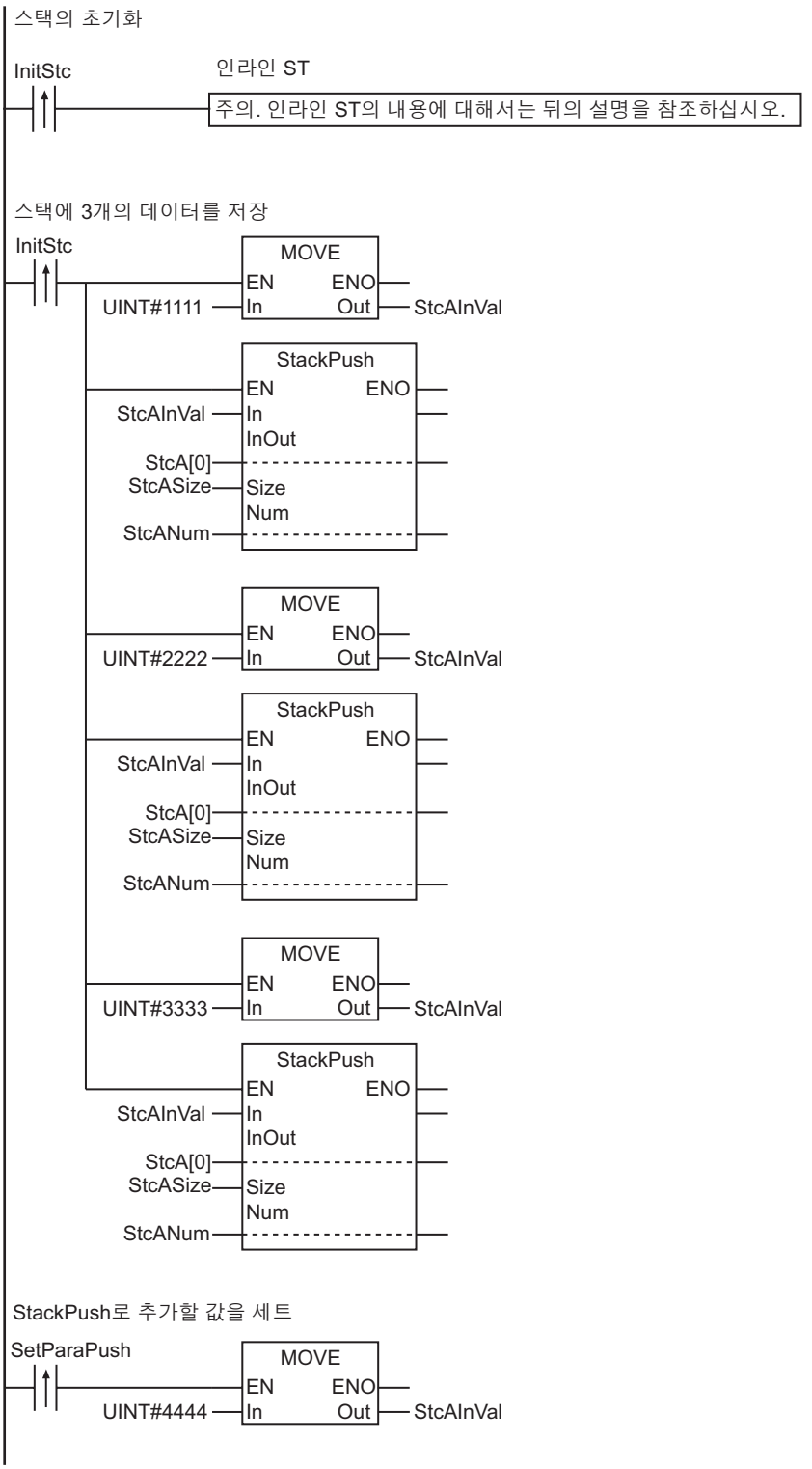
마지막으로, StackIns 명령을 사용하여 StcA[1]과 StcA[2] 사이에 데이터(UINT#5555)를 1개 삽입합니다. 스택의 데이터 수는 4개가 됩니다.

StackIns 명령 실행

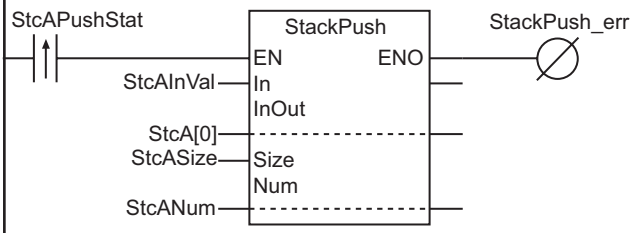


LD

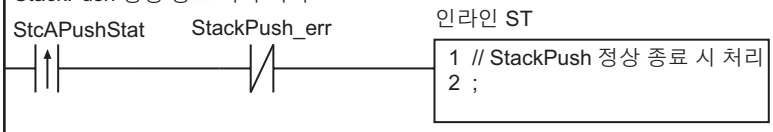
명칭	데이터 형식	초기값	코멘트
InitStc	BOOL	FALSE	스택의 초기화 조건
StcANum	UINT	0	스택의 저장 요소 수
StcA	ARRAY[0..9] OF UINT	[10(0)]	스택 배열
StcASize	UINT	0	스택의 요소 수
SetParaPush	BOOL	FALSE	StcAInVal을 세트하는 실행 조건
StcAInVal	UINT	0	StackPush로 추가할 값
StcAPushStat	BOOL	FALSE	StackPush의 실행 조건
StackPush_err	BOOL	FALSE	StackPush의 이상 플래그
StcALIFOStat	BOOL	FALSE	StackLIFO의 실행 조건
StcAOutVal	UINT	0	StackLIFO로 가져오는 값
StackLIFO_err	BOOL	FALSE	StackLIFO의 이상 플래그
SetParaIns	BOOL	FALSE	StcAInsVal과 StcAOffset을 세트하는 실행 조건
StcAInsVal	UINT	0	StackIns로 삽입할 값
StcAOffset	UINT	0	StackIns의 오프셋값
StcAInsStat	BOOL	FALSE	StackIns의 실행 조건
StackIns_err	BOOL	FALSE	StackIns의 이상 플래그


스택의 초기화
InitStc
인라인 ST
주의. 인라인 ST의 내용에 대해서는 뒤의 설명을 참조하십시오.
스택에 3개의 데이터를 저장
InitStc
MOVE
EN
ENO
UINT#1111
In
Out
StcAInVal
StackPush
EN
ENO
StcAInVal
In
InOut
StcA[0]
StcASize
Size
Num
StcANum
MOVE
EN
ENO
UINT#2222
In
Out
StcAInVal
StackPush
EN
ENO
StcAInVal
In
InOut
StcA[0]
StcASize
Size
Num
StcANum
MOVE
EN
ENO
UINT#3333
In
Out
StcAInVal
StackPush
EN
ENO
StcAInVal
In
InOut
StcA[0]
StcASize
Size
Num
StcANum
StackPush로 추가할 값을 세트
SetParaPush
MOVE
EN
ENO
UINT#4444
In
Out
StcAInVal

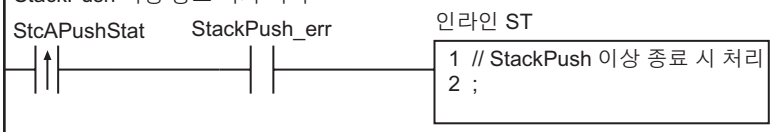
StackPush 명령으로 데이터 추가



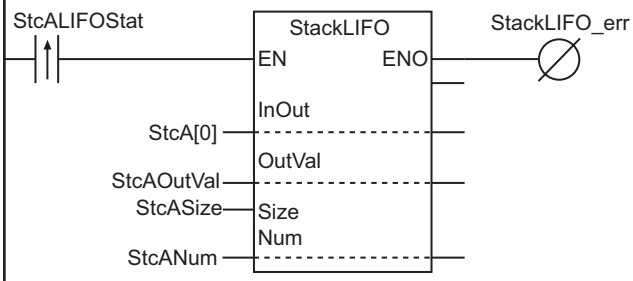
StackPush 정상 종료 시의 처리



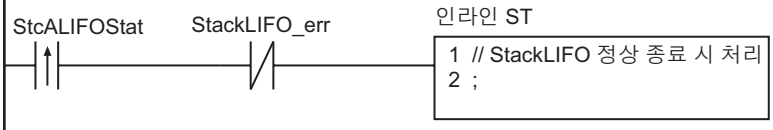
StackPush 이상 종료 시의 처리



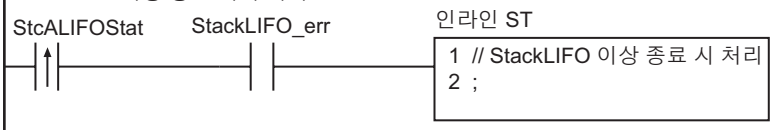
StackLIFO 명령으로 데이터 가져오기



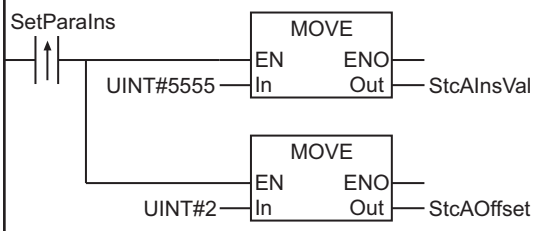
StackLIFO 정상 종료 시의 처리

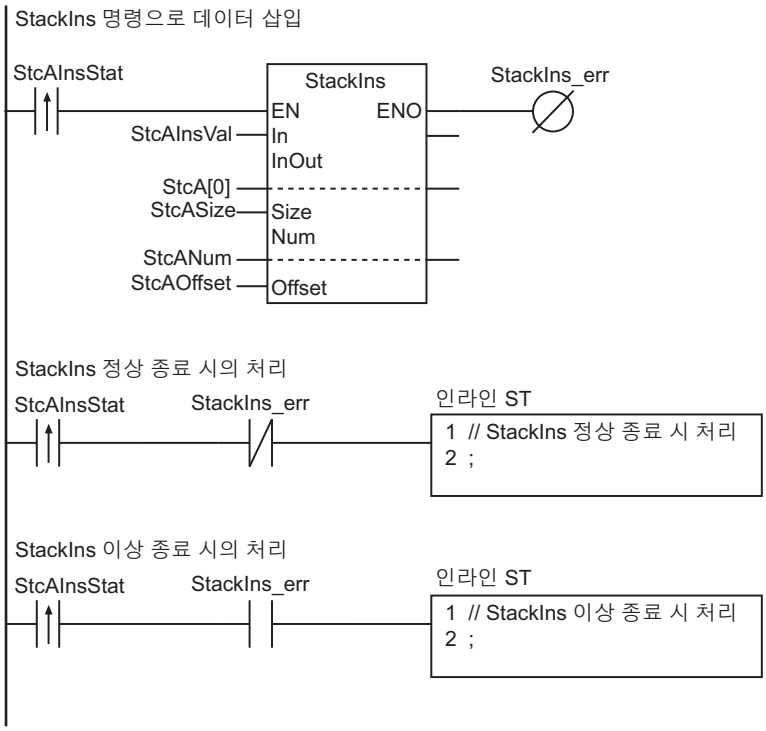


StackLIFO 이상 종료 시의 처리



StackIns로 삽입할 값과 오프셋을 세트





● 인라인 ST의 내용

```
StcANum:=0;
Clear(StcA);
StcASize:=SizeOfAry(StcA);
```

ST

명칭	데이터 형식	초기값	코멘트
InitStc	BOOL	FALSE	스택의 초기화 조건
preInitStc	BOOL	FALSE	이전 태스크 주기의 InitStc의 값
StcANum	UINT	0	스택의 저장 요소 수
StcA	ARRAY[0..9] OF UINT	[10(0)]	스택 배열
StcASize	UINT	0	스택의 요소 수
StcAPushStat	BOOL	FALSE	StackPush의 실행 조건
preStcAPushStat	BOOL	FALSE	이전 태스크 주기의 StcAPushStat의 값
StcAInVal	UINT	0	StackPush로 추가할 값
StcAPush_OK	BOOL	FALSE	StackPush의 정상 종료 플래그
StcAPushNormalEnd	BOOL	FALSE	StackPush의 정상 종료 시 처리
StcAPushErrorEnd	BOOL	FALSE	StackPush의 이상 종료 시 처리
StcALIFOStat	BOOL	FALSE	StackLIFO의 실행 조건
preStcALIFOStat	BOOL	FALSE	이전 태스크 주기의 StcALIFOStat의 값
StcAOutVal	UINT	0	StackLIFO로 가져오는 값
StcALIFO_OK	BOOL	FALSE	StackLIFO의 정상 종료 플래그
StcALIFONormalEnd	BOOL	FALSE	StackLIFO의 정상 종료 시 처리
StcALIFOErrorEnd	BOOL	FALSE	StackLIFO의 이상 종료 시 처리
StcAInsStat	BOOL	FALSE	StackIns의 실행 조건
preStcAInsStat	BOOL	FALSE	이전 태스크 주기의 StcAInsStat의 값
StcAInsVal	UINT	0	StackIns로 삽입할 값
StcAOffset	UINT	0	StackIns의 오프셋값
StcAIns_OK	BOOL	FALSE	StackIns의 정상 종료 플래그
StcAInsNormalEnd	BOOL	FALSE	StackIns의 정상 종료 시 처리
StcAInsErrorEnd	BOOL	FALSE	StackIns의 이상 종료 시 처리

```
// 스택의 초기화
IF ( (InitStc=TRUE) AND (preInitStc=FALSE) ) THEN
   StcANum:=0;
   Clear(StcA);
   StcASize:=SizeOfAry(StcA);
END_IF;

// 스택에 3개의 데이터를 저장
IF ( (InitStc=TRUE) AND (preInitStc=FALSE) ) THEN
   StackPush(In:=UINT#1111, InOut:=StcA[0], Size:=StcASize, Num:=StcANum);
   StackPush(In:=UINT#2222, InOut:=StcA[0], Size:=StcASize, Num:=StcANum);
   StackPush(In:=UINT#3333, InOut:=StcA[0], Size:=StcASize, Num:=StcANum);
END_IF;

preInitStc:=InitStc;

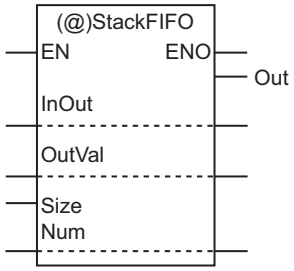
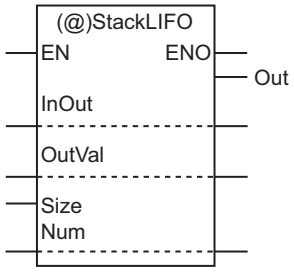
// StackPush 명령으로 데이터 추가
IF ( (StcAPushStat=TRUE) AND (preStcAPushStat=FALSE) ) THEN
   StcAInVal:=UINT#4444;
   StackPush(
      In      :=StcAInVal,          // 추가할 값
      InOut  :=StcA[0],             // 스택 배열의 시작 요소
      Size   :=StcASize,            // 스택의 요소 수
      Num   :=StcANum,              // 스택의 저장 요소 수
      ENO   =>StcAPush_OK);   // 정상 종료 플래그
   IF (StcAPush_OK=TRUE) THEN
      StcAPushNormalEnd:=TRUE; // 정상 종료 시 처리
   ELSE
      StcAPushErrorEnd:=TRUE;    // 이상 종료 시 처리
   END_IF;
END_IF;
```

```
preStcAPushStat:=StcAPushStat;
// StackLIFO 명령으로 데이터 가져오기
IF ( (StcALIFOStat=TRUE) AND (preStcALIFOStat=FALSE) ) THEN
   StackLIFO(
      InOut  :=StcA[0],                  // 스택 배열의 시작 요소
      OutVal :=StcAOutVal,               // 스택에서 가져오는 값
      Size   :=StcASize,                 // 스택의 요소 수
      Num    :=StcANum,                  // 스택의 저장 요소 수
      ENO    =>StcALIFO_OK);             // 정상 종료 플래그
   IF (StcALIFO_OK=TRUE) THEN
      StcALIFONormalEnd:=TRUE;    // 정상 종료 시 처리
   ELSE
      StcALIFOErrorEnd:=TRUE;     // 이상 종료 시 처리
   END_IF;
END_IF;
preStcALIFOStat:=StcALIFOStat;

// StackIns 명령으로 데이터 삽입
IF ( (StcAInsStat=TRUE) AND (preStcAInsStat=FALSE) ) THEN
   StcAInsVal:=UINT#5555;
   StcAOffset:=UINT#2;
   StackIns(
      In     :=StcAInsVal,               // 스택에 삽입할 값
      InOut  :=StcA[0],                  // 스택 배열의 시작 요소
      Size   :=StcASize,                 // 스택의 요소 수
      Num    :=StcANum,                  // 스택의 저장 요소 수
      Offset :=StcAOffset,               // 삽입할 오프셋 위치
      ENO    =>StcAIns_OK);              // 정상 종료 플래그
   IF (StcAIns_OK=TRUE) THEN
      StcAInsNormalEnd:=TRUE;     // 정상 종료 시 처리
   ELSE
      StcAInsErrorEnd:=TRUE;      // 이상 종료 시 처리
   END_IF;
END_IF;
preStcAInsStat:=StcAInsStat;
```

StackFIFO / StackLIFO

StackFIFO : 스택의 최하위 값을 가져옵니다.
StackLIFO : 스택의 최상위 값을 가져옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StackFIFO	선입선출	FUN		StackFIFO(InOut, OutVal, Size, Num);
StackLIFO	후입선출	FUN		StackLIFO(InOut, OutVal, Size, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Size	스택의 요소 수	입력	스택의 배열 요소 수	데이터 형식에 따름	–	1
InOut[] 배열	스택 배열	입출력	스택을 구성하는 배열	데이터 형식에 따름	–	–
OutVal	출력값		스택에서 출력하는 값, 구조체 전체			
Num	스택의 저장 요소 수		스택에 저장되어 있는 요소 수			
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size							○													
InOut[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
OutVal	InOut[]의 요소와 동일한 데이터 형식																			
Num							○													
Out	○																			

기능

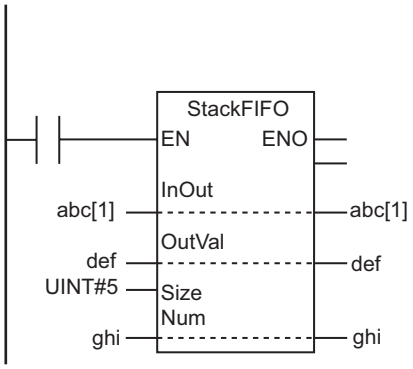
현재 스택 배열 InOut[]에 저장 요소 수 'Num'개의 요소가 저장되어 있는 것으로 간주합니다. 거기서 값을 가져와서 출력값 'OutVal'에 대입합니다.
스택의 요소 수 'Size'에는 InOut[] 중에서 스택으로 사용할 요소 수를 지정합니다.

● StackFIFO

스택의 최하위 값을 가져옵니다. InOut[0]의 값을 'OutVal'에 대입합니다.
그 다음, InOut[1] 이후의 'Num'-1개의 요소를 각각 스택 배열의 하위 방향으로 하나씩 이동시킵니다.
마지막으로, 'Num'을 감소시킵니다.

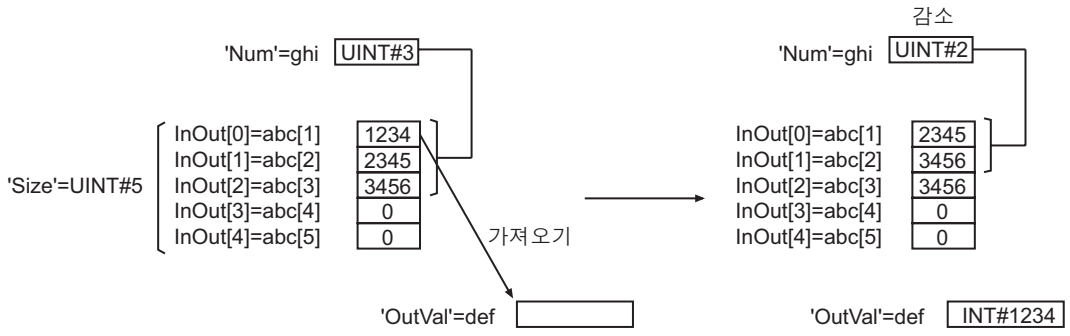
'Size'=UINT#5, 'Num'=UINT#3인 경우의 예가 아래에 나와 있습니다.

LD



ST

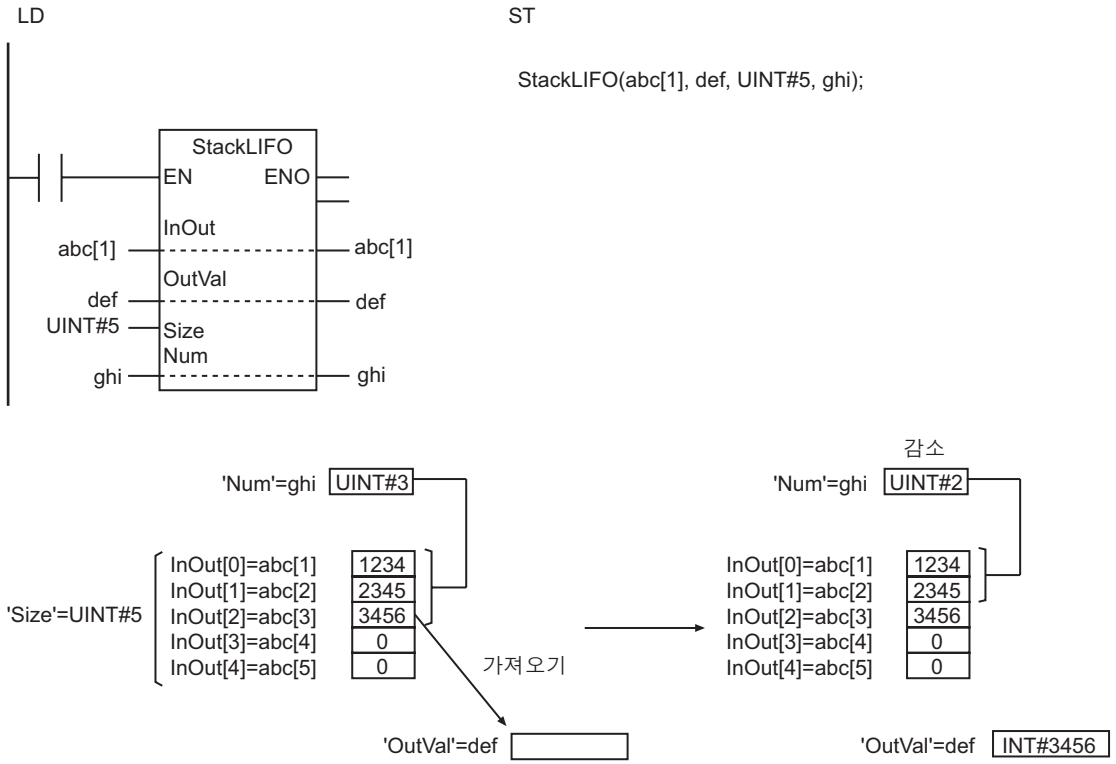
```
StackFIFO(abc[1], def, UINT#5, ghi);
```



● StackLIFO

스택의 최상위 값을 가져옵니다. InOut['Num'-1]의 값을 'OutVal'에 대입합니다.
그리고 'Num'을 감소시킵니다.

'Size'=UINT#5, 'Num'=UINT#2인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

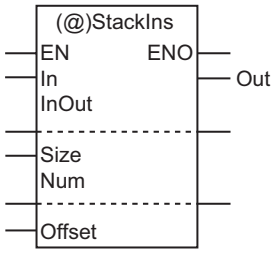
- InOut[]과 'OutVal'의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- InOut[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Size' 또는 'Num'이 0인 경우, InOut[], 'Num', 'OutVal'의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'OutVal'은 변화하지 않습니다.
  - 'Num' 및 'Size'의 값이 0 이외이고 'Num' > 'Size'인 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우
  - InOut[]이 STRING 형식 배열이고 요소 중 하나가 NULL 문자로 끝나지 않은 경우
  - InOut[]이 STRING 형식 배열이고 요소의 바이트 수가 'OutVal'의 크기를 초과한 경우

샘플 프로그램

🕮 'StackPush 명령(2-470페이지)'의 샘플 프로그램을 참조하십시오.

StackIns

스택의 임의의 위치에 값을 삽입합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
StackIns	스택 데이터 삽입	FUN		StackIns(In, InOut, Size, Num, Offset);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	삽입값	입력	스택에 삽입할 값, 구조체 전체, 구조체의 1개 멤버	데이터 형식에 따름	–	(*)
Size	스택의 요소 수		스택의 배열 요소 수			1
Offset	오프셋 위치		'In'을 삽입할 스택의 오프셋 위치			0
InOut[] 배열	스택 배열	입출력	스택을 구성하는 배열	데이터 형식에 따름	–	–
Num	스택의 저장 요소 수		스택에 저장되어 있는 요소 수			
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

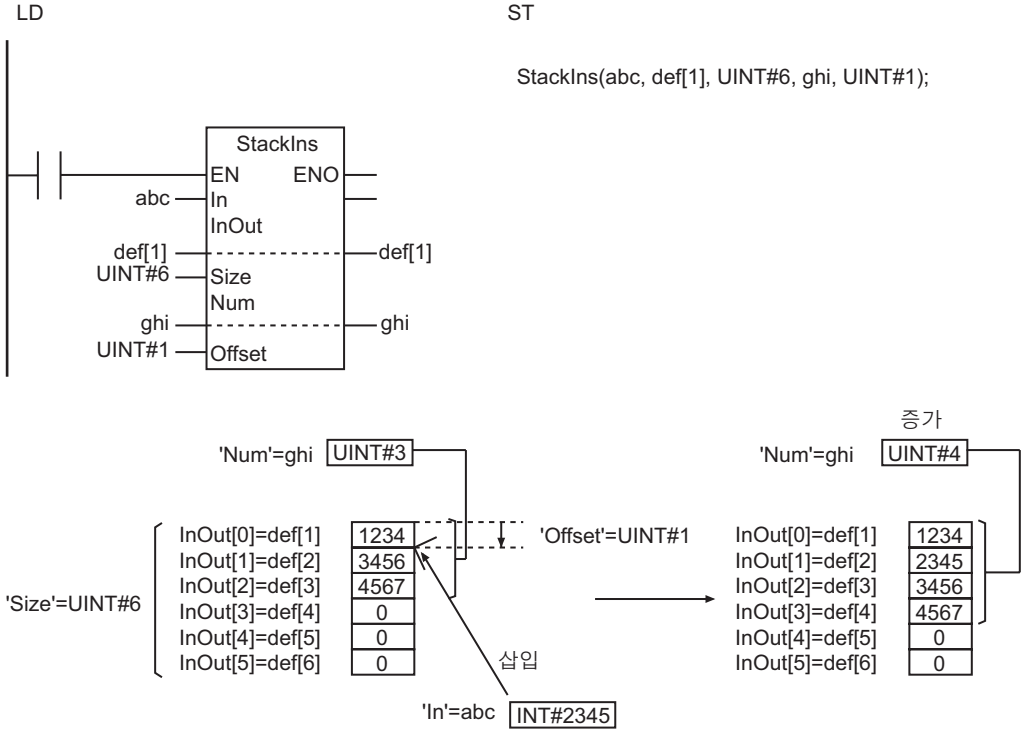
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
Size							○													
Offset							○													
InOut[] 배열	'In'과 동일한 데이터 형식을 요소로 하는 배열																			
Num							○													
Out	○																			

기능

현재 스택 배열 InOut[]에 저장 요소 수 'Num'개의 요소가 저장되어 있는 것으로 간주합니다. 그리고 오프셋 위치 'Offset'으로 정해지는 위치 InOut['Offset']에 삽입값 'In'을 삽입합니다.
또한 그보다 상위의 요소, 즉 InOut['Offset']부터 InOut['Num'-1]은 각각 스택 배열의 상위 방향으로 하나씩 이동시킵니다.

그 후에 'Num'을 증가시킵니다.
스택의 요소 수 'Size'에는 InOut[] 중에서 스택으로 사용할 요소 수를 지정합니다.

'Size'=UINT#6, 'Num'=UINT#3, 'Offset'=UINT#1인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

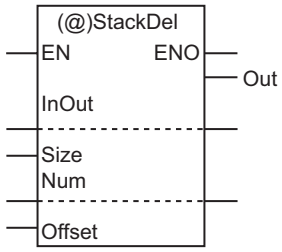
- 'In'과 InOut[]의 데이터 형식을 동일하게 하십시오. 데이터 형식이 다르면 빌드 시에 이상 상태가 됩니다.
- InOut[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Size'의 값이 0인 경우, InOut[], 'Num'의 값은 변화하지 않습니다.
- 'In'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'In'이 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Size'의 값이 0 이외이고 'Size' > 'Num' ≧ 'Offset'이 아닌 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우
  - 'In', InOut[]이 STRING 형식이고 'In'의 바이트 수가 InOut[]의 크기를 초과한 경우

샘플 프로그램

🕮 'StackPush 명령(2-470페이지)'의 샘플 프로그램을 참조하십시오.

StackDel

스택의 임의의 위치의 값을 삭제합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
StackDel	스택 데이터 삭제	FUN		StackDel(InOut, Size, Num, Offset);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Size	스택의 요소 수	입력	스택의 배열 요소 수	데이터 형식에 따름	–	1
Offset	오프셋 위치		삭제할 스택의 오프셋 위치			0
InOut[] 배열	스택 배열	입출력	스택을 구성하는 배열	데이터 형식에 따름	–	–
Num	스택의 저장 요소 수		스택에 저장되어 있는 요소 수			
Out	반환값	출력	항상 TRUE	TRUE만	–	–

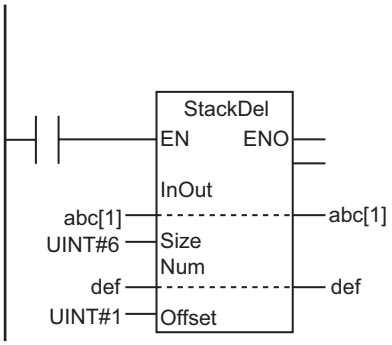
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size							○													
Offset							○													
InOut[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Num							○													
Out	○																			

기능

현재 스택 배열 InOut[]에 저장 요소 수 'Num'개의 요소가 저장되어 있는 것으로 간주합니다. 그리고 오프셋 위치 'Offset'으로 정해지는 위치 InOut['Offset']의 값을 삭제합니다.
또한 그보다 상위의 요소, 즉 InOut['Offset'+1]부터 InOut['Num'-1]은 각각 스택 배열의 하위 방향으로 하나씩 이동시킵니다.
그 후에 'Num'을 감소시킵니다.
스택의 요소 수 'Size'에는 InOut[] 중에서 스택으로 사용할 요소 수를 지정합니다.

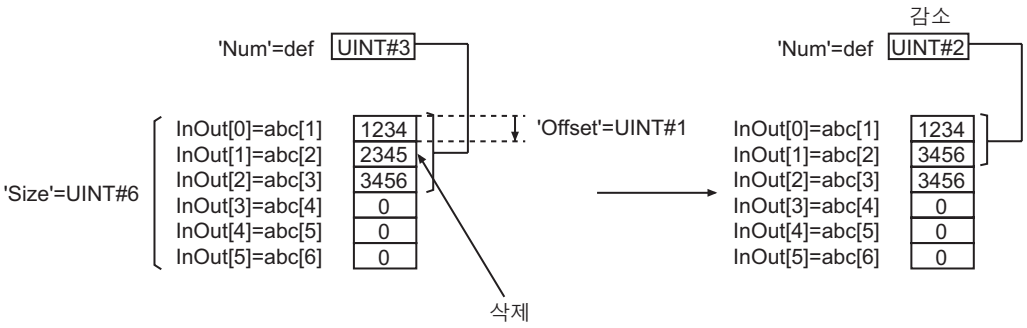
'Size'=UINT#6, 'Num'=UINT#3, 'Offset'=UINT#1인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
StackDel(abc[1], UINT#6, def, UINT#1);
```

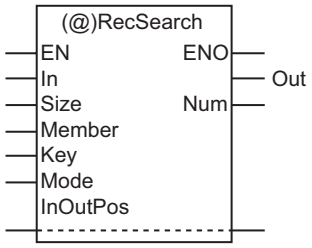


사용상의 주의 사항

- InOut[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Size' 또는 'Num'의 값이 0인 경우, InOut[], 'Num'의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 InOut[]은 변화하지 않습니다.
  - 'Num' 및 'Size'의 값이 0 이외이고 'Size' ≧ 'Num' > 'Offset'이 아닌 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우

RecSearch

구조체를 요소로 하는 배열에서 검색 키와 일치하는 요소를 지정된 방법으로 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RecSearch	레코드 검색	FUN		Out: =RecSearch(In, Size, Member, Key, Mode, InOutPos, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	검색 대상 배열	입력	검색 대상이 되는 구조체를 요소로 하는 배열	–	–	(*)
Size	검색 대상의 요소 수		검색 대상의 배열 요소 수	데이터 형식에 따름		1
Member	검색 대상 멤버		In[]의 구조체의 검색 대상 멤버			(*)
Key	검색 키		검색값			
Mode	검색 방법		검색 방법	_LINEAR, _BIN_ASC, _BIN_DESC		_LINEAR
InOutPos[] 배열	일치 요소의 요소 번호	입출력	일치 요소의 요소 번호	데이터 형식에 따름	–	–
Out	검색 결과	출력	TRUE : 일치한 요소 있음 FALSE : 일치한 요소 없음	데이터 형식에 따름	–	–
Num	일치 요소의 개수		일치 요소의 개수			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	구조체를 요소로 하는 배열을 지정																			
Size							○													
Member						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	In[]의 검색 대상 멤버와 동일한 데이터 형식															(*)				
Key	'Member'와 동일한 데이터 형식																			
Mode	열거 형식 _eSEARCH_MODE 열거자는 기능 설명 참조																			
InOutPos[] 배열							○													
Out	○																			
Num							○													

* TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

구조체를 요소로 하는 배열 In[]의 'Size'개의 요소, 즉 In[0]부터 In['Size'-1] 중에서 구조체의 검색 대상 멤버 'Member'의 값이 검색 키 'Key'와 일치하는 것을 검색합니다.
'Member'에는 In[] 가운데 한 요소의 검색 대상 멤버를 인수로 전달합니다.
일치한 요소가 있으면 검색 결과 'Out'의 값이 TRUE가 됩니다. 그리고, 일치 요소의 요소 번호가 InOutPos[0]에, 일치 요소의 수가 'Num'에 각각 대입됩니다. 일치 요소가 2개 이상 있는 경우에는 In[] 중에서 가장 하위에 있는 일치 요소의 요소 번호가 InOutPos[0]에 대입됩니다.
일치한 요소가 없으면 'Out'의 값은 FALSE가 되고 InOutPos[0]과 'Num'은 0이 됩니다.

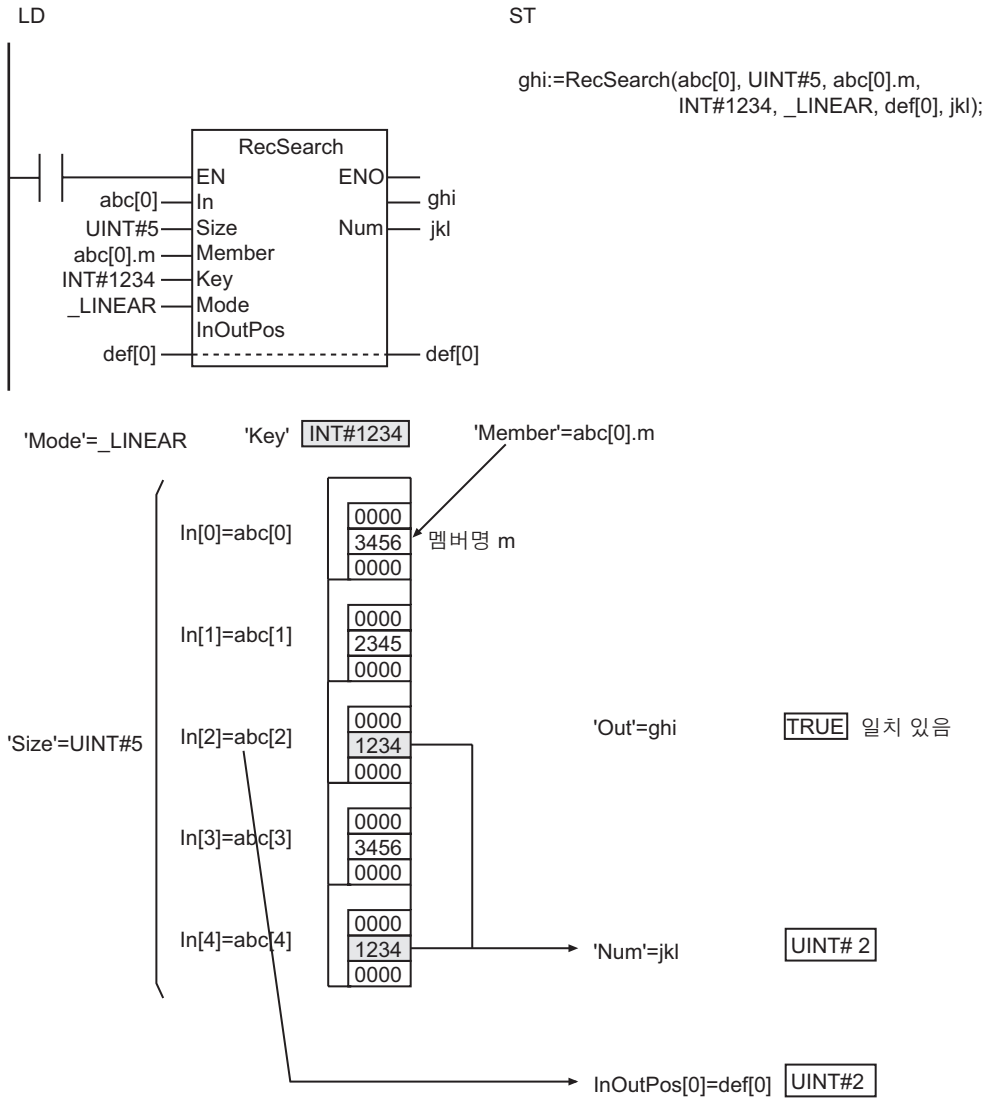
In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

검색 방법 'Mode'의 데이터 형식은 열거 형식 _eSEARCH_MODE입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LINEAR	선형 탐색
_BIN_ASC	오름차순 이분 탐색
_BIN_DESC	내림차순 이분 탐색

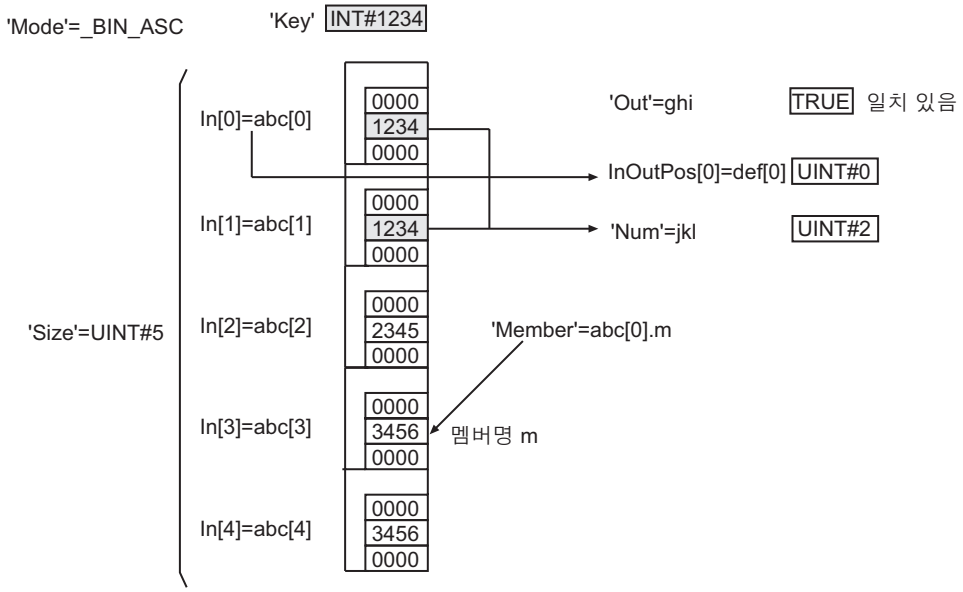
선형 탐색의 경우, In[]의 시작 요소부터 순서대로 검색을 수행합니다.

'Size'=UINT#5, 'Key'=INT#1234, 'Mode'=_LINEAR의 예가 아래에 나와 있습니다.

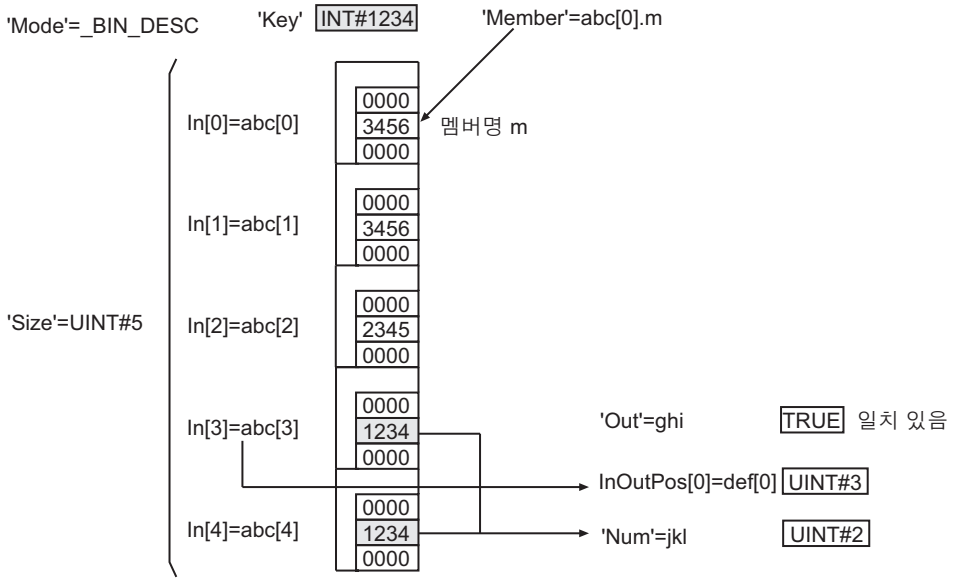


오름차순 이분 탐색의 경우, 이 명령을 실행하기 전에 In[]에 전달하는 입력 파라메터의 배열 요소를 오름차순으로 정렬해두어야 합니다. 그리고 이 명령을 실행하면 이분 탐색이 수행됩니다.

위와 동일한 예를 오름차순 이분 탐색한 경우, 배열 요소의 순서와 처리 결과는 다음과 같습니다.



내림차순 이분 탐색의 경우, 이 명령을 실행하기 전에 In[]에 전달하는 입력 파라메터의 배열 요소를 내림차순으로 정렬해두어야 합니다. 그리고 이 명령을 실행하면 이분 탐색이 수행됩니다.
위와 동일한 예를 내림차순 이분 탐색한 경우, 배열 요소의 순서와 처리 결과는 다음과 같습니다.



참고

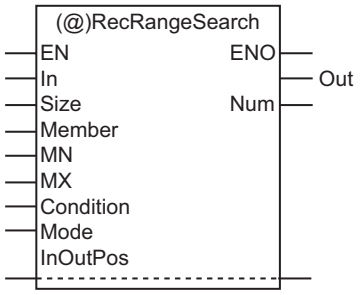
- In[] 전체가 보다 상위의 구조체 멤버라도 상관없습니다.
  (예) In[0]=str0.str1[0]
- In[]은 2차원 이상의 배열이라도 상관없습니다. 그 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에 각각 대입됩니다.
- In[]이 3차원 배열인 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에, 제3차원 요소 번호가 InOutPos[2]에 각각 대입됩니다.
- TIME 형식, DT 형식, TOD 형식의 검색을 수행하는 경우, 'Member'와 'Key'의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- In[]은 구조체를 요소로 하는 배열로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 'key'와 'Member'의 데이터 형식은 동일하게 설정하십시오. 서로 다르면 빌드 시에 이상 상태가 됩니다.
- In[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Member'가 실수인 경우, 값에 따라 오차가 발생하므로 기대한 결과를 얻지 못하는 경우가 있습니다.
- 'Key'가 실수인 경우, 'Key'에 비수를 지정하지 마십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 FALSE, 'Num'의 값은 0이 됩니다. InOutPos[]는 변화하지 않습니다.
- 'Mode'의 값이 BIN_ASC(또는 _BIN_DESC)인 경우, In[]의 요소가 오름차순(또는 내림차순)으로 정렬되어 있지 않으면 올바른 결과를 얻을 수 없습니다. 이 명령 실행 전에 요소를 오름차순(또는 내림차순)으로 정렬하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', InOutPos[], 'Num'은 변화하지 않습니다.
  - 'Mode'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우
  - 'Member'가 In[]의 멤버가 아닌 경우
  - InOutPos[]의 배열 크기가 In[]의 차원 수보다 작은 경우
  - 'Member'가 STRING 형식이고 NULL 문자로 끝나지 않은 경우

RecRangeSearch

구조체를 요소로 하는 배열에서 검색 조건의 범위와 일치하는 요소를 지정된 방법으로 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RecRange Search	범위 지정 레코드 검색	FUN		Out: =RecRangeSearch(In, Size, Member, MN, MX, Condition, Mode, InOutPos, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	검색 대상 배열	입력	검색 대상이 되는 구조체를 요소로 하는 배열	–		(*)
Size	검색 대상의 요소 수		검색 대상의 배열 요소 수	데이터 형식에 따름		1
Member	검색 대상 멤버		In[]의 구조체의 검색 대상 멤버			(*)
MN	검색 조건 하한값		검색 조건 하한값			
MX	검색 조건 상한값		검색 조건 상한값		–	
Condition	검색 조건		검색 조건	_EQ_BOTH, _EQ_MIN, _EQ_MAX, _NE_BOTH		_EQ_ BOTH
Mode	검색 방법		검색 방법	_LINEAR, _BIN_ASC, _BIN_DESC		_LINE AR
InOutPos[] 배열	일치 요소의 요소 번호	입출력	일치 요소의 요소 번호	데이터 형식에 따름	–	–
Out	검색 결과	출력	TRUE : 일치한 요소 있음 FALSE : 일치한 요소 없음	데이터 형식에 따름	–	–
Num	일치 요소의 개수		일치 요소의 개수			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	구조체를 요소로 하는 배열을 지정																			
Size							○													

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Member						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
																(*)				
	In[]의 검색 대상 멤버와 동일한 데이터 형식																			
MN	'Member'와 동일한 데이터 형식																			
MX	'Member'와 동일한 데이터 형식																			
Condition	열거 형식 _eSEARCH_CONDITION 열거자는 기능 설명 참조																			
Mode	열거 형식 _eSEARCH_MODE 열거자는 기능 설명 참조																			
InOutPos[] 배열							○													
Out	○																			
Num							○													

* TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

구조체를 요소로 하는 배열 In[]의 'Size'개의 요소, 즉 In[0]부터 In['Size'-1] 중에서 구조체의 검색 대상 멤버 'Member'의 값이 검색 조건과 일치하는 것을 검색합니다.
검색 조건과 검색 방법은 'Condition'과 'Mode'로 지정합니다. 자세한 내용은 뒤에서 다룹니다.
'Member'에는 In[] 가운데 한 요소의 검색 대상 멤버를 인수로 전달합니다.
조건과 일치한 요소가 있으면 검색 결과 'Out'의 값이 TRUE가 됩니다. 그리고 일치 요소의 요소 번호가 InOutPos[0]에, 일치 요소의 수가 'Num'에 각각 대입됩니다. 일치 요소가 2개 이상 있는 경우에는 In[] 중에서 가장 하위에 있는 일치 요소의 요소 번호가 InOutPos[0]에 대입됩니다.
일치한 요소가 없으면 'Out'의 값은 FALSE가 되고 InOutPos[0]과 'Num'은 0이 됩니다.

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

검색 조건 'Condition'의 데이터 형식은 열거 형식 _eSEARCH_CONDITION입니다. 열거자의 의미는 다음과 같습니다.

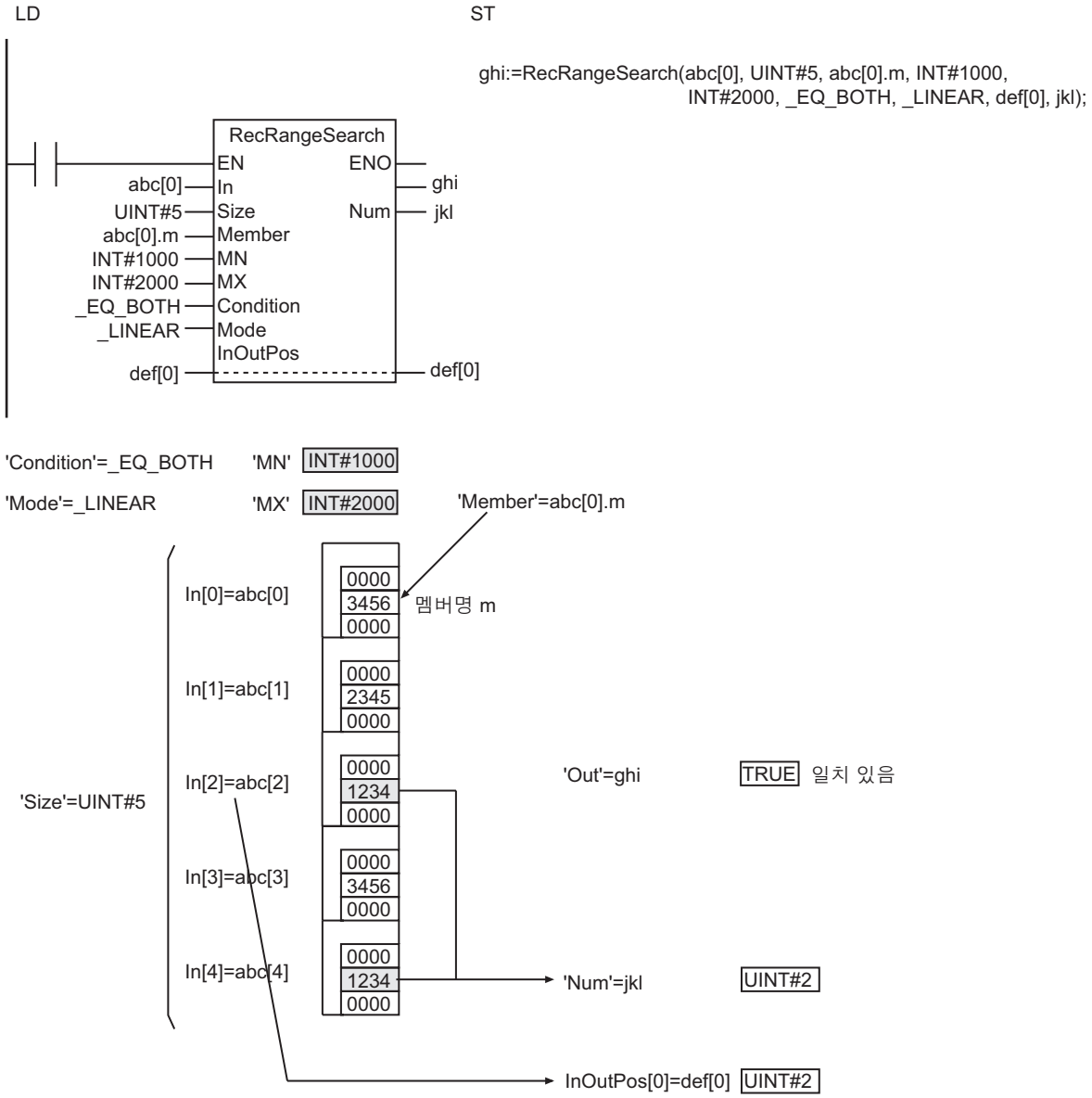
열거자	의미
_EQ_BOTH	'MN' ≦ 'Member' ≦ 'MX'
_EQ_MIN	'MN' ≦ 'Member' < 'MX'
_EQ_MAX	'MN' < 'Member' ≦ 'MX'
_NE_BOTH	'MN' < 'Member' < 'MX'

검색 방법 'Mode'의 데이터 형식은 열거 형식 _eSEARCH_MODE입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LINEAR	선형 탐색
_BIN_ASC	오름차순 이분 탐색
_BIN_DESC	내림차순 이분 탐색

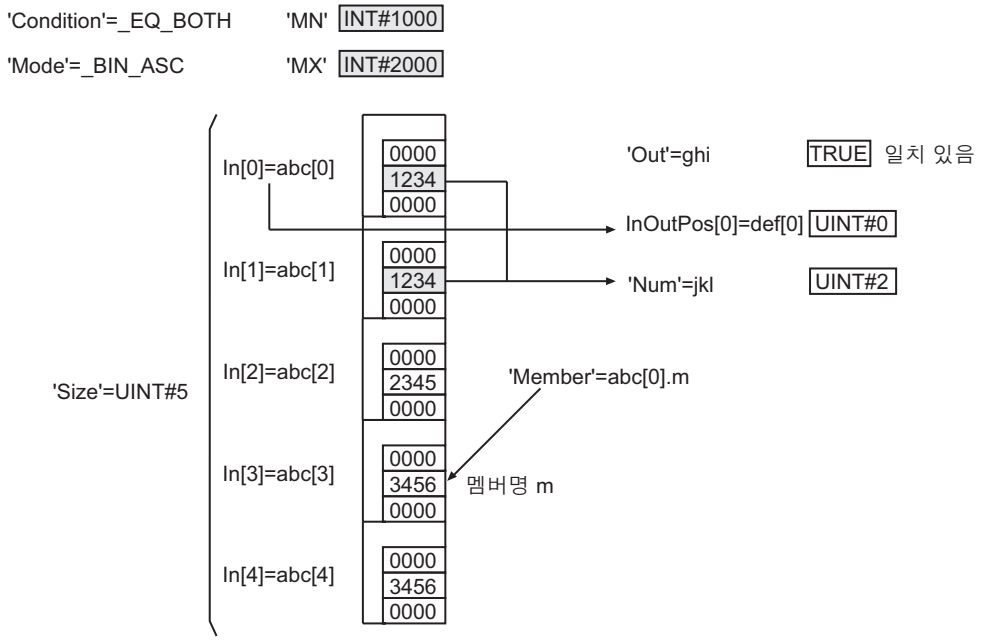
선형 탐색의 경우, In[]의 시작 요소부터 순서대로 검색을 수행합니다.

'Size'=UINT#5, 'MN'=INT#1000, 'MX'=INT#2000, 'Condition'=_EQ_BOTH, 'Mode'=_LINEAR의 예가 아래에 나와 있습니다.

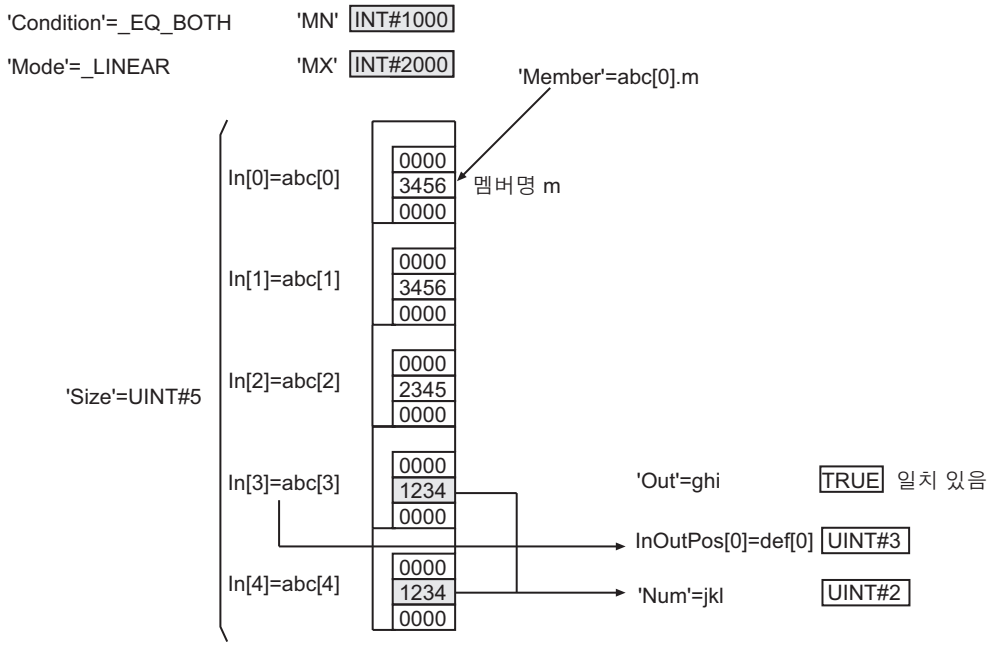


오름차순 이분 탐색의 경우, 이 명령을 실행하기 전에 In[]에 전달하는 입력 파라메터의 배열 요소를 오름차순으로 정렬해두어야 합니다. 그리고 이 명령을 실행하면 이분 탐색이 수행됩니다.

위와 동일한 예를 오름차순 이분 탐색한 경우, 배열 요소의 순서와 처리 결과는 다음과 같습니다.



내림차순 이분 탐색의 경우, 이 명령을 실행하기 전에 In[]에 전달하는 입력 파라메터의 배열 요소를 내림차순으로 정렬해두어야 합니다. 그리고 이 명령을 실행하면 이분 탐색이 수행됩니다.
위와 동일한 예를 내림차순 이분 탐색한 경우, 배열 요소의 순서와 처리 결과는 다음과 같습니다.



참고

- In[] 전체가 보다 상위의 구조체 멤버라도 상관없습니다.
  (예) In[0]=str0.str1[0]
- In[]은 2차원 이상의 배열이라도 상관없습니다. 그 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에 각각 대입됩니다.
- In[]이 3차원 배열인 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에, 제3차원 요소 번호가 InOutPos[2]에 각각 대입됩니다.

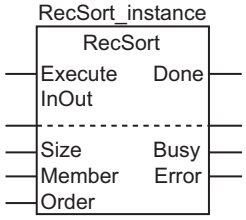
- TIME 형식, DT 형식, TOD 형식의 검색을 수행하는 경우, 'Member', 'MN', 'MX'의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'Member', 'MN', 'MX'의 데이터 형식은 In[]의 검색 대상 멤버의 데이터 형식과 동일하게 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- In[]은 구조체를 요소로 하는 배열로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- In[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Member'가 실수인 경우, 값에 따라 오차가 발생하므로 기대한 결과를 얻지 못하는 경우가 있습니다.
- 'MN', 'MX'가 실수인 경우, 비수를 지정하지 마십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 FALSE, 'Num'의 값은 0이 됩니다. InOutPos[]는 변화하지 않습니다.
- 'Mode'의 값이 _BIN_ASC(또는 _BIN_DESC)인 경우, In[]의 요소가 오름차순(또는 내림차순)으로 정렬되어 있지 않으면 올바른 결과를 얻을 수 없습니다. 이 명령 실행 전에 요소를 오름차순(또는 내림차순)으로 정렬하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', InOutPos[], 'Num'은 변화하지 않습니다.
  - 'MN' > 'MX'인 경우
  - 'Condition'의 값이 유효 범위 밖인 경우
  - 'Mode'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우
  - 'Member'가 In[]의 멤버가 아닌 경우
  - InOutPos[]의 배열 크기가 In[]의 차원 수보다 작은 경우

RecSort

구조체를 요소로 하는 배열의 요소를 정렬합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
RecSort	레코드 정렬	FB		RecSort_instance(Execute, InOut, Size, Member, Order, Done, Busy, Error);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Size	정렬 대상의 요소 수	입력	정렬 대상의 배열 요소 수	데이터 형식에 따름	–	1
Member	정렬 대상 멤버		In[]의 구조체의 정렬 대상 멤버			(*)
Order	정렬 순서		정렬 순서	_ASC, _DESC		_ASC
InOut[] 배열	정렬 대상 배열	입출력	정렬 대상이 되는 구조체를 요소로 하는 배열	–	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Size							○													
Member						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	InOut[]의 검색 대상 멤버와 동일한 데이터 형식															(*)				
Order	열거 형식 _eSORT_ORDER 열거자는 기능 설명 참조																			
InOut[] 배열	구조체를 요소로 하는 배열을 지정																			

* TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

'Execute'의 값이 TRUE일 때 구조체를 요소로 하는 배열 InOut[]의 'Size'개의 요소, 즉 InOut[0]부터 InOut['Size'-1]을 구조체의 정렬 대상 멤버 'Member'의 값으로 정렬합니다. 정렬 순서는 'Order'로 지정합니다. 자세한 내용은 뒤에서 다룹니다.
'Member'에는 In[] 가운데 한 요소의 정렬 대상 멤버를 인수로 전달합니다.

InOut[]에 전달하는 입출력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

정렬 순서 'Order'의 데이터 형식은 열거 형식 _eSORT_ORDER입니다. 열거자의 의미는 다음과 같습니다.

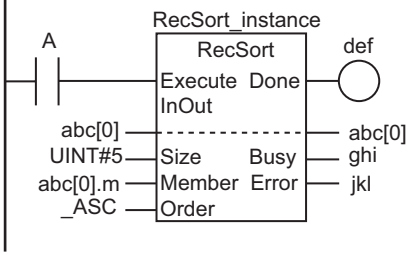
열거자	의미
_ASC	오름차순
_DESC	내림차순

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.
STRING	'LTascii / LEascii / GTascii / GEascii 명령(2-105페이지)'과 동일한 사양입니다. 해당 부분을 참조하십시오.

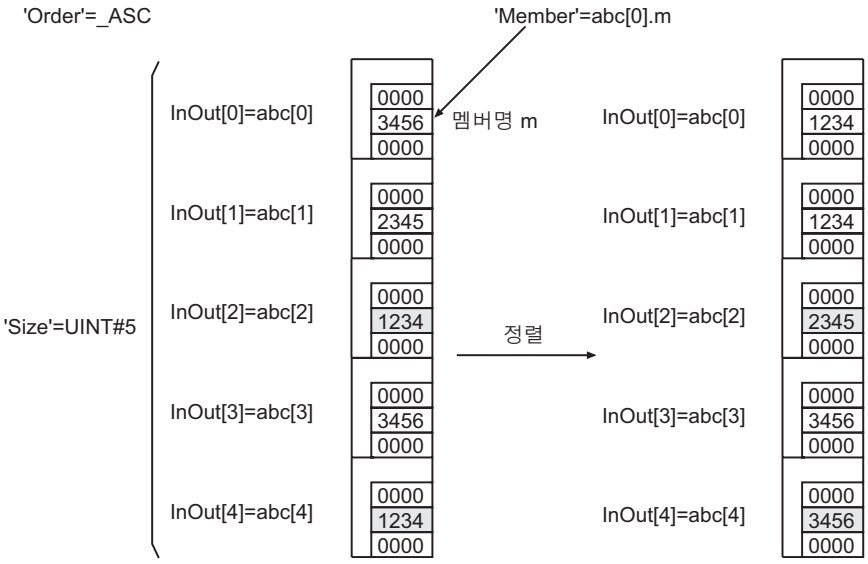
'Size'=UINT#5, 'Order'=_ASC인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
RecSort_instance(A, abc[0], UINT#5, abc[0].m, _ASC, def, ghi, jkl);
```



참고

- 이 명령 실행 중에 전원 OFF가 발생한 경우, InOut[]의 내용이 손상될 가능성이 있습니다. 그러므로, 이 명령을 실행 완료할 때마다 InOut[]의 내용을 백업해두면 손상이 발생해도 복구할 수 있습니다. 샘플 프로그램을 참조하십시오.
- TIME 형식, DT 형식, TOD 형식의 정렬을 수행하는 경우, 'Member' 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 'TruncTime 명령(2-628페이지)', 'TruncDt 명령(2-632페이지)', 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- InOut[]은 구조체를 요소로 하는 배열로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'Member'가 실수인 경우, 값에 따라 오차가 발생하므로 기대한 결과를 얻지 못하는 경우가 있습니다.
- InOut[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 'Size'의 값이 0인 경우, 'Done'의 값은 TRUE가 되며 InOut[]은 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'가 TRUE가 됩니다.
  - 'Order'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 InOut[]의 배열 영역을 초과한 경우
  - 'Member'가 InOut[]의 멤버가 아닌 경우
  - 'Member'가 STRING 형식이고 NULL 문자로 끝나지 않은 경우

샘플 프로그램

구조체 MyStr을 요소로 하는 배열 Abc[]를 오름차순으로 정렬합니다. 정렬 대상 멤버는 Abc[].m입니다. 처리 중에 전원 OFF가 발생하더라도 데이터가 손실되지 않도록, 정렬 실행 전에 Abc[]를 Abc_backup[]이라는 변수명으로 백업해둡니다. 만일 전원 OFF가 발생하면 Abc_backup[]을 Abc[]로 되돌려서 다시 정렬합니다.

글로벌 변수의 정의

데이터 형식

명칭	데이터 형식	코멘트
MyStr	STRUCT	구조체
l	BOOL	멤버
m	INT	멤버
n	REAL	멤버

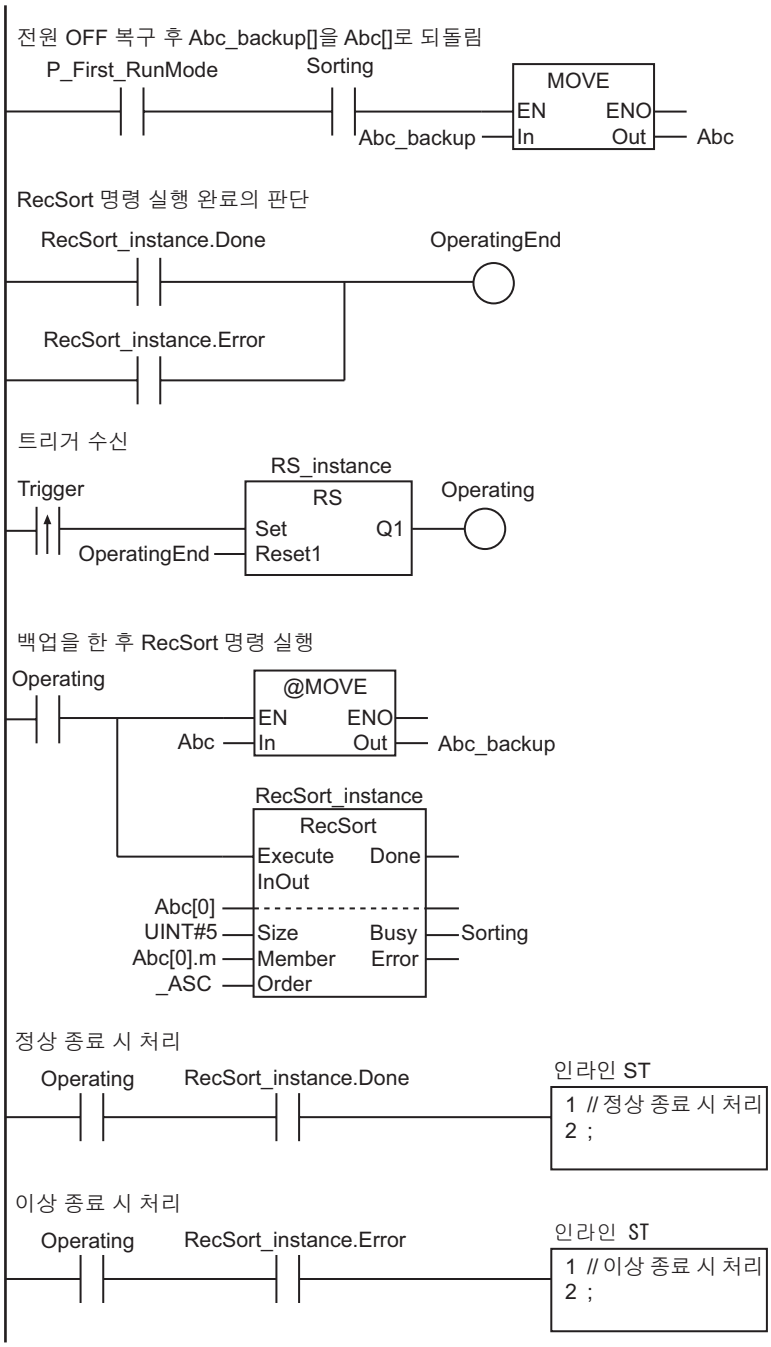
글로벌 변수

명칭	데이터 형식	초기값	유지	코멘트
Abc	ARRAY[0..4] OF MyStr	[5((l:=FALSE,m:=0,n:=0.0))]	☑	정렬 대상 배열
Abc_backup	ARRAY[0..4] OF MyStr	[5((l:=FALSE,m:=0,n:=0.0))]	☑	Abc[]의 백업

LD

내부 변수	명칭	데이터형식	초기값	유지	코멘트
	Sorting	BOOL	FALSE	☑	처리 중(유지 있음)
	OperatingEnd	BOOL	FALSE	☐	처리 종료
	Trigger	BOOL	FALSE	☐	실행 조건
	Operating	BOOL	FALSE	☐	처리 중
	RS_instance	RS		☐	
	RecSort_instance	RecSort		☐	

외부 변수	명칭	데이터 형식	코멘트
	Abc	ARRAY[0..4] OF MyStr	정렬 대상 배열
	Abc_backup	ARRAY[0..4] OF MyStr	Abc[]의 백업



ST

내부 변수	명칭	데이터 형식	초기값	유지	코멘트
	Sorting	BOOL	FALSE	☑	처리 중(유지 있음)
	Trigger	BOOL	FALSE	☐	실행 조건
	LastTrigger	BOOL	FALSE	☐	이전 테스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	☐	처리 시작
	Operating	BOOL	FALSE	☐	처리 중
	RS_instance	RS		☐	
	RecSort_instance	RecSort		☐	

외부 변수	명칭	데이터 형식	코멘트
	Abc	ARRAY[0..4] OF MyStr	정렬 대상 배열
	Abc_backup	ARRAY[0..4] OF MyStr	Abc[]의 백업

```
// 전원 OFF 복구 후, Abc_backup[]을 Abc[]로 되돌림
IF ( (P_First_RunMode = TRUE) AND (Sorting = TRUE) ) THEN
    Abc:=Abc_backup;
END_IF;

// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// RecSort 명령 초기화
IF (OperatingStart=TRUE) THEN
    Abc_backup:=Abc;
    RecSort_instance(
        Execute:=FALSE,          // 기동 조건
        InOut   :=Abc[0],        // 정렬 대상 배열
        Member:=Abc[0].m);       // 정렬 대상 멤버
    OperatingStart:=FALSE;
END_IF;

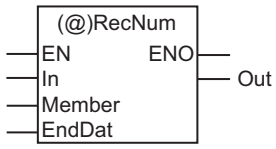
// RecSort 명령 실행
IF (Operating=TRUE) THEN
    RecSort_instance(
        Execute:=TRUE,
        InOut   :=Abc[0],
        Size    :=UINT#5,
        Member:=Abc[0].m,
        Order   :=_ASC,
        Busy    =>Sorting);

    IF (RecSort_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating:=FALSE;
    END_IF;

    IF (RecSort_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating:=FALSE;
    END_IF;
END_IF;
```

RecNum

구조체를 요소로 하는 배열의 종단 데이터까지의 레코드 수를 구합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RecNum	레코드 수 취득	FUN		Out:=RecNum(In, Member, EndDat);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	대상 배열	입력	처리 대상이 되는 구조체를 요소로 하는 배열	–	–	(*)
Member	대상 멤버		In[]의 구조체의 처리 대상 멤버	데이터 형식에 따름		
EndDat	종단 데이터		종단 데이터			
Out	레코드 수	출력	레코드 수	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다.빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	구조체를 요소로 하는 배열을 지정																			
Member	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○ (*)	○	○	○	○
	열거 형식도 지정 가능*2																			
	In[]의 처리 대상 멤버와 동일한 데이터 형식																			
EndDat	'Member'와 동일한 데이터 형식																			
Out							○													

*1. TIME 형식, DATE 형식, TOD 형식, DT 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

*2. CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 지정할 수 있습니다.

기능

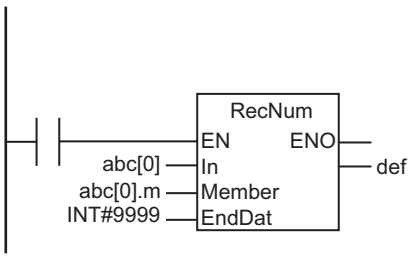
구조체를 요소로 하는 배열 In[]의 선두부터 대상 멤버 'Member'의 값이 종단 데이터 'EndDat'와 일치하는 요소를 검색합니다. 그 결과 'EndDat'와 일치한 요소 직전까지의 요소 수(레코드 수)를 'Out'에 대입합니다.

'Member'에는 In[] 가운데 한 요소의 대상 멤버를 인수를 전달합니다.

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

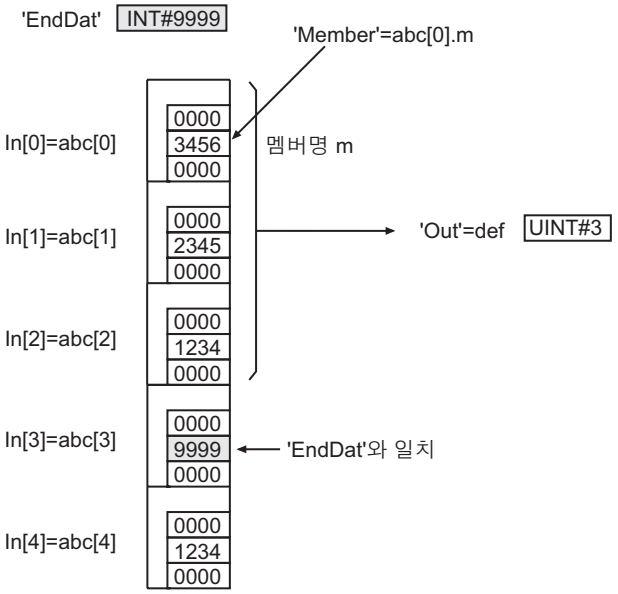
'EndDat'=INT#9999인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=RecNum(abc[0], abc[0].m, INT#9999);
```



참고

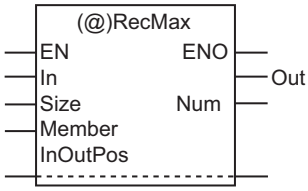
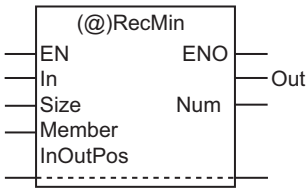
- In[] 전체가 보다 상위의 구조체 멤버라도 상관없습니다.
  (예) In[0]=str0.str1[0]
- TIME 형식, DT 형식, TOD 형식의 검색을 수행하는 경우, 'Member'와 'EndDat'의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- In[]은 구조체를 요소로 하는 배열로 하십시오. 그렇지 않으면 빌드 시에 이상 상태가 됩니다.
- 'Member'와 'EndDat'의 데이터 형식은 동일하게 설정하십시오. 서로 다르면 빌드 시에 이상 상태가 됩니다.
- In[]의 멤버 중에 'EndDat'와 일치하는 것이 없는 경우, In[]의 모든 요소 수가 'Out'에 대입됩니다.
- 'Member'가 실수인 경우, 값에 따라 오차가 발생하므로 기대한 결과를 얻지 못하는 경우가 있습니다.
- 'EndDat'가 실수인 경우, 'EndDat'에 비수를 지정하지 마십시오.
- In[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Member'가 In[]의 멤버가 아닌 경우
  - 'Member'가 STRING 형식이고 NULL 문자로 끝나지 않은 경우

RecMax / RecMin

RecMax : 구조체를 요소로 하는 배열에서 지정된 멤버의 최대값을 검색합니다.
RecMin : 구조체를 요소로 하는 배열에서 지정된 멤버의 최소값을 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
RecMax	레코드 최대값 검색	FUN		Out:=RecMax(In, Size, Member, InOutPos, Num);
RecMin	레코드 최소값 검색	FUN		Out:=RecMin(In, Size, Member, InOutPos, Num);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	검색 대상 배열	입력	검색 대상이 되는 구조체를 요소로 하는 배열	–	–	(*)
Size	검색 대상의 요소 수		검색 대상의 배열 요소 수	데이터 형식에 따름		1
Member	검색 대상 멤버		In[]의 구조체의 검색 대상 멤버			(*)
InOutPos[] 배열	검색 결과의 요소 번호	입출력	검색 결과의 요소 번호	데이터 형식에 따름	–	–
Out	검색 결과	출력	검색 결과	데이터 형식에 따름	–	–
Num	검색 개수		검색 개수			

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열	구조체를 요소로 하는 배열을 지정																			
Size							○													
Member						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	In[]의 검색 대상 멤버와 동일한 데이터 형식															(*)				
InOutPos[] 배열							○													
Out						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
																(*)				
Num							○													

* TIME 형식, DATE 형식, TOD 형식, DT 형식, STRING 형식은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 지정할 수 있습니다.

기능

구조체를 요소로 하는 배열 In[]의 'Size'개의 요소, 즉 In[0]부터 In['Size'-1] 중에서 구조체의 검색 대상 멤버 'Member'의 값을 검색합니다.
'Member'에는 In[] 가운데 한 요소의 검색 대상 멤버를 인수로 전달합니다.
검색 결과의 요소 번호가 InOutPos[0]에, 검색 요소 수가 'Num'에 각각 대입됩니다. 검색 결과가 2개 이상 있는 경우에는 In[] 중에서 가장 하위에 있는 검색 결과의 요소 번호가 InOutPos[0]에 대입됩니다.

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

정수, 실수 이외의 데이터 형식의 값에 대한 크기 관계는 다음과 같이 판단합니다.

데이터 형식	크기 관계
TIME	값이 큰 쪽이 크다고 판단합니다.
DATE, TOD, DT	날짜 또는 시각이 더 뒤인 쪽을 크다고 판단합니다.
STRING	📖 'LTascii / LEascii / GTascii / GEascii 명령(2-105페이지)'과 동일한 사양입니다. 해당 부분을 참조하십시오.

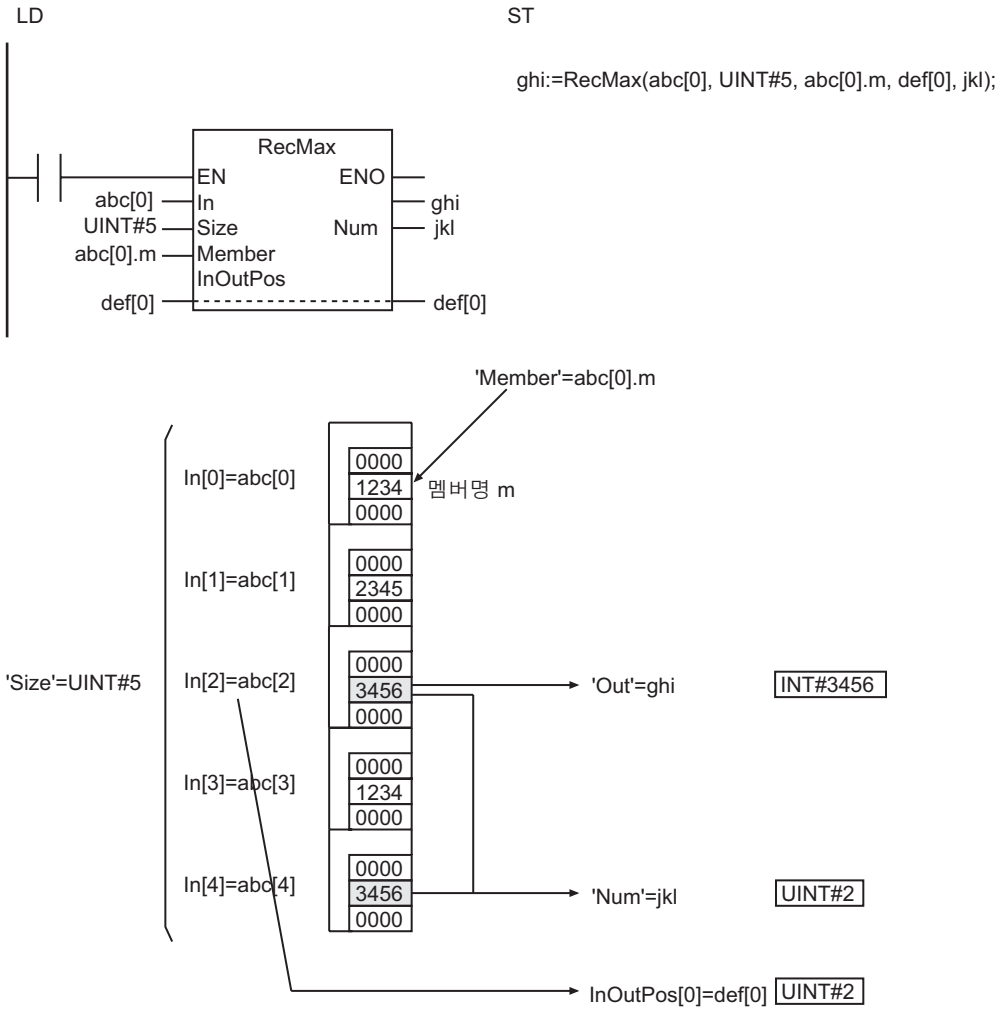
● RecMax

최대값 검색을 수행합니다. 검색 결과 'Out'에는 검색 대상 멤버의 최대값이 대입됩니다.

● RecMin

최소값 검색을 수행합니다. 검색 결과 'Out'에는 검색 대상 멤버의 최소값이 대입됩니다.

RecMax 명령에서 'Size'=UINT#5인 경우의 예가 아래에 나와 있습니다.



참고

- In[] 전체가 보다 상위의 구조체 멤버라도 상관없습니다.
  (예) In[0]=str0.str1[0]
- In[]은 2차원 이상의 배열이라도 상관없습니다. 그 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에 각각 대입됩니다.
- In[]이 3차원 배열인 경우, 검색 조건과 일치하는 요소의 제1차원 요소 번호가 InOutPos[0]에, 제2차원 요소 번호가 InOutPos[1]에, 제3차원 요소 번호가 InOutPos[2]에 각각 대입됩니다.
- TIME 형식, DT 형식, TOD 형식의 검색을 수행하는 경우, 'Member' 간의 정밀도를 맞추십시오. 값의 정밀도를 맞추기 위한 명령에는 🕮 'TruncTime 명령(2-628페이지)', 🕮 'TruncDt 명령(2-632페이지)', 🕮 'TruncTod 명령(2-636페이지)'이 있습니다.

사용상의 주의 사항

- 'Member'와 'Out'의 데이터 형식이 다른 경우, 다음 데이터 형식 중에서 'Out'의 유효 범위에 'Member'의 유효 범위가 포함되도록 선택하십시오.
  - USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL
- 'Member'가 실수인 경우, 값에 따라 오차가 발생하므로 기대한 결과를 얻지 못하는 경우가 있습니다.
- In[]에 배열 중간의 요소를 전달한 경우, 그 요소 이후가 처리 대상이 됩니다.

- 'In'이 열거 형식인 경우, 'In'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 이상이 됩니다.
- 'Size'의 값이 0인 경우, 'Out', 'Num'의 값은 0이 됩니다. InOutPos[]의 값은 변화하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out', InOutPos[], 'Num'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우
  - 'Member'가 In[]의 멤버가 아닌 경우
  - InOutPos[]의 배열 크기가 In[]의 차원 수보다 작은 경우
  - 'Member'가 STRING 형식이고 NULL 문자로 끝나지 않은 경우

FCS 명령

명령	명칭	페이지
StringSum	SUM값 산출	2-510
StringLRC	LRC값 산출(문자열)	2-512
StringCRCCCITT	CRC-CCITT값 산출(문자열)	2-514
StrCRC16	CRC-16값 산출(문자열)	2-516
AryLRC_**	LRC값 산출(배열) 그룹	2-518
AryCRCCCITT	CRC-CCITT값 산출(배열)	2-520
AryCRC16	CRC-16값 산출(배열)	2-522

StringSum

문자열의 SUM값을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StringSum	SUM값 산출	FUN	(@)StringSum EN ENO In Out Size	Out:=StringSum(In, Size);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	대상 문자열	입력	대상 문자열	데이터 형식에 따름	–	''
Size	바이트 크기		SUM값의 바이트 크기	1,2	바이트	1
Out	SUM값	출력	SUM값	'Size'로 지정된 바이트 수	바이트	–

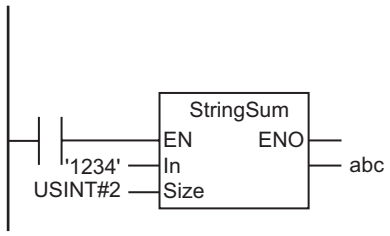
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Size						○														
Out																				○

기능

대상 문자열 'In'의 SUM값(각 문자의 문자 코드의 총합)을 산출합니다. SUM값 'Out'은 바이트 크기 'Size'로 지정한 바이트 수가 됩니다.
'Out'은 16진수 문자열로 표현되며 끝에는 NULL 문자가 저장됩니다.

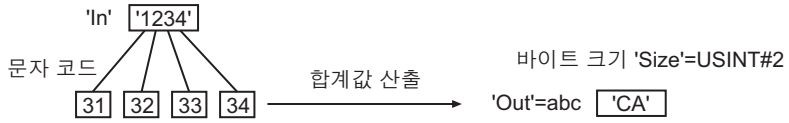
'In'='1234', 'Size'=USINT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=StringSum('1234', USINT#2);
```



위의 예에서 'Size'=USINT#1인 경우 'Out'의 값은 'A'가 됩니다.

사용상의 주의 사항

- 'In'의 문자 코드의 총합이 'Size'로 표현할 수 있는 자릿수를 초과하는 경우에는 상위 자리를 버립니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'In'의 바이트 수가 0(NULL 문자만)인 경우

StringLRC

문자열의 LRC값(수평 패리티)을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StringLRC	LRC값 산출 (문자열)	FUN	(@)StringLRC EN ENO In Out	Out:=StringLRC(In);

변수

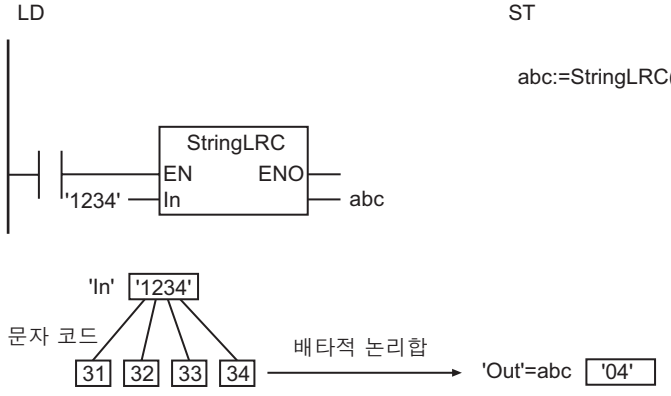
	명칭	입/출력	내용	유효 범위	단위	초기값
In	대상 문자열	입력	대상 문자열	데이터 형식에 따름	–	''
Out	LRC값	출력	LRC값	최대 3바이트(반각 영숫자 2문자+끝의 NULL 문자)	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out																				○

기능

대상 문자열 'In'의 LRC값(수평 패리티)을 산출합니다. LRC값이란 'In'의 각 문자의 문자 코드의 배타적 논리합을 말합니다.
LRC값 'Out'은 16진수 문자열로 표현되며, 끝에는 NULL 문자가 저장됩니다.

'In'='1234'인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'의 바이트 수가 0(NULL 문자만)인 경우

StringCRCCCITT

문자열의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
String CRCCCITT	CRC-CCITT값 산출 (문자열)	FUN	(@)StringCRCCCITT EN ENO In Out Initial OutOrder	Out:=StringCRCCCITT(In, Initial, OutOrder);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	대상 문자열	입력	대상 문자열	데이터 형식에 따름	–	''
Initial	초기값		CRC-CCITT값 산출의 초기값			0
OutOrder	바이트 순서		'In'을 처리하는 바이트 순서	_LOW_HIGH, _HIGH_LOW		_HIGH _LOW
Out	CRC-CCITT값	출력	CRC-CCITT값	5바이트(반각 영숫자 4문자+끝의 NULL 문자)	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Initial			○																	
OutOrder	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
Out																				○

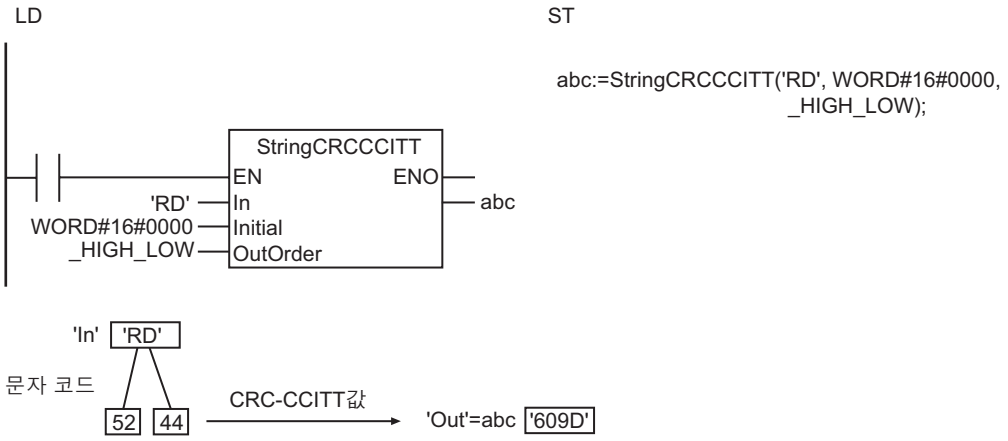
기능

대상 문자열 'In'의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.
CRC-CCITT값 'Out'은 16진수 문자열로 표현되며, 끝에는 NULL 문자가 저장됩니다.

CRC-CCITT값을 산출할 때의 초기값은 'Initial'로 지정합니다. 바이트 순서는 'OutOrder'로 지정합니다.
'OutOrder'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

'In'='RD', 'Initial'=WORD#16#0000, 'OutOrder'=_HIGH_LOW인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'OutOrder'의 값이 유효 범위 밖인 경우
- 'In'의 바이트 수가 0(NULL 문자만)인 경우

StringCRC16

문자열의 CRC-16값을 MODBUS 방식으로 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
StringCRC16	CRC-16값 산출 (문자열)	FUN	(@)StringCRC16 EN ENO In Out Initial OutOrder	Out:=StringCRC16(In, Initial, OutOrder);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	대상 문자열	입력	대상 문자열	데이터 형식에 따름	–	"
Initial	초기값		CRC-16값 산출의 초기값			16#FFFF
OutOrder	바이트 순서		'In'을 처리하는 바이트 순서	_LOW_HIGH, _HIGH_LOW		_LOW_HIGH
Out	CRC-16값	출력	CRC-16값	5바이트(반각 영숫자 4문자+끝의 NULL 문자)	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Initial			○																	
OutOrder	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
Out																				○

기능

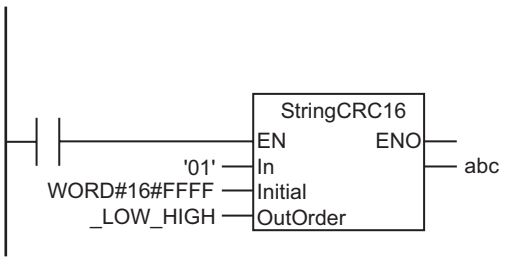
대상 문자열 'In'의 CRC-16값을 MODBUS 방식으로 산출합니다.
CRC-16값 'Out'은 16진수 문자열로 표현되며, 끝에는 NULL 문자가 저장됩니다.

CRC-16값을 산출할 때의 초기값은 'Initial'로 지정합니다. 바이트 순서는 'OutOrder'로 지정합니다.
'OutOrder'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

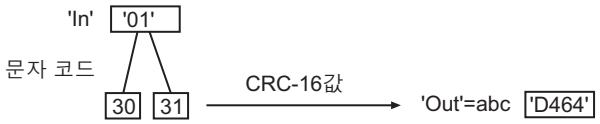
'In'='01', 'Initial'=WORD#16#FFFF, 'OutOrder'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=StringCRC16('01', WORD#16#FFFF,
                 _LOW_HIGH);
```



사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'OutOrder'의 값이 유효 범위 밖인 경우
- 'In'의 바이트 수가 0(NULL 문자만)인 경우

AryLRC_**

배열의 LRC값을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryLRC_**	LRC값 산출 (배열) 그룹	FUN	(@)AryLRC_** EN ENO In Out Size 단, **는 비트 열의 데이터 형식명	Out: =AryLRC_**(In, Size); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	연산 대상의 요소 수		In[]의 요소 수			1
Out	LRC값	출력	LRC값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

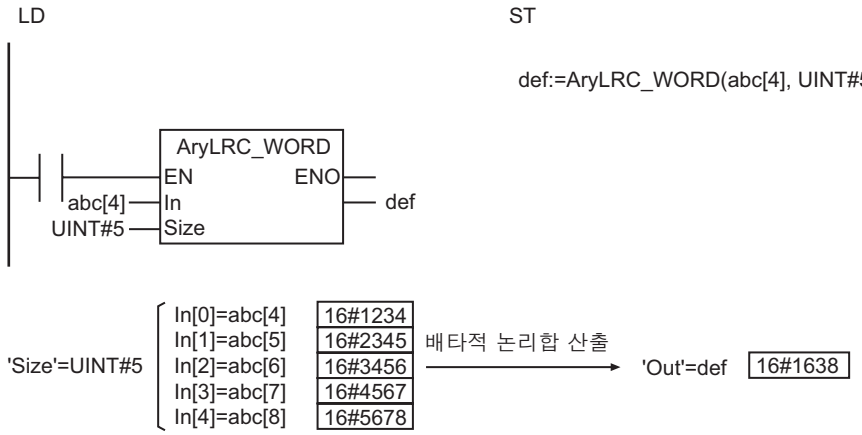
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○	○	○	○															
Size							○													
Out	In[]과 동일한 데이터 형식																			

기능

연산 대상 배열 In[]에서 In[0] 이후의 'Size'개의 배열 요소의 LRC값(배타적 논리합)을 산출합니다.
명령 이름은 In[]의 데이터 형식에 따라 다릅니다. 예를 들면, In[]이 WORD 형식인 경우, 명령 이름은 AryLRC_WORD가 됩니다.

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

AryLRC_WORD 명령에서 'Size'=UINT#5인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- In[]과 'Out'의 데이터 형식을 동일하게 하십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 16#00이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

AryCRCCCITT

배열의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCRC CCITT	CRC-CCITT값 산출(배열)	FUN	(@)AryCRCCCITT EN ENO In Out Size Initial OutOrder	Out:=AryCRCCCITT(In, Size, Initial, OutOrder);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	연산 대상의 요소 수		In[]의 요소 수			1
Initial	초기값		CRC-CCITT값 산출의 초기값			0
OutOrder	바이트 순서		'In'을 처리하는 바이트 순서	_LOW_HIGH, _HIGH_LOW		_HIGH _LOW
Out	CRC-CCITT값	출력	CRC-CCITT값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Size							○													
Initial			○																	
OutOrder	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
Out			○																	

기능

연산 대상 배열 In[]에서 In[0] 이후의 'Size'개의 배열 요소의 CRC-CCITT값을 XMODEM 방식으로 산출합니다.

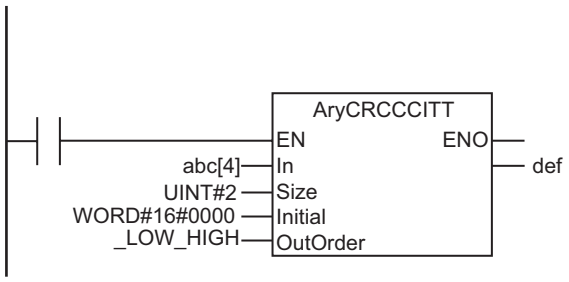
CRC-CCITT값을 산출할 때의 초기값은 'Initial'로 지정합니다. 바이트 순서는 'OutOrder'로 지정합니다. 'OutOrder'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

'Size'=UINT#2, 'Initial'=WORD#16#0000, 'OutOrder'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=AryCRCCCITT(abc[4], UINT#2,
                 WORD#16#0000,
                 _LOW_HIGH);
```



사용상의 주의 사항

- 'Size'의 값이 0인 경우, 'Out'의 값은 WORD#16#0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'OutOrder'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

AryCRC16

배열의 CRC-16값을 MODBUS 방식으로 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AryCRC16	CRC-16값 산출 (배열)	FUN	(@)AryCRC16 EN ENO In Out Size Initial OutOrder	Out:=AryCRC16(In, Size, Initial, OutOrder);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In[] 배열	연산 대상 배열	입력	연산 대상 배열	데이터 형식에 따름	–	(*)
Size	연산 대상의 요소 수		In[]의 요소 수			1
Initial	초기값		CRC-16값 산출의 초기값			16#FFFF
OutOrder	바이트 순서		'In'을 처리하는 바이트 순서	_LOW_HIGH, _HIGH_LOW		_LOW _HIGH
Out	CRC-16값	출력	CRC-16값	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In[] 배열		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Size							○													
Initial			○																	
OutOrder	열거 형식 _eBYTE_ORDER 열거자는 기능 설명 참조																			
Out			○																	

기능

연산 대상 배열 In[]에서 In[0] 이후의 'Size'개의 배열 요소의 CRC-16값을 MODBUS 방식으로 산출합니다.

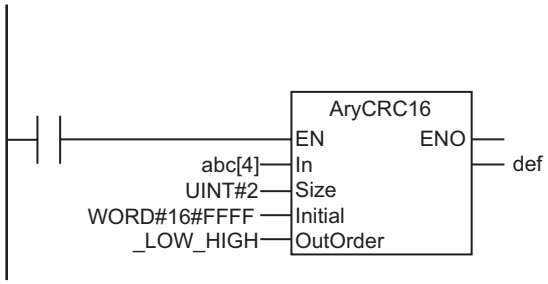
CRC-16값을 산출할 때의 초기값은 'Initial'로 지정합니다. 바이트 순서는 'OutOrder'로 지정합니다.
'OutOrder'의 데이터 형식은 열거 형식 _eBYTE_ORDER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_LOW_HIGH	하위 바이트가 먼저, 상위 바이트가 나중
_HIGH_LOW	상위 바이트가 먼저, 하위 바이트가 나중

In[]에 전달하는 입력 파라메터는 array[3]과 같이 반드시 요소 번호도 포함해서 지정합니다.

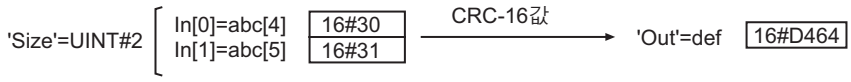
'Size'=UINT#2, 'Initial'=WORD#16#FFFF, 'OutOrder'=_LOW_HIGH인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
def:=AryCRC16(abc[4], UINT#2,
              WORD#16#FFFF,
              _LOW_HIGH);
```



사용상의 주의 사항

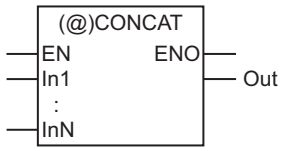
- 'Size'의 값이 0인 경우, 'Out'의 값은 WORD#16#0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'OutOrder'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 In[]의 배열 영역을 초과한 경우

문자열 명령

명령	명칭	페이지
CONCAT	문자열 · 연결	2-526
LEFT/RIGHT	문자열 · 좌측에서 가져오기 문자열 · 우측에서 가져오기	2-528
MID	문자열 · 임의의 위치에서 가져오기	2-530
FIND	문자열 · 검색	2-532
LEN	문자열 · 길이 검출	2-534
REPLACE	문자열 · 치환	2-535
DELETE	문자열 · 삭제	2-537
INSERT	문자열 · 삽입	2-539
GetByteLen	문자열 · 바이트 수 취득	2-541
ClearString	문자열 · 클리어	2-542
ToUCase/ToLCase	문자열 · 대문자 변환 문자열 · 소문자 변환	2-543
TrimL/TrimR	문자열 · 좌측 트림 문자열 · 우측 트림	2-545
AddDelimiter	구분 문자 부가 문자열 쓰기	2-547
SubDelimiter	문자열 읽기 구분 문자 삭제	2-558

CONCAT

2~5개의 문자열을 연결합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CONCAT	문자열 · 연결	FUN		Out:=CONCAT(In1, ⋯, InN);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1~InN	연결 대상	입력	연결할 문자열, N은 2~5	데이터 형식에 따름	–	''(*)
Out	연결 결과	출력	연결 결과 문자열	데이터 형식에 따름	–	–

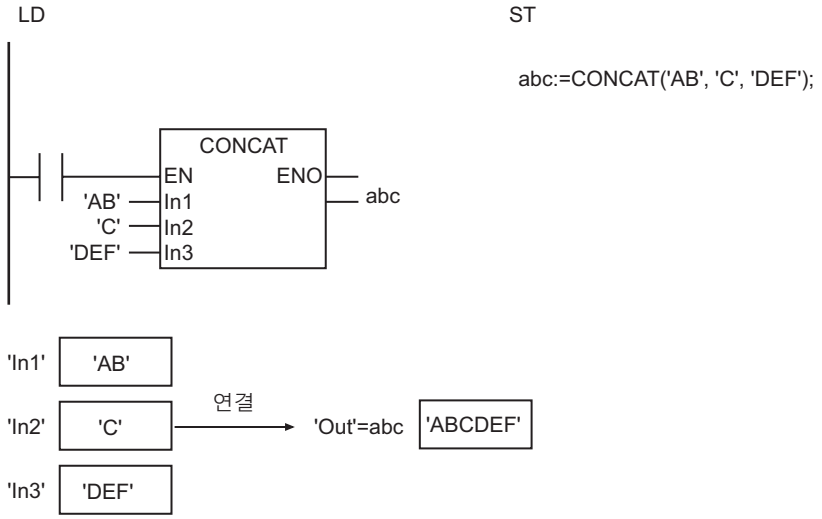
* InN에 연결할 입력 파라메터를 생략한 경우, 초기값은 적용되지 않고 빌드 시 이상이 발생합니다. 예를 들어, N=3인 경우 In1과 In2에 연결할 입력 파라메터를 생략하면 초기값이 적용되지만 In3에 연결할 입력 파라메터를 생략하면 빌드 시 이상이 발생합니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1~InN																				○
Out																				○

기능

2~5개의 연결 대상 문자열 'In1'~'InN'을 이 순서대로 연결합니다. 끝에는 NULL 문자가 붙습니다.

'In1'='AB', 'In2'='C', 'In3'='DEF'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'ABCDEF'가 됩니다.



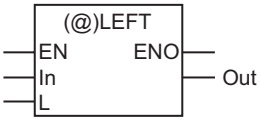
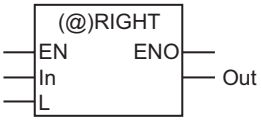
사용상의 주의 사항

다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 연결한 문자열의 길이가 1986바이트를 초과하는 경우

LEFT / RIGHT

문자열에서 지정된 수의 문자열을 가져옵니다.
LEFT : 문자열의 좌측(선두)에서 가져옵니다.
RIGHT : 문자열의 우측(끝)에서 가져옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LEFT	문자열 · 좌측에서 가져오기	FUN		Out:=LEFT(In, L);
RIGHT	문자열 · 우측에서 가져오기	FUN		Out:=RIGHT(In, L);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	가져올 대상	입력	가져올 대상 문자열	데이터 형식에 따름	–	''
L	문자 수		가져올 문자 수	0~1985		1
Out	가져온 결과	출력	가져온 결과 문자열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
L							○													
Out																				○

기능

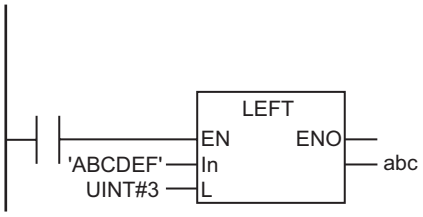
가져올 대상 문자열 'In'에서 문자 수 'L'로 지정한 문자 수의 문자열을 가져옵니다. 가져온 결과 'Out'의 끝에는 NULL 문자를 부가합니다.

● LEFT

'In'의 좌측(선두)에서 가져옵니다.

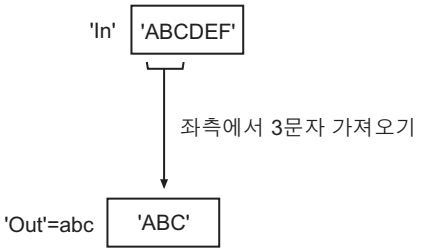
'In'='ABCDEF', 'L'=UINT#3인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'ABC'가 됩니다.

LD



ST

```
abc:=LEFT('ABCDEF', UINT#3);
```

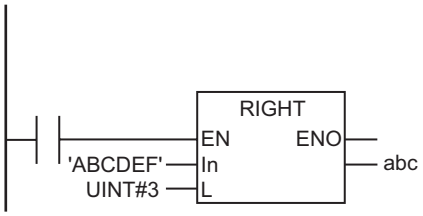


● RIGHT

'In'의 우측(끝)에서 가져옵니다.

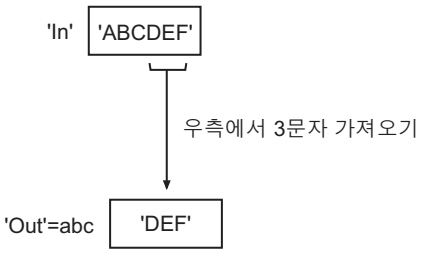
'In'='ABCDEF', 'L'=UINT#3인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'DEF'가 됩니다.

LD



ST

```
abc:=RIGHT('ABCDEF', UINT#3);
```

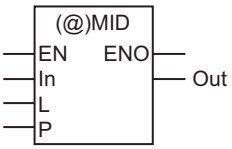


사용상의 주의 사항

- 'L'의 값이 'In'의 문자 수보다 크거나 유효 범위 내에 있는 경우, 이상 상태가 되지 않으며 'In'의 모든 문자가 'Out'에 복사됩니다.
- 'L'의 값이 0인 경우, 이상 상태가 되지는 않으며 NULL 문자만 'Out'에 대입합니다.
- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'에 문자 코드 이상이 있는 경우

MID

문자열의 임의의 위치로부터 지정 수의 문자열을 가져옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MID	문자열 · 임의의 위치에서 가져오기	FUN		Out:=MID(In, L, P);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	가져올 대상	입력	가져올 대상 문자열	데이터 형식에 따름	–	''
L	문자 수		가져올 문자 수	0~1985		1
P	가져올 시작 위치		가져올 시작 위치	1~1985		
Out	가져온 결과	출력	가져온 결과 문자열	데이터 형식에 따름	–	–

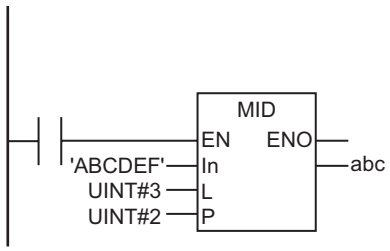
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
L							○													
P							○													
Out																				○

기능

대상 문자열 'In'에서 문자 수 'L'로 지정한 수의 문자열을 가져옵니다. 가져올 시작 위치는 시작 위치 'P'로 지정합니다. 가져온 결과 'Out'의 끝에는 NULL 문자를 부가합니다.

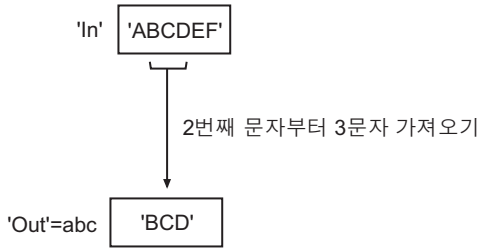
'In'='ABCDEF', 'L'=UINT#3, 'P'=UINT#2인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'BCD'가 됩니다.

LD



ST

```
abc:=MID('ABCDEF', UINT#3, UINT#2);
```

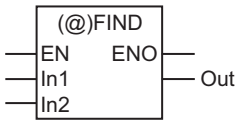


사용상의 주의 사항

- 'L'의 값이 0인 경우, 이상 상태가 되지는 않으며 NULL 문자만 'Out'에 대입합니다.
- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'에 문자 코드 이상이 있는 경우
  - 'In'에 'P'의 값으로 지정한 위치부터 'L'의 값으로 지정한 수만큼의 문자가 없는 경우
  - 'P'의 값이 0인 경우

FIND

문자열 중에서 지정 문자열의 위치를 검색합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FIND	문자열 · 검색	FUN		Out:=FIND(In1, In2);

변수

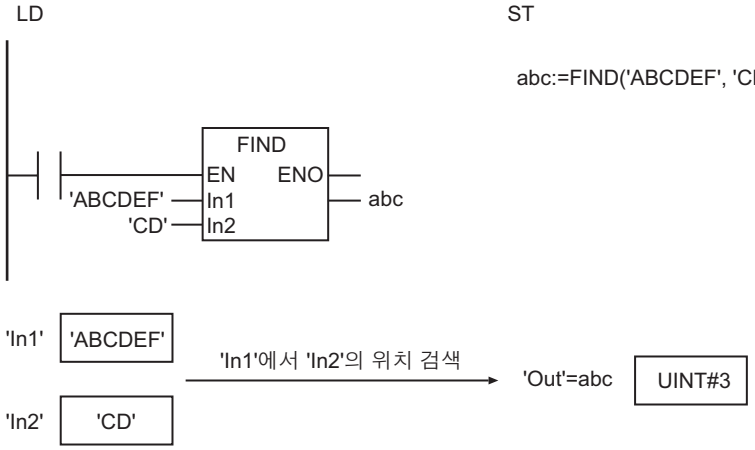
	명칭	입/출력	내용	유효 범위	단위	초기값
In1	검색 대상	입력	검색 대상 문자열	데이터 형식에 따름	–	''
In2	검색 문자열		검색할 문자열			
Out	검색 결과	출력	검색 결과	0~1985	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				○
In2																				○
Out							○													

기능

검색 대상 'In1' 중에서 검색 문자열 'In2'의 위치를 검색합니다. 검색 결과 'Out'에는 'In2'가 'In1'의 선두에서 몇 번째 문자에 있는지가 대입됩니다.
'In1'에 'In2'가 존재하지 않는 경우, 'Out'의 값은 0이 됩니다.

'In1'='ABCDEF', 'In2'='CD'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 UINT#3이 됩니다.



사용상의 주의 사항

- 'In2'는 'In1'보다 문자 수를 작게 지정하십시오. 그렇지 않은 경우, 'Out'의 값은 0이 됩니다.
- 'In1'에 'In2'가 여러 개 포함되어 있는 경우, 'Out'의 값은 'In1'의 선두부터 검색하여 처음 나타나는 'In2'의 위치가 됩니다.
- 'In1', 'In2'의 값이 모두 NULL 문자인 경우, 'Out'의 값은 1이 됩니다.
- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In1' 또는 'In2'에 문자 코드 이상이 있는 경우

LEN

문자열의 문자 수를 구합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
LEN	문자열 · 길이 검출	FUN	(@)LEN EN ENO In Out	Out:=LEN(In);

변수

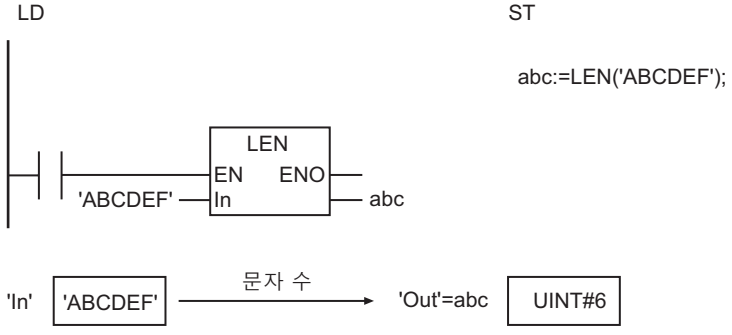
	명칭	입/출력	내용	유효 범위	단위	초기값
In	길이 검출 대상	입력	길이 검출 대상 문자열	데이터 형식에 따름	–	''
Out	검출 결과	출력	길이 검출 결과	0~1985	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out							○													

기능

길이 검출 대상 문자열 'In'의 문자 수를 구합니다. 'In'의 끝에 있는 NULL 문자는 문자 수에 포함되지 않습니다.

'In'='ABCDEF'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 UINT#6이 됩니다.

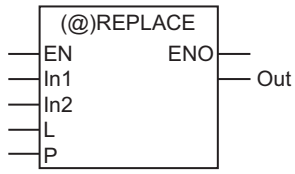


사용상의 주의 사항

- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'에 문자 코드 이상이 있는 경우

REPLACE

문자열의 일부를 다른 문자열로 치환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
REPLACE	문자열 · 치환	FUN		Out:=REPLACE(In1, In2, L, P);

변수

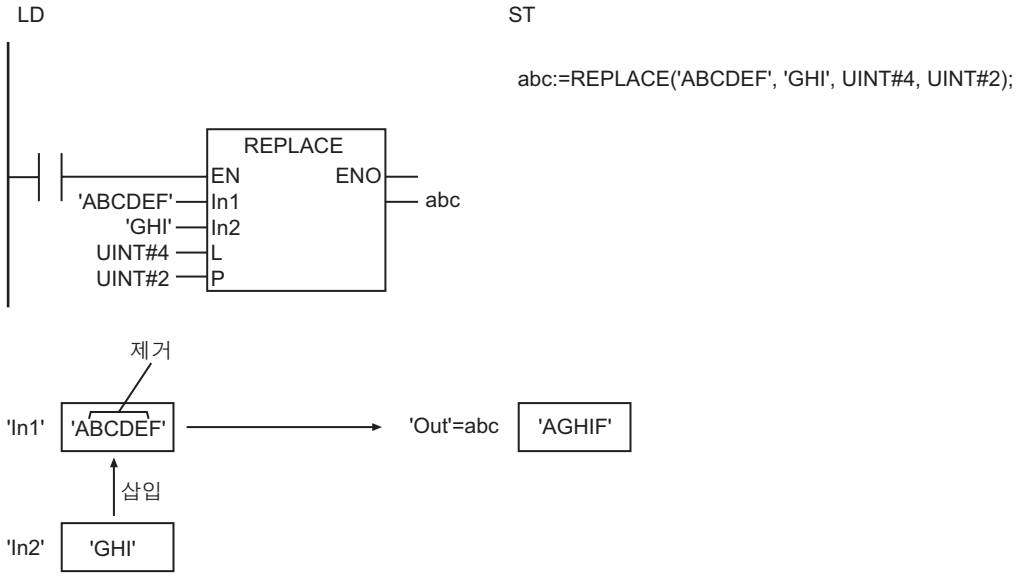
	명칭	입/출력	내용	유효 범위	단위	초기값
In1	치환 대상	입력	치환 대상 문자열	데이터 형식에 따름	–	''
In2	치환 문자열		치환할 문자열			
L	문자 수		치환할 문자 수	0~1985		1
P	치환 시작 위치		치환을 시작할 위치	1~1985		
Out	치환 결과	출력	치환 후의 문자열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				○
In2																				○
L							○													
P							○													
Out																				○

기능

치환 대상 문자열 'In1'의 일부를 치환 문자열 'In2'로 치환합니다.
우선 'In1'에서 치환 시작 위치 'P'의 값으로 지정한 위치부터 문자 수 'L'의 값만큼의 문자를 제거합니다.
빈 위치에 'In2'를 채워 넣습니다. 치환 결과 'Out'의 끝에는 NULL 문자가 붙습니다.

'In1'='ABCDEF', 'In2'='GHI', 'P'=UINT#2, 'L'=UINT#4인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'AGHIF'가 됩니다.

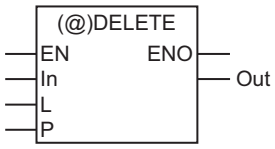


사용상의 주의 사항

- 'L'의 값이 0인 경우, 이상 상태가 되지는 않으며 'In1'의 모든 문자가 'Out'에 복사됩니다.
- 'In2'의 값이 0인 경우, 'In1'의 선두에서 'P'번째 문자부터 'L' 문자만큼 삭제됩니다.
- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In1'에 문자 코드 이상이 있는 경우
  - 'In1'에 'P'의 값으로 지정한 위치부터 'L'의 값으로 지정한 수만큼의 문자가 없는 경우
  - 'P'의 값이 0인 경우
  - 치환 후의 문자열의 길이가 1986바이트를 초과하는 경우

DELETE

문자열의 일부 또는 전체를 삭제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DELETE	문자열 · 삭제	FUN		Out:=DELETE(In, L, P);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	삭제 대상	입력	삭제 대상 문자열	데이터 형식에 따름	–	"
L	문자 수		삭제할 문자 수	0~1985		1
P	삭제 시작 위치		삭제를 시작할 위치	1~1985		
Out	삭제 결과	출력	삭제 후의 문자열	데이터 형식에 따름	–	–

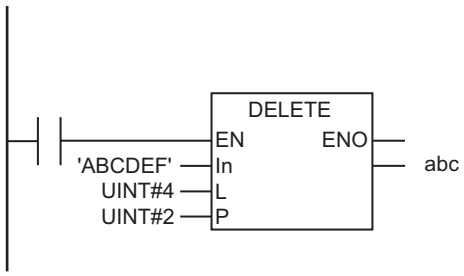
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
L							○													
P							○													
Out																				○

기능

삭제 대상 문자열 'In'에서 삭제 시작 위치 'P'의 값으로 지정한 위치부터 문자 수 'L'의 값만큼의 문자를 삭제합니다. 삭제 결과 'Out'의 끝에는 NULL 문자가 붙습니다.

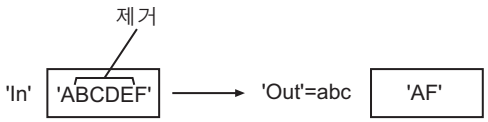
'In'='ABCDEF', 'L'=UINT#4, 'P'=UINT#2인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'AF'가 됩니다.

LD



ST

```
abc:=DELETE('ABCDEF', UINT#4, UINT#2);
```

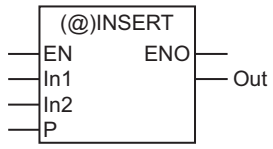


사용상의 주의 사항

- 'L'의 값이 0인 경우, 이상 상태가 되지는 않으며 'In'의 모든 문자가 'Out'에 복사됩니다.
- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'에 문자 코드 이상이 있는 경우
  - 'In'에 'P'의 값으로 지정한 위치부터 'L'의 값으로 지정한 수만큼의 문자가 없는 경우
  - 'P'의 값이 0인 경우

INSERT

문자열에 다른 문자열을 삽입합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
INSERT	문자열 · 삽입	FUN		Out:=INSERT(In1, In2, P);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	삽입 대상	입력	삽입 대상 문자열	데이터 형식에 따름	–	"
In2	삽입 문자열		삽입할 문자열			
P	삽입 시작 위치		삽입을 시작할 위치	0~1985		0
Out	삽입 결과	출력	삽입 후의 문자열	데이터 형식에 따름	–	–

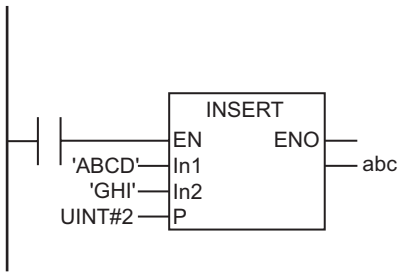
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				○
In2																				○
P							○													
Out																				○

기능

삽입 대상 문자열 'In1'에서 삽입 시작 위치 'P'의 값으로 지정한 위치에 삽입 문자열 'In2'를 삽입합니다. 삽입 결과 'Out'의 끝에는 NULL 문자가 붙습니다.

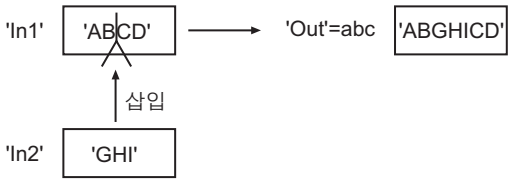
'In1'='ABCD', 'In2'='GHI', 'P'=UINT#2인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'ABGHICD'가 됩니다.

LD



ST

```
abc:=INSERT('ABCD', 'GHI', UINT#2);
```



참고

'P'의 값이 0인 경우는 'In2'의 뒤에 'In1'이 연결됩니다.

사용상의 주의 사항

- 다중 바이트 문자도 1문자로 계산합니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In1'에 문자 코드 이상이 있는 경우
  - 'P'의 값이 'In1'의 문자 수보다 큰 경우
  - 삽입 후의 문자열의 길이가 1986바이트를 초과하는 경우

GetByteLen

문자열의 바이트 수를 카운트합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetByteLen	문자열 · 바이트 수 취득	FUN	(@)GetByteLen EN ENO In Out	Out:=GetByteLen(In);

변수

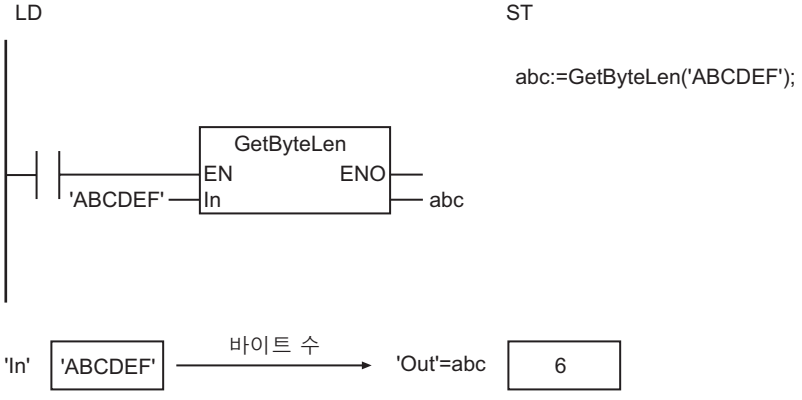
	명칭	입/출력	내용	유효 범위	단위	초기값
In	카운트 대상	입력	바이트 수를 카운트할 대상 문자열	데이터 형식에 따름	–	''
Out	바이트 수	출력	바이트 수	0~1985	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out							○													

기능

카운트 대상 문자열 'In'의 시작부터 끝까지의 바이트 수를 카운트합니다. 문자열 끝의 NULL 문자는 바이트 수에 포함되지 않습니다.

'In'='ABCDEF'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 6이 됩니다.

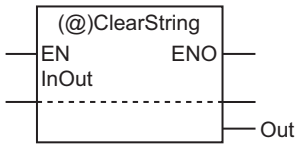


참고

'In'에 ASCII 문자만 포함되어 있는 경우, 이 명령의 결과는 LEN 명령의 결과와 일치합니다.

ClearString

문자열을 클리어합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ClearString	문자열 · 클리어	FUN		ClearString(InOut);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
InOut	클리어 문자열	입출력	클리어할 문자열	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

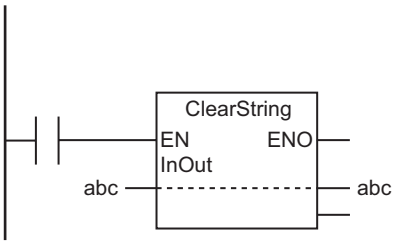
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
InOut																				○
Out	○																			

기능

클리어 문자열 'InOut'을 클리어합니다. 'InOut'의 모든 영역에 NULL 문자를 저장합니다.

기술 예가 아래에 나와 있습니다. STRING 형식 변수 abc의 값이 모두 NULL 문자가 됩니다.

LD



ST

```
ClearString(abc);
```

'InOut'의 모든 영역에 NULL 문자를 저장합니다.

abc가 5문자의 STRING 형식 변수인 경우

'InOut'=abc

NULL	NULL	NULL	NULL	NULL

사용상의 주의 사항

이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용되지 않습니다.

ToUCase/ToLCase

ToUCase : 문자열 중의 반각 알파벳을 모두 대문자로 변환합니다.
ToLCase : 문자열 중의 반각 알파벳을 모두 소문자로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ToUCase	문자열 · 대문자 변환	FUN	(@)ToUCase EN ENO In Out	Out:=ToUCase(In);
ToLCase	문자열 · 소문자 변환	FUN	(@)ToLCase EN ENO In Out	Out:=ToLCase(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	변환 대상	입력	변환 대상 문자열	데이터 형식에 따름	–	''
Out	변환 결과	출력	변환 후의 문자열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out																				○

기능

● ToUCase

변환 대상 문자열 'In'의 반각 알파벳을 모두 대문자로 변환합니다.

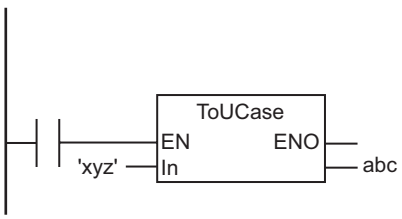
● ToLCase

변환 대상 문자열 'In'의 반각 알파벳을 모두 소문자로 변환합니다.

두 명령은 모두 문자열 끝의 NULL 문자를 포함하여 출력합니다. 또한 반각 알파벳 이외의 문자에는 영향이 없습니다.

ToUCase 명령에서 'In'='xyz'인 예가 아래에 나와 있습니다. 변수 abc의 값은 'XYZ'가 됩니다.

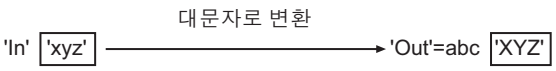
LD



ST

```
abc:=ToUCase('xyz');
```

'In'의 반각 알파벳을 모두 대문자로 변환합니다.

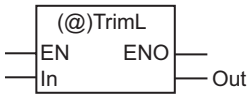


사용상의 주의 사항

- 전각 알파벳은 변환 대상에서 제외됩니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'에 문자 코드 이상이 있는 경우

TrimL/TrimR

TrimL : 문자열에서 앞의 공백을 삭제합니다.
TrimR : 문자열에서 끝의 공백을 삭제합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TrimL	문자열 · 좌측 트림	FUN		Out:=TrimL(In);
TrimR	문자열 · 우측 트림	FUN		Out:=TrimR(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	삭제 대상	입력	삭제 대상 문자열	데이터 형식에 따름	–	''
Out	삭제 결과	출력	삭제 후의 문자열	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Out																				○

기능

● TrimL

삭제 대상 문자열 'In'의 선두부터 처음으로 나타나는 문자까지의 공백을 삭제합니다. 앞에 공백이 없는 경우에는 아무 처리도 하지 않습니다.

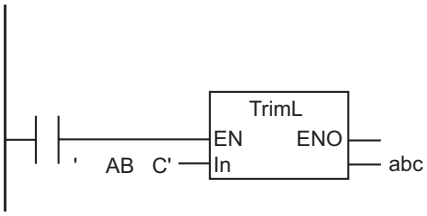
● TrimR

삭제 대상 문자열 'In'의 마지막 문자에서 끝까지의 공백을 삭제합니다. 끝에 공백이 없는 경우에는 아무 처리도 하지 않습니다.

두 명령은 모두 문자열 끝의 NULL 문자를 포함하여 출력합니다. 또한 ASCII 코드의 공백(16#20)과 일본어의 전각 공백(16#E38080)을 모두 공백으로 간주합니다.

TrimL 명령에서 'In='　AB　C'인 경우의 예가 아래에 나와 있습니다. 변수 abc의 값은 'AB　C'가 됩니다.

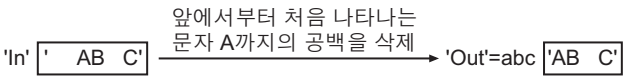
LD



ST

```
abc:=TrimL('　AB　C');
```

'In'의 앞에서부터 처음 나타나는 문자까지의 공백을 삭제합니다.



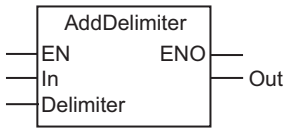
사용상의 주의 사항

다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'에 문자 코드 이상이 있는 경우

AddDelimiter

구조체 전체의 값을 구분 문자를 부가한 문자열로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
AddDelimiter	구분 문자 부가 문자열 쓰기	FUN		Out:=AddDelimiter(In, Delimiter);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	입력 구조체	입력	문자열 변환 대상이 되는 구조체 전체	멤버의 데이터 형식에 따름	–	*1
Delimiter	구분 문자		구분 문자	_COMMA _TAB _SEMICOLON _SPACE	–	_COMMA
Out	반환값	출력	구분 문자를 부가한 문자열	최대 1986바이트(반각 영숫자 1985문자+끝의 NULL 문자)	–	–

*1. 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	구조체 전체																			
Delimiter	열거 형식 _eDELIMITER 열거자는 기능 설명 참조																			
Out																				○

기능

입력 구조체 'In'의 멤버 값을 선두의 멤버부터 순서대로 문자열로 변환하여, 구분 문자 'Delimiter'로 구분하여 연결합니다. 연결된 문자열은 반환값 'Out'에 출력됩니다. 'Out'의 끝에는 NULL 문자가 부가됩니다.

'Delimiter'의 데이터 형식은 열거 형식 _eDELIMITER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_COMMA	','(쉼표)
_TAB	'$T'(탭)
_SEMICOLON	';'(세미콜론)
_SPACE	' '(공백 문자)

'In'의 멤버의 값은 데이터 형식에 따라 다음과 같이 변환됩니다.

● 부울 형식

FALSE는 '0', TRUE는 '1'로 변환됩니다.

● 비트 열 형식

16진수 영숫자로 표현한 것을 문자열로 변환합니다. 단, 16진수임을 표시하는 접두어 '16#'은 문자열에 포함되지 않습니다.
멤버의 값이 멤버의 데이터 형식의 최대 자릿수에 채워지지 않는 경우, 상위 자리를 '0'으로 채웁니다. 즉, 제로 패딩을 합니다.
데이터 형식에 따라 문자열의 문자 수가 다음과 같이 달라집니다.

멤버의 데이터 형식	문자 수
BYTE	반각 영숫자 2문자
WORD	반각 영숫자 4문자
DWORD	반각 영숫자 8문자
LWORD	반각 영숫자 16문자

몇 가지 예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
BYTE#16#AB	'AB'
LWORD#16#0123	'0000000000000123'

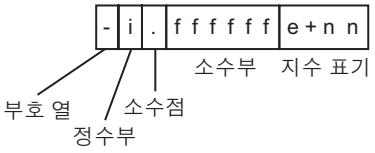
● 정수 형식

정수의 값을 문자열로 변환합니다. 상위 자리의 0은 문자열로 출력되지 않습니다. 멤버의 값이 음수인 경우에는 문자열의 앞에 음수 부호 '-'를 붙입니다.
몇 가지 예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
UINT#0012	'12'
LINT#-12	'-12'

● 실수 형식

멤버의 값을 변환한 문자열의 구성은 다음과 같습니다.



항목	내용
부호 열	멤버의 값이 음수인 경우에만 '-'(마이너스)가 부가됩니다. 멤버의 값이 양수인 경우, '+'(플러스)는 부가되지 않습니다.
정수부	항상 1자리로 표기됩니다.
소수점	멤버의 값이 소수가 아니라도 항상 표기됩니다.
소수부	멤버가 REAL 형식인 경우 6자리, LREAL 형식인 경우 14자리로 표기됩니다.
지수 표기	항상 표기됩니다. 'e'는 지수 표기의 e를 나타냅니다. nn의 자릿수는 2 또는 3입니다. nn의 부호는 멤버 값의 절대값이 1.0 이상인 경우 '+'(플러스), 1.0 미만인 경우 '-'(마이너스)입니다. 단, 멤버의 값이 0인 경우에는 '+'(플러스)입니다.

멤버의 값이 무한대, 비수인 경우 문자열은 다음과 같습니다.

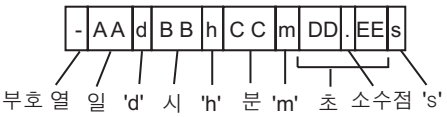
멤버의 값	문자열
+∞	'inf'
-∞	'-inf'
비수	'nan' 또는 '-nan'

몇 가지 예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
REAL#3.14e1	'3.140000e+01'
REAL#-123.4567	'-1.234567e+02'
REAL#0	'0.000000e+00'
LREAL#0.00123456789	'1.23456789000000e-03'
LREAL#1.0e308	'1.00000000000000e+308'

● 지속 시간 형식

멤버의 값을 변환한 문자열의 구성은 다음과 같습니다.



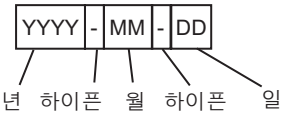
항목	내용
부호 열	멤버의 값이 음수인 경우에만 '-'(마이너스)가 부가됩니다. 멤버의 값이 양수인 경우, '+'(플러스)는 부가되지 않습니다.
일	항상 표기됩니다. 값의 범위는 0~106751입니다. 상위 자리에는 0을 채우지 않습니다.
시	항상 2자리로 표기됩니다. 값의 범위는 00~23입니다.
분	항상 2자리로 표기됩니다. 값의 범위는 00~59입니다.
초	항상 표기됩니다. DD의 값은 항상 2자리로 표기되고 범위는 00~59입니다. EE의 값은 항상 9자리로 표기되고 범위는 000000000~999999999입니다.
'd', 'h', 'm', 's', 소수점	항상 표기됩니다.

몇 가지 예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
T#-180122000ms	'-2d02h02m02.000000000s'
T#100d2h3m5.678s	'100d02h03m05.678000000s'
T#2h3m5.678s	'0d02h03m05.678000000s'

● 날짜 형식

멤버의 값을 변환한 문자열의 구성은 다음과 같습니다.

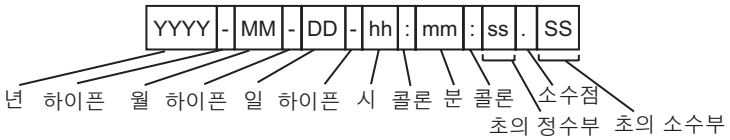


월, 일은 각각 2자리의 문자열로 변환됩니다.
예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
D#2010-1-2	'2010-01-02'

● 날짜 시각 형식

멤버의 값을 변환한 문자열의 구성은 다음과 같습니다.

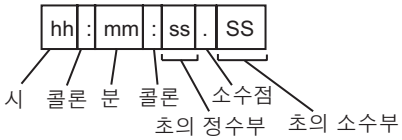


월 MM, 일 DD, 시 hh, 분 mm, 초의 정수부 ss는 각각 2자리의 문자열로 변환됩니다. 초의 소수부 SS는 9자리의 문자열로 변환됩니다.
예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
DT#1994-09-23-12:16:8.12	'1994-09-23-12:16:08.120000000'

● 시각 형식

멤버의 값을 변환한 문자열의 구성은 다음과 같습니다.



시 hh, 분 mm, 초의 정수부 ss는 각각 2자리의 문자열로 변환됩니다. 초의 소수부 SS는 9자리의 문자열로 변환됩니다.
예가 아래에 나와 있습니다.

멤버의 값	변환 후의 문자열
TOD#2:16:28.12	'02:16:28.120000000'

● 문자열 형식

그대로의 문자열이 됩니다. 단, 끝의 NULL 문자는 포함되지 않습니다.
예를 들어 멤버의 값이 끝에 NULL 문자를 포함하는 'ABC'인 경우에는 끝에 NULL 문자가 포함되지 않은 문자열 'ABC'로 변환됩니다.

● 구조체

멤버의 데이터 형식이 구조체 이외의 형식인 계층까지 멤버의 값을 앞에서부터 순서대로 변환합니다. 각 멤버의 데이터 형식의 변환 규칙에 따라 멤버의 값을 문자열로 변환합니다.
예를 들어, 구조체 A의 멤버의 데이터 형식에 구조체 B가 있는 경우는 다음과 같이 변환됩니다. 이 예에서는 구분 문자로 쉼표를 사용합니다.



● 열거 형식

열거 형식의 값을 DINT 형식의 값으로 변환합니다.
예를 들어, 열거 형식 Color의 취득 가능한 열거자는 red, yellow, green의 3가지로 합니다. 각각의 열거자에 대응하는 값의 경우 red는 1, yellow는 2, green은 3으로 합니다. 열거 형식 Color의 멤버 값이 yellow인 경우에는 문자열은 '2'가 됩니다.

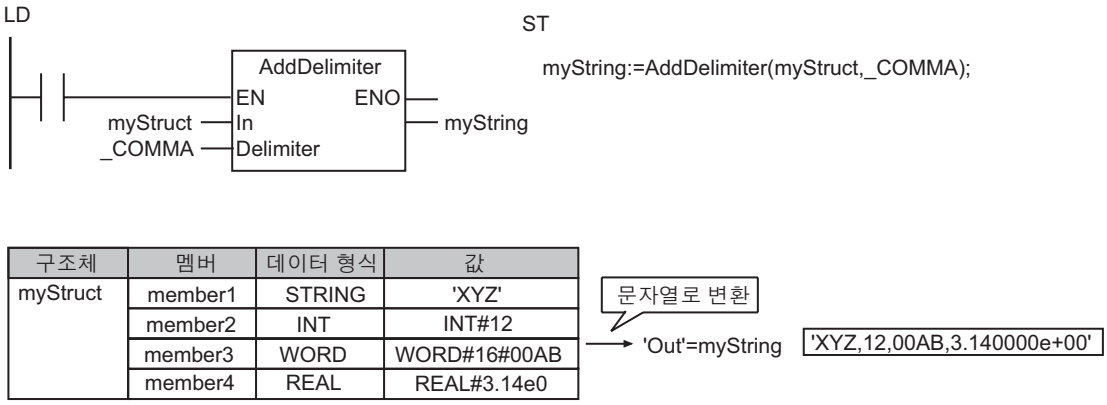
● **배열**

각 요소 사이를 구분 문자로 구분합니다. 각 요소의 값은 배열의 데이터 형식 변환 규칙에 따라 변환됩니다. 1차원 배열만 변환할 수 있습니다.

예를 들어, INT 형식의 배열 myArray[0..2]가 있다고 가정합니다. myArray[0]의 값이 INT#225, myArray[1]의 값이 INT#-128, myArray[2]의 값이 INT#0, 구분 문자가 쉼표인 경우 문자열은 '225, -128, 0'이 됩니다.

기술 예

구조체 'myStruct'를 문자열 'myString'으로 변환하는 경우의 기술 예가 아래에 나와 있습니다. 구분 문자는 ','(쉼표)를 사용합니다.



구조체	멤버	데이터 형식	값
myStruct	member1	STRING	'XYZ'
	member2	INT	INT#12
	member3	WORD	WORD#16#00AB
	member4	REAL	REAL#3.14e0

참고

- SD 메모리 카드 내의 지정된 CSV 파일에 값을 쓰는 경우 'FilePuts 명령(2-1084페이지)'과 이 명령을 함께 사용하면 편리합니다. 사용 방법은 샘플 프로그램을 참조하십시오.
- 이 명령으로 변환한 문자열은 'SubDelimiter 명령(2-558페이지)'에서 구조체의 멤버 값으로 읽을 수 있습니다.

사용상의 주의 사항

- 'In'의 멤버 값의 일부로는 구분 문자를 사용하지 마십시오. 구분 문자를 'In'의 멤버 값의 일부로 사용하면 SubDelimiter 명령에서 문자열을 구조체의 멤버 값으로 변환할 때 정확하게 변환할 수 없습니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 변환 결과의 문자열이 끝의 NULL 문자를 포함하여 1986바이트를 초과하는 경우
  - 'In'에 2차원 이상의 배열의 멤버가 있는 경우
  - 'In'에 공용체의 멤버가 있는 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

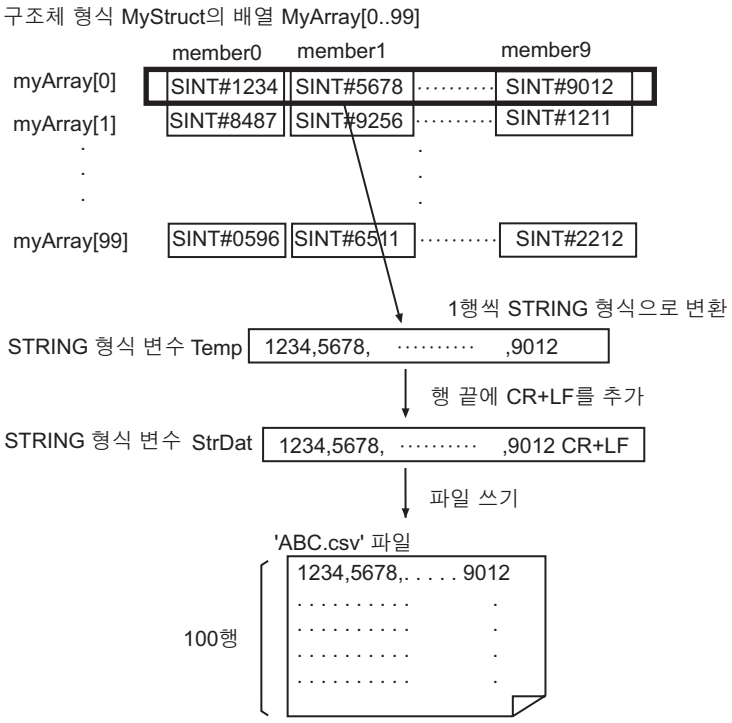
샘플 프로그램

10개의 SINT 형식의 변수를 멤버로 가진 구조체 myStruct가 있다고 가정합니다. 구조체 myStruct의 배열 myArray[0..99]의 내용을 100행의 CSV 파일 형식으로 SD 메모리 카드의 'ABC.csv' 파일에 저장합니다. 1행에는 배열 요소의 멤버 수치가 문자열로 10개 나열되며, 그 사이에는 쉼표를 삽입합니다. 행의 끝에는 행바꿈 코드 CR+LF를 부가합니다.

처리 순서는 다음과 같습니다.

1. **FileOpen 명령을 사용해서 파일 'ABC.csv'를 엽니다.**
2. **AddDelimiter 명령으로 myArray[]의 요소를 1행만 STRING 형식의 변수 Temp로 변환합니다.**
3. **CONCAT 명령으로 Temp와 CR+LF를 결합하여 STRING 형식의 변수 StrDat에 저장합니다.**
4. **StrDat를 파일에 씁니다.**
5. **위의 2~4를 100행 분량 반복합니다.**
6. **FileClose 명령을 사용해서 파일을 닫습니다.**

구조체	멤버	데이터 형식
myStruct	member0	SINT
	member1	SINT
	⋮	
	member9	SINT



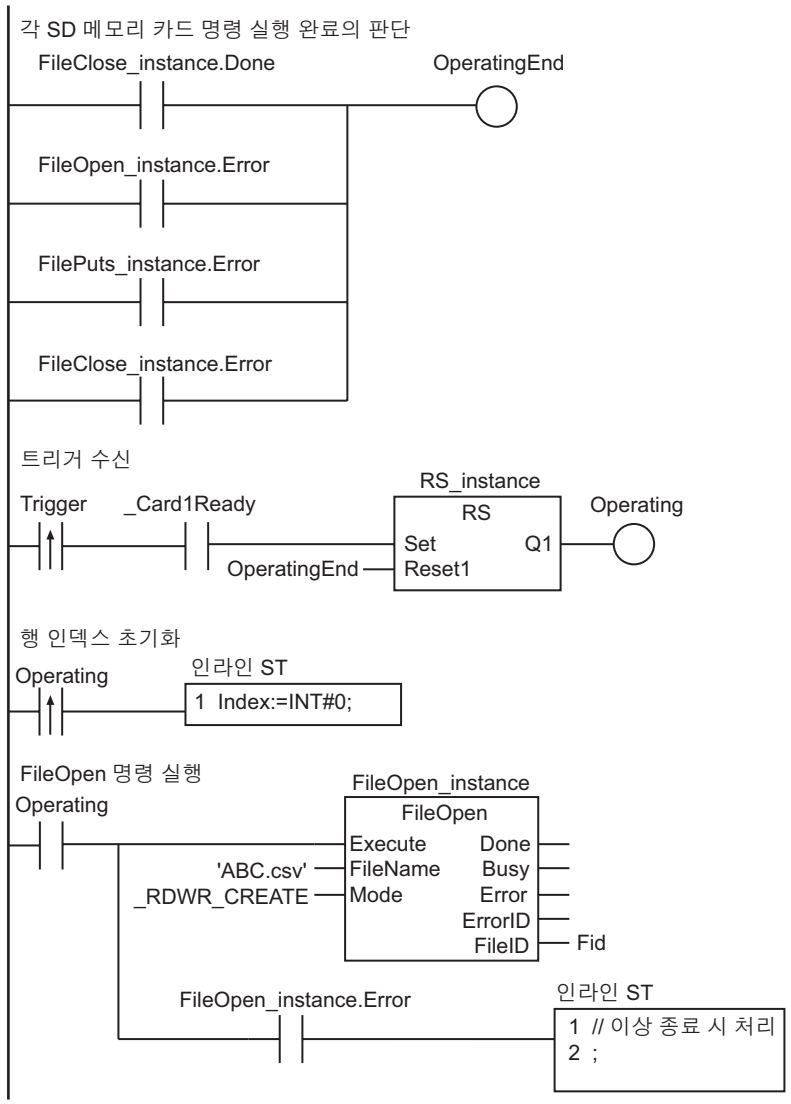
● 데이터 형식의 정의

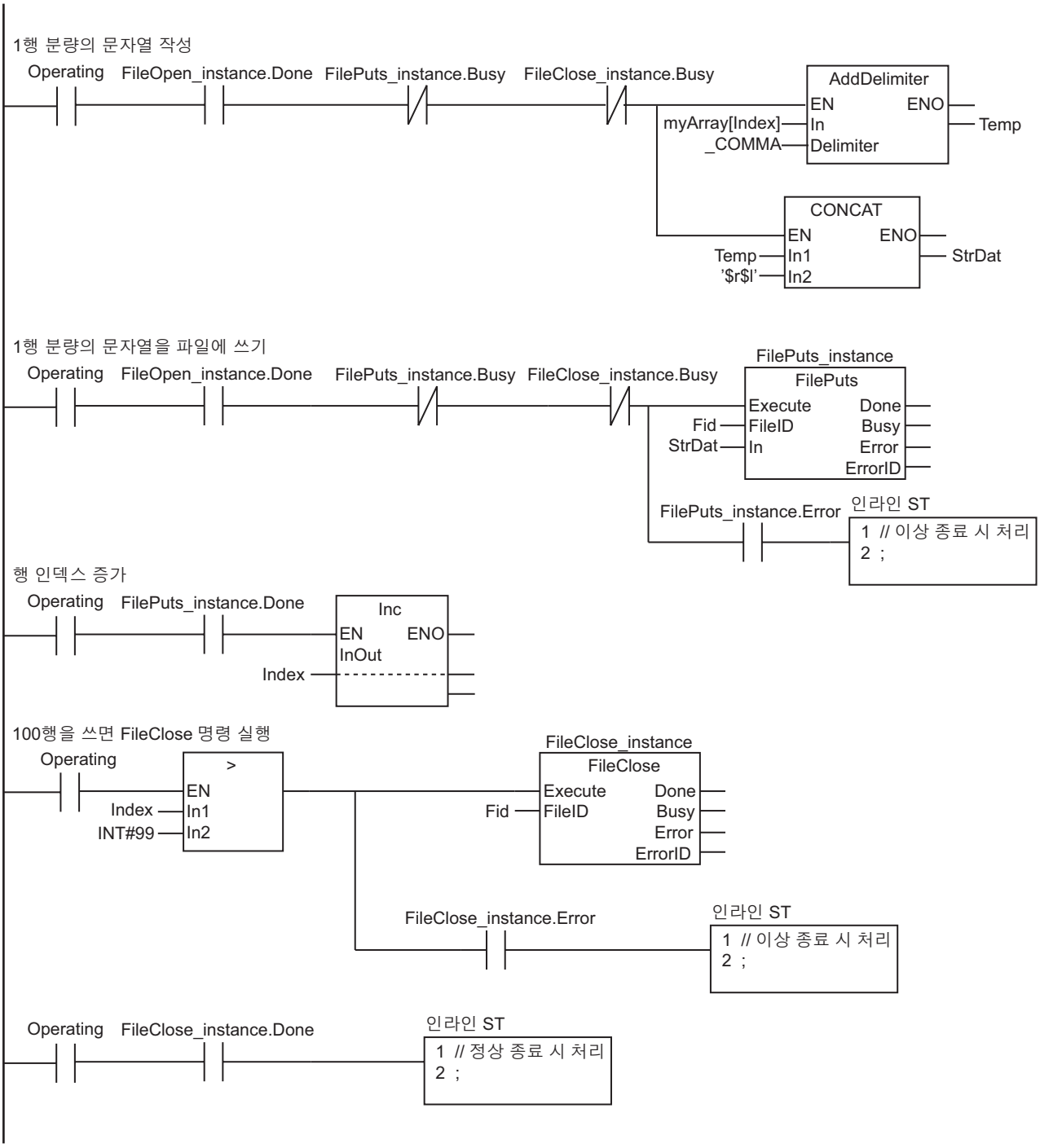
명칭	데이터 형식	코멘트
myStruct	STRUCT	구조체
member0	SINT	멤버
member1	SINT	멤버
member2	SINT	멤버
member3	SINT	멤버
member4	SINT	멤버
member5	SINT	멤버
member6	SINT	멤버
member7	SINT	멤버
member8	SINT	멤버
member9	SINT	멤버

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	False	처리 종료
	Trigger	BOOL	False	실행 조건
	Operating	BOOL	False	처리 중
	Index	INT	0	인덱스
	Fid	DWORD	16#0	파일 ID
	StrDat	STRING[256]	''	문자열 데이터
	myArray	ARRAY[0..99] OF myStruct	[100((member0:=0,member1:=0,member2:=0,member3:=0,member4:=0,member5:=0,member6:=0,member7:=0,member8:=0,member9:=0))]	수치 데이터
	Temp	STRING[256]	''	임시
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FilePuts_instance	FilePuts		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




1행 분량의 문자열 작성
Operating
FileOpen_instance.Done
FilePuts_instance.Busy
FileClose_instance.Busy
AddDelimiter
EN
ENO
myArray[Index]
In
Temp
_COMMA
Delimiter
CONCAT
EN
ENO
Temp
In1
StrDat
'rl'
In2
1행 분량의 문자열을 파일에 쓰기
Operating
FileOpen_instance.Done
FilePuts_instance.Busy
FileClose_instance.Busy
FilePuts_instance
FilePuts
Execute
Done
Fid
FileID
Busy
StrDat
In
Error
ErrorID
FilePuts_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
행 인덱스 증가
Operating
FilePuts_instance.Done
Inc
EN
ENO
InOut
Index
100행을 쓰면 FileClose 명령 실행
Operating
>
EN
Index
In1
INT#99
In2
FileClose_instance
FileClose
Execute
Done
Fid
FileID
Busy
Error
ErrorID
FileClose_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
Operating
FileClose_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	False	실행 조건
	LastTrigger	BOOL	False	이전 태스크 주기의 Trigger 값
	OperatingStart	BOOL	False	처리 시작
	Operating	BOOL	False	처리 중
	Stage	INT	0	상태 이동
	Index	INT	0	인덱스
	Fid	DWORD	16#0	파일 ID
	StrDat	STRING[256]	''	문자열 데이터
	myArray	ARRAY[0..99] OF myStruct	[100((member0:=0,member1:=0,member2:=0,member3:=0,member4:=0,member5:=0,member6:=0,member7:=0,member8:=0,member9:=0))]	수치 데이터
	Temp	STRING[256]	''	임시
	FileOpen_instance	FileOpen		
	FilePuts_instance	FilePuts		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동에서 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
   OperatingStart:=TRUE;
   Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
   FileOpen_instance(Execute:=FALSE);
   FilePuts_instance(Execute:=FALSE);
   FileClose_instance(Execute:=FALSE);
   Stage          :=INT#1;
   Index          :=INT#0;      // 행 인덱스 초기화
   OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
   CASE Stage OF
   1 :                          // 파일 열기
      FileOpen_instance(
         Execute  :=TRUE,
         FileName:='ABC.csv',              // 파일명
         Mode     :=_RDWR_CREATE,    // 파일 읽기
         FileID  =>Fid);                   // 파일 ID

      IF (FileOpen_instance.Done=TRUE) THEN
         Stage:=INT#2;                  // 정상 종료
      END_IF;

      IF (FileOpen_instance.Error=TRUE) THEN
```

```
      Stage:=INT#99;              // 이상 종료
    END_IF;

  2 :                             // 1 행 분량의 문자열 작성
    StrDat:='';

    Temp :=AddDelimiter(myArray[Index],_COMMA);
    StrDat:=CONCAT(In1:=Temp, In2:='$r$l');

    Stage:=INT#3;

  3 :                             // 문자열 쓰기
    FilePuts_instance(
      Execute:=TRUE,
      FileID   :=Fid,
      In       :=StrDat);

    IF (FilePuts_instance.Done=TRUE) THEN
      Index:=Index+INT#1;

      IF (Index>INT#99) THEN  // 100 행을 씀
        Stage:=INT#4;
      ELSE
        FilePuts_instance(Execute:=FALSE);
        Stage:=INT#2;
      END_IF;
    END_IF;

    IF (FilePuts_instance.Error=TRUE) THEN
      Stage:=INT#99;       // 이상 종료
    END_IF;

  4 :                             // 파일 닫기
    FileClose_instance(
      Execute:=TRUE,
      FileID   :=Fid);            // 파일 ID

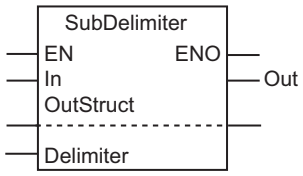
    IF (FileClose_instance.Done=TRUE) THEN
      Operating:=FALSE;    // 정상 종료
    END_IF;

    IF (FileClose_instance.Error=TRUE) THEN
      Stage:=INT#99;       // 이상 종료
    END_IF;

  99 :                            // 이상 종료 처리
    Operating:=FALSE;
  END_CASE;
END_IF;
```

SubDelimiter

문자열에서 구분 문자로 구분된 데이터를 읽어서 구조체의 멤버 값으로 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SubDelimiter	문자열 읽기 구분 문자 삭제	FUN		Out:=SubDelimiter(In, OutStruct, Delimiter);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	입력 문자열	입력	구조체의 멤버 값으로 변환할 데이터를 구분 문자로 구분한 문자열	최대 1986바이트(반각 영숫자 1985문자+끝의 NULL 문자)	–	''
Delimiter	구분 문자		구분 문자	_COMMA, _TAB, _SEMICOLON, _SPACE	–	_COMMA
OutStruct	저장대상 구조체	입출력	변환된 데이터를 저장할 구조체 전체	최대 8192바이트	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				○
Delimiter	열거 형식 _eDELIMITER 열거자는 기능 설명 참조																			
OutStruct	구조체 전체																			
Out	○																			

기능

입력 문자열 'In'의 구분 문자 'Delimiter'로 구분한 문자열 데이터를 저장대상 구조체 'OutStruct'의 멤버 값으로 변환하여, 각 멤버 값으로 앞에서부터 순서대로 저장합니다.

'Delimiter'의 데이터 형식은 열거 형식 _eDELIMITER입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_COMMA	','(쉼표)
_TAB	'$T'(탭)
_SEMICOLON	';'(세미콜론)
_SPACE	' '(공백 문자)

'In'의 구분 문자로 구분된 데이터 수가 'OutStruct'의 멤버 수보다 많은 경우 남은 데이터를 무시합니다.
'In'의 구분 문자로 구분된 데이터 수가 'OutStruct'의 멤버 수보다 적은 경우는 데이터가 부족한 분량의 멤버의 값을 변경하지 않습니다.
'OutStruct'의 멤버가 구조체이고 'In'의 데이터가 해당 구조체의 멤버 수보다 적은 경우에도 도중까지의 데이터를 저장합니다.
'OutStruct'의 멤버가 배열이고 'In'의 데이터가 해당 배열의 요소 수보다 적은 경우에도 도중까지의 데이터를 저장합니다.

'In'의 구분 문자로 구분된 데이터는 문자열 형식입니다. 'OutStruct'의 멤버의 데이터 형식에 따라 문자열 형식의 데이터를 다음과 같이 변환합니다.

● 부울 형식

문자열 형식의 데이터가 'FALSE' 또는 '0'인 경우, FALSE로 변환됩니다. 'TRUE' 또는 '1'인 경우, TRUE로 변환됩니다.
단, 다음과 같은 예외가 있습니다.

- '0' 또는 '1'의 앞에 '0'이 연속되어 있는 경우는 이를 무시합니다.
- 'FALSE', 'TRUE'의 대문자와 소문자는 구분하지 않습니다.

문자열 형식의 데이터가 'FALSE', 'TRUE', '0', '1' 이외의 값인 경우는 변환할 수 없습니다.

● 비트 열 형식

변환 규칙은 'STRING_TO_**(문자열→비트 열 변환 그룹) 명령(2-288페이지)'과 동일합니다.
16진수 수치를 나타내는 데이터가 아닌 경우에는 변환할 수 없습니다.

● 정수 형식

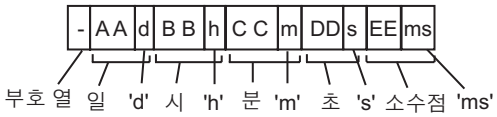
변환 규칙은 'STRING_TO_**(문자열→정수 변환 그룹) 명령(2-286페이지)'과 동일합니다.
정수 형식의 수치를 나타내는 데이터가 아닌 경우에는 변환할 수 없습니다.

● 실수 형식

변환 규칙은 'STRING_TO_**(문자열→실수 변환 그룹) 명령(2-290페이지)'과 동일합니다.
실수 형식의 수치를 나타내는 데이터가 아닌 경우에는 변환할 수 없습니다.

● 지속 시간 형식

다음 구조의 데이터를 지속 시간 형식의 값으로 변환합니다.



항목	내용
부호 열	'+'(플러스) 또는 부호 열이 없는 경우는 멤버의 값이 양수가 됩니다. '-'(마이너스)인 경우는 멤버의 값이 음수가 됩니다.
일	AA의 값의 소수점 이하 12번째 자리 다음은 버립니다.
시	BB의 값의 소수점 이하 12번째 자리 다음은 버립니다.
분	CC의 값의 소수점 이하 11번째 자리 다음은 버립니다.
초	DD의 값의 소수점 이하 10번째 자리 다음은 버립니다.
밀리초	EE의 값의 소수점 이하 7번째 자리 다음은 버립니다.

(주의) 1 부호 열, 일, 시, 분, 초, 밀리초의 값 앞에 ' '(공백 문자)가 있는 경우는 ' '(공백 문자)를 무시합니다.
2 AA, BB, CC, DD, EE의 값에서 문자와 문자 사이에 단독 '_'(밑줄)이 포함되는 경우, '_'(밑줄)을 무시합니다.
3 일, 시, 분, 초, 밀리초의 값이 '.'(마침표)를 사용한 실수인 경우에도 변환할 수 있습니다.
4 일, 시, 분, 초, 밀리초 중 하나가 데이터에 포함되어 있으면 다른 것들이 생략되어 있어도 변환할 수 있습니다.
5 일, 시, 분, 초, 밀리초의 값 앞에 '0'이 부가되어 있어도 변환할 수 있습니다.

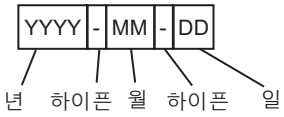
단, 다음과 같은 경우에는 변환할 수 없습니다.

- 위의 구조가 아닌 경우
- 부호 열과 일 사이에 '_'(밑줄)이 있는 경우
- '.'(마침표)와 '_'(밑줄)이 연속되는 경우

예를 들어, 문자열 형식의 데이터가 '-0.5d48h0.123456789ms'인 경우 멤버의 값은 T#-2d12h0m0s0.123456ms(T#-216000000.123456ms)가 됩니다.

● 날짜 형식

다음 구조의 데이터를 날짜 형식의 값으로 변환합니다.



단, 다음과 같은 예외가 있습니다.

- 년, 월, 일의 값 앞에 ' '(공백 문자)가 있는 경우 ' '(공백 문자)를 무시합니다.
- 년, 월, 일의 값에서 문자와 문자 사이에 단독 '_'(밑줄)이 포함되는 경우 '_'(밑줄)을 무시합니다.
- 년, 월, 일의 값 앞에 '0'이 부가되어 있어도 변환할 수 있습니다.

다음과 같은 경우에는 변환할 수 없습니다.

- 위의 구조가 아닌 경우
- 존재하지 않는 날짜인 경우

예를 들어, 문자열 형식의 데이터가 '1970-1-01'인 경우, 멤버의 값은 D#1970-01-01이 됩니다.

● 날짜 시각 형식

다음 구조의 데이터를 날짜 시각 형식의 값으로 변환합니다.



항목	내용
년, 월, 일	날짜를 나타내는 년월일입니다.
시	값의 범위는 0~23입니다.
분	값의 범위는 0~59입니다.
초	값의 범위는 0~59.999999999입니다. 값이 정수인 경우 소수점은 필요 없습니다.
하이픈, 콜론	항상 필요합니다.

(주의) 1 년, 월, 일, 시, 분, 초의 값 앞에 ' '(공백 문자)가 있는 경우 ' '(공백 문자)를 무시합니다.
2 년, 월, 일, 시, 분, 초의 값에서 문자와 문자 사이에 단독 '_'(밑줄)이 포함되는 경우 '_'(밑줄)을 무시합니다.
3 년, 월, 일, 시, 분, 초의 값 앞에 '0'이 부가되어 있어도 변환할 수 있습니다.

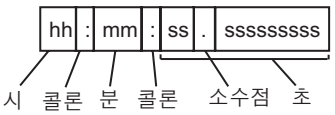
단, 다음과 같은 경우에는 변환할 수 없습니다.

- 위의 구조가 아닌 경우
- 존재하지 않는 날짜인 경우

예를 들어, 문자열 형식의 데이터가 '1970-01-23-4:56:07.89'인 경우, 멤버의 값은 DT#1970-01-23-04:56:07.89가 됩니다.

● 시각 형식

다음 구조의 데이터를 시각 형식의 값으로 변환합니다.



항목	내용
시	값의 범위는 0~23입니다.
분	값의 범위는 0~59입니다.
초	값의 범위는 0~59.999999999입니다. 값이 정수인 경우 소수점인 '.'(마침표)는 필요 없습니다.
콜론	항상 필요합니다.

(주의) 1 시, 분, 초의 값 앞에 ' '(공백 문자)가 있는 경우 ' '(공백 문자)를 무시합니다.
2 시, 분, 초의 값에서 문자와 문자 사이에 단독 '_'(밑줄)이 포함되는 경우 '_'(밑줄)을 무시합니다.
3 시, 분, 초의 값 앞에 '0'이 부가되어 있어도 변환할 수 있습니다.

단, 다음과 같은 경우에는 변환할 수 없습니다.

- 위의 구조가 아닌 경우
- '.'(마침표)와 '_'(밑줄)이 연속되는 경우

예를 들어, 문자열 형식의 데이터가 '12:23:34.567'인 경우, 멤버의 값은 TOD#12:23:34.567이 됩니다.

● 문자열 형식

데이터의 끝에 NULL 문자를 부가한 값이 멤버의 값이 됩니다. 단, 문자열이 멤버의 데이터 크기를 초과한 경우는 변환할 수 없습니다.
예를 들어, 문자열 형식의 데이터가 끝에 NULL 문자가 없는 'ABC'인 경우, 멤버의 값은 끝에 NULL 문자를 부가한 'ABC'가 됩니다.

● 구조체

문자열 형식의 데이터를 멤버의 데이터 형식 변환 규칙에 따라 변환합니다. 멤버가 구조체 이외의 데이터 형식인 계층까지 앞에서부터 순서대로 변환한 값을 멤버의 값으로 저장합니다. 예를 들어, 구조체 A의 멤버에 구조체 B가 있는 경우는 다음과 같이 변환됩니다.



● 열거 형식

DINT 형식의 변수를 나타내는 문자열 형식의 데이터를 열거 형식의 열거자로 변환합니다. 정수 형식과 동일한 규칙을 사용하여 DINT 형식으로 변환하고 DINT 형식의 값을 열거 형식의 값으로 하여, 열거 형식의 값에 대응하는 열거자로 변환합니다.
단, 문자열 형식의 데이터가 DINT 형식의 수치를 나타내는 문자열이 아닌 경우에는 변환할 수 없습니다.
예를 들어, 열거 형식 Color의 취득 가능한 열거자는 red, yellow, green의 3가지로 합니다. 각각의 열거자에 대응하는 값의 경우 red는 1, yellow는 2, green은 3으로 합니다. 데이터가 '3'인 경우, 멤버의 값은 green이 됩니다.

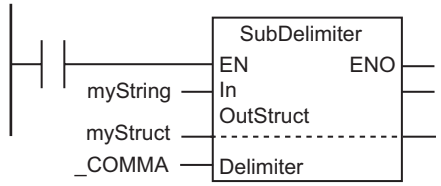
● 배열

구분 문자로 구분된 데이터를 각 요소의 값으로 변환합니다. 변환 규칙은 배열의 데이터 형식에 따릅니다. 멤버가 1차원 배열인 경우만 변환할 수 있습니다.
예를 들어, 멤버에 BYTE 형식의 배열 myString[0..3]이 있다고 가정합니다. 문자열 'AA,BB,CC,DD'의 쉼표로 구분된 데이터를 각 요소의 값으로 변환하는 경우, myString[0]은 BYTE#16#AA, mySring[1]은 BYTE#16#BB, myString[2]는 BYTE#16#CC, myString[3]은 BYTE#16#DD가 됩니다.

기술 예

문자열 'myString'의 쉼표로 구분된 데이터를 읽어서 구조체 'myStruct'의 멤버의 값으로 저장하는 경우의 기술 예가 아래에 나와 있습니다.

LD



ST

```
SubDelimiter(myString,myStruct,_COMMA);
```

myString 'XYZ,1234,aB,3.14'

구조체	멤버	데이터 형식	값
myStruct	member1	STRING	' '
	member2	INT	INT#0
	member3	WORD	WORD#16#0000
	member4	REAL	REAL#0

MyString의 데이터를 MyStruct의 멤버의 값으로 저장

구조체	멤버	데이터 형식	값
myStruct	member1	STRING	'XYZ'
	member2	INT	INT#1234
	member3	WORD	WORD#16#00AB
	member4	REAL	REAL#3.14e+0

참고

- SD 메모리 카드 내의 지정된 CSV 파일에서 값을 읽는 경우 'FileGets 명령(2-1077페이지)'과 이 명령을 함께 사용하면 편리합니다. 사용 방법은 샘플 프로그램을 참조하십시오.
- 'AddDelimiter 명령(2-547페이지)'으로 변환한 문자열을 구조체의 데이터로 반환하는 경우에 이 명령을 사용합니다.

사용상의 주의 사항

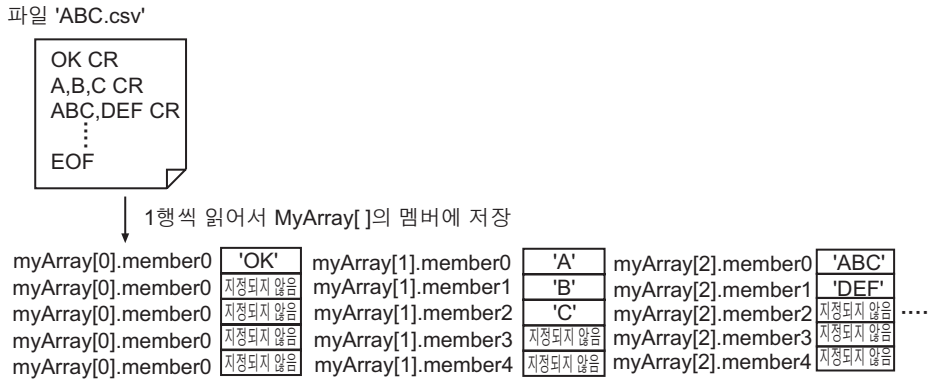
- 'In'에 구분 문자가 연속되는 경우, 구분 문자로 구분되는 데이터가 존재하지 않습니다. 데이터가 존재하지 않는 경우, 'OutStruct'의 멤버의 값은 잘못된 값이 됩니다.
- 구분 문자를 'In'에서 구분 문자 이외의 용도로 사용하지 마십시오. 구분 문자 이외의 용도로 사용한 경우에도 이 명령은 구분 문자로 인식하여 동작합니다.
- 'OutStruct'에 문자열 형식의 멤버가 있는 경우, 'In' 안의 대응하는 데이터에는 끝에 NULL 문자를 부가하지 마십시오. 'In'의 끝 이외에서 NULL 문자를 사용하면 NULL 문자까지의 문자열만 변환할 수 있습니다.
- 'OutStruct'에 열거 형식의 멤버가 있는 경우, 'In'의 해당 부분은 열거자로 정의한 값을 나타내는 데이터로 하십시오. 열거 형식 변수의 값이 열거자로 정의된 값이 아니어도 이상 상태가 되지는 않습니다.
- 다음과 같은 경우, 이상이 발생합니다. ENO는 FALSE가 되고 'OutStruct'의 값은 잘못된 값이 됩니다.
  - 'OutStruct'의 멤버의 데이터 형식의 값으로 변환할 수 없는 경우
  - 변환 결과가 'OutStruct'의 멤버의 데이터 형식의 값의 유효 범위 밖인 경우
  - 'OutStruct'에 2차원 이상의 배열의 멤버가 있는 경우
  - 'OutStruct'에 공용체의 멤버가 있는 경우
  - 'OutStruct'의 크기가 8192바이트 이상인 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

'ABC.csv' 파일에는 행바꿈 코드 CR로 행바꿈된 문자열이 여러 행 저장되어 있습니다. 1행의 문자열은 여러 데이터를 쉼표로 구분하여 나타내고 있습니다.
이 파일에서 문자열을 1행씩 읽어서 쉼표로 구분된 데이터를 구조체 myStruct의 배열 변수 myArray[]의 멤버 값으로 앞에서부터 순서대로 저장합니다. myStruct 는 5개의 STRING 형식의 변수를 멤버로 가진 구조체로 합니다.
파일을 끝까지 읽으면(파일 끝 EOF까지 읽으면) 처리를 종료합니다.



처리 순서는 다음과 같습니다.

1 **FileOpen 명령을 사용해서 파일 'ABC.csv'를 엽니다.**

2 **FileGets 명령을 사용하여 파일의 내용을 1행 읽습니다.**

3 **SubDelimiter 명령을 사용하여 쉼표로 구분된 문자열을 myArray[ ]의 멤버의 값으로 저장합니다.**

4 **위의 2~3을 파일 끝 EOF를 읽을 때까지 계속합니다.**

5 **FileClose 명령을 사용해서 파일을 닫습니다.**

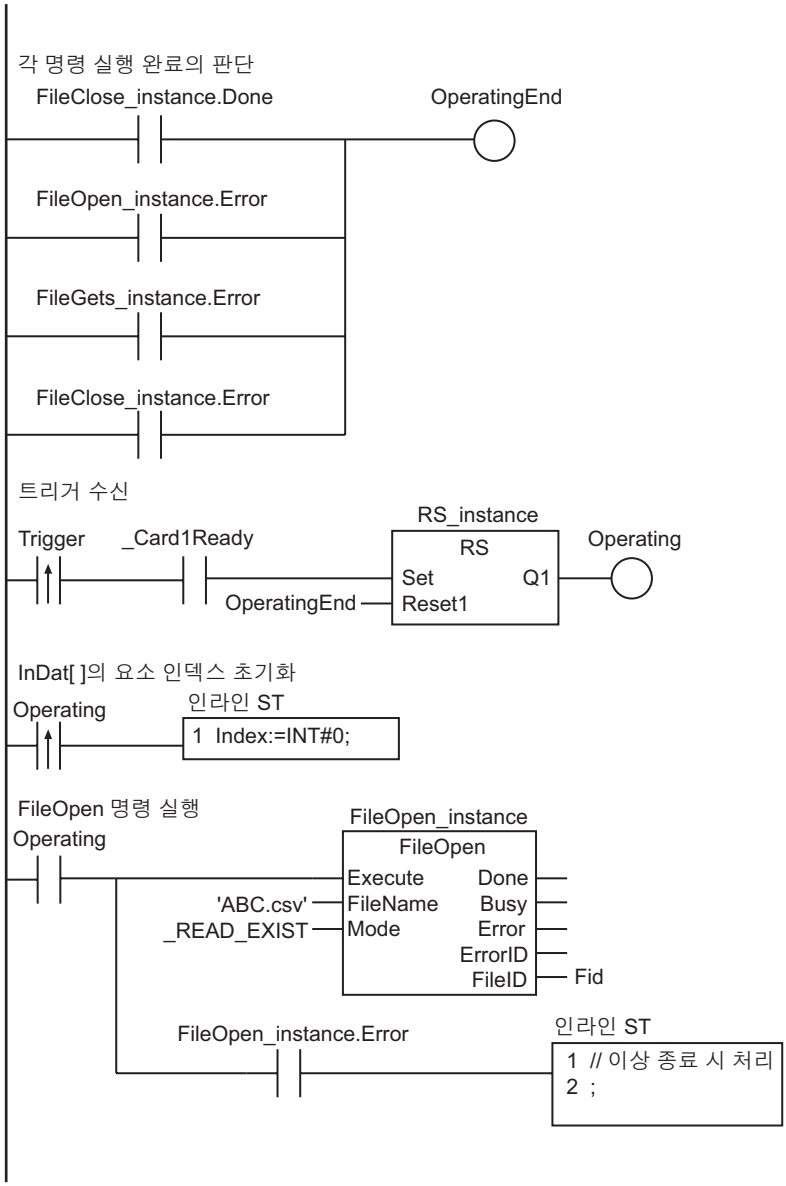
● 데이터 형식의 정의

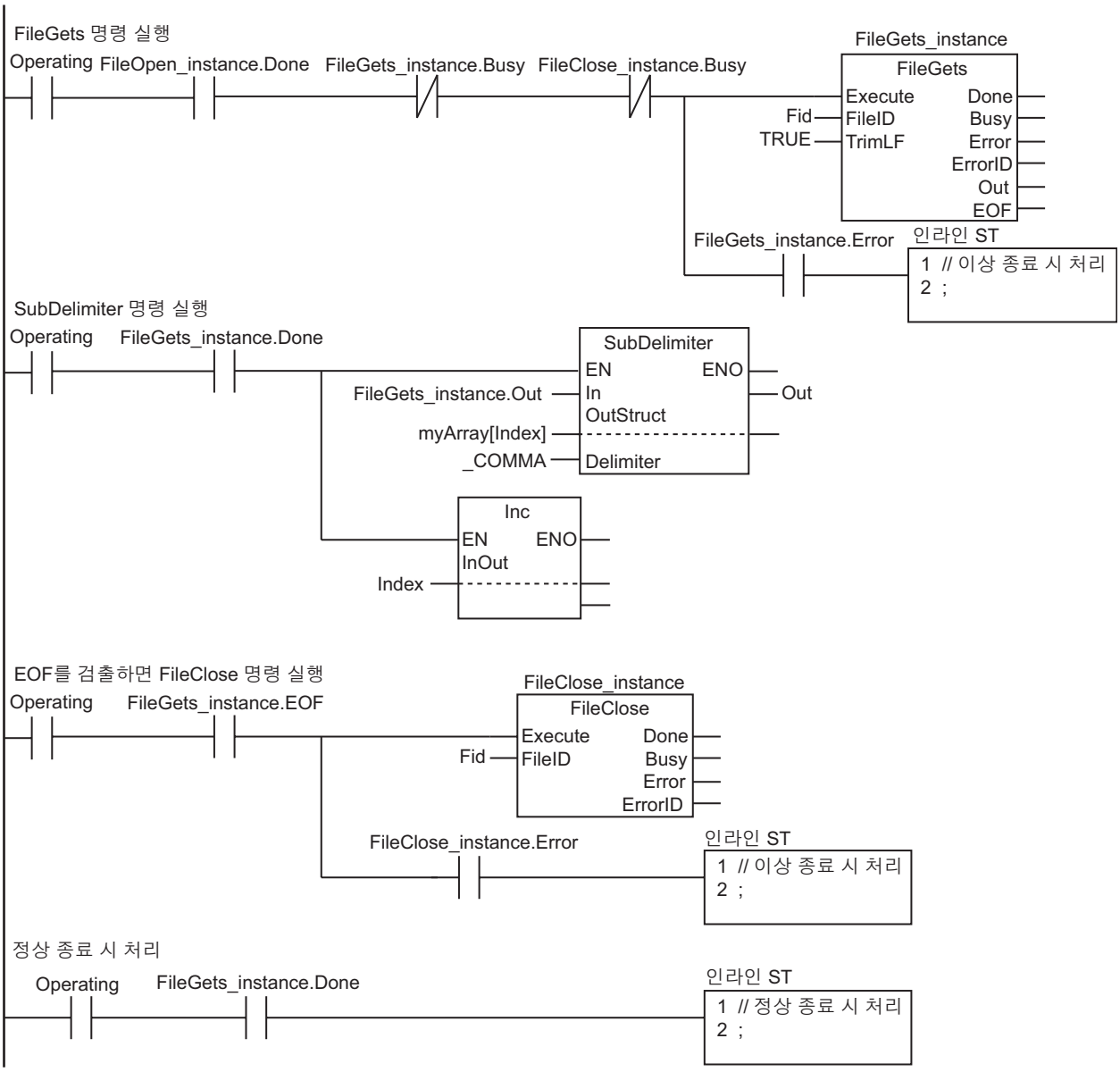
명칭	데이터 형식	코멘트
myStruct	STRUCT	구조체
member0	STRING	멤버
member1	STRING	멤버
member2	STRING	멤버
member3	STRING	멤버
member4	STRING	멤버

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	False	처리 종료
	Trigger	BOOL	False	실행 조건
	Operating	BOOL	False	처리 중
	Index	INT	0	myArray[ ]의 요소 인덱스
	Fid	DWORD	16#0	파일 ID
	myArray	ARRAY[0..999] OF myStruct	[1000((member0:='',member1:='',member2:='',member3:='',member4:=''))]	정수 데이터
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FileGets_instance	FileGets		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




FileGets 명령 실행
Operating
FileOpen_instance.Done
FileGets_instance.Busy
FileClose_instance.Busy
FileGets_instance
FileGets
Execute
Done
Fid
FileID
Busy
TRUE
TrimLF
Error
ErrorID
Out
EOF
FileGets_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
SubDelimiter 명령 실행
Operating
FileGets_instance.Done
SubDelimiter
EN
ENO
FileGets_instance.Out
In
Out
OutStruct
myArray[Index]
_COMMA
Delimiter
Inc
EN
ENO
InOut
Index
EOF를 검출하면 FileClose 명령 실행
Operating
FileGets_instance.EOF
FileClose_instance
FileClose
Execute
Done
Fid
FileID
Busy
Error
ErrorID
FileClose_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
정상 종료 시 처리
Operating
FileGets_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	False	실행 조건
	LastTrigger	BOOL	False	이전 태스크 주기의 Trigger 값
	OperatingStart	BOOL	False	처리 시작
	Operating	BOOL	False	처리 중
	myArray	ARRAY[0..999] OF myStruct	[1000((member0:='',member1:='',member2:='',member3:='',member4:=''))]	정수 데이터
	Stage	INT	0	상태 이동
	Index	INT	0	myArray[ ]의 요소 인덱스
	Fid	DWORD	16#0	파일 ID
	FileOpen_instance	FileOpen		
	FileGets_instance	FileGets		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동에서 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
   OperatingStart :=TRUE;
   Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
   FileOpen_instance(Execute:=FALSE);
   FileGets_instance(Execute:=FALSE);
   FileClose_instance(Execute:=FALSE);
   Stage          :=INT#1;
   Index          :=INT#0;
   OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
   CASE Stage OF
   1 :                        // 파일 열기
      FileOpen_instance(
         Execute  :=TRUE,
         FileName :='ABC.csv',        // 파일명
         Mode     :=_READ_EXIST,  // 파일 읽기
         FileID  =>Fid);              // 파일 ID

      IF (FileOpen_instance.Done=TRUE) THEN
         Stage:=INT#2;          // 정상 종료
      END_IF;

      IF (FileOpen_instance.Error=TRUE) THEN
         Stage:=INT#99;       // 이상 종료
      END_IF;
```

```
  2 :                       // 문자열 읽기
      FileGets_instance(
        Execute:=TRUE,
        FileID   :=Fid,
        TrimLF  :=TRUE);

      IF (FileGets_instance.Done=TRUE) THEN
        // 읽은 문자열을 myArray[ ] 의 멤버의 값으로 저장
        SubDelimiter(FileGets_instance.Out,myArray[Index],_COMMA);
        Index:=Index+INT#1;

        // 파일 끝에 도달
        IF (FileGets_instance.EOF=TRUE) THEN
          Stage:=INT#3;     // 정상 종료
        ELSE
          FileGets_instance(Execute:=FALSE);
        END_IF;
      END_IF;

      IF (FileGets_instance.Error=TRUE) THEN
        Stage:=INT#99;      // 이상 종료
      END_IF;

  3 :                       // 파일 닫기
      FileClose_instance(
        Execute:=TRUE,
        FileID :=Fid);              // 파일 ID

      IF (FileClose_instance.Done=TRUE) THEN
        Operating:=FALSE;  // 정상 종료
      END_IF;

      IF (FileClose_instance.Error=TRUE) THEN
        Stage:=INT#99;      // 이상 종료
      END_IF;

   99 :                     // 이상 종료 처리
       Operating:=FALSE;
  END_CASE;
END_IF;
```

시간/시각 명령

명령	명칭	페이지	명령	명칭	페이지
ADD_TIME	시간 가산	2-570	TodToSec	시각→초 변환	2-603
ADD_TOD_TIME	시각과 시간의 가산	2-572	SecToDt	초→날짜 시각 변환	2-604
ADD_DT_TIME	날짜 시각과 시간의 가산	2-574	SecToDate	초→날짜 변환	2-606
SUB_TIME	시간 감산	2-576	SecToTod	초→시각 변환	2-608
SUB_TOD_TIME	시각과 시간의 감산	2-578	TimeToNanoSec	시간→나노초 변환	2-610
SUB_TOD_TOD	시각 감산	2-580	TimeToSec	시간→초 변환	2-611
SUB_DATE_DATE	날짜 감산	2-581	NanoSecToTime	나노초→시간 변환	2-612
SUB_DT_DT	날짜 시각 감산	2-582	SecToTime	초→시간 변환	2-613
SUB_DT_TIME	날짜 시각과 시간의 감산	2-584	ChkLeapYear	윤년 판별	2-615
MULTIME	시간 승산	2-586	GetDaysOfMonth	달의 일수 취득	2-616
DIVTIME	시간 제산	2-588	DaysToMonth	일수→월 변환	2-618
CONCAT_DATE_TOD	날짜와 시각의 결합	2-590	GetDayOfWeek	요일 취득	2-620
DT_TO_TOD	날짜 시각으로부터 시각 잘라내기	2-592	GetWeekOfYear	주 취득	2-622
DT_TO_DATE	날짜 시각으로부터 날짜 잘라내기	2-594	DtToDateStruct	시각 분해	2-624
SetTime	시계 보정	2-596	DateStructToDt	시각 결합	2-626
GetTime	시각 취득	2-598	TruncTime	시간 버림	2-628
DtToSec	날짜 시각→초 변환	2-600	TruncDt	날짜 시각 버림	2-632
DateToSec	날짜→초 변환	2-602	TruncTod	시각 버림	2-636

ADD_TIME

시간과 시간을 가산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ADD_TIME	시간 가산	FUN	(@)ADD_TIME EN ENO In1 Out In2	Out:=ADD_TIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피가산 시간	입력	피가산 시간	데이터 형식에 따름	ns	T#0s
In2	가산 시간		가산 시간			
Out	가산 결과 시간	출력	가산 결과 시간	데이터 형식에 따름	ns	–

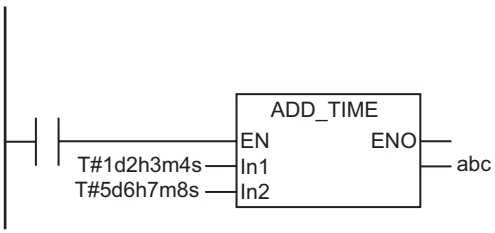
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																○				
In2																○				
Out																○				

기능

시간 'In1'과 'In2'를 가산합니다. 가산 결과 'Out'도 시간입니다.

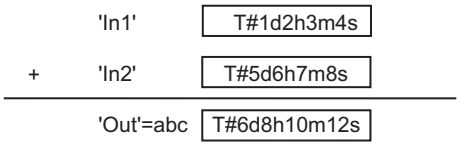
'In1'=T#1d2h3m4s, 'In2'=T#5d6h7m8s인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=ADD_TIME(T#1d2h3m4s, T#5d6h7m8s);
```



사용상의 주의 사항

가산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- T#106751d_23h_47m_16s_854.775807ms + T#0.000001ms
  → T#-106751d_23h_47m_16s_854.775808ms
- T#-106751d_23h_47m_16s_854.775808ms + T#-0.000001ms
  → T#106751d_23h_47m_16s_854.775807ms

ADD_TOD_TIME

시각에 시간을 가산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ADD_TOD_ TIME	시각과 시간의 가산	FUN	(@)ADD_TOD_TIME EN ENO In1 Out In2	Out:=ADD_TOD_TIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피가산 시각	입력	피가산 시각	데이터 형식에 따름	시분초	TOD# 0:0:0
In2	가산 시간		가산 시간		ns	T#0s
Out	가산 결과 시각	출력	가산 결과 시각	데이터 형식에 따름	시분초	–

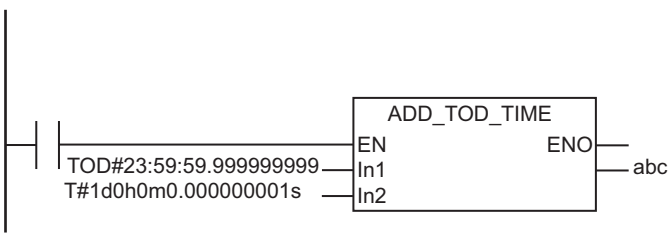
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																		○		
In2																○				
Out																		○		

기능

시각 'In1'에 시간 'In2'를 가산합니다. 가산 결과 'Out'은 시각입니다.

'In1'=TOD#23:59:59.999999999, 'In2'=T#1d0h0m0.000000001s인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=ADD_TOD_TIME(TOD#23:59:59.999999999,
                  T#1d0h0m0.000000001s);
```

	'In1'	TOD#23:59:59.999999999
+	'In2'	T#1d0h0m0.000000001s
	'Out'=abc	TOD#0:0:0.000000000

사용상의 주의 사항

가산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- TOD#23:59:59.999999999 + T#0.000001ms → TOD#0:0:0.000000000
- TOD#0:0:0.000000000 + T#-0.000001ms → TOD#23:59:59.999999999

ADD_DT_TIME

날짜 시각에 시간을 가산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ADD_DT_ TIME	날짜 시각과 시간의 가산	FUN	(@)ADD_DT_TIME EN ENO In1 Out In2	Out:=ADD_DT_TIME(In1, In2);

변수

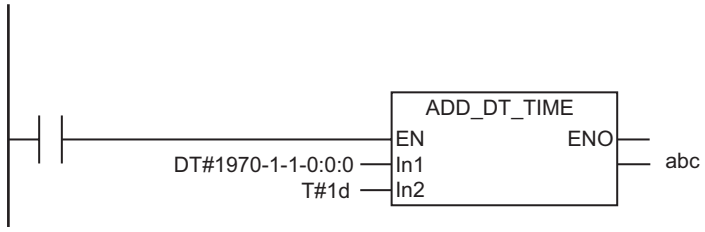
	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피가산 날짜 시각	입력	피가산 날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
In2	가산 시간		가산 시간		ns	T#0s
Out	가산 결과 날짜 시각	출력	가산 결과 날짜 시각	데이터 형식에 따름	연월일시분초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																			○	
In2																○				
Out																			○	

기능

날짜 시각 'In1'에 시간 'In2'를 가산합니다. 가산 결과 'Out'은 날짜 시각입니다. 윤년도 포함합니다.
'In1'=DT#1970-1-1-0:0:0, 'In2'=T#1d인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=ADD_DT_TIME(DT#1970-1-1-0:0:0, T#1d);
```

	'In1'	DT#1970-1-1-0:0:0
+	'In2'	T#1d
	'Out'=abc	DT#1970-1-2-0:0:0

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

사용상의 주의 사항

가산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- DT#2554-7-21-23:34:33.709551615 + T#0.000001ms → DT#1970-1-1-0:0:0
- DT#1970-1-1-0:0:0 + T#-0.000001ms → DT#2554-7-21-23:34:33.709551615

SUB_TIME

시간에서 시간을 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_TIME	시간 감산	FUN	(@)SUB_TIME EN ENO In1 Out In2	Out:=SUB_TIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 시간	입력	피감산 시간	데이터 형식에 따름	ns	T#0s
In2	감산 시간		감산 시간			
Out	감산 결과 시간	출력	감산 결과 시간	데이터 형식에 따름	ns	–

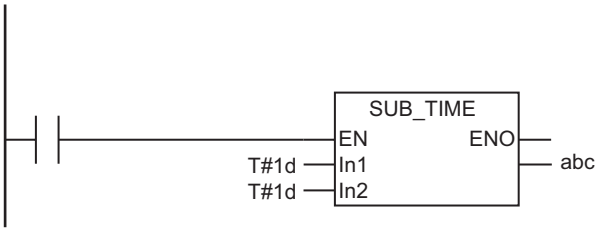
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																○				
In2																○				
Out																○				

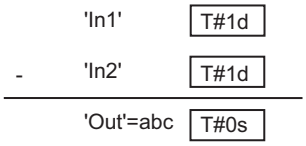
기능

시간 'In1'에서 시간 'In2'를 감산합니다. 감산 결과 'Out'도 시간입니다.

'In1'='In2'=T#1d인 경우의 예가 아래에 나와 있습니다.

LD





ST

```
abc:=SUB_TIME(T#1d, T#1d);
```

사용상의 주의 사항

감산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- T#106751d_23h_47m_16s_854.775807ms - T#-0.000001ms
  → T#-106751d_23h_47m_16s_854.775808ms
- T#-106751d_23h_47m_16s_854.775808ms - T#0.000001ms
  → T#106751d_23h_47m_16s_854.775807ms

SUB_TOD_TIME

시각에서 시간을 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_TOD_ TIME	시각과 시간의 감산	FUN	(@)SUB_TOD_TIME EN ENO In1 Out In2	Out:=SUB_TOD_TIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 시각	입력	피감산 시각	데이터 형식에 따름	시분초	TOD# 0:0:0
In2	감산 시간		감산 시간		ns	T#0s
Out	감산 결과 시각	출력	감산 결과 시각	데이터 형식에 따름	시분초	–

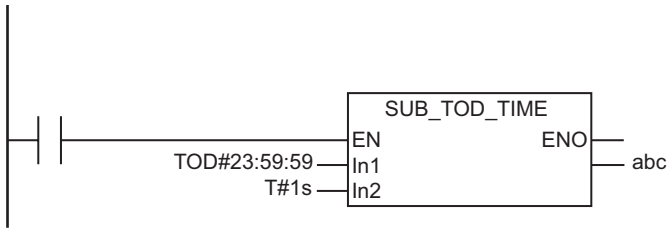
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																		○		
In2																○				
Out																		○		

기능

시각 'In1'에서 시간 'In2'를 감산합니다. 감산 결과 'Out'은 시각입니다.

'In1'=TOD#23:59:59, 'In2'=T#1s인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SUB_TOD_TIME(TOD#23:59:59, T#1s);
```

	'In1'	TOD#23:59:59
-	'In2'	T#1s
	'Out'=abc	TOD#23:59:58

사용상의 주의 사항

감산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- TOD#23:59:59.999999999 - T#-0.000001ms → TOD#0:0:0
- TOD#0:0:0 - T#0.000001ms → TOD#23:59:59.999999999

SUB_TOD_TOD

시각에서 시각을 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_TOD_ TOD	시각 감산	FUN	(@)SUB_TOD_TOD EN ENO In1 Out In2	Out:=SUB_TOD_TOD(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 시각	입력	피감산 시각	데이터 형식에 따름	시분초	TOD# 0:0:0
In2	감산 시각		감산 시각			
Out	감산 결과 시간	출력	감산 결과 시간	데이터 형식에 따름	ns	–

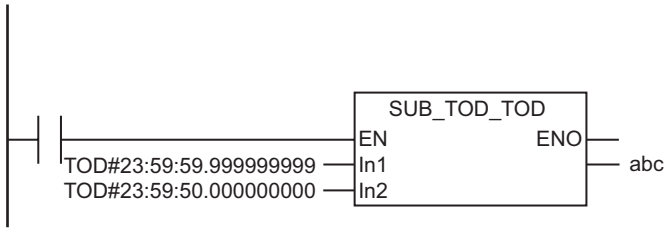
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																		○		
In2																		○		
Out																○				

기능

시각 'In1'에서 시각 'In2'를 감산합니다. 감산 결과 'Out'은 시간입니다.

'In1'=TOD#23:59:59.999999999, 'In2'=TOD#23:59:50.000000000인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SUB_TOD_TOD(TOD#23:59:59.999999999,
                 TOD#23:59:50.000000000);
```

	'In1'	TOD#23:59:59.999999999
-	'In2'	TOD#23:59:50.000000000
	'Out'=abc	T#0d0h0m9.999999999s

SUB_DATE_DATE

날짜에서 날짜를 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_DATE _DATE	날짜 감산	FUN	(@)SUB_DATE_DATE EN ENO In1 Out In2	Out:=SUB_DATE_DATE(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 날짜	입력	피감산 날짜	데이터 형식에 따름	연월일	D#1970-1-1
In2	감산 날짜		감산 날짜			
Out	감산 결과 시간	출력	감산 결과 시간	데이터 형식에 따름	ns	–

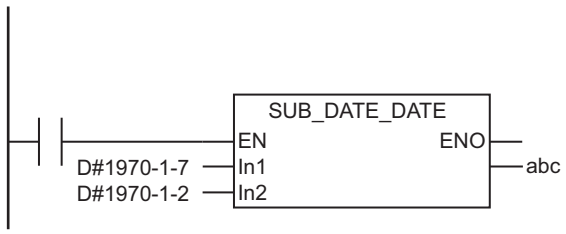
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																	○			
In2																	○			
Out																○				

기능

날짜 'In1'에서 날짜 'In2'를 감산합니다. 감산 결과 'Out'은 시간입니다.

'In1'=D#1970-1-7, 'In2'=D#1970-1-2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SUB_DATE_DATE(D#1970-1-7, D#1970-1-2);
```

	'In1'	D#1970-1-7
-	'In2'	D#1970-1-2
	'Out'=abc	T#5d0h0m0.000000000s

SUB_DT_DT

날짜 시각에서 날짜 시각을 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_DT_DT	날짜 시각 감산	FUN	(@)SUB_DT_DT EN ENO In1 Out In2	Out:=SUB_DT_DT(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 날짜 시각	입력	피감산 날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
In2	감산 날짜 시각		감산 날짜 시각			
Out	감산 결과 시간	출력	감산 결과 시간	데이터 형식에 따름	ns	–

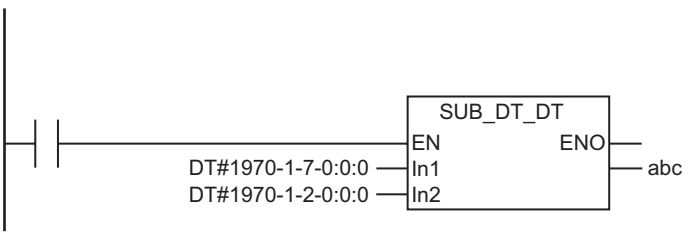
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																			○	
In2																			○	
Out																○				

기능

날짜 시각 'In1'에서 날짜 시각 'In2'를 감산합니다. 감산 결과 'Out'은 시간입니다.

'In1'=DT#1970-1-7-0:0:0, 'In2'=DT#1970-1-2-0:0:0인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SUB_DT_DT(DT#1970-1-7-0:0:0,
               DT#1970-1-2-0:0:0);
```

	'In1'	DT#1970-1-7-0:0:0
-	'In2'	DT#1970-1-2-0:0:0
	'Out'=abc	T#5d

관련된 시스템 정의 변수

변수명	명칭	데이터형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

사용상의 주의 사항

연산 결과가 'Out'의 유효 범위 밖인 경우, 'Out'의 값은 잘못된 값이 됩니다.

SUB_DT_TIME

날짜 시각에서 시간을 감산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SUB_DT_ TIME	날짜 시각과 시간의 감산	FUN	(@)SUB_DT_TIME EN ENO In1 Out In2	Out:=SUB_DT_TIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피감산 날짜 시각	입력	피감산 날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
In2	감산 시간		감산 시간		ns	T#0s
Out	감산결과 날짜 시각	출력	감산 결과 날짜 시각	데이터 형식에 따름	연월일시분초	–

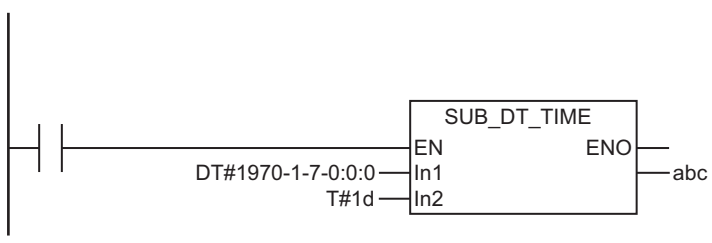
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																			○	
In2																○				
Out																			○	

기능

날짜 시각 'In1'에서 시간 'In2'를 감산합니다. 감산 결과 'Out'은 날짜 시각입니다. 윤년도 포함합니다.

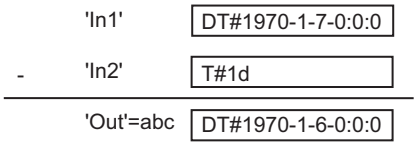
'In1'=DT#1970-1-7-0:0:0, 'In2'=T#1d인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SUB_DT_TIME(DT#1970-1-7-0:0:0, T#1d);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

사용상의 주의 사항

감산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.

- DT#2554-7-21-23:34:33.709551615 - T#-0.000001ms → DT#1970-1-1-0:0:0
- DT#1970-1-1-0:0:0 - T#0.000001ms → DT#2554-7-21-23:34:33.709551615

MULTIME

시간을 지정 승수로 승산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MULTIME	시간 승산	FUN	(@)MULTIME EN ENO In1 Out In2	Out:=MULTIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피승산 시간	입력	피승산 시간	데이터 형식에 따름	ns	T#0s
In2	승수		승수		–	(*)
Out	승산 결과 시간	출력	승산 결과 시간	데이터 형식에 따름	ns	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

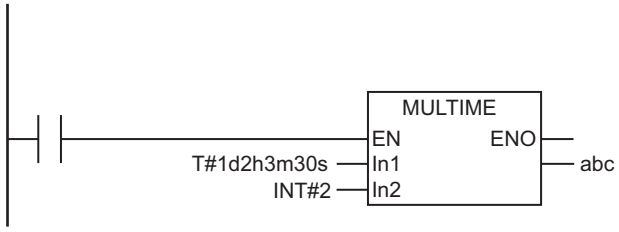
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																○				
In2						○	○	○	○	○	○	○	○	○	○					
Out																○				

기능

시간 'In1'과 승수 'In2'를 승산합니다. 승산 결과 'Out'은 시간입니다.

'In1'=T#1d2h3m30s, 'In2'=INT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=MULTIME(T#1d2h3m30s, INT#2);
```



사용상의 주의 사항

- 'In2'가 실수인 경우, 승산 결과의 1ns 미만은 오사오입합니다. 오사오입이란, 다음 표와 같은 처리를 의미합니다.

1ns 미만의 값	처리	예
0.5보다 작음	버림	1.49→1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2
0.5보다 큼	올림	1.51→2

- 'In2'의 값이 0, +∞, -∞, 비수인 경우, 'Out'의 값은 다음과 같습니다.

'In2'의 값	'Out'의 값
0	T#0s
+∞	T#-106751d23h47m16.854775808s
-∞	T#-106751d23h47m16.854775808s
비수	T#-106751d23h47m16.854775808s

- 승산 결과가 'Out'의 유효 범위를 초과한 경우, 이상 상태가 되지 않고 다음과 같이 동작합니다.
  - T#53375d_23h_53m_38s_427.387904ms * USINT#2
    → T#-106751d_23h_47m_16s_854.775808ms
  - T#-53375d_23h_53m_38s_427.387905ms * USINT#2
    → T#106751d_23h_47m_16s_854.775806ms

DIVTIME

시간을 지정 제수로 제산합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DIVTIME	시간 제산	FUN	(@)DIVTIME EN ENO In1 Out In2	Out:=DIVTIME(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	피제산 시간	입력	피제산 시간	데이터 형식에 따름	ns	T#0s
In2	제수		제수		–	(*)
Out	제산 결과 시간	출력	제산 결과 시간	데이터 형식에 따름	ns	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

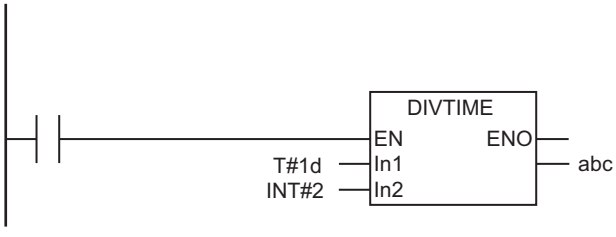
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																○				
In2						○	○	○	○	○	○	○	○	○	○					
Out																○				

기능

시간 'In1'을 제수 'In2'로 제산합니다. 제산 결과 'Out'은 시간입니다.

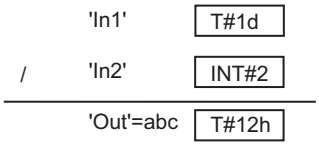
'In1'=T#1d, 'In2'=INT#2인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=DIVTIME(T#1d, INT#2);
```



사용상의 주의 사항

- 'In2'의 값이 0, +∞, -∞, 비수인 경우, 'Out'의 값은 다음과 같습니다.

'In2'의 값	'Out'의 값
0	T#-106751d23h47m16.854775808s
+∞	T#0s
-∞	T#0s
비수	비수

- 'In2'가 실수인 경우, 수 ns의 오차가 발생하는 경우가 있습니다.
- 'In2'가 실수인 경우, 제산 결과의 1ns 미만은 오사오입합니다. 오사오입이란, 다음 표와 같은 처리를 의미합니다.

1ns 미만의 값	처리	예
0.5보다 작음	버림	1.49→1
0.5	1의 자리의 값이 짝수라면 버리고 홀수라면 올림	1.50→2 2.50→2
0.5보다 큼	올림	1.51→2

- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In2'가 정수 형식이고 값이 0인 경우

CONCAT_DATE_TOD

날짜와 시각을 결합합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CONCAT_ DATE_TOD	날짜와 시각의 결합	FUN	(@)CONCAT_DATE_TOD EN In1 In2 ENO Out	Out:=CONCAT_DATE_TOD(In1, In2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In1	날짜	입력	날짜	데이터 형식에 따름	연월일	D#1970-1-1
In2	시각		시각		시분초	TOD#0:0:0
Out	결합 결과 날짜 시각	출력	결합 결과 날짜 시각	데이터 형식에 따름	연월일시분초	–

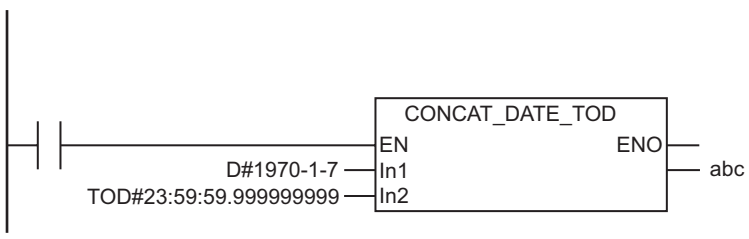
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																	○			
In2																		○		
Out																			○	

기능

날짜 'In1'과 시각 'In2'를 결합합니다. 결합 결과 'Out'은 날짜 시각입니다.

'In1'=D#1970-1-7, 'In2'=TOD#23:59:59.999999999인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=CONCAT_DATE_TOD(D#1970-1-7,
                     TOD#23:59:59.999999999);
```

	'In1'	D#1970-1-7
+	'In2'	TOD#23:59:59.999999999
	'Out'=abc	DT#1970-1-7-23:59:59.999999999

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 결합 결과가 'Out'의 유효 범위 밖('In1'의 값이 D#2554-7-21이며 'In2'의 값이 TOD#23:34:33.709551615보다 큼)인 경우

DT_TO_TOD

날짜 시각으로부터 시각을 잘라냅니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DT_TO_TOD	날짜 시각으로부터 시각 잘라내기	FUN	(@)DT_TO_TOD EN ENO In Out	Out:=DT_TO_TOD(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜 시각	입력	날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	시각	출력	시각	데이터 형식에 따름	시분초	–

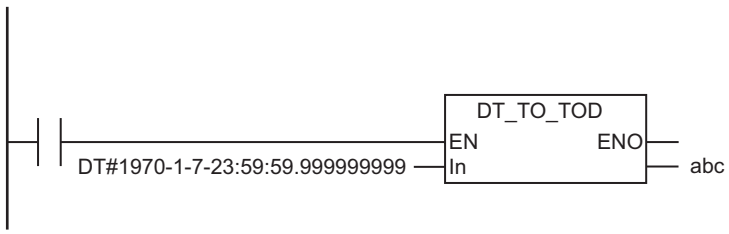
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out																		○		

기능

날짜 시각 'In'에서 시각 부분만을 잘라냅니다.

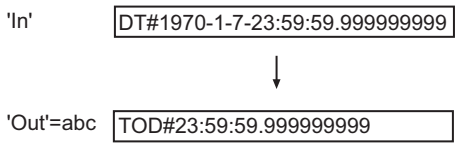
'In'=DT#1970-1-7-23:59:59.999999999인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=DT_TO_TOD(DT#1970-1-7-23:59:59.999999999);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

DT_TO_DATE

날짜 시각으로부터 날짜를 잘라냅니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DT_TO_DATE	날짜 시각으로부터 날짜 잘라내기	FUN	(@)DT_TO_DATE EN ENO In Out	Out:=DT_TO_DATE(In);

변수

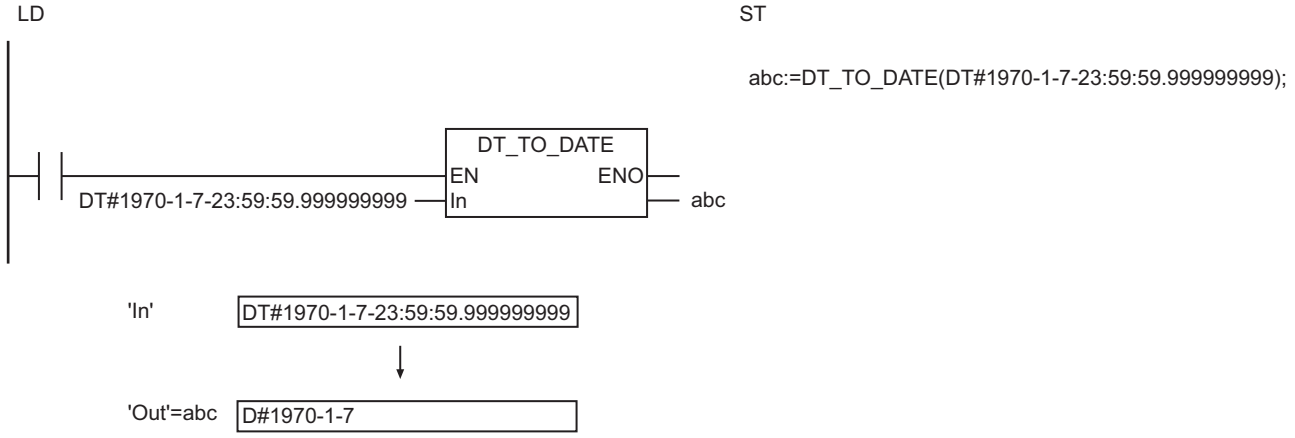
	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜 시각	입력	날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	날짜	출력	날짜	데이터 형식에 따름	연월일	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out																	○			

기능

날짜 시각 'In'에서 날짜 부분만을 잘라냅니다.

'In'=DT#1970-1-7-23:59:59.999999999인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

SetTime

시스템 시각을 보정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetTime	시계 보정	FUN	(@)SetTime EN ENO In Out	SetTime(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	보정 시각 데이터	입력	시스템 시각을 보정하기 위한 현재 날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	반환값	출력	항상 TRUE	TRUE만	–	–

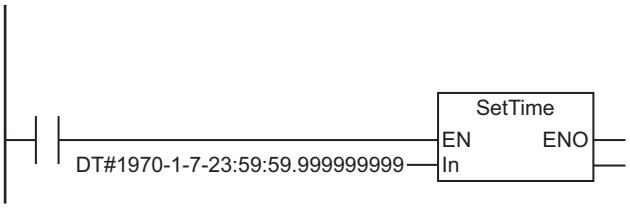
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out	○																			

기능

날짜 시각 'In'의 값을 시스템 시각으로 설정합니다.

'In'=DT#1970-1-7:23:59:59.999999999인 경우의 기술 예가 아래에 나와 있습니다.

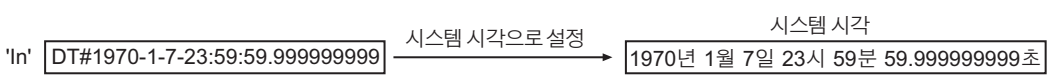
LD



ST

```
SetTime(DT#1970-1-7-23:59:59.999999999);
```

'In'의 값을 시스템 시각으로 설정합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

시스템 시각을 보정하는 방법에는, 이 명령 이외에 다음과 같은 것이 있습니다.

- Sysmac Studio를 사용하는 방법
- NTP 기능을 사용하는 방법

사용상의 주의 사항

- 'In'에는 GMT(그리니치 표준시)가 아닌, 설정된 표준 시간대의 표준시를 지정하십시오.
- 'In'에는 GMT의 1970-1-1-0:0:0.000000000을 하회하는 시각은 설정할 수 없습니다.
- 내장 시계를 갱신하면 타임래그가 발생합니다. 따라서 이 명령 실행 직후에 시각을 취득하면 갱신 전의 시각이 취득되는 경우가 있습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우
  - 'In'의 값이 GMT의 1970-1-1-0:0:0.000000000을 하회하는 경우

GetTime

현재 시각을 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetTime	시각 취득	FUN	(@)GetTime EN ENO Out	Out:=GetTime();

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	현재 시각	출력	현재 시각	데이터 형식에 따름	연월일시분초	–

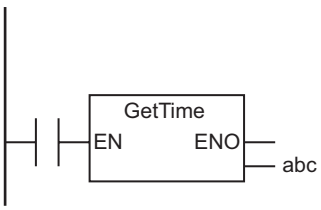
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out																			○	

기능

현재 시각을 읽어옵니다.
현재 시각은 GMT(그리니치 표준시)가 아닌, 설정된 표준 시간대의 표준시입니다.

기술 예가 아래에 나와 있습니다. 변수 abc에 현재 시각이 대입됩니다.

LD



ST

```
abc:=GetTime();
```

현재 시각을 읽어와서 abc에 대입합니다.

1970년 1월 7일 23시 59분 59.999999999초인 경우

시스템 시각

1970년 1월 7일 23시 59분 59.999999999초 → 시스템 시각의 대입 → 'Out'=abc DT#1970-1-7-23:59:59.999999999

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

- 현재 시각을 시스템 시각(1970년 1월 1일 0시 0분 0초부터의 초 수)으로 변환하고자 하는 경우에는 'DtToSec 명령(2-600페이지)'을 사용하십시오.
- 현재 시각을 연월일시분초로 분해하고자 하는 경우에는 'DtToDateStruct 명령(2-624페이지)'을 사용하십시오.
- 요일을 취득하고자 하는 경우에는 'GetDayOfWeek 명령(2-620페이지)'을 사용하십시오.

DtToSec

날짜 시각을 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DtToSec	날짜 시각→초 변환	FUN	(@)DtToSec EN ENO In Out	Out:=DtToSec(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜 시각	입력	날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	초	출력	1970년 1월 1일 0시 0분 0초부터의 초 수	0~18446744073	초	–

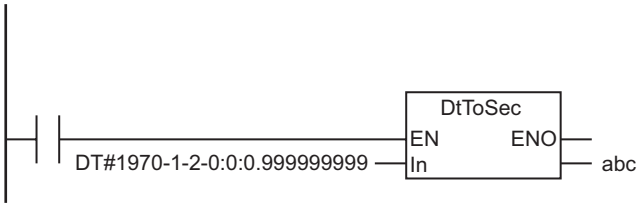
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out													○							

기능

날짜 시각 'In'을 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다. 변환 후의 값의 단위는 초입니다. 1초 미만의 값은 버립니다.

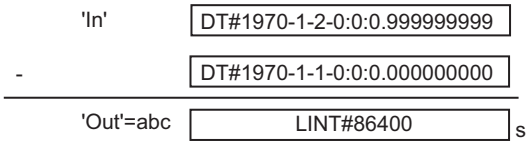
'In'=DT#1970-1-2-0:0:0.999999999인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=DtToSec(DT#1970-1-2-0:0:0.999999999);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜 시각으로 변환하고자 하는 경우에는 'SecToDt 명령 (2-604페이지)'을 사용하십시오.

DateToSec

날짜를 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DateToSec	날짜→초 변환	FUN	(@)DateToSec EN ENO In Out	Out:=DateToSec(In);

변수

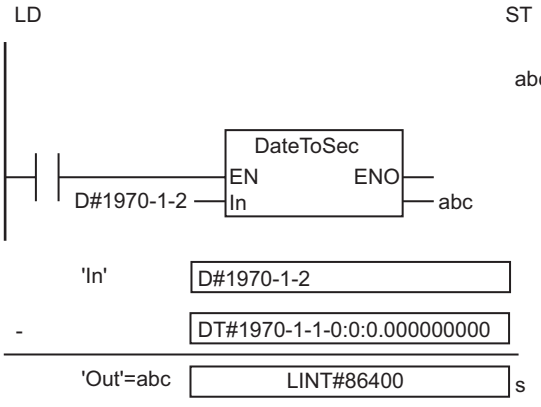
	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜	입력	날짜	데이터 형식에 따름	연월일	D#1970-1-1
Out	초	출력	1970년 1월 1일 0시 0분 0초부터의 초 수	0~18446659200	초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																	○			
Out													○							

기능

날짜 'In'의 0시 0분 0초의 시각을 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환합니다. 변환 후의 값의 단위는 초입니다.

'In'=D#1970-1-2인 경우의 예가 아래에 나와 있습니다.



참고

1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜로 변환하고자 하는 경우에는 🕮 'SecToDate 명령 (2-606페이지)'을 사용하십시오.

TodToSec

시각을 0시 0분 0초부터의 초 수로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TodToSec	시각→초 변환	FUN	(@)TodToSec EN ENO In Out	Out:=TodToSec(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	시각	입력	시각	데이터 형식에 따름	시분초	TOD#0:0:0
Out	초	출력	0시 0분 0초부터의 초 수	0~86399	초	–

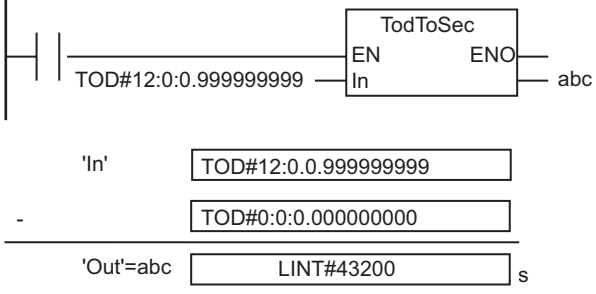
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																		○		
Out													○							

기능

시각 'In'을 0시 0분 0초부터의 초 수로 변환합니다. 변환 후의 값의 단위는 초입니다. 1초 미만의 값은 버립니다.

'In'=TOD#12:0:0.999999999인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=TodToSec(TOD#12:0:0.999999999);
```


참고

0시 0분 0초부터의 초 수를 시각으로 변환하고자 하는 경우에는 🕮 'SecToTod 명령(2-608페이지)'을 사용하십시오.

SecToDt

1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜 시각으로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SecToDt	초→날짜 시각 변환	FUN	(@)SecToDt EN ENO In Out	Out:=SecToDt(In);

변수

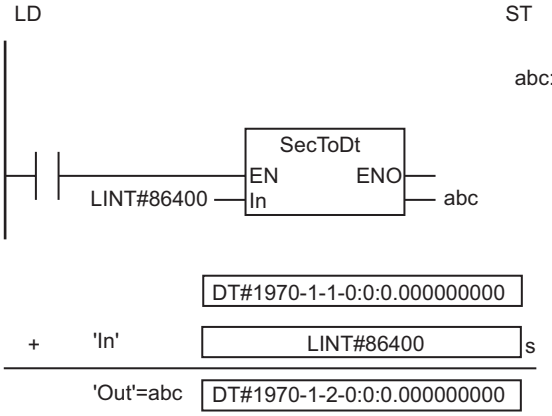
	명칭	입/출력	내용	유효 범위	단위	초기값
In	초	입력	1970년 1월 1일 0시 0분 0초부터의 초 수	0~18446744073	초	0
Out	날짜 시각	출력	날짜 시각	데이터 형식에 따름	연월일시분초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In													○							
Out																			○	

기능

1970년 1월 1일 0시 0분 0초부터의 초 수 'In'을 날짜 시각으로 변환합니다.

'In'=LINT#86400인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

날짜 시각을 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환하고자 하는 경우에는 'DtToSec 명령 (2-600페이지)'을 사용하십시오.

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'의 값이 유효 범위 밖인 경우

SecToDate

1970년 1월 1일 0시 0분 0초부터의 초 수를 날짜로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SecToDate	초→날짜 변환	FUN	(@)SecToDate EN ENO In Out	Out:=SecToDate(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	초	입력	1970년 1월 1일 0시 0분 0초부터의 초 수	0~18446744073	초	0
Out	날짜	출력	날짜	데이터 형식에 따름	연월일	–

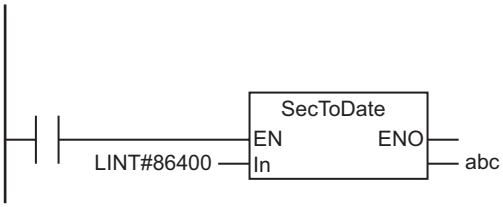
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In													○							
Out																	○			

기능

1970년 1월 1일 0시 0분 0초부터의 초 수 'In'을 날짜로 변환합니다. 1일 미만의 값은 버립니다.

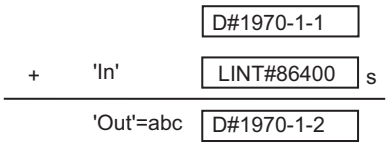
'In'=LINT#86400인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SecToDate(LINT#86400);
```



참고

날짜를 1970년 1월 1일 0시 0분 0초부터의 초 수로 변환하고자 하는 경우에는 📖 'DateToSec 명령 (2-602페이지)'을 사용하십시오.

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'의 값이 유효 범위 밖인 경우

SecToTod

0시 0분 0초부터의 초 수를 시각으로 변환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SecToTod	초→시각 변환	FUN	(@)SecToTod EN ENO In Out	Out:=SecToTod(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	초	입력	0시 0분 0초부터의 초 수	데이터 형식에 따름 (*)	초	0
Out	시각	출력	시각	데이터 형식에 따름	시분초	–

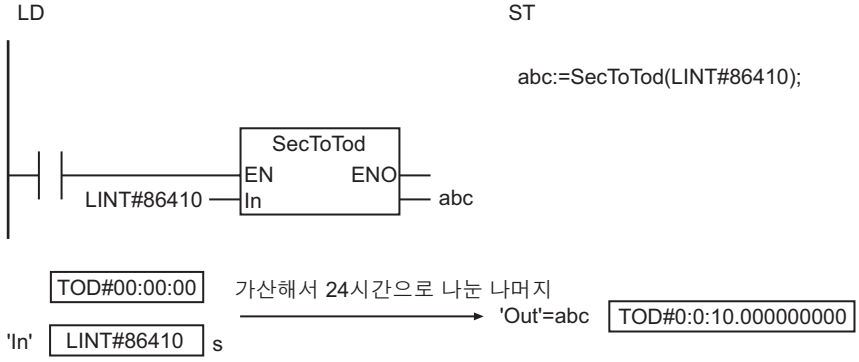
* 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In													○							
Out																		○		

기능

0시 0분 0초부터의 초 수 'In'을 시각으로 변환합니다. 'In'의 값이 24시간 이상인 경우에는 'In'의 값을 24시간으로 나눈 나머지를 시각으로 변환합니다.

'In'=LINT#86410인 경우의 예가 아래에 나와 있습니다.



참고

시각을 0시 0분 0초부터의 초 수로 변환하고자 하는 경우에는 'TodToSec 명령(2-603페이지)'을 사용하십시오.

사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'의 값이 유효 범위 밖인 경우

TimeToNanoSec

시간을 나노초 수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TimeToNanoSec	시간→나노초 변환	FUN	(@)TimeToNanoSec EN ENO In Out	Out:=TimeToNanoSec(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	시간	입력	시간	데이터 형식에 따름	ns	T#0s
Out	초 수	출력	초 수	(*)	ns	–

* -9223372036854775808~9223372036854775807

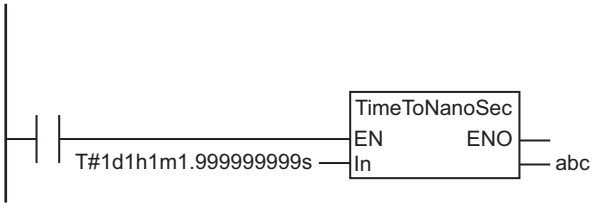
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																○				
Out													○							

기능

시간 'In'을 나노초 수로 변환합니다.

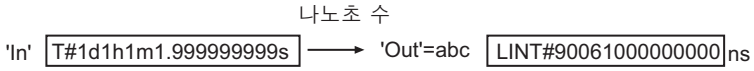
'In'=T#1d1h1m1.999999999s인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=TimeToNanoSec(T#1d1h1m1.999999999s);
```



참고

나노초를 시간으로 변환하고자 하는 경우에는 'NanoSecToTime 명령(2-612페이지)'을 사용하십시오.

TimeToSec

시간을 초 수로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TimeToSec	시간→초 변환	FUN	(@)TimeToSec EN ENO In Out	Out:=TimeToSec(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	시간	입력	시간	데이터 형식에 따름	ns	T#0s
Out	초 수	출력	초 수	-9223372036~ 9223372036	초	–

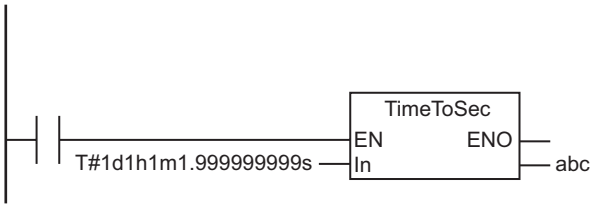
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																○				
Out													○							

기능

시간 'In'을 초 수로 변환합니다. 1초 미만의 값은 버립니다.

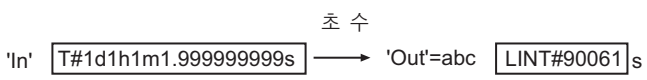
'In'=T#1d1h1m1.999999999s인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=TimeToSec(T#1d1h1m1.999999999s);
```



참고

초를 시간으로 변환하고자 하는 경우에는 'SecToTime 명령(2-613페이지)'을 사용하십시오.

사용상의 주의 사항

'In'의 단위는 나노초, 'Out'의 단위는 초입니다.

NanoSecToTime

나노초 수를 시간으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NanoSecTo Time	나노초→시간 변환	FUN	(@)NanoSecToTime EN ENO In Out	Out:=NanoSecToTime(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	초 수	입력	초 수	(*)	ns	0
Out	시간	출력	시간	데이터 형식에 따름	ns	–

* -9223372036854775808~9223372036854775807

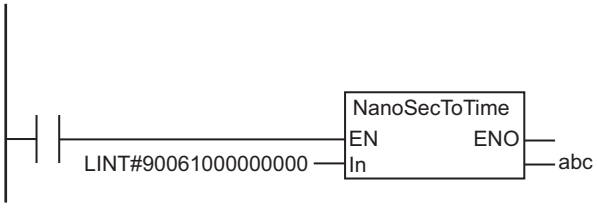
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In													○							
Out																○				

기능

나노초 수 'In'을 시간으로 변환합니다.

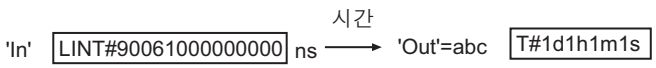
'In'=LINT#900610000000000인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=NanoSecToTime(LINT#90061000000000);
```



참고

시간을 나노초로 변환하고자 하는 경우에는 'TimeToNanoSec 명령(2-610페이지)'을 사용하십시오.

SecToTime

초 수를 시간으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SecToTime	초→시간 변환	FUN	(@)SecToTime EN ENO In Out	Out:=SecToTime(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	초 수	입력	초 수	-9223372036~ 9223372036	초	0
Out	시간	출력	시간	데이터 형식에 따름	ns	–

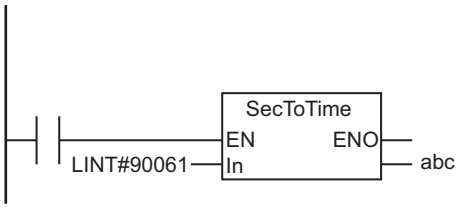
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In													○							
Out																○				

기능

초 수 'In'을 시간으로 변환합니다.

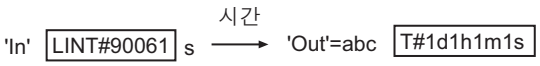
'In'=LINT#90061인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=SecToTime(LINT#90061);
```



참고

시간을 초로 변환하고자 하는 경우에는 'TimeToSec 명령(2-611페이지)'을 사용하십시오.

사용상의 주의 사항

- 'In'의 단위는 초, 'Out'의 단위는 나노초입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'In'의 값이 유효 범위 밖인 경우

ChkLeapYear

지정된 연도가 윤년인지 판정합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
ChkLeapYear	윤년 판별	FUN	(@)ChkLeapYear EN ENO In Out	Out:=ChkLeapYear(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	서기년도	입력	서기년도	1970~2554	년	1970
Out	판정 결과	출력	TRUE : 윤년임 FALSE : 윤년이 아님	데이터 형식에 따름	–	–

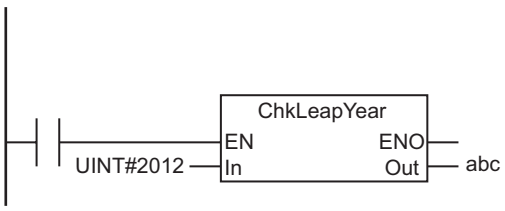
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In							○													
Out	○																			

기능

서기년도 'In'이 윤년인지 판정합니다. 윤년이라면, 판정 결과 'Out'의 값이 TRUE가 됩니다. 그렇지 않으면 FALSE가 됩니다.

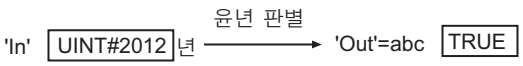
'In'=UINT#2012인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=ChkLeapYear(UINT#2012);
```



사용상의 주의 사항

'In'의 값이 유효 범위 밖인 경우, 이상 상태는 되지 않고 'Out'의 값은 잘못된 값이 됩니다.

GetDaysOfMonth

지정된 달의 일수를 취득합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
GetDaysOfMonth	달의 일수 취득	FUN	(@)GetDaysOfMonth EN ENO Year Out Month	Out:=GetDaysOfMonth(Year, Month);

변수

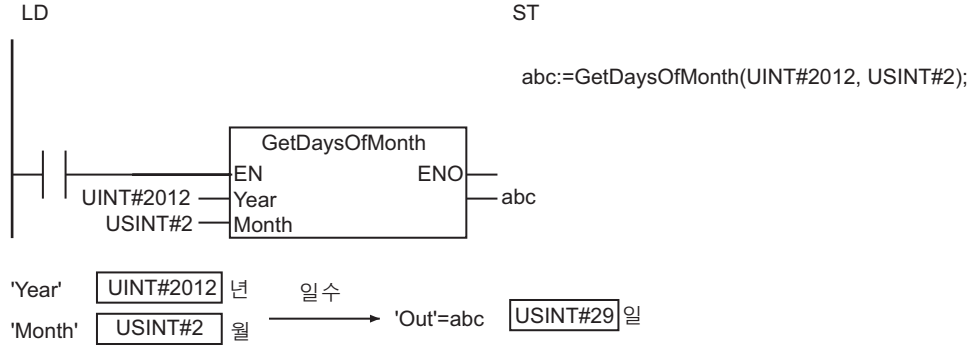
	명칭	입/출력	내용	유효 범위	단위	초기값
Year	년	입력	서기년도	1970~2554	년	1970
Month	월		월	1~12	월	1
Out	일수	출력	일수	28~31	일	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Year							○													
Month						○														
Out						○														

기능

'Year'로 표시된 연도의 'Month'로 표시된 달의 일수를 취득합니다.

'Year'=UINT#2012, 'Month'=USINT#2인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 'Year'의 값이 유효 범위 밖인 경우, 이상 상태는 되지 않고 'Out'의 값은 잘못된 값이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Month'의 값이 유효 범위 밖인 경우

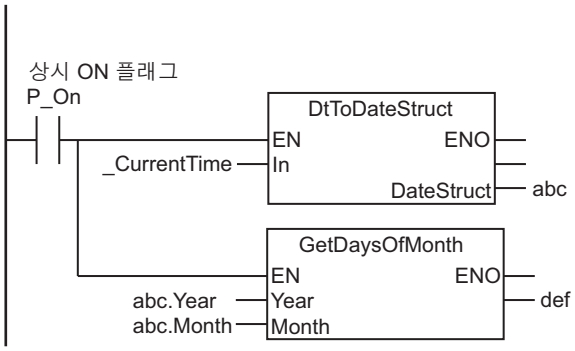
샘플 프로그램

이 달의 일수를 취득합니다.

LD

내부 변수	명칭	데이터형식	초기값	코멘트
	abc	_sDT	(Year:=0, Month:=0, Day:=0, Hour:=0, Min:=0, Sec:=0, NSec:=0)	날짜 시각
	def	USINT	0	이 달의 일수

내부 변수	명칭	데이터 형식	상수	코멘트
	_CurrentTime	DATE_AND_TIME	☑	시스템 시각



ST

내부 변수	명칭	데이터형식	초기값	코멘트
	abc	_sDT	(Year:=0, Month:=0, Day:=0, Hour:=0, Min:=0, Sec:=0, NSec:=0)	날짜 시각
	def	USINT	0	이 달의 일수

내부 변수	명칭	데이터 형식	상수	코멘트
	_CurrentTime	DATE_AND_TIME	☑	시스템 시각

```
DtToDateStruct(_CurrentTime, abc);
def:=GetDaysOfMonth(abc.Year, abc.Month);
```

DaysToMonth

1월 1일부터의 일수를 통해, 그 날이 몇 월인지 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DaysTo Month	일수→월 변환	FUN	(@)DaysToMonth EN ENO Year Out Days	Out:=DaysToMonth(Year, Days);

변수

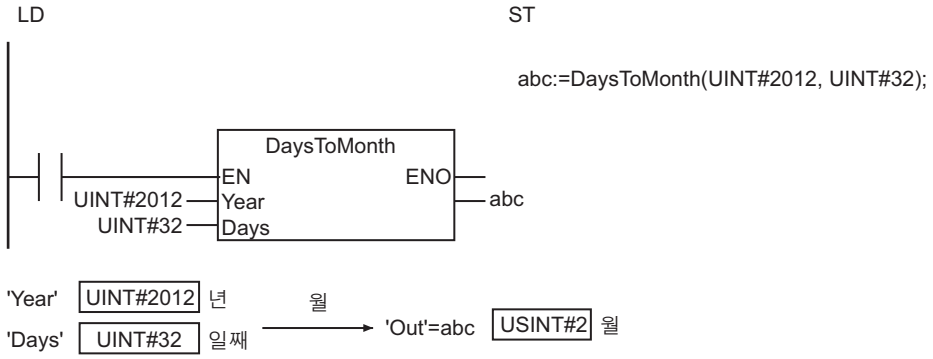
	명칭	입/출력	내용	유효 범위	단위	초기값
Year	년	입력	서기년도	1970~2554	년	1970
Days	일수		1월 1일부터의 일수	1~365 'Year'가 윤년인 경우는 1~366	일	1
Out	월	출력	월	1~12	월	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Year							○													
Days							○													
Out						○														

기능

년 'Year'의 1월 1일부터의 일수 'Days'로부터, 그 날이 몇 월인지 산출합니다.

'Year'=UINT#2012, 'Days'=UINT#32인 경우의 예가 아래에 나와 있습니다.

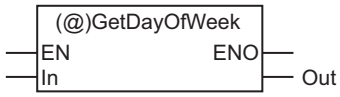


사용상의 주의 사항

- 'Year'의 값이 유효 범위 밖인 경우, 이상 상태는 되지 않고 'Out'의 값은 잘못된 값이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Days'의 값이 유효 범위 밖인 경우

GetDayOfWeek

지정된 연월일의 요일을 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetDayOf Week	요일 취득	FUN		Out:=GetDayOfWeek(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연월일	입력	연월일	데이터 형식에 따름	연월일	(*)
Out	요일	출력	요일	_MON, _TUE, _WED, _THU, _FRI, _SAT, _SUN	요일	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																	○		○	
Out	열거 형식 _eDAYOFWEEK 열거자는 기능 설명 참조																			

기능

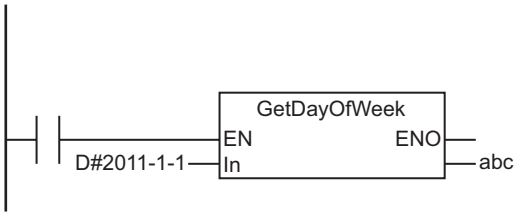
연월일 'In'으로 표시된 연월일의 요일을 취득합니다.

'Out'의 데이터 형식은 열거 형식 _eDAYOFWEEK입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_MON	월요일
_TUE	화요일
_WED	수요일
_THU	목요일
_FRI	금요일
_SAT	토요일
_SUN	일요일

'In'=D#2011-1-1인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=GetDayOfWeek(D#2011-1-1);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

GetWeekOfYear

지정된 연월일이 그 연도의 몇째 주인지 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetWeekOf Year	주 취득	FUN	(@)GetWeekOfYear EN ENO In Out	Out:=GetWeekOfYear(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	연월일	입력	연월일	데이터 형식에 따름	연월일	(*)
Out	주	출력	그 연도의 몇째 주인가	1~54	주	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

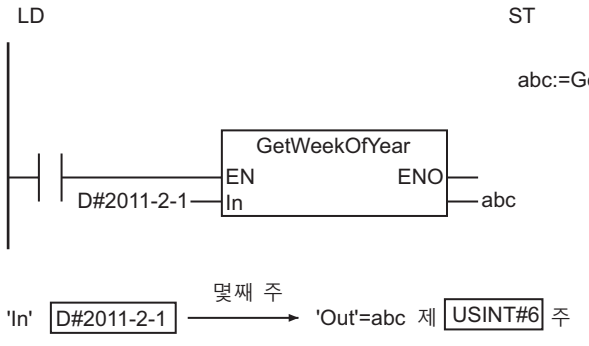
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																	○		○	
Out						○														

기능

연월일 'In'으로 표시된 연월일이 그 연도의 몇째 주인지 산출합니다.
월요일에서 일요일까지를 1주간의 단위로 해서, 일요일에서 월요일로 변하는 타이밍에 주 수를 카운트 업합니다.

1월 1일은 반드시 첫째 주가 됩니다.
예를 들면, 1월 1일이 목요일이라고 하면, 1월 1일(목)부터 1월 4일(일)이 첫째 주가 되며, 1월 5일(월)부터 1월 11일(일)이 둘째 주가 됩니다.

'In'=D#2011-2-1인 경우의 예가 아래에 나와 있습니다.

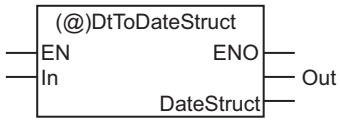


관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

DtToDateStruct

날짜 시각을 '년' '월' '일' '시' '분' '초' '나노초'로 분해합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DtToDate Struct	시각 분해	FUN		Out:=DtToDateStruct(In, DateStruct);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜 시각	입력	날짜 시각	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Out	반환값	출력	항상 TRUE	TRUE만	–	–
DateStruct	날짜 시각		'년' '월' '일' '시' '분' '초' '나노초'로 분해된 날짜 시각	–		

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Out	○																			
DateStruct	구조체 _sDT에 대한 자세한 내용은 기능 설명 참조																			

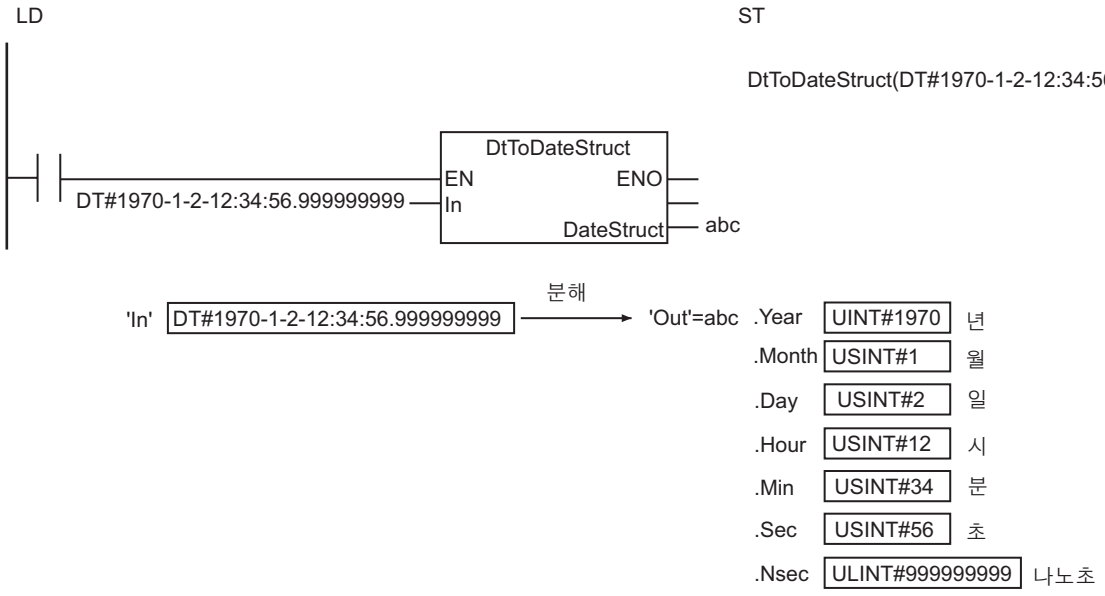
기능

날짜 시각 'In'을 '년' '월' '일' '시' '분' '초' '나노초'로 분해합니다.

분해한 날짜 시각 'DateStruct'의 데이터 형식은 구조체 _sDT입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
DateStruct	날짜 시각	'년' '월' '일' '시' '분' '초' '나노초'로 분해된 날짜 시각	_sDT	–	–	–
Year	년	년	UINT	1970~2554	년	–
Month	월	월	USINT	1~12	월	
Day	일	일	USINT	1~31	일	
Hour	시	시	USINT	0~23	시	
Min	분	분	USINT	0~59	분	
Sec	초	초	USINT	0~59	초	
Nsec	나노초	나노초	ULINT	0~999999999	나노초	

'In'=DT#1970-1-2-12:34:56.999999999인 경우의 예가 아래에 나와 있습니다.

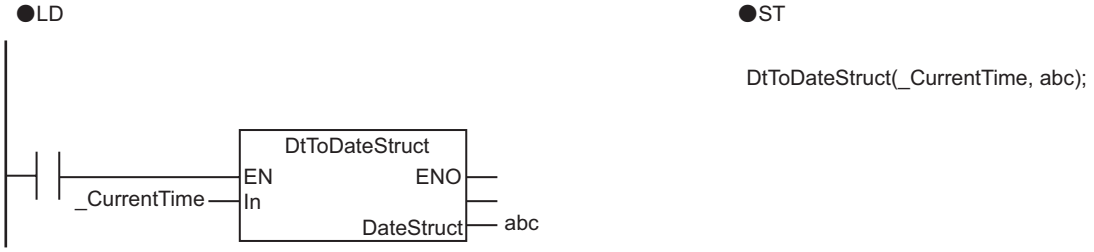


관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

- 분해된 '년' '월' '일' '시' '분' '초' '나노초'를 결합해서 날짜 시간으로 하고자 하는 경우에는 'DateStructToDt 명령(2-626페이지)'을 사용하십시오.
- 현재 시각을 구하는 예가 아래에 나와 있습니다.



사용상의 주의 사항

이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.

DateStructToDt

'년''월''일''시''분''초''나노초'로 분해된 날짜 시각을 결합합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
DateStructToDt	시각 결합	FUN	(@)DateStructToDt EN ENO In Out	Out:=DateStructToDt(In);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	날짜 시각	입력	'년''월''일''시''분''초''나노초'로 분해된 날짜 시각	–	–	–
Out	날짜 시각	출력	날짜 시각	데이터 형식에 따름	연월일시분초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	구조체 _sDT에 대한 자세한 내용은 기능 설명 참조																			
Out																			○	

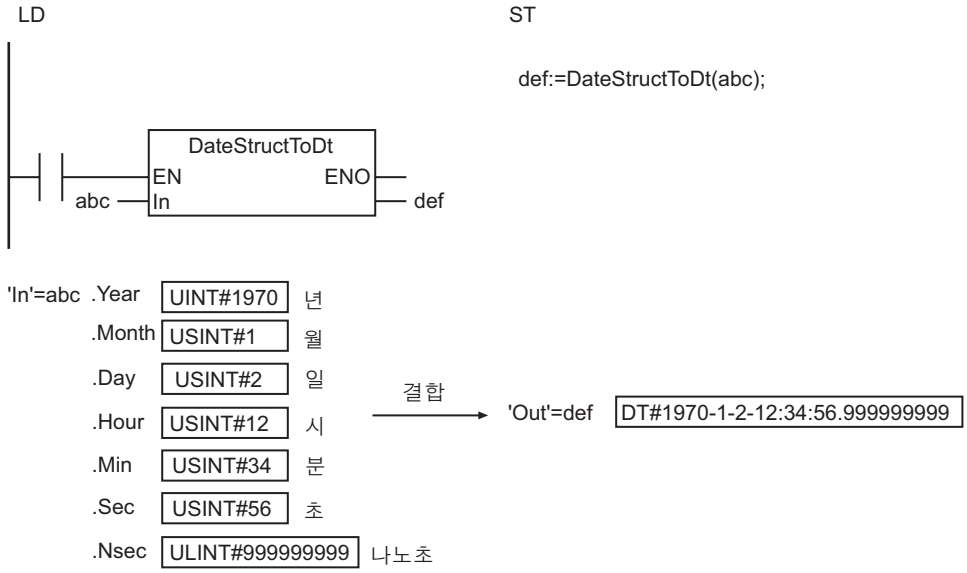
기능

'년''월''일''시''분''초''나노초'로 분해된 날짜 시각 'In'을 결합합니다.

'In'의 데이터 형식은 구조체 _sDT입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
In	날짜 시각	'년''월''일''시''분''초''나노초'로 분해된 날짜 시각	_sDT	–	–	–
Year	년	년	UINT	1970~2554	년	1970
Month	월	월	USINT	1~12	월	1
Day	일	일	USINT	1~31	일	
Hour	시	시	USINT	0~23	시	0
Min	분	분	USINT	0~59	분	
Sec	초	초	USINT	0~59	초	
Nsec	나노초	나노초	ULINT	0~999999999	나노초	

'In'의 각 멤버의 값이 'Year'=UINT#1970, 'Month'=USINT#1, 'Day'=USINT#2, 'Hour'USINT#=12, 'Min'=USINT#34, 'Sec'=USINT#56, 'Nsec'=ULINT#999999999인 경우의 예가 아래에 나와 있습니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_CurrentTime	시스템 시각	DT	시스템 내장 시계의 시각. 1970년 1월 1일 0시 0분 0초부터의 초 수

참고

날짜 시각을 '년' '월' '일' '시' '분' '초' '나노초'로 분해하고자 하는 경우에는 'DtToDateStruct 명령 (2-624페이지)'을 사용하십시오.

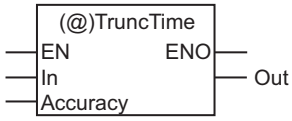
사용상의 주의 사항

다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.

- 'In'의 멤버 중 하나의 값이 유효 범위 밖인 경우
- 처리 결과가 'Out'의 유효 범위 밖인 경우

TruncTime

TIME 형식 변수에서 지정된 단위 미만의 값을 버립니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
TruncTime	시간 버림	FUN		Out:=TruncTime(In, Accuracy);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
In	대상 시간	입력	시간을 버리는 대상	데이터 형식에 따름	ns	T#0s
Accuracy	버리는 단위		버리지 않는 최소 시간 단위	_NANOSEC, _MICROSEC, _MILLISEC, _SEC	–	_NANOSEC
Out	버린 후 시간	출력	버린 후의 시간	데이터 형식에 따름	ns	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																○				
Accuracy	열거 형식 _eSUBSEC 열거자는 기능 설명 참조																			
Out																○				

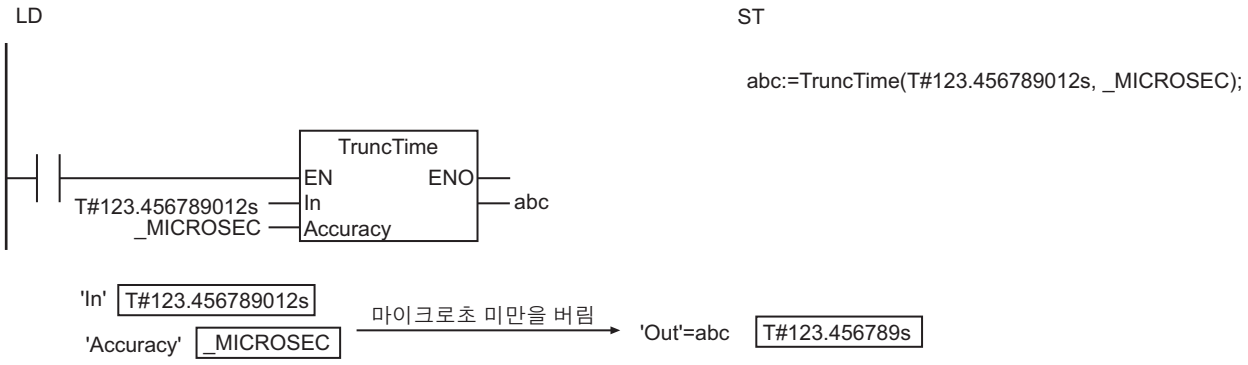
기능

대상 시간 'In'의 버리는 단위 'Accuracy' 미만의 값을 버립니다. 버린 후의 값은 버린 후 시간 'Out'에 저장됩니다.

'Accuracy'의 데이터 형식은 열거 형식 _eSUBSEC입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_NANOSEC	나노초
_MICROSEC	마이크로초
_MILLISEC	밀리초
_SEC	초

'In'=TIME#123.456789012s, 'Accuracy'=_MICROSEC인 경우의 예가 아래에 나와 있습니다.



참고

'EQ, = 명령(2-94페이지)' 등을 사용하여 시간 간의 크기를 비교할 때 이 명령을 사용하여 양쪽의 정밀도를 맞추십시오.

사용상의 주의 사항

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 사용할 수 있습니다.

샘플 프로그램

센서 출력의 ON 시간이 한계값 이상인지 확인하는 샘플 프로그램입니다.
동작 모드에는 한계값 설정 모드와 실행 모드가 있으며, 각각 다음과 같은 동작을 합니다.

동작 모드	동작
한계값 설정	센서 출력의 ON 시간을 계측하여 그 값을 한계값으로 설정합니다.
실행	센서 출력의 ON 시간을 계측하여 그 값과 한계값을 비교합니다. ON 시간이 한계값 이상이면 정상으로 판단합니다.

시간의 비교는 ms 단위의 정밀도로 수행합니다. 따라서 TruncTime 명령을 사용하여 계측 시간의 ms 미만의 값을 버립니다.
현재의 동작 모드는 변수 RecentMode에 저장되어 있는 것으로 가정합니다. 판단 결과는 변수 Result에 저장합니다. Result의 값은 TRUE가 정상, FALSE가 이상을 의미합니다.

글로벌 변수의 정의

데이터 형식 열거 형식

이름	열거값	코멘트
Mode		동작 모드
SET	0	한계값 설정
EXEC	1	실행

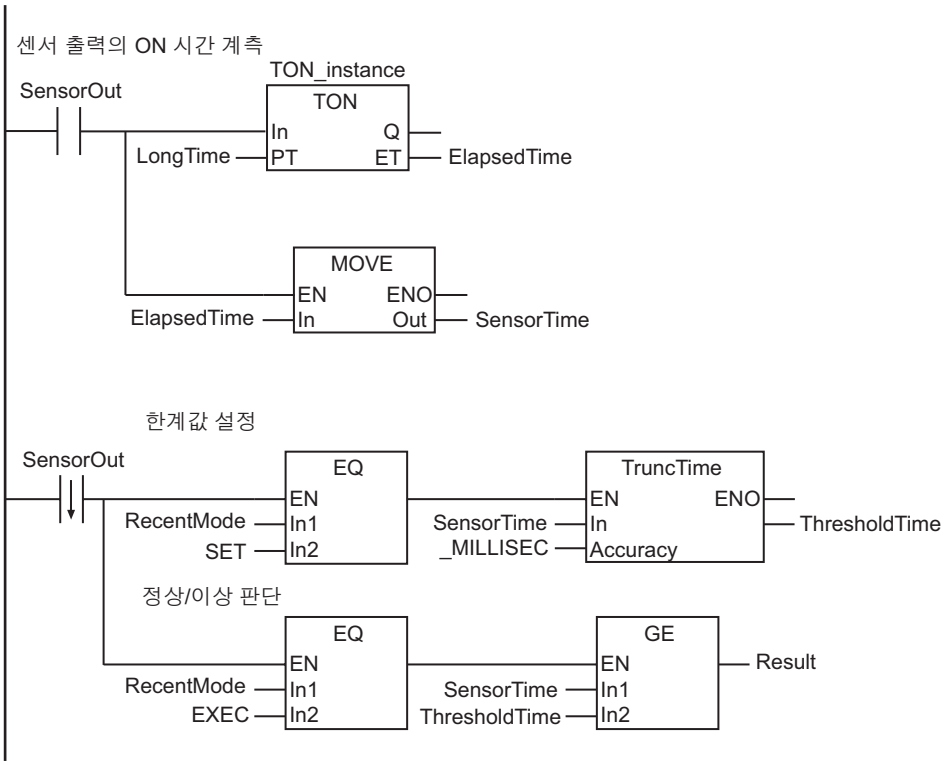
글로벌 변수

이름	데이터 형식	초기값	코멘트
RecentMode	Mode	SET	현재의 동작 모드

LD

내부 변수	이름	데이터 형식	초기값	코멘트
	SensorOut	BOOL	FALSE	센서 출력
	ElapsedTime	TIME	T#0s	경과 시간
	SensorTime	TIME	T#0s	센서의 ON 시간
	LongTime	TIME	T#1h	센서의 ON 시간보다 충분히 긴 시간
	ThresholdTime	TIME	T#0s	한계값
	Result	BOOL	FALSE	판단 결과 TRUE: 정상, FALSE: 이상
	TON_instance	TON		

외부 변수	이름	데이터 형식	코멘트
	RecentMode	Mode	현재의 동작 모드



ST

내부 변수	이름	데이터 형식	초기값	코멘트
	SensorOut	BOOL	FALSE	센서 출력
	ElapsedTime	TIME	T#0s	경과 시간
	SensorTime	TIME	T#0s	센서의 ON 시간
	LongTime	TIME	T#1h	센서의 ON 시간보다 충분히 긴 시간
	SensorDone	BOOL	FALSE	센서 출력 OFF 플래그
	ThresholdTime	TIME	T#0s	한계값
	Result	BOOL	FALSE	판단 결과 TRUE: 정상, FALSE: 이상
	TON_instance	TON		
	F_TRIG_instance	F_TRIG		

외부 변수	이름	데이터 형식	코멘트
	RecentMode	Mode	현재의 동작 모드

```
// TON 명령 실행
TON_instance(
   In:=SensorOut,        // 타이머 입력
   PT:=LongTime,         // 설정 시간
   ET=>ElapsedTime);  // 경과 시간

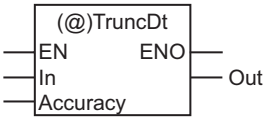
// TON의 경과 시간을 센서 ON 시간으로 설정
IF (SensorOut=TRUE) THEN
   SensorTime:=ElapsedTime;
END_IF;

// 센서 출력의 종료 감지
F_TRIG_instance(Clk:=SensorOut, Q=>SensorDone);
Result:=FALSE;

// 한계값 설정
IF (SensorDone=TRUE AND RecentMode=SET) THEN
   ThresholdTime:=TruncTime(
                  In          :=SensorTime,
                  Accuracy :=_MILLISEC);  // 정밀도 ms 단위
// 정상/이상 판단
ELSIF (SensorDone=TRUE AND RecentMode=EXEC) THEN
   IF (SensorTime >= ThresholdTime) THEN
      Result:=TRUE;
   END_IF;
END_IF;
```

TruncDt

DT 형식 변수에서 지정된 단위 미만의 값을 버립니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
TruncDt	날짜 시각 버림	FUN		Out:=TruncDt(In, Accuracy);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
In	대상 날짜 시각	입력	날짜 시각을 버리는 대상	데이터 형식에 따름	연월일시분초	DT#1970-1-1-0:0:0
Accuracy	버리는 단위		버리지 않는 최소 시각 단위	_NANOSEC, _MICROSEC, _MILLISEC, _SEC	–	_NANOSEC
Out	버린 후 날짜 시각	출력	버린 후의 날짜 시각	데이터 형식에 따름	연월일시분초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																			○	
Accuracy	열거 형식 _eSUBSEC 열거자는 기능 설명 참조																			
Out																			○	

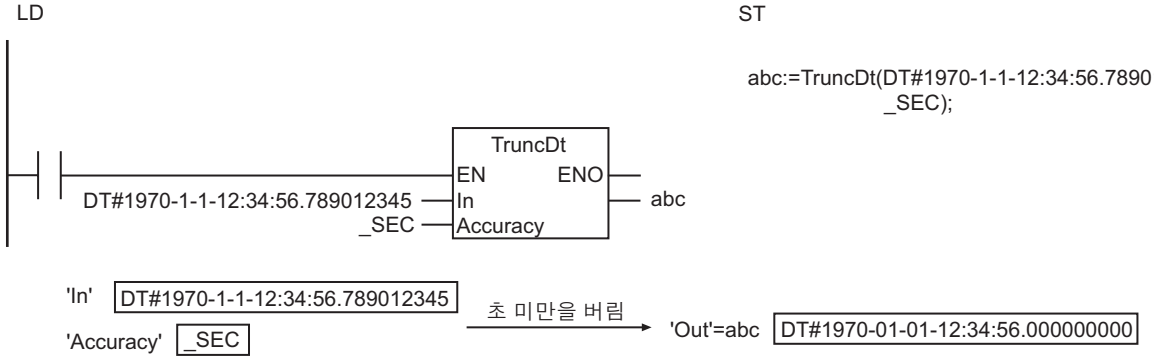
기능

대상 날짜 시각 'In'의 버리는 단위 'Accuracy' 미만의 값을 버립니다. 버린 후의 값은 버린 후 날짜 시각 'Out'에 저장됩니다.

'Accuracy'의 데이터 형식은 열거 형식 _eSUBSEC입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_NANOSEC	나노초
_MICROSEC	마이크로초
_MILLISEC	밀리초
_SEC	초

'In'=DT#1970-1-1-12:34:56.789012345, 'Accuracy'=_SEC인 경우의 예가 아래에 나와 있습니다.



참고

'EQ, = 명령(2-94페이지)' 등을 사용하여 날짜 시각 간의 크기를 비교할 때 이 명령을 사용하여 양쪽의 정밀도를 맞추십시오.

사용상의 주의 사항

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 사용할 수 있습니다.

샘플 프로그램

센서 출력이 ON된 날짜 시각과 그 때의 전압값을 기록하는 샘플 프로그램입니다.
날짜 시각은 ms 단위의 정밀도로 기록합니다.

센서 출력은 SensorOut, 전압값은 Voltage에 저장되어 있는 것으로 가정합니다. 현재의 날짜 시각은 GetTime 명령으로 취득합니다.
기록할 날짜 시각과 전압값은 해당 날짜 시각 및 전압값을 멤버로 하는 구조체 Recent의 형식으로 스택(Stack)에 순서대로 저장됩니다.

글로벌 변수의 정의

데이터 형식

이름	데이터 형식	코멘트
Record	STRUCT	구조체
DandT	DT	날짜 시각
Voltage	REAL	전압값

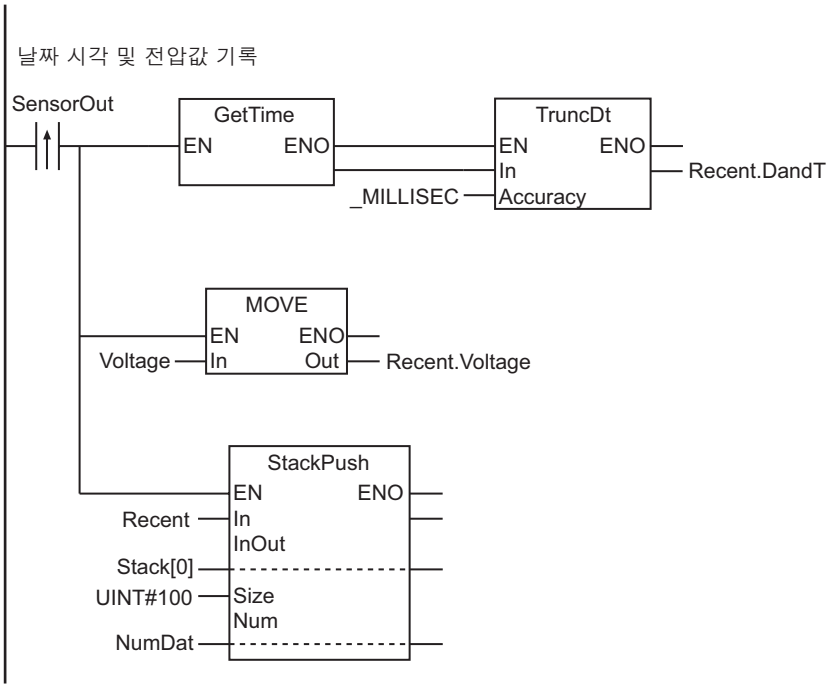
글로벌 변수

이름	데이터 형식	초기값	코멘트
Recent	Record	(DandT:=DT#1970-1-1-0:0:0,Voltage:=0.0)	현재값
Stack	ARRAY[0..99] OF Record	[100((DandT:=DT#1970-1-1-0:0:0,Voltage:=0.0))]	스택

LD

내부 변수	이름	데이터형식	초기값	코멘트
	SensorOut	BOOL	FALSE	센서 출력
	Voltage	REAL	0.0	전압값
	NumDat	UINT	UINT#0	현재의 기록 데이터 수

외부 변수	이름	데이터 형식	코멘트
	Recent	Record	현재값
	Stack	ARRAY[0..99] OF Record	스택



ST

내부 변수	이름	데이터형식	초기값	코멘트
	Trigger	BOOL	FALSE	트리거
	SensorOut	BOOL	FALSE	센서 출력
	Voltage	REAL	0.0	전압값
	NumDat	UINT	UINT#0	현재의 기록 데이터 수
	R_TRIG_instance	R_TRIG		

외부 변수	이름	데이터 형식	코멘트
	Recent	Record	현재값
	Stack	ARRAY[0..99] OF Record	스택

```
// 센서 출력이 ON되면 트리거 기동
R_TRIG_instance(SensorOut, Trigger);

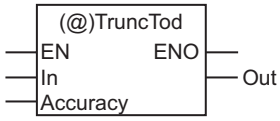
IF (Trigger=TRUE) THEN
   // 현재의 날짜 시각을 ms 단위의 정밀도로 저장
   Recent.DandT:=TruncDt(
                    In       :=GetTime(),    // 날짜 시각 취득
                   Accuracy :=_MILLISEC); // 정밀도 ms 단위
```

```
    // 전압값의 현재값을 취득
    Recent.Voltage:=Voltage;

    // 날짜 시각과 전압값을 스택에 저장
    StackPush(
        In     :=Recent,      // 날짜 시각과 전압값
        InOut :=Stack[0],     // 스택 배열
        Size   :=UINT#100,    // 스택 배열의 요소 수 100
        Num   :=NumDat);      // 현재의 기록 데이터 수
END_IF;
```

TruncTod

TOD 형식 변수에서 지정된 단위 미만의 값을 버립니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
TruncTod	시각 버림	FUN		Out:=TruncTod(In, Accuracy);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
In	대상 시각	입력	시각을 버리는 대상	데이터 형식에 따름	시분초	TOD# 0:0:0
Accuracy	버리는 단위		버리지 않는 최소 시각 단위	_NANOSEC, _MICROSEC, _MILLISEC, _SEC	–	_NANOSEC
Out	버린 후 시각	출력	버린 후의 시각	데이터 형식에 따름	시분초	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																		○		
Accuracy	열거 형식 _eSUBSEC 열거자는 기능 설명 참조																			
Out																		○		

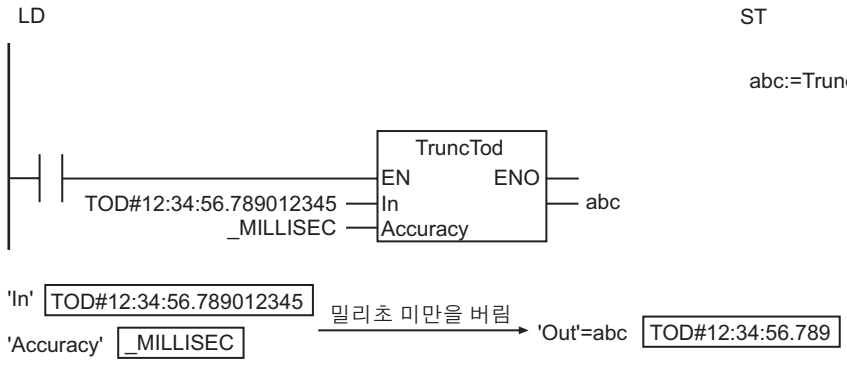
기능

대상 시각 'In'의 버리는 단위 'Accuracy' 미만의 값을 버립니다. 버린 후의 값은 버린 후 시각 'Out'에 저장됩니다.

'Accuracy'의 데이터 형식은 열거 형식 _eSUBSEC입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_NANOSEC	나노초
_MICROSEC	마이크로초
_MILLISEC	밀리초
_SEC	초

'In'=TOD#12:34:56.789012345, 'Accuracy'=_MILLISEC인 경우의 예가 아래에 나와 있습니다.



참고

'EQ, = 명령(2-94페이지)' 등을 사용하여 시각 간의 크기를 비교할 때 이 명령을 사용하여 양쪽의 정밀도를 맞추십시오.

사용상의 주의 사항

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.01 이상, Sysmac Studio Ver.1.02 이상에서 사용할 수 있습니다.

샘플 프로그램

센서 출력이 ON된 시각과 그 때의 전압값을 기록하는 샘플 프로그램입니다.
시각은 초 단위의 정밀도로 기록합니다.

센서 출력은 SensorOut, 전압값은 Voltage에 저장되어 있는 것으로 가정합니다. 현재 시각은 GetTime 명령과 DT_TO_TOD 명령을 조합하여 취득합니다.
기록할 시각과 전압값은 해당 시각 및 전압값을 멤버로 하는 구조체 Recent의 형식으로 스택(Stack)에 순서대로 저장됩니다.

글로벌 변수의 정의

데이터 형식

이름	데이터 형식	코멘트
Record	STRUCT	구조체
TofD	TOD	시각
Voltage	REAL	전압값

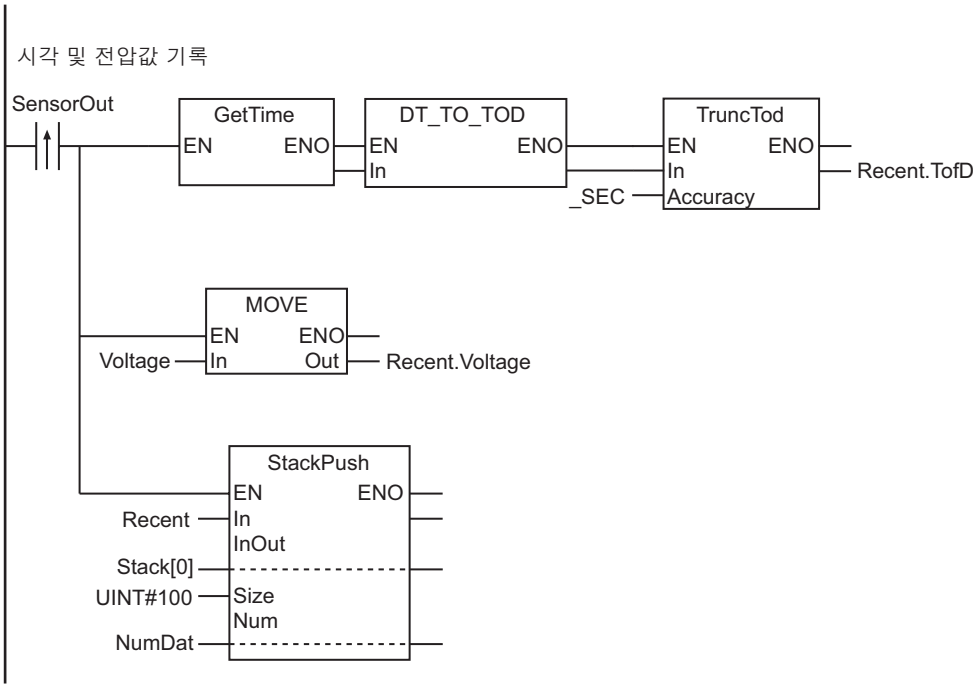
글로벌 변수

이름	데이터 형식	초기값	코멘트
Recent	Record	(TofD:=TOD#0:0:0,Voltage:=0.0)	현재값
Stack	ARRAY[0..99] OF Record	[100((TofD:=TOD#0:0:0,Voltage:=0.0))]	스택

LD

내부 변수	이름	데이터 형식	초기값	코멘트
	SensorOut	BOOL	FALSE	센서 출력
	Voltage	REAL	0.0	전압값
	NumDat	UINT	UINT#0	현재의 기록 데이터 수

외부 변수	이름	데이터 형식	코멘트
	Recent	Record	현재값
	Stack	ARRAY[0..99] OF Record	스택



ST

내부 변수	이름	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	트리거
	SensorOut	BOOL	FALSE	센서 출력
	TmpTod	TOD	TOD#0:0:0	임시 변수
	Voltage	REAL	0.0	전압값
	NumDat	UINT	UINT#0	현재의 기록 데이터 수
	R_TRIG_instance	R_TRIG		

외부 변수	이름	데이터 형식	코멘트
	Recent	Record	현재값
	Stack	ARRAY[0..99] OF Record	스택

```
// 센서 출력이 ON되면 트리거 기동
R_TRIG_instance(SensorOut, Trigger);

IF (Trigger=TRUE) THEN
   // 현재 시각을 초 단위의 정밀도로 저장
   TmpTod      :=DT_TO_TOD(GetTime()); // 시각 취득
   Recent.TofD :=TruncTod(
```

```
            In          :=TmpTod,
            Accuracy :=_SEC);   // 정밀도 초 단위

    // 전압값의 현재값을 취득
    Recent.Voltage:=Voltage;

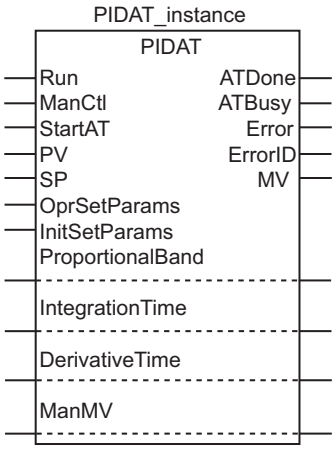
    // 시각과 전압값을 스택에 저장
    StackPush(
        In      :=Recent,       // 시각과 전압값
        InOut :=Stack[0],       // 스택 배열
        Size   :=UINT#100,  // 스택 배열의 요소 수 100
        Num   :=NumDat);    // 현재의 기록 데이터 수
END_IF;
```

아날로그 제어 명령

명령	명칭	페이지
PIDAT	자동 조정 가능한 PID 연산	2-642
TimeProportionalOut	시분할 비례 출력	2-670
LimitAlarm_**	상하한 경보 그룹	2-686
LimitAlarmDv_**	상하한 편차 경보 그룹	2-690
LimitAlarmDvStbySeq_**	대기 시퀀스가 있는 상하한 편차 경보 그룹	2-694
ScaleTrans	스케일 변환	2-709
AC_StepProgram	스텝 프로그램	2-712

PIDAT

자동 조정 가능한 PID 연산(목표값 필터 형식 2자유도 PID 제어)을 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
PIDAT	자동 조정 가능한 PID 연산	FB		PIDAT_instance(Run, ManCtl, StartAT, PV, SP, OprSetParams, InitSetParams, ProportionalBand, IntegrationTime, DerivativeTime, ManMV, ATDone, ATBusy, Error, ErrorID, MV);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Run	실행 조건	입력	TRUE : 실행 FALSE : 정지	데이터 형식에 따름	–	FALSE
ManCtl	매뉴얼/자동 전환		TRUE : 매뉴얼 운전 FALSE : 자동 운전			
StartAT	자동 조정 실행 조건		TRUE : 실행 FALSE : 중지			
PV	측정값 (현재값)		측정값(현재값)	(*1)		0
SP	목표값		목표값			
OprSet Params	운전 중 설정 파라메터		운전 중의 각종 설정 파라메터	–		–
InitSet Params	초기 설정 파라메터		초기 설정 파라메터			
Proportional Band	비례대	입출력	비례대	0.01~1000.00	%FS	–
Integration Time	적분 시간		적분 시간 값이 클수록 적분 효과 작음. 0인 경우는 적분 동작 없음	T#0.0000s~ T#10000.0000s (*2)	s	
Derivative Time	미분 시간		미분 시간 값이 클수록 미분 효과 큼. 0인 경우는 미분 동작 없음	T#0.0000s~ T#10000.0000s (*2)		
ManMV	수동 조작량		수동 조작량	-320~320	%	

	명칭	입/출력	내용	유효 범위	단위	초기값
ATDone	자동 조정 정상 종료	출력	TRUE : 정상 종료 FALSE : (*3)	데이터 형식에 따름	–	–
ATBusy	자동 조정 실행 중		TRUE : 실행 중 FALSE : 실행 중이 아님			
MV	조작량		조작량	-320~320	%	

*1 입력 범위 하한 'InitSetParams.RngLowLmt'의 값~입력 범위 상한 'InitSetParams.RngUpLmt'의 값

*2 0.0001s 미만은 버림

*3 이상 종료 또는 자동 조정을 실행하지 않고 PID 연산 중 또는 PID 연산 중이 아님

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Run	○																			
ManCtl	○																			
StartAT	○																			
PV														○						
SP														○						
OprSet Params	구조체 _sOPR_SET_PARAMS 자세한 내용은 기능 설명 참조																			
InitSet Params	구조체 _sINIT_SET_PARAMS 자세한 내용은 기능 설명 참조																			
Proportional Band														○						
Integration Time																○				
Derivative Time																○				
ManMV														○						
ATDone	○																			
ATBusy	○																			
MV														○						

기능

PID 연산을 수행하여 온도 조절기 등의 조작량을 제어합니다.
실행 조건 'Run'의 값이 TRUE가 되면 PID 연산을 시작합니다. 그 후에 'Run'의 값이 TRUE인 동안, 주기적으로 측정값 'PV'의 취득→PID 연산→조작량 'MV'의 출력을 반복합니다.
'Run'의 값이 FALSE가 되면 PID 연산을 종료합니다.

자동 조정 기능을 갖고 있어서 PID 정수의 최적값을 자동으로 구할 수 있습니다.
자동 조정 실행 조건 'StartAT'의 값이 TRUE가 되면 PID 정수의 자동 조정을 실행합니다.

구조체의 사양

운전 중 설정 파라메터 'OprSetParams'의 데이터 형식은 구조체 _sOPR_SET_PARAMS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
OprSetParams	운전 중 설정 파라메터	운전 중의 각종 설정 파라메터	_sOPR_SET_PARAMS	–	–	–
MVLowLmt	조작량 리미트 하한값	'MV'를 리미트 제어하는 하한값	REAL	-320~320 (*)	%	0
MVUpLmt	조작량 리미트 상한값	'MV'를 리미트 제어하는 상한값	REAL			100
ManReset Val	수동 리셋값	비례 동작으로, 편차 0인 경우의 'MV'의 값	REAL	-320~320		0
MVTrackSw	MV 추적 스위치	TRUE: ON FALSE: OFF	BOOL	데이터 형식에 따름	–	FALSE
MVTrackVal	MV 추적값	MV 추적 시 'MV'에 설정되는 값	REAL	-320~320	%	0
StopMV	정지 시 조작량	명령 실행 정지 시 'MV'에 설정되는 값	REAL			
ErrorMV	이상 시 조작량	이상 발생 시 'MV'에 설정되는 값	REAL			
Alpha	2자유도 PID 파라메터 α	목표값 필터 계수 α 0인 경우, 목표값 필터 무효	REAL	0.00~1.00	–	0.65
ATCalcGain	자동 조정 산출 게인	자동 조정 결과의 조정 계수 값이 클수록 안정성 중시, 작을수록 즉응성 중시	REAL	0.1~10.0		1.0
ATHystrs	자동 조정 히스테리시스	리미트 사이클의 히스테리시스 크기	REAL		%FS	0.2

* 단, 'MvLowLmt'<'MVUpLmt'

초기 설정 파라메터 'InitSetParams'의 데이터 형식은 구조체 _sINIT_SET_PARAMS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
InitSetParams	초기 설정 파라메터	초기 설정 파라메터	_sINIT_SET_PARAMS	–	–	–
SampTime	샘플링 주기	PID 연산을 수행하는 주기	TIME	T#0.0001s~ T#100.0000s	s	T#0.1s
RngLowLmt	입력 범위 하한	'PV' 및 'SP'의 하한값	REAL	-32000~ 32000(*)	–	0
RngUpLmt	입력 범위 상한	'PV' 및 'SP'의 상한값	REAL			100
DirOpr	동작 방향	TRUE: 정방향 동작 FALSE: 역방향 동작	BOOL	데이터 형식에 따름		FALSE

* 단, 'RngLowLmt'<'RngUpLmt'

각 변수의 의미

이 명령의 각 변수의 의미를 자세히 설명합니다.

● 'Run'(실행 조건)

이 명령의 실행 조건입니다.

값이 TRUE인 동안 PID 연산을 수행합니다. 값이 FALSE가 되면 PID 연산을 정지합니다.

● 'ManCtl'(매뉴얼/자동 전환)

이 명령에는 매뉴얼 운전과 자동 운전의 2가지 모드가 있습니다.

'ManCtl'의 값에 따라 어떤 모드로 운전할지 결정됩니다.

'ManCtl'의 값	운전 모드	'MV'의 값
TRUE	매뉴얼	'ManMV'의 값 (PID 연산은 수행하지 않습니다.)
FALSE	자동	PID 연산으로 산출한 값

● 'StartAT'(자동 조정 실행 조건)

PID 정수의 자동 조정 실행 조건입니다.

'StartAT'의 값이 TRUE인 상태에서 'Run'의 값이 FALSE→TRUE가 되면 운전 시작 시에 자동 조정을 실행합니다.

'Run'의 값이 TRUE이고 PID 연산 중에 'StartAT'의 값이 FALSE→TRUE가 되면 운전 중에 자동 조정을 시작합니다.

어떤 경우라도 자동 조정 중에 'StartAT'의 값이 TRUE→FALSE가 되면 자동 조정을 중지합니다.

자동 조정 기능에 대한 자세한 내용은 뒤에서 다룹니다.

● 'PV'(측정값)

제어 대상의 측정값입니다.

● 'SP'(목표값)

제어 대상의 목표값입니다.

● 'MVLowLmt'(조작량 리미트 하한값), 'MVUpLmt'(조작량 리미트 상한값)

'MV'의 값은 리미트 제어할 수 있습니다.

'MVLowLmt'와 'MVUpLmt'는 리미트 제어의 하한값과 상한값입니다.

반드시 'MVLowLmt'<'MVUpLmt'가 되도록 설정하십시오.

PID 연산으로 얻은 조작량	'MV'의 값
'MVLowLmt' 이하	'MvLowLmt'
'MVLowLmt' 이상 'MVUpLmt' 이하	PID 연산으로 얻은 조작량
'MVUpLmt' 이상	'MVUpLmt'

단, 'MV'에 정지 시 조작량 'StopMV', 이상 시 조작량 'ErrorMV', 수동 조작량 'ManMV' 중 하나가 설정된 경우에는 리미트 제어의 대상이 아닙니다.

● 'ManResetVal'(수동 리셋값)

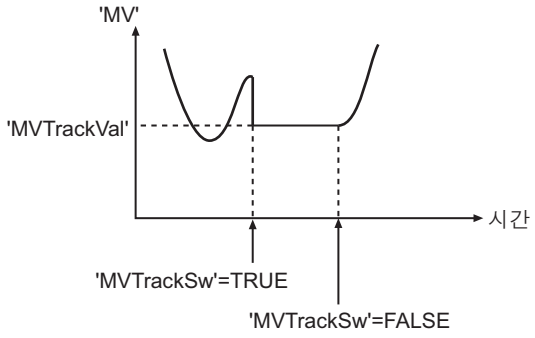
비례 동작에서 편차('PV'와 'SP'의 차)가 0이 된 경우의 'MV'의 값입니다.

'ManResetVal'의 값에 따라 비례 동작 범위의 위치가 결정됩니다.

적분 동작을 수행하는 경우, 수동 리셋값의 설정은 무효가 됩니다. 따라서 'ManResetVal'의 설정은 'IntegrationTime'의 값이 0인 경우에 유효합니다.

● 'MVTrackSw'(MV 추적 스위치)

자동 운전 시, 'MV'에 외부의 입력값(MV 추적값)을 강제로 설정하는 기능을 MV 추적이라고 합니다.
'MVTrackSw'의 값이 TRUE인 동안, MV 추적을 실행합니다.
'MVTrackSw'의 값이 FALSE가 되면 'MV'의 값은 PID 연산 결과로 되돌아갑니다. 이때 'MV'의 값은 범프리스(급변 없음)로 변화합니다.



● 'MVTrackVal'(MV 추적값)

MV 추적으로 'MV'에 강제로 설정하는 값입니다.
'MVTrackVal'의 값은 'MVLowLmt'와 'MVUpLmt'에 의해 리미트 제어됩니다.

● 'StopMV'(정지 시 조작량)

'Run'의 값이 FALSE가 되어 이 명령을 실행 정지했을 때 'MV'에 설정되는 값입니다.

● 'ErrorMV'(이상 시 조작량)

이상이 발생('Err'의 값이 TRUE)했을 때 'MV'에 설정되는 값입니다.
'ErrorMV'의 값이 유효 범위 안(-320~320)에 있지 않은 경우, 이상 시의 'MV'의 값은 0이 됩니다.

● 'Alpha'(2자유도 PID 파라메터 α)

목표값 필터의 계수를 결정하는 파라메터입니다.
자세한 내용은 목표값 필터 형식 2자유도 PID 제어 설명을 참조하십시오.
'Alpha'의 값은 통상적으로 0.65로 설정하십시오.

● 'ATCalcGain'(자동 조정 산출 게인)

자동 조정 실행으로 산출한 PID 정수를 실제 PID 정수에 반영할 때의 계수입니다.
1.00을 지정하면 자동 조정 결과를 그대로 반영하게 됩니다.
'ATCalcGain'의 값은 안정성을 중시할 때는 크게 하고, 즉응성을 중시할 때는 작게 합니다.

● 'ATHystrs'(자동 조정 히스테리시스)

자동 조정 시의 리미트 사이클 히스테리시스 크기를 나타냅니다.
'ATHystrs'의 값은 작을수록 높은 정확도로 조정 가능합니다. 단, 측정값이 안정되지 않아서 정상적인 조정이 어려운 경우에는 크게 합니다.
자세한 내용은 자동 조정 설명을 참조하십시오.

● 'SampTime'(샘플링 주기)

PID 연산을 수행하는 주기의 최소값입니다.
자세한 내용은 PID 연산의 실행 타이밍에 대한 설명을 참조하십시오. PID 연산을 실행한 뒤의 경과 시간이 'SampTime' 미만이면 다음 PID 연산이 수행되지 않습니다.

● 'RngLowLmt'(입력 범위 하한), 'RngUpLmt'(입력 범위 상한)

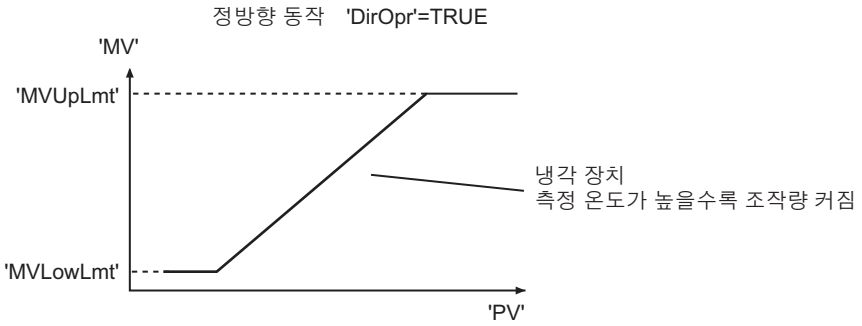
'PV' 및 'SP'의 하한값과 상한값입니다.
'PV' 및 'SP'에 연결하는 파라메터의 값이 이 범위에서 벗어난 경우, 이상 상태가 됩니다.
반드시 'RngLowLmt'<'RngUpLmt'가 되도록 설정하십시오.

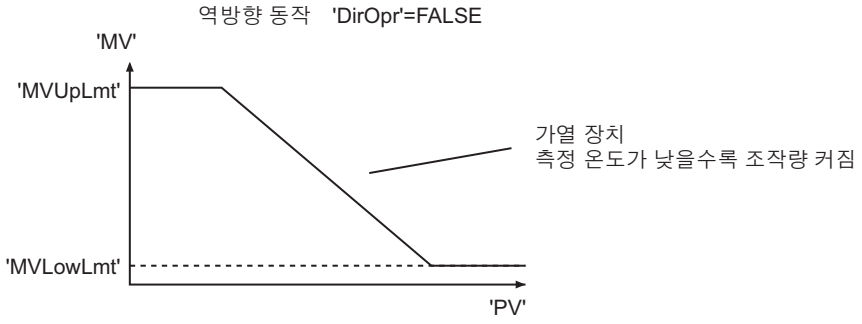
● 'DirOpr'(동작 방향)

'PV'의 크기에 따라 'MV'를 크게 할지 작게 할지 지정합니다.
각각을 정방향 동작, 역방향 동작이라고 합니다.

'DirOpr'의 값	의미	'MV'의 값
TRUE	정방향 동작	'PV'가 클수록 큼
FALSE	역방향 동작	'PV'가 클수록 작음

정방향 동작과 역방향 동작의 차이를 온도 제어의 예를 들어 설명합니다.
냉각 장치의 조작량을 제어하는 경우가 정방향 동작이 됩니다. 즉, 측정 온도가 높을수록 냉각 장치의 조작량을 크게 합니다. 이에 반해, 가열 장치의 조작량을 제어하는 경우가 역방향 동작입니다. 측정 온도가 낮을수록 가열 장치의 조작량을 크게 합니다.





● 'ProportionalBand'(비례대)

3개의 PID 정수 가운데 1개입니다. 자세한 내용은 비례 동작에 대한 설명을 참조하십시오.
'ProportionalBand'의 값이 클수록 오프셋이 커집니다. 작으면 헌팅이 발생합니다.

● 'IntegrationTime'(적분 시간)

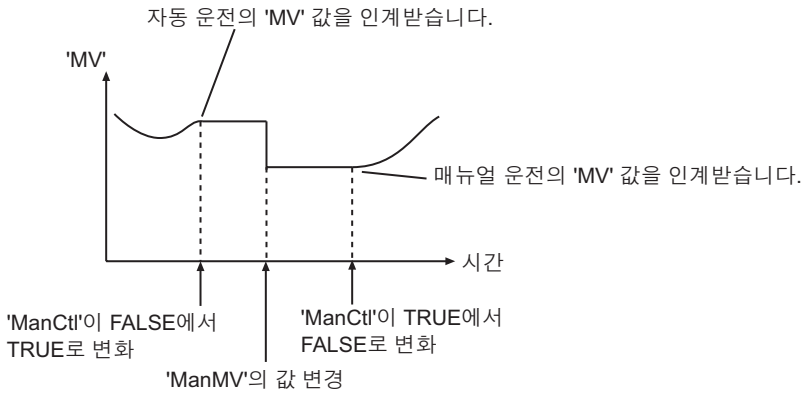
3개의 PID 정수 가운데 1개입니다. 자세한 내용은 적분 동작에 대한 설명을 참조하십시오.
'IntegrationTime'의 값이 클수록 적분 동작이 약해집니다.

● 'DerivativeTime'(미분 시간)

3개의 PID 정수 가운데 1개입니다. 자세한 내용은 미분 동작에 대한 설명을 참조하십시오.
'DerivativeTime'의 값이 클수록 미분 동작이 강해집니다.

● 'ManMV'(수동 조작량)

매뉴얼 운전 시('ManCtl'=TRUE)'에 'MV'에 설정되는 값입니다.
단, 자동 운전에서 매뉴얼 운전으로 전환된 직후에는 자동 운전의 'MV'의 값을 그대로 인계받습니다.
'ManMV'의 값이 'MV'에 설정되는 것은 매뉴얼 운전으로 전환한 후에 'ManMV'의 값을 변경했을 때입니다.
또한 매뉴얼 운전에서 자동 운전으로 전환된 직후에도 매뉴얼 운전의 'MV'의 값을 인계받습니다.
또한 'ManMV'의 값은 'MVLowLmt'와 'MVUpLmt' 사이가 아니라도 상관없습니다.



● 'ATDone'(자동 조정 정상 종료)

자동 조정이 정상 종료되었음을 나타내는 플래그입니다.
자동 조정이 정상 종료된 후 'StartAT'의 값이 TRUE인 동안 TRUE가 됩니다.
FALSE가 되는 것은 다음과 같은 경우입니다.

- 자동 조정 이상 종료 시
- 자동 조정 실행 중('ATBusy'의 값이 TRUE)
- 자동 조정을 실행하지 않고 PID 연산 중
- PID 연산 중이 아님('Run'의 값이 FALSE)
- 'StartAT'의 값이 FALSE임

● 'ATBusy'(자동 조정 실행 중)

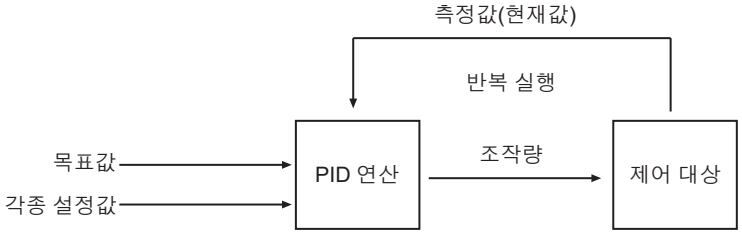
자동 조정이 실행 중임을 나타내는 플래그입니다.
자동 조정 실행 중에 TRUE가 됩니다. 그 이외에는 FALSE가 됩니다.

● 'MV'(조작량)

제어 대상에 전달하는 조작량입니다.

PID 연산의 개요

PID 연산이란 제어 대상의 현재값을 측정하여 그 값을 목표값에 가깝게 맞추기 위해 조작량 조절을 반복하는 피드백 제어 연산 방법의 하나입니다.
따라서 이 명령의 입력은 측정값, 목표값 및 연산을 위한 각종 설정값이며 출력은 조작량입니다. 측정→연산→조작량 출력을 주기적으로 반복함으로써 현재값이 목표값에 가까워집니다.

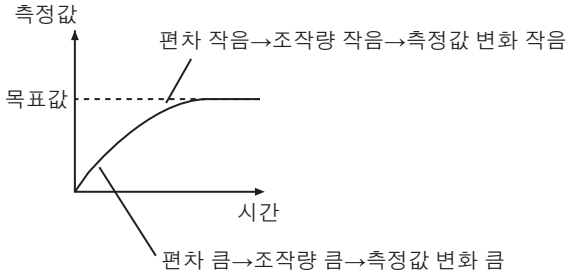


비례 동작(P), 적분 동작(I), 미분 동작(D)

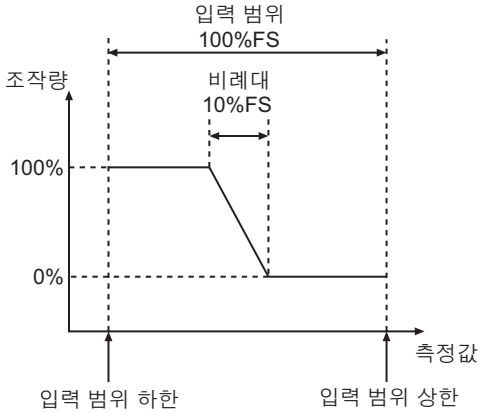
PID 연산은 비례 동작(Proportional action), 적분 동작(Integral action), 미분 동작(Derivative action)의 합성에 의해 수행됩니다. 다음에서 각 동작에 대해 설명합니다.

● 비례 동작(P)

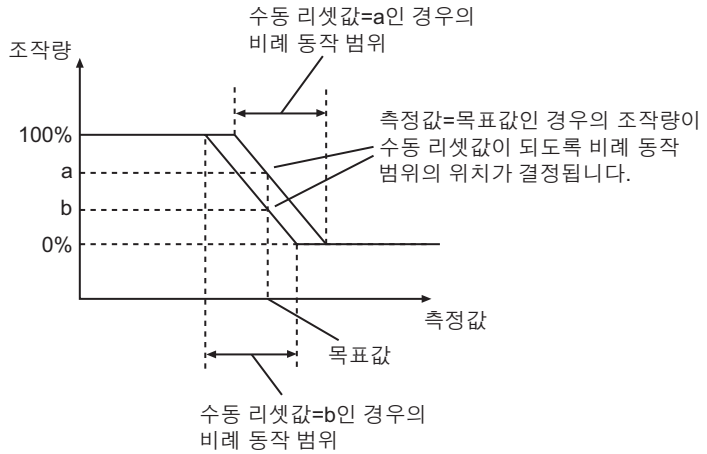
측정값과 목표값의 차(편차)에 비례해서 조작량의 절대값을 크게 하는 제어 방법입니다. 제어 대상의 측정값은 아래 그림과 같이 추이가 변합니다.



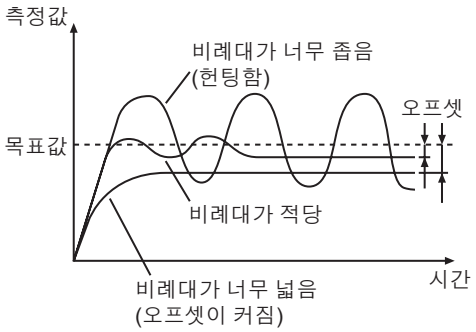
비례 동작의 설정값 중 하나로 비례대가 있습니다.
비례대란 비례 동작을 수행하는 측정값의 폭을 말합니다. 측정값이 비례 동작 범위를 벗어난 경우, 조작량에는 100% 또는 0%를 설정합니다.
비례대는 입력 범위 중에서 비례 동작을 수행하는 범위의 비율(%FS)로 표현합니다. 비례대의 값이 10%FS인 경우의 그림이 아래에 나와 있습니다.



비례 동작의 또 하나의 설정값이 수동 리셋값입니다.
수동 리셋값이란 편차가 0인 경우의 조작량을 말합니다. 수동 리셋값에 따라 측정값과 조작량의 그래프에서 비례 동작 범위의 위치가 결정됩니다.
수동 리셋값과 비례 동작 범위의 관계가 아래에 나와 있습니다.
측정값과 목표값이 일치했을 때의 조작량이 수동 리셋값과 일치하도록 비례 동작 범위의 위치가 결정됩니다.



수동 리셋값이 적정하지 않으면 편차를 완전히 0으로 할 수 없습니다. 이렇게 남은 편차를 오프셋(잔류 편차)이라고 합니다. 비례대를 좁게 하면 오프셋을 작게 할 수 있습니다. 하지만 너무 좁으면 측정값이 목표값에서 멈추지 않고 지나치는 현상이 발생합니다. 이를 오버슈트라고 합니다. 또한 측정값이 안정되지 않고 목표값 부근에서 진동하는 현상을 헌팅이라고 합니다.

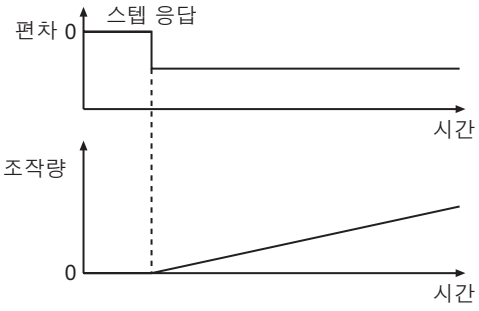


● 적분 동작(I)

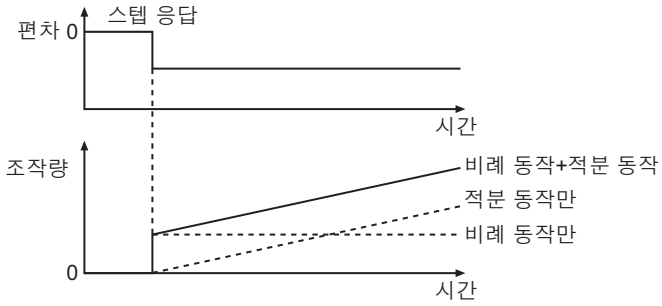
비례 동작만으로 오프셋을 0으로 하려면, 비례대와 수동 리셋값의 세밀한 조정이 필요합니다. 또한 오프셋의 크기가 외란에 의해 변화하는 경우, 그러한 조정을 빈번하게 해야 합니다. 그런 번거로움을 덜기 위해 비례 동작과 함께 사용하는 것이 적분 동작입니다. 적분 동작은 편차를 시간축에서 누적(적분)하여 그 값에 비례해 조작량의 절대값을 크게 하는 제어 방법입니다.
적분 동작을 수행하는 경우, 수동 리셋값의 설정은 무효가 됩니다.

스텝상 편차가 발생한 경우의 적분 동작에 의한 조작량 추이가 아래의 좌측 그림에 나와 있습니다. 또한 비례 동작과 적분 동작을 함께 사용한 경우의 조작량 추이가 아래의 우측 그림에 나와 있습니다.

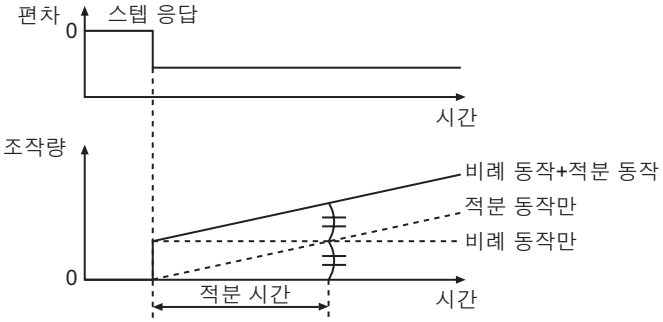
적분 동작에 의한 조작량



비례 동작과 적분 동작의 병용에 의한 조작량



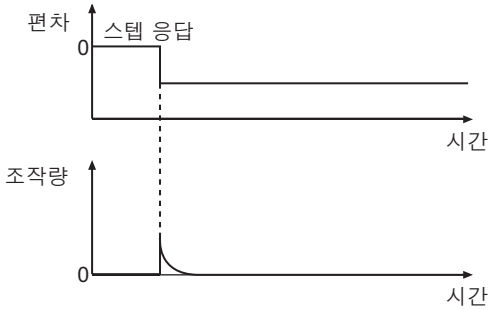
적분 동작의 설정값에 적분 시간이 있습니다. 이는 스텝상 편차에 대해 적분 동작에 의한 조작량이 비례 동작에 의한 조작량과 같아질 때까지의 시간을 말합니다. 적분 시간이 짧을수록 적분 동작이 강해집니다. 적분 시간이 짧으면 그만큼 오프셋을 0으로 할 때까지의 시간이 짧아지는 반면, 헌팅 발생의 요인이 됩니다.



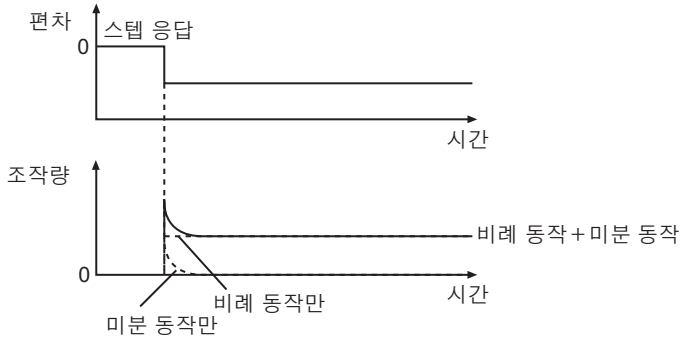
● 미분 동작(D)

비례 동작과 적분 동작을 함께 사용하면, 오프셋을 0으로 해서 측정값을 목표값에 일치시킬 수 있습니다. 하지만 외란이 발생하여 측정값이 갑자기 변화한 경우, 그것을 원래대로 되돌리려면 어느 정도의 시간이 필요합니다. 외란이 발생하더라도 측정값을 단시간에 목표값으로 되돌릴 목적으로 사용하는 것이 미분 동작입니다. 미분 동작은 편차를 시간축에 대해 미분하여 그 크기에 비례해 조작량의 절대값을 크게 하는 제어 방법입니다. 즉, 측정값의 변화가 클수록 미분 동작에 의한 조작량의 절대값이 커집니다.
스텝상 편차가 발생한 경우의 미분 동작에 의한 조작량 추이가 아래에 나와 있습니다. 또한 비례 동작과 미분 동작을 함께 사용한 경우의 조작량 추이도 나와 있습니다.

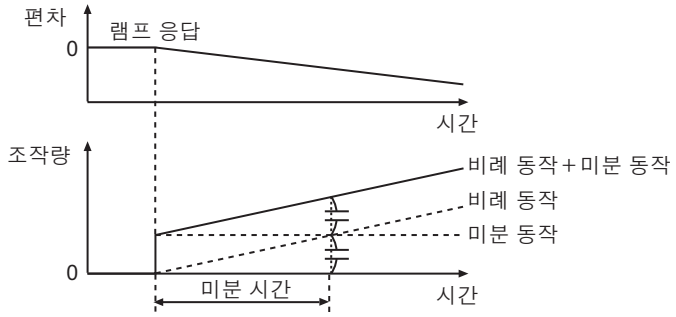
미분 제어에 의한 조작량



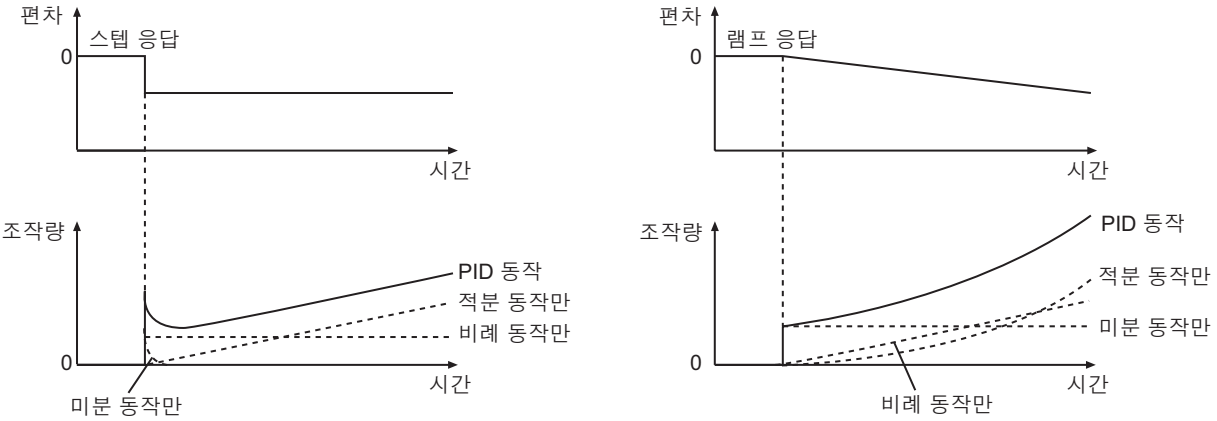
비례 동작과 미분 동작의 병용에 의한 조작량



미분 동작의 설정값에 적분 시간이 있습니다. 이는 램프상 편차에 대해 미분 동작에 의한 조작량이 비례 동작에 의한 조작량과 같아질 때까지의 시간을 말합니다. 미분 시간이 길수록 미분 동작이 강해집니다. 미분 시간이 길면 그만큼 외란에 대한 대응은 빨라지는 반면, 헌팅 발생의 요인이 됩니다.



비례 동작, 적분 동작, 미분 동작의 모든 조작량을 가산한 것이 PID 동작의 조작량이 됩니다. 스텝상 편차, 램프상 편차에 대한 PID 동작의 조작량 추이가 아래에 나와 있습니다.



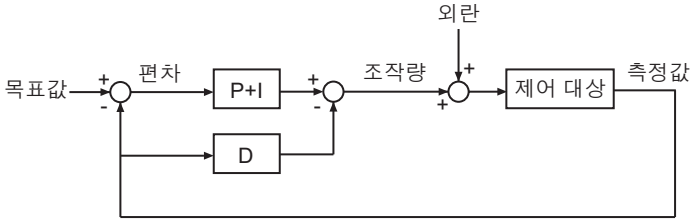
목표값 필터 형식 2자유도 PID 제어

PID 제어에서 조정이 필요한 주요한 설정값은 비례대, 적분 시간, 미분 시간의 3가지입니다. 이를 아울러서 PID 정수라고 합니다.

PID 정수의 값은 다음 2가지의 PID 제어 성능에 영향을 줍니다.

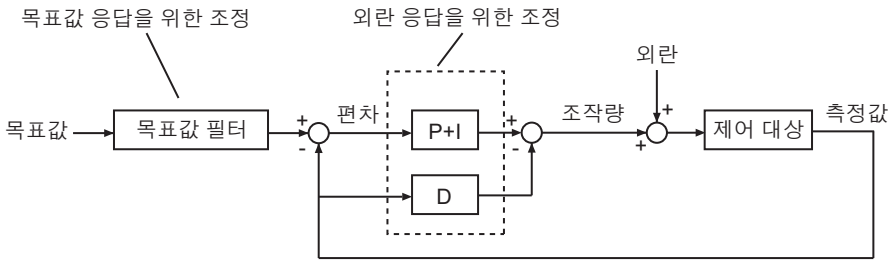
- 목표값 응답 성능: 목표값이 변화한 경우, 그것을 추종하는 성능
- 외란 응답 성능: 외란이 발생하여 측정값이 크게 변화한 경우 그것을 수정하는 성능

아래에 기본 PID 제어 블록도가 나와 있습니다. 목표값과 외란은 입력되는 위치가 다르기 때문에 목표값 응답 성능과 외란 응답 성능에 모두 최적인 PID 정수를 찾는 것은 어렵습니다. 즉, PID 정수를 설정할 때 목표값 응답 성능을 중시하면 외란에 대한 대응이 늦어지고 외란 응답 성능을 중시하면 오버슈트가 발생하게 됩니다.

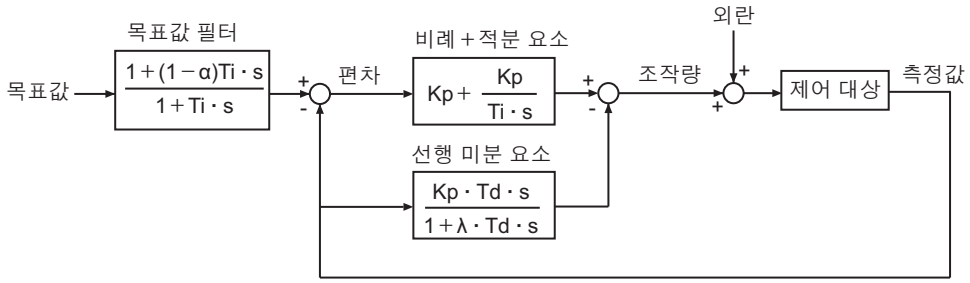


목표값 응답 성능과 외란 응답 성능을 양립시키는 것이 2자유도 PID 제어입니다. 2자유도란 목표값 응답 성능과 외란 응답 성능을 조정하는 파라메터 2종류를 독립적으로 갖는다는 의미입니다. 블록도는 다음과 같습니다. 새롭게 목표값 필터를 설정하고, 거기에 또 하나의 조정 파라메터를 갖습니다.

PID 정수는 외란 응답 성능이 최대가 되도록 조정합니다. 그리고 그 설정이 목표값 응답에 대해서도 최적이 되도록 목표값 필터를 사용하여 목표값을 가공합니다. PID 정수의 값과 목표값 필터의 설정값은 독립적으로 조정 가능하므로 목표값 응답 성능과 외란 응답 성능을 모두 향상시킬 수 있습니다.



이 명령의 각 블록의 연산식이 아래에 나와 있습니다. 목표값 필터의 값(목표값에 대한 계수)은 적분 시간과 2자유도 PID 파라메터 α로 조정합니다. α의 값은 0.65가 최적이며, 일반적으로 변경하지 않습니다. α의 값이 작을수록 목표값 필터의 영향은 작아집니다.



Kp : 비례 정수
Ti : 적분 시간
Td : 미분 시간
S : 라플라스 연산자
α : 2자유도 PID 파라메터
λ : 불완전 미분 계수

운전 시작 순서

이 명령은 적정한 PID 정수를 사용하여 운전해야 합니다. 이를 위한 운전 시작 방법에는 다음 2가지가 있습니다.

● PID 정수의 적정값을 모르는 경우

운전 시작 시에 자동 조정을 수행하여 PID 정수의 적정값을 구합니다.
'StartAT'의 값을 TRUE로 한 상태에서 'Run'의 값을 FALSE에서 TRUE로 변화시킵니다.
처음 자동 조정을 실행해서 구한 PID 정수를 사용하여 PID 연산을 시작합니다.

● PID 정수의 적정값을 아는 경우

PID 정수의 적정값을 'ProportionalBand', 'IntegrationTime', 'DerivativeTime'에 설정하고 'Run'의 값을 FALSE에서 TRUE로 변화시킵니다.
'ProportionalBand', 'IntergrationTime', 'DerivativeTime'은 입출력 변수입니다. 정수는 입력 파라메터로 사용할 수 없습니다. 반드시 적당한 변수를 정의하여 거기에 값을 대입한 것을 입력 파라메터로 사용하십시오.

PID 정수는 운전 중에 변경할 수 있습니다. 또한 운전 중에 자동 조정을 실행하는 것도 가능합니다. 그 경우 'StartAT'의 값을 TRUE로 합니다.

제어 상태와 조작량

조작량 'MV'는 제어 상태에 따라 다음과 같이 결정됩니다.

<table>
<tr><th rowspan="2">제어 상태</th><th colspan="5">변수의 값</th><th rowspan="2">조작량 'MV'</th></tr>
<tr><th>매뉴얼/자동 전환 'ManCtl'</th><th>실행 조건 'Run'</th><th>이상 종료 'Error'</th><th>수동 추적 스위치 'Man TrackSw'</th><th>자동 조정 실행 중 'ATBusy'</th></tr>
<tr><td>이상 종료</td><td rowspan="5">FALSE</td><td rowspan="4">TRUE</td><td>TRUE</td><td>–</td><td rowspan="2">FALSE</td><td>이상 시 조작량 'ErrorMV'</td></tr>
<tr><td>자동 운전 중

MV 추적</td><td rowspan="3">FALSE</td><td>TRUE</td><td>MV 추적값 'MVTrackVal'</td></tr>
<tr><td>자동 운전 중

자동 조정 실행 중</td><td rowspan="2">FALSE</td><td>TRUE</td><td>조작량 리미트 상한값과 조작량 리미트 하한값을 반복</td></tr>
<tr><td>자동 운전 중

자동 조정 비실행</td><td rowspan="3">FALSE</td><td>그 시점에서의 PID 정수에 따라 산출된 값</td></tr>
<tr><td>명령 실행 정지</td><td>FALSE</td><td rowspan="2">–</td><td rowspan="2">–</td><td>정지 시 조작량 'StopMV'</td></tr>
<tr><td>매뉴얼 운전 중</td><td>TRUE</td><td>–</td><td>수동 조작량 'ManMV'</td></tr>
</table>

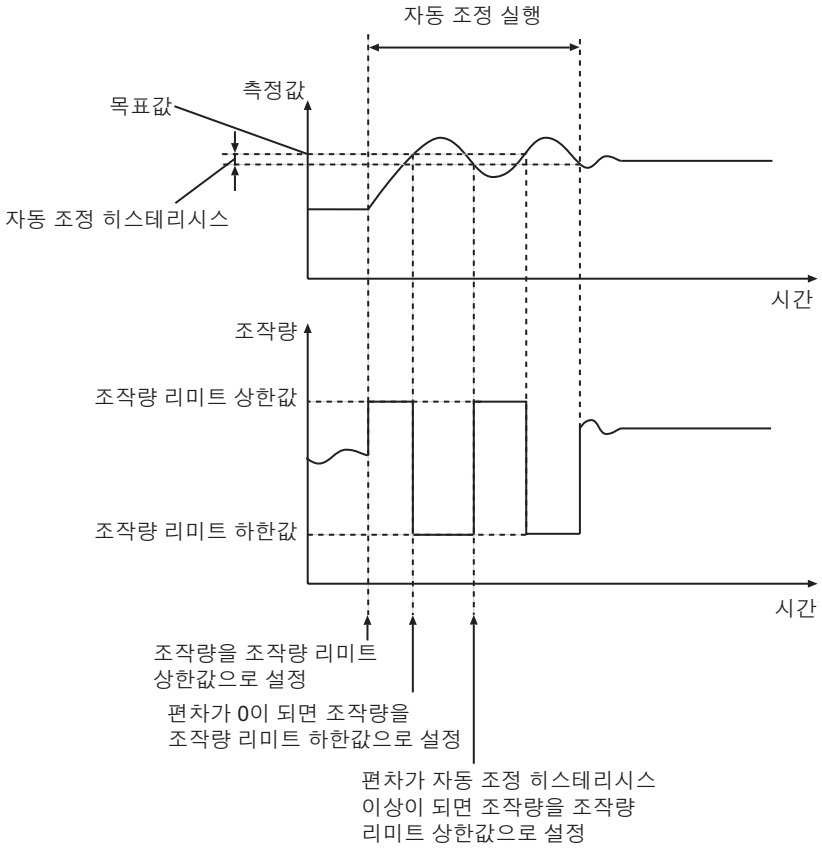
자동 조정

2자유도 PID 파라메터 α를 조정하는 경우는 별로 없으므로, 이 명령에서 주로 조정하는 설정값은 PID 정수가 됩니다.

이 명령에는 PID 정수를 자동 조정하는 기능이 있습니다. 자동 조정 방법으로는 리미트 사이클법을 사용합니다. 리미트 사이클법이란 강제로 조작량을 일시적으로 조작량 리미트 상한값 및 조작량 리미트 하한값으로 만들어서, 그 때의 측정값의 변화에 따라 PID 정수의 최적값을 구하는 방법입니다.

목표값>측정값인 경우에 자동 조정 실행 지시가 있으면 조작량을 우선 조작량 리미트 상한값으로 설정합니다. 그 후 편차가 0이 되면 조작량을 조작량 리미트 하한값으로 설정합니다. 그리고 편차가 자동 조정 히스테리시스의 값 이상이 되면 조작량을 다시 조작량 리미트 상한값으로 설정합니다. 이를 2회 반복하여 PID 정수의 최적값을 산출합니다.

목표값<측정값인 경우에 자동 조정 실행 지시가 있으면 조작량을 우선 조작량 리미트 하한값으로 설정합니다. 그 후에는 위에 기술한 것과 동일한 순서로 PID 정수의 최적값을 산출합니다.



자동 조정은 'Run'의 값이 TRUE이고 PID 연산 중에 'StartAT'의 값이 FALSE→TRUE가 되면 운전 중에 실행됩니다. 또는 'StartAT'의 값이 TRUE인 상태에서 'Run'의 값이 FALSE→TRUE가 되면 운전 시작 시에 실행됩니다.

자동 조정이 정상 종료되면 산출된 PID 정수의 최적값이 즉시 반영됩니다.

자동 조정 중('ATBusy'=TRUE)에 'StartAT'의 값을 FALSE로 하면 자동 조정을 중지합니다. 이 경우, 자동 조정 시작 직전의 PID 정수로 PID 연산을 재개합니다.

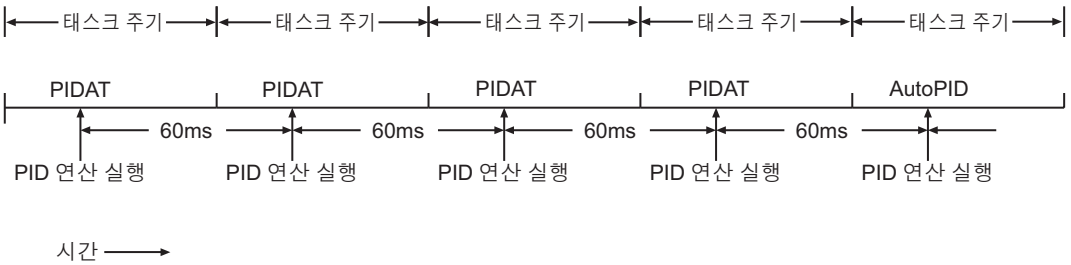
PID 연산의 실행 타이밍

PID 연산은 주기적으로 반복 실행됩니다. PID 연산이 실행되는 것은 사용자 프로그램에서 이 명령이 실행될 때입니다.

단, 이전 PID 연산 실행 이후의 경과 시간이 샘플링 주기 'SampTime' 미만인 경우에는 실행되지 않습니다. 또한 이전 PID 연산 실행 이후의 경과 시간이 'SampTime' 이상인 경우에 실행되며 남은 시간(이전의 실행 이후의 경과 시간-'SampTime')이 다음 회로 넘어갑니다. 자세한 내용은 다음 그림을 참조하십시오.

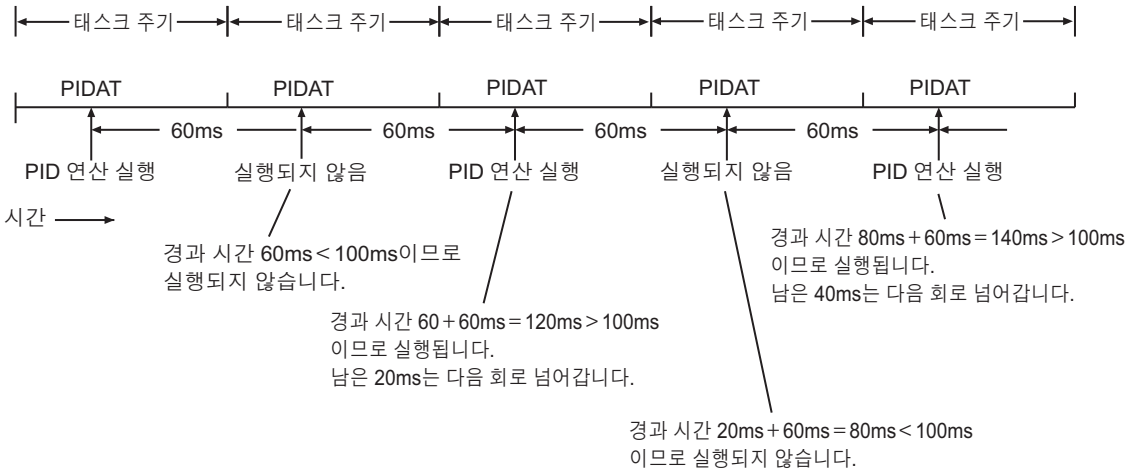
태스크 주기＝60ms　'SampTime'<60ms인 경우

태스크 주기≧'SampTime'이므로 반드시 1태스크 주기에 1회 실행됩니다.



태스크 주기＝60ms　'SampTime'=100ms인 경우

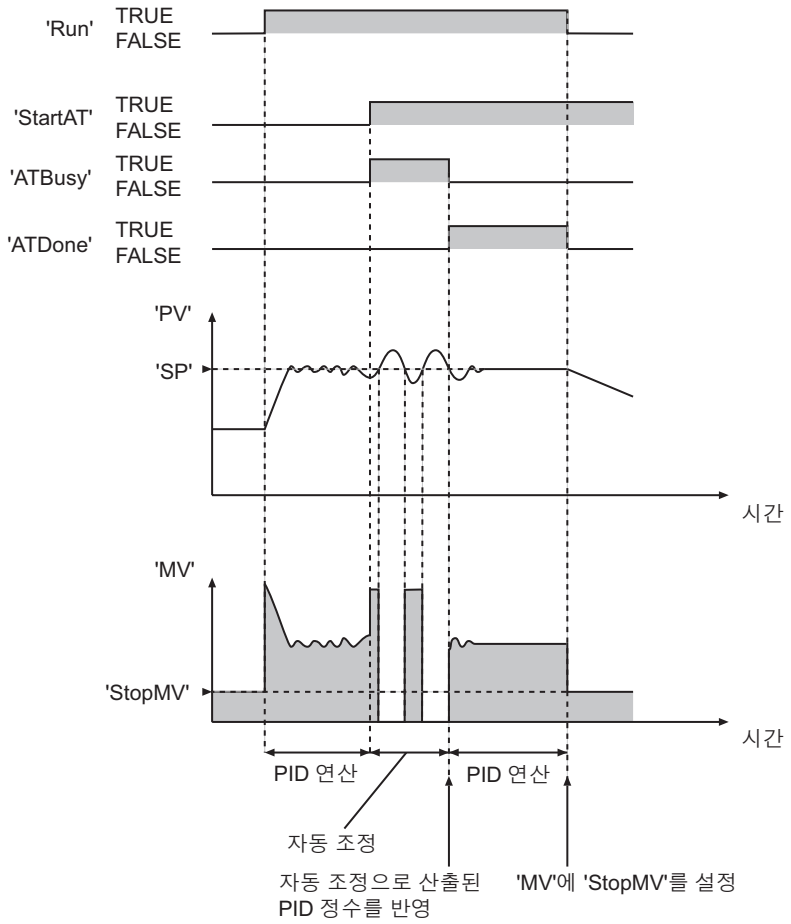
태스크 주기<'SampTime'이므로 실행되지 않는 경우가 있습니다.



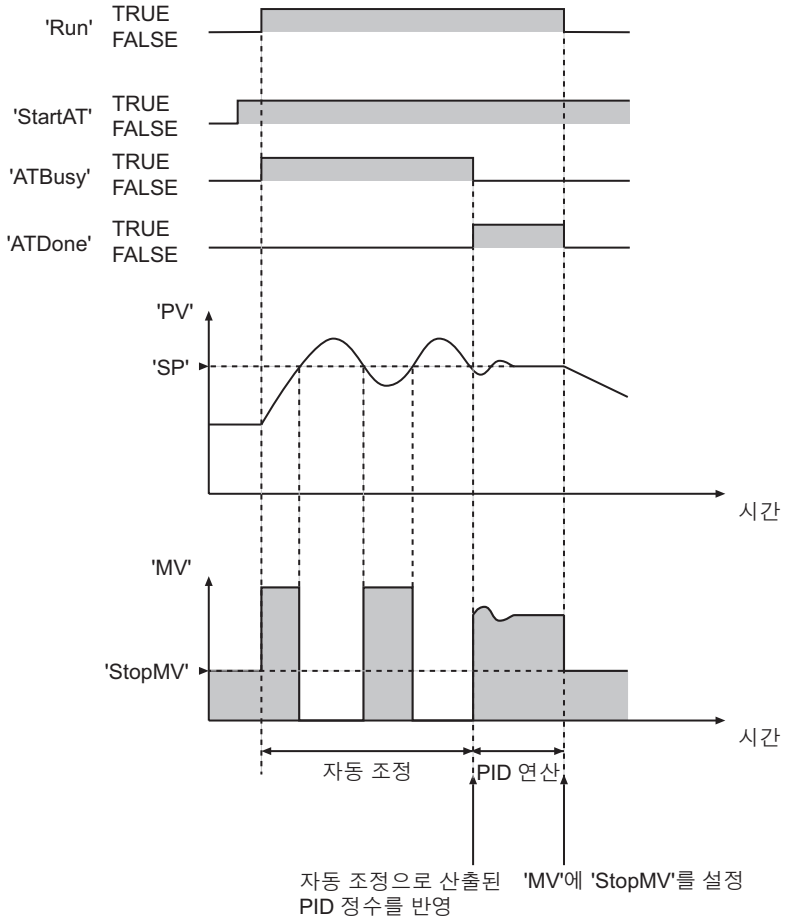
타이밍 차트

몇 가지 경우에 대한 각 변수의 타이밍 차트가 나와 있습니다.

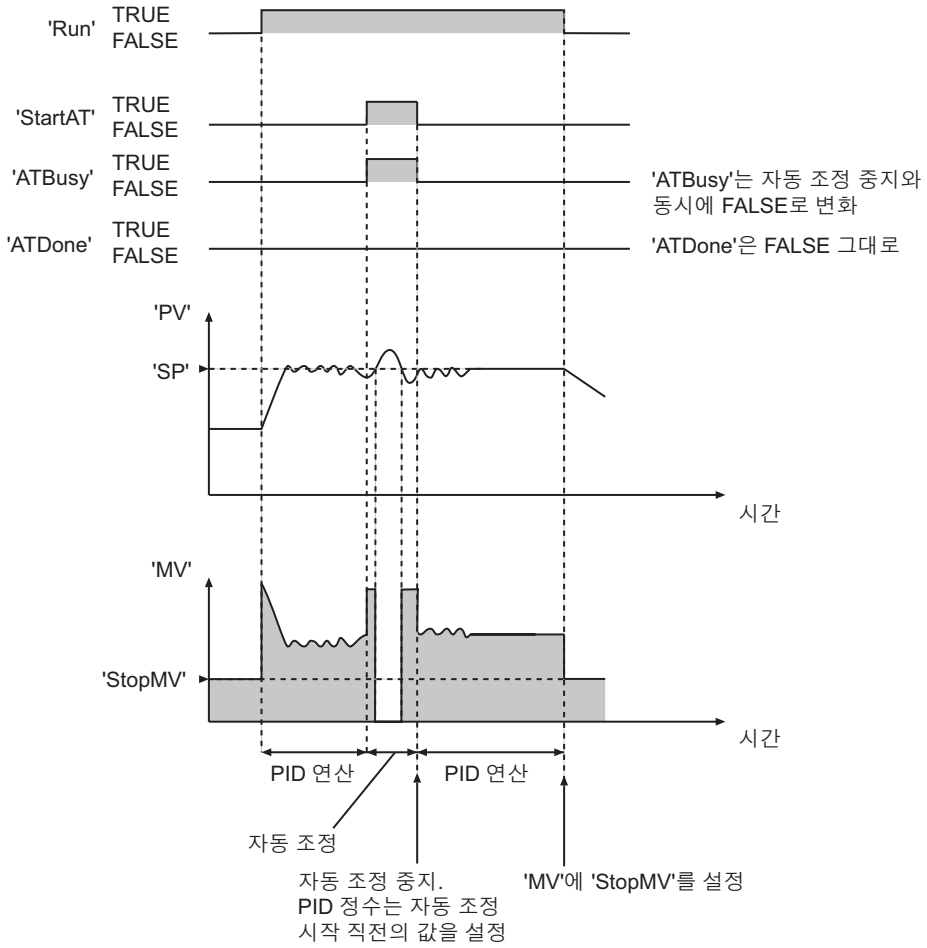
● 자동 운전 중에 자동 조정을 실행한 경우



● 이 명령을 처음 실행할 때 자동 조정을 실행한 경우



● 자동 조정을 중지한 경우

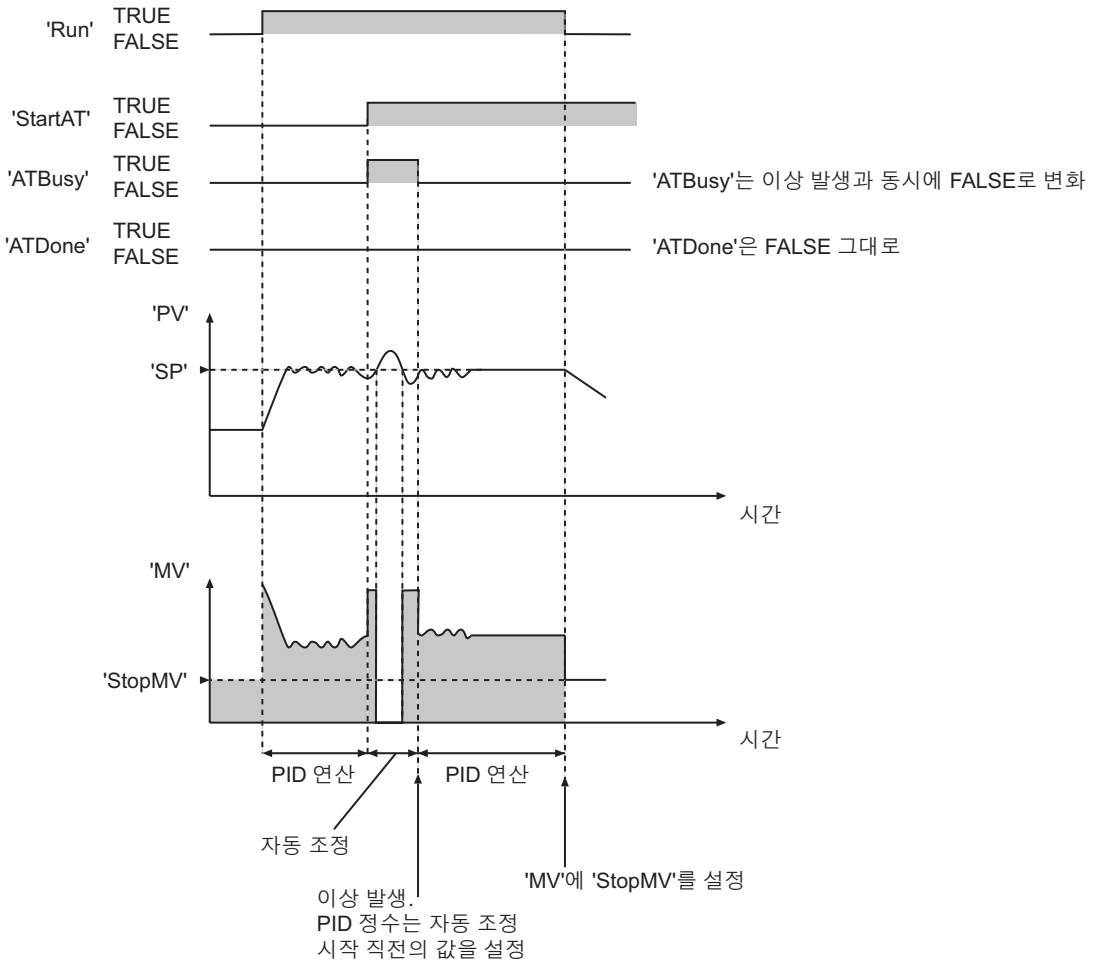

'Run'
TRUE
FALSE
'StartAT'
TRUE
FALSE
'ATBusy'
TRUE
FALSE
'ATBusy'는 자동 조정 중지와
동시에 FALSE로 변화
'ATDone'
TRUE
FALSE
'ATDone'은 FALSE 그대로
'PV'
'SP'
시간
'MV'
'StopMV'
시간
PID 연산
PID 연산
자동 조정
자동 조정 중지.
PID 정수는 자동 조정
시작 직전의 값을 설정
'MV'에 'StopMV'를 설정

● 자동 조정 중에 자동 조정 이상이 발생한 경우

다음과 같은 경우, 자동 조정 이상 상태가 되어 자동 조정을 중지합니다.

- 조작량=조작량 리미트 상한값에서 편차가 0이 될 때까지의 시간이 19999s를 초과함
- 조작량=조작량 리미트 하한값에서 편차가 'ATHystrs' 이상이 될 때까지의 시간이 19999s를 초과함

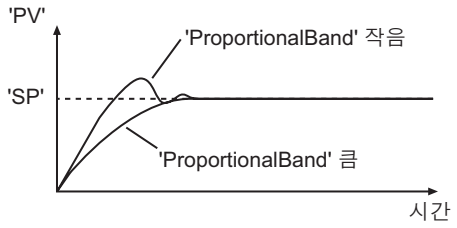
자동 조정을 중지하면 자동 조정 시작 직전의 PID 정수로 PID 연산을 재개합니다.



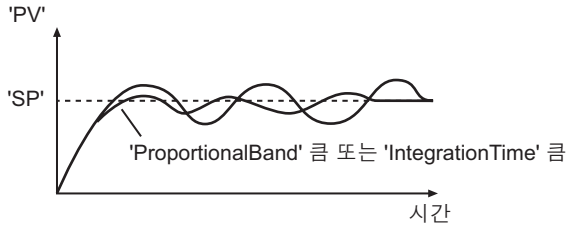
참고

PID 정수의 조정 방법

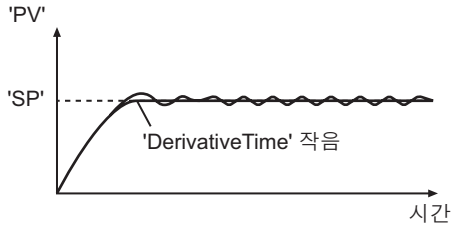
- 제어 대상이 안정될 때까지 다소 시간이 걸려도 헌팅을 발생시키고 싶지 않은 경우는 'ProportionalBand'의 값을 크게 합니다. 반대로, 약간의 헌팅이 발생하더라도 신속하게 안정시키고 싶은 경우에는 'ProportionalBand'의 값을 작게 합니다.



- 헌팅을 반복하는 경우는 'ProportionalBand'를 크게 하거나 'IntegrationTime'을 크게 합니다.



- 헌팅을 짧은 주기로 반복하는 경우, 'DerivativeTime'을 작게 합니다.



온도를 조절하는 경우의 PID 정수 초기값

이 명령을 온도 조절에 사용하는 경우, PID 정수의 초기값으로 다음 값을 참고로 하십시오. PID 정수 이외에는 각 변수의 초기값을 사용하십시오.

변수	참고 초기값(*)
'ProportionalBand'	10%FS
'IntegrationTime'	233s
'DerivativeTime'	40s

* 자동 조정을 실행한 경우, 그 결과에 따르십시오.

사용상의 주의 사항

- 'PV' 및 'SP'의 값은 'RngLowLmt'의 값 이상, 'RngUpLmt'의 값 이하여야 합니다. 이를 위해서는 다음에 나온 것처럼 이러한 변수의 단위계를 맞추십시오.

단위계	'PV', 'SP'의 값	'RngLowLmt', 'RngUpLmt'의 값
%FS	'PV'=(측정값의 물리량-MIN)/(MAX-MIN)×100 'SP'=(목표값의 물리량-MIN)/(MAX-MIN)×100 (*)	'RngLowLmt' =0 'RngUpLmt' =100
물리량	'PV'=측정값의 물리량 'SP'=목표값의 물리량	'RngLowLmt' =MIN 'RngUpLmt' =MAX(*)

* MAX: 입력 범위 상한의 물리량, MIN: 입력 범위 하한의 물리량

- 각 변수의 변경 가능 여부는 제어 상태에 따라 다음과 같습니다.

변수	제어 상태		
	명령 실행 정지 (*1)	운전 중 (자동 조정 비실행) (*2)	운전 중 (자동 조정 실행 중) (*3)
'Run'	○	○	○
'ManCtl'	○	○	○
'StartAT'	○	○	○
'PV'	○	○	○
'SP'	○	○	×
'MvLowLmt'	○	○	×
'MVUpLmt'	○	○	×
'ManResetVal'	○	○	×
'MVTrackSw'	○	○	×
'MVTrackVal'	○	○	×
'StopMV'	○	○	○
'ErrorMV'	○	○	○
'Alpha'	○	○	×
'ATCalcGain'	○	○	×
'ATHystrs'	○	○	×
'SampTime'	○	×	×
'RngLowLmt'	○	×	×
'RngUpLmt'	○	×	×
'DirOpr'	○	×	×
'ProportionalBand'	○	○	×
'IntegrationTime'	○	○	×
'DerivativeTime'	○	○	×
'ManMV'	○	○	○

*1 'ManCtl'=TRUE 또는 'Run'=FALSE 또는 'Error'=TRUE 또는 'MVTrackSw'=TRUE
*2 'ManCtl'=FALSE 및 'Run'=TRUE 및 'Error'=FALSE 및 'MVTrackSw'=FALSE 및 'ATBusy'=FALSE
*3 'ManCtl'=FALSE 및 'Run'=TRUE 및 'Error'=FALSE 및 'MVTrackSw'=FALSE 및 'ATBusy'=TRUE

- 'SampTime'의 경우 100ns 미만은 버려집니다.
- 'ManCtl'의 값이 TRUE일 때 'StartAT'의 값을 TRUE로 한 경우, 다음에 'ManCtl'의 값이 FALSE가 된 시점에서 자동 조정을 시작합니다.
- 'ErrorMV'의 값이 유효 범위 안(-320~320)에 있지 않은 경우, 이상 시의 'MV'의 값은 0이 됩니다.
- 자동 조정 중에 'ManCtl'의 값을 TRUE로 하면 자동 조정을 중지합니다.
- 자동 조정 이상이 발생해도 'Error'의 값은 TRUE가 되지 않습니다.

- 다음과 같은 경우, 이상이 발생합니다. 'Error'는 TRUE가 되고 'ErrorID'에는 이상 ID가 대입됩니다. 'ATDone', 'ATBusy'는 FALSE가 됩니다. 'ManCtl'과 'Run'의 값이 모두 FALSE이면 'MV'에 'ErrorMV'의 값이 설정됩니다. 단, 'ErrorMV'의 값이 유효 범위 밖인 경우, 'MV'의 값은 0이 됩니다.

이상 내용	'ErrorID'의 값
입력 변수 중 하나의 값이 유효 범위 밖	16#0400
'RngLowLmt'≧'RngUpLmt'	16#0401
'MvLowLmt'≧'MVUpLmt'	

- 상기 이외의 조건으로 이상 정지시키려는 경우, 이상 발생 시에 'Run'의 값을 FALSE로 하는 처리를 마련하십시오.
- 'PV' 또는 'SP'의 값이 유효 범위 밖이라 이상이 발생한 후, 값을 유효 범위 내로 변경하더라도 5초 동안은 이상 상태를 유지합니다. 즉, 5초 동안은 'Error'의 값이 FALSE인 상태를 유지합니다.
- 이상 해제 후 'Run'의 값이 TRUE이면 자동으로 PID 연산을 재개합니다. 'Run'과 'StartAT'의 값이 모두 TRUE이면 자동으로 자동 조정을 재개합니다.
- 이상 유무의 판정은 샘플링 주기마다 수행됩니다.

샘플 프로그램

PIDAT 명령을 사용하여 온도 제어를 수행합니다. PID AT 명령의 조작량은 시분할 비례 출력으로 변환하여 히터에 출력합니다. 여기서는 시분할 비례 출력으로의 변환에 타이머 명령을 사용합니다. 시분할 비례 출력으로의 변환에 TimeProportionalOut 명령을 사용하는 경우는 'TimeProportionalOut 명령 (2-670페이지)'의 샘플 프로그램을 참조하십시오.

사양

다음과 같은 사양으로 온도 제어를 수행합니다.

항목	사양
입력 종류	K형 열전대
입력 유니트	절연형 고분해능 풀 멀티 입력 유니트 CJ1W-PH41U
출력 유니트	트랜지스터 출력 유니트 CJ1W-OD212
목표 온도	90℃
PID 연산의 샘플링 주기	100ms
출력 제어 주기	1s

구성 · 설정

입력 유니트 CJ1W-PH41U의 설정은 다음과 같습니다.

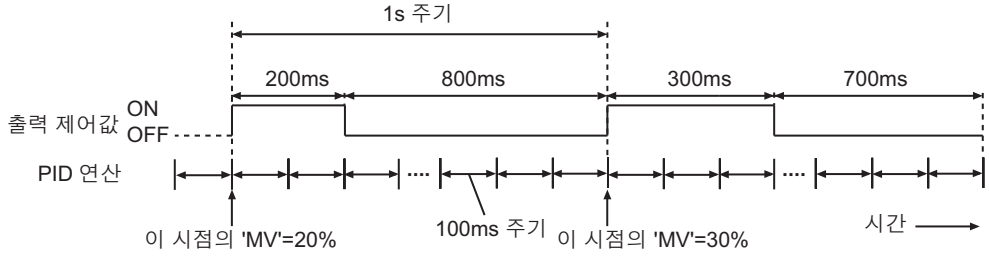
항목명	설정값
Input1:Input signal type	K(1)

I/O 맵 설정은 다음과 같습니다.

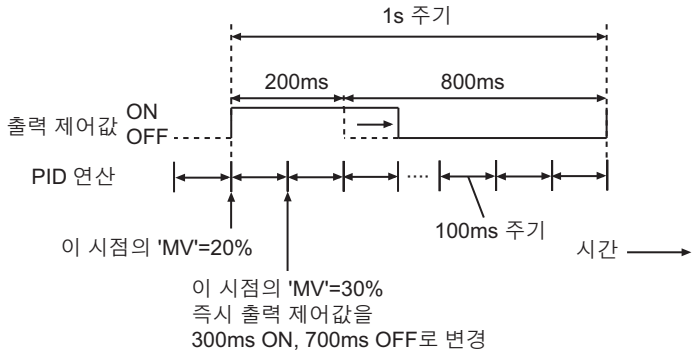
유니트	포트	설명	변수
CJ1W-PH41U	Ch1_AIInPV	입력 번호 1 측정값(INT 형식)	AI1
CJ1W-OD212	Ch1_Out00	출력 CH1 접점 00	DO1

처리 내용

- PIDAT 명령의 조작량 'MV'를 취득하여 온도 조절기로의 출력을 제어합니다. 온도 조절기의 출력은 ON 또는 OFF의 2가지입니다.
- PIDAT 명령의 샘플링 주기 'InitSetParams.SampTime'의 값은 100ms입니다. 태스크 주기는 100ms보다 충분히 짧은 것으로 가정합니다. 따라서 'MV'의 값은 거의 100ms 주기로 갱신됩니다.
- 출력 제어 주기는 1s입니다. 그 사이의 출력 제어값의 ON 시간과 OFF 시간을 시분할 비례 출력으로 제어합니다. 예를 들면, 취득한 'MV'의 값이 20%였을 경우, 온도 조절기로의 출력은 200ms 동안 ON, 이어서 800ms 동안 OFF합니다. 이를 1s 주기로 반복합니다.



- 최신 'MV'의 값이 출력 제어값을 결정했을 때의 'MV'의 값보다 작은 경우, 출력 제어값은 변화하지 않습니다. 최신 'MV'의 값이 출력 제어값을 결정했을 때의 'MV'의 값보다 큰 경우에는 그 값을 즉시 출력 제어값에 반영합니다. 예를 들면 'MV'의 값이 20%일 때 출력 제어값을 결정했다고 가정합니다(ON 200ms, OF F800ms). 그로부터 100ms 후 새로운 'MV'의 값이 30%인 경우, 출력 제어값은 즉시 ON 300ms, OFF 700ms로 변화합니다.



- 자동 조정을 실행하여 'MV'의 값이 100%가 된 경우에는 제어 주기에 관계없이 즉시 출력 제어값을 ON으로 합니다.

응용 프로그램

글로벌 변수의 정의

글로벌 변수

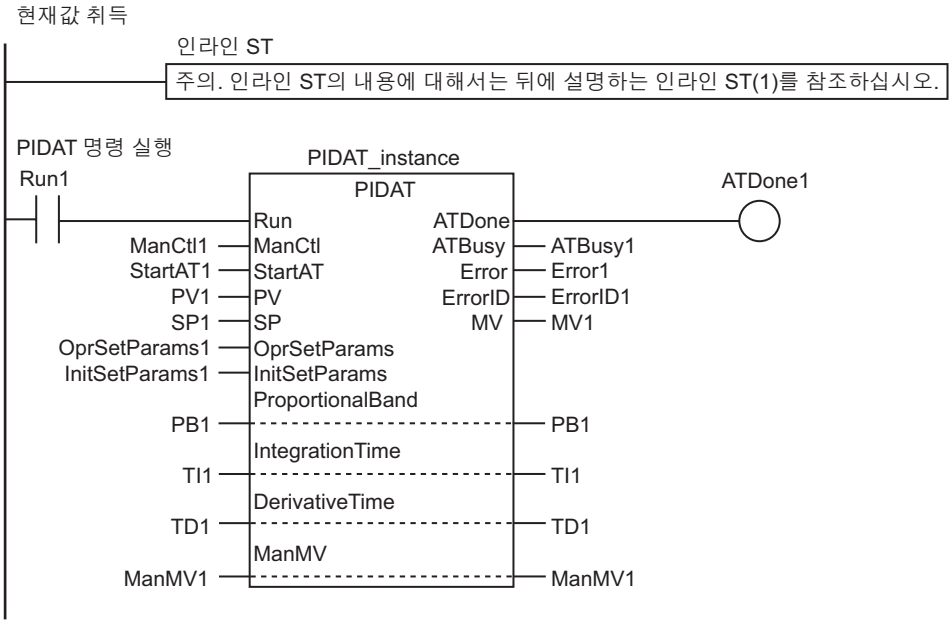
명칭	데이터형식	할당대상*1	코멘트
AI1	INT	IOBus://rack#0/slot#0/Ch1_AIInPV	입력 번호 1 측정값(INT 형식)
DO1	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out00	출력 CH1 접점 00

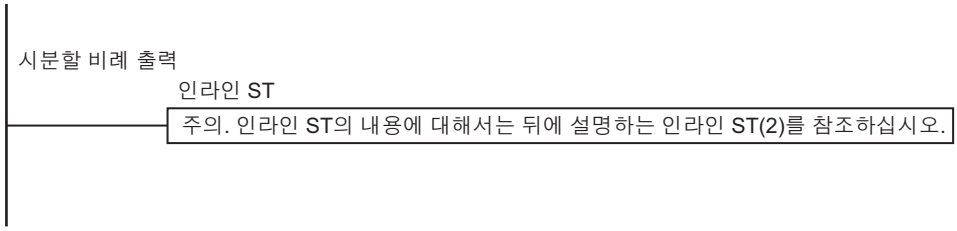
*1. CJ1W-PH41U를 랙 번호 0의 슬롯 번호 0, CJ1W-OD212를 랙 번호 0의 슬롯 번호 1에 장착한 경우의 할당대상입니다.

(주의) I/O 맵 설정에 의해 각 유니트의 포트에 할당된 글로벌 변수가 자동으로 생성됩니다.

LD

명칭	데이터 형식	초기값	유지	코멘트
Run1	BOOL	FALSE	☐	실행 조건
ManCtl1	BOOL	FALSE	☐	매뉴얼/자동 전환
StartAT1	BOOL	FALSE	☐	자동 조정 실행 조건
PV1	REAL	0.0	☐	측정값(현재값)
SP1	REAL	90	☐	목표값
OprSetParams1	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=FALSE, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams1	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:=0.0, RngUpLmt:=1000.0, DirOpr:=FALSE)	☐	초기 설정 파라메터
PB1	REAL	10	☑	비례대
TI1	TIME	T#0S	☑	적분 시간
TD1	TIME	T#0S	☑	미분 시간
ManMV1	REAL	0.0	☐	수동 조작량
ATDone1	BOOL	FALSE	☐	자동 조정 정상 종료
ATBusy1	BOOL	FALSE	☐	자동 조정 실행 중
Error1	BOOL	FALSE	☐	이상
ErrorID1	WORD	16#0	☐	이상 ID
MV1	REAL	0.0	☐	조작량
PulseOnTime	TIME	T#0s	☐	제어 출력의 ON 시간
PulseCycTime	TIME	T#1s	☐	제어 주기
ResetPulse	BOOL	FALSE	☐	타이머 리셋
PIDAT_instance	PIDAT		☐	
TOF_instance	TOF		☐	
TON_instance	TON		☐	





● 인라인 ST(1)의 내용

```
PV1:=INT_TO_REAL(AI1)/REAL#10.0;      // 현재값 AI1을 실수로 변환
                                      // CJ1W-PH41U의 출력은 측정 온도의 10배이므로 10.0으로
                                         제산
```

● 인라인 ST(2)의 내용

```
PulseOnTime:=MULTIME(PulseCycTime, MV1/REAL#100.0);       // 출력 제어값 ON 시간 산출
TOF_instance(In:=BOOL#FALSE, PT:=PulseOnTime, Q=>DO1);    // TOF 명령으로 ON/OFF 전환
TON_instance(In:=BOOL#TRUE, PT:=PulseCycTime, Q=>ResetPulse);   // TON 명령으로 타이머 리셋
                                                                   시간 계측
IF (ResetPulse=BOOL#TRUE) THEN                            // 타이머 리셋
   TOF_instance(In:=BOOL#TRUE);
   TON_instance(In:=BOOL#FALSE);
END_IF;
IF ( (ATBusy1=BOOL#TRUE) & (MV1=REAL#100.0) ) THEN        // 자동 조정일 때 MV1=100%
   DO1:=BOOL#TRUE;                                        // 이 경우, 출력 제어값을 즉시 ON
END_IF;
```

ST

명칭	데이터 형식	초기값	유지	코멘트
Run1	BOOL	FALSE	☐	실행 조건
ManCtl1	BOOL	FALSE	☐	매뉴얼/자동 전환
StartAT1	BOOL	FALSE	☐	자동 조정 실행 조건
PV1	REAL	0.0	☐	측정값(현재값)
SP1	REAL	90	☐	목표값
OprSetParams1	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=FALSE, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams1	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:=0.0, RngUpLmt:=1000.0, DirOpr:=FALSE)	☐	초기 설정 파라메터
PB1	REAL	10	☑	비례대
TI1	TIME	T#0S	☑	적분 시간
TD1	TIME	T#0S	☑	미분 시간
ManMV1	REAL	0.0	☐	수동 조작량
ATDone1	BOOL	FALSE	☐	자동 조정 정상 종료
ATBusy1	BOOL	FALSE	☐	자동 조정 실행 중
Error1	BOOL	FALSE	☐	이상
ErrorID1	WORD	16#0	☐	이상 ID
MV1	REAL	0.0	☐	조작량
PulseOnTime	TIME	T#0S	☐	제어 출력의 ON 시간
PulseCycTime	TIME	T#1S	☐	제어 주기
ResetPulse	BOOL	FALSE	☐	타이머 리셋
PIDAT_instance	PIDAT		☐	
TOF_instance	TOF		☐	
TON_instance	TON		☐	

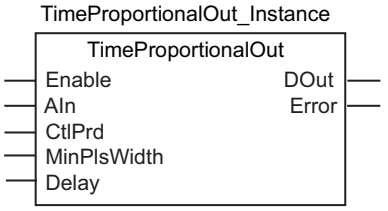
```
// 현재값 AI1을 실수로 변환
PV1:=INT_TO_REAL(AI1)/REAL#10.0;     // CJ1W-PH41U의 출력은 측정 온도의 10배이므로 10.0으로
                                        제산

// PIDAT 명령 실행
PIDAT_instance(
   Run              :=Run1,
   ManCtl           :=ManCtl1,
   StartAT          :=StartAT1,
   PV               :=PV1,
   SP               :=SP1,
   OprSetParams     :=OprSetParams1,
   InitSetParams    :=InitSetParams1,
   ProportionalBand :=PB1,
   IntegrationTime  :=TI1,
   DerivativeTime   :=TD1,
   ManMV            :=ManMV1,
   ATDone           =>ATDone1,
   ATBusy           =>ATBusy1,
   Error            =>Error1,
   ErrorID          =>ErrorID1,
   MV               =>MV1);
```

```
// 시분할 비례 출력
PulseOnTime:=MULTIME(PulseCycTime, MV1/REAL#100.0);              // 출력 제어값 ON 시간 산출
TOF_instance(In:=BOOL#FALSE, PT:=PulseOnTime, Q=>DO1);           // TOF 명령으로 ON/OFF 전환
TON_instance(In:=BOOL#TRUE, PT:=PulseCycTime, Q=>ResetPulse);    // TON 명령으로 타이머 리셋
                                                                    시간 계측
IF (ResetPulse=BOOL#TRUE) THEN                                   // 타이머 리셋
   TOF_instance(In:=BOOL#TRUE);
   TON_instance(In:=BOOL#FALSE);
END_IF;
IF ( (ATBusy1=BOOL#TRUE) & (MV1=REAL#100.0) ) THEN               // 자동 조정일 때 MV1=100%
   DO1:=BOOL#TRUE;                                               // 이 경우, 출력 제어값을 즉시
                                                                    ON
END_IF;
```

TimeProportionalOut

조작량을 시분할 비례 출력으로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TimeProportionalOut	시분할 비례 출력	FB		TimeProportionalOut_instance(Enable, AIn, CtlPrd, MinPlsWidth, Delay, DOut, Error);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	동작 명령	입력	TRUE : 실행 FALSE : 시분할 비례 출력 리셋	데이터 형식에 따름	–	FALSE
AIn	조작량		조작량	0~100	%	0
CtlPrd	제어 주기		시분할 비례 출력의 제어 주기	T#0.1s~T#100s	s	T#2s
MinPlsWidth	최소 펄스 폭		최소 펄스 폭	0~50	%	1
Delay	딜레이		온 딜레이 시간	0~100	%	0
DOut	시분할 비례 출력	출력	TRUE : 시분할 비례 출력 ON FALSE : 시분할 비례 출력 OFF	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
AIn														○						
CtlPrd																○				
MinPlsWidth														○						
Delay														○						
DOut	○																			

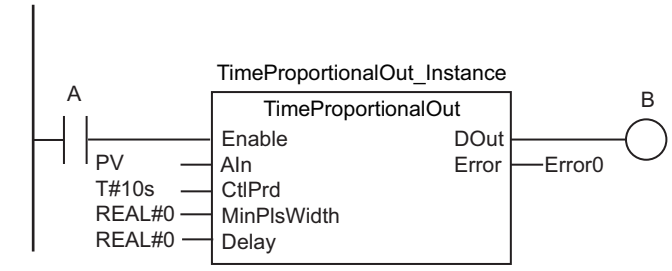
기능

PID 제어 등의 조작량을 시분할 비례 출력으로 변환합니다. 시분할 비례 출력이란 조작량을 ON과 OFF의 시간비에 비례하여 변환한 출력입니다.

동작 명령 'Enable'이 TRUE인 동안 조작량 'AIn'의 값을 제어 주기 'CtlPrd'의 시분할 비례 출력 'DOut'으로 변환합니다. 'Enable'이 FALSE일 때는 시분할 비례 출력을 리셋합니다. 'DOut'과 'Error'는 FALSE가 됩니다. 'CtlPrd', 'MinPlsWidth' 및 'Delay'의 값은 'Enable'이 FALSE에서 TRUE로 변화했을 때 반영됩니다.

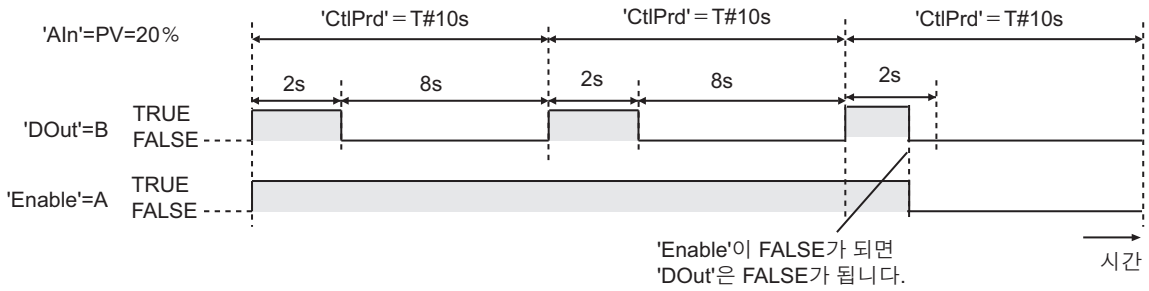
'CtlPrd'의 값이 10s, 'AIn'의 값이 20%인 경우의 기술 예가 아래에 나와 있습니다. 'Enable'이 TRUE인 동안 'DOut'은 2s 동안 TRUE가 되고 이어서 8s 동안 FALSE가 됩니다. 이를 10s 주기로 반복합니다.

LD



ST

```
TimeProportionalOut_instance(A,PV,T#10s,REAL#0,REAL#0,B,Error0);
```



시분할 비례 출력 'DOut'의 분해능

'AIn'의 값을 'DOut'으로 변환하는 최소 단위를 'DOut'의 분해능이라고 합니다.
'DOut'의 분해능보다 'AIn' 값의 분해능이 높으면 'AIn'을 'DOut'의 분해능에 맞춰 반올림한 값을 'DOut'으로 변환합니다.

'DOut'의 분해능은 다음 식으로 산출합니다.

'DOut'의 분해능(%)=태스크 주기 ÷ 'CtlPrd' ×100

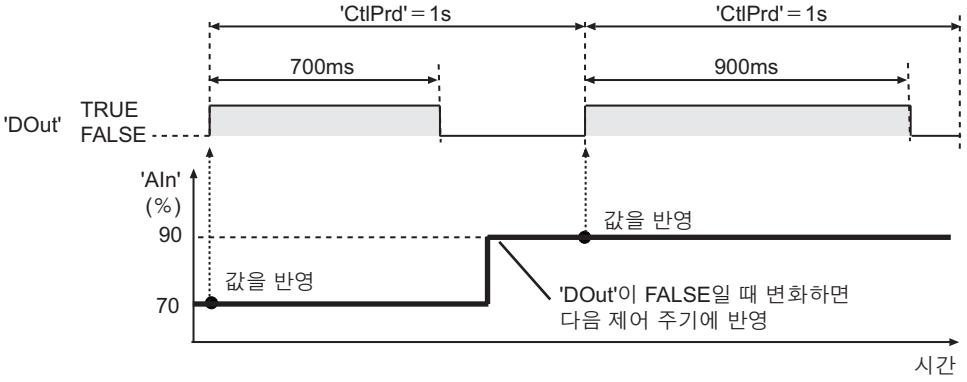
예를 들어, 태스크 주기가 1ms, 'CtlPrd'의 값이 1s인 경우, 'DOut'의 분해능은 0.1%가 됩니다. 이 경우 'AIn' 값의 소수점 이하 2자리 아래는 반올림됩니다.

조작량 'AIn' 값의 반영 타이밍

'AIn' 값의 반영 타이밍은 'DOut'이 FALSE인지, TRUE인지에 따라 달라집니다.

● 'DOut'이 FALSE인 경우

'DOut'이 FALSE인 동안에 'AIn'의 값이 변화한 경우는 다음 제어 주기에서 반영됩니다.

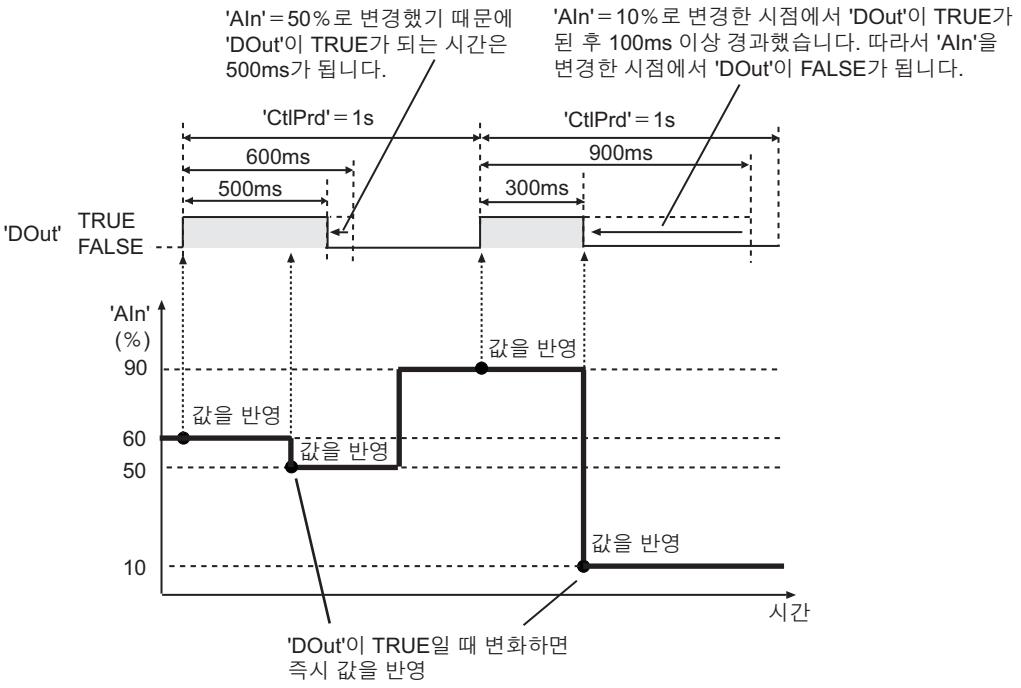


● 'DOut'이 TRUE인 경우

'DOut'이 TRUE인 동안에 'AIn' 값이 변화한 경우는 즉시 반영됩니다.

예를 들어, 제어 주기 'CtlPrd'의 값이 1s인 경우, 다음과 같이 동작합니다.

- 제어 주기를 시작할 때 'AIn'의 값이 60%인 경우, 'DOut'이 TRUE인 동안 'AIn'의 값이 50%로 변화하면 'DOut'은 500ms 동안만 TRUE가 됩니다.
- 제어 주기를 시작할 때 'AIn'의 값이 90%인 경우, 'DOut'이 TRUE가 되고 300ms 경과 후 'AIn'의 값이 10%로 변화한다고 가정합니다. 이 경우, 10%에 해당하는 100ms는 이미 경과했으므로 'DOut'은 즉시 FALSE가 됩니다.



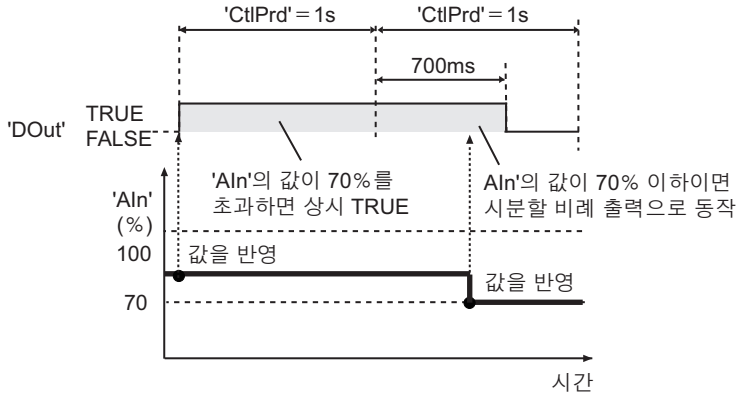
최소 펄스 폭 'MinPlsWidth'에 의한 시분할 비례 출력 'DOut'의 동작

최소 펄스 폭은 'DOut'이 TRUE 또는 FALSE의 값을 유지하는 시간의 최소값입니다. 최소 펄스 폭 'MinPlsWidth'를 설정하면 'DOut'의 채터링을 감소시키는 효과가 있습니다. 예를 들어, 냉각 제어에서는 팬의 ON/OFF를 전환하는 횟수를 줄여서 전력 소비를 억제할 수 있습니다.

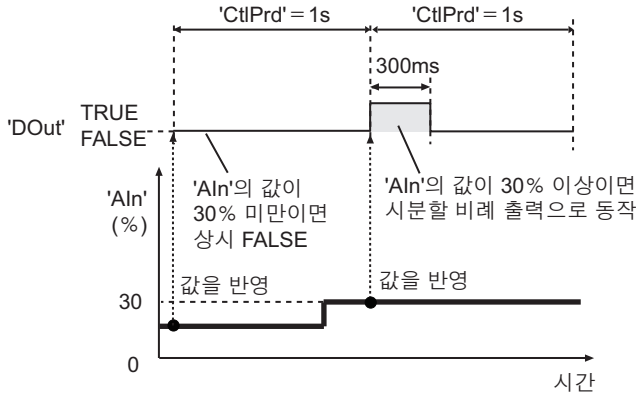
'MinPlsWidth'와 'AIn'의 값의 관계에 따라 'DOut'은 다음과 같이 동작합니다.

'MinPlsWidth'와 'AIn'의 값의 관계	'DOut'의 동작
'AIn' < 'MinPlsWidth'	상시 FALSE
'MinPlsWidth' ≦ 'AIn' ≦ 100 − 'MinPlsWidth'	시분할 비례로 출력
'AIn' > 100 − 'MinPlsWidth'	상시 TRUE

예를 들어, 'MinPlsWidth'가 30%인 경우 'DOut'은 다음 그림과 같이 동작합니다.
'AIn'의 값이 70%를 초과하는 동안은 상시 TRUE, 70% 이하가 되면 시분할 비례 출력이 됩니다.



'AIn'의 값이 30% 미만인 동안은 상시 FALSE, 30% 이상이 되면 시분할 비례 출력이 됩니다.



딜레이 'Delay'에 의한 시분할 비례 출력 'DOut'의 동작

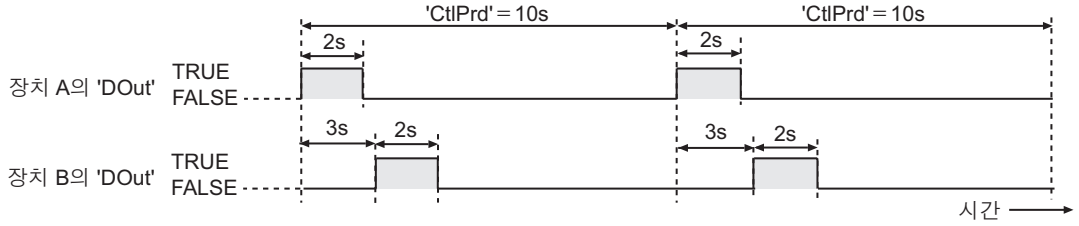
제어 주기의 시작에서 설정 시간이 경과한 후에 'DOut'을 'TRUE'로 하는 것을 딜레이라고 합니다. 이 명령을 여러 개 사용하는 경우, 각각의 'DOut'이 TRUE가 되는 타이밍이 겹치지 않도록 'Delay'의 설정을 통해 조정할 수 있습니다. 따라서 'DOut'이 동시에 ON될 가능성을 줄이는 효과가 있습니다. 예를 들어 복수의 히터를 동작시키는 경우, 'Delay'에서 각각의 히터에 대한 출력이 ON되는 타이밍을 지연시켜서 동시에 사용하는 전력을 줄일 수 있습니다.

제어 주기의 시작에서 딜레이 'Delay' 비율의 시간이 경과한 후 'DOut'이 TRUE가 됩니다.

예를 들어 동일한 제어 주기의 장치 A, 장치 B에 대해 다음 값을 설정한다고 가정합니다.

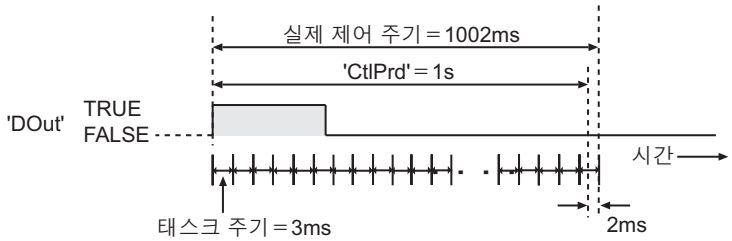
장치	'Delay'의 값	'AIn'의 값	'CtlPrd'의 값
장치 A	0%	20%	10s
장치 B	30%		

장치 A의 'DOut'은 제어 주기를 시작할 때 TRUE가 됩니다. 장치 B의 'DOut'은 제어 주기를 시작하고 3s 후에 TRUE가 됩니다.



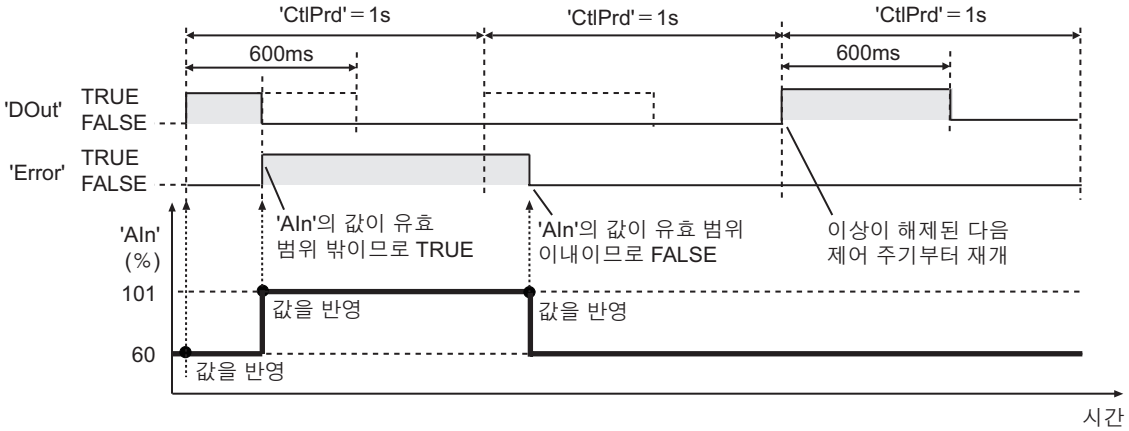
사용상의 주의 사항

- 프로그램을 할당한 태스크 주기의 정수 배를 제어 주기 'CtlPrd' 값으로 설정하십시오. 태스크 주기의 정수 배 이외의 값을 설정한 경우는 제어 주기 'CtlPrd' 종료 후 다음 태스크 실행 타이밍까지가 실제 제어 주기가 됩니다. 예를 들어, 태스크 주기가 3ms, 'CtlPrd'의 값이 1s인 경우는 'CtlPrd' 종료 후 다음 태스크 실행 타이밍까지의 1002ms가 실제 제어 주기가 됩니다.

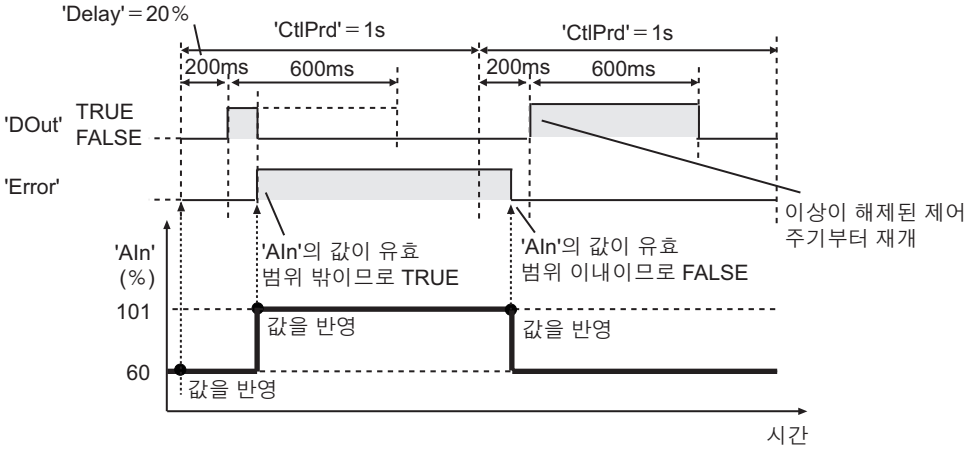


- 'DOut'의 분해능이 0.1% 이하가 되도록 태스크 주기와 제어 주기 'CtlPrd'의 값을 설정하십시오. 'DOut'의 분해능이 0.1%를 초과하면 'DOut'이 TRUE가 되는 비율과 'AIn' 값의 오차가 커져서 제어 성능이 저하됩니다. 예를 들어 'CtlPrd'가 10s인 경우는 태스크 주기를 10ms 이하로 설정하십시오.
- 이 명령을 여러 개 사용하여 각각의 제어 주기를 동기화시키고 싶은 경우에는 이 명령을 동일한 프로그램 내에서 사용하십시오. 다른 프로그램에서 이 명령을 사용하면 프로그램의 실행 타이밍에 따라 제어 주기가 지연됩니다.
- 'AIn', 'CtlPrd', 'MinPlsWidth', 'Delay'의 값이 유효 범위 밖인 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 되고 'DOut'은 FALSE가 됩니다. 'AIn'의 값이 유효 범위 밖이면 이상을 해제하는 타이밍에 따라 'DOut'의 동작이 다음과 같이 달라집니다.

- 'DOut'이 TRUE가 되는 타이밍을 경과하여 이상이 해제된 경우는 다음 제어 주기부터 'DOut'이 시분할 비례 출력을 재개합니다.



- 'DOut'이 TRUE가 되는 타이밍 전에 이상이 해제된 경우에는 이상이 해제된 제어 주기부터 'DOut'이 시분할 비례 출력을 재개합니다.

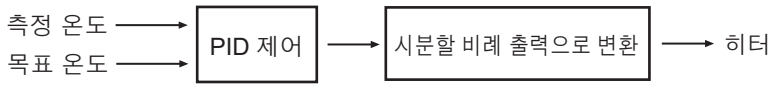


버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

상하한 경보, 상하한 편차 경보가 있는 4점의 온도 제어를 수행합니다. 제어 방식은 PID 제어를 사용하며, PID 제어의 조작량은 시분할 비례 출력으로 변환하여 히터에 출력합니다.



사양

다음과 같은 사양으로 온도 제어를 수행합니다.

항목	사양
입력 유니트	절연형 고분해능 풀 멀티 입력 유니트 CJ1W-PH41U
입력 종류	모두 K형 열전대
출력 유니트	트랜지스터 출력 유니트 CJ1W-OD212
목표 온도	100℃
상한 온도	200℃
하한 온도	0℃
상하한 경보의 히스테리시스	5℃
상한 편차 온도	50℃
하한 편차 온도	50℃
상하한 편차 경보의 히스테리시스	3℃
PID 연산의 샘플링 주기	100ms
출력 제어 주기	1s

구성 · 설정

입력 유니트 CJ1W-PH41U의 설정은 다음과 같습니다.

항목명	설정값
Input1:Input signal type	K(1)
Input2:Input signal type	K(1)
Input3:Input signal type	K(1)
Input4:Input signal type	K(1)

I/O 맵 설정은 다음과 같습니다.

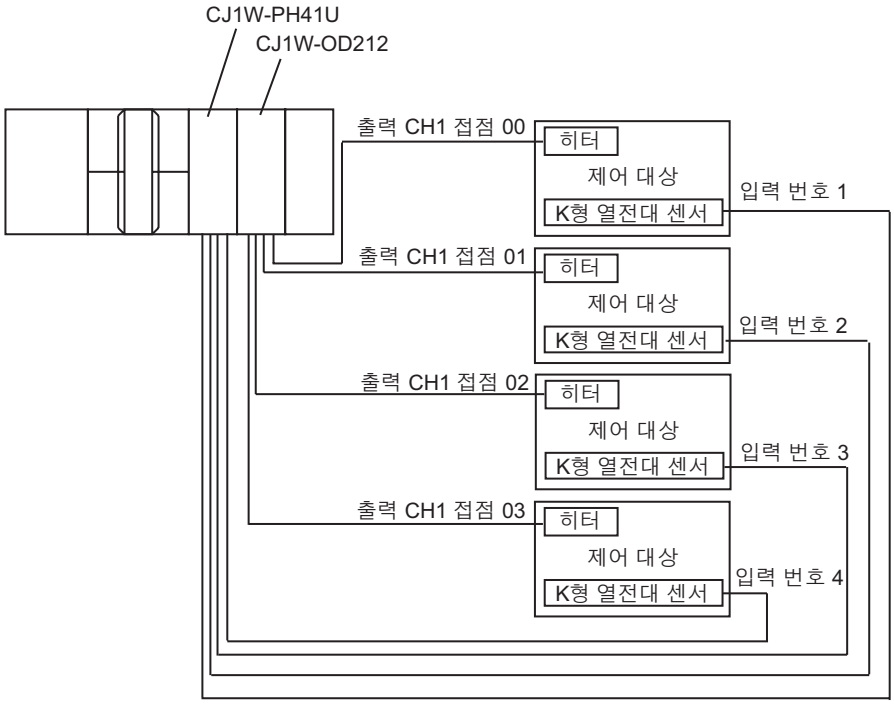
유니트	포트	설명	변수
CJ1W-PH41U	Ch1_AIInPV	입력 번호 1 측정값(INT 형식)	AI1
	Ch2_AIInPV	입력 번호 2 측정값(INT 형식)	AI2
	Ch3_AIInPV	입력 번호 3 측정값(INT 형식)	AI3
	Ch4_AIInPV	입력 번호 4 측정값(INT 형식)	AI4
CJ1W-OD212	Ch1_Out00	출력 CH1 접점 00	DO1
	Ch1_Out01	출력 CH1 접점 01	DO2
	Ch1_Out02	출력 CH1 접점 02	DO3
	Ch1_Out03	출력 CH1 접점 03	DO4

4점의 온도 제어에 대한 입력 및 출력의 대응은 다음과 같습니다.

입력	출력
AI1	DO1
AI2	DO2
AI3	DO3
AI4	DO4

프로그램을 할당하는 태스크의 주기는 1ms로 합니다.

● 구성도

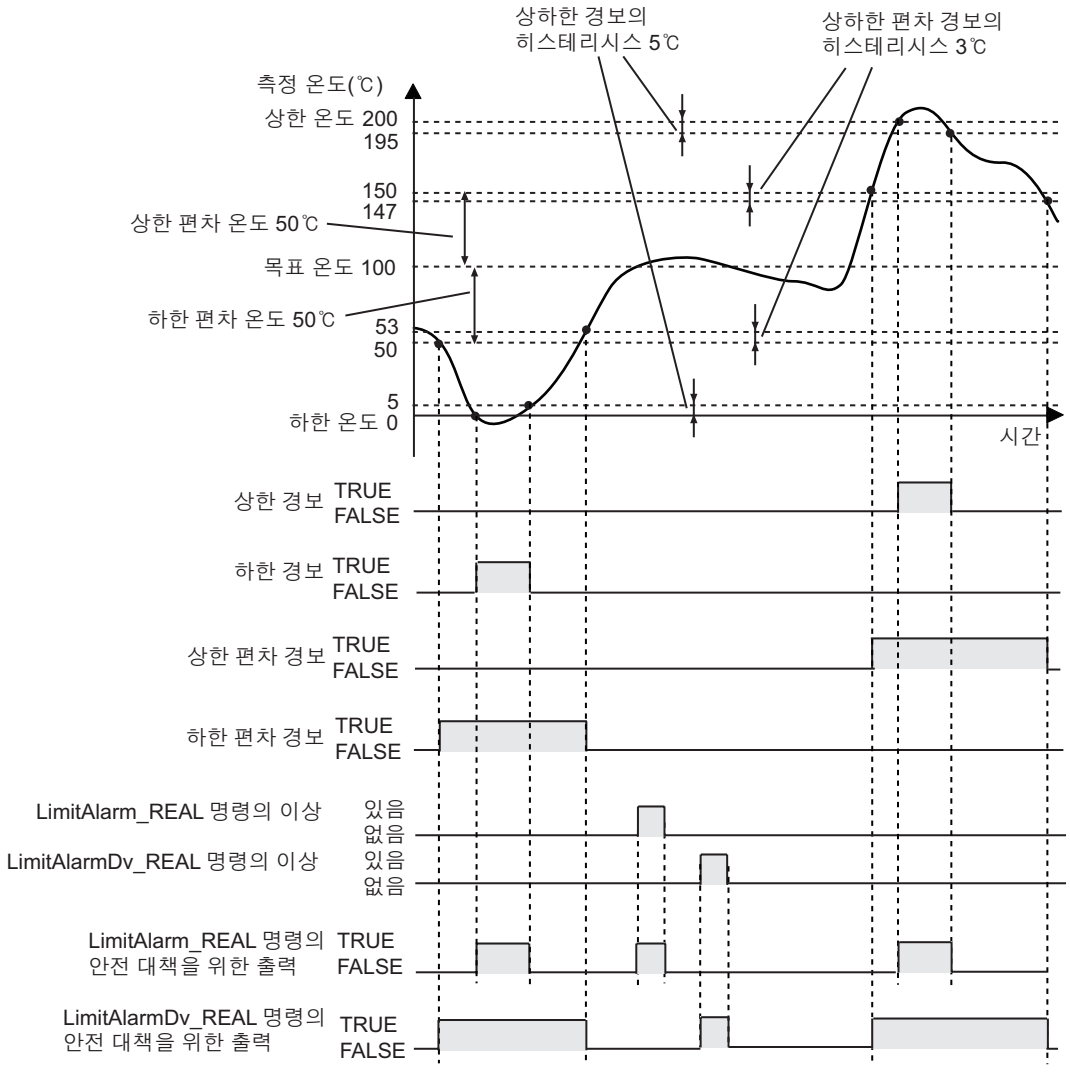


처리 내용

다음 처리를 4점 분량 수행합니다.

1. **측정 온도를 취득합니다.**
2. **LimitAlarm_REAL 명령을 사용하여 측정 온도에 대해 상하한 경보를 출력합니다.**
3. **LimitAlarm_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우는 안전 대책을 위한 출력을 합니다.**
4. **LimitAlarmDv_REAL 명령을 사용하여 목표 온도에서 측정 온도의 편차에 대해 상하한 편차 경보를 출력합니다.**
5. **LimitAlarmDv_REAL 명령에서 이상이 발생한 경우 또는 상하한 편차 경보를 출력한 경우는 안전 대책을 위한 출력을 합니다.**
6. **PIDAT 명령을 사용하여 온도를 제어합니다.**
7. **TimeProportionalOut 명령을 사용하여 조작량을 시분할 비례로 히터에 출력합니다.**

- **상하한 경보, 상하한 편차 경보의 동작**



응용 프로그램

글로벌 변수의 정의

글로벌 변수

명칭	데이터 형식	할당대상*1	코멘트
AI1	INT	IOBus://rack#0/slot#0/Ch1_AIInPV	입력 번호 1 측정값(INT 형식)
AI2	INT	IOBus://rack#0/slot#0/Ch2_AIInPV	입력 번호 2 측정값(INT 형식)
AI3	INT	IOBus://rack#0/slot#0/Ch3_AIInPV	입력 번호 3 측정값(INT 형식)
AI4	INT	IOBus://rack#0/slot#0/Ch4_AIInPV	입력 번호 4 측정값(INT 형식)
DO1	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out00	출력 CH1 접점 00
DO2	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out01	출력 CH1 접점 01
DO3	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out02	출력 CH1 접점 02
DO4	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out03	출력 CH1 접점 03

*1. CJ1W-PH41U를 랙 번호 0의 슬롯 번호 0, CJ1W-OD212를 랙 번호 0의 슬롯 번호 1에 장착한 경우의 할당대상입니다.

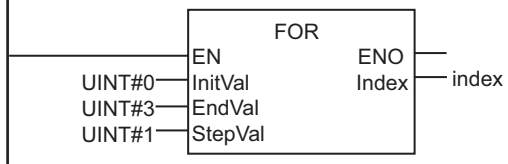
(주의) I/O 맵 설정에 의해 각 유니트의 포트에 할당된 글로벌 변수가 자동으로 생성됩니다.

LD

명칭	데이터 형식	초기값	유지	코멘트
index	UINT	0	☐	루프 인덱스
LimitAlarm_ON	BOOL	True	☐	상하한 경보 실행
LimitAlarmDv_ON	BOOL	True	☐	상하한 편차 경보 실행
TimeProportionalOut_ON	BOOL	True	☐	시분할 비례 출력 실행
AI	INT	0	☐	현재값
PV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	측정값
SP	ARRAY[0..3] OF REAL	[4(100)]	☐	목표 온도
DOut_TPO	BOOL	False	☐	시분할 비례 출력
HighVal	ARRAY[0..3] OF REAL	[4(200)]	☐	상하한 경보의 상한 설정값
LowVal	ARRAY[0..3] OF REAL	[4(0.0)]	☐	상하한 경보의 하한 설정값
Hystrs_LimitAlarm	ARRAY[0..3] OF REAL	[4(5)]	☐	상하한 경보의 히스테리시스
Q_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보 출력
HighAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 경보
LowAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 경보
Error_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarm_REAL 명령의 이상
Alm_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보의 안전 대책을 위한 출력
DvHighVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 상한 편차 설정값
DvLowVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 하한 편차 설정값
Q_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보 출력
HighAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 편차 경보
LowAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 편차 경보

명칭	데이터 형식	초기값	유지	코멘트
Error_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarmDv_REAL 명령의 이상
Hystrs_LimitAlarmDv	ARRAY[0..3] OF REAL	[4(3)]	☐	상하한 편차 경보의 히스테리시스
Alm_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보의 안전 대책을 위한 출력
Run	ARRAY[0..3] OF BOOL	[4(False)]	☐	실행 조건
ManCtl	ARRAY[0..3] OF BOOL	[4(False)]	☐	매뉴얼/자동 전환
StartAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 조건
OprSetParams	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=False, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:=-10.0, RngUpLmt:=1000.0, DirOpr:=False)	☐	초기 설정 파라메터
PB	ARRAY[0..3] OF REAL	[4(10)]	☑	비례대
TI	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	적분 시간
TD	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	미분 시간
ManMV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	수동 조작량
ATDone	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 정상 종료
ATBusy	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 중
Error_PIDAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	PIDAT 명령의 이상
ErrorID	ARRAY[0..3] OF WORD	[4(16#0)]	☐	PIDAT 명령의 이상 ID
MV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	조작량
CtlPrd	ARRAY[0..3] OF TIME	[4(T#1s)]	☐	제어 주기
MinPlsWidth	ARRAY[0..3] OF REAL	[4(0.0)]	☐	최소 펄스 폭
Delay	ARRAY[0..3] OF REAL	[4(0.0)]	☐	온 딜레이 시간
Error_TimeProportionalOut	ARRAY[0..3] OF BOOL	[4(False)]	☐	TimeProportionalOut 명령의 이상
LimitAlarm_REAL_instance	LimitAlarm_REAL		☐	
LimitAlarmDv_REAL_instance	LimitAlarmDv_REAL		☐	
PIDAT_instance	PIDAT		☐	
TimeProportionalOut_instance	TimeProportionalOut		☐	

4점 분량의 온도 제어

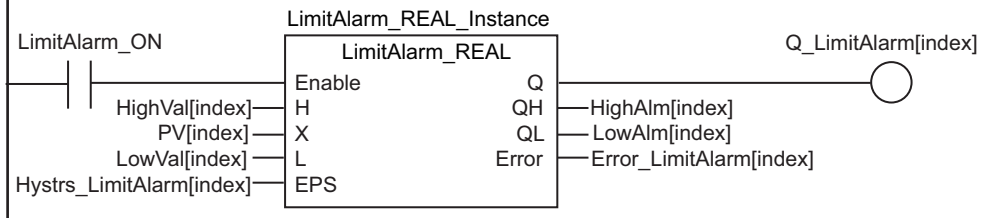


현재값 취득

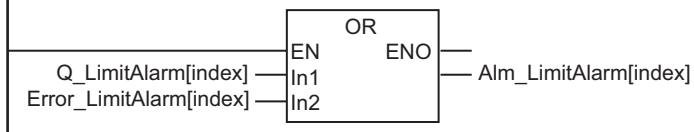
인라인 ST

주의. 인라인 ST의 내용에 대해서는 뒤에 설명하는 인라인 ST(1)를 참조하십시오.

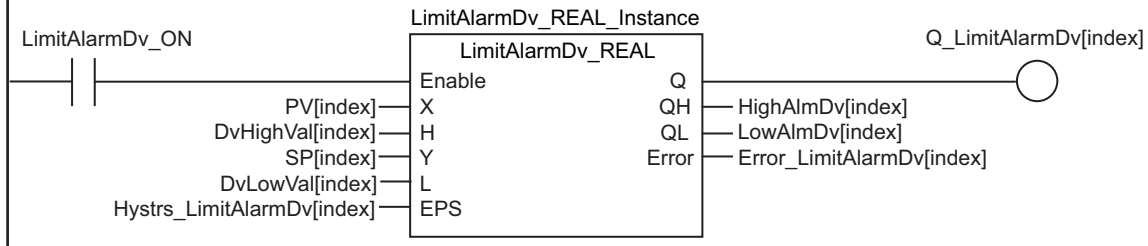
상하한 경보



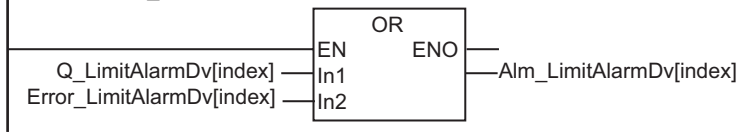
LimitAlarm_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우의 안전 대책을 위한 출력

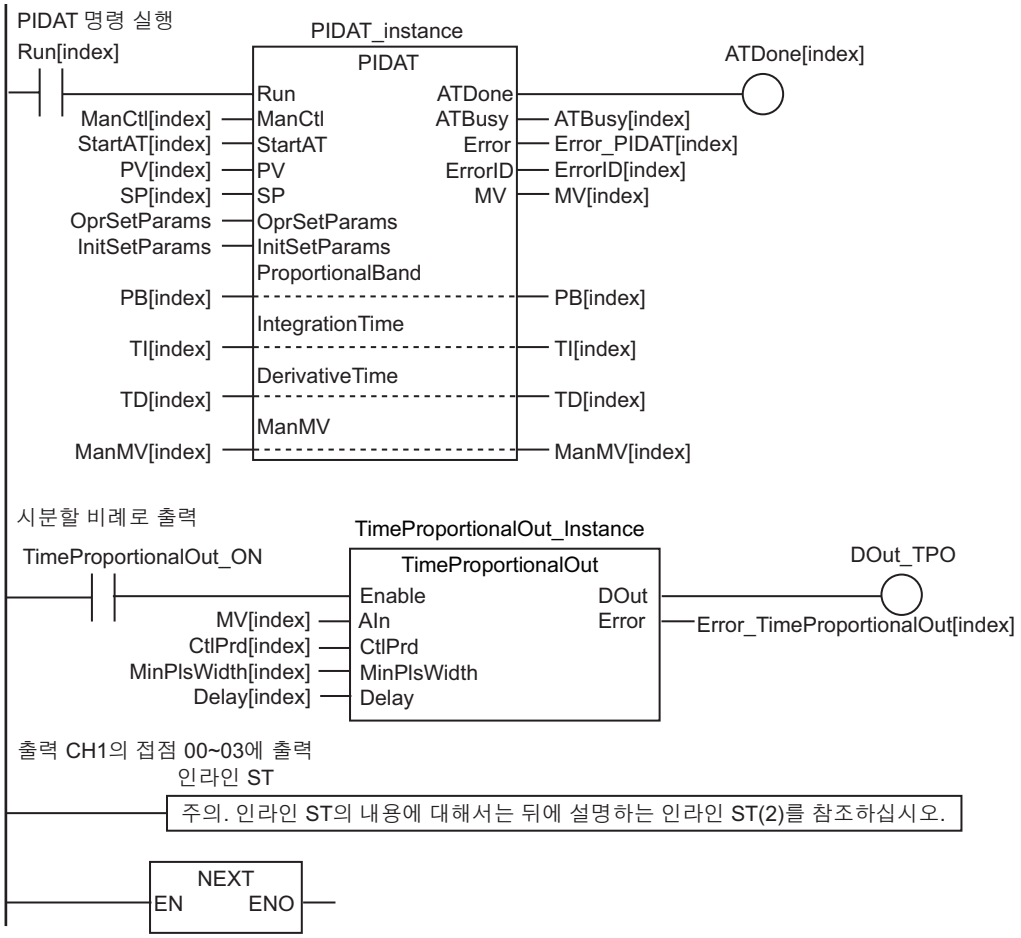


상하한 편차 경보



LimitAlarmDv_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우의 안전 대책을 위한 출력





● 인라인 ST(1)의 내용

```
// 입력 1~4의 값을 취득
CASE index OF
  INT#0:
    AI:=AI1;
  INT#1:
    AI:=AI2;
  INT#2:
    AI:=AI3;
  ELSE
    AI:=AI4;
 END_CASE;

// 현재값 AI를 실수로 변환
PV[index]:=INT_TO_REAL(AI)/REAL#10.0;     // CJ1W-PH41U의 출력은 측정 온도의 10배이므로 10.0
                                             으로 제산
```

● 인라인 ST(2)의 내용

```
// 출력 CH1의 접점 00~03에 출력
CASE index OF
  INT#0:
    DO1:=DOut_TPO;
  INT#1:
    DO2:=DOut_TPO;
  INT#2:
    DO3:=DOut_TPO;
  ELSE
    DO4:=DOut_TPO;
END_CASE;
```

ST

명칭	데이터 형식	초기값	유지	코멘트
index	UINT	0	☐	루프 인덱스
LimitAlarm_ON	BOOL	True	☐	상하한 경보 실행
LimitAlarmDv_ON	BOOL	True	☐	상하한 편차 경보 실행
TimeProportional Out_ON	BOOL	True	☐	시분할 비례 출력 실행
AI	INT	0	☐	현재값
PV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	측정값
SP	ARRAY[0..3] OF REAL	[4(100)]	☐	목표 온도
DOut_TPO	BOOL	False	☐	시분할 비례 출력
HighVal	ARRAY[0..3] OF REAL	[4(200)]	☐	상하한 경보의 상한 설정값
LowVal	ARRAY[0..3] OF REAL	[4(0.0)]	☐	상하한 경보의 하한 설정값
Hystrs_LimitAlarm	ARRAY[0..3] OF REAL	[4(5)]	☐	상하한 경보의 히스테리시스
Q_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보 출력
HighAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 경보
LowAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 경보
Error_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarm_REAL 명령의 이상
Alm_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보의 안전 대책을 위한 출력
DvHighVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 상한 편차 설정값
DvLowVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 하한 편차 설정값
Q_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보 출력
HighAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 편차 경보
LowAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 편차 경보
Error_LimitAlarm Dv	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarmDv_REAL 명령의 이상
Hystrs_LimitAlarm Dv	ARRAY[0..3] OF REAL	[4(3)]	☐	상하한 편차 경보의 히스테리시스
Alm_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보의 안전 대책을 위한 출력
Run	ARRAY[0..3] OF BOOL	[4(False)]	☐	실행 조건
ManCtl	ARRAY[0..3] OF BOOL	[4(False)]	☐	매뉴얼/자동 전환
StartAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 조건
OprSetParams	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=False, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:=-10.0, RngUpLmt:=1000.0, DirOpr:=False)	☐	초기 설정 파라메터
PB	ARRAY[0..3] OF REAL	[4(10)]	☑	비례대
TI	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	적분 시간
TD	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	미분 시간
ManMV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	수동 조작량

명칭	데이터 형식	초기값	유지	코멘트
ATDone	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 정상 종료
ATBusy	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 중
Error_PIDAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	PIDAT 명령의 이상
ErrorID	ARRAY[0..3] OF WORD	[4(16#0)]	☐	PIDAT 명령의 이상 ID
MV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	조작량
CtlPrd	ARRAY[0..3] OF TIME	[4(T#1s)]	☐	제어 주기
MinPlsWidth	ARRAY[0..3] OF REAL	[4(0.0)]	☐	최소 펄스 폭
Delay	ARRAY[0..3] OF REAL	[4(0.0)]	☐	온 딜레이 시간
Error_TimeProportionalOut	ARRAY[0..3] OF BOOL	[4(False)]	☐	TimeProportionalOut 명령의 이상
LimitAlarm_REAL_instance	LimitAlarm_REAL		☐	
LimitAlarmDv_REAL_instance	LimitAlarmDv_REAL		☐	
PIDAT_instance	PIDAT		☐	
TimeProportionalOut_instance	TimeProportionalOut		☐	

```
// 4점 분량의 온도 제어
FOR index:=UINT#0 TO UINT#3 BY UINT#1 DO

  //입력 1~4의 값을 취득
    CASE index OF
      INT#0:
        AI:=AI1;
      INT#1:
        AI:=AI2;
      INT#2:
        AI:=AI3;
      ELSE
        AI:=AI4;
    END_CASE;

  // 현재값 AI를 실수로 변환
  PV[index]:=INT_TO_REAL(AI)/REAL#10.0;   // CJ1W-PH41U의 출력은 측정 온도의 10배이므로 10.0
                                             으로 제산

  // 상하한 경보
  LimitAlarm_REAL_instance(
    Enable :=LimitAlarm_ON,
    H      :=HighVal[index],
    X      :=PV[index],
    L      :=LowVal[index],
    EPS    :=Hystrs_LimitAlarm[index],
    Q      =>Q_LimitAlarm[index],
    QH     =>HighAlm[index],
    QL     =>LowAlm[index],
    Error  =>Error_LimitAlarm[index]);

  // LimitAlarm_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우의 안전 대책을 위한
    출력
  Alm_LimitAlarm[index]:=Q_LimitAlarm[index] OR Error_LimitAlarm[index];

  // 상하한 편차 경보
```

```
LimitAlarmDv_REAL_instance(
  Enable :=LimitAlarmDv_ON,
  X      :=PV[index],
  H      :=DvHighVal[index],
  Y      :=SP[index],
  L      :=DvLowVal[index],
  EPS    :=Hystrs_LimitAlarmDv[index],
  Q      =>Q_LimitAlarmDv[index],
  QH     =>HighAlmDv[index],
  QL     =>LowAlmDv[index],
  Error  =>Error_LimitAlarmDv[index]);

// LimitAlarmDv_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우의 안전 대책을
  위한 출력
Alm_LimitAlarmDv[index]:=Q_LimitAlarmDv[index] OR Error_LimitAlarmDv[index];

// PIDAT 명령 실행
PIDAT_instance(
  Run                :=Run[index],
  ManCtl             :=ManCtl[index],
  StartAT            :=StartAT[index],
  PV                 :=PV[index],
  SP                 :=SP[index],
  OprSetParams       :=OprSetParams,
  InitSetParams      :=InitSetParams,
  ProportionalBand   :=PB[index],
  IntegrationTime    :=TI[index],
  DerivativeTime     :=TD[index],
  ManMV              :=ManMV[index],
  ATDone             =>ATDone[index],
  ATBusy             =>ATBusy[index],
  Error              =>Error_PIDAT[index],
  ErrorID            =>ErrorID[index],
  MV                 =>MV[index]);

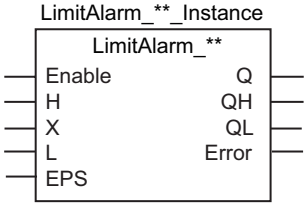
// 시분할 비례로 출력
TimeProportionalOut_instance(
  Enable         :=TimeProportionalOut_ON,
  AIn            :=MV[index],
  CtlPrd         :=CtlPrd[index],
  MinPlsWidth    :=MinPlsWidth[index],
  Delay          :=Delay[index],
  DOut           =>DOut_TPO,
  Error          =>Error_TimeProportionalOut[index]);

//출력 Ch1의 접점 00~03에 출력
CASE index OF
   INT#0:
     DO1:=DOut_TPO;
   INT#1:
     DO2:=DOut_TPO;
   INT#2:
     DO3:=DOut_TPO;
   ELSE
     DO4:=DOut_TPO;
 END_CASE;

END_FOR;
```

LimitAlarm_**

입력값이 하한 설정값 미만의 값 또는 상한 설정값을 초과한 값인 경우에 경보를 출력합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
LimitAlarm_**	상하한 경보 그룹	FB	 단, **는 REAL, LREAL 중 하나	LimitAlarm_**_instance(Enable, H, X, L, EPS, Q, QH, QL, Error); 단, **는 REAL, LREAL 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	동작 명령	입력	TRUE : 실행 FALSE : 경보 리셋	데이터 형식에 따름	–	FALSE
H	상한 설정값		입력값의 상한 설정값			0
X	입력값		감시 대상이 되는 값			
L	하한 설정값		입력값의 하한 설정값			
EPS	히스테리시스		경보의 히스테리시스	데이터 형식에 따름*1		
Q	경보 출력	출력	TRUE : 상한 경보와 하한 경보 중 하나가 ON FALSE : 상한 경보와 하한 경보가 모두 OFF	데이터 형식에 따름	–	–
QH	상한 경보		TRUE : 상한 경보 ON FALSE : 상한 경보 OFF			
QL	하한 경보		TRUE : 하한 경보 ON FALSE : 하한 경보 OFF			

*1. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
H														○	○					
X	'H'와 같은 데이터 형식																			
L	'H'와 같은 데이터 형식																			
EPS	'H'와 같은 데이터 형식																			
Q	○																			
QH	○																			
QL	○																			

기능

입력값이 하한 설정값에서 상한 설정값 사이의 값인지 감시합니다. 입력값이 하한 설정값 미만의 값 또는 상한 설정값을 초과한 값인 경우는 경보를 출력합니다. 온도 제어 시 측정 온도의 감시 등에 사용합니다.

동작 명령 'Enable'이 TRUE인 동안 입력값 'X'의 값을 감시합니다. 'X'의 값이 상한 설정값 'H'의 값을 초과하는 경우, 상한 경보 'QH'는 TRUE가 됩니다. 'X'의 값이 하한 설정값 'L'의 값 미만인 경우, 하한 경보 'QL'은 TRUE가 됩니다. 'QH', 'QL' 중 어느 한 값이 TRUE인 경우, 경보 출력 'Q'가 TRUE가 됩니다. 'X', 'H', 'L', 히스테리시스 'EPS'의 값은 'Enable'이 TRUE인 동안 상시 반영됩니다.

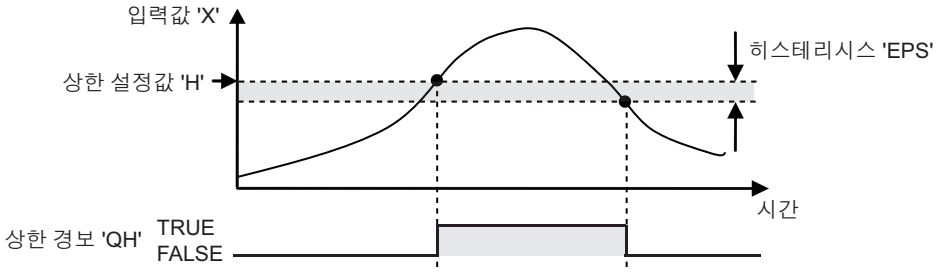
'Enable'이 FALSE인 경우에는 경보를 리셋합니다. 리셋에 의해 'Q', 'QH', 'QL'은 모두 FALSE가 됩니다.

'H', 'X', 'L', 'EPS'의 데이터 형식은 REAL, LREAL 중 하나입니다. 명령의 명칭은 'H', 'X', 'L', 'EPS'의 데이터 형식에 따라 다릅니다. 예를 들어, 명령의 명칭이 LimitAlarm_LREAL인 경우 'H', 'X', 'L', 'EPS'의 데이터 형식은 모두 LREAL이 됩니다.

상한 경보 'QH'의 동작

상한 경보 'QH'의 값은 다음과 같이 변화합니다. 히스테리시스의 설정을 통해 경보가 헌팅하는 것을 방지할 수 있습니다.

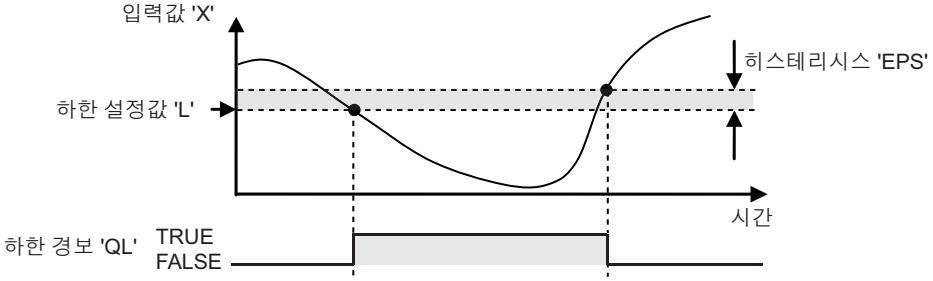
- 입력값 'X' > 상한 설정값 'H'인 경우, TRUE가 됩니다.
- 입력값 'X' < 상한 설정값 'H' - 히스테리시스 'EPS'인 경우, FALSE가 됩니다.



하한 경보 'QL'의 동작

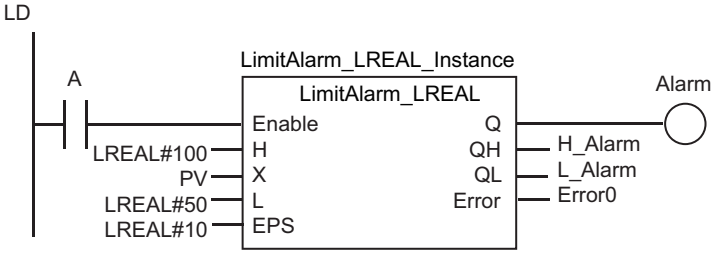
하한 경보 'QL'의 값은 다음과 같이 변화합니다. 히스테리시스의 설정을 통해 경보가 헌팅하는 것을 방지할 수 있습니다.

- 입력값 'X' < 하한 설정값 'L'인 경우, TRUE가 됩니다.
- 입력값 'X' > 하한 설정값 'L' + 히스테리시스 'EPS'인 경우, FALSE가 됩니다.

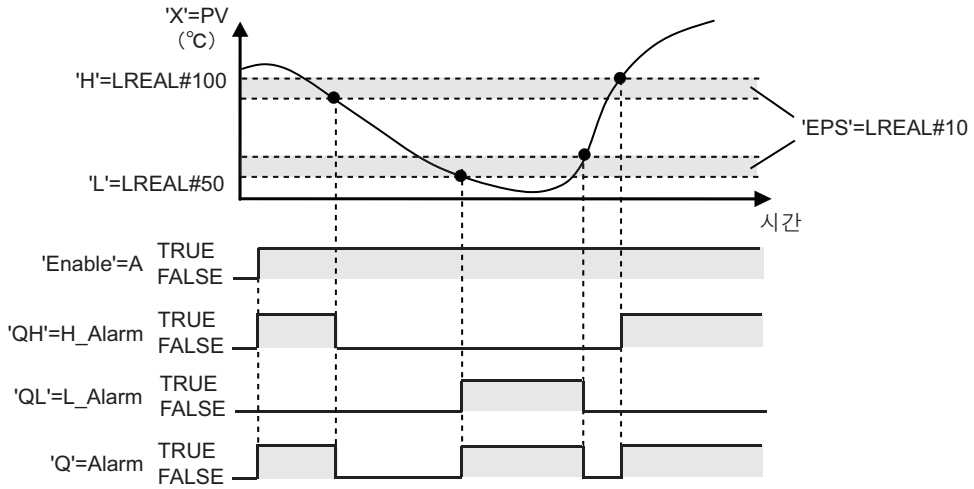


기술 예

상한 설정값 'H'가 100℃, 하한 설정값 'L'이 50℃, 히스테리시스 'EPS'가 10℃인 경우의 기술 예가 아래에 나와 있습니다.

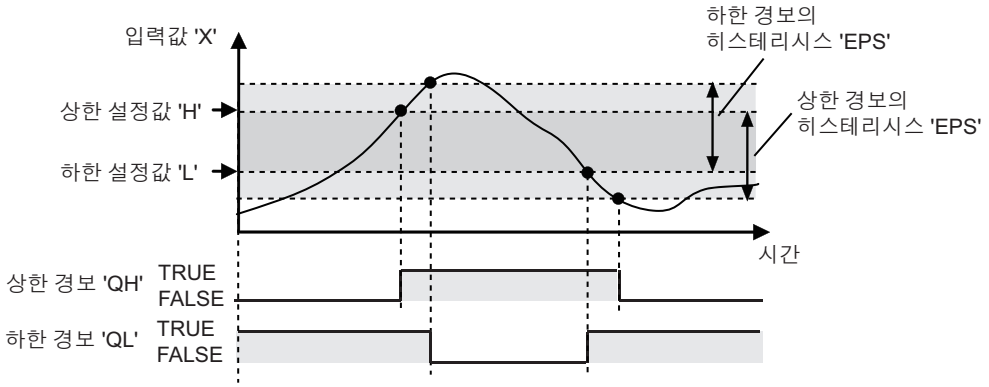



```
ST
 LimitAlarm_LREAL_instance(A,LREAL#100,PV,LREAL#50,LREAL#10,Alarm,H_Alarm,L_Alarm,Error0);
```

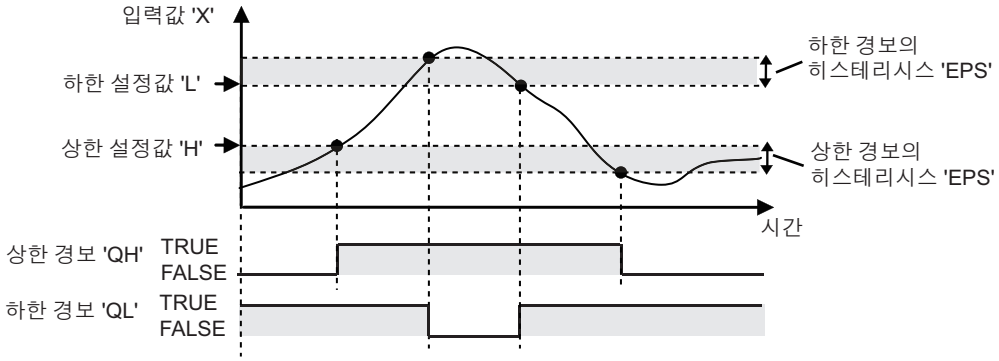


참고

- 명령 실행 시간을 단축하고 싶은 경우에는 LimitAlarm_REAL 명령을 사용하십시오.
- 'H'－'L'＜'EPS'가 되는 설정도 가능합니다. 이 경우, 'QH'와 'QL'은 양쪽 모두 TRUE가 될 수도 있습니다.



- 'H'＜'L'이 되는 설정도 가능합니다. 이 경우, 'QH' 또는 'QL' 중 하나가 항상 TRUE가 됩니다.



사용상의 주의 사항

- 'EPS'의 값이 유효 범위 밖인 경우 이상 상태가 됩니다. 'Error'는 TRUE가 되고 'Q', 'QH', 'QL'은 FALSE가 됩니다.
- 경보를 출력한 경우에 온도 제어의 출력을 OFF하는 등 이 명령을 안전 대책의 용도로 사용할 수 있습니다. 이 경우는 이상에 의해 'Q', 'QH' 및 'QL'이 FALSE가 되어도 안전 대책이 될 수 있도록 사용하십시오. 사용 예는 🕮 'TimeProportionalOut 명령(2-670페이지)'의 샘플 프로그램을 참조하십시오.

버전 관련 정보

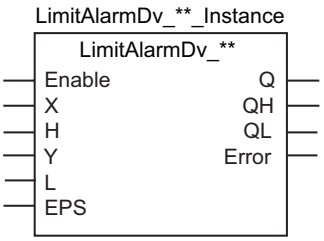
이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

🕮 'TimeProportionalOut 명령(2-670페이지)'의 샘플 프로그램을 참조하십시오.

LimitAlarmDv_**

기준값에서의 입력값의 편차가 하한 편차 설정값 또는 상한 편차 설정값을 초과한 값인 경우에 경보를 출력합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
LimitAlarmDv_**	상하한 편차 경보 그룹	FB	 단, **는 REAL, LREAL 중 하나	LimitAlarmDv_**instance(Enable, X, H, Y, L, EPS, Q, QH, QL, Error); 단, **는 REAL, LREAL 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	동작 명령	입력	TRUE : 실행 FALSE : 경보 리셋	데이터 형식에 따름	–	FALSE
X	입력값		감시 대상이 되는 값			0
H	상한 편차 설정값		기준값에 대한 상위 방향 편차 경보의 설정값			
Y	기준값		편차의 기준값			
L	하한 편차 설정값		기준값에 대한 하위 방향 편차 경보의 설정값			
EPS	히스테리시스		경보의 히스테리시스	데이터 형식에 따름*1		
Q	편차 경보 출력	출력	TRUE : 상한 편차 경보와 하한 편차 경보 중 하나가 ON FALSE : 상한 편차 경보와 하한 편차 경보가 모두 OFF	데이터 형식에 따름	–	–
QH	상한 편차 경보		TRUE : 상한 편차 경보 ON FALSE : 상한 편차 경보 OFF			
QL	하한 편차 경보		TRUE : 하한 편차 경보 ON FALSE : 하한 편차 경보 OFF			

*1. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
X														○	○					
H	'X'와 같은 데이터 형식																			
Y	'X'와 같은 데이터 형식																			
L	'X'와 같은 데이터 형식																			
EPS	'X'와 같은 데이터 형식																			
Q	○																			
QH	○																			
QL	○																			

기능

기준값에서의 입력값의 편차가 하한 편차 설정값 또는 상한 편차 설정값까지의 값인지 감시합니다. 편차가 하한 편차 설정값 또는 상한 편차 설정값을 초과한 값인 경우는 경보를 출력합니다. 온도 제어 시 목표 온도에서의 측정 온도의 편차를 감시하는 경우 등에 사용합니다.

동작 명령 'Enable'이 TRUE인 동안 기준값 'Y'에서의 입력값 'X'의 편차를 감시합니다. 'Y'에서의 'X'의 상위 방향 편차가 상한 편차 설정값 'H'의 값을 초과하는 경우, 상한 편차 경보 'QH'는 TRUE가 됩니다. 'Y'에서의 'X'의 하위 방향 편차가 하한 편차 설정값 'L'의 값을 초과하는 경우, 하한 편차 경보 'QL'은 TRUE가 됩니다. 'QH', 'QL' 중 어느 한 값이 TRUE인 경우, 경보 출력 'Q'가 TRUE가 됩니다. 'X', 'H', 'Y', 'L', 히스테리시스 'EPS'의 값은 'Enable'이 TRUE인 동안 상시 반영됩니다.

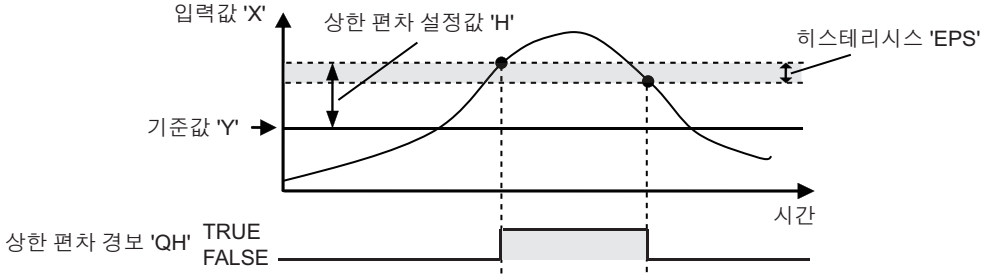
'Enable'이 FALSE인 경우에는 경보를 리셋합니다. 리셋에 의해 'Q', 'QH', 'QL'은 모두 FALSE가 됩니다.

'X', H', 'Y', 'L', 'EPS'의 데이터 형식은 REAL, LREAL 중 하나입니다. 명령의 명칭은 'X', H', 'Y', 'L', 'EPS'의 데이터 형식에 따라 다릅니다. 예를 들어, 명령의 명칭이 LimitAlarmDv_LREAL인 경우, 'X', 'H', 'Y', 'L', 'EPS'의 데이터 형식은 모두 LREAL이 됩니다.

상한 편차 경보 'QH'의 동작

상한 편차 경보 'QH'는 기준값 'Y'에 대한 상위 방향의 편차 경보입니다. 'QH'의 값은 다음과 같이 변화합니다. 히스테리시스를 통해 편차 경보가 헌팅하는 것을 방지할 수 있습니다.

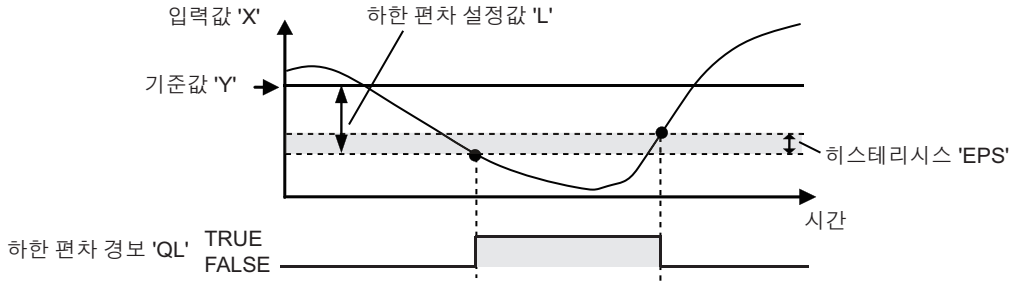
- 입력값 'X' - 기준값 'Y' > 상한 편차 설정값 'H'인 경우, TRUE가 됩니다.
- 입력값 'X' - 기준값 'Y' < 상한 편차 설정값 'H' - 히스테리시스 'EPS'인 경우, FALSE가 됩니다.



하한 편차 경보 'QL'의 동작

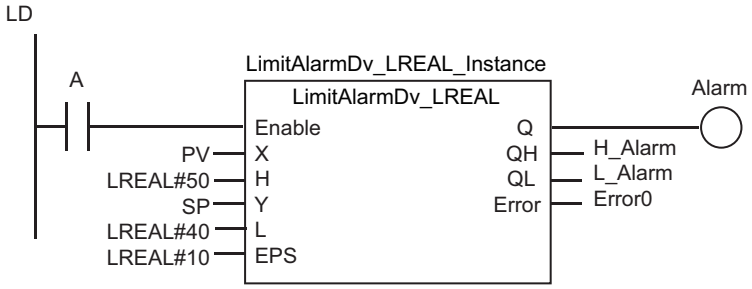
하한 편차 경보 'QH'은 기준값 'Y'에 대한 하위 방향의 편차 경보입니다. 'QL'의 값은 다음과 같이 변화합니다. 히스테리시스를 통해 편차 경보가 헌팅하는 것을 방지할 수 있습니다.

- -(입력값 'X' - 기준값 'Y')＞하한 편차 설정값 'L'인 경우, TRUE가 됩니다.
- -(입력값 'X' - 기준값 'Y')＜하한 편차 설정값 'L' - 히스테리시스 'EPS'인 경우, FALSE가 됩니다.

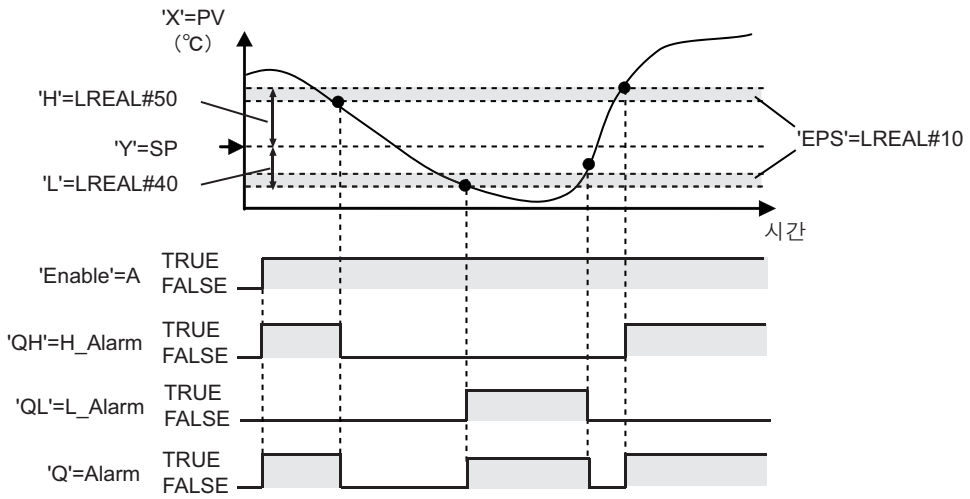


기술 예

상한 편차 설정값 'H'가 50℃, 하한 편차 설정값 'L'이 40℃, 히스테리시스 'EPS'가 10℃인 경우의 기술 예가 아래에 나와 있습니다.

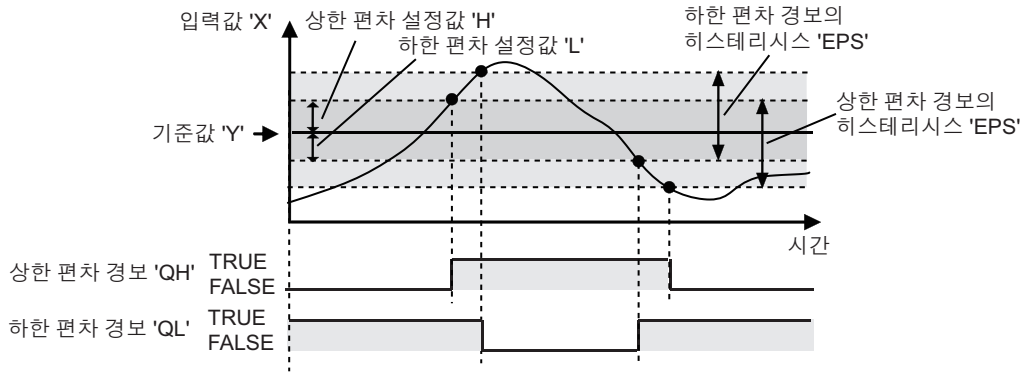



```
ST
 LimitAlarmDv_LREAL_instance(A,PV,LREAL#50,SP,LREAL#40,LREAL#10,Alarm,H_Alarm,L_Alarm,Error0);
```

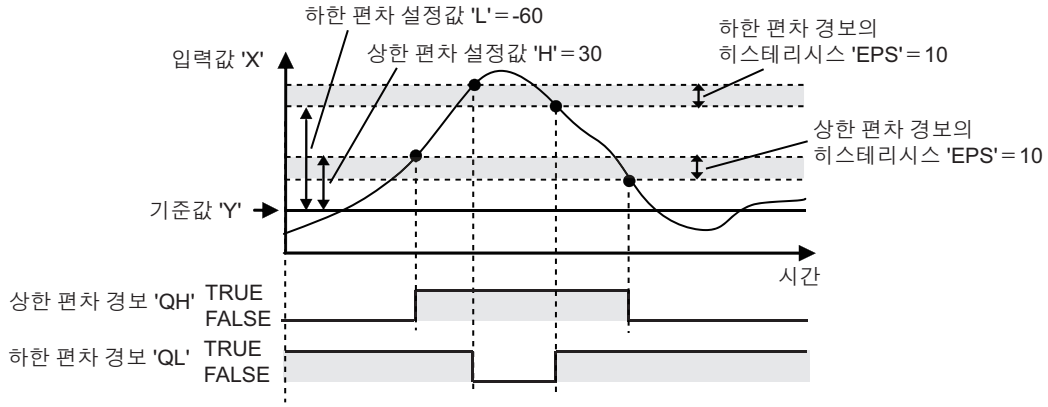


참고

- 명령 실행 시간을 단축하고 싶은 경우에는 LimitAlarmDv_REAL 명령을 사용하십시오.
- 'H'+'L'＜'EPS'가 되는 설정도 가능합니다. 이 경우, 'QH'와 'QL'은 양쪽 모두 TRUE가 될 수도 있습니다.



- 'H'+'L'＜0이 되는 설정도 가능합니다. 이 경우, 'QH'와 'QL' 중 하나가 항상 TRUE가 됩니다. 예를 들어, 'L'의 값이 -60, 'H'의 값이 30인 경우는 다음 그림과 같이 동작합니다.



사용상의 주의 사항

- 'EPS'의 값이 유효 범위 밖인 경우 이상 상태가 됩니다. 'Error'는 TRUE가 되고 'Q', 'QH', 'QL'은 FALSE가 됩니다.
- 편차 경보를 출력한 경우에 온도 제어의 출력을 OFF하는 등 이 명령을 안전 대책의 용도로 사용할 수 있습니다. 이 경우는 이상에 의해 'Q', 'QH' 및 'QL'이 FALSE가 되어도 안전 대책이 될 수 있도록 사용하십시오. 사용 예는 'TimeProportionalOut 명령(2-670페이지)'의 샘플 프로그램을 참조하십시오.

버전 관련 정보

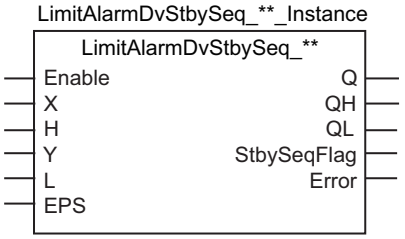
이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

'TimeProportionalOut 명령(2-670페이지)'의 샘플 프로그램을 참조하십시오.

LimitAlarmDvStbySeq_**

대기 시퀀스가 있는 상하한 편차 경보를 출력합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
LimitAlarmDvStbySeq_**	대기 시퀀스가 있는 상하한 편차 경보 그룹	FB	 단, **는 REAL, LREAL 중 하나	LimitAlarmDvStbySeq_**_instance(Enable, X, H, Y, L, EPS, Q, QH, QL, StbySeqFlag, Error); 단, **는 REAL, LREAL 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	동작 명령	입력	TRUE : 실행 FALSE : 경보 리셋	데이터 형식에 따름	–	FALSE
X	입력값		편차 경보의 대상이 되는 값			
H	상한 편차 설정값		기준값에 대한 상위 방향 편차 경보의 설정값			0
Y	기준값		편차의 기준값			
L	하한 편차 설정값		기준값에 대한 하위 방향 편차 경보의 설정값			
EPS	히스테리시스		경보의 히스테리시스	데이터 형식에 따름*1		
Q	편차 경보 출력	출력	TRUE : 상한 편차 경보와 하한 편차 경보 중 하나가 ON FALSE : 상한 편차 경보와 하한 편차 경보가 모두 OFF	데이터 형식에 따름	–	–
QH	상한 편차 경보		TRUE : 상한 편차 경보 ON FALSE : 상한 편차 경보 OFF			
QL	하한 편차 경보		TRUE : 하한 편차 경보 ON FALSE : 하한 편차 경보 OFF			
StbySeqFlag	대기 시퀀스 유효 플래그		TRUE : 유효 FALSE : 무효			

*1. 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
X														○	○					
H	'X'와 같은 데이터 형식																			
Y	'X'와 같은 데이터 형식																			
L	'X'와 같은 데이터 형식																			
EPS	'X'와 같은 데이터 형식																			
Q	○																			
QH	○																			
QL	○																			
StbySeqFlag	○																			

기능

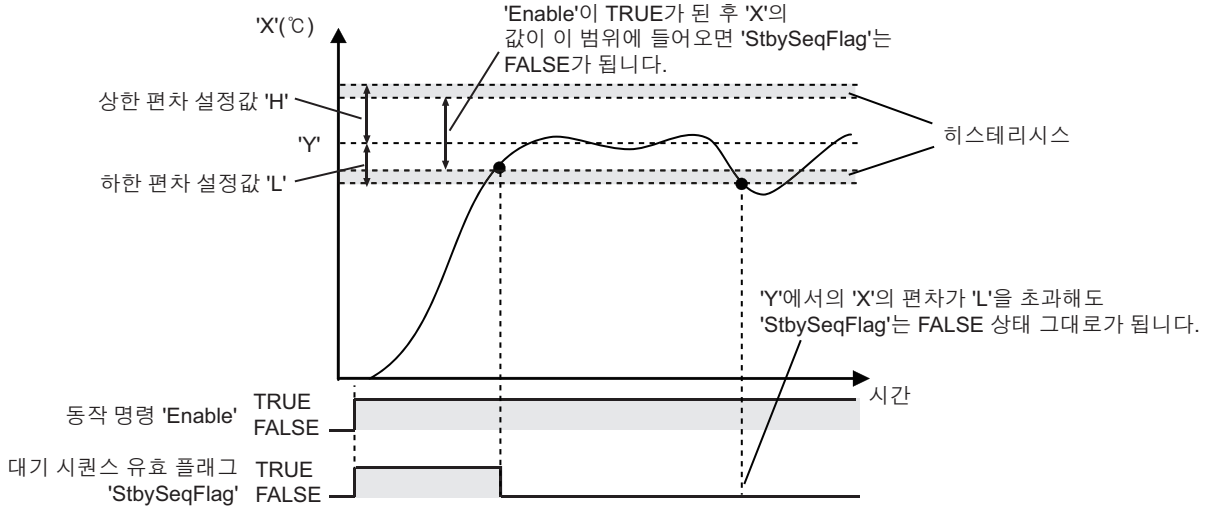
기준값에서의 입력값의 편차가 하한 편차 설정값까지, 또한 상한 편차 설정값까지의 값인지 감시합니다. 편차가 하한 편차 설정값 또는 상한 편차 설정값을 초과한 값인 경우는 경보를 출력합니다. 단, 일단 편차가 하한 편차 설정값까지, 또한 상한 편차 설정값까지의 값이 될 때까지는 경보를 출력하지 않습니다. 온도 제어 시 측정 온도가 안정될 때까지는 편차 경보를 출력하지 않는 경우 등에 사용합니다.

동작 명령 'Enable'이 TRUE인 동안 기준값 'Y'에서의 입력값 'X'의 편차를 감시합니다. 단, 대기 시퀀스 유효 플래그 'StbySeqFlag'가 TRUE인 동안은 편차를 감시하지 않습니다.
'Y'에서의 'X'의 상위 방향 편차가 상한 편차 설정값 'H'의 값을 초과하는 경우, 상한 편차 경보 'QH'는 TRUE가 됩니다. 'Y'에서의 'X'의 하위 방향 편차가 하한 편차 설정값 'L'의 값을 초과하는 경우, 하한 편차 경보 'QL'은 TRUE가 됩니다. 'QH', 'QL' 중 어느 한 값이 TRUE인 경우, 경보 출력 'Q'가 TRUE가 됩니다.
'X', 'H', 'Y', 'L', 'EPS'의 값은 'Enable'이 TRUE인 동안 상시 반영됩니다.

'Enable'이 FALSE인 경우에는 경보를 리셋합니다. 리셋에 의해 'Q', 'QH', 'QL', 'StbySeqFlag'는 모두 FALSE가 됩니다.

'StbySeqFlag'는 'Enable'이 TRUE가 된 후 다음 조건을 모두 충족하면 FALSE가 됩니다. 'StbySeqFlag'는 한번 FALSE가 되면 'Enable'이 FALSE에서 TRUE로 변화할 때까지는 TRUE가 되지 않습니다.

- 입력값 'X' - 기준값 'Y' < 상한 편차 설정값 'H' - 히스테리시스 'EPS'
- -(입력값 'X' - 기준값 'Y') < 하한 편차 설정값 'L' - 히스테리시스 'EPS'

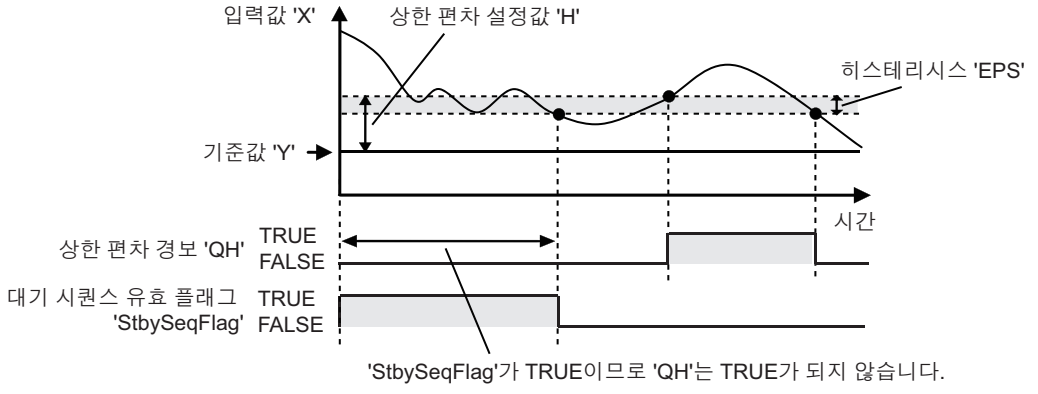


'X', 'H', 'Y', 'L', 'EPS'의 데이터 형식은 REAL, LREAL 중 하나입니다. 명령의 명칭은 'X', H', 'Y', 'L', 'EPS'의 데이터 형식에 따라 다릅니다. 예를 들어, 명령의 명칭이 LimitAlarmDvStbySeq_LREAL인 경우, 'X', 'H', 'Y', 'L', 'EPS'의 데이터 형식은 모두 LREAL이 됩니다.

상한 편차 경보 'QH'의 동작

상한 편차 경보 'QH'는 기준값 'Y'에 대한 상위 방향의 편차 경보입니다. 'StbySeqFlag'가 FALSE인 경우 'QH'의 값은 다음과 같이 변화합니다. 히스테리시스를 설정하여 편차 경보가 헌팅하는 것을 방지할 수 있습니다.

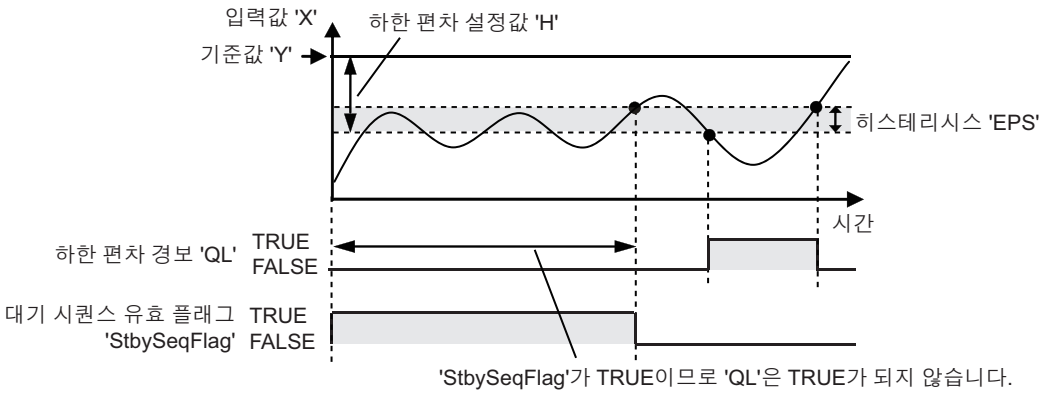
- 입력값 'X' - 기준값 'Y' > 상한 편차 설정값 'H'인 경우, TRUE가 됩니다.
- 입력값 'X' - 기준값 'Y' < 상한 편차 설정값 'H' - 히스테리시스 'EPS'인 경우, FALSE가 됩니다.



하한 편차 경보 'QL'의 동작

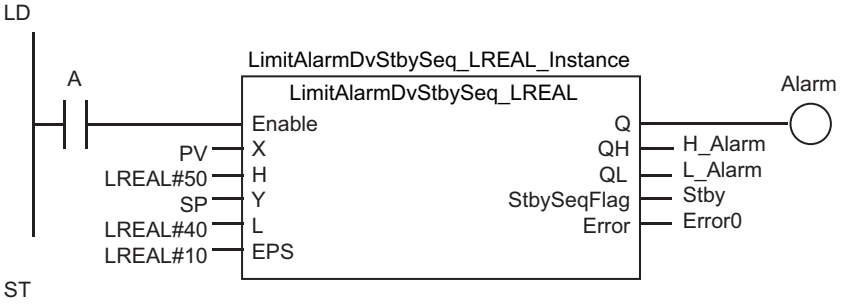
하한 편차 경보 'QH'는 기준값 'Y'에 대한 하위 방향의 편차 경보입니다. 'StbySeqFlag'가 FALSE인 경우 'QL'의 값은 다음과 같이 변화합니다. 히스테리시스를 설정하여 편차 경보가 헌팅하는 것을 방지할 수 있습니다.

- -(입력값 'X' - 기준값 'Y')＞하한 편차 경보 'L'인 경우, TRUE가 됩니다.
- -(입력값 'X' - 기준값 'Y')＜하한 편차 경보 'L' - 히스테리시스 'EPS'인 경우, FALSE가 됩니다.



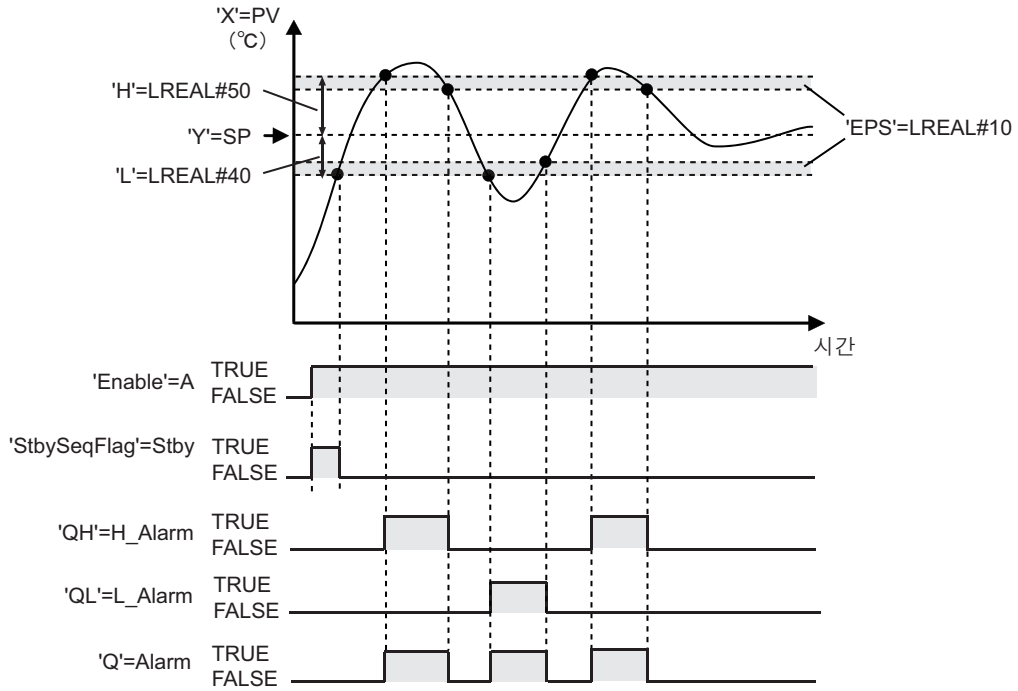
기술 예

상한 편차 설정값 'H'가 50℃, 하한 편차 설정값 'L'이 40℃, 히스테리시스 'EPS'가 10℃인 경우의 기술 예가 아래에 나와 있습니다.



ST

```
LimitAlarmDvStbySeq_LREAL_Instance(A,PV,LREAL#50,SP,LREAL#40,LREAL#10,Alarm,H_Alarm,L_Alarm,Stby,Error0);
```



참고

- 명령 실행 시간을 단축하고 싶은 경우에는 LimitAlarmDvStbySeq_REAL 명령을 사용하십시오.
- 'H'+'L'＜'EPS'가 되는 설정도 가능합니다. 이 경우, 'QH'와 'QL'은 양쪽 모두 TRUE가 될 수도 있습니다. 'LimitAlarmDv_** 명령(2-690페이지)'을 참조하십시오.
- 'H'+'L'＜0이 되는 설정도 가능합니다. 이 경우, 'StbySeqFlag'가 FALSE인 동안은 'QH'와 'QL' 중 하나가 항상 TRUE가 됩니다. 'LimitAlarmDv_** 명령(2-690페이지)'을 참조하십시오.

사용상의 주의 사항

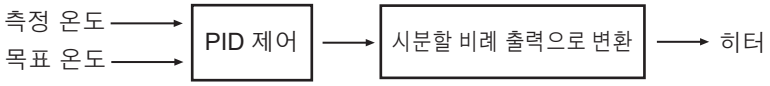
- 'EPS'의 값이 유효 범위 밖인 경우 이상 상태가 됩니다. 'Error'는 TRUE가 되고 'Q', 'QH', 'QL'은 FALSE가 됩니다.
- 편차 경보를 출력한 경우에 온도 제어의 출력을 OFF하는 등 이 명령을 안전 대책의 용도로 사용할 수 있습니다. 이 경우는 이상에 의해 'Q', 'QH' 및 'QL'이 FALSE가 되어도 안전 대책이 될 수 있도록 사용하십시오. 사용 예는 샘플 프로그램을 참조하십시오.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

샘플 프로그램

상하한 경보, 대기 시퀀스가 있는 상하한 편차 경보가 부가된 4점의 온도 제어를 수행합니다. 제어 방식은 PID 제어를 사용하며, PID 제어의 조작량은 시분할 비례 출력으로 변환하여 히터에 출력합니다.



사양

다음과 같은 사양으로 온도 제어를 수행합니다.

항목	사양
입력 유니트	절연형 고분해능 풀 멀티 입력 유니트 CJ1W-PH41U
입력 종류	모두 K형 열전대
출력 유니트	트랜지스터 출력 유니트 CJ1W-OD212
목표 온도	100℃
상한 온도	200℃
하한 온도	0℃
상하한 경보의 히스테리시스	5℃
상한 편차 온도	50℃
하한 편차 온도	50℃
상하한 편차 경보의 히스테리시스	3℃
PID 연산의 샘플링 주기	100ms
출력 제어 주기	1s

구성 · 설정

입력 유니트 CJ1W-PH41U의 설정은 다음과 같습니다.

항목명	설정값
Input1:Input signal type	K(1)
Input2:Input signal type	K(1)
Input3:Input signal type	K(1)
Input4:Input signal type	K(1)

I/O 맵 설정은 다음과 같습니다.

유니트	포트	설명	변수
CJ1W-PH41U	Ch1_AIInPV	입력 번호 1 측정값(INT 형식)	AI1
	Ch2_AIInPV	입력 번호 2 측정값(INT 형식)	AI2
	Ch3_AIInPV	입력 번호 3 측정값(INT 형식)	AI3
	Ch4_AIInPV	입력 번호 4 측정값(INT 형식)	AI4
CJ1W-OD212	Ch1_Out00	출력 CH1 접점 00	DO1
	Ch1_Out01	출력 CH1 접점 01	DO2
	Ch1_Out02	출력 CH1 접점 02	DO3
	Ch1_Out03	출력 CH1 접점 03	DO4

4점의 온도 제어에 대한 입력 및 출력의 대응은 다음과 같습니다.

입력	출력
AI1	DO1
AI2	DO2
AI3	DO3
AI4	DO4

프로그램을 할당하는 태스크의 주기는 1ms로 합니다.

● 구성도

🕮 'TimeProportionalOut 명령(2-670페이지)'의 샘플 프로그램을 참조하십시오.

처리 내용

다음 처리를 4점 분량 수행합니다.

1 **측정 온도를 취득합니다.**

2 **LimitAlarm_REAL 명령을 사용하여 측정 온도에 대해 상하한 경보를 출력합니다.**

3 **LimitAlarm_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우는 안전 대책을 위한 출력을 합니다.**

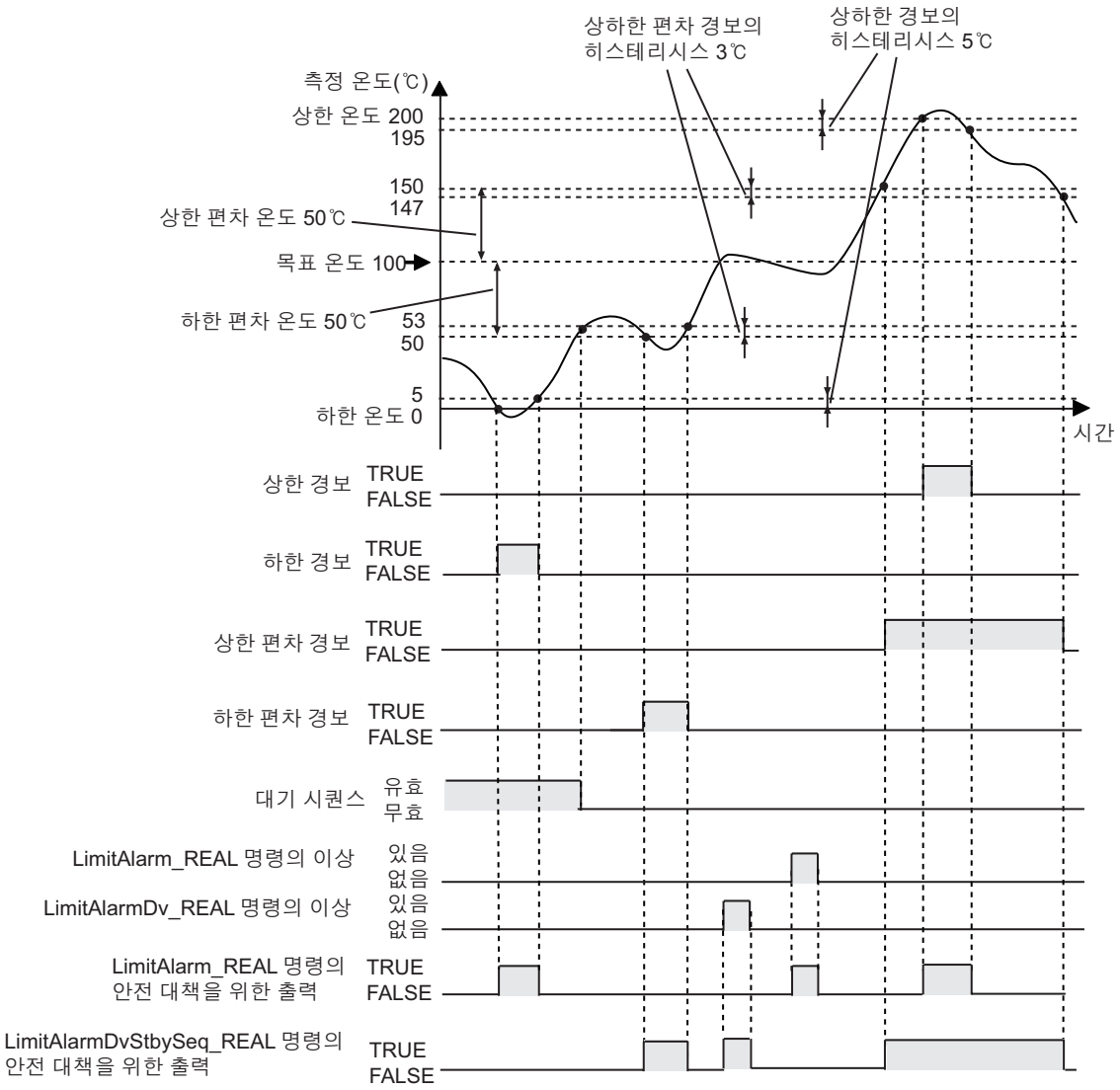
4 **LimitAlarmDvStbySeq_REAL 명령을 사용하여 목표 온도에서 측정 온도의 편차에 대해 대기 시퀀스가 있는 상하한 편차 경보를 출력합니다.**

5 **LimitAlarmDvStbySeq_REAL 명령에서 이상이 발생한 경우 또는 상하한 편차 경보를 출력한 경우는 안전 대책을 위한 출력을 합니다.**

6 **PIDAT 명령을 사용하여 온도를 제어합니다.**

7 **TimeProportionalOut 명령을 사용하여 조작량을 시분할 비례로 히터에 출력합니다.**

● **상하한 경보, 대기 시퀀스가 있는 상하한 편차 경보의 동작**



응용 프로그램

글로벌 변수의 정의

글로벌 변수

명칭	데이터 형식	할당대상*1	코멘트
AI1	INT	IOBus://rack#0/slot#0/Ch1_AIInPV	입력 번호 1 측정값(INT 형식)
AI2	INT	IOBus://rack#0/slot#0/Ch2_AIInPV	입력 번호 2 측정값(INT 형식)
AI3	INT	IOBus://rack#0/slot#0/Ch3_AIInPV	입력 번호 3 측정값(INT 형식)
AI4	INT	IOBus://rack#0/slot#0/Ch4_AIInPV	입력 번호 4 측정값(INT 형식)
DO1	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out00	출력 CH1 접점 00
DO2	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out01	출력 CH1 접점 01
DO3	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out02	출력 CH1 접점 02
DO4	BOOL	IOBus://rack#0/slot#1/Ch1_Out/Ch1_Out03	출력 CH1 접점 03

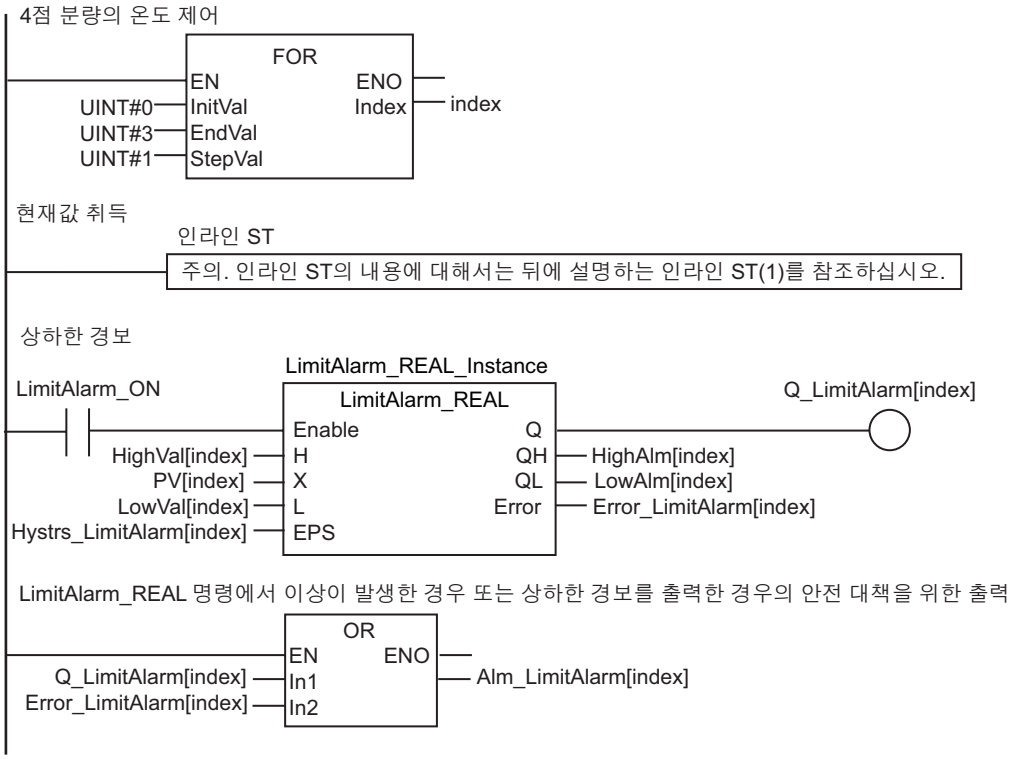
*1. CJ1W-PH41U를 랙 번호 0의 슬롯 번호 0, CJ1W-OD212를 랙 번호 0의 슬롯 번호 1에 장착한 경우의 할당대상입니다.

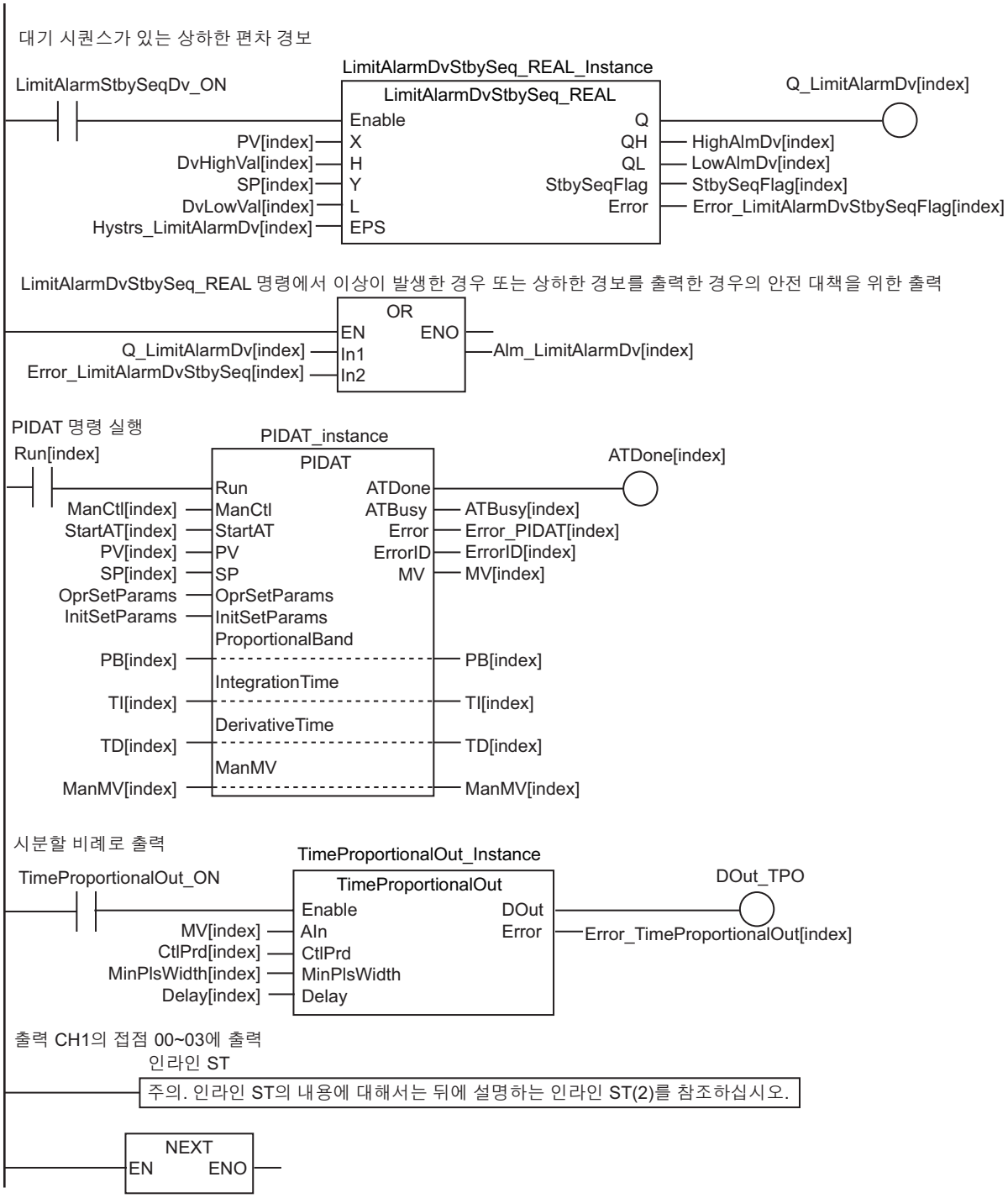
(주의) I/O 맵 설정에 의해 각 유니트의 포트에 할당된 글로벌 변수가 자동으로 생성됩니다.

LD

명칭	데이터 형식	초기값	유지	코멘트
index	UINT	0	☐	루프 인덱스
LimitAlarm_ON	BOOL	True	☐	상하한 경보 실행
LimitAlarmDvStbySeq_ON	BOOL	True	☐	대기 시퀀스가 있는 상하한 편차 경보 실행
TimeProportionalOut_ON	BOOL	True	☐	시분할 비례 출력 실행
AI	INT	0	☐	현재값
PV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	측정값
SP	ARRAY[0..3] OF REAL	[4(100)]	☐	목표 온도
DOut_TPO	BOOL	False	☐	시분할 비례 출력
HighVal	ARRAY[0..3] OF REAL	[4(200)]	☐	상하한 경보의 상한 설정값
LowVal	ARRAY[0..3] OF REAL	[4(0.0)]	☐	상하한 경보의 하한 설정값
Hystrs_LimitAlarm	ARRAY[0..3] OF REAL	[4(5)]	☐	상하한 경보의 히스테리시스
Q_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보 출력
HighAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 경보
LowAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 경보
Error_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarm_REAL 명령의 이상
Alm_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보의 안전 대책을 위한 출력
DvHighVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 상한 편차 설정값
DvLowVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 하한 편차 설정값
Q_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보 출력
HighAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 편차 경보
LowAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 편차 경보
StbySeqFlag	ARRAY[0..3] OF BOOL	[4(False)]	☐	대기 시퀀스 유효 플래그
Error_LimitAlarmDvStbySeq	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarmDvStbySeq_REAL 명령의 이상
Hystrs_LimitAlarmDv	ARRAY[0..3] OF REAL	[4(3)]	☐	상하한 편차 경보의 히스테리시스
Alm_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보의 안전 대책을 위한 출력
Run	ARRAY[0..3] OF BOOL	[4(False)]	☐	실행 조건
ManCtl	ARRAY[0..3] OF BOOL	[4(False)]	☐	매뉴얼/자동 전환
StartAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 조건
OprSetParams	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=False, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:=-10.0,RngUpLmt:=1000.0, DirOpr:=False)	☐	초기 설정 파라메터
PB	ARRAY[0..3] OF REAL	[4(10)]	☑	비례대
TI	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	적분 시간
TD	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	미분 시간

명칭	데이터 형식	초기값	유지	코멘트
ManMV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	수동 조작량
ATDone	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 정상 종료
ATBusy	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 중
Error_PIDAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	PIDAT 명령의 이상
ErrorID	ARRAY[0..3] OF WORD	[4(16#0)]	☐	PIDAT 명령의 이상 ID
MV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	조작량
CtlPrd	ARRAY[0..3] OF TIME	[4(T#1s)]	☐	제어 주기
MinPlsWidth	ARRAY[0..3] OF REAL	[4(0.0)]	☐	최소 펄스 폭
Delay	ARRAY[0..3] OF REAL	[4(0.0)]	☐	온 딜레이 시간
Error_TimeProportionalOut	ARRAY[0..3] OF BOOL	[4(False)]	☐	TimeProportionalOut 명령의 이상
LimitAlarm_REAL_instance	LimitAlarm_REAL		☐	
LimitAlarmDvStbySeq_REAL_instance	LimitAlarmDvStbySeq_REAL		☐	
PIDAT_instance	PIDAT		☐	
TimeProportionalOut_instance	TimeProportionalOut		☐	





● 인라인 ST(1)의 내용

```
// 입력 1~4의 값을 취득
CASE index OF
  INT#0:
    AI:=AI1;
  INT#1:
    AI:=AI2;
  INT#2:
    AI:=AI3;
  ELSE
    AI:=AI4;
END_CASE;
```

```
// 현재값 AI 를 실수로 변환
PV[index]:=INT_TO_REAL(AI)/REAL#10.0;   // CJ1W-PH41U 의 출력은 측정 온도의 10 배이므로 10.0
                                           으로 제산
```

● 인라인 ST(2)의 내용

```
// 출력 CH1의 접점 00~03에 출력
CASE index OF
  INT#0:
    DO1:=DOut_TPO;
  INT#1:
    DO2:=DOut_TPO;
  INT#2:
    DO3:=DOut_TPO;
  ELSE
    DO4:=DOut_TPO;
 END_CASE;
```

ST

명칭	데이터 형식	초기값	유지	코멘트
index	UINT	0	☐	루프 인덱스
LimitAlarm_ON	BOOL	True	☐	상하한 경보 실행
LimitAlarmDvStbySeq_ON	BOOL	True	☐	대기 시퀀스가 있는 상하한 편차 경보 실행
TimeProportionalOut_ON	BOOL	True	☐	시분할 비례 출력 실행
AI	INT	0	☐	현재값
PV	ARRAY[0..3] OF REAL	0.0	☐	측정값
SP	ARRAY[0..3] OF REAL	[4(100)]	☐	목표 온도
DOut_TPO	BOOL	False	☐	시분할 비례 출력
HighVal	ARRAY[0..3] OF REAL	[4(200)]	☐	상하한 경보의 상한 설정값
LowVal	ARRAY[0..3] OF REAL	[4(0.0)]	☐	상하한 경보의 하한 설정값
Hystrs_LimitAlarm	ARRAY[0..3] OF REAL	[4(5)]	☐	상하한 경보의 히스테리시스
Q_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보 출력
HighAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 경보
LowAlm	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 경보
Error_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarm_REAL 명령의 이상
Alm_LimitAlarm	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 경보의 안전 대책을 위한 출력
DvHighVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 상한 편차 설정값
DvLowVal	ARRAY[0..3] OF REAL	[4(50)]	☐	상하한 편차 경보의 하한 편차 설정값
Q_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보 출력
HighAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상한 편차 경보
LowAlmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	하한 편차 경보
StbySeqFlag	ARRAY[0..3] OF BOOL	[4(False)]	☐	대기 시퀀스 유효 플래그
Error_LimitAlarmDvStbySeq	ARRAY[0..3] OF BOOL	[4(False)]	☐	LimitAlarmDvStbySeq_REAL 명령의 이상
Hystrs_LimitAlarmDv	ARRAY[0..3] OF REAL	[4(3)]	☐	상하한 편차 경보의 히스테리시스
Alm_LimitAlarmDv	ARRAY[0..3] OF BOOL	[4(False)]	☐	상하한 편차 경보의 안전 대책을 위한 출력
Run	ARRAY[0..3] OF BOOL	[4(False)]	☐	실행 조건

명칭	데이터 형식	초기값	유지	코멘트
ManCtl	ARRAY[0..3] OF BOOL	[4(False)]	☐	매뉴얼/자동 전환
StartAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 조건
OprSetParams	_sOPR_SET_PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=False, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	☐	운전 중 설정 파라메터
InitSetParams	_sINIT_SET_PARAMS	(SampTime:=T#100ms, RngLowLmt:= -10.0,RngUpLmt:=1000.0, DirOpr:=False)	☐	초기 설정 파라메터
PB	ARRAY[0..3] OF REAL	[4(10)]	☑	비례대
TI	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	적분 시간
TD	ARRAY[0..3] OF TIME	[4(T#0S)]	☑	미분 시간
ManMV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	수동 조작량
ATDone	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 정상 종료
ATBusy	ARRAY[0..3] OF BOOL	[4(False)]	☐	자동 조정 실행 중
Error_PIDAT	ARRAY[0..3] OF BOOL	[4(False)]	☐	PIDAT 명령의 이상
ErrorID	ARRAY[0..3] OF WORD	[4(16#0)]	☐	PIDAT 명령의 이상 ID
MV	ARRAY[0..3] OF REAL	[4(0.0)]	☐	조작량
CtlPrd	ARRAY[0..3] OF TIME	[4(T#1s)]	☐	제어 주기
MinPlsWidth	ARRAY[0..3] OF REAL	[4(0.0)]	☐	최소 펄스 폭
Delay	ARRAY[0..3] OF REAL	[4(0.0)]	☐	온 딜레이 시간
Error_TimeProportionalOut	ARRAY[0..3] OF BOOL	[4(False)]	☐	TimeProportionalOut 명령의 이상
LimitAlarm_REAL_instance	LimitAlarm_REAL		☐	
LimitAlarmDvStbySeq_REAL_instance	LimitAlarmDvStbySeq_REAL		☐	
PIDAT_instance	PIDAT		☐	
TimeProportionalOut_instance	TimeProportionalOut		☐	

```
// 4점 분량의 온도 제어
FOR index:=UINT#0 TO UINT#3 BY UINT#1 DO

  // 입력 1~4의 값을 취득
  CASE index OF
    INT#0:
      AI:=AI1;
    INT#1:
      AI:=AI2;
    INT#2:
      AI:=AI3;
    ELSE
      AI:=AI4;
  END_CASE;

  // 현재값 AI를 실수로 변환
PV[index]:=INT_TO_REAL(AI)/REAL#10.0;    // CJ1W-PH41U의 출력은 측정 온도의 10배이므로 10.0
                                            으로 제산
```

```
// 상하한 경보
LimitAlarm_REAL_instance(
  Enable :=LimitAlarm_ON,
  H      :=HighVal[index],
  X      :=PV[index],
  L      :=LowVal[index],
  EPS    :=Hystrs_LimitAlarm[index],
  Q      =>Q_LimitAlarm[index],
  QH     =>HighAlm[index],
  QL     =>LowAlm[index],
  Error  =>Error_LimitAlarm[index]);

// LimitAlarm_REAL 명령에서 이상이 발생한 경우 또는 상하한 경보를 출력한 경우의 안전 대책을 위한 출력
 Alm_LimitAlarm[index]:=Q_LimitAlarm[index] OR Error_LimitAlarm[index];

// 대기 시퀀스가 있는 상하한 편차 경보
LimitAlarmDvStbySeq_REAL_instance(
  Enable       :=LimitAlarmDvStbySeq_ON,
  X            :=PV[index],
  H            :=DvHighVal[index],
  Y            :=SP[index],
  L            :=DvLowVal[index],
  EPS          :=Hystrs_LimitAlarmDv[index],
  Q            =>Q_LimitAlarmDv[index],
  QH           =>HighAlmDv[index],
  QL           =>LowAlmDv[index],
  StbySeqFlag =>StbySeqFlag[index],
  Error        =>Error_LimitAlarmDvStbySeq[index]);

// LimitAlarmDvStbySeq_REAL 명령에서 이상이 발생한 경우
// 또는 상하한 경보를 출력한 경우의 안전 대책을 위한 출력
Alm_LimitAlarmDv[index]:=Q_LimitAlarmDv[index] OR Error_LimitAlarmDvStbySeq[index];

// PIDAT 명령 실행
PIDAT_instance(
  Run              :=Run[index],
  ManCtl           :=ManCtl[index],
  StartAT          :=StartAT[index],
  PV               :=PV[index],
  SP               :=SP[index],
  OprSetParams     :=OprSetParams,
  InitSetParams    :=InitSetParams,
  ProportionalBand :=PB[index],
  IntegrationTime  :=TI[index],
  DerivativeTime   :=TD[index],
  ManMV            :=ManMV[index],
  ATDone           =>ATDone[index],
  ATBusy           =>ATBusy[index],
  Error            =>Error_PIDAT[index],
  ErrorID          =>ErrorID[index],
  MV               =>MV[index]);

// 시분할 비례로 출력
TimeProportionalOut_instance(
  Enable       :=TimeProportionalOut_ON,
  AIn          :=MV[index],
  CtlPrd       :=CtlPrd[index],
  MinPlsWidth  :=MinPlsWidth[index],
```

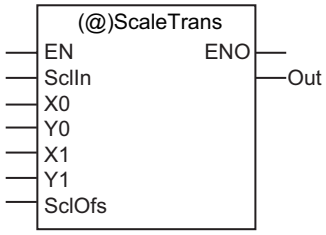
```
    Delay          :=Delay[index],
    DOut           =>DOut_TPO,
    Error          =>Error_TimeProportionalOut[index]);

  //출력 Ch1의 접점 00~03에 출력
  CASE index OF
    INT#0:
      DO1:=DOut_TPO;
    INT#1:
      DO2:=DOut_TPO;
    INT#2:
      DO3:=DOut_TPO;
    ELSE
      DO4:=DOut_TPO;
  END_CASE;

END_FOR;
```

ScaleTrans

입력값을 입력 범위에서 출력 범위로 변환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ScaleTrans	스케일 변환	FUN		Out:=ScaleTrans(SclIn,X0,Y0,X1, Y1,SclOfs);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
SclIn	입력값	입력	변환의 대상이 되는 값	데이터 형식에 따름	–	(*)
X0	입력 범위 하한값		입력 범위의 하한값			0
Y0	출력 범위 하한값		출력 범위의 하한값			
X1	입력 범위 상한값		입력 범위의 상한값			
Y1	출력 범위 상한값		출력 범위의 상한값			
SclOfs	오프셋 값		출력값에 더할 오프셋 값			
Out	반환값	출력	항상 TRUE	TRUE만	–	–
Dout						

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SclIn														○	○					
X0														○	○					
X1														○	○					
Y0														○	○					
Y1														○	○					
SclOfs														○	○					
Out														○	○					

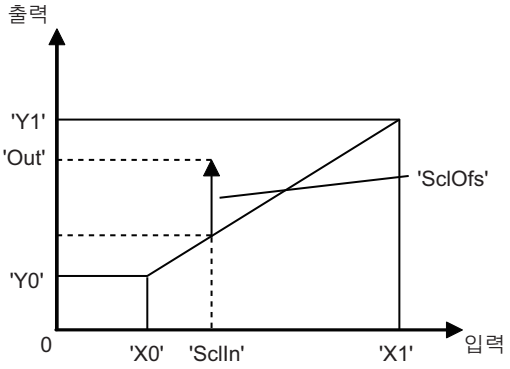
기능

입력값 'SclIn'의 값을 입력 범위에서 출력 범위로 변환합니다.
입력 범위는 입력 범위 하한값 'x0'과 입력 범위 상한값 'x1'로 지정합니다. 출력 범위는 출력 범위 하한값 'x0'과 출력 범위 상한값 'x1'로 지정합니다.

그리고 출력 범위로 변환한 값에 오프셋 값 'SclOfs'를 가산한 값을 출력값 'Out'으로 출력합니다. 'SclOfs'는 온도 제어의 오차 조정 등에 사용합니다.

변환식은 다음과 같습니다.

$$'Out' = \frac{'Y1'-'Y0'}{'X1'-'X0'} ('SclIn'-'X0')+'Y0'+'SclOfs'$$



기술 예

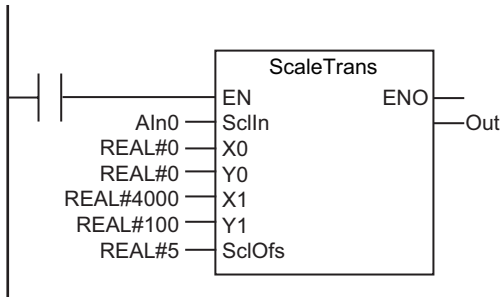
입력 범위가 0~4000, 출력 범위가 0~100%이고, 입력값 2500을 변환하는 경우의 기술 예가 아래에 나와 있습니다. 출력값에 더하는 오프셋 값은 5%입니다.

'SclIn'=REAL#2500, 'X0'=REAL#0, 'X1'=REAL#4000, 'Y0'=REAL#0,

'Y1'=REAL#100, 'SclOfs'=REAL#5입니다.

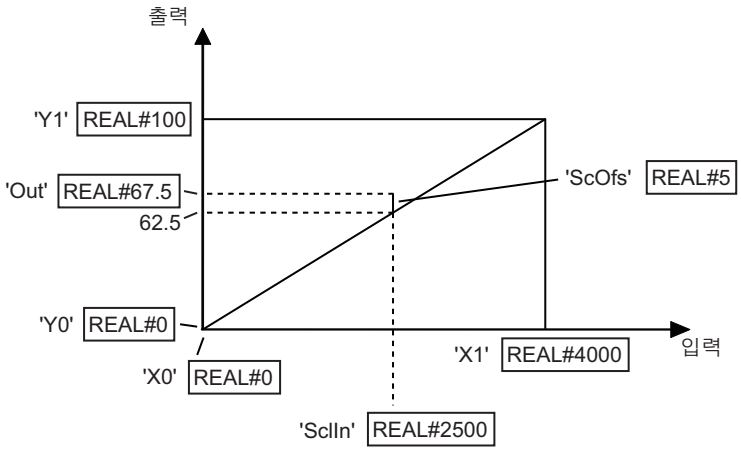
'Out'의 값은 REAL#67.5입니다.

LD



ST

```
Out :=ScaleTrans(AIn0,REAL#0,REAL#0,REAL#4000,
                 REAL#100,REAL#5);
```



입력 범위가 0~4000, 출력 범위가 0~100이고, 입력값 2500을 변환하면 62.5
그리고 오프셋 값 5를 더하면 'Out'=REAL#67.5입니다.

참고

- 'ScIIn'을 PIDAT 명령의 'PV' 및 'SP'의 값의 범위로 스케일 변환하는 경우는 'Y0', 'Y1'에 다음 파라메터를 전달하십시오.

변수	파라메터
'Y0'	PIDAT 명령의 입력 범위 하한 'InitSetParams.RngLowLmt'
'Y1'	PIDAT 명령의 입력 범위 상한 'InitSetParams.RngUpLmt'

- 'X1'<'X0'이나 'Y1'<'Y0'이 되는 설정도 가능합니다.

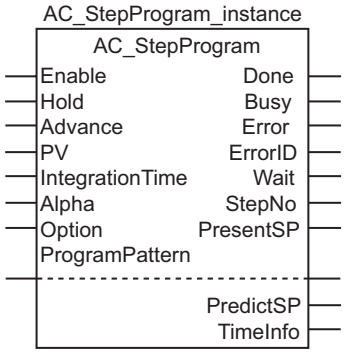
사용상의 주의 사항

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.

AC_StepProgram

지정된 프로그램 패턴에 따라 태스크 주기마다 현재 목표값과 예측 목표값을 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
AC_Step Program	스텝 프로그램	FB		AC_StepProgram_instance(Enable, Hold, Advance, PV, IntegrationTime, Alpha, Option, ProgramPattern, Done, Busy, Error, ErrorID, Wait, StepNo, PresentSP, PredictSP, TimeInfo);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	동작 명령	입력	TRUE : 실행 FALSE : 비실행	데이터 형식에 따름	–	FALSE
Hold	유지		TRUE : 유지 기능 실행 FALSE : 유지 기능 비실행			
Advance	전진		FALSE에서 TRUE로 기동할 때 실행할 스텝 번호 증가			
PV	측정값 (현재값)		측정값(현재값) PIDAT 명령의 'PV'와 동일.			0
IntegrationTime	적분 시간		적분 시간 PIDAT 명령의 ëIntegrationTimeí과 동일.	T#0.0000s~T#10000.0000s*1	s	T#0s
Alpha	2자유도 PID 파라메터 α		2자유도 PID 파라메터 α PIDAT 명령의 'OprSetParams.Alpha'와 동일.	0.00Å`1.00	–	0.65
Option	옵션		옵션 자세한 내용은 기능 설명 참조	–		–
Program Pattern[] 배열	프로그램 패턴	입출력	프로그램 패턴	–	–	–

	명칭	입/출력	내용	유효 범위	단위	초기값
Wait	대기 중	출력	TRUE : 대기 중 FALSE : 대기 중이 아님	데이터 형식에 따름	–	–
StepNo	현재 스텝 번호		현재 실행 중인 스텝 번호	0~99		
PresentSP	현재 목표값		산출한 현재 목표값	데이터 형식에 따름		
PredictSP	예측 목표값		산출한 예측 목표값			
TimeInfo	시간 정보		이 명령의 진행 상황을 감시하기 위한 시간 정보	–		

*1. 0.0001s 미만은 버림

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
Hold	○																			
Advance	○																			
PV														○						
Integration Time																○				
Alpha														○						
Option	구조체 _sAC_STEP_OPTION 자세한 내용은 기능 설명 참조																			
Program Pattern[] 배열*1*2*3	구조체 _sAC_STEP_DATA 자세한 내용은 기능 설명 참조 배열 전체 지정																			
Wait	○																			
StepNo						○														
PresentSP														○						
PredictSP														○						
TimeInfo	구조체 _sAC_STEP_TIME 자세한 내용은 기능 설명 참조																			

*1. 배열의 최대 요소 수는 100입니다.

*2. 1차원 배열입니다. 2차원 이상의 배열을 지정한 경우는 빌드 시에 이상 상태가 됩니다.

*3. 선두의 요소 번호는 0입니다.선두의 요소 번호가 0 이외의 배열을 지정한 경우는 빌드 시에 이상 상태가 됩니다.

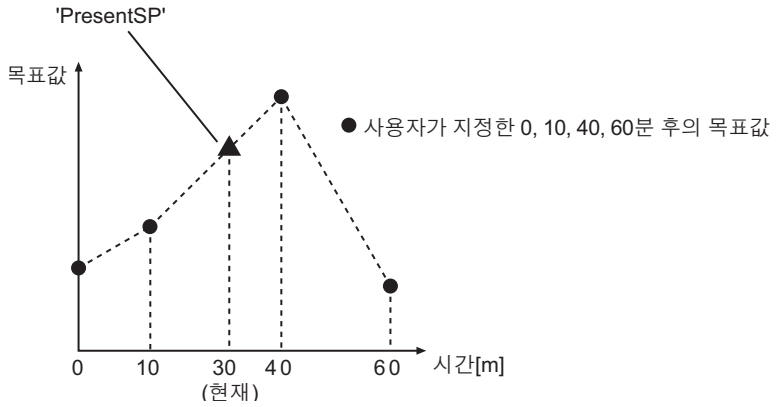
기능

PIDAT 명령과 연동시켜 온도 조절기 등의 조작량 제어를 실행하는 경우의 현재 목표값 'PresentSP'와 예측 목표값 'PredictSP'를 태스크 주기마다 산출합니다.

현재 목표값이란 현재 태스크 주기에서의 목표값을 말합니다. 예측 목표값이란 현재 목표값에 2자유도 PID 제어의 지연 보정을 실시한 목표값을 말합니다. 예측 목표값 'PredictSP'를 PIDAT 명령의 목표값 'SP'에 전달하면 PIDAT 명령에 의한 프로그램 제어의 추종성을 개선할 수 있습니다.

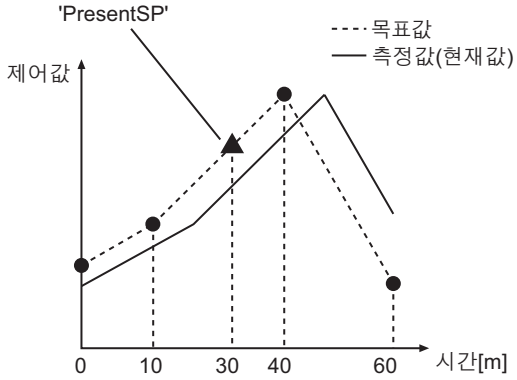
현재 목표값 'PresentSP'

현재 목표값 'PresentSP'란 현재 태스크 주기에서의 목표값을 말합니다. 예를 들어, 사용자가 제어 시작 0, 10, 40, 60분 후의 목표값을 아래 그림과 같이 설정했다고 가정합니다. 현재 시각은 제어 시작에서 30분 후라고 가정합니다. 이 명령은 제어 시작 10분 후와 40분 후의 목표값을 선형 보간하여 'PresentSP'를 산출합니다.

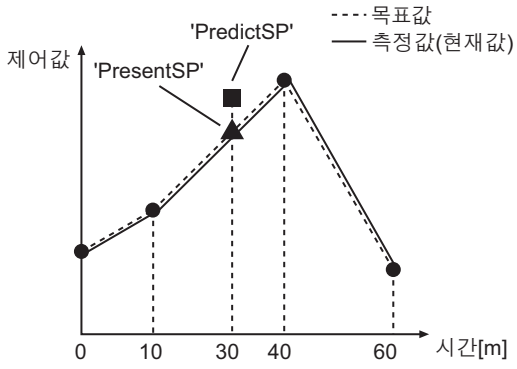


예측 목표값 'PredictSP'

예측 목표값 'PredictSP'란 'PresentSP'에 2자유도 PID 제어의 지연 보정을 실시한 목표값을 말합니다. 'PresentSP'를 보정 없이 PIDAT 명령의 'SP'에 전달하면 PIDAT 명령의 'PV'는 목표값과 일치하지 않습니다. 그 예는 아래 그림과 같습니다.



이 지연을 보정하는 것이 'PredictSP'입니다. 이 명령은 적분 시간 'IntegrationTime'이나 2자유도 PID 파라메터 α 'Alpha'의 값을 기초로 'PredictSP'를 산출합니다. 'PredictSP'를 PIDAT 명령의 'SP'에 전달하면 PIDAT 명령의 프로그램 제어 추종성을 개선할 수 있습니다.



구조체의 사양

옵션 'Option'의 데이터 형식은 구조체 _sAC_STEP_OPTION입니다.사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Option	옵션	옵션	_sAC_STEP_OPTION	–	–	–
StartAtPV	현재값 시작	TRUE: 현재값 시작 기능 실행 FALSE: 현재값 시작 기능 비실행	BOOL	데이터 형식에 따름	–	FALSE
StartStepNo	시작 스텝 번호	처리를 시작하는 스텝 번호	USINT	0~99		0
EndStepNo	종료 스텝 번호	처리를 종료하는 스텝 번호*1	USINT			
Reserved	예비	예비	ARRAY[0..31] OF BYTE	데이터 형식에 따름		32개의 요소가 모두 0

*1. 0을 설정한 경우, ProgramPattern[]의 최대 요소 번호를 종료 스텝 번호로 간주합니다.

프로그램 패턴 ProgramPattern[]의 요소의 데이터 형식은 구조체 _sAC_STEP_DATA입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Program Pattern	프로그램 패턴	프로그램 패턴	_sAC_STEP_DATA	–	–	–
ReachSP	도달 목표값	해당 스텝의 도달 목표값	REAL	데이터 형식에 따름		0
TimeWidth	시간 폭	해당 스텝의 시간 폭*1	TIME		s	T#0s
WaitWidth	대기 판정 폭	해당 스텝의 대기 판정 폭*2	REAL		–	0
WaitTime Limit	대기 시간 상한	해당 스텝의 대기 시간 상한*1*3	TIME		s	T#0s

*1. 분해능은 1태스크 주기입니다.
*2. 0 이하의 값을 설정한 경우, 0으로 간주합니다.
*3. 0 이하의 값을 설정한 경우, T#0s로 간주합니다.

시간 정보 'TimeInfo'의 데이터 형식은 구조체 _sAC_STEP_TIME입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
TimeInfo	시간 정보	시간 정보	_sAC_STEP_TIME	–	–	–
ProgramTime	프로그램 시간	스텝 0부터 'EndStepNo'까지의 'TimeWidth'의 합계	TIME	0 또는 양수값	s	T#0s
ElapseTime	경과 시간	이 명령의 실행 시작부터 현재까지의 시간*1	TIME			
Progress Time	진척 시간	이 명령의 실행 시작부터 현재까지의 시간*2	TIME			
LeftTime	남은 시간	현재부터 모든 처리 종료까지의 시간*2	TIME			
StepProgressTime	스텝 내 진척 시간	실행 중인 스텝의 시작부터 현재까지의 시간*2	TIME			
StepLeftTime	스텝 내의 남은 시간	현재부터 실행 중인 스텝의 종료까지의 시간*2	TIME			

*1. 대기 시간이 포함되며 유지 시간은 포함되지 않습니다.
*2. 대기 시간이나 유지 시간은 포함되지 않습니다.

각 변수의 의미

이 명령의 각 변수의 의미를 자세히 설명합니다.

● 'Enable'(동작 명령)

이 명령의 실행 조건입니다.
'Enable'의 값을 FALSE에서 TRUE로 변경하면 이 명령이 실행됩니다. 'Enable'의 값을 FALSE로 하면 이 명령의 실행이 정지됩니다.

● 'Hold'(유지)

유지 기능을 실행하기 위한 플래그입니다.
'Hold' 값을 TRUE로 하면 유지 기능이 실행됩니다.
유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

● 'Advance'(전진)

이 명령의 실행 중에 값이 FALSE에서 TRUE로 변화하면 처리가 다음 스텝으로 이동합니다.
전진 기능에 대한 자세한 내용은 뒤에서 다룹니다.

● 'PV'(측정값, 현재값)

제어 대상의 측정값입니다. PIDAT 명령의 'PV'와 동일한 의미입니다.

● 'IntegrationTime'(적분 시간)

PIDAT 명령의 'IntegrationTime'과 동일한 의미입니다.

● 'Alpha'(2자유도 PID 파라메터 α)

PIDAT 명령의 'OprSetParams.Alpha'와 동일한 의미입니다.

- **'StartAtPV'(현재값 시작)**

  현재값 시작 기능 실행 플래그입니다.
  'StartAtPV' 값을 TRUE로 하면 현재값 시작 기능이 실행됩니다.
  현재값 시작 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'StartStepNo'(시작 스텝 번호), 'EndStepNo'(종료 스텝 번호)**

  프로그램 패턴의 스텝 중 처리를 시작하는 스텝의 번호와 종료하는 스텝의 번호입니다.
  'EndStepNo'=0인 경우는 ProgramPattern[]의 최대 요소 번호를 종료 스텝 번호로 간주합니다.
  프로그램 패턴이나 스텝에 대한 자세한 내용은 뒤에서 다룹니다.

- **'ReachSP'(도달 목표값)**

  프로그램 패턴 내의 각 스텝 종료 시 도달해야 하는 목표값입니다.
  프로그램 패턴이나 스텝에 대한 자세한 내용은 뒤에서 다룹니다.

- **'TimeWidth'(시간 폭)**

  프로그램 패턴 내 각 스텝의 시간 폭입니다.
  프로그램 패턴이나 스텝에 대한 자세한 내용은 뒤에서 다룹니다.

- **'WaitWidth'(대기 판정 폭)**

  프로그램 패턴 내의 각 스텝에서 대기 기능을 실행할지 여부를 판정하기 위한 한계값입니다.
  대기 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'WaitTimeLimit'(대기 시간 상한)**

  프로그램 패턴 내 각 스텝에서 대기 기능을 실행한 경우의 대기 시간의 상한값입니다.
  'WaitTimeLimit'의 값이 T#0s인 경우는 대기 시간의 상한값이 무한대의 의미가 됩니다.
  대기 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'Wait'(대기 중)**

  대기 기능을 실행 중인지 여부를 나타내는 플래그입니다.
  'Wait'의 값이 TRUE이면 대기 기능 실행 중입니다.
  대기 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'StepNo'(현재 스텝 번호)**

  현재 실행 중인 스텝 번호입니다.
  프로그램 패턴이나 스텝에 대한 자세한 내용은 뒤에서 다룹니다.

- **'PresentSP'(현재 목표값)**

  앞에서 다룬 현재 목표값입니다.

- **'PredictSP'(예측 목표값)**

  앞에서 다룬 예측 목표값입니다.

- **'ProgramTime'(프로그램 시간)**

  프로그램 패턴의 스텝 0부터 'EndStepNo'까지의 'TimeWidth'의 합계 값입니다.
  프로그램 패턴이나 스텝에 대한 자세한 내용은 뒤에서 다룹니다.

- **'ElapseTime'(경과 시간)**

  이 명령의 실행 시작부터 현재까지의 시간입니다. 단, 대기 시간은 포함되지만, 유지 시간은 포함되지 않습니다.
  대기 기능이나 유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'ProgressTime'(진척 시간)**

  이 명령의 실행 시작부터 현재까지의 시간입니다. 단, 대기 시간이나 유지 시간은 포함되지 않습니다.
  대기 기능이나 유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'LeftTime'(남은 시간)**

  현재부터 모든 처리 종료까지의 시간입니다. 단, 대기 시간이나 유지 시간은 포함되지 않습니다.
  대기 기능이나 유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

- **'StepProgressTime'(스텝 내 진척 시간)**

  프로그램 패턴 내에서 실행 중인 스텝의 시작부터 현재까지의 시간입니다. 단, 대기 시간이나 유지 시간은 포함되지 않습니다.
  프로그램 패턴, 스텝, 대기 기능, 유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

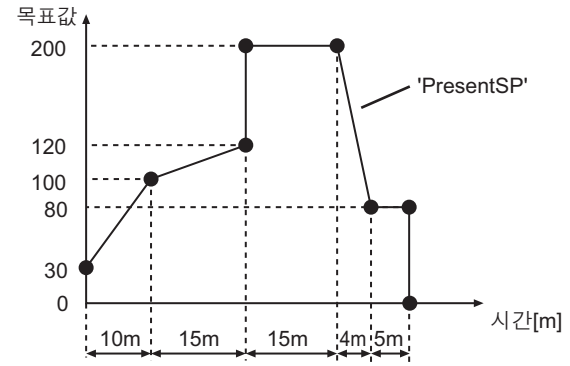
- **'StepLeftTime'(스텝 내 남은 시간)**

  프로그램 패턴 내에서 현재부터 실행 중인 스텝의 처리 종료까지의 시간입니다. 단, 대기 시간이나 유지 시간은 포함되지 않습니다.
  프로그램 패턴, 스텝, 대기 기능, 유지 기능에 대한 자세한 내용은 뒤에서 다룹니다.

프로그램 패턴

프로그램 패턴이란 이 명령 실행 시작부터 종료까지의 처리를 복수의 스텝으로 분할하여 스텝마다 도달 목표값과 시간 폭을 시계열로 나열한 것입니다. 프로그램 패턴은 데이터 형식 _sAC_STEP_DATA를 요소로 하는 배열 ProgramPattern[]으로 표현합니다. ProgramPattern[]의 각 요소가 각 스텝에 대응합니다. 아래에 프로그램 패턴의 예가 나와 있습니다. ProgramPattern[]의 각 요소의 'ReachSP'와 'TimeWidth'의 값이 아래 표와 같은 경우, 이 명령 실행 시작부터의 시간과 목표값의 관계는 다음 그림과 같습니다.

	ProgramPattern[]의 요소 번호							
	0	1	2	3	4	5	6	7
스텝 번호	0	1	2	3	4	5	6	7
'ReachSP'의 값	30	100	120	200	200	80	80	0
'TimeWidth'의 값	T#0s	T#10m	T#15m	T#0s	T#15m	T#4m	T#5m	T#0s



각 스텝의 목표값을 선형 보간하여 각 시점의 'PresentSP'의 값이 산출됩니다. 위 그림에서 실선으로 표시된 것이 'PresentSP'입니다. 태스크 주기마다 해당 시점의 'PresentSP' 값이 출력됩니다.

● 'TimeWidth'의 값과 스텝의 시간 폭의 관계

'TimeWidth'의 값과 스텝의 시간 폭의 관계는 다음과 같습니다.

'TimeWidth'의 값	스텝 번호	스텝의 시간 폭
T#0s	0	T#0s로 간주합니다.
	0 이외	1태스크 주기로 간주합니다.
양수값	–	'TimeWidth'의 값이 스텝의 시간 폭이 됩니다.
음수값	–	1태스크 주기로 간주합니다.

● 스텝의 시간 폭이 1태스크 주기 미만인 경우의 동작

스텝의 시간 폭의 분해능은 1태스크 주기입니다. 스텝의 시간 폭이 1태스크 주기 미만인 경우의 동작은 다음과 같습니다.

스텝 번호	스텝의 시간 폭	동작
0	T#0s	스텝 0의 'ReachSP'의 값이 'PresentSP'의 초기값이 됩니다. 실제 처리는 스텝 1부터 시작됩니다.
	T#0s 이외	해당 스텝의 처리를 1태스크 주기만 수행한 후 다음 스텝으로 처리가 이동합니다.
0 이외	–	

시작 스텝 번호 'StartStepNo'와 종료 스텝 번호 'EndStepNo'

프로그램 패턴 내 임의의 스텝을 처리의 시작 스텝 및 종료 스텝으로 설정할 수 있습니다. 시작 스텝 번호를 'StartStepNo'로, 종료 스텝 번호를 'EndStepNo'로 설정합니다.
예를 들어, 'StartStepNo'=3, 'EndStepNo'=6으로 하여 이 명령을 실행하면 처리가 스텝 3에서 시작되어 스텝 6에서 종료됩니다.

● 명령 실행 중 'StartStepNo', 'EndStepNo' 값의 변경

명령 실행 중에 'StartStepNo' 및 'EndStepNo'의 값을 변경할 수 있습니다. 값을 변경하면 다음과 같이 동작합니다.

변수	변경할 스텝 번호	동작
'StartStepNo'	–	변경 후의 'StartStepNo'의 선두부터 처리를 시작합니다.
'EndStepNo'	실행 중인 스텝 번호 이상의 값으로 변경	변경 후의 'EndStepNo'가 종료된 시점에서 처리가 종료됩니다.
	실행 중인 스텝 번호 미만의 값으로 변경	변경한 시점에서 처리가 종료됩니다. 'Done'의 값이 TRUE가 됩니다.

대기 기능

제어 대상의 지연으로 의해 'PV' 값이 실행 중인 스텝의 'TimeWidth'내에 'ReachSP'의 값에 도달하지 않는 경우가 있습니다. 이런 경우, 'TimeWidth'의 시간 폭을 초과하여, 실행 중인 스텝의 처리를 유지하는 기능을 대기 기능이라고 합니다.
대기 기능에 관련된 변수는 ProgramPattern[] 내의 대기 판정값 'WaitWidth'와 대기 시간 상한 'WaitTimeLimit' 및 대기 중 'Wait' 입니다.

● 대기 기능 실행 판정

실행 중인 스텝의 종료 시간에 'ReachSP'와 'PV'의 차가 'WaitWidth'를 초과한 경우 대기 기능이 실행됩니다.

● 대기 기능 종료 타이밍

대기 기능 실행 시작부터 'WaitTimeLimit' 이내의 시간에 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하이면, 해당 시점에서 대기 기능 실행을 종료하고 다음 스텝으로 처리가 이동합니다.
대기 기능 실행 시작부터 'WaitTimeLimit' 이내의 시간에 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하가 되지 않았으면, 'WaitTimeLimit'의 시간 경과 후 대기 기능 실행을 종료하고 다음 스텝으로 처리가 이동합니다. 단, 'WaitTimeLimit'의 값이 T#0s인 경우는 대기 시간 상한이 무한대가 됩니다. 따라서 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하가 될 때까지 시간 제한 없이 대기 기능을 계속 실행합니다.

● 대기 기능 실행의 감시

대기 기능을 실행 중인지 여부는 대기 중 'Wait'의 값을 감시하면 알 수 있습니다. 대기 기능 실행 중에는 'Wait'의 값이 TRUE가 됩니다. 대기 기능 실행이 종료되면 'Wait'의 값이 FALSE가 됩니다.

● 대기 기능 실행 중의 시간 측정

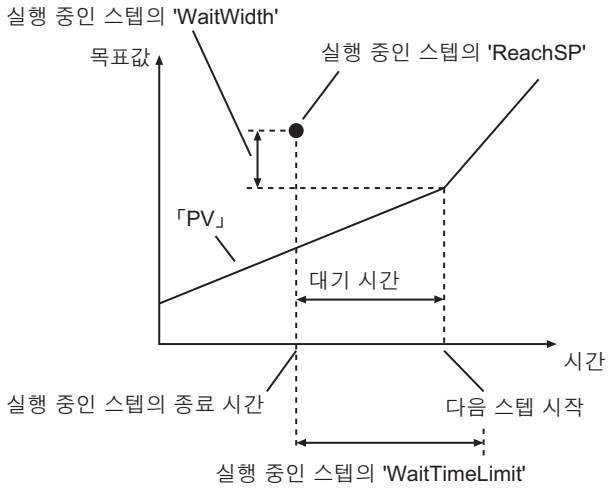
대기 기능 실행 중의 시간 관계 각 변수의 동작은 다음과 같습니다.

변수	동작
'ElapseTime'	시간 측정을 계속합니다.
'ProgressTime'	시간 측정을 정지하고 대기 기능 실행 시작 시점의 값을 유지합니다. 대기 기능 실행이 종료되면 해당 값에서 시간 측정을 재개합니다.
'LeftTime'	
'StepProgressTime'	실행 중인 스텝의 'TimeWidth' 값이 되고, 해당 값을 유지합니다.
'StepLeftTime'	0이 되고, 해당 값을 유지합니다.

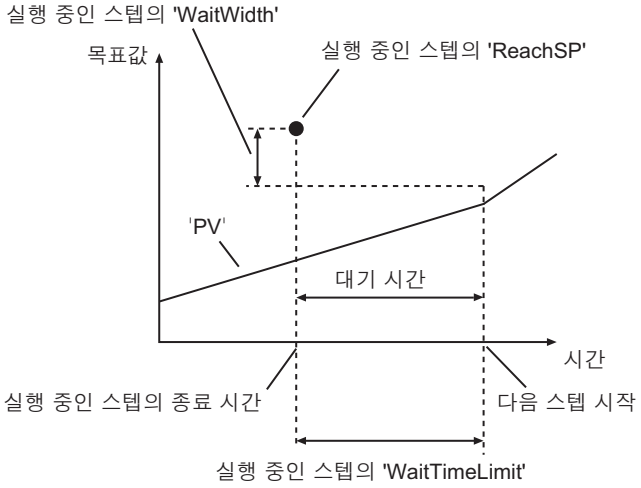
● 대기 기능 실행 중의 'PresentSP'와 'PredictSP'

대기 기능 실행 중의 'PresentSP'와 'PredictSP'는 모두 'ReachSP'의 값을 유지합니다.

아래에 'WaitTimeLimit' 이내의 시간에 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하가 된 경우의 'PV'의 그래프가 나와 있습니다. 실행 중인 스텝의 종료 시간이 와도 'ReachSP'와 'PV'의 차가 'WaitWidth'를 초과하므로 대기 기능을 실행합니다. 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하가 된 시점에서 다음 스텝의 처리를 시작합니다.



아래에 'WaitTimeLimit' 이내의 시간에 'ReachSP'와 'PV'의 차가 'WaitWidth' 이하가 되지 않은 경우의 'PV'의 그래프가 나와 있습니다. 'WaitTimeLimit'의 시간 경과 후 다음 스텝의 처리를 시작합니다.



유지 기능

유지 기능은 유지 'Hold'의 값을 TRUE로 하여 실행 중인 스텝의 처리를 강제로 유지시키는 기능입니다.
유지 기능 실행 중에는 시간 관계의 모든 변수의 시간 측정이 정지됩니다.
'Hold'의 값을 FALSE로 하면 시간 관계 변수의 시간 측정이 재개됩니다.

● 유지 기능 실행 중의 시간 측정

유지 기능 실행 중의 시간 관계 각 변수의 동작은 다음과 같습니다.

변수	동작
'ElapseTime'	시간 측정을 정지하고 유지 기능 실행 시작 시점의 값을 유지합니다. 유지 기능 실행이 종료되면 해당 값에서 시간 측정을 재개합니다.
'ProgressTime'	
'LeftTime'	
'StepProgressTime'	
'StepLeftTime'	

● 유지 기능 실행 중의 'PresentSP'와 'PredictSP'

유지 기능 실행 중의 'PresentSP'는 유지 기능 실행 시작 시점의 값을 유지합니다.
유지 기능 실행 중의 'PredictSP'는 'PresentSP'와 동일한 값이 됩니다.

● 대기 기능 실행 중의 유지 기능 실행

대기 기능 실행 중에 유지 기능을 실행하면 대기 기능의 실행이 일단 종료됩니다. 따라서 'Wait'의 값은 FALSE가 됩니다. 그 후, 유지 기능 실행을 종료한 시점에서 대기 기능 실행 판정이 다시 이루어집니다.

현재값 시작 기능

현재값 시작 기능은 'PV'의 현재값과 'PresentSP'의 값이 일치하는 시점부터 처리를 시작하는 기능입니다. 현재값 시작 'StartAtPV'의 값을 TRUE로 한 상태에서 'Enable'의 값을 TRUE로 하면 현재값 시작 기능이 실행됩니다.
현재값 시작 기능이 실행되면 다음 순서로 처리가 실행됩니다.

1 **'PV'의 현재값이 취득됩니다.**

2 **스텝 0부터 최종 스텝을 향하여 'PV'의 현재값과 'PresentSP'의 값이 최초로 일치하는 시각이 탐색됩니다.**
단, 'PresentSP'의 값이 스텝 0의 선두부터 상승하는 경우는 'PresentSP'의 값이 하강하기 직전까지만 탐색합니다. 동일하게, 'PresentSP'의 값이 스텝 0의 선두부터 하강하는 경우는 'PresentSP'의 값이 상승하기 직전까지만 탐색합니다.

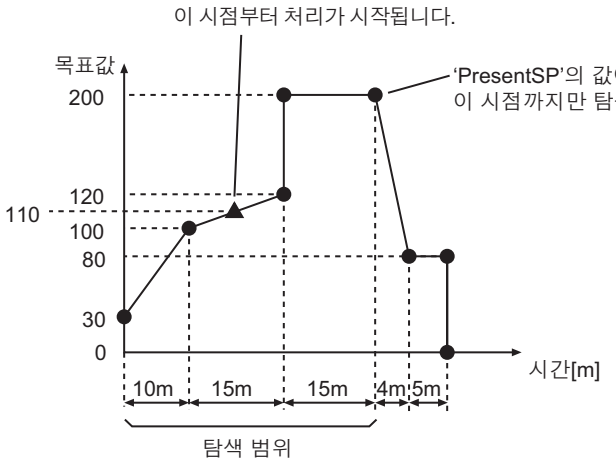
3 **위에서 탐색한 시점부터 처리가 시작됩니다.**

탐색 범위 중에 'PV'의 현재값과 'PresentSP'의 값이 일치하는 시각이 존재하지 않는 경우는 스텝 0부터 처리가 시작됩니다.

아래에 현재값 시작 기능 실행의 예가 나와 있습니다. ProgramPattern[]의 내용은 다음 표와 같습니다.

	ProgramPattern[]의 요소 번호							
	0	1	2	3	4	5	6	7
스텝 번호	0	1	2	3	4	5	6	7
'ReachSP'의 값	30	100	120	200	200	80	80	0
'TimeWidth'의 값	T#0s	T#10m	T#15m	T#0s	T#15m	T#4m	T#5m	T#0s

'PresentSP'의 값은 스텝 0의 값에서 상승 중입니다. 따라서 'PresentSP'의 값이 하강을 시작하고, 처리 시작 후 40m 시점까지만 탐색합니다.
이 명령 실행 시작 시의 'PV'의 값이 110이었다고 가정합니다. 그 경우, 아래 그림에 나온 것처럼 'PresentSP'=110이 되는 시점부터 처리가 시작됩니다.



● 현재값 시작 기능 실행 시의 시간 측정

현재값 시작 기능 실행 시의 시간 관계 각 변수의 동작은 다음과 같습니다.

변수	동작
'ElapseTime'	0이 됩니다.
'ProgressTime'	스텝 0부터 탐색한 시점까지의 시간입니다.
'LeftTime'	현재부터 'EndStepNo' 종료까지의 시간입니다.
'StepProgressTime'	실행 중인 스텝의 선두부터 탐색한 시점까지의 시간입니다.
'StepLeftTime'	현재부터 실행 중인 스텝의 종료까지의 시간입니다.

● 명령 실행 중의 'StartAtPV' 값 변경

이 명령의 실행 중에 'StartAtPV'의 값을 변경해도 무시됩니다.

전진 기능

이 명령의 실행 중에 'Advance'의 값을 FALSE에서 TRUE로 변경하면 처리가 다음 스텝의 선두로 이동합니다.

● 전진 기능 실행 시의 시간 측정

전진 기능 실행 시의 시간 관계 각 변수의 동작은 다음과 같습니다.

변수	동작
'ElapseTime'	시간 측정을 계속합니다.
'ProgressTime'	스텝 0부터 실행 중인 스텝까지의 'TimeWidth'의 합계입니다.
'LeftTime'	다음 스텝부터 'EndStepNo' 종료까지의 시간입니다.
'StepProgressTime'	처리가 다음 스텝의 선두로 이동하므로 0이 됩니다.
'StepLeftTime'	다음 스텝의 'TimeWidth'의 값이 됩니다.

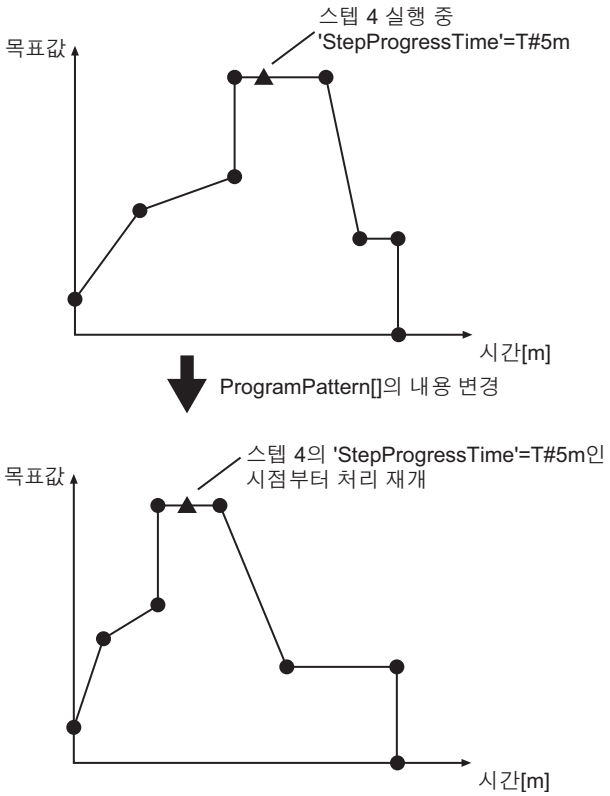
● 'StartStepNo'의 값 변경과 전진 기능의 병용

'StartStepNo'의 값을 변경하는 동시에 'Advance'의 값을 FALSE에서 TRUE로 변경한 경우는 'StartStepNo' 값의 변경이 우선됩니다. 따라서 처리는 'StartStepNo'의 선두로 이동합니다.

명령 실행 중의 프로그램 패턴 변경

명령 실행 중에 ProgramPattern[]의 내용을 변경할 수 있습니다. ProgramPattern[]의 내용을 변경하면 'PresentSP'가 다시 산출됩니다. 처리는 변경 전에 실행했던 스텝 번호의 'StepProgressTime'의 시점에서 재개됩니다.이전 스텝의 내용을 변경해도 상관없습니다.

예를 들어, 스텝 4 실행 중에 ProgramPattern[]의 내용을 변경했다고 가정합니다. 변경 전의 'StepProgressTime'의 값은 T#5m이었다고 가정합니다. 변경 후에는 스텝 4의 'StepProgressTime'=T#5m의 시점에서 처리가 재개됩니다.



변경 후 해당 스텝의 'TimeWidth'의 값이 'StepProgressTime'의 값보다 작은 경우는 다음 스텝의 선두부터 처리가 재개됩니다.

● 명령 실행 중에 프로그램 패턴을 변경한 경우의 시간 측정

명령 실행 중에 프로그램 패턴을 변경한 경우의 시간 관계 각 변수의 동작은 다음과 같습니다.

변수	동작
'ProgramTime'	변경 후 스텝 0부터 'EndStepNo'까지의 'TimeWidth'의 합계가 됩니다.
'ElapseTime'	시간 측정을 계속합니다.
'ProgressTime'	변경 후 스텝 0부터 실행 중인 스텝 번호의 1개 이전 스텝까지의 'TimeWidth'의 합계에 'StepProgressTime'을 가산한 값이 됩니다.
'LeftTime'	변경 후 현재부터 'EndStepNo' 종료까지의 시간이 됩니다.
'StepProgressTime'	변경 전의 값부터 시간 측정을 계속합니다.
'StepLeftTime'	변경 후, 실행 중인 스텝의 현재부터 스텝 종료까지의 시간입니다.

● 대기 기능 실행 중의 프로그램 패턴 변경

대기 기능 실행 중에 프로그램 패턴을 변경한 경우, 다시 산출된 'PresentSP'에 대해 대기 기능 실행 판정이 다시 이루어집니다.

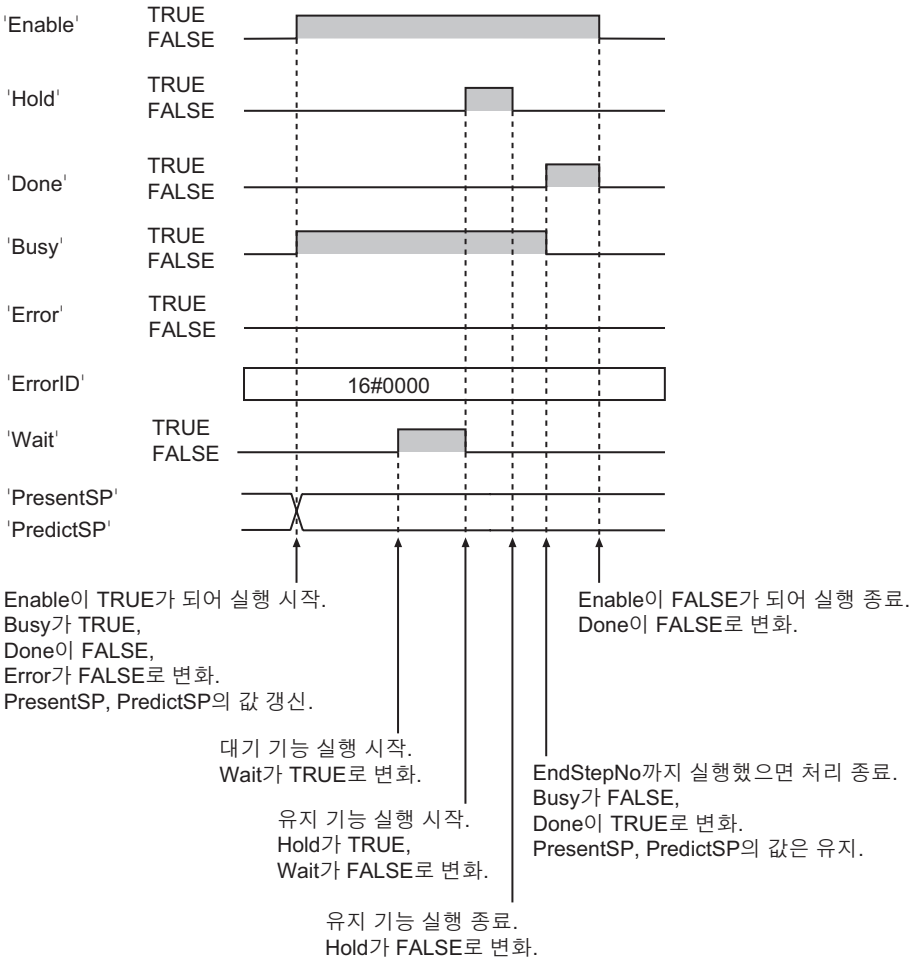
단, 'StepProgressTime'의 값이 변경 후의 'WaitTimeLimit'의 값보다 큰 경우는 즉시 대기 기능 실행을 종료하고 다음 스텝으로 처리가 이동합니다.

● 유지 기능 실행 중의 프로그램 패턴 변경

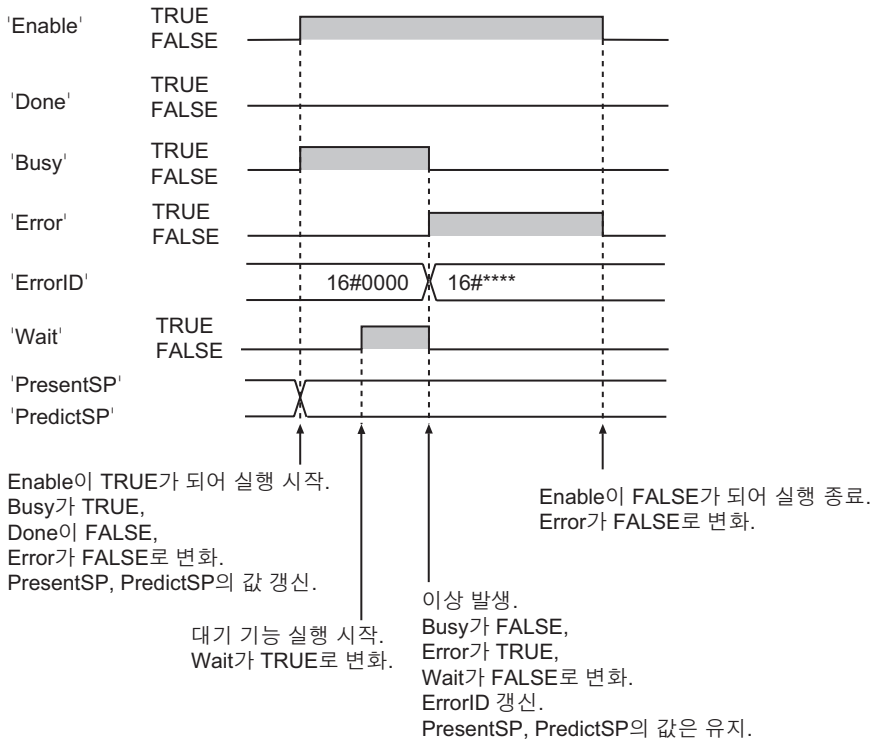
유지 기능 실행 중에 프로그램 패턴을 변경한 경우, 다시 산출된 'PresentSP'에 대해 유지 기능이 계속됩니다.

타이밍 차트

아래에 정상일 때의 타이밍 차트가 나와 있습니다.



아래에 이상 발생 시의 타이밍 차트가 나와 있습니다.



사용상의 주의 사항

- 다음과 같은 경우, 이상이 발생하며 'Error'는 TRUE가 되고 'ErrorID'에는 이상 ID가 대입됩니다.

이상 내용	'ErrorID'의 값
'IntegrationTime', 'Alpha', 'StartStepNo', 'EndStepNo' 중 하나의 값이 유효 범위 밖	16#0400
ProgramPattern[] 배열의 최종 요소 번호가 99를 초과	16#0416

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상, Sysmac Studio Ver.1.07 이상에서 사용할 수 있습니다.

샘플 프로그램

AC_StepProgram 명령의 스텝마다 최적의 PID 파라메터를 사용하여 온도 제어를 실행하는 샘플 프로그램입니다.

처리 내용

이 샘플 프로그램은 다음 2가지 처리를 수행합니다.

- 스텝마다 최적의 PID 파라메터 산출
- 프로그램 패턴에 따른 온도 제어

각각의 처리에 대해 설명합니다.

● 스텝마다 최적의 PID 파라메터 산출

프로그램 패턴에 따른 온도 제어를 수행하기 전에 미리 각 스텝에 대한 최적의 PID 파라메터를 산출해 둡니다. PID 파라메터의 산출에는 PIDAT 명령의 자동 조정 기능을 사용합니다.
산출된 PID 파라메터는 스텝 번호를 첨자로 사용하는 구조체 형식 배열 PIDbank[]에 저장됩니다. PIDbank[]의 요소의 각 멤버는 비례대, 적분 시간, 미분 시간의 값을 나타냅니다.

처리 순서는 다음과 같습니다.

1 **사용자가 AC_StepProgram 명령의 동작 명령 'ACSP_Enable'의 값을 TRUE로 합니다.**
AC_StepProgram 명령이 기동하고 현재 스텝 번호 'StepNo'의 값이 0이 됩니다.

2 **사용자가 PIDAT 명령의 실행 조건 'Run'의 값을 TRUE로 합니다.**
PIDAT 명령이 기동합니다.

3 **사용자가 자동 조정 실행 조건 'StartAT'의 값을 FALSE에서 TRUE로 합니다.**
AC_StepProgram 명령의 유지 'Hold'의 값이 TRUE가 되고 유지 기능이 실행됩니다.
PIDAT 명령의 자동 조정 기능이 실행되고 스텝 0에 대한 최적의 PID 파라메터가 산출됩니다.

4 **자동 조정이 종료됩니다.**
PIDAT 명령의 자동 조정 정상 종료 'ATDone'의 값이 TRUE가 됩니다.
산출된 PID 파라메터가 PIDbank[0]에 저장됩니다.

5 **사용자가 AC_StepProgram 명령의 'Hold'의 값을 FALSE로 합니다.**
AC_StepProgram 명령의 유지 기능이 해제됩니다.
잠시 시간이 지나면 스텝이 다음으로 이동하고 'StepNo'의 값이 1이 됩니다.

6 **사용자가 스텝 수만큼 상기 3~5를 반복합니다.**
모든 스텝에 대해 최적의 PID 파라메터가 PIDbank[]에 저장됩니다.

● 프로그램 패턴에 따른 온도 제어

각 스텝에 대한 최적의 PID 파라메터를 사용하여 프로그램 패턴에 따른 온도 제어를 수행합니다.
처리 순서는 다음과 같습니다.

1 **사용자가 AC_StepProgram 명령의 동작 명령 'ACSP_Enable'의 값을 TRUE로 합니다.**
AC_StepProgram 명령이 기동하고 스텝 번호 'StepNo'의 값이 0이 됩니다.

2 **사용자가 PIDAT 명령의 실행 조건 'Run'의 값을 TRUE로 합니다.**
PIDAT 명령이 기동합니다.

3 **태스크 주기마다 PIDAT 명령의 조작량 'MV'가 출력됩니다.**

4 **TimeProportionalOut 명령에 의해 'MV'의 값에 대한 시분할 비례 출력이 실행됩니다.**

5 **잠시 시간이 지나면 스텝이 다음으로 이동합니다.**

6 **상기 3~5가 종료 스텝까지 반복됩니다.**

Sysmac Studio의 설정

이 샘플 프로그램을 실행하기 위해서는 Sysmac Studio에서 네트워크 구성, I/O 맵, 데이터 형식 정의의 설정을 해두어야 합니다.

● 네트워크 설정

네트워크 구성은 다음과 같습니다. EtherCAT의 노드 주소 1에 아래 구성의 슬레이브 터미널이 연결되어 있습니다. 각각에는 다음과 같은 장치명을 할당합니다.

유니트 번호	형식	유니트	장치명
0	NX-ECC201	EtherCAT 커플러 유니트	E001
1	NX-TS2101	온도 입력 유니트	N1
2	NX-OD3121	디지털 출력 유니트	N2

● I/O 맵

I/O 맵 설정은 다음과 같습니다.

위치	포트	설명	R/W	데이터 형식	변수	변수 종류
Unit1	Ch1 Measured Value REAL*1	채널의 측정값(REAL 형식)	R	REAL	N1_Ch1_Measured_Value_REAL	글로벌 변수
Unit1	Ch2 Measured Value REAL*2	채널의 측정값(REAL 형식)	R	REAL	N1_Ch2_Measured_Value_REAL	글로벌 변수
Unit2	Output Bit 00	출력 접점 00	W	BOOL	N2_Output_Bit_00	글로벌 변수
Unit2	Output Bit 01	출력 접점 01	W	BOOL	N2_Output_Bit_01	글로벌 변수
Unit2	Output Bit 02	출력 접점 02	W	BOOL	N2_Output_Bit_02	글로벌 변수
Unit2	Output Bit 03	출력 접점 03	W	BOOL	N2_Output_Bit_03	글로벌 변수

*1. 온도 입력 유니트 NX-TS2101의 I/O 항목에 0x6003:01(Ch1 Measured Value REAL)을 추가해야 합니다.
*2. 온도 입력 유니트 NX-TS2101의 I/O 항목에 0x6003:02(Ch2 Measured Value REAL)를 추가해야 합니다.

● 데이터 형식 정의

다음과 같이 구조체 형식 sPID_BANK를 정의합니다.

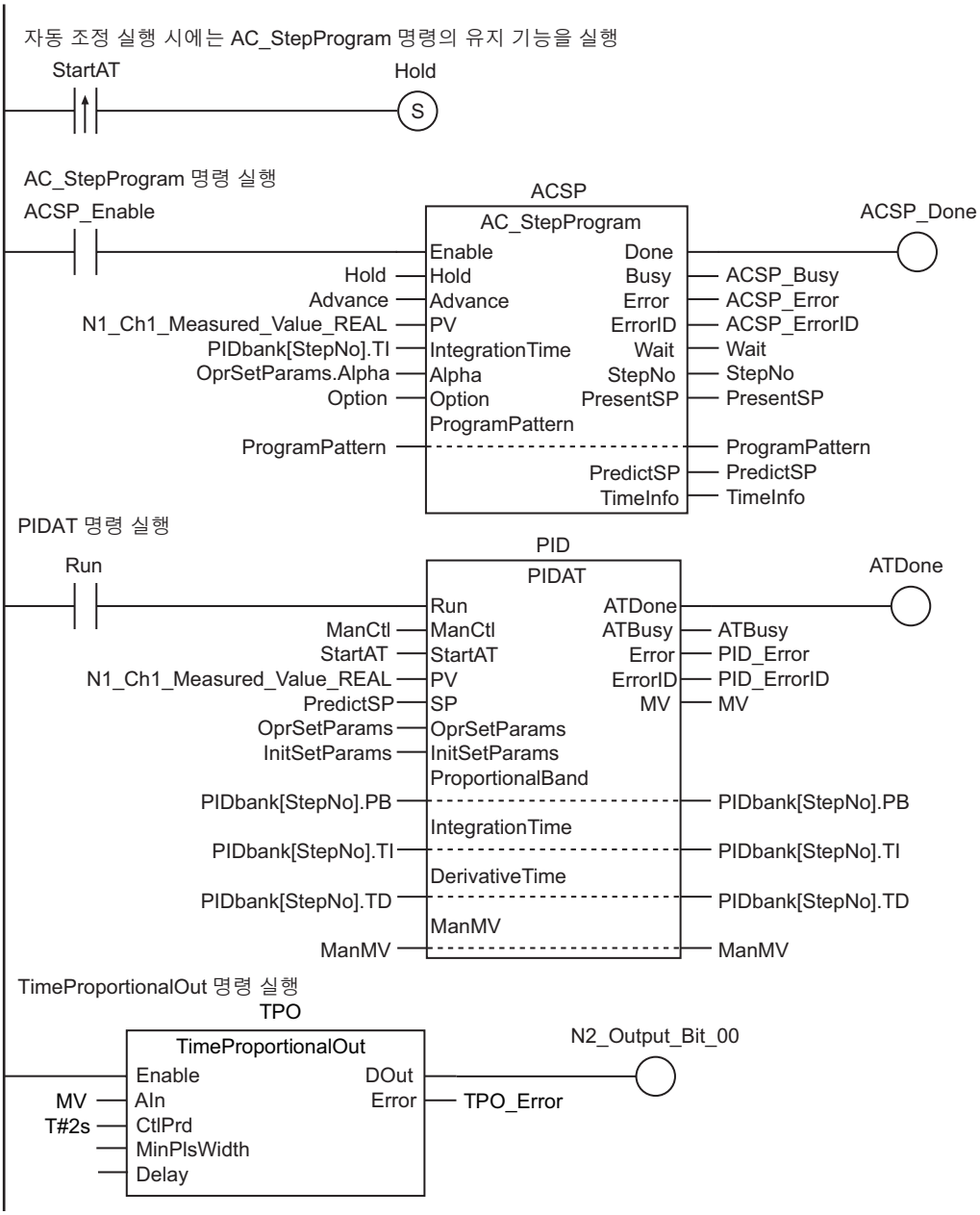
구조체 형식		명칭	데이터 형식	코멘트
	▼	sPID_BANK	STRUCT	PID 파라메터 구조체
		PB	REAL	비례대
		TI	TIME	적분 시간
		TD	TIME	미분 시간

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	ACSP_Enable	BOOL	FALSE	AC_StepProgram의 동작 명령
	Hold	BOOL	FALSE	유지
	Advance	BOOL	FALSE	전진
	Option	_sAC_STEP _OPTION	(StartAtPV:=FALSE, StartStepNo:=0, EndStepNo:=7, Reserved:=[32(16#0)])	옵션
	ProgramPattern	ARRAY[0..7] OF _sAC_STEP_DATA	[(ReachSP:=30.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=100.0, TimeWidth:=T#10m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=120.0, TimeWidth:=T#15m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=150.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=150.0, TimeWidth:=T#15m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=80.0, TimeWidth:=T#4m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=80.0, TimeWidth:=T#5m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=10.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m)]	프로그램 패턴
	ACSP_Busy	BOOL	FALSE	AC_StepProgram 실행 중
	ACSP_Error	BOOL	FALSE	AC_StepProgram 에러
	ACSP_ErrorID	WORD	WORD#16#0	AC_StepProgram 에러 코드
	Wait	BOOL	FALSE	대기 중
	StepNo	USINT	0	현재 스텝 번호
	PresentSP	REAL	0.0	현재 목표값

내부 변수	명칭	데이터 형식	초기값	코멘트
	PredictSP	REAL	0.0	예측 목표값
	TimeInfo	_sAC_STEP_TIME	(ProgramTime:=T#0s, ElapseTime:=T#0s, ProgressTime:=T#0s, LeftTime:=T#0s, StepProgressTime:=T#0s, StepLeftTime:=T#0s)	시간 정보
	ACSP_Done	BOOL	FALSE	AC_StepProgram 실행 완료
	Run	BOOL	FALSE	PIDAT 명령 실행 조건
	ManCtl	BOOL	FALSE	매뉴얼/자동 전환
	StartAT	BOOL	FALSE	자동 조정 실행 조건
	OprSetParams	_sOPR_SET _PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=FALSE, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	운전 중 설정 파라메터
	InitSetParams	_sINIT_SET _PARAMS	(SampTime:=T#250ms, RngLowLmt:=-200.0, RngUpLmt:=1300.0, DirOpr:=FALSE)	초기 설정 파라메터
	ManMV	REAL	0.0	수동 조작량
	ATBusy	BOOL	FALSE	자동 조정 실행 중
	PID_ErrorID	WORD	WORD#16#0	PIDAT 에러 코드
	PID_Error	BOOL	FALSE	PIDAT 에러
	MV	REAL	0.0	조작량
	ATDone	BOOL	FALSE	자동 조정 정상 종료
	TPO_Error	BOOL	FALSE	TimeProportionalOut 에러
	PIDbank	ARRAY[0..7] OF sPID_BANK	[8((PB:=10, TI:=T#233s, TD:=T#60s))]	최적 PID 파라메터 저장 배열
	ACSP	AC_StepProgram		
	PID	PIDAT		
	TPO	TimeProportionalOut		

외부 변수	명칭	데이터 형식	상수	코멘트
	N1_Ch1_Measured_Value_REAL	REAL	☐	채널의 측정값(REAL 형식)
	N2_Output_Bit_00	BOOL	☐	출력 접점


자동 조정 실행 시에는 AC_StepProgram 명령의 유지 기능을 실행
StartAT
Hold
S
AC_StepProgram 명령 실행
ACSP
ACSP_Enable
AC_StepProgram
ACSP_Done
Enable
Done
Hold
Hold
Busy
ACSP_Busy
Advance
Advance
Error
ACSP_Error
N1_Ch1_Measured_Value_REAL
PV
ErrorID
ACSP_ErrorID
PIDbank[StepNo].TI
IntegrationTime
Wait
Wait
OprSetParams.Alpha
Alpha
StepNo
StepNo
Option
Option
PresentSP
PresentSP
ProgramPattern
ProgramPattern
ProgramPattern
PredictSP
PredictSP
TimeInfo
TimeInfo
PIDAT 명령 실행
PID
Run
PIDAT
ATDone
Run
ATDone
ManCtl
ManCtl
ATBusy
ATBusy
StartAT
StartAT
Error
PID_Error
N1_Ch1_Measured_Value_REAL
PV
ErrorID
PID_ErrorID
PredictSP
SP
MV
MV
OprSetParams
OprSetParams
InitSetParams
InitSetParams
ProportionalBand
PIDbank[StepNo].PB
PIDbank[StepNo].PB
IntegrationTime
PIDbank[StepNo].TI
PIDbank[StepNo].TI
DerivativeTime
PIDbank[StepNo].TD
PIDbank[StepNo].TD
ManMV
ManMV
ManMV
TimeProportionalOut 명령 실행
TPO
TimeProportionalOut
N2_Output_Bit_00
Enable
DOut
MV
AIn
Error
TPO_Error
T#2s
CtlPrd
MinPlsWidth
Delay

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	ACSP_Enable	BOOL	FALSE	AC_StepProgram의 동작 명령
	Hold	BOOL	FALSE	유지
	Advance	BOOL	FALSE	전진
	Option	_sAC_STEP _OPTION	(StartAtPV:=FALSE, StartStepNo:=0, EndStepNo:=7, Reserved:=[32(16#0)])	옵션
	ProgramPattern	ARRAY[0..7] OF _sAC_STEP_DATA	[(ReachSP:=30.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=100.0, TimeWidth:=T#10m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=120.0, TimeWidth:=T#15m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=150.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=150.0, TimeWidth:=T#15m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=80.0, TimeWidth:=T#4m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=80.0, TimeWidth:=T#5m, WaitWidth:=3.0, WaitTimeLimit:=T#1m), (ReachSP:=10.0, TimeWidth:=T#0s, WaitWidth:=3.0, WaitTimeLimit:=T#1m)]	프로그램 패턴
	ACSP_Busy	BOOL	FALSE	AC_StepProgram 실행 중
	ACSP_Error	BOOL	FALSE	AC_StepProgram 에러
	ACSP_ErrorID	WORD	WORD#16#0	AC_StepProgram 에러 코드
	Wait	BOOL	FALSE	대기 중
	StepNo	USINT	0	현재 스텝 번호
	PresentSP	REAL	0.0	현재 목표값

내부 변수	명칭	데이터 형식	초기값	코멘트
	PredictSP	REAL	0.0	예측 목표값
	TimeInfo	_sAC_STEP_TIME	(ProgramTime:=T#0s, ElapseTime:=T#0s, ProgressTime:=T#0s, LeftTime:=T#0s, StepProgressTime:=T#0s, StepLeftTime:=T#0s)	시간 정보
	ACSP_Done	BOOL	FALSE	AC_StepProgram 실행 완료
	Run	BOOL	FALSE	PIDAT 명령 실행 조건
	ManCtl	BOOL	FALSE	매뉴얼/자동 전환
	StartAT	BOOL	FALSE	자동 조정 실행 조건
	PreStartAT	BOOL	TRUE	이전 태스크 주기의 자동 조정 실행 조건
	OprSetParams	_sOPR_SET _PARAMS	(MVLowLmt:=0.0, MVUpLmt:=100.0, ManResetVal:=0.0, MVTrackSw:=FALSE, MVTrackVal:=0.0, StopMV:=0.0, ErrorMV:=0.0, Alpha:=0.65, ATCalcGain:=1.0, ATHystrs:=0.2)	운전 중 설정 파라메터
	InitSetParams	_sINIT_SET _PARAMS	(SampTime:=T#250ms, RngLowLmt:=-200.0, RngUpLmt:=1300.0, DirOpr:=FALSE)	초기 설정 파라메터
	ManMV	REAL	0.0	수동 조작량
	ATBusy	BOOL	FALSE	자동 조정 실행 중
	PID_ErrorID	WORD	WORD#16#0	PIDAT 에러 코드
	PID_Error	BOOL	FALSE	PIDAT 에러
	MV	REAL	0.0	조작량
	ATDone	BOOL	FALSE	자동 조정 정상 종료
	TPO_Error	BOOL	FALSE	TimeProportionalOut 에러
	PIDbank	ARRAY[0..7] OF sPID_BANK	[8((PB:=10, TI:=T#233s, TD:=T#60s))]	최적 PID 파라메터 저장 배열
	TPO_Enable	BOOL	FALSE	TimeProportionalOut의 동작 명령
	MinPlsWidth	REAL	0.0	최소 펄스 폭
	Delay	REAL	0.0	딜레이
	ACSP	AC_StepProgram		
	PID	PIDAT		
	TPO	TimeProportionalOut		

외부 변수	명칭	데이터 형식	상수	코멘트
	N1_Ch1_Measured_Value_REAL	REAL	┌┐	채널의 측정값(REAL 형식)
	N2_Output_Bit_00	BOOL	┌┐	출력 접점

TPO_Enable := TRUE;

```
// 자동 조정 실행 시에는 AC_StepProgram 명령의 유지 기능을 실행
IF StartAT AND PreStartAT=FALSE THEN
    Hold := TRUE;
END_IF;
PreStartAT := StartAT;

// AC_StepProgram 명령 실행
IF ACSP_Enable THEN
   ACSP(Enable :=ACSP_Enable,
        Hold            :=Hold,
        Advance         :=Advance,
        PV              :=N1_Ch1_Measured_Value_REAL,
        IntegrationTime :=PIDbank[StepNo].TI,
        Alpha           :=OprSetParams.Alpha,
        Option          :=Option,
        ProgramPattern  :=ProgramPattern,
        Done            =>ACSP_Done,
        Busy            =>ACSP_Busy,
        Error           =>ACSP_Error,
        ErrorID         =>ACSP_ErrorID,
        Wait            =>Wait,
        StepNo          =>StepNo,
        PresentSP       =>PresentSP,
        PredictSP       =>PredictSP,
        TimeInfo        =>TimeInfo);
END_IF;

// PIDAT 명령 실행
IF Run THEN
   PID(Run              :=Run,
        ManCtl          :=ManCtl,
        StartAT         :=StartAT,
        PV              :=N1_Ch1_Measured_Value_REAL,
        SP              :=PredictSP,
        OprSetParams    :=OprSetParams,
        InitSetParams   :=InitSetParams,
        ProportionalBand :=PIDbank[StepNo].PB,
        IntegrationTime :=PIDbank[StepNo].TI,
        DerivativeTime  :=PIDbank[StepNo].TD,
        ManMV           :=ManMV,
        ATDone          =>ATDone,
        ATBusy          =>ATBusy,
        Error           =>PID_Error,
        ErrorID         =>PID_ErrorID,
        MV=>MV);
END_IF;

// TimeProportionalOut 명령 실행
TPO(Enable          :=TPO_Enable,
        AIn         :=MV,
        CtlPrd      :=T#2s,
        MinPlsWidth :=MinPlsWidth,
        Delay       :=Delay,
        DOut        =>N2_Output_Bit_00,
        Error       =>TPO_Error);
```

시스템 제어 명령

명령	명칭	페이지
TraceSamp	데이터 트레이스 샘플링	2-738
TraceTrig	데이터 트레이스 트리거 조건 성립	2-741
GetTraceStatus	데이터 트레이스 Status 읽어오기	2-744
SetAlarm	사용자 이상 발생	2-747
ResetAlarm	사용자 이상 해제	2-752
GetAlarm	사용자 이상 Status 취득	2-754
ResetPLCError	PLC 이상 해제	2-756
GetPLCError	PLC 이상 Status 취득	2-760
ResetCJBError	I/O 버스 이상 해제	2-762
GetCJBError	I/O 버스 이상 Status 취득	2-764
GetEIPError	EtherNet/IP 이상 Status 취득	2-766
ResetMCError	모션 제어 이상 해제	2-768
GetMCError	모션 제어 이상 Status 취득	2-773
ResetECError	EtherCAT 이상 해제	2-775
GetECError	EtherCAT 이상 Status 취득	2-777
SetInfo	사용자 정보 발생	2-780
ResetUnit	유니트 재시작	2-782
GetNTPStatus	NTP Status 읽어오기	2-786
RestartNXUnit	NX 유니트 재시작	2-788
NX_ChangeWriteMode	NX 유니트 쓰기 모드 변경	2-793
NX_SaveParam	NX 유니트 파라메터 저장	2-797

TraceSamp

데이터 트레이스 샘플링을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
TraceSamp	데이터트레이스 샘플링	FUN	(@)TraceSamp EN ENO TraceNo Out Point	TraceSamp(TraceNo, Point);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
TraceNo	트레이스 번호	입력	트레이스 번호	(*)	–	0
Point	샘플링 포인트 번호		샘플링 포인트 번호	데이터 형식에 따름		
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* CPU 유니트의 형식이 NJ501-□□□□인 경우는 0~3, NJ301-□□□□인 경우는 0~1

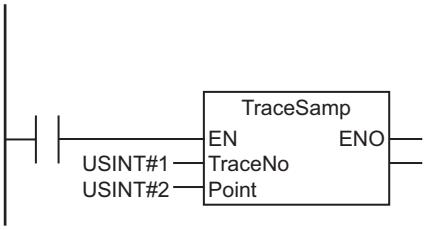
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
TraceNo						○														
Point						○														
Out	○																			

기능

데이터 트레이스 샘플링을 실행합니다.
Sysmac Studio를 통해 지정된 샘플링 설정에 따라, 샘플링 대상의 모든 변수의 현재값을 읽어와서 트레이스 번호 'TraceNo'와 샘플링 포인트 번호 'Point'의 값을 부가해서 트레이스 메모리에 저장합니다.
단, 이 명령이 실행되는 것은 데이터 트레이스 실행 중이며 Sysmac Studio에 의한 트레이스의 샘플링 타이밍 설정이 "샘플링 명령"인 경우뿐입니다.

기술 예가 아래에 나와 있습니다. 트레이스 번호 1, 샘플링 포인트 번호 2의 값을 부가해서 샘플링 대상의 모든 변수의 현재값을 트레이스 메모리에 저장합니다.

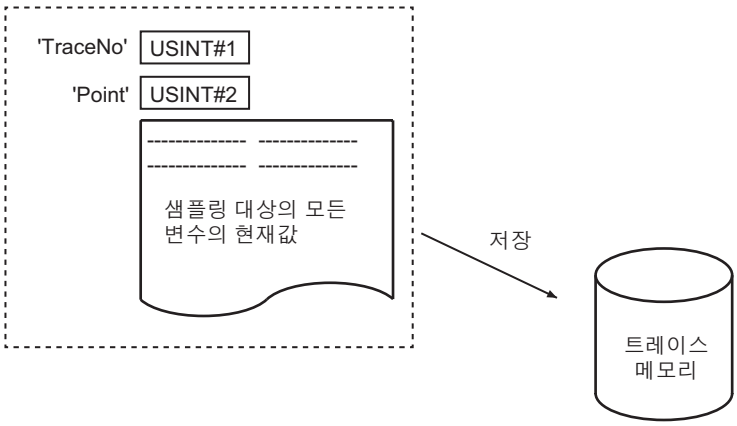
LD



ST

```
TraceSamp(USINT#1, USINT#2);
```

샘플링 대상의 모든 변수의 현재값을 읽어와서 트레이스 번호 'TraceNo'와 샘플링 포인트 번호 'Point'를 부가해서 트레이스 메모리에 저장합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
(*1)	트레이스 정보	(*2)	트레이스 정보 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

*1. CPU 유니트의 형식이 NJ501-□□□□인 경우는 변수명 _PLC_TraceSta[0..3], NJ301-□□□□인 경우는 변수명 _PLC_TraceSta[0..1]

*2. _sTRACE_STA[]

참고

- 데이터 트레이스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.
- 트레이스란, 지정한 조건일 때 지정한 변수의 값을 샘플링하는 기능입니다. 각 조건은 Sysmac Studio로 지정합니다.
- 이 명령은 프로그램 상에 여러 개 배치할 수 있습니다. 조건에 맞게 샘플링을 수행하는 프로그램을 기술할 수 있습니다.
- 'Point'를 적절히 설정하면, Sysmac Studio의 트레이스 창 화면에서 어느 TraceSamp 명령으로 샘플링한 값인지를 확인할 수 있습니다. 'Point'를 생략한 경우의 값은 0입니다.

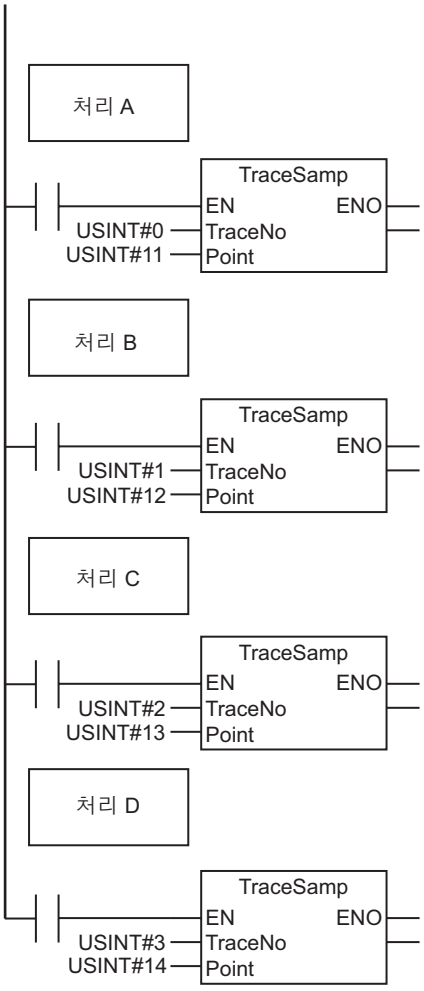
사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 아무 것도 수행하지 않고 정상 종료합니다.
  - 데이터 트레이스가 정지된 경우
  - 트레이스 설정의 샘플링 타이밍이 "샘플링 명령" 이외인 경우
  - 'TraceNo'의 값이 Sysmac Studio로 설정된 트레이스 번호가 아닌 경우
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'TraceNo'의 값이 유효 범위 밖인 경우

샘플 프로그램

A~D의 각 처리의 완료 시점에 샘플링을 실행합니다.
각 시점에서의 각 변수의 값이 기록됩니다.

LD



ST

```
처리 A
TraceSamp(USINT#0, USINT#11);
처리 B
TraceSamp(USINT#1, USINT#12);
처리 C
TraceSamp(USINT#2, USINT#13);
처리 D
TraceSamp(USINT#3, USINT#14);
```

TraceTrig

데이터 트레이스의 트리거를 발생시킵니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
TraceTrig	데이터 트레이스 트리거 조건 성립	FUN	(@)TraceTrig EN ENO TraceNo Out	TraceTrig(TraceNo);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
TraceNo	트레이스 번호	입력	트레이스 번호	(*)	–	0
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* CPU 유니트의 형식이 NJ501-□□□□인 경우는 0~3, NJ301-□□□□인 경우는 0~1

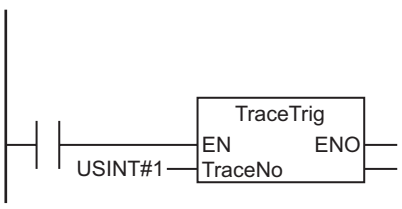
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
TraceNo						○														
Out	○																			

기능

데이터 트레이스의 트리거를 강제적으로 발생시킵니다. Sysmac Studio에 의해 지정된 트리거 조건이 성립하는지의 여부와는 관계없습니다.
이 명령을 실행할 때 트레이스 번호 'TraceNo'의 데이터 트레이스가 실행 중이라면 샘플링을 실행합니다.

기술 예가 아래에 나와 있습니다. 트레이스 번호 1의 데이터 트레이스 트리거를 발생시킵니다.

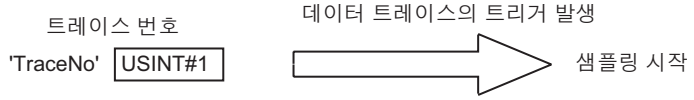
LD



ST

```
TraceTrig(USINT#1);
```

트레이스 번호 'TraceNo'인 데이터 트레이스의 트리거를 발생시킵니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
(*1)	트레이스 정보	(*2)	트레이스 정보 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

*1 CPU 유니트의 형식이 NJ501-□□□□인 경우는 변수명 _PLC_TraceSta[0..3], NJ301-□□□□인 경우는 변수명 _PLC_TraceSta[0..1]

*2 _sTRACE_STA[]

참고

- 데이터 트레이스 기능에 대한 자세한 설명은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.
- 이 명령은 프로그램 상에 여러 개 배치할 수 있습니다. 조건에 맞춰서, 트리거 조건을 성립시키는 프로그램을 기술할 수 있습니다.
- 변수와 변수의 비교 등, 통상적인 트리거 조건에서는 설정할 수 없는 타이밍에 트리거 조건을 성립시키는 프로그램을 기술할 수 있습니다.

사용상의 주의 사항

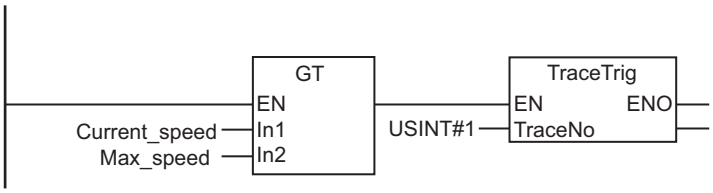
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 아무 것도 수행하지 않고 정상 종료합니다.
  - 데이터 트레이스가 정지된 경우
  - 이미 트리거 조건이 성립된 경우
  - 'TraceNo'의 값이 Sysmac Studio로 설정된 트레이스 번호가 아닌 경우
  - 'TraceNo'로 지정한 트레이스 번호의 트레이스 타입이 '연속 트레이스'인 경우
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'TraceNo'의 값이 유효 범위 밖인 경우

샘플 프로그램

속도의 현재값이 최대 한도값을 초과하면, 강제적으로 데이터 트레이스 트리거를 발생시켜서 각 변수의 값을 기록합니다.
현재 속도 Current_speed의 값이 속도 상한값 Max_speed의 값을 초과하면, TraceTrig 명령을 실행합니다.

LD

명칭	데이터형식	초기값	코멘트
Current_speed	INT	0	현재 속도
Max_speed	INT	20	속도 상한값



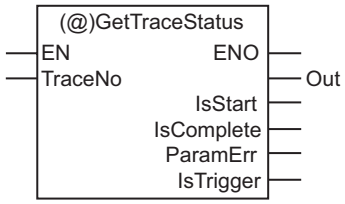
ST

명칭	데이터형식	초기값	코멘트
Current_speed	INT	0	현재 속도
Max_speed	INT	20	속도 상한값

```
IF (Current_speed > Max_speed) THEN
   TraceTrig(USINT#1);
END_IF;
```

GetTraceStatus

데이터 트레이스의 실행 Status를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetTrace Status	데이터 트레이스 Status 읽어오기	FUN		GetTraceStatus(TraceNo, IsStart, IsComplete, ParamErr, IsTrigger);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
TraceNo	트레이스 번호	입력	트레이스 번호	(*)	–	0
Out	반환값	출력	항상 TRUE	TRUE만	–	–
IsStart	실행 중 플래그		TRUE : 트레이스 실행 중 FALSE : 트레이스 실행 중이 아님	데이터 형식에 따름		
IsComplete	완료 플래그		TRUE : 트레이스 실행 완료 FALSE : 트레이스 실행 중 또는 실행되지 않음			
ParamErr	파라메터 이상 플래그		TRUE : 트레이스 설정 이상 있음 FALSE : 트레이스 설정 이상 없음			
IsTrigger	트리거 플래그		TRUE : 트레이스 트리거 조건 성립 FALSE : 트레이스 트리거 조건 비성립			

* CPU 유니트의 형식이 NJ501-□□□□인 경우는 0~3, NJ301-□□□□인 경우는 0~1

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
TraceNo						○														
Out	○																			
IsStart	○																			
IsComplete	○																			
ParamErr	○																			
IsTrigger	○																			

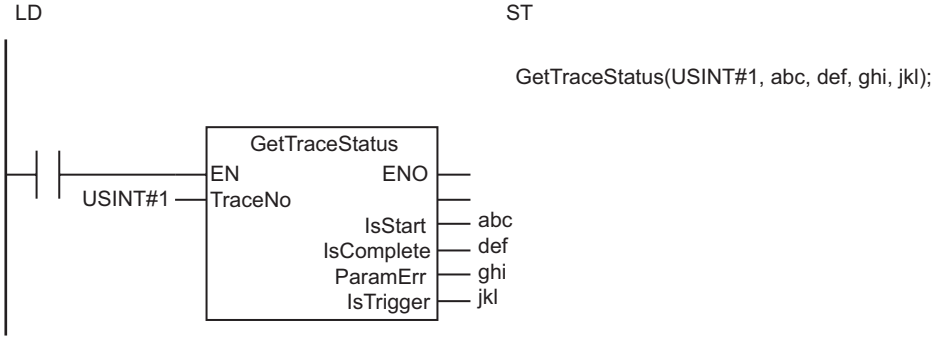
기능

트레이스 번호 'TraceNo'로 지정한 데이터 트레이스의 실행 Status를 읽어옵니다. 읽어오는 정보는 실행 중 플래그 'IsStart', 완료 플래그 'IsComplete', 파라메터 이상 플래그 'ParamErr', 트리거 플래그 'IsTrigger'입니다.

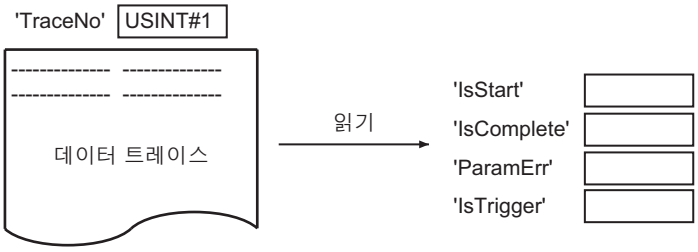
'ParamErr'의 값이 TRUE가 되는 것은, 다음과 같이 트레이스 설정에 이상이 있는 경우입니다.

- 트리거 설정, 샘플링 설정에 지정한 변수가 존재하지 않음
- 샘플링 타이밍의 설정이 '지정 태스크 주기'가 되어, 지정한 태스크가 존재하지 않음

기술 예가 아래에 나와 있습니다. 트레이스 번호 1의 실행 Status를 읽어옵니다.



트레이스 번호 'TraceNo'로 지정한 데이터 트레이스의 실행 Status를 읽어옵니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
(*1)	트레이스 정보	(*2)	트레이스 정보 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

*1. CPU 유니트의 형식이 NJ501-□□□□인 경우는 변수명 _PLC_TraceSta[0..3], NJ301-□□□□인 경우는 변수명 _PLC_TraceSta[0..1]

*2. _sTRACE_STA[]

참고

데이터 트레이스 기능에 대한 자세한 설명은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

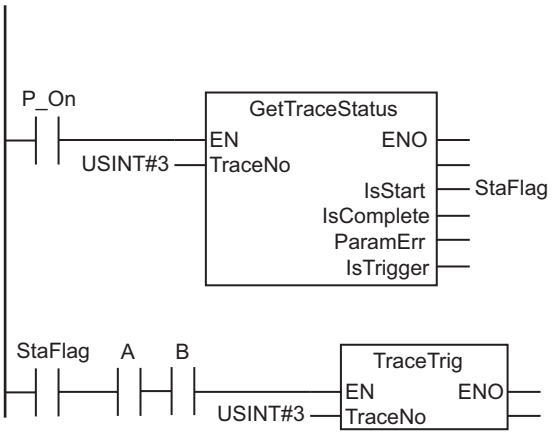
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 이 명령에서 읽어오는 정보는 시스템 정의 변수 _PLC_TraceSta[]의 내용입니다. 이 변수에는 직접 액세스할 수 없습니다. 이 변수의 내용을 읽어오려면 반드시 이 명령을 사용하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'TraceNo'의 값이 유효 범위 밖인 경우

샘플 프로그램

GetTraceStatus 명령으로 트레이스 번호 3의 실행 Status를 읽어옵니다. 데이터 트레이스 실행 중이며 변수 A와 B의 값이 모두 TRUE라면, TraceTrig 명령을 실행해서 데이터 트레이스의 트리거를 발생시킵니다.

LD

명칭	데이터 형식	초기값	코멘트
StaFlag	BOOL	FALSE	트레이스의 실행 Status
A	BOOL	FALSE	
B	BOOL	FALSE	



ST

명칭	데이터 형식	초기값	코멘트
StaFlag	BOOL	FALSE	트레이스의 실행 Status
A	BOOL	FALSE	
B	BOOL	FALSE	

```
GetTraceStatus(TraceNo:=USINT#3, IsStart=>StaFlag);

IF ( (StaFlag=TRUE) AND (A=TRUE) AND (B=TRUE) ) THEN
   TraceTrig(TraceNo:=USINT#3);
END_IF;
```

SetAlarm

사용자 이상을 발생시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetAlarm	사용자 이상 발생	FUN	(@)SetAlarm EN ENO Code Out Info1 Info2	SetAlarm(Code, Info1, Info2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Code	이벤트 코드	입력	발생시키는 사용자 이상의 이벤트 코드	1~40000	–	1
Info1	부속 정보 1		사용자 이상 발생과 함께 이벤트 로그에 기록하는 값	데이터 형식에 따름		(*)
Info2	부속 정보 2					
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Code							○													
Info1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Info2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Out	○																			

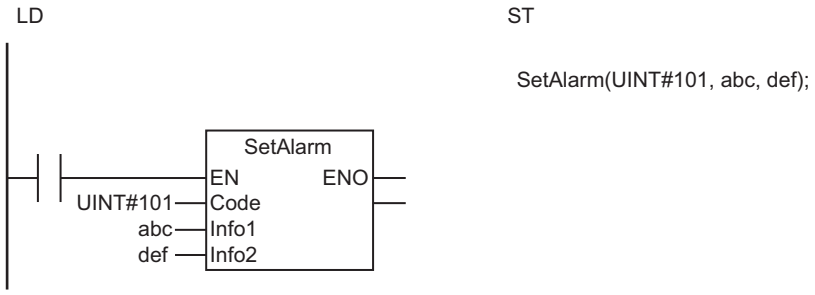
기능

Sysmac Studio의 이벤트 설정 테이블에 정의한 이벤트 코드 'Code'에 대응하는 사용자 이상을 발생시킵니다.
또한, 이벤트 코드의 중요도에 대응하는 사용자 이벤트 로그 영역에, 발생 시각, 이벤트 명칭, 이벤트 그룹, 이벤트 코드 'Code', 이벤트 중요도, 부속 정보 'Info1', 'Info2', 상세 정보의 값을 기록합니다. 발생 시각은 자동적으로 취득한 값을 기록합니다. 이벤트 명칭, 이벤트 그룹, 상세 정보는 Sysmac Studio로 설정된 정보를 기록합니다.

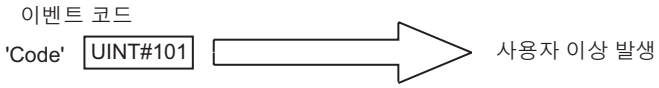
이벤트 중요도는 다음과 같이 이벤트 코드별로 정해진 분류를 기록합니다. 분류 번호가 작을수록 이벤트 중요도는 높아집니다.

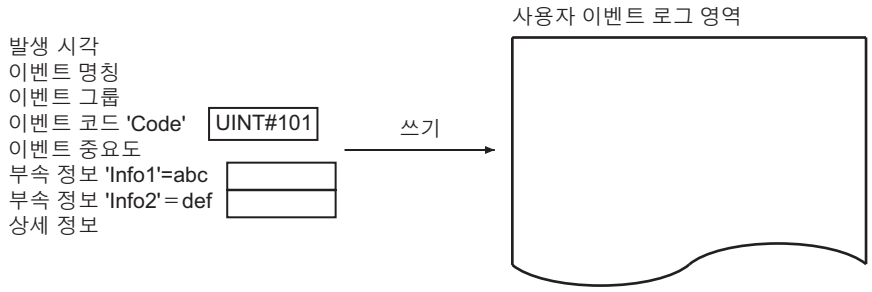
이벤트 코드	분류 User fault Level
1~5000	1
5001~10000	2
10001~15000	3
15001~20000	4
20001~25000	5
25001~30000	6
30001~35000	7
35001~40000	8

기술 예가 아래에 나와 있습니다. 이벤트 코드 101에 대응하는 사용자 이상을 발생시킵니다. 부속 정보로서, 변수 abc, def의 값을 기록합니다.



이벤트 코드 'Code'에 대응하는 사용자 이상을 발생시킵니다.
또한 사용자 이벤트 로그 영역에 발생 시각, 이벤트 명칭, 이벤트 그룹,
이벤트 코드 'Code', 이벤트 중요도, 부속 정보 'Info1', 'Info2',
상세 정보의 값을 기록합니다.





관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_AlarmFlag	사용자 이상의 이상 Status	WORD	사용자 이상의 감지 유무를 나타내는 플래그. 비트 0부터 비트 7이 사용자 이상의 각 중요도(1~8)의 발생 상황을 나타냅니다. 자세한 내용은 📖 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

참고

'Info1', 'Info2'에는 글로벌 변수를 지정하거나 로컬 변수를 지정할 수 있습니다.

사용상의 주의 사항

- 동시에 발생시킬 수 있는 사용자 이상은 8개의 이벤트 중요도별로 32건씩(최대 256건)입니다.
- 이미 같은 이벤트 코드의 사용자 이상이 발생 중인 경우, 이벤트 로그에 기록하지 않습니다.
- 'Info1', 'Info2'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수로 하면 빌드 시에 이상 상태가 됩니다.
- 'Code'의 값에 Sysmac Studio로 설정하지 않은 이벤트 코드를 설정한 경우, 이상 상태가 되지는 않습니다. 이 경우, 이벤트 그룹, 상세 정보는 사용자 이벤트 로그 영역에 기록되지 않습니다. 또한, 'Code'의 값이 이벤트 명칭으로 기록됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'Code'의 값이 유효 범위 밖인 경우
  - 최대 건수 이상의 사용자 이상을 발생시키려고 한 경우

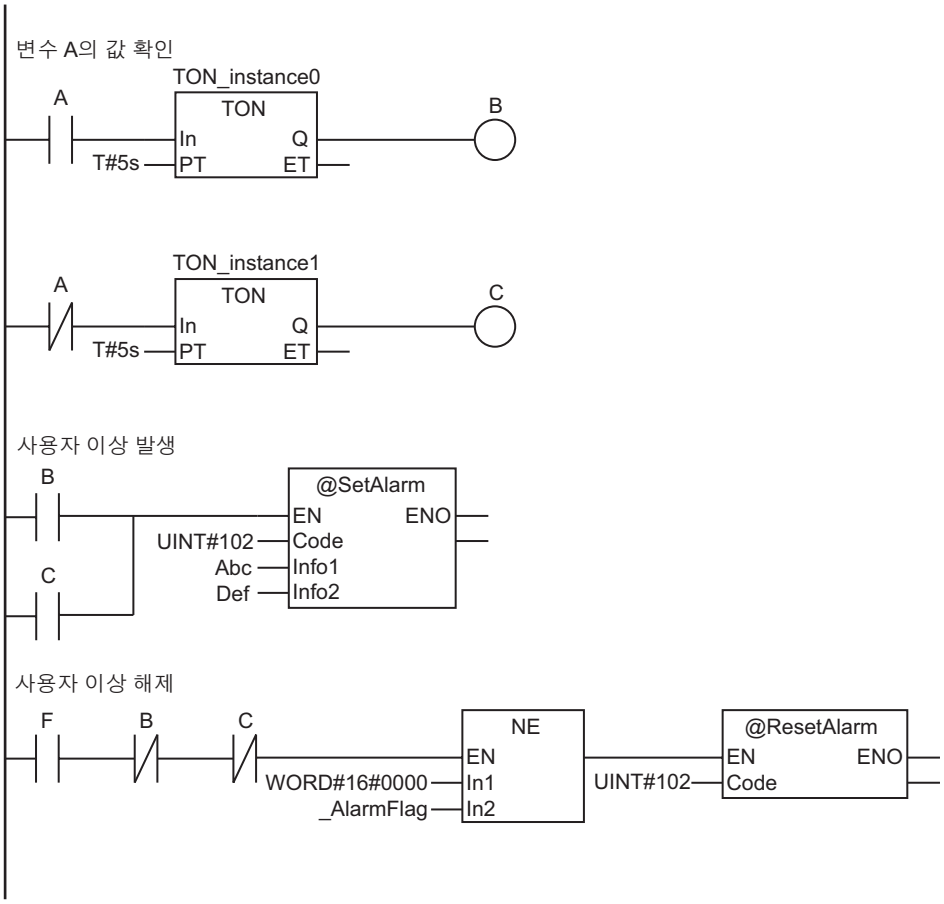
샘플 프로그램

변수 A의 값은 정상 시에는 5s마다 TRUE와 FALSE를 반복하도록 합니다.
A의 값을 감시해서, 5s 이상 값이 변화하지 않은 경우에 이벤트 코드 102의 사용자 이상을 발생시킵니다.
부속 정보는 UINT#123과 UINT#456으로 합니다.
변수 F의 값이 TRUE가 되면 사용자 이상을 해제합니다.

LD

내부 변수	명칭	데이터 형식	초기값
	A	BOOL	FALSE
	B	BOOL	FALSE
	C	BOOL	FALSE
	F	BOOL	FALSE
	Abc	UINT	123
	Def	UINT	456
	TON_instance0	TON	
	TON_instance1	TON	

외부 변수	명칭	데이터형식	상수	코멘트
	_AlarmFlag	WORD	☐	사용자 이상의 이상 Status


변수 A의 값 확인
TON_instance0
TON
A
B
In
Q
T#5s
PT
ET
TON_instance1
TON
A
C
In
Q
T#5s
PT
ET
사용자 이상 발생
B
C
@SetAlarm
EN
ENO
UINT#102
Code
Abc
Info1
Def
Info2
사용자 이상 해제
F
B
C
NE
EN
WORD#16#0000
In1
_AlarmFlag
In2
UINT#102
@ResetAlarm
EN
ENO
Code

ST

내부 변수	명칭	데이터 형식	초기값
	A	BOOL	FALSE
	B	BOOL	FALSE
	C	BOOL	FALSE
	F	BOOL	FALSE
	Abc	UINT	123
	Def	UINT	456
	TON_instance0	TON	
	TON_instance1	TON	

외부 변수	명칭	데이터형식	상수	코멘트
	_AlarmFlag	WORD	☐	사용자 이상의 이상 Status

```
// 변수 A의 값 확인
IF (A=TRUE) THEN
   TON_instance0(In:=TRUE, PT:=T#5s, Q=>B);
ELSE
   TON_instance0(In:=FALSE, Q=>B);
END_IF;

IF (A=FALSE) THEN
   TON_instance1(In:=TRUE, PT:=T#5s, Q=>C);
ELSE
   TON_instance1(In:=FALSE, Q=>C);
END_IF;

// 사용자 이상 발생
IF (B=TRUE) OR (C=TRUE) THEN
   SetAlarm(
      Code:=UINT#102,
      Info1 :=Abc,
      info2 :=Def);
END_IF;

// 사용자 이상 해제
IF (F=TRUE) & (B=FALSE) & (C=FALSE) & (_AlarmFlag<>WORD#16#0000) THEN
   ResetAlarm(Code:=UINT#102);
END_IF;
```

ResetAlarm

사용자 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetAlarm	사용자 이상 해제	FUN	(@)ResetAlarm EN ENO Code Out	ResetAlarm(Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Code	이벤트 코드	입력	해제하는 사용자 이상의 이벤트 코드 0 : 모든 사용자 이상 해제	0~40000	–	0
Out	반환값	출력	항상 TRUE	TRUE만	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Code							○													
Out	○																			

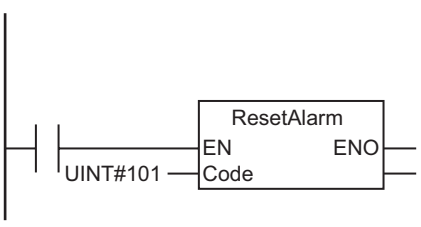
기능

이벤트 코드 'Code'로 지정한 사용자 이상을 해제합니다.
그리고 사용자 이벤트 로그 영역에 사용자 이상의 개별 해제를 의미하는 이벤트를 기록합니다. 이 이벤트의 이벤트 코드는 No.65533, 중요도는 "User Information"입니다.

'Code'의 값이 0인 경우, 발생한 모든 사용자 이상을 해제합니다.
이 경우, 사용자 이벤트 로그 영역에 사용자 이상의 전체 해제를 의미하는 이벤트를 기록합니다. 이 이벤트의 이벤트 코드는 No.65534, 중요도는 "User Information"입니다.

기술 예가 아래에 나와 있습니다. 이벤트 코드 101에 대응하는 사용자 이상을 해제합니다.

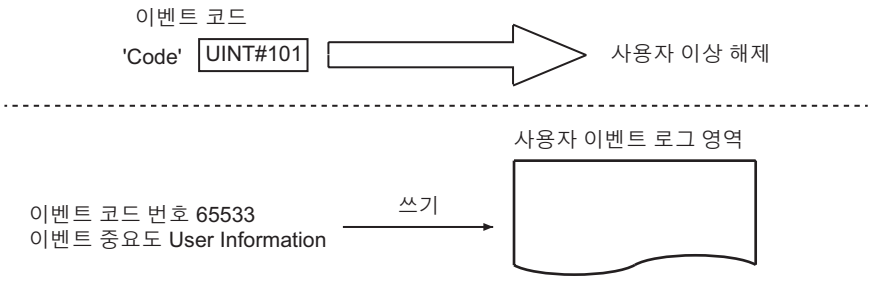
LD



ST

```
ResetAlarm(UINT#101);
```

이벤트 코드 'Code'로 지정한 사용자 이상을 해제합니다.
또한 사용자 이벤트 로그 영역에 사용자 이상의 개별 해제를 의미하는 이벤트를 기록합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_AlarmFlag	사용자 이상의 이상 Status	WORD	사용자 이상의 감지 유무를 나타내는 플래그 비트 0부터 비트 7이 사용자 이상의 각 중요도(1~8)의 발생 상황을 나타냅니다. 자세한 내용은 📖 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

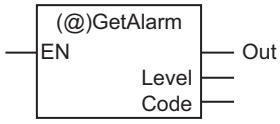
- 'Code'로 지정한 사용자 이상이 발생하지 않은 경우, 이상 상태가 되지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'Code'의 값이 유효 범위 밖인 경우

샘플 프로그램

📖 'SetAlarm 명령(2-747페이지)'의 샘플 프로그램을 참조하십시오.

GetAlarm

발생 중인 사용자 이상에서 최고 중요 이벤트의 중요도(사용자 이상 수준 1~8)와 최고 중요 이벤트 코드를 취득합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
GetAlarm	사용자 이상 Status 취득	FUN		Out:=GetAlarm(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 사용자 이상 있음 FALSE : 발생 중인 사용자 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 이벤트 중요도		발생 중인 사용자 이상 가운데 최고 중요 이벤트 중요도 0 : 사용자 이상 없음 1~8 : 이벤트 중요도	0~8		
Code	최고 중요 이벤트 코드		발생 중인 사용자 이상 가운데 최고 중요 이벤트 코드 0 : 사용자 이상 없음 1~40000 : 이벤트 중요도	0~40000		

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code							○													

기능

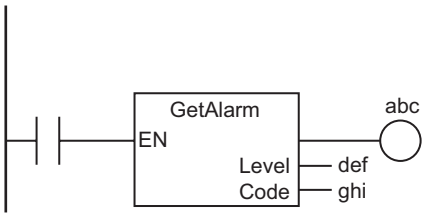
발생 중인 사용자 이상의 최고 중요 이벤트 중요도 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.

발생 중인 사용자 이상이 없는 경우, 이상 유무 'Out'의 값이 FALSE가 됩니다.

최고 중요 이벤트 중요도의 사용자 이상이 복수 발생한 경우, 'Code'의 값은 가장 먼저 발생한 사용자 이상의 이벤트 코드가 됩니다.

기술 예가 아래에 나와 있습니다.

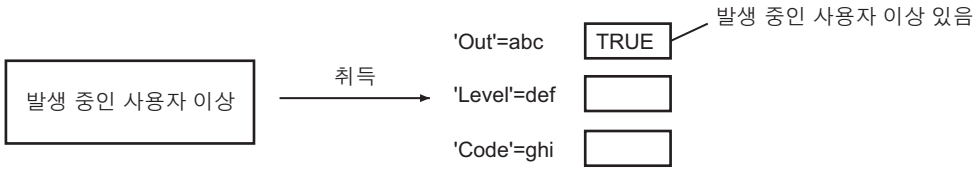
LD



ST

```
abc:=GetAlarm(def, ghi);
```

발생 중인 사용자 이상의 최고 중요 이벤트 중요도 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_AlarmFlag	사용자 이상의 이상 Status	WORD	사용자 이상의 감지 유무를 나타내는 플래그 비트 0부터 비트 7이 사용자 이상의 각 중요도(1~8)의 발생 상황을 나타냅니다. 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

ResetPLCError

PLC 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetPLC Error	PLC 이상 해제	FB	ResetPLCError_instance ResetPLCError Execute Done Busy Error ErrorID	ResetPLCError(Execute, Done, Busy, Error, ErrorID);

변수

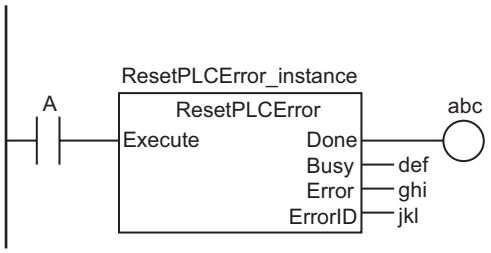
공통의 변수만

기능

PLC 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.

기술 예가 아래에 나와 있습니다.

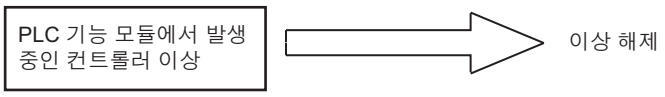
LD



ST

```
ResetPLCError_instance(A, abc, def, ghi, jkl);
```

PLC 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_PLC_ErrSta	PLC 기능 모듈 이상 Status	WORD	PLC 기능 모듈에서 감지한 이상 상태 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

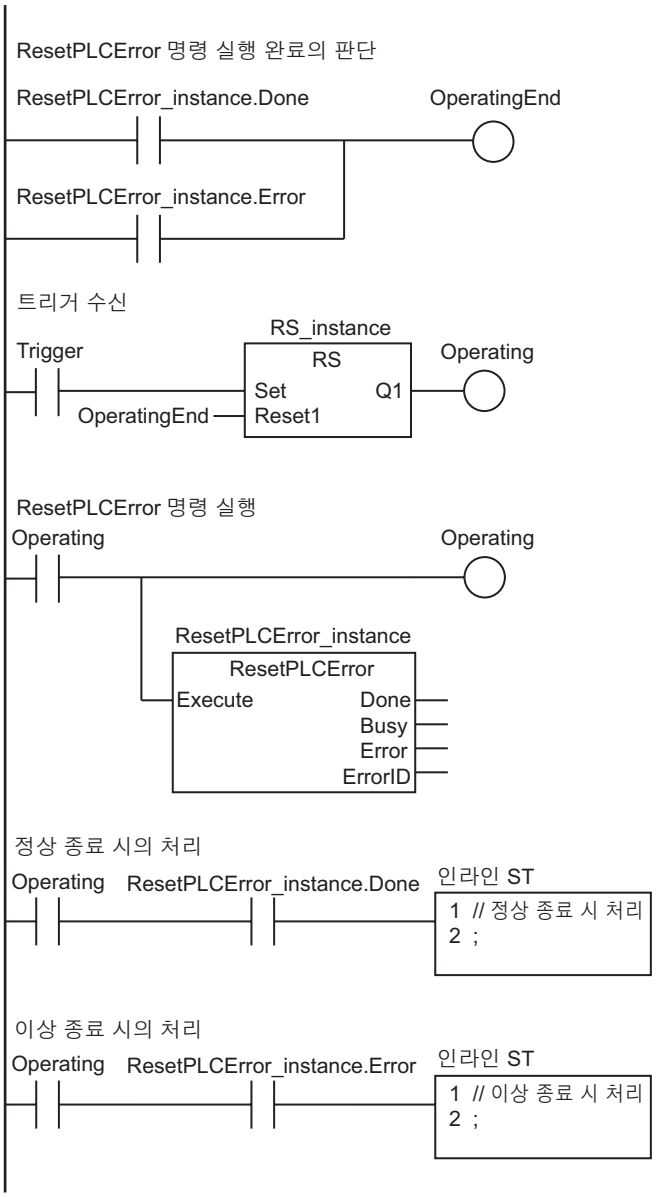
이 명령의 실행 직후에, 이상이 해제된다는 보장은 없습니다. 이상이 해제되었는지의 여부는 GetPLCError 명령으로 확인하십시오.

샘플 프로그램

Trigger의 값이 FALSE→TRUE가 되면 ResetPLCError 명령을 실행합니다. ResetPLCError 명령이 정상 종료(Done의 값이 TRUE)되면 정상 종료 시 처리를 실행합니다. 이상 종료(Error의 값이 TRUE)되면 이상 종료 시 처리를 실행합니다.

LD

명칭	데이터 형식	초기값	코멘트
OperatingEnd	BOOL	FALSE	처리 종료
Trigger	BOOL	FALSE	실행 조건
Operating	BOOL	FALSE	처리 중
RS_instance	RS		
ResetPLCError_instance	ResetPLCError		



ST

명칭	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
OperatingStart	BOOL	FALSE	처리 시작
Operating	BOOL	FALSE	처리 중
ResetPLCError_instance	ResetPLCError		

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
   OperatingStart:=TRUE;
   Operating:=TRUE;
END_IF;
LastTrigger:=Trigger;

// ResetPLCError_instance 초기화
IF (OperatingStart=TRUE) THEN
   ResetPLCError_instance(Execute:=FALSE);
   OperatingStart:=FALSE;
END_IF;

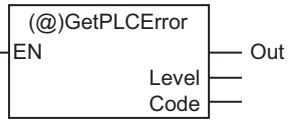
// ResetPLCError 명령 실행
IF (Operating=TRUE) THEN
   ResetPLCError_instance(Execute:=TRUE);

   IF (ResetPLCError_instance.Done=TRUE) THEN
      // 정상 종료 시 처리
      Operating:=FALSE;
   END_IF;

   IF (ResetPLCError_instance.Error=TRUE) THEN
      // 이상 종료 시 처리
      Operating:=FALSE;
   END_IF;
END_IF;
```

GetPLCError

PLC 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetPLCError	PLC 이상 Status 취득	FUN		Out:=GetPLCError(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 컨트롤러 이상 있음 FALSE : 발생 중인 컨트롤러 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 Status		PLC 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 이벤트 코드 0 : 컨트롤러 이상 없음 2 : 부분 정지 이상 수준 3 : 가벼운 이상 수준	0,2,3		
Code	최고 중요 이벤트 코드		PLC 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 이벤트 코드 16#0000_0000 : 컨트롤러 이상 없음 16#0007_0000~ 16#FFFF_FFFF : 이벤트 코드	16#00000000 16#00070000~ 16#FFFFFFFF		

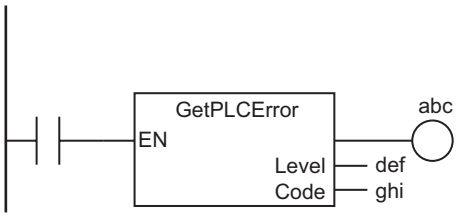
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code				○																

기능

PLC 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.
발생 중인 컨트롤러 이상이 없는 경우, 이상 유무 'Out'의 값이 FALSE가 됩니다.
최고 중요 이벤트 중요도의 컨트롤러 이상이 복수 발생한 경우, 'Code'의 값은 가장 먼저 발생한 컨트롤러 이상의 이벤트 코드가 됩니다.

기술 예가 아래에 나와 있습니다.

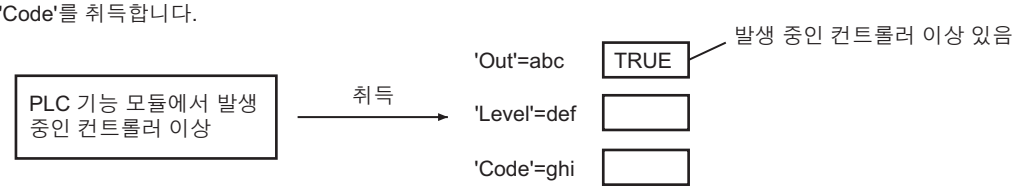
LD



ST

```
abc:=GetPLCError(def, ghi);
```

PLC 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.

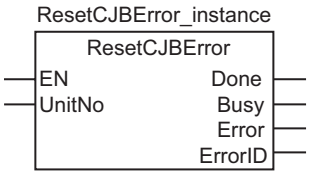


관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_PLC_ErrSta	PLC 기능 모듈 이상 Status	WORD	PLC 기능 모듈에서 감지한 이상 상태 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

ResetCJBError

I/O 버스에서 발생 중인 컨트롤러 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetCJB Error	I/O 버스 이상 해제	FB		ResetCJBError_instance(Execute, UnitNo, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitNo	유니트 번호	입력	이상을 해제하는 유니트 번호	_CBU_No00~ _CBU_No15, _SIO_No00~ _SIO_No95 _UNIT_ALL	–	_UNIT _ALL

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitNo	열거 형식 _eUnitNo 열거자는 기능 설명 참조																			

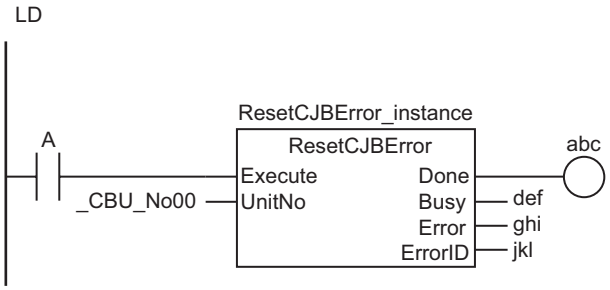
기능

I/O 버스에서 발생 중인 컨트롤러 이상을 해제합니다.
유니트 번호 'UnitNo'로 지정한 유니트가 CJ 시리즈의 고기능 유니트인 경우, 그 유니트의 재시작도 수행합니다.

'UnitNo'의 데이터 형식은 열거 형식 _eUnitNo입니다. 열거자의 의미는 다음과 같습니다.

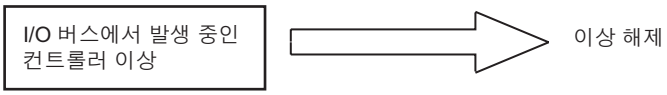
열거자	의미
_CBU_No00~_CBU_No15	CPU 고기능 유니트의 유니트 번호 00~15
_SIO_No00~_SIO_No95	고기능 I/O 유니트의 호기 번호 00~95
_UNIT_ALL	모든 유니트

'UnitNo'=_CBU_No00인 경우의 예가 아래에 나와 있습니다. I/O 버스에서 발생 중인 컨트롤러 이상을 해제함과 동시에, 유니트 번호 0인 CPU 고기능 유니트의 재시작도 수행합니다.

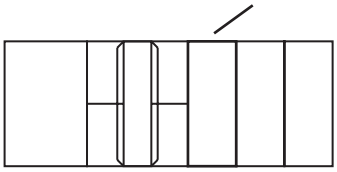



```
ST
ResetCJBError_instance(A, _CBU_No00,
                       abc, def, ghi, jkl);
```

I/O 버스에서 발생 중인 컨트롤러 이상을 해제합니다.
또한 유니트 번호 'UnitNo'인 CPU 고기능 유니트를 재시작합니다.



'UnitNo'＝_CBU_No00 유니트 번호 0인 CPU 고기능 유니트를 재시작



관련된 시스템 정의 변수

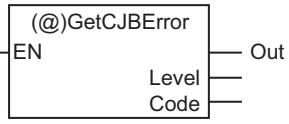
변수명	명칭	데이터 형식	내용
_CJB_ErrSta	I/O 버스 이상 Status	WORD	I/O 버스에서 감지한 이상 상태 자세한 내용은 📖 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

- 이 명령의 실행 직후에, 이상이 해제된다는 보장은 없습니다. 이상이 해제되었는지의 여부는 GetCJBError 명령으로 확인하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'UnitNo'의 값이 유효 범위 밖인 경우
  - 'UnitNo'로 지정한 유니트가 존재하지 않는 경우

GetCJBError

I/O 버스에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetCJBError	I/O 버스 이상 Status 취득	FUN		Out:=GetCJBError(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 컨트롤러 이상 있음 FALSE : 발생 중인 컨트롤러 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 Status		I/O 버스에서 발생 중인 컨트롤러 이상 가운데 최고 중요 Status 0 : 컨트롤러 이상 없음 2 : 부분 정지 이상 수준 3 : 가벼운 이상 수준	0,2,3		
Code	최고 중요 이벤트 코드		I/O 버스에서 발생 중인 컨트롤러 이상 가운데 최고 중요 이벤트 코드 16#0000_0000 : 컨트롤러 이상 없음 16#0007_0000~ 16#FFFF_FFFF : 이벤트 코드	16#00000000 16#00070000~ 16#FFFFFFFF		

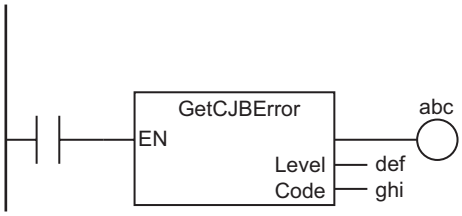
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code				○																

기능

I/O 버스에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.
발생 중인 컨트롤러 이상이 없는 경우, 이상 유무 'Out'의 값이 FALSE가 됩니다.
최고 중요 이벤트 중요도의 컨트롤러 이상이 복수 발생한 경우, 'Code'의 값은 가장 먼저 발생한 컨트롤러 이상의 이벤트 코드가 됩니다.

기술 예가 아래에 나와 있습니다.

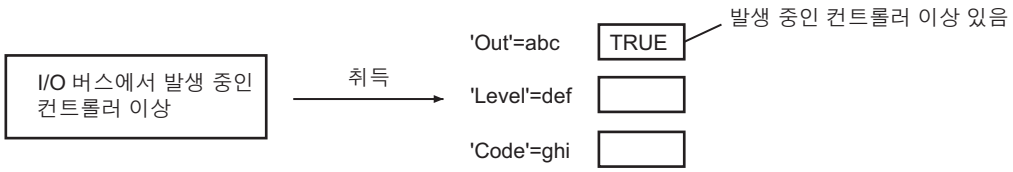
LD



ST

```
abc:=GetCJBError(def, ghi);
```

I/O 버스에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.

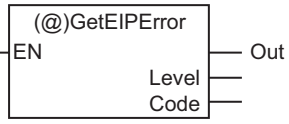


관련된 시스템 정의 변수

변수명	명칭	데이터형식	내용
_CJB_ErrSta	I/O 버스 이상 Status	WORD	I/O 버스에서 감지한 이상 상태 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

GetEIPError

EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetEIPError	EtherNet/IP 이상 Status 취득	FUN		Out:=GetEIPError(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 컨트롤러 이상 있음 FALSE : 발생 중인 컨트롤러 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 Status		EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 Status 0 : 컨트롤러 이상 없음 2 : 부분 정지 이상 수준 3 : 가벼운 이상 수준	0,2,3		
Code	최고 중요 이벤트 코드		EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 이벤트 코드 16#0000_0000 : 컨트롤러 이상 없음 16#0007_0000~ 16#FFFF_FFFF : 이벤트 코드	16#00000000 16#00070000~ 16#FFFFFFFF		

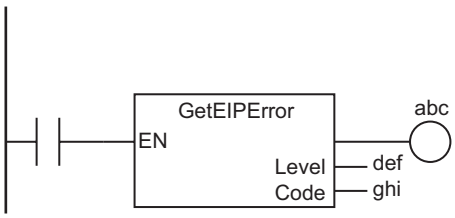
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code				○																

기능

EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.
발생 중인 컨트롤러 이상이 없는 경우, 이상 유무 'Out'의 값이 FALSE가 됩니다.
최고 중요 이벤트 중요도의 컨트롤러 이상이 복수 발생한 경우, 'Code'의 값은 가장 먼저 발생한 컨트롤러 이상의 이벤트 코드가 됩니다.

기술 예가 아래에 나와 있습니다.

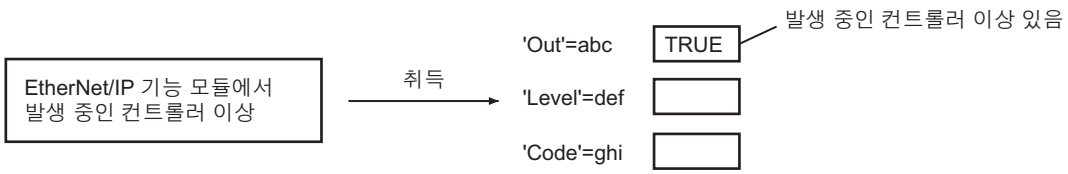
LD



ST

```
abc:=GetEIPError(def, ghi);
```

EtherNet/IP 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_ErrSta	EtherNet/IP 기능 모듈 이상 Status	WORD	EtherNet/IP 기능 모듈에서 감지한 이상 상태 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

ResetMCError

모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetMCError	모션 제어 이상 해제	FB	ResetMCError_instance ResetMCError Execute / Done Busy Failure Error ErrorID	ResetMCError_instance(Execute, Done, Busy, Failure Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Failure	부정 종료	출력	TRUE : 이상 해제 실패 FALSE : 이상 해제 성공	데이터 형식에 따름	–	–

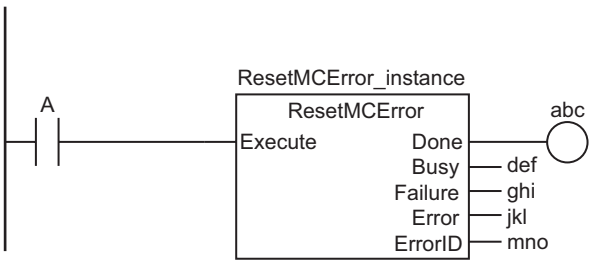
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Failure	○																			

기능

모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.
해제에 실패한 경우, 'Failure'의 값이 TRUE가 됩니다.

기술 예가 아래에 나와 있습니다.

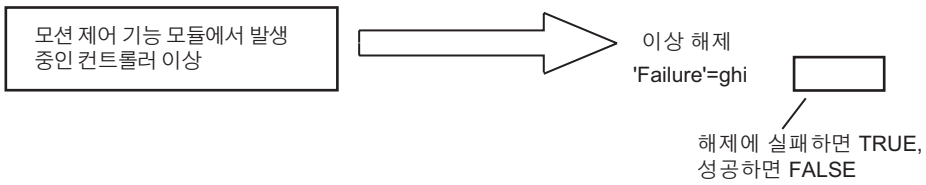
LD



ST

```
ResetMCError_instance(A, abc, def,
                      ghi, jkl, mno);
```

모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.
해제에 실패한 경우, 'Failure'의 값이 TRUE가 됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_MC_ErrSta	모션 제어 이상 Status	WORD	모션 제어 기능 모듈에서 감지한 이상 상태. 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

사용상의 주의 사항

- 이 명령의 실행 직후에, 이상이 해제된다는 보장은 없습니다. 이상이 해제되었는지의 여부는 GetMCError 명령으로 확인하십시오.
- MC 시운전 중에 이 명령을 실행하려고 하면, 'Busy'의 값이 TRUE 상태로 유지되어, 실행할 수 없습니다.
- 오므론 서보 드라이버 G5 시리즈에 대해 이 명령을 사용하는 경우, ResetECError 명령이 동시에 실행되지 않도록 명령의 배타적 제어를 실시하십시오. 이 명령과 ResetECError 명령을 동시에 실행하면 G5 시리즈가 그 이후의 SDO를 수신하지 못하게 되는 경우가 있습니다.

버전 관련 정보

- CPU 유니트 Ver.1.02 이상에서는 이 명령의 인스턴스를 100개까지만 생성할 수 있습니다.
- CPU 유니트 Ver.1.02 이상에서 이 명령의 인스턴스가 101개 이상 생성된 사용자 프로그램을 컨트롤러에 전송하면 컨트롤러 이상이 발생합니다. 컨트롤러 이상은 사용자 프로그램의 전송 방법에 따라 다음과 같이 달라집니다.

사용자 프로그램의 전송 방법	컨트롤러 이상의 이벤트 코드	컨트롤러 이상의 중요도
'동기' 기능을 사용하여 프로젝트 전송	10250000Hex	전체 정지 이상 레벨
	571D0000Hex	감시 정보
온라인 편집 내용 전송	571D0000Hex	감시 정보

- CPU 유니트 Ver.1.01 이하에서는 이 명령의 인스턴스가 101개 이상 생성된 사용자 프로그램을 컨트롤러에 전송해도 상기의 컨트롤러 이상은 발생하지 않습니다. 단, 이 명령의 인스턴스를 너무 많이 생성하면 사용자 프로그램의 규모가 커져 전체 정지 이상 레벨의 컨트롤러 이상이 발생할 수 있습니다.

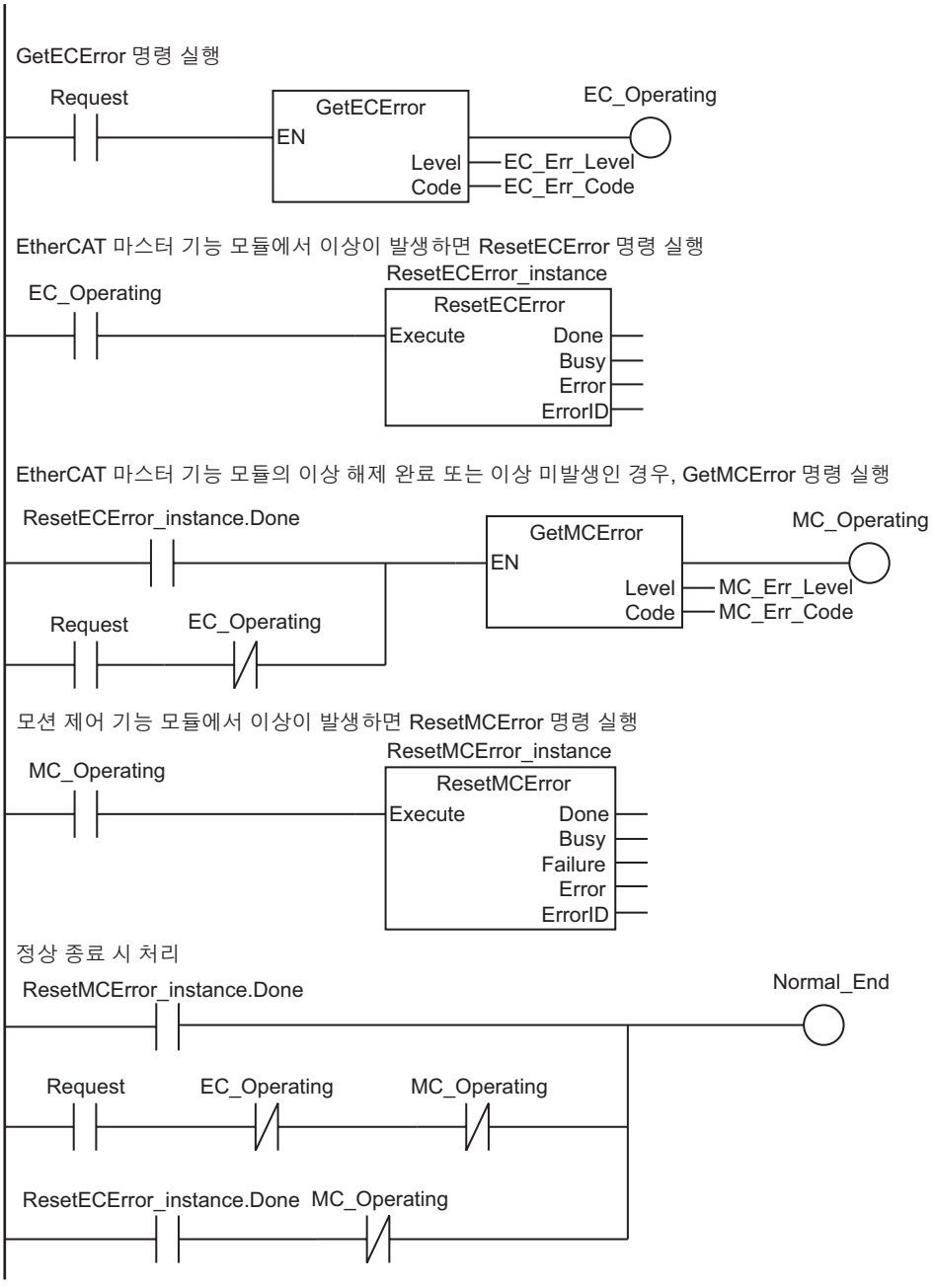
샘플 프로그램

EtherCAT 마스터 기능 모듈 및 모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 감지합니다. 만일 이상이 발생했다면, 그 이상을 해제합니다.
처리 순서는 다음과 같습니다.

1. **GetECError 명령으로 EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상을 감지합니다.**
2. **만일 이상이 발생했다면, ResetECError 명령으로 이상을 해제합니다.**
3. **GetMCError 명령으로 모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상을 감지합니다.**
4. **만일 이상이 발생했다면, ResetMCError 명령으로 이상을 해제합니다.**

LD

명칭	데이터 형식	초기값	코멘트
Request	BOOL	FALSE	이상 감지 해제 요구
EC_Err_Level	UINT	0	EtherCAT 마스터 기능 모듈 최고 중요 이벤트 중요도
EC_Err_Code	DWORD	DWORD#16#0	EtherCAT 마스터 기능 모듈 최고 중요 이벤트 코드
EC_Operating	BOOL	FALSE	EtherCAT 마스터 기능 모듈 이상 해제 중
MC_Err_Level	UINT	0	모션 제어 기능 모듈 최고 중요 이벤트 중요도
MC_Err_Code	DWORD	DWORD#16#0	모션 제어 기능 모듈 최고 중요 이벤트 코드
MC_Operating	BOOL	FALSE	모션 제어 기능 모듈 이상 해제 중
Normal_End	BOOL	FALSE	정상 종료
ResetECError_instance	ResetECError		
ResetMCError_instance	ResetMCError		



ST

명칭	데이터 형식	초기값	코멘트
Request	BOOL	FALSE	이상 감지 해제 요구
EC_Error	BOOL	FALSE	EtherCAT 마스터 기능 모듈 이상 있음
EC_Err_Level	UINT	0	EtherCAT 마스터 기능 모듈 최고 중요 이벤트 중요도
EC_Err_Code	DWORD	DWORD#16#0	EtherCAT 마스터 기능 모듈 최고 중요 이벤트 코드
EC_Stage	INT	0	EtherCAT 마스터 기능 모듈 이상 해제 상태
MC_Error	BOOL	FALSE	모션 제어 기능 모듈 이상 있음
MC_Err_Level	UINT	0	모션 제어 기능 모듈 최고 중요 이벤트 중요도
MC_Err_Code	DWORD	DWORD#16#0	모션 제어 기능 모듈 최고 중요 이벤트 코드
MC_Stage	INT	0	모션 제어 기능 모듈 이상 해제 상태
ResetECError_instance	ResetECError		
ResetMCError_instance	ResetMCError		

```
IF (Request=TRUE) THEN                                  // 이상 해제 요구를 판정
    EC_Error:=GetECError (EC_Err_Level, EC_Err_Code);   // EtherCAT 마스터 기능 모듈의 컨트롤러 이상 감지
    MC_Error:=GetMCError (MC_Err_Level, MC_Err_Code);   // 모션 제어 기능 모듈의 컨트롤러 이상 감지

    IF (EC_Error=TRUE) THEN    // EtherCAT 마스터 기능 모듈의 컨트롤러 이상 있음
        CASE EC_Stage OF
         0 :                   // 초기화
            ResetECError_instance(Execute:=FALSE);
            EC_Stage:=INT#1;
        1 :                    // EtherCAT 마스터 기능 모듈의 컨트롤러 이상 해제 중
            ResetECError_instance(Execute:=TRUE);
            IF (ResetECError_instance.Done=TRUE) THEN
                EC_Stage:=INT#99;       // 정상 종료
            END_IF;
            IF (ResetECError_instance.Error=TRUE) THEN
                EC_Stage:=INT#98;       // 이상 종료
            END_IF;
        99 :                   // 정상 종료 시 처리
            EC_Stage:=INT#0;
        98 :                   // 이상 종료 시 처리
            EC_Stage:=INT#0;
        END_CASE;
    END_IF;

    IF (MC_Error=TRUE) THEN   // 모션 제어 기능 모듈의 컨트롤러 이상 있음
        CASE MC_Stage OF
         0 :                   // 초기화
            ResetMCError_instance(Execute:=FALSE);
            MC_Stage:=INT#1;
        1 :                    // 모션 제어 기능 모듈의 컨트롤러 이상 해제 중
            IF (EC_Error=FALSE) THEN                    // 모든 슬레이브 복귀
                ResetMCError_instance(Execute:=TRUE);
                IF (ResetMCError_instance.Done=TRUE) THEN
                    MC_Stage:=INT#99;  // 정상 종료
                END_IF;
                IF ( (ResetMCError_instance.Error=TRUE) OR (ResetMCError_instance.Failure=TRUE) ) THEN
                    MC_Stage:=INT#98;  // 이상 종료
                END_IF;
            END_IF;
        99 :                   // 정상 종료 시 처리
            MC_Stage:=INT#0;
        98 :                   // 이상 종료 시 처리
            MC_Stage:=INT#0;
        END_CASE;
    END_IF;
END_IF;
```

GetMCError

모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status(부분 정지 이상, 가벼운 이상)와 최고 중요 이벤트 코드를 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetMCError	모션 제어 이상 Status 취득	FUN	(@)GetMCError EN Out Level Code	Out: =GetMCError(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 컨트롤러 이상 있음 FALSE : 발생 중인 컨트롤러 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 Status		모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 Status 0 : 컨트롤러 이상 없음 2 : 부분 정지 이상 수준 3 : 가벼운 이상 수준	0,2,3		
Code	최고 중요 이벤트 코드		모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상 가운데 최고 중요 이벤트 코드 16#0000_0000 : 컨트롤러 이상 없음 16#0007_0000~ 16#FFFF_FFFF : 이벤트 코드	16#00000000 16#00070000~ 16#FFFFFFFF		

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code				○																

기능

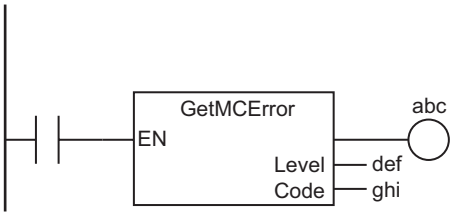
모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.

발생 중인 컨트롤러 이상이 없는 경우, 이상 유무 'Out'의 값이 FALSE가 됩니다.

최고 중요 이벤트 중요도의 컨트롤러 이상이 복수 발생한 경우, 'Code'의 값은 가장 먼저 발생한 컨트롤러 이상의 이벤트 코드가 됩니다.

기술 예가 아래에 나와 있습니다.

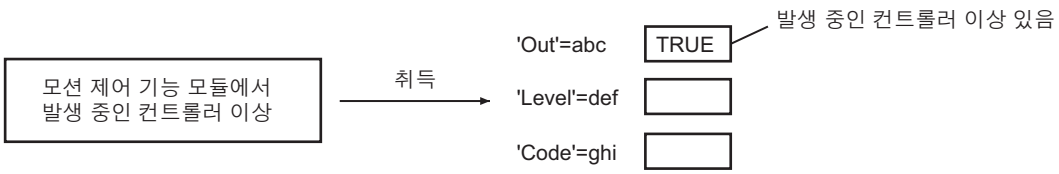
LD



ST

```
abc:=GetMCError(def, ghi);
```

모션 제어 기능 모듈에서 발생 중인 컨트롤러 이상의 최고 중요 Status 'Level'과 최고 중요 이벤트 코드인 'Code'를 취득합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_MC_ErrSta	모션 제어 이상 Status	WORD	모션 제어 기능 모듈에서 감지한 이상 상태 자세한 내용은 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

샘플 프로그램

🕮 'ResetMCError 명령(2-768페이지)'의 샘플 프로그램을 참조하십시오.

ResetECError

EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetECError	EtherCAT 이상 해제	FB	ResetECError_instance ResetECError Execute Done Busy Error ErrorID	ResetECError_instance(Execute, Done, Busy, Error, ErrorID);

변수

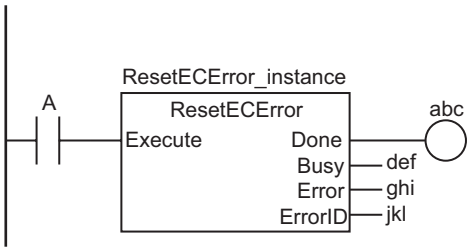
공통의 변수만

기능

EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.

기술 예가 아래에 나와 있습니다.

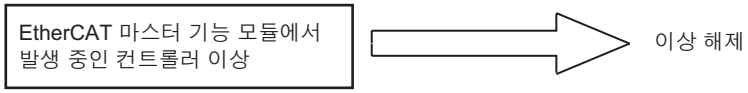
LD



ST

```
ResetECError_instance(A, abc, def, ghi, jkl);
```

EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상을 해제합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_ErrSta	내장 EtherCAT 이상	WORD	EtherCAT 마스터 기능 모듈 이상의 집약 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"를 참조하십시오.

사용상의 주의 사항

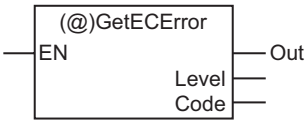
- 이 명령의 실행 직후에, 이상이 해제된다는 보장은 없습니다. 이상이 해제되었는지의 여부는 GetECError 명령으로 확인하십시오.
- 오므론 서보 드라이버 G5 시리즈에 대해 이 명령을 사용하는 경우, ResetMCError 명령, MC_Reset 명령, MC_GroupReset 명령 중 하나가 동시에 실행되지 않도록 명령의 배타적 제어를 실시하십시오. 이 명령과 이들 3가지 명령 중 하나를 동시에 실행하면 G5 시리즈가 그 이후의 SDO를 수신하지 못하게 되는 경우가 있습니다.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령을 실행 중에는 이 명령을 실행할 수 없습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - EtherCAT 마스터 기능 모듈에서 발생 중인 컨트롤러 이상 해제 처리 중에, 다시 이 명령이 실행된 경우
  - EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 중 하나가 실행 중인 경우

샘플 프로그램

🕮 'ResetMCError 명령(2-768페이지)'의 샘플 프로그램을 참조하십시오.

GetECError

EtherCAT 마스터 기능 모듈에서 발생 중인 이상을 검출합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
GetECError	EtherCAT 이상 Status 취득	FUN		Out:=GetECError(Level, Code);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	이상 유무	출력	TRUE : 발생 중인 이상 있음*1 FALSE : 발생 중인 이상 없음	데이터 형식에 따름	–	–
Level	최고 중요 Status		발생 중인 이상 가운데 최고 중요 Status*1 0 : 이상 없음 2 : 부분 정지 이상 수준 3 : 가벼운 이상 수준	0,2,3		
Code	최고 중요 이벤트 코드		발생 중인 이상 가운데 최고 중요 이벤트 코드*1	16#00000000 16#00070000~ 16#FFFFFFFF		

*1 검출할 수 있는 이상의 종류는 CPU 유니트 및 Sysmac Studio의 버전에 따라 다릅니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
Level							○													
Code				○																

기능

EtherCAT 마스터 기능 모듈에서 발생 중인 이상을 검출합니다.
'Out'의 값은 이상이 있는 경우에 TRUE, 이상이 없는 경우에 FALSE가 됩니다.
'Level'은 발생 중인 이상 가운데 가장 중요한 Status를 나타냅니다.
'Code'는 발생 중인 이상 가운데 가장 중요한 이벤트 코드를 나타냅니다.

검출할 이상의 종류와 출력 변수의 값

이 명령으로 검출할 수 있는 이상의 종류는 CPU 유니트의 유니트 버전에 따라 다릅니다. 각 버전에서 검출하는 이상의 종류는 다음과 같습니다.

CPU 유니트의 유니트 버전	검출할 이상의 종류
Ver.1.02 이상	통신 포트 이상, 마스터 이상, 슬레이브 이상
Ver.1.01 이하	통신 포트 이상, 마스터 이상

CPU 유니트의 유니트 버전, EtherCAT 마스터 기능 모듈의 상태 및 각 출력 변수 값과의 관계는 다음과 같습니다.

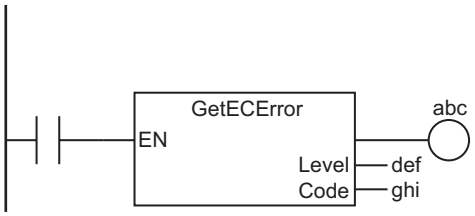
CPU 유니트의 유니트 버전	EtherCAT 마스터 기능 모듈의 상태	'Out'의 값	'Level'의 값	'Code'의 값
Ver.1.04 이상	이상 없음	FALSE	0	16#0000_0000
	통신 포트 이상 또는 마스터 이상 있음	TRUE	발생 중인 이상 중 가장 중요한 Status 2: 부분 정지 이상 레벨 3: 가벼운 이상 레벨	발생 중인 이상 중 가장 중요한 이벤트 코드*1 16#0007_0000~ 16#FFFF_FFFF
	슬레이브 이상 있음			16#0000_0000
Ver.1.02, Ver.1.03	이상 없음	FALSE	0	16#0000_0000
	통신 포트 이상 또는 마스터 이상 있음	TRUE	발생 중인 이상 중 가장 중요한 Status 2: 부분 정지 이상 레벨 3: 가벼운 이상 레벨	발생 중인 이상 중 가장 중요한 이벤트 코드*1 16#0007_0000~ 16#FFFF_FFFF
	슬레이브 이상 있음	FALSE		16#0000_0000
Ver.1.00, Ver.1.01	이상 없음	FALSE	0	16#0000_0000
	통신 포트 이상 또는 마스터 이상 있음	TRUE	발생 중인 이상 중 가장 중요한 Status 2: 부분 정지 이상 레벨 3: 가벼운 이상 레벨	발생 중인 이상 중 가장 중요한 이벤트 코드*1 16#0007_0000~ 16#FFFF_FFFF
	슬레이브 이상 있음	FALSE	0	16#0000_0000

*1 가장 중요한 이벤트 중요도의 이상이 여러 개 발생한 경우, 가장 먼저 발생한 이상의 이벤트 코드가 됩니다.

기술 예

기술 예가 아래에 나와 있습니다.

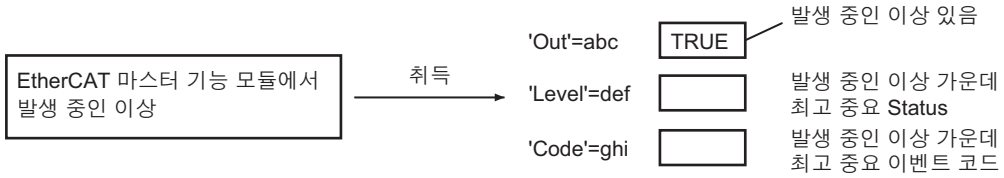
LD



ST

```
abc:=GetECError(def, ghi);
```

EtherCAT 마스터 기능 모듈에서 발생 중인 통신 포트 이상, 마스터 이상, 슬레이브 이상을 검출합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_ErrSta	내장 EtherCAT 이상	WORD	EtherCAT 마스터 기능 모듈 이상의 집약(*2)
_EC_PortErr(*1)	통신 포트 이상	WORD	EtherCAT 마스터용 통신 포트 이상의 집약(*2)
_EC_MstrErr(*1)	마스터 이상	WORD	EtherCAT 마스터 이상과 EtherCAT 마스터가 감지한 슬레이브 이상의 집약(*2)
_EC_SlavErr	슬레이브 이상	WORD	EtherCAT 슬레이브 전체의 이상 Status의 집약(*2)

*1 이 명령이 취득하는 것은 _EC_PortErr(통신 포트 이상)과 _EC_MstrErr(마스터 이상)에 해당하는 이상입니다.

*2 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"를 참조하십시오.

사용상의 주의 사항

버전 관련 정보

이 명령에서 슬레이브 이상을 검출할 수 있는 것은 CPU 유니트 Ver.1.02 이상입니다.

샘플 프로그램

'ResetMCError 명령(2-768페이지)'의 샘플 프로그램을 참조하십시오.

SetInfo

사용자 정보를 발생시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SetInfo	사용자 정보 발생	FUN	(@)SetInfo EN ENO Code Out Info1 Info2	SetInfo(Code, Info1, Info2);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Code	이벤트 코드	입력	발생시키는 사용자 정보의 이벤트 코드	40001~60000	–	40001
Info1	부속 정보 1		사용자 정보 발생과 함께 이벤트 로그에 기록하는 값	데이터 형식에 따름		(*)
Info2	부속 정보 2					
Out	반환값	출력	항상 TRUE	TRUE만	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

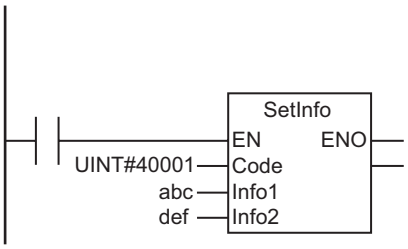
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Code							○													
Info1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Info2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
Out	○																			

기능

이벤트 코드 'Code'로 지정한 사용자 정보를 발생시킵니다.
또한, 이벤트 코드의 중요도에 대응하는 사용자 이벤트 로그 영역에, 발생 시각, 이벤트 코드 'Code', 이벤트 중요도, 부속 정보 'Info1', 'Info2'의 값을 기록합니다.

기술 예가 아래에 나와 있습니다. 이벤트 코드 40001에 대응하는 사용자 정보를 발생시킵니다. 부속 정보로서 변수 abc, def의 값을 기록합니다.

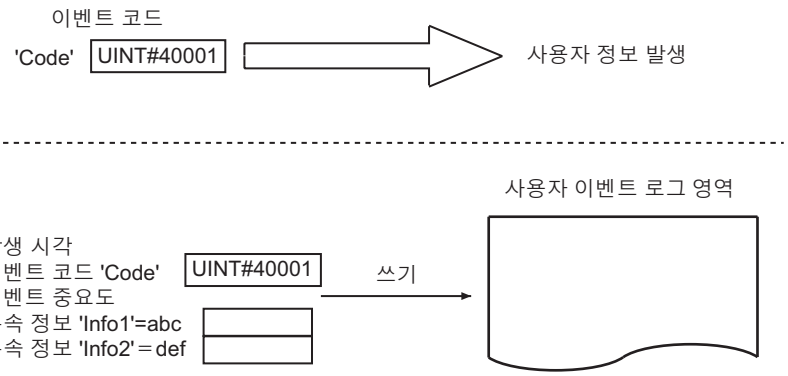
LD



ST

```
SetInfo(UINT#40001, abc, def);
```

이벤트 코드 'Code'로 지정한 사용자 정보를 발생시킵니다.
또한, 이벤트 코드의 중요도에 대한 사용자 이벤트 로그 영역에 발생 시각,
이벤트 코드 'Code', 이벤트 중요도, 부속 정보 'Info1', 'Info2'의 값을 기록합니다.

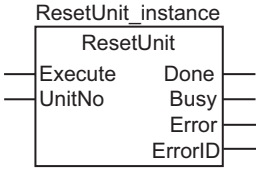


사용상의 주의 사항

- 'Info1', 'Info2'에 전달하는 입력 파라메터에는 반드시 변수를 지정하십시오. 부속 정보를 사용하지 않는 경우라도 더미 변수를 지정하십시오. 정수를 지정하면 빌드 시에 이상 상태가 됩니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 됩니다.
  - 'Code'의 값이 유효 범위 밖인 경우

ResetUnit

CPU 고기능 유니트 또는 고기능 I/O 유니트를 재시작합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ResetUnit	유니트 재시작	FB		ResetUnit_instance(Execute, UnitNo, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitNo	유니트 번호	입력	재시작하는 유니트 번호	_CBU_No00~ _CBU_No15, _SIO_No00~ _SIO_No95	–	_CBU _No00

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitNo	열거 형식 _eUnitNo 열거자는 기능 설명 참조																			

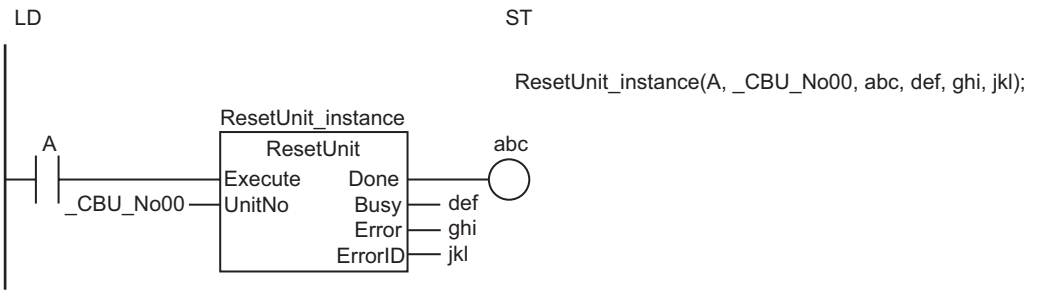
기능

CPU 고기능 유니트 또는 고기능 I/O 유니트를 재시작합니다.
재시작할 유니트를 유니트 번호 'UnitNo'로 지정합니다.

'UnitNo'의 데이터 형식은 열거 형식 _eUnitNo입니다. 열거자의 의미는 다음과 같습니다.

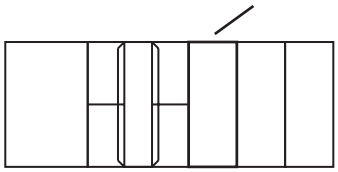
열거자	의미
_CBU_No00~_CBUNo15	CPU 고기능 유니트의 유니트 번호 00~15
_SIO_No00~_SIO_No95	고기능 I/O 유니트의 호기 번호 00~95

'UnitNo'=_CBU_No00인 경우의 예가 아래에 나와 있습니다. 유니트 번호 0인 CPU 고기능 유니트를 재시작합니다.



유니트 번호 'UnitNo'인 CPU 고기능 유니트 또는 고기능 I/O 유니트를 재시작합니다.

'UnitNo'＝_CBU_No00 유니트 번호 0인 CPU 고기능 유니트를 재시작



사용상의 주의 사항

- 'UnitNo'로 지정한 유니트가 재시작 중인 경우, 이 명령은 이상 상태가 되지 않습니다. 그대로 'Busy'의 값이 TRUE가 되며, 재시작 완료 시에 'Done'의 값이 TRUE가 됩니다. 재시작 요구는 큐잉되지 않습니다.
- 운전 시작 시에 'Execute'의 값이 TRUE인 경우, 유니트의 재시작을 실행합니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'UnitNo'의 값이 유효 범위 밖인 경우
  - 'UnitNo'로 지정한 유니트가 존재하지 않는 경우
  - 재시작에 실패한 경우

샘플 프로그램

Trigger의 값이 FALSE→TRUE가 되면, 유니트 번호 0인 시리얼 커뮤니케이션 유니트의 시리얼 포트 1의 전송 속도를 38400bps로 설정하고 유니트를 재시작합니다.

글로벌 변수의 정의

글로벌 변수

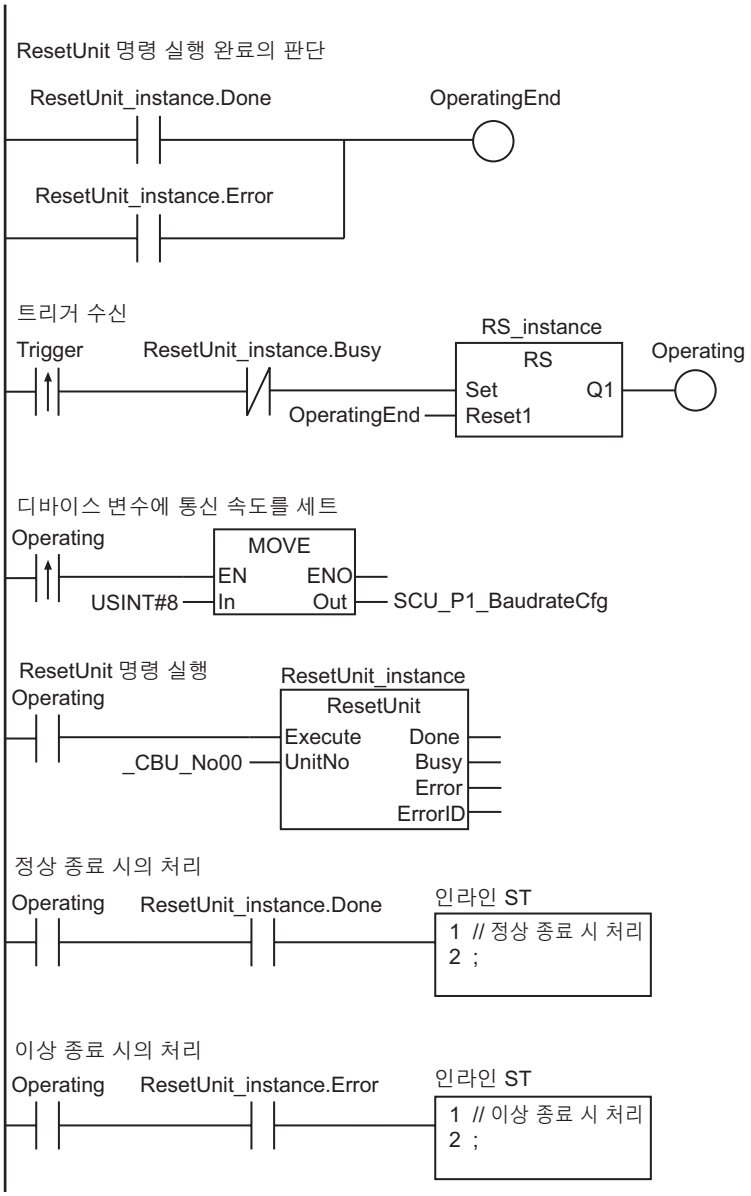
명칭	데이터형식	초기값	할당대상 *1	유지	코멘트
SCU_P1_BaudrateCfg	USINT	0	IOBus://rack#0/slot#0/P1_BaudrateCfg	☑	전송 속도

*1. 시리얼 커뮤니케이션 유니트를 랙 번호 0의 슬롯 번호 0에 장착한 경우의 할당대상입니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	RS_instance	RS		
	ResetUnit_instance	ResetUnit		

외부 변수	명칭	데이터 형식	코멘트
	SCU_P1_BaudrateCfg	USINT	전송 속도



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	ResetUnit_instance	ResetUnit		

외부 변수	명칭	데이터 형식	코멘트
	SCU_P1_BaudrateCfg	USINT	전송 속도

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (ResetUnit_instance.Busy=FALSE) ) THEN
    OperatingStart:=TRUE;
    Operating:=TRUE;
END_IF;
LastTrigger:=Trigger;

// ResetUnit_instance 초기화 및 디바이스 변수에 전송 속도 세트
IF (OperatingStart=TRUE) THEN
    ResetUnit_instance(Execute:=FALSE);
    SCU_P1_BaudrateCfg:=USINT#8;
    OperatingStart:=FALSE;
END_IF;

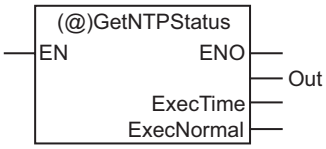
// ResetUnit 명령 실행
IF (Operating=TRUE) THEN
    ResetUnit_instance(
        Execute:=TRUE,          // 기동 조건
        UnitNo  :=_CBU_No00); // 유니트 번호

    IF (ResetUnit_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating:=FALSE;
    END_IF;

    IF (ResetUnit_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating:=FALSE;
    END_IF;
END_IF;
```

GetNTPStatus

NTP의 Status를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetNTP Status	NTP Status 읽어오기	FUN		GetNTPStatus(ExecTime, ExecNormal);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	반환값	출력	항상 TRUE	TRUE만	–	–
ExecTime	NTP 최종 정상 동작 시각		NTP 최종 정상 동작 시각	데이터 형식에 따름	연월일시분초	
Exec Normal	NTP 정상 종료 플래그		TRUE : 정상 종료 FALSE : 이상 종료		–	

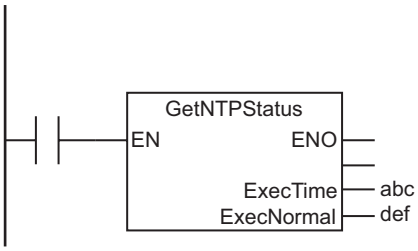
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			
ExecTime																			○	
Exec Normal	○																			

기능

NTP의 Status를 읽어옵니다. 읽어오는 정보는 NTP 최종 정상 동작 시각 'ExecTime', 정상 종료 플래그 'ExecNormal'입니다.

기술 예가 아래에 나와 있습니다.

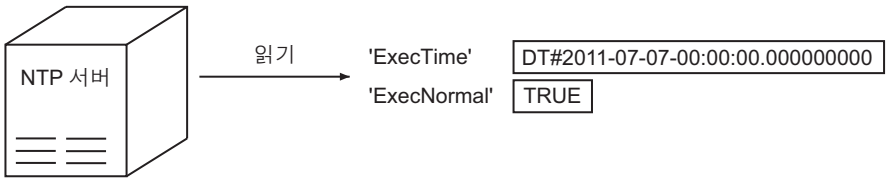
LD



ST

```
GetNTPStatus(abc, def);
```

NTP의 Status를 읽어옵니다.
NTP 최종 정상 동작 시각이 2011년 7월 7일 0시 0분 0초인 경우 'ExecTime'과 'ExecNormal'의 값은 다음과 같습니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_NTPResult	NTP Status	(*)	NTP Status 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

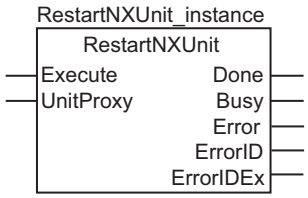
* _sNTP_RESULT

사용상의 주의 사항

- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 이 명령에서 읽어오는 정보는 시스템 정의 변수 '_EIP_NTPResult'의 내용입니다. 이 변수에는 직접 액세스할 수 없습니다. 이 변수의 내용을 읽어오려면 반드시 이 명령을 사용하십시오.

RestartNXUnit

통신 커플러 유니트 또는 NX 유니트를 재시작합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Restart NXUnit	NX 유니트 재시작	FB		RestartNXUnit_instance(Execute, UnitProxy, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitProxy	지정 유니트	입력	재시작할 슬레이브 터미널의 통신 커플러 유니트 또는 NX 유니트	–	–	(*)

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitProxy	구조체_sNXUNIT_ID 자세한 내용은 기능 설명 참조																			

기능

통신 커플러 유니트 또는 NX 유니트를 재시작합니다.
지정된 유니트를 단독으로 재시작할 수 있습니다.
단, 통신 커플러 유니트만 단독으로 재시작할 수는 없습니다.
통신 커플러 유니트를 지정한 경우는 거기에 연결되어 있는 NX 유니트도 모두 재시작합니다.
재시작할 유니트는 'UnitProxy'로 지정합니다.

'UnitProxy'의 데이터 형식은 구조체_sNXUNIT_ID입니다. 사양은 다음과 같습니다.

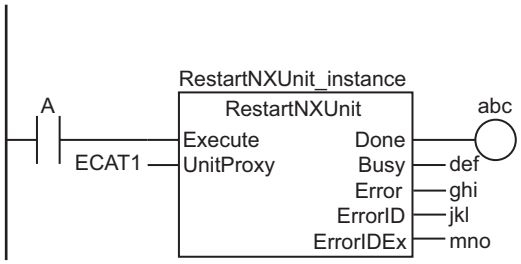
변수	명칭	내용	데이터 형식
UnitProxy	지정 유니트	지정할 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소	UINT
IPAdr	IP 주소	통신 커플러 유니트의 IP 주소	BYTE[5]
UnitNo	유니트 번호	지정할 유니트의 유니트 번호	UDINT
Path	경로	지정할 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

'UnitProxy'에는 지정할 통신 커플러 유니트 또는 NX 유니트에 할당된 디바이스 변수를 전달하십시오.

기술 예

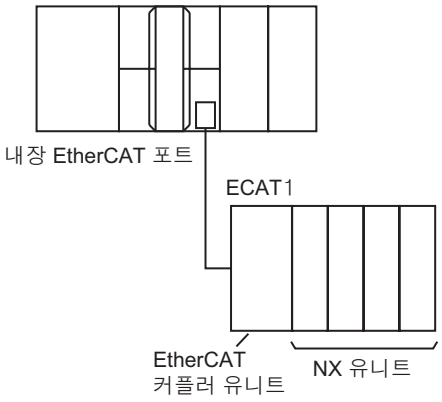
EtherCAT 커플러 유니트와 거기에 연결되어 있는 NX 유니트를 모두 재시작하는 경우의 예가 아래에 나와 있습니다. EtherCAT 커플러 유니트에는 _sNXUNIT_ID 형식의 'ECAT1'이라는 변수가 할당되어 있는 것으로 합니다.

LD



ST

```
RestartNXUnit_instance(A, ECAT1,
                       abc, def, ghi, jkl, mno);
```



EtherCAT 커플러 유니트 'ECAT1'과 거기에 연결되어 있는 NX 유니트를 모두 재시작합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브가 통신 가능한지 여부를 나타냅니다. TRUE : 통신 가능 FALSE: 통신 불가능

참고

통신 커플러 유니트 또는 NX 유니트에 다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 유니트 재시작 시

1 **📖 'NX_ChangeWriteMode 명령(2-793페이지)'을 사용하여 데이터를 쓸 수 있는 모드로 대상 유니트를 변경합니다.**

2 **📖 'NX_WriteObj 명령(2-854페이지)'을 사용하여 대상 유니트에 데이터를 씁니다.**

3 **📖 'NX_SaveParam 명령(2-797페이지)'을 사용하여, 기입된 데이터를 저장합니다.**

4 **이 명령을 사용하여 대상 유니트를 재시작합니다.**

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'UnitProxy'의 경우, 모션 제어 축(데이터 형식_sAXIS_REF)에 할당된 유니트를 지정하면 모션 제어 모듈의 컨트롤러 이상이 발생합니다. 'ResetMCError(2-768페이지)'으로 컨트롤러 이상을 해제하십시오.
- 'UnitProxy'에는 Sysmac Studio의 I/O 맵에서 통신 커플러 유니트 또는 NX 유니트에 할당한 디바이스 변수를 지정합니다. 디바이스 변수 할당 방법은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어 편(SBCA-359H 이상)"을 참조하십시오.
- 이 명령 또는 'NX_ChangeWriteMode(2-793페이지)'의 실행 중에 이 명령을 실행하려고 한 경우, 나중에 실행하려고 한 명령은 대기 처리됩니다. 최대 대기 처리 수는 192입니다. 192를 초과하면 이상이 발생합니다. 또한 대기 처리되어 있는 시간은 타임아웃 대상에서 제외됩니다.
- 이 명령이 대기 처리되어 있는 동안 'Busy'의 값은 TRUE가 됩니다.
- 이 명령은 NX 메시지 통신 이상과 관련된 명령입니다. NX 메시지 통신 이상과 관련된 명령을 한번에 여러 개 실행하면 NX 메시지 통신 이상이 발생할 가능성이 있습니다. NX 메시지 통신 이상과 관련된 명령에 대해서는 'NX 메시지 통신 이상 관련 명령(A-164페이지)'을 참조하십시오.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 실행 중에는 이 명령을 실행할 수 없습니다. 실행하면 이상이 발생합니다.

• 이상이 발생한 경우, 'Error'는 TRUE가 됩니다. 'ErrorID', 'ErrorIDEx'의 값과 그에 대응하는 이상 내용은 다음과 같습니다.

'ErrorID'의 값	'ErrorIDEx'의 값	이상 내용
16#0419	16#00000000	'UnitProxy'의 데이터 형식이 잘못되었습니다.
16#2C00	16#00000401	지정한 유니트가 이 명령의 대상에서 제외됩니다.
	16#00002231	지정된 유니트가 초기 처리 중이므로 이 명령을 실행할 없습니다. 정상 동작 중이 되기를 기다린 후 이 명령을 실행하십시오.
	16#00002601	지정한 유니트가 명령을 지원하지 않습니다. 유니트의 버전을 확인하십시오.
	16#00010000	지정된 유니트가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00110000	지정된 포트 번호가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00120000 16#00130000 16#00150000	'UnitProxy'의 값이 잘못되었습니다. 지정할 통신 커플러 유니트를 나타내는 변수를 다시 설정하십시오.
	16#00140000	지정된 노드 주소가 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00310000	지정된 유니트가 연결 대상이 아닙니다. 유니트의 버전을 확인하십시오.
	16#80010000	지정된 유니트가 다른 작업 중입니다. 명령을 다시 실행하십시오.
	16#80020000 16#80030000 16#81030000 16#82000000	통신 네트워크에서 이상이 발생했습니다. 통신 부하를 줄이십시오.
	16#80040000 16#81000000 16#82010000 16#82040000 16#82050000 16#90000000	통신 네트워크에서 이상이 발생했습니다. 유니트나 케이블의 연결을 확인하십시오. 유니트의 전원이 켜져 있는지 확인하십시오.
16#2C01	16#00000000	이 명령 또는 NX_ChangeWriteMode 명령이 192를 초과하여 대기 처리되었습니다.
16#2C02	16#00000000	통신 실행 중에 타임아웃이 발생했습니다.
16#2C05	–	EtherCAT 네트워크에서 이상이 발생했습니다. 'UnitProxy'의 값이나 EtherCAT 네트워크 구성을 확인하십시오.
16#2C06	16#00000000	지정된 유니트는 Sysmac Studio에서의 지시로 이미 재시작이 실행되었습니다. 따라서 이 명령을 실행할 필요가 없습니다.
16#2C07	16#00000000	지정된 유니트의 슬레이브 노드 주소에 이 명령의 대상이 아닌 슬레이브가 연결되어 있습니다. 'UnitProxy'의 값이나 EtherCAT 네트워크 구성을 확인하십시오.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.
단, 다음은 버전에 따라 '지정된 NX 유니트를 재시작하는 기능'을 지원하지 않는 경우도 있습니다.

- CPU 유니트
- Sysmac Studio
- EtherCAT 커플러 유니트
- NX 유니트

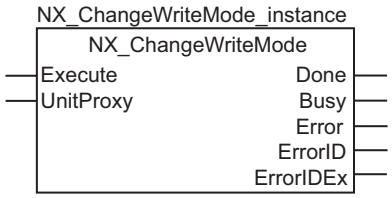
'지정된 NX 유니트를 재시작하는 기능'을 지원하지 않는 버전을 사용하는 경우는 재시작할 유니트에 통신 커플러 유니트만 지정할 수 있습니다.
'지정된 NX 유니트를 재시작하는 기능'을 지원하는 버전은 "NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361C 이상)"를 참조하십시오.

샘플 프로그램

'NX_WriteObj(2-854페이지)'의 샘플 프로그램을 참조하십시오.

NX_ChangeWriteMode

통신 커플러 유니트 또는 NX 유니트를 데이터를 쓸 수 있는 모드로 변경합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_Change WriteMode	NX 유니트 쓰기 모드 변경	FB		NX_ChangeWriteMode_instance(Execute, UnitProxy, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitProxy	지정 유니트	입력	모드를 변경할 유니트	–	–	(*)

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitProxy	구조체_sNXUNIT_ID 자세한 내용은 기능 설명 참조																			

기능

통신 커플러 유니트 또는 NX 유니트를 데이터를 쓸 수 있는 모드로 변경합니다.
모드를 변경할 유니트를 'UnitProxy'로 지정합니다.
'Done'의 값이 TRUE가 되면 데이터를 쓸 수 있는 상태가 됩니다.

'UnitProxy'의 데이터 형식은 구조체_sNXUNIT_ID입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식
UnitProxy	지정 유니트	지정할 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소	UINT
IPAdr	IP 주소	통신 커플러 유니트의 IP 주소	BYTE[5]
UnitNo	유니트 번호	지정할 유니트의 유니트 번호	UDINT
Path	경로	지정할 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

'UnitProxy'에는 지정할 유니트에 할당된 디바이스 변수를 전달하십시오.

관련 명령과 실행 순서

이 명령은 통신 커플러 유니트 또는 NX 유니트에 다음과 같은 속성을 가진 데이터를 쓰는 경우에 사용합니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 유니트 재시작 시

관련 명령과 실행 순서는 다음과 같습니다.

1 **이 명령을 사용하여 데이터를 쓸 수 있는 모드로 대상 유니트를 변경합니다.**

2 **🕮 'NX_WriteObj(2-854페이지)'을 사용하여 대상 유니트에 데이터를 씁니다.**

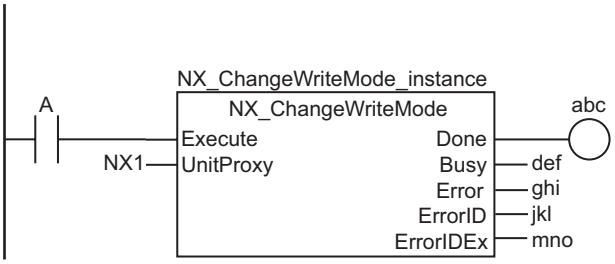
3 **🕮 'NX_SaveParam(2-797페이지)'을 사용하여, 기입된 데이터를 저장합니다.**

4 **🕮 'RestartNXUnit(2-788페이지)'을 사용하여 대상 유니트를 재시작합니다.**

기술 예

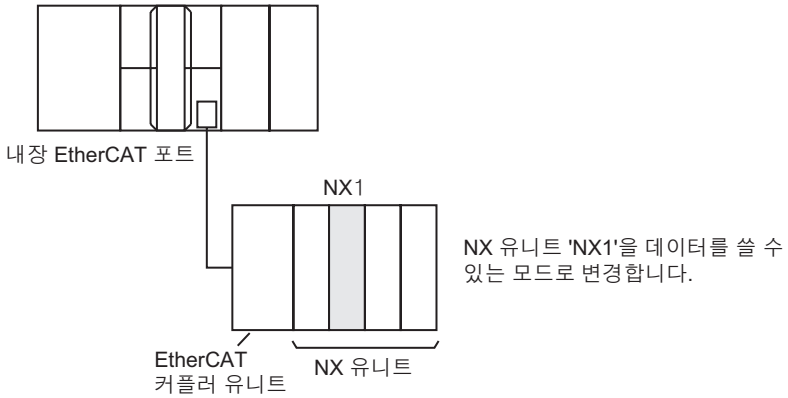
NX 유니트 'NX1'을 데이터를 쓸 수 있는 모드로 변경하는 경우의 예가 아래에 나와 있습니다. 대상 NX 유니트에는 _sNXUNIT_ID 형식의 'NX1'이라는 변수가 할당되어 있는 것으로 합니다.

LD



ST

```
NX_ChangeWriteMode_instance(A, NX1, abc, def,
                            ghi, jkl, mno);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브가 통신 가능한지 여부를 나타냅니다. TRUE : 통신 가능 FALSE: 통신 불가능

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'UnitProxy'의 경우, 모션 제어 축(데이터 형식_sAXIS_REF)에 할당된 유니트를 지정하면 모션 제어 모듈의 컨트롤러 이상이 발생합니다. 그 경우는 'ResetMCError(2-768페이지)'으로 컨트롤러 이상을 해제하십시오.
- 'UnitProxy'에는 Sysmac Studio의 I/O 맵에서 통신 커플러 유니트 또는 NX 유니트에 할당한 디바이스 변수를 지정합니다. 디바이스 변수 할당 방법은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359H 이상)"을 참조하십시오.
- 이 명령 또는 'RestartNXUnit(2-788페이지)'의 실행 중에 이 명령을 실행하려고 한 경우, 나중에 실행하려고 한 명령은 대기 처리됩니다. 최대 대기 처리 수는 192입니다. 192를 초과하면 이상이 발생합니다. 또한 대기 처리되어 있는 시간은 타임아웃 대상에서 제외됩니다.
- 이 명령이 대기 처리되어 있는 동안 'Busy'의 값은 TRUE가 됩니다.
- 이 명령은 NX 메시지 통신 이상과 관련된 명령입니다. NX 메시지 통신 이상과 관련된 명령을 한번에 여러 개 실행하면 NX 메시지 통신 이상이 발생할 가능성이 있습니다. NX 메시지 통신 이상과 관련된 명령에 대해서는 'NX 메시지 통신 이상 관련 명령(A-164페이지)'을 참조하십시오.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 실행 중에는 이 명령을 실행할 수 없습니다. 실행하면 이상이 발생합니다.

- 이상이 발생한 경우, 'Error'는 TRUE가 됩니다. 'ErrorID', 'ErrorIDEx'의 값과 그에 대응하는 이상 내용은 다음과 같습니다.

'ErrorID'의 값	'ErrorIDEx'의 값	이상 내용
16#0419	16#00000000	'UnitProxy'의 데이터 형식이 잘못되었습니다.
16#2C00	16#00001101	'UnitProxy'의 값이 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00002621	다른 클라이언트가 실행한 이 명령을 처리 중이므로 모드를 변경할 수 없습니다. 잠시 후 다시 실행하십시오.
	16#00010000	지정된 유니트가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00110000	지정된 포트 번호가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00120000 16#00130000 16#00150000	'UnitProxy'의 값이 잘못되었습니다. 지정할 유니트를 나타내는 변수를 다시 설정하십시오.
	16#00140000	지정된 노드 주소가 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00310000	지정된 유니트가 연결 대상이 아닙니다. 유니트의 버전을 확인하십시오.
	16#80010000	지정된 유니트가 다른 작업 중입니다. 명령을 다시 실행하십시오.
	16#80020000 16#80030000 16#81030000 16#82000000	통신 네트워크에서 이상이 발생했습니다. 통신 부하를 줄이십시오.
	16#80040000 16#81000000 16#82010000 16#82040000 16#82050000 16#90000000	통신 네트워크에서 이상이 발생했습니다. 유니트나 케이블의 연결을 확인하십시오. 유니트의 전원이 켜져 있는지 확인하십시오.
16#2C01	16#00000000	이 명령 또는 RestartNXUnit 명령이 192를 초과하여 대기 처리되었습니다.
16#2C02	16#00000000	통신 실행 중에 타임아웃이 발생했습니다.
16#2C05	–	EtherCAT 네트워크에서 이상이 발생했습니다. 'UnitProxy'의 값이나 EtherCAT 네트워크 구성을 확인하십시오.
16#2C07	16#00000000	지정된 유니트의 슬레이브 노드 주소에 이 명령의 대상이 아닌 슬레이브가 연결되어 있습니다. 'UnitProxy'의 값이나 EtherCAT 네트워크 구성을 확인하십시오.

버전 관련 정보

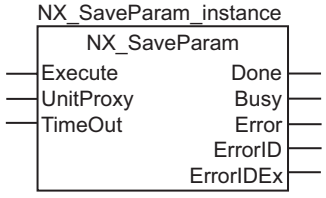
이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.

샘플 프로그램

'NX_WriteObj(2-854페이지)'의 샘플 프로그램을 참조하십시오.

NX_SaveParam

통신 커플러 유니트 또는 NX 유니트에 기록한 데이터를 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_Save Param	NX 유니트 파라메터 저장	FB		NX_SaveParam_instance(Execute, UnitProxy, TimeOut, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitProxy	지정 유니트	입력	데이터를 저장할 유니트	–	–	(*)
TimeOut	타임아웃 시간		타임아웃 시간 0인 경우는 2.0s	0~60000	ms	2000 (2.0s)

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitProxy	구조체_sNXUNIT_ID 자세한 내용은 기능 설명 참조																			
TimeOut							○													

기능

통신 커플러 유니트 또는 NX 유니트에 기록한 데이터를 저장합니다.
데이터를 저장할 유니트를 'UnitProxy'로 지정합니다.
'Done'의 값이 TRUE가 되면 데이터 저장이 완료됩니다.
데이터를 쓰는 명령은 'NX_WriteObj(2-854페이지)'입니다.
이 명령 실행 후에는 전원 OFF가 발생해도 전원 OFF 유지 속성이 있는 데이터 값은 유지됩니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다. 그 경우, 유니트의 데이터 저장은 실행되지 않습니다.

'UnitProxy'의 데이터 형식은 구조체_sNXUNIT_ID입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식
UnitProxy	지정 유니트	지정할 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소	UINT
IPAdr	IP 주소	통신 커플러 유니트의 IP 주소	BYTE[5]
UnitNo	유니트 번호	지정할 유니트의 유니트 번호	UDINT
Path	경로	지정할 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

'UnitProxy'에는 지정할 유니트에 할당된 디바이스 변수를 전달하십시오.

관련 명령과 실행 순서

이 명령은 통신 커플러 유니트 또는 NX 유니트에 기록한 데이터의 속성에 따라 다른 관련 명령과 연계하여 실행해야 합니다. 각각의 경우의 실행 순서는 다음과 같습니다.

● 실행 순서 1

다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 유니트 재시작 시

1 **🕮 'NX_ChangeWriteMode(2-793페이지)'을 사용하여 데이터를 쓸 수 있는 모드로 대상 유니트를 변경합니다.**

2 **🕮 'NX_WriteObj(2-854페이지)'을 사용하여 대상 유니트에 데이터를 씁니다.**

3 **이 명령을 사용하여, 기입된 데이터를 저장합니다.**

4 **🕮 'RestartNXUnit(2-788페이지)'을 사용하여 대상 유니트를 재시작합니다.**

● 실행 순서 2

다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 쓴 직후

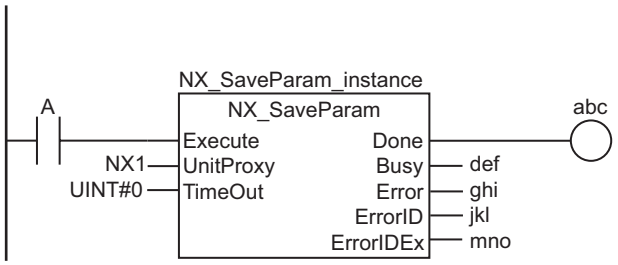
1 **🕮 'NX_WriteObj(2-854페이지)'을 사용하여 대상 유니트에 데이터를 씁니다.**

2 **이 명령을 사용하여, 기입된 데이터를 저장합니다.**

기술 예

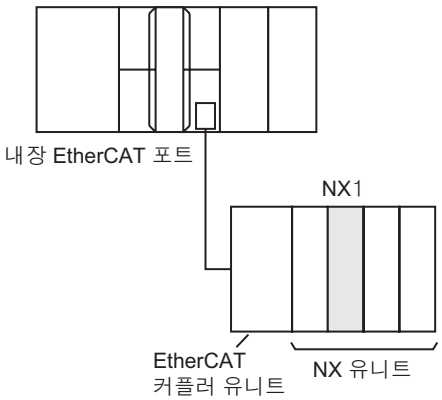
NX 유니트 'NX1'에 쓴 데이터를 저장하는 경우의 예가 아래에 나와 있습니다. 대상 NX 유니트에는 _sNXUNIT_ID 형식의 'NX1'이라는 변수가 할당되어 있는 것으로 합니다.

LD



ST

```
NX_SaveParam_instance(A, NX1, UINT#0,
                      abc, def, ghi, jkl, mno);
```



NX 유니트 'NX1'에 기록한 데이터를 저장합니다.

데이터 쓰기는 NX_WriteObj 명령으로 실행해 두십시오.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브가 통신 가능한지 여부를 나타냅니다. TRUE : 통신 가능 FALSE: 통신 불가능

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'UnitProxy'로 지정한 유니트가 데이터 저장 중인 경우, 이 명령은 이상이 되지 않습니다. 그대로 'Busy'의 값이 TRUE가 되고, 데이터 저장 완료 시에 'Done'의 값이 TRUE가 됩니다. 데이터 저장 요구는 대기 처리되지 않습니다.
- 유니트에 데이터를 쓰지 않고 이 명령을 실행해도 이상이 되지는 않습니다.
- 유니트에는 데이터 쓰기 가능 횟수가 제한되어 있는 것이 있습니다. 자세한 내용은 각 유니트의 설명서를 참조하십시오.
- 'UnitProxy'에는 Sysmac Studio의 I/O 맵에서 통신 커플러 유니트 또는 NX 유니트에 할당한 디바이스 변수를 지정합니다. 디바이스 변수 할당 방법은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편 (SBCA-359H 이상)"을 참조하십시오.
- 전원 OFF 시, 유지 속성이 있는 데이터를 쓰고 저장하고 싶은 경우는 'NX_WriteObj(2-854페이지)' 실행 후에 이 명령을 실행하십시오. 이 명령을 실행하지 않고 대상 유니트가 재시작되면 NX 개체의 데이터는 쓰기 전의 값으로 돌아갑니다.
- 이 명령은 NX 메시지 통신 이상과 관련된 명령입니다. NX 메시지 통신 이상과 관련된 명령을 한번에 여러 개 실행하면 NX 메시지 통신 이상이 발생할 가능성이 있습니다. NX 메시지 통신 이상과 관련된 명령에 대해서는 'NX 메시지 통신 이상 관련 명령(A-164페이지)'을 참조하십시오.

• 이상이 발생한 경우, 'Error'는 TRUE가 됩니다. 'ErrorID', 'ErrorIDEx'의 값과 그에 대응하는 이상 내용은 다음과 같습니다.

'ErrorID'의 값	'ErrorIDEx'의 값	이상 내용
16#0400	16#00000000	• 'UnitProxy'의 값이 유효 범위 밖입니다. • 'TimeOut'의 값이 유효 범위 밖입니다.
16#0419	16#00000000	'UnitProxy'의 데이터 형식이 잘못되었습니다.
16#2C00	16#0000250F	하드웨어 액세스에 실패했습니다. 명령을 다시 실행하십시오.
	16#00002602	지정한 유니트에서는 이 명령을 사용할 없습니다. 유니트의 버전을 확인하십시오.
	16#00010000	지정된 유니트가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00110000	지정된 포트 번호가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00120000 16#00130000 16#00150000	'UnitProxy'의 값이 잘못되었습니다. 지정할 유니트를 나타내는 변수를 다시 설정하십시오.
	16#00140000	지정된 노드 주소가 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00310000	지정된 유니트가 연결 대상이 아닙니다. 유니트의 버전을 확인하십시오.
	16#80010000	지정된 유니트가 다른 작업 중입니다. 명령을 다시 실행하십시오.
	16#80020000 16#80030000 16#81030000 16#82000000	통신 네트워크에서 이상이 발생했습니다. 통신 부하를 줄이십시오.
	16#80040000 16#81000000 16#82010000 16#82040000 16#82050000 16#90000000	통신 네트워크에서 이상이 발생했습니다. 유니트나 케이블의 연결을 확인하십시오. 유니트의 전원이 켜져 있는지 확인하십시오.
16#2C01	16#00000000	동시에 실행할 수 있는 명령 수를 초과했습니다.
16#2C02	16#00000000	통신 실행 중에 타임아웃이 발생했습니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.

샘플 프로그램

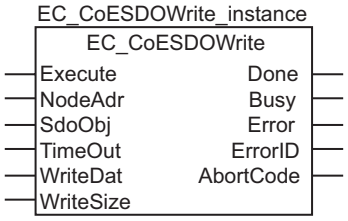
'NX_WriteObj(2-854페이지)'의 샘플 프로그램을 참조하십시오.

EtherCAT 통신 명령

명령	명칭	페이지
EC_CoESDOWrite	CoE SDO 쓰기	2-804
EC_CoESDORead	CoE SDO 읽기	2-807
EC_StartMon	패킷 모니터 기능 시작	2-813
EC_StopMon	패킷 모니터 기능 정지	2-819
EC_SaveMon	패킷 저장	2-821
EC_CopyMon	패킷 전송	2-823
EC_DisconnectSlave	EtherCAT 슬레이브 이탈	2-825
EC_ConnectSlave	EtherCAT 슬레이브 재가입	2-833
EC_ChangeEnable Setting	EtherCAT 슬레이브의 유효/무효 전환	2-835
NX_WriteObj	NX 개체 쓰기	2-854
NX_ReadObj NX	개체 읽기	2-869

EC_CoESDOWrite

EtherCAT 네트워크상의 지정된 슬레이브가 가진 CoE(*)의 개체에 값을 씁니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_CoESDO Write	CoE SDO 쓰기	FB		EC_CoESDOWrite_instance(Execute, NodeAdr, SdoObj, TimeOut, WriteDat, WriteSize, Done, Busy, Error, ErrorID, AbortCode);

* CAN application protocol over EtherCAT의 약자입니다.

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
NodeAdr	슬레이브 노드 주소	입력	액세스할 슬레이브의 노드 주소	1~192	–	–
SdoObj	SDO 파라메터		SDO 파라메터	–		
TimeOut	타임아웃 시간		0 : 2.0s 1~65535 : 0.1~6553.5s	데이터 형식에 따름	0.1s	20 (2.0s)
WriteDat	쓰기 데이터		쓰기 데이터		–	–
WriteSize	쓰기 데이터 크기		쓰기 데이터 크기(*)	1~2048	바이트	
AbortCode	Abort 코드	출력	CoE로 규정된 SDO 액세스의 응답 코드 0 : 정상 종료	데이터 형식에 따름	–	–

* 쓰기 데이터가 BOOL 형식 또는 BOOL 형식 배열 등인 경우, 쓰기 데이터 크기가 1바이트 미만인 경우가 있습니다. 그 경우, 'WriteSize'의 값은 1로 하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
NodeAdr							○													
SdoObj	구조체 _sSDO_ACCESS에 대한 자세한 내용은 기능 설명 참조																			
TimeOut							○													
WriteDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			
WriteSize							○													
AbortCode				○																

기능

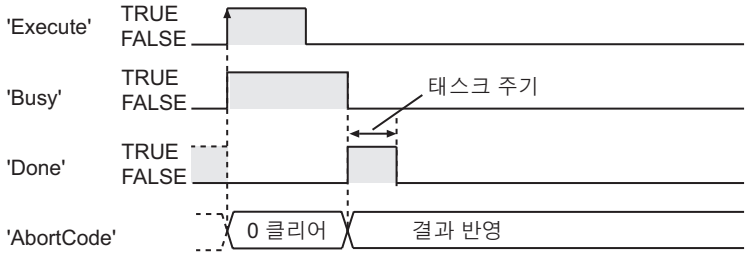
슬레이브 노드 주소 'NodeAdr'로 지정한 노드가 가진 CoE의 개체에 데이터를 씁니다.
쓸 데이터는 'WriteDat'의 내용입니다. 쓸 데이터 크기는 'WriteSize'로 지정합니다.
SDO 파라메터는 'SdoObj'로 지정합니다.

'SdoObj'의 데이터 형식은 구조체 _sSDO_ACCESS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
SdoObj	SDO 파라메터	SDO 파라메터	_sSDO_ACCESS	–	–	–
Index	인덱스	CoE로 정의되는 Object Dictionary의 인덱스 번호	UINT	1~65535	–	–
Subindex	하위 인덱스	CoE로 정의되는 Object Dictionary의 하위 인덱스 번호	USINT	데이터 형식에 따름		
IsComplete Access	Complete 액세스	SDO의 Complete 액세스 기능 지정 TRUE : 모든 하위 인덱스의 데이터에 액세스 FALSE : 지정한 하위 인덱스의 데이터에 액세스	BOOL			

쓰기 완료 후, 타임아웃 시간 'TimeOut'으로 지정한 시간만큼 응답을 기다립니다. 응답은 'AbortCode'에 저장됩니다. 정상 종료 시의 'AbortCode'의 값은 0입니다. 단, 'AbortCode'에 값이 저장되는 것은 'ErrorID'의 값이 16#1804(SDO Abort 응답)인 경우에 한합니다.
'AbortCode'의 값과 의미는 슬레이브에 의해 규정됩니다. 슬레이브 측의 설명서를 참조하십시오.

타이밍 차트가 아래에 나와 있습니다. 'AbortCode'에 값이 저장되는 것은 명령 처리가 종료되어 'Busy'의 값이 FALSE가 될 때입니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브의 통신 가능 여부를 나타냅니다. TRUE : 통신 가능 FALSE : 통신 불가

참고

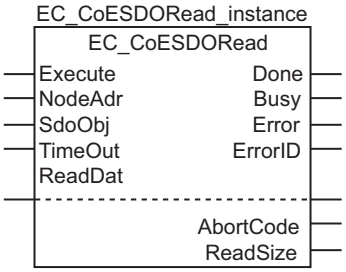
- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.
- SDO Abort 코드는 'SDO Abort 코드 목록 명령(A-165페이지)'을 참조하십시오.

사용상의 주의 사항

- 'WriteDat'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - EtherCAT 마스터가 메시지 통신 가능한 상태가 아닌 경우
  - 'NodeAdr'로 지정한 슬레이브가 존재하지 않는 경우
  - 'NodeAdr'로 지정한 슬레이브가 통신 가능한 상태가 아닌 경우
  - 슬레이브가 이상 응답을 반환한 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

EC_CoESDORead

EtherCAT 네트워크상의 지정된 슬레이브가 가진 CoE(*)의 개체에서 값을 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_CoESDO Read	CoE SDO 읽기	FB		EC_CoESDORead_instance(Execute, NodeAdr, SdoObj, TimeOut, ReadDat, Done, Busy, Error, ErrorID, AbortCode, ReadSize);

* CAN application protocol over EtherCAT의 약자입니다.

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
NodeAdr	슬레이브 노드 주소	입력	액세스할 슬레이브의 노드 주소	1~192	–	–
SdoObj	SDO 파라메터		SDO 파라메터	–		
TimeOut	타임아웃 시간		0 : 2.0s 1~65535 : 0.1~6553.5s	데이터 형식에 따름	0.1s	0 (2.0s)
AbortCode	Abort 코드	출력	CoE로 규정된 SDO 액세스의 응답 코드 0 : 정상 종료	데이터 형식에 따름	–	–
ReadSize	읽기 데이터 크기		읽어서 'ReadDat'에 저장한 데이터의 크기(*)		바이트	
ReadDat	읽기 데이터	입출력	읽기 데이터 저장용 버퍼	데이터 형식에 따름	–	–

* 읽기 데이터가 BOOL 형식 또는 BOOL 형식 배열 등인 경우, 읽기 데이터 크기가 1바이트 미만인 경우가 있습니다. 그 경우, 'ReadSize'의 값은 1이 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
NodeAdr							○													
SdoObj	구조체 _sSDO_ACCESS에 대한 자세한 내용은 기능 설명 참조																			
TimeOut							○													
AbortCode				○																
ReadSize							○													
ReadDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			

기능

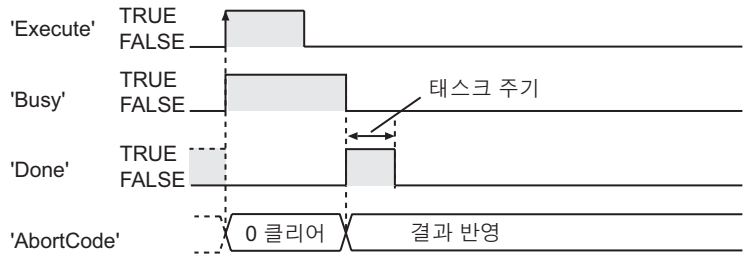
슬레이브 노드 주소 'NodeAdr'로 지정한 노드가 가진 CoE의 개체에서 데이터를 읽습니다.
읽은 데이터는 'ReadDat'에 저장합니다. 그리고 저장한 데이터 크기를 'ReadSize'에 저장합니다.
'ReadSize'의 값은 저장이 성공한 경우에만 유효합니다.
SDO 파라메터는 'SdoObj'로 지정합니다.

'SdoObj'의 데이터 형식은 구조체 _sSDO_ACCESS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
SdoObj	SDO 파라메터	SDO 파라메터	_sSDO_ ACCESS	–	–	–
Index	인덱스	CoE로 정의되는 Object Dictionary의 인덱스 번호	UINT	1~65535	–	–
Subindex	하위 인덱스	CoE로 정의되는 Object Dictionary의 하위 인덱스 번호	USINT	데이터 형식에 따름		
IsComplete Access	Complete 액세스	SDO의 Complete 액세스 기능 지정 TRUE : 모든 하위 인덱스의 데이터에 액세스 FALSE : 지정한 하위 인덱스의 데이터에 액세스	BOOL			

읽기 완료 후, 타임아웃 시간 'TimeOut'으로 지정한 시간만큼 응답을 기다립니다. 응답은 'AbortCode'에 저장됩니다. 정상 종료 시의 'AbortCode'의 값은 0입니다. 단, 'AbortCode'에 값이 저장되는 것은 'ErrorID'의 값이 16#1804(SDO Abort 응답)인 경우에 한합니다.
'AbortCode'의 값과 의미는 슬레이브에 의해 규정됩니다. 슬레이브 측의 설명서를 참조하십시오.

타이밍 차트가 아래에 나와 있습니다. 'AbortCode'에 값이 저장되는 것은 명령 처리가 종료되어 'Busy'의 값이 FALSE가 될 때입니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브의 통신 가능 여부를 나타냅니다. TRUE : 통신 가능 FALSE : 통신 불가

참고

- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.
- SDO Abort 코드는 'SDO Abort 코드 목록 명령(A-165페이지)'을 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - EtherCAT 마스터가 메시지 통신 가능한 상태가 아닌 경우
  - 'NodeAdr'로 지정한 슬레이브가 존재하지 않는 경우
  - 'NodeAdr'로 지정한 슬레이브가 통신 가능한 상태가 아닌 경우
  - 슬레이브가 이상 응답을 반환한 경우
  - 'ReadDat'의 크기보다 읽은 데이터 크기가 큰 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

EtherCAT의 SDO 메시지를 사용하여 오므론 서보 드라이버 R88D-KN01L-ECT 버전 V1.02에서 소프트웨어 버전을 읽습니다. 이 슬레이브의 노드 주소는 1로 합니다.
소프트웨어 버전의 개체 인덱스는 16#100A, 하위 인덱스는 0입니다.

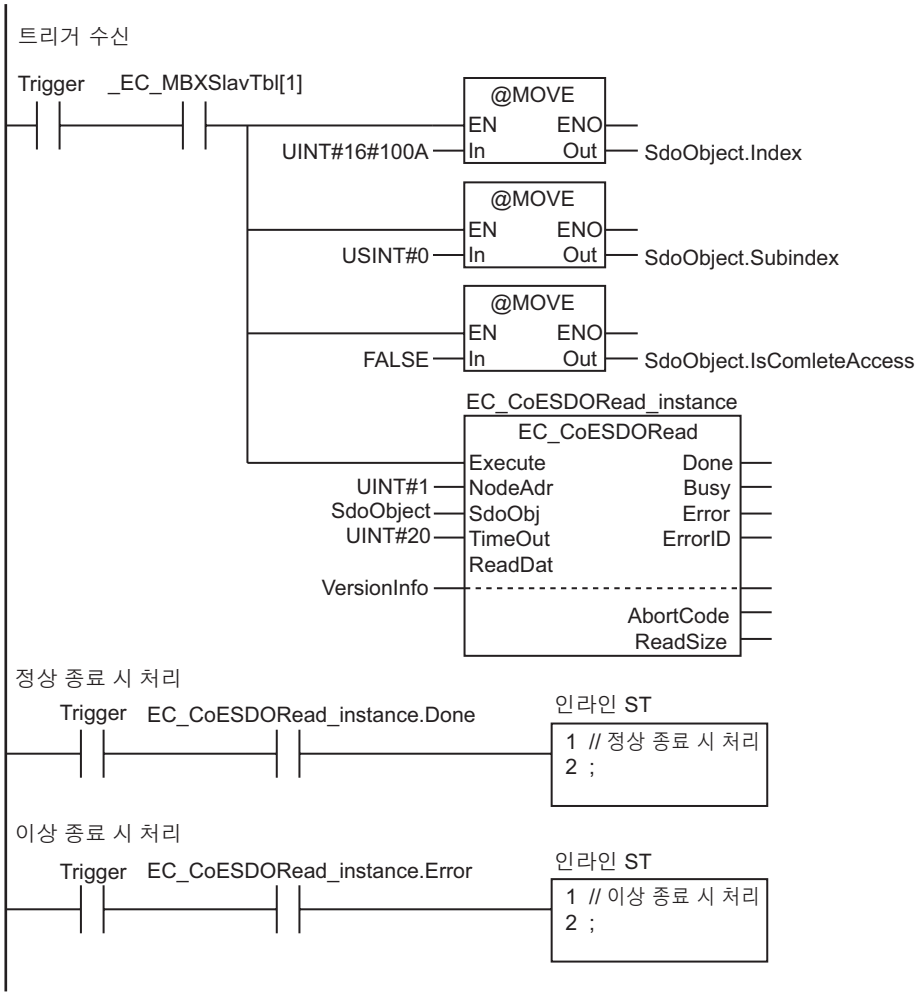
읽은 값은 STRING 형식 변수 VersionInfo에 저장합니다.



LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	SdoObject	_sSDO_ACCESS	(Index:=0, Subindex:=0,IsCompleteAccess:=FALSE)	SDO 파라메터
	VersionInfo	STRING[256]	''	읽기 데이터
	EC_CoESDORead_instance	EC_CoESDORead		

외부 변수	명칭	데이터 형식	정수	코멘트
	_EC_MBXSlavTbl	ARRAY[1..192] OF BOOL	▽	메시지 통신 가능 슬레이브 테이블



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	SdoObject	_sSDO_ACCESS	(Index:=0, Subindex:=0,IsCompleteAccess:=FALSE)	SDO 파라메터
	DoSdoRead	BOOL	FALSE	처리 중
	VersionInfo	STRING[256]	''	읽기 데이터
	NormalEnd	UINT	0	정상 종료
	ErrorEnd	UINT	0	이상 종료
	EC_CoESDORead_instance	EC_CoESDORead		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EC_MBXSlavTbl	ARRAY[1..192] OF BOOL	☑	메시지 통신 가능 슬레이브 테이블

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (DoSdoRead=FALSE) AND (_EC_MBXSlavTbl[1]=TRUE) ) THEN
    DoSdoRead                 :=TRUE;
    SdoObject.Index           :=UINT#16#100A;
    SdoObject.Subindex        :=USINT#0;
    SdoObject.IsCompleteAccess:=FALSE;
    EC_CoESDORead_instance(
        Execute:=FALSE,           // 인스턴스 초기화
        ReadDat:=VersionInfo);    // 더미
END_IF;

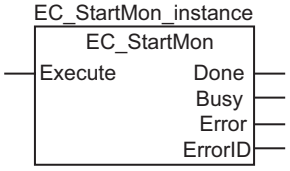
// EC_CoESDORead 명령 실행
IF (DoSdoRead=TRUE) THEN
    EC_CoESDORead_instance(
        Execute  :=TRUE,
        NodeAdr :=UINT#1,         // 노드 주소 1
        SdoObj   :=SdoObject,     // SDO 파라메터
        TimeOut :=UINT#20,        // 타임아웃 시간 2.0s
        ReadDat:=VersionInfo);    // 읽기 데이터

    IF (EC_CoESDORead_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        NormalEnd:=NormalEnd+UINT#1;
    ELSIF (EC_CoESDORead_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        ErrorEnd    :=ErrorEnd+UINT#1;
    END_IF;

END_IF;
```

EC_StartMon

EtherCAT 통신의 패킷 모니터링 기능을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_StartMon	EtherCAT 패킷 모니터링 기능 시작	FB		EC_StartMon_instance(Execute, Done, Busy, Error, ErrorID);

변수

공통의 변수만

기능

EtherCAT 통신의 패킷 모니터링 기능을 실행합니다.
패킷 모니터링 기능이란 EtherCAT 통신의 패킷을 최근의 것으로 일정 수만큼만 수집하는 기능을 말합니다. 수집 수가 일정 수를 초과하면 오래된 패킷부터 순서대로 폐기합니다.
이 명령이 실행되면 'EC_StopMon' 명령이 실행될 때까지 패킷 모니터링 기능을 계속 실행합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_PktMonStop	패킷 모니터링 정지 중	BOOL	패킷 모니터링 기능이 정지되어 있는지 나타냅니다. TRUE : 정지되어 있음 FALSE : 정지되어 있지 않음
_EC_PktSaving	패킷 데이터 파일 저장 중	BOOL	패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장 중인지를 나타냅니다. TRUE : 저장 중 FALSE : 저장 중이 아님

참고

- 이 명령 실행 중에는 수집한 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장할 수 없습니다.
- 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장하는 경우, 다음과 같이 합니다. 우선 EC_StopMon 명령으로 패킷 모니터링 기능을 정지시키고, 다음으로 EC_SaveMon 명령으로 저장을 실행합니다.
- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 CPU 유니트 NJ501-□□□□에서만 사용할 수 있습니다. NJ301-□□□□에서는 사용할 수 없습니다.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

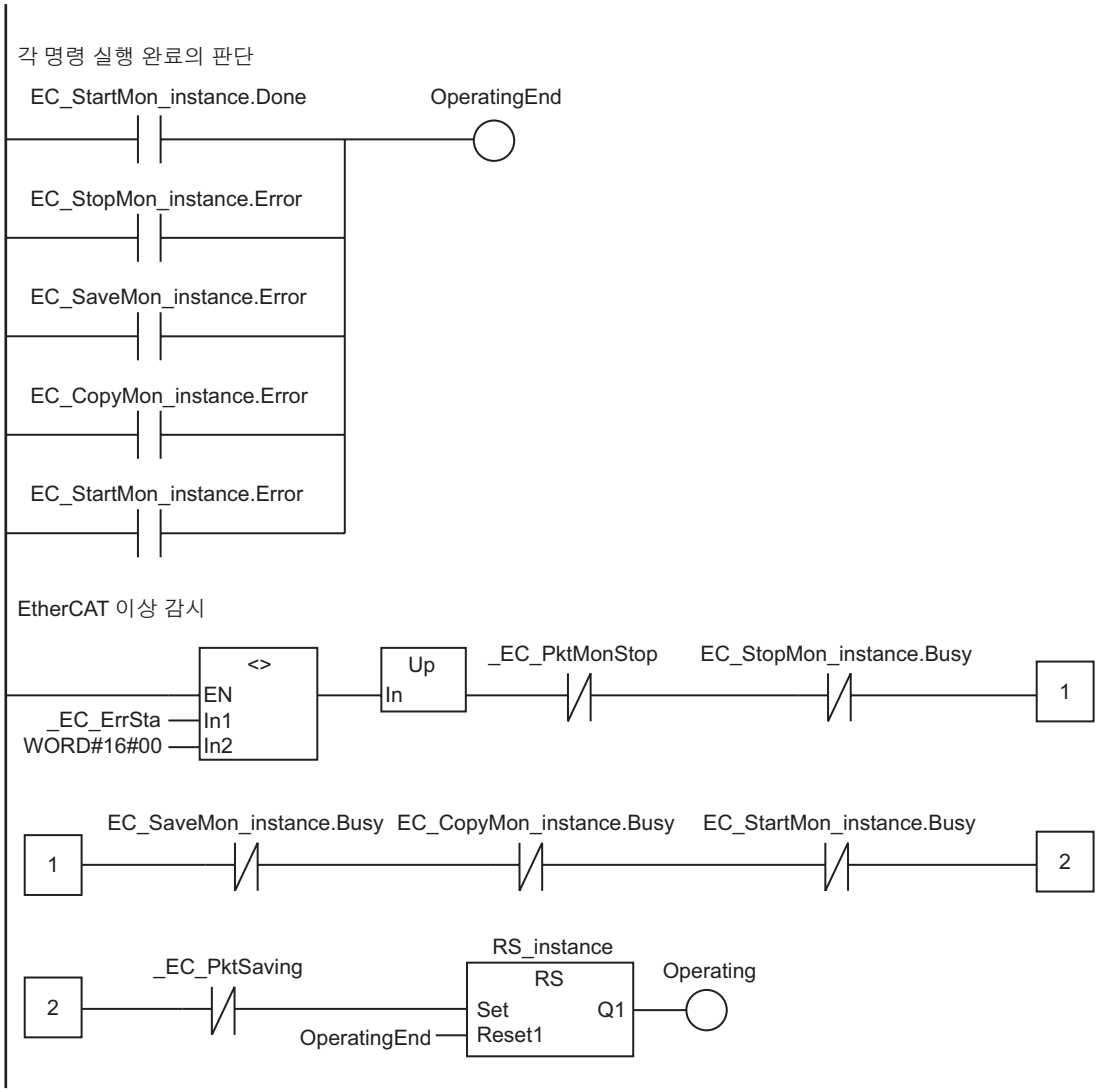
EtherCAT의 슬레이브 이상이 발생한 경우, EtherCAT 통신 패킷을 SD 메모리 카드에 저장합니다. 파일명은 'PacketFile'로 합니다.
처리 순서는 다음과 같습니다.

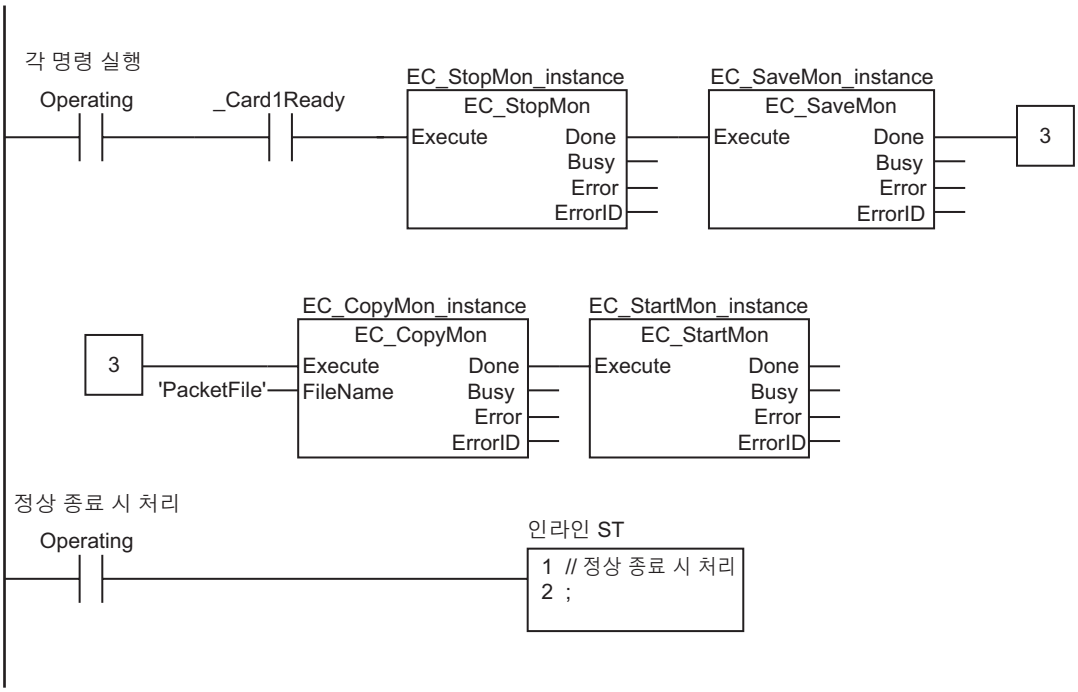
1. **내장 EtherCAT 이상의 시스템 정의 변수 '_EC_ErrSta'를 감시하다가 이상이 발생하면 처리를 시작합니다.**
2. **EC_StopMon 명령을 사용하여 EtherCAT 통신의 패킷 모니터링 기능을 정지시킵니다.**
3. **EC_SaveMon 명령을 사용하여 EtherCAT 통신의 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장합니다.**
4. **그 파일을 EC_CopyMon 명령을 사용하여 SD 메모리 카드에 복사합니다.**
5. **EC_StartMon 명령을 사용하여 다시 EtherCAT 통신의 패킷 모니터링 기능을 실행합니다.**

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Operating	BOOL	FALSE	실행 조건
	RS_instance	RS		
	EC_StopMon_instance	EC_StopMon		
	EC_SaveMon_instance	EC_SaveMon		
	EC_CopyMon_instance	EC_CopyMon		
	EC_StartMon_instance	EC_StartMon		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EC_ErrSta	WORD	☑	내장 EtherCAT 이상
	_EC_PktMonStop	BOOL	☑	패킷 모니터링 정지 중
	_EC_PktSaving	BOOL	☑	패킷 데이터 파일 저장 중
	_Card1Ready	BOOL	☑	SD 메모리 카드 사용 가능 플래그




각 명령 실행
Operating
_Card1Ready
EC_StopMon_instance
EC_StopMon
Execute
Done
Busy
Error
ErrorID
EC_SaveMon_instance
EC_SaveMon
Execute
Done
Busy
Error
ErrorID
3
3
'PacketFile'
EC_CopyMon_instance
EC_CopyMon
Execute
FileName
Done
Busy
Error
ErrorID
EC_StartMon_instance
EC_StartMon
Execute
Done
Busy
Error
ErrorID
정상 종료 시 처리
Operating
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	EC_Err	BOOL	FALSE	EtherCAT 마스터 기능 모듈의 컨트롤러 이상 발생
	EC_Err_Trigger	BOOL	FALSE	EC_Err의 기동
	DoEC_PktSave	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	R_TRIG_instance	R_TRIG		
	EC_StopMon_instance	EC_StopMon		
	EC_SaveMon_instance	EC_SaveMon		
	EC_CopyMon_instance	EC_CopyMon		
	EC_StartMon_instance	EC_StartMon		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EC_ErrSta	WORD	☑	내장 EtherCAT 이상
	_EC_PktMonStop	BOOL	☑	패킷 모니터링 정지 중
	_EC_PktSaving	BOOL	☑	패킷 데이터 파일 저장 중
	_Card1Ready	BOOL	☑	SD 메모리 카드 사용 가능 플래그

```
// _EC_ErrSta의 기동으로 시퀀스 시작
EC_Err:=(_EC_ErrSta <> WORD#16#00);
R_TRIG_instance(Clk:=EC_Err, Q=>EC_Err_Trigger);

IF ( (EC_Err_Trigger=TRUE) AND (DoEC_PktSave=FALSE) AND (_EC_PktMonStop=FALSE)
     AND (_EC_PktSaving=FALSE) AND (_Card1Ready=TRUE) ) THEN
    DoEC_PktSave:=TRUE;
    Stage              :=INT#1;
    EC_StopMon_instance(Execute:=FALSE);     // 인스턴스 초기화
    EC_SaveMon_instance(Execute:=FALSE);
    EC_CopyMon_instance(Execute:=FALSE);
    EC_StartMon_instance(Execute:=FALSE);
END_IF;

 // 각 명령 실행
IF (DoEC_PktSave=TRUE) THEN
    CASE Stage OF
    1 :                                          // EtherCAT 패킷 모니터링 기능 정지
        EC_StopMon_instance(
            Execute    :=TRUE);

        IF (EC_StopMon_instance.Done=TRUE) THEN
            Stage:=INT#2;                        // 정상 종료
        ELSIF (EC_StopMon_instance.Error=TRUE) THEN
            Stage:=INT#10;                       // 이상 종료
        END_IF;

    2 :                  // EtherCAT 패킷 데이터를 내부 파일에 저장
        EC_SaveMon_instance(
            Execute :=TRUE);

        IF (EC_SaveMon_instance.Done=TRUE) THEN
            Stage:=INT#3;                        // 정상 종료
        ELSIF (EC_SaveMon_instance.Error=TRUE) THEN
            Stage:=INT#20;                       // 이상 종료
        END_IF;
```

```
    3 :                                                 // EtherCAT 패킷 데이터 파일을 SD 메모리 카드에 복사
        EC_CopyMon_instance(
            Execute  :=TRUE,
            FileName:='PacketFile');

        IF (EC_CopyMon_instance.Done=TRUE) THEN
            Stage:=INT#4;                               // 정상 종료
        ELSIF (EC_CopyMon_instance.Error=TRUE) THEN
            Stage:=INT#30;                              // 이상 종료
        END_IF;

    4 :                                                 // EtherCAT 패킷 모니터링 기능 재개
        EC_StartMon_instance(
            Execute  :=TRUE);

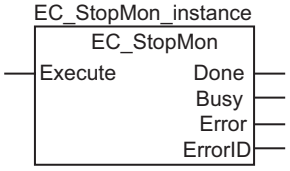
        IF (EC_StartMon_instance.Done=TRUE) THEN
            Stage:=INT#0;                               // 정상 종료
        ELSIF (EC_StartMon_instance.Error=TRUE) THEN
            Stage:=INT#40;                              // 이상 종료
        END_IF;

    0 :                                                 // 정상 종료 시 처리
        DoEC_PktSave:=FALSE;

    ELSE                                                // 이상 종료 시 처리
        DoEC_PktSave:=FALSE;
    END_CASE;
END_IF;
```

EC_StopMon

EtherCAT 통신의 패킷 모니터링 기능을 정지합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_StopMon	EtherCAT 패킷 모니터링 기능 정지	FB		EC_StopMon_instance(Execute, Done, Busy, Error, ErrorID);

변수

공통의 변수만

기능

EtherCAT 통신의 패킷 모니터링 기능을 정지합니다.
패킷 모니터링 기능이란 EtherCAT 통신의 패킷을 최근의 것으로 일정 수만큼만 수집하는 기능을 말합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_PktMonStop	패킷 모니터링 정지 중	BOOL	패킷 모니터링 기능이 정지되어 있는지 나타냅니다. TRUE : 정지되어 있음 FALSE : 정지되어 있지 않음
_EC_PktSaving	패킷 데이터 파일 저장 중	BOOL	패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장 중인지를 나타냅니다. TRUE : 저장 중 FALSE : 저장 중이 아님

참고

- 수집한 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장하는 경우, 다음과 같이 합니다. 우선, 이 명령으로 패킷 모니터링 기능을 정지시키고 EC_SaveMon 명령으로 저장을 실행합니다.
- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 CPU 유니트 NJ501-□□□□에서만 사용할 수 있습니다. NJ301-□□□□에서는 사용할 수 없습니다.

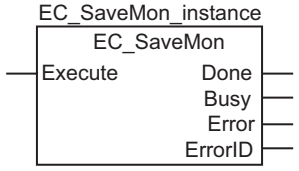
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 패킷 모니터링 기능이 이미 정지 상태인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

'EC_StartMon 명령(2-813페이지)'의 샘플 프로그램을 참조하십시오.

EC_SaveMon

EtherCAT 통신의 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_SaveMon	EtherCAT 패킷 저장	FB		EC_SaveMon_instance(Execute, Done, Busy, Error, ErrorID);

변수

공통의 변수만

기능

EtherCAT 통신의 패킷 모니터링 기능을 사용하여 수집한 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장합니다.
패킷 모니터링 기능이란 EtherCAT 통신의 패킷을 최근의 것으로 일정 수만큼만 수집하는 기능을 말합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_PktMonStop	패킷 모니터링 정지 중	BOOL	패킷 모니터링 기능이 정지되어 있는지 나타냅니다. TRUE : 정지되어 있음 FALSE : 정지되어 있지 않음
_EC_PktSaving	패킷 데이터 파일 저장 중	BOOL	패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장 중인지를 나타냅니다. TRUE : 저장 중 FALSE : 저장 중이 아님

참고

- 이 명령 실행 중에는 패킷 모니터링 기능을 실행할 수 없습니다.
- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 CPU 유니트 NJ501-□□□□에서만 사용할 수 있습니다. NJ301-□□□□에서는 사용할 수 없습니다.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.

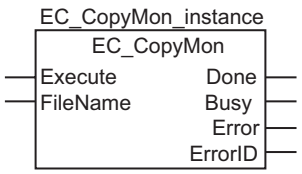
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- 이 명령은 패킷 모니터링 기능 실행 중에는 사용할 수 없습니다. 미리 EC_StopMon 명령을 사용하여 패킷 모니터링 기능을 정지시키십시오.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 패킷 모니터링 기능이 실행 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

'EC_StartMon 명령(2-813페이지)'의 샘플 프로그램을 참조하십시오.

EC_CopyMon

EtherCAT 통신의 CPU 유니트의 메인 메모리상의 내부 파일에 있는 패킷 데이터를 SD 메모리 카드로 전송합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
EC_CopyMon	EtherCAT 패킷 전송	FB		EC_CopyMon_instance(Execute, FileName, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	파일명	입력	SD 메모리 카드 내의 파일명	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○

기능

EtherCAT 통신의 CPU 유니트의 메인 메모리상의 내부 파일에 있는 패킷 데이터를 SD 메모리 카드로 전송합니다.
SD 메모리 카드 내의 파일명을 'FileName'으로 지정합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_PktSaving	패킷 데이터 파일 저장 중	BOOL	패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장 중인지를 나타냅니다. TRUE : 저장 중 FALSE : 저장 중이 아님

참고

EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

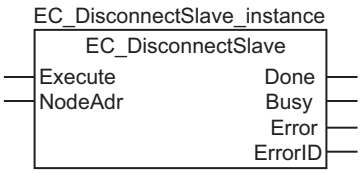
- 이 명령은 CPU 유니트 NJ501-□□□□에서만 사용할 수 있습니다. NJ301-□□□□에서는 사용할 수 없습니다.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- 이 명령은 패킷 모니터링 저장 실행 중에는 사용할 수 없습니다.
- 이 명령을 사용하기 전에 미리 EC_SaveMon 명령을 사용하여 패킷 데이터를 CPU 유니트의 메인 메모리상의 내부 파일에 저장해두십시오.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 패킷 데이터 파일을 저장 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

'EC_StartMon 명령(2-813페이지)'의 샘플 프로그램을 참조하십시오.

EC_DisconnectSlave

EtherCAT 네트워크상의 지정된 슬레이브를 네트워크로부터 이탈시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_ DisconnectSlave	EtherCAT 슬레이브 이탈	FB		EC_DisconnectSlave_instance(Execute, NodeAdr, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
NodeAdr	슬레이브 노드 주소	입력	이탈시킬 슬레이브의 노드 주소	1~192	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
NodeAdr							○													

기능

슬레이브 노드 주소 'NodeAdr'로 지정한 슬레이브를 EtherCAT 네트워크로부터 이탈시킵니다.
네트워크로부터의 이탈이란 슬레이브가 네트워크에 존재하지만 동작하지 않는 상태로 만드는 것을 의미합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_EntrySlavTbl[i] i는 노드 주소	네트워크 가입 슬레이브 테이블	BOOL[]	슬레이브가 네트워크에 가입(존재)했는지를 나타냅니다. TRUE : 가입(존재)함 FALSE : 가입(존재)하지 않음
_EC_DisconnSlavTbl[i] i는 노드 주소	이탈 명령 중 슬레이브 테이블	BOOL[]	슬레이브가 이탈 명령 중인지를 나타냅니다. TRUE : 이탈 명령 중 FALSE : 이탈 명령 중이 아님
_EC_DisableSlavTbl[i] i는 노드 주소	무효 설정 슬레이브 테이블	BOOL[]	슬레이브가 무효 설정되어 있는지를 나타냅니다. TRUE : 무효 설정됨 FALSE : 무효 설정되지 않음

참고

EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- 이탈시킨 슬레이브의 후속으로 데이지 체인 연결(Out 포트에 연결)된 슬레이브가 존재하는 경우, 그 슬레이브도 EtherCAT 네트워크에서 이탈합니다.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령을 실행 중에는 이 명령을 실행할 수 없습니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'NodeAdr'로 지정한 슬레이브가 EtherCAT 네트워크에 가입(존재)하지 않은 경우. 즉, 네트워크 가입 슬레이브 테이블 _EC_EntrySlavTbl[i]의 값이 FALSE인 경우
  - 'NodeAdr'로 지정한 슬레이브가 무효 설정된 경우
  - EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 중 하나가 실행 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

슬레이브 1을 EtherCAT 네트워크로부터 이탈시키거나 재개입시킵니다.
Trigger1이 입력되면 EC_DisconnectSlave 명령을 사용하여 이탈시키고 Trigger2가 입력되면 EC_ConnectSlave 명령을 사용하여 재가입시킵니다.

명령의 배타적 제어

EC_DisconnectSlave 명령과 EC_ConnectSlave 명령은 동시에 실행할 수 없습니다. 또한 어느 명령이든 복수의 태스크 주기에 걸쳐 실행됩니다. 따라서 먼저 실행된 쪽 명령의 완료를 확인한 후 다음 명령을 실행합니다. 그 때문에 'ExclusiveFlg(명령 배타 플래그)' 변수를 사용하며 'ExclusiveFlg'의 값이 TRUE이면 어떤 명령이 실행 중임을 의미합니다. 'ExclusiveFlg'의 값이 TRUE인 동안은 다음 명령이 실행되지 않습니다.

EC_DisconnectSlave 명령과 EC_ConnectSlave 명령은 다른 태스크에 있는 이들 명령과도 동시에 실행할 수 없습니다. 따라서 이 샘플 프로그램에서는 'ExclusiveFlg'를 글로벌 변수로 처리하고 있습니다. 이렇게 하면 이 샘플 프로그램과 다른 태스크 내의 명령을 배타적으로 제어할 수 있습니다. 단, 그 경우는

다른 태스크 내에서도 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행해야 합니다.
또한 EC_DisconnectSlave 명령과 EC_ConnectSlave 명령은 EC_ChangeEnableSetting 명령과 동시에 실행할 수 없습니다. 'EC_ChangeEnableSetting 명령(2-835페이지)'의 샘플 프로그램은 이 샘플 프로그램과 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행하는 예입니다.

글로벌 변수의 정의

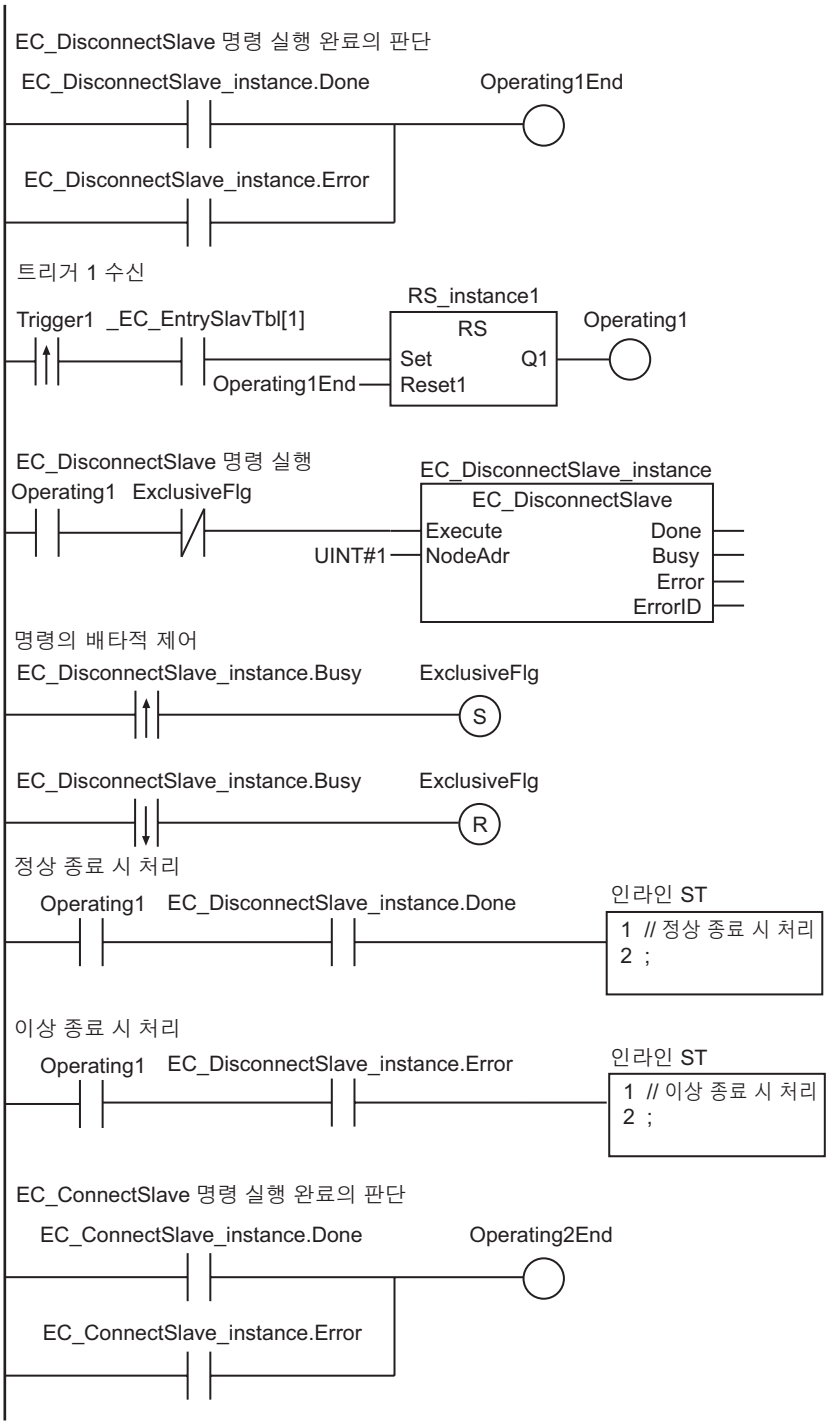
글로벌 변수

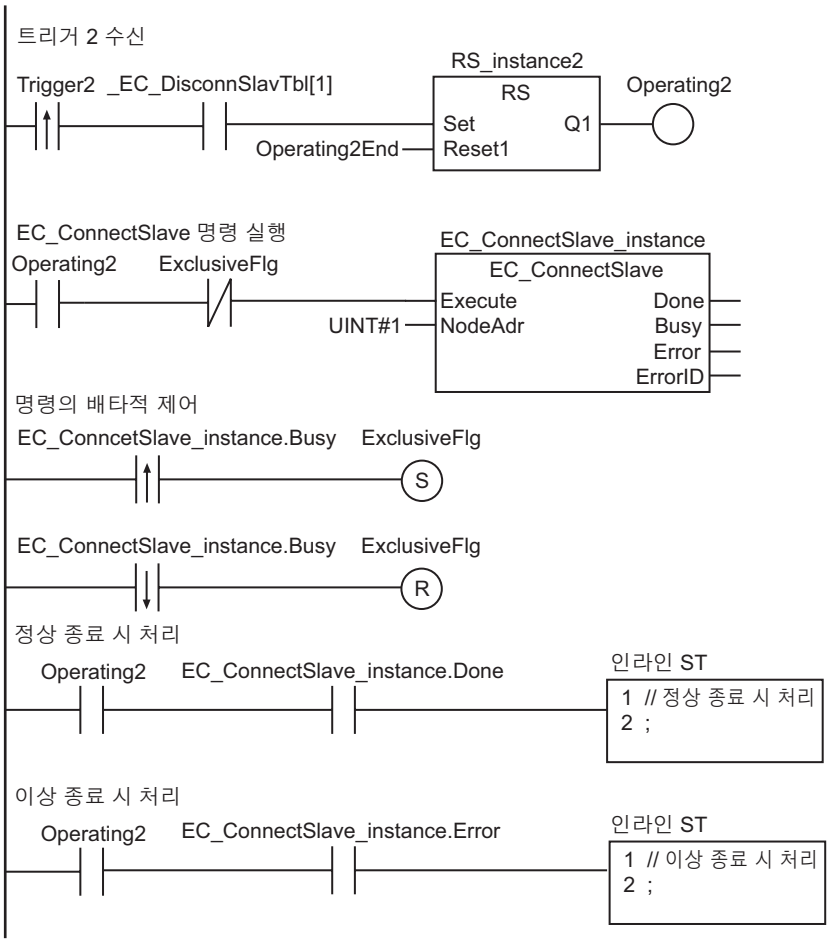
이름	데이터 형식	초기값	코멘트
ExclusiveFlg	BOOL	FALSE	명령 배타 플래그

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Operating1End	BOOL	FALSE	처리 종료 1
	Trigger1	BOOL	FALSE	실행 조건 1
	Operating1	BOOL	FALSE	처리 중 1
	RS_instance1	RS		
	EC_DisconnectSlave_instance	EC_DisconnectSlave		
	Operating2End	BOOL	FALSE	처리 종료 2
	Trigger2	BOOL	FALSE	실행 조건 2
	Operating2	BOOL	FALSE	처리 중 2
	RS_instance2	RS		
	EC_ConnectSlave_instance	EC_ConnectSlave		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EC_EntrySlavTbl	ARRAY[1..192] OF BOOL	☑	네트워크 가입 슬레이브 테이블
	_EC_DisconnSlavTbl	ARRAY[1..192] OF BOOL	☑	이탈 명령 중 슬레이브 테이블
	ExclusiveFlg	BOOL	☐	명령 배타 플래그


EC_DisconnectSlave 명령 실행 완료의 판단
EC_DisconnectSlave_instance.Done
Operating1End
EC_DisconnectSlave_instance.Error
트리거 1 수신
RS_instance1
RS
Trigger1
_EC_EntrySlavTbl[1]
Set
Q1
Operating1
Operating1End
Reset1
EC_DisconnectSlave 명령 실행
EC_DisconnectSlave_instance
EC_DisconnectSlave
Operating1
ExclusiveFlg
Execute
Done
UINT#1
NodeAdr
Busy
Error
ErrorID
명령의 배타적 제어
EC_DisconnectSlave_instance.Busy
ExclusiveFlg
S
EC_DisconnectSlave_instance.Busy
ExclusiveFlg
R
정상 종료 시 처리
Operating1
EC_DisconnectSlave_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
Operating1
EC_DisconnectSlave_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
EC_ConnectSlave 명령 실행 완료의 판단
EC_ConnectSlave_instance.Done
Operating2End
EC_ConnectSlave_instance.Error


트리거 2 수신
RS_instance2
RS
Trigger2 _EC_DisconnSlavTbl[1]
Set
Q1
Operating2
Operating2End
Reset1
EC_ConnectSlave 명령 실행
EC_ConnectSlave_instance
EC_ConnectSlave
Operating2
ExclusiveFlg
Execute
Done
UINT#1
NodeAdr
Busy
Error
ErrorID
명령의 배타적 제어
EC_ConncetSlave_instance.Busy
ExclusiveFlg
S
EC_ConnectSlave_instance.Busy
ExclusiveFlg
R
정상 종료 시 처리
Operating2
EC_ConnectSlave_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
Operating2
EC_ConnectSlave_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger1	BOOL	FALSE	실행 조건 1
	LastTrigger1	BOOL	FALSE	이전 태스크 주기의 Trigger 1의 값
	Operating1Start	BOOL	FALSE	처리 시작 1
	Operating1	BOOL	FALSE	처리 중 1
	DisconnectSet	BOOL	FALSE	EC_DisconnectSlave 명령 실행 중
	DisconnectReset	BOOL	FALSE	EC_DisconnectSlave 명령 비실행 중
	EC_DisconnectSlave_instance	EC_DisconnectSlave		
	Trigger2	BOOL	FALSE	실행 조건 2
	LastTrigger2	BOOL	FALSE	이전 태스크 주기의 Trigger 2의 값
	Operating2Start	BOOL	FALSE	처리 시작 2
	Operating2	BOOL	FALSE	처리 중 2
	ConnectSet	BOOL	FALSE	EC_ConnectSlave 명령 실행 중
	ConnectReset	BOOL	FALSE	EC_ConnectSlave 명령 비실행 중
	EC_ConnectSlave_instance	EC_ConnectSlave		
	R_TRIG_instance1	R_TRIG		
	F_TRIG_instance1	F_TRIG		
	RS_instance1	RS		
	R_TRIG_instance2	R_TRIG		
	F_TRIG_instance2	F_TRIG		
	RS_instance2	RS		

내부 변수	명칭	데이터 형식	상수	코멘트
	_EC_EntrySlavTbl	ARRAY[1..192] OF BOOL	✉	네트워크 가입 슬레이브 테이블
	_EC_DisconnSlavTbl	ARRAY[1..192] OF BOOL	✉	이탈 명령 중 슬레이브 테이블
	ExclusiveFlg	BOOL	⊓	명령 배타 플래그

```
// Trigger 1의 기동 검출
IF ( (Trigger1=TRUE) AND (LastTrigger1=FALSE) AND (_EC_EntrySlavTbl[1]=TRUE) ) THEN
    Operating1Start:=TRUE;
    Operating1      :=TRUE;
END_IF;
LastTrigger1:=Trigger1;

// EC_DisconnectSlave 명령 초기화
IF (Operating1Start=TRUE) THEN
    EC_DisconnectSlave_instance(Execute:=FALSE);
    Operating1Start:=FALSE;
END_IF;

// EC_DisconnectSlave 명령 실행
IF (Operating1=TRUE) THEN
    EC_DisconnectSlave_instance(
        Execute :=TRUE,
        NodeAdr:=UINT#1);

    // 명령의 배타적 제어
    R_TRIG_instance1(EC_DisconnectSlave_instance.Busy, DisconnectSet);
    F_TRIG_instance1(EC_DisconnectSlave_instance.Busy, DisconnectReset);
    RS_instance1(DisconnectSet, DisconnectReset, ExclusiveFlg);

    IF (EC_DisconnectSlave_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating1:=FALSE;
    END_IF;

    IF (EC_DisconnectSlave_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating1:=FALSE;
    END_IF;
END_IF;

// Trigger 2의 기동 검출
IF ( (Trigger2=TRUE) AND (LastTrigger2=FALSE) AND (_EC_DisconnSlavTbl[1]=TRUE) ) THEN
    Operating2Start:=TRUE;
    Operating2      :=TRUE;
END_IF;
LastTrigger2:=Trigger2;

// EC_ConnectSlave 명령 초기화
IF (Operating2Start=TRUE) THEN
    EC_ConnectSlave_instance(Execute:=FALSE);
    Operating2Start:=FALSE;
END_IF;

// EC_ConnectSlave 명령 실행
IF (Operating2=TRUE) THEN
    EC_ConnectSlave_instance(
        Execute :=TRUE,
        NodeAdr:=UINT#1);

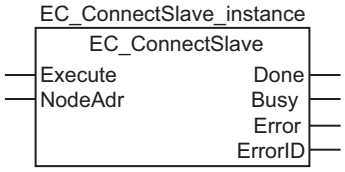
   // 명령의 배타적 제어
   R_TRIG_instance2(EC_ConnectSlave_instance.Busy, ConnectSet);
   F_TRIG_instance2(EC_ConnectSlave_instance.Busy, ConnectReset);
   RS_instance2(ConnectSet, ConnectReset, ExclusiveFlg);
```

```
    IF (EC_ConnectSlave_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating2:=FALSE;
    END_IF;

    IF (EC_ConnectSlave_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating2:=FALSE;
    END_IF;
END_IF;
```

EC_ConnectSlave

EtherCAT 네트워크상의 지정된 슬레이브를 네트워크에 재가입시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_Connect Slave	EtherCAT 슬레이브 재가입	FB		EC_ConnectSlave_instance(Execute, NodeAdr, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
NodeAdr	슬레이브 노드 주소	입력	등록할 슬레이브의 노드 주소	1~192	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
NodeAdr							○													

기능

슬레이브 노드 주소 'NodeAdr'로 지정한 슬레이브를 EtherCAT 네트워크에 재가입시킵니다.
네트워크 재가입이란 슬레이브가 네트워크에 존재하면서 동작하는 상태로 만드는 것을 의미합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_EntrySlavTbl[i] i는 노드 주소	네트워크 가입 슬레이브 테이블	BOOL[]	슬레이브가 네트워크에 가입(존재)했는지를 나타냅니다. TRUE : 가입(존재)함 FALSE : 가입(존재)하지 않음
_EC_DisconnSlavTbl[i] i는 노드 주소	이탈 명령 중 슬레이브 테이블	BOOL[]	슬레이브가 이탈 명령 중인지를 나타냅니다. TRUE : 이탈 명령 중 FALSE : 이탈 명령 중이 아님

참고

EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.

사용상의 주의 사항

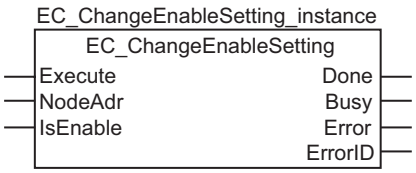
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용 가능합니다.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령을 실행 중에는 이 명령을 실행할 수 없습니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'NodeAdr'로 지정한 슬레이브가 EtherCAT 네트워크에 가입(존재)하지 않은 경우. 즉, 네트워크 가입 슬레이브 테이블 _EC_EntrySlavTbl[i]의 값이 FALSE인 경우
  - EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 중 하나가 실행 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

샘플 프로그램

'EC_DisconnectSlave 명령(2-825페이지)'의 샘플 프로그램을 참조하십시오.

EC_ChangeEnableSetting

EtherCAT 슬레이브의 유효/무효를 전환합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
EC_Change Enable Setting	EtherCAT 슬레이브의 유효/무효 전환	FB		EC_ChangeEnableSetting_instance(Execute, NodeAdr, IsEnable, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
NodeAdr	슬레이브 노드 주소	입력	유효/무효 전환을 수행할 EtherCAT 슬레이브의 노드 주소	1~192	–	1
IsEnable	유효/무효 전환 지시		지정한 EtherCAT 슬레이브를 유효로 할지, 무효로 할지에 대한 지시 TRUE : 유효 FALSE : 무효	데이터 형식에 따름		TRUE

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
NodeAdr							○													
IsEnable	○																			

기능

슬레이브 노드 주소 'NodeAdr'로 지정한 EtherCAT 슬레이브에 대해 유효/무효 전환을 수행합니다. 유효/무효 전환 지시 'IsEnable'의 값이 TRUE이면 유효가 되고, FALSE이면 무효가 됩니다.
이 명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 유효/무효 전환이 완료됩니다.

이 명령은 지정된 EtherCAT 슬레이브의 상태가 유효인지 또는 무효인지, 가입 중인지 또는 이탈 중인지, EtherCAT 네트워크 내에 존재하는지 또는 존재하지 않는지에 따라 정상 종료되거나 이상 종료됩니다. 이 명령 실행 전의 EtherCAT 슬레이브의 각 상태에 대한 이 명령 실행 후의 상태는 다음과 같습니다.

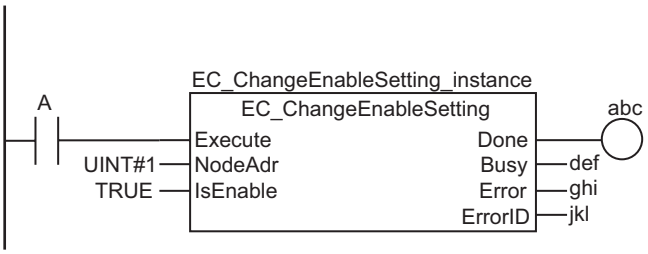
이 명령 실행 전의 상태			'IsEnable'의 값	이 명령 실행 후의 상태	
유효/무효	가입 중/이탈 중	존재*1		정상 종료/이상 종료	유효/무효
유효	가입 중	○	TRUE	정상 종료	유효
			FALSE		무효
유효	이탈 중	○	TRUE	이상 종료	유효(실행 전 그대로)
			FALSE		
		×	TRUE		
			FALSE		
무효	–*2	○	TRUE	정상 종료	유효
			FALSE		무효
무효	–*2	×	TRUE	이상 종료	무효(실행 전 그대로)
			FALSE		

*1 지정된 EtherCAT 슬레이브가 EtherCAT 네트워크 내에 물리적으로 설치되어 있는지, 분리되어 있는지를 나타냅니다. ○: 설치되어 있음 ×: 분리되어 있음

*2 무효로 설정되어 있는 EtherCAT 슬레이브에는 가입 중/이탈 중의 구별이 없습니다.

노드 주소 1의 EtherCAT 슬레이브를 유효로 하는 경우의 예는 다음과 같습니다. 'NodeAdr'에는 UINT#1을, 'IsEnable'에는 TRUE를 지정합니다.

LD



ST

```
EC_ChangeEnableSetting_instance(A, UINT#1, TRUE,
                                abc, def, ghi, jkl);
```

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_EntrySlavTbl[i] i는 노드 주소	네트워크 가입 슬레이브 테이블	BOOL[]	슬레이브가 네트워크에 가입(존재)되어 있는지 여부를 나타냅니다. TRUE : 가입(존재)되어 있음 FALSE : 가입(존재)하지 않음
_EC_DisconnSlavTbl[i] i는 노드 주소	이탈 명령 중 슬레이브 테이블	BOOL[]	슬레이브가 이탈 명령 중인지 여부를 나타냅니다. TRUE : 이탈 명령 중 FALSE : 이탈 명령 중이 아님
_EC_DisableSlavTbl[i] i는 노드 주소	무효 설정 슬레이브 테이블	BOOL[]	슬레이브가 무효인지 여부를 나타냅니다. TRUE : 무효 FALSE : 유효 또는 네트워크에 가입(존재)하지 않음

참고

- EtherCAT 통신에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서 (SBCD-358)"를 참조하십시오.
- EtherCAT 슬레이브를 EtherCAT 네트워크에 재가입시키려면 'EC_ConnectSlave 명령(2-833페이지)' 명령을 사용하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈의 EtherCAT 포트에만 사용할 수 있습니다.
- EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령을 실행 중에는 이 명령을 실행할 수 없습니다.
- 이 명령의 실행 결과는 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 이 명령 실행 후에 CPU 유니트의 전원을 껐다 켜거나 사용자 프로그램의 다운로드를 실행하면 EtherCAT 슬레이브의 유효/무효 설정은 Sysmac Studio에서 설정한 내용으로 돌아갑니다.
- EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령의 경우 동시에 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상이 발생하며 'Error'는 TRUE가 됩니다.
  - 'NodeAdr'에서 지정한 슬레이브가 EtherCAT 네트워크에 가입(존재)되어 있지 않은 경우. 즉, 네트워크 가입 슬레이브 테이블 _EC_EntrySlavTbl[i]의 값이 FALSE인 경우
  - 'NodeAdr'의 값이 유효 범위 밖인 경우
  - EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령, ResetECError 명령, RestartNXUnit 명령, NX_ChangeWriteMode 명령 중 하나가 실행 중인 경우
  - EC_CoESDOWrite 명령, EC_CoESDORead 명령, EC_StartMon 명령, EC_StopMon 명령, EC_SaveMon 명령, EC_CopyMon 명령, EC_DisconnectSlave 명령, EC_ConnectSlave 명령, EC_ChangeEnableSetting 명령을 동시에 32개를 초과하여 실행한 경우

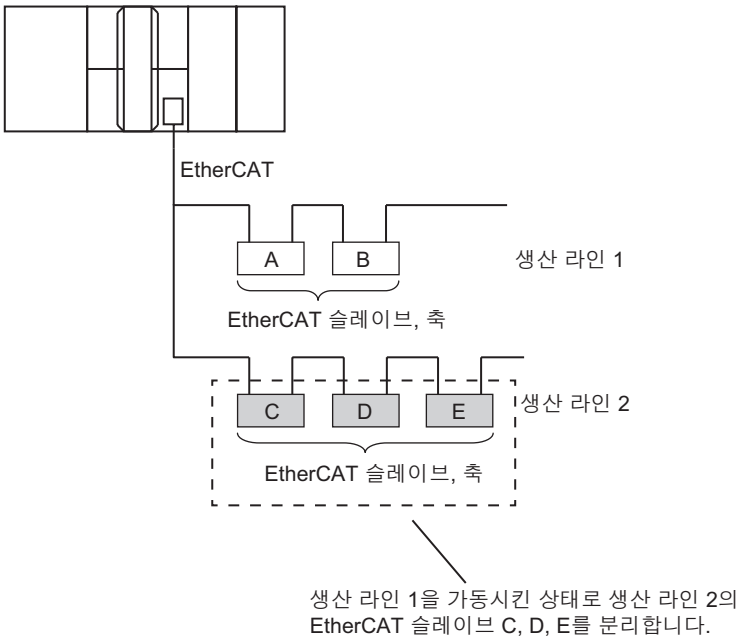
버전 관련 정보

이 명령은 CPU 유니트 Ver.1.04 이상, Sysmac Studio Ver.1.05 이상에서 사용할 수 있습니다.

샘플 프로그램

● EtherCAT 슬레이브를 EtherCAT 네트워크에서 분리하는 예

다음 그림에 나온 시스템의 생산 라인 1을 가동시킨 상태에서 생산 라인 2의 EtherCAT 슬레이브 C, D, E를 분리합니다. EtherCAT 슬레이브 C, D, E에는 모션 제어 축도 설정되어 있는 것으로 가정합니다. 따라서 EtherCAT 슬레이브를 무효화하는 동시에 축 사용을 미사용 축으로 변경하는 처리도 수행합니다.



조작 순서

이 샘플 프로그램의 조작 순서는 다음과 같습니다.

1. **조작자가 표시기의 버튼을 누르면 실행 조건이 ON이 됩니다.**
2. **컨트롤러가 EtherCAT 슬레이브 C, D, E를 무효화합니다. 또한 각각의 축 사용을 미사용 축으로 설정합니다.**
3. **3개의 EtherCAT 슬레이브의 무효화, 미사용 축으로 변경이 모두 완료되면 컨트롤러가 분리 가능 램프를 점등시킵니다.**
4. **조작자가 분리 가능 램프의 점등을 확인한 후 3개의 EtherCAT 슬레이브를 분리합니다.**

축 사용을 미사용 축으로 변경하는 명령

축 사용을 미사용 축으로 변경하려면 MC_ChangeAxisUse 명령을 사용합니다. MC_ChangeAxisUse 명령 사양에 대한 자세한 내용은 "NJ 시리즈 명령 참조 설명서 모션편(SBCE-364E 이상)"을 참조하십시오.

명령의 배타적 제어

EC_ChangeEnableSetting 명령은 동시에 여러 개 실행할 수 없습니다. 또한 EC_ChangeEnable Setting 명령은 복수의 태스크 주기에 걸쳐 실행됩니다. 따라서 먼저 실행된 EC_Change EnableSetting 명령의 완료를 확인한 후 다음 EC_ChangeEnableSetting 명령을 실행합니다. 그 때문에 'ExclusiveFlg(명령 배타 플래그)' 변수를 사용하며, 'ExclusiveFlg'의 값이 TRUE이면 EC_ChangeEnableSetting 명령이 실행되고 있음을 의미합니다. 'ExclusiveFlg'의 값이 TRUE인 동안은 다음 EC_ChangeEnableSetting 명령을 실행하지 않습니다.

EC_ChangeEnableSetting 명령은 다른 태스크 내의 EC_ChangeEnableSetting 명령과도 동시에 실행할 수 없습니다. 따라서 이 샘플 프로그램에서는 'ExclusiveFlg'를 글로벌 변수로 처리하고 있습니다. 이렇게 하면 이 샘플 프로그램과 다른 태스크 내의 EC_ChangeEnableSetting 명령을 배타적으로 제어할 수 있

습니다. 단, 그 경우는 다른 태스크 내에서도 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행해야 합니다.
또한 EC_ChangeEnableSetting 명령은 EC_DisconnectSlave 명령이나 EC_ConnectSlave 명령과도 동시에 실행할 수 없습니다. 'EC_DisconnectSlave 명령(2-825페이지)'의 샘플 프로그램은 이 샘플 프로그램과 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행하는 예입니다.

각 EtherCAT 슬레이브의 축 변수와 노드 주소

EtherCAT 슬레이브 C, D, E의 축에 할당된 축 변수와 노드 주소는 다음과 같습니다.

EtherCAT 슬레이브	축 변수	노드 주소
C	MC_Axis000	1
D	MC_Axis001	2
E	MC_Axis002	3

글로벌 변수의 정의

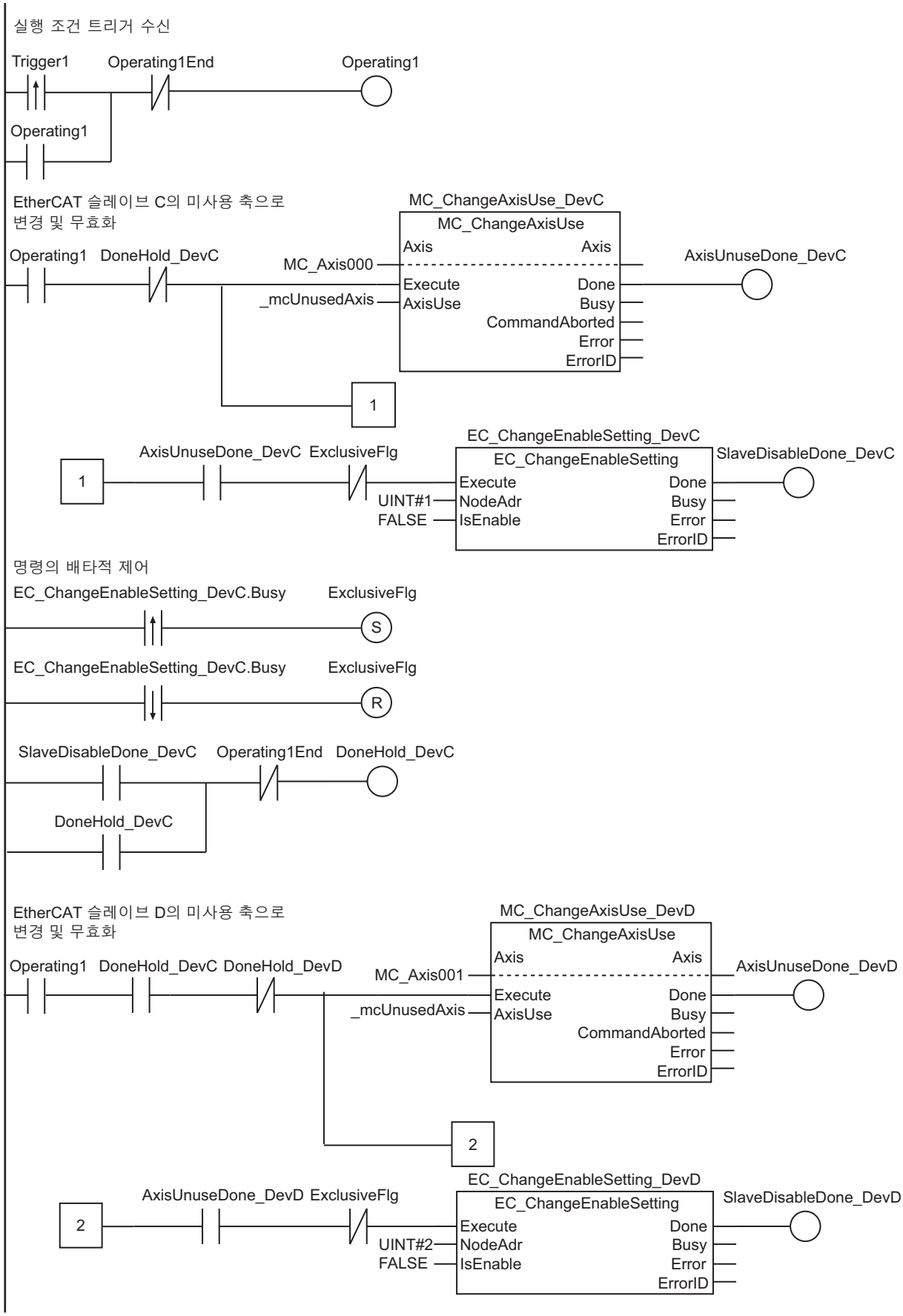
글로벌 변수

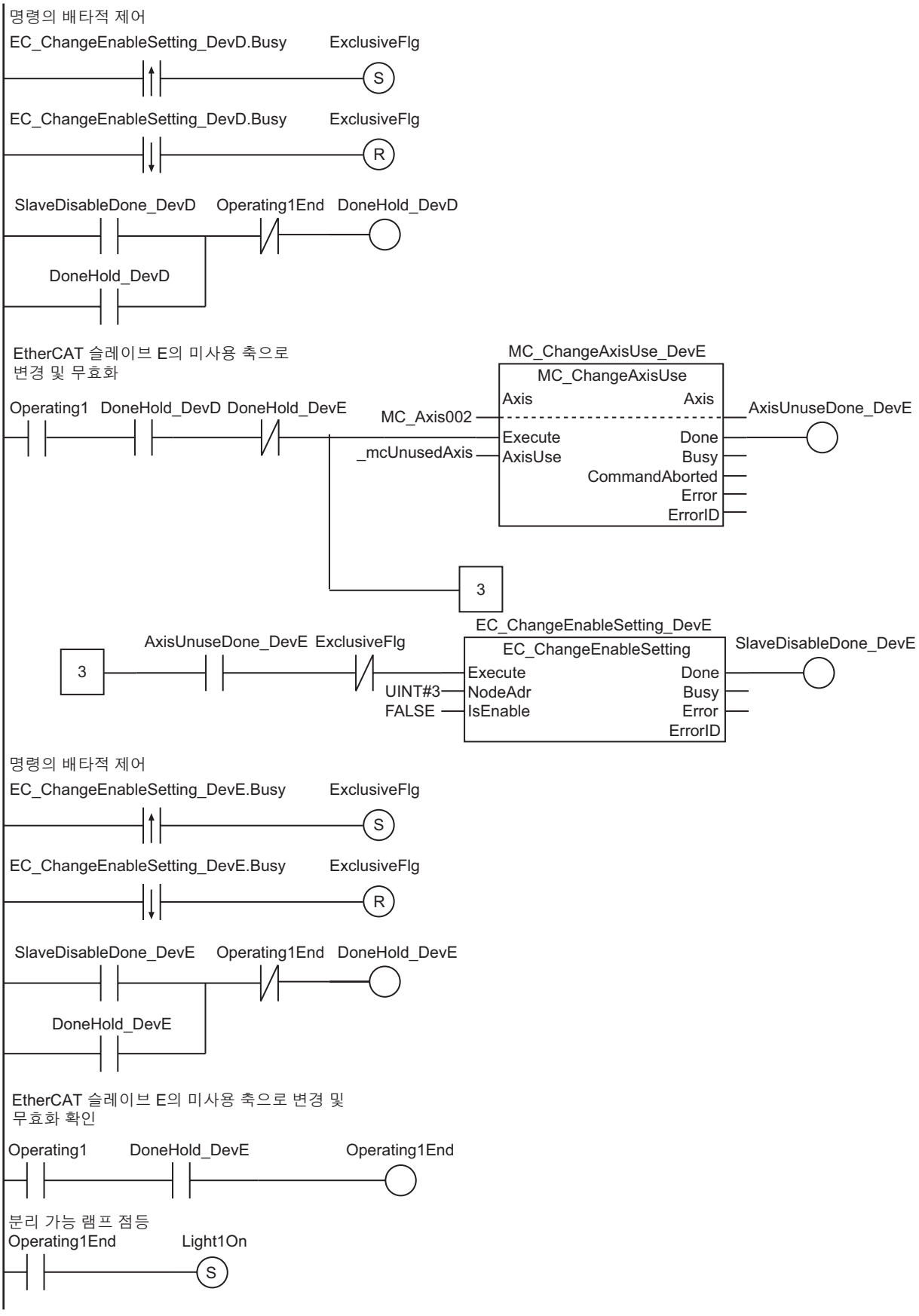
명칭	데이터 형식	초기값	할당대상	상수	코멘트
MC_Axis000	_sAXIS_REF		MC://_MC_AX[0]	☑	EtherCAT 슬레이브 C의 축 변수
MC_Axis001	_sAXIS_REF		MC://_MC_AX[1]	☑	EtherCAT 슬레이브 D의 축 변수
MC_Axis002	_sAXIS_REF		MC://_MC_AX[2]	☑	EtherCAT 슬레이브 E의 축 변수
ExclusiveFlg	BOOL	FALSE		☐	명령 배타 플래그

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Operating1End	BOOL	FALSE	처리 종료
	Trigger1	BOOL	FALSE	실행 조건
	Operating1	BOOL	FALSE	처리 중
	AxisUnuseDone_DevC	BOOL	FALSE	EtherCAT 슬레이브 C의 축 사용을 미사용 축으로 변경 완료
	SlaveDisableDone_DevC	BOOL	FALSE	EtherCAT 슬레이브 C의 무효화 완료
	DoneHold_DevC	BOOL	FALSE	EtherCAT 슬레이브 C의 처리 완료 유지
	AxisUnuseDone_DevD	BOOL	FALSE	EtherCAT 슬레이브 D의 축 사용을 미사용 축으로 변경 완료
	SlaveDisableDone_DevD	BOOL	FALSE	EtherCAT 슬레이브 D의 무효화 완료
	DoneHold_DevD	BOOL	FALSE	EtherCAT 슬레이브 D의 처리 완료 유지
	AxisUnuseDone_DevE	BOOL	FALSE	EtherCAT 슬레이브 E의 축 사용을 미사용 축으로 변경 완료
	SlaveDisableDone_DevE	BOOL	FALSE	EtherCAT 슬레이브 E의 무효화 완료
	DoneHold_DevE	BOOL	FALSE	EtherCAT 슬레이브 E의 처리 완료 유지
	Light1On	BOOL	FALSE	분리 가능 램프 점등
	MC_ChangeAxisUse_DevC	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevC	EC_ChangeEnableSetting		
	MC_ChangeAxisUse_DevD	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevD	EC_ChangeEnableSetting		
	MC_ChangeAxisUse_DevE	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevE	EC_ChangeEnableSetting		

외부 변수	명칭	데이터 형식	상수	코멘트
	MC_Axis000	_sAXIS_REF	☑	EtherCAT 슬레이브 C의 축 변수
	MC_Axis001	_sAXIS_REF	☑	EtherCAT 슬레이브 D의 축 변수
	MC_Axis002	_sAXIS_REF	☑	EtherCAT 슬레이브 E의 축 변수
	ExclusiveFlg	BOOL	☐	명령 배타 플래그


실행 조건 트리거 수신
Trigger1
Operating1End
Operating1
Operating1
EtherCAT 슬레이브 C의 미사용 축으로
변경 및 무효화
MC_ChangeAxisUse_DevC
MC_ChangeAxisUse
Axis
Axis
Operating1
DoneHold_DevC
MC_Axis000
Execute
Done
AxisUnuseDone_DevC
_mcUnusedAxis
AxisUse
Busy
CommandAborted
Error
ErrorID
1
EC_ChangeEnableSetting_DevC
EC_ChangeEnableSetting
AxisUnuseDone_DevC
ExclusiveFlg
SlaveDisableDone_DevC
1
Execute
Done
UINT#1
NodeAdr
Busy
FALSE
IsEnable
Error
ErrorID
명령의 배타적 제어
EC_ChangeEnableSetting_DevC.Busy
ExclusiveFlg
S
EC_ChangeEnableSetting_DevC.Busy
ExclusiveFlg
R
SlaveDisableDone_DevC
Operating1End
DoneHold_DevC
DoneHold_DevC
EtherCAT 슬레이브 D의 미사용 축으로
변경 및 무효화
MC_ChangeAxisUse_DevD
MC_ChangeAxisUse
Axis
Axis
Operating1
DoneHold_DevC
DoneHold_DevD
MC_Axis001
AxisUnuseDone_DevD
Execute
Done
_mcUnusedAxis
AxisUse
Busy
CommandAborted
Error
ErrorID
2
EC_ChangeEnableSetting_DevD
EC_ChangeEnableSetting
AxisUnuseDone_DevD
ExclusiveFlg
SlaveDisableDone_DevD
2
Execute
Done
UINT#2
NodeAdr
Busy
FALSE
IsEnable
Error
ErrorID


명령의 배타적 제어
EC_ChangeEnableSetting_DevD.Busy
ExclusiveFlg
S
EC_ChangeEnableSetting_DevD.Busy
ExclusiveFlg
R
SlaveDisableDone_DevD
Operating1End
DoneHold_DevD
DoneHold_DevD
EtherCAT 슬레이브 E의 미사용 축으로
변경 및 무효화
MC_ChangeAxisUse_DevE
MC_ChangeAxisUse
Operating1
DoneHold_DevD
DoneHold_DevE
MC_Axis002
_mcUnusedAxis
Axis
Axis
Execute
AxisUse
Done
Busy
CommandAborted
Error
ErrorID
AxisUnuseDone_DevE
3
EC_ChangeEnableSetting_DevE
EC_ChangeEnableSetting
3
AxisUnuseDone_DevE
ExclusiveFlg
UINT#3
FALSE
Execute
NodeAdr
IsEnable
Done
Busy
Error
ErrorID
SlaveDisableDone_DevE
명령의 배타적 제어
EC_ChangeEnableSetting_DevE.Busy
ExclusiveFlg
S
EC_ChangeEnableSetting_DevE.Busy
ExclusiveFlg
R
SlaveDisableDone_DevE
Operating1End
DoneHold_DevE
DoneHold_DevE
EtherCAT 슬레이브 E의 미사용 축으로 변경 및
무효화 확인
Operating1
DoneHold_DevE
Operating1End
분리 가능 램프 점등
Operating1End
Light1On
S

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Operating1End	BOOL	FALSE	처리 종료
	Trigger1	BOOL	FALSE	실행 조건
	Operating1	BOOL	FALSE	처리 중
	Operating1Set	BOOL	FALSE	처리 시작
	Light1On	BOOL	FALSE	분리 가능 램프 점등
	DoneHold_DevC	BOOL	FALSE	EtherCAT 슬레이브 C의 처리 완료 유지
	DoneHold_DevD	BOOL	FALSE	EtherCAT 슬레이브 D의 처리 완료 유지
	DoneHold_DevE	BOOL	FALSE	EtherCAT 슬레이브 E의 처리 완료 유지
	ExclusiveFlgSet	BOOL	FALSE	명령 배타 플래그 ON
	ExclusiveFlgReset	BOOL	FALSE	명령 배타 플래그 OFF
	R_TRIG_instance1	R_TRIG		
	RS_instance1	RS		
	SR_instance1	SR		
	MC_ChangeAxisUse_DevC	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevC	EC_ChangeEnableSetting		
	R_TRIG_DevC	R_TRIG		
	F_TRIG_DevC	F_TRIG		
	RS_ExFlg_DevC	RS		
	RS_DevC	RS		
	MC_ChangeAxisUse_DevD	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevD	EC_ChangeEnableSetting		
	R_TRIG_DevD	R_TRIG		
	F_TRIG_DevD	F_TRIG		
	RS_ExFlg_DevD	RS		
	RS_DevD	RS		
	MC_ChangeAxisUse_DevE	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevE	EC_ChangeEnableSetting		
	R_TRIG_DevE	R_TRIG		
	F_TRIG_DevE	F_TRIG		
	RS_ExFlg_DevE	RS		
	RS_DevE	RS		

외부 변수	명칭	데이터 형식	상수	코멘트
	MC_Axis000	_sAXIS_REF	☑	EtherCAT 슬레이브 C의 축 변수
	MC_Axis001	_sAXIS_REF	☑	EtherCAT 슬레이브 D의 축 변수
	MC_Axis002	_sAXIS_REF	☑	EtherCAT 슬레이브 E의 축 변수
	ExclusiveFlg	BOOL	☐	명령 배타 플래그

```
// 실행 조건 트리거 수신
R_TRIG_instance1(Trigger1, Operating1Set);
RS_instance1(
  Set     :=Operating1Set,
  Reset1:=Operating1End,
  Q1      =>Operating1);

// EtherCAT 슬레이브 C의 축 사용을 미사용 축으로 변경
MC_ChangeAxisUse_DevC(
  Axis       :=MC_Axis000,
  Execute  :=(Operating1 & NOT(DoneHold_DevC)),
  AxisUse  :=_mcUnusedAxis);

// EtherCAT 슬레이브 C의 무효화
EC_ChangeEnableSetting_DevC(
  Execute  :=(Operating1 & MC_ChangeAxisUse_DevC.Done & NOT(ExclusiveFlg)),
  NodeAdr :=UINT#1,
  IsEnable :=FALSE);

// 명령의 배타적 제어
R_TRIG_DevC(EC_ChangeEnableSetting_DevC.Busy, ExclusiveFlgSet);
F_TRIG_DevC(EC_ChangeEnableSetting_DevC.Busy, ExclusiveFlgReset);
RS_ExFlg_DevC(
  Set     :=ExclusiveFlgSet,
  Reset1 :=ExclusiveFlgReset,
  Q1      =>ExclusiveFlg);
RS_DevC(
  Set     :=EC_ChangeEnableSetting_DevC.Done,
  Reset1 :=Operating1End,
  Q1      =>DoneHold_DevC);

// EtherCAT 슬레이브 D의 축 사용을 미사용 축으로 변경
MC_ChangeAxisUse_DevD(
  Axis       :=MC_Axis001,
  Execute :=(Operating1 & DoneHold_DevC & NOT(DoneHold_DevD)),
  AxisUse :=_mcUnusedAxis);

// EtherCAT 슬레이브 D의 무효화
EC_ChangeEnableSetting_DevD(
  Execute :=(Operating1 & DoneHold_DevC & MC_ChangeAxisUse_DevD.Done & NOT(ExclusiveFlg)),
  NodeAdr:=UINT#2,
  IsEnable:=FALSE);

// 명령의 배타적 제어
R_TRIG_DevD(EC_ChangeEnableSetting_DevD.Busy, ExclusiveFlgSet);
F_TRIG_DevD(EC_ChangeEnableSetting_DevD.Busy, ExclusiveFlgReset);
RS_ExFlg_DevD(
  Set     :=ExclusiveFlgSet,
  Reset1 :=ExclusiveFlgReset,
  Q1      =>ExclusiveFlg);
RS_DevD(
  Set     :=EC_ChangeEnableSetting_DevD.Done,
  Reset1 :=Operating1End,
  Q1      =>DoneHold_DevD);
```

```
// EtherCAT 슬레이브 E의 축 사용을 미사용 축으로 변경
MC_ChangeAxisUse_DevE(
  Axis      :=MC_Axis002,
  Execute :=(Operating1 & DoneHold_DevD & NOT(DoneHold_DevE)),
  AxisUse :=_mcUnusedAxis);

// EtherCAT 슬레이브 E의 무효화
EC_ChangeEnableSetting_DevE(
  Execute  :=(Operating1 & DoneHold_DevD & MC_ChangeAxisUse_DevE.Done & NOT(ExclusiveFlg)),
  NodeAdr :=UINT#3,
  IsEnable :=FALSE);

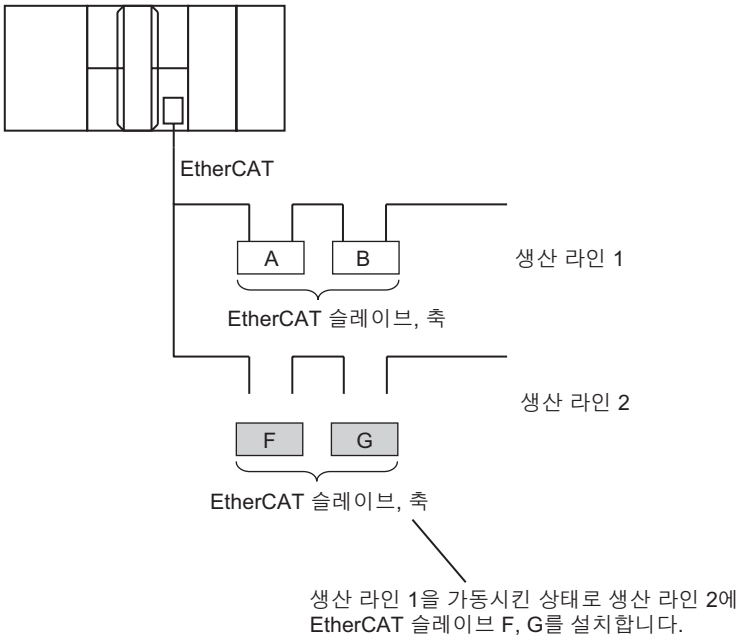
// 명령의 배타적 제어
R_TRIG_DevE(EC_ChangeEnableSetting_DevE.Busy, ExclusiveFlgSet);
F_TRIG_DevE(EC_ChangeEnableSetting_DevE.Busy, ExclusiveFlgReset);
RS_ExFlg_DevE(
  Set      :=ExclusiveFlgSet,
  Reset1 :=ExclusiveFlgReset,
  Q1       =>ExclusiveFlg);
RS_DevE(
  Set      :=EC_ChangeEnableSetting_DevE.Done,
  Reset1 :=Operating1End,
  Q1       =>DoneHold_DevE);

// EtherCAT 슬레이브 E의 미사용 축으로 변경 및 무효화 확인
Operating1End:=(Operating1 & DoneHold_DevE);

// 분리 가능 램프 점등
SR_instance1(
  Set1 :=Operating1End,
  Q1   =>Light1On);
```

● EtherCAT 슬레이브를 EtherCAT 네트워크에 설치하는 예

앞의 예와 동일한 시스템의 생산 라인 1을 가동시킨 상태에서 생산 라인 2에 EtherCAT 슬레이브 F, G를 설치합니다. EtherCAT 슬레이브 F, G에는 모션 제어 축도 설정하는 것으로 가정합니다. 따라서 EtherCAT 슬레이브를 유효화하는 동시에 축 사용을 사용 축으로 변경하는 처리도 수행합니다.



조작 순서

이 샘플 프로그램의 조작 순서는 다음과 같습니다.

1. **조작자가 EtherCAT 슬레이브 F, G를 설치합니다.**
2. **조작자가 표시기의 버튼을 누르면 실행 조건이 ON이 됩니다.**
3. **컨트롤러가 EtherCAT 슬레이브 F, G를 유효화합니다.**
4. **두 EtherCAT 슬레이브의 유효화, 사용 축으로 변경이 모두 완료되면 컨트롤러가 설치 완료 램프를 점등시킵니다.**

축 사용을 사용 축으로 변경하는 명령

축 사용을 사용 축으로 변경하려면 MC_ChangeAxisUse 명령을 사용합니다. MC_ChangeAxisUse 명령 사양에 대한 자세한 내용은 "NJ 시리즈 명령 참조 설명서 모션편(SBCE-364E 이상)"을 참조하십시오.

명령의 배타적 제어

EC_ChangeEnableSetting 명령은 동시에 여러 개 실행할 수 없습니다. 또한 EC_ChangeEnable Setting 명령은 복수의 태스크 주기에 걸쳐 실행됩니다. 따라서 먼저 실행된 EC_Change EnableSetting 명령의 완료를 확인한 후 다음 EC_ChangeEnableSetting 명령을 실행합니다. 그 때문에 'ExclusiveFlg(명령 배타 플래그)' 변수를 사용하며, 'ExclusiveFlg'의 값이 TRUE이면 EC_ChangeEnableSetting 명령이 실행되고 있음을 의미합니다. 'ExclusiveFlg'의 값이 TRUE인 동안은 다음 EC_ChangeEnableSetting 명령을 실행하지 않습니다.

EC_ChangeEnableSetting 명령은 다른 태스크 내의 EC_ChangeEnableSetting 명령과도 동시에 실행할 수 없습니다. 따라서 이 샘플 프로그램에서는 'ExclusiveFlg'를 글로벌 변수로 처리하고 있습니다. 이렇게 하면 이 샘플 프로그램과 다른 태스크 내의 EC_ChangeEnableSetting 명령을 배타적으로 제어할 수 있습니다. 단, 그 경우는 다른 태스크 내에서도 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행해야 합니다.
또한 EC_ChangeEnableSetting 명령은 EC_DisconnectSlave 명령이나 EC_ConnectSlave 명령과도 동시에 실행할 수 없습니다. 🕮 'EC_DisconnectSlave 명령(2-825페이지)'의 샘플 프로그램은 이 샘플 프로그램과 동일한 글로벌 변수 'ExclusiveFlg'를 사용하여 명령의 배타적 제어를 실행하는 예입니다.

각 EtherCAT 슬레이브의 축 변수와 노드 주소

EtherCAT 슬레이브 F, G의 축에 할당된 축 변수와 노드 주소는 다음과 같습니다.

EtherCAT 슬레이브	축 변수	노드 주소
F	MC_Axis003	4
G	MC_Axis004	5

글로벌 변수의 정의

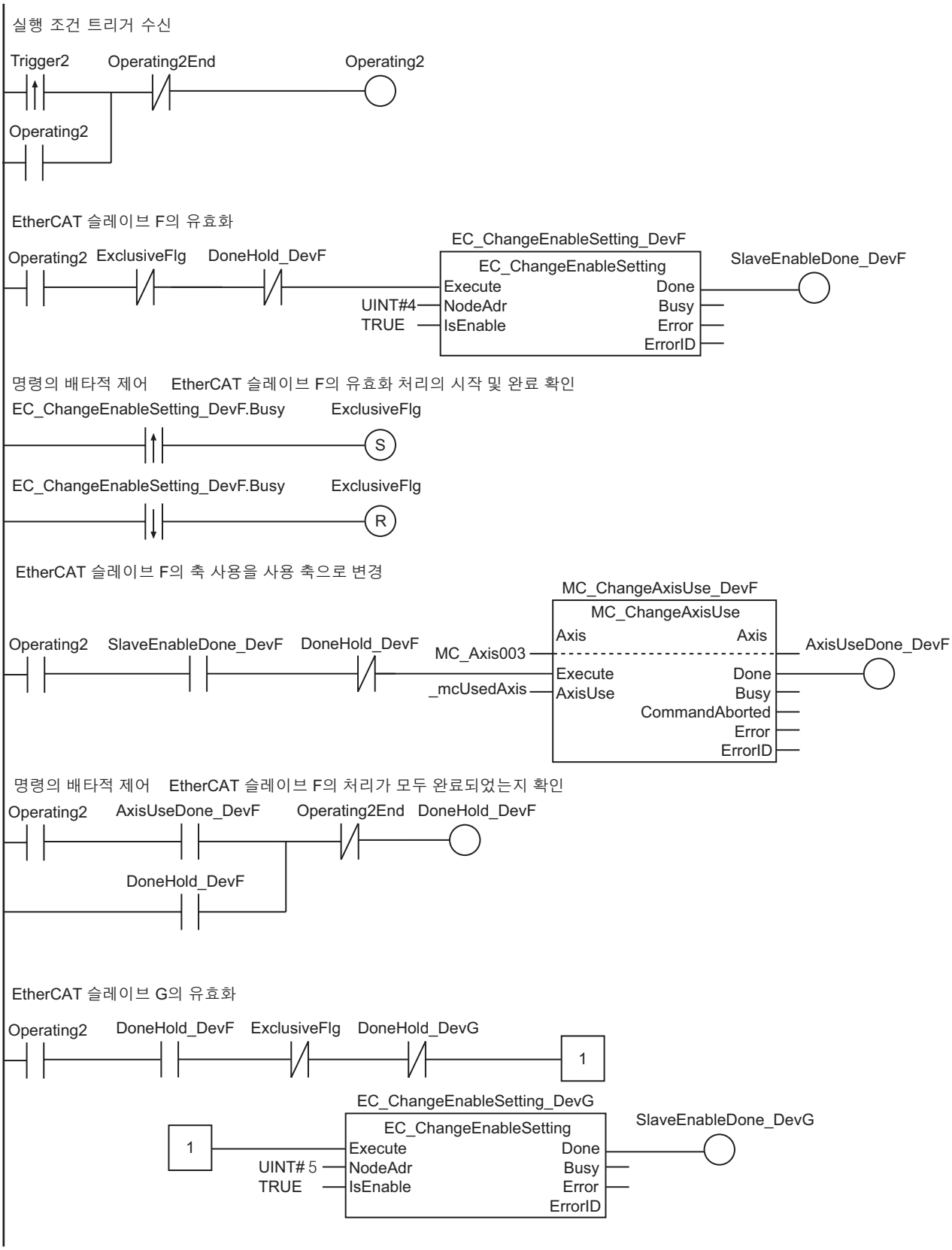
글로벌 변수

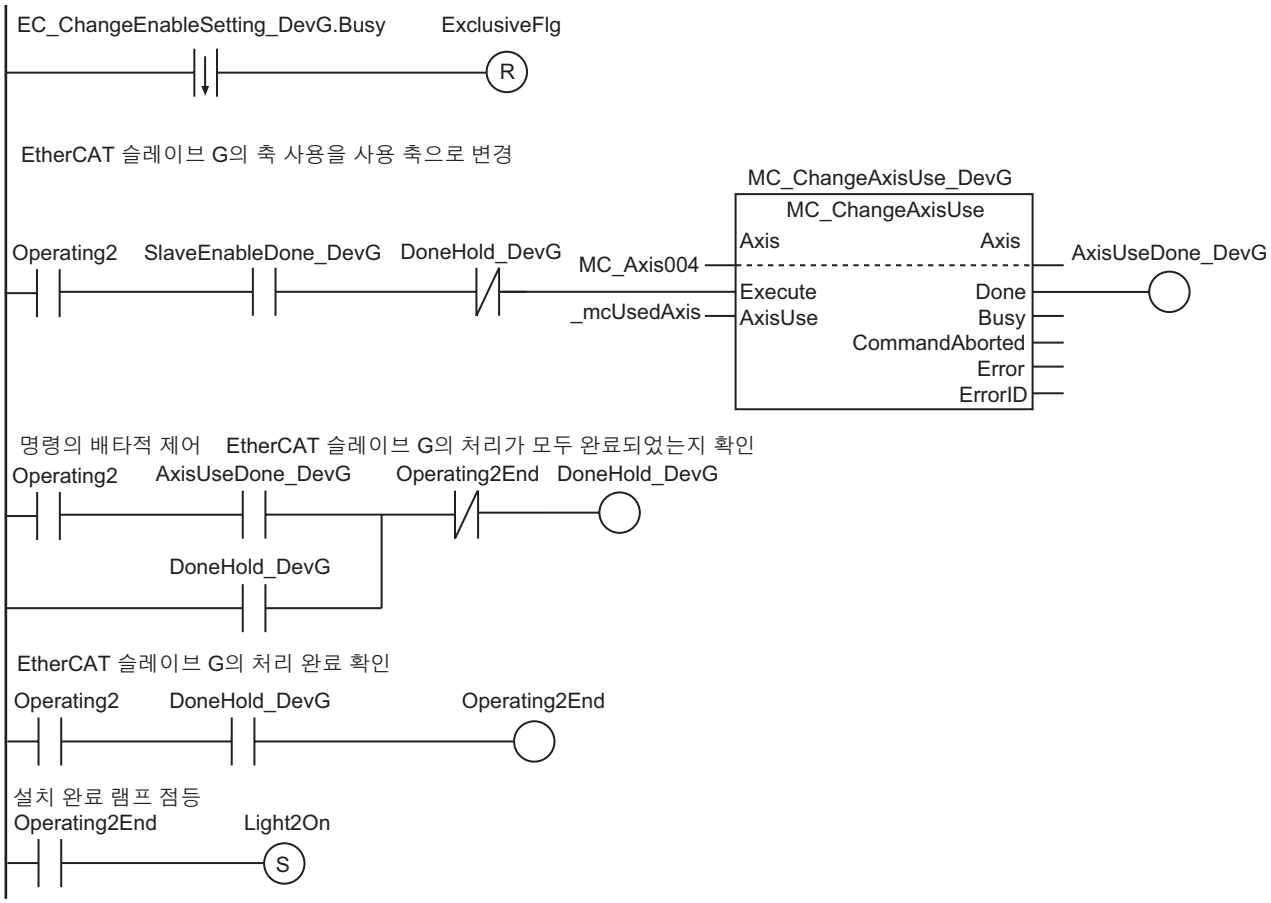
명칭	데이터 형식	초기값	할당대상	상수	코멘트
MC_Axis003	_sAXIS_REF		MC://_MC_AX[3]	☑	EtherCAT 슬레이브 F의 축 변수
MC_Axis004	_sAXIS_REF		MC://_MC_AX[4]	☑	EtherCAT 슬레이브 G의 축 변수
ExclusiveFlg	BOOL	FALSE		☐	명령 배타 플래그

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Operating2End	BOOL	FALSE	처리 종료
	Trigger2	BOOL	FALSE	실행 조건
	Operating2	BOOL	FALSE	처리 중
	AxisUseDone_DevF	BOOL	FALSE	EtherCAT 슬레이브 F의 축 사용을 사용 축으로 변경 완료
	SlaveEnableDone_DevF	BOOL	FALSE	EtherCAT 슬레이브 F의 유효화 완료
	DoneHold_DevF	BOOL	FALSE	EtherCAT 슬레이브 F의 처리 완료 유지
	AxisUseDone_DevG	BOOL	FALSE	EtherCAT 슬레이브 G의 축 사용을 사용 축으로 변경 완료
	SlaveEnableDone_DevG	BOOL	FALSE	EtherCAT 슬레이브 G의 유효화 완료
	DoneHold_DevG	BOOL	FALSE	EtherCAT 슬레이브 G의 처리 완료 유지
	Light2On	BOOL	FALSE	설치 완료 램프 점등
	MC_ChangeAxisUse_DevF	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevF	EC_ChangeEnableSetting		
	MC_ChangeAxisUse_DevG	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevG	EC_ChangeEnableSetting		

외부 변수	명칭	데이터 형식	상수	코멘트
	MC_Axis003	_sAXIS_REF	☑	EtherCAT 슬레이브 F의 축 변수
	MC_Axis004	_sAXIS_REF	☑	EtherCAT 슬레이브 G의 축 변수
	ExclusiveFlg	BOOL	☐	명령 배타 플래그


실행 조건 트리거 수신
Trigger2
Operating2End
Operating2
Operating2
EtherCAT 슬레이브 F의 유효화
Operating2
ExclusiveFlg
DoneHold_DevF
EC_ChangeEnableSetting_DevF
EC_ChangeEnableSetting
Execute
NodeAdr
IsEnable
UINT#4
TRUE
Done
Busy
Error
ErrorID
SlaveEnableDone_DevF
명령의 배타적 제어 EtherCAT 슬레이브 F의 유효화 처리의 시작 및 완료 확인
EC_ChangeEnableSetting_DevF.Busy
ExclusiveFlg
S
EC_ChangeEnableSetting_DevF.Busy
ExclusiveFlg
R
EtherCAT 슬레이브 F의 축 사용을 사용 축으로 변경
Operating2
SlaveEnableDone_DevF
DoneHold_DevF
MC_ChangeAxisUse_DevF
MC_ChangeAxisUse
Axis
Axis
MC_Axis003
Execute
AxisUse
_mcUsedAxis
Done
Busy
CommandAborted
Error
ErrorID
AxisUseDone_DevF
명령의 배타적 제어 EtherCAT 슬레이브 F의 처리가 모두 완료되었는지 확인
Operating2
AxisUseDone_DevF
Operating2End
DoneHold_DevF
DoneHold_DevF
EtherCAT 슬레이브 G의 유효화
Operating2
DoneHold_DevF
ExclusiveFlg
DoneHold_DevG
1
EC_ChangeEnableSetting_DevG
EC_ChangeEnableSetting
1
Execute
NodeAdr
IsEnable
UINT#5
TRUE
Done
Busy
Error
ErrorID
SlaveEnableDone_DevG


EC_ChangeEnableSetting_DevG.Busy
ExclusiveFlg
R
EtherCAT 슬레이브 G의 축 사용을 사용 축으로 변경
MC_ChangeAxisUse_DevG
MC_ChangeAxisUse
Operating2
SlaveEnableDone_DevG
DoneHold_DevG
MC_Axis004
_mcUsedAxis
Axis
Axis
Execute
AxisUse
Done
Busy
CommandAborted
Error
ErrorID
AxisUseDone_DevG
명령의 배타적 제어
EtherCAT 슬레이브 G의 처리가 모두 완료되었는지 확인
Operating2
AxisUseDone_DevG
Operating2End
DoneHold_DevG
DoneHold_DevG
EtherCAT 슬레이브 G의 처리 완료 확인
Operating2
DoneHold_DevG
Operating2End
설치 완료 램프 점등
Operating2End
Light2On
S

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Operating2End	BOOL	FALSE	처리 종료
	Trigger2	BOOL	FALSE	실행 조건
	Operating2	BOOL	FALSE	처리 중
	Operating2Set	BOOL	FALSE	처리 시작
	Light2On	BOOL	FALSE	설치 완료 램프 점등
	DoneHold_DevF	BOOL	FALSE	EtherCAT 슬레이브 F의 처리 완료 유지
	DoneHold_DevG	BOOL	FALSE	EtherCAT 슬레이브 G의 처리 완료 유지
	ExclusiveFlgSet	BOOL	FALSE	명령 배타 플래그 ON
	ExclusiveFlgReset	BOOL	FALSE	명령 배타 플래그 OFF
	R_TRIG_instance2	R_TRIG		
	RS_instance2	RS		
	SR_instance2	SR		
	MC_ChangeAxisUse_DevF	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevF	EC_ChangeEnableSetting		
	R_TRIG_DevF	R_TRIG		
	F_TRIG_DevF	F_TRIG		
	RS_ExFlg_DevF	RS		
	RS_DevF	RS		
	MC_ChangeAxisUse_DevG	MC_ChangeAxisUse		
	EC_ChangeEnableSetting_DevG	EC_ChangeEnableSetting		
	R_TRIG_DevG	R_TRIG		
	F_TRIG_DevG	F_TRIG		
	RS_ExFlg_DevG	RS		
	RS_DevG	RS		

외부 변수	명칭	데이터 형식	상수	코멘트
	MC_Axis003	_sAXIS_REF	☑	EtherCAT 슬레이브 F의 축 변수
	MC_Axis004	_sAXIS_REF	☑	EtherCAT 슬레이브 G의 축 변수
	ExclusiveFlg	BOOL	☐	명령 배타 플래그

```
// 실행 조건 트리거 수신
R_TRIG_instance2(Trigger2, Operating2Set);
RS_instance2(
  Set      :=Operating2Set,
  Reset1 :=Operating2End,
  Q1       =>Operating2);

// EtherCAT 슬레이브 F의 유효화
EC_ChangeEnableSetting_DevF(
  Execute  :=(Operating2 & NOT(ExclusiveFlg) & NOT(DoneHold_DevF)),
  NodeAdr :=UINT#4,
  IsEnable :=TRUE);

// 명령의 배타적 제어 EtherCAT 슬레이브 F의 유효화 처리의 시작 및 완료 확인
R_TRIG_DevF(EC_ChangeEnableSetting_DevF.Busy, ExclusiveFlgSet);
F_TRIG_DevF(EC_ChangeEnableSetting_DevF.Busy, ExclusiveFlgReset);
RS_ExFlg_DevF(
  Set      :=ExclusiveFlgSet,
  Reset1 :=ExclusiveFlgReset,
  Q1       =>ExclusiveFlg);

// EtherCAT 슬레이브 F의 축 사용을 사용 축으로 변경
MC_ChangeAxisUse_DevF(
  Axis       :=MC_Axis003,
  Execute  :=(Operating2 & EC_ChangeEnableSetting_DevF.Done & NOT(DoneHold_DevF)),
  AxisUse  :=_mcUsedAxis);

// 명령의 배타적 제어 EtherCAT 슬레이브 F의 처리가 모두 완료되었는지 확인
RS_DevF(
  Set      :=(Operating2 & MC_ChangeAxisUse_DevF.Done),
  Reset1 :=Operating2End,
  Q1       =>DoneHold_DevF);

// EtherCAT 슬레이브 G의 유효화
EC_ChangeEnableSetting_DevG(
  Execute  :=(Operating2 & DoneHold_DevF & NOT(ExclusiveFlg) & NOT(DoneHold_DevG)),
  NodeAdr :=UINT#5,
  IsEnable :=TRUE);

// 명령의 배타적 제어 EtherCAT슬레이브 G의 유효화 처리의 시작 및 완료 확인
R_TRIG_DevG(EC_ChangeEnableSetting_DevG.Busy, ExclusiveFlgSet);
F_TRIG_DevG(EC_ChangeEnableSetting_DevG.Busy, ExclusiveFlgReset);
RS_ExFlg_DevG(
  Set      :=ExclusiveFlgSet,
  Reset1 :=ExclusiveFlgReset,
  Q1       =>ExclusiveFlg);

// EtherCAT 슬레이브 G의 축 사용을 사용 축으로 변경
MC_ChangeAxisUse_DevG(
  Axis       :=MC_Axis004,
  Execute  :=(Operating2 & EC_ChangeEnableSetting_DevG.Done & NOT(DoneHold_DevG)),
  AxisUse  :=_mcUsedAxis);

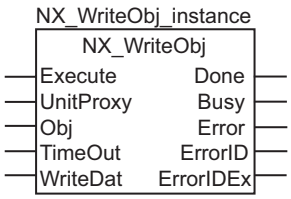
// 명령의 배타적 제어 EtherCAT 슬레이브 G의 처리가 모두 완료되었는지 확인
RS_DevG(
  Set      :=(Operating2 & MC_ChangeAxisUse_DevG.Done),
  Reset1 :=Operating2End,
  Q1       =>DoneHold_DevG);
```

```
// EtherCAT 슬레이브 G의 처리 완료 확인
Operating2End:=Operating2 & DoneHold_DevG;

// 설치 완료 램프 점등
SR_instance2(
  Set1 :=Operating2End,
  Q1   =>Light2On);
```

NX_WriteObj

통신 커플러 유니트 또는 NX 유니트의 NX 개체 데이터를 씁니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_WriteObj	NX 개체 쓰기	FB		NX_WriteObj_instance(Execute, UnitProxy, Obj, TimeOut, WriteDat, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitProxy	지정 유니트	입력	데이터를 쓸 유니트	–	–	(*)
Obj	개체 파라메터		개체 파라메터			–
TimeOut	타임아웃 시간		타임아웃 시간 0인 경우는 2.0s	0~60000	ms	2000 (2.0s)
WriteDat	쓰기 데이터		NX 개체에 쓸 데이터	데이터 형식에 따름	–	(*)

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitProxy	구조체_sNXUNIT_ID 자세한 내용은 기능 설명 참조																			
Obj	구조체_sNXOBJ_ACCESS 자세한 내용은 기능 설명 참조																			
TimeOut							○													
WriteDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	배열 전체도 지정 가능																			

기능

통신 커플러 유니트 또는 NX 유니트의 NX 개체에 'WriteDat'의 내용을 씁니다.
데이터를 쓸 유니트를 'UnitProxy'로 지정합니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다. 그 경우, 데이터 쓰기가 실행되지 않을 수 있습니다.

'UnitProxy'의 데이터 형식은 구조체_sNXUNIT_ID입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식
UnitProxy	지정 유니트	지정할 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소	UINT
IPAdr	IP 주소	통신 커플러 유니트의 IP 주소	BYTE[5]
UnitNo	유니트 번호	지정할 유니트의 유니트 번호	UDINT
Path	경로	지정할 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

'UnitProxy'에는 지정할 유니트에 할당된 디바이스 변수를 전달하십시오.

'Obj'의 데이터 형식은 구조체_sNXOBJ_ACCESS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Obj	개체 파라메터	개체 파라메터	_sNXOBJ_ACCESS	–	–	–
Index	인덱스	인덱스	UINT	데이터 형식에 따름	–	0
Subindex	하위 인덱스	하위 인덱스	USINT			

관련 명령과 실행 순서

이 명령은 통신 커플러 유니트 또는 NX 유니트에 기록한 데이터의 속성에 따라 다른 관련 명령과 연계하여 실행해야 합니다. 각각의 경우의 실행 순서는 다음과 같습니다.

● 실행 순서 1

다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 유니트 재시작 시

1 **📖 'NX_ChangeWriteMode(2-793페이지)'을 사용하여 데이터를 쓸 수 있는 모드로 대상 유니트를 변경합니다.**

2 **이 명령을 사용하여 대상 유니트에 데이터를 씁니다.**

3 **📖 'NX_SaveParam(2-797페이지)'을 사용하여, 기입된 데이터를 저장합니다.**

4 **📖 'RestartNXUnit(2-788페이지)'을 사용하여 대상 유니트를 재시작합니다.**

● 실행 순서 2

다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 있음
- 값이 반영되는 것은 쓴 직후

1 **이 명령을 사용하여 대상 유니트에 데이터를 씁니다.**

2 **📖 'NX_SaveParam(2-797페이지)'을 사용하여, 기입된 데이터를 저장합니다.**

● 실행 순서 3

다음과 같은 속성을 가진 데이터를 쓰는 경우의 순서입니다.

- 전원 OFF 유지 속성 없음

1 **이 명령을 사용하여 대상 유니트에 데이터를 씁니다.**

기술 예

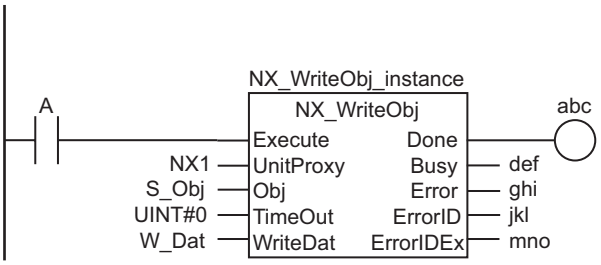
디지털 출력 유니트 NX-OD4121의 부하 차단 시 출력 설정을 현재값 유지로 하는 경우의 예가 아래에 나와 있습니다.

데이터를 쓸 유니트에는 _sNXUNIT_ID 형식의 'NX1'이라는 변수가 할당되어 있는 것으로 합니다.

또한 NX-OD4121의 부하 차단 시 출력 설정의 인덱스는 UINT#16#5011, 하위 인덱스는 USINT#1입니다.

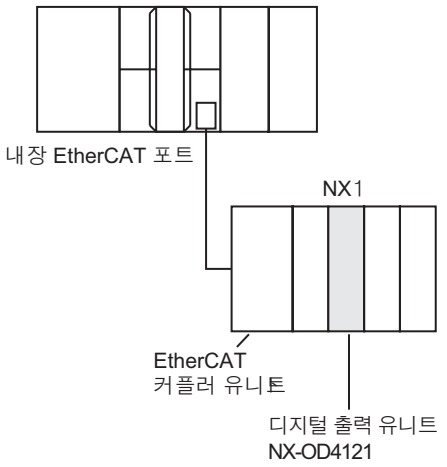
부하 차단 시 출력 설정을 현재값 유지로 하기 위해 쓰는 값은 BYTE#16#01입니다.

LD



ST

```
NX_WriteObj_instance(A, NX1, S_Obj, UINT#0,
                     W_Dat, abc, def, ghi, jkl, mno);
```



개체 파라메터 S_Obj의 내용

S_Obj.Index	UINT#16#5011
S_Obj.Subindex	USINT#1

쓰기 데이터 W_dat의 내용

W_dat	BYTE#16#01

디지털 출력 유니트 NX-OD4121의 부하 차단 시 출력 설정을 현재값으로 유지합니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브가 통신 가능한지 여부를 나타냅니다. TRUE : 통신 가능 FALSE: 통신 불가능

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'WriteDat'가 배열 전체인 경우, 지정한 유니트의 쓰기대상 NX 개체와 동일한 크기로 배열 전체의 크기를, 지정하십시오.
- 'UnitProxy'에는 Sysmac Studio의 I/O 맵에서 통신 커플러 유니트 또는 NX 유니트에 할당한 디바이스 변수를 지정합니다. 디바이스 변수 할당 방법은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편 (SBCA-359H 이상)"를 참조하십시오.
- 'WriteDat'에 전달하는 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 전원 OFF 시, 유지 속성이 있는 데이터를 쓰고 저장하고 싶은 경우는 이 명령 실행 후에 'NX_SaveParam(2-797페이지)'을 실행하십시오. NX_SaveParam 명령을 실행하지 않고 대상 유니트가 재시작되면 NX 개체의 데이터는 쓰기 전의 값으로 돌아갑니다.
- 이 명령은 NX 메시지 통신 이상과 관련된 명령입니다. NX 메시지 통신 이상과 관련된 명령을 한번에 여러 개 실행하면 NX 메시지 통신 이상이 발생할 가능성이 있습니다. NX 메시지 통신 이상과 관련된 명령에 대해서는 'NX 메시지 통신 이상 관련 명령(A-164페이지)'을 참조하십시오.

- 이상이 발생한 경우, 'Error'는 TRUE가 됩니다. 'ErrorID', 'ErrorIDEx'의 값과 그에 대응하는 이상 내용은 다음과 같습니다.

'ErrorID'의 값	'ErrorIDEx'의 값	이상 내용
16#0400	16#00000000	• 'UnitProxy'의 값이 유효 범위 밖입니다. • 'TimeOut'의 값이 유효 범위 밖입니다.
16#0419	16#00000000	• 'UnitProxy'의 데이터 형식이 잘못되었습니다. • 'WriteDat'의 데이터 형식이 잘못되었습니다.
16#041B	16#00000000	'WriteDat'에 2048바이트를 초과하는 데이터가 지정되어 있습니다.
16#2C00	16#00001110	'Obj.Index'의 값에 대응한 개체가 존재하지 않습니다.
	16#00001111	'Obj.Subindex'의 값에 대응한 개체가 존재하지 않습니다.
	16#00002101	쓰기대상 NX 개체가 쓰기 불가능합니다.
	16#00002110	'WriteDat'의 값이 쓰기대상 NX 개체의 값의 범위 밖입니다.
	16#00002210	지정된 유니트가 데이터를 쓸 수 있는 모드가 아닙니다.
	16#0000250F	하드웨어 액세스에 실패했습니다. 명령을 다시 실행하십시오.
	16#00002602	지정한 유니트에서는 이 명령을 사용할 없습니다. 유니트의 버전을 확인하십시오.
	16#00010000	지정된 유니트가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00110000	지정된 포트 번호가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00120000 16#00130000 16#00150000	'UnitProxy'의 값이 잘못되었습니다. 지정할 유니트를 나타내는 변수를 다시 설정하십시오.
	16#00140000	지정된 노드 주소가 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00310000	지정된 유니트가 연결 대상이 아닙니다. 유니트의 버전을 확인하십시오.
	16#80010000	지정된 유니트가 다른 작업 중입니다. 명령을 다시 실행하십시오.
	16#80020000 16#80030000 16#81030000 16#82000000	통신 네트워크에서 이상이 발생했습니다. 통신 부하를 줄이십시오.
	16#80040000 16#81000000 16#82010000 16#82040000 16#82050000 16#90000000	통신 네트워크에서 이상이 발생했습니다. 유니트나 케이블의 연결을 확인하십시오. 유니트의 전원이 켜져 있는지 확인하십시오.
16#2C01	16#00000000	동시에 실행할 수 있는 명령 수를 초과했습니다.
16#2C02	16#00000000	통신 실행 중에 타임아웃이 발생했습니다.
16#2C03	16#00000000	송신 메시지의 크기가 잘못되었습니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.

샘플 프로그램

- NX 유니트에 전원 OFF 유지 속성이 있고 유니트 재시작 시 값이 반영되는 데이터를 쓰는 예

EtherCAT 커플러 유니트에 연결되어 있는 AC 입력 유니트 NX-AD2203의 개체 파라메터 'Ch1 입력 이동 평균 시간'을 500 μs로 설정합니다.
EtherCAT 커플러 유니트의 노드 주소는 10으로 합니다.

개체 파라메터 'Ch1 입력 이동 평균 시간'의 사양은 다음과 같습니다.

항목	값
인덱스	16#5004
하위 인덱스	16#01
500 μs에 대응하는 설정 데이터	2

개체 파라메터 'Ch1 입력 이동 평균 시간'은 전원 OFF 유지 속성이 있고 유니트 재시작 시 값이 반영되는 데이터입니다. 따라서 처리 순서는 다음과 같습니다.

1 **NX_ChangeWriteMode 명령을 사용하여, 데이터를 쓸 수 있는 모드로 유니트를 변경합니다.**

2 **NX_WriteObj 명령을 사용하여 유니트에 데이터를 씁니다.**

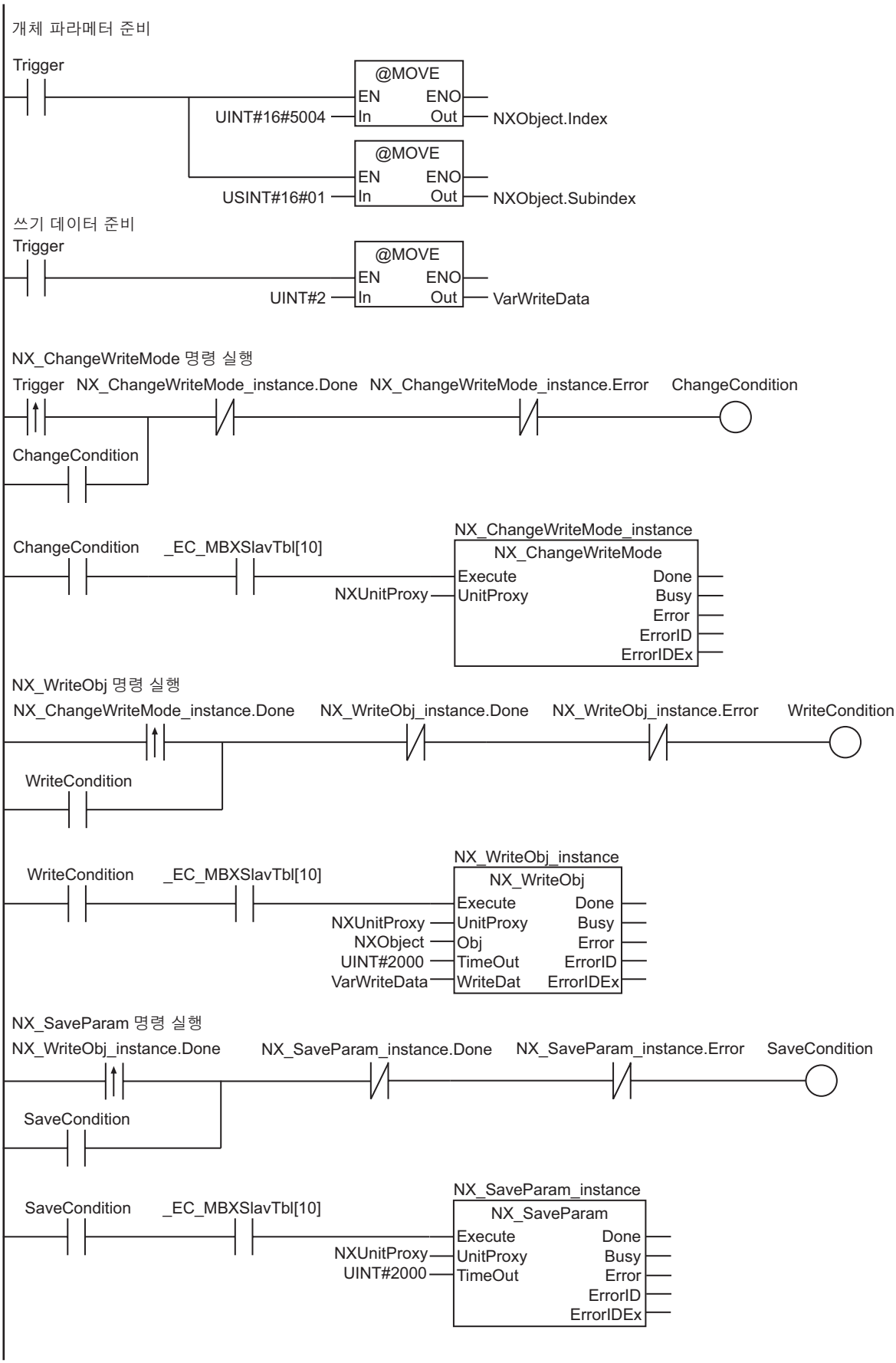
3 **NX_SaveParam 명령을 사용하여, 기입된 데이터를 저장합니다.**

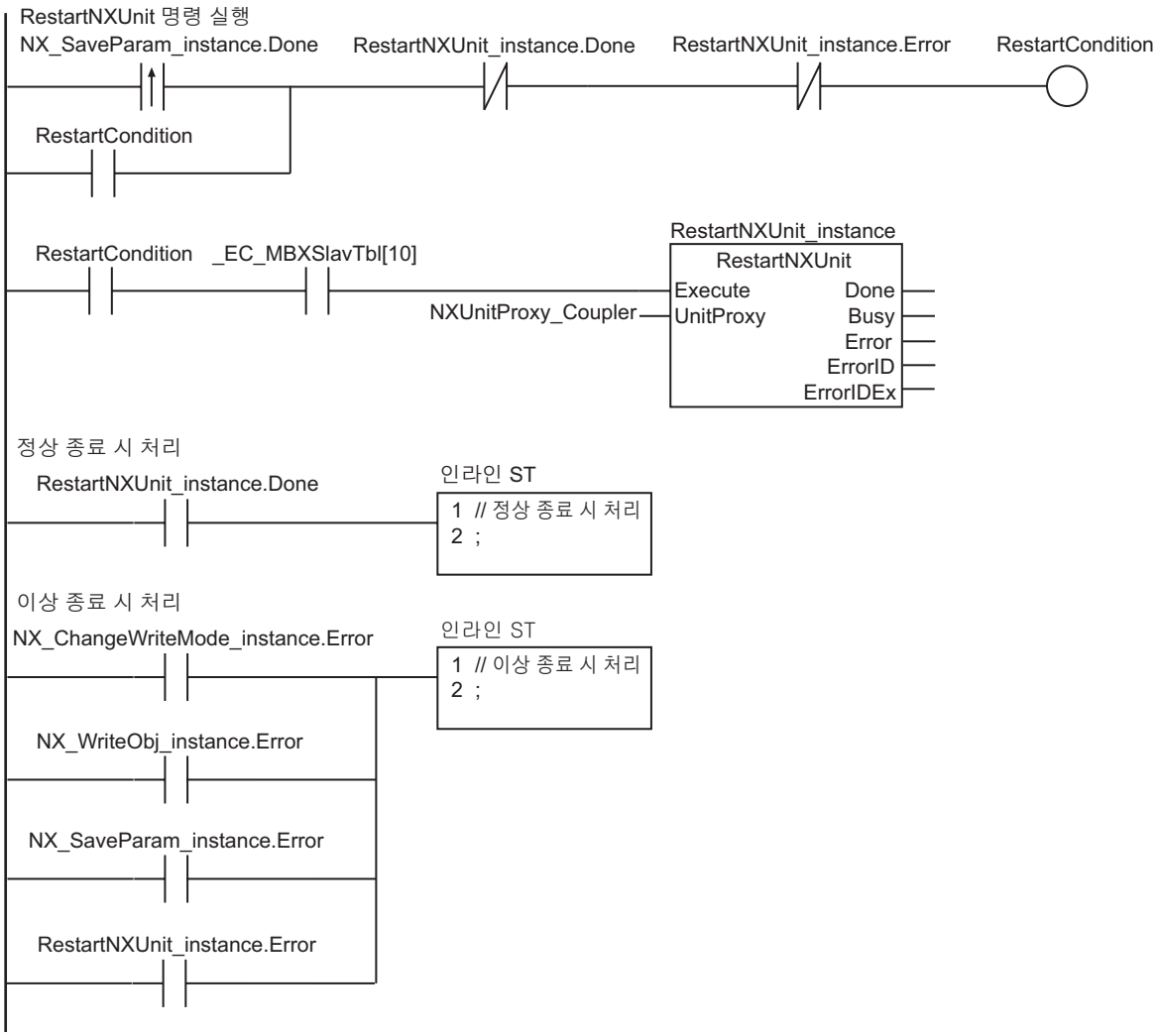
4 **RestartNXUnit 명령을 사용하여 유니트를 재시작합니다.**

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	ChangeCondition	BOOL	FALSE	쓰기 모드 변경 실행 조건
	WriteCondition	BOOL	FALSE	데이터 쓰기 실행 조건
	SaveCondition	BOOL	FALSE	데이터 저장 실행 조건
	RestartCondition	BOOL	FALSE	유니트 재시작 실행 조건
	NXUnitProxy	_sNXUNIT_ID		유니트 지정(DC 입력 유니트)
	NXUnitProxy_Coupler	_sNXUNIT_ID		유니트 지정(EtherCAT 커플러 유니트)
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	VarWriteData	UINT	0	쓰기 데이터
	NX_ChangeWriteMode_instance	NX_ChangeWriteMode		
	NX_WriteObj_instance	NX_WriteObj		
	NX_SaveParam_instance	NX_SaveParam		
	RestartNXUnit_instance	RestartNXUnit		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	☑	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블


개체 파라메터 준비
Trigger
@MOVE
EN
ENO
UINT#16#5004
In
Out
NXObject.Index
@MOVE
EN
ENO
USINT#16#01
In
Out
NXObject.Subindex
쓰기 데이터 준비
Trigger
@MOVE
EN
ENO
UINT#2
In
Out
VarWriteData
NX_ChangeWriteMode 명령 실행
Trigger
NX_ChangeWriteMode_instance.Done
NX_ChangeWriteMode_instance.Error
ChangeCondition
ChangeCondition
NX_ChangeWriteMode_instance
NX_ChangeWriteMode
ChangeCondition
_EC_MBXSlavTbl[10]
Execute
Done
NXUnitProxy
UnitProxy
Busy
Error
ErrorID
ErrorIDEx
NX_WriteObj 명령 실행
NX_ChangeWriteMode_instance.Done
NX_WriteObj_instance.Done
NX_WriteObj_instance.Error
WriteCondition
WriteCondition
NX_WriteObj_instance
NX_WriteObj
WriteCondition
_EC_MBXSlavTbl[10]
Execute
Done
NXUnitProxy
UnitProxy
Busy
NXObject
Obj
Error
UINT#2000
TimeOut
ErrorID
VarWriteData
WriteDat
ErrorIDEx
NX_SaveParam 명령 실행
NX_WriteObj_instance.Done
NX_SaveParam_instance.Done
NX_SaveParam_instance.Error
SaveCondition
SaveCondition
NX_SaveParam_instance
NX_SaveParam
SaveCondition
_EC_MBXSlavTbl[10]
Execute
Done
NXUnitProxy
UnitProxy
Busy
UINT#2000
TimeOut
Error
ErrorID
ErrorIDEx


RestartNXUnit 명령 실행
NX_SaveParam_instance.Done
RestartNXUnit_instance.Done
RestartNXUnit_instance.Error
RestartCondition
RestartCondition
RestartCondition
_EC_MBXSlavTbl[10]
RestartNXUnit_instance
RestartNXUnit
Execute
Done
NXUnitProxy_Coupler
UnitProxy
Busy
Error
ErrorID
ErrorIDEx
정상 종료 시 처리
RestartNXUnit_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
NX_ChangeWriteMode_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
NX_WriteObj_instance.Error
NX_SaveParam_instance.Error
RestartNXUnit_instance.Error

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	ChangeCondition	BOOL	FALSE	쓰기 모드 변경 실행 조건
	ChangeGo	BOOL	FALSE	쓰기 모드 변경 실행
	WriteCondition	BOOL	FALSE	데이터 쓰기 실행 조건
	WriteGo	BOOL	FALSE	데이터 쓰기 실행
	SaveCondition	BOOL	FALSE	데이터 저장 실행 조건
	SaveGo	BOOL	FALSE	데이터 저장 실행
	RestartCondition	BOOL	FALSE	유니트 재시작 실행 조건
	RestartGo	BOOL	FALSE	유니트 재시작 실행
	NXUnitProxy	_sNXUNIT_ID		유니트 지정(DC 입력 유니트)
	NXUnitProxy_Coupler	_sNXUNIT_ID		유니트 지정(EtherCAT 커플러 유니트)
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	VarWriteData	UINT	0	쓰기 데이터
	NormalEnd	UINT	0	정상 종료
	ErrorEnd	UINT	0	이상 종료
	NX_ChangeWriteMode _instance	NX_ChangeWriteMode		
	NX_WriteObj_instance	NX_WriteObj		
	NX_SaveParam_instance	NX_SaveParam		
	RestartNXUnit_instance	RestartNXUnit		
	R_Trig_instance	R_TRIG		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	☑	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블

```
// 개체 파라메터와 쓰기 데이터의 준비
R_Trig_instance(Clk   := Trigger);
IF (R_Trig_instance.Q=TRUE)THEN
  NXObject.Index      := UINT#16#5004;
  NXObject.Subindex   := USINT#1;
  VarWriteData        := UINT#2;
END_IF;

// NX_ChangeWriteMode 명령 실행
IF (Trigger = TRUE) THEN
  ChangeCondition   := TRUE;
END_IF;

IF ((NX_ChangeWriteMode_instance.Done=TRUE) OR (NX_ChangeWriteMode_instance.Error=TRUE))THEN
  ChangeCondition := FALSE;
END_IF;

ChangeGo := ChangeCondition & _EC_MBXSlavTbl[10];
NX_ChangeWriteMode_instance(
  Execute    := ChangeGo,
  UnitProxy  := NXUnitProxy);

// NX_WriteObj 명령 실행
IF (NX_ChangeWriteMode_instance.Done=TRUE) THEN
  WriteCondition := TRUE;
END_IF;
```

```
IF ((NX_WriteObj_instance.Done=TRUE) OR (NX_WriteObj_instance.Error=TRUE))THEN
  WriteCondition := FALSE;
END_IF;

WriteGo := WriteCondition & _EC_MBXSlavTbl[10];
NX_WriteObj_instance(
  Execute     := WriteGo,
  UnitProxy   := NXUnitProxy,
  Obj         := NXObject,
  TimeOut     := UINT#2000,
  WriteDat    := VarWriteData);

// NX_SaveParam 실행
IF (NX_WriteObj_instance.Done=TRUE) THEN
  SaveCondition := TRUE;
END_IF;

IF ((NX_SaveParam_instance.Done=TRUE) OR (NX_SaveParam_instance.Error=TRUE))THEN
  SaveCondition := FALSE;
END_IF;

SaveGo := SaveCondition & _EC_MBXSlavTbl[10];
NX_SaveParam_instance(
  Execute     := SaveGo,
  UnitProxy   := NXUnitProxy,
  TimeOut     := UINT#2000);

// RestartNXUnit 실행
IF (NX_SaveParam_instance.Done=TRUE) THEN
  RestartCondition := TRUE;
END_IF;

IF ((RestartNXUnit_instance.Done=TRUE) OR (RestartNXUnit_instance.Error=TRUE)) THEN
  RestartCondition := FALSE;
END_IF;

RestartGo := RestartCondition & _EC_MBXSlavTbl[10];
RestartNXUnit_instance(
  Execute     := SaveGo,
  UnitProxy   := NXUnitProxy_Coupler);

IF (RestartNXUnit_instance.Done=TRUE) THEN
   // 정상 종료 시 처리
   NormalEnd := NormalEnd + UINT#1;
ELSIF ((NX_ChangeWriteMode_instance.Error=TRUE) OR (NX_WriteObj_instance.Error=TRUE)
      OR (NX_SaveParam_instance.Error=TRUE) OR (RestartNXUnit_instance.Error=TRUE)) THEN
  // 이상 종료 시 처리
  ErrorEnd := ErrorEnd + UINT#1;
END_IF;
```

- NX 유니트에 전원 OFF 유지 속성이 있고, 쓴 직후에 값이 반영되는 데이터를 쓰는 예

EtherCAT 커플러 유니트에 연결되어 있는 온도 입력 유니트 NX-TS2101의 개체 파라메터 'Ch1 오프셋 값(1점 보정)'을 +0.3℃로 설정합니다.
EtherCAT 커플러 유니트의 노드 주소는 10으로 합니다.

개체 파라메터 'Ch1 오프셋 값(1점 보정)'의 사양은 다음과 같습니다.

항목	값
인덱스	16#5010
하위 인덱스	16#01
쓸 값	0.3

개체 파라메터 'Ch1 오프셋 값(1점 보정)'은 전원 OFF 유지 속성이 있고, 쓴 직후에 값이 반영되는 데이터입니다. 따라서 처리 순서는 다음과 같습니다.

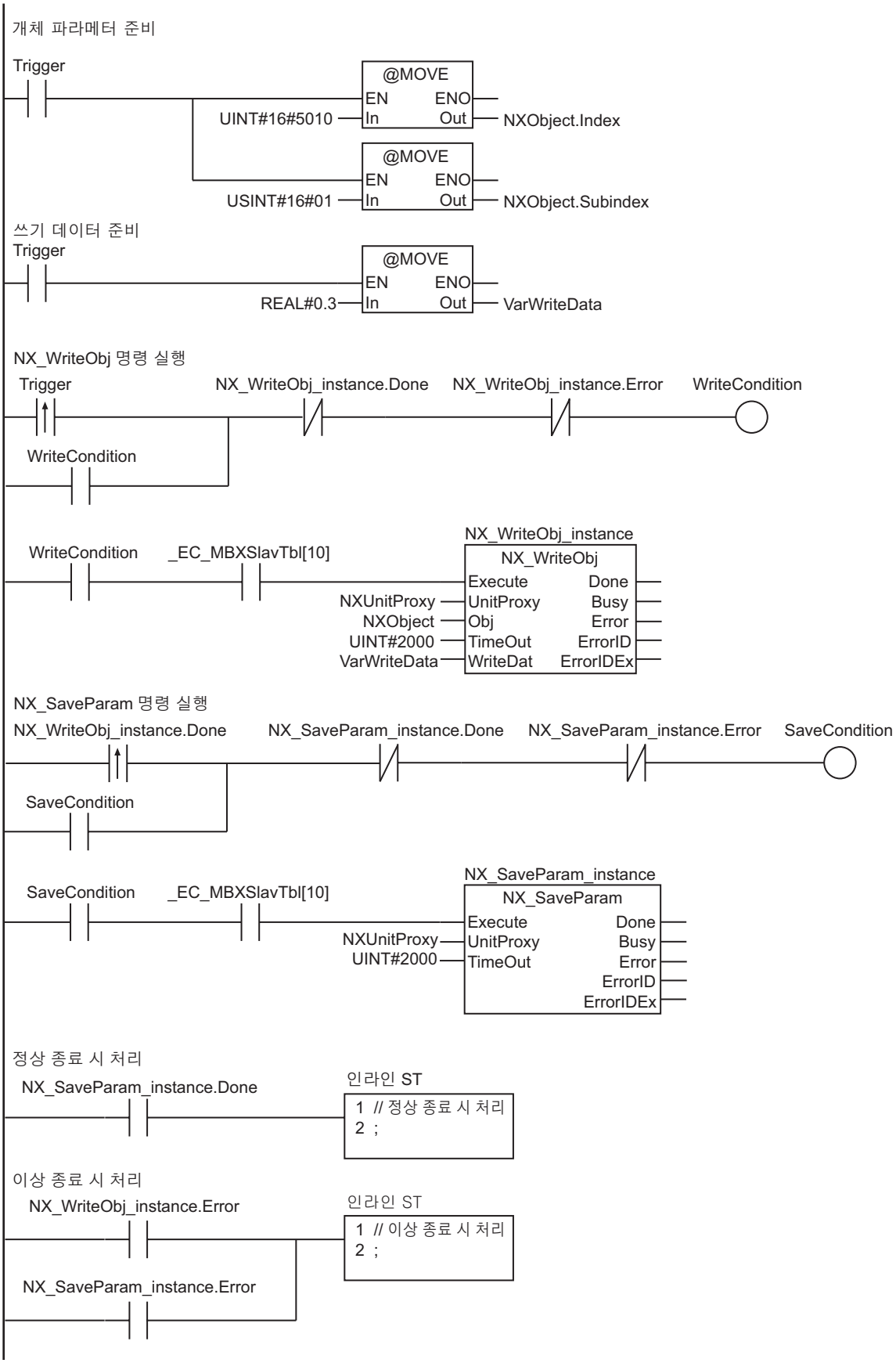
1 **NX_WriteObj를 사용하여 대상 유니트에 데이터를 씁니다.**

2 **NX_SaveParam을 사용하여, 기입된 데이터를 저장합니다.**

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	WriteCondition	BOOL	FALSE	데이터 쓰기 실행 조건
	SaveCondition	BOOL	FALSE	데이터 저장 실행 조건
	NXUnitProxy	_sNXUNIT_ID		유니트 지정(AC 입력 유니트)
	NXUnitProxy_Coupler	_sNXUNIT_ID		유니트 지정(EtherCAT 커플러 유니트)
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	VarWriteData	Real	0.0	쓰기 데이터
	NX_WriteObj_instance	NX_WriteObj		
	NX_SaveParam_instance	NX_SaveParam		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	⍁	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블


개체 파라메터 준비
Trigger
@MOVE
EN
ENO
UINT#16#5010
In
Out
NXObject.Index
@MOVE
EN
ENO
USINT#16#01
In
Out
NXObject.Subindex
쓰기 데이터 준비
Trigger
@MOVE
EN
ENO
REAL#0.3
In
Out
VarWriteData
NX_WriteObj 명령 실행
Trigger
NX_WriteObj_instance.Done
NX_WriteObj_instance.Error
WriteCondition
WriteCondition
WriteCondition
_EC_MBXSlavTbl[10]
NX_WriteObj_instance
NX_WriteObj
Execute
Done
NXUnitProxy
UnitProxy
Busy
NXObject
Obj
Error
UINT#2000
TimeOut
ErrorID
VarWriteData
WriteDat
ErrorIDEx
NX_SaveParam 명령 실행
NX_WriteObj_instance.Done
NX_SaveParam_instance.Done
NX_SaveParam_instance.Error
SaveCondition
SaveCondition
SaveCondition
_EC_MBXSlavTbl[10]
NX_SaveParam_instance
NX_SaveParam
Execute
Done
NXUnitProxy
UnitProxy
Busy
UINT#2000
TimeOut
Error
ErrorID
ErrorIDEx
정상 종료 시 처리
NX_SaveParam_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
NX_WriteObj_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
NX_SaveParam_instance.Error

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	WriteCondition	BOOL	FALSE	데이터 쓰기 실행 조건
	WriteGo	BOOL	FALSE	데이터 쓰기 실행
	SaveCondition	BOOL	FALSE	데이터 저장 실행 조건
	SaveGo	BOOL	FALSE	데이터 저장 실행
	NXUnitProxy	_sNXUNIT_ID		유니트 지정(온도 입력 유니트)
	NXUnitProxy_Coupler	_sNXUNIT_ID		유니트 지정(EtherCAT 커플러 유니트)
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	VarWriteData	REAL	0.0	쓰기 데이터
	NormalEnd	UINT	0	정상 종료
	ErrorEnd	UINT	0	이상 종료
	NX_WriteObj_instance	NX_WriteObj		
	NX_SaveParam_instance	NX_SaveParam		
	R_Trig_instance	R_TRIG		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	☑	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블

```
// 개체 파라메터와 쓰기 데이터의 준비
R_Trig_instance(Clk   := Trigger);
IF (R_Trig_instance.Q=TRUE)THEN
  NXObject.Index    := UINT#16#5004;
  NXObject.Subindex := USINT#1;
  VarWriteData      := UINT#2;
END_IF;

// NX_WriteObj 명령 실행
IF (Trigger=TRUE) THEN
  WriteCondition := TRUE;
END_IF;

IF ((NX_WriteObj_instance.Done=TRUE) OR (NX_WriteObj_instance.Error=TRUE))THEN
  WriteCondition := FALSE;
END_IF;

WriteGo := WriteCondition & _EC_MBXSlavTbl[10];
NX_WriteObj_instance(
  Execute   := WriteGo,
  UnitProxy := NXUnitProxy,
  Obj       := NXObject,
  TimeOut   := UINT#2000,
  WriteDat  := VarWriteData);

// NX_SaveParam 실행
IF (NX_WriteObj_instance.Done=TRUE) THEN
  SaveCondition := TRUE;
END_IF;

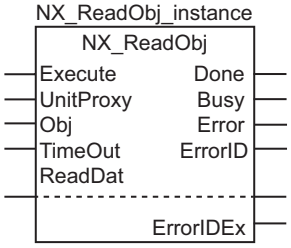
IF ((NX_SaveParam_instance.Done=TRUE) OR (NX_SaveParam_instance.Error=TRUE))THEN
  SaveCondition := FALSE;
END_IF;
```

```
SaveGo := SaveCondition & _EC_MBXSlavTbl[10];
NX_SaveParam_instance(
  Execute    := SaveGo,
  UnitProxy  := NXUnitProxy,
  TimeOut    := UINT#2000);

IF (NX_SaveParam_instance.Done=TRUE) THEN
   // 정상 종료 시 처리
   NormalEnd := NormalEnd + UINT#1;
ELSIF ((NX_WriteObj_instance.Error=TRUE) OR (NX_SaveParam_instance.Error=TRUE)) THEN
  // 이상 종료 시 처리
  ErrorEnd := ErrorEnd + UINT#1;
END_IF;
```

NX_ReadObj

통신 커플러 유니트 또는 NX 유니트의 NX 개체 에서 데이터를 읽습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_ReadObj	NX 개체 읽기	FB		NX_ReadObj_instance(Execute, UnitProxy, Obj, TimeOut, ReadDat, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
UnitProxy	지정 유니트	입력	데이터를 읽을 유니트	–	–	(*)
Obj	개체 파라메터		개체 파라메터			–
TimeOut	타임아웃 시간		타임아웃 시간 0인 경우는 2.0s	0~60000	ms	2000 (2.0s)
ReadDat	읽기 데이터	입출력	NX 개체에서 읽은 데이터	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitProxy	구조체_sNXUNIT_ID 자세한 내용은 기능 설명 참조																			
Obj	구조체_sNXOBJ_ACCESS 자세한 내용은 기능 설명 참조																			
TimeOut							○													
ReadDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	배열 전체도 지정 가능																			

기능

통신 커플러 유니트 또는 NX 유니트의 NX 개체에서 데이터를 읽어서 'ReadDat'에 저장합니다.
데이터를 읽을 유니트를 'UnitProxy'로 지정합니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다. 그 경우, 데이터 읽기는 실행되지 않습니다.

'UnitProxy'의 데이터 형식은 구조체_sNXUNIT_ID입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식
UnitProxy	지정 유니트	지정할 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소	UINT
IPAdr	IP 주소	통신 커플러 유니트의 IP 주소	BYTE[5]
UnitNo	유니트 번호	지정할 유니트의 유니트 번호	UDINT
Path	경로	지정할 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

'UnitProxy'에는 지정할 유니트에 할당된 디바이스 변수를 전달하십시오.

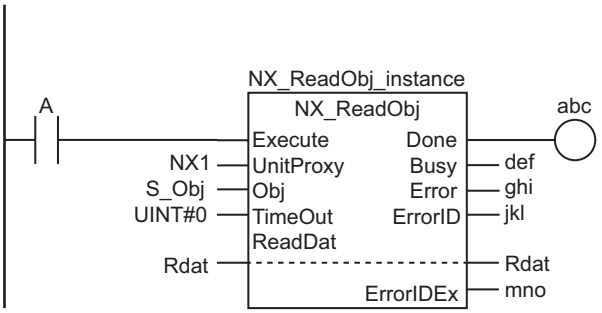
'Obj'의 데이터 형식은 구조체_sNXOBJ_ACCESS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Obj	개체 파라메터	개체 파라메터	_sNXOBJ_ACCESS	–	–	–
Index	인덱스	인덱스	UINT	데이터 형식에 따름	–	0
Subindex	하위 인덱스	하위 인덱스	USINT			

기술 예

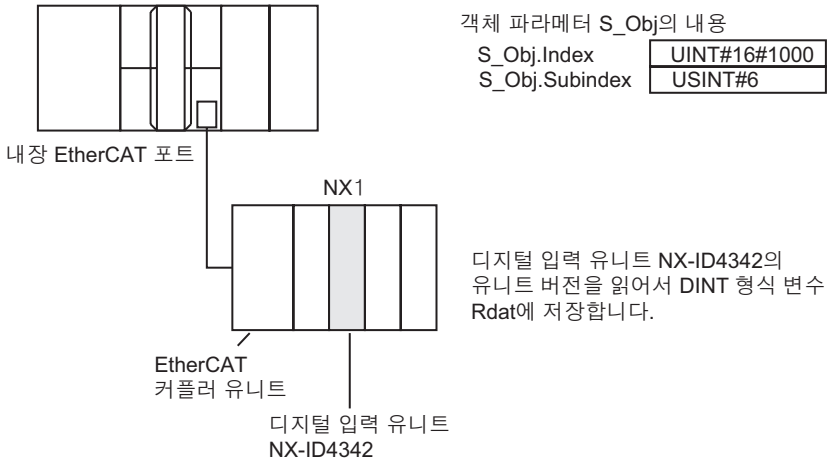
디지털 입력 유니트 NX-ID4342의 유니트 버전을 읽은 경우의 예가 아래에 나와 있습니다.
읽은 데이터는 UDINT 형식 변수 Rdat에 저장됩니다.
데이터를 읽을 유니트에는 _sNXUNIT_ID 형식의 'NX1'이라는 변수가 할당되어 있는 것으로 합니다.
또한 NX-ID4342의 유니트 버전의 인덱스는 UINT#16#1000, 하위 인덱스는 USINT#6입니다.

LD



ST

```
NX_ReadObj_instance(A, NX1, S_Obj, UINT#0,
                    Rdat, abc, def, ghi, jkl, mno);
```



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EC_MBXSlavTbl[i] i는 노드 주소	메시지 통신 가능 슬레이브 테이블	BOOL	대응하는 슬레이브가 통신 가능한지 여부를 나타냅니다. TRUE : 통신 가능 FALSE: 통신 불가능

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'ReadDat'가 배열 전체인 경우, 지정한 유니트의 읽기 원본인 NX 개체와 동일한 크기로 배열 전체의 크기를 지정하십시오.
- 'UnitProxy'에는 Sysmac Studio의 I/O 맵에서 통신 커플러 유니트 또는 NX 유니트에 할당한 디바이스 변수를 지정합니다. 디바이스 변수 할당 방법은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어 편(SBCA-359H 이상)"를 참조하십시오.
- 이 명령은 NX 메시지 통신 이상과 관련된 명령입니다. NX 메시지 통신 이상과 관련된 명령을 한번에 여러 개 실행하면 NX 메시지 통신 이상이 발생할 가능성이 있습니다. NX 메시지 통신 이상과 관련된 명령에 대해서는 'NX 메시지 통신 이상 관련 명령(A-164페이지)'을 참조하십시오.

- 이상이 발생한 경우, 'Error'는 TRUE가 됩니다. 'ErrorID', 'ErrorIDEx'의 값과 그에 대응하는 이상 내용은 다음과 같습니다.

'ErrorID'의 값	'ErrorIDEx'의 값	이상 내용
16#0400	16#00000000	• 'UnitProxy'의 값이 유효 범위 밖입니다. • 'TimeOut'의 값이 유효 범위 밖입니다.
16#0410	16#00000000	'ReadDat'가 STRING 형식이고 NULL 문자로 끝나지 않습니다.
16#0419	16#00000000	• 'UnitProxy'의 데이터 형식이 잘못되었습니다. • 'ReadDat'의 데이터 형식이 잘못되었습니다.
16#041C	16#00000000	'ReadDat'의 크기와 읽기 원본 NX 개체의 크기가 일치하지 않습니다.
16#2C00	16#00001110	'Obj.Index'의 값에 대응한 개체가 존재하지 않습니다.
	16#00001111	'Obj.Subindex'의 값에 대응한 개체가 존재하지 않습니다.
	16#0000250F	하드웨어 액세스에 실패했습니다. 명령을 다시 실행하십시오.
	16#00002602	지정한 유니트에서는 이 명령을 사용할 없습니다. 유니트의 버전을 확인하십시오.
	16#00010000	지정된 유니트가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00110000	지정된 포트 번호가 존재하지 않습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00120000 16#00130000 16#00150000	'UnitProxy'의 값이 잘못되었습니다. 지정할 유니트를 나타내는 변수를 다시 설정하십시오.
	16#00140000	지정된 노드 주소가 잘못되었습니다. 유니트의 구성이 정확한지 확인하십시오.
	16#00310000	지정된 유니트가 연결 대상이 아닙니다. 유니트의 버전을 확인하십시오.
	16#80010000	지정된 유니트가 다른 작업 중입니다. 명령을 다시 실행하십시오.
	16#80020000 16#80030000 16#81030000 16#82000000	통신 네트워크에서 이상이 발생했습니다. 통신 부하를 줄이십시오.
	16#80040000 16#81000000 16#82010000 16#82040000 16#82050000 16#90000000	통신 네트워크에서 이상이 발생했습니다. 유니트나 케이블의 연결을 확인하십시오. 유니트의 전원이 켜져 있는지 확인하십시오.
16#2C01	16#00000000	동시에 실행할 수 있는 명령 수를 초과했습니다.
16#2C02	16#00000000	통신 실행 중에 타임아웃이 발생했습니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 사용할 수 있습니다.

샘플 프로그램

EtherCAT 커플러 유니트 NX-ECC201의 개체 파라메터 'I/O 리프레시 방식 1'의 값을 읽습니다.
EtherCAT 커플러 유니트의 노드 주소는 10으로 합니다.

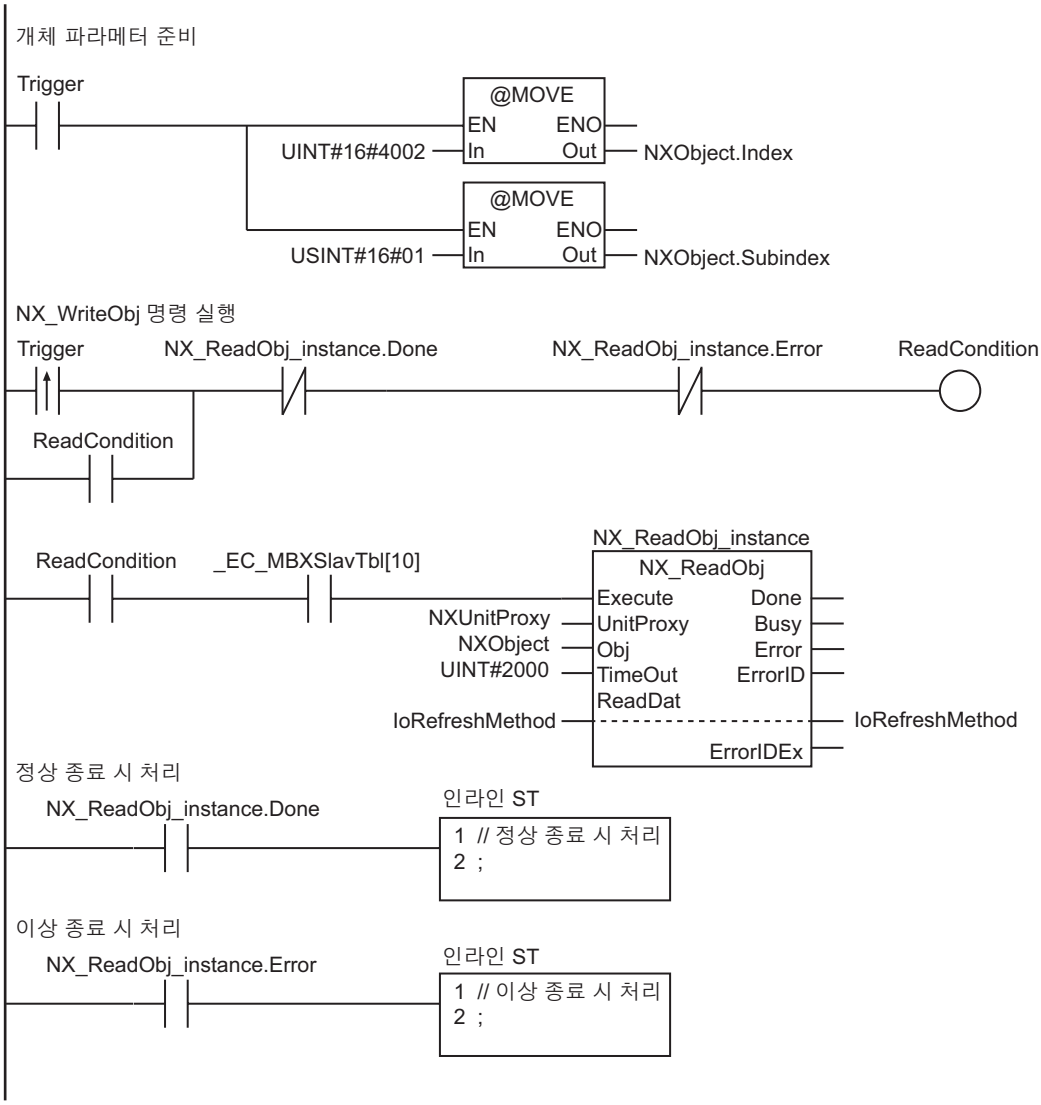
개체 파라메터 'I/O 리프레시 방식 1'의 인덱스와 하위 인덱스의 값은 다음과 같습니다.

항목	값
인덱스	16#4002
하위 인덱스	16#01

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	ReadCondition	BOOL	FALSE	데이터 읽기 실행 조건
	NXUnitProxy	_sNXUNIT_ID		유니트 지정
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	IoRefreshMethod	USINT	0	읽기 데이터
	NX_ReadObj_instance	NX_ReadObj		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	☑	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	ReadCondition	BOOL	FALSE	데이터 읽기 실행 조건
	ReadGo	BOOL	FALSE	데이터 읽기 실행
	NXUnitProxy	_sNXUNIT_ID		유니트 지정
	NXObject	_sNXOBJ_ACCESS	(Index:=0, Subindex:=0)	개체 파라메터
	IoRefreshMethod	USINT	0	읽기 데이터
	NormalEnd	UINT	0	정상 종료
	ErrorEnd	UINT	0	이상 종료
	R_Trig_instance	R_Trig		
	NX_ReadObj_instance	NX_ReadObj		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EC_MBXSlavTbl	▽	ARRAY[1..192] OF BOOL	메시지 통신 가능 슬레이브 테이블

```
// 개체 파라메터의 준비
R_Trig_instance(Clk    := Trigger);
IF (R_Trig_instance.Q=TRUE)THEN
  NXObject.Index       := UINT#16#4002;
  NXObject.Subindex  := USINT#1;
END_IF;

// NX_ReadObj의 실행
IF (Trigger=TRUE) THEN
  ReadCondition := TRUE;
END_IF;

IF ( (NX_ReadObj_instance.Done=TRUE) OR (NX_ReadObj_instance.Error=TRUE) ) THEN
  ReadCondition := FALSE;
END_IF;

ReadGo := ReadCondition & _EC_MBXSlavTbl[10];
NX_ReadObj_instance(
  Execute    := ReadGo,
  UnitProxy  := NXUnitProxy,
  Obj          := NXObject,
  TimeOut    := UINT#2000,
  ReadDat    := IoRefreshMethod);

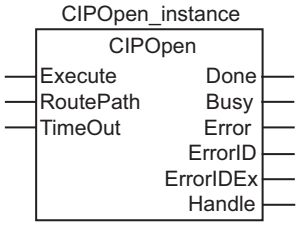
// 명령 실행 후의 처리
IF (NX_ReadObj_instance.Done=TRUE) THEN
  // 정상 종료 시 처리
  NormalEnd := NormalEnd + UINT#1;
ELSIF (NX_ReadObj_instance.Error=TRUE) THEN
  // 이상 종료 시 처리
  ErrorEnd := ErrorEnd + UINT#1;
END_IF;
```

EtherNet/IP 통신 명령

명령	명칭	페이지	명령	명칭	페이지
CIPOpen	CIPClass3(Large_Forward_Open) 연결 확립	2-878	SktUDPSend	UDP 소켓 송신	2-939
CIPOpenWithDataSize	CIPClass3 연결 확립 (크기 지정 있음)	2-887	SktTCPAccept	TCP 소켓 허용	2-942
CIPRead	변수 읽기 (Class3 Explicit 메시지)	2-890	SktTCPConnect	TCP 소켓 연결	2-945
CIPWrite	변수 쓰기 (Class3 Explicit 메시지)	2-895	SktTCPRcv	TCP 소켓 수신	2-953
CIPSend	임의 Explicit 메시지 송신 (Class3)	2-901	SktTCPSend	TCP 소켓 송신	2-956
CIPClose	CIP Class3 연결 중단	2-905	SktGetTCPStatus	TCP 소켓의 Status 읽기	2-959
CIPUCMMRead	변수 읽기 (UCMM Explicit 메시지)	2-907	SktClose	TCP/UDP 소켓 클로즈	2-962
CIPUCMMWrite	변수 쓰기 (UCMM Explicit 메시지)	2-912	SktClearBuf	TCP/UDP 소켓 수신 버퍼 클리어	2-965
CIPUCMMSend	임의 Explicit 메시지 송신 (UCMM)	2-919	ChangeIPAdr	IP 주소 변경	2-968
SktUDPCreate	UDP 소켓 생성	2-929	ChangeFTPAccount	FTP 계정 변경	2-977
SktUDPRcv	UDP 소켓 수신	2-936	ChangeNTPServerAdr	NTP 서버 수신처 변경	2-980

CIPOpen

지정한 상대 노드에 대해 CIP의 Class3(Large_Forward_Open) 연결을 확립합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPOpen	Class3(Large_ Forward_Open) 연결 확립	FB		CIPOpen_instance(Execute, RoutePath, TimeOut, Done, Busy, Error,} ErrorID, ErrorIDEx, Handle);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
RoutePath	라우팅 경로	입력	라우팅 경로	데이터 형식에 따름	–	–
TimeOut	타임아웃 시간		연결 타임아웃 시간	1~65535	0.1s	20 (2s)
Handle	핸들	출력	핸들	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
RoutePath																				○
TimeOut							○													
Handle	구조체 _sCIP_HANDLE에 대한 자세한 내용은 기능 설명 참조																			

기능

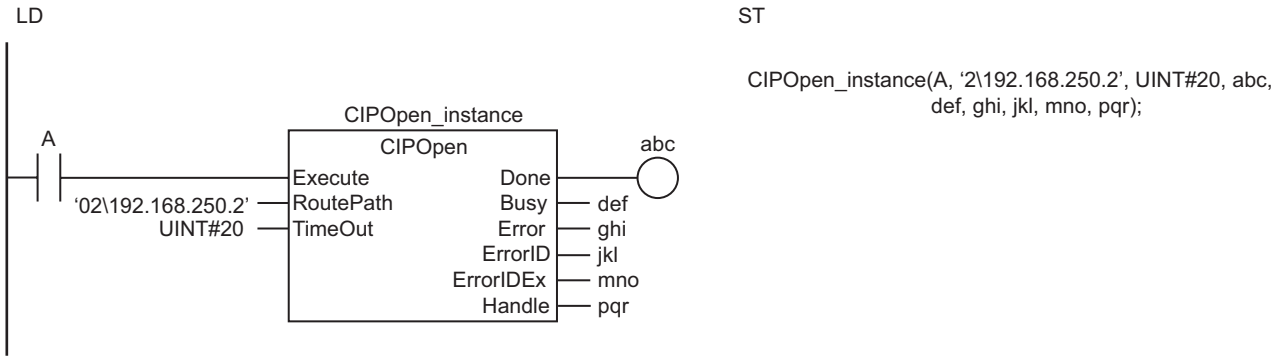
라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 상대 노드에 대해 CIP의 Class3(Large_ Forward_ Open) 연결을 확립합니다.
연결이 확립되면 핸들 'Handle'이 출력됩니다.

연결 타임아웃 시간은 'TimeOut'으로 지정합니다. CIPSend 명령, CIPWrite 명령, CIPRead 명령 중 하나를 실행한 후, 연결 타임아웃 시간 이내에 상대 노드에서 응답이 반환되지 않으면 통신 실패라고 판단됩니다. 연결 타임아웃 시간은 CIPRead 명령, CIPWrite 명령, CIPSend 명령이 실행하고 상대 노드에서 응답이 반환되면 리셋됩니다.

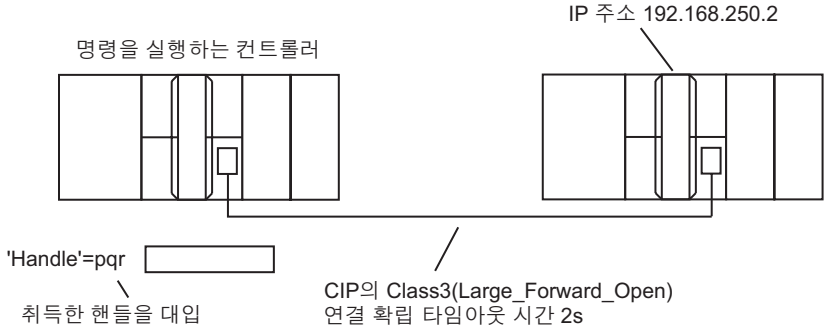
'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_ HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

'RoutePath'='02\192.168.250.2', 'TimeOut'=UINT#20인 경우의 기술 예가 아래에 나와 있습니다.
IP 주소 '192.168.250.2'의 상대 노드에 대해 CIP의 Class3(Large_Forward_Open) 연결을 확립합니다.
타임아웃 시간은 2s입니다. 변수 pqr에 핸들이 대입됩니다.



'RoutePath'로 지정한 CIP 네트워크상의 상대 노드에 대해
CIP의 Class3(Large_Forward_Open) 연결을 확립합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

- CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.
  - 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
  - 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"
- Forward Open으로 연결을 확립하는 경우는 🕮 "CIPOpenWithDataSize(2-887페이지)'을 사용하십시오.

버전 관련 정보

CIPOpenWithDataSize 명령은 CPU 유니트 Ver.1.06 이상, Sysmac Studio Ver.1.07 이상에서 사용할 수 있습니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 CIPRead 명령, CIPWrite 명령, CIPSend 명령 전에 실행해두어야 합니다.
- 이 명령으로 연결을 확립한 후, 최초의 타임아웃 시간은 'TimeOut'의 값이 100(10s) 이하라도 10s가 됩니다.
- 이 명령으로 확립한 연결은 CIPClose 명령을 사용해서 클로즈하십시오.
- 연결이 타임아웃이 되더라도 이 명령으로 생성한 핸들은 남습니다. 반드시 CIPClose 명령을 사용하여 연결을 클로즈하십시오.
- 이 명령으로 생성한 핸들은 프로그램 모드 변경 시 무효가 됩니다.
- 동시에 생성할 수 있는 핸들은 최대 32개입니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 이 명령에서는 'ErrorIDEx'를 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'TimeOut'의 값이 유효 범위 밖인 경우
  - 'RoutePath'의 문자열이 잘못된 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - CIPClass 연결의 리소스(32개)를 초과하여 연결을 개설하려고 한 경우
  - 연결 확립의 응답이 없는 경우
  - 연결을 확립할 상대 노드가 Large_Forward_Open에 대응하지 않는 경우
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - IP 중복 이상이 발생한 경우
  - TCP 연결이 모두 사용되고 있는 경우
  - BOOTP 서버 이상이 발생한 상태에서 이 명령이 실행된 경우

샘플 프로그램

CIP의 Class3 메시지를 사용하여 변수 쓰기, 변수 읽기, 메시지 송신을 수행합니다.
컨트롤러 간에는 EtherNet/IP 네트워크로 연결되며, 상대 노드의 IP 주소는 192.168.250.2로 합니다.
처리 순서는 다음과 같습니다.

1 **CIPOpen 명령을 사용하여 Class3(Large_Forward_Open) 연결을 확립합니다. 타임아웃 시간은 2s로 합니다.**

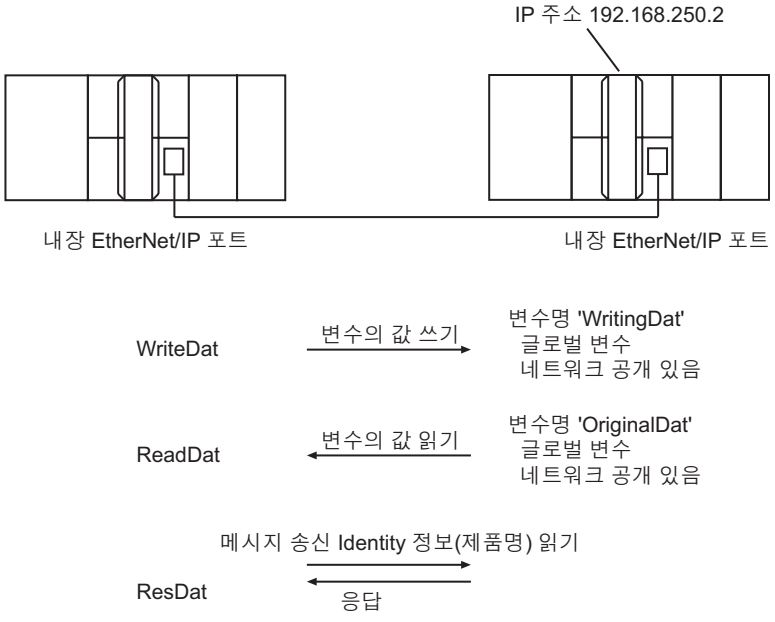
2 **CIPWrite 명령을 사용하여 상대 노드의 변수에 값을 씁니다. 타국 변수명은 'WritingDat', 기록할 값은 변수 WriteDat의 내용으로 합니다. 'WritingDat'은 상대 노드에서 글로벌 변수로 정의되어 있고 네트워크 공개 상태여야 합니다.**

3 **CIPRead 명령을 사용하여 상대 노드의 변수의 값을 읽습니다. 타국 변수명은 'OriginalDat', 읽은 값의 저장대상은 변수 ReadDat로 합니다. 'OriginalDat'은 상대 노드에서 글로벌 변수로 정의되어 있고 네트워크 공개 상태여야 합니다.**

4 **CIPSend 명령을 사용하여 상대 노드에 Explicit 메시지를 송신합니다. 메시지 내용은 Identity 정보(제품명) 읽기입니다. 그때의 클래스 ID, 인스턴스 ID, 속성 ID, 서비스 코드는 다음과 같습니다. 응답 데이터는 변수 ResDat에 저장합니다.**

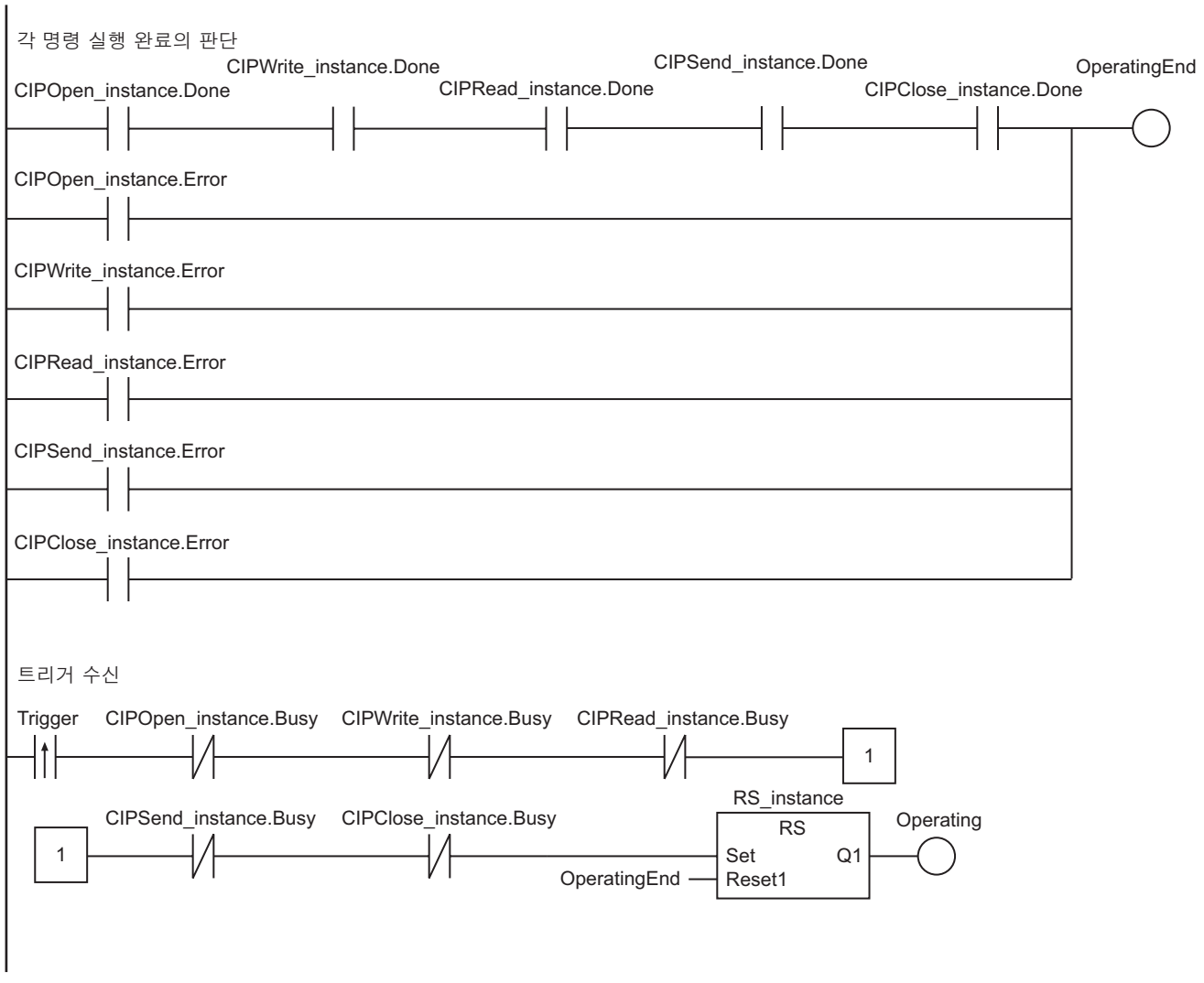
항목	값
클래스 ID	1
인스턴스 ID	1
속성 ID	7
서비스 코드	16#0E

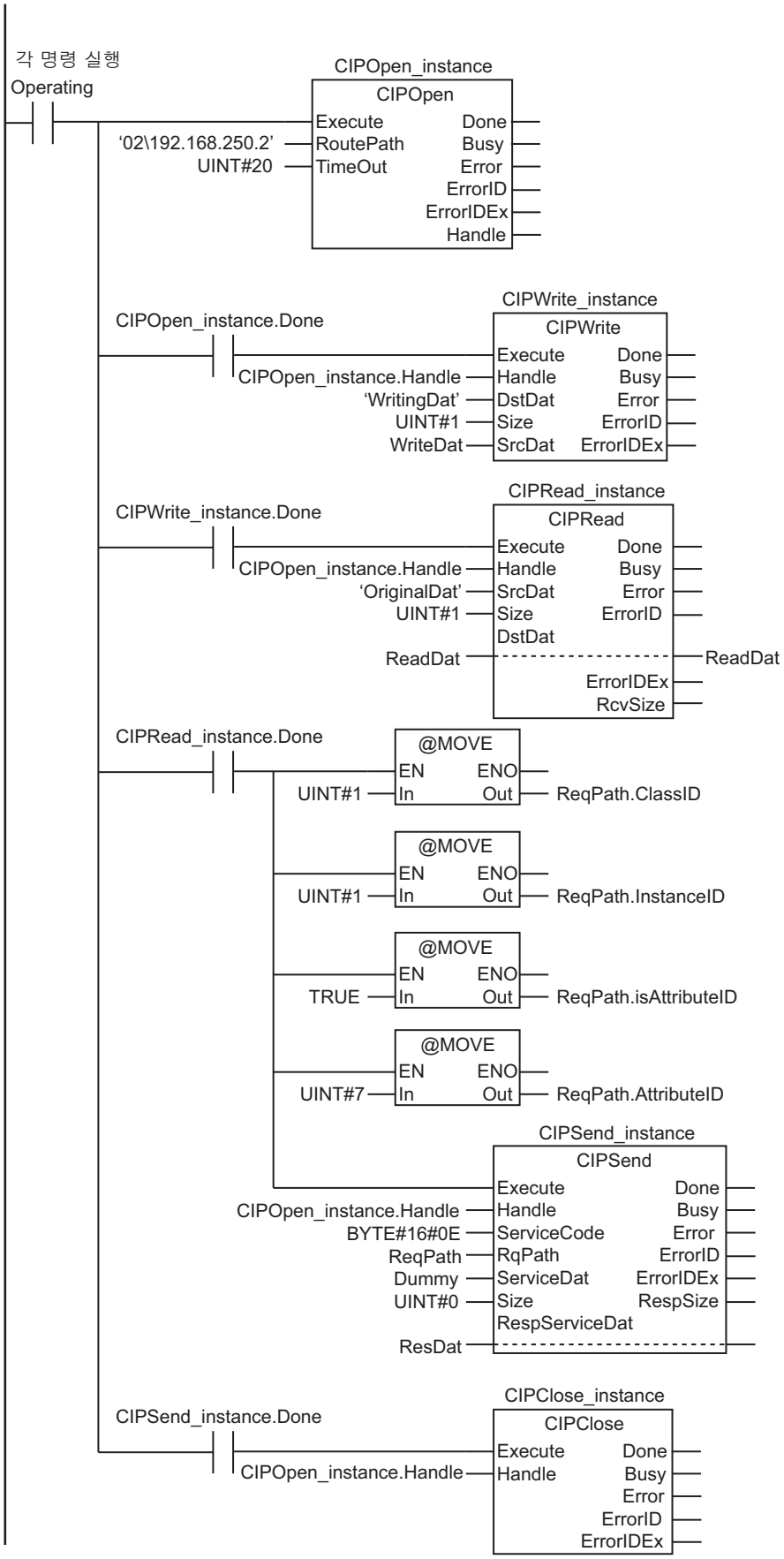
5 **CIPClose 명령을 사용하여 Class3 연결을 중단합니다.**

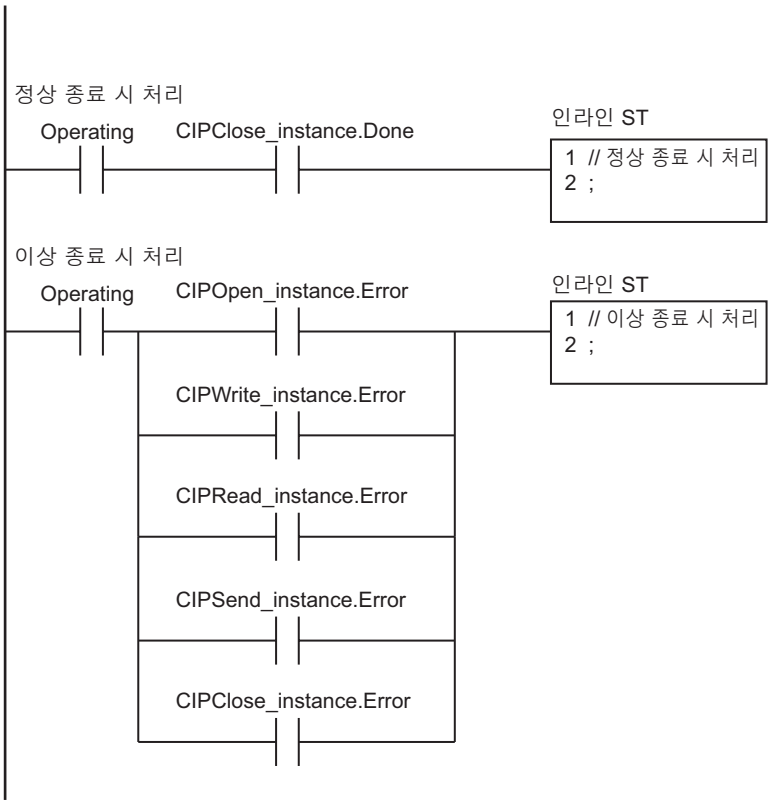


LD

명칭	데이터 형식	초기값	코멘트
OperatingEnd	BOOL	FALSE	처리 종료
Trigger	BOOL	FALSE	실행 조건
Operating	BOOL	FALSE	처리중
WriteDat	INT	1234	쓰기 데이터
ReadDat	INT	0	읽기 데이터
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=FALSE, AttributeID:=0)	요청 경로
ResDat	ARRAY[0..10] OF BYTE	[11(16#0)]	응답 데이터
Dummy	BYTE	16#0	더미
RS_instance	RS		
CIPOpen_instance	CIPOpen		
CIPWrite_instance	CIPWrite		
CIPRead_instance	CIPRead		
CIPSend_instance	CIPSend		
CIPClose_instance	CIPClose		




각 명령 실행
Operating
CIPOpen_instance
CIPOpen
Execute
RoutePath
TimeOut
Done
Busy
Error
ErrorID
ErrorIDEx
Handle
'02\192.168.250.2'
UINT#20
CIPWrite_instance
CIPWrite
CIPOpen_instance.Done
CIPOpen_instance.Handle
'WritingDat'
UINT#1
WriteDat
Execute
Handle
DstDat
Size
SrcDat
Done
Busy
Error
ErrorID
ErrorIDEx
CIPRead_instance
CIPRead
CIPWrite_instance.Done
CIPOpen_instance.Handle
'OriginalDat'
UINT#1
ReadDat
Execute
Handle
SrcDat
Size
DstDat
Done
Busy
Error
ErrorID
ErrorIDEx
RcvSize
ReadDat
CIPRead_instance.Done
@MOVE
EN
In
ENO
Out
UINT#1
ReqPath.ClassID
@MOVE
UINT#1
ReqPath.InstanceID
@MOVE
TRUE
ReqPath.isAttributeID
@MOVE
UINT#7
ReqPath.AttributeID
CIPSend_instance
CIPSend
CIPOpen_instance.Handle
BYTE#16#0E
ReqPath
Dummy
UINT#0
ResDat
Execute
Handle
ServiceCode
RqPath
ServiceDat
Size
RespServiceDat
Done
Busy
Error
ErrorID
ErrorIDEx
RespSize
CIPClose_instance
CIPClose
CIPSend_instance.Done
CIPOpen_instance.Handle
Execute
Handle
Done
Busy
Error
ErrorID
ErrorIDEx


정상 종료 시 처리
Operating
CIPClose_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
Operating
CIPOpen_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
CIPWrite_instance.Error
CIPRead_instance.Error
CIPSend_instance.Error
CIPClose_instance.Error

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoCIPTrigger	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	WriteDat	INT	0	쓰기 데이터
	ReadDat	INT	0	읽기 데이터
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=FALSE, AttributeID:=0)	요청 경로
	ResDat	ARRAY[0..10] OF BYTE	[11(16#0)]	응답 데이터
	Dummy	BYTE	16#0	더미
	CIPOpen_instance	CIPOpen		
	CIPWrite_instance	CIPWrite		
	CIPRead_instance	CIPRead		
	CIPSend_instance	CIPSend		
	CIPClose_instance	CIPClose		

외부 변수	명칭	상수	데이터 형식	코멘트
	_EIP_EtnOnlineSta	☑	BOOL	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoCIPTrigger=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoCIPTrigger:=TRUE;
    Stage          :=INT#1;
    CIPOpen_instance(Execute:=FALSE);      // 인스턴스 초기화
    CIPWrite_instance(
        Execute          :=FALSE,          // 인스턴스 초기화
        SrcDat           :=WriteDat);      // 더미
    CIPRead_instance(                      // 인스턴스 초기화
        Execute          :=FALSE,          // 더미
        DstDat           :=ReadDat);       // 더미
    CIPSend_instance(
        Execute          :=FALSE,          // 인스턴스 초기화
        ServiceDat       := Dummy,         // 더미
        RespServiceDat:=ResDat);           // 더미
    CIPClose_instance(Execute:=FALSE);     // 인스턴스 초기화
END_IF;


IF (DoCIPTrigger=TRUE) THEN
    CASE Stage OF
    1 :                                    // CIP Class3(Large_Forward_Open) 연결 확립
        CIPOpen_instance(
            Execute    :=TRUE,
            TimeOut    :=UINT#20,          // 타임아웃 시간 2.0s
            RoutePath:='02\192.168.250.2');// 라우팅 경로

        IF (CIPOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;                  // 정상 종료
        ELSIF (CIPOpen_instance.Error=TRUE) THEN
            Stage:=INT#10;                 // 이상 종료
        END_IF;

    2 :                                    // 변수 쓰기 요구
        CIPWrite_instance(
            Execute:=TRUE,
            Handle   :=CIPOpen_instance.Handle,  // 핸들
            DstDat   :='WritingDat',             // 타국 변수명
            Size     :=UINT#1,                   // 쓰기 요소 수
            SrcDat   :=WriteDat);                // 쓰기 데이터
```

```
        IF (CIPWrite_instance.Done=TRUE) THEN
            Stage:=INT#3;                          // 정상 종료
        ELSIF (CIPWrite_instance.Error=TRUE) THEN
            Stage:=INT#20;                         // 이상 종료
        END_IF;

    3 :                                    // 변수 읽기 요구
        CIPRead_instance(
            Execute:=TRUE,
            Handle  :=CIPOpen_instance.Handle,     // 핸들
            SrcDat  :='OriginalDat',               // 타국 변수명
            Size    :=UINT#1,                      // 읽기 요소 수
            DstDat  :=ReadDat);                    // 읽기 데이터

        IF (CIPRead_instance.Done=TRUE) THEN
            Stage:=INT#4;                          // 정상 종료
        ELSIF (CIPRead_instance.Error=TRUE) THEN
            Stage:=INT#30;                         // 이상 종료
        END_IF;

    4 :                                    // 임의 메시지 송신
        ReqPath.ClassID      :=UINT#01;
        ReqPath.InstanceID   :=UINT#01;
        ReqPath.isAttributeID:=TRUE;
        ReqPath.AttributeID  :=UINT#07;
        CIPSend_instance(
            Execute          :=TRUE,
            Handle           :=CIPOpen_instance.Handle,  // 핸들
            ServiceCode      :=BYTE#16#0E,               // 서비스 코드
            RqPath           :=ReqPath,                  // 요청 경로
            ServiceDat       :=Dummy,                    // 서비스 데이터
            Size             :=UINT#0,                   // 요소 수
            RespServiceDat:=ResDat);                     // 응답 데이터

        IF (CIPSend_instance.Done=TRUE) THEN
            Stage:=INT#5;                          // 정상 종료
        ELSIF (CIPSend_instance.Error=TRUE) THEN
            Stage:=INT#40;                         // 이상 종료
        END_IF;

    5 :                                    // CIP Class3 연결 중단 요구
       CIPClose_instance(
            Execute :=TRUE,
            Handle  :=CIPOpen_instance.Handle);      // 핸들

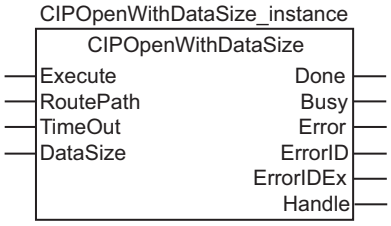
        IF (CIPClose_instance.Done=TRUE) THEN
            Stage:=INT#0;
        ELSIF (CIPClose_instance.Error=TRUE) THEN
            Stage:=INT#50;
        END_IF;

    0:                                 // 정상 종료 처리
        DoCIPTrigger:=FALSE;
        Trigger      :=FALSE;

    ELSE                               // 이상 종료 처리
        DoCIPTrigger:=FALSE;
        Trigger      :=FALSE;
    END_CASE;
END_IF;
```

CIPOpenWithDataSize

지정된 상대 노드에 대해, 지정된 데이터 길이 이하의 Class3 Explicit 메시지를 송수신할 수 있는 CIP Class3 연결을 확립합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPOpenWithDataSize	CIP Class3 연결 확립(크기 지정 있음)	FB		CIPOpen_instance(Execute, RoutePath, TimeOut, DataSize, Done, Busy, Error, ErrorID, ErrorIDEx, Handle);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
RoutePath	라우팅 경로	입력	라우팅 경로	데이터 형식에 따름	–	–
TimeOut	타임아웃 시간		타임아웃 시간	1~65535	0.1s	20 (2s)
DataSize	데이터 길이		데이터 길이	10~1994	바이트	1994
Handle	핸들	출력	핸들	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
RoutePath																				○
TimeOut							○													
DataSize							○													
Handle	구조체 _sCIP_HANDLE 자세한 내용은 기능 설명 참조																			

기능

라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 상대 노드에 대해 CIP의 Class3 연결을 확립합니다. 송수신할 Class3 Explicit 메시지의 데이터 길이를 데이터 길이 'DataSize'로 지정합니다.
Class3 연결 서비스는 'DataSize'의 값에 따라 다음과 같이 결정됩니다.

'DataSize'의 값[바이트]	서비스
509 이하	Forward_Open
510~1994	Large_Forward_Open

연결이 확립되면 핸들 'Handle'이 출력됩니다.

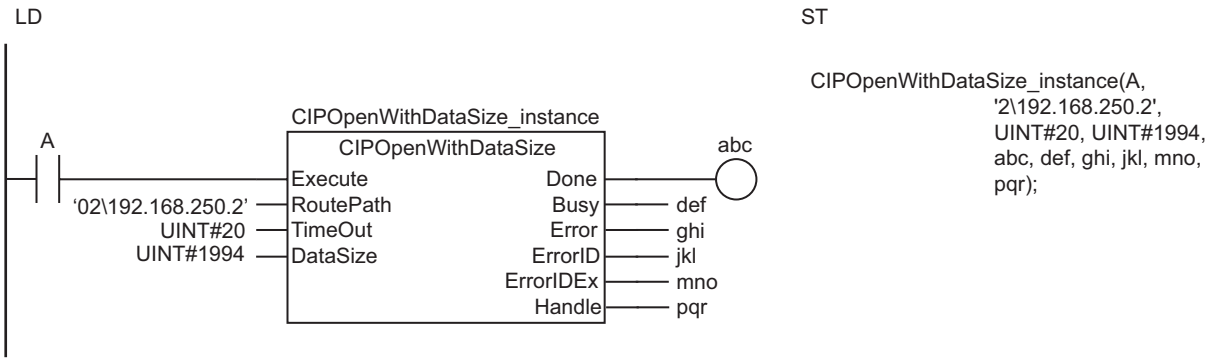
연결 타임아웃 시간은 'TimeOut'으로 지정합니다. CIPSend 명령, CIPWrite 명령, CIPRead 명령 중 하나를 실행한 후, 연결 타임아웃 시간 이내에 상대 노드에서 응답이 반환되지 않으면 통신 실패라고 판단됩

니다. 연결 타임아웃 시간은 CIPRead 명령, CIPWrite 명령, CIPSend 명령이 실행하고 상대 노드에서 응답이 반환되면 리셋됩니다.

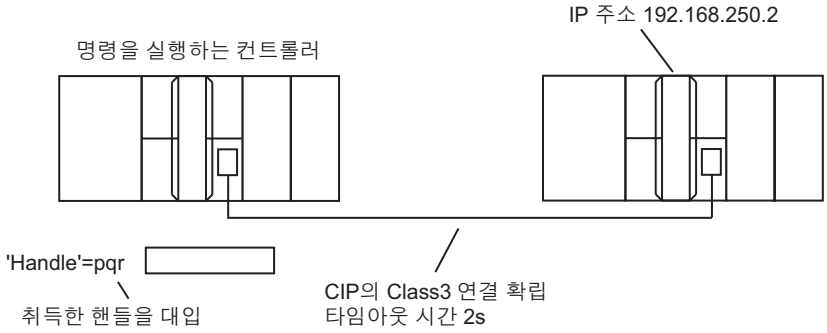
'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

'RoutePath'='02\192.168.250.2', 'TimeOut'=UINT#20인 경우의 기술 예가 아래에 나와 있습니다.
IP 주소 '192.168.250.2'의 상대 노드에 대해 CIP의 Class3 연결을 확립합니다. 데이터 길이는 1994바이트, 타임아웃 시간은 2s입니다. 변수 pqr에 핸들이 대입됩니다.



'RoutePath'로 지정한 CIP 네트워크상의 상대 노드에 대해 CIP의 Class3 연결을 확립합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE: 사용 가능 FALSE: 사용 불가능

참고

- CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.
  - 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
  - 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"
- Class3 연결 서비스가 Large_Forward_Open인 경우는 🕮 'CIPOpen(2-878페이지)'를 사용할 수도 있습니다.

사용상의 주의 사항

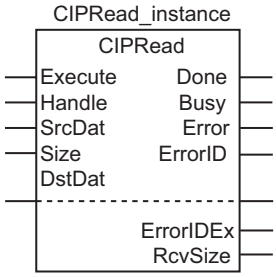
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 🕮 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령 또는 CIPOpen 명령을 CIPRead 명령, CIPWrite 명령, CIPSend 명령 전에 실행해야 합니다.
- 이 명령으로 연결을 확립한 후 최초의 타임아웃 시간은 'TimeOut'의 값이 100(10s) 이하라도 10s가 됩니다.
- 이 명령으로 확립한 연결은 CIPClose 명령을 사용하여 클로즈하십시오.
- 연결이 타임아웃되어도 이 명령으로 생성한 핸들은 남습니다. 반드시 CIPClose 명령을 사용하여 연결을 클로즈하십시오.
- 이 명령으로 생성된 핸들은 프로그램 모드 변경 시 무효가 됩니다.
- 동시에 생성할 수 있는 핸들은 최대 32개입니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 이 명령에서는 'ErrorIDEx'를 사용할 수 없습니다.
- 다음과 같은 경우, 이상이 발생하며, 'Error'는 TRUE가 됩니다.
  - 'TimeOut'의 값이 유효 범위 밖인 경우
  - 'RoutePath'의 문자열이 잘못된 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - CIPClass 연결의 리소스(32개)를 초과하여 연결을 개설하려고 한 경우
  - 연결 확립의 응답이 없는 경우
  - 'DataSize'의 값이 510 ~ 1994이고, 연결을 확립할 상대 노드가 Large_Forward_Open에 대응하지 않는 경우
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - IP 중복 이상이 발생한 경우
  - TCP 연결이 모두 사용되고 있는 경우
  - BOOTP 서버 이상이 발생한 상태에서 이 명령이 실행된 경우

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상, Sysmac Studio Ver.1.07 이상에서 사용할 수 있습니다.

CIPRead

CIP 네트워크상의 타국 컨트롤러의 변수의 값을 읽어옵니다(Class3 Explicit 메시지).

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPRead	변수 읽기 (Class3 Explicit 메시지)	FB		CIPRead_instance(Execute, Handle, SrcDat, Size, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx, RcvSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Handle	핸들	입력	CIPOpen 또는 CIPOpenWithDataSize 명령으로 취득한 핸들	–	–	–
SrcDat	타국 변수명		읽을 타국 컨트롤러의 변수명	데이터 형식에 따름		"
Size	읽기 요소 수		읽을 요소 수	0~1988		1
DstDat	읽기 데이터	입출력	읽은 데이터의 값	데이터 형식에 따름	–	–
RcvSize	읽기 데이터 크기	출력	읽은 데이터의 크기	0~1988	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Handle	구조체 _sCIP_HANDLE에 대한 자세한 내용은 기능 설명 참조																			
SrcDat																				○
Size							○													
DstDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열, 구조체 전체, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능(*)																			
RcvSize							○													

* STRING 형식 배열은 지정할 수 없습니다.

기능

'Handle'로 지정한 CIP 네트워크상의 타국 컨트롤러 내의 타국 변수명 'SrcDat'으로 지정한 네트워크 변수의 값을 읽습니다.
읽은 값은 읽기 데이터 'DstDat'에 저장합니다.
읽을 요소 수는 'Size'로 지정합니다. 'SrcDat'가 배열인 경우는 읽을 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 'Size'의 값이 0인 경우, 'SrcDat'가 배열인지의 여부에 관계없이 읽기를 수행하지 않습니다.

읽기 완료 후, 읽기 데이터 크기 'RcvSize'에 읽은 데이터의 크기가 바이트 단위로 대입됩니다. 읽을 수 있는 데이터의 최대 크기는 연결 확립을 CIPOpen 명령으로 수행했는지, CIPOpenWithDataSize 명령으로 수행했는지에 따라 다음과 같이 달라집니다.

연결 확립을 수행한 명령	읽을 수 있는 데이터의 최대 크기[바이트]
CIPOpen	1988
CIPOpenWithDataSize	CIPOpenWithDataSize 'DataSize'-6

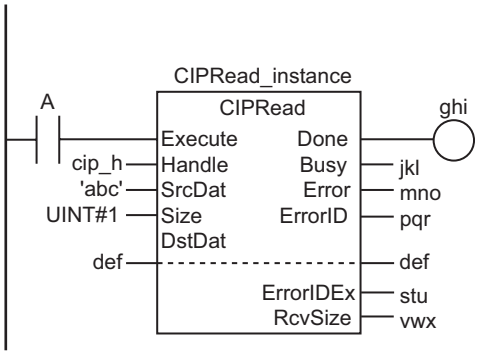
'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다.

타국의 변수 abc의 값을 읽어 자국의 변수 def에 저장하는 경우의 예가 아래에 나와 있습니다. 읽기 요소 수 'Size'의 값은 UINT#1입니다. abc, def의 데이터 형식은 SINT입니다. SINT 형식의 데이터 크기는 1바이트이므로 읽기 데이터 크기 vwx의 값은 UINT#1이 됩니다.

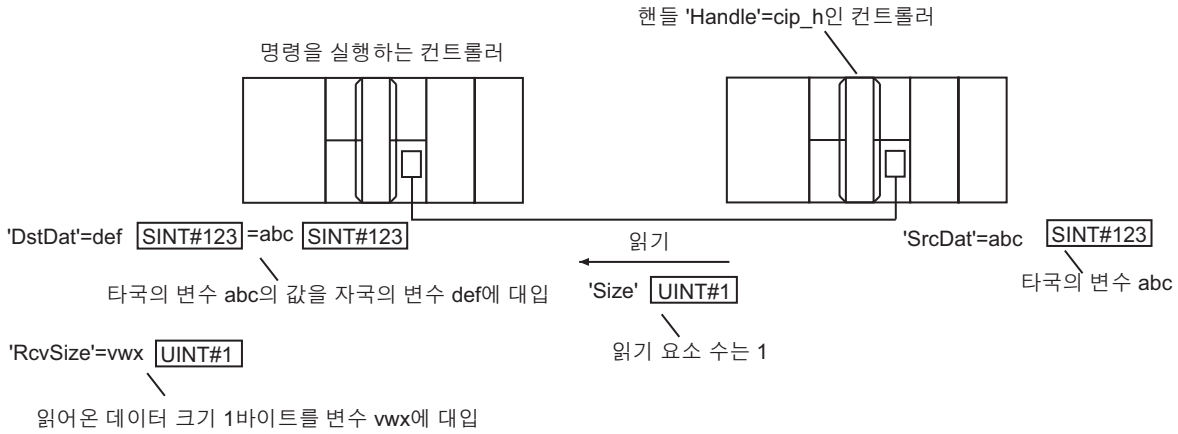
LD



ST

```
CIPRead_instance(A, cip_h, 'abc', UINT#1,
                 def, ghi, jkl, mno, pqr,
                 stu, vwx);
```

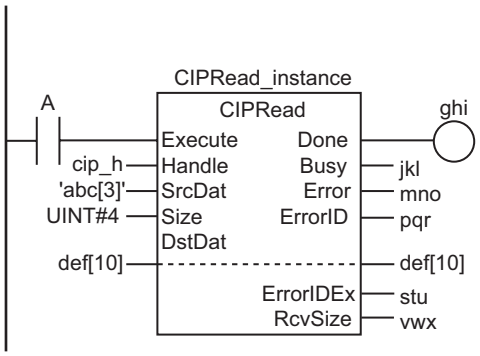
핸들 'Handle'로 지정한 CIP 네트워크상의 타국의 변수 'SrcDat'의 값을 자국의 변수 'DstDat'에 대입합니다.
읽기 요소 수는 'Size'로 지정합니다. 읽은 데이터 크기는 'RcvSize'에 대입됩니다.



배열을 읽는 경우

읽을 데이터가 배열인 경우, 'SrcDat'에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다. 또한 'DstDat'에도 첨자가 있는 배열의 요소를 파라메터로 전달합니다.
타국의 배열 변수 abc[3]~abc[6]의 4개 요소를 읽어 자국의 배열 변수 def[10]~def[13]에 저장하는 경우의 예가 아래에 나와 있습니다. abc, def의 데이터 형식은 INT입니다. INT 형식의 데이터 크기는 2바이트이므로 읽기 데이터 크기 vwx의 값은 UINT#8이 됩니다.

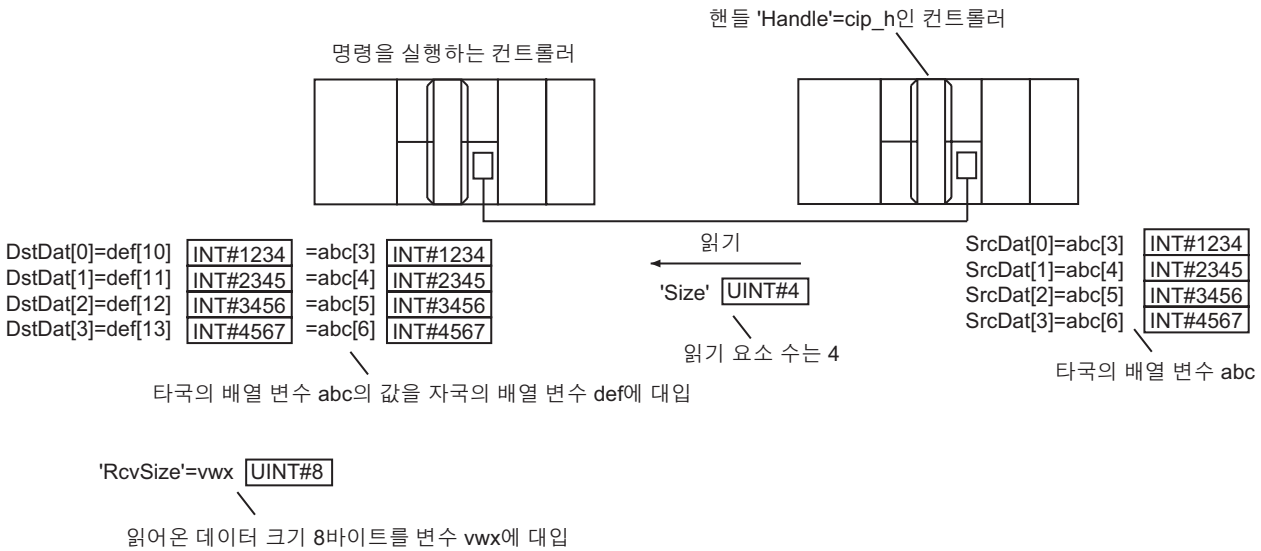
LD



ST

```
CIPRead_instance(A, cip_h, 'abc[3]', UINT#4,
                 def[10], ghi, jkl, mno, pqr,
                 stu, vwx);
```

타국의 배열 변수 abc[3]~abc[6]의 값을 자국의 배열 변수 def[10]~def[13]에 대입합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 📖 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 반드시 CIPOpen 명령 또는 CIPOpenWithDataSize 명령을 실행하여 'Handle'을 취득하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결된 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 읽을 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- CJ 유니트용 메모리의 주소를 직접 지정하여 데이터를 읽을 수는 없습니다. CJ 유니트용 메모리의 특정 주소를 읽는 경우, 사전에 그 메모리 주소를 AT 지정을 통해 변수에 할당해두십시오.
- 자국 측의 CJ 유니트용 메모리 주소를 직접 지정하여 데이터를 저장할 수도 없습니다. CJ 유니트용 메모리의 특정 주소에 저장하는 경우는 사전에 그 메모리 주소를 AT 지정을 통해 'DstDat'에 할당해두십시오.
- 'SrcDat'로 사용할 수 있는 문자의 조건은 다음과 같습니다.

항목	사양
최대 바이트 수	127바이트
문자 코드	UTF-8
사용 가능한 문자	영숫자(대문자와 소문자의 구별 없음), 반각 가타카나, 멀티 바이트 문자, '_'(밑줄)
조합하여 사용할 수 없는 문자	• ASCII 코드의 숫자 0~9(문자 코드 16#30~16#39)로 시작되는 문자열 • ASCII 코드의 '_'(밑줄)이 1문자만 있는 문자열 • ASCII 코드의 '_'(밑줄)이 2개 이상 연속으로 포함된 문자열 • 첫 번째 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 마지막 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 처음 2개의 문자가 'P_'인 문자열

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'SrcDat'의 문자열이 잘못된 경우
  - 읽은 값의 데이터 형식과 'DstDat'의 데이터 형식이 일치하지 않는 경우
  - 읽은 값의 데이터 크기가 'DstDat'의 영역을 초과하는 경우
  - 'DstDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'Handle.Handle'의 값이 유효 범위 밖인 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 생성한 연결이 타임아웃된 경우
  - 'SrcDat'의 크기가 (CIPOpenWithDataSize 명령의 'DataSize'-6바이트)를 초과한 경우

- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 CIP 메시지 이상 코드의 의미를 갖습니다. 그 내용은 다음과 같습니다.

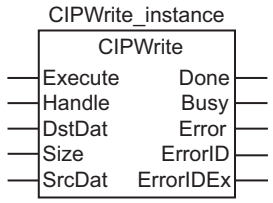
값	이상 내용
16#02000000	통신대상 노드가 고부하 상태이므로 정상적인 통신을 수행할 수 없습니다.
16#04000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 기본 데이터 형식 • 열거 형식 전체 • 구조체 전체 • 공용체 전체 • 배열 전체
16#05000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 열거 형식의 열거자 • 구조체의 1개 멤버 • 공용체의 1개 멤버 • 배열의 요소
16#0C008010	지정된 타국 변수를 다운로드 중입니다.
16#0C008011	
16#11000000	'Size'의 값이 현재 읽기 가능한 데이터 크기를 초과했습니다.
16#20008017	지정된 타국 변수가 배열이 아닌데도 읽기 요소 수가 1 이외입니다.
16#20008018	지정된 타국 변수가 배열이고, 읽기 요소 수가 그 배열의 요소 수를 초과합니다.
16#26000000	지정된 타국 변수가 NULL문자뿐입니다.

샘플 프로그램

🕮 'CIPOpen 명령(2-878페이지)'의 샘플 프로그램을 참조하십시오.

CIPWrite

CIP 네트워크상의 타국 컨트롤러의 변수에 값을 씁니다(Class3 Explicit 메시지).

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPWrite	변수 쓰기 (Class3 Explicit 메시지)	FB		CIPWrite_instance(Execute, Handle, DstDat, Size, SrcDat, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Handle	핸들	입력	CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 취득한 핸들	–	–	–
DstDat	타국 변수명		쓸 타국 컨트롤러의 변수명	데이터 형식에 따름		''
Size	쓰기 요소 수		쓸 요소 수	0~1980		1
SrcDat	쓰기 데이터		쓸 데이터의 값	데이터 형식에 따름		*1

*1. 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Handle	구조체 _sCIP_HANDLE에 대한 자세한 내용은 기능 설명 참조																			
DstDat																				○
Size							○													
SrcDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열*1, 구조체 전체, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			

*1. STRING 형식 배열은 지정할 수 없습니다.

기능

'Handle'로 지정한 CIP 네트워크상의 타국 컨트롤러 내의 타국 변수명 'DstDat'로 지정한 네트워크 변수에 값을 씁니다.
기록할 값은 쓰기 데이터 'SrcDat'의 내용입니다.
쓸 요소 수는 'Size'로 지정합니다. 'DstDat'가 배열인 경우는 쓸 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 'Size'의 값이 0인 경우, 'DstDat'가 배열인지의 여부에 관계없이 쓰기를 수행하지 않습니다.

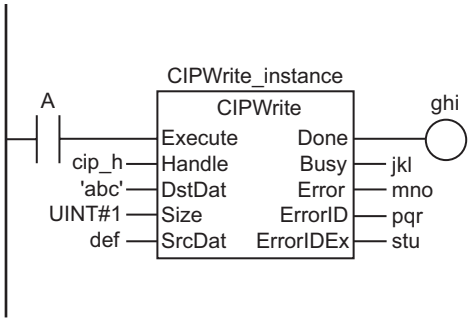
'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다.

타국의 변수 abc에 자국의 변수 def의 값을 쓰는 경우의 예가 아래에 나와 있습니다. 쓰기 요소 수 'Size'의 값은 UINT#1입니다.

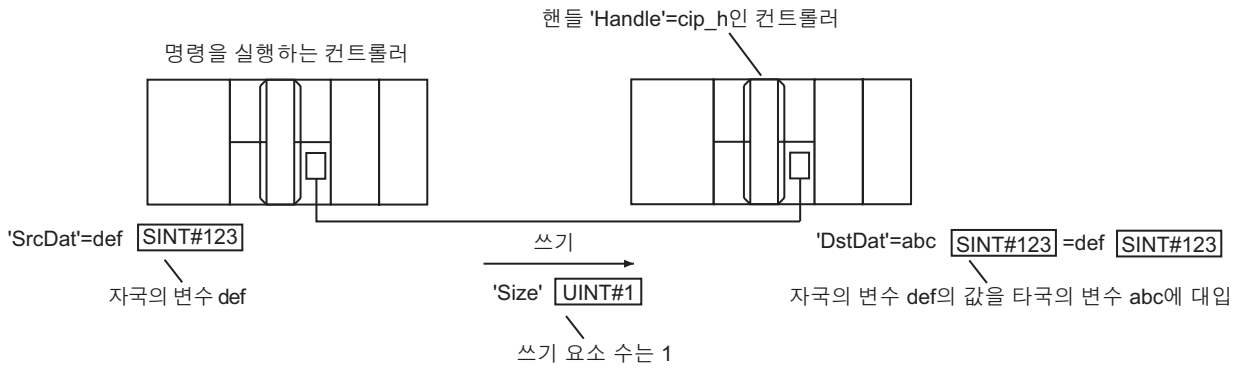
LD



ST

```
CIPWrite_instance(A, cip_h, 'abc', UINT#1,
                  def, ghi, jkl, mno, pqr, stu);
```

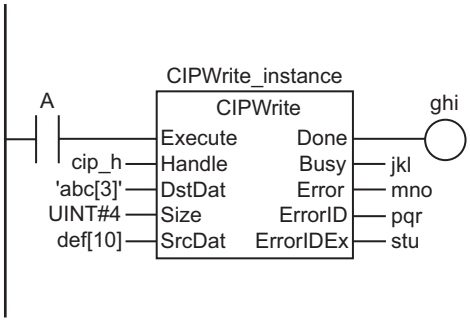
핸들 'Handle'로 지정한 CIP 네트워크상의 타국의 변수 'DstDat'에 자국의 변수 'SrcDat'의 값을 대입합니다. 쓰기 요소 수는 'Size'로 지정합니다.



배열에 쓰는 경우

쓸 데이터가 배열인 경우, 'DstDat'에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다. 또한 'SrcDat'에도 첨자가 있는 배열의 요소를 파라메터로 전달합니다.
타국의 배열 변수 abc[3]~abc[6]의 4개 요소에 자국의 배열 변수 def[10]~def[13]의 값을 쓰는 경우의 예가 아래에 나와 있습니다.

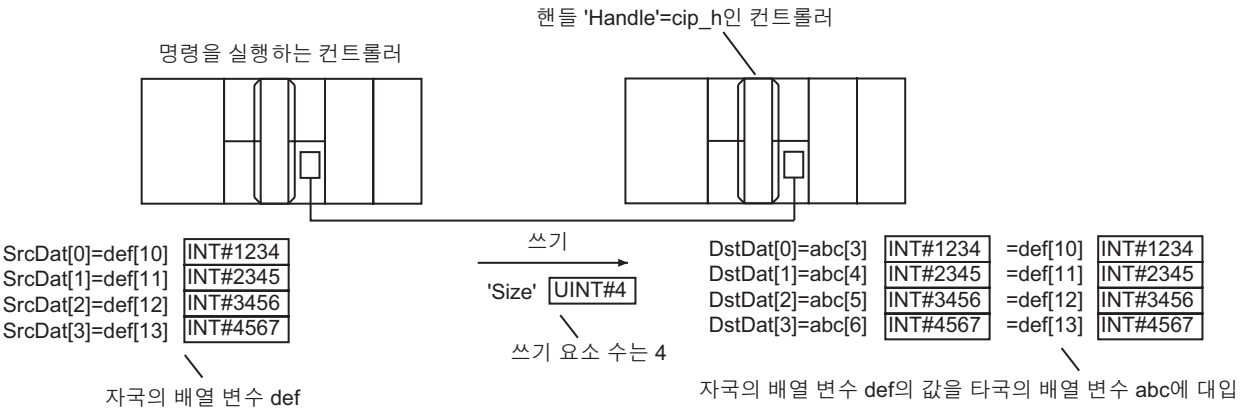
LD



ST

```
CIPWrite_instance(A, cip_h, 'abc[3]', UINT#4, def,
                  ghi[10], jkl, mno, pqr, stu);
```

타국의 배열 변수 abc[3]~abc[6]에 자국의 배열 변수 def[10]~def[13]의 값을 저장합니다.



쓸 수 있는 데이터 크기

쓸 수 있는 데이터의 최대 크기는 'DstDat'로 지정한 변수의 데이터 형식 및 변수명에 따라 다음과 같이 달라집니다.

쓸 수 있는 데이터 크기[바이트]=기본 크기-'DstDat'의 변수명 크기

위 식의 항목	의미
기본 크기	CIPOpen 명령으로 연결 확립을 수행한 경우 • 'DstDat'로 지정한 변수의 데이터 형식이 구조체 또는 STRING 형식인 경우, 1986바이트입니다. • 상기 이외의 데이터 형식인 경우, 1988바이트입니다. CIPOpenWithDataSize 명령으로 연결 확립을 수행한 경우 • 'DstDat'에 지정된 변수의 데이터 형식이 구조체 또는 STRING 형식인 경우, 다음 식으로 구할 수 있습니다. 기본 크기[바이트]=CIPOpenWithDataSize 명령의 'DataSize'-8 • 상기 이외의 데이터 형식인 경우, 다음 식으로 구할 수 있습니다. 기본 크기[바이트]=CIPOpenWithDataSize 명령의 'DataSize'-6
'DstDat'의 변수명 크기	• 변수명의, 구조체의 각 계층의 ASCII 코드 바이트 수의 합계+(계층 수*2)로 계산합니다. • 각 계층의 문자열의 ASCII 코드 바이트 수가 홀수인 경우, 거기에 1을 더합니다. • 어떤 계층이 배열인 경우에는 다시(배열의 차원 수*4)를 더합니다. • 구조체의 계층 구분을 나타내는 '.', 배열의 '[', ']', ', '는 변수명 크기에 포함되지 않습니다. (예 1) 'DstDat'의 변수명이 'aaa.bbbbb[1, 2, 3].cc'인 경우 • 제1계층 aaa의 문자열은 3바이트입니다. 홀수이므로 1을 더해서 4바이트로 만듭니다. • 제2계층 bbbbb[1, 2, 3]의 bbbbb의 문자열은 5바이트입니다. 홀수이므로 1을 더해 서 6바이트로 만듭니다. • 그리고 제2계층 bbbbb[1, 2, 3]은 3차원 배열이므로 3*4=12를 더해서 18바이트로 만듭니다. • 제3계층 cc의 문자열은 2바이트입니다. 짝수이므로 2바이트 그대로 사용합니다. • 제1계층 4바이트, 제2계층 18바이트, 제3계층 2바이트에 계층 수 3*2=6을 더해서 변수명 크기는 30바이트가 됩니다. (예 2) 'DstDat'의 변수명이 'val'인 경우 • 제1계층 val의 문자열은 3바이트입니다. 홀수이므로 1을 더해서 4바이트로 만듭니다. • 거기에 계층 수 1*2=2를 더해서 변수명 크기는 6바이트가 됩니다. (예 3) 'DstDat'의 변수명이 'array[8]'인 경우 • 제1계층 array의 문자열은 5바이트입니다. 홀수이므로 1을 더해서 6바이트로 만듭니다. • 배열의 차원 수는 1입니다. 따라서 1*4=4를 더합니다. • 거기에 계층 수 1*2=2를 더해서 변수명 크기는 12바이트가 됩니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 🕮 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 반드시 CIPOpen 명령 또는 CIPOpenWithDataSize 명령을 실행하여 'Handle'을 취득하십시오.
- 'SrcDat'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'SrcDat'가 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 쓸 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- CJ 유니트용 메모리의 주소를 직접 지정하여 데이터를 쓸 수는 없습니다. CJ 유니트용 메모리의 특정 주소에 쓰는 경우, 사전에 그 메모리 주소를 AT 지정을 통해 변수에 할당해두십시오.
- 자국 측의 CJ 유니트용 메모리 주소를 직접 지정할 수도 없습니다. CJ 유니트용 메모리의 특정 주소의 값을 쓰는 경우는 사전에 그 메모리 주소를 AT 지정을 통해 'SrcDat'에 할당해두십시오.
- 'DstDat'로 사용할 수 있는 문자의 조건은 다음과 같습니다.

항목	사양
최대 바이트 수	127바이트
문자 코드	UTF-8
사용 가능한 문자	영숫자(대문자와 소문자의 구별 없음), 반각 가타카나, 멀티 바이트 문자, '_'(밑줄)
조합하여 사용할 수 없는 문자	• ASCII 코드의 숫자 0~9(문자 코드 16#30~16#39)로 시작되는 문자열 • ASCII 코드의 '_'(밑줄) 이 1문자만 있는 문자열 • ASCII 코드의 '_'(밑줄)이 2개 이상 연속으로 포함된 문자열 • 첫 번째 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 마지막 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 처음 2개의 문자가 'P_'인 문자열

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'DstDat'의 문자열이 잘못된 경우
  - 'Size'의 값이 'SrcDat'의 영역을 초과하는 경우
  - 'SrcDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'Handle.Handle'의 값이 유효 범위 밖인 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 생성한 연결이 타임아웃된 경우
  - 'DstDat'의 크기와 'SrcDat'의 값의 합계가 (CIPOpenWithDataSize 명령의 'DataSize'-6바이트)를 초과한 경우
- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 CIP 메시지 이상 코드의 의미를 갖습니다. 그 내용은 다음과 같습니다.

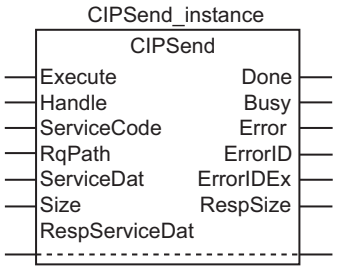
값	이상 내용
16#02000000	통신대상 노드가 고부하 상태이므로 정상적인 통신을 수행할 수 없습니다.
16#04000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 기본 데이터 형식 • 열거 형식 전체 • 구조체 전체 • 공용체 전체 • 배열 전체
16#05000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 열거 형식의 열거자 • 구조체의 1개 멤버 • 공용체의 1개 멤버 • 배열의 요소
16#0C008010 16#0C008011	지정된 타국 변수를 다운로드 중입니다.
16#1F000102	• 지정된 타국 변수에 상수 속성이 설정되어 있으므로 쓸 수 없습니다. • 쓰기 데이터와 쓰기 요소 수가 맞지 않습니다.
16#20008017	지정된 타국 변수가 배열이 아닌데도 쓰기 요소 수가 1 이외입니다.
16#20008018	지정된 타국 변수가 배열이고, 쓰기 요소 수가 그 배열의 요소 수를 초과합니다.
16#20008022	서버측 변수의 데이터 형식과 클라이언트측 변수의 데이터 형식이 일치하지 않습니다.
16#20008028	• 지정된 타국 변수가 열거 형식이고, 쓰기 데이터가 열거자에 없는 값입니다. • 지정된 타국 변수가 범위 형식 지정 속성을 갖고 있으며 쓰기 데이터가 범위 밖의 값입니다.
16#26000000	지정된 타국 변수명이 NULL문자뿐니다.

샘플 프로그램

'CIPOpen 명령(2-878페이지)'의 샘플 프로그램을 참조하십시오.

CIPSend

CIP 네트워크상의 지정된 장치에 Class3의 CIP 메시지를 송신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPSend	임의 Explicit 메시지 송신 (Class3)	FB		CIPSend_instance(Execute, Handle, ServiceCode, RqPath, ServiceDat, Size, RespServiceDat, Done, Busy, Error, ErrorID, ErrorIDEx, RespSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Handle	핸들	입력	CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 취득한 핸들	–	–	–
Service Code	서비스 코드		서비스 코드	데이터 형식에 따름		
RqPath	요청 경로		요청 경로(클래스 ID, 인스턴스 ID, 속성 ID)	–		
ServiceDat	서비스 데이터		송신할 서비스 데이터	데이터 형식에 따름		(*)
Size	송신 요소 수		송신할 요소 수			1
Resp ServiceDat	응답 데이터	입출력	응답 데이터	데이터 형식에 따름	–	–
RespSize	응답 데이터 크기	출력	응답 데이터의 크기	데이터 형식에 따름	바이트	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Handle	구조체 _sCIP_HANDLE에 대한 자세한 내용은 기능 설명 참조																			
Service Code		○																		
RqPath	구조체 _sREQUEST_PATH에 대한 자세한 내용은 기능 설명 참조																			
ServiceDat		○	○	○	○	○	○	○	○	○	○	○	○	○	○					
	배열, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			
Size							○													
Resp ServiceDat		○	○	○	○	○	○	○	○	○	○	○	○	○	○					
	배열, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			
RespSize							○													

기능

서비스 코드 'ServiceCode'로 지정한 서비스에 대해 서비스 데이터 'ServiceDat'를 Class3 Explicit 메시지로 송신합니다.
송신대상은 핸들 'Handle'로 지정합니다.
요청 경로는 'RqPath'로 지정합니다.

송신할 요소 수는 'Size'로 지정합니다. 'ServiceDat'가 배열인 경우는 송신할 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 서비스 데이터가 필요 없는 경우는 'Size'의 값을 0으로 합니다.

송신 후, 응답 데이터가 'RespServiceDat'에 저장됩니다. 또한 응답 데이터의 크기가 바이트 단위로 'RespSize'에 저장됩니다.

'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

'RqPath'의 데이터 형식은 구조체 _sREQUEST_PATH입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
RqPath	요청 경로	요청 경로(클래스 ID, 인스턴스 ID, 속성 ID)	_sREQUEST_PATH	–	–	–
ClassID	클래스 ID	클래스 ID	UINT	데이터 형식에 따름	–	0
InstanceID	인스턴스 ID	인스턴스 ID	UINT			
isAttributeID	속성 유무	TRUE : 속성 ID 사용 FALSE : 속성 ID 사용하지 않음	BOOL			FALSE
AttributeID	속성 ID	속성 ID	UINT			0

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다. 'ErrorIDEx'의 값과 의미는 상대 노드에 의해 규정됩니다. 상대 노드의 설명서를 참조하십시오.

배열을 송신하는 경우

'ServiceDat' 또는 'RespServiceDat'가 배열인 경우, 그러한 변수에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다.

읽고 쓸 수 있는 데이터 크기

읽을 수 있는 데이터의 최대 크기는 연결 확립을 CIPOpen 명령으로 수행했는지, CIPOpenWithDataSize 명령으로 수행했는지에 따라 다음과 같이 달라집니다.

연결 확립을 수행한 명령	읽을 수 있는 데이터의 최대 크기
CIPOpen	1990바이트
CIPOpenWithDataSize	1990바이트를 상한으로 서버가 반환하는 응답을 수신 가능

쓸 수 있는 데이터의 최대 크기는 요청 경로의 속성 유무와 연결 확립을 수행한 명령에 따라 다음과 같이 달라집니다.

쓸 수 있는 데이터 크기[바이트]=기본 크기-속성 유무

위 식의 항목	의미
기본 크기	• CIPOpen 명령으로 연결 확립을 수행한 경우, 1992바이트 • CIPOpenWithDataSize 명령으로 연결 확립을 수행한 경우, CIPOpenWithDataSize 명령의 'DataSize'-2
속성 유무	• 속성 ID를 사용하는 경우, 12바이트 • 속성 ID를 사용하지 않는 경우, 8바이트

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 반드시 CIPOpen 명령 또는 CIPOpenWithDataSize 명령을 실행하여 'Handle'을 취득하십시오.
- 'ServiceDat'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 쓸 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Size'의 값이 쓰기 데이터 영역을 초과하는 경우
  - 'Size'의 값이 'ServiceDat'의 영역을 초과하는 경우
  - 'RespSize'의 값이 'RespServiceDat'의 영역을 초과하는 경우
  - 'ServiceDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - 'RespServiceDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'Handle.Handle'의 값이 유효 범위 밖인 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 생성한 연결이 타임아웃된 경우
  - 'RqPath'와 'ServiceDat'의 크기의 합계가(CIPOpenWithDataSize 명령의 'DataSize'-2바이트)를 초과한 경우

샘플 프로그램

'CIPOpen 명령(2-878페이지)'의 샘플 프로그램을 참조하십시오.

CIPClose

지정한 핸들의 CIP Class3의 연결을 중단합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CIPClose	CIPClass3 연결 중단	FB	CIPClose_instance CIPClose Execute / Handle Done / Busy / Error / ErrorID / ErrorIDEx	CIPClose_instance(Execute, Handle, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Handle	핸들	입력	CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 취득한 핸들	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Handle	구조체 _sCIP_HANDLE에 대한 자세한 내용은 기능 설명 참조																			

기능

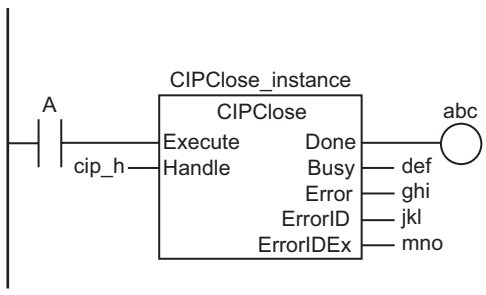
핸들 'Handle'로 지정한 CIP Class3 연결을 중단합니다.

'Handle'의 데이터 형식은 구조체 _sCIP_HANDLE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Handle	핸들	핸들	_sCIP_HANDLE	–	–	–
Handle	핸들	핸들	UDINT	데이터 형식에 따름	–	–

기술 예가 아래에 나와 있습니다. 'Handle'=cip_h로 지정한 CIP Class3 연결을 중단합니다.

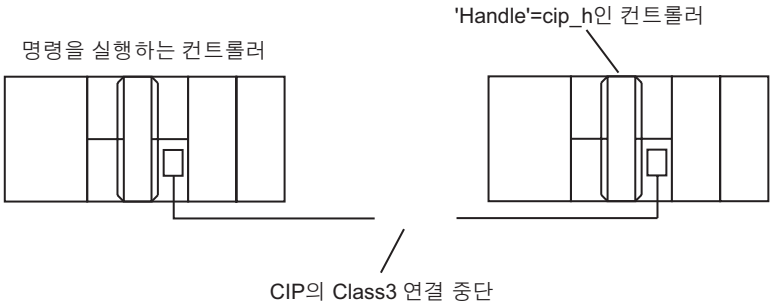
LD



ST

```
CIPClose_instance(A, cip_h, abc, def,
                  ghi, jkl, mno);
```

'Handle'로 지정한 CIP Class3 연결을 중단합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

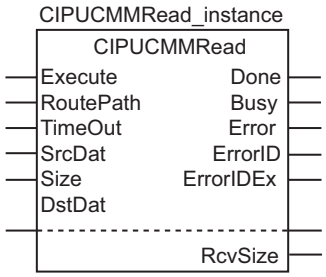
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 🕮 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'Handle'은 CIPOpen 명령 또는 CIPOpenWithDataSize 명령으로 취득한 것을 지정하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 이 명령에서는 'ErrorIDEx'를 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Handle.Handle'의 값이 유효 범위 밖인 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우

샘플 프로그램

🕮 'CIPOpen 명령(2-878페이지)'의 샘플 프로그램을 참조하십시오.

CIPUCMMRead

지정한 CIP 네트워크상의 타국 컨트롤러의 변수의 값을 읽어옵니다(UCMM Explicit 메시지).

명령	명칭	FB/FUN	그래픽 표현	ST 표현
CIPUCMMRead	변수 읽기 (UCMM Explicit 메시지)	FB		CIPUCMMRead_instance(Execute, RoutePath, TimeOut, SrcDat, Size, DstDat Done, Busy, Error, ErrorID, ErrorIDEx, RcvSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
RoutePath	라우팅 경로	입력	라우팅 경로	데이터 형식에 따름	–	–
TimeOut	타임아웃 시간		타임아웃 시간	1~65535	0.1s	20 (2s)
SrcDat	타국 변수명		읽을 타국 컨트롤러의 변수명	데이터 형식에 따름	–	''
Size	읽기 요소 수		읽을 요소 수	0~496		1
DstDat	읽기 데이터	입출력	읽은 데이터의 값	데이터 형식에 따름	–	–
RcvSize	읽기 데이터 크기	출력	읽은 데이터의 크기	0~496	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
RoutePath																				○
TimeOut							○													
SrcDat																				○
Size							○													
DstDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열, 구조체 전체, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능(*)																			
RcvSize							○													

* STRING 형식 배열은 지정할 수 없습니다.

기능

라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 타국 컨트롤러 내의 타국 변수명 'SrcDat'으로 지정한 네트워크 변수의 값을 읽습니다.
읽은 값은 읽기 데이터 'DstDat'에 저장합니다.

읽을 요소 수는 'Size'로 지정합니다. 'SrcDat'가 배열인 경우는 읽을 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 'Size'의 값이 0인 경우, 'SrcDat'가 배열인지의 여부에 관계없이 읽기를 수행하지 않습니다.

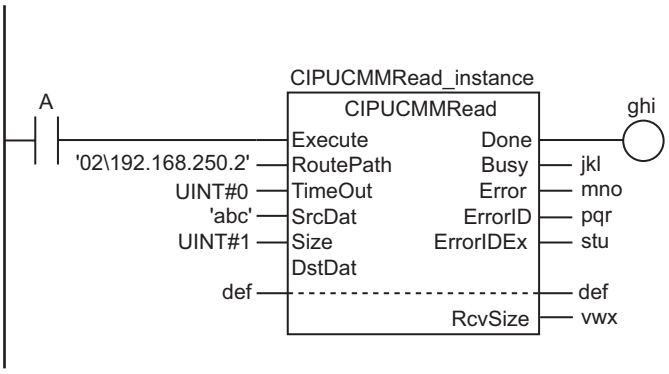
읽기 완료 후, 읽기 데이터 크기 'RcvSize'에 읽은 데이터의 크기가 바이트 단위로 대입됩니다. 읽을 수 있는 데이터의 최대 크기는 496바이트입니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다.

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다.

타국의 변수 abc의 값을 읽어 자국의 변수 def에 저장하는 경우의 예가 아래에 나와 있습니다. 읽기 요소 수 'Size'의 값은 UINT#1입니다. abc, def의 데이터 형식은 SINT입니다. SINT 형식의 데이터 크기는 1바이트이므로 읽기 데이터 크기 vwx의 값은 UINT#1이 됩니다.

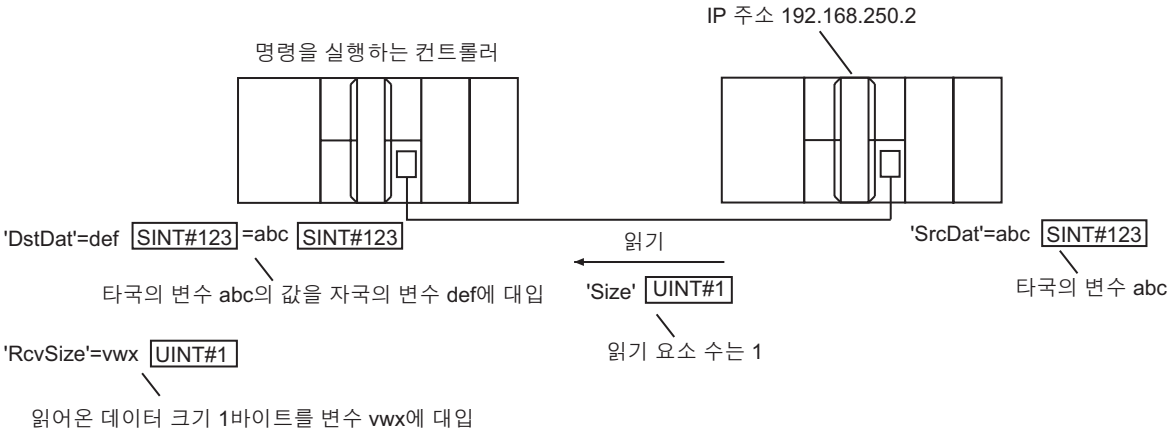
LD



ST

```
CIPUCMMRead_instance(A, '2\192.168.250.2',
                     UINT#0, 'abc', UINT#1,
                     def, ghi, jkl, mno, pqr,
                     stu, vwx);
```

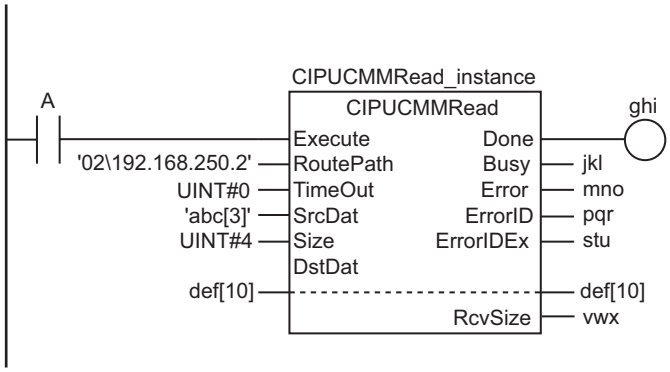
라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 타국의 변수 'SrcDat'의 값을 자국의 변수 'DstDat'에 대입합니다.
읽기 요소 수는 'Size'로 지정합니다. 읽은 데이터 크기는 'RcvSize'에 대입됩니다.



배열을 읽는 경우

읽을 데이터가 배열인 경우, 'SrcDat'에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다. 또한 'DstDat'에도 첨자가 있는 배열의 요소를 파라메터로 전달합니다.
타국의 배열 변수 abc[3]~abc[6]의 4개 요소를 읽어 자국의 배열 변수 def[10]~def[13]에 저장하는 경우의 예가 아래에 나와 있습니다. abc, def의 데이터 형식은 INT입니다. INT 형식의 데이터 크기는 2바이트이므로 읽기 데이터 크기 vwx의 값은 UINT#8이 됩니다.

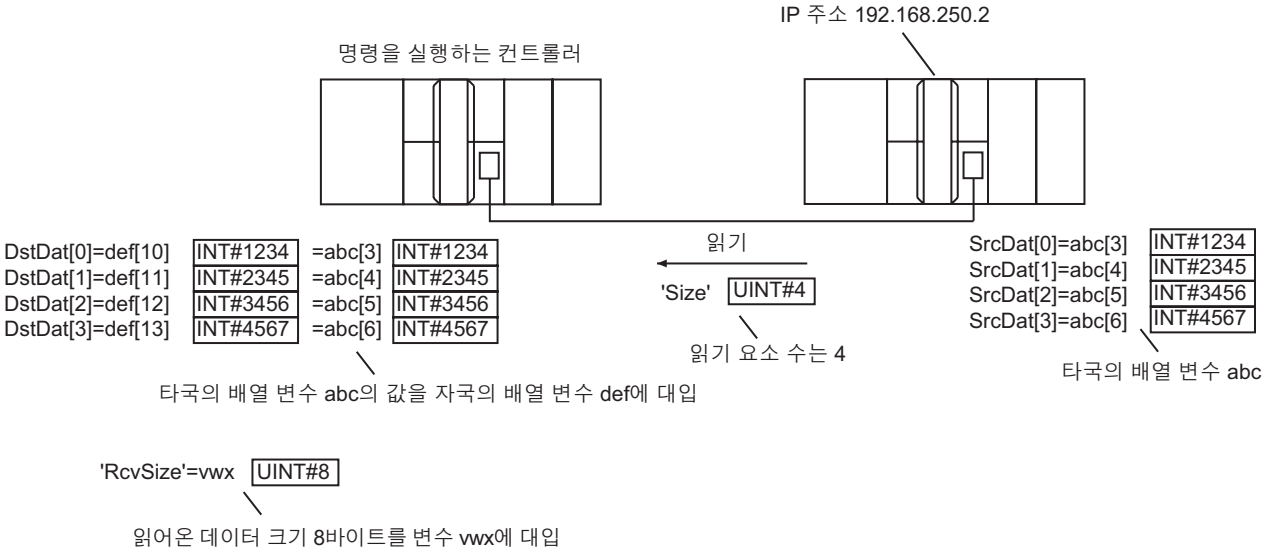
LD



ST

```
CIPUCMMRead_instance(A, '2\192.168.250.2',
                     UINT#0, 'abc[3]', UINT#4,
                     def[10], ghi, jkl, mno, pqr,
                     stu, vwx);
```

타국의 배열 변수 abc[3]~abc[6]의 값을 자국의 배열 변수 def[10]~def[13]에 대입합니다.



'RcvSize'=vwx UINT#8

읽어온 데이터 크기 8바이트를 변수 vwx에 대입

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 📖 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 📖 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 📖 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 읽을 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- CJ 유니트용 메모리의 주소를 직접 지정하여 데이터를 읽을 수는 없습니다. CJ 유니트용 메모리의 특정 주소를 읽는 경우, 사전에 그 메모리 주소를 AT 지정을 통해 변수에 할당해두십시오.
- 자국 측의 CJ 유니트용 메모리 주소를 직접 지정하여 데이터를 저장할 수도 없습니다. CJ 유니트용 메모리의 특정 주소에 저장하는 경우는 사전에 그 메모리 주소를 AT 지정을 통해 'DstDat'에 할당해두십시오.
- 'SrcDat'로 사용할 수 있는 문자의 조건은 다음과 같습니다.

항목	사양
최대 바이트 수	127바이트
문자 코드	UTF-8
사용 가능한 문자	영숫자(대문자와 소문자의 구별 없음), 반각 가타카나, 멀티 바이트 문자, '_'(밑줄)
조합하여 사용할 수 없는 문자	• ASCII 코드의 숫자 0~9(문자 코드 16#30~16#39)로 시작되는 문자열 • ASCII 코드의 '_'(밑줄)이 1문자만 있는 문자열 • ASCII 코드의 '_'(밑줄)이 2개 이상 연속으로 포함된 문자열 • 첫 번째 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 마지막 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 처음 2개의 문자가 'P_'인 문자열

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'TimeOut'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'SrcDat'의 문자열이 잘못된 경우
  - 읽은 값의 데이터 형식과 'DstDat'의 데이터 형식이 일치하지 않는 경우
  - 읽은 값의 데이터 크기가 'DstDat'의 영역을 초과하는 경우
  - 'DstDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'RoutePath'의 문자열이 잘못된 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - 타임아웃 시간이 지나도 응답이 없는 경우
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - BOOTP 서버 이상이 발생한 상태에서 이 명령이 실행된 경우
  - IP 중복 이상이 발생한 경우
- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 CIP 메시지 이상 코드의 의미를 갖습니다. 그 내용은 다음과 같습니다.

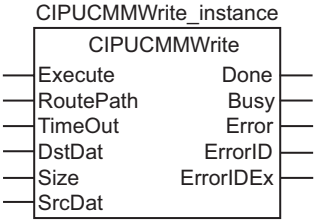
값	이상 내용
16#02000000	통신대상 노드가 고부하 상태이므로 정상적인 통신을 수행할 수 없습니다.
16#04000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 기본 데이터 형식 • 열거 형식 전체 • 구조체 전체 • 공용체 전체 • 배열 전체
16#05000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 열거 형식의 열거자 • 구조체의 1개 멤버 • 공용체의 1개 멤버 • 배열의 요소
16#0C008010	지정된 타국 변수를 다운로드 중입니다.
16#0C008011	
16#11000000	'Size'의 값이 현재 읽기 가능한 데이터 크기를 초과했습니다.
16#20008017	지정된 타국 변수가 배열이 아닌데도 읽기 요소 수가 1 이외입니다.
16#20008018	지정된 타국 변수가 배열이고, 읽기 요소 수가 그 배열의 요소 수를 초과합니다.
16#26000000	지정된 타국 변수가 NULL문자뿐입니다.

샘플 프로그램

🕮 'CIPUCMMSend 명령(2-919페이지)'의 샘플 프로그램을 참조하십시오.

CIPUCMMWrite

CIP 네트워크상의 타국 컨트롤러의 변수에 값을 씁니다(UCMM Explicit 메시지).

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPUCMM Write	변수 쓰기 (UCMM Explicit 메시지)	FB		CIPUCMMWrite_instance(Execute, RoutePath, TimeOut, DstDat, Size, SrcDat, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
RoutePath	라우팅 경로	입력	라우팅 경로	데이터 형식에 따름	–	–
TimeOut	타임아웃 시간		타임아웃 시간	1~65535	0.1s	20 (2s)
DstDat	타국 변수명		쓸 타국 컨트롤러의 변수명	데이터 형식에 따름	–	''
Size	쓰기 요소 수		쓸 요소 수	0~488		1
SrcDat	쓰기 데이터		쓸 데이터의 값	데이터 형식에 따름		*1

*1. 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
RoutePath																				○
TimeOut							○													
DstDat																				○
Size							○													
SrcDat	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열*1, 구조체 전체, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			

*1. STRING 형식 배열은 지정할 수 없습니다.

기능

라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 타국 컨트롤러 내의 타국 변수명 'DstDat'로 지정한 네트워크 변수에 값을 씁니다.
기록할 값은 쓰기 데이터 'SrcDat'의 내용입니다.

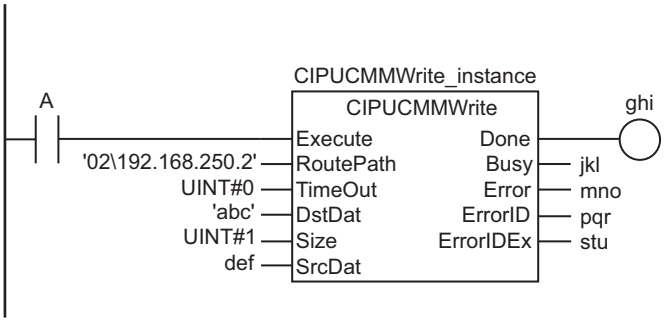
쓸 요소 수는 'Size'로 지정합니다. 'DstDat'가 배열인 경우는 쓸 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 'Size'의 값이 0인 경우, 'DstDat'가 배열인지의 여부에 관계없이 쓰기를 수행하지 않습니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다.

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다.

타국의 변수 abc에 자국의 변수 def의 값을 쓰는 경우의 예가 아래에 나와 있습니다. 쓰기 요소 수 'Size'의 값은 UINT#1입니다.

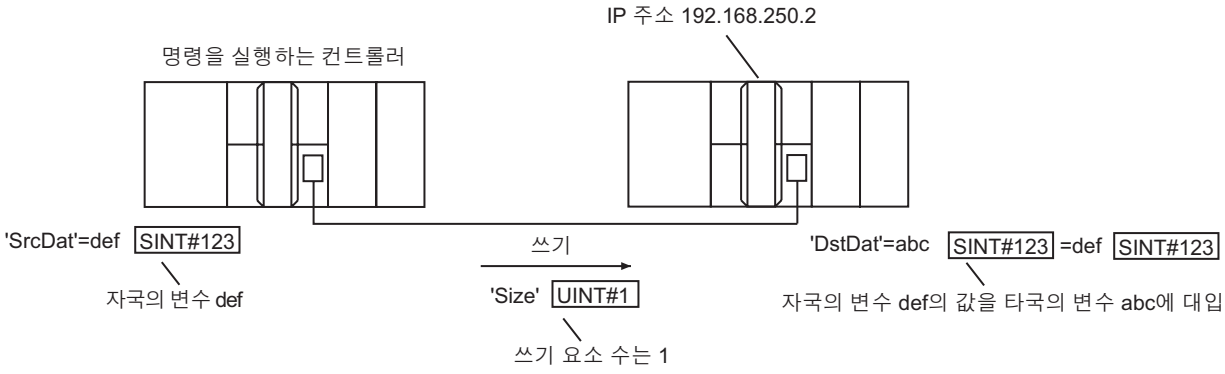
LD



ST

```
CIPUCMMWrite_instance(A, '2\192.168.250.2',
                      UINT#0, 'abc', UINT#1,
                      def, ghi, jkl, mno, pqr,
                      stu);
```

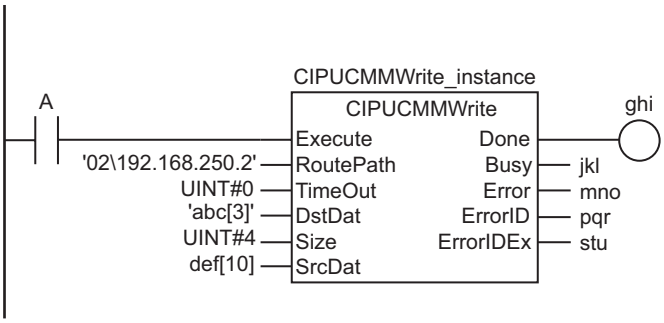
라우팅 경로 'RoutePath'로 지정한 CIP 네트워크상의 타국의 변수 'DstDat'에 자국의 변수 'SrcDat'의 값을 대입합니다. 쓰기 요소 수는 'Size'로 지정합니다.



배열에 쓰는 경우

쓸 데이터가 배열인 경우, 'DstDat'에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다. 또한 'SrcDat'에도 첨자가 있는 배열의 요소를 파라메터로 전달합니다.
타국의 배열 변수 abc[3]~abc[6]의 4개 요소에 자국의 배열 변수 def[10]~def[13]의 값을 쓰는 경우의 예가 아래에 나와 있습니다.

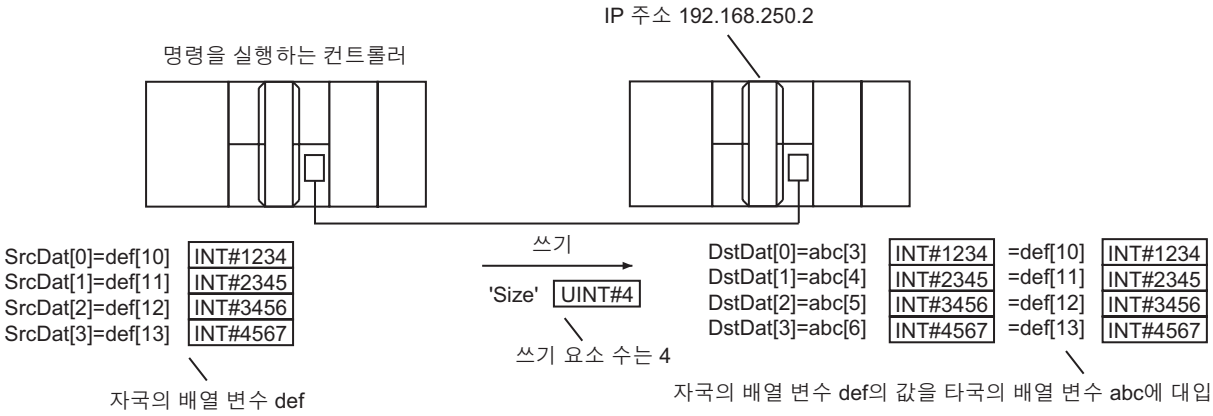
LD



ST

```
CIPUCMMWrite_instance(A, '2\192.168.250.2',
                      UINT#0, 'abc[3]', UINT#4,
                      def[10], ghi, jkl, mno, pqr,
                      stu);
```

타국의 배열 변수 abc[3]~abc[6]에 자국의 배열 변수 def[10]~def[13]의 값을 저장합니다.



쓸 수 있는 데이터 크기

쓸 수 있는 데이터의 최대 크기는 'DstDat'로 지정한 변수의 데이터 형식 및 변수명, 그리고 라우팅 경로에 따라 다음과 같이 달라집니다.

쓸 수 있는 데이터 크기[바이트]=기본 크기-'DstDat'의 변수명 크기-경로 정보 크기

위 식의 항목	의미
기본 크기	• 'DstDat'로 지정한 변수의 데이터 형식이 구조체 또는 STRING 형식인 경우, 494바이트입니다. • 상기 이외의 데이터 형식인 경우, 496바이트입니다.
'DstDat'의 변수명 크기	• 변수명의, 구조체의 각 계층의 ASCII 코드 바이트 수의 합계+(계층 수*2)로 계산합니다. • 각 계층의 문자열의 ASCII 코드 바이트 수가 홀수인 경우, 거기에 1을 더합니다. • 어떤 계층이 배열인 경우에는 다시 (배열의 차원 수*4)를 더합니다. • 구조체의 계층 구분을 나타내는 '.', 배열의 '[', ']', ', '는 변수명 크기에 포함되지 않습니다. (예 1) 'DstDat'의 변수명이 'aaa.bbbbb[1, 2, 3].cc'인 경우 • 제1계층 aaa의 문자열은 3바이트입니다. 홀수이므로 1을 더해서 4바이트로 만듭니다. • 제2계층 bbbbb[1, 2, 3]의 bbbbb의 문자열은 5바이트입니다. 홀수이므로 1을 더해서 6바이트로 만듭니다. • 그리고 제2계층 bbbbb[1, 2, 3]은 3차원 배열이므로 3*4=12를 더해서 18바이트로 만듭니다. • 제3계층 cc의 문자열은 2바이트입니다. 짝수이므로 2바이트 그대로 사용합니다. • 제1계층 4바이트, 제2계층 18바이트, 제3계층 2바이트에 계층 수 3*2=6을 더해서 변수명 크기는 30바이트가 됩니다. (예 2) 'DstDat'의 변수명이 'val'인 경우 • 제1계층 val의 문자열은 3바이트입니다. 홀수이므로 1을 더해서 4바이트로 만듭니다. • 거기에 계층 수 1*2=2를 더해서 변수명 크기는 6바이트가 됩니다. (예 3) 'DstDat'의 변수명이 'array[8]'인 경우 • 제1계층 array의 문자열은 5바이트입니다. 홀수이므로 1을 더해서 6바이트로 만듭니다. • 배열의 차원 수는 1입니다. 따라서 1*4=4를 더합니다. • 거기에 계층 수 1*2=2를 더해서 변수명 크기는 12바이트가 됩니다.

위 식의 항목	의미
경로 정보 크기	• 홉이 없는 경우, 경로 정보 크기는 0바이트입니다.(*) • 홉이 있는 경우, 경로 정보 크기는 라우팅 경로 크기+12바이트입니다. • 라우팅 경로 크기란 라우팅 경로 문자열의 ASCII 코드의 바이트 수를 말합니다. • 단, 다음과 같은 사항에 주의하십시오. • 주소부의 선두에 '#'이 붙는 경우, 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 주소부의 선두에 '#'이 붙지 않는 경우, 네트워크부는 2바이트로 계산합니다. • 주소부의 선두에 '#'이 붙지 않고 주소부의 ASCII 코드의 바이트 수가 홀수인 경우, 그 값에 1바이트를 가산합니다. • 라우팅 경로의 계층 구분을 나타내는 '\'는 라우팅 경로 크기에 포함되지 않습니다. • 처음의 홉은 라우팅 경로 크기에 포함되지 않습니다. (예1) 라우팅 경로가 '01\#11\02\192.168.250.2\01\#01'인 경우 • 처음의 홉은 라우팅 경로 크기에 포함되지 않으므로 선두의 '01\#11'은 무시합니다. • 그 다음에 나오는 네트워크 종류 '02'는 2바이트입니다. • 다음의 주소부 '192.168.250.2'는 13바이트입니다. 홀수이므로 1을 더해서 14바이트로 만듭니다. • 그 다음의 '01\#01'은 주소부에 #이 붙으므로 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 상기 값을 모두 더해서 라우팅 경로 크기는 18바이트가 됩니다. • 라우팅 경로 크기에 12바이트를 더해서 경로 정보 크기는 30바이트가 됩니다. (예 2) 라우팅 경로가 '02\192.168.250.2\01\#00'인 경우 • 처음의 홉은 라우팅 경로 크기에 포함되지 않으므로 선두의 '02\192.168.250.2'는 무시합니다. • 그 다음의 '01#00'은 주소부에 #이 붙으므로 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 따라서 라우팅 경로 크기는 2바이트가 됩니다. • 라우팅 경로 크기에 12바이트를 더해서 경로 정보 크기는 14바이트가 됩니다. (예 3) 라우팅 경로가 '02\192.168.250.2'인 경우 • 홉이 없으므로 경로 정보 크기는 0바이트입니다.

*홉이란 송신 노드와 수신 노드 사이에서 라우팅을 수행하는 것을 말합니다. 예를 들면, 라우팅 경로가 02\192.168.250.2\01\#00인 경우, 일단 IP 주소 192.168.250.2의 노드로 라우팅하여 00호기에 메시지를 송신하므로, 홉이 하나 있는 것이 됩니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'SrcDat'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'SrcDat'가 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 쓸 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- CJ 유니트용 메모리의 주소를 직접 지정하여 데이터를 쓸 수는 없습니다. CJ 유니트용 메모리의 특정 주소에 쓰는 경우, 사전에 그 메모리 주소를 AT 지정을 통해 변수에 할당해두십시오.
- 자국 측의 CJ 유니트용 메모리 주소를 직접 지정할 수도 없습니다. CJ 유니트용 메모리의 특정 주소의 값을 쓰는 경우, 사전에 그 메모리 주소를 AT 지정을 통해 'SrcDat'에 할당해두십시오.
- 'DstDat'로 사용할 수 있는 문자의 조건은 다음과 같습니다.

항목	사양
최대 바이트 수	127바이트
문자 코드	UTF-8
사용 가능한 문자	영숫자(대문자와 소문자의 구별 없음), 반각 가타카나, 멀티 바이트 문자, '_'(밑줄)
조합하여 사용할 수 없는 문자	• ASCII 코드의 숫자 0~9(문자 코드 16#30~16#39)로 시작되는 문자열 • ASCII 코드의 '_'(밑줄)이 1문자만 있는 문자열 • ASCII 코드의 '_'(밑줄)이 2개 이상 연속으로 포함된 문자열 • 첫 번째 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 마지막 문자가 ASCII 코드의 '_'(밑줄)인 문자열 • 처음 2개의 문자가 'P_'인 문자열

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'TimeOut'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'DstDat'의 문자열이 잘못된 경우
  - 'Size'의 값이 'SrcDat'의 영역을 초과하는 경우
  - 'SrcDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'RoutePath'의 문자열이 잘못된 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - 타임아웃 시간이 지나도 응답이 없는 경우
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - IP 중복 이상이 발생한 경우
- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 CIP 메시지 이상 코드의 의미를 갖습니다. 그 내용은 다음과 같습니다.

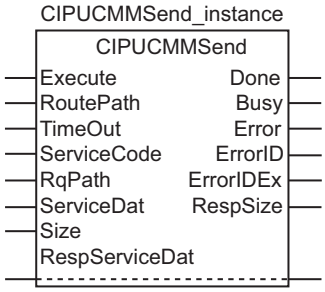
값	이상 내용
16#02000000	통신대상 노드가 고부하 상태이므로 정상적인 통신을 수행할 수 없습니다.
16#04000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 기본 데이터 형식 • 열거 형식 전체 • 구조체 전체 • 공용체 전체 • 배열 전체
16#05000000	지정된 타국 변수가 다음과 같은 데이터 형식이고 타국에 존재하지 않습니다. • 열거 형식의 열거자 • 구조체의 1개 멤버 • 공용체의 1개 멤버 • 배열의 요소
16#0C008010	지정된 타국 변수를 다운로드 중입니다.
16#0C008011	
16#1F000102	지정된 타국 변수에 상수 속성이 설정되어 있으므로 쓸 수 없습니다.
16#20008017	지정된 타국 변수가 배열이 아닌데도 쓰기 요소 수가 1 이외입니다.
16#20008018	지정된 타국 변수가 배열이고, 쓰기 요소 수가 그 배열의 요소 수를 초과합니다.
16#20008028	• 지정된 타국 변수가 열거 형식이고 쓰기 데이터가 열거자에 없는 값입니다. • 지정된 타국 변수가 범위 지정 속성을 갖고 있으며 쓰기 데이터가 범위 밖의 값입니다.
16#26000000	지정된 타국 변수명이 NULL문자뿐입니다.

샘플 프로그램

🕮 'CIPUCMMSend 명령(2-919페이지)'의 샘플 프로그램을 참조하십시오.

CIPUCMMSend

CIP 네트워크상의 지정된 장치에 UCMM의 CIP 메시지를 송신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
CIPUCMM Send	임의 Explicit 메시지 송신 (UCMM)	FB		CIPUCMMSend_instance(Execute, RoutePath, TimeOut, ServiceCode, RqPath, ServiceDat, Size, RespServiceDat, Done, Busy, Error, ErrorID, ErrorIDEx, RespSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
RoutePath	라우팅 경로	입력	라우팅 경로	데이터 형식에 따름	–	–
TimeOut	타임아웃 시간		타임아웃 시간	1~65535	0.1s	20 (2.0s)
Service Code	서비스 코드		서비스 코드	데이터 형식에 따름	–	–
RqPath	요청 경로		요청 경로(클래스 ID, 인스턴스 ID, 속성 ID)	–		
ServiceDat	명령 데이터		송신할 데이터	데이터 형식에 따름		(*)
Size	송신 요소 수		송신할 요소 수			1
Resp ServiceDat	응답 데이터	입출력	응답 데이터	데이터 형식에 따름	–	–
RespSize	응답 데이터 크기	출력	응답 데이터의 크기	데이터 형식에 따름	바이트	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
RoutePath																				○
TimeOut							○													
Service Code		○																		
RqPath	구조체 _sREQUEST_PATH에 대한 자세한 내용은 기능 설명 참조																			
ServiceDat		○	○	○	○	○	○	○	○	○	○	○	○	○	○					
	배열, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			
Size							○													
Resp ServiceDat		○	○	○	○	○	○	○	○	○	○	○	○	○	○					
	배열, 구조체의 1개 멤버, 공용체의 1개 멤버도 지정 가능																			
RespSize							○													

기능

서비스 코드 'ServiceCode'로 지정한 서비스에 대해 명령 데이터 'ServiceDat'를 UCMM 메시지로 송신합니다.
송신대상은 라우팅 경로 'RoutePath'로 지정합니다.
요청 경로는 'RqPath'로 지정합니다.

송신할 요소 수는 'Size'로 지정합니다. 'ServiceDat'가 배열인 경우는 송신할 요소 수를, 배열이 아닌 경우는 반드시 1을 지정합니다. 서비스 데이터가 필요 없는 경우는 'Size'의 값을 0으로 합니다.

송신 후, 응답 데이터가 'RespServiceDat'에 저장됩니다. 또한 응답 데이터의 크기가 바이트 단위로 'RespSize'에 저장됩니다.

타임아웃 시간은 'TimeOut'으로 지정합니다. 타임아웃 시간 이내에 응답이 반환되지 않으면 통신 실패로 판단됩니다.

'RqPath'의 데이터 형식은 구조체 _sREQUEST_PATH입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
RqPath	요청 경로	요청 경로(클래스 ID, 인스턴스 ID, 속성 ID)	_sREQUEST_PATH	–	–	–
ClassID	클래스 ID	클래스 ID	UINT	데이터 형식에 따름	–	0
InstanceID	인스턴스 ID	인스턴스 ID	UINT			
isAttribute ID	속성 유무	TRUE : 속성 ID 사용 FALSE : 속성 ID 사용하지 않음	BOOL			FALSE
AttributeID	속성 ID	속성 ID	UINT			0

'ErrorID'의 값이 WORD#16#1C00인 경우, 'ErrorIDEx'에 CIP 메시지 이상 코드가 저장됩니다.
'ErrorIDEx'의 값과 의미는 상대 노드에 의해 규정됩니다. 상대 노드의 설명서를 참조하십시오.

배열을 송신하는 경우

'ServiceDat' 또는 'RespServiceDat'가 배열인 경우, 그러한 변수에는 반드시 첨자가 있는 배열의 요소를 파라메터로 전달합니다.

읽고 쓸 수 있는 데이터 크기

읽을 수 있는 데이터의 최대 크기는 492바이트입니다.
쓸 수 있는 데이터의 최대 크기는 요청 경로의 속성 유무 및 라우팅 경로에 따라 다음과 같이 달라집니다.

쓸 수 있는 데이터 크기[바이트]=기본 크기-속성 유무-경로 정보 크기

위 식의 항목	의미
기본 크기	500바이트
속성 유무	속성 ID를 사용하는 경우, 12바이트 속성 ID를 사용하지 않는 경우, 8바이트
경로 정보 크기	• 홉이 없는 경우, 경로 정보 크기는 0바이트입니다.(*) • 홉이 있는 경우, 경로 정보 크기는 라우팅 경로 크기+12바이트입니다. • 라우팅 경로 크기란 라우팅 경로 문자열의 ASCII 코드의 바이트 수를 말합니다. • 단, 다음과 같은 사항에 주의하십시오. • 주소부의 선두에 '#'이 붙는 경우, 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 주소부의 선두에 '#'이 붙지 않는 경우, 네트워크부는 2바이트로 계산합니다. • 주소부의 선두에 '#'이 붙지 않고 주소부의 ASCII 코드의 바이트 수가 홀수인 경우, 그 값에 1바이트를 가산합니다. • 라우팅 경로의 계층 구분을 나타내는 '\'는 라우팅 경로 크기에 포함되지 않습니다. • 처음의 홉은 라우팅 경로 크기에 포함되지 않습니다. (예 1) 라우팅 경로가 '01\#11\02\192.168.250.2\01\#01'인 경우 • 처음의 홉은 라우팅 경로 크기에 포함되지 않으므로, 선두의 '01\#11'은 무시합니다. • 그 다음에 나오는 네트워크 종류 '02'는 2바이트입니다. • 다음의 주소부 '192.168.250.2'는 13바이트입니다. 홀수이므로 1을 더해서 14바이트로 만듭니다. • 그 다음의 '01\#01'은 주소부에 #이 붙으므로 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 상기 값을 모두 더해서 라우팅 경로 크기는 18바이트가 됩니다. • 라우팅 경로 크기에 12바이트를 더해서 경로 정보 크기는 30바이트가 됩니다. (예 2) 라우팅 경로가 '02\192.168.250.2\01\#00'인 경우 • 처음의 홉은 라우팅 경로 크기에 포함되지 않으므로, 선두의 '02\192.168.250.2'는 무시합니다. • 그 다음의 '01#00'은 주소부에 #이 붙으므로 네트워크부와 주소부를 합해서 2바이트로 계산합니다. • 따라서 라우팅 경로 크기는 2바이트가 됩니다. • 라우팅 경로 크기에 12바이트를 더해서 경로 정보 크기는 14바이트가 됩니다. (예 3) 라우팅 경로가 '02\192.168.250.2'인 경우 • 홉이 없으므로 경로 정보 크기는 0바이트입니다.

*홉이란 송신 노드와 수신 노드 사이에서 라우팅을 수행하는 것을 말합니다. 예를 들면, 라우팅 경로가 02\192.168.250.2\01\#00인 경우, 일단 IP 주소 192.168.250.2의 노드로 라우팅하여 00호기에 메시지를 송신하므로, 홉이 하나 있는 것이 됩니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

CIP 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
- 🕮 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 🕮 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'ServiceDat'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트 또는 NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트를 경유해야만 사용할 수 있습니다.
- 타국 컨트롤러가 오므론 제품인 경우, 쓸 수 있는 변수는 네트워크 공개가 된 것에 한합니다. 사전에 그 변수를 네트워크 공개로 지정해두십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'TimeOut'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 쓰기 데이터 영역을 초과하는 경우
  - 'Size'의 값이 'ServiceDat'의 영역을 초과하는 경우
  - 'RespSize'의 값이 'RespServiceDat'의 영역을 초과하는 경우
  - 'ServiceDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - 'RespServiceDat'에 지원하지 않는 데이터 형식을 지정한 경우
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - IP 중복 이상이 발생한 경우
  - BOOTP 서버 이상이 발생한 상태에서 이 명령이 실행된 경우
  - CIP에서 규정된 에러 응답이 반환된 경우
  - 'RoutePath'의 문자열이 잘못된 경우
  - CIP 관련 명령이 동시에 32개 넘게 실행된 경우
  - 타임아웃 시간이 지나도 응답이 없는 경우

샘플 프로그램

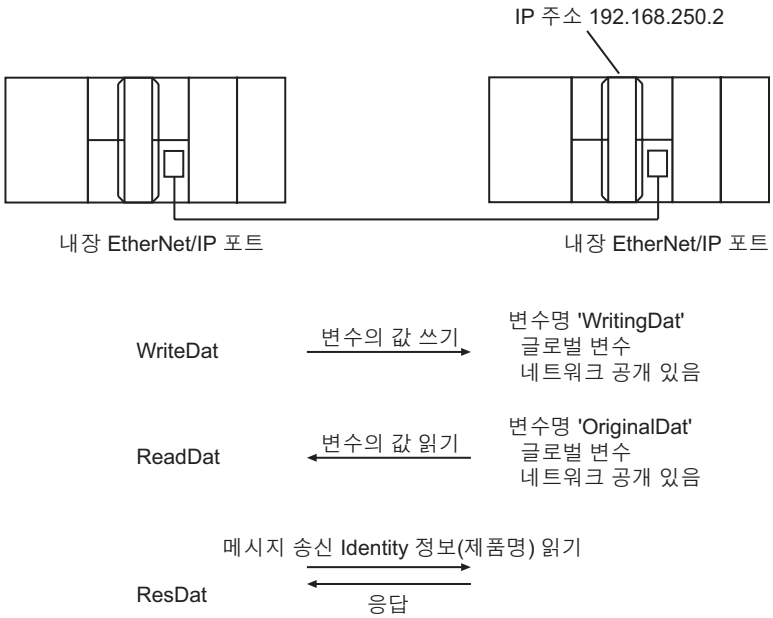
CIP의 UCMM 메시지를 사용하여 변수 쓰기, 변수 읽기, 메시지 송신을 수행합니다.
컨트롤러 간에는 EtherNet/IP 네트워크로 연결되며, 상대 노드의 IP 주소는 192.168.250.2로 합니다.
처리 순서는 다음과 같습니다.

1 **CIPUCMMWrite 명령을 사용하여 상대 노드의 변수에 값을 씁니다. 타국 변수명은 'WritingDat', 기록할 값은 변수 WriteDat의 내용으로 합니다. 'WritingDat'는 상대 노드에서 글로벌 변수로 정의되어 있고 네트워크 공개 상태여야 합니다.**

2 **CIPUCMMRead 명령을 사용하여 상대 노드의 변수의 값을 읽습니다. 타국 변수명은 'OriginalDat', 읽은 값의 저장대상은 변수 ReadDat로 합니다. 'OriginalDat'는 상대 노드에서 글로벌 변수로 정의되어 있고 네트워크 공개 상태여야 합니다.**

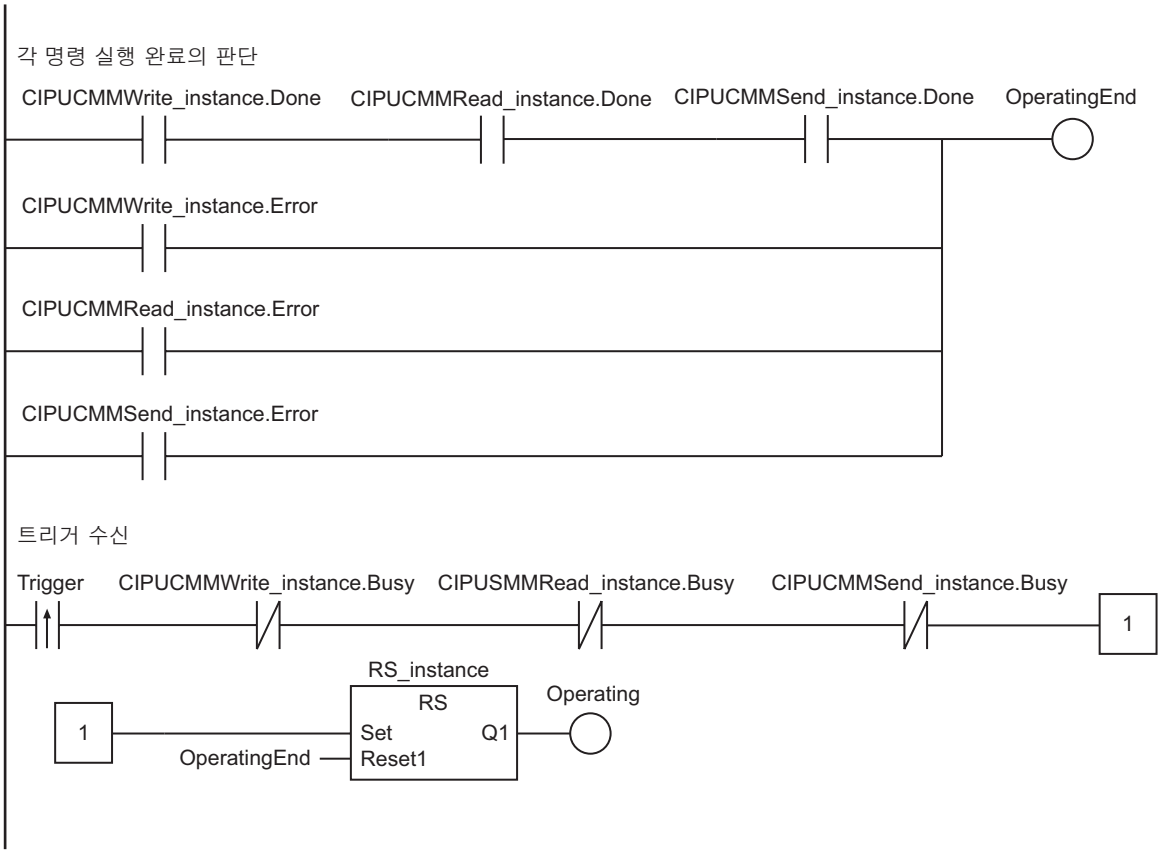
3 **CIPUCMMSend 명령을 사용하여 상대 노드에 Explicit 메시지를 송신합니다. 메시지 내용은 Identity 정보(제품명) 읽기입니다. 그때의 클래스 ID, 인스턴스 ID, 속성 ID, 서비스 코드는 다음과 같습니다. 응답 데이터는 변수 ResDat에 저장합니다.**

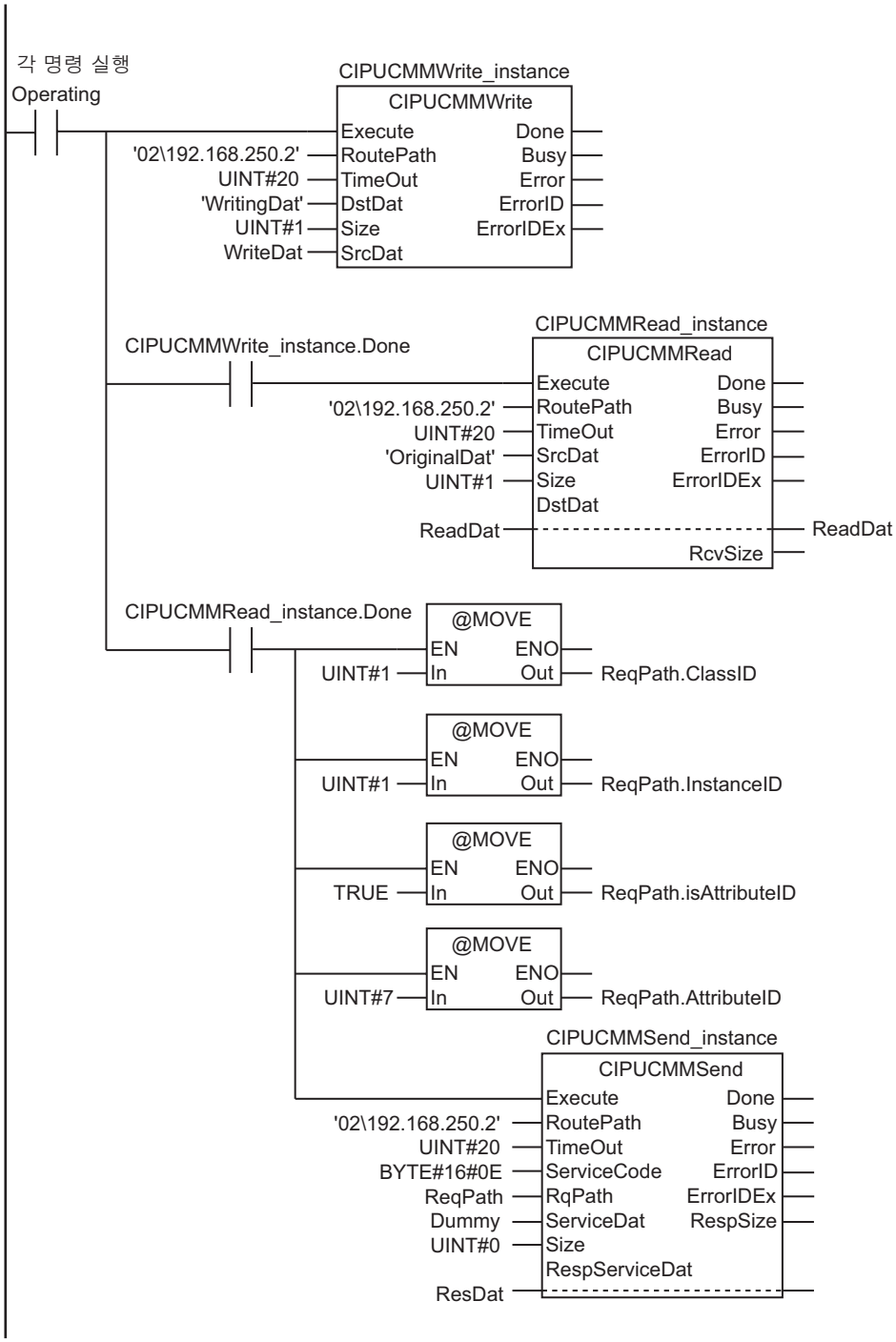
항목	값
클래스 ID	1
인스턴스 ID	1
속성 ID	7
서비스 코드	16#0E

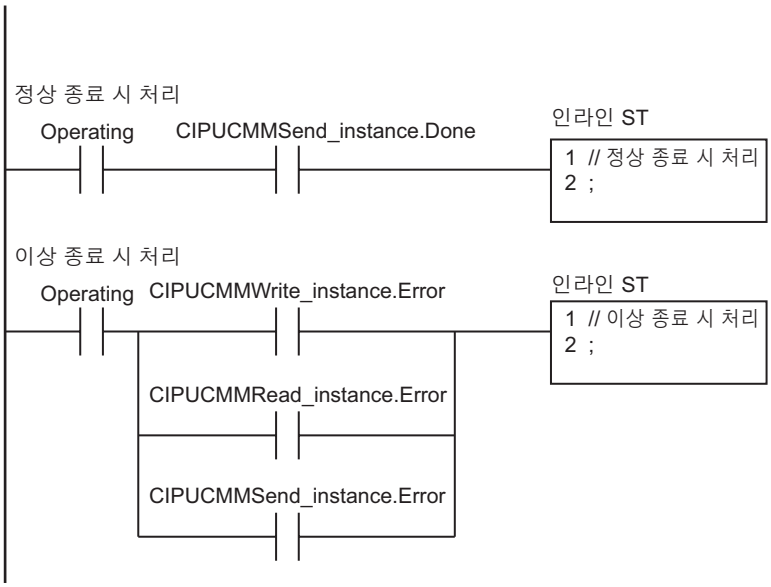


LD

명칭	데이터 형식	초기값	코멘트
OperatingEnd	BOOL	FALSE	처리 종료
Trigger	BOOL	FALSE	실행 조건
Operating	BOOL	FALSE	처리 중
WriteDat	INT	1234	쓰기 데이터
ReadDat	INT	0	읽기 데이터
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=FALSE, AttributeID:=0)	요청 경로
ResDat	ARRAY[0..10] OF BYTE	[11(16#0)]	응답 데이터
Dummy	BYTE	16#0	더미
RS_instance	RS		
CIPUCMMWrite_instance	CIPUCMMWrite		
CIPUCMMRead_instance	CIPUCMMRead		
CIPUCMMSend_instance	CIPUCMMSend		




각 명령 실행
Operating
CIPUCMMWrite_instance
CIPUCMMWrite
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
'WritingDat'
DstDat
ErrorID
UINT#1
Size
ErrorIDEx
WriteDat
SrcDat
CIPUCMMRead_instance
CIPUCMMWrite_instance.Done
CIPUCMMRead
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
'OriginalDat'
SrcDat
ErrorID
UINT#1
Size
ErrorIDEx
DstDat
ReadDat
ReadDat
RcvSize
CIPUCMMRead_instance.Done
@MOVE
EN
ENO
UINT#1
In
Out
ReqPath.ClassID
@MOVE
EN
ENO
UINT#1
In
Out
ReqPath.InstanceID
@MOVE
EN
ENO
TRUE
In
Out
ReqPath.isAttributeID
@MOVE
EN
ENO
UINT#7
In
Out
ReqPath.AttributeID
CIPUCMMSend_instance
CIPUCMMSend
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
BYTE#16#0E
ServiceCode
ErrorID
ReqPath
RqPath
ErrorIDEx
Dummy
ServiceDat
RespSize
UINT#0
Size
RespServiceDat
ResDat


정상 종료 시 처리
Operating
CIPUCMMSend_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;
이상 종료 시 처리
Operating
CIPUCMMWrite_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
CIPUCMMRead_instance.Error
CIPUCMMSend_instance.Error

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoUCMMTrigger	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	WriteDat	INT	0	쓰기 데이터
	ReadDat	INT	0	읽기 데이터
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=FALSE, AttributeID:=0)	요청 경로
	ResDat	ARRAY[0..10] OF BYTE	[11(16#0)]	응답 데이터
	Dummy	BYTE	16#0	더미
	CIPUCMMWrite_instance	CIPUCMMWrite		
	CIPUCMMRead_instance	CIPUCMMRead		
	CIPUCMMSend_instance	CIPUCMMSend		

외부 변수	명칭	정수	데이터 형식	코멘트
	_EIP_EtnOnlineSta	☑	BOOL	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoUCMMTrigger=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoUCMMTrigger:=TRUE;
    Stage              :=INT#1;
    CIPUCMMWrite_instance(
        Execute        :=FALSE,          // 인스턴스 초기화
        SrcDat         :=WriteDat);      // 더미
    CIPUCMMRead_instance(                // 인스턴스 초기화
        Execute        :=FALSE,          // 더미
        DstDat         :=ReadDat);       // 더미
    CIPUCMMSend_instance(
        Execute        :=FALSE,          // 인스턴스 초기화
        ServiceDat     := Dummy,         // 더미
        RespServiceDat:=ResDat);         // 더미
END_IF;

IF (DoUCMMTrigger=TRUE) THEN
    CASE Stage OF
    1 :                                  // 변수 쓰기 요구
        CIPUCMMWrite_instance(
            Execute    :=TRUE,
            RoutePath:='02\192.168.250.2',   // 라우팅 경로
            TimeOut    :=UINT#20,            // 타임아웃 시간
            DstDat     :='WritingDat',       // 타국 변수명
            Size       :=UINT#1,             // 쓰기 요소 수
            SrcDat     :=WriteDat);          // 쓰기 데이터

        IF (CIPUCMMWrite_instance.Done=TRUE) THEN
            Stage:=INT#2;                    // 정상 종료
        ELSIF (CIPUCMMWrite_instance.Error=TRUE) THEN
            Stage:=INT#10;                   // 이상 종료
        END_IF;
```

```
    2 :                                          // 변수 읽기 요구
        CIPUCMMRead_instance(
            Execute    :=TRUE,
            RoutePath:='02\192.168.250.2',       // 라우팅 경로
            TimeOut   :=UINT#20,                 // 타임아웃 시간
            SrcDat     :='OriginalDat',          // 타국 변수명
            Size       :=UINT#1,                 // 읽기 요소 수
            DstDat     :=ReadDat);               // 읽기 데이터

        IF (CIPUCMMRead_instance.Done=TRUE) THEN
            Stage:=INT#3;                        // 정상 종료
        ELSIF (CIPUCMMRead_instance.Error=TRUE) THEN
            Stage:=INT#40;                       // 이상 종료
        END_IF;

    3 :                                          // 임의 메시지 송신
        ReqPath.ClassID      :=UINT#01;
        ReqPath.InstanceID  :=UINT#01;
        ReqPath.isAttributeID:=TRUE;
        ReqPath.AttributeID  :=UINT#07;
        CIPUCMMSend_instance(
            Execute          :=TRUE,
            RoutePath        :='02\192.168.250.2',  // 라우팅 경로
            TimeOut          :=UINT#20,             // 타임아웃 값
            ServiceCode      :=BYTE#16#0E,          // 서비스 코드
            RqPath           :=ReqPath,             // 요청 경로
            ServiceDat       := Dummy,              // 서비스 데이터
            Size             :=UINT#0,              // 요소 수
            RespServiceDat:=ResDat);                // 응답 데이터

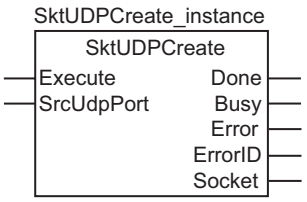
        IF (CIPUCMMSend_instance.Done=TRUE) THEN
            Stage:=INT#0;                          // 정상 종료
        ELSIF (CIPUCMMSend_instance.Error=TRUE) THEN
            Stage:=INT#30;                         // 이상 종료
        END_IF;

    0:                                 // 정상 종료 처리
        DoUCMMTrigger:=FALSE;
        Trigger             :=FALSE;

    ELSE                               // 이상 종료 처리
        DoUCMMTrigger:=FALSE;
        Trigger             :=FALSE;
    END_CASE;
END_IF;
```

SktUDPCreate

내장 EtherNet/IP의 UDP 소켓 생성 요구(서버 포트 오픈)를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktUDP Create	UDP 소켓 생성	FB		SktUDPCreate_instance(Execute, SrcUdpPort, Done, Busy, Error, ErrorID, Socket);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
SrcUdp Port	로컬 UDP 포트 번호	입력	로컬 UDP 포트 번호	1~65535	–	1
Socket	소켓	출력	소켓	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SrcUdp Port							○													
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			

기능

로컬 UDP 포트 번호 'SrcUdpPort'로 지정한 포트를 오픈합니다(소켓 함수 Socket(), Bind()의 실행).
오픈한 소켓의 정보가 'Socket'에 저장됩니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 UDP 포트가 오픈됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 이 명령으로 생성한 핸들은 SktClose 명령을 사용하여 클로즈하십시오.
- 이 명령으로 생성한 핸들은 프로그램 모드 변경 시 무효가 됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'SrcUdpPort'의 값이 유효 범위 밖인 경우
  - 'SrcUdpPort'로 지정한 포트가 이미 오픈되어 있거나 클로즈 처리 중인 경우
  - 'SrcUdpPort'로 지정한 포트가 이미 사용되고 있는 경우

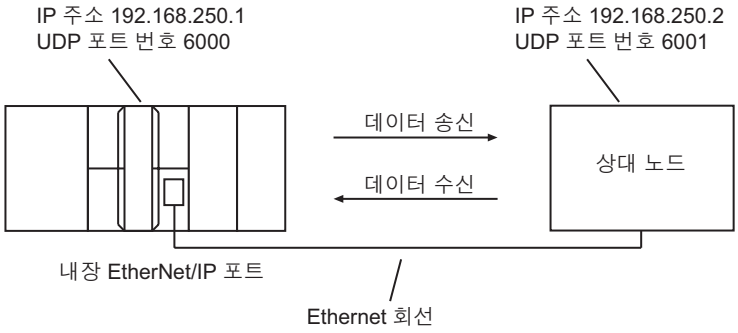
버전 관련 정보

CPU 유니트의 유니트 버전에 따라 동시에 오픈할 수 있는 소켓 수가 다음과 같이 달라집니다. 소켓 수는 UDP 소켓과 TCP 소켓을 합한 수입니다.

CPU 유니트의 버전	소켓 수
Ver.1.03 이상	최대 30
Ver.1.02 이하	최대 16

샘플 프로그램

NJ 시리즈 컨트롤러와 상대 노드 사이에서 소켓 서비스 기능(UDP)을 사용하여 데이터 송수신을 수행합니다.



처리 순서는 다음과 같습니다.

1 SktUDPCreate 명령으로 UDP 소켓의 생성 요구를 수행합니다.

2 SktUDPSend 명령으로 송신 요구를 수행합니다. 송신할 데이터는 SendSocketDat[]의 내용으로 합니다.

3 SktUDPRcv 명령으로 수신 요구를 수행합니다. 수신한 데이터는 RcvSocketDat[]에 저장합니다.

4 SktClose 명령으로 소켓을 클로즈합니다.

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoSendAndRcv	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	수신 데이터
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	소켓
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	송신 데이터
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EIP_EtnOnlineSta	BOOL	☑	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoSendAndRcv=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoSendAndRcv:=TRUE;
    Stage              :=INT#1;
    SktUDPCreate_instance(Execute:=FALSE);  // 인스턴스 초기화
    SktUDPSend_instance(                    // 인스턴스 초기화
        Execute :=FALSE,
        SendDat:=SendSocketDat[0]);         // 더미
    SktUDPRcv_instance(                     // 인스턴스 초기화
        Execute:=FALSE,
        RcvDat :=RcvSocketDat[0]);          // 더미
    SktClose_instance(Execute:=FALSE);      // 인스턴스 초기화
END_IF;


IF (DoSendAndRcv=TRUE) THEN
    CASE Stage OF
    1 :                                     // 생성 요구
        SktUDPCreate_instance(
            Execute     :=TRUE,
            SrcUdpPort:=UINT#6000,          // 로컬 UDP 포트 번호
            Socket      =>WkSocket);        // 소켓

        IF (SktUDPCreate_instance.Done=TRUE) THEN
            Stage:=INT#2;                   // 정상 종료
        ELSIF (SktUDPCreate_instance.Error=TRUE) THEN
            Stage:=INT#10;                  // 이상 종료
        END_IF;

    2 :                                     // 송신 요구
        WkSocket.DstAdr.PortNo:=UINT#6001;
        WkSocket.DstAdr.IpAdr   :='192.168.250.2';
        SktUDPSend_instance(
            Execute :=TRUE,
            Socket  :=WkSocket,             // 소켓
```

```
            SendDat:=SendSocketDat[0],                  // 송신 데이터
            Size    :=UINT#2000);                       // 송신 데이터 크기

        IF (SktUDPSend_instance.Done=TRUE) THEN
            Stage:=INT#3;                               // 정상 종료
        ELSIF (SktUDPSend_instance.Error=TRUE) THEN
            Stage:=INT#20;                              // 이상 종료
        END_IF;

    3 :                                                 // 수신 요구
        SktUDPRcv_instance(
            Execute :=TRUE,
            Socket  :=WkSocket,                         // 소켓
            TimeOut:=UINT#0,                            // 타임아웃 시간
            Size    :=UINT#2000,                        // 수신 데이터 크기
            RcvDat  :=RcvSocketDat[0]);                 // 수신 데이터

        IF (SktUDPRcv_instance.Done=TRUE) THEN
            Stage:=INT#4;                               // 정상 종료
        ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
            Stage:=INT#30;                              // 이상 종료
        END_IF;

    4 :                                                 // 클로즈 요구
        SktClose_instance(
            Execute:=TRUE,
            Socket  :=WkSocket);                        // 소켓

        IF (SktClose_instance.Done=TRUE) THEN
            Stage:=INT#0;                               // 정상 종료
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage:=INT#40;                              // 이상 종료
        END_IF;

    0 :                                                 // 정상 종료
        DoSendAndRcv:=FALSE;
        Trigger     :=FALSE;

    ELSE                                                // 이상에 의한 중단
        DoSendAndRcv:=FALSE;
        Trigger     :=FALSE;
    END_CASE;

END_IF;
```

● 상대 노드의 사용자 프로그램

이 예에서는 상대 노드에도 사용자 프로그램이 필요합니다. 송신과 수신 순서는 상기 처리 순서와 반대가 됩니다.

1 **SktUDPCreate 명령으로 UDP 소켓의 생성 요구를 수행합니다.**

2 **SktUDPRcv 명령으로 수신 요구를 수행합니다. 수신한 데이터는 RcvSocketDat[]에 저장합니다.**

3 **SktUDPSend 명령으로 송신 요구를 수행합니다. 송신할 데이터는 SendSocketDat[]의 내용으로 합니다.**

4 **SktClose 명령으로 소켓을 클로즈합니다.**

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoSendAndRcv	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	수신 데이터
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	소켓
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	송신 데이터
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EIP_EtnOnlineSta	BOOL	☑	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoSendAndRcv=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoSendAndRcv:=TRUE;
    Stage              :=INT#1;
    SktUDPCreate_instance(Execute:=FALSE);  // 인스턴스 초기화
    SktUDPSend_instance(                    // 인스턴스 초기화
        Execute :=FALSE,
        SendDat:=SendSocketDat[0]);         // 더미
    SktUDPRcv_instance(                     // 인스턴스 초기화
        Execute:=FALSE,
        RcvDat :=RcvSocketDat[0]);          // 더미
    SktClose_instance(Execute:=FALSE);      // 인스턴스 초기화
END_IF;


IF (DoSendAndRcv=TRUE) THEN
    CASE Stage OF
    1 :                                     // 생성 요구
        SktUDPCreate_instance(
            Execute     :=TRUE,
            SrcUdpPort:=UINT#6001,          // 로컬 UDP 포트 번호
            Socket      =>WkSocket);        // 소켓

        IF (SktUDPCreate_instance.Done=TRUE) THEN
            Stage:=INT#2;                   // 정상 종료
        ELSIF (SktUDPCreate_instance.Error=TRUE) THEN
            Stage:=INT#10;                  // 이상 종료
        END_IF;
```

```
    2 :                                          // 수신 요구
        SktUDPRcv_instance(
            Execute :=TRUE,
            Socket   :=WkSocket,                 // 소켓
            TimeOut:=UINT#0,                     // 타임아웃 시간
            Size      :=UINT#2000,               // 수신 데이터 크기
            RcvDat  :=RcvSocketDat[0]);          // 수신 데이터

        IF (SktUDPRcv_instance.Done=TRUE) THEN
            Stage:=INT#3;                        // 정상 종료
        ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
            Stage:=INT#20;                       // 이상 종료
        END_IF;

    3 :                                          // 송신 요구
        WkSocket.DstAdr.PortNo:=UINT#6000;
        WkSocket.DstAdr.IpAdr   :='192.168.250.1';
        SktUDPSend_instance(
            Execute :=TRUE,
            Socket   :=WkSocket,                 // 소켓
            SendDat:=SendSocketDat[0],           // 송신 데이터
            Size      :=UINT#2000);              // 송신 데이터 크기

         IF (SktUDPSend_instance.Done=TRUE) THEN
            Stage:=INT#4;                        // 정상 종료
        ELSIF (SktUDPSend_instance.Error=TRUE) THEN
            Stage:=INT#30;                       // 이상 종료
        END_IF;

    4 :                                          // 클로즈 요구
        SktClose_instance(
            Execute:=TRUE,
            Socket  :=WkSocket);                 // 소켓

        IF (SktClose_instance.Done=TRUE) THEN
            Stage:=INT#0;                        // 정상 종료
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage:=INT#40;                       // 이상 종료
        END_IF;

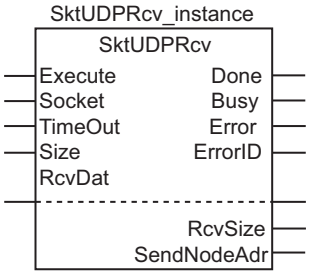
    0 :                                          // 정상 종료
        DoSendAndRcv:=FALSE;
        Trigger           :=FALSE;

    ELSE                                         // 이상에 의한 중단
        DoSendAndRcv:=FALSE;
        Trigger           :=FALSE;
    END_CASE;

END_IF;
```

SktUDPRcv

내장 EtherNet/IP의 UDP 소켓 수신 버퍼 내의 데이터를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktUDPRcv	UDP 소켓 수신	FB		SktUDPRcv_instance(Execute, Socket, TimeOut, Size, RcvDat, Done, Busy, Error, ErrorID, RcvSize, SendNodeAdr);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–
TimeOut	타임아웃 시간		0 : 타임아웃 없음 1~65535 : 0.1~6553.5s	데이터 형식에 따름	0.1s	0
Size	저장 크기		수신 버퍼에서 읽을 데이터 크기	0~2000	바이트	1
RcvDat[] 배열	수신 데이터	입출력	수신 데이터	데이터 형식에 따름	–	–
RcvSize	수신 데이터 크기	출력	실제로 RcvDat[]에 저장한 데이터 크기	0~2000	바이트	–
SendNode Adr	송신원 노드 주소		송신원 노드 주소	–	–	

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			
TimeOut							○													
Size							○													
RcvDat[] 배열		○																		
RcvSize							○													
SendNode Adr	구조체 _sSOCKET_ADDRESS에 대한 자세한 내용은 기능 설명 참조																			

기능

'Socket'으로 지정한 소켓의 수신 버퍼 내의 데이터를 수신 데이터 RcvDat[]에 저장합니다. 저장할 데이터의 크기는 'Size'로 지정합니다.
그리고 데이터 저장 후에 실제로 저장한 데이터 크기를 'RcvSize'에 대입합니다.

송신원의 노드 주소는 'SendNodeAdr'에 저장합니다.
수신 버퍼에 데이터가 없을 경우, 타임아웃 시간 'TimeOut'에 설정된 시간만큼 수신 데이터를 기다립니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 RcvDat[]로의 데이터 저장이 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

'SendNodeAdr'의 데이터 형식은 구조체 _sSOCKET_ADDRESS입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
SendNodeAdr	송신원 노드 주소	송신원 노드 주소	_sSOCKET_ADDRESS	–	–	–
PortNo	포트 번호	송신원 노드의 UDP 포트 번호	UINT	1~65535	–	–
IpAdr	IP 주소	송신원 노드의 IP 주소	STRING	데이터 형식에 따름		

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

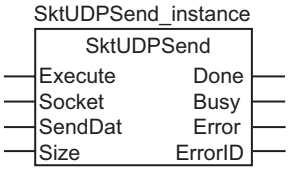
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 한 번에 수신할 수 있는 데이터 크기는 최대 2000바이트입니다.
- 지정한 소켓이 수신하는 데이터 크기가 'Size'의 값보다 작은 경우, 수신 중인 데이터만이 RcvDat[]에 저장됩니다. 그리고 저장한 데이터 크기를 'RcvSize'에 대입합니다.
- 지정한 소켓이 수신하는 데이터 크기가 'Size'의 값보다 큰 경우, 'Size'만큼의 데이터만이 RcvDat[]에 저장됩니다.
- 'Size'의 값이 0인 경우, 수신 데이터 읽기를 수행하지 않습니다.
- 수신 버퍼에 데이터가 없을 경우, 소켓을 SktClose 명령으로 클로즈하면 타임아웃 시간 미만이더라도 수신 데이터를 기다리지 않고 정상 종료합니다. 이 경우, 'RcvSize'의 값은 0이 됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'Socket'으로 지정한 소켓이 수신 처리 중인 경우
  - 'Socket'으로 지정한 소켓이 오픈되어 있지 않은 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우

샘플 프로그램

'SktUDPCreate 명령(2-929페이지)'의 샘플 프로그램을 참조하십시오.

SktUDPSend

내장 EtherNet/IP의 UDP 포트에서 데이터를 송신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktUDPSend	UDP 소켓 송신	FB		SktUDPSend_instance(Execute, Socket, SendDat, Size, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–
SendDat[] 배열	송신 데이터		송신 데이터	데이터 형식에 따름		
Size	송신 데이터 크기		송신 데이터 크기	0~2000	바이트	1

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			
SendDat[] 배열		○																		
Size							○													

기능

'Socket'으로 지정한 소켓에서 송신 데이터 SendDat[]를 송신합니다.
송신할 데이터 크기는 'Size'로 지정합니다.
상대 노드는 'Socket.DstAdr'로 지정합니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 송신 버퍼에 SendDat[]의 저장이 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 한 번에 송신할 수 있는 데이터 크기는 최대 2000바이트입니다. SendDat[]의 배열 크기가 2000바이트보다 크더라도 2000바이트까지만 송신할 수 있습니다. 또한 브로드캐스트 주소 지정 시에는 1472바이트까지만 송신할 수 있습니다.
- 'Size'의 값이 0인 경우, 0바이트의 송신 데이터를 회선에 송신합니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

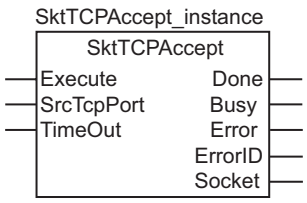
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket'으로 지정한 소켓이 송신 처리 중인 경우
  - 'Socket'으로 지정한 소켓이 오픈되어 있지 않은 경우
  - 'Socket'으로 상대 노드를 도메인명으로 지정하여 주소 해결에 실패한 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우
  - 'Size'의 값이 SendDat[]의 요소 수를 초과한 경우

샘플 프로그램

'SktUDPCreate 명령(2-929페이지)'의 샘플 프로그램을 참조하십시오.

SktTCPAccept

내장 EtherNet/IP의 TCP 소켓 허용 요구를 수행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktTCP Accept	TCP 소켓 허용	FB		SktTCPAccept_instance(Execute, SrcTcpPort, TimeOut, Done, Busy, Error, ErrorID, Socket);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
SrcTcpPort	로컬 TCP 포트 번호	입력	로컬 TCP 포트 번호	1~65535	–	1
TimeOut	타임아웃 시간		0 : 타임아웃 없음 1~65535 : 0.1~6553.5s	데이터 형식에 따름	0.1s	0
Socket	소켓	출력	소켓	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SrcTcpPort							○													
TimeOut							○													
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			

기능

로컬 TCP 포트 번호 'SrcTcpPort'로 지정한 포트의 허용 요구를 수행합니다(소켓 함수 Socket(), Bind(), Listen(), Accept() 실행).
타임아웃 시간 'TimeOut'에 설정된 시간만큼 통신 상대로부터의 연결 확립을 기다립니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 연결이 확립됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	-	–	–
PortNo	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

- 소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.
- 이 명령을 여러 번 실행하여 하나의 로컬 포트 번호로 복수의 클라이언트로부터의 연결을 개설할 수 있습니다. 그 경우, 연결별로 다른 소켓을 반환합니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 이 명령으로 생성한 핸들은 SktClose 명령을 사용하여 클로즈하십시오.
- 이 명령으로 생성한 핸들은 프로그램 모드 변경 시 무효가 됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'SrcTcpPort'의 값이 유효 범위 밖인 경우
  - 'SrcTcpPort'로 지정한 TCP 포트가 오픈 처리 중인 경우
  - 'SrcTcpPort'로 지정한 TCP 포트가 클로즈 처리 중인 경우
  - 'TimeOut'으로 지정한 시간만큼 기다려도 연결이 확립되지 않는 경우

버전 관련 정보

CPU 유니트의 유니트 버전에 따라 동시에 오픈할 수 있는 소켓 수가 다음과 같이 달라집니다. 소켓 수는 UDP 소켓과 TCP 소켓을 합한 수입니다.

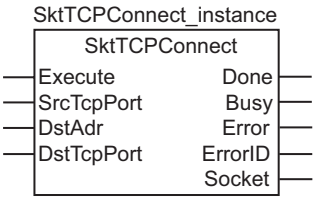
CPU 유니트의 버전	소켓 수
Ver.1.03 이상	최대 30
Ver.1.02 이하	최대 16

샘플 프로그램

'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

SktTCPConnect

내장 EtherNet/IP에서 상대 TCP 포트로 연결합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktTCP Connect	TCP 소켓 연결	FB		SktTCPConnect_instance(Execute, SrcTcpPort, DstAdr, DstTcpPort, Done, Busy, Error, ErrorID, Socket);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
SrcTcpPort	로컬 TCP 포트 번호	입력	로컬 TCP 포트 번호 단, 0인 경우에는 Well-Known 포트 이외의 1024에서 비어 있는 TCP 포트를 자동 할당	데이터 형식에 따름	–	0
DstAdr	상대 주소		상대의 IP 주소 또는 호스트명	최대 200바이트		–
DstTcp Port	상대 TCP 포트 번호		상대 TCP 포트 번호	1~65535		1
Socket	소켓	출력	소켓	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SrcTcpPort							○													
DstAdr																				○
DstTcp Port							○													
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			

기능

로컬 TCP 포트 번호 'SrcTcpPort'와 상대 주소 'DstAdr'의 상대 TCP 포트 번호 'DstTcpPort'를 연결하기 위한 연결 요구를 수행합니다(소켓 함수 Connect()의 실행).
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 연결이 확립됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 이 명령으로 생성한 핸들은 SktClose 명령을 사용하여 클로즈하십시오.
- 이 명령으로 생성한 핸들은 프로그램 모드 변경 시 무효가 됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'DstAdr'의 값이 유효 범위 밖인 경우
  - 'DstTcpPort'의 값이 유효 범위 밖인 경우
  - 'SrcTcpPort'로 지정한 TCP 포트가 이미 오픈되어 있는 경우
  - 'DstAdr'로 지정한 상대 노드가 존재하지 않는 경우
  - 'DstAdr', 'DstTcpPort'로 지정한 상대 노드가 연결 대기 상태가 아닌 경우
  - 'DstAdr'로 지정한 호스트명의 주소 해결에 실패한 경우
  - 동일 클라이언트(IP 주소, TCP 포트)와의 연결이 이미 확립된 경우

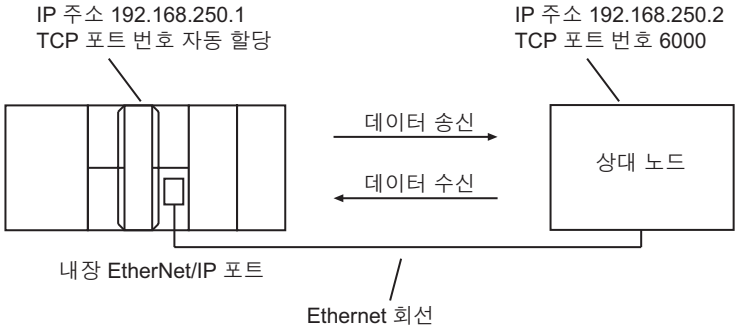
버전 관련 정보

CPU 유니트의 유니트 버전에 따라 동시에 오픈할 수 있는 소켓 수가 다음과 같이 달라집니다. 소켓 수는 UDP 소켓과 TCP 소켓을 합한 수입니다.

CPU 유니트의 버전	소켓 수
Ver.1.03 이상	최대 30
Ver.1.02 이하	최대 16

샘플 프로그램

NJ 시리즈 컨트롤러와 상대 노드 사이에서 소켓 서비스 기능(TCP)을 사용하여 데이터 송수신을 수행합니다.



처리 순서는 다음과 같습니다.

1 **SktTCPConnect 명령으로 상대 노드의 TCP 포트에 연결 요구를 수행합니다.**

2 **SktClearBuf 명령으로 TCP 소켓의 수신 버퍼를 클리어합니다.**

3 **SktGetTCPStatus 명령으로 TCP 소켓의 Status를 읽어옵니다.**

4 **SktTCPSend 명령으로 송신 요구를 수행합니다. 송신할 데이터는 SendSocketDat[]의 내용으로 합니다.**

5 **SktTCPRcv 명령으로 수신 요구를 수행합니다. 수신한 데이터는 RcvSocketDat[]에 저장합니다.**

6 **SktClose 명령으로 소켓을 클로즈합니다.**

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoTCP	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	수신 데이터
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	소켓
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	송신 데이터
	SktTCPConnect_instance	SktTCPConnect		
	SktClearBuf_instance	SktClearBuf		
	SktGetTCPStatus_instance	SktGetTCPStatus		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EIP_EtnOnlineSta	BOOL	☐	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoTCP=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoTCP:=TRUE;
    Stage  :=INT#1;
    SktTCPConnect_instance(Execute:=FALSE);     // 인스턴스 초기화
    SktClearBuf_instance(Execute:=FALSE);       // 인스턴스 초기화
    SktGetTCPStatus_instance(Execute:=FALSE);   // 인스턴스 초기화
    SktTCPSend_instance(                        // 인스턴스 초기화
        Execute:=FALSE,
        SendDat:=SendSocketDat[0]);             // 더미
    SktTCPRcv_instance(                         // 인스턴스 초기화
        Execute:=FALSE,
        RcvDat :=RcvSocketDat[0]);              // 더미
    SktClose_instance(Execute:=FALSE);          // 인스턴스 초기화
END_IF;


IF (DoTCP=TRUE) THEN
    CASE Stage OF
    1 :                                 // 연결 요구
        SktTCPConnect_instance(
            Execute    :=TRUE,
            SrcTcpPort:=UINT#0,                 // 로컬 UDP 포트 번호 자동 할당
            DstAdr     :='192.168.250.2',       // 상대 IP 주소
            DstTcpPort:=UINT#6000,              // 상대 TCP 포트 번호
            Socket     =>WkSocket);             // 소켓

        IF (SktTCPConnect_instance.Done=TRUE) THEN
            Stage:=INT#2;                       // 정상 종료
        ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
            Stage:=INT#10;                      // 이상 종료
        END_IF;
```

```
2 :                                     // 수신 버퍼 클리어
  SktClearBuf_instance(
    Execute:=TRUE,
    Socket  :=WkSocket);                // 소켓

  IF (SktClearBuf_instance.Done=TRUE) THEN
    Stage:=INT#3;                       // 정상 종료
  ELSIF (SktClearBuf_instance.Error=TRUE) THEN
    Stage:=INT#20;                      // 이상 종료
  END_IF;

3 :                                     // Status 읽기 요구
  SktGetTCPStatus_instance(
    Execute:=TRUE,
    Socket  :=WkSocket);                // 소켓

  IF (SktGetTCPStatus_instance.Done=TRUE) THEN
    Stage:=INT#4;                       // 정상 종료
  ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
    Stage:=INT#30;                      // 이상 종료
  END_IF;

4 :                                     // 송신 요구
  SktTCPSend_instance(
    Execute :=TRUE,
    Socket  :=WkSocket,                 // 소켓
    SendDat:=SendSocketDat[0],          // 송신 데이터
    Size    :=UINT#2000);               // 송신 데이터 크기

  IF (SktTCPSend_instance.Done=TRUE) THEN
    Stage:=INT#5;                       // 정상 종료
  ELSIF (SktTCPSend_instance.Error=TRUE) THEN
    Stage:=INT#40;                      // 이상 종료
  END_IF;

5 :                                     // 수신 요구
  SktTCPRcv_instance(
    Execute :=TRUE,
    Socket  :=WkSocket,                 // 소켓
    TimeOut:=UINT#0,                    // 타임아웃 시간
    Size    :=UINT#2000,                // 수신 데이터 크기
    RcvDat  :=RcvSocketDat[0]);         // 수신 데이터

  IF (SktTCPRcv_instance.Done=TRUE) THEN
    Stage:=INT#6;                       // 정상 종료
  ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
    Stage:=INT#50;                      // 이상 종료
  END_IF;

6 :                                     // 클로즈 요구
  SktClose_instance(
    Execute:=TRUE,
    Socket  :=WkSocket);                // 소켓

  IF (SktClose_instance.Done=TRUE) THEN
    Stage:=INT#0;                       // 정상 종료
  ELSIF (SktClose_instance.Error=TRUE) THEN
    Stage:=INT#40;                      // 이상 종료
  END_IF;
```

```
    0 :                                     // 정상 종료
        DoTCP:=FALSE;
        Trigger :=FALSE;

    ELSE                                    // 이상에 의한 중단
        DoTCP:=FALSE;
        Trigger :=FALSE;
    END_CASE;

END_IF;
```

● 상대 노드의 사용자 프로그램

이 예에서는 상대 노드에도 사용자 프로그램이 필요합니다. 송신과 수신 순서는 상기 처리 순서와 반대가 됩니다.

1. **SktTCPAccept 명령으로 TCP 소켓의 허용 요구를 수행합니다.**
2. **SktTCPRcv 명령으로 수신 요구를 수행합니다. 수신한 데이터는 RcvSocketDat[]에 저장합니다.**
3. **SktTCPSend 명령으로 송신 요구를 수행합니다. 송신할 데이터는 SendSocketDat[]의 내용으로 합니다.**
4. **SktClose 명령으로 소켓을 클로즈합니다.**

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	DoTCP	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	수신 데이터
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=''), DstAdr:=(PortNo:=0, IpAdr:=''))	소켓
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	송신 데이터
	SktTCPAccept_instance	SktTCPAccept		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

외부 변수	명칭	데이터 형식	상수	코멘트
	_EIP_EtnOnlineSta	BOOL	☐	온라인

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (DoTCP=FALSE) AND (_Eip_EtnOnlineSta=TRUE) ) THEN
    DoTCP:=TRUE;
    Stage  :=INT#1;
    SktTCPAccept_instance(Execute:=FALSE);          // 인스턴스 초기화
    SktTCPSend_instance(                            // 인스턴스 초기화
        Execute:=FALSE,
        SendDat:=SendSocketDat[0]);                 // 더미
    SktTCPRcv_instance(                             // 인스턴스 초기화
        Execute:=FALSE,
        RcvDat :=RcvSocketDat[0]);                  // 더미
    SktClose_instance(Execute:=FALSE);              // 인스턴스 초기화
END_IF;


IF (DoTCP=TRUE) THEN
    CASE Stage OF
    1 :                                 // 허용 요구
        SktTCPAccept_instance(
            Execute     :=TRUE,
            SrcTcpPort:=UINT#6000,                  // 로컬 TCP 포트 번호
            TimeOut     :=UINT#0,                   // 타임아웃 시간
            Socket      =>WkSocket);                // 소켓

        IF (SktTCPAccept_instance.Done=TRUE) THEN
            Stage:=INT#2;                           // 정상 종료
        ELSIF (SktTCPAccept_instance.Error=TRUE) THEN
            Stage:=INT#10;                          // 이상 종료
        END_IF;
```

```
    2 :                                     // 수신 요구
        SktTCPRcv_instance(
            Execute :=TRUE,
            Socket   :=WkSocket,                        // 소켓
            TimeOut:=UINT#0,                            // 타임아웃 시간
            Size      :=UINT#2000,                      // 수신 데이터 크기
            RcvDat  :=RcvSocketDat[0]);                 // 수신 데이터

        IF (SktTCPRcv_instance.Done=TRUE) THEN
            Stage:=INT#3;                               // 정상 종료
        ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
            Stage:=INT#20;                              // 이상 종료
        END_IF;

    3 :                                     // 송신 요구
        SendSocketDat:=RcvSocketDat;
        SktTCPSend_instance(
            Execute :=TRUE,
            Socket   :=WkSocket,                        // 소켓
            SendDat:=SendSocketDat[0],                  // 송신 데이터
            Size      :=UINT#2000);                     // 송신 데이터 크기

         IF (SktTCPSend_instance.Done=TRUE) THEN
            Stage:=INT#4;                               // 정상 종료
        ELSIF (SktTCPSend_instance.Error=TRUE) THEN
            Stage:=INT#30;                              // 이상 종료
        END_IF;

    4 :                                     // 클로즈 요구
        SktClose_instance(
            Execute:=TRUE,
            Socket  :=WkSocket);                        // 소켓

        IF (SktClose_instance.Done=TRUE) THEN
            Stage:=INT#0;                               // 정상 종료
        ELSIF (SktClose_instance.Error=TRUE) THEN
            Stage:=INT#40;                              // 이상 종료
        END_IF;

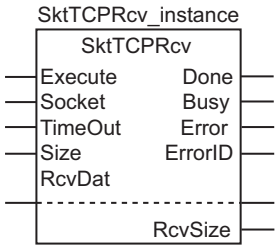
    0 :                                     // 정상 종료
        DoTCP:=FALSE;
        Trigger:=FALSE;

    ELSE                                    // 이상에 의한 중단
        DoTCP:=FALSE;
        Trigger:=FALSE;
    END_CASE;

END_IF;
```

SktTCPRcv

내장 EtherNet/IP의 지정된 TCP 소켓의 수신 버퍼 내의 데이터를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktTCPRcv	TCP 소켓 수신	FB		SktTCPRcv_instance(Execute, Socket, TimeOut, Size, RcvDat, Done, Busy, Error, ErrorID, RcvSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–
TimeOut	타임아웃 시간		0 : 타임아웃 없음 1~65535 : 0.1~6553.5s	데이터 형식에 따름	0.1s	0
Size	저장 크기		수신 버퍼에서 읽을 데이터 크기	0~2000	바이트	1
RcvDat[] 배열	수신 데이터	입출력	수신 데이터	데이터 형식에 따름	–	–
RcvSize	수신 데이터 크기	출력	실제로 RcvDat[]에 저장한 데이터 크기	0~2000	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			
TimeOut							○													
Size							○													
RcvDat[] 배열		○																		
RcvSize							○													

기능

'Socket'으로 지정한 소켓의 수신 버퍼 내의 데이터를 수신 데이터 RcvDat[]에 저장합니다. 저장할 데이터의 크기는 'Size'로 지정합니다.
그리고 데이터 저장 후에 실제로 저장한 데이터 크기를 'RcvSize'에 대입합니다.
수신 버퍼에 데이터가 없는 경우는 타임아웃 시간 'TimeOut'에 설정된 시간만큼 수신 데이터를 기다립니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 RcvDat[]로의 데이터 저장이 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 한 번에 읽을 수 있는 데이터 크기는 최대 2000바이트입니다. RcvDat[]의 배열 크기가 2000바이트보다 크더라도 2000바이트까지만 읽을 수 있습니다.
- 지정한 소켓이 수신하는 데이터 크기가 'Size'의 값보다 작은 경우, 수신 중인 데이터만이 RcvDat[]에 저장됩니다. 그리고 저장한 데이터 크기를 'RcvSize'에 대입합니다.
- 지정한 소켓이 수신하는 데이터 크기가 'Size'의 값보다 큰 경우, 'Size'만큼의 데이터만이 RcvDat[]에 저장됩니다.
- 'Size'의 값이 0인 경우, 수신 데이터 읽기를 수행하지 않습니다.

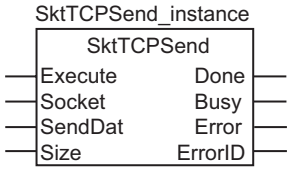
- 수신 버퍼에 데이터가 없을 경우, 소켓을 SktClose 명령으로 클로즈하면 타임아웃 시간 미만이더라도 이상 종료됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket'으로 지정한 소켓이 수신 처리 중인 경우
  - 'Socket'으로 지정한 소켓의 연결이 확립되어 있지 않은 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우
  - 'TimeOut'으로 지정한 시간만큼 기다려도 데이터를 수신하지 못한 경우
  - 소켓을 SktClose 명령으로 클로즈한 경우

샘플 프로그램

🕮 'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

SktTCPSend

내장 EtherNet/IP의 지정된 TCP 포트에서 데이터를 송신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktTCPSend	TCP 소켓 송신	FB		SktTCPSend_instance(Execute, Socket, SendDat, Size, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–
SendDat[] 배열	송신 데이터		송신 데이터	데이터 형식에 따름		
Size	송신 데이터 크기		송신 데이터 크기	0~2000	바이트	1

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			
SendDat[] 배열		○																		
Size							○													

기능

'Socket'으로 지정한 소켓에서 송신 데이터 SendDat[]를 송신합니다.
송신할 데이터 크기는 'Size'로 지정합니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- 한 번에 송신할 수 있는 데이터 크기는 최대 2000바이트입니다. SendDat[]의 배열 크기가 2000바이트보다 크더라도 2000바이트까지만 송신할 수 있습니다.
- 'Size'의 값이 0인 경우, 데이터 송신을 수행하지 않습니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

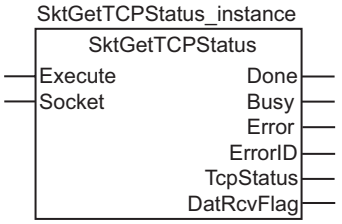
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket'으로 지정한 소켓이 송신 처리 중인 경우
  - 'Socket'으로 지정한 소켓의 연결이 확립되어 있지 않은 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우

샘플 프로그램

🕮 'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

SktGetTCPStatus

TCP 소켓의 Status를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktGetTCP Status	TCP 소켓의 Status 읽기	FB		SktGetTCPStatus_instance(Execute, Socket, Done, Busy, Error, ErrorID, TcpStatus, DatRcvFlag);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–
TcpStatus	TCP 연결 Status	출력	TCP 연결 Status	(*)	–	–
DatRcv Flag	수신 데이터 유무 플래그		TRUE : 수신 데이터 있음 FALSE : 수신 데이터 없음	데이터 형식에 따름		

* _CLOSED, _LISTEN, _SYN_SENT, _SYN_RECEIVED, _ESTABLISHED, _CLOSE_WAIT, _FIN_WAIT1, _CLOSING, _LAST_ACK, _FIN_WAIT2, _TIME_WAIT

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			
TcpStatus	열거 형식 _eCONNECTION_STATE 열거자는 기능 설명 참조																			
DatRcv Flag	○																			

기능

'Socket'으로 지정한 소켓의 TCP 연결 Status 'TcpStatus'를 취득합니다.
또한 수신 버퍼에 수신 데이터가 존재하는 경우에는 수신 데이터 유무 플래그 'DatRcvFlag'의 값을 TRUE로 합니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 'TcpStatus'와 'DatRcvFlag'로의 데이터 저장이 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	데이터 송수신 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

'TcpStatus'의 데이터 형식은 열거 형식 _eCONNECTION_STATE입니다.
열거자는 TCP의 각 상태를 나타냅니다. 각각의 의미는 다음과 같습니다.

열거자	TCP의 상태	의미
_CLOSED	CLOSED	연결이 클로즈됨
_LISTEN	LISTEN	서버측이 패시브 오픈으로 연결 확립 요구(SYN)를 기다리고 있음
_SYN_SENT	SYN SENT	클라이언트측이 액티브 오픈으로 연결 확립 요구(SYN)를 송신하고, 그에 대한 확인 응답(SYN+ACK)을 기다리고 있음
_SYN_RECEIVED	SYN RECEIVED	서버측이 연결 확립 요구(SYN)에 대한 확인 응답(SYN+ACK)을 송신하고, 그에 대한 확인 응답(ACK)을 기다리고 있음
_ESTABLISHED	ESTABLISHED	연결이 확립됨
_CLOSE_WAIT	CLOSE WAIT	서버측이 연결 중단 요구(FIN)에 대한 확인 응답(ACK)을 송신하고, 서버의 응용 프로그램을 중단할 수 있게 되기를 기다리고 있음
_FIN_WAIT1	FIN WAIT-1	클라이언트측이 연결 중단 요구(FIN)를 송신하고, 그에 대한 확인 응답(ACK)을 기다리고 있음
_CLOSING	CLOSING	클라이언트측과 서버측이 동시에 연결 중단 요구(FIN)를 수신하고, 그에 대한 확인 응답(ACK)을 기다리고 있음
_LAST_ACK	LAST-ACK	서버측이 연결 중단 요구(FIN)를 송신하고, 그에 대한 확인 응답(ACK)을 기다리고 있음
_FIN_WAIT2	FIN WAIT-2	클라이언트측이 연결 중단 요구(FIN)를 기다리고 있음
_TIME_WAIT	TIME WAIT	클라이언트측이 연결 중단 요구(FIN)에 대한 확인 응답(ACK)을 수신하고, 서버에 도달하여 처리되기를 기다리고 있음

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

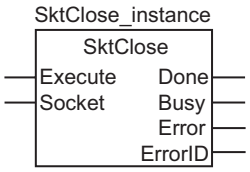
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우

샘플 프로그램

'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

SktClose

내장 EtherNet/IP의 지정된 TCP 소켓 또는 UDP 소켓을 클로즈합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktClose	TCP/UDP 소켓 클로즈	FB		SktClose_instance(Execute, Socket, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			

기능

'Socket'으로 지정한 소켓을 클로즈합니다.
TCP 소켓을 지정한 경우, 연결 중단을 수행한 후에 클로즈합니다.
소켓의 핸들 'Socket.Handle'의 값이 0인 경우, 소켓 서비스 기능으로 사용하고 있는 모든 TCP/UDP 포트를 클로즈합니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 TCP/UDP 소켓의 클로즈가 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	클로즈할 연결의 핸들 0 : 소켓 서비스 기능에서 사용하고 있는 모든 TCP/UDP 연결을 클로즈	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- SktUDPRcv 명령 또는 SktTCPRcv 명령을 실행하여 지정한 핸들의 소켓에서 수신을 기다리는 도중에 이 명령을 실행한 경우, 수신 대기를 취소합니다.

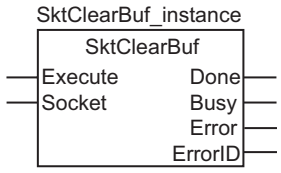
- 하나의 로컬 포트 번호로 복수의 연결을 개설한 경우, 지정한 소켓의 연결만을 클로즈합니다.
- 소켓의 핸들 'Socket.Handle'의 값이 0인 경우는 SktTCPAccept 명령의 연결 대기도 취소됩니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 로컬 IP 주소의 설정 이상이 발생한 경우
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우

샘플 프로그램

🕮 'SktUDPCreate 명령(2-929페이지)' 또는 🕮 'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

SktClearBuf

내장 EtherNet/IP의 지정된 TCP 소켓 또는 UDP 소켓의 수신 버퍼를 클리어합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SktClearBuf	TCP/UDP 소켓 수신 버퍼 클리어	FB		SktClearBuf_instance(Execute, Socket, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Socket	소켓	입력	소켓	–	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Socket	구조체 _sSOCKET에 대한 자세한 내용은 기능 설명 참조																			

기능

'Socket'으로 지정한 소켓의 수신 버퍼를 클리어합니다.
명령이 정상 종료('Done'의 값이 TRUE로 변화)된 시점에서 수신 버퍼의 클리어가 완료됩니다.

'Socket'의 데이터 형식은 구조체 _sSOCKET입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Socket	소켓	소켓	_sSOCKET	–	–	–
Handle	핸들	수신 버퍼를 클리어할 소켓의 핸들	UDINT	데이터 형식에 따름	–	–
SrcAdr(*)	로컬 주소	자신의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		
DstAdr(*)	상대 주소	상대의 IP 주소, 포트 번호	_sSOCKET_ADDRESS	–	–	–
PortNo(*)	포트 번호	포트 번호	UINT	1~65535	–	–
IpAdr(*)	IP 주소	IP 주소 또는 호스트명 단, 호스트명인 경우에는 DNS 설정 또는 Hosts 설정 필수	STRING	데이터 형식에 따름		

* 이러한 멤버는 이 명령에서 사용하지 않습니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE : 사용 가능 FALSE : 사용 불가

참고

소켓 서비스 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 NJ 시리즈 CPU 유니트의 내장 EtherNet/IP에만 사용할 수 있습니다.
- SktUDPCreate 명령, SktUDPRcv 명령, SktUDPSend 명령, SktTCPAccept 명령, SktTCPConnect 명령, SktTCPRcv 명령, SktTCPSend 명령, SktGetTCPStatus 명령, SktClose 명령, SktClearBuf 명령에서 동시 실행 가능한 개수는 최대 32개입니다.

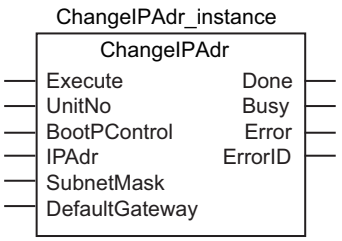
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Socket'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'Socket'으로 지정한 소켓이 존재하지 않는 경우
  - 'Socket.Handle'로 지정한 핸들이 존재하지 않는 경우

샘플 프로그램

'SktTCPConnect 명령(2-945페이지)'의 샘플 프로그램을 참조하십시오.

ChangeIPAdr

내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 IP 주소를 변경합니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
ChangeIPAdr	IP 주소 변경	FB		ChangeIPAdr_instance(Execute, UnitNo, BootPControl, IPAdr, SubnetMask, DefaultGateway, Done, Busy, Error, ErrorID);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
UnitNo	유니트 번호	입력	IP 주소를 변경할 유니트 번호	_CBU_CPU, _CBU_No00~ _CBUNo15	–	_CBU _No00
BootP Control	IP 주소 취득 방법+설정 타이밍		IP 주소 취득 방법과 설정 타이밍	0~2		0
IPAdr[] 배열 *1	IP 주소		IP 주소	*2		–
SubnetMask [] 배열*1	서브넷 마스크		서브넷 마스크			
DefaultGateway[] 배열*1	기본 게이트웨이		기본 게이트웨이			

*1. 요소 번호는 0~3, 요소 수는 4입니다.

*2. 유니트 번호 'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 유효 범위가 달라집니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitNo	열거 형식 _eUnitNo 열거자는 기능 설명 참조																			
BootP Control							○													
IPAdr[] 배열		○																		
	배열 전체를 지정																			
SubnetMask[] 배열		○																		
	배열 전체를 지정																			
DefaultGateway[] 배열		○																		
	배열 전체를 지정																			

기능

유니트 번호 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 IP 주소를 IP 주소 취득 방법+설정 타이밍 'BootPControl'에 따라 변경합니다.
'UnitNo'로 내장 EtherNet/IP를 지정한 경우, 이 명령의 실행이 완료되면 일단 링크 OFF가 되고, 그 후 링크 ON이 됩니다. 링크 ON이 되면 변경된 IP 주소로 통신이 가능해집니다.
'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, 이 명령의 실행이 완료되면 EtherNet/IP 유니트가 재시작됩니다. 재시작이 완료되면 변경된 IP 주소로 통신이 가능해집니다.

이 명령을 사용하여 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 IP 주소를 표시기 등에서 변경할 수 있습니다.

'UnitNo'의 데이터 형식은 열거 형식 _eUnitNo입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_CBU_CPU	내장 EtherNet/IP 포트
_CBU_No00~_CBUNo15	EtherNet/IP 유니트의 유니트 번호 00~15

'BootPControl'의 값에 따라 새로운 IP 주소의 취득 방법과 설정 타이밍이 다음과 같이 결정됩니다.

'BootPControl'의 값	IP 주소 취득 방법	IP 주소 설정 타이밍
0	IP 주소 IPAdr[], 서브넷 마스크 SubnetMask[], 기본 게이트웨이 DefaultGateway[]의 값이 나타내는 IP 주소를 취득	이 명령 실행 시 1회만 설정(고정 설정)
1	BOOTP 서버에서 IP 주소를 취득	이 명령 실행 시 한번 설정하고 그 후에는 컨트롤러의 전원을 ON할 때마다 설정
2	BOOTP 서버에서 IP 주소를 취득	이 명령 실행 시 1회만 설정(고정 설정)

IPAdr[], SubnetMask[], DefaultGateway[]의 요소 번호 0~3에는 IP 주소, 서브넷 마스크, 기본 게이트웨이의 값을 상위부터 순서대로 설정합니다. 예를 들어 IP 주소를 130.58.17.32로 변경하는 경우, IPAdr[0]=BYTE#16#82, IPAdr[1]=BYTE#16#3A, IPAdr[2]=BYTE#16#11, IPAdr[3]=BYTE#16#20입니다.

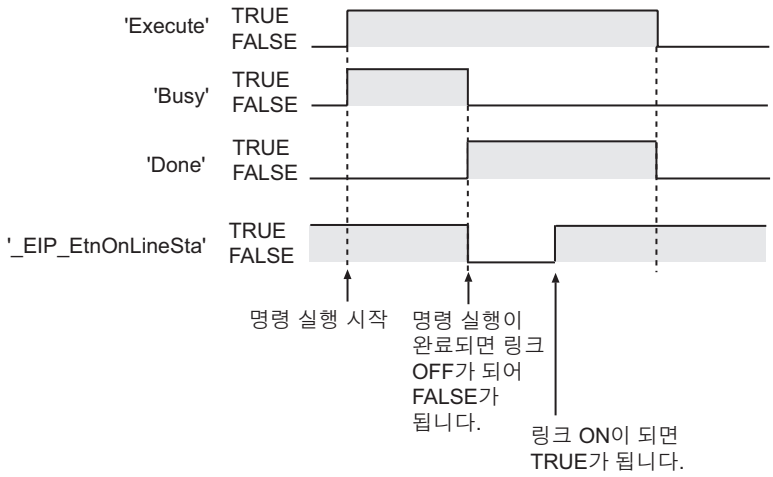
'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 IPAdr[], SubnetMask[], DefaultGateway[]의 값의 유효 범위는 다음과 같습니다. 이들 값의 유효 범위는 'BootPControl'의 값이 0인 경우에만 유효합니다.

'UnitNo'의 지정	입력 변수	유효 범위
내장 EtherNet/IP 포트	IPAdr[] 배열	아래에 나온 IP 주소 이외가 유효 • 127, 0, 255로 시작하는 IP 주소 • 호스트 ID의 전체 비트가 0인 IP 주소 또는 전체 비트가 1인 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255) • AutoIP*1용으로 예약된 IP 주소(169.254.0.0 ~ 169.254.255.255) • USB 포트의 IP 주소(192.168.255.0 ~ 192.168.255.255)
	SubnetMask[] 배열	192.0.0.0~255.255.255.252
	DefaultGateway[] 배열	아래에 나온 IP 주소 이외가 유효 • 127, 0, 255로 시작하는 IP 주소 • 전체 비트가 1인 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255) • AutoIP용으로 예약된 IP 주소(169.254.0.0 ~ 169.254.255.255) • USB 포트의 IP 주소(192.168.255.0 ~ 192.168.255.255)
EtherNet/IP 유니트	IPAdr[] 배열	아래에 나온 IP 주소 이외가 유효 • 127로 시작하는 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255)
	SubnetMask[] 배열	• 0.0.0.0 • 192.0.0.0~255.255.255.252
	DefaultGateway[] 배열	아래에 나온 IP 주소 이외가 유효 • 127로 시작하는 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255)

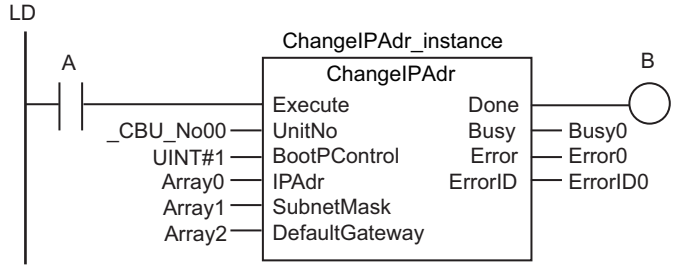
*1. AutoIP는 Windows 98 이상의 OS에 탑재된 IP 주소 자동 할당 기능입니다.

'BootPControl'의 값이 1 또는 2인 경우는 IPAdr[], SubnetMask[], DefaultGateway[]의 값이 무시됩니다. IPAdr[], SubnetMask[], DefaultGateway[]의 값이 유효 범위 밖이어도 상관없습니다.

'UnitNo'로 내장 EtherNet/IP 포트를 지정한 경우는 시스템 정의 변수 '_EIP_EtnOnlineSta'를 사용하여 링크 ON인지 여부를 확인할 수 있습니다. 'Busy'가 TRUE에서 FALSE로 바뀌면 링크 OFF가 되고 '_EIP_EtnOnlineSta'는 FALSE가 됩니다. 그 후 링크 ON이 되면 '_EIP_EtnOnlineSta'는 FALSE에서 TRUE가 됩니다.



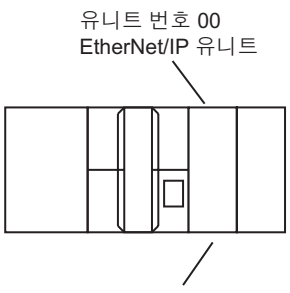
유니트 번호 00인 EtherNet/IP 유니트의 IP 주소를 BOOTP 서버에서 매회 취득하여 변경하는 경우의 기술 예가 아래에 나와 있습니다. 'Execute'=A를 표시기 등에서 ON으로 하여 BOOTP 서버에서 취득한 IP 주소로 변경합니다. 그 후에는 전원을 ON할 때 마다 BOOTP 서버에서 IP 주소를 취득하여 변경합니다.



ST

```
ChangeIPAdr_instance(A,_CBU_No00,UINT#1,Array0,Array1,Array2,B,Busy0,Error0,ErrorID0);
```

유니트 번호 00인 EtherNet/IP 유니트에 BOOTP 서버에서 취득한 IP 주소를 설정합니다.
그 후에는 전원을 ON할 때 마다 BOOTP 서버에서 취득한 IP 주소를 다시 설정합니다.



관련된 시스템 정의 변수

변수명	이름	데이터형식	내용
_EIP_EtnOnlineSta	온라인	BOOL	내장 EtherNet/IP 포트의 통신 기능의 상태 TRUE: 사용 가능 FALSE: 사용 불가능

참고

- 'UnitNo'로 내장 EtherNet/IP를 지정한 경우, 이 명령을 실행하면 다음 항목이 이벤트 로그에 기록됩니다.
  - 링크 OFF 감지
  - IP 주소 결정
- CPU 유니트의 쓰기 방지가 유효인 경우에도 이 명령으로 IP 주소를 변경할 수 있습니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'UnitNo'로 내장 EtherNet/IP를 지정한 경우, 이 명령의 실행이 완료되면 내장 EtherNet/IP는 일단 링크 OFF가 됩니다. 링크 OFF가 되어도 설비에 지장이 없는지 확인한 후 이 명령을 실행하십시오.
- 'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, 이 명령의 실행이 완료되면 EtherNet/IP 유니트가 재시작됩니다. 재시작되어도 설비에 지장이 없는지 확인한 후 이 명령을 실행하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'UnitNo', 'BootPControl'의 값 중 하나가 유효 범위 밖
  - 'BootPControl'의 값이 0이고 IPAdr[], SubnetMask[], DefaultGateway[]의 값 중 하나가 유효 범위 밖
  - 'UnitNo'로 지정한 EtherNet/IP 유니트의 유니트 번호가 EtherNet/IP 유니트 이외에서 사용되는 경우
  - 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트가 재시작 중
  - 사용자 프로그램 또는 CIP 메시지 등에 의해 IP 주소, FTP 설정, NTP 서버 수신처를 변경하는 중
  - 'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, EtherNet/IP 유니트의 재시작에 실패
  - Network Configurator에서 태그 데이터 링크 설정을 다운로드 중

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

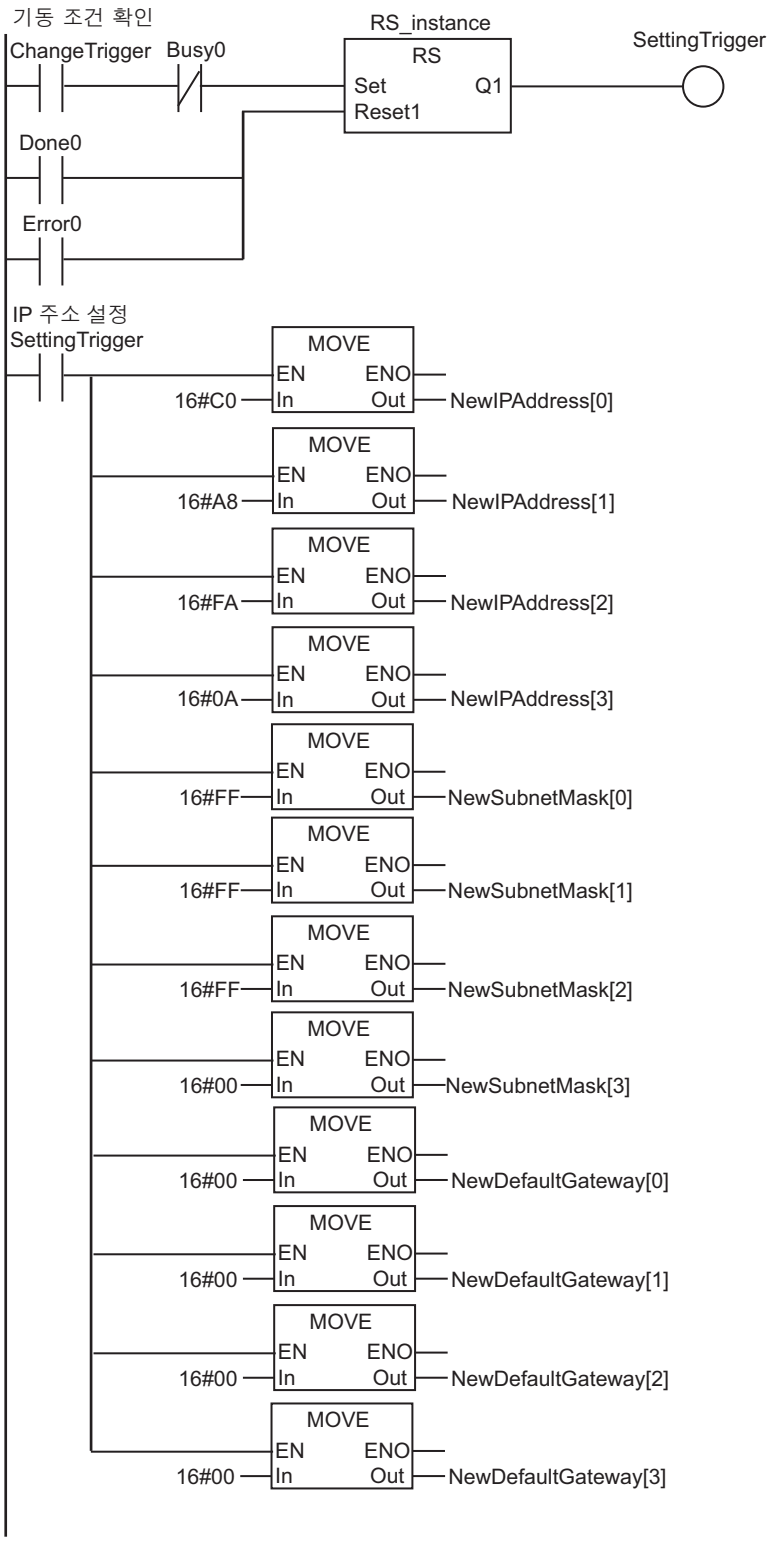
샘플 프로그램

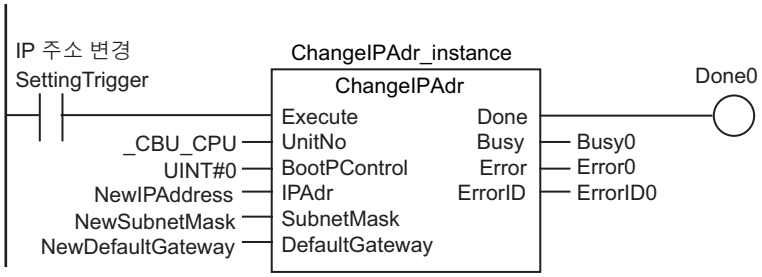
내장 EtherNet/IP 포트의 IP 주소를 다음과 같은 IP 주소의 고정 설정으로 변경합니다.

항목	내용
IP 주소	192.168.250.10
서브넷 마스크	255.255.255.0
기본 게이트웨이	0.0.0.0

LD

이름	데이터 형식	초기값	코멘트
ChangeTrigger	BOOL	False	변경 플래그
SettingTrigger	BOOL	False	IP 주소 변경 중 플래그
Done0	BOOL	False	IP 주소 변경 완료
Error0	BOOL	False	IP 주소 변경 이상
Busy0	BOOL	False	IP 주소 변경 중
ErrorID0	WORD	16#0	IP 주소 변경 이상 ID
NewIPAddress	ARRAY[0..3] OF BYTE	[4(16#0)]	IP 주소
NewSubnetMask	ARRAY[0..3] OF BYTE	[4(16#0)]	서브넷 마스크
NewDefaultGateway	ARRAY[0..3] OF BYTE	[4(16#0)]	기본 게이트웨이
RS_instance	RS		
ChangeIPAdr_instance	ChangeIPAdr		


기동 조건 확인
ChangeTrigger
Busy0
RS_instance
RS
Set
Reset1
Q1
SettingTrigger
Done0
Error0
IP 주소 설정
SettingTrigger
MOVE
EN
ENO
16#C0
In
Out
NewIPAddress[0]
16#A8
NewIPAddress[1]
16#FA
NewIPAddress[2]
16#0A
NewIPAddress[3]
16#FF
NewSubnetMask[0]
NewSubnetMask[1]
NewSubnetMask[2]
16#00
NewSubnetMask[3]
NewDefaultGateway[0]
NewDefaultGateway[1]
NewDefaultGateway[2]
NewDefaultGateway[3]


IP 주소 변경
SettingTrigger
ChangeIPAdr_instance
ChangeIPAdr
Execute
Done
Done0
_CBU_CPU
UnitNo
Busy
Busy0
UINT#0
BootPControl
Error
Error0
NewIPAddress
IPAdr
ErrorID
ErrorID0
NewSubnetMask
SubnetMask
NewDefaultGateway
DefaultGateway

ST

이름	데이터 형식	초기값	코멘트
ChangeTrigger	BOOL	False	변경 플래그
SettingTrigger	BOOL	False	IP 주소 변경 중 플래그
Done0	BOOL	False	IP 주소 변경 완료
Error0	BOOL	False	IP 주소 변경 이상
Busy0	BOOL	False	IP 주소 변경 중
ErrorID0	WORD	16#0	IP 주소 변경 이상 ID
NewIPAddress	ARRAY[0..3] OF BYTE	[4(16#0)]	IP 주소
NewSubnetMask	ARRAY[0..3] OF BYTE	[4(16#0)]	서브넷 마스크
NewDefaultGateway	ARRAY[0..3] OF BYTE	[4(16#0)]	기본 게이트웨이
RS_instance	RS		
ChangeIPAdr_instance	ChangeIPAdr		

```
// 기동 조건 확인
IF((ChangeTrigger=TRUE)AND(Busy0=FALSE))THEN
  SettingTrigger:= TRUE;
END_IF;

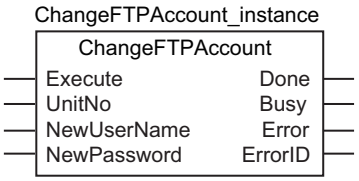
IF((Done0=TRUE)OR(Error0=TRUE))THEN
  SettingTrigger:= FALSE;
END_IF;

// IP 주소 설정
IF(SettingTrigger=TRUE)THEN
  NewIPAddress[0]       := 16#C0;
  NewIPAddress[1]       := 16#A8;
  NewIPAddress[2]       := 16#FA;
  NewIPAddress[3]       := 16#0A;
  NewSubnetMask[0]      := 16#FF;
  NewSubnetMask[1]      := 16#FF;
  NewSubnetMask[2]      := 16#FF;
  NewSubnetMask[3]      := 16#00;
  NewDefaultGateway[0]  := 16#00;
  NewDefaultGateway[1]  := 16#00;
  NewDefaultGateway[2]  := 16#00;
  NewDefaultGateway[3]  := 16#00;
END_IF;

// IP 주소 변경
ChangeIPAdr_instance(
      Execute           := SettingTrigger,
      UnitNo            := _CBU_CPU,
      BootPControl      := UINT#0,
      IPAdr             := NewIPAddress,
      SubnetMask        := NewSubnetMask,
      DefaultGateway    := NewDefaultGateway,
      Done              =>Done0,
      Busy              =>Busy0,
      Error             =>Error0,
      ErrorID           =>ErrorID0);
```

ChangeFTPAccount

내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 FTP 로그인명과 비밀번호를 변경합니다

명령	이름	FB/ FUN	그래픽 표현	ST 표현
ChangeFTP Account	FTP 계정 변경	FB		ChangeFTPAccount_instance(Execute, UnitNo, NewUserName, NewPassword, Done, Busy, Error, ErrorID);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
UnitNo	유니트 번호	입력	FTP의 로그인명과 비밀번호를 변경할 유니트 번호	_CBU_CPU, _CBU_No00~ _CBUNo15	–	_CBU _No00
New UserName	로그인명		로그인명	반각 영숫자 1~12문자(대문자와 소문자의 구별 있음)		–
New Password	비밀번호		비밀번호	*1		

*1. 유니트 번호 'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 유효 범위가 달라집니다. 자세한 내용은 기능 설명을 참조하십시오.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitNo	열거 형식 _eUnitNo 열거자는 기능 설명 참조																			
New UserName																				○
New Password																				○

기능

유니트 번호 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 FTP 로그인명 'NewUserName'과 비밀번호 'NewPassword'를 변경합니다.
'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, 이 명령의 실행이 완료되면 EtherNet/IP 유니트가 재시작됩니다. 재시작이 완료되면 변경된 로그인명과 비밀번호를 사용할 수 있습니다.

이 명령을 사용하여 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 FTP 로그인명과 비밀번호를 표시기 등에서 변경할 수 있습니다.

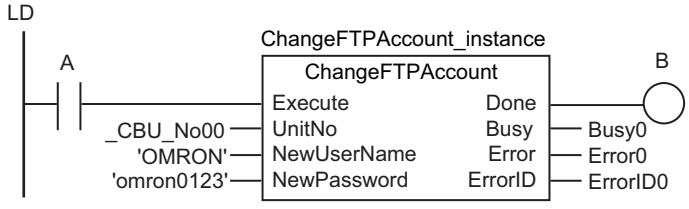
'UnitNo'의 데이터 형식은 열거 형식 _eUnitNo입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_CBU_CPU	내장 EtherNet/IP 포트
_CBU_No00~_CBUNo15	EtherNet/IP 유니트의 유니트 번호 00~15

'NewPassword' 값의 유효 범위는 'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 다음과 같습니다.

'UnitNo'의 지정	유효 범위
내장 EtherNet/IP 포트	반각 영숫자 8~32문자(대문자와 소문자의 구별 있음)
EtherNet/IP 유니트	반각 영숫자 1~8문자(대문자와 소문자의 구별 있음)

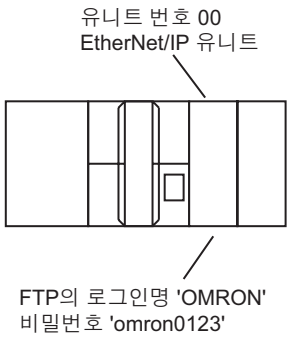
유니트 번호 00인 EtherNet/IP 유니트의 FTP 로그인명과 비밀번호를 변경하는 경우의 기술 예가 아래에 나와 있습니다. 'Execute'=A를 표시기 등에서 ON으로 하여 FTP 로그인명을 'OMRON'으로, 비밀번호를 'omron0123'으로 변경합니다.



ST

```
ChangeFTPAccount_instance(A,_CBU_No00,'OMRON','omron0123',B,Busy0,Error0,ErrorID0);
```

유니트 번호 00인 EtherNet/IP 유니트의 FTP 로그인명을 'OMRON'으로,
비밀번호를 'omron0123'으로 변경합니다.



참고

- 이 명령을 사용하면 CPU 유니트의 쓰기 방지가 유효일 때에도 FTP의 로그인명과 비밀번호를 변경할 수 있습니다.
- FTP 서버에 연결하여 파일을 전송하고 있을 때 이 명령을 사용하여 FTP의 로그인명과 비밀번호를 변경해도 파일 전송은 속행됩니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.

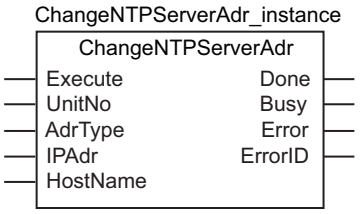
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'UnitNo', 'NewUserName', 'NewPassword'의 값 중 하나가 유효 범위 밖
  - 'UnitNo'로 지정한 EtherNet/IP 유니트의 유니트 번호가 EtherNet/IP 유니트 이외에서 사용되는 경우
  - 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트가 재시작 중
  - 사용자 프로그램 또는 CIP 메시지 등에 의해 IP 주소, FTP 설정, NTP 서버 수신처를 변경하는 중
  - 'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, EtherNet/IP 유니트의 재시작에 실패

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

ChangeNTPServerAdr

내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 NTP 서버 수신처를 변경합니다.

명령	이름	FB/ FUN	그래픽 표현	ST 표현
ChangeNTP ServerAdr	NTP 서버 수신처 변경	FB		ChangeNTPServerAdr_instance(Execute, UnitNo, AdrType, IPAdr, HostName, Done, Busy, Error, ErrorID);

변수

	이름	입/출력	내용	유효 범위	단위	초기값
UnitNo	유니트 번호	입력	NTP 서버의 수신처를 변경할 유니트 번호	_CBU_CPU, _CBU_No00~ _CBUNo15	–	_CBU_ No00
AdrType	서버 설정 방법		NTP 서버 수신처의 설정 방법 TRUE : IP 주소 FALSE : 호스트 이름	데이터 형식에 따름		FALSE
IPAdr[] 배열 *1	IP 주소		NTP 서버 수신처의 IP 주소	*2		–
HostName	호스트 이름		NTP 서버 수신처의 호스트 이름	1~200문자의 반각 영숫자와 "-"(하이픈), "."(점)*3		

*1. 요소 번호는 0~3, 요소 수는 4입니다.

*2. 유니트 번호 'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 유효 범위가 달라집니다. 자세한 내용은 기능 설명을 참조하십시오.

*3. "."(점)과 "."(점) 사이는 반각 영숫자 1~63문자입니다. 'HostName'의 유효 범위는 서버 설정 방법 'AdrType'이 FALSE인 경우에만 유효합니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
UnitNo	열거 형식 _eUnitNo 열거자는 기능 설명 참조																			
AdrType	○																			
IPAdr[] 배열		○																		
	배열 전체를 지정																			
HostName																				○

기능

유니트 번호 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 NTP 서버 수신처를 변경합니다.
서버 설정 방법 'AdrType'이 TRUE일 때는 NTP 서버 수신처를 IP 주소 IPAdr[]로 변경합니다. 'AdrType'이 FALSE일 때는 NTP 서버 수신처를 호스트 이름 'HostName'으로 변경합니다.

'UnitNo'로 EtherNet/IP 유니트를 지정한 경우, 이 명령의 실행이 완료되면 EtherNet/IP 유니트가 재시작됩니다. 재시작이 완료되면 변경된 NTP 서버 수신처를 사용할 수 있습니다.

이 명령을 사용하여 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트의 NTP 서버 수신처를 표시기 등에서 변경할 수 있습니다.

'UnitNo'의 데이터 형식은 열거 형식 _eUnitNo입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_CBU_CPU	내장 EtherNet/IP 포트
_CBU_No00~_CBUNo15	EtherNet/IP 유니트의 유니트 번호 00~15

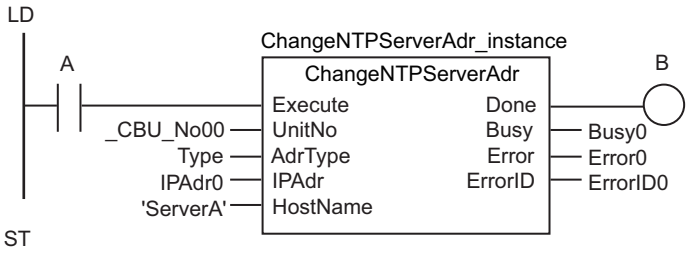
IPAdr[]의 요소 번호 0~3에는 IP 주소의 값을 상위부터 순서대로 설정합니다. 예를 들어 NTP 서버 수신처를 IP 주소 130.58.17.32로 변경하는 경우는 IPAdr[0]=BYTE#16#82, IPAdr[1]=BYTE#16#3A, IPAdr[2]=BYTE#16#11, IPAdr[3]=BYTE#16#20입니다.

IPAdr[] 값의 유효 범위는 'UnitNo'로 내장 EtherNet/IP 포트를 지정할지, EtherNet/IP 유니트를 지정할지에 따라 다음과 같습니다. 이들 값의 유효 범위는 'AdrType'의 값이 TRUE인 경우에만 유효합니다.

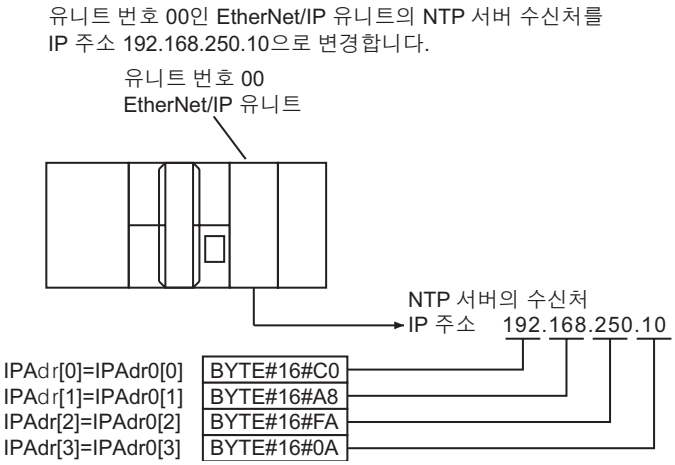
'UnitNo'의 지정	유효 범위
내장 EtherNet/IP 포트	아래에 나온 IP 주소 이외가 유효 • 127, 0, 255로 시작하는 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255) • AutoIP*1용으로 예약된 IP 주소(169.254.0.0 ~ 169.254.255.255) • USB 포트의 IP 주소(192.168.255.0 ~ 192.168.255.255)
EtherNet/IP 유니트	아래에 나온 IP 주소 이외가 유효 • 127로 시작하는 IP 주소 • 클래스 D의 IP 주소(224.0.0.0 ~ 239.255.255.255) • 클래스 E의 IP 주소(240.0.0.0 ~ 255.255.255.255)

*1. AutoIP는 Windows 98 이상의 OS에 탑재된 IP 주소 자동 할당 기능입니다.

유니트 번호 00인 EtherNet/IP 유니트의 NTP 서버 수신처를 변경하는 경우의 기술 예가 아래에 나와 있습니다. 'Execute'=A를 표시기 등에서 ON으로 하여 NTP 서버 수신처를 변경합니다.
예를 들어, IPAdr0[0]=BYTE#16#C0, IPAdr0[1]=BYTE#16#A8, IPAdr0[2]=BYTE#16#FA, IPAdr0[3]=BYTE#16#0A로 가정합니다. 'AdrType'=Type이 TRUE인 경우, NTP 서버 수신처를 IP 주소 192.168.250.10으로 변경합니다. 'AdrType'=Type이 FALSE인 경우, NTP 서버 수신처를 호스트 이름 ServerA로 변경합니다.




```
ChangeNTPServerAdr_instance(A,_CBU_No00,Type,IPAdr0,'ServerA',B,Busy0,Error0,ErrorID0);
```



참고

- 이 명령을 사용하면 CPU 유니트의 쓰기 방지가 유효인 경우에도 NTP 서버의 수신처를 변경할 수 있습니다.
- 변경할 NTP 서버의 NTP 동작 타이밍이 지정 시간 간격인 경우, 이 명령 실행 완료 시점부터 지정 시간 간격을 계산합니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되는 것으로 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'UnitNo'의 값이 유효 범위 밖
  - 'AdrType'이 TRUE이고 IPAdr[]의 값이 유효 범위 밖
  - 'AdrType'이 FALSE이고 'HostName'의 값이 유효 범위 밖
  - 'UnitNo'로 지정한 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트가 재시작 중
  - 'UnitNo'로 지정한 EtherNet/IP 유니트의 유니트 번호가 EtherNet/IP 유니트 이외에서 사용되는 경우

- 사용자 프로그램 또는 CIP 메시지 등에 의해 IP 주소, FTP 설정, NTP 서버 수신처를 변경하는 중

버전 관련 정보

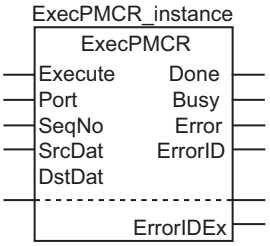
이 명령은 CPU 유니트 Ver.1.02 이상, Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다.

시리얼 통신 명령

명령	명칭	페이지
ExecPMCR	프로토콜 매크로	2-986
SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	2-1000
SerialRcv/ SerialRcvNoClear	시리얼 커뮤니케이션 유니트 시리얼 포트 입력/수신 버퍼 삭제 안 함	2-1011
SendCmd	명령 송신	2-1026

ExecPMCR

시리얼 커뮤니케이션 유니트에 등록된 송수신 시퀀스(프로토콜 데이터)의 실행을 의뢰합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ExecPMCR	프로토콜 매크로	FB		ExecPMCR_instance(Execute, Port, SeqNo, SrcDat, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Port	상대방 포트 지정	입력	상대방 포트 지정	–	–	–
SeqNo	송수신 시퀀스 번호		송수신 시퀀스 번호	0~999		0
SrcDat[] 배열	송신 데이터 배열		송신 데이터 배열	데이터 형식에 따름		(*)
DstDat[] 배열	수신 데이터 배열	입출력	수신 데이터 배열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Port	구조체 _sPORT에 대한 자세한 내용은 기능 설명 참조																			
SeqNo							○													
SrcDat[] 배열			○																	
DstDat[] 배열			○																	

기능

송수신 시퀀스 번호 'SeqNo'로 지정한 시퀀스를 지정된 상대방 포트 'Port'에서 실행할 것을 의뢰합니다.

송신의 경우, 송신 데이터 배열 SrcDat[]의 2번째 요소(SrcDat[1])부터 송신을 수행합니다. 송신할 배열 요소 수는 SrcDat[0]으로 지정합니다.

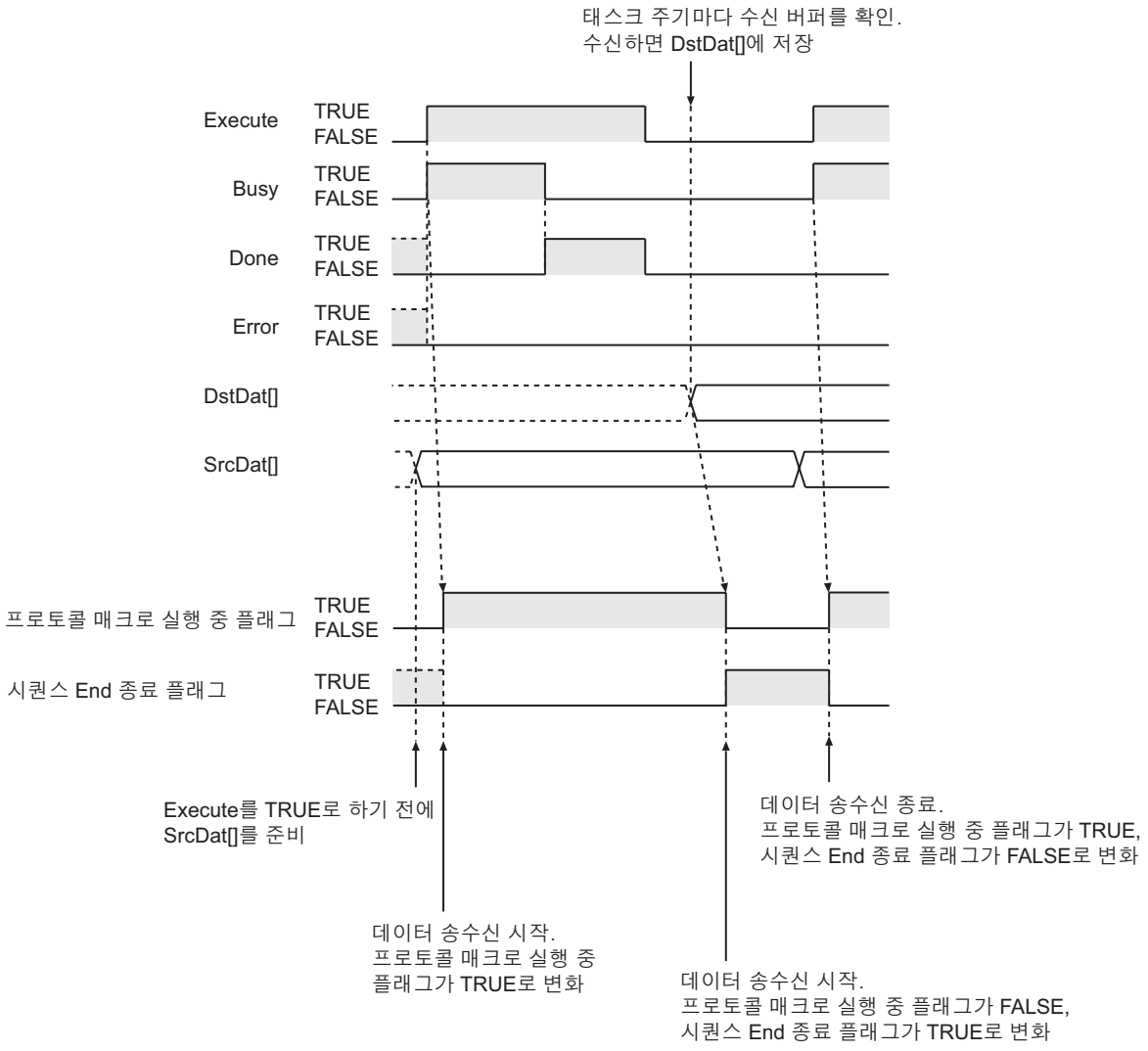
수신을 수행하여 성공한 경우, 수신 데이터 배열 DstDat[]의 2번째 요소(DstDat[1]) 이후에 수신 데이터가 저장됩니다. 수신 데이터 요소 수는 DstDat[0]에 저장됩니다.

수신에 실패한 경우, 미리 DstDat[0]에 저장해둔 요소 수만큼 이 명령 실행 전의 DstDat[1] 이후의 값이 저장됩니다.

상대방 포트 지정 'Port'의 데이터 형식은 구조체 _sPORT입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Port	상대방 포트 지정	상대방 포트 지정	_sPORT	–	–	–
UnitNo	유니트 번호	시리얼 커뮤니케이션 유니트의 유니트 번호	_eUnitNo	_CBU_No00~ _CBU_No15	–	_CBU _No00
PhysicPort No	시리얼 포트 번호	시리얼 커뮤니케이션 유니트의 시리얼 포트 번호	USINT	1,2		1

타이밍 차트가 아래에 나와 있습니다. 'Done'의 값이 TRUE가 된 후에도 통신은 끝까지 수행됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Port_numUsingPort	사용 포트 수	USINT	현재 사용 중인 포트 수
_Port_isAvailable	네트워크 통신 명령 실행 가능 플래그	BOOL	TRUE : 사용 가능 포트 있음 FALSE : 사용 가능 포트 없음
_CJB_SCU##P1ChgSta, _CJB_SCU##P2ChgSta (*)	시리얼 커뮤니케이션 유니트 ##호기 포트 1/2 용 시리얼 포트 설정 변경 중 플래그	BOOL	TRUE : 시리얼 포트 설정 변경 중 FALSE : 시리얼 포트 설정 변경 중이 아님

* #에는 시리얼 커뮤니케이션 유니트의 유니트 번호가 들어갑니다.

관련된 준 사용자 정의 변수

변수명	명칭	데이터 형식	내용
P#_PmrExecSta(*)	프로토콜 매크로 실행 중 플래그	BOOL	TRUE : 실행 중 FALSE : 비실행 중 또는 실행 실패
P#_PmrSeqEndSta(*)	시퀀스 End 종료 플래그	BOOL	TRUE : 시퀀스 End 종료 FALSE : 시퀀스 End 종료가 아님
P#_PmrSeqAbtSta(*)	시퀀스 Abort 종료 플래그	BOOL	TRUE : 시퀀스 Abort 종료 FALSE : 시퀀스 Abort 종료가 아님

* #에는 시리얼 커뮤니케이션 유니트의 포트 번호가 들어갑니다.

참고

프로토콜 매크로 기능에 대한 자세한 내용은 "SYSMAC CX-Protocol 조작 설명서(SBCA-307)"를 참조하십시오.

사용상의 주의 사항

- 이 명령은 프로토콜 매크로를 실행 기동시키는 것입니다. 프로토콜 매크로 실행 상태는 프로토콜 매크로 실행 중 플래그의 시스템 정의 변수 'P#PmrExecSta'로 확인하십시오.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- DstDat[]는 CJ 유니트 메모리의 주소에 AT 지정되어 있어야 합니다.
- 직접 지정, 링크 채널 지정인 경우에는 SrcDat[0] 및 DstDat[0]의 값을 0으로 하십시오. 그 이외의 정수 및 변수를 설정하면 이상 상태가 되어 이 명령이 실행되지 않습니다.
- DstDat[0]의 값이 0 또는 1인 경우, 수신에 실패하면 DstDat[]의 요소의 값은 모두 0이 됩니다.
- 이 명령은 포트가 사용 가능한 경우에만 실행할 수 있습니다. 따라서 이 명령의 입력 조건에 네트워크 통신 명령 실행 가능 플래그의 시스템 정의 변수 '_Port_isAvailable'을 a 접점으로 삽입하십시오.
- 이 명령은 'Busy'의 값이 TRUE인 동안에는 실행할 수 없습니다. 따라서 이 명령의 입력 조건으로 'Busy'를 b 접점으로 삽입하십시오.
- 프로토콜 매크로 실행 중 플래그의 준 사용자 정의 변수 'P#_PmrExecSta'는 이 명령 실행 시작 시에 TRUE가 되며, 송수신 시퀀스의 실행이 종료되어 DstDat[]에 수신 데이터가 완전히 저장된 시점에서 FALSE가 됩니다. 그 동안, 동일한 시리얼 포트에 대해 이 명령을 실행할 수 없습니다. 따라서 이 명령의 입력 조건으로 'P#_PmrExecSta'를 b 접점으로 삽입하십시오.
- 이 명령을 ST 프로그램으로 기술하는 경우, 이 명령 실행 중에는 태스크 주기마다 이 명령의 제어가 수행되도록 하십시오. 그렇게 하지 않으면 정상적인 처리를 할 수 없는 경우가 있습니다.

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 시리얼 통신 모드가 프로토콜 매크로 모드 이외일 때 이 명령이 실행된 경우
  - '_Port_isAvailable'의 값이 FALSE인 경우
  - 'SeqNo'의 값이 유효 범위 밖인 경우
  - 'SeqNo'의 값이 시리얼 커뮤니케이션 유니트에 등록되지 않은 경우
  - 'Port.UnitNo' 또는 'Port.PhysicPortNo'의 값이 유효 범위 밖인 경우
  - 지정한 유니트 번호에 CJ 시리얼 커뮤니케이션 유니트가 장착되지 않은 경우
  - SrcDat[0]의 값이 SrcDat[]의 크기를 초과한 경우
  - DstDat[0]의 값이 DstDat[]의 크기를 초과한 경우
  - SrcDat[0] 또는 DstDat[0]의 값이 250CH를 초과한 경우
  - 통신이 실패한 경우
  - DstDat[]는 CJ 유니트 메모리의 주소에 AT 지정되지 않은 경우

- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 통신 응답 코드로서의 의미를 갖습니다. 그 값과 이상 내용은 다음과 같습니다. 'ErrorIDEx'는 에러 코드 'ErrorID'의 값이 WORD#16#0800인 경우에 출력됩니다.

값	이상 내용	대처 방법
16#00000001	통신 서비스를 중단했습니다.	• 데이터 링크 기동 상태를 확인하십시오. • 제3노드의 전송처 영역의 용량을 확인하십시오.
16#00000101	로컬 노드가 네트워크에 가입하지 않았습니다.	로컬 노드를 네트워크에 가입시키십시오.
16#00000102	토큰 타임아웃이 발생했습니다.	로컬 노드를 최대 노드 주소 범위 내로 설정하십시오.
16#00000103	재전송 초과가 발생했습니다.	노드간 테스트를 실행하여 정상이 아닌 경우, 사용 환경을 확인하십시오.
16#00000104	송신 허가 프레임 수 초과가 발생했습니다.	네트워크 내에서 실행되고 있는 이벤트의 상황을 확인하고 1태스크 주기 내의 이벤트 수를 줄이십시오. 또는 송신 허가 프레임 수를 늘리십시오.
16#00000105	로컬 노드의 노드 주소가 설정 범위 밖입니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000106	로컬 노드의 노드 주소가 네트워크 내에서 중복되어 있습니다.	중복되어 있는 노드의 주소를 변경하십시오.
16#00000201	상대 노드가 네트워크에 가입하지 않았습니다.	상대 노드를 네트워크에 가입시키십시오.
16#00000202	송신대상으로 지정한 호기 주소의 유니트가 존재하지 않습니다.	송신대상 네트워크 주소의 호기 주소를 정확하게 설정하십시오.
16#00000203	제3노드가 네트워크에 가입하지 않았습니다.	• 제3노드로 지정된 유니트의 주소를 확인하십시오. • 제3노드에는 1노드만 지정하십시오.
16#00000204	상대 노드가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 상대 노드에 집중되지 않도록 시스템을 재검토하십시오.
16#00000205	응답 타임아웃이 발생했습니다.	통신 파라메터의 설정을 확인하십시오.
16#00000206	전송로에 이상이 있습니다.	• 다시 시도하십시오. • 이상이 자주 발생하는 경우는 노이즈의 상황을 확인하십시오.
16#00000301	통신 컨트롤러 이상이 발생했습니다.	해당 유니트의 사용 설명서를 참조하여 적절하게 처리하십시오.
16#00000302	상대 노드의 CPU 유니트에 이상이 있습니다.	상대 노드의 CPU 유니트 설명서를 참조하여 이상을 제거하십시오.
16#00000303	해당 컨트롤러에 이상이 있어서 응답이 반환되지 않습니다.	네트워크의 통신 상황을 확인한 후 해당 컨트롤러를 재기동하십시오. 그래도 이상이 발생하는 경우는 해당 컨트롤러를 교환하십시오.
16#00000304	유니트 번호 설정이 잘못되었습니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000401	송신한 명령이 지원되지 않습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00000402	유니트의 기종 또는 버전이 지원되지 않습니다.	유니트의 기종과 버전을 확인하십시오.
16#00000501	상대 주소 설정 이상입니다.	라우팅 테이블에 수신대상 주소를 설정하십시오.
16#00000502	라우팅 테이블이 등록되지 않았습니다.	송신원 노드, 수신대상 노드, 중계 노드에 라우팅 테이블을 설정하십시오.
16#00000503	라우팅 테이블 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.

값	이상 내용	대처 방법
16#00000504	중계 횟수 초과가 발생했습니다.	네트워크를 재구성하거나 라우팅 테이블을 재검토하여 명령의 사용 범위를 3계층 이내로 하십시오.
16#00001001	명령 길이가 최대 명령 길이를 초과했습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001002	명령 길이가 최소 명령 길이 미만입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001003	명령에 지정된 쓰기 요소 수와 실제 쓰기 데이터 수가 일치하지 않습니다.	쓰기 요소 수와 실제 쓰기 데이터 수를 일치시키십시오.
16#00001004	명령 형식 이상입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001005	헤더 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.
16#00001101	영역 종류가 없습니다.	명령의 변수, 파라메터 종류 코드를 참조한 후 해당하는 코드를 설정하십시오.
16#00001102	액세스 크기 이상입니다.	변수, 파라메터의 액세스 크기를 정확하게 설정하십시오.
16#00001103	주소 범위 밖이 지정되었습니다.	처리 가능 범위 내의 주소를 지정하십시오.
16#00001104	주소 범위를 초과했습니다.	• 처리 가능 범위 내의 주소를 지정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#00001106	등록되지 않은 송수신 시퀀스 번호가 지정되었습니다.	송수신 시퀀스 번호를 수정하거나 CX-Protocol로 시퀀스를 추가하십시오.
16#00001109	상관 관계 이상이 발생했습니다.	• 명령 데이터의 대소 관계를 정확하게 설정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110A	데이터가 중복되어 있습니다.	• 실행 중인 처리를 중단하거나 종료할 때까지 기다린 후 명령을 실행하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110B	응답이 최대 응답 길이를 초과했습니다.	명령 배열 내의 요소 수를 정확하게 설정하십시오.
16#0000110C	기타 파라메터 이상	명령 배열의 내용을 정확하게 설정하십시오.
16#00002002	보호 중입니다.	보호를 해제한 후 명령을 다시 실행하십시오.
16#00002003	등록 테이블이 없습니다.	테이블을 정확하게 설정하십시오.
16#00002004	검색 데이터와 일치하는 데이터가 존재하지 않습니다.	검색 데이터를 정확하게 설정하십시오.
16#00002005	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002006	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002007	비교 이상이 발생했습니다.	• 메모리 내용을 확인하고 정확한 데이터로 재기입하십시오. • 파일 내용을 확인하십시오.
16#00002101	읽기 전용 영역이므로 액세스할 수 없습니다.	쓰기 방지를 해제한 후 명령을 다시 실행하십시오.

값	이상 내용	대처 방법
16#00002102	보호 중 또는 데이터 링크 테이블 쓰기 불가능 상태입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 시스템 설정을 데이터 링크 테이블에 설정하십시오.
16#00002103	등록할 수 없습니다.	• 필요 없는 파일을 삭제한 후 파일을 작성하거나 새 파일용 메모리를 준비하십시오. • 열려 있는 파일을 닫은 후 명령을 다시 실행하십시오.
16#00002105	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002106	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002107	동일한 파일명이 존재합니다.	쓸 파일의 파일명을 변경하고 명령을 다시 실행하십시오.
16#00002108	변경에 의해 이상이 발생하므로 변경할 수 없습니다.	설정을 변경하십시오.
16#00002201	이미 프로토콜 매크로를 실행 중이므로 동작 불가능합니다.	프로토콜 매크로 실행 중 플래그를 b 접점에 설정하십시오.
16#00002202	동작 모드 이상입니다.	동작 모드를 확인하십시오.
16#00002203	명령의 동작 모드가 다릅니다(프로그램 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002204	명령의 동작 모드가 다릅니다(디버그 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002205	명령의 동작 모드가 다릅니다(모니터 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002206	명령의 동작 모드가 다릅니다(운전 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002207	지정 노드가 관리국이 아닙니다.	네트워크의 관리국 노드를 확인하십시오.
16#00002208	명령의 동작 모드가 다릅니다.	스텝 활성 상태를 확인하십시오.
16#00002211	유니트가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 해당 유니트에 집중되지 않도록 시스템을 재검토하십시오.
16#00002301	파일 장치가 없습니다.	미디어를 장착하십시오. 또는 EM 포맷을 실시하십시오.
16#00002302	파일 메모리가 없습니다.	파일 메모리 장착을 확인하십시오.
16#00002303	시계가 내장되어 있지 않습니다.	기종의 사양을 확인하십시오.
16#00002401	프로토콜 매크로 데이터에서 SUM 값 이상이 발생했거나 데이터 전송 중입니다.	CX-Protocol로 프로토콜 매크로 데이터를 다시 전송하십시오.
16#00002502	처리 대상 메모리에 이상이 있습니다.	대상 메모리에 정확한 데이터를 다시 전송하십시오.
16#00002503	등록된 I/O 유니트 구성과 실제 유니트의 구성이 서로 다릅니다	I/O 유니트 구성을 확인하십시오.
16#00002504	등록된 I/O 점수 또는 리모트 I/O 점수가 최대값을 초과했습니다.	I/O 점수, 리모트 I/O 점수를 정확하게 설정하십시오.
16#00002505	CPU 유니트와 CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#00002506	랙 번호, 유니트 번호, I/O 주소 중 하나의 설정이 중복되어 있습니다.	중복된 내용을 다시 설정하십시오.
16#00002507	CPU 유니트와 I/O 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.

값	이상 내용	대처 방법
16#00002509	SYSMAC BUS/2 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250A	CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250D	채널 설정이 중복되어 있습니다.	I/O 채널을 정확하게 설정하십시오.
16#0000250F	메모리 이상입니다.	• 내부 메모리의 경우, 정확한 데이터를 쓰고 명령을 다시 실행하십시오. • 메모리 카드나 EM 파일 메모리의 경우, 확장 메모리의 포맷 명령을 실행하십시오. • 상기의 처리를 실시해도 이상이 해제되지 않는 경우는 메모리를 교환하십시오.
16#00002510	종단국 설정 이상입니다.	종단국을 정확하게 설정하십시오.
16#00002601	보호 해제 중입니다.	보호를 해제할 필요가 없습니다.
16#00002602	비밀번호가 일치하지 않습니다.	정확한 비밀번호를 지정하십시오.
16#00002604	보호 중입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002605	서비스 실행 중입니다.	실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002606	서비스 정지 중입니다.	필요에 따라 해당 서비스를 실행하십시오.
16#00002607	실행 권한이 없습니다.	• 데이터 링크 참가 노드에서 실행하십시오. • 재기동해도 이상이 재발하는 경우는 컨트롤러를 교환하십시오.
16#00002608	환경이 설정되지 않았습니다.	필요한 설정을 하십시오.
16#00002609	필요한 항목이 설정되지 않았습니다.	필요한 항목을 설정하십시오.
16#0000260A	지정 번호의 정의가 완료되었습니다.	등록되지 않은 작업이나 트랜지션 번호로 변경하고 명령을 다시 실행하십시오.
16#0000260B	이상 해제가 불가능합니다.	이상의 원인이 된 요인을 해제하고 이상 해제 명령을 실행하십시오.
16#00003001	액세스 권한이 없습니다.	액세스 권한이 해제될 때까지 기다린 후 명령을 다시 실행하십시오.
16#00004001	서비스가 중단되었습니다.	서비스가 중단된 요인을 해제하고 명령을 다시 실행하십시오.

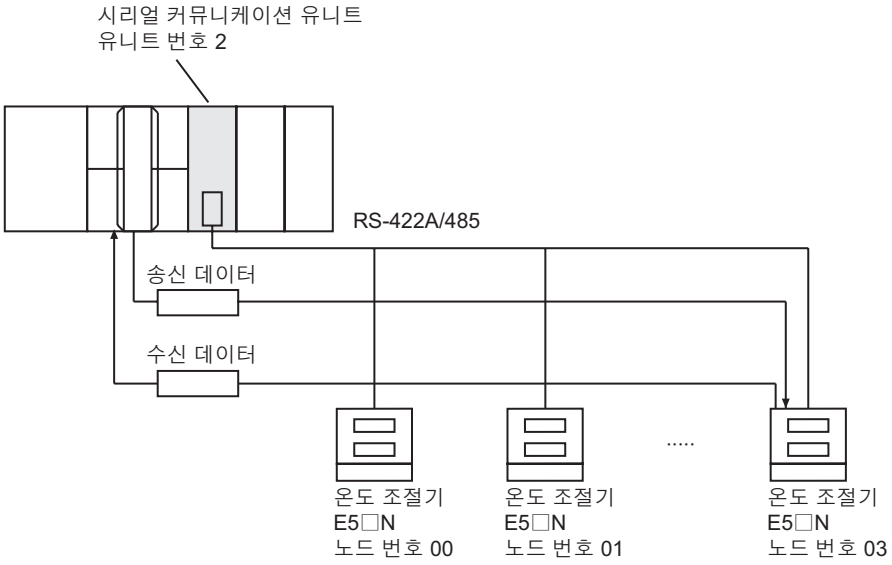
(주의) 위 표의 값 이외에 종료 코드의 비트 6, 7, 15의 값이 TRUE가 되는 경우가 있습니다. 비트 6, 7의 값이 TRUE인 경우는 송신대상 CPU 유니트에 이상이 발생했음을 나타냅니다. 비트 15의 값이 TRUE인 경우는 네트워크 중계 시 이상이 발생했음을 나타냅니다.

샘플 프로그램

CJ 시리즈의 시리얼 커뮤니케이션 유니트를 사용하여 오므론의 온도 조절기와의 데이터 송수신을 수행합니다. 송수신 프로토콜은 CompoWay/F 마스터의 시퀀스 번호 610(변수 영역 읽기)에서 온도 조절기의 현재값을 읽습니다.
컨트롤러에서는 송신 데이터 배열 SendData[]의 내용을 송신합니다.
온도 조절기에서 수신한 데이터는 수신 데이터 배열 RecvData[]에 저장됩니다.

주요한 통신 사양이 아래에 나와 있습니다.

항목	내용
사용 유니트	시리얼 커뮤니케이션 유니트
유니트 번호	2
포트 번호	1(RS422/485)
송수신 시퀀스 번호	610(변수 영역 읽기)
상대방 노드 번호	3
읽기 내용	현재값



시퀀스 번호 610(변수 영역 읽기)의 송수신 데이터의 할당 내용은 다음과 같습니다.

송신 데이터 WORD 형식 배열

SendData[0]	송신 데이터 채널 수	
SendData[1]	비어 있음	노드 번호
SendData[2]	변수 종류	
SendData[3]	읽기 시작 주소	
SendData[4]	요소 수	

수신 데이터 WORD 형식 배열

RecvData[0]	수신 데이터 채널 수
RecvData[1]	응답 코드
RecvData[2]	수신 데이터
RecvData[3]	

이 예에서의 송신 데이터 SendData[]와 수신 데이터 RecvData[]의 내용은 다음과 같습니다.

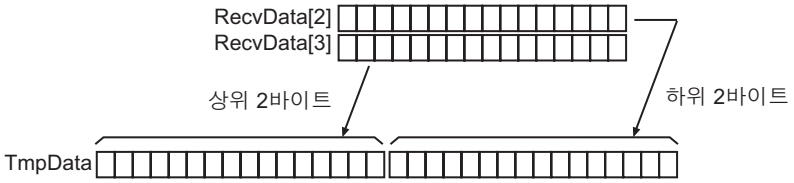
송신 데이터 WORD 형식 배열

변수	항목	이 예의 내용	값
SendData[0]	송신 데이터 채널 수	SendData[0]~SendData[4]의 5채널(워드)을 송신합니다.	WORD#16#0005
SendData[1]	노드 번호	노드 번호 3으로 합니다.	WORD#16#0003
SendData[2]	변수 종류+읽기 시작 주소의 상위 1바이트	현재값 읽기는 변수 종류가 BYTE#16#C0, 읽기 시작 주소가 WORD#16#00입니다.	WORD#16#C000
SendData[3]	읽기 시작 주소의 하위 1바이트 +고정값 BYTE#16#00		WORD#16#0000
SendData[4]	요소 수	요소 수는 1로 합니다.	WORD#16#0001

수신 데이터 WORD 형식 배열

변수	항목	이 예의 내용	값
RecvData[0]	수신 데이터 채널 수	RecvData[0]~RecvData[3]의 4채널(워드)을 수신합니다.	WORD#16#0004
RecvData[1]	응답 코드	정상 종료인 경우 WORD#16#0000이 반환됩니다.	
RecvData[2]	수신 데이터	온도 조절기의 현재값의 하위 2바이트가 반환됩니다.	
RecvData[3]		온도 조절기의 현재값의 상위 2바이트가 반환됩니다.	

수신에 성공하면 온도 조절기의 현재값의 하위 2바이트(RecvData[2])와 상위 2바이트(RecvData[3])를 조합하여 TmpData에 대입합니다.



글로벌 변수의 정의

글로벌 변수

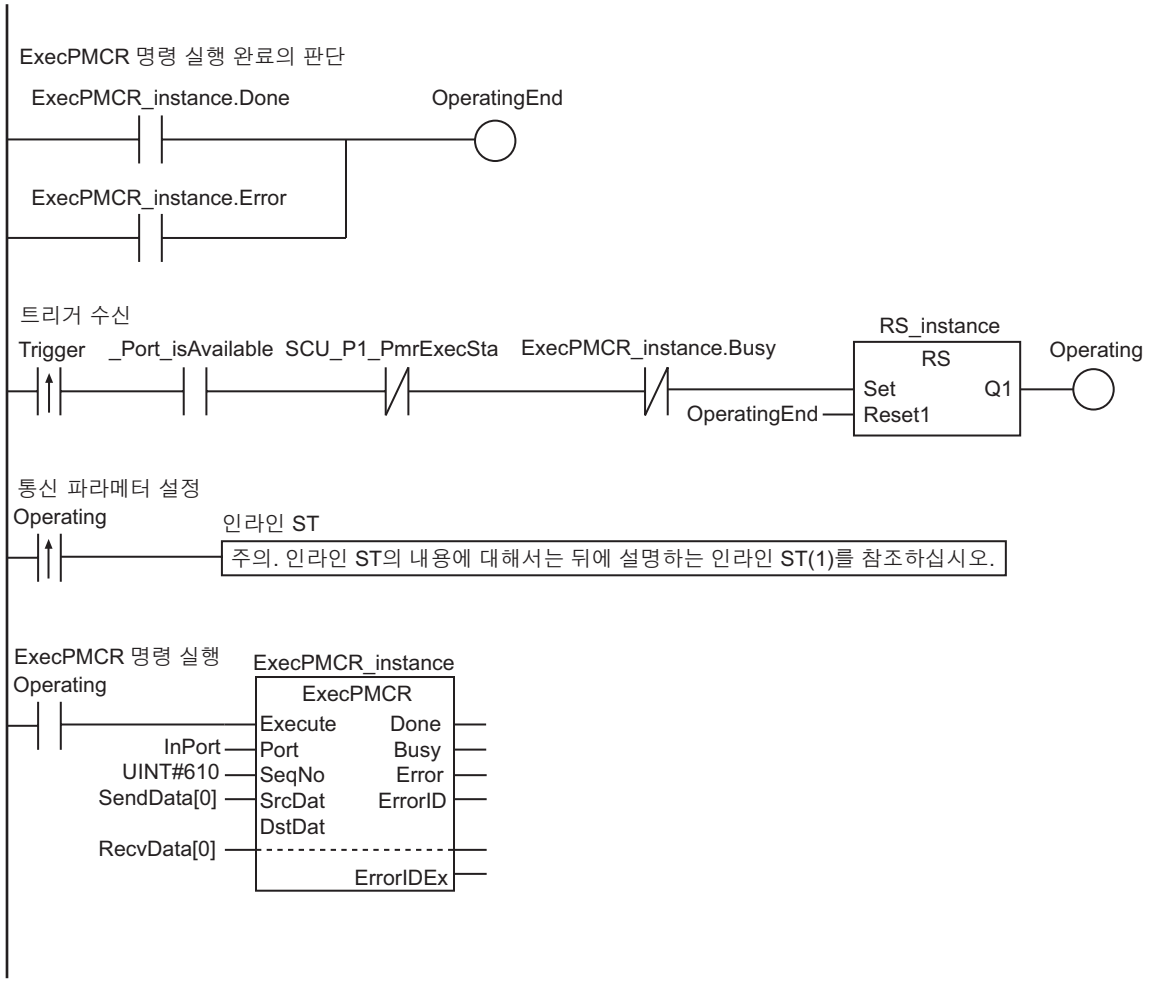
명칭	데이터 형식	할당대상 *1	코멘트
SCU_P1_PmrSeqEndSta	BOOL	IOBus://rack#0/slot#0/P1_PmrSta/P1_PmrSeqEndSta	시퀀스 End 종료 플래그
SCU_P1_PmrSeqAbtSta	BOOL	IOBus://rack#0/slot#0/P1_PmrSta/P1_PmrSeqAbtSta	시퀀스 Abort 종료 플래그
SCU_P1_PmrExecSta	BOOL	IOBus://rack#0/slot#0/P1_PmrSta/P1_PmrExecSta	프로토콜 매크로 실행 중 플래그

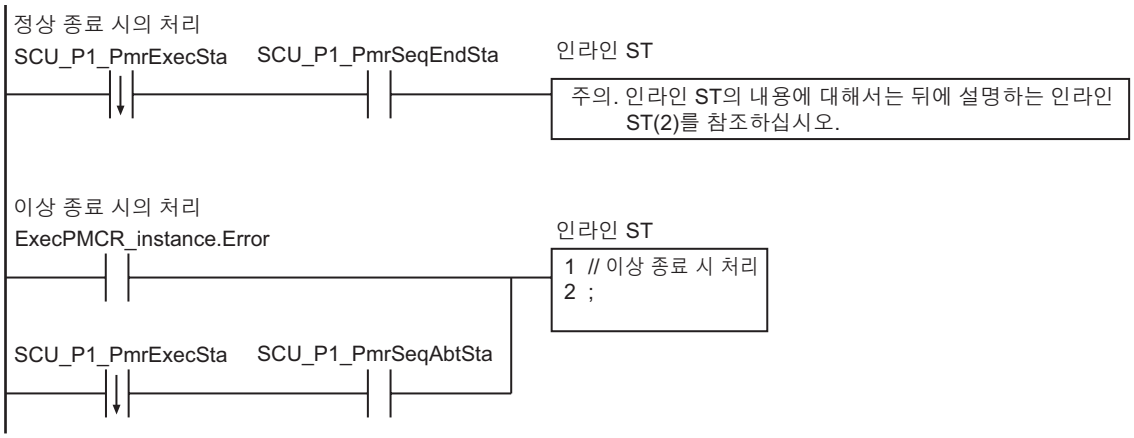
*1. 시리얼 커뮤니케이션 유니트를 랙 번호 0의 슬롯 번호 0에 장착한 경우의 할당대상입니다.

LD

내부 변수	명칭	데이터 형식	초기값	할당대상	유지	코멘트
	OperatingEnd	BOOL	FALSE		☐	처리 종료
	Trigger	BOOL	FALSE		☐	실행 조건
	Operating	BOOL	FALSE		☐	처리 중
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)		☐	포트 지정
	SendData	ARRAY[0..4] OF WORD	[5(16#0)]		☐	송신 데이터
	RecvData	ARRAY[0..3] OF WORD	[4(16#0)]	%D200	☑	수신 데이터
	TmpData	DINT	0		☐	현재값
	RS_instance	RS			☐	
	ExecPMCR_instance	ExecPMCR			☐	

외부 변수	명칭	데이터 형식	코멘트
	SCU_P1_PmrSeqEndSta	BOOL	시퀀스 End 종료 플래그
	SCU_P1_PmrSeqAbtSta	BOOL	시퀀스 Abort 종료 플래그
	SCU_P1_PmrExecSta	BOOL	프로토콜 매크로 실행 중 플래그
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그





● 인라인 ST(1)의 내용

```
InPort.UnitNo          :=_CBU_No02;   // 시리얼 커뮤니케이션 유니트의 유니트 번호 2
InPort.PhysicPortNo  :=USINT#1;       // 포트 번호 1
SendData[0]            :=WORD#16#0005;
SendData[1]            :=WORD#16#0003;
SendData[2]            :=WORD#16#C000;
SendData[3]            :=WORD#16#0000;
SendData[4]            :=WORD#16#0001;
RecvData[0]            :=WORD#16#0004;
```

● 인라인 ST(2)의 내용

```
// 정상 종료 시 처리
TmpData:=DWORD_TO_DINT(SHL(WORD_TO_DWORD(
            RecvData[3]), 16) OR WORD_TO_DWORD(RecvData[2]) );
```

ST

내부 변수	명칭	데이터 형식	초기값	할당대상	유지	코멘트
	State	INT	0		☐	이동 상태
	Trigger	BOOL	FALSE		☐	실행 조건
	LastTrigger	BOOL	FALSE		☐	이전 태스크 주기의 Trigger의 값
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)		☐	포트 지정
	SendData	ARRAY[0..4] OF WORD	[5(16#0)]		☐	송신 데이터
	RecvData	ARRAY[0..3] OF WORD	[4(16#0)]	%D200	☑	수신 데이터
	End_ExecPMCR	BOOL	FALSE		☐	ExecPMCR 명령 실행 종료
	TmpData	DINT	0		☐	현재값
	RS_instance	RS			☐	
	ExecPMCR_instance	ExecPMCR			☐	
	F_TRIG_instance	F_TRIG			☐	

외부 변수	명칭	데이터 형식	코멘트
	SCU_P1_PmrSeqEndSta	BOOL	시퀀스 End 종료 플래그
	SCU_P1_PmrSeqAbtSta	BOOL	시퀀스 Abort 종료 플래그
	SCU_P1_PmrExecSta	BOOL	프로토콜 매크로 실행 중 플래그
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그

```
// 트리거 수신
IF (State=INT#0) THEN
   IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Port_isAvailable=TRUE) AND
(SCU_P1_PmrExecSta<>TRUE)
        AND (ExecPMCR_instance.Busy<>TRUE) ) THEN
      State:=INT#1;
   END_IF;
END_IF;
LastTrigger:=Trigger;

// 통신 파라메터 설정과 ExecPMCR 명령 초기화
IF (State=INT#1) THEN
   InPort.UnitNo         :=_CBU_No02;   // 시리얼 커뮤니케이션 유니트의 유니트 번호 2
   InPort.PhysicPortNo :=USINT#1;       // 포트 번호 1
   SendData[0]           :=WORD#16#0005;
   SendData[1]           :=WORD#16#0003;
   SendData[2]           :=WORD#16#C000;
   SendData[3]           :=WORD#16#0000;
   SendData[4]           :=WORD#16#0001;
   RecvData[0]           :=WORD#16#0004;
   ExecPMCR_instance(
      Execute  :=FALSE,         // ExecPMCR 명령 초기화
      SrcDat   :=SendData[0],  // 더미
      DstDat   :=RecvData[0]);
    State:=INT#2;
END_IF;
```

```
// ExecPMCR 명령실행
IF (State=INT#2) THEN
   ExecPMCR_instance(
   Execute :=TRUE,
   Port       :=InPort,
   SeqNo   :=UINT#610,
   SrcDat   :=SendData[0],
   DstDat   :=RecvData[0]);

   F_TRIG_instance(SCU_P1_PmrExecSta, End_ExecPMCR);

   IF (End_ExecPMCR=TRUE) THEN
      End_ExecPMCR:=FALSE;
      State:=INT#3;
   END_IF;

   IF (ExecPMCR_instance.Error=TRUE) THEN
      State:=INT#5;
   END_IF;
END_IF;

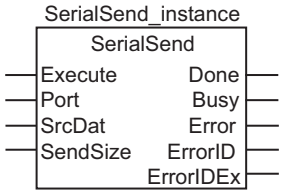
// ExecPMCR 명령 실행 종료 확인
IF (State=INT#3) THEN
   IF (SCU_P1_PmrSeqEndSta=TRUE) THEN
      State:=INT#4;
   END_IF;
   IF (SCU_P1_PmrSeqAbtSta=TRUE) THEN
      State:=INT#5;
   END_IF;
END_IF;

IF (State=INT#4) THEN
   // 고정 종료 시 처리
   TmpData:=DWORD_TO_DINT(SHL(WORD_TO_DWORD(RecvData[3]), 16)
         OR WORD_TO_DWORD(RecvData[2]));
   State:=INT#0;
END_IF;

IF (State=INT#5) THEN
   // 정상 종료 시 처리
   State:=INT#0;
END_IF;
```

SerialSend

시리얼 커뮤니케이션 유니트의 시리얼 포트에서 무수순으로 데이터를 송신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	FB		SerialSend_instance(Execute, Port, SrcDat, SendSize, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Port	상대방 포트 지정	입력	상대방 포트 지정	–	–	–
SrcDat[] 배열	송신 데이터 배열		송신 데이터 배열	데이터 형식에 따름		(*)
SendSize	송신 데이터 크기		SrcDat[] 중에서 송신할 데이터 크기	0~256	바이트	1

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Port	구조체 _sPORT에 대한 자세한 내용은 기능 설명 참조																			
SrcDat[] 배열		○																		
SendSize							○													

기능

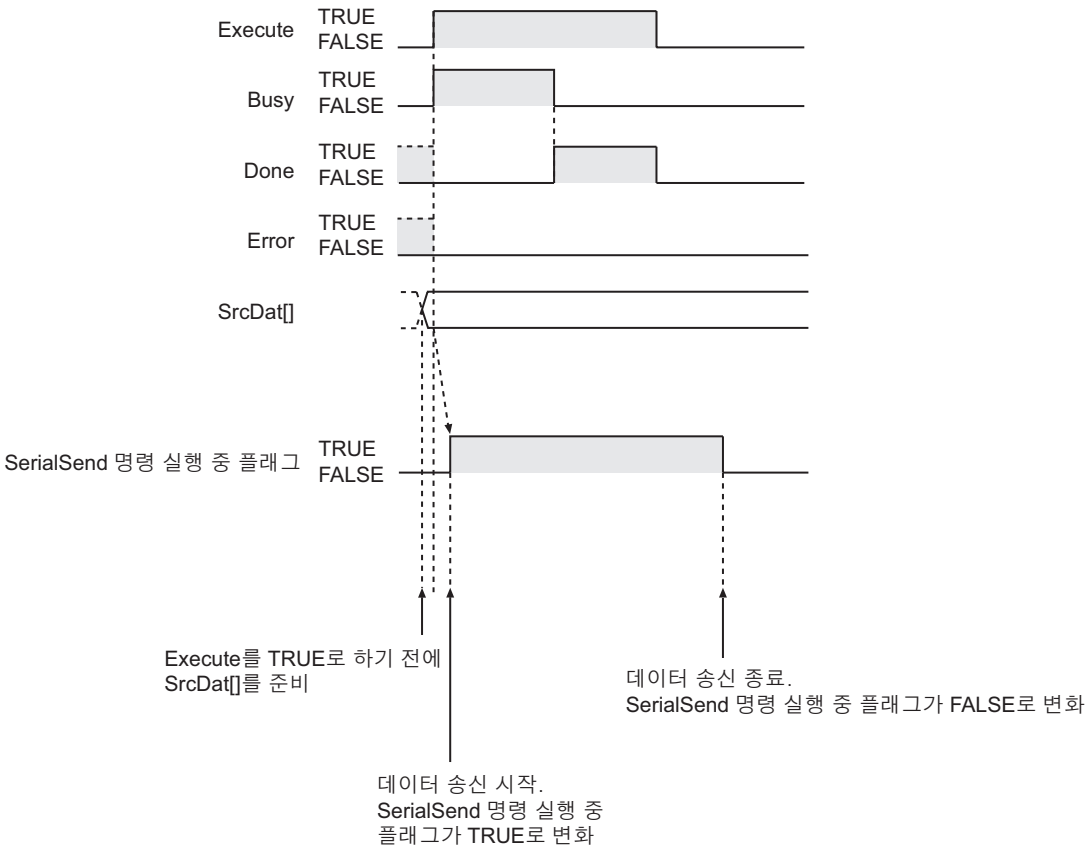
시리얼 커뮤니케이션 유니트의 'Port'로 지정한 포트에서 무수순으로 데이터를 송신합니다.
송신할 데이터는 송신 데이터 배열 SrcDat[]의 내용입니다. 송신할 데이터 크기는 송신 데이터 크기 'SendSize'로 지정합니다.

송신 데이터에 시작 코드 및 종료 코드를 부가하는 경우, 고기능 유니트의 할당 DM 설정 영역에서 설정을 수행합니다.
시작 코드, 종료 코드를 부가한 경우의 송신 가능 바이트 수는 최대 259입니다(시작 코드 1바이트+종료 코드 2바이트(CR+LF 지정 시)+송신 데이터 256바이트).

상대방 포트 지정 'Port'의 데이터 형식은 구조체 _sPORT입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Port	상대방 포트 지정	상대방 포트 지정	_sPORT	–	–	–
UnitNo	유니트 번호	시리얼 커뮤니케이션 유니트의 유니트 번호	_eUnitNo	_CBU_No00~ _CBU_No15	–	_CBU _No00
PhysicPort No	시리얼 포트 번호	시리얼 커뮤니케이션 유니트의 시리얼 포트 번호	USINT	1,2		1

타이밍 차트가 아래에 나와 있습니다. 'Done'의 값이 TRUE가 된 후에도 통신은 끝까지 수행됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Port_numUsingPort	사용 포트 수	USINT	현재 사용 중인 포트 수
_Port_isAvailable	네트워크 통신 명령 실행 가능 플래그	BOOL	TRUE : 사용 가능 포트 있음 FALSE : 사용 가능 포트 없음

관련된 준 사용자 정의 변수

변수명	명칭	데이터 형식	내용
P#_NopSerialSendExec Sta(*)	SerialSend 명령 실행 중 플래그	BOOL	TRUE : 실행 중 FALSE : 비실행 중
P#_NopStartCodeYNCfg(*)	무수순 시작 코드 유무	BOOL	TRUE : 있음 FALSE : 없음
P#_NopEndCodeYNCfg(*)	무수순 종료 코드 유무	BOOL	TRUE : 있음 FALSE : 없음
P#_NopCRLFCfg(*)	무수순 CRLF 지정	BOOL	TRUE : 지정함 FALSE : 지정하지 않음
P#_NopStartCodeCfg(*)	무수순 시작 코드	USINT	16#00~16#FF
P#_NopEndCodeCfg(*)	무수순 종료 코드	USINT	16#00~16#FF

* #에는 시리얼 커뮤니케이션 유니트의 포트 번호가 들어갑니다.

참고

무수순 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- "CJ 시리즈 시리얼 커뮤니케이션 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-354)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 무수순 모드로 설정한 시리얼 커뮤니케이션 유니트의 시리얼 포트에만 사용할 수 있습니다.
- 'SendSize'의 값이 0인 경우, 송신을 수행하지 않습니다. 그 경우, 명령 실행 시에 'Done'의 값이 TRUE가 됩니다.
- 시작 코드, 종료 코드를 부가하는 경우, 'SendSize'에는 시작 코드, 종료 코드를 포함하지 않는 값을 설정하십시오.
- 이 명령은 포트가 사용 가능한 경우에만 실행할 수 있습니다. 따라서 이 명령의 입력 조건에 네트워크 통신 명령 실행 가능 플래그의 시스템 정의 변수 '_Port_isAvailable'을 a 접점으로 삽입하십시오.
- 이 명령은 'Busy'의 값이 TRUE인 동안에는 실행할 수 없습니다. 따라서 이 명령의 입력 조건으로 'Busy'를 b 접점으로 삽입하십시오.
- 이 명령은 SerialSend 명령 실행 중 플래그의 준 사용자 정의 변수 'P#NopSerialSendExecSta'의 값이 TRUE인 경우에는 실행할 수 없습니다. 따라서 이 명령의 입력 조건으로 'P#NopSerialSendExecSta'를 b 접점으로 삽입하십시오.
- 이 명령을 ST 프로그램으로 기술하는 경우, 이 명령 실행 중에는 태스크 주기마다 이 명령의 제어가 수행되도록 하십시오. 그렇게 하지 않으면 정상적인 처리를 할 수 없는 경우가 있습니다.

- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 시리얼 통신 모드가 무수순 이외이고 이 명령이 실행된 경우
  - '_Port_isAvailable'의 값이 FALSE인 경우
  - 'Port.UnitNo' 또는 'Port.PhysicPortNo'의 값이 유효 범위 밖인 경우
  - 지정한 유니트 번호에 CJ 시리얼 커뮤니케이션 유니트가 장착되지 않은 경우
  - 'SendSize'의 값이 유효 범위 밖인 경우
  - 'SendSize'의 값이 SrcDat[]의 크기를 초과한 경우
  - 통신이 실패한 경우
  - 유니트 재시작 중에 이 명령을 실행한 경우

? 확장 에러 코드 'ErrorIDEx'는 이 명령에서 통신 응답 코드로서의 의미를 갖습니다. 그 값과 이상 내용은 다음과 같습니다. 'ErrorIDEx'는 에러 코드 'ErrorID'의 값이 WORD#16#0800인 경우에 출력됩니다.

값	이상 내용	대처 방법
16#00000001	통신 서비스를 중단했습니다.	• 데이터 링크 기동 상태를 확인하십시오. • 제3노드의 전송처 영역의 용량을 확인하십시오.
16#00000101	로컬 노드가 네트워크에 가입하지 않았습니다.	로컬 노드를 네트워크에 가입시키십시오.
16#00000102	토큰 타임아웃이 발생했습니다.	로컬 노드를 최대 노드 주소 범위 내로 설정하십시오.
16#00000103	재전송 초과가 발생했습니다.	노드간 테스트를 실행하여 정상이 아닌 경우, 사용 환경을 확인하십시오.
16#00000104	송신 허가 프레임 수 초과가 발생했습니다.	네트워크 내에서 실행되고 있는 이벤트의 상황을 확인하고 1태스크 주기 내의 이벤트 수를 줄이십시오. 또는 송신 허가 프레임 수를 늘리십시오.
16#00000105	로컬 노드의 노드 주소가 설정 범위 밖입니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000106	로컬 노드의 노드 주소가 네트워크 내에서 중복되어 있습니다.	중복되어 있는 노드의 주소를 변경하십시오.
16#00000201	상대 노드가 네트워크에 가입하지 않았습니다.	상대 노드를 네트워크에 가입시키십시오.
16#00000202	송신대상으로 지정한 호기 주소의 유니트가 존재하지 않습니다.	송신대상 네트워크 주소의 호기 주소를 정확하게 설정하십시오.
16#00000203	제3노드가 네트워크에 가입하지 않았습니다.	• 제3노드로 지정된 유니트의 주소를 확인하십시오. • 제3노드에는 1노드만 지정하십시오.
16#00000204	상대 노드가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 상대 노드에 집중되지 않도록 시스템을 재검토하십시오.
16#00000205	응답 타임아웃이 발생했습니다.	통신 파라메터의 설정을 확인하십시오.
16#00000206	전송로에 이상이 있습니다.	• 다시 시도하십시오. • 이상이 자주 발생하는 경우는 노이즈의 상황을 확인하십시오.
16#00000301	통신 컨트롤러 이상이 발생했습니다.	해당 유니트의 사용 설명서를 참조하여 적절하게 처리하십시오.
16#00000302	상대 노드의 CPU 유니트에 이상이 있습니다.	상대 노드의 CPU 유니트 설명서를 참조하여 이상을 제거하십시오.
16#00000303	해당 컨트롤러에 이상이 있어서 응답이 반환되지 않습니다.	네트워크의 통신 상황을 확인한 후 해당 컨트롤러를 재기동하십시오. 그래도 이상이 발생하는 경우는 해당 컨트롤러를 교환하십시오.
16#00000304	유니트 번호 설정이 잘못되었습니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000401	송신한 명령이 지원되지 않습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00000402	유니트의 기종 또는 버전이 지원되지 않습니다.	유니트의 기종과 버전을 확인하십시오.
16#00000501	상대 주소 설정 이상입니다.	라우팅 테이블에 수신대상 주소를 설정하십시오.
16#00000502	라우팅 테이블이 등록되지 않았습니다.	송신원 노드, 수신대상 노드, 중계 노드에 라우팅 테이블을 설정하십시오.
16#00000503	라우팅 테이블 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.

값	이상 내용	대처 방법
16#00000504	중계 횟수 초과가 발생했습니다.	네트워크를 재구성하거나 라우팅 테이블을 재검토하여 명령의 사용 범위를 3계층 이내로 하십시오.
16#00001001	명령 길이가 최대 명령 길이를 초과했습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001002	명령 길이가 최소 명령 길이 미만입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001003	명령에 지정된 쓰기 요소 수와 실제 쓰기 데이터 수가 일치하지 않습니다.	쓰기 요소 수와 실제 쓰기 데이터 수를 일치시키십시오.
16#00001004	명령 형식 이상입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001005	헤더 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.
16#00001101	영역 종류가 없습니다.	명령의 변수, 파라메터 종류 코드를 참조한 후 해당하는 코드를 설정하십시오.
16#00001102	액세스 크기 이상입니다.	변수, 파라메터의 액세스 크기를 정확하게 설정하십시오.
16#00001103	주소 범위 밖이 지정되었습니다.	처리 가능 범위 내의 주소를 지정하십시오.
16#00001104	주소 범위를 초과했습니다.	• 처리 가능 범위 내의 주소를 지정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#00001106	등록되지 않은 송수신 시퀀스 번호가 지정되었습니다.	송수신 시퀀스 번호를 수정하거나 CX-Protocol로 시퀀스를 추가하십시오.
16#00001109	상관 관계 이상이 발생했습니다.	• 명령 데이터의 대소 관계를 정확하게 설정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110A	데이터가 중복되어 있습니다.	• 실행 중인 처리를 중단하거나 종료할 때까지 기다린 후 명령을 실행하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110B	응답이 최대 응답 길이를 초과했습니다.	명령 배열 내의 요소 수를 정확하게 설정하십시오.
16#0000110C	기타 파라메터 이상	명령 배열의 내용을 정확하게 설정하십시오.
16#00002002	보호 중입니다.	보호를 해제한 후 명령을 다시 실행하십시오.
16#00002003	등록 테이블이 없습니다.	테이블을 정확하게 설정하십시오.
16#00002004	검색 데이터와 일치하는 데이터가 존재하지 않습니다.	검색 데이터를 정확하게 설정하십시오.
16#00002005	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002006	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002007	비교 이상이 발생했습니다.	• 메모리 내용을 확인하고 정확한 데이터로 재기입하십시오. • 파일 내용을 확인하십시오.
16#00002101	읽기 전용 영역이므로 액세스할 수 없습니다.	쓰기 방지를 해제한 후 명령을 다시 실행하십시오.

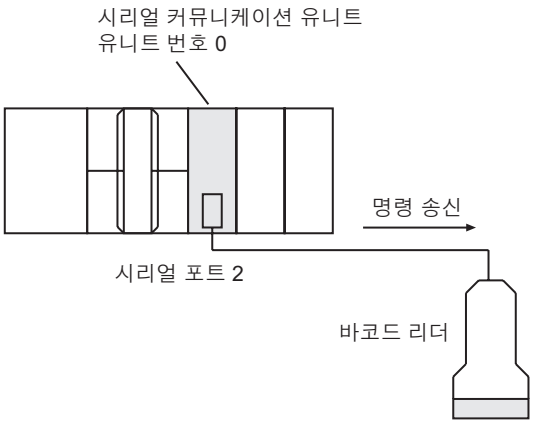
값	이상 내용	대처 방법
16#00002102	보호 중 또는 데이터 링크 테이블 쓰기 불가능 상태입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 시스템 설정을 데이터 링크 테이블에 설정하십시오.
16#00002103	등록할 수 없습니다.	• 필요 없는 파일을 삭제한 후 파일을 작성하거나 새 파일용 메모리를 준비하십시오. • 열려 있는 파일을 닫은 후 명령을 다시 실행하십시오.
16#00002105	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002106	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002107	동일한 파일명이 존재합니다.	쓸 파일의 파일명을 변경하고 명령을 다시 실행하십시오.
16#00002108	변경에 의해 이상이 발생하므로 변경할 수 없습니다.	설정을 변경하십시오.
16#00002201	이미 프로토콜 매크로를 실행 중이므로 동작 불가능합니다.	프로토콜 매크로 실행 중 플래그를 b 접점에 설정하십시오.
16#00002202	동작 모드 이상입니다.	동작 모드를 확인하십시오.
16#00002203	명령의 동작 모드가 다릅니다(프로그램 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002204	명령의 동작 모드가 다릅니다(디버그 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002205	명령의 동작 모드가 다릅니다(모니터 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002206	명령의 동작 모드가 다릅니다(운전 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002207	지정 노드가 관리국이 아닙니다.	네트워크의 관리국 노드를 확인하십시오.
16#00002208	명령의 동작 모드가 다릅니다.	스텝 활성 상태를 확인하십시오.
16#00002211	유니트가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 해당 유니트에 집중되지 않도록 시스템을 재검토하십시오.
16#00002301	파일 장치가 없습니다.	미디어를 장착하십시오. 또는 EM 포맷을 실시하십시오.
16#00002302	파일 메모리가 없습니다.	파일 메모리 장착을 확인하십시오.
16#00002303	시계가 내장되어 있지 않습니다.	기종의 사양을 확인하십시오.
16#00002401	프로토콜 매크로 데이터에서 SUM 값 이상이 발생했거나 데이터 전송 중입니다.	CX-Protocol로 프로토콜 매크로 데이터를 다시 전송하십시오.
16#00002502	처리 대상 메모리에 이상이 있습니다.	대상 메모리에 정확한 데이터를 다시 전송하십시오.
16#00002503	등록된 I/O 유니트 구성과 실제 유니트의 구성이 서로 다릅니다	I/O 유니트 구성을 확인하십시오.
16#00002504	등록된 I/O 점수 또는 리모트 I/O 점수가 최대값을 초과했습니다.	I/O 점수, 리모트 I/O 점수를 정확하게 설정하십시오.
16#00002505	CPU 유니트와 CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#00002506	랙 번호, 유니트 번호, I/O 주소 중 하나의 설정이 중복되어 있습니다.	중복된 내용을 다시 설정하십시오.
16#00002507	CPU 유니트와 I/O 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.

값	이상 내용	대처 방법
16#00002509	SYSMAC BUS/2 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250A	CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250D	채널 설정이 중복되어 있습니다.	I/O 채널을 정확하게 설정하십시오.
16#0000250F	메모리 이상입니다.	• 내부 메모리의 경우, 정확한 데이터를 쓰고 명령을 다시 실행하십시오. • 메모리 카드나 EM 파일 메모리의 경우, 확장 메모리의 포맷 명령을 실행하십시오. • 상기의 처리를 실시해도 이상이 해제되지 않는 경우는 메모리를 교환하십시오.
16#00002510	종단국 설정 이상입니다.	종단국을 정확하게 설정하십시오.
16#00002601	보호 해제 중입니다.	보호를 해제할 필요가 없습니다.
16#00002602	비밀번호가 일치하지 않습니다.	정확한 비밀번호를 지정하십시오.
16#00002604	보호 중입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002605	서비스 실행 중입니다.	실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002606	서비스 정지 중입니다.	필요에 따라 해당 서비스를 실행하십시오.
16#00002607	실행 권한이 없습니다.	• 데이터 링크 참가 노드에서 실행하십시오. • 재기동해도 이상이 재발하는 경우는 컨트롤러를 교환하십시오.
16#00002608	환경이 설정되지 않았습니다.	필요한 설정을 하십시오.
16#00002609	필요한 항목이 설정되지 않았습니다.	필요한 항목을 설정하십시오.
16#0000260A	지정 번호의 정의가 완료되었습니다.	등록되지 않은 작업이나 트랜지션 번호로 변경하고 명령을 다시 실행하십시오.
16#0000260B	이상 해제가 불가능합니다.	이상의 원인이 된 요인을 해제하고 이상 해제 명령을 실행하십시오.
16#00003001	액세스 권한이 없습니다.	액세스 권한이 해제될 때까지 기다린 후 명령을 다시 실행하십시오.
16#00004001	서비스가 중단되었습니다.	서비스가 중단된 요인을 해제하고 명령을 다시 실행하십시오.

(주의) 위 표의 값 이외에 종료 코드의 비트 6, 7, 15의 값이 TRUE가 되는 경우가 있습니다. 비트 6, 7의 값이 TRUE인 경우는 송신대상 CPU 유니트에 이상이 발생했음을 나타냅니다. 비트 15의 값이 TRUE인 경우는 네트워크 중계 시 이상이 발생했음을 나타냅니다.

샘플 프로그램

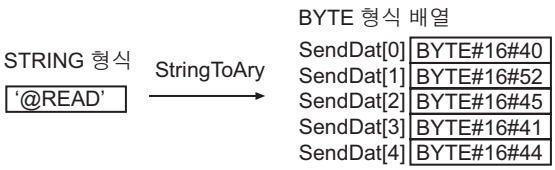
CJ 시리즈의 시리얼 커뮤니케이션 유니트(유니트 번호 0, 장치명 'Barcode')의 시리얼 포트 2에 연결한 바코드 리더에 무수순 명령을 송신합니다. 송신하는 명령은 신(Scene) 번호 취득 명령 '@READ'입니다. 송신 데이터는 배열 변수 SendDat[]의 내용입니다. 시작 코드는 없음, 종료 코드는 16#OD(CR)로 합니다.



시리얼 커뮤니케이션 유니트는 다음과 같이 설정합니다.

항목명	설정값
포트 2: 임의 설정 유무	임의 설정
포트 2: 시리얼 통신 모드	무수순
포트 2: 데이터 길이	8비트
포트 2: 정지 비트	1 비트
포트 2: 패리티	없음
포트 2: 전송 속도	38400bps
포트 2: 무수순 종료 코드	D
포트 2: 무수순 시작 코드 유무	없음
포트 2: 무수순 종료 코드 유무	있음(종료 코드를 임의 설정)

SendDat[]에는 문자열 '@READ'를 1문자씩 분해하여 그 문자 코드를 각 배열 요소에 저장해둡니다. 따라서 SendDat[0]에는 BYTE#16#40(@), sendData[1]에는 BYTE#16#52(R)를 저장합니다. 그 처리에는 StringToAry 명령을 사용합니다.



글로벌 변수의 정의

글로벌 변수

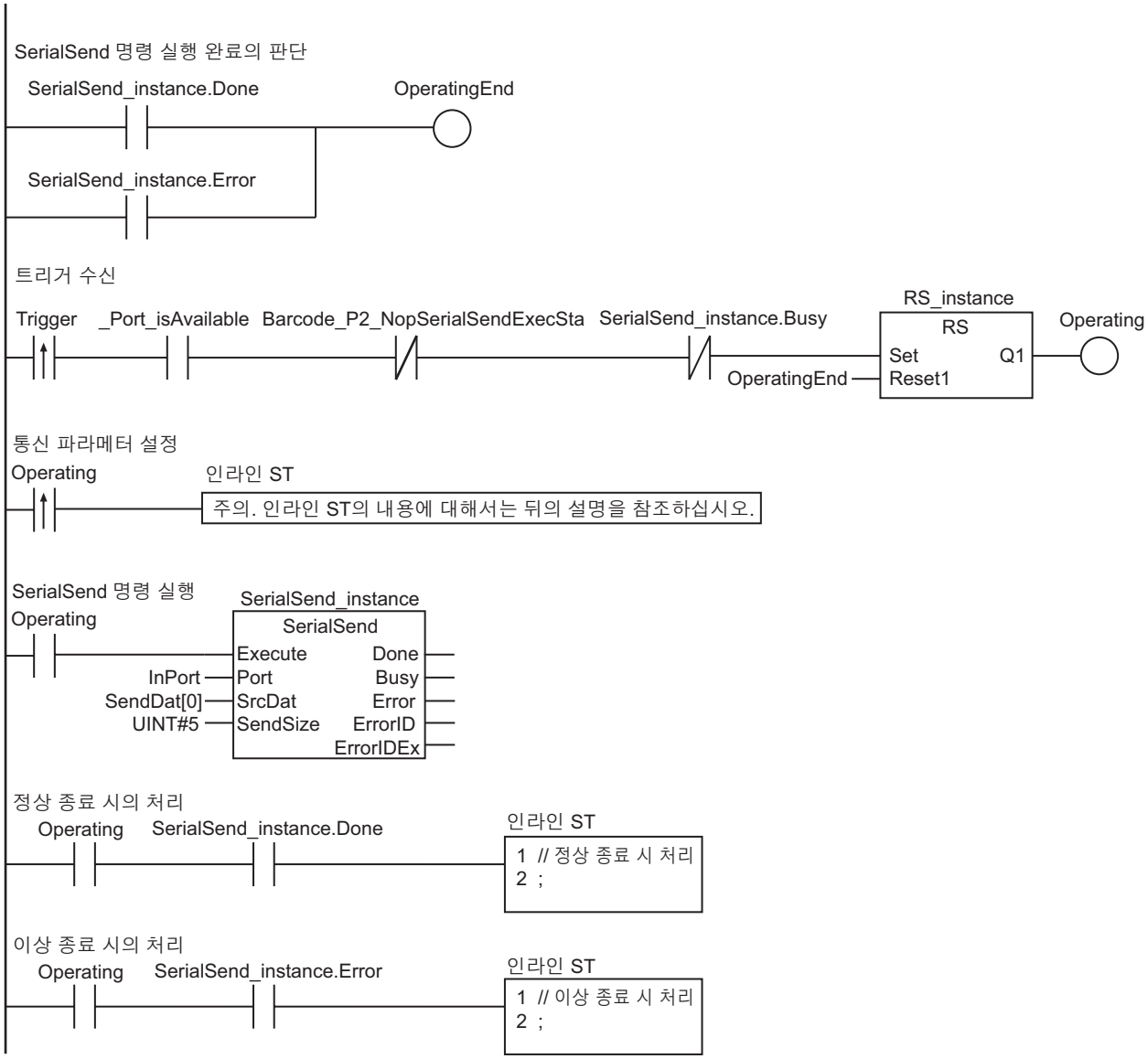
명칭	데이터형식	할당대상*1	코멘트
Barcode_P2_NopSerialSendExecSta	BOOL	IOBus://rack#0/slot#0/P2_NopSta/ P2_NopSerialSendExecSta	SerialSend 명령 실행 중 플래그

*1. 시리얼 커뮤니케이션 유니트를 랙 번호 0의 슬롯 번호 0에 장착한 경우의 할당대상입니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)	포트 지정
	SendDat	ARRAY[0..4] OF BYTE	[5(16#0)]	송신 데이터
	RS_instance	RS		
	SerialSend_instance	SerialSend		

내부 변수	명칭	데이터 형식	코멘트
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그
	Barcode_P2_NopSerialSendExecSta	BOOL	SerialSend 명령 실행 중 플래그



● 인라인 ST의 내용

```
StringToAry(In          :='@READ', AryOut:=SendDat[0]);   // SendDat[] 준비
InPort.UnitNo           :=_CBU_No00;                      // 시리얼 커뮤니케이션 유니트의 유니트 번호 0
InPort.PhysicPortNo     :=USINT#2;                        // 시리얼 포트 2
```

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)	포트 지정
	SendDat	ARRAY[0..4] OF BYTE	[5(16#0)]	송신 데이터
	SerialSend_instance	SerialSend		

외부 변수	명칭	데이터 형식	코멘트
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그
	Barcode_P2_NopSerialSendExecSta	BOOL	SerialSend 명령 실행 중 플래그

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Port_isAvailable=TRUE)
     AND (Barcode_P2_NopSerialSendExecSta=FALSE) AND (SerialSend_instance.Busy=FALSE) ) THEN
   OperatingStart  :=TRUE;
   Operating       :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 통신 파라메터 설정과 SerialSend 명령 초기화
IF (OperatingStart=TRUE) THEN
   SerialSend_instance(
      Execute  :=FALSE,
      SrcDat   :=SendDat[0]);
   StringToAry(In:='@READ', AryOut:=SendDat[0]);
   InPort.UnitNo          :=_CBU_No00;      // 시리얼 커뮤티케이션 유니트의 유니트 번호 0
   InPort.PhysicPortNo    :=USINT#2;        // 시리얼 포트 2
   OperatingStart         :=FALSE;
END_IF;

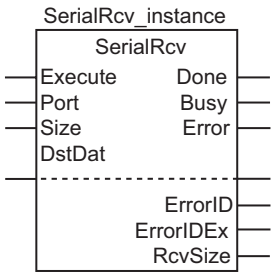
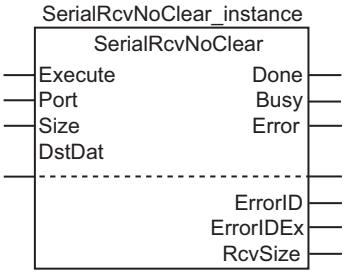
// SerialSend 명령 실행
IF (Operating=TRUE) THEN
   SerialSend_instance(
      Execute   :=TRUE,
      Port      :=InPort,          // 포트 지정
      SrcDat    :=SendDat[0],      // 송신 데이터
      SendSize  :=UINT#5);         // 송신 데이터 크기

   IF (SerialSend_instance.Done=TRUE) THEN
      // 정상 종료 시 처리
      Operating:=FALSE;
   END_IF;

   IF (SerialSend_instance.Error=TRUE) THEN
      // 이상 종료 시 처리
      Operating:=FALSE;
  END_IF;
END_IF;
```

SerialRcv / SerialRcvNoClear

시리얼 커뮤니케이션 유니트의 시리얼 포트에서 무수순의 수신 데이터를 읽어옵니다.
SerialRcv : 읽은 후, 수신 버퍼는 전체 클리어됩니다.
SerialRcvNoClear : 읽은 후, 수신 버퍼는 전체 클리어되지 않습니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SerialRcv	시리얼 커뮤니케이션 유니트 시리얼 포트 입력	FB		SerialRcv_instance(Execute, Port, Size, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx, RcvSize);
SerialRcvNoClear	시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함	FB		SerialRcvNoClear_instance(Execute, Port, Size, DstDat, Done, Busy, Error, ErrorID, ErrorIDEx, RcvSize);

변수

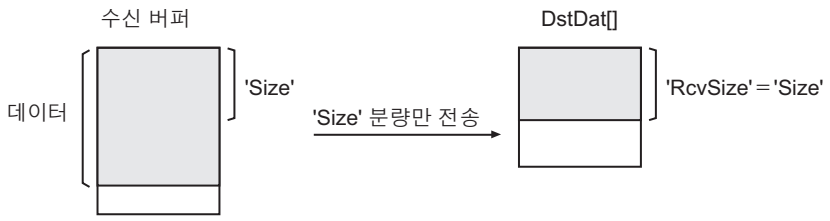
	명칭	입/출력	내용	유효 범위	단위	초기값
Port	상대방 포트 지정	입력	상대방 포트 지정	–	–	–
Size	수신 데이터 크기		DstDat[]에 저장할 수신 데이터 크기	0~256	바이트	1
DstDat[] 배열	수신 데이터 배열	입출력	수신 데이터 배열	데이터 형식에 따름	–	–
RcvSize	수신 데이터 저장 크기	출력	DstDat[]에 실제로 저장한 수신 데이터 배열 크기	0~256	바이트	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Port	구조체 _sPORT에 대한 자세한 내용은 기능 설명 참조																			
Size							○													
DstDat[] 배열		○																		
RcvSize							○													

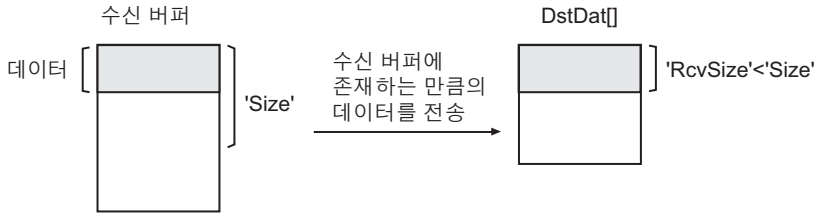
기능

지정한 포트 'Port'에서 무수순으로 수신한 데이터는 먼저 시리얼 커뮤니케이션 유니트의 수신 버퍼에 저장됩니다. 이 명령은 그 수신 버퍼의 데이터를 수신 데이터 크기 'Size'만큼 수신 데이터 배열 DstDat[]에 전송합니다.
전송 후 실제로 DstDat[]에 저장한 배열 요소 수를 수신 데이터 저장 크기 'RcvSize'에 대입합니다.
수신 버퍼 내의 데이터량이 'Size'보다 작은 경우에는 수신 버퍼에 존재하는 만큼의 데이터를 DstDat[]에 전송합니다. 그 경우에도 실제로 DstDat[]에 저장한 데이터 크기를 'RcvSize'에 대입합니다.

수신 버퍼 내의 데이터가 'Size' 이상인 경우



수신 버퍼 내의 데이터가 'Size'보다 적은 경우



수신 데이터의 시작 코드, 종료 코드

수신 데이터의 시작 코드, 종료 코드를 사용자 프로그램 내에서 인식하려면 디바이스 변수를 사용합니다. DstDat[]에는 수신 데이터에서 시작 코드, 종료 코드를 삭제한 것을 저장합니다.

부가할 내용	디바이스 변수(포트 1의 경우)	값
시작 코드에 임의의 값	P1_NopStartCodeYNCfg	TRUE
	P1_NopStartCodeCfg	시작 코드 (16#00~16#FF)
종료 코드에 임의의 값	P1_NopEndCodeYNCfg	TRUE
	P1_NopCRLFCfg	FALSE
	P1_NopEndCodeCfg	종료 코드 (16#00~16#FF)
종료 코드에 CR+LF	P1_NopEndCodeYNCfg	TRUE
	P1_NopCRLFCfg	TRUE

시작 코드, 종료 코드를 부가한 경우의 수신 가능 바이트 수는 최대 259입니다(시작 코드 1바이트+종료 코드 2바이트(CR+LF 지정 시)+송신 데이터 256바이트).

상대방 포트 지정 'Port'의 데이터 형식

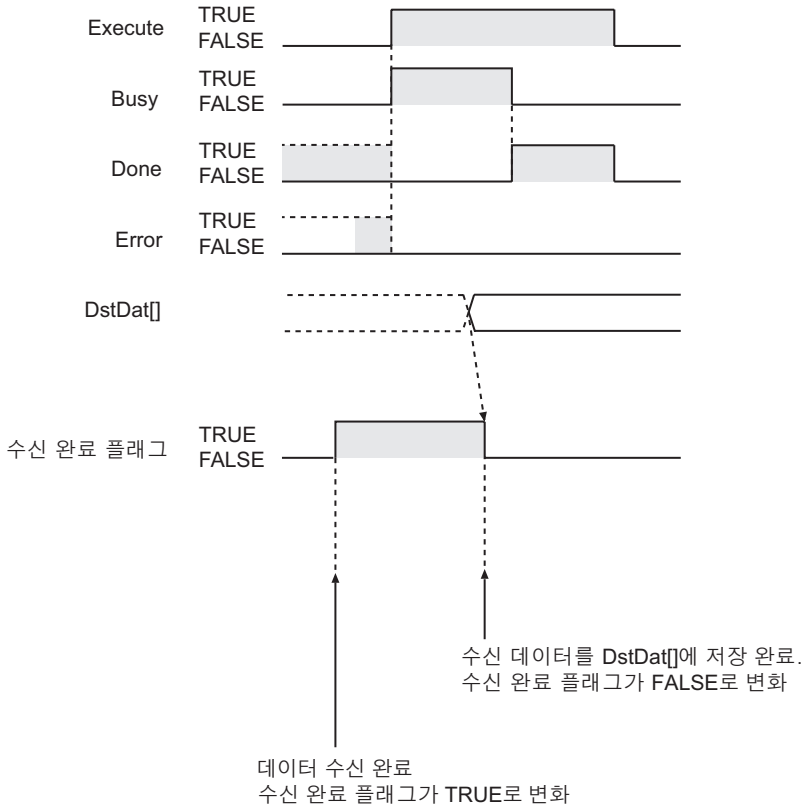
상대방 포트 지정 'Port'의 데이터 형식은 구조체 _sPORT입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Port	상대방 포트 지정	상대방 포트 지정	_sPORT	–	–	–
UnitNo	유니트 번호	시리얼 커뮤니케이션 유니트의 유니트 번호	_eUnitNo	_CBU_No00~ _CBU_No15	–	_CBU _No00
PhysicPort No	시리얼 포트 번호	시리얼 커뮤니케이션 유니트의 시리얼 포트 번호	USINT	1,2		1

타이밍 차트

타이밍 차트가 아래에 나와 있습니다.

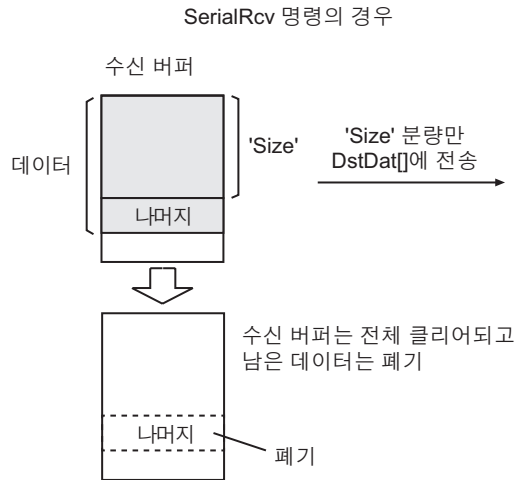
타이밍 차트가 아래에 나와 있습니다.



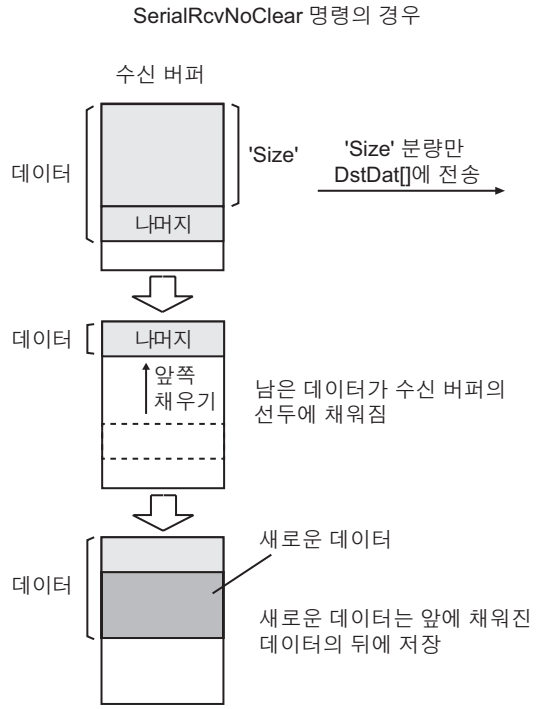
SerialRcv 명령과 SerialRcvNoClear 명령의 차이

SerialRcv 명령과 SerialRcvNoClear 명령은 수신 버퍼에서 DstDat[]에 데이터가 전송된 후 수신 버퍼가 전체 클리어되는지 여부에 서로 차이가 있습니다.

SerialRcv 명령은 데이터 전송 후 수신 버퍼가 전체 클리어됩니다. 따라서 수신 버퍼 내의 데이터가 'Size'보다 많은 경우, 전송 후에 남은 데이터는 폐기됩니다.



SerialRcvNoClear 명령은 데이터 전송 후 전송된 데이터만 클리어하며, 수신 버퍼에 남은 데이터는 수신 버퍼의 선두로 이동하여 채워집니다. 그 후 새로 수신 버퍼에 들어간 데이터는 앞 부분에 채워진 데이터의 뒤에 저장됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Port_numUsingPort	사용 포트 수	USINT	현재 사용 중인 포트 수
_Port_isAvailable	네트워크 통신 명령 실행 가능 플래그	BOOL	TRUE : 사용 가능 포트 있음 FALSE : 사용 가능 포트 없음

관련된 준 사용자 정의 변수

변수명	명칭	데이터 형식	내용
P#_NopRcvOvfSta(*)	수신 초과 플래그	BOOL	TRUE : 유니트가 지정된 수신 데이터 수 이상을 수신 (수신 완료 플래그가 TRUE가 된 후의 수신 있음) FALSE : 유니트가 지정된 수신 데이터 수 이상을 수신하지 않음
P#_NopRcvCompleteSta(*)	수신 완료 플래그	BOOL	TRUE : 수신 완료 FALSE : 미수신 또는 수신 중
P#_NopRcvCntSta(*)	수신 카운터	UINT	16#0000~16#0100 : 수신 데이터의 바이트 수
P#_NopStartCodeYNCfg(*)	무수순 시작 코드 유무	BOOL	TRUE : 있음 FALSE : 없음
P#_NopEndCodeYNCfg(*)	무수순 종료 코드 유무	BOOL	TRUE : 있음 FALSE : 없음
P#_NopCRLFCfg(*)	무수순 CRLF 지정	BOOL	TRUE : 지정함 FALSE : 지정하지 않음
P#_NopRcvDatSzCfg(*)	무수순 수신 데이터 수	USINT	16#01~16#FF : 1~255바이트 16#00 : 256바이트
P#_NopStartCodeCfg(*)	무수순 시작 코드	USINT	16#00~16#FF
P#_NopEndCodeCfg(*)	무수순 종료 코드	USINT	16#00~16#FF
P#_TransErr(*)	전송 이상	BOOL	TRUE : 이상 발생 있음 FALSE : 이상 발생 없음
P#_OverRunErr(*)	오버런 이상	BOOL	TRUE : 이상 발생 있음 FALSE : 이상 발생 없음

* #에는 시리얼 커뮤니케이션 유니트의 포트 번호가 들어갑니다.

참고

- 수신 완료 플래그 'P#_NopRcvCompleteSta'의 값이 TRUE가 되는 것은 다음과 같은 경우입니다.
  - 무수순 수신 데이터 수 'P#NopRcvDatSzCfg'의 설정값만큼의 데이터를 수신함
  - 지정한 종료 코드를 수신함
  - 256바이트의 데이터를 수신함
- 수신 초과 플래그 'P#_NopRcvOvfSta'의 값이 TRUE가 되는 것은 다음과 같은 경우입니다.
  - 수신 완료 플래그 'P#_NopRcvCompleteSta'의 값이 TRUE인 상태에서 이 명령을 실행하지 않고 데이터를 수신함
  - 무수순 수신 데이터 수 'P#NopRcvDatSzCfg'의 설정값을 초과하는 데이터를 수신함
- 무수순 통신에 대한 자세한 내용은 다음 설명서를 참조하십시오.
  - 🕮 "CJ 시리즈 시리얼 커뮤니케이션 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-354)"

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 수신 완료 플래그 'P#_NopRcvCompleteSta'의 값이 TRUE일 때 실행하십시오.
- 데이터 수신 시에는 반드시 이 명령을 실행하여 수신 버퍼의 데이터를 DstDat[]로 전송하십시오. 그렇게 하지 않으면 다음 데이터를 수신할 수 없습니다.
- 259바이트의 데이터를 수신하면 자동으로 수신을 정지합니다. 그 후에 이 명령을 실행하지 않고 데이터를 다시 수신하면 오버런 이상 'P#_OverRunErr'의 값이 TRUE가 됩니다.
- 시작 코드, 종료 코드를 부가하는 경우, 'Size'에는 시작 코드, 종료 코드를 포함하지 않는 값을 설정하십시오.
- 이 명령은 무수순 모드로 설정한 시리얼 커뮤니케이션 유니트의 시리얼 포트에만 사용할 수 있습니다.
- 'Size'의 값을 0으로 하면 수신 버퍼의 데이터를 DstDat[]에 전송하지 않습니다. 이 경우, 수신 완료 플래그 'P#_NopRcvCompleteSta'와 수신 초과 플래그 'P#_NopRcvOvfSta'의 값은 FALSE가 됩니다. 또한 수신 카운터 'P#_NopRcvCntSta'의 값은 0이 됩니다.
- 이 명령은 포트가 사용 가능한 경우에만 실행할 수 있습니다. 따라서 이 명령의 입력 조건에 네트워크 통신 명령 실행 가능 플래그의 시스템 정의 변수 '_Port_isAvailable'을 a 접점으로 삽입하십시오.
- 이 명령은 수신 완료 플래그 'P#_NopRcvCompleteSta'의 값이 TRUE일 때 실행하십시오.
- 이 명령은 'Busy'의 값이 TRUE인 동안에는 실행할 수 없습니다. 따라서 이 명령의 입력 조건으로 'Busy'를 b 접점으로 삽입하십시오.
- 이 명령을 ST 프로그램으로 기술하는 경우, 이 명령 실행 중에는 태스크 주기마다 이 명령의 제어가 수행되도록 하십시오. 그렇게 하지 않으면 정상적인 처리를 할 수 없는 경우가 있습니다.
- SerialRcv 명령을 실행하면 시리얼 커뮤니케이션 유니트의 수신 버퍼가 전체 클리어됩니다. 따라서 수신 버퍼의 데이터를 분할하여 DstDat[]에 전송할 수는 없습니다.
- SerialRcv 명령의 경우, 'Size'로 지정된 크기 이상의 수신 데이터는 다음 SerialRcv 명령 실행 시 폐기됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 시리얼 통신 모드가 무수순 이외이고 이 명령이 실행된 경우
  - '_Port_isAvailable'의 값이 FALSE인 경우
  - 'Port.UnitNo' 또는 'Port.PhysicPortNo'의 값이 유효 범위 밖인 경우
  - 지정한 유니트 번호에 CJ 시리얼 커뮤니케이션 유니트가 장착되지 않은 경우
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Size'의 값이 DstDat[]의 크기를 초과한 경우
  - 통신이 실패한 경우
  - 유니트 재시작 중에 이 명령을 실행한 경우

- 확장 에러 코드 'ErrorIDEx'는 이 명령에서 통신 응답 코드로서의 의미를 갖습니다. 그 값과 이상 내용은 다음과 같습니다. 'ErrorIDEx'는 에러 코드 'ErrorID'의 값이 WORD#16#0800인 경우에 출력됩니다.

값	이상 내용	대처 방법
16#00000001	통신 서비스를 중단했습니다.	• 데이터 링크 기동 상태를 확인하십시오. • 제3노드의 전송처 영역의 용량을 확인하십시오.
16#00000101	로컬 노드가 네트워크에 가입하지 않았습니다.	로컬 노드를 네트워크에 가입시키십시오.
16#00000102	토큰 타임아웃이 발생했습니다.	로컬 노드를 최대 노드 주소 범위 내로 설정하십시오.
16#00000103	재전송 초과가 발생했습니다.	노드간 테스트를 실행하여 정상이 아닌 경우, 사용 환경을 확인하십시오.
16#00000104	송신 허가 프레임 수 초과가 발생했습니다.	네트워크 내에서 실행되고 있는 이벤트의 상황을 확인하고 1태스크 주기 내의 이벤트 수를 줄이십시오. 또는 송신 허가 프레임 수를 늘리십시오.
16#00000105	로컬 노드의 노드 주소가 설정 범위 밖입니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000106	로컬 노드의 노드 주소가 네트워크 내에서 중복되어 있습니다.	중복되어 있는 노드의 주소를 변경하십시오.
16#00000201	상대 노드가 네트워크에 가입하지 않았습니다.	상대 노드를 네트워크에 가입시키십시오.
16#00000202	송신대상으로 지정한 호기 주소의 유니트가 존재하지 않습니다.	송신대상 네트워크 주소의 호기 주소를 정확하게 설정하십시오.
16#00000203	제3노드가 네트워크에 가입하지 않았습니다.	• 제3노드로 지정된 유니트의 주소를 확인하십시오. • 제3노드에는 1노드만 지정하십시오.
16#00000204	상대 노드가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 상대 노드에 집중되지 않도록 시스템을 재검토하십시오.
16#00000205	응답 타임아웃이 발생했습니다.	통신 파라메터의 설정을 확인하십시오.
16#00000206	전송로에 이상이 있습니다.	• 다시 시도하십시오. • 이상이 자주 발생하는 경우는 노이즈의 상황을 확인하십시오.
16#00000301	통신 컨트롤러 이상이 발생했습니다.	해당 유니트의 사용 설명서를 참조하여 적절하게 처리하십시오.
16#00000302	상대 노드의 CPU 유니트에 이상이 있습니다.	상대 노드의 CPU 유니트 설명서를 참조하여 이상을 제거하십시오.
16#00000303	해당 컨트롤러에 이상이 있어서 응답이 반환되지 않습니다.	네트워크의 통신 상황을 확인한 후 해당 컨트롤러를 재기동하십시오. 그래도 이상이 발생하는 경우는 해당 컨트롤러를 교환하십시오.
16#00000304	유니트 번호 설정이 잘못되었습니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000401	송신한 명령이 지원되지 않습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00000402	유니트의 기종 또는 버전이 지원되지 않습니다.	유니트의 기종과 버전을 확인하십시오.
16#00000501	상대 주소 설정 이상입니다.	라우팅 테이블에 수신대상 주소를 설정하십시오.
16#00000502	라우팅 테이블이 등록되지 않았습니다.	송신원 노드, 수신대상 노드, 중계 노드에 라우팅 테이블을 설정하십시오.
16#00000503	라우팅 테이블 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.

값	이상 내용	대처 방법
16#00000504	중계 횟수 초과가 발생했습니다.	네트워크를 재구성하거나 라우팅 테이블을 재검토하여 명령의 사용 범위를 3계층 이내로 하십시오.
16#00001001	명령 길이가 최대 명령 길이를 초과했습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001002	명령 길이가 최소 명령 길이 미만입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001003	명령에 지정된 쓰기 요소 수와 실제 쓰기 데이터 수가 일치하지 않습니다.	쓰기 요소 수와 실제 쓰기 데이터 수를 일치시키십시오.
16#00001004	명령 형식 이상입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001005	헤더 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.
16#00001101	영역 종류가 없습니다.	명령의 변수, 파라메터 종류 코드를 참조한 후 해당하는 코드를 설정하십시오.
16#00001102	액세스 크기 이상입니다.	변수, 파라메터의 액세스 크기를 정확하게 설정하십시오.
16#00001103	주소 범위 밖이 지정되었습니다.	처리 가능 범위 내의 주소를 지정하십시오.
16#00001104	주소 범위를 초과했습니다.	• 처리 가능 범위 내의 주소를 지정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#00001106	등록되지 않은 송수신 시퀀스 번호가 지정되었습니다.	송수신 시퀀스 번호를 수정하거나 CX-Protocol로 시퀀스를 추가하십시오.
16#00001109	상관 관계 이상이 발생했습니다.	• 명령 데이터의 대소 관계를 정확하게 설정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110A	데이터가 중복되어 있습니다.	• 실행 중인 처리를 중단하거나 종료할 때까지 기다린 후 명령을 실행하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110B	응답이 최대 응답 길이를 초과했습니다.	명령 배열 내의 요소 수를 정확하게 설정하십시오.
16#0000110C	기타 파라메터 이상	명령 배열의 내용을 정확하게 설정하십시오.
16#00002002	보호 중입니다.	보호를 해제한 후 명령을 다시 실행하십시오.
16#00002003	등록 테이블이 없습니다.	테이블을 정확하게 설정하십시오.
16#00002004	검색 데이터와 일치하는 데이터가 존재하지 않습니다.	검색 데이터를 정확하게 설정하십시오.
16#00002005	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002006	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002007	비교 이상이 발생했습니다.	• 메모리 내용을 확인하고 정확한 데이터로 재기입하십시오. • 파일 내용을 확인하십시오.
16#00002101	읽기 전용 영역이므로 액세스할 수 없습니다.	쓰기 방지를 해제한 후 명령을 다시 실행하십시오.

값	이상 내용	대처 방법
16#00002102	보호 중 또는 데이터 링크 테이블 쓰기 불가능 상태입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 시스템 설정을 데이터 링크 테이블에 설정하십시오.
16#00002103	등록할 수 없습니다.	• 필요 없는 파일을 삭제한 후 파일을 작성하거나 새 파일용 메모리를 준비하십시오. • 열려 있는 파일을 닫은 후 명령을 다시 실행하십시오.
16#00002105	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002106	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002107	동일한 파일명이 존재합니다.	쓸 파일의 파일명을 변경하고 명령을 다시 실행하십시오.
16#00002108	변경에 의해 이상이 발생하므로 변경할 수 없습니다.	설정을 변경하십시오.
16#00002201	이미 프로토콜 매크로를 실행 중이므로 동작 불가능합니다.	프로토콜 매크로 실행 중 플래그를 b 접점에 설정하십시오.
16#00002202	동작 모드 이상입니다.	동작 모드를 확인하십시오.
16#00002203	명령의 동작 모드가 다릅니다(프로그램 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002204	명령의 동작 모드가 다릅니다(디버그 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002205	명령의 동작 모드가 다릅니다(모니터 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002206	명령의 동작 모드가 다릅니다(운전 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002207	지정 노드가 관리국이 아닙니다.	네트워크의 관리국 노드를 확인하십시오.
16#00002208	명령의 동작 모드가 다릅니다.	스텝 활성 상태를 확인하십시오.
16#00002211	유니트가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 해당 유니트에 집중되지 않도록 시스템을 재검토하십시오.
16#00002301	파일 장치가 없습니다.	미디어를 장착하십시오. 또는 EM 포맷을 실시하십시오.
16#00002302	파일 메모리가 없습니다.	파일 메모리 장착을 확인하십시오.
16#00002303	시계가 내장되어 있지 않습니다.	기종의 사양을 확인하십시오.
16#00002401	프로토콜 매크로 데이터에서 SUM 값 이상이 발생했거나 데이터 전송 중입니다.	CX-Protocol로 프로토콜 매크로 데이터를 다시 전송하십시오.
16#00002502	처리 대상 메모리에 이상이 있습니다.	대상 메모리에 정확한 데이터를 다시 전송하십시오.
16#00002503	등록된 I/O 유니트 구성과 실제 유니트의 구성이 서로 다릅니다	I/O 유니트 구성을 확인하십시오.
16#00002504	등록된 I/O 점수 또는 리모트 I/O 점수가 최대값을 초과했습니다.	I/O 점수, 리모트 I/O 점수를 정확하게 설정하십시오.
16#00002505	CPU 유니트와 CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#00002506	랙 번호, 유니트 번호, I/O 주소 중 하나의 설정이 중복되어 있습니다.	중복된 내용을 다시 설정하십시오.
16#00002507	CPU 유니트와 I/O 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.

값	이상 내용	대처 방법
16#00002509	SYSMAC BUS/2 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250A	CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250D	채널 설정이 중복되어 있습니다.	I/O 채널을 정확하게 설정하십시오.
16#0000250F	메모리 이상입니다.	• 내부 메모리의 경우, 정확한 데이터를 쓰고 명령을 다시 실행하십시오. • 메모리 카드나 EM 파일 메모리의 경우, 확장 메모리의 포맷 명령을 실행하십시오. • 상기의 처리를 실시해도 이상이 해제되지 않는 경우는 메모리를 교환하십시오.
16#00002510	종단국 설정 이상입니다.	종단국을 정확하게 설정하십시오.
16#00002601	보호 해제 중입니다.	보호를 해제할 필요가 없습니다.
16#00002602	비밀번호가 일치하지 않습니다.	정확한 비밀번호를 지정하십시오.
16#00002604	보호 중입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002605	서비스 실행 중입니다.	실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002606	서비스 정지 중입니다.	필요에 따라 해당 서비스를 실행하십시오.
16#00002607	실행 권한이 없습니다.	• 데이터 링크 참가 노드에서 실행하십시오. • 재기동해도 이상이 재발하는 경우는 컨트롤러를 교환하십시오.
16#00002608	환경이 설정되지 않았습니다.	필요한 설정을 하십시오.
16#00002609	필요한 항목이 설정되지 않았습니다.	필요한 항목을 설정하십시오.
16#0000260A	지정 번호의 정의가 완료되었습니다.	등록되지 않은 작업이나 트랜지션 번호로 변경하고 명령을 다시 실행하십시오.
16#0000260B	이상 해제가 불가능합니다.	이상의 원인이 된 요인을 해제하고 이상 해제 명령을 실행하십시오.
16#00003001	액세스 권한이 없습니다.	액세스 권한이 해제될 때까지 기다린 후 명령을 다시 실행하십시오.
16#00004001	서비스가 중단되었습니다.	서비스가 중단된 요인을 해제하고 명령을 다시 실행하십시오.

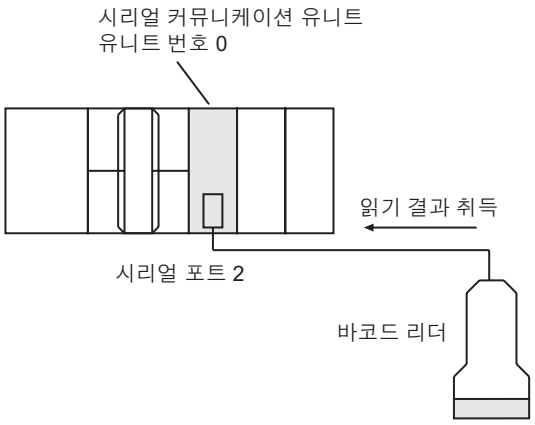
(주의) 위 표의 값 이외에 종료 코드의 비트 6, 7, 15의 값이 TRUE가 되는 경우가 있습니다. 비트 6, 7의 값이 TRUE인 경우는 송신대상 CPU 유니트에 이상이 발생했음을 나타냅니다. 비트 15의 값이 TRUE인 경우는 네트워크 중계 시 이상이 발생했음을 나타냅니다.

버전 관련 정보

SerialRcvNoClear 명령은 CPU 유니트 Ver.1.03 이상, Sysmac Studio Ver.1.04 이상, 시리얼 커뮤니케이션 유니트 Ver.2.1 이상의 조합에서 사용할 수 있습니다.

샘플 프로그램

CJ 시리즈의 시리얼 커뮤니케이션 유니트(유니트 번호 0, 장치명 'Barcode')의 시리얼 포트 2에 연결한 바코드 리더의 읽기 결과를 취득합니다.
수신 데이터는 배열 변수 RecvDat[]에 저장됩니다. 시작 코드는 없음, 종료 코드는 16#OD(CR)로 합니다.



시리얼 커뮤니케이션 유니트의 설정은 다음과 같이 합니다.

항목명	설정값
포트 2: 임의 설정 유무	임의 설정
포트 2: 시리얼 통신 모드	무수순
포트 2: 데이터 길이	8비트
포트 2: 정지 비트	1 비트
포트 2: 패리티	없음
포트 2: 전송 속도	38400bps
포트 2: 무수순 종료 코드	D
포트 2: 무수순 시작 코드 유무	없음
포트 2: 무수순 종료 코드 유무	있음(종료 코드를 임의 설정)

RecvDat[]에 저장되는 데이터는 바코드의 수치를 1문자씩 문자 코드로 표현한 비트 열입니다. RecvDat[]의 배열의 1개 요소가 바코드의 1문자에 대응합니다. 그것을 우선 AryToString 명령을 사용해서 문자열 RecvStringDat로 변환합니다. 그리고 STRING_TO_ULINT 명령을 사용하여 ULINT 형식 정수 Code로 변환합니다.



글로벌 변수의 정의

글로벌 변수

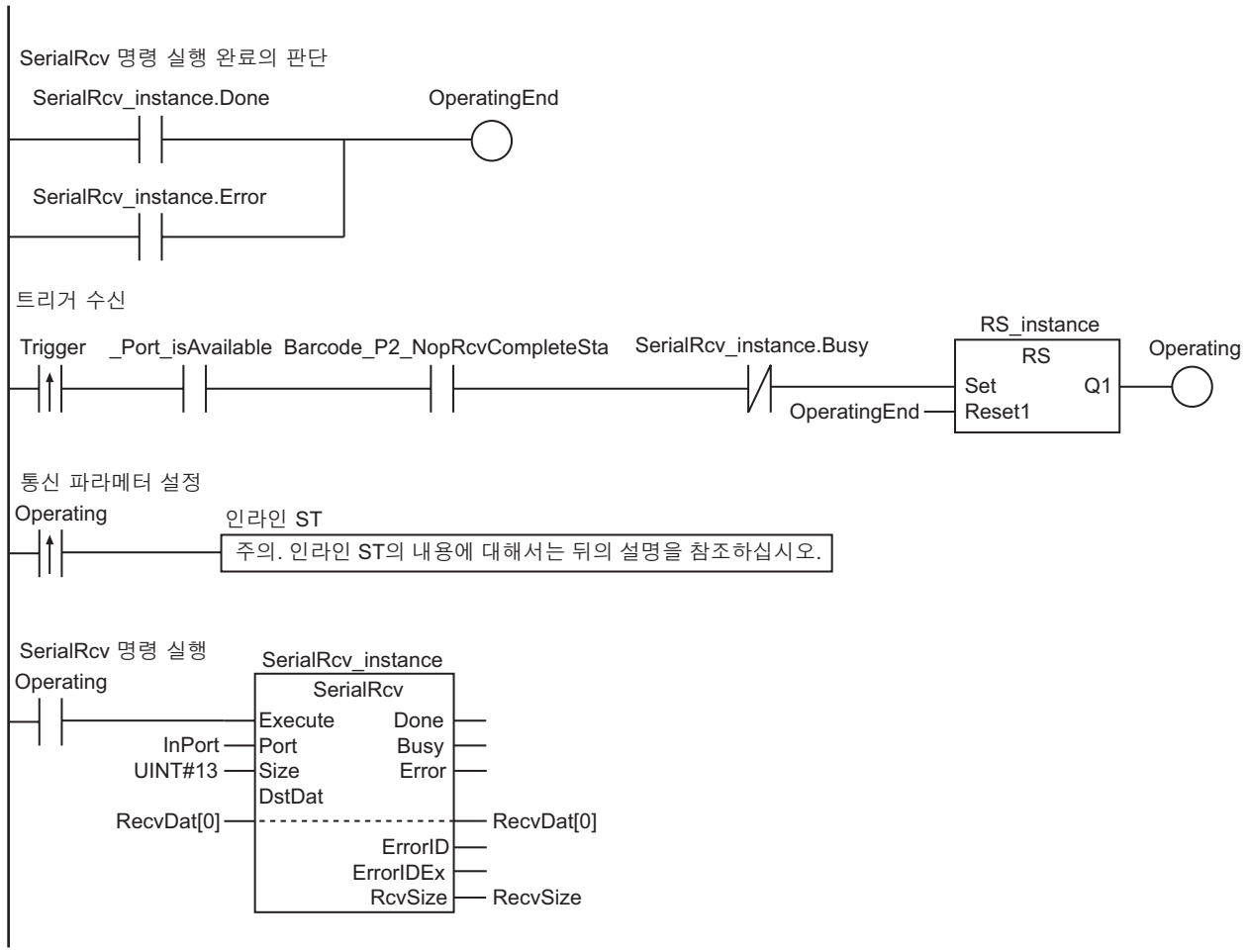
명칭	데이터 형식	할당대상*1	코멘트
Barcode_P2_NopRcvCompleteSta	BOOL	IOBus://rack#0/slot#0/P2_NopSta/P2_NopRcvCompleteSta	수신 완료 플래그

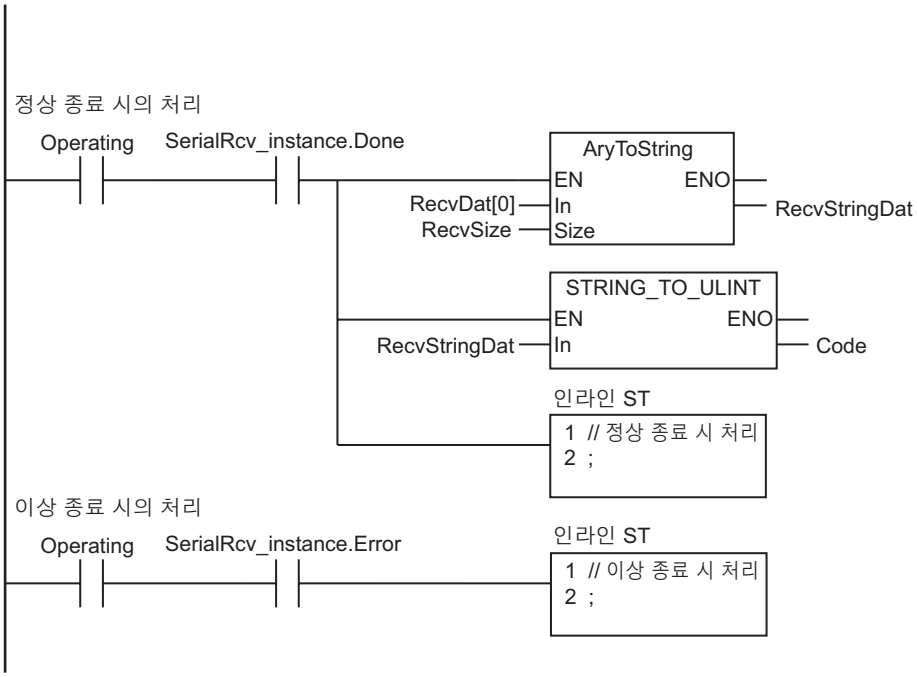
*1.시리얼 커뮤니케이션 유니트를 랙 번호 0의 슬롯 번호 0에 장착한 경우의 할당대상입니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)	포트 지정
	RecvDat	ARRAY[0..12] OF BYTE	[13(16#0)]	수신 데이터
	RecvSize	UINT	0	수신 데이터 크기
	RecvStringDat	STRING[255]	''	바코드 문자열
	Code	ULINT	0	바코드 정수
	RS_instance	RS		
	SerialRcv_instance	SerialRcv		

외부 변수	명칭	데이터 형식	코멘트
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그
	Barcode_P2_NopRcvCompleteSta	BOOL	수신 완료 플래그





● 인라인 ST의 내용

```
InPort.UnitNo       :=_CBU_No00;    // 시리얼 커뮤니케이션 유니트의 유니트 번호 0
InPort.PhysicPortNo :=USINT#2;      // 시리얼 포트 2
```

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	InPort	_sPORT	(UnitNo:=_CBU_No00, PhysicPortNo:=0)	포트 지정
	RecvDat	ARRAY[0..12] OF BYTE	[13(16#0)]	수신 데이터
	RecvSize	UINT	0	수신 데이터 크기
	RecvStringDat	STRING[255]	''	바코드 문자열
	Code	ULINT	0	바코드 정수
	SerialRcv_instance	SerialRcv		

외부 변수	명칭	데이터 형식	코멘트
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그
	Barcode_P2_NopRcvCompleteSta	BOOL	수신 완료 플래그

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Port_isAvailable=TRUE)
     AND (Barcode_P2_NopRcvCompleteSta=TRUE) AND (SerialRcv_instance.Busy=FALSE) ) THEN
   OperatingStart  :=TRUE;
   Operating       :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 통신 파라메터 설정과 SerialRcv 명령 초기화
IF (OperatingStart=TRUE) THEN
   SerialRcv_instance(
      Execute   :=FALSE,        // 인스턴스 초기화
      Port      :=InPort,       // 포트 지정
      Size      :=UINT#13,      // 수신 데이터 크기
      DstDat    :=RecvDat[0],   // 수신 데이터
      RcvSize   =>RecvSize);    // 실제로 수신한 데이터 크기
   InPort.UnitNo          :=_CBU_No00;     // 시리얼 커뮤니케이션 유니트의 유니트 번호 0
   InPort.PhysicPortNo :=USINT#2;          // 시리얼 포트 2
   OperatingStart         :=FALSE;
END_IF;

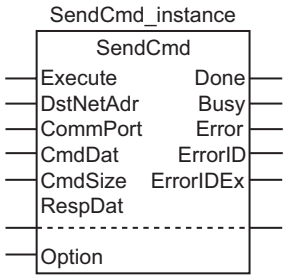
// SerialRcv 명령 실행
IF (Operating=TRUE) THEN
   SerialRcv_instance(
      Execute   :=TRUE,
      Port      :=InPort,
      Size      :=UINT#13,
      DstDat    :=RecvDat[0],
      RcvSize   =>RecvSize);

   IF (SerialRcv_instance.Done=TRUE) THEN
      // 정상 종료 시 처리
      RecvStringDat  :=AryToString(In:=RecvDat[0], Size:=RecvSize);  //문자 코드를 문자열로 변환
      Code           :=STRING_TO_ULINT(RecvStringDat);              // 문자열을 정수로 변환
      Operating      :=FALSE;
   END_IF;

   IF (SerialRcv_instance.Error=TRUE) THEN
      // 이상 종료 시 처리
      Operating:=FALSE;
   END_IF;
END_IF;
```

SendCmd

시리얼 게이트웨이 기능으로 시리얼 커뮤니케이션 유니트에 명령을 송신합니다. 또는 DeviceNet 유니트나 CompoNet 마스터 유니트에 Explicit 명령을 송신합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
SendCmd	명령 송신	FB		SendCmd_instance(Execute, DstNetAdr, CommPort, CmdDat, CmdSize, RespDat, Option, Done, Busy, Error, ErrorID, ErrorIDEx);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
DstNetAdr	송신대상 네트워크 주소	입력	송신대상 네트워크 주소	–	–	–
CommPort	송신대상 시리얼 포트 지정		송신대상 시리얼 포트의 지정	_NONE만		_NONE
CmdDat[] 배열	명령 배열		송신할 명령	데이터 형식에 따름		(*)
CmdSize	명령 데이터 크기		명령 데이터 크기	2~최대 데이터 길이(네트워크 종류에 따름)	바이트	2
Option	응답		응답 감시 및 재송신 지정	–	–	–
RespDat[] 배열	응답 저장 배열	입출력	응답을 저장하는 배열	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
DstNetAdr	구조체 _sDNET_ADR에 대한 자세한 내용은 기능 설명 참조																			
CommPort	열거 형식 _ePORT 열거자는 기능 설명 참조																			
CmdDat[] 배열		○																		
CmdSize							○													
Option	구조체 _sRESPONSE에 대한 자세한 내용은 기능 설명 참조																			
RespDat[] 배열		○																		

기능

송신대상 네트워크 주소 'DstNetAdr', 송신대상 시리얼 포트 지정 'CommPort'로 지정한 송신대상에 명령 배열 CmdDat[]의 내용을 전송합니다.
CmdDat[]의 몇 번째 요소까지를 명령으로 간주할 것인지는 명령 데이터 크기 'CmdSize'로 지정합니다.
그 후에 반환된 응답을 응답 저장 배열 RespDat[]에 저장합니다.

'DstNetAdr'의 데이터 형식은 구조체 _sDNET_ADR입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
DstNetAdr	송신대상 네트워크 주소	송신대상 네트워크 주소	_sDNET_ADR	–	–	–
NetNo	네트워크 주소	네트워크 주소	USINT	0만	–	0
NodeNo	노드 주소	노드 주소	USINT			
UnitNo	호기 주소	호기 주소	BYTE	데이터 형식에 따름		16#00

'CommPort'의 데이터 형식은 열거 형식 _ePORT입니다.
열거 형식 _ePORT의 열거자의 의미는 다음과 같습니다.

열거자	의미
_NONE	송신대상이 시리얼 포트(상위 링크 모드)가 아님

'Option'의 데이터 형식은 구조체 _sRESPONSE입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
Option	응답	응답 감시 및 재송신 지정	_sRESPONSE	–	–	–
isNonResp	응답 필요 없음	TRUE : 응답 필요 없음 FALSE : 응답 필요	BOOL	데이터 형식에 따름	–	FALSE
TimeOut	타임아웃 시간	타임아웃 시간 0인 경우는 2.0s	UINT		0.1s	20 (2.0s)
Retry	재송신 횟수	재송신 횟수	USINT	0~15	회	0

응답 필요 없음 플래그 'Option.isNonResp'의 값이 FALSE인 경우, 타임아웃 시간 'Option.TimeOut'의 값 이내에 응답이 없으면 응답이 있을 때까지 명령을 재송신합니다. 재송신 횟수는 'Option.Retry'로 지정합니다.
타임아웃 시간은 'Option.TimeOut'×0.1s입니다. 단, 'Option.TimeOut'의 값이 0인 경우는 2.0s가 됩니다.
'Option.TimeOut'의 초기값도 2.0s입니다.

관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Port_numUsingPort	사용 포트 수	USINT	현재 사용 중인 포트 수
_Port_isAvailable	네트워크 통신 명령 실행 가능 플래그	BOOL	TRUE : 사용 가능 포트 있음 FALSE : 사용 가능 포트 없음

참고

- 노이즈 등의 영향으로 통신 도중에 명령 또한 응답이 손실되는 경우가 있습니다. 그러므로 'Option.Retry'의 값을 0 이외로 하여 응답이 반환되지 않는 경우에는 재송신 처리를 수행하도록 하면 신뢰성이 향상됩니다.
- 시리얼 게이트웨이 기능으로 시리얼 포트를 지정하는 경우, 'DstNetAdr.UnitNo'에 시리얼 포트의 호기 주소를 설정합니다. 시리얼 커뮤니케이션 유니트의 포트의 호기 주소는 다음과 같습니다.
  - 포트 1을 사용하는 경우
    호기 주소=BYTE#16#80+BYTE#16#04×유니트 번호(16진수)
    (예) 유니트 번호 1의 경우
    BYTE#16#80+BYTE#16#04×1＝BYTE#16#84
  - 포트 2를 사용하는 경우
    호기 주소=BYTE#16#81+BYTE#16#04×유니트 번호(16진수)
    (예) 유니트 번호 2의 경우
    BYTE#16#81+BYTE#16#04×2＝BYTE#16#89

사용상의 주의 사항

- 이 명령은 포트가 사용 가능한 경우에만 실행할 수 있습니다. 따라서 이 명령의 입력 조건에 네트워크 통신 명령 실행 가능 플래그의 시스템 정의 변수 '_Port_isAvailable'을 a 접점으로 삽입하십시오.
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 이 명령을 ST 프로그램으로 기술하는 경우, 이 명령 실행 중에는 태스크 주기마다 이 명령의 제어가 수행되도록 하십시오. 그렇게 하지 않으면 정상적인 처리를 할 수 없는 경우가 있습니다.
- 'CmdSize'의 값이 0인 경우, 명령을 전송하지 않습니다. 그 경우, 명령 실행 시에 'Dome'의 값이 TRUE가 됩니다.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 📖 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'CommPort'의 값이 유효 범위 밖인 경우
  - 'DstNetAdr'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'CmdSize'의 값이 유효 범위 밖인 경우
  - 'Option'의 멤버 중 하나가 유효 범위 밖인 경우
  - 'CmdSize'의 값이 CmdDat[]의 크기를 초과한 경우
  - 응답의 크기가 RespDat[]의 크기를 초과한 경우
  - '_Port_isAvailable'의 값이 FALSE인 경우
  - 통신이 실패한 경우

? 확장 에러 코드 'ErrorIDEx' 는 이 명령에서 통신 응답 코드로서의 의미를 갖습니다 . 그 값과 이상 내용은 다음과 같습니다 . 'ErrorIDEx' 는 에러 코드 'ErrorID' 의 값이 WORD#16#0800 인 경우에 출력됩니다 .

값	이상 내용	대처 방법
16#00000001	통신 서비스를 중단했습니다.	• 데이터 링크 기동 상태를 확인하십시오. • 제3노드의 전송처 영역의 용량을 확인하십시오.
16#00000101	로컬 노드가 네트워크에 가입하지 않았습니다.	로컬 노드를 네트워크에 가입시키십시오.
16#00000102	토큰 타임아웃이 발생했습니다.	로컬 노드를 최대 노드 주소 범위 내로 설정하십시오.
16#00000103	재전송 초과가 발생했습니다.	노드간 테스트를 실행하여 정상이 아닌 경우, 사용 환경을 확인하십시오.
16#00000104	송신 허가 프레임 수 초과가 발생했습니다.	네트워크 내에서 실행되고 있는 이벤트의 상황을 확인하고 1태스크 주기 내의 이벤트 수를 줄이십시오. 또는 송신 허가 프레임 수를 늘리십시오.
16#00000105	로컬 노드의 노드 주소가 설정 범위 밖입니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000106	로컬 노드의 노드 주소가 네트워크 내에서 중복되어 있습니다.	중복되어 있는 노드의 주소를 변경하십시오.
16#00000201	상대 노드가 네트워크에 가입하지 않았습니다.	상대 노드를 네트워크에 가입시키십시오.
16#00000202	송신대상으로 지정한 호기 주소의 유니트가 존재하지 않습니다.	송신대상 네트워크 주소의 호기 주소를 정확하게 설정하십시오.
16#00000203	제3노드가 네트워크에 가입하지 않았습니다.	• 제3노드로 지정된 유니트의 주소를 확인하십시오. • 제3노드에는 1노드만 지정하십시오.
16#00000204	상대 노드가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 상대 노드에 집중되지 않도록 시스템을 재검토하십시오.
16#00000205	응답 타임아웃이 발생했습니다.	통신 파라메터의 설정을 확인하십시오.
16#00000206	전송로에 이상이 있습니다.	• 다시 시도하십시오. • 이상이 자주 발생하는 경우는 노이즈의 상황을 확인하십시오.
16#00000301	통신 컨트롤러 이상이 발생했습니다.	해당 유니트의 사용 설명서를 참조하여 적절하게 처리하십시오.
16#00000302	상대 노드의 CPU 유니트에 이상이 있습니다.	상대 노드의 CPU 유니트 설명서를 참조하여 이상을 제거하십시오.
16#00000303	해당 컨트롤러에 이상이 있어서 응답이 반환되지 않습니다.	네트워크의 통신 상황을 확인한 후 해당 컨트롤러를 재기동하십시오. 그래도 이상이 발생하는 경우는 해당 컨트롤러를 교환하십시오.
16#00000304	유니트 번호 설정이 잘못되었습니다.	시리얼 커뮤니케이션 유니트의 로터리 스위치를 정확하게 설정하십시오.
16#00000401	송신한 명령이 지원되지 않습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00000402	유니트의 기종 또는 버전이 지원되지 않습니다.	유니트의 기종과 버전을 확인하십시오.
16#00000501	상대 주소 설정 이상입니다.	라우팅 테이블에 수신대상 주소를 설정하십시오.
16#00000502	라우팅 테이블이 등록되지 않았습니다.	송신원 노드, 수신대상 노드, 중계 노드에 라우팅 테이블을 설정하십시오.
16#00000503	라우팅 테이블 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.

값	이상 내용	대처 방법
16#00000504	중계 횟수 초과가 발생했습니다.	네트워크를 재구성하거나 라우팅 테이블을 재검토하여 명령의 사용 범위를 3계층 이내로 하십시오.
16#00001001	명령 길이가 최대 명령 길이를 초과했습니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001002	명령 길이가 최소 명령 길이 미만입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001003	명령에 지정된 쓰기 요소 수와 실제 쓰기 데이터 수가 일치하지 않습니다.	쓰기 요소 수와 실제 쓰기 데이터 수를 일치시키십시오.
16#00001004	명령 형식 이상입니다.	명령 배열의 내용을 정확하게 설정하십시오.
16#00001005	헤더 이상입니다.	라우팅 테이블을 정확하게 설정하십시오.
16#00001101	영역 종류가 없습니다.	명령의 변수, 파라메터 종류 코드를 참조한 후 해당하는 코드를 설정하십시오.
16#00001102	액세스 크기 이상입니다.	변수, 파라메터의 액세스 크기를 정확하게 설정하십시오.
16#00001103	주소 범위 밖이 지정되었습니다.	처리 가능 범위 내의 주소를 지정하십시오.
16#00001104	주소 범위를 초과했습니다.	• 처리 가능 범위 내의 주소를 지정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#00001106	등록되지 않은 송수신 시퀀스 번호가 지정되었습니다.	송수신 시퀀스 번호를 수정하거나 CX-Protocol로 시퀀스를 추가하십시오.
16#00001109	상관 관계 이상이 발생했습니다.	• 명령 데이터의 대소 관계를 정확하게 설정하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110A	데이터가 중복되어 있습니다.	• 실행 중인 처리를 중단하거나 종료할 때까지 기다린 후 명령을 실행하십시오. • 데이터 링크 테이블을 정확하게 설정하십시오.
16#0000110B	응답이 최대 응답 길이를 초과했습니다.	명령 배열 내의 요소 수를 정확하게 설정하십시오.
16#0000110C	기타 파라메터 이상	명령 배열의 내용을 정확하게 설정하십시오.
16#00002002	보호 중입니다.	보호를 해제한 후 명령을 다시 실행하십시오.
16#00002003	등록 테이블이 없습니다.	테이블을 정확하게 설정하십시오.
16#00002004	검색 데이터와 일치하는 데이터가 존재하지 않습니다.	검색 데이터를 정확하게 설정하십시오.
16#00002005	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002006	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002007	비교 이상이 발생했습니다.	• 메모리 내용을 확인하고 정확한 데이터로 재기입하십시오. • 파일 내용을 확인하십시오.
16#00002101	읽기 전용 영역이므로 액세스할 수 없습니다.	쓰기 방지를 해제한 후 명령을 다시 실행하십시오.

값	이상 내용	대처 방법
16#00002102	보호 중 또는 데이터 링크 테이블 쓰기 불가능 상태입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 시스템 설정을 데이터 링크 테이블에 설정하십시오.
16#00002103	등록할 수 없습니다.	• 필요 없는 파일을 삭제한 후 파일을 작성하거나 새 파일용 메모리를 준비하십시오. • 열려 있는 파일을 닫은 후 명령을 다시 실행하십시오.
16#00002105	해당 프로그램 번호가 없습니다.	유효한 프로그램 번호를 설정하십시오.
16#00002106	해당 파일이 없습니다.	하위 디렉토리명을 포함하여 파일명을 정확하게 설정하십시오.
16#00002107	동일한 파일명이 존재합니다.	쓸 파일의 파일명을 변경하고 명령을 다시 실행하십시오.
16#00002108	변경에 의해 이상이 발생하므로 변경할 수 없습니다.	설정을 변경하십시오.
16#00002201	이미 프로토콜 매크로를 실행 중이므로 동작 불가능합니다.	프로토콜 매크로 실행 중 플래그를 b 접점에 설정하십시오.
16#00002202	동작 모드 이상입니다.	동작 모드를 확인하십시오.
16#00002203	명령의 동작 모드가 다릅니다(프로그램 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002204	명령의 동작 모드가 다릅니다(디버그 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002205	명령의 동작 모드가 다릅니다(모니터 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002206	명령의 동작 모드가 다릅니다(운전 모드).	컨트롤러의 동작 모드를 확인하십시오.
16#00002207	지정 노드가 관리국이 아닙니다.	네트워크의 관리국 노드를 확인하십시오.
16#00002208	명령의 동작 모드가 다릅니다.	스텝 활성 상태를 확인하십시오.
16#00002211	유니트가 다른 작업 중입니다.	재전송 횟수 설정을 늘리거나 통신이 해당 유니트에 집중되지 않도록 시스템을 재검토하십시오.
16#00002301	파일 장치가 없습니다.	미디어를 장착하십시오. 또는 EM 포맷을 실시하십시오.
16#00002302	파일 메모리가 없습니다.	파일 메모리 장착을 확인하십시오.
16#00002303	시계가 내장되어 있지 않습니다.	기종의 사양을 확인하십시오.
16#00002401	프로토콜 매크로 데이터에서 SUM 값 이상이 발생했거나 데이터 전송 중입니다.	CX-Protocol로 프로토콜 매크로 데이터를 다시 전송하십시오.
16#00002502	처리 대상 메모리에 이상이 있습니다.	대상 메모리에 정확한 데이터를 다시 전송하십시오.
16#00002503	등록된 I/O 유니트 구성과 실제 유니트의 구성이 서로 다릅니다	I/O 유니트 구성을 확인하십시오.
16#00002504	등록된 I/O 점수 또는 리모트 I/O 점수가 최대값을 초과했습니다.	I/O 점수, 리모트 I/O 점수를 정확하게 설정하십시오.
16#00002505	CPU 유니트와 CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#00002506	랙 번호, 유니트 번호, I/O 주소 중 하나의 설정이 중복되어 있습니다.	중복된 내용을 다시 설정하십시오.
16#00002507	CPU 유니트와 I/O 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.

값	이상 내용	대처 방법
16#00002509	SYSMAC BUS/2 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250A	CPU 고기능 유니트의 데이터 전송에 이상이 있습니다.	유니트나 연결 케이블을 확인하십시오. 이상을 해제한 후 이상 해제 명령을 실행하십시오.
16#0000250D	채널 설정이 중복되어 있습니다.	I/O 채널을 정확하게 설정하십시오.
16#0000250F	메모리 이상입니다.	• 내부 메모리의 경우, 정확한 데이터를 쓰고 명령을 다시 실행하십시오. • 메모리 카드나 EM 파일 메모리의 경우, 확장 메모리의 포맷 명령을 실행하십시오. • 상기의 처리를 실시해도 이상이 해제되지 않는 경우는 메모리를 교환하십시오.
16#00002510	종단국 설정 이상입니다.	종단국을 정확하게 설정하십시오.
16#00002601	보호 해제 중입니다.	보호를 해제할 필요가 없습니다.
16#00002602	비밀번호가 일치하지 않습니다.	정확한 비밀번호를 지정하십시오.
16#00002604	보호 중입니다.	• 쓰기 방지를 해제한 후 명령을 다시 실행하십시오. • 실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002605	서비스 실행 중입니다.	실행 중인 서비스가 종료될 때까지 기다리거나 서비스를 정지시킨 후 명령을 다시 실행하십시오.
16#00002606	서비스 정지 중입니다.	필요에 따라 해당 서비스를 실행하십시오.
16#00002607	실행 권한이 없습니다.	• 데이터 링크 참가 노드에서 실행하십시오. • 재기동해도 이상이 재발하는 경우는 컨트롤러를 교환하십시오.
16#00002608	환경이 설정되지 않았습니다.	필요한 설정을 하십시오.
16#00002609	필요한 항목이 설정되지 않았습니다.	필요한 항목을 설정하십시오.
16#0000260A	지정 번호의 정의가 완료되었습니다.	등록되지 않은 작업이나 트랜지션 번호로 변경하고 명령을 다시 실행하십시오.
16#0000260B	이상 해제가 불가능합니다.	이상의 원인이 된 요인을 해제하고 이상 해제 명령을 실행하십시오.
16#00003001	액세스 권한이 없습니다.	액세스 권한이 해제될 때까지 기다린 후 명령을 다시 실행하십시오.
16#00004001	서비스가 중단되었습니다.	서비스가 중단된 요인을 해제하고 명령을 다시 실행하십시오.

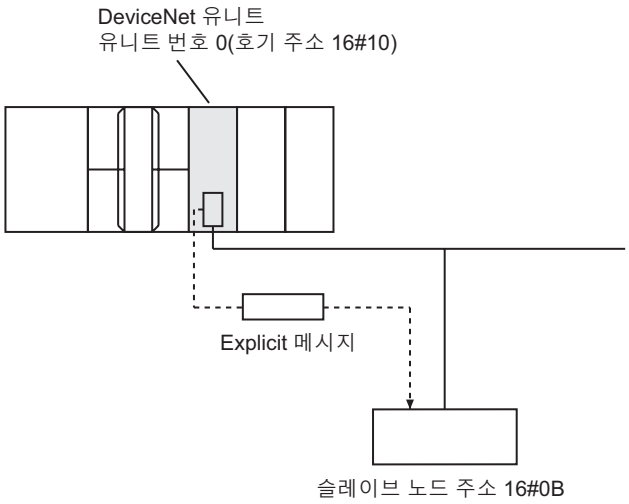
(주의) 위 표의 값 이외에 종료 코드의 비트 6, 7, 15의 값이 TRUE가 되는 경우가 있습니다. 비트 6, 7의 값이 TRUE인 경우는 송신대상 CPU 유니트에 이상이 발생했음을 나타냅니다. 비트 15의 값이 TRUE인 경우는 네트워크 중계 시 이상이 발생했음을 나타냅니다.

샘플 프로그램

DeviceNet 유니트를 경유하여 Explicit 메시지를 전송합니다. 호기 주소 16#10인 DeviceNet을 경유하여 노드 주소 16#0B의 슬레이브에서 벤더 ID를 읽습니다.

주요한 통신 사양이 아래에 나와 있습니다.

항목	내용
DeviceNet 유니트의 호기 주소	16#10
슬레이브 노드 주소	16#0B
서비스 코드	16#0E
클래스 ID	1
인스턴스 ID	1
속성 ID	1
타임아웃 시간	2.0s
재송신 횟수	2



명령 배열 SendDat[] 및 응답 저장 배열 RecvDat[]의 내용은 다음과 같습니다.

명령 배열 BYTE 형식 배열

배열 요소	항목	이 예의 내용	값
SendDat[0]	명령 코드	Explicit 메시지 송신 명령 코드는 16#2801입니다.	BYTE#16#28
SendDat[1]			BYTE#16#01
SendDat[2]	슬레이브 노드 주소	노드 주소 16#0B로 합니다.	BYTE#16#0B
SendDat[3]	서비스 코드	지정한 속성의 값 읽기(Get Attribute Single)의 서비스 코드는 16#0E입니다.	BYTE#16#0E
SendDat[4]	클래스 ID	Identity 개체의 클래스 ID는 16#0001입니다.	BYTE#16#00
SendDat[5]			BYTE#16#01
SendDat[6]	인스턴스 ID	–	BYTE#16#00
SendDat[7]			BYTE#16#01
SendDat[8]	속성 ID	벤더 ID의 속성 ID(Vendor ID)는 16#01입니다.	BYTE#16#01

응답 저장 배열 BYTE 형식 배열

배열 요소	항목	이 예의 내용
RecvDat[0]	명령 코드	Explicit 메시지 송신 명령 코드는 16#2801입니다.
RecvDat[1]		
RecvDat[2]	종료 코드	정상 종료인 경우, 16#0000입니다.
RecvDat[3]		
RecvDat[4]	슬레이브 노드 주소 이후의 수신 바이트 수	4바이트입니다.
RecvDat[5]		
RecvDat[6]	슬레이브 노드 주소	정상 종료인 경우, 노드 주소 16#0B가 됩니다.
RecvDat[7]	서비스 코드	정상 종료의 서비스 코드는 16#8E입니다.
RecvDat[8]	벤더 ID	슬레이브의 벤더 ID입니다.
RecvDat[9]		

글로벌 변수의 정의

글로벌 변수

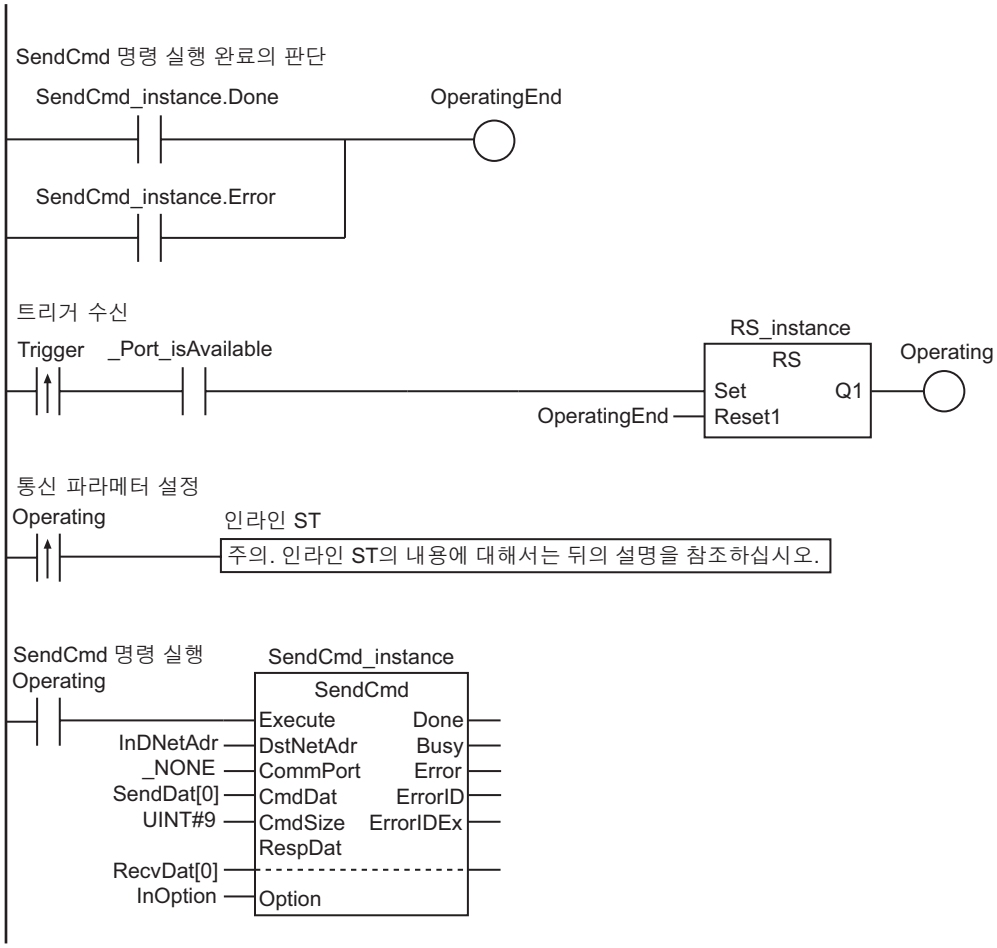
명칭	데이터 형식	할당대상*1	코멘트
DeviceNet_OnlineSta	BOOL	IOBus://rack#0/slot#0/Unit2Sta/OnlineSta	온라인

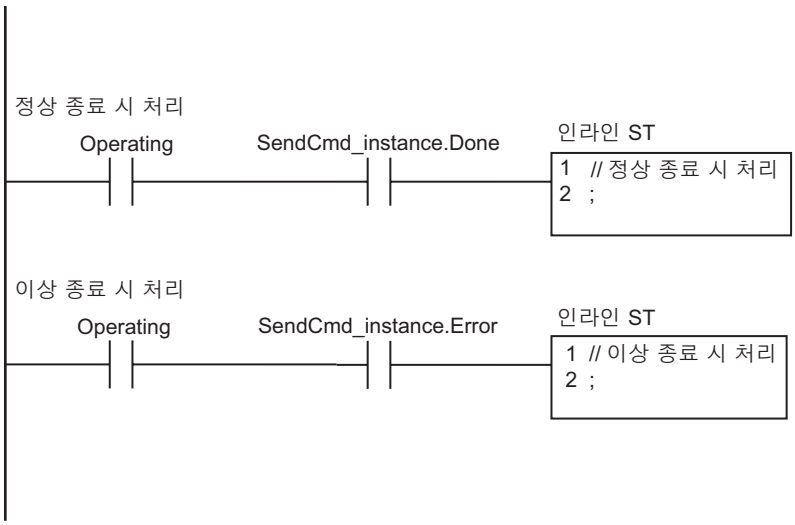
*1. 시리얼 커뮤니케이션 유니트를 랙 번호 0의 슬롯 번호 0에 장착한 경우의 할당대상입니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	InDNetAdr	_sDNET_ADR	(NetNo:=0, NodeNo:=0, UnitNo:=16#0)	송신대상 내트워크 주소
	InOption	_sRESPONSE	(isNonResp:=FALSE, TimeOut:=0, Retry:=0)	응답
	SendDat	ARRAY[0..8] OF BYTE	[9(16#0)]	송신 데이터
	RecvDat	ARRAY[0..9] OF BYTE	[10(16#0)]	수신 데이터
	RS_instance	RS		
	SendCmd_instance	SendCmd		

외부 변수	명칭	데이터 형식	코멘트
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그





● 인라인 ST의 내용

```
InDNetAdr.NetNo     :=USINT#0;        // 네트워크 주소 설정
InDNetAdr.NodeNo    :=USINT#0;
InDNetAdr.UnitNo    :=BYTE#16#10;
InOption.isNonResp  :=FALSE;          // 응답 설정
InOption.TimeOut    :=UINT#20;
InOption.Retry      :=USINT#2;
SendDat[0]          :=BYTE#16#28;     // 명령 배열 설정
SendDat[1]          :=BYTE#16#01;
SendDat[2]          :=BYTE#16#0B;
SendDat[3]          :=BYTE#16#0E;
SendDat[4]          :=BYTE#16#00;
SendDat[5]          :=BYTE#16#01;
SendDat[6]          :=BYTE#16#00;
SendDat[7]          :=BYTE#16#01;
SendDat[8]          :=BYTE#16#01;
```

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger 의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	InDNetAdr	_sDNET_ADR	(NetNo:=0, NodeNo:=0, UnitNo:=16#0)	송신대상 네트워크 주소
	InOption	_sRESPONSE	(isNonResp:=FALSE, TimeOut:=0, Retry:=0)	응답
	SendDat	ARRAY[0..8] OF BYTE	[9(16#0)]	송신 데이터
	RecvDat	ARRAY[0..9] OF BYTE	[10(16#0)]	수신 데이터
	SendCmd_instance	SendCmd		

내부 변수	명칭	데이터 형식	코멘트
	DeviceNet_OnlineSta	BOOL	온라인
	_Port_isAvailable	BOOL	네트워크 통신 명령 실행 가능 플래그

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Port_isAvailable=TRUE)
     AND (DeviceNet_OnlineSta=TRUE) ) THEN
   OperatingStart  :=TRUE;
   Operating       :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 통신 파라메터 설정과 SendCmd 명령 초기화
IF (OperatingStart=TRUE) THEN
   SendCmd_instance(
      Execute     :=FALSE,
      DstNetAdr   :=InDNetAdr,
      CommPort    :=_NONE,
      CmdDat      :=SendDat[0],
      CmdSize     :=UINT#9,
      RespDat     :=RecvDat[0],
      Option      :=InOption);
   InDNetAdr.NetNo      :=USINT#0;          // 네트워크 주소 설정
   InDNetAdr.NodeNo     :=USINT#0;
   InDNetAdr.UnitNo     :=BYTE#16#10;
   InOption.isNonResp   :=FALSE;            // 응답 설정
   InOption.TimeOut     :=UINT#20;
   InOption.Retry       :=USINT#2;
   SendDat[0]           :=BYTE#16#28;       // 명령 배열 설정
   SendDat[1]           :=BYTE#16#01;
   SendDat[2]           :=BYTE#16#0B;
   SendDat[3]           :=BYTE#16#0E;
   SendDat[4]           :=BYTE#16#00;
   SendDat[5]           :=BYTE#16#01;
   SendDat[6]           :=BYTE#16#00;
   SendDat[7]           :=BYTE#16#01;
   SendDat[8]           :=BYTE#16#01;
   OperatingStart       :=FALSE;
END_IF;
```

```
// SendCmd 명령 실행
IF (Operating=TRUE) THEN
   SendCmd_instance(
      Execute     :=TRUE,
      DstNetAdr   :=InDNetAdr,
      CommPort    :=_NONE,
      CmdDat      :=SendDat[0],
      CmdSize     :=UINT#9,
      RespDat     :=RecvDat[0],
      Option      :=InOption);

   IF (SendCmd_instance.Done=TRUE) THEN
      // 정상 종료 시 처리
      Operating:=FALSE;
   END_IF;

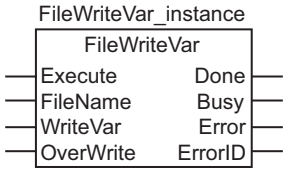
   IF (SendCmd_instance.Error=TRUE) THEN
      // 이상 종료 시 처리
      Operating:=FALSE;
   END_IF;
END_IF;
```

SD 메모리 카드 명령

명령	명칭	페이지
FileWriteVar	변수 파일 쓰기	2-1040
FileReadVar	변수 파일 읽기	2-1046
FileOpen	파일 열기	2-1051
FileClose	파일 닫기	2-1055
FileSeek	파일 검색	2-1058
FileRead	파일 읽기	2-1061
FileWrite	파일 쓰기	2-1069
FileGets	문자열 읽기	2-1077
FilePuts	문자열 쓰기	2-1084
FileCopy	파일 복사	2-1092
FileRemove	파일 삭제	2-1100
FileRename	파일명 변경	2-1105
DirCreate	디렉토리 작성	2-1110
DirRemove	디렉토리 삭제	2-1113

FileWriteVar

SD 메모리 카드 내의 지정된 파일에 1개의 변수의 값을 바이너리 형식으로 써넣습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileWriteVar	변수 파일 쓰기	FB		FileWriteVar_instance(Execute, FileName, WriteVar, OverWrite, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	지정 파일명	입력	쓰는 파일명	최대 66바이트 (반각 영숫자 65 문자+끝의 NULL 문자)	–	''
WriteVar	지정 변수		쓰는 변수	데이터 형식에 따름		*1
OverWrite	덮어쓰기 허가		TRUE : 덮어쓰기 허가 FALSE : 덮어쓰기 방지			FALSE

*1 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○
WriteVar	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			
OverWrite	○																			

기능

SD 메모리 카드 내의 'FileName'으로 지정한 파일에, 1개의 변수 'WriteVar'의 값을 바이너리 형식으로 써넣습니다.
'WriteVar'에는 열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버를 지정할 수도 있습니다.

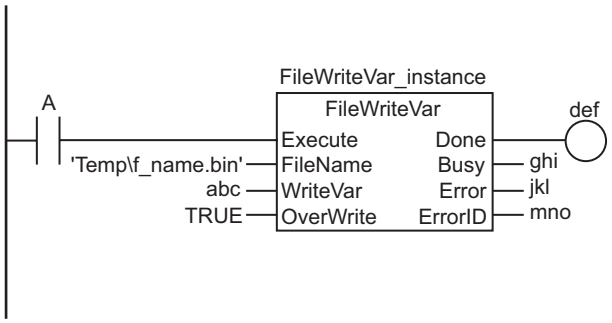
SD 메모리 카드 내에 'FileName'과 같은 이름의 파일이 존재하지 않는 경우에는 신규로 파일을 작성합니다.
'FileName'이 디렉토리를 포함해 기술되어 있으며 SD 메모리 카드 내에 그 디렉토리도 존재하지 않는 경우, 디렉토리별로 신규 작성합니다. 단, 디렉토리별로 신규 작성하는 것은, 지정된 디렉토리의 최하층만이 존재하지 않는 경우입니다.

SD 메모리 카드 내에 'FileName'과 같은 이름의 파일이 이미 존재하는 경우, 덮어쓰기 허가 'OverWrite'의 값에 따라 다음과 같은 처리를 수행합니다.

'OverWrite'의 값	처리
TRUE(덮어쓰기 허가)	그 파일에 덮어씁니다.
FALSE(덮어쓰기 방지)	그 파일에 덮어쓰지 못하며, 이상 상태가 됩니다.

기술 예가 아래에 나와 있습니다. 'Temp\f_name.bin'이라는 이름의 파일에 배열 변수 abc의 전체를 씁니다. abc는 요소 수 3인 INT 형식 배열 변수라고 가정합니다.

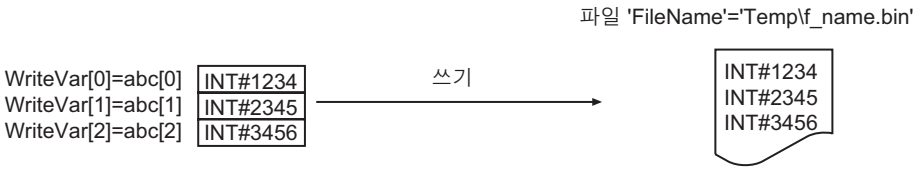
LD



ST

```
FileWriteVar_instance(A, 'Temp\f_name.bin', abc,
                      TRUE, def, ghi, jkl, mno);
```

SD 메모리 카드 내의 'FileName'으로 지정한 파일에 변수 'WriteVar'의 값을 바이너리 형식으로 기입합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타낸 플래그입니다. 이 플래그는 자동적으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'WriteVar'에 전달하는 입력 파라메터는 반드시 변수로 하십시오. 정수를 넘겨주면 빌드 시에 이상 상태가 됩니다.
- 'WriteVar'이 열거 형식인 경우, 열거자를 직접 전달할 수는 없습니다. 열거자를 직접 전달하면 빌드 시에 이상 상태가 됩니다.
- 지정 파일의 크기가 'WriteVar'의 크기보다 큰 경우, 이상은 발생하지 않고 'WriteVar'의 크기만큼 데이터를 씁니다. 그리고 이 명령 실행 후, 지정 파일은 'WriteVar'의 크기에 맞게 작아집니다.
- 쓰인 데이터는 바이트 단위로, 하위 바이트→상위 바이트 순서(리틀 엔디언)로 나열합니다.
- 'WriteVar'이 구조체 전체인 경우, 그 구성에 따라 각 멤버 사이에 조정용 영역이 삽입되는 경우가 있습니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - SD 메모리 카드가 여유 용량 부족인 경우
  - 'FileName'의 값이 파일명으로서 잘못된 경우
  - 작성 가능한 파일 수, 디렉토리 수를 초과한 경우
  - 'FileName'과 같은 이름의 파일이 이미 존재하며, 액세스 중인 경우
  - 'FileName'과 같은 이름의 파일이 이미 존재하며, 'OverWrite'의 값이 FALSE인 경우
  - 'FileName'과 같은 이름의 파일이 이미 존재하며, 그 파일이 쓰기 금지인 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'FileName'의 값이 파일명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

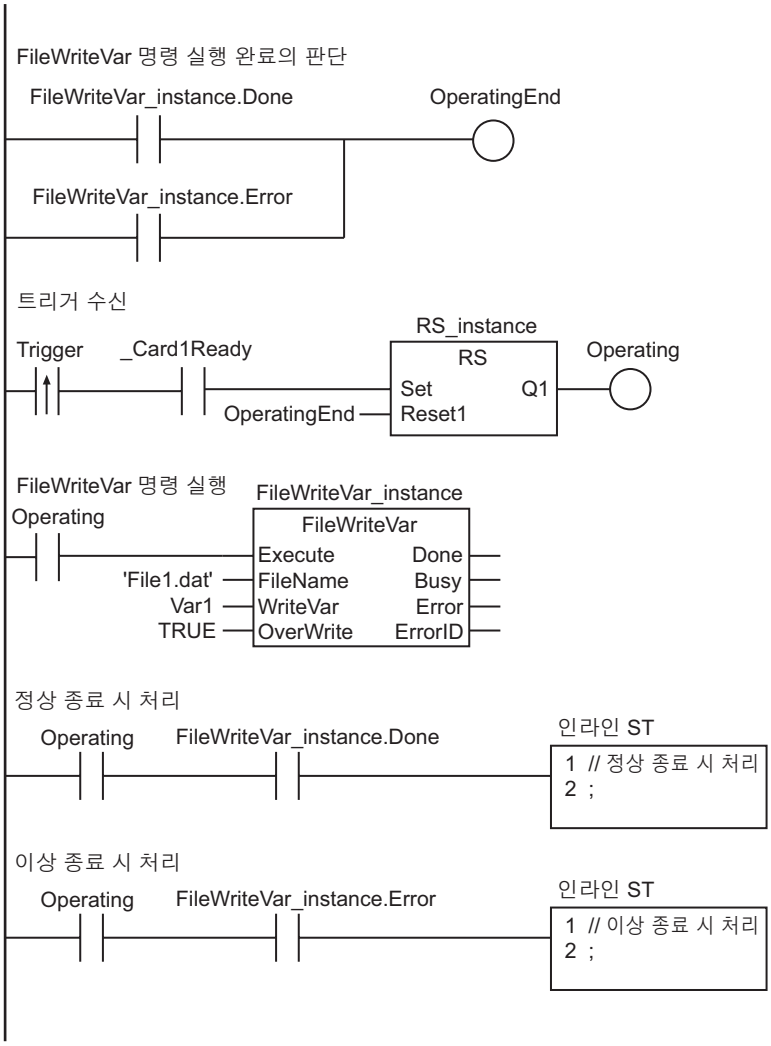
샘플 프로그램

배열 변수 Var1[]의 전체를 파일 'File1.dat'에 씁니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Var1	ARRAY[0..999] OF INT	[1000(0)]	쓰기 데이터
	RS_instance	RS		
	FileWriteVar_instance	FileWriteVar		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 테스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	Var1	ARRAY[0..999] OF INT	[1000(0)]	지정 변수
	FileWriteVar_instance	FileWriteVar		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// FileWriteVar 명령 초기화
IF (OperatingStart=TRUE) THEN
    FileWriteVar_instance(
        Execute:=FALSE,
        WriteVar:=Var1);
    OperatingStart :=FALSE;
END_IF;

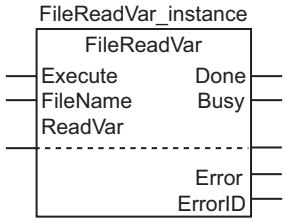
// FileWriteVar 명령 실행
IF (Operating=TRUE) THEN
    FileWriteVar_instance(
        Execute   :=TRUE,
        FileName :='File1.dat',   // 지정 파일명
        WriteVar  :=Var1,         // 지정 변수
        OverWrite:=TRUE);         // 덮어쓰기 허가

    IF (FileWriteVar_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating:=FALSE;
    END_IF;

    IF (FileWriteVar_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating:=FALSE;
    END_IF;
END_IF;
```

FileReadVar

SD 메모리 카드 내의 지정된 파일의 값을 바이너리 형식으로 읽어 변수에 써넣습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileReadVar	변수 파일 읽기	FB		FileReadVar_instance(Execute, FileName, ReadVar, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	지정 파일명	입력	읽는 파일명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''
ReadVar	읽기대상 변수	입출력	읽은 값의 저장대상 변수	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○
ReadVar	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버도 지정 가능																			

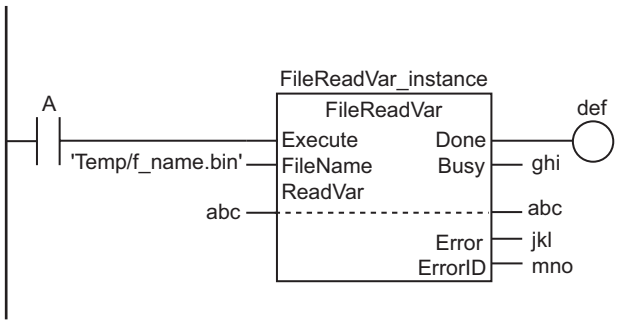
기능

SD 메모리 카드 내의 'FileName'으로 지정한 파일 내의 값을 바이너리 형식으로 읽습니다. 읽은 값은 읽기대상 변수 'ReadVar'에 대입됩니다.

'ReadVar'에는 열거 형식, 배열 전체, 배열의 1개 요소, 구조체 전체, 구조체의 1개 멤버를 지정할 수도 있습니다.

기술 예가 아래에 나와 있습니다. 'Temp/f_name.bin'이라는 이름의 파일의 내용을 읽어서 배열 변수 abc[]에 씁니다. abc는 요소 수 3인 INT 형식 배열 변수라고 가정합니다.

LD



ST

```
FileReadVar_instance(A, 'Temp/f_name.bin', abc,
    def, ghi, jkl, mno);
```

SD 메모리 카드 내의 'FileName'으로 지정한 파일 내의 값을 바이너리 형식으로 읽어 변수 'ReadVar'에 대입합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 지정 파일의 크기가 'ReadVar'의 크기보다 큰 경우, 이상은 발생하지 않고 'ReadVar'의 크기만큼 데이터를 읽습니다.
- 지정 파일의 크기가 'ReadVar'의 크기보다 작은 경우, 이상은 발생하지 않고 지정 파일의 크기만큼 데이터를 읽습니다. 그리고, 'ReadVar'의 남은 영역은 이 명령 실행 전의 값이 유지됩니다.
- 읽는 데이터는 바이트 단위이고, 하위 바이트→상위 바이트 순서(리틀 엔디언)로 나열합니다.
- 'ReadVar'이 구조체 전체인 경우, 그 구성에 따라 각 멤버 사이에 조정용 영역이 삽입되는 경우가 있습니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - 'FileName'으로 지정한 파일이 존재하지 않는 경우
  - 'FileName'의 값이 파일명으로서 잘못된 경우
  - 'FileName'으로 지정한 파일이 액세스 중인 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'FileName'의 값이 파일명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

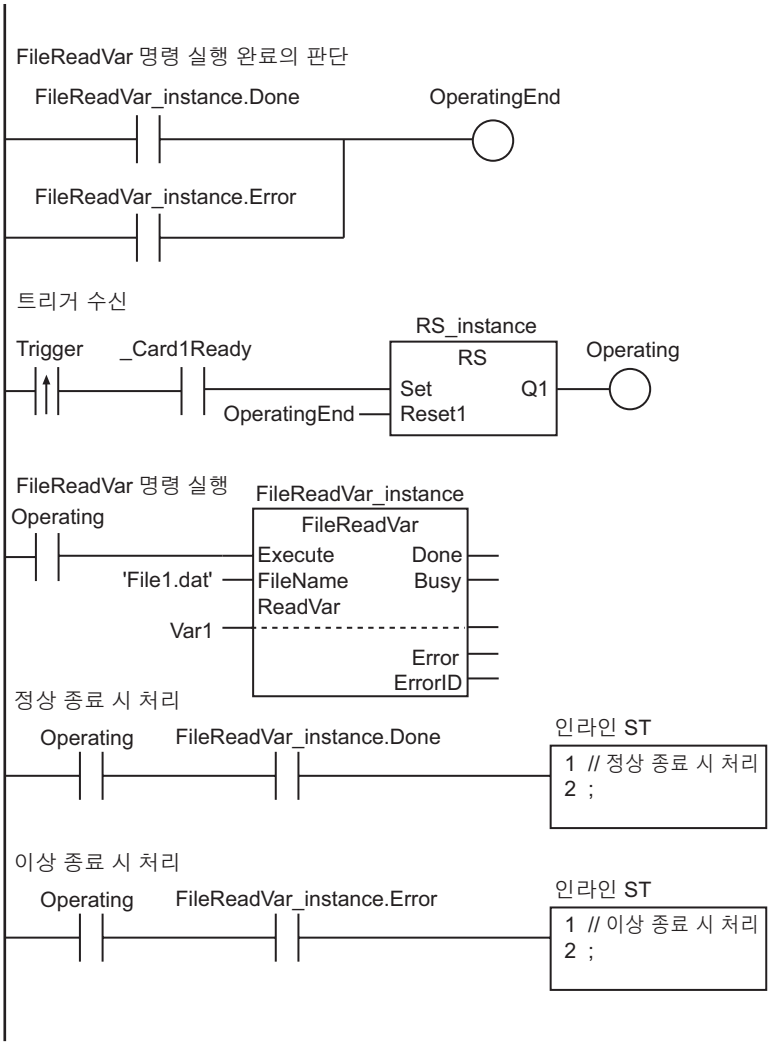
샘플 프로그램

파일 'File1.dat'의 내용을 읽어서 배열 변수 Var1에 저장합니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Var1	ARRAY[0..999] OF INT	[1000(0)]	읽기 데이터
	RS_instance	RS		
	FileReadVar_instance	FileReadVar		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	Var1	ARRAY[0..999] OF INT	[1000(0)]	읽기대상 변수
	FileReadVar_instance	FileReadVar		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동 검출
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// FileReadVar 명령 초기화
IF (OperatingStart=TRUE) THEN
    FileReadVar_instance(
        Execute :=FALSE,
        ReadVar:=Var1);
    OperatingStart :=FALSE;
END_IF;

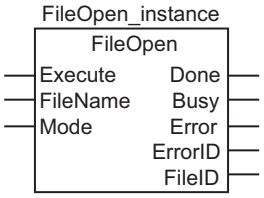
// FileReadVar 명령 실행
IF (Operating=TRUE) THEN
    FileReadVar_instance(
        Execute   :=TRUE,
        FileName :='File1.dat',    // 지정 파일명
        ReadVar   :=Var1);         // 읽기대상 변수

    IF (FileReadVar_instance.Done=TRUE) THEN
        // 정상 종료 시 처리
        Operating:=FALSE;
    END_IF;

    IF (FileReadVar_instance.Error=TRUE) THEN
        // 이상 종료 시 처리
        Operating:=FALSE;
    END_IF;
END_IF;
```

FileOpen

SD 메모리 카드 내의 지정된 파일을 엽니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
FileOpen	파일 열기	FB		FileOpen_instance(Execute, FileName, Mode, Done, Busy, Error, ErrorID, FileID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	지정 파일명	입력	여는 파일명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''
Mode	열기 모드		파일을 여는 모드	(*)		_READ_EXIST
FileID	파일 ID	출력	연 파일의 파일 ID	데이터 형식에 따름	–	–

* _READ_EXIST, _RDWR_EXIST, _WRITE_CREATE, _RDWR_CREATE, _WRITE_APPEND, _RDWR_APPEND

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○
Mode	열거 형식 _eFOPEN_MODE 열거자는 기능 설명 참조																			
FileID				○																

기능

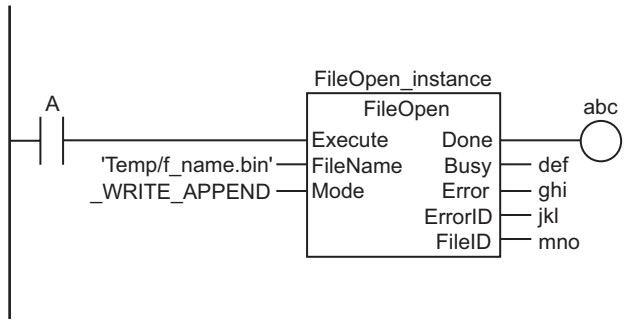
SD 메모리 카드 내의 'FileName'으로 지정한 파일을 'mode'로 지정한 모드에서 엽니다.
파일을 연 결과, 파일 ID 'FileID'가 출력됩니다. 'FileID'는 FileRead 명령, FileWrite 명령 등으로 파일을 지정할 때 사용합니다.

'Mode'의 데이터 형식은 열거 형식 _eFOPEN_MODE입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_READ_EXIST	텍스트 파일을 읽기 위해 엽니다. 읽기는 파일의 앞에서부터 시작합니다.
_RDWR_EXIST	읽기 및 쓰기용으로 파일을 엽니다. 읽기 및 쓰기는 파일의 앞에서부터 시작합니다.
_WRITE_CREATE	쓰기용으로 파일을 엽니다. 파일이 이미 존재하는 경우, 그 내용을 폐기하고 파일 크기를 0으로 합니다. 존재하지 않는 경우에는 파일을 신규 작성합니다. 쓰기는 파일의 앞에서부터 시작합니다. 단, 파일이 이미 존재하며 그 파일에 쓰기 금지가 설정되어 있는 경우에는 이상 상태가 되어 파일을 열 수 없습니다.
_RDWR_CREATE	읽기 및 쓰기용으로 파일을 엽니다. 파일이 이미 존재하는 경우, 그 내용을 폐기하고 파일 크기를 0으로 합니다. 존재하지 않는 경우에는 파일을 신규 작성합니다. 읽기 및 쓰기는 파일의 앞에서부터 시작합니다.
_WRITE_APPEND	추가 쓰기용으로 파일을 엽니다. 파일이 존재하지 않는 경우에는 파일을 신규 작성합니다. 추가 쓰기는 파일의 끝에서부터 시작합니다. 단, 파일이 이미 존재하며 그 파일에 쓰기 금지가 설정되어 있는 경우에는 이상 상태가 되어 파일을 열 수 없습니다.
_RDWR_APPEND	읽기 및 추가 쓰기용으로 파일을 엽니다. 파일이 존재하지 않는 경우에는 파일을 신규 작성합니다. 읽기는 파일의 앞에서부터 시작합니다. 추가 쓰기는 파일의 끝에서부터 시작합니다.

기술 예가 아래에 나와 있습니다. "Temp/f_name.bin'이라는 이름의 파일을 추가 쓰기용으로 엽니다. 변수 mno에 파일 ID가 대입됩니다.

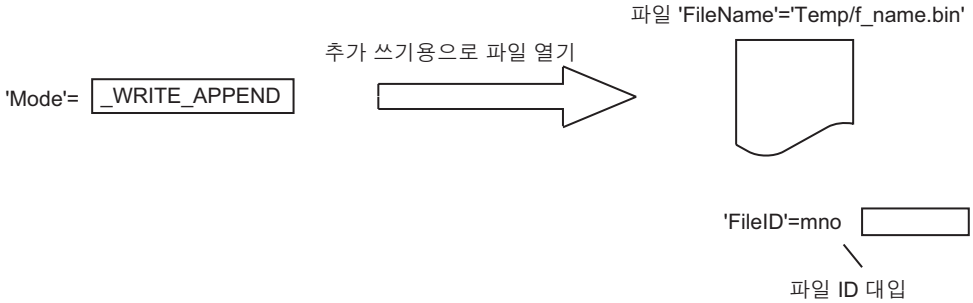
LD



ST

```
FileOpen_instance(A, 'Temp/f_name.bin',
                  _WRITE_APPEND, abc,
                  def, ghi, jkl, mno);
```

SD 메모리 카드 내의 'FileName'으로 지정한 파일을 추가 쓰기용으로 엽니다.
변수 'FileID'에 파일 ID가 대입됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령은 FileSeek 명령, FileRead 명령, FileWrite 명령, FileGets 명령, FilePuts 명령 전에 실행해두어야 합니다.
- 이 명령으로 파일 열기를 한 파일은, 사용 후에 반드시 File Close 명령을 실행해서 파일을 닫으십시오.
- 'FileID'에 값이 저장되는 것은 이 명령이 완료된 시점입니다. 즉, 'Done'의 값이 FALSE에서 TRUE로 변화한 시점입니다.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.

- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다. 하지만 파일이 열린 상태는 계속되므로 FileClose 명령을 실행해서 파일을 닫으십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우라도, 파일이 열려 있는 상태는 계속됩니다. FileClose 명령을 실행해서 파일을 닫으십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 파일이 열린 상태는 계속되지만, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다. 파일 읽기/쓰기를 수행하기 위해서는, 일단 파일을 닫은 후 다시 파일을 여십시오.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - 'Mode'의 값이 _READ_EXIST 또는 _RDWR_EXIST이며 'FileName'으로 지정한 파일이 존재하지 않는 경우
  - 'FileName'의 값이 파일명으로서 잘못된 경우
  - 작성 가능한 파일 수, 디렉토리 수를 초과한 경우
  - 'FileName'으로 지정한 파일이 액세스 중인 경우
  - 'FileName'으로 지정한 파일이 쓰기 금지 상태인 경우
  - 5개 이상의 파일을 동시에 열려고 한 경우
  - 'FileName'의 값이 파일명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우
  - 'Mode'의 값이 유효 범위 밖인 경우

샘플 프로그램

🕮 'FileRead 명령(2-1061페이지)', 🕮 'FileWrite 명령(2-1069페이지)', 🕮 'FileGets 명령(2-1077페이지)', 🕮 'FilePuts 명령(2-1084페이지)'의 샘플 프로그램을 참조하십시오.

FileClose

SD 메모리 카드 내의 지정된 파일을 닫습니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
FileClose	파일 닫기	FB	FileClose_instance FileClose Execute Done FileID Busy Error ErrorID	FileClose_instance(Execute, FileID, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	닫는 파일의 파일 ID	데이터 형식에 따름	–	0

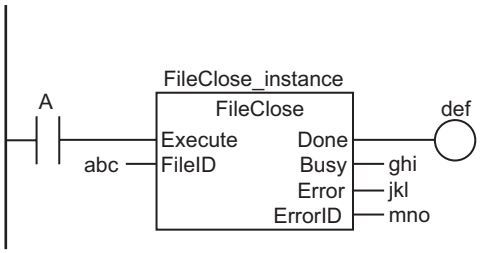
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																

기능

SD 메모리 카드 내의 'FileID'로 지정한 파일을 닫습니다.

기술 예가 아래에 나와 있습니다. 변수 abc의 값을 파일 ID로 하는 파일을 닫습니다.

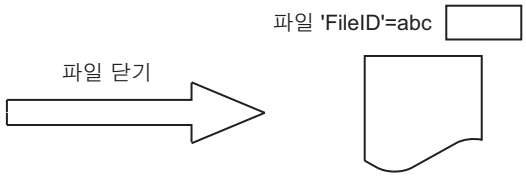
LD



ST

```
FileClose_instance(A, abc, def, ghi, jkl, mno);
```

SD 메모리 카드 내의 'FileID'로 지정한 파일을 닫습니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

FileOpen 명령으로 파일을 열어두어야 하는 명령은 FileSeek 명령, FileRead 명령, FileWrite 명령, FileGets 명령, FilePuts 명령입니다.

사용상의 주의 사항

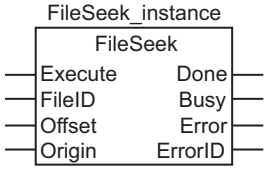
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'FileID'는 미리 FileOpen 명령을 실행해서 취득해두어야 합니다.
- FileOpen 명령으로 파일 열기를 한 파일은, 사용 후에 반드시 이 명령을 실행해서 파일을 닫으십시오.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다. 하지만 파일이 열린 상태는 계속되므로 FileClose 명령을 실행해서 파일을 닫으십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우라도, 파일이 열려 있는 상태는 계속됩니다. FileClose 명령을 실행해서 파일을 닫으십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 파일이 열린 상태는 계속되지만, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다. 파일 읽기/쓰기를 수행하기 위해서는, 일단 파일을 닫은 후 다시 파일을 여십시오.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 이미 닫혀 있는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우

샘플 프로그램

'FileRead 명령(2-1061페이지)', 'FileWrite 명령(2-1069페이지)', 'FileGets 명령(2-1077페이지)', 'FilePuts 명령(2-1084페이지)'의 샘플 프로그램을 참조하십시오.

FileSeek

SD 메모리 카드 내의 지정된 파일에 파일 위치 지시자를 설정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileSeek	파일 검색	FB		FileSeek_instance(Execute, FileID, Offset, Origin, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	파일 위치 지시자를 설정하는 파일의 파일 ID	데이터 형식에 따름	–	0
Offset	오프셋		'Origin'으로부터의 오프셋 위치		바이트	
Origin	기준 위치		파일 위치 지시자의 기준 위치	_SEEK_SET, _SEEK_CUR, _SEEK_END	–	_SEEK_SET

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																
Offset												○								
Origin	열거 형식 _eFSEEK_ORIGIN 열거자는 기능 설명 참조																			

기능

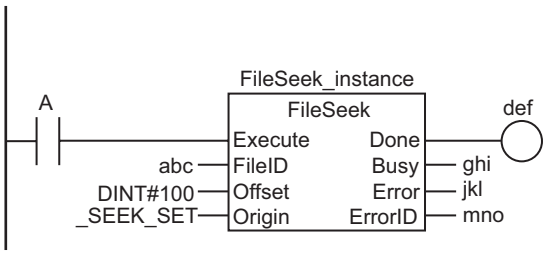
SD 메모리 카드 내의 파일 ID 'fileID'로 지정한 파일에, 파일 위치 지시자를 설정합니다. 파일 위치 지시자란, FileRead 명령 및 FileWrite 명령 등을 실행했을 때 읽기 및 쓰기를 시작하는 파일 내의 위치를 말합니다. 예를 들면, 파일의 앞에서부터 읽기를 수행하고자 하는 경우는 FileSeek 명령으로 파일 위치 지시자를 파일의 앞에 설정한 후 FileRead 명령을 실행합니다.
파일 위치 지시자가 설정되는 것은 기준 위치 'Origin'에 오프셋 'Offset'을 더한 위치입니다.

'Origin'의 데이터 형식은 열거 형식 _eFSEEK_ORIGIN입니다. 열거자의 의미는 다음과 같습니다.

열거자	의미
_SEEK_SET	파일의 앞
_SEEK_CUR	현재의 파일 위치 지시자의 위치
_SEEK_END	파일의 끝

기술 예가 아래에 나와 있습니다. 파일의 앞에서부터 100바이트의 위치에 파일 위치 지시자를 설정합니다.

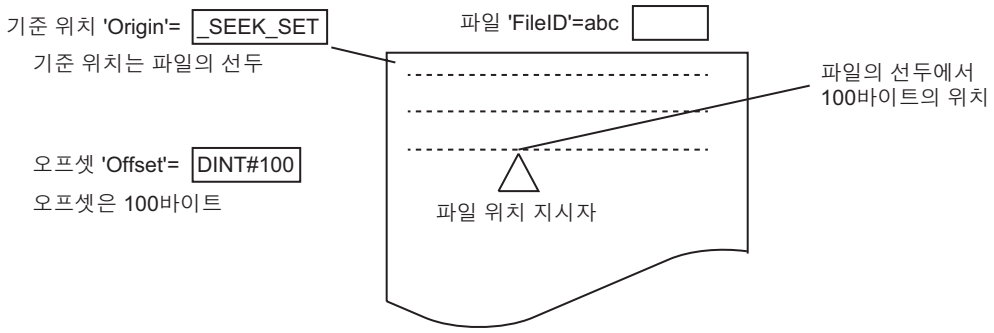
LD



ST

```
FileSeek_instance(A, abc, DINT#100,
                  _SEEK_SET, def,
                  ghi, jkl, mno);
```

SD 메모리 카드 내의 'FileID'로 지정한 파일에 파일 위치 지시자를 설정합니다.
파일 위치 지시자의 위치는 파일 선두에서 'Offset'을 더한 위치입니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

사용상의 주의 사항

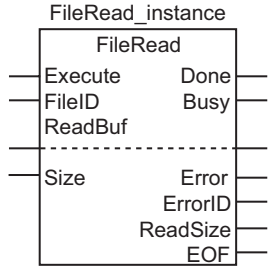
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 FileOpen 명령을 실행해서 'FileID'를 취득해두어야 합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - 'Origin'의 값이 유효 범위 밖인 경우
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - 'Origin'과 'Offset'으로 지정한 위치가 파일 크기를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

'FileRead 명령(2-1061페이지)', 'FileWrite 명령(2-1069페이지)'의 샘플 프로그램을 참조하십시오.

FileRead

SD 메모리 카드 내의 지정된 파일의 데이터를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileRead	파일 읽기	FB		FileRead_instance(Execute, FileID, ReadBuf, Size, Done, Busy, Error, ErrorID, ReadSize, EOF);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	읽는 파일의 파일 ID	데이터 형식에 따름	–	0
Size	읽기 요소 수		읽는 요소 수			1
ReadBuf[] 배열	읽기 버퍼	입출력	읽은 데이터의 쓰기대상	데이터 형식에 따름	–	–
ReadSize	실제로 읽은 요소 수	출력	실제로 읽은 요소 수	데이터 형식에 따름	–	–
EOF	파일 끝		파일 끝에 도달했는지 판정 TRUE : 도달함 FALSE : 도달하지 않음			

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																
Size							○													
ReadBuf[]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
배열	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
ReadSize							○													
EOF	○																			

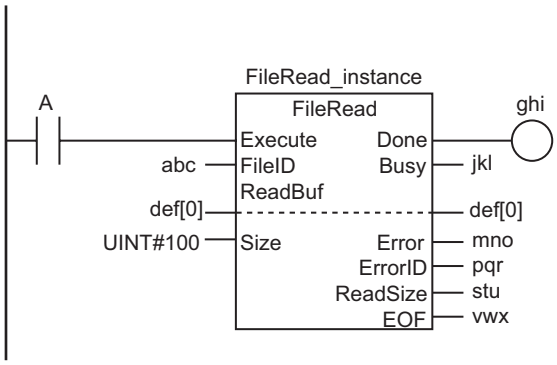
기능

SD 메모리 카드 내의 파일 ID 'FileID'로 지정한 파일의 파일 위치 지시자가 있는 위치로부터 데이터를 읽어서, 읽기 버퍼 ReadBuf[]에 저장합니다.
파일 위치 지시자는 미리 FileSeek 명령으로 임의의 위치에 설정해둡니다.
읽는 데이터의 양은 (ReadBuf[]의 데이터 형식의 크기)×'Size'입니다. 즉, ReadBuf[]의 'Size' 개의 요소만큼입니다.
ReadBuf[]는 열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열이라도 상관없습니다.

실제로 읽은 요소 수가 'ReadSize'에 저장됩니다. 일반적으로는 'Size'의 값과 'ReadSize'의 값이 일치합니다. 만일 파일 위치 지시자가 있는 위치에서부터 파일 끝까지의 데이터 양이 'Size'보다 작은 경우, 이상은 발생하지 않고 파일 끝까지의 데이터가 ReadBuf[]에 저장됩니다. 이때, 'ReadSize'의 값은 'Size'의 값보다 작아집니다.
또한, 파일 끝까지 읽은 경우, 파일 끝 'EOF'의 값이 TRUE가 됩니다. 그렇지 않은 경우, 'EOF'의 값은 FALSE가 됩니다.

기술 예가 아래에 나와 있습니다. 읽기 버퍼 def[]가 BYTE 형식 배열이라고 하면, 파일에서 100바이트의 데이터를 읽습니다.

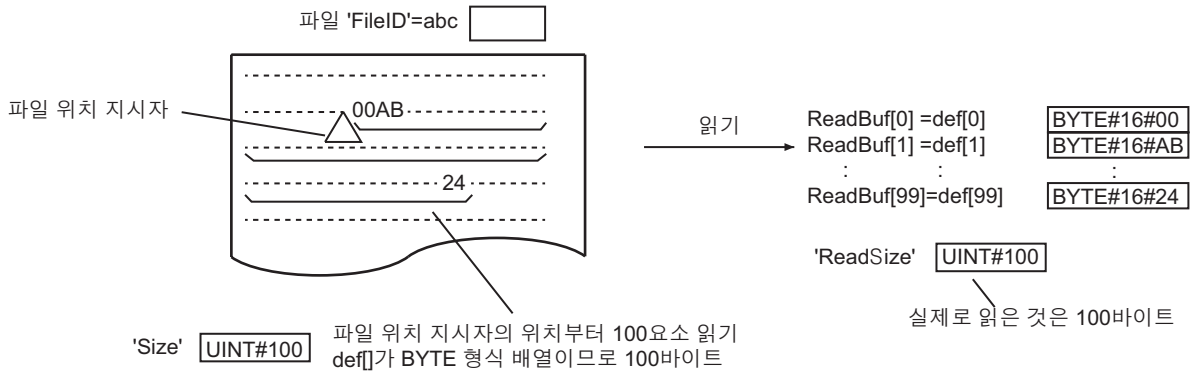
LD



ST

```
FileRead_instance(A, abc, def[0], UINT#100,
                  ghi, jkl, mno, pqr, stu, vwx);
```

SD 메모리 카드 내의 'FileID'로 지정한 파일에서 파일 위치 지시자가 있는 위치로부터 'Size' 요소 분량만 읽어서 읽기 버퍼 ReadBuf[]에 저장합니다.
그 후 실제로 읽은 데이터 크기가 'ReadSize'에 출력됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 파일의 끝까지의 데이터를 읽어서, 그 크기가 ReadBuf[]의 데이터 형식의 크기로 나뉘어 떨어지지 않는 경우, ReadBuf[]의 데이터 크기에 채워지지 않는 데이터는 폐기됩니다. 단, 파일 위치 지시자는 파일 끝까지 진행하며 'EOF'의 값은 TRUE가 됩니다.
- ReadBuf[]의 'Size' 번째 이후의 요소(읽기로 덮어쓰기가 되지 않는 요소)는 이 명령 실행 전의 값이 유지됩니다.
- 이 명령을 실행하기 전에 FileOpen 명령을 실행해서 'FileID'를 취득해두어야 합니다.
- 읽는 데이터는 바이트 단위로, 하위 바이트→상위 바이트 순서(리틀 엔디언)로 나열합니다.
- 'EOF'에 값이 저장되는 것은 이 명령이 완료된 시점입니다. 즉, 'Done'의 값이 FALSE에서 TRUE로 변화한 시점입니다.
- ReadBuf[]가 구조체를 요소로 하는 배열인 경우, 그 구성에 따라 각 멤버 사이에 조정용 영역이 삽입되는 경우가 있습니다.
- 이 명령 실행 중에 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - ReadBuf[]의 배열 요소 수가 'Size'의 값보다 작은 경우
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - 'FileID'로 지정한 파일이 읽기 가능한 모드로 열려 있지 않은 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

파일 'ABC.bin'의 앞에서부터 2바이트째의 위치에서 4바이트 분량의 데이터를 읽어서 BYTE 형식 배열 변수 InDat[]에 저장합니다.
처리 순서는 다음과 같습니다.

1 **FileOpen 명령을 사용해서 파일 'ABC.bin'을 엽니다.**

2 **FileSeek 명령을 사용해서, 파일 위치 지시자를 파일의 앞에서부터 2바이트째에 설정합니다.**

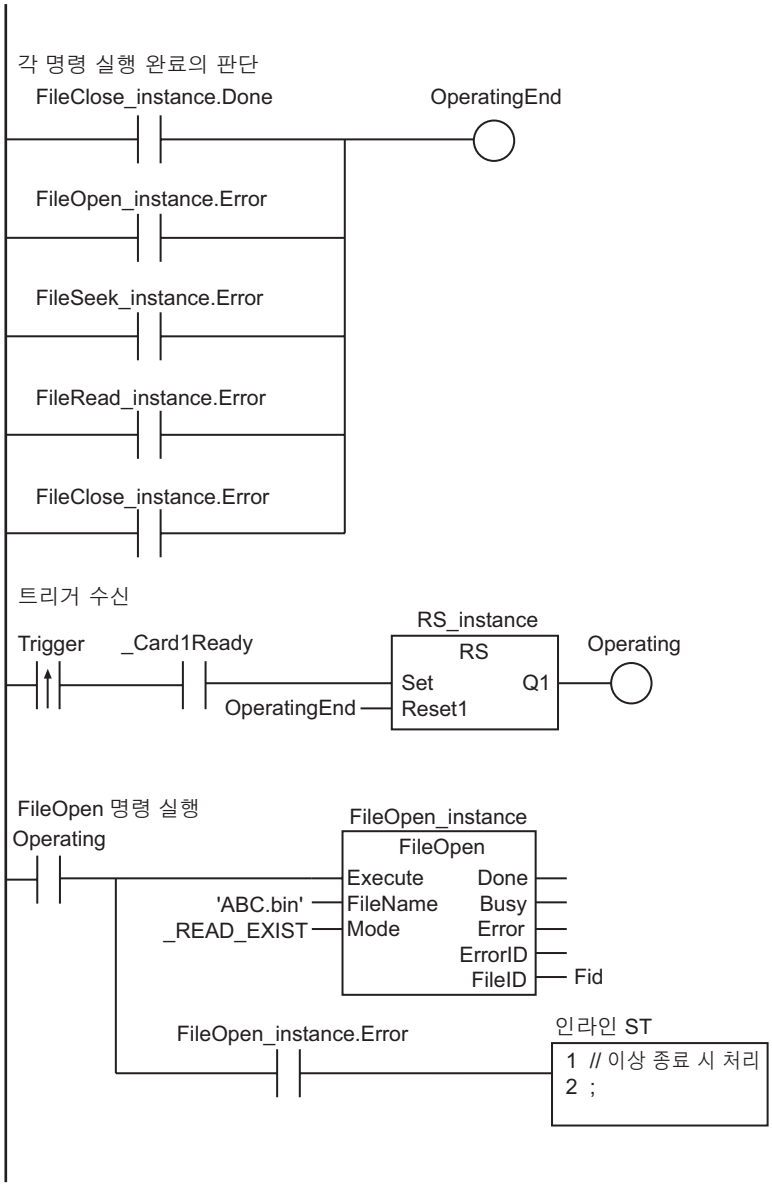
3 **FileRead 명령을 사용해서, 파일 위치 지시자의 위치로부터 4바이트 분량의 데이터를 읽어서 배열 변수 InDat[]에 저장합니다.**

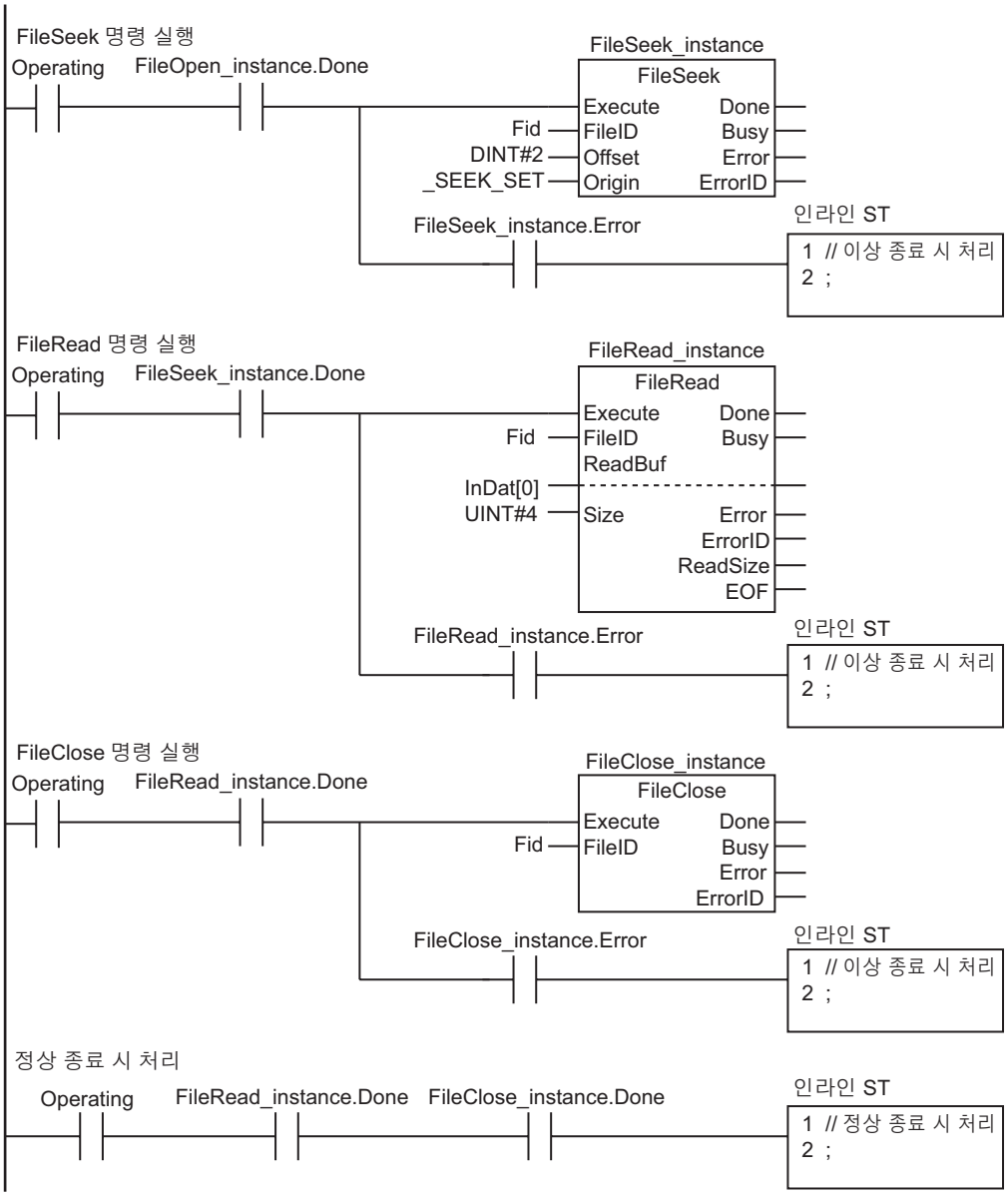
4 **FileClose 명령을 사용해서 'ABC.bin'을 닫습니다.**

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Fid	DWORD	16#0	파일 ID
	InDat	ARRAY[0..999] OF BYTE	[1000(16#0)]	읽기 데이터
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FileSeek_instance	FileSeek		
	FileRead_instance	FileRead		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




FileSeek 명령 실행
Operating
FileOpen_instance.Done
FileSeek_instance
FileSeek
Execute
Done
Fid
FileID
Busy
DINT#2
Offset
Error
_SEEK_SET
Origin
ErrorID
FileSeek_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
FileRead 명령 실행
Operating
FileSeek_instance.Done
FileRead_instance
FileRead
Execute
Done
Fid
FileID
Busy
ReadBuf
InDat[0]
UINT#4
Size
Error
ErrorID
ReadSize
EOF
FileRead_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
FileClose 명령 실행
Operating
FileRead_instance.Done
FileClose_instance
FileClose
Execute
Done
Fid
FileID
Busy
Error
ErrorID
FileClose_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
정상 종료 시 처리
Operating
FileRead_instance.Done
FileClose_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	InDat	ARRAY[0..999] OF BYTE	[1000(16#0)]	읽기 데이터
	Stage	INT	0	상태 이동
	Fid	DWORD	16#0	파일 ID
	FileOpen_instance	FileOpen		
	FileSeek_instance	FileSeek		
	FileRead_instance	FileRead		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileOpen_instance(Execute:=FALSE);     // 인스턴스 초기화
    FileSeek_instance(Execute:=FALSE);     // 인스턴스 초기화
    FileRead_instance(
        Execute:=FALSE,                    // 인스턴스 초기화
        ReadBuf:=InDat[0]);                // 더미
    FileClose_instance(Execute:=FALSE);    // 인스턴스 초기화
    Stage          :=INT#1;
    OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
    CASE Stage OF
    1 :                          // 파일 열기
        FileOpen_instance(
            Execute  :=TRUE,
            FileName:='ABC.bin',           // 파일명
            Mode     :=_READ_EXIST,        // 파일 읽기
            FileID   =>Fid);               // 파일 ID

        IF (FileOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;                  // 정상 종료
        END_IF;

        IF (FileOpen_instance.Error=TRUE) THEN
            Stage:=INT#99;                 // 이상 종료
        END_IF;
```

```
    2 :                             // 파일 검색
        FileSeek_instance(
            Execute:=TRUE,
            FileID   :=Fid,                     // 파일 ID
            Offset   :=DINT#2,                  // 파일 위치 지시자. 선두에서 2바이트째
            Origin   :=_SEEK_SET);              //

        IF (FileSeek_instance.Done=TRUE) THEN
            Stage:=INT#3;                       // 정상 종료
        END_IF;

        IF (FileSeek_instance.Error=TRUE) THEN
            Stage:=INT#99;                      // 이상 종료
        END_IF;

    3 :                             // 파일 읽기
        FileRead_instance(
            Execute :=TRUE,
            FileID    :=Fid,                    // 파일 ID
            ReadBuf:=InDat[0],                  // 읽기 버퍼
            Size      :=UINT#4);                // 읽기 요소 수 4바이트

        IF (FileRead_instance.Done=TRUE) THEN
            Stage:=INT#4;               // 정상 종료
        END_IF;

        IF (FileRead_instance.Error=TRUE) THEN
            Stage:=INT#99;              // 이상 종료
        END_IF;

    4 :                             // 파일 닫기
        FileClose_instance(
            Execute:=TRUE,
            FileID   :=Fid);                    // 파일 ID

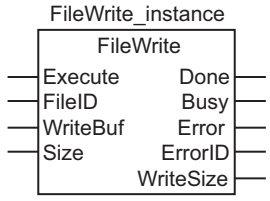
        IF (FileClose_instance.Done=TRUE) THEN
            Operating:=FALSE;           // 정상 종료
        END_IF;

        IF (FileClose_instance.Error=TRUE) THEN
            Stage:=INT#99;              // 이상 종료
        END_IF;

    99 :
        Operating:=FALSE;               // 이상 종료 처리
    END_CASE;
END_IF;
```

FileWrite

SD 메모리 카드 내의 지정된 파일에 데이터를 써넣습니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
FileWrite	파일 쓰기	FB		FileWrite_instance(Execute, FileID, WriteBuf, Size, Done, Busy, Error, ErrorID, WriteSize);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	쓰는 파일의 파일 ID	데이터 형식에 따름	–	0
WriteBuf[] 배열	쓰기 버퍼		쓰는 데이터			(*)
Size	쓰기 요소 수		쓰는 요소 수			1
WriteSize	실제로 쓴 요소 수	출력	실제로 쓴 요소 수	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

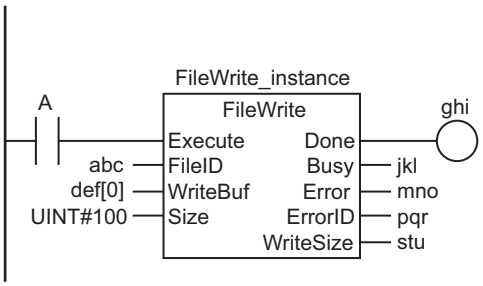
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																
WriteBuf[] 배열	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열도 지정 가능																			
Size							○													
WriteSize							○													

기능

SD 메모리 카드 내의 파일 ID 'FileID'로 지정한 파일의 파일 위치 지시자가 있는 위치에 데이터를 씁니다.
파일 위치 지시자는 미리 FileSeek 명령으로 임의의 위치에 설정해둡니다.
쓰는 데이터는 쓰기 버퍼 WriteBuf[]의 내용입니다.
쓰는 데이터의 양은 (WriteBuf[]의 데이터 형식의 크기)×'Size'입니다. 즉, WriteBuf[]의 'Size' 개의 요소 만큼입니다.
WriteBuf[]는 열거 형식을 요소로 하는 배열, 구조체를 요소로 하는 배열이라도 상관없습니다.
실제로 쓴 데이터 크기가 'WriteSize'로 출력됩니다.

기술 예가 아래에 나와 있습니다. 쓰기 버퍼 def[]가 BYTE 형식이라고 하면, 파일에 100바이트의 데이터를 씁니다.

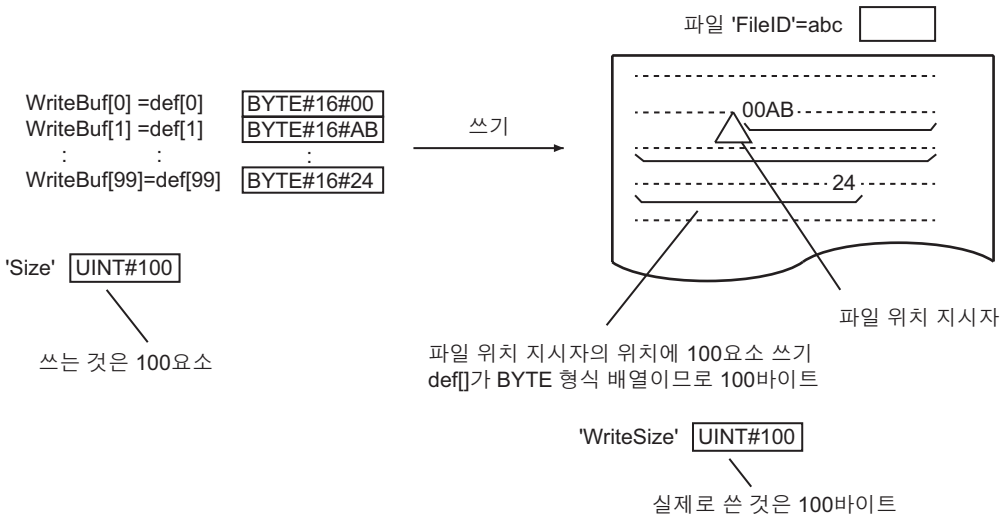
LD



ST

```
FileWrite_instance(A, abc, def[0], UINT#100, ghi,
                   jkl, mno, pqr, stu);
```

SD 메모리 카드 내의 'FileID'로 지정한 파일에서 파일 위치 지시자가 있는 위치에 쓰기 버퍼 WriteBuf[]의 내용을 'Size' 요소 분량만 씁니다.
그 후, 실제로 쓴 데이터 크기가 'WriteSize'로 출력됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 FileOpen 명령을 실행해서 'FileID'를 취득해두어야 합니다.
- 쓰이는 데이터는 바이트 단위로, 하위 바이트→상위 바이트 순서(리틀 엔디언)로 나열됩니다.
- WriteBuf[]가 구조체를 요소로 하는 배열인 경우, 그 구성에 따라 각 멤버 사이에 조정용 영역이 삽입되는 경우가 있습니다.
- 이 명령 실행 중에 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - WriteBuf[]의 배열 요소 수가 'Size'의 값보다 작은 경우
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - SD 메모리 카드가 여유 용량 부족인 경우
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - 'FileID'로 지정한 파일이 쓰기 가능한 모드로 열려 있지 않은 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

파일명 'ABC.bin'의, 앞에서부터 2바이트째의 위치에 4바이트 분량의 데이터를 씁니다. 쓰는 데이터는 BYTE 형식 배열 변수 OutDat[]의 내용입니다.
처리 순서는 다음과 같습니다.

1 **FileOpen 명령을 사용해서 파일 'ABC.bin'을 엽니다.**

2 **FileSeek 명령을 사용해서, 파일 위치 지시자를 파일의 앞에서부터 2바이트째에 설정합니다.**

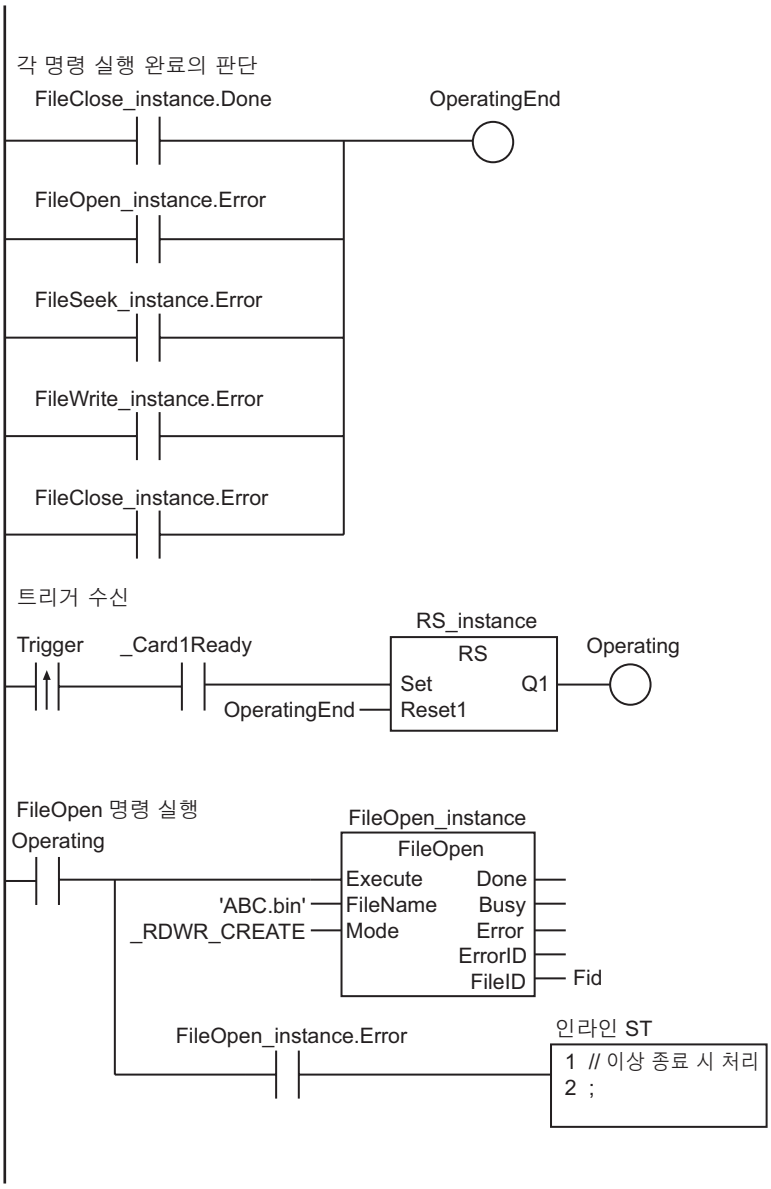
3 **FileWrite 명령을 사용해서 파일 위치 지시자의 위치에 배열 변수 OutDat[]의 값을 4바이트 분량 씁니다.**

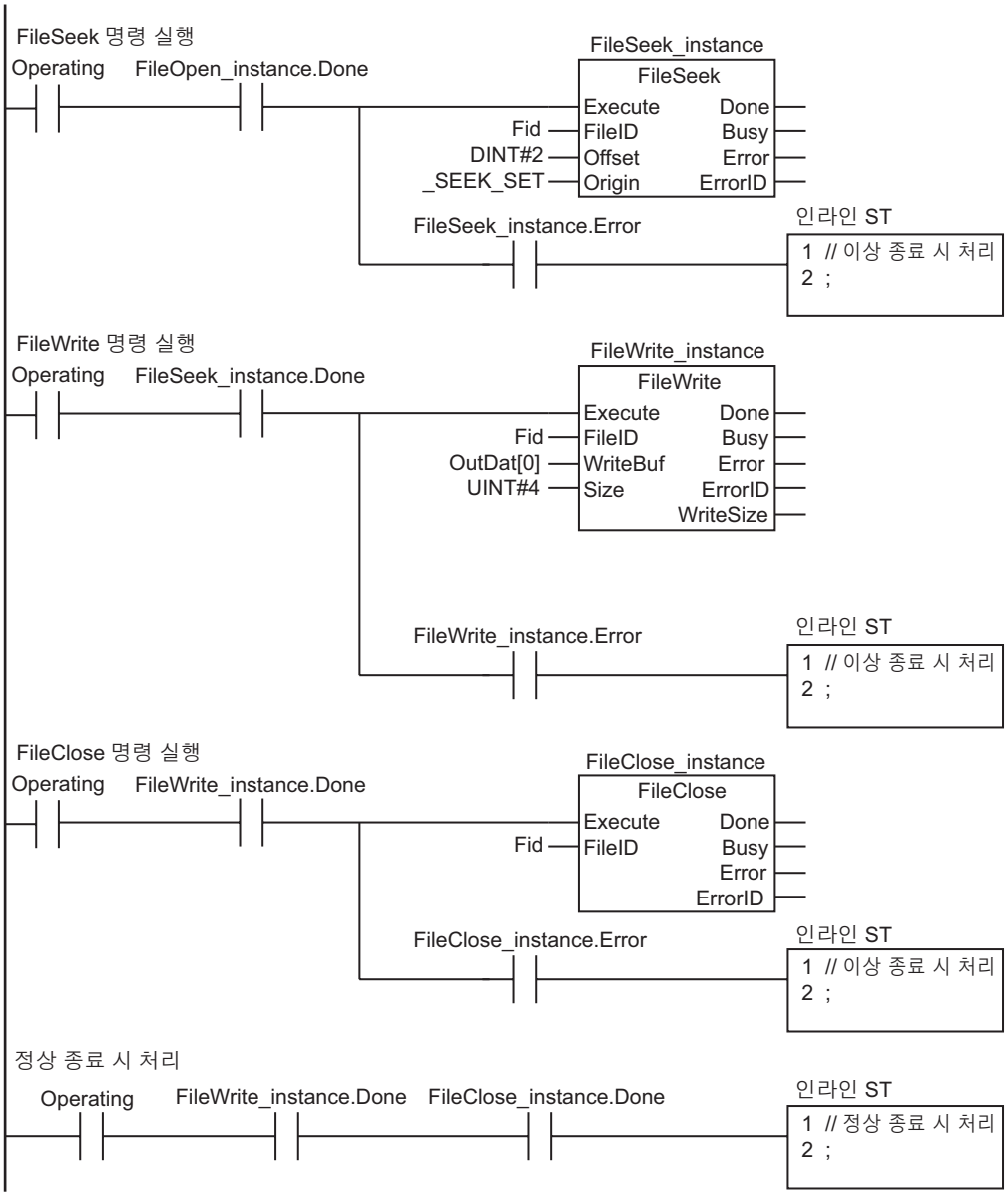
4 **FileClose 명령을 사용해서 'ABC.bin'을 닫습니다.**

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Fid	DWORD	16#0	파일 ID
	OutDat	ARRAY[0..999] OF BYTE	[1000(16#0)]	쓰기 데이터
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FileSeek_instance	FileSeek		
	FileWrite_instance	FileWrite		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




FileSeek 명령 실행
Operating
FileOpen_instance.Done
FileSeek_instance
FileSeek
Execute
Done
Fid
FileID
Busy
DINT#2
Offset
Error
_SEEK_SET
Origin
ErrorID
FileSeek_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
FileWrite 명령 실행
Operating
FileSeek_instance.Done
FileWrite_instance
FileWrite
Execute
Done
Fid
FileID
Busy
OutDat[0]
WriteBuf
Error
UINT#4
Size
ErrorID
WriteSize
FileWrite_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
FileClose 명령 실행
Operating
FileWrite_instance.Done
FileClose_instance
FileClose
Execute
Done
Fid
FileID
Busy
Error
ErrorID
FileClose_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
정상 종료 시 처리
Operating
FileWrite_instance.Done
FileClose_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	OutDat	ARRAY[0..999] OF BYTE	[1000(16#0)]	쓰기 데이터
	Stage	INT	0	상태 이동
	Fid	DWORD	16#0	파일 ID
	FileOpen_instance	FileOpen		
	FileSeek_instance	FileSeek		
	FileWrite_instance	FileWrite		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileOpen_instance(Execute:=FALSE);
    FileSeek_instance(Execute:=FALSE);
    FileWrite_instance(
        Execute :=FALSE,
        WriteBuf:=OutDat[0]);
    FileClose_instance(Execute:=FALSE);
    Stage          :=INT#1;
    OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
    CASE Stage OF
    1 :                          // 파일 열기
        FileOpen_instance(
            Execute  :=TRUE,
            FileName:='ABC.bin',              // 파일명
            Mode     :=_RDWR_CREATE,          // 파일 읽기 및 쓰기
            FileID   =>Fid);                  // 파일 ID

        IF (FileOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;                     // 정상 종료
        END_IF;

        IF (FileOpen_instance.Error=TRUE) THEN
            Stage:=INT#99;                    // 이상 종료
        END_IF;
```

```
    2 :                             // 파일 검색
        FileSeek_instance(
            Execute:=TRUE,
            FileID    :=Fid,                    // 파일 ID
            Offset    :=DINT#2,                 // 파일 위치 지시자. 선두에서 2바이트째
            Origin    :=_SEEK_SET);             //

        IF (FileSeek_instance.Done=TRUE) THEN
            Stage:=INT#3;
        END_IF;                                 // 정상 종료

        IF (FileSeek_instance.Error=TRUE) THEN
            Stage:=INT#99;
        END_IF;                                 // 이상 종료

    3 :                             // 파일 쓰기
        FileWrite_instance(
            Execute :=TRUE,
            FileID     :=Fid,                   // 파일 ID
            WriteBuf:=OutDat[0],                // 쓰기 버퍼
            Size       :=UINT#4);               // 쓰기 요소 수 4바이트

        IF (FileWrite_instance.Done=TRUE) THEN
            Stage:=INT#4;
        END_IF;                                 // 정상 종료

        IF (FileWrite_instance.Error=TRUE) THEN
            Stage:=INT#99;
        END_IF;                                 // 이상 종료

    4 :                             // 파일 닫기
        FileClose_instance(
            Execute:=TRUE,
            FileID    :=Fid);                   // 파일 ID

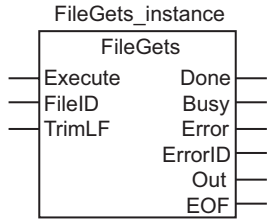
        IF (FileClose_instance.Done=TRUE) THEN
            Operating:=FALSE;
        END_IF;                                 // 정상 종료

        IF (FileClose_instance.Error=TRUE) THEN
            Stage:=INT#99;
        END_IF;                                 // 이상 종료

    99 :
        Operating:=FALSE;                       // 이상 종료 처리
    END_CASE;
END_IF;
```

FileGets

SD 메모리 카드 내의 지정된 파일로부터 1행의 문자열을 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileGets	문자열 읽기	FB		FileGets_instance(Execute, FileID, TrimLF, Done, Busy, Error, ErrorID, Out, EOF);

변수

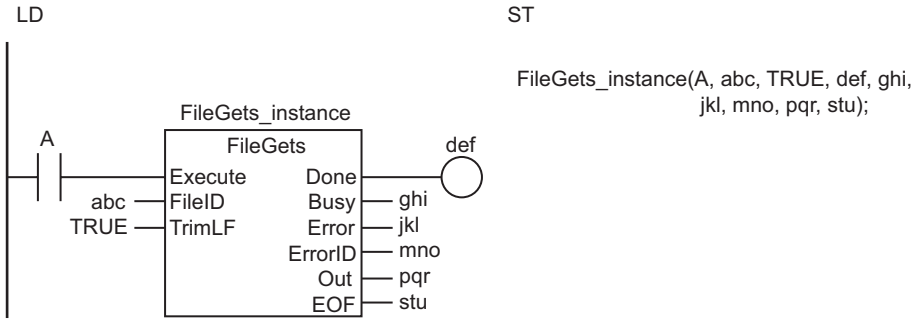
	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	읽는 파일의 파일 ID	데이터 형식에 따름	–	0
TrimLF	행바꿈 코드 삭제 지정		읽는 문자열의 행바꿈 코드의 삭제 지정 TRUE : 삭제함 FALSE : 삭제하지 않음			FALSE
Out	읽기 문자열	출력	읽은 문자열	데이터 형식에 따름	–	–
EOF	파일 끝		파일 끝에 도달했는지 판정 TRUE : 도달함 FALSE : 도달하지 않음			

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																
TrimLF	○																			
Out																				○
EOF	○																			

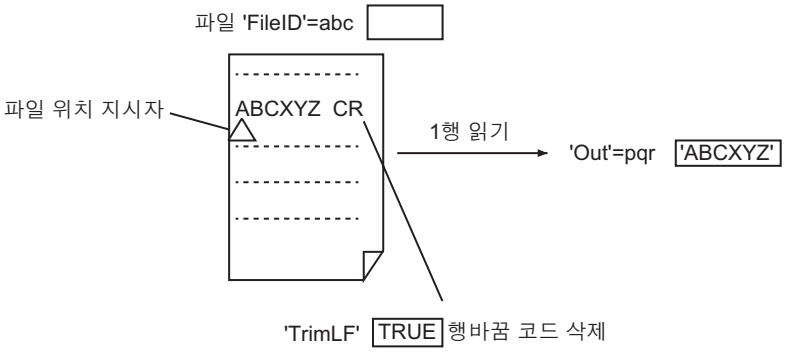
기능

SD 메모리 카드 내의 파일 ID 'FileID'로 지정한 파일의, 파일 위치 지시자가 있는 위치로부터 1행 분량의 문자열을 읽습니다.
파일 위치 지시자는 미리 FileSeek 명령으로 임의의 위치에 설정해둡니다.
행과 행의 구분은 행바꿈 코드로 식별합니다. 읽은 문자열은 읽기 문자열 'Out'에 씁니다.
행바꿈 코드는 CR, LF, CR+LF, 3종류 중에서 자동으로 판별합니다. 행바꿈 코드 삭제 지정 'TrimLF'의 값이 TRUE인 경우, 문자열 중의 행바꿈 코드를 삭제하고 'Out'에 씁니다.
또한, 파일 끝까지 읽은 경우, 파일 끝 'EOF'의 값이 TRUE가 됩니다. 그렇지 않은 경우, 'EOF'의 값은 FALSE가 됩니다.

기술 예가 아래에 나와 있습니다. 파일에서 1행 분량의 문자열을 읽고 행바꿈 코드를 삭제해서 pqr에 씁니다.



SD 메모리 카드 내의 'FileID'로 지정한 파일에서 파일 위치 지시자가 있는 위치부터
1행의 문자열을 읽어서 읽기 문자열 'Out'에 저장합니다. 이때 행바꿈 코드는 삭제됩니다.



관련된 시스템 정의 변수

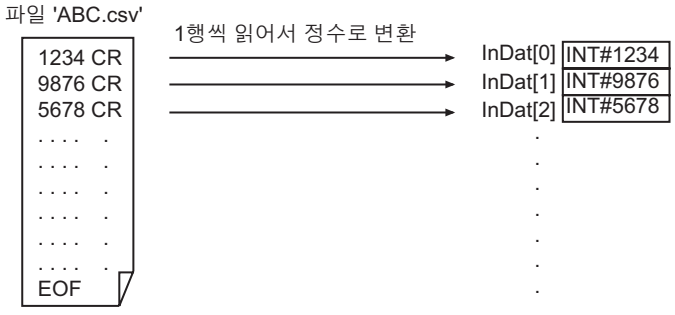
변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 1행의 문자열 길이가 1986바이트(문자 코드는 UTF-8로, 끝의 NULL 문자를 포함합니다)를 초과하는 경우, 1985바이트째까지의 문자열에 끝의 NULL 문자를 붙여서 'Out'에 저장합니다.
- 이 명령을 실행하기 전에 FileOpen 명령을 실행해서 'FileID'를 취득해두어야 합니다.
- 이 명령 실행 중에 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - 'FileID'로 지정한 파일이 읽기 가능한 모드로 열려 있지 않은 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

파일명 'ABC.csv'의 파일에는 행바꿈 코드 CR로 구분된 문자열이 여러 행 저장되어 있는 것으로 간주합니다. 문자열은 모두 숫자의 문자열입니다.
이 파일을 1행씩 읽고, 문자열을 정수로 변환해서 INT 형식 배열 변수 InDat[]에 저장합니다.
파일을 끝까지 읽으면(파일 끝 EOF까지 읽으면) 처리를 종료합니다.



처리 순서는 다음과 같습니다.

1 FileOpen 명령을 사용해서 파일 'ABC.csv'를 엽니다.

2 FileGets 명령을 사용해서 파일의 내용을 1행 읽습니다.

3 STRING_TO_INT 명령을 사용해서, 읽은 문자열을 정수로 변환하여 InDat[]에 저장합니다.

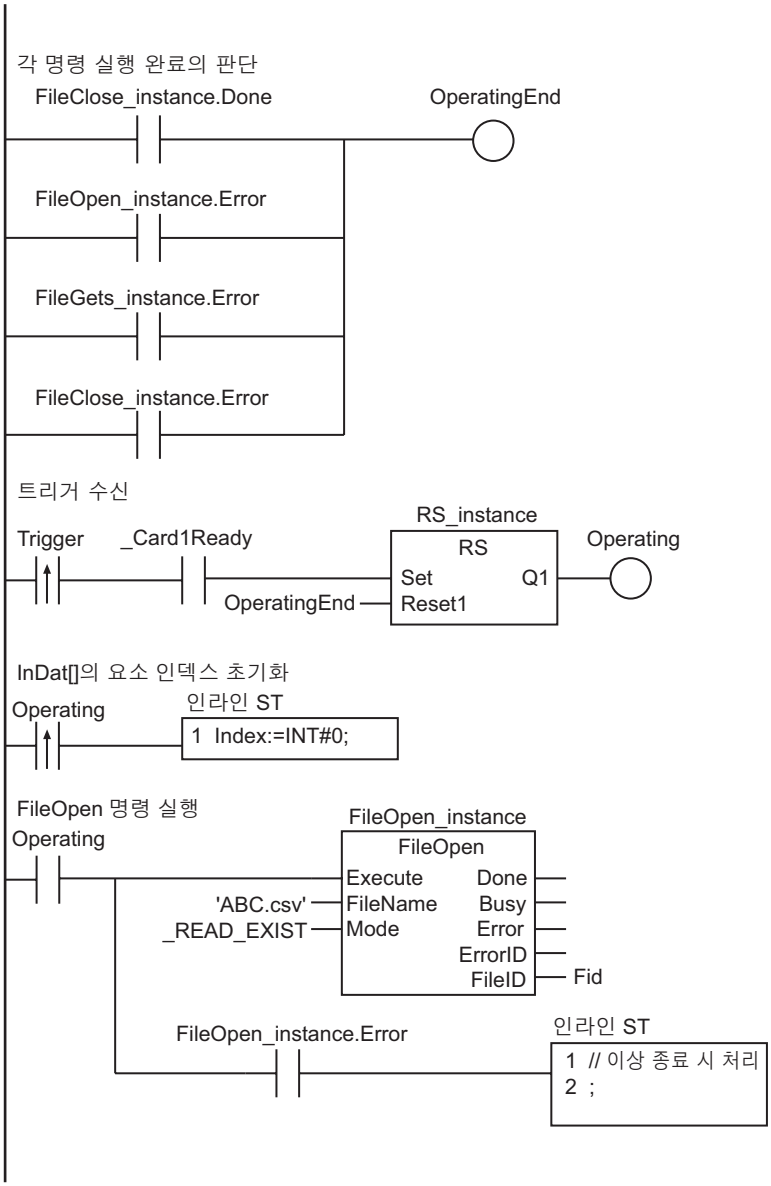
4 상기 2~3을 파일 종단 EOF를 읽을 때까지 계속합니다.

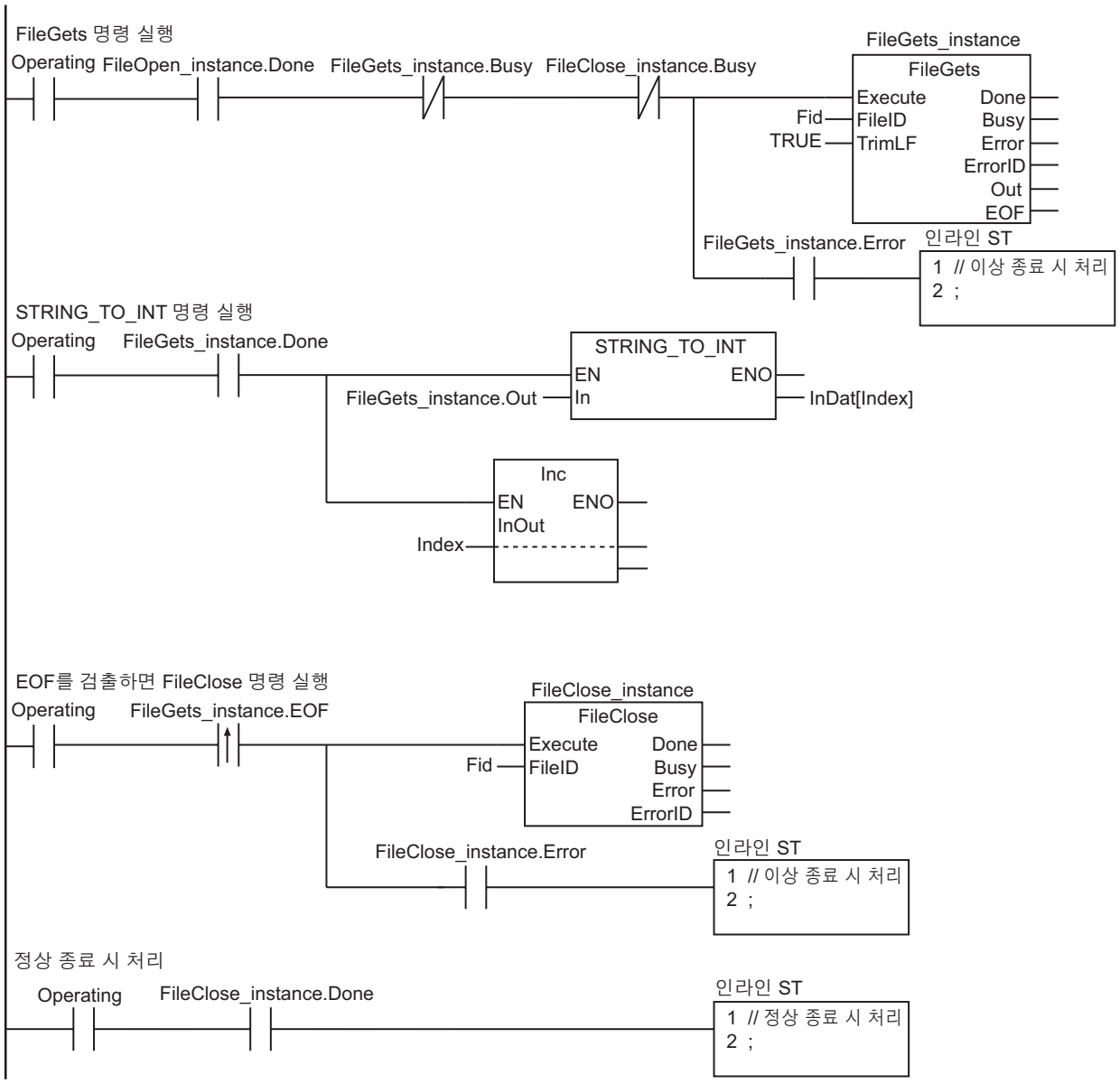
5 FileClose 명령을 사용해서 파일을 닫습니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Index	INT	0	InDat[]의 요소 인덱스
	Fid	DWORD	16#0	파일 ID
	InDat	ARRAY[0..999] OF INT	[1000(0)]	정수 데이터
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FileGets_instance	FileGets		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




FileGets 명령 실행
Operating
FileOpen_instance.Done
FileGets_instance.Busy
FileClose_instance.Busy
FileGets_instance
FileGets
Execute
Done
Fid
FileID
Busy
TRUE
TrimLF
Error
ErrorID
Out
EOF
FileGets_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
STRING_TO_INT 명령 실행
Operating
FileGets_instance.Done
STRING_TO_INT
EN
ENO
FileGets_instance.Out
In
InDat[Index]
Inc
EN
ENO
InOut
Index
EOF를 검출하면 FileClose 명령 실행
Operating
FileGets_instance.EOF
FileClose_instance
FileClose
Execute
Done
Fid
FileID
Busy
Error
ErrorID
FileClose_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
정상 종료 시 처리
Operating
FileClose_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	InDat	ARRAY[0..999] OF INT	[1000(0)]	정수 데이터
	Stage	INT	0	상태 이동
	Index	INT	0	InDat[]의 요소 인덱스
	Fid	DWORD	16#0	파일 ID
	FileOpen_instance	FileOpen		
	FileGets_instance	FileGets		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileOpen_instance(Execute:=FALSE);
    FileGets_instance(Execute:=FALSE);
    FileClose_instance(Execute:=FALSE);
    Stage          :=INT#1;
    Index          :=INT#0;
    OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
    CASE Stage OF
    1 :                     // 파일 열기
        FileOpen_instance(
            Execute  :=TRUE,
            FileName:='ABC.csv',          // 파일명
            Mode     :=_READ_EXIST,       // 파일 읽기
            FileID   =>Fid);              // 파일 ID

        IF (FileOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;           // 정상 종료
        END_IF;

        IF (FileOpen_instance.Error=TRUE) THEN
            Stage:=INT#99;          // 이상 종료
        END_IF;
```

```
    2 :                     // 문자열 읽기
        FileGets_instance(
            Execute:=TRUE,
            FileID   :=Fid,
            TrimLF  :=TRUE);

        IF (FileGets_instance.Done=TRUE) THEN
            // 읽은 문자열을 정수로 변환
            InDat[Index]:=STRING_TO_INT(FileGets_instance.Out);
            Index:=Index+INT#1;

            // 파일 종단에 도달했음
            IF (FileGets_instance.EOF=TRUE) THEN
                Stage:=INT#3;       // 정상 종료
            ELSE
                FileGets_instance(Execute:=FALSE);
            END_IF;
        END_IF;

        IF (FileGets_instance.Error=TRUE) THEN
            Stage:=INT#99;          // 이상 종료
        END_IF;

    3 :                     // 파일 닫기
        FileClose_instance(
            Execute:=TRUE,
            FileID   :=Fid);                    // 파일 ID

        IF (FileClose_instance.Done=TRUE) THEN
            Operating:=FALSE;       // 정상 종료
        END_IF;

        IF (FileClose_instance.Error=TRUE) THEN
            Stage:=INT#99;          // 이상 종료
        END_IF;

    99 :                    // 이상 종료 처리
        Operating:=FALSE;
    END_CASE;
END_IF;
```

FilePuts

SD 메모리 카드 내의 지정된 파일에 문자열을 써넣습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FilePuts	문자열 쓰기	FB	FilePuts_instance FilePuts Execute Done FileID Busy In Error ErrorID	FilePuts_instance(Execute, FileID, In, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileID	파일 ID	입력	쓰는 파일의 파일 ID	데이터 형식에 따름	–	0
In	쓰기 문자열		쓰는 문자열			''

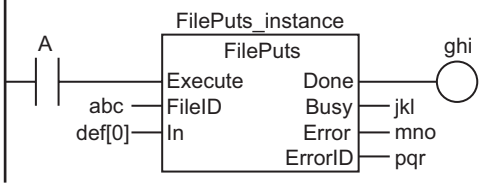
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileID				○																
In																				○

기능

SD 메모리 카드 내의 파일 ID 'FileID'로 지정한 파일의 파일 위치 지시자가 있는 위치에 문자열을 씁니다.
파일 위치 지시자는 미리 FileSeek 명령으로 임의의 위치에 설정해둡니다.
쓰는 문자열은 쓰기 문자열 'In'의 내용입니다.

기술 예가 아래에 나와 있습니다. 배열의 1개 요소 def[0]의 내용을 파일에 씁니다.

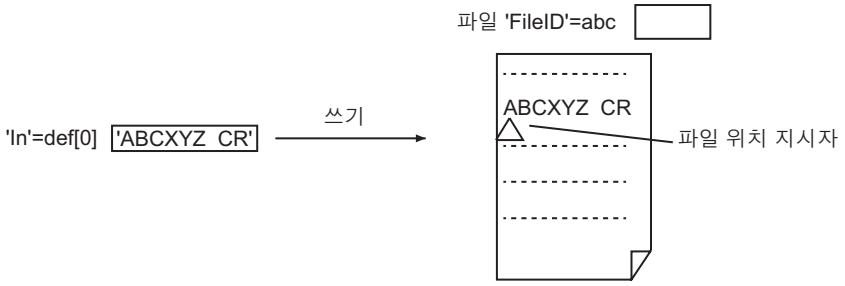
LD



ST

```
FilePuts_instance(A, abc, def[0], ghi, jkl, mno, pqr);
```

SD 메모리 카드 내의 'FileID'로 지정한 파일에서 파일 위치 지시자가 있는 위치에 쓰기 문자열 'In'의 내용을 씁니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

문자열을 쓴 후에 행바꿈을 하려는 경우, 'In'의 끝에 행바꿈 코드를 부가하십시오.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 이 명령을 실행하기 전에 FileOpen 명령을 실행해서 'FileID'를 취득해두어야 합니다.
- 이 명령 실행 중에 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - SD 메모리 카드가 여유 용량 부족인 경우
  - 'FileID'로 지정한 파일이 존재하지 않는 경우
  - 'FileID'로 지정한 파일이 액세스 중인 경우
  - 'FileID'로 지정한 파일이 쓰기 가능한 모드로 열려 있지 않은 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

INT 형식의 배열 변수 Dat[0..9, 0..99]의 내용을, 100행의 CSV 파일 형식으로 파일 'ABC.csv'에 저장합니다.

1행에는 수치의 문자열이 10개 나열되며, 그 사이에는 쉼표를 삽입합니다. 행의 끝에는 행바꿈 코드 CR+LF를 부가합니다.

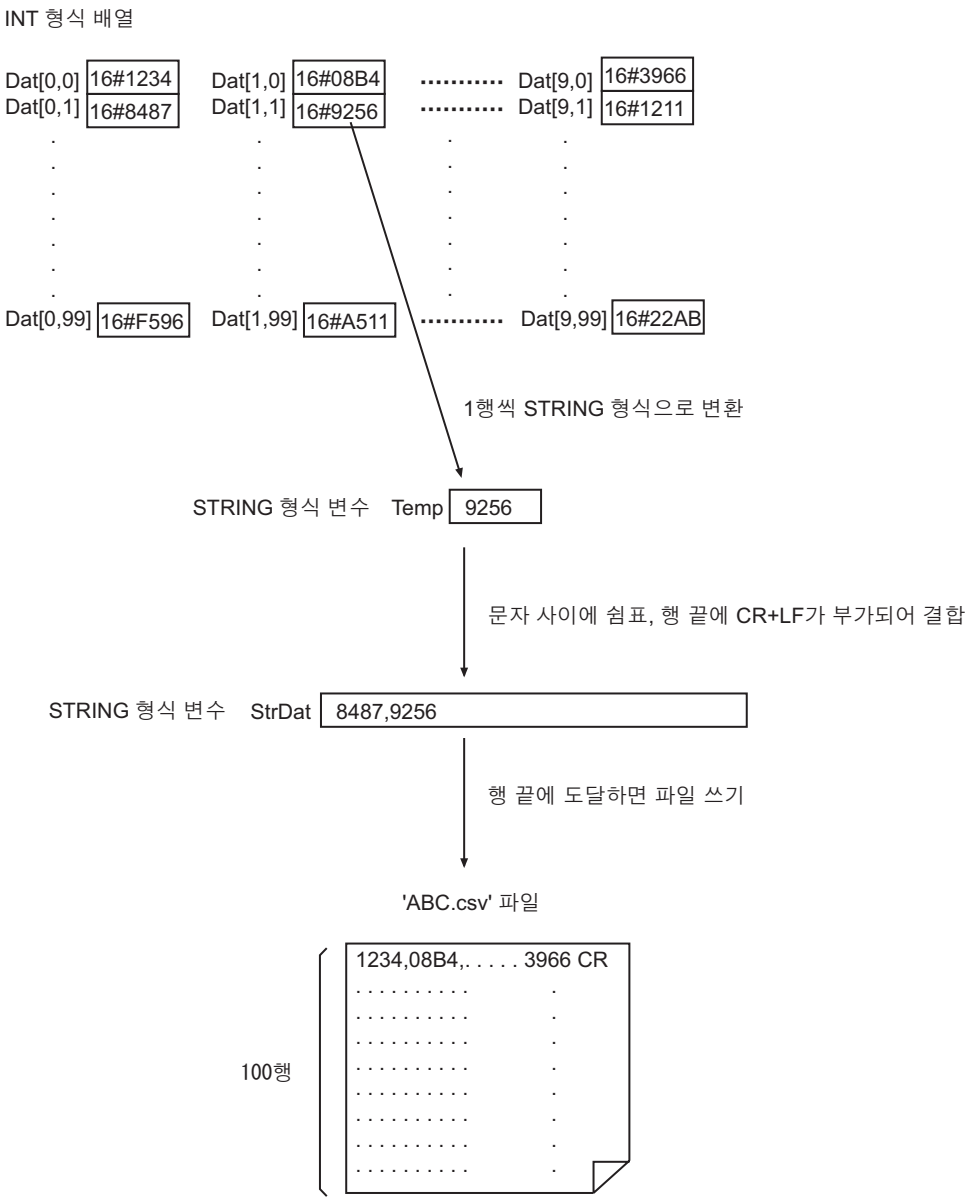
순서는 다음과 같습니다.

1 **Dat[]의 요소를 1개 문자열 변환해서 STRING 형식의 변수 Temp에 저장합니다.**

2 **Temp의 뒤에, 행 끝이 아니라면 쉼표를, 행 끝이라면 CR+LF를 부가해서 STRING 형식 변수 StrDat에 결합합니다.**

3 **행 끝에 도달하면 StrDat를 파일에 씁니다.**

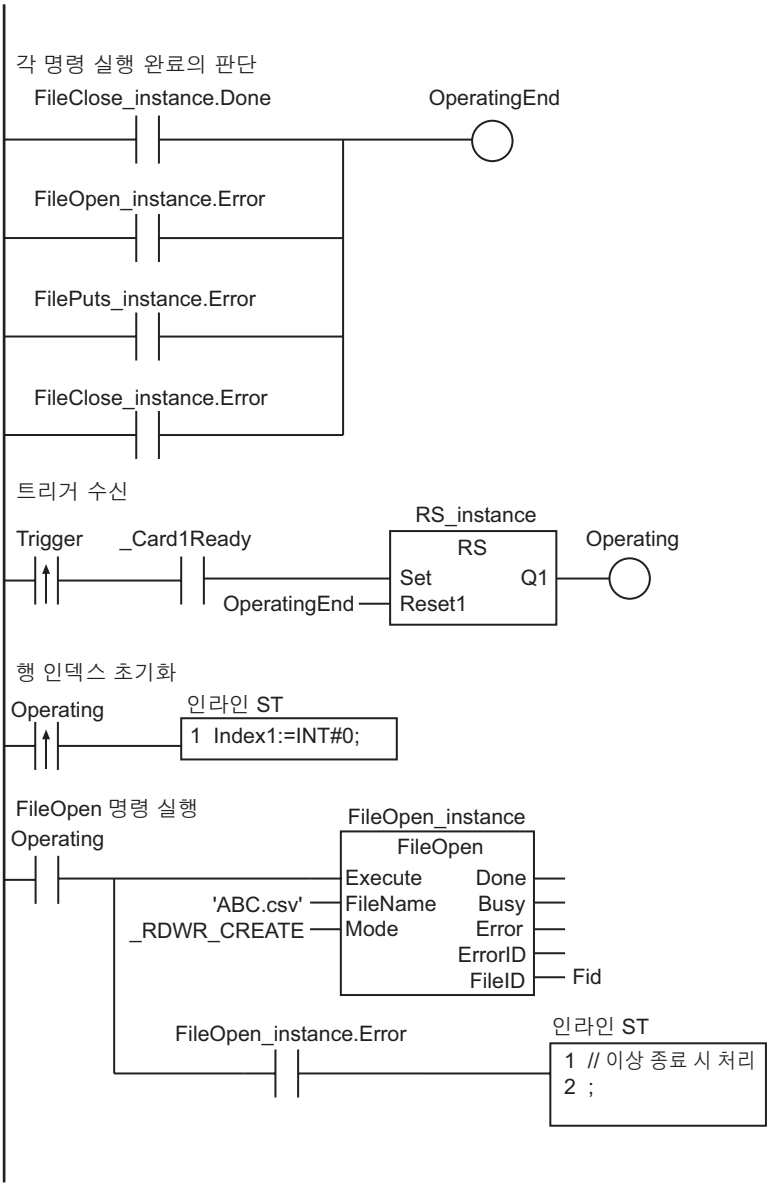
4 **1에서부터 3까지를 100행 분량 반복합니다.**

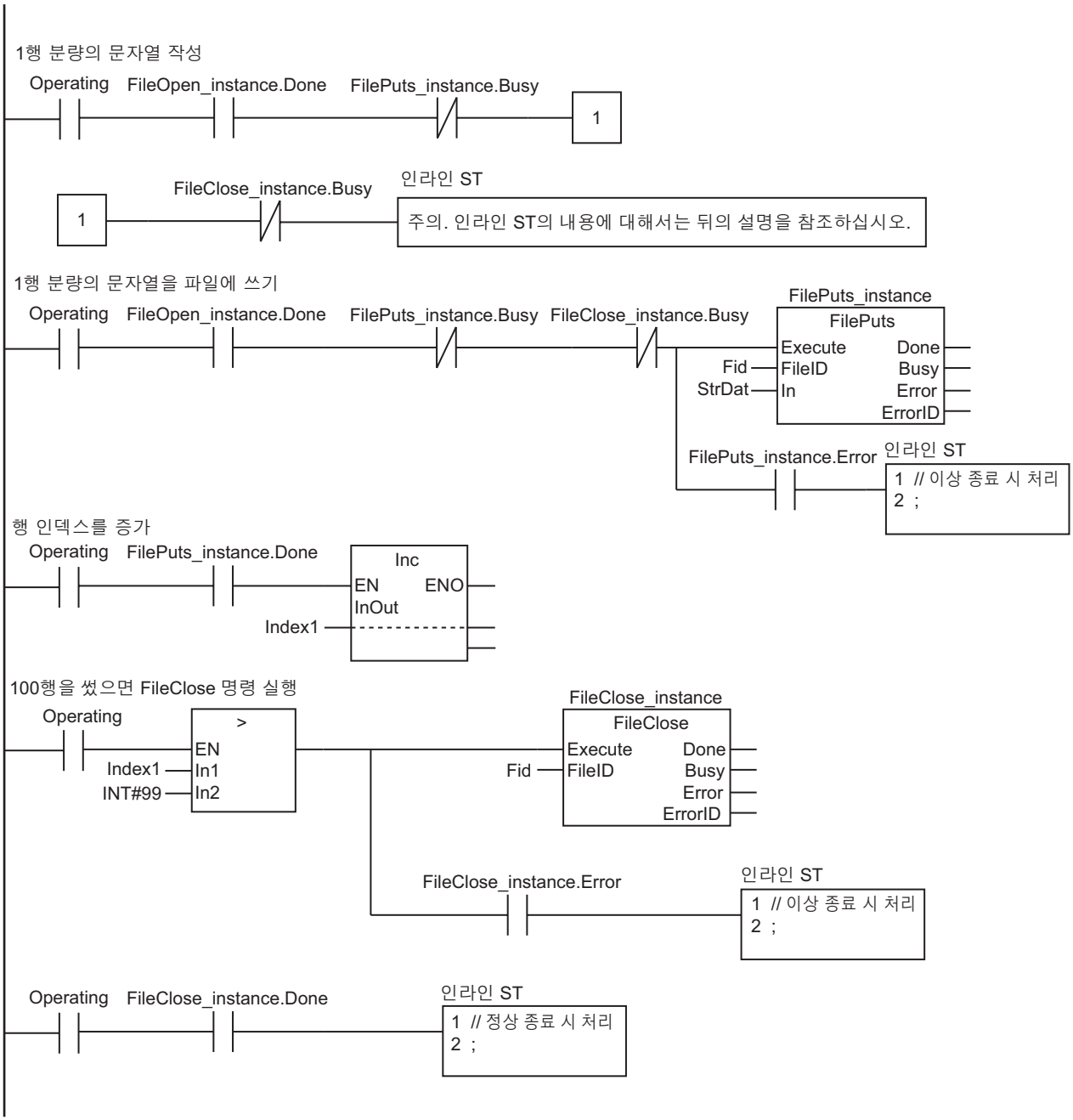

INT 형식 배열
Dat[0,0] 16#1234
Dat[0,1] 16#8487
Dat[1,0] 16#08B4
Dat[1,1] 16#9256
Dat[9,0] 16#3966
Dat[9,1] 16#1211
Dat[0,99] 16#F596
Dat[1,99] 16#A511
Dat[9,99] 16#22AB
1행씩 STRING 형식으로 변환
STRING 형식 변수 Temp 9256
문자 사이에 쉼표, 행 끝에 CR+LF가 부가되어 결합
STRING 형식 변수 StrDat 8487,9256
행 끝에 도달하면 파일 쓰기
'ABC.csv' 파일
1234,08B4,. . . . . 3966 CR
100행

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	Index0	INT	0	열 인덱스
	Index1	INT	0	행 인덱스
	Fid	DWORD	16#0	파일 ID
	StrDat	STRING[255]	''	문자열 데이터
	Dat	ARRAY[0..99,0..9] OF INT	[1000(0)]	수치 데이터
	Temp	STRING[255]	''	임시
	RS_instance	RS		
	FileOpen_instance	FileOpen		
	FilePuts_instance	FilePuts		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그





● 인라인 ST의 내용

```
StrDat:='';

// 0~8번째 문자열을 연결

FOR Index0:=INT#0 TO INT#8 BY INT#1 DO
   Temp     :=INT_TO_STRING(Dat[Index1, Index0]);
   Temp     :=CONCAT(In1:=Temp, In2:=',');
   StrDat   :=CONCAT(IN1:=StrDat, In2:=Temp);
END_FOR;

// 9번째 문자열을 연결하고 CR+LF를 추가
Temp   :=INT_TO_STRING(Dat[Index1, Index0]);
Temp   :=CONCAT(In1:=Temp, In2:='$r$l');
StrDat  :=CONCAT(In1:=StrDat, In2:=Temp);
```

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	Index0	INT	0	열 인덱스
	Index1	INT	0	행 인덱스
	Fid	DWORD	16#0	파일 ID
	StrDat	STRING[255]	''	문자열 데이터
	Dat	ARRAY[0..99,0..9] OF INT	[1000(0)]	수치 데이터
	Temp	STRING[255]	''	임시
	FileOpen_instance	FileOpen		
	FilePuts_instance	FilePuts		
	FileClose_instance	FileClose		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileOpen_instance(Execute:=FALSE);
    FilePuts_instance(Execute:=FALSE);
    FileClose_instance(Execute:=FALSE);
    Stage          :=INT#1;
    Index1         :=INT#0;    // 행 인덱스 초기화
    OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
    CASE Stage OF
    1 :                        // 파일 열기
        FileOpen_instance(
            Execute  :=TRUE,
            FileName:='ABC.csv',          // 파일명
            Mode     :=_RDWR_CREATE,      // 파일 읽기
            FileID   =>Fid);              // 파일 ID

        IF (FileOpen_instance.Done=TRUE) THEN
            Stage:=INT#2;             // 정상 종료
        END_IF;

        IF (FileOpen_instance.Error=TRUE) THEN
            Stage:=INT#99;            // 이상 종료
        END_IF;
```

```
  2 :                       // 1행 분량의 문자열 작성
      StrDat:=' ' ;

      // 0~8번째 문자열을 연결
      FOR Index0:=INT#0 TO INT#8 BY INT#1 DO
          Temp :=INT_TO_STRING(Dat[Index1, Index0]);
          Temp :=CONCAT(In1:=Temp, In2:=',');
          StrDat:=CONCAT(IN1:=StrDat, In2:=Temp);
      END_FOR;

      // 9번째 문자열을 연결해서 CR+LF를 부가
      Temp  :=INT_TO_STRING(Dat[Index1, Index0]);
      Temp  :=CONCAT(In1:=Temp, In2:='$r$l');
      StrDat:=CONCAT(In1:=StrDat, In2:=Temp);

      Stage:=INT#3;

  3 :                       // 문자열 쓰기
      FilePuts_instance(
          Execute:=TRUE,
          FileID    :=Fid,
          In          :=StrDat);

      IF (FilePuts_instance.Done=TRUE) THEN
          Index1:=Index1+INT#1;

          IF (Index1>INT#99) THEN   // 100행 쓰기 완료
              Stage:=INT#4;
          ELSE
              FilePuts_instance(Execute:=FALSE);
              Stage:=INT#2;
          END_IF;
      END_IF;

      IF (FilePuts_instance.Error=TRUE) THEN
          Stage:=INT#99;          // 이상 종료
      END_IF;

  4 :                       // 파일 닫기
      FileClose_instance(
          Execute:=TRUE,
          FileID    :=Fid);                    // 파일 ID

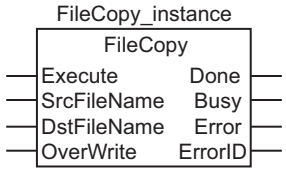
      IF (FileClose_instance.Done=TRUE) THEN
          Operating:=FALSE;       // 정상 종료
      END_IF;

      IF (FileClose_instance.Error=TRUE) THEN
          Stage:=INT#99;          // 이상 종료
      END_IF;

  99 :                      // 이상 종료 처리
      Operating:=FALSE;
    END_CASE;
END_IF;
```

FileCopy

SD 메모리 카드 내의 지정된 파일을 복사합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileCopy	파일 복사	FB		FileCopy_instance(Execute, SrcFileName, DstFileName, OverWrite, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
SrcFile Name	복사원 파일명	입력	복사원 파일명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''
DstFile Name	복사대상 파일명		복사대상 파일명			
OverWrite	덮어쓰기 허가		TRUE : 덮어쓰기 허가 FALSE : 덮어쓰기 방지	데이터 형식에 따름		FALSE

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SrcFile Name																				○
DstFile Name																				○
OverWrite	○																			

기능

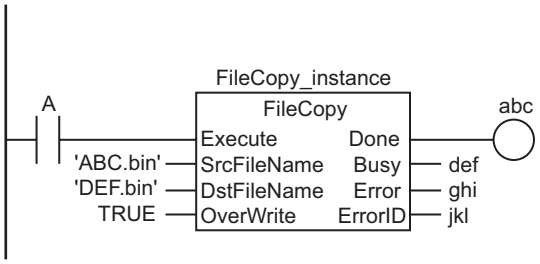
SD 메모리 카드 내의 복사원 파일명 'SrcFileName'으로 지정한 파일을 복사대상 파일명 'DstFileName'에 복사합니다.

SD 메모리 카드 내에 'DstFileName'과 같은 이름의 파일이 이미 존재하는 경우, 덮어쓰기 허가 'OverWrite'의 값에 따라 다음과 같은 처리를 수행합니다.

'OverWrite'의 값	처리
TRUE(덮어쓰기 허가)	그 파일에 덮어씁니다.
FALSE(덮어쓰기 방지)	그 파일에 덮어쓰지 못하며, 이상 상태가 됩니다.

기술 예가 아래에 나와 있습니다. 파일 'ABC.bin'을 파일 'DEF.bin'에 덮어씁니다.

LD

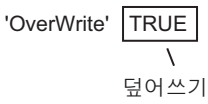


ST

```
FileCopy_instance(A, 'ABC.bin', 'DEF.bin',
                  TRUE, abc, def, ghi, jkl);
```

SD 메모리 카드 내의 복사원 파일명 'SrcFileName'으로 지정한 파일을 복사대상 파일명 'DstFileName'에 덮어씁니다.





관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 복사에 실패한 경우, SD 메모리 카드 내에 'DstFileName'으로 지정한 파일이 불완전한 상태로 남는 경우가 있습니다.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - SD 메모리 카드가 여유 용량 부족인 경우
  - 'SrcFileName'으로 지정한 파일이 존재하지 않는 경우
  - 'SrcFileName'의 값이 파일명으로서 잘못된 경우
  - 'DstFileName'의 값이 파일명으로서 잘못된 경우
  - 작성 가능한 파일 수, 디렉토리 수를 초과한 경우
  - 'SrcFileName' 또는 'DstFileName'으로 지정한 파일이 액세스 중인 경우
  - 'DstFileName'과 같은 이름의 파일이 이미 존재하며, 'OverWrite'의 값이 FALSE인 경우
  - 'DstFileName'과 같은 이름의 파일이 이미 존재하며, 그 파일이 쓰기 금지인 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'DstFileName'의 값이 파일명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

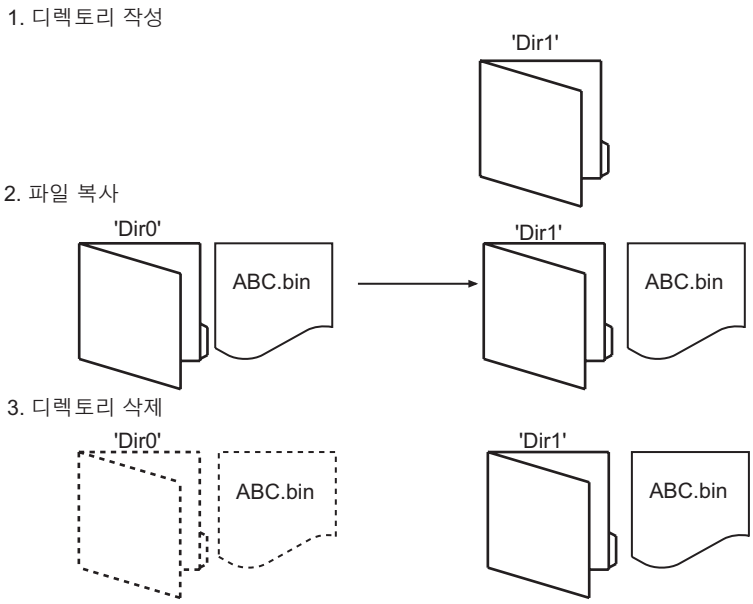
샘플 프로그램

다음과 같은 순서로 파일 이동을 수행합니다.

1 **DirCreate 명령을 사용해서 SD 메모리 카드 내에 신규 디렉토리 'Dir1'을 작성합니다.**

2 **FileCopy 명령을 사용해서, 기존의 디렉토리 'Dir0' 내의 파일 'ABC.bin'을 디렉토리 'Dir1'에 복사합니다.**

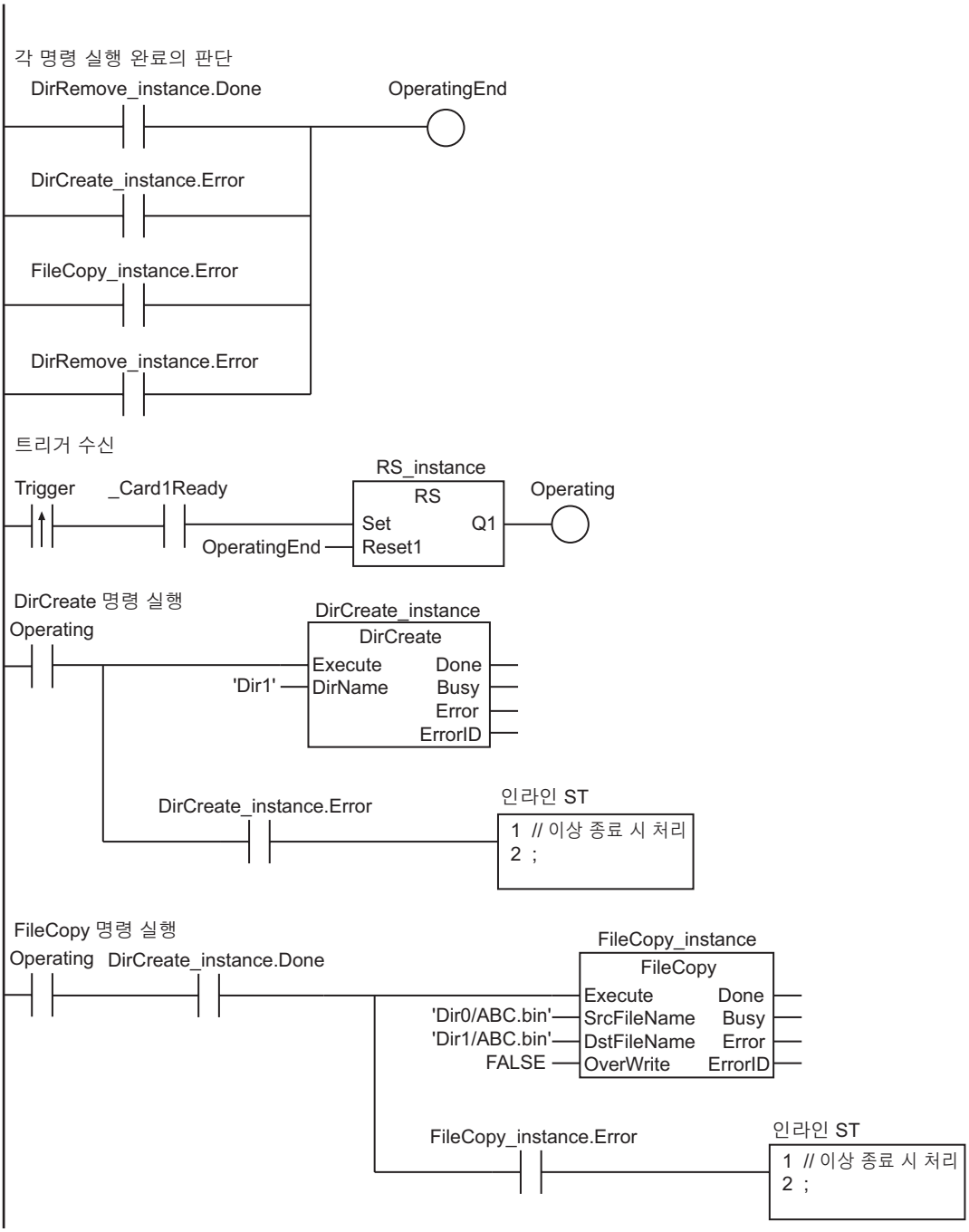
3 **DirRemove 명령을 사용해서 복사원 디렉토리 'Dir0'을 삭제합니다.**

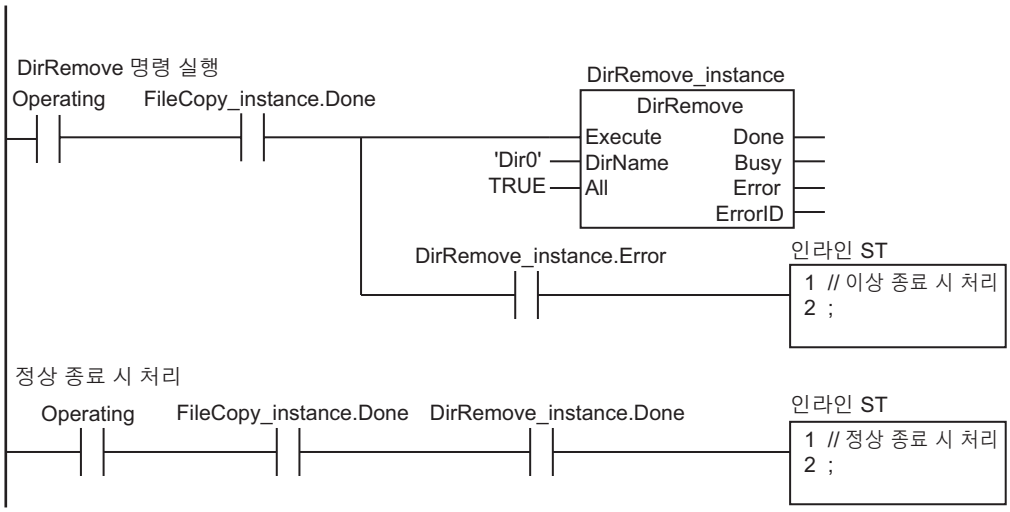


LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	RS_instance	RS		
	DirCreate_instance	DirCreate		
	FileCopy_instance	FileCopy		
	DirRemove_instance	DirRemove		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그




DirRemove 명령 실행
Operating
FileCopy_instance.Done
DirRemove_instance
DirRemove
Execute
Done
'Dir0'
DirName
Busy
TRUE
All
Error
ErrorID
DirRemove_instance.Error
인라인 ST
1 // 이상 종료 시 처리
2 ;
정상 종료 시 처리
Operating
FileCopy_instance.Done
DirRemove_instance.Done
인라인 ST
1 // 정상 종료 시 처리
2 ;

ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	Stage	INT	0	상태 이동
	DirCreate_instance	DirCreate		
	FileCopy_instance	FileCopy		
	DirRemove_instance	DirRemove		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    DirCreate_instance(Execute:=FALSE);
    FileCopy_instance(Execute:=FALSE);
    DirRemove_instance(Execute:=FALSE);
    Stage           :=INT#1;
    OperatingStart:=FALSE;
END_IF;

// 각 명령 실행
IF (Operating=TRUE) THEN
    CASE Stage OF
    1 :                         // 디렉토리 작성
        DirCreate_instance(
            Execute :=TRUE,
            DirName:='Dir1');                   // 디렉토리명

        IF (DirCreate_instance.Done=TRUE) THEN
            Stage:=INT#2;                       // 정상 종료
        END_IF;

        IF (DirCreate_instance.Error=TRUE) THEN
            Stage:=INT#99;                      // 이상 종료
        END_IF;
```

```
    2 :                             // 파일 복사
        FileCopy_instance(
            Execute :=TRUE,
            SrcFileName:='Dir0/ABC.bin',        // 복사원 파일명
            DstFileName:='Dir1/ABC.bin',        // 복사대상 파일명
            OverWrite    :=FALSE);              // 덮어쓰기 금지

        IF (FileCopy_instance.Done=TRUE) THEN
            Stage:=INT#3;
        END_IF;

        IF (FileCopy_instance.Error=TRUE) THEN
            Stage:=INT#99;
        END_IF;

    3 :                             // 디렉토리 삭제
        DirRemove_instance(
            Execute :=TRUE,
            DirName:='Dir0',                    // 디렉토리명
            All         :=TRUE);                // 파일 및 하위 디렉토리별 삭제

        IF (DirRemove_instance.Done=TRUE) THEN
            Operating:=FALSE;                   // 정상 종료
        END_IF;

        IF (DirRemove_instance.Error=TRUE) THEN
            Stage:=INT#99;                      // 이상 종료
        END_IF;

    99 :                            // 이상 종료 처리
        Operating:=FALSE;
    END_CASE;
END_IF;
```

FileRemove

SD 메모리 카드 내의 지정된 파일을 삭제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
FileRemove	파일 삭제	FB	FileRemove_instance FileRemove Execute Done FileName Busy Error ErrorID	FileRemove_instance(Execute, FileName, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	지정 파일명	입력	삭제할 파일명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''

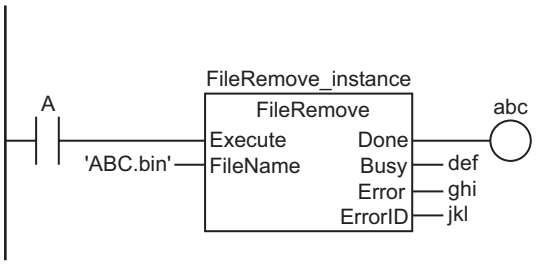
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○

기능

SD 메모리 카드 내의 지정 파일명 'FileName'으로 지정한 파일을 삭제합니다.

기술 예가 아래에 나와 있습니다. 파일 'ABC.bin'을 삭제합니다.

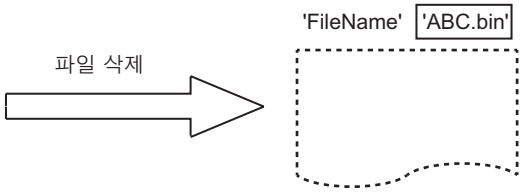
LD



ST

```
FileRemove_instance(A, 'ABC.bin', abc,
                    def, ghi, jkl);
```

SD 메모리 카드 내의 'FileName'으로 지정한 파일을 삭제합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - 'FileName'으로 지정한 파일이 존재하지 않는 경우
  - 'FileName'으로 지정한 파일이 액세스 중인 경우
  - 'FileName'과 같은 이름의 파일이 이미 존재하며, 그 파일이 쓰기 금지인 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'FileName'의 값이 파일명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

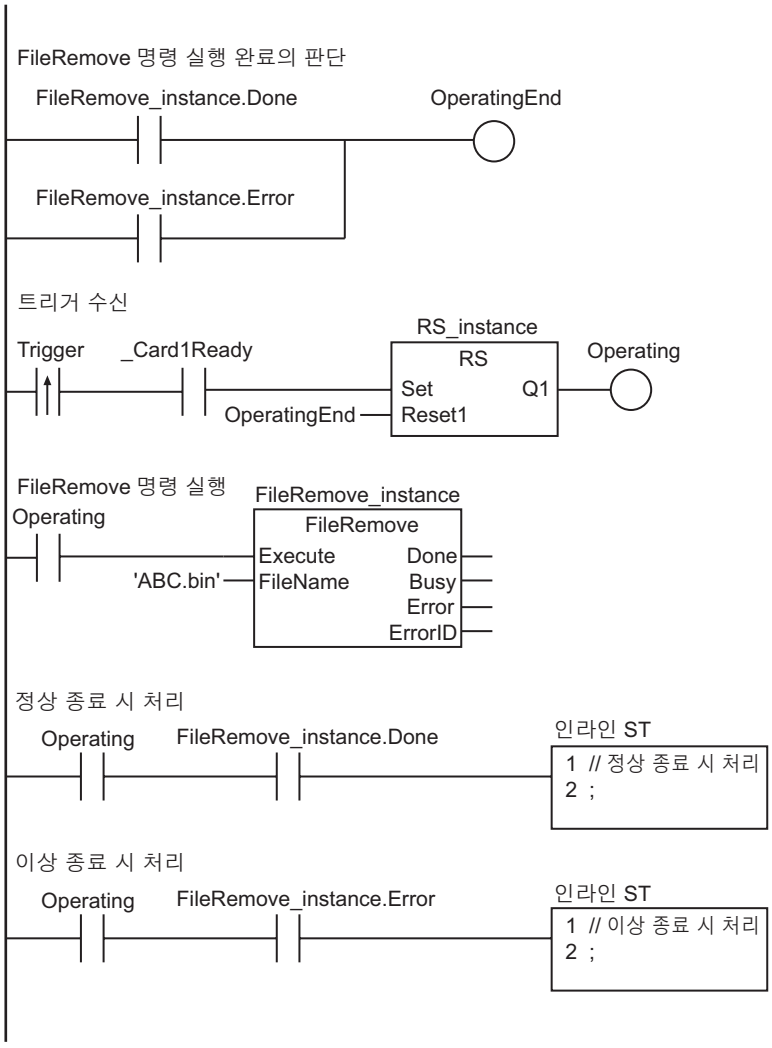
샘플 프로그램

SD 메모리 카드 내의 파일 'ABC.bin'을 삭제합니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	RS_instance	RS		
	FileRemove_instance	FileRemove		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	FileRemove_instance	FileRemove		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileRemove_instance(Execute:=FALSE);
    OperatingStart:=FALSE;
END_IF;

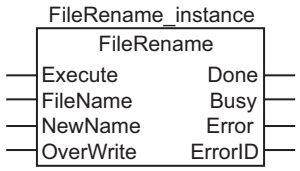
// FileRemove 명령 실행
IF (Operating=TRUE) THEN
    FileRemove_instance(
            Execute  :=TRUE,
            FileName:='ABC.bin');     // 파일명

    IF (FileRemove_instance.Done=TRUE) THEN
        Operating:=FALSE;             // 정상 종료
    END_IF;

    IF (FileRemove_instance.Error=TRUE) THEN
        Operating:=FALSE;             // 이상 종료
    END_IF;
END_IF;
```

FileRename

SD 메모리 카드 내의 지정된 파일 및 디렉토리의 명칭을 변경합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
FileRename	파일명 변경	FB		FileRename_instance(Execute, FileName, NewName, OverWrite, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
FileName	원본 파일명	입력	원래의 파일명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''
NewName	변경 후 파일명		변경 후의 파일명			
OverWrite	덮어쓰기 허가		TRUE : 덮어쓰기 허가 FALSE : 덮어쓰기 방지	데이터 형식에 따름		FALSE

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
FileName																				○
NewName																				○
OverWrite	○																			

기능

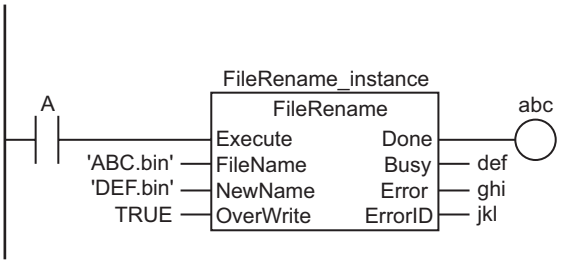
SD 메모리 카드 내의 원본 파일명 'FileName'으로 지정한 파일 및 디렉토리의 명칭을, 변경 후 파일명 'NewName'으로 변경합니다.

SD 메모리 카드 내에 'NewName'과 같은 이름의 파일 및 디렉토리가 이미 존재하는 경우, 덮어쓰기 허가 'OverWrite'의 값에 따라 다음과 같은 처리를 수행합니다.

'OverWrite'의 값	처리
TRUE(덮어쓰기 허가)	그 파일 및 디렉토리에 덮어씁니다.
FALSE(덮어쓰기 방지)	그 파일 및 디렉토리에 덮어쓰지 못하며, 이상 상태가 됩니다.

기술 예가 아래에 나와 있습니다. 파일 'ABC.bin'의 명칭을 'DEF.bin'으로 변경합니다.

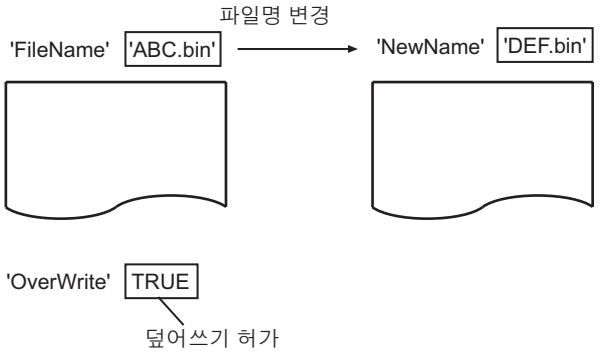
LD



ST

```
FileRename_instance(A, 'ABC.bin', 'DEF.bin',
                    TRUE, abc, def, ghi, jkl);
```

SD 메모리 카드 내의 원본 파일명 'FileName'으로 지정한 파일의 이름을 변경한 후 파일명 'NewName'에 덮어써서 변경합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 'FileName'과 'NewName'의 디렉토리가 다른 경우, 파일은 'NewName'으로 지정한 디렉토리로 이동합니다.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - 'FileName'으로 지정한 파일 디렉토리가 존재하지 않는 경우
  - 'FileName' 또는 'NewName'의 값이 파일명 및 디렉토리명으로서 잘못된 경우
  - 'FileName'으로 지정한 파일이 액세스 중인 경우
  - 'FileName'으로 지정한 디렉토리 내에 하위 디렉토리가 존재하며, 'OverWrite'의 값이 TRUE인 경우
  - 'NewName'과 같은 이름의 파일이 이미 존재하며, 'OverWrite'의 값이 FALSE인 경우
  - 'NewName'과 같은 이름의 파일이 이미 존재하며, 그 파일이 쓰기 금지 상태이고 'OverWrite'의 값이 TRUE인 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'NewName'의 값이 파일명 및 디렉토리명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우
  - 작성 가능한 디렉토리 수를 초과한 경우

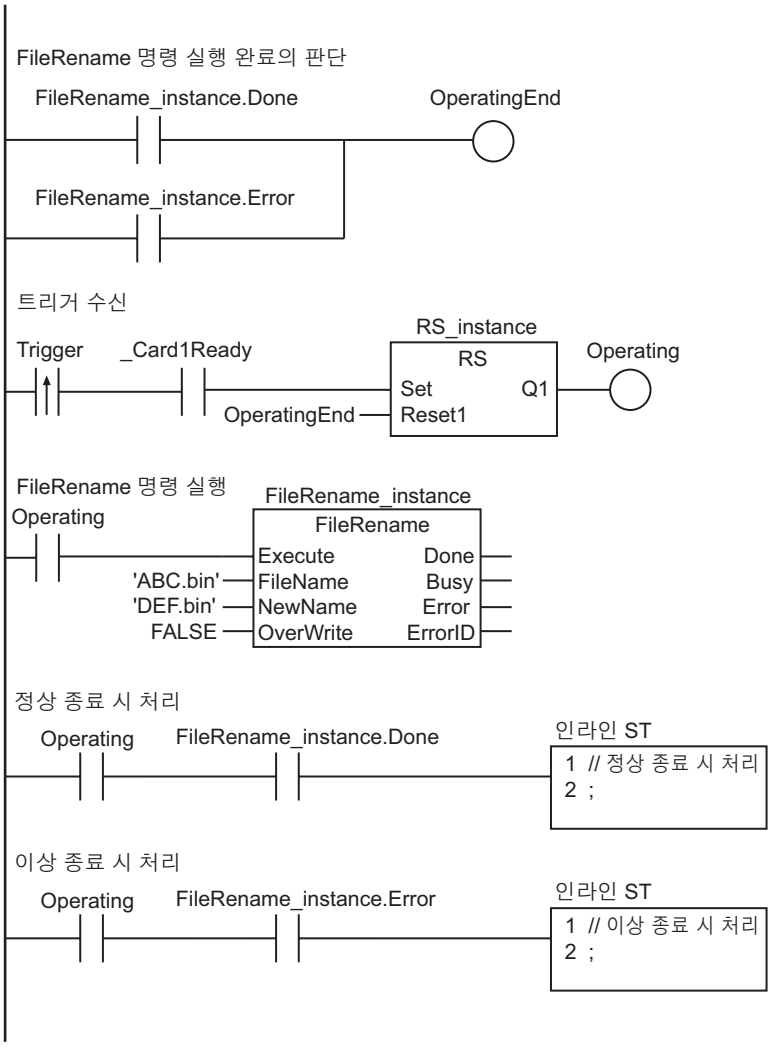
샘플 프로그램

SD 메모리 카드 내의 파일 'ABC.bin'의 명칭을 'DEF.bin'으로 변경합니다.

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	OperatingEnd	BOOL	FALSE	처리 종료
	Trigger	BOOL	FALSE	실행 조건
	Operating	BOOL	FALSE	처리 중
	RS_instance	RS		
	FileRename_instance	FileRename		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	Trigger	BOOL	FALSE	실행 조건
	LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
	OperatingStart	BOOL	FALSE	처리 시작
	Operating	BOOL	FALSE	처리 중
	FileRename_instance	FileRename		

외부 변수	명칭	데이터 형식	코멘트
	_Card1Ready	BOOL	SD 메모리 카드 사용 가능 플래그

```
// Trigger의 기동으로 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) AND (_Card1Ready=TRUE) ) THEN
    OperatingStart:=TRUE;
    Operating      :=TRUE;
END_IF;
LastTrigger:=Trigger;

// 인스턴스 초기화
IF (OperatingStart=TRUE) THEN
    FileRename_instance(Execute:=FALSE);
    OperatingStart:=FALSE;
END_IF;

// FileRename 명령 실행
IF (Operating=TRUE) THEN
    FileRename_instance(
            Execute   :=TRUE,
            FileName :='ABC.bin',     // 원본 파일명
            NewName:='DEF.bin',       // 변경 후 파일명
            OverWrite :=FALSE);       // 덮어쓰기 금지

    IF (FileRename_instance.Done=TRUE) THEN
        Operating:=FALSE;             // 정상 종료
    END_IF;

    IF (FileRename_instance.Error=TRUE) THEN
        Operating:=FALSE;             // 이상 종료
    END_IF;
END_IF;
```

DirCreate

SD 메모리 카드 내에 지정된 명칭의 디렉토리를 작성합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DirCreate	디렉토리 작성	FB	DirCreate_instance DirCreate Execute Done DirName Busy Error ErrorID	DirCreate_instance(Execute, DirName, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
DirName	작성 디렉토리명	입력	작성할 디렉토리명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''

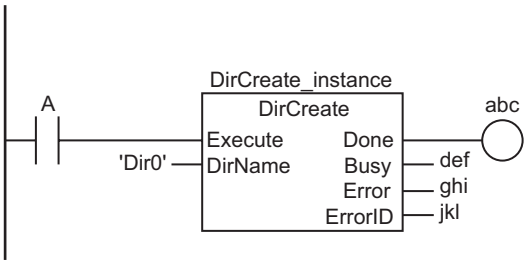
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
DirName																				○

기능

SD 메모리 카드 내에, 작성 디렉토리명 'DirName'으로 지정한 명칭의 디렉토리를 작성합니다.

기술 예가 아래에 나와 있습니다. 디렉토리 'Dir0'을 작성합니다.

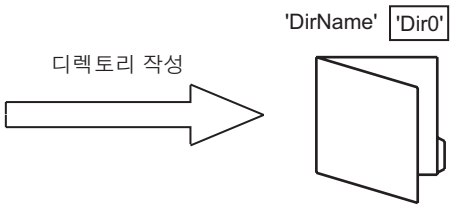
LD



ST

```
DirCreate_instance(A, 'Dir0', abc,
                   def, ghi, jkl);
```

SD 메모리 카드 내에 'DirName'으로 지정한 이름의 디렉토리를 작성합니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

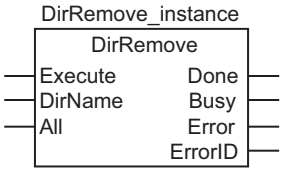
- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - SD 메모리 카드가 여유 용량 부족인 경우
  - 작성 가능한 디렉토리 수를 초과한 경우
  - 'DirName'으로 지정한 디렉토리가 이미 존재하는 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'DirName'의 값이 디렉토리명으로서 잘못된 경우
  - 'DirName'의 값이 디렉토리명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우
  - 'FileName'으로 지정한 파일이 액세스 중인 경우

샘플 프로그램

'FileCopy 명령(2-1092페이지)'의 샘플 프로그램을 참조하십시오.

DirRemove

SD 메모리 카드 내의 지정된 디렉토리를 삭제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
DirRemove	디렉토리 삭제	FB		DirRemove_instance(Execute, DirName, All, Done, Busy, Error, ErrorID);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
DirName	삭제 디렉토리명	입력	삭제할 디렉토리명	최대 66바이트 (반각 영숫자 65문자 +끝의 NULL 문자)	–	''
All	전체 지정		디렉토리 내에 파일/하위 디렉토리가 존재하는 경우의 지정 TRUE : 파일/하위 디렉토리마다 삭제함 FALSE : 삭제하지 않음	데이터 형식에 따름		FALSE

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
DirName																				○
All	○																			

기능

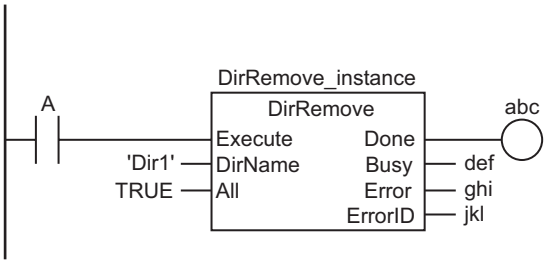
SD 메모리 카드 내의 삭제 디렉토리명 'DirName'으로 지정한 디렉토리를 삭제합니다.

지정한 디렉토리 내에 파일 및 하위 디렉토리가 존재하는 경우, 전체 지정 'All'의 값에 따라 다음과 같은 처리를 수행합니다.

'All'의 값	처리
TRUE	파일 및 하위 디렉토리마다 지정한 디렉토리를 삭제합니다.
FALSE	지정한 디렉토리를 삭제하지 않고 이상 상태가 됩니다.

기술 예가 아래에 나와 있습니다. 디렉토리 'Dir1'을 삭제합니다.

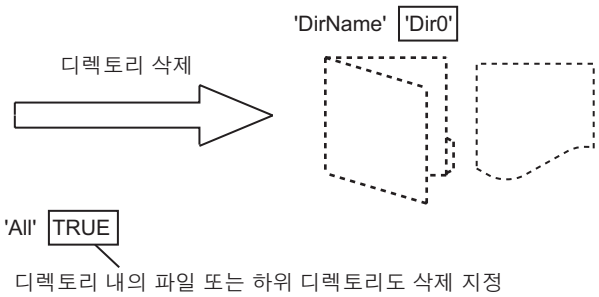
LD



ST

```
DirRemove_instance(A, 'Dir1', TRUE, abc,
                   def, ghi, jkl);
```

SD 메모리 카드 내에서 'DirName'으로 지정한 이름의 디렉토리를 삭제합니다.
디렉토리 내의 파일 또는 하위 디렉토리도 삭제됩니다.



관련된 시스템 정의 변수

변수명	명칭	데이터 형식	내용
_Card1Ready	SD 메모리 카드 사용 가능 플래그	BOOL	SD 메모리 카드가 물리적으로 장착되어, 마운트 처리가 정상적으로 종료되고 명령 및 통신 명령으로 액세스 가능해졌는지 나타내는 플래그입니다. TRUE : 사용 가능 FALSE : 사용 불가능
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	BOOL	SD 메모리 카드가 장착되어, 사용 가능해진 상태에서 SD 메모리 카드에 쓰기 방지가 걸려있는지 나타내는 플래그입니다. TRUE : 쓰기 방지 있음 FALSE : 쓰기 방지 없음
_Card1Err	SD 메모리 카드 이상 플래그	BOOL	사양 외의 SD 메모리 카드(예: SDHC 카드)가 장착된 경우, 또는 포맷 이상(FAT16 이외이거나 파일 시스템이 손상)인지를 나타내는 플래그입니다. TRUE : 이상 있음 FALSE : 이상 없음
_Card1Access	SD 메모리 카드 액세스 중 플래그	BOOL	SD 메모리 카드에 액세스 중인지를 나타내는 플래그입니다. TRUE : 액세스 중 FALSE : 액세스 중이 아님
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	BOOL	SD 메모리 카드 액세스 중에 전원 OFF가 발생해 실행 중인 처리가 이상 종료되었음을 나타내는 플래그입니다. 이 플래그는 자동으로 클리어되지 않습니다. TRUE : 이상 있음 FALSE : 이상 없음

참고

파일명의 루트 디렉토리는 SD 메모리 카드 바로 아래를 가리킵니다.

사용상의 주의 사항

- 이 명령은 'Execute'의 값이 FALSE가 되거나 실행 시간이 태스크 주기를 초과해도 처리를 끝까지 계속합니다. 처리가 정상 종료되었는지의 여부는 'Done'의 값이 TRUE가 되었는지를 통해 확인하십시오.
- 'Execute', 'Done', 'Busy', 'Error'의 타이밍 차트는 '이 장을 보는 방법(2-3페이지)'을 참조하십시오.
- 파일이 열린 상태에서 CPU 유니트 동작 모드를 프로그램으로 변경한 경우, 또는 전체 정지 이상 레벨의 컨트롤러 이상이 발생한 경우 그 파일을 강제적으로 닫습니다. 단, 그때 실행 중인 데이터의 읽기와 쓰기는 끝까지 수행합니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치 조작으로 전원 공급을 정지한 경우, 파일은 손상되지 않습니다.
- 파일이 열려 있는 상태에서 전원 공급 정지 스위치를 조작하지 않고 SD 메모리 카드를 빼낸 경우, 파일의 내용이 손상될 가능성이 있습니다. SD 메모리 카드를 빼낼 때는 반드시 전원 공급 정지를 수행하십시오.
- 파일이 열려 있는 상태에서 전원 공급 정지 또는 SD 메모리 카드를 빼낸 경우, 다시 SD 메모리 카드를 장착하더라도 파일의 읽기/쓰기는 불가능합니다.
- 'DirName'으로 지정한 디렉토리가 쓰기 금지인 경우, 이상 상태가 되며 그 디렉토리는 삭제되지 않습니다. 하지만, 그 디렉토리 내에 존재하는 쓰기 금지가 아닌 파일 및 쓰기 금지가 아닌 디렉토리는 삭제됩니다.
- 동일한 파일에 대한 엑세스를 동시에 실시하지 마십시오. 사용자 프로그램 내에서 SD 메모리 카드 명령 간의 배타적 제어를 실시하십시오.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Error'는 TRUE가 됩니다.
  - SD 메모리 카드가 사용 가능 상태가 아닌 경우
  - SD 메모리 카드가 쓰기 방지 중인 경우
  - 'All'의 값이 TRUE이며 'DirName'으로 지정한 디렉토리가 다른 명령으로부터 액세스 중인 경우
  - 'All'의 값이 FALSE이며, 'DirName'으로 지정한 디렉토리 내에 파일 및 디렉토리가 존재하는 경우
  - 'DirName'으로 지정한 디렉토리가 쓰기 금지인 경우
  - 'DirName'으로 지정한 디렉토리 내에 쓰기 금지 파일 및 쓰기 금지 디렉토리가 존재하는 경우
  - 변수에 'FileID'를 갖지 않는 SD 메모리 카드 관련 명령(FileWriteVar, FileReadVar, FileCopy, DirCreate, FileRemove, DirRemove, FileRename)을 5개 이상 동시에 실행한 경우
  - 'DirName'으로 지정한 디렉토리가 존재하지 않는 경우
  - 'DirName'의 값이 디렉토리명으로 사용할 수 있는 바이트 수를 초과한 경우
  - SD 메모리 카드 액세스 중에 어떤 이상이 발생해서 액세스할 수 없게 된 경우

샘플 프로그램

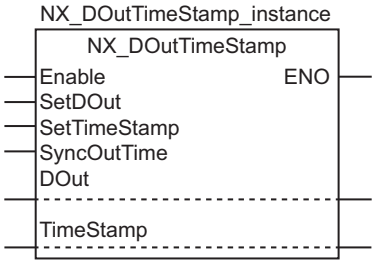
'FileCopy 명령(2-1092페이지)'의 샘플 프로그램을 참조하십시오.

타임 스탬프 명령

명령	명칭	페이지
NX_DOutTimeStamp	타임 스탬프 디지털 출력 쓰기	2-1118
NX_AryDOutTimeStamp	타임 스탬프 디지털 출력 배열 쓰기	2-1124

NX_DOutTimeStamp

타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점에 값을 씁니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_DOut TimeStamp	타임 스탬프 디지털 출력 쓰기	FB		NX_DOutTimeStamp_instance(Enable, SetDOut, SetTimeStamp, SyncOutTime, DOut, TimeStamp);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	유효	입력	TRUE: 'SetDOut' 값을 출력 FALSE: 하강 시 출력을 FALSE 로 설정	데이터 형식에 따름	–	FALSE
SetDOut	출력값		출력값			
SetTime Stamp	지정 타임 스탬프		출력되는 시각		ns	0
SyncOut Time	동기 출력 시각 정보		EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Synchronous Output'			(*)
DOut	DOut 유니트 출력 접점	입출력	타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수 'Output Bit **'	데이터 형식에 따름	–	–
TimeStamp	타임 스탬프		타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수 'Output Bit ** Time Stamp'		ns	

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
SetDOut	○																			
SetTime Stamp									○											
SyncOut Time									○											
DOut	○																			
TimeStamp									○											

기능

유효 'Enable'의 값이 TRUE인 경우, 지정한 시각에 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점에 출력값 'SetDOut'을 씁니다.
'Enable'의 값이 TRUE에서 FALSE로 하강하면 다음 태스크 주기부터 출력 접점의 값이 FALSE가 됩니다.
지정된 시각과 출력 시각의 오차는 최대 μs입니다.

동기 출력 시각 정보 'SyncOutTime'은 타임 스탬프 방식 대응 디지털 출력 유니트가 연결되어 있는 EtherCAT 커플러 유니트의 기준 시각 정보입니다. 연결되어 있는 EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Syncronous Output'을 지정하십시오.
단, EtherCAT 커플러 유니트의 I/O 항목에 0x200A:02(Time Stamp of Synchronous Output)를 추가해야 합니다.

DOut 유니트 출력 접점 'DOut'에는 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점에 할당된 디바이스 변수 'Output Bit **'를 지정하십시오.

타임 스탬프 'TimeStamp'에는 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점의 시각 정보에 할당된 디바이스 변수 'Output Bit ** Time Stamp'를 지정하십시오.

출력 시각의 지정

출력 시각은 다음 순서로 지정합니다.

1 **기준 시각으로 사용할 유니트의 접점의 시각 정보에 할당된 디바이스 변수를 취득합니다.**

2 **취득한 시각 정보와 출력 접점에 데이터를 쓰는 시각의 차를 ns 단위의 수치로 하여 1)에서 취득한 변수에 더합니다.**

3 **더한 결과를 이 명령의 지정 타임 스탬프 'SetTimeStamp'에 전달합니다.**

자세한 내용은 이 명령의 샘플 프로그램을 참고하십시오.

사용상의 주의 사항

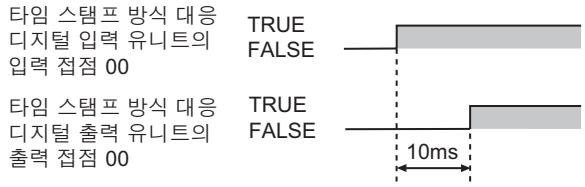
- 이 명령은 타임 스탬프 방식 대응 디지털 출력 유니트에 대해서만 실행할 수 있습니다. 단, 타임 스탬프 방식 대응 디지털 출력 유니트를 연결하지 않은 상태에서 이 명령을 실행해도 이상 상태가 되지는 않습니다.
- EtherCAT 통신 이상 또는 태스크 주기 초과가 발생한 경우는 지정된 시각에 쓸 수 없는 경우가 있습니다. 그 경우, 다음 태스크 주기 이후에 출력됩니다.
- 이 명령에서 사용하는 디바이스 변수를 다른 명령이나 프로그램에서 변경하거나 참조하는 경우는 배타적으로 처리하십시오.
- 'SyncOutTime'에는 타임 스탬프 방식 대응 디지털 출력 유니트가 직접 연결되어 있는 EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Synchronous Output'을 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.
- 'DOut'과 'TimeStamp'에는 출력할 타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수를 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.
- 'DOut'과 'TimeStamp'에는 동일한 유니트의 동일한 채널 번호의 디바이스 변수를 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.
- 'SetTimeStamp'에 과거의 시각을 지정하면 다음 태스크 주기에 쓰기가 실행됩니다. 이 경우, 'TimeStamp'의 값은 0이 됩니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상, Sysmac Studio Ver.1.07 이상에서 사용할 수 있습니다.

샘플 프로그램

타임 스탬프 방식 대응 디지털 입력 유니트의 입력 접점 00의 값이 FALSE에서 TRUE로 변화한 경우, 그 10ms 후에 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점 00의 값을 FALSE에서 TRUE로 변화시킵니다.



네트워크 구성

네트워크 구성은 다음과 같습니다. EtherCAT의 노드 주소 1에 아래 구성의 슬레이브 터미널이 연결되어 있습니다. 각각에는 다음과 같은 장치명이 할당되어 있습니다.

유니트 번호	형식	유니트	장치명
0	NX-ECC201	EtherCAT 커플러 유니트	E001
1	NX-ID3344	타임 스탬프 방식 대응 디지털 입력 유니트	N1
2	NX-OD2154	타임 스탬프 방식 대응 디지털 출력 유니트	N2

유니트 동작 설정

타임 스탬프 방식 대응 디지털 입력 유니트의 유니트 동작 설정은 다음과 같습니다.

항목명	설정값	의미
타임 스탬프(트리거 설정)/Input Bit 00 Trigger Setting	FALSE	변화 시각을 취득하는 에지: 기동
타임 스탬프(모드 설정)/Input Bit 00 Mode Setting	TRUE	변화 시각을 취득하는 동작 모드: 단발 취득(최초로 변화된 시각)

I/O 맵

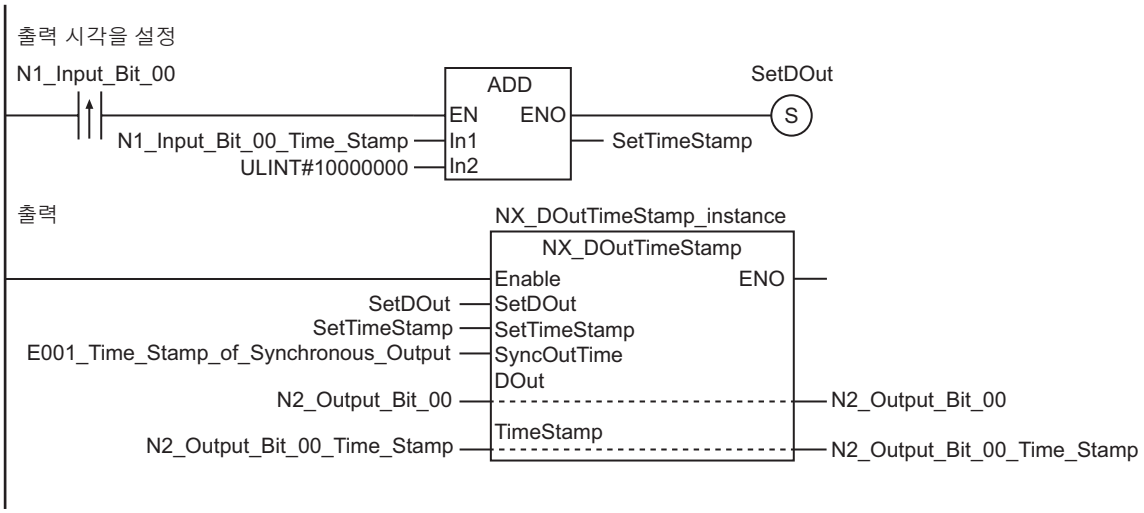
I/O 맵 설정은 다음과 같습니다.

위치	포트	설명	R/W	데이터 형식	변수	변수 종류
Node1	Time Stamp of Synchronous Output	연결된 NX 유니트가 동기 출력한 타이밍의 시각 정보가 저장됩니다. (단위: ns)	R	ULINT	E001_Time_Stamp_of_Synchronous_Output	글로벌 변수
Unit1	Input Bit 00	입력 접점 00	R	BOOL	N1_Input_Bit_00	글로벌 변수
Unit1	Input Bit 00 Time Stamp	입력 접점 00의 변화 시각	R	ULINT	N1_Input_Bit_00_Time_Stamp	글로벌 변수
Unit2	Output Bit 00 Time Stamp	출력 접점 00의 지정 동작 시각	W	ULINT	N2_Output_Bit_00_Time_Stamp	글로벌 변수
Unit2	Output Bit 00	출력 접점 00	W	BOOL	N2_Output_Bit_00	글로벌 변수

LD

내부변수	명칭	데이터 형식	초기값	코멘트
	SetTimeStamp	ULINT	0	지정 타임 스탬프
	SetDOut	BOOL	FALSE	출력값
	NX_DOutTimeStamp _instance	NX_DOutTimeStamp		

외부변수	명칭	데이터 형식	상수	코멘트
	N1_Input_Bit_00	BOOL	┌┐	입력 접점 00
	N1_Input_Bit_00_Time_Stamp	ULINT	┌┐	입력 접점 00의 변화 시각
	E001_Time_Stamp_of_Synchronous _Output	ULINT	┌┐	연결된 NX 유니트가 동기 출력한 타이밍의 시각 정보
	N2_Output_Bit_00	BOOL	┌┐	출력 접점 00
	N2_Output_Bit_00_Time_Stamp	ULINT	┌┐	출력 접점 00의 지정 동작 시각



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	SetEN	BOOL	FALSE	실행 조건
	SetTimeStamp	ULINT	0	지정 타임 스탬프
	SetDOut	BOOL	FALSE	출력값
	R_TRIG_instance	R_TRIG		
	NX_DOutTimeStamp _instance	NX_DOutTimeStamp		

외부 변수	명칭	데이터 형식	상수	코멘트
	N1_Input_Bit_00	BOOL	┌┐	입력 접점 00
	N1_Input_Bit_00_Time_Stamp	ULINT	┌┐	입력 접점 00의 변화 시각
	E001_Time_Stamp_of_Synchronous_Output	ULINT	┌┐	연결된 NX 유니트가 동기 출력한 타이밍의 시각 정보
	N2_Output_Bit_00	BOOL	┌┐	출력 접점 00
	N2_Output_Bit_00_Time_Stamp	ULINT	┌┐	출력 접점 00의 지정 동작 시각

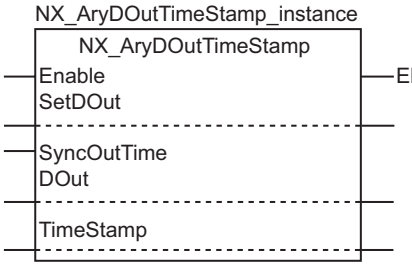
```
// 실행 트리거 입력
R_TRIG_instance( N1_Input_Bit_00, SetEN);

// 출력 시각 설정
IF ( SetEN = TRUE ) THEN
   SetDOut       := TRUE;
   SetTimeStamp:= N1_Input_Bit_00_Time_Stamp + ULINT#10000000;
END_IF;

// 출력
NX_DOutTimeStamp_instance(
  Enable          := TRUE,
  SetDOut         := SetDOut,
  SetTimeStamp  := SetTimeStamp,
  SyncOutTime   := E001_Time_Stamp_of_Synchronous_Output,
  DOut            := N2_Output_Bit_00,
  TimeStamp     := N2_Output_Bit_00_Time_Stamp);
```

NX_AryDOutTimeStamp

타임 스탬프 방식 대응 디지털 출력 유니트에서 펄스를 출력시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
NX_AryDOut TimeStamp	타임 스탬프 디지털 출력 배열 쓰기	FB		NX_AryDOutTimeStamp _instance(Enable, SetDOut, SyncOutTime, DOut, TimeStamp);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Enable	유효	입력	TRUE: 'SetDOut' 설정에 따라 출력을 수행 FALSE: 하강 시 출력을 FALSE로 설정	데이터 형식에 따름	–	FALSE
SyncOut Time	동기 출력 시각 정보		EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Synchronous Output'		ns	(*)
SetDOut	출력 펄스	입출력	출력 펄스	–	–	–
DOut	DOut 유니트 출력 접점		타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수 'Output Bit **'	데이터 형식에 따름		
TimeStamp	타임 스탬프		타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수 'Output Bit ** Time Stamp'		ns	

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
SyncOut Time									○											
SetDOut	구조체 _sOUTPUT_REF 자세한 내용은 기능 설명 참조																			
DOut	○																			
TimeStamp									○											

기능

유효 'Enable'의 값이 TRUE인 경우, 지정한 시각에 타임 스탬프 방식 대응 디지털 출력 유니트에서 출력 펄스 'SetDOut'에 설정한 펄스를 출력합니다.

'Enable'의 값이 TRUE에서 FALSE로 하강했을 때 타임 스탬프 방식 대응 디지털 출력 유니트로의 출력을 FALSE로 합니다.
지정된 시각과 출력 시각의 오차는 최대 μs입니다.

동기 출력 시각 정보 'SyncOutTime'은 타임 스탬프 방식 대응 디지털 출력 유니트가 연결되어 있는 EtherCAT 커플러 유니트의 기준 시각 정보입니다. 연결되어 있는 EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Syncronous Output'을 지정하십시오.
단, EtherCAT 커플러 유니트의 I/O 항목에 0x200A:02(Time Stamp of Synchronous Output)를 추가해야 합니다.

DOut 유니트 출력 접점 'DOut'에는 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점에 할당된 디바이스 변수 'Output Bit **'를 지정하십시오.

타임 스탬프 'TimeStamp'에는 타임 스탬프 방식 대응 디지털 출력 유니트의 출력 접점의 시각 정보에 할당된 디바이스 변수 'Output Bit ** Time Stamp'를 지정하십시오.

출력 시각의 지정

출력 시각은 다음 순서로 지정합니다.

1. **기준 시각으로 사용할 유니트의 접점의 시각 정보에 할당된 디바이스 변수를 취득합니다.**
2. **취득한 시각 정보와 출력 접점을 ON하는 시각의 차를 ns 단위의 수치로 하여 1)에서 취득한 변수에 더합니다.**
3. **더한 결과를 이 명령의 SetDOut.OnTime[]에 전달합니다.**
4. **2)와 마찬가지로, 취득한 시각 정보와 출력 접점을 OFF하는 시각의 차를 ns 단위의 수치로 하여 1)에서 취득한 변수에 더합니다.**
5. **더한 결과를 이 명령의 SetDOut.OffTime[]에 전달합니다.**

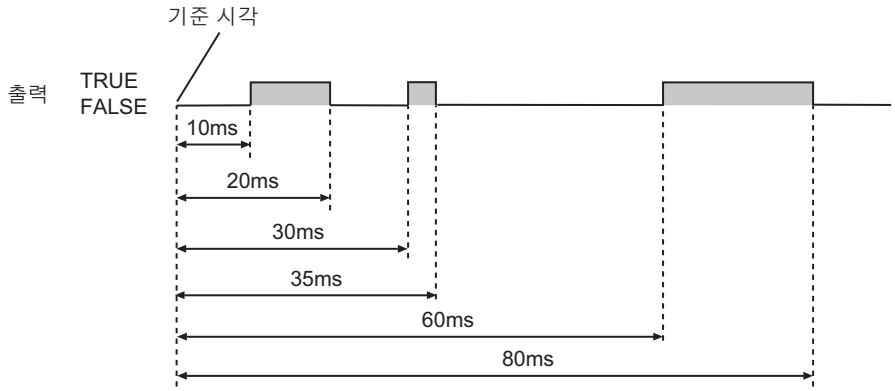
출력 펄스의 지정

출력 펄스 'SetDOut'의 데이터 형식은 구조체 _sOUTPUT_REF입니다. 사양은 다음과 같습니다.

변수	명칭	내용	데이터 형식	유효 범위	단위	초기값
SetDOut	출력 펄스	출력 펄스	_sOUTPUT_REF	–	–	–
EnableOut	출력 유효	출력 유효 플래그 TRUE: 'OnTime', 'OffTime'의 설정이 유효 FALSE: 'OnTime', 'OffTime'의 설정이 무효	BOOL	데이터 형식에 따름	–	FALSE
OnTime[] 배열	ON 시각	출력 접점을 ON하는 시각	ARRAY[0..15] OF ULINT		ns	모든 요소가 0
OffTime[] 배열	OFF 시각	출력 접점을 OFF하는 시각	ARRAY[0..15] OF ULINT			

ON 시각 OnTime[]과 OFF 시각 OffTime[]은 각각 요소 수 16의 배열입니다. 2개 배열의 동일 요소 번호의 값이 1개 펄스의 ON 시각과 OFF 시각을 나타냅니다. 따라서 최대 16개의 펄스를 지정할 수 있습니다. 2개 배열의 동일 요소 번호의 값이 양쪽 모두 0이면 해당 요소 번호 이상의 값은 무효가 됩니다.
예를 들어, OnTime[]과 OffTime[]의 요소의 값이 아래와 같은 경우, 출력은 다음 그림과 같습니다. 다음 표의 값은 지정된 시각을 기준 시각의 몇 ms 후인지로 표현하고 있습니다.

변수	요소 번호				
	0	1	2	3	4
OnTime[]	10ms 후	30ms 후	60ms 후	0	90ms 후
OffTime[]	20ms 후	35ms 후	80ms 후	0	100ms 후



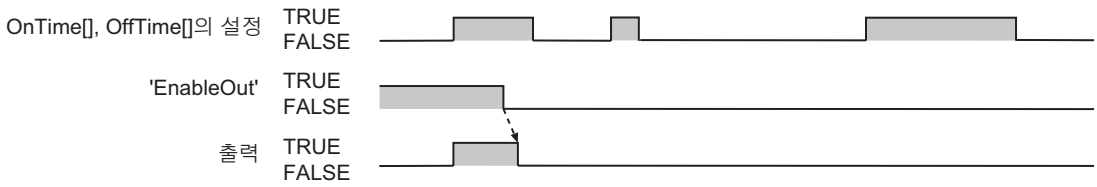
OnTime[]과 OffTime[]의 요소의 값을 시간의 오름차순으로 정렬할 필요는 없습니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우도 출력은 위의 그림과 같습니다.

변수	요소 번호				
	0	1	2	3	4
OnTime[]	30ms 후	60ms 후	10ms 후	0	90ms 후
OffTime[]	35ms 후	80ms 후	20ms 후	0	100ms 후

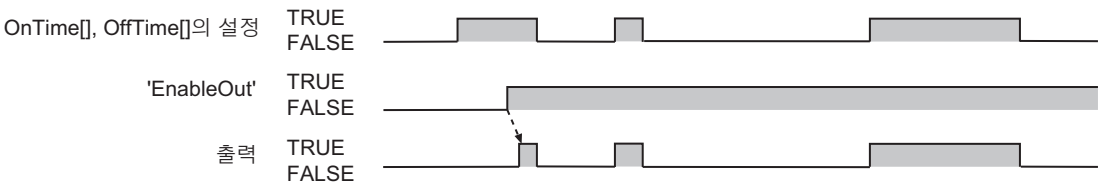
● 출력 유효 'EnableOut'

출력 유효 'EnableOut'은 OnTime[], OffTime[]의 설정을 유효로 하는 플래그입니다. 'EnableOut'의 값이 FALSE인 경우, OnTime[], OffTime[]의 값에 관계 없이 출력값은 FALSE가 됩니다.

'EnableOut'의 값을 이 명령 실행 중에 변경할 수 있습니다.
'EnableOut'의 값을 TRUE에서 FALSE로 변경하면 출력값이 FALSE가 됩니다.

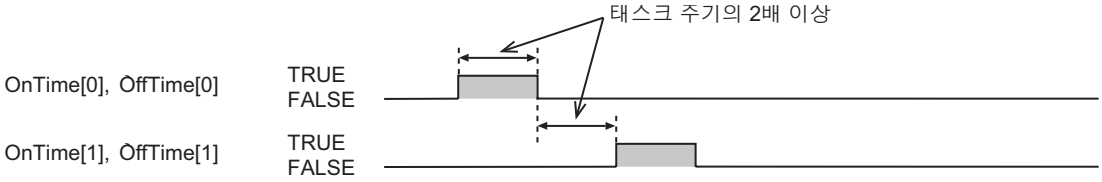


'EnableOut'의 값을 FALSE에서 TRUE로 변경하면 OnTime[], OffTime[]의 값이 유효가 됩니다.



● 출력 펄스의 최소 폭

출력 펄스를 1μs의 시간 정밀도로 출력하기 위해서는 OnTime[]과 OffTime[]의 간격을 태스크 주기의 2배 이상으로 하십시오. 2배 미만의 값을 지정하면 지정한 펄스가 출력되지 않는 경우나 지정한 ON/OFF 시각에서 1 태스크 주기가 지연되어 출력되는 경우가 있습니다.

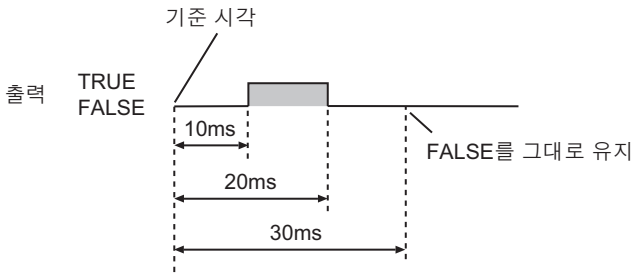


자세한 동작에 대해서는 이후의 설명을 참조하십시오.

● OnTime[]과 OffTime[]의 동일 요소 번호의 값이 같은 경우

OnTime[]과 OffTime[]의 동일 요소 번호의 값이 같은 경우, 출력은 FALSE가 됩니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

변수	요소 번호		
	0	1	2
OnTime[]	10ms 후	30ms 후	0
OffTime[]	20ms 후	30ms 후	0



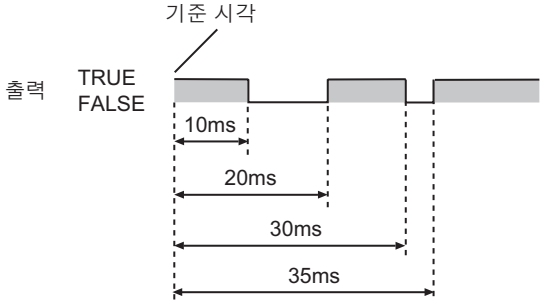
● 동일 요소 번호의 OffTime[]의 요소의 값보다 OnTime[]의 요소의 값이 큰 경우

동일 요소 번호의 OffTime[]의 요소의 값보다 OnTime[]의 요소의 값이 큰 경우, 출력값은 먼저 FALSE가 된 후 TRUE가 됩니다.

또한 OffTime[]의 요소의 최소값보다 OnTime[]의 요소의 최소값이 큰 경우, 이 명령 실행 후 즉시 출력값이 TRUE가 됩니다. 그리고 OffTime[]의 요소의 최대값보다 OnTime[]의 요소의 최대값이 큰 경우, 이 명령 실행 종료 후 출력값은 TRUE가 유지됩니다.

따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

변수	요소 번호		
	0	1	2
OnTime[]	20ms 후	35ms 후	0
OffTime[]	10ms 후	30ms 후	0

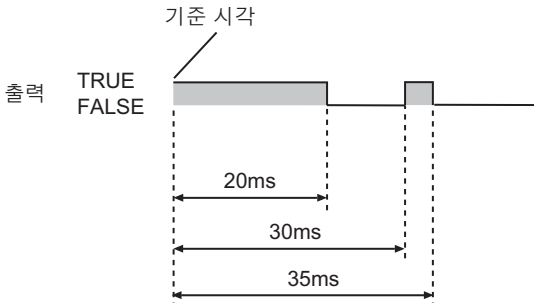


● OnTime[]과 OffTime[] 중 한 요소의 값이 0인 경우

OnTime[]과 OffTime[] 중 한 요소가 0인 경우, 이 명령 실행 후 즉시 출력이 TRUE 또는 FALSE로 되는 것을 의미합니다.

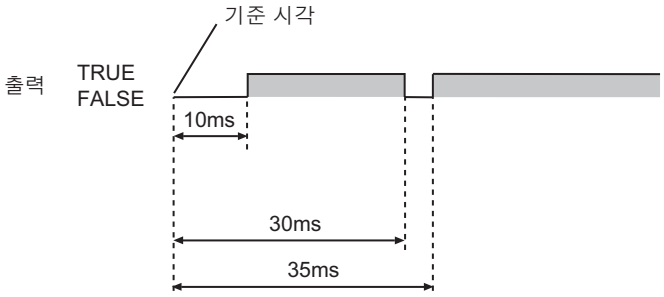
OnTime[]의 요소의 값만 0인 경우, 이 명령 실행 후 출력값이 즉시 TRUE가 됩니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

변수	요소 번호		
	0	1	2
OnTime[]	0	30ms 후	0
OffTime[]	20ms 후	35ms 후	0



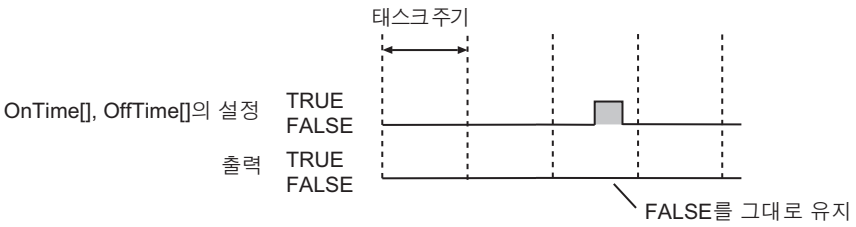
OffTime[]의 요소의 값만 0인 경우, 이 명령 실행 후 출력값이 즉시 FALSE가 됩니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

변수	요소 번호		
	0	1	2
OnTime[]	10ms 후	35ms 후	0
OffTime[]	0	30ms 후	0



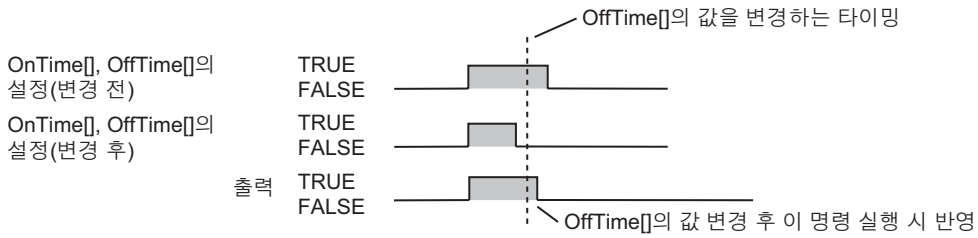
● 동일 태스크 주기 내에서 출력값을 TRUE로 하는 설정과 FALSE로 하는 설정이 연속된 경우

동일 태스크 주기 내에서 출력을 TRUE로 하는 설정과 FALSE로 하는 설정이 연속된 경우, 출력값은 변화하지 않습니다.



● 이 명령이 유효일 때 OnTime[]이나 OffTime[]의 값을 변경한 경우

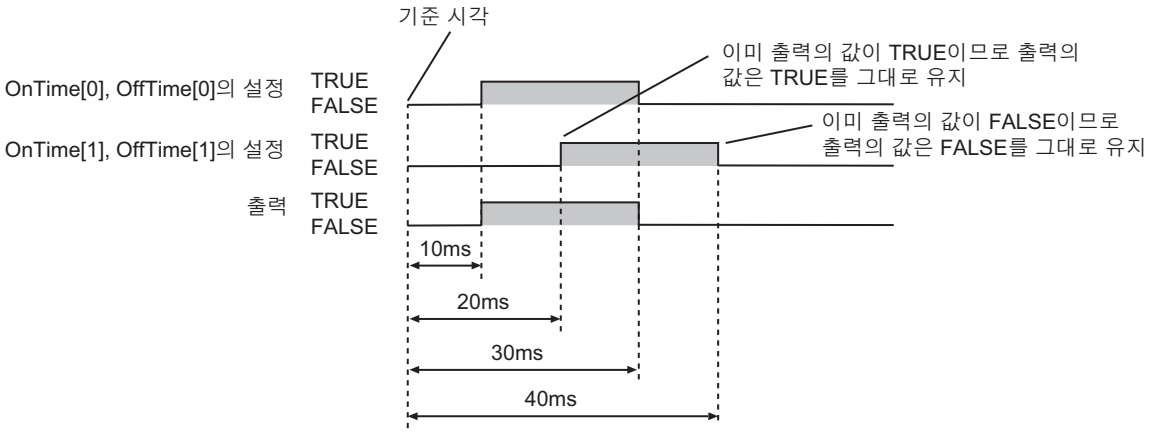
이 명령이 유효일 때 OnTime[]이나 OffTime[]의 값을 변경할 수 있습니다. 변경 내용은 변경 후 이 명령 실행 시부터 반영됩니다.



● 출력값을 TRUE로 하는 설정이 중복된 경우

출력값을 TRUE로 하는 설정이 중복된 경우, 이상이 되지는 않으며 출력값은 TRUE를 그대로 유지합니다. 출력값을 FALSE로 하는 설정이 중복된 경우도 동일합니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

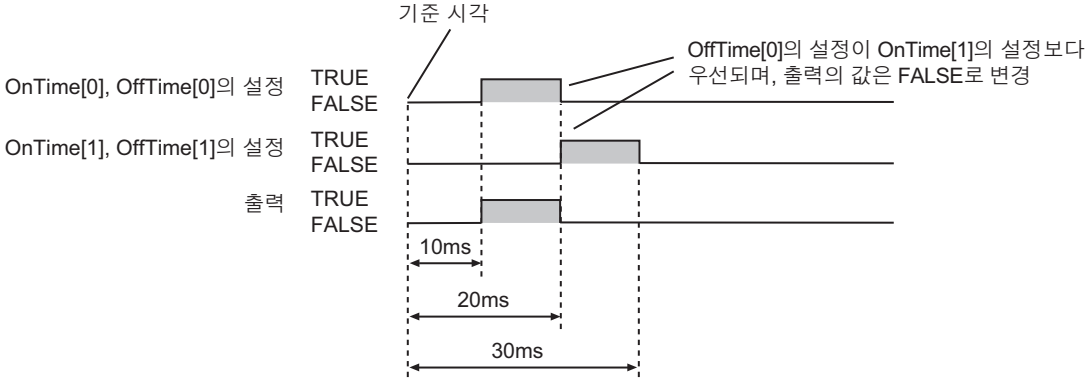
변수	요소 번호		
	0	1	2
OnTime[]	10ms 후	20ms 후	0
OffTime[]	30ms 후	40ms 후	0



● 출력값을 TRUE로 하는 설정과 FALSE로 하는 설정이 동일한 시각에 중복된 경우

출력값을 TRUE로 하는 설정과 FALSE로 하는 설정이 동일한 시각에 중복된 경우, 이상이 되지는 않으며 OnTime[]과 OffTime[]의 요소 번호가 작은 쪽 설정이 우선됩니다. 따라서 2개 배열의 요소의 값이 다음과 같은 경우, 출력은 아래 그림과 같습니다.

변수	요소 번호		
	0	1	2
OnTime[]	10ms 후	20ms 후	0
OffTime[]	20ms 후	30ms 후	0



참고

이 명령은 MC_DigitalCamSwitch 명령을 사용할 때 사용합니다. MC_DigitalCamSwitch 명령에 대한 자세한 내용은 'NJ 시리즈 명령 참조 설명서 모션편(SBCE-364G 이상)'을 참조하십시오.

사용상의 주의 사항

- 이 명령은 타임 스탬프 방식 대응 디지털 출력 유니트에 대해서만 실행할 수 있습니다. 단, 타임 스탬프 방식 대응 디지털 출력 유니트를 연결하지 않은 상태에서 이 명령을 실행해도 이상 상태가 되지는 않습니다.
- EtherCAT 통신 이상 또는 태스크 주기 초과가 발생한 경우는 지정된 시각에 출력할 수 없는 경우가 있습니다. 그 경우, 다음 태스크 주기 이후에 출력됩니다.
- 이 명령에서 사용하는 디바이스 변수를 다른 명령이나 프로그램에서 변경하거나 참조하는 경우는 배타적으로 처리하십시오.
- 'SyncOutTime'에는 타임 스탬프 방식 대응 디지털 출력 유니트가 직접 연결되어 있는 EtherCAT 커플러 유니트의 디바이스 변수 'Time Stamp of Synchronous Output'을 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.
- 'DOut'과 'TimeStamp'에는 출력할 타임 스탬프 방식 대응 디지털 출력 유니트의 디바이스 변수를 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.
- 'DOut'과 'TimeStamp'에는 동일한 유니트의 동일한 채널 번호의 디바이스 변수를 지정하십시오. 단, 그 이외의 변수를 지정해도 이상 상태가 되지는 않습니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상, Sysmac Studio Ver.1.07 이상에서 사용할 수 있습니다.

샘플 프로그램

샘플 프로그램은 'NJ 시리즈 명령 참조 설명서 모션편(SBCE-364G 이상)'에서 MC_DigitalCamSwitch 명령을 참조하십시오.

기타 명령

명령	명칭	페이지
ReadNbit_**	N비트 읽기 그룹	2-1132
WriteNbit_**	N비트 쓰기 그룹	2-1134
ChkRange	범위 형식 변수 확인	2-1136
GetMyTaskStatus	로컬 태스크 Status 읽기	2-1138
Task_IsActive	태스크 실행 중 판정	2-1141
Lock/Unlock	태스크 간 배타적 잠금/ 태스크 간 배타적 잠금 해제	2-1143
ActEventTask	이벤트 태스크 기동	2-1143
Get**Clk	클록 펄스 취득 그룹	2-1154
Get**Cnt	가산 프리 러닝 카운터 취득 그룹	2-1155

ReadNbit_**

비트 열 내의 복수 비트를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ReadNbit_**	N비트 읽기 그룹	FUN	(@)ReadNbit_** EN ENO In Out Pos Size 단, **는 비트 열의 데이터 형식명	Out:=ReadNbit_**(In, Pos, Size); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	읽기 원본	입력	읽기 원본의 비트 열	데이터 형식에 따름	–	0
Pos	읽기 위치		읽는 비트 위치	0~'In'의 비트 수-1		
Size	읽기 크기		읽는 비트 수	0~'In'의 비트 수		1
Out	읽기 결과	출력	읽기 결과	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Pos						○														
Size						○														
Out	'In'과 같은 데이터 형식																			

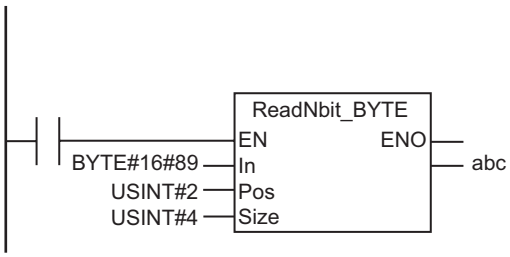
기능

읽기 원본 비트 열 'In'의 읽기 위치 'Pos'를 포함하는 상위 'Size' 비트의 값을 읽기 결과 'Out'에 대입합니다.

명령 이름은 'In' 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In' 'Out'이 WORD 형식인 경우, 명령 이름은 ReadNbit_WORD가 됩니다.

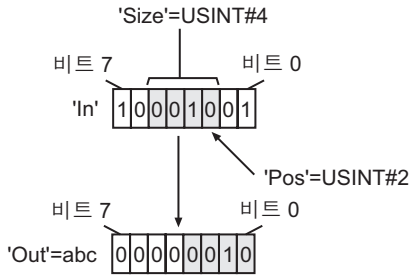
ReadNbit_BYTE 명령에서 'In'=BYTE#16#89, 'Pos'=USINT#2, 'Size'=USINT#4인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=ReadNbit_BYTE(BYTE#16#89, USINT#2,
                   USINT#4);
```



참고

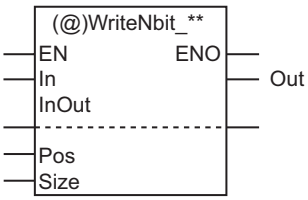
비트 열에 복수 비트를 쓰는 경우, WriteNbit_** 명령을 사용하십시오.

사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식을 동일하게 하십시오.
- 'Size'의 값이 0인 경우, 'Out'의 값은 16#0이 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'Out'은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Pos'의 값이 유효 범위 밖인 경우
  - 'In'이 'Pos'의 값으로 지정한 위치부터 'Size'의 값으로 나타낸 수만큼의 비트 열을 가지고 있지 않은 경우

WriteNbit_**

비트 열 내에 복수 비트를 써넣습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
WriteNbit_**	N비트 쓰기 그룹	FUN	 단, **는 비트 열의 데이터 형식명	WriteNbit_**(In, InOut, Pos, Size); 단, **는 비트 열의 데이터 형식명

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	읽기 원본	입력	'InOut'에 기록할 값을 읽는 비트 열	데이터 형식에 따름	–	0
Pos	쓰기 위치		쓰는 비트 위치	0~'InOut'의 비트 수-1		
Size	쓰기 크기		쓰는 비트 수	0~'In'의 비트 수		1
InOut	쓰기 대상	입출력	읽기 결과	데이터 형식에 따름	–	–
Out	반환값	출력	항상 TRUE	TRUE만	–	–

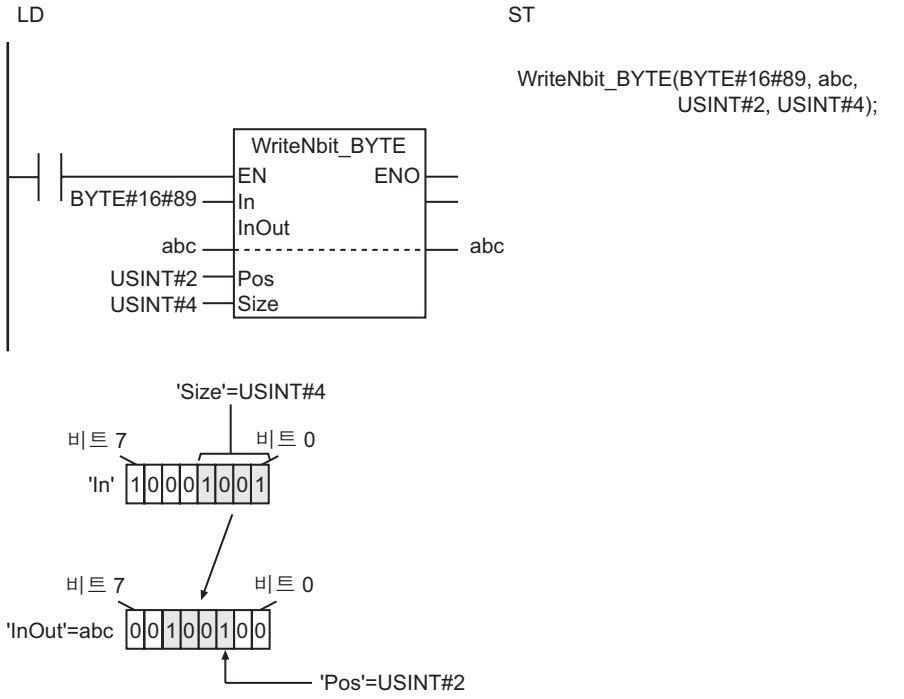
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○															
Pos						○														
Size						○														
InOut	'In'과 같은 데이터 형식																			
Out	○																			

기능

우선 읽기 원본 'In'의 하위 'Size' 비트를 읽습니다. 그 다음, 읽은 값을 쓰기 대상 'InOut'의 쓰기 위치 'Pos'에 덮어씁니다.

명령 이름은 'In' 'Out'의 데이터 형식에 따라 다릅니다. 예를 들면, 'In' 'Out'이 WORD 형식인 경우, 명령 이름은 WriteNbit_WORD가 됩니다.

WriteNbit_BYTE 명령에서 'In'=BYTE#16#89, 'Pos'=USINT#2, 'Size'=USINT#4인 경우의 예가 아래에 나와 있습니다.



참고

비트 열의 복수 비트를 읽는 경우, ReadNbit_** 명령을 사용하십시오.

사용상의 주의 사항

- 'In'과 'Out'의 데이터 형식을 동일하게 하십시오.
- 'Size'의 값이 0인 경우, 'InOut'의 값은 변화하지 않습니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.
- 다음과 같은 경우, 이상 상태가 됩니다. ENO는 FALSE가 되고 'InOut'은 변화하지 않습니다.
  - 'Size'의 값이 유효 범위 밖인 경우
  - 'Pos'의 값이 유효 범위 밖인 경우
  - 'InOut'이 'Pos'의 값으로 지정한 위치부터 'Size'의 값으로 나타낸 수만큼의 비트 열을 가지고 있지 않은 경우

ChkRange

변수의 값이 범위 형식 지정의 유효 범위 내에 있는지 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ChkRange	범위 형식 변수 확인	FUN	(@)ChkRange EN In Val Out	Out:=ChkRange(In, Val);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
In	확인 대상	입력	확인 대상	데이터 형식에 따름	–	(*)
Val	범위 지정 변수		범위 지정 변수	범위 지정에 따름		
Out	확인 결과	출력	확인 결과	데이터 형식에 따름	–	–

* 입력 파라메터가 생략된 경우, 초기값은 적용되지 않습니다. 빌드 시에 이상 상태가 됩니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						○	○	○	○	○	○	○	○							
Val	범위 지정 원본이 되는 기본 데이터 형식은 'In'과 같음																			
Out	○																			

기능

확인 대상 'In'의 값이 범위 지정 변수 'Val'의 유효 범위 내에 있는지 판정합니다. 유효 범위 내에 있으면 확인 결과 'Out'은 TRUE, 없으면 FALSE가 됩니다.

참고

범위 지정은, 정수 형식(USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT)의 변수에 대해 정의할 수 있습니다.

사용상의 주의 사항

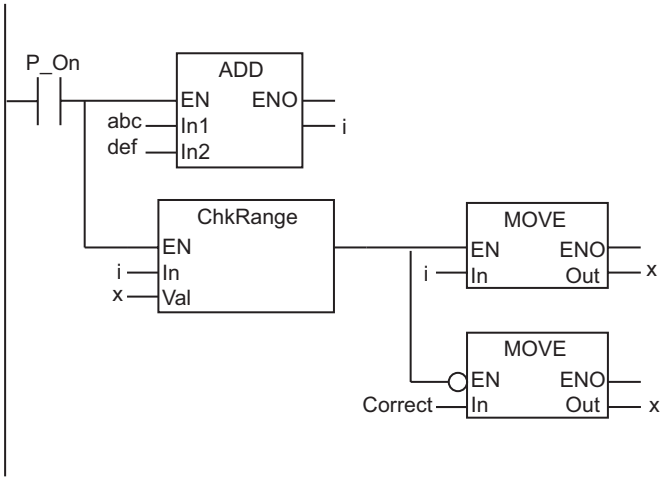
- 'In'이 범위 지정 변수가 아닌 경우, 'Out'의 값은 TRUE가 됩니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

샘플 프로그램

가산 결과 i가 범위 지정 변수 x의 유효 범위 내(10~99)에 있는지 판정합니다. 만일 유효 범위 내에 있으면 i의 값을 x에 대입합니다. 만일 유효 범위 내에 없으면 변수 Correct의 값을 x에 대입합니다.

LD

명칭	데이터 형식	초기값
i	INT	0
abc	INT	0
def	INT	0
x	INT(10..99)	10
Correct	INT	0



ST

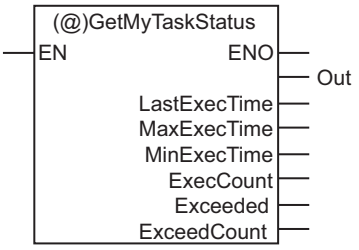
명칭	데이터 형식	초기값
i	INT	0
abc	INT	0
def	INT	0
Chk	BOOL	FALSE
x	INT(10..99)	10
Correct	INT	0

```
i := abc+def;
Chk:=ChkRange(i, x);  // 범위형 변수 체크

IF (Chk=TRUE) THEN
    x := i;                 // i의 값이 범위 안이라면 i를 x에 대입
ELSE
    x := Correct;           // i의 값이 범위 안이 아니라면 Correct를 x에 대입
END_IF;
```

GetMyTaskStatus

로컬 태스크의 태스크 Status를 읽어옵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
GetMyTask Status	로컬 태스크 Status 읽기	FUN		GetMyTaskStatus(LastExecTime, MaxExecTime, MinExecTime, ExecCount, Exceeded, ExceedCount);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	반환값	출력	항상 TRUE	TRUE만	–	–
LastExec Time	이전 태스크 실행 시간		이전의 로컬 태스크 실행 시간	데이터 형식에 따름 (*)	ns	
MaxExec Time	태스크 실행 시간 최대값		로컬 태스크 실행 시간의 최대값			
MinExec Time	태스크 실행 시간 최소값		로컬 태스크 실행 시간의 최소값			
ExecCount	태스크 실행 횟수		로컬 태스크 실행 횟수	데이터 형식에 따름	–	
Exceeded	태스크 주기 초과 플래그		TRUE : 이전의 로컬 태스크 실행이 태스크 주기 내에 완료되지 않았음 FALSE: 이전의 로컬 태스크 실행이 태스크 주기 내에 완료되었음			
Exceed Count	태스크 주기 초과 횟수		로컬 태스크가 태스크 주기를 초과한 횟수			

* 음수는 포함되지 않습니다.

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out						○														
LastExec Time																○				
MaxExec Time																○				
MinExec Time																○				
ExecCount								○												
Exceeded	○																			

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Exceed Count								○												

기능

로컬 태스크의 태스크 상태를 취득합니다.
태스크 상태란, 이전 태스크 실행 시간 'LastExecTime', 태스크 실행 시간 최대값 'MaxExecTime', 태스크 실행 시간 최소값 'MinExecTime', 태스크 실행 횟수 'ExecCount', 태스크 주기 초과 플래그 'Exceeded', 태스크 주기 초과 횟수 'ExceedCount'입니다.

참고

'MaxExecTime', 'MinExecTime', 'ExecCount', 'ExceedCount'의 값이 리셋되는 것은 다음과 같은 타이밍입니다.

- 운전 시작 시
- Sysmac Studio의 태스크 실행 시간 모니터링 화면에서 리셋 실행 시

사용상의 주의 사항

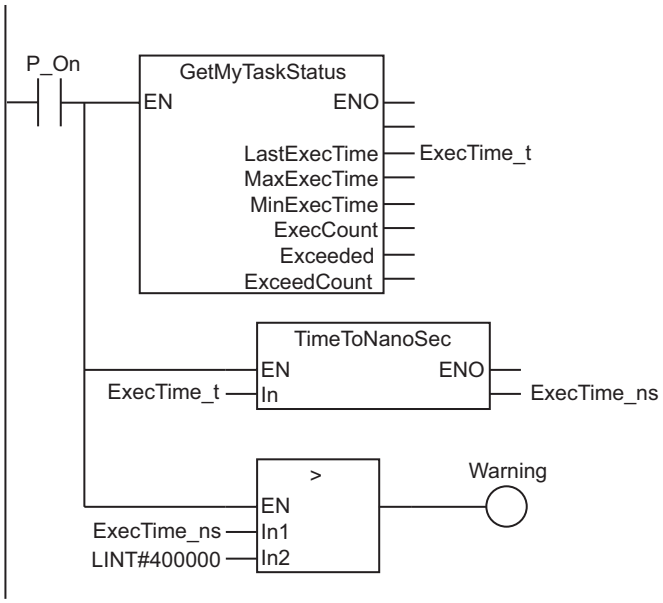
- 'ExecCount' 및 'ExceedCount'의 값은 UDINT 형식의 최대값(4,294,967,295)을 초과하면 0으로 되돌아갑니다.
- 이 명령을 ST 프로그램에서 사용하는 경우, 반환값 'Out'은 사용하지 않습니다.

샘플 프로그램

로컬 태스크의 태스크 상태를 취득합니다. 만일 이전 태스크 실행 시간이 400us(400000ns)를 초과하면, 변수 Warning의 값을 TRUE로 합니다.

LD

명칭	데이터 형식	초기값	코멘트
ExecTime_t	TIME	T#0s	이전 태스크 실행 시간(TIME 형식)
ExecTime_ns	LINT	0	이전 태스크 실행 시간(LINT 형식 나노초 수)
Warning	BOOL	FALSE	경고



ST

명칭	데이터 형식	초기값	코멘트
ExecTime_t	TIME	T#0s	이전 태스크 실행 시간(TIME 형식)
ExecTime_ns	LINT	0	이전 태스크 실행 시간(LINT 형식 나노초 수)
Warning	BOOL	FALSE	경고

```
GetMyTaskStatus(LastExecTime=>ExecTime_t);  // 전회 태스크 주기 취득
ExecTime_ns:=TimeToNanoSec(ExecTime_t);     // 전회 태스크 주기를 TIME 형식에서 나노초 수로 변환
IF (ExecTime_ns>DINT#400000) THEN           // 전회 태스크 주기가 400,000ns를 초과하는 경우,
    Warning:=TRUE;                          //   변수 Warning에 TRUE를 대입
ELSE
    Warning:=FALSE;
END_IF;
```

Task_IsActive

지정된 태스크가 실행 중인지 판정합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
Task_IsActive	태스크 실행 중 판정	FUN	(@)Task_IsActive EN TaskName Out	Out:=Task_IsActive(TaskName);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
TaskName	태스크명	입력	태스크명	최대 64바이트 (반각 영숫자 63문자 +끝의 NULL 문자)	–	''
Out	실행 중 판정	출력	TRUE : 실행 중 또는 대기 상태 FALSE : 실행 중이 아님	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
TaskName																				○
Out	○																			

기능

태스크명 'TaskName'으로 지정한 태스크가 실행 중 또는 대기 상태인지 판정합니다. 대기 상태란, 태스크가 기동한 후, 그보다 우선 순위가 높은 태스크가 기동되었기 때문에 처리가 중단된 상태를 말합니다. 실행 중 또는 대기 상태라면, 실행 중 판정 'Out'의 값이 TRUE가 됩니다. 비실행 중이라면 'Out'의 값이 FALSE가 됩니다.

사용상의 주의 사항

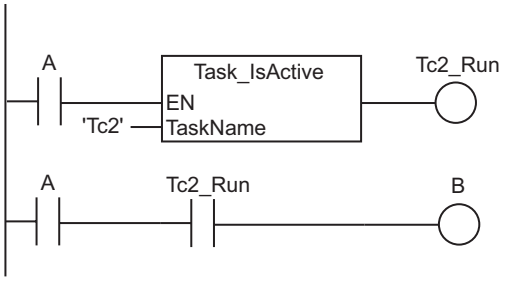
- 'TaskName'에는 문자열을 대입한 변수는 지정할 수 없습니다. 문자열을 직접 지정하십시오.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.
- 다음과 같은 경우, 이상 상태가 됩니다. 'Out'은 변화하지 않습니다.
  - 'TaskName'으로 지정한 태스크가 존재하지 않는 경우

샘플 프로그램

변수 A의 값이 TRUE가 되면, 주기 태스크 Tc2가 기동 중인지 조사합니다. 기동 중이라면 변수 B의 값을 TRUE로 합니다.

LD

명칭	데이터 형식	초기값	코멘트
A	BOOL	FALSE	
B	BOOL	FALSE	
Tc2_Run	BOOL	FALSE	태스크 Tc2의 실행 상태



ST

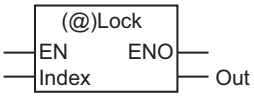
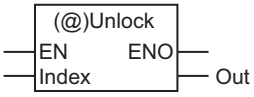
명칭	데이터 형식	초기값	코멘트
A	BOOL	FALSE	
B	BOOL	FALSE	
Tc2_Run	BOOL	FALSE	태스크 Tc2의 실행 상태

```
IF (A=TRUE) THEN
    // 태스크 실행 중 판정
    Tc2_Run:=Task_IsActive('Tc2');
    // Tc2가 실행 중이라면 변수 B에 TRUE를 대입
    IF (Tc2_Run=TRUE) THEN
        B := TRUE;
    END_IF;

END_IF;
```

Lock / Unlock

Lock : 태스크 간 배타적 잠금을 시작합니다. 다른 태스크의 동일한 잠금 번호 구간은 실행할 수 없게 됩니다.
Unlock : 태스크 간 배타적 잠금을 종료합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Lock	태스크 간 배타적 잠금	FUN		Lock(Index);
Unlock	태스크 간 배타적 잠금 해제	FUN		Unlock(Index);

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Index	잠금 번호	입력	잠금 번호	데이터 형식에 따름	–	0
Out	반환값	출력	항상 TRUE	TRUE만	–	–

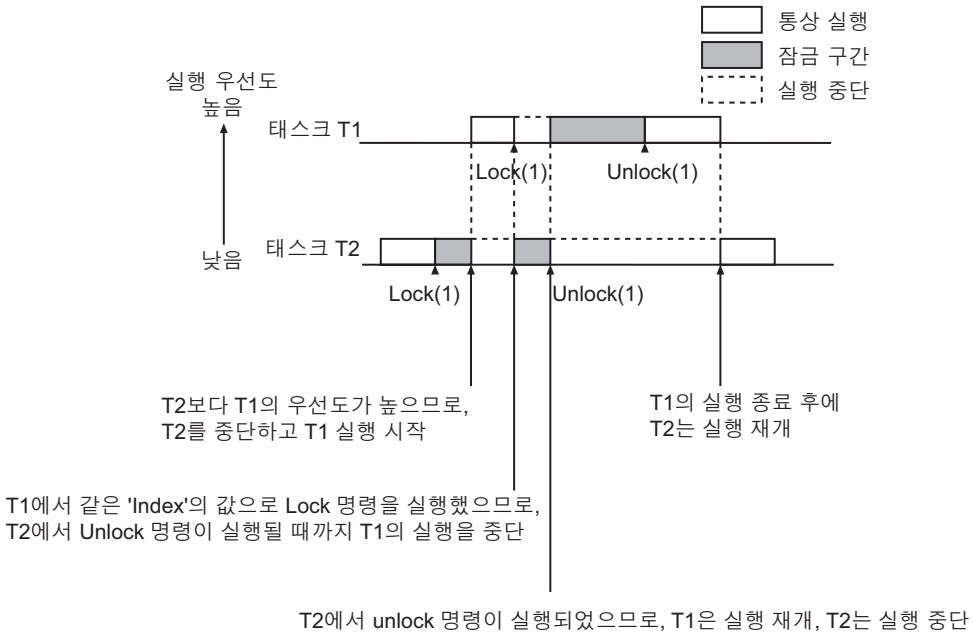
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Index						○														
Out	○																			

기능

Lock 명령에서부터 Unlock 명령까지의 구간(잠금 구간)을 실행 중에 다른 태스크의 같은 잠금 번호의 잠금 구간을 실행하지 못하게 합니다.
잠금 번호는 'Index'로 지정합니다.

동작 예가 아래에 나와 있습니다.

태스크 T1, T2는 모두 'Index'=1의 잠금 구간을 지닌 것으로 합니다. 먼저 T2의 잠금 명령이 실행되면, T2에서 Unlock 명령이 실행될 때까지 T1의 잠금 구간은 실행할 수 없습니다.



'Index'의 값이 다른 잠금 구간은 서로 영향을 주지 않습니다.

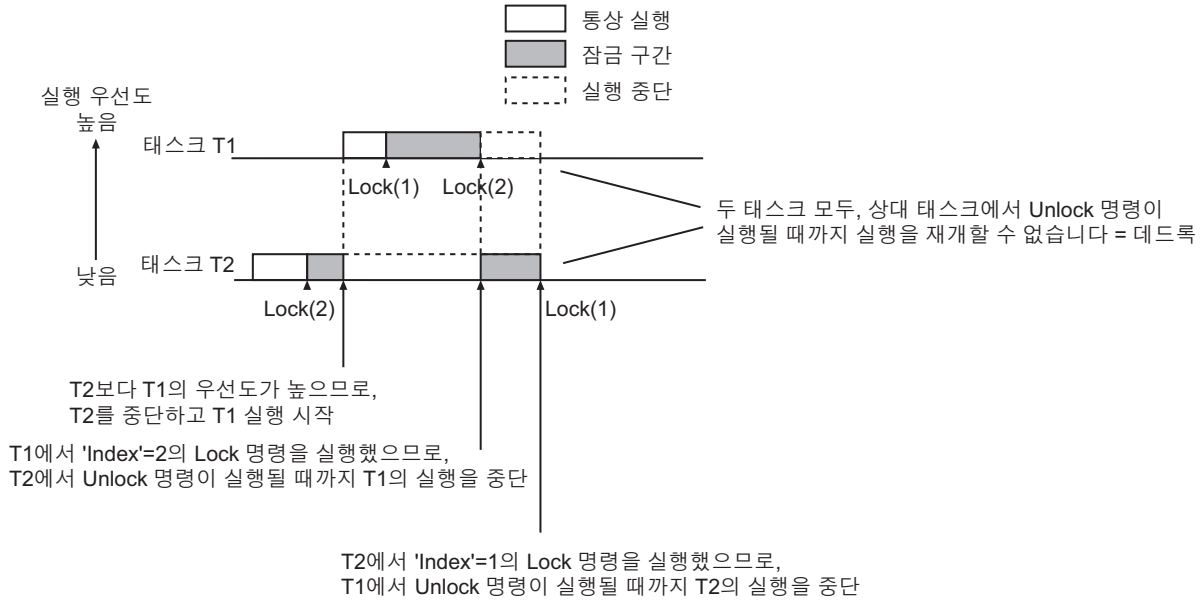
참고

- 이 명령은 복수의 태스크로부터 같은 데이터를 읽을 가능성이 있는 경우 등에 사용합니다. 어떤 태스크가 데이터를 읽고 있는 동안은, 다른 태스크로부터 그 데이터를 읽을 수 없도록 합니다.
- 'Index'의 값이 다르다면, 복수의 Lock 명령과 Unlock 명령의 세트가 동일한 POU에 존재하더라도 상관없습니다. 또한, 그 명령들의 배치가 중첩 구조라도 상관없습니다.

사용상의 주의 사항

- 잠금 구간은 필요 최소한의 길이로 하십시오. 잠금 구간이 길면 태스크의 실행 주기를 초과할 가능성이 있습니다.
- Lock 명령과 Unlock 명령은 동일한 POU의 동일 섹션 내에서 반드시 짝을 이루어 사용하십시오.
- 동시에 설정할 수 있는 잠금 구간은 최대 16777215개입니다.

- 복수의 태스크에서 복수의 Lock 명령을 사용하는 경우, 그 명령들의 배치에 따라서 데드록이 되는 경우가 있습니다. 데드록이 된 경우, 태스크 실행 타임아웃 이상이 발생해서 전체 정지됩니다. 데드록이 되는 경우의 예가 아래에 나와 있습니다.



- 다음과 같은 경우, 이상 상태가 됩니다. 'Out'은 변화하지 않습니다.
  - 잠금 구간을, 동시에 16777215개가 넘게 설정하려고 한 경우

샘플 프로그램

태스크 T1의 프로그램 P1과 태스크 T2의 프로그램 P2에서 동일한 글로벌 변수 GTable1에 액세스합니다.
쓰기 요구 WriteReq의 값이 TRUE가 되면, P1은 레코드 배열 GTable1.Record[]에 1레코드의 쓰기를 수행하고, 인덱스 GTable1.Index를 증가시킵니다.
읽기 요구 ReadReq의 값이 TRUE가 되면, P2는 GTable1.Index를 감소시키고, GTable1.Record[]로부터 1레코드의 읽기를 수행합니다.
Lock 명령을 사용해서 읽기와 쓰기가 동시에 발생하지 않도록 합니다.



글로벌 변수 GTable의 정의

데이터 형식

명칭	데이터 형식	코멘트
USERTABLE	STRUCT	레코드 저장 구조체
Index	INT	인덱스
Record	ARRAY[0..99] OF LREAL	레코드 배열

글로벌 변수

명칭	데이터 형식	초기값	코멘트
GTable1	USERTABLE	(Index:=0, Record:=[100(0.0)])	레코드 저장 구조체

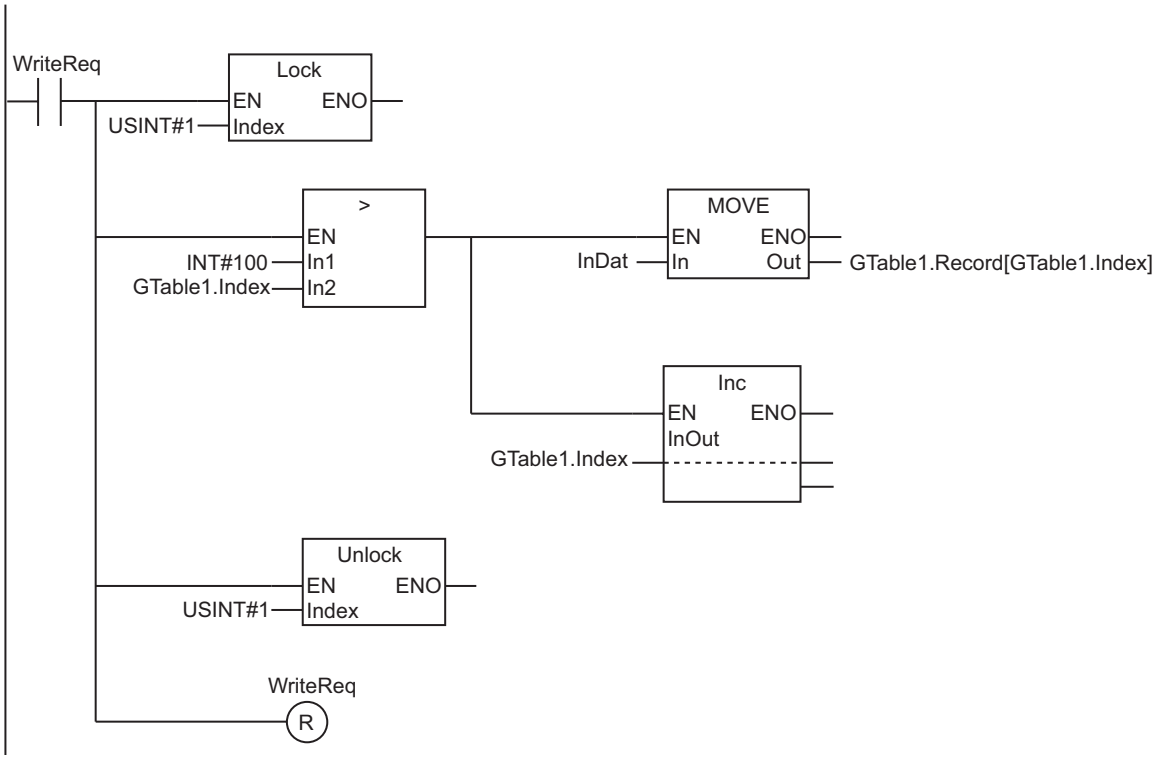
프로그램 P1

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	WriteReq	BOOL	FALSE	쓰기 요구
	InDat	LREAL	0.0	쓰기 데이터

외부 변수	명칭	데이터 형식	코멘트
	GTable1	USERTABLE	레코드 저장 구조체

프로그램 P1



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	WriteReq	BOOL	FALSE	쓰기 요구
	InDat	LREAL	0.0	쓰기 데이터

외부 변수	명칭	데이터 형식	코멘트
	GTable1	USERTABLE	레코드 저장 구조체

```
// 쓰기 요구 검출
IF (WriteReq=TRUE) THEN

    // Lock 명령 실행
    Lock(USINT#1);

    IF (INT#100>GTable1.Index) THEN
        GTable1.Record[GTable1.Index]:=InDat;
        GTable1.Index                 :=GTable1.Index+INT#1;
    END_IF;

    // Unlock 명령 실행
    Unlock(USINT#1);
    WriteReq:=FALSE;

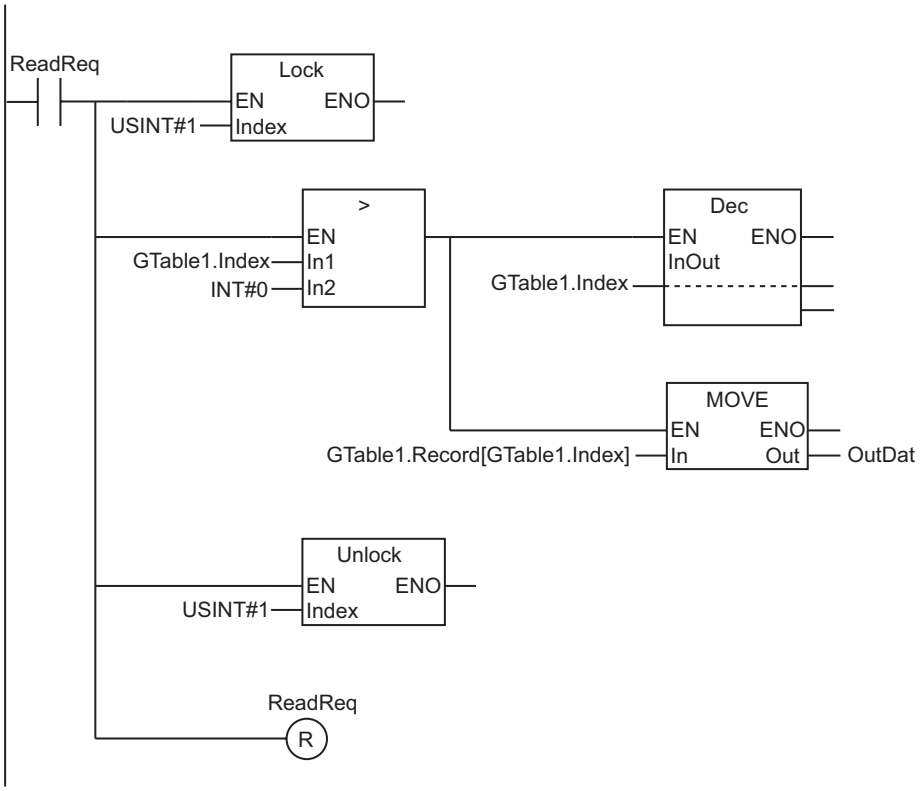
END_IF;
```

프로그램 P2

LD

내부 변수	명칭	데이터 형식	초기값	코멘트
	ReadReq	BOOL	FALSE	읽기 요구
	OutDat	LREAL	0.0	읽기 데이터

외부 변수	명칭	데이터 형식	코멘트
	GTable1	USERTABLE	레코드 저장 구조체



ST

내부 변수	명칭	데이터 형식	초기값	코멘트
	ReadReq	BOOL	FALSE	읽기 요구
	OutDat	LREAL	0.0	읽기 데이터

외부 변수	명칭	데이터 형식	코멘트
	GTable1	USERTABLE	레코드 저장 구조체

```
// 읽기 요구 검출
IF (ReadReq=TRUE) THEN

    // Lock 명령 실행
    Lock(USINT#1);

    IF (GTable1.Index>INT#0) THEN
        GTable1.Index:=GTable1.Index-INT#1;
        OutDat          :=GTable1.Record[GTable1.Index];
    END_IF;

    // Unlock 명령 실행
    Unlock(USINT#1);
    ReadReq:=FALSE;

END_IF;
```

ActEventTask

이벤트 태스크를 기동합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
ActEventTask	이벤트 태스크 기동	FUN	(@)ActEventTask EN ENO TaskName Out	ActEventTask(TaskName);

변수

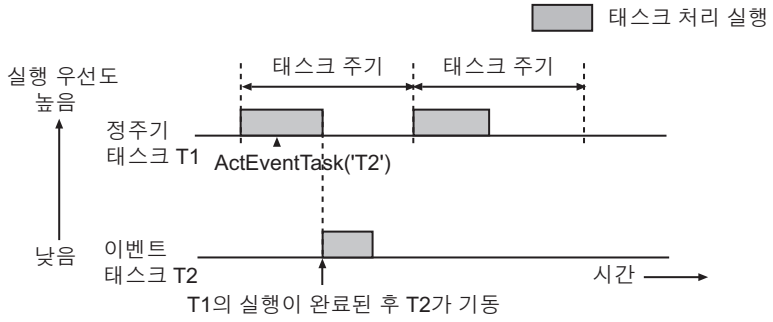
	명칭	입/출력	내용	유효 범위	단위	초기값
TaskName	태스크명	입력	기동할 이벤트 태스크의 태스크명	최대 64바이트 (반각 영숫자 63문자+끝의 NULL 문자)	–	"
Out	반환값	출력	TRUE: 명령을 실행하며 이상 상태가 아님 FALSE: 명령을 실행하지 않거나 이상 상태임	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
TaskName																				○
Out	○																			

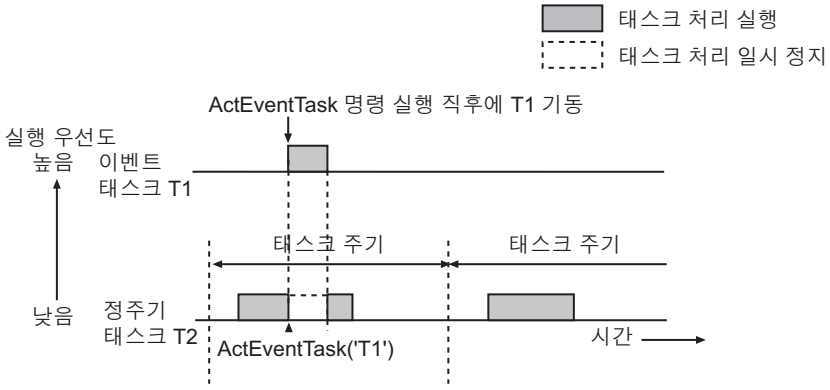
기능

태스크명 'TaskName'인 이벤트 태스크를 1회 기동합니다. 이벤트 태스크는 태스크의 실행 우선도에 따라 동작합니다.

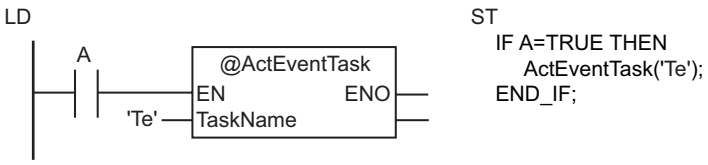
이 명령을 실행하는 태스크보다 실행 우선도가 낮은 이벤트 태스크를 기동하는 경우, 이 명령을 실행하는 태스크의 실행이 완료된 후 이벤트 태스크를 기동합니다. 예를 들어, 정주기 태스크 T1보다 이벤트 태스크 T2의 실행 우선도가 낮다고 가정합니다. T1에서 T2를 지정하여 ActEventTask 명령을 실행하면 T1의 실행이 완료된 후 T2를 기동합니다.



이 명령을 실행하는 태스크보다 실행 우선도가 높은 이벤트 태스크를 기동하는 경우, 이 명령을 실행하는 태스크의 실행을 일시 정지하고 이벤트 태스크를 기동합니다. 예를 들어, 이벤트 태스크 T1보다 정주기 태스크 T2의 실행 우선도가 낮다고 가정합니다. T2에서 T1을 지정하여 ActEventTask 명령을 실행하면 T2 실행을 일시 정지하고 T1을 기동합니다.

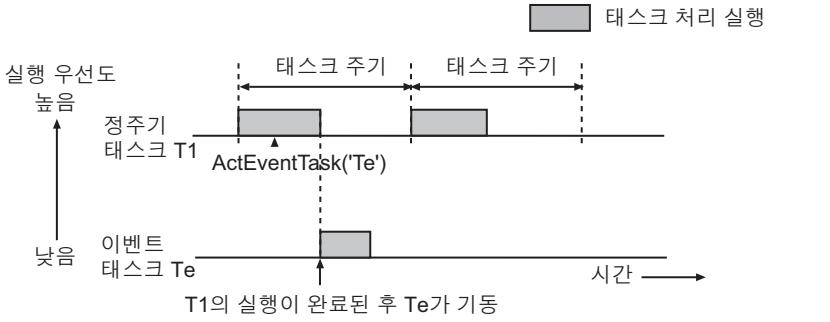


기술 예가 아래에 나와 있습니다. 변수 A의 값이 TRUE인 경우에 이벤트 태스크 'Te'를 기동합니다.




```
ST
  IF A=TRUE THEN
    ActEventTask('Te');
  END_IF;
```

이러한 기술 내용이 있는 프로그램이 정주기 태스크 T1에 할당되어 있고 T1보다 Te의 실행 우선도가 낮다고 가정합니다. 이 경우, T1에서 이 명령이 실행되면 T1의 실행이 완료된 후 Te를 기동합니다.



관련된 시스템 정의 변수

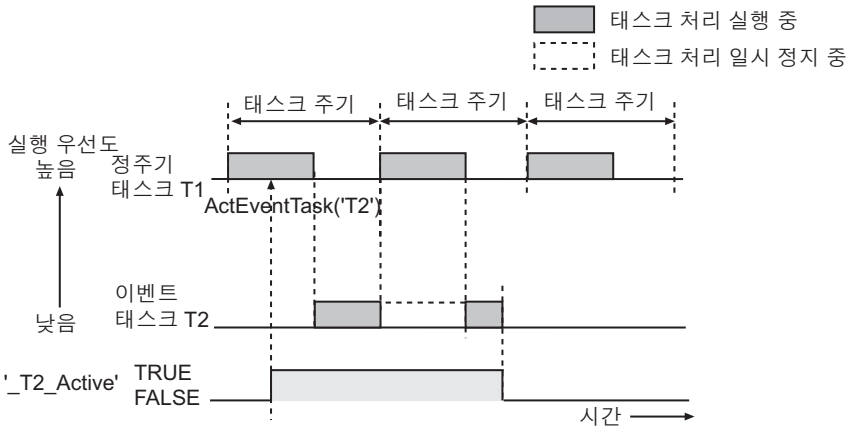
변수명	이름	데이터 형식	내용
_**_Active*1	태스크 실행 중 플래그	BOOL	태스크 실행 상태.*2 TRUE: 실행 중 FALSE: 정지 중

*1. **는 태스크명입니다.
*2. 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오.

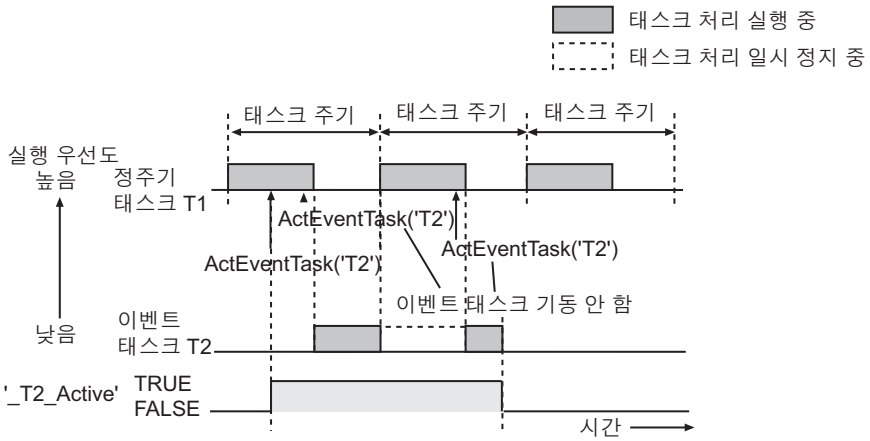
참고

시스템 정의 변수 '_**_Active'의 동작

- 이 명령을 실행하면 지정된 이벤트 태스크의 시스템 정의 변수 '_**_Active'가 TRUE가 됩니다. 이벤트 태스크의 실행이 완료되면 FALSE가 됩니다. 예를 들어, 정주기 태스크 T1보다 이벤트 태스크 T2의 실행 우선도가 낮다고 가정합니다. T1에서 T2를 지정하여 ActEventTask 명령을 실행하면 시스템 정의 변수 '_T2_Active'는 다음 그림과 같이 변화합니다.



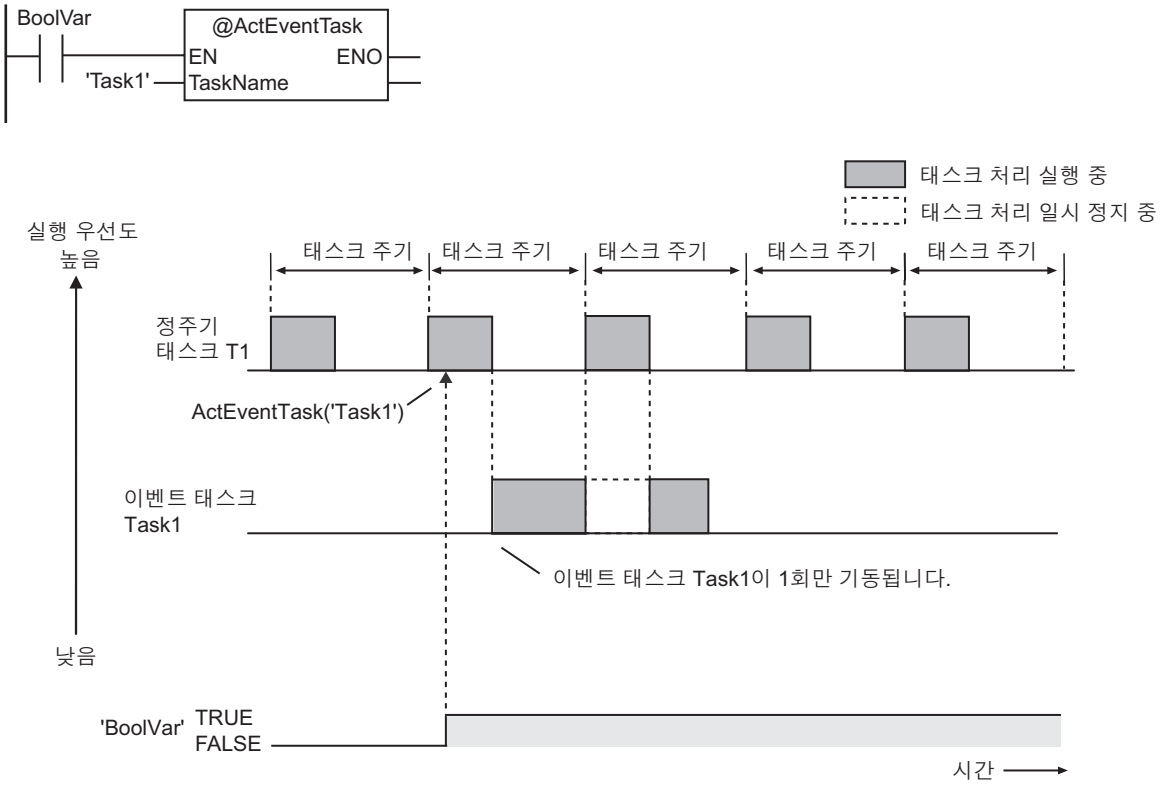
- 이벤트 태스크의 시스템 정의 변수 '_**_Active'가 TRUE인 동안에 이 명령을 실행해도 이벤트 태스크는 기동되지 않습니다.



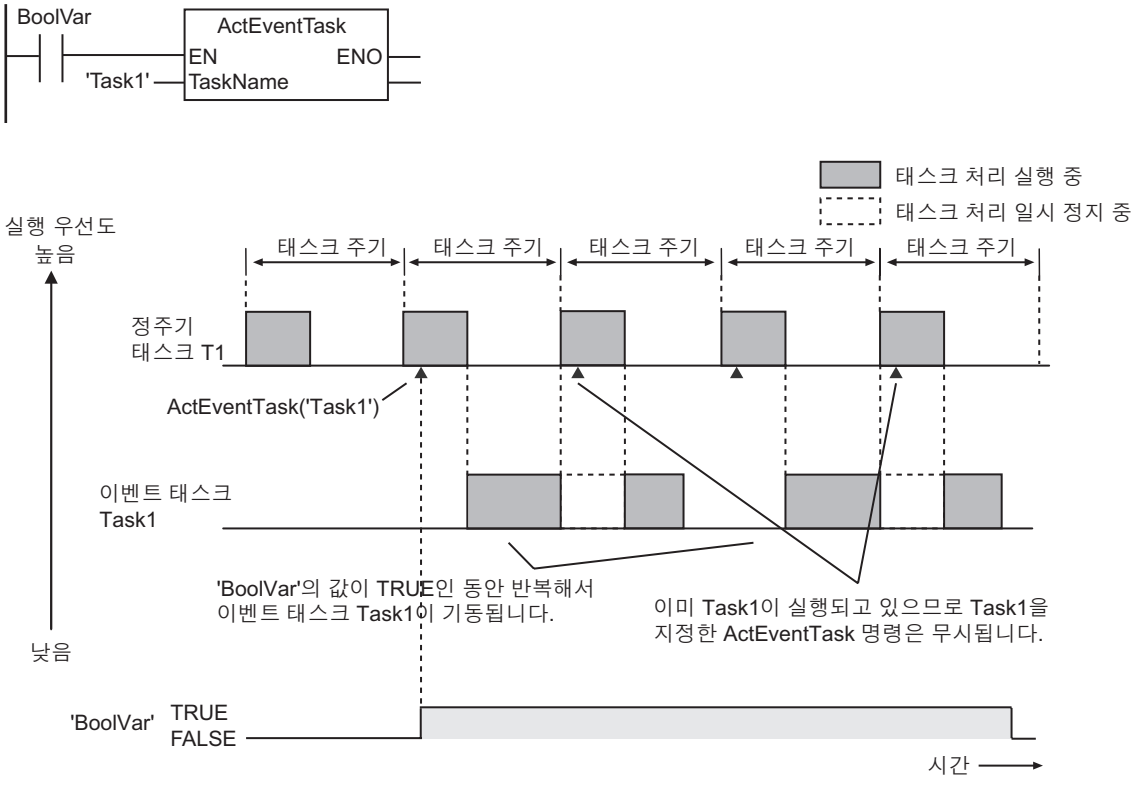
이벤트 태스크를 1회만 기동하는 경우와 반복 기동하는 경우

지정된 변수의 값이 변화했을 때 1회만 이벤트 태스크를 기동하려는 경우와 변수가 지정한 값인 동안 반복해서 이벤트 태스크를 기동하려는 경우에는 사용자 프로그램을 다음과 같이 재기입하십시오.

예 1) 변수의 값이 변화했을 때 이벤트 태스크를 1회만 기동하는 경우
다음 그림과 같이 이 명령에 입력 상승 미분의 동작 옵션을 추가하면 BOOL 형식 변수 BoolVar의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크 Task1이 1회만 기동됩니다.



예 2) 변수가 지정한 값인 동안 반복해서 이벤트 태스크를 기동하는 경우
다음 그림과 같이 이 명령에 입력 상승 미분의 동작 옵션을 추가하지 않으면 BOOL 형식 변수 BoolVar의 값이 TRUE인 동안, 반복해서 이벤트 태스크 Task1이 기동됩니다. 단, Task1 실행 중에는 Task1을 지정한 이 명령을 실행해도 무시됩니다.



사용상의 주의 사항

- 명령 실행 시간을 단축하고 싶은 경우에는 이벤트 태스크를 기동시키고 싶을 때만 이 명령을 실행하십시오. 시스템 정의 변수 '_**_Active'가 TRUE인 동안에 이 명령을 실행하는 경우, 이벤트 태스크를 기동하지 않아도 명령 실행 시간이 필요해집니다.
- 'TaskName'의 이벤트 태스크가 존재하지 않는 경우, 이상 상태가 됩니다. 'ENO'는 FALSE가 됩니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.03 이상, Sysmac Studio Ver.1.04 이상에서 사용할 수 있습니다.

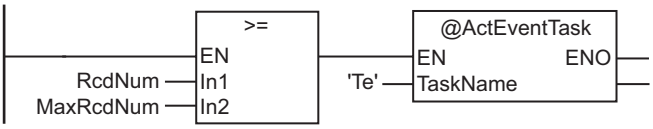
샘플 프로그램

● 변수의 값이 지정된 조건에 일치한 경우에 이벤트 태스크를 기동하는 예

변수 RcdNum이 변수 MaxRcdNum 미만에서 MaxRcdNum 이상으로 변화한 경우에 이벤트 태스크 Te를 1회만 기동합니다.

LD

이름	데이터 형식	초기값
RcdNum	INT	0
MaxRcdNum	INT	100



ST

이름	데이터 형식	초기값
RcdNum	INT	0
MaxRcdNum	INT	100
met	BOOL	FALSE

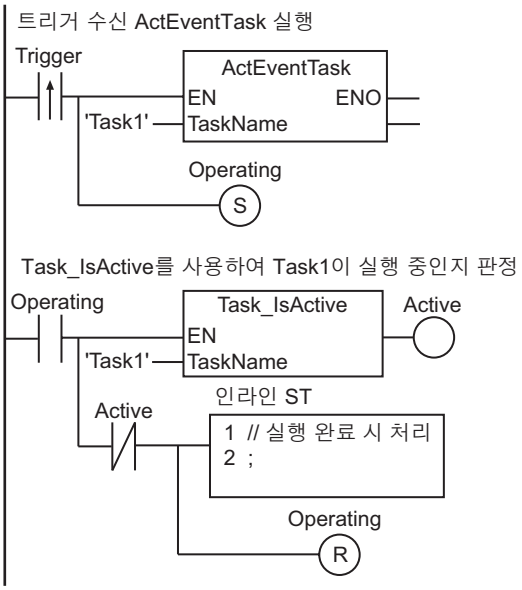
```
IF (RcdNum>=MaxRcdNum) THEN
  IF (met=FALSE) THEN
      ActEventTask('Te');
      met:=TRUE;
  END_IF;
ELSE
  met:=FALSE;
END_IF;
```

● 이벤트 태스크의 실행 완료를 확인한 후 다음 처리를 수행하는 예

변수 Trigger 기동 시 이벤트 태스크 Task1을 기동합니다. Task_IsActive 명령을 사용하여 Task1의 실행 완료를 확인합니다.

LD

이름	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
Operating	BOOL	FALSE	이벤트 태스크 실행 확인 중
Active	BOOL	FALSE	이벤트 태스크 실행 중



ST

이름	데이터 형식	초기값	코멘트
Trigger	BOOL	FALSE	실행 조건
LastTrigger	BOOL	FALSE	이전 태스크 주기의 Trigger의 값
Operating	BOOL	FALSE	이벤트 태스크 실행 확인 중
Active	BOOL	FALSE	이벤트 태스크 실행 중

```
// Trigger 기동에서 시퀀스 시작
IF ( (Trigger=TRUE) AND (LastTrigger=FALSE) ) THEN
  ActEventTask('Task1');       // 이벤트 태스크 Task1 기동
  Operating:=TRUE;
END_IF;
LastTrigger:=Trigger;

// Task1이 실행 중인지 판정
IF (Operating=TRUE) THEN
  Active:=Task_IsActive('Task1');

  IF (Active=FALSE) THEN       // Task1 실행 완료
    Operating:=FALSE;
  END_IF;

END_IF;
```

Get**Clk

지정된 주기의 클록 펄스를 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Get**Clk	클록 펄스 취득 그룹	FUN	Get**Clk EN — Out 단, **은 100us, 1ms, 10ms, 20ms, 100ms, 1s, 1min 중 하나	Out:=Get**Clk(); 단, **는 100us, 1ms, 10ms, 20ms, 100ms, 1s, 1min 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	클록 펄스	출력	클록 펄스	데이터 형식에 따름	–	–

	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out	○																			

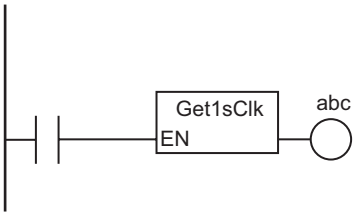
기능

지정한 주기의 클록 펄스를 출력합니다.
클록 펄스의 주기는 100us, 1ms, 10ms, 20ms, 100ms, 1s, 1min 중 하나입니다.

명령 이름은 클록 펄스의 주기에 따라 다릅니다. 예를 들면, 클록 펄스의 주기가 10ms인 경우, 명령 이름은 Get10msClk가 됩니다.

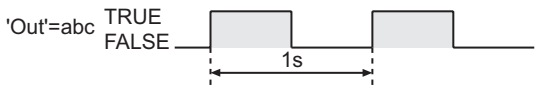
Get1sClk 명령인 경우의 예가 아래에 나와 있습니다.

LD



ST

```
abc:=Get1sClk();
```



사용상의 주의 사항

- 명령 실행 시, 처음의 'Out'의 값이 TRUE인지 FALSE인지는 정해져 있지 않습니다.
- 이 명령을 래더도 프로그램에서 사용한 경우, 이 명령의 회로의 이전 단계에서 이상이 발생하면 'Out'의 값은 FALSE가 됩니다.

Get**Cnt

지정된 주기의 프리 러닝 카운터 값을 취득합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
Get**Cnt	가산 프리 러닝 카운터 취득 그룹	FUN	Get**Cnt EN ENO Out 단, **는 100ns, 1us, 1ms, 10ms, 100ms, 1s 중 하나	Out:=Get**Cnt(); 단, **는 100ns, 1us, 1ms, 10ms, 100ms, 1s 중 하나

변수

	명칭	입/출력	내용	유효 범위	단위	초기값
Out	카운트값	출력	프리 러닝 카운터의 값	데이터 형식에 따름	–	–

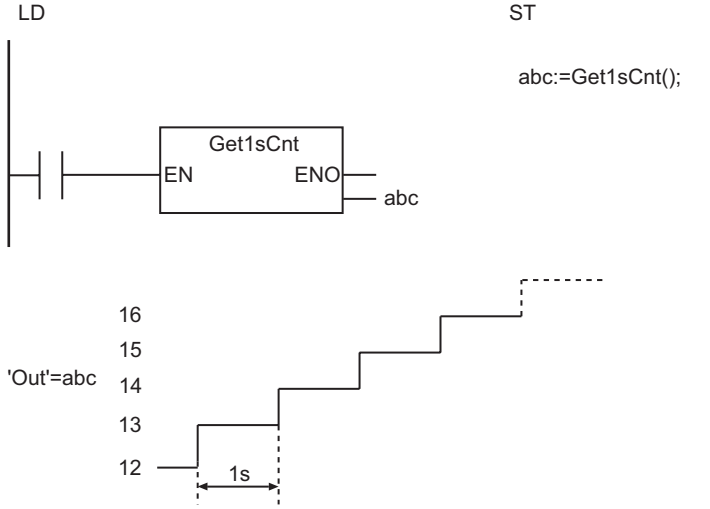
	부울	비트 열				정수								실수		시각, 지속 시간, 날짜, 문자열				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Out									○											

기능

지정한 주기의 프리 러닝 카운터의 값을 취득합니다.
프리 러닝 카운터란, 일정 주기로 카운트 업하는 카운터를 말하는 것으로 'Out'이 카운트값입니다. 카운터의 주기는 100ns, 1us, 1ms, 10ms, 100ms, 1s 중 하나입니다.

명령 이름은 카운터의 주기에 따라 다릅니다. 예를 들면, 카운터의 주기가 10ms인 경우, 명령 이름은 Get10msCnt가 됩니다.

Get1sCnt 명령인 경우의 예가 아래에 나와 있습니다.



사용상의 주의 사항

- 프리 러닝 카운터는 전원 ON과 함께 카운트를 시작합니다. 그리고, ULINT 형식의 유효 범위 18,446,744,073,709,551,615를 초과하면, 0으로 되돌아가서 카운트 업을 계속합니다.
- 이 명령은 프리 러닝 카운터값을 취득할 뿐이며, 0 리셋을 하는 것은 아닙니다.
- 'Out'의 값이 어디서부터 시작되는지는 정해져 있지 않습니다. 반드시 0에서부터 시작되는 것은 아닙니다.

A

부록

A-1 ErrorID로 확인할 수 있는 에러 코드 목록 A-2
A-2 에러 코드의 개요 A-14
A-3 에러 코드에 대한 자세한 내용 A-41
A-4 이벤트 태스크에서 사용할 수 없는 명령 A-162
A-5 NX 메시지 통신 이상 관련 명령 A-164
A-6 SDO Abort 코드 목록 A-165
A-7 버전 관련 정보 A-166

A-1 ErrorID로 확인할 수 있는 에러 코드 목록

명령 실행 시 발생하는 이상에는 해당 내용에 맞춰 에러 코드가 할당되어 있습니다. 출력 변수의 에러 코드 'ErrorID'가 있는 명령을 사용하는 경우, 에러 코드를 사용하여 이상 처리를 프로그래밍할 수 있습니다. 'ErrorID'가 있는 명령과 해당 명령에서 발생할 가능성이 있는 에러 코드의 목록이 아래에 나와 있습니다. 에러 코드의 내용은 '에러 코드에 대한 자세한 내용(A-41페이지)'을 참조하십시오.

참고

'ErrorID'가 없는 명령의 이상은 이벤트 로그 기능을 통해 이벤트로 확인할 수 있습니다.

종류	명령	명칭	에러 코드	에러명
아날로그 제어 명령	PIDAT	자동 조정 가능한 PID 연산	16#0400	입력값 범위 밖
			16#0401	수치 관계 오류
	AC_StepProgram	스텝 프로그램	16#0400	입력값 범위 밖
시스템 제어 명령	ResetPLCError	PLC 이상 해제	–	–
	ResetCJBError	I/O 버스 이상 해제	16#0400	입력값 범위 밖
			16#040D	지정 유니트 오류
	ResetMCError	모션 제어 이상 해제	–	–
	ResetECError	EtherCAT 이상 해제	16#041A	명령 다중 기동
	ResetUnit	유니트 재시작	16#0400	입력값 범위 밖
			16#040D	지정 유니트 오류
			16#040F	유니트 재시작 실패
	RestartNXUnit	NX 유니트 재시작	16#0400	입력값 범위 밖
			16#0419	데이터 형식 오류
			16#2C00	NX 메시지 이상
			16#2C01	NX 메시지 리소스 초과
			16#2C02	NX 메시지 타임아웃
			16#2C05	NX 메시지 네트워크 이상 (EtherCAT)
			16#2C06	지정 유니트 외부 재시작 이미 실행
			16#2C07	명령 대상 이외의 유니트 지정
	NX_ChangeWriteMode	NX 유니트 쓰기 모드 변경	16#0400	입력값 범위 밖
			16#0419	데이터 형식 오류
			16#2C00	NX 메시지 이상
			16#2C01	NX 메시지 리소스 초과
			16#2C02	NX 메시지 타임아웃
			16#2C05	NX 메시지 네트워크 이상 (EtherCAT)
			16#2C07	명령 대상 이외의 유니트 지정

종류	명령	명칭	에러 코드	에러명
시스템 제어 명령	NX_SaveParam	NX 유니트 파라메터 저장	16#0400	입력값 범위 밖
			16#0419	데이터 형식 오류
			16#2C00	NX 메시지 이상
			16#2C01	NX 메시지 리소스 초과
			16#2C02	NX 메시지 타임아웃
EtherCAT 통신 명령	EC_CoESDOWrite	CoE SDO 쓰기	16#0400	입력값 범위 밖
			16#1800	EtherCAT 통신 에러
			16#1801	EtherCAT 슬레이브 부재
			16#1802	EtherCAT 타임아웃
			16#1804	SDO Abort 에러
			16#1808	통신 리소스 초과
	EC_CoESDORead	CoE SDO 읽기	16#0400	입력값 범위 밖
			16#1800	EtherCAT 통신 에러
			16#1801	EtherCAT 슬레이브 부재
			16#1802	EtherCAT 타임아웃
			16#1803	수신 버퍼 초과
			16#1804	SDO Abort 에러
			16#1808	통신 리소스 초과
	EC_StartMon	EtherCAT 패킷 모니터 기능 시작	16#1805	패킷 모니터 저장 중
			16#1807	패킷 모니터 기능 동작 중
			16#1808	통신 리소스 초과
	EC_StopMon	EtherCAT 패킷 모니터 기능 정지	16#1806	패킷 모니터 기능 지원 안됨
			16#1808	통신 리소스 초과
	EC_SaveMon	EtherCAT 패킷 저장	16#1805	패킷 모니터 저장 중
			16#1807	패킷 모니터 기능 동작 중
			16#1808	통신 리소스 초과

종류	명령	명칭	에러 코드	에러명
EtherCAT 통신 명령	EC_CopyMon	EtherCAT 패킷 전송	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1403	지정 파일 오류
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
			16#1808	통신 리소스 초과
	EC_DisconnectSlave	EtherCAT 슬레이브 이탈	16#1800	EtherCAT 통신 에러
			16#1801	EtherCAT 슬레이브 부재
			16#1808	통신 리소스 초과
	EC_ConnectSlave	EtherCAT 슬레이브 재가입	16#1800	EtherCAT 통신 에러
			16#1801	EtherCAT 슬레이브 부재
			16#1808	통신 리소스 초과
	EC_ChangeEnable Setting	슬레이브의 유효/무효 전환	16#1800	EtherCAT 통신 에러
			16#1801	EtherCAT 슬레이브 부재
			16#1808	통신 리소스 초과
	NX_WriteObj	NX 개체 쓰기	16#0400	입력값 범위 밖
			16#0419	데이터 형식 오류
			16#041B	데이터 용량 초과
			16#2C00	NX 메시지 이상
			16#2C01	NX 메시지 리소스 초과
			16#2C02	NX 메시지 타임아웃
			16#2C03	NX 메시지 길이 오류

종류	명령	명칭	에러 코드	에러명
EtherCAT 통신 명령	NX_ReadObj	NX 개체 읽기	16#0400	입력값 범위 밖
			16#0410	문자열 형식 이상
			16#0419	데이터 형식 오류
			16#041C	데이터 크기 불일치
			16#2C00	NX 메시지 이상
			16#2C01	NX 메시지 리소스 초과
			16#2C02	NX 메시지 타임아웃
EtherNet/IP 통신 명령	CIPOpen	CIPClass3(Large_Forward_Open) 연결 확립	16#0400	입력값 범위 밖
			16#1C00	Explicit 이상
			16#1C01	라우팅 경로 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#1C05	Class3 연결 확립 실패
			16#2000	로컬 IP 주소 설정 에러
			16#2003	상태 이상
			16#2004	로컬 IP 주소 결정 안됨
	CIPOpenWithDataSize	CIPClass3 연결 확립(크기 지정 있음)	16#0400	입력값 범위 밖
			16#1C00	Explicit 이상
			16#1C01	라우팅 경로 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#1C05	Class3 연결 확립 실패
			16#2000	로컬 IP 주소 설정 에러
			16#2003	상태 이상
			16#2004	로컬 IP 주소 결정 안됨
	CIPRead	변수 읽기(Class3 Explicit 메시지)	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#0419	데이터 형식 오류
			16#1C00	Explicit 이상
			16#1C02	CIP 핸들 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#1C06	CIP 통신 데이터 크기 초과

종류	명령	명칭	에러 코드	에러명
EtherNet/IP 통신 명령	CIPWrite	변수 쓰기(Class3 Explicit 메시지)	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#0407	영역 초과
			16#1C00	Explicit 이상
			16#1C02	CIP 핸들 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#1C06	CIP 통신 데이터 크기 초과
	CIPSend	임의 Explicit 메시지 송신(Class3)	16#0406	영역 범위 외 지정
			16#0407	영역 초과
			16#0419	데이터 형식 오류
			16#1C00	Explicit 이상
			16#1C02	CIP 핸들 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#1C06	CIP 통신 데이터 크기 초과
	CIPClose	CIP Class3 연결 중단	16#1C02	CIP 핸들 오류
			16#1C03	CIP 통신 리소스 초과
	CIPUCMMRead	변수 읽기(UCMM Explicit 메시지)	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#0419	데이터 형식 오류
			16#1C00	Explicit 이상
			16#1C01	라우팅 경로 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#2000	로컬 IP 주소 설정 에러
			16#2004	로컬 IP 주소 결정 안됨
	CIPUCMMWrite	변수 쓰기(UCMM Explicit 메시지)	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#0419	데이터 형식 오류
			16#1C00	Explicit 이상
			16#1C01	라우팅 경로 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#2000	로컬 IP 주소 설정 에러
			16#2004	로컬 IP 주소 결정 안됨

종류	명령	명칭	에러 코드	에러명
EtherNet/IP 통신 명령	CIPUCMMSend	임의 Explicit 메시지 송신(UCMM)	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#0407	영역 초과
			16#0419	데이터 형식 오류
			16#1C00	Explicit 이상
			16#1C01	라우팅 경로 오류
			16#1C03	CIP 통신 리소스 초과
			16#1C04	CIP 타임아웃
			16#2000	로컬 IP 주소 설정 에러
			16#2004	로컬 IP 주소 결정 안됨
	SktUDPCreate	UDP 소켓 생성	16#0400	입력값 범위 밖
			16#2000	로컬 IP 주소 설정 에러
			16#2001	TCP/UDP 포트 사용 불가능
			16#2002	주소 해결 실패
			16#2003	상태 이상
			16#2004	로컬 IP 주소 결정 안됨
			16#2008	소켓 통신 리소스 초과
	SktUDPRcv	UDP 소켓 수신	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#2003	상태 이상
			16#2006	소켓 타임아웃
			16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktUDPSend	UDP 소켓 송신	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#2002	주소 해결 실패
			16#2003	상태 이상
			16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktTCPAccept	TCP 소켓 허용	16#0400	입력값 범위 밖
			16#2000	로컬 IP 주소 설정 에러
			16#2002	주소 해결 실패
			16#2003	상태 이상
			16#2004	로컬 IP 주소 결정 안됨
			16#2006	소켓 타임아웃
			16#2008	소켓 통신 리소스 초과

종류	명령	명칭	에러 코드	에러명
EtherNet/IP 통신 명령	SktTCPConnect	TCP 소켓 연결	16#0400	입력값 범위 밖
			16#2000	로컬 IP 주소 설정 에러
			16#2002	주소 해결 실패
			16#2003	상태 이상
			16#2008	소켓 통신 리소스 초과
	SktTCPRcv	TCP 소켓 수신	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#2003	상태 이상
			16#2006	소켓 타임아웃
			16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktTCPSend	TCP 소켓 송신	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#2003	상태 이상
			16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktGetTCPStatus	TCP 소켓의 Status 읽기	16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktClose	TCP/UDP 소켓 클로즈	16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	SktClearBuf	TCP/UDP 소켓 수신 버퍼 클리어	16#2007	소켓 핸들 오류
			16#2008	소켓 통신 리소스 초과
	ChangeIPAdr	IP 주소 변경	16#0400	입력값 범위 밖
			16#040D	지정 유니트 오류
			16#2400	실행 권한 없음
	ChangeFTPAccount	FTP 계정 변경	16#0400	입력값 범위 밖
			16#040D	지정 유니트 오류
			16#2400	실행 권한 없음
	ChangeNTPServerAdr	NTP 서버 수신처 변경	16#0400	입력값 범위 밖
			16#040D	지정 유니트 오류
			16#2400	실행 권한 없음

종류	명령	명칭	에러 코드	에러명
시리얼 통신 명령	ExecPMCR	프로토콜 매크로	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#0407	영역 초과
			16#040D	지정 유니트 오류
			16#0413	메모리 이외의 주소 지정
			16#0419	데이터 형식 오류
			16#0C00	시리얼 통신 모드 오류
			16#0800	FINS 이상
			16#0801	FINS 포트 사용 불가능
	SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#040D	지정 유니트 오류
			16#0419	데이터 형식 오류
			16#0C00	시리얼 통신 모드 오류
			16#0800	FINS 이상
			16#0801	FINS 포트 사용 불가능
	SerialRcv	시리얼 커뮤니케이션 유니트 시리얼 포트 입력	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#040D	지정 유니트 오류
			16#0419	데이터 형식 오류
			16#0C00	시리얼 통신 모드 오류
			16#0800	FINS 이상
			16#0801	FINS 포트 사용 불가능
	SerialRcvNoClear	시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함	16#0400	입력값 범위 밖
			16#0407	영역 초과
			16#040D	지정 유니트 오류
			16#0419	데이터 형식 오류
			16#0C00	시리얼 통신 모드 오류
			16#0800	FINS 이상
			16#0801	FINS 포트 사용 불가능
	SendCmd	명령 송신	16#0400	입력값 범위 밖
			16#0406	영역 범위 외 지정
			16#0407	영역 초과
			16#0419	데이터 형식 오류
			16#0800	FINS 이상
			16#0801	FINS 포트 사용 불가능

종류	명령	명칭	에러 코드	에러명
SD 메모리 카드 명령	FileWriteVar	변수 파일 쓰기	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1403	지정 파일 오류
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#1409	동일한 파일명 있음
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	FileReadVar	변수 파일 읽기	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	FileOpen	파일 열기	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1403	지정 파일 오류
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	FileClose	파일 닫기	16#1400	SD 메모리 카드 사용 불가능
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#140E	SD 메모리 카드 액세스 실패

종류	명령	명칭	에러 코드	에러명
SD 메모리 카드 명령	FileSeek	파일 검색	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#1407	오프셋 범위 밖
			16#140E	SD 메모리 카드 액세스 실패
	FileRead	파일 읽기	16#0406	영역 범위 외 지정
			16#0419	데이터 형식 오류
			16#1400	SD 메모리 카드 사용 불가능
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#1406	열기 모드 불일치
			16#140E	SD 메모리 카드 액세스 실패
	FileWrite	파일 쓰기	16#0406	영역 범위 외 지정
			16#0419	데이터 형식 오류
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#1406	열기 모드 불일치
			16#140E	SD 메모리 카드 액세스 실패
	FileGets	문자열 읽기	16#1400	SD 메모리 카드 사용 불가능
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#1406	열기 모드 불일치
			16#140E	SD 메모리 카드 액세스 실패
	FilePuts	문자열 쓰기	16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#1406	열기 모드 불일치
			16#140E	SD 메모리 카드 액세스 실패

종류	명령	명칭	에러 코드	에러명
SD 메모리 카드 명령	FileCopy	파일 복사	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1403	지정 파일 오류
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#1409	동일한 파일명 있음
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	FileRemove	파일 삭제	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1403	지정 파일 오류
			16#1405	파일 액세스 중
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	FileRename	파일명 변경	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1403	지정 파일 오류
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#1408	빈 디렉토리 이외
			16#1409	동일한 파일명 있음
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패

종류	명령	명칭	에러 코드	에러명
SD 메모리 카드 명령	DirCreate	디렉토리 작성	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1402	SD 메모리 카드 용량 부족
			16#1404	파일 · 디렉토리 최대 수 초과
			16#1405	파일 액세스 중
			16#1409	동일한 파일명 있음
			16#140B	파일 열기 최대 수 초과
			16#140C	지정 디렉토리 오류
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패
	DirRemove	디렉토리 삭제	16#0400	입력값 범위 밖
			16#1400	SD 메모리 카드 사용 불가능
			16#1401	SD 메모리 카드 쓰기 방지
			16#1405	파일 액세스 중
			16#1408	빈 디렉토리 이외
			16#140A	지정 파일 쓰기 금지
			16#140B	파일 열기 최대 수 초과
			16#140C	지정 디렉토리 오류
			16#140D	파일 · 디렉토리명 최대 길이 초과
			16#140E	SD 메모리 카드 액세스 실패

A-2 에러 코드의 개요

명령에서 발생하는 이상(이벤트)의 목록이 아래에 나와 있습니다. 이벤트 코드의 하위 4자리가 명령의 에러 코드를 나타냅니다. 각 에러 코드의 설명은 대응하는 이벤트 코드의 설명을 참조하십시오. 예를 들어, 명령의 에러 코드가 16#0400인 경우는 54010400Hex의 이벤트 코드 설명을 참조하십시오.

표에 나오는 중요도는 다음과 같은 의미입니다.
- 전체 : 전체 정지 이상 레벨
- 부분 : 부분 정지 이상 레벨
- 가벼움 : 가벼운 이상 레벨
- 감시 : 감시 정보
- 일반 : 일반 정보

이벤트 코드란의 괄호 안에는 에러 코드가 추가된 CPU 유니트의 유니트 버전이 표시됩니다.

각 에러 코드에 대한 자세한 내용은 '에러 코드에 대한 자세한 내용(A-41페이지)'을 참조하십시오.

NJ 시리즈 이상(이벤트)의 개념과 모든 이벤트 코드에 대해서는 'NJ 시리즈 트러블 슈팅 설명서(SBCA-361)'를 참조하십시오.
명령의 이벤트 코드는 CPU 유니트의 유니트 버전 Ver.1.02 이상에서 지원됩니다.

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54010400Hex	입력값 범위 밖	명령의 입력 파라메터가 입력 변수의 범위를 초과했습니다. 또는 정수 0으로 제산/나머지 연산을 수행했습니다.	• 명령의 입력 파라메터가 입력 변수의 범위를 초과함. 또는 정수 0으로 제산/나머지 연산을 수행함				○		A-42페이지
54010401Hex	수치 관계 오류	명령의 입력 파라메터 값의 관계가 조건을 충족하지 않습니다. 또는 명령 실행 연산 중/결과의 수치가 조건을 충족하지 못했습니다.	• 입력 파라메터 값의 관계가 조건을 충족하지 않음 • 명령 실행 연산 중/결과의 수치가 조건을 충족하지 않음				○		A-42페이지
54010402Hex	부동 소수점 에러	명령의 부동 소수점 수의 입력 파라메터에 대해 수치 이외의 것이 입력되었습니다.	• 명령의 부동 소수점 수의 입력 파라메터에 대해 수치 이외의 것이 입력됨				○		A-43페이지
54010403Hex	BCD가 아님	명령의 BCD 입력 파라메터에 대해 BCD 데이터가 아닌 값이 입력되었습니다.	• 명령의 BCD 입력 파라메터에 대해 16진수 A, B, C, D, E, F가 입력됨				○		A-43페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54010404Hex	부호가있는 BCD 에러	명령의 부호가 있는 BCD 데이터의 입력 파라메터에 대해 최상위 자리에 잘못된 값이 입력되었습니다.	• 명령의 부호가 있는 BCD 데이터의 입력 파라메터에 대해 최상위 자리에 잘못된 값이 입력됨 • BCD 형식에 '_BCD0'이 지정된 경우는 최상위 자리가 2~F가 됨 • BCD 형식에 '_BCD2'가 지정된 경우는 최상위 자리가 A, B, C, D, E 중 하나가 됨 • BCD 형식에 '_BCD3'이 지정된 경우는 최상위 자리가 B, C, D, E가 됨				○		🕮 A-44페이지
54010405Hex	지정 비트 위치 오류	명령으로 지정된 비트 위치가 잘못되었습니다.	• 명령으로 지정된 비트 위치가 지정된 데이터의 범위를 초과함				○		🕮 A-44페이지
54010406Hex	영역범위외 지정	명령으로 지정된 데이터의 메모리 주소 또는 데이터 크기가 적절하지 않습니다.	• 명령으로 지정된 데이터의 메모리 주소가 유효 범위 밖인 경우, 명령으로 지정된 데이터의 크기가 유효 범위를 초과하는 경우, 변수의 데이터 형식과 데이터 크기가 맞지 않는 경우 등을 생각할 수 있음				○		🕮 A-45페이지
54010407Hex	영역 초과	명령의 연산 결과가 출력 파라메터의 데이터 영역의 범위를 초과했습니다.	• 배열 요소 수 등 명령의 처리 결과가 출력 파라메터의 데이터 영역의 범위를 초과함				○		🕮 A-45페이지
54010409Hex	컨트롤러 이상 해제 실패	컨트롤러 이상이 발생하지 않은 상태에서 컨트롤러 이상 해제 명령을 실행했습니다.	• 컨트롤러 이상이 발생하지 않은 상태에서 컨트롤러 이상 해제 명령을 실행함				○		🕮 A-46페이지
5401040BHex	사용자이상 해제 실패	사용자 이상이 발생하지 않은 상태에서 사용자 이상 해제 명령을 실행했습니다.	• 사용자 이상이 발생하지 않은 상태에서 사용자 이상 해제 명령을 실행함				○		🕮 A-46페이지
5401040CHex	사용자이상 최대 건수	사용자 이상 명령에서 최대 건수를 초과하는 사용자 이상을 발생시키려고 했습니다.	• 사용자 이상 명령에서 최대 건수를 초과하는 사용자 이상을 발생시키려고 함				○		🕮 A-47페이지
5401040DHex	지정 유니트 오류	명령으로 지정된 유니트가 존재하지 않습니다.	• 유니트 구성 정보로 존재하지 않는 유니트를 지정함 • 유니트 구성 정보에 존재하지만 실체는 존재하지 않는 유니트를 지정함				○		🕮 A-47페이지
5401040FHex	유니트 재시작 실패	고기능 유니트의 재시작에 실패했습니다.	• 고기능 유니트가 처리 중				○		🕮 A-48페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54010410Hex	문자열 형식 이상	명령에 입력된 문자열이 올바른 문자열이 아닙니다.	• 문자열을 수치로 변환하는 명령의 입력 문자열이 수치를 나타내는 문자열 또는 양수값을 나타내는 문자열이 아님 • 입력 문자열이 NULL 문자로 끝나지 않음				○		🕮 A-48페이지
54010411Hex	지정 프로그램 오류	명령으로 지정된 프로그램이 존재하지 않습니다.	• 해당 명령으로 지정된 프로그램이 존재하지 않거나 삭제됨				○		🕮 A-49페이지
54010413Hex	메모리 이외의 주소 지정	CJ 유니트용 메모리에 지정해야 하는 변수가 CJ 유니트용 메모리에 지정되지 않았습니다.	• CJ 유니트용 메모리에 지정해야 하는 변수가 CJ 유니트용 메모리에 AT 지정되지 않음				○		🕮 A-49페이지
54010414Hex	스택 언더플로우	스택에 데이터가 존재하지 않습니다.	• 데이터가 존재하지 않는 스택에서 데이터를 가져오려고 함				○		🕮 A-50페이지
54010416Hex	배열의 요소 수 · 차원 수 오류	명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위를 초과했습니다.	• 명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위를 초과함				○		🕮 A-50페이지
54010417Hex	지정 태스크 오류	명령으로 지정된 태스크가 존재하지 않습니다.	• 지정된 태스크가 존재하지 않음				○		🕮 A-51페이지
54010418Hex	지정 태스크가 허가되지 않음	지정이 허가되지 않은 태스크를 명령으로 지정했습니다.	• 로컬 태스크를 지정하거나 프라이머리 주기 태스크 및 주기 태스크를 지정함				○		🕮 A-51페이지
54010419Hex	데이터 형식 오류	명령에서 사용할 수 없는 형식의 데이터가 입력/입출력으로 지정되었습니다.	• 명령에서 사용할 수 없는 형식의 데이터를 입력/입출력으로 지정함				○		🕮 A-52페이지
5401041AHex	명령 다중 기동	다중 기동할 수 없는 명령을 다중 기동했습니다.	• 다중 기동할 수 없는 명령을 여러 개 동시에 실행하려고 함				○		🕮 A-52페이지
5401041BHex (Ver.1.02)	데이터 용량 초과	명령에 전달된 데이터가 너무 커서 처리할 수 없습니다.	• 명령이 처리할 수 있는 용량보다 큰 데이터가 명령에 전달됨				○		🕮 A-53페이지
5401041CHex (Ver.1.04)	데이터 크기 불일치	명령의 입력/입출력에서 지정된 데이터가 대상 파라메터의 크기와 일치하지 않습니다.	• 대상 파라메터의 크기와 일치하지 않는 크기의 데이터를 명령의 입력/입출력에 지정함						🕮 A-53페이지
5401041DHex (Ver.1.05)	명령 동시 실행 리소스 초과	동시에 실행 가능한 해당 명령군의 리소스를 초과하여 명령이 실행되었습니다.	• 동시에 실행할 수 있는 개수를 초과하여 해당 명령이 실행됨. 해당 명령군 및 최대 개수는 다음과 같습니다. • DB 연결 명령: 32개						🕮 A-54페이지
54010800Hex	FINS 이상	FINS 명령의 송수신에서 이상이 발생했습니다.	• FINS 명령의 송수신에서 이상이 발생함				○		🕮 A-54페이지
54010801Hex	FINS 포트 사용 불가능	FINS 포트를 사용 중입니다.	• FINS 포트 사용 중				○		🕮 A-55페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54010C00Hex	시리얼 통신 모드 오류	시리얼 커뮤니케이션 유니트가 명령의 실행에 필요한 시리얼 통신 모드가 아닙니다.	• 시리얼 커뮤니케이션 유니트의 시리얼 통신 포트가 명령에서 전제로 하는 모드로 설정되어 있지 않음				○		🕮 A-55페이지
54010C02Hex	포트 설정 변경 다중 기동	포트 설정 변경 명령을 실행하는 중에 다른 포트 설정 변경 명령을 실행했습니다.	• 포트 설정 변경 명령을 실행하는 중에 다른 포트 설정 변경 명령을 실행함				○		🕮 A-56페이지
54011400Hex	SD 메모리 카드 사용 불가능	명령 실행 시 SD 메모리 카드에 액세스하지 못했습니다.	• SD 메모리 카드가 설치되어 있지 않거나 정확하게 삽입되지 않음 • SD 메모리 카드가 손상됨 • SD 메모리 카드 슬롯이 손상됨				○		🕮 A-56페이지
54011401Hex	SD 메모리 카드 쓰기 방지	명령 실행 시 쓰기 방지된 SD 메모리 카드에 쓰려고 했습니다.	• 쓰기 방지된 SD 메모리 카드에 쓰려고 함				○		🕮 A-57페이지
54011402Hex	SD 메모리 카드 용량 부족	명령 실행 중에 SD 메모리 카드에 쓸 때 SD 메모리 카드의 용량 부족이 발생했습니다.	• SD 메모리 카드의 용량 부족이 발생함				○		🕮 A-57페이지
54011403Hex	지정 파일 오류	명령으로 지정된 파일이 존재하지 않습니다.	• 지정된 파일이 존재하지 않음				○		🕮 A-58페이지
54011404Hex	파일 · 디렉토리 최대 수 초과	명령 실행 중에 파일/디렉토리를 작성할 때 파일/디렉토리 수가 최대 수를 초과했습니다.	• 파일/디렉토리 수가 최대 수를 초과함				○		🕮 A-58페이지
54011405Hex	파일 액세스 중	명령으로 지정된 파일을 사용 중이므로 액세스하지 못했습니다.	• 다른 명령에서 액세스 중인 파일에 또 다른 명령에서 쓰기 또는 읽기를 실행하려고 함				○		🕮 A-59페이지
54011406Hex	열기 모드 불일치	명령의 파일 조작이 파일의 열기 모드와 일치하지 않습니다.	• 파일 열기 명령으로 지정된 파일의 열기 모드와 그 이후 SD 메모리 카드용 명령에 의한 파일 조작에 부정합이 있음				○		🕮 A-59페이지
54011407Hex	오프셋 범위 밖	명령으로 지정된 오프셋으로 지정되는 주소에 액세스할 수 없습니다.	• 파일 크기를 초과하여 액세스하려고 함				○		🕮 A-60페이지
54011408Hex	빈 디렉토리 이외	디렉토리 삭제 명령을 실행하거나 디렉토리 이름을 변경하려고 했지만 디렉토리 내용이 비어 있지 않습니다.	• 디렉토리 삭제 명령을 실행하려고 했지만 디렉토리 내용이 비어 있지 않음 • 디렉토리 이름을 변경하려고 했지만 디렉토리 안에 디렉토리가 존재함				○		🕮 A-60페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54011409Hex	동일한 파일명 있음	명령으로 지정된 파일과 동일한 이름의 파일이 존재하므로 명령을 실행하지 못했습니다.	• 명령으로 지정된 작성 파일과 동일한 이름의 파일이 이미 존재함				○		🕮 A-61페이지
5401140AHex	지정 파일 쓰기 금지	명령 실행 시 쓰기 방지된 파일 또는 디렉토리에 쓰려고 했습니다.	• 명령으로 지정된 쓰기 파일 또는 디렉토리에 쓰기 방지가 지정되어 있음				○		🕮 A-61페이지
5401140BHex	파일 열기 최대 수 초과	명령 실행 중에 파일을 열 때 열린 파일 수가 최대 수를 초과했습니다.	• 명령 실행 중에 파일을 열 때 열린 파일 수가 최대 수를 초과함				○		🕮 A-62페이지
5401140CHex	지정 디렉토리 오류	명령으로 지정된 디렉토리가 존재하지 않습니다.	• 명령으로 지정된 디렉토리가 존재하지 않음				○		🕮 A-62페이지
5401140DHex	파일 · 디렉토리명 최대 길이 초과	명령으로 지정된 파일명 또는 디렉토리명이 너무 깁니다.	• 명령으로 지정된 작성 파일명 또는 작성 디렉토리명이 너무 김				○		🕮 A-63페이지
5401140EHex	SD 메모리 카드 액세스 실패	SD 메모리 카드에 액세스하지 못했습니다.	• SD 메모리 카드가 손상됨 • SD 메모리 카드 슬롯이 손상됨				○		🕮 A-63페이지
54011800Hex	EtherCAT 통신 에러	명령 실행 시 EtherCAT 네트워크에 액세스하지 못했습니다.	• EtherCAT 네트워크가 실행 가능한 상태가 아님				○		🕮 A-64페이지
54011801Hex	EtherCAT 슬레이브 부재	명령 실행 시 대상 슬레이브에 액세스하지 못했습니다.	• 대상 슬레이브가 존재하지 않음 • 대상 슬레이브가 실행 가능한 상태가 아님				○		🕮 A-64페이지
54011802Hex	EtherCAT 타임아웃	명령 실행 시 EtherCAT 슬레이브에 대한 액세스가 타임아웃되었습니다.	• 대상 슬레이브와의 통신이 타임아웃됨				○		🕮 A-65페이지
54011803Hex	수신 버퍼 초과	명령 실행 시 EtherCAT 슬레이브에서의 수신 데이터가 수신 버퍼를 초과했습니다.	• 슬레이브에서의 수신 데이터 크기가 수신 버퍼를 초과함				○		🕮 A-65페이지
54011804Hex	SDO Abort 에러	명령 실행 시 EtherCAT 슬레이브에서 SDO Abort 에러를 수신했습니다.	• 슬레이브 사양에 따름				○		🕮 A-66페이지
54011805Hex	패킷 모니터 저장 중	EtherCAT의 패킷 모니터 파일 저장 중에 패킷 모니터 조작 관련 명령이 실행되었습니다.	• EtherCAT의 패킷 모니터의 저장 중에 패킷 모니터 조작 관련 명령이 실행됨				○		🕮 A-66페이지
54011806Hex	패킷 모니터 기능 시작 안 함	EtherCAT의 패킷 모니터 정지 중에 패킷 모니터 정지 명령이 실행되었습니다.	• EtherCAT의 패킷 모니터 기능 정지 중에 패킷 모니터 정지 명령을 실행함				○		🕮 A-67페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54011807Hex	패킷 모니터 기능 동작 중	EtherCAT의 패킷 모니터 동작 중에 패킷 모니터 시작 명령이 실행되었습니다.	• EtherCAT의 패킷 모니터 기능 동작 중에 다시 패킷 모니터 시작 명령을 실행함				○		A-67페이지
54011808Hex	통신 리소스 초과	동시에 32개를 넘는 EtherCAT 통신 명령이 실행되었습니다.	• 동시에 32개를 넘는 EtherCAT 통신 명령이 실행됨. EtherCAT 통신 명령은 다음과 같음 • EC_CoESDOWrite 명령 • EC_CoESDORead 명령 • EC_ConnectSlave 명령 • EC_DisconnectSlave 명령 • EC_StartMon 명령 • EC_SaveMon 명령 • EC_StopMon 명령 • EC_CopyMon 명령				○		A-68페이지
54011809Hex (Ver.1.01)	패킷 모니터 기능 지원 안됨	패킷 모니터 기능을 사용할 수 없습니다.	• 패킷 모니터 기능이 탑재되지 않은 CPU 유니트에 대해 패킷 모니터 기능의 명령이 실행됨				○		A-68페이지
54011C00Hex	Explicit 이상	CIP 통신 명령에서 Explicit 메시지로 에러 응답 코드가 반환되었습니다.	• 이상의 내용에 따름				○		A-69페이지
54011C01Hex	라우팅 경로 오류	CIP 통신 명령으로 지정된 라우팅 경로의 형식이 잘못되었습니다.	• CIP 통신 명령으로 지정된 라우팅 경로의 형식이 잘못됨				○		A-69페이지
54011C02Hex	CIP 핸들 오류	CIP 통신 명령으로 지정된 핸들이 잘못되었습니다.	• CIP 통신 명령으로 지정된 핸들이 잘못됨				○		A-70페이지
54011C03Hex	CIP 통신 리소스 초과	동시에 실행 가능한 CIP 통신 명령 리소스를 초과하여 명령이 실행되었습니다.	• 동시에 32개를 넘는 CIP 통신 명령이 실행됨 • 동시에 32개를 넘는 핸들을 사용하려고 함				○		A-70페이지
54011C04Hex	CIP 타임아웃	CIP 통신 명령의 실행 중에 타임아웃이 발생했습니다.	• 지정된 IP 주소의 장치가 존재하지 않음 • 지정된 핸들의 CIP 연결이 타임아웃되었으므로 닫힘 • 상대 장치의 전원 OFF • 상대 장치의 통신 정지 • EtherNet/IP의 Ethernet 케이블 커넥터의 연결 불량 • EtherNet/IP의 Ethernet 케이블 단선 • 노이즈				○		A-71페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54011C05Hex (Ver.1.06)	Class3 연결 확립 실패	CIP 통신 명령에서 Class3 연결 확립에 실패했습니다.	• Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해 CIPOpen 명령을 실행함 • Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해 데이터 크기를 510바이트 이상으로 설정하여 CIPOpenWithDataSize 명령을 실행함				○		🕮 A-72페이지
54011C06Hex (Ver.1.06)	CIP 통신 데이터 크기 초과	CIP 통신 명령에서 송신 가능한 데이터 크기를 초과하는 Class3 Explicit 메시지를 송신하려고 했습니다.	• CIPRead 명령, CIPWrite 명령, CIPSend 명령의 입력 변수로 설정한 데이터 크기가 CIPOpenWithDataSize 명령으로 설정한 데이터 크기를 초과함				○		🕮 A-72페이지
54012000Hex	로컬 IP 주소 설정 에러	로컬 IP 주소의 설정 에러가 발생한 상태에서 명령을 실행했습니다.	• 로컬 IP 주소의 설정 에러가 발생한 상태에서 명령을 실행함				○		🕮 A-73페이지
54012001Hex	TCP/UDP 포트 사용 불가능	명령 실행 시 UDP 또는 TCP 포트가 이미 사용되고 있습니다.	• UDP 또는 TCP 포트가 이미 사용되고 있음				○		🕮 A-74페이지
54012002Hex	주소 해결 실패	명령에서 도메인명이 지정된 상대 노드의 주소 해결에 실패했습니다.	• 해당 명령으로 지정된 도메인명이 잘못됨 • 컨트롤러의 hosts 설정/DNS 설정이 잘못됨 • DNS 서버의 설정이 잘못됨				○		🕮 A-74페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54012003Hex	상태 이상	명령 실행 시의 상태가 적절하지 않습니다.	• SktUDPRcv 명령의 경우 • 소켓이 수신 처리 중 • 소켓이 열리지 않음 • SktUDPSend 명령의 경우 • 소켓이 송신 처리 중 • 소켓이 열리지 않음 • SktTCPAccept 명령의 경우 지정된 TCP 포트가 아래 중 하나 • 열기 처리 중 • 클로즈 처리 중 • 동일한 IP 주소, TCP 포트에서 이 명령이 이미 연결을 확립함 • SktTCPConnect 명령의 경우 • 입력 변수 'SrcTcpPort'에 지정된 TCP 포트가 이미 열려 있음 • 입력 변수 'DstAdr'로 지정된 상대 노드가 존재하지 않음 • 입력 변수 'DstAdr', 'DstTcpPort'로 지정된 상대 노드가 연결 대기 중이 아님 • SktTCPRcv 명령의 경우 • 지정된 소켓이 수신 처리 중 • 지정된 소켓의 연결이 확립되지 않음 • SktTCPSend 명령의 경우 • 지정된 소켓이 송신 처리 중 • 지정된 소켓의 연결이 확립되지 않음				○		A-75페이지
54012004Hex	로컬 IP 주소 결정 안됨	소켓 서비스 명령 실행 시 로컬 IP 주소가 결정되지 않았습니다.	• BOOTP 서버의 설정 이상 • BOOTP 서버 부재 • 기동 직후이므로 로컬 IP 주소가 결정되지 않았음				○		A-76페이지
54012006Hex	소켓 타임아웃	소켓 서비스 명령에서 타임아웃이 발생했습니다.	• SktTCPAccept 명령: 사용자가 지정한 타임아웃 시간 이내에 상대 노드에서 연결 요구가 없음 • SktTCPRcv 명령, SktUDPRcv 명령: 사용자가 지정한 타임아웃 시간 이내에 상대 노드에서 데이터가 수신되지 않음				○		A-76페이지
54012007Hex	소켓 핸들 오류	소켓 서비스 명령으로 지정된 핸들이 잘못되었습니다.	• 소켓 서비스 명령으로 지정된 핸들이 잘못됨				○		A-77페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54012008 Hex	소켓 통신 리소스 초과	동시에 실행 가능한 소켓 서비스 명령 리소스를 초과하여 명령이 실행되었습니다.	• 동시에 32개를 넘는 소켓 서비스 명령이 실행됨 • 동시에 30개(CPU 유니트 Ver.1.02 이하는 16개)를 넘는 소켓 핸들을 사용하려고 함				○		🕮 A-77페이지
54012400 Hex (Ver.1.02)	실행 권한 없음	실행 불가능한 상태에서 EtherNet/IP 포트의 설정을 변경하는 명령을 실행했습니다.	• 내장 EtherNet/IP 포트의 재시작 중에 내장 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령을 실행함 • CJ 시리즈 EtherNet/IP 유니트의 재시작 중에 해당 유니트의 설정을 변경하는 명령을 실행함 • 명령 또는 CIP 메시지를 사용한 내장 EtherNet/IP 포트의 설정 변경 중에 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령을 실행함 • 명령 또는 CIP 메시지를 사용한 CJ 시리즈 EtherNet/IP 유니트의 설정 변경 중에 해당 유니트의 설정을 변경하는 명령을 실행함 • 명령으로 지정된 호기 번호가 내장 EtherNet/IP 포트, CJ 시리즈 EtherNet/IP 유니트가 아님				○		🕮 A-78페이지
54012401 Hex (Ver.1.02)	설정 반영 실패	CJ 시리즈 EtherNet/IP 유니트에서 변경된 설정을 반영하지 못했습니다.	• CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령의 실행 중에 해당 유니트 또는 내장 EtherNet/IP 포트가 재시작됨				○		🕮 A-79페이지
54012402 Hex (Ver.1.02)	명령 동시 실행 수 초과	컨트롤러 통신 설정 명령이 동시에 실행 가능한 수를 초과하여 실행되었습니다.	• 동시에 2개 이상의 컨트롤러 통신 설정 명령이 실행됨				○		🕮 A-79페이지
54012C00 Hex (Ver.1.05)	NX 메시지 이상	NX 메시지로 에러 응답이 반환되었습니다.	• 이상의 내용에 따름				○		🕮 A-80페이지
54012C01 Hex (Ver.1.05)	NX 메시지 리소스 초과	동시에 실행 가능한 NX 메시지 명령의 리소스를 초과하여 명령이 실행되었습니다.	• 동시에 32개를 넘는 NX 메시지 명령이 실행됨				○		🕮 A-80페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54012C02Hex (Ver.1.05)	NX 메시지 타임아웃	NX 메시지의 실행 중에 타임아웃이 발생했습니다.	• 지정된 NX 유니트가 존재하지 않음 • NX 메시지가 타임아웃되어서 종료됨 • 상대 유니트의 전원 OFF • 상대 유니트의 통신 정지 • 통신 케이블의 커넥터 연결 불량 • 통신 케이블의 단선 • 노이즈				○		A-81페이지
54012C03Hex (Ver.1.05)	NX 메시지 길이 오류	NX 메시지 길이가 잘못되었습니다.	• WriteDat 또는 Path로 지정한 크기가 너무 김				○		A-81페이지
54012C05Hex (Ver.1.05)	NX 메시지 네트워크 이상 (EtherCAT)	NX 메시지 경로상의 EtherCAT 통신에서 이상이 발생했습니다.	• NX 메시지의 경로 중 EtherCAT 통신에서 이상이 발생				○		A-82페이지
54012C06Hex (Ver.1.05)	지정 유니트 외부 재시작 이미 실행	명령 실행 시, 이미 Sysmac Studio에서 재시작이 실행되었습니다.	• 명령 실행 시, 이미 Sysmac Studio에서 재시작이 실행됨				○		A-82페이지
54012C07Hex (Ver.1.05)	명령 대상 이외의 유니트 지정	지정된 유니트의 슬레이브 노드 주소에 해당 명령의 대상이 아닌 슬레이브가 연결되어 있습니다.	• 지정된 유니트의 슬레이브 노드 주소에 해당 명령의 대상이 아닌 슬레이브가 연결되어 있음				○		A-83페이지
54013461Hex	프로세스 데이터 개체 설정 부족	PDO 매핑이 정확하지 않습니다.	• 모션 명령에서 필요한 PDO 매핑이 되어 있지 않음 • 해당 명령에 대응하는 개체가 없는 대상 기기에 대해 해당 명령을 실행함 • 오므론 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에 대해 트리거 조건으로 Z 상(_mcEncoderMark)을 지정하여 모션 명령을 기동함				○		A-79페이지
54015420Hex	전자 기어 분자 설정 범위 밖	모션 제어 명령의 입력 변수 'RatioNumerator'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-85페이지
54015421Hex	전자 기어 분모 설정 범위 밖	모션 제어 명령의 입력 변수 'RatioDenominator'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-85페이지
54015422Hex	목표 속도 설정 범위 밖	모션 제어 명령의 입력 변수 'Velocity'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-86페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015423 Hex	가속도 설정 범위 밖	모션 제어 명령의 입력 변수 'Acceleration'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-86페이지
54015424 Hex	감속도 설정 범위 밖	모션 제어 명령의 입력 변수 'Deceleration'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-87페이지
54015425 Hex	저크 설정 범위 밖	모션 제어 명령의 입력 변수 'Jerk'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-87페이지
54015427 Hex	토크 경사 설정 범위 밖	모션 제어 명령의 입력 변수 'TorqueRamp'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-88페이지
54015428 Hex	주축 계수 설정 범위 밖	모션 제어 명령의 입력 변수 'MasterScaling'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-88페이지
54015429 Hex	종축 계수 설정 범위 밖	모션 제어 명령의 입력 변수 'SlaveScaling'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-89페이지
5401542A Hex	피딩 속도 설정 범위 밖	모션 제어 명령의 입력 변수 'FeedVelocity'에 지정된 파라메터가 범위 밖입니다.	• 피딩 속도(입력 변수 'FeedVelocity')가 초기값 그대로(0)임				○		📖 A-89페이지
5401542B Hex	버퍼 모드 선택 범위 밖	모션 제어 명령의 입력 변수 'BufferMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-90페이지
5401542C Hex	좌표계 선택 범위 밖	모션 제어 명령의 입력 변수 'CoordSystem'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-90페이지
5401542D Hex	원호 보간 모드 선택 범위 밖	모션 제어 명령의 입력 변수 'CircMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-91페이지
5401542E Hex	방향 선택 범위 밖	모션 제어 명령의 입력 변수 'Direction'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		📖 A-91페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401542F Hex	경로 선택 범위 밖	모션 제어 명령의 입력 변수 'PathChoice'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		🕮 A-92페이지
54015430 Hex	위치 종류 선택 범위 밖	모션 제어 명령의 입력 변수 'ReferenceType'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		🕮 A-92페이지
54015431 Hex	이동 방법 선택 범위 밖	모션 제어 명령의 입력 변수 'MoveMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		🕮 A-93페이지
54015432 Hex	트랜지션 모드 선택 범위 밖	모션 제어 명령의 입력 변수 'TransitionMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖 • 'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하고 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정				○		🕮 A-93페이지
54015433 Hex	계속 방법 선택 범위 밖	모션 제어 명령의 입력 변수 'Continuous(Reserved)'의 값이 변경되었습니다.	• 입력 변수 'Continuous(Reserved)'의 값이 변경됨				○		🕮 A-94페이지
54015434 Hex	가감산 방법 선택 범위 밖	모션 제어 명령의 입력 변수 'CombineMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		🕮 A-94페이지
54015435 Hex	동기 시작 조건 지정 범위 밖	모션 제어 명령의 입력 변수 'LinkOption'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		🕮 A-95페이지
54015436 Hex	주축과 종축 동일	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 동일				○		🕮 A-95페이지
54015437 Hex	주축과 보조축 동일	모션 제어 명령의 입력 변수 'Master'와 'Auxiliary'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Master'와 'Auxiliary'의 파라메터가 동일				○		🕮 A-96페이지
54015438 Hex	주축/종축 축 번호가 오름차순이 아님	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축 번호가 오름차순이 아닙니다.	• 명령의 입력 변수 'ReferenceType'에 '_mcLatestCommand' 지정 시, 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 오름차순이 아님				○		🕮 A-96페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015439 Hex	잘못된 캠 테이블 지정	모션 제어 명령의 입력 변수 'CamTable'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 변수 'CamTable'에 캠 데이터 변수 이외를 지정				○		🕮 A-97페이지
5401543A Hex	동기 정지 중	모션 제어의 동기 제어 명령이 실행되었지만 실행 가능한 조건이 아닙니다.	• MC_CamOut(캠 동작 해제) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령이 실행 중이 아님 • MC_GearOut(기어 동작 해제) 명령이 실행되었지만 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령이 실행 중이 아님 • MC_Phasing(주축 상대값 위상 보정) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령, MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령이 실행 중이 아님				○		🕮 A-97페이지
5401543B Hex	모션 명령 재기동 불가능	재기동할 수 없는 모션 제어 명령의 재기동을 수행했습니다.	• 재기동할 수 없는 모션 제어 명령의 재기동				○		🕮 A-98페이지
5401543C Hex	모션 명령 다중 기동 불가능	동일한 대상(MC 공통/축/축 그룹)에 대해 동시에 실행할 수 없는 기능을 여러 개 실행했습니다.	• 동일한 대상(MC 공통/축)에 대해 동시에 실행할 수 없는 기능을 여러 개 실행				○		🕮 A-98페이지
5401543D Hex	축 종류 부적합	엔코더 축에 대해 동작 명령을 실행했습니다.	• 엔코더 축에 대해 동작 명령을 실행				○		🕮 A-99페이지
5401543E Hex	다축 협조 동작 중 명령 기동 불가능	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행했습니다.	• 다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행				○		🕮 A-99페이지
5401543F Hex	축 그룹 무효 상태에서 다축 협조 명령 기동	축 그룹 무효 상태의 축 그룹에 대해 다축 협조 명령을 기동했습니다.	• 축 그룹 무효 상태의 축 그룹에 대해 다축 협조 명령을 기동				○		🕮 A-100페이지
54015440 Hex	축 그룹 유효 불가능	MC_GroupEnable(축 그룹 유효) 명령 실행에 실패했습니다.	• MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성축에 정지 중이 아닌 축이 있음 • MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성축에 MC_TouchProbe(외부 래치 유효) 명령 실행 중인 축이 있음				○		🕮 A-100페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015441Hex	운전 불가능(서보 OFF) 축 동작 지시	서보 OFF 중인 축에 대해 동작 명령을 실행했습니다.	• 서보 OFF 중인 축에 대해 동작 명령을 실행 • MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 통해 원점 사전 설정을 EtherCAT 프로세스 데이터 통신이 확립되지 않은 축에 대해 실행함				○		A-101페이지
54015442Hex	구성축 강제 정지 중 에러	구성축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행했습니다.	• 구성축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행함				○		A-101페이지
54015443Hex	모션 명령 다중 기동 수 초과	버퍼 모드인 Buffered, Blending을 통해 버퍼링되는 모션 제어 명령의 버퍼링 수가 한계를 넘었습니다.	• 축 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 넘음 • 축 그룹 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 넘음				○		A-102페이지
54015444Hex	이동량 부족	위치 결정 명령의 다중 기동/재기동 시, 지정된 감속도 또는 가속도로 지정된 동작을 실행할 수 없습니다.	• '가감속 초과'를 '이상으로 정지'로 설정한 경우, 위치 결정 명령의 다중 기동/재기동 시 지정된 감속도/가속도로 목표 위치에서 정지 불가능				○		A-102페이지
54015445Hex	블렌딩 중계 속도에 도달하기 위한 이동량 부족	중계 속도로 가감속하기 위한 이동량이 부족합니다.	• '가감속 초과'에 '이상으로 정지'가 설정된 경우 현재 명령을 중계 속도로 가감속하기 위한 이동량이 부족함				○		A-103페이지
54015446Hex	사다리꼴 패턴 캠 등속 이동량 부족	주축의 등속 이동량이 '0' 미만입니다.	• MC_MoveLink(사다리꼴 패턴 캠) 명령에서 주축의 등속 이동량이 '0' 미만				○		A-103페이지
54015447Hex	위치 지정 기어 동작 목표 속도 부족	MC_GearInPos(위치 지정 기어 동작) 명령에서 종축의 '목표 속도'가 작기 때문에 필요한 속도에 도달하지 못했습니다.	• MC_GearInPos(위치 지정 기어 동작) 명령에서 입력 변수 'Velocity(목표 속도)'의 값이 (명령 기동 시의 주축 속도x기어비)보다 작음				○		A-104페이지
54015448Hex	원호 보간 시작점과 종료점이 동일	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식 지정 시 시작점과 종료점이 동일한 위치입니다. 또는 통과점 지정 방식 지정 시 시작점과 종료점 및 통과점이 동일한 위치입니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식이 지정되고 시작점과 종료점의 위치가 동일 • MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식이 지정되고 시작점과 종료점 및 통과점의 위치가 동일				○		A-104페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015449 Hex	원호 보간 중심점 지정 위치 범위 밖	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식 지정 시 중심점의 위치 지정이 허용 범위를 벗어났습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식이 지정되고 시작점과 중심점의 거리 및 종료점과 중심점의 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위를 벗어남				○		A-105페이지
5401544A Hex	카운트 모드 설정에 의한 명령 기동 이상	카운트 모드가 로터리 모드로 설정되어 있는 축에 대해 로터리 모드에서 사용할 수 없는 명령을 실행했습니다.	• 카운트 모드가 로터리 모드로 설정되어 있는 축을 로터리 모드에서 사용할 수 없는 명령에서 사용				○		A-105페이지
5401544C Hex	파라메터 선택 범위 밖	모션 제어 명령의 입력 변수 'ParameterNumber'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-106페이지
5401544D Hex	정지 방법 선택 범위 밖	모션 제어 명령의 입력 변수 'StopMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-106페이지
5401544E Hex	트리거 입력 조건의 래치 ID 선택 범위 밖	모션 제어 명령의 입력 변수 'TriggerInput::LatchID'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-107페이지
5401544F Hex	MC 설정 쓰기의 설정 범위 밖	모션 제어 명령의 입력 변수 'SettingValue'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖 • 파라메터 지정과 설정값의 데이터 형식이 일치하지 않음				○		A-107페이지
54015450 Hex	트리거 입력 조건의 모드 선택 범위 밖	모션 제어 명령의 입력 변수 'TriggerInput::Mode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-108페이지
54015451 Hex	트리거 입력 조건의 드라이브 트리거 입력 신호 선택 범위 밖	모션 제어 명령의 입력 변수 'TriggerInput::InputDrive'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-108페이지
54015453 Hex	모션 명령 재기동 불가능(축 지정)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Axis'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		A-109페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015454 Hex	모션 명령 재기동 불가능(버퍼 모드 선택)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'BufferMode'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-109페이지
54015455 Hex	모션 명령 재기동 불가능(방향 선택)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Direction'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-110페이지
54015456 Hex	모션 명령 재기동 불가능(반복 모드)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Periodic'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-110페이지
54015457 Hex	모션 명령 재기동 불가능(축 그룹 지정)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'AxesGroup'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-111페이지
54015458 Hex	모션 명령 재기동 불가능(약동 설정)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Jerk'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-111페이지
54015459 Hex	모션 명령 재기동 불가능(주축)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Master'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-112페이지
5401545A Hex	모션 명령 재기동 불가능(Master Offset)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MasterOffset'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-112페이지
5401545B Hex	모션 명령 재기동 불가능(Master Scaling)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MasterScaling'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-113페이지
5401545C Hex	모션 명령 재기동 불가능(Master Start Distance)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MasterStart Distance'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-113페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401545D Hex	모션 명령 재기동 불가능(Continuous)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Continuous'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-114페이지
5401545E Hex	모션 명령 재기동 불가능(Move Mode)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MoveMode'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-114페이지
5401545F Hex	잘못된 보조축 지정	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않습니다.	• 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않는 변수				○		📖 A-115페이지
54015460 Hex	잘못된 축 지정	모션 제어 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않습니다.	• 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않는 변수				○		📖 A-115페이지
54015461 Hex	잘못된 축 그룹 지정	모션 제어 명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지않거나사용 축 그룹이 아닙니다.	• 명령의 'AxesGroup'에 지정된 축 그룹이 존재하지 않는 변수 • 명령의 'AxesGroup'에 지정된 축 그룹이 사용 축 그룹으로 설정되어 있지 않음				○		📖 A-116페이지
54015462 Hex	잘못된 주축 지정	모션 제어 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않거나 동기 주축이 아닙니다.	• 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않는 변수 • MC_Phasing(주축 상대값 위상 보정) 명령의 경우, 입력 변수 'Master'에 지정된 축이 동기 주축이 아님				○		📖 A-116페이지
54015463 Hex	모션 명령 재기동 불가능(Slave Offset)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'SlaveOffset'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-117페이지
54015464 Hex	모션 명령 재기동 불가능(Slave Scaling)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'SlaveScaling'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-117페이지
54015465 Hex	모션 명령 재기동 불가능(Start Position)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'StartPosition'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-118페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015466Hex	원점 미확정 상태에서 명령 기동 이상	원점이 확정되지 않은 상태에서 고속 원점 복귀 또는 보간 명령이 실행되었습니다.	• 원점이 확정되지 않은 상태에서 고속 원점 복귀가 실행됨 • 원점 미확정 상태를 포함한 구성축에 대해 보간 명령이 실행됨				○		A-118페이지
54015467Hex	모션 명령 재기동 불가능(위치 종류)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'ReferenceType'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		A-119페이지
54015468Hex	사용하지 않는 축 지정(주축)	모션 제어 명령으로 지정된 주축이 사용하지 않는 축입니다.	• 모션 제어 명령으로 지정된 주축이 사용하지 않는 축				○		A-119페이지
54015469Hex	시작 위치 설정 범위 밖	모션 제어 명령의 입력 변수 'FirstPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-120페이지
5401546AHex	종료 위치 설정 범위 밖	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖				○		A-120페이지
5401546BHex	잘못된 시작 위치/종료 위치 · 크기 관계(리니어 모드)	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 입력 변수 'FirstPosition'에 지정된 파라메터보다 작은 값입니다.	• 카운트 모드가 리니어 모드일 때 명령의 입력 파라메터에서 'LastPosition'의 값이 'FirstPosition'의 값보다 작음				○		A-121페이지
5401546CHex	주축 동기 위치 설정 범위 외	모션 제어 명령의 입력 변수 'MasterSyncPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-121페이지
5401546DHex	종축 동기 위치 설정 범위 외	모션 제어 명령의 입력 변수 'SlaveSyncPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-122페이지
5401546EHex	트리거 입력 조건의 래치 ID 중복	모션 제어 명령으로 지정된 래치 ID가 여러 명령에서 중복되어 있습니다.	• MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동일한 래치 ID가 동시에 사용됨 • MC_TouchProbe(외부 래치 유효) 명령 이외에서 사용 중인 래치를 MC_AbortTrigger(외부 래치 무효) 명령에서 중지하려고 함				○		A-122페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401546F Hex	저크 오버라이드값범위 외	모션 제어 명령의 입력 변수 'JerkFactor'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		📖 A-123페이지
54015470 Hex	가감속 속도 오버라이드 값 범위 외	모션 제어 명령의 입력 변수 'AccFactor'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		📖 A-123페이지
54015471 Hex	시작 위치 방식 지정 범위 외	모션 제어 명령의 입력 변수 'StartMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		📖 A-124페이지
54015472 Hex	모션 명령 재기동 불가능(시작 위치 방식)	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'StartMode'의 파라메터를 변경했습니다.	• 명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함				○		📖 A-124페이지
54015474 Hex	사용하지 않는 축 지정(보조축)	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축입니다.	• 명령의 'Auxiliary'에 지정된 축이 사용하지 않는 축				○		📖 A-125페이지
54015475 Hex	위치 지정 기어 지정값 이상	모션 제어 명령에 입력된 속도/가속도/감속도에서는 동기 동작을 할 수 없습니다.	• 명령에 입력된 속도/가속도/감속도에서는 지정된 동기 동작을 수행할 수 없음				○		📖 A-125페이지
54015476 Hex	위치 지정 기어 주축 속도 제로	모션 제어 명령의 기동 시 주축의 속도가 '0'입니다.	• 명령의 기동 시 주축의 속도가 '0'				○		📖 A-126페이지
54015478 Hex	목표 위치 설정 범위 외	모션 제어 명령의 입력 변수 'Position'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외 • 로터리 모드의 축에서 목표 위치가 링 설정의 범위 외				○		📖 A-126페이지
54015479 Hex	이동 거리 범위 외	모션 제어 명령의 입력 변수 'Distance'에 지정된 파라메터가 범위 밖 또는 'Distance' 값을 가산한 목표 위치가 범위 밖입니다.	• 명령의 입력 파라메터의 절대값을 펄스 단위로 변환한 경우 40비트의 범위를 초과함 • 리니어 모드의 축에서 이동 거리를 가산한 목표 위치를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함				○		📖 A-127페이지
5401547A Hex	캠 테이블 시작점 위치 설정 범위 외	모션 제어 명령의 입력 변수 'StartPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		📖 A-127페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401547B Hex	캠 동작(주축 추종) 시작 위치 설정 범위 외	모션 제어 명령의 입력 변수 'MasterStartDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-128페이지
5401547C Hex	원호 보간 반경 지정 이상	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식 지정 시 지정된 반경으로 원호의 궤적을 작성할 수 없습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식이 지정되고, 지정된 반경으로 원호의 궤적을 작성할 수 없음				○		A-128페이지
5401547D Hex	원호 보간 반경 오버플로우	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반경이 최대값을 벗어났습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반경을 펄스 단위로 변환한 경우 40비트의 범위를 벗어남				○		A-129페이지
5401547E Hex	원호축지정 범위 외	모션 제어 명령의 입력 변수 'CircAxes'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외 • 'CircAxes'에 지정된 축이 축 그룹 설정의 구성축에 포함되지 않음 • 'CircAxes'의 2축에 동일한 축을 지정함				○		A-129페이지
5401547F Hex	보조축/종축 · 축 번호가 오름차순이 아님	모션 제어 명령으로 지정한 입력 변수 'Auxiliary'와 'Slave'의 파라메터 값이 오름차순으로 정렬되지 않았습니다.	• 명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 오름차순이 아님				○		A-130페이지
54015480 Hex	캠 테이블 속성갱신 중 데이터 오름차순 이상	유효한 데이터 수 계산 중에 위상이 오름차순이 아닌 것으로 판명되었습니다. 또는 계산한 결과, 유효 데이터 수가 '0'인 것으로 판명되었습니다.	• 유효한 데이터 수 계산 중에 위상이 오름차순이 아닌 것으로 판명됨 • 계산한 결과, 유효 데이터 수가 '0'인 것으로 판명됨				○		A-130페이지
54015481 Hex	MC 설정 쓰기의 대상 범위 외	모션 제어 명령의 입력 변수 'Target'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-131페이지
54015482 Hex	주축 이동거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-131페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015483 Hex	주축 가속 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistanceInACC'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-132페이지
54015484 Hex	주축 감속 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistanceInDEC'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-132페이지
54015487 Hex	실행 모드 선택 범위 외	모션 제어 명령의 입력 변수 'ExecutionMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-133페이지
54015488 Hex	축간 편차 허용값 범위 외	모션 제어 명령의 입력 변수 'PermittedDeviation'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-133페이지
54015489 Hex	통과점 위치/중심 위치/반경 지정 범위 외	모션 제어 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 범위 밖입니다.	• 통과점 지정 또는 중심 지정의 경우, 'AuxPoint' 값을 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함 • 반경 지정의 경우 'AuxPoint[0]'의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함				○		🕮 A-134페이지
5401548A Hex	종료점 지정 범위 외	모션 제어 명령의 입력 변수 'EndPoint'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함				○		🕮 A-134페이지
5401548B Hex	종축 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'SlaveDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터를 펄스 단위로 변환한 경우 40비트의 범위를 초과함				○		🕮 A-135페이지
5401548C Hex	위상 보정량 범위 외	모션 제어 명령의 입력 변수 'PhaseShift'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터의 절대값을 펄스 단위로 변환한 경우 40비트의 범위를 초과함				○		🕮 A-135페이지
5401548D Hex	피딩 거리 범위 외	모션 제어 명령의 입력 변수 'FeedDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터의 절대값을 펄스 단위로 변환한 경우 40비트의 범위를 초과함				○		🕮 A-136페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401548E Hex	보조축/종축 동일	모션 제어 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 동일				○		A-136페이지
5401548F Hex	상대 위치 선택 범위 외	모션 제어 명령의 입력 변수 'Relative'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-137페이지
54015490 Hex	캠 트랜지션 지정 선택 범위 외	모션 제어 명령의 입력 변수 'CamTransition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-137페이지
54015491 Hex	동기 제어 해제 모드 선택 범위 외	모션 제어 명령의 입력 변수 'OutMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-138페이지
54015492 Hex	외부 래치 유효 명령 실행 불가능	CSP 모드 이외의 제어 모드일 때 입력 변수 'StopMode'에 '_mcImmediateStop (즉시 정지)'을 지정하여 드라이버 모드의 MC_TouchProbe (외부 래치 유효) 명령을 실행했습니다.	• CSP 모드 이외의 제어 모드일 때 또는 서보 OFF일 때 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정하여 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행함				○		A-138페이지
54015493 Hex	주축 오프셋 설정 범위 외	모션 제어 명령의 입력 변수 'MasterOffset'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함				○		A-139페이지
54015494 Hex	종축 오프셋 설정 범위 외	모션 제어 명령의 입력 변수 'SlaveOffset'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함				○		A-139페이지
54015495 Hex	명령 현재 위치 카운트 선택 범위 외	모션 제어 명령의 입력 변수 'CmdPosMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-140페이지
54015496 Hex	주축 기어비 분자 범위 외	모션 제어 명령의 입력 변수 'RatioNumeratorMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		A-140페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015497 Hex	주축 기어비 분모 범위 외	모션 제어 명령의 입력 변수 'RatioDenominatorMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-141페이지
54015498 Hex	보조축 기어비 분자범위 외	모션 제어 명령의 입력 변수 'RatioNumeratorAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-141페이지
54015499 Hex	보조축 기어비 분모범위 외	모션 제어 명령의 입력 변수 'RatioDenominatorAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-142페이지
5401549A Hex	주축 위치 종류 선택 범위 외	모션 제어 명령의 입력 변수 'ReferenceTypeMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-142페이지
5401549B Hex	보조축 위치 종류 선택 범위 외	모션 제어 명령의 입력 변수 'ReferenceTypeAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-143페이지
5401549C Hex	목표 위치 링 카운터 범위 외	실행된 명령의 목표 위치가 링 카운터의 범위 밖이므로 동작할 수 없습니다.	• 링 카운터 범위에 0이 포함되지 않는 설정일 때 고속 원점 복귀가 실행됨				○		🕮 A-143페이지
5401549D Hex (Ver.1.01)	축 그룹 구성축 설정 범위 외	모션 제어 명령의 입력 변수 'Axes'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-144페이지
5401549E Hex (Ver.1.04)	축 사용 설정 범위 밖	모션 제어 명령의 입력 변수 'AxisUse'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖						🕮 A-144페이지
54015700 Hex (Ver.1.03)	원점 복귀 파라메터 설정 범위 외	모션 제어 명령의 입력 변수 'HomingParameter'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외				○		🕮 A-145페이지
54015702 Hex (Ver.1.04)	축 미사용 전환 이상	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.	• 정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함						🕮 A-145페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015703Hex (Ver.1.06)	축 사용 전환 불가	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.	• 사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함				○		🕮 A-146페이지
54015720Hex (Ver.1.04)	축 사용 전환 시 모션 제어 파라메터 설정 이상	사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못되었습니다.	• MC_ChangeAxisUse(축 사용 변경) 명령을 사용하여 미사용 축에서 사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못됨 • 모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨 • 비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함				○		🕮 A-147페이지
54015721Hex (Ver.1.04)	축 사용 전환 시 필수 프로세스 데이터 개체 미설정	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 설정되어 있지 않습니다.	• 사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 PDO 매핑 설정에 설정되어 있지 않음 • 모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨 • 비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함 • [축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령이 실행됨				○		🕮 A-148페이지
54015722Hex (Ver.1.06)	피드백 위치 오버 플로우/언더 플로우 발생 중	피드백 위치 오버 플로우/언더 플로우 발생 중에 실행할 수 없는 명령을 기동했습니다.	• 피드백 위치 오버 플로우 또는 언더 플로우 발생 중에 실행할 수 없는 명령을 기동함				○		🕮 A-149페이지
54015723Hex (Ver.1.06)	스위치 구조체의 트랙 번호 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-149페이지
54015724Hex (Ver.1.06)	스위치 구조체의 ON 시작 위치 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'FirstOnPosition'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-150페이지
54015725Hex (Ver.1.06)	스위치 구조체의 ON 종료 위치 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'LastOnPosition'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-150페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54015726Hex (Ver.1.06)	스위치 구조체의 방향 선택 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'AxisDirection'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-151페이지
54015727Hex (Ver.1.06)	스위치 구조체의 스위치 모드 선택 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'CamSwitchMode'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-151페이지
54015728Hex (Ver.1.06)	스위치 구조체의 ON 시간 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'Duration'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-152페이지
54015729Hex (Ver.1.06)	트랙 옵션 구조체의 ON 시각 보정 설정 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OnCompensation'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-152페이지
5401572AHex (Ver.1.06)	트랙 옵션 구조체의 OFF 시각 보정 설정 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OffCompensation'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-153페이지
5,401,572BHex (Ver.1.06)	스위치 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-153페이지
5401572CHex (Ver.1.06)	출력 신호 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'Outputs'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-154페이지
5401572DHex (Ver.1.06)	트랙 옵션 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		🕮 A-154페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
5401572EHex (Ver.1.06)	출력 신호와 트랙 옵션의 배열 요소 수 불일치	모션 제어 명령의 입출력 변수 'Outputs'와 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 일치하지 않습니다.	• '명령의 입출력 변수로 지정한 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하지 않음				○		A-155페이지
5401572FHex (Ver.1.06)	모션 명령 다중 기동 불가능(주축)	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경했습니다.	• '명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경				○		A-156페이지
54015730Hex (Ver.1.06)	모션 명령 다중 기동 불가능(위치 종류 선택)	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경했습니다.	• '명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경				○		A-156페이지
54015731Hex (Ver.1.06)	스위치 구조체의 동일 트랙 지정 수 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 동일 트랙 번호 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 스위치 구조체 형식 변수의 'TrackNumber'에 지정된 동일 트랙 번호의 수가 1트랙에 지정 가능한 개수의 범위 밖				○		A-157페이지
54016440Hex	목표 위치 정방향 소프트웨어 리미트 초과	지정한 위치가 정방향 소프트웨어 리미트의 범위 밖입니다.	• 명령의 입력 변수 'Position'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 밖 • 시작 위치가 정방향 소프트웨어 리미트 범위 밖일 때 소프트웨어 리미트의 범위와 반대 방향으로 동작을 지정한 명령 실행 • 통과점을 지정한 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 외				○		A-158페이지
54016441Hex	목표 위치 역방향 소프트웨어 리미트 초과	지정한 위치가 역방향 소프트웨어 리미트의 범위 밖입니다.	• 명령의 입력 변수 'Position'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 밖 • 시작 위치가 역방향 소프트웨어 리미트 범위 밖일 때 소프트웨어 리미트의 범위와 반대 방향으로 동작을 지정한 명령 실행 • 통과점을 지정한 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 외				○		A-159페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조대상
				전체	부분	가벼움	감시	일반	
54016442 Hex	명령 위치 오버 플로우/언더 플로우 발생 중	명령 위치 오버 플로우/언더 플로우 발생 중에 위치 결정, 오버 플로우/언더 플로우 방향의 명령 또는 방향을 지정할 수 없는 명령을 실행했습니다.	• 명령 위치 오버 플로우/언더 플로우 발생 중에 다음을 수행함 • 위치 결정 명령을 실행 • 오버 플로우/언더 플로우 방향의 연속 제어 명령을 실행 • 방향을 확인할 수 없는 명령(동기 기능, 토크 제어)을 실행				○		A-159페이지
54016443 Hex	정방향 한계 입력 중	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했습니다.	• 정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행 또는 정방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않은 명령을 실행. 정방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행				○		A-160페이지
54016444 Hex	역방향 한계 입력 중	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했습니다.	• 역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행 또는 역방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않은 명령을 실행. 역방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행				○		A-160페이지
54017422 Hex	서보 주회로 전원 OFF 상태	서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행했습니다.	• 서보 드라이버의 주회로 전원 'OFF' 상태에서 서보 ON 실행				○		A-161페이지

A-3 에러 코드에 대한 자세한 내용

명령에서 발생하는 이상(이벤트 코드)에 대한 자세한 정보가 아래에 나와 있습니다. 이벤트 코드의 하위 4자리가 명령의 에러 코드를 나타냅니다. 각 에러 코드의 설명은 대응하는 이벤트 코드의 설명을 참조하십시오. 예를 들어, 명령의 에러 코드가 16#0400인 경우는 54010400Hex의 이벤트 코드 설명을 참조하십시오.

표 보는 방법

각 이상에 대한 설명에서 사용하는 표의 각 항목의 의미를 [ ] 안에 표시합니다.

이벤트명	[이상(이벤트)의 명칭입니다.]			**이벤트 코드**	[이상(이벤트)의 코드입니다.]	
내용	[이상(이벤트)의 내용입니다.]					
발생원	[이상(이벤트)이 발생한 부분입니다.]		**발생원 상세 정보**	[발생원에 대한 자세한 내용입니다.]	**검출 타이밍**	[이상 검출의 타이밍입니다.]
이상의 속성	**중요도**	[제어에 대한 영향의 크기입니다.](*1)	**복구 방법**	[복구 방법입니다.](*2)	**로그 카테고리**	[저장되는 로그의 종류입니다.](*3)
발생의 영향	**사용자 프로그램**	[사용자 프로그램의 실행 상태입니다.](*4)	**동작**	[이상(이벤트) 발생 시의 동작에 관한 특기 사항입니다.]		
LED	내장 EtherNet/IP 포트용 LED, 내장 EtherCAT 포트용 LED의 표시 상태입니다. 발생원이 EtherCAT 마스터 기능 모듈, EtherNet/IP 기능 모듈일 때만 기재합니다.]					
시스템 정의 변수	**변수명**		**데이터 형식**		**명칭**	
	[이상을 감지하는 시스템 정의 변수, 이상의 영향을 받는 시스템 정의 변수, 이상의 원인이 되는 시스템 정의 변수의 변수명과 데이터 형식, 명칭입니다.]					
발생 요인과 처리	**발생 요인(추정 원인)**		**처리 대책**		**재발 방지**	
	[이상(이벤트) 발생 요인과 처리 대책, 재발 방지 방법입니다.]					
부속 정보	[Sysmac Studio/NS 시리즈 표시기에 표시되는 부속 정보의 내용입니다.]					
주의점/비고	[그 밖의 주의점, 제약 사항, 보충 설명 등입니다.]					

*1 다음 중 하나입니다.
전체 정지 이상: 전체 정지 이상 레벨
부분 정지 이상: 부분 정지 이상 레벨
가벼운 이상: 가벼운 이상 레벨
감시 정보
일반 정보

*2 다음 중 하나입니다.
자동 복구: 요인이 제거되면 자동으로 정상 상태로 돌아갑니다.
이상 해제: 요인이 제거된 후 이상 해제를 실행하면 정상 상태로 돌아갑니다.
전원 다시 켜기: 요인이 제거된 후 컨트롤러의 전원을 다시 켜면 정상 상태로 돌아갑니다.
컨트롤러 리셋: 요인이 제거된 후 컨트롤러를 리셋하면 정상 상태로 돌아갑니다.
발생 요인에 따름: 발생 요인에 따라 달라집니다.

*3 다음 중 하나입니다.
시스템: 시스템 이벤트 로그
액세스: 액세스 이벤트 로그

*4 다음 중 하나입니다.
계속: 사용자 프로그램을 계속 실행합니다.
정지: 사용자 프로그램의 실행을 정지합니다.
시작: 사용자 프로그램의 실행을 시작합니다.

이벤트명	입력값 범위 밖			이벤트 코드	54010400Hex	
내용	명령의 입력 파라메터가 입력 변수의 범위를 초과했습니다. 또는 정수 0으로 제산/나머지 연산을 수행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		−		−	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위를 초과함. 또는 정수 0으로 제산/나머지 연산을 수행함		해당 명령의 입력 변수의 범위를 확인해서 입력 파라메터가 범위를 초과하지 않도록, 또는 정수 0으로 제산/나머지 연산을 하지 않도록 하십시오.		명령에 대한 입력 파라메터의 값이 입력 범위를 초과하지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	수치 관계 오류			이벤트 코드	54010401Hex	
내용	명령의 입력 파라메터 값의 관계가 조건을 충족하지 않습니다. 또는 명령 실행 연산 중/결과의 수치가 조건을 충족하지 못했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		−		−	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	입력 파라메터 값의 관계가 조건을 충족하지 않음		해당 명령의 입력 변수의 의미와 관계를 확인하고 입력 파라메터 값의 관계가 충족되도록 수정하십시오.		명령의 입력 파라메터 값이 입력 변수의 관계를 충족하도록 하십시오.	
	명령 실행 연산 중/결과의 수치가 조건을 충족하지 않음		해당 명령의 처리를 확인하고 부적절한 연산 결과를 초래하지 않는 입력 파라메터의 값으로 설정하십시오.		명령의 처리를 확인하고 처리 연산에서 이 이상의 발생 요인을 만들지 않는 입력 파라메터로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	부동 소수점 수 에러			이벤트 코드		54010402Hex	
내용	명령의 부동 소수점 수의 입력 파라메터에 대해 수치 이외의 것이 입력되었습니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	없음		–			–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	명령의 부동 소수점 수의 입력 파라메터에 대해 수치 이외의 것이 입력됨		해당 명령의 부동 소수점 수의 입력 파라메터에 수치를 입력하도록 수정하십시오.			명령의 부동 소수점 수의 입력 파라메터를 수치로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	BCD가 아님			이벤트 코드		54010403Hex	
내용	명령의 BCD 입력 파라메터에 대해 BCD 데이터가 아닌 값이 입력되었습니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	없음		–			–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	명령의 BCD 입력 파라메터에 대해 16진수의 A, B, C, D, E, F가 입력됨		해당 명령의 BCD의 입력 파라메터에 BCD 데이터를 입력하도록 수정하십시오			명령의 BCD 입력 파라메터를 BCD 데이터로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	부호가 있는 BCD 에러		이벤트 코드	54010404 Hex	
내용	명령의 부호가 있는 BCD 데이터의 입력 파라메터에 대해 최상위 자리에 잘못된 값이 입력되었습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 부호가 있는 BCD 데이터의 입력 파라메터에 대해 최상위 자리에 잘못된 값이 입력됨 • BCD 형식에 '_BCD0'이 지정된 경우는 최상위 자리가 2~F가 됨 • BCD 형식에 '_BCD2'가 지정된 경우는 최상위 자리가 A, B, C, D, E 중 하나가 됨 • BCD 형식에 '_BCD3'이 지정된 경우는 최상위 자리가 B, C, D, E가 됨		해당 명령의 BCD의 입력 파라메터에 올바른 부호가 있는 BCD 데이터를 입력하도록 수정하십시오		명령의 부호가 있는 BCD 데이터의 입력 파라메터의 최상위 자리에 올바른 값을 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	지정 비트 위치 오류		이벤트 코드	54010405 Hex	
내용	명령으로 지정된 비트 위치가 잘못되었습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령으로 지정된 비트 위치가 지정된 데이터의 범위를 초과함		해당 명령으로 지정되는 비트 위치가 지정된 데이터의 범위에 포함되도록 수정하십시오.		명령으로 지정되는 비트 위치가 지정된 데이터의 범위에 포함되도록 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	영역 범위 외 지정			이벤트 코드		54010406Hex	
내용	명령으로 지정된 데이터의 메모리 주소 또는 데이터 크기가 적절하지 않습니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	없음		–			–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	명령으로 지정된 데이터의 메모리 주소가 유효 범위 밖인 경우, 명령으로 지정된 데이터의 크기가 유효 범위를 초과하는 경우, 변수의 데이터 형식과 데이터 크기가 맞지 않는 경우 등을 생각할 수 있음		해당 명령으로 지정되는 데이터의 메모리 주소나 데이터 크기가 유효 범위 안에 포함되도록 수정하십시오.			명령으로 지정되는 데이터의 메모리 주소나 데이터 크기가 유효 범위 안에 포함되도록 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	영역 초과			이벤트 코드		54010407Hex	
내용	명령의 연산 결과가 출력 파라메터의 데이터 영역의 범위를 초과했습니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	없음		–			–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	배열 요소 수 등 명령의 처리 결과가 출력 파라메터의 데이터 영역의 범위를 초과함		해당 명령의 처리 결과가 출력 파라메터의 데이터 영역 범위에 포함되는 입력 파라메터가 되도록 수정하십시오.			명령의 처리 결과가 출력 파라메터의 데이터 영역 범위에 포함되는 입력 파라메터가 되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	컨트롤러 이상 해제 실패			이벤트 코드	54010409Hex	
내용	컨트롤러 이상이 발생하지 않은 상태에서 컨트롤러 이상 해제 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 출력 및 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	컨트롤러 이상이 발생하지 않은 상태에서 컨트롤러 이상 해제 명령을 실행함		컨트롤러 이상이 발생했을 때 해당 명령을 실행하도록 프로그램을 수정하십시오.		컨트롤러 이상이 발생했을 때 해당 명령을 실행하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	사용자 이상 해제 실패			이벤트 코드	5401040BHex	
내용	사용자 이상이 발생하지 않은 상태에서 사용자 이상 해제 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 출력 및 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	사용자 이상이 발생하지 않은 상태에서 사용자 이상 해제 명령을 실행함		사용자 이상이 발생했을 때 해당 명령을 실행하도록 프로그램을 수정하십시오.		사용자 이상이 발생했을 때 해당 명령을 실행하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	사용자 이상 최대 건수			이벤트 코드	5401040C Hex	
내용	사용자 이상 명령에서 최대 건수를 초과하는 사용자 이상을 발생시키려고 했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 출력 및 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	사용자 이상 명령에서 최대 건수를 초과하는 사용자 이상을 발생시키려고 함		사용자 이상 해제 명령을 실행하십시오. 사용자 이상 건수를 확인하려면 시스템 정의 변수의 사용자 이상 건수를 모니터링합니다.		사용자 이상 명령을 호출하는 조건으로 사용자 이상 건수를 확인하는 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	지정 유니트 오류			이벤트 코드	5401040D Hex	
내용	명령으로 지정된 유니트가 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 출력 및 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	유니트 구성 정보로 존재하지 않는 유니트를 지정함		해당 명령으로 지정되는 호기 번호/유니트 번호가 유니트 구성에 존재하고, 실체도 존재하는 유니트를 지정하도록 수정하십시오.		명령으로 지정되는 호기 번호/유니트 번호가 유니트 구성에 존재하고 실체도 존재하는 유니트를 지정하도록 하십시오.	
	유니트 구성 정보에 존재하지만 실체는 존재하지 않는 유니트를 지정함					
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	유니트 재시작 실패		이벤트 코드	5401040F Hex	
내용	고기능 유니트의 재시작에 실패했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 출력 및 유니트의 동작에는 영향 없음	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	고기능 유니트가 처리 중	잠시 후 고기능 유니트를 재시작하십시오.	고기능 유니트의 Status가 처리 중이 아닌 것을 확인한 후 고기능 유니트를 재시작하도록 프로그램을 변경하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	문자열 형식 이상		이벤트 코드	54010410 Hex	
내용	명령에 입력된 문자열이 올바른 문자열이 아닙니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	문자열을 수치로 변환하는 명령의 입력 문자열이 수치를 나타내는 문자열 또는 양수값을 나타내는 문자열이 아님	해당 명령에 입력되어 있는 문자열이 그 명령에 적합한 문자열이 되도록 수정하십시오	문자열을 수치로 변환하는 명령인 경우, 변환할 입력 문자열이 수치를 나타내도록 하십시오. 양수 조건이 있는 경우는 양수를 나타내도록 하십시오.
	입력 문자열이 NULL 문자로 끝나지 않음	해당 명령에 입력되어 있는 문자열이 NULL 문자로 끝나도록 수정하십시오.	문자열을 수치로 변환하는 명령인 경우, 입력 문자열이 NULL 문자로 끝나도록 하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	지정 프로그램 오류			이벤트 코드		54010411Hex
내용	명령으로 지정된 프로그램이 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	해당 명령으로 지정된 프로그램이 존재하지 않거나 삭제됨		해당 명령으로 지정되는 프로그램이 존재하는 프로그램이 되도록 수정하십시오. 또는 해당 명령으로 지정된 프로그램을 추가하십시오.		명령으로 지정되는 프로그램을 존재하는 프로그램으로 지정하십시오. 또한 사용하는 프로그램을 삭제하지 않도록 주의하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	메모리 이외의 주소 지정			이벤트 코드		54010413Hex
내용	CJ 유니트용 메모리에 지정해야 하는 변수가 CJ 유니트용 메모리에 지정되지 않았습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CJ 유니트용 메모리에 지정해야 하는 변수가 CJ 유니트용 메모리에 AT 지정되지 않음		CJ 유니트용 메모리로 지정해야 하는 변수를 AT 지정으로 CJ 유니트용 메모리에 지정하도록 수정하십시오.		CJ 유니트용 메모리로 지정해야 하는 변수를 AT 지정으로 CJ 유니트용 메모리에 지정하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	스택 언더 플로우		이벤트 코드	54010414Hex	
내용	스택에 데이터가 존재하지 않습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	데이터가 존재하지 않는 스택에서 데이터를 가져오려고 함		스택에 데이터를 저장한 후에 데이터를 가져오도록 프로그램을 수정하십시오.	스택에 데이터를 저장한 후에 데이터를 가져오도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	배열의 요소 수 · 차원 수 오류		이벤트 코드	54010416Hex	
내용	명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위를 초과했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위를 초과함		명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위 안에 포함되도록 수정하십시오.	명령 배열의 입출력 파라메터에 대해 요소 수 · 차원 수가 범위 안에 포함되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	지정 태스크 오류			이벤트 코드	54010417Hex	
내용	명령으로 지정된 태스크가 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	지정된 태스크가 존재하지 않음		존재하는 태스크를 지정하도록 프로그램을 수정하십시오.		존재하는 태스크를 지정하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	지정 태스크가 허가되지 않음			이벤트 코드	54010418Hex	
내용	지정이 허가되지 않은 태스크를 명령으로 지정했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	로컬 태스크를 지정하거나 프라이머리 주기 태스크 및 주기 태스크를 지정함		이벤트 태스크와 로컬 태스크 이외를 지정하도록 프로그램을 수정하십시오.		이벤트 태스크 및 다른 태스크를 지정하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	데이터 형식 오류		이벤트 코드	54010419Hex	
내용	명령에서 사용할 수 없는 형식의 데이터가 입력/입출력으로 지정되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령에서 사용할 수 없는 형식의 데이터를 입력/입출력으로 지정함		해당 명령의 입력/입출력 변수의 형식을 확인하고 입력할 데이터를 사용할 수 있는 형식으로 수정하십시오.	해당 명령의 입력/입출력 변수의 형식을 확인하고 사용할 수 있는 형식으로 데이터를 입력하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	명령 다중 기동		이벤트 코드	5401041AHex	
내용	다중 기동할 수 없는 명령을 다중 기동했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	다중 기동할 수 없는 명령을 여러 개 동시에 실행하려고 함		다중 기동할 수 없는 명령의 경우는 실행 중인 인스턴스가 완료된 후 다른 인스턴스를 실행하도록 프로그램을 수정하십시오.	다중 기동할 수 없는 명령의 경우는 실행 중인 인스턴스가 완료된 후 다른 인스턴스를 실행하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	데이터 용량 초과		이벤트 코드	5401041BHex*1		
내용	명령에 전달된 데이터가 너무 커서 처리할 수 없습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령이 처리할 수 있는 용량보다 큰 데이터가 명령에 전달됨		해당 명령에 전달할 수 있는 데이터 용량 이하 크기의 데이터를 명령에 전달하도록 프로그램을 수정하십시오.		명령이 처리할 수 있는 데이터 용량을 확인하고 그 이하 크기의 데이터를 명령에 전달하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.02에서 추가된 에러 코드(16#041B)입니다.

이벤트명	데이터 크기 불일치		이벤트 코드	5401041CHex*1		
내용	명령의 입력/입출력에서 지정된 데이터가 대상 파라메터의 크기와 일치하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	대상 파라메터의 크기와 일치하지 않는 크기의 데이터를 명령의 입력/입출력에 지정함		대상 파라메터의 크기를 확인하고 입력하는 데이터의 크기가 일치하도록 프로그램을 수정하십시오.		대상 파라메터의 크기를 확인하고 입력하는 데이터의 크기가 일치하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 에러 코드(16#041C)입니다.

이벤트명	명령 동시 실행 리소스 초과		이벤트 코드	5401 041D Hex*1		
내용	동시에 실행 가능한 해당 명령군의 리소스를 초과하여 명령이 실행되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	동시에 실행할 수 있는 개수를 초과하여 해당 명령이 실행됨. 해당 명령군 및 최대 개수는 다음과 같습니다. • DB 연결 명령: 32개		동시에 실행하는 해당 명령이 최대 개수 이하가 되도록 프로그램을 수정하십시오.		동시에 실행하는 해당 명령이 최대 개수 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#041D)입니다.

이벤트명	FINS 이상		이벤트 코드	54010800Hex		
내용	FINS 명령의 송수신에서 이상이 발생했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	FINS 명령의 송수신에서 이상이 발생함		해당 명령의 출력 변수 'ErrorIDEx'의 값을 확인하고, 이 설명서에서 해당 명령의 확장 에러 코드 'ErrorIDEx'의 설명을 참조하십시오.		명령의 'ErrorIDEx'의 설명을 미리 참조하여 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	FINS 포트 사용 불가능		이벤트 코드	54010801Hex	
내용	FINS 포트를 사용 중입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 통신 출력, 유니트의 동작에는 영향 없음	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	FINS 포트 사용 중		입력 조건으로 '_Port_isAvailable'을 a 접점에 삽입하도록 프로그램을 수정하십시오.	입력 조건으로 '_Port_isAvailable'을 a 접점에 삽입하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	시리얼 통신 모드 오류		이벤트 코드	54010C00Hex	
내용	시리얼 커뮤니케이션 유니트가 명령의 실행에 필요한 시리얼 통신 모드가 아닙니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 통신 출력은 명령의 사양에 따르며 유니트의 동작에는 영향 없음	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	시리얼 커뮤니케이션 유니트의 시리얼 통신 포트가 명령에서 전제로 하는 모드로 설정되어 있지 않음		명령 실행에 필요한 시리얼 통신 모드로 설정을 변경하십시오. 또는 설정된 모드에서 실행 가능한 명령만 사용하도록 프로그램을 수정하십시오.	시리얼 커뮤니케이션 유니트를 명령 실행에 필요한 시리얼 통신 모드로 설정해 두십시오. 또는 설정된 모드에서 실행 가능한 명령만 사용하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	포트 설정 변경 다중 기동		이벤트 코드	54010C02Hex	
내용	포트 설정 변경 명령을 실행하는 중에 다른 포트 설정 변경 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 통신 출력은 명령의 사양에 따르며 유니트의 동작은 1번째 포트 설정의 변경 내용에 따릅니다.	

시스템 정의 변수	변수명	데이터 형식	명칭
	_CJB_SCU##1ChgSta	BOOL	시리얼 커뮤니케이션 유니트 ##호기 포트 1용 시리얼 포트 설정 변경 중 플래그
	_CJB_SCU##2ChgSta	BOOL	시리얼 커뮤니케이션 유니트 ##호기 포트 2용 시리얼 포트 설정 변경 중 플래그

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	포트 설정 변경 명령을 실행하는 중에 다른 포트 설정 변경 명령을 실행함	포트 설정 변경 중에는 해당 명령을 실행하지 않도록 프로그램을 수정하십시오.	포트 설정 변경 중에는 해당 명령을 실행하지 않도록 프로그램을 작성하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	SD 메모리 카드 사용 불가능		이벤트 코드	54011400Hex	
내용	명령 실행 시 SD 메모리 카드에 액세스하지 못했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	SD 메모리 카드가 설치되어 있지 않거나 정확하게 삽입되지 않음	SD 메모리 카드를 확실히 설치하십시오.	SD 메모리 카드가 확실히 설치되어 있는지 확인하십시오.
	SD 메모리 카드가 손상됨	정상 작동이 확인된 SD 메모리 카드로 교환하십시오.	없음
	SD 메모리 카드 슬롯이 손상됨	상기와 같은 처리를 해도 이 이상이 재발하는 경우는 CPU 유니트를 교환하십시오.	없음

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	SD 메모리 카드 쓰기 방지		이벤트 코드	54011401Hex	
내용	명령 실행 시 쓰기 방지된 SD 메모리 카드에 쓰려고 했습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	쓰기 방지된 SD 메모리 카드에 쓰려고 함		SD 메모리 카드의 쓰기 방지를 해제하십시오. 해제하려면 SD 메모리 카드 측면의 손잡이를 잠금 위치(LOCK)에서 쓰기 가능한 위치로 이동시킵니다.		SD 메모리 카드에 쓰는 경우에는 쓰기 방지되어 있지 않은 SD 메모리 카드를 사용하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	SD 메모리 카드 용량 부족		이벤트 코드	54011402Hex	
내용	명령 실행 중에 SD 메모리 카드에 쓸 때 SD 메모리 카드의 용량 부족이 발생했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	SD 메모리 카드의 용량 부족이 발생함		여유 용량이 충분한 SD 메모리 카드로 교환하십시오.		SD 메모리 카드에 추가로 쓰는 경우에는 여유 용량이 충분한 SD 메모리 카드를 사용하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • SD 메모리 카드 액세스 중에 SD 메모리 카드를 빼지 마십시오. SD 메모리 카드나 데이터가 손상될 수 있습니다.					

이벤트명	지정 파일 오류		이벤트 코드	54011403Hex	
내용	명령으로 지정된 파일이 존재하지 않습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	지정된 파일이 존재하지 않음	해당 명령으로 지정되는 파일명이 존재하는 파일명이 되도록 수정하십시오. 또는 파일명을 명령에서 지정된 이름으로 수정하십시오.	명령으로 지정되는 파일명을 존재하는 파일명으로 지정하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	파일 · 디렉토리 최대 수 초과		이벤트 코드	54011404Hex	
내용	명령 실행 중에 파일/디렉토리를 작성할 때 파일/디렉토리 수가 최대 수를 초과했습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	파일/디렉토리 수가 최대 수를 초과함	불필요한 파일/디렉토리를 삭제하십시오. 또는 파일/디렉토리 수가 FAT16/FAT32의 최대 수에 대해 충분히 여유가 있는 SD 메모리 카드로 교환하십시오.	파일/디렉토리 수가 너무 많아지지 않도록 불필요해진 파일/디렉토리를 적당히 삭제하십시오. 또한 파일이 계속 늘어나는 운용 방식인 경우는 정기적으로 SD 메모리 카드를 교환하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	파일 액세스 중			이벤트 코드	54011405Hex	
내용	명령으로 지정된 파일을 사용 중이므로 액세스하지 못했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	다른 명령에서 액세스 중인 파일에 또 다른 명령에서 쓰기 또는 읽기를 실행하려고 함		파일에 액세스하는 다른 명령의 출력 변수 'Busy'가 FALSE인 것을 조건으로 해당 명령을 실행하도록 프로그램을 수정하십시오.		파일에 액세스하는 복수의 명령을 실행하는 경우는 해당 명령들의 출력 변수 'Busy'를 명령 실행 조건으로 사용하여 명령이 동시에 실행되지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	열기 모드 불일치			이벤트 코드	54011406Hex	
내용	명령의 파일 조작이 파일의 열기 모드와 일치하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	파일 열기 명령으로 지정된 파일의 열기 모드와 그 이후 SD 메모리 카드용 명령에 의한 파일 조작에 부정합이 있음		파일 열기 명령으로 지정되는 열기 모드를 파일 조작에 맞는 모드가 되도록 수정하십시오.		파일 열기 명령으로 지정되는 열기 모드를 그 후의 파일 조작에 맞는 모드가 되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	오프셋 범위 밖			이벤트 코드	54011407Hex	
내용	명령으로 지정된 오프셋으로 지정되는 주소에 액세스할 수 없습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	파일 크기를 초과하여 액세스하려고 함		해당 명령으로 지정되는 오프셋을 작게 하십시오.		파일 형식의 식별 정보 등을 파일 내에 포함시켜 두고, 이 정보를 확인하여 적절히 파일을 검색하는 프로그램으로 변경하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	빈 디렉토리 이외			이벤트 코드	54011408Hex	
내용	디렉토리 삭제 명령을 실행하거나 디렉토리 이름을 변경하려고 했지만 디렉토리 내용이 비어 있지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	디렉토리 삭제 명령을 실행하려고 했지만 디렉토리 내용이 비어 있지 않음		해당 디렉토리의 파일을 모두 삭제하십시오.		디렉토리를 삭제하는 명령을 사용하는 경우 또는 디렉토리 이름을 변경하는 경우는 사전에 디렉토리의 내용을 확인해 두십시오.	
	디렉토리 이름을 변경하려고 했지만 디렉토리 안에 디렉토리가 존재함		해당 디렉토리 내의 디렉토리를 모두 삭제하십시오.			
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	동일한 파일명 있음			이벤트 코드	54011409Hex	
내용	명령으로 지정된 파일과 동일한 이름의 파일이 존재하므로 명령을 실행하지 못했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령으로 지정된 작성 파일과 동일한 이름의 파일이 이미 존재함		해당 명령으로 지정되는 파일명이 이미 존재하는 파일명과 중복되지 않도록 수정하십시오. 또는 미리 해당 파일을 삭제하십시오.		파일을 작성하는 명령을 사용하는 경우는 명령으로 지정되는 파일명이 이미 존재하는 파일명과 중복되지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • 파일을 삭제하는 경우, 필요한 파일이 아님을 확인한 후 삭제하십시오.					

이벤트명	지정 파일 쓰기 금지			이벤트 코드	5401140AHex	
내용	명령 실행 시 쓰기 방지된 파일 또는 디렉토리에 쓰려고 했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령으로 지정된 쓰기 파일 또는 디렉토리에 쓰기 방지가 지정되어 있음		해당 명령으로 지정한 쓰기 파일의 쓰기 방지를 해제하십시오. 또는 쓰기 파일명을 변경하십시오.		쓰기 파일에는 쓰기 방지를 설정하지 마십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • 파일의 쓰기 방지를 해제하는 경우는 해당 파일에 덮어써도 문제가 없는지 확인하십시오.					

이벤트명	파일 열기 최대 수 초과			이벤트 코드	5401140B Hex	
내용	명령 실행 중에 파일을 열 때 열린 파일 수가 최대 수를 초과했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 실행 중에 파일을 열 때 열린 파일 수가 최대 수를 초과함		동시에 여는 파일 수를 줄이도록 프로그램을 수정하십시오.		사용하는 파일 수를 적게 하십시오. 또는 동시에 여는 파일 수가 너무 많아지지 않도록 더 이상 열 필요가 없는 파일을 닫도록 하는 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	지정 디렉토리 오류			이벤트 코드	5401140C Hex	
내용	명령으로 지정된 디렉토리가 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령으로 지정된 디렉토리가 존재하지 않음		해당 명령으로 지정되는 디렉토리가 존재하는 디렉토리가 되도록 수정하십시오. 또는 해당 디렉토리를 사전에 작성하십시오.		디렉토리를 참조하는 명령을 사용하는 경우는 명령으로 지정되는 디렉토리는 존재하는 디렉토리가 되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	파일 · 디렉토리명 최대 길이 초과		이벤트 코드	5401140D Hex	
내용	명령으로 지정된 파일명 또는 디렉토리명이 너무 깁니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	명령으로 지정된 작성 파일명 또는 작성 디렉토리명이 너무 김	해당 명령으로 지정되는 파일명 또는 디렉토리명이 FAT16/FAT32의 제한 이내에 들어가도록 프로그램을 수정하십시오.	지정되는 파일명 또는 디렉토리명이 FAT16/FAT32의 제한 이내에 들어가도록 프로그램을 작성하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	SD 메모리 카드 액세스 실패		이벤트 코드	5401140E Hex	
내용	SD 메모리 카드에 액세스하지 못했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 유니트의 동작에는 영향 없음	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	SD 메모리 카드가 손상됨	SD 메모리 카드를 교환하십시오.	없음
	SD 메모리 카드 슬롯이 손상됨	상기의 처리를 실시해도 이 이상이 발생하는 경우는 CPU 유니트를 교환하십시오.	없음

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	EtherCAT 통신 에러		이벤트 코드	54011800Hex	
내용	명령 실행 시 EtherCAT 네트워크에 액세스하지 못했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	EtherCAT 네트워크가 실행 가능한 상태가 아님		EtherCAT 마스터 상태의 Status에서 EtherCAT 네트워크의 동작 상태를 확인하고 필요한 조치를 취하여 원인을 제거하십시오.	이상의 내용에 따름	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	EtherCAT 슬레이브 부재		이벤트 코드	54011801Hex	
내용	명령 실행 시 대상 슬레이브에 액세스하지 못했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	대상 슬레이브가 존재하지 않음		존재하는 노드 주소를 지정하십시오.	대상 슬레이브에는 존재하는 노드 주소를 지정하십시오.	
	대상 슬레이브가 실행 가능한 상태가 아님		대상 슬레이브의 Status를 확인하여 대상 슬레이브를 실행 가능 상태로 만드십시오.	대상 슬레이브를 실행 가능한 상태로 만드십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	EtherCAT 타임아웃		이벤트 코드	54011802Hex	
내용	명령 실행 시 EtherCAT 슬레이브에 대한 액세스가 타임아웃되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	대상 슬레이브와의 통신이 타임아웃됨		대상 슬레이브의 동작 상태를 확인하고 필요한 조치를 취하여 원인을 제거하십시오.	이상의 내용에 따름	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	수신 버퍼 초과		이벤트 코드	54011803Hex	
내용	명령 실행 시 EtherCAT 슬레이브에서의 수신 데이터가 수신 버퍼를 초과했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료. 슬레이브에서의 데이터 수신 불가능	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	슬레이브에서의 수신 데이터 크기가 수신 버퍼를 초과함		수신 버퍼의 크기를 슬레이브로부터의 수신 데이터 크기 이상으로 설정하십시오.	수신 버퍼의 크기를 슬레이브로부터의 수신 데이터 크기 이상으로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	SDO Abort 에러			이벤트 코드	54011804Hex	
내용	명령 실행 시 EtherCAT 슬레이브에서 SDO Abort 에러를 수신했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	슬레이브 사양에 따름		슬레이브의 설명서를 참조하여 조치를 취하십시오.		슬레이브의 설명서를 참조하여 방지 대책을 실시하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	패킷 모니터 저장 중			이벤트 코드	54011805Hex	
내용	EtherCAT의 패킷 모니터 파일 저장 중에 패킷 모니터 조작 관련 명령이 실행되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	EtherCAT의 패킷 모니터의 저장 중에 패킷 모니터 조작 관련 명령이 실행됨		패킷 모니터의 저장 작업이 완료된 후 패킷 모니터링 조작 관련 명령을 실행하십시오. 패킷 모니터의 저장이 완료되었는지 여부는 패킷 모니터 저장 중 Status를 통해 확인할 수 있습니다.		패킷 모니터 조작 관련 명령은 패킷 모니터링 저장 완료 후에 실행하십시오. 패킷 모니터의 저장이 완료되었는지 여부는 패킷 모니터 저장 중 Status를 통해 확인할 수 있습니다.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	패킷 모니터 기능 시작 안 함		이벤트 코드	54011806Hex	
내용	EtherCAT의 패킷 모니터 정지 중에 패킷 모니터 정지 명령이 실행되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	EtherCAT의 패킷 모니터 기능 정지 중에 패킷 모니터 정지 명령이 실행됨		패킷 모니터 기능 시작 후에 패킷 모니터 정지 명령을 실행하십시오. 패킷 모니터 기능이 동작 중인지 여부는 패킷 모니터 기능 동작 Status를 통해 확인할 수 있습니다.	패킷 모니터 정지 명령은 패킷 모니터 기능 시작 후에 실행하십시오. 패킷 모니터 기능이 동작 중인지 여부는 패킷 모니터 기능 동작 Status를 통해 확인할 수 있습니다.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	패킷 모니터 기능 동작 중		이벤트 코드	54011807Hex	
내용	EtherCAT의 패킷 모니터 동작 중에 패킷 모니터 시작 명령이 실행되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	EtherCAT의 패킷 모니터 기능 동작 중에 다시 패킷 모니터 시작 명령이 실행됨		패킷 모니터 기능 정지 후에 패킷 모니터 시작 명령을 실행하십시오. 패킷 모니터 기능이 정지 중인지 여부는 패킷 모니터 기능 동작 Status를 통해 확인할 수 있습니다.	패킷 모니터 시작 명령은 패킷 모니터 기능 정지 후에 실행하십시오. 패킷 모니터 기능이 정지 중인지 여부는 패킷 모니터 기능 동작 Status를 통해 확인할 수 있습니다.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	통신 리소스 초과		이벤트 코드	54011808Hex	
내용	동시에 32개를 넘는 EtherCAT 통신 명령이 실행되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	동시에 32개를 넘는 EtherCAT 통신 명령이 실행됨. EtherCAT 통신 명령은 다음과 같음 • EC_CoESDOWrite 명령 • EC_CoESDORead 명령 • EC_ConnectSlave 명령 • EC_DisconnectSlave 명령 • EC_StartMon 명령 • EC_SaveMon 명령 • EC_StopMon 명령 • EC_CopyMon 명령		동시에 실행하는 EtherCAT 명령이 32개 이하가 되도록 프로그램을 수정하십시오.	동시에 실행하는 EtherCAT 명령이 32개 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	패킷 모니터 기능 지원 안됨		이벤트 코드	54011809Hex *1	
내용	패킷 모니터 기능을 사용할 수 없습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	패킷 모니터 기능이 탑재되지 않은 CPU 유니트에 대해 패킷 모니터 기능의 명령이 실행됨		EC_StartMon, EC_SaveMon, EC_StopMon, EC_CopyMon 명령을 실행하지 마십시오. 패킷 모니터를 사용하는 경우는 패킷 모니터 기능을 탑재한 CPU 유니트 기종을 사용하십시오.	패킷 모니터 기능이 탑재되지 않은 CPU 유니트 기종에 대해 패킷 모니터 기능의 명령을 실행하지 마십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

*1 CPU 유니트의 유니트 버전 Ver.1.01에서 추가된 에러 코드(16#1809)입니다.

이벤트명	Explicit 이상			이벤트 코드	54011C00Hex	
내용	CIP 통신 명령에서 Explicit 메시지로 에러 응답 코드가 반환되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	이상의 내용에 따름		해당 명령의 출력 변수 'ErrorIDEx'의 값을 확인하고, 이 설명서에서 CIP 메시지 이상 코드의 설명을 참조하십시오.		이상의 내용에 따릅니다. 이 설명서에서 CIP 메시지 이상 코드의 설명을 참조하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	라우팅 경로 오류			이벤트 코드	54011C01Hex	
내용	CIP 통신 명령으로 지정된 라우팅 경로의 형식이 잘못되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CIP 통신 명령으로 지정된 라우팅 경로의 형식이 잘못됨		해당 명령으로 지정하는 라우팅 경로가 올바르게 되도록 프로그램을 수정하십시오.		명령으로 지정하는 라우팅 경로가 올바르게 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	CIP 핸들 오류		이벤트 코드	54011C02Hex	
내용	CIP 통신 명령으로 지정된 핸들이 잘못되었습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CIP 통신 명령으로 지정된 핸들이 잘못됨		해당 명령으로 지정하는 핸들이 CIPOpen 명령으로 취득한 핸들이 되도록 프로그램을 수정하십시오.		명령으로 지정하는 핸들이 CIPOpen 명령으로 취득한 핸들이 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	CIP 통신 리소스 초과		이벤트 코드	54011C03Hex	
내용	동시에 실행 가능한 CIP 통신 명령 리소스를 초과하여 명령이 실행되었습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	동시에 32개를 넘는 CIP 통신 명령이 실행됨		동시에 실행하는 CIP 통신 명령이 32개 이하가 되도록 프로그램을 수정하십시오.		동시에 실행하는 CIP 통신 명령이 32개 이하가 되도록 프로그램을 작성하십시오.	
	동시에 32개를 넘는 핸들을 사용하려고 함		동시에 사용하는 핸들이 32개 이하가 되도록 프로그램을 수정하십시오.		동시에 사용하는 핸들이 32개 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	CIP 타임아웃		이벤트 코드	54011C04 Hex	
내용	CIP 통신 명령의 실행 중에 타임아웃이 발생했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	지정된 IP 주소의 장치가 존재하지 않음		지정하는 IP 주소와 상대 장치의 IP 주소가 일치하도록 수정하십시오.	지정하는 IP 주소와 송신대상 장치의 IP 주소가 일치하도록 하십시오.	
	지정된 핸들의 CIP 연결이 타임아웃되었으므로 닫힘		연결의 타임아웃 시간 이내에 해당 명령을 실행하십시오. 또는 연결의 타임아웃 시간을 길게 하십시오.	연결의 타임아웃 시간 이내에 해당 명령을 실행하십시오.	
	상대 장치의 전원 OFF		상대 장치의 상태를 확인하고 정상적으로 기동시키십시오.	상대 장치의 상태를 확인하고 정상적으로 기동시키십시오.	
	상대 장치의 통신 정지				
	EtherNet/IP의 Ethernet 케이블 커넥터의 연결 불량		커넥터가 정확하게 감합되어 있는지 확인하고 다시 연결하십시오.	커넥터를 확실히 연결하십시오.	
	EtherNet/IP의 Ethernet 케이블 단선		Ethernet 케이블을 교환하십시오.	없음	
	노이즈		노이즈가 많은 경우에는 노이즈 대책을 실시하십시오.	노이즈가 많은 경우에는 노이즈 대책을 실시하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	Class3 연결 확립 실패			이벤트 코드	54011C05Hex*1	
내용	CIP 통신 명령에서 Class3 연결 확립에 실패했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		−		−	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해 CIPOpen 명령을 실행함		Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해서는 CIPOpenWithDataSize 명령을 사용하고, 데이터 크기가 509바이트 이하가 되도록 프로그램을 수정하십시오.		Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해서는 CIPOpenWithDataSize 명령을 사용하고, 데이터 크기가 509바이트 이하가 되도록 프로그램을 작성하십시오.	
	Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해 데이터 크기를 510바이트 이상으로 설정하여 CIPOpenWithDataSize 명령을 실행함		Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해서는 CIPOpenWithDataSize 명령의 데이터 크기가 509바이트 이하가 되도록 프로그램을 수정하십시오.		Class3(Large_Forward_Open)에 대응하지 않는 장치에 대해서는 CIPOpenWithDataSize 명령의 데이터 크기가 509바이트 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#1C05)입니다.

이벤트명	CIP 통신 데이터 크기 초과			이벤트 코드	54011C06Hex*1	
내용	CIP 통신 명령에서 송신 가능한 데이터 크기를 초과하는 Class3 Explicit 메시지를 송신하려고 했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		−		−	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CIPRead 명령, CIPWrite 명령, CIPSend 명령의 입력 변수로 설정한 데이터 크기가 CIPOpenWithDataSize 명령으로 설정한 데이터 크기를 초과함		해당 명령의 데이터 크기가 CIPOpenWithDataSize 명령으로 설정한 데이터 크기 이하가 되도록 프로그램을 수정하십시오. 또는 해당 명령의 데이터 크기 이상이 되도록 CIPOpenWithDataSize 명령의 데이터 크기를 설정하여 연결을 확립하십시오.		해당 명령의 데이터 크기가 CIPOpenWithDataSize 명령으로 설정한 데이터 크기 이하가 되도록 프로그램을 작성하십시오. 또는 해당 명령의 데이터 크기 이상이 되도록 CIPOpenWithDataSize 명령의 데이터 크기를 설정하여 연결을 확립하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#1C06)입니다.

이벤트명	로컬 IP 주소 설정 에러		이벤트 코드	54012000Hex	
내용	로컬 IP 주소의 설정 에러가 발생한 상태에서 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	없음	–	–

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	로컬 IP 주소의 설정 에러가 발생한 상태에서 명령을 실행함	해당 명령 실행 시에 'TCP/IP 기본 설정 에러'(IP 주소의 설정 에러)가 발생했습니다. 'TCP/IP 기본 설정 에러'의 처리 대책을 실시해서 이상을 제거하십시오.	'TCP/IP 기본 설정 이상'(IP 주소의 설정 에러)이 발생하지 않도록 정확하게 IP 주소를 설정하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	TCP/UDP 포트 사용 불가능		이벤트 코드	54012001Hex	
내용	명령 실행 시 UDP 또는 TCP 포트가 이미 사용되고 있습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	UDP 또는 TCP 포트가 이미 사용되고 있음		해당 명령으로 지정한 포트 번호가 미사용 포트 번호가 되도록 프로그램을 수정하십시오.	사용되고 있는 포트 번호를 명령에서 지정하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주소 해결 실패		이벤트 코드	54012002Hex	
내용	명령에서 도메인명이 지정된 상대 노드의 주소 해결에 실패했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	해당 명령으로 지정된 도메인명이 잘못됨		해당 명령으로 지정하는 도메인명을 올바른 도메인명으로 수정하십시오.	명령에서 올바른 도메인명을 지정했는지 확인하십시오.	
	컨트롤러의 hosts 설정/DNS 설정이 잘못됨		컨트롤러의 hosts 설정/DNS 설정을 수정하십시오.	컨트롤러의 hosts 설정/DNS 설정이 잘못되지 않았는지 확인하십시오.	
	DNS 서버의 설정이 잘못됨		DNS 서버의 설정을 수정하십시오.	DNS 서버의 설정이 잘못되지 않았는지 확인하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

<table>
<tr><td>이벤트명</td><td colspan="2">상태 이상</td><td>이벤트 코드</td><td colspan="2">54012003Hex</td></tr>
<tr><td>내용</td><td colspan="5">명령 실행 시의 상태가 적절하지 않습니다.</td></tr>
<tr><td>발생원</td><td colspan="2">PLC 기능 모듈</td><td>발생원 상세 정보</td><td>명령</td><td>검출 타이밍</td><td>명령 실행 시</td></tr>
<tr><td>이상의 속성</td><td>중요도</td><td>감시 정보</td><td>복구 방법</td><td>−</td><td>로그 카테고리</td><td>시스템</td></tr>
<tr><td>발생의 영향</td><td>사용자 프로그램</td><td>계속</td><td>동작</td><td colspan="3">해당 명령을 사양대로 종료</td></tr>
<tr><td>시스템 정의 변수</td><td colspan="2">변수명</td><td colspan="2">데이터 형식</td><td colspan="2">명칭</td></tr>
<tr><td></td><td colspan="2">없음</td><td colspan="2">−</td><td colspan="2">−</td></tr>
<tr><td>발생 요인과 처리</td><td colspan="2">발생 요인(추정 원인)</td><td colspan="2">처리 대책</td><td colspan="2">재발 방지</td></tr>
<tr><td></td><td colspan="2">• SktUDPRcv 명령의 경우
  • 소켓이 수신 처리 중
  • 소켓이 열리지 않음
• SktUDPSend 명령의 경우
  • 소켓이 송신 처리 중
  • 소켓이 열리지 않음
• SktTCPAccept 명령의 경우
지정된 TCP 포트가 아래 중 하나
  • 열기 처리 중
  • 클로즈 처리 중
  • 동일한 IP 주소, TCP 포트에서 이 명령이 이미 연결을 확립함
• SktTCPConnect 명령의 경우
  • 입력 변수 'SrcTcpPort'에 지정된 TCP 포트가 이미 열려 있음
  • 입력 변수 'DstAdr'로 지정된 상대 노드가 존재하지 않음
  • 입력 변수 'DstAdr', 'DstTcpPort'로 지정된 상대 노드가 연결 대기 중이 아님
• SktTCPRcv 명령의 경우
  • 지정된 소켓이 수신 처리 중
  • 지정된 소켓의 연결이 확립되지 않음
• SktTCPSend 명령의 경우
  • 지정된 소켓이 송신 처리 중
  • 지정된 소켓의 연결이 확립되지 않음</td><td colspan="2">해당 명령에 대응하는 이 이상의 발생 요인을 제거하십시오.</td><td colspan="2">발생 요인이 있는 상태에서 명령을 실행하지 않도록 하십시오.</td></tr>
<tr><td>부속 정보</td><td colspan="6">부속 정보 1: 이상의 발생 위치
부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호
부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함
부속 정보 4: 확장 에러 코드(ErrorIDEx)</td></tr>
<tr><td>주의점/비고</td><td colspan="6">이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.</td></tr>
</table>

이벤트명	로컬 IP 주소 결정 안됨		이벤트 코드	54012004 Hex	
내용	소켓 서비스 명령 실행 시 로컬 IP 주소가 결정되지 않았습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	BOOTP 서버의 설정 이상		BOOTP 서버의 설정에 오류가 있는 경우에는 설정을 수정하십시오.	BOOTP 서버의 설정에 오류가 없는지 확인하십시오.	
	BOOTP 서버 부재		BOOTP 서버가 정상적으로 네트워크에 연결되어 있는지, 정상적으로 기동되어 있는지 확인한 후 BOOTP 서버에 연결할 수 있도록 하십시오.	BOOTP 서버가 정상적으로 네트워크에 연결되어 있고 정상적으로 기동되어 있는지 확인하십시오.	
	기동 직후이므로 로컬 IP 주소가 결정되지 않았음		로컬 IP 주소가 결정되기를 기다린 후 소켓 서비스 명령을 실행하십시오.	로컬 IP 주소가 결정되기를 기다린 후 소켓 서비스 명령을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	소켓 타임아웃		이벤트 코드	54012006 Hex	
내용	소켓 서비스 명령에서 타임아웃이 발생했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	SktTCPAccept 명령: 사용자가 지정한 타임아웃 시간 이내에 상대 노드에서 연결 요구가 없음		해당 명령을 실행한 후부터 타임아웃될 때까지 상대 노드로부터 연결 요구가 실시되도록 시스템이나 프로그램을 수정하십시오. 또는 타임아웃 시간을 크게 하십시오.	명령을 실행한 후부터 타임아웃될 때까지 상대 노드로부터 연결 요구가 실시되도록 시스템이나 프로그램을 작성하십시오.	
	SktTCPRcv 명령, SktUDPRcv 명령: 사용자가 지정한 타임아웃 시간 이내에 상대 노드에서 데이터가 수신되지 않음		해당 명령을 실행한 후부터 타임아웃될 때까지 상대 노드로부터 데이터 송신이 실시되도록 시스템이나 프로그램을 수정하십시오. 또는 타임아웃 시간을 크게 하십시오.	명령을 실행한 후부터 타임아웃될 때까지 상대 노드로부터 데이터 송신이 실시되도록 시스템이나 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	소켓 핸들 오류		이벤트 코드	54012007Hex	
내용	소켓 서비스 명령으로 지정된 핸들이 잘못되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	소켓 서비스 명령으로 지정된 핸들이 잘못됨		해당 명령으로 지정하는 소켓 핸들이 다음 중 하나의 명령으로 취득된 핸들이 되도록 프로그램을 수정하십시오. • SktUDPCreate 명령 • SktTCPConnect 명령 • SktTCPAccept 명령	해당 명령으로 지정하는 소켓 핸들이 다음 중 하나의 명령으로 취득된 핸들이 되도록 프로그램을 작성하십시오. • SktUDPCreate 명령 • SktTCPConnect 명령 • SktTCPAccept 명령	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	소켓 통신 리소스 초과		이벤트 코드	54012008Hex	
내용	동시에 실행 가능한 소켓 서비스 명령 리소스를 초과하여 명령이 실행되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	동시에 32개를 넘는 소켓 서비스 명령이 실행됨		동시에 실행하는 소켓 서비스 명령이 32개 이하가 되도록 프로그램을 수정하십시오.	동시에 실행하는 소켓 서비스 명령이 32개 이하가 되도록 프로그램을 작성하십시오.	
	동시에 30개(CPU 유니트 Ver.1.02 이하는 16개)를 넘는 소켓 핸들을 사용하려고 함		동시에 사용하는 소켓 핸들이 30개(CPU 유니트 Ver.1.02 이하는 16개) 이하가 되도록 프로그램을 수정하십시오.	동시에 사용하는 소켓 핸들이 30개(CPU 유니트 Ver.1.02 이하는 16개) 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

<table>
<tr><td>이벤트명</td><td colspan="2">실행 권한 없음</td><td>이벤트 코드</td><td colspan="2">54012400Hex *1</td></tr>
<tr><td>내용</td><td colspan="5">실행 불가능한 상태에서 EtherNet/IP 포트의 설정을 변경하는 명령을 실행했습니다.</td></tr>
<tr><td>발생원</td><td colspan="2">PLC 기능 모듈</td><td>발생원 상세 정보</td><td>명령</td><td>검출 타이밍</td><td>명령 실행 시</td></tr>
<tr><td>이상의 속성</td><td>중요도</td><td>감시 정보</td><td>복구 방법</td><td>−</td><td>로그 카테고리</td><td>시스템</td></tr>
<tr><td>발생의 영향</td><td>사용자 프로그램</td><td>계속</td><td>동작</td><td colspan="3">영향 없음</td></tr>
<tr><td rowspan="2">시스템 정의 변수</td><td colspan="2">변수명</td><td colspan="2">데이터 형식</td><td colspan="2">명칭</td></tr>
<tr><td colspan="2">없음</td><td colspan="2">−</td><td colspan="2">−</td></tr>
<tr><td rowspan="6">발생 요인과 처리</td><td colspan="2">발생 요인(추정 원인)</td><td colspan="2">처리 대책</td><td colspan="2">재발 방지</td></tr>
<tr><td colspan="2">내장 EtherNet/IP 포트의 재시작 중에 내장 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령을 실행함</td><td colspan="2" rowspan="4">내장 EtherNet/IP 포트, CJ 시리즈 EtherNet/IP 유니트의 재시작이나 설정 변경이 종료된 후 설정을 변경하는 명령을 실행하십시오.</td><td colspan="2" rowspan="4">내장 EtherNet/IP 포트, CJ 시리즈 EtherNet/IP 유니트의 재시작이나 설정 변경이 수행되고 있지 않을 때 설정을 변경하는 명령을 실행하십시오.</td></tr>
<tr><td colspan="2">CJ 시리즈 EtherNet/IP 유니트의 재시작 중에 해당 유니트의 설정을 변경하는 명령을 실행함</td></tr>
<tr><td colspan="2">명령 또는 CIP 메시지를 사용한 내장 EtherNet/IP 포트의 설정 변경 중에 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령을 실행함</td></tr>
<tr><td colspan="2">명령 또는 CIP 메시지를 사용한 CJ 시리즈 EtherNet/IP 유니트의 설정 변경 중에 해당 유니트의 설정을 변경하는 명령을 실행함</td></tr>
<tr><td colspan="2">명령으로 지정된 호기 번호가 내장 EtherNet/IP 포트, CJ 시리즈 EtherNet/IP 유니트가 아님</td><td colspan="2">내장 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 호기 번호를 명령에서 지정하십시오. 유니트 구성이 잘못된 경우는 유니트 구성을 올바르게 설정하십시오.</td><td colspan="2">내장 EtherNet/IP 포트 또는 CJ 시리즈 EtherNet/IP 유니트의 호기 번호를 명령에서 지정하십시오.</td></tr>
<tr><td>부속 정보</td><td colspan="6">부속 정보 1: 이상의 발생 위치
부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호
부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함
부속 정보 4: 확장 에러 코드(ErrorIDEx)</td></tr>
<tr><td>주의점/비고</td><td colspan="6">이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.</td></tr>
</table>

*1 CPU 유니트의 유니트 버전 Ver.1.02에서 추가된 에러 코드(16#2400)입니다.

이벤트명	설정 반영 실패		이벤트 코드	54012401Hex *1		
내용	CJ 시리즈 EtherNet/IP 유니트에서 변경된 설정을 반영하지 못했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령의 실행 중에 해당 유니트 또는 내장 EtherNet/IP 포트가 재시작됨		CJ 시리즈 EtherNet/IP 유니트, 내장 EtherNet/IP 포트의 재시작이 종료된 후 설정을 변경하는 명령을 다시 실행하십시오.		CJ 시리즈 EtherNet/IP 유니트의 설정을 변경하는 명령의 실행 중에 해당 유니트 또는 내장 EtherNet/IP 포트를 재시작하지 마십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.02에서 추가된 에러 코드(16#2401)입니다.

이벤트명	명령 동시 실행 수 초과		이벤트 코드	54012402Hex *1		
내용	컨트롤러 통신 설정 명령이 동시에 실행 가능한 수를 초과하여 실행되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	동시에 2개 이상의 컨트롤러 통신 설정 명령이 실행됨		동시에 실행하는 컨트롤러 통신 설정 명령이 1개가 되도록 프로그램을 수정하십시오.		동시에 실행하는 컨트롤러 통신 설정 명령이 1개가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.02에서 추가된 에러 코드(16#2402)입니다.

이벤트명	NX 메시지 이상		이벤트 코드	54012C00Hex*1	
내용	NX 메시지로 에러 응답 코드가 반환되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	이상의 내용에 따름		해당 명령의 출력 변수 'ErrorIDEx'의 값을 확인하고, 이 설명서에서 NX 메시지 이상 코드의 설명을 참조하십시오.	이상의 내용에 따릅니다. 이 설명서에서 NX 메시지 이상 코드의 설명을 참조하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C00)입니다.

이벤트명	NX 메시지 리소스 초과		이벤트 코드	54012C01Hex*1	
내용	동시에 실행 가능한 NX 메시지 명령의 리소스를 초과하여 명령이 실행되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	없음		–	–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	동시에 32개를 넘는 NX 메시지 명령이 실행됨		동시에 실행하는 NX 메시지 명령이 32개 이하가 되도록 프로그램을 수정하십시오.	동시에 실행하는 NX 메시지 명령이 32개 이하가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C01)입니다.

이벤트명	NX 메시지 타임아웃		이벤트 코드	54012C02Hex*1		
내용	NX 메시지의 실행 중에 타임아웃이 발생했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	지정된 NX 유니트가 존재하지 않음		유니트 지정과 상대 유니트의 구성 등이 일치하도록 수정하십시오.		유니트 지정과 상대 유니트의 구성 등이 일치하도록 하십시오.	
	NX 메시지가 타임아웃되었으므로 닫힘		해당 명령의 입력 변수 'TimeOut'에 지정된 응답 감시 시간을 길게 하십시오.		적절한 응답 감시 시간을 입력 변수 'TimeOut'에 지정한 후 명령을 실행하십시오.	
	상대 유니트의 전원 OFF		상대 유니트의 상태를 확인하고 정상적으로 기동시키십시오.		상대 유니트의 상태를 확인하고 정상적으로 기동시키십시오.	
	상대 유니트의 통신 정지					
	통신 케이블의 커넥터 연결 불량		커넥터가 정확하게 감합되어 있는지 확인하고 다시 연결하십시오.		커넥터를 확실히 연결하십시오.	
	통신 케이블의 단선		통신 케이블을 교환하십시오.		없음	
	노이즈		노이즈가 많은 경우에는 노이즈 대책을 실시하십시오.		노이즈가 많은 경우에는 노이즈 대책을 실시하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C02)입니다.

이벤트명	NX 메시지 길이 오류		이벤트 코드	54012C03Hex*1		
내용	NX 메시지 길이가 잘못되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	WriteDat 또는 Path로 지정한 크기가 너무 김		WriteDat 또는 Path로 지정한 크기가 제한 이내가 되도록 프로그램을 수정하십시오.		WriteDat 또는 Path로 지정한 크기가 제한 이내가 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C03)입니다.

이벤트명	NX 메시지 네트워크 이상(EtherCAT)		이벤트 코드	54012C05Hex*1		
내용	NX 메시지 경로상의 EtherCAT 통신에서 이상이 발생했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	NX 메시지의 경로 중 EtherCAT 통신에서 이상이 발생		EtherCAT 통신에서 발생한 이상을 확인하고, 이상을 해제한 후 해당 명령을 실행하십시오.		이상의 내용에 따름	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C05)입니다.

이벤트명	지정 유니트 외부 재시작 이미 실행		이벤트 코드	54012C06Hex*1		
내용	명령 실행 시, 이미 Sysmac Studio에서 재시작이 실행되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		–		–	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 실행 시, 이미 Sysmac Studio에서 재시작이 실행됨		이미 Sysmac Studio에서 재시작이 실행되고 있는 경우는 명령에서 재시작할 필요가 없습니다.		운전 중에 Sysmac Studio에서 재시작하지 마십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C06)입니다.

이벤트명	명령 대상 이외의 유니트 지정		이벤트 코드	54012C07Hex*1		
내용	지정된 유니트의 슬레이브 노드 주소에 해당 명령의 대상이 아닌 슬레이브가 연결되어 있습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	없음		−		−	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	지정된 유니트의 슬레이브 노드 주소에 해당 명령의 대상이 아닌 슬레이브가 연결되어 있음		네트워크 구성 정보에 지정되어 있는 해당 명령의 대상 유니트를 연결하십시오.		지정된 유니트의 슬레이브 노드 주소에 해당 명령의 대상이 아닌 슬레이브를 연결하지 마십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.05에서 추가된 에러 코드(16#2C07)입니다.

<table>
<tr><td>이벤트명</td><td colspan="2">프로세스 데이터 개체 설정 부족</td><td>이벤트 코드</td><td colspan="2">54013461Hex</td></tr>
<tr><td>내용</td><td colspan="5">PDO 매핑이 정확하지 않습니다.</td></tr>
<tr><td>발생원</td><td colspan="2">PLC 기능 모듈</td><td>발생원 상세 정보</td><td>명령</td><td>검출 타이밍</td><td>명령 기동 시</td></tr>
<tr><td>이상의 속성</td><td>중요도</td><td>감시 정보</td><td>복구 방법</td><td>－</td><td>로그 카테고리</td><td>시스템</td></tr>
<tr><td>발생의 영향</td><td>사용자 프로그램</td><td>계속</td><td>동작</td><td colspan="3">해당 명령을 사양대로 종료</td></tr>
<tr><td rowspan="2">시스템 정의 변수</td><td colspan="2">변수명</td><td colspan="2">데이터 형식</td><td colspan="2">명칭</td></tr>
<tr><td colspan="2">_MC_AX[*].MFaultLvl.Active</td><td colspan="2">BOOL</td><td colspan="2">축 가벼운 이상 발생 중</td></tr>
<tr><td rowspan="4">발생 요인과 처리</td><td colspan="2">발생 요인(추정 원인)</td><td colspan="2">처리 대책</td><td colspan="2">재발 방지</td></tr>
<tr><td colspan="2">모션 명령에서 필요한 PDO 매핑이 되어 있지 않음</td><td colspan="2">해당 명령에서 필요한 PDO를 매핑하십시오.
필요한 PDO 매핑에 대해서는 해당 명령의 '기능 설명'을 참조하십시오.</td><td colspan="2">사용하는 명령에서 필요한 PDO를 매핑하십시오.
각 명령에서 필요한 PDO 매핑(서보 드라이버의 설정)에 대해서는 "NJ 시리즈 사용 설명서 모션 제어 편"을 참조하십시오.</td></tr>
<tr><td colspan="2">해당 명령에 대응하는 개체가 없는 대상 기기에 대해 해당 명령을 실행함</td><td colspan="2">해당 명령에 대응하는 기능이 없는 대상 기기가 있습니다.
대상 기기의 설명서를 참조하여 해당 명령에 대응하는지 확인하고 대응하지 않는 명령을 실행하지 않도록 프로그램을 수정하십시오.</td><td colspan="2">대상 기기의 설명서를 참조하여 대응하지 않는 명령을 실행하지 않도록 프로그램을 작성하십시오.</td></tr>
<tr><td colspan="2">오므론 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에 대해 트리거 조건으로 Z상(_mcEncoderMark)을 지정하여 모션 명령을 기동함</td><td colspan="2">오므론 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에서는 트리거 조건으로 외부 입력(_mcEXT)을 사용하십시오.</td><td colspan="2">오므론 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에서는 트리거 조건으로 외부 입력(_mcEXT)을 사용하십시오.</td></tr>
<tr><td>부속 정보</td><td colspan="6">부속 정보 1: 이상의 발생 위치
부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호
부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함
부속 정보 4: 확장 에러 코드(ErrorIDEx)</td></tr>
<tr><td>주의점/비고</td><td colspan="6">이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.</td></tr>
</table>

이벤트명	전자 기어 분자 설정 범위 밖		이벤트 코드		54015420Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumerator'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	전자 기어 분모 설정 범위 밖		이벤트 코드		54015421Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominator'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	목표 속도 설정 범위 밖		이벤트 코드	54015422Hex	
내용	모션 제어 명령의 입력 변수 'Velocity'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	가속도 설정 범위 밖		이벤트 코드	54015423Hex	
내용	모션 제어 명령의 입력 변수 'Acceleration'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	감속도 설정 범위 밖		이벤트 코드	54015424Hex	
내용	모션 제어 명령의 입력 변수 'Deceleration'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	저크 설정 범위 밖		이벤트 코드	54015425Hex	
내용	모션 제어 명령의 입력 변수 'Jerk'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	토크 경사 설정 범위 밖		이벤트 코드	54015427Hex	
내용	모션 제어 명령의 입력 변수 'TorqueRamp'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축 계수 설정 범위 밖		이벤트 코드	54015428Hex	
내용	모션 제어 명령의 입력 변수 'MasterScaling'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	종축 계수 설정 범위 밖		이벤트 코드	54015429Hex		
내용	모션 제어 명령의 입력 변수 'SlaveScaling'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	피딩 속도 설정 범위 밖		이벤트 코드	5401542AHex		
내용	모션 제어 명령의 입력 변수 'FeedVelocity'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	피딩 속도(입력 변수 'FeedVelocity')가 초기값 그대로(0)임		피딩 속도(입력 변수 'FeedVelocity')에 양수값을 지정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	버퍼 모드 선택 범위 밖		이벤트 코드	5401542BHex	
내용	모션 제어 명령의 입력 변수 'BufferMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	좌표계 선택 범위 밖		이벤트 코드	5401542CHex	
내용	모션 제어 명령의 입력 변수 'CoordSystem'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	원호 보간 모드 선택 범위 밖		이벤트 코드	5401542DHex	
내용	모션 제어 명령의 입력 변수 'CircMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	방향 선택 범위 밖		이벤트 코드	5401542EHex	
내용	모션 제어 명령의 입력 변수 'Direction'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	경로 선택 범위 밖			이벤트 코드	5401542F Hex	
내용	모션 제어 명령의 입력 변수 'PathChoice'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	위치 종류 선택 범위 밖			이벤트 코드	54015430 Hex	
내용	모션 제어 명령의 입력 변수 'ReferenceType'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	이동 방법 선택 범위 밖		이벤트 코드	54015431Hex	
내용	모션 제어 명령의 입력 변수 'MoveMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	트랜지션 모드 선택 범위 밖		이벤트 코드	54015432Hex	
내용	모션 제어 명령의 입력 변수 'TransitionMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
	'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하고 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정		'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하는 경우에는 'TransitionMode'를 '_mcTMNone'으로 지정하고, 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정하는 경우에는 'BufferMode'에서 '_mcBlendingLow', '_mcBlendingPrevious', '_mcBlendingNext', '_mcBlendingHigh'를 지정하십시오.	'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하는 경우에는 'TransitionMode'를 '_mcTMNone'으로 지정하고, 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정하는 경우에는 'BufferMode'에서 '_mcBlendingLow', '_mcBlendingPrevious', '_mcBlendingNext', '_mcBlendingHigh'를 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	계속 방법 선택 범위 밖		이벤트 코드	54015433Hex	
내용	모션 제어 명령의 입력 변수 'Continuous(Reserved)'의 값이 변경되었습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	입력 변수 'Continuous(Reserved)'의 값이 변경됨		입력 변수 'Continuous(Reserved)'의 값이 변경되지 않도록 프로그램을 수정하십시오.	입력 변수 'Continuous(Reserved)'의 값이 변경되지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	가감산 방법 선택 범위 밖		이벤트 코드	54015434Hex	
내용	모션 제어 명령의 입력 변수 'CombineMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	동기 시작 조건 지정 범위 밖		이벤트 코드	54015435Hex		
내용	모션 제어 명령의 입력 변수 'LinkOption'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축과 종축 동일		이벤트 코드	54015436Hex		
내용	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축이 동일합니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Master'와 'Slave'의 파라메터가 동일		해당 명령의 입력 변수 'Master'와 'Slave'에 지정되는 축이 다르도록 파라메터를 수정하십시오.		명령의 입력 변수 'Master'와 'Slave'에는 다른 축을 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축과 보조축 동일		이벤트 코드	54015437Hex	
내용	모션 제어 명령의 입력 변수 'Master'와 'Auxiliary'에 지정된 축이 동일합니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 변수 'Master'와 'Auxiliary'의 파라메터가 동일		해당 명령의 입력 변수 'Master'와 'Auxiliary'에 지정되는 축이 다르도록 파라메터를 수정하십시오.	명령의 입력 변수 'Master'와 'Auxiliary'에는 다른 축을 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축/종축 축 번호가 오름차순이 아님		이벤트 코드	54015438Hex	
내용	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축 번호가 오름차순이 아닙니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 변수 'ReferenceType'에 '_mcLatestCommand' 지정 시, 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 오름차순이 아님		해당 명령의 입력 변수 'ReferenceType'에 '_mcLatestCommand'를 지정하는 경우에는 해당 명령의 입력 변수 'Master'와 'Slave'에 지정되는 축 번호가 오름차순이 되도록 파라메터를 수정하십시오. 또는 주축 위치 종류 선택에서 '_mcCommand'를 지정하십시오.	변수 'ReferenceType'으로 '_mcLatestCommand'를 지정하는 경우에는 입력 변수의 주축과 종축이 오름차순이 되도록 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	잘못된 캠 테이블 지정			이벤트 코드		54015439Hex	
내용	모션 제어 명령의 입력 변수 'CamTable'에 지정된 파라메터가 범위 밖입니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식		명칭		
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중		
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중		
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지		
	명령의 입력 변수 'CamTable'에 캠 데이터 변수 이외를 지정		해당 명령의 입력 변수 'CamTable'에 지정되는 파라메터가 캠 데이터 변수가 되도록 파라메터를 수정하십시오.		명령의 입력 변수 'CamTable'에는 캠 데이터 변수를 지정하십시오.		
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	동기 정지 중			이벤트 코드		5401543AHex	
내용	모션 제어의 동기 제어 명령이 실행되었지만 실행 가능한 조건이 아닙니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식		명칭		
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중		
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지		
	• MC_CamOut(캠 동작 해제) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령 실행 중이 아님 • MC_GearOut(기어 동작 해제) 명령이 실행되었지만 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령이 실행 중이 아님 • MC_Phasing(주축 상대값 위상 보정) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령, MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령이 실행 중이 아님		해당 명령 실행 시에 실행 가능한 조건이 되도록 프로그램을 수정하십시오.		동기 명령 실행 시에는 실행 가능한 조건이 되도록 하십시오.		
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	모션 명령 재기동 불가능		이벤트 코드	5401543B Hex	
내용	재기동할 수 없는 모션 제어 명령의 재기동을 수행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_COM.MFaultLvl.Active	BOOL	MC 공통 가벼운 이상 발생 중
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	재기동할 수 없는 모션 제어 명령의 재기동	해당 명령의 출력 변수 'Busy'가 FALSE가 될 때까지 입력 변수 'Execute'가 기동되지 않도록 프로그램을 수정하십시오.	재기동할 수 없는 명령을 사용하는 경우는 입력 변수 'Execute'의 기동 조건에 해당 명령의 출력 변수 'Busy'가 FALSE인 것을 포함하도록 하십시오. 또는 명령을 다시 기동시키기 전에 일단 명령을 정지하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	모션 명령 다중 기동 불가능		이벤트 코드	5401543C Hex	
내용	동일한 대상(MC 공통/축/축 그룹)에 대해 동시에 실행할 수 없는 기능을 여러 개 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_COM.MFaultLvl.Active	BOOL	MC 공통 가벼운 이상 발생 중
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	동일한 대상(MC 공통/축)에 대해 동시에 실행할 수 없는 기능을 여러 개 실행	해당 명령의 다중 기동 사양을 확인하고 동시에 실행할 수 없는 명령을 동시에 실행하지 않도록 프로그램을 수정하십시오.	사용하는 명령의 다중 기동 사양을 확인하고 동시에 실행할 수 없는 명령을 동시에 실행하지 않도록 하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	축 종류 부적합		이벤트 코드	5401543DHex	
내용	엔코더 축에 대해 동작 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	엔코더 축에 대해 동작 명령을 실행		해당 명령의 축 종류 설정에 서보 축 또는 가상 서보 축을 지정하거나 엔코더 축에 대해 해당 동작 명령을 실행하지 않도록 프로그램을 수정하십시오.	동작 명령은 서보 축 또는 가상 서보 축에 대해 실행하도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	다축 협조 동작 중 명령 기동 불가능		이벤트 코드	5401543EHex	
내용	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행		해당 축이나 해당 축 그룹이 다축 협조 동작 중이 아닐 때 해당 명령을 실행하도록 프로그램을 수정하십시오.	다축 협조 동작 중이 아닌 축 또는 축 그룹에 대해 축 동작 명령을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	축 그룹 무효 상태에서 다축 협조 명령 기동		이벤트 코드		5401543F Hex	
내용	축 그룹 무효 상태의 축 그룹에 대해 다축 협조 명령을 기동했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	축 그룹 무효 상태의 축 그룹에 대해 다축 협조 명령을 기동		축 그룹을 유효 상태로 한 후에 해당 명령을 실행하도록 프로그램을 수정하십시오. 축 그룹을 유효 상태로 하려면 MC_GroupEnable(축 그룹 유효) 명령을 실행하십시오.		축 그룹을 유효하게 한 후 다축 협조 동작 명령을 실행하십시오. 축 그룹을 유효 상태로 하려면 MC_GroupEnable(축 그룹 유효) 명령을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	축 그룹 유효 불가능		이벤트 코드		54015440 Hex	
내용	MC_GroupEnable(축 그룹 유효) 명령 실행에 실패했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성축에 정지 중이 아닌 축이 있음		모든 구성축이 정지 중일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 수정하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다.		모든 구성축이 정지 중일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 작성하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다.	
	MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성축에 MC_TouchProbe(외부 래치 유효) 명령 실행 중인 축이 있음		모든 구성축에서 MC_TouchProbe(외부 래치 유효) 명령이 실행되지 않는 상태일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 수정하십시오.		모든 구성축에서 MC_TouchProbe(외부 래치 유효) 명령이 실행되지 않는 상태일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	운전 불가능(서보 OFF) 축 동작 지시		이벤트 코드	54015441Hex	
내용	서보 OFF 중인 축에 대해 동작 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	서보 OFF 중인 축에 대해 동작 명령을 실행		서보가 ON한 후에 해당 명령을 실행하도록 프로그램을 수정하십시오.	서보가 ON한 후에 축 동작 명령을 실행하도록 하십시오.	
	MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 통해 원점 사전 설정을 EtherCAT 프로세스 데이터 통신이 확립되지 않은 축에 대해 실행함		EtherCAT 마스터의 시스템 정의 변수 '_EC_PDSlavTbl'(프로세스 데이터 통신 중 슬레이브 테이블)이 FALSE가 되면 해당 요인을 제거하고, '_EC_PDSlavTbl'이 TRUE가 된 후에 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 통해 원점 사전 설정을 실행하십시오.	컨트롤러의 전원 ON이나 다운로드 직후 또는 슬레이브의 통신 이상 해제, 이탈, 재가입 직후에 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 사용하여 원점 사전 설정을 실행할 경우는 EtherCAT 마스터의 시스템 정의 변수 '_EC_PDSlavTbl'(프로세스 데이터 통신 중 슬레이브 테이블)이 TRUE가 된 것을 확인한 후 실행하도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	구성축 강제 정지 중 에러		이벤트 코드	54015442Hex	
내용	구성축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	구성축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행함		구성축의 강제 정지 명령 입력 변수 'Execute'를 FALSE로 하여 실행을 멈추고, 이상 해제를 실행한 후에 동작 명령을 전송하도록 하십시오.	구성축의 강제 정지 명령 입력 변수 'Execute'를 FALSE로 해서 종료시킨 후 축 그룹의 동작 명령을 실행하도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	모션 명령 다중 기동 수 초과			이벤트 코드		54015443Hex
내용	버퍼 모드인 Buffered, Blending을 통해 버퍼링되는 모션 제어 명령의 버퍼링 수가 한계를 넘었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	축 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 넘음		해당 명령 실행 시에 버퍼링 수가 상한을 넘지 않도록 프로그램을 수정하십시오.		축 명령인 경우는 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 초과하지 않도록 하십시오.	
	축 그룹 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 넘음				축 그룹 명령인 경우는 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 초과하지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	이동량 부족			이벤트 코드		54015444Hex
내용	위치 결정 명령의 다중 기동/재기동 시, 지정된 감속도 또는 가속도로 지정된 동작을 실행할 수 없습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	'가감속 초과'를 '이상으로 정지'로 설정한 경우, 위치 결정 명령의 다중 기동/재기동 시 지정된 감속도/가속도로 목표 위치에서 정지 불가능		해당 명령의 동작 사양을 바탕으로 위치 결정 동작이 다중 기동/재기동하는 타이밍에 지정한 감속도 또는 가속도로 목표 위치를 초과하지 않도록 프로그램을 수정하십시오. 또는 '가감속 초과'를 '이상으로 정지' 이외의 설정으로 변경하십시오.		사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오. 또는 '가감속 초과'를 '이상으로 정지' 이외의 것으로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	블렌딩 중계 속도에 도달하기 위한 이동량 부족		이벤트 코드	54015445Hex	
내용	중계 속도로 가감속하기 위한 이동량이 부족합니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	'가감속 초과'에 '이상으로 정지'가 설정된 경우 현재 명령을 중계 속도로 가감속하기 위한 이동량이 부족함		해당 명령의 동작 사양을 바탕으로 이동량이 부족하지 않도록 프로그램을 수정하십시오. 또는 '가감속 초과'를 '이상으로 정지' 이외의 설정으로 변경하십시오.	사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오. 또는 '가감속 초과'를 '이상으로 정지' 이외의 것으로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	사다리꼴 패턴 캠 등속 이동량 부족		이벤트 코드	54015446Hex	
내용	주축의 등속 이동량이 '0' 미만입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_MoveLink(사다리꼴 패턴 캠) 명령에서 주축의 등속 이동량이 '0' 미만		(주축 이동 거리≧주축 가속 이동 거리+주축 감속 이동 거리)가 되도록 프로그램을 수정하십시오.	사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	위치 지정 기어 동작 목표 속도 부족		이벤트 코드		54015447Hex	
내용	MC_GearInPos(위치 지정 기어 동작) 명령에서 종축의 '목표 속도'가 작기 때문에 필요한 속도에 도달하지 못했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_GearInPos(위치 지정 기어 동작) 명령에서 입력 변수 'Velocity(목표 속도)'의 값이 (명령 기동 시의 주축 속도x기어비)보다 작음		해당 명령의 동작 사양을 바탕으로 (명령 기동 시의 주축 속도×기어비) 보다 커지도록 입력 변수 'Velocity (목표 속도)'의 값을 설정하십시오.		사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	원호 보간 시작점과 종료점이 동일		이벤트 코드		54015448Hex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식 지정 시 시작점과 종료점이 동일한 위치입니다. 또는 통과점 지정 방식 지정 시 시작점과 종료점 및 통과점이 동일한 위치입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식이 지정되고 시작점과 종료점의 위치가 동일		해당 명령의 시작점과 종료점의 위치가 동일한 상태에서 반경 지정 방식을 지정하지 않도록 프로그램을 수정하십시오.		반경 지정 방식의 원호 보간을 실행하려면 시작점과 종료점이 동일하지 않도록 설정하십시오.	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식이 지정되고 시작점과 종료점 및 통과점의 위치가 동일		해당 명령의 시작점과 종료점과 통과점의 위치가 동일한 상태에서 통과점 지정 방식을 지정하지 않도록 프로그램을 수정하십시오.		통과점 지정 방식의 원호 보간을 실행하려면 시작점과 종료점과 통과점이 동일하지 않도록 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	원호 보간 중심점 지정 위치 범위 밖		이벤트 코드		54015449Hex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식 지정 시 중심점의 위치 지정이 허용 범위를 벗어났습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식이 지정되고 시작점과 중심점의 거리 및 종료점과 중심점의 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위를 벗어남		시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위보다 작은 중심점이 설정되도록 수정하십시오.		시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위 이하가 되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	카운트 모드 설정에 의한 명령 기동 이상		이벤트 코드		5401544AHex	
내용	카운트 모드가 로터리 모드로 설정되어 있는 축에 대해 로터리 모드에서 사용할 수 없는 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	카운트 모드가 로터리 모드로 설정되어 있는 축을 로터리 모드에서 사용할 수 없는 명령에서 사용		해당 명령으로 지정된 해당 축의 카운트 모드 설정을 리니어 모드로 변경하십시오.		사용할 명령을 실행할 수 있는 카운트 모드를 확인하고 축의 카운트 모드를 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	파라메터 선택 범위 밖			이벤트 코드		5401544C Hex
내용	모션 제어 명령의 입력 변수 'ParameterNumber'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	정지 방법 선택 범위 밖			이벤트 코드		5401544D Hex
내용	모션 제어 명령의 입력 변수 'StopMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	트리거 입력 조건의 래치 ID 선택 범위 밖		이벤트 코드	5401544E Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::LatchID'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	MC 설정 쓰기의 설정 범위 밖		이벤트 코드	5401544F Hex	
내용	모션 제어 명령의 입력 변수 'SettingValue'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_COM.MFaultLvl.Active		BOOL	MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
	파라메터 지정과 설정값의 데이터 형식이 일치하지 않음		파라메터 지정과 설정값의 데이터 형식이 일치하도록 수정하십시오.	파라메터 지정과 설정값의 데이터 형식을 일치시키십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	트리거 입력 조건의 모드 선택 범위 밖		이벤트 코드		54015450Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::Mode'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	트리거 입력 조건의 드라이브 트리거 입력 신호 선택 범위 밖		이벤트 코드		54015451Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::InputDrive'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(축 지정)		이벤트 코드		54015453Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Axis'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(버퍼 모드 선택)		이벤트 코드		54015454Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'BufferMode'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(방향 선택)		이벤트 코드	54015455Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Direction'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	모션 명령 재기동 불가능(반복 모드)		이벤트 코드	54015456Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Periodic'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	모션 명령 재기동 불가능(축 그룹 지정)		이벤트 코드		54015457Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'AxesGroup'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(약동 설정)		이벤트 코드		54015458Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Jerk'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(주축)		이벤트 코드	54015459Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Master'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	모션 명령 재기동 불가능(MasterOffset)		이벤트 코드	5401545AHex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MasterOffset'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	모션 명령 재기동 불가능(MasterScaling)		이벤트 코드	5401545B Hex		
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MasterScaling'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(MasterStartDistance)		이벤트 코드	5401545C Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterStartDistance'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(Continuous)		이벤트 코드	5401545D Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'Continuous'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(MoveMode)		이벤트 코드	5401545E Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'MoveMode'의 파라메터를 변경했습니다.				
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	잘못된 보조축 지정		이벤트 코드	5401545F Hex		
내용	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않는 변수		해당 명령으로 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	잘못된 축 지정		이벤트 코드	54015460 Hex		
내용	모션 제어 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않는 변수		해당 명령으로 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	잘못된 축 그룹 지정			이벤트 코드		54015461Hex
내용	모션 제어 명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지 않거나 사용 축 그룹이 아닙니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지 않는 변수		해당 명령으로 지정된 축 그룹을 존재하는 축 그룹 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
	명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 사용 축 그룹으로 설정되지 않음		해당 명령으로 지정된 축 그룹을 사용 축 그룹으로 설정하십시오.		입력 변수 'AxesGroup'에 지정하는 축 그룹은 사용 축 그룹으로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	잘못된 주축 지정			이벤트 코드		54015462Hex
내용	모션 제어 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않거나 동기 주축이 아닙니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Master'에 지정된 축이 존재하지 않는 변수		해당 명령으로 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
	MC_Phasing(주축 상대값 위상 보정) 명령의 경우, 입력 변수 'Master'에 지정된 축이 동기 주축이 아님		MC_Phasing(주축 상대값 위상 보정) 명령의 입력 변수 'Master'에 입력된 변수를 동기 명령의 주축에 지정된 축으로 수정하십시오.		MC_Phasing(주축 상대값 위상 보정) 명령의 입력 변수 'Master'에 입력된 변수를 동기 명령의 주축에 지정된 축으로 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(SlaveOffset)		이벤트 코드	54015463Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveOffset'을 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(SlaveScaling)		이벤트 코드	54015464Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveScaling'을 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(StartPosition)		이벤트 코드		54015465Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'StartPosition'을 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	원점 미확정 상태에서 명령 기동 이상		이벤트 코드		54015466Hex	
내용	원점이 확정되지 않은 상태에서 고속 원점 복귀 또는 보간 명령이 실행되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	원점이 확정되지 않은 상태에서 고속 원점 복귀가 실행됨		원점 복귀를 통해 원점을 확정하고 나서 해당 고속 원점 복귀를 실행하십시오.		원점 복귀를 통해 원점 확정 상태로 한 후 고속 원점 복귀 명령을 실행하십시오.	
	원점 미확정 상태를 포함한 구성축에 대해 보간 명령이 실행됨		원점 복귀를 통해 모든 구성축의 원점을 확정하고 나서 해당 보간 명령을 실행하십시오.		원점 복귀를 통해 모든 구성축의 원점을 확정하고 나서 보간 명령을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • 원점 복귀 실행 후에도 '현재 위치 변경' 명령을 실행하면 원점 미확정 상태가 되므로 다시 한번 원점 복귀를 실행해 원점 확정 상태로 해야 합니다.					

이벤트명	모션 명령 재기동 불가능(위치 종류)		이벤트 코드		54015467Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'ReferenceType'을 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	사용하지 않는 축 지정(주축)		이벤트 코드		54015468Hex	
내용	모션 제어 명령으로 지정된 주축이 사용하지 않는 축입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	모션 제어 명령으로 지정된 주축이 사용하지 않는 축		해당 명령으로 지정된 주축을 사용 축으로 수정하십시오.		모션 제어 명령으로 지정하는 주축에는 사용 축을 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	시작 위치 설정 범위 밖			이벤트 코드		54015469Hex
내용	모션 제어 명령의 입력 변수 'FirstPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	종료 위치 설정 범위 밖			이벤트 코드		5401546AHex
내용	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	잘못된 시작 위치/종료 위치 · 크기 관계(리니어 모드)		이벤트 코드		5401546BHex	
내용	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 입력 변수 'FirstPosition'에 지정된 파라메터보다 작은 값입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	카운트 모드가 리니어 모드일 때 명령의 입력 파라메터에서 'LastPosition'의 값이 'FirstPosition'의 값보다 작음		해당 명령으로 지정된 'LastPosition'의 값이 'FirstPosition'의 값 이상이 되도록 수정하십시오. 또는 카운트 모드를 로터리 모드로 변경하십시오.		해당 명령으로 지정된 'LastPosition'의 값이 'FirstPosition'의 값 이상이 되도록 프로그램을 작성하십시오. 또는 해당 축의 카운트 모드가 로터리 모드로 되어 있는지 확인하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축 동기 위치 설정 범위 밖		이벤트 코드		5401546CHex	
내용	모션 제어 명령의 입력 변수 'MasterSyncPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	종축 동기 위치 설정 범위 밖		이벤트 코드	5401546D Hex	
내용	모션 제어 명령의 입력 변수 'SlaveSyncPosition'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	트리거 입력 조건의 래치 ID 중복		이벤트 코드	5401546E Hex	
내용	모션 제어 명령으로 지정된 래치 ID가 여러 명령에서 중복되어 있습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동일한 래치 ID가 동시에 사용됨		같은 래치 ID가 해당 명령과 다른 명령에서 동시에 사용되지 않도록 프로그램을 수정하십시오. 다른 래치 ID를 사용하도록 하거나 같은 래치 ID를 사용하는 명령을 동시에 실행하지 않도록 하십시오. 또한 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령 실행 중에는 래치 기능 1/2가 모두 사용 중인 것으로 취급됩니다.	MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동시에 동일한 래치 ID를 사용하지 않도록 하십시오.
	MC_TouchProbe(외부 래치 유효) 명령 이외에서 사용 중인 래치를 MC_AbortTrigger(외부 래치 무효) 명령에서 중지하려고 함		외부 래치 유효 이외의 명령에서 사용 중인 래치를 외부 래치 무효 명령으로 중지하지 않도록 하십시오.	외부 래치 유효 이외의 명령에서 사용 중인 래치는 외부 래치 무효 명령을 실행하지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • 래치 ID를 변경하려면 다른 명령에서 해당 래치 ID가 동시에 사용 중이 아님을 확인하십시오.				

이벤트명	저크 오버라이드값 범위 밖			이벤트 코드		5401546F Hex
내용	모션 제어 명령의 입력 변수 'JerkFactor'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	가감속 속도 오버라이드값 범위 밖			이벤트 코드		54015470 Hex
내용	모션 제어 명령의 입력 변수 'AccFactor'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	시작 위치 방식 지정 범위 밖		이벤트 코드		54015471Hex	
내용	모션 제어 명령의 입력 변수 'StartMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	모션 명령 재기동 불가능(시작 위치 방식)		이벤트 코드		54015472Hex	
내용	모션 제어 명령의 재기동 시 변경할 수 없는 입력 변수 'StartMode'의 파라메터를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 재기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	사용하지 않는 축 지정(보조축)		이벤트 코드	54015474Hex	
내용	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축		해당 명령으로 지정된 축을 사용 축으로 설정하십시오. 또는 사용 축을 지정하도록 파라메터를 수정하십시오.	명령으로 지정하는 축은 사용 축으로 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	위치 지정 기어 지정값 이상		이벤트 코드	54015475Hex	
내용	모션 제어 명령에 입력된 속도/가속도/감속도에서는 동기 동작을 할 수 없습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령에 입력된 속도/가속도/감속도에서는 지정된 동기 동작을 수행할 수 없음		MC_GearInPos(위치 지정 기어 동작) 명령의 동작 사양에 기초하여 동기 동작이 가능한 설정이 되도록 프로그램을 수정하십시오.	사용하는 명령의 처리를 확인하고 동기 동작이 가능하도록 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	위치 지정 기어 주축 속도 제로		이벤트 코드	54015476Hex	
내용	모션 제어 명령의 기동 시 주축의 속도가 '0'입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 기동 시 주축의 속도가 '0'	해당 명령 기동 시에 주축의 속도가 '0'이 되지 않도록 프로그램을 수정하십시오.	명령 실행 시에 주축의 속도가 '0'이 되지 않도록 프로그램을 작성하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	목표 위치 설정 범위 밖		이벤트 코드	54015478Hex	
내용	모션 제어 명령의 입력 변수 'Position'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 파라메터가 입력 변수 범위 밖	해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.
	로터리 모드의 축에서 목표 위치가 링 설정의 범위 밖	로터리 모드인 축에서는 목표 위치를 링 설정의 범위 이내가 되도록 수정하십시오.	로터리 모드인 축에서는 목표 위치를 링 설정의 범위 이내가 되도록 하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	이동 거리 범위 밖		이벤트 코드	54015479Hex		
내용	모션 제어 명령의 입력 변수 'Distance'에 지정된 파라메터가 범위 밖 또는 'Distance' 값을 가산한 목표 위치가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함		해당 명령의 입력 변수 'Distance'에서 지정되는 입력 파라메터를 이동 거리나 목표 위치가 범위 밖이 되지 않도록 수정하십시오.		명령에서 이동 거리나 목표 위치가 범위 밖이 되지 않도록 프로그램을 작성하십시오.	
	리니어 모드인 축에서 이동 거리를 가산한 목표 위치를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함					
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	캠 테이블 시작점 위치 설정 범위 밖		이벤트 코드	5401547AHex		
내용	모션 제어 명령의 입력 변수 'StartPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	캠 동작(주축 추종) 시작 위치 설정 범위 밖		이벤트 코드		5401547BHex	
내용	모션 제어 명령의 입력 변수 'MasterStartDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	원호 보간 반경 지정 이상		이벤트 코드		5401547CHex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식 지정 시 지정된 반경으로 원호의 궤적을 작성할 수 없습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 반경 지정 방식이 지정되고, 지정된 반경으로 원호의 궤적을 작성할 수 없음		원호의 궤적을 작성할 수 있도록 반경을 수정하십시오.		사용하는 명령의 처리를 확인하여 원호의 궤적을 작성할 수 있도록 반경을 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	원호 보간 반경 오버 플로우		이벤트 코드	5401547DHex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반경이 최대값을 벗어났습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반경을 펄스 단위로 변환한 경우 40비트의 범위를 벗어남		해당 명령의 동작 사양에 기초하여 원호의 반경을 펄스 단위로 변환했을 때 40비트의 범위를 넘지 않도록 입력 파라메터를 수정하십시오. 통과점을 지정하는 경우: 시작점, 통과점, 종료점 중심점을 지정하는 경우: 시작점, 종료점, 중심점	사용하는 명령의 처리를 확인하고 원호의 반경을 펄스 단위로 변환했을 때 40비트의 범위를 넘지 않도록 입력 파라메터를 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	• 이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다. • 반경 지정 방식인 경우에 반경이 최대값을 넘으면 '통과점 위치/중심 위치/반경 지정 범위 외' 이상이 발생합니다.				

이벤트명	원호 축 지정 범위 밖		이벤트 코드	5401547EHex	
내용	모션 제어 명령의 입력 변수 'CircAxes'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	해당 명령에 입력된 파라메터를 입력 변수의 범위를 넘지 않도록 설정하십시오.	
	'CircAxes'에 지정된 축이 축 그룹 설정의 구성축에 포함되지 않음		'CircAxes'에 지정하는 축을 축 그룹 구성에 포함하도록 설정을 변경하십시오.	'CircAxes'에 지정하는 축을 축 그룹 구성에 포함하십시오.	
	'CircAxes'의 2축에 동일한 축을 지정함		'CircAxes'에 지정하는 2축이 다른 축이 되도록 수정하십시오.	'CircAxes'에 지정하는 2축이 서로 다르게 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	보조축/종축 · 축 번호가 오름차순이 아님		이벤트 코드		5401547F Hex	
내용	모션 제어 명령으로 지정한 입력 변수 'Auxiliary'와 'Slave'의 파라메터 값이 오름차순으로 정렬되지 않았습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 오름차순이 아님		해당 명령의 'Auxiliary'와 'Slave'의 입력 파라메터에 지정하는 축 번호가 오름차순이 되도록 수정하십시오.		'Auxiliary'와 'Slave'에 지정하는 축 번호가 오름차순이 되도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	캠 테이블 속성 갱신 중 데이터 오름차순 이상		이벤트 코드		54015480 Hex	
내용	유효한 데이터 수 계산 중에 위상이 오름차순이 아닌 것으로 판명되었습니다. 또는 계산한 결과, 유효 데이터 수가 '0'인 것으로 판명되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 실행 중
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	유효한 데이터 수 계산 중에 위상이 오름차순이 아닌 것으로 판명됨		캠 데이터를 위상이 오름차순이 되도록 수정하십시오.		캠 테이블 데이터의 위상이 오름차순이 되도록 하십시오.	
	계산한 결과, 유효 데이터 수가 '0'인 것으로 판명됨		캠 테이블 데이터를 '0' 이외의 위상도 포함하도록 수정하십시오.		캠 테이블 데이터를 '0' 이외의 위상도 포함하도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	MC 설정 쓰기의 대상 범위 밖			이벤트 코드	54015481Hex	
내용	모션 제어 명령의 입력 변수 'Target'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축 이동 거리 지정 범위 밖			이벤트 코드	54015482Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축 가속 이동 거리 지정 범위 밖		이벤트 코드	54015483Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistanceInACC'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축 감속 이동 거리 지정 범위 밖		이벤트 코드	54015484Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistanceInDEC'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	실행 모드 선택 범위 밖		이벤트 코드	54015487Hex		
내용	모션 제어 명령의 입력 변수 'ExecutionMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	축간 편차 허용값 범위 밖		이벤트 코드	54015488Hex		
내용	모션 제어 명령의 입력 변수 'PermittedDeviation'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	통과점 위치/중심 위치/반경 지정 범위 밖		이벤트 코드	54015489Hex	
내용	모션 제어 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식		명칭
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지
	통과점 또는 중심으로 지정된 경우 'AuxPoint'의 값을 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.
	반경으로 지정된 경우 'AuxPoint[0]'의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함				
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	종료점 지정 범위 밖		이벤트 코드	5401548AHex	
내용	모션 제어 명령의 입력 변수 'EndPoint'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식		명칭
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지
	명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	종축 이동 거리 지정 범위 밖		이벤트 코드	5401548B Hex		
내용	모션 제어 명령의 입력 변수 'SlaveDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터를 펄스 단위로 변환한 경우 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	위상 보정량 범위 밖		이벤트 코드	5401548C Hex		
내용	모션 제어 명령의 입력 변수 'PhaseShift'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터의 절대값을 펄스 단위로 변환한 경우 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	피딩 거리 범위 밖		이벤트 코드	5401548D Hex	
내용	모션 제어 명령의 입력 변수 'FeedDistance'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터의 절대값을 펄스 단위로 변환한 경우 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	보조축/종축 동일		이벤트 코드	5401548E Hex	
내용	모션 제어 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정된 축이 동일합니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 동일		해당 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정되는 축이 다르도록 파라메터를 수정하십시오.	명령 입력 변수의 보조축과 종축에는 다른 축을 지정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	상대 위치 선택 범위 밖			이벤트 코드		5401548F Hex	
내용	모션 제어 명령의 입력 변수 'Relative'에 지정된 파라메터가 범위 밖입니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL			축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.			명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	캠 트랜지션 지정 선택 범위 밖			이벤트 코드		54015490 Hex	
내용	모션 제어 명령의 입력 변수 'CamTransition'에 지정된 파라메터가 범위 밖입니다.						
발생원	PLC 기능 모듈		발생원 상세 정보	명령		검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–		로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료			
시스템 정의 변수	변수명		데이터 형식			명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL			축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책			재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.			명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)						
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.						

이벤트명	동기 제어 해제 모드 선택 범위 밖		이벤트 코드		54015491Hex	
내용	모션 제어 명령의 입력 변수 'OutMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	외부 래치 유효 명령 실행 불가능		이벤트 코드		54015492Hex	
내용	CSP 모드 이외의 제어 모드일 때 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정하여 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	CSP 모드 이외의 제어 모드일 때 또는 서보 OFF일 때 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정하여 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행함		'_mcImmediateStop(즉시 정지)'을 지정하고 드라이버 모드의 외부 래치 유효 명령을 실행하는 경우는 제어 모드를 CSP 모드로 하십시오.		'_mcImmediateStop(즉시 정지)'을 지정하고 드라이버 모드의 외부 래치 유효 명령을 실행하는 경우는 제어 모드를 CSP 모드로 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	주축 오프셋 설정 범위 밖		이벤트 코드	54015493Hex		
내용	모션 제어 명령의 입력 변수 'MasterOffset'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	종축 오프셋 설정 범위 밖		이벤트 코드	54015494Hex		
내용	모션 제어 명령의 입력 변수 'SlaveOffset'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터를 펄스 단위로 변환한 경우 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	명령 현재 위치 카운트 선택 범위 밖		이벤트 코드	54015495Hex	
내용	모션 제어 명령의 입력 변수 'CmdPosMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축 기어비 분자 범위 밖		이벤트 코드	54015496Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumeratorMaster'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축 기어비 분모 범위 밖		이벤트 코드	54015497Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominatorMaster'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	보조축 기어비 분자 범위 밖		이벤트 코드	54015498Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumeratorAuxiliary'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	보조축 기어비 분모 범위 밖		이벤트 코드	54015499Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominatorAuxiliary'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	주축 위치 종류 선택 범위 밖		이벤트 코드	5401549AHex	
내용	모션 제어 명령의 입력 변수 'ReferenceTypeMaster'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	보조축 위치 종류 선택 범위 밖		이벤트 코드	5401549B Hex	
내용	모션 제어 명령의 입력 변수 'ReferenceTypeAuxiliary'에 지정된 파라메터가 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	목표 위치 링 카운터 범위 밖		이벤트 코드	5401549C Hex	
내용	실행된 명령의 목표 위치가 링 카운터의 범위 밖이므로 동작할 수 없습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	링 카운터 범위에 0이 포함되지 않도록 설정되었을 때 고속 원점 복귀를 실행함		링 카운터 범위에 0이 포함되지 않도록 설정되었을 때는 고속 원점 복귀를 실행할 수 없습니다. 고속 원점 복귀를 실행하지 않도록 프로그램을 수정하십시오. 또는 해당 축의 링 카운터 범위에 0이 포함되도록 설정을 변경하십시오.	링 카운터 범위에 0이 포함되지 않도록 설정되었을 때는 고속 원점 복귀를 실행할 수 없습니다. 고속 원점 복귀를 실행하지 않도록 프로그램을 작성하십시오. 또는 해당 축의 링 카운터 범위에 0이 포함되도록 설정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	축 그룹 구성축 설정 범위 밖		이벤트 코드	5401549D Hex *1		
내용	모션 제어 명령의 입력 변수 'Axes'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.01에서 추가된 에러 코드(16#549D)입니다.

이벤트명	축 사용 설정 범위 밖		이벤트 코드	5401549EHex *1		
내용	모션 제어 명령의 입력 변수 'AxisUse'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 에러 코드(16#549E)입니다.

이벤트명	원점 복귀 파라메터 설정 범위 밖		이벤트 코드		54015700Hex*1	
내용	모션 제어 명령의 입력 변수 'HomingParameter'에 지정된 파라메터가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 에러 코드(16#5700)입니다.

이벤트명	축 미사용 전환 이상		이벤트 코드		54015702Hex*1	
내용	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함		이상을 해제하고 축이 정지 중일 때, 명령 속도 포화 상태가 아닐 때 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다. 축 변수 'Details.VelLimit'가 TRUE이면 해당 축은 명령 속도 포화 상태입니다.		축이 정지 중일 때, 명령 속도 포화 상태가 아닐 때 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 에러 코드(16#5702)입니다.

이벤트명	축 사용 전환 불가			이벤트 코드	54015703Hex*1	
내용	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함		사용하는 CPU 유니트의 사용 실제 축 최대 수를 넘지 않도록 프로그램을 수정하십시오.		사용하는 CPU 유니트의 사용 실제 축 최대 수를 넘지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5703)입니다.

이벤트명	축 사용 전환 시 모션 제어 파라메터 설정 이상		이벤트 코드	54015720Hex*1		
내용	사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못되었습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_ChangeAxisUse(축 사용 변경) 명령을 사용하여 미사용 축에서 사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못됨		Sysmac Studio를 사용하여 이상이 발생한 축의 [축 사용]을 [사용 축]으로 변경하고, 이상 위치를 확인한 후 수정하십시오. 이상이 발생하지 않으면 [미사용 축]으로 다시 변경하고 다운로드하십시오.		[사용 축] 설정에서 정상적으로 동작하는지 확인한 후 [미사용 축]으로 설정하고 다운로드하십시오.	
	모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨		Sysmac Studio에서 모션 제어 파라메터 설정을 다운로드하십시오.		파라메터 설정 저장 중에 전원을 OFF하지 마십시오.	
	비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함		상기의 처리를 실시해도 이 이상이 해제되지 않는 경우는 CPU 유니트를 교환하십시오.		없음	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1 CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 에러 코드(16#5720)입니다.

이벤트명	축 사용 전환 시 필수 프로세스 데이터 개체 미설정		**이벤트 코드**		54015721Hex*1	
내용	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 설정되어 있지 않습니다.					
발생원	PLC 기능 모듈		**발생원 상세 정보**	명령	**검출 타이밍**	명령 기동 시
이상의 속성	**중요도**	감시 정보	**복구 방법**	−	**로그 카테고리**	시스템
발생의 영향	**사용자 프로그램**	계속	**동작**	해당 명령을 사양대로 종료		
시스템 정의 변수	**변수명**		**데이터 형식**		**명칭**	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	**발생 요인(추정 원인)**		**처리 대책**		**재발 방지**	
	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 PDO 매핑 설정에 설정되어 있지 않음		Sysmac Studio의 [PDO 매핑 설정 편집]에서 이상이 발생한 축의 필수 개체를 설정하십시오. 필수 개체에 대해서는 📖 "NJ 시리즈 명령 참조 설명서 모션편 (SBCE-364)"을 참조하십시오.		[사용 축] 설정에서 정상적으로 동작하는지 확인한 후 [미사용 축]으로 설정하고 다운로드하십시오.	
	모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨		Sysmac Studio에서 모션 제어 파라메터 설정을 다운로드하십시오.		파라메터 설정 저장 중에 전원을 OFF하지 마십시오.	
	비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함		상기의 처리를 실시해도 이 이상이 해제되지 않는 경우는 CPU 유니트를 교환하십시오.		없음	
	[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령이 실행됨		[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해서 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하지 않도록 프로그램을 수정하십시오.		[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해서 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하지 않도록 프로그램을 수정하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 에러 코드(16#5721)입니다.

이벤트명	피드백 위치 오버 플로우/언더 플로우 발생 중		이벤트 코드		54015722Hex*1	
내용	피드백 위치 오버 플로우/언더 플로우 발생 중에 실행할 수 없는 명령을 기동했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	피드백 위치 오버 플로우 또는 언더 플로우 발생 중에 실행할 수 없는 명령을 기동함		이상 해제를 실행하고 '현재 위치 변경' 또는 '원점 복귀'를 통해 오버 플로우 또는 언더 플로우 상태를 해제하십시오.		오버 플로우 또는 언더 플로우가 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5722)입니다.

이벤트명	스위치 구조체의 트랙 번호 설정 범위 밖		이벤트 코드		54015723Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5723)입니다.

이벤트명	스위치 구조체의 ON 시작 위치 설정 범위 밖		이벤트 코드		54015724Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'FirstOnPosition'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5724)입니다.

이벤트명	스위치 구조체의 ON 종료 위치 설정 범위 밖		이벤트 코드		54015725Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'LastOnPosition'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5725)입니다.

이벤트명	스위치 구조체의 방향 선택 범위 밖		이벤트 코드		54015726Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'AxisDirection'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5726)입니다.

이벤트명	스위치 구조체의 스위치 모드 선택 범위 밖		이벤트 코드		54015727Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'CamSwitchMode'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL			
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5727)입니다.

이벤트명	스위치 구조체의 ON 시간 설정 범위 밖		이벤트 코드	54015728Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'Duration'의 값이 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].Obsr.Active		BOOL	축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.	명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5728)입니다.

이벤트명	트랙 옵션 구조체의 ON 시각 보정 설정 범위 밖		이벤트 코드	54015729Hex*1	
내용	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OnCompensation'의 값이 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].Obsr.Active		BOOL	축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.	명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5729)입니다.

이벤트명	트랙 옵션 구조체의 OFF 시각 보정 설정 범위 밖		이벤트 코드		5401572A Hex*1	
내용	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OffCompensation'의 값이 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572A)입니다.

이벤트명	스위치 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드		5401572B Hex*1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL			
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572B)입니다.

이벤트명	출력 신호 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드		5401572C Hex*1	
내용	모션 제어 명령의 입출력 변수 'Outputs'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572C)입니다.

이벤트명	트랙 옵션 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드		5401572D Hex*1	
내용	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572D)입니다.

이벤트명	출력 신호와 트랙 옵션의 배열 요소 수 불일치		이벤트 코드	5401572E Hex*1		
내용	모션 제어 명령의 입출력 변수 'Outputs'와 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 일치하지 않습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수에 지정된 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하지 않음		해당 명령의 입출력 변수에 지정된 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하도록 수정하십시오.		명령의 입출력 변수에 지정하는 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572E)입니다.

이벤트명	모션 명령 다중 기동 불가능(주축)		이벤트 코드	5401572F Hex*1		
내용	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경		해당 명령의 다중 기동 시 입출력 변수 'Master'의 값을 변경하지 않도록 프로그램을 수정하십시오.		해당 명령의 다중 기동 시 입출력 변수 'Master'의 값을 변경하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#572F)입니다.

이벤트명	모션 명령 다중 기동 불가능(위치 종류 선택)		이벤트 코드	54015730 Hex*1		
내용	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경		해당 명령의 다중 기동 시 입출력 변수 'ReferenceType'의 값을 변경하지 않도록 프로그램을 수정하십시오.		해당 명령의 다중 기동 시 입출력 변수 'ReferenceType'의 값을 변경하지 않도록 프로그램을 작성하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5730)입니다.

이벤트명	스위치 구조체의 동일 트랙 지정 수 범위 밖		이벤트 코드	54015731Hex*1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 동일 트랙 번호 수가 범위 밖입니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 스위치 구조체 형식 변수의 'TrackNumber'에 지정된 동일 트랙 번호의 수가 1트랙에 지정 가능한 개수의 범위 밖		'TrackNumber'에 지정된 동일 트랙 번호의 수가 1트랙에 지정할 수 있는 개수를 넘지 않도록 값을 수정하십시오.		'TrackNumber'에 지정된 동일 트랙 번호의 수가 1트랙에 지정할 수 있는 개수를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 에러 코드(16#5731)입니다.

이벤트명	목표 위치 정방향 소프트웨어 리미트 초과		**이벤트 코드**		54016440Hex	
내용	지정한 위치가 정방향 소프트웨어 리미트의 범위 밖입니다.					
발생원	PLC 기능 모듈		**발생원 상세 정보**	명령	**검출 타이밍**	명령 기동 시
이상의 속성	**중요도**	감시 정보	**복구 방법**	−	**로그 카테고리**	시스템
발생의 영향	**사용자 프로그램**	계속	**동작**	해당 명령을 사양대로 종료		

시스템 정의 변수	**변수명**	**데이터 형식**	**명칭**
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리	**발생 요인(추정 원인)**	**처리 대책**	**재발 방지**
	명령의 입력 변수 'Position'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 밖	해당 명령의 입력 변수 'Position'에서 지정된 파라메터를 정방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	명령의 입력 변수 'Position'에 지정된 입력 파라메터가 정방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.
	시작 위치가 정방향 소프트웨어 리미트 범위 밖일 때 소프트웨어 리미트의 범위와 반대 방향으로 동작을 지정한 명령 실행	해당 명령에서의 이동 방향이 정방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 수정하십시오.	시작 위치가 정방향 소프트웨어 리미트 범위 밖이면 이동 방향이 정방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 작성하십시오.
	통과점을 지정한 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 밖	해당 명령의 입력 변수 'AuxPoint'에서 지정된 파라메터를 정방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.

부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.

이벤트명	목표 위치 역방향 소프트웨어 리미트 초과		이벤트 코드	54016441Hex	
내용	지정한 위치가 역방향 소프트웨어 리미트의 범위 밖입니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 변수 'Position'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 밖		해당 명령의 입력 변수 'Position'에서 지정된 파라메터를 역방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	명령의 입력 변수 'Position'에 지정된 입력 파라메터가 역방향 소프트웨어 리미트의 범위 이내가 되도록 설정하십시오.	
	시작 위치가 역방향 소프트웨어 리미트 범위 밖일 때 소프트웨어 리미트의 범위와 반대 방향으로 동작을 지정한 명령 실행		해당 명령에서의 이동 방향이 역방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 수정하십시오.	시작 위치가 역방향 소프트웨어 리미트 범위 밖이면 이동 방향이 역방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 작성하십시오.	
	통과점을 지정한 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 밖		해당 명령의 입력 변수 'AuxPoint'에서 지정된 파라메터를 역방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	명령 위치 오버 플로우/언더 플로우 발생 중		이벤트 코드	54016442Hex	
내용	명령 위치 오버 플로우/언더 플로우 발생 중에 위치 결정, 오버 플로우/언더 플로우 방향의 명령 또는 방향을 지정할 수 없는 명령을 실행했습니다.				
발생원	PLC 기능 모듈	발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 위치 오버 플로우/언더 플로우 발생 중에 다음을 수행함 • 위치 결정 명령을 실행 • 오버 플로우/언더 플로우 방향의 연속 제어 명령을 실행 • 방향을 확인할 수 없는 명령(동기 기능, 토크 제어)을 실행		이상 해제를 실행하고 원점 복귀 또는 현재 위치 사전 설정을 통해 오버 플로우/언더 플로우 상태를 해제하십시오.	오버 플로우/언더 플로우가 발생하지 않도록 설정하십시오.	
부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호				
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.				

이벤트명	정방향 한계 입력 중			이벤트 코드	54016443Hex	
내용	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행 또는 정방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않은 명령을 실행. 정방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행		이상을 해제하고 역방향으로 복귀 동작을 실행하십시오. 축 그룹 동작 명령의 실행 중에 이상이 발생한 경우는 이상이 발생한 축 그룹을 무효로 한 후 상기 복귀 동작을 수행하십시오. 다시 이 이상이 발생하면 정방향 한계 신호의 연결 상태, 정방향 한계 입력의 논리 설정, 기동 명령의 실행 조건을 확인하고 오류를 수정하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정 양쪽을 확인하십시오.		정방향 한계 신호의 연결 상태, 정방향 한계 입력의 논리 설정, 기동 명령의 실행 조건에 오류가 없는 것을 확인하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정 양쪽을 확인하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	역방향 한계 입력 중			이벤트 코드	54016444Hex	
내용	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행 또는 역방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않은 명령을 실행. 역방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행		이상을 해제하고 정방향으로 복귀 동작을 실행하십시오. 축 그룹 동작 명령의 실행 중에 이상이 발생한 경우는 이상이 발생한 축 그룹을 무효로 한 후 상기 복귀 동작을 수행하십시오. 다시 이 이상이 발생하면 역방향 한계 신호의 연결 상태, 역방향 한계 입력의 논리 설정, 기동 명령의 실행 조건을 확인하고 오류를 수정하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정 양쪽을 확인하십시오.		역방향 한계 신호의 연결 상태, 역방향 한계 입력의 논리 설정, 기동 명령의 실행 조건에 오류가 없는 것을 확인하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정 양쪽을 확인하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

이벤트명	서보 주회로 전원 OFF 상태		이벤트 코드	54017422Hex		
내용	서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행했습니다.					
발생원	PLC 기능 모듈		발생원 상세 정보	명령	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복구 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령을 사양대로 종료		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	서보 드라이버의 주회로 전원 'OFF' 상태에서 서보 ON 실행		이 이상이 발생한 축의 서보 드라이버 주회로 전원을 'ON'한 후 서보 ON을 실행하십시오.		서보 드라이버의 주회로 전원을 'ON'한 후 서보 ON을 실행하십시오.	
부속 정보	부속 정보 1: 이상의 발생 위치 부속 정보 2: 이상의 발생 위치에 대한 자세한 회로 번호. 프로그램 섹션인 경우는 섹션 선두에서의 회로 번호. ST인 경우는 행 번호 부속 정보 3: 이상이 발생한 명령의 이름, 명령 인스턴스명. 후보가 여러 개인 경우는 복수 표시. 확인할 수 없는 경우에는 표시 안 함 부속 정보 4: 확장 에러 코드(ErrorIDEx)					
주의점/비고	이상 발생 후에 프로그램을 변경한 경우는 부속 정보가 정확하게 표시되지 않을 가능성이 있습니다.					

A-4 이벤트 태스크에서 사용할 수 없는 명령

이벤트 태스크는 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 태스크 주기마다 반복 실행되는 태스크가 아닙니다. 따라서 복수의 태스크 주기에 걸쳐 실행되는 명령을 포함하는 사용자 프로그램은 이벤트 태스크에 할당할 수 없습니다.
아래 표의 명령이 복수의 태스크 주기에 걸쳐 실행되는 명령입니다. 이러한 명령은 이벤트 태스크에 할당하는 사용자 프로그램 내에서 사용하지 마십시오. 사용하면 빌드 시에 이상 상태가 됩니다.

종류	명령	명칭	페이지
스택/테이블 명령	RecSort	레코드 정렬	2-497
아날로그 제어 명령	PIDAT	자동 조정 가능한 PID 연산	2-642
	AC_StepProgram	스텝 프로그램	2-712
시스템 제어 명령	ResetPLCError	PLC 이상 해제	2-756
	ResetCJBError	I/O 버스 이상 해제	2-762
	ResetMCError	모션 제어 이상 해제	2-768
	ResetECError	EtherCAT 이상 해제	2-775
	ResetUnit	유니트 재시작	2-782
	RestartNXUnit	NX 유니트 재시작	2-788
	NX_ChangeWriteMode	NX 유니트 쓰기 모드 변경	2-793
	NX_SaveParam	NX 유니트 파라메터 저장	2-797
EtherCAT 통신 명령	EC_CoESDOWrite	CoE SDO 쓰기	2-804
	EC_CoESDORead	CoE SDO 읽기	2-807
	EC_StartMon	EtherCAT 패킷 모니터 기능 시작	2-813
	EC_StopMon	EtherCAT 패킷 모니터 기능 정지	2-819
	EC_SaveMon	EtherCAT 패킷 저장	2-821
	EC_CopyMon	EtherCAT 패킷 전송	2-823
	EC_DisconnectSlave	EtherCAT 슬레이브 이탈	2-825
	EC_ConnectSlave	EtherCAT 슬레이브 재가입	2-833
	EC_ChangeEnable Setting	EtherCAT 슬레이브의 유효/무효 전환	2-835
	NX_WriteObj	NX 개체 쓰기	2-854
	NX_ReadObj	NX 개체 읽기	2-869

종류	명령	명칭	페이지
EtherNet/IP 통신 명령	CIPOpen	CIPClass3(Large_Forward_Open) 연결 확립	2-878
	CIPOpenWithDataSize	CIPClass3 연결 확립(크기 지정 있음)	2-887
	CIPRead	변수 읽기(Class3 Explicit 메시지)	2-890
	CIPWrite	변수 쓰기(Class3 Explicit 메시지)	2-895
	CIPSend	임의 Explicit 메시지 송신(Class3)	2-901
	CIPClose	CIP Class3 연결 중단	2-905
	CIPUCMMRead	변수 읽기(UCMM Explicit 메시지)	2-907
	CIPUCMMWrite	변수 쓰기(UCMM Explicit 메시지)	2-912
	CIPUCMMSend	임의 Explicit 메시지 송신(UCMM)	2-919
	SktUDPCreate	UDP 소켓 생성	2-929
	SktUDPRcv	UDP 소켓 수신	2-936
	SktUDPSend	UDP 소켓 송신	2-939
	SktTCPAccept	TCP 소켓 허용	2-942
	SktTCPConnect	TCP 소켓 연결	2-945
	SktTCPRcv	TCP 소켓 수신	2-953
	SktTCPSend	TCP 소켓 송신	2-956
	SktGetTCPStatus	TCP 소켓의 Status 읽기	2-959
	SktClose	TCP/UDP 소켓 클로즈	2-962
	SktClearBuf	TCP/UDP 소켓 수신 버퍼 클리어	2-965
	ChangeIPAdr	IP 주소 변경	2-968
	ChangeFTPAccount	FTP 계정 변경	2-977
	ChangeNTPServerAdr	NTP 서버 수신처 변경	2-980
시리얼 통신 명령	ExecPMCR	프로토콜 매크로	2-986
	SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	2-1000
	SerialRcv	시리얼 커뮤니케이션 유니트 시리얼 포트 입력	2-1011
	SerialRcvNoClear	시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함	2-1011
	SendCmd	명령 송신	2-1026
SD 메모리 카드 명령	FileWriteVar	변수 파일 쓰기	2-1040
	FileReadVar	변수 파일 읽기	2-1046
	FileOpen	파일 열기	2-1051
	FileClose	파일 닫기	2-1055
	FileSeek	파일 검색	2-1058
	FileRead	파일 읽기	2-1061
	FileWrite	파일 쓰기	2-1069
	FileGets	문자열 읽기	2-1077
	FilePuts	문자열 쓰기	2-1084
	FileCopy	파일 복사	2-1092
	FileRemove	파일 삭제	2-1100
	FileRename	파일명 변경	2-1105
	DirCreate	디렉토리 작성	2-1110
	DirRemove	디렉토리 삭제	2-1113
타임 스탬프 명령	NX_DOutTimeStamp	타임 스탬프 디지털 출력 쓰기	2-1118
	NX_AryDOut TimeStamp	타임 스탬프 디지털 출력 배열 쓰기	2-1124

A-5 NX 메시지 통신 이상 관련 명령

다음 명령을 한번에 여러 개 실행하면 'NX 메시지 통신 이상'이라는 이상이 발생할 가능성이 있습니다. 'NX 메시지 통신 이상'이 발생한 경우는 다음 명령의 실행 수를 줄이십시오. 'NX 메시지 통신 이상'이 발생하는 조건은 통신 부하 등에 따라 달라집니다.

종류	명령	이름	페이지
시스템 제어 명령	RestartNXUnit	NX 유니트 재시작	2-788
	NX_ChangeWriteMode	NX 유니트 쓰기 모드 변경	2-793
	NX_SaveParam	NX 유니트 파라메터 저장	2-797
EtherCAT 통신 명령	NX_WriteObj	NX 개체 쓰기	2-854
	NX_ReadObj	NX 개체 읽기	2-869

버전 관련 정보

NX 메시지 통신 이상은 CPU 유니트 Ver.1.05 이상, Sysmac Studio Ver.1.06 이상에서 발생하는 이상입니다.

A-6 SDO Abort 코드 목록

참고 정보로, EtherCAT 통신의 SDO Abort 코드가 정리되어 있습니다. 실제 통신에서 사용되는 Abort 코드는 슬레이브 측에 의해 규정됩니다. 프로그래밍할 때는 슬레이브 측의 설명서를 참조하십시오.

값	의미
16#05030000	토글 비트 변화 없음
16#05040000	SDO 프로토콜 타임아웃
16#05040001	무효/불확실한 클라이언트/서버 명령 지정자
16#05040005	메모리 범위를 벗어남
16#06010000	개체에 대한 미지원 액세스
16#06010001	쓰기 전용 개체에 대한 읽기 액세스
16#06010002	읽기 전용 개체에 대한 쓰기 액세스
16#06020000	Object Dictionary에 존재하지 않는 개체
16#06040041	개체를 PDO에 매핑 불가능
16#06040042	매핑된 개체 수/길이가 PDO 길이 초과
16#06040043	일반적인 파라메터 불일치
16#06040047	장치의 일반적인 내부 불일치
16#06060000	하드웨어 에러에 의한 액세스 실패
16#06070010	데이터 형식 불일치, 서비스 파라메터 길이 불일치
16#06070012	데이터 형식 불일치, 서비스 파라메터 길이가 너무 김
16#06070013	데이터 형식 불일치, 서비스 파라메터 길이가 너무 짧음
16#06090011	하위 인덱스가 존재하지 않음
16#06090030	파라메터 값이 범위를 벗어남(쓰기 액세스만 해당)
16#06090031	기록된 파라메터의 값이 너무 큼
16#06090032	기록된 파라메터의 값이 너무 작음
16#06090036	최대값이 최소값보다 작음
16#08000000	일반적인 에러
16#08000020	데이터를 응용 프로그램에 전송/저장할 수 없음
16#08000021	로컬 제어*1로 인해 데이터를 응용 프로그램에 전송/저장할 수 없음
16#08000022	현재의 장치 상태에서는 데이터를 응용 프로그램에 전송/저장할 수 없음
16#08000023	Object Dictionary 동적 생성 실패 또는 Object Dictionary가 존재하지 않음

*1. 슬레이브 고유의 내부 상태 문제

인용원 EtherCAT 사양서 Part6 응용 프로그램 계층 프로토콜 사양
문서 번호: ETG.1000.6 S(R) V1.0.2(J02)

A-7 버전 관련 정보

여기에는 CPU 유니트와 Sysmac Studio의 각 버전 갱신에 의해 사양이 변경된 명령과 신규 명령의 목록이 나와 있습니다.
또한 Sysmac StudioV1.02에서 발생하는 에러 메시지 '명령이 의도하지 않은 동작을 일으킬 가능성이 있습니다.'가 표시되었을 때의 대처 방법에 대해 설명합니다.

버전 갱신에 의해 사양이 변경된 명령 및 신규 명령 목록

명령에는 CPU 유니트와 Sysmac Studio의 각 버전에 따라 대응 상황이나 사양이 다른 경우가 있습니다. 해당 내용이 아래에 나와 있습니다.
CPU 유니트와 Sysmac Studio의 양쪽 버전에 기재된 명령은 양쪽 모두 해당 버전 이상이어야 한다는 의미입니다.

종류	명령	명칭	신규/사양 변경	버전		페이지
				CPU 유니트	Sysmac Studio	
ST 구문 명령	FOR	반복	사양변경	–	Ver.1.08	2-41
시퀀스 입력 명령	R_TRIG	상승 미분	사양 변경	Ver.1.02	–	2-44
시퀀스 출력 명령	RS	리셋 우선 유지	사양 변경	–	Ver.1.03	2-50
	SR	세트 우선 유지	사양 변경	–	Ver.1.03	2-52
비교 명령	EQ,=	비교 EQ	사양 변경	–	Ver.1.02	2-94
	NE,<>	비교 NE	사양 변경	–	Ver.1.02	2-96
	LT,<	비교 LT	사양 변경	–	Ver.1.02	2-98
	LE,<=	비교 LE	사양 변경	–	Ver.1.02	2-98
	GT,>	비교 GT	사양 변경	–	Ver.1.02	2-98
	GE,>=	비교 GE	사양 변경	–	Ver.1.02	2-98
	ZoneCmp	영역 비교	사양 변경	Ver.1.01	Ver.1.02	2-110
카운터 명령	CTD	감산 카운터	사양 변경	–	Ver.1.03	2-142
	CTD_**	감산 카운터 그룹	사양 변경	–	Ver.1.03	2-144
	CTU	가산 카운터	사양 변경	–	Ver.1.03	2-146
	CTU_**	가산 카운터 그룹	사양 변경	–	Ver.1.03	2-148
	CTUD	가역 카운터	사양 변경	–	Ver.1.03	2-151
	CTUD_**	가역 카운터 그룹	사양 변경	–	Ver.1.03	2-155
데이터 형식 변환 명령	EnumToNum	열거 형식→정수 변환	신규	Ver.1.02	Ver.1.03	2-299
	NumToEnum	정수→열거 형식 변환	신규	Ver.1.02	Ver.1.03	2-301
선택 명령	SEL	비트 선택	사양 변경	Ver.1.02	Ver.1.03	2-316
	MUX	멀티플렉서	사양 변경	Ver.1.02	Ver.1.03	2-318
	AryMax	배열 변수의 최대 값 검색	사양 변경	Ver.1.01	Ver.1.02	2-329
	AryMin	배열 변수의 최소 값 검색	사양 변경	Ver.1.01	Ver.1.02	2-329
	ArySearch	배열 검색	사양 변경	Ver.1.01	Ver.1.02	2-332
이동 명령	SHL	N비트 좌측 이동	사양 변경	Ver.1.02	Ver.1.03	2-378

종류	명령	명칭	신규/사양 변경	버전		페이지
				CPU 유니트	Sysmac Studio	
이동 명령	SHR	N비트 우측 이동	사양 변경	Ver.1.02	Ver.1.03	2-378
	ROL	N비트 좌회전	사양 변경	Ver.1.02	Ver.1.03	2-382
	ROR	N비트 우회전	사양 변경	Ver.1.02	Ver.1.03	2-382
데이터 변환 명령	UTF8ToSJIS	문자 코드 변환 (UTF-8→SJIS)	신규	Ver.1.01	Ver.1.02	2-404
	SJISToUTF8	문자 코드 변환 (SJIS→UTF-8)	신규	Ver.1.01	Ver.1.02	2-406
	PWLApproxNoLineChk	꺾은선 근사 변환 (꺾은선 데이터 확인 없음)	신규	Ver.1.03	Ver.1.04	2-408
	PWLLineChk	꺾은선 데이터 확인	신규	Ver.1.03	Ver.1.04	2-413
스택/테이블 명령	RecSearch	레코드 검색	사양 변경	Ver.1.01	Ver.1.02	2-487
	RecRangeSearch	범위 지정 레코드 검색	사양 변경	Ver.1.01	Ver.1.02	2-492
	RecSort	레코드 정렬	사양 변경	Ver.1.01	Ver.1.02	2-497
	RecNum	레코드 수 취득	사양 변경	Ver.1.01	Ver.1.02	2-502
				Ver.1.02	Ver.1.03	
	RecMax	레코드 최대값 검색	사양 변경	Ver.1.01	Ver.1.02	2-504
	RecMin	레코드 최소값 검색	사양 변경	Ver.1.01	Ver.1.02	2-504
문자열 명령	AddDelimiter	구분 문자 부가 문자열 쓰기	신규	Ver.1.02	Ver.1.03	2-547
	SubDelimiter	문자열 읽기 구분 문자 삭제	신규	Ver.1.02	Ver.1.03	2-558
시간/시각 명령	TruncTime	시간 버림	신규	Ver.1.01	Ver.1.02	2-628
	TruncDt	날짜 시각 버림	신규	Ver.1.01	Ver.1.02	2-632
	TruncTod	시각 버림	신규	Ver.1.01	Ver.1.02	2-636
아날로그 제어 명령	TimeProportionalOut	시분할 비례 출력	신규	Ver.1.02	Ver.1.03	2-670
	LimitAlarm_**	상하한 경보 그룹	신규	Ver.1.02	Ver.1.03	2-686
	LimitAlarmDv_**	상하한 편차 경보 그룹	신규	Ver.1.02	Ver.1.03	2-690
	LimitAlarmDvStbySeq_**	대기 시퀀스 부가 상하한 편차 경보 그룹	신규	Ver.1.02	–	2-694
	ScaleTrans	스케일 변환	신규	Ver.1.05	Ver.1.06	2-709
	AC_StepProgram	스텝 프로그램	신규	Ver.1.06	Ver.1.07	2-712

종류	명령	명칭	신규/사양 변경	버전		페이지
				CPU 유니트	Sysmac Studio	
시스템 제어 명령	ResetMCError	모션 제어 이상 해제	사양 변경	Ver.1.02	Ver.1.03	2-768
	GetECError	EtherCAT 이상 Status 취득	사양 변경	Ver.1.02	Ver.1.03	2-773
	RestartNXUnit	NX 유니트 재시작	신규	Ver.1.05	Ver.1.06	2-788
			사양 변경	Ver.1.07	Ver.1.08	
	NX_Change WriteMode	NX 유니트 쓰기 모드 변경	신규	Ver.1.05	Ver.1.06	2-793
	NX_SaveParam	NX 유니트 파라메터 저장	신규	Ver.1.05	Ver.1.06	2-797
EtherCAT 통신 명령	EC_Change EnableSetting	EtherCAT 슬레이브의 유효/무효 전환	신규	Ver.1.04	Ver.1.05	2-835
	NX_WriteObj	NX 개체 쓰기	신규	Ver.1.05	Ver.1.06	2-854
	NX_ReadObj	NX 개체 읽기	신규	Ver.1.05	Ver.1.06	2-869
EtherNet/IP 통신 명령	CIPOpenWith DataSize	CIPClass3 연결 확립(크기 변경 있음)	신규	Ver.1.06	Ver.1.07	2-887
	SktUDPCreate	UDP 소켓 생성	사양 변경	Ver.1.03	–	2-929
	SktTCPAccept	TCP 소켓 허용	사양 변경	Ver.1.03	–	2-942
	SktTCP Connect	TCP 소켓 연결	사양 변경	Ver.1.03	–	2-945
	ChangeIPAdr	IP 주소 변경	신규	Ver.1.02	Ver.1.03	2-968
	ChangeFTP Account	FTP 계정 변경	신규	Ver.1.02	Ver.1.03	2-977
	ChangeNTP ServerAdr	NTP 서버 수신처 변경	신규	Ver.1.02	Ver.1.03	2-980
시리얼 통신 명령	SerialRcvNo Clear(*)	시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함	신규	Ver.1.03	Ver.1.04	2-1011
타임 스탬프 명령	NX_DOut TimeStamp	타임 스탬프 디지털 출력 쓰기	신규	Ver.1.06	Ver.1.07	2-1118
	NX_AryDOut TimeStamp	타임 스탬프 디지털 출력 배열 쓰기	신규	Ver.1.06	Ver.1.07	2-1124
기타 명령	ActEventTask	이벤트 태스크 기동	신규	Ver.1.03	Ver.1.04	2-1148

* SerialRcvNoClear 명령은 CPU 유니트 Ver.1.03 이상, Sysmac Studio Ver.1.04 이상, 시리얼 커뮤니케이션 유니트 Ver.2.1 이상의 조합에서 사용할 수 있습니다.

에러 메시지 '명령이 의도하지 않은 동작을 일으킬 가능성이 있습니다.'가 표시되었을 때의 대처 방법

Sysmac Studio에서 '명령이 의도하지 않은 동작을 일으킬 가능성이 있습니다. 자세한 내용은 명령어 설명서(기본편)에서 확인하십시오. '라는 에러 메시지가 표시되는 경우가 있습니다. 이 문제는 사용자 프로그램의 제약으로 인한 것이므로 사용자 프로그램을 수정하여 대처할 수 있습니다.
여기에서는 이 에러 메시지가 표시되는 조건이나 대처 방법에 대해 설명합니다.

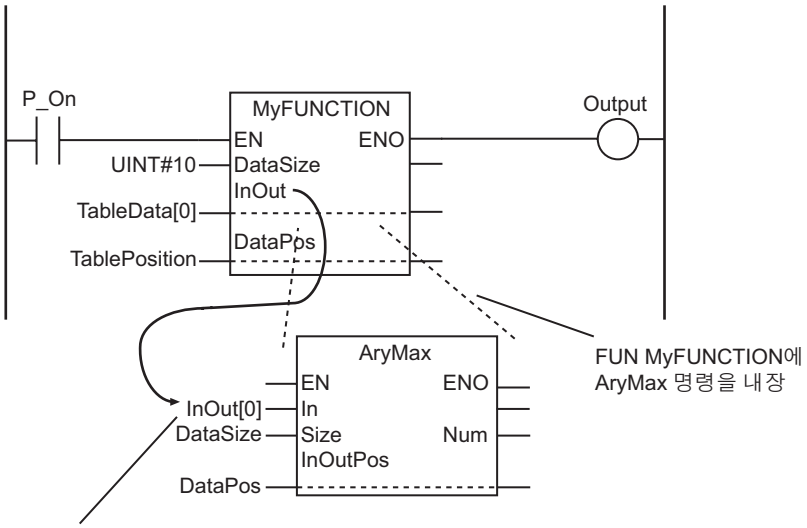
버전 관련 정보

이 에러 메시지는 Sysmac Studio Ver.1.02에서만 표시됩니다.

에러 메시지가 표시되는 조건

에러 메시지는 FB 또는 FUN의 입출력 변수를 해당 FB나 FUN 내에서 특정 명령의 특정 변수에 전달한 경우에 표시됩니다.
에러 메시지가 표시되는 명령과 변수 목록은 뒤에 나와 있습니다.

에러 메시지가 표시되는 사용자 프로그램의 예



에러 메시지가 표시되는 명령과 변수

이 이상은 특정 명령의 특정 변수에서만 발생합니다. 해당 목록이 아래에 나와 있습니다.

종류	명령 이름	명칭	변수명	페이지
비교 명령	TableCmp	테이블 비교	Table,AryOut	2-112
	AryCmpEQ	배열 일괄 비교 EQ	In1,In2,AryOut	2-115
	AryCmpNE	배열 일괄 비교 NE	In1,In2,AryOut	2-115

종류	명령 이름	명칭	변수명	페이지
비교 명령	AryCmpLT	배열 일괄 비교 LT	In1,In2,AryOut	2-117
	AryCmpLE	배열 일괄 비교 LE	In1,In2,AryOut	2-117
	AryCmpGT	배열 일괄 비교 GT	In1,In2,AryOut	2-117
	AryCmpGE	배열 일괄 비교 GE	In1,In2,AryOut	2-117
	AryCmpEQV	배열 요소 비교 EQ	In1,AryOut	2-119
	AryCmpNEV	배열 요소 비교 NE	In1,AryOut	2-119
	AryCmpLTV	배열 요소 비교 LT	In1,AryOut	2-121
	AryCmpLEV	배열 요소 비교 LE	In1,AryOut	2-121
	AryCmpGTV	배열 요소 비교 GT	In1,AryOut	2-121
	AryCmpGEV	배열 요소 비교 GE	In1,AryOut	2-121
산술 명령	AryAdd	배열 일괄 가산	In1,In2,AryOut	2-209
	AryAddV	배열 요소 가산	In1,AryOut	2-211
	ArySub	배열 일괄 감산	In1,In2,AryOut	2-213
	ArySubV	배열 요소 감산	In1,AryOut	2-215
	AryMean	배열 요소의 평균값 산출	In	2-217
	ArySD	배열 요소의 표준 편차	In	2-219
BCD 변환 명령	AryToBCD	배열 일괄 BCD 변환	In,AryOut	2-245
	AryToBin	배열 일괄 부호가 없는 정수 변환	In,AryOut	2-247
비트 열 연산 명령	AryAnd	배열 논리곱	In1,In2,AryOut	2-312
	AryOr	배열 논리합	In1,In2,AryOut	2-312
	AryXor	배열 배타적 논리합	In1,In2,AryOut	2-312
	AryXorN	배열 배타적 논리합 부정	In1,In2,AryOut	2-312
선택 명령	AryMax	배열 변수의 최대값 검색	In	2-329
	AryMin	배열 변수의 최소값 검색	In	2-329
	ArySearch	배열 검색	In	2-332
데이터 전송 명령	TransBits	다중 비트 전송	InOut	2-343
	AryExchange	배열 데이터 교환	InOut1,InOut2	2-351
	AryMove	배열의 전송	In,AryOut	2-353
	Clear	초기화	InOut	2-355
이동 명령	AryShiftReg	이동 레지스터	InOut	2-370
	AryShiftRegLR	좌우 이동 레지스터	InOut	2-372
	ArySHL	배열의 N요소 좌측 이동	InOut	2-375
	ArySHR	배열의 N요소 우측 이동	InOut	2-375
	NSHLC	N비트 데이터 N비트 좌측 이동 CY 있음	InOut	2-380
	NSHRC	N비트 데이터 N비트 우측 이동 CY 있음	InOut	2-380
데이터 변환 명령	Decoder	비트 디코더	InOut	2-389
	Encoder	비트 엔코더	In	2-392
	ColmToLine_**	비트 열→ 행 변환 그룹	In	2-395
	LineToColm	비트 행→ 열 변환	InOut	2-397
	PWLApprox	꺾은선 근사 변환	Line	2-413
	MovingAverage	이동 평균	Buf	2-416
	StringToAry	문자열→ 변환	AryOut	2-445
	AryToString	배열→변환	In	2-447
	DispartDigit	4비트 분리	AryOut	2-449
	UniteDigit_**	4비트 결합 그룹	In	2-451
	Dispart8Bit	바이트 단위 데이터 분리	AryOut	2-453
	Unite8Bit_**	바이트 단위 데이터 결합 그룹	In	2-455
	ToAryByte	바이트 배열로 변환	In,AryOut	2-457
	AryByteTo	바이트 배열에서 변환	In,OutVal	2-462
	SizeOfAry	배열 요소 수 취득	In	2-467
스택/테이블 명령	StackPush	스택 데이터 저장	InOut	2-470

종류	명령 이름	명칭	변수명	페이지
스택/테이블 명령	StackFIFO	선입선출	InOut,OutVal	2-480
	StackLIFO	후입선출	InOut,OutVal	2-480
	StackIns	스택 데이터 삽입	InOut	2-483
	StackDel	스택 데이터 삭제	InOut	2-485
	RecSearch	레코드 검색	In,Member, InOutPos	2-487
	RecRange Search	범위 지정 레코드 검색	In,Member, InOutPos	2-492
	RecSort	레코드 정렬	InOut, Member	2-497
	RecNum	레코드 수 취득	In,Member	2-502
	RecMax	레코드 최대값 검색	In,Member, InOutPos	2-504
	RecMin	레코드 최소값 검색	In,Member, InOutPos	2-504
FCS 명령	AryLRC_**	LRC값 산출(배열) 그룹	In	2-518
	AryCRCCCITT	CRC-CCITT값 산출(배열)	In	2-520
	AryCRC16	CRC-16값 산출(배열)	In	2-522
시스템 제어 명령	SetAlarm	사용자 이상 발생	Info1, Info2	2-747
	SetInfo	사용자 정보 발생	Info1, Info2	2-780
EtherCAT 통신 명령	EC_CoESDOWrite	CoE SDO 쓰기	WriteDat	2-804
	EC_CoESDORead	CoE SDO 읽기	ReadDat	2-807
EtherNet/IP 통신 명령	CIPRead	변수 읽기(Class3 Explicit 메시지)	DstDat	2-890
	CIPWrite	변수 쓰기(Class3 Explicit 메시지)	SrcDat	2-895
	CIPSend	임의 Explicit 메시지 송신(Class3)	ServiceDat, RespServiceDat	2-901
	CIPUCMMRead	변수 읽기(UCMM Explicit 메시지)	DstDat	2-907
	CIPUCMMWrite	변수 쓰기(UCMM Explicit 메시지)	SrcDat	2-912
	CIPUCMMSend	임의 Explicit 메시지 송신(UCMM)	ServiceDat, RespServiceDat	2-919
	SktUDPRcv	UDP 소켓 수신	RcvDat	2-936
	SktUDPSend	UDP 소켓 송신	SendDat	2-939
	SktTCPRcv	TCP 소켓 수신	RcvDat	2-953
	SktTCPSend	TCP 소켓 송신	SendDat	2-956
시리얼 통신 명령	ExecPMCR	프로토콜 매크로	SrcDat, DstDat	2-986
	SerialSend	시리얼 커뮤니케이션 유니트 시리얼 포트 출력	SrcDat	2-1000
	SerialRcv	시리얼 커뮤니케이션 유니트 시리얼 포트 입력	DstDat	2-1011
	SendCmd	명령 송신	CmdDat, RespDat	2-1026

종류	명령 이름	명칭	변수명	페이지
SD 메모리 카드 명령	FileWriteVar	변수 파일 쓰기	WriteVar	2-1040
	FileReadVar	변수 파일 읽기	ReadVar	2-1046
	FileRead	파일 읽기	ReadBuf	2-1061
	FileWrite	파일 쓰기	WriteBuf	2-1069
모션 제어 명령	MC_SetCamTable Property	캠 테이블 속성 갱신	CamTable	(*)
	MC_SaveCamTable	캠 테이블 저장	CamTable	(*)
	MC_Write	MC 설정 쓰기	Target, SettingValue	(*)
	MC_CamIn	캠 동작 시작	CamTable	(*)
	MC_ChangeAxes InGroup	축 그룹 구성축 쓰기	Axes	(*)
기타 명령	ChkRange	범위형 변수 확인	Val	2-1136

* 자세한 내용은 "NJ 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

대처 방법

에러 메시지가 표시되지 않도록 하려면 사용자 프로그램을 수정해야 합니다. 이를 위한 방법은 다음 2가지가 있습니다.

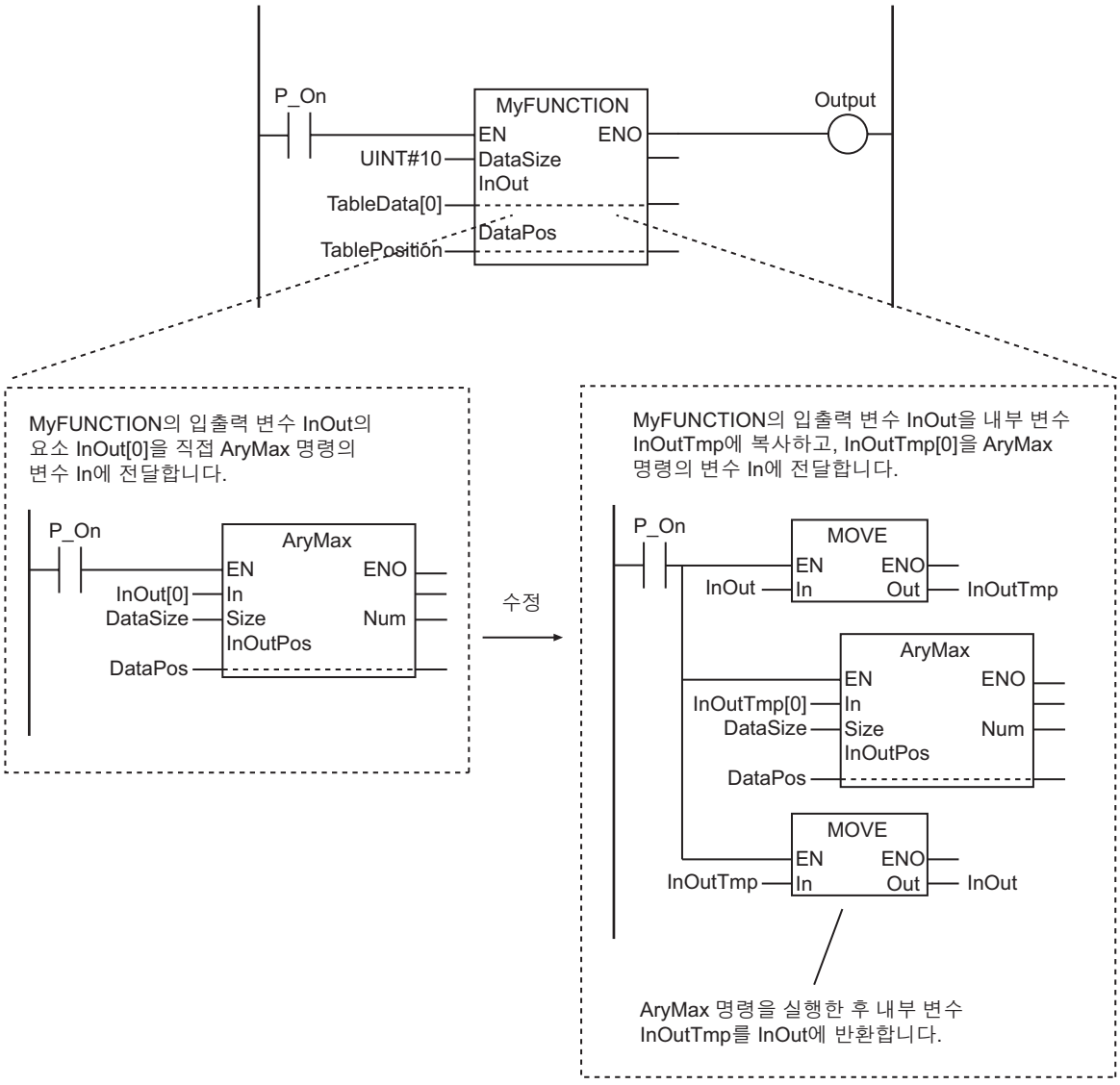
- 입출력 변수를 FB, FUN의 내부 변수에 복사한 후 명령에 전달
- 명령을 FB, FUN의 밖으로 출력

각각의 방법을 설명합니다.

● 입출력 변수를 FB, FUN의 내부 변수에 복사한 후 명령에 전달하는 방법

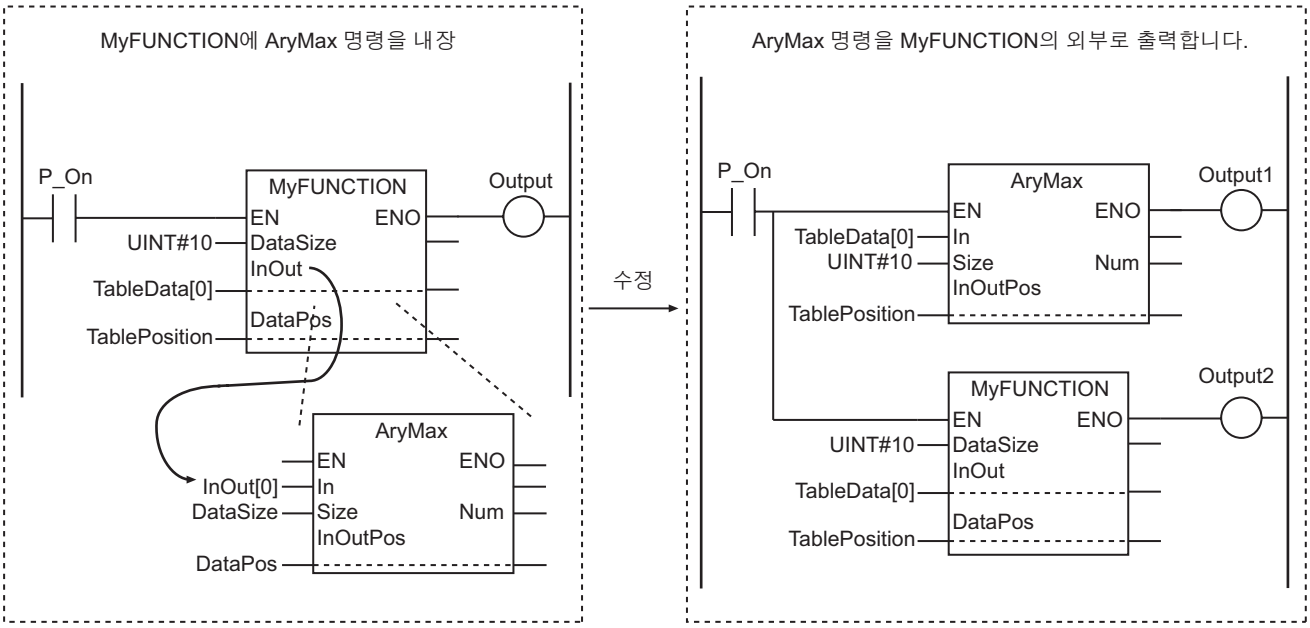
FB, FUN의 입출력 변수를 직접 명령으로 전달하지 않고 내부 변수에 복사한 후 전달합니다.
명령 실행에 의해 해당 내부 변수의 값이 바뀌는 경우에는 명령 실행 후 내부 변수를 원래 입출력 변수에 다시 복사합니다.

단, Clear 명령에서는 이 방법을 사용할 수 없습니다. Clear 명령을 사용하는 경우에는 명령을 FB, FUN의 밖으로 출력하는 방법을 실행하십시오.



● 명령을 FB, FUN의 밖으로 출력하는 방법

명령을 FB, FUN 내부에 넣지 않고 밖으로 출력합니다.



에러 메시지가 표시되어도 문제가 되지 않는 경우

에러 메시지가 표시되어도 문제 없이 명령을 사용할 수 있는 경우가 있습니다. 명령을 사용할 수 있는지 여부는 FB, FUN의 입출력 변수에 전달하는 파라메터에 의해 결정됩니다.
그 조건이 아래에 나와 있습니다.

명령 사용 가능 여부	FB, FUN의 입출력 변수에 전달하는 파라메터
사용 가능	기본 데이터 형식, 열거 형식, 배열 전체, 구조체 전체, 공용체 전체
사용 불가능	배열의 1개 요소, 구조체의 1개 멤버, 공용체의 1개 멤버

● 명령을 사용할 수 있는 예

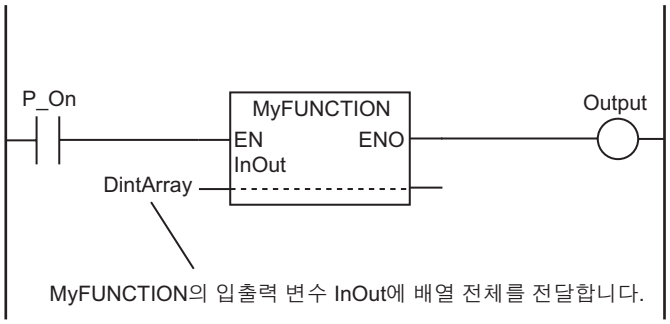
다음 예에서는 FB, FUN의 입출력 변수에 배열 전체를 전달하므로 내장된 명령을 사용할 수 있습니다.

MyFUNCTION의 호출원의 변수 테이블

명칭	데이터 형식
DintArray	ARRAY[0..9] OF DINT

MyFUNCTION의 변수 테이블

명칭	데이터 형식
InOut	ARRAY[0..9] OF DINT



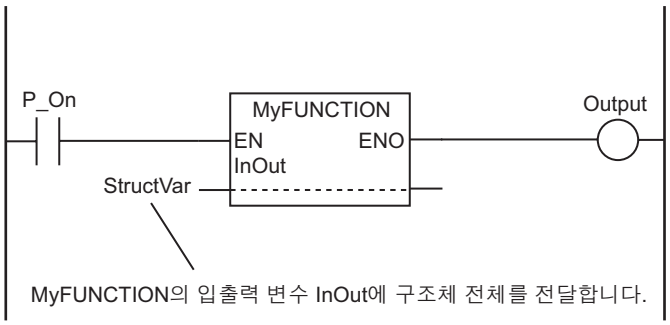
다음 예에서는 FB, FUN의 입출력 변수에 구조체 전체를 전달하므로 내장된 명령을 사용할 수 있습니다.

MyFUNCTION의 호출원의 변수 테이블

명칭	데이터 형식
StructVar	STRUCT

MyFUNCTION의 변수 테이블

명칭	데이터 형식
InOut	STRUCT



● 명령을 사용할 수 없는 예

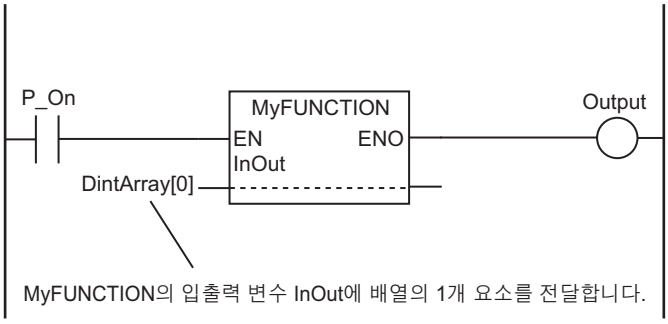
다음 예에서는 FB, FUN의 입출력 변수에 배열의 1개 요소를 전달하므로 내장된 명령을 사용할 수 없습니다.

MyFUNCTION의 호출원의 변수 테이블

명칭	데이터 형식
DintArray	ARRAY[0..9] OF DINT

MyFUNCTION의 변수 테이블

명칭	데이터 형식
InOut	DINT



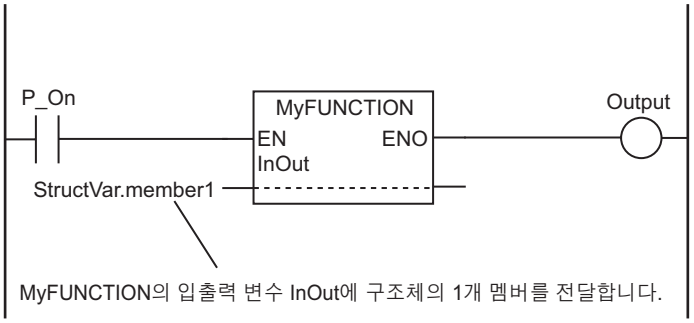
다음 예에서는 FB, FUN의 입출력 변수에 구조체의 1개 멤버를 전달하므로 내장된 명령을 사용할 수 없습니다.

MyFUNCTION의 호출원의 변수 테이블

명칭	데이터 형식
StructVar	STRUCT

MyFUNCTION의 변수 테이블

명칭	데이터 형식
InOut	DINT



이상 검출 옵션 설정

앞에서 설명한 대로 에러 메시지가 표시되어도 명령을 사용할 수 있는 경우가 있습니다. 이에 따라 Sysmac Studio에서는 이 이상을 감지할지 여부를 선택할 수 있습니다.
이러한 이상을 감지하지 않으려면 Sysmac Studio의 옵션 설정 화면에서 옵션 항목 '프로그램 확인'의 '입출력 변수를 특정 명령의 특정 인수에 전달하는 경우 에러'의 선택을 취소하십시오. Sysmac Studio의 자세한 조작 방법은 "Sysmac Studio Version 1 조작 설명서(SBCA-362C 이상)"를 참조하십시오.
단, 선택을 취소하기 전에 사용자 프로그램 내의 모든 명령이 사용 가능한 경우에 해당하는지 반드시 확인하십시오. 또한 선택을 취소한 경우에도 경고의 의미로 이 메시지가 표시됩니다.

⚠ 주 의

이 옵션의 선택을 취소한 경우, 명령이 의도하지 않은 동작을 수행하여 장치에 도 영향을 줄 수 있습니다.
'에러 메시지가 표시되어도 문제가 되지 않는 경우(A-174페이지)'에 기재된 사용할 수 있는 조건과 일치하는지 반드시 확인하십시오.



버전 관련 정보

에러 메시지 표시 및 상기의 옵션 설정은 Sysmac Studio Ver.1.02만 가능합니다.

I

색인

색인

기호

&(논리곱)..... 2-306
*(승산) 2-174
**(거듭제곱 연산)..... 2-203
_BCD_TO_*(BCD→부호가 없는 정수 변환 그룹)... 2-230
_TO_*(비트 열→비트 열 변환 그룹) 2-259
_TO_*(비트 열→실수 변환 그룹)..... 2-261
_TO_*(비트 열→정수 변환 그룹)..... 2-257
_TO_*(실수→비트 열 변환 그룹)..... 2-266
_TO_*(실수→실수 변환 그룹)..... 2-268
_TO_*(실수→정수 변환 그룹)..... 2-263
_TO_*(정수→비트 열 변환 그룹)..... 2-253
_TO_*(정수→실수 변환 그룹)..... 2-255
_TO_*(정수→정수 변환 그룹)..... 2-250
_TO_BCD_*(부호가 없는 정수→BCD 변환 그룹)... 2-233
**_TO_STRING(비트 열→문자열 변환 그룹)..... 2-272
**_TO_STRING(실수→문자열 변환 그룹)..... 2-274
**_TO_STRING(정수→문자열 변환 그룹)..... 2-270
+(가산)..... 2-162
+OU(가산 오버 플로우/언더 플로우 확인 있음)..... 2-165
-OU(감산 오버 플로우/언더 플로우 확인 있음)..... 2-171
-(감산)..... 2-168
/(제산)..... 2-180
<(비교 LT)..... 2-98
<=(비교 LE) 2-98
<>(비교 NE)..... 2-96
=(비교 EQ) 2-94
>(비교 GT)..... 2-98
>=(비교 GE) 2-98

숫자

100ms 타이머..... 2-138
16진수 문자열→수치 변환 그룹 2-430
1비트 리셋 2-59
1비트 세트..... 2-59
1비트 출력 2-61
4비트 결합 그룹..... 2-451
4비트 분리 2-449

A

ABS(절대값)..... 2-185
AccumulationTimer(적산 타이머) 2-135
ACOS(COS-1 연산)..... 2-192
AC_StepProgram(스텝 프로그램) 2-712
ActEventTask(이벤트 태스크 기동)..... 2-1148
ADD(가산)..... 2-162
AddDelimiter(구분 문자 부가 문자열 쓰기)..... 2-547
AddOU(가산 오버 플로우/언더 플로우 확인 있음)..... 2-165
ADD_TIME(시간 가산)..... 2-570
ADD_TOD_TIME(날짜 시각과 시간의 가산)..... 2-574
ADD_TOD_TIME(시각과 시간의 가산)..... 2-572
AND(논리곱) 2-18, 2-306
ANDN(부정 논리곱)..... 2-18
AryAdd(배열 일괄 가산) 2-209
AryAddV(배열 요소 가산)..... 2-211
AryAnd(배열 논리곱)..... 2-312
AryByteTo(바이트 배열에서 변환)..... 2-462
AryCmpEQ(배열 일괄 비교 EQ)..... 2-115
AryCmpEQV(배열 요소 비교 EQ) 2-119
AryCmpGE(배열 일괄 비교 GE)..... 2-117
AryCmpGEV(배열 요소 비교 GE) 2-121
AryCmpGT(배열 일괄 비교 GT)..... 2-117
AryCmpGTV(배열 요소 비교 GT) 2-121
AryCmpLE(배열 일괄 비교 LE) 2-117
AryCmpLEV(배열 요소 비교 LE)..... 2-121
AryCmpLT(배열 일괄 비교 LT) 2-117
AryCmpLTV(배열 요소 비교 LT)..... 2-121
AryCmpNE(배열 일괄 비교 NE) 2-115
AryCmpNEV(배열 요소 비교 NE)..... 2-119
AryCRC16(CRC-16값 산출 배열)..... 2-522
AryCRCCCITT(CRC-CCITT값 산출 배열) 2-520
AryExchange (배열 데이터 교환) 2-351
AryLRC_**(LRC값 산출 배열 그룹)..... 2-518
AryMax (배열 변수의 최대값 검색)..... 2-329
AryMean(배열 요소의 평균값 산출) 2-217
AryMin (배열 변수의 최소값 검색)..... 2-329
AryMove (배열의 전송)..... 2-353
AryOr(배열 논리합) 2-312
arySD(배열 요소의 표준 편차)..... 2-219
arySearch (배열 검색)..... 2-332
aryShiftReg(이동 레지스터)..... 2-370
aryShiftRegLR(좌우 이동 레지스터) 2-372
arySHL(배열의 N요소 좌측 이동) 2-375
arySHR(배열의 N요소 우측 이동)..... 2-375
arySub(배열 일괄 감산) 2-213
arySubV(배열 요소 감산)..... 2-215
aryToBCD(배열 일괄 BCD 변환)..... 2-245
aryToBin(배열 일괄 부호가 없는 정수 변환)..... 2-247
aryToString(배열→문자열 변환)..... 2-447
aryXor(배열 배타적 논리합)..... 2-312
aryXorN(배열 배타적 논리합 부정) 2-312
ASIN(SIN-1 연산)..... 2-192
ATAN(TAN-1 연산) 2-192

B

Band (데드 밴드 제어)..... 2-323
BCD→부호가 없는 정수 변환 그룹 2-230
BCDsToBin(부호가 있는 BCD→부호가 있는 정수 변환) 2-239

BCD_TO_**(BCD군→부호가 없는 정수 변환 그룹)... 2-236
BCD군→부호가 없는 정수 변환 그룹... 2-236
BIN 코드→그레이 코드 변환 그룹... 2-442
BinToBCDs_**(부호가 있는 정수→BCD 변환 그룹)... 2-242
BinToGray_**(BIN 코드→그레이 코드 변환 그룹)... 2-442
BitCnt(비트 카운터)... 2-394
BREAK(루프 중단)... 2-90

C

CASE(다방향 분기)... 2-32
ChangeFTPAccount(FTP 계정 변경)... 2-977
ChangeIPAdr(IP 주소 변경)... 2-968
ChangeNTPServerAdr(FTP 서버 수신처 변경)... 2-980
CheckReal(실수 확인)... 2-225
ChkLeapYear(윤년 판별)... 2-615
ChkRange(범위 형식 변수 확인)... 2-1136
CIP Class3 연결 중단... 2-905
CIP Class3 연결 확립... 2-878
CIP Class3 연결 확립(크기 지정 있음)... 2-887
CIPClose(CIP Class3 연결 중단)... 2-905
CIPOpen(CIP Class3 연결 확립)... 2-878
CIPOpenWithDataSize (CIP Class3 연결 확립 (크기 지정 있음))... 2-887
CIPRead(변수 읽기 Class3 Explicit 메시지)... 2-890
CIPSend(임의 Explicit 메시지 송신 Class3)... 2-901
CIPUCMMRead(변수 읽기 UCMM Explicit 메시지)... 2-907
CIPUCMMSend(임의 Explicit 메시지 송신 UCMM)... 2-919
CIPUCMMWrite(변수 쓰기 UCMM Explicit 메시지)... 2-912
CIPWrite(변수 쓰기 Class3 Explicit 메시지)... 2-895
Clear (초기화)... 2-355
ClearString(문자열ÅE클리어)... 2-542
Cmp(비교)... 2-108
CoE SDO 쓰기... 2-804
CoE SDO 읽기... 2-807
ColmToLine_**(비트 열→비트 행 변환 그룹)... 2-395
CONCAT_ DATE_TOD(날짜와 시각의 결합)... 2-590
CONCAT(문자열ÅE연결)... 2-526
Copy**To*** (비트 패턴 복사 비트 열→실수 그룹)... 2-359
Copy**To*** (비트 패턴 복사 실수→비트 열 그룹)... 2-365
Copy**ToNum (비트 패턴 복사 비트 열→부호가 있는 정수 그룹)... 2-357
Copy**ToNum (비트 패턴 복사 실수→부호가 있는 정수 그룹)... 2-366
CopyNumTo**(비트 패턴 복사 부호가 있는 정수→비트 열 그룹)... 2-361
CopyNumTo**(비트 패턴 복사 부호가 있는 정수→실수 그룹)... 2-363
COS 연산... 2-189
COS(COS 연산)... 2-189
COS-1 연산... 2-192
CRC-16값 산출(문자열)... 2-516
CRC-16값 산출(배열)... 2-522
CRC-CCITT값 산출(문자열)... 2-514
CRC-CCITT값 산출(배열)... 2-520
CTD(감산 카운터)... 2-142
CTD_**(감산 카운터 그룹)... 2-144
CTU(가산 카운터)... 2-146
CTU_**(가산 카운터 그룹)... 2-148
CTUD(가역 카운터)... 2-151
CTUD_**(가역 카운터 그룹)... 2-155

D

DateStructToDt(시각 결합)... 2-626
DateToSec(날짜→초 변환)... 2-602
DateToString(날짜→문자열 변환)... 2-439
DaysToMonth(일수→월 변환)... 2-618
Dec(감소)... 2-205
DegToRad(각도→라디안 변환)... 2-187
DELETE(문자열 · 삭제)... 2-537
DirCreate(디렉토리 작성)... 2-1110
DirRemove(디렉토리 삭제)... 2-1113
Dispart8Bit(바이트 단위 데이터 분리)... 2-453
DispartDigit(4비트 분리)... 2-449
DispartReal(실수의 가수 · 지수 분리)... 2-422
DIV(제산)... 2-180
DIVTIME(시간 제산)... 2-588
Down (하강 미분)... 2-44
DT_TO_DATE(날짜 시각으로부터 날짜 잘라내기)... 2-594
DtToDateStruct(시각 분해)... 2-624
DtToSec(날짜 시각→초 변환)... 2-600
DtToString(일시→문자열 변환)... 2-437
DT_TO_TOD(날짜 시각으로부터 시각 잘라내기)... 2-592

E

EC_ChangeEnableSetting(EtherCAT 슬레이브의 유효/무효 전환)... 2-835
EC_CoESDORead(CoE SDO 읽기)... 2-807
EC_CoESDOWrite(CoE SDO 쓰기)... 2-804
EC_ConnectSlave(EtherCAT 슬레이브 등록)... 2-833
EC_CopyMon(EtherCAT 패킷 전송)... 2-823
EC_DisconnectSlave(EtherCAT 슬레이브 이탈)... 2-825
EC_SaveMon(EtherCAT 패킷 저장)... 2-821
EC_StartMon(EtherCAT 패킷 모니터링 기능 시작)... 2-813
EC_StopMon(EtherCAT 패킷 모니터링 기능 정지)... 2-819
Encoder(비트 엔코더)... 2-392
End(종료)... 2-64
EnumToNum(열거 형식→정수 변환)... 2-299
EQ(비교 EQ)... 2-94
EQascii(문자열 비교 EQ)... 2-101
EtherCAT 슬레이브 등록... 2-833
EtherCAT 슬레이브 이탈... 2-825
EtherCAT 슬레이브의 유효/무효 전환... 2-835
EtherCAT 이상 Status 취득... 2-777
EtherCAT 이상 해제... 2-775
EtherCAT 패킷 모니터링 기능 시작... 2-813
EtherCAT 패킷 모니터링 기능 정지... 2-819
EtherCAT 패킷 저장... 2-821

EtherCAT 패킷 전송 2-823
EtherNet/IP 이상 Status 취득 2-766
Exchange (데이터 변환)..... 2-349
ExecPMCR(프로토콜 매크로)..... 2-986
EXP(자연 지수 연산)..... 2-201
EXPT(거듭제곱 연산)..... 2-203

F

FileClose(파일 닫기)..... 2-1055
FileCopy(파일 복사) 2-1092
FileGets(문자열 읽기)..... 2-1077
FileOpen(파일 열기)..... 2-1051
FilePuts(문자열 쓰기)..... 2-1084
FileRead(파일 읽기)..... 2-1061
FileReadVar(변수 파일 읽기)..... 2-1046
FileRemove(파일 삭제) 2-1100
FileRename(파일명 변경)..... 2-1105
FileSeek(파일 검색)..... 2-1058
FileWrite(파일 쓰기) 2-1069
FileWriteVar(변수 파일 쓰기)..... 2-1040
FIND(문자열 · 검색)..... 2-532
FixNumToString(고정 소수점 수→문자열 변환) 2-432
FOR(반복 시작)..... 2-83
Fraction(실수 소수부 추출)..... 2-223
FTP 계정 변경(FTP 계정 변경)..... 2-977
F_TRIG (상승 미분)..... 2-44

G

GE(비교 GE) 2-98
GEascii(문자열 비교 GE) 2-105
Get**Clk(클록 펄스 취득 그룹) 2-1154
Get**Cnt(가산 프리 러닝 카운터 취득 그룹)..... 2-1155
GetAlarm(사용자 이상 Status 취득)..... 2-754
GetByteLen(문자열 · 바이트 수 취득)..... 2-541
GetCJBError(I/O 버스 이상 Status 취득)..... 2-764
GetDayOfWeek(요일 취득) 2-620
GetDaysOfMonth(달의 일수 취득)..... 2-616
GetECError(EtherCAT 이상 Status 취득) 2-777
GetEIPError(EtherNet/IP 이상 Status 취득) 2-766
GetMCError(모션 제어 이상 Status 취득)..... 2-773
GetMyTaskStatus(로컬 태스크 Status 읽기)..... 2-1138
GetNTPStatus(NTP Status 읽어오기) 2-786
GetPLCError(PLC 이상 Status 취득)..... 2-760
GetTime(시각 취득)..... 2-598
GetTraceStatus(데이터 트레이스 Status 읽어오기).... 2-744
GetWeekOfYear(주 취득)..... 2-622
Gray(그레이 코드 변환)..... 2-399
GrayToBin_**(그레이 코드→BIN 코드 변환 그룹)..... 2-442
GT(비교 GT) 2-98
GTascii(문자열 비교 GT)..... 2-105

H

HexStringToNum_**(16진수 문자열→수치 변환 그룹) .. 2-430

I

I/O 버스 이상 Status 취득..... 2-764
I/O 버스 이상 해제..... 2-762
IF(선택) 2-28
Inc(증가)..... 2-205
INSERT(문자열 · 삽입)..... 2-539
IP 주소 변경(IP 주소 변경)..... 2-968

J

JMP(점프) 2-81

L

LD(로드)..... 2-16
LDN(로드 부정) 2-16
LE(비교 LE)..... 2-98
LEascii(문자열 비교 LE)..... 2-105
LEFT(문자열 · 좌측에서 가져오기)..... 2-528
LEN(문자열 · 길이 검출)..... 2-534
LIMIT(상하한 리미트 제한)..... 2-321
LimitAlarm_**(상하한 경보 그룹)..... 2-686
LimitAlarmDv_**(상하한 편차 경보 그룹) 2-690
LimitAlarmDvStbySeq_**(대기 시퀀스가 있는
상하한 편차 경보 그룹)..... 2-694
LineToColm(비트 행→비트 열 변환) 2-397
LN(자연 로그 연산)..... 2-198
Lock(태스크 간 배타적 잠금)..... 2-1143
LOG(상용 로그 연산)..... 2-198
LRC값 산출(문자열)..... 2-512
LRC값 산출(배열) 그룹 2-518
LRealToFormatString(실수LREAL→서식 있는
문자열 변환)..... 2-281
LT(비교 LT)..... 2-98
LTascii(문자열 비교 LT) 2-105

M

MAX (최대값 검색)..... 2-327
MC(마스터 컨트롤 시작) 2-66
MCR(마스터 컨트롤 종료)..... 2-66
MemCopy (메모리 복사) 2-345
MID(문자열 · 임의의 위치에서 가져오기) 2-530
MIN (최소값 검색) 2-327
MOD(나머지)..... 2-183
ModReal(실수 나머지) 2-221
MOVE (전송)..... 2-336
MoveBit (비트 전송) 2-339
MoveDigit (자릿수 전송)..... 2-341
MovingAverage(이동 평균)..... 2-416

MUL(승산)..... 2-174
MulOU(승산 오버 플로우/언더 플로우 확인 있음)..... 2-177
MULTIME(시간 승산)..... 2-586
MUX (멀티플렉서)..... 2-318

N

NanoSecToTime(나노초→시간 변환)..... 2-612
NE(비교 NE)..... 2-96
NEascii(문자열 비교 NE)..... 2-103
Neg(부호 반전)..... 2-387
NEXT(반복 종료)..... 2-83
NOT(비트 반전)..... 2-311
NTP Status 읽어오기..... 2-786
NSHLC(N비트 데이터 N비트 좌측 이동 CY 있음)..... 2-380
NSHRC(N비트 데이터 N비트 우측 이동 CY 있음)..... 2-380
NTP 서버 수신처 변경(NTP 서버 수신처 변경)..... 2-980
NumToDecString(고정 길이 10진수 문자열 변환)..... 2-427
NumToEnum(정수→열거 형식 변환)..... 2-301
NumToHexString(고정 길이 16진수 문자열 변환)..... 2-427
NX 개체 쓰기..... 2-854
NX 개체 읽기..... 2-869
NX 유니트 쓰기 모드 변경..... 2-793
NX 유니트 재시작..... 2-788
NX 유니트 파라메터 저장..... 2-797
NX_AryDOutTimeStamp(타임 스탬프 디지털 출력 배열 쓰기)..... 2-1124
NX_ChangeWriteMode(NX 유니트 쓰기 모드 변경).. 2-793
NX_DOutTimeStamp(타임 스탬프 디지털 출력 쓰기)..... 2-1118
NX_ReadObj(NX 개체 읽기)..... 2-869
NX_SaveParam(NX 유니트 파라메터 저장)..... 2-797
NX_WriteObj(NX 개체 쓰기)..... 2-854
N비트 데이터 N비트 우측 이동 CY 있음..... 2-380
N비트 데이터 N비트 좌측 이동 CY 있음..... 2-380
N비트 쓰기 그룹..... 2-1134
N비트 우측 이동..... 2-378
N비트 우회전..... 2-382
N비트 읽기 그룹..... 2-1132
N비트 좌측 이동..... 2-378
N비트 좌회전..... 2-382

O

OR(논리합)..... 2-21, 2-306
ORN(부정 논리합)..... 2-21
Out(출력)..... 2-24
OutABit(1비트 출력)..... 2-61
OutNot(부정 출력)..... 2-24

P

PIDAT(자동 조정 가능한 PID 연산)..... 2-642
PLC 이상 Status 취득..... 2-760
PLC 이상 해제..... 2-756
PWLApprox(꺾은선 근사 변환(꺾은선 확인 있음))..... 2-408
PWLApproxNoLineChk(꺾은선 근사 변환 (꺾은선 확인 없음))..... 2-408
PWLDatChk(꺾은선 데이터 확인)..... 2-413

R

RadToDeg(라디안→각도 변환)..... 2-187
Rand(난수 발생)..... 2-207
ReadNbit_**(N비트 읽기 그룹)..... 2-1132
RealToFormatString(실수REAL→서식 있는 문자열 변환)..... 2-276
RecMax(레코드 최대값 검색)..... 2-504
RecMin(레코드 최소값 검색)..... 2-504
RecNum(레코드 수 취득)..... 2-502
RecRangeSearch(범위 지정 레코드 검색)..... 2-492
RecSearch(레코드 검색)..... 2-487
RecSort(레코드 정렬)..... 2-497
REPEAT(반복)..... 2-38
REPLACE(문자열 · 치환)..... 2-535
Reset(리셋)..... 2-54
ResetABit(1비트 리셋)..... 2-59
ResetAlarm(사용자 이상 해제)..... 2-752
ResetBits(다중 비트 리셋)..... 2-57
ResetCJBError(I/O 버스 이상 해제)..... 2-762
ResetECError(EtherCAT 이상 해제)..... 2-775
ResetMCError(모션 제어 이상 해제)..... 2-768
ResetPLCError(PLC 이상 해제)..... 2-756
ResetUnit(유니트 재시작)..... 2-782
RestartNXUnit(NX 유니트 재시작)..... 2-788
RETURN(복귀)..... 2-65
RIGHT(문자열 · 우측에서 가져오기)..... 2-528
ROL(N비트 좌회전)..... 2-382
ROR(N비트 우회전)..... 2-382
Round(실수 반올림)..... 2-303
RoundUp(실수 올림)..... 2-303
RS(리셋 우선 유지)..... 2-50
R_TRIG (상승 미분)..... 2-44

S

ScaleTrans(스케일 변환)..... 2-709
SecToDate(초→날짜 변환)..... 2-606
SecToDt(초→날짜 시각 변환)..... 2-604
SecToTime(초→시각 변환)..... 2-613
SecToTod(초→시각 변환)..... 2-608
SEL (비트 선택)..... 2-316
SendCmd(명령 송신)..... 2-1026
SerialRcv(시리얼 커뮤니케이션 유니트 시리얼 포트 입력)..... 2-1011
SerialRcvNoClear(시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함)..... 2-1011
SerialSend(시리얼 커뮤니케이션 유니트 시리얼 포트 출력)..... 2-1000
Set(세트)..... 2-54

SetABit(1비트 세트)..... 2-59
SetAlarm(사용자 이상 발생)..... 2-747
SetBits(다중 비트 세트)..... 2-57
SetBlock (블록 설정)..... 2-347
SetInfo(사용자 정보 발생)..... 2-780
SetTime(시계 보정)..... 2-596
SHL(N비트 좌측 이동)..... 2-378
SHR(N비트 우측 이동)..... 2-378
SIN 연산..... 2-189
SIN(SIN 연산)..... 2-189
SIN-1 연산..... 2-192
SizeOfAry(배열 요소 수 취득)..... 2-467
SJISToUTF8(문자 코드 변환(SJIS→UTF-8))..... 2-406
SktClearBuf(TCP/UDP 소켓 수신 버퍼 클리어)..... 2-965
SktClose(TCP/UDP 소켓 클로즈)..... 2-962
SktGetTCPStatus(TCP 소켓의 Status 읽기)..... 2-959
SktTCPAccept(TCP 소켓 허용)..... 2-942
SktTCPConnect(TCP 소켓 연결)..... 2-945
SktTCPRcv(TCP 소켓 수신)..... 2-953
SktTCPSend(TCP 소켓 송신)..... 2-956
SktUDPCreate(UDP 소켓 크리에이트)..... 2-929
SktUDPRcv(UDP 소켓 수신)..... 2-936
SktUDPSend(UDP 소켓 송신)..... 2-939
SQRT(제곱근 연산)..... 2-196
SR(세트 우선 유지)..... 2-52
StackDel(스택 데이터 삭제)..... 2-485
StackFIFO(선입선출)..... 2-480
StackIns(스택 데이터 삽입)..... 2-483
StackLIFO(후입선출)..... 2-480
StackPush(스택 데이터 저장)..... 2-470
StringCRC16(CRC-16값 산출 문자열)..... 2-516
StringCRCCCITT(CRC-CCITT값 산출 문자열)..... 2-514
StringLRC(LRC값 산출 문자열)..... 2-512
StringSum(SUM값 산출)..... 2-510
STRING_TO_**(문자열→비트 열 변환 그룹)..... 2-288
STRING_TO_**(문자열→실수 변환 그룹)..... 2-290
STRING_TO_**(문자열→정수 변환 그룹)..... 2-286
StringToAry(문자열→배열 변환)..... 2-445
StringToFixNum(문자열→고정 소수점 수 변환)..... 2-434
SUB(감산)..... 2-168
SUB_DATE_DATE(날짜 감산)..... 2-581
SubDelimiter(문자열 읽기 구분 문자 삭제)..... 2-558
SUB_DT_DT(날짜 시각 감산)..... 2-582
SUB_DT_TIME(날짜 시각과 시간의 감산)..... 2-584
SubOU(감산 오버 플로우/언더 플로우 확인 있음)..... 2-171
SUB_TIME(시간 감산)..... 2-576
SUB_TOD_TIME(시각과 시간의 감산)..... 2-578
SUB_TOD_TOD(시각 감산)..... 2-580
SUM값 산출..... 2-510
Swap(바이트 스왑)..... 2-386

T

TableCmp(테이블 비교)..... 2-112
TAN(TAN 연산)..... 2-189
TAN-1 연산..... 2-192
Task_IsActive(태스크 실행 중 판정)..... 2-1141
TCP 소켓 송신..... 2-956
TCP 소켓 수신..... 2-953
TCP 소켓 연결..... 2-945
TCP 소켓 허용..... 2-942
TCP 소켓의 Status 읽기..... 2-959
TCP/UDP 소켓 수신 버퍼 클리어..... 2-965
TCP/UDP 소켓 클로즈..... 2-962
TestABit(비트 테스트)..... 2-47
TestABitN(비트 테스트 부정)..... 2-47
TimeProportionalOut(시분할 비례 출력)..... 2-670
Timer(100ms 타이머)..... 2-138
TimeToNanoSec(시간→나노초 변환)..... 2-610
TimeToSec(시간→초 변환)..... 2-611
TO_**(비트 열 변환 그룹)..... 2-295
TO_**(실수 변환 그룹)..... 2-297
TO_**(정수 변환 그룹)..... 2-293
ToAryByte(바이트 배열로 변환)..... 2-457
TodToSec(시각→초 변환)..... 2-603
TodToString(시각→문자열 변환)..... 2-440
TOF(오프 딜레이 타이머)..... 2-129
ToLCase(문자열 · 소문자 변환)..... 2-543
TON(온 딜레이 타이머)..... 2-124
ToUCase(문자열 · 대문자 변환)..... 2-543
TP(펄스 출력)..... 2-132
TraceSamp(데이터 트레이스 샘플링)..... 2-738
TraceTrig(데이터 트레이스 트리거 조건 성립)..... 2-741
TransBits (다중 비트 전송)..... 2-343
TrimL(문자열 · 좌측 트림)..... 2-545
TrimR(문자열 · 우측 트림)..... 2-545
TRUNC(실수 버림)..... 2-303
TruncDt(날짜 시각 버림)..... 2-632
TruncTime(시간 버림)..... 2-628
TruncTod(시각 버림)..... 2-636

U

UDP 소켓 송신..... 2-939
UDP 소켓 수신..... 2-936
UDP 소켓 크리에이트..... 2-929
Unite8Bit_**(바이트 단위 데이터 결합 그룹)..... 2-455
UniteDigit_**(4비트 결합 그룹)..... 2-451
UniteReal(가수 · 지수를 실수로 결합)..... 2-425
Unlock(태스크 간 배타적 잠금 해제)..... 2-1143
Up (상승 미분)..... 2-44
UTF8ToSJIS(문자 코드 변환(UTF-8→SJIS))..... 2-404

W

WHILE(반복)..... 2-36
WriteNbit_**(N비트 쓰기 그룹)..... 2-1134

X

XOR(배타적 논리합) 2-306
XORN(배타적 논리합 부정) 2-309

Z

Zone (데드 존 제어) 2-325
ZoneCmp(영역 비교) 2-110

가

가산 2-162
가산(오버 플로우/언더 플로우 확인 있음) 2-165
가산 카운터 2-146
가산 카운터 그룹 2-148
가산 프리 러닝 카운터 취득 그룹 2-1155
가수 · 지수를 실수로 결합 2-425
가역 카운터 2-151
가역 카운터 그룹 2-155
각도→라디안 변환 2-187
감산 2-168
감산(오버 플로우/언더 플로우 확인 있음) 2-171
감산 카운터 2-142
감산 카운터 그룹 2-144
감소 2-205
거듭제곱 연산 2-203
고정 길이 10진수 문자열 변환 2-427
고정 길이 16진수 문자열 변환 2-427
고정 소수점 수→문자열 변환 2-432
구분 문자 부가 문자열 쓰기 2-547
그레이 코드→BIN 코드 변환 그룹 2-442
그레이 코드 변환 2-399
꺾은선 근사 변환(꺾은선 확인 없음) 2-408
꺾은선 근사 변환(꺾은선 확인 있음) 2-408
꺾은선 데이터 확인(꺾은선 데이터 확인) 2-413

나

나노초→시간 변환 2-612
나머지 2-183
난수 발생 2-207
날짜 감산 2-581
날짜→문자열 변환 2-439
날짜 시각 감산 2-582
날짜 시각 버림(날짜 시각 버림) 2-632
날짜 시각→초 변환 2-600
날짜 시각과 시간의 가산 2-574
날짜 시각과 시간의 감산 2-584
날짜 시각으로부터 날짜 잘라내기 2-594
날짜 시각으로부터 시각 잘라내기 2-592
날짜→초 변환 2-602
날짜와 시각의 결합 2-590
논리곱 2-18, 2-306
논리합 2-21, 2-306

다

다방향 분기 2-32
다중 비트 리셋 2-57
다중 비트 세트 2-57
다중 비트 전송 2-343
달의 일수 취득 2-616
대기 시퀀스가 있는 상하한 편차 경보 그룹 2-694
데드 밴드 제어 2-323
데드 존 제어 2-325
데이터 교환 2-349
데이터 트레이스 Status 읽어오기 2-744
데이터 트레이스 샘플링 2-738
데이터 트레이스 트리거 조건 성립 2-741
디렉토리 삭제 2-1113
디렉토리 작성 2-1110
디코더(비트 디코더) 2-389

라

라디안→각도 변환 2-187
레코드 검색 2-487
레코드 수 취득 2-502
레코드 정렬 2-497
레코드 최대값 검색 2-504
레코드 최소값 검색 2-504
로드 2-16
로드 부정 2-16
로컬 태스크 Status 읽기 2-1138
루프 중단 2-90
리셋 2-54
리셋 우선 유지 2-50

마

마스터 컨트롤 시작 2-66
마스터 컨트롤 종료 2-66
멀티플렉서 2-318
메모리 복사 2-345
명령 송신 2-1026
모션 제어 이상 Status 취득 2-773
모션 제어 이상 해제 2-768
문자 코드 변환(SJIS→UTF-8) 2-406
문자열→고정 소수점 수 변환 2-434
문자열→배열 변환 2-445
문자열 비교 EQ 2-101
문자열 비교 GE 2-105
문자열 비교 GT 2-105
문자열 비교 LE 2-105
문자열 비교 LT 2-105
문자열 비교 NE 2-103
문자열→비트 열 변환 그룹 2-288
문자열→실수 변환 그룹 2-290
문자열 쓰기 2-1084
문자열 · 연결 2-526

문자열 읽기 2-1077
문자열 읽기 구분 문자 삭제 2-558
문자열→정수 변환 그룹 2-286
문자열 · 검색 2-532
문자열 · 길이 검출 2-534
문자열 · 대문자 변환 2-543
문자열 · 바이트 수 취득 2-541
문자열 · 삭제 2-537
문자열 · 삽입 2-539
문자열 · 소문자 변환 2-543
문자열 · 우측 트림 2-545
문자열 · 우측에서 가져오기 2-528
문자열 · 좌측 트림 2-545
문자열 · 좌측에서 가져오기 2-528
문자열 · 임의의 위치에서 가져오기 2-530
문자열 · 치환 2-535
문자열 · 클리어 2-542
바이트 단위 데이터 결합 그룹 2-455
바이트 단위 데이터 분리 2-453
바이트 배열로 변환 2-457
바이트 배열에서 변환 2-462
바이트 스왑 2-386
반복(REPEAT) 2-38
반복(WHILE) 2-36
반복 시작 2-83
반복 종료 2-83
배열 검색 2-332
배열 논리곱 2-312
배열 논리합 2-312
배열 데이터 교환 2-351
배열→문자열 변환 2-447
배열 배타적 논리합 2-312
배열 배타적 논리합 부정 2-312
배열 변수의 최대값 검색 2-329
배열 변수의 최소값 검색 2-329
배열 요소 가산 2-211
배열 요소 감산 2-215
배열 요소 비교 EQ 2-119
배열 요소 비교 GE 2-121
배열 요소 비교 GT 2-121
배열 요소 비교 LE 2-121
배열 요소 비교 LT 2-121
배열 요소 비교 NE 2-119
배열 요소 수 취득 2-467
배열 요소의 평균값 산출 2-217
배열 요소의 표준 편차 2-219
배열 일괄 BCD 변환 2-245
배열 일괄 가산 2-209
배열 일괄 감산 2-213
배열 일괄 부호가 없는 정수 변환 2-247
배열 일괄 비교 EQ 2-115
배열 일괄 비교 GE 2-117
배열 일괄 비교 GT 2-117
배열 일괄 비교 LE 2-117
배열 일괄 비교 LT 2-117
배열 일괄 비교 NE 2-115
배열의 N요소 우측 이동 2-375
배열의 N요소 좌측 이동 2-375
배열의 전송 2-353
배타적 논리합 2-306
배타적 논리합 부정 2-309
범위 지정 레코드 검색 2-492
범위 형식 변수 확인 2-1136
변수 쓰기(Class3 Explicit 메시지) 2-895
변수 쓰기(UCMM Explicit 메시지) 2-912
변수 읽기(Class3 Explicit 메시지) 2-890
변수 읽기(UCMM Explicit 메시지) 2-907
변수 파일 쓰기 2-1040
변수 파일 읽기 2-1046
복귀 2-65
부정 논리곱 2-18
부정 논리합 2-21
부정 출력 2-24
부호 반전 2-387
부호가 없는 정수→BCD 변환 그룹 2-233
부호가 있는 BCD→부호가 있는 정수 변환 2-239
부호가 있는 정수→BCD 변환 그룹 2-242
블록 설정 2-347
비교 2-108
비교 EQ 2-94
비교 GE 2-98
비교 GT 2-98
비교 LE 2-98
비교 LT 2-98
비교 NE 2-96
비트 디코더 2-389
비트 반전 2-311
비트 선택 2-316
비트 엔코더 2-392
비트 열→문자열 변환 그룹 2-272
비트 열 변환 그룹 2-295
비트 열→비트 열 변환 그룹 2-259
비트 열→비트 행 변환 그룹 2-395
비트 열→실수 변환 그룹 2-261
비트 열→정수 변환 그룹 2-257
비트 전송 2-339
비트 카운터 2-394
비트 테스트 2-47
비트 테스트 부정 2-47
비트 패턴 복사(부호가 있는 정수→비트 열) 그룹 2-361
비트 패턴 복사(부호가 있는 정수→실수) 그룹 2-363
비트 패턴 복사(비트 열→부호가 있는 정수) 그룹 2-357
비트 패턴 복사(실수→부호가 있는 정수) 그룹 2-366
비트 패턴 복사(비트 열→실수) 그룹 2-359
비트 패턴 복사(실수→비트 열) 그룹 2-365
비트 행→비트 열 변환 2-397

사

사용자 이상 Status 취득 2-754
사용자 이상 발생 2-747
사용자 정보 발생 2-780
사용자 이상 해제 2-752
상승 미분 2-44
상용 로그 연산 2-198
상하한 경보 그룹 2-686
상하한 리미트 제한 2-321
상하한 편차 경보 그룹 2-690
선입선출 2-480
선택 2-28
세트 2-54
세트 우선 유지 2-52
스케일 변환 2-709
스택 데이터 삭제 2-485
스택 데이터 삽입 2-483
스택 데이터 저장 2-470
스텝 프로그램 2-712
승산 2-174
승산(오버 플로우/언더 플로우 확인 있음) 2-177
시각 감산 2-580
시각 결합 2-626
시각→문자열 변환 2-440
시각 버림(시각 버림) 2-636
시각 분해 2-624
시각→초 변환 2-603
시각 취득 2-598
시각과 시간의 가산 2-572
시각과 시간의 감산 2-578
시간 가산 2-570
시간 감산 2-576
시간→나노초 변환 2-610
시간 버림(시간 버림) 2-628
시간 승산 2-586
시간 제산 2-588
시간→초 변환 2-611
시계 보정 2-596
시리얼 커뮤니케이션 유니트 시리얼 포트 입력 2-1011
시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함(시리얼 커뮤니케이션 유니트 시리얼 포트 입력 수신 버퍼 삭제 안 함) 2-1011
시리얼 커뮤니케이션 유니트 시리얼 포트 출력 2-1000
시분할 비례 출력 2-670
실수(LREAL)→서식 있는 문자열 변환 2-281
실수(REAL)→서식 있는 문자열 변환 2-276
실수 나머지 2-221
실수→문자열 변환 그룹 2-274
실수 반올림 2-303
실수 버림 2-303
실수 변환 그룹 2-297
실수→비트 열 변환 그룹 2-266
실수 소수부 추출 2-223
실수→실수 변환 그룹 2-268
실수 올림 2-303
실수→정수 변환 그룹 2-263
실수 확인 2-225
실수의 가수 · 지수 분리 2-422

아

영역 비교 2-110
오프 딜레이 타이머 2-129
온 딜레이 타이머 2-124
요일 취득 2-620
유니트 재시작 2-782
윤년 판별 2-615
이동 레지스터 2-370
이동 평균 2-416
일수→월 변환 2-618
일시→문자열 변환 2-437
임의 Explicit 메시지 송신(Class3) 2-901
임의 Explicit 메시지 송신(UCMM) 2-919

자

자동 조정 가능한 PID 연산 2-642
자릿수 전송 2-341
자연 로그 연산 2-198
자연 지수 연산 2-201
적산 타이머 2-135
전송 2-336
절대값 2-185
점프 2-81
정수→문자열 변환 그룹 2-270
정수 변환 그룹 2-293
정수→비트 열 변환 그룹 2-253
정수→실수 변환 그룹 2-255
정수→정수 변환 그룹 2-250
제곱근 연산 2-196
제산 2-180
종료 2-64
좌우 이동 레지스터 2-372
주 취득 2-622
증가 2-205

차

초→날짜 변환 2-606
초→날짜 시각 변환 2-604
초→시각 변환 2-608
초→시간 변환 2-613
초기화 2-355
최대값 검색 2-327
최소값 검색 2-327
출력 2-24

카

클록 펄스 취득 그룹 2-1154

타

타임 스탬프 디지털 출력 배열 쓰기 2-1124
타임 스탬프 디지털 출력 쓰기 2-1118
태스크 간 배타적 잠금 2-1143
태스크 간 배타적 잠금 해제 2-1143
태스크 실행 중 판정 2-1141
테이블 비교 2-112

파

파일 검색 2-1058
파일 닫기 2-1055
파일 복사 2-1092
파일 삭제 2-1100
파일 쓰기 2-1069
파일 열기 2-1051
파일 읽기 2-1061
파일명 변경 2-1105
펄스 출력 2-132
프로토콜 매크로 2-986

하

하강 미분 2-44
후입선출 2-480

- 이 사용 설명서에 기재되지 않은 조건이나 환경에서 사용하거나 원자력 제어, 철도, 항공, 차량, 연소 장치, 의료 기기, 오락 기계, 안전 기기, 그 밖에 인명이나 재산에 큰 영향을 미치는 등 안전성이 각별히 요구되는 용도로 사용할 것을 검토하는 경우는 정격이나 성능에 여유가 있는 사용 방법이나 장애 시 안전 대책 등을 마련하는 한편, 당사 영업 담당자에게 상담하여 사양서 등을 확인하시기 바랍니다.
- 본 제품 중 외환 및 해외 무역법의 규정에 따라 수출허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우, 동법에 규정된 수출 허가, 승인(또는 직무 거래 허가)이 필요합니다.

Industrial Web ▶ http://www.ia.omron.co.kr

한국 오므론 제어기기 주식회사

137-920 서울시 서초구 서초동 1303-22 교보타워 B동 21층 TEL : 02-3483-7789 FAX : 02-3483-7788

카탈로그 번호 SBCA-360J-K6 2014년 01월 현재

© OMRON Corporation 2011 All Rights Reserved.
예고없이 사양 등을 변경하는 경우가 있으므로 양해 부탁드립니다.