OMRON

머신 자동화 컨트롤러

NJ/NX 시리즈

CPU 유니트

사용 설명서
소프트웨어편

NX701-17□□
NX701-16□□
NX102-12□□
NX102-11□□
NX102-10□□
NX102-90□□
NX1P2-11□□□□
NX1P2-10□□□□
NX1P2-90□□□□
NJ501-□5□□
NJ501-□4□□
NJ501-□3□□
NJ301-12□□
NJ301-11□□
NJ101-10□□
NJ101-90□□

CPU 유니트



SYSMAC
always in control

SBCA-467A

부탁의 말

- 본 설명서 내용의 일부 또는 전체를 무단으로 복사, 복제, 전재하는 것을 금지합니다.
- 본 설명서의 내용은 개선을 위해 예고 없이 사양 등이 변경되는 경우가 있으므로 미리 양지해 주시길 바랍니다.
- 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 OMRON 주식회사의 FA 기기 제품의 상표 또는 등록 상표입니다.
- Microsoft, Windows, Windows Vista, Excel, Visual Basic은 미국 및 기타 국가에서 미국 Microsoft Corporation의 등록 상표 또는 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 회사명 및 제품명은 각 회사의 상표 또는 등록 상표입니다.

저작권

스크린샷은 마이크로소프트사의 허가를 받아서 사용하고 있습니다.

머리말

NJ/NX 시리즈 CPU 유니트를 구매해 주셔서 감사합니다.
이 설명서는 NJ/NX 시리즈 CPU 유니트를 사용하는 데 필요한 정보를 소개합니다. 이 설명서를 잘 읽고 기능 및 성능 등을 충분히 숙지하여 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 자격증 또는 이와 동등한 지식을 가진 자)이 있는
- FA 기기 도입 담당자
- FA 시스템 설계자
- FA 기기 설치, 연결 담당자
- FA 현장 관리자

또한, 프로그래밍 언어에 관해 국제 표준 규격 IEC 61131-3 또는 일본 표준 규격 JIS B 3503에 규정된 내용을 숙지하는 사람을 대상으로 기술하였습니다.

대상 제품

이 설명서는 다음 제품을 대상으로 합니다.
- NX 시리즈 CPU 유니트
  - NX701-17□□
  - NX701-16□□
  - NX102-12□□
  - NX102-11□□
  - NX102-10□□
  - NX102-90□□
  - NX1P2-11□□□□
  - NX1P2-11□□□□1
  - NX1P2-10□□□□
  - NX1P2-10□□□□1
  - NX1P2-90□□□□
  - NX1P2-90□□□□1

- NJ 시리즈 CPU 유니트
  - NJ501-□5□□
  - NJ501-□4□□
  - NJ501-□3□□
  - NJ301-12□□
  - NJ301-11□□
  - NJ101-10□□
  - NJ101-90□□

각 제품의 일부 사양 또는 제약 사항이 다른 설명서에 기재된 경우가 있습니다. 📖'설명서의 구성(2페이지)' 및 📖'관련 설명서(28페이지)'를 확인하십시오.

설명서의 구성

이 제품의 설명서는 다음 표와 같이 여러 권으로 나누어져 있습니다. 목적에 따라 읽고 이 제품을 활용하십시오.

이 제품의 조작에는 주로 자동화 소프트웨어 Sysmac Studio를 사용합니다. Sysmac Studio에 대해서는 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

사용 목적	설명서																		
	기본 정보																		
	NX 시리즈 CPU 유니트 사용 설명서 하드웨어편	NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편	NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편	NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	NJ/NX 시리즈 CPU 유니트 사용 설명서 소프트웨어편	NX 시리즈 NX1P2 CPU 유니트 CPU 유니트 사용 설명서 내장 I/O, 확장 보드 기능편	NJ/NX 시리즈 명령 참조 설명서 기본편	NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편	NJ/NX 시리즈 명령 참조 설명서 모션편	NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서	NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서	NJ/NX 시리즈 CPU 유니트 사용 설명서 OPC UA편	NX 시리즈 CPU 유니트 사용 설명서 FINS 기능편	NJ/NX 시리즈 데이터베이스 연결 CPU 유니트 사용 설명서	NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용 설명서	NJ 시리즈 NJ Robotics CPU 유니트 사용 설명서	NJ/NY 시리즈 NC 통합 컨트롤러 사용 설명서	NJ/NX 시리즈 트러블 슈팅 설명서
NX701의 개요 알아보기	●																		
NX102의 개요 알아보기		●																	
NX1P2의 개요 알아보기			●																
NJ 시리즈의 개요 알아보기				●															
장착, 설치, 하드웨어 설정 실시	●	●	●	●															
모션 제어를 실시하는 경우								●											
EtherCAT을 사용하는 경우										●									
EtherNet/IP를 사용하는 경우											●								
소프트웨어 설정 실시					●														
모션 제어를 실시하는 경우								●											
EtherCAT을 사용하는 경우										●									
EtherNet/IP를 사용하는 경우											●								
OPC UA를 사용하는 경우												●							
FINS를 사용하는 경우													●						
데이터베이스 연결 서비스를 사용하는 경우														●					
GEM 서비스를 사용하는 경우															●				
로봇 제어를 실시하는 경우																●			
수치 제어를 실시하는 경우																	●		
NX1P2의 기능을 사용하는 경우						●													

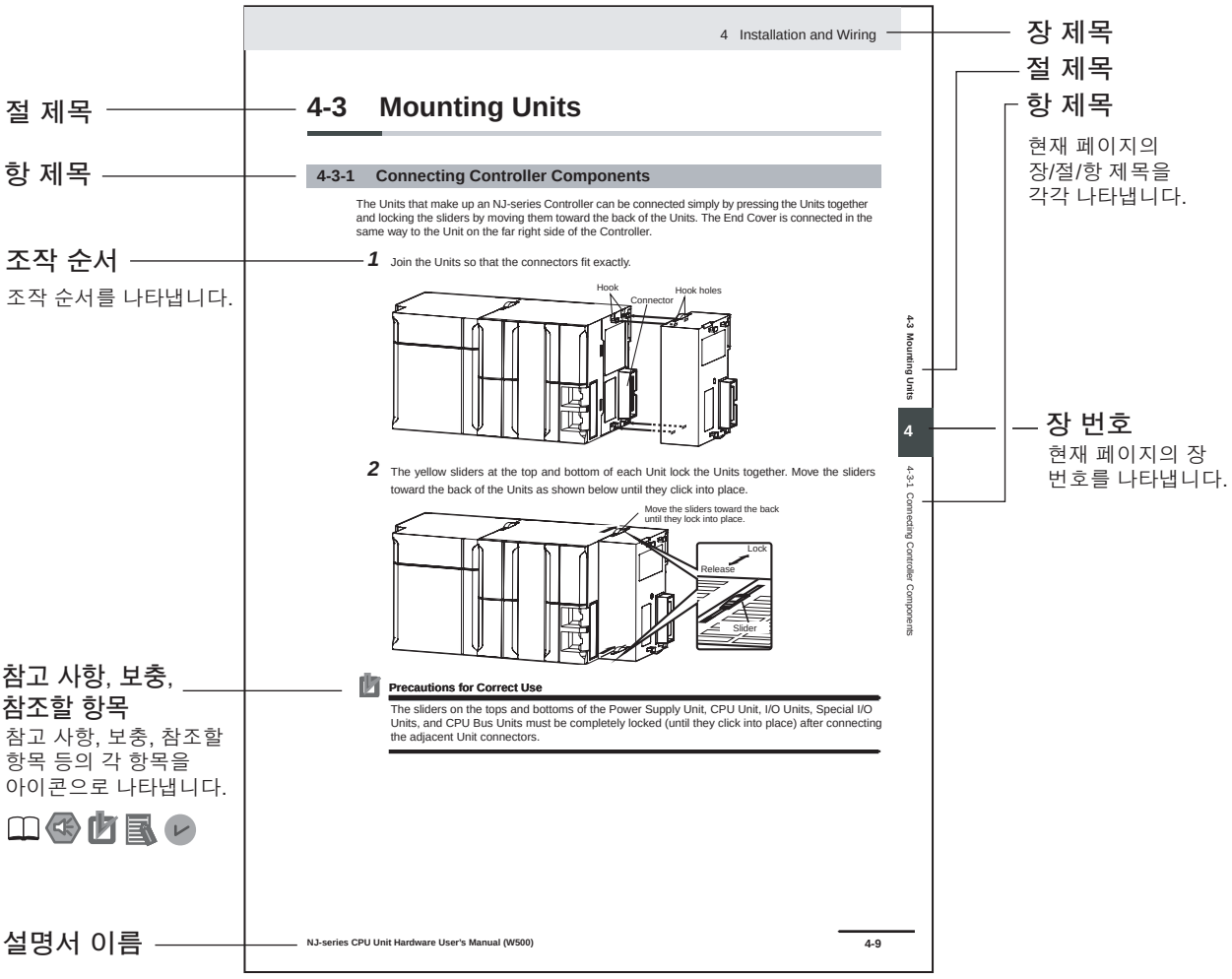
사용 목적	기본 정보							설명서										
	NX 시리즈 CPU 유니트 사용 설명서 하드웨어편	NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편	NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편	NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	NJ/NX 시리즈 CPU 유니트 사용 설명서 소프트웨어편	NX 시리즈 NX1P2 CPU 유니트 CPU 유니트 사용 설명서 내장 I/O, 옵션 보드 기능편	NJ/NX 시리즈 명령 참조 설명서 기본편	NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편	NJ/NX 시리즈 명령 참조 설명서 모션편	NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서	NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서	NJ/NX 시리즈 CPU 유니트 OPC UA편	NX 시리즈 CPU 유니트 사용 설명서 FINS 기능편	NJ/NX 시리즈 CPU 유니트 사용 설명서	NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용 설명서	NJ 시리즈 NJ Robotics CPU 유니트 사용 설명서	NJ/NY 시리즈 NC 통합 컨트롤러 사용 설명서	NJ/NX 시리즈 트러블 슈팅 설명서
사용자 프로그램 작성																		
모션 제어를 실시하는 경우								●	●									
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							
OPC UA를 사용하는 경우												●						
FINS를 사용하는 경우													●					
데이터베이스 연결 서비스를 사용하는 경우					●		●							●				
GEM 서비스를 사용하는 경우															●			
로봇 제어를 실시하는 경우																●		
수치 제어를 실시하는 경우																	●	
이상 처리를 집어넣는 경우																		●
NX1P2의 기능을 사용하는 경우						●												
동작 확인 및 디버그 실시																		
모션 제어를 실시하는 경우								●										
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							
OPC UA를 사용하는 경우												●						
FINS를 사용하는 경우													●					
데이터베이스 연결 서비스를 사용하는 경우					●									●				
GEM 서비스를 사용하는 경우															●			
로봇 제어를 실시하는 경우																●		
수치 제어를 실시하는 경우																	●	
NX1P2의 기능을 사용하는 경우						●												
이상 관리 기능 및 트러블 발생 시의 대처 방법 알아보기*1												△	△	△	△	△	△	●
보수 작업에 대해 알아보기	●	●	●	●														
모션 제어를 실시하는 경우								●										
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							

*1. 이상 관리의 개념 및 이상 항목에 대해서는 📖"NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오. 단, △ 마크가 부착된 설명서의 제품에서 발생하는 특유의 이상에 대한 자세한 내용은 △ 마크가 부착된 설명서를 참조하십시오.

설명서 읽는 방법

페이지 구성

이 설명서의 각 페이지 구성은 다음과 같습니다.



장 제목

절 제목

항 제목

현재 페이지의 장/절/항 제목을 각각 나타냅니다.

장 번호

현재 페이지의 장 번호를 나타냅니다.

절 제목

항 제목

조작 순서

조작 순서를 나타냅니다.

참고 사항, 보충, 참조할 항목

참고 사항, 보충, 참조할 항목 등의 각 항목을 아이콘으로 나타냅니다.

설명서 이름

이 페이지는 설명을 위한 견본이므로 실제 내용과는 다릅니다.

아이콘

이 설명서에서 사용하는 아이콘에는 다음과 같은 의미가 있습니다.

 **안전상의 중요 사항**

제품을 안전하게 사용하기 위해서 실시하거나 피해야 하는 사항을 나타냅니다.

 **사용상의 주의 사항**

제품의 동작 불능, 오작동 또는 성능 및 기능에 대한 악영향을 예방하기 위해 실시하거나 피해야 하는 사항을 나타냅니다.

 **참고**

필요에 따라 읽어야 하는 항목입니다.
알아두면 편리한 정보와 사용 시 참고가 되는 내용에 대해 설명합니다.



자세한 정보 또는 관련 정보가 기재되어 있는 위치를 나타냅니다.

 **버전 관련 정보**

CPU 유니트나 Sysmac Studio 버전에 따른 성능이나 기능의 차이를 설명합니다.

표기에 대한 유의 사항

이 설명서에서는 Sysmac Studio에서 실제 기기 컨트롤러에 데이터를 전송하는 것을 '다운로드', 실제 기기 컨트롤러에서 데이터를 Sysmac Studio에 전송하는 것을 '업로드'라고 기술합니다.
Sysmac Studio에서는 '다운로드' 및 '업로드'는 모두 '동기' 기능을 사용합니다. Sysmac Studio의 '동기' 기능은 PC 상의 데이터와 실제 기기 컨트롤러의 데이터를 자동 비교하여 어느 방향으로 전송할지를 사용자가 선택하는 기능입니다.

목차의 구성

1 NJ/NX 시리즈 컨트롤러의 개요

2 CPU 유니트의 동작

3 I/O 포트와 슬레이브/유니트 구성

4 컨트롤러 설정

5 태스크 설계

6 프로그래밍

7 동작 확인 및 운전

8 CPU 내장 기능

9 백업 기능

10 통신 설정

11 실제 사용 순서 예

12 이상의 확인과 처리

A 부록

I 색인

목차

머리말 1
대상 독자 1
대상 제품 1

설명서의 구성 2

설명서 읽는 방법 4
페이지 구성 4
아이콘 5
표기에 대한 유의 사항 5

목차의 구성 7

동의 사항 18

안전상의 주의 사항 20

안전상의 중요 사항 21

사용상의 주의 사항 22

법규와 규격 23
해외에서 사용하는 경우 23
EU 지령에 대한 적합성 23
KC 규격에 대한 적합성 24
선박 규격에 대한 적합성 24
소프트웨어 라이센스 및 저작권 24

버전 25
버전 확인 방법 25
CPU 유니트의 유니트 버전과 Sysmac Studio의 버전 29

관련 설명서 30

용어 설명 34

설명서 개정 이력 38

제1장 NX 시리즈 컨트롤러의 개요

1-1 NJ/NX 시리즈 컨트롤러의 개념 1-2
1-1-1 특징 1-2
1-1-2 시스템 구성의 개요 1-5
1-2 사양의 개요 1-13
1-3 NJ/NX 시리즈의 사용 순서 전체 1-16
1-3-1 순서의 개요 1-16
1-3-2 자세한 순서 1-17

제2장 CPU 유니트의 동작

2-1 CPU 유니트 동작의 개요 2-2
2-1-1 CPU 유니트 구조의 개요 2-2
2-1-2 CPU 유니트의 각 상태의 동작 개요 2-3

2-2 소프트웨어의 구조 2-4
2-2-1 소프트웨어 구성 2-4
2-2-2 소프트웨어의 동작 2-5

2-3 I/O와의 액세스 구조 2-12
2-3-1 변수의 종류 2-12
2-3-2 변수와 I/O의 액세스 2-15

2-4 NX 버스 기능 모듈의 I/O 리프레시 2-24
2-4-1 I/O 리프레시 방식 2-24
2-4-2 각 I/O 리프레시 방식의 동작 2-25

2-5 시퀀스 제어와 모션 제어의 구조 2-28
2-5-1 제어의 개요 2-28
2-5-2 시퀀스 제어의 구조 2-30
2-5-3 모션 제어의 구조 2-32
2-5-4 시퀀스 제어와 모션 제어의 동기 2-34

2-6 CPU 유니트 내의 데이터 개요 2-35

2-7 CPU 유니트의 각 상태의 동작 2-36
2-7-1 CPU 유니트의 상태 2-36
2-7-2 각 상태의 동작 2-38
2-7-3 동작 모드 2-40

제3장 I/O 포트와 슬레이브/유니트 구성

3-1 슬레이브/유니트 구성 순서 3-2

3-2 슬레이브/유니트 구성의 작성과 비교 3-4
3-2-1 EtherCAT 슬레이브 구성의 작성 3-4
3-2-2 유니트 구성의 작성 3-5
3-2-3 유니트 구성의 비교 3-5

3-3 I/O 포트와 디바이스 변수 3-7
3-3-1 I/O 포트 3-8
3-3-2 I/O 포트명 3-9
3-3-3 디바이스 변수 3-11

3-4 CPU 유니트에 대한 변수 할당 3-14
3-4-1 유니트에 대한 변수 할당 방법 3-14
3-4-2 유니트에 할당한 변수의 사용 방법 3-16

3-5 '축'의 작성과 서보/엔코더 입력 슬레이브 등의 할당 3-18
3-5-1 개요 3-18
3-5-2 축 변수/축 그룹 변수 3-18
3-5-3 '축' 및 축 변수 작성과 사용 방법 3-20

제4장 컨트롤러 설정

4-1 컨트롤러 설정의 개요 4-2

4-2 PLC 기능 모듈의 초기 설정 4-4
4-2-1 개요 4-4
4-2-2 컨트롤러 설정 4-4
4-2-3 태스크 설정 4-7
4-2-4 NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정 4-12
4-2-5 NJ 시리즈 CPU 유니트에서의 유니트 구성·유니트 설정 4-15

4-3 NX 유니트의 초기 설정 4-16
4-3-1 NX 유니트의 설정 항목 4-16
4-3-2 I/O 할당 설정 4-17
4-3-3 유니트 동작 설정 4-19

4-4 고기능 유니트의 초기 설정 4-20

4-5 모션 제어 기능 모듈의 초기 설정 4-22
4-5-1 개요 4-22
4-5-2 설정 방법 4-23

4-6 EtherCAT 마스터 기능 모듈의 초기 설정 4-24
4-7 EtherNet/IP 기능 모듈의 초기 설정 4-25
4-8 내장 I/O 초기 설정 4-26
4-9 옵션 보드 초기 설정 4-27
4-10 CJ 유니트용 메모리 설정 4-28
4-10-1 설정 순서 4-28
4-10-2 설정 화면 4-29
4-10-3 설정 내용 4-30

제5장 태스크 설계

5-1 태스크 설계 순서의 개요 5-3
5-2 태스크의 개요 5-5
5-2-1 태스크의 개념 5-5
5-2-2 태스크 관련 명령 5-7
5-2-3 태스크 관련 시스템 정의 변수 5-7
5-3 NX701의 태스크 사양과 기본 동작 5-9
5-3-1 NX701의 태스크 사양 5-9
5-3-2 NX701의 태스크 분할 개념 5-10
5-3-3 NX701의 태스크 기본 동작 5-10
5-3-4 NX701의 이벤트 태스크의 실행 조건 5-19
5-3-5 NX701의 이벤트 태스크 실행 타이밍 5-24
5-3-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-28
5-4 NX102 및 NX1P2의 태스크 사양과 기본 동작 5-29
5-4-1 NX102 및 NX1P2의 태스크 사양 5-29
5-4-2 NX102 및 NX1P2의 태스크 분할의 개념 5-30
5-4-3 NX102 및 NX1P2의 태스크 기본 동작 5-30
5-4-4 NX102 및 NX1P2의 이벤트 태스크 실행 조건 5-36
5-4-5 NX102 및 NX1P2의 이벤트 태스크 실행 타이밍 5-41
5-4-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-45
5-5 NJ 시리즈의 태스크 사양과 기본 동작 5-46
5-5-1 NJ 시리즈의 태스크 사양 5-46
5-5-2 NJ 시리즈의 태스크 분할의 개념 5-47
5-5-3 NJ 시리즈의 태스크 기본 동작 5-47
5-5-4 NJ 시리즈의 이벤트 태스크 실행 조건 5-55
5-5-5 NJ 시리즈의 이벤트 태스크 실행 타이밍 5-60
5-5-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-64
5-6 태스크 이외의 서비스 5-65
5-6-1 태스크 이외의 서비스 실행 우선도와 실행 순서 5-67
5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍 5-70
5-6-3 옵션 보드 서비스의 처리 내용과 실행 타이밍 5-74
5-6-4 시스템 서비스의 처리 내용과 실행 타이밍 5-75
5-7 태스크 관련 할당과 설정 5-78
5-7-1 I/O 리프레시의 태스크에 대한 할당 5-78
5-7-2 프로그램의 태스크에 대한 할당 5-86
5-7-3 태스크의 설정 항목 5-87
5-8 변수의 동시성 확보 5-90
5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법 5-90
5-8-2 컨트롤러 외부에서의 변수 액세스 5-96
5-9 태스크 관련 이상 5-100
5-10 태스크 실행 상태, 태스크 실행 시간의 모니터링 5-103
5-11 태스크 설계 방법 및 입출력 응답 시간 5-108
5-11-1 태스크 실행 시간 확인 5-108
5-11-2 태스크 설계의 예 5-111
5-11-3 시스템으로서의 입출력 응답 시간 5-112

제6장 프로그래밍

6-1 프로그래밍 순서의 개요 6-3

6-2 POU(프로그램 구성 단위) 6-5
6-2-1 POU의 개념 6-5
6-2-2 POU의 3가지 타입의 개요 6-6
6-2-3 프로그램, 펑션 블록, 펑션의 차이점 6-7
6-2-4 프로그램의 세부 내용 6-8
6-2-5 펑션 블록(FB)의 세부 내용 6-9
6-2-6 펑션(FUN)의 세부 내용 6-18
6-2-7 펑션 블록(FB), 펑션(FUN) 공통 동작 6-24
6-2-8 POU의 제한 사항 6-26

6-3 변수 6-28
6-3-1 변수의 개념 6-28
6-3-2 변수의 종류 6-28
6-3-3 POU에서 본 사용자 정의 변수의 종류 6-29
6-3-4 변수의 속성 6-30
6-3-5 데이터 형식 6-31
6-3-6 파생 데이터 형식 6-41
6-3-7 데이터 형식의 배열/범위 지정 6-51
6-3-8 변수의 각 속성의 설명 6-59
6-3-9 상태 변화에 따른 변수 값의 변화 6-67
6-3-10 펑션 블록(FB)의 인스턴스 6-82
6-3-11 변수 값 모니터링 6-82
6-3-12 변수명 등의 프로그램 관련 이름의 제한 6-83

6-4 상수(리터럴) 6-85
6-4-1 상수의 개념 6-85
6-4-2 각 데이터 형식의 기술 형식 6-85

6-5 프로그램 언어 6-90
6-5-1 프로그램 언어의 개념 6-90
6-5-2 래더도 언어(LD) 6-90
6-5-3 ST 언어 6-96

6-6 명령 6-134
6-6-1 명령의 개념 6-134
6-6-2 명령의 기본 지식 6-134
6-6-3 명령 에러 6-137

6-7 이름 공간 6-143
6-7-1 이름 공간의 개념 6-143
6-7-2 이름 공간의 사양 6-144
6-7-3 이름 공간 사용 순서 6-147

6-8 라이브러리 6-148
6-8-1 라이브러리의 개요 6-148
6-8-2 라이브러리 사양 6-149
6-8-3 라이브러리 부품의 사양 6-150
6-8-4 라이브러리 사용 순서 6-151

6-9 사용자 프로그램 작성상의 유의 사항 6-152
6-9-1 입력 변수/출력 변수/입출력 변수가 배열 지정 변수인 경우 6-152
6-9-2 입력 변수/출력 변수/입출력 변수가 구조체 형식 변수인 경우 6-152
6-9-3 마스터 컨트롤 6-153

제7장 동작 확인 및 운전

7-1 동작 확인 및 운전 순서의 개요 7-2

7-2 오프라인 디버그 7-3
7-2-1 시뮬레이션 기능의 특징 7-3
7-2-2 시뮬레이션 기능의 실행 7-3
7-2-3 시뮬레이션 기능의 설정 7-6

7-3 실제 기기에서의 동작 확인 및 운전 7-8
7-3-1 실제 기기에서의 동작 확인에서 운전 시작까지의 순서 7-8
7-3-2 프로젝트 전송 7-9
7-3-3 I/O의 배선 확인 7-9
7-3-4 MC 시운전 7-9
7-3-5 사용자 프로그램의 동작 확인 7-10
7-3-6 본 운전의 시작 7-10

제8장 CPU 내장 기능

8-1 데이터 관리·시계·운전 관련 기능 8-3
8-1-1 메모리 전체 클리어 8-3
8-1-2 시계 기능 8-3
8-1-3 운전 중 출력 기능 8-6

8-2 NX 유니트 관련 관리 기능 8-7
8-2-1 NX 버스 기능 모듈 8-7
8-2-2 CPU 유니트 상의 NX 유니트 장착 설정 기능 8-12
8-2-3 CPU 유니트 상의 NX 유니트 관련 재시작 기능 8-15
8-2-4 CPU 유니트 상의 NX 유니트 배선 확인 8-16
8-2-5 CPU 유니트 상의 NX 유니트 FAIL-SOFT 동작 기능 8-17
8-2-6 CPU 유니트 상의 NX 유니트 통전 시간 모니터링 기능 8-21

8-3 CJ 유니트 관련 관리 기능 8-22
8-3-1 기본 I/O 유니트 관련 8-22
8-3-2 고기능 유니트 관련 8-23

8-4 SD 메모리 카드 기능 8-25
8-4-1 SD 메모리 카드 기능 목록 8-25
8-4-2 지원하는 SD 메모리 카드 및 폴더/파일 사양 8-26
8-4-3 SD 메모리 카드 조작 명령 8-28
8-4-4 FTP 클라이언트 통신 명령 8-28
8-4-5 FTP 서버 기능 8-29
8-4-6 Sysmac Studio에서의 파일 조작 8-29
8-4-7 SD 메모리 카드의 수명 감지 기능 8-29
8-4-8 SD 메모리 카드 관련 시스템 정의 변수 목록 8-30
8-4-9 SD 메모리 카드의 자체 진단 기능 8-31
8-4-10 SD 메모리 카드 내의 파일 액세스의 배타 제어 8-33

8-5 보안 기능 8-34
8-5-1 사용자 프로그램 실행용 ID 인증 기능 8-35
8-5-2 사용자 프로그램 복원 정보 없는 전송 기능 8-38
8-5-3 프로젝트 파일 전체 보호 기능 8-39
8-5-4 데이터 보호 기능 8-40
8-5-5 조작 권한 인증 기능 8-42
8-5-6 CPU 유니트에 대한 쓰기 방지 기능 8-44
8-5-7 CPU 유니트 이름 기능 및 시리얼 ID 기능 8-45

8-6 디버그 기능 8-47
8-6-1 강제값 리프레시 8-47
8-6-2 현재값 변경 8-52
8-6-3 온라인 편집 8-54
8-6-4 데이터 트레이스 8-56
8-6-5 미분 모니터링 8-61

8-7 이벤트 로그 기능 8-66
8-7-1 개요 8-66
8-7-2 이벤트 로그 기능의 세부 내용 8-68
8-7-3 컨트롤러 이벤트(컨트롤러 이상/정보) 8-73
8-7-4 사용자 이벤트(사용자 이상/정보) 8-74

8-8 이벤트 중요도 변경 기능 8-80
8-8-1 이벤트 중요도 변경의 용도 8-80
8-8-2 이벤트 중요도 변경 가능한 이벤트 8-80
8-8-3 이벤트 중요도 변경 조작 방법 8-81

제9장 백업 기능

9-1 전체적인 백업 기능 9-3
9-1-1 백업 기능의 용도 9-3
9-1-2 백업 기능 조작 순서의 예 9-4
9-1-3 백업 기능의 대상 데이터 9-6
9-1-4 백업 기능의 종류 9-8
9-1-5 백업 기능의 종류와 데이터 그룹의 관계 9-10
9-1-6 백업 기능의 대상 범위 9-11

9-2 SD 메모리 카드 백업 기능 9-13
9-2-1 백업(컨트롤러에서 SD 메모리 카드) 9-14
9-2-2 복원(SD 메모리 카드에서 컨트롤러) 9-19
9-2-3 비교(컨트롤러와 SD 메모리 카드 간) 9-26

9-3 SD 메모리 카드에 대한 백업 금지 기능 9-31

9-4 SD 메모리 카드 자동 전송 기능 9-32

9-5 SD 메모리 카드 프로그램 전송 기능 9-34

9-6 Sysmac Studio 컨트롤러 백업 기능 9-40
9-6-1 백업(컨트롤러에서 PC) 9-41
9-6-2 복원(PC에서 컨트롤러) 9-42
9-6-3 비교(컨트롤러와 PC 간) 9-44

9-7 Sysmac Studio 백업 파일 가져오기/내보내기 기능 9-45

9-8 Sysmac Studio 변수/메모리 백업 기능 9-46
9-8-1 Sysmac Studio 변수/메모리 백업 기능의 대상 데이터 9-46
9-8-2 Sysmac Studio 변수/메모리 백업 기능의 조작 방법 9-46
9-8-3 CPU 유니트 형식 간의 호환성 9-47

9-9 EtherCAT 슬레이브 연결 시의 백업 기능 9-49
9-9-1 EtherCAT 슬레이브의 백업 대상 데이터 9-49
9-9-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-50
9-9-3 EtherCAT 슬레이브의 복원 실행 조건 9-51
9-9-4 백업 기능의 대상 EtherCAT 슬레이브 9-52

9-10 EtherCAT 슬레이브 터미널 연결 시의 백업 기능 9-55
9-10-1 EtherCAT 슬레이브 터미널의 백업 대상 데이터 9-55
9-10-2 EtherCAT 슬레이브 터미널의 각 상태에서의 백업 기능 실행 가능 여부 9-56
9-10-3 EtherCAT 슬레이브 터미널의 복원 실행 조건 9-56

9-11 NX 유니트 연결 시의 백업 기능 9-57
9-11-1 CPU 유니트 상의 NX 유니트의 백업 대상 데이터 9-57
9-11-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-57
9-11-3 CPU 유니트 상의 NX 유니트의 복원 실행 조건 9-58

9-12 CJ 유니트 연결 시의 백업 기능 9-59
9-12-1 CJ 유니트의 백업 대상 데이터 9-59
9-12-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-59
9-12-3 CJ 유니트의 복원 실행 조건 9-60

9-13 백업 관련 파일 9-61
9-13-1 백업 관련 파일의 종류 9-61
9-13-2 백업 파일의 사양 9-63
9-13-3 복원 명령 파일의 사양 9-64
9-13-4 자동 전송 명령 파일의 사양 9-66
9-13-5 컨트롤러 비교 결과 파일의 사양 9-69
9-13-6 EtherCAT 비교 결과 파일의 사양 9-70
9-13-7 EtherCAT 슬레이브 터미널 비교 결과 파일의 사양 9-71
9-13-8 NX 유니트 비교 결과 파일의 사양 9-72
9-13-9 CJ 유니트 비교 결과 파일의 사양 9-73

9-14 백업 기능의 파일 간 호환성 9-75
9-14-1 백업 기능 간 호환성 9-75
9-14-2 CPU 유니트의 형식 간의 호환성 9-76
9-14-3 CPU 유니트의 유니트 버전 호환성 9-77

9-15 백업 기능과 동시에 실행할 수 없는 기능 9-78

제10장 통신 설정

10-1 통신 시스템의 개요 10-2
10-1-1 개요 10-3
10-2 Sysmac Studio 연결 10-7
10-2-1 온라인 연결이 가능한 구성 10-7
10-2-2 온라인 연결할 수 없는 구성 10-8
10-3 다른 컨트롤러 및 슬레이브와의 연결 10-10
10-3-1 컨트롤러와 컨트롤러 간의 연결 구성 10-10
10-3-2 컨트롤러-슬레이브 간의 연결 구성 10-13
10-4 표시기 및 시리얼 통신 기기와의 연결 10-14
10-4-1 표시기와의 연결 10-14
10-4-2 시리얼 통신 기기와의 연결 10-15

제11장 실제 사용 순서 예

11-1 실제 사용 예 11-2
11-1-1 시스템 구성 11-2
11-1-2 동작 11-2
11-2 조작 순서 예의 개요 11-3
11-2-1 배선·설정 11-3
11-2-2 소프트웨어 설계 11-3
11-2-3 Sysmac Studio에서의 소프트웨어 설정 11-4
11-2-4 Sysmac Studio에서의 프로그래밍 11-8
11-2-5 Sysmac Studio에서의 시뮬레이션 11-9
11-2-6 실제 기기에 의한 동작 확인과 운전 11-9

제12장 이상의 확인과 처리

12-1 이상의 확인과 처리의 개요 12-2

부록

A-1 사양 A-3
A-1-1 일반 사양 A-3
A-1-2 성능 사양 A-3
A-1-3 기능 사양 A-13
A-2 NX701에서의 태스크 실제 처리 시간의 기준 계산 A-19
A-2-1 태스크 실제 처리 시간의 평균값 계산 방법 A-20
A-2-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-27
A-3 NX102에서의 태스크 실제 처리 시간의 기준 계산 A-30
A-3-1 태스크 실제 처리 시간의 평균값 계산 방법 A-31
A-3-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-37
A-4 NX1P2에서의 태스크 실제 처리 시간의 기준 계산 A-40
A-4-1 태스크 실제 처리 시간의 평균값 계산 방법 A-41
A-4-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-47
A-5 NJ 시리즈에서의 태스크 실제 처리 시간의 기준 계산 A-50
A-5-1 태스크 실제 처리 시간의 평균값 계산 방법 A-51
A-5-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-61
A-6 시스템 정의 변수 목록 A-64
A-6-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음) A-65
A-6-2 PLC 기능 모듈(카테고리명: _PLC) A-74
A-6-3 PLC 기능 모듈(카테고리명: _CJB) A-78
A-6-4 NX 버스 기능 모듈(카테고리명: _NXB) A-80

A-6-5 모션 제어 기능 모듈(카테고리명: _MC) A-82
A-6-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC) A-84
A-6-7 EtherNet/IP 기능 모듈(카테고리명: _EIP) A-89
A-6-8 이상 Status 각 비트의 의미 A-112

A-7 시스템 정의 변수의 개별 사양 A-114
A-7-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음) A-115
A-7-2 PLC 기능 모듈(카테고리명: _PLC) A-131
A-7-3 PLC 기능 모듈(카테고리명: _CJB) A-135
A-7-4 NX 버스 기능 모듈(카테고리명: _NXB) A-139
A-7-5 모션 제어 기능 모듈(카테고리명: _MC) A-142
A-7-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC) A-145
A-7-7 EtherNet/IP 기능 모듈(카테고리명: _EIP) A-154

A-8 CPU 유니트 내 데이터 속성 A-176

A-9 CJ 유니트용 메모리의 세부 내용 A-180
A-9-1 CIO 영역 A-180
A-9-2 내부 보조 릴레이 A-183
A-9-3 유지 릴레이 A-183
A-9-4 데이터 메모리 A-183
A-9-5 확장 데이터 메모리 A-184

A-10 변수의 메모리 확보 방법 A-185
A-10-1 변수의 메모리 확보 규칙 A-185
A-10-2 유의해야 하는 케이스 예 A-194

A-11 CX-Designer의 변수 테이블 등록 방법 A-198

A-12 EtherCAT 슬레이브 및 축 유효/무효 전환 A-201
A-12-1 EtherCAT 슬레이브 및 축을 사용하는 경우의 프로젝트 설정 A-201
A-12-2 EtherCAT 슬레이브 및 축 유효/무효 전환을 실행하는 명령 A-201
A-12-3 EtherCAT 슬레이브 및 축의 상태를 나타내는 시스템 정의 변수 A-202
A-12-4 프로그램의 실행/정지 명령 A-203
A-12-5 프로그램의 실행/정지 확인 A-203
A-12-6 Sysmac Studio에서의 설정 A-204
A-12-7 EtherCAT 슬레이브 및 축 유효/무효 전환의 어플리케이션 예 A-205

A-13 사용자 프로그램의 크기에 관한 제한 A-208
A-13-1 사용자 프로그램 각 부품의 제한 A-208
A-13-2 사용자 프로그램 각 부품의 수 A-212

A-14 Ver.1.02 이하 CPU 유니트의 교환 방법 A-214
A-14-1 CPU 유니트 내의 데이터 업로드 A-214
A-14-2 새로운 CPU 유니트 연결 A-217
A-14-3 CPU 유니트로의 데이터 다운로드 A-218

A-15 NX 시리즈의 버전 관련 정보 A-222
A-15-1 유니트 버전과 Sysmac Studio 버전의 관계 A-222
A-15-2 유니트 버전 갱신에 따른 지원 기능의 추가와 변경 A-224

A-16 NJ 시리즈의 버전 관련 정보 A-226
A-16-1 유니트 버전과 Sysmac Studio 버전의 관계 A-226
A-16-2 하드웨어 개정 번호와 Sysmac Studio 버전의 관계 A-228
A-16-3 유니트 버전 갱신에 따른 지원 기능의 추가와 변경 A-229
A-16-4 유니트 버전의 갱신에 의한 성능 개선 A-233

A-17 설명서 번호 변경 전의 개정 이력 A-235

색인

동의 사항

'당사 제품'에 대해 특별한 합의 사항이 없는 경우에는 고객의 구입처와 관계없이 이 동의 사항에 기재된 조건을 적용합니다.

● 정의

본 동의 사항에 대한 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 OMRON, 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, 사용 상의 주의 사항, 금지 사항 및 기타 사항
- '고객 용도': 고객이 '당사 제품'을 사용 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 규격 준수

● 기재 사항에 대한 주의 사항

'카탈로그 등'에 기재된 내용에 대해서는 다음 사항을 숙지하십시오.

- 정격값 및 성능값은 단독 시험에서 각 조건 하에서 얻은 값이며, 각 정격값 및 성능값의 복합 조건 하에서 얻는 값을 보증하지는 않습니다.
- 참고 데이터는 참고용으로 제공하는 것이며 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

다음 사항을 숙지한 후에 채택 또는 사용하십시오.

- 정격·성능 이외에 '사용 조건 등'을 준수하여 사용하십시오.
- 고객이 직접 '적합성 등'을 확인한 후에 '당사 제품'의 사용 가능 여부를 판단하십시오. '당사'는 '적합성 등'을 일체 보증하지 않습니다.
- '당사 제품'이 전체 고객 시스템 안에서 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 고장 나도 '고객 용도'의 위험을 최소화하는 안전 설계, (iii) 사용자에게 위험을 알리기 위한 안전 대책의 시스템 전체적인 구축, (iv) '당사 제품' 및 '고객 용도'의 정기적인 보수의 각 사항을 실시하십시오.
- '당사'는 DDoS 공격(분산형 DoS 공격) 및 컴퓨터 바이러스, 그 밖의 기술적인 유해 프로그램, 부정 액세스에 의해 '당사 제품' 및 설치된 소프트웨어 또는 모든 컴퓨터 기기, 컴퓨터 프로그램, 네트워크, 데이터 베이스가 감염되어도 그에 의해 직접 또는 간접적으로 발생한 손실, 손해, 그 밖의 비용에 대해 일절 책임을 지지 않습니다.
  고객이 직접 ① 안티 바이러스 보호, ② 데이터 입출력, ③ 분실 데이터 복원, ④ '당사 제품' 또는 설치된 소프트웨어에 대한 컴퓨터 바이러스 감염 방지, ⑤ '당사 제품'에 대한 부정 액세스 방지에 대한 충분한 조치를 하십시오.

- '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 경우에 '당사'는 '당사 제품'에 대해 어떠한 보증도 하지 않습니다. 단, 다음의 용도라도 '당사'가 의도한 특별한 제품 용도인 경우나 특별한 합의가 있는 경우에는 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 외의 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 및 충격을 받는 설비 등)
  (d) '카탈로그 등'에 기재되지 않은 조건이나 환경에서 쓰는 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 제품은 당사 영업 담당자에게 문의하십시오.

● 보증 조건

'당사 제품'의 보증 조건은 다음과 같습니다.

- 보증 기간: 구입 후 1년입니다.
  (단 '카탈로그 등'에 별도로 기재된 경우에는 제외됩니다.)
- 보증 내용: 고장 난 '당사 제품'에 대해 다음 중 하나를 '당사' 임의의 판단으로 실시합니다.
  (a) 당사 서비스 센터에서 고장 난 '당사 제품'을 무상 수리
  (단, 전자·기구 부품에 대해서는 수리하지 않습니다.)
  (b) 고장 난 '당사 제품'과 동일한 수량의 대체품 무상 제공
- 보증 대상 외: 고장의 원인이 다음 항목에 해당하는 경우에는 보증 대상에서 제외됩니다.
  (a) '당사 제품' 본래의 사용 방법 이외의 방법으로 사용
  (b) '사용 조건 등'에서 벗어난 사용
  (c) 본 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 경우에 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 사항의 내용은 다음 설명서를 참조하십시오.

- 🕮"NX 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-418)"
- 🕮"NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)"
- 🕮"NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)"
- 🕮"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"

안전상의 중요 사항

안전상의 중요 사항의 내용은 다음 설명서를 참조하십시오.

- 🕮"NX 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-418)"
- 🕮"NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)"
- 🕮"NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)"
- 🕮"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"

사용상의 주의 사항

사용상의 주의 사항은 다음 설명서를 참조하십시오.

- 🕮"NX 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-418)"
- 🕮"NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)"
- 🕮"NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)"
- 🕮"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"

법규와 규격

해외에서 사용하는 경우

본 제품 중 외환 및 해외 무역법의 규정에 따라 수출 허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우에 동법에 규정된 수출 허가, 승인(또는 직무 거래 허가)이 필요합니다.

EU 지령에 대한 적합성

적합한 지령

- EMC 지령
- 저전압 지령

적합의 개념

● EMC 지령

OMRON의 제품은 각종 기계, 제조 장치에 조립하여 사용하는 전기 기기이므로 조립한 기계 및 장치가 좀 더 간단하게 EMC 규격에 적합할 수 있도록 제품 자체와 관련된 EMC 규격(*)을 준수하기 위해 노력하고 있습니다.

그러나 고객의 기계 및 장치가 다양하며 EMC의 성능은 EU 지령 적합 제품을 조립한 기기 및 제어반의 구성, 배선 상태, 배치 상태 등에 따라 달라지므로 고객의 사용 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 중에
EMS(Electro-Magnetic Susceptibility: 전자파 내성)에 관해서는 EN 61131-2
EMI(Electro-Magnetic Interference: 전자파 간섭)에 대해서는 EN 61131-2
Radiated emission(복사성 방출)은 10m법에 따릅니다.

● 저전압 지령

전원 전압 50V AC~1,000V AC 및 75V DC~1,500V DC에서 동작하는 기기에 대해 필요한 안전성이 확보되어야 합니다. 적용 규격은 EN 61010-2-201입니다.

● EU 지령에 대한 적합성

NJ/NX 시리즈는 EU 지령에 적합합니다. 그러나 고객의 기계나 장치가 EU 지령에 적합하도록 하기 위해서는 다음 사항에 주의해야 합니다.

- NJ/NX 시리즈는 반드시 제어반 내에 설치하십시오.
- DC 전원 유니트 및 I/O 유니트에 연결하는 DC 전원은 SELV 사양 전원을 사용하십시오.
- NJ/NX 시리즈의 EU 지령 적합 제품은 EMI에 관한 공통 방출 규격에 적합하지만, 특히 Radiated emission(복사성 방출: 10m법)에 관해서는 사용하는 제어반의 구성, 연결되는 다른 기기와의 관계, 배선 등에 따라 달라질 수 있습니다.
  따라서 EU 지령 적합품인 NJ/NX 시리즈를 사용하는 경우에도 고객이 기계 및 장비 전체에서 EU 지령의 적합성을 확인하여 대처해야 합니다.

KC 규격에 대한 적합성

본 제품을 한국에서 사용하는 경우에는 다음 주의 사항을 준수하십시오.

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

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

선박 규격에 대한 적합성

본 제품은 각종 선박 규격에 적합합니다. 각 선박 규격에 적용하는 경우에 사용 조건이 설정되어 있으며, 설치 장소에 따라 사용할 수 없을 수도 있으므로 당사 영업사원에게 문의하십시오.

각 선박 규격의 사용 조건(NK, LR)

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

노이즈 필터

제조업체	형식
Cosel사	TAH-06-683

소프트웨어 라이센스 및 저작권

본 제품에는 타사 소프트웨어가 포함되어 있습니다. 이 소프트웨어와 관련된 라이센스 및 저작권은 http://www.fa.omron.co.jp/nj_info_j/를 참조하십시오.

버전

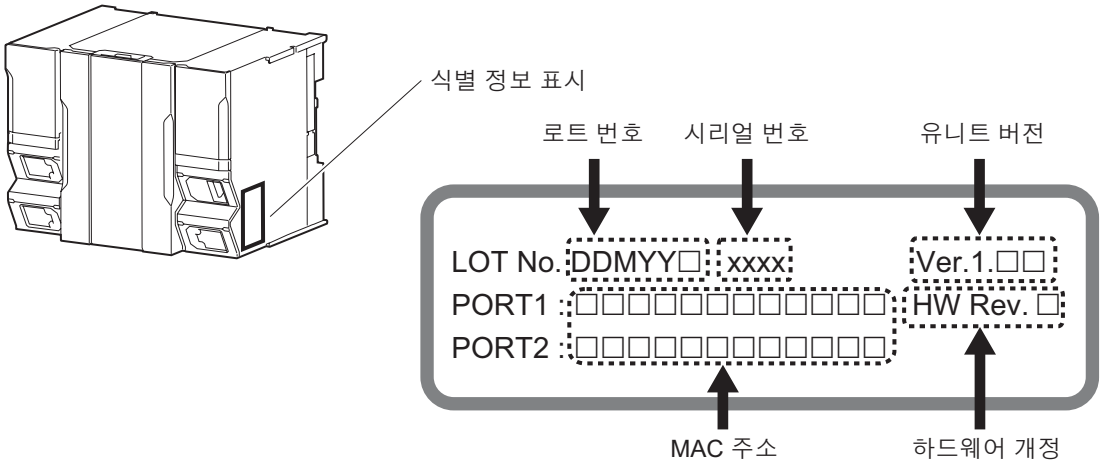
NJ/NX 시리즈의 각 유니트 및 EtherCAT 슬레이브는 하드웨어 개정 번호 및 유닛 버전 개정 번호로 하드웨어 및 소프트웨어의 각 버전을 관리합니다. 하드웨어 또는 소프트웨어에 사양 변경이 있을 때마다 하드웨어 개정 또는 유니트 버전이 갱신됩니다. 따라서 동일한 형식의 유니트나 EtherCAT 슬레이브라도 하드웨어 개정 또는 유니트 버전이 다르면 탑재 기능 및 성능이 다릅니다.

버전 확인 방법

버전은 식별 정보 표시 또는 Sysmac Studio를 통해 확인할 수 있습니다.

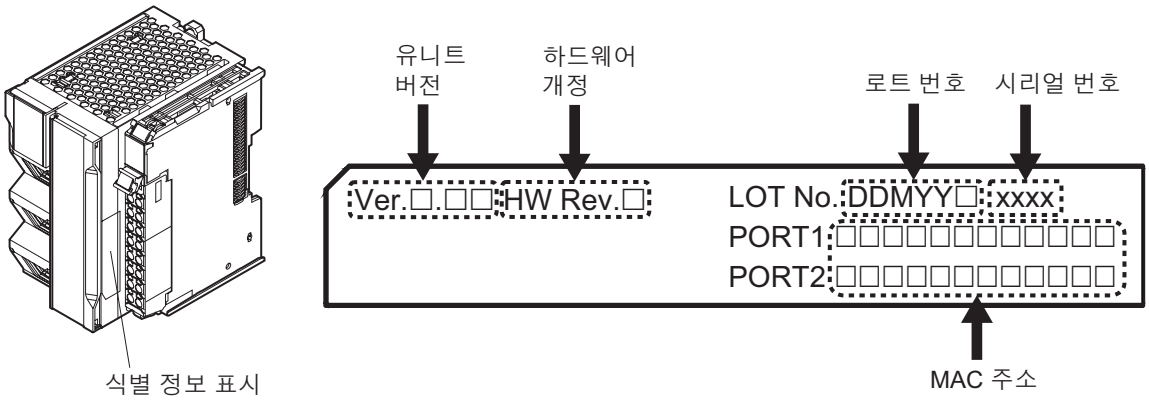
식별 정보 표시에 의한 확인

제품 측면의 식별 정보 표시를 통해 버전을 확인할 수 있습니다.
NX 시리즈 CPU 유니트 NX701-□□□□의 식별 정보 표시는 다음 그림과 같습니다.



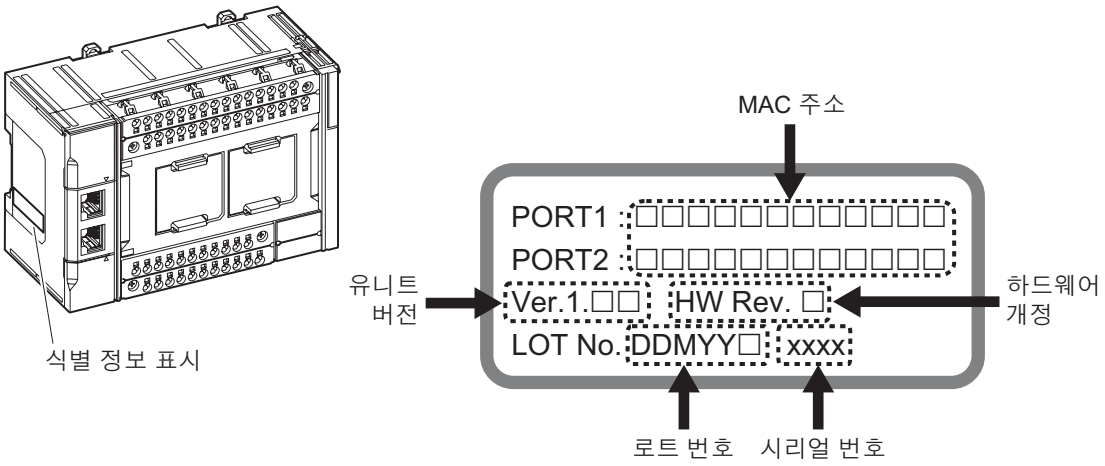
(주의) 하드웨어 개정이 '없음'인 유니트에서는 하드웨어 개정 표시가 없습니다.

NX 시리즈 CPU 유니트 NX102-□□□□의 식별 정보 표시는 다음 그림과 같습니다.



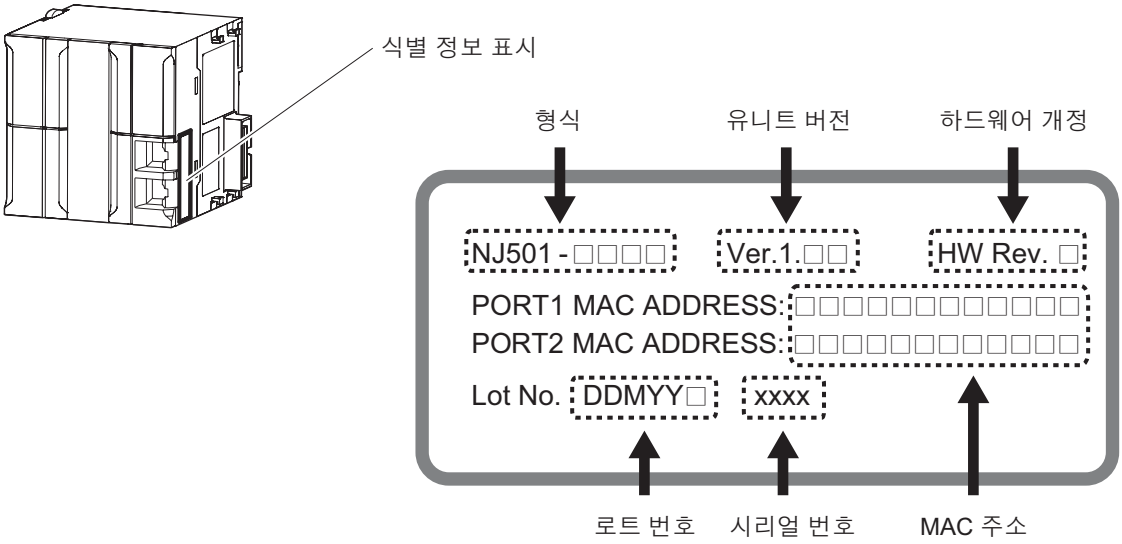
(주의) 하드웨어 개정이 '없음'인 유니트에서는 하드웨어 개정 표시가 없습니다.

NX 시리즈 CPU 유니트 NX1P2-□□□□□□□□의 식별 정보 표시는 다음 그림과 같습니다.



(주의) 하드웨어 개정이 '없음'인 유니트에서는 하드웨어 개정 표시가 없습니다.

NJ 시리즈 CPU 유니트 NJ501-□□□□의 식별 정보 표시는 다음 그림과 같습니다.



(주의) 하드웨어 개정이 '없음'인 유니트에서는 하드웨어 개정 표시가 없습니다.

Sysmac Studio를 통한 확인 방법

Sysmac Studio에서 버전을 확인할 수 있습니다. 유니트와 EtherCAT 슬레이브에서는 확인 방법이 다릅니다.

● NX 시리즈 CPU 유니트에서 버전 확인 방법

유니트 버전은 온라인 상태의 '생산 정보'를 통해 확인할 수 있습니다. 버전을 확인할 수 있는 유니트는 다음과 같습니다.

형식	버전을 확인할 수 있는 유니트
NX701-□□□□	CPU 유니트
NX102-□□□□	CPU 유니트, CPU 랙상의 NX 유니트
NX1P2-□□□□	CPU 유니트, CPU 랙상의 NX 유니트, 옵션 보드

1 **멀티 뷰 탐색기에서 [구성·설정]–[CPU·증설 랙]–[CPU 랙]을 마우스 오른쪽 버튼으로 클릭하고 [생산 정보 표시]를 선택합니다.**
[생산 정보] 사용자 대화 상자가 표시됩니다.

● NJ 시리즈 CPU 유니트에서 버전 확인 방법

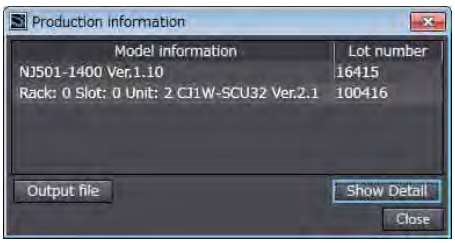
유니트 버전은 온라인 상태의 '생산 정보'를 통해 확인할 수 있습니다. 단, 버전을 확인할 수 있는 유니트는 CPU 유니트, CJ 시리즈의 고기능 I/O 유니트 및 CJ 시리즈의 CPU 고기능 유니트뿐입니다. CJ 유니트의 기본 I/O 유니트 버전은 Sysmac Studio에서 확인할 수 없습니다.

1 **멀티 뷰 탐색기의 [구성·설정]–[CPU·증설 랙]을 두 번 클릭합니다. 또는 [구성·설정]–[CPU·증설 랙]을 마우스 오른쪽 버튼으로 클릭하고 [편집]을 선택합니다.**
유니트 편집기가 표시됩니다.

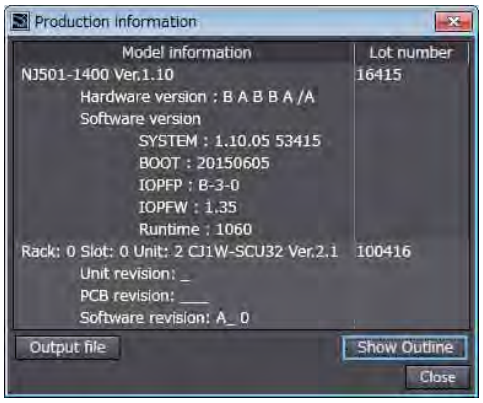
2 **유니트 편집기의 공란에서 마우스 오른쪽 버튼을 클릭하고 [생산 정보 표시]를 선택합니다.**
[생산 정보] 사용자 대화 상자가 표시됩니다.

● 생산 정보 표시 내용 전환

1 **[생산 정보] 사용자 대화 상자 오른쪽 아래의 [간이 표시] 또는 [상세 표시]를 선택합니다.**
[생산 정보]의 간이 표시와 상세 표시가 전환됩니다.



간이 표시



상세 표시

표시 내용은 간이 표시와 상세 표시에서 서로 다릅니다. 상세 표시에서는 유니트 버전, 하드웨어 버전 및 소프트웨어 버전이 표시됩니다. 간이 표시에서는 유니트 버전만 표시됩니다.

(주의) 하드웨어 개정은 하드웨어 버전의 우측에 '/'로 구분되어 표시됩니다. 하드웨어 개정이 '없음'인 유니트에서는 하드웨어 개정 표시가 없습니다.

● EtherCAT 슬레이브 버전 확인 방법

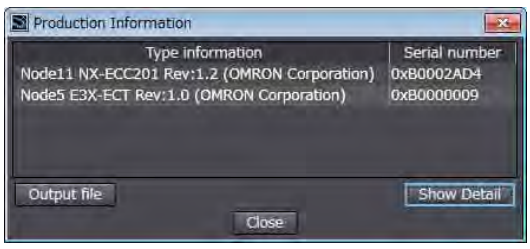
EtherCAT 슬레이브 버전은 온라인 상태의 [생산 정보]를 통해 확인할 수 있습니다. 확인 방법은 다음과 같습니다.

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

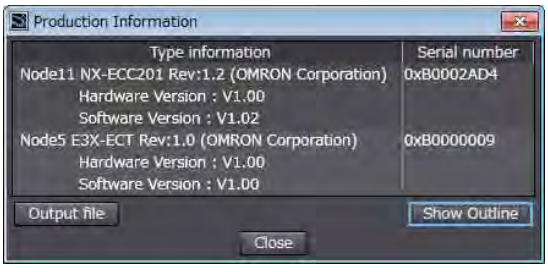
2 **EtherCAT 구성 편집 화면에서 마스터를 마우스 오른쪽 버튼으로 클릭하고 [생산 정보 표시]를 선택합니다.**
생산 정보 사용자 대화 상자가 표시됩니다.
유니트 버전은 'Rev'가 붙어서 표시됩니다.

● 생산 정보 표시 내용 전환

1 **[생산 정보] 사용자 대화 상자 오른쪽 아래의 [간이 표시] 또는 [상세 표시]를 선택합니다.**
[생산 정보]의 간이 표시와 상세 표시가 전환됩니다.



간이 표시



상세 표시

CPU 유니트의 유니트 버전과 Sysmac Studio의 버전

NJ/NX 시리즈 CPU 유니트의 유니트 버전에 따라 탑재한 기능이 다릅니다. 버전 업그레이드를 통해 추가된 기능을 사용할 때는 대응하는 버전의 Sysmac Studio를 사용해야 합니다.

CPU 유니트의 유니트 버전 종류와 Sysmac Studio의 버전의 관계 및 유니트 버전에 따른 지원 기능 목록은 'A-15 NX 시리즈의 버전 관련 정보(A-222페이지)' 및 'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

관련 설명서

이 설명서에 관련된 설명서는 다음 표와 같으므로 참조하시기 바랍니다.

설명서 이름	설명서 번호	형식	용도	내용
NX 시리즈 CPU 유니트 사용 설명서 하드웨어편	SBCA-418	NX701-□□□□	NX701 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고자 하는 경우, 주로 하드웨어 관련 정보	NX701의 시스템 전체 개요 및 CPU 유니트에 대해 다음 내용을 설명합니다. • 특징과 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검
NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편	SBCA-462	NX102-□□□□	NX102 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고자 하는 경우, 주로 하드웨어 관련 정보	NX102의 시스템 전체 개요 및 CPU 유니트에 대해 다음 내용을 설명합니다. • 특징과 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검
NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편	SBCA-448	NX1P2-□□□□	NX1P2 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고자 하는 경우, 주로 하드웨어 관련 정보	NX1P2의 시스템 전체 개요 및 CPU 유니트에 대해 다음 내용을 설명합니다. • 특징과 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검
NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	SBCA-358	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ 시리즈 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고자 하는 경우, 주로 하드웨어 관련 정보	NJ 시리즈 시스템 전체 개요 및 CPU 유니트에 대해 다음 내용을 설명합니다. • 특징과 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검
NJ/NX 시리즈 CPU 유니트 사용 설명서 소프트웨어편	SBCA-467	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 프로그래밍/시스템의 기동에 대해 알고자 하는 경우, 주로 소프트웨어 관련 정보	NJ/NX 시리즈 CPU 유니트에 대해 다음 내용을 설명합니다. • CPU 유니트의 동작 • CPU 유니트의 기능 • 초기 설정 • IEC 61131-3 기반 언어 사양과 프로그래밍
NX 시리즈 NX1P2 CPU 유니트 사용 설명서 내장 I/O, 옵션 보드 기능편	SBCA-449	NX1P2-□□□□	NX 시리즈 NX1P2 CPU 유니트에만 탑재하는 기능에 대한 자세한 내용 및 NJ/NX 시리즈의 기능 개요를 알고자 하는 경우	NX1P2 CPU 유니트의 기능 중 다음 내용을 설명합니다. • 내장 I/O • 시리얼 통신용 옵션 보드 • 아날로그 입출력용 옵션 보드 이와 함께 NJ/NX 시리즈 CPU 유니트의 다음 기능 개요를 설명합니다. • 모션 제어 기능 • EtherNet/IP 통신 기능 • EtherCAT 통신 기능

설명서 이름	설명서 번호	형식	용도	내용
NJ/NX 시리즈 명령 참조 설명서 기본편	SBCA-360	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈의 기본 명령 사양에 대해 자세히 알고자 하는 경우	각 명령(IEC 61131-3 사양)의 자세한 내용을 설명합니다.
NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편	SBCE-363	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	모션 제어의 설정이나 프로그래밍의 개념에 대해 알고자 하는 경우	모션 제어를 위한 CPU 유니트의 설정이나 동작, 프로그래밍의 개념에 대해 설명합니다.
NJ/NX 시리즈 명령 참조 설명서 모션편	SBCE-364	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	모션 명령 사양에 대해 자세히 알고자 하는 경우	각 모션 명령의 자세한 내용을 설명합니다.
NJ/NX 시리즈 CPU 유니트 내장 EtherCAT® 포트 사용 설명서	SBCD-358	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 내장 EtherCAT 포트를 사용하는 경우	내장 EtherCAT 포트에 대해 설명합니다. 개요, 구성, 기능, 설정을 설명합니다.
NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP™ 포트 사용 설명서	SBCD-359	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 내장 EtherNet/IP 포트를 사용하는 경우	내장 EtherNet/IP 포트에 관해 설명합니다. 기본 설정, 태그 데이터 링크, 기타 기능을 설명합니다.
NJ/NX 시리즈 CPU 유니트 사용 설명서 OPC UA편	SBCD-374	NX102-□□□□ NJ501-1□00	OPC UA를 사용하는 경우	OPC UA에 대해 설명합니다.
NX 시리즈 CPU 유니트 사용 설명서 FINS 기능편	SBCD-375	NX701-□□20 NX102-□□□□	NX 시리즈 CPU 유니트의 FINS 기능을 사용하는 경우	NX 시리즈 CPU 유니트의 FINS 기능에 대해 설명합니다.
NJ/NX 시리즈 데이터베이스 연결 CPU 유니트 사용 설명서	SBCA-411	NX701-□□20 NJ501-□□20 NJ101-□□20	NJ/NX 시리즈에서 데이터 베이스 연결 서비스 기능을 사용하는 경우	데이터 베이스 연결 서비스 기능에 대해 설명합니다.
NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용 설명서	SBCA-412	NJ501-1340	NJ 시리즈에서 GEM 서비스 기능을 사용하는 경우	GEM 서비스의 기능에 대해 설명합니다.
NJ 시리즈 NJ Robotics CPU 유니트 사용 설명서	SBCA-421	NJ501-4□□□	NJ 시리즈에서 로봇 제어를 실행하는 경우	로봇 제어 기능에 대해 설명합니다.
NJ/NY 시리즈 NC 통합 컨트롤러 사용 설명서	SBCE-428	NJ501-5300 NY532-5400	NJ/NY 시리즈에서 수치 제어를 실행하는 경우	수치 제어 기능에 대해 설명합니다.
NJ/NY 시리즈 명령 참조 설명서 G 코드편	SBCE-429	NJ501-5300 NY532-5400	G 코드/M 코드 명령 사양에 대해 자세히 알고자 하는 경우	G 코드/M 코드 명령의 자세한 내용을 설명합니다.
NJ/NX 시리즈 트러블 슈팅 설명서	SBCA-361	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈에서 검출하는 이상에 대해 자세히 알고자 하는 경우	NJ/NX 시리즈 시스템에서 검출하는 이상 관리의 개념과 각 이상 항목을 설명합니다.
Sysmac Studio Version 1 조작 설명서	SBCA-470	SYSMAC -SE2□□□	Sysmac Studio의 조작 방법, 기능에 대해 알고자 하는 경우	Sysmac Studio의 조작 방법을 설명합니다.
CNC Operator 조작 설명서	SBCE-430	SYSMAC -RTNC0□□□D	CNC Operator의 개요, 사용 방법을 알고자 하는 경우	CNC Operator의 개요, 설치 방법, 기본 조작, 연결 조작, 주요 기능의 조작 방법 등을 설명합니다.

설명서 이름	설명서 번호	형식	용도	내용
NX 시리즈 EtherCAT® 커플러 유니트 사용 설명서	SBCD-361	NX-ECC□□□	NX 시리즈 EtherCAT 커플러 유니트 및 EtherCAT 슬레이브 터미널의 사용 방법에 대해 알고자 하는 경우	NX 시리즈 EtherCAT 커플러 유니트와 NX 유니트로 구성되는 EtherCAT 슬레이브 터미널의 시스템 개요와 구성 방법 및 EtherCAT을 통해 NX 유니트를 설정, 제어, 모니터링하기 위한 EtherCAT 커플러 유니트의 하드웨어와 설정 방법, 기능에 대해 설명합니다.
NX 시리즈 데이터 참조 설명서	SBCA-410	NX-□□□□	NX 시리즈 각 유니트의 시스템 구성에 필요한 데이터를 목록으로 열람하고자 하는 경우	NX 시리즈 각 유니트의 '소비 전력', '질량' 등 시스템 구축에 필요한 데이터를 모아서 기재합니다.
NX 시리즈 NX 유니트 사용 설명서	SBCA-407	NX-ID□□□□ NX-IA□□□□ NX-OC□□□□ NX-OD□□□□ NX-MD□□□□	NX 유니트의 사용 방법에 대해 알고자 하는 경우	NX 유니트의 하드웨어나 설정 방법, 기능을 설명합니다. 다음과 같은 유니트 설명서가 있습니다. 디지털 I/O 유니트, 아날로그 I/O 유니트, 시스템 유니트, 위치 인터페이스 유니트, 커뮤니케이션 인터페이스 유니트, 로드 셀 입력 유니트, IO-Link 마스터 유니트
	SBCA-408	NX-AD□□□□ NX-DA□□□□		
	SBCA-440	NX-TS□□□□ NX-HB□□□□		
	SBCA-409	NX-PD1□□□ NX-PF0□□□ NX-PC0□□□ NX-TBX01		
	SBCE-374	NX-EC0□□□ NX-ECS□□□ NX-PG0□□□		
	SBCA-422	NX-CIF□□□		
	SBCA-439	NX-RS□□□□		
	SBCD-370	NX-ILM□□□		
NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	SGFM-710	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	NX 시리즈 세이프티 컨트롤 유니트의 사용 방법에 대해 알고자 하는 경우	NX 시리즈 세이프티 컨트롤 유니트의 하드웨어나 설정 방법, 기능을 설명합니다.
화상 처리 시스템 FH/FZ5 시리즈 사용 설명서	SDNB-712	FH-1□□□ FH-3□□□ FZ5-L35□ FZ5-6□□ FZ5-11□□	화상 처리 시스템 FH/FZ5 시리즈의 사용 방법에 대해 알고자 하는 경우	FH/FZ5 시리즈를 사용하기 위한 소프트웨어적인 기능, 설정, 조작에 대해 설명합니다.
시각 센서 FQ-M 시리즈 위치 결정 전용 시각 센서 사용 설명서	SDNB-706	FQ-MS12□	위치 결정 전용 시각 센서의 사용 방법에 대해 알고자 하는 경우	위치 결정 전용 시각 센서의 하드웨어나 설정 방법, 기능을 설명합니다.
시각 센서 FZ3 시리즈 사용 설명서	SCHB-762	FZ3-□□□□	시각 센서 FZ3 시리즈의 사용 방법에 대해 알고자 하는 경우	시각 센서 FZ3 시리즈의 소프트웨어적인 기능, 설정, 조작에 대해 설명합니다.
변위 센서 ZW 시리즈 화이버 동축 변위 센서 사용 설명서	SDNE-703	ZW-CE1□	변위 센서 ZW 시리즈의 사용 방법에 대해 알고자 하는 경우	변위 센서 ZW 시리즈의 하드웨어 및 설정 방법, 기능에 대해 설명합니다.
CJ 시리즈 고기능 유니트 사용 설명서 (NJ 시리즈 연결편)	SBCC-846	CJ1W-AD□□□ CJ1W-DA□□□ CJ1W-MAD42	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트에서 사용하는 방법을 알고자 하는 경우	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트에서 사용하는 방법(액세스 방법, 사용자 프로그램 I/F 등)이나 주의 사항에 대해 설명합니다. 다음과 같은 유니트 설명서가 있습니다. 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
	SBCC-847	CJ1W-TC□□□		
	SBCC-848	CJ1W-CT021		
	SBCC-849	CJ1W-PDC15 CJ1W-PH41U CJ1W-AD04U		
	SBCD-353	CJ1W-CRM21		
	SBCD-354	CJ1W-SCU□□		
	SBCD-355	CJ1W-EIP21		
	SBCD-357	CJ1W-DRM21		
	SDGR-703	CJ1W-V680□□□		

설명서 이름	설명서 번호	형식	용도	내용
프로그래머블 터미널 NA 시리즈 사용 설명서 소프트웨어편	SBSA-546	NA5-□W□□□□	프로그래머블 터미널 NA 시리즈의 페이지나 각 개체의 기능에 대해 알고자 하는 경우	프로그래머블 터미널 NA 시리즈의 페이지나 각 개체의 기능을 설명합니다.
프로그래머블 터미널 NS 시리즈 프로그래밍 설명서	SBSA-555	NS15-□□□□□ NS12-□□□□□ NS10-□□□□□ NS8-□□□□□ NS5-□□□□□	프로그래머블 터미널 NS 시리즈의 사용 방법에 대해 알고자 하는 경우	프로그래머블 터미널 NS 시리즈의 설정 방법 및 기능을 설명합니다.
CX-Designer 사용 설명서	SBSA-532	–	프로그래머블 터미널 NS 시리즈의 화면 데이터를 작성하고자 하는 경우	CX-Designer의 조작 방법에 대해 설명합니다.

용어 설명

용어	설명
AT(할당대상) 지정	변수의 속성 중 하나 변수의 할당대상을 사용자가 지정하는 속성, I/O 포트 또는 CJ 유니트용 메모리의 주소를 지정할 수 있음
CJ 시리즈 유니트	CJ 시리즈의 각종 유니트 중 NJ 시리즈 컨트롤러에서 사용할 수 있는 것
CJ 유니트용 메모리	CJ 시리즈 유니트 및 CJ 시리즈 네트워크에 액세스하는 경우 등에 변수의 할당대상으로 지정하는 I/O 메모리의 일종, 'AT(할당대상) 지정 있음' 속성의 변수로만 액세스할 수 있음
CPU 유니트	머신 자동화 컨트롤러 제어의 중심이 되는 유니트, 태스크 실행 및 각 유니트, 슬레이브의 I/O 리프레시 등을 실시함, NJ/NX 시리즈에서는 NX701-□□□□, NX1P2-□□□□, NJ501-□□□□ 등.
EtherCAT 마스터 기능 모듈	기능 모듈 중 하나, EtherCAT 마스터로서 EtherCAT 슬레이브를 컨트롤함
EtherNet/IP 기능 모듈	기능 모듈 중 하나, EtherNet/IP 포트를 컨트롤함
FB	펑션 블록의 약칭
FUN	펑션의 약칭
I/O 포트	CPU 유니트가 외부 장치(슬레이브/유니트)와 데이터를 교환하기 위한 논리적인 인터페이스
I/O 맵 설정	I/O 포트에 대해 변수를 할당한 설정, I/O 포트와 변수 사이의 할당 정보
I/O 리프레시	사전에 정해진 메모리 영역의 외부와의 주기적인 데이터 교환
MC 시운전	모터의 동작이나 배선 확인을 Sysmac Studio에서 실행하는 기능
NX 버스	NX 시리즈의 내부 버스. NX1P2 CPU 유니트가 탑재됨
NX 유니트	NX 시리즈 유니트 중에서 외부 연결 기기의 I/O 처리 등을 실시하는 유니트, 통신 커플러 유니트는 NX 유니트에 포함되지 않음
PDO 통신	프로세스 데이터 통신의 약칭, 프로세스 데이터 통신 주기(=프라이머리 정주기 태스크의 태스크 주기)로 마스터-슬레이브 간에서 주기적으로 데이터 교환을 하는 통신
PLC 기능 모듈	기능 모듈 중 하나, 사용자 프로그램의 실행, 모션 제어 기능 모듈에 대한 지시, USB·SD 메모리 카드와의 인터페이스 등을 실시함
POU	Program Organization Unit. IEC 61131-3에서 프로그램 실행 모델의 단위 사용자 프로그램 전체를 구성하는 기본적인 단위로서 알고리즘과 로컬 변수 테이블을 기술함 프로그램, 펑션 블록, 펑션의 3종류가 있음
SDO 통신	EtherCAT 통신의 일종으로, 임의의 타이밍에 정보 전달을 실시하는 서비스 데이터 개체(Service Data Objects: SDO)를 사용한 통신
Sysmac Studio	NJ/NX 시리즈의 설정, 프로그래밍, 디버그 및 트러블 슈팅을 실시하는 PC용 소프트웨어, 모션 제어용 기능 및 시뮬레이터 기능 포함
업로드	Sysmac Studio의 '동기' 기능에 의해 컨트롤러에서 데이터를 Sysmac Studio로 전송하는 것
일반 정보	컨트롤러 정보 및 사용자 정보의 이벤트 중요도를 나타내는 랭크 중 하나 이상 상태는 아니지만 이벤트 로그에 기록하여 사용자에게 통지하는 정보
이벤트 로그 기능	이벤트(이상 또는 사상)를 인식하고 해당 기록을 남기는 기능
이벤트 설정	사용자 이상 및 사용자 정보를 정의하는 것
이벤트 태스크	태스크의 실행 조건이 성립했을 때 한 번만 사용자 프로그램을 실행하는 태스크
인라인 ST	래더도 프로그램 중에 그 일부로 ST 프로그램을 쓰는 것 또는 쓰인 ST 프로그램
엣지	변수의 속성 중 하나 BOOL 형식 변수의 값이 FALSE→TRUE로 상승했을 때 또는 TRUE→FALSE로 하강했을 때, 펑션 블록에 TRUE 값을 전달함

용어	설명
캠 데이터 변수	캠 데이터를 구조체 배열화한 변수 위상 및 변위 데이터로 이루어지는 구조체 형식의 배열 지정 변수
감시 정보	컨트롤러 정보 및 사용자 정보의 이벤트 중요도를 나타내는 랭크 중 하나 제어에 영향이 없는 레벨의 가벼운 이상이지만 이벤트 로그에 기록하여 사용자에게 통지하는 정보
기능 모듈	CPU 유니트의 소프트웨어 구성을 기능별로 분할한 단위
기본 데이터 형식	IEC 61131-3에서 정의된 데이터 형식 부울 형식, 비트 열 형식, 정수 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시간 형식, 날짜 시각 형식, 문자열 형식이 있음 사용자가 정의하는 데이터 형식인 '파생 데이터 형식'의 반대말
강제값 리프레시	사용자가 디버그 등을 할 때 외부로부터의 입력 또는 외부로의 출력을 강제값으로 리프레시하는 기능 강제값 리프레시 대상 영역에 대해서도 사용자 프로그램 명령으로 덮어쓸 수 있음
공용체 형식	파생 데이터 형식의 일종으로 같은 데이터를 다른 데이터 형식으로 취급할 수 있도록 한 데이터 형식
글로벌 변수	모든 POU(프로그램, 펑션 블록, 펑션)에 의해 읽고 쓰기 가능한 변수
가벼운 이상 레벨의 컨트롤러 이상	NJ/NX 시리즈 컨트롤러의 어떤 기능 모듈의 일부 제어가 불가능해지는 이상 가벼운 이상 발생 후에도 NJ/NX 시리즈 CPU 유니트는 운전을 계속함
고기능 유니트 설정	고기능 유니트용 할당 DM 및 유니트 고유 설정의 총칭
구조체 형식	파생 데이터 형식의 일종으로 여러 가지 다른 데이터 형식을 계층적으로 하나로 묶은 데이터 형식
상수	변수의 속성 중 하나 상수 지정을 실시한 변수는 명령, ST 언어의 연산자 또는 CIP 메시지 통신에 의해 값을 쓸 수 없게 됨
컨트롤러	CPU 유니트가 직접 제어하는 범위를 가리킴 NX 시리즈 시스템에서는 CPU 랙+EtherCAT 슬레이브(범용/서보 드라이버 포함) NJ 시리즈 시스템에서는 CPU 랙+증설 랙+EtherCAT 슬레이브(범용/서보 드라이버 포함)
컨트롤러 이상	NJ/NX 시리즈 시스템에서 정의한 이상 '전체 정지 이상 레벨', '부분 정지 이상 레벨', '가벼운 이상 레벨', '감시 정보' 컨트롤러 이벤트의 총칭
컨트롤러 이벤트	NJ/NX 시리즈 시스템 이벤트의 일종 시스템이 사용자에게 통지할 필요가 있다고 정의하는 이상, 정보, 정의하는 발생 요인을 시스템이 감지하면 컨트롤러 이벤트가 발생함
컨트롤러 정보	NJ/NX 시리즈 시스템에서 정의한 이상이 아닌 정보, '일반 정보'의 컨트롤러 이벤트
서보/엔코더 입력 슬레이브	EtherCAT 슬레이브 중 '축'으로 할당할 수 있는 슬레이브를 가리킴, NJ/NX 시리즈에서는 서보 드라이버와 엔코더 입력 슬레이브를 가리킴
축	모션 제어 기능 모듈 내의 기능 단위, 외부의 서보 드라이버 구동 기구, 엔코더 입력 슬레이브의 검출 기구 등을 할당함
축 그룹	모션 제어 기능 모듈 내의 복수의 '축'을 묶은 기능 단위
축 그룹 변수	축 그룹별 각종 Status 정보 및 일부 축 그룹 파라메터 설정 정보 등을 구조체로 정의한 시스템 정의 변수 모션 제어 명령의 축 그룹 지정 및 축 그룹의 명령 보간 속도나 이상 정보 등의 모니터링에 사용됨
축 변수	축별 각종 Status 정보 및 일부 축 파라메터 설정 정보 등을 구조체로 정의한 시스템 정의 변수 모션 제어 명령의 축 지정 및 축의 명령 현재 위치나 이상 정보 등의 모니터링에 사용됨
시스템 공통 처리	CPU 유니트가 태스크 내에서 I/O 리프레시나 사용자 프로그램 실행과 함께 실행하는 시스템 처리, 변수의 태스크 간 배타 처리, 데이터 트레이스 처리 등을 실시함

용어	설명
시스템 서비스	CPU 유니트가 태스크 처리의 비는 시간에 실행하는 처리, 통신 처리, SD 메모리 카드 액세스, 자체 진단 처리 등의 처리를 실행함
시스템 정의 변수	모든 속성이 시스템에 의해 정의되며, 사용자는 이를 일체 변경할 수 없는 변수
초기값	변수의 속성 중 하나 다음과 같은 경우에 설정되는 변수의 값 • 전원 투입 시 • 운전 모드로 전환 시 • 사용자 프로그램 전송에 따라 초기화하는 지정 시 • 전체 정지 이상 레벨의 컨트롤러 이상 발생 시
슬레이브	리모트 I/O를 실행하는 마스터에 대한 슬레이브
슬레이브 터미널	통신 커플러 유니트에 NX 유니트 그룹을 장착한 빌딩 블록 형식의 리모트 I/O 유니트의 총칭, 슬레이브 터미널은 슬레이브의 일종
슬레이브/유니트 구성	EtherCAT 구성 및 유니트 구성의 총칭
앱솔루트 엔코더 원점 위치 오프셋	앱솔루트 엔코더 장착 서보 드라이버의 현재 위치를 복원하기 위해 CPU 유니트가 기억하고 있는 원점 확정 후의 명령 위치와 앱솔루트 엔코더에서 읽은 절대값 데이터의 차
전체 정지 이상 레벨의 컨트롤러 이상	NJ/NX 시리즈 컨트롤러 전체의 제어가 불가능해지는 이상, CPU 유니트는 사용자 프로그램의 실행을 바로 정지하고 리모트 I/O를 포함하는 모든 슬레이브/유니트에 대해 부하를 차단함
다운로드	Sysmac Studio의 '동기' 기능에 의해 Sysmac Studio에서 컨트롤러로 데이터를 전송하는 것
태스크	'프로그램 실행 시의 속성'을 말함
태스크 주기	프라이머리 정주기 태스크 및 정주기 태스크의 실행 간격
통신 커플러 유니트	NX 유니트 그룹과 상위 네트워크 마스터를 네트워크 상에서 리모트 I/O 통신하기 위한 인터페이스 유니트의 총칭, 예를 들어 EtherCAT 커플러 유니트는 EtherCAT 통신 네트워크의 통신 커플러 유니트
정주기 태스크	주기마다 I/O 리프레시 및 사용자 프로그램 실행을 하는 태스크
장치	CPU 유니트에서 I/O 리프레시를 하는 대상의 총칭, EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 옵션 보드 및 CJ 유니트를 나타냄
장치 출력	CPU 유니트에서 I/O 리프레시를 하는 대상의 출력
디바이스 변수	I/O 포트를 통해 특정한 장치에 액세스하기 위한 변수
동기	NJ/NX 시리즈 컨트롤러와 Sysmac Studio의 정보를 자동으로 비교하여 차이 유무, 위치, 내용을 계층적으로 표시하고 일치시키는 기능
이름 공간	펑션, 펑션 블록 정의, 데이터 형식의 이름을 분할·계층화하여 관리하는 시스템
네트워크 공개	변수의 속성 중 하나 CIP 메시지 통신 또는 태그 데이터 링크 기능에 의해 해당 변수를 다른 컨트롤러나 상위 PC 등에서도 읽고 쓸 수 있게 함
배열 지정	변수의 지정 중 하나, 동일한 데이터 형식 요소를 여러 개 묶어서 하나의 변수로 삼은 것, 각 요소는 선두부터 시리얼 번호(첨자)로 지정함
파생 데이터 형식	사양을 사용자가 정의하는 데이터 형식, 구조체 형식, 공용체 형식, 열거 형식이 있음
범위 지정	변수의 지정 중 하나, 변수를 미리 정해진 범위 내의 값만 취할 수 있다고 명시함
범용 슬레이브	EtherCAT 슬레이브 중 '축'으로 할당할 수 없는 슬레이브를 가리킴
펑션	연산 처리 등 입력에 대해 출력이 전적으로 결정되는 부품을 작성할 때 사용하는 POU
펑션 블록	타이머나 카운터 등 상태에 따라 동일한 입력에 대해 출력이 다른 경우가 있는 부품을 작성할 때 사용하는 POU
부분 정지 이상 레벨의 컨트롤러 이상	NJ/NX 시리즈 컨트롤러의 어떤 기능 모듈 전체의 제어가 불가능해지는 이상, 부분 정지 이상 발생 후에도 NJ/NX 시리즈 CPU 유니트는 운전을 계속함
프라이머리 정주기 태스크	가장 우선도가 높은 태스크

용어	설명
프로그램	펑션, 펑션 블록과 나란히 취급되는 3종류의 POU 중 하나 태스크에 할당되어 동작함
프로세스 데이터 통신	EtherCAT 통신의 일종으로, 정주기에 실시간 정보 교환을 실행하는 프로세스 데이터 개체(Process Data Objects: PDO)를 사용한 통신, 'PDO 통신'이라고도 함
변수	사용자 프로그램 중에서 사용하는 임의의 수치나 문자열을 표현하는 개념 임의의 값을 대입하여 변수의 값을 변경할 수 있음, 항상 값이 변하지 않는 '상수'의 반대말
변수 메모리	AT(할당대상) 지정이 없는 변수의 현재값, 'AT(할당대상) 지정 없음' 속성의 변수로만 액세스
유지	변수의 속성 중 하나 다음 경우에 이 속성을 설정했을 때는 값을 유지하고, 설정하지 않았을 때는 초기값이 됨 • 전원 OFF 후의 전원 투입 시 • 운전 모드로 전환 시 • 사용자 프로그램 전송에 따라 초기화하지 않는 지정 시
명령	POU의 알고리즘 중에 기재하는 OMRON의 최소 단위의 처리 요소 명령에는 래더도(접점, 코일), FB 형식, FUN 형식, ST 언어 구문이 있음
메인 메모리	CPU 유니트가 OS 사용자 프로그램 실행에 사용하는 CPU 유니트 내부의 메모리
모션 제어 기능 모듈	기능 모듈 중 하나, 사용자 프로그램 내의 모션 기능에 의해 부여된 명령에 따라 모션 제어를 실행함
모션 제어 명령	모션 제어 기능을 실행하기 위한 펑션 블록 정의의 명령 모션 제어 기능 모듈에서는 PLCopen®의 모션 제어용 펑션 블록을 준수한 명령과 모션 제어 기능 모듈의 독자적인 명령을 가짐
사용자 이벤트	NJ/NX 시리즈 시스템 이벤트의 일종. 사용자가 정의하는 이벤트, 사용자 이상, 사용자 정보의 총칭
사용자 정의 변수	모든 변수 속성이 사용자에 의해 정의되고, 사용자에 의한 변경이 가능한 변수
사용자 프로그램	복수의 프로그램 집합체
유니트	CPU 랙 또는 증설 랙에 장착 가능한 기기
유니트 구성	Sysmac Studio에서 설정한 유니트의 구성 정보이며, CPU 유니트에 연결된 유니트의 종류 및 연결 위치의 정보로 이루어짐
리터럴	사용자 프로그램 내에 사용되는 상수 표현
열거 형식	파생 데이터 형식의 일종, 사전에 준비된 이름 목록(열거자) 중 하나를 값으로 삼는 데이터 형식
열거자	열거 형식 변수가 취득하는 복수의 값을 각각 문자로 표현한 것 열거 형식 변수의 값은 열거자 중 어느 하나를 취함
로컬 변수	정의된 POU 내에서만 액세스 가능한 변수, 복수의 POU에서 액세스 가능한 '글로벌 변수'의 반대말 로컬 변수에는 내부 변수, 입력 변수, 출력 변수, 입출력 변수, 외부 변수가 포함됨

설명서 개정 이력

설명서 개정 기호는 앞표지와 뒷표지에 기재된 설명서 번호 끝에 표기하였습니다.



개정 기호	개정 연월	개정 이유 및 개정 페이지
A	2018년 4월	• 설명서 번호 변경 • NX102-□□□□ 추가 • CPU 유니트 Ver.1.30 이상의 지원 기능에 대한 기술을 추가 기재함 • 이벤트 코드 및 이상에 대한 설명의 "NJ/NX 시리즈 트러블 슈팅 설명서"로의 이동에 따른 개정

1

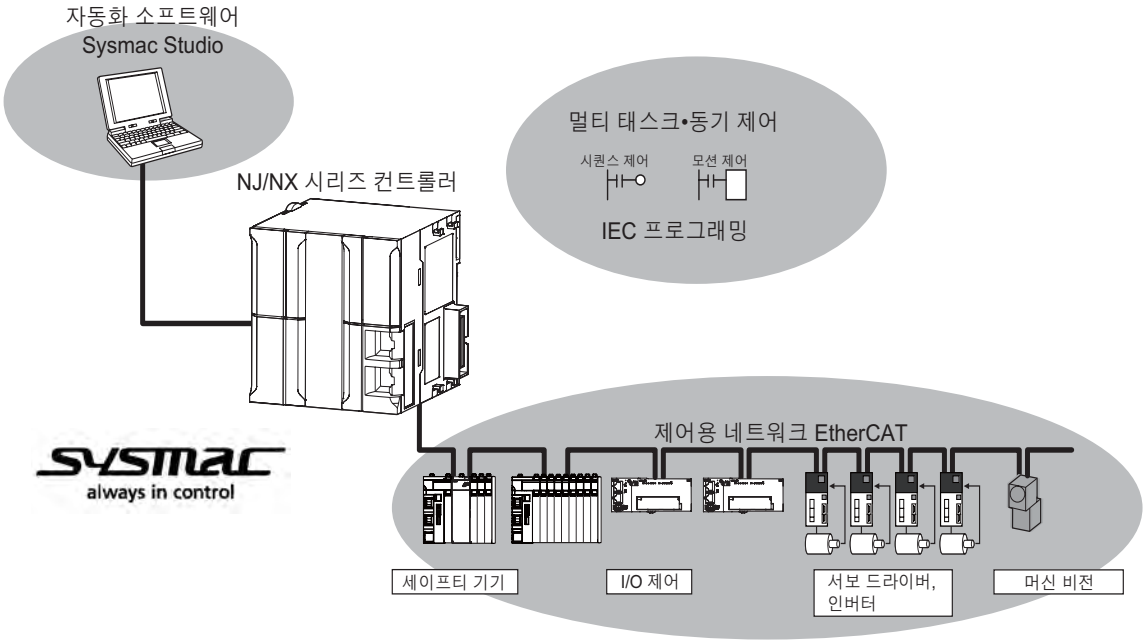
NX 시리즈 컨트롤러의 개요

이 장에서는 NJ/NX 시리즈 컨트롤러의 특징, 시스템 구성의 개요, 사양, 사용 순서에 대해 설명합니다.

1-1 NJ/NX 시리즈 컨트롤러의 개념 1-2
1-1-1 특징 1-2
1-1-2 시스템 구성의 개요 1-5
1-2 사양의 개요 1-13
1-3 NJ/NX 시리즈의 사용 순서 전체 1-16
1-3-1 순서의 개요 1-16
1-3-2 자세한 순서 1-17

1-1 NJ/NX 시리즈 컨트롤러의 개념

머신 자동화 컨트롤러 NJ/NX 시리즈는 기계 제어에 필요한 기능과 고속 성능과 산업용 컨트롤러로서의 안전성·신뢰성·보수성을 양립시킨 차세대 컨트롤러입니다.
기존 프로그래머블 컨트롤러의 기능을 포함하여 모션 제어에 필요한 각종 기능을 부가한 통합형 컨트롤러로 세이프티 기기, 시각 장치, 모션 기기, I/O 등의 입출력 기기를 고속인 EtherCAT상에서 동기하여 제어할 수 있습니다.
OMRON에서는 통일된 통신 사양이나 사용자 인터페이스 사양으로 설계된 OMRON 제어 기기를 Sysmac 장치로 제공합니다. 머신 자동화 컨트롤러 NJ/NX 시리즈는 EtherCAT 슬레이브 등의 Sysmac 장치 및 자동화 소프트웨어 Sysmac Studio와의 조합 시 최적의 기능과 조작성을 구현할 수 있도록 설계되었습니다.
Sysmac 장치에 의해 구성된 시스템에서는 기기를 서로 간단하게 연결하여 통일된 개념과 사용성으로 조작할 수 있습니다.



1-1-1 특징

하드웨어 상의 특징

- **제어용 네트워크 EtherCAT을 표준 장비**

  CPU 유니트는 EtherCAT 통신 마스터 기능 포트를 표준 장비했습니다.
  EtherCAT은 Ethernet 시스템을 기초로 하여 더욱 고속, 고효율 통신이 가능한 고성능 산업용 네트워크 시스템입니다. 각 노드가 Ethernet 프레임을 고속으로 전송하므로 짧은 통신 사이클 시간을 구현할 수 있습니다.
  제어용 네트워크 EtherCAT을 표준 채택하면 머신 제어에 필요한 I/O 시스템, 서보 드라이버, 인버터, 머신 비전 등의 기기를 하나의 네트워크로 연결할 수 있습니다.

- **EtherCAT 슬레이브 터미널을 지원**

  EtherCAT 슬레이브 터미널을 사용하면 공간을 절약할 수 있습니다. 또한, 다양한 종류의 NX 유니트로 플렉시블한 시스템을 구축할 수 있습니다.

● 세이프티 서브 시스템을 실현

NX 시리즈 세이프티 컨트롤 유니트를 사용하면 시퀀스 컨트롤, 모션 컨트롤 시스템에 세이프티 컨트롤을 통합할 수 있습니다.

버전 관련 정보

NX 시리즈 세이프티 컨트롤 유니트는 CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

● NX 유니트를 지원(NX102, NX1P2만 해당)

NX102 및 NX1P2 CPU 유니트에서는 EtherCAT 네트워크의 각종 슬레이브 이외에도 CPU 유니트 상에 각종 NX 유니트(디지털 I/O 유니트, 아날로그 I/O 유니트 등)를 장착해서 사용할 수 있습니다.

사용상의 주의 사항

- NX102 CPU 유니트에서는 NX 시리즈 세이프티 컨트롤 유니트는 NX-SL3□□□만 CPU 유니트 상에 장착해서 사용할 수 있습니다.
- NX1P2 CPU 유니트에서는 NX 시리즈 세이프티 컨트롤 유니트는 CPU 유니트 상에 장착해서 사용할 수 없습니다. NX 시리즈 세이프티 컨트롤 유니트를 사용하는 경우에는 EtherCAT 상의 하위 시스템으로 하십시오.

● CJ 시리즈용 유니트를 지원(NJ□01만 해당)

NJ 시리즈 CPU 유니트에서는 EtherCAT 네트워크의 각종 슬레이브 이외에도 I/O 버스 시스템 상에 각종 CJ 시리즈용 유니트(기본 I/O 유니트, 고기능 유니트)를 장착해서 사용할 수 있습니다.

● EtherNet/IP 통신 기능 포트 표준 장비

CPU 유니트에 EtherNet/IP 통신 기능 포트를 표준 장비했습니다.
EtherNet/IP는 Ethernet을 사용한 산업용 멀티 벤더 네트워크입니다. 컨트롤러 간의 네트워크나 필드 네트워크로 사용할 수 있습니다. 또한, 표준 Ethernet 기술이 사용되므로 다양한 범용 Ethernet 기기와도 혼재하여 사용할 수 있습니다.

● USB 포트를 표준 장비(NX701 / NJ□01만 해당)

PC에 설치한 지원 소프트웨어에서 USB로 CPU 유니트에 직접 연결할 수 있습니다.

사용상의 주의 사항

NX102 및 NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.

● SD 메모리 카드 슬롯 표준 장비

사용자 프로그램에서 CPU 유니트에 장착한 SD 메모리 카드에 액세스할 수 있습니다.

● 높은 하드웨어 신뢰성

PLC 수준의 하드웨어 신뢰성과 RAS 기능이 있습니다.

● 멀티 코어 프로세서를 사용한 병렬 실행(NX701/NX102/NX1P2)

CPU 유니트 NX701-□□□□는 멀티 코어 프로세서를 탑재하고 있으므로 여러 태스크를 병렬로 실행할 수 있습니다. 이것으로 대용량 데이터 통신의 성능 향상을 포함하여 대규모 장치도 고속으로 제어할 수 있습니다.

CPU 유니트 NX102-□□□□ 및 NX1P2-□□□□□□는 멀티 코어 프로세서를 탑재하고 있으므로 태스크의 실행과 태그 데이터 링크 서비스나 시스템 서비스를 병렬로 실행할 수 있습니다. 이를 통해 주변 툴 및 표시기 등과의 통신 성능을 확보할 수 있습니다.

소프트웨어 상의 특징

● 시퀀스 제어와 모션 제어를 통합

1대의 CPU 유니트에서 시퀀스 제어와 모션 제어를 둘 다 커버합니다. 이것으로 시퀀스 제어와 다축 동기 제어를 동시에 실현합니다. 동일한 제어 주기 중에 시퀀스 제어·모션 제어와 I/O 리프레시가 실행됩니다.
또한, 제어 주기는 EtherCAT의 프로세스 데이터 통신 주기와 일치합니다. 따라서 정주기로 편차가 적은 고정밀도의 시퀀스 제어 및 모션 제어를 실현할 수 있습니다.

● 멀티태스크 대응

복수의 태스크에 I/O 리프레시 및 사용자 프로그램 실행 등의 일련의 처리를 할당해서 각각에 실행 조건과 실행 순서를 지정하고, 그것을 조합하여 어플리케이션에 맞는 유연한 제어를 구축할 수 있습니다.

● 국제 표준 규격 IEC 61131-3을 준수하는 프로그래밍 언어 사양

IEC 61131-3을 준수한 언어 사양을 탑재하고 있습니다. 일부는 OMRON이 독자적으로 개량했습니다.
PLCopen®을 준수한 모션 제어 명령을 비롯하여 IEC 규정에 따른 풍부한 명령군(POU)이 있습니다.

● 메모리 맵을 의식하지 않는 변수를 사용한 프로그래밍

PC 상에서 고급 언어를 사용할 때와 같이 모든 데이터는 변수에 의해 액세스합니다. 작성한 변수는 CPU 유니트의 메모리 상에 자동으로 할당되므로 사용자가 그 사실을 의식할 필요가 없습니다.

● 풍부한 보안 기능

조작 권한 설정, ID에 따른 사용자 프로그램 실행 제한 등 풍부한 보안 기능이 있습니다.

● 컨트롤러 전체의 감시 기능

CPU 유니트는 장착된 유니트, EtherCAT 슬레이브를 포함한 컨트롤러 전체의 이벤트를 감시합니다. 이상 시의 대처 방법은 Sysmac Studio 또는 NS 시리즈 표시기에서 가이드합니다. 발생한 이벤트는 이력으로 기억됩니다.

● 자동화 소프트웨어 Sysmac Studio

Sysmac Studio는 컨트롤러뿐만 아니라 EtherCAT 상의 기기까지 하나의 소프트웨어로 해결하는 통합 개발 환경입니다. 기기의 차이에 관계없이 통일된 조작성을 제공합니다. 설계에서 디버그, 시뮬레이션, 셋업, 운용 시작 후의 변경까지 모든 공정을 지원합니다.

● 풍부한 시뮬레이션 기능

가상 컨트롤러의 실행 기능, 디버그 기능, 태스크 실행 시간 검증 기능 등 풍부한 시뮬레이션 기능이 있습니다.

1-1-2 시스템 구성의 개요

여기에서는 NX 시리즈와 NJ 시리즈의 시스템 구성 개요를 설명합니다.

NX701의 시스템 구성 개요

NX701의 시스템 구성은 다음과 같습니다.

● 기본 구성

NX701에서는 EtherCAT 네트워크 구성과 지원 소프트웨어를 기본 구성으로 합니다.

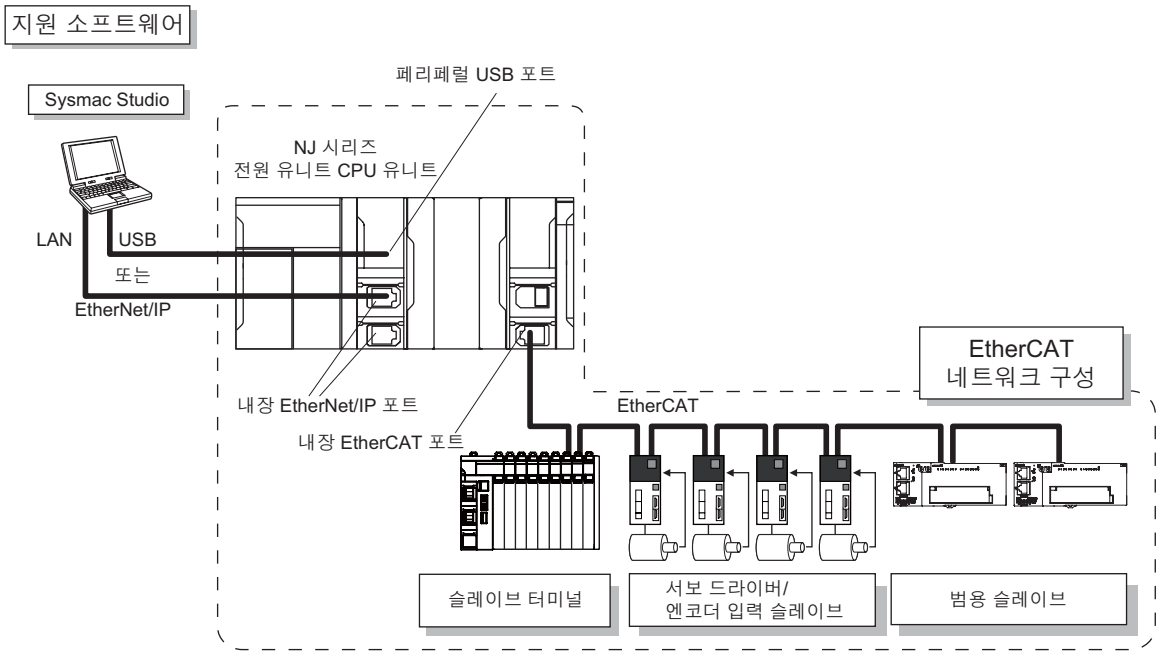
- EtherCAT 네트워크 구성

  내장 EtherCAT 포트를 사용해서 EtherCAT 슬레이브 터미널, 디지털I/O, 아날로그I/O 등의 범용 슬레이브 및 서보/엔코더 입력 슬레이브를 연결할 수 있습니다. 이 구성을 사용하면 정주기이면서 편차가 적은 고정밀도 시퀀스 제어 및 모션 제어를 실현할 수 있습니다.

- 지원 소프트웨어

  CPU 유니트의 페리페럴 USB 포트에 시판 USB 케이블로 연결합니다. 또는 내장 EtherNet/IP 포트에 Ethernet 케이블로 연결합니다.

  지원 소프트웨어 연결 구성의 자세한 내용은 '10-2 Sysmac Studio 연결(10-7페이지)'을 참조하십시오.



사용상의 주의 사항

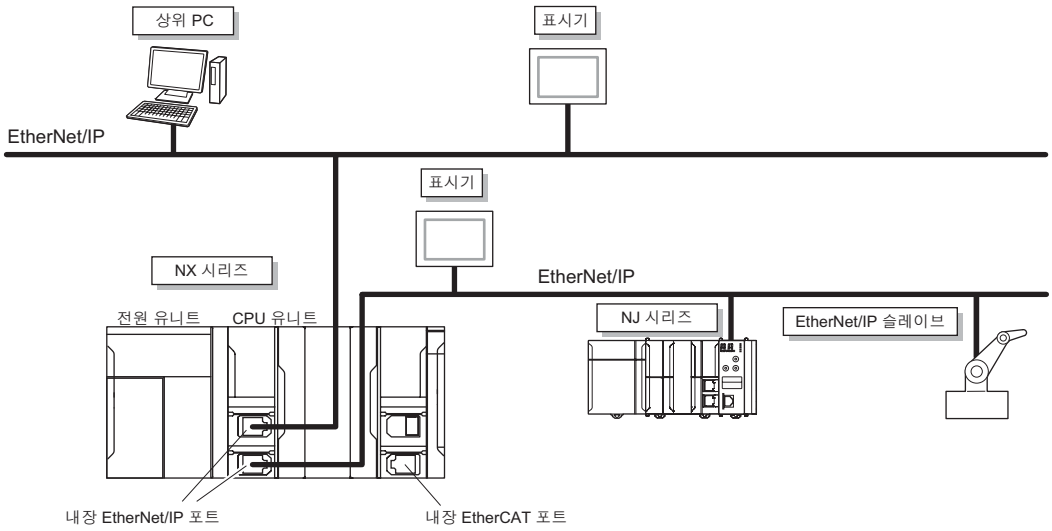
NX 유니트는 슬레이브 터미널에 연결하여 사용합니다 CPU 유니트의 NX 버스 커넥터는 향후 확장용이므로 NX 유니트를 연결하여 사용할 수 없습니다.

참고

통신 커플러 유니트에 Sysmac Studio를 직접 연결하여 슬레이브 터미널의 설정을 실시할 수도 있습니다. 자세한 내용은 "NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

● 네트워크 구성

- 내장 EtherNet/IP 포트에 상위 PC, 표시기, NJ/NX 시리즈를 연결합니다.
- NX701에서는 EtherNet/IP 포트가 2개 있습니다.



네트워크 구성의 자세한 내용은 🕮'제10장 통신 설정(10-1페이지)'을 참조하십시오.

● 지원 소프트웨어

NX701의 설정/모니터링/디버그를 실행하는 주변 툴은 다음과 같습니다.

- Sysmac Studio

  NX701에서는 컨트롤러 구성·설정, 프로그램 작성, 디버그, 시뮬레이션을 실행하는 지원 소프트웨어로 Sysmac Studio를 기본으로 사용합니다.

- 기타 지원 소프트웨어

  Sysmac Studio 이외에도 소프트웨어 패키지 Sysmac Studio 스탠더드 에디션에는 다음 소프트웨어가 포함됩니다.

구성 소프트웨어	용도
Sysmac Studio	시퀀스 제어 및 모션 제어를 포함하는 다음 이외의 모든 기능일 때 사용합니다.
Network Configurator	EtherNet/IP 포트에서 태그 데이터 링크할 때 사용합니다.*1

*1. 대상 장치가 NJ/NX 시리즈 컨트롤러인 경우에는 Sysmac Studio Ver.1.10 이상에서도 설정할 수 있습니다. CS/CJ 시리즈 PLC가 오리지네이터 장치인 경우는 Network Configurator에서 설정합니다.

NX102의 시스템 구성 개요

NX102의 시스템 구성은 다음과 같습니다.

● 기본 구성

NX102에서는 EtherCAT 네트워크 구성, NX 유니트 구성 및 지원 소프트웨어를 기본 구성으로 합니다.

- EtherCAT 네트워크 구성

  내장 EtherCAT 포트를 사용해서 EtherCAT 슬레이브 터미널, 디지털I/O, 아날로그I/O 등의 범용 슬레이브 및 서보/엔코더 입력 슬레이브를 연결할 수 있습니다. 이 구성을 사용하면 정주기이면서 편차가 적은 고정밀도 시퀀스 제어 및 모션 제어를 실현할 수 있습니다.

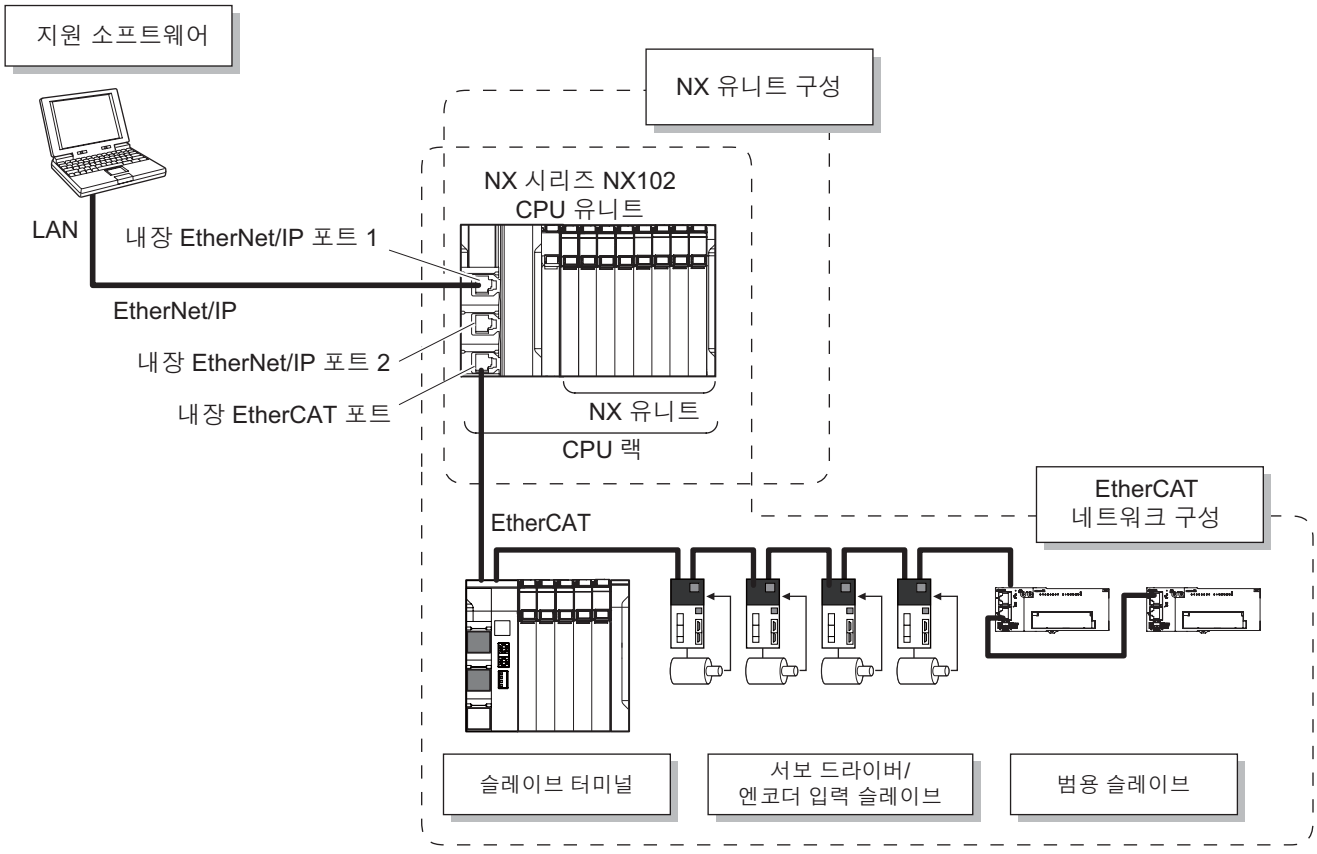
- NX 유니트 구성

  EtherCAT 네트워크 이외에도 NX 유니트(디지털 I/O 유니트, 아날로그 I/O 유니트 등)를 장착할 수 있습니다.

- 지원 소프트웨어

  내장 EtherNet/IP 포트에 Ethernet 케이블로 연결합니다.

  지원 소프트웨어 연결 구성의 자세한 내용은 '10-2 Sysmac Studio 연결(10-7페이지)'을 참조하십시오.



사용상의 주의 사항

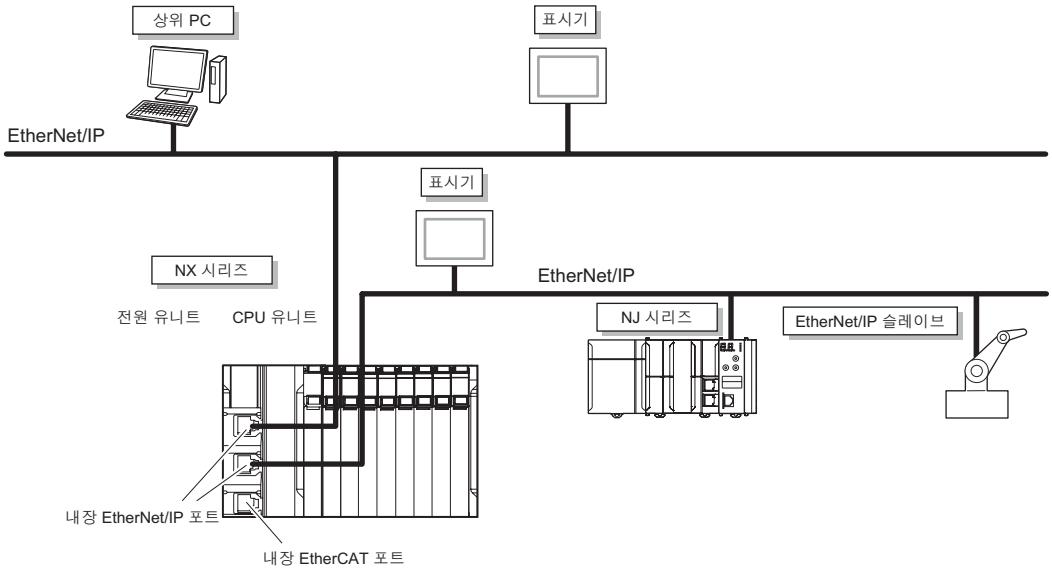
- NX102 CPU 유니트는 USB 포트를 장비하지 않았습니다.

참고

통신 커플러 유니트에 Sysmac Studio를 직접 연결하여 슬레이브 터미널의 설정을 실시할 수도 있습니다. 자세한 내용은 "NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

● 네트워크 구성

- 내장 EtherNet/IP 포트에 상위 PC, 표시기, NJ/NX 시리즈를 연결합니다.
- NX102에서는 EtherNet/IP 포트가 2개 있습니다.



네트워크 구성의 자세한 내용은 '제10장 통신 설정(10-1페이지)'을 참조하십시오.

● 지원 소프트웨어

NX102의 설정/모니터링/디버그를 실행하는 주변 툴은 다음과 같습니다.

- Sysmac Studio

  NX102에서는 컨트롤러 구성·설정, 프로그램 작성, 디버그, 시뮬레이션을 실행하는 지원 소프트웨어로 Sysmac Studio를 기본으로 사용합니다.

- 기타 지원 소프트웨어

  Sysmac Studio 이외에도 소프트웨어 패키지 Sysmac Studio 스탠더드 에디션에는 다음 소프트웨어가 포함됩니다.

구성 소프트웨어	용도
Sysmac Studio	시퀀스 제어 및 모션 제어를 포함하는 다음 이외의 모든 기능일 때 사용합니다.
Network Configurator	EtherNet/IP 포트에서 태그 데이터 링크할 때 사용합니다.*1

*1. 대상 장치가 NJ/NX 시리즈 컨트롤러인 경우에는 Sysmac Studio Ver.1.10 이상에서도 설정할 수 있습니다. CS/CJ 시리즈 PLC가 오리지네이터 장치인 경우는 Network Configurator에서 설정합니다.

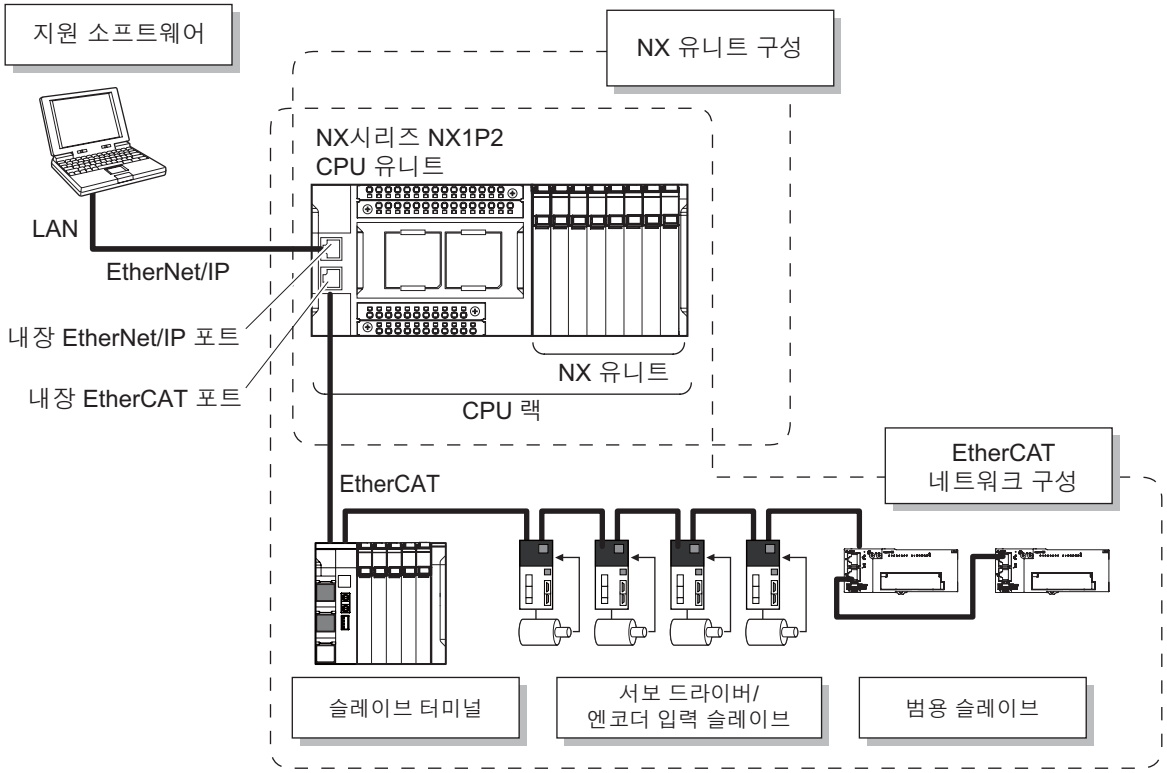
NX1P2의 시스템 구성 개요

NX1P2의 시스템 구성은 다음과 같습니다.

● 기본 구성

NX1P2에서는 EtherCAT 네트워크 구성, NX 유니트 구성 및 지원 소프트웨어를 기본 구성으로 합니다.

- EtherCAT 네트워크 구성
  내장 EtherCAT 포트를 사용해서 EtherCAT 슬레이브 터미널, 디지털I/O, 아날로그I/O 등의 범용 슬레이브 및 서보/엔코더 입력 슬레이브를 연결할 수 있습니다. 이 구성을 사용하면 정주기이면서 편차가 적은 고정밀도 시퀀스 제어 및 모션 제어를 실현할 수 있습니다.
- NX 유니트 구성
  EtherCAT 네트워크 이외에도 NX 유니트(디지털 I/O 유니트, 아날로그 I/O 유니트 등)를 장착할 수 있습니다.
- 지원 소프트웨어
  내장 EtherNet/IP 포트에 Ethernet 케이블로 연결합니다.
  지원 소프트웨어 연결 구성의 자세한 내용은 '10-2 Sysmac Studio 연결(10-7페이지)'을 참조하십시오.



사용상의 주의 사항

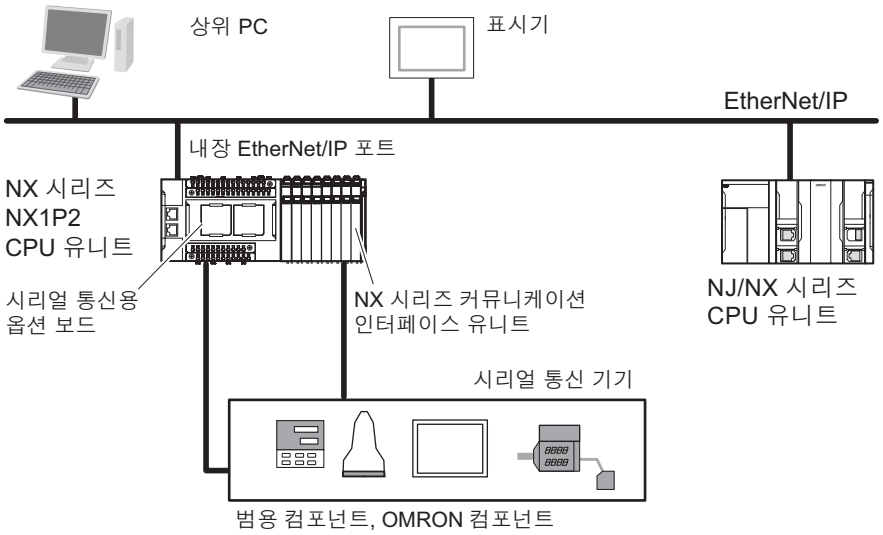
- NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.

참고

통신 커플러 유니트에 Sysmac Studio를 직접 연결하여 슬레이브 터미널의 설정을 실시할 수도 있습니다. 자세한 내용은 "NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

● 네트워크 구성

• 내장 EtherNet/IP 포트에 상위 PC, 표시기, NJ/NX 시리즈를 연결합니다.



네트워크 구성의 자세한 내용은 🕮'제10장 통신 설정(10-1페이지)'을 참조하십시오.

● 지원 소프트웨어

NX1P2의 설정/모니터링/디버그를 실행하는 주변 툴은 다음과 같습니다.

• Sysmac Studio

NX1P2에서는 컨트롤러 구성·설정, 프로그램 작성, 디버그, 시뮬레이션을 실행하는 지원 소프트웨어로 Sysmac Studio를 기본으로 사용합니다.

• 기타 지원 소프트웨어

Sysmac Studio 이외에도 소프트웨어 패키지 Sysmac Studio 스탠더드 에디션에는 다음 소프트웨어가 포함됩니다.

구성 소프트웨어	용도
Sysmac Studio	시퀀스 제어 및 모션 제어를 포함하는 다음 이외의 모든 기능일 때 사용합니다.
Network Configurator	EtherNet/IP 포트에서 태그 데이터 링크할 때 사용합니다.*1

*1. 대상 장치가 NJ/NX 시리즈 컨트롤러인 경우에는 Sysmac Studio Ver.1.10 이상에서도 설정할 수 있습니다. CS/CJ 시리즈 PLC가 오리지네이터 장치인 경우는 Network Configurator에서 설정합니다.

NJ 시리즈 시스템 구성의 개요

NJ 시리즈의 시스템 구성은 다음과 같습니다.

● 기본 구성

NJ 시리즈에서는 EtherCAT 네트워크 구성, CJ 유니트 구성 및 지원 소프트웨어를 기본 구성으로 합니다.

- EtherCAT 네트워크 구성

  내장 EtherCAT 포트를 사용해서 EtherCAT 슬레이브 터미널, 디지털I/O, 아날로그I/O 등의 범용 슬레이브 및 서보/엔코더 입력 슬레이브를 연결할 수 있습니다. 이 구성을 사용하면 정주기이면서 편차가 적은 고정밀도 시퀀스 제어 및 모션 제어를 실현할 수 있습니다.

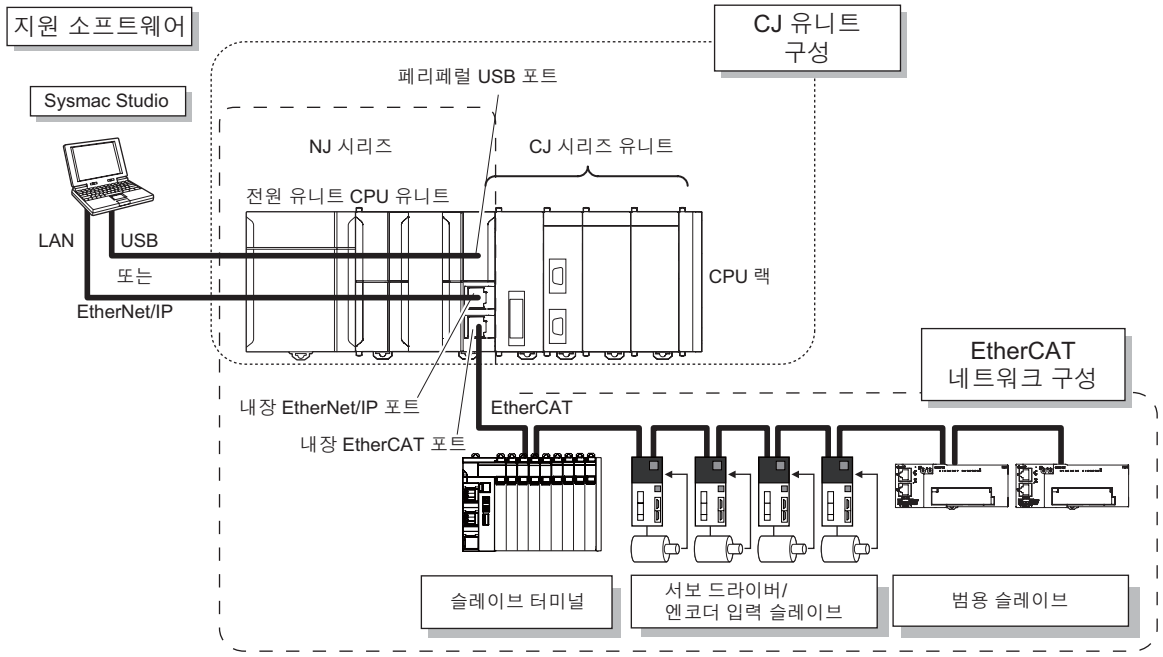
- CJ 유니트 구성

  EtherCAT 네트워크 이외에도 CJ 시리즈 유니트(기본 I/O 유니트, 고기능 유니트)를 장착할 수 있습니다. CPU 유니트가 장착된 CPU 랙 이외에도 증설 랙을 사용하여 CJ 시리즈 유니트를 증설할 수 있습니다.

- 지원 소프트웨어

  CPU 유니트의 페리페럴 USB 포트에 시판 USB 케이블로 연결합니다. 또는 내장 EtherNet/IP 포트에 Ethernet 케이블로 연결합니다.

  지원 소프트웨어 연결 구성의 자세한 내용은 '10-2 Sysmac Studio 연결(10-7페이지)'을 참조하십시오.

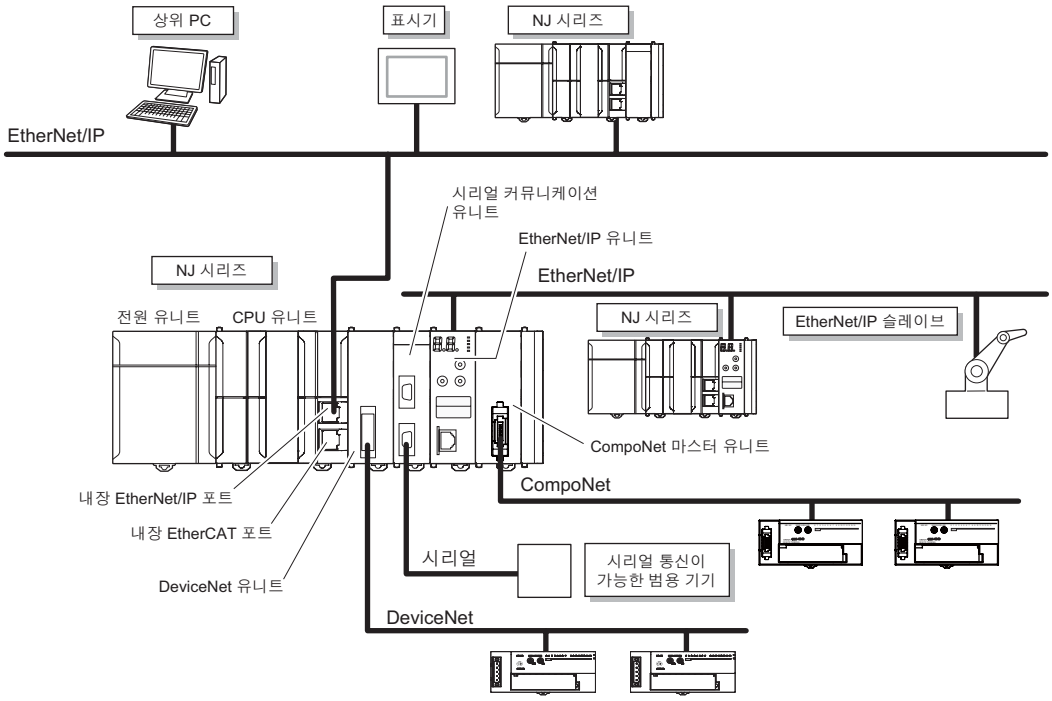


참고

통신 커플러 유니트에 Sysmac Studio를 직접 연결하여 슬레이브 터미널의 설정을 실시할 수도 있습니다. 자세한 내용은 "NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

● 네트워크 구성

- 내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트(CJ1W-EIP21)에 상위 PC, 표시기, NJ 시리즈를 연결합니다.
- DeviceNet은 DeviceNet 유니트에, CompoNet은 CompoNet 유니트에, 시리얼 통신을 사용하는 경우에는 시리얼 커뮤니케이션 유니트에 연결합니다.



네트워크 구성의 자세한 내용은 '제10장 통신 설정(10-1페이지)'을 참조하십시오.

● 지원 소프트웨어

NJ 시리즈의 설정/모니터링/디버그를 실행하는 주변 툴은 다음과 같습니다.

- Sysmac Studio

  NJ 시리즈에서는 컨트롤러 구성·설정, 프로그램 작성, 디버그, 시뮬레이션을 실행하는 지원 소프트웨어로 Sysmac Studio를 기본으로 사용합니다.

- 기타 지원 소프트웨어

  Sysmac Studio 이외에도 소프트웨어 패키지 Sysmac Studio 스탠더드 에디션에는 다음 소프트웨어가 포함됩니다.

구성 소프트웨어	용도
Sysmac Studio	시퀀스 제어 및 모션 제어를 포함하는 다음 이외의 모든 기능일 때 사용합니다.
Network Configurator	EtherNet/IP 포트 및 유니트에서 태그 데이터 링크할 때 사용합니다.*1
CX-Integrator	DeviceNet 유니트 및 CompoNet 마스터 유니트에서 리모트 I/O 통신을 실시하는 경우에 사용합니다.
CX-Protocol	시리얼 커뮤니케이션 유니트에서 프로토콜 매크로 기능을 사용할 때 사용합니다.
CX-Designer	NS 시리즈 표시기의 화면을 작성할 때 사용합니다.

*1. 대상 장치가 NJ/NX 시리즈 컨트롤러인 경우에는 Sysmac Studio Ver.1.10 이상에서도 설정할 수 있습니다. CS/CJ 시리즈 PLC가 오리지네이터 장치인 경우는 Network Configurator에서 설정합니다.

1-2 사양의 개요

여기에서는 NJ/NX 시리즈의 주요 성능 사양에 대해 설명합니다. 일반 사양, 성능 사양, 기능 사양은 'A-1 사양 (A-3페이지)'을 참조하십시오.

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
프로그래밍	프로그램 용량[*1]	크기		80MB	5MB	1.5MB	20MB	5MB	3MB
		수량	POU 정의 수	6,000	3,000	450	3,000	750	450
			POU 인스턴스 수	48,000	9,000	1,800	9,000(*)	3,000(*)	1,800
	변수 용량	유지 속성 있음[*2]	크기	4MB	1.5MB	32KB[*3]	2MB	0.5MB	
			변수의 수	40,000	10,000	5,000	10,000	5,000(*)	
		유지 속성 없음[*4]	크기	256MB	32MB	2MB	4MB	2MB	
			변수의 수	360,000	90,000	90,000	90,000	22,500	
	데이터 형식	데이터 형식의 수		8,000	1,000	1,000	2,000	1,000	
	CJ 유니트용 메모리(변수의 AT 지정에 의해 지정 가능)	채널 I/O(CIO)		–	6144채널(0~6143)[*5]		6144채널(0~6143)		
		워크 릴레이(WR)		–	512채널(W0~W511)[*5]		512채널(W0~W511)		
		유지 릴레이(HR)		–	1536채널(H0~H1535)[*6]		1536채널(H0~H1535)		
		데이터 메모리(DM)		–	32768채널(D0~D32767)	16000채널(D0~D15999)	32768채널(D0~D32767)		
		확장 데이터 메모리(EM)		–	32768채널 x25뱅크 (E0_00000~E18_32767)[*6 *7]	–	32768채널 x25뱅크 (E0_00000 ~ E18_32767)	32768채널x4뱅크 (E0_00000 ~ E3_32767)	
모션 제어				'CPU 유니트별 모션 제어의 성능 사양(A-10페이지)'을 참조하십시오.					
페리페럴(USB) 포트	지원 서비스			Sysmac Studio 연결	–	–	Sysmac Studio 연결		
	물리층			USB2.0 준수 B 커넥터	–	–	USB2.0 준수 B 커넥터		
	전송 거리			최대 5m	–	–	최대 5m		
내장 EtherNet/IP 포트	포트 수			2	2	1			
	물리층			10BASE-T/100BASE-TX/1000BASE-T	10BASE-T/100BASE-TX				
	프레임 길이			최대 1,514바이트					
	매체 액세스 방식			CSMA/CD					
	변조 방식			베이스밴드					
	토폴로지			스타형					
	전송 속도			1Gbps(1000BASE-T)	100Mbps(100BASE-TX)				
	전송 매체			트위스트 페어 케이블(실드 가공: STP): 카테고리 5, 5e 이상					
	전송 거리(허브와 노드 사이의 거리) 최대값			100m					
	캐스케이드 연결 최대 수			스위칭 허브 사용에 제한 없음					

항목			NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
내장 EtherNet/IP 포트	CIP 서비스: 태그 데이터 링크(사이클릭 통신)	최대 연결 수	256/포트 합계 512	32/포트 합계 64	32			
		패킷 간격*8	연결마다 설정 가능 0.5~10000ms(0.5ms 단위)	연결마다 설정 가능 1~10000ms(1ms 단위)	연결마다 설정 가능 2~10000ms(1ms 단위)	연결마다 설정 가능 1~10000ms(1ms 단위)(*)		
		유니트 허용 통신 대역	40,000pps*9 (하트비트 포함)	12,000pps*9(하트비트 포함)	3,000pps*9 (하트비트 포함)(*)			
		최대 태그 세트 수	256/포트 합계 512	32/포트 합계 40*10	32			
		태그 종류	네트워크 변수	네트워크 변수, CIO/WR/HR/DM		네트워크 변수, CIO/WR/HR/DM/EM		
		1연결(=1태그 세트)당 태그 수	8(태그 세트에 컨트롤러 Status가 포함되는 경우는 7)					
		최대 태그 수	256/포트 합계 512		256			
		1노드당 최대 링크 데이터 크기(전체 태그의 합계 크기)	369,664바이트	19,200바이트				
		1연결당 최대 데이터 크기	1,444바이트	600바이트				
		등록 가능한 최대 태그 세트 수	256/포트 합계 512 (1연결=1태그 세트)	32/포트 합계 40*10 (1연결=1태그 세트)	32(1연결=1태그 세트)			
		1태그 세트의 최대 크기	1,444바이트 (태그 세트에 컨트롤러 Status가 포함되는 경우는 2바이트 분량 사용)	600바이트(태그 세트에 컨트롤러 Status가 포함되는 경우는 2바이트 분량 사용)				
		멀티 캐스트 패킷 필터 기능*11	가능					
내장 EtherCAT 포트	통신 규격		IEC 61158 Type12					
	EtherCAT 마스터 사양		Class B 지원(Feature Pack Motion Control 지원)					
	물리층		100BASE-TX					
	변조 방식		베이스밴드					
	전송 속도		100Mbps(100BASE-TX)					
	Duplex 모드		Auto					
	토폴로지		라인, 데이지 체인, 분기 배선					
	전송 매체		카테고리 5 이상, 트위스트 페어 케이블(알루미늄 테이프와 편조 이중 차폐 실드 케이블, 스트레이트 권장)					
	노드 간 거리 최대값		100m					
	최대 슬레이브 수		512	64	16	192		64

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
시리얼 통신	통신 방식			–		반이중(시리얼 통신용 옵션 보드 연결 시)	–		
	동기 방식			–		비동기(시리얼 통신용 옵션 보드 연결 시)	–		
	전송 속도			–		1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2kbps(시리얼 통신용 옵션 보드 연결 시)	–		
유니트 구성	장착할 수 있는 유니트 수	1랙(CPU 랙 또는 증설 랙)에 장착할 수 있는 CJ 유니트의 최대 수		–			10		
		1랙(CPU 랙)당 최대 NX 유니트 수		–	32	8	–		
		시스템 전체의 최대 CJ 유니트 수		–			40		
		시스템 전체의 최대 NX 유니트 수		4,096 (EtherCAT 슬레이브 터미널 상)	400 (EtherCAT 슬레이브 터미널 상)	24 (CPU 랙 상: 8, EtherCAT 슬레이브 터미널 상: 16)	4,096 (EtherCAT 슬레이브 터미널 상)		400 (EtherCAT 슬레이브 터미널 상)
	증설 랙 최대 수			0			3		
	입출력 점수	CJ 유니트 최대 입출력 점수		–			2,560		
	전원(CPU 랙 또는 증설 랙)	형식		NX-PA9001 NX-PD7001	DC 입력의 비절연 전원을 CPU 유니트에 내장		NJ-P□3001		
		전원 OFF 확정 시간	AC 전원	30~45ms	–	–	30~45ms		
			DC 전원	5~20ms	2~8ms	2~8ms	22~25ms		
	NX 버스의 I/O 데이터의 최대 크기			–	IN:8,192바이트 OUT:8,192바이트		–		

*1. 실행 개체, 변수 테이블(변수명 등)의 용량입니다.
*2. CJ 유니트용 메모리의 HR, DM, EM은 포함되지 않습니다.
*3. CJ 유니트용 메모리가 포함됩니다.
*4. CJ 유니트용 메모리의 CIO, WR은 포함되지 않습니다.
*5. 유지 속성이 없는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*6. 유지 속성이 있는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*7. NX102 CPU 유니트에서는 유지 변수 메모리의 용량 상한이 1.5MB이므로 최대 채널에서 전체 뱅크를 동시에 생성할 수 없습니다.
*8. 노드 수에 좌우되지 않고 설정된 주기에 따라 회선상의 데이터를 갱신합니다.
*9. pps는 Packet Per Second를 의미하며, 1초간 처리 가능한 송수신 패킷 수를 나타냅니다.
*10. 합계 수가 40을 초과하는 태그 세트가 설정된 경우에는 '태그 데이터 링크 태그 세트 설정 수 초과'(840E0000Hex) 이벤트가 발생합니다.
*11. EtherNet/IP 포트에는 IGMP 클라이언트가 장착되어 있으므로 IGMP Snooping 호환 스위칭 허브를 사용하여 불필요한 멀티 캐스트 패킷을 필터링할 수 있습니다.

(주의) 표에 (*)가 표시된 항목은 버전 업그레이드로 성능 개선된 기능입니다. 버전 업그레이드에 대한 정보는 'A-15 NX 시리즈의 버전 관련 정보(A-222페이지)' 및 'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

1-3 NJ/NX 시리즈의 사용 순서 전체

여기에서는 NJ/NX 시리즈의 전체적인 사용 순서의 개요와 자세한 내용에 대해 설명합니다.

1-3-1 순서의 개요

NJ/NX 시리즈는 다음과 같은 흐름으로 사용합니다.

STEP 1. 소프트웨어 설계

시스템의 전체 구성 및 태스크 구성/프로그램/변수를 설계합니다.

STEP 1-1 I/O 및 처리의 설계
STEP 1-2 태스크 설계
STEP 1-3 프로그램 설계

STEP 2. 소프트웨어 설정 및 프로그래밍

STEP 1에서 설계한 시스템 구성을 지원 소프트웨어에서 작성하는 동시에 변수의 할당을 실시합니다. 또한, 태스크 및 프로그램의 작성을 실시하고 시뮬레이션 등의 디버그를 실시합니다.

STEP 2-1 슬레이브/유니트 구성
STEP 2-2 컨트롤러 설정
STEP 2-3 프로그래밍
STEP 2-4 오프라인 디버그

STEP 3. 장착 및 하드웨어 설정

각종 유니트를 장착하고 필요에 따라 하드웨어 설정을 실시합니다.

STEP 4. 배선

네트워크 케이블이나 I/O의 배선을 실시합니다.

STEP 5. 실제 기기에 의한 동작 확인과 운전

지원 소프트웨어와 실제 시스템에 연결하여 프로젝트를 다운로드합니다. 실제 기기에서 동작을 확인한 후에 운전을 시작합니다.

1-3-2 자세한 순서

STEP 1. 소프트웨어 설계

순서	내용	참조
STEP 1-1 **I/O와 처리 설계**	• 외부 I/O와 유니트 구성 • 외부 기기와 리프레시 주기 • 프로그램 처리 내용	🕮 "NX 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-418)" 🕮 NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)" 🕮 "NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)" 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-358)"
↓		
STEP 1-2 **태스크 설계**	• 태스크 구성 • 태스크-프로그램 관계 • 각 태스크의 주기 • 슬레이브/유니트의 리프레시 주기 • 태스크 간의 변수 배타 방법	🕮 '4-2-3 태스크 설정(4-7페이지)'
↓		
STEP 1-3 **프로그램 설계**		
POU(처리의 분할 단위) 설계	• 프로그램 • 펑션/펑션 블록 • 각 알고리즘의 언어 선택	🕮 '제6장 프로그래밍(6-1페이지)'
변수 설계	• POU 공통으로 사용하는 변수와 각 POU에서만 사용하는 변수를 분류 • 슬레이브/유니트에 액세스하기 위한 디바이스 변수의 변수명 설계 • 변수의 이름, 유지 등의 각 속성 설계 • 변수의 데이터 형식 설계	🕮 '6-3 변수(6-28페이지)'
↓		

STEP 2. 소프트웨어 설정 및 프로그래밍

순서	내용	Sysmac Studio에서의 조작	참조
프로젝트 새로 작성	1. Sysmac Studio에서 프로젝트를 새로 작성합니다. 2. 컨트롤러를 삽입합니다.	[새 프로젝트] 버튼 [삽입]–[컨트롤러]	🕮"Sysmac Studio Version 1 조작 설명서(SBCA-470)"
↓			

다음 '컨트롤러 구성·설정' 및 '프로그래밍/태스크 설정' 순서는 모두 가능합니다.

STEP 2-1 **슬레이브/유니트 구성**			
1) 슬레이브/유니트 구성 작성	1. 오프라인 또는 온라인에서 슬레이브/유니트 구성을 작성합니다. (온라인인 경우에 STEP 5의 온라인 연결을 실행한 후) 2. 슬레이브 터미널을 사용하는 경우에 슬레이브 터미널 설정을 실시합니다.	EtherCAT 슬레이브 설정 편집기 유니트 편집기	📖'3-2-1 EtherCAT 슬레이브 구성의 작성(3-4페이지)' 📖'3-2-2 유니트 구성의 작성(3-5페이지)' 📖"NX시리즈 EtherCAT 커플러 유니트 사용 설명서 (SBCD-361)"
↓			
2) I/O 포트에 디바이스 변수 할당	디바이스 변수를 변수 테이블에 등록 (변수명은 사용자 정의 또는 자동 생성)	I/O 맵	📖'3-3 I/O 포트와 디바이스 변수(3-7페이지)'
↓			
(모션 제어를 하는 경우)			
3) '축' 작성 및 서보/엔코더 입력 슬레이브에 대한 할당	'축'을 작성하여 실제 축 또는 가상 축에 할당 보간축 제어 시에는 '축 그룹'을 작성	구성·설정–모션 제어 설정에서 설정	📖'3-5 '축'의 작성과 서보/ 엔코더 입력 슬레이브 등의 할당(3-18페이지)'
↓			
STEP 2-2 **컨트롤러 설정**	Sysmac Studio에서 다음 파라메터를 설정합니다.		📖'제4장 컨트롤러 설정(4-1페이지)'
	PLC 기능 모듈 관련 초기 설정	구성·설정–동작 설정에서 설정	📖'4-2 PLC 기능 모듈의 초기 설정(4-4페이지)'
	(NX1P2 CPU 유니트인 경우) • 배터리를 장착하는 경우는 시계 기능에서 시각 데이터 설정 • NX 버스 기능 모듈의 초기 설정 • NX 유니트의 초기 설정 • 내장 I/O 초기 설정 • 옵션 보드 초기 설정 • CJ 유니트용 메모리 설정	각각 다음 설명에 따라 설정 • 컨트롤러–컨트롤러 시계 • 구성·설정–CPU 랙 • 구성·설정–컨트롤러 설정–내장 I/O 설정 • 구성·설정–컨트롤러 설정–옵션 보드 설정 • 구성·설정–컨트롤러 설정–CJ 유니트용 메모리 설정	📖'8-1-2 시계 기능(8-3페이지)' 📖'4-2-4 NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정(4-12페이지)' 📖'4-3 NX 유니트의 초기 설정(4-16페이지)' 📖"NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서 (SBCA-449)"
	(NJ 시리즈인 경우) 고기능 유니트의 초기 설정	구성·설정–CPU·증설랙에서 설정	📖'4-4 고기능 유니트의 초기 설정(4-20페이지)'
	(모션 제어를 하는 경우) 모션 제어 기능 모듈 관련 초기 설정	구성·설정–모션 제어 설정에서 설정	📖'4-5 모션 제어 기능 모듈의 초기 설정(4-22페이지)'
	EtherCAT 마스터 기능 모듈 관련 초기 설정	구성·설정–EtherCAT에서 설정	📖'4-6 EtherCAT 마스터 기능 모듈의 초기 설정(4-24페이지)'
	EtherNet/IP 기능 모듈 관련 초기 설정	구성·설정–내장 EtherNet/IP 포트 설정에서 설정	📖'4-7 EtherNet/IP 기능 모듈의 초기 설정(4-25페이지)'
↓			

STEP 2-3 **프로그래밍**			
1) 변수 등록	• Sysmac Studio에서 글로벌 변수 테이블에 POU 공통 변수 등록 • 각 프로그램의 로컬 변수 테이블 등록 • 각 펑션 블록, 펑션의 로컬 변수 테이블 등록	글로벌 변수 테이블 편집기 로컬 변수 테이블 편집기	🕮"Sysmac Studio Version 1 조작 설명서 (SBCA-470)" 🕮'6-3 변수 (6-28페이지)'
2) POU의 알고리즘 작성	POU(프로그램/펑션 블록/펑션)의 알고리즘을 각 언어로 작성합니다.	프로그래밍 편집기	🕮'제6장 프로그래밍(6-1페이지)' 🕮 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)", 🕮 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"
3) 태스크 설정	태스크 관련 설정을 합니다.	태스크 설정에서 설정	🕮'4-2-3 태스크 설정(4-7페이지)'
STEP 2-4 **오프라인 디버그**	시뮬레이터(가상 컨트롤러)로 알고리즘과 태스크 실행 시간을 확인		🕮'제7장 동작 확인 및 운전 (7-1페이지)'

STEP 3. 장착 및 하드웨어 설정

순서	내용	참조
1) 장착	• 유니트끼리 연결 • DIN 레일에 설치	🕮 "NX 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-418)" 🕮 "NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)" 🕮 "NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)" 🕮"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-358)"
2) 하드웨어 설정	• EtherCAT 슬레이브의 노드 주소 설정 • 고기능 유니트의 전면 로터리 스위치를 통한 호기 번호/유니트 번호 설정	🕮각 EtherCAT 슬레이브의 사용 설명서, 각 고기능 유니트의 사용 설명서

STEP 4. 배선

순서	내용	참조
1) Ethernet 케이블의 배선	• 내장 EtherCAT 포트의 배선 • 내장 EtherNet/IP 포트의 배선	🕮 "NX 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-418)" 🕮 "NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)" 🕮 "NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)" 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-358)"
2) I/O 배선	• EtherCAT 슬레이브의 I/O 배선 • 기본 I/O 유니트, 고기능 유니트의 배선	🕮 각 EtherCAT 슬레이브의 사용 설명서 🕮 "NX 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-418)" 🕮 "NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)" 🕮 "NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)" 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편 (SBCA-358)"
	• 배선 확인	🕮 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"
3) PC(Sysmac Studio)와의 배선	• USB 케이블의 배선*1 • 내장 EtherNet/IP 포트의 배선	🕮 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"

*1. NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.

STEP 5. 실제 기기에 의한 동작 확인 및 운전

순서	내용	Sysmac Studio에서의 조작	참조
1) Sysmac Studio와 온라인 연결, 프로젝트 다운로드	컨트롤러의 전원을 ON으로 하고 Sysmac Studio와 온라인 연결합니다. 그 후, 프로젝트를 다운로드합니다.*1 장착 유니트에서 STEP 2-1의 '슬레이브/유니트 구성 작성'을 하는 경우에는 STEP 2-1 전에 실시하십시오.	[컨트롤러]–[통신 설정] [컨트롤러]–[동기]	🕮'제7장 동작 확인 및 운전 (7-1페이지)'
↓			
2) 실제 기기 동작 확인	1. I/O 맵 또는 감시 창에서 실제 I/O로의 강제값 리프레시 등을 실시하여 배선을 확인합니다. 2. 모션 제어하는 경우는 프로그램 모드의 MC 시운전 화면에서 배선을 확인합니다. 그 다음에 조그 이송을 통해 모터의 방향 확인, 상대 위치 결정을 통해 이동량(전자 기어의 설정) 확인, 원점 복귀 동작을 확인합니다. 3. 운전 모드로 하여 사용자 프로그램의 동작을 확인합니다.	–	🕮'제7장 동작 확인 및 운전 (7-1페이지)'
↓			
3) 컨트롤러의 본 운전	본 운전을 시작합니다.	–	–

*1. 다운로드는 Sysmac Studio의 동기 기능을 사용합니다.

2

CPU 유니트의 동작

CPU 유니트를 사용하는데 필요한 지식으로 CPU 유니트의 구조 및 각 상태에서의 동작을 설명합니다.

2-1 CPU 유니트 동작의 개요 2-2
2-1-1 CPU 유니트 구조의 개요 2-2
2-1-2 CPU 유니트의 각 상태의 동작 개요 2-3
2-2 소프트웨어의 구조 2-4
2-2-1 소프트웨어 구성 2-4
2-2-2 소프트웨어의 동작 2-5
2-3 I/O와의 액세스 구조 2-12
2-3-1 변수의 종류 2-12
2-3-2 변수와 I/O의 액세스 2-15
2-4 NX 버스 기능 모듈의 I/O 리프레시 2-24
2-4-1 I/O 리프레시 방식 2-24
2-4-2 각 I/O 리프레시 방식의 동작 2-25
2-5 시퀀스 제어와 모션 제어의 구조 2-28
2-5-1 제어의 개요 2-28
2-5-2 시퀀스 제어의 구조 2-30
2-5-3 모션 제어의 구조 2-32
2-5-4 시퀀스 제어와 모션 제어의 동기 2-34
2-6 CPU 유니트 내의 데이터 개요 2-35
2-7 CPU 유니트의 각 상태의 동작 2-36
2-7-1 CPU 유니트의 상태 2-36
2-7-2 각 상태의 동작 2-38
2-7-3 동작 모드 2-40

2-1 CPU 유니트 동작의 개요

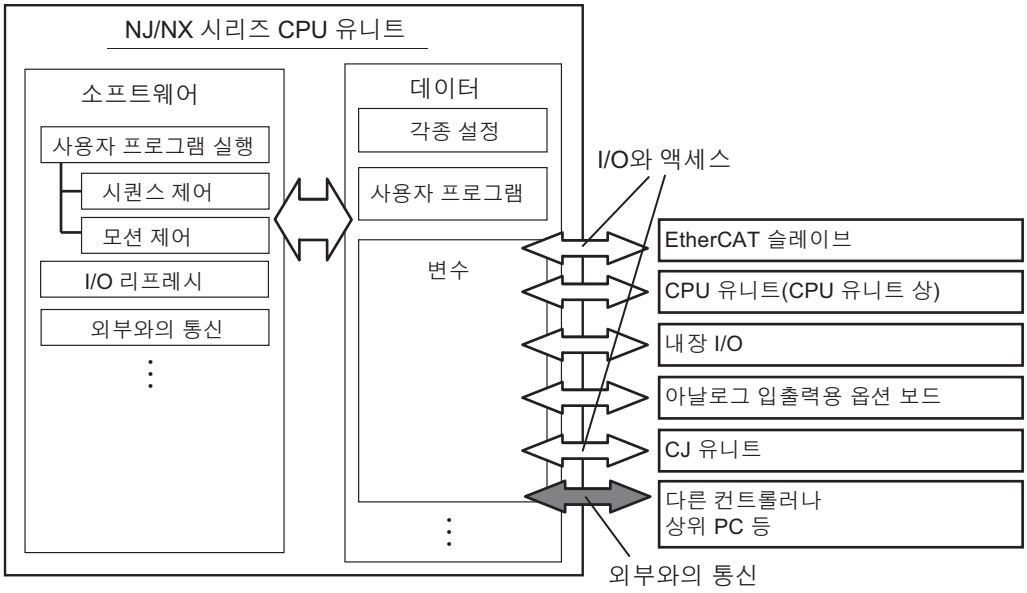
여기에서는 CPU 유니트의 동작으로 CPU 유니트의 구조 및 CPU 유니트의 각 상태의 동작의 개요를 설명합니다.

2-1-1 CPU 유니트 구조의 개요

NJ/NX 시리즈 CPU 유니트는 시퀀스 제어 및 모션 제어를 위한 사용자 프로그램 실행, I/O 리프레시, 외부와의 통신 등의 처리를 수행합니다. 이러한 처리는 CPU 유니트의 소프트웨어가 수행합니다.

또한, CPU 유니트에는 각종 설정, 사용자 프로그램, 변수 등의 데이터가 있습니다. CPU 유니트는 이러한 데이터를 사용하여 처리를 수행합니다. CPU 유니트와 I/O의 액세스 및 외부와의 통신에는 이러한 데이터 중에서 변수를 사용합니다.

CPU 유니트는 소프트웨어 및 변수를 사용한 I/O와의 액세스에 의해 시퀀스 제어와 모션 제어의 양쪽을 실행할 수 있습니다.



(주의) CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
(주의) NX 유니트(CPU 유니트 상)를 사용할 수 있는 것은 NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다. 내장 I/O 및 아날로그 입출력용 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.

이 장에서는 시퀀스 제어, 모션 제어 또는 양쪽의 제어에 CPU 유니트를 사용할 때의 기본 지식으로 다음 내용을 설명합니다.

내용	참조 페이지
소프트웨어의 구조	2-4페이지
I/O와의 액세스 구조	2-12페이지
시퀀스 제어와 모션 제어의 구조	2-28페이지
CPU 유니트의 데이터 개요	2-35페이지
CPU 유니트의 각 상태의 동작	2-36페이지

참고

변수를 사용한 외부와의 통신의 구조에 대해서는 다음 각 설명서를 참조하십시오.

통신	참조 설명서
EtherNet/IP	• 내장 EtherNet/IP 포트의 경우에 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)" • EtherNet/IP 유니트의 경우에 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-355)"
시리얼 통신	"CJ 시리즈 시리얼 커뮤니케이션 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-354)"

2-1-2 CPU 유니트의 각 상태의 동작 개요

CPU 유니트는 이상의 발생 및 동작 모드의 전환 등에 의해 각 상태로 이동합니다. CPU 유니트의 상태 이동에 의해 사용자 프로그램 실행, I/O 리프레시, 외부와의 통신의 처리 동작이 변화합니다.

'2-7CPU 유니트의 각 상태의 동작(2-36페이지)'에서는 CPU 유니트의 각 상태의 각 처리 동작을 설명합니다.

2-2 소프트웨어의 구조

여기에서는 CPU 유니트의 소프트웨어 구성 및 소프트웨어의 동작에 대해 설명합니다.

2-2-1 소프트웨어 구성

CPU 유니트의 소프트웨어는 5가지 기능으로 구성됩니다. 구성하는 기능의 단위를 '기능 모듈'이라고 합니다.
각 기능 모듈과 해당 처리는 다음과 같습니다.

기능 모듈명	처리
PLC 기능 모듈	태스크의 스케줄링, 각 기능 모듈로의 지시, 이벤트 로그, 사용자 프로그램의 실행, CJ 유니트의 I/O 리프레시*1, USB 포트 서비스*2, SD 메모리 카드 서비스, 데이터 트레이스 처리 등을 수행합니다.
NX 버스 기능 모듈*3	CPU 유니트의 NX 버스에 연결한 NX 유니트의 이벤트 로그, I/O 리프레시 등의 처리를 수행합니다.
모션 제어 기능 모듈	모션 제어 처리를 수행합니다.*4
EtherCAT 마스터 기능 모듈	EtherCAT 슬레이브의 I/O 리프레시*5, EtherCAT 메시지 통신*6 등 EtherCAT 마스터로서 EtherCAT 슬레이브와의 통신을 수행합니다.
EtherNet/IP 기능 모듈	태그 데이터 링크 처리, 내장 EtherNet/IP 포트 서비스 등 EtherNet/IP 통신 처리를 수행합니다.

*1. CJ 유니트는 CJ 시리즈 CPU 유니트에 장착 가능한 유니트 중 NJ 시리즈 CPU 유니트에 장착 가능한 유니트군를 말합니다.
*2. NX102 및 NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.
*3. NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.
*4. 모션 제어 명령으로 주어진 목표값(위치, 속도) 등의 지시에 따라 모션 연산을 실행하고, 명령값 출력, 상태 관리 및 정보 취득을 EtherCAT 마스터를 통해 실행합니다.
*5. 프로세스 데이터 통신(이후, PDO 통신으로 생략할 수 있음)에 의해 수행합니다. PDO 통신은 마스터와 슬레이브 간을 일정 주기로 주기적으로 데이터를 교환하는 통신 방식입니다.
*6. EtherCAT 마스터로 EtherCAT 슬레이브와의 통신을 수행합니다.

2-2-2 소프트웨어의 동작

소프트웨어가 실행하는 CPU 유니트의 처리에는 처리가 실행되는 CPU 유니트의 상태 및 처리의 실행 조건에 따라 다음 4종류가 있습니다.

CPU 유니트의 처리 종류	처리가 실행되는 CPU 유니트의 상태	실행 조건	처리 예
기동 처리	기동 중	전원 투입 시에만 실행됩니다.	전원 투입 시의 자체 진단
태스크로 실행되는 처리	정상 운전 중 및 이상 상태	할당된 태스크 내에서 실행됩니다. 태스크에는 주기적으로 반복해서 실행되는 것과 지정한 조건이 성립했을 때 1회만 실행되는 것이 있습니다.	사용자 프로그램 실행, I/O 리프레시
태그 데이터 링크 서비스		하드웨어 및 외부에서의 요구가 있었을 때만 실행됩니다.	태그 데이터 링크
옵션 보드 서비스*1			아날로그 입출력용 옵션 보드의 I/O 리프레시
시스템 서비스			USB 포트 서비스*2, SD 메모리 카드 서비스, 시리얼 통신용 옵션 보드의 통신 처리 *3

*1. 옵션 보드 서비스는 NX1P2 CPU 유니트에서만 실행됩니다.
*2. NX102 및 NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.
*3. 시리얼 통신용 옵션 보드는 NX1P2 CPU 유니트만 대응합니다.

CPU 유니트의 상태에 대해서는 '2-7-1 CPU 유니트의 상태(2-36페이지)'를 참조하십시오.

여기에서는 각 처리의 동작을 설명합니다.

기동 처리

기동 처리는 전원 투입 시에만 실행됩니다.
기동 처리로는 다음 처리를 실행합니다.

처리	내용
전원 투입 시의 자체 진단	'전원부 이상', 'CPU 리셋', 'CPU 이상(WDT 이상)', '잘못된 전원 연결'을 검출합니다.*1
데이터 확인	전원 OFF 유지 속성을 갖는 변수의 값 및 CJ 유니트용 메모리(DM/EM/HR페이지)*2 의 값이 전원 OFF 중에 유지되지 못한 경우에 시스템 정의 변수 '_RetainFail'(전원 OFF 유지 실패 플래그)을 TRUE로 합니다.
NX 유니트의 인식*3	컨트롤러에 장착된 모든 NX 유니트를 인식합니다.
CJ 유니트의 인식*4	컨트롤러에 장착된 모든 CJ 유니트를 인식합니다.
전원 투입 및 전원 OFF의 이벤트 기록	전원 투입 및 전원 OFF의 이벤트를 기록합니다.

*1. '전원부 이상', 'CPU 리셋', 'CPU 이상(WDT 이상)', '잘못된 전원 연결'에 대해서는 "NJ/NX 시리즈 트러블슈팅 설명서(SBCA-361)"의 'CPU 유니트가 동작할 수 없는 이상'을 참조하십시오.
*2. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.
*3. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*4. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

태스크로 실행되는 처리

● 태스크로 실행되는 처리의 종류

태스크에서는 다음과 같은 처리가 실행됩니다.

처리	내용
I/O 리프레시	EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트*1 , CPU 유니트의 내장 I/O*2 , CPU 유니트의 아날로그 입출력용 옵션 보드*3 및 CJ 유니트*4 의 기본 I/O 유니트 및 고기능 유니트와 데이터 입출력을 수행합니다.
사용자 프로그램 실행	시퀀스 제어의 사용자 프로그램을 실행합니다. 또한, 모션 제어에 명령을 전송합니다.
모션 제어	사용자 프로그램의 명령을 받아서 모션 제어를 실행합니다.
시스템 공통 처리	데이터 트레이스 처리, 태그 데이터 링크 처리 등을 실행합니다.

*1. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*2. 내장 I/O를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*3. 아날로그 입출력용 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*4. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

태스크로 실행되는 처리의 자세한 내용은 '5-3-3 NX701의 태스크 기본 동작(5-10페이지)', '5-4-3 NX102 및 NX1P2의 태스크 기본 동작(5-30페이지)' 및 '5-5-3 NJ 시리즈의 태스크 기본 동작(5-47페이지)'을 참조하십시오.

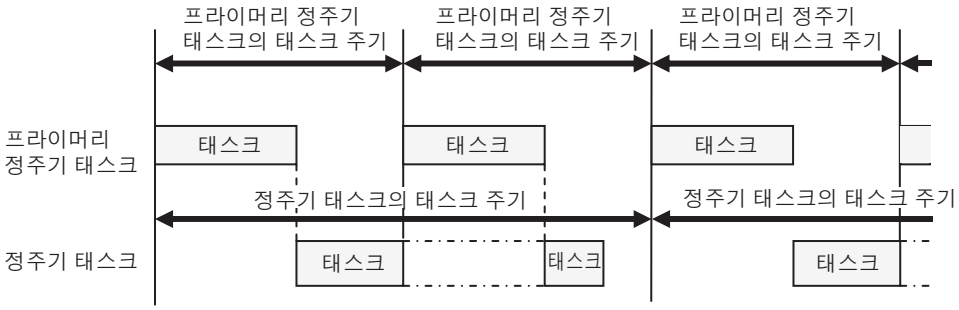
● 태스크의 동작

각 처리는 태스크에 할당됩니다. 태스크에는 실행 우선도와 실행 조건에 따라 다음 3종류가 있습니다.

정주기 태스크의 종류	실행 우선도 (값이 작을수록 높음)	실행 조건	주요 처리 내용
프라이머리 정주기 태스크	4 가장 실행 우선도가 높은 태스크입니다.	주기적으로 반복해서 실행됩니다.	I/O 리프레시, 사용자 프로그램 실행, 모션 제어
정주기 태스크	5*1 프라이머리 정주기 태스크 다음으로 우선도가 높은 태스크입니다.	주기적으로 반복해서 실행됩니다. 태스크 주기는 프라이머리 정주기 태스크의 태스크 주기의 정수배입니다.	I/O 리프레시, 사용자 프로그램 실행, 모션 제어
	16*2 ,17,18	주기적으로 반복해서 실행됩니다. 태스크 주기는 프라이머리 정주기 태스크의 태스크 주기의 정수배입니다.	실행 우선도에 따라 처리 내용이 다릅니다. 실행 우선도 16: I/O 리프레시, 사용자 프로그램 실행 실행 우선도 17, 18: 사용자 프로그램 실행
이벤트 태스크	8,48	사전에 지정한 조건이 성립했을 때 1회만 실행됩니다.	사용자 프로그램 실행

*1. 실행 우선도 5의 정주기 태스크를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*2. NX102 및 NX1P2 CPU 유니트에서는 실행 우선도 16의 정주기 태스크를 사용할 수 없습니다.

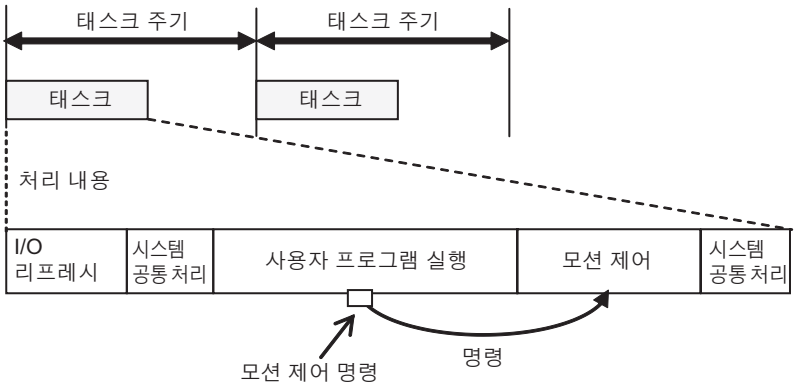
CPU 유니트는 실행 우선도가 높은 태스크를 우선적으로 실행합니다. 다음 그림은 프라이머리 정주기 태스크와 정주기 태스크의 동작 예입니다. 정주기 태스크 실행 중에 프라이머리 정주기 태스크가 실행되면 프라이머리 정주기 태스크의 실행을 우선합니다.



우선도가 높은 태스크가 실행 중인 우선도의 낮은 태스크 동작은 NX701 CPU 유니트, NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서 다릅니다. 태스크 동작의 자세한 내용은 '5-2 태스크의 개요(5-5페이지)'를 참조하십시오.

● 태스크로 실행되는 처리의 동작

태스크에 할당된 처리는 태스크 내에서 다음 그림의 순서로 실행됩니다. 프로그램에 모션 제어 명령이 기술된 경우에는 태스크 내의 '사용자 프로그램 실행'에서 '모션 제어'에 명령을 전송합니다. '모션 제어'는 명령을 바탕으로 모션 제어 처리를 수행합니다.



(주의) '모션 제어' 처리는 프라이머리 정주기 태스크와 실행 우선도 5의 정주기 태스크로 실행됩니다.
자세한 내용은 '5-11-3 시스템으로서의 입출력 응답 시간(5-112페이지)'을 참조하십시오.

참고

NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)로 모션 제어를 실행할 수 있습니다. 2가지 모션 제어를 구별해야 하는 경우에 프라이머리 정주기 태스크의 모션 제어를 '모션 제어 1', 정주기 태스크(실행 우선도 5)의 모션 제어를 '모션 제어 2'라고 합니다.

태그 데이터 링크 서비스

● 태그 데이터 링크 서비스의 처리

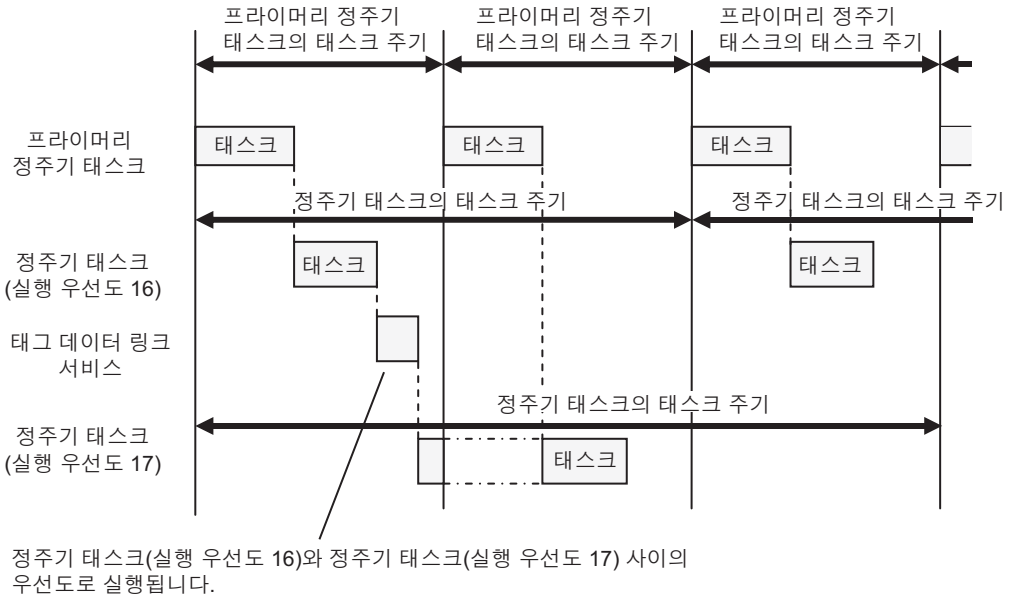
태그 데이터 링크 서비스는 EtherNet/IP 네트워크 상에서 다른 컨트롤러 및 장치와 태그를 사용하여 통신하는 처리입니다. CPU 유니트의 내장 EtherNet/IP 포트 및 CJ 시리즈 EtherNet/IP 유니트 CJ1W-EIP21을 사용합니다.

(주의) CJ 시리즈 EtherNet/IP 유니트 CJ1W-EIP21를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

● 태그 데이터 링크 서비스의 동작

태그 데이터 링크 서비스는 일정 주기로 실행됩니다. 주기 및 1회 처리 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동합니다.

NJ 시리즈 CPU 유니트의 태그 데이터 링크 서비스는 실행 우선도가 정주기 태스크(실행 우선도 16)와 정주기 태스크(실행 우선도 17) 사이입니다. 따라서 정주기 태스크(실행 우선도 17)보다도 우선적으로 실행되지만 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16)가 실행되면 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16)가 우선됩니다.



NX 시리즈 CPU 유니트의 태그 데이터 링크 서비스는 태스크의 실행과 병렬로 실행됩니다.

태그 데이터 링크 서비스의 자세한 내용은 📖'5-6 태스크 이외의 서비스(5-65페이지)'를 참조하십시오.

옵션 보드 서비스

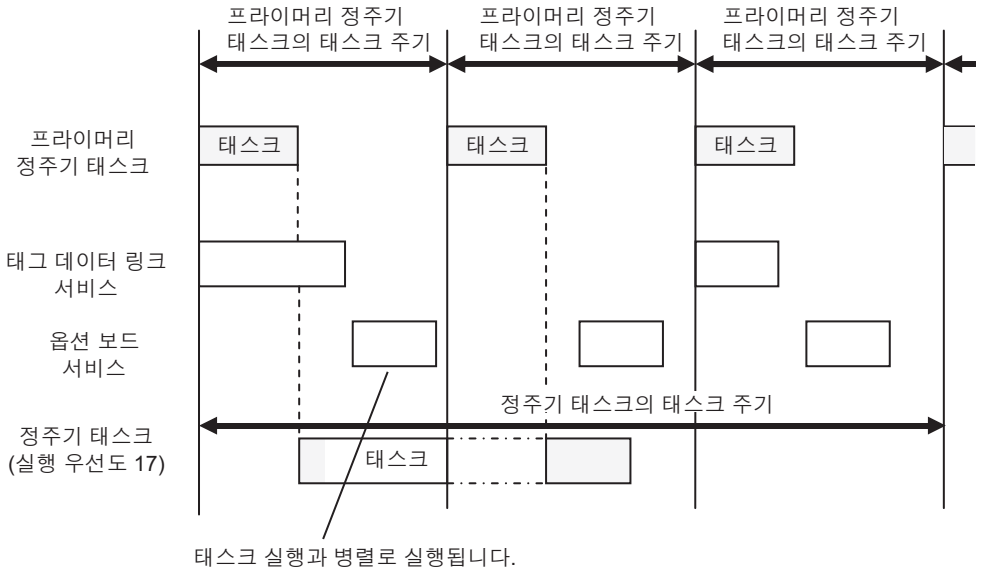
● 옵션 보드 서비스의 처리

옵션 보드 서비스는 NX1P2 CPU 유니트에 장착한 아날로그 입출력용 옵션 보드의 I/O 리프레시를 처리합니다.

(주의) 옵션 보드 서비스는 NX1P2 CPU 유니트에서만 실행됩니다.

● 옵션 보드 서비스의 동작

NX1P2 CPU 유니트의 옵션 보드 서비스는 태스크의 실행과 병렬로 실행됩니다. 단, 태그 데이터 링크 서비스의 실행 중에는 옵션 보드 서비스는 실행되지 않습니다.



옵션 보드 서비스의 자세한 내용은 '5-6 태스크 이외의 서비스(5-65페이지)'를 참조하십시오.

시스템 서비스

● 시스템 서비스의 종류

시스템 서비스에는 다음과 같은 처리가 있습니다.

처리	내용
USB 포트 서비스*1	• 주변 툴, 상위 PC에서의 서비스 요구에 대한 처리
내장 EtherNet/IP 포트 서비스	• 주변 툴, 표시기, 상위 PC 또는 다른 컨트롤러 등에서의 서비스 요구(CIP 등)에 대한 처리 • 통신 명령(CIP, 소켓)의 실행
내장 EtherCAT 포트 서비스	• EtherCAT 메시지 통신의 실행
시리얼 통신용 옵션 보드의 통신 처리*2	• 시리얼 통신용 옵션 보드의 통신 처리 실행
CJ 고기능 유니트 서비스*3	• CJ 고기능 유니트와의 이벤트 서비스 실행 • 통신 명령(CIP)의 실행
SD 메모리 카드 서비스	• FTP 클라이언트에서의 액세스 • Sysmac Studio에서의 SD 메모리 카드 조작 • SD 메모리 카드 명령의 실행
자체 진단 처리	• 하드웨어의 이상 감지

*1. NX102 및 NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.
*2. 시리얼 통신용 옵션 보드의 통신 처리는 NX1P2 CPU 유니트에서만 실행됩니다.
*3. CJ 고기능 유니트와의 할당 영역의 데이터 교환은 'I/O 리프레시' 시에 실행됩니다.
CJ 고기능 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

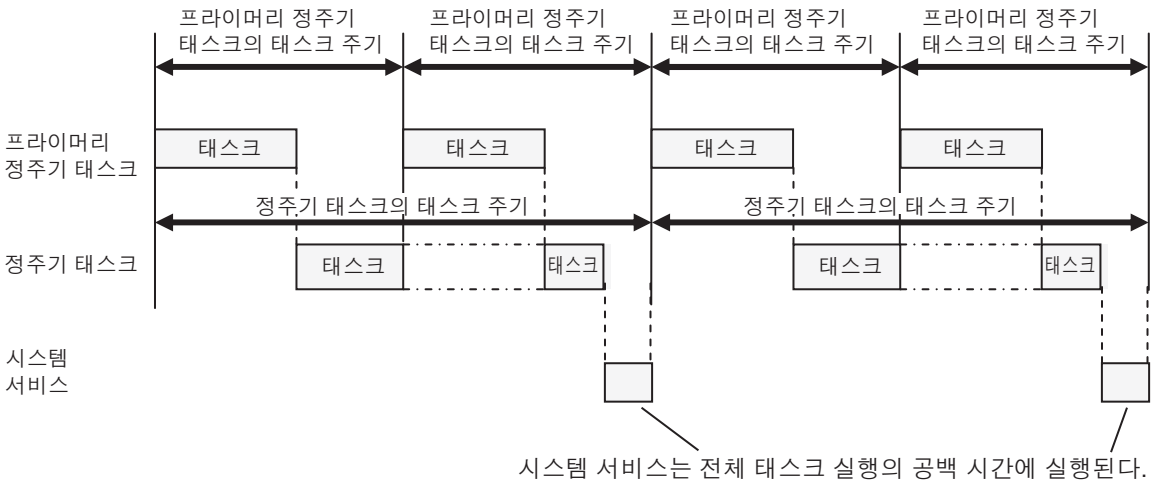
● 시스템 서비스의 동작

NX701 CPU 유니트에서는 하드웨어 및 CPU 유니트 외부에서의 요구가 있었던 경우에 시스템 서비스는 다른 처리와 병렬로 실행됩니다. 태스크 실행 등의 영향을 받지 않고 언제라도 실행됩니다.

NX102 CPU 유니트에서는 하드웨어 및 CPU 유니트 외부에서의 요구가 있었던 경우에 시스템 서비스는 다른 처리와 병렬로 실행됩니다. 태스크 실행 등의 영향을 받지 않고 언제라도 실행됩니다. 단, 태그 데이터 링크 서비스 실행 중에는 시스템 서비스는 실행되지 않습니다.

NX1P2 CPU 유니트에서는 하드웨어 및 CPU 유니트 외부에서의 요구가 있었던 경우에 시스템 서비스는 다른 처리와 병렬로 실행됩니다. 태스크 실행 등의 영향을 받지 않고 언제라도 실행됩니다. 단, 태그 데이터 링크 서비스 및 옵션 보드 서비스의 실행 중에는 시스템 서비스는 실행되지 않습니다.

NJ 시리즈 CPU 유니트에서는 하드웨어 및 CPU 유니트 외부에서의 요구가 있었던 경우에 시스템 서비스는 모든 태스크 실행의 비는 시간에 실행됩니다.



시스템 서비스의 자세한 내용은 '5-6 태스크 이외의 서비스(5-65페이지)'를 참조하십시오.

2-3 I/O와의 액세스 구조

CPU 유니트는 변수를 사용하여 I/O와 액세스합니다. 여기에서는 변수와 I/O와의 액세스의 구조에 대해 설명합니다.
본 설명서에서는 EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 아날로그 입출력용 옵션 보드 및 CJ 유니트를 I/O로 취급합니다. 태그 데이터 링크를 사용한 다른 컨트롤러와의 액세스에 대해서는 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.

2-3-1 변수의 종류

NJ/NX 시리즈 CPU 유니트에서는 사용자 프로그램에서 변수를 통해 I/O 및 CPU 유니트 내부 등에 액세스합니다.
변수는 사용자에 의한 속성 설정 가능 여부 및 액세스 대상에 따라 다음과 같이 분류됩니다.

<table>
<tr><th colspan="3">변수</th><th>속성 설정 가능 여부</th><th>액세스 대상</th></tr>
<tr><td colspan="3">사용자 정의 변수</td><td>모든 속성 설정 가능</td><td>CPU 유니트 내부</td></tr>
<tr><td rowspan="6">준 사용자 정의 변수</td><td rowspan="5">디바이스 변수</td><td>EtherCAT 슬레이브 디바이스 변수</td><td rowspan="6">일부 속성이 설정 가능*1</td><td>EtherCAT 슬레이브*2*3</td></tr>
<tr><td>NX 유니트 디바이스 변수*4</td><td>CPU 유니트 상의 NX 유니트</td></tr>
<tr><td>내장 I/O 디바이스 변수 *5</td><td>내장 I/O</td></tr>
<tr><td>옵션 보드 디바이스 변수*5</td><td>아날로그 입출력용 옵션 보드</td></tr>
<tr><td>CJ 유니트 디바이스 변수*6</td><td>CJ 시리즈 기본 I/O 유니트, CJ 시리즈 고기능 유니트</td></tr>
<tr><td colspan="2">캠 데이터 변수</td><td>서보 드라이버, 엔코더 입력 슬레이브, CPU 유니트 내부</td></tr>
<tr><td rowspan="6">시스템 정의 변수</td><td colspan="2">PLC 시스템 변수</td><td rowspan="6">모든 속성이 설정 불가능</td><td>CPU 유니트 내부</td></tr>
<tr><td rowspan="3">모션 제어 시스템 변수</td><td>MC 공통 변수</td><td rowspan="3">서보 드라이버, 엔코더 입력 슬레이브, CPU 유니트 내부</td></tr>
<tr><td>축 변수</td></tr>
<tr><td>축 그룹 변수</td></tr>
<tr><td colspan="2">EtherNet/IP 시스템 변수</td><td>내장 EtherNet/IP 포트</td></tr>
<tr><td colspan="2">EtherCAT 마스터 시스템 변수</td><td>내장 EtherCAT 마스터 포트</td></tr>
<tr><td colspan="2">NX 버스 시스템 변수*7</td><td>CPU 유니트 내부</td></tr>
</table>

*1. 설정 가능한 속성에 대해서는 '디바이스 변수의 속성(3-12페이지)'을 참조하십시오.
*2. EtherCAT 슬레이브에는 EtherCAT 슬레이브 터미널 상의 NX 유니트도 포함됩니다.
*3. Sysmac Studio Ver.1.08 이하에서는 '축'에 할당된 EtherCAT 슬레이브는 EtherCAT 슬레이브 디바이스 변수의 액세스 대상으로 할 수 없습니다.
*4. NX 유니트 디바이스 변수를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*5. 내장 I/O 디바이스 변수 및 옵션 보드 디바이스 변수를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*6. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*7. NX 버스 시스템 변수를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.

사용자 정의 변수

모든 속성이 사용자에 의해 정의되는 변수입니다. 사용자 정의 변수의 자세한 내용은 '6-3 변수(6-28 페이지)'를 참조하십시오.

준 사용자 정의 변수

변수 속성의 일부가 설정 가능한 변수입니다. 특정 대상의 데이터에 액세스하기 위해 사용합니다. 준 사용자 정의 변수는 액세스 대상에 따라 디바이스 변수와 캠 데이터 변수로 분류됩니다.

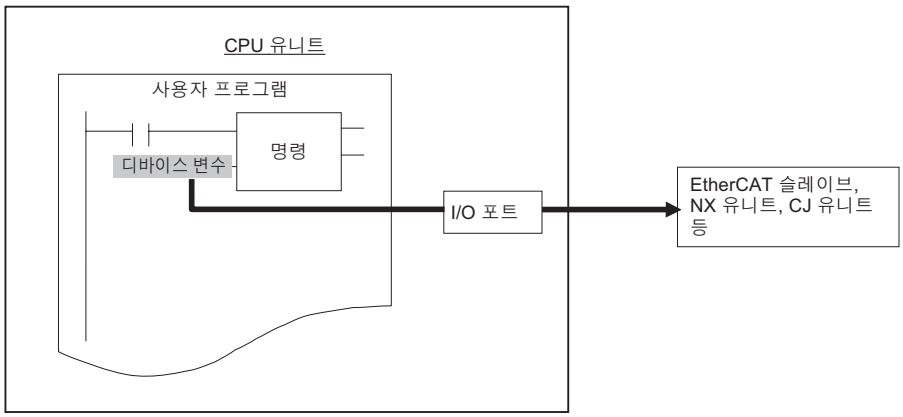
● 디바이스 변수

디바이스 변수는 장치의 데이터에 액세스하는 변수입니다. 장치는 CPU 유니트에서 I/O 리프레시를 하는 대상의 총칭입니다. EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 아날로그 입출력용 옵션 보드 및 CJ 유니트를 나타냅니다.
장치 및 장치의 액세스 대상 데이터에 따라 다음과 같은 종류가 있습니다.

디바이스 변수의 종류	장치	액세스 대상 데이터
EtherCAT 슬레이브 디바이스 변수	EtherCAT 슬레이브*1	EtherCAT 상의 슬레이브의 프로세스 데이터*2
NX 유니트 디바이스 변수*3	CPU 유니트 상의 NX 유니트	NX 유니트의 I/O 데이터
내장 I/O 디바이스 변수 *4	내장 I/O	내장 I/O의 I/O 데이터
옵션 보드 디바이스 변수	아날로그 입출력용 옵션 보드	아날로그 입출력용 옵션 보드의 I/O 데이터
CJ 유니트 디바이스 변수*5	CJ 시리즈 기본 I/O 유니트	기본 I/O 유니트의 실제 I/O 데이터
	CJ 시리즈 고기능 유니트	고기능 유니트의 운전용 데이터*6 , 설정용 데이터*7

*1. Sysmac Studio Ver.1.08 이하에서는 '축'에 할당된 EtherCAT 슬레이브는 EtherCAT 슬레이브 디바이스 변수의 액세스 대상으로 할 수 없습니다.
*2. 마스터와 슬레이브 사이에서 프로세스 데이터 통신 주기마다 교환하는 I/O 데이터입니다.
*3. NX 유니트 디바이스 변수를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*4. 내장 I/O 디바이스 변수 및 옵션 보드 디바이스 변수를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*5. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*6. CJ 유니트의 운전에 사용하는 데이터입니다. CJ 유니트용 메모리의 CIO 영역을 사용합니다.
*7. CJ 유니트의 설정에 사용하는 데이터입니다. CJ 유니트용 메모리의 DM 영역을 사용합니다.

디바이스 변수는 I/O 포트를 통해 EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 아날로그 입출력용 옵션 보드 및 CJ 유니트에 액세스합니다. I/O 포트는 장치와의 액세스용 논리 포트입니다.



I/O 포트 및 디바이스 변수의 자세한 내용은 "3-3-1 I/O 포트(3-8페이지)"를 참조하십시오.

● 캠 데이터 변수

모션 제어로 캠 테이블의 데이터에 액세스하는 변수입니다. 자세한 내용은 "NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

시스템 정의 변수

시스템 정의 변수는 NJ/NX 시스템에 따라 이름 등의 모든 속성이 사전에 정의되어 특정 기능이 할당된 변수입니다. 사용자가 이름 등의 속성을 변경할 수 없습니다.
특정 기능은 기능 모듈별로 있기 때문에 기능 모듈별로 시스템 정의 변수가 존재합니다. 시스템 정의 변수의 종류는 다음과 같습니다.

기능 모듈	시스템 정의 변수의 종류
PLC 기능 모듈	PLC 시스템 변수
NX 버스 기능 모듈*1	NX 버스 시스템 변수
모션 제어 기능 모듈	모션 제어 시스템 변수
EtherNet/IP 기능 모듈	EtherNet/IP 시스템 변수
EtherCAT 마스터 기능 모듈	EtherCAT 마스터 시스템 변수

*1. NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.

그리고 모션 제어 시스템 변수는 모션 제어 기능의 내용에 따라 다음과 같이 분류됩니다.

모션 제어 시스템 변수	기능의 내용
MC 공통 변수	모션 제어 기능 모듈 공통 처리
축 변수	'축' 제어
축 그룹 변수	축 그룹*1 의 제어

*1. 축 그룹은 복수의 '축'을 묶은 그룹입니다. 보간 동작에 사용합니다.

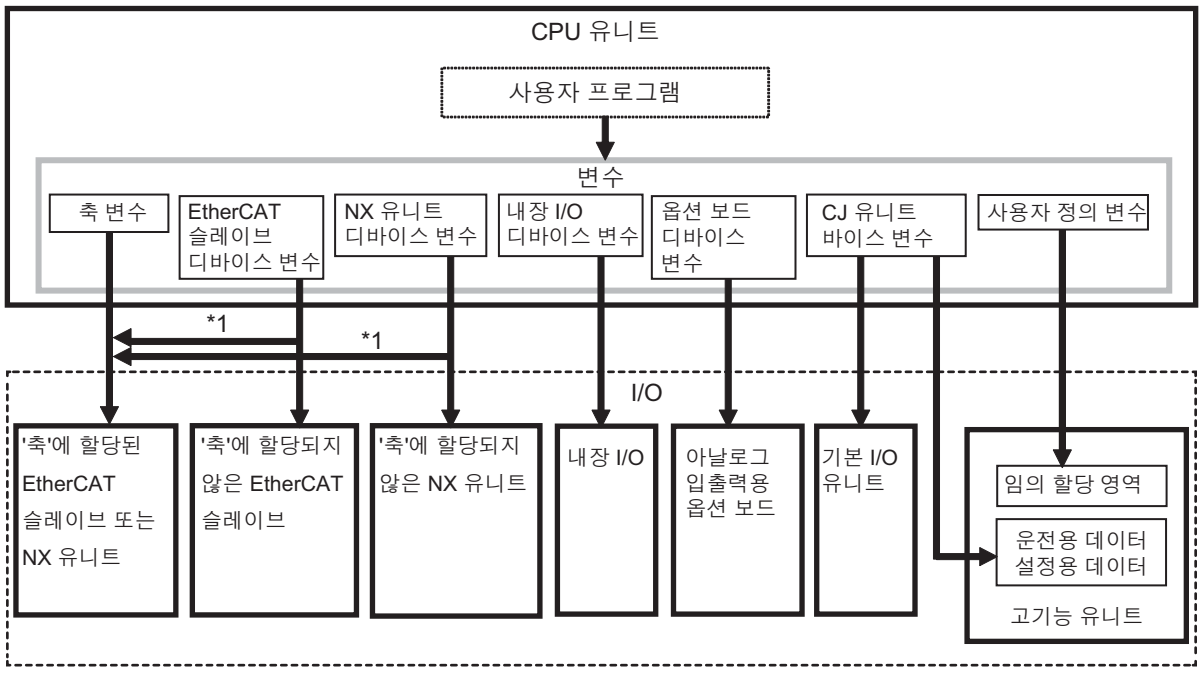
시스템 정의 변수의 자세한 내용은 'A-6 시스템 정의 변수 목록(A-64페이지)'을 참조하십시오.

2-3-2 변수와 I/O의 액세스

CPU 유니트는 사용자 프로그램에서 변수를 사용합니다. 변수는 할당된 I/O의 데이터와 액세스합니다. CPU 유니트에서는 다음과 같이 I/O와 변수를 할당합니다. CJ 유니트의 고기능 유니트는 액세스하는 데이터에 따라 사용하는 변수가 다릅니다.

I/O		데이터	변수
EtherCAT 슬레이브	'축'에 할당되지 않은 EtherCAT 슬레이브	-	EtherCAT 슬레이브 디바이스 변수
	'축'에 할당된 EtherCAT 슬레이브	-	• EtherCAT 슬레이브 디바이스 변수*1 • 축 변수
CPU 유니트 상의 NX 유니트*2	'축'에 할당되지 않은 NX 유니트	-	NX 유니트 디바이스 변수
	'축'에 할당된 NX 유니트	-	• NX 유니트 디바이스 변수 • 축 변수
내장 I/O*3		-	내장 I/O 디바이스 변수
아날로그 입출력용 옵션 보드*4		-	옵션 보드 디바이스 변수
CJ 유니트*5	기본 I/O 유니트	-	CJ 유니트 디바이스 변수
	고기능 유니트	• 운전용 데이터 • 설정용 데이터	CJ 유니트 디바이스 변수
		할당 임의 영역의 데이터*6	사용자 정의 변수

*1. Sysmac Studio Ver.1.08 이하에서는 '축'에 할당된 EtherCAT 슬레이브는 EtherCAT 슬레이브 디바이스 변수의 액세스 대상으로 할 수 없습니다.
*2. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*3. 내장 I/O를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*4. 아날로그 입출력용 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*5. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*6. 확장 기능 및 슬레이브의 입출력 등을 사용자가 메모리 영역에 할당해서 사용하는 데이터입니다. 할당 임의 영역의 데이터는 디바이스 변수로 액세스할 수 없습니다.



EtherCAT 슬레이브와의 액세스

EtherCAT 슬레이브와의 액세스 방법은 EtherCAT 슬레이브의 종류에 따라 다릅니다.

EtherCAT 슬레이브의 종류	액세스 방법
• 서보/엔코더 입력 슬레이브*1 • 범용 슬레이브	EtherCAT 슬레이브 디바이스 변수를 사용하여 I/O 포트를 통해 액세스
'축'에 할당된 서보/엔코더 입력 슬레이브	축 변수에서 직접 액세스*2

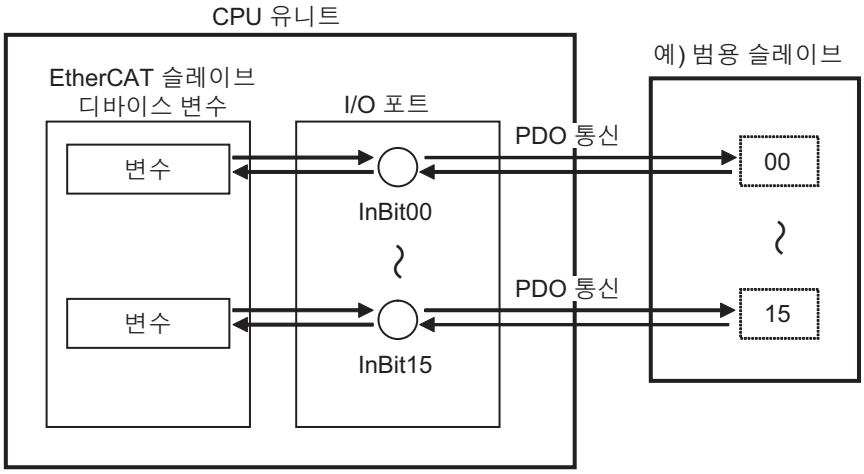
*1. Sysmac Studio Ver.1.08 이하에서는 '축'에 할당된 EtherCAT 슬레이브는 EtherCAT 슬레이브 디바이스 변수의 액세스 대상으로 할 수 없습니다.

*2. 서보 드라이버의 경우에는 1대의 서보 모터, 엔코더 입력 슬레이브의 경우에는 1개의 카운터를 하나의 '축'으로 축 변수에 할당해서 사용합니다.

(주의) '축'에 할당할 수 없는 EtherCAT 슬레이브를 '범용 슬레이브', '축'에 할당할 수 있는 EtherCAT 슬레이브를 '서보/엔코더 입력 슬레이브'라고 합니다. 서보 드라이버/엔코더 입력 슬레이브에 대해서는 "NJ/NX시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

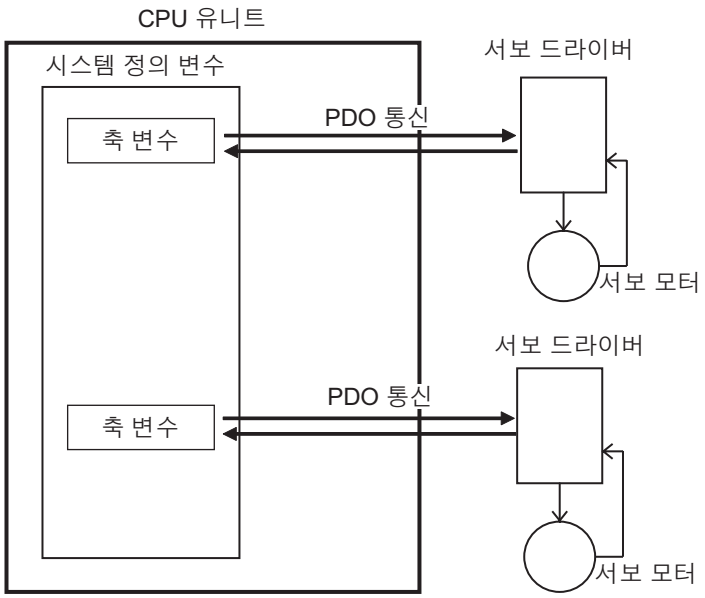
● '축'에 할당되지 않은 서보/엔코더 입력 슬레이브, 또는 범용 슬레이브와의 액세스

EtherCAT 슬레이브 디바이스 변수에서 I/O 포트를 통해 액세스합니다. I/O 포트에서의 액세스는 PDO 통신으로 실행됩니다.



● '축'에 할당된 서보/엔코더 입력 슬레이브와의 액세스

'축'에 할당된 서보/엔코더 입력 슬레이브에 축 변수가 직접 액세스합니다. 축 변수에서의 액세스는 PDO 통신으로 실행됩니다.
예를 들어 서보 드라이버를 사용하여 서보 모터를 제어하는 경우에 축 변수에 할당된 서보 모터에 대한 제어의 지시가 서보 드라이버로 송신됩니다. 서보 모터에서의 피드백은 축 변수를 사용하여 서보 드라이버에서 CPU 유니트로 송신됩니다.



축 변수의 자세한 내용은 '3-5-2 축 변수/축 그룹 변수(3-18페이지)'를 참조하십시오.

버전 관련 정보

Sysmac Studio Ver.1.09 이상에서는 '축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당할 수 있습니다.
디바이스 변수를 할당 가능한 I/O 포트는 다음 중 하나의 조건을 충족하는 것입니다.

- R/W 속성의 값이 R(읽기 전용)일 것
- R/W 속성의 값이 W(쓰기 전용)이고, Sysmac Studio의 [축 기본 설정] 화면의 [세부 내용 설정]에서 프로세스 데이터의 설정이 <할당 안 함>일 것

사용상의 주의 사항

'축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당하는 경우에 다음 조작을 실행하면 디바이스 변수의 할당이 해제되므로 주의하십시오.

(1) Sysmac Studio Ver.1.09 이상이고, '축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당

(2) 프로젝트 데이터 저장

(3) 해당 프로젝트 데이터를 Sysmac Studio Ver.1.08 이하에서 열기

참고

EtherCAT 통신에는 PDO 통신과 SDO 통신이 있습니다. 서보 모터의 위치 제어 등 일정한 제어 주기로 입출력 데이터 갱신을 실시하는 명령에 대해서는 PDO 통신을 사용합니다. 파라메터 전송 등 지정된 타이밍에 데이터 읽기 및 쓰기를 실시하는 명령에는 SDO 통신을 사용합니다. 자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"를 참조하십시오.

CPU 유니트 상의 NX 유니트와의 액세스

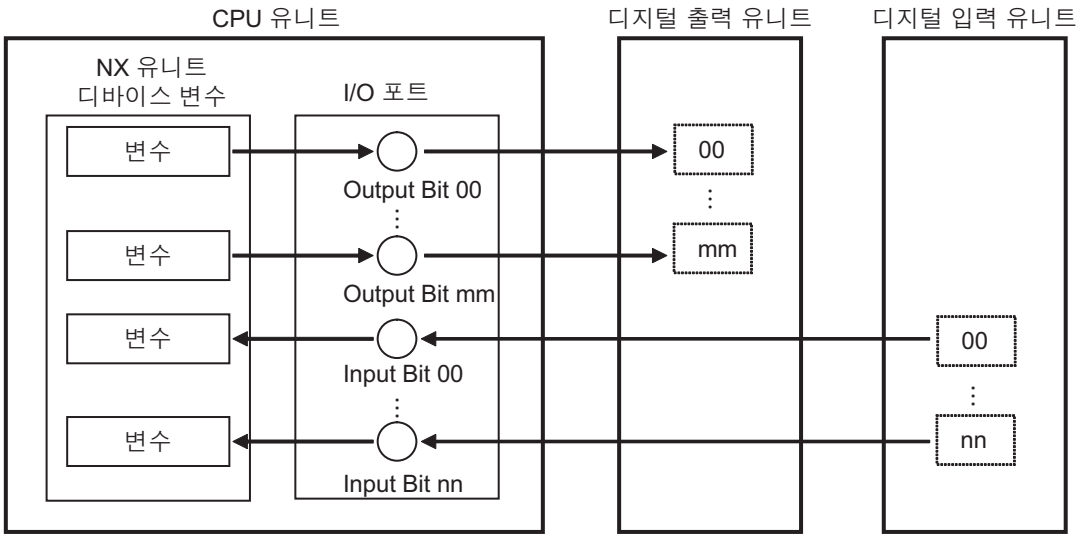
NX102 및 NX1P2 CPU 유니트 상의 NX 유니트와의 액세스 방법은 NX 유니트의 종류에 따라 다릅니다.

CPU 유니트 상의 NX 유니트의 종류*1	액세스 방법
'축'에 할당되지 않은 NX 유니트	NX 유니트 디바이스 변수를 사용하여 I/O 포트를 통해 액세스
'축'에 할당된 NX 유니트	축 변수에서 직접 액세스
	NX 유니트 디바이스 변수를 사용하여 I/O 포트를 통해 액세스

*1. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.

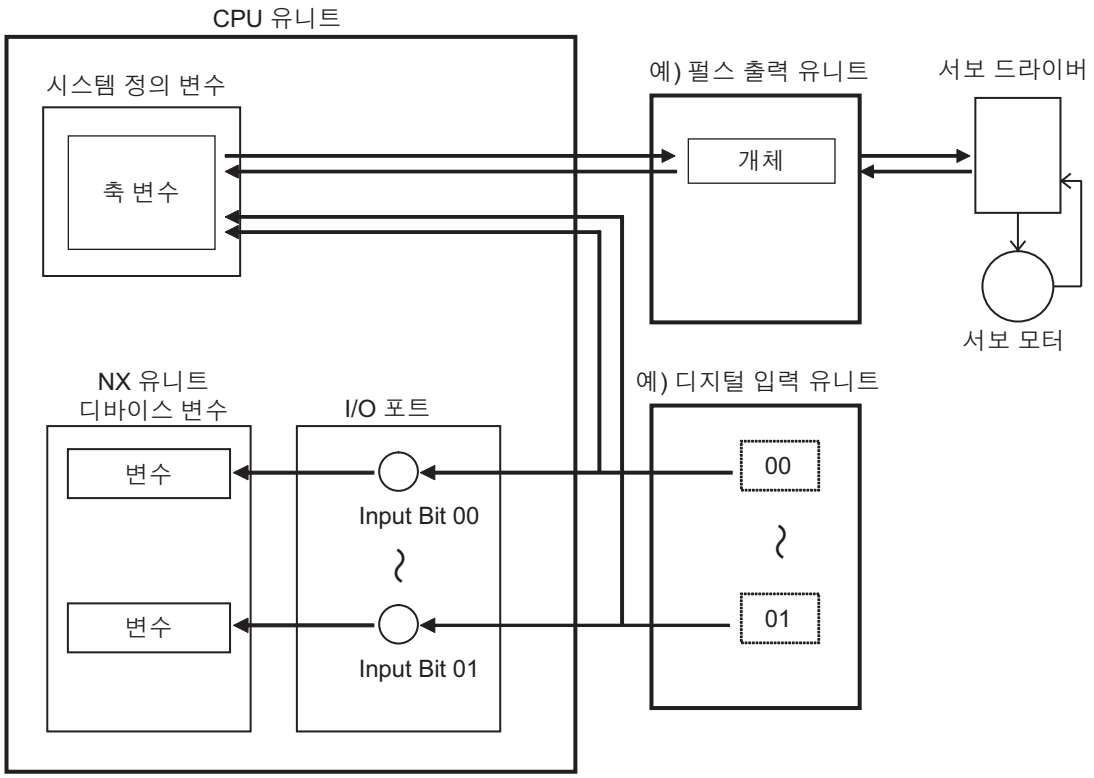
● '축'에 할당되지 않은 NX 유니트와의 액세스

NX 유니트 디바이스 변수에서 I/O 포트를 통해 액세스합니다.



● '축'에 할당된 NX 유니트와의 액세스

'축'에 할당된 NX 유니트에 축 변수가 직접 액세스합니다.
또한, '축'에 할당된 NX 유니트의 I/O 포트에 디바이스 변수를 할당할 수도 있습니다.



축 변수의 자세한 내용은 '3-5-2 축 변수/축 그룹 변수(3-18페이지)'를 참조하십시오.

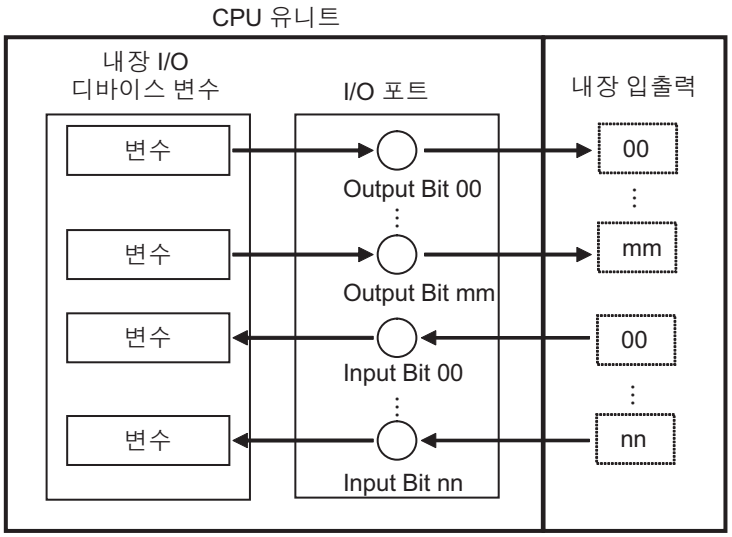
사용상의 주의 사항

'축'에 할당된 NX 유니트에서 디바이스 변수를 할당 가능한 I/O 포트는 다음 중 하나의 조건을 충족하는 것입니다.

- R/W 속성의 값이 R(읽기 전용)일 것
- R/W 속성의 값이 W(쓰기 전용)이고, Sysmac Studio의 [축 기본 설정] 화면의 [세부 내용 설정]에서 프로세스 데이터의 설정이 <할당 안 함>일 것

내장 I/O와의 액세스

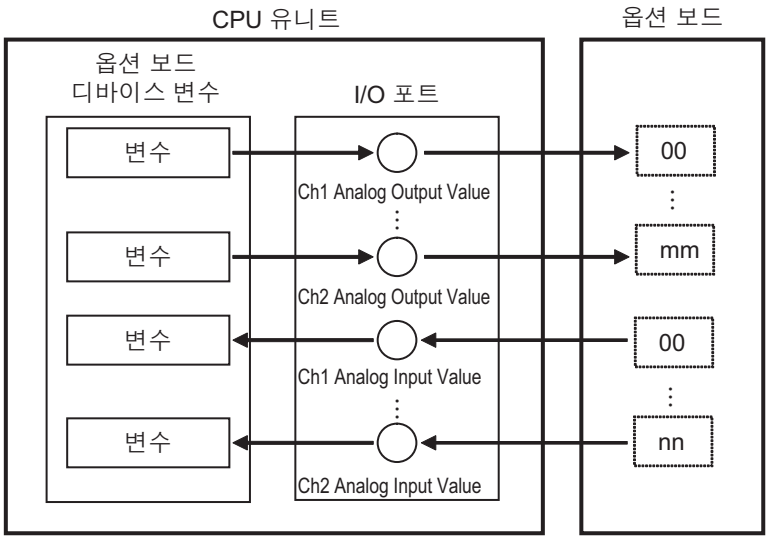
NX1P2 CPU 유니트의 내장 I/O와의 액세스는 내장 I/O 디바이스 변수에서 I/O 포트를 통해 실행합니다.



자세한 내용은 '3-3-1 I/O 포트(3-8페이지)'를 참조하십시오.

아날로그 입출력용 옵션 보드와의 액세스

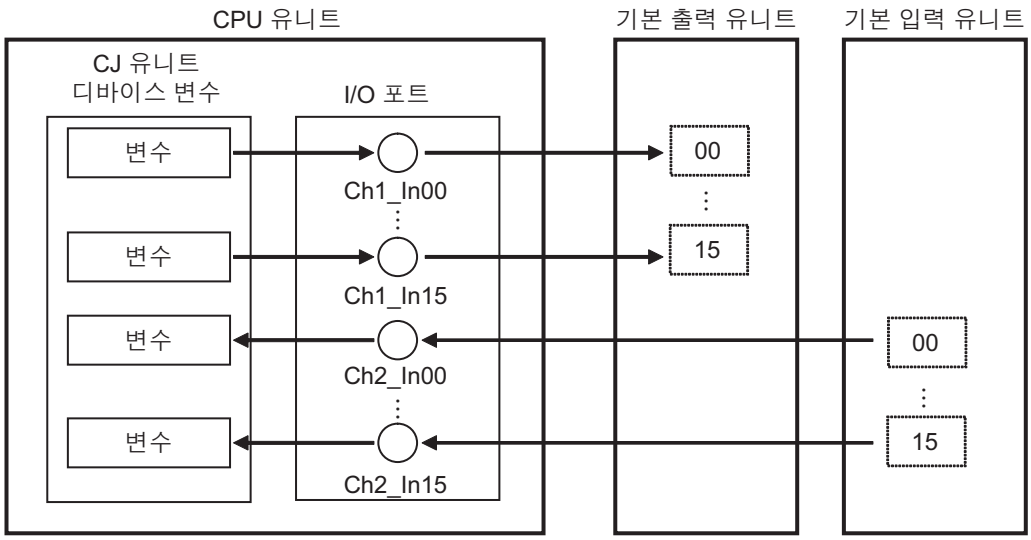
NX1P2 CPU 유니트의 아날로그 입출력용 옵션 보드와의 액세스는 옵션 보드 디바이스 변수에서 I/O 포트를 통해 실행합니다.



자세한 내용은 '3-3-1 I/O 포트(3-8페이지)'를 참조하십시오.

기본 I/O 유니트와의 액세스

NJ 시리즈 CPU 유니트에서의 기본 I/O 유니트와의 액세스는 CJ 유니트 디바이스 변수에서 I/O 포트를 통해 실행합니다.



자세한 내용은 '3-3-1 I/O 포트(3-8페이지)'를 참조하십시오.

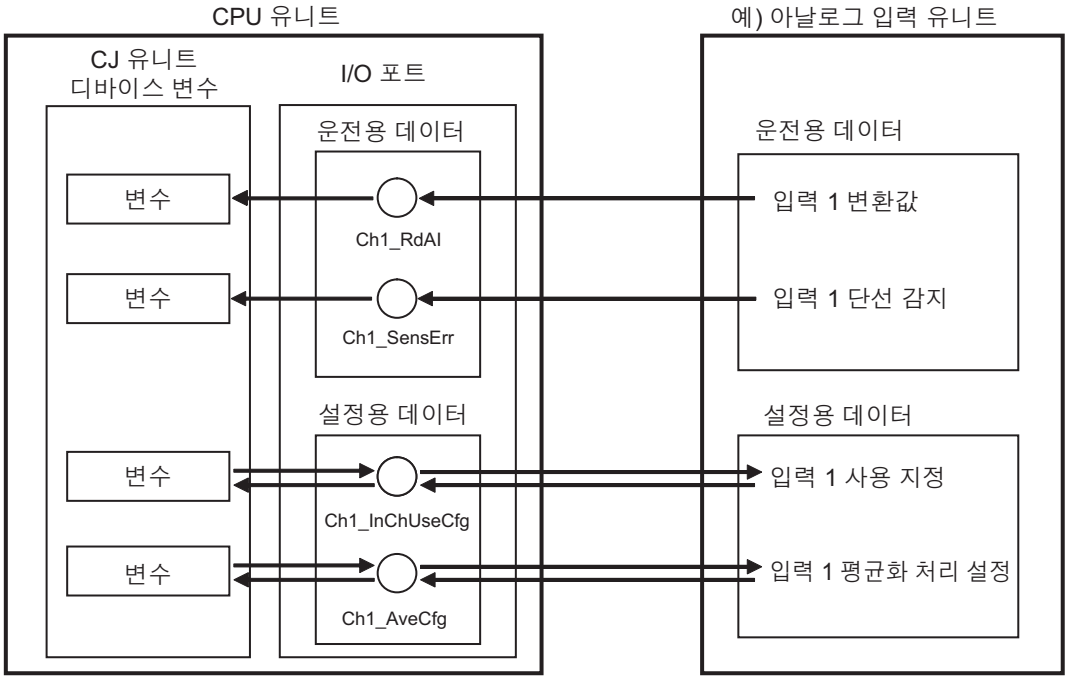
고기능 유니트와의 액세스

NJ 시리즈 CPU 유니트에서의 고기능 I/O 유니트와의 액세스 방법은 액세스 대상 데이터에 따라 2가지가 있습니다.

액세스 방법	대상 데이터
CJ 유니트 디바이스 변수에서 I/O 포트를 통해 액세스	• 운전용 데이터 • 설정용 데이터
사용자 정의 변수를 CJ 유니트용 메모리에 AT(할당대상) 지정하여 액세스	할당 임의 영역의 데이터

● CJ 유니트 디바이스 변수에서 I/O 포트를 통해 액세스

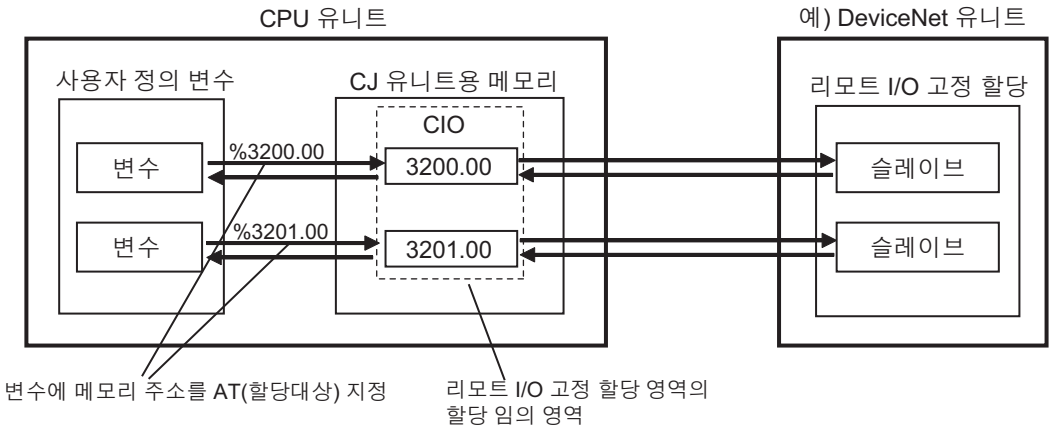
운전용 데이터 및 설정용 데이터와의 액세스는 CJ 유니트 디바이스 변수에서 I/O 포트를 통해 실행합니다. I/O 리프레시 시에 데이터 교환이 실행됩니다.



자세한 내용은 '3-3-1 I/O 포트(3-8페이지)'를 참조하십시오.

● 사용자 정의 변수를 CJ 유니트용 메모리에 AT(할당대상) 지정하여 액세스

할당 임의 영역과의 액세스는 사용자 정의 변수를 CJ 유니트용 메모리에 AT(할당대상) 지정하여 수행합니다. CJ 유니트용 메모리는 I/O 리프레시 시에 CJ 유니트와 데이터 교환이 실행됩니다.



할당 임의 영역에는 다음과 같은 영역이 있습니다.

- DeviceNet 유니트 사용 시의 고정 할당 영역
- DeviceNet 유니트 사용 시 및 CompoNet 마스터 유니트(통신 모드 번호 8) 사용 시의 CX-Integrator 에서 지정한 자유 할당 영역
- 고속 카운터 유니트 사용 시의 확장 메모리
- 프로세스 입출력 유니트 사용 시의 확장 메모리

CJ 유니트용 메모리에 대해서는 🕮 'A-9 CJ 유니트용 메모리의 세부 내용(A-180페이지)'을 참조하십시오. AT(할당대상) 지정에 대해서는 🕮 '6-3-8 변수의 각 속성의 설명(6-59페이지)'을 참조하십시오.

2-4 NX 버스 기능 모듈의 I/O 리프레시

여기에서는 NX102 및 NX1P2 CPU 유니트의 NX 버스 기능 모듈에서의 I/O 리프레시에 대해 설명합니다.

NX102 및 NX1P2 CPU 유니트의 NX 버스 기능 모듈은 CPU 유니트 상의 NX 유니트와 주기적으로 데이터 교환을 수행합니다.
데이터 교환은 프라이머리 정주기 태스크의 I/O 리프레시로 실행됩니다. 따라서 I/O 리프레시의 주기는 프라이머리 정주기 태스크의 태스크 주기입니다.

사용상의 주의 사항

NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.

2-4-1 I/O 리프레시 방식

NX1P2 CPU 유니트에 장착한 NX 유니트의 I/O 리프레시 방식은 다음과 같습니다.

NX 유니트의 I/O 리프레시 방식	실제 동작
프리 런 리프레시 방식과 입출력 동기 리프레시 방식 둘 다 갖는 NX 유니트	입출력 동기 리프레시 방식으로 동작
프리 런 리프레시 방식과 입출력 동기 리프레시 방식과 태스크 주기 우선 리프레시 방식을 갖는 NX 유니트	
타임 스탬프 방식만 갖는 NX 유니트	타임 스탬프 방식으로 동작
프리 런 리프레시 방식만 갖는 NX 유니트	프리 런 리프레시 방식으로 동작

(주의) NX102 및 NX1P2 CPU 유니트에는 다른 I/O 리프레시 방식의 NX 유니트가 혼재될 수 있습니다.
(주의) NX102 및 NX1P2 CPU 유니트에서는 NX 유니트의 I/O 리프레시 방식을 변경할 수 없습니다.

NX102 및 NX1P2 CPU 유니트에서 실제로 동작하는 I/O 리프레시 방식의 개요는 다음과 같습니다.

I/O 리프레시 방식		개요
입출력 동기 리프레시 방식		여러 NX 유니트 사이에서 입력 전송 타이밍이나 출력 갱신 타이밍이 동기함과 동시에 그 간격이 일정한 I/O 리프레시 방식입니다.
타임 스탬프 방식		NX 버스의 리프레시 주기와 비동기로 NX 버스가 입력 변화의 DC 시각*1을 기록하거나, 지정한 DC 시각에 출력을 수행하는 I/O 리프레시 방식입니다. NX 유니트와 CPU 유니트의 주기적인 데이터 교환은 NX 버스의 리프레시 주기로 수행합니다.
	변화 시각 입력 방식	입력 변화한 DC 시각을 입력 유니트가 기록하는 I/O 리프레시 방식입니다.
	시각 지정 출력 방식	지정한 DC 시각에 출력 유니트가 출력을 갱신하는 I/O 리프레시 방식입니다.
프리 런 리프레시 방식		NX 버스의 리프레시 주기와 NX 유니트의 입출력 갱신 주기가 비동기인 I/O 리프레시 방식입니다.

*1. DC(Distributed Clock, 분산 클록)에 기초한 동기 동작 가능한 슬레이브 및 유니트에서 공유하는 시계에 의한 시각입니다.

NX 유니트 형식에 따라 사용할 수 있는 I/O 리프레시 방식이 다릅니다. 사용하는 I/O 리프레시 방식에 따라 NX 유니트를 선정하십시오.
각 NX 유니트가 갖는 I/O 리프레시 방식은 각 NX 유니트의 사용 설명서를 참조하십시오.

2-4-2 각 I/O 리프레시 방식의 동작

NX102 및 NX1P2 CPU 유니트에서의 각 I/O 리프레시 방식의 동작에 대해 설명합니다.
I/O 리프레시 방식의 자세한 동작은 NX 유니트의 종류에 좌우됩니다. 각 NX 유니트의 사용 설명서를 참조하십시오.

입출력 동기 리프레시 방식의 동작

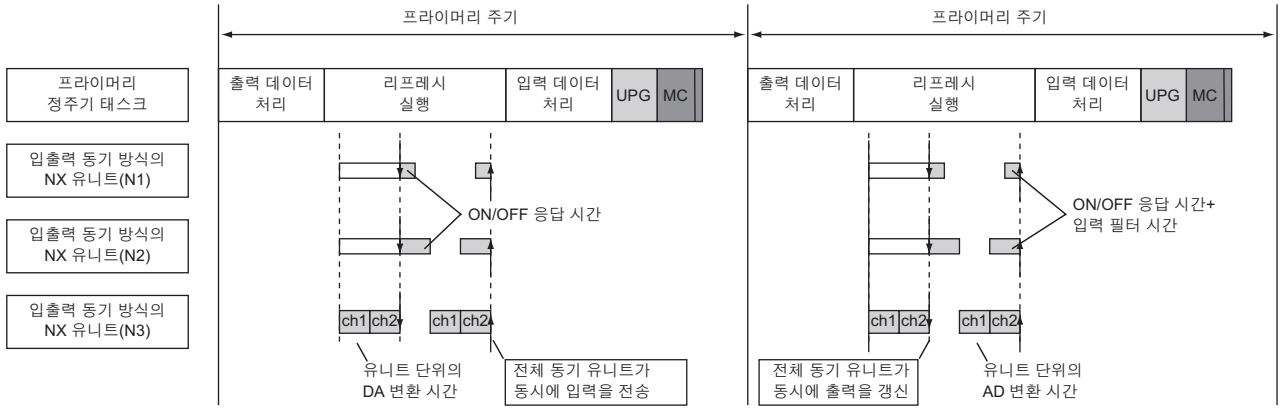
NX102 및 NX1P2 CPU 유니트 상의 입출력 동기 리프레시 방식의 모든 NX 유니트는 프라이머리 정주기 태스크의 I/O 리프레시와 동기한 타이밍에 동시 입력을 전송합니다. 또한, 마찬가지로 입력과 별도로 설정된 타이밍으로 동시 출력을 갱신합니다.

사용상의 주의 사항

CPU 유니트 상의 NX 유니트와 EtherCAT 슬레이브 및 EtherCAT 슬레이브 터미널이 입력을 전송하는 타이밍 및 출력을 갱신하는 타이밍은 동시가 될 수 없습니다.

● 입출력 동기 리프레시 방식의 동작 예

입출력 동기 리프레시 방식의 모든 NX 유니트는 동시 입력을 전송합니다.
입출력 동기 리프레시 방식의 모든 NX 유니트는 동시 출력을 갱신합니다.



타임 스탬프 방식의 동작

타임 스탬프 방식의 각 동작에 대해 설명합니다.

● 변화 시각 입력 방식의 동작

NX102 및 NX1P2 CPU 유니트 상의 변화 시각 입력 방식의 모든 NX 유니트는 입력이 변화한 DC 시각(이후, 변화 시각)을 기록합니다. 그 후, 프라이머리 정주기 태스크의 I/O 리프레시와 동기한 타이밍에 변화 시각을 전송합니다. 입력값은 항상 최신 값을 전송합니다.

참고

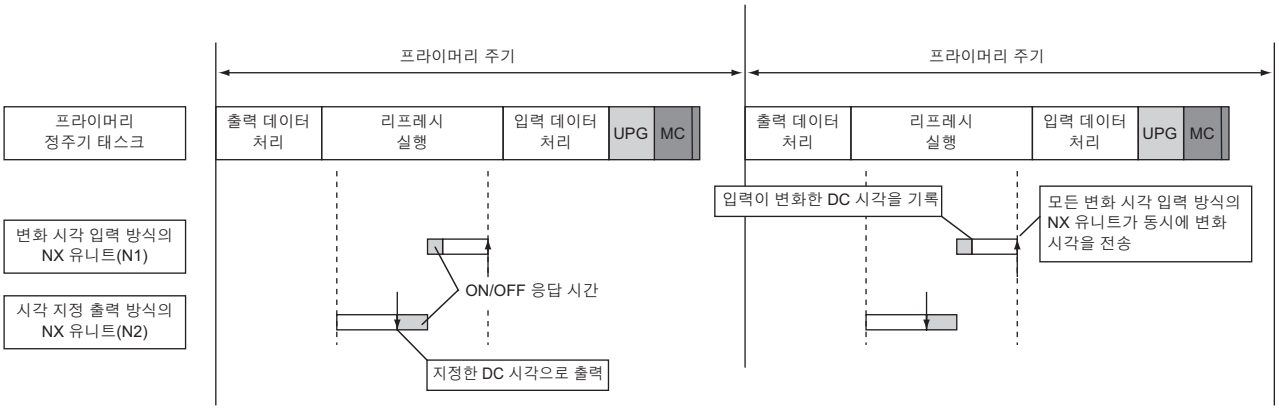
변화 시각을 전송하는 타이밍은 입력 동기 리프레시 방식의 NX 유니트가 입력을 전송하는 타이밍과 같습니다.

● 시각 지정 출력 방식의 동작

NX102 및 NX1P2 CPU 유니트 상의 시각 지정 출력 방식의 모든 NX 유니트는 NX 유니트별로 지정한 DC 시각에 출력을 갱신합니다.

● 타임 스탬프 방식의 동작 예

변화 시각 입력 방식의 모든 NX 유니트는 입력이 변화한 DC 시각을 NX 유니트별로 기록하고, 동일한 타이밍에 변화 시각을 전송합니다.
시각 지정 출력 방식의 모든 NX 유니트는 NX 유니트별로 지정한 DC 시각에 출력을 갱신합니다.

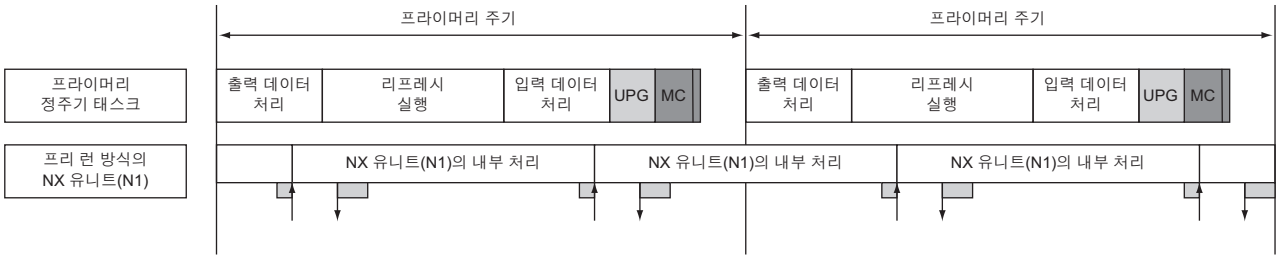


프리 런 리프레시 방식의 동작

프리 런 리프레시 방식에서는 NX 버스의 리프레시 주기와 NX 유니트의 입출력 갱신 주기가 비동기로 동작합니다.

● 프리 런 리프레시 방식의 동작 예

NX102 및 NX1P2 CPU 유니트에서의 프리 런 리프레시 방식의 동작 예는 다음과 같습니다.
프리 런 방식의 NX 유니트는 프라이머리 정주기 태스크의 I/O 리프레시와는 비동기인 유니트의 독자적인 타이밍에 입출력 처리합니다.

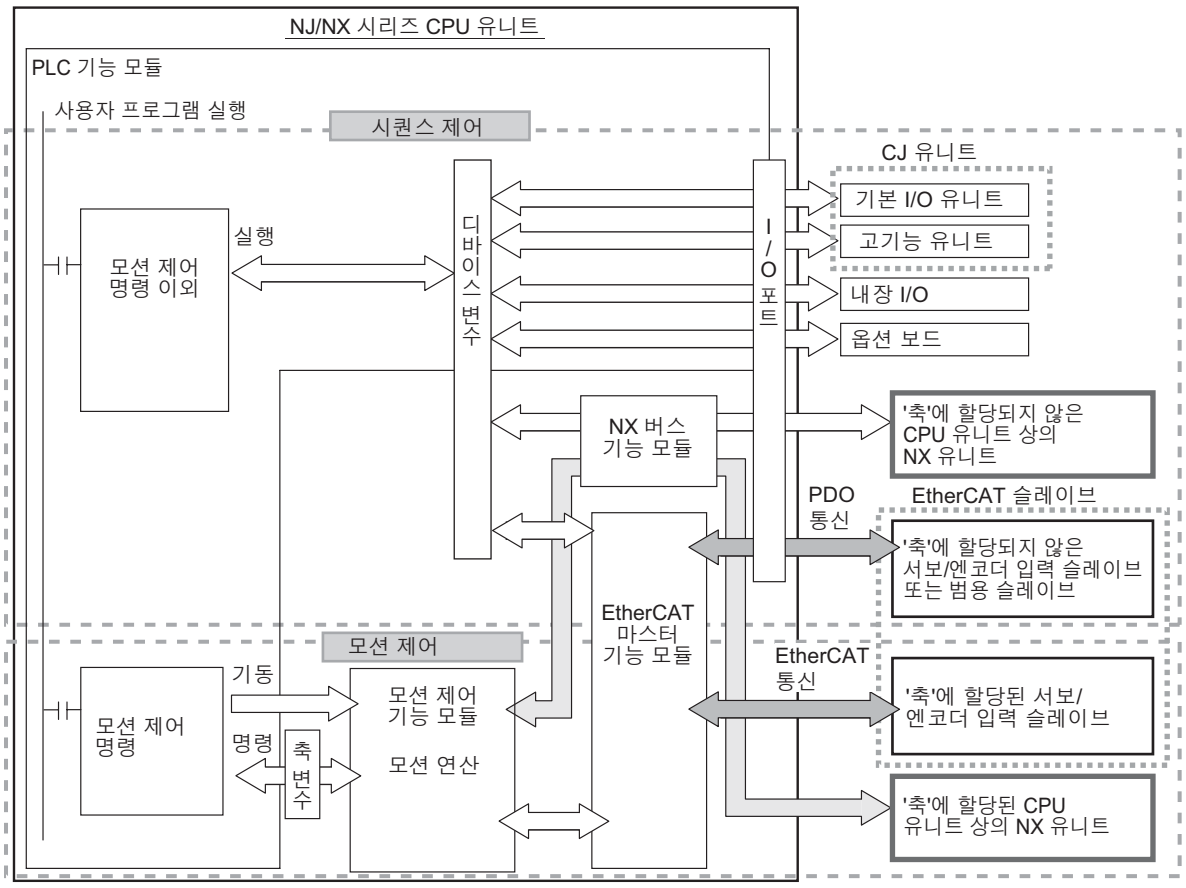


2-5 시퀀스 제어와 모션 제어의 구조

여기에서는 CPU 유니트의 시퀀스 제어와 모션 제어의 구조에 대해 설명합니다.

2-5-1 제어의 개요

NJ/NX 시리즈 CPU 유니트에서는 시퀀스 제어와 모션 제어를 실행할 수 있습니다.
시퀀스 제어는 사용자 프로그램에 기술된 모션 제어 명령 이외를 실행하고, EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 옵션 보드 및 CJ 유니트를 제어합니다. 제어에는 PLC 기능 모듈, NX 버스 기능 모듈 및 EtherCAT 마스터 기능 모듈을 사용합니다.
모션 제어는 사용자 프로그램에 기술된 모션 제어 명령을 실행하고, '축'에 할당된 EtherCAT 상의 서보 드라이버/엔코더 입력 슬레이브를 제어합니다. 제어에는 PLC 기능 모듈, 모션 제어 기능 모듈, NX 버스 기능 모듈 및 EtherCAT 마스터 기능 모듈을 사용합니다.



(주의) CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
(주의) NX 버스 기능 모듈을 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
(주의) 내장 I/O 및 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.

참고

제어의 구조에서 본 명령의 분류

제어의 구조상, 명령에는 크게 나눠 다음 2종류가 있습니다.

명령의 분류	정의
모션 제어 명령 이외 (시퀀스 제어)	PLC 기능 모듈 내의 사용자 프로그램 실행으로 처리가 완결되는 명령
모션 제어 명령	PLC 기능 모듈 내의 사용자 프로그램 실행에 따라 기동되고, 모션 제어 기능 모듈에 대해 지시를 내리는 명령 예) MC_Home(원점 복귀), MC_Move(위치 결정), MC_CamIn(캠 동작 시작) 등의 모션 제어 관련 각종 조작

모션 제어 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오. 모션 제어 명령 이외의 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

버전 관련 정보

Sysmac Studio Ver.1.09 이상에서는 '축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당할 수 있습니다.

디바이스 변수를 할당 가능한 I/O 포트는 다음 중 하나의 조건을 충족하는 것입니다.

- R/W 속성의 값이 R(읽기 전용)일 것
- R/W 속성의 값이 W(쓰기 전용)이고, Sysmac Studio의 [축 기본 설정] 화면의 [세부 내용 설정]에서 프로세스 데이터의 설정이 <할당 안 함>일 것

사용상의 주의 사항

'축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당하는 경우에 다음 조작을 실행하면 디바이스 변수의 할당이 해제되므로 주의하십시오.

(1) Sysmac Studio Ver.1.09 이상이고, '축'에 할당된 서보/엔코더 입력 슬레이브의 I/O 포트에 디바이스 변수를 할당

(2) 프로젝트 데이터 저장

(3) 해당 프로젝트 데이터를 Sysmac Studio Ver.1.08 이하에서 열기

2-5-2 시퀀스 제어의 구조

시퀀스 제어의 제어 대상 장치에 따라 구조가 다릅니다. 다음과 같이 시퀀스 제어의 구조로 기능 모듈의 동작, 제어 주기를 나타냅니다.

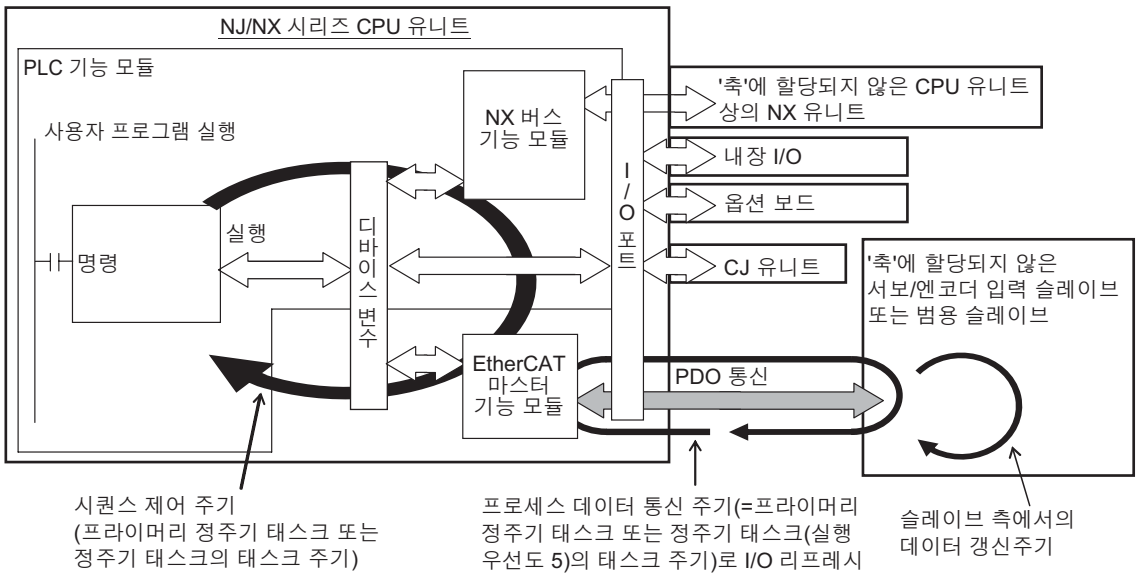
장치	시퀀스 제어의 구조	
	기능 모듈의 동작	제어 주기
'축'에 할당하지 않은 서보/엔코더 입력 슬레이브 *1 또는 범용 슬레이브	• PLC 기능 모듈은 사용자 프로그램을 실행하고, 디바이스 변수를 갱신합니다. • EtherCAT 마스터 기능 모듈은 디바이스 변수를 사용하여 I/O 포트를 통해 슬레이브와 데이터 교환합니다.	프로그램을 할당한 태스크의 태스크 주기(프라이머리 정주기 태스크 또는 정주기 태스크의 태스크 주기)*2
'축'에 할당하지 않은 CPU 유니트 상의 NX 유니트*3	• PLC 기능 모듈은 사용자 프로그램을 실행하고, 디바이스 변수를 갱신합니다. • NX 버스 기능 모듈은 디바이스 변수를 사용하여 I/O 포트를 통해 CPU 유니트 상의 NX 유니트와 데이터 교환합니다.	프라이머리 정주기 태스크*4
내장 I/O*5	PLC 기능 모듈에 의해 사용자 프로그램의 실행, 디바이스 변수의 갱신, 내장 I/O와의 데이터 교환을 수행합니다.	프라이머리 정주기 태스크
옵션 보드*6	PLC 기능 모듈에 의해 사용자 프로그램의 실행, 디바이스 변수의 갱신, 옵션 보드와의 데이터 교환을 수행합니다.	프라이머리 정주기 태스크*7
CJ 유니트*8	PLC 기능 모듈에 의해 사용자 프로그램의 실행, 디바이스 변수의 갱신, CJ 유니트와의 데이터 교환을 수행합니다.	프로그램을 할당한 태스크의 태스크 주기(프라이머리 정주기 태스크 또는 정주기 태스크의 태스크 주기)*9

*1. Sysmac Studio Ver.1.09 이상에서는 '축'에 할당된 서보/엔코더 입력 슬레이브도 I/O 포트에 디바이스 변수를 할당하면 시퀀스 제어를 할 수 있습니다.
*2. 슬레이브 측의 데이터 갱신 주기는 슬레이브의 설정에 좌우됩니다.
*3. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다. '축'에 할당된 CPU 유니트 상의 NX 유니트에 대해서도 I/O 포트에 디바이스 변수를 할당하면 시퀀스 제어할 수 있습니다.
*4. 프리 런 리프레시 방식으로 동작하는 NX 유니트의 입출력 갱신 주기와 CPU 유니트의 제어 주기는 비동기입니다.
*5. 내장 I/O를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*6. 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*7. 프라이머리 정주기 태스크와 옵션 보드 서비스의 데이터 교환 주기는 비동기입니다.
*8. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*9. CJ 유니트와의 데이터 교환의 주기는 CJ 유니트의 I/O 리프레시를 할당한 태스크의 주기입니다.

'축'에 할당하지 않은 서보/엔코더 입력 슬레이브 또는 범용 슬레이브는 프로세스 데이터 통신 주기 즉, 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 태스크 주기로 I/O 리프레시됩니다. 그러나 사용자 프로그램의 실행 및 디바이스 변수의 갱신은 프로그램을 할당한 태스크의 태스크 주기로 실행됩니다. 따라서 슬레이브의 값이 디바이스 변수에 반영되어 제어되는 것은 프로그램을 할당한 태스크의 태스크 주기가 됩니다.

슬레이브를 프로세스 데이터 통신 주기로 제어하고자 하는 경우에 슬레이브를 제어하는 프로그램을 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)에 할당해야 합니다.

자세한 내용은 📖'5-11-3 시스템으로서의 입출력 응답 시간(5-112페이지)'을 참조하십시오.



참고

- 정주기 태스크(실행 우선도 5)를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
- NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에서 프로세스 데이터 통신을 실행할 수 있습니다. 2가지 프로세스 데이터 통신 주기를 구별해야 하는 경우에 프라이머리 정주기 태스크의 통신 주기를 '프로세스 데이터 통신 주기 1', 정주기 태스크(실행 우선도 5)의 통신 주기를 '프로세스 데이터 통신 주기 2'라고 합니다.
- NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 프라이머리 정주기 태스크에서만 프로세스 데이터 통신을 수행합니다.
- CPU 유니트 상의 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
- 내장 I/O 및 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
- CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

2-5-3 모션 제어의 구조

모션 제어의 구조로 기능 모듈의 동작, 제어 주기를 나타냅니다.

● 기능 모듈의 동작

- PLC 기능 모듈은 사용자 프로그램 상의 모션 제어 명령을 기동하고, 모션 제어 기능 모듈에 대해 모션 제어의 명령을 전송합니다. 이 명령에는 축 변수를 사용합니다.
- 모션 제어 기능 모듈은 PLC 기능 모듈의 명령을 바탕으로 모션 연산을 실행하고, 연산 결과의 명령값으로 축 변수의 값을 갱신합니다.
- EtherCAT 마스터 기능 모듈은 축 변수의 명령값을 서보 드라이버 등에 EtherCAT 통신으로 송신합니다.
- NX 버스 기능 모듈은 축 변수의 명령값을 CPU 유니트 상의 NX 시리즈 펄스 출력 유니트 등에 출력합니다.

● 제어 주기

모션 제어 주기는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 태스크 주기입니다.

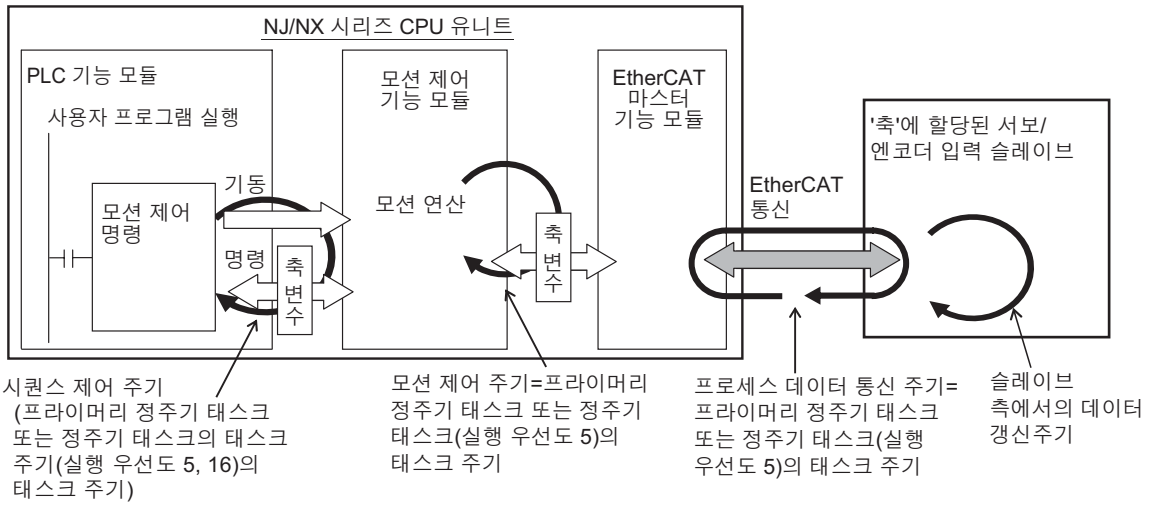
모션 제어 기능 모듈의 모션 연산 주기는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 태스크 주기로 실행됩니다. 또한, 제어 대상 '축'에 할당된 서보/엔코더 입력 슬레이브와는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 프로세스 데이터 통신 주기로 데이터 교환합니다. 프로세스 데이터 통신 주기는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)에 동기합니다.

그리고 NX102 및 NX1P2 CPU 유니트에서는 제어 대상 '축'에 할당된 CPU 유니트 상의 NX 유니트와 프라이머리 정주기 태스크에서 데이터 교환합니다.

이것으로 모션 제어 주기는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 태스크 주기가 되고, 여러 축을 완전히 동기하여 동작시킬 수 있습니다.

단, 다음과 같은 제약이 있습니다.

- 모션 제어 명령의 기동 및 모션 제어의 명령의 전송은 시퀀스 제어 주기로 실행됩니다.
- EtherCAT 슬레이브 측의 데이터 갱신 주기는 슬레이브의 설정에 좌우됩니다.



사용상의 주의 사항

- 정주기 태스크(실행 우선도 5)를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
- NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)로 모션 제어를 실행할 수 있습니다. 2가지 모션 제어를 구별해야 하는 경우에 프라이머리 정주기 태스크의 모션 제어를 '모션 제어 1', 정주기 태스크(실행 우선도 5)의 모션 제어를 '모션 제어 2'라고 합니다.
- NX102 및 NX1P2 CPU 유니트와 NJ 시리즈 CPU 유니트에서는 프라이머리 정주기 태스크에서만 모션 제어를 실행합니다.

참고

- 모션 제어 기능 모듈에서 제어하는 EtherCAT 슬레이브 및 NX102 및 NX1P2 CPU 유니트 상의 NX 유니트는 Sysmac Studio에서 '축'으로의 할당 설정을 실행합니다. 이를 통해 PLC 기능 모듈은 사용자 프로그램의 '모션 제어 명령'를 사용하여 모션 제어 기능 모듈로 명령을 출력할 수 있습니다. 그리고 모션 제어 기능 모듈에서 축 변수를 통해 정보를 취득할 수 있습니다.
- 모션 제어 명령이 기술된 프로그램을 할당하는 태스크의 종류에 따라 모션 제어 시스템으로서의 입출력 응답 시간이 다릅니다. 자세한 내용은 '5-11-3 시스템으로서의 입출력 응답 시간 (5-112페이지)'을 참조하십시오.

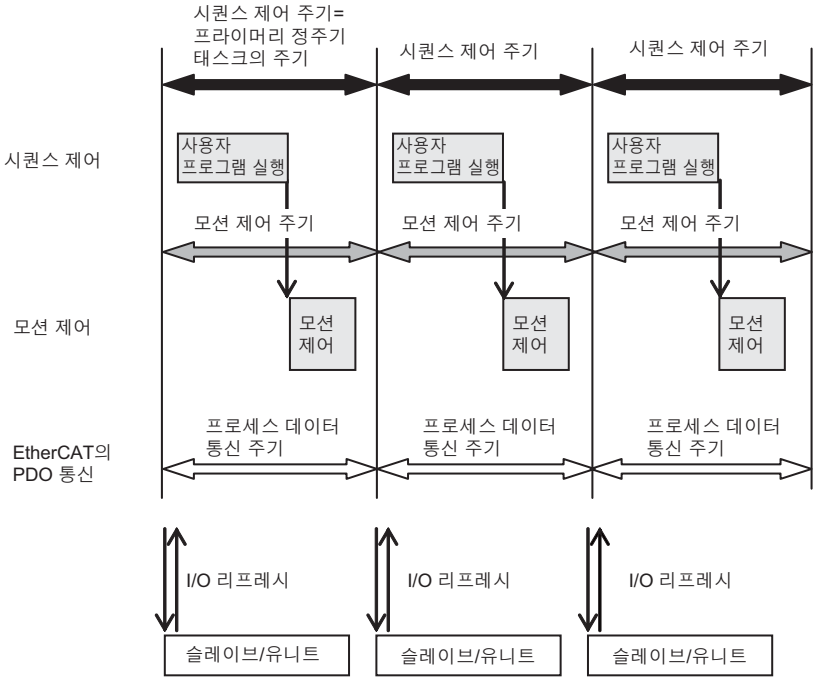
2-5-4 시퀀스 제어와 모션 제어의 동기

시퀀스 제어의 제어 주기는 프로그램 및 I/O 리프레시를 할당한 태스크의 태스크 주기입니다. 한편, 모션 제어는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)의 태스크 주기로 상시 실행됩니다. 또한, 모션 제어에 사용하는 EtherCAT 슬레이브의 프로세스 데이터 통신 주기는 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)에 동기합니다.

NX701 CPU 유니트에서는 축별 및 축 그룹별로 모션 연산을 실행하는 모션 제어를 설정할 수 있습니다. 모션 제어 1을 설정한 축 및 축 그룹은 프라이머리 정주기 태스크의 주기로 모션 제어와 프로세스 데이터 통신이 실행됩니다. 모션 제어 2를 설정한 축 및 축 그룹은 정주기 태스크(실행 우선도 5)의 주기로 모션 제어와 프로세스 데이터 통신이 실행됩니다.

NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 프라이머리 정주기 태스크의 주기로 축 및 축 그룹의 모션 제어와 프로세스 데이터 통신이 실행됩니다.

따라서 모션 제어가 실행되는 태스크에 시퀀스 제어의 프로그램을 할당하면 시퀀스 제어와 모션 제어의 제어 주기 및 EtherCAT의 프로세스 데이터 통신 주기를 동기시킬 수 있습니다.
프로그램을 프라이머리 정주기 태스크에 할당한 경우의 그림은 다음과 같습니다. 다음 그림에서는 시퀀스 제어 주기, 모션 제어 주기, EtherCAT의 프로세스 데이터 통신 주기가 모두 동기합니다.



참고

- NX701 CPU 유니트에서의 모션 제어, 태스크, 프로세스 데이터 통신 주기의 관계

모션 제어	실행하는 태스크	프로세스 데이터 통신 주기
모션 제어 1	프라이머리 정주기 태스크	프로세스 데이터 통신 주기 1
모션 제어 2	정주기 태스크(실행 우선도 5)	프로세스 데이터 통신 주기 2

- NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서의 모션 제어, 태스크, 프로세스 데이터 통신 주기의 관계

모션 제어	실행하는 태스크	프로세스 데이터 통신 주기
모션 제어	프라이머리 정주기 태스크	프로세스 데이터 통신 주기

2-6 CPU 유니트 내의 데이터 개요

CPU 유니트에는 각종 설정, 사용자 프로그램, 현재값 등의 데이터가 있습니다. 주요 데이터와 내용은 다음과 같습니다.

CPU 유니트의 모든 데이터에 대해서는 'A-8 CPU 유니트 내 데이터 속성(A-176페이지)'을 참조하십시오.

데이터 종류			내용
각종 설정	EtherCAT 구성	EtherCAT 슬레이브 구성	EtherCAT 슬레이브의 구성 정보
		EtherCAT 마스터 설정	EtherCAT 마스터의 통신 주기 등의 파라메터 설정
	유니트 구성·유니트 설정		CPU 유니트가 인식하기 위한 유니트의 구성 정보 및 각 유니트의 초기 설정 정보
	I/O 맵		유니트 구성에서 자동 생성된 I/O 포트의 변수로의 할당 정보
	컨트롤러 설정	동작 설정	전원 투입 시의 동작 모드, 보안 등의 설정 및 시스템 서비스 설정
		내장 EtherNet/IP 포트 설정	다음의 EtherNet/IP 포트 설정 TCP/IP 설정, Ethernet 설정, DHCP 설정, DNS 설정, FTP 설정, NTP 설정, SNMP 설정
	모션 제어 설정		'축'/'축 그룹'의 축 변수/축 그룹 변수 및 모션 제어 파라메터 설정
	캠 데이터 설정		모션 제어 명령의 캠 동작에서 사용하는 위상/변위 데이터로 이루어지는 캠 테이블
	이벤트 설정		사용자 정의의 이상/정보의 설정
	태스크 설정		태스크의 종류 및 수, 태스크의 실행 조건, 태스크 이름, 태스크 내에서 실행하는 프로그램 등의 설정
	데이터 트레이스 설정		데이터 트레이스의 트리거 조건 등의 설정
	태그 데이터 링크 테이블		EtherNet/IP에서의 태그 데이터 링크 설정
	컨트롤러 이름		CPU 유니트의 이름
	조작 권한 인증		Sysmac Studio가 CPU 유니트에 대해 조작하기 위한 권한 비밀번호
	내장 시계	설정 시각	CPU 유니트 내장 시계의 시각
		표준 시간대 설정	CPU 유니트 내장 시계의 표준 시간대 설정
사용자 프로그램	POU(프로그램 구성 단위)		프로그램, 펑션, 펑션 블록의 정의 로컬 변수 테이블 및 그 초기값을 포함
	데이터	데이터 형식	데이터 형식의 정의
		글로벌 변수	글로벌 변수의 속성 정보 초기값 속성 및 유지 속성 등
현재값	변수의 값		변수의 값
	CJ 유니트용 메모리의 값*1		CJ 유니트용 메모리 안의 CIO, WR, HR, DM, EM 영역의 값
기타	이벤트 로그		컨트롤러의 이상 이력 및 전원 ON/OFF와 운전 시작과 같은 이상 이외 사건의 이력
	앱솔루트 엔코더 원점 위치 오프셋		모션 제어에서의 앱솔루트 엔코더 장착 서보 드라이버의 현재 위치를 복원하기 위한 원점 확정 후의 명령 위치와 앱솔루트 엔코더에서 읽은 절대값 데이터의 차

*1. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.

2-7 CPU 유니트의 각 상태의 동작

CPU 유니트의 각 상태의 사용자 프로그램 실행, I/O 리프레시, 외부와의 통신 처리 동작을 설명합니다. 그리고 CPU 유니트의 상태 중 하나인 '정상 운전 중'에서 사용자 프로그램의 실행 상태를 전환하는 동작 모드에 대해서도 설명합니다.

2-7-1 CPU 유니트의 상태

CPU 유니트에는 '기동 중', '정상 운전 중', '이상 상태' 3가지 상태가 있습니다. 각 상태의 정의는 다음과 같습니다.

상태	정의
기동 중	소프트웨어가 기동 처리를 실행하고 있는 상태
정상 운전 중	태스크로 실행되는 처리 또는 시스템 서비스를 소프트웨어가 실행하고, 컨트롤러 이상이 발생하지 않은 상태
이상 상태	태스크로 실행되는 처리 또는 시스템 서비스를 소프트웨어가 실행하고, 컨트롤러 이상이 발생한 상태

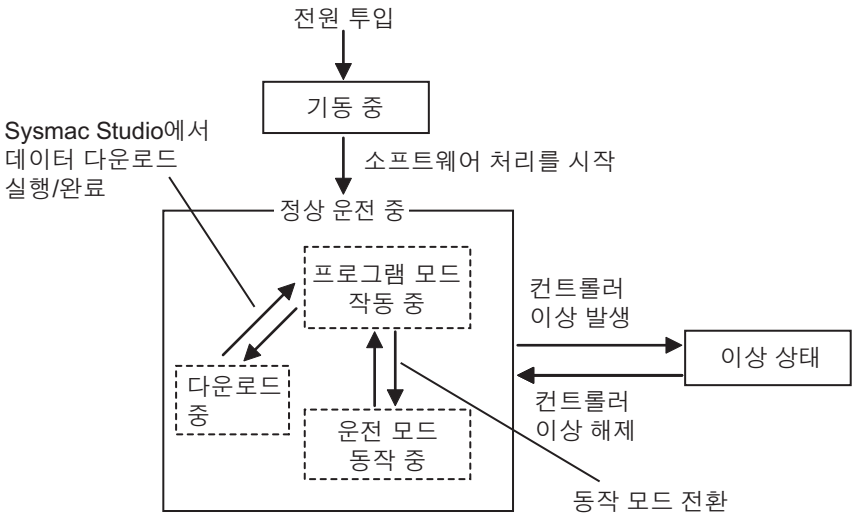
그리고 '정상 운전 중'에는 '프로그램 모드 동작 중', '운전 모드 동작 중', '다운로드 중' 3가지 상태가 있습니다. '정상 운전 중'인 CPU 유니트는 사용자의 조작에 의해 각 상태로 이동합니다. 각 상태의 정의는 다음과 같습니다.

상태	정의
프로그램 모드 동작 중	동작 모드가 프로그램 모드인 상태
운전 모드 동작 중	동작 모드가 운전 모드인 상태
다운로드 중	Sysmac Studio에서 데이터의 다운로드를 실행하고 있는 상태

(주의) 프로그램 모드와 운전 모드에 대해서는 '2-7-3 동작 모드(2-40페이지)'를 참조하십시오.

● CPU 유니트의 상태 이동

전원 투입 후에 CPU 유니트는 '기동 중'이 됩니다. CPU 유니트가 '기동 중'이 되고 10~20초 경과 후에 소프트웨어 처리를 시작하고, '정상 운전 중'이 됩니다. '정상 운전 중'에 컨트롤러 이상이 발생하면 '이상 상태'가 됩니다. 컨트롤러 이상이 해제되면 '정상 운전 중'으로 돌아갑니다.
'기동 중' 상태에서 '정상 운전 중'이 되면 CPU 유니트는 컨트롤러 설정에 설정된 동작 모드에 따라 '프로그램 모드 동작 중' 또는 '운전 모드 동작 중'이 됩니다. 다음으로 동작 모드의 전환에 의해 '프로그램 모드 동작 중'인 경우에는 '운전 모드 동작 중', '운전 모드 동작 중'인 경우에는 '프로그램 모드 동작 중'이 됩니다. '프로그램 모드 동작 중'에 Sysmac Studio에서 데이터의 다운로드를 실행하면 '다운로드 중'이 됩니다. 다운로드가 완료되면 '프로그램 모드 동작 중'으로 돌아갑니다.



참고

- CPU 유니트의 상태는 CPU 유니트 전면의 상태 표시 LED를 통해 확인할 수 있습니다. 상태 표시 LED에 의한 확인 방법은 '12-1 이상의 확인과 처리의 개요(12-2페이지)'를 참조하십시오.
- CPU 유니트의 상태가 변화할 때의 데이터의 동작에 대해서는 'A-8 CPU 유니트 내 데이터 속성(A-176페이지)'을 참조하십시오.
- CPU 유니트의 상태가 변화할 때의 변수의 값에 대해서는 '6-3-9 상태 변화에 따른 변수 값의 변화(6-67페이지)'를 참조하십시오.

2-7-2 각 상태의 동작

CPU 유니트의 상태에 따라 사용자 프로그램, I/O 리프레시, 외부와의 통신 동작이 변화합니다. '기동 중' 및 '정상 운전 중'일 때의 각 처리 동작은 다음과 같습니다.

'이상 상태'에 대해서는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"의 'CPU 유니트가 동작할 수 있는 이상'을 참조하십시오.

CPU 유니트의 처리	기동 중의 동작	정상 운전 중의 동작		
		프로그램모드 동작 중	운전 모드 동작 중	다운로드 중
사용자 프로그램	정지합니다.	정지합니다.	실행합니다.	정지합니다.
EtherCAT 슬레이브의 I/O 리프레시	정지합니다.	실행합니다.		EtherCAT 통신이 Safe-Operational 상태가 됩니다. *1 *2
CPU 유니트 상의 NX 유니트, 내장 I/O, 아날로그 입출력용 옵션 보드의 각 I/O 리프레시	정지합니다.	실행합니다.		정지합니다. *2
CJ 유니트의 I/O 리프레시	정지합니다.	실행합니다.		
외부와의 통신	정지합니다.	실행합니다.		실행합니다.*3

*1. 입력값만 갱신됩니다.
*2. 시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'로 장치 출력 유지 설정을 유효(16#A5A5)로 설정한 경우에 I/O 리프레시를 실행합니다. 장치 출력 유지 설정에 대해서는 '장치 출력 유지 설정(6-70페이지)'을 참조하십시오. 시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'는 CPU 유니트Ver.1.13 이상과 Sysmac Studio Ver.1.17 이상의 조합으로 사용할 수 있습니다.
*3. 태그 데이터 링크는 계속하지만 태그 데이터 링크의 값은 갱신되지 않습니다. 출력 태그의 값은 다운로드 시작 전의 값으로 유지됩니다. 입력 태그의 값은 변수에 반영되지 않습니다.

● I/O의 출력 접점의 값

I/O 리프레시 처리에 의한 각 상태의 I/O의 출력 접점의 값은 다음과 같습니다.

I/O의 출력 접점	기동 중의 동작	정상 운전 중의 동작		
		프로그램모드 동작 중	운전 모드 동작 중	다운로드 중
EtherCAT 슬레이브의 출력 접점	각 슬레이브의 설정에 따릅니다.*1	EtherCAT 슬레이브 디바이스 변수의 값이 됩니다.		각 슬레이브의 설정에 따릅니다.*2*3
CPU 유니트 상의 NX 유니트의 출력	각 유니트의 초기 값이 됩니다.	NX 유니트 디바이스 변수의 값이 됩니다.		각 유니트의 설정에 따릅니다. *3
내장 I/O의 출력 접점	OFF가 됩니다.	내장 I/O 디바이스 변수의 값이 됩니다.		OFF가 됩니다. *3
아날로그 입출력용 옵션 보드의 출력	OFF가 됩니다.	옵션 보드 디바이스 변수의 값이 됩니다.		OFF가 됩니다. *3
CJ 시리즈 기본 출력 유니트의 출력 접점	OFF가 됩니다.	CJ 유니트 디바이스 변수의 값이 됩니다. *3*4		

*1. 전원 투입 후에 EtherCAT 통신이 시작될 때까지의 슬레이브 설정에 대해서는 각 슬레이브의 설명서를 참조하십시오.
*2. 다운로드 완료 후에 EtherCAT 슬레이브는 초기화 처리를 수행합니다. 초기화 처리 중의 출력은 각 슬레이브의 설정에 따릅니다.
*3. 시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'로 장치 출력 유지 설정을 유효(16#A5A5)로 설정한 경우에 동작 모드 변경 시 및 다운로드 시에도 장치 출력을 유지합니다. 자세한 내용은 '6-3-9 상태 변화에 따른 변수 값의 변화(6-67페이지)'를 참조하십시오.
시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'는 CPU 유니트Ver.1.13 이상과 Sysmac Studio Ver.1.17 이상의 조합으로 사용할 수 있습니다.
*4. 단, 다운로드 완료 시 및 동작 모드 전환 시 CJ 유니트 디바이스 변수의 값은 초기값 속성에 따라 초기화됩니다.

변수의 초기값 속성에 대해서는 '6-3-8 변수의 각 속성의 설명(6-59페이지)'을 참조하십시오.
장치 출력 유지 설정의 자세한 내용은 '장치 출력 유지 설정(6-70페이지)'을 참조하십시오.



사용상의 주의 사항

- CPU 유니트 상의 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
- 내장 I/O 및 아날로그 입출력용 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
- CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.



참고

동작 모드 전환에 의한 서보 드라이버의 동작

모션 제어 중에 운전 모드에서 프로그램 모드로 전환된 경우에 각 축 파라메터에 설정된 최대 감속도로 감속 정지합니다.

EtherCAT 슬레이브의 초기화 처리 중의 동작 모드 전환

EtherCAT 슬레이브의 초기화 처리 중에 CPU 유니트의 동작 모드를 운전 모드로 변경할 수 있습니다. 이 경우에 통신이 확립되었는지 확인한 후에 슬레이브의 데이터를 제어에 사용하도록 프로그래밍하십시오. 사용자 프로그램에서는 시스템 정의 변수 '_EC_PDSlavTbl'(프로세스 데이터 통신 중 슬레이브 테이블)을 사용하여 전체 슬레이브의 프로세스 데이터의 출력과 입력이 유효인지 여부를 확인할 수 있습니다.

2-7-3 동작 모드

동작 모드는 동작 확인 및 본 운전 등의 목적에 맞춰서 전환하는 모드입니다. 동작 모드에는 목적에 맞춰서 운전 모드와 프로그램 모드가 있습니다. 각 동작 모드에서는 사용자 프로그램의 실행 상태가 다릅니다. 각 동작 모드의 목적과 사용자 프로그램의 실행 상태는 다음과 같습니다.

동작 모드	목적	사용자 프로그램의 실행 상태
운전 모드*1	사용자 프로그램을 실행시키는 시운전 및 본 운전	실행합니다.
프로그램 모드	I/O 배선 확인 등의 사용자 프로그램을 실행시키지 않고 실행하는 동작 확인	실행하지 않습니다.

*1. 초기 설정에서는 CPU 유니트가 '기동 중'에서 '정상 운전 중'이 되면 운전 모드로 동작합니다.

참고

동작 모드 전환에 의해 축 정지 및 모션 제어 명령 중단 등을 수행합니다. 동작 모드 전환 시의 모션 제어 기능 모듈의 동작은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

각 동작 모드의 Sysmac Studio 및 표시기에서의 조작 가능 여부

각 동작 모드의 Sysmac Studio 및 표시기에서의 주요 조작 가능 여부는 다음과 같습니다.

조작		운전 모드	프로그램 모드
Sysmac Studio	동기	불가능	가능
	온라인 편집	가능	
	강제값 리프레시	가능	
	변수 또는 CJ 유니트용 메모리*1 의 값 변경	가능	
표시기	변수 또는 CJ 유니트용 메모리*1의 값 변경	가능	

*1. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.

동작 모드 전환 시의 변수 값 유지

운전 모드에서 프로그램 모드 또는 프로그램 모드에서 운전 모드로 전환하면 변수의 유지 속성에 따라 변수값이 다음과 같아집니다.

변수의 유지 속성	변수의 값
비유지	초기값 설정이 있는 경우: 지정한 초기값 초기값 설정이 없는 경우: 시스템 정의의 초기값*1
유지	동작 모드 변경 전의 값을 유지

*1. 변수의 데이터 형식에 따라 다릅니다. '초기값 지정이 공백인 경우(6-63페이지)'를 참조하십시오.

CPU 유니트의 상태가 변화할 때의 변수의 값에 대해서는 '6-3-9 상태 변화에 따른 변수 값의 변화(6-67페이지)'를 참조하십시오.

동작 모드 설정 및 전환

전원 투입 후의 운전 시작 시에 CPU 유니트는 컨트롤러 설정에 따른 동작 모드에서 동작합니다. 또한, 정상 운전 중에는 사용자가 목적에 맞춰 동작 모드를 전환합니다. 설정 및 전환은 모두 Sysmac Studio에서 수행합니다.

● 전원 투입 후의 운전 시작 시의 동작 모드 설정

전원 투입 후의 운전 시작 시에는 '전원 투입 시 모드'로 설정한 동작 모드에서CPU 유니트가 동작합니다. 컨트롤러 설정의 '동작 설정'에 '전원 투입 시 모드'로 운전 모드 또는 프로그램 모드를 설정합니다. '전원 투입 시 모드'의 설정에 대해서는 '4-2-2 컨트롤러 설정(4-4페이지)'을 참조하십시오.

● 운전 중인 동작 모드 전환

동작 모드는 Sysmac Studio에 의해 전환할 수 있습니다. 메뉴 바의 [컨트롤러]-[동작 모드]에서 운전 모드 또는 프로그램 모드를 선택합니다.

안전상의 중요 사항

전원 투입 시의 동작 모드 설정 및 동작 모드의 전환이 설비에 영향을 주지 않는지 확인한 후 실행하십시오.

동작 모드 확인

동작 모드는 CPU 유니트의 RUN LED 또는 Sysmac Studio에서 확인할 수 있습니다.

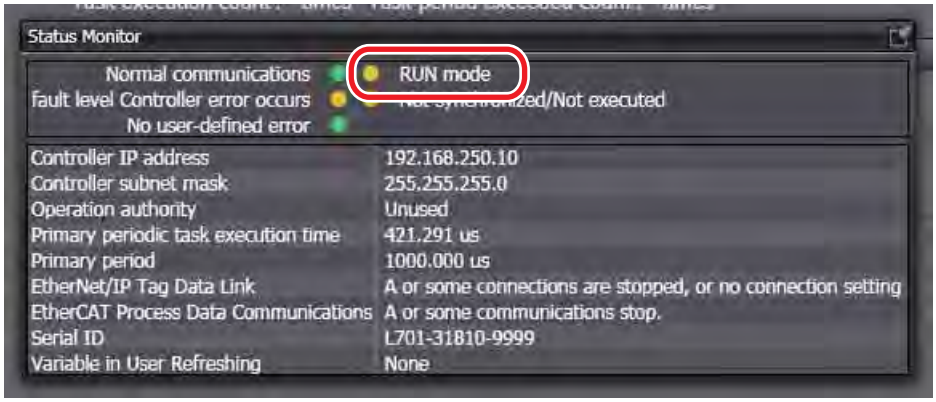
● RUN LED를 통해 확인

CPU 유니트의 RUN LED의 표시에 따라 동작 모드는 다음과 같습니다.

RUN LED의 표시	동작 모드
소등	프로그램 모드
점등	운전 모드

● Sysmac Studio에서 확인

동작 모드는 Sysmac Studio의 컨트롤러 Status 창에서 확인할 수 있습니다. 다음과 같은 컨트롤러 Status 창에서는 '운전 모드'를 표시합니다.



참고

NX701 CPU 유니트 또는NJ 시리즈 CPU 유니트에서 운전 모드 중임을 외부에 출력하고자 하는 경우에는 전원 유니트의 운전 중 출력을 사용하십시오. 전원 유니트의 운전 중 출력에 대해서는 "NX 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-418)" 및 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"을 참조하십시오.

3

I/O 포트와 슬레이브/유니트 구성

이 장에서는 I/O 포트의 사용 방법과 슬레이브/유니트 구성 작성과 순서, 각 기능의 할당에 대해 설명합니다.

3-1 슬레이브/유니트 구성 순서 3-2

3-2 슬레이브/유니트 구성의 작성과 비교 3-4

3-2-1 EtherCAT 슬레이브 구성의 작성 3-4
3-2-2 유니트 구성의 작성 3-5
3-2-3 유니트 구성의 비교 3-5

3-3 I/O 포트와 디바이스 변수 3-7

3-3-1 I/O 포트 3-8
3-3-2 I/O 포트명 3-9
3-3-3 디바이스 변수 3-11

3-4 CPU 유니트에 대한 변수 할당 3-14

3-4-1 유니트에 대한 변수 할당 방법 3-14
3-4-2 유니트에 할당한 변수의 사용 방법 3-16

3-5 '축'의 작성과 서보/ 엔코더 입력 슬레이브 등의 할당 3-18

3-5-1 개요 3-18
3-5-2 축 변수/축 그룹 변수 3-18
3-5-3 '축' 및 축 변수 작성과 사용 방법 3-20

3-1 슬레이브/유니트 구성 순서

여기에서는 슬레이브/유니트 구성의 순서를 나타냅니다.

슬레이브/유니트 구성은 전체 순서 중에서 다음과 같이 회색으로 표시한 위치에 대응합니다.

STEP 1. 소프트웨어 설계

STEP 1-1 I/O 및 처리의 설계
STEP 1-2 태스크 설계
STEP 1-3 프로그램 설계

↓

STEP 2. 소프트웨어 설정 및 프로그래밍

STEP 2-1 슬레이브/유니트 구성
STEP 2-2 컨트롤러 설정
STEP 2-3 프로그래밍
STEP 2-4 오프라인 디버그

↓

STEP 3. 장착 및 하드웨어 설정

↓

STEP 4. 배선

↓

STEP 5. 실제 기기에 의한 동작 확인과 운전

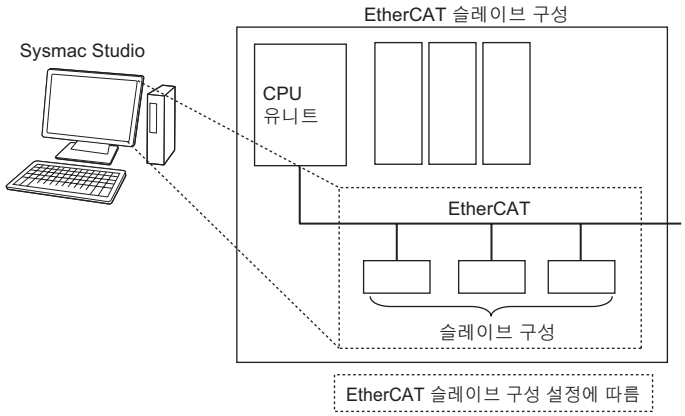
자세한 내용은 📖'1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.

STEP 1 EtherCAT 슬레이브 구성의 작성(EtherCAT을 사용하는 경우), 유니트 구성의 작성(NX 유니트 또는 CJ 유니트를 사용하는 경우)

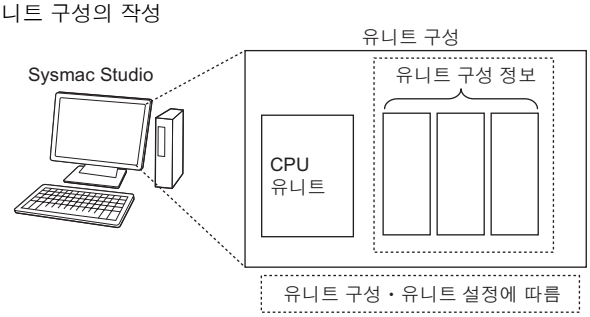
참조

'3-2-1 EtherCAT 슬레이브 구성의 작성(3-4페이지)'
'3-2-2 유니트 구성의 작성(3-5페이지)'

・ EtherCAT 슬레이브 구성의 작성



・ 유니트 구성의 작성

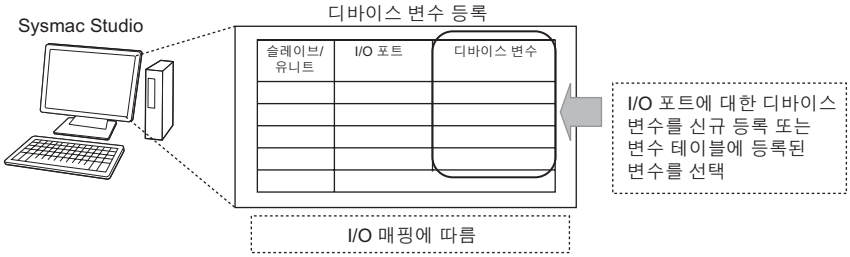


STEP 2 I/O 포트에 디바이스 변수 할당

참조

'2-3-1 변수의 종류(2-12페이지)'
'3-3 I/O 포트와 디바이스 변수(3-7페이지)'

・ 디바이스 변수 등록

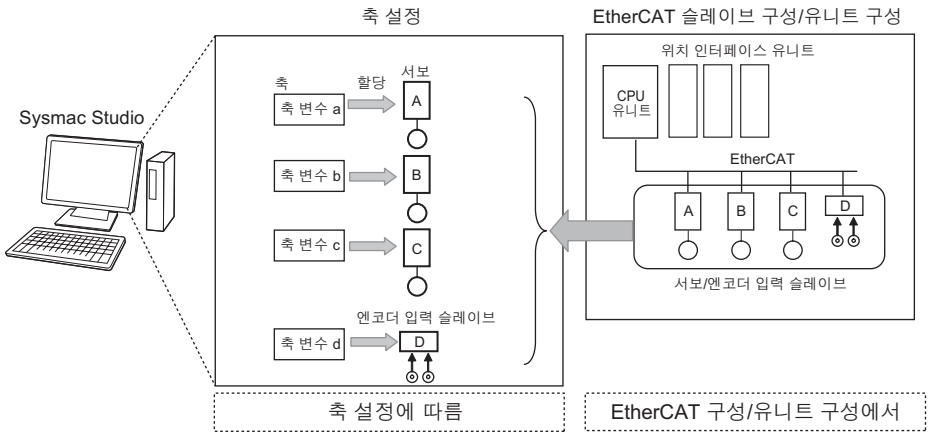


STEP3 '축'의 작성과 슬레이브/유니트에 대한 할당(모션 제어를 하는 경우)

참조

'3-5 '축'의 작성과 서보/엔코더 입력 슬레이브 등의 할당(3-18페이지)'

1. '축' 작성
2. EtherCAT 구성/유니트 구성상의 슬레이브/유니트에 대한 할당

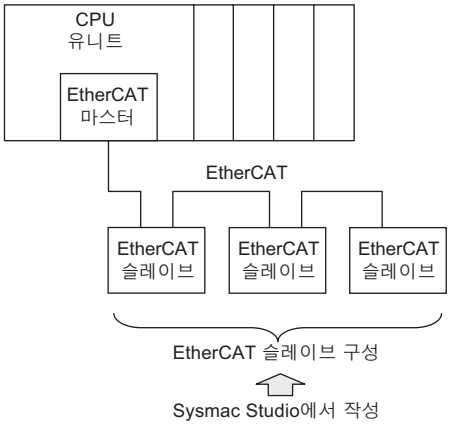


3-2 슬레이브/유니트 구성의 작성과 비교

컨트롤러 내의 각 슬레이브/유니트에 액세스하기 위해 Sysmac Studio에서 슬레이브 구성 및 유니트 구성을 작성합니다. 또한, Sysmac Studio에서 작성한 유니트 구성과 실제 유니트 구성이 동일한지 비교합니다.

3-2-1 EtherCAT 슬레이브 구성의 작성

CPU 유니트가 '옳다'고 인식하는 EtherCAT 슬레이브 구성을 Sysmac Studio의 EtherCAT 구성 편집 화면에서 작성합니다.



이것으로 구성 슬레이브의 'I/O 포트'가 자동 등록됩니다. 'I/O 포트'에는 나중에 사용자가 '디바이스 변수'를 할당합니다.
사용자 프로그램 상에서 '디바이스 변수'를 지정하면 각 슬레이브에 액세스할 수 있습니다.

EtherCAT 슬레이브 구성의 작성 방법은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"의 '[EtherCAT]의 구성·설정'을 참조하십시오.

참고

- EtherCAT 슬레이브 터미널을 연결하는 경우에는 EtherCAT 슬레이브 구성의 작성에 더해서 슬레이브 터미널의 구성과 동작 설정이 필요합니다. 슬레이브 터미널의 구성과 동작 설정의 자세한 내용은 📖"NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.
- NX1P2 CPU 유니트에서 내장 I/O 및 옵션 보드를 사용하는 경우에는 컨트롤러 설정에서 구성 등을 설정합니다.

버전 관련 정보

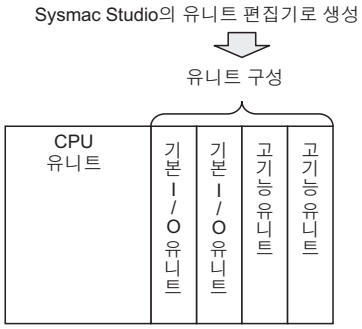
EtherCAT 슬레이브 터미널은 CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.

참고

EtherCAT 슬레이브가 서보/엔코더 입력 슬레이브인 경우에는 EtherCAT 슬레이브 구성에 등록한 후에 다시 '축'을 작성해서 축 변수를 자동 등록합니다.
📖'3-5 '축'의 작성과 서보/ 엔코더 입력 슬레이브 등의 할당(3-18페이지)'을 참조하십시오.

3-2-2 유니트 구성의 작성

CPU 유니트가 '옳다'고 인식하는 유니트 구성을 Sysmac Studio의 '유니트 구성·유니트 설정'의 유니트 편집기에서 작성합니다.



유니트 구성을 작성하면 실제 유니트 구성이 '옳다'고 판단한 '유니트 구성'과 일치하는지 여부가 전원 투입 시 자동으로 확인됩니다.
또한, '유니트 구성'에서 지정한 유니트에 대해 'I/O 포트'가 자동 등록됩니다.
'I/O 포트'에는 나중에 사용자가 '디바이스 변수'를 할당합니다.
'유니트 구성'의 유니트에 대해서는 사용자 프로그램 상에서 '디바이스 변수'를 통해 액세스합니다.

유니트 구성 작성 방법은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"의 '[CPU·증설 랙]의 구성·설정'을 참조하십시오.

사용상의 주의 사항

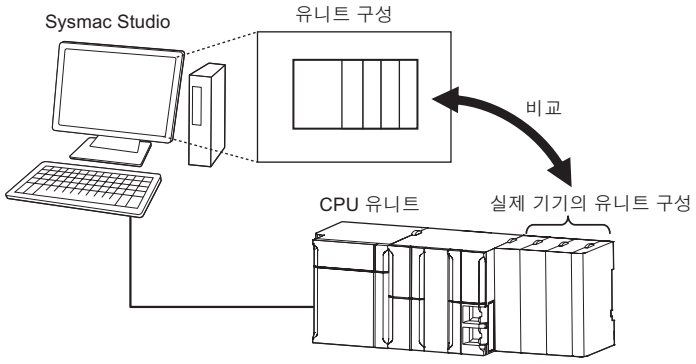
- '유니트 구성'을 작성하지 않아도 컨트롤러를 기동할 수 있습니다. 그러나 I/O 포트와 디바이스 변수가 자동 등록되지 않으므로 사용자 프로그램에서 각 유니트에 액세스할 수 없습니다.
- NX102 및 NX1P2 CPU 유니트에서 NX 유니트를 사용하는 경우 또는 NJ 시리즈 CPU 유니트에서 CJ 유니트를 사용하는 경우에 '유니트 구성'을 작성합니다
  NX701 CPU 유니트에서는 유니트를 사용하지 않으므로 '유니트 구성'은 없습니다.

3-2-3 유니트 구성의 비교

Sysmac Studio에서는 다음과 같은 유니트 구성을 비교할 수 있습니다.

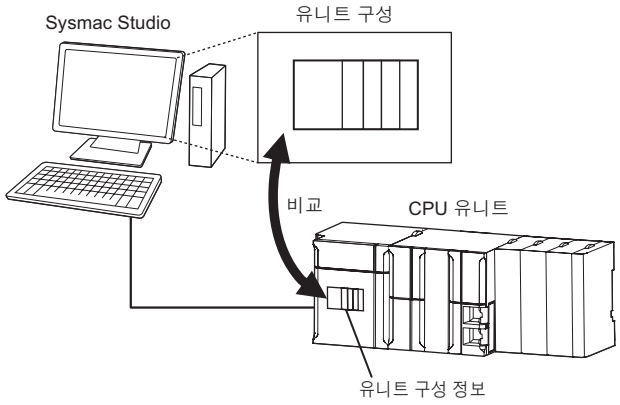
Sysmac Studio 상의 유니트 구성과 실제 기기의 유니트 구성 간의 비교

Sysmac Studio 상의 유니트 구성과 실제 기기의 유니트 구성이 일치하는지 여부를 비교합니다.
Sysmac Studio에서 처음 CPU 유니트에 유니트 구성을 전송하기 전에 해당 유니트 구성이 실제 기기의 유니트 구성과 일치하는지 여부를 확인하는 경우 등에 실행합니다.



Sysmac Studio 상의 유니트 구성과 CPU 유니트 내의 유니트 구성 간의 비교

Syamac Studio 상의 유니트 구성과 CPU 유니트 내에서 유지하는 유니트 구성이 일치하는지 여부를 비교합니다.
유니트 구성이 이미 전송된 CPU 유니트에 Sysmac Studio에서 유니트 구성을 전송하기 전에 해당 내용이 일치하는지 여부를 확인하는 경우 등에 실행합니다.



3-3 I/O 포트와 디바이스 변수

여기에서는 NJ/NX 시리즈 컨트롤러의 EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, 내장 I/O, 옵션 보드 및 CJ 시리즈 유니트에 액세스하기 위한 I/O 포트 및 디바이스 변수에 대해 설명합니다.



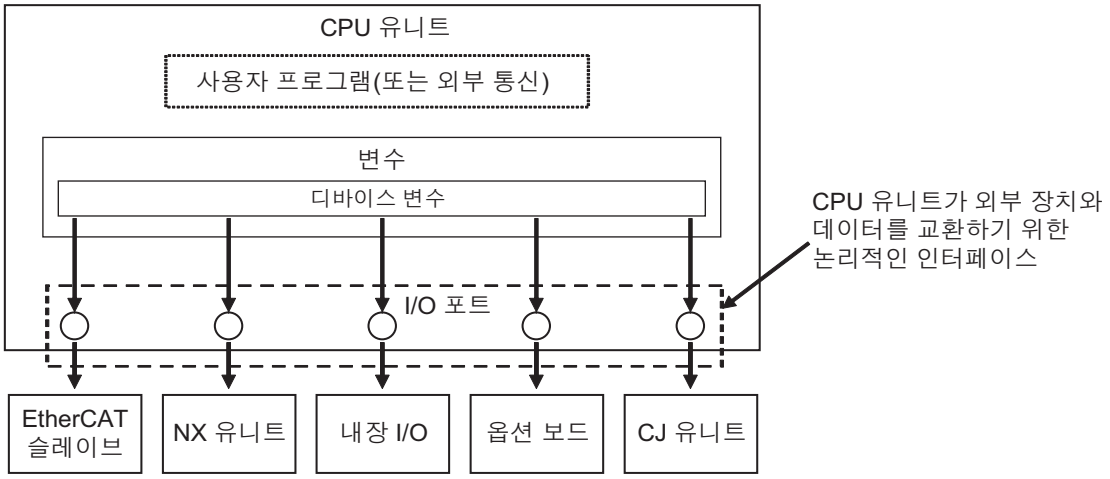
사용상의 주의 사항

- CPU 유니트 상의 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
- 내장 I/O 및 아날로그 입출력용 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
- CJ 시리즈 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

3-3-1 I/O 포트

'I/O 포트'는 CPU 유니트가 외부 장치(슬레이브/유니트), 내장 I/O 및 아날로그 입출력용 옵션 보드와 데이터를 교환하기 위한 논리적인 인터페이스입니다.

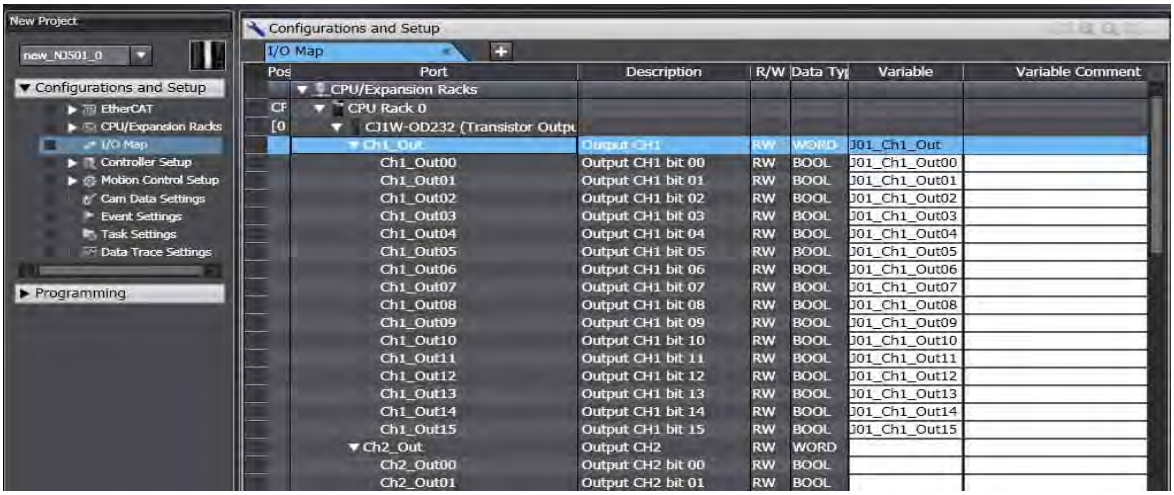
I/O 포트는 Sysmac Studio 상에서 슬레이브/유니트 구성 등을 작성하면 자동으로 생성됩니다. I/O 포트에 디바이스 변수를 할당하면 사용자 프로그램에서 액세스합니다.



'I/O 포트'는 Sysmac Studio에서 EtherCAT 슬레이브 구성 및 유니트 구성 등을 작성하거나 이들을 실제 기기 컨트롤러에서 읽으면 I/O 맵 상에 자동으로 등록됩니다.

등록된 'I/O 포트'는 Sysmac Studio의 'I/O 맵'을 통해 확인할 수 있습니다.

I/O 맵:



3-3-2 I/O 포트명

I/O 포트명은 자동으로 등록됩니다. I/O 포트명은 장치가 EtherCAT 슬레이브인 경우, CPU 유니트 상의 NX 유니트인 경우, 내장 I/O인 경우, 아날로그 입출력용 옵션 보드인 경우, CJ 시리즈 기본 I/O 유니트인 경우, CJ 시리즈 고기능 유니트인 경우에 다음과 같이 달라집니다.

장치가 EtherCAT 슬레이브인 경우

장치가 EtherCAT 슬레이브이고, 리모트 I/O 터미널인 경우에 I/O 포트명은 다음과 같습니다.

예) 16점 리모트 I/O 터미널인 경우:
Bit00~Bit15

그 밖의 슬레이브에서는 EtherCAT Object Dictionary에서 정의된 개체명 또는 그 일부가 됩니다.

예) 아날로그 입력 유니트인 경우
CH0_input16-bit

예) R88D-KN50H-ECT인 경우
Position actual value, Digital inputs 등

장치가 CPU 유니트 상의 NX 유니트인 경우

장치가 CPU 유니트 상의 NX 유니트인 경우에 I/O 포트명은 NX 유니트의 형식 및 기능별 고유명이 됩니다.

예) 디지털 입력 유니트인 경우:
Input Bit 00

예) 아날로그 출력 유니트인 경우:
Ch1 Analog Output Value

장치가 내장 I/O인 경우

장치가 내장 I/O인 경우에 I/O 포트명은 NX1P2 CPU 유니트의 형식별로 접점의 수만큼 생성됩니다.

예) NX1P2-9024DT인 경우:
입력 접점: Input Bit 00~Input Bit 13
출력 접점: Output Bit 00~Output Bit 09

장치가 옵션 보드인 경우

장치가 옵션 보드인 경우에 I/O 포트명은 옵션 보드의 형식 및 기능별 고유명이 됩니다.

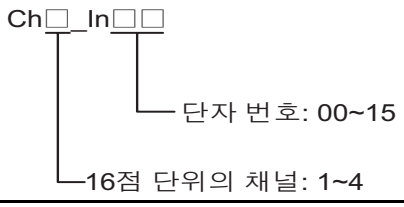
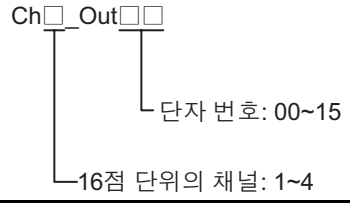
예) 아날로그 입력 옵션 보드인 경우:
Ch1 Analog Input Value

예) 아날로그 출력 옵션 보드인 경우:
Ch1 Analog Output Value

장치가 CJ 시리즈 기본 I/O 유니트인 경우

장치가 CJ 시리즈 기본 I/O 유니트인 경우에 I/O 포트명은 다음과 같은 일정한 규칙으로 생성됩니다.

- **기본 I/O 유니트의 I/O 포트명 규칙**

입력	출력
	

- **점수별 I/O 포트명의 예**

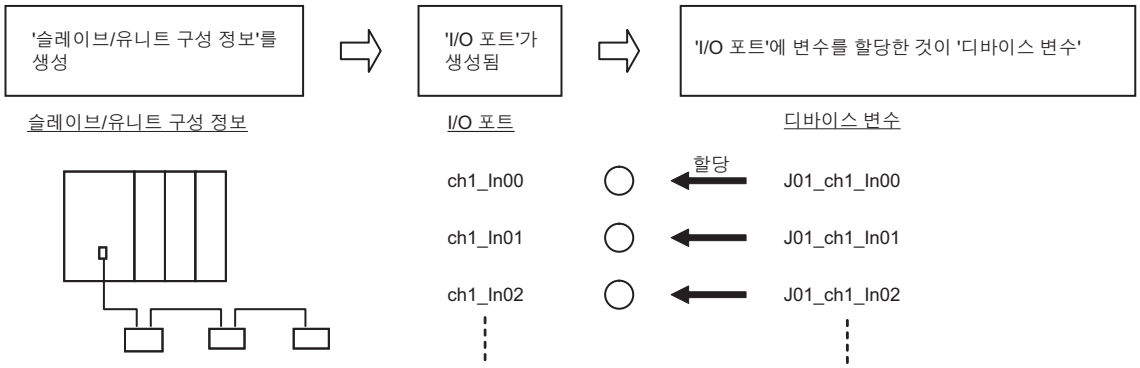
입력 점수 출력 점수	I/O 포트명			
	입력	데이터 형식	출력	데이터 형식
32점	Ch1_In	WORD	Ch1_Out	WORD
	Ch1_In00~Ch1_In 15	BOOL	Ch1_Out00~Ch1_Out 15	BOOL
	Ch2_In	WORD	Ch2_Out	WORD
	Ch2_In00~Ch2_In 15	BOOL	Ch2_Out00~Ch2_Out 15	BOOL

장치가 CJ 시리즈 고기능 유니트인 경우

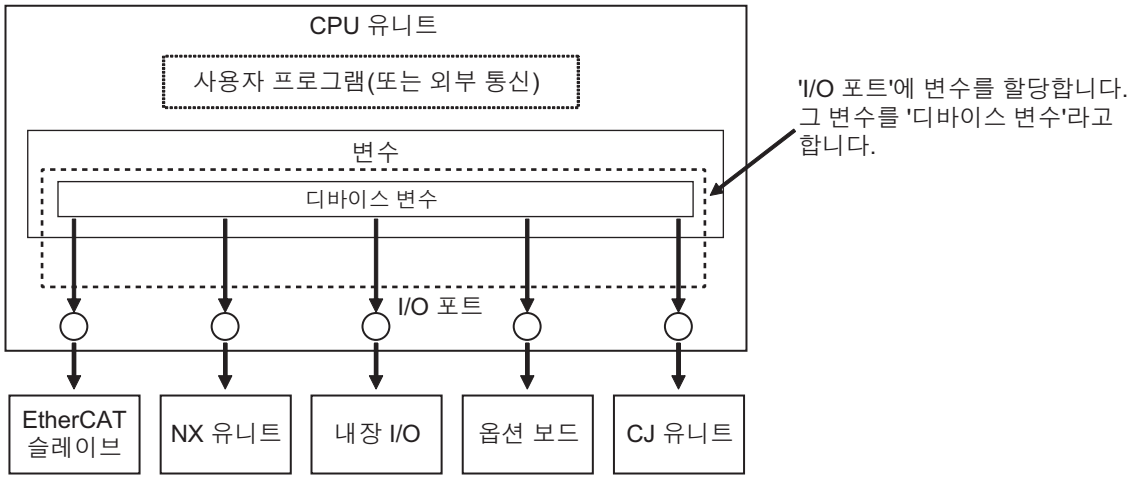
장치가 CJ 시리즈 고기능 유니트인 경우에 I/O 포트명은 유니트 형식 및 기능별 고유명이 됩니다.
예) 아날로그 입력 유니트 CJ1W-AD041-V1인 경우
Ch1_PkHdCmd, Ch1_AveCfg 등

3-3-3 디바이스 변수

NJ/NX 시리즈에서는 CPU 유니트의 외부 장치가 할당되는 메모리 공간 상의 주소가 아니라 이 'I/O 포트'에 대해 '변수'를 할당합니다.
이 변수를 '디바이스 변수'라고 합니다.



사용자 프로그램(또는 외부 통신)은 이 '디바이스 변수'에 따라 장치(슬레이브/유니트)에 액세스할 수 있습니다.



변수 전체에서 디바이스 변수의 위치에 대해서는 '2-3-1 변수의 종류(2-12페이지)'를 참조하십시오.

Sysmac Studio에서의 디바이스 변수 등록 방법은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

디바이스 변수의 속성

디바이스 변수의 속성은 다음과 같습니다. 또한, 설정 변경이 가능한 속성과 불가능한 속성이 있습니다.

속성	설정되는 내용	설정 변경
변수명	자동 생성하는 경우: [장치명_]+[I/O 포트명] '장치명'은 기본값에서는 • EtherCAT 슬레이브인 경우에 'E'+ '001부터 연속되는 번호'가 됩니다. • CJ 유니트인 경우에 'J'+ '01부터 연속되는 번호'가 됩니다. 'I/O 포트명'은 '3-3-2 I/O 포트명(3-9페이지)'을 참조하십시오. 수동 입력하는 경우: 임의의 변수명	가능
데이터 형식	'I/O 포트'의 데이터 형식에 따릅니다.	가능
AT(할당대상) 지정	• NX 유니트(EtherCAT 커플러 유니트 경유) ECAT://node#[EtherCAT 커플러 유니트 노드 주소, NX 유니트 번호]/[I/O 포트명] • NX 유니트(CPU 유니트 상) IOBus://unit#[NX 유니트 번호]/[I/O 포트명] • 내장 I/O BuiltInIO://cpu/#0/[I/O 포트명] • 옵션 보드 BuiltInIO://opt/#[물리 포트 번호]/[I/O 포트명] • EtherCAT 슬레이브 ECAT://node#[노드 주소]/[I/O 포트명] • CJ 유니트 IOBus://rack#[랙 번호]/slot#[슬롯 번호]/[I/O 포트명]	불가능
유지	• EtherCAT 슬레이브 디바이스 변수: 비유지 • NX 유니트 디바이스 변수: 비유지 • 내장 I/O 디바이스 변수: 비유지 • 옵션 보드 디바이스 변수: 비유지 • CJ 유니트 디바이스 변수 중에 '운전용 데이터'(CIO 영역)에 할당된 변수: '비유지', '설정용 데이터'(DM 영역)에 할당된 변수: '유지'	불가능
초기값	없음	가능
상수	없음	가능
네트워크 공개	비공개	가능
엣지	없음	불가능

(주의) CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
(주의) CPU 유니트 상의 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
(주의) 내장 I/O 및 옵션 보드를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
(주의) NX 유니트 디바이스 변수를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
(주의) 내장 I/O 디바이스 변수 및 옵션 보드 디바이스 변수를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
(주의) '물리 포트 번호'는 I/O 포트의 물리적인 위치를 나타냅니다. 옵션 보드 슬롯 1인 경우에 '0', 옵션 보드 슬롯 2인 경우 '1'이 됩니다.

각 속성의 의미는 '6-3-4 변수의 속성(6-30페이지)'을 참조하십시오.



참고

- I/O 맵 상에서 I/O 포트의 강제값 리프레시를 실행할 수 있습니다. 이에 따라 실제 I/O를 강제적으로 ON/OFF시켜서 배선 확인을 실행할 수 있습니다.
- I/O 맵 상에서는 디바이스 변수를 어느 변수 테이블(글로벌 변수 테이블, 각 POU의 로컬 변수 테이블)에 등록할지 선택할 수 있습니다.

3-4 CPU 유니트에 대한 변수 할당

명령에는 EtherCAT 슬레이브 터미널의 각 유니트 및 NX102, NX1P2 CPU 유니트 상의 NX 유니트를 변수의 형식으로 지정하는 형식이 있습니다. 따라서 사용자는 사전에 각 유니트에 변수를 할당해 둬야 합니다. 일단 유니트에 변수를 할당하면 유니트의 연결 위치를 변경해도 그 정보가 자동으로 변수 값에 반영됩니다. 따라서 유니트의 연결 위치를 변경할 때마다 변수를 다시 할당할 필요가 없습니다.

버전 관련 정보

유니트에 대한 변수의 할당은 CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합에서 사용할 수 있습니다.

참고

슬레이브 터미널 이외의 EtherCAT 슬레이브에도 변수를 할당할 수 있습니다. 타사 EtherCAT 슬레이브도 동일합니다. EtherCAT 슬레이브에 변수를 할당하는 방법은 EtherCAT 커플러 유니트 및 NX 유니트에 변수를 할당하는 방법과 동일합니다.

3-4-1 유니트에 대한 변수 할당 방법

Sysmac Studio에서 EtherCAT 슬레이브 터미널의 구성 설정 및 유니트 구성 설정을 실행해도 유니트에 할당하는 변수는 자동으로 작성되지 않습니다. 다음 순서로 유니트에 대한 변수의 할당 설정을 실행하십시오.

1 **Sysmac Studio의 [구성·설정]–[EtherCAT] 또는 [구성·설정]–[CPU·증설 랙]–[CPU 랙]을 통해 EtherCAT 슬레이브 터미널 또는 CPU 유니트 상의 NX 유니트의 구성 설정을 실행합니다.**

2 **[구성·설정]-[I/O 맵]을 통해 I/O 맵을 표시합니다.**
설정한 EtherCAT 슬레이브 터미널의 각 유니트의 I/O 맵이 표시됩니다.

3 **변수를 할당하는 유니트의 형식을 마우스 오른쪽 버튼으로 클릭하고 [노드 위치 포트 표시]를 선택합니다.**
I/O 맵에 'Node location information'이라는 포트가 추가됩니다.

4 **'Node location information'를 마우스 오른쪽 버튼으로 클릭하고 [디바이스 변수 생성]을 선택합니다.**
'Nodelocation information' 포트의 [변수]란에 변수명이 쓰여집니다.

유니트에 할당된 변수의 데이터 형식은 _sNXUNIT_ID 형식의 구조체입니다. _sNXUNIT_ID 형식 구조체에 대한 자세한 내용은 다음과 같습니다.

변수	이름	의미	데이터 형식
임의로 설정	지정 유니트	지정하는 유니트	_sNXUNIT_ID
NodeAdr	노드 주소	통신 커플러 유니트의 노드 주소*1	UINT
IPAdr	IP 주소*2	통신 커플러 유니트의 IP 주소*1	BYTE[5]
UnitNo	유니트 번호	지정하는 유니트의 유니트 번호	UDINT
Path	경로*2	지정하는 유니트까지의 경로 정보	BYTE[64]
PathLength	'Path'의 유효 데이터 길이	'Path'의 유효 데이터 길이	USINT

*1. CPU 유니트 상의 NX 유니트인 경우에 사용되지 않습니다.
*2. 컨트롤러 내부에서만 사용하는 정보입니다. 사용자가 참조 및 편집하지는 않습니다.

사용상의 주의 사항

유니트에 할당하는 변수의 값은 변수 등록 시 자동으로 설정됩니다. 변수의 값은 변경하지 마십시오. 사용자가 변수의 값을 변경하면 컨트롤러가 의도대로 동작하지 않을 가능성이 있습니다.

참고

슬레이브 터미널 이외의 EtherCAT 슬레이브에 변수를 할당할 때 변수의 데이터 형식은 _sECAT_ID 형식의 구조체입니다 자세한 내용은 다음과 같습니다.

변수	이름	의미	데이터 형식
임의로 설정	지정 슬레이브	지정하는 슬레이브	_sECAT_ID
NodeAdr	노드 주소	지정 슬레이브의 노드 주소	UINT

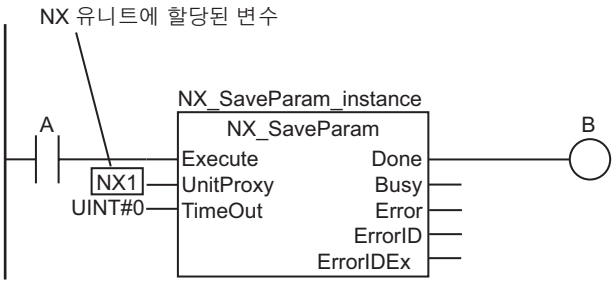
3-4-2 유니트에 할당한 변수의 사용 방법

유니트에 할당한 변수는 사용자 프로그램 상에서 다음과 같이 사용합니다. 어떤 경우에도 I/O 맵에서 유니트에 할당한 변수와 동일한 이름의 변수를 변수 테이블에 등록해 둬야 합니다. 해당 변수의 데이터 형식은 _sNXUNIT_ID 형식의 구조체입니다.

유니트 지정

유니트를 지정하는 명령에는 유니트에 할당한 변수를 그대로 파라메터로 전달합니다.

예) NX_SaveParam 명령의 실행
'NX1'이라는 변수를 할당한 NX 유니트를 지정하여 NX_SaveParam 명령을 실행합니다. 변수 'UnitProxy'에 'NX1'을 전달합니다.



유니트 속성의 지정

유니트에 할당한 변수의 멤버를 지정하면 유니트의 일부 속성만 지정할 수 있습니다.

예) NX 유니트의 유니트 번호에 의한 명령 실행
'NX1'이라는 변수를 할당한 NX 유니트의 유니트 번호가 2이면 그 유니트의 데이터 개체에서 데이터를 읽습니다. 변수 'NX1.UnitNo'가 유니트 번호를 나타냅니다.

```
IF (NX1.UnitNo = UINT#2) THEN
  NX_ReadObj_instance(Execute:=TRUE, UnitProxy:=NX1, Obj:=S_Obj, ReadDat:=Rdat);
END_IF;
```

복수 유니트의 지정

유니트에 할당하는 변수를 배열 지정으로 합니다. 루프 처리를 사용하여 복수의 유니트에 동일한 처리를 실행할 수 있습니다.

예) 복수의 NX 유니트를 데이터 쓰기 가능한 모드로 하는 명령 실행
'NX0', 'NX1', 'NX2'가 NX 유니트에 할당된 변수입니다. 각각을 배열 NXTable[0..2]에 대입한 다음 순서대로 NX_ChangeWriteMode 명령을 실행합니다.

- 변수 테이블

이름	데이터 형식
NXTable	ARRAY[0..2] OF_sNXUNIT_ID

- ST 프로그램

```
FOR i:= 0 TO 2 DO
  NX_ChangeWriteMode_instance[i](Execute:=FALSE);
END_FOR;

NXTable[0] := NX0;
NXTable[1] := NX1;
NXTable[2] := NX2;

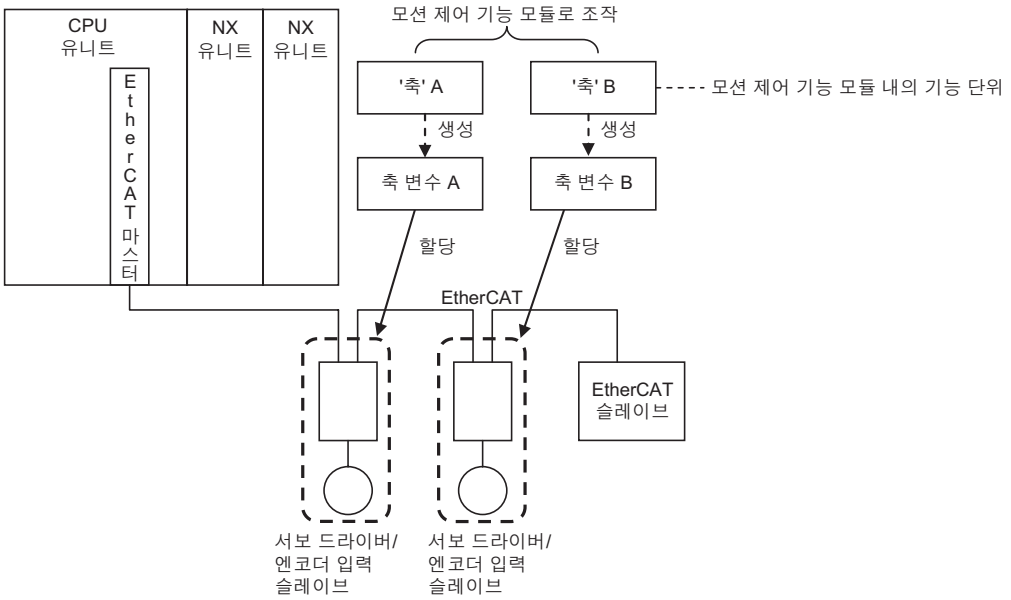
FOR i:= 0 TO 2 DO
  NX_ChangeWriteMode_instance[i](Execute:=TRUE, UnitProxy:=NXTable[i]);
END_FOR;
```

3-5 '축'의 작성과 서보/엔코더 입력 슬레이브 등의 할당

여기에서는 NJ/NX 시리즈 컨트롤러에서의 '축' 작성과 서보 드라이버/엔코더 입력 슬레이브/NX 유니트의 '축'에 대한 할당을 설명합니다.

3-5-1 개요

EtherCAT 슬레이브 안의 서보 드라이버 및 엔코더 입력 슬레이브 및 NX 유니트를 모션 제어 기능 모듈로 조작하는 경우에 Sysmac Studio 상에서 '축'을 작성하고, EtherCAT 상의 서보 드라이버 및 엔코더 입력 슬레이브 또는 NX 유니트에 할당합니다.
이를 통해 시스템 정의 변수의 축 변수가 자동으로 작성됩니다.



사용자 프로그램 상의 모션 제어 명령으로 축 변수를 지정하면 서보 드라이버 및 엔코더 입력 슬레이브 및 NX 유니트에 간단하게 액세스하여 조작할 수 있습니다.

3-5-2 축 변수/축 그룹 변수

축 변수/축 그룹 변수에는 다음과 같은 종류가 있습니다.

변수 종류		용도	액세스 대상 장치	생성 방법
축 변수	시스템 정의의 축 변수	1축을 단독으로 제어하는 경우	'축'에 할당된 EtherCAT 슬레이브(서보/엔코더 입력 슬레이브)/NX 유니트	표준으로 준비
	Sysmac Studio에서 '축'을 작성했을 때 자동 생성되는 축 변수			Sysmac Studio에서 '축'을 작성하고, 축 할당
축 그룹 변수	시스템 정의의 축 그룹 변수	여러 축을 다축 협조 제어하는 경우	'축 그룹'에 할당된 EtherCAT 슬레이브(서보/엔코더 입력 슬레이브)/NX 유니트	표준으로 준비
	Sysmac Studio에서 '축 그룹'을 작성했을 때 자동 생성되는 축 그룹 변수			Sysmac Studio에서 '축 그룹'을 작성

축 변수/축 그룹 변수의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

• 지정 방법

다음과 같은 Sysmac Studio 작성과 시스템 정의를 모두 지정할 수 있습니다.

종류	이름	
	축 변수	축 그룹 변수
Sysmac Studio에서 작성하는 것	MC_Axis***(***는 작성 순서대로 000에서부터 오름차순) 사용자가 임의로 변경할 수 있습니다.	MC_Group***(***는 작성 순서대로 000에서부터 오름차순) 사용자가 임의로 변경할 수 있습니다.
시스템 정의인 것	_MC_AX[***]*1 (배열 요소 번호는 작성 순서대로 0에서부터 오름차순)	_MC_GRP[***]*1 (배열 요소 번호는 작성 순서대로 0에서부터 오름차순)

*1. NX701 CPU 유니트에서는 _MC1_AX[***], _MC2_AX[***] 및 _MC1_GRP[***], _MC2_GRP[***]도 사용할 수 있습니다. 자세한 내용은 "NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

• 용도

다음 2종류가 있습니다.

(1) 모션 제어 명령에서의 축 지정 또는 축 그룹 지정:

모션 제어 명령의 입출력 변수에 지정하여 OMRON 서보/엔코더 입력 슬레이브 및 NX 유니트에 대해 조작할 수 있습니다.

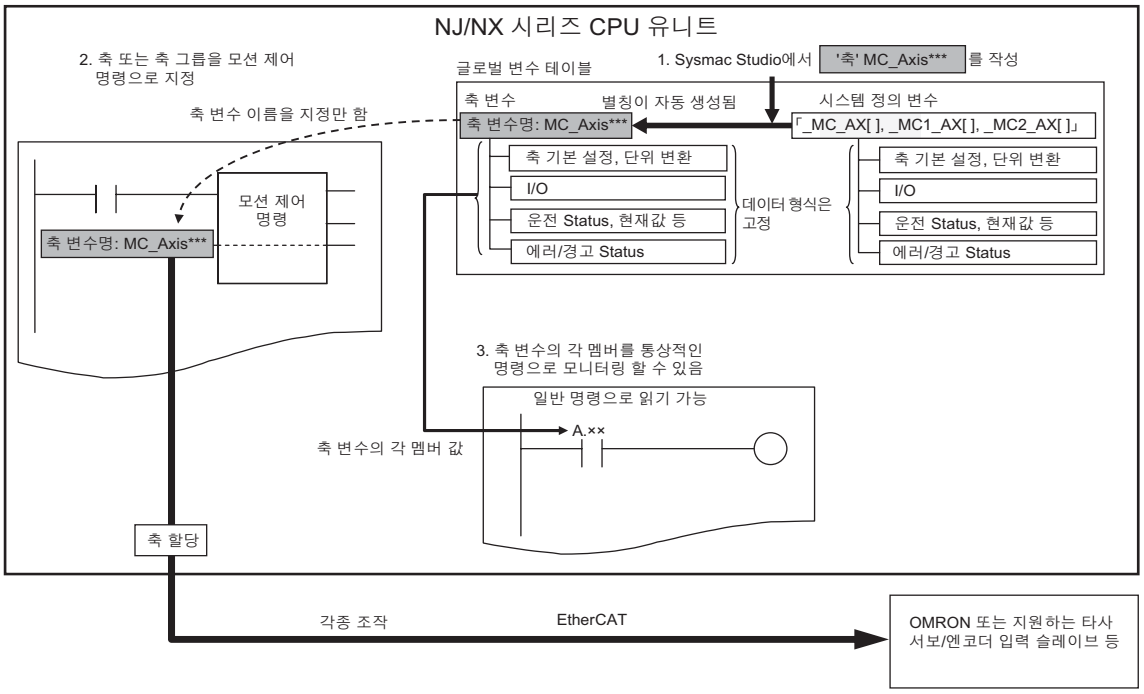
(2) 축 변수 각 멤버의 모니터링:

통상적인 명령에서 각 서보/엔코더 입력 슬레이브 및 NX 유니트의 현재 위치 및 이상 정보 등을 모니터링할 수 있습니다.

참고

축 변수의 세부 구조

1. Sysmac Studio에서 축 이름이 'A'인 '축'을 작성합니다.
   시스템 정의 변수의 축 변수를 바탕으로 변수명 'A'의 '축 변수'가 자동 생성됩니다. 축 변수는 축 기본 설정, 단위 변환, I/O, 운전 Status, 현재값, 에러/경고 Status로 이루어집니다.
2. 모션 제어 명령의 입출력 변수로 축 변수명 'A'를 지정합니다. OMRON 또는 지원 가능한 타사 서보/엔코더 입력 슬레이브/NX 유니트에 대해 액세스하여 각종 조작을 할 수 있습니다.
3. 필요에 따라 축 변수를 지정하여 통상적인 명령으로 각 서보/엔코더 입력 슬레이브/NX 유니트의 현재 위치 및 이상 정보 등을 모니터링할 수 있습니다.



3-5-3 '축' 및 축 변수 작성과 사용 방법

다음과 같이 '축' 및 축 변수를 작성하여 사용합니다.

1 **멀티 뷰 탐색기의 [구성·설정]–[모션 제어 설정]–[축 설정]을 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [추가]–[축 설정]을 선택합니다.**

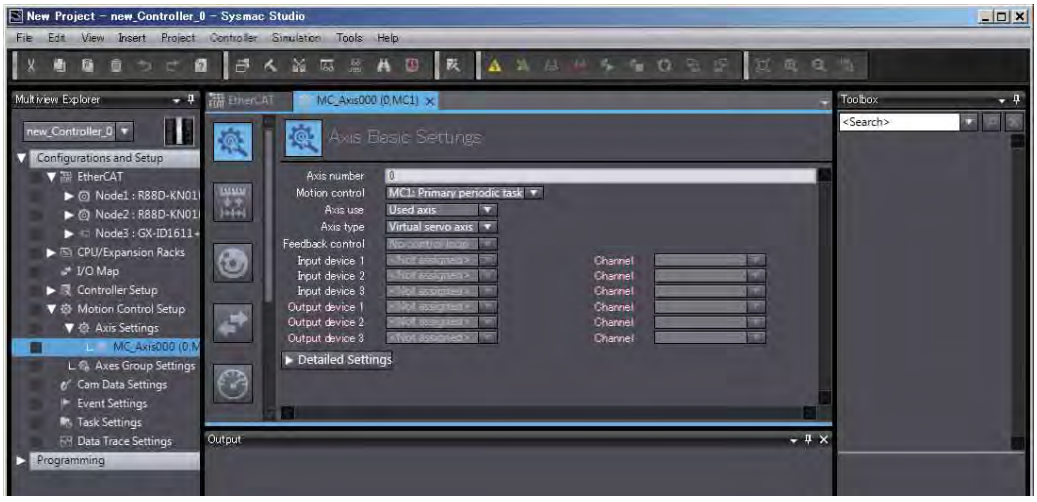
필요에 따라 축 변수명을 기본값인 'MC_Axis***'(***는 작성 순서대로 000에서부터 오름차순)에서 변경합니다.

2 **작성한 '축'에 Sysmac Studio의 EtherCAT 슬레이브 구성상의 서보 드라이버/엔코더 입력 슬레이브를 할당합니다.**

Sysmac Studio에서 다음과 같은 '축 기본 설정'을 실행합니다.

구분	파라메터명	설정 내용
축 기본 설정	축 번호	축을 작성한 순서대로 자동 설정됩니다.
	모션 제어*1	[프라이머리 정주기 태스크]를 선택합니다.
	축 사용	[사용 축]을 선택합니다.
	축 종류	서보 축 또는 엔코더 축을 선택합니다.
	입력/출력 장치	축에 할당하는 EtherCAT 슬레이브를 선택합니다.

*1. NX701 CPU 유니트인 경우에 설정합니다.

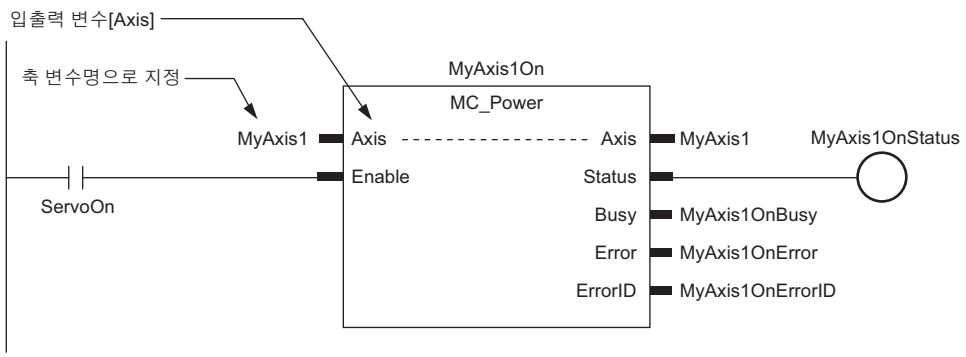


3 **Sysmac Studio에서 시운전에 필요한 설정(단위 변환, 카운트 모드, 리미트 등) 및 본 운전에 필요한 설정을 실행하고, 프로젝트로서 CPU 유니트에 전송합니다.**

4 **사용자 프로그램 상에서 모션 제어 명령의 입출력 변수 'Axis'에 축 변수명을 지정합니다.**

축 변수명은 '모션 제어 설정'에서 삽입한 '축 이름'('축 변수명') 또는 시스템 정의 변수를 지정합니다. 대응하는 서보 드라이버/엔코더 입력 슬레이브/NX 유니트에 대해 모션 제어를 실행할 수 있습니다. 다음은 축 변수명 'MyAxis1'로 지정한 경우의 예입니다.

예)



축 변수의 자세한 내용은 '3-5-2 축 변수/축 그룹 변수(3-18페이지)'를 참조하십시오.

사용상의 주의 사항

- NX701 CPU 유니트인 경우에 [축 설정] 및 [축 그룹 설정]의 [모션 제어]의 설정에 대응하는 시스템 정의 변수를 사용하십시오. 대응하지 않는 시스템 정의 변수를 사용하면 빌드 시에 에러가 됩니다.

[모션 제어] 설정	대응하는 시스템 정의 변수
MC1: 프라이머리 정주기 태스크	_MC_AX[***], _MC1_AX[***], _MC_GRP[***], _MC1_GRP[***]
MC2: 정주기 태스크(우선도 5)	_MC2_AX[***], _MC2_GRP[***]

4

컨트롤러 설정

이 장에서는 각 기능 모듈의 초기 설정에 대해 설명합니다.

4-1 컨트롤러 설정의 개요 4-2

4-2 PLC 기능 모듈의 초기 설정 4-4

4-2-1 개요 4-4
4-2-2 컨트롤러 설정 4-4
4-2-3 태스크 설정 4-7
4-2-4 NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정 4-12
4-2-5 NJ 시리즈 CPU 유니트에서의 유니트 구성·유니트 설정 4-15

4-3 NX 유니트의 초기 설정 4-16

4-3-1 NX 유니트의 설정 항목 4-16
4-3-2 I/O 할당 설정 4-17
4-3-3 유니트 동작 설정 4-19

4-4 고기능 유니트의 초기 설정 4-20

4-5 모션 제어 기능 모듈의 초기 설정 4-22

4-5-1 개요 4-22
4-5-2 설정 방법 4-23

4-6 EtherCAT 마스터 기능 모듈의 초기 설정 4-24

4-7 EtherNet/IP 기능 모듈의 초기 설정 4-25

4-8 내장 I/O 초기 설정 4-26

4-9 옵션 보드 초기 설정 4-27

4-10 CJ 유니트용 메모리 설정 4-28

4-10-1 설정 순서 4-28
4-10-2 설정 화면 4-29
4-10-3 설정 내용 4-30

4-1 컨트롤러 설정의 개요

여기에서는 컨트롤러 설정의 개요를 나타냅니다.

전체 순서 중에서 다음과 같이 회색으로 표시한 위치에 대응합니다.

STEP 1. 소프트웨어 설계

- STEP 1-1 I/O 및 처리의 설계
- STEP 1-2 태스크 설계
- STEP 1-3 프로그램 설계

STEP 2. 소프트웨어 설정 및 프로그래밍

- STEP 2-1 슬레이브/유니트 구성
- **STEP 2-2 컨트롤러 설정**
- STEP 2-3 프로그래밍
- STEP 2-4 오프라인 디버그

STEP 3. 장착 및 하드웨어 설정

STEP 4. 배선

STEP 5. 실제 기기에 의한 동작 확인과 운전

자세한 내용은 📖'1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.

컨트롤러 설정	참조
● **PLC 기본 모듈 관련 초기 설정:** 컨트롤러 설정: 전원 투입 시의 동작 모드, 쓰기 방지 유무, 시스템 서비스 감시 설정 등	'4-2 PLC 기능 모듈의 초기 설정(4-4페이지)'
● **NX 유니트의 초기 설정:** 유니트 구성·유니트 설정: NX 유니트의 초기 설정	'4-3 NX 유니트의 초기 설정(4-16페이지)'
● **고기능 유니트의 초기 설정:** 유니트 구성·유니트 설정: 고기능 유니트의 초기 설정	'4-4 고기능 유니트의 초기 설정(4-20페이지)'
● **모션 제어 기능 모듈 관련 초기 설정:** • 축 파라메터: 단축 동작인 경우의 모션 제어용 파라메터 • 축 그룹 파라메터: 다축 협조 제어 동작인 경우의 모션 제어용 파라메터 • 캠 데이터: 캠 동작에서 사용하는 위상과 변위 간의 설정 테이블	'4-5 모션 제어 기능 모듈의 초기 설정(4-22페이지)'
● **EtherCAT 마스터 기능 모듈 관련 초기 설정:** EtherCAT 구성상의 EtherCAT 마스터의 파라메터: EtherCAT 마스터의 프로세스 데이터 통신 주기 등의 파라메터 설정	'4-6 EtherCAT 마스터 기능 모듈의 초기 설정(4-24페이지)'
● **EtherNet/IP 기능 모듈 관련 초기 설정:** Ethernet 포트 설정: EtherNet/IP 포트의 TCP/IP 설정, Ethernet 설정 등	'4-7 EtherNet/IP 기능 모듈의 초기 설정(4-25페이지)'

4-2 PLC 기능 모듈의 초기 설정

여기에서는 PLC 기능 모듈의 초기 설정에 대해 설명합니다.

4-2-1 개요

PLC 기능 모듈 초기 설정에는 다음과 같은 내용이 있습니다.

- 컨트롤러 설정
- 태스크 설정

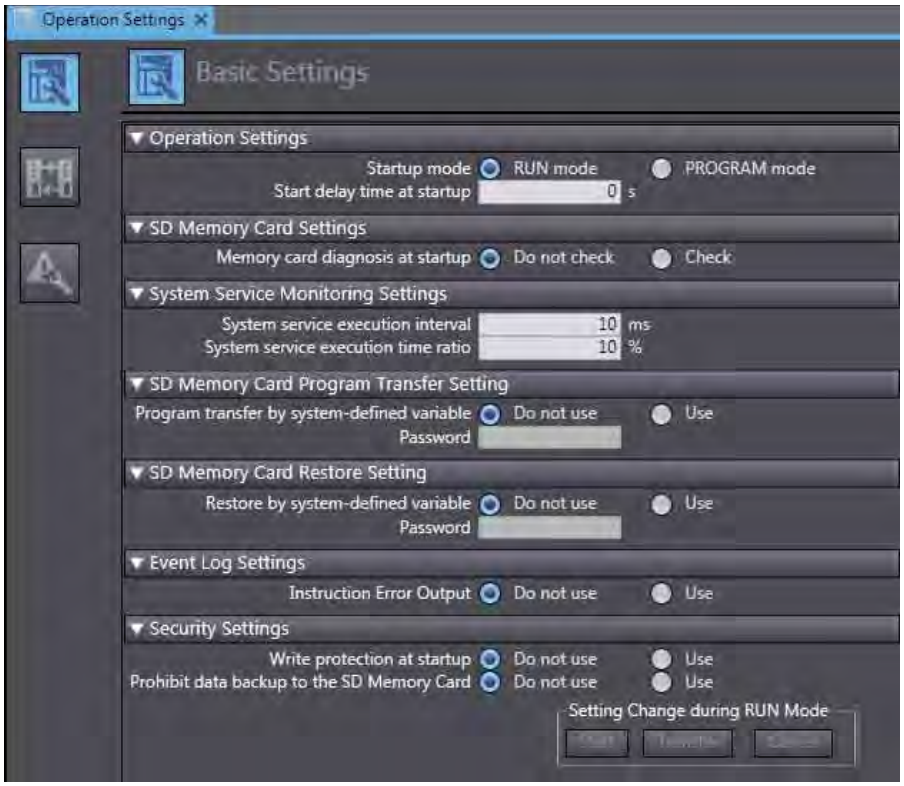
Sysmac Studio에서 [구성·설정]–[컨트롤러 설정], [태스크 설정]에서 각각 설정합니다.

4-2-2 컨트롤러 설정

동작 설정 탭

- **전반 설정**

  전원 투입 시 및 동작 모드 전환 시 등에서의 동작 정의 등 CPU 유니트가 지원하는 기능에 대한 설정입니다.



설정 항목	설정 중항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
동작 설정	전원 투입 시 모드	전원 투입 시의 CPU 유니트의 동작 모드를 설정합니다.	운전 모드 프로그램 모드	운전 모드	설정 전송 시	불가능
	전원 투입 시의 기동 대기 시간*1	전원 투입 후 기동 중에 NS 시리즈 표시기의 기동을 대기, 태그 비교를 우선적으로 실행하기 위한 시간을 설정합니다.*2	0~10s	0s	설정 전송 시	불가능
	배터리 관련 이상 감지*3	배터리 관련 이상 감지 실행 여부를 설정합니다.	사용 안 함 사용	사용 안 함	설정 전송 시	가능
SD 메모리 카드 설정	전원 투입 시의 SD 메모리 카드 진단	전원 투입 시에 장착된 SD 메모리 카드에 대해 자체 진단(파일 시스템의 확인과 복구)을 실행할지 여부를 설정합니다.	확인 안 함 확인	확인 안 함	설정 전송 시	불가능
시스템 서비스 감시 설정*4	시스템 서비스 실행 간격	시스템 서비스를 실행하는 간격을 설정합니다.	10ms~1s	10ms	설정 전송 시	불가능
	시스템 서비스 실행 시간 비율	CPU 유니트 전체의 처리에 대한 시스템 서비스를 실행하는 비율을 감시하는 값을 설정합니다.	5%~50%	10%	설정 전송 시	불가능
SD 메모리 카드프로그램 전송 설정*5	시스템 정의 변수에 의한 프로그램 전송	SD 메모리 카드 프로그램 전송을 사용할지 설정합니다.	사용 사용 안 함	사용 안 함	설정 전송 시	불가능
	비밀번호	SD 메모리 카드 프로그램 전송 실행 시 비교되는 비밀번호를 설정합니다.	반각 영숫자(대문자, 소문자는 구별), 8문자 이상, 32문자 이하	없음	설정 전송 시	불가능
SD 메모리 카드 복원 설정*6	시스템 정의 변수에 의한 복원	시스템 정의 변수에 의한 SD 메모리 카드 백업 기능의 복원을 사용할지 설정합니다.	사용 사용 안 함	사용 안 함	설정 전송 시	불가능
	비밀번호	시스템 정의 변수에 의한 SD 메모리 카드 백업 기능 복원 실행 시에 비교되는 비밀번호를 설정합니다.	반각 영숫자(대문자, 소문자는 구별), 8문자 이상, 32문자 이하	없음	설정 전송 시	불가능
이벤트 로그 설정*7	명령 에러 출력	명령 에러가 발생한 경우에 그에 대응하는 이벤트를 이벤트 로그에 출력할지 여부를 설정합니다.	사용 안 함 사용	사용 안 함	설정 전송 시	불가능
보안 설정	전원 투입 시의 쓰기 방지	컨트롤러 전원 투입 시 자동으로 쓰기 방지를 유효로 할지 여부를 설정합니다.	사용 안 함 사용	사용 안 함	전원 투입 시	가능
	SD 메모리 카드에 대한 백업 금지*8	SD 메모리 카드에 대한 백업을 금지할지 여부를 설정합니다.	사용 안 함 사용	사용 안 함	설정 전송 시	가능

*1. NS 시리즈 표시기를 CPU 유니트의 내장 EtherNet/IP 포트에 연결하고, NS 시리즈 표시기와 CPU 유니트를 동시에 전원 투입하는 경우에 유효가 되는 설정입니다.
CPU 유니트 Ver.1.10 이상과 Sysmac Studio Ver.1.12 이상의 조합에서 사용할 수 있습니다.

*2. 이 설정에 의해 NS 시리즈 표시기의 태그 비교의 처리 시간을 단축할 수 있는 경우가 있습니다. 태그 비교를 우선하는 경우에는 '10', 우선하지 않는 경우는 '0'을 설정하십시오.
'10'을 설정한 경우에 CPU 유니트는 전원 투입 후의 '기동 중'에 약 10초간 NS 시리즈 표시기의 태그 비교의 처리를 우선한 다음 '정상 운전 중'으로 이행합니다. '기동 중'에 태그 비교 처리의 일부를 우선하면 태그 비교가 완료될 때까지의 시간을 단축할 수 있습니다.
'1~10'을 설정한 경우에 NS 시리즈 표시기의 유무에 관계 없이 설정한 시간만 태그 비교의 처리를 우선하기 위해 CPU 유니트가 '정상 운전 중'으로 이행할 때까지의 시간이 길어집니다. NS 시리즈 표시기를 연결하지 않은 경우 또는 동시 전원 투입하지 않는 경우에는 '0'을 설정하십시오.
*3. NX102 및 NX1P2 CPU 유니트에서만 설정할 수 있습니다.
*4. NX 시리즈 CPU 유니트에서는 시스템 서비스 감시 설정이 없습니다.
*5. CPU 유니트 Ver.1.11 이상과 Sysmac Studio Ver.1.15 이상의 조합에서 사용할 수 있습니다.
*6. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합에서 사용할 수 있습니다.
*7. CPU 유니트 Ver.1.02 이상과 Sysmac Studio Ver.1.03 이상의 조합에서 사용할 수 있습니다.
*8. CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 사용할 수 있습니다.

조작 항목	기능 설명
운전 모드 재기입	운전 모드 중 재기입이 가능한 설정 항목의 설정값 재기입을 실행합니다. 조작에 대해서는 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오. 시작 : 설정값 재기입이 가능한 상태가 됩니다. 전송 : 설정값을 컨트롤러에 전송합니다. 설정값이 재기입됩니다. 취소 : 설정값 재기입 금지 상태가 됩니다.

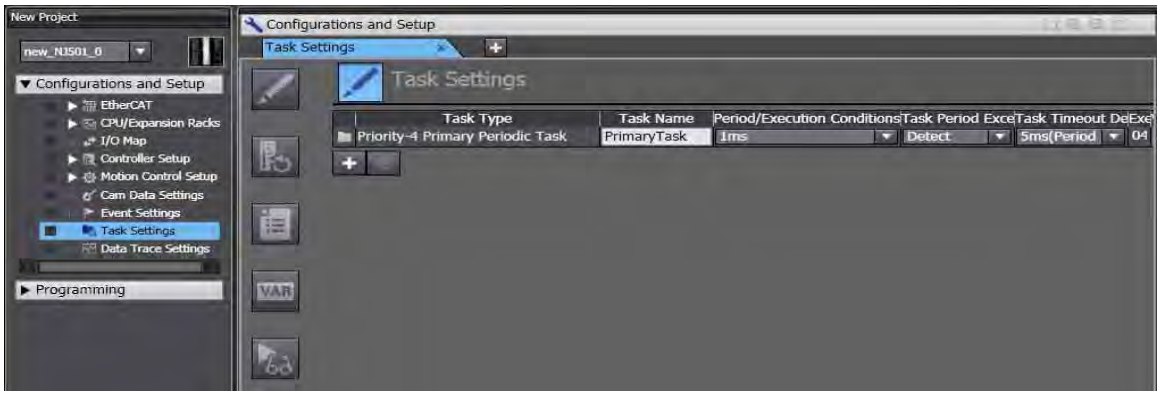
사용상의 주의 사항

[이벤트 로그 설정]–[명령 에러 출력]을 [사용]으로 한 경우에 이상이 있는 명령을 반복해서 실행하면 명령 에러가 매번 출력됩니다. 따라서 이벤트 로그의 기록 건수가 상한을 초과하는 경우가 있습니다. 그 때 오래된 정보부터 순서대로 덮어쓰기되므로 주의하십시오.

4-2-3 태스크 설정

- **태스크 설정**

  태스크 추가와 각 태스크 설정을 실행합니다.



설정 항목	설정 중항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
태스크 타입		태스크 타입을 선택합니다.	프라이머리 정주기 태스크 정주기 태스크 [우선도 5]*1 정주기 태스크 [우선도 16]*2 정주기 태스크 [우선도 17] 정주기 태스크 [우선도 18] 이벤트 태스크 [우선도 8] 이벤트 태스크 [우선도 48]	프라이머리 정주기 태스크	설정 전송 시	불가능
	태스크 실행 우선도	태스크 실행 우선도를 선택합니다.	'태스크 타입'에 따라 자동 선택	프라이머리 정주기 태스크: 4	설정 전송 시	불가능
태스크 이름		태스크 이름을 설정합니다.	문자열	프라이머리 정주기 태스크: PrimaryTask 정주기 태스크: PeriodicTask0 이벤트 태스크: EventTask0	설정 전송 시	불가능

설정 항목	설정 중항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
주기/ 실행 조건		태스크 주기를 선택합니다.	프라이머리 정주기 태스크/정주기 태스크: '5-3-1 NX701의 태스크 사양(5-9페이지)' 참조 '5-4-1 NX102 및 NX1P2의 태스크 사양(5-29페이지)' 참조 '5-5-1 NJ 시리즈의 태스크 사양(5-46페이지)' 참조 이벤트 태스크: 명령에 의한 실행, 변수의 조건식 일치	프라이머리 정주기 태스크: 1ms*3 정주기 태스크: 2ms(우선도 5) 10ms(우선도 16, 17, 18) 이벤트 태스크: 명령에 의한 실행	설정 전송 시	불가능
태스크 주기 초과 감지		태스크 주기 초과 시 에러를 감지할지 설정합니다.	• 감지(가벼운 이상 레벨의 컨트롤러 이상 발생) • 감지 안 함(감시 정보 레벨로 로그를 남김)	프라이머리 정주기 태스크, 정주기 태스크: 감지 이벤트 태스크: 감지하지 않고 고정	설정 전송 시	불가능
태스크 실행 타임아웃 시간		태스크 실행 타임아웃 시간을 설정합니다. 타임아웃 시간 초과를 검출하면 태스크 실행 타임아웃이 발생합니다.	프라이머리 정주기 태스크, 정주기 태스크 주기 시간×1배~ 주기 시간×5배 이벤트 태스크: 실행 우선도 8인 경우 1~500ms 실행 우선도 48인 경우 1ms~10s	프라이머리 정주기 태스크, 정주기 태스크 주기 시간×5배 이벤트 태스크: 실행 우선도 8인 경우 200ms 실행 우선도 48인 경우 1s	설정 전송 시	불가능
변수 액세스 시간[%]		컨트롤러 외부에서의 변수 액세스 시간의 태스크 주기에 대한 비율을 설정합니다.	프라이머리 정주기 태스크, 정주기 태스크: 1%~50% 이벤트 태스크: 없음	3%	설정 전송 시	불가능

*1. 실행 우선도 5의 정주기 태스크를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*2. NX102 및 NX1P2 CPU 유니트에서는 실행 우선도 16의 정주기 태스크를 사용할 수 없습니다.
*3. NX102 CPU 유니트, NX1P2 CPU 유니트, NJ101 CPU 유니트의 프라이머리 정주기 태스크의 기본값은 2ms입니다.

● **I/O를 제어하는 태스크 설정**

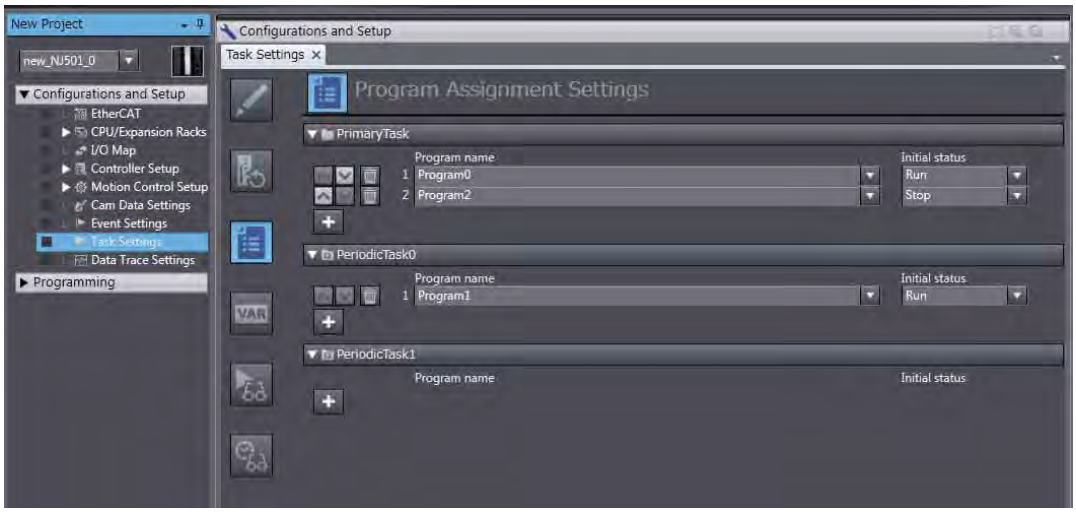
각 입출력의 리프레시 실행 타이밍을 설정합니다.



설정 항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
태스크 이름	유니트/슬레이브를 어느 태스크에서 I/O 리프레시 할지 할당합니다.	프라이머리 정주기 태스크/정주기 태스크	프라이머리 정주기 태스크	설정 전송 시	불가능

● **프로그램 할당 설정**

태스크별 프로그램 할당, 프로그램 실행 순서 및 각 프로그램의 운전 시작 시의 동작을 설정합니다.



설정 항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
프로그램 실행 순서	태스크에 어느 프로그램을 할당할지 및 태스크 내에서 해당 프로그램의 실행 순서를 설정합니다.	프로그램을 실행시키고자 하는 순서로 태스크에 위에서 아래로 할당합니다.	Program0	설정 전송 시	불가능
프로그램의 초기 상태*1	운전 시작 시에 해당 프로그램을 기동하는지 여부를 설정합니다.	기동 정지	기동	설정 전송 시	불가능

*1. CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

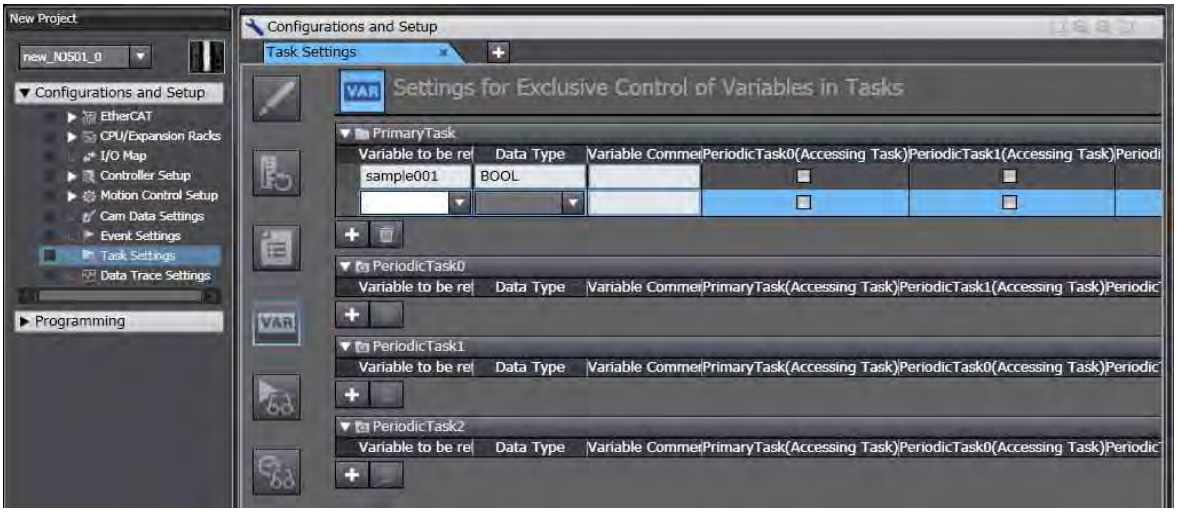
사용상의 주의 사항

- 슬레이브/유니트에 액세스하기 위한 디바이스 변수를 기술한 프로그램은 [I/O를 제어하는 태스크 설정]에서 대상 슬레이브/유니트에 할당한 태스크와 동일한 태스크에 할당하십시오. 다른 태스크에 할당한 경우에 빌드 시에 에러가 됩니다.
- [I/O를 제어하는 태스크 설정]의 기본값은 프라이머리 정주기 태스크입니다. 프라이머리 정주기 태스크 이외에 프로그램을 할당하는 경우에 사전에 [I/O를 제어하는 태스크 설정]을 다시 설정해두면 빌드 시의 에러 발생을 방지할 수 있습니다.
- NX 시리즈 CPU 유니트인 경우에 축에 할당된 슬레이브/유니트를 제어하기 위한 프로그램은 [축 설정] 및 [축 그룹 설정]의 [모션 제어] 설정에 대응하는 태스크에 할당하십시오. 대응하지 않는 태스크에 할당한 경우에 빌드 시에 에러가 됩니다.
  또한, 프로그램 중에 축 및 축 그룹을 지정하는 경우에 [모션 제어] 설정에 대응하는 시스템 정의 변수를 사용하십시오.

[모션 제어] 설정	프로그램 할당에 대응하는 태스크	대응하는 시스템 정의 변수
MC1: 프라이머리 정주기 태스크	프라이머리 정주기 태스크 정주기 태스크(실행 우선도 16)	_MC_AX[], _MC1_AX[], _MC_GRP[], _MC1_GRP[]
MC2: 정주기 태스크(우선도 5)	정주기 태스크(실행 우선도 5)	_MC2_AX[], _MC2_GRP[]

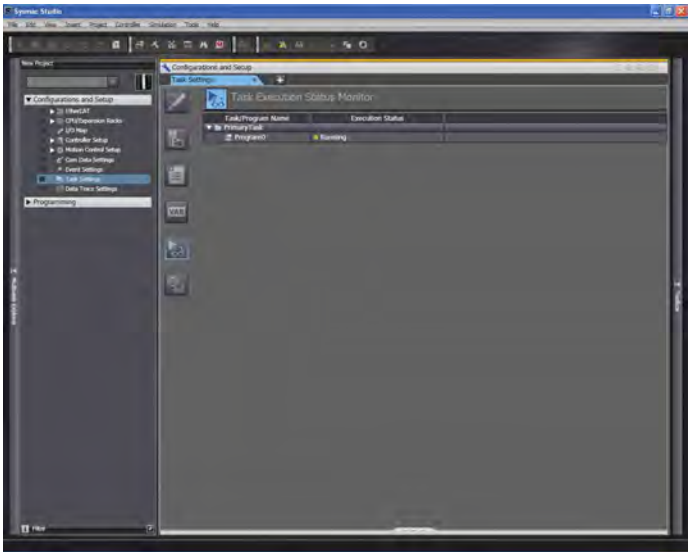
● 변수의 태스크 간 배타 제어 설정

글로벌 변수를 갱신하는 태스크와 참조하는 태스크를 설정합니다.

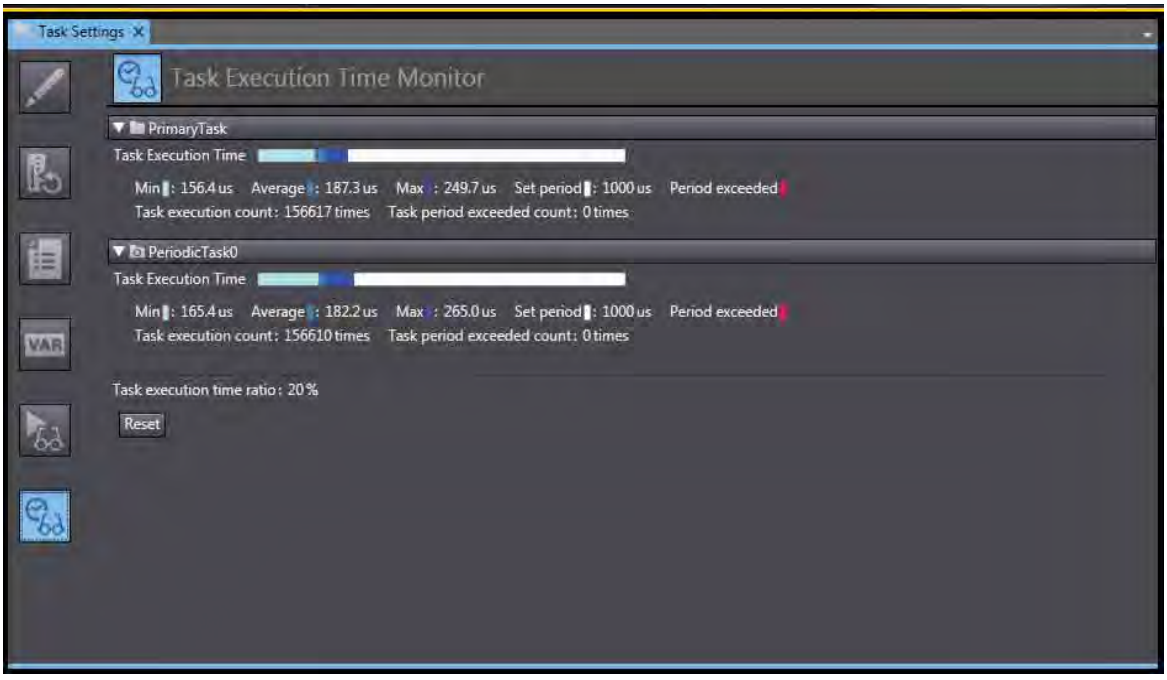


설정 항목	설정 중항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
태스크별	갱신하는 변수	프라이머리 정주기 태스크 또는 정주기 태스크 중에서 갱신 태스크로 하는 변수를 선택합니다.		없음	설정 전송 시	불가능
	데이터 형식	해당 변수의 데이터 형식	없음			
	변수 코멘트	해당 변수의 변수 코멘트	없음			
	참조 태스크	해당 변수의 참조 태스크를 선택합니다.				

- **태스크 실행 상태 모니터링**
  각 프로그램의 실행 상태를 표시합니다.



- **태스크 실행 시간 모니터링**
  각 태스크의 실행 시간을 표시합니다.



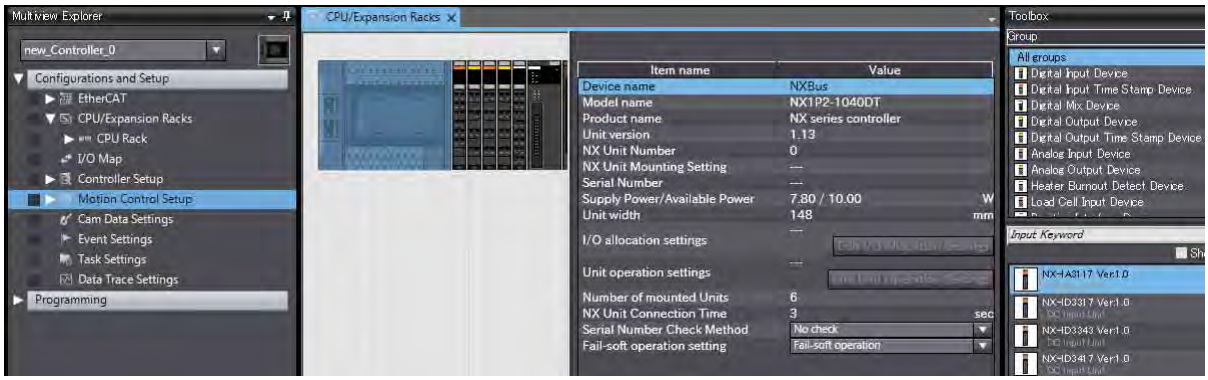
4-2-4 NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정

NX102 및 NX1P2 CPU 유니트에서의 NX 유니트의 구성·설정에 대해 설명합니다. NX701 CPU 유니트 및 NJ 시리즈 CPU 유니트에는 이 설정이 없습니다.

CPU 유니트 상의 NX 유니트의 구성의 작성 및 각 유니트의 설정을 실행합니다.
Sysmac Studio의 CPU·증설 랙 구성 편집 화면에서 설정합니다.

유니트 구성 정보

유니트 구성 정보는 NX102 및 NX1P2 CPU 유니트의 NX 버스에 장착되는 NX 유니트의 대수, 연결 순서, 개체 정보, CPU 유니트의 개체 정보 등 NX 버스 기능 모듈의 구성에 대한 정보입니다.



CPU 유니트를 선택한 경우에 설정 가능한 항목은 다음과 같습니다.

항목	기능 설명	설정값	기본값
시리얼 번호 확인 기준	'설정값=실제 기기'로 설정하면 유니트 구성 정보에 저장된 NX 유니트의 시리얼 번호(설정값)와 실제 NX 유니트의 시리얼 번호(실제 기기 값)를 전원 투입 시 또는 재시작 후에 비교 확인할지 여부를 선택합니다.*1 비교에서 일치하지 않은 경우에는 '유니트 구성 비교 이상'이 발생합니다.	확인 안 함 설정값=실제 기기	확인 안 함
FAIL-SOFT 동작 설정	NX 버스 기능 모듈의 FAIL-SOFT 동작을 실행할지 여부를 설정합니다. FAIL-SOFT 동작을 하는 경우에는 'FAIL-SOFT 실시'를 선택합니다. FAIL-SOFT 동작의 자세한 내용은 '8-2-5 CPU 유니트 상의 NX 유니트 FAIL-SOFT 동작 기능(8-17페이지)'을 참조하십시오.	FAIL-SOFT 실시 정지	FAIL-SOFT 실시

*1. [설정값=실제 기기]의 설정인 경우에 CPU 유니트 상의 NX 유니트를 교환하면 '유니트 구성 비교 이상'이 됩니다. 또한 동일한 형식의 유니트 간의 장착 위치를 변경한 경우에도 '유니트 구성 비교 이상'이 됩니다. [설정값=실제 기기]의 설정 그대로 NX 유니트를 교환 또는 동일한 형식의 유니트 간 장착 위치를 변경해야 하는 경우에는 유니트 구성 정보 수정과 CPU 유니트에 대한 다운로드가 필요합니다. 엄격한 기기 구성 관리가 필요한 경우에만 이 파라메터의 설정값을 [설정값=실제 기기]로 하십시오.

유니트 구성 변경 시의 주의사항

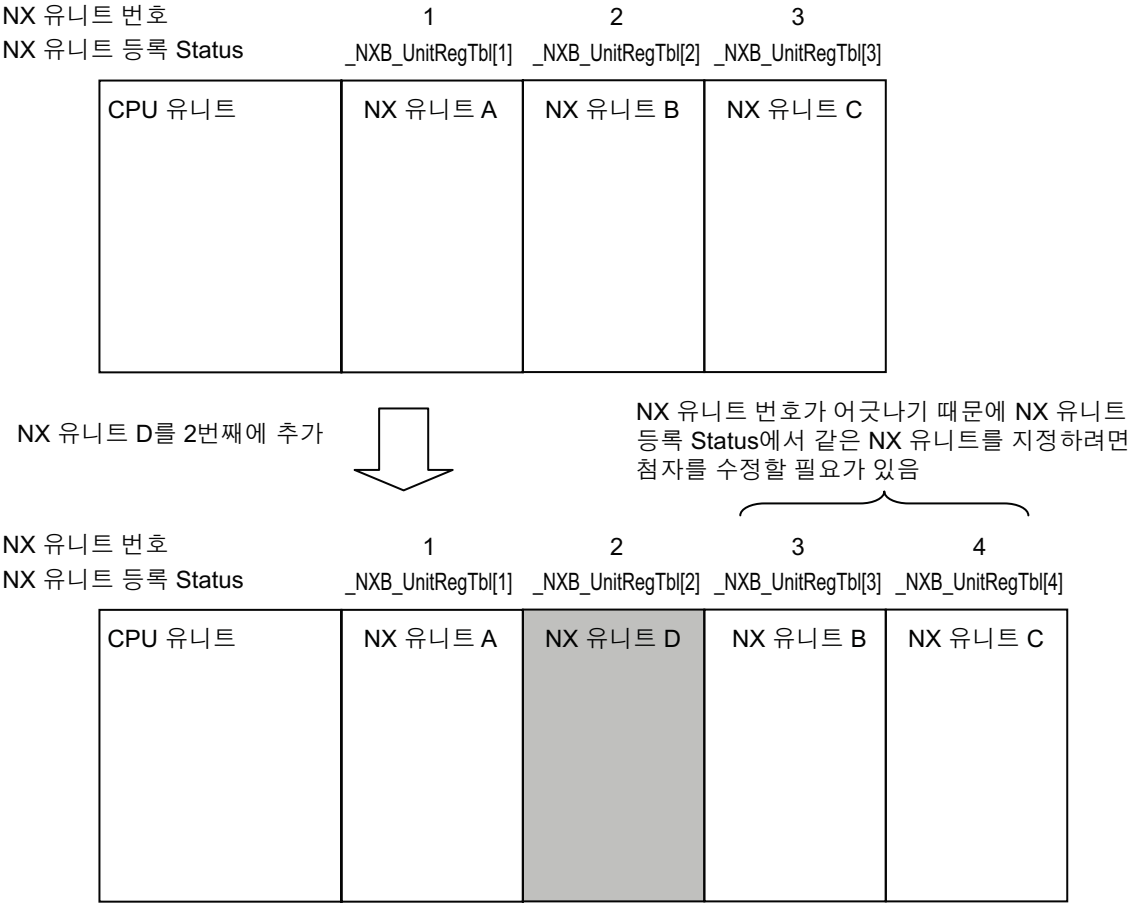
NX102 및 NX1P2 CPU 유니트에서의 NX 유니트의 구성 변경 시의 주의사항을 설명합니다.

- **NX 유니트 번호를 지정하는 I/O 데이터**

  I/O 데이터 중에는 NX 유니트 번호를 지정하여 액세스하는 시스템 정의 변수가 있습니다. CPU 유니트 상의 유니트 구성을 변경하면 프로그램 등에서 지정하는 유니트 번호를 수정해야 합니다.

  NX 유니트 번호를 지정하는 시스템 정의 변수에는 예를 들어 '_NXB_UnitRegTbl(NX 유니트 등록 Status)' 및 '_NXB_UnitErrFlagTbl(NX 유니트 이상 Status)' 등이 있습니다.
  이러한 Status 정보의 시스템 정의 변수는 BOOL 형식의 데이터 배열로 되어 있어서 배열의 첨자에 NX 유니트 번호를 지정합니다.

  예를 들어, NX 유니트 D를 구성에 추가하는 경우에 추가한 유니트에서 우측의 모든 NX 유니트의 NX 유니트 번호에 오차가 생깁니다.



배열의 첨자를 숫자로 지정한 경우에 동일한 NX 유니트를 지정하려면 프로그램 안의 첨자 숫자를 수정해야 합니다.

배열의 첨자의 지정에 유니트에 할당한 변수(_sNXUNIT_ID 형식)를 사용하면 유니트 구성을 변경해도 프로그램의 수정이 필요 없어집니다. _sNXUNIT_ID 형식의 변수는 구조체이고, 멤버 UnitNo에 해당 유니트의 NX 유니트 번호가 저장됩니다.
예를 들어, 앞의 그림에서 NX 유니트 B에 _sNXUNIT_ID 형식의 변수NXUnitB를 할당한 경우에는 _NXB_UnitRegTbl[NXUnitB.UnitNo]와 같은 배열의 첨자를 지정해두면 유니트 구성을 변경해도 프로그램을 수정할 필요가 없습니다.

참고

NX 유니트의 장착 설정 기능

NX 유니트를 줄이는 경우 및 사전에 NX 유니트 추가를 알고 있는 경우에는 NX 유니트의 장착 설정 기능으로 NX 유니트 번호가 바뀌지 않도록 할 수 있습니다.

NX 유니트의 장착 설정 기능에 대해서는 '8-2-2 CPU 유니트 상의 NX 유니트 장착 설정 기능 (8-12페이지)'을 참조하십시오.

4-2-5 NJ 시리즈 CPU 유니트에서의 유니트 구성·유니트 설정

NJ 시리즈 CPU 유니트에서의 설정에 대해 설명합니다. NX 시리즈 CPU 유니트에는 이 설정이 없습니다.

- **유니트 정보**



- **전체 유니트 공통**

  장치명을 설정합니다.
  유니트 편집기에서 유니트를 등록하면 자동적으로 다음과 같은 장치명이 생성됩니다.
  기본값: 'J' 뒤에 등록 순서대로 '01'부터 연속되는 번호
  임의의 이름으로 변경하는 것을 권장합니다.

참고

여기에서 설정한 '장치명'은 '디바이스 변수'를 자동 생성할 때, 'I/O 포트명' 앞에 부가됩니다.

- **고기능 유니트인 경우**

  고기능 유니트의 호기 번호/유니트 번호를 설정하십시오.

사용상의 주의 사항

고기능 유니트의 호기 번호/유니트 번호는 반드시 전면 로터리 스위치의 값과 동일한 값으로 설정하십시오. 만일 일치하지 않은 경우에 실제 기기의 동작은 전면 로터리 스위치의 값에 따라 동작합니다.

- **기본 I/O 유니트인 경우**

  기본 I/O 유니트의 유니트 정보이며, 다음을 설정합니다.

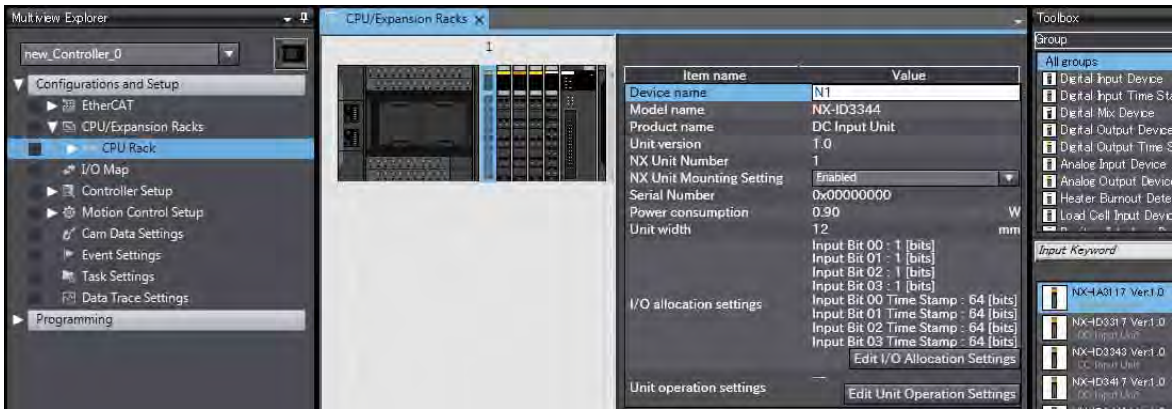
설정 항목	설정 중항목	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
유니트 정보 ※슬롯 단위로 설정	기본 입력 유니트의 입력 응답 시간 설정	기본 입력 유니트의 입력 응답 시간(ON 응답 시간 = OFF 응답 시간을 설정합니다. 필터 없이 또는 0.5ms~32ms까지 단계적으로 설정할 수 있습니다. 이 값을 크게 하면 채터링 및 외부 노이즈의 영향을 잘 받지 않도록 할 수 있습니다. 또한, 반대로 작게 하면(태스크 주기보다 긴 조건에서) 짧은 펄스 입력을 수신할 수 있습니다.	필터 없음 0.5ms 1ms 2ms 4ms 8ms 16ms 32ms	8ms	전원 투입 시(CPU 리셋 시 포함)	불가능

4-3 NX 유니트의 초기 설정

여기에서는 NX102 및 NX1P2 CPU 유니트 상의 NX 유니트의 초기 설정에 대해 설명합니다. NX701 CPU 유니트 및 NJ 시리즈 CPU 유니트에는 이 설정이 없습니다.

4-3-1 NX 유니트의 설정 항목

Sysmac Studio의 CPU·증설 랙 구성 편집 화면에서 설정하는 NX 유니트를 선택하여 설정합니다.



NX 유니트를 선택한 경우에 설정 가능한 항목은 다음과 같습니다.

항목	기능 설명	설정값	기본값
장치명	NX 유니트의 이름입니다.	임의*1	*2
NX 유니트의 장착 설정	NX 유니트의 장착를 유효로 할지, 무효로 할지 여부를 설정합니다. 기능의 자세한 내용은 '8-2-2 CPU 유니트 상의 NX 유니트 장착 설정 기능(8-12페이지)'을 참조하십시오.	유효 무효	유효
I/O 할당 설정	NX 유니트가 갖는 I/O 데이터 중에 어느 I/O 데이터를 교환할지 설정합니다.*3 편집하려면 'I/O 할당 설정 편집' 버튼을 누르십시오. 시스템 유니트에서는 편집할 수 없습니다. I/O 할당 설정의 개요와 편집 방법에 대해서는 '4-3-2 I/O 할당 설정(4-17페이지)'을 참조하십시오.	–	각 NX 유니트의 사용 설명서를 참조
유니트 동작 설정	NX 유니트의 유니트 동작 설정입니다. *4 편집하려면 '유니트 동작 설정 편집' 버튼을 누르십시오. 시스템 유니트에서는 편집할 수 없습니다. 유니트 동작 설정에 대해서는 '4-3-3 유니트 동작 설정(4-19페이지)'을 참조하십시오.	–	각 NX 유니트의 사용 설명서를 참조
유니트 어플리케이션 데이터	NX 유니트 특유의 기능을 동작시키기 위한 데이터입니다. 유니트 어플리케이션 데이터를 갖는 NX 유니트와 갖지 않는 NX 유니트가 있습니다.	–	각 NX 유니트의 사용 설명서를 참조

*1. 여기에서 설정한 '장치명'은 '디바이스 변수'를 자동 생성할 때, 'I/O 포트명' 앞에 부가됩니다.
*2. 'N' 다음에 등록 순서대로 '1'부터 연속되는 번호가 됩니다.
*3. 각 NX 유니트는 I/O 설정의 기본값을 가지고 있습니다. 표준적한 사용 방법을 사용할 때는 편집 작업은 필요 없습니다. 필요에 따라 설정을 변경하십시오.
*4. NX 유니트의 종류에 따라 설정 항목이 결정됩니다. 예를 들어, 디지털 입력 유니트에서는 입력 필터값의 설정, 디지털 출력 유니트에서는 부하 차단 시의 출력값의 설정 등이 있습니다. NX 유니트의 설정 항목 및 그 내용은 각 NX 유니트의 사용 설명서를 참조하십시오.

4-3-2 I/O 할당 설정

여기에서는 I/O 할당 설정의 개요와 편집에 대해 설명합니다.

I/O 할당 설정의 개요

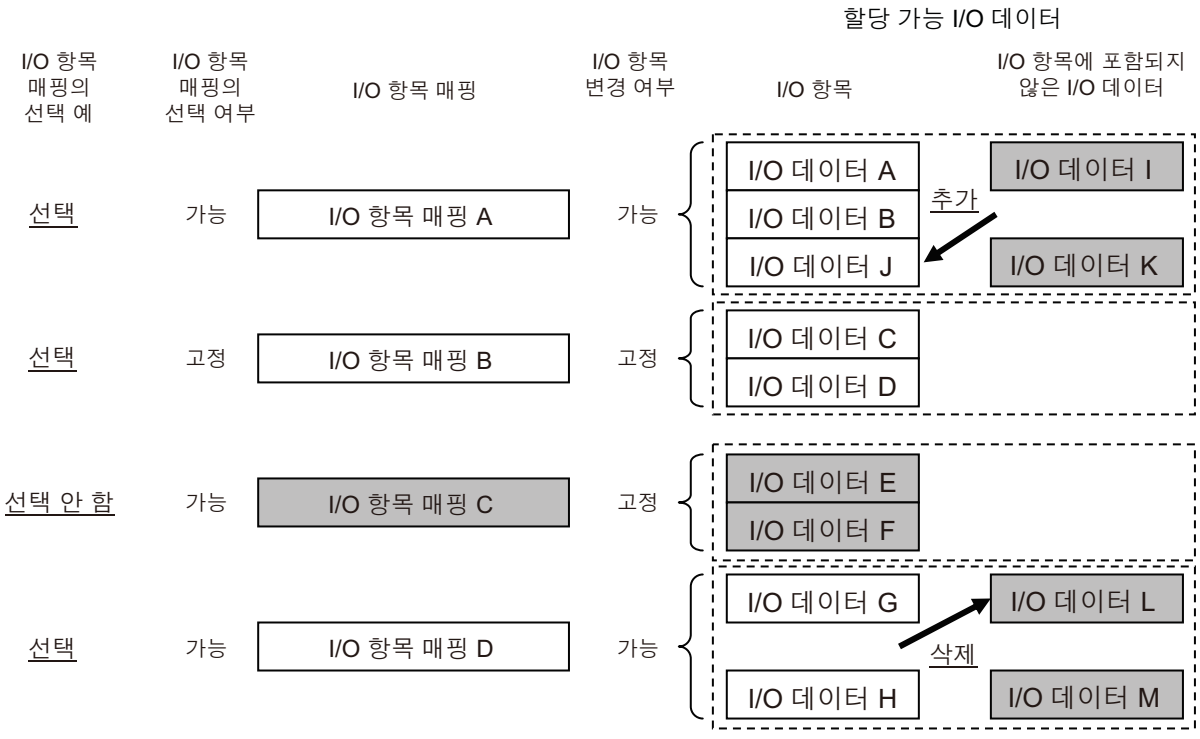
I/O 할당은 I/O 항목 매핑의 선택과 I/O 항목 매핑에 대한 I/O 항목의 등록에 따라 실행합니다.

- **I/O 항목 매핑의 선택**

  I/O 항목 매핑은 I/O 데이터의 설정을 정의한 것입니다. 유니트별로 I/O 항목 매핑이 있습니다.
  선택한 I/O 항목 매핑에 포함되는 I/O 항목의 데이터가 교환됩니다. I/O 항목 매핑의 선택에는 초기값이 설정되어 있습니다. 필요에 따라 I/O 항목 매핑의 선택을 변경하십시오. 선택이 필수인 I/O 항목 매핑에 대해서는 선택으로 고정되어 있고, 선택을 취소할 수 없습니다.

- **I/O 항목의 등록**

  I/O 항목 매핑에 할당된 I/O 데이터를 I/O 항목이라고 합니다.
  각 I/O 항목 매핑의 I/O 항목에는 초기값이 설정되어 있습니다. I/O 항목 매핑의 중에는 I/O 항목을 추가, 삭제하여 변경할 수 있는 것이 있습니다. 또한, I/O 항목 매핑별로 할당 가능한 I/O 데이터가 결정되어 있습니다. 필요에 따라 I/O 항목을 변경하십시오.

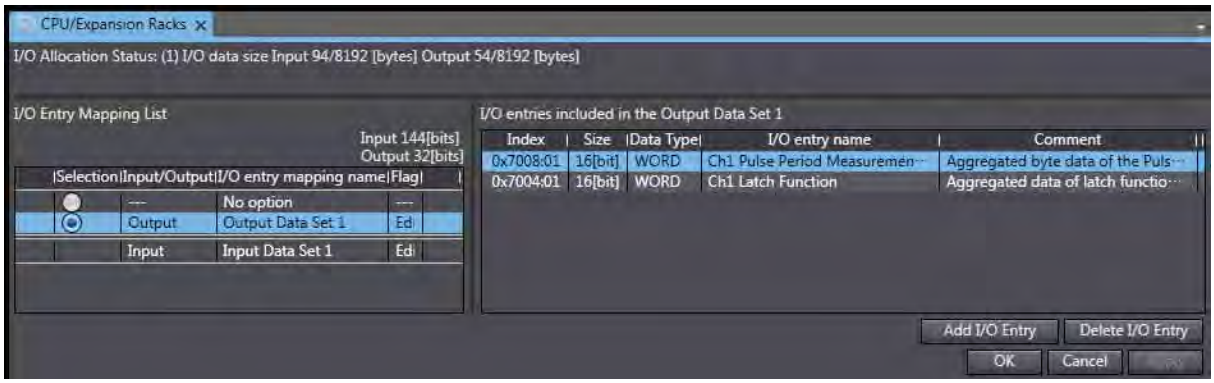


※음영 부분은 데이터 교환에 사용되지 않는 데이터

I/O 할당 설정 편집

필요에 따라 NX 유니트의 I/O 할당을 편집할 수 있습니다.

Sysmac Studio의 CPU·증설 랙 구성 편집 화면의 [I/O 할당 설정 편집] 버튼을 클릭하면 I/O 할당 설정 편집 화면이 표시됩니다.



이름/라벨	설명
I/O 할당 상황	CPU 유니트 전체 I/O 할당 사용 상황이 표시됩니다. • ①I/O 데이터 크기: CPU 유니트 전체에서 할당된 I/O 데이터의 크기가 표시됩니다. 분모는 할당 가능한 최대 크기입니다. I/O 데이터 크기는 I/O 데이터의 메모리 점유 크기를 나타냅니다. 각 I/O 항목 크기의 합계는 반드시 일치하지는 않으므로 주의하십시오. • ②I/O 항목 매핑 수: CPU 유니트 전체에서 할당된 I/O 항목 매핑의 수가 표시됩니다. 분모는 할당 가능한 최대 수입니다.
I/O 항목 매핑 목록	대상 유니트가 갖는 I/O 항목의 매핑 목록입니다. I/O 항목 매핑 목록은 IN/OUT 각각 최대 4개까지 표시됩니다. I/O 항목 매핑 목록의 항목은 다음과 같습니다. • 선택: 할당하는 I/O 항목 매핑의 선택입니다. 할당할 I/O 항목 매핑을 선택하십시오. I/O 항목 매핑을 I/O 할당 정보로 할당하지 않는 경우에는 [선택 안 함]을 선택하십시오. • 입출력: CPU 유니트에서 본 '입력' 또는 '출력'을 나타냅니다. • I/O 항목 매핑명: I/O 항목 매핑의 이름입니다. • 플래그: I/O 항목을 편집할 수 있는 경우에는 '편집 가능'으로 표시됩니다. • I/O 항목을 편집할 수 없는 경우는 '---'로 표시됩니다.
I/O 항목	I/O 항목 매핑 목록에서 선택된 I/O 항목 매핑에 포함되는 I/O 항목을 표시·편집합니다. I/O 항목은 다음과 같습니다. • 인덱스: NX 개체의 인덱스 번호입니다. 0x 다음에 '인덱스 번호: 하위 인덱스 번호'가 표시됩니다. • 크기: I/O 항목의 데이터 크기입니다. • 데이터 형식: I/O 항목의 데이터 형식입니다. • I/O 항목명: I/O 항목명입니다. • 코멘트: I/O 항목의 의미를 나타냅니다.
I/O 할당 설정 편집 화면의 제어 버튼	• [I/O 항목 추가]: 선택한 I/O 항목 매핑에 I/O 항목을 추가합니다. • [I/O 항목 삭제]: 선택한 I/O 항목 매핑에서 선택한 I/O 항목을 삭제합니다. • [OK]: I/O 할당 설정 편집을 유효로 하고 CPU·증설 랙 구성 편집 화면으로 돌아갑니다. • [취소]: I/O 할당 설정 편집을 취소하고 CPU·증설 랙 구성 편집 화면으로 돌아갑니다. • [적용]: I/O 할당 설정 편집을 유효로 하고 I/O 할당 설정 편집을 계속합니다.

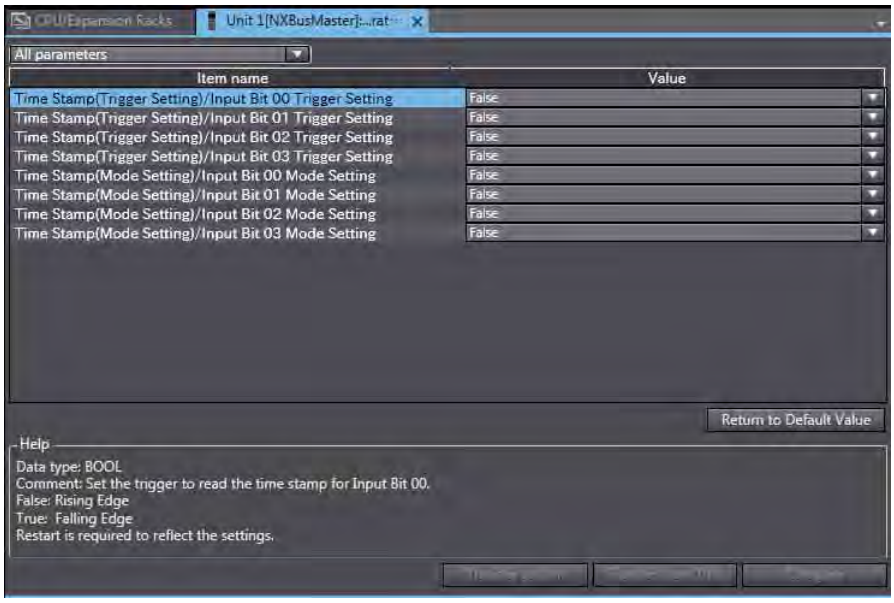
4-3-3 유니트 동작 설정

NX 유니트의 종류에 따라 설정 항목이 결정됩니다. 예를 들어, 디지털 입력 유니트에서는 입력 필터값의 설정, 디지털 출력 유니트에서는 부하 차단 시의 출력값의 설정 등이 있습니다.
NX 유니트의 설정 항목 및 그 내용은 각 NX 유니트의 사용 설명서를 참조하십시오.

유니트 동작 설정 편집

필요에 따라 NX 유니트의 동작 설정을 편집할 수 있습니다.

Sysmac Studio의 CPU·증설 랙 구성 편집 화면의 [유니트 동작 설정 편집] 버튼을 클릭하면 유니트 동작 설정 편집 화면이 표시됩니다.



사용상의 주의 사항

NX 유니트의 연결 위치를 변경하면 유니트 동작 설정이 기본값으로 돌아갑니다.
필요에 따라 유니트의 동작 설정을 다시 전송하십시오.

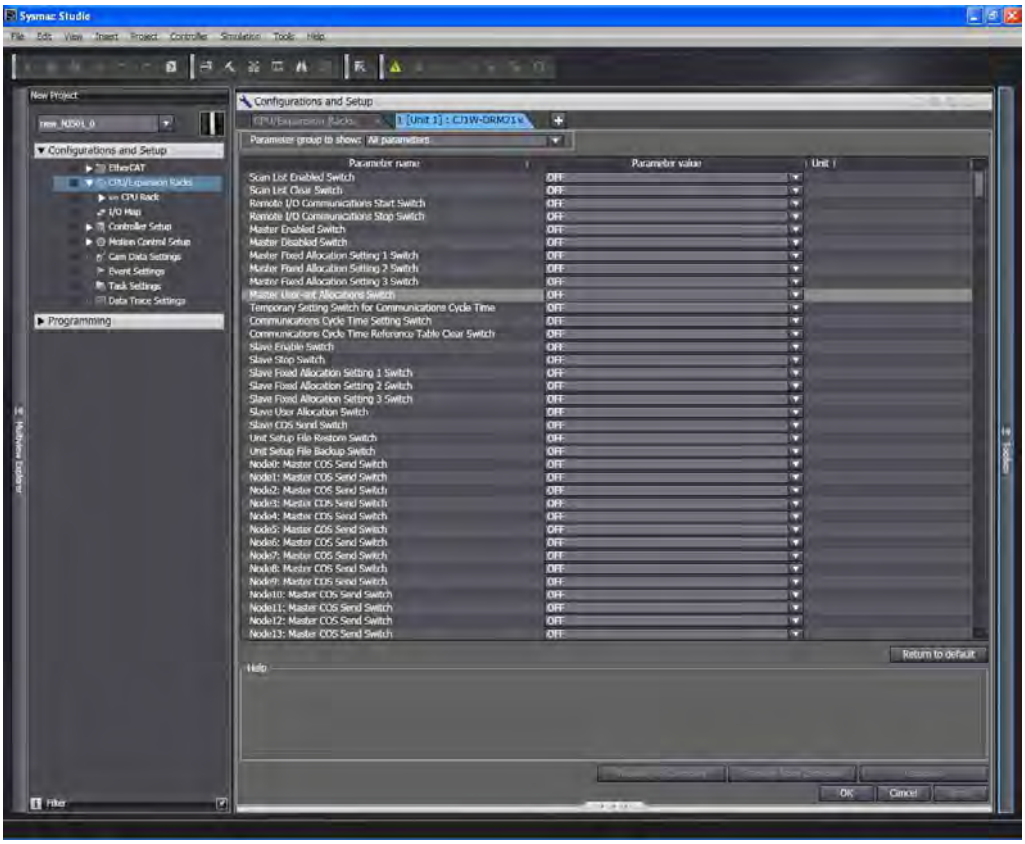
4-4 고기능 유니트의 초기 설정

여기에서는 NJ 시리즈 CPU 유니트에서의 고기능 유니트의 초기 설정에 대해 설명합니다. NX 시리즈 CPU 유니트에는 이 설정이 없습니다.
고기능 유니트의 초기 설정은 다음 중 하나의 방법으로 설정합니다.

방법 1: Sysmac Studio의 유니트 설정 화면에서 설정

1 **유니트 구성·유니트 설정에서 유니트를 선택합니다.**

2 **다음과 같은 유니트 설정 화면에서 설정합니다.**



3 **CPU 유니트에 온라인 연결하여 설정을 전송합니다.**

방법 2: Sysmac Studio의 I/O 맵에서 I/O 포트에 대해 값을 설정

1 **Sysmac Studio의 I/O 맵에서 I/O 포트에 대해 값을 설정합니다.**

2 **해당 유니트를 재시작 또는 컨트롤러 리셋 또는 컨트롤러의 전원 재투입을 실행합니다.**

방법3: Sysmac Studio에서 CJ 유니트용 디바이스 변수에 초기값을 설정

1 **Sysmac Studio에서 CJ 유니트용 디바이스 변수의 초기값을 설정합니다.**

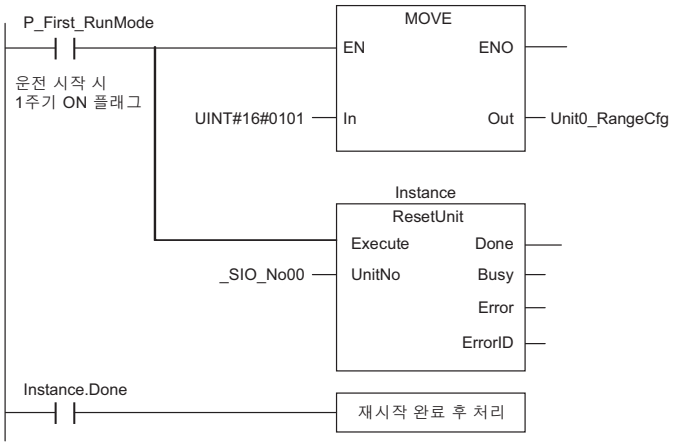
2 **Sysmac Studio에서 변수 테이블을 CPU 유니트에 전송합니다.**
이 때 '유지 변수의 현재값을 초기화'를 선택합니다.

3 **해당 유니트를 재시작 또는 컨트롤러 리셋 또는 컨트롤러의 전원 재투입을 실행합니다.**

방법 4: 명령에서 CJ 유니트용 디바이스 변수에 값을 설정

1 **사용자 프로그램 상에서 운전 시작 시에 MOVE(전송) 명령 등에 의해 CJ 유니트용 디바이스 변수에 대해 값을 설정하고, 해당 유니트를 재시작시킵니다.**

예)



안전상의 중요 사항

고기능 유니트의 설정을 변경하여 재시작할 때는 연결할 대상 기기의 안전을 충분히 확인한 후 실행하십시오.

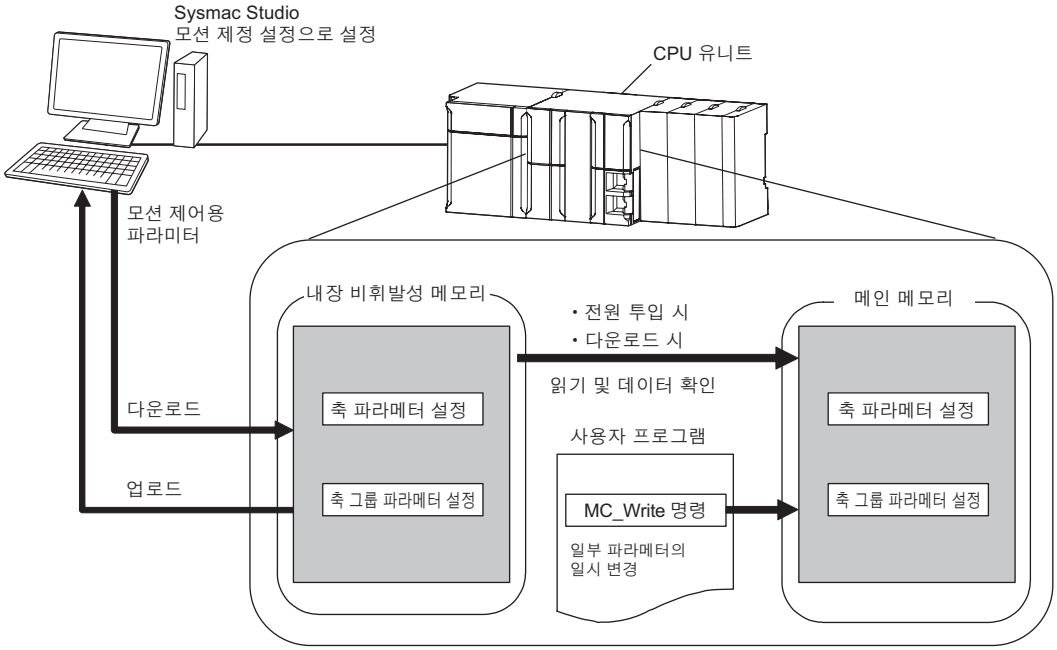
4-5 모션 제어 기능 모듈의 초기 설정

여기에서는 모션 제어 기능 모듈의 초기 설정에 대해 설명합니다.

4-5-1 개요

모션 제어 기능 모듈의 초기 설정을 '모션 제어 파라메터'라고 합니다.
'모션 제어 파라메터'에는 다음과 같은 내용이 있습니다.

- 축 파라메터: 단축 제어를 실행하기 위한 설정
- 축 그룹 파라메터: 다축 협조 제어를 실행하기 위한 설정

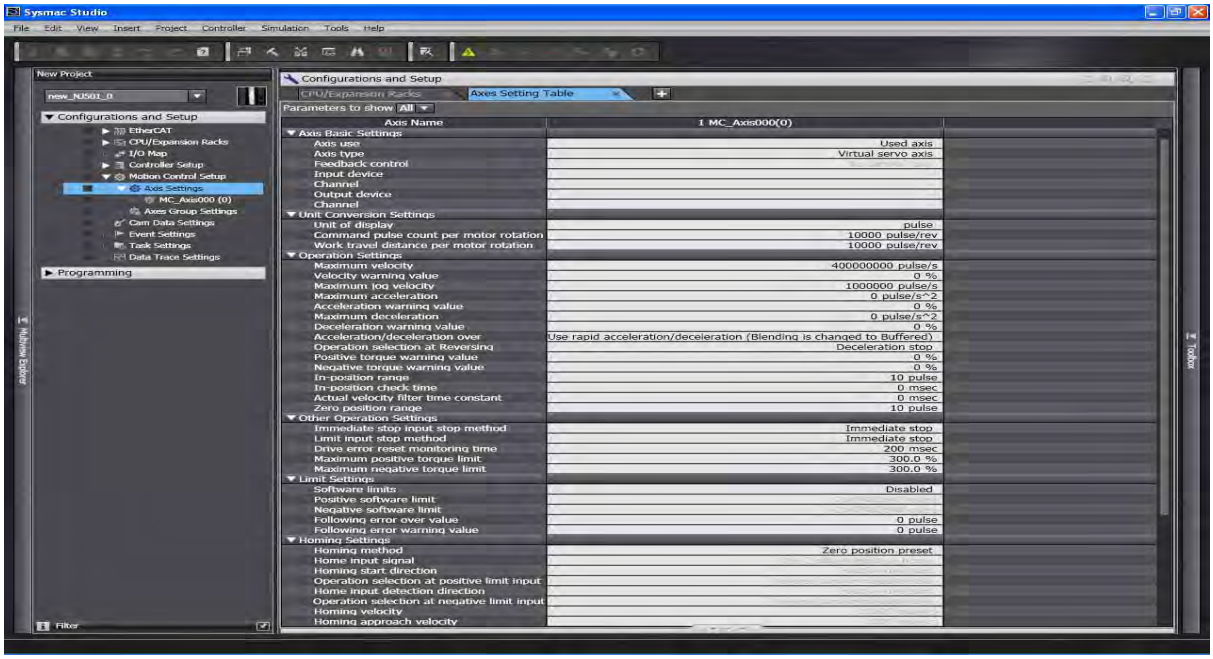


4-5-2 설정 방법

모션 제어용 파라메터의 설정은 다음 중 하나를 통해 설정합니다.

방법1: Sysmac Studio의 모션 제어 설정에서 설정

Sysmac Studio에서 다음과 같은 [구성·설정]–[모션 제어 설정]–[축 설정]을 마우스 오른쪽 버튼으로 클릭하고 [전체 축 설정 테이블]에서 설정합니다.



모션 제어용 파라메터를 CPU 유니트에 다운로드하면 CPU 유니트 내장 비휘발성 메모리에 저장됩니다. 전원 투입 시 또는 다운로드 시에 다운로드된 설정 데이터가 유효가 됩니다.

방법2: MC_Write 명령에서 설정

모션 제어용 파라메터의 일부는 MC_Write 명령으로 일시적으로 재기입할 수 있습니다.

자세한 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

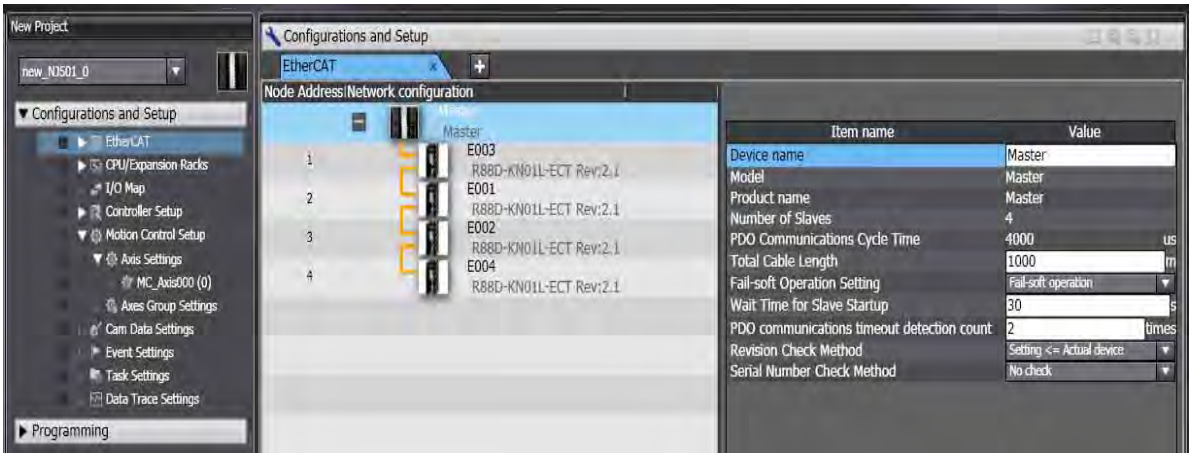
4-6 EtherCAT 마스터 기능 모듈의 초기 설정

여기에서는 EtherCAT 마스터 기능 모듈의 초기 설정에 대해 설명합니다.

EtherCAT 마스터 기능 모듈의 초기 설정에는 다음 내용 등이 있습니다.

- 장치명
- 케이블 총 길이
- FAIL-SOFT 동작 설정
- 전체 슬레이브 가입 대기 시간
- PDO 통신 연속 타임아웃 검출 횟수
- 개정 번호 확인 기준
- 시리얼 번호 확인 기준

Sysmac Studio에서 [구성·설정]–[EtherCAT]을 선택하고, Master를 선택하면 EtherCAT 마스터 기능 모듈의 초기 설정 화면이 표시됩니다.



자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"를 참조하십시오.

4-7 EtherNet/IP 기능 모듈의 초기 설정

여기에서는 EtherNet/IP 기능 모듈의 초기 설정에 대해 설명합니다.

EtherNet/IP 기능 모듈의 초기 설정에는 다음과 같은 내용이 있습니다.

- TCP/IP 설정
- LINK 설정
- FTP 설정
- NTP 설정
- SNMP 설정
- SNMP 트랩 설정
- FINS 설정

Sysmac Studio에서 [구성·설정]–[컨트롤러 설정]–[내장 EtherNet/IP 포트 설정]에서 설정합니다.

자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.

사용상의 주의 사항

NX701-□□00인 경우에 FINS 설정은 없습니다.

4-8 내장 I/O 초기 설정

여기에서는 NX1P2 CPU 유니트의 내장 I/O의 초기 설정에 대해 설명합니다.

내장 I/O의 초기 설정에는 다음과 같은 내용이 있습니다.
- 입력 필터 설정
- 부하 차단 시 출력 설정

Sysmac Studio에서 [구성·설정]–[컨트롤러 설정]–[내장 I/O 설정]에서 설정합니다.

자세한 내용은 "NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)"를 참조하십시오.

4-9 옵션 보드 초기 설정

여기에서는 NX1P2 CPU 유니트의 옵션 보드의 초기 설정에 대해 설명합니다.

옵션 보드의 초기 설정에는 다음과 같은 내용이 있습니다.

- 구성
- 옵션 보드 1 시리얼 설정
- 옵션 보드 2 시리얼 설정

Sysmac Studio에서 [구성·설정]–[컨트롤러 설정]–[옵션 보드 설정]에서 설정합니다.

자세한 내용은 "NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)"를 참조하십시오.

4-10 CJ 유니트용 메모리 설정

여기에서는 NX102 CPU 유니트 및 NX1P2 CPU 유니트의 CJ 유니트용 메모리의 초기 설정에 대해 설명합니다.

CJ 유니트용 메모리의 초기 설정에는 다음과 같은 내용이 있습니다. 각각 유효인지 무효인지와 유효인 경우의 크기를 설정합니다.

- CIO
- WR
- HR
- DM
- EM



참고

NX1P2 CPU 유니트에서는 EM을 사용할 수 없습니다.

Sysmac Studio에서 [구성·설정]–[컨트롤러 설정]–[메모리 설정]에서 설정합니다.

4-10-1 설정 순서

CJ 유니트용 메모리 설정 순서는 다음과 같습니다.

번호	순서	내용	참조
1	사용 메모리 결정	연결하는 기기와의 데이터 교환에서 사용하는 CJ 유니트용 메모리의 영역 종류 및 CH 수를 결정함	연결하는 기기의 설명서 및 기술 자료
2	CJ 유니트용 메모리 설정	Sysmac Studio로 연결하는 기기가 사용하는 CJ 유니트용 메모리의 메모리 종류 및 CH 수를 설정함	'4-10-2 설정 화면(4-29페이지)'
3	프로그래밍	CJ 유니트용 메모리를 사용하는 사용자 프로그램을 작성함 CH 수가 부족하면 번호 2로 돌아가서 다시 설정함	'제6장 프로그래밍'
4	프로젝트 다운로드	Sysmac Studio를 통해 프로젝트를 다운로드함	"Sysmac Studio Version 1 조작 설명서(SBCA-470)"
5	동작 확인 및 운전	작성한 사용자 프로그램 및 연결하는 기기의 동작을 확인함	'제7장 동작 확인 및 운전'

4-10-2 설정 화면

[구성·설정]의 [컨트롤러 설정]의 [메모리 설정]을 통해 표시되는 CJ 유니트용 메모리 설정 화면에서 설정합니다.

NX1P2 CPU 유니트를 사용한 경우의 설정 화면은 다음과 같습니다.



4-10-3 설정 내용

NX102 CPU 유니트인 경우의 설정 내용은 다음과 같습니다.

설정 항목	설정 중항목	설정 소항목	기능 설명	설정값	기본값	반영 타이밍	운전모드 중 재기입
CJ 유니트용 메모리 설정	CIO*1	유효	영역 종류 CIO의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 CIO의 크기를 지정합니다.	1-6144	6144	설정 전송 시	불가능
	WR*1	유효	영역 종류 WR의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 WR의 크기를 지정합니다.	1-512	512	설정 전송 시	불가능
	HR*2	유효	영역 종류 HR의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 HR의 크기를 지정합니다.	1-1536	512	설정 전송 시	불가능
	DM*2	유효	영역 종류 DM의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 DM의 크기를 지정합니다.	1-32768	4096	설정 전송 시	불가능
	EM0-EM18*2	유효	영역 종류 EM의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 EM의 크기를 지정합니다.	1-32768	32768	설정 전송 시	불가능

*1. 유지 속성이 없는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*2. 유지 속성이 있는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.

NX1P2 CPU 유니트인 경우의 설정 내용은 다음과 같습니다.

설정 항목	설정 중항목	설정 소항목	기능 설명	설정값	기본값	반영 타이밍	운전모드 중 재기입
CJ 유니트용 메모리 설정	CIO*1	유효	영역 종류 CIO의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 CIO의 크기를 지정합니다.	1-6144	6144	설정 전송 시	불가능
	WR*1	유효	영역 종류 WR의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 WR의 크기를 지정합니다.	1-512	512	설정 전송 시	불가능
	HR*2	유효	영역 종류 HR의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 HR의 크기를 지정합니다.	1-1536	512	설정 전송 시	불가능
	DM*2	유효	영역 종류 DM의 CJ 유니트용 메모리 생성 여부를 설정합니다.	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	영역 종류 DM의 크기를 지정합니다.	1-16000	4096	설정 전송 시	불가능

*1. 유지 속성이 없는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*2. 유지 속성이 있는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.

참고

CJ 유니트용 메모리 설정은 NX701-□□20에서도 가능합니다. NX701-□□20의 CJ 유니트용 메모리 설정 내용은 다음과 같습니다.

설정 항목	설정 중항목	설정 소항목	설정값	기본값	반영 타이밍	운전 모드 중 재기입
CJ 유니트용 메모리 설정	CIO*1	유효	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	1-6144	6144	설정 전송 시	불가능
	WR*1	유효	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	1-512	512	설정 전송 시	불가능
	HR*2	유효	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	1-1536	512	설정 전송 시	불가능
	DM*2	유효	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	1-32768	32768	설정 전송 시	불가능
	EM0-EM18*2	유효	유효 무효	무효	설정 전송 시	불가능
		크기 (CH 수)	1-32768	32768	설정 전송 시	불가능

*1. 유지 속성이 없는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*2. 유지 속성이 있는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.

5

태스크 설계

이 장에서는 태스크의 구조와 종류에 대해 설명합니다.

5-1 태스크 설계 순서의 개요 5-3

5-2 태스크의 개요 5-5
5-2-1 태스크의 개념 5-5
5-2-2 태스크 관련 명령 5-7
5-2-3 태스크 관련 시스템 정의 변수 5-7

5-3 NX701의 태스크 사양과 기본 동작 5-9
5-3-1 NX701의 태스크 사양 5-9
5-3-2 NX701의 태스크 분할 개념 5-10
5-3-3 NX701의 태스크 기본 동작 5-10
5-3-4 NX701의 이벤트 태스크의 실행 조건 5-19
5-3-5 NX701의 이벤트 태스크 실행 타이밍 5-24
5-3-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-28

5-4 NX102 및 NX1P2의 태스크 사양과 기본 동작 5-29
5-4-1 NX102 및 NX1P2의 태스크 사양 5-29
5-4-2 NX102 및 NX1P2의 태스크 분할의 개념 5-30
5-4-3 NX102 및 NX1P2의 태스크 기본 동작 5-30
5-4-4 NX102 및 NX1P2의 이벤트 태스크 실행 조건 5-36
5-4-5 NX102 및 NX1P2의 이벤트 태스크 실행 타이밍 5-41
5-4-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-45

5-5 NJ 시리즈의 태스크 사양과 기본 동작 5-46
5-5-1 NJ 시리즈의 태스크 사양 5-46
5-5-2 NJ 시리즈의 태스크 분할의 개념 5-47
5-5-3 NJ 시리즈의 태스크 기본 동작 5-47
5-5-4 NJ 시리즈의 이벤트 태스크 실행 조건 5-55
5-5-5 NJ 시리즈의 이벤트 태스크 실행 타이밍 5-60
5-5-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작 5-64

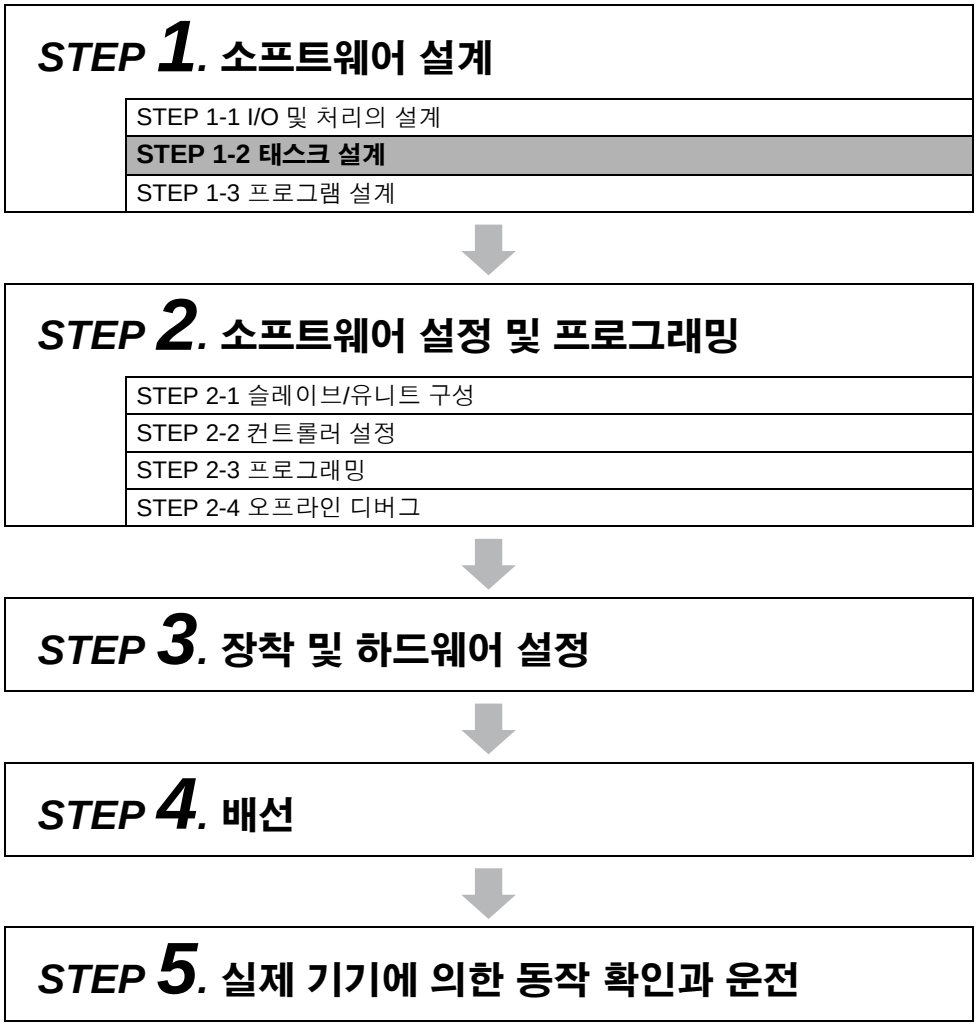
5-6 태스크 이외의 서비스 5-65
5-6-1 태스크 이외의 서비스 실행 우선도와 실행 순서 5-67
5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍 5-70
5-6-3 옵션 보드 서비스의 처리 내용과 실행 타이밍 5-74
5-6-4 시스템 서비스의 처리 내용과 실행 타이밍 5-75

5-7 태스크 관련 할당과 설정 5-78
5-7-1 I/O 리프레시의 태스크에 대한 할당 5-78
5-7-2 프로그램의 태스크에 대한 할당 5-86
5-7-3 태스크의 설정 항목 5-87
5-8 변수의 동시성 확보 5-90
5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법 5-90
5-8-2 컨트롤러 외부에서의 변수 액세스 5-96
5-9 태스크 관련 이상 5-100
5-10 태스크 실행 상태, 태스크 실행 시간의 모니터링 5-103
5-11 태스크 설계 방법 및 입출력 응답 시간 5-108
5-11-1 태스크 실행 시간 확인 5-108
5-11-2 태스크 설계의 예 5-111
5-11-3 시스템으로서의 입출력 응답 시간 5-112

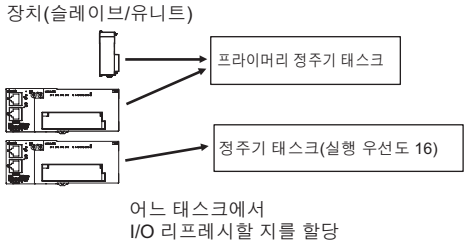
5-1 태스크 설계 순서의 개요

여기에서는 태스크 설계 순서의 개요를 나타냅니다.

전체 순서 중에서 다음과 같이 회색으로 표시한 위치에 대응합니다.



자세한 내용은 🕮'1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.

태스크 설계	참조
● **태스크 구성의 설계** 제어 대상 장치의 필요한 입출력 응답 성능에서 태스크 구성을 설계합니다.	'5-3-3 NX701의 태스크 기본 동작(5-10페이지)' '5-4-3 NX102 및 NX1P2의 태스크 기본 동작(5-30페이지)' '5-5-3 NJ 시리즈의 태스크 기본 동작(5-47페이지)' '5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'
● **어느 장치(슬레이브/유니트)를 프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 5, 16) 중 어느 것으로 I/O 리프레시할지 설계**  	'5-7-1 I/O 리프레시의 태스크에 대한 할당(5-78페이지)'
● **어느 프로그램을 프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 5, 16~18), 이벤트 태스크(실행 우선도8, 48) 중 어디에 할당할지 설계**	'5-7-2 프로그램의 태스크에 대한 할당(5-86페이지)'
● **태스크 간의 변수 배타 방법을 설계** 다른 태스크 사이에서 동일한 글로벌 변수를 사용하는 경우에 태스크 간의 변수 배타 방법을 설계합니다.	'5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'
● **컨트롤러 외부에서 변수를 액세스할 때의 태스크 설계** 컨트롤러 외부에서 CPU 유니트 내의 변수에 액세스하는 경우에 해당 변수 액세스와 어떤 특정 태스크의 사용자 프로그램 실행을 동기시킬 때 설계합니다. 변수 액세스에는 EtherNet/IP 태그 데이터 링크가 포함됩니다.	'5-8-2 컨트롤러 외부에서의 변수 액세스(5-96페이지)'

Sysmac Studio에서의 태스크 설정

태스크 설정	참조
● **PLC 기본 모듈 관련 초기 설정** 태스크 설정: 태스크 주기, 입출력 설정, 프로그램 할당 태스크 간 인터페이스 설정 등	'4-2 PLC 기능 모듈의 초기 설정(4-4페이지)'

Sysmac Studio에서의 오프라인 디버그

데스크탑 동작 확인	참조
● **시뮬레이터(가상 PLC)에서 시퀀스 제어 및 모션 제어를 데스크탑 디버그합니다.** ● **태스크 실행 시간 모니터링 화면에서 태스크 실행 시간을 모니터링합니다.**	'제7장 동작 확인 및 운전(7-1페이지)' '5-10 태스크 실행 상태, 태스크 실행 시간의 모니터링(5-103페이지)'

5-2 태스크의 개요

여기에서는 태스크의 개요에 대해 설명합니다.

5-2-1 태스크의 개념

태스크는 I/O 리프레시 및 사용자 프로그램 실행 등의 일련의 처리에서 실행 조건과 실행 순서를 지정하는 기능입니다.

태스크에는 실행 조건 및 실행 우선도에 따라 다음 3종류가 있습니다.

태스크의 종류	태스크 수	태스크 실행 우선도	정의	주요 처리 내용
프라이머리 정주기 태스크	1	4	설정한 태스크 주기에 1회 실행합니다. 어떤 태스크보다 최우선으로 실행합니다. 프라이머리 정주기 태스크의 모션 제어 명령 및 EtherCAT의 통신은 프라이머리 정주기 태스크의 주기로 실행합니다.	I/O 리프레시, 사용자 프로그램 실행, 모션 제어
정주기 태스크	0~1	5*1	설정한 태스크 주기에 1회 실행합니다. 프라이머리 정주기 태스크의 다음과 같은 우선도로 실행합니다. 실행 우선도 5인 정주기 태스크의 모션 제어 명령 및 EtherCAT의 통신은 실행 우선도 5의 정주기 태스크의 주기로 실행합니다.	I/O 리프레시, 사용자 프로그램 실행, 모션 제어
	0~3	16*2 , 17, 18	설정한 태스크 주기에 1회 실행합니다. 실행 우선도 16인 정주기 태스크의 모션 제어 명령 및 EtherCAT의 통신은 프라이머리 정주기 태스크의 주기로 실행합니다.	태스크 실행 우선도에 따라 처리 내용이 다릅니다. 실행 우선도 16: I/O 리프레시, 사용자 프로그램 실행 실행 우선도 17, 18: 사용자 프로그램 실행
이벤트 태스크	0~32	8, 48	지정한 실행 조건이 성립했을 때 1회만 실행합니다.	사용자 프로그램 실행

*1. 실행 우선도 5의 정주기 태스크를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*2. NX102 및 NX1P2 CPU 유니트에서는 실행 우선도 16의 정주기 태스크를 사용할 수 없습니다.

버전 관련 정보

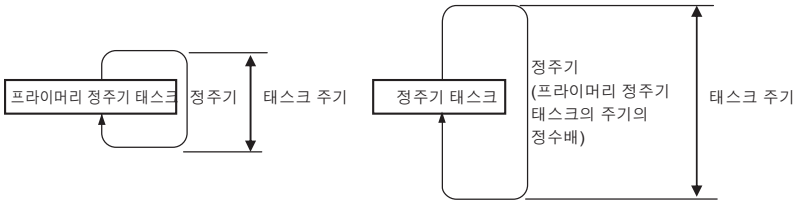
이벤트 태스크는 CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상과의 조합에서 사용할 수 있습니다.

참고

NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)로 모션 제어를 실행할 수 있습니다. 2가지 모션 제어를 구별해야 하는 경우에 각각 '모션 제어 1', '모션 제어 2'라고 합니다.

● **프라이머리 정주기 태스크와 정주기 태스크**

프라이머리 정주기 태스크 및 정주기 태스크는 모두 정주기로 실행됩니다.
프라이머리 정주기 태스크 및 정주기 태스크가 정주기로 실행되는 간격인 것을 '태스크 주기'라고 합니다.



각 태스크에는 1~128개의 프로그램을 할당할 수 있습니다. 할당된 프로그램은 할당 순서대로 실행됩니다. 그것을 '사용자 프로그램 실행'이라고 합니다.
CJ 유니트, EtherCAT 슬레이브와의 데이터 교환을 I/O 리프레시라고 합니다.
I/O 리프레시는 슬레이브/유니트별로 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5, 16)에 할당할 수 있습니다. 기본값은 모든 슬레이브/유니트의 I/O 리프레시가 프라이머리 정주기 태스크에 대한 할당으로 되어 있습니다.

● **이벤트 태스크**

이벤트 태스크는 지정한 실행 조건이 성립했을 때 1회만 실행됩니다. 이벤트 태스크의 실행 조건에는 다음 2종류가 있습니다.

실행 조건	사양
명령에 의한 실행	이벤트 태스크 실행 명령 ActEventTask가 실행되면 이벤트 태스크가 실행됩니다.
변수의 조건 일치에 의한 실행	지정한 변수가 사전에 설정한 조건에 일치하면 이벤트 태스크가 실행됩니다.

각 태스크에는 1~128개의 프로그램을 할당할 수 있습니다. 할당된 프로그램은 할당 순서대로 실행됩니다.

사용상의 주의 사항

- 이벤트 태스크에서는 I/O 리프레시 및 모션 제어가 실행되지 않습니다. 따라서 I/O 제어를 동반하는 프로그램 및 모션 제어 명령을 포함하는 프로그램을 이벤트 태스크에 할당할 수 없습니다.
- 이벤트 태스크는 태스크 주기마다 반복해서 실행되지 않습니다. 따라서 복수의 태스크 주기에 걸쳐 실행되는 명령을 포함하는 프로그램을 이벤트 태스크에 할당할 수 없습니다. 복수의 태스크 주기에 걸쳐 실행되는 명령은 SD 메모리 카드 명령 및 통신 명령 등 일부 기본 명령, 모든 모션 제어 명령, 모든 시뮬레이션 명령입니다. 이벤트 태스크에서 사용할 수 없는 기본 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

참고

NX701 CPU 유니트, NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 태스크의 동작이 다릅니다. 자세한 내용은 '5-3 NX701의 태스크 사양과 기본 동작(5-9페이지)', '5-4 NX102 및 NX1P2의 태스크 사양과 기본 동작(5-29페이지)' 및 '5-5 NJ 시리즈의 태스크 사양과 기본 동작(5-46페이지)'을 참조하십시오.

5-2-2 태스크 관련 명령

태스크 관련 명령으로 준비된 것은 다음과 같습니다.

명령	명령 이름	개요	
GetMyTaskStatus	로컬 태스크 상태 읽기	로컬 태스크의 다음과 같은 태스크 상태를 읽습니다. 이전 태스크 실행 시간, 태스크 실행 시간 최대값, 태스크 실행 시간 최소값, 태스크 실행 횟수, 태스크 주기 초과 플래그, 태스크 주기 초과 횟수	
GetMyTaskInterval	로컬 태스크 설정 주기 읽기	로컬 태스크의 태스크 주기를 읽습니다.	
Task_IsActive	태스크 실행 중 판정	지정한 태스크가 실행 중인지 여부를 반환합니다.	
Lock	태스크 간 배타 잠금	잠금 구간을 시작합니다.	다른 태스크의 동일한 번호의 잠금 구간은 동시에 실행할 수 없게 됩니다.
Unlock	태스크 간 배타 잠금 해제	잠금 구간을 종료합니다.	
ActEventTask	이벤트 태스크 기동	지정한 이벤트 태스크를 기동합니다.	

5-2-3 태스크 관련 시스템 정의 변수

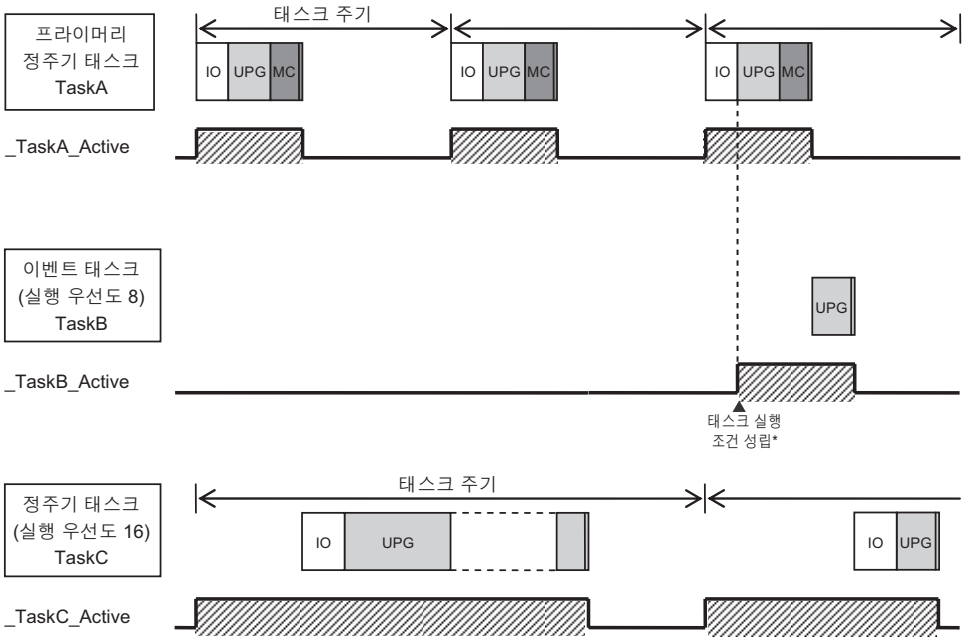
다음과 같이 태스크의 상태를 나타내는 시스템 정의 변수가 태스크별로 준비되어 있습니다.
단, 실시간성 및 각 기능 모듈의 이상 요인 Status와의 동시성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크 상태를 샘플링할 때 사용합니다.
사용자 프로그램에서 태스크의 상태를 참조하는 경우에는 GetMyTaskStatus 명령, Task_IsActive 명령을 사용하십시오.

변수명	이름	기능	데이터 형식	R/W
_<태스크 이름>_Active	태스크 실행 중 플래그	태스크 실행 중에는 TRUE가 됩니다. 태스크 실행 완료에서 태스크 주기의 종료까지 FALSE가 됩니다.	BOOL	R
_<태스크 이름>_LastExecTime	이전 태스크 실행 시간	이전에 실행한 태스크 실행 시간을 나타냅니다.	TIME	R
_<태스크 이름>_MaxExecTime	태스크 실행 시간 최대값	태스크 실행 시간의 최대값을 나타냅니다.	TIME	R
_<태스크 이름>_MinExecTime	태스크 실행 시간 최소값	태스크 실행 시간의 최소값을 나타냅니다.	TIME	R
_<태스크 이름>_ExecCount	태스크 실행 횟수	태스크의 실행 횟수를 표시합니다. 현재값이 데이터 형식의 최대값을 초과하면 현재값을 0으로 되돌려서 카운트를 계속합니다.	UDINT	R
_<태스크 이름>_Exceeded	태스크 주기 초과 플래그	태스크 주기를 초과하면 태스크 실행의 완료 시에 TRUE가 됩니다. 태스크 주기 이내에 태스크 실행이 완료되면 FALSE가 됩니다.	BOOL	R
_<태스크 이름>_ExceedCount	태스크 주기 초과 횟수	태스크 주기를 초과한 횟수가 저장됩니다. 현재값이 데이터 형식의 최대값을 초과하면 현재값을 0으로 되돌려서 카운트를 계속합니다.	UDINT	R

(주의) 예: 태스크 이름 'MainTask'의 태스크 주기 초과 플래그는 “_MainTask_Exceeded”입니다.

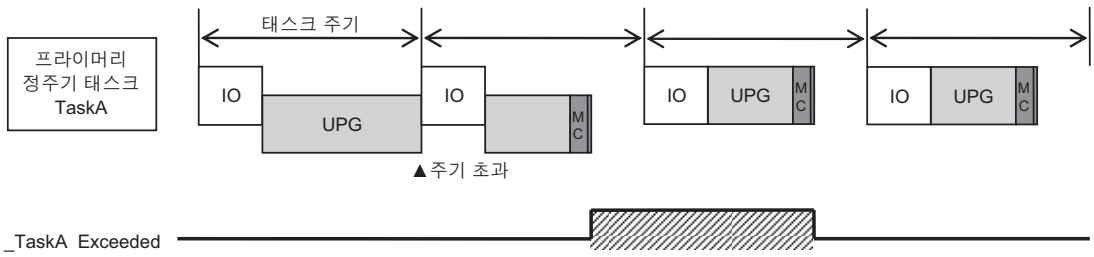
플래그 동작

- **태스크 실행 중 플래그(_<태스크 이름>_Active)**



* ActEventTask 명령으로 이벤트 태스크를 기동하는 경우에는 ActEventTask 실행시에 태스크 실행중 플래그가 ON합니다.

- **태스크 주기 초과 플래그(_<태스크 이름>_Exceeded)**



5-3 NX701의 태스크 사양과 기본 동작

여기에서는 멀티 코어 프로세서를 탑재한 NX701 CPU 유니트의 태스크 사양과 기본 동작에 대해 설명합니다.

5-3-1 NX701의 태스크 사양

태스크 사양은 다음과 같습니다.

항목	사양
태스크의 종류	• 프라이머리 정주기 태스크 • 정주기 태스크(우선도 5, 16, 17, 18) • 이벤트 태스크(우선도 8, 48)
태스크 수	• 프라이머리 정주기 태스크: 1 • 정주기 태스크: 0~4*1 • 이벤트 태스크: 0~32*2
1태스크당 프로그램의 수	최대 128
프라이머리 정주기 태스크의 태스크 주기	125μs 250μs~8ms(250μs 단위로 설정 가능)
정주기 태스크의 태스크 주기	• 우선도 5 125μs 250μs~100ms(250μs 단위로 설정 가능) • 우선도 16, 17, 18 1ms~100ms(250μs 단위로 설정 가능) 정주기 태스크의 태스크 주기는 프라이머리 정주기 태스크의 정수배를 설정합니다. 각 태스크의 태스크 주기의 최소공배수가 600ms를 초과하는 조합은 선택할 수 없습니다.

*1. 실행 우선도 5, 16, 17, 18이 각각 0 또는 1개입니다.
*2. 실행 우선도 8, 48이 각각 0~32개이고, 모두 합해서 최대 32개입니다.

사용상의 주의 사항

정주기 태스크(실행 우선도 5)를 사용하는 경우에 프라이머리 정주기 태스크의 태스크 주기를 4ms 이상으로 설정하지 마십시오. 4ms 이상으로 설정하면 슬레이브 어플리케이션 이상 또는 슬레이브 AL Status 이상 감지가 발생할 수 있습니다.

5-3-2 NX701의 태스크 분할 개념

모든 프로그램은 어느 한 태스크에 할당해야 합니다. 프로그램의 특성에 따라 다음과 같은 개념으로 할당하는 태스크를 결정하십시오.

태스크	할당하는데 적합한 프로그램
프라이머리 정주기 태스크	• 주기별 I/O 리프레시, 사용자 프로그램 실행, 모션 제어, 시스템 공통 처리의 실행 주기가 엄격하게 지켜져야 함 • 실행 우선도가 가장 높고, 고속한 응답이 필요한 제어를 포함함 • 실행 우선도가 가장 높은 모션 제어 명령을 포함함
정주기 태스크(우선도 5)	• 주기별 I/O 리프레시, 사용자 프로그램 실행, 모션 제어, 시스템 공통 처리의 실행 주기가 엄격하게 지켜져야 함 • 실행 우선도가 프라이머리 정주기 태스크 다음으로 높고. 고속한 응답이 필요한 제어를 포함함 • 실행 우선도가 비교적 높은 모션 제어 명령을 포함함 • 가장 고속으로 제어하는 처리를 프라이머리 정주기 태스크, 그 밖의 처리를 정주기 태스크(우선도 5)에 분할하여 제어하는 경우에 사용함
정주기 태스크(우선도 16)	• 실행 우선도는 비교적 낮지만 주기마다 사용자 프로그램, 시스템 공통 처리를 실행해야 함 • I/O 리프레시를 프라이머리 정주기 태스크에 할당한 일부의 슬레이브/유니트에 대한 제어를 포함 • 실행 우선도가 비교적 낮은 모션 제어 명령을 포함함 • 프라이머리 정주기 태스크로 제어하는 슬레이브/유니트에 대한 처리 중에 비교적 실행 우선도의 낮은 처리를 정주기 태스크(우선도 16)에 분할하여 제어하는 경우에 사용함
정주기 태스크(우선도 17, 18)	• 실행 우선도는 비교적 낮지만 주기마다 사용자 프로그램, 시스템 공통 처리를 실행해야 함 • 고속 응답을 필요로 하지 않는 연산 처리, 통신 처리를 포함함
이벤트 태스크	• 어떤 조건이 성립할 때만 실행함

5-3-3 NX701의 태스크 기본 동작

NX701 CPU 유니트는 멀티 코어 프로세서의 탑재에 의해 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 병렬로 실행할 수 있습니다. 각 태스크가 어떤 순서로 실행되는지는 각각의 태스크에 설정된 실행 우선도에 따라 결정됩니다.

참고

NX701 CPU 유니트에서는 태스크의 실행 및 태그 데이터 링크 서비스와 시스템 서비스를 병렬로 실행할 수 있습니다.

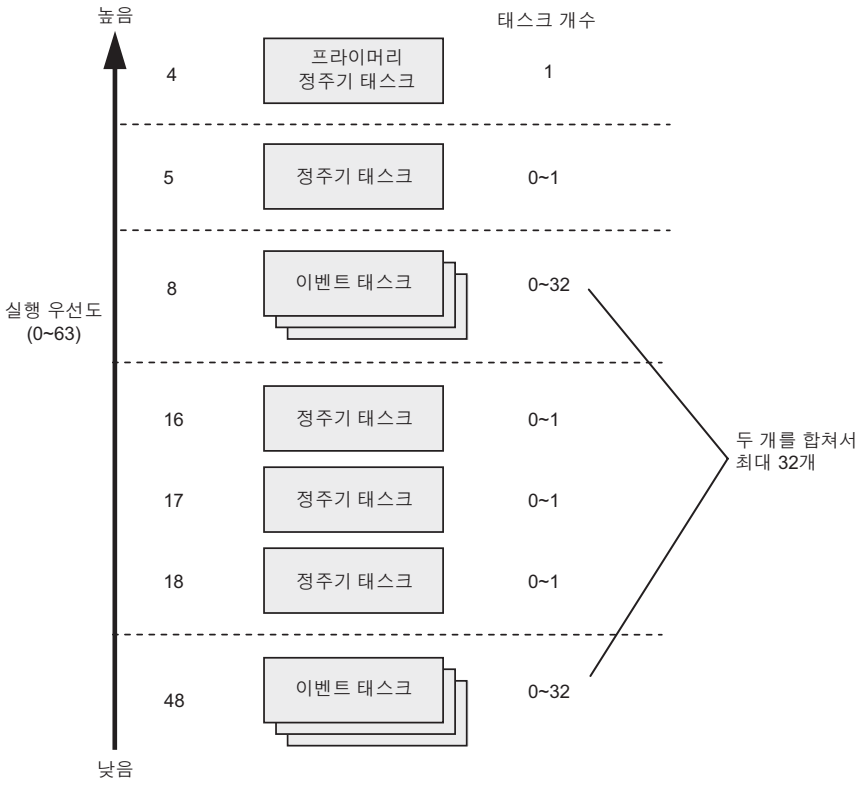
태스크 실행 우선도

태스크에는 종류에 따라 실행 우선도가 정해져 있습니다. CPU 유니트는 실행 우선도가 높은 태스크를 우선적으로 실행합니다.

어떤 태스크 Ta가 실행 중에 그보다 실행 우선도가 높은 Tb의 실행 조건이 성립한 경우에 Tb의 처리를 우선적으로 코어에 할당합니다.

태스크의 종류에 의한 실행 우선도는 다음과 같습니다. 실행 우선도 값이 작을수록 우선도가 높음을 나타냅니다.

태스크	실행 우선도	실행 우선도의 중복
프라이머리 정주기 태스크	4	–
정주기 태스크	5, 16, 17, 18	복수의 태스크에 동일한 실행 우선도를 설정할 수 없습니다.
이벤트 태스크	8, 48	복수의 태스크에 동일한 실행 우선도를 설정할 수 있습니다. 그 경우의 실행 순서에 대해서는 '5-3-5 NX701의 이벤트 태스크 실행 타이밍(5-24페이지)'을 참조하십시오.



프라이머리 정주기 태스크와 정주기 태스크의 태스크 주기

프라이머리 정주기 태스크와 정주기 태스크는 주기적으로 반복해서 실행되는 태스크입니다.
정주기 태스크의 태스크 주기는 반드시 프라이머리 정주기 태스크의 태스크 주기(' 프라이머리 주기'라고 함)의 정수배가 되게 설정합니다. 따라서 몇 번에 1번씩 둘의 실행 타이밍이 일치합니다.
예를 들어 프라이머리 주기가 1ms, 정주기 태스크(실행 우선도 16)의 태스크 주기가 4ms인 경우에는 프라이머리 정주기 태스크가 4회 실행될 때마다 프라이머리 정주기 태스크와 정주기 태스크 실행 타이밍이 일치합니다.

참고

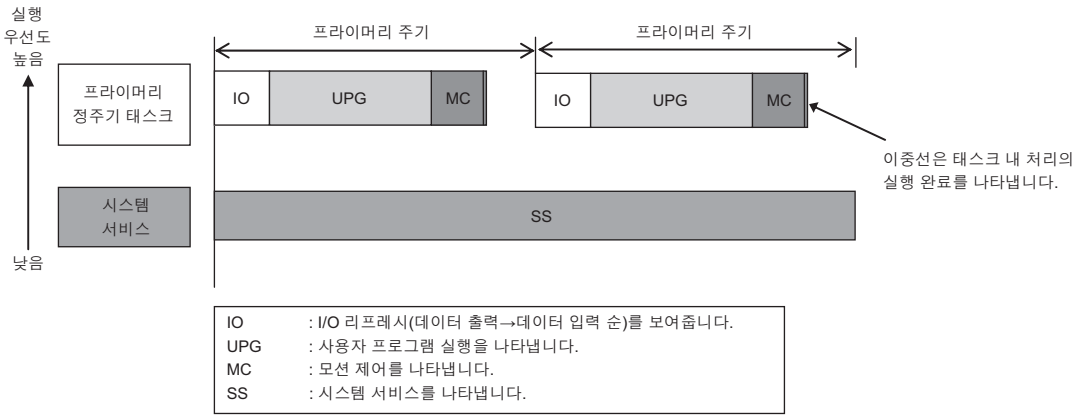
이벤트 태스크는 주기적으로 실행되는 것이 아니라 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 따라서 이벤트 태스크 실행 타이밍은 이벤트 태스크의 실행 조건이 성립한 타이밍과 이벤트 태스크의 실행 우선도에 따라 결정됩니다.

태스크의 실행 순서의 예

프라이머리 정주기 태스크와 정주기 태스크에 대해 실행 순서의 예시를 몇 가지 설명합니다.
이벤트 태스크의 실행 순서에 대해서는 '5-3-5 NX701의 이벤트 태스크 실행 타이밍(5-24페이지)'을 참조하십시오.

- **프라이머리 정주기 태스크뿐인 경우**

  프라이머리 주기마다 프라이머리 정주기 태스크가 실행됩니다.
  그림의 시스템 서비스는 통신 처리 등 CPU 유니트가 태스크 이외에 실행하는 처리입니다. 시스템 서비스의 자세한 내용은 '시스템 서비스의 처리 내용(5-75페이지)'을 참조하십시오.

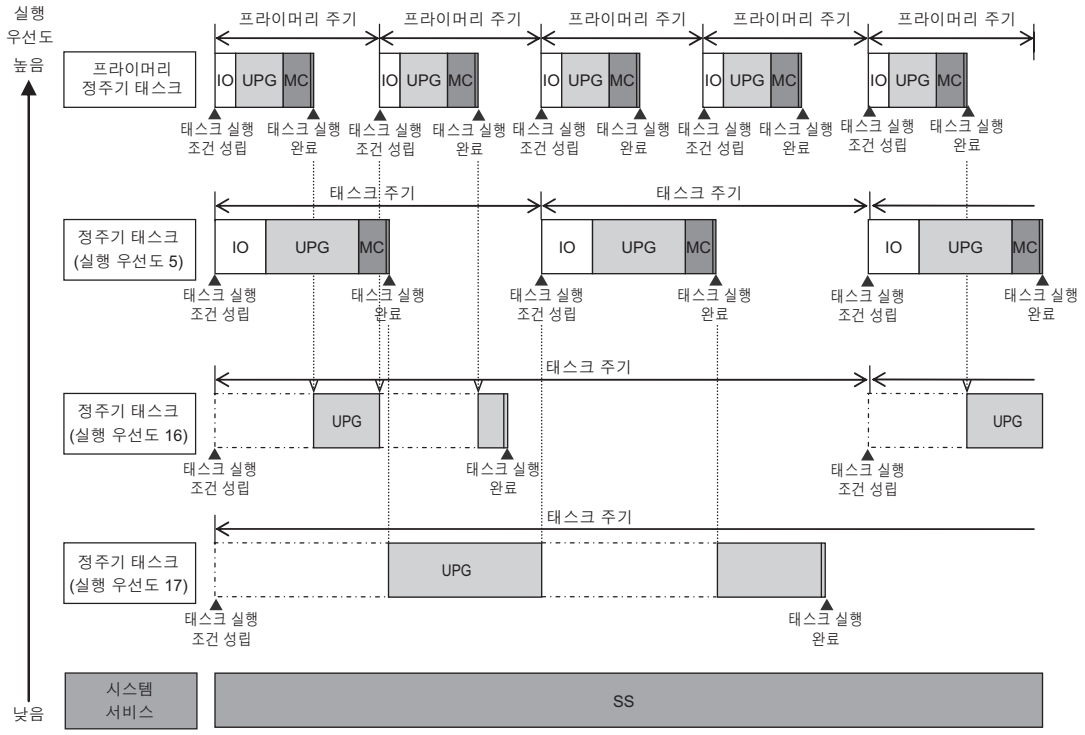


IO	: I/O 리프레시(데이터 출력→데이터 입력 순)를 보여줍니다.
UPG	: 사용자 프로그램 실행을 나타냅니다.
MC	: 모션 제어를 나타냅니다.
SS	: 시스템 서비스를 나타냅니다.

- **프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 5), 정주기 태스크(실행 우선도 16), 정주기 태스크(실행 우선도 17)가 혼재하는 경우**
  - 각 태스크는 다음 2가지 그룹으로 분류되어 실행됩니다.

그룹	태스크의 종류
그룹 1	프라이머리 정주기 태스크 정주기 태스크(실행 우선도 16)
그룹 2	정주기 태스크(실행 우선도 5, 17, 18) 이벤트 태스크(실행 우선도 8, 48)

  - 동일한 그룹 내의 각 태스크는 실행 우선도가 높은 순서대로 우선적으로 실행됩니다.
  - 프라이머리 정주기 태스크는 실행 우선도가 가장 높으므로 확실하게 프라이머리 주기로 실행됩니다.
  - 정주기 태스크(실행 우선도 5)는 프라이머리 정주기 태스크와 병렬로 실행됩니다.
  - 정주기 태스크(실행 우선도 16)는 프라이머리 정주기 태스크가 완료된 후 실행됩니다.
  - 정주기 태스크(실행 우선도 17)는 정주기 태스크(실행 우선도 5)보다 실행 우선도가 낮으므로 정주기 태스크(실행 우선도 5)가 실행되지 않는 시간에 실행됩니다.
  - 이 예에서는 정주기 태스크(실행 우선도 5)의 태스크 주기는 프라이머리 주기의 2배로 가정합니다. 또한, 정주기 태스크(실행 우선도 16)의 태스크 주기는 프라이머리 주기의 4배로 가정합니다. 따라서 정주기 태스크(실행 우선도 5)는 프라이머리 주기의 2회에 1번, 정주기 태스크(실행 우선도 16)는 프라이머리 주기의 4회에 1번, 실행 시작 타이밍이 동기합니다.
  - 시스템 서비스는 태스크의 영향을 받지 않고 언제라도 실행됩니다.



사용상의 주의 사항

복수의 태스크 사이에서 동일한 변수의 읽기/쓰기를 할 가능성이 있는 경우에는 반드시 '변수의 태스크 간 배타 제어 기능'을 사용하십시오. 그렇게 하지 않으면 어떤 태스크의 실행 중에 다른 태스크에 의해 변수 값이 변경되는 경우가 있습니다.

자세한 내용은 📖'5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

태스크와 동작 모드

CPU 유니트의 동작 모드와 태스크에는 다음과 같은 관계가 있습니다.

태스크	사양
프라이머리 정주기 태스크	• 운전 모드와 프로그램 모드에서 모두 실행됩니다.
정주기 태스크	• 단, 운전 모드에서만 사용자 프로그램 실행이 가능합니다.
이벤트 태스크	운전 모드에서만 실행됩니다.

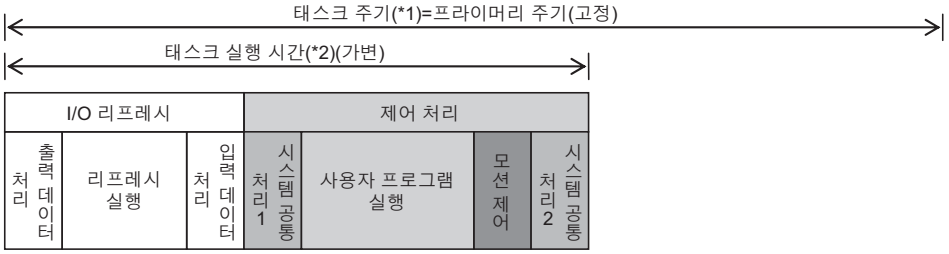
사용상의 주의 사항

- 동작 모드를 프로그램 모드에서 운전 모드로 전환한 시점에서 이벤트 태스크의 실행 조건이 이미 성립한 경우에 이벤트 태스크는 실행되지 않습니다. 이벤트 태스크가 실행되는 것은 동작 모드가 운전 모드인 상태에서 이벤트 태스크의 실행 조건이 성립하지 않다가 성립으로 변화한 때입니다.
- 동작 모드가 운전 모드에서도 전체 정지 이상 레벨의 이상 발생 중에는 이벤트 태스크가 실행되지 않습니다.

각 태스크 종류의 처리 내용

● **프라이머리 정주기 태스크**

태스크의 실행 우선도가 가장 높고, 고속·고정밀도로 실행할 수 있는 태스크입니다.
지정한 주기로 시스템 공통 처리, I/O 리프레시, 사용자 프로그램, 모션 제어가 실행됩니다.



*1 : 태스크 주기: 태스크가 정주기로 실행되는 설정된 고정 시간입니다.

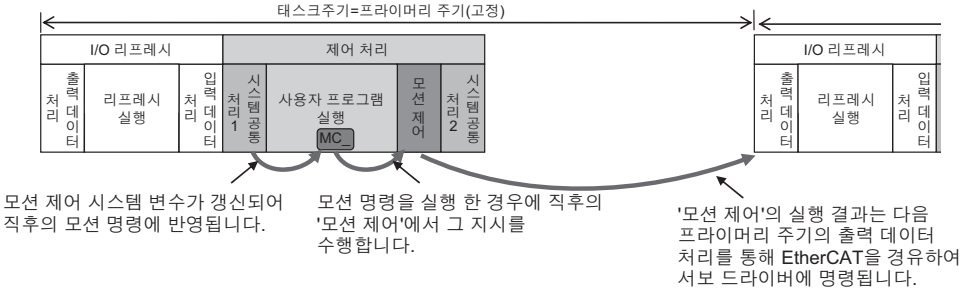
*2 : 태스크 실행 시간: 태스크 실행 조건 성립에서 실행 완료까지 실제로 걸린 시간입니다.

처리		처리 내용
I/O 리프레시	출력 데이터 처리	• I/O 리프레시를 실행하는 I/O(출력)에 대한 OUT 리프레시 데이터를 생성합니다. • 강제값 리프레시가 설정된 경우에는 OUT 리프레시 데이터에 강제값(출력)을 반영합니다.
	리프레시 실행	• I/O와의 데이터 교환을 실행합니다.
	입력 데이터 처리	• 이벤트 태스크의 조건식 일치 판정 • I/O 리프레시를 실행한 I/O(입력) 유니트의 IN 리프레시 데이터를 전송합니다. • 강제값 리프레시가 설정된 경우에는 전송한 IN 리프레시 데이터에 강제값(입력)을 반영합니다.
시스템 공통 처리 1		• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 모션 입력 처리*1 • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행		• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
모션 제어*2		• 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16)에 할당된 사용자 프로그램 상의 모션 제어 명령에서의 모션 제어의 지시를 실행합니다. • 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에 대한 모션 출력 처리
시스템 공통 처리 2		• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. 모션 제어 시스템 변수의 축 현재값(위치, 속도, 토크) 및 모션 제어 시스템 변수의 서보 드라이버 상태가 갱신됩니다.

*2. 프라이머리 정주기 태스크의 사용자 프로그램 상의 모션 제어 명령인 경우에 다음 그림과 같이 직후의 '모션 제어'에서 그 명령에서의 지시가 실행되고, 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에서 서보 드라이버에 출력됩니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.



또한, 정주기 태스크(실행 우선도 16)의 사용자 프로그램 상의 모션 제어 명령인 경우에는 그 정주기(실행 우선도 16)의 다음 프라이머리 정주기 태스크의 '모션 제어(MC)' 타이밍에 해당 명령에서의 지시가 실행됩니다.

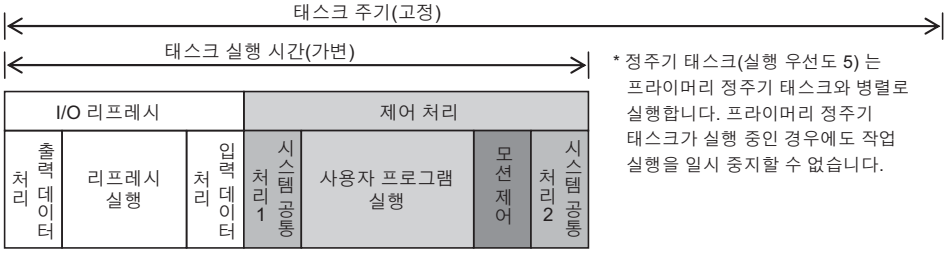
자세한 내용은 '5-11-3 시스템으로서의 입출력 응답 시간(5-112페이지)'을 참조하십시오.

- **정주기 태스크(실행 우선도 5)**

  태스크의 실행 우선도가 프라이머리 정주기 태스크 다음으로 높고. 고속·고정밀도로 실행할 수 있는 태스크입니다.

  지정한 주기로 시스템 공통 처리, I/O 리프레시, 사용자 프로그램, 모션 제어가 실행됩니다.

  정주기 태스크(실행 우선도 5)는 NX701 CPU 유니트에서만 사용할 수 있습니다.

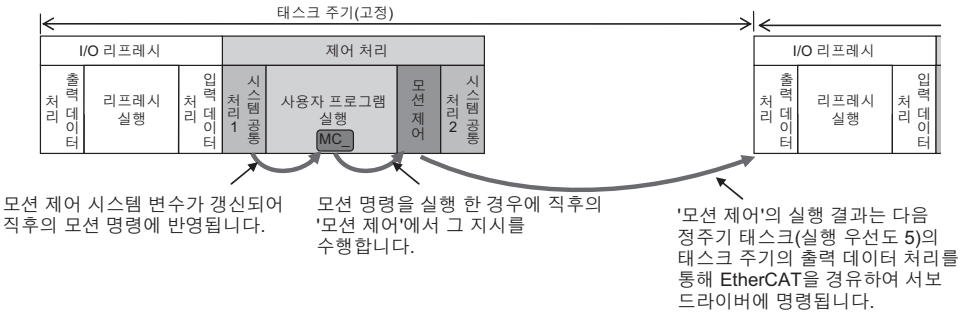


처리		처리 내용
I/O 리프레시	출력 데이터 처리	• I/O 리프레시를 실행하는 I/O(출력)에 대한 OUT 리프레시 데이터를 생성합니다. • 강제값 리프레시가 설정된 경우에는 OUT 리프레시 데이터에 강제값(출력)을 반영합니다.
	리프레시 실행	• I/O와의 데이터 교환을 실행합니다.
	입력 데이터 처리	• I/O 리프레시를 실행한 I/O(입력) 유니트의 IN 리프레시 데이터를 전송합니다. • 강제값 리프레시가 설정된 경우에는 전송한 IN 리프레시 데이터에 강제값(입력)을 반영합니다.
시스템 공통 처리 1		• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 모션 입력 처리*1 • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행		• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
모션 제어*2		• 정주기 태스크(실행 우선도 5)에 할당된 사용자 프로그램 상의 모션 제어 명령에서의 모션 제어의 지시를 실행합니다. • 다음 정주기 태스크(실행 우선도 5)의 'I/O 리프레시'에 대한 모션 출력 처리
시스템 공통 처리 2		• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. 모션 제어 시스템 변수의 축 현재값(위치, 속도, 토크) 및 모션 제어 시스템 변수의 서보 드라이버 상태가 갱신됩니다.

*2. 정주기 태스크(실행 우선도 5)의 사용자 프로그램 상의 모션 제어 명령인 경우에 다음 그림과 같이 직후의 '모션 제어'에서 그 명령에서의 지시가 실행되고, 다음 정주기 태스크(실행 우선도 5)의 'I/O 리프레시'에서 서보 드라이버로 출력됩니다.

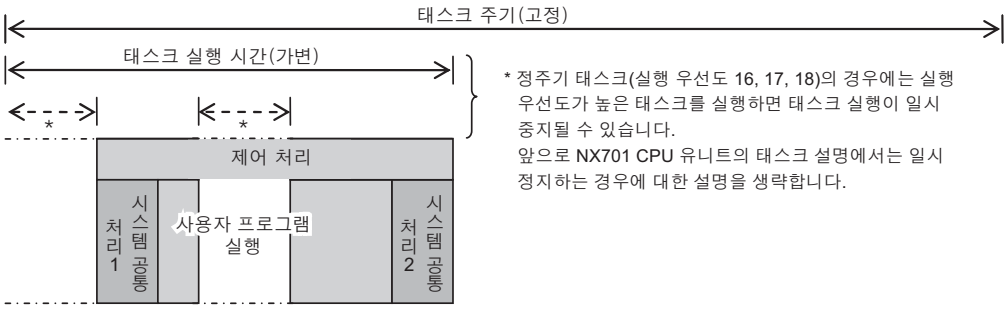
(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.



- **정주기 태스크(실행 우선도 16, 17, 18)**

  프라이머리 주기의 정수배로 지정된 태스크 주기로 사용자 프로그램을 실행하는 태스크입니다. 태스크 전체에서 0~3개 할당할 수 있습니다.
  정주기 태스크(실행 우선도 16)는 [I/O를 제어하는 태스크 설정]을 통해 정주기 태스크(실행 우선도 16)를 설정한 슬레이브/유니트에 대한 제어 프로그램을 기술할 수 있습니다.
  정주기 태스크로 I/O를 제어하지 않는 경우와 제어하는 경우에는 처리 내용이 다음과 같이 다릅니다.

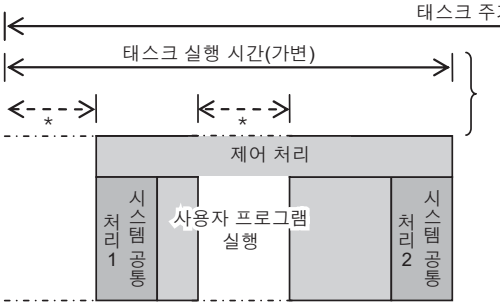
  • 정주기 태스크로 I/O를 제어하지 않는 경우



처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.

• 정주기 태스크(실행 우선도 16)로 I/O를 제어하는 경우



* 정주기 태스크(실행 우선도 16)의 경우에는 실행 우선도가 높은 태스크를 실행하면 태스크가 일시 중지될 수 있습니다.
앞으로 NX 시리즈 CPU 유니트의 태스크에 대한 설명에서는 일시 정지하는 경우의 설명을 생략합니다.

처리	처리 내용
시스템 공통 처리 1	• IN 리프레시 데이터의 반영*1 • 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다. • OUT 리프레시 데이터의 반영*2

*1. [I/O를 제어하는 태스크 설정]을 통해 정주기 태스크(실행 우선도 16)를 설정한 EtherCAT 슬레이브의 IN 리프레시 데이터를 전송합니다. 프라이머리 정주기 태스크의 I/O 리프레시 시 프로세스 데이터 통신으로 입력된 데이터가 IN 리프레시 데이터가 됩니다.

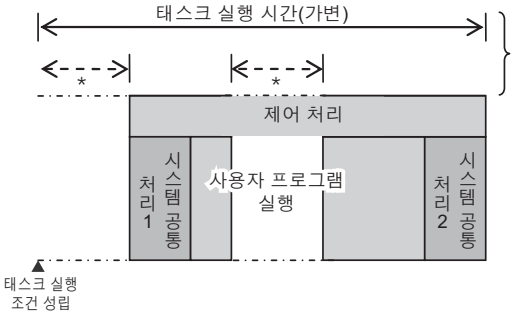
*2. [I/O를 제어하는 태스크 설정]을 통해 정주기 태스크(실행 우선도 16)를 설정한 EtherCAT 슬레이브의 사용자 프로그램의 실행 결과를 OUT 리프레시 데이터에 반영합니다. OUT 리프레시 데이터는 프라이머리 정주기 태스크의 I/O 리프레시 시 프로세스 데이터 통신으로 출력됩니다

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.

● **이벤트 태스크**

지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 태스크 전체에서 0~32개 할당할 수 있습니다.

이벤트 태스크의 처리 내용은 다음과 같습니다.



* 실행 우선도가 높은 태스크 실행으로 인해 태스크 실행을 일시 정지합니다.
앞으로 NX701 CPU 유니트의 태스크 설명에서는 일시 정지하는 경우에 대한 설명을 생략합니다.

처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우)*1
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우)*1

*1. 변수의 태스크 간 배타 제어 처리의 자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

5-3-4 NX701의 이벤트 태스크의 실행 조건

이벤트 태스크는 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 이벤트 태스크의 실행 조건에는 다음 2종류가 있습니다.

실행 조건	이벤트 태스크가 실행되는 타이밍	사용하는 경우
ActEventTask 명령에 의한 실행	ActEventTask 명령 실행 시	• 사용자 프로그램 내에서 어느 이벤트 태스크를 실행하는지 명시하고자 하는 경우 • 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정 시까지 변화할 가능성이 있는 경우
변수의 조건식 일치에 의한 실행	지정한 변수의 값이 지정한 조건식에 일치했을 때*1	이벤트 태스크 실행을 위한 사용자 프로그램을 생략해서 사용자 프로그램을 간략화하고자 하는 경우

*1. 지정한 변수의 값이 지정한 조건식에 일치했는지 여부를 판정하는 타이밍에 대해서는 '실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍(5-25페이지)'을 참조하십시오.

안전상의 중요 사항

이벤트 태스크의 실행 조건이 변수의 조건식과 일치하고 조건식에 다음 변수를 지정하는 경우에 조건이 일치할 때 이벤트 태스크가 기동하지 않거나 조건 불일치 시에 이벤트 태스크가 기동되는 경우가 있습니다.

- 모션 제어 시스템 변수를 제외하고, 데이터 크기가 16비트 이상인 구조체 멤버
- 데이터 크기가 16비트 이상의 배열 요소

위의 변수를 지정하여 일치 판정을 실시하는 경우에는 다음 중 하나를 실행하십시오.

- 64비트 이외의 기본 데이터 형식의 내부 변수에 복사하고, 복사한 내부 변수를 참조함
- 변수의 태스크 간 배타 제어 설정 기능을 사용하고, 프라이머리 정주기 태스크를 갱신 태스크로 함

ActEventTask 명령에 의한 이벤트 태스크 실행

사용자 프로그램 내에서 ActEventTask 명령이 실행되면 지정한 이벤트 태스크가 1회 실행됩니다.

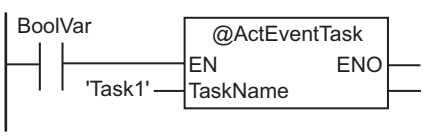
ActEventTask 명령의 자세한 사양은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

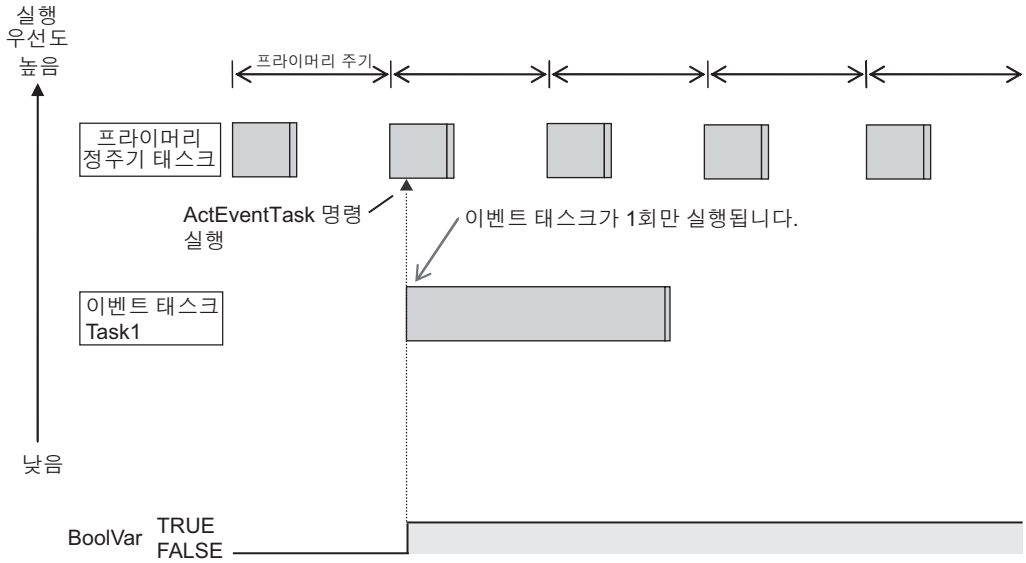
ActEventTask 명령에 의한 방법을 사용하면 사용자 프로그램을 봤을 때, 어느 이벤트 태스크를 기동하는지 잘 알 수 있습니다. 또한, 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정까지 변화할 가능성이 있는 경우에도 유효합니다.

- **ActEventTask 명령을 사용한 사용자 프로그램 예**

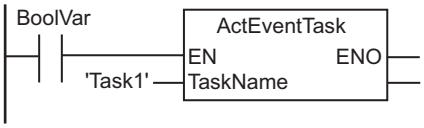
  예 1) 변수의 값이 변화했을 때 이벤트 태스크를 1회만 실행하는 경우

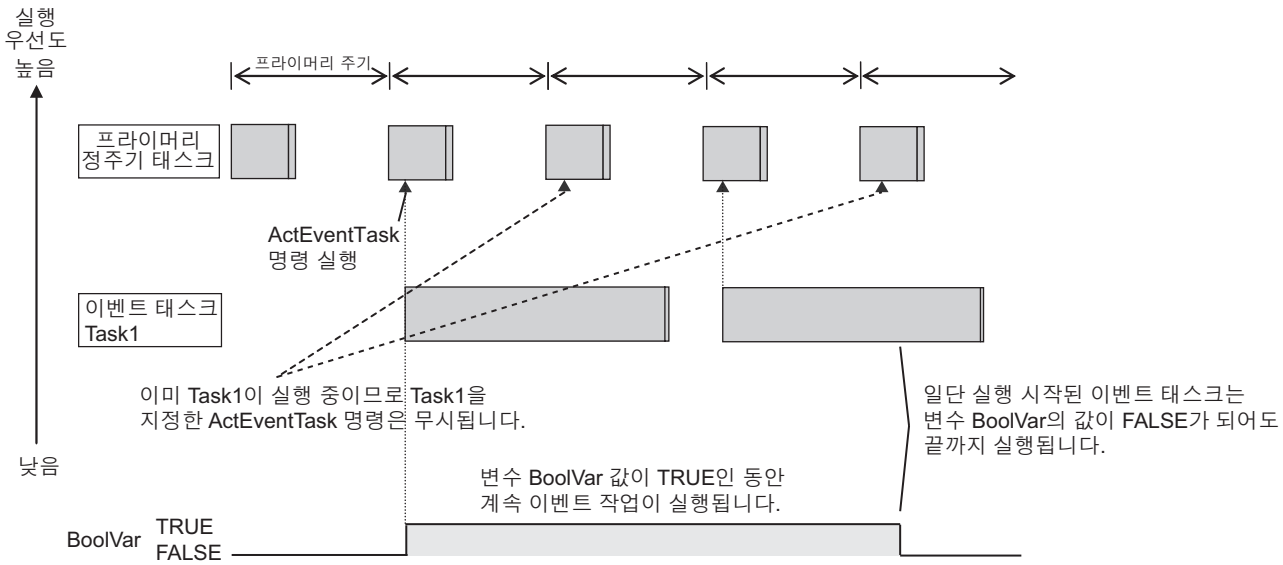
  다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하면 BOOL 형식 변수 BoolVar의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크 Task1이 1회만 실행됩니다.





예 2) 변수가 지정한 값인 동안 반복해서 이벤트 태스크를 실행하는 경우
다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하지 않으면 BOOL 형식 변수 BoolVar의 값이 TRUE인 동안 반복해서 이벤트 태스크 Task1이 실행됩니다. 단, Task1이 실행 중에 Task1을 지정한 ActEventTask 명령은 실행해도 무시됩니다. 또한, 일단 실행 시작된 이벤트 태스크는 실행 도중에 BoolVar의 값이 FALSE가 되어도 마지막까지 실행됩니다.





변수의 조건식 일치에 의한 이벤트 태스크 실행

변수의 조건식 일치에 의한 방법은 Sysmac Studio에서 지정한 변수 값이 지정한 조건식에 일치한 경우 이벤트 태스크가 1회 실행된다는 것입니다. 해당 변수의 값이 조건식에 일치하는 동안 이벤트 태스크가 계속 실행는 것이 아니라 변수 값이 조건식에 불일치에서 일치로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.

변수의 조건 일치에 의한 방법에서는 이벤트 태스크 실행을 위한 사용자 프로그램이 필요 없어집니다.

● 조건식을 지정할 수 있는 변수의 종류

조건식을 지정할 수 있는 변수의 종류는 다음과 같습니다.

변수의 종류		지정 가능 여부
시스템 정의 변수		가능*1
준 사용자 정의 변수		가능
사용자 정의 변수	글로벌 변수	가능
	프로그램 내에서 사용하는 변수	가능
	펑션 블록 내에서 사용하는 변수	가능*2
	펑션 내에서 사용하는 변수	불가능

*1. 다음과 같은 변수는 제외합니다.
EN, ENO, P_Off, P_CY, P_First_RunMode, P_First_Run, P_PRGER

*2. 입출력 변수는 제외합니다.

● 조건식을 지정할 수 있는 변수의 데이터 형식

조건식을 지정할 수 있는 변수의 데이터 형식은 다음과 같습니다.

데이터 형식의 분류	데이터 형식		지정 가능 여부
기본 데이터 형식	부울 형식, 비트 열 형식, 정수 형식, 실수 형식		가능
	지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		불가능
데이터 형식 지정	배열 지정	배열 전체	불가능
		요소	가능*1
파생 데이터 형식	구조체 형식	구조체 전체	불가능
		멤버	가능*2
	공용체 형식	공용체 전체	불가능
		멤버	가능*2
	열거 형식		가능

*1. 배열 요소의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

*2. 멤버의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

● **지정할 수 있는 조건식**

지정할 수 있는 조건식은 조건식을 지정하는 변수의 데이터 형식에 따라 다음과 같이 정해져 있습니다.

조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다.

데이터 형식	지정할 수 있는 조건식
부울 형식, 부울 형식의 배열 요소, 부울 형식의 구조체 멤버, 부울 형식의 공용체 멤버	상승 FALSE→TRUE
	하강 TRUE→FALSE
비트 열 형식, 정수 형식, 실수 형식 또는 이들을 데이터 형식으로 하는 배열 요소, 구조체 멤버, 공용체 멤버	변수 = {비교 상수}
	변수 ≠ {비교 상수}
	변수 > {비교 상수}
	변수≧{비교 상수}
	변수 < {비교 상수}
	변수≦{비교 상수}

● **비교 상수 값의 유효 범위**

조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다. 비교 상수의 값의 유효 범위는 조건식을 지정하는 변수의 데이터 형식 값의 유효 범위와 동일합니다.

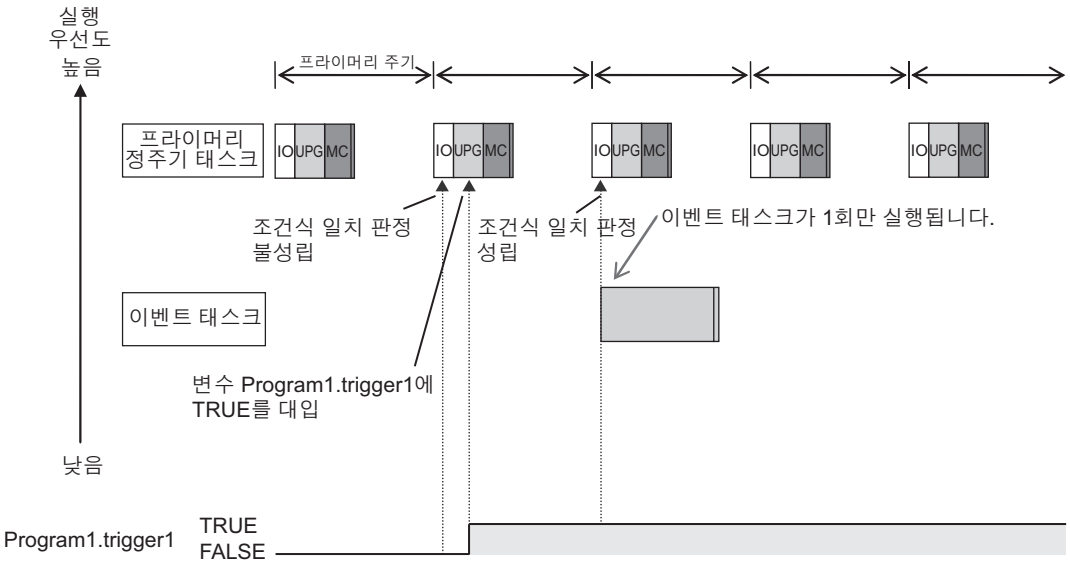
각 데이터 형식의 값의 유효 범위는 📖'기본 데이터 형식과 파생 데이터 형식(6-31페이지)'을 참조하십시오.

예를 들어 조건식을 지정하는 변수의 데이터 형식이 BYTE인 경우에 비교 상수 값의 유효 범위는 BYTE#16#00~BYTE#16#FF입니다.

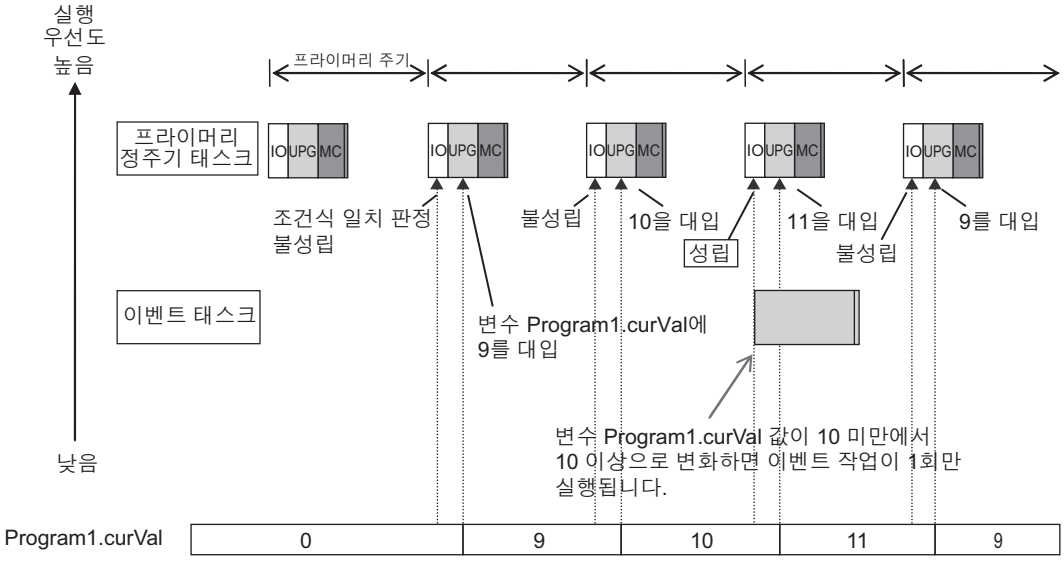
● **변수의 조건식 일치에 의한 이벤트 태스크 실행 예**

예 1) 이벤트 태스크 실행 조건을 BOOL 형식 변수 Program1.trigger1의 상승으로 한 경우

Program1.trigger1의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



예 2) 이벤트 태스크 실행 조건을 INT 형식 변수 Program1.curVal?10로 한 경우
Program1.curVal의 값이 10미만에서 10 이상으로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



사용상의 주의 사항

프라이머리 정주기 태스크 내에서 지정한 변수 값이 변화한 경우에 다음 프라이머리 정주기 태스크로 조건식과의 일치가 판정됩니다. 따라서 이벤트 태스크가 실행되는 것은 변수 값이 조건식을 충족한 주기의 다음 프라이머리 정주기 태스크로 조건식 일치 판정이 실행된 이후입니다.

5-3-5 NX701의 이벤트 태스크 실행 타이밍

이벤트 태스크의 실행 우선도는 8 또는 48입니다. 멀티 코어 프로세서를 탑재한 NX701 CPU 유니트에서는 실행 조건이 성립한 이벤트 태스크는 태스크의 실행 우선도에 따라 실행이 시작됩니다. 사용자 프로그램에 따라서는 프라이머리 정주기 태스크 및 정주기 태스크 또는 실행 우선도가 다른 이벤트 태스크와 병렬로 실행되는 경우도 있습니다.

복수의 태스크 사이에서 동일한 변수를 읽거나 쓸 가능성이 있는 경우에는에 반드시 다음 기능을 사용하여 프라이머리 정주기 태스크, 정주기 태스크 또는 실행 우선도가 다른 이벤트 태스크와의 실행을 제어하십시오.

목적	사용하는 기능
이벤트 태스크와 다른 태스크로 동일한 글로벌 변수에 액세스하는 경우	변수의 태스크 간 배타 제어 기능 • 변수의 태스크 간 배타 제어 설정 기능 • Lock(태스크 간 배타 잠금) 명령 • Unlock(태스크 간 배타 잠금 해제) 명령
다른 태스크의 실행 상태를 확인하는 경우	Task_IsActive(태스크 실행 중 판정) 명령

자세한 내용은 📖'5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

이벤트 태스크 실행 조건에 따라 이벤트 태스크 실행 타이밍이 다릅니다.
또한, 이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. 동일한 실행 우선도의 여러 이벤트 태스크 실행 조건이 성립한 경우에는 주의해야 합니다.

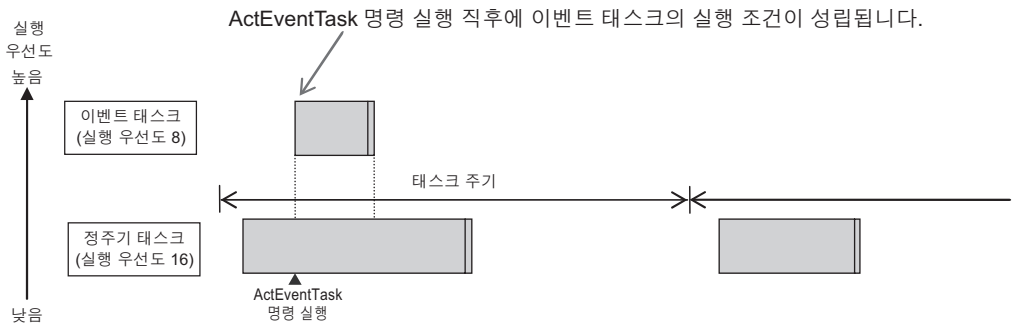
이벤트 태스크 실행 조건에 의한 실행 타이밍의 차이

이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우와 변수의 조건식 일치에 의한 경우에는 실행 조건의 성립 타이밍이 다릅니다.

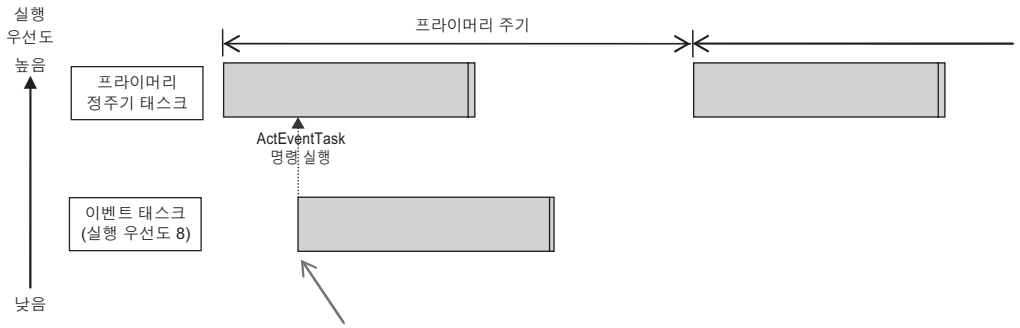
- **실행 조건이 ActEventTask 명령에 의한 경우의 실행 타이밍**

  이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우에는 ActEventTask 명령의 실행직후에 이벤트 태스크 실행 조건이 성립합니다. 실행 조건이 성립한 이벤트 태스크는 태스크의 실행 우선도에 따라 실행이 시작됩니다.

  예 1) 명령을 실행하는 태스크보다 실행 우선도가 높은 이벤트 태스크를 기동하는 경우



예 2) 명령을 실행하는 태스크보다 실행 우선도가 낮은 이벤트 태스크를 기동하는 경우

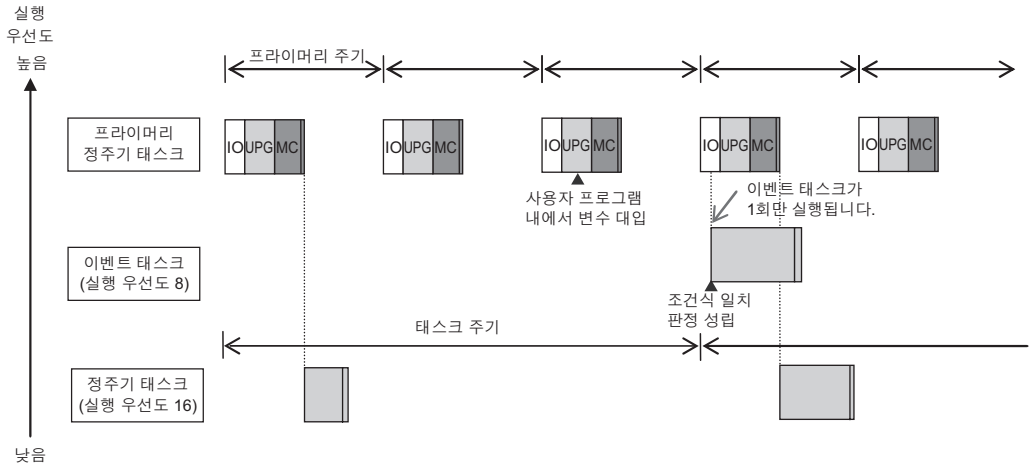


ActEventTask 명령 실행 직후에 이벤트 태스크의 실행 조건이 성립됩니다.

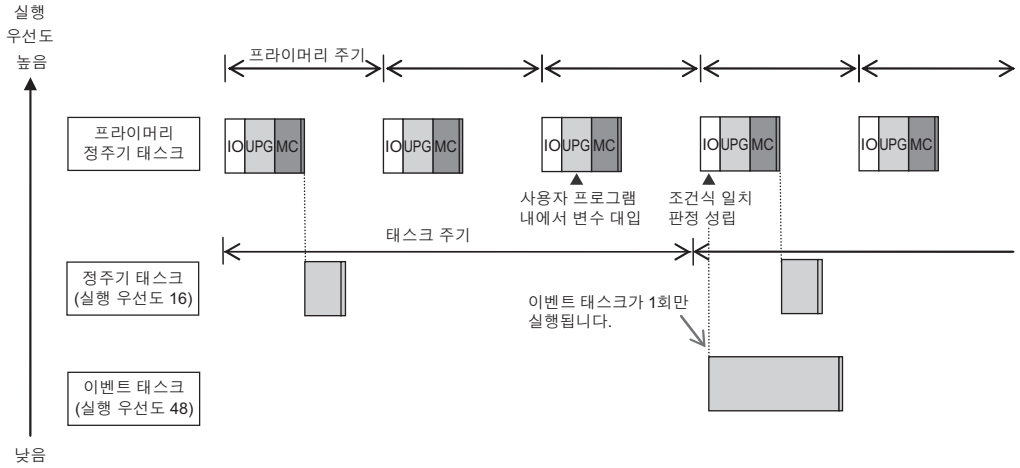
- **실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍**

조건식 일치 판정은 프라이머리 정주기 태스크 중에 실행됩니다. 조건식 판정 처리에서 조건이 일치했을 때 이벤트 태스크 실행 조건이 성립합니다. 실행 조건이 성립한 이벤트 태스크는 태스크의 실행 우선도에 따라 실행이 시작됩니다.

예 1) 정주기 태스크(실행 우선도 16)와 이벤트 태스크(실행 우선도 8)가 존재하는 경우



예 2) 정주기 태스크(실행 우선도 16)와 이벤트 태스크(실행 우선도 48)가 존재하는 경우



사용상의 주의 사항

- NX701 CPU 유니트에서는 조건식 일치가 프라이머리 정주기 태스크의 I/O 리프레시에 의해 발생하는 경우 및 프라이머리 정주기 태스크에 할당된 사용자 프로그램 실행에서 발생하는 경우에도 이벤트 태스크 실행 조건이 성립하는 타이밍은 다음과 같이 동일합니다.

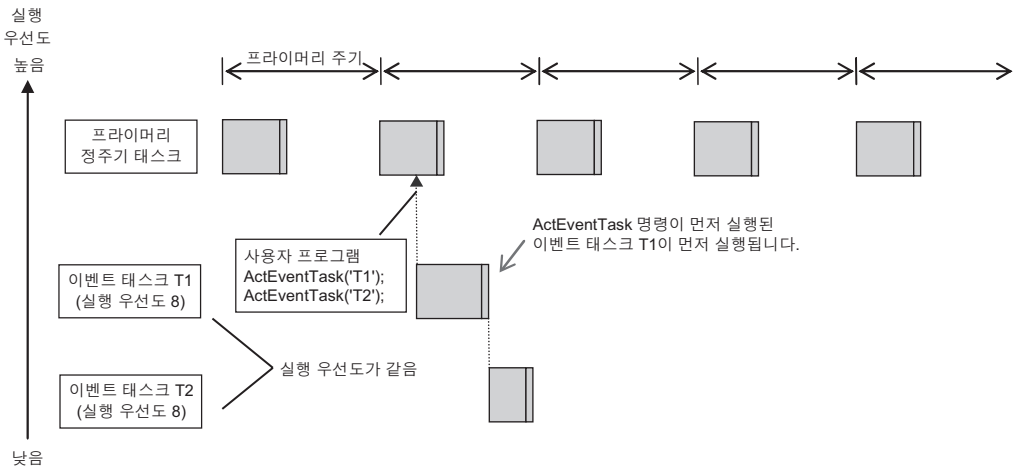
조건식 일치 발생 요인	이벤트 태스크 실행 조건이 성립하는 타이밍
프라이머리 정주기 태스크의 I/O 리프레시	다음 프라이머리 정주기 태스크 판정 처리
프라이머리 정주기 태스크의 사용자 프로그램 실행	다음 프라이머리 정주기 태스크 판정 처리

- 이벤트 태스크가 실행되는 것은 1회 이전 판정 시 조건식 불일치이고, 다음 판정 시 조건식 일치가 발생한 경우입니다. 따라서 불일치에서 일치로 변화가 있었다고 해도 다음 판정 시까지 불일치로 되돌아간 경우에는 이벤트 태스크는 실행되지 않습니다.
- NX701 CPU 유니트에서 데이터 크기가 64비트 이상의 내부 변수를 조건식으로 지정하는 경우에 프라이머리 정주기 태스크에 할당된 프로그램에서 정의된 내부 변수를 지정하십시오. 프라이머리 정주기 태스크 이외에 정의된 내부 변수를 지정하면 변수 값의 동시성이 확보되지 않는 경우가 있기 때문에 일치 판정이 올바르게 실행되지 않을 수 있습니다.

동일한 실행 우선도의 이벤트 태스크 실행 타이밍

이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. ActEventTask 명령에 의해 동일한 실행 우선도의 복수의 이벤트 태스크 실행 조건이 성립한 경우에는 명령을 실행한 순서대로 이벤트 태스크가 실행됩니다.

예 1) 2가지 ActEventTask 명령이 실행된 경우
다음 그림과 같이 먼저 실행된 ActEventTask 명령에 의해 지정된 이벤트 태스크 T1이 이벤트 태스크 T2보다도 먼저 실행됩니다.



예 2) 변수의 조건식 일치와 ActEventTask 명령이 혼재하는 경우

이벤트 태스크 T1, T2, T3 실행 조건이 다음과 같이 설정되었다고 가정합니다.

- T1: 변수 VarA의 조건식 일치
- T2: 변수 VarB의 조건식 일치
- T3: ActEventTask 명령

다음 그림과 같이 동일한 프라이머리 주기 중에 I/O 리프레시에 의한 VarA의 조건식 일치, 사용자 프로그램에서의 ActEventTask 명령 실행, 사용자 프로그램에서의 VarB의 조건식 일치가 발생한 경우에는 다음 순서로 동작합니다.

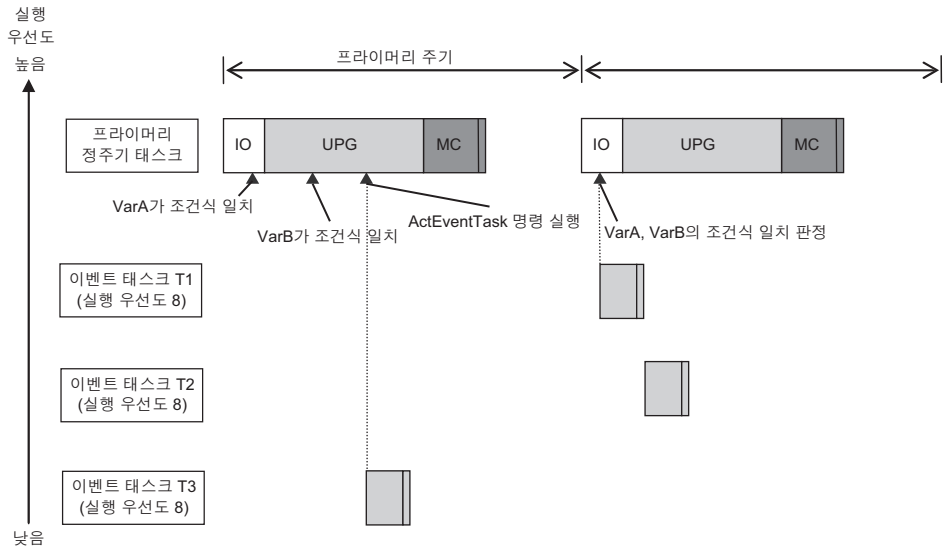
(1) I/O 리프레시에서 VarA가 조건식에 일치합니다.

(2) 사용자 프로그램에서 VarB가 조건식에 일치합니다.

(3) 이 시점에서는 아직 조건식의 일치 판정이 실행되지 않았으므로 T1과 T2는 실행되지 않습니다.

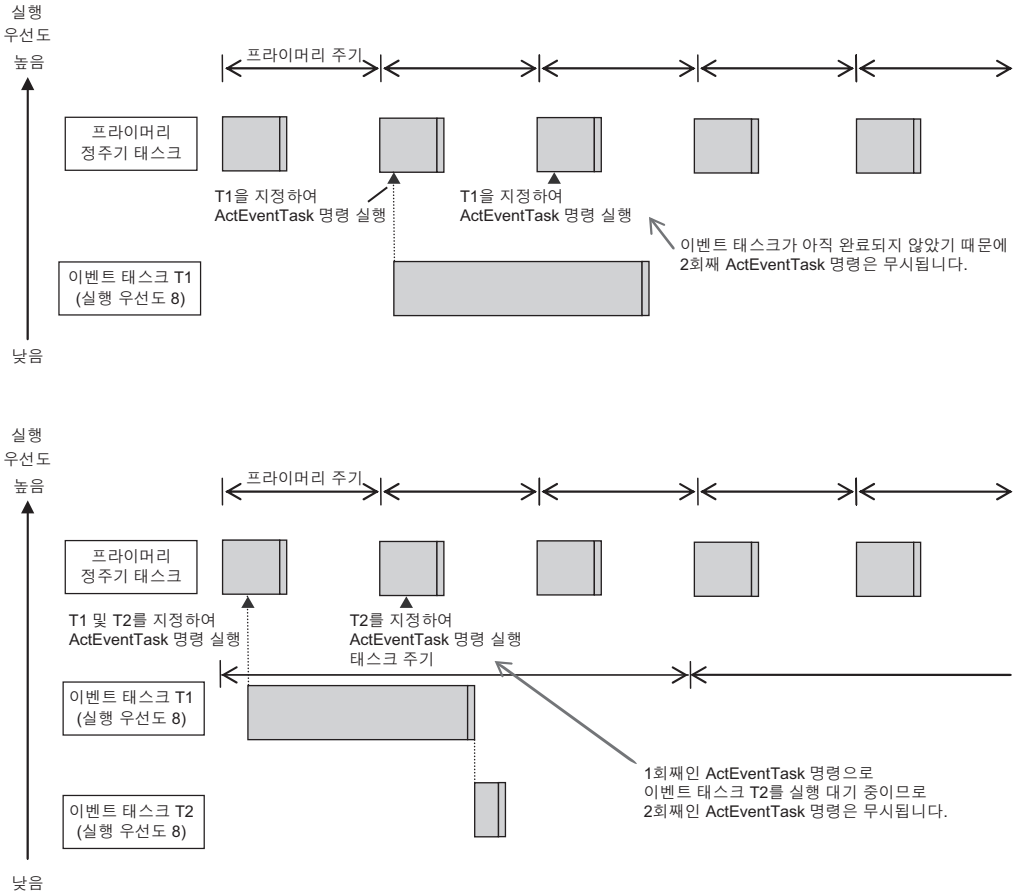
(4) 사용자 프로그램으로 ActEventTask 명령이 실행되고, T30이 실행됩니다.

(5) 다음 프라이머리 정주기 태스크의 리프레시 실행으로 VarA와 VarB가 조건식에 일치한다고 판정되고, T1과 T2가 실행됩니다.
동일한 주기에 복수의 조건식이 일치한다고 판정된 경우에 이벤트 태스크의 실행 순서가 불안정해집니다. 다음 그림은 T10이 먼저 실행된 경우입니다.



5-3-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작

이벤트 태스크 완료 전에 다시 해당 이벤트 태스크 실행 조건이 성립한 경우에는 2번째 실행 조건 성립은 무시됩니다. 이벤트 태스크 완료 전은 이벤트 태스크 실행 중 및 이벤트 태스크 실행 대기 중을 모두 포함합니다. 또한, 일단 실행이 시작된 이벤트 태스크는 실행 조건이 성립하지 않아도 마지막까지 실행됩니다.



5-4 NX102 및 NX1P2의 태스크 사양과 기본 동작

여기에서는 멀티 코어 프로세서를 탑재한 NX102 및 NX1P2 CPU 유니트의 태스크 사양과 기본 동작에 대해 설명합니다.

5-4-1 NX102 및 NX1P2의 태스크 사양

태스크 사양은 다음과 같습니다.

항목	사양
태스크의 종류	• 프라이머리 정주기 태스크 • 정주기 태스크(우선도 17, 18) • 이벤트 태스크(우선도 8, 48)
태스크 수	• 프라이머리 정주기 태스크: 1 • 정주기 태스크: 0~2*1 • 이벤트 태스크: 0~32*2
1태스크당 프로그램의 수	최대 128
프라이머리 정주기 태스크의 태스크 주기	NX102: 1ms~32ms(250μs 단위로 설정 가능) NX1P2: 2ms~8ms(250μs 단위로 설정 가능)
정주기 태스크의 태스크 주기	• 우선도 17, 18 NX102: 1ms~100ms(250μs 단위로 설정 가능) NX1P2: 2ms~100ms(250μs 단위로 설정 가능) 정주기 태스크의 태스크 주기는 프라이머리 정주기 태스크의 정수배를 설정합니다. 각 태스크의 태스크 주기의 최소공배수가 600ms를 초과하는 조합은 선택할 수 없습니다.

*1. 실행 우선도 17, 18이 각각 0 또는 1개입니다.
*2. 실행 우선도 8, 48이 각각 0~32개이고, 모두 합해서 최대 32개입니다.

5-4-2 NX102 및 NX1P2의 태스크 분할의 개념

모든 프로그램은 어느 한 태스크에 할당해야 합니다. 프로그램의 특성에 따라 다음과 같은 개념으로 할당하는 태스크를 결정하십시오.

태스크	할당하는데 적합한 프로그램
프라이머리 정주기 태스크	• 주기별 I/O 리프레시, 사용자 프로그램 실행, 모션 제어, 시스템 공통 처리의 실행 주기가 엄격하게 지켜져야 함 • 실행 우선도가 가장 높고, 고속한 응답이 필요한 제어를 포함함 • 실행 우선도가 가장 높은 모션 제어 명령을 포함함
정주기 태스크(우선도 17, 18)	• 실행 우선도는 비교적 낮지만 주기마다 사용자 프로그램, 시스템 공통 처리를 실행해야 함 • 고속 응답을 필요로 하지 않는 연산 처리, 통신 처리를 포함함
이벤트 태스크	• 어떤 조건이 성립할 때만 실행함

5-4-3 NX102 및 NX1P2의 태스크 기본 동작

NX102 및 NX1P2 CPU 유니트는 복수의 태스크를 동시에 실행할 수 없습니다. 각 태스크가 어떤 순서로 실행되는지는 각각의 태스크에 설정된 실행 우선도에 따라 결정됩니다.

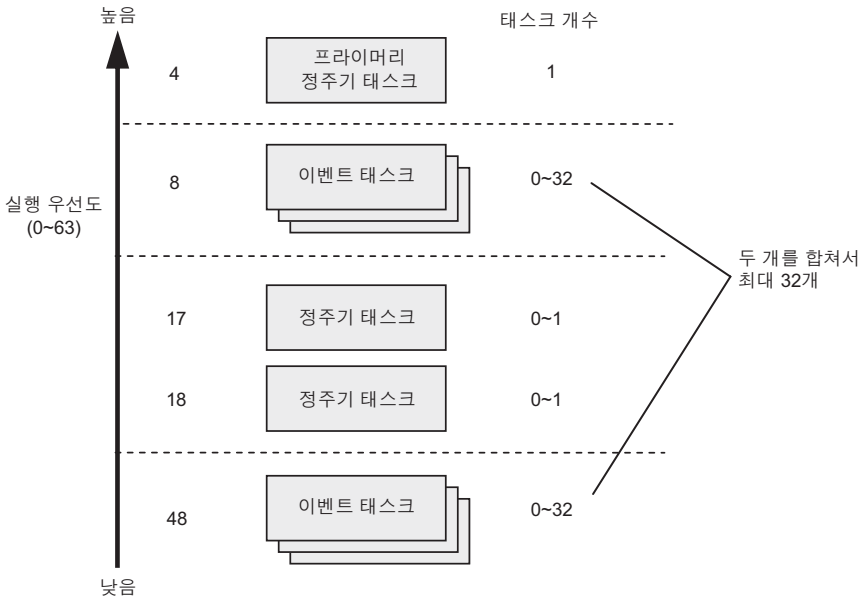
참고

- NX102 CPU 유니트에서는 태스크의 실행의 영향을 받지 않고, 태그 데이터 링크 서비스 및 시스템 서비스를 실행할 수 있습니다.
- NX1P2 CPU 유니트에서는 태스크의 실행의 영향을 받지 않고, 태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스를 실행할 수 있습니다.

태스크 실행 우선도

태스크에는 종류에 따라 실행 우선도가 정해져 있습니다.
어떤 태스크 Ta가 실행 중에 그보다 실행 우선도가 높은 태스크 Tb 실행 조건이 성립한 경우에는 Ta의 처리를 중단하고, 먼저 Tb를 실행합니다. 그리고 Tb의 처리가 종료된 후 Ta의 처리를 재개합니다.
태스크의 종류에 의한 실행 우선도는 다음과 같습니다. 실행 우선도 값이 작을수록 우선도가 높음을 나타냅니다.

태스크	실행 우선도	실행 우선도의 중복
프라이머리 정주기 태스크	4	–
정주기 태스크	17, 18	복수의 태스크에 동일한 실행 우선도를 설정할 수 없습니다.
이벤트 태스크	8, 48	복수의 태스크에 동일한 실행 우선도를 설정할 수 있습니다. 그 경우의 실행 순서에 대해서는 '5-4-5 NX102 및 NX1P2의 이벤트 태스크 실행 타이밍(5-41페이지)'을 참조하십시오.



프라이머리 정주기 태스크와 정주기 태스크의 태스크 주기

프라이머리 정주기 태스크와 정주기 태스크는 주기적으로 반복해서 실행되는 태스크입니다.
정주기 태스크의 태스크 주기는 반드시 프라이머리 정주기 태스크의 태스크 주기(' 프라이머리 주기' 라고 함)의 정수배가 되게 설정합니다. 따라서 몇 번에 1번씩 둘의 실행 타이밍이 일치합니다.
예를 들어 프라이머리 주기가 2ms, 정주기 태스크(실행 우선도 17)의 태스크 주기가 8ms인 경우에는 프라이머리 정주기 태스크가 4회 실행될 때마다 프라이머리 정주기 태스크와 정주기 태스크 실행 타이밍이 일치합니다.

참고

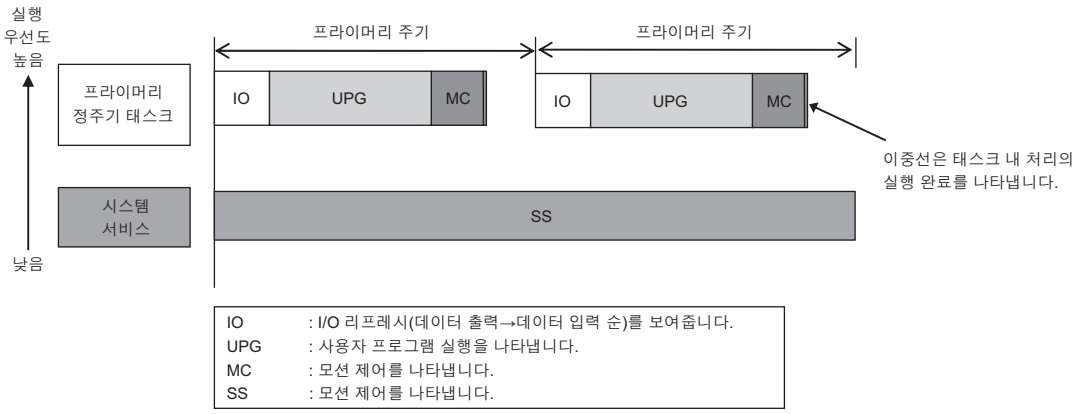
이벤트 태스크는 주기적으로 실행되는 것이 아니라 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 따라서 이벤트 태스크 실행 타이밍은 이벤트 태스크의 실행 조건이 성립한 타이밍과 이벤트 태스크의 실행 우선도에 따라 결정됩니다.

태스크의 실행 순서의 예

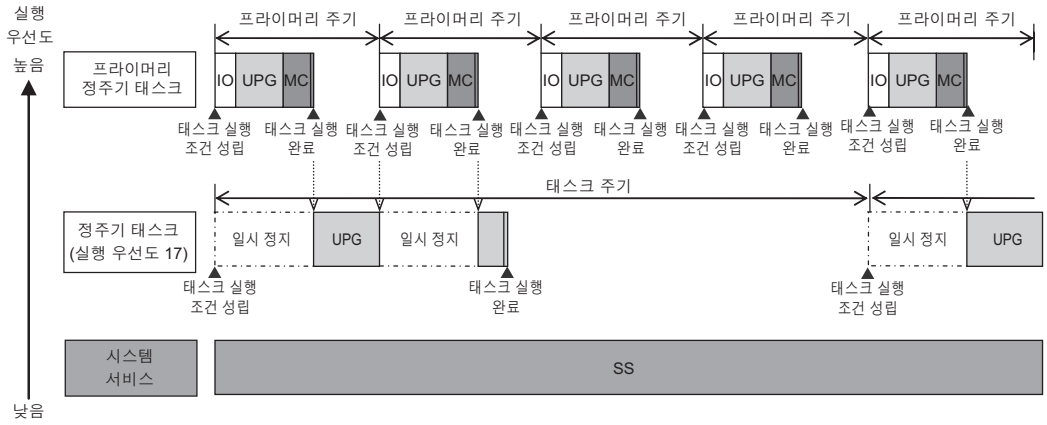
프라이머리 정주기 태스크와 정주기 태스크에 대해 실행 순서의 예시를 몇 가지 설명합니다.
이벤트 태스크의 실행 순서에 대해서는'5-4-5 NX102 및 NX1P2의 이벤트 태스크 실행 타이밍(5-41페이지)'을 참조하십시오.

- **프라이머리 정주기 태스크뿐인 경우**

  프라이머리 주기마다 프라이머리 정주기 태스크가 실행됩니다.
  그림의 시스템 서비스는 통신 처리 등 CPU 유니트가 태스크 이외에 실행하는 처리입니다. 시스템 서비스의 자세한 내용은 '시스템 서비스의 처리 내용(5-75페이지)'을 참조하십시오.



- **프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 17)가 혼재하는 경우**
  - 프라이머리 정주기 태스크는 실행 우선도가 가장 높으므로 확실하게 프라이머리 주기로 실행됩니다.
  - 정주기 태스크(실행 우선도 17)는 프라이머리 정주기 태스크보다 실행 우선도가 낮으므로 프라이머리 정주기 태스크가 실행되지 않는 시간에 실행됩니다.
  - 이 예에서는 정주기 태스크(실행 우선도 17)의 태스크 주기는 프라이머리 주기의 4배로 가정합니다. 따라서 정주기 태스크(실행 우선도 17)는 프라이머리 주기의 4회에 1번, 실행 시작 타이밍이 동기합니다.
  - 시스템 서비스는 태스크의 영향을 받지 않고 언제라도 실행됩니다.



사용상의 주의 사항

복수의 태스크 사이에서 동일한 변수의 읽기/쓰기를 할 가능성이 있는 경우에는 반드시 '변수의 태스크 간 배타 제어 기능'을 사용하십시오. 그렇게 하지 않으면 어떤 태스크의 실행 중에 다른 태스크에 의해 변수 값이 변경되는 경우가 있습니다.

자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

태스크와 동작 모드

CPU 유니트의 동작 모드와 태스크에는 다음과 같은 관계가 있습니다.

태스크	사양
프라이머리 정주기 태스크	• 운전 모드와 프로그램 모드에서 모두 실행됩니다.
정주기 태스크	• 단, 운전 모드에서만 사용자 프로그램 실행이 가능합니다.
이벤트 태스크	운전 모드에서만 실행됩니다.

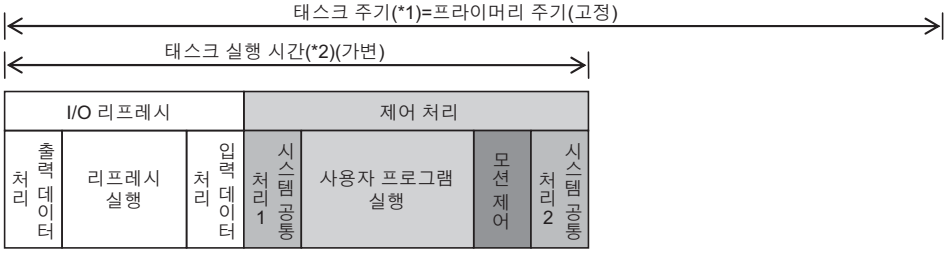
사용상의 주의 사항

- 동작 모드를 프로그램 모드에서 운전 모드로 전환한 시점에서 이벤트 태스크의 실행 조건이 이미 성립한 경우에 이벤트 태스크는 실행되지 않습니다. 이벤트 태스크가 실행되는 것은 동작 모드가 운전 모드인 상태에서 이벤트 태스크의 실행 조건이 성립하지 않다가 성립으로 변화한 때입니다.
- 동작 모드가 운전 모드에서도 전체 정지 이상 레벨의 이상 발생 중에는 이벤트 태스크가 실행되지 않습니다.

각 태스크 종류의 처리 내용

● **프라이머리 정주기 태스크**

태스크의 실행 우선도가 가장 높고, 고속·고정밀도로 실행할 수 있는 태스크입니다.
지정한 주기로 시스템 공통 처리, I/O 리프레시, 사용자 프로그램, 모션 제어가 실행됩니다.



*1 : 태스크 주기: 태스크가 정주기로 실행되는 설정된 고정 시간입니다.

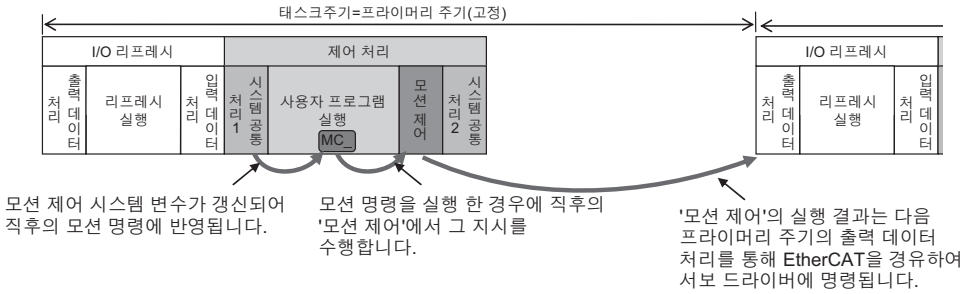
*2 : 태스크 실행 시간: 태스크 실행 조건 성립에서 실행 완료까지 실제로 걸린 시간입니다.

처리		처리 내용
I/O 리프레시	출력 데이터 처리	• I/O 리프레시를 실행하는 I/O(출력)에 대한 OUT 리프레시 데이터를 생성합니다. • 강제값 리프레시가 설정된 경우에는 OUT 리프레시 데이터에 강제값(출력)을 반영합니다.
	리프레시 실행	• I/O와의 데이터 교환을 실행합니다.
	입력 데이터 처리	• 이벤트 태스크의 조건식 일치 판정 • I/O 리프레시를 실행한 I/O(입력) 유니트의 IN 리프레시 데이터를 전송합니다. • 강제값 리프레시가 설정된 경우에는 전송한 IN 리프레시 데이터에 강제값(입력)을 반영합니다.
시스템 공통 처리 1		• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 모션 입력 처리*1 • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행		• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
모션 제어*2		• 프라이머리 정주기 태스크에 할당된 사용자 프로그램상의 모션 제어 명령에서의 모션 제어의 지시를 실행합니다. • 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에 대한 모션 출력 처리
시스템 공통 처리 2		• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. 모션 제어 시스템 변수의 축 현재값(위치, 속도, 토크) 및 모션 제어 시스템 변수의 서보 드라이버 상태가 갱신됩니다.

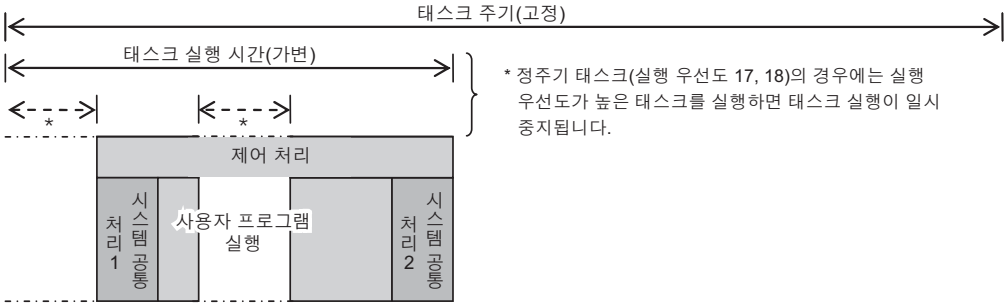
*2. 프라이머리 정주기 태스크의 사용자 프로그램 상의 모션 제어 명령인 경우에 다음 그림과 같이 직후의 '모션 제어'에서 그 명령에서의 지시가 실행되고, 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에서 서보 드라이버에 출력됩니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.



- **정주기 태스크(실행 우선도 17, 18)**

  프라이머리 주기의 정수배로 지정된 태스크 주기로 사용자 프로그램을 실행하는 태스크입니다. 태스크 전체에서 0~2개 할당할 수 있습니다.



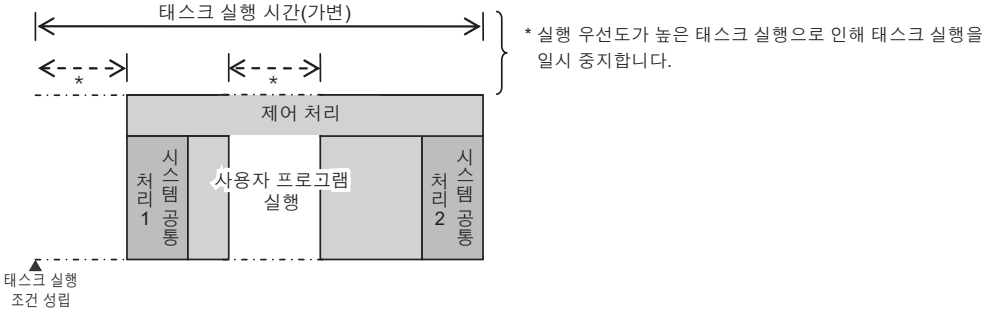
처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.

● **이벤트 태스크**

지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 태스크 전체에서 0~32개 할당할 수 있습니다.

이벤트 태스크의 처리 내용은 다음과 같습니다.



처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우)*1
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우)*1

*1. 변수의 태스크 간 배타 제어 처리의 자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

5-4-4 NX102 및 NX1P2의 이벤트 태스크 실행 조건

이벤트 태스크는 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 이벤트 태스크의 실행 조건에는 다음 2종류가 있습니다.

실행 조건	이벤트 태스크가 실행되는 타이밍	사용하는 경우
ActEventTask 명령에 의한 실행	ActEventTask 명령 실행 시	• 사용자 프로그램 내에서 어느 이벤트 태스크를 실행하는지 명시하고자 하는 경우 • 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정 시까지 변화할 가능성이 있는 경우
변수의 조건식 일치에 의한 실행	지정한 변수의 값이 지정한 조건식에 일치했을 때*1	이벤트 태스크 실행을 위한 사용자 프로그램을 생략해서 사용자 프로그램을 간략화하고자 하는 경우

*1. 지정한 변수의 값이 지정한 조건식에 일치했는지 여부를 판정하는 타이밍에 대해서는 '실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍(5-25페이지)'을 참조하십시오.

안전상의 중요 사항

이벤트 태스크의 실행 조건이 변수의 조건식과 일치하고 조건식에 다음 변수를 지정하는 경우에 조건이 일치할 때 이벤트 태스크가 기동하지 않거나 조건 불일치 시에 이벤트 태스크가 기동되는 경우가 있습니다.

- 모션 제어 시스템 변수를 제외하고, 데이터 크기가 16비트 이상인 구조체 멤버
- 데이터 크기가 16비트 이상의 배열 요소

위의 변수를 지정하여 일치 판정을 실시하는 경우에는 다음 중 하나를 실행하십시오.

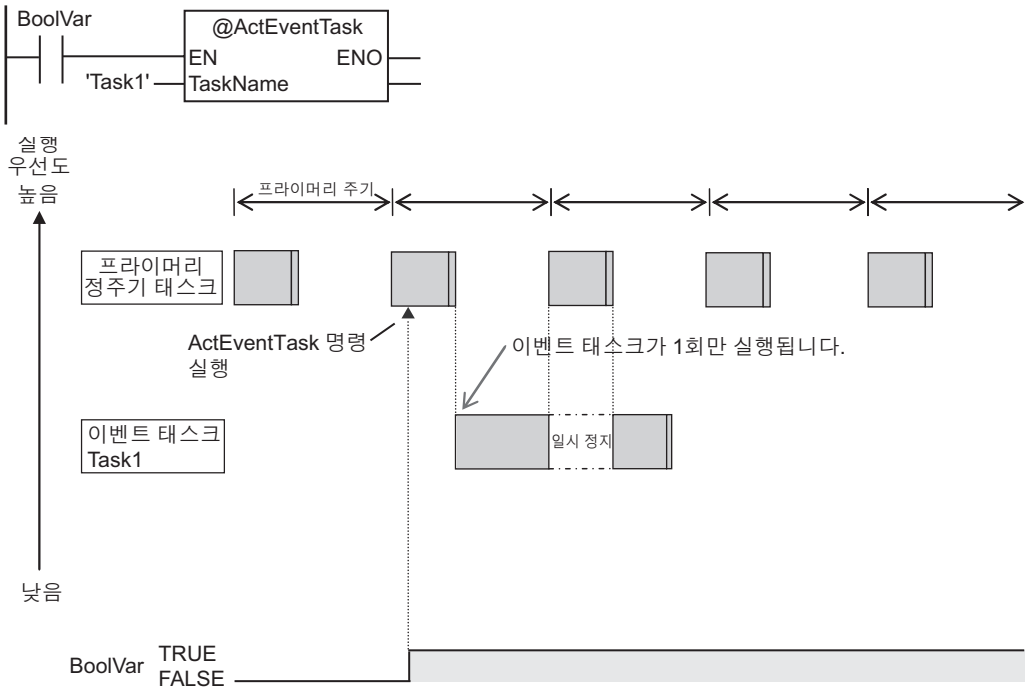
- 64비트 이외의 기본 데이터 형식의 내부 변수에 복사하고, 복사한 내부 변수를 참조함
- 변수의 태스크 간 배타 제어 설정 기능을 사용하고, 프라이머리 정주기 태스크를 갱신 태스크로 함

ActEventTask 명령에 의한 이벤트 태스크 실행

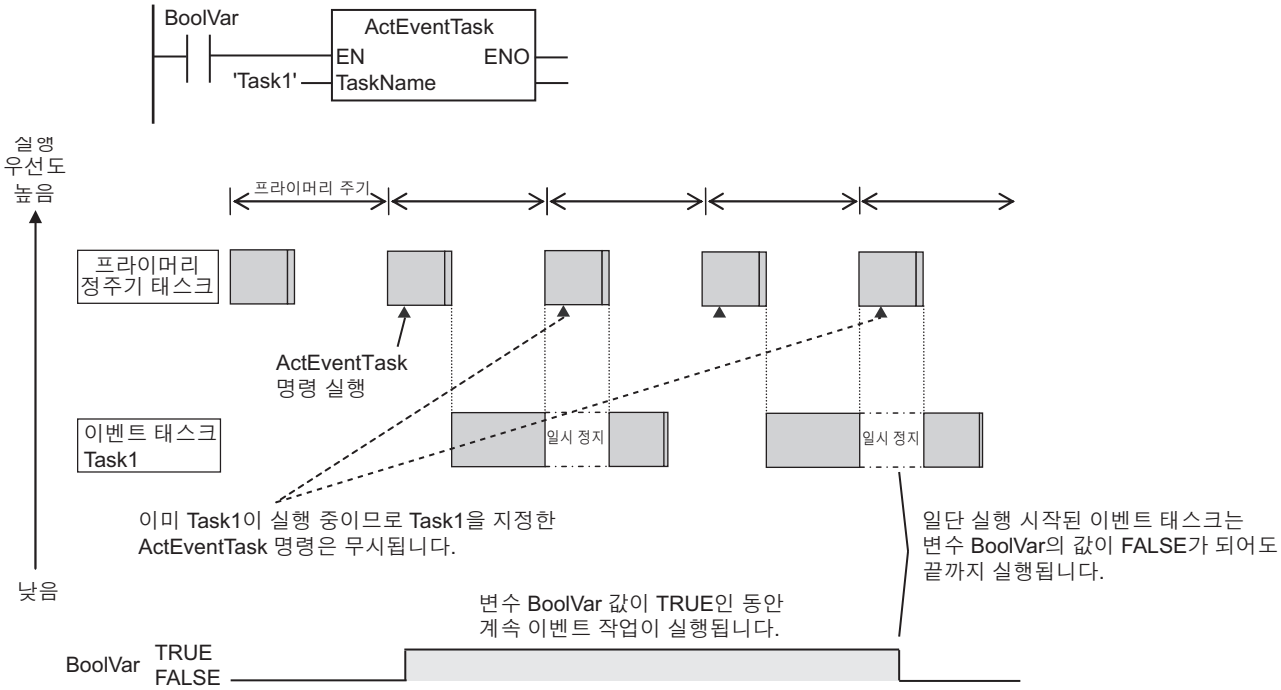
사용자 프로그램 내에서 ActEventTask 명령이 실행되면 지정한 이벤트 태스크가 1회 실행됩니다.
ActEventTask 명령의 자세한 사양은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.
ActEventTask 명령에 의한 방법을 사용하면 사용자 프로그램을 봤을 때, 어느 이벤트 태스크를 기동하는지 잘 알 수 있습니다. 또한, 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정까지 변화할 가능성이 있는 경우에도 유효합니다.

- **ActEventTask 명령을 사용한 사용자 프로그램 예**

  예 1) 변수의 값이 변화했을 때 이벤트 태스크를 1회만 실행하는 경우
  다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하면 BOOL 형식 변수 BoolVar의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크 Task1이 1회만 실행됩니다.



예 2) 변수가 지정한 값인 동안 반복해서 이벤트 태스크를 실행하는 경우
다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하지 않으면 BOOL 형식 변수 BoolVar의 값이 TRUE인 동안 반복해서 이벤트 태스크 Task1이 실행됩니다. 단, Task1이 실행 중에 Task1을 지정한 ActEventTask 명령은 실행해도 무시됩니다. 또한, 일단 실행 시작된 이벤트 태스크는 실행 도중에 BoolVar의 값이 FALSE가 되어도 마지막까지 실행됩니다.



변수의 조건식 일치에 의한 이벤트 태스크 실행

변수의 조건식 일치에 의한 방법은 Sysmac Studio에서 지정한 변수 값이 지정한 조건식에 일치한 경우 이벤트 태스크가 1회 실행된다는 것입니다. 해당 변수의 값이 조건식에 일치하는 동안 이벤트 태스크가 계속 실행는 것이 아니라 변수 값이 조건식에 불일치에서 일치로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.
변수의 조건 일치에 의한 방법에서는 이벤트 태스크 실행을 위한 사용자 프로그램이 필요 없어집니다.

- **조건식을 지정할 수 있는 변수의 종류**

  조건식을 지정할 수 있는 변수의 종류는 다음과 같습니다.

변수의 종류		지정 가능 여부
시스템 정의 변수		가능*1
준 사용자 정의 변수		가능
사용자 정의 변수	글로벌 변수	가능
	프로그램 내에서 사용하는 변수	가능
	펑션 블록 내에서 사용하는 변수	가능*2
	펑션 내에서 사용하는 변수	불가능

*1. 다음과 같은 변수는 제외합니다.
EN, ENO, P_Off, P_CY, P_First_RunMode, P_First_Run, P_PRGER
*2. 입출력 변수는 제외합니다.

● **조건식을 지정할 수 있는 변수의 데이터 형식**

조건식을 지정할 수 있는 변수의 데이터 형식은 다음과 같습니다.

데이터 형식의 분류	데이터 형식		지정 가능 여부
기본 데이터 형식	부울 형식, 비트 열 형식, 정수 형식, 실수 형식		가능
	지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		불가능
데이터 형식 지정	배열 지정	배열 전체	불가능
		요소	가능*1
파생 데이터 형식	구조체 형식	구조체 전체	불가능
		멤버	가능*2
	공용체 형식	공용체 전체	불가능
		멤버	가능*2
	열거 형식		가능

*1. 배열 요소의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

*2. 멤버의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

● **지정할 수 있는 조건식**

지정할 수 있는 조건식은 조건식을 지정하는 변수의 데이터 형식에 따라 다음과 같이 정해져 있습니다.

조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다.

데이터 형식	지정할 수 있는 조건식
부울 형식, 부울 형식의 배열 요소, 부울 형식의 구조체 멤버, 부울 형식의 공용체 멤버	상승 FALSE→TRUE
	하강 TRUE→FALSE
비트 열 형식, 정수 형식, 실수 형식 또는 이들을 데이터 형식으로 하는 배열 요소, 구조체 멤버, 공용체 멤버	변수 = {비교 상수}
	변수 ≠ {비교 상수}
	변수 > {비교 상수}
	변수≧{비교 상수}
	변수 < {비교 상수}
	변수≦{비교 상수}

● **비교 상수 값의 유효 범위**

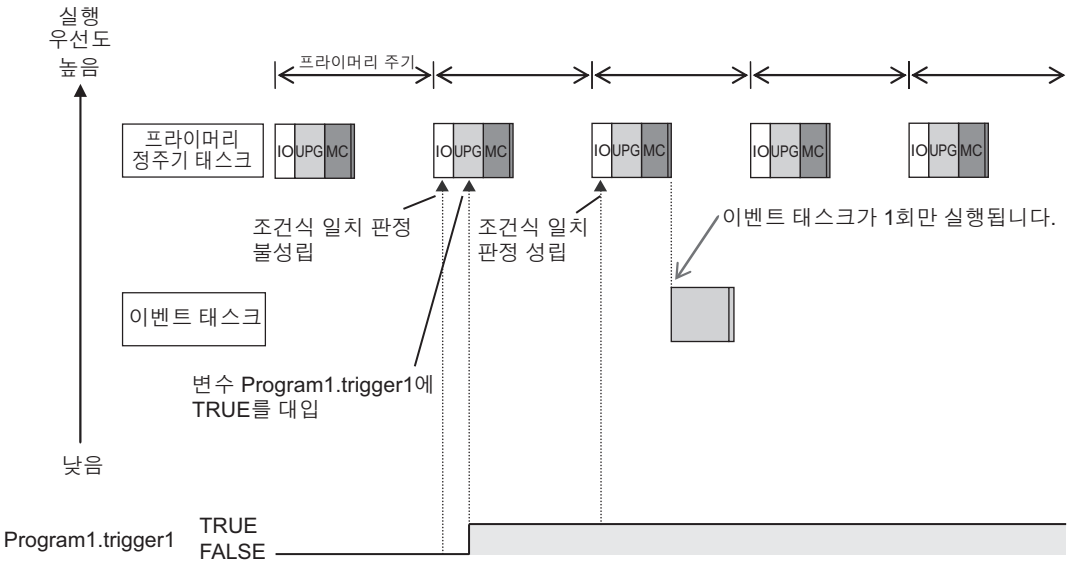
조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다. 비교 상수의 값의 유효 범위는 조건식을 지정하는 변수의 데이터 형식 값의 유효 범위와 동일합니다.

각 데이터 형식의 값의 유효 범위는 📖'기본 데이터 형식과 파생 데이터 형식(6-31페이지)'을 참조하십시오.

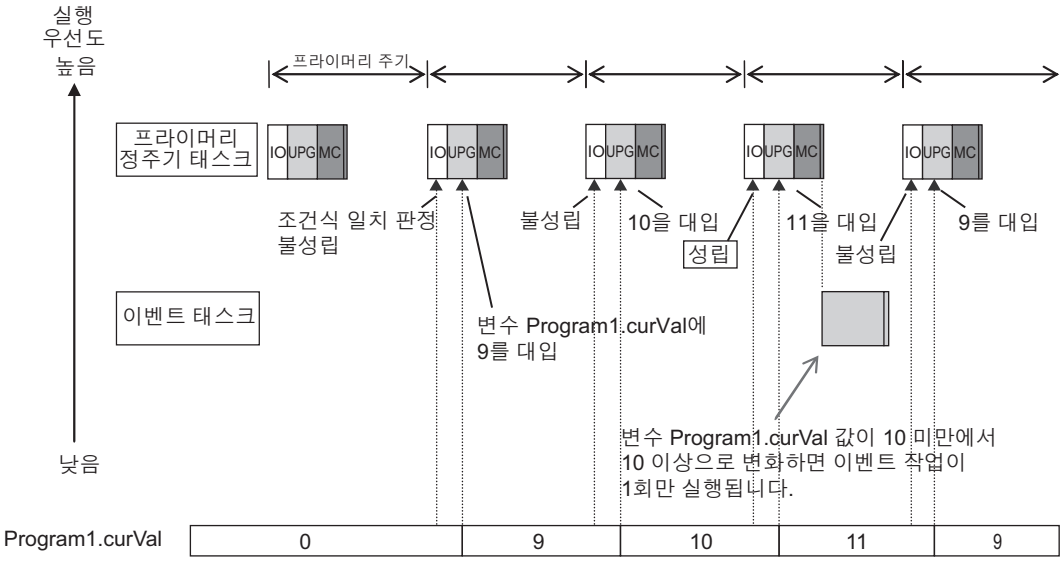
예를 들어 조건식을 지정하는 변수의 데이터 형식이 BYTE인 경우에 비교 상수 값의 유효 범위는 BYTE#16#00~BYTE#16#FF입니다.

- **변수의 조건식 일치에 의한 이벤트 태스크 실행 예**

예 1) 이벤트 태스크 실행 조건을 BOOL 형식 변수 Program1.trigger1의 상승으로 한 경우
Program1.trigger1의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



예 2) 이벤트 태스크 실행 조건을 INT 형식 변수 Program1.curVal?10로 한 경우
Program1.curVal의 값이 10미만에서 10 이상으로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



사용상의 주의 사항

프라이머리 정주기 태스크 내에서 지정한 변수 값이 변화한 경우에 다음 프라이머리 정주기 태스크로 조건식과의 일치가 판정됩니다. 따라서 이벤트 태스크가 실행되는 것은 변수 값이 조건식을 충족한 주기의 다음 프라이머리 정주기 태스크로 조건식 일치 판정이 실행된 이후입니다.

5-4-5 NX102 및 NX1P2의 이벤트 태스크 실행 타이밍

이벤트 태스크의 실행 우선도는 8 또는 48입니다. 다른 태스크 실행 중에 이벤트 태스크 실행 조건이 성립한 경우에 실행 우선도가 높은 태스크가 먼저 실행되고, 실행 우선도가 낮은 태스크의 실행은 중단됩니다. 이는 프라이머리 정주기 태스크 및 정주기 태스크와 동일합니다.

복수의 태스크 사이에서 동일한 변수를 읽거나 쓸 가능성이 있는 경우에는에 반드시 다음 기능을 사용하여 프라이머리 정주기 태스크, 정주기 태스크 또는 실행 우선도가 다른 이벤트 태스크와의 실행을 제어하십시오.

목적	사용하는 기능
이벤트 태스크와 다른 태스크로 동일한 글로벌 변수에 액세스하는 경우	변수의 태스크 간 배타 제어 기능 • 변수의 태스크 간 배타 제어 설정 기능 • Lock(태스크 간 배타 잠금) 명령 • Unlock(태스크 간 배타 잠금 해제) 명령
다른 태스크의 실행 상태를 확인하는 경우	Task_IsActive(태스크 실행 중 판정) 명령

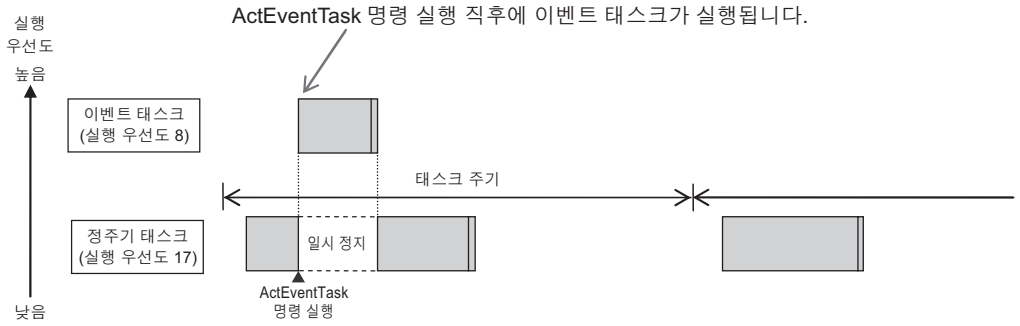
자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

이벤트 태스크 실행 조건에 따라 이벤트 태스크 실행 타이밍이 다릅니다.
또한, 이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. 동일한 실행 우선도의 여러 이벤트 태스크 실행 조건이 성립한 경우에는 주의해야 합니다.

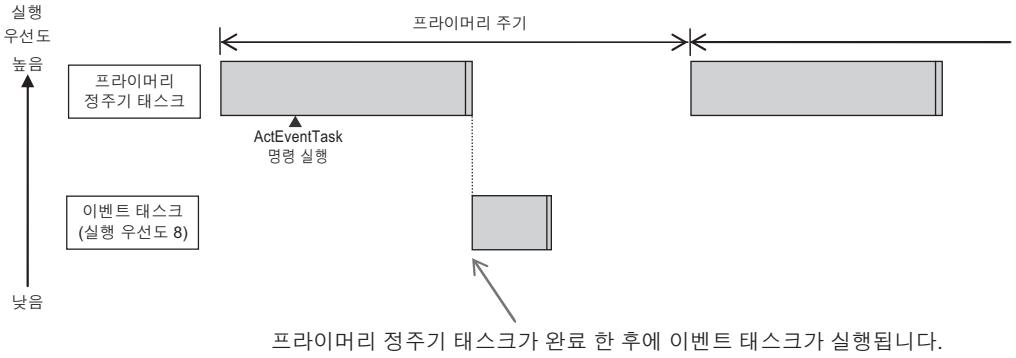
이벤트 태스크 실행 조건에 의한 실행 타이밍의 차이

이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우와 변수의 조건식 일치에 의한 경우에는 실행 조건의 성립 타이밍이 다릅니다.

- **실행 조건이 ActEventTask 명령에 의한 경우의 실행 타이밍**
  이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우에는 ActEventTask 명령의 실행직후에 이벤트 태스크 실행 조건이 성립합니다. 실행 조건이 성립한 이벤트 태스크는 태스크의 실행 우선도에 따라 실행이 시작됩니다.
  예 1) 명령을 실행하는 태스크보다 실행 우선도가 높은 이벤트 태스크를 기동하는 경우



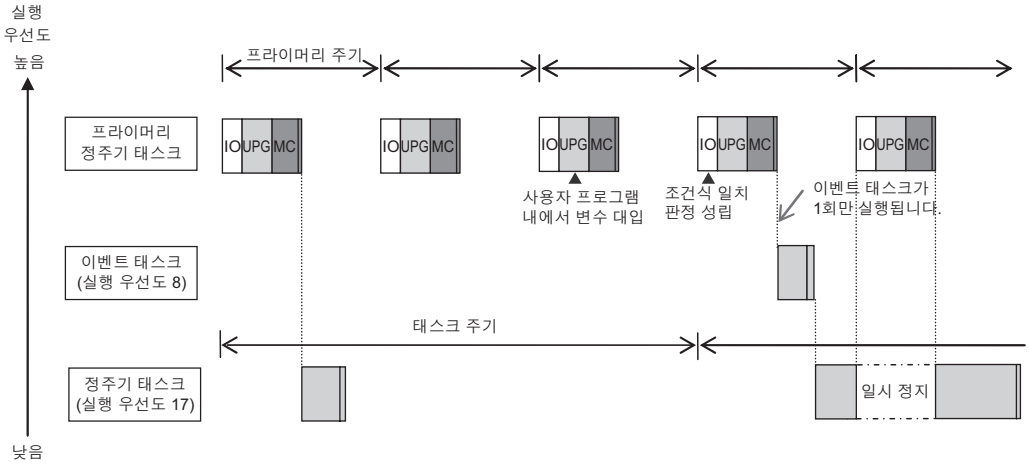
예 2) 명령을 실행하는 태스크보다 실행 우선도가 낮은 이벤트 태스크를 기동하는 경우



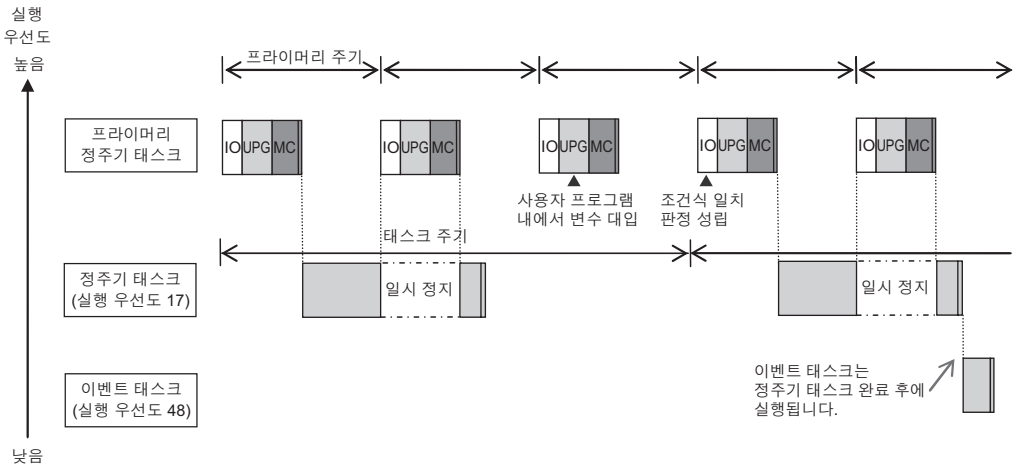
- **실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍**

  조건식 일치 판정은 프라이머리 정주기 태스크 중에 실행됩니다. 조건식 판정 처리에서 조건이 일치했을 때 이벤트 태스크 실행 조건이 성립합니다. 실행 조건이 성립한 이벤트 태스크는 태스크의 실행 우선도에 따라 실행이 시작됩니다.

  예 1) 정주기 태스크(실행 우선도 17)와 이벤트 태스크(실행 우선도 8)가 존재하는 경우
  이벤트 태스크(실행 우선도 8) 쪽이 정주기 태스크(실행 우선도 17)보다 실행 우선도가 높으므로 정주기 태스크(실행 우선도 17)는 이벤트 태스크 다음에 실행됩니다.



  예 2) 정주기 태스크(실행 우선도 17)와 이벤트 태스크(실행 우선도 48)가 존재하는 경우
  이벤트 태스크 쪽이 정주기 태스크(실행 우선도 17)보다 실행 우선도가 낮으므로 정주기 태스크(실행 우선도 17)가 완료된 후에 이벤트 태스크가 실행됩니다.



사용상의 주의 사항

- NX102 및 NX1P2 CPU 유니트에서는 조건식 일치가 프라이머리 정주기 태스크의 I/O 리프레시에 의해 발생하는 경우 및 프라이머리 정주기 태스크에 할당된 사용자 프로그램 실행에서 발생하는 경우에도 이벤트 태스크 실행 조건이 성립하는 타이밍은 다음과 같이 동일합니다.

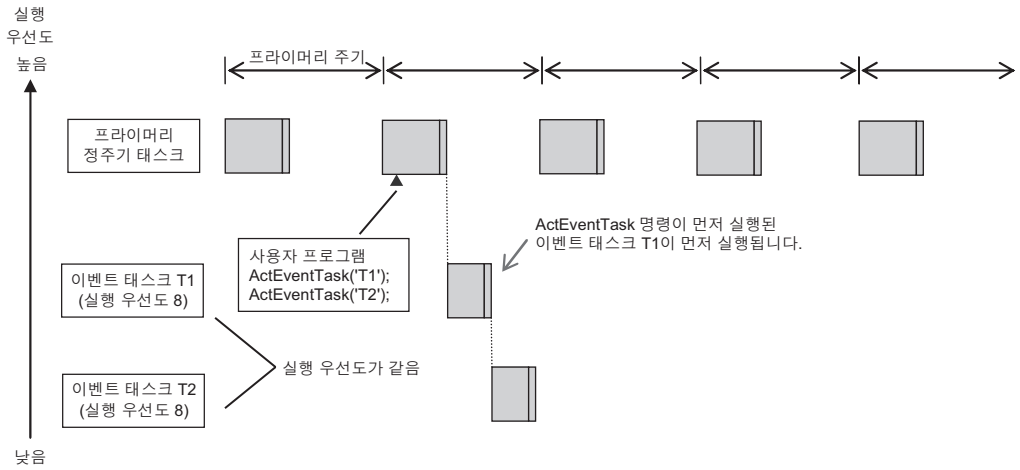
조건식 일치 발생 요인	이벤트 태스크 실행 조건이 성립하는 타이밍
프라이머리 정주기 태스크의 I/O 리프레시	다음 프라이머리 정주기 태스크 판정 처리
프라이머리 정주기 태스크의 사용자 프로그램 실행	다음 프라이머리 정주기 태스크 판정 처리

- 이벤트 태스크가 실행되는 것은 1회 이전 판정 시 조건식 불일치이고, 다음 판정 시 조건식 일치가 발생한 경우입니다. 따라서 불일치에서 일치로 변화가 있었다고 해도 다음 판정 시까지 불일치로 되돌아간 경우에는 이벤트 태스크는 실행되지 않습니다.

동일한 실행 우선도의 이벤트 태스크 실행 타이밍

이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. ActEventTask 명령에 의해 동일한 실행 우선도의 복수의 이벤트 태스크 실행 조건이 성립한 경우에는 명령을 실행한 순서대로 이벤트 태스크가 실행됩니다.

예 1) 2가지 ActEventTask 명령이 실행된 경우
다음 그림과 같이 먼저 실행된 ActEventTask 명령에 의해 지정된 이벤트 태스크 T1이 이벤트 태스크 T2보다도 먼저 실행됩니다.



예 2) 변수의 조건식 일치와 ActEventTask 명령이 혼재하는 경우

이벤트 태스크 T1, T2, T3 실행 조건이 다음과 같이 설정되었다고 가정합니다.

- T1: 변수 VarA의 조건식 일치
- T2: 변수 VarB의 조건식 일치
- T3: ActEventTask 명령

다음 그림과 같이 동일한 프라이머리 주기 중에 I/O 리프레시에 의한 VarA의 조건식 일치, 사용자 프로그램에서의 ActEventTask 명령 실행, 사용자 프로그램에서의 VarB의 조건식 일치가 발생한 경우에는 다음 순서로 동작합니다.

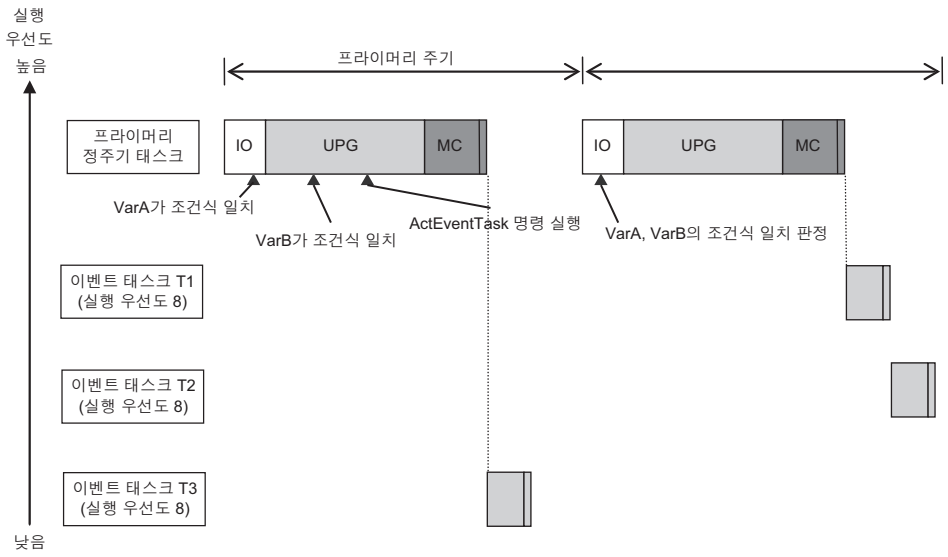
(1) I/O 리프레시에서 VarA가 조건식에 일치합니다.

(2) 사용자 프로그램에서 VarB가 조건식에 일치합니다.

(3) 이 시점에서는 아직 조건식의 일치 판정이 실행되지 않았으므로 T1과 T2는 실행되지 않습니다.

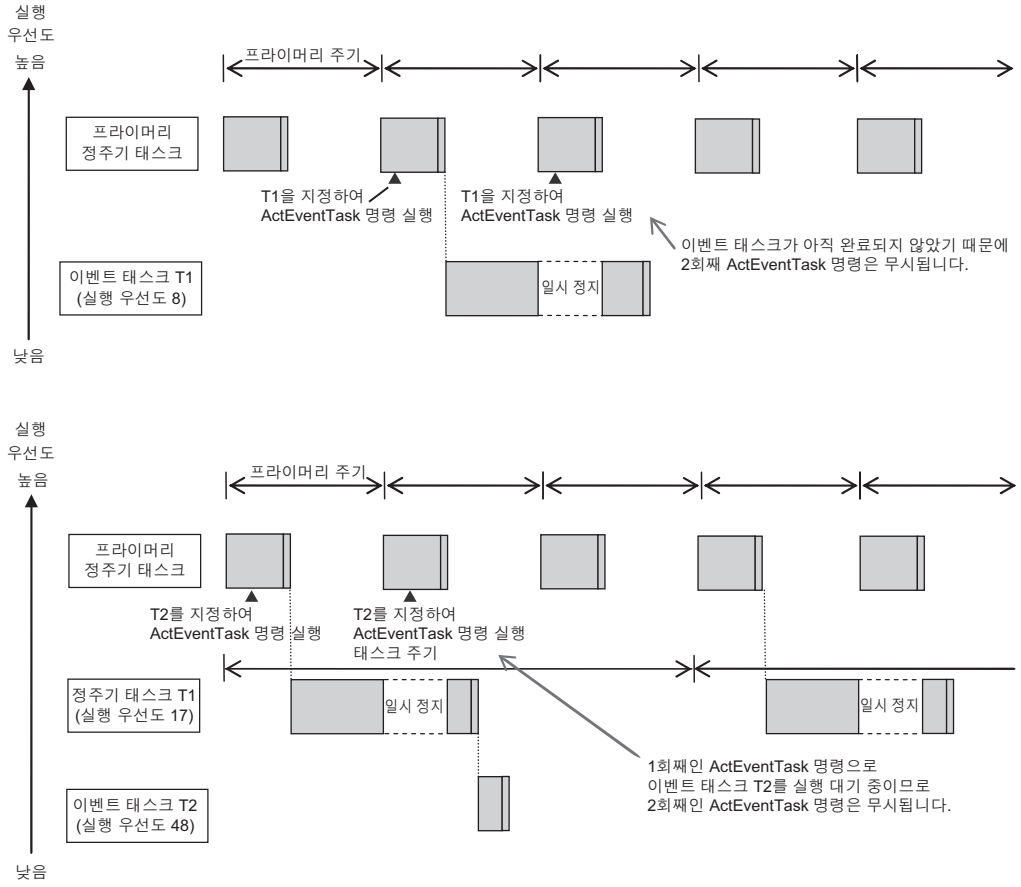
(4) 사용자 프로그램으로 ActEventTask 명령이 실행되고, T30이 실행됩니다.

(5) 다음 프라이머리 정주기 태스크의 리프레시 실행으로 VarA와 VarB가 조건식에 일치한다고 판정되고, T1과 T2가 실행됩니다.
동일한 주기에 복수의 조건식이 일치한다고 판정된 경우에 이벤트 태스크의 실행 순서가 불안정해집니다. 다음 그림은 T1이 먼저 실행된 경우입니다.



5-4-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작

이벤트 태스크 완료 전에 다시 해당 이벤트 태스크 실행 조건이 성립한 경우에는 2번째 실행 조건 성립은 무시됩니다. 이벤트 태스크 완료 전은 이벤트 태스크 실행 중 및 이벤트 태스크 실행 대기 중을 모두 포함합니다. 또한, 일단 실행이 시작된 이벤트 태스크는 실행 조건이 성립하지 않아도 마지막까지 실행됩니다.



5-5 NJ 시리즈의 태스크 사양과 기본 동작

여기에서는 NJ 시리즈 CPU 유니트의 태스크 사양과 기본 동작에 대해 설명합니다.

5-5-1 NJ 시리즈의 태스크 사양

태스크 사양은 다음과 같습니다.

항목	사양
태스크의 종류	• 프라이머리 정주기 태스크 • 정주기 태스크(우선도 16, 17, 18) • 이벤트 태스크(우선도 8, 48)
태스크 수	• 프라이머리 정주기 태스크: 1 • 정주기 태스크: 0~3*1 • 이벤트 태스크: 0~32*2
1태스크당 프로그램의 수	최대 128
프라이머리 정주기 태스크의 태스크 주기	500μs*3 , 1ms, 2ms, 4ms
정주기 태스크의 태스크 주기	정주기 태스크의 태스크 주기는 프라이머리 정주기 태스크의 정수배를 설정합니다. 설정할 수 있는 주기는 뒤에 설명하는 '정주기 태스크로 설정 가능한 태스크 주기'의 표를 참조하십시오.

*1. 실행 우선도 16, 17, 18이 각각 0 또는 1개입니다.
*2. 실행 우선도 8, 48이 각각 0~32개이고, 모두 합해서 최대 32개입니다.
*3. NJ301-□□□□에서는 유니트 버전 Ver.1.03 이상에서 사용할 수 있습니다.
NJ101-□□□□에서는 사용할 수 없습니다.

● 정주기 태스크로 설정 가능한 태스크 주기

프라이머리 정주기 태스크의 태스크 주기	설정 가능한 정주기 태스크의 태스크 주기
500μs*1	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
1ms	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
2ms	2ms, 4ms, 8ms, 10ms, 20ms, 30ms, 40ms, 50ms, 60ms, 100ms
4ms	4ms, 8ms, 20ms, 40ms, 60ms, 100ms

*1. NJ301-□□□□에서는 유니트 버전 Ver.1.03 이상에서 사용할 수 있습니다.
NJ101-□□□□에서는 사용할 수 없습니다.

5-5-2 NJ 시리즈의 태스크 분할의 개념

모든 프로그램은 어느 한 태스크에 할당해야 합니다. 프로그램의 특성에 따라 다음과 같은 개념으로 할당하는 태스크를 결정하십시오.

태스크	할당하는데 적합한 프로그램
프라이머리 정주기 태스크	• I/O 리프레시의 실행 주기가 엄격하게 지켜져야 함 • 실행 우선도가 가장 높음 • 실행 우선도가 높은 모션 제어 명령을 포함함
정주기 태스크(우선도 16)	• I/O 리프레시의 실행을 동반함 • 실행 우선도는 비교적 낮지만 주기적으로 실행해야 함 • 실행 우선도가 비교적 낮은 모션 제어 명령을 포함함
정주기 태스크(우선도 17, 18)	• 실행 우선도는 비교적 낮지만 주기적으로 실행해야 함
이벤트 태스크	• 어떤 조건이 성립할 때만 실행함

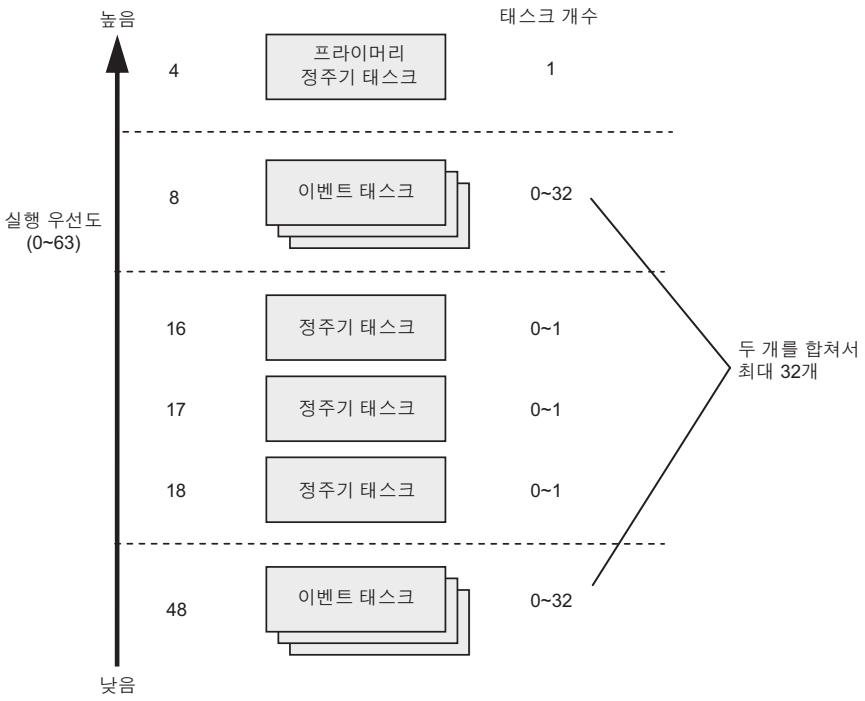
5-5-3 NJ 시리즈의 태스크 기본 동작

CPU 유니트는 복수의 태스크를 동시 실행할 수 없습니다. 각 태스크가 어떤 순서로 실행되는지는 각각의 태스크에 설정된 실행 우선도에 따라 결정됩니다.

태스크 실행 우선도

태스크에는 종류에 따라 실행 우선도가 정해져 있습니다.
어떤 태스크 Ta가 실행 중에 그보다 실행 우선도가 높은 태스크 Tb 실행 조건이 성립한 경우에는 Ta의 처리를 중단하고, 먼저 Tb를 실행합니다. 그리고 Tb의 처리가 종료된 후 Ta의 처리를 재개합니다.
태스크의 종류에 의한 실행 우선도는 다음과 같습니다. 실행 우선도 값이 작을수록 우선도가 높음을 나타냅니다.

태스크	실행 우선도	실행 우선도의 중복
프라이머리 정주기 태스크	4	–
정주기 태스크	16, 17, 18	복수의 태스크에 동일한 실행 우선도를 설정할 수 없습니다.
이벤트 태스크	8, 48	복수의 태스크에 동일한 실행 우선도를 설정할 수 있습니다. 그 경우의 실행 순서에 대해서는'5-5-5 NJ 시리즈의 이벤트 태스크 실행 타이밍(5-60페이지)'을 참조하십시오.



프라이머리 정주기 태스크와 정주기 태스크의 태스크 주기

프라이머리 정주기 태스크와 정주기 태스크는 주기적으로 반복해서 실행되는 태스크입니다.
정주기 태스크의 태스크 주기는 반드시 프라이머리 정주기 태스크의 태스크 주기(' 프라이머리 주기'라고 함)의 정수배가 되게 설정합니다. 따라서 몇 번에 1번씩 둘의 실행 타이밍이 일치합니다.
예를 들어 프라이머리 주기가 1ms, 정주기 태스크(실행 우선도 16)의 태스크 주기가 4ms인 경우에는 프라이머리 정주기 태스크가 4회 실행될 때마다 프라이머리 정주기 태스크와 정주기 태스크 실행 타이밍이 일치합니다.

참고

이벤트 태스크는 주기적으로 실행되는 것이 아니라 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 따라서 이벤트 태스크 실행 타이밍은 이벤트 태스크의 실행 조건이 성립한 타이밍과 이벤트 태스크의 실행 우선도에 따라 결정됩니다.

태스크의 실행 순서의 예

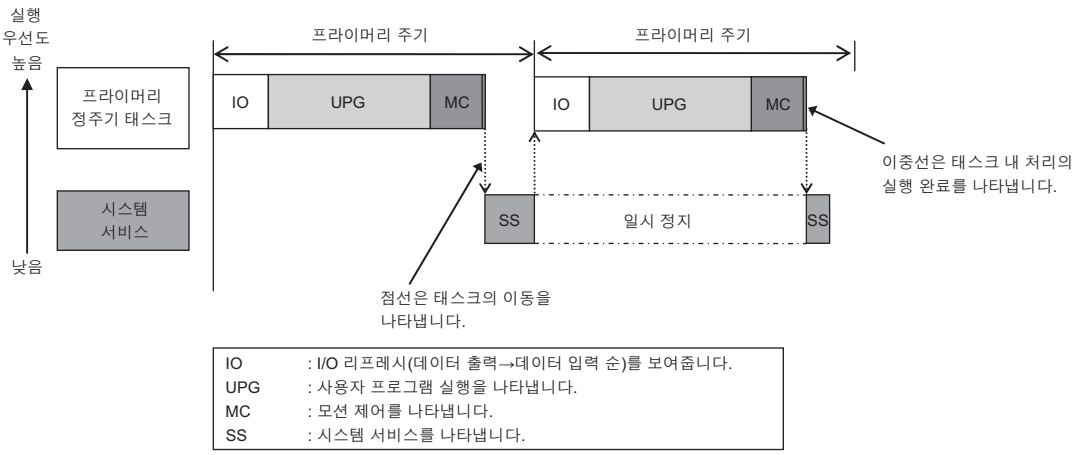
프라이머리 정주기 태스크와 정주기 태스크에 대해 실행 순서의 예시를 몇 가지 설명합니다.

이벤트 태스크의 실행 순서에 대해서는 '5-5-5 NJ 시리즈의 이벤트 태스크 실행 타이밍(5-60페이지)'을 참조하십시오.

- **프라이머리 정주기 태스크뿐인 경우**

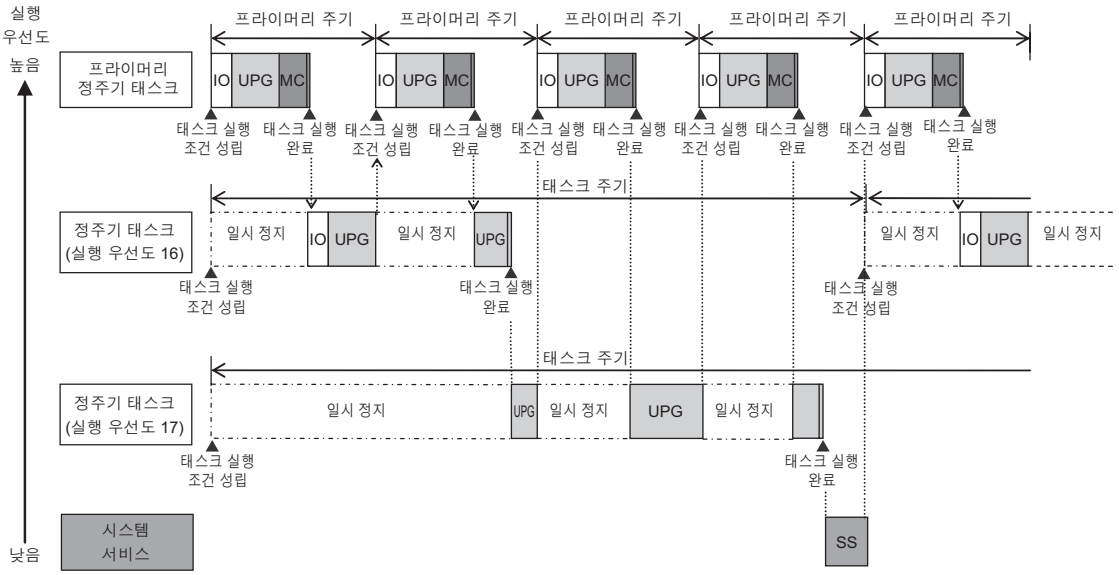
  프라이머리 주기마다 프라이머리 정주기 태스크가 실행됩니다.

  그림의 시스템 서비스는 통신 처리 등 CPU 유니트가 태스크 이외에 실행하는 처리입니다. 시스템 서비스는 태스크 실행의 비는 시간에 실행됩니다. 시스템 서비스의 자세한 내용은 '시스템 서비스의 처리 내용(5-75페이지)'을 참조하십시오.



IO	: I/O 리프레시(데이터 출력→데이터 입력 순)를 보여줍니다.
UPG	: 사용자 프로그램 실행을 나타냅니다.
MC	: 모션 제어를 나타냅니다.
SS	: 시스템 서비스를 나타냅니다.

- **프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 16), 정주기 태스크(실행 우선도 17)가 혼재하는 경우**
  - 프라이머리 정주기 태스크는 실행 우선도가 가장 높으므로 확실하게 프라이머리 주기로 실행됩니다.
  - 정주기 태스크(실행 우선도 16)는 프라이머리 정주기 태스크보다 실행 우선도가 낮으므로 프라이머리 정주기 태스크가 실행되지 않는 시간에 실행됩니다.
  - 정주기 태스크(실행 우선도 17)는 실행 우선도가 더 낮으므로 위의 2가지 태스크 모두 실행되지 않는 시간에 실행됩니다.
  - 이 예에서는 정주기 태스크(실행 우선도 16)의 태스크 주기는 프라이머리 주기의 4배로 가정합니다. 따라서 프라이머리 주기의 4회에 1번, 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16)의 실행 시작 타이밍이 동기합니다.
  - 시스템 서비스는 모든 태스크의 비는 시간에 실행됩니다.



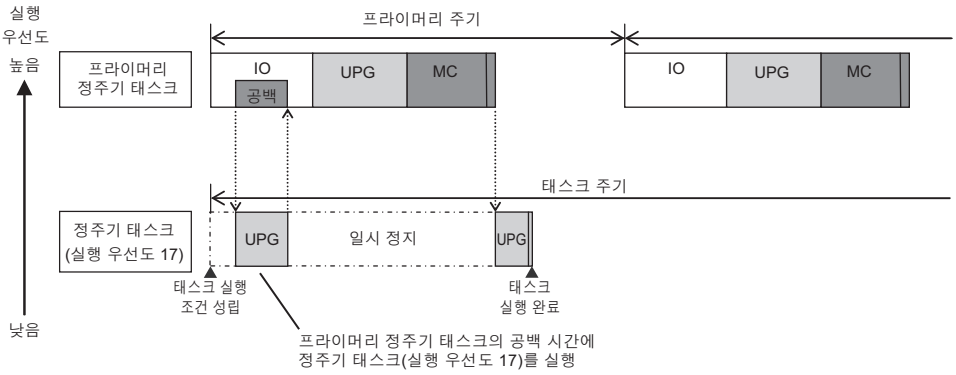
사용상의 주의 사항

복수의 태스크 사이에서 동일한 변수의 읽기/쓰기를 할 가능성이 있는 경우에는 반드시 '변수의 태스크 간 배타 제어 기능'을 사용하십시오. 그렇게 하지 않으면 어떤 태스크의 실행 중에 다른 태스크에 의해 변수 값이 변경되는 경우가 있습니다.

자세한 내용은 📖'5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

태스크의 비는 시간을 이용한 다른 태스크 실행

실행 우선도가 높은 태스크의 실행을 중단하고 그보다 실행 우선도가 낮은 태스크의 실행을 시작하지는 않습니다. 그러나 NJ 시리즈 CPU 유니트에서는 실행 우선도가 높은 태스크에 비는 시간이 발생한 경우에 그 시간을 이용하여 실행 우선도가 낮은 태스크가 실행을 시작하는 경우가 있습니다. 예를 들어 I/O 리프레시의 데이터 전송 완료 대기 시간 등입니다. 단, 실행 우선도가 높은 태스크가 처리를 재개하면 즉시 실행 우선도가 낮은 태스크의 처리를 중단합니다. 그 양상은 다음과 같습니다.



사용상의 주의 사항

프라이머리 정주기 태스크의 비는 시간이 발생해도 정주기 태스크(실행 우선도 16)의 사용자 프로그램은 반드시 프라이머리 정주기 태스크가 완료된 후 실행됩니다.
단, NJ 시리즈 CPU 유니트에서는 상기 그림과 같이 정주기 태스크(실행 우선도 17 또는 18)에는 그와 같은 제한이 없습니다.
또한, NJ 시리즈 CPU 유니트에서는 정주기 태스크(실행 우선도 16)의 I/O 리프레시에도 그러한 제한이 없고, 프라이머리 정주기 태스크의 비는 시간에 실행되는 경우가 있습니다.

태스크와 동작 모드

CPU 유니트의 동작 모드와 태스크에는 다음과 같은 관계가 있습니다.

태스크	사양
프라이머리 정주기 태스크	• 운전 모드와 프로그램 모드에서 모두 실행됩니다.
정주기 태스크	• 단, 운전 모드에서만 사용자 프로그램 실행이 가능합니다.
이벤트 태스크	운전 모드에서만 실행됩니다.

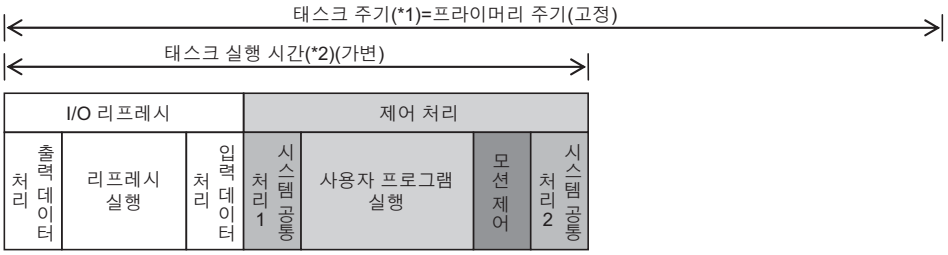
사용상의 주의 사항

- 동작 모드를 프로그램 모드에서 운전 모드로 전환한 시점에서 이벤트 태스크의 실행 조건이 이미 성립한 경우에 이벤트 태스크는 실행되지 않습니다. 이벤트 태스크가 실행되는 것은 동작 모드가 운전 모드인 상태에서 이벤트 태스크의 실행 조건이 성립하지 않다가 성립으로 변화한 때입니다.
- 동작 모드가 운전 모드에서도 전체 정지 이상 레벨의 이상 발생 중에는 이벤트 태스크가 실행되지 않습니다.

각 태스크 종류의 처리 내용

- **프라이머리 정주기 태스크**

  태스크의 실행 우선도가 가장 높고, 고속·고정밀도로 실행할 수 있는 태스크입니다.
  지정한 주기로 시스템 공통 처리, I/O 리프레시, 사용자 프로그램, 모션 제어가 실행됩니다. 프라이머리 정주기 태스크에는 '모션 제어(MC)' 처리가 있습니다.



*1 : 태스크 주기: 태스크가 정주기로 실행되는 설정된 고정 시간입니다.

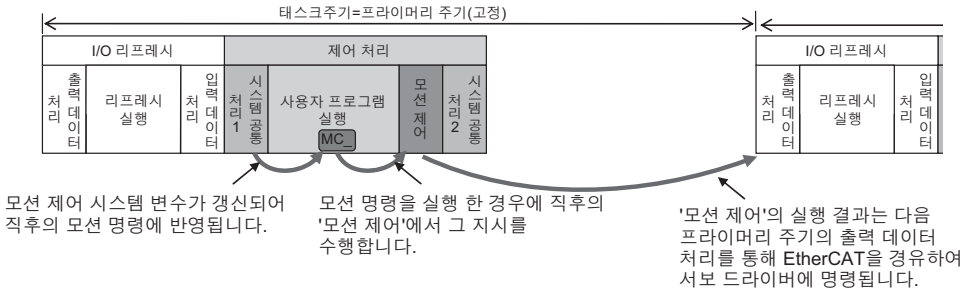
*2 : 태스크 실행 시간: 태스크 실행 조건 성립에서 실행 완료까지 실제로 걸린 시간입니다.

처리		처리 내용
I/O 리프레시	출력 데이터 처리	• I/O 리프레시를 실행하는 I/O(출력)에 대한 OUT 리프레시 데이터를 생성합니다. • 강제값 리프레시가 설정된 경우에는 OUT 리프레시 데이터에 강제값(출력)을 반영합니다.
	리프레시 실행	• I/O와의 데이터 교환을 실행합니다.
	입력 데이터 처리	• I/O 리프레시를 실행한 I/O(입력) 유니트의 IN 리프레시 데이터를 전송합니다. • 강제값 리프레시가 설정된 경우에는 전송한 IN 리프레시 데이터에 강제값(입력)을 반영합니다.
시스템 공통 처리 1		• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 모션 입력 처리*1 • 데이터 트레이스 처리(샘플링, 트리거 판정) • 이벤트 태스크의 조건식 일치 판정
사용자 프로그램 실행		• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
모션 제어*2		• 사용자 프로그램 상의 모션 제어 명령에서의 모션 제어 지시를 실행합니다. • 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에 대한 모션 출력 처리
시스템 공통 처리 2		• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. 모션 제어 시스템 변수의 축 현재값(위치, 속도, 토크) 및 모션 제어 시스템 변수의 서보 드라이버 상태가 갱신됩니다.

*2. 프라이머리 정주기 태스크의 사용자 프로그램 상의 모션 제어 명령인 경우에 다음 그림과 같이 직후의 '모션 제어'에서 그 명령에서의 지시가 실행되고, 다음 프라이머리 정주기 태스크의 'I/O 리프레시'에서 서보 드라이버에 출력됩니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.



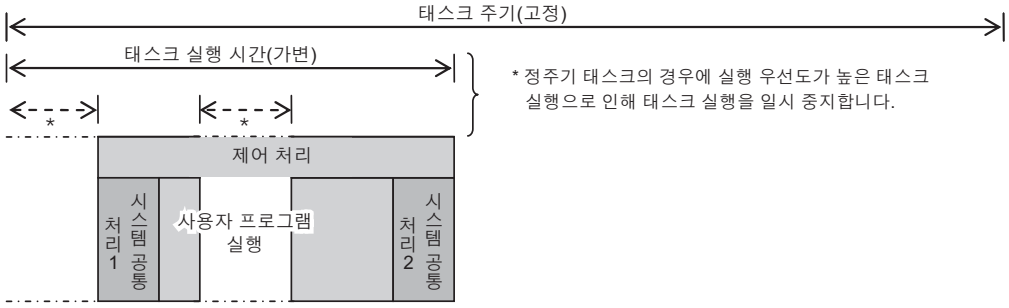
또한, 정주기 태스크의 사용자 프로그램 상의 모션 제어 명령인 경우에는 해당 정주기의 다음 프라이머리 정주기 태스크의 '모션 제어(MC)' 타이밍에 해당 명령에서의 지시가실행됩니다.

자세한 내용은 📖'5-11-3 시스템으로서의 입출력 응답 시간(5-112페이지)'을 참조하십시오.

- **정주기 태스크**

프라이머리 주기의 정수배로 지정된 태스크 주기로 사용자 프로그램을 실행하는 태스크입니다. 태스크 전체에서 0~3개 할당할 수 있습니다.
정주기 태스크(실행 우선도 16)는 I/O 리프레시도 실행할 수 있습니다.
정주기 태스크로 I/O를 제어하지 않는 경우와 제어하는 경우에는 처리 내용이 다음과 같이 다릅니다.

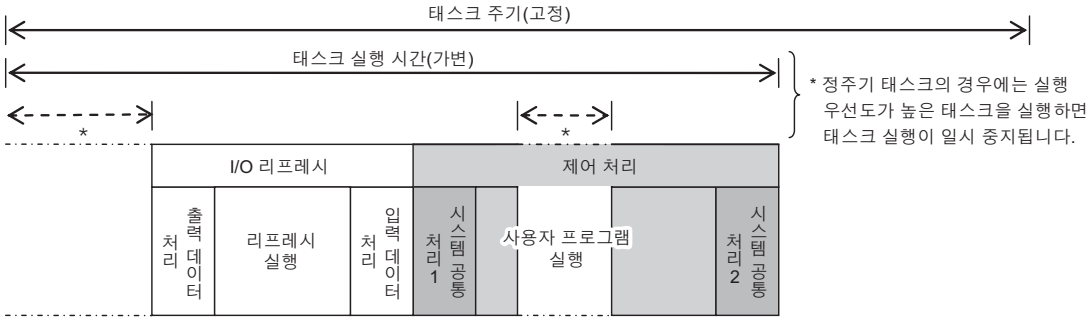
- 정주기 태스크로 I/O를 제어하지 않는 경우



처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.

• 정주기 태스크로 I/O를 제어하는 경우



처리		처리 내용
I/O 리프레시 *1	출력 데이터 처리	• I/O 리프레시를 실행하는 I/O(출력)에 대한 OUT 리프레시 데이터를 생성합니다. • 강제값 리프레시가 설정된 경우에는 OUT 리프레시 데이터에 강제값(출력)을 반영합니다.
	리프레시 실행	• I/O와의 데이터 교환을 실행합니다.
	입력 데이터 처리	• I/O 리프레시를 실행한 I/O(입력) 유니트의 IN 리프레시 데이터를 전송합니다.*2 • 강제값 리프레시가 설정된 경우에는 전송한 IN 리프레시 데이터에 강제값(입력)을 반영합니다.
시스템 공통 처리 1		• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우) • 데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행		• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2		• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우) • 태스크 실행과의 동시성을 확보시키기 위한 컨트롤러 외부에서의 변수 액세스 처리([태스크 설정]에서 설정한 '변수 액세스 시간' 분량) • EtherNet/IP 태그 데이터 링크 처리인 경우에는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. CJ 유니트의 I/O 리프레시를 실행합니다.

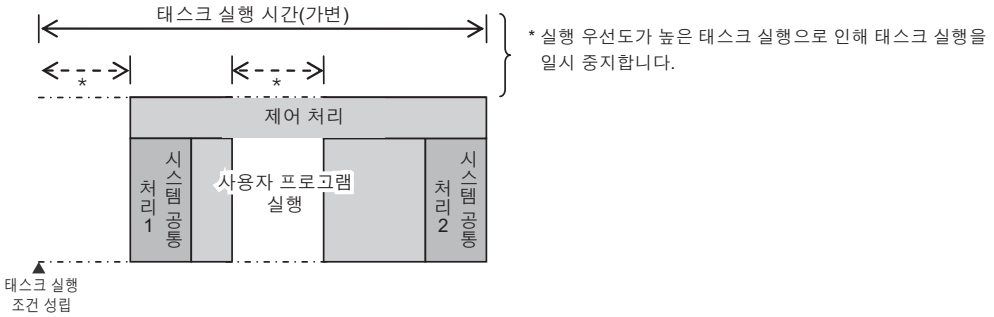
*2. 정주기 태스크(실행 우선도 16)에서 전송한 IN 리프레시 데이터를 NJ 시리즈의 프라이머리 정주기 태스크에서 참조하는 경우에는 참조하는 IN 리프레시 데이터를 정주기 태스크(실행 우선도 16)에서 글로벌 변수에 복사하고, 프라이머리 정주기 태스크에서는 복사한 글로벌 변수 값을 참조하십시오. 글로벌 변수는 변수 값의 동시성을 확보하십시오. 자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90 페이지)'을 참조하십시오.
상기 실행을 하지 않으면 정주기 태스크(실행 우선도 16)에서의 I/O 리프레시 실행 중에 프라이머리 정주기 태스크의 실행이 시작된 경우에 프라이머리 정주기 태스크에서 참조한 IN 리프레시 데이터 값의 동시성이 확보되지 않을 수 있습니다.

(주의) 위 표의 각 셀의 처리는 기재된 순서대로 실행됩니다.

● **이벤트 태스크**

지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 태스크 전체에서 0~32개 할당할 수 있습니다.

이벤트 태스크의 처리 내용은 다음과 같습니다.



처리	처리 내용
시스템 공통 처리 1	• 변수의 태스크 간 배타 제어 처리(참조 태스크인 경우)*1
사용자 프로그램 실행	• 태스크에 할당된 프로그램을 할당순으로 실행합니다.
시스템 공통 처리 2	• 변수의 태스크 간 배타 제어 처리(갱신 태스크인 경우)*1

*1. 변수의 태스크 간 배타 제어 처리의 자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

5-5-4 NJ 시리즈의 이벤트 태스크 실행 조건

이벤트 태스크는 지정한 실행 조건이 성립했을 때 1회만 실행되는 태스크입니다. 이벤트 태스크의 실행 조건에는 다음 2종류가 있습니다.

실행 조건	이벤트 태스크가 실행되는 타이밍	사용하는 경우
ActEventTask 명령에 의한 실행	ActEventTask 명령 실행 시	• 사용자 프로그램 내에서 어느 이벤트 태스크를 실행하는지 명시하고자 하는 경우 • 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정 시까지 변화할 가능성이 있는 경우
변수의 조건식 일치에 의한 실행	지정한 변수의 값이 지정한 조건식에 일치했을 때*1	이벤트 태스크 실행을 위한 사용자 프로그램을 생략해서 사용자 프로그램을 간략화하고자 하는 경우

*1. 지정한 변수의 값이 지정한 조건식에 일치했는지 여부를 판정하는 타이밍에 대해서는 '실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍(5-61페이지)'을 참조하십시오.

안전상의 중요 사항

이벤트 태스크의 실행 조건이 변수의 조건식과 일치하고 조건식에 다음 변수를 지정하는 경우에 조건이 일치할 때 이벤트 태스크가 기동하지 않거나 조건 불일치 시에 이벤트 태스크가 기동되는 경우가 있습니다.

- 모션 제어 시스템 변수를 제외하고, 데이터 크기가 16비트 이상인 구조체 멤버
- 데이터 크기가 16비트 이상의 배열 요소

위의 변수를 지정하여 일치 판정을 실시하는 경우에는 다음 중 하나를 실행하십시오.

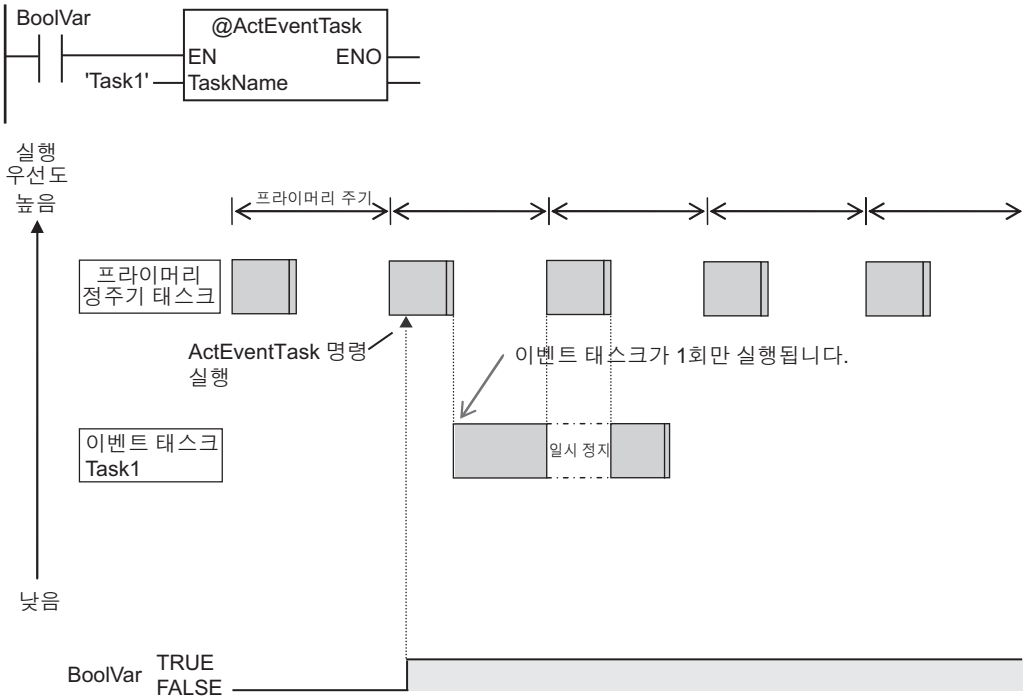
- 64비트 이외의 기본 데이터 형식의 내부 변수에 복사하고, 복사한 내부 변수를 참조함
- 변수의 태스크 간 배타 제어 설정 기능을 사용하고, 프라이머리 정주기 태스크를 갱신 태스크로 함

ActEventTask 명령에 의한 이벤트 태스크 실행

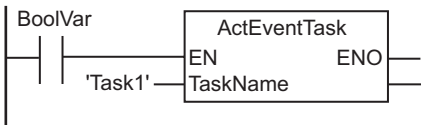
사용자 프로그램 내에서 ActEventTask 명령이 실행되면 지정한 이벤트 태스크가 1회 실행됩니다.
ActEventTask 명령의 자세한 사양은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.
ActEventTask 명령에 의한 방법을 사용하면 사용자 프로그램을 봤을 때, 어느 이벤트 태스크를 기동하는지 잘 알 수 있습니다. 또한, 이벤트 태스크 실행 조건이 변수의 조건식 일치 판정까지 변화할 가능성이 있는 경우에도 유효합니다.

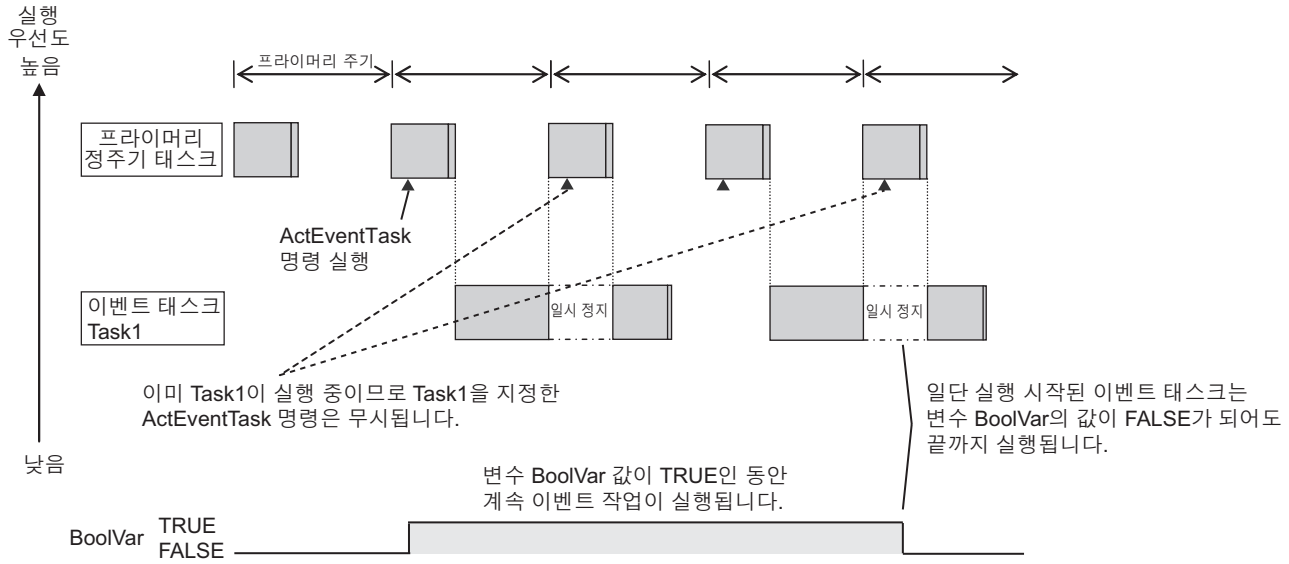
- **ActEventTask 명령을 사용한 사용자 프로그램 예**

  예 1) 변수의 값이 변화했을 때 이벤트 태스크를 1회만 실행하는 경우
  다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하면 BOOL 형식 변수 BoolVar의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크 Task1이 1회만 실행됩니다.



예 2) 변수가 지정한 값인 동안 반복해서 이벤트 태스크를 실행하는 경우
다음 그림과 같이 ActEventTask 명령에 입력 상승 미분의 동작 옵션을 부가하지 않으면 BOOL 형식 변수 BoolVar의 값이 TRUE인 동안 반복해서 이벤트 태스크 Task1이 실행됩니다. 단, Task1이 실행 중에 Task1을 지정한 ActEventTask 명령은 실행해도 무시됩니다. 또한, 일단 실행 시작된 이벤트 태스크는 실행 도중에 BoolVar의 값이 FALSE가 되어도 마지막까지 실행됩니다.





변수의 조건식 일치에 의한 이벤트 태스크 실행

변수의 조건식 일치에 의한 방법은 Sysmac Studio에서 지정한 변수 값이 지정한 조건식에 일치한 경우 이벤트 태스크가 1회 실행된다는 것입니다. 해당 변수의 값이 조건식에 일치하는 동안 이벤트 태스크가 계속 실행는 것이 아니라 변수 값이 조건식에 불일치에서 일치로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.
변수의 조건 일치에 의한 방법에서는 이벤트 태스크 실행을 위한 사용자 프로그램이 필요 없어집니다.

● 조건식을 지정할 수 있는 변수의 종류

조건식을 지정할 수 있는 변수의 종류는 다음과 같습니다.

변수의 종류		지정 가능 여부
시스템 정의 변수		가능*1
준 사용자 정의 변수		가능
사용자 정의 변수	글로벌 변수	가능
	프로그램 내에서 사용하는 변수	가능
	평션 블록 내에서 사용하는 변수	가능*2
	평션 내에서 사용하는 변수	불가능

*1. 다음과 같은 변수는 제외합니다.
EN, ENO, P_Off, P_CY, P_First_RunMode, P_First_Run, P_PRGER
*2. 입출력 변수는 제외합니다.

● **조건식을 지정할 수 있는 변수의 데이터 형식**

조건식을 지정할 수 있는 변수의 데이터 형식은 다음과 같습니다.

데이터 형식의 분류	데이터 형식		지정 가능 여부
기본 데이터 형식	부울 형식, 비트 열 형식, 정수 형식, 실수 형식		가능
	지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		불가능
데이터 형식 지정	배열 지정	배열 전체	불가능
		요소	가능*1
파생 데이터 형식	구조체 형식	구조체 전체	불가능
		멤버	가능*2
	공용체 형식	공용체 전체	불가능
		멤버	가능*2
	열거 형식		가능

*1. 배열 요소의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

*2. 멤버의 데이터 형식이 부울 형식, 비트 열 형식, 정수 형식, 실수 형식 중 하나인 경우로 제한됩니다.

● **지정할 수 있는 조건식**

지정할 수 있는 조건식은 조건식을 지정하는 변수의 데이터 형식에 따라 다음과 같이 정해져 있습니다.

조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다.

데이터 형식	지정할 수 있는 조건식
부울 형식, 부울 형식의 배열 요소, 부울 형식의 구조체 멤버, 부울 형식의 공용체 멤버	상승 FALSE→TRUE
	하강 TRUE→FALSE
비트 열 형식, 정수 형식, 실수 형식 또는 이들을 데이터 형식으로 하는 배열 요소, 구조체 멤버, 공용체 멤버	변수 =｛비교 상수｝
	변수 ≠｛비교 상수｝
	변수 >｛비교 상수｝
	변수≧｛비교 상수｝
	변수 <｛비교 상수｝
	변수≦｛비교 상수｝

● **비교 상수 값의 유효 범위**

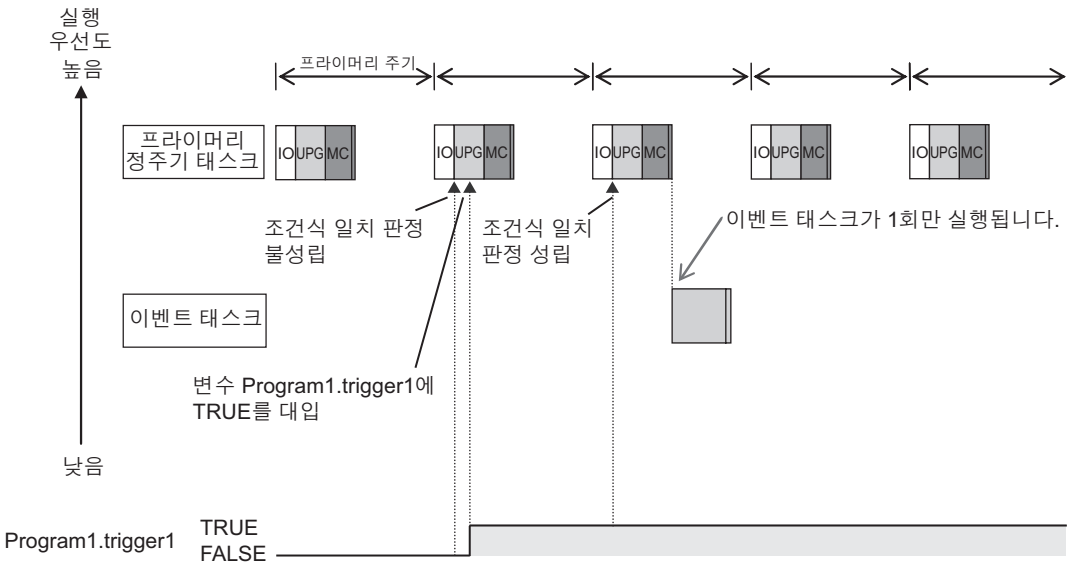
조건식을 지정하는 변수의 데이터 형식이 비트 열 형식, 정수 형식, 실수 형식인 경우에는 변수의 값과 비교하는 비교 상수를 설정해야 합니다. 비교 상수의 값의 유효 범위는 조건식을 지정하는 변수의 데이터 형식 값의 유효 범위와 동일합니다.

각 데이터 형식의 값의 유효 범위는 📖'기본 데이터 형식과 파생 데이터 형식(6-31페이지)'을 참조하십시오.

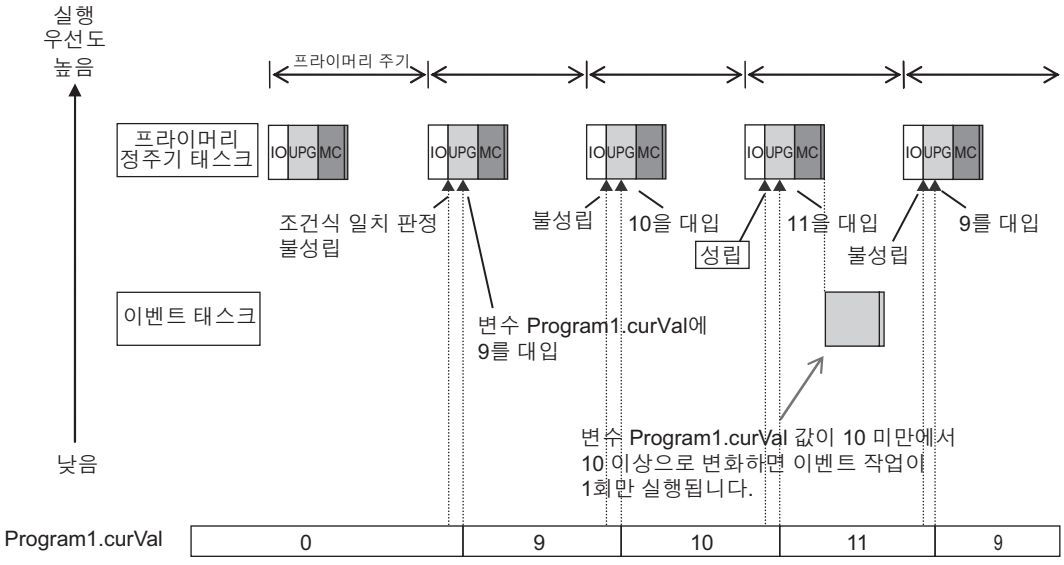
예를 들어 조건식을 지정하는 변수의 데이터 형식이 BYTE인 경우에 비교 상수 값의 유효 범위는 BYTE#16#00~BYTE#16#FF입니다.

● **변수의 조건식 일치에 의한 이벤트 태스크 실행 예**

예 1) 이벤트 태스크 실행 조건을 BOOL 형식 변수 Program1.trigger1의 상승으로 한 경우
Program1.trigger1의 값이 FALSE에서 TRUE로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



예 2) 이벤트 태스크 실행 조건을 INT 형식 변수 Program1.curVal?10로 한 경우
Program1.curVal의 값이 10미만에서 10 이상으로 변화했을 때 이벤트 태스크가 1회만 실행됩니다.



사용상의 주의 사항

조건식 일치 판정이 실행되는 것은 프라이머리 정주기 태스크의 사용자 프로그램 실행보다 이전입니다. 따라서 프라이머리 정주기 태스크의 사용자 프로그램 내에서 지정한 변수의 값이 조건식에 일치해도 해당 판정이 실행되는 것은 그 다음 프라이머리 정주기 태스크입니다. 따라서 이벤트 태스크가 실행되는 것은 변수의 값이 조건식에 일치한 프라이머리 정주기 태스크의 다음 프라이머리 정주기 태스크 완료 후입니다.

5-5-5 NJ 시리즈의 이벤트 태스크 실행 타이밍

이벤트 태스크의 실행 우선도는 8 또는 48입니다. 다른 태스크 실행 중에 이벤트 태스크 실행 조건이 성립한 경우에 실행 우선도가 높은 태스크가 먼저 실행되고, 실행 우선도가 낮은 태스크의 실행은 중단됩니다. 이는 프라이머리 정주기 태스크 및 정주기 태스크와 동일합니다. 단, 이벤트 태스크 실행 조건에 따라 이벤트 태스크 실행 타이밍이 다릅니다.
또한, 이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. 동일한 실행 우선도의 여러 이벤트 태스크 실행 조건이 성립한 경우에는 주의해야 합니다.

이벤트 태스크 실행 조건에 의한 실행 타이밍의 차이

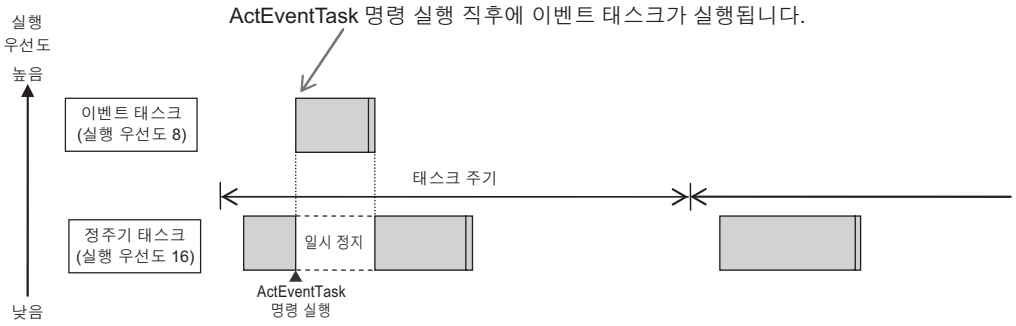
이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우와 변수의 조건식 일치에 의한 경우에는 실행 타이밍이 다릅니다.

- **실행 조건이 ActEventTask 명령에 의한 경우의 실행 타이밍**

  이벤트 태스크 실행 조건이 ActEventTask 명령에 의한 경우에는 ActEventTask 명령의 실행직후에 이벤트 태스크가 실행됩니다.

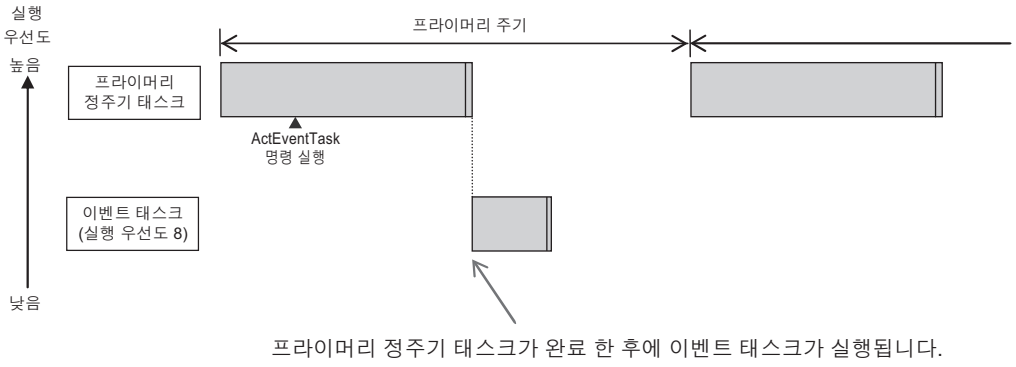
  예 1) ActEventTask 명령을 실행하는 정주기 태스크의 실행 우선도가 16, 이벤트 태스크의 실행 우선도가 8인 경우

  이벤트 태스크(실행 우선도 8) 쪽이 정주기 태스크(실행 우선도 16)보다 실행 우선도가 높으므로 정주기 태스크(실행 우선도 16)의 실행을 중단하고 이벤트 태스크(실행 우선도 8)가 실행됩니다.



  예 2) ActEventTask 명령을 프라이머리 정주기 태스크 내에 실행하는 경우

  이벤트 태스크 쪽이 프라이머리 정주기 태스크보다 실행 우선도가 낮으므로 프라이머리 정주기 태스크가 완료된 후 이벤트 태스크가 실행됩니다.

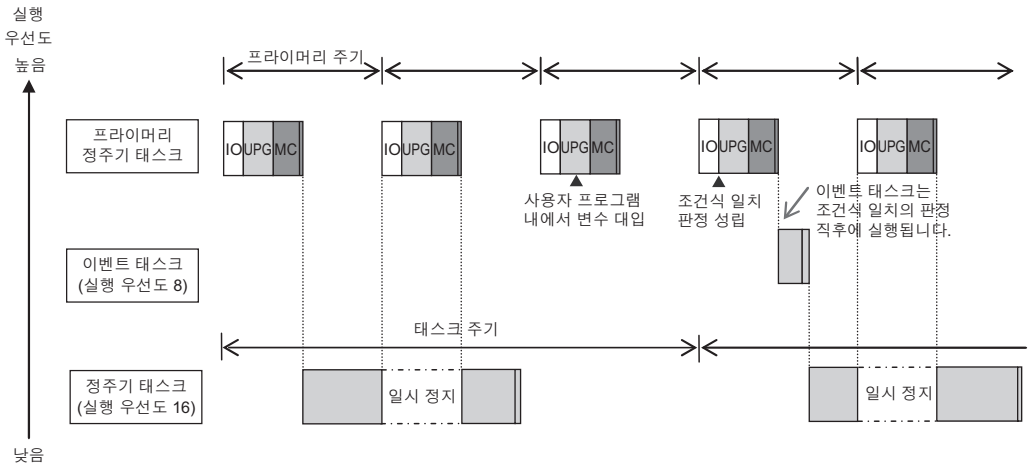


● **실행 조건이 변수의 조건식 일치에 의한 경우의 실행 타이밍**

조건식 일치 판정은 프라이머리 정주기 태스크 중에 실행됩니다. 따라서 이벤트 태스크가 실행되는 것은 지정한 변수의 값이 조건식에 일치한 후, 처음으로 실행되는 프라이머리 정주기 태스크 직후입니다. 단, 이벤트 태스크보다도 실행 우선도가 높은 태스크가 존재하는 경우에는 그 태스크 쪽이 먼저 실행됩니다.

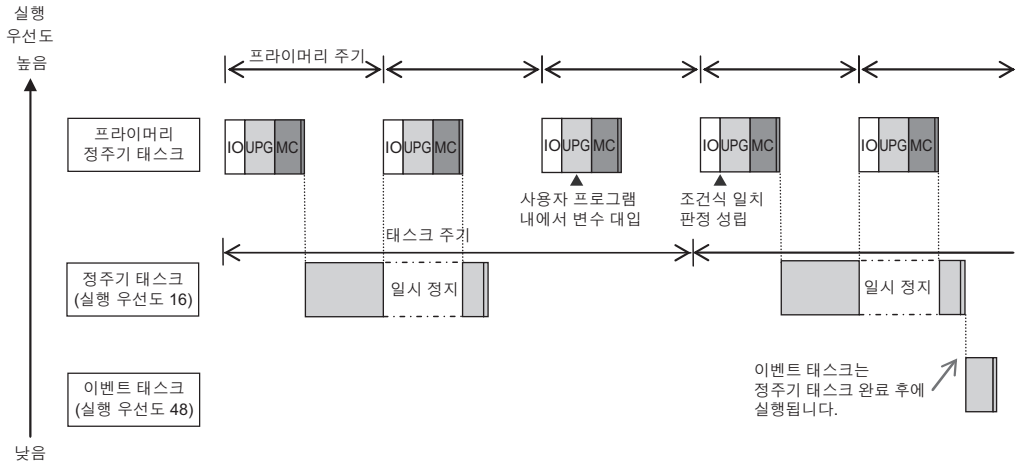
예 1) 정주기 태스크(실행 우선도 16)와 이벤트 태스크(실행 우선도 8)가 존재하는 경우

조건식 일치가 발생한 후, 처음으로 실행되는 프라이머리 정주기 태스크의 직후에 이벤트 태스크가 실행됩니다. 이벤트 태스크(실행 우선도 8) 쪽이 정주기 태스크(실행 우선도 16)보다 실행 우선도가 높으므로 정주기 태스크(실행 우선도 16)는 이벤트 태스크 다음에 실행됩니다.



예 2) 정주기 태스크(실행 우선도 16)와 이벤트 태스크(실행 우선도 48)가 존재하는 경우

이벤트 태스크 쪽이 정주기 태스크(실행 우선도 16)보다 실행 우선도가 낮으므로 정 주기 태스크(실행 우선도 16)가 완료된 후에 이벤트 태스크가 실행됩니다.



사용상의 주의 사항

- 조건식 일치가 프라이머리 정주기 태스크의 I/O 리프레시에 의해 발생하는 경우와 프라이머리 정주기 태스크에 할당된 사용자 프로그램 실행에서 발생하는 경우에는 이벤트 태스크 실행 타이밍이 다음과 같이 다릅니다. 조건식 일치 판정이 실행되는 것이 프라이머리 정주기 태스크 내의 시스템 공통 처리 1뿐이기 때문입니다. 프라이머리 정주기 태스크의 처리는 I/O 리프레시→시스템 공통 처리 1→사용자 프로그램 실행 순서대로 실행됩니다.

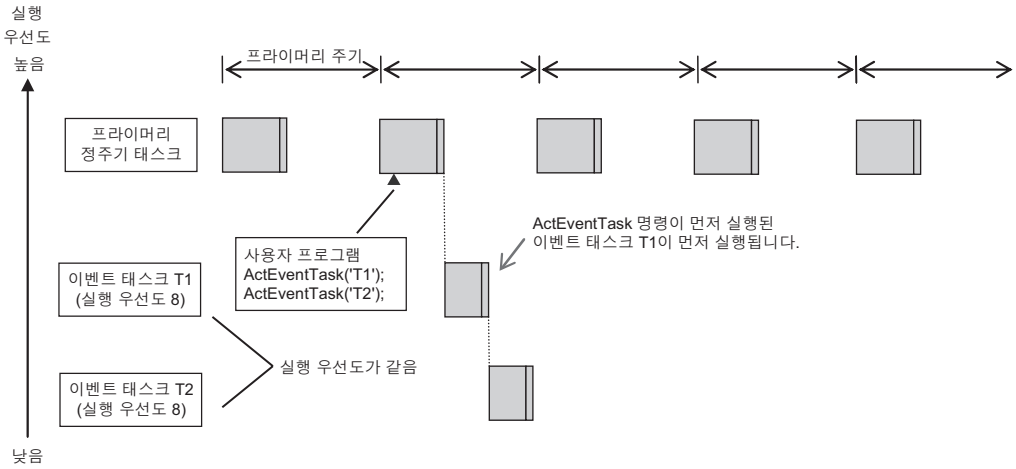
조건식 일치 발생 요인	이벤트 태스크 실행 타이밍
프라이머리 정주기 태스크의 I/O 리프레시	해당 프라이머리 정주기 태스크 완료 후
프라이머리 정주기 태스크의 사용자 프로그램 실행	다음 프라이머리 정주기 태스크 완료 후

- 이벤트 태스크가 실행되는 것은 1회 이전 판정 시 조건식 불일치이고, 다음 판정 시 조건식 일치가 발생한 경우입니다. 따라서 불일치에서 일치로 변화가 있었다고 해도 다음 판정 시까지 불일치로 되돌아간 경우에는 이벤트 태스크는 실행되지 않습니다.
- NJ 시리즈에서 데이터 크기가 64비트 이상의 내부 변수를 조건식으로 지정하는 경우에 프라이머리 정주기 태스크에 할당된 프로그램에서 정의된 내부 변수를 지정하십시오. 프라이머리 정주기 태스크 이외에 정의된 내부 변수를 지정하면 변수 값의 동시성이 확보되지 않는 경우가 있기 때문에 일치 판정이 올바르게 실행되지 않을 수 있습니다.

동일한 실행 우선도의 이벤트 태스크 실행 타이밍

이벤트 태스크는 복수의 태스크에 동일한 실행 우선도를 할당할 수 있습니다. 동일한 실행 우선도의 여러 이벤트 태스크 실행 조건이 성립한 경우에는 실행 조건이 성립한 순서대로 실행됩니다.

예 1) 2가지 ActEventTask 명령이 실행된 경우
다음 그림과 같이 먼저 실행된 ActEventTask 명령에 의해 지정된 이벤트 태스크 T1이 이벤트 태스크 T2보다도 먼저 실행됩니다.



예 2) 변수의 조건식 일치와 ActEventTask 명령이 혼재하는 경우

이벤트 태스크 T1, T2, T3 실행 조건이 다음과 같이 설정되었다고 가정합니다.

- T1: 변수 VarA의 조건식 일치
- T2: 변수 VarB의 조건식 일치
- T3: ActEventTask 명령

다음 그림과 같이 동일한 프라이머리 주기 중에 I/O 리프레시에 의한 VarA의 조건식 일치, 사용자 프로그램에서의 ActEventTask 명령 실행, 사용자 프로그램에서의 VarB의 조건식 일치가 발생한 경우에는 다음 순서로 동작합니다.

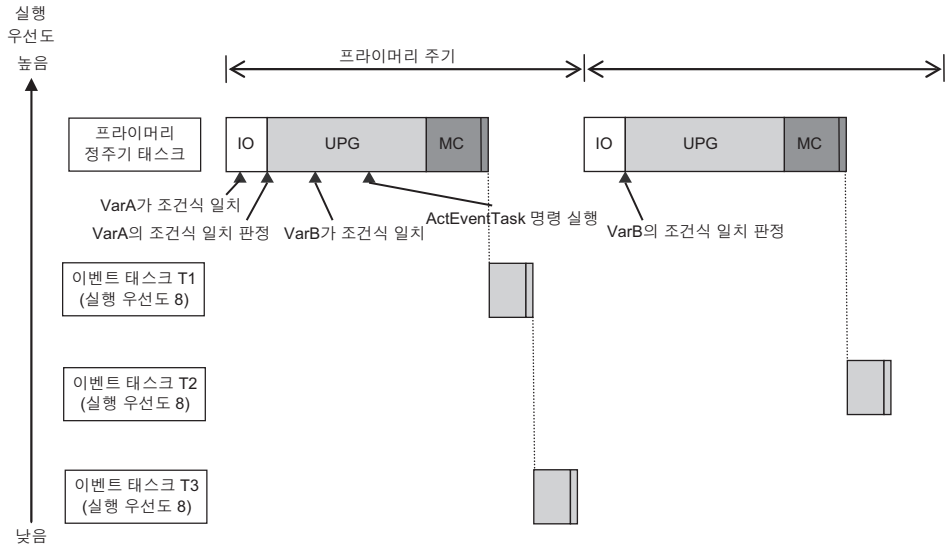
(1) I/O 리프레시에서 VarA가 조건식에 일치합니다.

(2) 시스템 공통 처리 1에서 VarA가 조건식에 일치한다고 판정되고, T10이 실행됩니다. VarB는 조건식에 일치하지 않으므로 T2는 실행되지 않습니다.

(3) 사용자 프로그램에서 VarB가 조건식에 일치합니다.

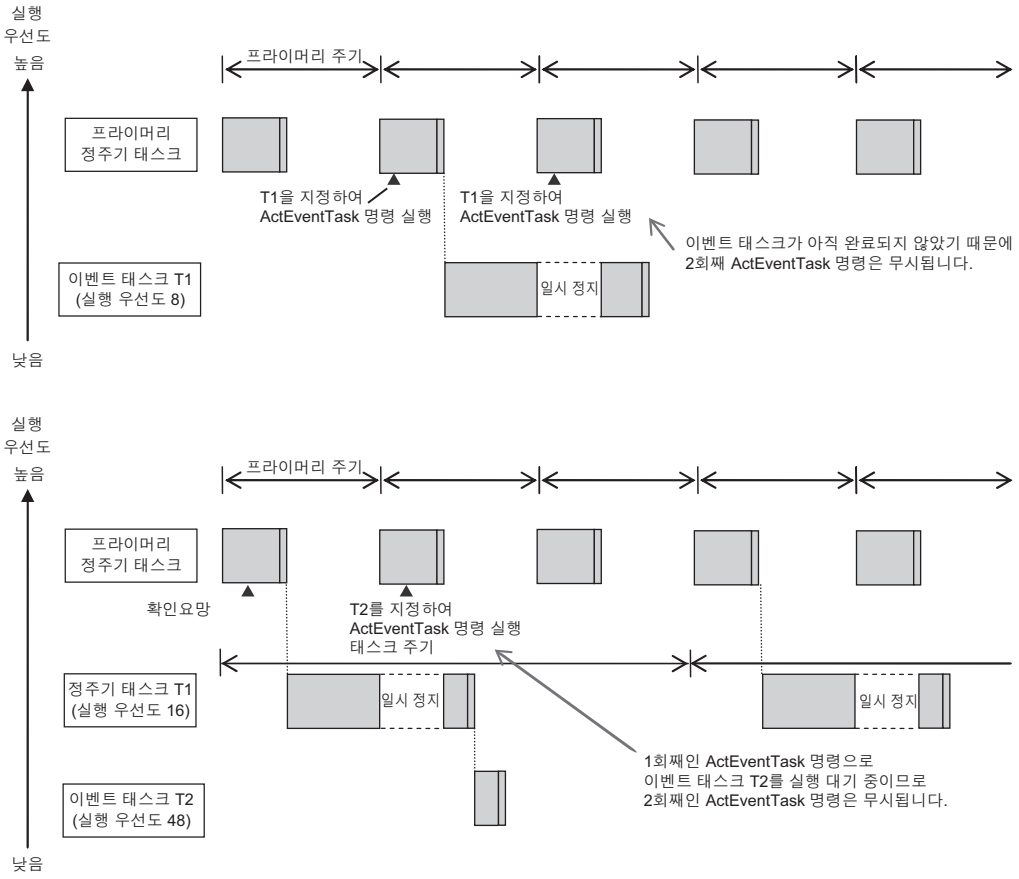
(4) 사용자 프로그램으로 ActEventTask 명령이 실행되고, T30이 실행됩니다.

(5) 다음 프라이머리 주기의 시스템 공통 처리 1에서 VarB가 조건식에 일치한다고 판정되고, T2가 실행됩니다.



5-5-6 이벤트 태스크 완료 전에 다시 실행 조건이 성립한 경우의 동작

이벤트 태스크 완료 전에 다시 해당 이벤트 태스크 실행 조건이 성립한 경우에는 2번째 실행 조건 성립은 무시됩니다. 이벤트 태스크 완료 전은 이벤트 태스크 실행 중 및 이벤트 태스크 실행 대기 중을 모두 포함합니다. 또한, 일단 실행이 시작된 이벤트 태스크는 실행 조건이 성립하지 않아도 마지막까지 실행됩니다.



5-6 태스크 이외의 서비스

CPU 유니트는 프라이머리 정주기 태스크, 정주기 태스크, 이벤트 태스크 이외에도 처리를 실행합니다. NX701 CPU 유니트, NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트별로 태스크 이외의 서비스에 대해 설명합니다.

- **NX701 CPU 유니트의 경우**

  NX701 CPU 유니트에서는 태그 데이터 링크 서비스, 시스템 서비스 2종류가 있습니다. 이 2가지 처리 내용, 실행 우선도, 실행 타이밍은 다음과 같습니다.

항목	태그 데이터 링크 서비스	시스템 서비스
실행 우선도	태스크 실행 및 시스템 서비스와 병렬로 실행됩니다.	태스크 실행 및 태그 데이터 링크 서비스와 병렬로 실행됩니다.
처리 내용	EtherNet/IP 네트워크 상에서 다른 컨트롤러 및 장치와 태그를 사용하여 데이터 교환을 실행합니다. CPU 유니트의 내장 EtherNet/IP 포트를 사용합니다. *1	USB 포트 서비스 및 SD 메모리 서비스 등 실행 우선도가 낮은 처리의 총칭입니다.
실행 타이밍	실행 간격 및 1회 처리 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동합니다.*2	하드웨어 및 CPU 유니트 외부에서의 요구가 있었던 경우 언제라도 실행됩니다.

*1. 태그 데이터 링크의 처리 내용의 자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.
*2. 태그 데이터 링크의 실행 간격과 실행 시간의 자세한 내용은 '5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍(5-70페이지)'을 참조하십시오.

- **NX102 CPU 유니트의 경우**

  NX102 CPU 유니트에서는 태그 데이터 링크 서비스, 시스템 서비스의 2종류가 있습니다. 이 2가지 처리 내용, 실행 우선도, 실행 타이밍은 다음과 같습니다.

항목	태그 데이터 링크 서비스	시스템 서비스
실행 우선도	태스크 실행 및 시스템 서비스와 병렬로 실행됩니다. 시스템 서비스보다 우선됩니다.	태스크의 실행과 병렬로 실행됩니다. 단, 태그 데이터 링크 서비스 실행 중에는 실행되지 않습니다.
처리 내용	EtherNet/IP 네트워크 상에서 다른 컨트롤러 및 장치와 태그를 사용하여 데이터 교환을 실행합니다. CPU 유니트의 내장 EtherNet/IP 포트를 사용합니다. *1	SD 메모리 서비스 등 실행 우선도가 낮은 처리의 총칭입니다.
실행 타이밍	실행 간격 및 1회 처리 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동합니다.*2	하드웨어 및 CPU 유니트 외부에서 요구가 있었던 경우에 실행 우선도에 따라 실행됩니다.

*1. 태그 데이터 링크의 처리 내용의 자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.
*2. 태그 데이터 링크의 실행 간격과 실행 시간의 자세한 내용은 '5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍(5-70페이지)'을 참조하십시오.

● NX1P2 CPU 유니트의 경우

NX1P2 CPU 유니트에서는 태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스의 3종류가 있습니다. 이 3가지 처리 내용, 실행 우선도, 실행 타이밍은 다음과 같습니다.

항목	태그 데이터 링크 서비스	옵션 보드 서비스	시스템 서비스
실행 우선도	태스크의 실행과 병렬로 실행됩니다. 옵션 보드 서비스, 시스템 서비스보다 우선됩니다.	태스크의 실행과 병렬로 실행됩니다. 단, 태그 데이터 링크 서비스 실행 중에는 실행되지 않습니다. 시스템 서비스보다 우선됩니다.	태스크의 실행과 병렬로 실행됩니다. 단, 태그 데이터 링크 서비스 및 옵션 보드 서비스의 실행 중에는 실행되지 않습니다.
처리 내용	EtherNet/IP 네트워크 상에서 다른 컨트롤러 및 장치와 태그를 사용하여 데이터 교환을 실행합니다. CPU 유니트의 내장 EtherNet/IP 포트를 사용합니다. *1	아날로그 입출력용 옵션 보드의 I/O 리프레시가 실행됩니다.	SD 메모리 서비스 등 실행 우선도가 낮은 처리의 총칭입니다.
실행 타이밍	실행 간격 및 1회 처리 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동합니다.*2	옵션 보드에서 요구가 있는 경우에 실행 우선도에 따라 실행됩니다.	하드웨어 및 CPU 유니트 외부에서 요구가 있었던 경우에 실행 우선도에 따라 실행됩니다.

*1. 태그 데이터 링크의 처리 내용의 자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.
*2. 태그 데이터 링크의 실행 간격과 실행 시간의 자세한 내용은 📖'5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍(5-70페이지)'을 참조하십시오.

● NJ 시리즈CPU 유니트의 경우

NJ 시리즈 CPU 유니트에서는 태그 데이터 링크 서비스, 시스템 서비스의 2종류가 있습니다. 이 2가지 처리 내용, 실행 우선도, 실행 타이밍은 다음과 같습니다.

항목	태그 데이터 링크 서비스	시스템 서비스
실행 우선도	정주기 태스크(실행 우선도 16)와 정주기 태스크(실행 우선도 17) 사이입니다.	모든 태스크보다 실행 우선도가 낮습니다.
처리 내용	EtherNet/IP 네트워크 상에서 다른 컨트롤러 및 장치와 태그를 사용하여 데이터 교환을 실행합니다. CPU 유니트의 내장 EtherNet/IP 포트 및 CJ 시리즈 EtherNet/IP 유니트 CJ1W-EIP21을 사용합니다. *1	USB 포트 서비스 및 SD 메모리 서비스 등 실행 우선도가 낮은 처리의 총칭입니다.
실행 타이밍	실행 간격 및 1회 처리 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동합니다.*2	태스크 및 태그 데이터 링크 서비스의 비는 시간에 실행됩니다.*3

*1. 태그 데이터 링크의 처리 내용의 자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)" 또는 📖"CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 연결편(SBCD-355)"을 참조하십시오. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*2. 태그 데이터 링크의 실행 간격과 실행 시간의 자세한 내용은 📖'5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍(5-70페이지)'을 참조하십시오.
*3. 단, NJ 시리즈 CPU 유니트에서 시스템 서비스 실행 시간이 충분히 확보되지 않는 경우에는 실행 우선도가 17 이상의 태스크 처리를 일시 정지하고 시스템 서비스가 실행됩니다. 시스템 서비스의 실행을 위한 시간은 Sysmac Studio의 '시스템 서비스 감시 설정'에서 설정합니다.

5-6-1 태스크 이외의 서비스 실행 우선도와 실행 순서

여기에서는 태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스의 실행 우선도와 실행 순서의 예를 설명합니다.

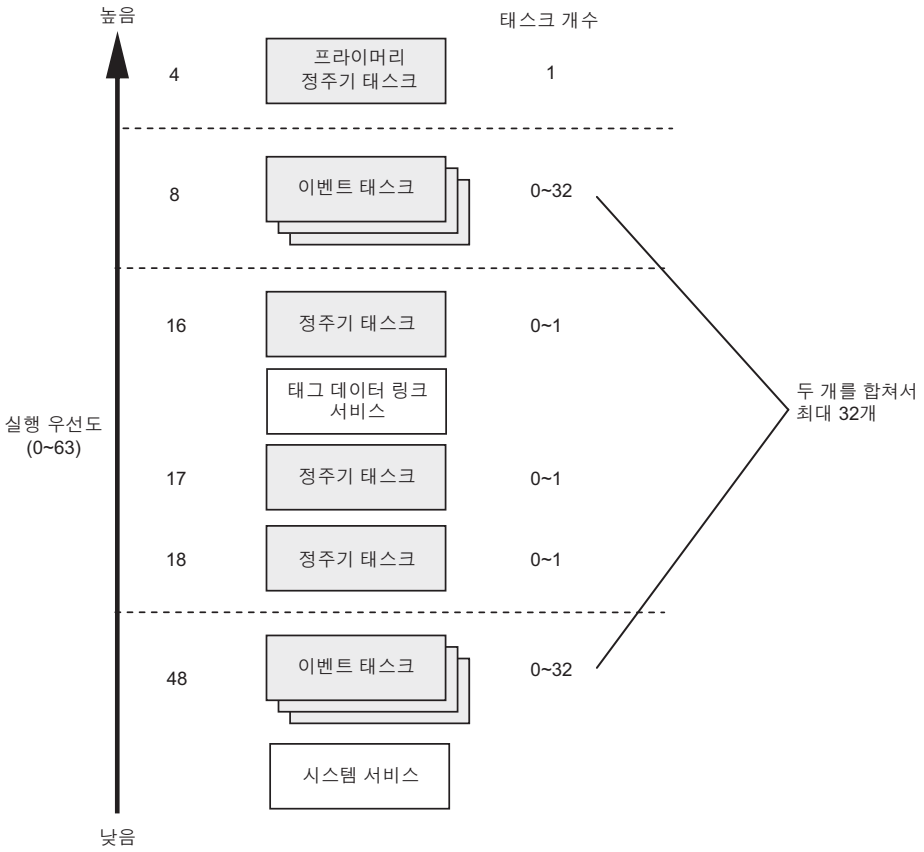
태스크 이외의 서비스의 실행 우선도

- **NX701 CPU 유니트의 경우**
  태그 데이터 링크 서비스는 태스크의 실행 우선도 및 시스템 서비스의 영향을 받지 않습니다.
  또한, 시스템 서비스는 태스크의 실행 우선도 및 태그 데이터 링크 서비스의 영향을 받지 않습니다.

- **NX102 CPU 유니트의 경우**
  태그 데이터 링크 서비스 및 시스템 서비스는 태스크의 실행 우선도의 영향을 받지 않습니다. 실행 우선도가 높은 순서대로 태그 데이터 링크 서비스, 시스템 서비스가 됩니다.

- **NX1P2 CPU 유니트의 경우**
  태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스는 태스크의 실행 우선도의 영향을 받지 않습니다. 실행 우선도가 높은 순서대로 태그 데이터 링크 서비스, 옵션 보드 서비스, 시스템 서비스가 됩니다.

- **NJ 시리즈CPU 유니트의 경우**
  태그 데이터 링크 서비스와 시스템 서비스의 실행 우선도는 다음 그림과 같습니다.
  태그 데이터 링크 서비스의 실행 우선도는 정주기 태스크(실행 우선도 16)와 정주기 태스크(실행 우선도 17) 사이입니다. 시스템 서비스의 실행 우선도는 모든 태스크보다도 낮습니다.

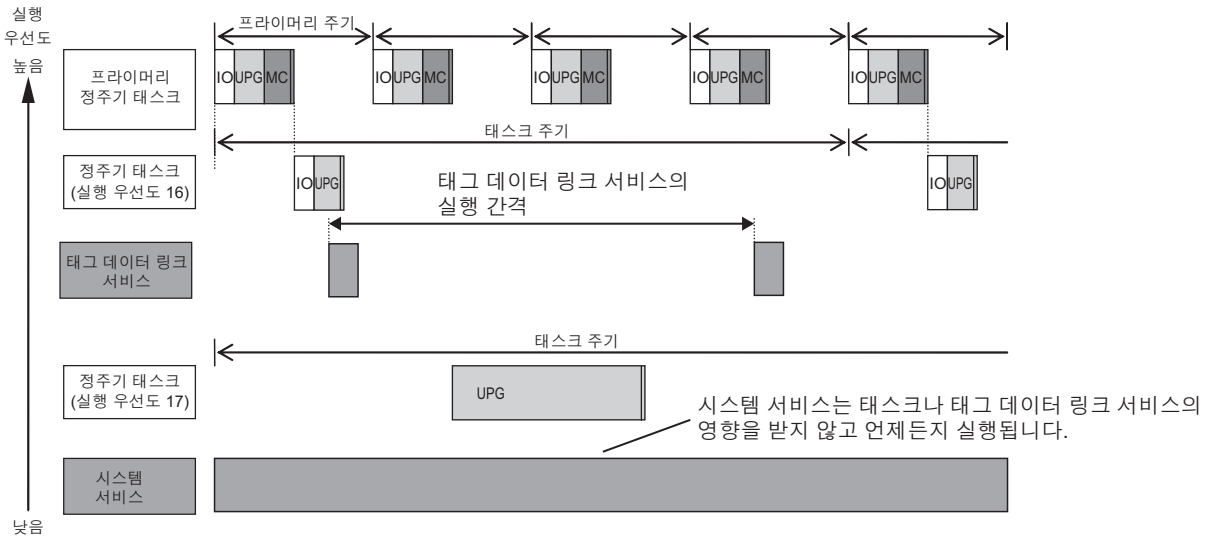


태스크 이외의 서비스 실행 순서의 예

프라이머리 정주기 태스크, 정주기 태스크(실행 우선도 16), 정주기 태스크(실행 우선도 17), 태그 데이터 링크 서비스, 옵션 보드 서비스, 시스템 서비스가 혼재하는 경우의 각각의 실행 순서 예는 다음과 같습니다.

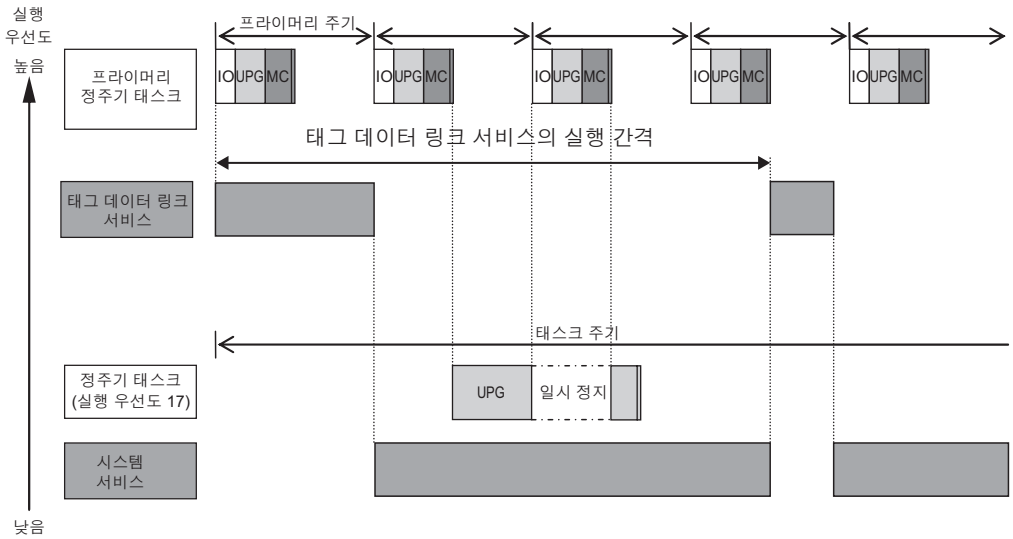
● NX701 CPU 유니트의 경우

- 태그 데이터 링크 서비스는 태스크 및 시스템 서비스의 영향을 받지 않고 실행됩니다.
- 시스템 서비스는 태스크 및 태그 데이터 링크 서비스의 영향을 받지 않고, 언제라도 실행됩니다.



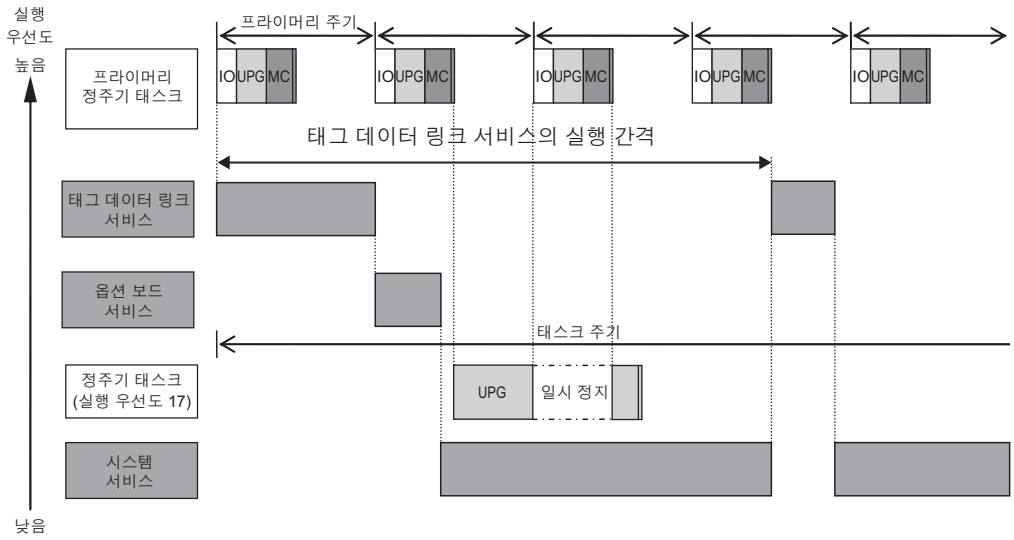
● NX102 CPU 유니트의 경우

- 태그 데이터 링크 서비스 및 시스템 서비스는 태스크의 실행과 병렬로 실행할 수 있습니다.
- 실행 우선도는 태그 데이터 링크 서비스, 시스템 서비스 순서입니다.



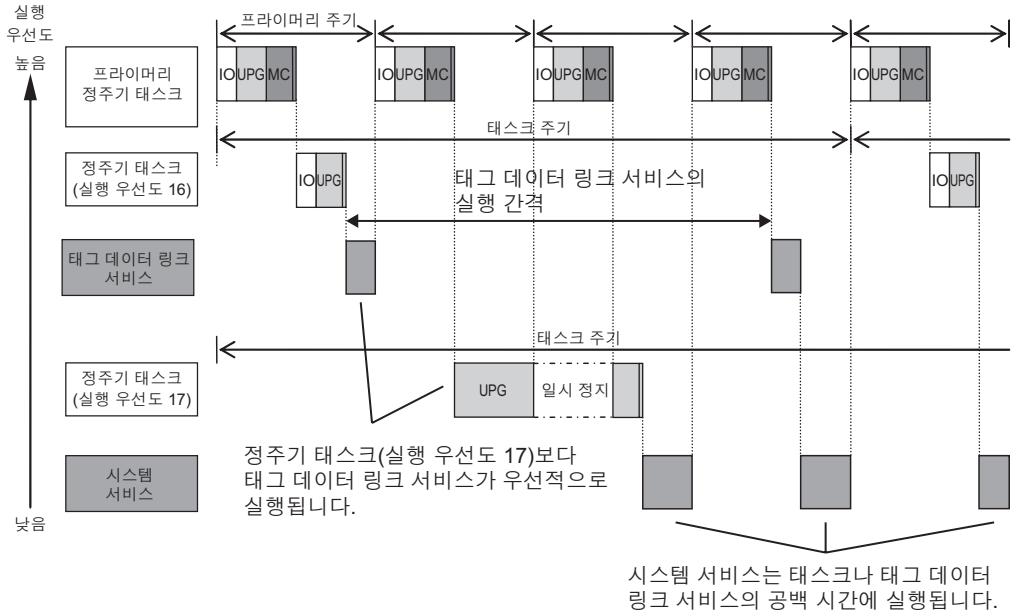
● NX1P2 CPU 유니트의 경우

- 태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스는 태스크 실행과 병렬로 실행할 수 있습니다.
- 실행 우선도는 태그 데이터 링크 서비스, 옵션 보드 서비스, 시스템 서비스 순서입니다.



● NJ 시리즈 CPU 유니트의 경우

- 태그 데이터 링크 서비스는 정주기 태스크(실행 우선도 17)보다도 우선적으로 실행됩니다. 그러나 그보다도 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16) 쪽이 우선적으로 실행됩니다.
- 시스템 서비스는 모든 태스크 및 태그 데이터 링크 서비스의 비는 시간에 실행됩니다.



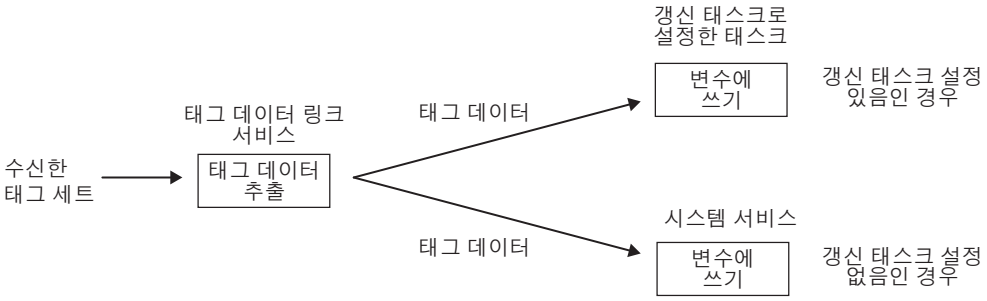
태그 데이터 링크 서비스의 실행 간격 및 1회의 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동됩니다. 자세한 내용은 '5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍 (5-70페이지)'을 참조하십시오.

5-6-2 태그 데이터 링크 서비스의 처리 내용과 실행 타이밍

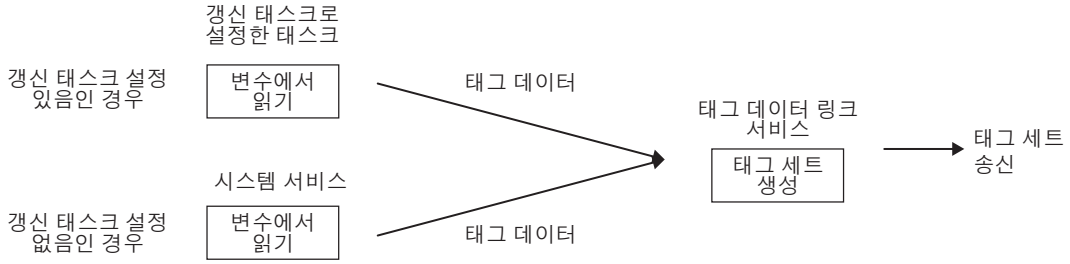
태그 데이터 링크 서비스의 처리 내용

태그 데이터 링크 처리는 여러 처리로 세분화되어 있습니다. 이들의 처리는 태그 데이터 링크 서비스, 시스템 서비스, 태스크에 의해 실행됩니다. CPU 유니트 내장 EtherNet/IP 포트를 사용하여 태그 데이터 링크 처리를 실시하는 경우의 예는 다음과 같습니다.

- **태그 데이터 수신 처리의 흐름**
  - 태그 데이터 링크 서비스에 따라 수신한 태그 세트에서 태그 데이터가 도출됩니다.
  - 갱신 태스크를 설정한 경우에 그 태스크에 의해 태그 데이터의 값이 태그에 할당한 변수에 기입됩니다.
  - 갱신 태스크를 설정하지 않은 경우에 시스템 서비스에 따라 태그 데이터의 값이 태그에 할당한 변수에 기입됩니다.



- **태그 데이터 송신 처리의 흐름**
  - 갱신 태스크를 설정한 경우에 그 태스크에 의해 태그 데이터의 값이 태그에 할당한 변수에서 읽어옵니다.
  - 갱신 태스크를 설정하지 않은 경우에 시스템 서비스에 따라 태그 데이터의 값이 태그에 할당한 변수에서 읽어옵니다.
  - 그 후, 태그 데이터 링크 서비스에 의해 태그 데이터에서 태그 세트가 생성되고 송신됩니다.



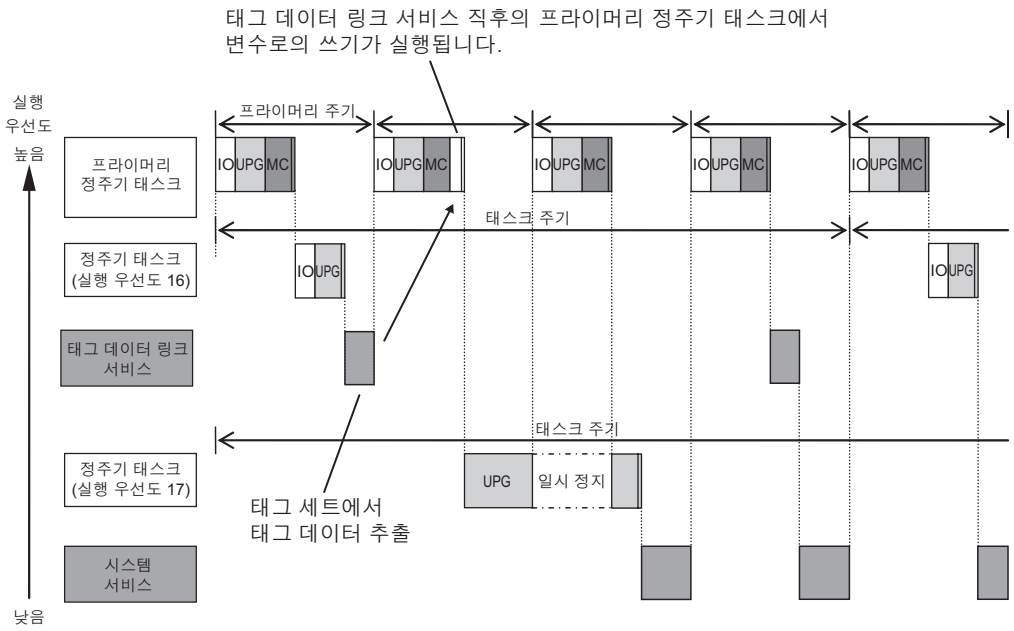
참고

갱신 태스크 설정이 있는 경우와 없는 경우의 태그 데이터 링크 처리의 타이밍의 차이

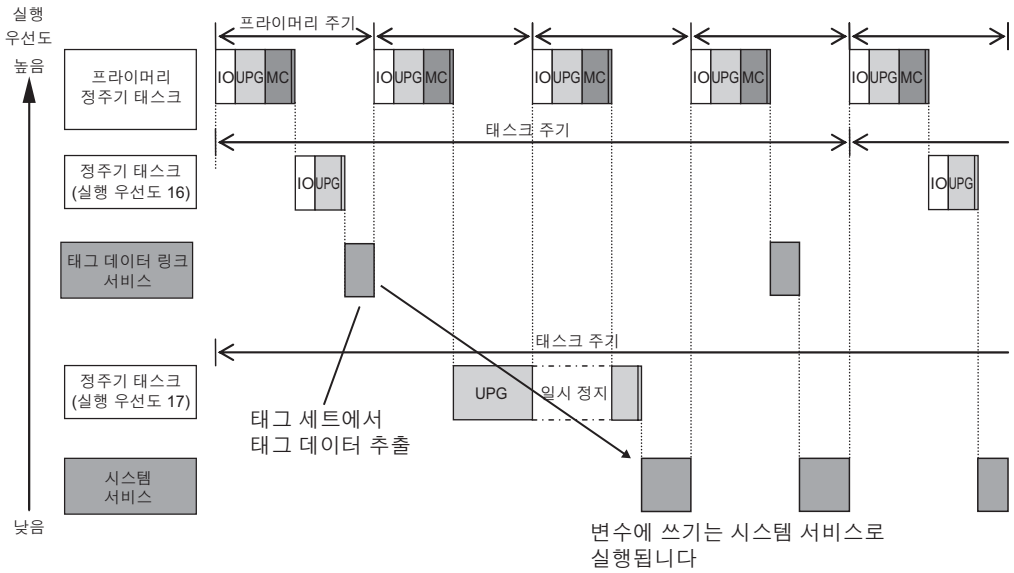
갱신 태스크 설정이 있는 경우와 없는 경우에는 변수에 값을 읽고 쓰는 처리가 다릅니다. 갱신 태스크 설정이 있는 경우에는 해당 태스크, 없는 경우에는 시스템 서비스에 의해 값의 읽기/쓰기가 실행됩니다. 따라서 태그 데이터 링크 처리의 실행 타이밍에도 차이가 발생합니다.

다음은 NJ 시리즈 CPU 유니트에서의 태그 데이터 링크 수신인 경우의 실행 타이밍의 차이입니다.

- 갱신 태스크 설정이 있는 경우
  프라이머리 정주기 태스크를 갱신 태스크로 설정한 경우의 예입니다. 변수에 대한 쓰기는 태그 데이터 링크 서비스 직후의 프라이머리 정주기 태스크로 실행됩니다.

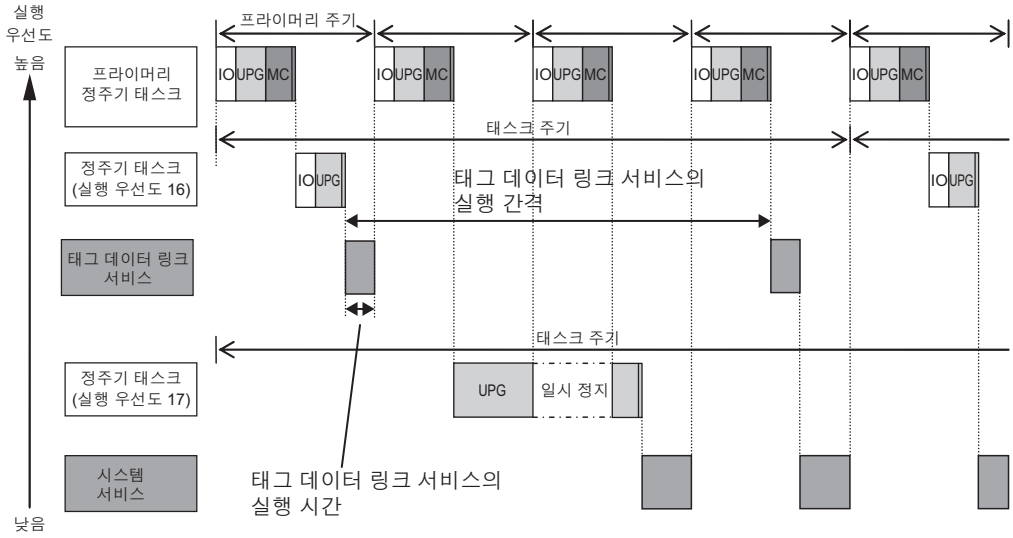


- 갱신 태스크 설정이 없는 경우
  변수에 대한 쓰기는 시스템 서비스로 실행됩니다.



태그 데이터 링크 서비스의 실행 타이밍

태그 데이터 링크 서비스의 실행 간격과 실행 시간은 CPU 유니트의 형식 및 태그 데이터 링크 설정에 따라 변동됩니다. 기준은 다음 표와 같습니다. 단, NJ 시리즈 CPU 유니트인 경우에 태그 데이터 링크 서비스보다도 실행 우선도가 높은 태스크가 실행된 경우에는 실행 간격도 실행 시간도 다음 값보다 길어질 가능성이 있습니다.



- **CPU 유니트 내장 EtherNet/IP 포트를 사용하는 경우**

CPU 유니트	태그 데이터 링크 설정			실행 간격 [ms]	실행 시간 [ms]
	태그 수	연결 수	RPI[ms]		
NJ501-□□□□	8	1	1	1	0.02~0.04
		4	2	1	0.02~0.10
		32	50	1	0.02~0.10
NJ301-□□□□	8	1	1	1	0.03~0.06
		4	2	1	0.03~0.15
		32	50	1	0.03~0.15
NJ101-□□□□	8	1	1	1	0.04~0.09
		4	2	1	0.04~0.28
		32	50	1	0.04~0.28

참고

NX 시리즈 CPU 유니트인 경우에 태그 데이터 링크 서비스의 실행은 정주기 태스크(실행 우선도 17, 18) 및 시스템 서비스에 영향을 주지 않습니다.

- CJ **시리즈** EtherNet/IP **유니트** CJ1W-EIP21**을 사용하는 경우**

CPU 유니트	태그 데이터 링크 설정				실행 간격 [ms]	실행 시간 [ms]
	태그 수	연결 수	RPI[ms]	송신/수신		
NJ501-□□□□	8	1	1	송신	3	0.10
				수신	3	0.10
		4	2	송신	4	0.22
				수신	4	0.13
		32	50	송신	12	0.68
				수신	6~35	0.23~0.66
NJ301-□□□□	8	1	1	송신	3	0.16
				수신	3	0.16
		4	2	송신	3	0.21
				수신	4	0.33
		32	50	송신	15	1.05
				수신	8~35	0.29~1.85
NJ101-□□□□	8	1	1	송신	3	0.21
				수신	3	0.21
		4	2	송신	6	0.46
				수신	13	0.80
		32	50	송신	12~50	2.97~4.09
				수신	25	2.36

버전 관련 정보

태그 데이터 링크 처리가 실행되는 처리는 CPU 유니트의 유니트 버전과 Sysmac Studio 버전의 조합에 따라 다음과 같이 다릅니다.

CPU 유니트의 유니트 버전	Sysmac Studio의 버전	
	Ver.1.03 이하	Ver.1.04 이상
Ver.1.03 이상	시스템 서비스 내의 내장 EtherNet/IP 포트 서비스	태그 데이터 링크 서비스*1
Ver.1.02 이하	시스템 서비스 내의 내장 EtherNet/IP 포트 서비스	

*1. Sysmac Studio에서 설정 유니트 버전을 Ver.1.02 이하로 설정한 경우에 태그 데이터 링크 처리는 시스템 서비스 내의 내장 EtherNet/IP 포트 서비스로 실행됩니다.

5-6-3 옵션 보드 서비스의 처리 내용과 실행 타이밍

옵션 보드 서비스의 처리 내용

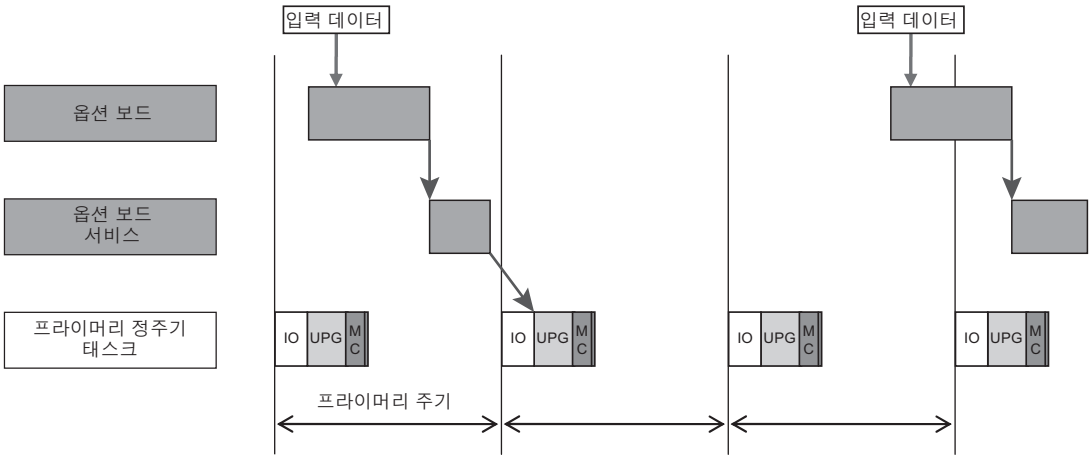
NX1P2 CPU 유니트의 옵션 보드 서비스의 처리에서는 아날로그 입출력용 옵션 보드의 I/O 리프레시가 실행됩니다.

옵션 보드 서비스의 실행 타이밍

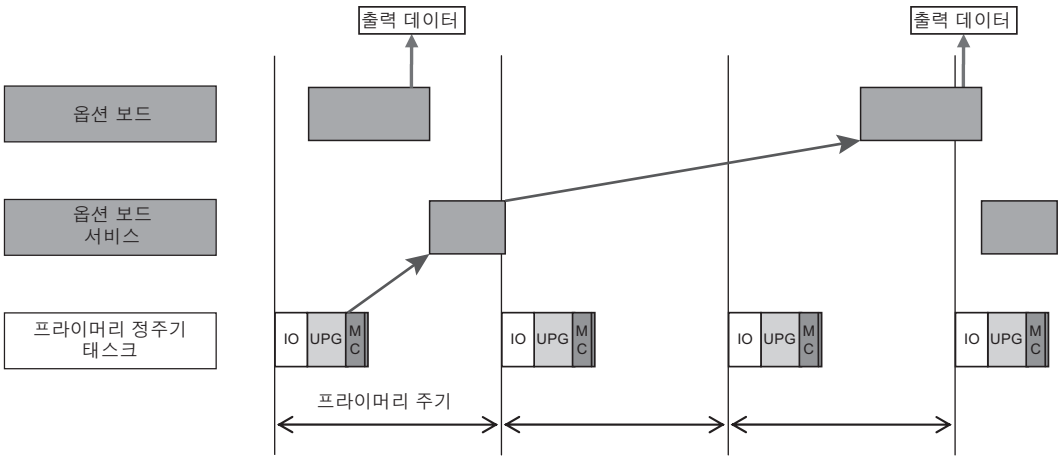
NX1P2 CPU 유니트의 옵션 보드 서비스는 태스크의 영향을 받지 않고 실행됩니다. 단, 태그 데이터 링크 서비스, 옵션 보드 서비스 및 시스템 서비스는 병렬로는 실행되지 않습니다.

옵션 보드 서비스는 옵션 보드에서 요구가 있는 경우에 실행 우선도에 따라 실행됩니다. 옵션 보드 내부의 처리, NX1P2 CPU 유니트의 옵션 보드 서비스 및 태스크 실행은 각각 비동기로 실행됩니다.

데이터를 입력하는 경우에 옵션 보드 내부 처리에서 입력 데이터가 전송되고, NX1P2 CPU 유니트의 옵션 보드 서비스로 처리됩니다. 옵션 보드 서비스에서 처리가 완료되면 다음 프라이머리 정주기 태스크의 사용자 프로그램 실행에서 입력 데이터를 이용할 수 있습니다.



데이터를 출력하는 경우에 사용자 프로그램 실행이 완료한 후의 옵션 보드 서비스에서 출력 데이터가 처리됩니다. 옵션 보드 서비스에서 처리가 완료되면 다음 옵션 보드 내부의 처리에서 출력 데이터가 출력됩니다.



5-6-4 시스템 서비스의 처리 내용과 실행 타이밍

시스템 서비스의 처리 내용

시스템 서비스의 처리에는 다음과 같은 내용이 있습니다.

시스템 서비스의 처리명	내용
USB 포트 서비스*1	• 주변 툴, 상위 PC에서의 서비스 요구에 대한 처리
내장 EtherNet/IP 포트 서비스	• 주변 툴, 표시기, 상위 PC 또는 다른 컨트롤러 등에서의 서비스 요구(CIP 등)에 대한 처리 • 통신 명령(CIP, 소켓)의 실행
내장 EtherCAT 포트 서비스	• EtherCAT 메시지 통신의 실행
시리얼 통신용 옵션 보드의 통신 처리*2	• 시리얼 통신용 옵션 보드의 통신 처리 실행
CJ 고기능 유니트 서비스*3	• CJ 고기능 유니트와의 이벤트 서비스 실행 • 통신 명령(CIP)의 실행
SD 메모리 카드 서비스	• FTP 클라이언트에서의 액세스 • Sysmac Studio에서의 SD 메모리 카드 조작 • SD 메모리 카드 명령의 실행
자체 진단 처리	• 하드웨어의 이상 감지

*1. NX102 및 NX1P2 CPU 유니트는 USB 포트를 장비하지 않았습니다.
*2. 시리얼 통신용 옵션 보드의 통신 처리는 NX1P2 CPU 유니트에서만 실행됩니다.
*3. CJ 고기능 유니트와의 할당 영역의 데이터 교환은 'I/O 리프레시' 시에 실행됩니다.
CJ 고기능 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

시스템 서비스의 실행 타이밍

NX701 CPU 유니트의 시스템 서비스는 태스크 및 태그 데이터 링크 서비스의 영향을 받지 않고, 언제라도 실행됩니다. 시스템 서비스의 실행 시간이 부족하지는 않습니다.

NX102 CPU 유니트의 시스템 서비스는 태스크의 영향을 받지 않고 실행됩니다. 단, 태그 데이터 링크 서비스 실행 중에는 시스템 서비스는 실행되지 않습니다.

NX1P2 CPU 유니트의 시스템 서비스는 태스크의 영향을 받지 않고 실행됩니다. 단, 태그 데이터 링크 서비스 및 옵션 보드 서비스의 실행 중에는 시스템 서비스는 실행되지 않습니다.

NJ 시리즈 CPU 유니트의 시스템 서비스는 태스크 및 태그 데이터 링크 서비스의 비는 시간에 실행됩니다.
단, 시스템 서비스 실행 시간이 충분히 확보되지 않는 경우에는 실행 우선도가 17 이상의 태스크 처리를 일시 정지하고 시스템 서비스가 실행됩니다.
시스템 서비스의 실행을 위한 시간은 Sysmac Studio의 '시스템 서비스 감시 설정'에서 설정합니다.

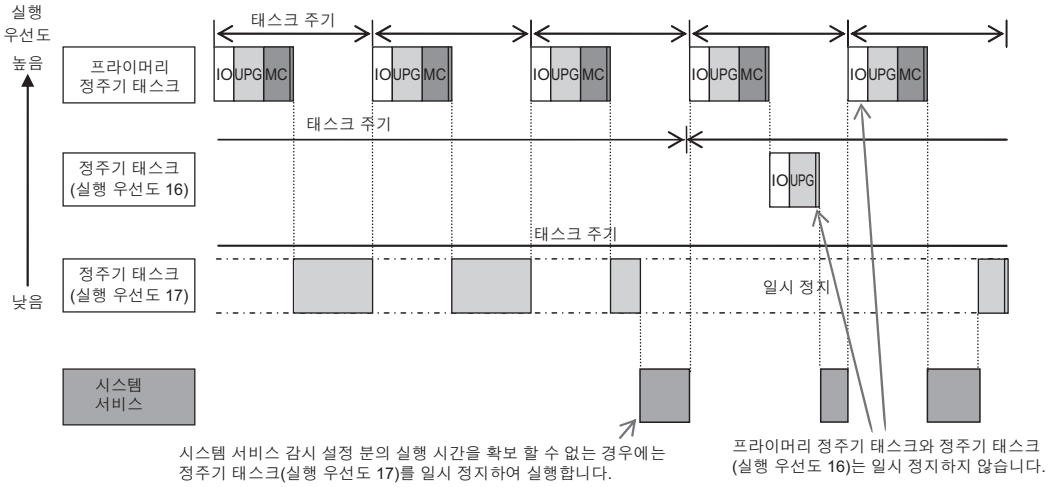
시스템 서비스의 처리 사이에 우선 순위는 없습니다. 각 처리는 타임 슬라이스되어 병렬로 실행됩니다.

- **시스템 서비스 실행 시간이 충분히 확보되지 않는 경우의 실행 타이밍**

  NJ 시리즈 CPU 유니트에서 태스크 및 태그 데이터 링크 서비스의 비는 시간이 적고, 시스템 서비스 실행 시간 이충분히 확보되지 않는 경우에는 실행 우선도가 17 이상의 태스크 실행을 일시 정지하고 시스템 서비스가 실행됩니다.

  실행 우선도가 17 이상인 태스크 실행을 일시 정지해도 시스템 서비스의 실행 시간이 확보되지 않는 경우에는 '시스템 서비스 시간 부족'이 발생하고, 사용자 프로그램의 실행이 정지됩니다.

  시스템 서비스의 실행을 위한 시간은 Sysmac Studio의 시스템 서비스 감시 설정에서 설정합니다.



- **시스템 서비스 감시 설정**

  시스템 서비스 감시 설정은 NJ 시리즈 CPU 유니트에서의 시스템 서비스 실행 시간이 충분히 확보되었는지 감시하기 위한 설정입니다. '시스템 서비스 실행 간격'에 모든 태스크가 실행하지 않는 시간이 '시스템 서비스 실행 간격'의 '시스템 서비스 실행 시간 비율' 이상인지 감시합니다.

  시스템 서비스 감시 설정은 Sysmac Studio의 '동작 설정'-'전반 설정'에서 설정할 수 있습니다.

설정 대항목	설정 중항목	설정 소항목 [단위]	기능 설명	설정값	기본값	반영 타이밍	운전 모드 중 재기입
동작 설정-동작 설정 탭-전체 설정	시스템 서비스 감시 설정	시스템 서비스 실행 간격	시스템 서비스를 실행하는 간격을 설정합니다.	10ms~1s	10ms	설정 전송 시	불가능
		시스템 서비스 실행 시간 비율	시스템 서비스를 실행하는 비율을 감시하는 값을 설정합니다.	5%~50%	10%	설정 전송 시	불가능

참고

NX 시리즈 CPU 유니트에서는 시스템 서비스 실행 시간이 부족하지는 않습니다. 따라서 시스템 서비스 감시 설정은 없습니다. 또한, 시스템 서비스 시간 부족 이상이 발생하는 경우도 없습니다.

사용상의 주의 사항

- NJ 시리즈 CPU 유니트에서는 시스템 서비스와 태스크 실행 시간을 충분히 확보하기 위해 시스템 서비스 처리의 응답 성능을 충족하는 범위에서 최소의 값을 시스템 서비스 감시 설정으로 설정하십시오. 시스템 서비스 감시 설정은 설정한 시간 분량의 시스템 서비스 실행 시간이 확보되었는지 감시하기 위한 설정입니다. 설정한 시간 분량의 시스템 서비스 실행을 보증하는 것은 아닙니다.
- NJ 시리즈 CPU 유니트에서는 시스템 서비스 실행 시간을 충분히 확보할 수 있도록 태스크를 설계하십시오. 태스크 설정 방법은 '5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'을 참조하십시오.
- NJ 시리즈 CPU 유니트에서 시스템 서비스의 실행 시간을 늘리고자 하는 경우에는 태스크 주기를 크게 하는 등 하여 전체 태스크의 비는 시간을 증가시키십시오.
- NJ 시리즈 CPU 유니트에서 실행 우선도 17 이상의 태스크를 일시 정지해도 시스템 서비스 실행 시간을 시스템 서비스 감시 설정 분량을 확보할 수 없는 경우에는 '시스템 서비스 시간 부족'이 발생하고, 사용자 프로그램 실행이 정지합니다.
- NJ 시리즈 CPU 유니트에서 시스템 서비스 실행 시간이 부족하여 실행 우선도 17 이상의 태스크가 일시 정지하는 경우에 일시 정지하는 태스크로 '태스크 주기 초과'가 발생할 가능성이 있습니다. 시스템 서비스가 시스템 서비스 감시 설정 분량 동작한 경우에도 실행 우선도 17 이상의 태스크가 주기 내에 완료되도록 태스크 설계를 하십시오.

5-7 태스크 관련 할당과 설정

여기에서는 태스크에 대한 할당과 설정에 대해 설명합니다.

5-7-1 I/O 리프레시의 태스크에 대한 할당

EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, CPU 유니트의 내장 I/O, CPU 유니트의 옵션 보드 및 CJ 시리즈 유니트의 I/O 리프레시를 태스크에 할당합니다. I/O 리프레시 대상에 따라 할당 가능한 태스크와 할당 단위가 다릅니다. 할당 단위는 하나의 I/O 리프레시에 할당할 수 있는 I/O 리프레시 대상의 그룹입니다. 예를 들어 할당 단위가 슬레이브 터미널인 경우에 1대의 통신 커플러 유니트에 복수의 NX 유니트를 연결해도 I/O 리프레시를 할당할 수 있는 태스크는 하나뿐입니다.

I/O 리프레시를 할당하지 않은 태스크로 입출력을 실행하고자 하는 경우에는 📖'I/O 리프레시를 할당하지 않은 태스크의 입출력(5-80페이지)'을 참조하십시오.

I/O 리프레시 대상, 할당 가능한 태스크, 할당 단위의 관계는 다음과 같습니다.

I/O 리프레시 대상	할당 가능한 태스크	할당 단위
통신 커플러 유니트 단, 슬레이브 터미널 상에 NX 시리즈 세이프티 컨트롤 유니트 있음	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5, 16)*1*2*3	슬레이브 터미널
통신 커플러 유니트 단, 슬레이브 터미널 상에 축에 할당된 NX 유니트 있음	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)*3	
통신 커플러 유니트 단, 슬레이브 터미널 상에 축 할당하고 있는 NX 유니트 없음	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5, 16)*2*3	
[축]에 할당된 EtherCAT 슬레이브	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5)*2*3	슬레이브
상기 이외의 EtherCAT 슬레이브	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5, 16)*2*3	
CPU 유니트 상의 NX 유니트*4, CPU 유니트의 내장 I/O*5	프라이머리 정주기 태스크	−
CJ 시리즈 기본 I/O 유니트*6	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)	유니트
CJ 시리즈의 고기능 I/O 유니트*6		
CJ 시리즈의 CPU 고기능 유니트*6		

*1. NX 시리즈 세이프티 컨트롤 유니트를 장착한 슬레이브 터미널이 여러 개 있는 경우에 정주기 태스크(실행 우선도 5)와 정주기 태스크(실행 우선도 5) 이외의 정주기 태스크를 조합해서 할당할 수 없습니다.
*2. 정주기 태스크(실행 우선도 5)를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*3. NX102 및 NX1P2 CPU 유니트에서는 프라이머리 정주기 태스크뿐입니다.
*4. CPU 유니트 상의 NX 유니트는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.
*5. CPU 유니트의 내장 I/O는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.
*6. CJ 시리즈의 각종 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

안전상의 중요 사항

CJ 시리즈 기본 유니트와 EtherCAT 슬레이브와 같이 다른 기능 모듈을 연계하여 장치를 제어하는 경우에는 한쪽의 기능 모듈이 정지해도 장치에 영향을 주지 않도록 사용자 프로그램 및 외부에서 대책을 실시하십시오. 각각의 기능 모듈에 부분 정지 이상 레벨의 이상이 발생한 경우에 해당하는 출력은 슬레이브 및 유니트의 사양에 따릅니다.

부분 정지 이상 레벨의 컨트롤러 이상에 대해서는 📖"NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"의 'CPU 유니트가 동작할 수 있는 이상'을 참조하십시오.

참고

EtherCAT 슬레이브와 CJ 유니트를 사용하는 경우에는 각각 다른 태스크에 I/O 리프레시를 할당하면 프라이머리 정주기 태스크의 태스크 실행 시간을 줄일 수 있습니다. 프라이머리 정주기 태스크에는 EtherCAT 슬레이브, 정주기 태스크(실행 우선도 16)에는 CJ 유니트를 할당하십시오.

Sysmac Studio에서의 설정 조작

축에 할당하지 않은 각 슬레이브/유니트는 Sysmac Studio의 [태스크 설정]–[I/O를 제어하는 태스크 설정]에서 I/O 리프레시하는 태스크를 설정합니다.

자세한 내용은 📖'I/O를 제어하는 태스크 설정(4-9페이지)'을 참조하십시오.

축에 할당된 각 슬레이브/유니트는 Sysmac Studio의 [모션 제어 설정]을 통해 사용하는 모션 제어를 선택하면 I/O 리프레시하는 태스크가 설정됩니다.

자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

I/O 리프레시 타이밍

EtherCAT 슬레이브, CPU 유니트 상의 NX 유니트, CPU 유니트의 내장 I/O, CPU 유니트의 옵션 보드 및 CJ 시리즈 유니트의 I/O 리프레시 타이밍은 다음과 같습니다.

I/O 리프레시 대상	I/O 리프레시하는 태스크	I/O 리프레시의 주기
EtherCAT 슬레이브	프라이머리 정주기 태스크	프라이머리 정주기 태스크의 태스크 주기
	정주기 태스크(실행 우선도 5)*1	정주기 태스크(실행 우선도 5)의 태스크 주기
	정주기 태스크(실행 우선도 16)	프라이머리 정주기 태스크의 태스크 주기*2
CPU 유니트 상의 NX 유니트*3	프라이머리 정주기 태스크	프라이머리 정주기 태스크의 태스크 주기
CPU 유니트의 내장 I/O*4	프라이머리 정주기 태스크	프라이머리 정주기 태스크의 태스크 주기
CJ 유니트*5	프라이머리 정주기 태스크	프라이머리 정주기 태스크의 태스크 주기
	정주기 태스크(실행 우선도 16)	정주기 태스크(실행 우선도 16)의 태스크 주기

*1. 정주기 태스크(실행 우선도 5)를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*2. EtherCAT 통신은 프라이머리 정주기 태스크의 I/O 리프레시로 실행됩니다. EtherCAT 슬레이브를 정주기 태스크(실행 우선도 16)로 제어하는 경우에 I/O 리프레시로의 데이터 갱신 주기는 정주기 태스크(실행 우선도 16)의 주기입니다.
*3. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*4. 내장 I/O를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*5. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

- **프로세스 데이터 통신의 우선 순위**

  EtherCAT의 프로세스 데이터 통신은 I/O 리프레시로 실행됩니다. NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에서 프로세스 데이터 통신을 실행할 수 있습니다.

  프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)로 프로세스 데이터 통신을 실행하는 경우에 프라이머리 정주기 태스크의 프로세스 데이터 통신이 우선됩니다. 따라서 2가지 태스크의 프로세스 데이터 통신 주기 및 프로세스 데이터 크기가 동일해도 정주기 태스크(실행 우선도 5)의 I/O 리프레시의 시간 쪽이 길어집니다.

사용자 프로그램에서 I/O에 액세스하는 방법

사용자 프로그램에서 I/O 포트에 액세스하려면 디바이스 변수를 사용합니다.
[I/O 를 제어하는 태스크 설정]에 설정한 태스크의 프로그램에서 디바이스 변수에 액세스하십시오.

I/O 리프레시를 할당하지 않은 태스크의 입출력

I/O 리프레시를 할당하지 않은 태스크에서 데이터를 출력하려고 하면 Sysmac Studio의 프로그램 확인 시 에러가 됩니다. 예를 들어 슬레이브 터미널에서 각 NX 유니트의 처리를 다른 태스크에 할당한 경우에 I/O 리프레시를 할당된 태스크에서만 데이터를 출력할 수 없습니다. 이와 같은 경우에는 I/O 리프레시를 할당하지 않은 태스크에서 I/O 리프레시를 할당한 태스크에 데이터를 전달하고, 그 후 I/O 리프레시를 할당한 태스크에서 데이터를 출력해야 합니다.

다음은 I/O 리프레시를 할당하지 않은 태스크에서 I/O 리프레시를 할당한 태스크에 데이터를 전달하여 출력하는 샘플 프로그램입니다.

또한, 이 샘플 프로그램에서는 I/O 리프레시를 할당하지 않은 태스크로 외부 데이터를 입력하는 처리를 실행하고 있습니다. 이와 같은 처리를 실행하려고 하면Sysmac Studio의 프로그램 확인 시 경고가 출력되지만 실행은 가능합니다.

- **유니트 구성**

  슬레이브 터미널을 사용합니다. 슬레이브 터미널의 유니트 구성은 다음과 같습니다.

형식	제품 이름
NX-ECC201	EtherCAT 커플러 유니트
NX-ID3317	DC 입력 유니트
NX-OD3256	트랜지스터 출력 유니트

- **I/O 맵**

  I/O 맵은 다음과 같습니다. 표에는 샘플 프로그램에서 사용하는 접점만 표시합니다.

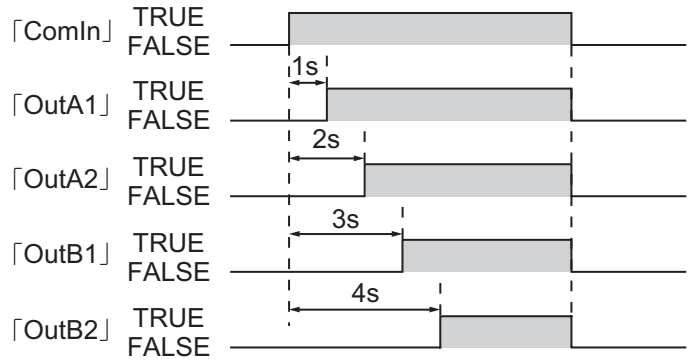
위치	포트	설명	R/W	데이터 형식	변수	변수 코멘트	변수 유형
Unit1	NX-ID3317						
	Input Bit 00	입력 접점 00	R	BOOL	ComIn	공통 입력값	글로벌 변수
Unit2	NX-OD3256						
	Output Bit 00	출력 접점 00	W	BOOL	OutA1	출력값 A1	글로벌 변수
	Output Bit 01	출력 접점 01	W	BOOL	OutA2	출력값 A2	글로벌 변수
	Output Bit 02	출력 접점 02	W	BOOL	OutB1	출력값 B1	글로벌 변수
	Output Bit 03	출력 접점 03	W	BOOL	OutB2	출력값 B2	글로벌 변수

● **입출력 사양**

입출력 사양은 다음과 같습니다.

- 'ComIn'의 값이 FALSE에서 TRUE로 변화하고 1초 후에 'OutA1', 2초 후에 'OutA2', 3초 후에 'OutB1', 4초 후에 'OutB2'의 값이 각각 FALSE에서 TRUE로 변화합니다.
- 'ComIn'의 값이 TRUE에서 FALSE로 변화했으면 'OutA1', 'OutA2', 'OutB1', 'OutB2'의 모든 값이 TRUE에서 FALSE로 변화합니다.

다음은 타이밍 차트의 예시입니다.



● **태스크의 처리 내용**

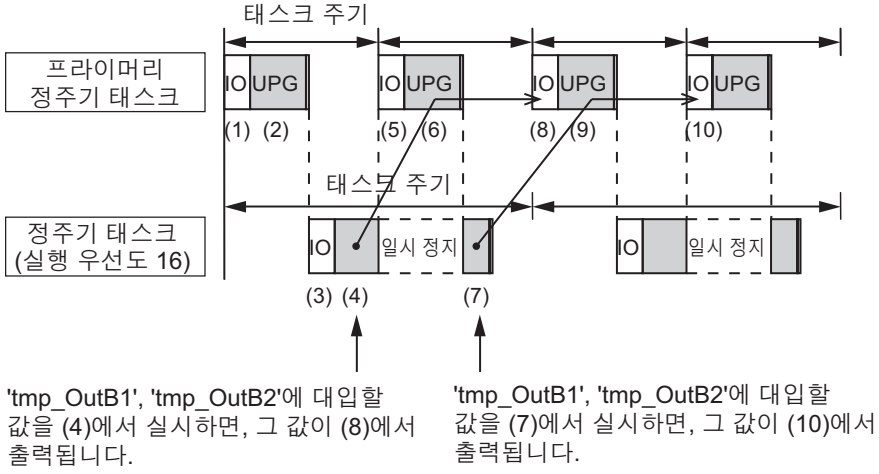
태스크에는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16)를 사용합니다. 프라이머리 정주기 태스크에 I/O 리프레시를 할당합니다.

'OutA1', 'OutA2'의 제어는 프라이머리 정주기 태스크로 실행합니다. 'ComIn'의 입력에서 연산, 'OutA1', 'OutA2'의 출력까지 일련의 처리를 모두 프라이머리 정주기 태스크로 실행합니다.

'OutB1', 'OutB2'의 제어는 'ComIn'의 입력에서 연산까지 정주기 태스크로 실행합니다. 정주기 태스크로의 연산 결과는 일시 변수의 글로벌 변수 'tmp_OutB1', 'tmp_OutB2'에 대입합니다. 프라이머리 정주기 태스크에서는 'tmp_OutB1', 'tmp_OutB2'의 값을 'OutB1', 'OutB2'에 대입한 후, 'OutB1', 'OutB2'가 출력됩니다.

정주기 태스크는 프라이머리 정주기 태스크의 실행에 따라 일시 정지하는 경우가 있습니다. 정주기 태스크로의 연산 결과를 'tmp_OutB1', 'tmp_OutB2'에 대입하는 처리를 정주기 태스크의 일시 정지 전에 실행할지 나중에 실행할지에 따라 정주기 태스크의 연산 결과가 프라이머리 정주기 태스크에 의해 출력되는 타이밍이 바뀝니다.

처리 흐름의 예는 다음 그림과 같습니다. 정주기 태스크의 태스크 주기는 프라이머리 정주기 태스크의 태스크 주기의 2배로 합니다.



(1) 프라이머리 정주기 태스크의 I/O 리프레시로 'ComIn'의 입력이 실행됩니다.

(2) 프라이머리 정주기 태스크의 사용자 프로그램 실행에서 연산을 실행하고, 'OutA1', 'OutA2'에 값을 대입합니다.
또한, 'tmp_OutB1', 'tmp_OutB2'의 값을 'OutB1', 'OutB2'에 대입합니다. 그러나 아직 정주기 태스크로'tmp_OutB1', 'tmp_OutB2'에 값을 대입하지 않았으므로 'tmp_OutB1', 'tmp_OutB2'의 값은 초기값이 되었습니다. 따라서 'OutB1', 'OutB2'의 값도 초기값입니다.

(3) 정주기 태스크의 I/O 리프레시로 'ComIn'의 입력이 실행됩니다.

(4) 정주기 태스크의 사용자 프로그램 실행으로 연산을 실행합니다.
정주기 태스크 실행 중에 프라이머리 정주기 태스크 실행이 시작되면 정주기 태스크는 일시 정지합니다. 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 정주기 태스크 일시 정지 전에 실행할지 나중에 실행할지는 정주기 태스크 내의 처리 타이밍에 따라 바뀝니다.

(5) 프라이머리 정주기 태스크의 I/O 리프레시로 'OutA1', 'OutA2', 'OutB1', 'OutB2'의 출력이 실행됩니다.
또한, 2번째 'ComIn'의 입력이 실행됩니다.

(6) 프라이머리 정주기 태스크의 사용자 프로그램 실행에서 연산을 실행하고, 'OutA1', 'OutA2'에 값을 대입합니다.
또한, 'tmp_OutB1', 'tmp_OutB2'의 값을 'OutB1', 'OutB2'에 대입합니다. (4)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행한 경우에는 정주기 태스크의 연산 결과가 'OutB1', 'OutB2'에 반영됩니다. (4)에서는 아직 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행하지 않은 경우에는 'tmp_OutB1', 'tmp_OutB2'의 값은 초기값이 되었습니다. 따라서 'OutB1', 'OutB2'의 값도 초기값입니다.

(7) 정주기 태스크의 사용자 프로그램 실행으로 (4)의 나머지 연산을 실행합니다.
(4)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행하지 않았던 경우에 여기에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행합니다.

(8) 프라이머리 정주기 태스크의 I/O 리프레시로 'OutA1', 'OutA2', 'OutB1', 'OutB2'의 출력이 실행됩니다.
(4)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행한 경우에는 정주기 태스크의 연산 결과가 'OutB1', 'OutB2'로 출력됩니다.
(7)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행한 경우에는 'OutB1', 'OutB2'의 초기값이 출력됩니다.

(9) 프라이머리 정주기 태스크의 사용자 프로그램 실행에서 연산을 실행하고, 'OutA1', 'OutA2'에 값을 대입합니다.
또한, 'tmp_OutB1', 'tmp_OutB2'의 값을 'OutB1', 'OutB2'에 대입합니다. (7)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행한 경우에는 여기에서 정주기 태스크의 연산 결과가 'OutB1', 'OutB2'에 반영됩니다.

(10) 프라이머리 정주기 태스크의 I/O 리프레시로 'OutA1', 'OutA2', 'OutB1', 'OutB2'의 출력이 실행됩니다.
(7)에서 'tmp_OutB1', 'tmp_OutB2'에 대한 값의 대입을 실행한 경우에는 여기에서 정주기 태스크의 연산 결과가 'OutB1', 'OutB2'로 출력됩니다.

또한, 'tmp_OutB1', 'tmp_OutB2'의 값이 프라이머리 정주기 태스크 내에서 참조되기 전에 정주기 태스크에 의해 덮어쓰이는 것을 막기 위해 Lock, Unlock 명령을 사용하여 태스크 간의 배타 제어를 실행합니다. 태스크 간의 배타 제어의 자세한 내용은 '5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법(5-90페이지)'을 참조하십시오.

- **글로벌 변수 테이블**

  글로벌 변수는 다음과 같습니다.

  글로벌 변수의 변수 테이블

이름	데이터 형식	초기값	할당 대상	코멘트
ComIn	BOOL	FALSE	ECAT://node#[1,1]/Input Bit 00	공통 입력값
OutA1	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 00	출력값 A1
OutA2	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 01	출력값 A2
OutB1	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 02	출력값 B1
OutB2	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 03	출력값 B2
tmp_OutB1	BOOL	FALSE		B1의 일시 변수
tmp_OutB2	BOOL	FALSE		B2의 일시 변수

- **프라이머리 정주기 태스크의 래더도**

  프라이머리 정주기 태스크의 래더도는 다음과 같습니다.

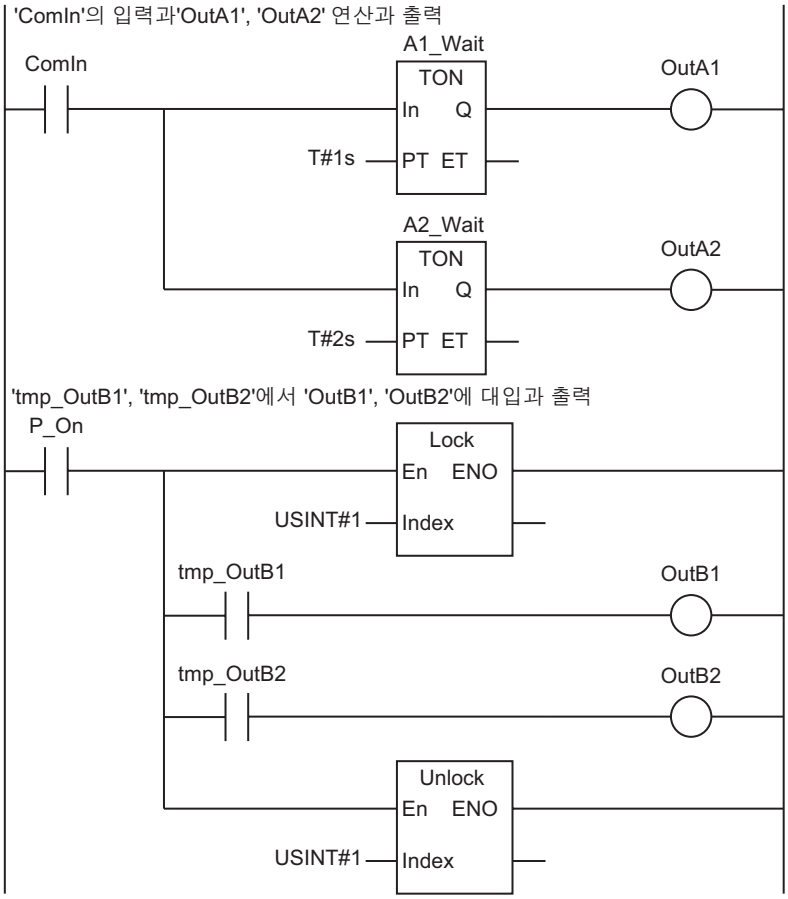
  내부 변수 테이블

이름	데이터 형식
A1_Wait	TON
A2_Wait	TON

  외부 변수 테이블

이름	데이터 형식	코멘트
ComIn	BOOL	공통 입력값
OutA1	BOOL	출력값 A1
OutA2	BOOL	출력값 A2
OutB1	BOOL	출력값 B1
OutB2	BOOL	출력값 B2
tmp_OutB1	BOOL	B1의 일시 변수
tmp_OutB2	BOOL	B2의 일시 변수

알고리즘 부분



- **정주기 태스크의 래더도**

  정주기 태스크의 래더도는 다음과 같습니다.

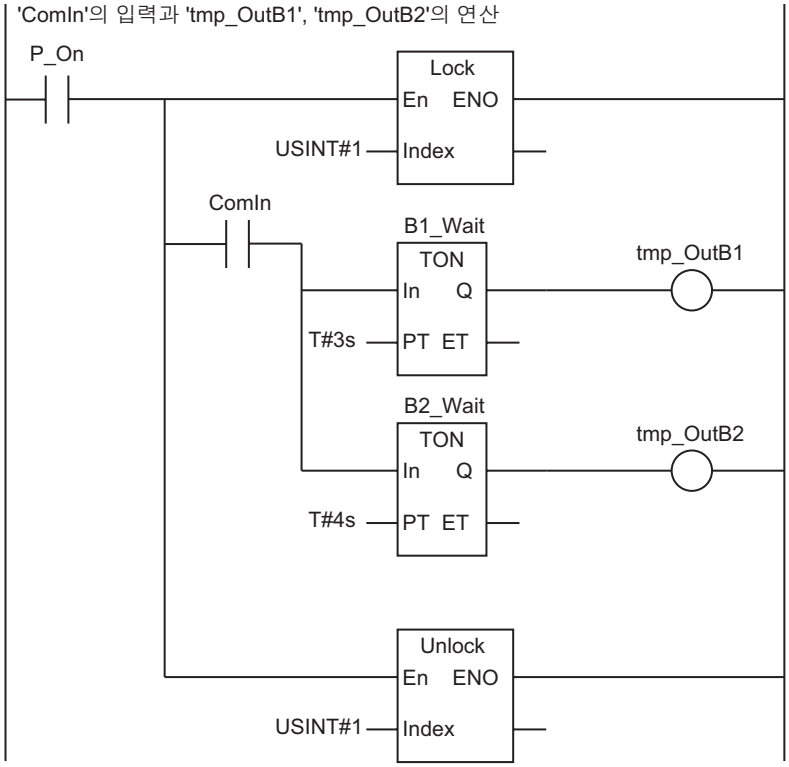
  내부 변수 테이블

이름	데이터 형식
B1_Wait	TON
B2_Wait	TON

  외부 변수 테이블

이름	데이터 형식	코멘트
ComIn	BOOL	공통 입력값
tmp_OutB1	BOOL	B1의 일시 변수
tmp_OutB2	BOOL	B2의 일시 변수

정주기 태스크의 알고리즘 부분



5-7-2 프로그램의 태스크에 대한 할당

실행하는 프로그램을 태스크에 할당합니다(1태스크당 최대 128개). 또한, 각 프로그램의 운전 시작 시 동작을 설정합니다.

프로그램 실행 순서

태스크 내에서의 프로그램 실행 순서를 Sysmac Studio에서 설정합니다.

프로그램의 운전 시작 시 초기 상태

각 프로그램의 운전 시작 시 동작을 설정합니다. 운전 시작 시 초기 상태란, 컨트롤러의 동작 모드가 프로그램 모드에서 운전 모드에 변경된 다음 해당 프로그램이 할당된 태스크가 처음으로 실행되었을 때 해당 프로그램을 실행시킬지 여부를 설정하는 것입니다. 설정 범위는 ' 기동', '정지' 중 하나입니다.
초기 상태가 '정지'인 경우에 PrgStart 명령에서 실행을 명령하면 다음에 해당 프로그램이 실행되는 타이밍부터 실행됩니다. 초기 상태가 '기동'인 경우에 PrgStop 명령에서 정지를 명령하면 다음에 해당 프로그램이 실행되는 타이밍부터 정지합니다.

- **Sysmac Studio에서의 설정 조작**

  Sysmac Studio에서 [구성·설정]–[태스크 설정]–[프로그램 할당 설정]에서 태스크에 대한 프로그램의 할당, 태스크 내에서의 프로그램 실행 순서 및 각 프로그램의 초기 상태를 설정합니다. 자세한 내용은 📖'프로그램 할당 설정(4-9페이지)'을 참조하십시오.

버전 관련 정보

Sysmac Studio의 프로그램의 운전 시작 시 초기 상태 설정 및 PrgStart 명령, PrgStop 명령은 CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

태스크에 할당할 수 있는 POU

태스크에는 0~128개의 프로그램을 할당할 수 있습니다.
할당 가능한 POU는 프로그램뿐입니다. FB 인스턴스 및 FUN을 직접 태스크에 할당할 수 없습니다. 동일한 프로그램을 복수의 태스크에 할당할 수 없습니다.

5-7-3 태스크의 설정 항목

각 태스크의 설정 항목은 다음과 같습니다.

● 프라이머리 정주기 태스크의 설정 항목

설정 항목		설정 범위	기본값	반영 타이밍	운전 모드 중 재기입
태스크 타입		'프라이머리 정주기 태스크'를 선택합니다.	–	설정 전송 시	불가능
	실행 우선도	'4'에 고정	–		
태스크 이름		문자열	PrimaryTask		
주기/ 실행 조건	태스크 주기*1	• NX701 CPU 유니트의 경우 125μs, 250μs~8ms(250μs 단위) • NX102 CPU 유니트의 경우 1~32ms(250μs 단위) • NX1P2 CPU 유니트의 경우 2~8ms(250μs 단위) • NJ 시리즈 CPU 유니트의 경우 500μs*2 , 1ms, 2ms, 4ms	1ms*3		
태스크 주기 초과 감지		태스크 실행 시간이 설정한 태스크 주기를 초과한 경우에 에러를 감지할지 여부를 설정합니다. • 감지(가벼운 이상 레벨의 컨트롤러 이상 발생) • 감지 안 함(감시 정보로 이벤트 로그에 남김) 자세한 내용은 📖'태스크 주기 초과 (5-100페이지)'를 참조하십시오.	감지		
태스크 실행 타임아웃 시간		무한 루프 등에 따라 태스크의 실제 처리가 완료되지 않았을 때의 타임아웃 감지 시간을 설정합니다. 태스크 주기의 배수로 설정합니다. 1배~5배 자세한 내용은 📖'태스크 실행 타임아웃 (5-101페이지)'을 참조하십시오.	5배		
변수 액세스 시간[%]		변수 액세스 시간의 태스크 주기에 대한 비율을 설정합니다. 1%~50% 📖'변수 액세스 시간 설정(5-98페이지)' 을 참조하십시오.	3%		

*1. EtherCAT 설정의 프로세스 데이터 통신 주기(프로세스 데이터 통신 주기 1)는 이 주기와 동일한 값이 됩니다.
*2. NJ301-□□□□에서는 유니트 버전 Ver.1.03 이상에서 사용할 수 있습니다.
NJ101-□□□□에서는 사용할 수 없습니다.
*3. NX102-□□□□, NX1P2-□□□□ 및 NJ101-□□□□의 프라이머리 정주기 태스크의 기본값은 2ms입니다.

● 정주기 태스크(실행 우선도 5)의 설정 항목

설정 항목		설정 범위	기본값	반영 타이밍	운전 모드 중 재기입
태스크 타입		'정주기 태스크(우선도 5)'를 선택합니다.*1	–	설정 전송 시	불가능
	실행 우선도	'5'가 자동으로 선택됩니다.	–		
태스크 이름		문자열	Periodic Task0		
주기/ 실행 조건	태스크 주기*2	• NX701 CPU 유니트의 경우 125μs, 250μs~100ms(250μs 단위)	2ms		
태스크 주기 초과 감지		프라이머리 정주기 태스크와 동일합니다.	프라이머리 정주기 태스크와 동일합니다.		
태스크 실행 타임아웃 시간					
변수 액세스 시간[%]					

*1. 정주기 태스크(실행 우선도 5)를 사용할 수 있는 것은 NX701 CPU 유니트뿐입니다.
*2. EtherCAT 설정의 프로세스 데이터 통신 주기 2는 이 주기와 동일한 값이 됩니다.

● 정주기 태스크(실행 우선도 16, 17, 18)의 설정 항목

설정 항목		설정 범위	기본값	반영 타이밍	운전 모드 중 재기입
태스크 타입		다음 중 하나를 선택합니다. '정주기 태스크(우선도 16)'*1 '정주기 태스크(우선도 17)' '정주기 태스크(우선도 18)'	–	설정 전송 시	불가능
	실행 우선도	'16' '17' '18' 중 하나가 자동으로 선택됩니다.	–		
태스크 이름		문자열	Periodic Task0		
주기/ 실행 조건	태스크 주기	• NX701 CPU 유니트의 경우 📖'5-3-1 NX701의 태스크 사양(5-9페이지)' 참조 • NX102 및 NX1P2 CPU 유니트인 경우 📖'5-4-1 NX102 및 NX1P2의 태스크 사양(5-29페이지)' 참조 • NJ 시리즈 CPU 유니트의 경우 📖'5-5-1 NJ 시리즈의 태스크 사양(5-46페이지)' 참조	10ms		
태스크 주기 초과 감지		프라이머리 정주기 태스크와 동일합니다.	프라이머리 정주기 태스크와 동일합니다.		
태스크 실행 타임아웃 시간					
변수 액세스 시간[%]					

*1. NX102 및 NX1P2 CPU 유니트에서는 정주기 태스크(우선도 16)를 사용할 수 없습니다.

● **이벤트 태스크의 설정 항목**

설정 항목	설정 범위	기본값	반영 타이밍	운전 모드 중 재기입
태스크 타입	다음 중 하나를 선택합니다. '이벤트 태스크(우선도 8)' '이벤트 태스크(우선도 48)'	−	설정 전송 시	불가능
태스크 이름	문자열	Event Task0		
실행 조건	'명령에 의한 실행', '변수의 조건식 일치' 중 하나를 선택합니다.	명령에의한 실행		
태스크 실행 타임아웃 시간	무한 루프 등에 따라 태스크의 실제 처리가 완료되지 않았을 때의 타임아웃 감지 시간을 설정합니다. 설정 단위는 1ms입니다. • 실행 우선도 '8'인 경우: 1~500ms • 실행 우선도 '48'인 경우: 1ms~10s	• 실행 우선도 '8'인 경우: 200ms • 실행 우선도 '48'인 경우: 1s		

● **Sysmac Studio에서의 설정 조작**

Sysmac Studio에서 [구성·설정]–[태스크 설정]에서 태스크 추가와 각 태스크 설정을 실행합니다.

자세한 내용은 📖'태스크 설정(4-7페이지)'을 참조하십시오.

5-8 변수의 동시성 확보

여기에서는 태스크 사이에서 변수 값의 동시성을 확보하는 방법과 컨트롤러 외부에서의 변수 액세스에 대해 설명합니다.

5-8-1 태스크 사이에서 변수 값의 동시성을 확보하는 방법

다른 복수의 태스크에서 동일한 글로벌 변수에 읽기/쓰기를 하는 경우에 태스크 사이에서 글로벌 변수 값의 동시성을 확보하는 방법에는 다음 2가지가 있습니다. 이를 총칭하여 '변수의 태스크 간 배타 제어 기능'이라고 합니다.

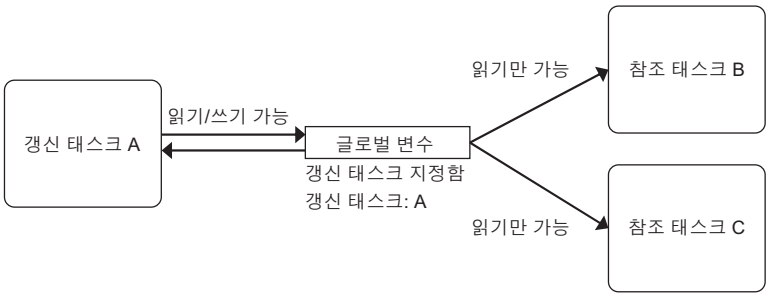
방법1: 하나의 태스크에서 글로벌 변수에 쓰고, 다른 복수 태스크에서 읽는 경우
변수의 태스크 간 배타 제어 설정 기능을 사용합니다.
방법2: 복수의 태스크에서 글로벌 변수에 쓰고자 하는 경우
태스크 간 배타 제어 명령을 사용합니다.

방법 1) 변수의 태스크 간 배타 제어 설정 기능

- **개요**

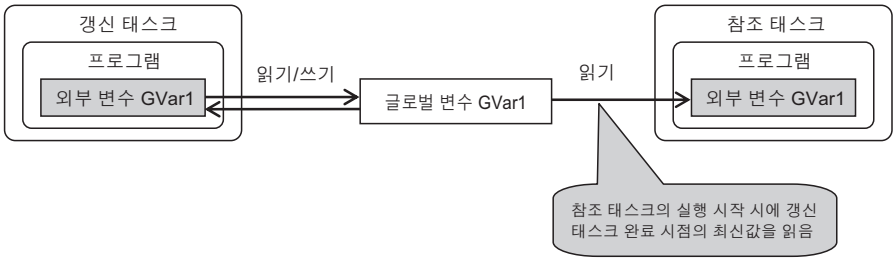
  글로벌 변수를 갱신하는 태스크와 참조하는 태스크를 지정하면 참조하는 측의 태스크에서 본 글로벌 변수 값의 동시성을 확보하는 기능입니다.

  글로벌 변수에 읽기/쓰기를 실행할 태스크(갱신 태스크)를 하나 지정합니다. 그 글로벌 변수에 대해 읽기만 실행할 태스크(참조 태스크)를 지정합니다. 이것으로 글로벌 변수 값의 동시성을 확보합니다.



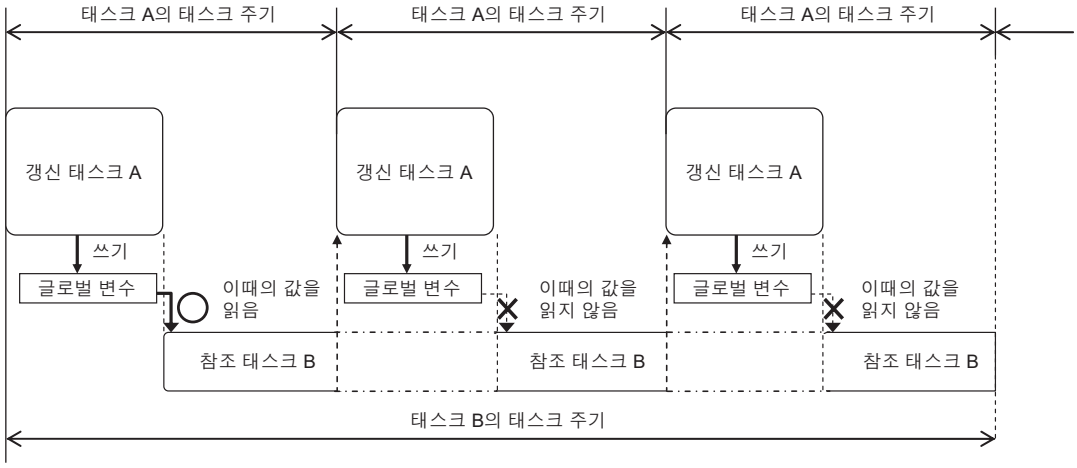
- **용도 예**

  프라이머리 정주기 태스크에 의해 쓰여지는 글로벌 변수가 있고, 해당 변수의 값을 정주기 태스크 측에서 읽을 때, 그 정주기 태스크 내에서 변수 값의 동시성을 확보할 때 사용합니다.



● **구조**

갱신 태스크가 설정된 글로벌 변수에 대해서는 참조 태스크는 자신의 태스크 실행 시작 시 갱신 태스크가 실행 완료되고 쓴 값을 읽습니다.

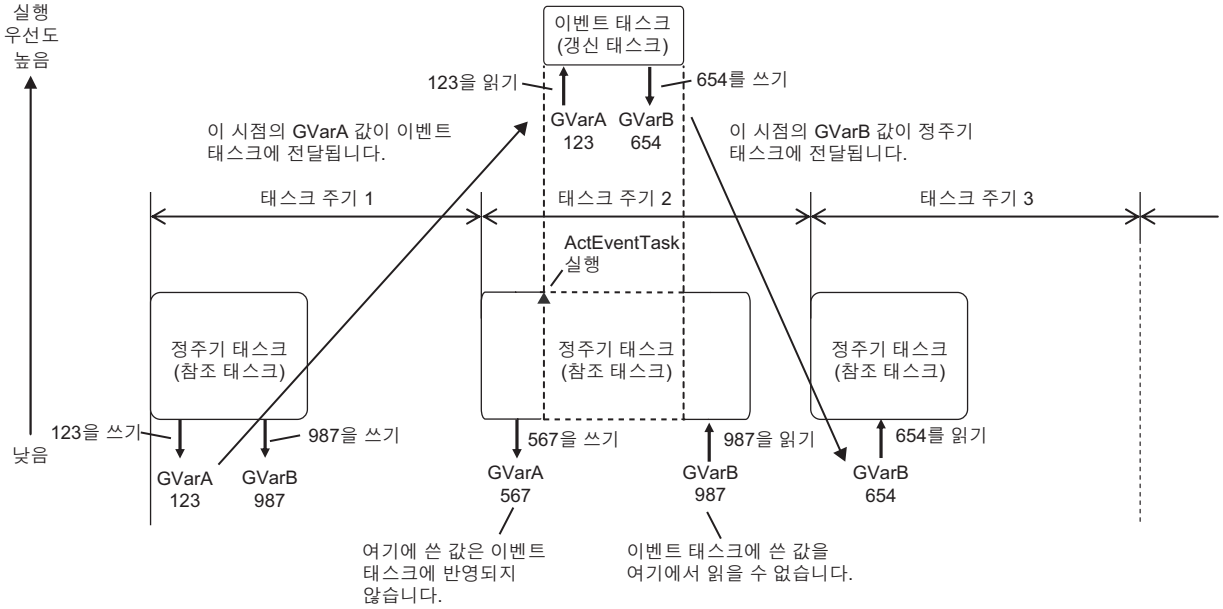


이것으로 특별한 사용자 프로그램 없이 태스크 내에서 글로벌 변수 값의 동시성을 확보할 수 있습니다.

참조 태스크에서 글로벌 변수에 값을 쓰는 명령을 사용하면 Sysmac Studio의 프로그램 확인에서 에러가 됩니다.

사용상의 주의 사항

ActEventTask 명령을 사용하는 경우에 2개의 태스크 사이에서 글로벌 변수의 어느 시점의 값을 읽는지, 어느 시점에 값이 갱신되는지 주의하십시오.
예를 들어 다음 그림의 경우에 이벤트 태스크가 읽는 글로벌 변수 GVarA의 값은 태스크 주기 1의 종료 시점의 값입니다. 따라서 태스크 주기 2에서 정주기 태스크가 GVarA의 값을 써도 그것은 이벤트 태스크에 반영되지 않습니다.
또한, 이벤트 태스크가 쓴 글로벌 변수 GVarB의 값이 정주기 태스크에 전달되는 것은 태스크 주기 3의 시작 시점입니다. 따라서 태스크 주기 2에서 정주기 태스크가 GVarB의 값을 읽어도 이벤트 태스크가 쓴 값을 읽을 수 없습니다.



이와 같이 ActEventTask 명령을 사용하여 이벤트 태스크를 실행하는 경우에 글로벌 변수 값의 전달에는 '변수의 태스크 간 배타 제어 기능'을 사용하지 마십시오.
ActEventTask 명령 사용 시 글로벌 변수의 동시성을 확보하기 위해서는 태스크 실행 중 판정 명령 Task_IsActive가 유효합니다. Task_IsActive 명령은 지정한 태스크가 실행 중인지 또는 대기 중인지 여부를 판정하는 명령입니다. 이 명령을 사용하여 이벤트 태스크 실행 중에는 이벤트 태스크가 쓰기를 실행하는 변수를 다른 태스크에서 참조하지 않도록 하십시오. Task_IsActive 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

- **제한**
  - 글로벌 변수 하나에 갱신 태스크는 하나만 지정합니다. 동일한 글로벌 변수에 대해 복수의 태스크가 쓰기 가능으로 하고자 하는 때는 뒤에 설명하는 태스크 간 배타 제어 명령을 사용하십시오.
  - 구조체/공용체 변수에 갱신 태스크를 설정하는 경우에는 구조체/공용체 변수 전체에서 하나의 갱신 태스크를 설정합니다. 구조체/공용체 멤버별로 갱신 태스크를 설정할 수 없습니다.
  - 배열 변수에 갱신 태스크를 설정하는 경우에는 배열 변수 전체에서 하나의 갱신 태스크를 설정합니다. 배열 요소별로 갱신 태스크를 설정할 수 없습니다.

사용상의 주의 사항

값의 동시성을 확보하고자 하는 글로벌 변수에는 참조 태스크도 갱신 태스크도 아닌 태스크에서 값을 쓰지 마십시오. 갱신 태스크를 설정한 글로벌 변수에 갱신 태스크도 참조 태스크도 아닌 태스크의 사용자 프로그램에서 읽기/쓰기를 한 경우 이 글로벌 변수 값의 동시성은 보증할 수 없습니다. 이와 같은 프로그램을 작성하면 프로그램 확인 시 경고가 출력됩니다.

참고

- 변수의 태스크 간 배타 제어 설정 기능을 사용하는 글로벌 변수는 해당 글로벌 변수를 나타내는 외부 변수를 데이터 트레이스의 샘플링 대상으로 설정할 수 있습니다. 이것으로 데이터 트레이스에서 갱신 태스크와 참조 태스크, 각각의 글로벌 변수 값을 샘플링할 수 있습니다. 데이터 트레이스로 대해서는 '8-6-4 데이터 트레이스(8-56페이지)'를 참조하십시오.

- **Sysmac Studio에서의 설정 조작**

  Sysmac Studio에서 [태스크 설정]–[변수의 태스크 간 배타 제어 설정]에서 갱신 태스크를 지정하는 글로벌 변수('갱신하는 변수') 및 그에 대한 참조 태스크('참조하는 태스크')를 설정합니다.

  자세한 내용은 '변수의 태스크 간 배타 제어 설정(4-10페이지)'을 참조하십시오.

방법 2)태스크 간 배타 제어 명령

복수의 태스크에서 동일한 글로벌 변수에 쓰는 경우에 태스크 간 배타 제어 명령(Lock/Unlock 명령)을 사용하여 글로벌 변수의 동시성을 확보합니다.

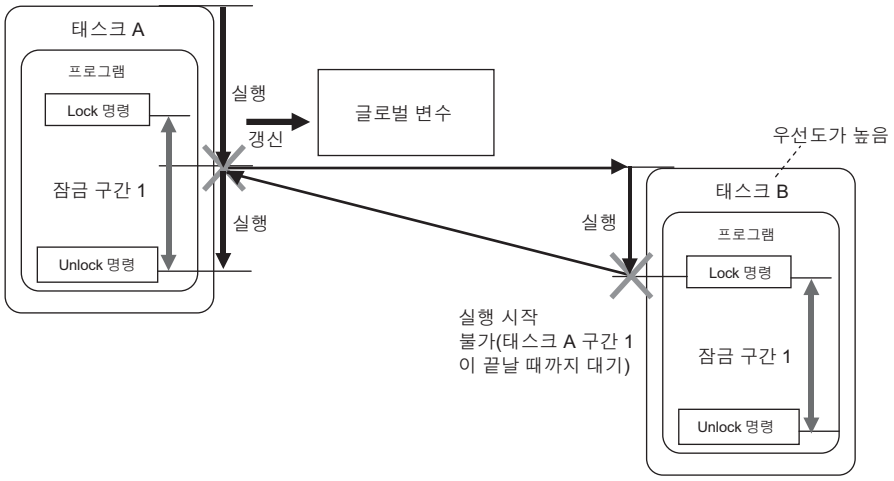
태스크 간 배타 제어 명령은 Lock 명령에서 Unlock 명령까지의 구간(잠금 구간) 실행 중에 다른 태스크의 동일한 잠금 번호의 잠금 구간을 실행할 수 없게 합니다. 글로벌 변수로의 쓰기를 잠금 구간에 하면 복수의 태스크에서 글로벌 변수를 쓰는 경우도 값의 동시성을 확보할 수 있습니다.

자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 Lock(태스크 간 배타 잠금) 명령, Unlock(태스크 간 배타 잠금 해제) 명령을 참조하십시오.

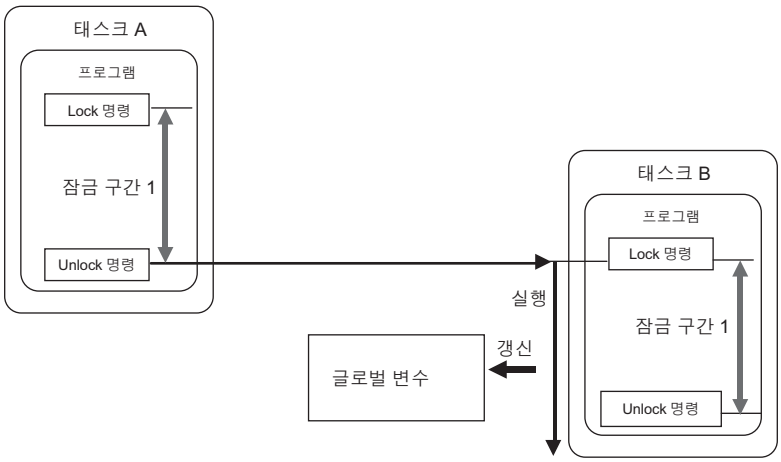
예)

태스크 A, 태스크 B는 모두 잠금 구간 1을 갖는 태스크이며, 태스크 A보다 태스크 B의 우선도가 높습니다.

태스크 A의 잠금 구간 1 실행 중에 태스크 B 실행 조건이 성립하면 태스크 A는 잠금 구간 1의 도중에 일시 정지되고, 태스크 B가 실행됩니다. 그러나 태스크 A에서 잠금 구간 1이 종료되지 않았으므로 태스크 B는 잠금 구간 1의 처리 전에 일시 정지됩니다. 태스크 B가 일시 정지되면 태스크 A의 잠금 구간 1이 재개됩니다.



태스크 A의 잠금 구간 1이 종료되면 태스크 A는 다시 일시 정지되고, 태스크 B의 잠금 구간 1 이후가 실행됩니다. 글로벌 변수의 쓰기 처리를 태스크 사이에서 배타 처리하여 값의 동시성을 확보합니다.



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 세트로 사용하십시오.

태스크 사이에서 동일한 글로벌 변수에 액세스하는 경우의 예

여기에서는 복수의 글로벌 변수를 사용하여 다른 태스크에 처리를 의뢰하는 경우에 대해 설명합니다. 다음 샘플 프로그램에서는 하나의 글로벌 변수 'gReq'를 처리의 배타 플래그로 하여 사용자 프로그램에서 배타 제어합니다. 이와 같은 경우에도 태스크 사이에서 사용하는 복수의 글로벌 변수 'gPar1', 'gPar2', 'gReq'에 액세스하는 로직에 대해 '변수의 태스크 간 배타 제어 기능'에 의한 배타 제어가 필요합니다.

태스크 간 배타 제어 명령(Lock/Unlock 명령)을 사용하여 배타 제어하는 샘플 프로그램은 다음과 같습니다.

- **글로벌 변수**

이름	데이터 형식	코멘트
gReq	BOOL	요구 플래그
gPar1	ULINT	파라메터 1
gPar2	DATA_AND_TIME	파라메터 2

● **처리를 의뢰하는 측의 태스크(MainTask)**

• 내부 변수

이름	데이터 형식	코멘트
ReqTrg	BOOL	요구 트리거
cCnt	ULINT	100ms 카운터값
cTime	DATA_AND_TIME	현재 시각

• ST 프로그램

```
cCnt:=Get100msCnt();                    (*100ms 카운터값의 취득*)
cTime:=GetTime();                       (*현재 시각의 취득*)

Lock(1);                                (*태스크 간 배타 잠금 시작*)
IF ReqTrg=TRUE AND gReq=FALSE THEN(*배타 플래그의 참조*)
  gPar1:=cCnt;                          (*SubTask로 처리하는 파라메터의 설정*)
  gPar2:=cTime;                         (*SubTask로 처리하는 파라메터의 설정*)
  gReq:=TRUE;
  ReqTrg:=FALSE;
END_IF;
Unlock(1);                              (*태스크 간 배타 잠금 해제*)
```

● **처리 의뢰를 받는 측의 태스크(SubTask)**

• 내부 변수

이름	데이터 형식	코멘트
ReqBusy	BOOL	–
UserDefFB_ins	UserDefFB	처리를 실행하는 사용자 정의 FB의 인스턴스

• ST 프로그램

```
Lock(1);                                (*태스크 간 배타 잠금 시작*)
IF gReq=TRUE AND ReqBusy=FALSE THEN(*배타 플래그의 참조*)
  ReqBusy:=TRUE;
  UserDefFB_ins.PutData:=gPar1;         (*MainTask에서 파라메터 읽기*)
  UserDefFB_ins.PutDate:=gPar2;         (*MainTask에서 파라메터 읽기*)
  gReq:=FALSE;                          (*배타 플래그의 리셋*)

  UserDefFB_ins.Execute:=TRUE;
END_IF;
Unlock(1);                              (*태스크 간 배타 잠금 해제*)

UserDefFB_ins();

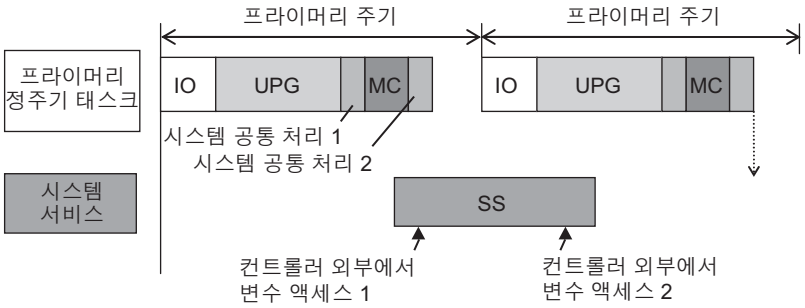
IF UserDefFB_ins.Done:=TRUE THEN
  UserDefFB_ins.Execute:=FALSE;
  ReqBusy:=FALSE;
END_IF;
```

5-8-2 컨트롤러 외부에서의 변수 액세스

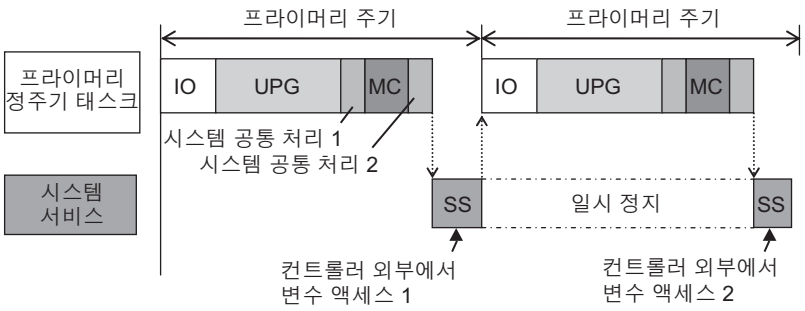
컨트롤러 외부에서의 변수 액세스는 시스템 서비스 내에서 실행됩니다. 시스템 서비스는 태스크 실행보다 낮은 우선도로 실행합니다. 따라서 컨트롤러 외부에서 복수의 변수에 액세스하는 경우에 각 변수 값의 갱신이 동일한 태스크 주기 내에 실행되지 않을 가능성이 있습니다. 사용자 프로그램 내에 값이 갱신된 변수와 갱신되지 않는 변수가 혼재하면 컨트롤러가 예기치 못한 동작을 하는 경우가 있습니다.
이를 막기 위해 컨트롤러 외부에서의 변수 액세스를 태스크의 '시스템 공통 처리 2'에서 실행시키면 복수의 변수 값의 갱신을 확실하게 동일 태스크 주기 내에 실행시킬 수 있습니다.

- **컨트롤러 외부에서의 변수 액세스를 시스템 서비스로 실행하는 경우**
  NX 시리즈 CPU 유니트, NJ 시리즈 CPU 유니트의 어떤 경우에도 복수의 변수 액세스가 동일 태스크 주기로 실행되지 않을 가능성이 있습니다.

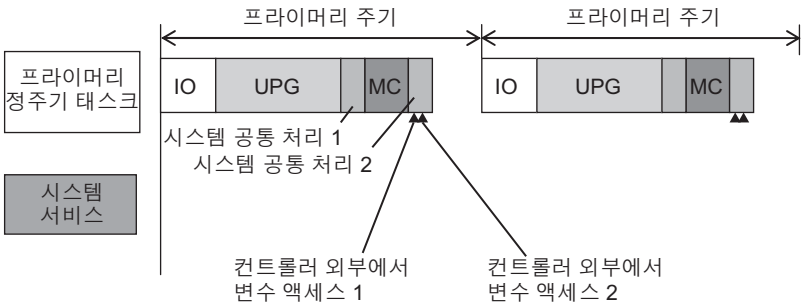
●NX 시리즈 CPU 유니트의 경우



●NJ 시리즈 CPU 유니트의 경우



- **컨트롤러 외부에서의 변수 액세스를 시스템 공통 처리 2에서 실행하는 경우**
  NX 시리즈 CPU 유니트, NJ 시리즈 CPU 유니트의 어떤 경우에도 복수의 변수 액세스가 동일 태스크 주기로 실행됩니다.



여기에서는 컨트롤러 외부에서의 변수 액세스를 태스크의 '시스템 공통 처리 2'로 실행시키는 방법을 설명합니다.

컨트롤러 외부에서의 변수 액세스 방법

컨트롤러 외부에서의 변수 액세스 방법에는 다음 4종류가 있습니다.

- Sysmac Studio
- NS/NA 시리즈 표시기
- EtherNet/IP의 태그 데이터 링크
- 상위 PC 등에서의 CIP 통신 명령

단, 변수 액세스 방법이 Sysmac Studio인 경우에 태스크의 '시스템 공통 처리 2'로 실행시킬 수 있는 것은 값의 갱신 처리뿐입니다. 값의 참조 처리는 시스템 서비스로 실행됩니다.

변수 액세스를 시스템 공통 처리 2로 실행하는 태스크

컨트롤러 외부에서의 변수 액세스를 '시스템 공통 처리 2'로 실행하는 태스크는 변수의 종류에 따라 다음과 같이 정해져 있습니다.

변수	값을 갱신하는 태스크
변수의 태스크 간 배타 제어 설정에서 지정한 글로벌 변수	Sysmac Studio의 [구성·설정]–[태스크 설정]–[변수의 태스크 간 배타 제어 설정]에서 지정한 갱신 태스크
EtherCAT 슬레이브용 디바이스 변수	Sysmac Studio의 [구성·설정]–[태스크 설정]에서 'I/O를 제어하는 태스크 설정'으로 지정한 태스크
CJ 기본 I/O 유니트용 디바이스 변수	
CJ 고기능 유니트용 디바이스 변수	프라이머리 정주기 태스크
CJ 유니트용 메모리에 AT(할당대상) 지정한 변수	

(주의) CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
(주의) CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

변수 액세스를 시스템 공통 처리 2로 실행하기 위한 설정

컨트롤러 외부에서의 변수 액세스를 태스크의 '시스템 공통 처리 2'로 실행하기 위해서는 Sysmac Studio에서 다음과 같은 2가지 설정을 실행해야 합니다.

- 변수의 태스크 간 배타 제어 설정(대상 변수가 글로벌 변수인 경우)
- 변수 액세스 시간 설정

변수의 태스크 간 배타 제어 설정

컨트롤러 외부에서의 변수 액세스를 태스크의 '시스템 공통 처리 2'로 실행하는 대상 변수가 글로벌 변수인 경우에 변수의 태스크 간 배타 제어 설정이 필요합니다. 변수의 태스크 간 배타 제어 기능이란 대상글로벌 변수를 갱신하는 태스크를 지정하면 그 이외의 태스크 및 컨트롤러 외부의 액세스원에서는 대상 글로벌 변수의 값을 갱신할 수 없게 하는 기능입니다.

변수의 태스크 간 배타 제어 기능의 자세한 내용은 📖'변수의 태스크 간 배타 제어 설정(4-10페이지)'을 참조하십시오.

사용상의 주의 사항

EtherNet/IP의 태그 데이터 링크를 사용하는 경우에는 동일한 태그 세트 내의 태그(네트워크 공개 속성의 변수)에 대해 반드시 동일한 태스크를 갱신 태스크로 하십시오. 그렇게 하지 않으면 태그 세트 내의 여러 태그의 갱신 타이밍이 다른 태스크 주기에 걸쳐질 가능성이 있습니다.

변수 액세스 시간 설정

컨트롤러 외부에서의 변수 액세스를 태스크의 '시스템 공통 처리 2'로 실행하면 태스크 실행 시간이 연장될 가능성이 있습니다. 사용자는 Sysmac Studio에서 변수로의 액세스에 걸리는 처리 시간의 상한값을 설정해야 합니다. 변수로의 액세스에 걸리는 처리 시간의 상한값을 '변수 액세스 시간'이라고 합니다.

- **변수로의 액세스 시간 산출 방법**

  변수로의 액세스 시간은 다음 식으로 산출합니다.

  변수로의 액세스 시간[μs] = 변수의 합계 크기[바이트]*a + 변수의 수*b + 액세스 수*c + d

  위 식 중의 상수 a~d의 값은 CPU 유니트의 형식에 따라 다음과 같이 바뀝니다.

CPU 유니트의 형식	상수의 값[μs]			
	a	b	c	d
NX701-□□□□	0.0005	0.033	2.67	7.22
NX102-□□□□	0.0040	0.240	3.27	25.21
NX1P2-□□□□	0.0040	0.240	3.27	25.21
NJ501-□□□□	0.0010	0.490*1	1.41	6.68
NJ301-□□□□	0.0015*2	0.560*3	2.15	7.52
NJ101-□□□□	0.0015	1.070	3.83	10.29

*1. CPU 유니트 Ver.1.02 이하는 '0.58'입니다.
*2. CPU 유니트 Ver.1.02 이하는 '0.0009'입니다.
*3. CPU 유니트 Ver.1.02 이하는 '1.03'입니다.

- **변수 액세스 시간 설정 방법**

  변수 액세스 시간은 Sysmac Studio의 [구성·설정]–[태스크 설정]–[태스크 설정] 화면에서 설정합니다. 설정은 태스크별로 실행하고, 태스크 주기에 대한 비율을 입력합니다. 기본값은 3%입니다. 자세한 설정 방법은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

- **변수 액세스 시간의 설정 예**

  변수 액세스 시간의 설정 예는 다음과 같습니다.
  CPU 유니트 NJ501-□□□□에서 동작하는 태스크에 대해 다음과 같은 3 종류의 컨트롤러 외부에서의 변수 액세스가 있다고 가정합니다.

액세스 번호	변수 액세스원	액세스하는 변수의 합계 크기[바이트]	액세스하는 변수의 수	액세스 수
1	EtherNet/IP의 태그 데이터 링크	600	8	1
2	EtherNet/IP의 태그 데이터 링크	200	4	1
3	CIP 통신 명령	1,000	1	1

  액세스 번호 1의 변수로의 액세스 시간은 산출식에 따라 다음 값이 됩니다.
  액세스 번호 1의 변수로의 액세스 시간=600*0.001+8*0.49+1*1.41+6.68
  =12.61[μs]

  동일하게 각 액세스 번호의 변수에 대한 액세스 시간은 다음 값이 됩니다.

액세스 번호	변수로의 액세스 시간[μs]
1	12.61
2	10.25
3	9.58

1태스크 주기 중에 이러한 액세스 중 어느 하나만 발생하지 않았다면 변수로의 액세스 시간이 가장 긴 액세스 번호 1에 맞춰 변수 액세스 시간을 설정합니다.
액세스 번호 1인 변수로의 액세스 시간이12.61μs이므로 태스크 주기가 500μs라고 하면 변수 액세스 시간은 12.61/500≒3%로 설정합니다.

1태스크 주기 중에 이러한 액세스가1회씩 발생한다고 하면 변수로의 액세스 시간은 산출식에 따라 다음 값이 됩니다.
변수로의 액세스 시간=(600+200+1000)*0.001+(8+4+1)*0.49+(1+1+1)*1.41+6.68
=19.08[μs]

태스크 주기가 500μs라고 하면 변수 액세스 시간은 19.08/500≒4%로 설정합니다.

- **변수 액세스 시간의 설정값보다 실제 액세스 시간이 길어진 경우의 처리**

변수 액세스 시간의 설정값보다 실제 액세스 시간이 길어진 경우에 1태스크 주기 중의 변수 액세스 횟수에 따라 다음과 같이 처리됩니다.
복수의 변수 액세스가 동일한 태스크 주기에 처리되도록 변수 액세스 시간은 충분히 여유 있게 설정하십시오.

1태스크 주기 중의 변수 액세스 횟수	처리
여러 번	변수 액세스 시간의 설정값에 처리되는 분량의 변수 액세스가 실행됩니다. 실행되지 않았던 변수 액세스는 다음 태스크 주기에 실행됩니다. 따라서 복수의 변수 액세스가 동일한 태스크 주기 내에 처리되지 않습니다.
1회	변수 액세스 시간의 설정값을 초과하여 변수 액세스가 실행됩니다. 따라서 태스크 실행 시간이 연장합니다.

5-9 태스크 관련 이상

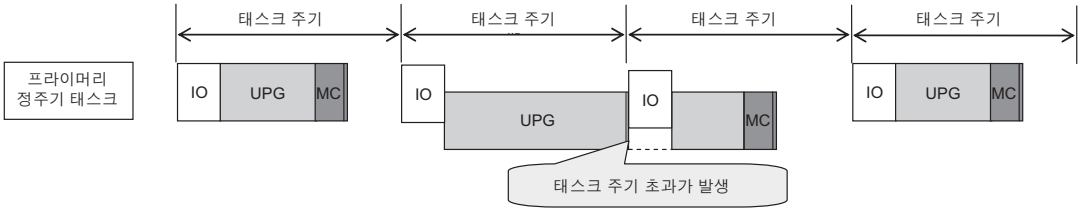
여기에서는 다음과 같은 이상에 대해 설명합니다.

- 태스크 주기 초과
- 모션 제어 주기 초과
- 태스크 실행 타임아웃
- I/O 리프레시 주기 초과
- 시스템 서비스 시간 부족

태스크 주기 초과

태스크 실행 시간이 설정한 태스크 주기를 초과한 경우에 '태스크 주기 초과'가 발생합니다.
가벼운 이상 레벨의 컨트롤러 이상입니다. 에러 발생 시에도 운전은 계속됩니다.
프라이머리 정주기 태스크, 정주기 태스크가 대상입니다.

또한, 설정에 따라 태스크 주기 초과를 발생시키지 않을 수도 있습니다. Sysmac Studio의 '태스크 설정'-'태스크 설정'의 '태스크 주기 초과 감지'에서 설정합니다. 기본값은 '감지'입니다.



이상명	이상 레벨	처리
태스크 주기 초과	가벼운 이상	태스크 설정 및 프로그램을 재검토하고 다시 다운로드합니다. Sysmac Studio에서 이상을 해제하십시오.

'태스크 주기 초과 감지' 설정에 관계없이 주기 내에 태스크 내의 처리가 완료되지 않은 경우에는 시스템 정의 변수의 '_<태스크 이름>_Exceeded'(태스크 주기 초과 플래그), '_<태스크 이름>_ExceedCount'(태스크 주기 초과 횟수), '_ErrSta'(컨트롤러 이상 Status) 및 이벤트 로그에 정보가 출력됩니다.
주기 내에 태스크의 처리가 완료되지 않은 경우의 I/O 리프레시 동작은 입출력의 대상에 따라 다음과 같이 다릅니다.

입출력 대상	주기 내에 태스크의 처리가 완료되지 않은 경우의 I/O 리프레시 동작
EtherCAT 슬레이브*1	출력: 이전의 주기 값을 출력합니다. 입력: I/O 리프레시는 실행하지만 입력 데이터는 실행 중인 사용자 프로그램에 반영되지 않습니다.
CPU 유니트 상의 NX 유니트*2	
내장 I/O*3	
CJ 유니트	태스크 실행이 완료될 때까지 I/O 리프레시를 실행하지 않습니다.

*1. EtherCAT 슬레이브 터미널 내의 NX 유니트를 포함합니다.
*2. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*3. 내장 I/O를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.

사용상의 주의 사항

'태스크 주기 초과'가 발생한 경우에 태스크의 주기 내로 들어가도록 사용자 프로그램을 작성하거나 또는 태스크의 주기를 설정하십시오.

모션 제어 주기 초과

'모션 제어(MC)' 처리가 프라이머리 주기(= 모션 제어 주기) 연속으로 완료되지 않은 경우에 발생합니다. 모션 제어 기능 모듈의 부분 정지 이상 레벨의 컨트롤러 이상이 됩니다. 이 때, '태스크 주기 초과'도 동시에 발생합니다.

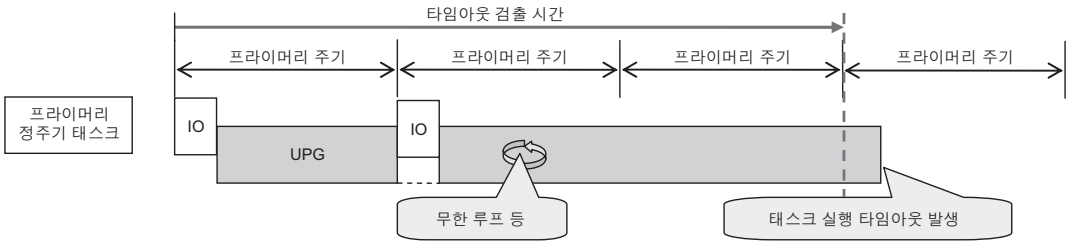
이상명	이상 레벨	처리
모션 제어 주기 초과	부분 정지 이상	프로그램의 처리 내용을 작게 하거나 동작에 문제가 발생하지 않는 범위에서 제어 주기를 더 큰 값으로 설정하십시오.

태스크 실행 타임아웃

지정한 시간('타임아웃 검출 시간')을 경과해도 태스크의 실제 처리가 완료되지 않은 경우에 '태스크 실행 타임아웃'이 발생합니다.

전체 정지 이상 레벨의 컨트롤러 이상입니다. 에러 발생 시 사용자 프로그램의 실행을 정지합니다.

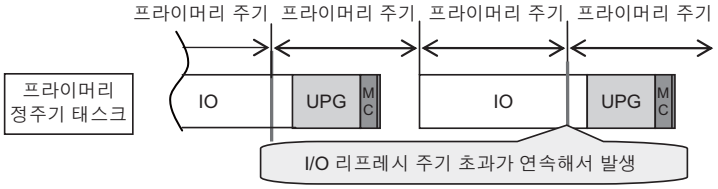
사용자 프로그램 내의 무한 루프 등 로직이 잘못되어 태스크가 정상 동작할 수 없는 경우에도 발생합니다.



이상명	이상 레벨	처리
태스크 실행 타임아웃	전체 정지 이상 레벨	태스크 설정을 재검토하고, 사용자 프로그램을 다시 다운로드합니다. 전원 재투입 또는 리셋 조작이 필요합니다.

I/O 리프레시 주기 초과

프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 5, 16)에서 태스크 주기 내에 I/O 리프레시가 완료하지 않는 상태가 연속하면 'I/O 리프레시 주기 초과'가 발생합니다.
전체 정지 이상 레벨의 컨트롤러 이상입니다. 에러 발생 시 사용자 프로그램의 실행을 정지합니다.



이상명	이상 레벨	처리
I/O 리프레시 주기 초과	전체 정지 이상 레벨	태스크 설정을 재검토하고, 다시 다운로드합니다. 전원 재투입 또는 리셋 조작이 필요합니다.

시스템 서비스 시간 부족

NJ 시리즈 CPU 유니트에서 시스템 서비스 감시 설정 분량의 시스템 서비스 실행 시간이 확보되지 않는 경우에 '시스템 서비스 시간 부족'이 발생합니다.
전체 정지 이상 레벨의 컨트롤러 이상입니다. 에러 발생 시 사용자 프로그램의 실행을 정지합니다.

이상명	이상 레벨	처리
시스템 서비스 시간 부족	전체 정지 이상 레벨	태스크 설정 및 시스템 서비스 감시 설정을 재검토하고, 다시 다운로드합니다. 전원 재투입 또는 리셋 조작이 필요합니다.

참고

NX 시리즈 CPU 유니트에서는 시스템 서비스 실행 시간이 부족하지는 않습니다. 따라서 시스템 서비스 감시 설정은 없습니다. 또한, 시스템 서비스 시간 부족 이상이 발생하는 경우도 없습니다.

5-10 태스크 실행 상태, 태스크 실행 시간의 모니터링

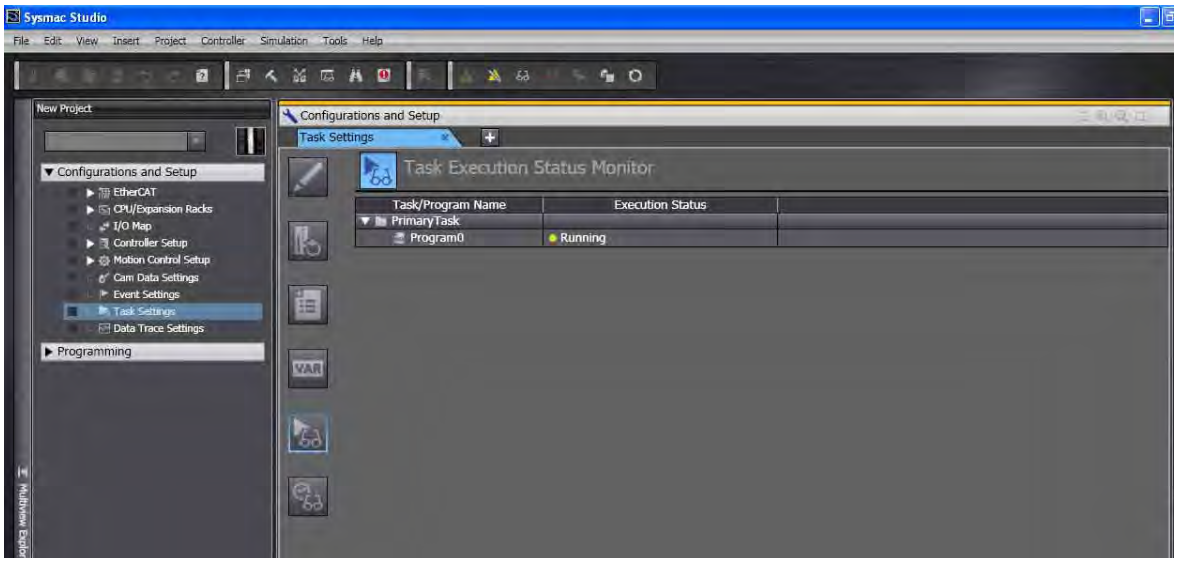
Sysmac Studio에서의 온라인 조작으로 태스크의 실행 상태, 태스크의 실행 시간을 모니터링할 수 있습니다.

태스크의 실행 상태 모니터링

각 태스크 내의 프로그램 실행 상태(기동 중/정지 중)를 Sysmac Studio에서 모니터링할 수 있습니다.

- **Sysmac Studio에서의 조작**

  Sysmac Studio에서 CPU 유니트에 온라인 연결하여 [구성·설정]–[태스크 설정]을 선택합니다. 태스크 실행 상태 모니터링 버튼을 클릭하면 다음과 같은 화면이 표시됩니다.



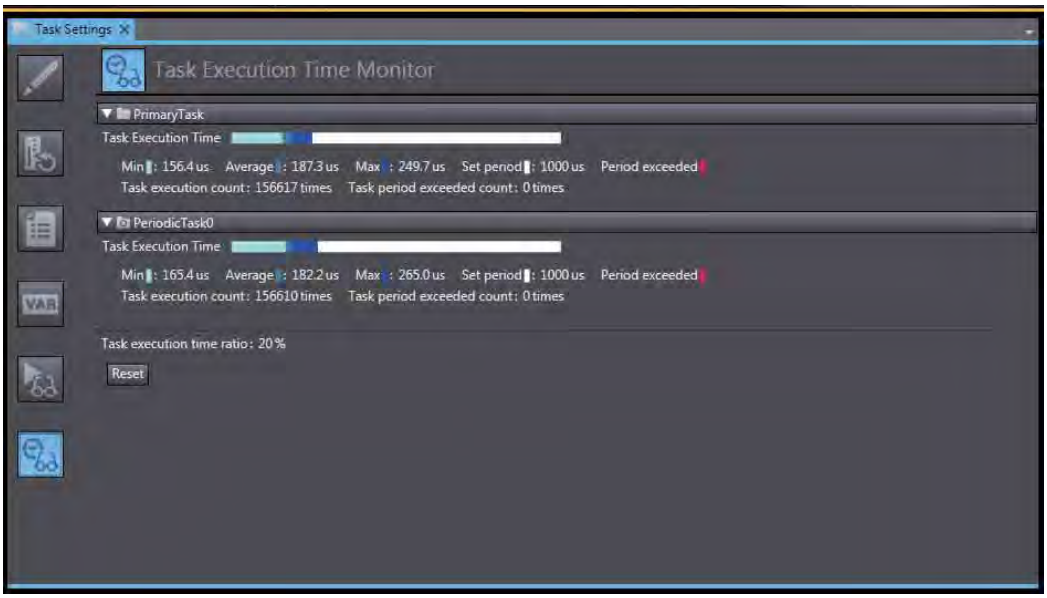
태스크 실행 시간 모니터링

각 태스크의 실행 시간을 Sysmac Studio에서 모니터링할 수 있습니다.

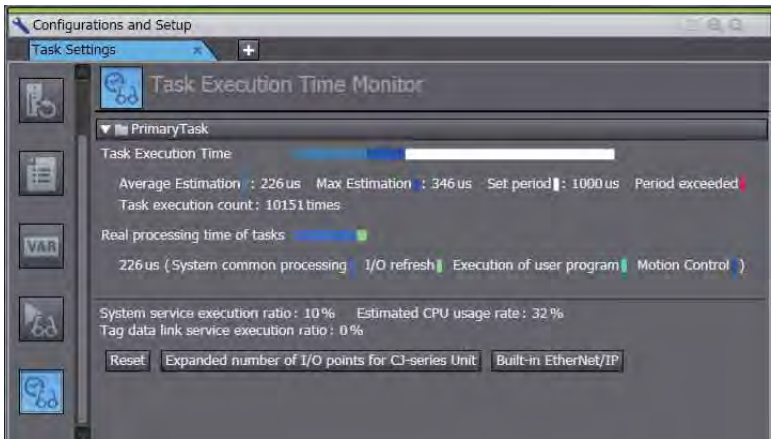
- **Sysmac Studio에서 모니터링할 수 있는 값**

  실제 기기 컨트롤러에 연결할 때와 시뮬레이터에 연결할 때는 화면이 다음과 같이 다릅니다.

  컨트롤러 연결 시



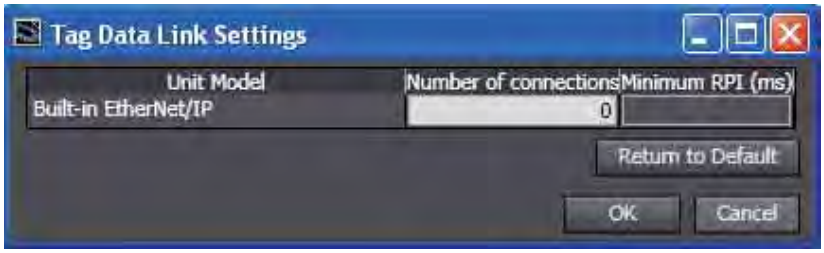
  시뮬레이터 연결 시



참고

NJ 시리즈 CPU 유니트에서는 시스템 서비스 실행 시간을 확인 또는 예측하기 위해 태스크 실행 시간 비율, 시스템 서비스 실행 시간 비율, 예측 CPU 사용율, 태그 데이터 링크 서비스 실행 시간 비율, CJ 유니트 확장 I/O점수, 내장 EtherNet/IP가 표시됩니다.
NX 시리즈 CPU 유니트의 시스템 서비스는 태스크 등의 영향을 받지 않고, 언제라도 실행되므로 위의 표시는 없습니다.

시뮬레이터 연결 내장 EtherNet/IP 화면



설정 항목은 다음과 같습니다.

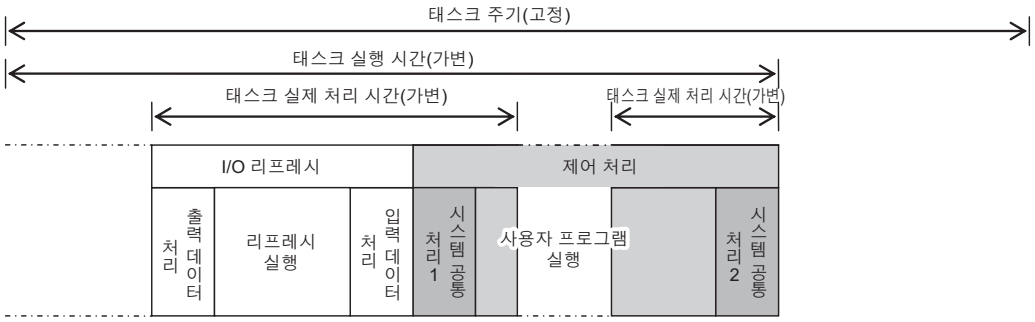
사용하는 포트/ 유니트	설정 항목	기능 설명	설정값	기본값
CPU 유니트 내장 EtherNet/IP 포트	연결 수	태그 데이터 링크의 연결 수입니다.	0~32	0
	최소 RPI	태그 데이터 링크 연결 중 최소 패킷 간격(RPI)의 설정값입니다.	1ms~10s (최소 단위1ms)	–
CJ 시리즈 EtherNet/ IP 유니트 CJ1W-EIP21*1	태그 수	태그 데이터 링크의 태그 수입니다.	0~256	0

*1. 'CJ 유니트 확장 I/O점수' 화면에서 입력합니다.

모니터링할 수 있는 항목은 다음과 같습니다.

모니터링 항목		설명	컨트롤러 연결 시	시뮬레이터 연결 시
태스크 실행 시간[*1]	최소	태스크 실행 시간의 최소값입니다.	있음	없음
	평균	태스크 실행 시간의 평균값입니다.		있음
	최대	태스크 실행 시간의 최대값입니다.		
설정 주기		설정한 태스크 주기입니다.[*2]		
주기 초과		태스크 실행 시간이 태스크 주기를 초과했을 때(시스템 정의 변수의 '태스크 주기 초과 플래그'가 TRUE인 경우)에 그 정도를 위의 바로 나타냅니다. [*2]		
태스크 실행 횟수		태스크의 실행 횟수를 표시합니다. 시스템 정의 변수의 '태스크 실행 횟수'의 값입니다.		
태스크 실제 처리 시간[*3]		시스템 공통 처리, I/O 리프레시, 사용자 프로그램 실행, 모션 제어 처리 시간 비율을 바로 나타냅니다. (시간(수치)은 표시되지 않습니다.)	없음	있음
	예측 평균	태스크 실제 처리 시간의 예측 평균값입니다.		
	예측 최대	태스크 실제 처리 시간의 예측 최대값입니다.		
태스크 실행 시간 비율[*4]		단위 시간당 모든 태스크 실행 시간의 비율입니다.	있음	없음

*1. 태스크의 실행 시작에서 실행 완료까지 실제로 걸린 시간입니다. 실행 시작에서 실행 완료될 때까지 실행된 다른 태스크 및 시스템 서비스의 시간도 포함합니다.
NX701 CPU 유니트에서의 시뮬레이터 연결 시에는 프라이머리 정주기 태스크만 표시됩니다.
*2. 이벤트 태스크인 경우에는 표시되지 않습니다.
*3. 태스크 실행 조건 성립에서 실행 완료까지 해당 태스크 자체에 걸린 시간입니다.
프라이머리 정주기 태스크인 경우에 태스크 실행 시간과 일치합니다.
정주기 태스크인 경우에 다음 그림과 같이 태스크 실행 시간에서(실행 조건 성립에서 실행 완료될 때까지 실행된) 다른 태스크와 시스템 서비스의 시간을 뺀 시간입니다.



*4. Sysmac Studio Ver.1.12에서 추가된 항목입니다. NX 시리즈 CPU 유니트에서는 표시되지 않습니다.

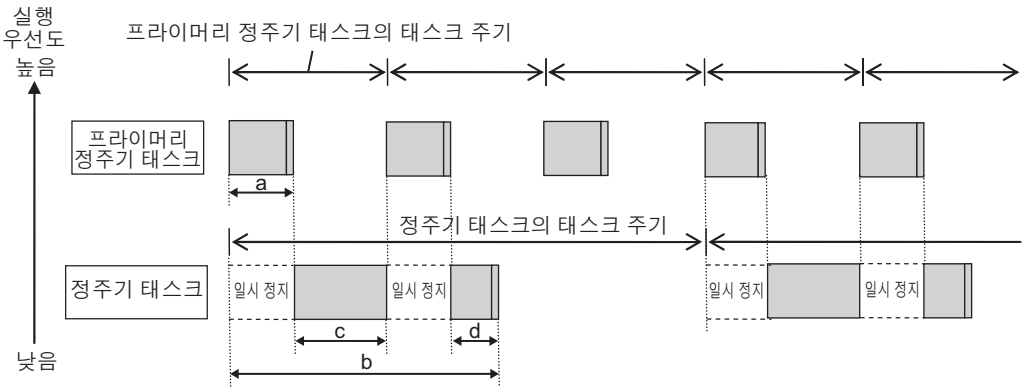
사용상의 주의 사항

Sysmac Studio의 시뮬레이터 연결 시의 위의 값은 실제 기기 컨트롤러 연결 시와 비교하여 오차가 있을 수 있습니다. 따라서 기준으로 활용하십시오. 반드시 실제 기기 컨트롤러 연결에서 확인한 후 설계의 검토 및 본 운전을 실행하도록 하십시오.

태스크 실행 시간과 태스크 실제 처리 시간의 의미

태스크 실행 시간 모니터링에서 표시되는 '태스크 실행 시간', '태스크 실제 처리 시간'의 의미는 다음과 같습니다. '태스크 실제 처리 시간'은 시뮬레이터 연결 시에만 표시됩니다.

표시되는 시간	의미
태스크 실행 시간	태스크 주기의 시작에서 태스크 처리가 종료될 때까지의 시간입니다. 단, 정주기 태스크가 그보다 실행 우선도가 높은 태스크 실행 중에 일시 정지한 시간도 포함합니다.
태스크 실제 처리 시간	태스크 주기의 시작에서 태스크 처리가 종료될 때까지의 시간입니다. 단, 정주기 태스크가 그보다 실행 우선도가 높은 태스크 실행 중에 일시 정지한 시간은 포함하지 않습니다.



a :프라이머리 정주기 태스크의 태스크 실행 시간
b: 정주기 태스크의 태스크 실행 시간
c+d: 정주기 태스크의 태스크 실제 처리 시간(시뮬레이터 연결시에만 해당)

5-11 태스크 설계 방법 및 입출력 응답 시간

여기에서는 태스크 설계의 지침, 태스크 실행 시간 예상 방법, 시스템 서비스 감시 설정 확인, 태스크 설계 예 및 입출력 응답 시간에 대해 설명합니다.

NJ/NX 시리즈 컨트롤러에서는 프라이머리 정주기 태스크 및 정주기 태스크는 설정된 주기에 기초해서 동작합니다.
실제 실행 시간이 설정된 주기를 초과한 경우 각종 에러 처리를 실행합니다.

여기에서는 프라이머리 정주기 태스크 하나(싱글 태스크)로 구성하는 경우를 예로 들어 예상 방법 및 평가 방법을 설명합니다.

안전상의 중요 사항

실제 컨트롤러는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간이 변동됩니다. 최종적으로는 반드시 실제 기기를 사용하여 모든 조건에서 성능 확인을 실시하고, 태스크 주기 초과가 발생하지 않는지, 적절한 통신 응답성이 확보되는지 확인하십시오.

5-11-1 태스크 실행 시간 확인

여기에서 예상한 태스크 실행 시간의 평균값 및 최대값이 설정된 태스크 주기에 충분히 들어가도록 설계하십시오.

- **데스크탑에서 계산하는 경우**

  'A-2 NX701에서의 태스크 실제 처리 시간의 기준 계산(A-19페이지)', 'A-3 NX102에서의 태스크 실제 처리 시간의 기준 계산(A-30페이지)', 'A-4 NX1P2에서의 태스크 실제 처리 시간의 기준 계산(A-40페이지)' 및 'A-5 NJ 시리즈에서의 태스크 실제 처리 시간의 기준 계산(A-50페이지)'을 참조하여 데스크탑에서 '태스크 실행 시간의 평균값'의 개략적인 예상을 실행합니다.
  또한, 데스크탑인 경우에는 최대값을 예상할 수 없습니다

- **Sysmac Studio의 시뮬레이터에서의 계산인 경우**

  Sysmac Studio의 시뮬레이터의 '태스크 실행 시간 모니터링'에서 태스크 실행 시간의 예측값(평균값 및 최대값)를 확인합니다.
  시뮬레이터 상에서 동작을 확인하기 위한 순서는 다음과 같습니다.

  1 유니트/슬레이브 구성의 작성, 글로벌 변수/디바이스 변수 등록, 축(축 변수) 작성을 실행합니다.

  2 검증용 프로그램을 작성합니다.

  3 태스크를 설정하고, 빌드를 실행합니다.

  4 시뮬레이터를 '실행 시간 예측 모드'로 기동합니다.

  5 하드웨어 개정 번호가 표시되는 유니트에서는 해당하는 하드웨어 개정 번호를 선택합니다.

  6 NJ 시리즈 CPU 유니트에서는 [태스크 실행 시간 모니터링]의 [CJ 유니트 확장 I/O점수 설정]에서 특정 CJ 고기능 유니트에서 사용자 정의 변수를 작성하여 AT(할당대상) 지정을 하는 확장 영역(예: DeviceNet 유니트의 고정 할당 I/O 영역)의 크기(OUT 채널 수, IN 채널 수)를 설정합니다.
  이 크기는 이들의 특정 고기능 유니트 사용 시의 I/O 리프레시 시간의 산출에 사용됩니다.

  7 [태스크 실행 시간 모니터링]에서 태스크 실행 시간의 예상을 실행합니다.

  시뮬레이터를 '실행 시간 예측 모드'로 실행 시, Sysmac Studio의 '태스크 실행 시간 모니터링 기능'에서는 다음과 같은 값을 확인할 수 있습니다.
  - 태스크 실행 시간(예측 평균값, 예측 최대값)
  - 태스크 실제 처리 시간(예측 평균값, 예측 최대값)
  - 시스템 공통 처리, I/O 리프레시 시간, 사용자 프로그램 실행 시간, 모션 제어 시간의 내역을 표시한 막대 그래프
  - CPU 사용율

    (주의) NX 시리즈 CPU 유니트에서는 프라이머리 정주기 태스크만 표시됩니다.

참고

Sysmac Studio의 시뮬레이터와 연결한 경우에는 실제 기기 컨트롤러 연결과 달리 다음을 확인할 수 있습니다.
- 'CPU 사용율':
  NJ 시리즈 CPU 유니트의 태스크 처리 시간 예측 최대값, 태그 데이터 링크 서비스 실행 시간 비율, 시스템 서비스 감시 설정으로 지정된 확보해야 하는 시스템 서비스 처리 시간의 합계가 태스크 주기에 대해 어느 정도 점유하고 있는지 표시합니다.
  CPU 사용율이 100%를 초과하면 태스크 처리 시간 및 시스템 서비스 감시 설정값에 대해 충분한 시간이 확보되지 않았다는 것을 의미합니다.
  NX 시리즈 CPU 유니트에서는 표시되지 않습니다.
- '태스크 실제 처리 시간':
  '태스크 실제 처리 시간'이란, 태스크의 실행 시작에서 실행 완료까지의 해당 태스크 자체에 걸린 시간입니다. 태스크의 실행 시작에서 실행 완료될 때까지 실행된 다른 태스크의 실행 시간은 포함되지 않습니다.

- **실제 기기 컨트롤러에서 계산하는 경우**

  실제 기기 컨트롤러 연결 시에 Sysmac Studio의 '태스크 실행 시간 모니터링 기능'으로는 다음 값을 확인할 수 있습니다.
  - 태스크 실행 시간(최소값, 평균값, 최대값)
  - 설정 주기
  - 태스크 실행 횟수
  - 주기 초과 횟수
  - 태스크 실행 시간 비율

  Sysmac Studio에서 표시하는 최대값은 실제 기기 컨트롤러에서 동작한 결과로서의 최대값입니다. 앞에서 설명한 것과 같이 태스크 실행 시간은 실제 기기 컨트롤러의 내부 상태에 따라 변동합니다. 그 결과 여기에서 모니터링한 최대값을 초과한 값이 될 가능성도 있습니다. 모니터링한 값 또는 다음과 같은 계산 결과 중 큰 쪽 값을 최대값의 기준으로 합니다.

NX 시리즈 CPU 유니트의 경우

([태스크 실행 시간의 평균값]+([태스크 실행 시간의 평균값]-[태스크 실행 시간의 최소값]))×1.2+25μs

NJ 시리즈 CPU 유니트의 경우

- 태스크 주기가 500μs인 경우:
  [태스크 실행 시간의 평균값]+([태스크 실행 시간의 평균값]-[태스크 실행 시간의 최소값])+100μs
- 태스크 주기가 1ms, 2ms, 4ms인 경우:
  [태스크 실행 시간의 평균값]+([태스크 실행 시간의 평균값]-[태스크 실행 시간의 최소값])+120μs

다음으로 NJ 시리즈 CPU 유니트인 경우에 시스템 서비스 실행 시간이 확보되었는지 확인합니다. 시스템 서비스 실행 시간이 충분히 확보되지 않은 경우에 Sysmac Studio에서의 온라인 조작 및 표시기 등의 외부 기기와의 통신 응답성이 저하할 가능성이 있습니다. NJ 시리즈 CPU 유니트의 시스템 서비스는 태스크 실행의 비는 시간에 실행됩니다.
다음을 [태스크 실행 시간 비율]의 기준으로 하십시오.

CPU 유니트의 형식	태스크 실행 시간 비율의 기준
NJ501-□□□□	90% 이하
NJ301-□□□□	80% 이하
NJ101-□□□□	70% 이하

버전 관련 정보

- [태스크 실행 시간 비율]은 Ver.1.12 이상의 Sycmac Studio에서 NJ 시리즈 CPU 유니트를 사용한 경우에 표시됩니다.
- CPU 유니트 NJ101-□□□□에는 Sysmac Studio Ver.1.13 이상을 사용하십시오. Sysmac Studio Ver.1.12 이하에서는 NJ101-□□□□을 사용할 수 없습니다.

사용상의 주의 사항

NX 시리즈 CPU 유니트에서는 시스템 서비스 실행 시간이 부족하지는 않습니다.
따라서 [태스크 실행 시간 비율]은 표시되지 않습니다.

참고

'태스크 실행 시간 모니터링 기능'에서 표시되는 태스크 실행 시간의 평균값은 10회분의 태스크 실행 시간의 이동 평균값입니다.

5-11-2 태스크 설계의 예

이 항에서는 프라이머리 정주기 태스크 하나(싱글 태스크)로 구성하는 경우를 예로 들어 설계 순서를 설명합니다.
단, 실제 어플리케이션 및 각종 조건에 따라 설계 순서 및 고려해야 하는 요소가 변화하므로 하나의 참고 사항으로 이해하십시오.

1 **장치 사양에서 필요한 시스템으로서의 각 입출력 응답 시간을 구합니다.**

2 **시스템으로서의 각 입출력 응답 시간에서 프라이머리 정주기 태스크의 '태스크 주기'를 결정합니다.**

3 **'태스크 실행 시간'이 결정한 '태스크 주기' 내에 들어가는지 먼저 확인합니다.**
데스크탑 또는 Sysmac Studio의 시뮬레이터를 사용하여 태스크 실행 시간의 예상 평균값 또는 예상 최대값의 예측합니다.

4 **NJ 시리즈 CPU 유니트에서는 시스템 서비스 시간이 감시 설정 내에 들어가는지 확인합니다.**
Sysmac Studio의 시뮬레이터인 경우에는 CPU 사용율을 확인합니다.

5 **실제 기기 컨트롤러에서 태스크 실행 시간이 태스크 주기 내에 들어가는지 검토합니다.**
Sysmac Studio를 실제 기기 컨트롤러에 온라인 연결하여 '태스크 실행 시간 모니터링 기능'에 의해 태스크 실행 시간을 검증합니다.

- **프라이머리 정주기 태스크 하나(싱글 태스크)만 설정한 태스크 주기에 들어가지 않는 경우에 다음 관점에서 정주기 태스크에 대한 태스크 분할을 검토합니다.**
  - NX701 CPU 유니트인 경우에 장치 제어에 필요한 처리 중 고속으로 제어하는 처리를 프라이머리 정주기 태스크에 그 밖의 처리를 정주기 태스크(실행 우선도 5)에 분할합니다.
  - 태스크 실행 시간을 단축하기 위해 프로그램 실행 명령(PrgStart) 명령 및 프로그램 정지 명령(PrgStop) 명령을 사용하고, 프라이머리 정주기 태스크에 할당한 프로그램을 필요할 때만 실행합니다.
  - 이를 실시해도 프라이머리 정주기 태스크의 태스크 주기에 들어가지 않는 경우에 프라이머리 정주기 태스크의 처리 중 고속, 고정밀도가 필요하지 않은 처리 부분을 정주기 태스크(실행 우선도 16) 또는 정주기 태스크(실행 우선도 17)로 분할합니다.
    단, 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16) 사이에서 축 변수 등을 참조하는 경우에 프라이머리 정주기 태스크의 태스크 실행 시간이 길어지는 경우가 있습니다. 자세한 내용은 "NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.
  - 이를 실행해도 프라이머리 정주기 태스크의 태스크 주기에 들어가지 않는 경우에는 프라이머리 정주기 태스크의 태스크 주기를 재검토하십시오.

- **위에서 태스크를 분할한 경우에 정주기 태스크는 프라이머리 정주기 태스크의 비는 시간에 따라 크게 변동합니다.**
  정주기 태스크인 경우에 산출한 태스크 실행 시간의 예측 평균값/최대값의 값을 2배로 한 값을 정주기 태스크의 태스크 주기로 설정하고, 그 후 조정을 실행하십시오.

5-11-3 시스템으로서의 입출력 응답 시간

시스템으로서 입력한 후 출력할 때까지의 시간은 다음과 같습니다.
사례별로 다릅니다.
또한, 실제로는 이 시간에 슬레이브/유니트의 외부와의 입력 응답 시간, 외부와의 출력 응답 시간이 더해집니다.

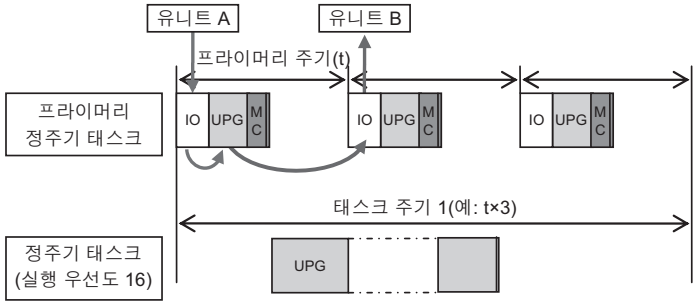
기본 I/O 유니트를 시퀀스 제어하는 경우

기본 I/O 유니트는 I/O 리프레시를 할당한 태스크로 외부와 I/O 리프레시됩니다.
EtherCAT 통신을 포함하는 입출력 응답 시간은 다음과 같습니다.

- **프라이머리 정주기 태스크의 사용자 프로그램으로 제어하는 경우**
  다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '프라이머리 주기'

예) 유니트 A, 유니트 B를 프라이머리 정주기 태스크로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

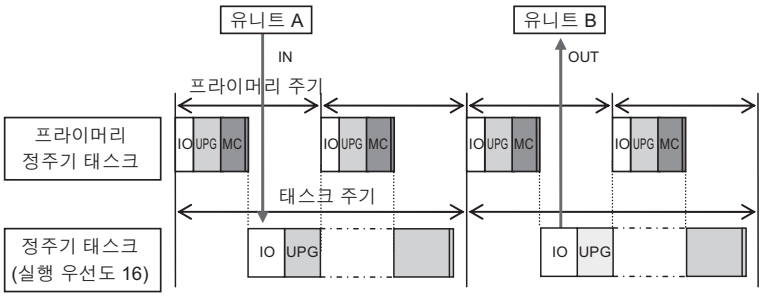
단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '프라이머리 주기'×2

- **정주기 태스크(실행 우선도 16)의 사용자 프로그램에서 제어하는 경우**
  다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'

예) 유니트 A, 유니트 B를 정주기 태스크(실행 우선도 16)로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'×2

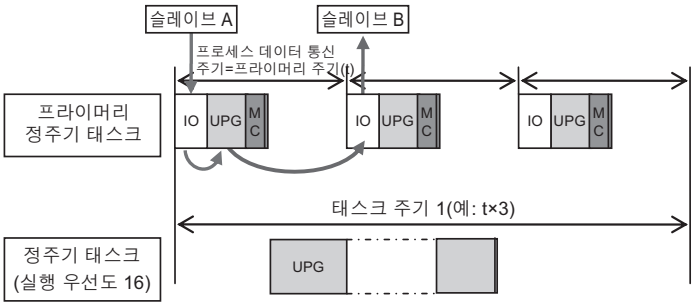
EtherCAT 슬레이브를 시퀀스 제어하는 경우

EtherCAT 슬레이브는 프라이머리 정주기 태스크의 I/O 리프레시로 외부와 EtherCAT 통신됩니다.
EtherCAT 통신을 포함하는 입출력 응답 시간은 다음과 같습니다.

- **프라이머리 정주기 태스크의 사용자 프로그램으로 제어하는 경우**
  다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '프라이머리 주기(= 프로세스 데이터 통신 주기)'

예) EtherCAT IN 슬레이브 A, EtherCAT OUT 슬레이브 B를 프라이머리 정주기 태스크로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

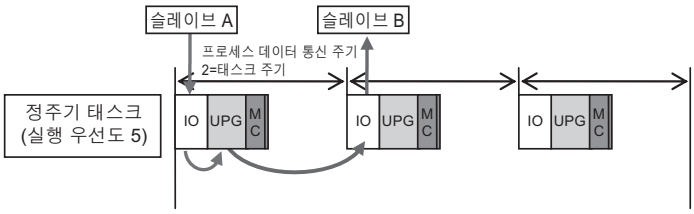
최대 입출력 응답 시간 = '프라이머리 주기(= 프로세스 데이터 통신 주기)'×2

- **정주기 태스크(실행 우선도 5)의 사용자 프로그램에서 제어하는 경우**

  NX701 CPU 유니트의 정주기 태스크(실행 우선도 5)의 사용자 프로그램에서 제어하는 경우에 다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '정주기 태스크(실행 우선도 5)의 주기(= 프로세스 데이터 통신 주기 2)'

  예) EtherCAT IN 슬레이브 A, EtherCAT OUT 슬레이브 B를 정주기 태스크(실행 우선도 5)로 제어하는 경우



  주의. 상기 그림은 1회 입출력만을 기술합니다.

  단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

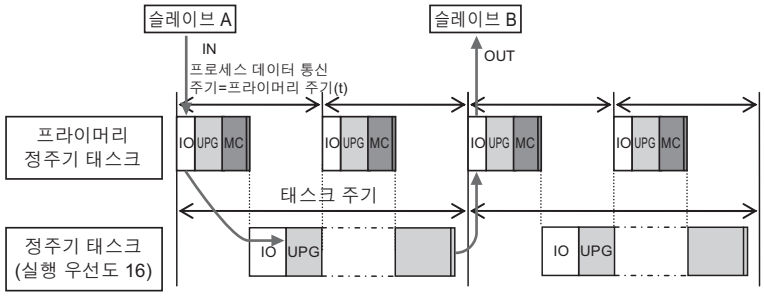
최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 5)의 주기(= 프로세스 데이터 통신 주기 2)'×2

- **정주기 태스크(실행 우선도 16)의 사용자 프로그램에서 제어하는 경우**

  다음과 같은 입출력 응답 시간에 응답합니다.

입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'

  예) EtherCAT IN 슬레이브 A, EtherCAT OUT 슬레이브 B를 정주기 태스크(실행 우선도 16)로 제어하는 경우



  주의. 상기 그림은 1회 입출력만을 기술합니다.

  단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'×2

참고

EtherCAT 커플러 유니트에 NX 유니트를 장착한 EtherCAT 슬레이브 터미널의 입출력 응답 시간에 대해서는 📖"NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

CPU 유니트 상의 NX 유니트를 시퀀스 제어하는 경우

NX102 및 NX1P2 CPU 유니트 상의 NX 유니트는 프라이머리 정주기 태스크의 I/O 리프레시로 외부와 데이터를 교환합니다.

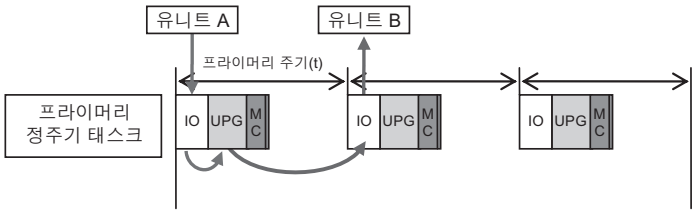
CPU 유니트 상의 NX 유니트에 대해서는 CPU 유니트 내부뿐의 입출력 응답 시간 및 NX 유니트 측의 처리를 포함한 출력 응답 시간의 양측에 대해 설명합니다.

CPU 유니트 내부만의 입출력 응답 시간은 다음과 같습니다.

- **프라이머리 정주기 태스크의 사용자 프로그램으로 제어하는 경우**
  다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '프라이머리 주기'

예) 입력 유니트 A, 출력 유니트 B를 프라이머리 정주기 태스크로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

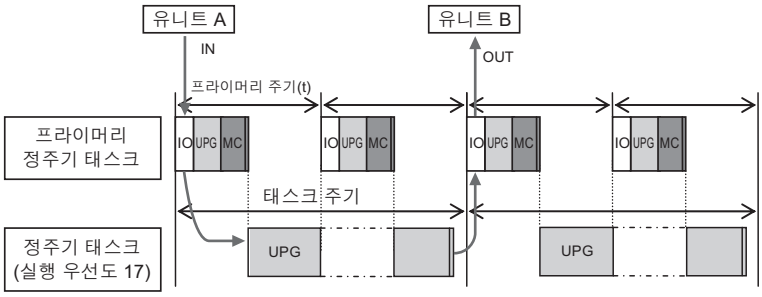
단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '프라이머리 주기'×2

- **정주기 태스크(실행 우선도 17)의 사용자 프로그램에서 제어하는 경우**
  다음과 같은 입출력 응답 시간에 응답합니다.

입출력 응답 시간 = '정주기 태스크(실행 우선도 17)의 주기'

예) 입력 유니트 A, 출력 유니트 B를 정주기 태스크(실행 우선도 17)에서 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 17)의 주기'×2

다음으로 CPU 유니트 상의 NX 유니트 측의 처리를 포함한 출력 응답 시간에 대해 설명합니다.
외부에서 NX 유니트에 입력된 신호를 CPU 유니트에서 연산 처리를 실행하고, 그 결과를 NX 유니트에서 외부에 출력할 때까지의 입출력 응답 시간입니다.

NX 유니트 측의 처리를 포함한 출력 응답 시간은 NX 유니트의 I/O 리프레시 방식에 좌우됩니다. NX 유니트의 I/O 리프레시 방식이 입출력 동기 리프레시 방식, 타임 스탬프 방식 및 프리 런 리프레시 방식의 각각에 대해 프라이머리 정주기 태스크의 사용자 프로그램으로 제어하는 경우를 설명합니다.

계산식의 각 요소는 다음과 같습니다.

Tnx-InProc	: NX 유니트 고유의 입력 데이터 처리 시간
Tnx-OutProc	: NX 유니트 고유의 출력 데이터 처리 시간
Tnx-Indelay	: NX 유니트 고유의 입력 지연 시간
Tnx-Outdelay	: NX 유니트 고유의 출력 지연 시간

각 요소는 NX 유니트별로 고유의 값이 됩니다. 각 요소의 값에 대해서는 "NX 시리즈 데이터 참조 설명서(SBCA-410G 이상)"의 부록을 참조하십시오.

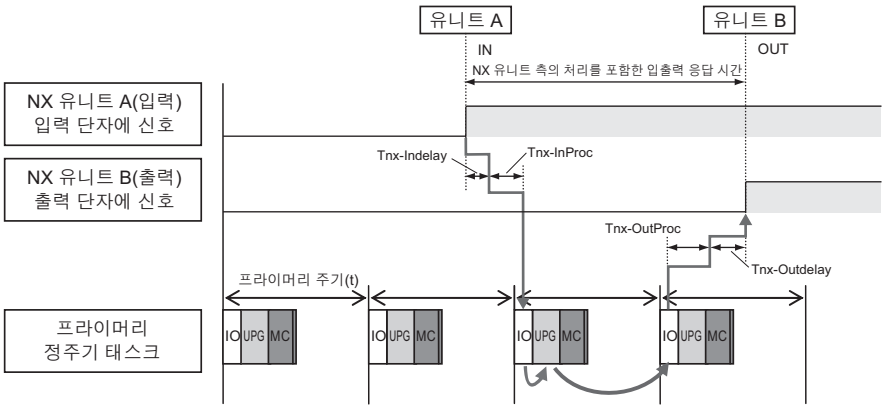
- **입출력 동기 리프레시 방식인 경우**

  입출력 동기 리프레시 방식에서는 CPU 유니트 상에 복수의 NX 유니트를 연결한 경우에 모든 입출력 동기 리프레시 방식의 NX 유니트가 동시에 입력 전송 및 출력 갱신을 실행합니다. 따라서 Tnx-InProc(NX 유니트 고유의 입력 데이터 처리 시간)과 Tnx-OutProc(NX 유니트 고유의 출력 데이터 처리 시간)은 입출력 동기 리프레시 방식의 NX 유니트 중 각각 NX 유니트의 최대값을 적용합니다.

  다음과 같은 입출력 응답 시간에 응답합니다.

  최소 입출력 응답 시간 = Tnx-InDelay+Tnx-InProc+'프라이머리 주기'+Tnx-OutDelay+Tnx-OutProc

  예) 입력 유니트 A, 출력 유니트 B를 프라이머리 정주기 태스크로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

  단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

  최대 입출력 응답 시간 = Tnx-InDelay + Tnx-InProc + '프라이머리 주기'×2
  +Tnx-OutDelay+Tnx-OutProc+NX 유니트 처리 시간*1

*1. NX 유니트 처리 시간에 대해서는 'NX 유니트 처리 시간(A-42페이지)'을 참조하십시오.

● **타임 스탬프 방식인 경우**

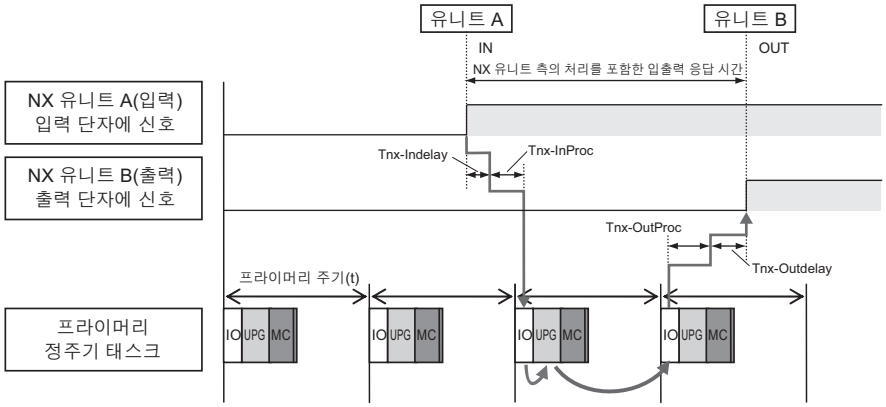
타임 스탬프 방식의 입출력 응답 시간은 입력의 변화 시각에서 일정 시간 후 출력할 때까지의 시간입니다. 일정 시간은 사용자 프로그램에서 지정합니다. 이 때, 일정 시간은 의도한 시각에 출력할 수 있는 시간을 설정해야 합니다.
여기에서는 의도한 시각에 출력할 수 있는 최단의 일정 시간을 타임 스탬프 방식의 최대 입출력 응답 시간이라고 정의하고 설명합니다.

타임 스탬프 방식의 변화 시각 입력 방식에서는 CPU 유니트 상에 복수의 NX 유니트를 연결한 경우에 모든 입력 동기 리프레시 방식의 NX 유니트가 입력을 모든 변화 시각 입력 방식의 NX 유니트가 변화 시각을 동시에 전송합니다. 따라서 Tnx-InProc(NX 유니트 고유의 입력 데이터 처리 시간)과 Tnx-OutProc(NX 유니트 고유의 출력 데이터 처리 시간)은 입출력 동기 리프레시 방식의 NX 유니트 중 각각 NX 유니트의 최대값을 적용합니다.

다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = Tnx-InDelay+Tnx-InProc+'프라이머리 주기'+Tnx-OutDelay+Tnx-OutProc

예) 입력 유니트 A, 출력 유니트 B를 프라이머리 정주기 태스크로 제어하는 경우



주의. 상기 그림은 1회 입출력만을 기술합니다.

단, 유니트에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = Tnx-InDelay + Tnx-InProc + '프라이머리 주기'×2 +Tnx-OutDelay+Tnx-OutProc+NX 유니트 처리 시간*1

*1. NX 유니트 처리 시간에 대해서는 'NX 유니트 처리 시간(A-42페이지)'을 참조하십시오.

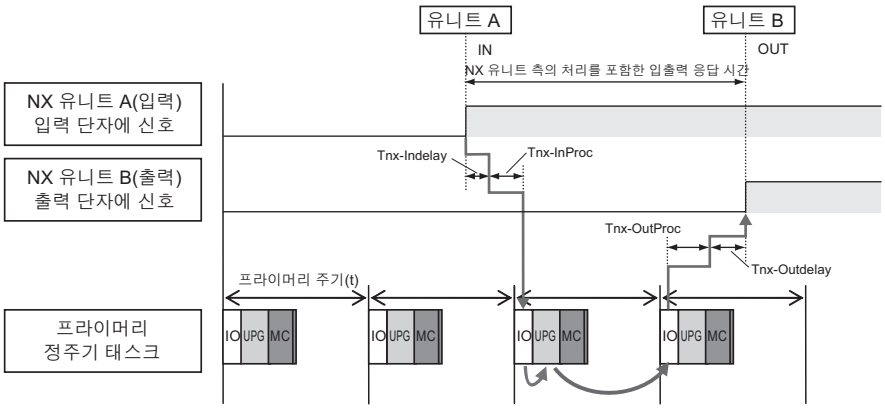
- **프리 런 리프레시 방식인 경우**

  프리 런 리프레시 방식에서는 NX 버스의 리프레시 주기와 NX 유니트의 입출력 갱신 주기가 비동기로 동작합니다.

  다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = Tnx-InDelay+Tnx-InProc+'프라이머리 주기'+Tnx-OutDelay+Tnx-OutProc

  예) 입력 유니트 A, 출력 유니트 B를 프라이머리 정주기 태스크로 제어하는 경우



  주의. 상기 그림은 1회 입출력만을 기술합니다.

  단, 유니트에서의 입력 타이밍 및 유니트에 대한 출력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = Tnx-InDelay + Tnx-InProc×2 + '프라이머리 주기'×2 + Tnx-OutDelay + Tnx-OutProc×2 + NX 유니트 처리 시간[*1]

  *1. NX 유니트 처리 시간에 대해서는 'NX 유니트 처리 시간(A-42페이지)'을 참조하십시오.

참고

프리 런 리프레시 방식에서는 입력 유니트, CPU 유니트 및 출력 유니트가 각각 비동기로 동작합니다. 따라서 입력 유니트가 입력 단자의 신호를 전송하는 타이밍 오차, CPU 유니트가 입력 유니트에서 IN 데이터를 전송하는 타이밍 오차 및 출력 유니트가 출력 단자에 신호를 출력하는 타이밍 오차가 발생합니다.
타이밍 오차는 각각 최대이며, Tnx-InProc, 프라이머리 주기 및 Tnx-OutProc입니다.

모션 제어 명령에서 모션 제어하는 경우

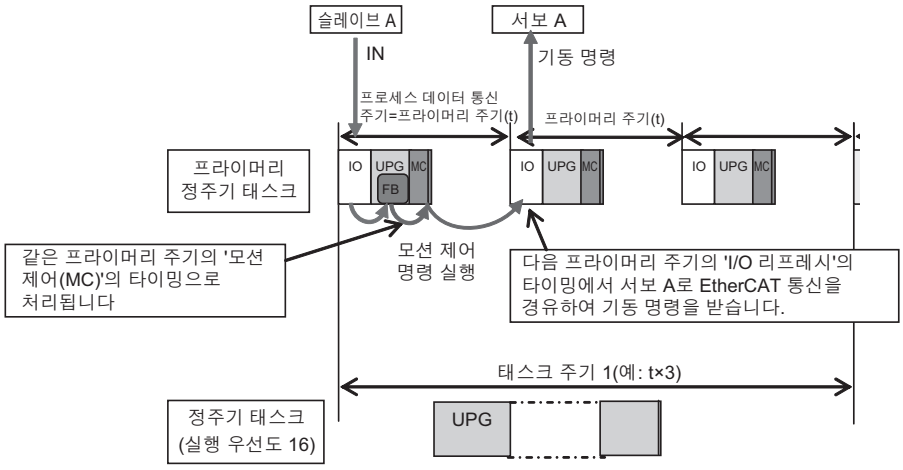
모션 제어 명령은 [축]에 할당된 서보/엔코더 입력 슬레이브에 대해 액세스합니다.
NX102 및 NX1P2 CPU 유니트의 모션 제어 명령은 프라이머리 정주기 태스크에 기술할 수 있습니다.
모션 제어 명령은 프라이머리 정주기 태스크 내의 '모션 제어(MC)'에서 실행됩니다.
NJ 시리즈 CPU 유니트의 모션 제어 명령은 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 기술할 수 있습니다.
어떤 경우에도 모션 제어 명령은 프라이머리 정주기 태스크 내의 '모션 제어(MC)'에서 실행됩니다.
NX701 CPU 유니트의 모션 제어 명령은 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16) 및 정주기 태스크(실행 우선도 5)에 기술할 수 있습니다.
프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 기술한 모션 제어 명령은 프라이머리 정주기 태스크 내의 '모션 제어(MC)'에서 실행됩니다.
정주기 태스크(실행 우선도 5)에 기술한 모션 제어 명령은 정주기 태스크(실행 우선도 5) 내의 '모션 제어(MC)'에서 실행됩니다.

EtherCAT 통신을 포함하는 입출력 응답 시간은 다음과 같습니다.

- **모션 제어 명령을 프라이머리 정주기 태스크에 기술한 경우**

  모션 제어 명령의 처리는 그 프라이머리 정주기 태스크 내의 직후의 '모션 제어(MC)'에서 실행됩니다.
  처리 결과는 그 다음 프라이머리 정주기 태스크의 'I/O 리프레시' 타이밍에 [축]에 할당된 EtherCAT 슬레이브 및 NX 유니트에 출력됩니다.
  다음과 같은 입출력 응답 시간에 응답합니다.

입출력 응답 시간 = '프라이머리 주기(= 프로세스 데이터 통신 주기)'



주의. 상기 그림은 1회 입출력만을 기술합니다.

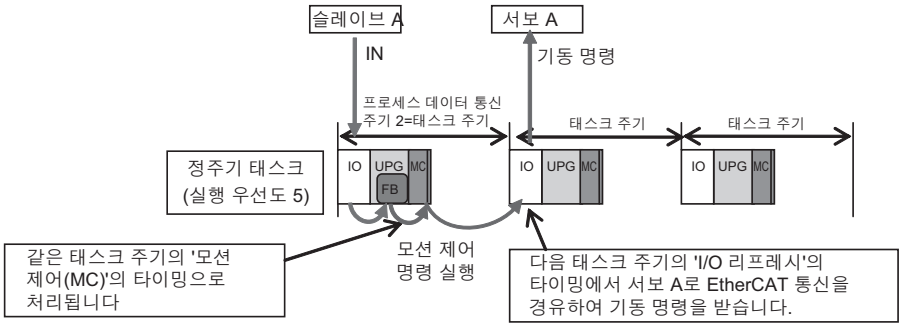
단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '프라이머리 주기(= 프로세스 데이터 통신 주기)'×2

- **모션 제어 명령을 정주기 태스크(실행 우선도 5)에 기술한 경우**

  모션 제어 명령의 처리는 그 정주기 태스크(실행 우선도 5) 내의 직후의 '모션 제어(MC)'에서 실행됩니다. 처리 결과는 그 다음 정주기 태스크(실행 우선도 5)의 'I/O 리프레시' 타이밍에 [축]에 할당된 EtherCAT 슬레이브에 출력됩니다.
  다음과 같은 입출력 응답 시간에 응답합니다.

입출력 응답 시간 = '정주기 태스크(실행 우선도 5)의 주기(= 프로세스 데이터 통신 주기)'



주의. 상기 그림은 1회 입출력만을 기술합니다.

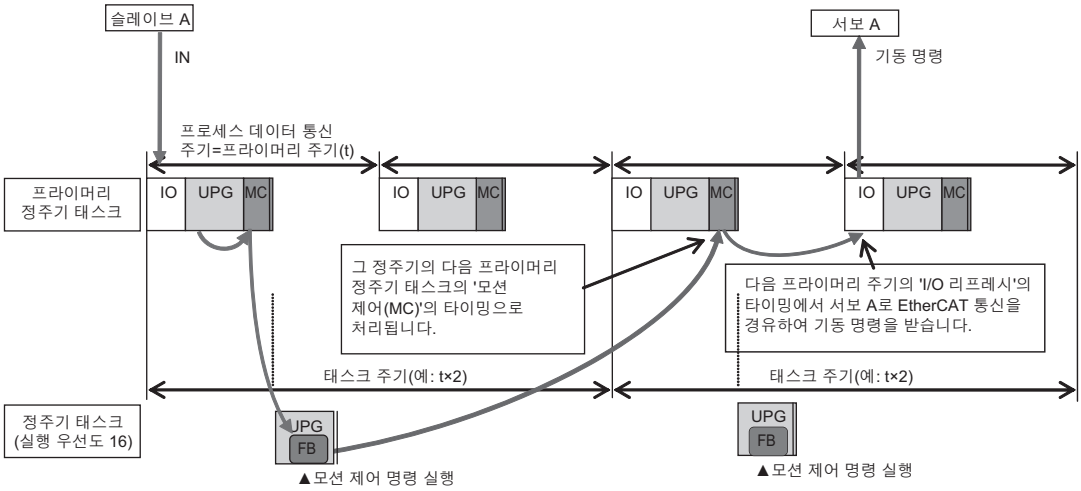
단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 입출력 응답 시간이 됩니다.

최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 5)의 주기(= 프로세스 데이터 통신 주기 2)'×2

- **모션 제어 명령을 정주기 태스크(실행 우선도 16)에 기술한 경우**

  모션 제어 명령의 처리는 해당 정주기 태스크(실행 우선도 16)의 다음 프라이머리 정주기 태스크 내의 '모션 제어(MC)'에서 실행됩니다. 처리 결과는 그 다음 프라이머리 정주기 태스크의 'I/O 리프레시' 타이밍에 [축]에 할당된 EtherCAT 슬레이브에 출력됩니다.
  따라서 모션 제어 명령의 실행 타이밍에 관계없이 다음과 같은 입출력 응답 시간에 응답합니다.

최소 입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'+ '프라이머리 주기(= 프로세스 데이터 통신 주기)'



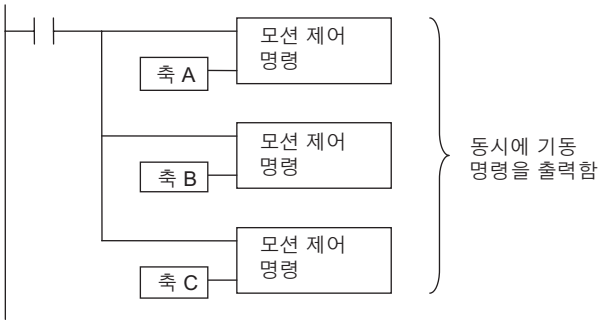
주의. 상기 그림은 1회 입출력만을 기술합니다.

단, 슬레이브에서의 입력 타이밍에 따라서는 다음과 같은 응답 시간이 됩니다.

최대 입출력 응답 시간 = '정주기 태스크(실행 우선도 16)의 주기'+ '프라이머리 주기(= 프로세스 데이터 통신 주기)'×2

- **복수 축의 동시 기동**

  동일한 태스크 내의 사용자 프로그램에서 동일한 태스크 주기에 다축을 기동한 경우에 동시에 기동할 수 있습니다.



참고

축을 제어하는 태스크 이외의 태스크에서 축 변수의 값을 참조할 수 있습니다.
자세한 사용 방법 및 주의사항에 대해서는 "NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어 편(SBCE-363)"을 참조하십시오.

6

프로그래밍

이 장에서는 프로그램의 종류 및 프로그램을 구성하는 변수 및 명령 등 프로그램의 자세한 내용에 대해 설명합니다.

6-1 프로그래밍 순서의 개요 6-3

6-2 POU(프로그램 구성 단위) 6-5

6-2-1 POU의 개념 6-5
6-2-2 POU의 3가지 타입의 개요 6-6
6-2-3 프로그램, 펑션 블록, 펑션의 차이점 6-7
6-2-4 프로그램의 세부 내용 6-8
6-2-5 펑션 블록(FB)의 세부 내용 6-9
6-2-6 펑션(FUN)의 세부 내용 6-18
6-2-7 펑션 블록(FB), 펑션(FUN) 공통 동작 6-24
6-2-8 POU의 제한 사항 6-26

6-3 변수 6-28

6-3-1 변수의 개념 6-28
6-3-2 변수의 종류 6-28
6-3-3 POU에서 본 사용자 정의 변수의 종류 6-29
6-3-4 변수의 속성 6-30
6-3-5 데이터 형식 6-31
6-3-6 파생 데이터 형식 6-41
6-3-7 데이터 형식의 배열/범위 지정 6-51
6-3-8 변수의 각 속성의 설명 6-59
6-3-9 상태 변화에 따른 변수 값의 변화 6-67
6-3-10 펑션 블록(FB)의 인스턴스 6-82
6-3-11 변수 값 모니터링 6-82
6-3-12 변수명 등의 프로그램 관련 이름의 제한 6-83

6-4 상수(리터럴) 6-85

6-4-1 상수의 개념 6-85
6-4-2 각 데이터 형식의 기술 형식 6-85

6-5 프로그램 언어 6-90

6-5-1 프로그램 언어의 개념 6-90
6-5-2 래더도 언어(LD) 6-90
6-5-3 ST 언어 6-96

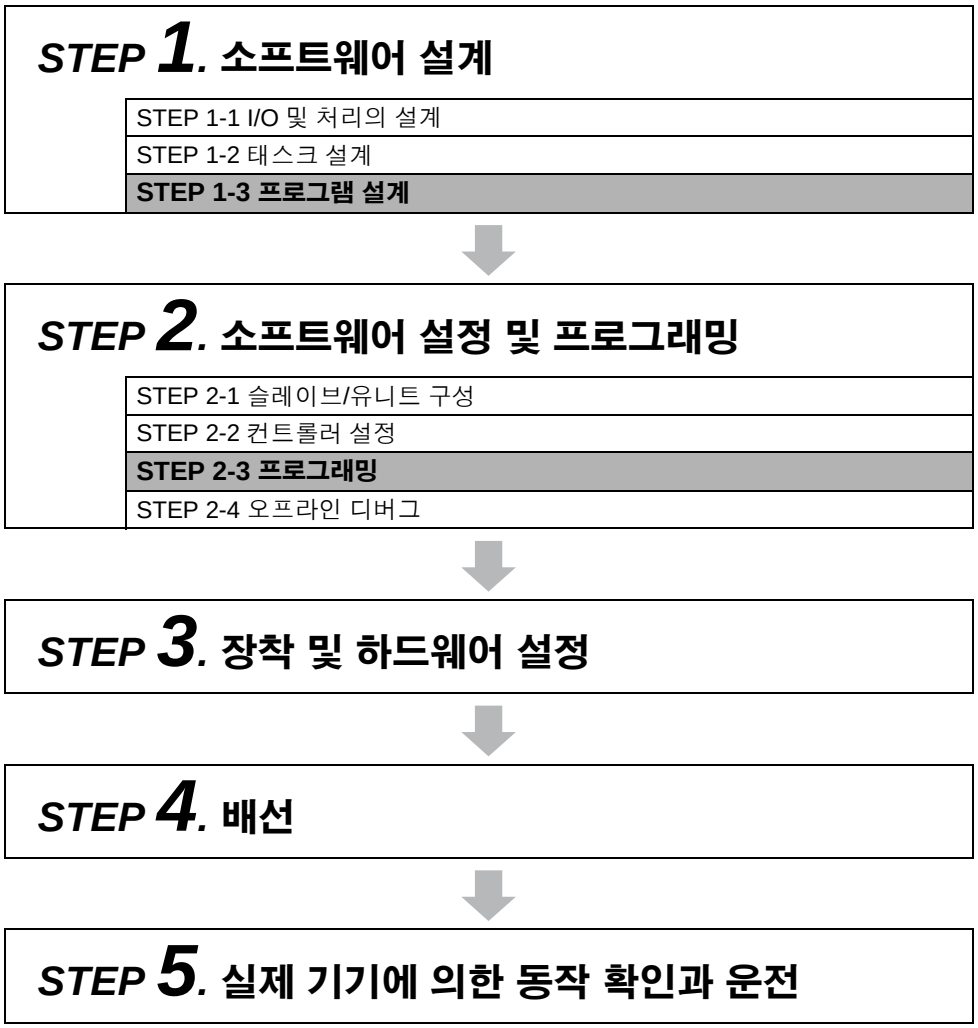
6-6 명령 6-134

6-6-1 명령의 개념 6-134
6-6-2 명령의 기본 지식 6-134
6-6-3 명령 에러 6-137

6-7 이름 공간 6-143
6-7-1 이름 공간의 개념 6-143
6-7-2 이름 공간의 사양 6-144
6-7-3 이름 공간 사용 순서 6-147
6-8 라이브러리 6-148
6-8-1 라이브러리의 개요 6-148
6-8-2 라이브러리 사양 6-149
6-8-3 라이브러리 부품의 사양 6-150
6-8-4 라이브러리 사용 순서 6-151
6-9 사용자 프로그램 작성상의 유의 사항 6-152
6-9-1 입력 변수/출력 변수/입출력 변수가 배열 지정 변수인 경우 6-152
6-9-2 입력 변수/출력 변수/입출력 변수가 구조체 형식 변수인 경우 6-152
6-9-3 마스터 컨트롤 6-153

6-1 프로그래밍 순서의 개요

여기에서는 프로그래밍 순서의 개요를 나타냅니다.
전체 순서 중에서 다음과 같이 회색으로 표시한 위치에 대응합니다.



자세한 내용은 📖'1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.

POU(처리의 분할 단위) 설계	참조
● **각각의 처리를 어느 POU로 실행시킬지 검토하여 설계** 주의: 펑션에는 펑션 블록 형식 명령, 펑션 블록을 기술할 수 없습니다	'6-2 POU(프로그램 구성 단위)(6-5페이지)'
● **어느 처리를 래더도, ST, 인라인 ST 중 어느 언어로 작성할지 설계** 주의: '인라인 ST'는 래더도 내에 ST 언어를 기술하는 래더도 언어의 요소	'6-5 프로그램 언어(6-90페이지)'



변수 설계	참조
● **어떤 사용자 정의 변수를 작성할지 설계**	'6-3-1 변수의 개념(6-28페이지)' '6-3-2 변수의 종류(6-28페이지)'
● **POU 공통으로 사용하는 변수(글로벌 변수)와 각 POU에서만 사용하는 변수(로컬 변수)를 분류**	'6-3-3 POU에서 본 사용자 정의 변수의 종류(6-29페이지)'
● **슬레이브/유니트에 액세스하기 위한 디바이스 변수의 변수명을 자동 생성할지, 사용자가 정의할지 설계**	'3-3 I/O 포트와 디바이스 변수(3-7페이지)'
● **변수의 다음과 같은 각 속성을 설계** 변수명, 데이터 형식, AT(할당대상) 지정, 초기값, 유지, 상수, 네트워크 공개 등 변수의 데이터 형식(배열 지정, 범위 지정, 구조체 형식, 열거 형식)을 설계	'6-3-4 변수의 속성(6-30페이지)' '6-3-5 데이터 형식(6-31페이지)' '6-3-6 파생 데이터 형식(6-41페이지)'
● **변수 설계 시에는 다음 사항에 주의하여 설계하십시오.** • 유지 설계: 전원 투입 시, 동작 모드 변경 시 등에 변수를 어느 값으로 하는지에 따라 유지 속성을 설계합니다. • 구조체 형식 설계: 사용하는 명령의 변수에서 구조체 데이터 형식이 사용되는 경우에 입력, 출력 또는 입출력 파라메터용으로 해당 구조체 데이터 형식을 사용하도록 설계합니다. 예) 통신 명령 • 배열 지정 설계: 사용하는 명령의 변수에서 배열 지정 변수가 사용되는 경우에 입력, 출력 또는 입출력 파라메터용으로 배열 지정 변수를 설계합니다. 예) 시프트 명령, 스택/테이블 명령 • AT 지정 설계: • 일부 명령의 입력 파라메터용 변수에 AT 지정 예) DeviceNet 유니트의 고정 할당 또는 자유 할당인 I/O 할당 • 네트워크 공개 설계: EtherNet/IP 태그 데이터 링크를 하는 변수를 설계합니다.	'6-3-4 변수의 속성(6-30페이지)' '6-3-5 데이터 형식(6-31페이지)' '6-3-6 파생 데이터 형식(6-41페이지)'

6-2 POU(프로그램 구성 단위)

NJ/NX 시리즈 CPU 유니트에서 동작하는 사용자 프로그램은 모두 POU(프로그램 구성 단위) 형식으로 작성하고, 이를 조합해서 사용합니다.
여기에서는 POU의 구성, 사양 등에 대해 자세하게 설명합니다.
Sysmac Studio에서 POU를 작성하는 방법에 대해서는 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.

6-2-1 POU의 개념

POU(Program Organization Unit)란 IEC 61131-3에서의 사용자 프로그램의 실행 모델 단위입니다. 사용자 프로그램을 구성하는 기본적인 요소로서, 로컬 변수 테이블과 알고리즘(일련의 코드/로직)을 기술합니다.
복수의 POU를 조합하여 사용자 프로그램 전체를 구성합니다.

POU의 구성 요소에는 다음 3종류가 있습니다.

- 프로그램
  메인 루틴에 상당하는 알고리즘을 기재하는 주요 요소입니다.
  알고리즘에 모든 명령, 펑션, 펑션 블록을 기술할 수 있습니다.
- 펑션 블록(이후 'FB'로 생략)
  입력 값이 동일해도 다른 값을 출력할 수 있는 요소입니다. 프로그램 또는 다른 FB에서 호출되어 기동합니다.
- 펑션(이후 'FUN'으로 생략)
  입력 값이 동일하면 항상 균일한 값을 출력할 수 있는 요소입니다. 프로그램, FB 또는 FUN에서 호출되어 기동합니다.

POU에서는 위의 3가지 요소를 조합해서 구성합니다. 또한, POU는 여러 개 작성할 수 있습니다.
작성한 사용자 프로그램은 태스크에 할당해서 기동시킵니다.

6-2-2 POU의 3가지 타입의 개요

프로그램의 개념

- **기동 방법/실행 조건**
  - 태스크가 기동하면 그 태스크에 할당된 프로그램을 기동합니다.
  - 상시 실행합니다.

- **기술**
  - POU에는 반드시 하나 이상의 프로그램을 기술해야 합니다. 1태스크당 프로그램 수는 최대 128개입니다.

펑션 블록(FB)의 개념

- **기동 방법/실행 조건**
  - 프로그램, FB에서의 호출에 의해 기동합니다.
  - 상시 실행합니다.
  - 조건이 성립했을 때만 실행시키고자 하는 경우에는 실행 조건을 의미하는 입력 변수를 정의하십시오.

- **기술**
  - 알고리즘에 모든 명령, 사용자 정의 FUN 및 사용자 정의 FB를 기술할 수 있습니다.
  - 내부 변수의 값을 유지할 수 있습니다. 따라서 타이머 및 카운터와 같이 상태를 가질 수 있습니다.
  - 사용자 정의의 FB와 시스템 정의의 FB가 있습니다.
    특히 사용자 정의의 FB는 '사용자 정의 FB'라고 합니다. 시스템 정의의 FB는 'FB 형식 명령'이라고 합니다.

FB의 자세한 내용은 📖'6-2-5 펑션 블록(FB)의 세부 내용(6-9페이지)'을 참조하십시오.

펑션(FUN)의 개념

- **기동 방법/실행 조건**
  - 프로그램, FB 또는 다른 FUN에서의 호출에 의해 기동합니다.
  - 실행 조건은 입력 변수(EN)로 지정합니다. EN이 TRUE인 경우에 1번 실행합니다.

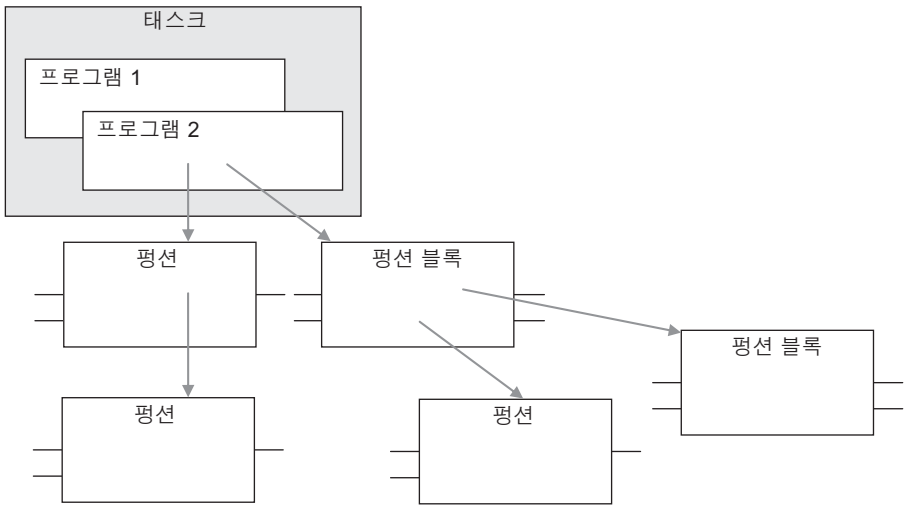
- **기술**
  - 알고리즘에 FB 형식 명령 및 사용자 정의 FB를 기술할 수 없습니다.
  - 내부 변수의 값을 유지할 수 없습니다. 따라서 입력값이 동일한 경우 출력값은 항상 일정합니다.
  - 사용자 정의의 FB와 시스템 정의의 FB가 있습니다.
    특히 사용자 정의의 FUN은 '사용자 정의 FUN'이라고 합니다. 시스템 정의의 FUN은 'FUN 형식 명령'이라고 합니다.

FUN의 자세한 내용은 📖'6-2-6 펑션(FUN)의 세부 내용(6-18페이지)'을 참조하십시오.

6-2-3 프로그램, 펑션 블록, 펑션의 차이점

항목 \ POU 종류		프로그램	펑션 블록(FB)	펑션(FUN)
기동 방법		할당된 태스크의 기동에 의해 기동	프로그램, 다른 FB에서의 호출	프로그램, FB, FUN에서의 호출
알고리즘 내	모든 명령	가능	가능	불가능
	사용자 정의 FUN	가능	가능	가능
	사용자 정의 FB	가능	가능	불가능
실행 조건		상시 실행	상시 실행 실행 조건은 입력 변수에 의해 지정	EN 입력으로 지정

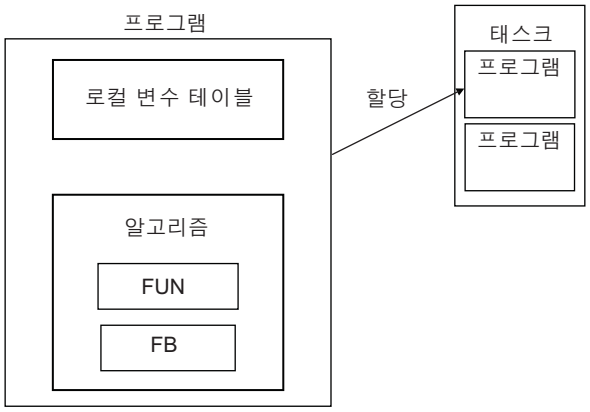
프로그램, FB, FUN의 계층 관계는 다음 그림과 같습니다.



6-2-4 프로그램의 세부 내용

프로그램의 구성

프로그램은 로컬 변수 테이블과 알고리즘 부분으로 구성됩니다.
알고리즘 부분 중에는 FUN이나 FB를 기술할 수 있습니다.



프로그램이 다른 POU에서 호출되지는 않습니다.

프로그램의 기동 조건

프로그램은 그것이 할당된 태스크가 기동하면 기동됩니다.

- **실행 순서**

  태스크에 할당된 프로그램은 태스크의 중에서의 실행 순서를 설정할 수 있습니다. Sysmac Studio의 [태스크 설정]–[프로그램 할당 설정]에서 지정합니다.

- **관련 시스템 정의 변수**

  프로그램은 각각 다음과 같은 로컬 변수의 시스템 정의 변수를 가지고 있습니다.

변수명	이름	기능	데이터 형식	R/W
P_First_Run Mode	운전 시작 시 1주기 ON 플래그	컨트롤러의 동작 모드를 프로그램 모드에서 운전 모드로 변경했을 때 해당 프로그램이 실행 중인 경우에 1태스크 주기만 TRUE가 됩니다. 해당 프로그램이 실행 중이 아니면 FALSE로 유지됩니다. CPU 유니트가 운전을 시작했을 때 초기 처리를 수행하는 경우 등에 사용합니다.	BOOL	R
P_First_Run*1	프로그램 기동 시 1주기 ON 플래그	해당 프로그램이 실행을 시작했을 때 1태스크 주기만 TRUE가 됩니다. *2 프로그램이 실행을 시작할 때마다 초기 처리를 수행하는 경우 등에 사용합니다.	BOOL	R
P_PRGER	명령 에러 발생 플래그	해당 프로그램 또는 해당 프로그램에서 호출한 FB나 FUN에서 명령 에러가 발생했을 때 계속 TRUE가 되는 플래그입니다. 한번 TRUE가 되면 사용자 프로그램에 의해 FALSE가 될 때까지 TRUE로 유지됩니다.	BOOL	RW
P_CY	캐리 플래그	일부 명령으로 갱신하는 플래그입니다.	BOOL	R

*1. CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.
*2. 프로그램의 실행/정지 명령에는 PrgStart 명령/PrgStop 명령을 사용합니다. PrgStart 명령은 P_First_Run을 TRUE로 하지 않고 프로그램을 실행하는 설정이 가능합니다.

6-2-5 펑션 블록(FB)의 세부 내용

펑션 블록(FB) 작성 순서

FB는 사전에 작성한 'FB정의'와 실제로 프로그램 상에서 사용하기 위한 '인스턴스'로 이루어집니다.
FB는 다음과 같은 순서로 작성합니다.

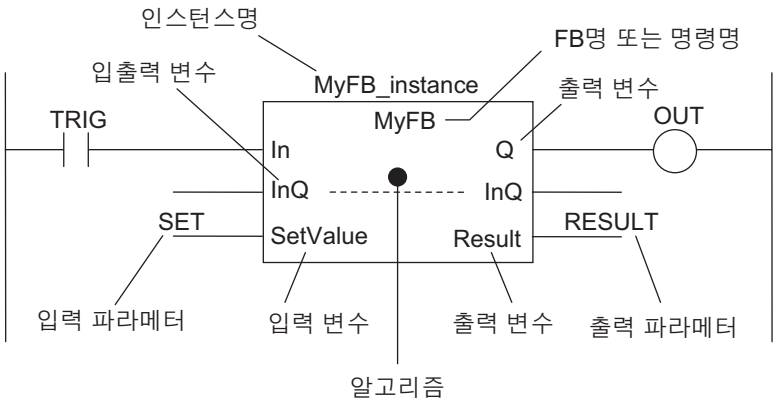
1 **FB 정의 작성**
알고리즘을 작성합니다.

2 **FB 정의를 인스턴스로 프로그램에 배치**
FB 정의를 프로그램 및 다른 FB에서 호출하여 실제로 배치합니다.
FB 정의 하나를 여러 프로그램 및 다른 FB에서 호출하여 배치할 수 있습니다.
FB 정의는 프로그램 및 다른 FB에 배치한 시점에 인스턴스로 개별적으로 취급되며, 프로그램 및 다른 FB 상에서 실제로 동작할 수 있게 됩니다.

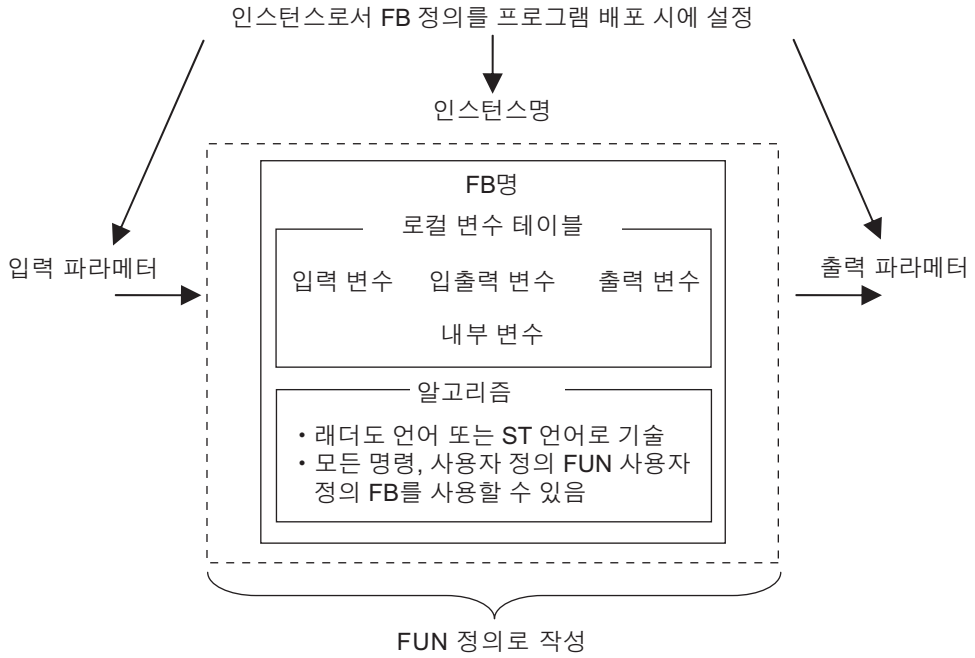
펑션 블록(FB)의 구성

래더도 언어에서는 FB는 다음과 같이 직사각형으로 표현됩니다. ST 언어에서의 표현은 뒤에 설명하는 'ST 언어에서의 펑션 블록(FB) 호출의 표현(6-11페이지)'을 참조하십시오.
FB는 다음 내용으로 구성됩니다.

- 래더도 언어로 표현한 FB:



- FB가 설정하는 항목
  FB 정의와 인스턴스로 배치 시 각각 다음과 같이 설정합니다.



- **FB명 또는 명령명**
  작성 시 붙인 FB 정의의 이름 또는 명령의 이름입니다.

- **인스턴스명**
  FB를 관리하기 위해 프로그램별로 배치한 FB에 대해 개별적으로 붙인 이름입니다.
  프로그램 및 다른 FB에서 FB를 호출할 때 인스턴스명을 설정합니다.

- **알고리즘**
  래더도 언어 또는 ST 언어로 기술할 수 있습니다.
  알고리즘 부분 중에 모든 명령, 사용자 정의FUN 및 사용자 정의 FB를 기술할 수 있습니다.

- **로컬 변수 테이블**
  입력 변수, 출력 변수, 입출력 변수, 내부 변수, 외부 변수가 정의되어 있습니다.
  '펑션 블록(FB) 내의 변수 사양(6-12페이지)'을 참조하십시오.

- **파라메터**

  입력 변수로의 입력 파라메터
  FB 실행 시작 시 FB 내의 입력 변수에 값을 전달합니다. 변수 또는 상수를 기술합니다.

  출력 변수에서의 출력 파라메터
  FB 실행 종료 시 FB 내의 출력 변수 값을 수신합니다. 변수를 기술합니다.

  입출력 변수—입출력 변수 사이에서 공유하는 입출력 파라메터
  FB 내에서 값을 변경합니다. 입력과 출력에 동일한 변수를 기술합니다.

참고

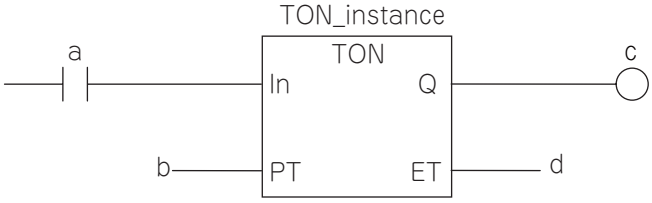
입력 파라메터, 출력 파라메터는 생략할 수도 있습니다. 동작의 자세한 내용은 '파라메터 생략 시의 동작(6-24페이지)'을 참조하십시오.

ST 언어에서의 펑션 블록(FB) 호출의 표현

ST 언어에서는 FB의 호출은 다음과 같이 표현합니다.

인스턴스명(입력 변수 1:= 입력 파라메터 1,· · · 입력 변수 N:= 입력 파라메터 N, 입출력 변수 1:= 입출력 파라메터 1,· · · 입출력 변수 N:= 입출력 파라메터 N, 출력 변수 1=> 출력 파라메터 1,· · · 출력 변수 N=> 출력 파라메터 N);

기술하는 파라메터의 순서를 FB 정의에 쓰인대로 하여 입력 변수명 등을 생략하고, 파라메터만으로 기술할 수도 있습니다.
이 경우에 파라메터의 수는 FB 정의의 입력 변수 등의 수와 일치해야 합니다.

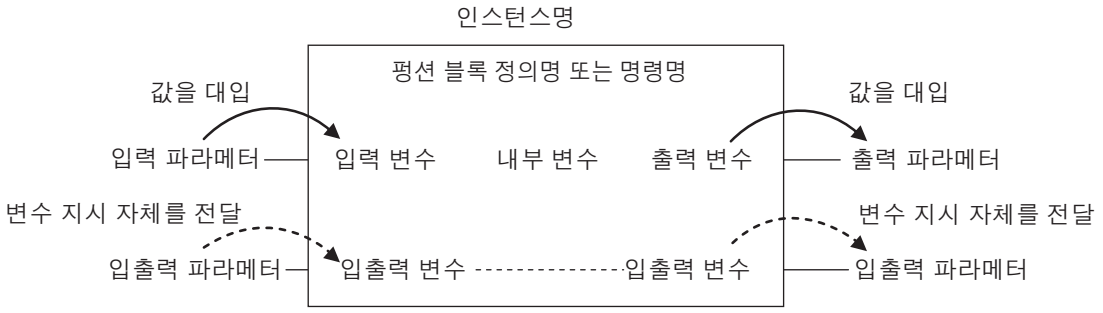


ST 언어로 표현한 FB:

```
인스턴스명
 /
TON_instance(In:=a, PT:=b, Q=>c, ET=>d);
TON_instance(In:=a, PT:=b, Q=>c); (*출력 ET를 생략한 경우*)
TON_instance (a, b, c, d); (*입력 변수, 출력 변수를 생략한 경우*)
```

자세한 내용은 뒤에 설명하는 📖'FB 호출(6-121페이지)'을 참조하십시오.

펑션 블록(FB) 내의 변수 사양



FB의 변수 사양은 다음과 같습니다.

변수	개수	사양
입력 변수*1	1~64	FB의 입력 인수로 FB 내부에서 사용되는(값을 변경할 수 없는) 변수입니다.
		• FB 실행 시작 시 입력 파라메터의 값을 입력 변수에 설정합니다. • 입력 파라메터에는 상수 또는 변수를 기술할 수 있습니다. • 입력 파라메터 생략: 📖'파라메터 생략 시의 동작(6-24페이지)'을 참조하십시오. • 상승/하강 엣지를 지정할 수 있습니다. • FB 외부에서 값을 참조 및 변경할 수 있습니다. '인스턴스명.입력 변수명'으로 지정합니다.*2
출력 변수*3	1~64	FB에서의 출력 인수로 FB 내부에서 사용되는 변수입니다.
		• FB 실행 종료 시 출력 변수의 값을 출력 파라메터에 설정합니다. • 출력 파라메터에 상수 또는 상수 속성이 있는 변수는 지정할 수 없습니다. • 출력 파라메터 연결을 생략할 수 있습니다. 생략한 경우에 출력 변수의 값은 어느 파라메터에도 대입되지 않습니다. • FB 외부에서 값을 참조할 수 있습니다. '인스턴스명.출력 변수명'으로 읽을 수 있습니다. 단, 쓸 수는 없습니다.
입출력 변수	0~64	FB에 대한 입출력으로 FB 내부에서 사용되는(값을 변경할 수 있는) 변수입니다.
		• '입출력 파라메터'를 전달하고 받습니다. • 입출력 파라메터에 상수 또는 상수 속성이 있는 변수는 지정할 수 없습니다. • FB 내에서 값을 변경하면 그 시점에서 입출력 파라메터의 값이 바뀝니다. • 입출력 파라메터를 생략할 수 없습니다.
내부 변수	무제한	FB 내에서의 일시적인 저장용 변수입니다.
		• 실행하지 않을 때도 실행 후에도 값이 유지됩니다. • 유지 속성을 갖게 할 수 있습니다. • FB 외부에서 값을 참조할 수 없습니다.
외부 변수	무제한	글로벌 변수를 참조하는 변수입니다.
EN	0	FB에서는 사용할 수 없습니다(사용자 정의 FB, FB 형식 명령 모두).
ENO	0 또는 1	일반적으로 정상 종료 시 TRUE, 이상 종료 시 또는 기동하지 않았을 때 FALSE가 되는 BOOL 형식의 출력 변수로 사용합니다.
		• 사용자 정의 FB에서는 생략할 수 있습니다. • 그 밖의 자세한 내용은 뒤에 설명하는 📖'ENO(6-21페이지)'를 참조하십시오.

*1. Sysmac Studio Ver.1.01 이하에서는 래더도 언어로 FB를 사용하는 경우에 ST 언어로 FB를 사용하는 경우 모두 BOOL 형식의 입력 변수가 1개 이상 필요합니다.
*2. Sysmac Studio Ver.1.07 이하에서는 FB 외부에서 입력 변수의 값을 변경할 수 없습니다. 단, FB 외부에서 참조할 수 있습니다.
*3. 래더도 언어로 FB를 사용하는 경우에 ENO를 포함하여 BOOL 형식의 출력 변수가 1개 이상 필요합니다.
ST 언어로 FB를 사용하는 경우에 Sysmac Studio Ver.1.01 이하에서는 ENO를 포함하여 BOOL 형식의 출력 변수가 1개 이상 필요합니다.

각 변수의 변수 속성 설정 가능 여부는 📖'6-3-4 변수의 속성(6-30페이지)'을 참조하십시오.

참고

글로벌 변수의 외부 변수에 대한 정의는 글로벌 변수와 동명의 변수를 FB 내의 외부 변수에서 정의하면 자동으로 설정됩니다.

- **ENO**

  ENO가 FALSE인 경우에 그 밖의 출력 변의 값은 이전의 값을 유지합니다.

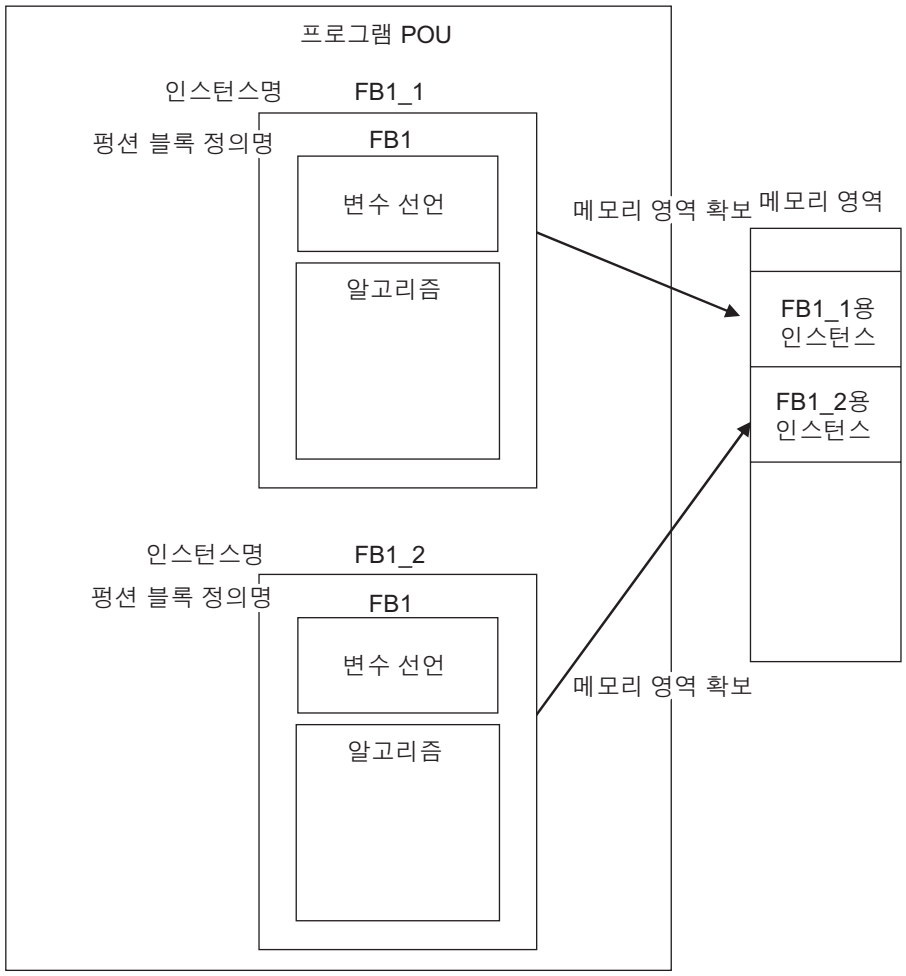
펑션 블록(FB)정의와 인스턴스

FB는 사전에 작성한 'FB정의'와 실제로 프로그램 상에서 사용하기 위한 '인스턴스'로 이루어집니다.
재이용의 기초가 되는 것을 'FB 정의'라고 합니다.
'FB 정의'에는 알고리즘과 로컬 변수 테이블을 기재합니다.

- **FB의 인스턴스**

  FB 정의를 프로그램 또는 다른 FB상에 배치하면 FB 정의는 실제 프로그램 및 FB의 일부로 취급됩니다.
  FB 정의를 프로그램 또는 다른 FB 상에 배치한 것은 '인스턴스'라고 합니다.
  각 인스턴스에는 인스턴스명이라는 식별자가 있으며, 각각 메모리 영역을 사용합니다.

  하나의 FB 정의를 바탕으로 복수의 '인스턴스'를 생성하면 동일 기능으로 다른 입출력 데이터 처리를 시킬 수 있습니다.

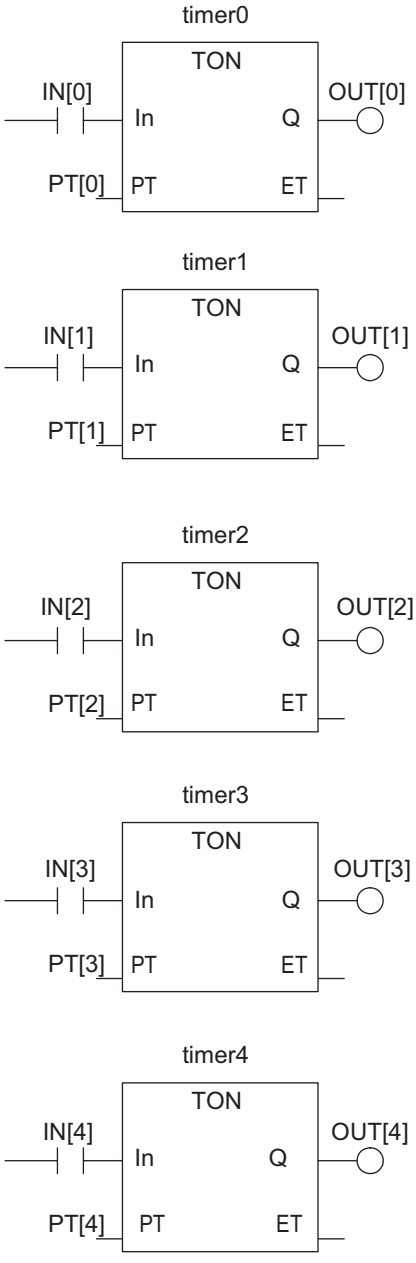


'인스턴스'는 다른 프로그램 및 다른 FB에서 읽을 수 없습니다. 인스턴스명이 동일해도 다른 프로그램 및 다른 FB에 배치된 경우에는 다른 인스턴스로 동작합니다.

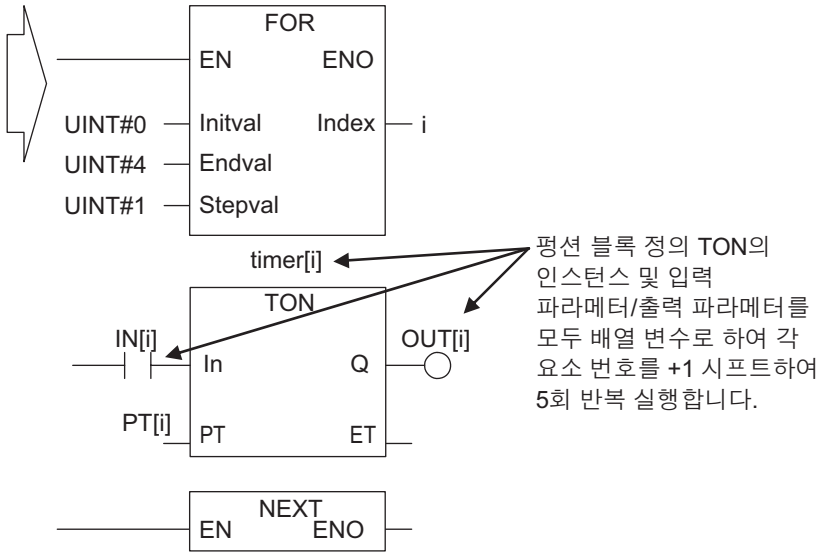
인스턴스의 배열 지정화

인스턴스에 대해 배열 지정할 수 있습니다.
요소 번호를 변수의 값에 의해 간접 지정하면 하나의 인스턴스명으로 복수의 인스턴스를 실행할 수 있습니다.
그리고 입력 파라메터 및 출력 파라메터도 배열 지정으로 하고, 공통의 변수로 요소 번호를 지정하면 하나의 인스턴스명으로 입력원, 출력 대상을 전환하여 복수의 인스턴스를 실행할 수 있습니다.
예)

인스턴스를 배열 지정하지 않을 때



인스턴스를 배열 지정했을 때



변수 테이블

변수명	데이터 형식
IN	ARRAY [0..4] OF BOOL
OUT	ARRAY [0..4] OF BOOL
PT	ARRAY [0..4] OF TIME
timer	ARRAY [0..4] OF TON
i	UINT

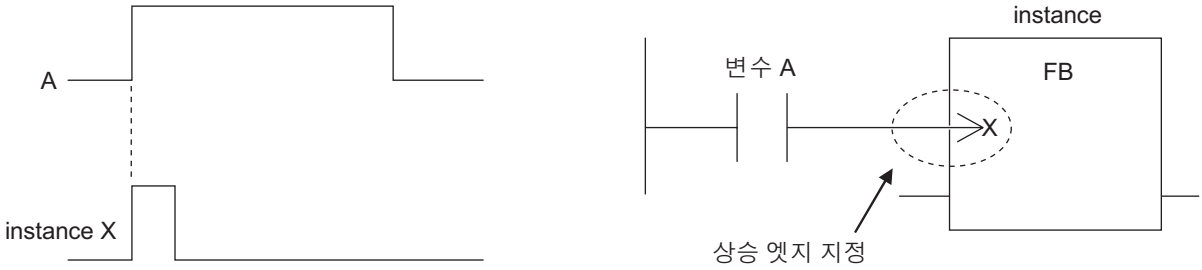
펑션 블록(FB)의 기동 조건

FB는 FUN과 달리 EN 입력이 없으며 상시(태스크 주기마다) 실행됩니다.

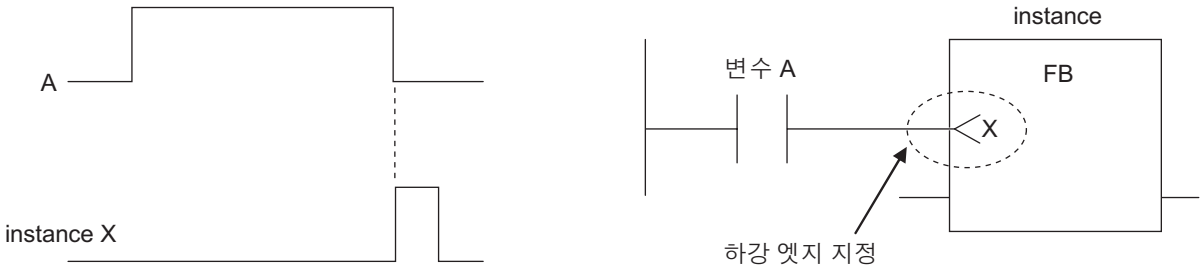
경우	FB 내 알고리즘		ENO	ENO 이외의 동작
통상 시	실행됩니다.	정상 종료 시	TRUE	출력 파라메터: 내부 알고리즘에 의해 값을 갱신 입출력 파라메터: 내부 알고리즘에 의해 값을 갱신
		이상 종료 시	FALSE	출력 파라메터: 유지 입출력 파라메터: 내부 알고리즘에 의해 값을 갱신
마스터 컨트롤 영역 내	전력 흐름 입력이 FALSE 상태에서 실행함		사용자 임의	ENO 값에 따라 상기 중 하나가 됨

전력 흐름 출력, 파라메터 출력에 대해서는 '6-5-2 래더도 언어(LD)(6-90페이지)'를 참조하십시오.

입력 변수에 엣지를 지정하면 해당 변수는 입력 파라메터 상승 시 TRUE가 됩니다.



하강 지정도 가능합니다.

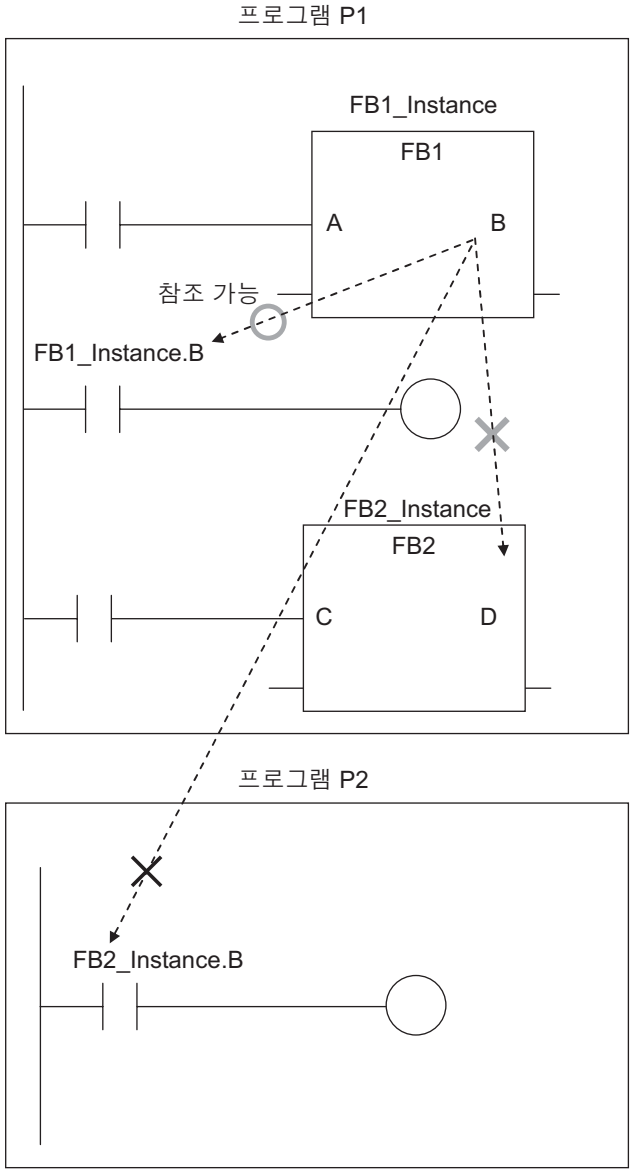


펑션 블록(FB)의 변수 외부에서의 참조

FB의 입력 변수 및 출력 변수는 FB 외부에서 참조할 수 있습니다. 변수 표기는 다음과 같습니다.
인스턴스명.변수명

(예) FB1의 인스턴스 'FB1_Instance' 내의 출력 변수 'B'인 경우
FB1_Instance.B

참조 가능한 범위는 해당 인스턴스가 있는 프로그램 내입니다.
그 프로그램 내의 다른 FB의 인스턴스에서는 참조할 수 없습니다. 또한, 다른 프로그램에서도 참조할 수 없습니다.



FB1의 입력 변수, 출력 변수는
・프로그램 P1에서 밖에 볼 수 없습니다.
・프로그램 P2에서 참조할 수 없습니다.
・FB2에서 참조할 수 없습니다.

다음 변수는 외부에서 참조할 수 없습니다. 참조한 경우에는 빌드 시 에러가 됩니다.

- FB의 입출력 변수
- 입력 파라메터가 생략된 경우에 초기값이 적용되지 않는 FB 형식 명령의 입력 변수

6-2-6 펑션(FUN)의 세부 내용

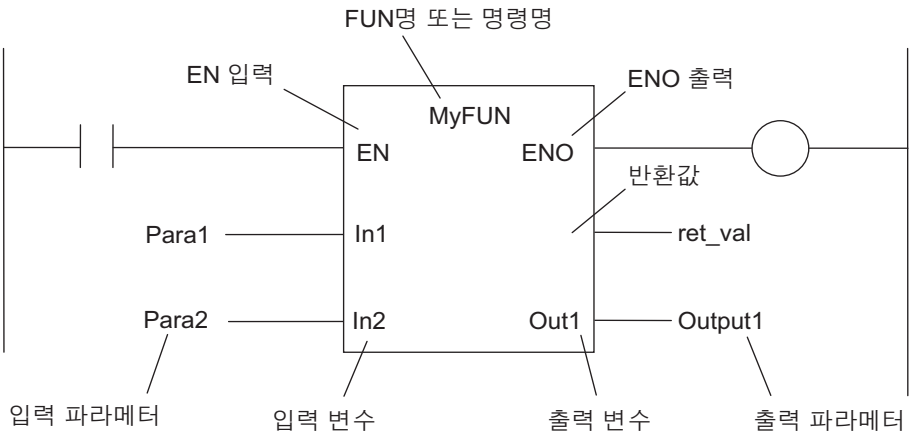
펑션(FUN)의 구성

래더도 언어에서는 FUN은 다음과 같이 직사각형으로 표현됩니다.

ST 언어에서의 표현은 뒤에 설명하는 'ST 언어에서의 펑션(FUN)의 표현(6-19페이지)'을 참조하십시오.

FUN은 다음 내용으로 구성됩니다.

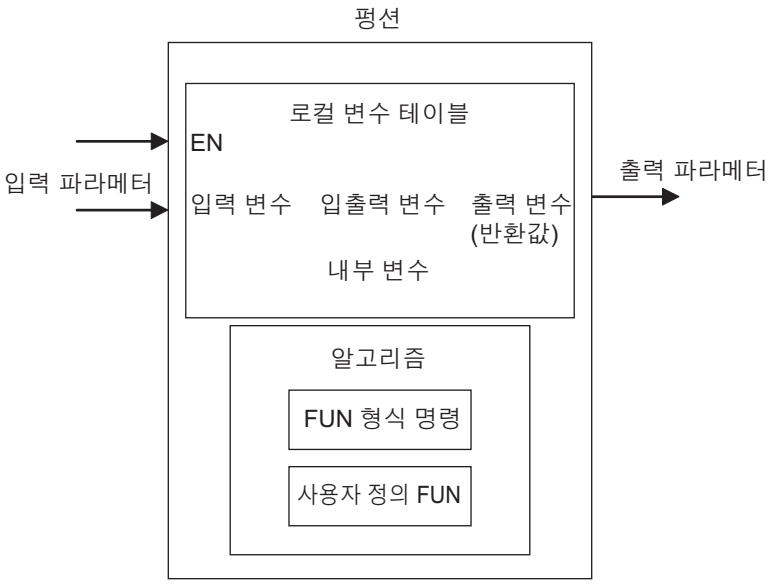
래더도 언어로 표현한 FUN:



- **FUN 정의명 또는 명령명**
  작성 시 붙인 FUN 정의의 이름 또는 명령의 이름입니다.

- **인스턴스명**
  없습니다.

- **알고리즘**
  래더도 언어 또는 ST 언어로 기술할 수 있습니다.
  알고리즘 부분 중에서 FUN 형식 명령, 사용자 정의 FUN을 기술할 수 있습니다.
  FB 형식 명령, 사용자 정의 FB를 기술할 수 없습니다.
  미분 명령(R_TRIG, UP)을 기술할 수 없습니다.
  시스템 정의 변수 P_First_RunMode, P_First_Run를 사용할 수 없습니다.



- **로컬 변수 테이블**
  입력 변수, 출력 변수, 입출력 변수, 내부 변수, 외부 변수를 등록합니다.
  '펑션(FUN) 내의 변수 사양(6-20페이지)'을 참조하십시오.

- **파라메터**

  입력 변수로의 입력 파라메터
  FUN 실행 시작 시 FUN 내의 입력 변수에 값을 전달합니다. 변수 또는 상수를 기술합니다.

  출력 변수에서의 출력 파라메터
  FUN 실행 종료 시 FUN 내의 출력 변수의 값을 수신합니다. 변수를 기술합니다.

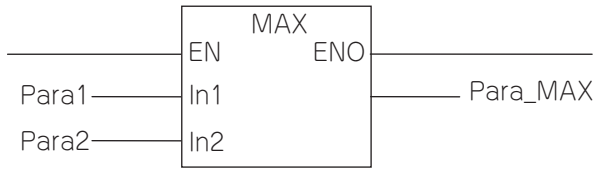
  입출력 변수—입출력 변수 사이에서 공용하는 입출력 파라메터
  FUN 내에서 값을 변경합니다. 입력과 출력에 동일한 변수를 기술합니다.

ST 언어에서의 펑션(FUN)의 표현

ST 언어에서는 FUN은 다음과 같이 표현합니다.

반환값:=FUN명(입력 변수 1:= 입력 파라메터 1, · · · 입력 변수 N:= 입력 파라메터 N, 입출력 변수 1:= 입출력 파라메터 1, · · · 입출력 변수 N:= 입출력 파라메터 N, 출력 변수 1=> 출력 파라메터 1, · · · 출력 변수 N=> 출력 파라메터 N);
단, 반환값을 생략할 수 있음

기술하는 파라메터 순서를 FUN 정의에 쓰인대로 하여 입력 변수명 등을 생략하고, 파라메터만으로 기술할 수도 있습니다. 이 경우에 파라메터의 수는 FUN 정의의 입력 변수 등의 수와 일치해야 합니다.
ST 언어로 표현한 FUN의 예:

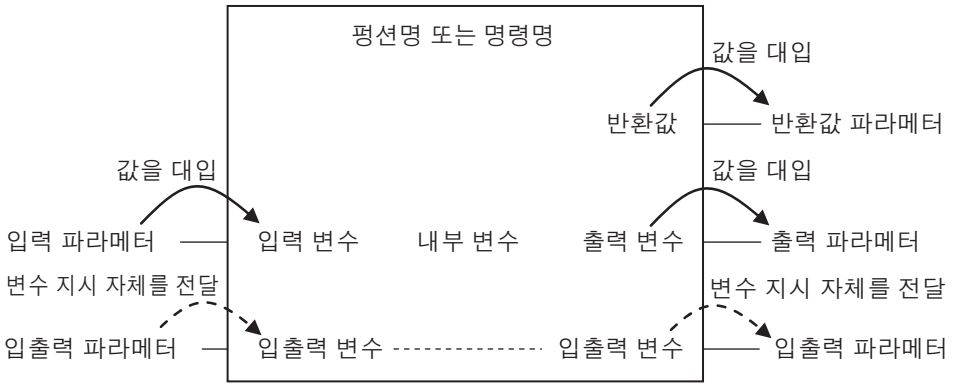


FUN명

```
Para_MAX := MAX(In1:=Para1, In2:=Para2);
Para_MAX := MAX(Para1, Para2);                (*입력 변수를 생략한 경우*)
```

자세한 내용은 뒤에 설명하는 'FUN 호출(6-124페이지)'을 참조하십시오.

펑션(FUN) 내의 변수 사양



FUN의 변수 사양은 다음과 같습니다.

변수	개수	사양
입력 변수	0~64	FUN의 입력 인수로 FUN 내부에서 사용되는(값을 변경할 수 없는) 변수입니다.
		• FUN 실행 시작 시 입력 파라메터의 값을 입력 변수에 설정합니다. • 입력 파라메터에는 상수 또는 변수를 기술할 수 있습니다. • 입력 파라메터 생략: 📖'파라메터 생략 시의 동작(6-24페이지)'을 참조하십시오. • FB와 달리 상승/하강 엣지를 지정할 수 없습니다. • FUN 외부에서 값을 참조할 수 없습니다. • OMRON이 제공하는 명령에는 입력 변수의 수를 변경할 수 있는 것이 있지만 사용자가 작성하는 FUN은 입력 변수의 수를 변경 가능하게 할 수 없습니다.
출력 변수	0~64	FUN에서의 출력 인수로 FUN 내부에서 사용되는 변수입니다.
		• FUN 실행 종료 시 출력 변수의 값을 출력 파라메터에 설정합니다. • 출력 파라메터에 상수 또는 상수 속성이 있는 변수는 지정할 수 없습니다. • ENO, 반환값을 포함해서 최소 1개는 BOOL 형식 출력 변수가 필요합니다. • 출력 파라메터 연결을 생략할 수 있습니다. 생략한 경우에 출력 변수의 값은 어느 파라메터에도 대입되지 않습니다. • FUN 외부에서 값을 참조할 수 없습니다. • 사용자 정의의 FUN의 경우에 FUN 내의 알고리즘으로 반드시 출력 변수의 값을 설정하십시오. FUN 내의 알고리즘으로 출력 변수를 설정하지 않으면 출력 변수의 값이 잘못됩니다.
입출력 변수	0~64	FUN에 대한 입출력으로 FUN 내부에서 사용되는(값을 변경할 수 있는) 변수입니다.
		• '입출력 파라메터' 자체(변수 지시 자체)를 주고 받습니다. • 입출력 파라메터에 상수 또는 상수 속성이 있는 변수는 지정할 수 없습니다. • FUN 내에서 값을 변경하면 그 시점에서 입출력 파라메터의 값이 바뀝니다. • 입출력 파라메터를 생략할 수 없습니다. • FUN 외부에서 값을 참조할 수 없습니다.
내부 변수	무제한	FUN 내에서의 일시적인 저장용 변수입니다.
		• 실행 종료 후에는 값이 유지되지 않습니다. • FUN 외부에서 값을 참조할 수 없습니다.
외부 변수	무제한	글로벌 변수를 참조하는 변수입니다.
EN	1	FUN을 기동하기 위한 BOOL 형식의 입력 변수입니다.
		• EN의 값이 TRUE일 때, FUN이 실행됩니다. FUN 내의 EN 값은 항상 TRUE입니다. • 반드시 1개 존재합니다(사용자 정의 FUN, FUN 형식 명령 모두).

변수	개수	사양
ENO	0 또는 1	일반적으로 정상 종료 시 TRUE, 이상 종료 시 또는 기동하지 않았을 때 FALSE가 되는 BOOL 형식의 출력 변수로 사용합니다.
		• 사용자 정의 FUN에서는 생략할 수 있습니다. • 그 밖의 자세한 내용은 뒤에 설명하는 'ENO(6-21페이지)'를 참조하십시오.
반환값	1	FUN이 알고리즘을 종료하고, 호출원에게 처리 결과로 반환하는 값입니다.
		• 반드시 1개 존재합니다. • 기본 데이터 형식 전체, 열거 형식을 지정할 수 있습니다. 배열 지정, 구조체 전체, 공용체 전체를 지정할 수 없습니다. • 사용자 정의의 FUN인 경우에 FUN 내의 알고리즘으로 반드시 반환값을 설정하십시오. FUN 내의 알고리즘으로 반환값을 설정하지 않으면 반환값이 잘못됩니다. • 그 밖의 자세한 내용은 뒤에 설명하는 '반환값(6-22페이지)'을 참조하십시오.

각 변수의 변수 속성 설정 가능 여부는 '6-3-4 변수의 속성(6-30페이지)'을 참조하십시오.

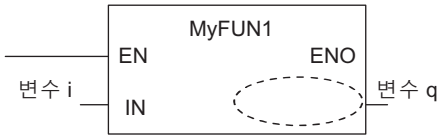
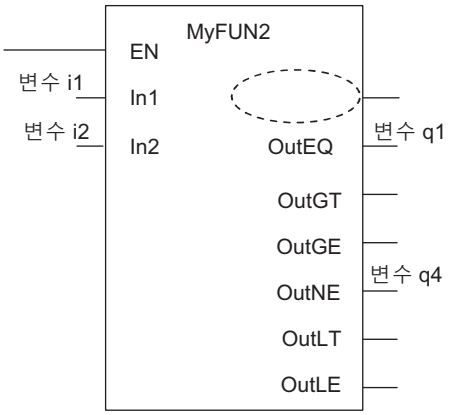
참고

FUN 내 변수 테이블에서 글로벌 변수를 외부 변수로 등록하면 글로벌 변수를 참조할 수 있습니다. 일반적으로 펑션은 입력 파라메터의 값에서 출력값이 일정해지도록 작성하는 것이 권장되고 있습니다. 펑션 내에서 글로벌 변수를 참조하고, 그에 따라 출력값이 영향을 받는 알고리즘은 바람직하지 않습니다. 따라서 Sysmac Studio에서 프로그램 확인 시 '펑션 내에서의 글로벌 변수 사용은 권장하지 않습니다. '라는 경고가 표시됩니다. 확인한 후 필요한 조치를 하십시오.

- **ENO**
  - ENO가 FALSE인 경우에 그 밖의 출력 변의 값은 이전의 값을 유지합니다.

● **반환값**

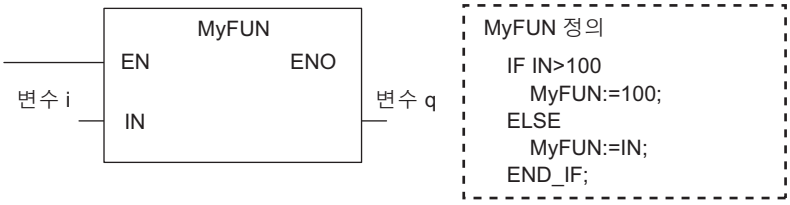
• 반환값은 래더도 언어상에서는 공백으로 표시됩니다.

경우	래더도 언어 표기	ST 언어 표기
반환값을 사용하는 경우		변수 q := MyFUN1(변수 i);
반환값을 사용하지 않는 경우		MyFUN2(In1:= 변수 i1, In2:= 변수 i2, OutEQ=>변수 q1, OutNE=>변수 q4);

• 래더도 언어, ST 언어 모두 호출 측에서 반환값을 사용하지 않을 수도 있습니다.

• 펑션 내의 알고리즘으로 반환값에 값을 설정하는 경우에 펑션명과 동명의 변수에 값을 설정하십시오.

예를 들어 'MyFUN'이라는 펑션인 경우에 반환값은 'MyFUN'이 됩니다.

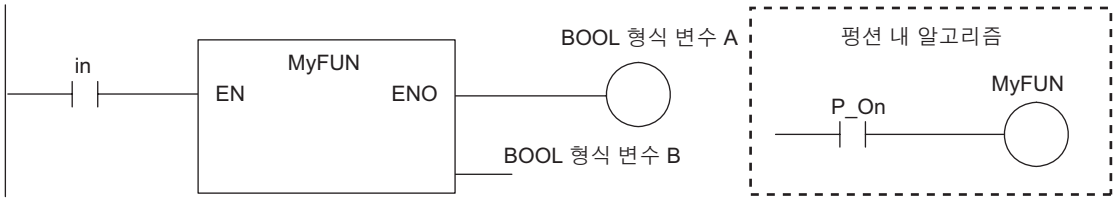


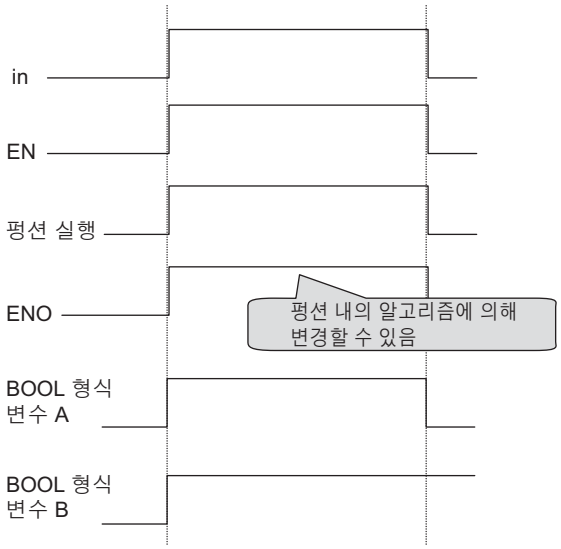
펑션(FUN)의 기동 조건

EN이 TRUE가 되면 FUN이 기동합니다. EN이 FALSE가 되면 FUN은 처리를 정지합니다.

입력 변수	FUN 내 알고리즘		ENO	ENO 이외의 동작
EN=TRUE 시	실행됩니다.	정상 종료 시	TRUE	출력 파라메터: 내부 알고리즘에 의해 값을 갱신 입출력 파라메터: 내부 알고리즘에 의해 값을 갱신
		이상 종료 시	FALSE	출력 파라메터: 유지 입출력 파라메터: 내부 알고리즘에 의해 값을 갱신
EN=FALSE 시	실행되지 않습니다		FALSE	출력 파라메터, 입출력 파라메터: 값을 유지
마스터 컨트롤 영역 내	실행되지 않습니다		FALSE	출력 파라메터, 입출력 파라메터: 값을 유지

예)





6-2-7 펑션 블록(FB), 펑션(FUN) 공통 동작

EN, ENO 사용/생략 가능 여부

FB /FUN에서의 EN, ENO 사용/생략 가능 여부는 다음과 같습니다.

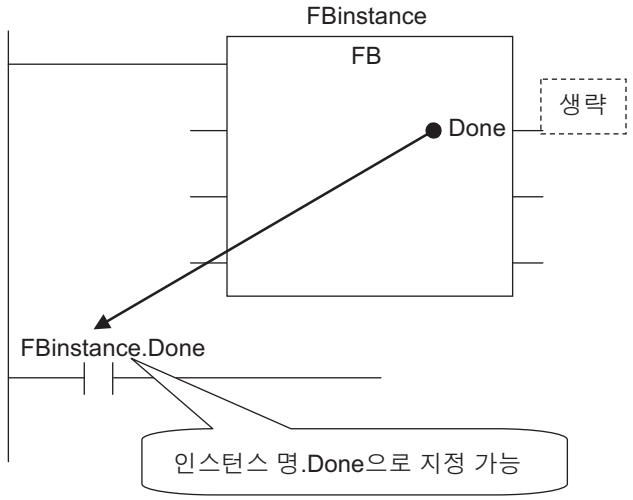
POU		EN	ENO
FB	사용자 정의	사용 불가 Sysmac Studio에서 변수 테이블에 정의하려고 하면 빌드 시 에러가 됩니다.	사용 가능(생략 가능) Sysmac Studio에서 사용자가 출력 변수로 정의합니다.
	명령	모든 FB 형식 명령은 EN을 사용하지 않습니다.	ENO를 사용하는 명령과 사용하지 않는 명령이 있습니다.
FUN	사용자 정의	사용 필수 FUN 작성 시에 Sysmac Studio가 기본값으로 변수 테이블에 자동으로 정의합니다.	사용 가능(생략 가능) Sysmac Studio에서 사용자가 출력 변수로 정의합니다.
	명령	모든 FUN 형식 명령은 EN을 사용합니다.	ENO를 사용하는 명령과 사용하지 않는 명령이 있습니다.

파라메터 생략 시의 동작

입력 파라메터와 출력 파라메터는 모두 생략할 수 있습니다.

	생략한 경우의 동작	
생략 대상 파라메터	**FB**	**FUN**
입력 변수로의 입력 파라메터	• 인스턴스의 최초 기동 시에는 초기값으로 동작합니다. • 이후에는 이전 값으로 동작합니다(항상 입력 변수가 생략되는 경우에는 계속 초기값으로 동작).	• EN은 TRUE로 동작합니다. • 그 밖의 입력 파라메터는 초기값으로 동작합니다.
출력 변수에서의 출력 파라메터	생략할 수 있습니다. 그 명령 외부에서 '인스턴스명.출력 변수명'으로 액세스할 수 있습니다(*)	생략할 수 있지만 생략 시에는 동작 결과를 얻을 수단이 없습니다.
입출력 변수로의 입출력 변수에서의 입출력 파라메터	생략할 수 없습니다.	생략할 수 없습니다.

* FB의 입력 변수/출력 변수를 '인스턴스명.변수명'으로 FB 외부(단, 해당 프로그램만 가능)에서 직접 참조할 수 있습니다. FUN에서는 참조할 수 없습니다.



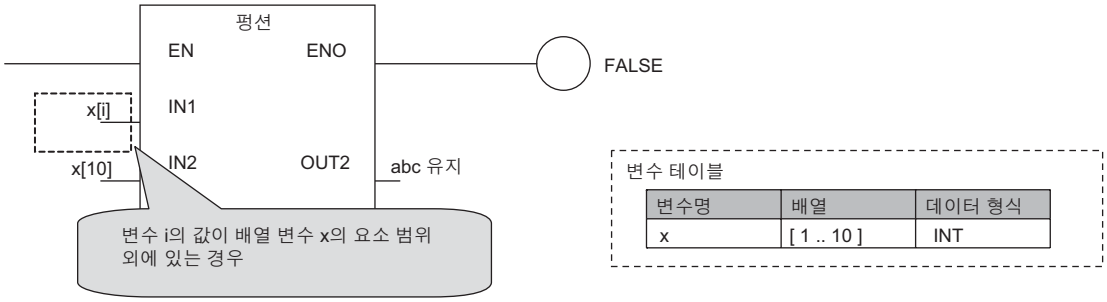
파라메터 이상 시의 동작

입력 파라메터, 출력 파라메터 또는 입출력 파라메터에 이상이 발생한 경우에 다음과 같은 동작이 됩니다.

- **입력에 이상 발생 시**

  입력 파라메터에 이상을 감지한 경우에 FB/FUN은 실행하지 않고 ENO를 FALSE로 합니다. 동시에 전력 흐름 출력은 FALSE로 하고, 그 이외는 값을 유지합니다.

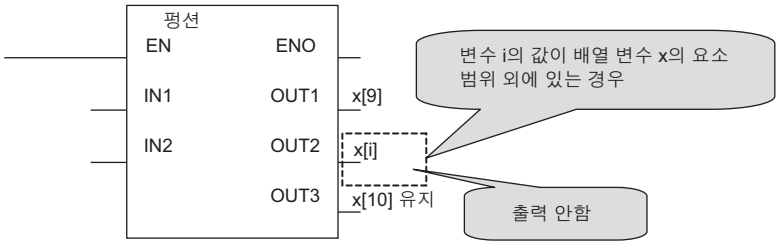
  예)



- **출력 파라메터에 이상 발생 시**

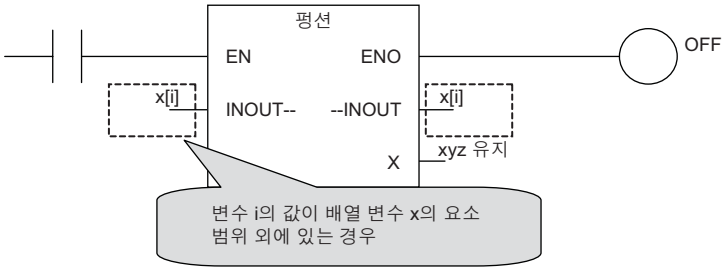
  출력 파라메터에 이상을 감지한 경우에 해당 파라메터 이상에 값을 출력하지 않고 값을 유지합니다.

  예)



- **입출력에 이상 발생 시**

  입출력 파라메터에 이상을 감지한 경우에 FB/FUN은 실행하지 않고 ENO를 FALSE로 합니다. 동시에 전력 흐름 출력은 FALSE로 하고, 그 이외는 값을 유지합니다.



재귀 호출

FB 및 FUN 모두 다음과 같은 재귀 호출을 할 수 없습니다. 그 경우에 Sysmac Studio에서 빌드 시 에러가 됩니다.

- 자신을 호출
- 호출된 FB 또는 FUN 측에서 원래의 부모를 호출

6-2-8 POU의 제한 사항

POU를 작성한 후의 각종 제한 사항을 설명합니다.

이름

POU명, FB의 인스턴스명의 제약은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.

복수의 인수 전달

복수의 인수를 FB 및 FUN에 전달하는 경우에 배열 지정 및 구조체 형식으로 데이터를 전달하는 형식으로 하면 프로그램이 간결해집니다.

이 경우에 입력 변수로 전달하기 보다 입출력 변수로 전달하는 쪽이 처리 시간이 짧아집니다. 단, 입출력 변수를 사용하면 FB 및 FUN에 인수로 전달한 데이터의 내용이 재기입된 경우에는 원래 값은 남지 않으므로 주의하십시오.

참고

배열 지정 변수와 구조체 형식 변수의 멤버 전체를 파라메터로 할 때

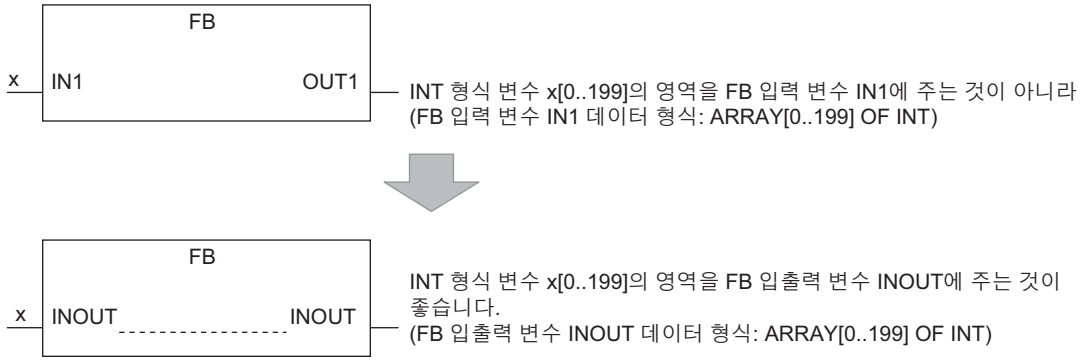
입력 파라메터, 출력 파라메터에 배열 지정 변수 또는 구조체 형식 변수의 멤버 전체를 지정할 수도 있습니다. 단, 기본 데이터 형식의 변수에 비해 데이터의 전달에 시간이 필요합니다(크기에 따라 다름).

따라서 FB인 경우에 배열 지정 변수와 구조체 형식 변수의 멤버 전체를 전달할 때는 입출력 변수에 전달하도록 설계하는 것을 권장합니다.

예 1) 배열 지정의 전체 요소를 지정할 때

프로그램 변수 테이블

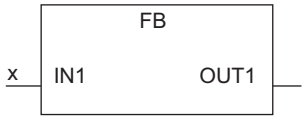
변수명	데이터 형식
x	ARRAY[0..199] OF INT



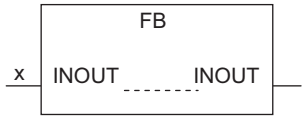
예 2) 구조체 형식 변수의 멤버 전체를 지정할 때

프로그램 변수 테이블

변수명	데이터 형식
x	MyStructure



I MyStructure 형식 변수 x의 멤버 전체 영역을 FB 입력 변수 IN1에주는 것이 아니라
(FB 입력 변수 IN1 데이터 형식: MyStructure)



MyStructure 형식 변수 x의 멤버 전체 영역을 FB 입출력 변수 INOUT에 주는 것이 좋습니다.
(FB 입출력 변수 INOUT 데이터 형식: MyStructure)

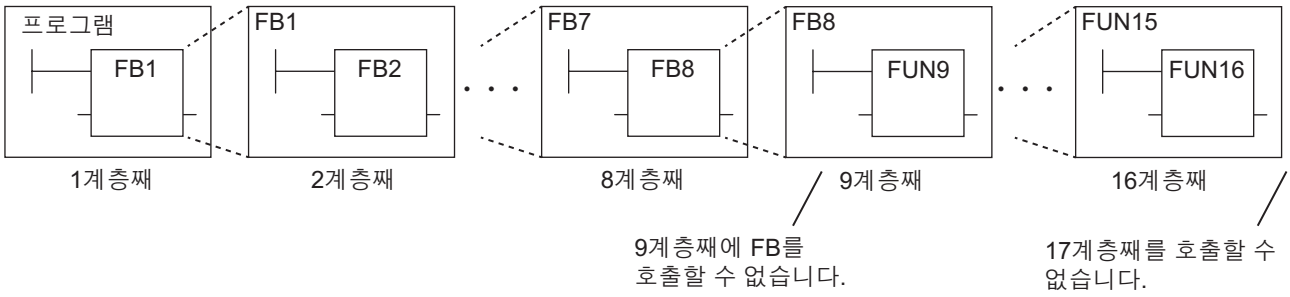
중첩 계층

프로그램에서 호출된 FB 및 FUN에서 다시 다른 FB 및 FUN을 호출하는 것을 중첩이라고 합니다. 사용자 정의의 FB 및 FUN에서 호출할 수 있는 POU와 중첩의 계층의 깊이에는 다음과 같은 제한이 있습니다. 제한을 초과하면 빌드 시 에러가 됩니다.

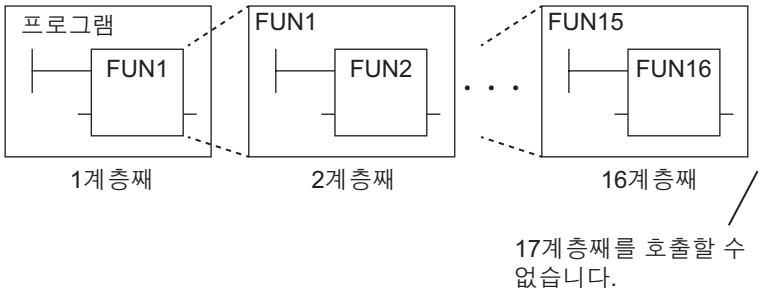
POU	호출할 수 있는 POU	계층의 깊이
FB	FB, FUN	최대 8계층
FUN	FUN	최대 16계층*1

*1. CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 할 수 있습니다. 그 이외는 최대 8계층입니다.

예 1) 프로그램에서 8번째 계층까지 FB를 호출할 수 있습니다. 다시 거기에서 16번째 계층까지 FUN을 호출할 수 있습니다.



예 2) 프로그램에서 16번째 계층까지 FUN을 호출할 수 있습니다.



6-3 변수

NJ/NX 시리즈 시스템에서는 외부와의 입출력 정보 교환 및 데이터 연산 등의 처리를 모두 변수를 통해 실시합니다.
여기에서는 변수의 사양에 대해 자세하게 설명합니다.
Sysmac Studio에서의 변수 설정 방법에 대해서는 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

6-3-1 변수의 개념

변수란 외부와 교환한 입출력 데이터 및 POU 내부 처리 시의 일시 데이터를 저장하는 용기입니다. 다시 말하면 이름 및 데이터 형식 등의 속성을 가진 데이터를 담는 용기입니다.

변수와 메모리 주소의 할당을 지정할 필요는 없습니다. 또한, 변수를 특정 메모리 주소에 할당할 수도 있습니다(*). NJ/NX 시리즈 CPU 유니트가 변수를 위한 메모리 영역을 자동으로 할당합니다.

* 일부 CJ 시리즈 고기능 유니트에서 특정 기능을 사용하는 경우 및 상위 링크(FINS)에서 시리얼 통신하는 경우에 사용하십시오. 변수 속성의 AT(할당대상) 지정에서 CJ 유니트용 메모리의 주소를 지정해야 합니다. 자세한 내용은 'AT(할당대상) 지정(6-59페이지)'을 참조하십시오.
* CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

6-3-2 변수의 종류

변수에는 크게 나눠서 다음 3종류가 있습니다.

- **사용자 정의 변수**
  모든 속성이 사용자에 의해 정의되는 변수입니다.
  이 장의 다음 설명에서는 이 '사용자 정의 변수'에 대해 설명합니다.

- **준 사용자 정의 변수**
  특정 장치/데이터에 액세스하기 위한 변수입니다.
  디바이스 변수와 캠 데이터 변수가 있습니다.
  디바이스 변수에 대해서는 '2-3-1 변수의 종류(2-12페이지)' 및 '3-3-1 I/O 포트(3-8페이지)'를 참조하십시오.

- **시스템 정의 변수**
  사전에 NJ/NX 시스템으로 이름 등의 모든 속성이 정의되어 있으며, 특정 기능이 할당된 변수입니다. 기능 모듈별로 준비되어 있습니다. 자세한 내용은 'A-6 시스템 정의 변수 목록(A-64페이지)'을 참조하십시오.

  변수 분류의 자세한 내용은 '2-3-1 변수의 종류(2-12페이지)'를 참조하십시오.

6-3-3 POU에서 본 사용자 정의 변수의 종류

사용자 정의 변수에는 POU에서의 기능에 따라 다음 6종류가 있습니다.

○: 정의 가능

사용자 정의 변수의 종류		POU 타입		
		프로그램	FB	FUN
로컬 변수	내부 변수	○	○	○
	입력 변수	없음	○	○
	출력 변수	없음	○	○
	입출력 변수	없음	○	○
글로벌 변수		○(*)	○(*)	○(*)
외부 변수		○	○	○

* 글로벌 변수를 외부 변수로 정의하면 외부 변수를 통해 글로벌 변수를 참조할 수 있습니다.

로컬 변수

POU(프로그램, FB, FUN)에서만 읽기/쓰기 가능한 변수입니다.
로컬 변수는 POU가 프로그램인 경우에 내부 변수와 같습니다. POU가 FB 또는 FUN인 경우에 내부 변수, 입력 변수, 출력 변수, 입출력 변수, 외부 변수의 총칭입니다.

- **내부 변수**
  POU의 내부에서만 사용하는 변수입니다.
  POU의 로컬 변수 테이블에서 정의(선언)합니다.
  POU 외부에서 내부 변수의 값을 참조할 수 없습니다.
  동일한 이름의 내부 변수를 다른 POU에서 선언해도 각각의 변수별로 다른 할당대상에 할당됩니다.

- **입력 변수**
  POU가 호출될 때 호출원에서 입력 파라메터의 값이 대입되는 변수입니다. POU의 로컬 변수 테이블에서 정의(선언)합니다.

- **출력 변수**
  POU의 처리가 종료되기 전에 출력 파라메터에 그 값이 대입되고, 호출처로 반환되는 변수입니다.
  POU의 로컬 변수 테이블에서 정의(선언)합니다.

- **입출력 변수**
  POU 호출 시 호출처에서 입출력 파라메터 자체(변수 지시 자체)가 대입되는 변수입니다. POU 내에서 재기입되면 그 시점에 입출력 파라메터의 값이 재기입됩니다.
  POU의 로컬 변수 테이블에서 정의(선언)합니다.

- **외부 변수**
  POU에서 그 외부를 참조하기 위한 변수입니다.
  등록된 글로벌 변수를 POU에서 참조합니다.

글로벌 변수

글로벌 변수 테이블에서 정의(선언)합니다.

슬레이브/유니트 구성에서 자동 생성한 디바이스 변수 및 전체 축 설정 테이블에서 자동 생성한 '축/축 그룹 변수'는 자동으로 글로벌 변수에 등록됩니다.

6-3-4 변수의 속성

변수에는 다음과 같은 속성을 설정 또는 부가할 수 있습니다.

변수의 속성 목록과 변수 종류별 속성 유무

- **변수의 속성 목록**

속성	설명	지정	초기 설정
변수명	변수를 식별하기 위한 이름입니다.	–	–
데이터 형식	변수에 저장하는 데이터를 취급하는 형식을 정의합니다.	–	BOOL
AT(할당대상) 지정	CJ 유니트의 특정 주소를 변수로 취급하는 경우에 해당 변수의 할당대상을 지정합니다.*1	• 지정 안 함 • 지정	지정 안 함
유지	다음 중 하나에 해당하는 경우 값을 유지할지 여부를 지정합니다. • 전원 OFF 후의 전원 투입 시 • 운전 모드로 전환 시 • 전체 정지 이상 레벨의 컨트롤러 이상 발생 시	• 유지: 왼쪽에 기재된 경우의 값을 유지(배터리가 있는 경우만) • 비유지: 초기값이 됨	비유지: 초기값이 됨
초기값	초기값 설정 '없음'과 '있음'을 선택할 수 있습니다. 초기값 설정 '있음'인 경우: 유지 지정 없이 다음 중 하나인 경우의 변수 값을 지정합니다. • 전원 투입 시 • 동작 모드 변경 시 • 전체 정지 이상 레벨의 컨트롤러 이상 발생 시 초기값 설정 '없음'인 경우: 값은 유지되지 않습니다.	초기값 설정 • 있음 • 없음	데이터 형식에 따릅니다. ('초기값(6-62페이지)'항을 참조하십시오.)
상수	변수 다운로드 시 초기값을 설정하고, 그 이후에는 값을 재기입할 수 없는 속성입니다.	상수로 사용/사용 안 함	–
네트워크 공개	CIP 통신 및 데이터 링크 기능을 사용하여 컨트롤러 외부에서 변수를 읽고 쓸 수 있도록 합니다.	• 공개 안 함 • 공개만 • 입력 • 출력	공개 안 함
엣지	FB의 입력 파라메터의 상승, 하강을 검출합니다. BOOL 형식의 입력 변수에만 설정할 수 있습니다.	• 없음 • 상승 • 하강	없음

*1. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

참고

태스크 간 배타 제어 기능

하나의 태스크에서만 글로벌 변수에 쓸 수 있도록 하면 처리 중에 글로벌 변수의 값이 변화하는 것을 방지합니다. 변수의 속성이 아니라 태스크 설정에서 지정합니다.

● **변수 종류별 속성의 유무**

변수 종류		변수명	데이터 형식	AT(할당대상) 지정	유지	초기값	상수	네트워크 공개	엣지
글로벌 변수		설정 가능	설정 가능	설정 가능	유지 가능	설정 가능	지정 가능	공개 가능	불가능
프로그램	내부 변수	설정 가능	설정 가능	설정 가능	유지 가능	설정 가능	지정 가능	공개 불가능	불가능
	외부 변수	설정 불가능	설정 불가능	설정 불가능	유지 불가능	설정 불가능	지정 가능	공개 불가능	불가능
펑션 블록	내부 변수	설정 가능	설정 가능	설정 가능	유지 가능	설정 가능	지정 가능	공개 불가능	불가능
	입력 변수	설정 가능	설정 가능	설정 불가능	유지 가능	설정 가능	지정 가능	공개 불가능	가능
	출력 변수	설정 가능	설정 가능	설정 불가능	유지 가능	설정 불가능	지정 불가능	공개 불가능	불가능
	입출력 변수	설정 가능	설정 가능	설정 불가능	유지 불가능	설정 불가능	지정 가능	공개 불가능	불가능
	외부 변수	설정 불가능	설정 불가능	설정 불가능	유지 불가능	설정 불가능	지정 가능	공개 불가능	불가능
펑션	내부 변수	설정 가능	설정 가능	설정 불가능	유지 불가능	설정 가능	지정 가능	공개 불가능	불가능
	입력 변수	설정 가능	설정 가능	설정 불가능	유지 불가능	설정 가능	지정 가능	공개 불가능	불가능
	출력 변수	설정 가능	설정 가능	설정 불가능	유지 불가능	설정 불가능	지정 불가능	공개 불가능	불가능
	입출력 변수	설정 가능	설정 가능	설정 불가능	유지 불가능	설정 불가능	지정 가능	공개 불가능	불가능
	외부 변수	설정 불가능	설정 불가능	설정 불가능	유지 불가능	설정 불가능	지정 가능	공개 불가능	불가능

6-3-5 데이터 형식

데이터 형식이란 변수가 표현하는 값의 형식과 범위를 특정하는 속성입니다.

데이터 형식의 종류에 따라 해당 변수 선언 시 확보되는 메모리의 크기가 다릅니다. 메모리의 크기가 클수록 표현할 수 있는 값의 범위가 커집니다.

또한, 명령의 입력/출력/입출력 변수의 데이터 형식은 명령별로 정해져 있습니다. 명령의 인수인 입력/출력/입출력 파라메터의 데이터 형식은 대응하는 입력/출력/입출력 변수의 데이터 형식에 맞춰서 설정하십시오.

기본 데이터 형식과 파생 데이터 형식

데이터 형식에는 사전에 사양이 결정된 기본 데이터 형식과 사용자가 사양을 정의하는 파생 데이터 형식이 있습니다.

● **기본 데이터 형식의 종류**

기본 데이터 형식은 다음과 같이 분류할 수 있습니다.

분류	정의
부울 형식	값이 0 또는 1 중 하나인 데이터 형식입니다.
비트 열 형식	비트의 열로 값을 표현한 데이터 형식입니다.
정수 형식	값이 정수값인 데이터 형식입니다.
실수 형식	값이 실수값인 데이터 형식입니다.
지속 시간 형식	값이 시간(일·시·분·초·밀리초)인 데이터 형식입니다.
시각 형식	값이 시각(시·분·초)인 데이터 형식입니다.
날짜 형식	값이 날짜(년·월·일)인 데이터 형식입니다.
날짜 시각 형식	값이 날짜 시각(년·월·일·시·분·초·밀리초)인 데이터 형식입니다.
문자열 형식	값이 문자열인 데이터 형식입니다.

기본 데이터 형식으로 20종류가 있습니다. 각각의 사양은 다음과 같습니다.

또한, 다음 표의 '데이터 크기(용량값)' 및 '얼라인먼트(한계값)'의 의미는
- 데이터 크기(용량값): 실제 값의 용량입니다.
- 얼라인먼트(한계값): 메모리를 확보할 때의 단위입니다.

분류	데이터 형식	데이터 크기 (용량값)	얼라인먼트 (한계값)	값의 범위	기술 방법
부울 형식	**BOOL**	16비트	2바이트	FALSE, TRUE	BOOL#0, BOOL#1 FALSE, TRUE
비트 열 형식	**BYTE**	8비트	1바이트	BYTE#16#00~FF	BYTE#2#01011010 BYTE#2#0101_1010 BYTE#16#5A (구분 문자 '_'를 사용할 수도 있습니다.)
	WORD	16비트	2바이트	WORD#16#0000~FFFF	
	DWORD	32비트	4바이트	DWORD#16#00000000~ FFFFFFFF	
	LWORD	64비트	8바이트	LWORD#16#0000000000000000~ FFFFFFFFFFFFFFFF	
정수 형식*1	**SINT**	8비트	1바이트	SINT#-128~+127	100 INT#2#00000000_01100100 INT#8#144 INT#10#100 INT#16#64 -100
	INT	16비트	2바이트	INT#-32768~+32767	
	DINT	32비트	4바이트	DINT#-2147483648~+2147483647	
	LINT	64비트	8바이트	LINT#-9223372036854775808~ +9223372036854775807	
	USINT	8비트	1바이트	USINT#0~+255	
	UINT	16비트	2바이트	UINT#0~+65535	
	UDINT	32비트	4바이트	UDINT#0~+4294967295	
	ULINT	64비트	8바이트	ULINT#0~+18446744073709551615	
실수 형식	**REAL**	32비트	4바이트	REAL#-3.402823e+38~-1.175495e-38 0 1.175495e-38~3.402823e+38 +∞/-∞	REAL#3.14 LREAL#3.14 3.14 -3.14 1.0E+6 1.234e4
	LREAL	64비트	8바이트	LREAL#-1.79769313486231e+308~-2.22507385850721e-308 0 2.22507385850721e-308~1.79769313486231e+308 +∞/-∞	
지속시간 형식*1*2	**TIME**	64비트	8바이트	T#-9223372036854.775808ms(T#-106751d_23h_47m_16s_854.775808ms)~T#+9223372036854.775807ms(T#+106751d_23h_47m_16s_854.775807ms)	T#12d3h3s T#3s56ms TIME#6d_10m TIME#16d_5h_3m_4s T#12d3.5h T#10.12s T#61m5s(T#1h1m5s와 동일) TIME#25h_3m
날짜 형식	**DATE**	64비트	8바이트	D#1970-01-01~D#2106-02-06 (1970년 1월 1일~ 2106년 2월 6일)	선두에 'DATE#' 또는 'date#' 또는 'D#' 또는 'd#'을 붙이고, '년-월-일'로 기술합니다. (예) d#1994-09-23
시각 형식*2	**TIME_ OF_DAY**	64비트	8바이트	TOD#00:00:00.000000000~TOD#23:59:59.999999999 (0시 0분 0.000000000초 ~23시 59분 59.999999999초)	선두에 'TIME_OF_DAY#' 또는 'time_of_day#' 또는 'TOD#' 또는 'tod #'을 붙이고, '시:분:초'로 기술합니다. (예) tod#12:16:28.12

분류	데이터 형식	데이터 크기 (용량값)	얼라인먼트 (한계값)	값의 범위	기술 방법
날짜시각 형식*2	**DATE_ AND_ TIME**	64비트	8바이트	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)	선두에 'DT#' 또는 'dt#'을 붙이고, '년-월-일-시:분:초'로 기술합니다. (예) dt#1994-09-23-12:16:28.12
문자열 형식	**STRING**	(반각 문자 수 +1)×8 비트*3	1바이트	문자 코드: UTF-8 0~1986바이트(반각 영숫자로 0~1985문자, 한국어로 0~661문자가 기준입니다). *4 기본값 크기는 256바이트입니다.	'"(반각의 작은 따옴표)로 묶어서 기술합니다. (예) 'OMRON' 'PLC'

*1. 지속 시간 형식과 정수 형식의 변환에는 NanoSecToTime 명령 및 TimeToNanoSec 명령을 사용하십시오. 명령 사양의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.
*2. 비교 명령에서의 변수 비교는 ns 단위로 실행됩니다. 비교의 단위를 변경할 때는 TruncTime 명령/TruncDt 명령/TruncTod 명령을 사용하십시오. 명령 사양의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.
*3. 문자열의 종단에는 NULL(1바이트)이 붙습니다. 따라서 취급하는 최대 문자 수+1문자 분량의 메모리를 확보하십시오. 예를 들어 반각 영숫자의 최대 문자 수가 10인 경우에는 11문자 분량(11바이트)의 STRING 형식 변수를 정의하십시오. STRING[11]
*4. 탭 및 행바꿈 코드 등을 기재하는 경우에는 '$'(반각 달러 기호)를 접두어로 한 이스케이프 문자를 기재합니다. 이스케이프 문자에 대해서는 '문자열 형식(6-88페이지)'을 참조하십시오.

사용상의 주의 사항

전체 변수가 필요로 하는 메모리 용량은 각 변수의 데이터 크기(용량값)의 합계와는 일치하지 않으므로 주의하십시오.
왜냐하면 데이터가 메모리에 배치되는 선두 위치는 자동으로 각 데이터 형식의 '얼라인먼트(한계값)'의 배수 위치에 배치되므로 각 데이터 형식 간에 틈새가 발생하는 경우가 있기 때문 입니다.
예를 들어 데이터 형식의 종류가 동일해도 데이터 형식의 순서에 따라 전체 점유 크기가 다릅니다.

각 데이터 형식의 '얼라인먼트(한계값)'는 구조체 형식 변수 등에서 다른 기기와 데이터 교환을 하는 경우에 데이터 위치를 맞춰야 할 때는 의식해야 합니다.
자세한 내용은 'A-10 변수의 메모리 확보 방법(A-185페이지)'을 참조하십시오.

참고

- 비트 열의 데이터 형식(BYTE, WORD, DWORD, LWORD)은 크기를 비교할 수 없습니다. 크기를 비교하는 경우에는 WORD_TO_UINT 명령 등의 데이터 형식 변환 명령을 사용하여 변수의 데이터 형식을 정수로 변환하십시오.

  예)
  BCD_data : WORD
  IF WORD_BCD_TO _UINT(BCD_data)> UINT#1234 THEN
- 정수의 데이터 형식(SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT)은 논리 연산을 할 수 없습니다. 논리 연산을 하는 경우에는 INT_TO_WORD 명령 등의 데이터 형식 변환 명령을 사용하여 변수의 데이터 형식을 비트 열로 변환하십시오.

  예)
  다음 샘플 프로그램에서는 INT 형식 변수 a의 값이 홀수인 경우에 a에 1이 더해집니다.

```
IF (INT_TO_WORD(a) AND WORD#16#0001) = WORD#16#0001 THEN
   a = a+1;
END_IF;
```

● 파생 데이터 형식의 종류

파생 데이터 형식은 그 사양을 사용자가 정의하는 데이터 형식입니다. Sysmac Studio의 '데이터 형식' 보기에 등록됩니다. 파생 데이터 형식으로는 다음과 같은 것이 있습니다.

종류	내용
구조체 형식	복수의 데이터 형식을 계층적으로 하나로 묶은 데이터 형식입니다.
공용체 형식	동일한 데이터를 경우에 따라 다른 데이터 형식으로 취급하도록 한 데이터 형식입니다.
열거 형식	사전에 준비된 이름 목록 중 하나를 값으로 삼는 데이터 형식입니다.

자세한 내용은 뒤에 설명하는 📖'6-3-6 파생 데이터 형식(6-41페이지)'을 참조하십시오.

● 데이터 형식에 대한 지정

각 데이터 형식에 대해 다음과 같은 배열 지정, 범위 지정이 가능합니다.

종류	내용
배열 지정	동일한 데이터 형식의 요소를 묶어서 선두부터 번호(첨자)로 지정합니다. 기본 데이터 형식 및 파생 데이터 형식으로 지정할 수 있습니다.
범위 지정	사전에 정해진 범위 내의 값을 취하는 것을 명시한 지정입니다. 기본 데이터 형식의 정수 형식으로 지정할 수 있습니다.

자세한 내용은 뒤에 설명하는 📖'6-3-7 데이터 형식의 배열/범위 지정(6-51페이지)'을 참조하십시오.

참고

데이터 형식에는 기본 데이터 형식과 파생 데이터 형식 이외에 POU 인스턴스 형식이 있습니다. POU 인스턴스 형식은 FB의 인스턴스의 데이터 형식입니다. FB의 인스턴스를 생성하려면 인스턴스명을 변수, FB 정의명을 데이터 형식으로 변수 테이블에 등록합니다.

데이터 형식의 사용 제한

각 POU에서 사용할 수 없는 데이터 형식은 다음과 같습니다.

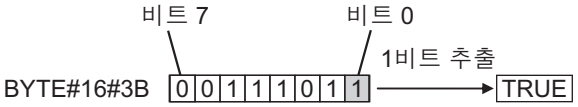
POU 종류	변수 종류	사용할 수 없는 데이터 형식	
		기본 데이터 형식	파생 데이터 형식
프로그램	내부 변수	없음	
	글로벌 변수	없음	
FUN	입력 변수, 출력 변수, 입출력 변수	없음	공용체 형식
	내부 변수	없음	
	반환값	없음	구조체 형식, 공용체 형식
FB	입력 변수, 출력 변수, 입출력 변수	없음	공용체 형식
	내부 변수	없음	

비트 열 형식, 실수 형식, 문자열 형식의 데이터 형식

아래에서 비트 열 형식, 실수 형식, 문자열 형식의 데이터 형식을 설명합니다.

- **비트 열 형식의 데이터 형식**

  비트 열 형식 변수의 비트 위치는 최하위 자리를 '비트 0'으로 합니다.
  또한, 비트 값은 1 또는 0 으로 표현하지만 1비트의 값을 BOOL 형식 변수로 표현하는 경우에는 1 이 TRUE, 0이 FALSE입니다.



- **실수 형식(REAL 형식, LREAL 형식)**

  실수(REAL 형식, LREAL 형식)는 실수 형식의 데이터 형식을 갖는 데이터 형식입니다.
  표현 방법 및 연산 처리에 대해 설명합니다.

데이터 크기

REAL 형식의 데이터 크기는 32비트, LREAL 형식의 데이터 크기는 64비트입니다.

데이터 형식

부동 소수점 형식이란, 실수를 부호, 지수, 가수로 표현하는 형식입니다.
어떤 실수를 다음 식에 대입하여 표현한 경우의 s의 값이 부호, e의 값이 지수, f의 값이 가수입니다.

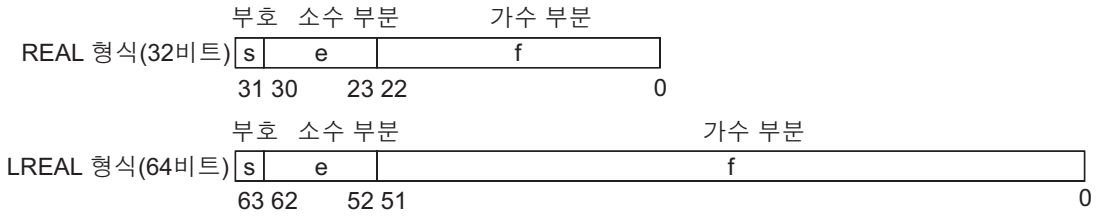
- REAL 형식

  수치 = $(-1)^s 2^{e-127}(1 + f \times 2^{-23})$

- LREAL 형식

  수치 = $(-1)^s 2^{e-1023}(1 + f \times 2^{-52})$

부동 소수점 형식의 데이터 형식은 IEEE754 규격을 준수합니다. 각각 다음과 같습니다.



(예) REAL 형식으로 -86.625 를 표현하는 경우

1 **음수이므로 s=1입니다.**

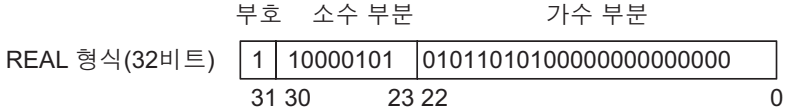
2 **86.625 를 2진수로 표현하면 1010110.101이 됩니다.**

3 **위의 값을 정규화하면 1.010110101×2^6이 됩니다.**

4 **위 식에서 e-127=6이 되고, e=133(2진수로는 10000101)이 됩니다.**

5 **1.010110101의 소수점 이하는010110101입니다. 여기에 가수 부분 23비트에 충분하지 않은 0을 부가한 것이 f가 됩니다. 따라서 f=01011010100000000000000입니다.**

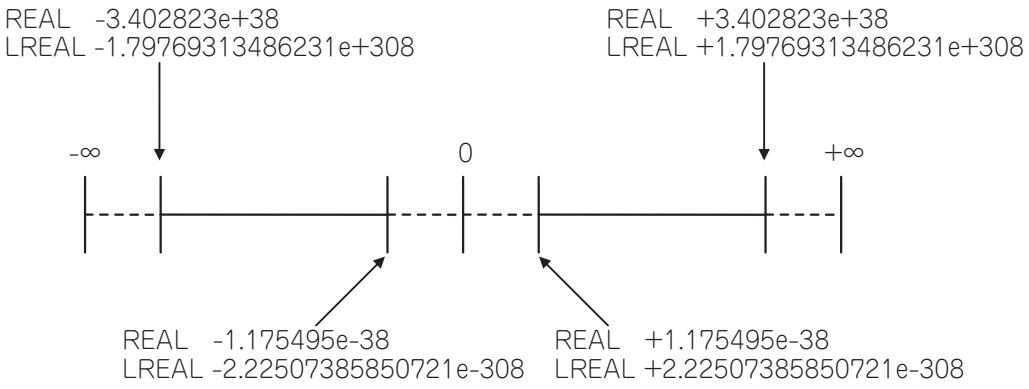
따라서 -86.625는 다음 그림과 같이 표현할 수 있습니다.



유효 범위

REAL 형식, LREAL 형식의 유효 범위는 다음 표와 같습니다.
0 근방에 표현할 수 없는 값의 범위가 존재합니다.

데이터 형식	- ∞	음수	0	양수	+ ∞
REAL	- ∞	-3.402823e+38~ -1.175495e-38	0	+1.175495e-38~ +3.402823e+38	+ ∞
LREAL	- ∞	-1.79769313486231e+308~ -2.22507385850721e-308	0	+2.22507385850721e-308~ +1.79769313486231e+308	+ ∞



특수 값

+ ∞, - ∞, +0, -0, NaN(숫자가 아님)을 특수 값이라고 합니다.
NaN(숫자가 아님)은 부동 소수점 수로 인정할 수 없는 데이터이며, 숫자로 취급되지 않습니다.
+0과 -0은 모두 수학적으로는 0과 동일한 의미이지만 연산 처리 내용에 차이가 있습니다. 자세한 내용은 뒤에 설명합니다.
특수 값인 부호 s, 지수 e, 가수 f의 값은 다음 표와 같습니다.

데이터 형식	특수 값	부호 s	지수 e	가수 f
REAL	+ ∞	0	255	0
	- ∞	1	255	0
	+0	0	0	0
	-0	1	0	0
	NaN(숫자가 아님)	-	255	0 이외

데이터 형식	특수 값	부호 s	지수 e	가수 f
LREAL	+ ∞	0	2047	0
	- ∞	1	2047	0
	+0	0	0	0
	-0	1	0	0
	NaN(숫자가 아님)	-	2047	0 이외

비정규화 수

부동 소수점 형식은 0 근방(절대값이 매우 작은 값)을 표현할 수 없습니다. 따라서 0 근방의 유효 범위를 확대하기 위해 설정된 것이 비정규화 수입니다. 비정규화 수는 통상의 데이터 형식으로 표현되는 값(정규화 수)보다 절대값의 작은 값을 표현할 수 있습니다.
지수 e = 0, 가수 f ≠ 0인 것이 비정규화 수로 간주되고, 값은 다음과 같이 표현됩니다.

- REAL 형식
  수치 = $(-1)^s 2^{-126}(f \times 2^{-23})$
- LREAL 형식
  수치 = $(-1)1^s 2^{-1022}(f \times 2^{-52})$

(예) REAL 형식으로 0.75×2^{-127}를 표현하는 경우

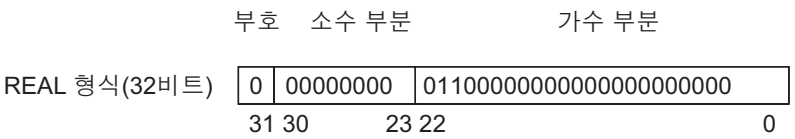
1 양수이므로 s=0이 됩니다.

2 0.75를 2진수로 표현하면 0.11입니다.

3 $(0.11)_2 \times 2^{-127} = 2^{-126}(f \times 2^{-23})$에서 $f = (0.11)_2 \times 2^{22}$가 됩니다.

4 위 식에서 f=01100000000000000000000이 됩니다.

따라서 0.75×2^{-127}은 다음 그림과 같이 표현할 수 있습니다.



비정규화 수는 정규화 수보다 유효 자리수가 적습니다. 따라서 정규화 수의 연산 결과가 비정규화 수가 되는 경우 또는 연산 도중의 결과가 비정규화 수가 되는 경우에 연산 결과의 유효 자리수가 정규화 수의 유효 자리수 미만이 될 수 있습니다.

연산 처리

부동 소수점 형식은 근사값 표현이기 때문에 참값과의 사이에 오차가 존재하는 경우가 있습니다. 또한, 값의 유효 범위에도 제한이 있습니다. 따라서 연산할 때는 반올림해서 오버플로, 언더플로 처리를 합니다.

사용상의 주의 사항

일반적으로 프로세서 등의 하드웨어가 다르면 실수 형식의 연산 결과가 달라지는 경우가 있습니다. 다른 CPU 유니트 형식으로 프로그램 및 라이브러리를 재이용하는 경우에는 실수 형식의 연산 결과를 충분히 확인하십시오.

반올림

참값이 가수의 유효 자리수를 초과한 경우에 다음과 같은 규칙에 따라 값을 반올림합니다.

- 부동 소수점 형식으로 표현할 수 있는 값 중에서 가장 참값에 가까운 값을 연산 결과로 합니다.
- 부동 소수점 형식으로 표현할 수 있는 값 중에서 가장 참값에 가까운 값이 2개 있는 경우(참값이 2개 근사값의 중앙값인 경우) 가수의 최하위 자리의 비트 값이 0인 쪽을 연산 결과로 합니다.

사용상의 주의 사항

두 값이 같은지 여부를 검증할 때 참값과의 오차의 배려

실수는 부동 소수점 형식으로 표현하기 때문에 참값과의 사이에 오차가 존재하는 경우가 있습니다. 두 값이 같은지 여부를 검증하는 경우에 이 오차 때문에 그림과 같은 결과를 얻지 못할 가능성이 있습니다.

예를 들어 0.1+0.2와 0.3의 비교 boolv := (0.1 + 0.2 = 0.3);을 실행한 경우에 BOOL 형식 변수 boolv의 값은 TRUE가 되지 못하고, FALSE가 됩니다.

이런 사태를 피하기 위해 실수의 두 값이 같은지 여부를 검증할 때는 EQ, =, NE, <>를 사용하지 마십시오. 그 대신 크기 비교 명령을 사용하여 양자의 차의 절대값이 충분히 작은 허용 범위 미만인지 여부를 검증하십시오.

예를 들어 REAL 형식 변수 real_a와 real_b의 합과 0.3이 같은지 여부를 검증하는 경우에는 다음과 같이 기술하십시오. boolv의 값이 TRUE가 되면 양자는 같다고 판단합니다.

```
boolv := (ABS((real_a + real_b) - 0.3) < 0.000001); //0.000001까지의 오차를
                                                    //허용 범위로 하는 경우
```

오버플로, 언더플로

참값의 절대값이 부동 소수점 형식으로 표현할 수 있는 값보다 커지는 것을 오버플로라고 합니다.
반대로 부동 소수점 형식으로 표현할 수 있는 값보다 작아지는 것을 언더플로라고 합니다.

- 오버플로인 경우에 참값의 부호가 양이면 연산 결과를 +∞로 합니다. 참값의 부호가 음이면 연산 결과를 -∞로 합니다.
- 언더플로인 경우에 참값의 부호가 양이면 연산 결과를 +0으로 합니다. 참값의 부호가 음이면 연산 결과를 -0으로 합니다.

특수 값의 연산

특수 값(+ ∞, - ∞, +0, -0, NaN(숫자가 아님))의 연산은 다음 규칙에 따릅니다.

- +∞와 -∞의 가산 결과는 NaN(숫자가 아님)이 됩니다.
- 부호가 동일한 무한대끼리의 뺄셈 결과는 NaN(숫자가 아님)이 됩니다.
- +0과 무한대, -0과 무한대의 곱셈 결과는 NaN(숫자가 아님)이 됩니다.
- +0끼리의 나눗셈 결과, -0끼리의 나눗셈 결과, 무한대끼리의 나눗셈 결과는 NaN(숫자가 아님)이 됩니다.
- +0과 -0의 덧셈 결과는 +0이 됩니다.
- +0끼리의 뺄셈 결과, -0끼리의 뺄셈 결과는 +0이 됩니다.
- NaN(숫자가 아님)을 포함한 사칙 연산의 결과는 NaN(숫자가 아님)이 됩니다.
- 비교 명령(Cmp 명령 등)에서는 +0과 -0은 같다고 판단합니다.
- 비교 명령에서는 비교 대상에 NaN(숫자가 아님)을 포함하는 경우에 항상 같지 않다고 판단합니다.

● **문자열 형식의 데이터 형식**

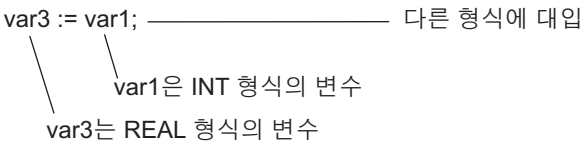
문자열 형식 변수의 종단에는 종단을 나타내는 문자로 NULL 문자(문자 코드 BYTE#16#00)가 부가됩니다.

데이터 형식 변환

다른 데이터 형식의 변수를 사용한 경우에 데이터 형식의 변환이 자동으로 발생할 수 있습니다. 또는 사용자가 형식 변환 명령에 의해 변환할 수도 있습니다.

데이터 형식 변환의 개념

모든 변수는 '데이터 형식'을 가지므로 프로그램에서는 데이터 형식에 따른 조작을 실행해야 합니다. 예를 들어 대입식에서 좌변과 우변은 동일한 데이터 형식을 가져야 합니다. 그러나 실제로는 기능의 실현을 위해 다른 데이터 형식의 변수에 대입해야 하는 경우가 있습니다.
예)



이와 같은 경우에 var1의 데이터를 var3의 데이터 형식에 적합하게 하기 위한 변환이 필요합니다. 이와 같은 변환을 '데이터 형식 변환' 또는 생략하여 '형식 변환'이라고 합니다.

- **데이터 형식 변환이 발생하는 경우**

  데이터 형식의 변환은 다음과 같은 2가지 경우에 실행합니다.

  (1) 데이터 형식 변환 명령을 사용하여 사용자가 명시적으로 변환

  (2) 대입/명령에 의한 자동적인 변환

  - ST 언어의 대입
  - 래더도 언어에서의 결선

참고

정수 형식과 TIME 형식의 변환에는 시간/시각 명령의 NanoSecToTime 명령 또는 TimeToNanoSec 명령을 사용하십시오. 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

6-3-6 파생 데이터 형식

파생 데이터 형식은 기본 데이터 형식을 바탕으로 구성을 정의한 데이터 형식입니다.
파생 데이터 형식으로는 다음과 같은 것이 있습니다.

- 구조체 형식
- 공용체 형식
- 열거 형식

파생 데이터 형식을 작성할 때, 데이터 형식명의 문자 수 등의 제약은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.

참고

NJ/NX 시리즈에서는 시스템 정의의 파생 데이터 형식으로 다음 3종류가 사전에 준비되어 있습니다.

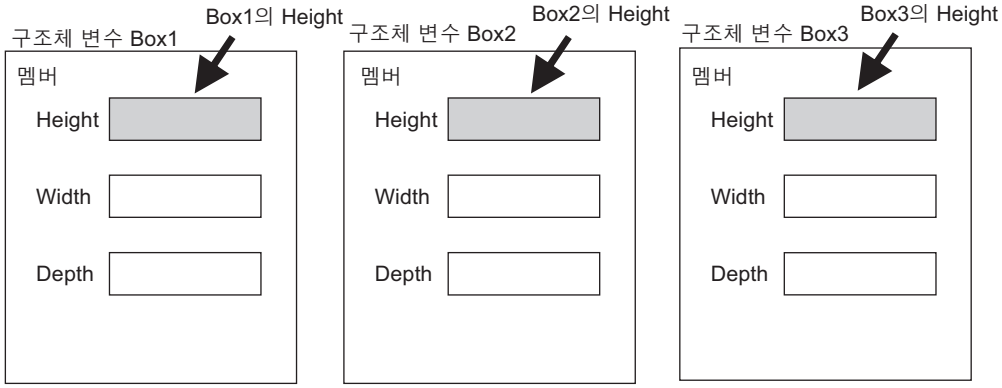
- 시스템 정의 변수의 구조체 형식
- 명령의 입력/출력/입출력 변수가 사용하는 구조체 형식 등의 파생 데이터 형식
- 고기능 유니트의 확장 메모리용 구조체 형식(단, 유니트 편집기 상에서 등록하여 사용 가능)

구조체 형식

구조체 형식은 데이터 형식이 동일한 또는 다른 복수의 데이터를 하나로 묶어서 취급하는 파생 데이터 형식입니다.
데이터를 구조체 형식으로 하면 데이터의 등록·변경 등의 관리를 간단히 할 수 있습니다.

예를 들어 "Box"라는 구조체 형식을 정의하는 경우에 멤버로 "Width", "Height", "Depth"의 3가지 데이터를 그룹화하여 정의합니다.
이 구조체 형식의 데이터 형식을 사용하여 "Box1"라는 변수를 등록하면 프로그램에서는 "Box1. Width", "Box1.Height"와 같이 '.(마침표)'를 사용하여 데이터를 계층적으로 지정할 수 있게 됩니다.

만일 새로운 다른 상자의 데이터가 필요했을 때는 "Box1"과 동일한 요령으로 "Box2"를 변수 테이블에 등록하기만 하면 됩니다.



사용하는 명령의 변수에서 구조체 형식이 사용된 경우에 입력, 출력 또는 입출력 파라메터용으로 해당 구조체 형식을 선택하여 변수를 등록해야 합니다.
예) 통신 명령

- **구조체 형식 변수의 표현**

각 멤버의 지정

구조체 형식을 구성하는 데이터를 '멤버'라고 합니다.
구조체 형식의 멤버는 구조체 형식 전체를 나타내는 변수명에 반각 마침표.와 멤버명을 붙여 써서 나타냅니다.
또한, 구조체 형식의 멤버가 다시 구조체 형식이어도 상관 없습니다.

예) abc.x 구조체 형식 변수 abc의 멤버 x
abc.Order.z 구조체 형식 변수 abc의 멤버(구조체 형식) Order의 멤버 z

전체 멤버(구조체 전체)의 지정

구조체를 구성하는 전체 멤버를 '구조체 전체'라고 합니다.
구조체 전체는 구조체의 변수명으로만 써서 나타냅니다.
위의 예에서는 'abc'라고 써서 나타냅니다.

- **구조체 형식 작성 방법**

1 **데이터 형식 테이블에서 구조체 형식의 데이터 형식을 작성합니다.**
데이터 형식명, 멤버, 데이터 형식을 지정합니다.

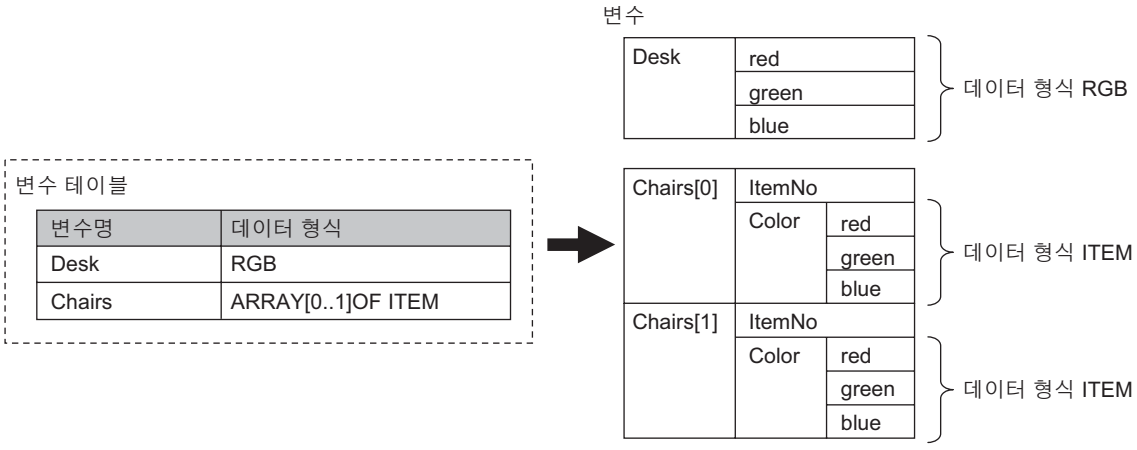
데이터 형식 테이블

이름	멤버	데이터 형식
RGB	red	INT
	green	INT
	blue	INT
ITEM	ItemNo	INT
	Color	RGB

2 **변수 테이블에서 데이터 형식으로 위의 구조체 형식 데이터 형식, 멤버명을 지정하여 변수를 등록합니다.**

변수 테이블

변수명	데이터 형식
Desk	RGB
Chairs	ARRAY[0..1]OF ITEM



● **구조체 형식의 사양**

구조체 형식의 사양은 다음과 같습니다.

항목	사양
구조체 형식명	알파벳의 대문자/소문자는 구별하지 않습니다. 금칙 문자 및 문자 수의 제한은 변수명과 동일합니다.
멤버의 데이터 형식	뒤에 설명하는 '구조체의 멤버의 데이터 형식' 표를 참조하십시오.
멤버의 속성	멤버명 코멘트
멤버 수	1~2048
구조체 형식의 계층 깊이	최대 8계층 (단, 멤버명을 포함하는 변수명이 511바이트 이하)
1구조체 변수의 최대 크기	NX701-□□□□: 8MB NX102-□□□□: 4MB NX1P2-□□□□: 1MB NJ501-□□□□: 4MB NJ301-□□□□: 2MB NJ101-□□□□: 2MB

구조체 형식 멤버의 데이터 형식

분류	데이터 형식	사용
기본 데이터 형식	부울 형식, 비트 열 형식, 정수 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시간 형식, 날짜 시각 형식, 문자열 형식	가능
	부울 형식, 비트 열 형식, 정수 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시간 형식, 날짜 시각 형식, 문자열 형식의 배열 지정	가능
파생 데이터 형식	구조체 형식(*), 공용체 형식, 열거 형식 * 재귀 또는 순환할 수 없습니다 (프로그램 확인 시 에러가 됩니다).	가능
	구조체 형식, 공용체 형식, 열거 형식의 배열 지정	가능
POU 인스턴스 형식		불가능

● **배열 지정과 구조체 형식**

구조체 형식을 요소로 하는 배열 지정을 설정할 수도 있습니다. 또한, 배열 지정을 멤버로 하는 구조체 형식을 설정할 수도 있습니다.

● **구조체 멤버의 오프셋 지정**

멤버의 오프셋 지정에 의해 구조체 형식별로 멤버의 메모리 배치를 임의로 설정할 수 있습니다. 이것으로 CJ 시리즈 CPU 유니트 및 외부 기기와 태그 데이터 링크하는 경우에 양자의 구조체 형식 멤버의 메모리 배치를 일치시킬 수 있습니다.

구조체 형식 멤버의 오프셋 종별은 'NJ', 'CJ', '임의' 중에서 선택할 수 있습니다. 'NJ'를 선택한 경우에는 자동으로 NJ/NX 시리즈에 최적인 메모리 배치입니다. NJ/NX 시리즈의 메모리 배치에 대해서는 📖'A-10 변수의 메모리 확보 방법(A-185페이지)'을 참조하십시오. CJ 시리즈 CPU 유니트와 태그 데이터 링크하는 예에 대해서는 📖'A-10-2 유의해야 하는 케이스 예(A-194페이지)'를 참조하십시오.

오프셋 종별의 의미는 다음과 같습니다.

오프셋 종별	의미
NJ	자동으로 NJ/NX 시리즈에 최적인 메모리 배치가 되고, 가장 고속으로 동작합니다.
CJ	자동으로 CJ 시리즈와 동일한 메모리 배치가 됩니다. CJ 시리즈 CPU 유니트의 메모리 배치에 맞출 수 있습니다.
임의	각 멤버의 메모리 배치를 임의로 설정할 수 있습니다. CJ 시리즈 이외의 외부 기기와 메모리 배치를 맞추는 경우에 사용합니다.

버전 관련 정보

멤버의 오프셋 지정 사용 여부는 CPU 유니트의 유니트 버전과 Sysmac Studio의 버전 조합에 따라 다음과 같이 달라집니다.

CPU 유니트의 유니트 버전	Sysmac Studio의 버전		
	Ver.1.01 이하	Ver.1.02	Ver.1.03 이상
Ver.1.01 이상	불가능	가능*1	가능
Ver.1.00	불가능	불가능	불가능

*1. 오프셋 종별의 선택 기능은 없습니다. 멤버의 오프셋은 임의로 설정합니다.

오프셋 설정

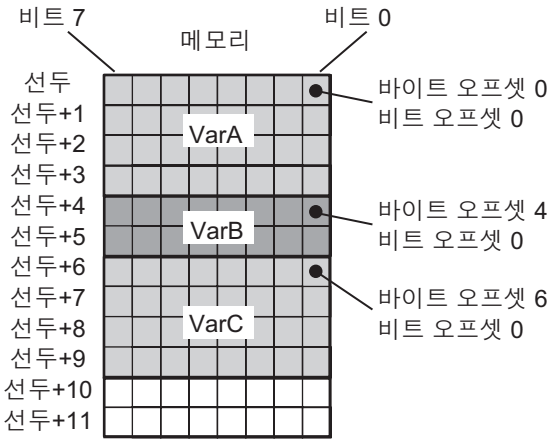
오프셋 종별을 '임의'로 한 경우에 구조체 형식의 멤버별로 메모리 배치 오프셋을 설정할 수 있습니다. 오프셋에는 '바이트 오프셋'과 '비트 오프셋'이 있습니다. 오프셋 종별을 'NJ', 'CJ'로 한 경우에는 자동으로 메모리 배치가 결정되므로 오프셋 설정은 필요 없습니다.
오프셋의 의미는 다음과 같습니다.

오프셋	의미	단위	값의 범위
바이트 오프셋	각 멤버의 구조체 형식 선두의 오프셋을 설정합니다. 기본 데이터 형식과 파생 데이터 형식의 모든 데이터 형식으로 사용할 수 있습니다.	바이트	*1
비트 오프셋	각 멤버의 바이트 오프셋으로 설정한 바이트 위치의 선두에서의 오프셋을 설정합니다.	비트	0~63

*1. NX 시리즈 CPU 유니트인 경우에는 0~8191, NJ 시리즈 CPU 유니트인 경우에는 0~1023이 됩니다.

예)
구조체 형식 편집기 설정이 다음과 같은 경우의 메모리 배치를 나타냅니다.

이름	데이터 형식	오프셋 종별	바이트 오프셋	비트 오프셋
StrA	STRUCT	임의		
VarA	DINT		0	0
VarB	INT		4	0
VarC	DINT		6	0



오프셋 설정 가능 여부

오프셋 종별을 '임의'로 한 경우에도 데이터 형식에 따라 오프셋이 고정인 것이 있습니다. 오프셋 설정 가능 여부를 다음과 같습니다.

분류	데이터 형식	바이트 오프셋	비트 오프셋
부울 형식	**BOOL**	설정 가능	설정 가능
비트 열 형식	**BYTE, WORD, DWORD, LWORD**	설정 가능	고정
정수 형식	**SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT**	설정 가능	고정
실수 형식	**REAL, LREAL**	설정 가능	고정
지속 시간 형식	**TIME**	설정 가능	고정
날짜 형식	**DATE**	설정 가능	고정
시각 형식	**TIME_OF_DAY**	설정 가능	고정
날짜 시각 형식*1	**DATE_AND_TIME**	설정 가능	고정
문자열 형식	**STRING**	설정 가능	고정
배열 지정		설정 가능	요소의 데이터 형식이 BOOL인 경우만 설정 가능
구조체 형식		설정 가능	고정
공용체 형식		설정 가능	고정
열거 형식		설정 가능	고정
POU 인스턴스 형식		고정	고정

멤버의 오프셋 지정의 제약

멤버의 오프셋 지정을 하는 경우에는 다음과 같은 제약이 있습니다. 멤버의 오프셋 지정을 한 구조체 형식을 멤버에 포함하는 구조체 형식도 동일한 제약이 있습니다.

- 오프셋 종별을 '임의'로 한 경우에 해당 구조체 형식의 모든 멤버에 대해 오프셋 설정이 필요합니다.
- 오프셋 지정한 구조체 형식의 멤버는 초기값을 임의의 값으로 설정할 수 없습니다. 초기값은 데이터 형식별로 기본값으로 결정된 값입니다. 자세한 내용은 📖'초기값 지정이 공백인 경우(6-63페이지)'를 참조하십시오.
- 구조체 형식의 메모리 크기는 각 멤버의 크기, 데이터 형식의 얼라인먼트(한계값)와 메모리 배치에 의해 결정됩니다.

멤버의 오프셋 지정에 관계되는 에러

멤버의 오프셋 지정에 관계되는 에러는 다음과 같습니다.

에러명	의미	오프셋 종별	대처 방법
오프셋 범위 외 에러	오프셋에 범위 외의 값이 설정되었습니다.	임의	오프셋의 값을 적절하게 변경하십시오.
오프셋 미설정 에러	오프셋이 설정되지 않은 멤버가 있습니다.	임의	모든 멤버에 오프셋을 설정하십시오.
메모리 배치 중복 에러	멤버의 메모리 배치가 중복되었습니다.	임의	오프셋의 값을 적절하게 변경하십시오.
초기값 설정 에러	변수 테이블 작성 시 오프셋 지정된 구조체 형식의 멤버에 초기값을 설정하려고 합니다.	CJ, 임의	초기값을 설정하지 마십시오.

- **구조체 전체를 파라메터로 하는 명령**

명령 중에는 구조체 변수를 파라메터로 전달할 수 있는 것이 있습니다. 그 경우에 입력 파라메터로 구조체 전체만 지정합니다.

예) MOVE 명령에 구조체 형식 멤버를 하나만 전달하는 경우와 구조체 변수 전체를 전달하는 경우

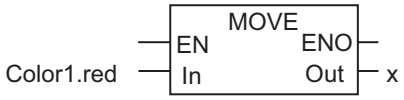
데이터 형식 테이블

이름	멤버	데이터 형식
RGB	red	UINT
	green	UINT
	blue	UINT

변수 테이블

변수명	데이터 형식
Color1	RGB

■구조체 형식 멤버를 1개만 지정



RGB의 구조체 형식 변수 Color1의 멤버 red만을 x에 전송

■구조체 형식 변수 전체를 지정



RGB의 구조체 형식 변수 Color1 전체를 x에 전송

특정 명령의 시스템 정의 구조체 형식 입력 변수에 값을 전달하는 방법

명령에 따라서는 입력 변수에 사전에 준비된 구조체 형식의 변수를 사용하는 특정 명령이 있습니다.

예) 시리얼 통신 명령의 입력 변수 'Port'(상대 포트 지정)는 데이터 형식명 '_sPORT'의 구조체.

그 특정 명령을 사용하는 경우에 다음과 같은 순서로 구조체 형식의 사용자 정의 변수를 작성하여 입력 파라메터에 지정하십시오.

1 **Sysmac Studio에는 명령용 시스템 정의의 데이터 형식이 사전에 등록되어 있습니다.**

Sysmac Studio에서 그 시스템 정의의 데이터 형식을 선택하여 사용자 정의의 구조체 형식 변수를 변수 테이블에 등록합니다.

2 **그 구조체 형식 변수의 각 멤버의 값을 사용자 프로그램에 의해 또는 초기값을 이용하여 설정합니다.**

3 **그 구조체 형식 변수를 명령의 입력 파라메터에 지정합니다.**

공용체 형식

공용체 형식은 동일한 데이터에 대해 복수의 다른 데이터 형식으로 액세스할 수 있도록 한 파생 데이터 형식입니다.

액세스 가능한 복수의 다른 데이터 형식으로 예를 들어 요소 수 16인 BOOL 형식의 배열 지정 및 16개인 BOOL 형식과 WORD 형식을 지정합니다.

- **공용체 형식의 표현**

각 멤버의 지정

공용체 형식을 정의할 때, 액세스 가능한 각 데이터 형식으로 이름을 붙입니다. 이 이름을 '멤버'라고 합니다.

공용체 형식의 멤버는 공용체 형식 전체를 나타내는 변수명에 반각 마침표.와 멤버명을 붙여서 나타냅니다.

예)
공용체 형식의 데이터 형식으로 예를 들어 'My Union'을 다음과 같이 정의

데이터 형식 정의:

이름	멤버	데이터 형식
My Union	data	WORD
	bit	ARRAY [0..15] OF BOOL

변수 테이블:

변수명	데이터 형식
Output	My Union

이라고 했을 때

Output.bit[0]: 공용체 형식 Output을 16비트 BOOL 형식의 배열 지정 변수로 간주한 경우의 요소 번호 0 즉 0번째 비트의 값
Output.data: 공용체 형식 Output을 1개의 WORD 형식 변수로 간주한 경우의 값

전체 멤버(공용체 전체)의 지정

공용체를 구성하는 전체 멤버를 '공용체 전체'라고 합니다.
공용체 전체는 공용체 변수명으로만 써서 나타냅니다.
위의 예에서는 'Output'이라고 써서 나타냅니다.

- **공용체 형식 작성 방법**

1 **공용체 형식 테이블에서 공용체 형식의 데이터 형식을 작성합니다.**
데이터 형식명 및 복수의 다른 데이터 형식을 그 멤버로 지정합니다.

2 **변수 테이블에서 데이터 형식으로 위의 공용체 형식을 지정하여 변수를 등록합니다.**

예)
공용체 형식의 데이터 형식으로 OUT16_ACCESS를 정의합니다.
이 공용체 형식의 멤버는 요소 수 16인 BOOL 형식의 배열 지정 및 WORD 형식으로 합니다.
데이터 형식 OUT16_ACCESS를 사용한 변수 Output을 등록합니다.
변수 Output에는 16비트 중 임의의 1비트에 BOOL 형식의 값을 읽고 쓰는 것도 WORD 형식의 값을 읽고 쓰는 것도 가능합니다.

데이터 형식 정의

이름	멤버	데이터 형식
OUT16ACCESS	BoolData	ARRAY[0..15] of BOOL
	ByteData	ARRAY[0..1] of BYTE
	WordData	WORD

변수명	데이터 형식
Output	OUT16ACCESS

BoolData [15]	・・・	BoolData [8]	BoolData [7]	・・・	BoolData [0]
ByteData [1]			ByteData [0]		
WordData					

```
Output.WordData := WORD#16#1234;
```

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BoolData	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
ByteData	16#12								16#34							
WordData	16#1234															

```
Output.BoolData[11] :=TRUE;
```

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BoolData	*	*	*	*	1	*	*	*	*	*	*	*	*	*	*	*
ByteData	ByteData[1]의 4bit째가 TRUE								변화 없음							
WordData	WordData의 11bit째가 TRUE															

- **공용체 형식의 사양**

항목	사양
멤버에 지정할 수 있는 데이터 형식	뒤에 설명하는 '공용체 형식의 멤버에 지정할 수 있는 데이터 형식' 표를 참조하십시오.
멤버 수	최대 4
초기값 설정	불가능 반드시 제로가 됩니다.

공용체 형식의 멤버에 지정할 수 있는 데이터 형식

분류	데이터 형식	사용
기본 데이터 형식	BOOL 형식, 비트 열 형식	가능
	BOOL 형식, 비트 열 형식의 배열 지정	가능
	그 밖의 기본 데이터 형식	불가능
파생 데이터 형식	구조체 형식, 공용체 형식, 열거 형식의 배열 지정	불가능
POU 인스턴스 형식	−	불가능

- **제한 사항**
  - 공용체의 초기값은 제로 고정입니다.
  - 공용체 전체 전송은 할 수 없습니다.
  - POU의 파라메터에 공용체 전체를 지정할 수 없습니다.

열거 형식(ENUM)

열거 형식은 변수의 값을 '열거자'라고 부르는 문자로 표현한 파생 데이터 형식입니다.
사전에 해당 변수가 취득하는 값을 열거자(문자)로 설정합니다.
열거 형식을 사용하면 변수 값의 의미를 사람이 이해하기 쉬워집니다.

- **열거 형식의 표현**

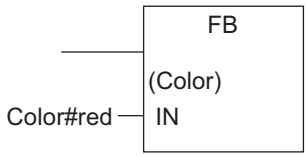
  열거 형식을 정의할 때, 변수가 취득하는 값에 열거자라고 하는 이름을 붙입니다.

- **열거 형식 작성 방법**

  1 열거 형식 테이블에서 열거 형식의 데이터 형식을 작성합니다.
  취득하는 열거자와 대응하는 값을 설정합니다.

  2 변수 테이블에서 데이터 형식으로 위의 열거 형식을 지정하여 변수를 등록합니다.

  예)
  열거 형식의 데이터 형식으로 Color를 작성합니다. 취득하는 열거자는 'red', 'yellow', 'green'의 3가지로 합니다. 열거자에 대응하는 숫자는 red=0, yellow=1, green=2로 합니다.
  데이터 Color를 사용한 변수 DiscColor를 등록합니다. 변수 DiscColor가 취득하는 값은 red(0), yellow(1), green(2) 중 하나입니다.



열거형 테이블

데이터 형식	
Color	ENUM
열거자	값
red	0
yellow	1
green	2

변수 테이블

변수명	데이터 형식
DiscColor	Color

- **열거 형식의 사양**

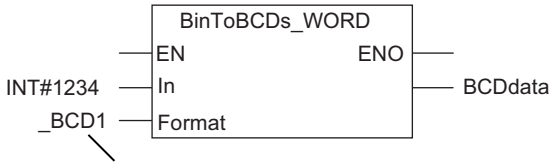
항목	사양
열거자	반각 영숫자만으로 하고, 알파벳의 대문자/소문자를 구별하지 않습니다. 금칙 문자는 변수명과 동일합니다. 동일한 열거자를 복수 지정한 경우에는 빌드 시 에러가 됩니다. 사용자 프로그램 내에 어떤 변수와 동일한 열거자를 지정한 경우 및 다른 열거 형식에 존재하는 열거자와 동일한 이름을 지정한 경우도 빌드 시 에러가 됩니다.
값	유효 범위는 -2147483648~+2147483647의 정수값입니다. 연속된 값이 아니어도 됩니다. 동일한 값을 복수 지정한 경우에는 빌드 시 에러가 됩니다. 주의. 열거 형식 변수의 크기는 비교할 수 없습니다. 일치/불일치만 비교할 수 있습니다.
열거자의 수	1~2048개

- **열거자를 FB, FUN의 파라메터로 하는 경우의 기술**

  열거자를 FB, FUN의 파라메터로 전달하는 경우에 다음과 같은 2가지 기술 방법이 있습니다.

열거자만 기술

파라메터가 열거 형식으로 지정된 FB, FUN인 경우에 열거자만 기술할 수 있습니다.
예) BCDToBins_WORD 명령에 열거자를 전달하는 경우

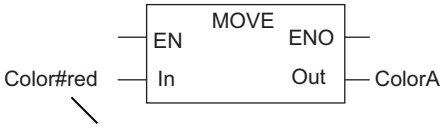


BinToBCDs_WORD 명령의 입력 변수 Format은 열거 형식의 지정이 있습니다.
따라서 열거자만의 형식으로 기술할 수 있습니다.

열거 형식#열거자의 기술

파라메터의 데이터 형식으로 특별히 지정이 없는 FB, FUN인 경우에 열거자만 기술하는 것은 무효입니다. 빌드 시 에러가 됩니다. 파라메터가 열거자임을 명시하기 위해 열거 형식#열거자의 형식으로 기술합니다.

예) MOVE 명령에 열거자를 전달하는 경우



MOVE 명령의 입력 변수 In은 데이터 형식의 지정이 없습니다.
열거자를 전달하는 경우에 열거 형식 #열거자 형식으로로 기술합니다.

참고

열거 형식의 파라메터가 지정된 FB, FUN인 경우에 파라메터를 열거 형식#열거자 형식으로 기술해도 됩니다.
따라서 위의 BinToBCDs_WORD 명령인 경우에 Format 변수에 전달하는 파라메터를 _eBCD_FORMAT# _BCD1이라고 기술해도 됩니다.

- **값 확인**

  명령 실행에 따라 열거 형식 변수에 값이 쓰일 때, 그 값이 열거 형식의 열거자로 정의된 것이 아니라도 에러가 되지는 않습니다. 따라서 값이 열거 형식의 열거자로 정의된 것인지 여부를 확인해야 하는 경우에는 사용자 프로그램 중에서 값을 확인하는 처리를 하십시오.

6-3-7 데이터 형식의 배열/범위 지정

각 데이터 형식의 변수에는 다음과 같은 속성을 지정할 수 있습니다.

- 배열 지정
  - 고정 길이 배열 지정
  - 가변 길이 배열 지정
- 범위 지정

배열 지정(ARRAY[]OF)

배열 지정은 동일한 속성의 데이터를 묶어서 하나의 그룹으로 취급하는 데이터 형식에 대한 지정입니다.
기본 데이터 형식, 파생 데이터 형식에 대해 지정할 수 있습니다.
모션 제어 시의 좌표값과 같이 동일한 개념의 데이터를 여러 개 취급하는 경우에 편리합니다.

● **배열 지정의 표현**

각 요소의 지정

배열 지정을 구성하는 각 데이터를 '요소'라고 합니다.
배열 지정 요소는 배열 지정 전체를 나타내는 배열 지정 변수명에 선두에서부터 센 첨자(요소 번호)를 붙여서 나타냅니다.
첨자는 반각 대괄호([ ])로 묶습니다. 첨자는 상수, 변수 모두 상관 없습니다. 또한, ST 언어인 경우에는 첨자를 식으로 표현할 수도 있습니다.

예)

변수 테이블

변수명	데이터 형식
Mem	ARRAY[0..99] OF INT

```
x:=10;
Mem[x]
```

배열 지정 변수 Mem의 요소 번호 x(변수 x의 값은 10이므로 요소 번호 10)

변수 테이블

변수명	데이터 형식
Data	ARRAY[0..99] OF INT

```
x:=10;
y:=20;
Data[x+y]
```

배열 지정 변수 Data의 요소 번호(x+y)(변수 x의 값은 10, 변수 y의 값은 20이므로 요소 번호 30)

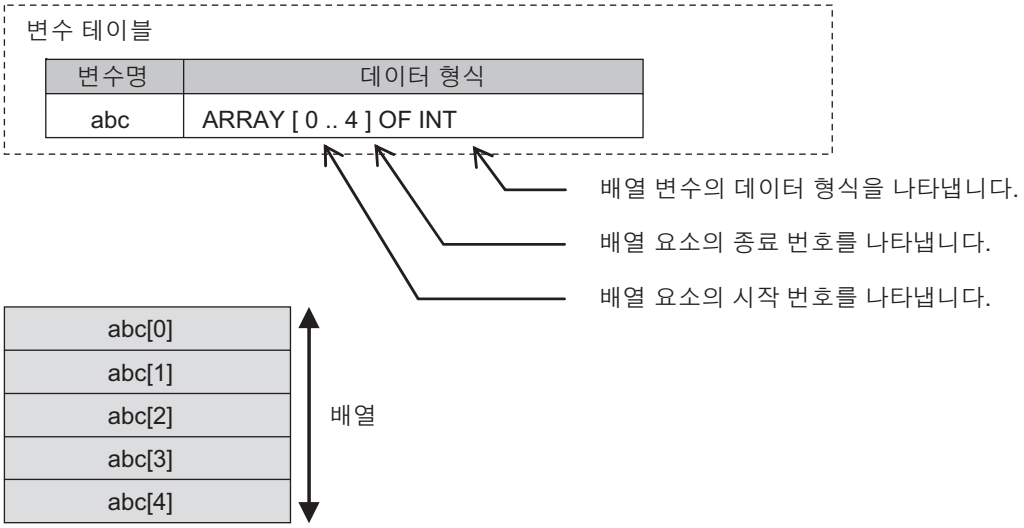
전체 요소(배열 전체) 지정

하나의 배열 지정을 구성하는 전체 요소를 '배열 전체'라고 합니다.
배열 전체는 배열 지정 변수명으로만 써서 나타냅니다.
위의 예에서는 'Mem', 'Data'라고 써서 나타냅니다.

- **배열 지정 작성 방법**

1 **변수 테이블의 데이터 형식 란에 "A"라고 입력하고, 후보가 되는 데이터 형식명에서 'ARRAY [?..?] OF ?'를 선택합니다.**

2 **[?..?] 부분의 좌측 ?에 요소의 시작 번호, 우측 ?에 종료 번호, OF ?의 ? 부분에 데이터 형식을 입력하여 변수를 등록합니다.**



참고

첨자에 시작 번호와 종료 번호를 지정하면 고정 길이 배열 지정 변수가 됩니다. 또한, 첨자에 반각 별표(*)를 지정하면 가변 길이 배열 지정 변수가 됩니다.
가변 길이 배열 지정으로 하는 경우에 고정 길이 배열 지정에 대해 제약 사항이 있습니다. 가변 길이 배열 지정의 자세한 내용은 '가변 길이 배열 지정(6-55페이지)'을 참조하십시오.

버전 관련 정보

가변 길이 배열 지정은 CPU 유니트의 유니트 버전 Ver.1.18 이상과 Sysmac Studio Ver.1.22 이상의 조합에서 사용할 수 있습니다.

● **배열 지정 사양**

항목	사양
배열 지정 변수의 최대 요소 수	65,535
요소 번호	0~65535 요소의 시작 번호가 0 이외이어도 상관 없습니다.
첨자	상수: 0~65535의 정수값 변수: 연산식: ST 언어인 경우만 연산식 지정 가능 예) y := x[a+b];

분류	데이터 형식		사용
기본 데이터 형식	정수 형식	SINT, INT, DINT, USINT, UINT , UDINT	가능
		LINT, ULINT	불가능
	부울 형식, 비트 열 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		불가능
파생 데이터 형식	구조체 형식, 공용체 형식, 열거 형식		불가능
POU 인스턴스 형식			불가능

1배열 변수의 최대 크기	NX701-□□□□: 8MB NX102-□□□□: 4MB NX1P2-□□□□: 1MB NJ501-□□□□: 4MB NJ301-□□□□: 2MB NJ101-□□□□: 2MB

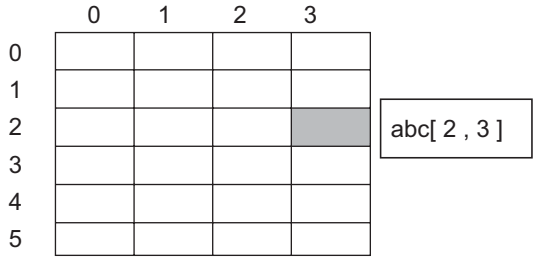
● **배열 지정의 차원**

1차원의 배열 지정은 요소가 1열로 나열된 1차원 데이터라고 생각할 수 있습니다.
동일하게 2차원 및 3차원의 배열 지정을 설정할 수 있습니다.
그 경우에 배열 지정 요소는 배열 지정 변수명에 차원과 동일한 수의 첨자를 붙여서 표현합니다. 차원 수의 최대값은 3입니다.

2차원 배열 지정

변수 테이블

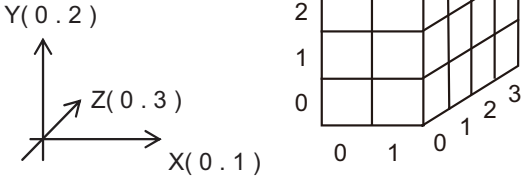
변수명	데이터 형식
abc	ARRAY [0..5 , 0..3] OF INT



3차원 배열 지정

변수 테이블

변수명	데이터 형식
ITEM	ARRAY [0..1, 0..2, 0..3] OF INT



● **배열 지정과 구조체 형식**

구조체 형식을 요소로 하는 배열 지정을 설정할 수도 있습니다.
또한, 배열 지정을 멤버로 하는 구조체 형식을 설정할 수도 있습니다.

구조체 형식을 요소로 하는 배열 지정

데이터 형식 테이블

데이터 형식	멤버	멤버
Str	x y	INT DINT

변수 테이블

변수명	데이터 형식
abc	ARRAY [1..4] OF Str



변수

abc[1].x
abc[1].y
:
abc[4].x
abc[4].y

배열 지정을 멤버로 하는 구조체 형식

데이터 형식 테이블

데이터 형식	멤버	데이터 형식
Str	x	ARRAY [0..1] OF INT
	y	DINT

변수 테이블

변수명	데이터 형식
abc	ARRAY [1..4] OF Str



변수

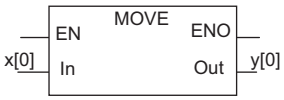
abc[1].x[0]
abc[1].x[1]
abc[1].y
:
abc[4].x[0]
abc[4].x[1]
abc[4].y

● **배열 지정을 파라메터로 하는 명령**

명령 중에는 배열 전체를 파라메터로 전달할 수 있는 것이 있습니다. 그 경우에 입력 파라메터로 배열 지정 변수명만 지정합니다.

예) MOVE 명령에 배열 지정 요소 하나만 전달하는 경우와 배열 전체를 전달하는 경우

■배열 1 요소를 지정

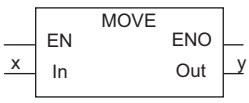


INT 형식 변수 x[0]을 y[0]에 전송

변수 테이블

변수명	데이터 형식
x	ARRAY [0 .. 199]OF INT
y	ARRAY [0 .. 199]OF INT

■배열 전체를 지정

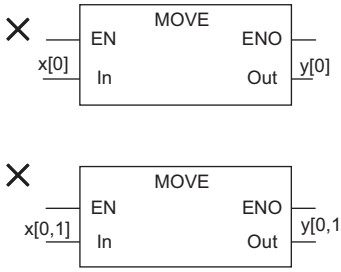


INT 형식 변수 x[0..199]를 y[0..199]에 전송

・제한 사항:
배열 전체를 지정하는 경우에는 요소 번호의 범위를 포함하여 동일한 데이터 형식이어야 합니다.

참고

다차원 배열의 부분적인 배열 전체는 파라메터로 지정할 수 없습니다



변수 테이블

변수명	데이터 형식
x	ARRAY [0..9 , 0..9 , 0..9]OF INT
y	ARRAY [0..9 , 0..9 , 0..9]OF INT

- **배열 지정의 보호 기능**

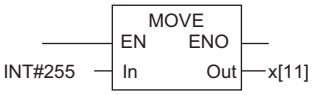
  배열의 요소 수를 초과해서 액세스하려고 한 경우에 다음과 같은 에러가 발생합니다.

첨자가 상수인 경우

Sysmac Studio에서 입력 시 및 프로그램 확인 시 에러 표시됩니다.

변수 테이블

변수명	데이터 형식
x	ARRAY [1..10] OF INT

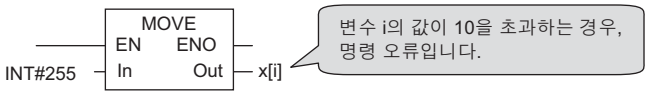


첨자가 변수인 경우

CPU 유니트가 명령 실행 후에 출력 파라메터를 출력 변수에 대입할 때 요소 수 초과 확인을 실행합니다. 지정된 첨자 변수가 배열 변수의 요소 수 를 초과하는 경우에는 명령 에러로 취급됩니다. 이 경우에 명령의 내부 처리는 정상 종료된 것이며, ENO의 값은 TRUE가 되므로 주의하십시오.

변수 테이블

변수명	데이터 형식
x	ARRAY [1..10] OF INT



가변 길이 배열 지정

배열 지정 변수를 선언할 때, 첨자에 반각 별표(*)를 지정하면 가변 길이 배열 지정 변수가 됩니다. 최대 요소 수, 차원 및 선언 가능한 데이터 형식 등은 고정 길이 배열의 사양과 동일합니다.
단, 변수 선언에서의 제약 사항 및 연산에서의 제약 사항이 있습니다.

- **변수 선언에서의 제약 사항**

  가변 길이 배열 지정 변수를 선언할 수 있는 것은 펑션(FUN)의 입출력 변수 및 펑션 블록(FB)의 입출력 변수뿐입니다. 그 밖의 변수에서는 선언할 수 없습니다.

선언하는 배열 변수	가능/불가능
FUN의 입출력 변수	가능
FB의 입출력 변수	가능
FUN의 입력 변수/출력 변수	불가능
FB의 입력 변수/출력 변수	불가능
내부 변수, 외부 변수, 글로벌 변수	불가능
구조체 형식 변수의 멤버/공용체 형식 변수의 멤버	불가능

2차원 및 3차원으로 가변 길이 배열 지정 변수를 선언하는 경우에 모든 차원에 반각 별표(*)를 지정해야 합니다. 가변 길이의 차원과 고정 길이의 차원이 혼재한 변수는 선언할 수 없습니다.

선언하는 배열 변수	지정 예	가능/불가능
모든 차원이 가변 길이 배열	• ARRAY[*] • ARRAY[*, *] • ARRAY[*, *, *]	가능
가변 길이의 차원과 고정 길이의 차원이 혼재	• ARRAY[*, 3..5] • ARRAY[*, *, 3..5]	불가능
시작 번호 또는 종료 번호에 대한 개별 지정	• ARRAY[1..*] • ARRAY[*..*]	불가능

- **연산에서의 제약 사항**

펑션(FUN) 및 펑션 블록(FB)의 입출력 변수가 가변 길이 배열인 경우에 지정할 수 있는 입출력 파라메터는 데이터 형식 및 차원 수가 동일한 배열 변수입니다. 데이터 형식과 차원 수가 동일하면 가변 길이 배열, 고정 길이 배열 모두 상관 없습니다. 데이터 형식 또는 차원 수의 다른 배열 변수를 지정할 수 없습니다.

펑션(FUN) 및 펑션 블록(FB)의 입출력 변수가 고정 길이 배열인 경우에 입출력 파라메터에 가변 길이 배열 변수를 지정할 수 없습니다.

FUN/FB의 입출력 변수의 예	지정하는 입출력 파라메터의 예	가능/불가능
데이터 형식이 INT, 차원 수가 1인 가변 길이 배열	데이터 형식이 INT, 차원 수가 1인 가변 길이 배열	가능
데이터 형식이 INT, 차원 수가 1인 가변 길이 배열	데이터 형식이 INT, 차원 수가 1인 고정 길이 배열	가능
데이터 형식이 INT, 차원 수가 1인 가변 길이 배열	데이터 형식이 UINT, 차원 수가 1인 가변 길이 배열	불가능
데이터 형식이 INT, 차원 수가 1인 가변 길이 배열	데이터 형식이 INT, 차원 수가 2인 가변 길이 배열	불가능
데이터 형식이 INT, 차원 수가 1인 고정 길이 배열	가변 길이 배열	불가능

ST 언어로 배열 전체의 대입식을 기술하는 경우에 좌변 또는 우변 중 어느 한쪽이라도 가변 길이 배열 변수가 포함되면 빌드 시에 에러가 됩니다.

동일하게 래더도 언어의 MOVE(전송) 명령에서 배열 전체를 전송하는 경우에 출력 파라메터 또는 입력 파라메터 중 어느 한쪽이라도 가변 길이 배열 변수가 포함되면 빌드 시 에러가 됩니다.

좌변/출력 파라메터	우변/입력 파라메터	ST 언어에서의 기술 예*1	가능/불가능
가변 길이 배열 변수	가변 길이 배열 변수	varlen_ary := varlen_ary;	불가능
가변 길이 배열 변수	고정 길이 배열 변수	varlen_ary := fixed_ary;	불가능
고정 길이 배열 변수	가변 길이 배열 변수	fixed_ary := varlen_ary;	불가능
고정 길이 배열 변수	고정 길이 배열 변수	fixed_ary := fixed_ary;	가능

*1. 'varlen_ary'는 가변 길이 배열 변수, 'fixed_ary'는 고정 길이 배열 변수를 나타냅니다.

- **가변 길이 배열 변수의 요소 시작 번호/종료 번호**

사용자 프로그램을 실행하면 요소의 시작 번호와 종료 번호가 정해집니다. LOWER_BOUND 명령 및 UPPER_BOUND 명령을 사용하면 가변 길이 배열 변수의 차원별 시작 번호 및 종료 번호를 취득할 수 있습니다.

LOWER_BOUND 명령 및 UPPER_BOUND 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

범위 지정((..))

범위 지정은 다음과 같은 정수 형식의 변수에 대해 그 값의 범위를 한정하는 것을 명시하는 지정입니다.

분류	데이터 형식
정수 형식	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT

다음과 같은 경우 입력한 값이 허용 범위 내인지 여부를 판단할 수 있습니다.

- 변수의 초기값을 지정할 때
- CIP 메시지 통신으로 값을 썼을 때

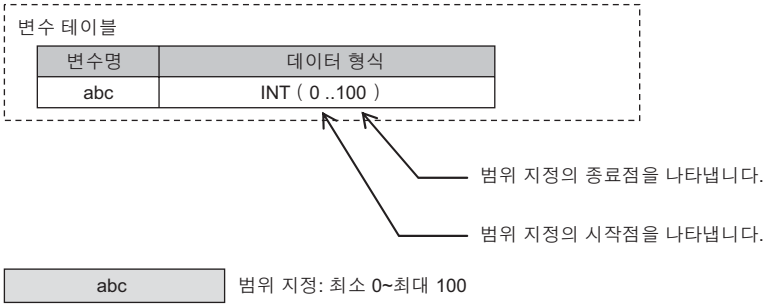
● 범위 지정 작성 방법

변수 테이블의 데이터 형식의 란에서 데이터 형식명 다음에 (시작점..종료점)을 입력합니다.

시작점: 변수에 저장 가능한 최소값

종료점: 변수에 저장 가능한 최대값

예)



abc 범위 지정: 최소 0~최대 100

● 범위 지정의 사양

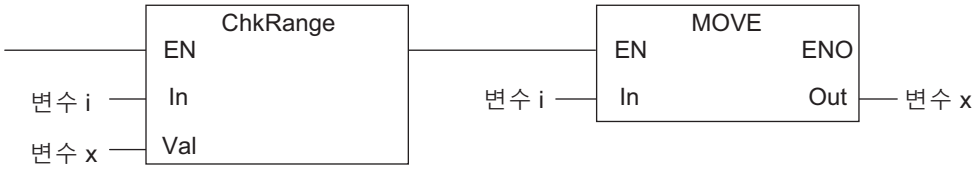
항목	사양			
지정할 수 있는 데이터 형식	정수 형식			
범위 외의 값을 쓰려고 했을 때의 동작	**경우**		**동작**	
	사용자 프로그램		에러가 되지 않고, 값은 재기입됩니다. 명령 실행에 의해 변수 값이 변화할 때, CPU 유니트는 범위를 확인하지 않습니다.	
	통신	**Sysmac Studio에서의 쓰기, CIP 메시지**	정수 형식인 경우	명령 에러가 됩니다.
			정수 형식 배열 변수의 요소인 경우	
			정수 형식의 구조체 형식의 멤버인 경우	
			정수 형식의 구조체 전체인 경우	명령 에러가 되지 않고, 값은 재기입됩니다.
			정수 형식 배열 전체인 경우	
		태그 데이터 링크(내장 EtherNet/IP, EtherNet/IP 유니트 모두)	범위를 지정한 멤버에 대한 단독 쓰기는 에러가 됩니다. 범위를 지정한 멤버를 포함한 구조체 전체에 대한 쓰기는 에러가 되지 않습니다.	
	슬레이브/유니트에서의 IN 리프레시		에러가 되지 않고, 값은 재기입됩니다.	
	에러가 되지 않고 값은 재기입됩니다.		에러가 되지 않고, 값은 재기입됩니다.	

사용상의 주의 사항

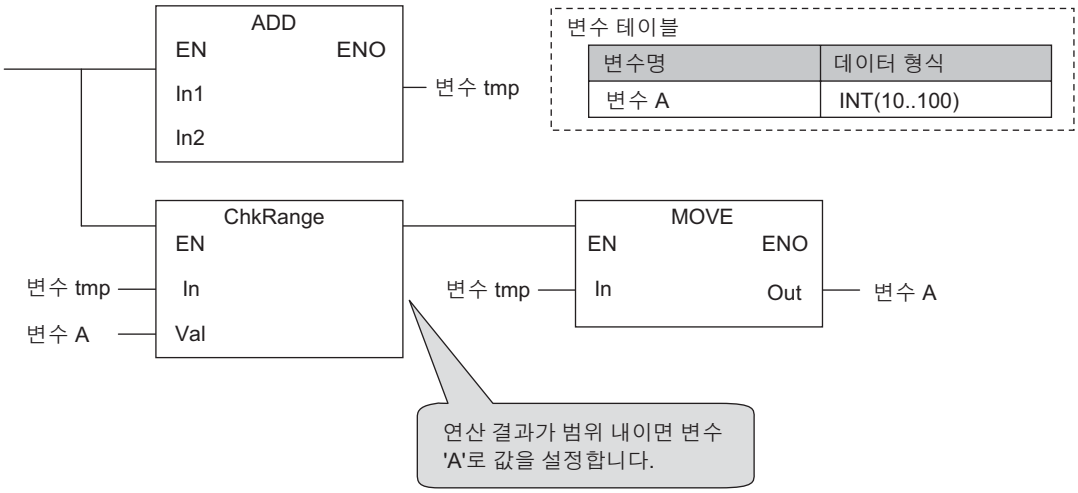
변수의 범위 지정은 사용자 프로그램에 의한 명령의 실행에 의해 변수의 값이 변화한 경우까지 확인하지는 않습니다. 사용자 프로그램 실행에 의해 변화한 변수 값의 범위를 확인하고자 하는 경우에 다음과 같이 범위를 확인하는 명령을 사용하십시오.

변수 테이블

변수명	데이터 형식
변수 x	INT(10..100)



또한, 사칙 연산 결과 등을 설정하는 경우에는 사전 확인을 할 수 없으므로 다음과 같이 사칙 연산(예: ADD) 후의 값의 범위를 확인하십시오.



범위 지정을 사용하는 경우에 초기값에 주의하십시오.
Sysmac Studio의 초기값의 열이 공백인 경우에 초기값은 0이 됩니다. 0을 포함하지 않은 범위를 유효 범위로 한 경우에도 동일하게 0이 됩니다.

6-3-8 변수의 각 속성의 설명

데이터 형식 이외의 변수의 각 속성을 아래에서 설명합니다.

변수명

변수명은 해당 변수를 식별하기 위한 이름입니다.
하나의 POU 내에 동일한 변수명의 변수를 복수 선언할 수 없습니다. 단, 다른 POU에서 동일한 변수명의 로컬 변수를 선언하는 것은 가능하고, 이들은 다른 변수로 취급됩니다.

변수명의 제약은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.

AT(할당대상) 지정

이 속성은 변수의 할당 메모리 주소를 CJ 유니트용 메모리에 확인하기 위해 사용합니다.
주로 다음과 같은 고기능 유니트의 할당 임의 영역을 지정할 때 지정합니다.

- DeviceNet 유니트 사용 시의 고정 할당 영역
- DeviceNet 유니트 사용 시 및 CompoNet 마스터 유니트(통신 모드 번호 8)의 CX-Integrator에서 지정한 자유 할당 영역
- 고속 카운터 유니트 사용 시의 확장 메모리
- 프로세스 입출력 유니트 사용 시의 확장 메모리
- EtherNet/IP 유니트 사용 시의 대상 노드 정보 및 사용자 정의 Status 영역용 데이터 영역

이 속성을 설정하지 않은 경우에 변수는 메모리의 임의의 주소(변수 메모리)에 자동으로 할당됩니다.

사용상의 주의 사항

CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

참고

디바이스 변수인 경우에 I/O 포트에 대해 할당하여 자동으로 내부적으로 AT(할당대상) 지정됩니다.

- **할당대상 영역**

  할당대상으로 지정할 수 있는 영역은 다음과 같습니다.

영역	표현
CIO	0~6143
WR	W0~511
HR	H0~1535
DM	D0~32767*1
EM*2	NJ501-□□□□: E0_0~E18_32767 NJ301-□□□□: E0_0~E3_32767 NJ101-□□□□: E0_0~E3_32767

*1. NX1P2-□□□□에서는 D0~15999가 됩니다.
*2. NX1P2-□□□□에서는 EM을 지원하지 않습니다.

변수의 데이터 형식에 따른 데이터 할당은 다음과 같습니다.

변수의 데이터 형식	할당 위치
BOOL	비트 단위로 지정 가능
BYTE/SINT/USINT	데이터 할당 선두 위치를 지정한 CJ 주소의 bit0 또는 bit8 중 어디에 배치할지 지정 가능 예 1) D100의 bit0에 AT 지정한 경우(%D100) D100: 16#**12....1바이트의 데이터(12)가 bit0에서 저장됩니다. 예 2) D100의 bit8에 AT 지정한 경우(%D100.8) D100: 16#12**....1바이트의 데이터(12)가 bit8에서 저장됩니다.
WORD/INT/UINT	지정한 CJ 주소의 비트 0에서 데이터 크기 분량을 순차 저장
DWORD/DINT/UDINT **REAL**	
LWORD/LINT/ULINT **LREAL**	
STRING	데이터 할당 선두 위치를 지정한 CJ 주소의 비트 0 또는 비트 8 중 어디에 배치할지 지정 가능
TIME **DATE** **TIME_OF_DAY** **DATE_AND_TIME**	지정한 CJ 주소의 비트 0에서 데이터 크기 분량을 순차 저장

- **AT(할당대상) 지정할 수 있는 변수의 종류**

  변수 단위로 지정합니다. 배열 변수, 구조체 변수, 공용체 변수는 전체에 대해 지정합니다.

- **AT(할당대상) 지정한 변수의 속성**

	지정	비고
이름	가능	
데이터 형식	가능	
유지	가능	AT(할당대상) 지정대상 CJ 유니트용 메모리의 속성에 따르지 않는 경우에 에러가 됩니다.
초기값	가능	메모리의 값을 그대로 이용하는 경우에는 '초기값 지정 없음(None)'을 설정합니다.
상수	가능	명령에 의한 값을 쓸 수 없습니다.
네트워크 공개	가능	
엣지	불가능	(펑션 블록의 입력 변수에 대해서만 가능)

- **AT(할당대상) 지정의 입력 및 표시**

  AT(할당대상) 지정하는 경우에 Sysmac Studio의 변수 테이블의 '할당대상' 열에 다음 항목을 입력합니다. 변수 테이블 및 I/O 맵의 '할당대상' 열에 다음 항목이 표시됩니다.

변수 종류	'할당대상' 열의 입력 및 표시	예
워드 주소에 AT(할당대상) 지정한 사용자 정의 변수	%[채널 주소]	%D100
비트 주소에 AT(할당대상) 지정한 사용자 정의 변수	%[채널 주소].[비트 위치]	%W0.00

또한, 다음과 같은 변수도 내부적으로는 AT(할당대상) 지정되어 있고, 동일하게 '할당대상' 열에 다음과 같이 표시됩니다.

변수 종류	'할당대상' 열에서의 표시	예
CJ 유니트 디바이스 변수	IOBus://rack#[랙 번호]/slot#[슬롯 번호]/[I/O 포트명]	기본 I/O 유니트인 경우: IOBus://rack#0/slot#1/Ch1_In/Ch1_In00 고기능 유니트인 경우: IOBus://rack#0/slot#1/PeakHoldCmd/ch1_PeakHoldCmd
EtherCAT 슬레이브 디바이스 변수	EtherCAT 슬레이브 터미널 상의 NX 유니트인 경우: ECAT://node#[노드 주소.NX 유니트 번호]/[I/O 포트명]	ECAT://node#[10.15]/Input1
	상기 이외의 경우: ECAT://node#[노드 주소]/[I/O 포트명]	ECAT://node#1/Input1
NX 유니트 디바이스 변수*1	IOBus://unit#[NX 유니트 번호]/[I/O 포트명]	IOBus://unit#8/Input Bit 00
내장 I/O 디바이스 변수*2	BuiltInIO://cpu/#0/[I/O 포트명]	BuiltInIO://cpu/#0/Input Bit 00
옵션 보드 디바이스 변수*2	BuiltInIO://opt/#[물리 포트 번호*3]/[I/O 포트명]	BuiltInIO://cpu/#1/Ch1 Analog Input Value
축 변수	MC://_MC_AX[ ] MC://_MC1_AX[ ]*4 MC://_MC2_AX[ ]*4	MC://_MC_AX[1]
축 그룹 변수	MC://_MC_GRP[ ] MC://_MC1_GRP[ ]*4 MC://_MC2_GRP[ ]*4	MC://_MC_GRP[1]

*1. NX102 및 NX1P2 CPU 유니트에서만 사용 가능한 시스템 정의 변수입니다.
*2. NX1P2 CPU 유니트에서만 사용 가능한 시스템 정의 변수입니다.
*3. '물리 포트' 번호는 I/O 포트의 물리적인 위치를 나타냅니다. 옵션 보드 슬롯 1인 경우에 '0', 옵션 보드 슬롯 2인 경우 '1'이 됩니다.
*4. NX701 CPU 유니트에서만 사용 가능한 시스템 정의 변수입니다.

사용상의 주의 사항

동일한 주소에 복수의 변수를 할당할 수도 있지만 프로그램의 가독성 및 디버그의 어려움 때문에 권장할 수 없습니다. 이 경우에 초기값을 설정하는 변수는 하나만으로 하십시오. 각각의 변수에 다른 초기값 설정하면 초기값이 잘못됩니다.

유지

이 속성은 다음과 같은 경우 변수 값을 유지할지 여부를 지정하기 위해 사용합니다.

- 전원 OFF 후의 전원 투입 시
- 동작 모드 전환 시
- 전체 정지 이상 레벨의 컨트롤러 이상 발생 시

이 속성을 설정하지 않은 경우에 변수의 값은 위와 같은 경우 초기값이 됩니다.

생산 수 등 CPU 유니트의 전원이 OFF된 후에도 유지하고자 하는 데이터를 저장하는 변수는 유지로 지정해두면 편리합니다.

AT(할당대상) 지정한 변수에는 할당대상 영역에 따라 유지/비유지를 맞춰야 합니다.
(유지 영역: HR, DM, EM
비유지 영역: CIO, WR)

- **유지를 유효로 하기 위해 필요한 조건**
  CPU 본체에는 배터리를 장착하십시오.

- **유지 변수에 초기값을 반영시키고자 하는 경우**
  사용자 프로그램 전송 시 '유지 변수의 현재값을 초기화'를 선택합니다

- **유지/비유지 동작의 차이**
  유지에 관계가 있는 사례와 그 때 변수에 설정되는 값의 관계는 다음과 같습니다.

사례		변수의 값	
		유지	비유지
전원 OFF 후의 전원 투입 시		유지됨	유지되지 않음
동작 모드 전환 시			
전체 정지 이상 레벨의 컨트롤러 이상 발생 시			
사용자 프로그램 전송 시	'유지 변수 초기화 안 함' 설정		
	'유지 변수 초기화' 설정	유지되지 않음	

- **유지 지정할 수 있는 변수의 종류**
  변수 단위로 지정합니다. 배열 변수, 구조체 변수, 공용체 변수는 전체에 대해 지정합니다.

초기값

초기값은 다음과 같은 경우에 설정되는 변수의 값입니다.
- 전원 투입 시
- 운전 모드⇔프로그램 모드로 전환 시
- 사용자 프로그램 전송 시 '유지 변수 초기화' 설정인 경우
- 전체 정지 이상 레벨의 컨트롤러 이상 발생 시

초기값을 설정해두면 변수를 초기화하는 프로그램을 생략할 수 있습니다. 예를 들어 레시피 등의 데이터를 초기값으로 사전에 설정해둘 수도 있습니다.
또한, 초기값을 설정하지 않아도 상관 없습니다.

- **초기값을 설정할 수 있는 변수의 종류**
  변수의 종류에 따라 초기값을 설정할 수 있는 것과 할 수 없는 것이 있습니다. 목록은 다음과 같습니다.

변수의 종류	초기값 설정
글로벌 변수	가능
내부 변수	
입력 변수	
출력 변수	
FUN의 반환값	불가능
입출력 변수	
외부 변수	

● **초기값 있음/없음 선택**

변수 작성 시 초기값 있음/없음을 선택할 수 있습니다.

초기값 지정함

초기값

(공백) 또는

초기값
3.14

초기값 지정 안 함

초기값
None

종류별 선택 가능 여부는 다음과 같습니다.

종류		예	초기값 있음/없음 선택
기본 데이터 형식 변수		**aaa**	가능
배열 지정 변수	**배열 전체**	**bbb**	가능
	요소	**bbb[2]**	불가능
구조체 형식 변수	**구조체 전체**	**ddd**	가능
	멤버	**ddd.xxx**	불가능
공용체 형식 변수	**공용체 전체**	**eee**	불가능(초기값은 제로 고정)
	멤버	**eee.word**	초기값은 반드시 제로
열거 형식 변수		**ccc**	가능
POU 인스턴스 형식		**instance**	불가능

참고

기본 I/O 유니트의 일부는 하나의 I/O 포트에 대해 비트 열 형식과 부울 형식 등 복수의 액세스 수단을 갖는 것이 있습니다. 이 I/O 포트에 대해 초기값을 이용하는 경우에는 한쪽을 'None(초기값 지정 없음)'으로 하십시오.

● **초기값이 설정되는 타이밍**

초기값을 설정한 변수에 실제로 초기값이 대입되는 타이밍은 다음과 같습니다.

- 전원 투입 시
- 동작 모드가 프로그램에서 운전, 운전에서 프로그램으로 전환했을 때
- '변수 초기화'를 지정하여 사용자 프로그램을 전송했을 때
- 전체 정지 이상 레벨의 컨트롤러 이상 발생 시

● **초기값 지정이 공백인 경우**

초기값 지정이 공백인 경우에는 변수의 초기값은 다음과 같습니다.

데이터 형식		초기값
부울 형식/비트 열 형식	**BOOL, BYTE, WORD, DWORD, LWORD**	0
정수 형식	**SINT, INT, DINT, LINT USINT, UINT, UDINT, ULINT**	0
실수 형식	**REAL, LREAL**	0.0
지속 시간 형식·날짜 형식·시각 형식·날짜 시각 형식	**TIME**	T#0S
	DATE	D#1970-01-01
	TIME_OF_DAY	TOD#00:00:00
	DATE_AND_TIME	DT#1970-01-01-00:00:00
문자열 형식	**STRING**	'' (공백 문자)

● **배열 지정 변수의 초기값**

데이터 형식	초기값의 사양
배열 지정	• 요소별로 초기값을 설정할 수 있습니다. • 초기값을 지정하는 경우에는 모든 요소에 대해 값을 설정하거나 공백으로 해야 합니다.

● **파생 데이터 형식의 초기값**

데이터 형식 자체에 초기값을 지정하는 것이 아니라 변수별로 초기값을 지정합니다.

데이터 형식	초기값의 사양
구조체 형식	• 멤버별로 초기값을 설정할 수 있습니다. • 초기값을 지정하는 경우에는 모든 요소에 대해 값을 설정하거나 공백으로 해야 합니다.
공용체 형식	• 지정할 수 없습니다. 항상 제로가 됩니다.
열거 형식	• 지정할 수 있습니다.

● **초기값이 적용되지 않는 변수**

다음과 같은 변수인 경우에 전원 투입 시 초기값이 적용되지 않고, 전원 OFF 직전의 값이 유지됩니다.

- 유지 속성의 변수
- AT(할당) 지정 변수(유지 영역, DM, HR/EM 영역 지정 시에만)

사용상의 주의 사항

CPU 본체에 배터리가 장착되지 않은 경우에 위의 변수 값도 초기값이 됩니다.

상수

상수는 명령, ST 언어의 연산자 또는 CIP 메시지 통신에 의해 값을 쓰는 것을 금지하는 속성입니다. 이것으로 프로그램의 변수에 대한 값의 쓰기를 방지할 수 있습니다.
이 속성의 변수인 경우에 초기값이 설정된 이후는 명령에 의한 쓰기를 금지할 수 있습니다.
만일 POU 내에 상수 속성이 있는 변수에 값을 쓰는 명령이 존재하면 빌드 시점에서 에러가 됩니다.

● **동작**

POU 내에 상수 속성이 있는 변수에 값을 쓰는 명령 또는 연산자가 존재할 때의 동작은 다음과 같습니다.

쓰기원		값을 쓰려고 했을 때의 동작
사용자 프로그램		프로그램 확인 시 에러가 됩니다. Sysmac Studio가 빌드 시 확인을 실행하여 빌드 에러가 됩니다.
통신	**Sysmac Studio에서의 쓰기**	불가능
	CIP 메시지	명령 에러가 됩니다.
	태그 데이터 링크	태그 데이터 링크 시작 시 에러가 발생하지만 태그 데이터 링크는 계속됩니다. 단, 상수 속성을 갖는 변수의 값은 재기입되지 않습니다.
슬레이브/유니트에서의 IN 리프레시		에러가 되지 않고, 값은 재기입됩니다.
강제값 리프레시		

- **상수를 지정할 수 있는 단위**
  변수 단위로 지정합니다. 배열 변수, 구조체 변수, 공용체 변수는 전체에 대해 지정합니다.



참고

'상수' 속성은 사용자 프로그램에서의 쓰기도 금지합니다.

네트워크 공개

네트워크 공개 속성은 CIP 메시지 통신 또는 태그 데이터 링크 기능에 따라 해당 변수를 외부(다른 컨트롤러 및 상위 PC 등)에서도 읽고 쓸 수 있도록 하는(이를 '네트워크 공개'라고 함) 속성입니다.
이 속성을 설정하지 않은 경우에 변수는 해당 변수를 선언한 컨트롤러에서 읽고 쓸 수 있고, 외부(다른 컨트롤러 및 상위 PC 등)에서는 읽고 쓸 수 없습니다.

네트워크 공개된 변수를 '네트워크 변수'라고 합니다.

- **네트워크 공개의 사양**
  네트워크 공개에는 '공개만 해당', '입력', '출력'의 지정이 있습니다. 각각의 사양은 다음과 같습니다.

네트워크 공개		사양
공개 안 함		외부에서는 액세스할 수 없음 지원 소프트웨어에서는 공개 유무에 관계없이 액세스할 수 있음
공개	공개만 해당	CIP 통신에 의한 외부에서의 액세스가 가능한 변수 태그 데이터 링크는 불가능
	입력	CIP 통신 또는 태그 데이터 링크에 의한 외부에서의 액세스가 가능한 변수 태그 데이터 링크 시에는 데이터의 입력용(다른 CPU 유니트→로컬 CPU 유니트) 변수가 됨
	출력	CIP 통신 또는 태그 데이터 링크에 의한 외부에서의 액세스가 가능한 변수 태그 데이터 링크 시에는 데이터의 출력용(로컬 CPU 유니트→다른 CPU 유니트) 변수가 됨

- **네트워크 공개 지정할 수 있는 단위**
  변수 단위로 지정합니다. 배열 변수, 구조체 변수, 공용체 변수는 전체에 대해 지정합니다.

엣지

엣지 속성은 BOOL 형식 변수의 값이 FALSE→TRUE로 상승했을 때 또는 TRUE→FALSE로 하강했을 때, FB에 TRUE 값을 전달하는 속성입니다.
FB의 BOOL 형식의 입력 변수에만 설정할 수 있습니다.

- **용도**

  입력 파라메터의 상승/하강만 FB측에서 수신하고자 하는 경우에 사용합니다.
  예를 들어 입력 파라메터에 변화가 있었을 때만, FB를 기동할 때 등에 사용합니다.

- **동작**
  - '상승'을 지정하면 그 입력 변수에 연결한 입력 파라메터의 값이 FALSE에서 TRUE로 상승했을 때만 입력 변수의 값이 TRUE가 됩니다.
  - '하강'을 지정하면 입력 파라메터가 TRUE에서 FALSE로 하강했을 때만 입력 변수의 값이 TRUE가 됩니다.

지정	입력 파라메터의 값	변수의 값
상승	FALSE→TRUE	TRUE
	그 이외	FALSE
하강	TRUE→FALSE	TRUE
	그 이외	FALSE
없음	−	입력 파라메터의 값에 맞춰 매번 변화합니다

6-3-9 상태 변화에 따른 변수 값의 변화

여기에서는 CPU 유니트의 상태 변화에 따른 변수 값의 변화에 대해 설명합니다.

버전 관련 정보

CPU 유니트 Ver.1.13 이상과 Sysmac Studio Ver.1.17 이상의 조합에서는 시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'의 설정을 통해 동작 모드 변경 시 및 다운로드 시의 동작을 설정할 수 있습니다.

전원 투입, 동작 모드 변경, 다운로드 시에 CPU 유니트 내부의 변수의 값은 다음 표와 같이 변화합니다.

변수의 유지 속성	변수 종류		전원 투입 시	동작 모드 변경 시	다운로드 후	
					'유지 변수 초기화' 설정 시	'유지 변수 초기화 안 함' 설정 시
비유지	사용자 정의 변수		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1			
		CJ 유니트용 메모리의 CIO/WR 영역에 AT(할당대상) 지정	• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 값을 유지함	
	EtherCAT 슬레이브 디바이스 변수*2		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1			
	NX 유니트 디바이스 변수*2		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1			
	내장 I/O 디바이스 변수*2		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1			
	옵션 보드 디바이스 변수*2		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1			
	CJ 유니트 디바이스 변수*2		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 16#00*1		• 초기값 설정 있음: 지정한 초기값이 됨 • 초기값 설정 없음(None): 값을 유지함	
	CJ 유니트용 메모리 CIO/WR 영역*2*3		16#00		값을 유지	

<table>
<tr><th rowspan="2">변수의 유지 속성</th><th colspan="2" rowspan="2">변수 종류</th><th rowspan="2">전원 투입 시</th><th rowspan="2">동작 모드 변경 시</th><th colspan="2">다운로드 후</th></tr>
<tr><th>'유지 변수 초기화' 설정 시</th><th>'유지 변수 초기화 안 함' 설정 시</th></tr>
<tr><td rowspan="5">유지</td><td rowspan="3">사용자 정의 변수</td><td>유지 조건을 충족*4</td><td rowspan="5">변화 없음
(전원 OFF 전의 값을 유지)</td><td rowspan="5">변화 없음
(동작 모드 변경된 값을 유지)</td><td rowspan="2">• 초기값 설정 있음: 지정한 초기값이 됨
• 초기값 설정 없음 (None): 16#00*1</td><td>다운로드 전의 값을 유지</td></tr>
<tr><td>유지 조건이 충족되지 않음*3</td><td>• 초기값 설정 있음: 지정한 초기값이 됨
• 초기값 설정 없음 (None): 16#00*1</td></tr>
<tr><td>CJ 유니트용 메모리의 HR/DM/EM 영역에 AT(할당대상) 지정</td><td>• 초기값 설정 있음: 지정한 초기값이 됨
• 초기값 설정 없음 (None): AT(할당대상) 지정 메모리 값에 따른 값이 됨</td><td></td></tr>
<tr><td colspan="2">CJ 유니트용 디바이스 변수</td><td>• 초기값 설정 있음: 지정한 초기값이 됨
• 초기값 설정 없음 (None): AT(할당대상) 지정 메모리 값에 따른 값이 됨</td><td>AT(할당대상) 지정 메모리 값에 따른 값이 됨</td></tr>
<tr><td colspan="2">CJ 유니트용 메모리 HR/DM/EM 영역*2</td><td>값을 유지
• '설정용 데이터 '(DM 영역)에 CJ 유니트 디바이스 변수가 할당된 경우에는 변수의 초기값 설정에 따름
• 사용자 정의 변수가 할당된 경우에는 변수의 초기값 설정에 따름</td><td>값을 유지</td></tr>
</table>

*1. 데이터 형식에 따라 16#00 이외의 값이 되는 경우가 있습니다. 자세한 내용은 📖'초기값 지정이 공백인 경우(6-63페이지)'를 참조하십시오.
*2. 시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'로 장치 출력 유지 설정을 유효(16#A5A5)로 설정한 경우에 동작 모드 변경 시 및 다운로드 시에도 장치 출력을 유지합니다.
장치 출력 유지 설정에 대해서는 📖'장치 출력 유지 설정(6-70페이지)'을 참조하십시오.
*3. 사용자 정의 변수 및 CJ 유니트 디바이스 변수가 AT(할당대상) 지정된 경우에는 제외합니다.
*4. 유지 조건에 대해서는 📖'유지 조건(6-72페이지)'을 참조하십시오.

전체 정지 이상 레벨의 컨트롤러 이상 발생 시, 온라인 편집 시 CPU 유니트 내부의 변수 값은 다음 표와 같이 변화합니다.

변수의유지 속성	변수 종류	전체 정지 이상 레벨의 컨트롤러 이상 발생 시	온라인 편집 시	
			온라인 편집 대상 내의 POU에 추가한 변수	온라인 편집 대상 내의 POU에 이미 존재했던 변수
비유지	사용자 정의 변수/ 디바이스 변수	• 초기값 설정 있을 때: 지정한 초기값이 됨 • 초기값 설정 없을 때: 16#00	• 초기값 설정 있을 때: 지정한 초기값이 됨 • 초기값 설정 없을 때: 16#00	변화 없음
	CJ 유니트용 메모리 CIO/WR 영역	16#00	변화 없음	
유지	사용자 정의 변수/ 디바이스 변수	변화 없음 (이상 발생 전의 값을 유지)	• 초기값 설정 있을 때: 지정한 초기값이 됨 • 초기값 설정 없을 때: 16#00	
	CJ 유니트용 메모리 HR/DM/EM 영역		변화 없음	
기타	강제값 리프레시 상태	해제	변화 없음	

사용상의 주의 사항

CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

장치 출력 유지 설정

시스템 정의 변수 '_DeviceOutHoldCfg(장치 출력 유지 설정)'의 설정을 통해 동작 모드 변경 시 및 다운로드 시의 동작을 설정할 수 있습니다.

장치 출력 유지 설정을 무효(16#A5A5 이외)로 한 경우에 장치 출력은 '6-3-9 상태 변화에 따른 변수 값의 변화(6-67페이지)'에서 설명한 동작이 됩니다.
장치 출력 유지 설정을 유효(16#A5A5)로 한 경우에 동작 모드 변경 시 및 다운로드 시에도 장치 출력을 유지합니다.

상태	장치 출력 유지 설정	
	무효(16#A5A5 이외)	유효(16#A5A5)
동작 모드 변경 시	변수를 초기화	변수의 값을 유지
다운로드 시	I/O 리프레시를 정지하고, 다운로드 후에 변수를 초기화	I/O 리프레시를 실행하고, 다운로드 후에 변수의 값을 유지

- **대상이 되는 장치 출력**

  장치 출력 유지 설정의 대상이 되는 장치 출력은 다음과 같습니다.
  - EtherCAT 슬레이브 디바이스 변수
  - NX 유니트 디바이스 변수
  - 내장 I/O 디바이스 변수
  - 옵션 보드 디바이스 변수
  - CJ 유니트 디바이스 변수(운전용 데이터)
  - CJ 유니트용 메모리의 CIO/WR 영역

- **다운로드 시 장치 출력을 유지할 수 있는 조건**

  다운로드 시 장치 출력을 유지하기 위한 조건으로 변경할 수 있는 항목에 제한이 있습니다.
  변경한 경우에도 장치 출력을 유지할 수 있는 항목은 다음과 같습니다.

항목	
EtherCAT	EtherCAT 슬레이브의 백업 파라메터 설정
CPU·증설 랙	CJ 유니트의 고기능 유니트 설정
I/O 맵	변수명
	변수 코멘트
	변수 유형
이벤트 설정	
태스크 설정	프로그램 할당 설정
프로그래밍-POU	프로그램
	섹션
	외부/내부 변수
	펑션
	펑션 블록
프로그래밍-데이터	데이터 형식
	글로벌 변수
EtherNet/IP 연결 설정	
DB 연결 명령	

상기 이외의 항목을 변경하고, 다운로드한 경우에는 장치 출력을 유지하지 않습니다.

또한 Sysmac Studio의 동기 창에서[다음을 전송 안 함(모두 동기의 대상 외), -CJ 고기능 유니트의 파라메터/EtherCAT 슬레이브의 백업 파라메터]를 선택하십시오. 선택을 취소하면 장치 출력을 유지하지 않습니다.

장치 출력을 유지하고, EtherCAT 슬레이브의 백업 파라메터 및 NX 버스의 유니트 동작 설정을 변경하고자 하는 경우에는 다음과 같이 실행하십시오.

1. **사용자 프로그램만 변경하고, 다운로드합니다.**
2. **EtherCAT 구성 편집 화면의 슬레이브를 선택합니다.**
3. **EtherCAT 구성의 우측에 표시되는 슬레이브의 파라메터 설정에서 [백업 파라메터 설정 편집] 버튼을 클릭합니다.**
4. **[백업 파라메터 설정 편집] 창의 [전송[PC→ 슬레이브]]를 클릭합니다.**
5. **CPU·증설 랙 구성 편집 화면에서 유니트를 마우스 오른쪽 버튼으로 클릭하고 [유니트 동작 설정 편집]을 선택합니다.**
6. **유니트 동작 설정의 [전송[PC→ 유니트]]를 클릭합니다.**

[전송[PC→ 슬레이브]] 또는 [전송[PC→ 유니트]]를 실행하면 전송 완료 후에 전송 대상 슬레이브 또는 유니트가 재시작됩니다. 이 경우에 전송 대상의 슬레이브 또는 유니트의 장치 출력은 유지되지 않습니다. 그러나 전송 대상 외의 슬레이브 및 유니트의 장치 출력은 유지됩니다.

- **관련 시스템 정의 변수**

  시스템 정의 변수에 의한 방법으로 백업을 실시하는 경우에 관련 시스템 정의 변수에는 다음과 같은 것이 있습니다. 시스템 정의 변수의 자세한 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'을 참조하십시오.

변수명	이름	기능	데이터 형식	R/W
_DeviceOutHoldCfg	장치 출력 유지 설정	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력을 유지하는 경우에 16#A5A5로 합니다. 16#A5A5 이외에서는 동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력은 초기화됩니다.	WORD	RW
_DeviceOutHoldStatus	장치 출력 유지 상태	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력이 유지된 경우에 TRUE가 됩니다. 장치 출력 유지 설정이 16#A5A5 이외일 때 또는 전체 정지 이상 레벨의 컨트롤러 이상 발생 시에 대상 장치 출력이 초기화된 경우에 FALSE가 됩니다.	BOOL	R

유지 조건

유지 조건이란, 다운로드 전과 후 다음 모든 조건을 충족하는 것을 말합니다.

- 변수명이 동일할 것
- 데이터 형식명이 동일할 것
- 유지 속성이 '유지'일 것

유지 속성이 '유지'인 변수에 대해 POU명, 변수명을 변경했을 때의 변수 값의 자세한 내용은 'POU명 변경 및 POU 새로 작성 등에 따른 유지 변수의 값(6-72페이지)', 데이터 형식을 변경했을 때의 변수 값의 자세한 내용은 '유지 변수의 데이터 형식을 변경했을 때의 변수 값(6-75페이지)'을 참조하십시오.

버전 관련 정보

CPU 유니트의 유니트 버전 Ver.1.10 이하인 경우에는 다운로드 전과 후 다음과 같은 모든 조건을 충족하는 것을 말합니다.

- 변수명이 동일할 것
- 데이터 형식명 및 데이터 형식의 크기가 동일할 것
- 유지 속성이 '유지'일 것

POU명 변경 및 POU 새로 작성 등에 따른 유지 변수의 값

Sysmac Studio에서 작성한 프로젝트를 컨트롤러에 다운로드하면 컨트롤러는 Sysmac Studio에서 표시되는 POU명과 다른 POU명으로 취급합니다. 따라서 Sysmac Studio에서 동일한 POU명이 표시되는 경우에도 컨트롤러는 다른 POU로 취급하는 경우가 있습니다.

Sysmac Studio에서 POU명의 변경 및 POU의 새로 작성 등의 조작을 실행한 후에 다운로드를 실행한 경우의 컨트롤러가 취급하는 POU명의 유지 속성이 '유지'인 로컬 변수의 값에 대해 설명합니다.

- **Sysmac Studio에서 조작하기 전의 상태**

  컨트롤러에는 Sysmac Studio에서 작성된 프로젝트 'Project_A'가 다운로드되어 있습니다. 'Project_A'에는 'POU_A'가 있고, 'POU_A'에는 로컬 변수 VarA가 있습니다.
  Symac Studio에서 'POU_A'라고 표시되는 POU가 컨트롤러 측에서는 'POU_XXX'로 취급되는 것으로 가정합니다. 또한, 컨트롤러측의 로컬 변수 VarA의 현재값은 '20'으로 변화한 것으로 가정합니다.

Sysmac Studio 측	
프로젝트명	: Project_A
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20

● **POU명을 변경한 경우**

Sysmac Studio에서 'POU_A'의 이름을 'POU_B'로 변경하고, 다운로드한 경우에도 컨트롤러 측의 POU명 및 로컬 변수 VarA의 현재값은 변경되지 않습니다.

Sysmac Studio 측	
프로젝트명	: Project_A
POU명	: POU_B
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20

● **POU를 새로 작성한 경우**

Sysmac Studio에서 로컬 변수 VarA가 있는 'POU_C'를 새로 작성하고, 다운로드한 경우에 컨트롤러 측은 새로운 'POU_YYY' 및 새로운 로컬 변수 VarA로 취급합니다.
따라서 새 로컬 변수 VarA의 값은 초기값인 '0'이 됩니다.

Sysmac Studio 측	
프로젝트명	: Project_A
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0
POU명	: POU_C
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20
POU명	: POU_YYY
변수명	: VarA
VarA의 현재값	: 0

● **POU를 복사하여 POU를 작성한 경우**

Sysmac Studio에서 'POU_A'를 복사하여 'POU_A_copy'를 작성하고, 다운로드한 경우에 컨트롤러 측은 새로운 'POU_ZZZ' 및 새로운 로컬 변수 VarA로 취급합니다.
따라서 새 로컬 변수 VarA의 값은 초기값인 '0'이 됩니다.

Sysmac Studio 측	
프로젝트명	: Project_A
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0
POU명	: POU_A_copy
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20
POU명	: POU_ZZZ
변수명	: VarA
VarA의 현재값	: 0

● **내보낸 프로젝트 파일을 복사한 경우**

Sysmac Studio에서 'Project_A'를 복사하여 'Project_A_copy'를 작성하고, 다운로드한 경우에도 컨트롤러 측의 POU명 및 로컬 변수 VarA의 현재값은 변경되지 않습니다.

Sysmac Studio 측	
프로젝트명	: Project_A_copy
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A_copy
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20

'Project_A'를 복사하는 시점에 존재하지 않았던 로컬 변수 VarB를 'Project_A'에 추가하여 다운로드하고, 그 후 'Project_A_copy'에 동일한 정의의 로컬 변수 VarB를 추가하여 다운로드한 경우에 컨트롤러 측의 로컬 변수 VarB의 현재값은 변경되지 않습니다.
'Project_A'에서 로컬 변수 VarB를 복사하거나 변수 테이블을 병합한 경우도 동일하게 컨트롤러 측의 로컬 변수 VarB의 현재값은 변경되지 않습니다.

Sysmac Studio 측	
프로젝트명	: Project_A_copy
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0
변수명	: VarB
VarB의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_A_copy
POU명	: POU_XXX
변수명	: VarA
VarA의 현재값	: 20
변수명	: VarB
VarB의 현재값	: 50

● **별도로 작성한 프로젝트에 동일한 이름의 POU를 작성**

Sysmac Studio에서 'Project_A'의 POU명과 동일한 'POU_A'를 갖는 'Project_B'를 작성하여 다운로드한 경우에 컨트롤러 측은 새로운 'POU_YYY' 및 새로운 로컬 변수 VarA로 취급합니다. 따라서 새 로컬 변수 VarA의 값은 초기값인 '0'이 됩니다.

Sysmac Studio 측	
프로젝트명	: Project_B
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_B
POU명	: POU_YYY
변수명	: VarA
VarA의 현재값	: 0

● **다른 프로젝트에 원래 프로젝트의 POU를 복사한 경우**

Sysmac Studio에서 'Project_B'를 작성하고, 'Project_A'에서 복사한 'POU_A'를 붙여넣고 다운로드한 경우에 컨트롤러 측은 새로운 'POU_YYY' 및 새로운 로컬 변수 VarA로 취급합니다. 따라서 새 로컬 변수 VarA의 값은 초기값인 '0'이 됩니다.

Sysmac Studio 측	
프로젝트명	: Project_B
POU명	: POU_A
변수명	: VarA
VarA의 초기값	: 0

다운로드

컨트롤러 측	
프로젝트명	: Project_B
POU명	: POU_YYY
변수명	: VarA
VarA의 현재값	: 0

유지 변수의 데이터 형식을 변경했을 때의 변수 값

Sysmac Studio에서 유지 속성이 '유지'인 변수의 데이터 형식을 변경한 후에 '유지'인 변수의 현재값을 포함한 복원 및 전송 기능, 다운로드 등의 조작을 실행한 경우의 컨트롤러측 변수의 값에 대해 설명합니다.

● 조작과 변수의 값

유지 속성이 '유지'인 변수의 동작에는 2가지 동작 패턴이 있습니다. 조작과 동작 패턴의 관계는 다음 표와 같습니다.

조작	변수의 값의 동작 패턴
• Sysmac Studio 컨트롤러 백업 기능에 의한 복원 • SD 메모리 카드 백업 기능에 의한 복원 • SD 메모리 카드 자동 전송 기능 • SD 메모리 카드 프로그램 전송 기능	패턴 1
• Sysmac Studio 변수/메모리 백업 기능에 의한 변수의 복원	패턴 2
• Sysmac Studio에서 다운로드	패턴 2*1

*1. CPU 유니트의 유니트 버전Ver.1.10 이하인 경우에는 패턴 1이 됩니다. 이 차이는 CPU 유니트의 유니트 버전에만 좌우됩니다. 프로젝트의 장치에 설정한 유니트 버전에는 좌우되지 않습니다.

아래에서 패턴 1과 패턴 2의 동작에 대해 데이터 형식별로 설명합니다.

- **기본 데이터 형식의 변경**

Sysmac Studio에서 변경하기 전의 변수는 다음과 같은 정의로 합니다.

변수명	데이터 형식	할당 대상
ABC	INT	–
DEF	INT	%D100

각 변수의 유지 속성은 '유지'로 하고, 현재값은 다음과 같다고 가정합니다.

변수/메모리	현재값
ABC	10
DEF	20

CJ 유니트용 메모리의 DM100과 DM101의 현재값이 다음과 같다고 가정합니다.

변수/메모리	현재값
D100	20
D101	50

데이터 형식의 변경 내용과 컨트롤러측의 변수의 동작의 관계는 다음 표와 같습니다.

변경 내용	패턴 1	패턴 2
변경 없음	변수의 값은 유지됩니다. ABC = 10 DEF = 20	
변수명을 'ABC'에서 'GHI'로 변경	변수 값은 초기값이 됩니다. GHI = 초기값	
변수 'ABC'의 데이터 형식을 INT에서 DINT로 변경	변수 값은 초기값이 됩니다. ABC = 초기값	
변수 'ABC'의 유지 속성을 '비유지'로 변경	변수 값은 초기값이 됩니다.*1 ABC = 초기값	
변수 'DEF'의 할당대상을 'D100'에서 'D101'로 변경	변수의 값은 변경한 할당대상의 값이 됩니다. DEF = 50	
변수 'DEF'의 할당대상을 삭제	변수 값은 초기값이 됩니다. DEF = 초기값	

*1. 유지 속성이 '비유지'인 변수를 '유지'로 변경한 경우도 변수의 값은 초기값이 됩니다.

- **구조체 형식의 변경**

Sysmac Studio에서 변경하기 전의 구조체 형식은 다음과 같은 정의로 합니다.

데이터 형식명	멤버	멤버의 데이터 형식
STR_1	x	INT
	y	INT

변수 ABC의 데이터 형식이 'STR_1'이고, 유지 속성은 '유지'로 하고, 각 멤버의 현재값이 다음과 같다고 가정합니다.

멤버	현재값
ABC.x	100
ABC.y	-200

데이터 형식의 변경 내용과 컨트롤러측의 변수의 동작의 관계는 다음 표와 같습니다.
유지되는 조건과 초기값이 되는 조건의 양쪽에 해당하는 경우에 변수의 값은 초기값이 됩니다.

변경 내용	패턴 1	패턴 2
변경 없음	멤버의 값은 유지됩니다. ABC.x = 100 ABC.y = -200	
변수명을 'ABC'에서 'DEF'로 변경	멤버의 값은 초기값이 됩니다. DEF.x = 초기값 DEF.y = 초기값	
멤버 'x'를 'x1'로 변경	멤버의 값은 유지됩니다. ABC.x1 = 100 ABC.y = -200	변경한 멤버의 값은 초기값이 됩니다. ABC.x1 = 초기값 ABC.y = -200
멤버 'x'의 데이터 형식을 INT에서 WORD로 변경*1 (데이터 크기는 동일)	멤버의 값은 유지됩니다. ABC.x = WORD#16#0064(= 100) ABC.y = -200	
멤버 'y'의 데이터 형식을 INT에서 UINT로 변경*1 (데이터 크기는 동일하고, 부호 있음에서 부호 없음으로 변경)	멤버의 값은 유지됩니다. 변경한 멤버도 그대로 대입되므로 절대값과 부호가 변화합니다. ABC.x = 100 ABC.y = 65336	
멤버 'y'의 데이터 형식을 INT에서 DINT로 변경*1 (데이터 크기가 확장)	멤버의 값은 초기값이 됩니다. ABC.x = 초기값 ABC.y = 초기값	멤버의 값은 유지됩니다. 변경한 멤버는 부호가 확장되고, 절대값과 부호는 변화하지 않습니다. ABC.x = 100 ABC.y = -200
멤버 'y'의 데이터 형식을 INT에서 UDINT로 변경*1 (데이터 크기가 확장하고, 부호 있음에서 부호 없음으로 변경)	멤버의 값은 초기값이 됩니다. ABC.x = 초기값 ABC.y = 초기값	멤버의 값은 유지됩니다. 변경한 멤버는 부호가 확장되고, 절대값과 부호가 변화합니다. ABC.x = 100 ABC.y = 4294967096
멤버 'y'의 데이터 형식을 INT에서 SINT로 변경*1 (데이터 크기가 축소)	멤버의 값은 초기값이 됩니다. ABC.x = 초기값 ABC.y = 초기값	데이터 크기를 축소한 멤버의 값은 초기값이 됩니다. ABC.x = 100 ABC.y = 초기값
멤버 'x'를 'y'에 멤버 'y'를 'x'로 변경	잘못된 멤버의 값은 유지됩니다. ABC.y = 100 ABC.x = -200	멤버의 값은 유지됩니다. ABC.y = -200 ABC.x = 100

변경 내용	패턴 1	패턴 2
멤버 'y' 다음에 INT의 멤버 'z'를 추가	멤버의 값은 초기값이 됩니다.*2 ABC.x = 초기값 ABC.y = 초기값 ABC.z = 초기값	기존 멤버의 값은 유지됩니다. 추가한 멤버의 값은 초기값이 됩니다. ABC.x = 100 ABC.y = -200 ABC.z = 초기값
멤버 'x'와 멤버 'y' 사이에 INT의 멤버 'z'를 추가	멤버의 값은 초기값이 됩니다. *2 ABC.x = 초기값 ABC.z = 초기값 ABC.y = 초기값	기존 멤버의 값은 유지됩니다. 추가한 멤버의 값은 초기값이 됩니다. ABC.x = 100 ABC.z = 초기값 ABC.y = -200
멤버 'x'의 배열의 요소 수 를 '1'에서 '3'으로 변경	멤버의 값은 초기값이 됩니다. *2 ABC.x[0] = 초기값 ABC.x[1] = 초기값 ABC.x[2] = 초기값 ABC.y = 초기값	기존 멤버의 값은 유지됩니다. 배열의 요소 수를 변경한 멤버는 선두에서 유지되고, 확장된 멤버의 값은 초기값이 됩니다. ABC.x[0] = 100 ABC.x[1] = 초기값 ABC.x[2] = 초기값 ABC.y = -200

*1. 데이터 형식을 변경하는 경우에는 암시적 캐스트에 따르는 것으로 가정합니다. 암시적 캐스트에 대해서는 📖'암시적 캐스트(6-127페이지)'를 참조하십시오.

*2. 다운로드하기 전에 변수의 값을 업로드해두고, 다운로드한 후 변수의 값을 다운로드하면 기존 멤버의 값을 유지할 수 있습니다.

- **공용체 형식의 변경**

Sysmac Studio에서 변경하기 전의 공용체 형식은 다음과 같은 정의로 합니다.

데이터 형식명	멤버	멤버의 데이터 형식
UNI_1	x	WORD
	y	DWORD

변수 ABC의 데이터 형식이 'UNI_1'이고, 유지 속성은 '유지'로 하고, 각 멤버의 현재값이 다음과 같다고 가정합니다.

멤버	현재값
ABC	WORD#16#0100

데이터 형식의 변경 내용과 컨트롤러측의 변수의 동작의 관계는 다음 표와 같습니다.
유지되는 조건과 초기값이 되는 조건의 양쪽에 해당하는 경우에 변수의 값은 초기값이 됩니다.

변경 내용	패턴 1	패턴 2
변경 없음	멤버의 값은 유지됩니다. ABC = WORD#16#0100	
변수명을 'ABC'에서 'DEF'로 변경	멤버의 값은 초기값이 됩니다. DEF = 초기값	
멤버 'x'의 데이터 형식을 WORD에서 BYTE로 변경*1 (공용체 형식으로서의 데이터 크기는 동일)	멤버의 값은 유지됩니다. ABC = WORD#16#0100	
멤버 'x'의 데이터 형식을 WORD에서 LWORD로 변경*1 (공용체 형식으로서의 데이터 크기가 확장)	멤버의 값은 초기값이 됩니다. ABC = 초기값	멤버의 값은 유지됩니다. ABC = WORD#16#0100
멤버 'y'의 데이터 형식을 DWORD에서 BYTE로 변경*1 (공용체 형식으로서의 데이터 크기가 축소)	멤버의 값은 초기값이 됩니다. ABC = 초기값	
멤버 'x'를 'y'에 멤버 'y'를 'x'로 변경	멤버의 값은 유지됩니다. ABC = WORD#16#0100	
멤버 'y' 다음에 BYTE의 멤버 'z'를 추가 (공용체 형식으로서의 데이터 크기는 동일)	멤버의 값은 유지됩니다. ABC = WORD#16#0100	
멤버 'y' 다음에 LWORD의 멤버 'z'를 추가 (공용체 형식으로서의 데이터 크기가 확장)	멤버의 값은 초기값이 됩니다. ABC = 초기값	멤버의 값은 유지됩니다. ABC = WORD#16#0100
멤버 'y' 다음에 BYTE의 멤버 'z'를 추가하고 멤버 'y'를 삭제 (공용체 형식으로서의 데이터 크기가 축소)	멤버의 값은 초기값이 됩니다. ABC = 초기값	

*1. 데이터 형식을 변경하는 경우에는 암시적 캐스트에 따르는 것으로 가정합니다. 암시적 캐스트에 대해서는 '암시적 캐스트(6-127페이지)'를 참조하십시오.

- **배열 변수의 변경**

Sysmac Studio에서 변경하기 전의 배열 변수는 다음과 같은 정의로 합니다.

변수명	데이터 형식
ABC	ARRAY[1..3] OF INT

변수ABC의 유지 속성은 '유지'로 하고, 각 요소의 현재값이 다음과 같다고 가정합니다.

멤버	현재값
ABC[1]	100
ABC[2]	200
ABC[3]	300

데이터 형식의 변경 내용과 컨트롤러측의 변수의 동작의 관계는 다음 표와 같습니다.
유지되는 조건과 초기값이 되는 조건의 양쪽에 해당하는 경우에 변수의 값은 초기값이 됩니다.

변경 내용	패턴 1	패턴 2
변경 없음	요소의 값은 유지됩니다. ABC[1] = 100 ABC[2] = 200 ABC[3] = 300	
변수명을 'ABC'에서 'DEF'로 변경	요소의 값은 초기값이 됩니다. DEF[1] = 초기값 DEF[2] = 초기값 DEF[3] = 초기값	
요소의 데이터 형식을 INT에서 UINT로 변경*1 (데이터 크기는 동일)	요소의 값은 초기값이 됩니다. ABC[1] = 초기값 ABC[2] = 초기값 ABC[3] = 초기값	
요소의 데이터 형식을 INT에서 DINT로 변경*1 (데이터 크기가 확장)	요소의 값은 초기값이 됩니다. ABC[1] = 초기값 ABC[2] = 초기값 ABC[3] = 초기값	
요소의 데이터 형식을 INT에서 SINT로 변경*1 (데이터 크기가 축소)	요소의 값은 초기값이 됩니다. ABC[1] = 초기값 ABC[2] = 초기값 ABC[3] = 초기값	
배열 요소의 종료 번호를 '3'에서 '4'로 변경*2 (요소 수가 '3'에서 '4'로 확장)	요소의 값은 초기값이 됩니다. ABC[1] = 초기값 ABC[2] = 초기값 ABC[3] = 초기값 ABC[4] = 초기값	기존 요소 번호의 요소 값은 유지됩니다. 새로운 요소 번호의 요소 값은 초기값이 됩니다. ABC[1] = 100 ABC[2] = 200 ABC[3] = 300 ABC[4] = 초기값
배열 요소의 시작 번호를 '1'에서 '0'으로 변경*2 (요소 수가 '3'에서 '4'로 확장)	요소의 값은 초기값이 됩니다. ABC[0] = 초기값 ABC[1] = 초기값 ABC[2] = 초기값 ABC[3] = 초기값	기존 요소 번호의 요소 값은 유지됩니다. 새로운 요소 번호의 요소 값은 초기값이 됩니다. ABC[0] = 초기값 ABC[1] = 100 ABC[2] = 200 ABC[3] = 300
배열 요소의 종료 번호를 '3'에서 '2'로 변경*2 (요소 수가 '3'에서 '2'로 축소)	요소의 값은 초기값이 됩니다. ABC[1] = 초기값 ABC[2] = 초기값	기존 요소 번호의 요소 값은 유지됩니다. ABC[1] = 100 ABC[2] = 200

변경 내용	패턴 1	패턴 2
배열 요소의 시작 번호를 '1'에서 '2'로 변경*2 (요소 수가 '3'에서 '2'로 축소)	요소의 값은 초기값이 됩니다. ABC[2] = 초기값 ABC[3] = 초기값	기존 요소 번호의 요소 값은 유지됩니다. ABC[2] = 200 ABC[3] = 300
배열 요소의 시작 번호를 '1'에서 '3', 종료 번호를 '3'에서 '5'로 변경*2 (요소 수는 동일)	요소 수가 동일한 경우에 요소 값은 선두에서부터 유지됩니다.*3 ABC[3] = 100 ABC[4] = 200 ABC[5] = 300	기존 요소 번호의 요소 값은 유지됩니다. 새로운 요소 번호의 요소 값은 초기값이 됩니다. ABC[3] = 300 ABC[4] = 초기값 ABC[5] = 초기값

*1. 데이터 형식을 변경하는 경우에는 암시적 캐스트에 따르는 것으로 가정합니다. 암시적 캐스트에 대해서는 '암시적 캐스트(6-127페이지)'를 참조하십시오.
*2. 변수의 데이터가 기본 데이터 형식 또는 파생 데이터 형식 중 어느 것이라도 동일한 결과가 됩니다.
*3. 요소 수가 확장 또는 축소되는 경우에 모든 요소 값은 초기값이 됩니다.

- **열거 형식의 변경**

Sysmac Studio에서 변경하기 전의 열거 형식은 다음과 같은 정의로 합니다.

데이터 형식명	열거자	값
ENU_1	x	1
	y	2

변수ABC의 데이터 형식이 'ENU_1'이고, 유지 속성은 '유지'로 하고, 변수의 현재값이 다음과 같다고 가정합니다.

멤버	현재값
ABC	1(= x)

데이터 형식의 변경 내용과 컨트롤러측의 변수의 동작의 관계는 다음 표와 같습니다.

변경 내용	패턴 1	패턴 2
변경 없음	변수의 값은 유지됩니다. ABC = 1 (= x)	
변수명을 'ABC'에서 'DEF'로 변경	변수 값은 초기값이 됩니다. ABC = 초기값	
열거자 'y' 다음에 값이 '3'인 열거자 'z'를 추가	변수의 값은 유지됩니다. ABC = 1 (= x)	
열거자 'x'를 삭제	열거자가 존재하지 않는 경우도 변수의 값은 유지됩니다 ABC = 1	
열거자 'X'의 값을 '1'에서 '5'로 변경	열거자의 값을 변경해도 변수의 값은 유지됩니다. ABC = 1	

6-3-10 펑션 블록(FB)의 인스턴스

인스턴스화된 FB는 데이터 형식의 하나로 로컬 변수 테이블에 등록 및 표시됩니다.

참고

FB의 인스턴스는 인스턴스가 기술된 프로그램의 로컬 변수 취급되며, 해당 프로그램의 로컬 변수 테이블에 등록 및 표시됩니다. 글로벌 변수가 되지 않는 점에 주의하십시오.

6-3-11 변수 값 모니터링

변수 값은 Sysmac Studio의 감시 창에서 모니터링할 수 있습니다.

1 **[표시]–[감시 창]을 선택합니다.**
감시 창이 표시됩니다.

2 **컨트롤러에 온라인 연결하고, 다음 중 하나의 방법으로 변수를 등록합니다.**

(1) 감시 창의 이름 셀에 변수명을 입력합니다.

(2) 편집기 및 변수 테이블에서 감시 창으로 드래그 앤 드롭하여 등록합니다.
해당 변수의 현재값이 표시됩니다.

6-3-12 변수명 등의 프로그램 관련 이름의 제한

다음은 프로그램 관련 이름의 제한 목록입니다.

문자 제한

프로그램 관련 이름	사용 가능한 문자	예약어	멀티바이트 문자 사용 가능 여부	대문자/소문자 구별	최대 크기 (NULL 포함하지 않음)	문자 코드
변수명 (POU 인스턴스명 포함)	● **사용할 수 있는 문자** • 0~9, A~Z, a~z • 반각 문자 • _(언더 바) • 멀티바이트 문자(예: 한글, 일본어) 예약어는 뒤에 설명하는 '예약어'를 참조하십시오. ● **조합해서 사용할 수 없는 문자** • 숫자(0~9)로 시작하는 문자열 • P_로 시작하는 문자열 • 언더 바로 시작하는 문자열 • 언더 바(_)를 2개 이상 연속한 문자열 • 언더 바(_)로 끝나는 문자열 • 식별자를 구성하는 문자열의 접두사 또는 접미사에 1문자 이상의 확장 공백류(전각 공백 등 Unicode까지 확장한 공백 계열 문자)를 포함하는 문자열	뒤에 설명하는 '예약어' 참조	가능	사용 안 함	127바이트	UTF-8*1
POU 정의명						
데이터 형식명						
구조체 멤버명 공용체 멤버명						
열거자						
태스크 이름					63바이트	
이름 공간명					93바이트	
변수명의 전체 경로					네트워크 변수: 255 바이트 그 외: 511 바이트	
장치명					127바이트	
섹션명				사용		
축 이름				사용 안 함		
축 그룹명						
캠 테이블명						

*1. UTF-8에서는 반각 영숫자 1문자는 1바이트, 멀티바이트 문자는 1바이트 이상을 사용합니다. 한글은 3바이트가 기준입니다.

예약어

다음과 같은 이름은 프로그램 확인 시 에러가 됩니다.

- "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 명령과 동일한 이름
- "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"의 명령과 동일한 이름
- 시스템이 내부에서 사용하는 예약어

이름 중복 금지

다음과 같은 이름의 중복은 금지합니다. 프로그램 확인 시 에러가 됩니다.

- 동일한 CPU 유니트에서의 글로벌 변수명 중복
- 동일한 POU 내에서의 변수명 중복
- 동일한 POU 내에서의 섹션명 중복
- 동일한 구조체 또는 공용체 내에서의 멤버명 중복
- 동일한 열거 형식 내에서의 열거자 중복
- 로컬 변수명과 글로벌 변수명의 중복
- POU명과 데이터 형식명의 중복
- 데이터 형식명과 변수명의 중복
- 열거자와 다른 열거 형식 열거자의 중복
- 열거자와 변수명의 중복

6-4 상수(리터럴)

여기에서는 상수에 대해 자세하게 설명합니다.

6-4-1 상수의 개념

변수의 값은 해당 변수에 대입하는 데이터에 따라 바뀝니다. 이에 비해 값이 바뀌지 않는 것이 상수입니다.

상수는 변수와 달리 메모리에 저장된 정보는 아닙니다. 상수는 선언하지 않아도 POU의 알고리즘 부분에 기술할 수 있습니다.

NJ/NX 시리즈에서는 상수에도 변수와 동일하게 '데이터 형식'이라는 개념이 있습니다.

6-4-2 각 데이터 형식의 기술 형식

각 데이터 형식의 상수 기술 형식은 다음과 같습니다. 상수를 결정된 기술 형식 이외로 기술하면 빌드 시 에러가 됩니다.

부울 형식

부울 형식은 데이터 형식명으로 'BOOL'을 사용합니다. 또한, 값으로 1 또는 0 이외에 'TRUE'와 'FALSE'를 사용할 수 있습니다. 기술 형식과 의미는 다음과 같습니다.

기술 형식	의미
TRUE 또는 FALSE	TRUE, BOOL#1, BOOL#TRUE, 1은 등가입니다. FALSE, BOOL#0, BOOL#FALSE, 0은 등가입니다.
BOOL#1 또는 BOOL#0	
BOOL#TRUE 또는 BOOL#FALSE	
1 또는 0*1	

*1. 1 또는 0의 기술은 Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다. Sysmac Studio Ver.1.02 이하에서 1 또는 0를 기술하면 빌드 시 에러가 됩니다.

비트 열 형식

비트 열 형식은 데이터 형식명으로 'BYTE', 'WORD', 'DWORD', 'LWORD' 중 하나를 사용합니다. 또한, 기수로 '2', '8', '10', '16' 중 하나를 사용합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
{데이터 형식명}#{기수}#{수치}	WORD#16#0064
{기수}#{수치}*1	16#0064
{수치}*1*2	100

*1. 데이터 형식명을 생략한 기술은 Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다. Sysmac Studio Ver.1.02 이하에서 데이터 형식명을 생략한 기술을 하면 빌드 시 에러가 됩니다.
*2. 기수는10 즉, 10진수라고 해석됩니다.

버전 관련 정보

비트 열 형식의 기수로 '10'을 기술할 수 있는 것은 Sysmac Studio Ver.1.03 이상입니다. Sysmac Studio Ver1.02 이하에서 기수를 '10'으로 하면 빌드 시 에러가 됩니다.

정수 형식

정수 형식은 데이터 형식명으로 'SINT', 'USINT', 'INT', 'UINT', 'DINT', 'UDINT', 'LINT', 'ULINT' 중 하나를 사용합니다. 또한, 기수로 '2', '8', '10', '16' 중 하나를 사용합니다.
기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
{데이터 형식명}#{기수}#{수치}	INT#10#-1
{기수}#{수치}*1	10#-1
{데이터 형식명}#{수치}*2	INT#-1
{수치}*1*2	-1

*1. 데이터 형식명을 생략한 기술은 Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다. Sysmac Studio Ver.1.02 이하에서 데이터 형식명을 생략한 기술을 하면 빌드 시 에러가 됩니다.
*2. 기수는10 즉, 10진수라고 해석됩니다.

실수 형식

실수 형식은 데이터 형식명으로 'REAL'과 'LREAL' 중 하나를 사용합니다. 또한, '10'만 실수 형식으로 사용할 수 있습니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
{데이터 형식명}#{기수}#{수치}	LREAL#10#-3.14
{기수}#{수치}*1	10#-3.14
{데이터 형식명}#{수치}*2	LREAL#-3.14
{수치}*1*2	-3.14

*1. 데이터 형식명을 생략한 기술은 Sysmac Studio Ver.1.03 이상에서 사용할 수 있습니다. Sysmac Studio Ver.1.02 이하에서 데이터 형식명을 생략한 기술을 하면 빌드 시 에러가 됩니다.
*2. 기수는10 즉, 10진수라고 해석됩니다.

지속 시간 형식

지속 시간 형식은 데이터 형식명으로 'TIME'과 'T' 중 하나를 사용합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
TIME#{일}d{시}h{분}m{초} s{밀리초}ms	TIME#61m5s
T#{일}d{시}h{분}m{초}s{밀리초}ms	T#61m5s

지속 시간 형식 상수의 기술에는 다음과 같은 규칙이 있습니다.

- {일}, {시}, {분}, {초}, {밀리초}를 모두 기술할 필요는 없습니다. 그 중 하나 이상을 기술합니다.
- 'TIME#12d3.5h'와 같이 소수를 기술할 수 있습니다.
- 시간의 유효 범위를 초과한 기술이 가능합니다. 예를 들어 'T#-61m5s'는 'T#-1h1m5s'와 동일한 의미입니다.
- 수치는 10진수로 해석됩니다. 10진수 이외의 수를 기술한 경우에는 빌드 시 에러가 됩니다.
- 시간의 배열을 바꾼 기술이 가능합니다. 예를 들어 'T#1h2d'는 'T#2d1h'와 동일한 의미입니다.

날짜 형식

날짜 형식은 데이터 형식명으로 'DATE'와 'D' 중 하나를 사용합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
DATE#{연}-{월}-{일}	DATE#2010-1-10
D#{연}-{월}-{일}	D#2010-1-10

날짜 형식 상수의 기술에는 다음과 같은 규칙이 있습니다.

- {연}, {월}, {일} 각각의 선두에 하나 이상의 0을 붙일 수 있습니다. 예를 들어 'DATE#2010-01-10'은 'D#2010-1-10'과 동일한 의미입니다.
- 날짜의 유효 범위를 초과하여 기술하면 빌드 시 에러가 됩니다. 예를 들어 'D#2010-01-35'는 에러가 됩니다.
- 수치는 10진수로 해석됩니다. 10진수 이외의 수를 기술한 경우에는 빌드 시 에러가 됩니다.

시각 형식

시각 형식은 데이터 형식명으로 'TIME_OF_DAY'와 'TOD' 중 하나를 사용합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
TIME_OF_DAY#{시}:{분}:{초}	TIME_OF_DAY#23:59:59.999999999
TOD#{시}:{분}:{초}	TOD#23:59:59.999999999

시각 형식 상수의 기술에는 다음과 같은 규칙이 있습니다.

- {시}, {분}, {초} 각각의 선두에 하나 이상의 0을 붙일 수 있습니다. 예를 들어 'TOD#23:01:01'과 'TOD#23:1:1'은 동일한 의미입니다.
- 시분초의 유효 범위를 초과한 숫자를 기술하면 빌드 시 에러가 됩니다. 예를 들어 'TOD#24:00:00'은 에러가 됩니다.
- 수치는 10진수로 해석됩니다. 10진수 이외의 수를 기술한 경우에는 빌드 시 에러가 됩니다.

날짜 시각 형식

날짜 시각 형식은 데이터 형식명으로 'DATE_AND_TIME'과 'DT' 중 하나를 사용합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
DATE_AND_TIME#{연}-{월} -{일}-{시}:{분}:{초}	DATE_AND_TIME#2010-10-10-23:59:59.123
DT#{연}-{월}-{일}-{시}:{분} :{초}	DT#2010-10-10-23:59:59.123

날짜 시각 형식 상수의 기술에는 다음과 같은 규칙이 있습니다.

- {연}, {월}, {일}, {시}, {분}, {초}의 각각의 선두에 하나 이상의 0을 붙일 수 있습니다. 예를 들어 'DT#2010-01-10-23:01:01'과 'DT#2010-1-10-23:1:1'은 동일한 의미입니다.
- 연월일시분초의 유효 범위를 초과한 숫자를 기술하면 빌드 시 에러가 됩니다. 예를 들어 'DT#2010-01-35-00:00:00' 및 'DT#2010-01-30-24:00:00'은 에러가 됩니다.
- 수치는 10진수로 해석됩니다. 10진수 이외의 수를 기술한 경우에는 빌드 시 에러가 됩니다.

문자열 형식

문자열 형식은 문자열을 2개의 반각 작은 따옴표로 묶어서 기술합니다. 또한, 데이터 형식명으로 'STRING'을 사용할 수도 있습니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
'{문자열}'	'이것은 문자열'
STRING#'{문자열}'*1	STRING#'이것은 문자열'

*1. 'STRING'을 사용한 기술은 Sysmac Studio Ver.1.08 이상에서 가능합니다.

문자열의 기술에는 다음과 같은 규칙이 있습니다.

- 문자 수 0인 문자열도 기술할 수 있습니다. 그 경우의 기술은 <">가 됩니다.
- 다음과 같이 여러 행에 걸쳐 있는 문자열을 기술하면 빌드 시 에러가 됩니다.
  strVar := 'ABC
  DEF'
- 탭 및 행바꿈 코드 등을 기술하는 경우에는 선두에 '$'를 붙인 이스케이프 문자를 사용합니다. 이스케이프 문자의 이름과 의미는 다음과 같습니다.

이스케이프 문자	이름	의미
$$	반각 달러 문자	반각 달러 기호 문자($: 문자 코드 0x24)
$'	반각 작은 따옴표	반각 작은 따옴표 기호 문자(': 문자 코드 0x27)
$"	반각 큰 따옴표	반각 큰 따옴표 기호 문자(": 문자 코드 0x22)
$L 또는 $l	행바꿈	커서를 다음 행으로 이동합니다. LF 제어 문자(Line Feed: 문자 코드 0x0A)
$N 또는 $n	행바꿈	커서를 다음 행으로 이동합니다. NL 제어 문자(New Line: 문자 코드 0x0A)
$P 또는 $p	페이지 바꿈	커서를 다음 페이지로 이동합니다. FF 제어 문자(Form Feed: 문자 코드 0x0C)
$R 또는 $r	복귀	커서를 행의 선두로 이동합니다. CR 제어 문자(Carriage Return: 문자 코드 0x0D)
$T 또는 $t	탭 문자	탭을 표시합니다. 탭 문자(문자 코드 0x09)
$(문자 코드)	문자 코드 직접 지정	문자 코드를 2자리의 16진수로 직접 지정합니다. 문자 코드의 범위는 00~FF입니다. 예를 들어 '$L'은 '$0A'와 동일한 의미입니다. 또한, 2바이트 이상의 문자는 1바이트마다 $ 를 붙입니다. 예를 들어 문자 코드가 0xE38182인 히라가나의 '아(あ)'는 '$E3$81$82'라고 기술합니다.

- 문자 코드를 직접 지정하여 문자를 기술할 수도 있습니다. 그 경우도 선두에 '$'를 붙입니다. 문자 코드는 2자리의 16진수로 기술합니다. 예를 들어 행바꿈 '$L'의 문자 코드는 0x0A이므로 '$0A'라고 기술할 수 있습니다.
- 문자 코드를 직접 지정하여 2바이트 이상의 문자를 기술하는 경우에는 1바이트마다 '$'를 붙입니다. 예를 들어 문자 코드가 0xEAB080인 한글의 '가'는 'EAB0$80'이라고 기술합니다.

열거 형식

열거 형식은 열거 형식의 데이터 형식명과 열거자를 기술합니다. 기술 형식과 기술 예는 다음과 같습니다.

기술 형식	기술 예
{열거 형식의 데이터 형식명}#{열거자}	_eDAYOFWEEK#_WED



참고

파라메터의 데이터 형식으로 열거 형식이 지정된 FB 및 FUN에 열거자를 전달하는 경우에는 열거 형식의 데이터 형식명을 생략하고 열거자만 기술할 수 있습니다.
예를 들어 BinToBCDs 명령의 입력 변수 'Format'에는 열거 형식_eBCD_FORMAT이 지정되어 있습니다. 따라서 열거 형식의 데이터 형식명과 열거자를 기술한 '_eBCD_FORMAT#_BCD0' 이외에 열거 형식의 데이터 형식명을 생략한 '_BCD0'도 기술할 수 있습니다.

6-5 프로그램 언어

여기에서는 각 프로그램 언어에 대해 자세하게 설명합니다.

Sysmac Studio에서의 프로그램 언어 입력 방법은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

6-5-1 프로그램 언어의 개념

POU(프로그램, FB, FUN) 내의 알고리즘을 표현하기 위한 언어를 프로그램 언어라고 합니다.

NJ/NX 시리즈에서 사용할 수 있는 프로그램 언어는 래더도 언어(LD), ST 언어(Structured Text: 구조화 텍스트)의 2종류입니다.

6-5-2 래더도 언어(LD)

래더도 언어는 전기 회로의 이미지로 기술한 그래픽을 사용하는 프로그램 언어입니다. FB 및 FUN을 포함한 처리의 각 부품이 모두 그림으로 표시되어 있고, 이들을 직선으로 연결하여 일련의 알고리즘을 구축합니다. 래더도 언어로 쓴 알고리즘을 래더도라고 합니다.

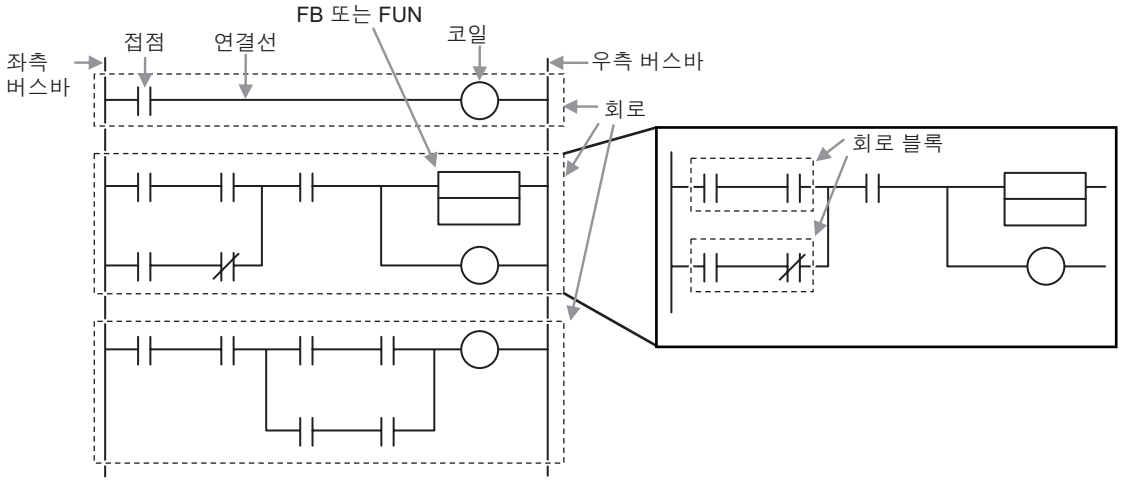
래더도 언어의 구성 요소

래더도 언어는 좌우의 버스바, 연결선, 래더도 구문(접점, 코일), FB 및 FUN로 구성됩니다(*).

* 점프 명령 및 라벨 명령뿐인 경우에는 점프 및 레이블을 나타내는 기호로 표시합니다.

알고리즘은 이들을 연결한 여러 회로로 구성됩니다.

회로는 좌측 버스바에서 우측 버스바까지의 구성 요소의 연결을 말합니다. 그리고 회로는 (논리 시작을 나타내는)LD/LDN 명령을 선두로 한 회로 블록으로 구성됩니다.



- **버스바**

  래더도의 좌측 및 우측의 세로선을 버스바라고 합니다.

  각각의 버스바는 TRUE 또는 FALSE 중 하나의 상태를 취합니다. 이는 래더도를 전기 회로로 간주한 경우의 통전 상태와 비통전 상태를 나타냅니다.

  래더도가 쓰인 POU가 기동하면 좌측 버스바 값이 TRUE가 됩니다. 그리고 좌측 버스바에 연결된 접점 등의 구성 요소에도 TRUE 상태가 전달됩니다. 이들 구성 요소의 동작에 따라 TRUE의 상태가 차례로 우측 방향으로 전달됩니다.

  이 TRUE 상태의 흐름을 '전력 흐름'이라고 합니다. 버스바은 전력 흐름의 원천이라고 말할 수 있습니다.

- **연결선**
  버스바 및 각종 구성 요소를 연결하는 수평 방향의 직선을 연결선이라고 합니다.
  연결선은 TRUE 또는 FALSE 중 어느 한 상태를 취하며, 전력 흐름을 우측 방향으로 전달하는 역할을 합니다.

- **접점**
  접점은 연결선 상에 배치되고, 전력 흐름을 받으면 동작합니다.
  접점에는 여러 종류가 있고, 각각의 사양에 따라 좌측에서 온 전력 흐름을 우측으로 전달하거나 차단합니다.
  접점이 전력 흐름을 우측으로 전달하면 접점 우측의 연결선의 상태가 TRUE가 되고, 차단하면 FALSE가 됩니다.
  각 접점 사양의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

- **코일**
  코일은 연결선 상에 배치되고, 전력 흐름을 받으면 동작합니다.
  코일의 기능은 변수에 TRUE 또는 FALSE의 값을 쓰는 것입니다.
  코일에도 여러 종류가 있습니다. 각 코일 사양의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

- **FB, FUN**
  FB, FUN은 연결선상에 배치되고, 전력 흐름을 받으면 동작합니다.
  각 명령 사양의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

래더도 실행 순서

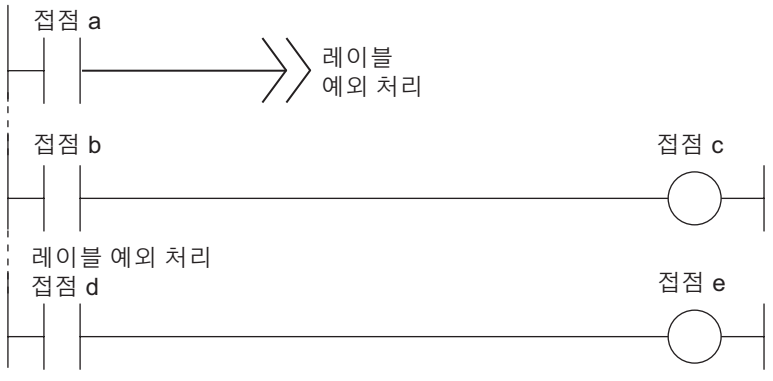
접점, 코일, FB, FUN은 전력 흐름을 받으면 실행됩니다.
실행 순서는 래더도의 위에서 아래입니다. 또한, 동일한 높이의 요소에 대해서는 좌측에서 우측 순서로 실행됩니다.

래더도의 종료

래더도는 위에서부터 순서대로 실행되고, 가장 아래까지 실행되면 처리를 종료합니다.
단, 도중에 END 명령 또는 RETURN 명령이 있었으면 그 시점에서 처리를 종료합니다. 이들 명령보다 나중에 쓴 처리는 실행되지 않습니다.

래더도의 실행 제어

래더도는 기본적으로는 위에서부터 순서대로 실행되지만 실행 제어 명령을 사용하면 실행 순서를 변경할 수 있습니다.
예를 들어 다음과 같은 경우에 접점 a가 TRUE가 되면 '예외 처리'의 라벨이 부가된 지점으로 실행이 이동합니다.



래더도에서의 FB, FUN 연결

- **연결 형태**

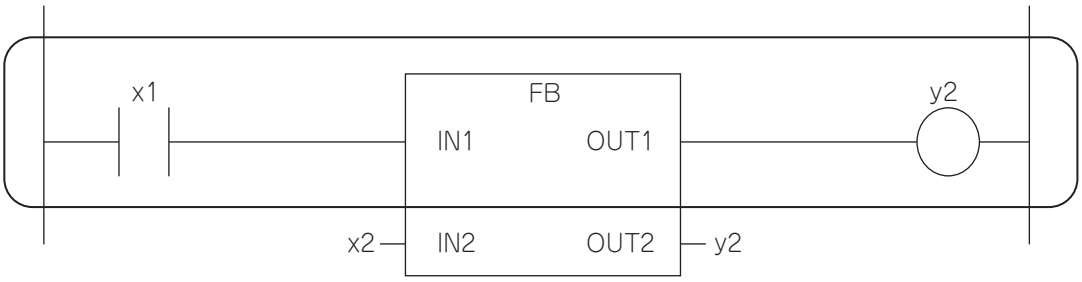
FB 및 FUN에는 다음과 같은 2가지 연결 형태가 있습니다.

(1) 전력 흐름 입력, 출력

래더도 언어에서 FB 및 FUN의 입력 변수를 좌측 버스바, 출력 변수를 우측 버스바에 연결할 수 있는 경우의 BOOL 형식의 입력, BOOL 형식의 출력을 말합니다.

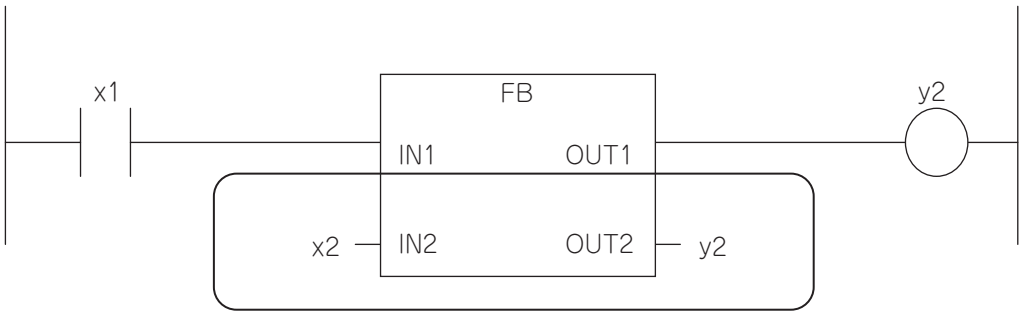
예)

좌측 버스바에 연결되는 전력 흐름은 접점(CONTACT), 우측 버스바에는 코일(COIL)이 연결됩니다.



(2) 파라메터 입력, 파라메터 출력

래더도 언어에서 FB 및 FUN의 입력 변수, 출력 변수를 좌측 버스바과 우측 버스바에 연결할 수 없는 경우를 말합니다.



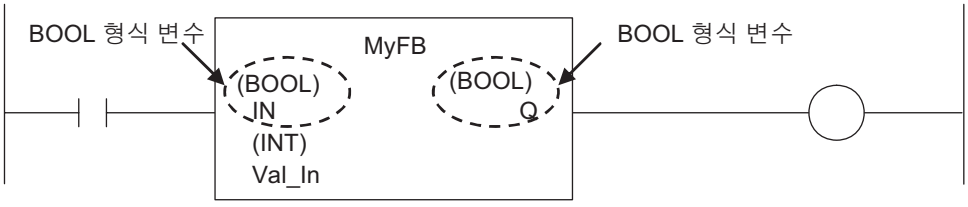
입력 및 출력의 파라메터에는 다음과 같이 변수 또는 상수를 지정할 수 있습니다.

FB/FUN 측 변수	입력 파라메터	출력 파라메터
입력 변수	변수 또는 상수 지정 가능	−
출력 변수	−	변수만 지정 가능
입출력 변수	변수만 지정 가능	변수만 지정 가능

- **BOOL 형식 변수의 수**

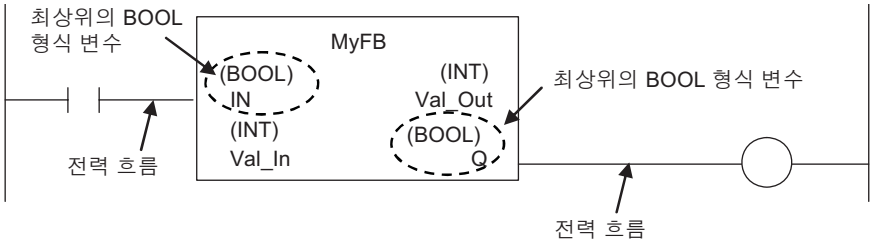
FB 및 FUN의 입력 및 출력에는 반드시 BOOL 형식 변수(EN, ENO 포함)가 하나 이상 필요합니다.

예)

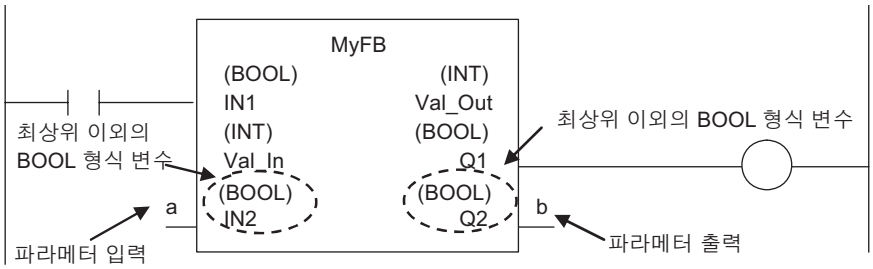


- **BOOL 형식 변수의 위치에 의한 연결 형태**

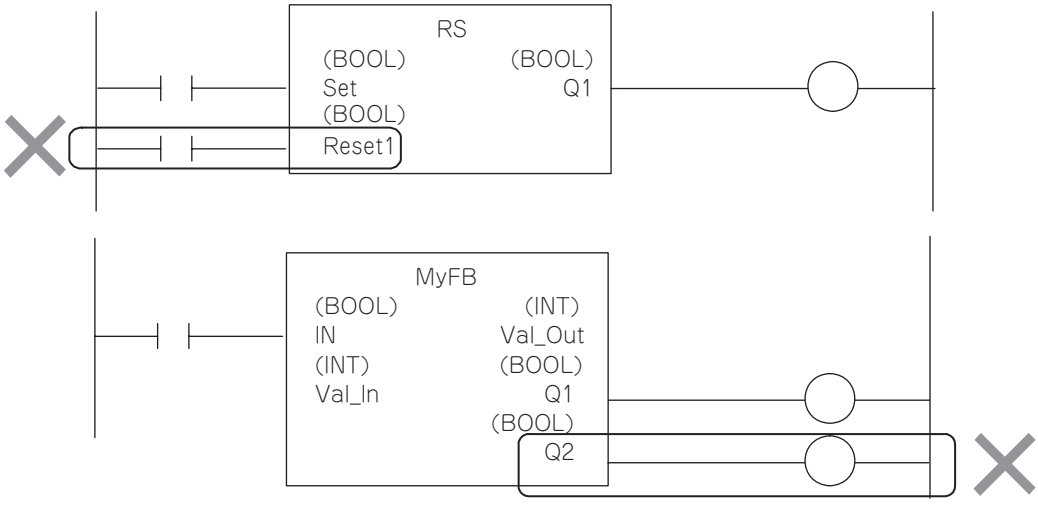
  최상위에 있는 BOOL 형식 변수는 좌측 버스바 및 우측 버스바에 연결됩니다. 즉, 전력 흐름 입력 및 전력 흐름 출력이 됩니다.



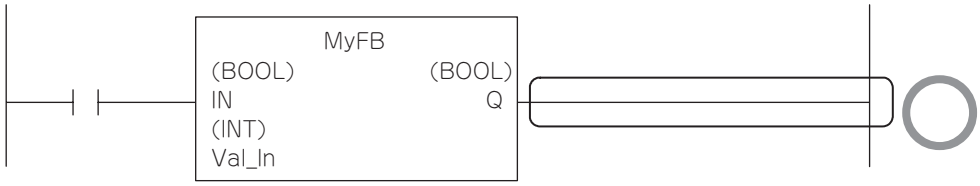
FB 및 FUN에 대한 전력 흐름 입력, 전력 흐름 출력은 각각 하나뿐입니다.
최상위 이외의 BOOL 형식 변수는 파라메터 입력 및 파라메터 출력이 됩니다.



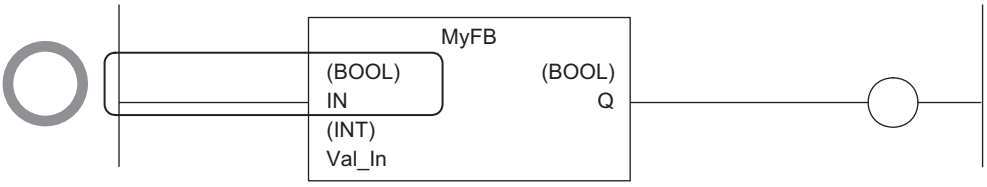
따라서 다음과 같이 복수의 BOOL 형식 변수를 좌측 버스바 또는 우측 버스바에 연결할 수 없습니다.



우측 버스바에 OUT 명령 등이 없고, 직접 연결해도 됩니다.

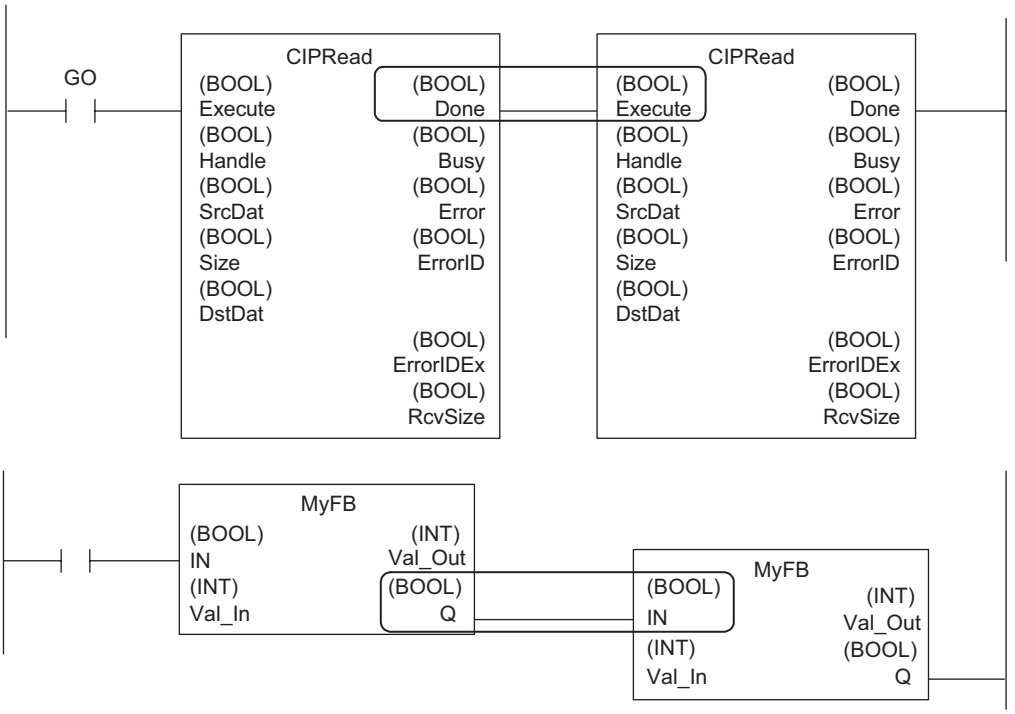


LD 명령 등이 없고 좌측 버스바에 직접 연결해도 됩니다.



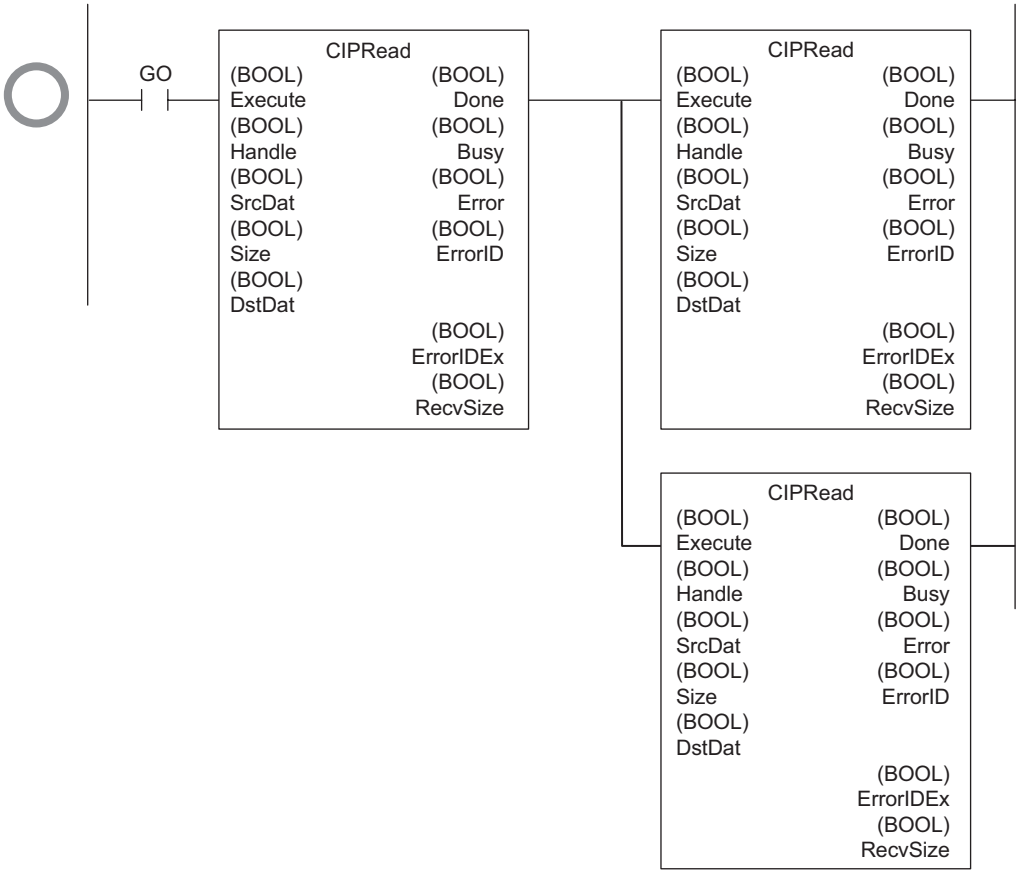
- **캐스케이드 연결**

  FB/FUN의 출력을 다른 FB/FUN의 입력과 하는 캐스케이드 연결은 전력 흐름 출력과 전력 입력 간에서만 가능합니다.



• 전력 흐름 출력의 분기는 가능합니다.

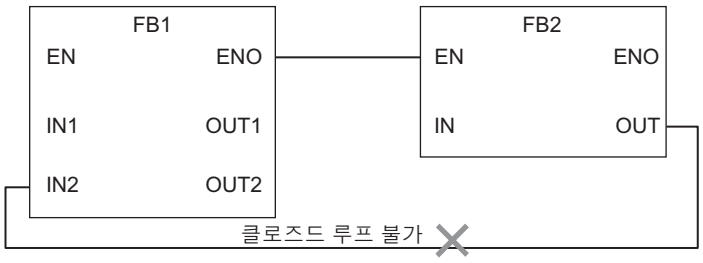
예)



제한

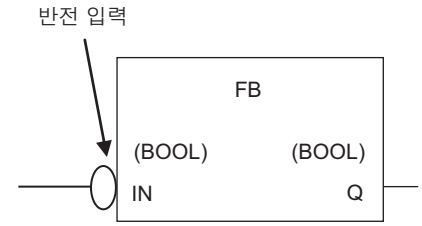
- 클로즈드 루프 및 연결선의 교차는 할 수 없습니다.

  예)



- **입력의 반전**

  BOOL 형식의 입력 변수에 대해 값을 반전하여 입력할 수 있습니다.



TRUE→FALSE

FALSE→TRUE

인라인 ST

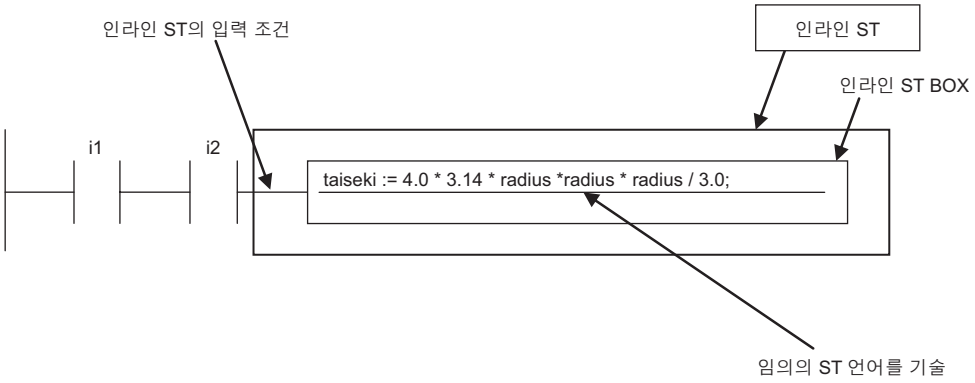
- **개요**

  인라인 ST는 래더도 내의 '인라인 ST BOX'라는 상자(공백의 텍스트 입력 영역) 안에 ST 언어를 기술하는 래더도 언어의 요소입니다.

  이것으로 숫자 연산 처리 및 문자열 조작 처리를 래더도 언어 내에서 간단하게 기술할 수 있습니다.

  '인라인 ST BOX'에 대한 연결선이 입력 조건이 되고, 그에 따라 ST BOX 내의 ST 언어가 실행됩니다.

  다음 그림과 같이 실행됩니다.



또한, 인라인 ST는 래더도 언어로 기재된 회로 요소의 일종으로 취급됩니다. 따라서 FB 및 FUN과 달리 입력 변수, 출력 변수, 입출력 변수는 없습니다.

- **인라인 ST 내의 ST 언어의 제한**

  인라인 ST BOX 내에는 임의의 ST 언어 프로그램을 기술할 수 있습니다.

- **인라인 ST 실행 조건**

  인라인 ST 실행 조건은 다음과 같습니다.

상태	동작
입력 조건이 TRUE인 경우	입력 조건에 따라서 동작 입력 조건은 전력 흐름 또는 좌측 버스바 직접 연결이 가능합니다. 상승 또는 하강 지정은 입력 조건의 접점으로 지정
입력 조건이 FALSE인 경우	아무것도 하지 않음
마스터 컨트롤에 의해 리셋한 경우	아무것도 하지 않음

- **인라인 ST 내의 변수의 유효 범위(Scope)**

  인라인 ST에서 액세스할 수 있는 변수의 유효 범위(Scope)는 인라인 ST가 기술되는 래더도의 POU와 동일합니다.

- **인라인 ST의 제한 사양**

항목	내용
1회로당 인라인 ST 수	1개

6-5-3 ST 언어

ST(Structured Text) 언어는 IEC 61131-3에 규정된 공업 제어(주로 프로그래머블 컨트롤러)용 고급 언어입니다.

준비된 제어 구문, 연산자, 함수에 의해 래더도 언어에서는 기술하기 어려운 수치 연산 처리를 부분적으로 처리하는데 적합합니다.

ST 언어에는 다음 특징이 있습니다.

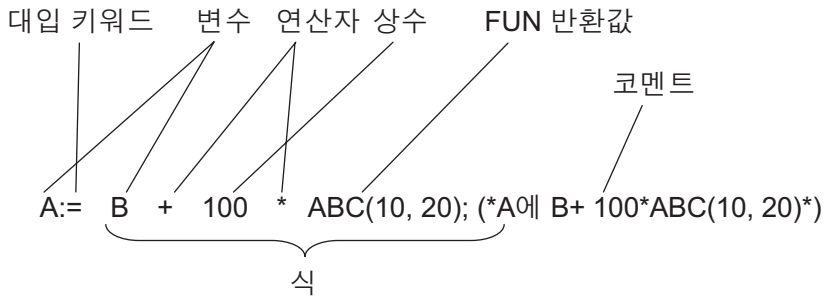
- 루프 구문 및 IF THEN ELSE 등의 제어 구문을 준비했습니다.
- C 언어 등의 고급 언어와 동일한 기술 및 한자를 포함한 문자 코멘트로 보기 쉬운 프로그램을 작성할 수 있습니다.

```
1
2       //TableNo 결정
3       FOR i:=0 TO ItemNum DO
4
5           IF (MinNo[i] <= ItemBox[i]) AND (ItemBox[i] <= MaxNo[i]) THEN   //범위 내
6               TableNo[i] := ItemBox[i];
7               RangeOK[i] := TRUE;
8
9           ELSIF (ItemBox[i] > MaxNo[i]) THEN                              //상한 초과
10              TableNo[i] := MaxNo[i];
11              RangeOK[i] := FALSE;
12
13          ELSE                                                            //하한 초과
14              TableNo[i] := MinNo[i];
15              RangeOK[i] := FALSE;
16
17          END_IF;
18
19      END_FOR;
```

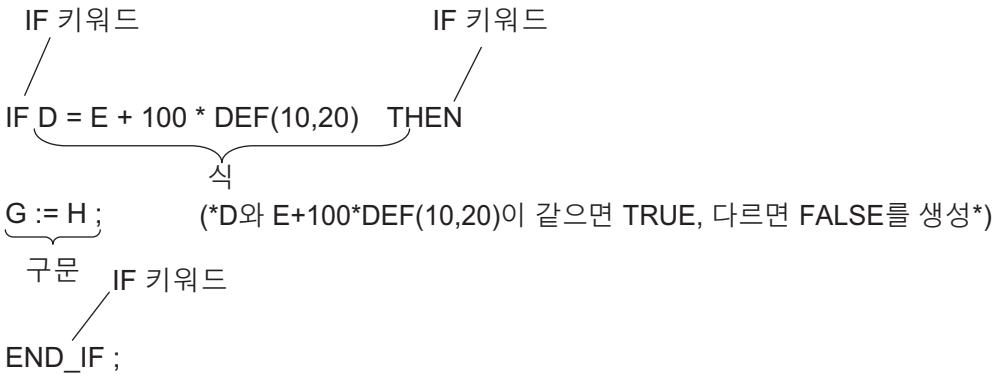
ST 언어의 구성

ST 언어의 프로그램은 하나 이상의 구문으로 구성됩니다. 하나의 구문이 하나의 처리를 표현합니다.
구문이 위에서 아래로 1행씩 순서대로 실행되면 ST 언어로 쓴 프로그램(ST 프로그램)의 처리가 진행됩니다.
구문은 키워드와 식으로 구성됩니다. 키워드는 대입 및 실행 제어 등을 나타내는 기호 및 문자열입니다.
식은 변수, 상수, FUN의 반환값 또는 이들과 연산자를 조합하여 숫자를 생성하는 기술입니다. 즉, 구문은 단독으로 완결되는 처리를 나타내고, 식은 생성한 숫자와 키워드를 조합하여 구문을 구성합니다.

대입 구문의 예:

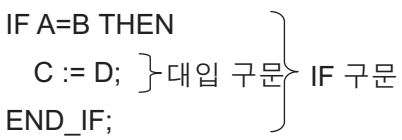


IF 구문의 예:



ST 언어의 표현

- **구문의 구분**
  - 구문은 반드시 반각 세미콜론(;)으로 종료합니다. 단순히 행바꿈한 것만으로는 구문의 종료가 되지 않습니다. 따라서 긴 구문을 여러 행에 걸쳐서 기술할 수 있습니다.
  - 하나의 구문에 하나의 반각 세미콜론(;)을 기술합니다. 예를 들어 아래에서는 IF 구문 하나 안에 대입 구문이 하나 존재하고, 각각의 구문의 마지막에 반각 세미콜론(;)을 기술합니다.



- **코멘트**
  - 프로그램의 내용을 사람이 이해하기 쉽게 하기 위해 프로그램 안에 임의의 코멘트를 기술할 수 있습니다.
  - 코멘트로 기술한 구문은 실행되지 않습니다.
  - 코멘트 기술에는 다음과 같은 2종류가 있습니다.

코멘트의 기술	예	비고
반각 괄호와 반각 별표 (*와 *)로 묶습니다.	(* 여러 행 코멘트 IF ErrCode = 3 THEN Value := 1000; END_IF; 여기까지 *)	여러 행의 코멘트를 기술할 수 있습니다. 코멘트의 중첩은 할 수 없습니다.
반각 슬래시 2개//로 시작하고, 행바꿈으로 종료합니다.	//코멘트 //A := SIN(X)^2 ;	1행씩만 코멘트할 수 있습니다.

- **스페이스(공백), 행바꿈, 탭**
  - 스페이스(공백), 행바꿈, 탭 또는 이들의 임의의 조합을 임의의 수, 임의의 위치에 기술해도 상관 없습니다.
    따라서 가독성을 고려하여 연산자/키워드와 식 사이를 행바꿈하거나 구문 앞에 공백 및 탭을 입력할 수 있습니다.
  - 연산자/키워드와 변수명 사이에는 스페이스(공백), 행바꿈, 탭을 반드시 입력하십시오.

예) ■는 스페이스(공백), 행바꿈, 탭 등을 입력해야 하는 장소입니다.

```
IF■A>0■THEN■X:=10;
ELSE
    X:=0;
END_IF;
```

- **소문자/대문자, 반각/전각**
  - 연산자/키워드 및 변수명은 소문자, 대문자를 구별하지 않습니다(어느 경우라도 상관 없습니다).
  - 연산자/키워드는 반각으로 입력하십시오. 전각으로 입력하면 문법 에러가 됩니다.

- **변수명의 금칙 문자**

변수명의 제약은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.

- **문자열**

문자열의 제약은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.

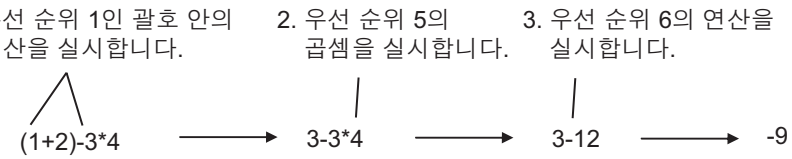
ST 언어의 키워드 연산자

- **구문의 키워드**

키워드	의미	예
:=	대입	d := 10;
	FB/FUN의 호출	FBname(para1 := 10, para2 := 20); 자세한 내용은 'FB 호출(6-121페이지)'을 참조
RETURN	반환	
IF	선택	IF d < e THEN f := 1; ELSIF d = e THEN f :=2; ELSE f := 3; END_IF;
CASE	다방향 분기	CASE f OF 1: g :=11; 2: g :=12; ELSE g :=0; END_CASE;
FOR	반복(1)	FOR i := 100 TO 1 BY -1 DO Val[i] := i; END_FOR;
WHILE	반복(2)	WHILE Val < MaxVal DO Val := Val + 1; END_WHILE;
REPEAT	반복(3)	REPEAT Val := Val + 1; UNTIL(Val > 4) END_REPEAT;
EXIT	루프의 중단	FOR i := 1 TO 100 DO FOR j := 1 TO 10 DO IF Val[i, j]>100 THEN EXIT; END_IF; END_FOR; END_FOR;
;	빈 구문	Val[i] := i ;; (*빈 구문*) WHILE(Var <>0) DO ;; (*빈 구문*) END_WHILE;
(*텍스트*)	코멘트	(*여러 행 코멘트 IF MyFun (ErrorCode) = 3 THEN ReturnValue := GetDetail(); END_IF; 여기까지*)
//텍스트	코멘트	A := SIN(X) ^ 2 + COS (Y) ^2 + 10; // A := SIN(X) ^ 2 + COS (Y) ^2 + 5;

● **연산자**

다음 표는 연산자의 종류와 우선 순위입니다.
하나의 식에 우선 순위가 다른 연산자가 혼재된 경우에 우선 순위가 높은 연산을 먼저 실행합니다.
하나의 식에는 최대 64개의 연산자를 기술할 수 있습니다.
예) X:=(1+2)-3*4;인 경우에 변수 X에는 -9가 대입됩니다.



연산	연산자	기술 예와 평가값	우선 순위
괄호	()	(1+2)*(3+4) 값 21	1
FB, FUN의 호출		FUN1(FUN2(Var2A, Var2B), Var1B) 계층적으로 FB, FUN을 호출한 경우에 아래 계층이 먼저 실행됩니다. 위의 예에서는 FUN2가 실행된 후 FUN1이 실행됩니다.	2
부호	+, -	+100 -100	3
부정	NOT	NOT TRUE 값 FALSE	
거듭제곱	**	-2**2 값 4 거듭제곱 연산자보다 - 부호 쪽이 우선 순위가 먼저입니다. 따라서 위의 예에서는 -2**2=(-2)**2가 되므로 값은 4입니다. 2**3**2 값 64 거듭제곱 연산자가 여러 개 중복된 경우에 좌측에서부터 연산을 실행합니다. 위의 예에서는 2**3**2=(2**3)**2가 되므로 값은 64입니다.	4
곱셈	*	100*200 값 20000	5
나눗셈	/	100/200 값 0.5	
나머지	MOD	10 MOD 7 값 3 -17 MOD 6 값 -5 -17 MOD (-6) 값 -5 17 MOD 6 값 5 17 MOD (-6) 값 5	
덧셈	+	100+200 값 300	6
뺄셈	−	100-200 값 -100	

연산	연산자	기술 예와 평가값	우선 순위
비교	<, >, < =, > =	100<200 비교 결과가 올바르면 값은 TRUE, 올바르지 않으면 값은 FALSE가 됩니다. 위의 예 100<200은 올바르므로 값은 TRUE가 됩니다	7
일치	=	100=200 2개의 값이 일치하면 값은 TRUE, 일치하지 않으면 값은 FALSE가 됩니다. 위의 예 100=200은 일치하지 않으므로 값은 FALSE가 됩니다.	8
불일치	< >	100<>200 2개의 값이 불일치이면 값은 TRUE, 불일치가 아니면 값은 FALSE가 됩니다. 위의 예 100<>200은 불일치이므로 값은 TRUE가 됩니다	
논리곱	AND,&	1비트의 논리곱을 전체 비트 분량 실행합니다. 1비트의 논리곱의 결과는 다음과 같습니다. 0 AND 0 = 0 0 AND 1 = 0 1 AND 0 = 0 1 AND 1 = 1 0101 AND 1100 값 0100	9
배타적 논리합	XOR	1비트의 배타적 논리합을 전체 비트 분량 실행합니다. 1비트의 배타적 논리합의 결과는 다음과 같습니다. 0 XOR 0 = 0 0 XOR 1 = 1 1 XOR 0 = 1 1 XOR 1 = 0 0101 XOR 1100 값 1001	10
논리합	OR	1비트의 논리합을 전체 비트 분량 실행합니다. 1비트의 논리합의 결과는 다음과 같습니다. 0 OR 0 = 0 0 OR 1 = 1 1 OR 0 = 1 1 OR 1 = 1 0101 OR 1100 값 1101	11

사용상의 주의 사항

동일한 펑션을 계층적으로 기술하면 의도한 동작을 하지 않는 경우가 있습니다. 반드시 다음 예와 같이 구문을 나눠서 기술하십시오.

예) × out := MyFunc(In1:=x1, In2:=MyFunc(In1:=x2, In2:=x3));

○ tmp := MyFunc(In1:=x2, In2:=x3);
out := MyFunc(In1:=x1, In2:=tmp);

사용상의 주의 사항

연산자의 우선 순위는 규격 및 제조업체의 차이에 따라 달라지는 경우가 있습니다. 특히 거듭제곱 연산자의 우선 순위에는 주의해야 합니다. 의도한 순서로 연산을 실행하기 위해 괄호를 사용하여 각 연산의 우선 순위를 명시하는 것을 권장합니다.
예) X:= -2**3**4;는 X:=((-2)**3)**4; 라고 기술합니다.

참고

연산은 데이터 형식에 따라 실행됩니다.
예를 들어 정수 형식 상호 간의 연산은 정수 형식으로 실행됩니다. 따라서 INT 형식의 변수A = 3, B = 2에서 A/B의 연산을 실행하면 연산 결과는 1.5가 아니라 소수점 이하를 버려서 1이 됩니다. 따라서 (A/B)*2의 연산 결과는 3이 아니라 2가 됩니다.

- **각 연산자의 피연산자에 사용 가능한 데이터 형식**

하나의 연산자에 사용되는 피연산자가 모두 동일한 데이터 형식인 경우에 다음과 같은 데이터 형식을 피연산자에 설정할 수 있습니다.
단, 연산자에 다른 데이터 형식의 피연산자가 설정된 경우에 암시적 캐스트가 발생합니다.
암시적 캐스트에 대해서는 🕮'암시적 캐스트(6-127페이지)'를 참조하십시오.

데이터 형식	대입 연산자	인수 설정 연산자	수치 연산자	나머지 연산자	거듭제곱 연산자	비교 연산자	등가 연산자	논리 연산	양/음 기호
	:=	:= =>	+ - * /	MOD	**	< <= >= >	= <>	NOT AND & OR XOR	+ -
BOOL 형식	○	○	×	×	×	×	○	○	×
비트 열 형식	○	○	×	×	×	×	○	○	×
정수 형식	○	○	○	○	○*1	○	○	×	○
실수 형식	○	○	○	×	○	○	○	×	○
지속 시간 형식	○	○	×	×	×	×	×	×	○
날짜 형식	○	○	×	×	×	×	×	×	×
시각 형식	○	○	×	×	×	×	×	×	×
날짜 시각	○	○	×	×	×	×	×	×	×
문자열 형식	○	○	×	×	×	×*2	×*2	×	×
열거 형식	○	○	×	×	×	×	○	×	×
구조체 부모	○	○	×	×	×	×	×	×	×
배열 부모	○	○	×	×	×	×	×	×	×

○: 가능
×: 빌드 시 에러가 됩니다

*1. NX701 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 Sysmac Studio의 프로젝트의 장치 선택 화면에 있는 [버전]에서 [Ver.1.15] 이하를 선택한 경우에 정수 형식의 변수를 피연산자에 설정해도 실수 형식으로 연산됩니다. 연산 결과에 반올림 오차가 포함될 때는 소수점 이하를 버리므로 의도한 값이 되지 않을 수 있습니다. 소수점 이하를 반올림하는 경우에는 EXPT 명령과TO_**(정수 변환 그룹) 명령을 조합해서 사용하십시오.
예) TO_INT(EXPT(X,Y))

*2. 문자열 형식 변수의 비교는 연산자가 아니라 명령(EQascii 등)으로 실행하십시오.

ST 언어의 구문

- **대입**

개요:

좌변(변수)에 우변(식의 값)을 대입합니다.

예약어:

:=
콜론(:)과 등호(=)의 조합

구문의 구조:

```
<변수>:=<식, 변수 또는 상수>;
```

용도:

변수에 값을 대입할 때 사용합니다.
예를 들어 초기값 설정, 계산 결과 저장 등에 사용합니다.

설명:

<식의 값>을 <변수>에 대입(저장)합니다.

예:

예 1) 식 X+1의 결과를 변수 A에 대입

```
A:=X+1;
```

예 2) 변수 B의 값을 변수 A에 대입

```
A:=B;
```

예 3) 상수 10을 변수 A에 대입

```
A:=10;
```

유의 사항:

- 대입원의 데이터 형식은 대입 대상 변수의 데이터 형식과 일치하거나 암시적 캐스트에서 허가된 데이터 형식의 조합이어야 합니다. 그렇지 않은 경우에는 빌드 시 에러가 됩니다.
- 대입원이 문자열인 경우에 대입 대상인 STRING 형식 변수의 크기는 대입원의 문자열 크기보다 크게 하십시오. 그렇지 않은 경우에는 에러가 됩니다.
- 변수가 STRING 형식인 경우에 (좌변의 변수 크기)>(우변의 변수에 저장되는 문자열에 크기)이면 대입할 수 있습니다.
  예)
  다음과 같은 경우에는 대입할 수 있습니다.
- 변수 테이블:

변수명	데이터 형식	크기
Var1	STRING	10
Var2	STRING	20

- 사용자 프로그램:
  Var2 :='ABC';
  Var1 := Var2;

공용체 형식 변수 전체에 대해서는 대입할 수 없습니다. 멤버 단위로 대입하십시오.

- **RETURN**

개요:

ST를 사용하는 프로그램의 종류에 따라 다음과 같습니다.

<ST 프로그램>

도중에 ST 프로그램을 종료하고 다음 프로그램을 실행합니다.

<FB의 인스턴스 내 FUN에서 사용하는 ST>

도중에 FB, FUN을 강제 종료하고 FB, FUN의 호출원의 다음 명령을 실행합니다.

<인라인 ST>

RETURN 구문이 존재하는 인라인 ST를 포함해서 POU를 종료합니다.

예약어:

RETURN

구문의 구조:

```
RETURN;
```

용도:

실행 중인 프로그램, FB 또는 FUN을 강제 종료할 때 사용합니다.

- **IF(하나의 조건)**

개요:

어떤 조건이 성립했을 때, 지정한 구문을 실행합니다. 성립되지 않았을 때, 다른 구문을 실행합니다.
조건 성립과 성립하지 않는 경우를 다음과 같이 표현합니다.

TRUE: 조건이 성립하는 경우
FALSE: 조건이 성립하지 않는 경우

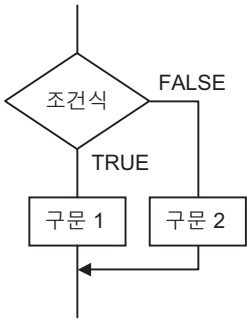
예약어:

IF, THEN, (ELSE), END_IF
주의: ELSE는 생략 가능

구문의 구조:

```
IF <조건식>THEN
  <구문 1>;
ELSE
  <구문 2>;
END_IF;
```

처리 플로우도:



용도:

어떤 하나의 조건(조건식)이 성립했는지 아닌지에 따라 경우를 나눌 때 사용합니다.

설명:

<조건식> = TRUE인 경우에 <구문 1>을 실행합니다.
<조건식> = FALSE인 경우에 <구문 2>를 실행합니다.

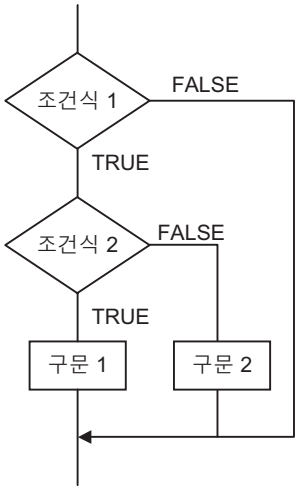
유의 사항:

- IF는 반드시 END_IF와 세트로 사용해야 합니다.
- <조건식>에는 평가 결과가 TRUE 또는 FALSE인 식(예를 들어 IF A>10), BOOL 형식 변수(예를 들어 IF A)를 기술할 수 있습니다.
- <구문 1>, <구문 2>에 여러 행에 걸쳐 프로그램을 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.

예) <구문 1>에 다시 IF 구문을 넣은 경우

```
IF <조건식 1>THEN
    IF <조건식 2>THEN
                <구문 1>;
    ELSE
                <구문 2>;
                END_IF;
END_IF;
```

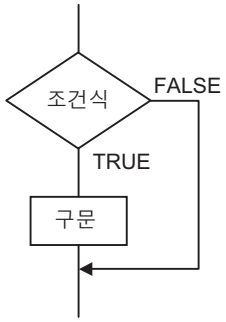
처리 플로우도:



또한, ELSE는 위와 같이 직전의 THEN에 대응합니다.

- <구문 1>, <구문 2> 중에는 모두 하나 이상의 구문을 실행할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- ELSE 구문은 생략할 수 있습니다. 생략한 경우에 <조건식>=FALSE이면 아무 것도 실행하지 않습니다.

처리 플로우도:



예:

예 1) 변수 A>0이 TRUE인 경우에 변수 X에 숫자 10을 대입합니다. 변수 A>0이 FALSE인 경우에 변수 X에 숫자 0을 대입합니다.

```
IF A>0 THEN
    X:=10;
ELSE
    X:=0;
END_IF;
```

예 2) 변수 A>0 변수 B>1이 TRUE인 경우에 변수 X에 숫자 10을 대입하고, 변수 Y에 숫자 20을 대입합니다. 변수 A>0과 변수 B>1이 FALSE인 경우에 변수 X에 숫자 0을 대입하고, 변수 Y에 숫자 0을 대입합니다.

```
IF A>0 AND B>1 THEN
    X:=10;Y:=20;
ELSE
    X:=0;Y:=0;
END_IF;
```

예 3) BOOL 형식의 변수 A = TRUE인 경우에 변수 X에 숫자 10을 대입합니다. 변수 A = FALSE가 FALSE인 경우에 변수 X에 숫자 0을 대입합니다.

```
IF A THEN X:=10;
ELSE X:=0;
END_IF;
```

- **IF(복수의 조건)**

개요:

어떤 조건이 성립했을 때, 지정한 구문을 실행합니다. 성립하지 않은 경우에 다시 다른 조건이 성립하면 다른 구문을 실행합니다. 모두 성립하지 않은 경우에 다시 다른 구문을 실행합니다.
조건 성립과 성립하지 않는 경우를 다음과 같이 표현합니다.

TRUE: 조건이 성립하는 경우
FALSE: 조건이 성립하지 않는 경우

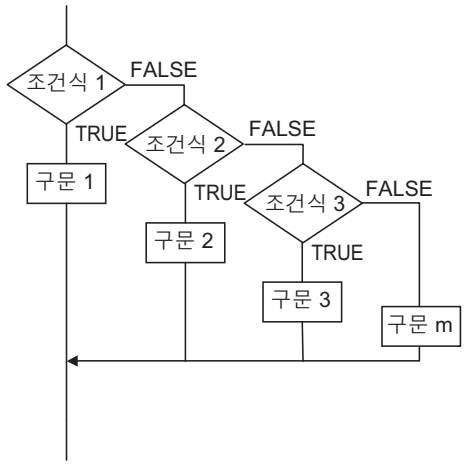
예약어:

IF, THEN, ELSIF, (ELSE), END_IF
주의: ELSE는 생략 가능

구문의 구조:

```
IF <조건식1> THEN <구문 1>;
    ELSIF <조건식2> THEN <구문 2>;
    ELSIF <조건식3> THEN <구문 3>;
            .
            .
            .
    ELSIF <조건식n>THEN <구문 n>;
ELSE<구문 m>;
END_IF;
```

처리 플로우도:



용도:

어떤 복수의 조건(조건식)이 성립했는지 아닌지에 따라 경우를 나눌 때 사용합니다.

설명:

<조건식 1> = TRUE인 경우에 <구문 1>을 실행합니다.
<조건식 1> = FALSE인 경우에
<조건식 2> = TRUE이면 <구문 2>를 실행합니다.
<조건식 2> = FALSE인 경우에
<조건식 3> = TRUE이면 <구문 3>을 실행합니다.
.
.
.
<조건식 n> = TRUE이면 <구문 n>을 실행합니다.
어떤 조건에도 합치되지 않은 경우에 <구문 m>을 실행합니다.

유의 사항:

- IF는 반드시 END_IF와 세트로 사용해야 합니다.
- <조건식 □>에는 평가 결과가 TRUE 또는 FALSE인 식을 기술합니다. 예) IF(A>10)
  또한, <조건식 □>에는 식이 아니라 BOOL 형식의 변수(FUN의 BOOL 형식 반환값을 포함)만 지정할 수도 있습니다. 그 경우에 변수의 값이 TRUE = 평가 결과 TRUE, FALSE = 평가 결과 FALSE입니다.
- <구문 □>에 여러 행에 걸쳐서 프로그램을 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- ELSE 구문은 생략할 수 있습니다. 생략한 경우에 어떤 조건에도 합치되지 않을 때는 아무 것도 실행하지 않습니다.

예:
변수 A>0이 TRUE인 경우에 변수 X에 숫자 10을 대입합니다.
변수 A>0이 FALSE인 경우에 변수 B = 1이 TRUE이면 변수 X에 숫자 1을 대입합니다.
변수 A>0이 FALSE인 경우에 변수 B = 2가 TRUE이면 변수 X에 숫자 2를 대입합니다.
어떤 조건에도 합치되지 않을 때, 변수 X에 숫자 0을 대입합니다.

```
IF A>0 THEN X:=10;
    ELSIF B=1 THEN X:=1;
    ELSIF B=2 THEN X:=2;
ELSE X:=0;
END_IF;
```

- **CASE**

개요:

정수식의 값에 일치하는 정수의 선택값을 가진 구문을 실행합니다.

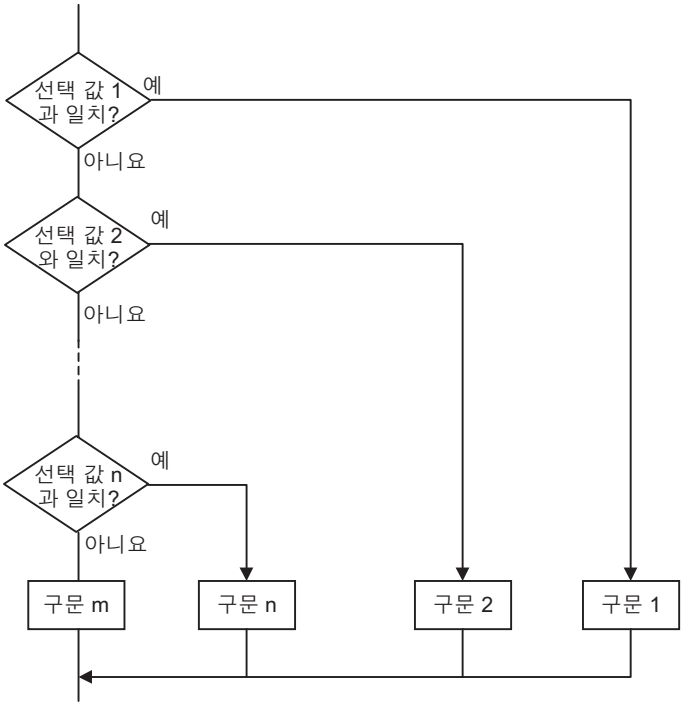
예약:

CASE

구문의 구조:

```
CASE <정수식> OF
    <정수식의 값 1>:<구문 1>;
    <정수식의 값 2>:<구문 2>;
            .
            .
            .
    <정수식의 값 n>:<구문 n>;
ELSE<구문 m>;
END_CASE;
```

처리 플로우도:



용도:

어떤 정수의 값에 의해 경우 나누기를 할 때 사용합니다.

설명:

<정수식>이 <정수식의 값 n>과 일치했을 때 <구문 n>을 실행합니다.
<정수식>이 어떤 <정수식의 값 n>도 일치되지 않았을 때 <구문 m>을 실행합니다.

유의 사항:

- CASE는 반드시 END_CASE와 세트로 사용해야 합니다.
- <정수식>에는
  · 정수 형식 및 열거 형식의 변수 (예: abc)
  · 정수식 (예: abc+def)
  · 정수값을 반환값으로 하는 FUN (예: xyz())
  를 기술합니다.
- <구문 □>에 여러 행에 걸쳐서 프로그램을 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- <정수식의 값 n>으로 복수의 정수의 OR 논리를 지정할 때는 숫자를 쉼표(,)로 구분합니다. 연속하는 정수를 지정할 때는 시작 정수와 종료 정수 사이를 마침표 2개(..)로 구분합니다.
  예 1) 특정 정수값에 조건을 지정할 수 있으며, 복수의 정수값에 동일한 조건을 지정할 수도 있습니다.

```
CASE A OF
    1: X:=1;    ---------------  숫자 A가 1일 때, 변수 X에 수치 1을 대입
    2: X:=2;    ---------------  숫자 A가 2일 때, 변수 X에 수치 2를 대입
    3: X:=3;    ---------------  숫자 A가 3일 때, 변수 X에 수치 3을 대입
ELSE
    X:=0;       ---------------  어느 것에도 일치하지 않을 때에는 변수 X에 0을 대입
END_CASE;
```

```
CASE A OF
    1: X:=1;    ---------------  숫자 A가 1일 때, 변수 X에 수치 1을 대입
    2,5: X:=2;  ---------------  변수 A가 2 또는 5일 때, 변수 X에 수치 2를 대입
    6..10: X:=3; --------------  변수 A가 6~10 중 하나 일 때,변수 X에 수치 3을 대입
    11,12,15..20: X:=4; -------  변수 A가 11, 12 또는 15~20 일 때, 변수 X에 수치 4를 대입
ELSE
    X:=0;       ---------------  어느 것에도 일치하지 않을 때에는 변수 X에 0을 대입
END_CASE;
```

예 2) <정수식>에는 정수 형식의 변수·식·FUN의 반환값 및 열거 형식의 변수·FUN의 반환값을 기술할 수 있습니다. 예시는 다음과 같습니다.

- 정수·열거 형식 변수의 예

```
CASE ColorVar OF
    RED:
        X := 0;
    BLUE:
        X := 1;
    ELSE
        X := 2;
END_CASE;
```

- 정수식의 예

```
CASE (a1 + a2) OF
    0:
        X := 0;
    1:
        X := 1;
    ELSE
        X := 2;
END_CASE;
```

- 정수·열거 형식 FUN의 반환값의 예

```
CASE FUN( ) OF
    0:  ---------------------  FUN()의 반환값에 따라 분기시킴
        X := 10;
    1:
        X := 11;
    ELSE
        X := 12;
END_CASE;
```

CASE에 사용 가능한 데이터 형식

분류	데이터 형식		<정수식>
기본 데이터 형식	정수 형식		○
	부울 형식, 비트 열 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		×
데이터 형식 지정	배열 지정	배열 전체	×
		요소	○ 정수 형식,열거 형식만 가능
파생 데이터 형식	구조체 형식	구조체 전체	×
		멤버	○ 정수 형식,열거 형식만 가능
	공용체 형식	공용체 전체	×
		멤버	○ 정수 형식,열거 형식만 가능
	열거 형식		○

- **FOR**

개요:

변수(이후, 반복 변수)가 어떤 값에서 어떤 값이 될 때까지 어떤 구문을 반복해서 실행합니다.
조건 성립과 성립하지 않는 경우를 다음과 같이 표현합니다.

TRUE: 조건이 성립하는 경우
FALSE: 조건이 성립하지 않는 경우

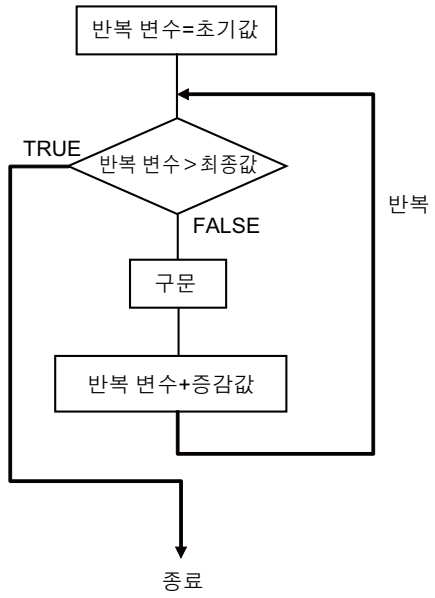
예약어:

FOR, TO, (BY), DO, END_FOR
주의: BY는 생략 가능

구문의 구조:

```
FOR <반복 변수>:= <초기값> TO <최종값> BY <증감값> DO
    <구문>;
END_FOR;
```

처리 플로우도:



용도:

반복 횟수를 미리 알고 있는 경우에 사용합니다.
특히 어떤 반복 변수의 값에 따라 배열 지정 변수의 각 요소를 전환해서 지정하는 경우에 유효합니다.

설명:

<초기값>, <최종값>, <증감값>을 평가하여 결정합니다.
<반복 변수>가 <초기값>인 경우 <구문>을 실행합니다.
실행 후 <반복 변수>에 <증감값>으로 주어진 값을 더하고, <반복 변수><<최종값>의 값이면 <구문>을 실행합니다.
실행 후 <반복 변수>에 <증감값>으로 주어진 값을 더하고, <반복 변수><<최종값>의 값이면 <구문>을 실행합니다.
이를 반복합니다.
<반복 변수>><최종값>이 되면 종료합니다.
또한, <증감값>이 음수인 경우에는 크기 비교 기호의 방향이 반대입니다.

유의 사항:

- 반복 변수가 부호가 있는 정수 형식인 경우 <증감값>에 마이너스 값을 지정할 수 있습니다.
- FOR는 반드시 END_FOR와 세트로 사용합니다.
- 반복 변수에서는 최종값으로 처리를 실행한 후 (최종값+증감값)이 되고, 반복 처리를 종료합니다.
  예) 다음과 같은 ST 구문에서는 FOR 구문 종료 시 i의 값은 101이 됩니다.

```
FOR i:=0 TO 100 DO
    X[i]:=0;
END_FOR;
//이 시점에서 i는 101가 됨
```

- FOR 구문의 내측에서 반복 변수를 직접 변경하는 처리는 기술하지 마십시오. 예기치 못한 동작을 할 수 있습니다.
  예)

```
FOR i:=0 TO 100 BY 1 DO
    X[i]:=0;
    i:=i+INT#5;
END_FOR;
```

- <구문>에 여러 행에 걸쳐서 프로그램을 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- BY<증감값>는 생략할 수 있습니다. 생략한 경우에 BY는 1로 실행됩니다.
- <초기값>, <최종값>, <증감값>에는 정수 형식(SINT, INT, DINT, LINT, USINT, UINT, UDINT 또는 ULINT)의 변수 또는 정수값을 지정할 수 있습니다. 또한, FUN의 정수 형식 반환값도 지정할 수 있습니다.
  예 1) 반복 변수 n = 0에서 5개 간격으로 50까지 반복해서 배열 지정 변수 SP[n]에 100을 대입합니다.

```
FOR n := 0 TO 50 BY 5 DO
    SP[n] := 100;
END_FOR;
```

예 2) 배열 지정 변수 DATA[n]의 요소의 DATA[1]에서 DATA[50]의 합계값을 산출하여 변수 SUM에 대입합니다.

```
IF a THEN
    FOR n := 0 TO 50 BY 1 DO
        DATA[n]:= 1 ;
    END_FOR;

    FOR n := 0 TO 50 BY 1 DO
        SUM:= SUM + DATA[n] ;
    END_FOR;

    a:=FALSE;
END_IF;
```

예 3) 배열 지정 변수 DATA[n]의 요소의 DATA[1]에서 DATA[50] 중에서 최대값과 최소값을 검색합니다. 최대값을 변수 MAX에 대입하고, 최소값을 변수 MIN에 대입합니다. 또한, DATA[n]의 값은 0~1000의 범위 내로 합니다.

```
MAX :=0;
MIN :=1000;
FOR n :=1 TO 50 BY 1 DO
    IF DATA[n] > MAX THEN
        MAX :=DATA[n];
    END_IF;
    IF DATA[n] < MAX THEN
        MIN :=DATA[n];
    END_IF;
END_FOR;
```

- <반복 변수>에 <초기값>에서 <증감값>을 더하고, <최종값>을 초과할 때까지 FOR 구문을 포함하는 프로그램 실행 시간이 태스크 주기 설정 시간을 초과한 경우에 태스크 주기 초과가 발생합니다.
  - 로직 상 <반복 변수>가 <최종값>에 도달하지 않은 경우

  예)

```
FOR i := 0 TO 100 BY 1 DO
    intArray[i] := i;
    i := INT#50;      ---------- 무한 루프가 되어, 태스크 주기 초과가 발생합니다.
END_FOR;
```

  예)

```
FOR i := 0 TO 100 BY 0 DO      -------- 무한 루프가 되어, 태스크 주기 초과가 발생합니다.
    ;
END_FOR;
```

- <반복 변수>가 <최종값>을 초과한 시점에 오버플로/언더플로가 발생하는 경우

  예)
  변수 i가 USINT 형식인 경우에 최종값 256을 표현할 수 있습니다. 0으로 간주하여 무한 루프가 됩니다.

```
FOR i := 0 TO 254 BY 2 DO
    INTArray[i] := i;
END_FOR;
```

버전 관련 정보

Sysmac Studio Ver.1.08 이상인 경우에 <최종값> 및 <증감값>에는 연산식을 기술할 수 있습니다. 단, <최종값> 및 <증감값>이 평가되는 것은 FOR 루프 내의 연산이 실행되기 전입니다. FOR 루프 내의 연산이 시작된 후 <최종값> 및 <증감값>이 변화하지는 않습니다.
다음 그림의 경우 <최종값>은 10, <증감값>은 3이 됩니다. FOR 루프 내의 연산이 시작되고, 변수 A 및 변수 C의 값이 변해도 <최종값>은 10 그대로, <증감값>은 3 그대로입니다.

```
A := INT#1;
B := INT#2;
C := INT#10;

FOR i := 0 TO C BY A+B DO
    INTArray[i] := i;
    A := B + i;
    C := C + i;
END_FOR;
```

Sysmac Studio Ver.1.07 이하에서 <최종값> 및 <증감값>에 연산식을 기술하면 빌드 시 에러가 되므로 주의하십시오.

FOR에 사용 가능한 데이터 형식

분류	데이터 형식		<반복 변수>, <초기값>, <최종값>, <증감값>[*1]
기본 데이터 형식	부울 형식, 비트 열 형식, 실수 형식, 지속 시간 형식, 날짜 형식, 시각 형식, 날짜 시각 형식, 문자열 형식		×
	정수 형식		○
데이터 형식 지정	배열 지정	배열 전체	×
		요소	○ 정수 형식,열거 형식만 가능[*2]
파생 데이터 형식	구조체 형식	구조체 전체	×
		멤버	○ 정수 형식,열거 형식만 가능[*2]
	공용체 형식	공용체 전체	×
		멤버	×
	열거 형식		○[*2]

*1. <반복 변수>, <최종값>, <증감값>은 3종류 모두 데이터 형식을 일치시키십시오. 데이터 형식이 일치하지 않으면 Sysmac Studio의 프로그램 빌드 시 에러가 됩니다.
*2. <반복 변수>, <최종값>, <증감값>에는 열거 형식을 사용할 수 없습니다.

- **WHILE**

개요:

어떤 조건식이 TRUE인 동안 지정한 구문을 반복해서 실행합니다.

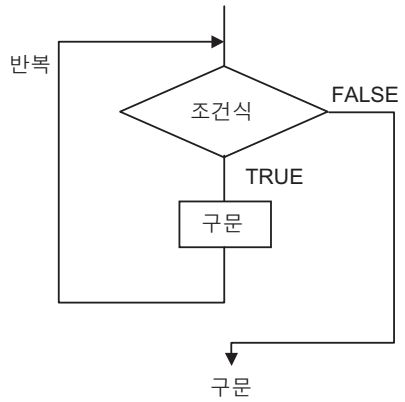
예약어:

WHILE, DO, END_WHILE

구문의 구조:

```
WHILE <조건식>DO
    <구문>;
END_WHILE;
```

처리 플로우도:



용도:

반복 횟수를 모르는(조건에 따라 몇 번이 될지 모르는) 경우에 어떤 조건식이 성립하는 동안 어떤 처리를 반복할 때 사용합니다.
조건식이 성립할 때만 어떤 처리를 실행할 때 사용합니다(이전 판정의 반복).

설명:

<구문> 실행 전에 <조건식>이 평가됩니다.
<조건식>이 TRUE인 경우에 <구문>을 실행합니다. 그 후, <조건식>이 다시 평가됩니다. 이를 반복해서 실행합니다.
<조건식>이 FALSE인 경우에 <구문>을 실행하지 않고 <조건식>의 평가를 종료합니다.

유의 사항:

- WHILE은 반드시 END_WHILE과 세트로 사용해야 합니다.
- <구문> 실행 전에 <조건식>이 FALSE인 경우에 <구문>을 실행하지 않고 종료합니다.
- <구문 1>, <구문 2>에 여러 행에 걸쳐 프로그램을 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- <구문> 중에서 하나 이상의 구문을 실행할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- <조건식>에는 식이 아니라 BOOL 형식의 변수(FUN의 BOOL 형식 반환값 포함)를 지정할 수도 있습니다.

  예 1) 7의 배수이고 1000을 초과하는 수를 산출하여 변수 A에 대입합니다.

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

  예 2) 변수 X＜3000인 동안 X의 값을 2배로 하고, 그 값을 배열 지정 변수 DATA[1]에 대입합니다. 다시 X의 값을 2배로 하고, 그 값을 배열 지정 변수 DATA[2]에 대입합니다. 이를 반복합니다.

```
n := 1;
X := 1;
WHILE X < 3000 DO
    X:= X*INT#10#2;
    DATA[n]:= X;
    n := n+INT#1;
END_WHILE;
```

- 올바른 <조건식>을 쓰지 못한 경우에 프로그램 실행 시간이 연장되고, 태스크 주기 초과를 발생시킬 수 있는 요인이 됩니다.

  예)

```
boolVar := TRUE;
WHILE boolVar DO
intVar := intVar + INT#1;
END_WHILE;
```

- **REPEAT**

조건 성립과 성립하지 않는 경우를 다음과 같이 표현합니다.
TRUE: 조건이 성립하는 경우
FALSE: 조건이 성립하지 않는 경우

개요:

어떤 구문을 실행하여 어떤 조건식이 TRUE이 될 때까지 그 구문을 반복합니다.

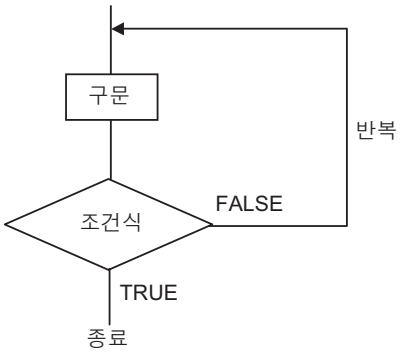
예약어:

REPEAT, UNTIL, END_ REPEAT

구문의 구조:

```
REPEAT
    <구문>;
UNTIL <조건식>
END_REPEAT;
```

처리 플로우도:



용도:

반복 횟수를 모르는(조건에 따라 몇 번이 될지 모르는) 경우에 어떤 처리 후의 조건식이 성립하는 동안 해당 처리를 반복할 때 사용합니다.
어떤 처리를 실행한 결과에 따라 반복할지 여부를 판단할 때 사용합니다(이후 판정의 반복).

설명:

<구문>이 처음에 무조건 실행됩니다. 그 후, <조건식>이 평가됩니다.
<조건식>이 FALSE인 경우에 <구문>을 다시 실행합니다.
<조건식>이 TRUE인 경우에 <구문>을 실행하지 않고 종료합니다.

유의 사항:

- REPEAT는 반드시 END_REPEAT와 세트로 사용해야 합니다.
- <구문>의 실행 전에 <조건식>이 TRUE인 경우에도 <구문>이 실행됩니다.
  즉, <구문>은 적어도 1회는 실행됩니다.
- <구문>에 구문의 키워드로 정하는 구문을 여러 행에 걸쳐서 기술할 수 있습니다. 그 경우에 구문과 구문 사이를 ;으로 구분합니다.
- <조건식>에는 식이 아니라 BOOL 형식의 변수(FUN의 BOOL 형식 반환값 포함)를 지정할 수도 있습니다.

예) 1에서 10까지의 숫자를 더하고, 변수 TOTAL에 대입합니다.

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

- 올바른 <조건식>을 쓰지 못한 경우에 프로그램 실행 시간이 연장되고, 태스크 주기 초과를 발생시킬 수 있는 요인이 됩니다.

  예)

```
intVar := INT#1;
REPEAT
    intVar := intVar + INT#1;
UNTIL intVar = INT#0
END_REPEAT;
```

- **EXIT**

개요:

반복 구분(FOR 구문, WHILE 구문, REPEAT 구문) 중에서만 사용하고, 반복 구문을 종료시킵니다.
IF 구문 중에서 사용하면 어떤 조건이 성립했을 때, 반복 구문을 도중에 빼낼 수 있습니다.

예약어:

EXIT

구문의 구조(예: IF 구문 중에서 사용):

```
FOR(WHILE, REPEAT) 구문
            .
            .
            .
IF<조건식>THEN EXIT;
END_IF;
            .
            .
            .
END_FOR(WHILE, REPEAT);
```

용도:

종료 조건이 성립하기 전에 반복해서 처리를 강제적으로 종료할 때 사용합니다.

설명(예: IF 구문 중에서 사용):

<조건식>이 TRUE인 경우에 반복 구문(FOR 구문, WHILE 구문, REPEAT 구문)을 강제 종료하고, EXIT 이후에 기술된 구문은 실행되지 않습니다.
(주의) 1 <조건식>에는 식이 아니라 BOOL 형식의 변수만 지정할 수도 있습니다.
2 <구문>의 실행 전에 <조건식>이 TRUE인 경우에도 <구문>이 실행됩니다.

예:

예 1) 변수 n = 1에서 1개 간격으로 50까지 반복해서 배열 지정 변수 DATA[n]에 n을 가산합니다. 단, DATA[n]이 100을 초과하면 종료합니다.

```
IF A THEN
    DATA[3] :=98;
    FOR n := 1  TO 50 BY 1 DO
        DATA[n] := DATA[n] + n;
        IF DATA[n] > 100 THEN EXIT;
        END_IF;
    END_FOR;
    A :=FALSE;
END_IF;
```

- **FB 호출**

개요:

FB를 호출합니다.

예약어: 없음

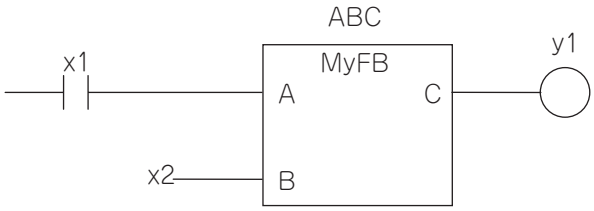
구문의 구조:

FB의 인스턴스명 다음에 소괄호 안에 인수 지정(지정한 변수의 값을 호출 대상 FB의 입력 변수에 전달하는 지정)과 반환값 지정(호출 대상 FB의 출력 변수 값을 지정한 변수에 수신하는 지정)을 기술합니다.
다음과 같은 ① 또는 ② 중 어느 한 기술 방식이 가능합니다.
사용자 프로그램의 가독성을 고려하여 ①을 권장합니다.

기술 방식 ①:

호출하는 FB의 변수명과 호출원의 파라메터명을 모두 기술하는 경우
ABC(A:=x1, B:=x2, C=>y1);
ABC : FB의 인스턴스명
A, B : 호출하는 FB의 입력 변수명 또는 입출력 변수명
x1, x2 : 호출원의 입력 파라메터 또는 입출력 파라메터(상수 포함)
C : 호출하는 FB의 출력 변수명
y1 : 호출원의 출력 파라메터

- 래더도 언어로의 표현

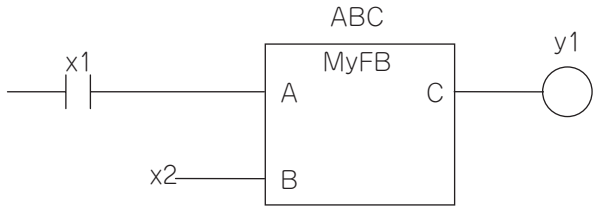


- 각 인수 및 반환값의 기술 순서는 임의입니다.
- 입력 변수명과 입력 파라메터명은 생략할 수 있습니다. 생략한 경우에 이전 호출 시 입력 변수에 대입된 값이 입력 변수에 대입됩니다. 처음 호출된 경우에는 초기값이 입력 변수에 대입됩니다.
- 출력 변수와 출력 파라메터는 생략할 수 있습니다. 생략한 경우에 출력 변수의 값은 어디에도 대입되지 않습니다.

기술 방식 ②:

호출하는 FB의 변수명을 생략하고, 호출원의 파라메터명만 기술하는 경우
ABC(x1, x2, y1);
ABC : FB의 인스턴스명
A, B : 생략합니다. (호출하는 FB의 입력 변수명 또는 입출력 변수명)
x1, x2 : 호출원의 입력 파라메터 또는 입출력 파라메터(상수 포함)
C : 생략합니다. (호출하는 FB의 출력 변수명 또는 상수)
y1 : 호출원의 출력 파라메터

- 래더도 언어로의 표현



- 기술하는 파라메터의 순서는 FB의 정의에 정해져 있습니다. FB의 로컬 변수 정의의 위에서 아래의 순서입니다.

용도:

FB를 호출합니다.

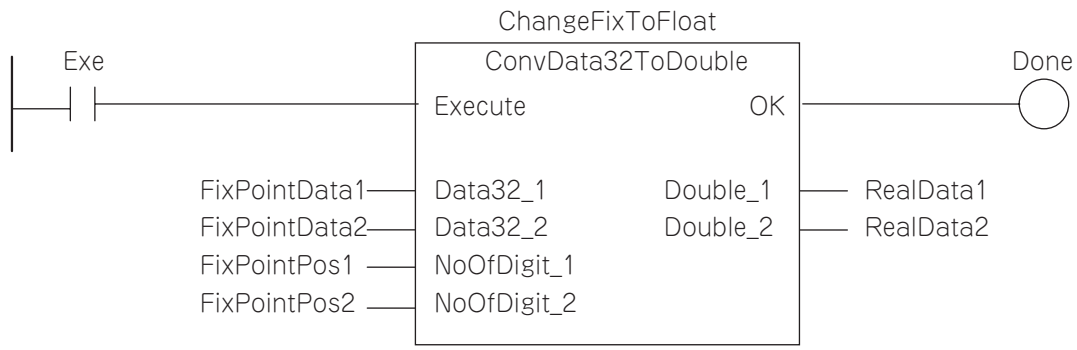
예

- 프로그램

- 기술 ①
  ChangeFixToFloat(Execute:=Exe,Data32_1:=FixPointData1,
  Data32_2:=FixPointData2,
  NoOfDigit_1:=FixPointPos1,
  NoOfDigit_2:=FixPointPos2,OK=>Done,Double_1=>RealData1,
  Double_2=>RealData2);

- 기술 ②
  ChangeFixToFloat(Exe, FixPointData1, FixPointData2, FixPointPos1, FixPointPos2,
  Done, RealData1, RealData2);

- 래더도 언어로의 표현



- 펑션 블록의 정의
  펑션 블록명: ConvData32ToDouble
  펑션 블록의 변수

입력/출력	변수명	데이터 형식
입력 변수	Execute	BOOL
	Data32_1	DINT
	Data32_2	DINT
	NoOfDigit_1	INT
	NoOfDigit_2	INT
출력 변수	OK	BOOL
	Double_1	LREAL
	Double_2	LREAL

- 프로그램의 변수

변수명	데이터 형식	코멘트
ChangeFixToFloat	ConvData32ToDouble	고정 소수점→부동 소수점 변환
Exe	BOOL	기동 트리거
FixPointData1	DINT	소수점 위치 지정 데이터#1
FixPointPos1	INT	소수점 이하 자릿수#1
FixPointData2	DINT	소수점 위치 지정 데이터#2
FixPointPos2	INT	소수점 이하 자릿수#2
Done	BOOL	정상 종료
RealData1	LREAL	부동 소수점 데이터#1
RealData2	LREAL	부동 소수점 데이터#2

파라메터 생략 가능 여부

FB 호출 시 파라메터에 값을 설정할 필요가 없는 경우에는 기술하는 파라메터를 생략할 수 있습니다. 파라메터 생략 가능 여부는 다음과 같습니다.

○: 기술 가능 생략한 경우에는 파라메터 대신 초기값이 대입됩니다.
×: 빌드 시 에러가 됩니다

POU의 종류	호출하는 POU 측의 변수	기술 패턴		기술가능 여부
		기술하는 파라메터	예	
FB	기술 있음 (기술 방식 ①)	모든 파라메터를 기술	instance(x:=a,y:=b,z:=c);	○
		파라메터를 여러 개 기술	instance(x:=a,y:=b);	
		파라메터를 1개 기술	instance(y:=b);	
		파라메터 기술 없음	instance(x:=);	×
	기술 없음 (기술 방식 ②)	모든 파라메터를 기술	instance(a,b,c);	○
		모든 파라메터 없음	instance();	
		선두 파라메터만 기술	instance(a);	×
		파라메터를 1개 기술	instance(a, ,);	
		파라메터를 여러 개 기술	instance(a,b);	

- **FUN 호출**

개요:

FUN을 호출합니다.

예약어: 없음

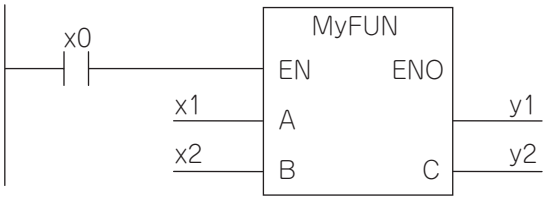
구문의 구조:

대입 키워드(:=)의 좌변에 반환값을 대입하는 출력 파라메터를 기술합니다. 우변에는 FUN명 뒤의 괄호 안에 인수 지정(지정한 변수의 값을 호출 대상 FUN의 입력 변수에 전달하는 지정)을 기술합니다.
다음과 같은 ① 또는 ② 중 어느 한 기술 방식이 가능합니다.
사용자 프로그램의 가독성을 고려하여 ①을 권장합니다.

기술 방식 ①:

```
IF (x0=TRUE) THEN
  y1 := MyFUN(A:=x1, B:=x2, C=>y2);
END_IF;
```

- 래더도 언어로의 표현



MyFUN : FUN명
x0 : FUN의 호출 유무
A, B : 호출하는 FUN의 입력 변수명
x1, x2 : 호출하는 FUN의 입력 파라메터
C : 호출하는 FUN의 출력 변수명
y1 : 호출하는 FUN의 반환값 저장 장소
y2 : 호출하는 FUN의 출력 파라메터

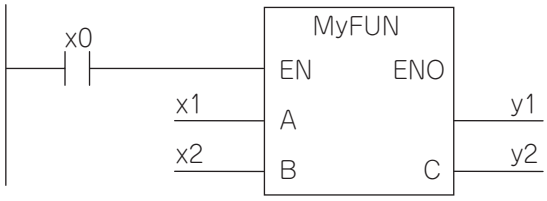
- 각 인수의 기술 순서는 임의입니다.
- 입력 변수명과 입력 파라메터명은 생략할 수 있습니다. 생략한 경우에 입력 변수에는 초기값이 대입됩니다.
- EN도 생략할 수 있습니다. 생략한 경우에는 EN에 TRUE가 대입됩니다.

기술 방식 ②:

호출하는 FUN의 변수명을 생략하고, 호출원의 파라메터명만 기술하는 경우

```
IF (x0=TRUE) THEN
  y1 := MyFUN(x1, x2, y2);
END_IF;
```

- 래더도 언어로의 표현



MyFUN : FUN명
x0 : FUN의 호출 유무
A, B : 호출하는 FUN의 입력 변수명
x1, x2 : 호출하는 FUN의 입력 파라메터
C : 호출하는 FUN의 출력 변수명
y1 : 호출하는 FUN의 반환값 저장 장소
y2 : 호출하는 FUN의 출력 파라메터

- 기술하는 파라메터의 순서는 FUN의 정의에 정해져 있습니다. FUN의 로컬 변수 정의의 위에서 아래의 순서입니다.

예

- 프로그램

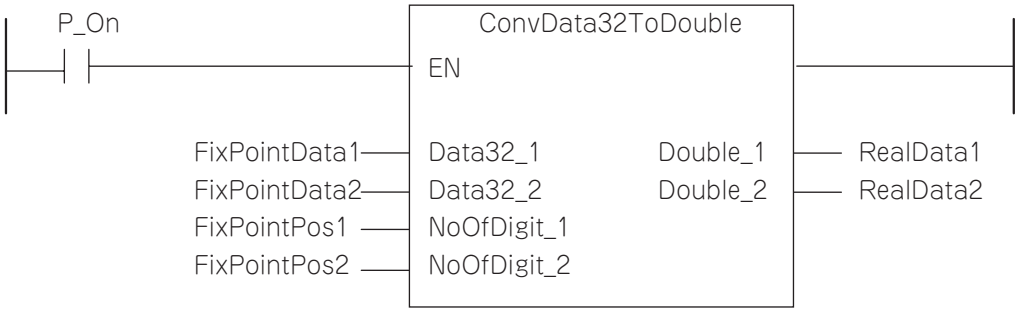
- 기술 ①

```
ConvData32ToDouble(Data32_1:=FixPointData1,Data32_2:=FixPointData2,
NoOfDigit_1:=FixPointPos1, NoOfDigit_2:=FixPointPos2,
Double_1=>RealData1, Double_2=>RealData2);
```

- 기술 ②

```
ConvData32ToDouble(FixPointData1, FixPointData2, FixPointPos1, FixPointPos2,
RealData1, RealData2);
```

- 래더도 언어로의 표현



- 펑션의 정의
  펑션명: ConvData32ToDouble
  펑션의 변수

In/Out	변수명	데이터 형식
입력 변수	Execute	BOOL
	Data32_1	DINT
	Data32_2	DINT
	NoOfDigit_1	INT
	NoOfDigit_2	INT
출력 변수	Double_1	LREAL
	Double_2	LREAL
반환값	-	BOOL

- 프로그램의 변수

변수명	데이터 형식	코멘트
ChangeFixToFloat	ConvData32ToDouble	고정 소수점→부동 소수점 변환
Exe	BOOL	기동 트리거
FixPointData1	DINT	소수점 위치 지정 데이터#1
FixPointPos1	INT	소수점 이하 자릿수#1
FixPointData2	DINT	소수점 위치 지정 데이터#2
FixPointPos2	INT	소수점 이하 자릿수#2
Done	BOOL	정상 종료
RealData1	LREAL	부동 소수점 데이터#1
RealData2	LREAL	부동 소수점 데이터#2

용도:

FUN을 호출합니다.

파라메터 생략 가능 여부

FUN 호출 시 파라메터에 값을 설정할 필요가 없는 경우에는 기술하는 파라메터를 생략할 수 있습니다.
파라메터 생략 가능 여부는 다음과 같습니다.

○: 기술 가능 생략한 경우에는 파라메터 대신 초기값이 대입됩니다.
×: 빌드 시 에러가 됩니다

POU의 종류	호출하는 POU 측의 변수	기술 패턴		기술가능 여부
		기술하는 파라메터	예	
FUN	기술 있음 (기술 방식 ①)	모든 파라메터를 기술	FUN(x:=a,y:=b,z:=c);	○
		파라메터를 여러 개 기술	FUN(x:=a,y:=b);	
		파라메터를 1개 기술	FUN(y:=b);	
		파라메터 기술 없음	FUN(x:=);	×
	기술 없음 (기술 방식 ②)	모든 파라메터를 기술	FUN(a,b,c)	○
		파라메터 기술 없음	FUN();	
		선두 파라메터만 기술	FUN(a);	×
		파라메터를 1개 기술	FUN(a, ,);	
		파라메터를 여러 개 기술	FUN(a,b);	

ST 언어에 대한 주의 사항

ST 언어로 사용자 프로그램을 작성할 때 주의해야 하는 사항을 나타냅니다.

- **암시적 캐스트**
  피연산자의 데이터 형식이 다음과 같이 일치하지 않는 경우에 암시적 캐스트 규칙에 따라 데이터 형식이 자동으로 변환되어 처리됩니다.
  또한, 암시적 캐스트 규칙을 충족하지 않은 경우에는 빌드 시 에러가 됩니다.

  (1) 대입 구문의 우변 식 내에서 피연산자의 데이터 형식이 일치하지 않은 경우
  예)

  ```
  A:=INT#10+SINT#2;
  ```

  (2) 대입 구문의 우변과 좌변의 피연산자의 데이터 형식이 일치하지 않은 경우
  예)

  ```
    A:=B+C;
   /     \
  REAL   INT
  ```

  (3) 구문의 식 내에서 피연산자의 데이터 형식이 일치하지 않은 경우
  예)

  ```
      INT  LINT
        \   /
  CASE A+B OF
    INT#1:
      def:=INT#10;
  ```

아래에서 이 3 가지 경우의 캐스트 규칙을 설명합니다.

대입 구문의 우변에서 연산하는 경우의 캐스트 규칙

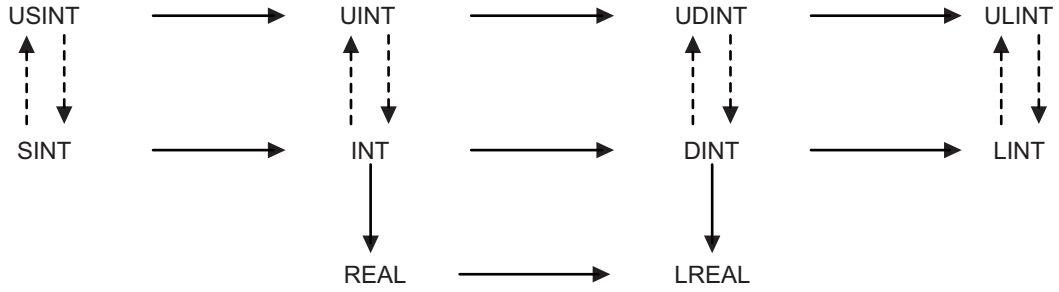
- 우변의 피연산자는 연산자의 피연산자에 사용 가능한 데이터 형식으로 허가된 데이터 형식이면 모두 혼재할 수 있습니다.
- 우변의 피연산자 중 최대 순위를 갖는 데이터 형식이 우변의 데이터 형식이 됩니다.
  (데이터 형식의 순위는 아래의 [데이터 형식의 순위표]를 참조하십시오).

[데이터 형식의 순위표]:
순위가 클수록 그 데이터 형식이 표현할 수 있는 숫자의 범위가 큰 것을 의미합니다.

순위	데이터 형식
1	SINT
2	USINT
3	INT
4	UINT
5	DINT
6	UDINT
7	LINT
8	ULINT
9	REAL
10	LREAL
11	BYTE
12	WORD
13	DWORD
14	LWORD

대입 구문의 우변의 값을 좌변에 대입하는 경우의 캐스트 규칙

다음 그림에서 대입원의 형식에서 대입 대상의 형식에 화살표가 그려진 경우에는 캐스트됩니다.
화살표가 그려지지 않은 경우에는 빌드 시 에러가 됩니다.



——→ 대입 시에 숫자의 부호와 절대값은 변화하지 않습니다.

- - - - - → 대입 시에 숫자의 부호와 절대값이 변화하는 경우가 있습니다.

예)
```
intVar := -1;          (* intVar := 16#FFFF *)
uintVar := 1;
uintVar := intVar;     (* uintVar := 16#FFFF,
                          즉 -1을 대입했지만 결과는 65535이 된다*)
```

화살표는 직접 연결되지 않아도 대입할 수 있습니다.
예를 들어 SINT→USINT→UINT→UDINT→ULINT로 연결되었으므로 ULINT:= SINT는 기술할 수 있습니다.

사용상의 주의 사항

UDINT→ULINT, DINT→LINT, DINT→LREAL의 캐스트인 경우에 다음에 주의하십시오.
이들은 모두, 데이터 크기 32비트의 데이터 형식에서 64비트의 데이터 형식으로의 캐스트입니다.
이 경우에 대입 구문의 우변의 연산 결과가 32비트로 표현할 수 있는 값의 범위를 초과하면 올바른 수치가 대입되지 않을 수 있습니다.

예) 아래의 사용자 프로그램에서는 3번째 행의 덧셈 결과가 32비트로 표현할 수 있는 값의 범위를 초과합니다. 그 결과, 오버플로가 발생하고 LintVar에는 0이 대입됩니다.

```
UdintVar := UDINT#16#FFFF_FFFF;          //32비트의 상한값
DintVar  := DINT#1;                      // 1
LintVar  := (UdintVar + DintVar) / DINT#2;  // (32비트의 상한값+1) / 2
```

이와 같은 경우에는 사전에 각 숫자를 64비트의 데이터 형식으로 변환한 후 연산을 실행하십시오.
위의 예시는 다음과 같이 변경합니다.

```
LintTmp1 := UDINT_TO_LINT(UDINT#16#FFFF_FFFF);  // UDINT→LINT 변환
LintTmp2 := DINT_TO_LINT(DINT#1);               // DINT→LINT 변환
LintVar  := (LintTmp1 + LintTmp2) / DINT#2;
```

구문의 식 내에서의 캐스트 규칙

구문의 식에서도 대입 구문 우변의 연산식 및 우변에서 좌변으로 대입 시의 암시적 캐스트 규칙이 동일하게 적용됩니다.

예)

```
CASE (A+B+C) OF
    Result1:
        ~;
    ResultN:
        ~;
END_CASE;
```

- **FUN 실행 순서**

FUN을 포함하는 식에서는 FUN의 실행 순서가 전적으로 결정되지는 않습니다. CPU 유니트 및 Sysmac Studio의 버전 및 기술 방식에 따라 FUN의 실행 순서는 달라집니다. 따라서 다음과 같이 FUN의 실행 순서에 따라 결과가 다를 가능성이 있는 식은 뒤에 설명하는 방법으로 기술하십시오.

- 동일한 글로벌 변수에 액세스하는 FUN을 여러 개 포함하는 식
- FUN과 그 FUN의 실행에 의해 값이 변화하는 변수를 포함하는 식

동일한 글로벌 변수에 액세스하는 FUN을 여러 개 포함하는 식

다음 식의 경우에 3개 FUN의 실행 순서는 연산자의 우선 순위에 따라 결정하는 연산의 순서와는 관계없습니다. 따라서 3개의 FUN이 FUN2, FUN3, FUN1의 순서대로 실행된다고 단정할 수 없습니다.

```
result := FUN1() + FUN2() * FUN3();
```

위 식의 3개의 FUN이 동일한 글로벌 변수에 쓰기를 포함하는 액세스를 하는 FUN인 경우에 FUN의 실행 순서에 따라 변수 'result'의 값이 다를 수 있습니다.
이와 같은 경우에는 3개 FUN의 실행 순서를 일정하게 하기 위해 식을 나눠서 기술합니다. FUN2, FUN3, FUN1의 순서로 실행시키고자 하는 경우에는 다음과 같이 기술하십시오.

```
tmp2  := FUN2();
tmp3  := FUN3();
result := FUN1() + tmp2 * tmp3;
```

FUN과 그 FUN의 실행에 의해 값이 변화하는 변수를 포함하는 식

다음 식에는 FUN과 그 FUN의 실행에 의해 값이 변화하는 변수가 포함되었습니다.

```
result := varA + FUN4(out => varA);
```

위 식의 경우에 우변 제1항의 변수 'varA'의 평가가 FUN4보다 먼저 실행된다고 단정할 수 없습니다. 따라서 'varA'의 평가와 FUN4의 실행 순서에 따라 변수 'result'의 값이 다를 수 있습니다.
이와 같은 경우에는 'var4'의 평가와 FUN4의 실행 순서를 일정하게 하기 위해 식을 나눠서 기술합니다.
'varA'의 평가를 FUN4보다 먼저 실행시키고자 하는 경우에는 다음과 같이 기술하십시오.

```
tmp    := varA;
result := tmp + FUN4(out => varA);
```

'varA'의 평가를 FUN4보다 뒤에 실행시키고자 하는 경우에는 다음과 같이 기술하십시오.

```
tmp    := FUN4(out => varA);
result := varA + tmp;
```

- **데이터 형식 지정 없는 상수를 포함하는 식의 연산 정밀도**

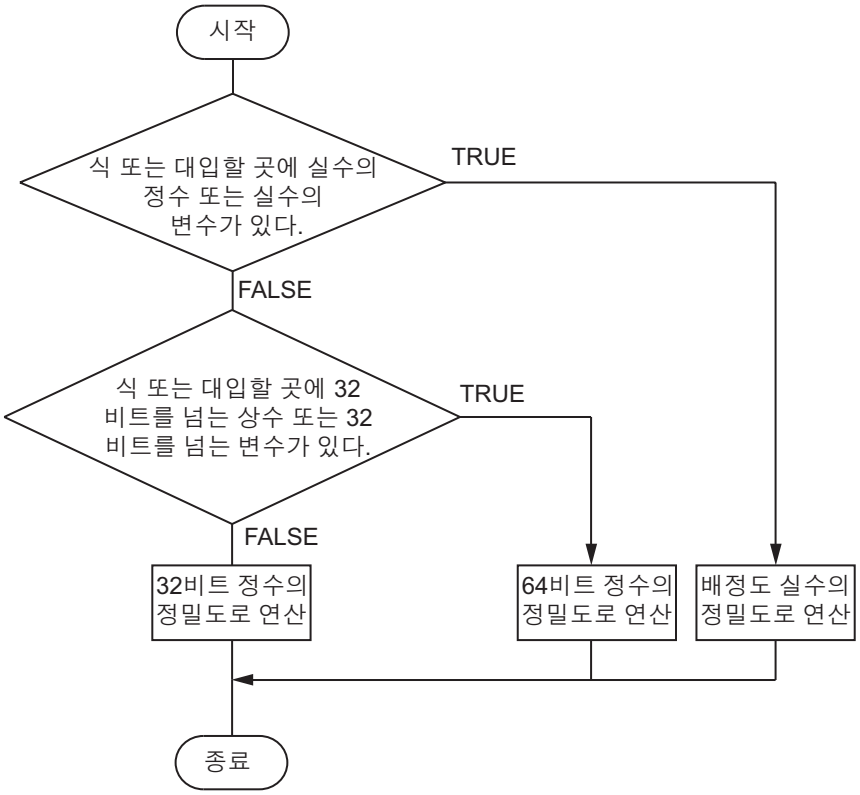
  데이터 형식 지정 없는 상수를 포함하는 식의 연산 정밀도는 그 상수의 표기, 식 및 대입 대상에 쓴 변수 및 상수의 데이터 형식에 따라 자동으로 결정됩니다.

상수의 표기

상수가 '100'과 같이 소수점 없이 표기되면 정수로 처리됩니다. '100.0'과 같이 소수점이 표기되면 실수로 처리됩니다.

식의 연산 정밀도

식의 연산 정밀도는 식 및 대입 대상에 쓴 변수 및 상수의 데이터 형식에 따라 32비트 정수, 64비트 정수, 배정도 실수 중 하나가 됩니다. 식의 연산 정밀도는 다음 규칙으로 결정됩니다.



예) realv:= 2 + 3 * 4; //변수 realv의 데이터 형식은 REAL
변수 realv가 실수 형식이므로 배정도 실수의 정밀도로 연산됩니다. 연산 결과는 14.0입니다.

realv:= 2 + 3.0 *4; //변수 realv의 데이터 형식은 REAL
상수 3.0이 실수 형식의 표기이고, 변수 realv도 실수 형식이므로 배정도 실수의 정밀도로 연산됩니다. 연산 결과는 14.0입니다.

lintv:= 2 + 3 * 4; //변수lintv의 데이터 형식은 LINT
실수 형식의 상수 및 변수가 없고 , 변수 lintv가 32비트를 초과하는 상수이므로 64비트 정수의 정밀도로 연산됩니다. 연산 결과는 14입니다.

intv:= 2 + 3 * 4; //변수 intv의 데이터 형식은 INT
실수 형식의 상수 및 변수가 없고, 32비트를 초과하는 상수도 없으므로 32비트 정수의 정밀도로 연산됩니다. 연산 결과는 14입니다.

단, 나눗셈의 연산 정밀도는 제수와 피제수로만 결정됩니다. 연산 정밀도를 결정하는 규칙은 앞의 흐름도와 같습니다.

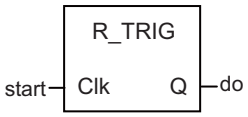
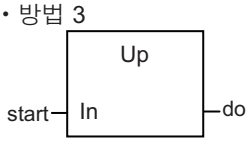
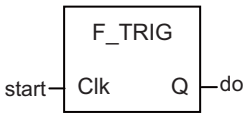
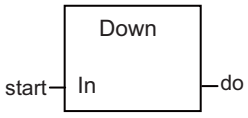
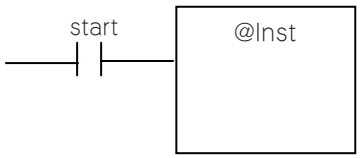
예) realv:= 2/3 * 4; //변수 realv의 데이터 형식은 REAL
나눗셈 2/3는 피제수와 제수 모두 32비트를 초과하지 않는 정수이므로 32비트 정수의 정밀도로 연산됩니다. 나눗셈 결과는 정수인 0이 됩니다.
그 후, 변수 realv가 실수 형식이므로 0*4는 배정도 실수의 정밀도로 연산됩니다. 연산 결과는 0.0입니다.

사용상의 주의 사항

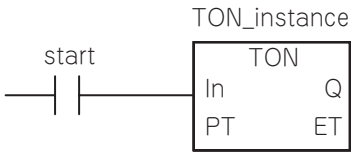
데이터 형식 지정 없는 상수를 포함하는 식의 연산 정밀도는 그 상수의 표기 및 식 내의 변수 및 상수의 데이터 형식 등에서 자동으로 결정됩니다. 따라서 사용자가 의도하지 않은 정밀도로 연산될 가능성이 있습니다. 상수는 REAL#1.0과 같이 데이터 형식을 지정해서 표기하는 것을 권장합니다.

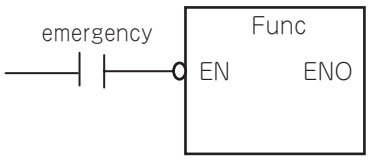
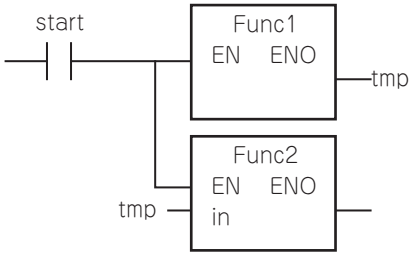
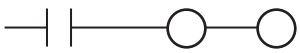
ST와 래더도의 차이

ST와 래더도 사이에 차이가 있는 예는 다음과 같습니다.

항목	래더도	ST(인라인 ST 포함)
접점의 미분	● **상승** • 방법 1 start do ----|↑|----------○ • 방법 2 R_TRIG_instance  • 방법 3  	● **상승** • **방법 1** R_TRIG_instance (Clk:=start, Q=>do); ※ R_TRIG_instance는 R_TRIG 명령의 인스턴스
	● **하강** • 방법 1 start do ----|↓|----------○ • 방법 2 F_TRIG_instance  • 방법 3  	● **하강** • **방법 1** F_TRIG_instance (Clk:=start, Q=>do); ※ F_TRIG_instance는 F_TRIG 명령의 인스턴스
명령의 미분	● **상승 미분**  	● **상승 미분** 해당하는 것이 없으므로 로직으로 실현 예를 들어 • **방법 1** R_TRIG_instance (Clk:=start, Q=>do); IF (do = TRUE) THEN Inst(); END_IF; • **방법 2** IF (start = TRUE) THEN IF (pre_start = FALSE) THEN Inst(); END_IF; END_IF; pre_start:=start;//이전 값의 갱신

항목	래더도	ST(인라인 ST 포함)

복수 태스크 주기 명령의 기술	명령 실행에서 완료까지 여러 사이클이 필요하고, 전력 흐름이 OFF일 때 리셋 동작하는 TON 명령에서 하나의 인스턴스 선언으로 계시와 리셋을 겸함
	 
		다음과 같이 계시용과 리셋용 2가지 인스턴스를 선언
		IF (start = TRUE) THEN
		TON_instance(In:=TRUE, 이후 생략); // 타이머 기동
		ELSE
		TON_instance(In:=FALSE, 이후 생략); // 타이머 리셋
		END_IF;

FB/펑션 인수 반전 지정	BOOL 형식 인수와 FB/펑션의 교점에 ○를 부가하여 반전을 표시 	인수에 NOT 연산자를 부가 ※ NOT 연산자는 인수로 제한하지 않고 모든 BOOL 형식 변수에 부가 가능 IF (NOT emergency) THEN Func(); END_IF;
다단 연결		IF(start=TRUE) THEN Func2(in := Func1()); END_IF;
래더도의 후속	Out 명령 이후 OUT 명령만 후속 가능 	인라인 ST 다음에 래더도는 후속 불가능 
프로그램의 분할 단위	섹션 기능 있음	섹션 기능 없음

6-6 명령

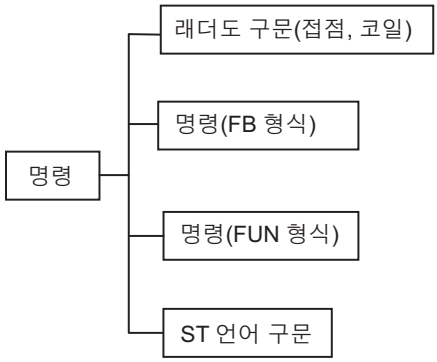
여기에서는 사전에 NJ/NX 시리즈에서 정의하는 명령에 대해 설명합니다.

명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)" 및 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

6-6-1 명령의 개념

명령은 POU의 알고리즘 중에 기재하는 OMRON의 최소 단위의 처리 요소입니다.

명령에는 다음과 같은 종류가 있습니다.



프로그램, 사용자 정의의 FB, 사용자 정의의 FUN은 이 명령의 조합으로 구성됩니다.

6-6-2 명령의 기본 지식

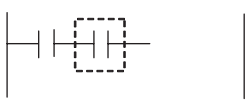
명령의 기본적인 사양은 FB, FUN의 사양에 따릅니다.

여기에서는 명령 특유의 사양에 대해 설명합니다.

래더도 구문(접점·코일)

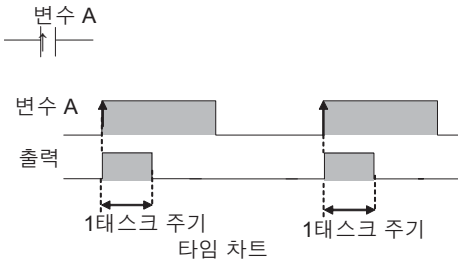
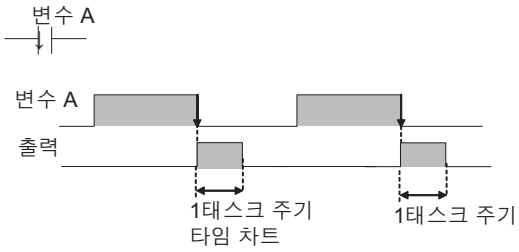
- **배치 가능 위치**

  래더도 언어의 접점·코일의 형태를 한 명령에는 다음과 같이 배치 가능한 위치가 있습니다.

분류		배치 가능 위치	그림
입력계	논리 시작 형식 (LD 형식)	좌측 버스바에 직접 연결 또는 회로 블록의 시작점	
	다음 단계 연결 형식	논리 시작 형식과 출력 형식의 중간 위치	
출력계		우측 버스바에 직접 연결	

● **동작 옵션**

래더도 언어의 명령(접점)에도 위쪽 화살표(↑), 아래쪽 화살표(↓)를 부가하면 상승 미분, 하강 미분 동작을 할 수 있습니다.

상승 (↑)		태스크 주기마다 접점의 읽기/비교/비트 테스트 등의 처리를 실행하고, 결과가 FALSE → TRUE로 변화하면 전력 흐름을 출력하는 명령입니다. 출력하는 전력 흐름은(1태스크 주기 TRUE 이후) 다음 태스크 주기에 FALSE가 됩니다.
하강 (↓)		태스크 주기마다 접점의 읽기 등의 처리를 실행하고, 결과가 TRUE → FALSE로 변화하면 전력 흐름을 출력하는 명령입니다. 출력하는 전력 흐름은(1태스크 주기 TRUE 이후) 다음 태스크 주기에 FALSE가 됩니다.

펑션 블록(FB) 형식의 명령

● **실행 조건**

FB 형식 명령의 실행 조건은 명령에 따라 다릅니다.
실행 조건용 특정 입력 변수가 명령별로 정의되어 있습니다.
예: Execute: 상승 엣지 지정, 하강 엣지 지정
Enable: 태스크 주기마다 실행

FB 형식 명령은 그 명령의 호출원인 POU가 실행되는 동안 실행을 계속합니다.

● **동작 옵션**

지정할 수 없습니다.

펑션(FUN) 형식의 명령

- **실행 조건**

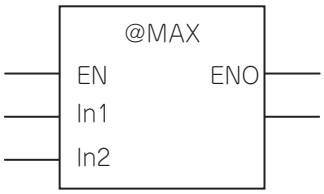
  FUN 형식 명령 실행 조건은 모두 EN 입력입니다. EN이 TRUE인 동안 태스크 주기마다 실행됩니다.

- **동작 옵션**

  래더도 언어인 경우에 다음 동작 옵션을 부가하면 그 실행 조건을 상승 지정으로 할 수 있습니다. ST 언어에는 동작 옵션이 없습니다.

동작 옵션		옵션 기호	
미분 옵션	상승	@	명령을 입력 상승 미분형 명령으로 합니다. EN 상승 시에 1회만 명령을 실행합니다.

  동작 옵션을 부가하려면 명령명 앞에 위의 옵션 기호를 붙입니다.
  예)



펑션 블록(FB) 형식/펑션(FUN) 형식의 명령 공통

- **조건 플래그**

  명령의 처리의 결과로 값이 대입되는 시스템 정의 변수를 '조건 플래그'라고 합니다. NJ/NX 시리즈의 조건 플래그는 캐리 플래그(P_CY)뿐입니다.
  캐리 플래그는 다음 중 하나의 의미를 가집니다.
  - 명령의 처리 결과가 출력 변수의 데이터 형식으로 표현할 수 있는 범위를 초과했는지 여부
  - 비트 열 데이터의 비트 시프트 명령 등에서 자릿수 오버플로가 발생했는지 여부의 판정에 대한 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 각 명령의 설명을 참조하십시오.

6-6-3 명령 에러

명령 실행에 따라 발생하는 이상을 명령 에러가 합니다. 여기에서는 명령 에러의 발생 타이밍, 명령 에러에서 검출되는 이상, 명령 에러 발생 시의 동작 등에 대해 설명합니다.

명령 에러 발생 타이밍

명령 에러 발생 타이밍에는 다음 3가지가 있습니다. 명령 에러 발생 타이밍에 따라 검출되는 이상 및 이상 발생 시의 동작이 다릅니다.

- 명령 실행 전에 입력 파라메터 또는 입출력 파라메터의 값을 확인했을 때 발생
- 명령 실행 중에 명령의 내부 처리에 따라 발생
- 명령 실행 후 출력 파라메터의 값을 확인했을 때 발생

명령 에러에서 검출되는 이상

명령 에러에서 검출되는 이상은 다음과 같습니다. 명령 에러의 발생 타이밍에 따라 검출되는 이상이 다릅니다.

- **명령 실행 전, 명령 실행 후에 검출되는 이상**

  명령 실행 전, 명령 실행 후에 검출되는 이상은 다음과 같습니다.
  - 배열 지정 변수의 배열 영역을 초과한 요소에 대한 읽기/쓰기
  - 정의된 바이트 수보다 긴 문자열을 STRING 형식 변수에 대입
  - NULL 문자로 끝나지 않는 문자열을 STRING 형식 변수에 대입
  - 정수 형식 변수의 0 나눗셈

- **명령 실행 중에 검출되는 이상**

  명령 실행 중에 검출되는 이상은 명령에 따라 다릅니다. 각 명령에서 검출되는 이상의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

명령 에러 발생 시의 동작

명령 에러 발생의 유무에 따라 동작이 달라지는 요소에 출력 변수 'ENO', 출력 변수 'Error', 출력 변수 'ErrorID', 시스템 정의 변수 'P_PRGER', 이벤트가 있습니다. 각각에 대해 설명합니다.

- **출력 변수 'ENO', 출력 변수 'Error', 출력 변수 'ErrorID'**

  이상 발생의 유무를 나타내는 출력 변수에 'ENO'(Enable Out), 'Error'(에러), 'ErroriD'(에러 코드)가 있습니다. 명령에 따라 어느 출력 변수를 갖는지가 다릅니다. 각 변수의 의미와 명령 에러 발생 시의 값은 다음과 같습니다. 명령 에러 발생 타이밍에 따라 동작이 다릅니다.

출력 변수	데이터 형식	의미	명령 에러 발생 시의 값		
			명령 실행 전*1	명령 실행 중	명령 실행 후*2
'ENO'	BOOL	TRUE: 정상 종료 FALSE: 이상 종료, 실행 중, 실행 안 함	FALSE	FALSE	TRUE
'Error'	BOOL	TRUE: 이상 종료 FALSE: 정상 종료, 실행 중, 실행 안 함	FALSE	TRUE	FALSE
'ErrorID'	WORD	이상 종료 시에는 에러 코드, 정상 종료 시에는 WORD#16#0	WORD#16#0	에러 코드	WORD#16#0

*1. 명령 실행 전에 명령 에러가 발생한 경우에는 명령이 실행되지 않습니다. 따라서 각 출력 파라메터의 값은 명령 실행 전 그대로 유지됩니다.

*2. 명령 실행 후 명령 에러가 발생한 경우에는 명령 자체는 정상 종료했다고 판단됩니다. 따라서 각 출력 파라메터에는 명령의 출력 변수 값이 대입됩니다. 단, 이상이 발생한 출력 파라메터의 값은 명령 실행 전 그대로 유지됩니다.

- **시스템 정의 변수 'P_PRGER'**

  시스템 정의 변수 'P_PRGER'는 명령 에러 발생 플래그입니다. 명령 에러가 발생하면 명령 에러 발생 타이밍에 관계없이 값이 TRUE가 됩니다. 정상 종료 시에는 값을 유지합니다. 'P_PRGER'의 자세한 내용은 📖'명령 에러 발생 플래그(6-141페이지)'를 참조하십시오.

- **이벤트**

  명령 에러가 발생하면 이상 내용에 대한 이벤트가 발생합니다. 이벤트의 자세한 내용은 📖'명령 에러 발생에 따른 이벤트 발생(6-142페이지)'을 참조하십시오.

버전 관련 정보

명령 에러 발생 시 이벤트가 발생하는 것은 CPU 유니트 Ver.1.02 이상입니다.

래더도 언어에서의 출력 파라메터

래더도 언어로 기술된 명령, 사용자 정의의 펑션, 사용자 정의의 펑션 블록이 정상 종료한 경우와 명령 에러가 발생한 경우의 출력 파라메터의 값은 다음과 같습니다.

조건	출력 파라메터의 종류	출력 파라메터의 값
정상 종료 시	전력 흐름 출력	내부 알고리즘에 따라 값이 갱신됩니다.
	파라메터 출력 BOOL 형식	
	파라메터 출력 BOOL 형식 이외	
명령 에러 발생 시	전력 흐름 출력	FALSE가 됩니다.
	파라메터 출력 BOOL 형식	그 때까지의 값을 유지합니다.
	파라메터 출력 BOOL 형식 이외	

ST 언어로 기술된 POU에서의 구문 에러 발생 시의 동작

- **대입 구문에서의 에러**

ST 언어로 기재된 대입 구문에 에러가 발생한 경우에는 그 행을 실행하지 않습니다.

```
5  a = b / (c + d) + e * f + ABS(g);
6  x := 1;
```

예를 들어, 5번째 줄의 b/(c+d)에서 0으로 나누기가 발생한 경우에는 5번째 줄의 실행을 중단하고(a의 값은 변하지 않음) 6번째 줄을 실행한다.

사용자 작성 펑션의 출력 ENO가 OFF인 경우도 동일합니다.

```
5  a=사용자 작성 펑션 (b) + c;
6  x := 1;
```

사용자 작성 펑션의 출력 인 ENO가 OFF일 때, 5번째 줄의 실행을 중단하고(a의 값은 변하지 않음) 6번째 줄을 실행한다.

- **IF 구문에서의 에러**

ST 언어의 구문에서 에러가 발생한 경우에 구문에 기초해서 에러 처리를 실행합니다.
아래에서 (c+d)의 값이 제로인 경우에 IF~END_IF 간은 실행하지 않습니다.

POU「○○」

```
5  IF a = b / (c + d) THEN
6    x := 1;
7  ELSE
8    x := 2;
9  END_IF;
10 y := 10;
 :  :
   IF P_PRGER = TRUE THEN
     x :=초기 값; (* 이상 발생시 처리 *)
     y := 초기 값;
   END_IF;
```

사용자는 에러가 발생한 경우의 안전 처리를 기재한다.

- **ST 언어에서의 구문 에러**

ST 언어에서의 구문 에러는 다음과 같습니다.

- 배열[i]의 요소 수 초과
- 입출력 변수에 파라메터가 설정되지 않음
- 문자열 형식의 대입: 좌변의 문자열 형식 크기(Byte)＜우변 문자열 길이(Byte)인 경우
- 나눗셈의 0 나누기(부동 소수점 연산은 제외)

※부동 소수점수의 값이 NaN(숫자가 아님)인 경우의 연산은 연산 결과가 NaN(숫자가 아님)이 되지만, 에러로는 간주되지 않습니다.

- **구문 에러 발생 시의 동작**

P_PRGER 플래그가 TRUE가 되고, 다음과 같이 됩니다.

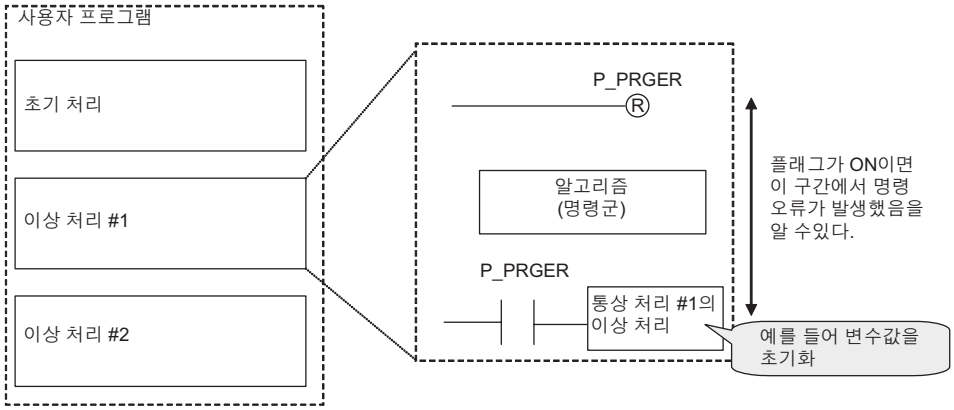
구문	에러 발생 부분	동작
대입 구문		그 행을 실행 안 함
제어 구문	**IF 조건식 내**	IF~END_IF 간을 실행 안 함
	CASE 조건식	CASE~END_CASE 간을 실행 안 함
	FOR 조건식 내	FOR~END_FOR 간을 실행 안 함
	WHILE 조건식 내	WHILE~END_WHILE 간을 실행 안 함
	REPEAT 조건식 내	REPEAT~END_REPEAT 간을 실행 안 함

명령 에러 발생 플래그

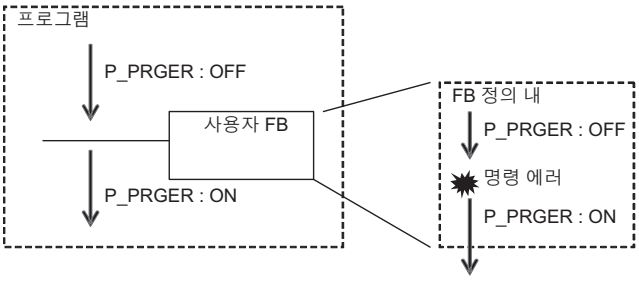
래더도 알고리즘 내에 명령 에러 또는 ST 알고리즘에서 구문/함수에 에러가 있었던 경우에 시스템 정의 변수 'P_PRGER'(명령 에러 발생 플래그)가 TRUE가 됩니다.
P_PRGER 플래그는 프로그램의 로컬 변수입니다. 프로그램 내에 명령 에러가 발생했을 때 TRUE가 되고, 다음 태스크 주기에서도 값을 유지합니다.

변수명	이름	기능	데이터 형식	값의 범위	초기값	R/W
P_PRGER	명령 에러 발생 플래그	명령 에러가 발생했을 때 TRUE가 계속되는 플래그입니다. 한번 TRUE가 되면 사용자 프로그램에 의해 FALSE가 될 때까지 TRUE로 유지됩니다.	BOOL	FALSE/TRUE	FALSE	RW

P_PRGER 플래그는 사용자에 의한 덮어쓰기도 가능합니다. 따라서 다음과 같이 일단 사용자 조작으로 FALSE로 하면 특정 범위에서 명령 에러가 발생했는지 여부를 판단할 수도 있습니다. 한번 TRUE가 되면 동작 모드를 변경하거나 또는 사용자 프로그램 등으로 덮어쓰기할 때까지값을 계속 유지합니다.
예)



프로그램 내에서 사용하는 사용자 정의 FB 내에 명령 에러가 발생한 경우도 P_PRGER 플래그는 TRUE가 됩니다.
예)



명령 에러 발생에 따른 이벤트 발생

명령 에러 발생에 따라 그에 대응하는 이벤트가 발생합니다.

이벤트의 확인 방법의 자세한 내용은 '8-7이벤트 로그 기능(8-66페이지)'을 참조하십시오. 또한, 이벤트의 내용에 대해서는 "NJ/NX 시리즈 명령 참조 기본편(SBCA-360)"을 참조하십시오.

사용상의 주의 사항

- 명령 에러 발생 시 이벤트가 발생하는 것은 Sysmac Studio의 컨트롤러 설정에서 [이벤트 로그 설정]–[명령 에러 출력]을 [사용]으로 설정한 것입니다. Sysmac Studio의 컨트롤러 설정의 자세한 내용은 '4-2-2컨트롤러 설정(4-4페이지)' 및 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.
- 명령 에러 발생 후 사용자 프로그램을 변경하면 이벤트 로그에 정보가 올바르게 기록되지 않는 경우가 있습니다.
- 이상이 있는 명령을 반복해서 실행한 경우에 명령 에러는 매번 발생합니다. 따라서 이벤트 로그의 기록 건수가 상한을 초과하는 경우가 있습니다. 그 때 오래된 정보부터 순서대로 덮어쓰기되므로 주의하십시오.

버전 관련 정보

- 명령 에러 발생 시 이벤트가 발생하는 것은 CPU 유니트 Ver.1.02 이상입니다.
- 명령 에러 발생 시 명령 에러 출력 사용/사용 안 함의 전환은 CPU 유니트 Ver.1.02 이상과 Sysmac Studio Ver.1.03 이상의 조합에서 사용할 수 있습니다.

참고

- 모션 제어 명령의 실행 시 이상이 발생한 경우에 명령 에러에 따른 이벤트와 모션 제어 명령 고유의 이벤트 모두 발생합니다. 모션 제어 명령의 이벤트의 자세한 내용은 "NJ/NX 시리즈 명령 참조 모션편(SBCE-364)"을 참조하십시오.
- Sysmac Studio의 컨트롤러 설정에서 [이벤트 로그 설정]–[명령 에러 출력]을 [사용 안 함]으로 설정해도 모션 제어 명령 고유의 이벤트는 발생합니다.

6-7 이름 공간

여기에서는 이름 공간의 사양과 사용 방법에 대해 설명합니다. 조작 방법에 대해서는 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

버전 관련 정보

이름 공간은 CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.

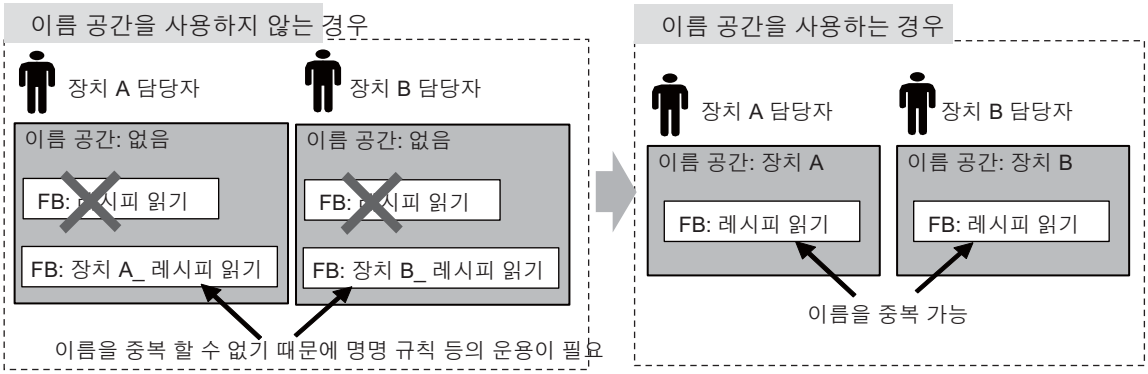
6-7-1 이름 공간의 개념

이름 공간은 FB 정의 등을 그룹으로 구분하여 계층 구조로 관리하는 구조입니다. 이는 파일을 폴더에 구분하여 디렉토리의 계층 구조로 관리하는 것과 같습니다. 이름 공간을 사용하지 않는 경우에 FB 정의 등에 동일한 이름을 중복해서 사용할 수 없지만 이름 공간을 사용하여 다른 이름 공간을 설정하면 중복해서 사용할 수 있습니다.
이름 공간은 사용해도 되고 안 해도 됩니다.

이름 공간의 특징

다음과 같은 특징이 있습니다.

- **이름의 중복 회피**
  다른 이름 공간이면 FB 정의명 등에 동일한 이름을 사용할 수 있습니다.
  예를 들어 여러 장치를 프로그래밍하는 경우 장치별로 다른 담당자가 프로그래밍을 하면 FB 정의명 등이 중복될 가능성이 있습니다. 이름 공간을 사용하지 않는 경우에는 명명 규칙의 운용 등에 의해 이름의 중복을 피합니다. 이름 공간을 사용하여 장치별로 이름 공간을 설정하면 장치 사이에서의 이름의 중복을 의식하지 않고 프로그래밍할 수 있습니다.



6-7-2 이름 공간의 사양

여기에서는 이름 공간의 대상과 표현 및 사용 선언에 대해 설명합니다.

이름 공간의 대상

이름 공간에서 취급할 수 있는 대상은 다음과 같습니다.

대상	세부 내용
POU 정의	FB 정의명, FUN 정의명
데이터 형식	구조체 형식명, 공용체 형식명, 열거 형식명

이름 공간의 표현

이름 공간은 '\(백슬래시)'로 구분해서 표현합니다. POU의 알고리즘 내에서 사용하는 경우에는 이름 공간의 선두에 '\\(백슬래시 2개)'를 부가합니다.
'\(백슬래시)'는 Windows의 국가 및 언어 설정 및 폰트 등에 따라 '￥'이 표시되는 경우도 있습니다.

예)

사용하는 상황	표현
알고리즘 이외에서 사용	장치 A\레시피 읽기
알고리즘 내에서 사용	\\장치 A\레시피 읽기

- **완전 한정명과 단순명**

  이름 공간명을 생략하지 않고 이름 공간의 대상을 표현한 이름을 완전 한정명이라고 합니다. 이름 공간명을 생략한 이름을 단순명이라고 합니다.
  POU 정의의 알고리즘 내에서는 동일한 이름 공간에 속하는 POU 정의를 단순명으로 기술할 수 있습니다.

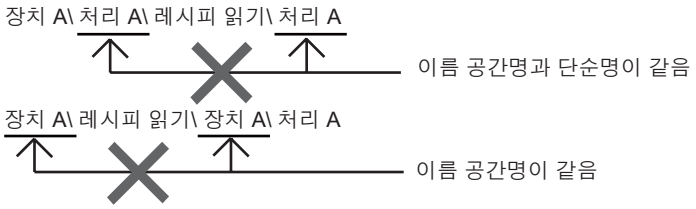
  예) 장치 A\ 레시피 읽기
  이름 공간명 단순명
  완전 한정명

- **이름 공간의 표현 제한**
  - 사용할 수 있는 문자는 변수명과 동일합니다. 자세한 내용은 '6-3-12 변수명 등의 프로그램 관련 이름의 제한(6-83페이지)'을 참조하십시오.
  - 이름 공간의 문자 수 제한은 다음과 같습니다.

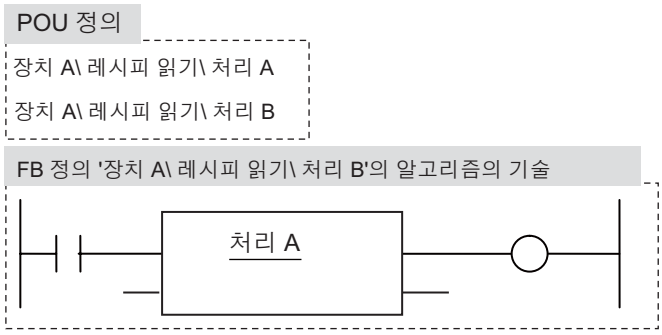
이름	최대 크기	문자 코드
이름 공간명	93바이트	UTF-8
단순명	127바이트	

사용상의 주의 사항

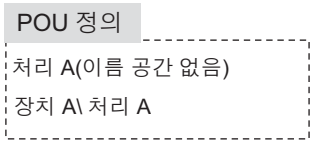
- 이름 공간명과 단순명이 동일한 경우에는 빌드 시 에러가 됩니다.



- POU 정의의 알고리즘 내에서는 동일한 이름 공간에 속하는 POU 정의를 단순명으로 기술할 수 있지만 그 이름 공간의 상위 계층에 동일한 단순명의 POU 정의 또는 데이터 형식 등이 존재하면 빌드 시 에러가 됩니다.
  예를 들어 다음 그림과 같이 POU 정의가 존재하는 경우에 이름 공간 '장치 A\레시피 읽기' 상에 있는 FB 정의 '처리 B'의 알고리즘 내의 기술로 동일한 이름 공간에 속하는 '장치 A\레시피 읽기\처리 A'를 단순명으로 호출할 수 있습니다.



  단, 다음과 같이 이름 공간 '장치 A\레시피 읽기'보다 상위 계층에도 POU 정의 '장치 A'가 존재하는 경우에는 복수의 단순명 '처리 A'가 존재하므로 빌드 시 에러가 됩니다.
  이 경우에는 완전 한정명으로 기술하거나 단순명을 변경하십시오.



- 시스템이 내부에서 사용하는 예약어와 합치한 경우에는 프로그램 확인 시 에러가 됩니다.

이름 공간의 사용 선언

프로그래밍 시에 POU 정의의 알고리즘 내에서 사용하는 이름 공간을 사전에 선언하는 것을 사용 선언이라고 합니다.
POU 정의로 사용 선언하면 사용 선언한 이름 공간에 속하는 POU 정의 등을 단순명으로 사용할 수 있습니다. 사용 선언한 경우에도 완전 한정명을 사용할 수 있습니다.
POU 정의의 알고리즘 내에서는 POU 정의와 동일한 이름 공간에 속하는 FB 정의 및 FUN 정의를 사용하는 경우에 사용 선언하지 않아도 단순명을 사용할 수 있습니다.
하나의 POU 정의에 대해 복수의 사용 선언이 가능합니다.

- **기술 예**

  FB 정의 작성 시 사용하는 대상의 FB 정의의 이름 공간을 사용 선언하지 않은 경우와 사용 선언하는 경우의 기술 예를 설명합니다.

예)
FB 정의 '리프터'의 알고리즘 내에 FB 정의 '레시피 읽기' 및 FB 정의 '상한 계산'을 기술하는 예입니다. 각각의 FB 정의는 다른 이름 공간에 속합니다. FB 정의 '리프터'에는 이름 공간 '장치 C'만 사용 선언합니다.
이름 공간 '장치 C'에 속하지 않는 FB 정의 '레시피 읽기'는 완전 한정명으로 기술합니다. 이름 공간 '장치 C'에 속하는 FB 정의 '상한 계산'은 단순명으로 기술할 수 있습니다.

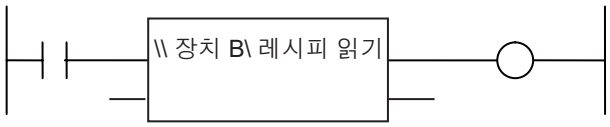
이름 공간	대상의 종류	단순명
장치 A	FB 정의	리프터
장치 B	FB 정의	레시피 읽기
장치 C	FB 정의	상한 계산

FB 정의 '리프터'의 [이름 공간-사용 선언]의 기술

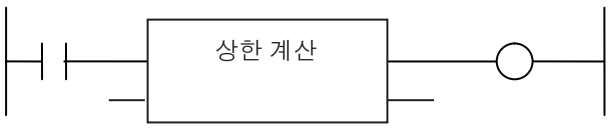
장치 C

FB 정의 '리프터'의 알고리즘 내를 기술

•이름 공간을 사용 선언하지 않는 경우
FB 정의 '레시피 읽기'가 속한 이름 공간 '장치 B'를 사용 선언하지 않았으므로 완전한정명으로 기술합니다.



•이름 공간을 사용 선언한 경우
FB 정의 '상한 계산'이 속한 이름 공간 '장치 C'를 사용 선언했기 때문에 단순명으로 기술할 수 있습니다.



- **사용 선언의 제한**

단순명은 POU 정의의 알고리즘 내에서만 사용할 수 있습니다.

사용상의 주의 사항

- 다음과 같은 경우에는 프로그램 확인에서 에러가 됩니다.
  - 존재하지 않는 이름 공간을 사용 선언한 경우
  - 하나의 POU 정의에 복수의 이름 공간을 사용 선언하고, 각각의 이름 공간에 동일한 이름의 POU 정의 및 데이터 형식 등이 존재하는 경우
- 아래에서 POU 정의 및 데이터 형식 등에 동일한 이름을 붙이면 빌드 시 에러가 됩니다.
  - POU 정의가 속하는 이름 공간과 그 상위 계층의 이름 공간
  - 사용 선언한 이름 공간
  - 이름 공간 없음

이름 공간	대상의 종류	단순명	
장치A\리프터	FB 정의	처리 A	✕ POU 정의가 속한 이름 공간
장치 A	FB 정의	처리 A	✕ POU 정의가 속한 이름 공간의 상위 계층
장치 B	FB 정의	처리 B	
		처리 A	✕ 사용 선언한 이름 공간
없음	FB 정의	처리 A	✕ 이름 공간 없음

FB 정의 '처리 A'의 [이름 공간-사용 선언]의 기술

장치 B

참고

프로그램명에 대해 이름 공간을 설정할 수 없지만 프로그램의 알고리즘 내에서 사용하는 부품의 이름 공간을 사용 선언할 수 있습니다.

6-7-3 이름 공간 사용 순서

Sysmac Studio를 사용하여 이름 공간 설정과 사용 선언을 실행합니다.
데이터 형식을 작성하거나 FB 정의 및 FUN 정의 등의 부품를 작성하거나 할 때, 순서 1과 순서 2의 설정을 실행합니다.
이름 공간을 설정한 부품를 사용할 때, 순서 3의 사용 선언을 실행합니다.

1. **데이터 형식 편집기에서 대상의 데이터 형식으로 이름 공간을 설정합니다.**
2. **대상의 FB 정의 및 FUN 정의의 속성에 이름 공간을 설정합니다.**
3. **래더 편집기 또는 ST 편집기로 대상의 FB 정의 및 FUN 정의의 속성에 설정한 이름 공간을 사용 선언합니다.**
4. **사용자 프로그램에 대상의 데이터 형식 및 FB 정의 등을 기술합니다.**

조작의 자세한 내용은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

6-8 라이브러리

여기에서는 라이브러리의 사양에 대해 설명합니다. 조작 방법의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

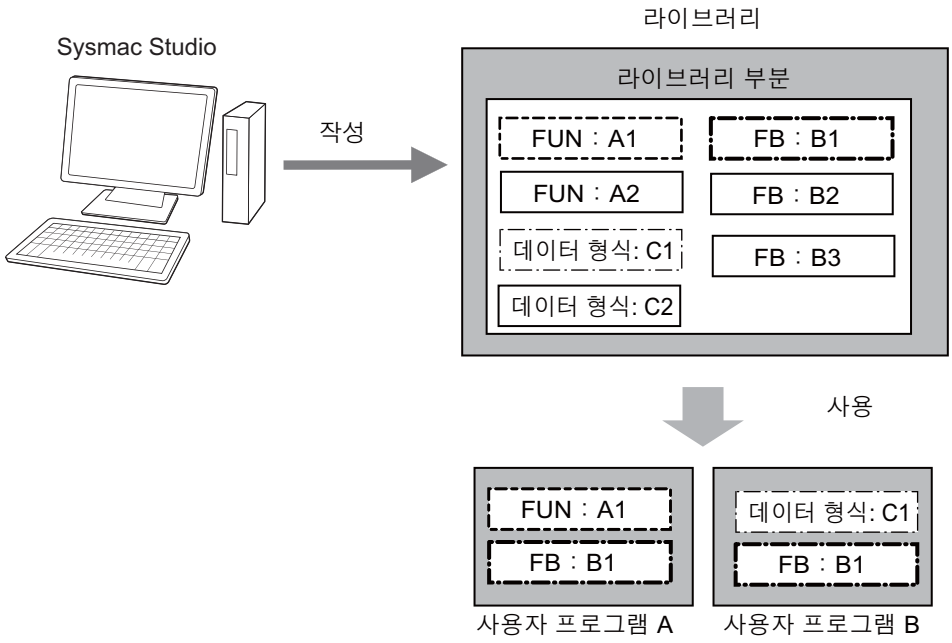
버전 관련 정보

라이브러리는 CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.

6-8-1 라이브러리의 개요

POU 정의 및 데이터 형식을 유용하기 위한 부품으로 이용 가능한 형식으로 묶은 것을 라이브러리라고 합니다. 라이브러리의 부품을 라이브러리 부품이라고 합니다.
NJ/NX 시리즈 CPU 유니트에서는 라이브러리의 작성과 이용이 가능합니다.

다음 그림에서는 라이브러리 안의 라이브러리 부품 중에 사용자 프로그램 A가 FUN: A1과 FB: B1을 사용하고, 사용자 프로그램 B가 데이터 형식: C1와 FB: B1을 각각 사용합니다.



6-8-2 라이브러리 사양

라이브러리의 설정, 동기에 대해 설명합니다.

라이브러리의 설정

라이브러리에는 다음과 같은 항목을 설정할 수 있습니다.

설정 항목	설명
이름	라이브러리의 이름입니다.
버전	라이브러리의 버전입니다.
작성자	라이브러리 작성자의 이름입니다. (임의)
작성일	라이브러리를 새로 작성한 날짜입니다.
갱신일	라이브러리를 갱신한 날짜입니다.
코멘트	라이브러리의 코멘트입니다. (임의)
회사명	라이브러리를 작성한 회사의 이름입니다. (임의)
ID	라이브러리 비교용의 고유한 ID입니다. 자동 생성되고 변경할 수 없습니다.
소스 표시 불가능	소스의 표시 또는 비표시를 선택할 수 있습니다.*1
첨부 파일	복수의 임의의 파일을 첨부할 수 있습니다.

*1. 라이브러리 부품에 데이터 보호가 설정된 경우에는 소스 코드를 표시할 때 비밀번호가 필요합니다.

라이브러리 부품은 다른 라이브러리를 참조하여 작성할 수도 있습니다. 이 경우에는 참조할 항목의 라이브러리의 데이터도 포함하는지 여부를 선택합니다.

참고

참조할 항목의 라이브러리를 포함한다고 가정했을 때, 참조할 항목의 라이브러리 복사를 포함하여 참조원 라이브러리가 작성됩니다. 따라서 라이브러리의 파일 1개로 가능하지만 참조원의 라이브러리가 여러 개일 때는 개별적으로 변경해야 합니다.
포함하지 않는다고 가정했을 때는 참조할 항목의 라이브러리는 포함하지 않고 참조원의 라이브러리가 작성됩니다. 따라서 라이브러리의 파일은 참조원과 참조할 항목의 여러 개로 분할되지만 참조원의 라이브러리가 여러 개일 때의 변경은 참조할 항목의 라이브러리만으로 가능합니다.

- **라이브러리 부품의 선택**
  라이브러리에 포함되는 라이브러리 부품을 선택합니다.

라이브러리의 동기

라이브러리 단위로 Sysmac Studio에서의 다운로드, 컨트롤러에서의 업로드, 컨트롤러와의 비교가 가능합니다.

참고

- 소스를 표시하는 라이브러리가 전송된 컨트롤러에 Sysmac Studio에서 '프로그램의 소스를 전송 안 함' 설정으로 프로젝트를 전송한 경우에는 라이브러리의 소스는 전송되지 않습니다.
- 메모리 전체 클리어하면 컨트롤러 내의 라이브러리는 삭제됩니다.

6-8-3 라이브러리 부품의 사양

라이브러리 부품의 대상과 라이브러리 부품의 설정에 대해 설명합니다.

라이브러리 부품의 대상

다음과 같은 데이터를 라이브러리 부품으로 취급할 수 있습니다.

라이브러리 부품	세부 내용
POU 정의	FB, FUN
데이터 형식*1	구조체 형식, 공용체 형식, 열거 형식

*1. Sysmac Studio의 부품 선택 화면에서 데이터 형식은 항상 라이브러리에 포함됩니다.

라이브러리 부품의 설정

각각의 라이브러리 부품에 다음과 같은 항목을 설정할 수 있습니다.

설정 항목	정의
이름	라이브러리 부품의 이름입니다.
이름 공간	라이브러리 부품의 이름 공간입니다.
버전*1	라이브러리 부품의 버전입니다.
작성자*1	라이브러리 부품 작성자의 이름입니다. (임의)
작성일*1	라이브러리 부품을 새로 작성한 날짜입니다.
갱신일*1	라이브러리 부품을 갱신한 날짜입니다.
코멘트	라이브러리 부품의 코멘트입니다. (임의)

*1. Sysmac Studio의 각 POU 정의의 속성으로 FB와FUN에 대해서만 설정할 수 있습니다.

6-8-4 라이브러리 사용 순서

라이브러리 작성 순서와 이용 순서는 다음과 같습니다.

라이브러리 작성 순서

라이브러리는 프로젝트로 작성합니다. 다음과 같은 순서로 라이브러리를 작성해서 저장합니다.

1 **라이브러리 프로젝트를 새로 작성합니다.**
프로젝트 창의 프로젝트 종류에 '라이브러리 프로젝트'를 선택하여 프로젝트를 작성합니다.

2 **라이브러리 부품을 작성합니다.**
라이브러리 프로젝트에 POU 정의 및 데이터 형식 등을 작성하고 동작을 확인합니다.

3 **라이브러리의 설정을 합니다.**
라이브러리 프로젝트에 라이브러리 부품 선택, 소스 표시, 첨부 파일 등의 설정을 합니다.

4 **라이브러리 파일로 저장합니다.**
[라이브러리 작성] 창에서 라이브러리 파일로 저장합니다.

참고

- 이미 작성된 프로젝트에서도 프로젝트에 등록된 장치가 컨트롤러만 있는 경우에는 프로젝트의 속성에서 종류를 '라이브러리 프로젝트'로 변경할 수 있습니다.
- 라이브러리 프로젝트에서는 라이브러리 부품의 대상 이외의 데이터도 작성할 수 있지만 라이브러리 부품으로 선택 가능한 것은 라이브러리 부품의 대상뿐입니다.
- 라이브러리 부품으로 작성하는 펑션, 펑션 블록 정의, 데이터 형식의 이름에는 여러 라이브러리 간의 이름 중복을 방지하기 위해 이름 공간의 사용을 권장합니다. 이름 공간에 대해서는 '6-7 이름 공간(6-143페이지)'을 참조하십시오.

라이브러리 이용 순서

작성한 라이브러리를 프로젝트에 읽어서 사용자 프로그램에서 이용할 수 있습니다. 다음 순서로 라이브러리를 이용합니다.

1 **라이브러리를 지정합니다.**
라이브러리를 이용하는 프로젝트의 [라이브러리 참조] 창에서 이용할 라이브러리 파일을 지정합니다.

2 **라이브러리 부품을 사용하여 프로그래밍합니다.**
읽은 라이브러리의 라이브러리 부품을 프로젝트 내에서 사용합니다.
사용 방법은 통상적인 펑션, 펑션 블록 정의, 데이터 형식과 동일합니다.

6-9 사용자 프로그램 작성상의 유의 사항

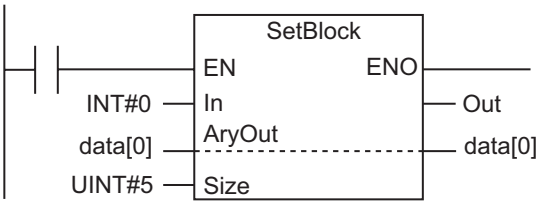
여기에서는 사용자 프로그램 작성상의 유의 사항에 대해 설명합니다.

6-9-1 입력 변수/출력 변수/입출력 변수가 배열 지정 변수인 경우

일부 명령에는 배열 변수를 취급하는 명령이 존재합니다.
예)

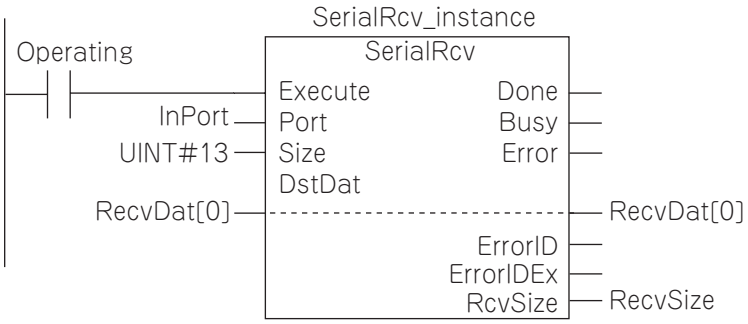
데이터 형식 테이블

변수명	데이터 형식
data	ARRAY[0..4] OF INT



6-9-2 입력 변수/출력 변수/입출력 변수가 구조체 형식 변수인 경우

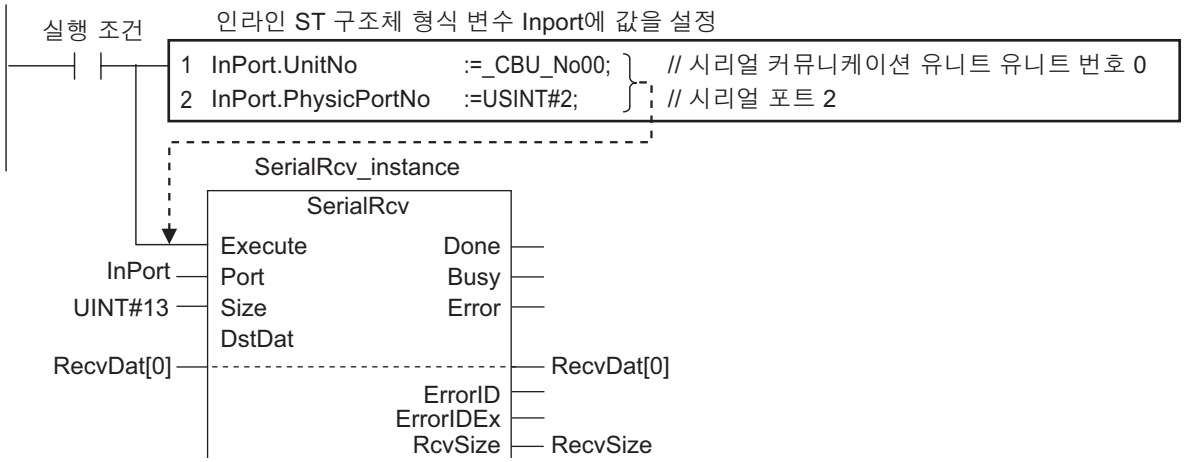
일부 명령에는 입력 변수, 출력 변수, 입출력 변수의 데이터 형식이 구조체 형식인 것이 있습니다.
예)



이 경우에 사전에 입력 파라메터, 출력 파라메터, 입출력 파라메터용에 구조체 형식 변수를 작성하고, 거기에 MOVE 명령 등으로 값을 설정해야 합니다.
예)

변수 테이블

변수명	데이터 형식
InPort	_sPORT



6-9-3 마스터 컨트롤

개요

마스터 컨트롤은 MC(마스터 컨트롤 시작) 명령과 MCR(마스터 컨트롤 종료) 명령 사이에 존재하는 처리를 OFF 출력시키는 기능입니다.

마스터 컨트롤은 비교적 긴 일련의 처리 실행 조건을 제어할 때 편리합니다.

자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 MC/MCR 명령 항을 참조하십시오.

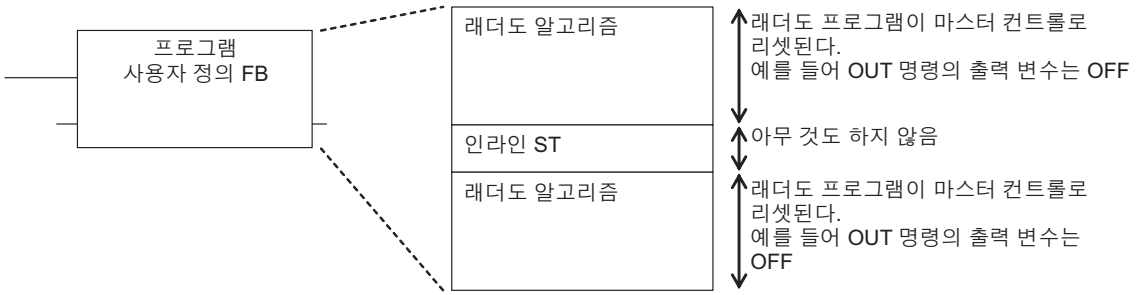
마스터 컨트롤이 사용할 수 있는 프로그램 언어

마스터 컨트롤은 래더도 언어로 사용할 수 있습니다.

ST 언어에서는 사용할 수 없습니다. 래더도 프로그램 중의 인라인 ST에서도 사용할 수 없습니다.

예)

마스터 컨트롤 영역 내:



마스터 컨트롤 리셋 시의 영역 내 명령의 동작

마스터 컨트롤 리셋 시의 영역 내 명령의 동작에 대해서는 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 'MC/MCR 명령' 항을 참조하십시오.

7

동작 확인 및 운전

이 장에서는 오프라인 디버그를 포함한 동작 확인 내용과 동작 확인에서 운전까지의 순서를 설명합니다.

7-1 동작 확인 및 운전 순서의 개요 7-2

7-2 오프라인 디버그 7-3

7-2-1 시뮬레이션 기능의 특징 7-3
7-2-2 시뮬레이션 기능의 실행 7-3
7-2-3 시뮬레이션 기능의 설정 7-6

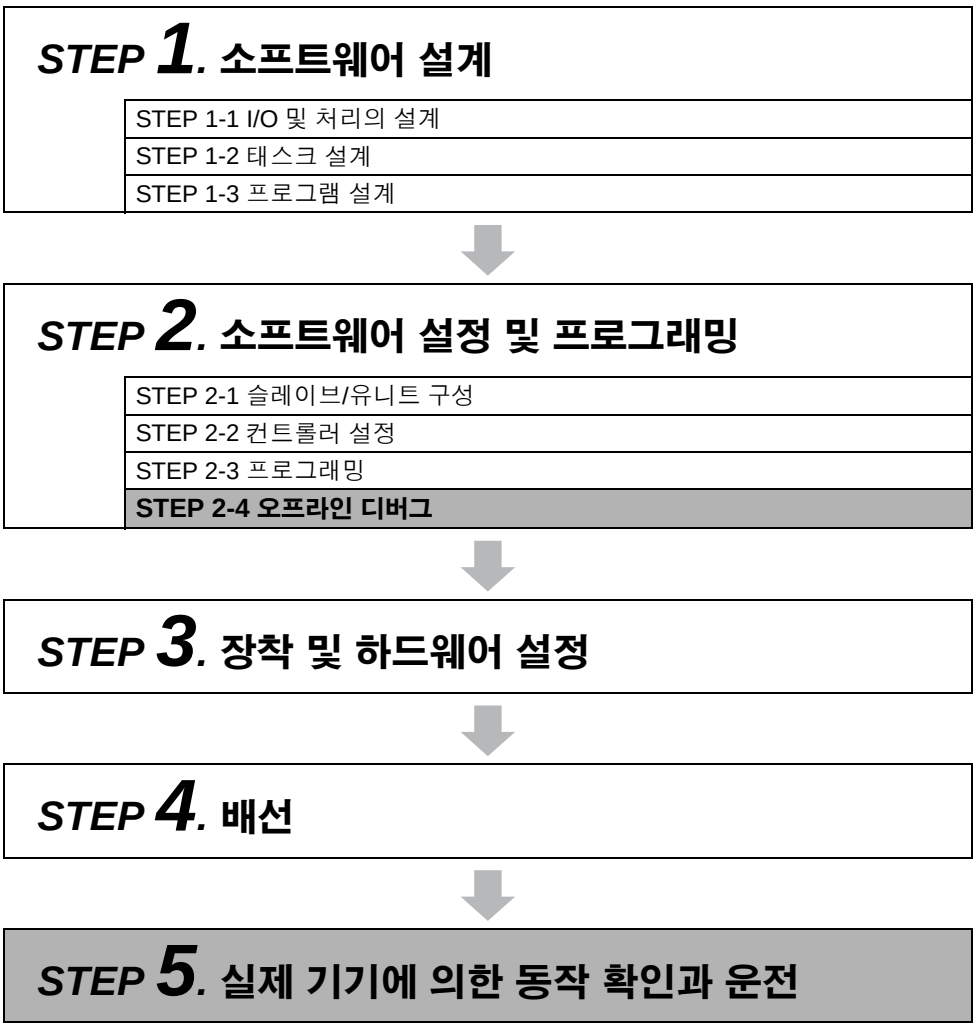
7-3 실제 기기에서의 동작 확인 및 운전 7-8

7-3-1 실제 기기에서의 동작 확인에서 운전 시작까지의 순서 7-8
7-3-2 프로젝트 전송 7-9
7-3-3 I/O의 배선 확인 7-9
7-3-4 MC 시운전 7-9
7-3-5 사용자 프로그램의 동작 확인 7-10
7-3-6 본 운전의 시작 7-10

7-1 동작 확인 및 운전 순서의 개요

동작 확인 및 운전은 전체 순서 중에 다음 그림의 회색으로 표시한 위치에 대응합니다. STEP 2-4 오프라인 디버그에서는 컨트롤러와 온라인 연결하지 않고 시뮬레이션 기능으로 동작을 확인합니다. STEP 5. 실제 기기에 의한 동작 확인 및 운전에서는 컨트롤러와 온라인 연결하여 실제 기기의 동작을 확인하고, 동작 확인이 완료되면 운전을 시작합니다.

전체 순서는 '1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.



7-2 오프라인 디버그

여기에서는 오프라인 디버그에서 사용하는 시뮬레이션 기능에 대해 설명합니다. 시뮬레이션 기능은 NJ/NX 시리즈의 동작을 PC 상에서 시뮬레이트하면 PC에서만 사용자 프로그램의 동작을 확인하는 기능입니다. 시뮬레이션 기능은 실제 기기 컨트롤러에서는 실현할 수 없는 디버그 기능을 가지므로 사용자 프로그램 개발 및 디버그의 효율을 향상시킬 수 있습니다.

7-2-1 시뮬레이션 기능의 특징

시뮬레이션 기능은 컨트롤러와 온라인 연결하여 동작 확인을 하는 온라인 디버그에 비해 다음과 같은 경우에 유효합니다.

- 브레이크 포인트, 스텝 실행, 일시 정지 등을 사용하여 프로그램의 로직을 확인
- 시뮬레이션 대상을 선택하고, 일부 프로그램만의 동작을 확인
- 시뮬레이션 실행 속도 변경에 의해 실제보다 느린 속도로 동작을 확인
- 태스크 실행 시간 모니터링에 의해 태스크 실행 시간의 예상값을 계측
- 컨트롤러의 외부에서의 입력 등을 디버그 프로그램에서 동작시킴

7-2-2 시뮬레이션 기능의 실행

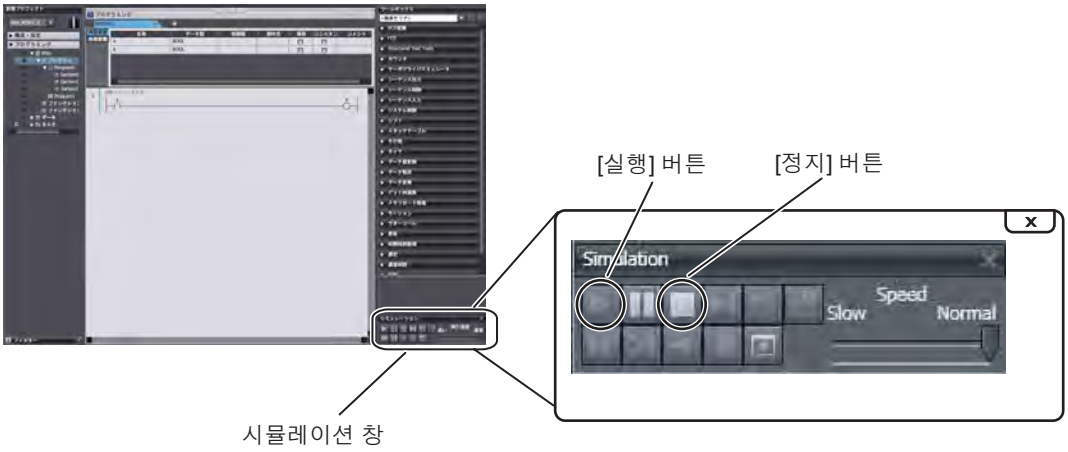
시뮬레이션 기능에서는 다음을 실행할 수 있습니다.

- 시뮬레이터의 기동과 정지
- 프로그램 로직의 확인
- 태스크 실행 시간 예측
- 온라인 연결 시의 디버그 기능

시뮬레이터의 기동과 정지

시뮬레이션 기능은 Sysmac Studio의 시뮬레이션 창에서 시뮬레이터를 기동하여 실행합니다. 시뮬레이션 기능에 의한 동작 확인이 완료되면 시뮬레이터를 정지합니다.
시뮬레이션 실행과 정지 조작의 개요는 다음과 같습니다.

1 **Sysmac Studio에서 메인 메뉴에서 [표시]–[시뮬레이션 창]을 선택합니다.**
툴 화면 우측 아래에 시뮬레이션 창이 표시됩니다.



2 **시뮬레이션 창의 [실행] 버튼을 누릅니다.**
프로그램이 시뮬레이터에 전송되고, 시뮬레이션이 시작됩니다. 시뮬레이션이 시작되면 편집기 등의 Sysmac Studio 화면이 컨트롤러와의 온라인 연결 시와 동일한 상태가 됩니다.

3 **동작 확인이 완료되면 시뮬레이션 창의 [정지] 버튼을 눌러서 시뮬레이터를 정지합니다.**

프로그램 로직의 확인

프로그램 로직의 정당성을 확인하기 위해 시뮬레이터의 동작을 중지하거나 1스텝씩 실행할 수 있습니다. 시뮬레이션 창의 버튼으로 다음 조작을 할 수 있습니다.

조작명	조작 내용
브레이크 포인트	프로그램 상의 지정한 위치에서 프로그램 실행이 일시 정지됩니다.
스텝 실행	ST 언어의 프로그램에서는 1행, 래더도 언어의 프로그램에서는 1명령을 실행하면 시뮬레이터가 일시 정지됩니다.
연속 스텝 실행	일정한 간격으로 연속해서 스텝이 실행됩니다.
일시 정지	시뮬레이션의 실행이 일시 정지됩니다.
스텝 인 실행	펑션 블록 또는 펑션의 소스 내를 스텝 실행됩니다.
스텝 아웃 실행	스텝 인 실행 중인 펑션 블록 또는 펑션이 종료될 때까지 실행됩니다.
1주기 실행	현재 실행하는 태스크가 1주기 분량 실행됩니다. 다음 주기의 프로그램 선두에서 일시 정지됩니다.
조건 브레이크 포인트	브레이크 포인트에 정지 조건을 설정하고, 조건 성립 시에만 프로그램 실행이 일시 정지됩니다.

태스크 실행 시간 예측

시뮬레이터를 실행 시간 예측 모드로 실행하면 태스크의 실행 시작에서 실행 완료까지 걸리는 시간(태스크 실행 시간)의 예측값이 태스크 실행 시간 모니터링에 표시됩니다. 태스크 실행 시간의 평균, 최대 예측값이 표시됩니다. 태스크 실행 시간 모니터링의 표시는 '태스크 실행 시간 모니터링(5-104페이지)'을 참조하십시오.

사용상의 주의 사항

- 하드웨어 개정 번호가 표시되는 유니트에서는 해당하는 하드웨어 개정 번호를 선택하십시오.
- 태스크 실행 시간 예측값은 반드시 실제 기기의 실행 시간과 일치하는 것은 아닙니다. 사용자 프로그램에 기재된 로직의 실행 유무, 외부에서의 입력 조건 및 통신 유무에 따라서는 실제 기기의 실행 시간이 예측 최대값을 초과하는 경우가 있습니다. 따라서 예측값은 태스크 설계의 기준으로 사용하십시오. 반드시 실제 기기 컨트롤러에 연결해서 태스크 실행 시간을 확인한 후에 설계의 검토 및 본 운전을 실행하십시오.

온라인 연결 시의 디버그 기능

시뮬레이터에서는 컨트롤러와의 온라인 연결 시 사용하는 디버그 기능을 일부만 실행할 수 있습니다. 컨트롤러에서의 온라인 디버그와 시뮬레이터에 의한 오프라인 디버그의 디버그 기능의 차이는 다음 표와 같습니다 각 디버그 기능에서의 컨트롤러와 시뮬레이터의 동작의 차이는 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

(○: 사용할 수 있음 ×: 사용할 수 없음)

디버그 기능	컨트롤러	시뮬레이터
모니터링	○	○
감시 창에서의 모니터링	○	○
I/O 맵에서의 모니터링	○	○
미분 모니터링	○	○
접점의 설정/리셋	○	○
접점의 강제값 리프레시 TRUE/FALSE/ 해제	○	○
데이터의 현재값 변경	○	○
메모리 전체 클리어	○	×
상호 참조 팝업	○	○
온라인 편집	○	○
컨트롤러 Status 모니터링	○	×
태스크 실행 상태 모니터링	○	○
축 Status 모니터링(MC 모니터링 테이블)	○	○
동작 모드의 변경	○	×
컨트롤러의 리셋	○	×
데이터 트레이스	○	○
트리거 설정	○	○
샘플링 변수 설정	○	○
트레이스 시작.정지	○	○
트레이스 결과 표시	○	○
트레이스 결과 내보내기	○	○
3D 장치 모델 작성	○	○
디지털·아날로그 차트에서의 표시	○	○
3D 형식으로 궤적 표시	○	○
태스크 실행 시간 모니터링	○	○
실행 처리 시간 예측	×	○
프로그램의 시뮬레이션 디버그	×	○
시뮬레이션 프로그램의 설정	×	○
시뮬레이션 실행 속도의 변경	×	○
브레이크 포인트 설정	×	○
스텝 실행	×	○
트러블 슈팅	○	○
이상 정보 모니터링	○	○
이상 이력 표시	○	○
이벤트 테이블 설정	○	○
사용자 메모리의 사용량 모니터링	○	○
시계 정보 설정	○	×
액세스 권한의 강제 해제	○	×

7-2-3 시뮬레이션 기능의 설정

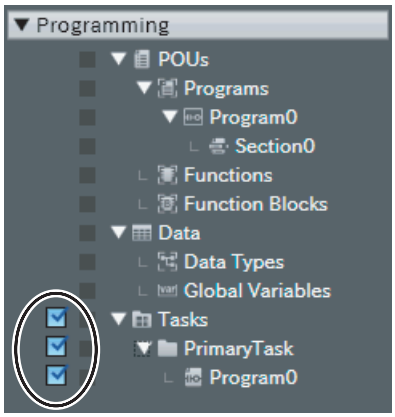
시뮬레이션 기능에서는 다음 설정을 할 수 있습니다.

- 시뮬레이션 프로그램의 설정
- 디버그 프로그램의 설정
- 시뮬레이션 실행 속도 설정

시뮬레이션 프로그램의 설정

시뮬레이션 기능에서는 태스크 단위 또는 프로그램 단위로 시뮬레이션 대상을 설정하여 실행할 수 있습니다. 전체 프로그램 중 시뮬레이션 대상인 일부 프로그램만 시뮬레이터로 작동시킬 수 있습니다.
시뮬레이션 프로그램을 설정하는 조작의 개요는 다음과 같습니다.

1 **시뮬레이션 창을 표시합니다.**
멀티 뷰 탐색기의 [프로그래밍]–[태스크]의 프로그램 좌측에 시뮬레이션 대상 설정용 확인란이 표시됩니다.



2 **태스크 단위 또는 프로그램 단위로 시뮬레이션 대상의 설정용 확인란을 확인합니다.**

디버그 프로그램의 설정

디버그 프로그램은 오프라인 디버그에 의한 동작 확인에서 사용하는 프로그램입니다. 디버그 프로그램에는 컨트롤러의 외부에서의 가상의 입력 처리 및 사용자 이상의 강제 발생 등을 기술합니다. 디버그 프로그램은 시뮬레이터 상에서만 실행할 수 있습니다.

디버그 프로그램을 작성하는 조작의 개요는 다음과 같습니다.

1 **멀티 뷰 탐색기에서 [프로그래밍]–[POU]–[프로그램]을 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [디버그용 추가]–[ST] 또는 [래더]를 선택합니다.**
디버그 프로그램이 새로 작성됩니다.

2 **작성된 디버그 프로그램에 테스트용 프로그램 코드를 기술합니다.**

3 **디버그 프로그램을 태스크에 할당합니다.**

작성된 통상적인 프로그램을 디버그 프로그램으로 변경할 수도 있습니다.

1 **멀티 뷰 탐색기의 [프로그래밍]–[POU]–[프로그램]의 각 프로그램을 마우스 오른쪽 버튼으로 클릭하고 [디버그용 설정]–[유효]로 합니다.**

시뮬레이션 속도의 설정

시뮬레이션 창의 '시뮬레이션 속도 변경 슬라이더'로 0.1~1배 범위에서 PC 상의 실행 속도를 변경할 수 있습니다. 시뮬레이션 실행 중, 정지 중의 어느 쪽으로도 변경할 수 있습니다.
시뮬레이터 실행을 실제보다 느리게 표시하고자 하는 경우에 사용합니다.



7-3 실제 기기에서의 동작 확인 및 운전

여기에서는 실제 기기에서의 동작 확인에서 운전 시작까지의 순서와 조작의 개요에 대해 설명합니다.

7-3-1 실제 기기에서의 동작 확인에서 운전 시작까지의 순서

실제 기기의 동작 확인에서 운전 시작까지의 순서의 개요는 다음과 같습니다.

STEP 1 Sysmac Studio와의 온라인 연결과 프로젝트 다운로드	참조
1. 컨트롤러의 전원을 투입합니다. 2. Sysmac Studio와 온라인 연결합니다.*1 3. Sysmac Studio에서 프로젝트(사용자 프로그램 및 유니트 구성 등의 각종 설정)를 전송합니다.	📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"

*1. NX102 및 NX1P2 CPU 유니트는 페리페럴 USB 포트가 없으므로 USB로 연결할 수 없습니다. Sysmac Studio와의 연결 방법에 대해서는 📖'10-2 Sysmac Studio 연결(10-7페이지)'을 참조하십시오.



STEP 2 실제 기기 동작 확인	참조
1. 프로그램 모드에서 I/O 맵 또는 래더 편집기에서 실제 I/O에 대한 강제값 리프레시 등을 실시하여 I/O의 배선을 확인합니다. 2. 모션 제어를 하는 경우에는 MC 시운전 화면에서 배선 확인, 조그 이송을 통해 모터의 방향 확인, 상대 위치 결정을 통해 이동량(전자 기어의 설정) 확인, 원점 복귀 동작을 확인합니다. 3. 운전 모드로 설정하여 사용자 프로그램의 동작을 확인합니다.	📖'7-3-3 I/O의 배선 확인(7-9페이지)' 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)" 📖'7-3-5 사용자 프로그램의 동작 확인(7-10페이지)'



STEP 3 컨트롤러의 본 운전	참조
1. 설계대로 동작하는지 확인한 후 본 운전을 시작합니다.	–



참고

Sysmac Studio에서 프로젝트를 다운로드하려면 '동기 기능'에서 PC에서 컨트롤러에 대한 전송을 실행합니다. '동기'의 자세한 내용은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

7-3-2 프로젝트 전송

Sysmac Studio에서 실제 기기 컨트롤러로 프로젝트를 전송합니다.

1 **컨트롤러와 온라인 연결한 후 [컨트롤러] 메뉴에서 [동기]를 선택합니다.**
PC 상의 데이터와 실제 기기 컨트롤러의 데이터가 자동 비교됩니다.

2 **[전송[PC→ 컨트롤러]] 버튼을 클릭합니다.**

7-3-3 I/O의 배선 확인

Sysmac Studio의 감시 창 등에서 강제값 리프레시를 실행하여 I/O의 배선을 확인합니다. I/O의 배선은 배선을 확인하는 슬레이브 또는 유니트의 I/O에 대해 값을 쓰고, 결과를 모니터링하여 확인할 수 있습니다.

강제값 리프레시의 자세한 내용은 📖'8-6-1 강제값 리프레시(8-47페이지)'를 참조하십시오.

7-3-4 MC 시운전

MC 시운전은 사용자 프로그램 없이 Sysmac Studio에서 주로 다음을 실행할 수 있습니다.

- 배선 확인: 서보 드라이버 연결 커넥터의 입출력 신호의 모니터링,
  서보 드라이버의 상태 모니터링
- 모터의 동작 및 방향 확인: 서보ON, 조그 이송
- 전자 기어의 설정 확인: 상대값 위치 결정을 실행하고, 이동량 확인/변경
- 원점 복귀 확인: 원점 복귀 동작의 확인

Sysmac Studio에서 컨트롤러에 온라인 연결하여 MC 시운전 화면에서 실행합니다.

자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

다음 조작을 실행합니다.

1 **필요한 배선을 한 후에 Sysmac Studio에서 컨트롤러에 온라인 연결합니다.**

2 **'축' 작성, 축 할당, 다음 축 파라메터 설정을 합니다.**
MC 시운전에 필요한 축 파라메터 설정:
표시 단위 선택, 모터 1회전당 펄스 수 및 이동량
최고 속도, 조그 최고 속도, 최대 가속도·감속도,
소프트웨어 리미트 기능 선택, 소프트웨어 리미트, 카운트 모드

3 **MC 시운전 화면을 열고 다음을 실행합니다.**
예)
- 모니터링·배선 확인
- 조그 이송으로 모터의 방향 확인
- 상대 위치 결정에서 이동량(전자 기어 설정) 확인
- 원점 복귀 동작 확인

7-3-5 사용자 프로그램의 동작 확인

실제 기기를 사용한 사용자 프로그램의 동작 확인은 CPU 유니트의 동작 모드를 운전 모드로 하여 실행합니다. 동작 확인에는 다음과 같은 항목이 있습니다.

- 사용자 프로그램의 동작을 확인
- 온라인 편집으로 사용자 프로그램을 수정
- 데이터 트레이스로 사용자 프로그램 동작을 확인

사용자 프로그램의 동작을 확인

사용자 프로그램의 동작 확인은 다음 순서로 실행합니다.

1 **사용자 프로그램의 실행 상태를 모니터링**

2 **접점 및 코일의 상태, 변수의 값을 변경하고 동작 확인**

- **사용자 프로그램의 실행 상태를 모니터링**
  컨트롤러의 접점 및 코일의 ON/OFF 상태, 변수의 현재값을 모니터링합니다. 모니터링은 Sysmac Studio의 래더 편집기, 감시 창, I/O 맵에서 실행할 수 있습니다.

- **접점 및 코일의 상태, 변수의 값을 변경하고 동작 확인**
  사용자 프로그램의 접점 및 코일의 ON/OFF 상태, 변수의 값을 변경하고, 설계대로 동작이 가능한지 모니터링으로 확인합니다. 접점 및 코일의 상태를 변경하려면 '강제값 리프레시', 변수의 값을 변경하려면 '현재값 변경'을 사용합니다.
  자세한 내용은 📖'8-6-1 강제값 리프레시(8-47페이지)', 📖'8-6-2 현재값 변경(8-52페이지)'을 참조하십시오.

온라인 편집으로 사용자 프로그램을 수정

동작 확인에 의해 수정이 필요하다고 판단된 사용자 프로그램을 '온라인 편집'으로 수정합니다. '온라인 편집'을 사용하면 CPU 유니트의 운전을 정지하지 않고 사용자 프로그램을 변경할 수 있습니다.
자세한 내용은 📖'8-6-3 온라인 편집(8-54페이지)'을 참조하십시오.

데이터 트레이스로 동작 확인

'데이터 트레이스'를 사용하여 접점 및 코일이 ON/OFF하는 타이밍 및 변수 값의 변동 등을 확인합니다.
자세한 내용은 📖'8-6-4 데이터 트레이스(8-56페이지)'를 참조하십시오.

7-3-6 본 운전의 시작

동작 모드를 운전 모드로 하여 본 운전을 시작합니다. 작성한 사용자 프로그램 및 각종 데이터나 설정값을 충분히 동작 확인한 후에 본 운전을 실시하십시오.

8

CPU 내장 기능

이 장에서는 CPU 유니트에서 사용할 수 있는 기능에 대해 설명합니다.

8-1 데이터 관리·시계·운전 관련 기능 8-3
8-1-1 메모리 전체 클리어 8-3
8-1-2 시계 기능 8-3
8-1-3 운전 중 출력 기능 8-6

8-2 NX 유니트 관련 관리 기능 8-7
8-2-1 NX 버스 기능 모듈 8-7
8-2-2 CPU 유니트 상의 NX 유니트 장착 설정 기능 8-12
8-2-3 CPU 유니트 상의 NX 유니트 관련 재시작 기능 8-15
8-2-4 CPU 유니트 상의 NX 유니트 배선 확인 8-16
8-2-5 CPU 유니트 상의 NX 유니트 FAIL-SOFT 동작 기능 8-17
8-2-6 CPU 유니트 상의 NX 유니트 통전 시간 모니터링 기능 8-21

8-3 CJ 유니트 관련 관리 기능 8-22
8-3-1 기본 I/O 유니트 관련 8-22
8-3-2 고기능 유니트 관련 8-23

8-4 SD 메모리 카드 기능 8-25
8-4-1 SD 메모리 카드 기능 목록 8-25
8-4-2 지원하는 SD 메모리 카드 및 폴더/파일 사양 8-26
8-4-3 SD 메모리 카드 조작 명령 8-28
8-4-4 FTP 클라이언트 통신 명령 8-28
8-4-5 FTP 서버 기능 8-29
8-4-6 Sysmac Studio에서의 파일 조작 8-29
8-4-7 SD 메모리 카드의 수명 감지 기능 8-29
8-4-8 SD 메모리 카드 관련 시스템 정의 변수 목록 8-30
8-4-9 SD 메모리 카드의 자체 진단 기능 8-31
8-4-10 SD 메모리 카드 내의 파일 액세스의 배타 제어 8-33

8-5 보안 기능 8-34
8-5-1 사용자 프로그램 실행용 ID 인증 기능 8-35
8-5-2 사용자 프로그램 복원 정보 없는 전송 기능 8-38
8-5-3 프로젝트 파일 전체 보호 기능 8-39
8-5-4 데이터 보호 기능 8-40
8-5-5 조작 권한 인증 기능 8-42
8-5-6 CPU 유니트에 대한 쓰기 방지 기능 8-44
8-5-7 CPU 유니트 이름 기능 및 시리얼 ID 기능 8-45

8-6 디버그 기능 8-47
8-6-1 강제값 리프레시 8-47
8-6-2 현재값 변경 8-52
8-6-3 온라인 편집 8-54
8-6-4 데이터 트레이스 8-56
8-6-5 미분 모니터링 8-61
8-7 이벤트 로그 기능 8-66
8-7-1 개요 8-66
8-7-2 이벤트 로그 기능의 세부 내용 8-68
8-7-3 컨트롤러 이벤트(컨트롤러 이상/정보) 8-73
8-7-4 사용자 이벤트(사용자 이상/정보) 8-74
8-8 이벤트 중요도 변경 기능 8-80
8-8-1 이벤트 중요도 변경의 용도 8-80
8-8-2 이벤트 중요도 변경 가능한 이벤트 8-80
8-8-3 이벤트 중요도 변경 조작 방법 8-81

8-1 데이터 관리·시계·운전 관련 기능

여기에서는 데이터 관리 기능, 시계 관련, 운전 관련 기능에 대해 설명합니다.

8-1-1 메모리 전체 클리어

Sysmac Studio에서 CPU 유니트 내의 사용자 프로그램, 컨트롤러 구성·설정, 변수, 앱솔루트 엔코더 원점 위치 오프셋을 공장 출하 상태로 초기화할 수 있습니다. 이 조작을 '메모리 전체 클리어'라고 합니다.

사용상의 주의 사항

- 메모리 전체 클리어는 프로그램 모드 시에만 실행할 수 있습니다.
- 보안 기능의 'CPU 유니트에 대한 쓰기 방지'가 설정된 경우에 실행할 수 없습니다.
- 메모리 전체 클리어 중에 CPU 유니트의 전원을 OFF하지 마십시오.

메모리 전체 클리어 후에는 공장 출하 상태의 CPU 유니트에서 시스템 구성을 작성한 직후와 동일한 동작이 됩니다.

- **Sysmac Studio를 사용한 조작**

  Sysmac Studio를 CPU 유니트에 온라인 연결하여 [컨트롤러] 메뉴에서 [메모리 전체 클리어]를 선택합니다.

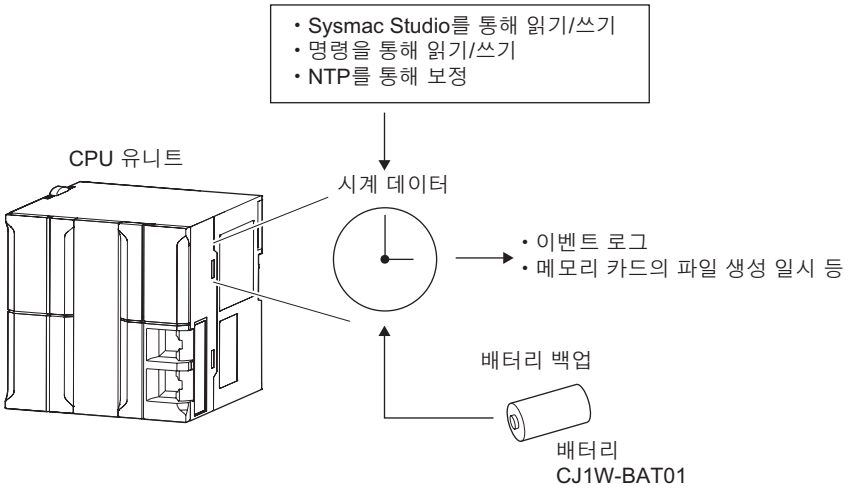
  조작의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

8-1-2 시계 기능

개요

CPU 유니트에는 시계(RTC)가 내장되어 있습니다.
이 시계의 시각 데이터가 이벤트 로그 및 SD 메모리 카드의 파일 생성 일시 등에 반영됩니다.
다음과 같은 기능이 있습니다.

- Sysmac Studio에서 읽기/쓰기
- 명령에서 읽기/쓰기
- 시스템 정의 변수에서 읽기(쓰기는 불가능)
- NTP에 의한 시계 보정



사용상의 주의 사항

- 시계의 시각 데이터는 전원 OFF 시, 배터리로 유지되므로 전원 투입 시의 시각 데이터는 불안정해집니다. 또한, 전원 OFF 시에도 전원 투입 후에 EtherNet/IP 네트워크에서 NTP에 의해 시각 데이터를 다시 설정하면 시계 자체는 가능합니다.
- NX102 및 NX1P2에서 배터리 없이 운전할 때는 내장 커패시터로 시계의 시각 데이터가 유지됩니다. 내장 커패시터에서의 유지 기간을 초과해서 장치의 전원을 OFF하면 시각 데이터가 초기값이 됩니다. 이벤트 로그 등으로 시각 데이터를 사용하는 경우에는 전원을 투입할 때마다 시각 데이터를 설정하십시오.

- **시각 데이터의 범위**
  - NX 시리즈 CPU 유니트의 경우 : 1970-01-01~2069-12-31
  - NJ 시리즈 CPU 유니트의 경우 : 1970-01-01~2106-02-06

- **표준 시간대와 현지 시간 설정**

  처음 사용할 때, 표준 시간대와 현지 시간(시각 데이터)을 등록하십시오.
  표준 시간대와 현지 시간은 Sysmac Studio의 시계 설정 화면에서 설정합니다.
  EtherCAT 슬레이브 및 CJ 유니트가 CPU 유니트에서 읽는 시각 데이터 및 설정하는 시각 데이터는 표준 시간대에 맞춰 현지 시간이 됩니다.

참고

배터리가 장착되지 않은 경우 또는 배터리 수명이 다한 경우에 표준 시간대의 설정 자체는 유지되지만 시각 데이터는 유지되지 않고 잘못된 값이 됩니다.

시각 데이터 설정 방법

다음과 같은 방법이 있습니다.

- **Sysmac Studio에서 변경**

  Sysmac Studio를 사용하여 PC의 시각에 동기시키는 형식으로 내장 시계의 시각 데이터를 설정할 수 있습니다.

- **명령에서 변경**

  SetTime 명령을 사용하여 시각 데이터를 설정할 수 있습니다.

- **NTP 서버에서 변경**
  EtherNet/IP 상의 NTP 서버를 사용하여 시각 데이터를 설정할 수 있습니다.

NTP에 의한 시계 보정

- **목적**
  네트워크 시스템에서는 시스템 전체에서 시각 데이터가 공통이어야 합니다. 이를 간단하게 맞출 수 있는 구조로 NTP 기능을 제공합니다.

- **사양**
  NTP 클라이언트 기능을 지원합니다.
  자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.

시각 데이터 읽는 방법

시각 데이터가 잘못된 경우에는 그대로 잘못된 값을 읽습니다.

- **명령에서 읽기**
  사용자 프로그램 내에서는 GetTime 명령을 사용하여 시각 데이터를 읽을 수 있습니다.

- **시스템 정의 변수에서 읽기(쓰기는 불가능)**
  다음과 같은 시스템 정의 변수를 사용하여 시각 데이터를 읽을 수 있습니다.
  '_CurrentTime'(시스템 시각)

- **Sysmac Studio를 사용하는 읽기**
  Sysmac Studio를 사용하여 [컨트롤러] 메뉴의 [컨트롤러 시계]에서 시각 데이터를 읽어서 표시할 수 있습니다.

기록

시계 데이터를 변경한 경우에 이벤트 로그에 기록됩니다.
단, NTP에 의해 보정된 경우에 이벤트 로그에는 기록되지 않습니다.

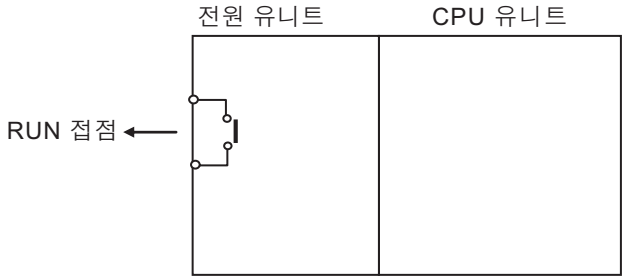
관련 시스템 정의 변수

변수명	이름	기능	데이터 형식	R/W
_CurrentTime	시스템 시각	CPU 유니트의 내부 시계 데이터가 저장됩니다.	DATE_AND_TIME	R

8-1-3 운전 중 출력 기능

개요

전원 유니트 NX-PA9001, NX-PD7001, NJ-P□3001에서 CPU 유니트의 운전 중임을 전달하는 RUN 접점(운전 중 출력) 신호가 출력됩니다.



RUN 접점(운전 중 출력)은 다음과 같이 동작합니다.

상태	동작
운전 중(운전 모드)	ON
기동 중 (전원 투입 후, 전원 투입 시의 동작 모드 설정에 따른 운전 상태로 이행할 때까지)	OFF
운전 정지 중(프로그램 모드)	
전체 정지 이상 레벨의 컨트롤러 이상 발생 중	

전원 유니트 NX-PA9001, NX-PD7001, NJ-P□3001의 RUN 접점(운전 중 출력)의 정격은 다음과 같습니다.

항목	내용
접점 구성	1a
개폐 능력	AC250V 2A(저항 부하) AC120V 0.5A(유도 부하) DC24V 2A(저항 부하)

용도

- CPU 유니트가 정상적으로 동작하고, 운전되는지 상위에 통지하는 생존 신호
- 복수 CPU 유니트의 기동 완료의 동기
- CPU 유니트의 운전 시작에 의한 인터록 해제

안전상의 중요 사항

전원 투입부터 운전 모드까지 이행하여 10~20초 정도 걸립니다. 그 동안의 출력은 슬레이브 및 유니트의 사양에 따릅니다.
전원 유니트 운전 중 출력을 이용하는 등 외부 기기가 오작동하지 않도록 안전 보장 회로를 만드십시오.

8-2 NX 유니트 관련 관리 기능

여기에서는 NX102 및 NX1P2 CPU 유니트 상의 NX 유니트 관련 관리 기능에 대해 설명합니다.
CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.

8-2-1 NX 버스 기능 모듈

NX 버스 기능 모듈은 NX 버스의 마스터(이후, NX 버스 마스터로 생략)로 CPU 유니트의 NX 버스에 연결한 NX 유니트의 이벤트 로그 관리, Status 관리, I/O 리프레시 등의 처리를 실행합니다.

NX 버스 기능 모듈이 취급하는 I/O 데이터에는 NX 버스 기능 모듈이 NX 버스 마스터로서 관리하는 NX 유니트의 Status 및 NX 유니트별 I/O 데이터의 2종류가 있습니다. 모두 변수를 할당할 수 있는 I/O 포트입니다.

이러한 I/O 데이터에 액세스하는 변수로 I/O 포트에 할당한 디바이스 변수 및 NX 버스 기능 모듈의 시스템 정의 변수의 2종류가 있습니다.

다음에는 NX 버스 마스터로서 관리하는 NX 유니트의 Status, NX 유니트별 I/O 데이터, I/O 포트에 대한 디바이스 변수 할당 및 디바이스 변수를 사용하는 프로그램 예시에 대해 설명합니다.

NX 버스 마스터로서 관리하는 NX 유니트의 Status

NX 버스 기능 모듈이 NX 버스 마스터로서 관리는 NX 유니트의 Status에는 I/O 포트에 할당한 디바이스 변수만으로 액세스할 수 있는 Status, 시스템 정의 변수만으로 액세스할 수 있는 Status, 디바이스 변수와 시스템 정의 변수의 양쪽에서 액세스할 수 있는 Status가 있습니다.

- **NX 버스 마스터로서 관리하는 NX 유니트의 Status 목록**

이름	I/O 포트	시스템 정의 변수
NX 유니트 등록 Status	NX Unit Registration Status	_NXB_UnitRegTbl
NX 유니트 메시지 통신 가능 Status	NX Unit Message Enabled Status	_NXB_UnitMsgActiveTbl
NX 유니트 I/O 데이터 통신 중 Status	NX Unit I/O Data Active Status	_NXB_UnitIOActiveTbl
NX 유니트 이상 Status	NX Unit Error Status	_NXB_UnitErrFlagTbl
입력 동기 시각 정보	Time Stamp of Synchronous Input	–
출력 동기 시각 정보	TimeStamp of Synchronous Output	–
NX 버스 기능 모듈 이상 Status	–	_NXB_ErrSta
NX 버스 기능 모듈 마스터 이상 Status	–	_NXB_MstrErrSta
NX 버스 기능 모듈 유니트 이상 Status	–	_NXB_UnitErrStaTbl

● NX 버스 마스터로서 관리하는 NX 유니트 Status의 내용

이름	내용
NX 유니트 등록 Status	NX 유니트가 유니트 구성 정보에 등록되었는지 여부를 나타냅니다. 각 비트의 의미는 다음과 같습니다. TRUE: 등록 있음 FALSE: 등록 없음 유니트 구성 정보가 등록된 경우에는 등록된 유니트가 TRUE가 됩니다. 유니트 구성 정보가 자동 생성(실제 유니트 구성 정보만으로 유니트 구성 정보의 등록 없음)인 경우에는 모든 유니트가 FALSE가 됩니다. 미장착 유니트로 설정된 NX 유니트는 'TRUE'입니다. 각 비트는 다음 타이밍에 갱신됩니다. • [유니트 구성 정보가 등록된 경우] FALSE → TRUE: 시스템 기동 시 TRUE→ FALSE: 구성 정보 해제 시 • [유니트 구성 정보가 자동 생성인 경우] FALSE → TRUE: 구성 정보 확정 시 TRUE→ FALSE: 없음(유니트 구성 정보가 자동 생성인 경우에는 항상 FALSE)
NX 유니트 메시지 통신 가능 Status	NX 유니트가 메시지 통신 가능한지 여부를 나타냅니다. 각 비트의 의미는 다음과 같습니다. TRUE: 메시지 통신 가능 FALSE: 메시지 통신 불가능 다음 조건을 충족하는 NX 유니트를 '메시지 통신 가능'으로 합니다. • 비교 일치함(유니트 구성 정보가 등록된 경우만 해당) • NX 유니트가 WDT 이상이 아님 미장착 유니트로 설정된 NX 유니트는 'FALSE'가 됩니다. 각 비트는 해당NX 유니트의 메시지 통신 가능 상태가 변화했을 때 갱신됩니다.
NX 유니트 I/O 데이터 통신 중 Status	NX 유니트의 I/O 데이터 통신이 가능한지 여부를 나타냅니다. 각 비트의 의미는 다음과 같습니다. TRUE: NX 유니트의 I/O 데이터를 제어에 사용할 수 있음 FALSE: NX 유니트의 I/O 데이터를 제어에 사용할 수 없음 미장착 유니트로 설정된 NX 유니트는 'FALSE'가 됩니다. 각 비트는 해당 NX 유니트의 동작 상태가 변화했을 때 갱신됩니다. NX 유니트 등록 Status와 이 Status의 양쪽이 TRUE인 경우에 대상의 NX 유니트는 정상적으로 동작합니다.
NX 유니트 이상 Status	NX 유니트에서 이상이 발생했는지 여부를 나타냅니다. 각 비트의 의미는 다음과 같습니다. TRUE: 이상 있음 FALSE: 이상 없음 각 비트가 TRUE가 되는 것은 이상의 중요도가 다음과 같은 경우입니다. • 전체 정지 이상 • 부분 정지 이상 • 가벼운 이상 • 감시 정보 미장착 유니트로 설정된 NX 유니트는 'FALSE'가 됩니다. 각 비트는 다음 타이밍에 갱신됩니다. FALSE → TRUE: 이상 발생 시 TRUE → FALSE: 이상 해제 시 이상 요인이 제거된 경우에도 사용자에 의해 이상이 해제될 때까지는 FALSE가 되지 않습니다.
입력 동기 시각 정보	입출력 동기 리프레시 방식의 NX 유니트가 동기 입력한 타이밍의 시간 정보입니다. 단위는 ns입니다.

이름	내용
출력 동기 시각 정보	입출력 동기 리프레시 방식의 NX 유니트가 동기 출력한 타이밍의 시간 정보입니다. 단위는 ns입니다.
NX 버스 기능 모듈 이상 Status	NX 버스 기능 모듈의 이상 Status입니다. 'NX 버스 기능 모듈 마스터 이상 Status'와 모든 NX 유니트의 'NX 버스 기능 모듈 유니트 이상 Status'를 집약한 Status입니다.
NX 버스 기능 모듈 마스터 이상 Status	CPU 유니트의 NX 버스 기능 모듈에서 감지한 이상 Status입니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.
NX 버스 기능 모듈 유니트 이상 Status	CPU 유니트 상의 NX 유니트에서 감지한 이상 Status입니다. 이 Status는 WORD 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.

관련 시스템 정의 변수의 자세한 내용은 'A-7-4 NX 버스 기능 모듈(카테고리명: _NXB)(A-139페이지)'을 참조하십시오.

- **NX 버스 마스터로서 관리하고 있는 NX 유니트의 Status의 갱신 타이밍**

이름	NX 유니트의 I/O 포트	시스템 정의 변수
NX 유니트 등록 Status	태스크 실행과 비동기	태스크 실행과 비동기
NX 유니트 메시지 통신 가능 Status	태스크 실행과 비동기	태스크 실행과 비동기
NX 유니트 I/O 데이터 통신 중 Status	프라이머리 정주기 태스크와 동기	프라이머리 정주기 태스크와 동기
NX 유니트 이상 Status	프라이머리 정주기 태스크와 동기	프라이머리 정주기 태스크와 동기
입력 동기 시각 정보	프라이머리 정주기 태스크와 동기	−
출력 동기 시각 정보	프라이머리 정주기 태스크와 동기	−
NX 버스 기능 모듈 이상 Status	−	태스크 실행과 비동기
NX 버스 기능 모듈 마스터 이상 Status	−	태스크 실행과 비동기
NX 버스 기능 모듈 유니트 이상 Status	−	태스크 실행과 비동기

NX 유니트별 I/O 데이터

NX 유니트의 형식 및 기능별 고유의 데이터입니다 NX 유니트의 I/O 포트에 할당한 디바이스 변수에서만 액세스할 수 있습니다.

NX 유니트별 I/O 데이터의 자세한 내용은 각 NX 유니트의 사용 설명서를 참조하십시오.

I/O 포트에 디바이스 변수 할당

Sysmac Studio에서 유니트 구성 정보를 작성하면 NX 버스 마스터로 관리하는 NX 유니트의 Status 및 CPU 유니트 상에 장착한 NX 유니트의 I/O 데이터가 I/O 포트로 자동 등록됩니다.
이러한 Status 및 I/O 데이터의 I/O 포트에 할당한 변수가 디바이스 변수입니다.

● I/O 포트명

NX 버스 마스터로서 관리하는 NX 유니트의 Status는 NX 유니트별로 다음 6 종류의 I/O 포트명이 됩니다.

이름	I/O 포트명	데이터 형식
NX 유니트 등록 Status	장치명+NX Unit Registration Status	BOOL
NX 유니트 메시지 통신 가능 Status	장치명+NX Unit Message Enabled Status	BOOL
NX 유니트 I/O 데이터 통신 중 Status	장치명+NX Unit I/O Data Active Status	BOOL
NX 유니트 이상 Status	장치명+NX Unit Error Status	BOOL
입력 동기 시각 정보	장치명+Time Stamp of Synchronous Input	ULINT
출력 동기 시각 정보	장치명+TimeStamp of Synchronous Output	ULINT

예) 장치명이 'N1'인 'NX 유니트 등록 Status'인 경우:
N1 NX Unit Registration Status
예) 장치명이 'N2'인 '입력 동기 시각 정보'인 경우:
N2 Time Stamp of Synchronous Input

CPU 유니트 상에 장착한 NX 유니트의 I/O 데이터의 I/O 포트명은 NX 유니트의 형식 및 기능별 고유명이 됩니다.

예) 디지털 입력 유니트인 경우:
Input Bit 00
예) 아날로그 출력 유니트인 경우:
Ch1 Analog Output Value

● 디바이스 변수 등록

Sysmac Studio의 I/O 맵에서 I/O 포트에 디바이스 변수를 할당합니다. 그에 따라 디바이스 변수를 변수 테이블에 등록합니다.

Sysmac Studio에서 디바이스 변수를 등록하는 방법은 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.

● 디바이스 변수의 속성

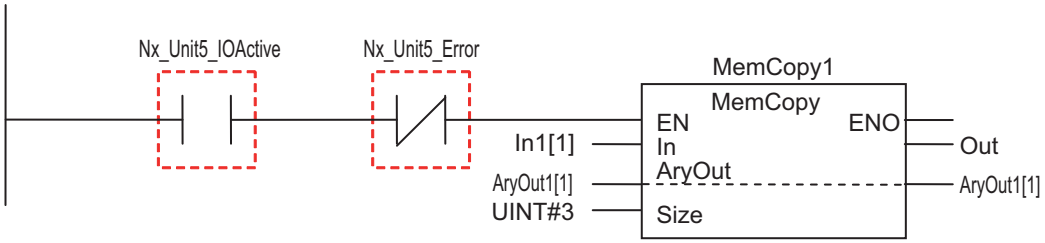
디바이스 변수의 속성에 대해서는 '디바이스 변수의 속성(3-12페이지)'을 참조하십시오.

디바이스 변수를 사용하는 프로그램 예

여기에서는 NX 유니트 디바이스 변수를 사용하는 경우의 프로그램 예를 설명합니다.

- **NX 유니트별로 I/O 데이터의 유효성을 판단**

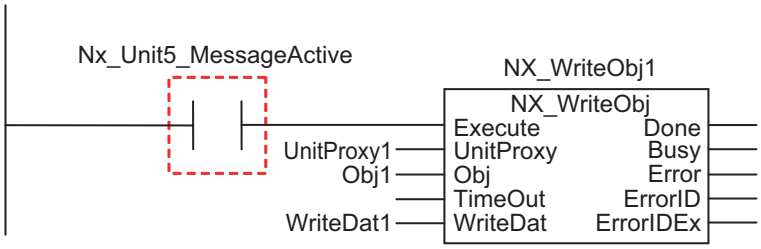
  CPU 유니트 상의 NX 유니트(NX 유니트 번호: 5)의 I/O 데이터 유효성을 판단하기 위한 샘플 프로그램은 다음과 같습니다. 'NX 유니트 I/O 데이터 통신 중 Status' 및 'NX 유니트 이상 Status'를 사용합니다. I/O 데이터가 유효한 경우에 NX 유니트(NX 유니트 번호: 5)의 I/O 데이터를 조작합니다.



Nx_Unit5_IOActive: 유니트 번호가 5인 NX 유니트 I/O 데이터 통신 중 Status를 나타내는 디바이스 변수
Nx_Unit5_Error: 유니트 번호가 5의 NX 유니트 이상 Status를 나타내는 디바이스 변수

- **NX 유니트별로 메시지 통신 가능한지 여부를 판단**

  CPU 유니트 상의 NX 유니트(NX 유니트 번호: 5)의 메시지 통신 가능 여부를 확인하기 위한 샘플 프로그램은 다음과 같습니다. 'NX 유니트 메시지 통신 가능 Status'를 사용합니다.
  메시지 통신이 가능한 경우에 NX 유니트(NX 유니트 번호: 5)에 NX 개체 쓰기 명령으로 메시지를 송신합니다.



Nx_Unit5_MessageActive: 유니트 번호가 5인 NX 유니트 메시지 통신 가능 Status를 나타내는 디바이스 변수

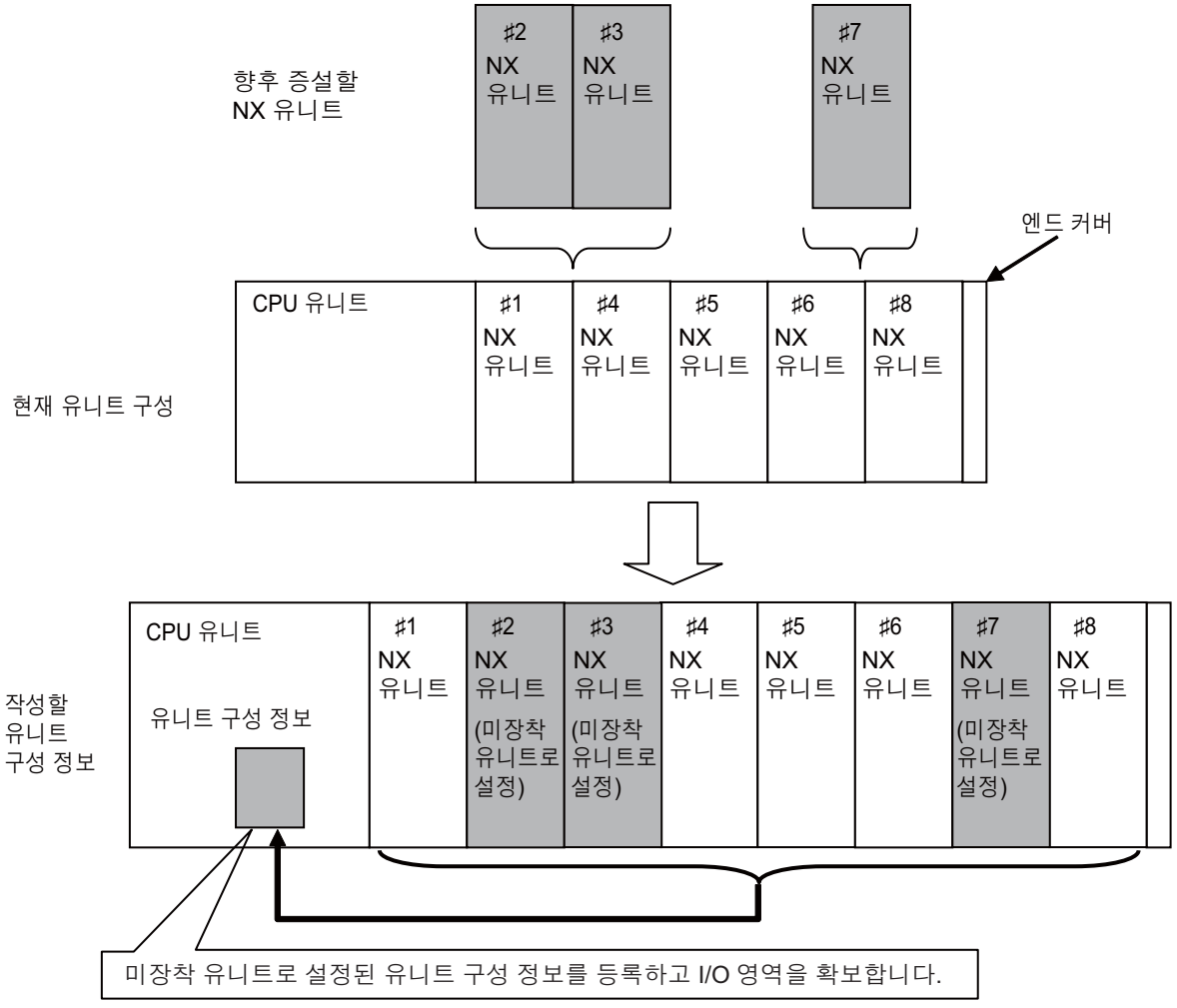
8-2-2 CPU 유니트 상의 NX 유니트 장착 설정 기능

여기에서는 CPU 유니트 상의 NX 유니트 장착 설정 기능에 대해 설명합니다.

기능의 개요

향후 추가할 예정이 있는 NX 유니트를 미장착 유니트로 유니트 구성 정보에 등록할 수 있습니다. 이것으로 실제 구성에서는 장착하지 않은 NX 유니트에 대해 사전에 프로그램을 작성해둘 수 있습니다.
장치 셋업 시 등 일시적으로 특정 NX 유니트가 입수되지 않는 경우 등에도 이 기능을 사용할 수 있습니다.

- 실제 구성에서 장착하는 NX 유니트와 동일하게 미장착 NX 유니트의 I/O 영역도 확보됩니다.
- 미장착 NX 유니트에도 NX 유니트 번호가 할당됩니다. 따라서 미장착으로 설정했던 NX 유니트를 실제 구성에서 장착하는 설정으로 변경해도 CPU 유니트 상의 다른 NX 유니트의 NX 유니트 번호는 바뀌지 않습니다.



미장착 유니트로 설정된 NX 유니트의 동작 사양

미장착 유니트에 설정된 NX 유니트의 동작 사양은 다음 표와 같습니다.

항목	동작
EtherCAT 마스터와 리프레시 데이터의 대역 확보 (I/O 영역의 확보)	확보됩니다.
EtherCAT 마스터와의 I/O 리프레시	리프레시하지 않습니다.
이벤트 검출	검출하지 않습니다.
NX 유니트에 대한 NX 유니트 번호의 배포	대상이 존재하지 않으므로 배포되지 않습니다.
메시지 통신	대상이 존재하지 않으므로 불가능합니다.
Sysmac Studio의 동기 기능에 의한 전송	대상 외입니다.
유니트 동작 설정의 전송	대상 외입니다.
Sysmac Studio 컨트롤러 백업 기능	대상 외입니다.
SD 메모리 카드 백업 기능	대상 외입니다.
명령	파라메터의 읽기 및 쓰기는 불가능합니다. 명령 에러가 됩니다.
메모리 전체 클리어	대상 외입니다.
백업/복원 조작에 의한 설정 정보의 읽기/쓰기	대상 외입니다.
이벤트 로그 읽기	대상 외입니다.
Status 정보의 통지	대상 외입니다.

Sysmac Studio에서의 유니트 구성 작성 시의 NX 유니트의 소비 전력 계산, 유니트 폭 계산에는 미장착 유니트로 설정된 NX 유니트를 포함해서 계산합니다.



안전상의 중요 사항

작성한 사용자 프로그램 및 각종 데이터나 설정값은 동작을 충분히 확인한 후에 본 운전을 실시하십시오.

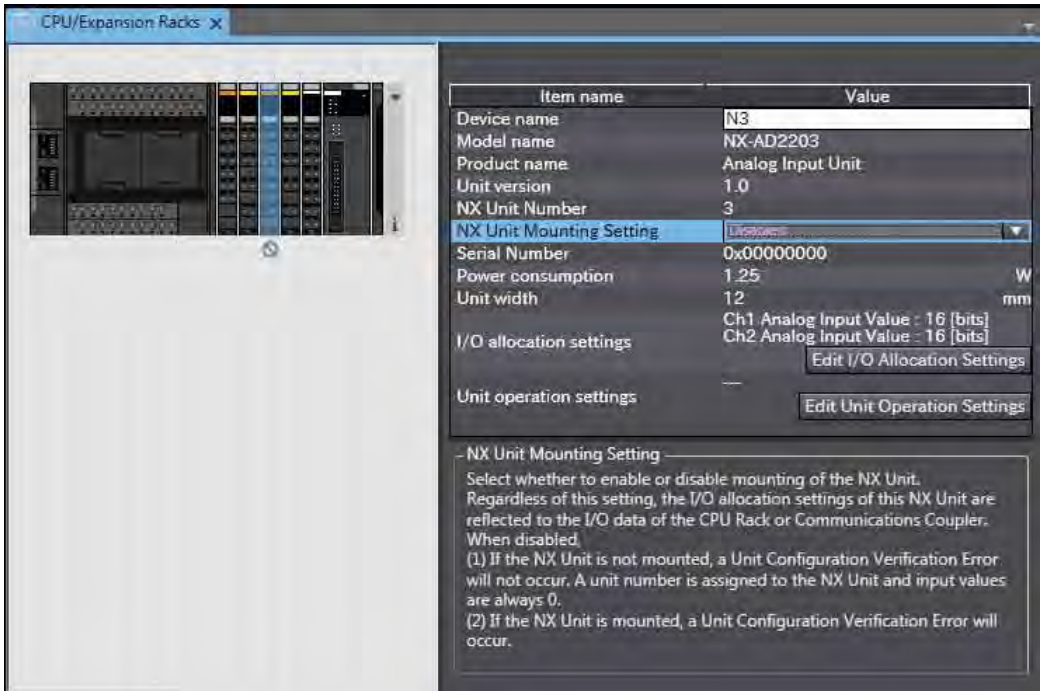


사용상의 주의 사항

미장착 유니트로 설정된 NX 유니트가 장착된 경우에는 유니트 구성 비교 이상이 발생합니다.

미장착 유니트 설정 방법

Sysmac Studio의 CPU·증설 랙 구성 편집 화면에서 대상의 NX 유니트를 선택하고, [NX 유니트 장착 설정]을 [무효]로 설정하면 미장착 유니트가 됩니다. 설정 변경 후에는 반드시 유니트 구성 정보를 실제 기기로 전송하십시오.



8-2-3 CPU 유니트 상의 NX 유니트 관련 재시작 기능

여기에서는 NX1P2 CPU 유니트 상의 NX 유니트의 재시작 기능에 대해 설명합니다. 재시작 기능은 컨트롤러의 전원을 OFF하지 않고 NX 유니트의 설정값을 유효로 합니다.

재시작 기능의 종류

NX 유니트의 재시작에는 다음과 같은 종류가 있습니다.

종류	기능
NX 버스 기능 모듈의 재시작	CPU 유니트 상의 모든 NX 유니트에 대해 재시작을 실행합니다.
NX 유니트 재시작	지정한 NX 유니트에 대해 재시작을 실행합니다.

NX 버스 기능 모듈의 재시작

CPU 유니트 상의 모든 NX 유니트에 대해 재시작을 실행합니다.
NX 버스 기능 모듈을 지정하여 재시작을 실행합니다. CPU 유니트의 태스크 실행에는 영향을 주지 않습니다. 또한, 이상이 발생하지도 않습니다.

재시작 수단은 다음과 같습니다.

- Sysmac Studio
- RestartNXUnit 명령

RestartNXUnit 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

NX 유니트 재시작

재시작하고자 하는 NX 유니트 1대를 지정하여 재시작을 실행합니다.
NX 유니트의 재시작 수단은 다음과 같습니다.

- Sysmac Studio
- RestartNXUnit 명령

RestartNXUnit 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.
또한, NX 버스 기능 모듈 및 재시작 지정하지 않은 NX 유니트는 동작을 계속합니다.

NX 유니트의 재시작 조작 방법

Sysmc Studio를 온라인 연결합니다. CPU·증설 랙 구성 편집 화면에서 CPU 유니트를 마우스 오른쪽 버튼으로 클릭하고 [NX 버스/NX 유니트의 재시작]–[예]를 선택하면 CPU 유니트 상의 모든 NX 유니트에 대해 재시작을 실행합니다.
동일하게 재시작하고자 하는 NX 유니트 1대를 마우스 오른쪽 버튼으로 클릭하고 [NX 버스/NX 유니트의 재시작]–[예]를 선택하면 지정한 NX 유니트에 대해 재시작을 실행합니다.

8-2-4 CPU 유니트 상의 NX 유니트 배선 확인

여기에서는 NX102 및 NX1P2 CPU 유니트 상의 NX 유니트의 배선 확인에 대해 설명합니다.

장치 셋업 시 등 사용자 프로그램이 작성되지 않는 경우에도 Sysmac Studio를 사용하여 CPU 유니트 상의 NX 유니트와 입출력 기기 간의 배선을 확인할 수 있습니다.

- **준비**

  NX 유니트의 구성 정보를 작성하고, CPU 유니트에 전송한 경우에는 Sysmac Studio의 유니트 구성 정보와 실제 유니트 구성이 일치하는지 확인하십시오.
  NX 유니트의 구성 정보를 CPU 유니트에 전송하지 않은 경우에는 온라인 상태에서 구성 편집 화면 상에서 마우스 오른쪽 버튼으로 클릭해서 [실제 유니트 구성과 비교·병합]을 선택하고, 구성을 자동 구축하십시오.

- **배선 확인 방법**

대상	확인 방법
입력 확인	Sysmac Studio를 온라인 연결하고, I/O 맵 상에서 유니트의 I/O 포트 현재값을 모니터링합니다.
출력 확인	Sysmac Studio를 온라인 연결하고, I/O 맵 상에서 유니트의 I/O 포트를 강제값 리프레시 또는 설정/리셋합니다.

강제값 리프레시에 대해서는 📖'8-6-1 강제값 리프레시(8-47페이지)'를 참조하십시오.

Sysmac Studio의 조작에 대해서는 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

참고

- 위치 인터페이스 유니트 등의 NX 유니트를 축에 할당한 경우에는 Sysmac Studio의 MC 시운전 기능을 사용하십시오. 자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.
- EtherCAT 커플러 유니트 상의 NX 유니트의 배선 확인은 EtherCAT 슬레이브 터미널의 I/O 확인 기능을 사용하십시오. 자세한 내용은 📖"NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"를 참조하십시오.

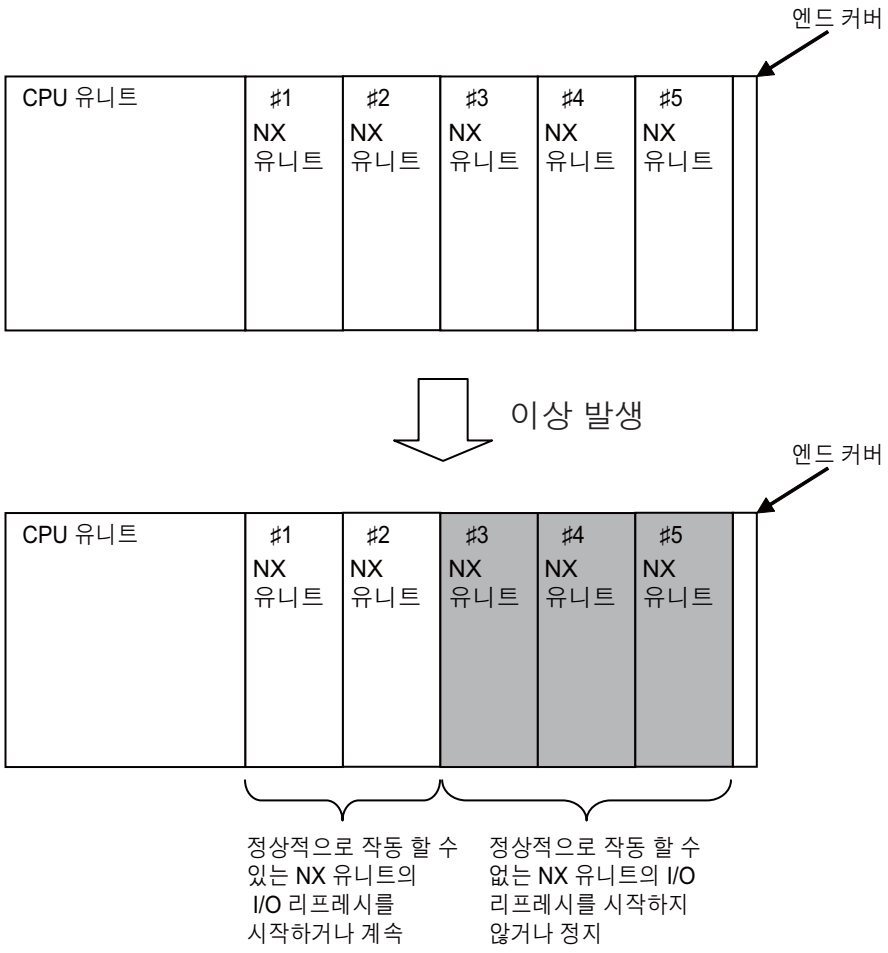
8-2-5 CPU 유니트 상의 NX 유니트 FAIL-SOFT 동작 기능

여기에서는 NX102 및 NX1P2 CPU 유니트의 NX 버스 기능 모듈의 FAIL-SOFT 동작 기능에 대해 설명합니다.
NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.

NX 버스 기능 모듈의 이상 발생 시 CPU 유니트 상의 정상적으로 동작할 수 있는 NX 유니트만을 대상으로 I/O 리프레시를 시작 또는 계속하는 기능입니다.

예를 들어 다음과 같은 경우에 사용합니다.

- CPU 유니트 상의 모든 NX 유니트를 즉시 정지하면 위험한 경우
- 사용자 프로그램 또는 사용자 조작으로 안전하게 정지할 수 있을 때까지는 CPU 유니트 상의 NX 유니트 동작을 계속하고자 하는 경우
- 장치를 모두 정지하지 않고 일부 장치의 가동을 계속하고자 하는 경우



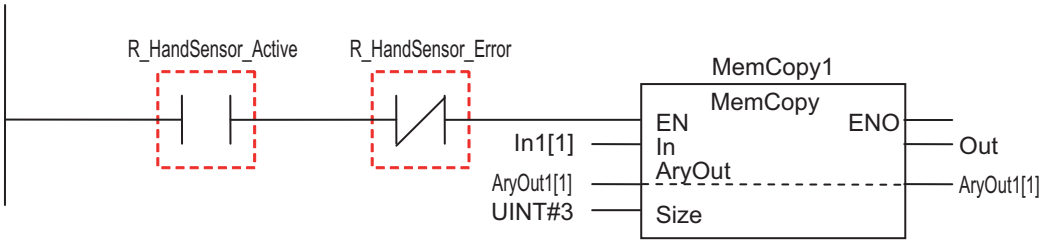
안전상의 중요 사항

FAIL-SOFT 동작 설정을 변경한 경우에 변경으로 인한 이상 발생 시 출력 상태가 변할 수 있습니다. 안전을 확인한 후 사용하십시오.

사용상의 주의 사항

FAIL-SOFT 동작하는 경우에는 유니트의 I/O 데이터의 유효성을 판단하기 위한 프로그램을 반드시 작성하십시오. 작성하지 않으면 I/O 리프레시를 계속하는 유니트와 계속하지 않는 유니트를 사용자 프로그램에서 판단할 수 없습니다.
유니트의 I/O 데이터 유효성을 판단하려면 NX 유니트 디바이스 변수 중 'NX 유니트 I/O 데이터 통신 중 Status'와 'NX 유니트 이상 Status'를 나타내는 변수를 사용합니다.

CPU 유니트 상의 NX 유니트(NX 유니트 번호: 5)의 I/O 데이터 유효성을 판단하기 위한 샘플 프로그램은 다음과 같습니다.



R_HandSensor_Active: 유니트 번호가 5인 NX 유니트 IO 통신 중 Status를 나타내는 디바이스 변수
R_HandSensor_Error: 유니트 번호가 5인 NX 유니트 이상 Status를 나타내는 디바이스 변수

이상 발생 시의 동작

NX 버스 기능 모듈이 FAIL-SOFT 동작하는 경우와 하지 않는 경우에는 이상 발생 시의 동작이 다음과 같습니다.

운전 상태	기동 시 이상이 발생했을 때의 동작	정상 동작 중에 이상이 발생했을 때의 동작
FAIL-SOFT 동작하는 경우	NX 버스 기능 모듈은 정상적으로 동작할 수 있는 NX 유니트와의 I/O 리프레시를 시작합니다. 정상적으로 동작할 수 없는 NX 유니트와의 I/O 리프레시는 시작하지 않습니다.	NX 버스 기능 모듈은 정상적으로 동작할 수 있는 NX 유니트와의 I/O 리프레시를 계속합니다. 정상적으로 동작할 수 없는 NX 유니트와의 I/O 리프레시는 정지합니다.
FAIL-SOFT 동작하지 않는 경우*1	NX 버스 기능 모듈은 모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.	NX 버스 기능 모듈은 모든 NX 유니트와의 I/O 리프레시를 정지합니다.

*1. 이상 발생 시에 CPU 유니트 상의 모든 NX 유니트와의 I/O 리프레시를 시작하지 않거나 정지한 상태를 '전체 정지'라고 합니다.

NX 버스 기능 모듈의 이상 발생 시의 동작은 I/O 리프레시의 동작을 제외하고, FAIL-SOFT 동작하는 경우와 하지 않는 경우에 관계없이 동일한 동작이 됩니다. 구체적으로는 발생한 이상을 통지하고, 이벤트 로그에 남깁니다.

FAIL-SOFT 동작 설정 방법

- **FAIL-SOFT 동작함**
  FAIL-SOFT 동작하는 경우에는 Sysmac Studio의 CPU·증설 랙 구성 편집 화면에서 CPU 유니트를 선택하고, [FAIL-SOFT 동작 설정]을 [FAIL-SOFT 사용]으로 설정합니다.
  또한 NX 버스 기능 모듈의 [FAIL-SOFT 동작 설정]의 초기값은 [FAIL-SOFT 사용]입니다.

- **FAIL-SOFT 동작하지 않음**
  FAIL-SOFT 동작하지 않는 경우에는 Sysmac Studio의 CPU·증설 랙 구성 편집 화면에서 CPU 유니트를 선택하고 [FAIL-SOFT 동작 설정]을 [정지]로 설정합니다.

사용상의 주의 사항

- 설정 변경 후에는 변경한 설정을 CPU 유니트로 반드시 전송하십시오.
- NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정에 대해서는 '4-2-4 NX102 및 NX1P2 CPU 유니트에서의 유니트 구성·유니트 설정(4-12페이지)'을 참조하십시오.

FAIL-SOFT 동작 대상이 되는 이상

FAIL-SOFT 동작 대상이 되는 이상의 일례는 다음과 같습니다.

- NX 버스 통신 이상
- 등록 NX 유니트 미장착
- NX 유니트 통신 타임아웃
- NX 유니트 초기화 이상
- NX 유니트 기동 이상

[FAIL-SOFT 사용]으로 설정해도 이상 발생 요인에 따라서는 CPU 유니트 기동 시 NX 버스 기능 모듈이 모든 NX 유니트와의 I/O 리프레시를 시작하지 않는 경우가 있습니다.
발생 요인에 대한 동작의 자세한 내용은 '유니트 구성 비교 이상에서의 발생 요인과 이상 시 동작(8-20 페이지)'을 참조하십시오.

FAIL-SOFT 동작 대상이 되는 이상에 대해서는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.
FAIL-SOFT 동작 대상이 아닌 이상이 발생한 경우에 [FAIL-SOFT 사용]으로 설정해도 NX 버스 기능 모듈은 모든 NX 유니트와의 I/O 리프레시를 정지합니다.

유니트 구성 비교 이상에서의 발생 요인과 이상 시 동작

[FAIL-SOFT 사용]으로 설정해도 CPU 유니트 기동 시의 이상 발생 요인에 따라 I/O 리프레시를 시작하지 않는 경우가 있습니다.
예시는 다음과 같습니다.

유니트 구성 정보와 실제 구성의 예							구성 설명	CPU 유니트의 기동 시의 동작
		NX 유니트 번호						
		1	2	3	4	5		
유니트 구성 정보		A	B	C	D	E*1	형식 A, B, C, D와 E의 유니트를 왼쪽에 기재된 연결 순서로 CPU 유니트에 장착하는 구성입니다.	−
실제 구성	사례 1	A	B	C	−	−	형식 D의 유니트를 설치하지 않은 구성입니다.	NX 유니트 번호 1, 2와 3은 FAIL-SOFT 동작에 따라 I/O 리프레시를 시작합니다.
	사례 2	A	C	D	−	−	형식 B의 유니트를 설치하지 않은 구성입니다.	모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.
	사례 3	A	B	D	C	−	형식 C와 형식 D의 유니트 연결 순서를 바꿔서 설치한 구성입니다.	모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.
	사례 4	A	B	C	D	D	NX 유니트 번호 5에 형식 D의 유니트를 1대 많게 설치한 구성입니다.	모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.
	사례 5	A	B	C	F	−	NX 유니트 번호 4에 유니트 구성 정보에 존재하지 않는 형식 F의 유니트를 설치한 구성입니다.	모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.
	사례 6	A	B	C	D	E	NX 유니트 번호 5에 장착 설정 기능으로 '무효'로 설정한 형식 E의 유니트를 설치한 구성입니다.	모든 NX 유니트와의 I/O 리프레시를 시작하지 않습니다.

*1. [NX 유니트 장착 설정]이 [무효]로 설정된 것으로 가정합니다.

8-2-6 CPU 유니트 상의 NX 유니트 통전 시간 모니터링 기능

여기에서는 NX1P2 CPU 유니트 상의 NX 유니트의 통전 시간 모니터링 기능에 대해 설명합니다.
CPU 유니트 상의 NX 유니트가 로컬 유니트에 공급된 유니트 전원의 누적 통전 시간을 기록해서 Sysmac Studio에 표시하는 기능입니다.

통전 시간 모니터링 기능의 사양

통전 시간 모니터링 기능의 사양은 다음과 같습니다.

항목	사양
표시 단위	• 누적 통전 시간이 1시간 미만 : 분 • 누적 통전 시간이 1시간 이상 : 시
갱신 간격	• 누적 통전 시간이 24시간 미만 : 10분 • 누적 통전 시간이 24시간 이상 : 1시간
계측 오차	1시간/월 이내
공장 출하 시 설정	0분

누적 통전 시간 확인 방법

Sysmac Studio에서 CPU 유니트 상의 NX 유니트 누적 통전 시간을 확인할 수 있습니다.
또는 다음과 같은 전용 명령에서도 누적 통전 시간을 확인할 수 있습니다.

- NX_ReadTotalPowerOnTime 명령

여기에서는 Sysmac Studio에 의한 확인 방법을 설명합니다. NX_ReadTotalPowerOnTime 명령의 자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

- **Sysmac Studio에서의 누적 통전 시간 확인 방법**
  Sysmac Studio에서는 [생산 정보]에서 확인할 수 있습니다.

  1 온라인 상태로 하십시오.

  2 멀티 뷰 탐색기에서 [구성·설정]–[CPU·증설 랙]–[CPU 랙]을 마우스 오른쪽 버튼으로 클릭하고 [생산 정보 표시]를 선택하십시오.
  [생산 정보] 사용자 대화 상자가 표시됩니다.
  상세 표시로 전환하면 NX 유니트별 누적 통전 시간을 확인할 수 있습니다.

- **기록할 수 없게 된 경우의 표시**
  비휘발성 메모리 하드웨어 고장 등으로 누적 통전 시간을 기록할 수 없게 된 경우에 Sysmac Studio의 누적 통전 시간은 '잘못된 기록'이라고 표시됩니다.

- **통전 시간 모니터링 기능을 지원하지 않는 유니트의 표시**
  통전 시간 모니터링 기능을 지원하지 않는 유니트인 경우에 Sysmac Studio의 누적 통전 시간은 '---'로 표시됩니다.

- **읽기에 실패한 경우의 표시**
  읽기에 실패한 경우에 Sysmac Studio의 누적 통전 시간은 '---'로 표시됩니다.

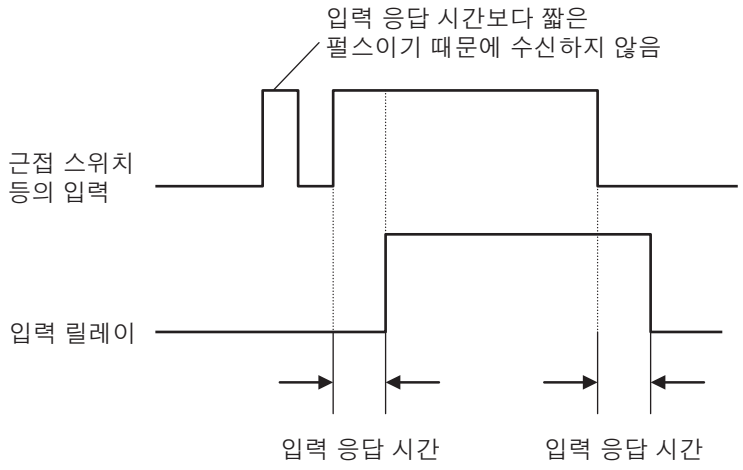
8-3 CJ 유니트 관련 관리 기능

여기에서는 유니트 관련 관리 기능에 대해 설명합니다.
CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

8-3-1 기본 I/O 유니트 관련

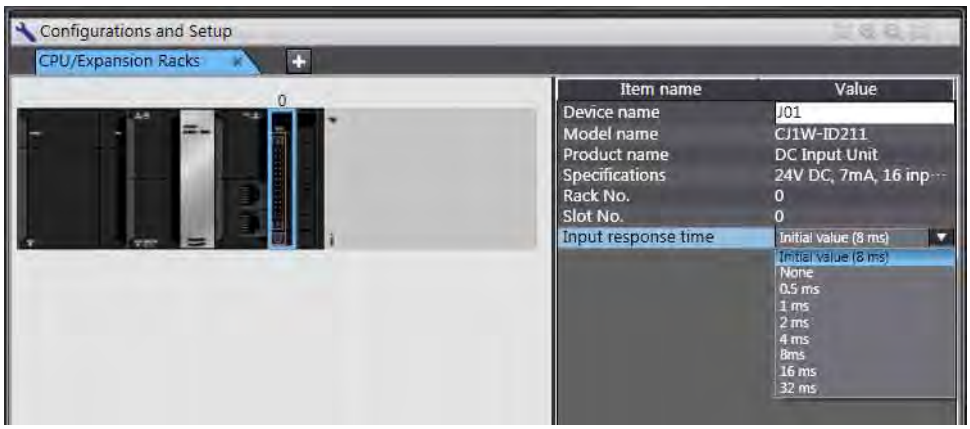
개요

입력 응답 시간을 길게 하면 채터링 및 외부 노이즈의 영향을 받기 어렵게 할 수 있습니다. 반대로 짧게 하면(ON 시간 또는 OFF 시간이 리프레시 시간보다 긴 조건이면) 짧은 펄스 입력을 수신할 수 있습니다.



설정 방법

Sysmac Studio의 멀티 뷰 탐색기에서 [구성·설정]–[CPU·증설 랙]을 두 번 클릭하고, 기본 I/O 유니트의 유니트 정보의 '입력 응답 시간'의 풀다운에서 입력 응답 시간을 선택합니다.



설정을 유효화하기 위해서는 다음 중 하나를 실행해야 합니다.

- 컨트롤러 전원의 OFF/ON
- Sysmac Studio에서의 컨트롤러 리셋(CPU 유니트 전체)

관련 시스템 정의 변수

기본 입력 유니트의 입력 응답 시간의 설정값은 다음과 같은 시스템 정의 변수에 출력됩니다.

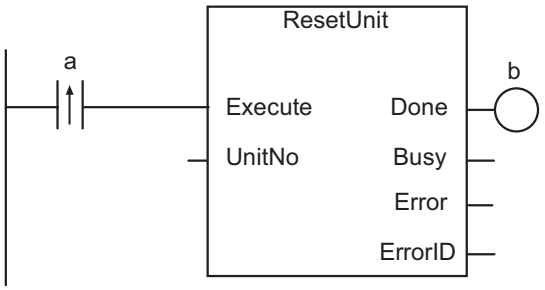
변수명	이름	기능	데이터 형식	R/W
_CJB_InRespTm	기본 입력 유니트의 입력 시상수	기본 입력 유니트의 입력 시상수를 0.1ms 단위로 표시합니다.	ARRAY [0..3, 0..9] OF UINT	R

8-3-2 고기능 유니트 관련

고기능 유니트 재시작

컨트롤러의 전원을 OFF로 하지 않고 고기능 유니트(고기능 I/O 유니트 또는 CPU 고기능 유니트)의 유니트 설정값을 유효로 합니다.
다음과 같은 ResetUnit(유니트 재시작) 명령에 의해 실행합니다.

명령 이름	명령	기능
유니트 재시작	ResetUnit	CPU 고기능 유니트 또는 고기능 I/O 유니트를 재시작합니다.



실행 조건 a가 FALSE→TRUE가 되면, 여러 차례의 태스크 주기에 걸쳐 고기능 유니트를 재시작합니다.
재시작이 정상적으로 완료되면 출력 변수 'Done'(정상 완료)이 TRUE가 되고, b가 TRUE가 됩니다.

다음 중 하나로 고기능 유니트 설정을 변경한 경우에 유니트의 재시작 또는 컨트롤러의 전원 OFF→ON이 필요합니다.

- Sysmac Studio의 고기능 유니트 설정 화면에서 편집한 경우
- I/O 맵 및 감시 창에서 편집한 경우
- 사용자 프로그램 내에 설정한 경우

● **관련 시스템 정의 변수**

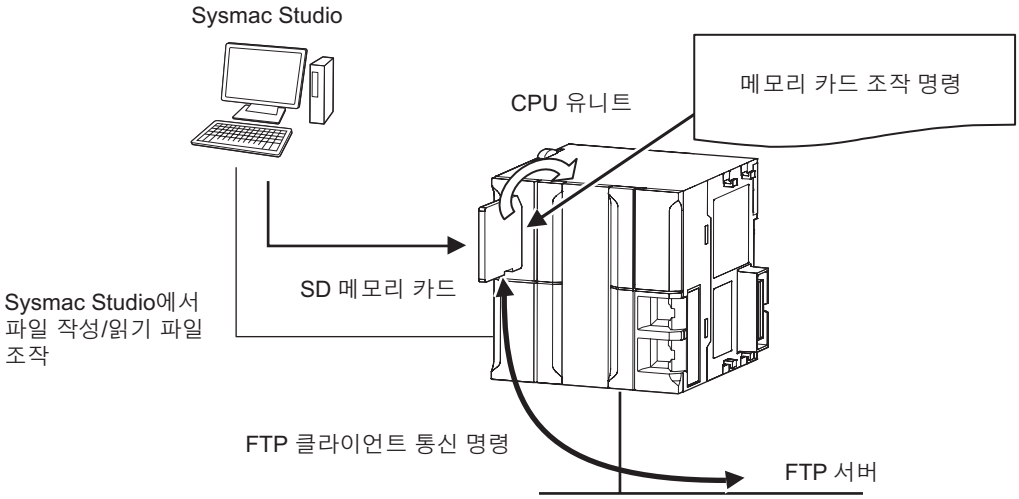
변수명	이름	기능	데이터 형식	R/W
_CJB_CBU00InitSta . . . _CJB_CBU15InitSta	CPU 고기능 유니트 초기화 중 플래그	CPU 고기능 유니트가 초기화 중일 때 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 유니트 번호가 대응됩니다.	BOOL	R
_CJB_SIO00InitSta . . . _CJB_SIO95InitSta	고기능 I/O 유니트 초기화 중 플래그	고기능 I/O 유니트가 초기화 중에 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 호기 번호가 대응됩니다.	BOOL	R
_CJB_CBU00Restart . . . _CJB_CBU15Restart	CPU 고기능 유니트 재시작 플래그	상승 시 CPU 고기능 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 호기 번호에 대응됩니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다.	BOOL	RW
_CJB_SIO00Restart . . . _CJB_SIO95Restart	고기능 I/O 유니트 재시작 플래그	상승 시 고기능 I/O 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 호기 번호에 대응됩니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다.	BOOL	RW

8-4 SD 메모리 카드 기능

여기에서는 SD 메모리 카드 기능에 대해 설명합니다.

8-4-1 SD 메모리 카드 기능 목록

NJ/NX 시리즈 CPU 유니트에서 지원하는 SD 메모리 카드 기능은 다음과 같습니다.



기능명	개요
SD 메모리 카드 조작 명령	사용자 프로그램의 명령으로 SD 메모리 카드에 액세스하는 기능입니다.
FTP 클라이언트 통신 명령*1	CPU 유니트에서 Ethernet의 컴퓨터 및 컨트롤러에 FTP를 사용하여 파일을 전송하기 위한 명령입니다.
FTP 서버 기능	인트라넷 내의 FTP 클라이언트에서 FTP 명령을 사용하여 EtherNet/IP 경유로 SD 메모리 카드 내의 대용량 파일을 읽고 쓰는 기능입니다
Sysmac Studio에서의 파일 조작 기능	Sysmac Studio에서 CPU 유니트에 장착한 SD 메모리 카드에 대해 파일 조작을 실시하는 기능입니다. 사용자는 SD 메모리 카드에 컨트롤러용 파일뿐만 아니라 범용 문서 파일 등도 저장할 수 있습니다.
SD 메모리 카드의 수명 감지 기능	SD 메모리 카드의 수명을 시스템 정의 변수 또는 이벤트 로그로 통지하는 기능입니다.
SD 메모리 카드 백업 기능, SD 메모리 카드 자동 전송 기능, SD 메모리 카드 프로그램 전송 기능 *2	CPU 유니트에 장착한 SD 메모리 카드를 사용하여 컨트롤러의 사용자 프로그램 및 데이터 등을 백업 및 복원, 비교하는 기능입니다. 자세한 내용은 '9-2 SD 메모리 카드 백업 기능(9-13페이지)', '9-4 SD 메모리 카드 자동 전송 기능(9-32페이지)', '9-5 SD 메모리 카드 프로그램 전송 기능(9-34페이지)'을 참조하십시오.

*1. FTP 클라이언트 통신 명령은 CPU 유니트Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.
*2. CPU 유니트 Ver.1.11 이상과 Sysmac Studio Ver.1.15 이상의 조합에서 사용할 수 있습니다.

8-4-2 지원하는 SD 메모리 카드 및 폴더/파일 사양

SD 메모리 카드의 사양

본 제품이 지원하는 메모리 카드는 SD 카드 및 SDHC 카드입니다. 단, OMRON SD 메모리 카드만 동작을 보증합니다. 동작을 보증하지 않는 SD 카드 및 SDHC 카드를 사용한 경우에 올바르게 동작하지 않을 가능성이 있습니다.

형식	카드 유형	용량[GB]	포맷	재기입 횟수	쓰기 방지
HMC-SD291*1	SD 카드	2	FAT16	10만 회	SD 메모리 카드 자체의 하드웨어 스위치로 설정 가능
HMC-SD491	SDHC 카드	4	FAT32		

*1. NJ501-□□□□인 경우에 하드웨어 개정 번호가 "A", 유니트 버전이 Ver.1.15 이상인 CPU 유니트에서는 HMC-SD291을 사용할 수 없습니다.

다음과 같은 경우에 시스템 정의 변수 '_Card1Err'(SD 메모리 카드 에러 플래그)가 TRUE가 됩니다(감시 정보 레벨).

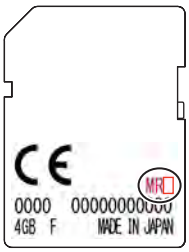
- 형식 이상인 경우

이 때, 앞면의 'SD PWR' LED는 소등되고, SD 메모리 카드에 액세스할 수 없습니다.

사용상의 주의 사항

뒷면에 "MR□"라고 표시된 SD 메모리 카드를 NX701-□□□□ 또는 NJ501-□□□□에서 사용하는 경우에 CPU 유니트의 유니트 버전은 다음과 같이 하십시오.

- NX701-□□□□: Ver.1.14 이상
- NJ501-□□□□: Ver.1.13 이상



폴더/파일 사양

- **문자 제한**

사용자가 이름을 붙이는 경우	사용할 수 있는 문자	예약어	멀티바이트 문자 사용 가능 여부	대문자/ 소문자 구별	최대 크기 (종단의 NULL 문자는 포함하지 않음)
볼륨 라벨명	0~9, A~Z, a~z % - _ @ ! ` ( ) ~ = # & + ^ [ ] { } , . ; 반각 문자*1	CON, PRN, AUX, CLOCK$, NUL, COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT0, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9	불가능*2	사용 안 함	11바이트
디렉토리명	0~9, A~Z, a~z $ % ' - _ @ ! ` ( ) ~ = # & + ^ [ ] { } , . ; 반각 문자				65바이트
파일명					65바이트

*1. 선두 문자에 공백은 사용할 수 없습니다.
*2. 멀티바이트 문자(한글, 일본어 등)는 PC에서 사용할 수 있지만 CPU 유니트의 폴더/파일명으로는 사용할 수 없습니다.

- **서브 디렉토리 계층**

  최대 5계층까지(예: f1/f2/f3/f4/f5/abc.txt)

- **저장할 수 있는 최대 파일 수**

  SD 메모리 카드에 저장할 수 있는 최대 파일 수는 파일을 저장하는 디렉토리 계층에 따라 다릅니다. 각각의 최대 파일 수는 다음과 같습니다. 단, 다음 표의 값은 파일명이 8.3 형식인 경우입니다. 파일명이 긴 경우에는 저장할 수 있는 최대 파일 수가 다음 표의 값보다 적어집니다.

디렉토리 계층	포맷	저장할 수 있는 최대 파일 수
루트 디렉토리	FAT16	511
	FAT32	65,533
서브 디렉토리	FAT16, FAT32	65,533

- **1파일 최대 크기**

  1파일의 최대 크기는 2,147,483,647바이트(2GB-1바이트)입니다.

8-4-3 SD 메모리 카드 조작 명령

다음 명령을 사용하여 SD 메모리 카드에 대한 각종 조작을 실행할 수 있습니다.

명령 이름	명령	기능
변수 파일 읽기	**FileReadVar**	SD 메모리 카드 내의 파일(바이너리 형식)을 읽고, 지정한 변수(배열 변수 및 구조체 변수도 가능)에 저장
변수 파일 쓰기	**FileWriteVar**	지정한 변수(배열 변수 및 구조체 변수도 가능)의 값을 바이너리 형식으로 SD 메모리 카드 내의 파일에 씀 파일명에 지정한 디렉토리가 존재하지 않는 경우에는 디렉토리 자체도 작성
파일 열기	**FileOpen**	지정한 파일을 염
파일 닫기	**FileClose**	지정한 파일을 닫음
파일 찾기	**FileSeek**	지정한 파일의 파일 위치 지시자를 설정
파일 읽기	**FileRead**	지정한 파일에서 데이터를 읽음
파일 쓰기	**FileWrite**	지정한 파일에 데이터를 씀
문자열 읽기	**FileGets**	지정한 파일에서 1행 분량의 문자열을 읽음
문자열 쓰기	**FilePuts**	지정한 파일에 1행 분량의 문자열을 씀
파일 삭제	**FileRemove**	SD 메모리 카드 내의 지정한 파일을 삭제
파일명 변경	**FileRename**	지정한 파일의 파일명 또는 디렉토리명을 변경
파일 복사	**FileCopy**	지정한 파일을 다른 파일명으로 복사
디렉토리 작성	**DirCreate**	SD 메모리 카드 내에 디렉토리를 작성
디렉토리 삭제	**DirRemove**	SD 메모리 카드 내의 지정한 디렉토리를 삭제

8-4-4 FTP 클라이언트 통신 명령

FTP 클라이언트 통신 명령은 NJ/NX 시리즈 CPU 유니트에서 Ethernet 의 컴퓨터나 컨트롤러에 FTP를 사용하여 파일을 전송하기 위한 명령입니다. 다음 명령에서는 명령을 실행할 때 SD 메모리 카드의 파일 읽기/쓰기가 실행됩니다.

명령 이름	명령	기능
FTP 서버에 파일 업로드	**FTPPutFile**	FTP 클라이언트의 SD 메모리 카드 내에 어떤 파일을 FTP 서버에 업로드
FTP 서버에서 파일 다운로드	**FTPGetFile**	FTP 서버 내의 파일을 FTP 클라이언트의 SD 메모리 카드에 다운로드

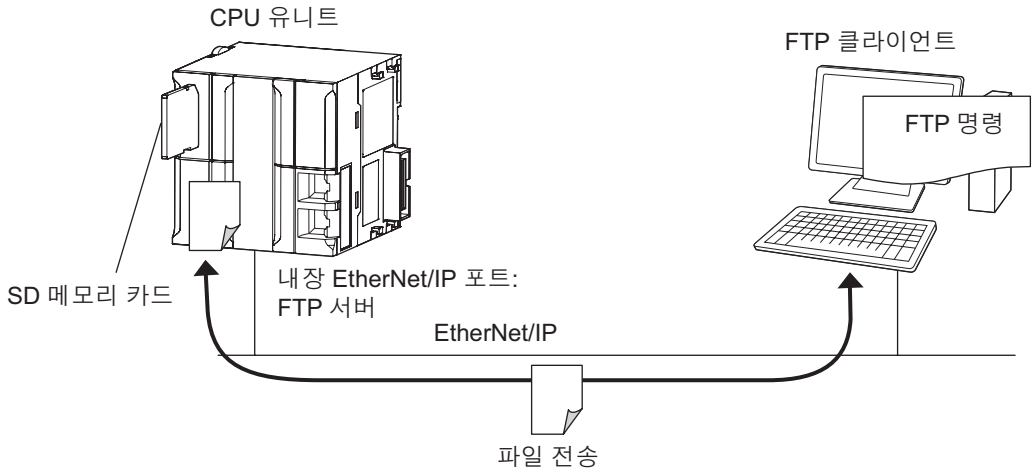
버전 관련 정보

FTP 클라이언트 통신명령은 CPU 유니트Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

8-4-5 FTP 서버 기능

FTP 클라이언트에서 내장 EtherNet/IP 포트에 대해 FTP 명령을 전송하면 EtherNet/IP를 경유하여 SD 메모리 카드 내의 파일을 읽고 쓸 수 있습니다.

자세한 내용은 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"를 참조하십시오.



8-4-6 Sysmac Studio에서의 파일 조작

Sysmac Studio에서 CPU 유니트에 장착한 SD 메모리 카드에 대해 파일 조작을 실행할 수 있습니다.
SD 메모리 카드에는 컨트롤러용 파일뿐만 아니라 문서 파일 등도 저장할 수 있습니다.

8-4-7 SD 메모리 카드의 수명 감지 기능

SD 메모리 카드가 수명에 의해 물리적으로 파괴되기 전에 SD 메모리 카드의 수명을 파악할 수 있습니다.
다음과 같이 SD 메모리 카드의 수명을 감지할 수 있습니다.

- 시스템 정의 변수 '_Card1Deteriorated'(SD 메모리 카드 수명 경고 플래그)
- 이벤트 로그에 'SD 메모리 카드 수명(감시 정보)' 등록

감지 타이밍은 전원 투입 시와 SD 메모리 카드 장착 중(정기적으로 확인)입니다.

SD 메모리 카드의 수명을 감지한 경우에 SD 메모리 카드 내의 데이터 저장 또는 SD 메모리 카드 교환 등을 실행하십시오.

8-4-8 SD 메모리 카드 관련 시스템 정의 변수 목록

SD 메모리 카드 기능의 상태를 통지하는 시스템 정의 변수는 다음과 같습니다.

변수명	이름	기능	데이터 형식
_Card1Ready	SD 메모리 카드 사용 가능 플래그	SD 메모리 카드가 인식되었을 때 TRUE가 됩니다. 인식하지 못했을 때 FALSE가 됩니다. TRUE: 사용 가능 FALSE: 사용 불가능	BOOL
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	SD 메모리 카드에 쓰기 방지가 설정되었을 때 TRUE가 됩니다. TRUE: 쓰기 방지 있음 FALSE: 쓰기 방지 없음	BOOL
_Card1Err	SD 메모리 카드 에러 플래그	사용이 불가능한 SD 메모리 카드가 장착됐을 때 또는 포맷에 이상이 발생했을 때 TRUE가 됩니다. TRUE: 에러 있음 FALSE: 에러 없음	BOOL
_Card1Access	SD 메모리 카드 액세스 중 플래그*1	SD 메모리 카드에 액세스 중에 TRUE가 됩니다. TRUE: 액세스 중 FALSE: 액세스 중이 아님	BOOL
_Card1Deteriorated	SD 메모리 카드 수명 경고 플래그	SD 메모리 카드의 수명을 감지했을 때 TRUE가 됩니다. TRUE: 수명 감지 있음 FALSE: 수명 감지 없음	BOOL
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그*2	SD 메모리 카드 액세스 중에 컨트롤러의 전원 OFF가 발생했을 때 TRUE가 됩니다. TRUE; SD 메모리 카드 액세스 중에 전원 OFF가 있었음 FALSE: 정상	BOOL

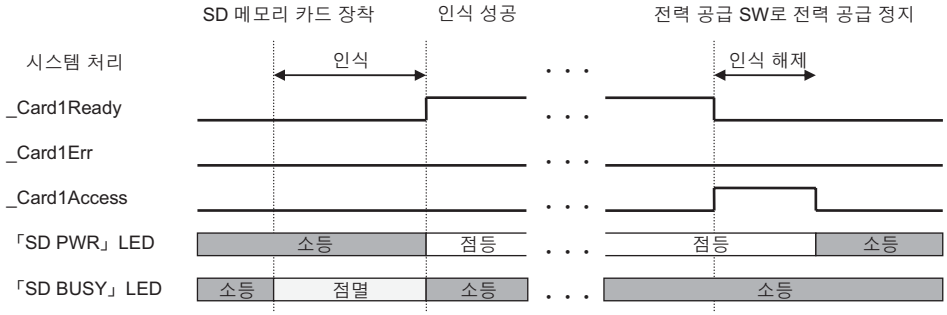
*1. SD 메모리 카드 액세스 중 플래그('_Card1Access') 사용 시의 주의 사항
SD 메모리 카드 액세스 중 플래그는 외부로의 통지를 목적으로 한 플래그입니다. SD 메모리 카드에 대한 액세스 상태를 실시간으로 반영하지는 않습니다. 따라서 사용자 프로그램 내에서는 이 플래그를 참조하지 마십시오. SD 메모리 카드에 대한 액세스 상태를 실시간으로 반영하지 않으므로 사용자 프로그램의 잘못된 동작의 원인이 될 수 있습니다.

*2. SD 메모리 카드 액세스 중 전원 OFF 플래그('_Card1PowerFail') 사용 시의 주의 사항
SD 메모리 카드 액세스 중 전원 OFF 플래그가 TRUE가 된 경우에는 SD 메모리 카드 내에 올바른 파일이 존재하는지 또한 장치가 올바르게 동작하는지 확인하십시오.
올바른 파일이 존재하지 않는 경우 또는 장치가 올바르게 동작하지 않는 경우에는 SD 메모리 카드에 올바른 파일을 다운로드하십시오. CPU 유니트를 전원 재투입 또는 컨트롤러 리셋하여 장치가 올바르게 동작하는지 확인하십시오.
처리가 완료됐으면 SD 메모리 카드 액세스 중 전원 OFF 플래그를 FALSE로 하십시오. (이 플래그는 자동으로 FALSE가 되지 않습니다).

(주의) SD 메모리 카드 백업 기능에 관련 시스템 정의 변수는 '9-2 SD 메모리 카드 백업 기능(9-13페이지)'을 참조하십시오.

참고

SD 메모리 카드의 인식/장착 해제 시의 타이밍 차트



8-4-9 SD 메모리 카드의 자체 진단 기능

전원 투입 시에 장착된 SD 메모리 카드에 대해 자체 진단을 실행할 수 있습니다.

Sysmac Studio에서 다음과 같은 '구성·설정'-'컨트롤러 설정'-'동작 설정'에서 전원 투입 시 다음과 같은 자체 진단의 실행 여부를 선택할 수 있습니다.

- 파일 시스템의 확인
- CHKDSK 상당의 확인
- 확인 실패 시의 복구 기능

설정 대항목	설정 중항목	설정 소항목	기능 설명	설정값
동작 설정-동작 설정 탭-전반 설정	SD 메모리 카드 설정	전원 투입 시의 메모리 카드 진단*1	전원 투입 시에 장착된 SD 메모리 카드에 대해 자체 진단(파일 시스템의 확인과 복구)을 실행할지 여부를 설정합니다.	확인 안 함 확인

*1. SD 메모리 카드 자체가 잠금 설정(쓰기 방지 설정)된 경우에 자체 진단은 실행되지 않습니다.

● **자체 진단 실행 결과**

사례	LED			이상의 종류	처리	비고
	RUN	SD PWR	SD BUSY			
자가 진단 실행 중	점멸	소등	점등	−	−	
자체 진단 종료 시 문제가 없었던 경우	−	점등	소등	정상	없음	
SD 메모리 카드의 포맷이 잘못된 경우	−	소등	소등	감시 정보	Sysmac Studio 를 사용하여 포맷하십시오.	
파일 시스템 확인에서 에러를 검출하고, 파일 시스템 복구를 자동으로 실행한 경우	−	소등	복구 시: 점멸 완료 후: 소등	감시 정보	Sysmac Studio 에서의 파일 조작 또는 PC에 SD 메모리 카드를 장착하여 복구에 의해 삭제된 파일이 없는지 확인합니다.	손상된 파일을 검출한 경우에는 손상된 파일 복구를 시도합니다.
SD 메모리 카드가 고장 난 경우	−	소등	소등	감시 정보	SD 메모리 카드를 교환하십시오.	



안전상의 중요 사항

전원 투입 시 복구 기능이 동작한 경우에 운전 모드로 이행할 때까지 시간이 필요합니다. 그 동안 출력은 OFF가 되고, 외부와 통신도 할 수 없습니다.
전원 유니트 RUN 접점 출력을 이용하는 등 외부 기기가 오작동하지 않도록 안전 보장 회로를 만드십시오.



사용상의 주의 사항

전원 투입 시의 SD 메모리 카드 자체 진단 설정을 유효로 한 경우에도 SD 메모리 카드 액세스 중에 CPU 유니트의 전원을 끄지 마십시오.
손상된 파일이 있는 경우에 SD 메모리 카드 복구 기능을 사용하여 복구를 시도하고, 복구할 수 없는 경우에는 전원 투입 시 자동으로 삭제될 가능성이 있습니다.

8-4-10 SD 메모리 카드 내의 파일 액세스의 배타 제어

SD 메모리 카드 내의 파일 액세스 방법에는 다음 4 종류가 있습니다.

(1) FTP 서버 기능

(2) SD 메모리 카드 조작 명령

(3) FTP 클라이언트 통신 명령

(4) Sysmac Studio에서의 파일 조작

SD 메모리 카드 내의 파일 액세스를 동일한 파일에 대해 동시에 실행하면 쓰는 도중인 파일을 읽거나, 읽는 도중인 파일에 덮어쓰는 등 사용자가 의도하지 않는 처리가 실행될 가능성이 있습니다. 따라서 이러한 처리가 동일한 파일에 대해 동시에 실행되지 않도록 배타 제어를 실행해야 합니다. 배타 제어가 필요한 처리의 조합은 다음과 같습니다. 배타 제어가 필요한 조합의 처리를 하는 경우에는 한쪽의 처리가 종료됐는지 확인한 후 다른 한쪽 처리를 실행하십시오.

<table>
<tr><th colspan="3" rowspan="3"></th><th colspan="6">먼저 실행 중인 액세스</th></tr>
<tr><th colspan="2" rowspan="2">명령*1</th><th colspan="2">Sysmac Studio에서의 파일 조작</th><th colspan="2">FTP 서버 기능</th></tr>
<tr><th>읽기</th><th>쓰기</th><th>읽기</th><th>쓰기</th><th>읽기</th><th>쓰기</th></tr>
<tr><td rowspan="5">나중에 실행하는 액세스</td><td rowspan="2">명령</td><td>읽기</td><td colspan="2" rowspan="2">자동으로 배타 제어되고, 나중에서 실행하는 명령이 에러가 됩니다.</td><td>배타 제어는 필요 없습니다.</td><td rowspan="2">배타 제어를 실행하십시오.</td><td>배타 제어는 필요 없습니다.</td><td rowspan="4">배타 제어를 실행하십시오.</td></tr>
<tr><td>쓰기</td><td>배타 제어를 실행하십시오.</td><td>배타 제어를 실행하십시오.</td></tr>
<tr><td>Sysmac Studio에서의 파일 조작</td><td>읽기</td><td>배타 제어는 필요 없습니다.</td><td>배타 제어를 실행하십시오.</td><td>－</td><td>－</td><td>배타 제어는 필요 없습니다.</td></tr>
<tr><td>Sysmac Studio에서의 파일 조작</td><td>쓰기</td><td colspan="2">배타 제어를 실행하십시오.</td><td>－</td><td>－</td><td>배타 제어를 실행하십시오.</td></tr>
<tr><td>FTP 서버 기능</td><td>읽기</td><td>배타 제어는 필요 없습니다.</td><td>배타 제어를 실행하십시오.</td><td>배타 제어는 필요 없습니다.</td><td>배타 제어를 실행하십시오.</td><td>－</td><td>－</td></tr>
<tr><td>FTP 서버 기능</td><td>쓰기</td><td colspan="4">배타 제어를 실행하십시오.</td><td>－</td><td>－</td></tr>
</table>

*1. 명령에는 SD 메모리 카드 조작 명령과 FTP 클라이언트 통신 명령이 포함됩니다.

8-5 보안 기능

NJ/NX 시리즈에서 사용할 수 있는 보안 기능에 대해 설명합니다.

보안 기능을 사용하면 자산 보호를 위해 컨트롤러 내의 사용자 프로그램 및 각종 데이터를 보호할 수 있습니다. 또한, 잘못된 조작을 방지하기 위해 Sysmac Studio에서의 조작을 제한할 수도 있습니다.

NJ/NX 시리즈는 다음과 같은 보안 기능을 가지고 있습니다.

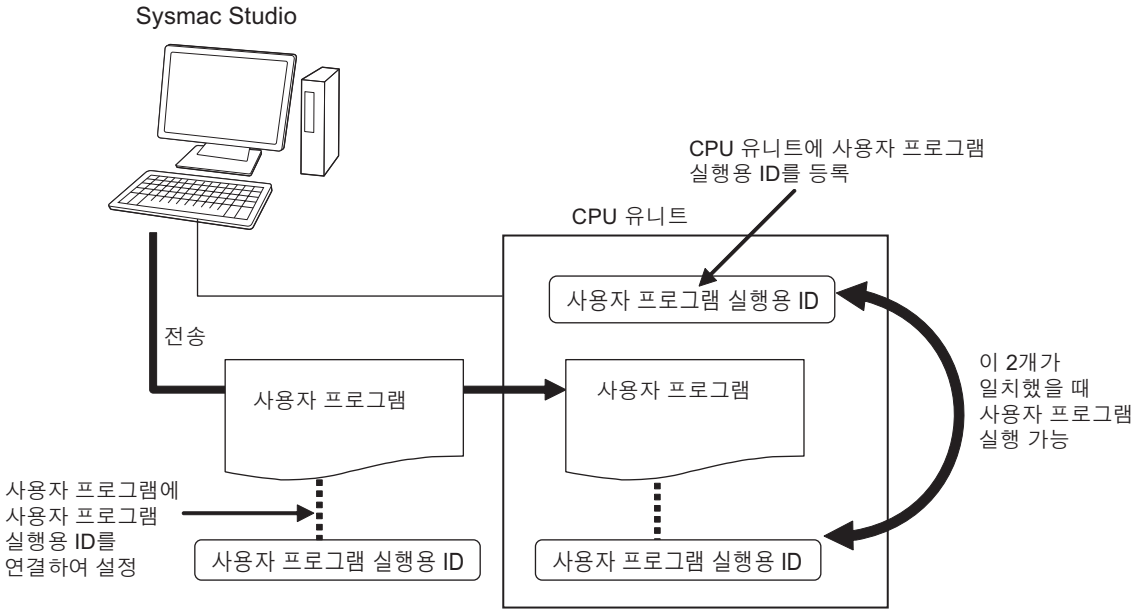
목적	보안 기능	기능의 개요	참조
자산 도용 방지	사용자 프로그램 실행용 ID 인증 기능	사용자 프로그램을 복제해도 다른 CPU 유니트에서는 동작되지 않도록 합니다.	'8-5-1 사용자 프로그램 실행용 ID 인증 기능(8-35페이지)'
	사용자 프로그램 복원 정보 없는 전송 기능	CPU 유니트에 프로그램 복원 정보(소스)를 전송하지 않는 것으로 다른 PC에서 업로드했을 때 프로그램을 표시할 수 없도록 합니다.	'8-5-2 사용자 프로그램 복원 정보 없는 전송 기능(8-38페이지)'
	프로젝트 파일 전체 보호 기능	프로젝트 파일에 비밀번호 보호를 설정하여 사용자 자산 정보를 보호합니다.	'8-5-3 프로젝트 파일 전체 보호 기능(8-39페이지)'
	데이터 보호 기능*1	프로젝트 중에 보호하고자 하는 데이터에 보호 기능을 부여하여 사용자 자산 정보를 보호합니다.	'8-5-4 데이터 보호 기능(8-40페이지)'
잘못된 조작 방지	조작 권한 인증 기능	Sysmac Studio에서 CPU 유니트에 대해 조작 권한을 설정하여 조작할 수 있는 기능을 제한합니다.	'8-5-5 조작 권한 인증 기능(8-42페이지)'
	CPU 유니트에 대한 쓰기 방지 기능	컨트롤러 내부 데이터를 Sysmac Studio에서 재기입할 수 없게 합니다.	'8-5-6 CPU 유니트에 대한 쓰기 방지 기능(8-44페이지)'
잘못된 연결 방지	CPU 유니트 이름 기능	PC와 CPU 유니트의 컨트롤러의 이름 또는 시리얼 ID가 일치하는지 확인하여 잘못해서 온라인에 연결하는 것을 방지합니다.	'8-5-7 CPU 유니트 이름 기능 및 시리얼 ID 기능(8-45페이지)'

*1. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.

8-5-1 사용자 프로그램 실행용 ID 인증 기능

개요

사전에 CPU 유니트에 특정 ID('사용자 프로그램 실행용 ID')를 설정해두고, 그와 동일한 ID를 링크로 설정한 사용자 프로그램만 실행할 수 있습니다.



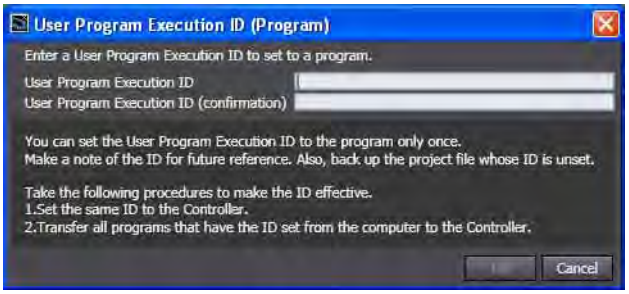
이것으로 어떤 사용자 프로그램을 다른 CPU 유니트(하드웨어)에서는 사용할 수 없도록 할 수 있습니다.

또한, 이 기능은 보호 기능과 달리 프로그램을 표시 및 편집할 수는 있습니다.

조작 방법

1 **아직 사용자 프로그램 실행용 ID를 할당하지 않은 프로젝트 파일을 반드시 미리 백업해둡니다.**

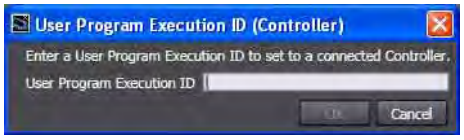
2 **Sysmac Studio에서 오프라인으로 사용자 프로그램 측에 사용자 프로그램 실행용 ID를 할당합니다.**



사용상의 주의 사항

한번 사용자 프로그램 실행용 ID를 할당한 사용자 프로그램은 해당 ID를 변경 및 삭제할 수 없습니다.
만일 변경하고자 하는 경우에는 순서 1에서 사전에 백업한 프로젝트 파일을 읽어와서 다시 다른 사용자 프로그램 실행용 ID를 할당하십시오. 삭제하고자 하는 경우에는 순서 1에서 사전에 백업한 프로젝트 파일을 사용하십시오.

3 **Sysmac Studio를 온라인 연결하여 CPU 유니트 측에 2에서 할당한 사용자 프로그램 실행용 ID를 등록합니다.**



이벤트 로그에 사용자 프로그램 실행용 ID의 CPU 유니트에 대한 등록이 기록됩니다.
이 때, 사용자 프로그램 실행용 ID가 이미 등록된 경우에도 덮어쓰기로 등록됩니다.

4 **동일한 사용자 프로그램 실행용 ID를 할당한 사용자 프로그램을 CPU 유니트에 전송합니다.**
만일, 사용자 프로그램 측과 CPU 유니트 측의 사용자 프로그램 실행용 ID가 일치하지 않은 경우 또는 다른 한쪽에 존재하지 않는 경우에는 운전 모드로 변경했을 때 ID 비교 에러(전체 정지 이상 레벨의 컨트롤러 이상)가 발생하여 운전할 수 없습니다.

사용상의 주의 사항

한번 사용자 프로그램 실행용 ID를 할당한 CPU 유니트는 해당 ID를 삭제 및 읽을 수 없습니다. 만일 삭제하고자 하는 경우에는 CPU 유니트를 메모리 전체 클리어하십시오.

- **'ID 비교 에러' 발생 시의 조작**

CPU 유니트 측의 사용자 프로그램 실행용 ID가 올바르지 않거나 또는 등록되지 않은 경우:

Sysmac Studio에서 CPU 유니트에 온라인 연결하여

1 **올바른 사용자 프로그램 실행용 ID를 CPU 유니트에 등록하십시오.**

2 **컨트롤러 전원을 OFF→ON하거나 또는 Sysmac Studio에서 리셋 조작을 하십시오.**

사용자 프로그램 측의 사용자 프로그램 실행용 ID가 올바르지 않거나 또는 할당되지 않은 경우:

1 **백업했던 프로젝트 파일을 Sysmac Studio에서 읽어서 올바른 사용자 프로그램 실행용 ID를 할당하십시오.**

2 **Sysmac Studio를 CPU 유니트에 온라인 연결하여 사용자 프로그램을 전송하십시오.**

3 **컨트롤러 전원을 OFF→ON하거나 또는 Sysmac Studio에서 컨트롤러 리셋 조작을 하십시오.**

- **기타 사양**

사용자 프로그램 측의 사용자 프로그램 실행용 ID를 삭제하고자 하는 경우:

백업했던 프로젝트 파일을 Sysmac Studio에서 읽으십시오.

CPU 유니트 측의 사용자 프로그램 실행용 ID를 삭제하고자 하는 경우:

Sysmac Studio를 CPU 유니트에 온라인 연결하여 메모리 전체 클리어하십시오.

사용자 프로그램 측의 사용자 프로그램 실행용 ID를 확인하고자 하는 경우:

보안의 관점에서 Sysmac Studio 등에서 사용자 프로그램에 할당한 사용자 프로그램 실행용 ID를 확인할 수 없도록 되어 있습니다. 백업해둔 프로젝트 파일을 바탕으로 다시 할당하십시오.

CPU 유니트 측의 사용자 프로그램 실행용 ID를 확인하고자 하는 경우:

보안의 관점에서 Sysmac Studio 등에서 CPU 유니트에 등록한 사용자 프로그램 실행용 ID를 확인할 수 없습니다. 메모리 전체 클리어한 후 다시 등록하십시오.

사양

- **사용자 프로그램 실행용 ID의 비교의 사양**

비교 타이밍

CPU 유니트는 운전 시작 시 CPU 유니트에 등록된 사용자 프로그램 실행용 ID와 사용자 프로그램에 할당된 사용자 프로그램 실행용 ID를 비교합니다.

비교 조건

비교 조건은 다음과 같습니다.
A, B는 ID를 의미합니다.

CPU 유니트 측에 등록한 사용자 프로그램 실행용 ID	사용자 프로그램 측에 할당한 사용자 프로그램 실행용 ID	이상	운전
A	A	없음	가능
없음	없음		
없음	A	ID 비교 에러	불가능
A	없음		
A	B		

비교 불일치 시의 동작

비교 불일치인 경우에 'ID 비교 에러'(전체 정지 이상 레벨의 컨트롤러 이상)가 되고, CPU 유니트는 운전할 수 없습니다.
이상을 해제하기 위해서는 컨트롤러의 전원을 OFF→ON으로 하거나 Sysmac Studio에서 컨트롤러 리셋을 실행해야 합니다.

- **사용자 프로그램 실행용 ID로 사용할 수 있는 문자의 사양**

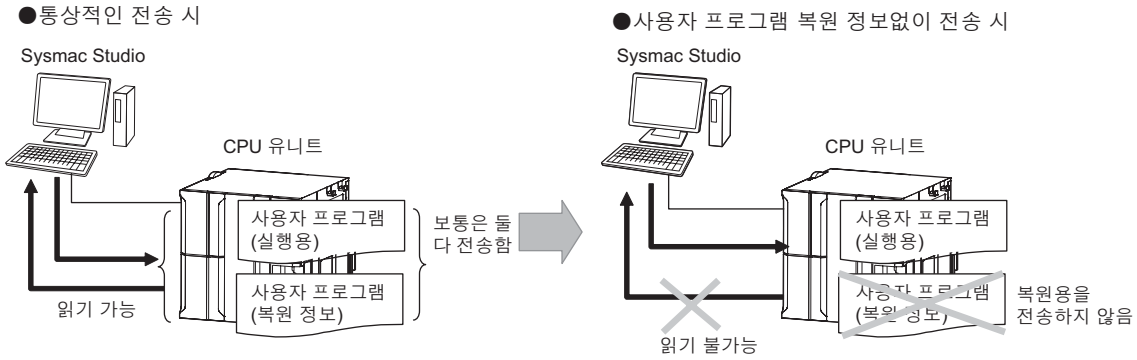
사용할 수 있는 문자	대문자/소문자 구별	최대 크기(NULL 포함하지 않음)
0~9, A~Z, a~z	사용	8~32문자

8-5-2 사용자 프로그램 복원 정보 없는 전송 기능

CPU 유니트에 프로그램 복원 정보(소스)를 전송하지 않는 것으로 다른 PC에서 업로드했을 때 프로그램을 표시할 수 없도록 합니다.

개요

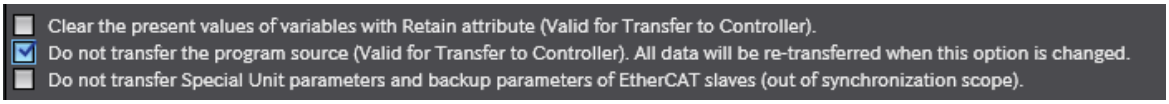
통상적으로는 Sysmac Studio에서 CPU 유니트에 사용자 프로그램을 전송할 때에 복원을 위한 정보도 전송합니다.
이 기능은 이 때 복원을 위한 정보를 전송하지 않는 것으로 사용자 프로그램 읽기를 못하게 하는 기능입니다.



현장에서 사용자 프로그램을 보수하는 일이 없고, 사용자 프로그램 데이터의 도용을 방지하고자 할 때 사용합니다.

조작 방법

사용자 프로그램을 CPU 유니트에 전송할 때 Sysmac Studio의 동기 창에서 '프로그램의 소스를 전송 안 함' 설정을 확인하고 '전송[PC–컨트롤러]' 버튼을 클릭합니다.

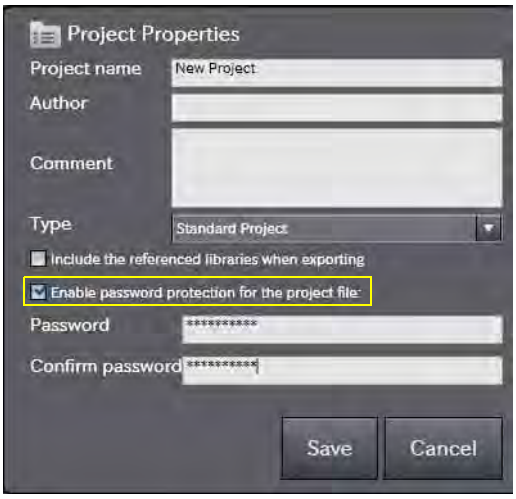


8-5-3 프로젝트 파일 전체 보호 기능

프로젝트 파일에 비밀번호 보호를 설정하여 사용자 자산 정보를 보호합니다.

조작 방법

프로젝트에 비밀번호를 설정할 때의 조작을 설명합니다.
프로젝트 파일에 이름을 붙여 저장할 때 '프로젝트 파일의 비밀번호 보호 유효'를 선택하면 비밀번호를 설정할 수 있게 됩니다.



비밀번호를 설정한 프로젝트를 열 때의 조작을 설명합니다.
비밀번호 설정된 프로젝트 파일을 열 때 또는 가져올 때 [비밀번호 입력] 사용자 대화 상자가 표시됩니다.



이 기능의 자세한 조작 방법은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

8-5-4 데이터 보호 기능

프로젝트의 데이터 중에 보호하고자 하는 데이터에 보호 기능을 부여하여 사용자 자산 정보를 보호합니다.

버전 관련 정보

데이터 보호 기능은 CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.

개요

프로젝트의 데이터 중에 보호하고자 하는 데이터의 단위로 보호를 설정하고, 데이터에 대한 액세스를 제한합니다.
보호를 설정할 때 액세스 제한 강도를 3종류 중에서 선택하여 설정합니다.
액세스 제한을 설정한 데이터에 액세스하려면 보호의 일시 해제가 필요합니다. 일시 해제의 유효 기간은 실행한 조작에 따라 다릅니다.

보호 대상 데이터

보호 설정할 수 있는 데이터는 다음과 같습니다.

- 래더도 프로그램(프로그램/펑션/펑션 블록 공통)
- ST 프로그램(프로그램/펑션/펑션 블록 공통)
- 캠 프로파일

액세스 제한의 종류

액세스 제한의 강도는 다음 3종류 중에서 선택합니다. 단, 캠 프로파일은 변경 금지만 선택할 수 있습니다.

- 복제, 표시, 변경 금지
- 표시, 변경 금지
- 변경 금지

데이터에 대한 액세스 조작과 액세스 제한의 종류별 제한과 보호를 일시 해제했을 때의 유효 기간의 관계는 다음과 같습니다.

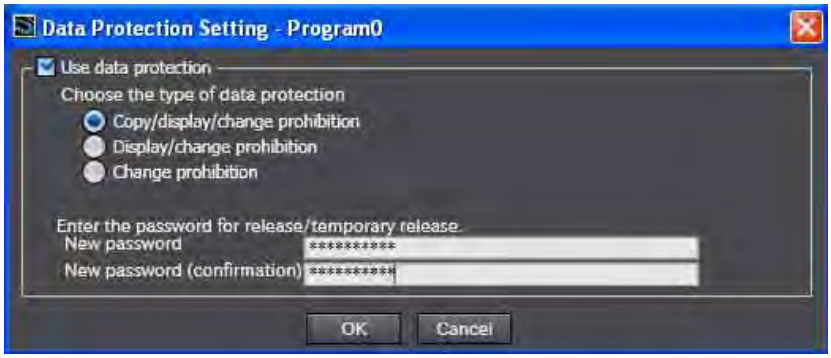
(○: 조작 제한 가능, −: 조작 제한 불가능)

데이터 액세스 조작	액세스 제한의 종류			일시 해제 유효 기간
	복제, 표시, 변경 금지	표시, 변경 금지	변경 금지	
대상물 표시	○	○	−	프로젝트가 열려 있는 동안
대상물 인쇄	○	○	−	
대상물 변경	○	○	○	
대상물 복사	○	−	−	조작마다 일시 해제가 필요
비교 결과(개요) 표시	−	−	−	이 조작은 제한되지 않습니다.
비교 결과(세부 내용) 표시	○	○	−	프로젝트가 열려 있는 동안
이벤트 로그 및 상호 참조에서 내용으로 점프	○	○	−	
라이브러리에 등록	○	−	−	일시 해제할 수 없습니다.

조작 방법

이 기능은 Sysmac Studio를 오프라인으로 조작합니다.
설정 내용은 프로젝트에 저장됩니다. Sysmac Studio의 동기 기능으로 프로젝트를 전송하면 전송원(PC/컨트롤러)인 대상 데이터의 데이터 보호 설정이 전송대상(컨트롤러/PC)에 전송됩니다.

보호 설정하려면 Sysmac Studio의 [컨트롤러]–[보안]–[데이터 보호 설정/해제]에서 설정합니다.



보호 일시 해제하려면 [컨트롤러]–[보안]–[변경 금지 일시 해제]를 선택합니다.



보호 일시 해제를 종료하려면 [컨트롤러]–[보안]–[변경 금지 일시 해제 종료]를 선택합니다.

이 기능의 자세한 조작 방법은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

8-5-5 조작 권한 인증 기능

개요

조작 실수에 의해 장치나 사람에게 위해를 가할 가능성이 있으므로 조작 권한에 맞춰 온라인에서의 조작 기능을 제한하는 기능입니다.
예를 들어 다음과 같은 경우입니다.

- I/O 모니터링 : 쓰기, 강제값 리프레시 등
- 컨트롤러 조작 : 동작 모드 변경, 온라인 편집, 축 시운전 등

Sysmac Studio에서CPU 유니트에 조작 권한 비밀번호를 등록하고, 온라인 연결 시 조작 권한 비밀번호를 입력하면 조작 권한 카테고리에 따른 온라인에서의 조작 기능을 사용할 수 있게 됩니다.

관리자가 조작 권한별로 비밀번호를 설정하고, 기술에 따라 조작 권한명과 비밀번호를 사용자에 연락하여 운용합니다.

조작 권한 관련 설정의 조작의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

조작 방법

조작 권한 인증 기능은 Sysmac Studio의 [컨트롤러]–[보안]–[조작 권한의 설정]을 선택하여 실행합니다.
조작 방법의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

사양

● **조작 권한의 종류**

Sysmac Studio의 조작 권한의 종류는 권한이 큰 순서대로 다음 5종류가 있습니다.

한글명	영어명	비밀번호 설정
관리자	Administrator	필요
설계자*1	Planning Engineer	임의*2[조작 권한 인증 설정] 사용자 대화 상자의 비밀번호 입력 생략 시의 조작 권한에서 필요 여부를 설정합니다.
보전자	Maintenance	
조작자*1	Operator	
관찰자*1	Observer	불필요

*1. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.
*2. [조작 권한 인증 설정] 사용자 대화 상자의 비밀번호 입력 생략 시의 조작 권한에서 필요 여부를 설정합니다. 생략 시의 조작 권한보다 상위의 권한에는 비밀번호가 필요합니다. 동일하거나 하위의 권한에는 비밀번호가 필요 없습니다.

● **대상이 되는 온라인 조작 기능의 예**

온라인 조작의 종류별 조작 권한의 가능 여부의 예시는 다음과 같습니다.

자세한 내용은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

(○: 조작 가능, △: 조작 시 매회 인증 실시, ×: 조작 불가능)

Status 모니터링 기능(예)	관리자	설계자	보전자	조작자	관찰자
이상 모니터링(트러블 슈팅)	○	○	○	○	○

I/O 모니터링 기능(예)	관리자	설계자	보전자	조작자	관찰자
I/O 모니터링(읽기)	○	○	○	○	×
I/O 모니터링(쓰기)	○	○	○	△	×
설정/리셋	○	○	○	△	×
강제값 리프레시	○	○	○	×	×

컨트롤러 조작 기능(예)	관리자	설계자	보전자	조작자	관찰자
운전 모드/프로그램 모드	○	○	△	×	×
온라인 편집	○	○	△	×	×
컨트롤러 리셋	○	○	×	×	×
이상 해제(트러블 슈팅)	○	○	○	△	×
축 시운전, 축 시운전 재개	○	○	△	×	×
사용자 프로그램 실행용 ID(컨트롤러)	○	×	×	×	×
CPU 유니트의 쓰기 방지	○	○	○	×	×

● **비밀번호 사양**

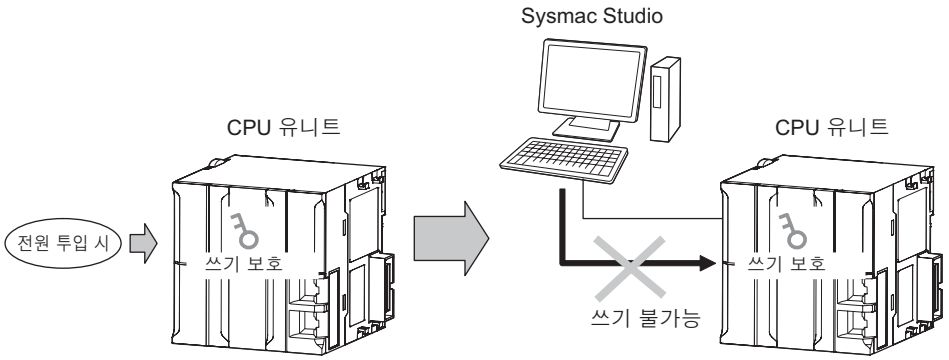
항목	내용
유효 문자 수	8문자 이상이고 32문자 이하
사용 가능 문자	반각 영숫자(대·소문자 구별)

8-5-6 CPU 유니트에 대한 쓰기 방지 기능

사용자 프로그램 자산 보호 또는 잘못된 조작 방지를 위해 CPU 유니트 내의 각종 데이터를 쓸 수 없도록 하는 기능입니다. 다음 2가지가 가능합니다.

- **CPU 유니트의 전원 투입 시 쓰기 방지 설정**

  CPU 유니트의 전원 투입 시 자동으로 쓰기 방지를 유효로 하는 기능입니다.

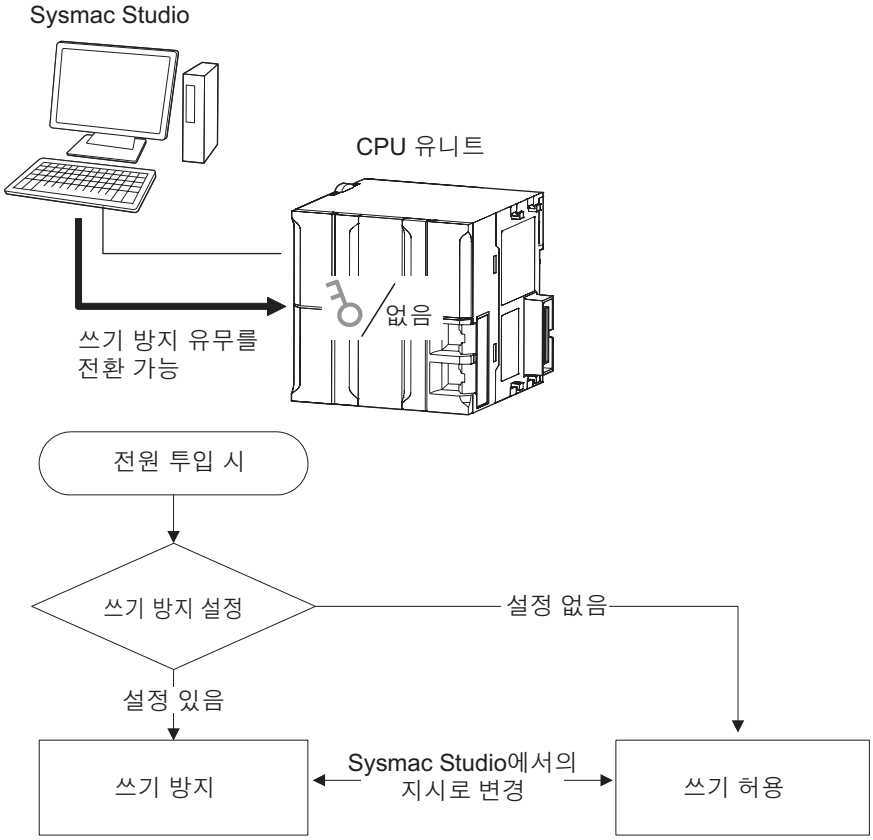


Sysmac Studio에서의 조작 방법은 Sysmac Studio의 '구성·설정'–'컨트롤러 설정'–[동작 설정]에서 전원 투입 시 자동으로 쓰기 방지를 실행할지 여부를 설정합니다.

설정 대항목	설정 중항목	설정 소항목	기능 설명	설정값
동작 설정- 동작 설정 탭- 전반 설정	보안 설정	전원 투입 시의 쓰기 방지	쓰기 방지 유효로 함/유효로 안 함을 설정합니다.	쓰기 방지 사용 안 함 쓰기 방지 사용

- **Sysmac Studio에서 쓰기 방지 '설정'/'해제'의 전환**

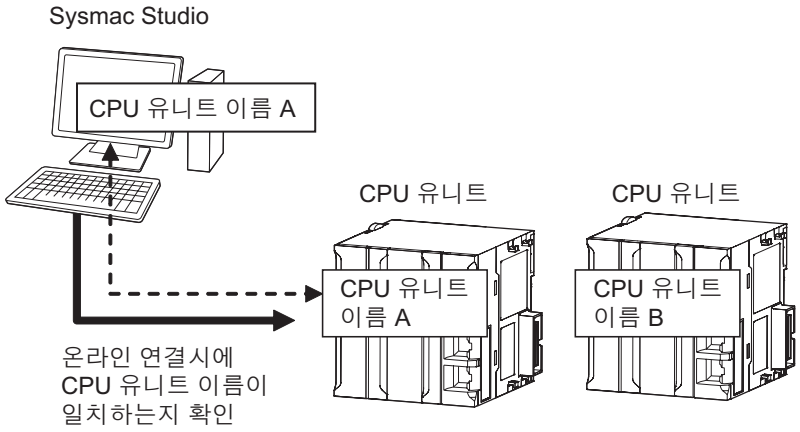
  Sysmac Studio에서 온라인 연결하고 [컨트롤러] 메뉴에서 [보안]-[CPU 유니트에 대한 쓰기 방지]에서 '설정'/'해제'를 전환합니다.



8-5-7 CPU 유니트 이름 기능 및 시리얼 ID 기능

개요

CPU 유니트 내에 CPU 유니트 이름을 등록하십시오.
Sysmac Studio에서의 온라인 연결 시에 프로젝트상의 CPU 유니트 이름과 연결대상 CPU 유니트의 CPU 유니트 이름이 일치하는지 여부를 확인할 수 있습니다.
이것으로 Sysmac Studio에서 잘못된 CPU 유니트에 대한 연결을 방지할 수 있습니다. 특히 EtherNet/IP 네트워크를 통한 조작인 경우 유효합니다.



그리고 CPU 유니트 이름 이외에 CPU 유니트의 생산 정보에 기초한 시리얼 ID를 식별할 수도 있습니다 (옵션).

설정 방법

1 **Sysmac Studio에서 프로젝트 작성 시 CPU 유니트 이름을 설정합니다.**
CPU 유니트 이름은 다음과 같이 표시됩니다.



이름을 변경하는 경우에는 컨트롤러의 아이콘을 마우스 오른쪽 버튼으로 클릭하고 '이름 변경'에서 편집합니다.

2 **CPU 유니트와 처음으로 온라인 연결할 때 Sysmac Studio는 CPU 유니트 내에 CPU 유니트 이름을 기억할지 여부를 묻습니다.**

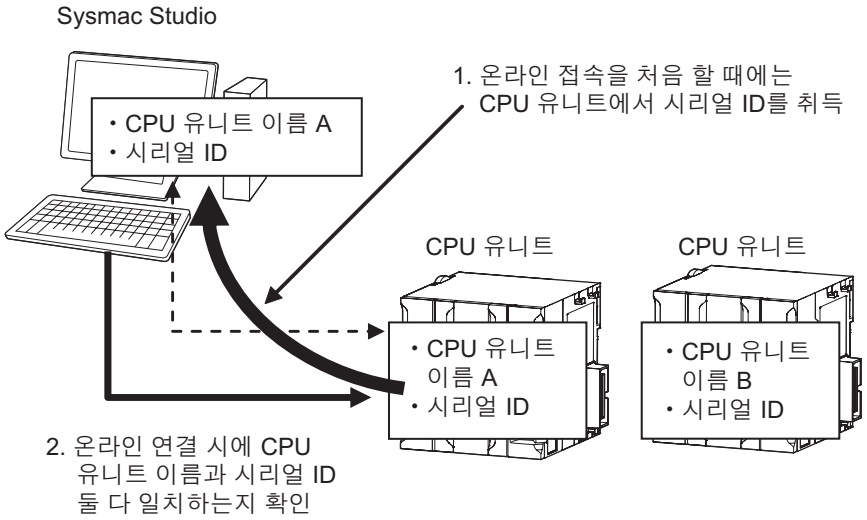
3 **그 후, CPU 유니트와 온라인 연결 시에Sysmac Studio는 프로젝트의 CPU 유니트 이름과 연결하려고 하는 CPU 유니트의 CPU 유니트 이름을 비교합니다.**
일치하지 않는 경우에는 경고 대화 상자가 표시되고, 그대로 연결을 계속할지 여부를 묻습니다.

참고

NetworkConfigurator에서는 EtherNet/IP 포트에 이름을 붙일 수 있습니다.

시리얼 ID 기능

Sysmac Studio가 처음으로 온라인 연결할 때 CPU 유니트의 생산 정보에 기초한 시리얼 ID를 취득하여 프로젝트에 기억시킬 수도 있습니다.
그 후, Sysmac Studio가 온라인 연결할 때, CPU 유니트 이름과 시리얼 ID의 양쪽을 비교합니다. 그에 따라 더 엄밀한 CPU 유니트의 비교가 가능합니다.



8-6 디버그 기능

여기에서는 디버그 기능에 대해 설명합니다.

NJ/NX 시리즈 컨트롤러에는 다음과 같은 디버그 기능이 있습니다.
- 강제값 리프레시
- 현재값 변경
- 온라인 편집
- 데이터 트레이스
- 미분 모니터링

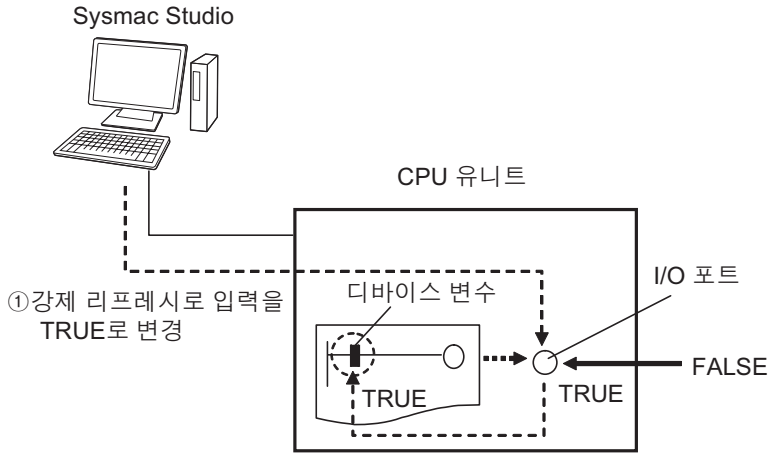
8-6-1 강제값 리프레시

개요

강제값 리프레시는 사용자가 디버그 시 등에 외부에서의 입력 및 외부에 대한 출력을 Sysmac Studio에서 지정한 강제값으로 리프레시할 수 있는 기능입니다.
강제값 리프레시는 지정한 디바이스 변수가 아니라 디바이스 변수에 할당된 I/O 포트에 대해 실행됩니다.
강제값 리프레시로 지정한 상태는 Sysmac Studio에서 강제 상태 해제 지시를 할 때까지 계속 유지합니다. (CPU 유니트의 상태 변화에 대한 유지/해제에 대해서는 뒤에 설명하는 '강제값 리프레시 유지/해제'를 참조하십시오).
또한, 이상 정지 시, 메모리 전체 클리어 시, 모드 변경 시, 전원 OFF 시, 다운로드 시에는 유지되지 않고 클리어됩니다.

- **입력**

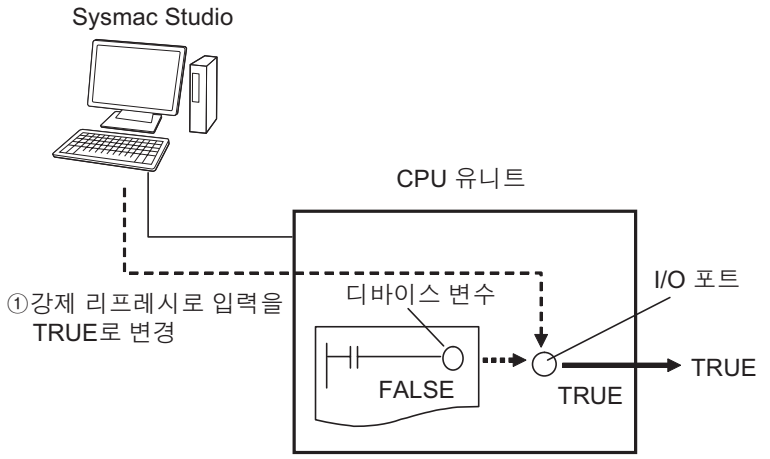
  외부에서의 입력에 관계없이 I/O 포트와 디바이스 변수가 강제값 리프레시로 지정한 상태가 됩니다.



②I/O 포트, 디바이스 변수가 TRUE가 됨

- **출력**

  I/O 포트와 외부에 대한 출력이 강제값 리프레시로 지정한 상태가 됩니다.
  사용자 프로그램에 있는 I/O 포트에 할당된 디바이스 변수의 상태는 강제값 리프레시로 지정한 상태가 아니라 사용자 프로그램의 결과에 따라 변화합니다.



대상 영역

강제값 리프레시는 다음과 같은 I/O 포트 또는 CJ 유니트용 메모리에 대해 실행할 수 있습니다.

- EtherCAT 슬레이브의 I/O 포트
- NX102 및 NX1P2 CPU 유니트의 NX 버스 기능 모듈이 취급하는 데이터로 구성되는 I/O 포트
- NX1P2 CPU 유니트의 내장 I/O의 I/O 포트
- NX1P2 CPU 유니트의 아날로그 입출력용 옵션 보드의 I/O 포트
- CJ 시리즈 기본 I/O 유니트의 I/O 포트
- CJ 시리즈 고기능 유니트의 I/O 포트
- 변수에서 AT(할당대상) 지정된 DeviceNet 슬레이브 및 CompoNet 슬레이브의 입출력 릴레이

강제값 리프레시를 래더 편집기 및 감시 창에서 실행하는 경우에 변수를 통해 I/O 포트 또는 CJ용 유니트 메모리의 상태를 변경합니다.

(주의) NX102 및 NX1P2 CPU 유니트의 NX 버스 기능 모듈이 취급하는 데이터로 구성되는 I/O 포트는 NX 유니트별 I/O 데이터의 I/O 포트만 대상이 됩니다. NX 버스 기능 모듈이 NX 버스 마스터로서 관리하는 NX 유니트의 Status의 I/O 포트는 대상에서 제외됩니다.

(주의) CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

동시에 강제값 리프레시할 수 있는 수

강제값 리프레시할 수 있는 개수는 다음과 같습니다.

- CJ 시리즈 유니트: 합계 64점
- EtherCAT 슬레이브: 합계 64점

위의 개수는 외부 입출력 접점의 수입니다. 예를 들어 복수의 변수가 하나의 외부 입출력 접점을 할당대상으로 지정한 경우에는 1개로 셉니다.

용도

- **입력**
  - 입력 신호를 의사적으로 줘서 사용자 프로그램의 디버그를 실행하는 경우
  - 고장 시에만 있을 수 있는 상태를 만드는 경우(배타적인 2접점 동시 ON/OFF 등)

- **출력**
  - 배선 확인을 위해 출력을 ON/OFF하는 경우
  - 사용자 프로그램의 결과에 관계없이 동작시키지 않고 출력을 강제적으로 OFF하는 경우

조작 방법

다음과 같은 화면에서 조작할 수 있습니다.

- 프로그램의 화면(래더도 언어)
- I/O 맵
- 감시 창

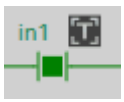
- **래더 편집기에서의 강제값 리프레시 조작 순서**

1 **[컨트롤러]의 [모니터]에서 모니터링을 ON합니다.**

2 **멀티 뷰 탐색기의 [프로그래밍]을 통해 대상의 래더도 프로그램 또는 래더도 펑션 블록/펑션을 두 번 클릭합니다.**
래더 편집기에 회로가 모니터링 상태로 표시됩니다.

3 **강제값 리프레시로On하는 접점, 코일을 선택하여 마우스 오른쪽 버튼으로 클릭–[강제값 리프레시]–[TRUE]를 선택합니다. 또는 강제값 리프레시로 Off하고자 하는 접점, 코일을 선택하여 마우스 오른쪽 버튼으로 클릭–[강제값 리프레시]–[FALSE]를 선택합니다.**

4 **래더 편집기의 대상 접점, 코일이 강제적으로 On 또는 Off 상태가 되고, On인 경우에는 도통 상태, Off인 경우에는 비전도 상태가 됩니다.**
접점, 코일에는 다음과 같은 강제 상태 마크가 표시됩니다.



래더도

강제 상태의 'TRUE', 'FALSE' 표시는 강제값 리프레시로 지정한 상태를 표시하며, 접점 및 코일의 현재값을 나타내는 것은 아닙니다.

강제 상태 마크	조작
	강제값 리프레시(TRUE)를 지정한 경우
	강제값 리프레시(FALSE)를 지정한 경우

참고

강제값 리프레시한 변수의 AT(할당대상) 지정대상과 동일한 메모리 주소를 지정한 변수가 있었던 경우에 해당 변수에도 강제 상태 마크가 표시됩니다.

동작 모드, 전원 OFF와의 관련

- **강제값 리프레시할 수 있는 동작 모드**
  프로그램 모드, 운전 모드 중 어느 모드에서도 실행할 수 있습니다.
  단, 전체 정지 이상 레벨의 컨트롤러 이상 발생 중에는 강제할 수 없습니다.

- **동작 모드 변경 시 또는 전원 OFF 시의 강제값 리프레시 상태**
  운전 모드⇔프로그램 모드 변경 시 또는 전원 OFF 시 강제값 리프레시 상태는 해제됩니다.

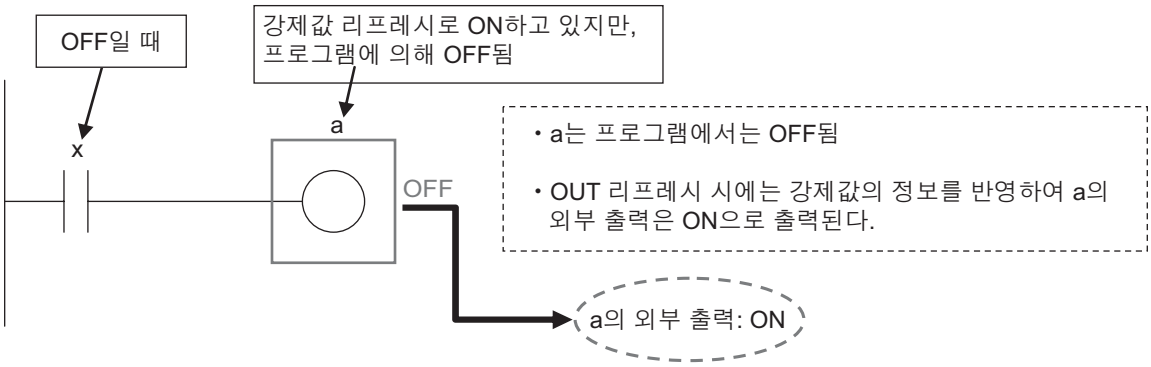
강제값 리프레시 유지/해제

CPU 유니트의 상태 변화에 대해 강제값 리프레시의 유지/해제는 다음과 같습니다.

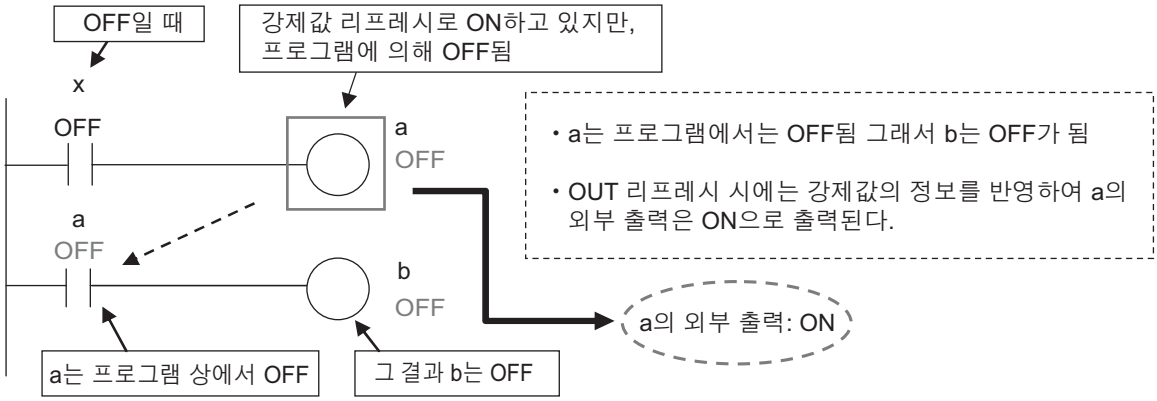
상태 변화		강제값 리프레시 상태
전원 투입 시		해제
동작 모드 변경 시	**운전 모드→프로그램 모드** **프로그램 모드→운전 모드**	해제
다운로드 후		해제
전체 정지 이상 레벨의 컨트롤러 이상 발생 시		해제
온라인 편집 시		유지

강제값을 프로그램에서 사용하는 경우의 유의 사항

사용자 프로그램에서 강제값 리프레시를 실행하는 경우에는 강제값 리프레시 대상으로 한 변수는 사용자 프로그램에 의해 덮어씁니다. 따라서 강제값 리프레시로 지정한 상태로 유지되지 않습니다.
단, 외부에 대한 리프레시는 사용자 프로그램의 변수 상태로 관계없이 강제값 리프레시로 지정한 값으로 계속 출력합니다.
강제값을 프로그램에서 사용할 때는 프로그램의 변수와 외부 출력의 상태에 차이가 있으므로 주의하십시오.
예) a를 강제값 리프레시로 ON한 경우



다시 그 접점을 받았을 때:



안전상의 중요 사항

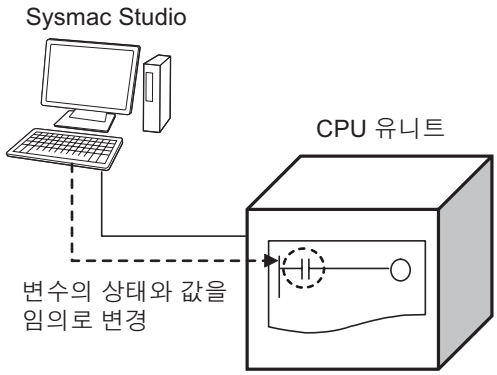
- 강제값 리프레시 조작은 설비에 영향이 없는지 확인한 후에 실행하십시오.
- 강제값 리프레시는 사용자 프로그램 실행 결과에 관계없이 설정된 값으로 덮어쓰기해서 I/O 리프레시를 하는 기능입니다.
  따라서 I/O 리프레시의 대상이 아닌 접점은 이 기능에 따라 일단 설정된 값이 되지만 그 후 사용자 프로그램에 의해 다시 쓰입니다.

강제 상태의 차이에 따라 제어 대상이 예기치 못한 동작을 할 위험이 있습니다.

8-6-2 현재값 변경

개요

현재값 변경은 프로그램 및 설정이 올바르게 동작하는지 여부를 확인하기 위해 Sysmac Studio에서 프로그램 및 설정에서 사용하고 있는 변수 값 변경 및 접점 및 코일을 ON/OFF(TRUE/FALSE)할 수 있는 기능입니다.



안전상의 중요 사항

설비에 영향이 없는지 확인한 후 현재값을 변경하십시오.

용도

- **접점 및 코일의 ON/OFF(TRUE/FALSE)를 임의로 변경하는 경우**
  임의의 BOOL 형식 변수의 상태를 ON/OFF합니다. 이 상태는 사용자 프로그램에서의 덮어쓰기로 실행 상태로, 모드 변경 및 전원 재시작에 따라 초기값의 상태으로 돌아갑니다.
  래더 편집기, 감시 창, I/O 맵에서 실행할 수 있습니다.

- **변수 값을 임의로 변경하는 경우**
  사용자 정의 변수나 시스템 변수, 디바이스 변수의 현재값을 임의의 값으로 변경하여 반영하는 것을 말합니다. 감시 창에서 수행할 수 있습니다.

안전상의 중요 사항

설비에 영향이 없는지 확인한 후 현재값을 변경하십시오.

조작 방법

현재값 변경은 다음과 같은 화면에서 조작할 수 있습니다. 각 화면에서의 자세한 조작 방법은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

- 프로그램 화면(래더도 언어, ST 언어)
- I/O 맵
- 감시 창

현재값 변경으로 외부로의 출력 상태를 변경하는 경우의 유의 사항

현재값 변경에 의해 CJ 시리즈 기본 출력 유니트 또는 EtherCAT 출력 슬레이브의 I/O 포트에 출력하는 상태를 변경하는 경우에는 다음에 주의하십시오.

- **운전 모드 시 I/O 맵에서 현재값을 변경하는 경우**

  I/O 맵에서 I/O 포트 값을 변경해도 사용자 프로그램 실행 결과로 덮어씁니다.
  따라서 현재값 변경으로 변경한 값으로는 외부에 출력되지 않습니다.
  I/O 포트의 값을 임의의 값으로 변경하여 외부에 출력하고자 하는 경우에는 강제값 리프레시로 실행하십시오.

- **프로그램 모드 시 감시 창에서 현재값을 변경하는 경우**

  감시 창에서 디바이스 변수(*)(외부 변수, 로컬 변수)의 값을 변경해도 외부로는 출력되지 않습니다.
  외부에 임의의 값을 출력하고자 하는 경우에는 다음과 같이 실행하십시오.
  - 강제 리프레시하는 경우
  - 감시 창에서 디바이스 변수(*)(글로벌 변수)에 대해 현재값을 변경하는 경우

  * CJ 시리즈 기본 출력 유니트 또는 EtherCAT 출력 슬레이브의 I/O 포트에 할당한 디바이스 변수입니다. 또한, CJ 시리즈 기본 출력 유니트가 할당되는 출력 릴레이에 AT 지정한 글로벌 변수인 경우도 동일합니다.

- **CJ 시리즈 기본 출력 유니트의 출력 릴레이에 상당하는 I/O 메모리 주소에 대해 직접 외부에서 쓴 경우의 주의 사항**

  태그 데이터 링크에 의해 CJ 시리즈 기본 출력 유니트가 할당되는 출력 릴레이에 상당하는 I/O 메모리 주소에 직접 액세스하고, I/O 메모리 주소의 상태를 변경해도 변경한 값은 사용자 프로그램 실행 결과로 덮어씁니다.
  따라서 태그 데이터 링크에 따라 직접 I/O 메모리 주소에 쓴 값으로 외부에 출력할 수 없습니다.

(주의) CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

8-6-3 온라인 편집

여기에서는 온라인 편집의 개요를 설명합니다. 자세한 내용은 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.

개요

온라인 편집은 Sysmac Studio를 사용하여 직접CPU 유니트 내의 사용자 프로그램 일부를 변경 및 추가하는 기능입니다.

다음을 선택하여 온라인 편집할 수 있습니다.

- POU(프로그램, FB, FUN)
  래더도 언어로 기술된 프로그램인 경우에는 섹션을 선택합니다.
- 글로벌 변수

용도

CPU 유니트의 운전을 중지하지 않고 사용자 프로그램을 변경하고자 하는 경우 사용합니다.

Sysmac Studio에서의 조작

- **온라인 편집 조작**

1 **온라인 편집 대상을 선택합니다.**

2 **[프로젝트] 메뉴에서 [온라인 편집]–[시작]을 선택합니다.**

3 **변경을 실행합니다.**

4 **[프로젝트] 메뉴에서 [온라인 편집]–[변경 전송]을 선택합니다.**

5 **반영 확인을 실행합니다.**

6 **온라인 편집 후의 사용자 프로그램에서 동작이 시작됩니다.**

⚠ 주의

입출력 타이밍이 어긋나도 영향이 없다는 것을 확인한 다음 온라인 편집하십시오. 온라인 편집을 실행하면 태스크 실행 시간이 태스크 주기를 상회하거나 외부와의 입출력이 갱신되지 않아서 입력 신호를 읽지 못하는 일이 발생하거나 출력 타이밍이 불규칙해 질 수 있습니다.



사용상의 주의 사항

- 온라인 편집에서 편집한 회로에 포함되는 미분 명령의 미분 상태는 초기화됩니다.
- 온라인 편집 내용을 반영할 때, 태스크 실행 시간이 연장됩니다. 태스크 주기는 온라인 편집에 의해 '태스크 주기 초과'가 되지 않도록 설정하십시오.
- 온라인 편집 반영 중(*)에 CPU 유니트가 전원 OFF된 경우에 전원 OFF 복귀 시 전체 정지 이상 레벨의 컨트롤러 이상('사용자 프로그램/컨트롤러 구성·설정 시 전송 에러', '잘못된 사용자 프로그램/컨트롤러 구성·설정', '비휘발성 메모리가 복구됨' 중 하나)이 발생합니다. 이 때, 사용자 프로그램을 다시 다운로드하십시오.
- 온라인 편집 반영 중(*)에 MC_SaveCamTable(캠 테이블 저장) 명령을 실행하지 마십시오. 실행하면 온라인 편집에 의한 변경이 올바르게 반영되지 않을 가능성이 있습니다.

  * 반영 확인 창에서 [예]를 선택한 후 '반영 확인 창'이 닫힐 때까지를 가리킵니다. 단, CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합에서는 '반영 확인 창' 다음에 표시되는 '내장 비휘발성 메모리 백업 중 창'이 닫힐 때까지를 가리킵니다.

8-6-4 데이터 트레이스

프로그램 없이 지정한 변수를 샘플링하는 기능입니다. Sysmac Studio로 읽어서 확인 및 파일을 저장할 수 있습니다. 장치 셋업, 운용/보수로 사용할 수 있습니다.

아래에서 데이터 트레이스의 개요를 설명합니다. 조작 방법은 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.

사용상의 주의 사항

다음 변수를 데이터 트레이스 샘플링 대상으로 하는 경우에 정확한 데이터를 샘플링할 수 없는 경우가 있습니다.

- 모션 제어 시스템 변수를 제외하고, 데이터 크기가 16비트 이상인 구조체 멤버
- 데이터 크기가 16비트 이상의 배열 요소

위의 변수로 데이터를 샘플링하는 경우에는 다음 중 하나를 실행하십시오.

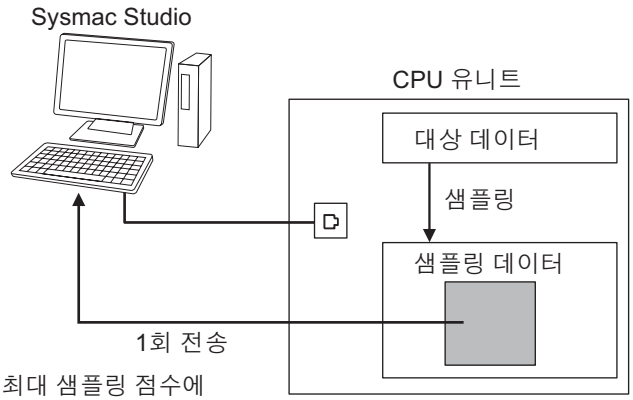
- 64비트 이외의 기본 데이터 형식의 내부 변수에 복사하고, 복사한 변수를 데이터 트레이스 대상으로 함
- 변수의 태스크 간 배타 제어 설정 기능을 사용하고, 데이터 트레이스의 샘플링 대상으로 하는 태스크를 갱신 태스크로 함

트레이스 방법은 다음 2종류가 있습니다.

- **트리거 트레이스**

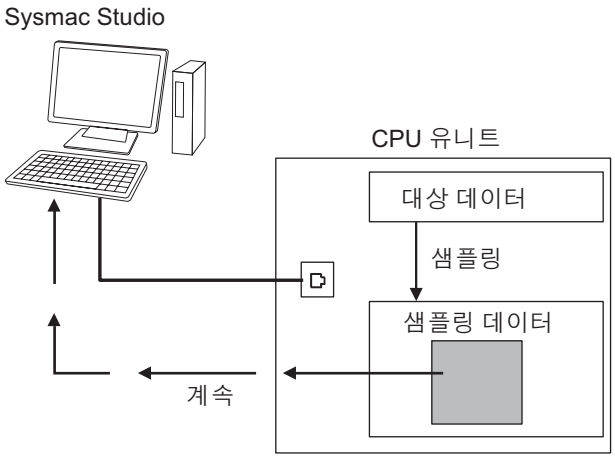
  트리거 조건을 설정하여 사건이 발생한 전후의 데이터를 기록합니다. 최대 샘플링 점수에 도달하면 자동으로 샘플링을 종료합니다. Sysmac Studio와 연결하지 않을 때도 트리거 조건 성립 시 데이터 트레이스하고, Sysmac Studio와 연결한 후 업로드할 수 있습니다.

  - 변수의 현재값 변경 상황에서 프로그램의 흐름을 확인할 수 있습니다.
  - 의도하지 않는 변수 값의 변경 요인을 조사할 때 사용합니다.



- **연속 트레이스**

  트리거 없이 샘플링을 시작하고, 10,000점을 초과해도 샘플링을 계속합니다. 샘플링 된 데이터는 순차적으로 PC에 전송되어 파일에 저장됩니다.
  표시 버퍼 크기가 되면 자동으로 CSV 파일에 저장합니다.
  이것으로 트레이스를 시작한 후 정지할 때까지의 장기간의 트레이스 결과 데이터를 복수의 CSV 파일로 저장할 수 있습니다.



데이터 트레이스의 사양

데이터 트레이스의 사양은 다음과 같습니다.

항목		내용
데이터 트레이스의 종류	**트리거 트레이스**	샘플링을 시작하는 트리거 조건을 설정하고, 조건 성립 전후의 데이터를 기록합니다.
	연속 트레이스	샘플링된 데이터는 순차적으로 PC에 전송되어 파일에 저장됩니다.
샘플링 타이밍	**지정 태스크 주기**	태스크를 지정하고, 그 태스크의 주기를 샘플링 주기로 합니다.*1
	고정 시간 간격	입력한 시간을 샘플링 주기로 합니다. 단, 시간은 프라이머리 주기 태스크의 정수 배로 반올림됩니다.
	트레이스 샘플링 명령	사용자 프로그램에서TraceSamp 명령이 실행된 타이밍에 샘플링을 실행합니다.
샘플링 대상 설정*2	**최대 대상 수**	NX701-□□□□: 192변수 NX102-□□□□: 48변수 NX1P2-□□□□: 48변수 NJ501-□□□□: 192변수 NJ301-□□□□: 48변수 NJ101-□□□□: 48변수
	데이터 형식	기본 데이터 형식(문자열은 제외) 배열(요소를 지정) 열거 형식 구조체 형식/공용체 형식(멤버를 지정)
최대 레코드 수		변수 하나당 10,000점
트리거 위치 설정		전체 트레이스 시간(수)에서의 트리거 위치를 설정합니다.
트리거 설정	**조건의 데이터 형식**	기본 데이터 형식(시각, 시간 길이, 날짜, 문자열은 제외) 배열(요소를 지정), 구조체/공용체(멤버를 지정)
	조건식*3	다음 조건식 성립 시 트레이스가 시작됩니다. BOOL: 상승, 하강 BOOL 이외: =, >, ≧, <, ≦, ≠*4
	Sysmac Studio에서 지시	트리거ON 버튼을 클릭했을 때 트레이스가 시작됩니다.
	데이터 트레이스 트리거 조건 성립 명령	TraceTrig 명령을 실행했을 때 트레이스가 시작됩니다.
	판정 타이밍	• 샘플링 타이밍 설정이 트레이스 샘플링 명령 이외인 경우에는 지정한 태스크 주기에 트리거 판정이 1회 실행됩니다. • 샘플링 타이밍 설정이 트레이스 샘플링 명령인 경우에는 명령 실행 시 트리거 판정이 실행됩니다.
	딜레이값	트리거 성립 전/성립 후의 샘플링 수 비율을 슬라이더로 설정합니다. (예: 20%/80%)

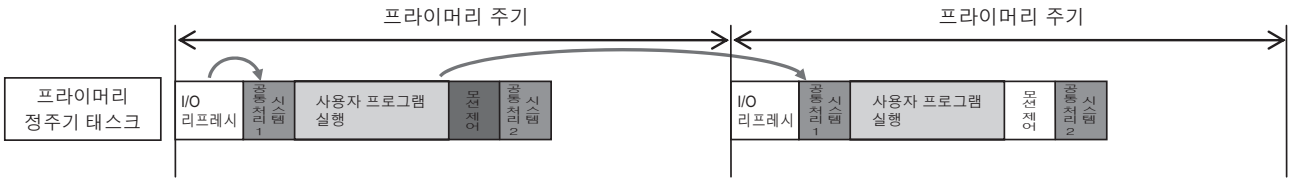
항목		내용
트레이스 시작	**Sysmac Studio에서 지시**	Sysmac Studio의 '실행 버튼'을 누르면 시작됩니다.
	운전 시 시작 설정	운전 시작(프로그램 모드→운전 모드) 시 자동으로 트레이스가 시작됩니다.
트레이스 정지	**트리거 트레이스**	• 샘플링 상한값인 10,000점에 도달하면 정지합니다. • Sysmac Studio의 '정지 버튼'을 누르면 정지합니다.
	연속 트레이스	• 상한 도달 시의 설정이 '트레이스 정지'인 경우에는 상한 크기 또는 상한 기간에 도달하면 정지합니다. • Sysmac Studio의 '정지 버튼'을 누르면 정지합니다.
연속 트레이스 전용 설정	**데이터 저장 기간 상한**	연속 트레이스 데이터의 저장 기간 상한을 설정합니다.
	데이터 저장 크기 상한	연속 트레이스에서 저장되는 파일의 합계 크기 상한을 설정합니다.
	파일 1개당 데이터 수	연속 트레이스 시, 하나의 파일에 저장하는 데이터 수를 설정합니다.
	파일 저장 장소	연속 트레이스 시 데이터를 저장하는 파일을 작성하는 장소를 지정합니다.
	파일명 접두사	자동으로 붙여지는 파일명의 선두 부분의 이름을 지정합니다.
	상한 도달 시의 설정	저장 기간 상한 또는 저장 크기 상한에 달했을 때의 동작을 지정합니다. • 트레이스 정지 • 오래된 파일부터 삭제하고 계속 실행
트레이스 결과 표시	**그래프 표시**	X축: 시간, Y축: 변수의 값이 되는 그래프를 표시합니다. BOOL 형식과 그 이외의 변수를 하나의 그래프에 표시할 수도 있습니다.
	테이블 표시	각 변수의 최대값, 최소값, 평균값 및 지정된 시간의 값을 테이블 형식으로 표시합니다.
	3D 모션 모니터링 표시	3D 공간에 가상 기구 모델을 설치하고, 모션축의 명령 위치 및 현재 위치에 맞춰서 기구의 동작을 가상 표시합니다.
트레이스 데이터 내보내기	**CSV 파일 내보내기**	트레이스 결과 및 트레이스 번호 이외의 설정을 CSV 형식 파일에 저장합니다.
동시에 기동할 수 있는 데이터 트레이스*5		NX701-□□□□: 4개 NX102-□□□□: 2개 NX1P2-□□□□: 2개 NJ501-□□□□: 4개 NJ301-□□□□: 2개 NJ101-□□□□: 2개
트레이스 데이터 가져오기		CSV 형식으로 저장된 트레이스 결과를 현재 표시된 그래프에 겹쳐서 표시합니다.
저장		트레이스 결과를 트레이스 설정과 함께 프로젝트 내에 저장합니다.
인쇄		그래프를 인쇄합니다. Sysmac Studio의 인쇄 기능이 사용됩니다.

*1. NJ 시리즈 및 NX701에서 데이터 크기가 64비트 이상인 내부 변수를 샘플링 대상으로 하는 경우에 대상 내부 변수를 정의한 프로그램을 할당한 태스크를 지정하십시오. 다른 태스크를 지정하면 변수 값의 동시성이 확보되지 않을 수 있습니다.
*2. 시스템 정의 변수의 EN, ENO, P_off, P_on, P_CY, P_First_RunMode, P_First_Run, P_PRGER 및 펑션 블록의 인스턴스에서 사용되는 입출력 변수, 펑션 내의 변수는 샘플링 대상에서 제외됩니다.
*3. 데이터 트레이스 시작 시점에서 트리거의 조건식이 성립 상태라도 데이터 트레이스는 시작하지 않습니다.
*4. 여러 조건식의 조합은 불가능합니다. 또한, 비교 상수의 유효 범위는 조건식 좌변의 변수 형식의 리터럴 표현의 유효 범위에 좌우됩니다.
*5. NX701 및 NJ501인 경우에는0~3, NX102, NX1P2, NJ301 및 NJ101인 경우에는0~1의 트레이스 번호가 설정됩니다. 이 번호를 사용하여 명령 실행 및 시스템 정의 변수로의 액세스를 실행합니다.

데이터 트레이스의 동작

데이터 트레이스 처리(샘플링 및 트리거 판정)는 I/O 리프레시 처리 후에 사용자 프로그램 실행 전의 '시스템 공통 처리 1'에서 실행됩니다.

예) 샘플링 타이밍에 '프라이머리 정주기 태스크'를 지정한 경우에는 다음 그림의 '시스템 공통 처리 1' 타이밍에 데이터 트레이스가 실행됩니다.



참고

태스크 처리에서는 'I/O 리프레시, 사용자 프로그램, 모션 제어'는 동일한 태스크 주기이지만 데이터 트레이스 표시에서는 '사용자 프로그램, 모션 제어, 다음 태스크 주기의 I/O 리프레시'가 동일한 태스크 주기로 표시됩니다. 또한 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"에 기재된 타이밍 차트는 태스크 주기를 기준으로 기재했으므로 데이터 트레이스의 표시와는 다릅니다.

지정 태스크 주기로 샘플링한 경우의 데이터 트레이스 동작 및 실행 결과의 표시 예는 다음과 같습니다.

예 1)

센서의 입력 신호에 할당한 변수 'Sensor1'이 TRUE가 되었을 때, 변수 'SysRun'을 TRUE로 하는 프로그램을 작성한 경우

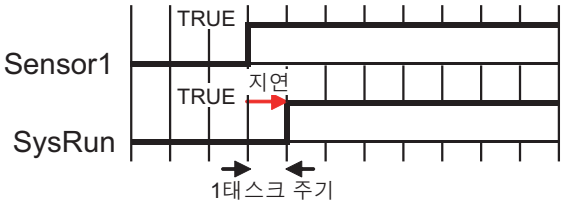


데이터 트레이스의 동작 및 실행 결과 표시는 다음과 같습니다.

(1) 시스템 공통 처리 1의 데이터 트레이스 처리에서 'Sensor1'의 TRUE를 취득함
(2) 사용자 프로그램 중에서 'SysRun'을 TRUE로 함
(3) 다음 프라이머리 주기의 시스템 공통 처리 1의 데이터 트레이스 처리에서 'SysRun'의 TRUE를 취득함

따라서 데이터 트레이스에서의 표시는 'Sensor1'보다 1 태스크 주기 지연으로 'SysRun'이 TRUE가 됩니다.

데이터 트레이스 표시

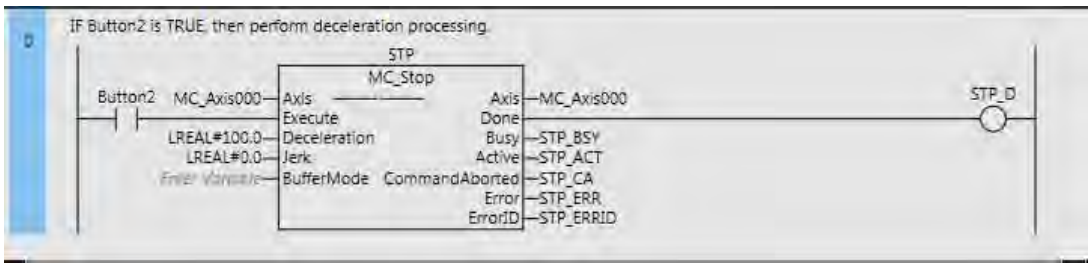


참고

사용자 프로그램 중에서 변수 값을 변경한 경우에 데이터 트레이스 표시에서는 값의 변화와 I/O 리프레시에 의한 출력 처리가 동일한 태스크 주기로 변화합니다.

예 2)

속도 제어 동작 중에 푸시 버튼의 입력 신호에 할당한 변수 'Button2'가 TRUE가 되면 축0('MC_Axis000')을 감속 정지시키는 프로그램을 작성한 경우

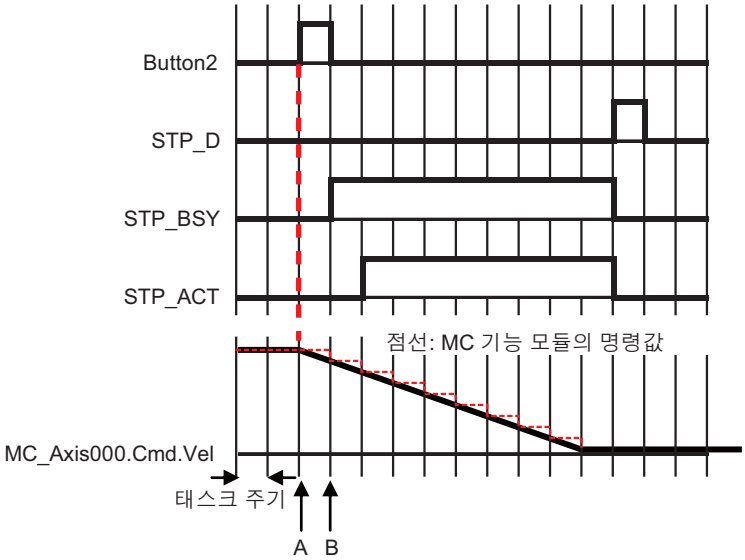


데이터 트레이스의 동작 및 실행 결과 표시는 다음과 같습니다.

(1) 시스템 공통 처리 1의 데이터 트레이스 처리에서 'Button2'의 TRUE를 취득함
(2) 사용자 프로그램 중에서 'STP_BSY'를 TRUE로 하고, 모션 제어 기능 모듈이 감속 처리를 실행
(3) 다음 프라이머리 주기의 시스템 공통 처리 1의 데이터 트레이스 처리에서 STP_BSY의 TRUE와 모션 변수의 취득을 실행
(4) 사용자 프로그램 중에서 'STP_ACT'를 TRUE로 함
(5) 다음 프라이머리 주기의 시스템 공통 처리 1의 데이터 트레이스 처리에서 STP_ACT의 TRUE를 취득함

모션 제어 기능 모듈의 명령값은 사용자 프로그램 중의 'STP_BSY'가 TRUE가 되고, 모션 제어 기능 모듈에서 감속 처리가 시작된 시점에서 변화를 시작합니다(다음 그림 B의 시점). 명령값은 프라이머리 정주기 태스크에 동기하여 계단 형태로 변화합니다.

이에 대해 데이터 트레이스는 이전 주기의 값과 다음 주기의 값을 직선으로 보간합니다. 따라서 모션 변수의 명령 속도 'MC_Axis000.Cmd.Vel'은 명령값이 변화하기 1주기 전, 즉 'Button2'가 TRUE가 되는 타이밍(다음 그림 A의 시점)에 감속하도록 표시됩니다. 감속에서 1주기 지연되어 'STP_BSY'가 TRUE, 다시 1주기 지연되어 'STP_ACT'가 TRUE가 되게 표시됩니다.



참고

모션 제어 명령을 포함하는 펑션 블록에서는 펑션 블록 실행 시작 시 입력 파라메터의 값이 입력 변수에 설정되고, 펑션 블록 실행 종료 시 출력 변수의 값이 출력 파라메터에 설정됩니다('펑션 블록(FB) 내의 변수 사양(6-12페이지)' 참조). 데이터 트레이스의 표시도 입력 파라메터와 입력 변수, 출력 파라메터와 출력 변수가 동일한 태스크 주기로 변화합니다.

관련 시스템 정의 변수

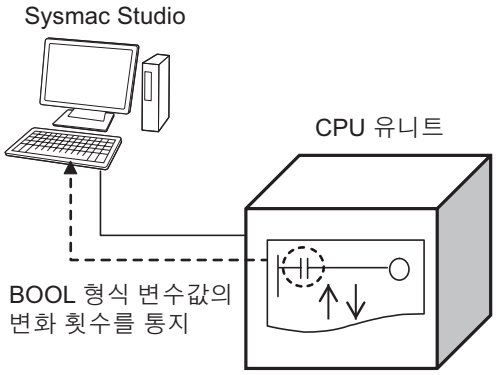
변수명	멤버명	이름	기능	데이터 형식	R/W
_PLC_TraceSta[0..3]*1				_sTRACE_STA	R
	.IsStart	트레이스 실행 중 ON 플래그	트레이스를 시작했을 때 TRUE가 됩니다.	BOOL	R
	.IsComplete	트레이스 완료 시 ON 플래그	트레이스 실행이 완료됐을 때 TRUE가 됩니다. 다음으로 트레이스를 시작했을 때, FALSE가 됩니다.	BOOL	R
	.IsTrigger	트레이스 트리거 모니터링 플래그	트레이스 트리거 조건이 성립됐을 때 TRUE가 됩니다. 다음으로 트레이스를 시작했을 때, FALSE가 됩니다.	BOOL	R
	.ParamErr	트레이스 파라메터 이상 플래그	트레이스 시작 시에 트레이스 설정이 이상일 때 TRUE가 됩니다. 정상일 때 FALSE가 됩니다.	BOOL	R

*1. 데이터 트레이스 0~3에 대응합니다.

(주의) 이 시스템의 정의 변수는 사용자 프로그램상에서는 사용할 수 없습니다. 사용자 프로그램에서 데이터 트레이스의 상태를 참조하는 경우에는 GetTraceStatus 명령을 사용하십시오.

8-6-5 미분 모니터링

미분 모니터링은 지정한 BOOL 형식 변수의 값이 지정한 조건에 일치한 횟수를 통지하는 기능입니다. 프라이머리 정주기 태스크의 태스크 주기(프라이머리 주기)별로 조건 일치 판정을 실행하고, 그 때까지의 조건 일치 횟수를 통지합니다.



버전 관련 정보

미분 모니터링은 CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 사용할 수 있습니다.

용도

외부에서의 입력 신호 및 사용자 프로그램 내의 접점의 ON/OFF 유무 및 횟수를 확인할 수 있습니다. 설비 셋업 시 및 가동 중의 동작 불량의 원인 조사 등에 도움이 됩니다.

미분 모니터링의 사양

미분 모니터링의 사양은 다음과 같습니다.

항목		사양
미분 모니터링 조건	설정 수	최대 8
	지정 변수	• BOOL 형식 • BOOL 형식 배열의 1요소 • 구조체, 공용체의 BOOL 형식의 1멤버
	조건식	• 상승(FALSE→TRUE) • 하강(TRUE→FALSE)
조건 일치 판정	타이밍	프라이머리 주기에 1회
	횟수 카운트	지정 변수별로 조건 일치 횟수를 카운트
시작, 정지	시작 조건	Sysmac Studio에서 지시
	정지 조건	• Sysmac Studio에서 지시 • 전체 정지 이상 레벨의 컨트롤러 이상 발생 • 사용자 프로그램의 다운로드 • 메모리 전체 클리어 • Sysmac Studio의 온라인 연결 해제
동작 모드		• 운전 모드 • 프로그램 모드

미분 모니터링 조건

미분 모니터링의 설정 대상이 되는 변수 및 변화 종류를 미분 모니터링 조건이라고 합니다. 이들의 사양을 설명합니다.

- **설정 수**

  최대 8입니다. 즉, 8개의 변수 값의 조건 일치 횟수를 병행하여 검출할 수 있습니다.

- **지정 변수**

  미분 모니터링 대상으로 지정할 수 있는 변수의 데이터 형식은 다음과 같습니다.
  - BOOL 형식
  - BOOL 형식 배열의 1요소
  - 구조체, 공용체의 BOOL 형식의 1멤버

  배열 전체, 구조체 전체, 공용체 전체는 지정할 수 없습니다.

  또한, 다음과 같은 변수의 종류를 지정할 수 있습니다.

변수의 종류		지정 가능 여부
시스템 정의 변수		가능*1
준 사용자 정의 변수		가능
사용자 정의 변수	글로벌 변수	가능
	프로그램 내에서 사용하는 변수	가능
	펑션 블록 내에서 사용하는 변수	가능*2
	펑션 내에서 사용하는 변수	불가능

*1. 다음과 같은 변수는 제외합니다.
EN, ENO, P_Off, P_CY, P_First_RunMode, P_First_Run, P_PRGER

*2. 입출력 변수는 제외합니다.

- **조건식**

  검출하는 변수의 변화 조건을 조건식이라고 합니다. 조건식은 다음 2종류가 있으며, 둘 중 하나를 선택합니다. 조건식은 변수별로 지정할 수 있습니다.
  - 상승(FALSE→TRUE)
  - 하강(TRUE→FALSE)

사용상의 주의 사항

예를 들어 조건식을 '상승'으로 설정했다고 가정합니다. 미분 모니터링 기능을 시작한 시점에서 지정한 변수의 값이 TRUE라도 상승이라고 판정되지 않습니다. 변수의 값이 일단 FALSE가 되고 난 후에 TRUE가 되었을 때 상승이라고 판정됩니다.

조건 일치 판정 타이밍과 횟수 카운트

지정한 변수의 조건 일치 판정은 프라이머리 주기마다 실행합니다. 이전 판정 시의 변수 값과 이번 판정 시의 변수 값을 비교합니다. 그 결과, 지정한 조건식에 일치하는 변수 값의 변화가 있으면 그 횟수를 카운트합니다.

- 조건 일치 횟수 카운트는 변수별로 실행합니다.
- 카운트 값은 미분 모니터링 시작 시 0으로 리셋됩니다.
- 변수별로 카운트 값을 0으로 리셋할 수 없습니다.

사용상의 주의 사항

- 프라이머리 주기 내에 조건식에 일치하는 값의 변화가 여러 번 있어도 1회의 프라이머리 주기에 하나만 카운트됩니다.
- 이전 판정 시와 이번 판정 시의 변수 값이 동일하면 그 사이에 값의 변화가 있었다고 해도 조건식에 일치했다고 간주되지는 않습니다.

시작 조건과 정지 조건

미분 모니터링을 시작하려면 Sysmac Studio에서 지시합니다.
미분 모니터링을 정지하려면 보통 Sysmac Studio에서 지시합니다. 그러나 다음과 같은 경우도 미분 모니터링은 정지합니다.

- 전체 정지 이상 레벨의 컨트롤러 이상 발생
- 사용자 프로그램의 다운로드
- 메모리 전체 클리어
- Sysmac Studio의 온라인 연결 해제

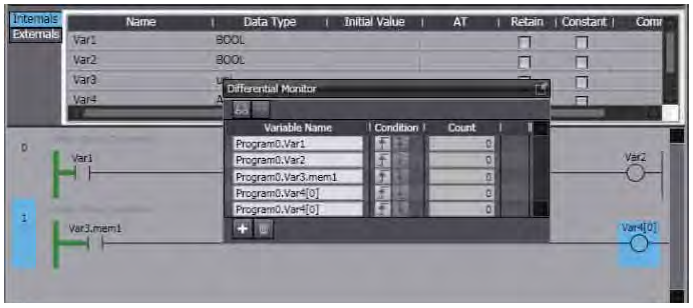
조작 방법

미분 모니터링의 조작 방법은 다음과 같습니다. 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

1 **Sysmac Studio의 [표시]–[미분 모니터링]을 선택합니다.**

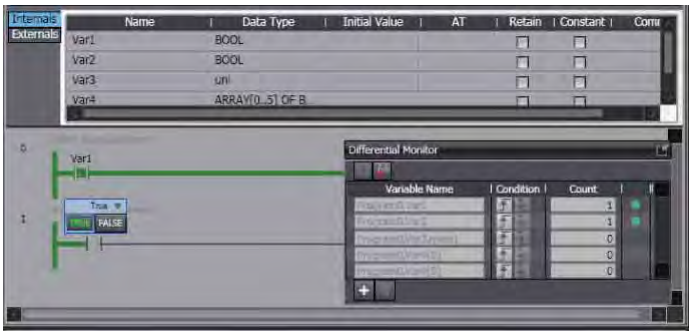
2 **미분 모니터링의 대상으로 지정할 수 있는 변수를 선택한 후 마우스 오른쪽 버튼으로 클릭하고 [미분 모니터링에 추가]를 선택합니다.**

3 **미분 모니터링 창에서 변수별로 미분 모니터링의 조건식을 설정합니다.**



4 **사용자 프로그램을 실행합니다.**

미분 모니터링 창에 변수별로 조건 성립 횟수가 표시됩니다.



미분 모니터링 기능의 유의 사항

미분 모니터링 기능을 사용할 때는 다음 사항에 유의하십시오.

- **미분 모니터링 실행 중 Sysmac Studio와의 통신 중단**

  미분 모니터링 실행 중에 통신 케이블의 단선, Sysmac Studio의 이상 종료 등의 원인으로 Sysmac Studio와의 통신이 중단되었다고 가정합니다. 그 경우에 CPU 유니트는 미분 모니터링 실행을 계속합니다. 다시미분 모니터링을 실행하는 경우에는 Sysmac Studio와의 통신을 재개한 후, 일단 미분 모니터링을 정지하십시오.

- **미분 모니터링의 복수 실행 불가능**

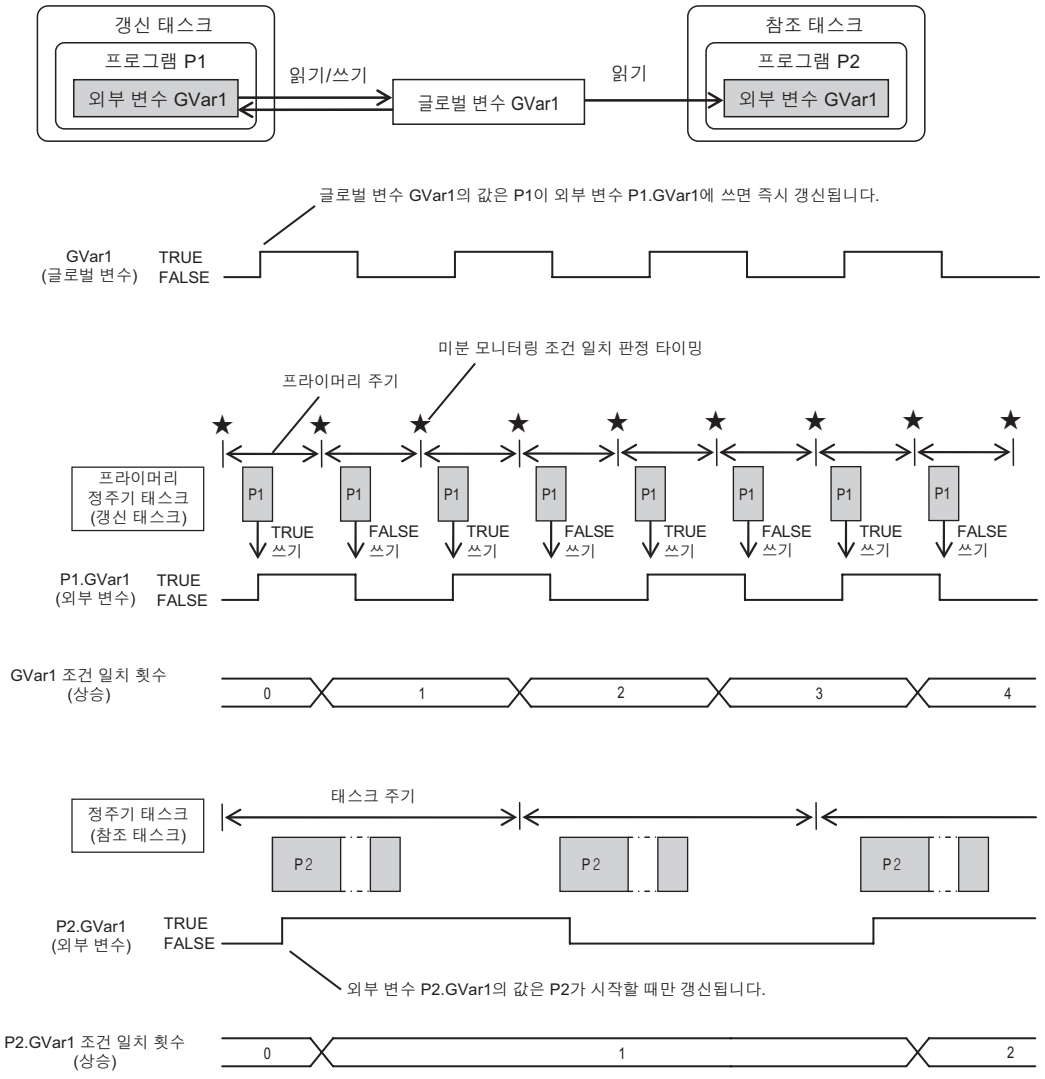
  미분 모니터링은 복수의 Sysmac Studio에서 동시에 실행할 수 없습니다.

- **글로벌 변수, 외부 변수의 지정**

  글로벌 변수도 POU 내에서 그것을 지정한 외부 변수도 미분 모니터링의 지정 변수로 할 수 있습니다. 단, 글로벌 변수와 외부 변수는 값의 갱신 타이밍이 다르다는 점에 주의하십시오.
  글로벌 변수는 값을 쓰면 즉시 갱신됩니다. 한편, 외부 변수는 그 외부 변수를 선언한 POU가 기동하는 타이밍에만 갱신됩니다.
  다음 그림에서 설명합니다. 다음 2가지 변수를 모니터링하는 것으로 가정합니다.

변수명	변수의 종류	읽기/쓰기 POU
GVar1	글로벌 변수	프라이머리 정주기 태스크에 할당된 프로그램 P1
P2.GVar1	프로그램 P2에서 GVar1을 지정하여 선언된 외부 변수	정주기 태스크에 할당된 프로그램 P2

  글로벌 변수 GVar1은 프라이머리 정주기 태스크에 할당된 프로그램 P1에서 읽고 쓰여지므로 주기마다 쓰기가 발생한다면 프라이머리 주기로 갱신되는 것입니다. 한편, 외부 변수 P2.GVar1은 정주기 태스크에 할당된 프로그램 P2가 기동하는 타이밍에만 갱신됩니다. 즉 정주기 태스크의 태스크 주기로만 갱신되는 것입니다. 정주기 태스크의 태스크 주기는 프라이머리 주기보다 길기 때문에 P2.GVar1은 GVar1보다 갱신 횟수가 적어집니다.



8-7 이벤트 로그 기능

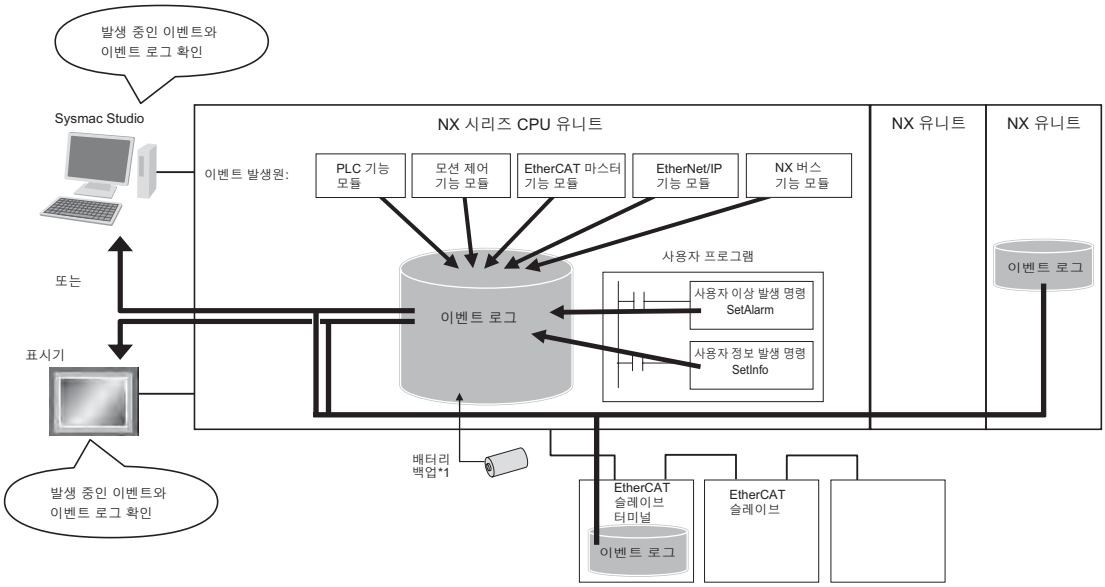
여기에서는 이벤트 로그 기능에 대해 설명합니다.

8-7-1 개요

이벤트 로그 기능은 NJ/NX 시리즈 컨트롤러에서 발생한 이상, 상태 변화 및 사용자 정의 사건과 같은 이벤트(*)를 기록하는 기능입니다.

* '이벤트'는 컨트롤러에서 비계획적으로 발생하는 이상을 포함한 사건입니다. 이상과 컨트롤러 및 사용자 정의에 따라 사용자에게 통지해야 하는 이상이 아닌 정보를 합한 총칭입니다.
  다음과 같은 2종류의 4가지 분류가 있습니다.
  - 컨트롤러 이벤트
    컨트롤러 이상
    컨트롤러 정보
  - 사용자 이벤트
    사용자 이상
    사용자 정보

● **NX 시리즈 CPU 유니트의 경우**

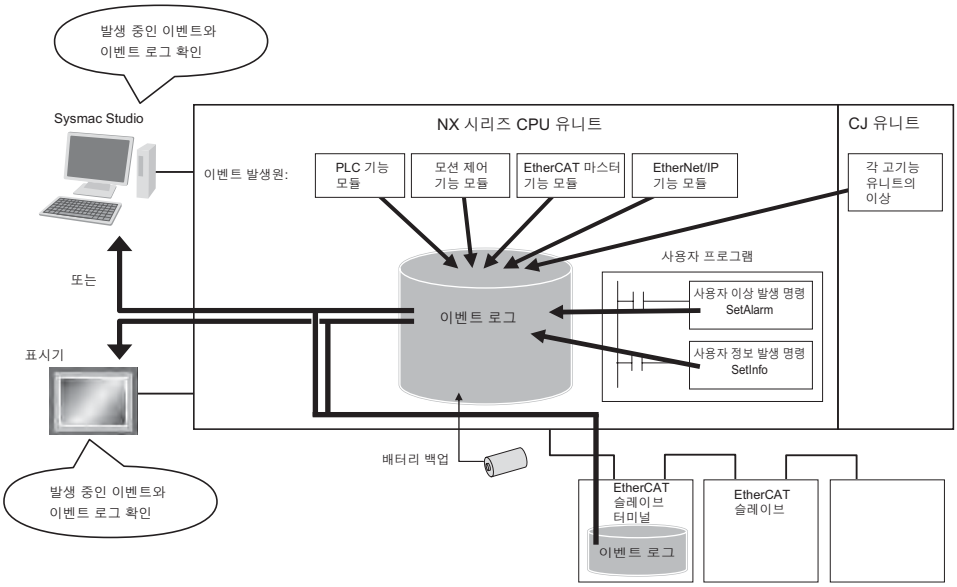


*1. 이벤트 로그는 NX701의 경우에는 배터리 백업 메모리에 저장합니다.
NX102 및 NX1P2의 경우에 비휘발성 메모리에 저장됩니다.

사용상의 주의 사항

- NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.
- NX 유니트를 장착할 수 있는 CPU 유니트는 NX102 및 NX1P2 CPU 유니트뿐입니다.
- 표시기의 문제 해결사 대응 범위는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

● **NJ 시리즈 CPU 유니트의 경우**



NS 시리즈 표시기에서 이벤트를 확인하는 경우에는 NS 시리즈 표시기를 CPU 유니트의 내장 EtherNet/IP 포트로 연결하십시오.

사용상의 주의 사항

- CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
- 표시기의 문제 해결사 대응 범위는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

특징

이벤트 로그 기능에는 다음과 같은 특징이 있습니다.

- 이상 이력에 추가하여 전원 투입/전원 OFF 및 운전 시작 등의 각종 이력도 기록합니다.
- 이 기록을 시계열로 확인할 수 있으므로 문제 발생 시 이상 요인의 추측에 활용할 수 있습니다.

기록할 수 있는 내용

이벤트에는 다음과 같은 종류가 있습니다.

● **시스템 정의의 이벤트(컨트롤러 이벤트)**

컨트롤러가 자동으로 감지하는 이벤트입니다. CPU 유니트의 각 기능 모듈, NX 유니트, NX 시리즈 슬레이브 터미널, EtherCAT 슬레이브, CJ 유니트의 이벤트입니다.
다음과 같은 이벤트가 있습니다.

- 컨트롤러 이상
- 컨트롤러 정보

● **사용자가 정의한 이벤트(사용자 이벤트)**

사용자가 설계하는 어플리케이션 상의 이벤트입니다. 명령 실행에 따라 다음과 같은 이벤트가 가능합니다.

- 사용자 이상
- 사용자 정보

Sysmac Studio 또는 표시기를 사용하여 이벤트 로그를 읽을 수 있습니다.

8-7-2 이벤트 로그 기능의 세부 내용

이벤트 발생원

컨트롤러의 어느 위치에서 이벤트가 발생하고 있는지 식별하는 정보입니다.
컨트롤러 이벤트와 사용자 이벤트 각각의 발생원은 다음과 같습니다.

- **컨트롤러 이벤트 발생원**

  컨트롤러 이벤트에서는 CPU 유니트 내에 존재하는 기능 모듈이 발생원이 됩니다.
  또한 기능 모듈에 따라서는 더 상세한 이벤트 발생원이 존재합니다. 이를 '이벤트 발생원 세부 내용'이라고 합니다.
  다음과 같은 것이 있습니다.

이벤트 발생원	이벤트 발생원 세부 내용
PLC 기능 모듈	명령, 전원, 내장 I/O, 옵션 보드, I/O 버스 마스터, CJ 시리즈 유니트*1
NX 버스 기능 모듈*2	마스터, NX 유니트
모션 제어 기능 모듈	공통, 축, 축 그룹
EtherCAT 마스터 기능 모듈	통신 포트, EtherCAT 마스터, EtherCAT 슬레이브
EtherNet/IP 기능 모듈	통신 포트/통신 포트 1/통신 포트 2, CIP/CIP1/CIP2, FTP, NTP, SNMP

*1. CJ 시리즈의 고기능 유니트/EtherCAT 슬레이브 내의 이상 이력은 대상이 아닙니다. 개별 지원 소프트웨어로 읽습니다.
*2. NX102 및 NX1P2 CPU 유니트만 NX 버스 기능 모듈을 탑재했습니다.

- **사용자 이벤트 발생원**

  사용자 이벤트에서는 PLC 기능 모듈이 이벤트 발생원입니다.

카테고리

이벤트 로그의 분류를 나타내는 정보입니다.
Sysmac Studio 또는 표시기에서 참조할 때의 단위입니다.

이벤트의 종류	이벤트 로그의 카테고리	내용
컨트롤러 이벤트	시스템 로그	컨트롤러 자체에서 감지한 이벤트의 기록입니다. CJ 유니트의 이상도 포함됩니다.
	액세스 로그	사용자 작업에 의해 컨트롤러의 동작에 영향을 준 이벤트의 기록입니다.
사용자 이벤트	사용자 이벤트 로그	사용자가 정의한 이벤트의 기록입니다.

기록 건수

이벤트 로그에는 각각 다음과 같은 레코드 건수만큼 기록할 수 있습니다.
CPU 유니트는 아래에 있는 기록 가능한 건수를 초과해서 이벤트가 발생한 경우에 가장 오래된 정보부터 덮어씁니다.

이벤트의 종류	이벤트 로그의 카테고리	기록 가능 건수					
		NX701	NX102	NX1P2	NJ501	NJ301	NJ101
컨트롤러 이벤트	시스템 로그	2,048	768	576	1,024	512	512
	액세스 로그	1,024	576	528	1,024	512	512
사용자 이벤트	사용자 이벤트 로그	1,024	512	512	1,024	512	512

전원 OFF 유지

전원 OFF가 발생한 경우에 NJ 시리즈 CPU 유니트 및 NX701 CPU 유니트는 배터리에 의해 NX102 및 NX1P2 CPU 유니트는 비휘발성 메모리에 의해 이벤트 로그를 유지합니다.

사용상의 주의 사항

배터리를 사용하여 이벤트 로그를 유지하는 CPU 유니트에서는 배터리 없이는 유지할 수 없으므로 주의하십시오.
필요에 따라 정기적으로 이벤트 로그 내보내기를 실시하십시오.

이벤트 코드

이벤트의 종류에 따라 사전에 시스템이 컨트롤러 이벤트에 할당한 코드 또는 사용자가 사용자 이벤트에 할당한 코드입니다. 컨트롤러 이벤트의 이벤트 코드는 16진수 8자리로 표시됩니다.
발생 중인 이상의 이벤트 코드는 이상 Status 취득 명령을 통해 취득할 수 있습니다.
사용자 이벤트에는 10진수로 1~60000의 이벤트 코드를 부여할 수 있습니다.

이벤트 중요도

각 이벤트에는 중요도를 나타내는 '이벤트 중요도'가 있습니다.
이벤트 중요도의 분류는 이벤트의 종류에 따라 다릅니다. 컨트롤러 이벤트와 사용자 이벤트가 독립적으로 정의됩니다.

- **컨트롤러 이벤트의 경우**

  컨트롤러 이벤트는 제어에 대한 영향의 크기에 따라 중요도를 다음 5단계로 분류합니다.

번호	중요도		분류
1	높음	컨트롤러 이상	전체 정지 이상 레벨
2	↕		부분 정지 이상 레벨
3			가벼운 이상 레벨
4			감시 정보 레벨
5	낮음	컨트롤러 정보	일반 정보 레벨

  이상의 중요도가 높을수록 컨트롤러가 제공하는 기능이 손상된 것이며, 복구에 대한 긴급도도 높아집니다.
  이 분류는 이상 발생 시 Sysmac Studio나 표시기에 표시됩니다.

- **사용자 이벤트의 경우**

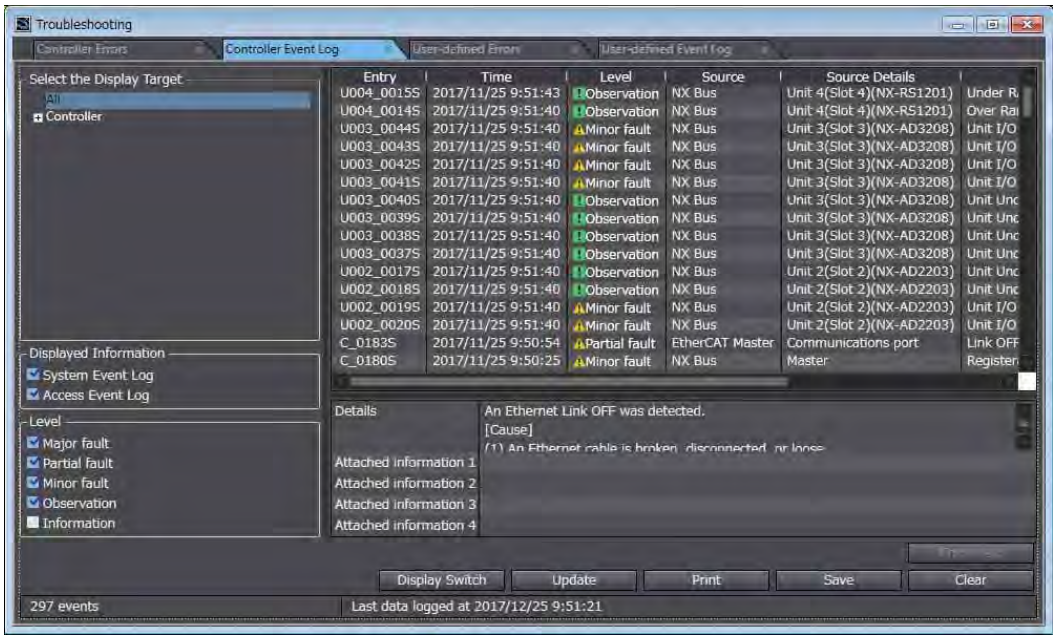
  사용자 이벤트를 NJ/NX 시리즈 시스템이 제공하는 다음과 같은 중요도로 분류할 수 있습니다.
  사용자 이벤트를 어느 중요도에서 발생시킬지 정의합니다.

번호	중요도	분류	의미
1	높음	User fault Level1	어플리케이션상의 사용자 정의 이상을 나타내는 이벤트. 사용자는 SetAlarm(사용자 이상 발생) 명령을 실행하여 이벤트를 발생시킵니다.
2	↕	User fault Level2	
3		User fault Level3	
4		User fault Level4	
5		User fault Level5	
6		User fault Level6	
7		User fault Level7	
8		User fault Level8	
9	낮음	User Information	어플리케이션상의 사용자 정의 정보를 나타내는 이벤트. 사용자는 SetInfo(사용자 정보 발생) 명령을 실행하여 이벤트를 발생시킵니다.

이벤트 로그 표시

Sysmac Studio 또는 표시기는 '컨트롤러 로그', '사용자 로그'의 카테고리별로 이벤트 로그 목록을 표시합니다. '컨트롤러 로그'에는 '액세스 로그'와 '시스템 로그'가 표시됩니다.
또한, Sysmac Studio는 CJ 유니트 또는 EtherCAT 슬레이브 내에 기록된 이상 이력을 표시할 수도 있습니다.

Sysmac Studio에서는 다음과 같이 목록이 표시됩니다. 목록에서 이벤트를 선택하면 해당되는 상세 정보가 표시됩니다.



참고

Sysmac Studio 또는 표시기의 버전에 따라 대응하지 않는 이벤트가 컨트롤러에서 발생한 경우에는 발생원을 '알 수 없음', 이벤트명을 '알 수 없는 이벤트'로 표시합니다. 이벤트 코드와 부속 정보는 그대로 표시됩니다.

이벤트 로그 삭제

- **Sysmac Studio/표시기에서 삭제**

  Sysmac Studio 또는 표시기에서 이벤트 로그의 컨트롤러 이벤트와 사용자 이벤트별로 이벤트 로그를 삭제할 수 있습니다.

사용상의 주의 사항

- Sysmac Studio/표시기에서 CPU 유니트 내의 이벤트 로그를 삭제하는 경우에 지금까지 취득한 모든 이벤트의 정보가 삭제되어도 괜찮은지 확인한 후 실행하십시오. 특히 일반 정보 레벨/감시 정보 레벨의 컨트롤러 이벤트나 사용자 이벤트의 경우에 한 번도 확인하지 않았을 가능성이 있으므로 주의하십시오.
- 표시기에서 이벤트 로그를 삭제하는 경우의 제약 사항에 대해서는 "NJ/NX 시리즈 트러블슈팅 설명서(SBCA-361)"를 참조하십시오.

- **메모리 전체 클리어를 통한 삭제**

  Sysmac Studio에서 NJ/NX 시리즈 CPU 유니트에 대해 메모리 전체 클리어 조작을 실행한 경우에 이벤트 로그를 삭제할지 여부를 선택할 수 있습니다.

이벤트 로그 내보내기

Sysmac Studio 또는 표시기를 사용해서 표시한 이벤트 로그의 내용을 CSV 파일로 내보낼 수 있습니다.

8-7-3 컨트롤러 이벤트(컨트롤러 이상/정보)

개요

NJ/NX 시리즈의 시스템에서 정의한 이상/정보입니다.
NJ/NX 시리즈 시스템이 이상 요인, 정보 요인을 감지하면 발생합니다.

- **컨트롤러 이상**

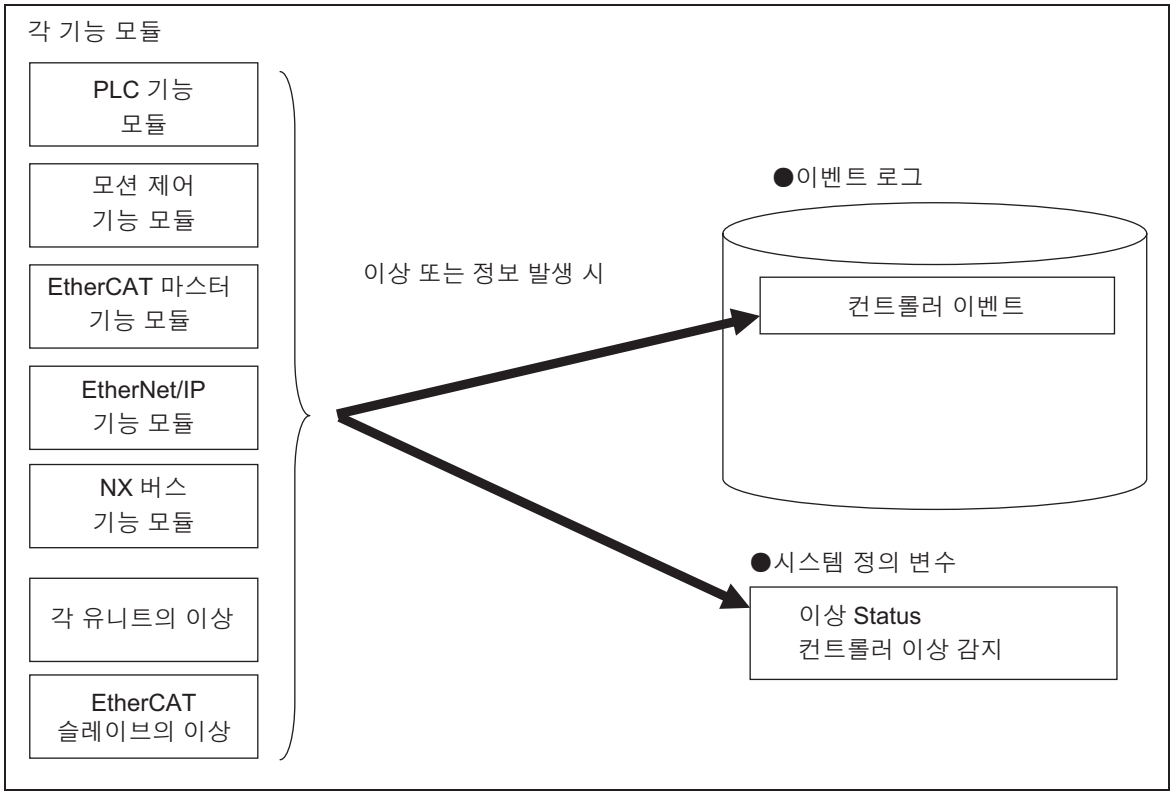
  시스템 정의의 이상입니다.
  '전체 정지 이상 레벨', '부분 정지 이상 레벨', '가벼운 이상 레벨', '감시 정보 레벨'의 컨트롤러 이벤트의 총칭입니다.
  CPU 유니트의 각 기능 모듈, NX 유니트, NX 시리즈 슬레이브 터미널, EtherCAT 슬레이브, CJ 유니트의 각 이상을 감지합니다. 이 때, 컨트롤러 이상이 이벤트 로그에 기록됩니다.
  발생한 컨트롤러 이상의 상황을 사용자 프로그램에서 확인하는 경우에는 이상 Status 취득 명령에 따라 시스템 정의 변수의 '이상 Status 변수'를 참조합니다.

  (주의) CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
  (주의) CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

  컨트롤러 이상은 동작 모드의 변경에 의해 해제되지 않습니다.
  컨트롤러 이상의 자세한 내용은 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

- **컨트롤러 정보**

  이상이 아닌 시스템 정의의 통지 정보입니다. '일반 정보 레벨'의 컨트롤러 이벤트입니다.
  예를 들어 전원 ON/OFF, 운전 시작/정지, Sysmac Studio의 온라인 연결, 사용자 프로그램의 다운로드 등의 이상 이외의 사건입니다.



8-7-4 사용자 이벤트(사용자 이상/정보)

개요

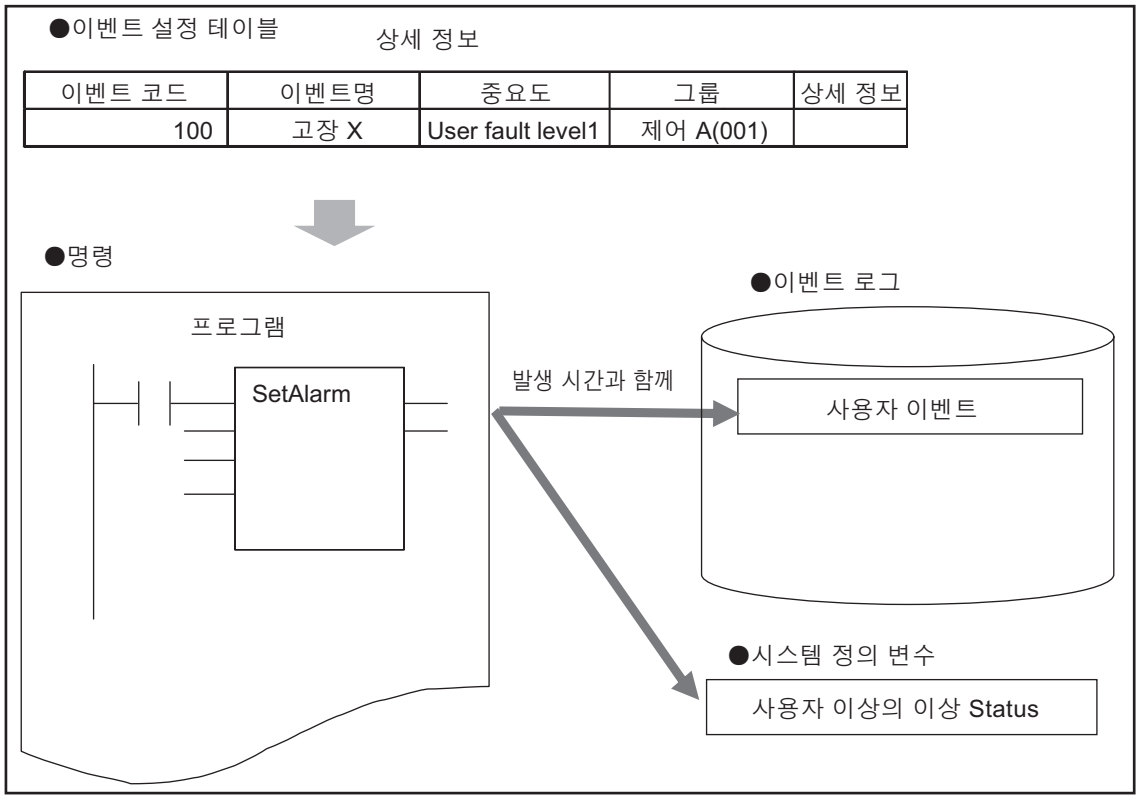
사용자가 정의한 이상/정보입니다. 명령을 사용하여 발생시킬 수 있습니다.

- **사용자 이상**

  사용자 정의의 이상입니다.
  사용자 이상 발생(SetAlarm) 명령을 사용하여 사용자 이상을 발생시킵니다. 이 때, 사용자 이상이 이벤트 로그에 기록됩니다.
  대응하는 시스템 정의 변수가 TRUE가 됩니다.
  사용자 이상은 동작 모드 변경에 의해 해제되지 않습니다.

- **사용자 정보**

  이상이 아닌 사용자 정의의 통지 정보입니다.
  사용자 정보 발생(SetInfo) 명령을 사용하여 사용자 정보를 발생시킵니다.
  이 때, 사용자 정보가 이벤트 로그에 기록됩니다.



사용 순서

다음 순서로 사용합니다.

- **사용자 이상**

1. 이벤트 설정 테이블에 '사용자 이상'을 등록
2. 사용자 이상 발생 명령 SetAlarm을 실행
   (이벤트 설정 테이블에서 정의한 이벤트 코드를 지정)
3. 사용자 이상 발생
4. 해당하는 시스템 정의 변수 '_AlarmFlag'(사용자 이상 Status)가 TRUE가 됨 그 조건에서 임의의 처리를 실행
5. Sysmac Studio, 명령 또는 표시기에서 사용자 이상을 확인

- **사용자 정보**

1. 이벤트 설정 테이블에 '사용자 정보'를 등록
2. 사용자 정보 발생 명령 SetInfo를 실행
3. 이벤트 로그에 대한 등록을 확인

이벤트 설정 테이블의 설정

'사용자 이상' 또는 '사용자 정보'를 발생시키려면 먼저 Sysmac Studio에서 '이벤트 설정 테이블'에 '사용자 이상' 또는 '사용자 정보'를 사전에 등록합니다.
여기에서 설정한 사용자 이벤트는 Sysmac Studio 또는 표시기에서 동일한 내용을 표시할 수 있습니다.
이벤트 설정 테이블에 등록할 수 있는 이벤트의 총 수는 최대 5120건입니다.

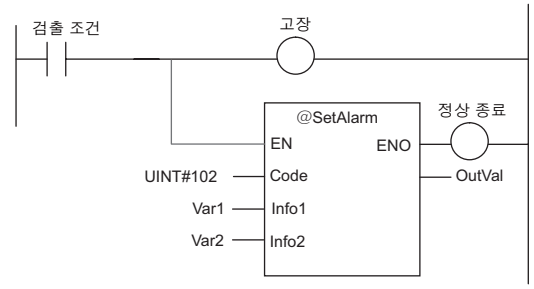
이벤트 설정 테이블 설정 화면

이벤트 설정 테이블

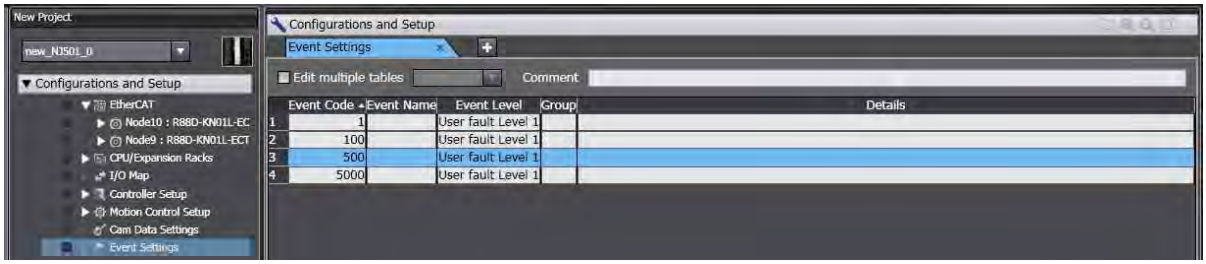
이벤트 코드	이벤트명	중요도	그룹	상세 정보
10001	故障X	User fault Level3	제어 A(001)	
:	:	:	:	:

상세 정보
■내용 고장 X가 발생했습니다. ■대처 안전을 확인하고 요인 코드에 따라 처리합니다.

프로그램 예



이벤트 설정 테이블에 다음을 설정합니다.



● **이벤트 설정 테이블의 내용**

항목	내용	값
이벤트 코드	이벤트를 식별하는 번호를 이벤트 중요도에 따라 지정할 수 있습니다.	사용자 이상: 1~40000 사용자 정보: 40001~60000
이벤트명	이벤트의 이름을 나타내는 제목을 기재할 수 있습니다.	최대 128문자
중요도	이벤트의 중요도를 지정합니다. 중요도를 숫자로 표시하고, 숫자가 작을수록 중요도가 높다고 간주됩니다.	사용자 이상: User fault Level1~8 사용자 정보: User Information
그룹	이벤트의 발생 부분 및 분류 등을 나타내는 그룹명을 지정할 수 있습니다. 각 이벤트를 사용자 정의로 그룹화할 수 있습니다.	최대 32문자 사용할 수 있는 문자 제한 없음 대문자/소문자의 구별 있음 예약어 없음
상세 정보	이벤트의 내용을 설명하는 메시지를 기재할 수 있습니다. 사용자가 자유롭게 임의의 문자열을 기입할 수 있습니다. Sysmac Studio/표시기에 이벤트를 표시할 때 사용합니다.	최대 1024문자 사용할 수 있는 문자 제한 없음 대문자/소문자의 구별 있음 예약어 없음
전체 정지 이상 레벨의 컨트롤러 이상 발생 시, 표시기에 표시하는 이상의 상세 정보	뒤에 설명하는 '참고: 전체 정지 이상 레벨의 컨트롤러 이상 발생 시 표시기에 사용자 메시지를 표시하는 기능:'을 참조하십시오.	최대 128문자 사용할 수 있는 문자 제한 없음 대문자/소문자 구별 있음 예약어 없음
코멘트	복수의 테이블별로 정보를 첨부할 수 있습니다.	

참고

동일한 이벤트 코드에 대해 복수의 지역 또는 복수의 사용자용으로 최대 9개 언어(9종류)를 설정할 수 있습니다. 이벤트 설정 테이블은 Sysmac Studio에서 Microsoft Excel 파일의 클립보드를 경유하여 가져오기를 할 수 있습니다.

참고

전체 정지 이상 레벨의 컨트롤러 이상 발생 시 표시기에 사용자 메시지를 표시하는 기능:
전체 정지 이상 레벨의 컨트롤러 이상 발생 시, 사용자 프로그램의 실행은 정지합니다.
NJ/NX 시리즈에서는 전체 정지 이상 레벨의 컨트롤러 이상 발생 시 표시기에 사용자용 메시지를 표시하는 기능이 있습니다. 그 메시지는 Sysmac Studio의 이벤트 설정 테이블의 사용자 이벤트 목록 하단에 설정합니다.

● **이벤트 중요도와 이벤트 코드**

이벤트 유형	중요도	이벤트 중요도의 분류명(*)	대응하는 이벤트 코드의 범위	설명
사용자 이상	높음	User fault Level1	1~5000	사용자는 8단계 중에서 선택
		User fault Level2	5001~10000	
		User fault Level3	10001~15000	
		User fault Level4	15001~20000	
		User fault Level5	20001~25000	
		User fault Level6	25001~30000	
		User fault Level7	30001~35000	
	낮음	User fault Level8	35001~40000	
사용자 정보	가장 낮음	User Information	40001~60000	이벤트 유형이 '사용자 정보'인 중요도

* 사용자 이상의 중요도는 컨트롤러 이상의 중요도와 다른 분류입니다.

사용상의 주의 사항

이벤트 설정 테이블을 갱신하고 CPU 유니트에 전송한 경우에 그 이전의 사용자 이벤트의 이벤트 로그는 오래된 내용으로 기록되어 있습니다. 이벤트 설정 테이블은 새로워졌기 때문에 불일치가 발생하는 경우가 있으므로 주의하십시오.

관련 명령

사용자 이상을 발생, 해제하는 명령과 발생한 사용자 이상을 확인하는 명령이 있습니다.

● 사용자 이상을 발생, 해제하는 명령

다음과 같은 명령으로 사용자 이상을 발생/해제, 사용자 정보를 발생시킵니다.
동시에 발생 중인 상태로 할 수 있는 사용자 이상의 수는 각 이벤트 중요도에 32건씩, 합계 최대 256건입니다.

명령 이름	명령	기능
사용자 이상 발생 명령	SetAlarm	사용자 이상을 발생시킵니다.
사용자 이상 해제 명령	ResetAlarm	사용자 이상을 해제합니다.
사용자 정보 발생 명령	SetInfo	임의의 사용자 정보를 이벤트 로그에 기록합니다.

● 사용자 이상을 확인하는 명령

사용자 이상 Status 취득(GetAlarm) 명령으로 사용자 이상의 발생 유무와 발생 중인 사용자 이상 중 가장 중요한 이벤트 중요도와 그 이벤트 코드를 취득할 수 있습니다.

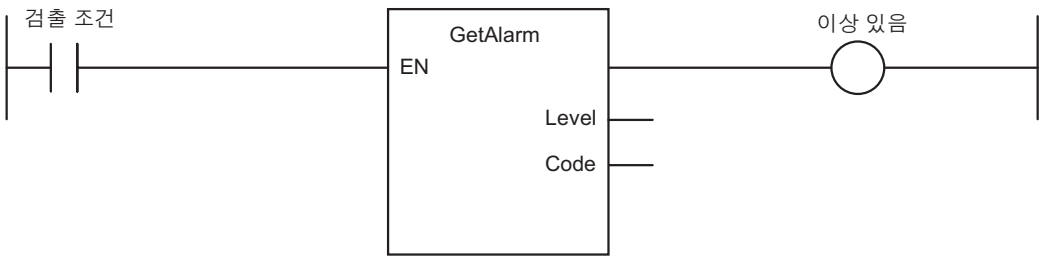
예)

이벤트 설정 테이블

이벤트 코드	이벤트명	중요도	그룹	상세 정보
10001	고장 X	User fault Level3	제어 A(001)	
:	:	:	:	:

상세 정보
■내용 고장 X가 발생했습니다. ■대처 안전을 확인하고 요인 코드에 따라 처리합니다.

실제 프로그램



참고

사용자 이상에 따라 발생한 컨트롤러 이상에 대해 대처 방법 등의 메시지를 부가할 수 있습니다. 발생 중인 컨트롤러 이상에 대해 이상 Status 및 이벤트 코드를 GetPLCError 등의 컨트롤러 이상의 Status 취득 명령으로 취득하고, 그 정보로 트리거에 사용자 이상을 발생시킵니다.

예 1

배터리 전압 저하 발생 시, 이벤트 코드(16#000B0000)를 취득하고, 다음과 같은 메시지를 표시합니다.

배터리 수명이 다됨 배터리의 교환 전에는 5분 이상 통전하고, 전원 OFF 후 5분 이내에 새 배터리로 교환하십시오.

예 2

부분 정지 이상 레벨의 컨트롤러 이상 발생 시, 이벤트 중요도(가장 중요한 Status: 2)를 취득하고, 다음과 같은 메시지를 표시합니다.

장치가 고장 났습니다. 다음 연락처로 문의하십시오. <서비스 센터 연락처> 접수 시간 AM 8:00~PM 9:00 TEL ***-****-****

사용자 이상 관련 시스템 정의 변수

변수명	이름	기능	데이터 형식	R/W
_AlarmFlag	사용자 이상 Status	사용자 이상 발생 중에는 이벤트 중요도에 해당하는 비트가 TRUE가 됩니다. User fault Level 1~8이 비트 00~07에 대응합니다.	WORD	R

이벤트 로그에 기록

사용자 이상 또는 사용자 정보 발생 시, ResetAlarm에 의해 해제했을 때 이벤트 로그가 기록됩니다. 그 때, 이벤트 로그의 사용자 이벤트 로그에는 발생 시각, 이벤트 코드, 이벤트 중요도, 부속 정보 1, 2가 기록됩니다.

사용자 이상 해제

사용자 이상은 NJ/NX 시리즈 컨트롤러의 전원을 OFF→ON으로 하면 해제됩니다.

Sysmac Studio, 명령(사용자 이상 해제 명령　ResetAlarm) 또는 표시기에서 이상을 해제할 수도 있습니다.

자세한 내용은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

8-8 이벤트 중요도 변경 기능

NJ/NX 시리즈 컨트롤러에서 발생하는 이상, 상태 변화, 사용자 정의 사건 등을 총칭하여 이벤트라고 합니다. 사용자는 Sysmac Studio의 표시 및 CPU 유니트의 전면 LED의 동작에 따라 어떤 이벤트가 발생했는지 알 수 있습니다.
이벤트에는 시스템에 의해 사전에 정의된 컨트롤러 이벤트와 사용자가 임의로 정의하는 사용자 이벤트가 있습니다. 그리고 각 컨트롤러 이벤트는 5단계의 이벤트 중요도로 분류됩니다. 이벤트 중요도의 자세한 내용은 '이벤트 중요도(8-70페이지)'를 참조하십시오.
이벤트 중요도 변경 기능은 각 컨트롤러 이벤트에 할당된 이벤트 중요도를 사용자가 임의로 변경할 수 있는 기능입니다.

버전 관련 정보

이벤트 중요도 변경 기능은 CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 사용할 수 있습니다.

8-8-1 이벤트 중요도 변경의 용도

발생한 컨트롤러 이벤트의 중요도에 따라 CPU 유니트의 전면 LED 등의 동작이 사전에 규정되어 있습니다. 이벤트 중요도 변경 기능을 사용하면 임의의 이벤트 발생에 대해 이와 같은 컨트롤러의 동작을 변경할 수 가능입니다.
예를 들어 ERROR LED는 이벤트 중요도가 가벼운 이상 레벨이라면 점멸하지만 감시 정보 레벨이라면 소등된 상태 그대로입니다. 이벤트 중요도 변경 기능을 사용하면 임의의 이벤트 발생에 대해 ERROR LED가 소등되도록 설정하거나 점멸하도록 설정할 수 있습니다.
컨트롤러의 이벤트 중요도별 동작에 대해서는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

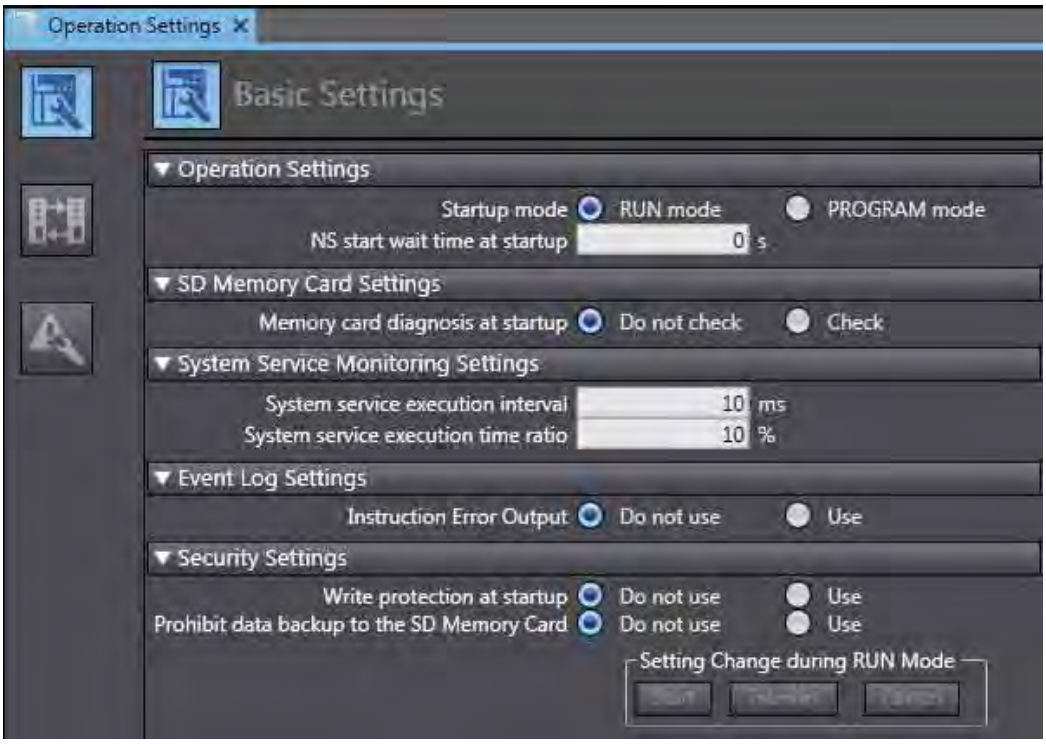
8-8-2 이벤트 중요도 변경 가능한 이벤트

이벤트의 중요도가 변경 가능한지 여부는 이벤트별로 정해져 있습니다.
컨트롤러 이벤트의 종류, 중요도, 중요도 변경 가능 여부의 자세한 내용은 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

8-8-3 이벤트 중요도 변경 조작 방법

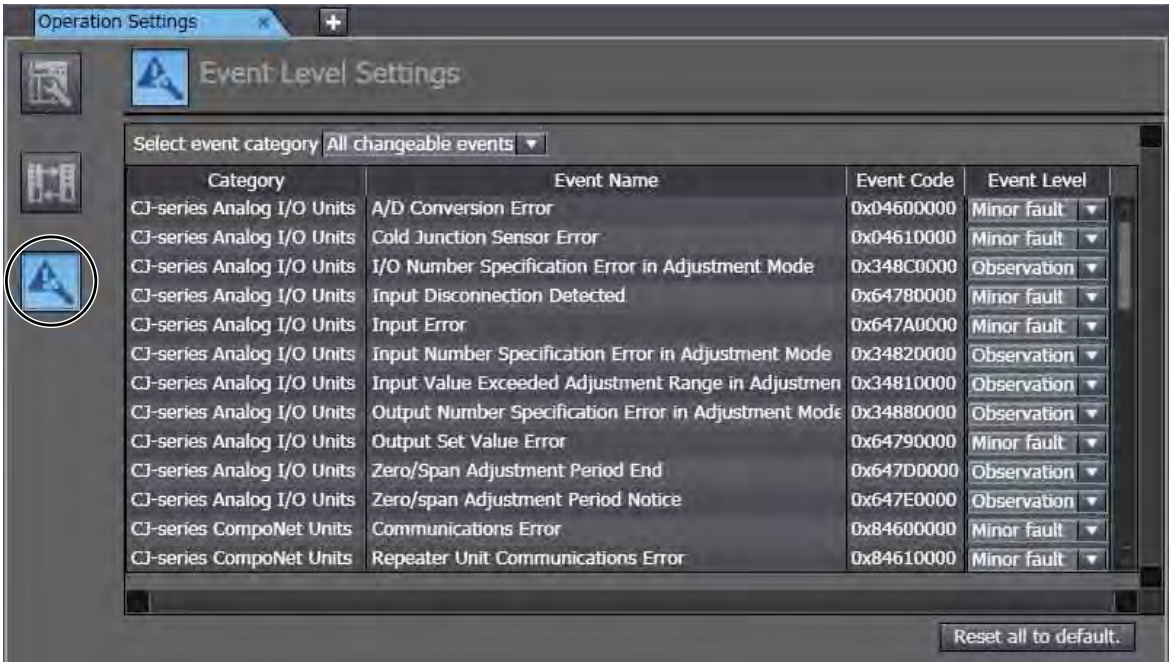
1 **Sysmac Studio의 [구성·설정]–[컨트롤러 설정]–[동작 설정]을 두 번 클릭 또는 마우스 오른쪽 버튼으로 클릭하고 [편집]을 선택합니다.**

편집 창에 동작 설정의 [전반 설정] 보기가 표시됩니다.



2 **[이상 중요도 설정] 버튼을 클릭합니다.**

중요도 변경이 가능한 이벤트의 목록이 표시됩니다.



3 **중요도를 변경하는 이벤트의 [중요도]를 변경대상의 중요도에 설정합니다.**

사용상의 주의 사항

Sysmac Studio에서 이벤트 중요도 변경 설정을 실행하고, 컨트롤러에 다운로드했을 때, 이미 그 이벤트가 발생했던 경우에 다운로드 시작 시 그 이벤트는 일단 해제됩니다. 그리고 다운로드 중에 다시 그 이벤트가 발생한 경우에 변경 전의 중요도에 따라 동작합니다. 다운로드 완료 후 그 이벤트가 발생한 경우에는 변경 후의 중요도에 따른 동작으로 전환됩니다.

9

백업 기능

이 장에서는 NJ/NX 시리즈 컨트롤러의 각종 설정 데이터의 백업 기능에 대해 설명합니다. 백업 기능에는 몇 가지 종류가 있고, 각각 백업 대상이 되는 데이터 및 저장 장소가 다릅니다. 먼저 전체적인 백업 기능에 대해 설명하고, 그 후, 종류별로 설명합니다.

9-1 전체적인 백업 기능 9-3
9-1-1 백업 기능의 용도 9-3
9-1-2 백업 기능 조작 순서의 예 9-4
9-1-3 백업 기능의 대상 데이터 9-6
9-1-4 백업 기능의 종류 9-8
9-1-5 백업 기능의 종류와 데이터 그룹의 관계 9-10
9-1-6 백업 기능의 대상 범위 9-11

9-2 SD 메모리 카드 백업 기능 9-13
9-2-1 백업(컨트롤러에서 SD 메모리 카드) 9-14
9-2-2 복원(SD 메모리 카드에서 컨트롤러) 9-19
9-2-3 비교(컨트롤러와 SD 메모리 카드 간) 9-26

9-3 SD 메모리 카드에 대한 백업 금지 기능 9-31

9-4 SD 메모리 카드 자동 전송 기능 9-32

9-5 SD 메모리 카드 프로그램 전송 기능 9-34

9-6 Sysmac Studio 컨트롤러 백업 기능 9-40
9-6-1 백업(컨트롤러에서 PC) 9-41
9-6-2 복원(PC에서 컨트롤러) 9-42
9-6-3 비교(컨트롤러와 PC 간) 9-44

9-7 Sysmac Studio 백업 파일 가져오기/내보내기 기능 9-45

9-8 Sysmac Studio 변수/메모리 백업 기능 9-46
9-8-1 Sysmac Studio 변수/메모리 백업 기능의 대상 데이터 9-46
9-8-2 Sysmac Studio 변수/메모리 백업 기능의 조작 방법 9-46
9-8-3 CPU 유니트 형식 간의 호환성 9-47

9-9 EtherCAT 슬레이브 연결 시의 백업 기능 9-49
9-9-1 EtherCAT 슬레이브의 백업 대상 데이터 9-49
9-9-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-50
9-9-3 EtherCAT 슬레이브의 복원 실행 조건 9-51
9-9-4 백업 기능의 대상 EtherCAT 슬레이브 9-52

9-10 EtherCAT 슬레이브 터미널 연결 시의 백업 기능 9-55
9-10-1 EtherCAT 슬레이브 터미널의 백업 대상 데이터 9-55
9-10-2 EtherCAT 슬레이브 터미널의 각 상태에서의 백업 기능 실행 가능 여부 9-56
9-10-3 EtherCAT 슬레이브 터미널의 복원 실행 조건 9-56
9-11 NX 유니트 연결 시의 백업 기능 9-57
9-11-1 CPU 유니트 상의 NX 유니트의 백업 대상 데이터 9-57
9-11-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-57
9-11-3 CPU 유니트 상의 NX 유니트의 복원 실행 조건 9-58
9-12 CJ 유니트 연결 시의 백업 기능 9-59
9-12-1 CJ 유니트의 백업 대상 데이터 9-59
9-12-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부 9-59
9-12-3 CJ 유니트의 복원 실행 조건 9-60
9-13 백업 관련 파일 9-61
9-13-1 백업 관련 파일의 종류 9-61
9-13-2 백업 파일의 사양 9-63
9-13-3 복원 명령 파일의 사양 9-64
9-13-4 자동 전송 명령 파일의 사양 9-66
9-13-5 컨트롤러 비교 결과 파일의 사양 9-69
9-13-6 EtherCAT 비교 결과 파일의 사양 9-70
9-13-7 EtherCAT 슬레이브 터미널 비교 결과 파일의 사양 9-71
9-13-8 NX 유니트 비교 결과 파일의 사양 9-72
9-13-9 CJ 유니트 비교 결과 파일의 사양 9-73
9-14 백업 기능의 파일 간 호환성 9-75
9-14-1 백업 기능 간 호환성 9-75
9-14-2 CPU 유니트의 형식 간의 호환성 9-76
9-14-3 CPU 유니트의 유니트 버전 호환성 9-77
9-15 백업 기능과 동시에 실행할 수 없는 기능 9-78

9-1 전체적인 백업 기능

NJ/NX 시리즈의 백업 기능은 다음과 같은 3가지를 말합니다.

기능	내용
백업	컨트롤러 내의 각종 설정 데이터를 묶어서 SD 메모리 카드나 PC에 저장하는 기능입니다. 저장하는 파일을 백업 파일이라고 합니다.
복원	SD 메모리 카드 및 PC 내의 백업 파일을 컨트롤러에 전송하는 기능입니다. 컨트롤러 내의 각종 설정 데이터가 백업 실행 시의 데이터로 치환됩니다.
비교	SD 메모리 카드 및 PC 내의 백업 파일의 내용과 현재 컨트롤러 내의 각종 설정 데이터가 일치하는지 여부를 판정하는 기능입니다.

여기에서는 전체적인 백업 기능으로 다음 내용을 설명합니다.

항목	내용
백업 기능의 용도	백업 기능의 유효한 사용 방법을 설명합니다.
백업 기능 조작 순서의 예	백업 기능은 간단한 조작으로 실행할 수 있습니다. 예시는 다음과 같습니다.
백업 기능의 대상 데이터	백업 기능을 사용하면 연결된 유니트 및 슬레이브 내의 어느 데이터를 저장할 수 있는지를 설명합니다.
백업 기능의 종류	백업 기능에는 데이터의 저장 장소 등의 차이에 따라 몇 가지 종류가 있습니다. 어떤 종류가 있는지, 이들의 차이에 대해 설명합니다.
백업 기능의 종류와 데이터 그룹의 관계	백업 기능의 종류에 따라 대상이 되는 데이터 그룹이 다릅니다. 백업 기능의 종류와 데이터 그룹의 관계를 설명합니다.
백업 기능의 대상 범위	백업 기능을 사용하면 연결된 어느 유니트 및 슬레이브의 설정 데이터를 저장할 수 있는지를 설명합니다.

안전상의 중요 사항

- 하드웨어 개정이 다르면 성능이 다를 수 있습니다. 하드웨어 개정 번호가 다른 CPU 유니트에 사용자 프로그램 및 각종 데이터, 설정값을 전송할 때에는 동작을 충분히 확인한 후에 본 운전을 실시하십시오.
- NX 시리즈 CPU 유니트인 경우 변수의 현재값 백업은 유지 변수가 갱신되지 않은 상태에서 실시하는 것을 권장합니다.
  유지 변수의 값이 갱신되는 상태에서 다음과 같은 변수를 백업하면 정확하게 저장되지 않을 수 있습니다.
  - 데이터 크기가 16비트 이상인 구조체의 멤버
  - 데이터 크기가 16비트 이상의 배열 요소

9-1-1 백업 기능의 용도

백업 기능은 다음과 같은 용도로 사용합니다.

항목	용도
프로그램, 설정 변경	가동 중인 장치의 사용자 프로그램 및 설정을 변경하는 경우
하드웨어 교환	CPU 유니트, 유니트, 슬레이브의 하드웨어를 교환하는 경우
장치 이상 시의 원인 조사	장치에 이상이 발생했을 때 원인을 해석하기 위해 컨트롤러의 데이터를 저장하는 경우
장치의 백업과 원상회복	장치에 이상이 발생했을 때, 장치를 정상한 상태 데이터로 복원하는 경우 또한, 사전에 가동 중인 장치의 데이터를 백업하는 경우
장치의 제조	동일한 장치를 제조하기 위해 복제원의 장 데이터를 구입 후 초기 상태인 장치로 전송하는 경우

9-1-2 백업 기능 조작 순서의 예

백업 기능을 사용하면 간단한 조작으로 컨트롤러의 백업, 복원, 비교를 실행할 수 있습니다.
다음과 같이 CPU 유니트의 전면 스위치를 사용하여SD 메모리 카드에 백업, 복원, 비교를 실행하는 경우의 조작 순서를 나타냅니다.

사용상의 주의 사항

NX701 CPU 유니트의 경우에 딥 스위치는 1에서 8까지 8개 있습니다. 백업 기능을 사용할 때는 5에서 8을 모두 OFF로 하십시오.

백업 조작 순서

- **백업 준비**

1 **CPU 유니트에 SD 메모리 카드를 장착합니다.**

2 **CPU 유니트의 딥 스위치 1에서 4를 1: OFF, 2: OFF, 3: ON, 4: OFF로 설정합니다.**

- **백업 실행**

3 **SD 메모리 카드 전원 공급 정지 버튼을 3초간 계속 누릅니다.**
백업이 시작됩니다. SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다.
백업이 완료되면 SD PWR가 점등으로 바뀝니다.

- **백업 조작 종료**

4 **CPU 유니트의 딥 스위치 1에서 4를 모두 OFF로 합니다.**

5 **SD 메모리 카드 전원 공급 정지 버튼을 누르고, SD PWR의 소등을 확인합니다.**

6 **SD 메모리 카드를 꺼냅니다.**

복원 조작 순서

- **복원 준비**

1 **CPU 유니트 및 EtherCAT 슬레이브의 전원을 OFF합니다.**

2 **CPU 유니트에 백업 파일이 저장된 SD 메모리 카드를 장착합니다.**

3 **CPU 유니트의 딥 스위치 1에서 4를 1: OFF, 2: OFF, 3: ON, 4: ON으로 설정합니다.**

- **복원 실행**

4 **CPU 유니트 및 EtherCAT 슬레이브의 전원을 ON합니다.**
복원이 시작됩니다. SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다.
복원이 완료되면 SD PWR가 점등으로 바뀝니다.

● **복원 조작 종료**

5 **SD 메모리 카드 전원 공급 정지 버튼을 누르고, SD PWR의 소등을 확인합니다.**

6 **CPU 유니트 및 EtherCAT 슬레이브의 전원을 OFF합니다.**

● **통상 운전 시작**

7 **SD 메모리 카드를 꺼냅니다.**

8 **CPU 유니트의 딥 스위치 1에서 4를 모두 OFF로 합니다.**

9 **CPU 유니트 및 EtherCAT 슬레이브의 전원을 ON합니다.**

사용상의 주의 사항

EtherCAT 슬레이브를 연결한 경우의 복원

- EtherCAT 슬레이브를 연결하여 복원을 실행한 경우에는 복원 완료 후 반드시 CPU 유니트 및 EtherCAT 슬레이브의 전원을 재투입하십시오. 전원을 재투입하지 않고 운전을 시작하면 컨트롤러가 의도하지 않는 동작을 할 가능성이 있습니다.
- EtherCAT 슬레이브를 연결하여 복원을 실행하고, 그 후 비교하는 경우에는 일단 CPU 유니트 및 EtherCAT 슬레이브의 전원을 OFF하고, 안전 모드에서 기동시킨 후 비교하십시오. 통상적인 전원 재투입을 실행하면 비교 실행 전에 컨트롤러가 운전을 시작합니다. 따라서 비교 불일치 그대로 운전이 시작될 가능성이 있습니다. 안전 모드의 자세한 내용은 "NJ/NX 시리즈 트러블슈팅 설명서(SBCA-361)"를 참조하십시오.

비교 조작 순서

● **비교 준비**

1 **CPU 유니트에 백업 파일이 저장된 SD 메모리 카드를 장착합니다.**

2 **CPU 유니트의 딥 스위치 1에서 4를 모두 OFF로 합니다.**

● **비교 실행**

3 **SD 메모리 카드 전원 공급 정지 버튼을 3초간 계속 누릅니다.**
비교가 시작됩니다. SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다.
비교 일치로 비교가 완료되면 SD PWR가 점등으로 바뀝니다.
비교 불일치로 비교가 완료되면 SD PWR가 0.5s 점등, 0.5s 소등에서 점멸로 바뀝니다.

● **비교 조작 종료**

4 **SD 메모리 카드 전원 공급 정지 버튼을 누르고, SD PWR의 소등을 확인합니다.**

5 **SD 메모리 카드를 꺼냅니다.**

9-1-3 백업 기능의 대상 데이터

백업 기능의 대상 데이터는 다음과 같습니다.
또한, 백업 기능의 세부 설명에서는 대상 데이터를 나타낼 때 '데이터 그룹'을 사용합니다.

데이터 그룹	데이터 항목
사용자 프로그램·설정	EtherCAT 구성(EtherCAT 슬레이브 구성, EtherCAT 마스터 설정) 유니트 구성·유니트 설정*1 I/O 맵 컨트롤러 설정(동작 설정, 내장 EtherNet/IP 포트 설정, 내장 I/O 설정, 옵션 보드 구성·설정)*2 모션 제어 설정 캠 데이터 설정 이벤트 설정 태스크 설정 데이터 트레이스 설정 태그 데이터 링크 테이블 컨트롤러 이름 조작 권한 인증 사용자 프로그램 실행용 ID(사용자 프로그램 내) 내장 시계(표준 시간대 설정) POU 데이터(데이터 형식, 글로벌 변수) CJ 유니트용 메모리 설정*3
내장 EtherNet/IP 포트의 IP 주소*4	내장 EtherNet/IP 포트 설정의 TCP/IP 설정 중 설정 유형, IP 주소, 서브넷 마스크, 기본 게이트웨이
변수의 현재값	유지 속성이 있는 변수의 값*5
CJ 유니트용 메모리의 현재값	CJ 유니트용 메모리 안의 HR, DM, EM 영역의 값*3
유니트·슬레이브 설정	EtherCAT 슬레이브의 백업 파라메터*6 CJ 유니트 내의 파라메터*3*7 NX 유니트 설정
앱솔루트 엔코더 원점 위치 오프셋	앱솔루트 엔코더가 부착된 서보 드라이버의 현재 위치를 복원하기 위한 설정값

*1. NX102 및 NX1P2 CPU 유니트 상의 NX 유니트에서는 구성 정보, 유니트 동작 설정 및 유니트 어플리케이션 데이터가 대상이 됩니다.
*2. 내장 I/O 설정, 옵션 보드 설정이 대상이 되는 것은 NX1P2 CPU 유니트뿐입니다.
*3. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다. CJ 유니트 내의 파라메터를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*4. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합으로 내장 EtherNet/IP 포트 설정의 IP 주소는 하나의 데이터 그룹으로 사용할 수 있습니다.
상기 이외의 조합에서는 IP 주소는 사용자 프로그램·설정에 포함됩니다.
*5. 유지 속성이 있는 시스템 정의 변수 중 백업 기능의 대상에서 제외되는 시스템 정의 변수가 있습니다. 자세한 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'에서 개별 사양을 확인하십시오.
*6. EtherCAT 슬레이브 중에는 설정 파라메터의 일부 또는 전부가 백업 대상이 아닌 모델이 있습니다. 백업 기능의 대상EtherCAT 슬레이브의 자세한 내용은 '9-9-4 백업 기능의 대상 EtherCAT 슬레이브(9-52페이지)'를 참조하십시오.
*7. 백업 기능의 대상이 되는 구체적인 데이터는 "CJ 시리즈 CJ2 CPU 유니트 사용 설명서 하드웨어편(SBCA-349)"을 참조하십시오.

안전상의 중요 사항

앱솔루트 엔코더 원점 위치 오프셋에 대한 주의 사항

CPU 유니트는 앱솔루트 엔코더의 정보인 '앱솔루트 엔코더 원점 위치 오프셋'을 배터리에 의해 유지합니다. 다음 중 하나의 조건을 충족하는 경우에는 앱솔루트 엔코더 원점 위치 오프셋을 복원 대상에서 제외하고 복원을 실시하십시오. 그 후, 새로 앱솔루트 엔코더 원점 설정을 실시하십시오. 앱솔루트 엔코더 원점 설정을 하지 않은 경우에 제어 대상이 예기치 못한 동작을 할 위험이 있습니다.

- 서보 모터와 드라이버의 개체가 백업을 작성했을 때의 개체와 다름
- 백업을 작성한 이후에 앱솔루트 엔코더 설정을 실행
- 앱솔루트 엔코더의 절대값 데이터가 지워짐

9-1-4 백업 기능의 종류

NJ/NX 시리즈에서 사용할 수 있는 백업 기능에는 데이터의 저장 장소가 SD 메모리 카드인 경우와 PC인 경우가 있습니다. 또한, 조작 방법으로 CPU 유니트 전면 스위치에 의한 것, 시스템 정의 변수에 의한 것, Sysmac Studio 화면에 의한 것 3가지가 있습니다.

SD 메모리 카드에 데이터를 저장하는 기능

SD 메모리 카드에 데이터를 백업, 복원, 비교하는 것이 SD 메모리 카드 백업 기능입니다. 관련 기능으로 SD 메모리 카드 백업 금지 기능, SD 메모리 카드 자동 전송 기능 및 SD 메모리 카드 프로그램 전송 기능이 있습니다.

기능명		내용	조작 방법			참조
			CPU 유니트 전면 스위치	시스템 정의 변수	Sysmac Studio 화면	
SD 메모리 카드 백업 기능	백업	컨트롤러의 데이터를 SD 메모리 카드에 백업 파일로 저장합니다.	○	○	○	📖'9-2-1 백업(컨트롤러에서 SD 메모리 카드)(9-14 페이지)'
	복원	SD 메모리 카드 백업 파일의 데이터를 컨트롤러에 전송합니다.	○	○		📖'9-2-2 복원(SD 메모리 카드에서 컨트롤러)(9-19 페이지)'
	비교	컨트롤러의 데이터와 SD 메모리 카드의 백업 파일 데이터를 비교합니다.	○	○	○	📖'9-2-3 비교(컨트롤러와 SD 메모리 카드 간)(9-26 페이지)'
SD 메모리 카드 백업 금지 기능		SD 메모리 카드에 대한 백업을 금지합니다.			○	📖'9-3 SD 메모리 카드에 대한 백업 금지 기능(9-31페이지)'
SD 메모리 카드 자동 전송 기능		전원 투입 시에 SD 메모리 카드의 백업 파일 데이터를 자동으로 컨트롤러에 읽어옵니다. 읽은 후의 CPU 유니트의 동작 모드는 [전원 투입 시 모드]의 설정에 따릅니다.	○			📖'9-4 SD 메모리 카드 자동 전송 기능(9-32페이지)'
SD 메모리 카드 프로그램 전송 기능		시스템 정의 변수에 의해 SD 메모리 카드에 저장된 프로그램을 컨트롤러에 전송합니다. 전송 후의 CPU 유니트의 동작 모드는 [전원 투입 시 모드]의 설정에 따릅니다.		○		📖'9-5 SD 메모리 카드 프로그램 전송 기능(9-34 페이지)'

PC에 데이터를 저장하는 기능

컨트롤러의 데이터를 PC에 백업, 복원, 비교하는 것이 Sysmac Studio 컨트롤러 백업 기능입니다.
컨트롤러를 사용하지 않고 Sysmac Studio의 프로젝트에서 각 데이터를 PC에 저장하고, 읽어오는 것이 Sysmac Studio 백업 파일 가져오기/내보내기 기능입니다.
배터리로 저장되는 현재값만 PC에 백업, 복원하는 것이 Sysmac Studio 변수/메모리 백업 기능입니다.

기능명		내용	조작 방법			참조
			CPU 유니트 전면 스위치	시스템 정의 변수	Sysmac Studio 화면	
Sysmac Studio 컨트롤러 백업 기능	백업	컨트롤러의 데이터를 PC에 백업 파일로 저장합니다.			○	'9-6-1 백업(컨트롤러에서 PC)(9-41페이지)'
	복원	PC 내의 백업 파일의 데이터를 컨트롤러에 전송합니다.			○	'9-6-2 복원(PC에서 컨트롤러)(9-42페이지)'
	비교	컨트롤러의 데이터와 PC 백업 파일의 데이터를 비교합니다.			○	'9-6-3 비교(컨트롤러와 PC 간)(9-44페이지)'
Sysmac Studio 백업 파일 가져오기/내보내기 기능	내보내기	컨트롤러를 사용하지 않고 Sysmac Studio의 프로젝트에서 백업 파일을 작성합니다.			○	'9-7 Sysmac Studio 백업 파일 가져오기/내보내기 기능(9-45페이지)'
	가져오기	컨트롤러를 사용하지 않고 백업 파일을 Sysmac Studio의 프로젝트에 읽습니다.			○	
Sysmac Studio 변수/메모리 백업 기능	백업	배터리로 유지되는 데이터의 현재값을 PC에 파일(XML 형식)로 백업(읽기)합니다.			○	'9-8 Sysmac Studio 변수/메모리 백업 기능(9-46페이지)'
	복원	배터리로 유지되는 데이터의 현재값을 PC에서 CPU 유니트로 복원(쓰기)합니다.			○	

버전 관련 정보

- SD 메모리 카드 백업 기능, SD 메모리 카드 자동 전송 기능, Sysmac Studio 컨트롤러 백업 기능, Sysmac Studio 백업 파일 가져오기/내보내기 기능은 CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 사용할 수 있습니다.
- SD 메모리 카드 프로그램 전송 기능은 CPU 유니트 Ver.1.11 이상과 Sysmac Studio Ver.1.15 이상의 조합에서 사용할 수 있습니다.
- 시스템 정의 변수에 의한 SD 메모리 카드 백업 기능의 복원은 CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합에서 사용할 수 있습니다.

참고

백업 기능은 시스템 서비스로 실행합니다. 따라서 운전 모드에서 백업 및 비교를 실행한 경우에 NJ 시리즈 CPU 유니트에서는 실행 완료까지 시간이 걸리는 경우가 있습니다. 운전 모드에서 백업 및 비교를 실행하는 경우에는 시스템 서비스 실행 시간을 충분히 확보하십시오. 프로그램 모드에서 시스템 서비스를 실행하면 처리 시간을 단축할 수 있습니다.

9-1-5 백업 기능의 종류와 데이터 그룹의 관계

백업 기능의 종류에 따라 대상이 되는 데이터 그룹이 다릅니다. 백업 기능의 종류와 대상 데이터 그룹의 관계는 다음과 같습니다.

(○: 대상 ×: 대상 외)

백업 기능의 종류		데이터 그룹					
		사용자 프로그램·설정	내장 EtherNet/IP 포트의 IP 주소*2	변수의 현재값	CJ 유니트용 메모리의 현재값*1	유니트·슬레이브 설정	앱솔루트 엔코더 원점 위치 오프셋
SD 메모리 카드 백업 기능	백업	○	○	○*3	○*4	○	○
	복원	○	○	○*3	○*4	○	○
	비교	○*5	○	×	×	○	×
SD 메모리 카드 자동 전송 기능*6		○	○	○*3	○*4	×	×
SD 메모리 카드 프로그램 전송 기능*7		○	○	○*3	○*4	×	×
Sysmac Studio 컨트롤러 백업 기능	백업	○	○	○*3	○*4	○*8	○
	복원	○	○	○*3	○*4	○*8	○
	비교	○*5	○	×	×	○*8	×
Sysmac Studio 백업 파일 가져오기/내보내기 기능	백업 파일 내보내기	○*9	○	×	×	×	×
	백업 파일 가져오기	○*9	○	×	×	○	×
Sysmac Studio 변수/메모리 백업 기능	백업 복원	×	×	○*3	○*4	×	○

*1. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.
*2. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합으로 내장 EtherNet/IP 포트 설정의 IP 주소는 하나의 데이터 그룹으로 사용할 수 있습니다.
상기 이외의 조합에서는 IP 주소는 사용자 프로그램·설정에 포함됩니다.
*3. 백업의 대상 데이터가 유지 속성의 변수의 현재값인 경우에만 처리합니다.
*4. 백업의 대상 데이터가 유지 속성의 CJ 유니트용 메모리의 현재값인 경우에만 처리합니다.
*5. 사용자 프로그램·설정 데이터 그룹 중 데이터 트레이스 설정은 비교하지 않습니다.
*6. 대상 데이터 그룹 중 사용자 프로그램·설정 이외의 데이터는 자동 전송 명령 파일에서 '전송'으로 지정한 것만 전송합니다.
*7. 대상 데이터 그룹 중 시스템 정의 변수를 전송 대상으로 한 것만 전송합니다.
*8. 백업 대상이 CJ 유니트인 경우에는 CJ 유니트 내의 파라메터를 대상으로 합니다. EtherCAT 슬레이브인 경우에는 EtherCAT 슬레이브의 백업 파라메터를 대상으로 합니다.

*9. 다음과 같은 데이터는 처리 대상에서 제외됩니다. Sysmac Studio의 버전에 따라 대상 외의 데이터가 다릅니다.
Sysmac Studio의 버전이 Ver.1.16 이상인 경우
컨트롤러 이름 중에 내장 EtherNet/IP 포트명
유니트 구성·유니트 설정 중의 CPU 고기능 유니트 영역
조작 권한 인증
데이터 트레이스 설정
Sysmac Studio 버전이 Ver.1.15 이하인 경우
컨트롤러 이름 중에 내장 EtherNet/IP 포트명
컨트롤러 설정 중의 내장 EtherNet/IP용 태그 데이터 링크 설정
유니트 구성·유니트 설정 중의 CPU 고기능 유니트 영역
조작 권한 인증
데이터 트레이스 설정

참고

Sysmac Studio 변수/메모리 백업 기능에서 취급하는 파일은 기타 백업 기능에서 취급하는 파일과 달라서 호환성이 없습니다.

Sysmac Studio 변수/메모리 백업 기능의 자세한 내용은 '9-8 Sysmac Studio 변수/메모리 백업 기능(9-46페이지)'을 참조하십시오.

9-1-6 백업 기능의 대상 범위

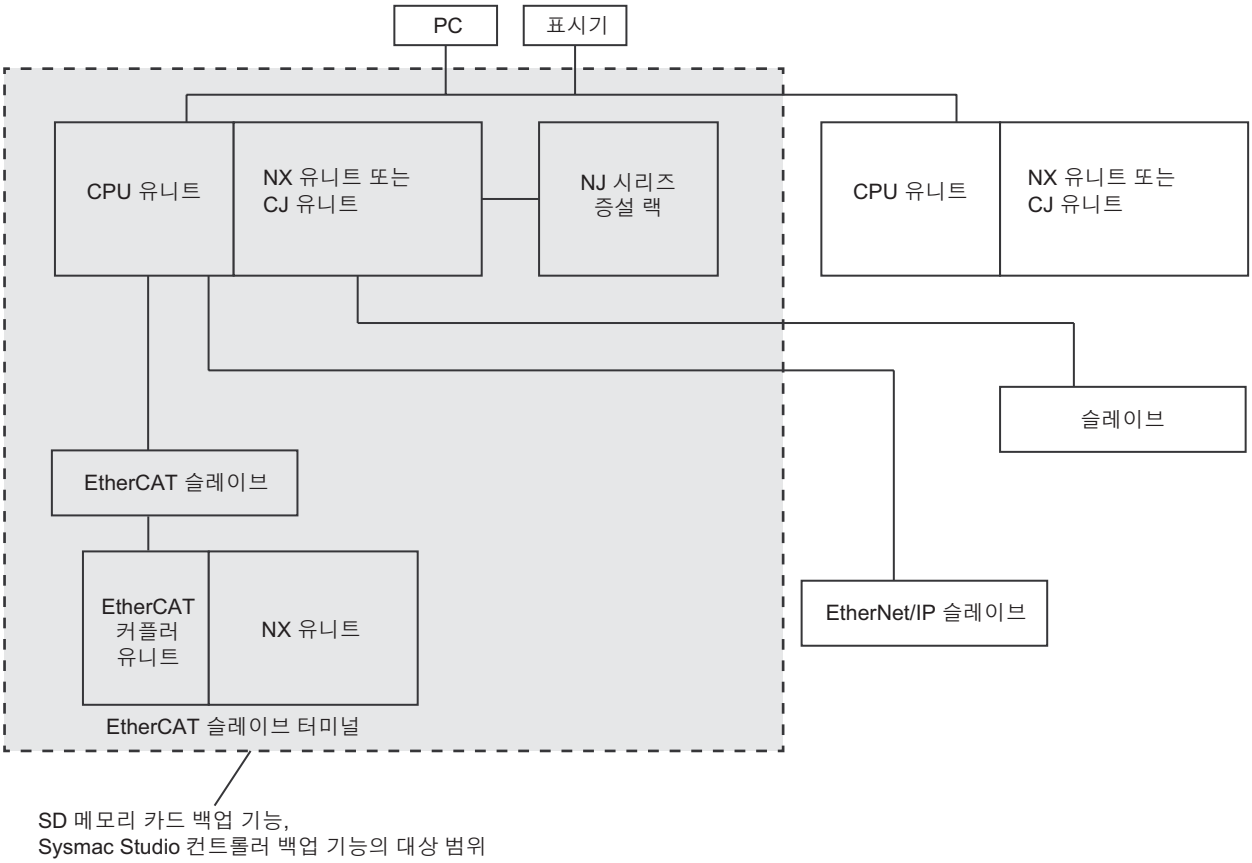
백업 기능에 따라 대상이 되는 슬레이브 및 유니트가 다릅니다. 각 백업 기능의 대상 슬레이브, 유니트는 다음과 같습니다.

(○: 대상 ×: 대상 외)

백업 기능의 종류	슬레이브, 유니트						
	NJ/NX 시리즈 CPU 유니트	EtherCAT 슬레이브*1	CPU 유니트 상의 NX 유니트*2	CJ 유니트*3		EtherNet/IP 슬레이브	PC, 표시기
				유니트, 마스터 유니트	슬레이브		
SD 메모리 카드 백업 기능	○	○*4	○	○	×	×	×
SD 메모리 카드 자동 전송 기능	○	×	×	×	×	×	×
SD 메모리 카드 프로그램 전송 기능	○	×	×	×	×	×	×
Sysmac Studio 컨트롤러 백업 기능	○	○*4	○	○	×	×	×
Sysmac Studio 백업 파일 가져오기/내보내기 기능	○	○*5	○	×	×	×	×
Sysmac Studio 변수/메모리 백업 기능	○	×	×	×	×	×	×

*1. EtherCAT 슬레이브 터미널을 포함합니다. 백업 기능의 대상이 EtherCAT 슬레이브 터미널인 경우에 EtherCAT 커플러 유니트도 각 NX 유니트도 백업 기능의 대상입니다.
*2. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX1P2 CPU 유니트뿐입니다.
*3. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.
*4. 세이프티 컨트롤 유니트는 대상에서 제외됩니다. 세이프티 컨트롤 유니트의 설정 데이터 가져오기/내보내기 기능은 "NX 시리즈 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)"를 참조하십시오.
*5. 가져오기만 가능합니다. 내보내기 기능은 대상에서 제외됩니다.

SD 메모리 카드 백업 기능 및 Sysmac Studio 컨트롤러 백업 기능의 대상 슬레이브, 유니트는 다음 그림의 범위입니다.



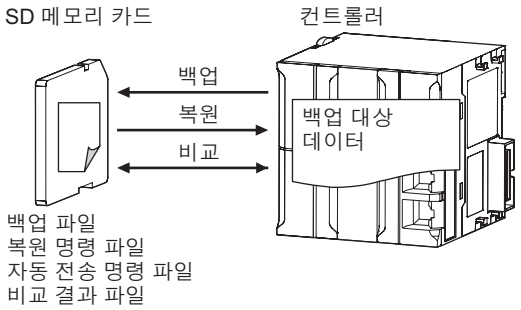
(주의) CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
(주의) CJ 유니트 및 NJ 시리즈 증설 랙을 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

버전 관련 정보

EtherCAT 슬레이브 터미널의 연결은 CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합에서 가능합니다.

9-2 SD 메모리 카드 백업 기능

SD 메모리 카드를 사용하여 컨트롤러의 데이터의 백업, 복원, 비교를 실행하는 기능입니다.



백업을 실행하면 SD 메모리 카드에 '백업 파일', '복원 명령 파일', '자동 전송 명령 파일'이 지정 디렉토리에 작성됩니다. 또한, 비교를 실행하면 '비교 결과 파일'이 지정 디렉토리에 작성됩니다.
이를 총칭하여 '백업 관련 파일'이라고 합니다.
백업 관련 파일의 역할은 다음과 같습니다.

파일	역할			
	내용	백업	복원	비교
백업 파일	백업 관련 기능으로 취급하는 컨트롤러의 데이터를 파일화한 것입니다.	작성	참조	참조
복원 명령 파일	복원을 실행할 때, 복원 대상의 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 복원 대상을 지정합니다.	작성	참조	참조
자동 전송 명령 파일	SD 메모리 카드 자동 전송을 실행할 때, 전송하는 대상의 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다.	작성	아무것도 하지 않음	아무것도 하지 않음
비교 결과 파일	비교를 실행한 후에 비교 결과를 저장하는 파일입니다.	아무것도 하지 않음	아무것도 하지 않음	작성

이 기능의 조작 방법과 대상 디렉토리, 실행 가능한 동작 모드는 다음과 같습니다.

조작 방법	디렉토리*1	실행 가능한 동작 모드		
		백업	복원	비교
CPU 유니트 전면 스위치*2	루트 디렉토리	운전 모드, 프로그램 모드	전원 투입 시	운전 모드, 프로그램 모드
시스템 정의 변수*3*4	시스템 정의 변수로 지정한 디렉토리	운전 모드, 프로그램 모드	운전 모드, 프로그램 모드*5	운전 모드, 프로그램 모드
Sysmac Studio SD 메모리 카드 화면	화면에서 지정한 디렉토리	운전 모드, 프로그램 모드	실행 불가능	운전 모드, 프로그램 모드

*1. SD 메모리 카드 내에만 지정할 수 있습니다.
*2. 복원 또는 비교할 때는 사전에 백업 파일과 복원 명령 파일을 루트 디렉토리에 저장해두십시오.
*3. 표시기에서 지시할 때 사용합니다. 복원 관련 시스템 정의 변수만은 사용자 프로그램에서도 액세스할 수 있습니다.
*4. 백업 중, 복원 중 또는 비교 중에 다른 표시기에서 백업 및 비교의 지시가 나오지 않도록 사용자측에서 관리하십시오. 그렇지 않은 경우에는 의도한 동작을 하지 않는 경우가 있습니다.
*5. Ver.1.13 이하의 CPU 유니트는 시스템 정의 변수에 의한 복원에 대응하지 않았으므로 실행할 수 없습니다.

9-2-1 백업(컨트롤러에서 SD 메모리 카드)

컨트롤러의 데이터를 CPU 유니트에 장착한 SD 메모리 카드에 저장합니다.

처리 내용

- 백업의 대상 데이터는 모든 데이터 그룹입니다.
- 백업을 실행하면 SD 메모리 카드에 '백업 파일', '복원 명령 파일', '자동 전송 명령 파일'이 지정 디렉토리에 작성됩니다.
- 지정 디렉토리에 기존 백업 관련 파일이 있는 경우에는 덮어씁니다.
- SD 메모리 카드에 백업 관련 파일을 쓰는 도중에 에러 발생한 경우에는 기존의 백업 관련 파일이 삭제됩니다. 그리고 새 백업 관련 파일도 작성되지 않습니다.
- 새 백업 관련 파일 작성 전에 에러가 발생한 경우에는 기존의 파일이 유지되고, 새 파일은 작성되지 않습니다.
- 백업 실행 중에 SD 메모리 카드 전원 공급 정지 버튼을 눌러도 전원 공급을 계속합니다.
- 백업 완료 후, SD 메모리 카드는 마운트 상태를 지속합니다.

조작 방법

● CPU 유니트 전면 스위치를 사용한 방법

처리 단계	조작 방법
시작 지시	딥 스위치 1에서 4[*1]를 1: OFF, 2: OFF, 3: ON, 4: OFF로 하고, SD 메모리 카드 전원 공급 정지 버튼을 3초간 계속 누르면 실행이 시작됩니다.
실행 중	실행 시작 직후[*2] SD PWR가 점등→0.5s 소등→점등의 패턴으로 1회 점멸합니다. 백업 실행 중 SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	정상 종료인 경우 SD PWR가 점등됩니다. 이상 종료인 경우 SD PWR가 0.5s 점등, 0.5s 소등으로 점멸됩니다. SD 메모리 카드 전원 공급 정지 버튼을 누르면 점등됩니다. [*1]

*1. NX701 CPU 유니트인 경우에 딥 스위치의 5에서 8을 모두 OFF로 하십시오.
*2. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR가 소등된 상태 그대로가 됩니다.

● 시스템 정의 변수 _Card1BkupCmd(SD 메모리 카드 백업 지시)를 사용하는 방법

처리 단계	조작 방법
시작 지시	시스템 정의 변수 _Card1BkupCmd.DirName(디렉토리명)에 저장 장소의 디렉토리명을 저장합니다. (예) 디렉토리 dirA 아래의 서브 디렉토리 dirB를 지정: "dirA/dirB" 시스템 정의 변수 _Card1BkupCmd.ExecBkup(백업 기동 플래그)를 FALSE→TRUE로 하면 실행이 시작됩니다.
취소 지시	백업 실행의 지시를 취소합니다. 시스템 정의 변수 _Card1BkupCmd.CancelBkup(백업 취소 플래그)를 TRUE로 하면 백업 처리가 이상 종료합니다.
실행 중	시스템 정의 변수 _Card1BkupSta.Active(실행 중 플래그)가 TRUE가 됩니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	정상 종료인 경우 시스템 정의 변수 _Card1BkupSta.Done(완료 플래그)이 TRUE가 됩니다. 이상 종료인 경우 시스템 정의 변수 _Card1BkupSta.Err(에러 플래그)가 TRUE가 됩니다.

(주의) 사용자 프로그램에서는 이 시스템 정의 변수에 액세스할 수 없습니다.

● Sysmac Studio의 SD 메모리 카드 화면에 의한 방법

처리 단계	조작 방법
시작 지시	Sysmac Studio SD 메모리 카드 화면의 [SD 메모리 카드 백업] 버튼을 누르고, 저장 장소의 디렉토리명을 지정하여 백업 실행을 지시합니다.
실행 중	백업 진행 상황을 나타내는 화면이 표시됩니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	완료 메시지가 표시되고, 계속해서 백업 데이터 비교 실시가 확인됩니다.

조작의 자세한 내용은 📖"Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

● 전용 명령을 사용하는 방법

처리 단계	조작 방법
시작 지시	사용자 프로그램 내에 BackupToMemoryCard 명령을 실행합니다.
실행 중	BackupToMemoryCard 명령의 출력 변수 Busy의 값이 TRUE가 됩니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	정상 종료인 경우 BackupToMemoryCard 명령의 출력 변수 Done의 값이 TRUE가 됩니다. 이상 종료인 경우 BackupToMemoryCard 명령의 출력 변수 Error의 값이 TRUE가 됩니다. BackupToMemoryCard 명령의 출력 변수 ErrorID에 에러 코드가 저장됩니다.

버전 관련 정보

BackupToMemoryCard 명령은 CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상에서 사용할 수 있습니다.

관련 시스템 정의 변수

시스템 정의 변수에 의한 방법으로 백업을 실시하는 경우에 관련 시스템 정의 변수에는 다음과 같은 것이 있습니다. 시스템 정의 변수의 자세한 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'을 참조하십시오.

변수명 / 멤버명	이름	기능	데이터 형식	R/W
_Card1BkupCmd*1	SD 메모리 카드 백업 지시		_sBKUP_CMD	RW
ExecBkup*1	백업 기동 플래그	컨트롤러의 데이터를 SD 메모리 카드로 백업할 때 TRUE로 합니다.	BOOL	RW
CancelBkup*1	백업 취소 플래그	SD 메모리 카드로의 백업을 중지할 때 TRUE로 합니다.	BOOL	RW
DirName*1	디렉토리명	백업 대상인 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING(64)	RW
_Card1BkupSta*1	SD 메모리 카드 백업 Status		_sBKUP_STA	R
Done*1	완료 플래그	백업 실행 완료일 때 TRUE가 됩니다.	BOOL	R
Active*1	실행 중 플래그	백업 실행 중일 때 TRUE가 됩니다.	BOOL	R
Err*1	에러 플래그	백업 처리가 이상 종료했을 때 TRUE가 됩니다.	BOOL	R
_BackupBusy	백업 관련 기능 실행 중 플래그	백업, 복원, 비교 실행 중일 때 TRUE가 됩니다.	BOOL	R

*1. 사용자 프로그램에서는 이 시스템 정의 변수에 액세스할 수 없습니다.

참고

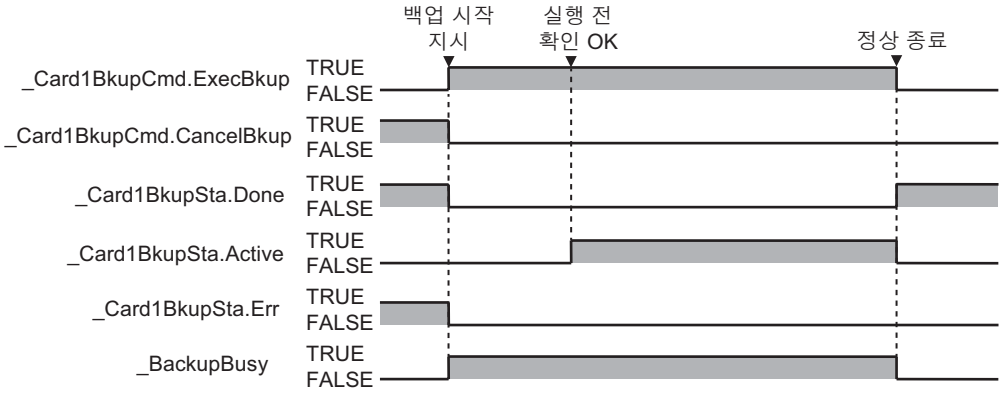
- NJ/NX 시리즈 컨트롤러와 NA 시리즈 표시기를 연결하여 사용하는 경우의 변수 매핑에 대해서는 "프로그래머블 터미널 NA 시리즈 사용 설명서 소프트웨어편(SBSA-546)"을 참조하십시오.
- NJ/NX 시리즈 컨트롤러와 NS 시리즈 표시기를 연결하여 사용하는 경우 CX-Designer의 변수 테이블에 이러한 시스템 정의 변수를 등록하는 방법에 대해서는 'A-11 CX-Designer의 변수 테이블 등록 방법(A-198페이지)'을 참조하십시오.

타이밍 차트

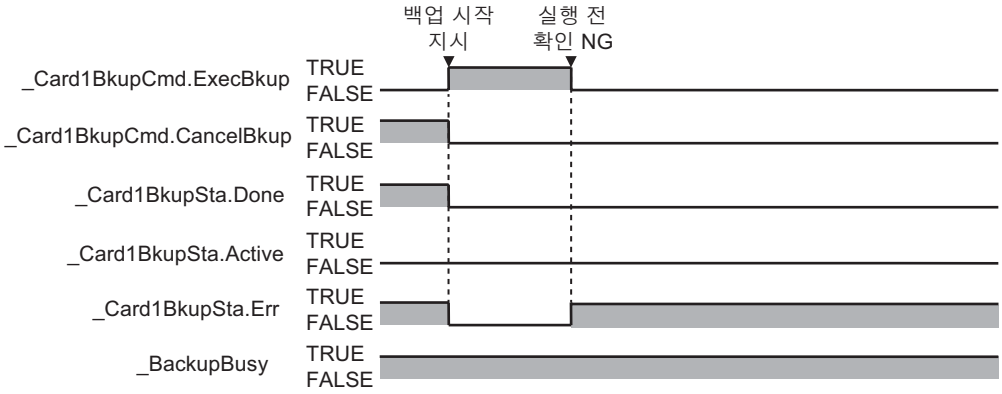
시스템 정의 변수를 사용하는 방법으로 백업을 실시하는 경우의 시스템 정의 변수의 동작은 다음과 같습니다.

그림의 '실행 전 확인'은 백업이 시작되기 전에 SD 메모리 카드가 존재하는지 여부 등을 확인하는 처리입니다. _Card1BkupSta.Active(실행 중 플래그)의 값이 TRUE가 되는 것은 실행 전 확인 결과가 OK가 되고, 실제로 백업이 시작되었을 때입니다.

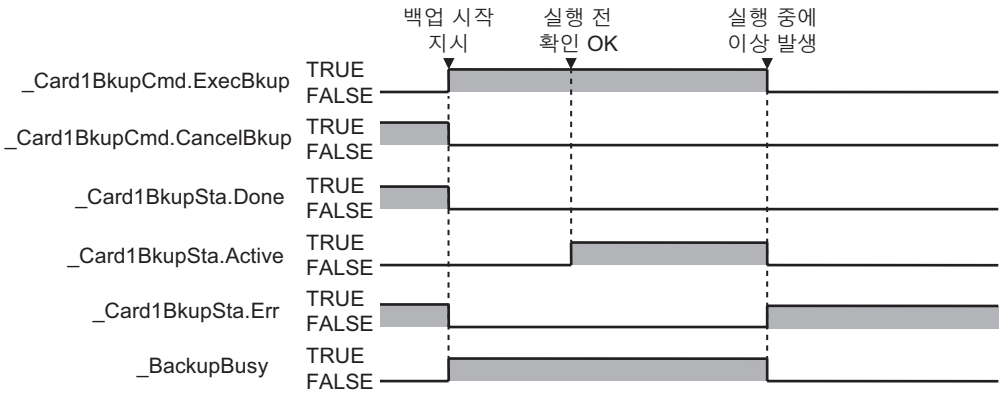
- **정상 동작**



- **다른 백업 기능 실행 중이라서 백업을 시작할 수 없었던 경우**

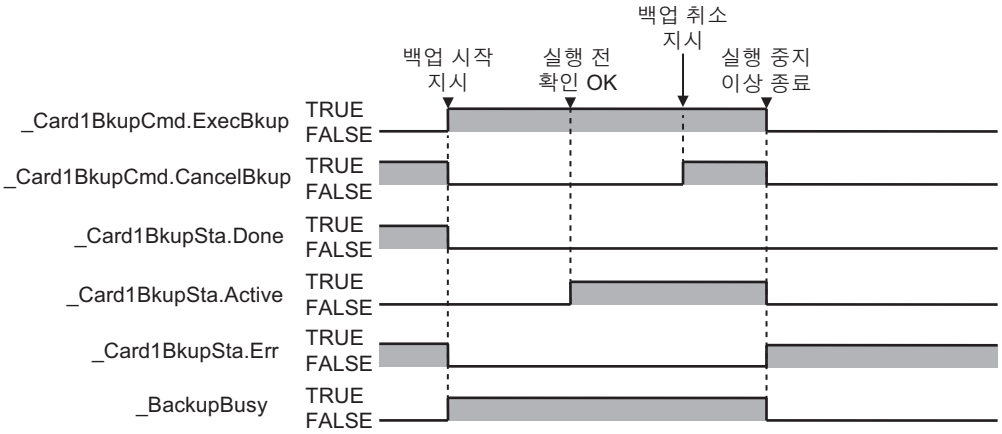


- **정상적으로 백업을 시작한 후에 어떤 원인에 의해 백업에 실패한 경우**



● **백업 실행 중에 취소 지시를 내린 경우**

취소 지시를 내린 후 백업 실행 중지까지의 시간은 백업 처리의 진행 상황에 따라 변화합니다.



처리 시간과 백업 파일의 크기

백업에 필요한 시간은 CPU 유니트, 동작 모드, 유니트 구성, 사용자 프로그램의 내용 등에 따라 다릅니다. 또한, 백업 파일의 크기는 유니트 구성 및 사용자 프로그램의 내용 등에 따라 다릅니다. 백업 시간과 백업 파일 크기의 기준은 다음과 같습니다.

CPU 유니트	동작 모드	연결한 EtherCAT 슬레이브	연결한 NX 유니트 또는 CJ 유니트	연결한 옵션 보드	사용자 정의의 POU 수	사용자 프로그램의 메모리 크기[MB]	백업 시간 [s]	백업 파일의 크기[MB]
NX701-□□□□	프로그램 모드	*1	–	–	113	7.13	약 30	29.45
NX102-□□□□		*2	*3	–	15	0.37	약 30	13.8
NX1P2-□□□□		*2	*3	*4	15	0.35	약 30	2.96
NJ501-□□□□		*1	*5	–	53	2.36	약 50	20
NJ301-□□□□		*6		–	20	0.53	약 30	9.85
NJ101-□□□□		*2		–	15	0.38	약 30	9.94

*1. AC 서보 드라이버 R88D-KNA-ECT: 32대, 디지털 I/O 터미널 GX-ID1611: 32대, 디지털 I/O 터미널 GX-OD1611: 32대의 구성
*2. AC 서보 드라이버 R88D-KNA-ECT: 2대, 디지털 I/O 터미널 GX-ID1611: 2대, 디지털 I/O 터미널 GX-OD1611: 2대의 구성
*3. I/O 전원 추가 공급 유니트 NX-PF0630: 1대, 디지털 입력 유니트 NX-ID5342: 3대, 디지털 출력 유니트 NX-OD3153: 2대, 아날로그 입력 유니트 NX-AD4608: 1대, 아날로그 출력 유니트 NX-DA3605: 1대의 구성
*4. 시리얼 통신용 옵션 보드 NX1W-CIF01: 2대의 구성
*5. 시리얼 커뮤니케이션 유니트 CJ1W-SCU22: 4대, EtherNet/IP 유니트 CJ1W-EIP21: 1대의 구성
*6. AC 서보 드라이버 R88D-KNA-ECT: 8대, 디지털 I/O 터미널 GX-ID1611: 8대, 디지털 I/O 터미널 GX-OD1611: 8대의 구성

9-2-2 복원(SD 메모리 카드에서 컨트롤러)

CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터를 컨트롤러에 전송합니다.
이 기능의 조작은 CPU 유니트 전면 스위치에 의한 방법 및 시스템 정의 변수에 의한 방법으로 가능합니다.

CPU 유니트 전면 스위치에 의한 방법으로는 전원 투입 시 전송을 시작합니다. 복원하는 대상은 복원 명령 파일을 사용하여 지정할 수 있습니다. 백업 관련 파일을 읽어올 디렉토리를 지정할 수 없습니다. 복원하는 백업 파일은 SD 메모리 카드의 루트 디렉토리에 저장해두어야 합니다.

시스템 정의 변수에 의한 방법에서는 시스템 정의 변수에 의해 복원하는 대상 및 백업 관련 파일을 읽어올 디렉토리를 지정하고, 전송 시작을 지시합니다. 또한, 컨트롤러 설정에서 이 기능을 사용할지 여부 및 비밀번호를 설정할 수 있습니다. 표시기 조작 등을 통해 SD 메모리 카드 내의 백업 파일 데이터로 CPU 유니트를 운전하고자 하는 경우 사용합니다.

처리 내용

CPU 유니트 전면 스위치에 의한 방법 및 시스템 정의 변수에 의한 방법에 대해 각각의 처리 내용을 설명합니다.

● CPU 유니트 전면 스위치를 사용한 방법

- SD 메모리 카드의 루트 디렉토리에 저장된 백업 파일의 데이터를 컨트롤러에 전송합니다.
- 복원 대상 데이터는 루트 디렉토리에 저장된 복원 명령 파일(RestoreCommand.ini)에서 '복원'으로 지정한 데이터 그룹입니다. 복원 명령 파일의 자세한 내용은 📖'9-13-3 복원 명령 파일의 사양(9-64페이지)'을 참조하십시오.
- SD 메모리 카드의 루트 디렉토리에 복원 명령 파일이 존재하지 않는 경우에 루트 디렉토리에 존재하는 복원 가능한 백업 파일의 데이터를 모두 복원합니다.
- 실행 완료 후에는 동작 모드가 '프로그램 모드'가 되고, 그대로는 운전을 시작할 수 없습니다. 운전을 시작하려면 딥 스위치를 모두 OFF로 한 후에 컨트롤러의 전원 재투입 또는 '컨트롤러 리셋'을 실행하십시오.
- 복원 실행 완료 후에는 모든 EtherCAT 슬레이브의 전원을 재투입하십시오.
- 복원 실행 중인 CPU 유니트의 상태는 '기동 중'입니다.
- 복원 처리 시작 전 확인에서 에러가 발생한 경우에는 컨트롤러 내의 데이터는 복원 전의 데이터를 유지합니다.
- 복원 처리 실행 중에 컨트롤러의 전원 OFF가 발생한 경우에는 '사용자 프로그램/컨트롤러 구성·설정 전송 에러(전체 정지 이상 레벨의 컨트롤러 이상)'가 발생합니다. 이 때 컨트롤러 내의 데이터는 올바르지 않으므로 다음 방법으로 이상을 제거하십시오.
  - 복원을 다시 실시
  - Sysmac Studio에서 메모리 전체 클리어, 프로젝트의 다운로드를 실시
- 백업 파일의 유니트·슬레이브 구성과 복원 대상인 실제 유니트·슬레이브 구성이 다른 경우에 복원을 실행하면 복원 실행 에러가 발생합니다.
- NX1P2 CPU 유니트에서 백업 파일의 옵션 보드 구성과 복원 대상인 실제 옵션 보드 구성이 다른 경우에도 복원할 수 있습니다. 단, 옵션 보드는 동작하지 않습니다. 자세한 내용은 📖"NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)"를 참조하십시오.

- 유지 속성이 있는 변수의 현재값이 복원 대상이 아닌 경우에 해당 변수의 현재값은 복원 전의 값을 유지합니다. 단, 변수 자체를 유지하는 조건을 충족하지 않은 경우에는 변수의 값을 초기화합니다. 변수 자체를 유지하는 조건은 다음과 같습니다.
  - 복원 전의 변수와 복원하는 변수의 변수명, 데이터 형식명, 데이터 형식의 크기가 모두 동일할 것
- NX102 CPU 유니트 및 NX1P2 CPU 유니트인 경우에 백업 파일의 CJ 유니트용 메모리 설정 내용이며, CJ 유니트용 메모리가 생성됩니다.
- NX102 CPU 유니트 및 NX1P2 CPU 유니트에서 CJ 유니트용 메모리의 현재값이 복원 대상이 아닌 경우에 HR, DM, EM 영역에 대해서는 복원 전의 값을 유지합니다. 단, HR, DM, EM 영역이 새로 생성되었거나 영역이 확장된 경우에 이들의 영역에 대해서는 초기값이 됩니다.
- 복원 완료 후, SD 메모리 카드는 마운트 상태를 지속합니다.
- 복원 완료 후의 CPU 유니트의 쓰기 방지 상태는 '전원 투입 시의 쓰기 방지' 설정에 따릅니다.

● 시스템 정의 변수에 의한 방법

사용상의 주의 사항

- 복원 대상 데이터는 시스템 정의 변수로 지정한 데이터 그룹입니다. 복원 명령 파일(RestoreCommand.ini)의 지정은 영향을 주지 않습니다. 관련 시스템 정의 변수의 자세한 내용은 📖'관련 시스템 정의 변수(9-23페이지)'를 참조하십시오.
- 예상치 못한 복원을 방지하기 위해 비밀번호를 매번 입력해야 하도록 설계하십시오.
- 이 기능을 실행하면 자동으로 컨트롤러가 리셋됩니다. 컨트롤러 리셋 중의 출력은 슬레이브 및 유니트의 사양에 따릅니다. 또한, 컨트롤러 리셋 중에는 컨트롤러 외부에서 컨트롤러 내의 변수에 액세스할 수 없습니다.
- 복원에 의해 전송 전에 참조했던 컨트롤러 측의 변수가 삭제된 경우에 표시기에 따라서는 삭제된 변수를 해결할 수 없어서 시스템 정의 변수에 액세스할 수 없는 경우가 있습니다. 예를 들어 NS 시리즈 표시기인 경우에 태그 비교 결과가 일치하지 않으면 태그 비교 결과 목록을 표시하고, 다른 화면으로 이동할 수 없게 됩니다. 표시기에서 사용하는 변수가 삭제되지 않았는지 사전에 확인하십시오.
- 이 기능 실행 중에 전원 OFF한 경우에 다음 번 전원 투입 시 '사용자 프로그램/컨트롤러 구성·설정 전송 에러'(이벤트 코드 10200000Hex) 등의 이상이 발생하는 경우가 있습니다.
- 컨트롤러 설정의 [시스템 정의 변수에 의한 복원]을 [사용 안 함]으로 설정한 백업 파일을 전송한 경우에 이후에는 시스템 정의 변수에 의한 복원을 실행할 수 없습니다. 반복해서 실행하는 경우에는 [사용]으로 설정한 백업 파일을 전송하십시오.
- 복원 중에 다른 백업 기능을 실행할 수 없습니다.

- 시스템 정의 변수 _Card1RestoreCmd(SD 메모리 카드 복원 지시)의 조작에 의해 CPU 유니트에 장착한 SD 메모리 카드에 저장된 데이터를 컨트롤러에 전송합니다.
- 복원 대상 백업 파일은 시스템 정의 변수로 지정한 디렉토리에 저장된 파일입니다. 대상 백업 파일은 사전에 SD 메모리 카드의 임의의 디렉토리에 저장해두어야 합니다.
- 컨트롤러 설정의 [SD 메모리 카드 복원 설정]의 [비밀번호]에 설정한 비밀번호와 시스템 정의 변수 _Card1RestoreCmd.Password에 설정한 비밀번호가 일치하지 않는 경우에 '복원 시작 실패'가 발생합니다.

- 복원을 시작하면 시스템 정의 변수 _Card1RestoreCmd.Password에 설정한 비밀번호는 초기화됩니다.
- 복원 실행 도중에 자동으로 컨트롤러 리셋됩니다.
- 컨트롤러 리셋 후의 CPU 유니트의 상태는 '기동 중'입니다. 그 후에 복원 실행이 완료되면 동작 모드가 '프로그램 모드'가 되고, 그대로는 운전을 시작할 수 없습니다. 운전을 시작하려면 컨트롤러의 전원 재투입 또는 '컨트롤러 리셋'을 실행하십시오.
- 복원 실행 완료 후에는 모든 EtherCAT 슬레이브의 전원을 재투입하십시오.
- 복원 처리 시작 전 확인 또는 실행 전 확인에서 에러가 발생했을 때, 컨트롤러 내의 데이터는 전송 전의 데이터를 유지합니다.
- 복원 처리 실행 중에 컨트롤러의 전원 OFF가 발생했을 때는 '사용자 프로그램/컨트롤러 구성·설정 전송 에러(전체 정지 이상 레벨의 컨트롤러 이상)'가 발생합니다. 이 때 컨트롤러 내의 데이터는 올바르지 않으므로 다음 방법으로 이상을 제거하십시오.
  - 복원을 다시 실시
  - Sysmac Studio에서 메모리 전체 클리어, 프로젝트의 다운로드를 실시
- 복원 대상 데이터 이외는 현재값을 유지합니다.
- 유지 속성이 있는 변수의 현재값이 전송 대상이 아닌 경우에 해당 변수의 현재값은 복원 전의 값을 유지합니다. 단, 변수 자체를 유지하는 조건을 충족하지 않은 경우에는 변수의 값을 초기화합니다. 변수 자체를 유지하는 조건은 다음과 같습니다.
  - 전송 전의 변수와 전송하는 변수의 변수명, 데이터 형식명, 데이터 형식의 크기가 모두 동일할 것
- NX102 CPU 유니트 및 NX1P2 CPU 유니트에서 CJ 유니트용 메모리의 현재값이 복원 대상이 아닌 경우에 HR, DM, EM 영역에 대해서는 복원 전의 값을 유지합니다. 단, HR, DM, EM 영역이 새로 생성되었거나 영역이 확장된 경우에 이들의 영역에 대해서는 초기값이 됩니다.
- 복원 중에 SD 메모리 카드 전원 공급 정지 버튼을 눌러도 전원 공급을 계속합니다.
- 복원 완료 후, SD 메모리 카드는 마운트 상태를 지속합니다.
- 복원 완료 후의 CPU 유니트의 쓰기 방지 상태는 '전원 투입 시의 쓰기 방지' 설정에 따릅니다.

조작 방법

● CPU 유니트 전면 스위치를 사용한 방법

처리 단계	조작 순서
시작 지시	딥 스위치 1에서 4[*1] 를 1: OFF, 2: OFF, 3: ON, 4: ON으로 하여 컨트롤러의 전원을 투입합니다.
실행 중	복원 실행 중 SD PWR가 3s 점등, 0.5s 소등, RUN이 0.5s 점등, 0.5s 소등으로 각각 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다.
실행 결과	정상적으로 종료한 경우 SD PWR가 점등됩니다. 이상 종료한 경우 SD PWR가 0.5s 점등, 0.5s 소등으로 점멸됩니다. SD 메모리 카드 전원 공급 정지 버튼을 누르면 점등됩니다.[*2]

*1. NX701 CPU 유니트인 경우에 딥 스위치의 5에서 8을 모두 OFF로 하십시오.
*2. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR는 소등된 상태 그대로 유지됩니다.

● 시스템 정의 변수 _Card1RestoreCmd(SD 메모리 카드 복원 지시)에 의한 방법

처리 단계	조작 순서
시작 전의 준비	시스템 정의 변수에 의한 복원을 사용하는 경우에는 컨트롤러 설정의 [시스템 정의 변수에 의한 복원]을 [사용]으로 설정해둡니다.*1
시작 지시	시스템 정의 변수 _Card1RestoreCmd.DirName(디렉토리명)에 백업 파일이 저장된 디렉토리명을 저장합니다. (예) 디렉토리 dirA 아래의 서브 디렉토리 dirB를 지정: "dirA/dirB" 시스템 정의 변수 _Card1RestoreCmd.Password(비밀번호)에 비밀번호를 저장합니다.*2 내장 EtherNet/IP 포트의 IP 주소를 복원 대상으로 할 때는 시스템 정의 변수 _Card1RestoreCmdTargetIPAdr(IP 주소의 전송 플래그)를 TRUE로 합니다.*3 변수의 현재값(유지 속성만 해당)를 복원 대상으로 할 때는 시스템 정의 변수 _Card1RestoreCmdTargetVariable(변수의 현재값(유지 속성만 해당)의 전송 플래그)을 TRUE로 합니다. CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 복원 대상으로 할 때는 시스템 정의 변수 _Card1RestoreCmdTargetMemory(CJ 유니트용 메모리의 현재값(유지 메모리만 해당)의 전송 플래그)를 TRUE로 합니다. 유니트·슬레이브 설정을 복원 대상으로 할 때는 시스템 정의 변수 _Card1RestoreCmdTargetUnitConfig(유니트·슬레이브 설정의 전송 플래그)를 TRUE로 합니다. 앱솔루트 엔코더 원점 위치 오프셋을 복원 대상으로 할 때는 시스템 정의 변수 _Card1RestoreCmdTargetAbsEncoder(앱솔루트 엔코더 원점 위치 오프셋의 전송 플래그)를 TRUE로 합니다. 시스템 정의 변수 _Card1RestoreCmd.Exec(복원 기동 플래그)를 FALSE→TRUE로 하면 실행이 시작됩니다.
실행 중	시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)가 TRUE가 됩니다. 시스템 정의 변수 _Card1RestoreSta.Active(실행 중 플래그)가 TRUE가 됩니다. 복원 실행 중 SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다. RUN이 0.5s 점등, 0.5s 소등으로 점멸합니다. 복원 실행 도중에 자동으로 컨트롤러 리셋됩니다.
실행 결과	정상적으로 종료한 경우 SD PWR가 점등됩니다. RUN이 소등됩니다. 시스템 정의 변수 _Card1RestoreSta.Done(완료 플래그)이 TRUE가 됩니다. 시작 전 확인에서 이상 종료한 경우 SD PWR가 점등됩니다.*4 RUN과 ERR는 시작 지시 전의 상태입니다. 시스템 정의 변수 _Card1RestoreSta.Err(에러 플래그)가 TRUE가 됩니다. 시작 전 확인에서 이상 종료한 경우에는 컨트롤러 리셋이 발생하지 않습니다. 실행 전 확인 또는 실행 중에 이상 종료한 경우 SD PWR가 점등됩니다. *4 RUN이 소등됩니다. 시스템 정의 변수 _Card1RestoreSta.Err(에러 플래그)가 TRUE가 됩니다.

*1. 컨트롤러 설정의 [시스템 정의 변수에 의한 복원]을 [사용 안 함]으로 설정한 백업 파일을 전송한 경우에 이후에는 시스템 정의 변수에 의한 복원을 실행할 수 없습니다. 반복해서 실행하는 경우에는 [사용]으로 설정한 백업 파일을 전송하십시오.

*2. 시스템 정의 변수에 의한 복원을 시작하면 비밀번호가 초기화됩니다. 시작할 때마다 비밀번호를 저장하십시오. 컨트롤러 설정의 [SD 메모리 카드 복원 설정]의 [비밀번호]에 비밀번호가 설정되지 않은 경우에 시스템 정의 변수 _Card1RestoreCmd.Password의 값이 초기값이면 복원을 시작합니다. 초기값 이외이면 복원을 시작하지 않습니다.

*3. IP 주소에는 설정 유형, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.

*4. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR는 소등된 상태 그대로 유지됩니다.

관련 시스템 정의 변수

관련 시스템 정의 변수는 다음과 같습니다. 시스템 정의 변수의 자세한 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'을 참조하십시오.

변수명 멤버명	이름	기능	데이터 형식	R/W
_Card1RestoreCmd	SD 메모리 카드 복원 지시		_sRESTORE_CMD	RW
Exec	복원 기동 플래그	SD 메모리 카드 내의 백업 파일의 데이터를 컨트롤러로 복원할 때 TRUE로 합니다.	BOOL	RW
DirName	디렉토리명	시스템 정의 변수에 의한 복원 대상인 백업 파일이 저장된 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING(64)	RW
Password	비밀번호	시스템 정의 변수에 의한 복원 시작 시에 비교될 비밀번호를 지정합니다. 시스템 정의 변수에 의한 복원을 시작할 때마다 비밀번호가 초기화됩니다.	STRING(33)	RW
_Card1RestoreSta	SD 메모리 카드 복원 Status		_sRESTORE_STA	R
Done	완료 플래그	복원 실행 완료일 때 TRUE가 됩니다.	BOOL	R
Active	실행 중 플래그	복원 실행 중일 때 TRUE가 됩니다.	BOOL	R
Err	에러 플래그	복원이 이상 종료됐을 때 TRUE가 됩니다.	BOOL	R

변수명	이름	기능	데이터 형식	R/W
_Card1RestoreCmdTargetUserProgram	사용자 프로그램·설정 전송 플래그	시스템 정의 변수에 의한 복원에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. 시스템 정의 변수에 의한 복원일 때는 반드시 TRUE로 하십시오.	BOOL	RW
_Card1RestoreCmdTargetIPAdr	IP 주소 전송 플래그	시스템 정의 변수에 의한 복원에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 유형, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.	BOOL	RW
_Card1RestoreCmdTargetVariable	변수의 현재값(유지 속성만 해당) 전송 플래그	시스템 정의 변수에 의한 복원에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW
_Card1RestoreCmdTargetMemory	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그	시스템 정의 변수에 의한 복원에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW
_Card1RestoreCmdTargetUnitConfig	유니트·슬레이브 설정 전송 플래그	시스템 정의 변수에 의한 복원에서 유니트·슬레이브 설정을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW
_Card1RestoreCmdTargetAbsEncoder	앱솔루트 엔코더 원점 위치 오프셋 전송 플래그	시스템 정의 변수에 의한 복원에서 앱솔루트 엔코더 원점 위치 오프셋을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW

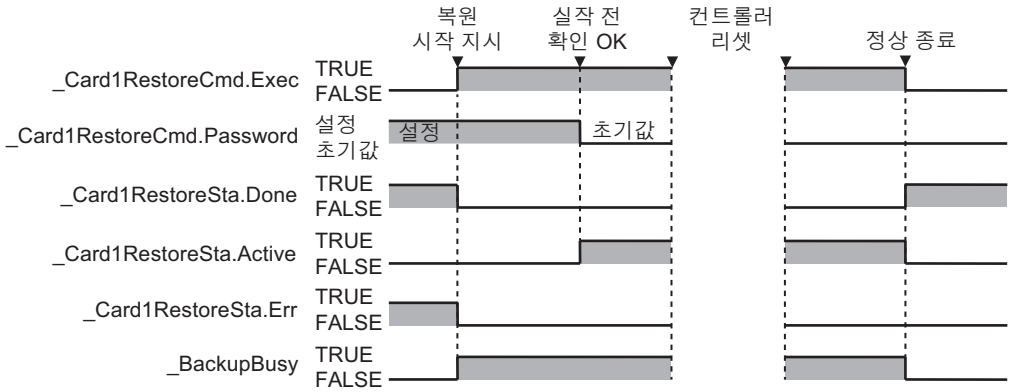
참고

- NJ/NX 시리즈 컨트롤러와 NA 시리즈 표시기를 연결하여 사용하는 경우의 변수 매핑에 대해서는 "프로그래머블 터미널 NA 시리즈 사용 설명서 소프트웨어편(SBSA-546)"을 참조하십시오.
- NJ/NX 시리즈 컨트롤러와 NS 시리즈 표시기를 연결하여 사용하는 경우 CX-Designer의 변수 테이블에 이러한 시스템 정의 변수를 등록하는 방법에 대해서는 'A-11 CX-Designer의 변수 테이블 등록 방법(A-198페이지)'을 참조하십시오.

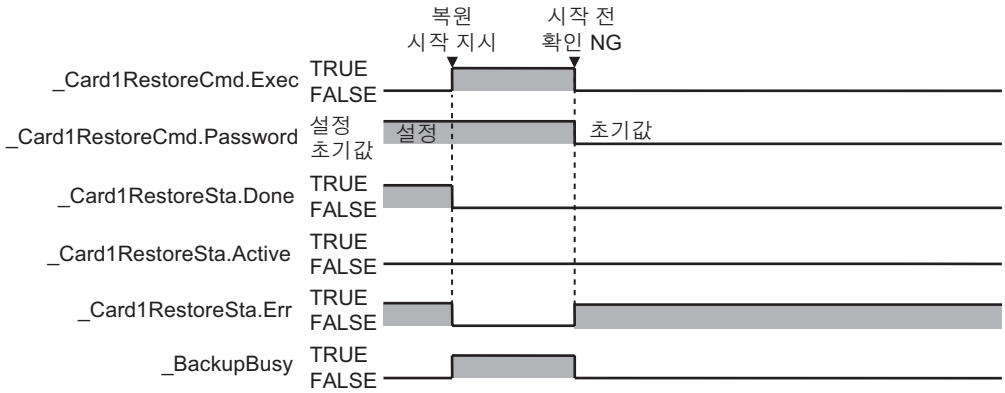
타이밍 차트

시스템 정의 변수를 사용하는 방법으로 SD 메모리 카드 백업 기능을 복원하는 경우의 시스템 정의 변수의 동작은 다음과 같습니다.
그림에 나오는 '시작 전 확인'은 복원이 시작되기 전에 비밀번호가 일치하는지 여부 등을 확인하는 처리입니다. _Card1PrgRestoreSta.Active(실행 중 플래그)의 값이 TRUE가 되는 것은 시작 전 확인 결과가 OK가 되었을 때입니다.

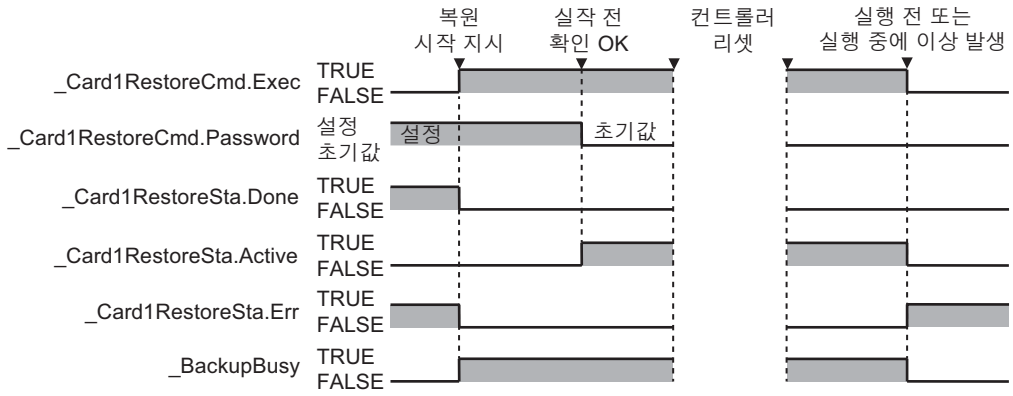
- **정상 동작**



- **다른 백업 기능 실행 중이라서 복원을 시작할 수 없었던 경우**



● **정상적으로 복원을 시작한 후에 어떤 원인에 의해 복원에 실패한 경우**



처리 시간

복원에 필요한 시간은 CPU 유니트, 유니트 구성, 사용자 프로그램의 내용 등에 따라 다릅니다. 다음은 복원 시간의 기준입니다.

CPU 유니트	연결한 EtherCAT 슬레이브	연결한 NX 유니트 또는 CJ 유니트	연결한 옵션 보드	사용자 정의의 POU 수	사용자 프로그램의 메모리 크기[MB]	복원 시간[s]
NX701-□□□□	*1	–	–	113	7.13	약 50
NX102-□□□□	*2	*3	–	15	0.37	약 70
NX1P2-□□□□	*2	*3	*4	15	0.35	약 60
NJ501-□□□□	*1	*5	–	53	2.36	약 100
NJ301-□□□□	*6		–	20	0.53	약 70
NJ101-□□□□	*2		–	15	0.38	약 70

*1. AC 서보 드라이버 R88D-KNA-ECT: 32대, 디지털 I/O 터미널 GX-ID1611: 32대, 디지털 I/O 터미널 GX-OD1611: 32대의 구성
*2. AC 서보 드라이버 R88D-KNA-ECT: 2대, 디지털 I/O 터미널 GX-ID1611: 2대, 디지털 I/O 터미널 GX-OD1611: 2대의 구성
*3. I/O 전원 추가 공급 유니트 NX-PF0630: 1대, 디지털 입력 유니트 NX-ID5342: 3대, 디지털 출력 유니트 NX-OD3153: 2대, 아날로그 입력 유니트 NX-AD4608: 1대, 아날로그 출력 유니트 NX-DA3605: 1대의 구성
*4. 시리얼 통신용 옵션 보드 NX1W-CIF01: 2대의 구성
*5. 시리얼 커뮤니케이션 유니트 CJ1W-SCU22: 4대, EtherNet/IP 유니트 CJ1W-EIP21: 1대의 구성
*6. AC 서보 드라이버 R88D-KNA-ECT: 8대, 디지털 I/O 터미널 GX-ID1611: 8대, 디지털 I/O 터미널 GX-OD1611: 8대의 구성

9-2-3 비교(컨트롤러와 SD 메모리 카드 간)

컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일의 데이터를 비교합니다.

처리 내용

- 컨트롤러의 데이터와 SD 메모리 카드의 지정 디렉토리에 저장된 백업 파일의 데이터를 비교합니다.
- 비교 대상 데이터는 복원 명령 파일(RestoreCommand.ini)에서 '복원'으로 지정한 데이터 그룹입니다.
- 변수의 현재값, CJ 유니트용 메모리의 현재값, 앱솔루트 엔코더 원점 위치 오프셋은 비교 실행 중에 값이 재기입될 수 있기 때문에 비교하지 않습니다.
- 비교를 실행하면 '비교 결과 파일(VerifyResult.log)'이 지정 디렉토리에 작성되고, 비교 결과가 저장됩니다. 비교 실행 전에 지정 디렉토리에 비교 결과 파일이 있을 때는 덮어씁니다. 단, SD 메모리 카드가 쓰기 방지된 경우에는 비교 결과 파일은 작성되지 않습니다.
- SD 메모리 카드의 지정 디렉토리에 복원 명령 파일이 존재하지 않는 경우에 지정 디렉토리에 존재하는 비교 가능한 백업 파일의 데이터를 모두 비교합니다.
- 백업 파일의 유니트·슬레이브 구성과 비교 대상인 실제 유니트·슬레이브 구성이 다른 경우에 비교 에러가 발생합니다.
- 비교 완료 후에 SD 메모리 카드는 마운트 상태를 지속합니다.

조작 방법

● CPU 유니트 전면 스위치를 사용한 방법

처리 단계	조작 순서
시작 지시	딥 스위치 1에서 4[*1] 를 1: OFF, 2: OFF, 3: OFF, 4: OFF로 하고, SD 메모리 카드 전원 공급 정지 버튼을 3초간 계속 누르면 실행이 시작됩니다.
실행 중	실행 시작 직후[*2] SD PWR가 점등→0.5s 소등→점등의 패턴으로 1회 점멸합니다. 비교 실행 중 SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	정상 종료되고 비교 일치인 경우 SD PWR가 점등됩니다. 정상 종료되고 비교 불일치인 경우 SD PWR가 0.5s 점등, 0.5s 소등으로 점멸됩니다. SD 메모리 카드 전원 공급 정지 버튼을 누르면 점등됩니다. 이상 종료인 경우 SD PWR가 0.5s 점등, 0.5s 소등으로 점멸됩니다. SD 메모리 카드 전원 공급 정지 버튼을 누르면 점등됩니다. [*1]

*1. NX701 CPU 유니트인 경우에 딥 스위치의 5에서 8을 모두 OFF로 하십시오.
*2. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR는 소등된 상태 그대로 유지됩니다.

● 시스템 정의 변수 _Card1BkupCmd(SD 메모리 카드 백업 지시)

처리 단계	조작 순서
시작 지시	시스템 정의 변수 _Card1BkupCmd.DirName(디렉토리명)에 저장 장소의 디렉토리명을 저장합니다. (예) 디렉토리 dirA 아래의 서브 디렉토리 dirB를 지정: "dirA/dirB" 시스템 정의 변수 _Card1BkupCmd.ExecVefy(비교 기동 플래그)를 FALSE→TRUE로 하면 실행이 시작됩니다.
취소 지시	비교 실행 지시를 취소합니다. 시스템 정의 변수 _Card1BkupCmd.CancelVefy(비교 취소 플래그)를 TRUE로 하면 비교 처리가 이상 종료합니다.
실행 중	시스템 정의 변수 _Card1VefySta.Active(실행 중 플래그)가 TRUE가 됩니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.
실행 결과	정상 종료되고 비교 일치인 경우 시스템 정의 변수 _Card1BkupSta.Done(완료 플래그)과 _Card1BkupSta.VefyRslt(비교 결과 플래그)가 TRUE가 됩니다. 정상 종료되고 비교 불일치인 경우 시스템 정의 변수 _Card1BkupSta.Done(완료 플래그)이 TRUE, _Card1BkupSta.VefyRslt(비교 결과 플래그)가 FALSE가 됩니다. 이상 종료인 경우 시스템 정의 변수 _Card1BkupSta.Err(에러 플래그)가 TRUE가 됩니다.

(주의) 사용자 프로그램에서는 이 시스템 정의 변수에 액세스할 수 없습니다.

● Sysmac Studio의 SD 메모리 카드 화면에 의한 방법

처리 단계	처리 순서
시작 지시	Sysmac Studio SD 메모리 카드 화면의 [SD 메모리 카드 비교] 버튼을 누르고, 비교 대상인 저장 장소의 디렉토리명을 지정하여 비교 실행을 지시합니다.
실행 중	비교 진행 상황을 나타내는 화면이 표시됩니다. 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다. SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다.
실행 결과	비교 결과를 나타내는 화면이 표시됩니다.

조작의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

관련 시스템 정의 변수

시스템 정의 변수에 의한 방법으로 비교를 실시하는 경우에 관련 시스템 정의 변수에는 다음과 같은 것이 있습니다. 시스템 정의 변수의 자세한 내용은 📖'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'을 참조하십시오.

변수명 / 멤버명	이름	기능	데이터 형식	R/W
_Card1BkupCmd*1	SD 메모리 카드 백업 지시		_sBKUP_CMD	RW
ExecVefy*1	비교 기동 플래그	컨트롤러와 SD 메모리 카드 내의 백업 파일을 비교할 때 TRUE로 합니다.	BOOL	RW
CancelVefy*1	비교 취소 플래그	컨트롤러와 SD 메모리 카드 내의 백업 파일 비교를 중지할 때 TRUE로 합니다.	BOOL	RW
DirName*1	디렉토리명	백업 대상인 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING(64)	RW
_Card1VefySta*1	SD 메모리 카드 비교 Status		_sVEFY_STA	R
Done*1	완료 플래그	비교 실행 완료일 때 TRUE가 됩니다.	BOOL	R
Active*1	실행 중 플래그	비교 실행 중일 때 TRUE가 됩니다.	BOOL	R
VefyRslt*1	비교 결과 플래그	비교 일치일 때 TRUE, 비교 불일치일 때 FALSE가 됩니다.	BOOL	R
Err*1	에러 플래그	비교 처리가 이상 종료되었을 때 TRUE가 됩니다.	BOOL	R
_BackupBusy	백업 관련 기능 실행 중 플래그	백업, 복원, 비교 실행 중일 때 TRUE가 됩니다.	BOOL	R

*1. 사용자 프로그램에서는 이 시스템 정의 변수에 액세스할 수 없습니다.



참고

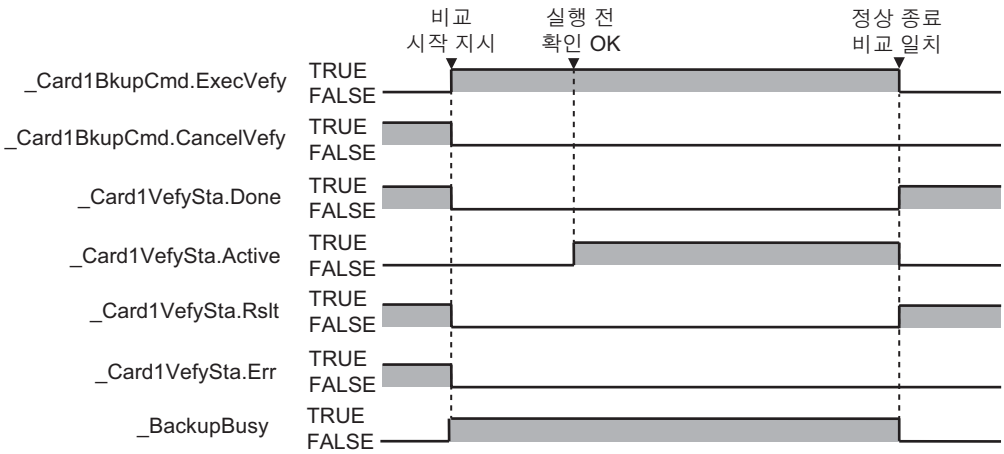
- NJ/NX 시리즈 컨트롤러와 NA 시리즈 표시기를 연결하여 사용하는 경우의 변수 매핑에 대해서는 📖"프로그래머블 터미널 NA 시리즈 사용 설명서 소프트웨어편(SBSA-546)"을 참조하십시오.
- NJ/NX 시리즈 컨트롤러와 NS 시리즈 표시기를 연결하여 사용하는 경우 CX-Designer의 변수 테이블에 이러한 시스템 정의 변수를 등록하는 방법에 대해서는 📖'A-11 CX-Designer의 변수 테이블 등록 방법(A-198페이지)'을 참조하십시오.

타이밍 차트

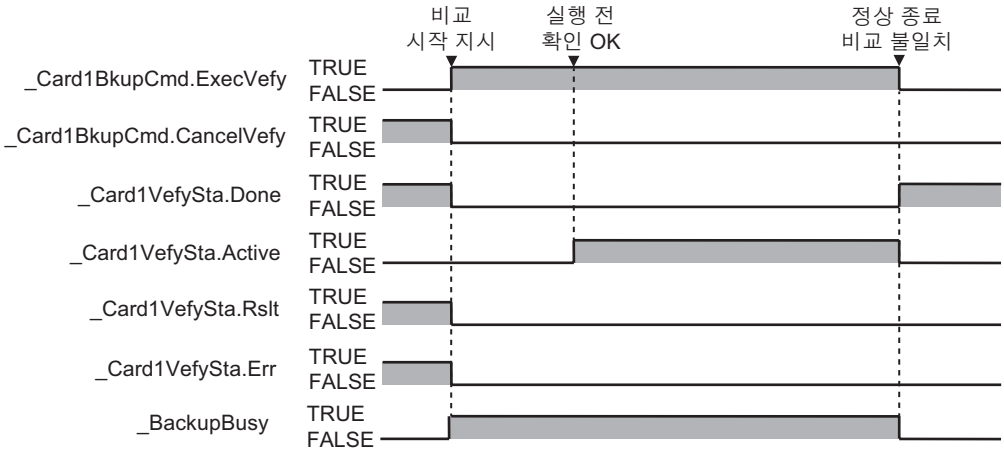
시스템 정의 변수를 사용하는 방법으로 비교를 실시하는 경우의 시스템 정의 변수의 동작은 다음과 같습니다.

그림의 '실행 전 확인'은 비교가 시작되기 전에 SD 메모리 카드가 존재하는지 여부 등을 확인하는 처리입니다. _Card1VefySta.Active(실행 중 플래그)의 값이 TRUE가 되는 것은 실행 전 확인의 결과가 OK가 되고, 실제로 비교가 시작되었을 때입니다.

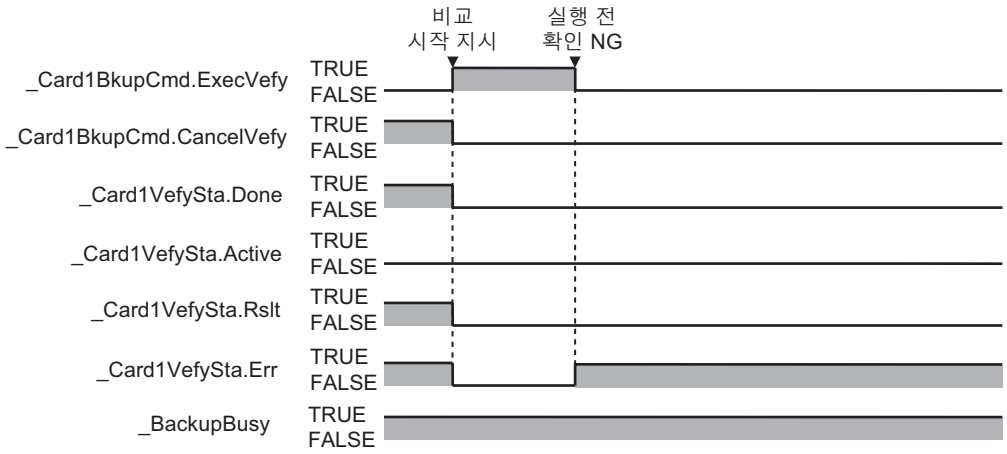
- **정상 동작에서 비교 일치인 경우**



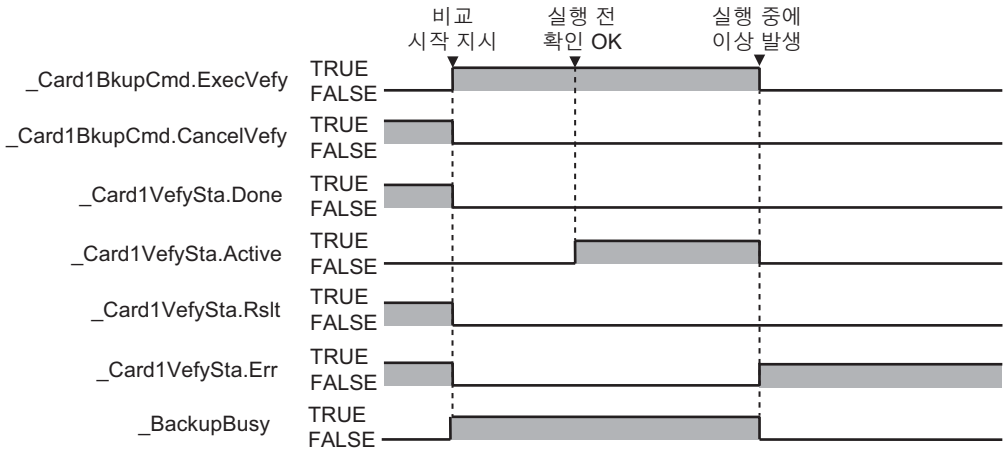
- **정상 동작에서 비교 불일치인 경우**



● **다른 백업 기능 실행 중이라서 비교를 시작할 수 없었던 경우**

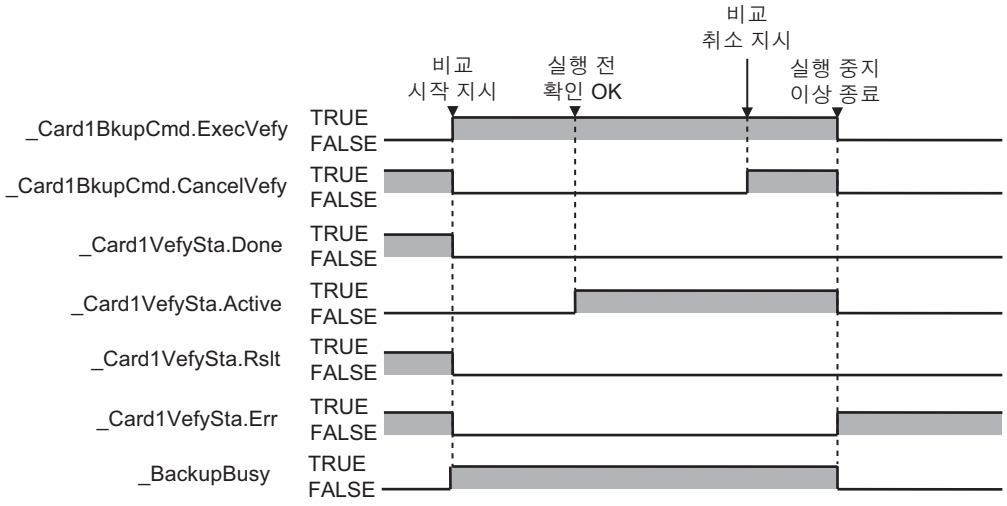


● **정상적으로 비교를 시작한 후에 어떤 원인에 의해 비교에 실패한 경우**



● **비교 실행 중에 취소 지시를 내린 경우**

취소 지시를 내린 후 비교 실행 중지까지의 시간은 비교 처리 진행 상황에 따라 변화합니다.



9-3 SD 메모리 카드에 대한 백업 금지 기능

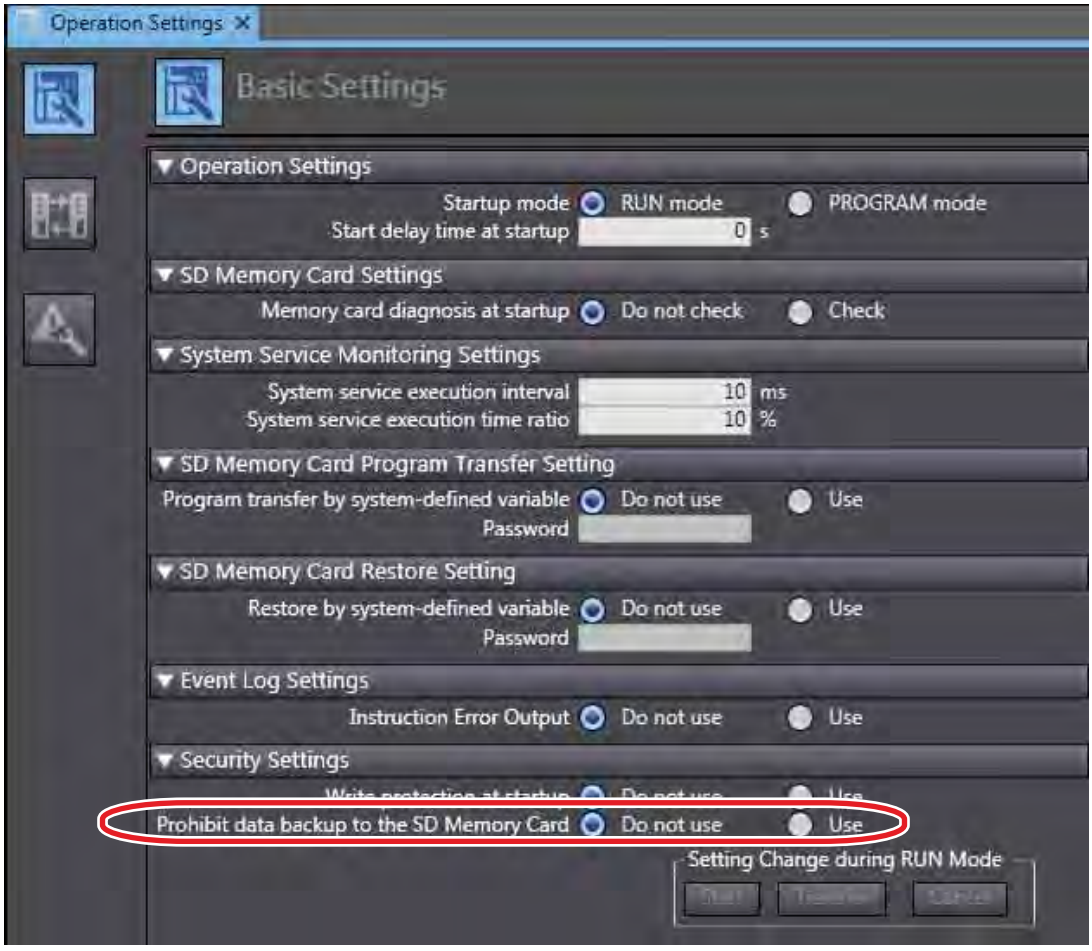
SD 메모리 카드에 대한 백업을 금지하는 기능입니다. 이 기능을 사용하면 사용자 자산 정보를 보호할 수 있습니다.

SD 메모리 카드에 대한 백업 금지 기능의 대상은 다음 3가지입니다.

- CPU 유니트 전면 스위치에 의한 백업
- 시스템 정의 변수에 의한 백업
- Sysmac Studio SD 메모리 카드 화면에 의한 백업

BackupToMemoryCard 명령에 의한 백업은 백업 금지 기능의 대상에서 제외됩니다. 즉, SD 메모리 카드 백업 금지 설정을 해도 BackupToMemoryCard 명령에 의한 백업은 실행할 수 있습니다.

SD 메모리 카드에 대한 백업 금지 조작 방법은 다음과 같습니다.
Sysmac Studio의 '구성·설정'–'컨트롤러 설정'–[동작 설정]의 전반 설정 탭으로 SD 메모리 카드의 백업 금지를 [사용]으로 설정합니다.

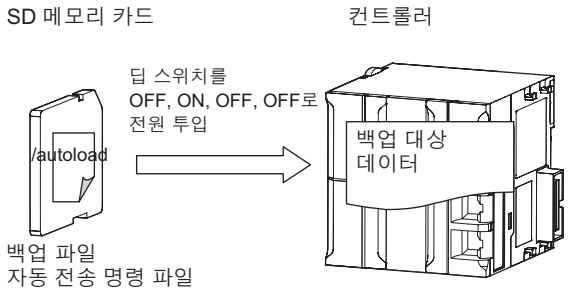


버전 관련 정보

BackupToMemoryCard 명령은 CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상에서 사용할 수 있습니다.

9-4 SD 메모리 카드 자동 전송 기능

전원 투입 시에 CPU 유니트에 장착한 SD 메모리 카드의 '/autoload' 디렉토리에 저장된 백업 파일의 데이터를 자동으로 컨트롤러에 전송하는 기능입니다.
SD 메모리 카드 내의 백업 파일 데이터로 CPU 유니트를 운전하고자 하는 경우 사용합니다.
이 기능의 조작은 CPU 유니트 전면 스위치를 사용하는 방법만 가능합니다.



백업 기능을 사용하여 작성된 '백업 파일'과 '자동 전송 명령 파일'을 사용합니다. 이 2개의 파일을 사전에 '/autoload' 디렉토리에 저장해둡니다.

파일	역할
백업 파일	백업 관련 기능으로 취급하는 컨트롤러의 데이터를 파일화한 것입니다.
자동 전송 명령 파일	자동 전송을 실행할 때 전송하는 대상인 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다.

이 기능의 조작 방법과 대상 디렉토리, 실행 타이밍은 다음과 같습니다.

조작 방법	디렉토리	실행 타이밍
CPU 유니트 전면 스위치	SD 메모리 카드의 /autoload 디렉토리	전원 투입 시

처리 내용

- 전원 투입 시 SD 메모리 카드의 디렉토리 '/autoload'에 저장된 백업 파일의 데이터를 자동으로 컨트롤러에 전송합니다.
- 자동 전송의 대상 데이터는 '/autoload'에 저장된 자동 전송 명령 파일(AutoloadCommand.ini)에서 지정한 데이터 그룹입니다. 자동 전송 명령 파일의 자세한 내용은 '9-13-4 자동 전송 명령 파일의 사양 (9-66페이지)'을 참조하십시오.
- SD 메모리 카드의 '/autoload'에 자동 전송 명령 파일이 존재하지 않는 경우에 '/autoload'에 존재하는 전송 가능한 백업 파일의 데이터를 모두 전송합니다.
- 자동 전송 완료 후의 동작 모드는 [컨트롤러 설정]의 [전원 투입 시 모드]의 설정에 따릅니다.
- 자동 전송 실행 중인 CPU 유니트의 상태는 '기동 중'입니다.
- 자동 전송 처리 시작 전 확인에서 에러가 발생했을 때는 컨트롤러 내의 데이터는 전송 전의 데이터를 유지합니다.

- 자동 전송 처리 실행 중에 컨트롤러의 전원 OFF가 발생했을 때는 '사용자 프로그램/컨트롤러 구성·설정 전송 에러(전체 정지 이상 레벨의 컨트롤러 이상)'가 발생합니다. 이 때 컨트롤러 내의 데이터는 올바르지 않으므로 다음 방법으로 이상을 제거하십시오.
  - 자동 전송을 다시 실시
  - Sysmac Studio에서 메모리 전체 클리어, 프로젝트의 다운로드를 실시
- 자동 전송 대상 데이터 이외는 현재값을 유지합니다.
- 유지 속성이 있는 변수의 현재값이 전송 대상이 아닌 경우에 해당 변수의 현재값은 전송 전의 값을 유지합니다. 단, 변수 자체를 유지하는 조건을 충족하지 않은 경우에는 변수의 값을 초기화합니다. 변수 자체를 유지하는 조건은 다음과 같습니다.
  - 전송 전의 변수와 전송하는 변수의 변수명, 데이터 형식명, 데이터 형식의 크기가 모두 동일할 것
- NX102 CPU 유니트 및 NX1P2 CPU 유니트인 경우에 백업 파일의 CJ 유니트용 메모리 설정 내용이며, CJ 유니트용 메모리가 생성됩니다.
- NX102 CPU 유니트 및 NX1P2 CPU 유니트에서 CJ 유니트용 메모리의 현재값이 복원 대상이 아닌 경우에 HR, DM, EM 영역에 대해서는 복원 전의 값을 유지합니다. 단, 새로 생성되거나 영역이 확장된 HR, DM, EM 영역에 대해서는 초기값이 됩니다.
- 자동 전송 완료 후에 SD 메모리 카드는 마운트 상태를 지속합니다.
- 자동 전송 완료 후의 CPU 유니트의 쓰기 방지 상태는 '전원 투입 시의 쓰기 방지' 설정에 따릅니다.

조작 방법

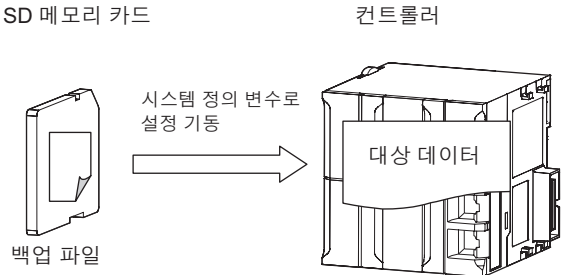
● CPU 유니트 전면 스위치를 사용한 방법

처리 단계	조작 순서
시작 지시	딥 스위치 1에서 4[*1] 를 1: OFF, 2: ON, 3: OFF, 4: OFF로 하여 컨트롤러의 전원을 투입합니다.
실행 중	전송 실행 중 SD PWR가 3s 점등, 0.5s 소등, RUN이 0.5s 점등, 0.5s 소등으로 각각 점멸합니다. SD BUSY가 점멸합니다. 점멸 패턴은 일정치 않습니다.
실행 결과	정상적으로 종료한 경우 SD PWR가 점등됩니다. 완료 후의 동작 모드는 [컨트롤러 설정]의 [전원 투입 시 모드]의 설정에 따릅니다. 이상 종료한 경우 RUN이 소등, ERR가 점등되고, 그리고 전체 정지 이상 레벨의 컨트롤러 이상이 발생합니다. SD PWR가 점등됩니다.[*2]

*1. NX701 CPU 유니트인 경우에 딥 스위치의 5에서 8을 모두 OFF로 하십시오.
*2. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR는 소등된 상태 그대로 유지됩니다.

9-5 SD 메모리 카드 프로그램 전송 기능

시스템 정의 변수 _Card1PrgTransferCmd(SD 메모리 카드 프로그램 전송 지시)에 의해 CPU 유니트에 장착한 SD 메모리 카드에 저장된 프로그램을 컨트롤러에 전송하는 기능입니다.
컨트롤러 설정에서 이 기능을 사용할지 여부 및 비밀번호를 설정할 수 있습니다.
표시기 조작 등을 통해 SD 메모리 카드 내의 백업 파일 프로그램으로 CPU 유니트를 운전하고자 하는 경우 사용합니다.



백업 기능을 사용하여 작성된 '백업 파일'을 사용합니다. 이 파일을 사전에 SD 메모리 카드의 임의의 디렉토리에 저장해둡니다.
시스템 정의 변수로 '백업 파일'을 저장한 디렉토리를 지정합니다.

파일	역할
백업 파일	백업 관련 기능으로 취급하는 컨트롤러의 데이터를 파일화한 것입니다.

이 기능의 조작 방법과 대상 디렉토리, 실행 가능한 동작 모드는 다음과 같습니다.

조작 방법	디렉토리*1	실행 가능한 동작 모드
시스템 정의 변수	시스템 정의 변수로 지정한 디렉토리	운전 모드, 프로그램 모드

*1. SD 메모리 카드 내에만 지정할 수 있습니다.

사용상의 주의 사항

- 전원 투입 시의 동작 모드 설정이 운전 모드의 프로그램을 이 기능을 사용하여 전송하면 전송 전의 상태 및 설정에 관계없이 전송 완료 후 운전 모드로 전환됩니다. 이 기능은 장치가 운전을 시작해도 문제가 없는 상태에서 실행하십시오.
- 예상치 못한 프로그램 전송을 방지하기 위해 비밀번호를 매번 입력해야 하도록 설계하십시오.
- 이 기능을 실행하면 자동으로 컨트롤러가 리셋됩니다. 컨트롤러 리셋 중의 출력은 슬레이브 및 유니트의 사양에 따릅니다. 또한, 컨트롤러 리셋 중에는 컨트롤러 외부에서 컨트롤러 내의 변수에 액세스할 수 없습니다.
- EtherCAT 슬레이브를 사용하거나 모션 제어를 실행하는 경우에 프로그램 전송 완료 후 EtherCAT 마스터 기능 모듈 및 모션 제어 기능 모듈의 이상이 발생할 수 있습니다. 이상이 발생했을 때는 프로그램 전송 완료 후 이상을 해제하십시오. 사용자 프로그램에서 이상을 해제하는 경우에는 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"의 ResetMCError 명령의 샘플 프로그램을 참조하십시오
- 프로그램 전송에 의해 전송 전에 참조했던 컨트롤러 측의 변수가 삭제된 경우에 표시기에 따라서는 삭제된 변수를 해결할 수 없어서 시스템 정의 변수에 액세스할 수 없는 경우가 있습니다. 예를 들어 NS 시리즈 표시기인 경우에 태그 비교 결과가 일치하지 않으면 태그 비교 결과 목록을 표시하고, 다른 화면으로 이동할 수 없게 됩니다. 표시기에서 사용하는 변수가 삭제되지 않았는지 사전에 확인하십시오.
- 이 기능 실행 중에 전원 OFF한 경우에 다음 번 전원 투입 시 '사용자 프로그램/컨트롤러 구성·설정 전송 에러'(이벤트 코드 10200000Hex) 등의 이상이 발생하는 경우가 있습니다.

- 컨트롤러 설정의 [시스템 정의 변수에 의한 프로그램 전송]을 [사용 안 함]으로 설정한 백업 파일을 전송한 경우에 이후에는 프로그램 전송을 실행할 수 없습니다. 반복해서 실행하는 경우에는 [사용]으로 설정한 백업 파일을 전송하십시오.
- 프로그램 전송 중에 다른 백업 기능을 실행할 수 없습니다.

처리 내용

- 시스템 정의 변수 _Card1PrgTransferCmd(SD 메모리 카드 프로그램 전송 지시)의 조작에 의해 CPU 유니트에 장착한 SD 메모리 카드에 저장된 프로그램을 컨트롤러로 전송합니다.
- 프로그램 전송 대상 백업 파일은 시스템 정의 변수로 지정한 디렉토리에 저장된 파일입니다. 대상 백업 파일은 사전에 SD 메모리 카드의 임의의 디렉토리에 저장해두어야 합니다.
- 프로그램 전송 대상 데이터는 시스템 정의 변수로 지정한 데이터 그룹입니다. 관련 시스템 정의 변수의 자세한 내용은 '관련 시스템 정의 변수(9-37페이지)'를 참조하십시오.
- 컨트롤러 설정의 [SD 메모리 카드 프로그램 전송 설정]의 [비밀번호]에 설정한 비밀번호와 시스템 정의 변수 _Card1PrgTransferCmd.Password에 설정한 비밀번호가 일치하지 않는 경우에 'SD 메모리 카드 프로그램 전송 시작 실패'가 발생합니다.
- 프로그램 전송을 시작하면 시스템 정의 변수 _Card1PrgTransferCmd.Password에 설정한 비밀번호는 초기화됩니다.
- 프로그램 전송 실행 도중에 자동으로 컨트롤러 리셋됩니다. 컨트롤러 리셋 중의 출력은 슬레이브 및 유니트의 사양에 따릅니다.
- 컨트롤러 리셋 후의 CPU 유니트의 상태는 '기동 중'입니다. 그 후, 프로그램 전송의 실행이 완료되면 CPU 유니트의 상태는 '정상 운전 중'이 되고, 프로그램 전송 기능으로 전송한 [컨트롤러 설정]의 [전원 투입 시 모드]에서 설정한 동작 모드로 동작합니다.
  단, CPU 유니트 전면 스위치가 안전 모드인 경우에는 프로그램 모드에서 동작합니다.
- 프로그램 전송 처리 시작 전 확인 또는 실행 전 확인에서 에러가 발생했을 때는 컨트롤러 내의 데이터는 전송 전의 데이터를 유지합니다.
- 프로그램 전송 처리 실행 중에 컨트롤러의 전원 OFF가 발생했을 때는 '사용자 프로그램/컨트롤러 구성·설정 전송 에러(전체 정지 이상 레벨의 컨트롤러 이상)'가 발생합니다. 이 때 컨트롤러 내의 데이터는 올바르지 않으므로 다음 방법으로 이상을 제거하십시오.
  - SD 메모리 카드 프로그램 전송을 다시 실시
  - SD 메모리 카드 자동 전송 기능 또는 SD 메모리 카드 백업 기능의 복원을 실시
  - Sysmac Studio에서 메모리 전체 클리어, 프로젝트의 다운로드를 실시
- 프로그램 전송 대상 데이터 이외는 현재값을 유지합니다.
- 유지 속성이 있는 변수의 현재값이 전송 대상이 아닌 경우에 해당 변수의 현재값은 전송 전의 값을 유지합니다. 단, 변수 자체를 유지하는 조건을 충족하지 않은 경우에는 변수의 값을 초기화합니다. 변수 자체를 유지하는 조건은 다음과 같습니다.
  - 전송 전의 변수와 전송하는 변수의 변수명, 데이터 형식명, 데이터 형식의 크기가 모두 동일할 것
- NX102 CPU 유니트 및 NX1P2 CPU 유니트인 경우에 백업 파일의 CJ 유니트용 메모리 설정 내용이며, CJ 유니트용 메모리가 생성됩니다.
- NX102 CPU 유니트 및 NX1P2 CPU 유니트에서 CJ 유니트용 메모리의 현재값이 복원 대상이 아닌 경우에 HR, DM, EM 영역에 대해서는 복원 전의 값을 유지합니다. 단, HR, DM, EM 영역이 새로 생성되었거나 영역이 확장된 경우에 이들의 영역에 대해서는 초기값이 됩니다.
- 프로그램 전송 중에 SD 메모리 카드 전원 공급 정지 버튼을 눌러도 전원 공급을 계속합니다.
- 프로그램 전송 완료 후, SD 메모리 카드는 마운트 상태를 지속합니다.
- 프로그램 전송 완료 후의 CPU 유니트의 쓰기 방지 상태는 '전원 투입 시의 쓰기 방지' 설정에 따릅니다.

조작 방법

● 시스템 정의 변수 _Card1PrgTransferCmd(SD 메모리 카드 프로그램 전송 지시)

처리 단계	조작 순서
시작 전의 준비	SD 메모리 카드 프로그램 전송 기능을 사용하는 경우에는 컨트롤러 설정의 [시스템 정의 변수에 의한 프로그램 전송]을 [사용]으로 설정해둡니다.*1
시작 지시	시스템 정의 변수 _Card1PrgTransferCmd.DirName(디렉토리명)에 백업 파일이 저장된 디렉토리명을 저장합니다. (예) 디렉토리 dirA 아래의 서브 디렉토리 dirB를 지정: "dirA/dirB" 시스템 정의 변수 _Card1PrgTransferCmd.Password(비밀번호)에 비밀번호를 저장합니다.*2 내장 EtherNet/IP 포트 설정을 전송 대상으로 할 때는 시스템 정의 변수 _Card1PrgTransferCmd.TargetIPAdr(IP 주소의 전송 플래그)를 TRUE로 합니다.*3 변수의 현재값(유지 속성만 해당)를 전송 대상으로 할 때는 시스템 정의 변수 _Card1PrgTransferCmd.TargetVariable(변수의 현재값(유지 속성만 해당)의 전송 플래그)를 TRUE로 합니다. CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때는 시스템 정의 변수 _Card1PrgTransferCmd.TargetMemory(CJ 유니트용 메모리의 현재값(유지 메모리만 해당)의 전송 플래그)를 TRUE로 합니다. 시스템 정의 변수 _Card1PrgTransferCmd.Exec(프로그램 전송 기동 플래그)를 FALSE→TRUE로 하면 실행이 시작됩니다.
실행 중	시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)가 TRUE가 됩니다. 시스템 정의 변수 _Card1PrgTransferSta.Active(실행 중 플래그)가 TRUE가 됩니다. 프로그램 전송 실행 중 SD PWR가 3s 점등, 0.5s 소등으로 점멸합니다. SD BUSY가 점멸(점멸 패턴은 일정치 않음)합니다. RUN이 0.5s 점등, 0.5s 소등으로 점멸합니다. 프로그램 전송 실행 도중에 자동으로 컨트롤러 리셋됩니다.
실행 결과	정상적으로 종료한 경우 SD PWR가 점등됩니다. 완료 후의 동작 모드는 [컨트롤러 설정]의 [전원 투입 시 모드]의 설정에 따릅니다. 시스템 정의 변수 _Card1PrgTransferSta.Done(완료 플래그)이 TRUE가 됩니다. 시작 전 확인에서 이상 종료한 경우 SD PWR가 점등됩니다.*4 RUN과 ERR는 시작 지시 전의 상태입니다. 시스템 정의 변수 _Card1PrgTransferSta.Err(에러 플래그)가 TRUE가 됩니다. 시작 전 확인에서 이상 종료한 경우에는 컨트롤러 리셋이 발생하지 않습니다. 실행 전 확인 또는 실행 중에 이상 종료한 경우 SD PWR가 점등됩니다. *4 RUN이 소등, ERR가 점등되고, 그리고 전체 정지 이상 레벨의 컨트롤러 이상이 발생합니다. 시스템 정의 변수 _Card1PrgTransferSta.Err(에러 플래그)가 TRUE가 됩니다.

*1. 컨트롤러 설정의 [시스템 정의 변수에 의한 프로그램 전송]을 [사용 안 함]으로 설정한 백업 파일을 전송한 경우에 이후에는 프로그램 전송을 실행할 수 없습니다. 반복해서 실행하는 경우에는 [사용]으로 설정한 백업 파일을 전송하십시오.

*2. SD 메모리 카드 프로그램 전송을 시작하면 비밀번호가 초기화됩니다. 시작할 때마다 비밀번호를 저장하십시오. 컨트롤러 설정의 [SD 메모리 카드 프로그램 전송 설정]의 [비밀번호]에 비밀번호가 설정되지 않은 경우에 시스템 정의 변수 _Card1PrgTransferCmd.Password의 값이 초기값이면 프로그램 전송을 시작합니다. 초기값 이외이면 전송을 시작하지 않습니다.

*3. IP 주소에는 설정 유형, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.

*4. SD 메모리 카드를 장착하지 않은 경우에는 SD PWR는 소등된 상태 그대로 유지됩니다.

관련 시스템 정의 변수

관련 시스템 정의 변수는 다음과 같습니다. 시스템 정의 변수의 자세한 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'을 참조하십시오.

변수명 / 멤버명	이름	기능	데이터 형식	R/W
_Card1PrgTransferCmd	SD 메모리 카드 프로그램 전송 지시		_sPRGTRANSFER_CMD	RW
Exec	프로그램 전송 기동 플래그	SD 메모리 카드 프로그램 전송 기능에 의해 SD 메모리 카드의 백업 파일 데이터를 컨트롤러로 전송할 때 TRUE로 합니다.	BOOL	RW
DirName	디렉토리명	백업 대상인 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING (64)	RW
Password	비밀번호	프로그램 전송 시작 시에 비교되는 비밀번호를 지정합니다. SD 메모리 카드 프로그램 전송을 시작할 때마다 비밀번호가 초기화됩니다.	STRING(33)	RW
TargetUserProgram	사용자 프로그램·설정 전송 플래그 *1	프로그램 전송에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. SD 메모리 카드 프로그램 전송 기능에서는 반드시 TRUE로 하십시오.	BOOL	RW
TargetIPAdr	IP 주소 전송 플래그	프로그램 전송에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 유형, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.	BOOL	RW
TargetVariable	변수의 현재값(유지 속성만 해당) 전송 플래그	프로그램 전송에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW
TargetMemory	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그	프로그램 전송에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	RW
_Card1PrgTransferSta	SD 메모리 카드 프로그램 전송 Status		_sPRGTRANSFER_STA	R
Done	완료 플래그	프로그램 전송 실행 완료일 때 TRUE가 됩니다.	BOOL	R
Active	실행 중 플래그	프로그램 전송 실행 중일 때 TRUE가 됩니다.	BOOL	R
Err	에러 플래그	프로그램 전송이 이상 종료됐을 때 TRUE가 됩니다.	BOOL	R

*1. 사용자 프로그램·설정 데이터 그룹의 데이터가 설정 대상입니다. 단, 내장 EtherNet/IP 포트의 IP 주소는 IP 주소의 전송 플래그로 설정합니다.

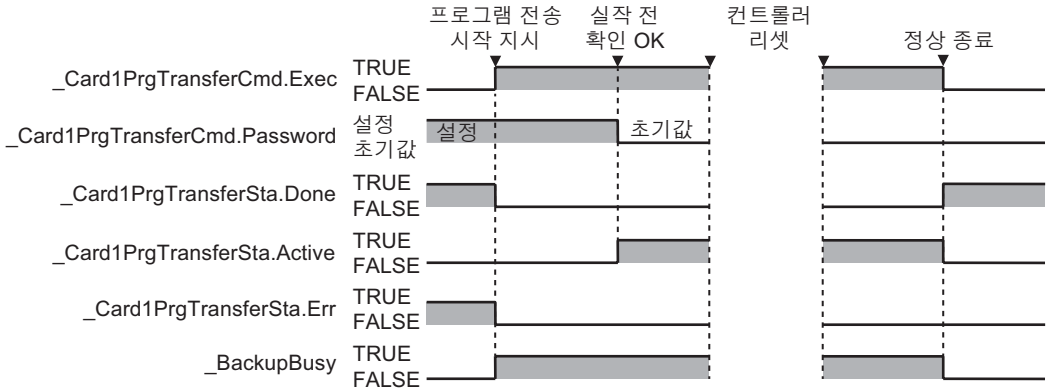
참고

- NJ/NX 시리즈 컨트롤러와 NA 시리즈 표시기를 연결하여 사용하는 경우의 변수 매핑에 대해서는 "프로그래머블 터미널 NA 시리즈 사용 설명서 소프트웨어편(SBSA-546)"을 참조하십시오.
- NJ/NX 시리즈 컨트롤러와 NS 시리즈 표시기를 연결하여 사용하는 경우 CX-Designer의 변수 테이블에 이러한 시스템 정의 변수를 등록하는 방법에 대해서는 'A-11 CX-Designer의 변수 테이블 등록 방법(A-198페이지)'을 참조하십시오.

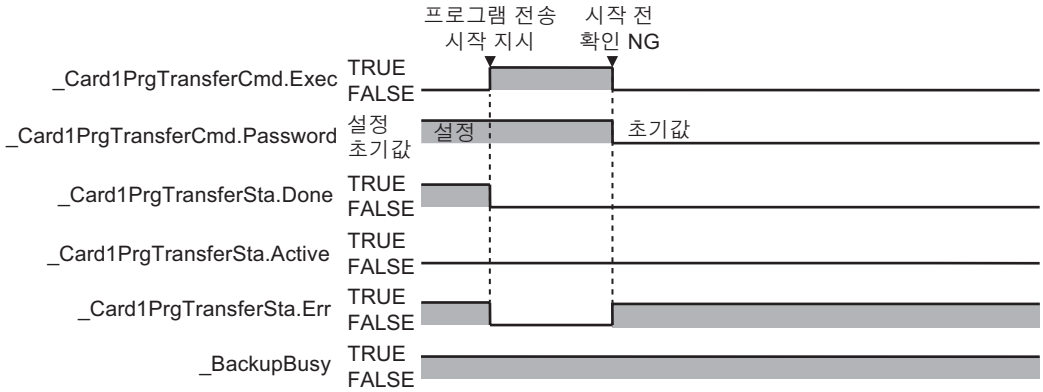
타이밍 차트

시스템 정의 변수를 사용하는 방법으로 SD 메모리 카드 전송을 실시하는 경우의 시스템 정의 변수의 동작은 다음과 같습니다.
그림에 나오는 '시작 전 확인'은 프로그램 전송이 시작되기 전에 비밀번호가 일치하는지 여부 등을 확인하는 처리입니다. _Card1PrgTransferSta.Active(실행 중 플래그)의 값이 TRUE가 되는 것은 시작 전 확인 결과가 OK가 되었을 때입니다.

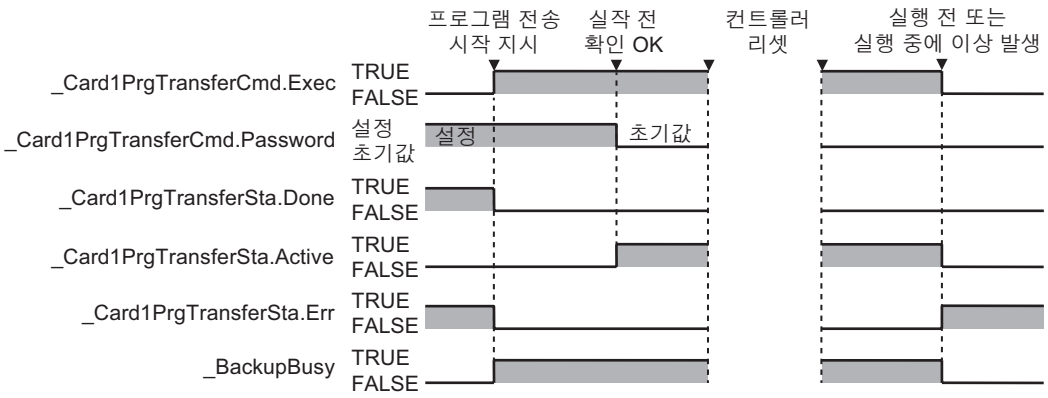
- **정상 동작**



- **다른 백업 기능 실행 중이라서 프로그램 전송을 시작할 수 없었던 경우**

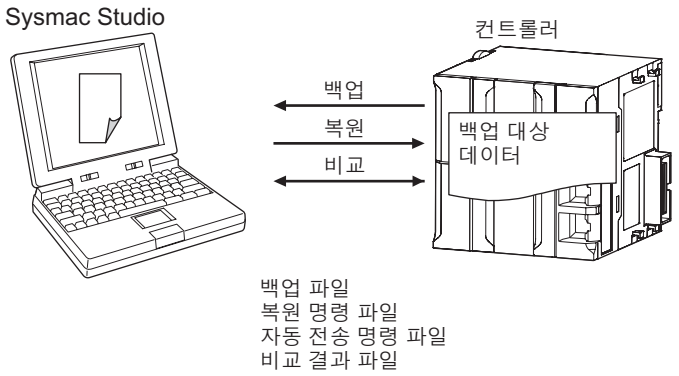


● **정상적으로 프로그램 전송을 시작한 후에 어떤 원인에 의해 프로그램 전송에 실패한 경우**



9-6 Sysmac Studio 컨트롤러 백업 기능

Sysmac Studio를 사용하여 PC에서 컨트롤러의 데이터를 백업, 복원, 비교하는 기능입니다.



백업을 실행하면 '백업 파일', '복원 명령 파일', '자동 전송 명령 파일'이 PC의 지정 디렉토리에 작성됩니다. 이들 백업 관련 파일의 역할은 다음과 같습니다.

파일	역할			
	내용	백업	복원	비교
백업 파일	백업 관련 기능으로 취급하는 컨트롤러의 데이터를 파일화한 것입니다.	작성	참조	참조
복원 명령 파일	복원을 실행할 때, 전송하는 대상인 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다.	작성	참조	참조
자동 전송 명령 파일	SD 메모리 카드 자동 전송을 실행할 때, 전송하는 대상의 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다.	작성	아무것도 하지 않음	아무것도 하지 않음

이 기능의 실행 가능한 동작 모드는 다음과 같습니다.

처리	실행 가능한 동작 모드
백업	운전 모드, 프로그램 모드
복원	프로그램 모드
비교	운전 모드, 프로그램 모드

참고

백업 또는 비교 실행 중에 CPU 유니트의 동작 모드 변경이 가능합니다.
단, CPU 유니트의 메모리가 고장인 경우 등과 같이 정상적으로 백업 또는 비교를 실행할 수 없을 때는 에러가 됩니다.

9-6-1 백업(컨트롤러에서 PC)

컨트롤러의 데이터를 PC의 지정 디렉토리에 저장합니다.

처리 내용

- 백업 데이터 내의 유니트·슬레이브 설정은 연결된 모든 EtherCAT 슬레이브 및 연결된 모든 CJ 유니트만 선택할 수 있습니다.
- 각 데이터 그룹의 백업 조건은 다음 표와 같습니다.

데이터 그룹	백업 조건
사용자 프로그램·설정	'CPU 유니트' 선택 시
내장 EtherNet/IP 포트의 IP 주소*1	'CPU 유니트' 선택 시
변수의 현재값	'CPU 유니트' 선택 시
CJ 유니트용 메모리의 현재값	'CPU 유니트' 선택 시
이벤트 로그	'CPU 유니트' 선택 시
유니트·슬레이브 설정	'CPU 유니트 상의 NX 유니트' 선택 시*2, 'CJ 유니트' 선택 시*3, 'EtherCAT 슬레이브' 선택 시
앱솔루트 엔코더 원점 위치 오프셋	'CPU 유니트' 선택 시

*1. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합인 경우에 대상이 됩니다.
*2. CPU 유니트 상의 NX 유니트를 선택할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*3. CJ 유니트를 선택할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

- 백업을 실행하면 '백업 파일', '복원 명령 파일', '자동 전송 명령 파일'이 PC의 지정 디렉토리에 작성됩니다.
- 지정 디렉토리에 기존 백업 관련 파일이 있는 경우에는 덮어씁니다.
- 지정 디렉토리에 백업 관련 파일을 쓰는 도중에 에러 발생한 경우에는 기존의 백업 관련 파일이 삭제됩니다. 그리고 새 백업 관련 파일도 작성되지 않습니다.
- 새 백업 관련 파일 작성 전에 에러가 발생한 경우에는 기존의 파일이 유지되고, 새 파일은 작성되지 않습니다.
- 실행 중에는 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.

조작 방법

1 **Sysmac Studio의 [툴]–[백업]–[컨트롤러 백업]을 실행합니다.**

2 **백업 파일, 복원 명령 파일, 자동 전송 명령 파일을 작성할 디렉토리를 지정합니다.**

3 **백업 확인 화면에서 [실행] 버튼을 누릅니다.**
백업이 실행되고 백업 파일, 복원 명령 파일, 자동 전송 명령 파일이 작성됩니다.

9-6-2 복원(PC에서 컨트롤러)

PC의 지정 디렉토리 내의 백업 파일 데이터를 컨트롤러에 복원합니다.
프로그램 모드에서만 실행할 수 있습니다.

처리 내용

- PC의 지정 디렉토리 내의 백업 파일 데이터를 컨트롤러에 복원합니다.
- 복원 대상 데이터 그룹은 Sysmac Studio에서 선택할 수 있습니다. 복원되기 위한 조건은 다음 표와 같습니다.

데이터 그룹	복원 조건
사용자 프로그램·설정	'CPU 유니트' 선택 시
내장 EtherNet/IP 포트의 IP 주소*1	'내장 EtherNet/IP 포트의 IP 주소' 선택 시
변수의 현재값	'유지 속성 변수의 현재값' 선택 시
CJ 유니트용 메모리의 현재값	'유지 속성 CJ 유니트용 메모리의 현재값' 선택 시*2
유니트·슬레이브 설정	'CPU 유니트 상의 NX 유니트' 선택 시*3, 'CJ 유니트' 선택 시*2, 'EtherCAT 슬레이브' 선택 시
앱솔루트 엔코더 원점 위치 오프셋	'앱솔루트 엔코더 원점 위치 오프셋' 선택 시

*1. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합인 경우에 대상이 됩니다.
*2. 유지 속성 CJ 유니트용 메모리의 현재값을 선택할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.
*3. CPU 유니트 상의 NX 유니트를 선택할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.

- 복원 처리 시작 전 확인에서 에러가 발생한 경우에는 컨트롤러 내의 데이터는 복원 전의 데이터를 유지합니다.
- 복원 처리 실행 중에 컨트롤러의 전원 OFF가 발생한 경우에는 '사용자 프로그램/컨트롤러 구성·설정 전송 에러(전체 정지 이상 레벨의 컨트롤러 이상)'가 발생합니다. 이 때 컨트롤러 내의 데이터는 올바르지 않으므로 다음 방법으로 이상을 제거하십시오.
  - 복원을 다시 실시
  - Sysmac Studio에서 메모리 전체 클리어, 프로젝트의 다운로드를 실시
- 유지 속성이 있는 변수의 현재값이 복원 대상이 아닌 경우에 해당 변수의 현재값은 복원 전의 값을 유지합니다. 단, 변수 자체를 유지하는 조건을 충족하지 않은 경우에는 변수의 값을 초기화합니다. 변수 자체를 유지하는 조건은 다음과 같습니다.
  - 복원 전의 변수와 복원하는 변수의 변수명, 데이터 형식명, 데이터 형식의 크기가 모두 동일할 것
- NX1P2 CPU 유니트에서 백업 파일의 옵션 보드 구성과 복원 대상인 실제 옵션 보드 구성이 다른 경우에도 복원할 수 있습니다. 단, 옵션 보드는 동작하지 않습니다. 자세한 내용은 "NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)"를 참조하십시오.
- NX102 CPU 유니트 및 NX1P2 CPU 유니트인 경우에 백업 파일의 CJ 유니트용 메모리 설정 내용이며, CJ 유니트용 메모리가 생성됩니다.
- NX102 CPU 유니트 및 NX1P2 CPU 유니트에서 CJ 유니트용 메모리의 현재값이 복원 대상이 아닌 경우에 HR, DM, EM 영역에 대해서는 복원 전의 값을 유지합니다. 단, HR, DM, EM 영역이 새로 생성되었거나 영역이 확장된 경우에 이들의 영역에 대해서는 초기값이 됩니다.
- 복원 실행 완료 후에는 모든 EtherCAT 슬레이브의 전원을 재투입하십시오.

조작 방법

1 **Sysmac Studio의 [툴]–[백업]–[컨트롤러 복원]을 실행합니다.**

2 **백업 파일, 복원 명령 파일이 저장된 디렉토리를 지정합니다.**

3 **복원 확인 화면에서 [실행] 버튼을 누릅니다.**
복원이 실행됩니다.

9-6-3 비교(컨트롤러와 PC 간)

컨트롤러의 데이터와 PC의 지정 디렉토리 내의 백업 파일 데이터를 비교합니다.

처리 내용

- 컨트롤러의 데이터와 PC의 지정 디렉토리 내의 백업 파일 데이터를 비교합니다. 비교 대상 데이터 그룹은 Sysmac Studio에서 선택할 수 있습니다. 비교되기 위한 조건은 다음 표와 같습니다. 일괄 지정을 하면 이러한 데이터가 모두 비교됩니다.

데이터 그룹	비교 조건
사용자 프로그램·설정	'CPU 유니트' 선택 시
내장 EtherNet/IP 포트의 IP 주소*1	'내장 EtherNet/IP 포트의 IP 주소' 선택 시
유니트·슬레이브 설정	'CPU 유니트 상의 NX 유니트' 선택 시*2, 'CJ 유니트' 선택 시*3, 'EtherCAT 슬레이브' 선택 시

*1. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합인 경우에 대상이 됩니다.
*2. CPU 유니트 상의 NX 유니트를 선택할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*3. CJ 유니트를 선택할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

- 비교 결과는 Sysmac Studio의 화면에 표시됩니다.
- 실행 중에는 시스템 정의 변수 _BackupBusy(백업 관련 기능 실행 중 플래그)의 값이 TRUE가 됩니다.

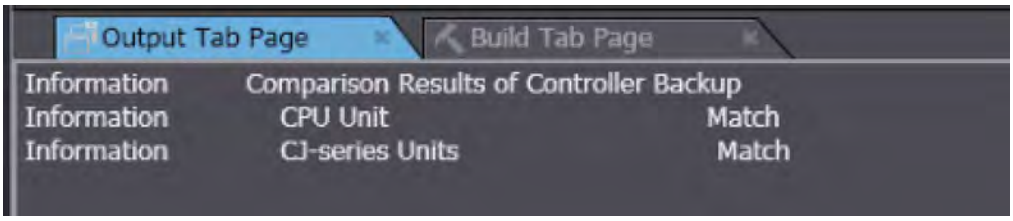
조작 방법

1 **Sysmac Studio의 [툴]–[백업]–[백업 파일의 비교]를 실행합니다.**

2 **백업 파일이 저장된 디렉토리를 지정합니다.**

3 **비교 확인 화면에서 [실행] 버튼을 누릅니다.**

비교가 실행되고, 백업 파일이 저장된 디렉토리에 비교 결과 파일이 작성됩니다. 또한, 출력 창에 비교 결과가 표시됩니다.



9-7 Sysmac Studio 백업 파일 가져오기/내보내기 기능

컨트롤러를 사용하지 않고 Sysmac Studio의 프로젝트에서 PC의 지정 디렉토리에 백업 파일을 작성 및 읽어오는 기능입니다.
대상 데이터는 다음과 같습니다.

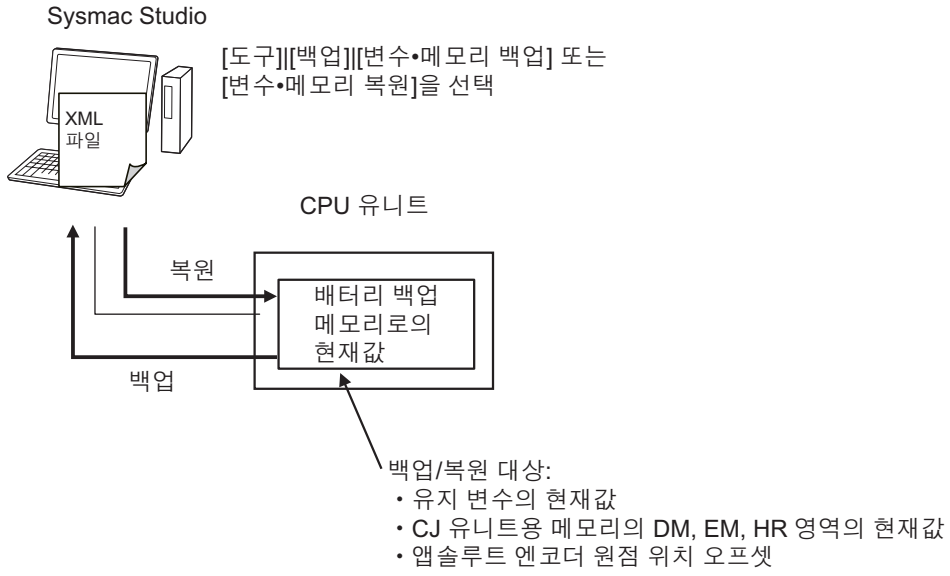
기능		데이터 그룹					
		사용자 프로그램·설정	내장 EtherNet/IP 포트의 IP 주소*2	변수의 현재값	CJ유니트용 메모리의 현재값*1	유니트·슬레이브 설정	앱솔루트 엔코더 원점 위치 오프셋
Sysmac Studio 백업 파일 가져오기/내보내기 기능	백업 파일 내보내기	○*3	○	×	×	×	×
	백업 파일 가져오기	○*4	○	×	×	○	×

*1. CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.
*2. CPU 유니트 Ver.1.14 이상과 Sysmac Studio Ver.1.18 이상의 조합으로, 내장 EtherNet/IP 포트 설정의 IP 주소는 하나의 데이터 그룹으로 사용할 수 있습니다.
상기 이외의 조합에서는 IP 주소는 사용자 프로그램·설정에 포함됩니다.
*3. 다음과 같은 데이터는 처리 대상에서 제외됩니다.
컨트롤러 이름 중에 내장 EtherNet/IP 포트명
컨트롤러 설정 중의 내장 EtherNet/IP용 태그 데이터 링크 설정
유니트 구성·유니트 설정 중의 CPU 고기능 유니트 영역
조작 권한 인증
데이터 트레이스 설정
*4. 다음과 같은 데이터는 처리 대상에서 제외됩니다. Sysmac Studio의 버전에 따라 대상 외의 데이터가 다릅니다.
Sysmac Studio의 버전이 Ver.1.16 이상인 경우
컨트롤러 이름 중에 내장 EtherNet/IP 포트명
유니트 구성·유니트 설정 중의 CPU 고기능 유니트 영역
조작 권한 인증
데이터 트레이스 설정
Sysmac Studio 버전이 Ver.1.15 이하인 경우
컨트롤러 이름 중에 내장 EtherNet/IP 포트명
컨트롤러 설정 중의 내장 EtherNet/IP용 태그 데이터 링크 설정
유니트 구성·유니트 설정 중의 CPU 고기능 유니트 영역
조작 권한 인증
데이터 트레이스 설정

이 기능의 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

9-8 Sysmac Studio 변수/메모리 백업 기능

CPU 유니트 내의 배터리 백업 메모리의 현재값을 PC에 파일(XML 형식)로 백업(읽기) 또한 PC에서 CPU 유니트에 복원(쓰기)할 수 있습니다.
여기에서는 Sysmac Studio 변수/메모리 백업 기능의 대상 데이터, 조작 방법 및 CPU 유니트 형식 간의 호환성에 대해 설명합니다.



9-8-1 Sysmac Studio 변수/메모리 백업 기능의 대상 데이터

Sysmac Studio 변수/메모리 백업 기능은 다음과 같은 데이터를 대상으로 합니다.

- 유지 속성 변수의 현재값
- CJ 유니트용 메모리의 DM, EM, HR 영역의 현재값
- 앱솔루트 엔코더 원점 위치 오프셋

(주의) CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트 뿐입니다.

버전 관련 정보

CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합인 경우에 특정 변수만 선택하여 유지 속성의 변수의 현재값을 백업/복원할 수 있습니다.

9-8-2 Sysmac Studio 변수/메모리 백업 기능의 조작 방법

Sysmac Studio에서의 조작 방법은 다음과 같습니다.
온라인 연결 상태에서 [툴] 메뉴의 [백업]–[변수·메모리 백업] 또는 [변수·메모리 복원]을 선택합니다.
자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

9-8-3 CPU 유니트 형식 간의 호환성

Sysmac Studio 변수/메모리 백업 기능은 백업원과 복원 대상인 CPU 유니트의 형식이 달라도 복원할 수 있습니다.



참고

데이터베이스 연결 CPU 유니트, SECS/GEM 탑재CPU 유니트, NJ Robotics CPU 유니트 및 NC 통합 컨트롤러 등과는 호환성이 없습니다.
이러한 CPU 유니트의 호환성에 대해서는 각 CPU 유니트의 설명서를 참조하십시오.



버전 관련 정보

백업원과 복원 대상인 CPU 유니트의 형식이 달라도 복원이 가능한 것은 CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합인 경우입니다. 그 이외인 경우의 호환성은 다음과 같습니다.

(○: 호환성 있음, ×: 호환성 없음)

백업원인 CPU 유니트 형식	복원 대상인 CPU 유니트 형식	
	NJ501-1500 NJ501-1400 NJ501-1300	NJ301-1200 NJ301-1100
NJ501-1500, NJ501-1400, NJ501-1300	○	×
NJ301-1200, NJ301-1100	×	○

단, 백업/복원의 대상 데이터에 따라 다음과 같은 주의가 필요합니다.

유지 속성 변수의 현재값

유지 속성 변수의 현재값에 대해서는 특별히 주의할 필요는 없습니다. 백업원과 복원 대상인 CPU 유니트의 형식에 관계없이 문제없이 복원할 수 있습니다.

CJ 유니트용 메모리의 DM, EM, HR 영역의 현재값

CJ 유니트용 메모리에 대해서는 백업 파일 내에 저장된 영역과 복원 대상인 실제 CPU 유니트에 존재하는 영역이 다른 경우에 중복된 영역만 복원됩니다.

CJ 유니트용 메모리의 DM, EM, HR 영역의 현재값에 대해 백업원과 복원 대상인 CPU 유니트의 형식이 다른 경우의 주의 사항을 NJ 시리즈 CPU 유니트를 예로 들어 설명합니다.

백업원인 CPU 유니트 형식	복원 대상인 CPU 유니트 형식	주의 사항
NJ501-1300, NJ501-1400, NJ501-1500	NJ301-1200, NJ301-1100, NJ101-1000, NJ101-9000	확장 데이터 메모리(EM)는 백업 파일 중 E0~E3만 복원됩니다. 백업 파일 중 E4~E18의 값은 무시됩니다.
NJ301-1200, NJ301-1100, NJ101-1000, NJ101-9000	NJ501-1300, NJ501-1400, NJ501-1500	확장 데이터 메모리(EM)는 백업 파일 중 E0~E3만 복원됩니다. CJ 유니트의 E4~E18은 복원하기 전의 값이 유지됩니다.

백업 파일 내에 저장된 영역과 복원 대상인 실제 CPU 유니트에 존재하는 영역이 다른 경우의 예시를 다음에서 설명합니다.

- **백업 파일 내의 영역이 복원 대상 영역보다 큰 경우**

백업 시 DM 영역의 크기=8CH		복원 대상인 DM 영역의 크기=4CH	
주소	값	주소	값
DM0	0x0001	DM0	0x0001
DM1	0x0002	DM1	0x0002
DM2	0x0003	DM2	0x0003
DM3	0x0004	DM3	0x0004
DM4	0x0005	–	–
DM5	0x0006	–	–
DM6	0x0007	–	–
DM7	0x0008	–	–

- **백업 파일 내의 영역이 복원 대상 영역보다 작은 경우**

백업 시 DM 영역의 크기=8CH		복원 대상인 DM 영역의 크기=4CH	
주소	값	주소	값
DM0	0x0001	DM0	0x0001
DM1	0x0002	DM1	0x0002
DM2	0x0003	DM2	0x0003
DM3	0x0004	DM3	0x0004
–	–	DM4	변화 없음
–	–	DM5	변화 없음
–	–	DM6	변화 없음
–	–	DM7	변화 없음

앱솔루트 엔코더 원점 위치 오프셋

앱솔루트 엔코더 원점 위치 오프셋에 대해 백업원과 복원 대상인 CPU 유니트의 형식이 다른 경우의 주의 사항을 NJ 시리즈 CPU 유니트를 예로 들어 설명합니다.

백업원인 CPU 유니트 형식	복원 대상인 CPU 유니트 형식	주의 사항
NJ501-1300, NJ501-1400, NJ501-1500	NJ301-1200, NJ301-1100, NJ101-1000	백업원의 유효 축 수에 관계없이 백업 파일 중 선두 축에서 복원 대상인 유효 축 수 분량이 복원됩니다. 백업 파일 중의 여분의 데이터는 무시됩니다.
NJ301-1200, NJ301-1100, NJ101-1000	NJ501-1300, NJ501-1400, NJ501-1500	백업원의 유효 축 수에 관계없이 백업 파일 중 선두 축에서 복원 대상인 유효 축 수 분량이 복원됩니다. 백업 파일의 데이터보다 복원 대상인 유효 축이 많은 경우에 여분의 축의 데이터는 복원하기 전의 값이 유지됩니다.

9-9 EtherCAT 슬레이브 연결 시의 백업 기능

EtherCAT 슬레이브는 SD 메모리 카드 백업 기능, Sysmac Studio 컨트롤러 백업 기능 및 Sysmac Studio 백업 가져오기 기능의 대상입니다.
여기에서는 EtherCAT 슬레이브를 연결한 경우의 백업 대상 데이터, 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부, 복원 실행의 조건, EtherCAT 슬레이브의 모델별 주의 사항에 대해 설명합니다.

참고

백업 기능의 대상이 EtherCAT 슬레이브 터미널인 경우의 자세한 내용은 '9-10 EtherCAT 슬레이브 터미널 연결 시의 백업 기능(9-55페이지)'을 참조하십시오.

9-9-1 EtherCAT 슬레이브의 백업 대상 데이터

EtherCAT 슬레이브의 백업 대상이 되는 데이터는 다음과 같습니다.

설정	백업 대상 데이터
EtherCAT 마스터 설정	형식명, 제품 이름, 슬레이브 수, PDO 통신 주기, FAIL-SOFT 동작 설정, 전체 슬레이브 가입 대기 시간, PDO 통신 연속 타임아웃 검출 횟수, 개정 번호 확인 기준, 시리얼 번호 확인 기준
EtherCAT 슬레이브 설정	장치명, 형식명, 제품 이름, 개정 번호, 노드 주소, 유효/무효 설정, 시리얼 번호, PDO 매핑 설정, DC 유효, 참조 클록 기능, 초기화 파라메터 설정

9-9-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부

컨트롤러의 각 상태에서의 EtherCAT 슬레이브의 백업, 복원, 비교의 실행 가능 여부의 사양은 다음과 같습니다.

컨트롤러 상태		실행 가능 여부		
		백업	복원	비교
링크 오프 중		불가능*1	불가능*2	가능*3
잘못된 마스터 상태*4		불가능*1	불가능*2	가능*3
구성 정보와 네트워크 구성 불일치*5		불가능*1	불가능*2	가능*3
백업 시와 네트워크 구성 불일치		가능	불가능*2	가능*3
네트워크 구성에 무효 설정의 슬레이브가 존재	실제 구성에 슬레이브 존재	가능*6	가능*6	가능*6
	실제 구성에 슬레이브 부재	가능	가능	가능
네트워크 구성에 '이탈' 지시에 따라 이탈 상태가 된 슬레이브가 존재	실제 구성에 슬레이브 존재	불가능*1	가능 이탈 상태의 슬레이브도 대상	가능 이탈 상태의 슬레이브도 대상
	실제 구성에 슬레이브 부재	불가능*1	불가능*2	가능*3
슬레이브 초기화 이상		불가능*1	불가능*2	가능*3

*1. 이벤트 로그에 백업 실행 실패(EtherCAT 슬레이브)가 기록됩니다.
*2. 이벤트 로그에 복원 실행 실패(EtherCAT 슬레이브)가 기록됩니다.
*3. 비교 결과가 불일치가 됩니다.
*4. '슬레이브 노드 주소 중복', '네트워크 구성 정보 이상', '네트워크 구성 이상', '슬레이브 초기화 이상', 'FAIL-SOFT 동작 설정을 “정지 설정”으로 설정한 경우의 네트워크 구성 비교 이상', '링크 오프 이상'을 말합니다.
*5. '백업 시와 네트워크 구성이 완전히 일치하지 않는 경우(분기 슬레이브의 연결 포트가 다른 것은 불일치로 간주)', '네트워크 구성 정보와 네트워크 구성이 일치하지 않는 경우(분기 슬레이브의 연결 포트가 다른 것은 불일치로 간주)'를 말합니다.
*6. CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합에서는 무효 설정 슬레이브도 대상이 됩니다. 그 이외는 무효 설정 슬레이브는 대상에서 제외됩니다.

9-9-3 EtherCAT 슬레이브의 복원 실행 조건

EtherCAT 슬레이브의 백업 데이터를 복원하기 위해서는 다음 조건을 충족해야 합니다.

- 백업 파일 내에 EtherCAT 슬레이브의 데이터가 존재
- 네트워크 구성 정보와 복원 대상인 실제 네트워크 구성이 일치
- EtherCAT 슬레이브에 사전에 설정된 개정 번호 값이 일치 단, 일치 조건은 백업 파일 내에 저장된 '개정 번호 확인 기준'에 따릅니다. 또한, '개정 번호 확인 기준'이 '확인 안 함'으로 설정되어 있어도 '설정 개정 번호'>'실제 기기 개정 번호'인 경우에는 복원할 수 없습니다. 또한, 개정 번호 값은 사용자가 변경할 수 없습니다.
- 백업 파일 내에 저장된 '시리얼 번호 비교 유무'가 '비교'로 설정된 경우에 시리얼 번호가 일치
- 하드웨어 스위치로 노드 주소가 설정된 경우에 노드 주소가 일치

사용상의 주의 사항

- 복원 실행 완료 후에는 모든 EtherCAT 슬레이브의 전원을 재투입하십시오.
- 복원 실행 완료 후에는 전체 슬레이브가 이탈 상태가 됩니다. 대상 슬레이브의 이탈 상태를 해제하려면 대상 슬레이브에 대한 재가입 지시가 필요합니다.
- 백업 파일에 저장된 '시리얼 번호 비교의 유무'가 '비교'로 설정된 경우에 EtherCAT 슬레이브의 하드웨어 교환을 실행하면 복원할 수 없게 됩니다. 그 경우에는 Sysmac Studio에서 네트워크 구성을 변경하고, 다운로드 및 슬레이브 파라메터의 전송을 실행해서 원상 복구하십시오. 또한, 노드 주소를 하드웨어 스위치에서 설정하는 경우에는 백업 시와 동일하게 설정하십시오.

9-9-4 백업 기능의 대상 EtherCAT 슬레이브

백업 기능의 대상이 되는 EtherCAT 슬레이브는 다음과 같습니다. 주의 사항을 준수하여 사용하십시오.

EtherCAT 슬레이브의 모델	주의 사항
NX 시리즈 EtherCAT 커플러 유니트 NX-ECC	EtherCAT 슬레이브 터미널의 세이프티 컨트롤 유니트는 백업, 복원, 비교 대상에서 제외됩니다. 세이프티 컨트롤 유니트의 설정 데이터 가져오기/내보내기 기능은 📖"NX 시리즈 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)"를 참조하십시오.
AC 서보 드라이버 R88D-1SN□□□-ECT	*1
AC 서보 드라이버(회전형) R88D-KN□□□-ECT	*1*2
AC 서보 드라이버(리니어형) R88D-KN□□□-ECT-L	*1*2
인버터 3G3AX-MX2-ECT 3G3AX-RX-ECT	• Ver.1.11 이상의 CPU 유니트인 경우에 인버터의 파라메터는 복원 대상에서 제외됩니다. 인버터에 대해 '복원 실행 실패(EtherCAT 슬레이브)'(이벤트 코드 10300000Hex) 이벤트가 발생하지는 않습니다. • 뒤에 설명하는 📖'인버터 3G3AX-MX2-ECT, 3G3AX-RX-ECT의 파라메터 기입 방법(9-54페이지)'으로 인버터에 Sysmac Studio에서의 파라메터를 쓰십시오. 파라메터를 쓰지 않고 비교를 실행하면 인버터에 대해서는 EtherCAT 슬레이브 비교 결과 파일 내에서 'Result=Not matched'가 될 가능성이 있습니다. • Ver.1.10 이하인 CPU 유니트인 경우에 인버터의 제약에 의해 복원을 실행할 수 없는 경우가 있습니다. 복원 실행 시 '복원 실행 실패(EtherCAT 슬레이브)'(이벤트 코드 10300000Hex) 이벤트가 발생한 경우에 뒤에 설명하는 📖'인버터 3G3AX-MX2-ECT, 3G3AX-RX-ECT의 파라메터 기입 방법(9-54페이지)'으로 인버터에 Sysmac Studio에서의 파라메터를 쓰십시오. 또한 인버터의 복원에 실패한 경우에도 인버터 이외의 유니트·슬레이브 설정, 사용자 프로그램·설정 및 변수의 현재값 등 모든 대상 데이터는 복원이 완료됩니다.
시각 센서 FH-3□□□□ FH-1□□□□	센서 설정 데이터(신(Scene) 데이터, 시스템 데이터 등)는 백업, 복원, 비교 대상에서 제외됩니다. 센서의 설정 데이터를 외부 파일에 전송하거나 센서에 전송하려면 Sysmac Studio의 구성·설정의 편집 화면에서 [툴]–[센서 데이터]–[파일에 저장] 또는 [파일에서 읽기]를 실행하십시오. 자세한 내용은 📖"화상 처리 시스템 FH/FZ5시리즈 사용 설명서(SDNB-712)"를 참조하십시오.
시각 센서 FQ-M□□□□-ECT FQ-M□□□□-M-ECT	센서 설정 데이터(신(Scene) 데이터, 시스템 데이터 등)는 백업, 복원, 비교 대상에서 제외됩니다. 센서의 설정 데이터를 외부 파일에 전송하거나 센서에 전송하려면 Sysmac Studio의 구성·설정의 편집 화면에서 [툴]–[센서 데이터]–[파일에 저장] 또는 [파일에서 읽기]를 실행하십시오. 자세한 내용은 📖"시각 센서 FQ-M 시리즈 위치 결정 전용 시각 센서 사용 설명서(SDNB-706)"를 참조하십시오.

EtherCAT 슬레이브의 모델	주의 사항
시각 센서 FZM1-□□□-ECT	센서 설정 데이터(신(Scene) 데이터, 시스템 데이터 등)는 백업, 복원, 비교 대상에서 제외됩니다. 센서의 설정 데이터를 USB 메모리에 저장하거나 컨트롤러에 읽어오려면 전용 툴을 사용하십시오. 자세한 내용은 📖"시각 센서 FZ3 시리즈 사용 설명서(SCHB-762)"를 참조하십시오.
디지털 I/O 터미널 GX-□D16□□ GX-□D32□8 GX-OC1601	*1
아날로그 I/O 터미널 GX-AD0471 GX-DA0271	*1
엔코더 입력 터미널 GX-EC0211 GX-EC0241	—
EtherCAT 분기 슬레이브 GX-JC0□	백업해야 하는 데이터를 내장하지 않습니다.
화이버 동축 변위 센서 ZW-CE1□	모든 설정 데이터가 백업, 복원, 비교 대상에서 제외됩니다. 설정 데이터를 저장하거나 컨트롤러에 읽어오는 방법은 📖"변위 센서 ZW 시리즈 화이버 동축 변위 센서 사용 설명서(SDNE-703)"를 참조하십시오.
디지털 타입 센서 E3NW-ECT E3X-ECT	센서의 파라메터는 백업, 복원, 비교 대상에서 제외됩니다.
타사 슬레이브	• ESI에서 올바르게 정의된 데이터만 백업, 복원, 비교 대상입니다. ESI에서 정의되지 않는 데이터를 백업, 복원, 비교하는 경우에는 별도의 전용 툴을 사용하십시오. • 백업, 복원, 비교에 실패한 경우에는 대처 방법을 슬레이브 제조업체에 문의하십시오.

*1. 복원 실행 완료 후에는 슬레이브의 전원을 재투입하십시오. 복원 후의 비교는 슬레이브의 전원을 재투입한 후 실행하십시오. 전원을 재투입하기 전에 비교를 실행하면 비교 불일치가 됩니다.

*2. 다음 중 하나의 조건에 해당하는 경우에는 백업에서 비교까지 또는 복원에서 비교될 때까지 서보 ON 상태로 하지 마십시오. 백업에서 비교까지의 간 또는 복원에서 비교될 때까지 서보 ON 상태로 하면 비교 전에 파라메터가 갱신되어 비교 불일치가 될 가능성이 있습니다.

실시간 자동 조정 모드가 유효로 설정됨(개체 3002h의 값이 1~4 또는 6)

적응 필터가 유효로 설정됨(개체 3200h의 값이 1 또는 2)

인버터 3G3AX-MX2-ECT, 3G3AX-RX-ECT의 파라메터 기입 방법

Ver.1.11 이상의 CPU 유니트인 경우에 인버터의 파라메터는 복원 대상에서 제외됩니다.
Ver.1.10 이하인 CPU 유니트인 경우에 인버터3G3AX-MX2-ECT, 3G3AX-RX-ECT를 연결하여 복원을 실행하면 '복원 실행 실패(EtherCAT 슬레이브)'(이벤트 코드 10300000Hex) 이벤트가 발생합니다.
백업한 파라메터를 인버터에 쓰려면 Sysmac Studio를 다음과 같이 조작하십시오. 자세한 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

1 **Sysmac Studio 백업 파일 가져오기 기능을 사용하여 백업 파일 내 인버터의 파라메터를 읽습니다.**
Sysmac Studio의 구성·설정의 인버터 파라메터 탭에 인버터의 파라메터가 표시됩니다.

2 **읽은 파라메터의 인버터의 형식과 실제로 연결된 인버터의 형식이 일치하는지 확인합니다.**

3 **구성·설정의 인버터의 파라메터 탭의 전송[PC→드라이브] 기능을 사용하여 파라메터를 인버터에 씁니다.**



사용상의 주의 사항

인버터 3G3AX-MX2-ECT의 인버터 모드 선택(파라메터 번호 b171) 기능을 사용하고자 하는 경우에는 파라메터를 쓰기 전에 연결된 인버터의 모드를 백업 시의 모드로 전환하십시오. 또한, 모드 전환을 유효로 하기 위해서는 모드 전환 조작 후 인버터 초기화가 필요합니다.



참고

Ver.1.10 이하인 CPU 유니트에서 인버터의 복원에 실패한 경우에도 인버터 이외의 유니트·슬레이브 설정, 사용자 프로그램·설정 및 변수의 현재값 등 모든 대상 데이터는 복원이 완료됩니다.

9-10 EtherCAT 슬레이브 터미널 연결 시의 백업 기능

EtherCAT 슬레이브 터미널은 SD 메모리 카드 백업 기능, Sysmac Studio 컨트롤러 백업 기능 및 Sysmac Studio 백업 가져오기 기능의 대상입니다.
여기에서는 EtherCAT 슬레이브 터미널을 연결한 경우의 백업 대상 데이터, 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부, 복원 실행의 조건에 대해 설명합니다.

사용상의 주의 사항

EtherCAT 슬레이브 터미널의 세이프티 컨트롤 유니트는 백업, 복원, 비교 대상에서 제외됩니다. 세이프티 컨트롤 유니트의 설정 데이터 가져오기/내보내기 기능은 "NX 시리즈 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)"를 참조하십시오.

버전 관련 정보

EtherCAT 슬레이브 터미널은 CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합에서 연결할 수 있습니다.

9-10-1 EtherCAT 슬레이브 터미널의 백업 대상 데이터

EtherCAT 슬레이브 터미널의 백업 대상 데이터는 EtherCAT 커플러 유니트와NX 유니트에서 다릅니다. 각각의 백업 대상 데이터는 다음과 같습니다.

(○: 대상 ×: 대상 외)

유니트	데이터	백업	복원	비교
EtherCAT 커플러 유니트	구성 정보*1	○	○	○
	유니트 동작 설정	○	○	○
NX 유니트	구성 정보*1	○	○	○
	유니트 동작 설정	○	○	○
	유니트 어플리케이션 데이터*2	○	○	○

*1. 유니트 구성 정보 및 I/O 할당 정보의 총칭입니다.
*2. 각 NX 유니트의 고유의 데이터입니다. 유니트 어플리케이션 데이터가 없는 NX 유니트도 있습니다.

사용상의 주의 사항

백업원과 동일한 유니트 구성의 EtherCAT 슬레이브 터미널에 백업 데이터를 복원하는 경우에 하드웨어 스위치를 백업원과 동일한 설정으로 하십시오. 하드웨어 스위치 설정이 백업 데이터와 다른 경우에는 복원할 수 없습니다. 감시 정보 이벤트인 '복원 시작 실패(EtherCAT 슬레이브)'가 발생합니다.

9-10-2 EtherCAT 슬레이브 터미널의 각 상태에서의 백업 기능 실행 가능 여부

EtherCAT 슬레이브 터미널의 각 상태에서의 백업, 복원, 비교의 실행 가능 여부의 사양은 다음과 같습니다.

EtherCAT 슬레이브 터미널의 상태	실행 가능 여부		
	백업	복원	비교
유니트 구성 정보 자동 생성 시	가능*1	가능*2	가능
NX 유니트 가입 대기	불가능*3	불가능*4	가능*5
EtherCAT 커플러 유니트 또는 NX 유니트에서 워치독 타이머 이상 발생 중	불가능*3	불가능*4	가능*5
버스 컨트롤러 이상 발생 중	불가능*3	불가능*4	가능*5
유니트 구성 정보 이상 발생 중	불가능*3	가능	가능*5
유니트 구성 비교 이상 발생 중	불가능*3	가능	가능*5
백업 시와 유니트 구성 정보 불일치	—	불가능*4	가능*5

*1. '유니트 구성 정보가 존재 안 함'이라는 정보가 백업됩니다.
*2. 복원 후에도 유니트 구성 자동 생성 상태가 됩니다.
*3. 이벤트 로그에 백업 실행 실패가 기록됩니다.
*4. 이벤트 로그에 복원 실행 실패가 기록됩니다.
*5. 비교 결과가 불일치가 됩니다.

9-10-3 EtherCAT 슬레이브 터미널의 복원 실행 조건

EtherCAT 슬레이브 터미널의 백업 데이터를 복원하기 위해서는 다음 조건을 충족해야 합니다.

- 백업 파일 내에 EtherCAT 커플러 유니트 및 NX 유니트의 데이터가 존재
- 백업원의 유니트 구성 정보와 복원 대상인 실제 유니트 구성이 일치
- 백업원의 EtherCAT 커플러 유니트의 시리얼 번호와 복원 대상인 EtherCAT 커플러 유니트의 시리얼 번호가 일치 단, 백업 파일 내의 [통신 커플러 유니트]–[유니트 동작 설정]–[시리얼 번호 확인 기준]의 설정이 '설정값 = 실제 기기'인 경우로 한정
- 백업원인 NX 유니트의 시리얼 번호와 복원 대상인 NX 유니트의 시리얼 번호가 일치 단, 백업 파일 내의 [통신 커플러 유니트]–[유니트 동작 설정]–[시리얼 번호 확인 기준]의 설정이 '설정값 = 실제 기기'인 경우로 한정
- 백업원의 EtherCAT 커플러 유니트의 하드웨어 스위치 정보와 복원 대상인 EtherCAT 커플러 유니트의 하드웨어 스위치 설정이 완전히 일치
- 백업원인 EtherCAT 커플러 유니트의 유니트 버전 설정값이 복원 대상인 EtherCAT 커플러 유니트의 실제 기기 유니트 버전과 일치하거나 또는 오래됨
- 백업원인 NX 유니트의 유니트 버전 설정값이 복원 대상인 NX 유니트의 실제 기기 유니트 버전과 일치하거나 또는 오래됨

9-11 NX 유니트 연결 시의 백업 기능

NX102 및 NX1P2 CPU 유니트 상의 NX 유니트는 SD 메모리 카드 백업 기능과 Sysmac Studio 컨트롤러 백업 기능의 대상입니다.
여기에서는 NX1P2 CPU 유니트에 NX 유니트를 연결한 경우의 백업 대상 데이터, 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부, 복원 실행의 조건에 대해 설명합니다.

사용상의 주의 사항

NX102 CPU 유니트에서는 세이프티 컨트롤 유니트 NX-SL3□□□를 장착할 수 있지만 세이프티 컨트롤 유니트는 백업, 복원, 비교 대상에서 제외됩니다. 세이프티 컨트롤 유니트의 설정 데이터 가져오기/내보내기 기능은 "NX 시리즈 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)"를 참조하십시오.

9-11-1 CPU 유니트 상의 NX 유니트의 백업 대상 데이터

CPU 유니트 상의 NX 유니트의 백업 대상 데이터는 다음과 같습니다.

(○: 대상 ×: 대상 외)

유니트	데이터	백업	복원	비교
NX 유니트	구성 정보*1	○	○	○
	유니트 동작 설정	○	○	○
	유니트 어플리케이션 데이터*2	○	○	○

*1. 유니트 구성 정보 및 I/O 할당 설정의 총칭입니다.
*2. 각 NX 유니트의 고유의 데이터입니다. 유니트 어플리케이션 데이터가 없는 NX 유니트도 있습니다.

9-11-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부

컨트롤러의 각 상태에서의 백업, 복원, 비교의 실행 가능 여부의 사양은 다음과 같습니다.

컨트롤러 상태	실행 가능 여부		
	백업	복원	비교
유니트 구성 정보 자동 생성 시	가능*1	가능*2	가능
NX 유니트에서 워치독 타이머 이상 발생 중	불가능*3	불가능*4	가능*5
NX 버스 컨트롤러 이상 발생 중	불가능*3	불가능*4	가능*5
유니트 구성 비교에 대한 이상 발생 중	불가능*3	가능	가능*5
백업 시와 유니트 구성 정보 불일치	—	불가능*4	가능*5

*1. '유니트 구성 정보가 존재 안 함'이라는 정보가 백업됩니다.
*2. 복원 후에도 유니트 구성 자동 생성 상태가 됩니다.
*3. 이벤트 로그에 백업 실행 실패(NX 유니트)가 기록됩니다.
*4. 이벤트 로그에 복원 실행 실패(NX 유니트)가 기록됩니다.
*5. 비교 결과가 불일치가 됩니다.

9-11-3 CPU 유니트 상의 NX 유니트의 복원 실행 조건

CPU 유니트 상의 NX 유니트의 백업 데이터를 복원하기 위해서는 다음 조건을 충족해야 합니다.

- 백업 파일 내에 해당 CPU 유니트 및 해당 CPU 유니트 상의 NX 유니트의 데이터가 존재
- 백업원의 유니트 구성 정보와 복원 대상인 실제 유니트 구성이 일치
- 백업원인 NX 유니트의 시리얼 번호와 복원 대상인 NX 유니트의 시리얼 번호가 일치 단, 백업 파일 내의 [CPU 랙]–[시리얼 번호 확인 기준]의 설정이 '설정값 = 실제 기기'인 경우로 한정
- 백업원인 NX 유니트의 유니트 버전 설정값이 복원 대상인 NX 유니트의 실제 기기 유니트 버전과 일치하거나 또는 오래됨

9-12 CJ 유니트 연결 시의 백업 기능

CJ 유니트는 SD 메모리 카드 백업 기능과 Sysmac Studio 컨트롤러 백업 기능의 대상입니다.
여기에서는 CJ 유니트를 연결한 경우의 백업 대상 데이터, 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부, 복원 실행의 조건에 대해 설명합니다.

사용상의 주의 사항

CJ 유니트를 연결할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

9-12-1 CJ 유니트의 백업 대상 데이터

백업 대상 데이터는 CJ 유니트용 메모리의 현재값, CJ 유니트 내의 파라메터입니다. 이러한 데이터에는 CJ 유니트 내에 있는 데이터와 CPU 유니트 내에 있는 데이터가 있습니다.
그러나 이들은 자동으로 일괄 백업, 복원, 비교되므로 사용자는 데이터의 소재를 의식할 필요는 없습니다.

9-12-2 컨트롤러의 각 상태에서의 백업 기능 실행 가능 여부

컨트롤러의 각 상태에서의 CJ 유니트의 백업, 복원, 비교의 실행 가능 여부의 사양은 다음과 같습니다.

컨트롤러 상태	실행 가능 여부		
	백업	복원	비교
'I/O 버스 확인 에러' 발생 중 '엔드 커버 연결 안 됨' 발생 중 '유니트/증설 랙 연결이 잘못됨' 발생 중 '유니트 번호 중복' 발생 중 'I/O 점수 초과' 발생 중 'I/O 설정 확인 에러' 발생 중	불가능*1	불가능*2	가능*3
CJ 유니트 재시작 중	불가능*1	가능	가능*3

*1. 이벤트 로그에 백업 실행 실패(CJ 유니트)가 기록됩니다.
*2. 이벤트 로그에 복원 실행 실패(CJ 유니트)가 기록됩니다.
*3. 비교 결과가 불일치가 됩니다.

9-12-3 CJ 유니트의 복원 실행 조건

CJ 유니트의 백업 데이터를 복원하기 위해서는 다음 조건을 충족해야 합니다.

- 백업 파일 내에 CJ 유니트의 데이터가 존재
- 백업 파일 내의 유니트 구성과 복원 대상인 실제 유니트 구성이 일치
- 해당 CJ 유니트 고유의 조건을 충족(해당 CJ 유니트 고유의 조건에 대해서는 각 CJ 유니트의 설명서를 참조하십시오).



사용상의 주의 사항

CJ 유니트를 연결한 상태에서 SD 메모리 카드 백업 기능의 복원 또는 Sysmac Studio 컨트롤러 백업 기능의 복원을 실행한 경우에 'CPU 유니트 서비스 감시 이상'이 발생할 수 있습니다. 이는 CPU 유니트에서 CJ 유니트에 대한 서비스 처리가 일정 시간 내에 완료되지 않았던 것을 의미합니다. 그러나 이것은 복원에 시간이 걸린 경우에 발생하는 것이지 이상은 아닙니다. 그 경우의 응답 시간은 다음과 같습니다.

- 통신계 CJ 유니트인 경우에 MS LED가 적색으로 점멸합니다.
- 7세그먼트 LED가 있는 CJ 유니트인 경우에 7세그먼트 LED의 표시가 "HE"가 됩니다.
- ERH LED가 있는 CJ 유니트인 경우에 ERH LED가 점등합니다.
- 컨트롤러 이벤트 로그에 이벤트 코드=00000002Hex, CPU 유니트 서비스 감시 이상 또는 리프레시 타임아웃 이벤트가 등록됩니다.
- CJ 시리즈 고속 카운터 유니트 CJ1W-CT021을 연결한 경우에 컨트롤러 이벤트 로그에 이벤트 코드=68010000Hex, 유니트 에러, 부속 정보1=0002Hex 이벤트가 등록됩니다.

9-13 백업 관련 파일

여기에서는 백업 관련 파일의 사양에 대해 설명합니다.
백업 관련 파일은 'Sysmac Studio 변수/메모리 백업 기능' 이외의 백업 기능으로 사용하는 파일입니다.

9-13-1 백업 관련 파일의 종류

백업 관련 파일에는 백업 파일, 복원 명령 파일, 자동 전송 명령 파일, 비교 결과 파일의 4종류가 있습니다.

- **백업 파일**
  백업 관련 기능으로 취급하는 컨트롤러의 데이터를 파일화한 것입니다. 백업을 실행했을 때 작성됩니다.

- **복원 명령 파일**
  SD 메모리 카드에서 복원을 실행할 때, 전송하는 대상의 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다. 백업을 실행했을 때 작성됩니다.

- **자동 전송 명령 파일**
  SD 메모리 카드에서 자동 전송을 실행할 때, 전송하는 대상의 데이터 그룹을 지정하는 파일입니다. PC의 텍스트 편집기로 파일을 편집하여 전송 대상을 지정합니다. 백업을 실행했을 때 작성됩니다.

● **비교 결과 파일**

컨트롤러의 데이터와CPU 유니트에 장착한 SD 메모리 카드의 백업 파일의 데이터와의 비교 결과를 저장하는 파일입니다.

비교 결과 파일에는 다음 4종류가 있습니다. 모두 SD 메모리 카드 백업 기능 비교를 실행했을 때 작성됩니다.

비교 결과 파일	내용
컨트롤러 비교 결과 파일	복원 명령 파일로 지정된 모든 백업 데이터의 비교 결과가 기술된 파일입니다.
EtherCAT 슬레이브 비교 결과 파일	EtherCAT 슬레이브별 비교 결과가 기술된 파일입니다. 복원 명령 파일에서 유니트·슬레이브 설정을 복원 대상으로 하고, 백업 파일 내에 EtherCAT 슬레이브 설정이 포함된 경우에 작성됩니다.
EtherCAT 슬레이브 터미널 비교 결과 파일	EtherCAT 커플러 유니트 및 NX 유니트별 비교 결과가 기술된 파일입니다. 다음 3개의 모든 조건에 해당하는 경우에 작성됩니다. • 복원 명령 파일에서 유니트·슬레이브 설정을 복원 대상으로 함 • 백업 파일 내에 EtherCAT 슬레이브 설정이 포함됨 • EtherCAT 슬레이브 터미널을 연결함 EtherCAT 슬레이브 터미널 비교 결과 파일이 작성되는 경우에는 반드시 EtherCAT 슬레이브 비교 결과 파일도 동시에 작성됩니다.
NX 유니트 비교 결과 파일*1	CPU 유니트 상의 NX 유니트별 비교 결과가 기술된 파일입니다. 복원 명령 파일에서 유니트·슬레이브 설정을 복원 대상으로 하고, 백업 파일 내에 CPU 유니트 상의 NX 유니트 설정이 포함된 경우에 작성됩니다.
CJ 유니트 비교 결과 파일*2	CJ 유니트별 비교 결과가 기술된 파일입니다. 복원 명령 파일에서 유니트·슬레이브 설정을 복원 대상으로 하고, 백업 파일 내에 CJ 유니트 설정이 포함된 경우에 작성됩니다.

*1. CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.
*2. CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

9-13-2 백업 파일의 사양

백업 파일의 파일명, 작성 타이밍, 작성 디렉토리는 다음과 같습니다.

파일명

백업 파일의 파일명은 CPU 유니트의 시리즈에 따라 다릅니다.

파일	대상 CPU 유니트	파일명
백업 파일	NX 시리즈 CPU 유니트	NXBackup.dat
	NJ 시리즈 CPU 유니트	NJBackup.dat

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	CPU 유니트 전면 스위치	백업 실행 시	SD 메모리 카드 내의 루트 디렉토리
	시스템 정의 변수	백업 실행 시	SD 메모리 카드 내의 시스템 정의 변수로 지정한 디렉토리
	Sysmac Studio SD 메모리 카드 화면	백업 실행 시	SD 메모리 카드 내의 화면에서 사용자가 지정한 디렉토리
	전용 명령*1	백업 실행 시	SD 메모리 카드 내의 BackupToMemoryCard 명령의 입력 변수로 사용자가 지정한 디렉토리
Sysmac Studio 컨트롤러 백업 기능	Sysmac Studio 컨트롤러 백업 화면	백업 실행 시	PC 내의 화면에서 사용자가 지정한 디렉토리
Sysmac Studio 백업 파일 가져오기/내보내기 기능	Sysmac Studio 백업 파일 내보내기 화면	내보내기 실행 시	PC 내의 화면에서 사용자가 지정한 디렉토리

*1. CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

9-13-3 복원 명령 파일의 사양

복원 명령 파일의 파일명, 작성 타이밍, 작성 디렉토리, 데이터 그룹의 지정 방법은 다음과 같습니다.

파일명

파일	파일명
복원 명령 파일	RestoreCommand.ini

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	CPU 유니트 전면 스위치	백업 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	백업 실행 시	백업 파일과 동일한 디렉토리
	Sysmac Studio SD 메모리 카드 화면	백업 실행 시	백업 파일과 동일한 디렉토리
Sysmac Studio 컨트롤러 백업 기능	Sysmac Studio 컨트롤러 백업 화면	백업 실행 시	백업 파일과 동일한 디렉토리
Sysmac Studio 백업 파일 가져오기/내보내기 기능	Sysmac Studio 백업 파일 내보내기 화면	내보내기 실행 시	백업 파일과 동일한 디렉토리

복원하는 데이터 그룹 지정 방법

복원 명령 파일은 데이터 그룹별로 복원을 지정하기 위한 파일입니다.
데이터 그룹별로 지정한 부분을 PC의 텍스트 편집기로 편집하면 지정을 변경할 수 있습니다.
예를 들어 다음 표의 '파일 내용' 8번째 행의 'Variable=yes'를 'Variable=no'로 변경하면 '변수의 현재값'이 복원 대상에서 제외됩니다.

파일 내용(파일 작성 시의 기본값)	설명
[Restore] ; --- User Program and Configuration. --- ; Always select "yes". UserProgram=yes	사용자 프로그램·설정 yes 고정: 복원 필수
; --- IP Address of Built-in EtherNet/IP Port Settings. --- ; "yes":will be restored, "no":will not be restored IPAdr=yes	내장 EtherNet/IP 포트의 IP 주소 yes/no: 복원함/복원 안 함
; --- Present values of variables (Retained variables only). --- ; "yes":will be restored, "no":will not be restored Variable=yes	변수의 현재값(유지 속성만 해당) yes/no: 복원함/복원 안 함
; --- Present values of memory used for CJ-series Units (Holding, EM, and DM Area data). --- ; "yes":will be restored, "no":will not be restored Memory=yes	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) yes/no: 복원함/복원 안 함
:---Unit/Slave Parameters.--- ; "yes";will be restored."no";will not be restored UnitConfig=yes	유니트·슬레이브 설정 yes/no: 복원함/복원 안 함
; --- Absolute encoder home offset. --- ; "yes":will be restored, "no":will not be restored AbsEncoder=yes	앱솔루트 엔코더 원점 위치 오프셋 yes/no: 복원함/복원 안 함

(주의) 1 이 표의 파일 내용은 복원 명령 파일 작성 시의 내용입니다. 기술된 모든 데이터 그룹이 복원 대상입니다.
2 백업 파일 작성 시 백업 파일 내에 있고, 복원 가능한 데이터 그룹의 지정을 기술한 파일이 기술됩니다.
3 모두 반각 영숫자입니다. 또한 명령은 대문자와 소문자를 구별하지 않습니다.
4 Ver.1.13 이하의 CPU 유니트에서 백업한 경우 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목은 작성되지 않습니다.
Ver.1.14 이상의 CPU 유니트에서 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목이 작성되지 않은 복원 명령 파일을 사용하면 'IPAdr=yes'로 동작합니다.
Ver.1.13 이하의 복원 명령 파일과 Ver.1.14 이상의 복원 명령 파일의 호환성에 대해서는 '복원 명령 파일의 호환성(9-66페이지)'을 참조하십시오.

사용상의 주의 사항

복원 명령 파일을 편집하는 경우에는 선택 가능한 데이터 그룹의 'yes/no'만 기술하십시오. 그 이외의 기술을 편집하여 복원하면 컨트롤러가 의도하지 않는 동작을 하는 경우가 있습니다.

복원 명령 파일의 호환성

Ver.1.13 이하의 복원 명령 파일과 Ver.1.14 이상의 복원 명령 파일의 호환성을 나타냅니다.

복원 명령 파일과 백업 파일을 작성한 CPU 유니트의 유니트 버전	복원 대상인 CPU 유니트의 유니트 버전	
	Ver.1.13 이하	Ver.1.14 이상
Ver.1.13 이하	복원 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 복원됨	복원 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 복원됨
Ver.1.14 이상	복원 불가능 Ver.1.13 이하의 CPU 유니트에서는 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목을 해석할 수 없기 때문에 감시 정보의 '복원 시작 실패'가 발생, 이 때의 이상의 세부 내용은 '0104Hex: 복원 명령 파일의 기술 내용이 잘못됨'이 됨	복원 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 IPAdr의 yes/no에 따름

9-13-4 자동 전송 명령 파일의 사양

자동 전송 명령 파일의 파일명, 작성 타이밍, 작성 디렉토리, 데이터 그룹의 지정 방법은 다음과 같습니다.

파일명

파일	파일명
자동 전송 명령 파일	AutoloadCommand.ini

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	CPU 유니트 전면 스위치	백업 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	백업 실행 시	백업 파일과 동일한 디렉토리
	Sysmac Studio SD 메모리 카드 화면	백업 실행 시	백업 파일과 동일한 디렉토리
Sysmac Studio 컨트롤러 백업 기능	Sysmac Studio 컨트롤러 백업 화면	백업 실행 시	백업 파일과 동일한 디렉토리
Sysmac Studio 백업 파일 가져오기/내보내기 기능	Sysmac Studio 백업 파일 내보내기 화면	내보내기 실행 시	백업 파일과 동일한 디렉토리

자동 전송하는 데이터 그룹 지정 방법

자동 전송 명령 파일은 데이터 그룹별로 자동 전송을 지정하기 위한 파일입니다.
데이터 그룹별로 지정한 부분을 PC의 텍스트 편집기로 편집하면 지정을 변경할 수 있습니다.
예를 들어 다음 표의 '파일 내용' 8번째 행의 'Variable=yes'를 'Variable=no'로 변경하면 '변수의 현재값'이 자동 전송 대상에서 제외됩니다.

파일 내용(파일 작성 시의 기본값)	설명
[Autoload] ; --- User Program and Configuration. --- ; Always select "yes". UserProgram=yes	사용자 프로그램·설정 yes 고정: 전송 필수
; --- IP Address of Built-in EtherNet/IP Port Settings. --- ; "yes":will be transferred, "no":will not be transferred IPAdr=yes	내장 EtherNet/IP 포트의 IP 주소 yes/no: 전송함/전송 안 함
; --- Present values of variables (Retained variables only). --- ; "yes":will be transferred, "no":will not be transferred Variable=yes	변수의 현재값(유지 속성만 해당) yes/no: 전송함/전송 안 함
; --- Present values of memory used for CJ-series Units (Holding, EM, and DM Area data). --- ; "yes":will be transferred, "no":will not be transferred Memory=yes	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) yes/no: 전송함/전송 안 함

(주의) 1 이 표의 파일 내용은 자동 전송 명령 파일 작성 시의 내용입니다. 기술된 모든 데이터 그룹이 자동 전송 대상입니다.
2 백업 파일 작성 시 백업 파일 내에 있고, 자동 전송 가능한 데이터 그룹의 지정을 기술한 파일이 기술됩니다.
3 모두 반각 영숫자입니다. 또한 명령은 대문자와 소문자를 구별하지 않습니다.
4 Ver.1.13 이하의 CPU 유니트에서 백업한 경우 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목은 작성되지 않습니다.
Ver.1.14 이상의 CPU 유니트에서 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목이 작성되지 않은 복원 명령 파일을 사용하면 'IPAdr=yes'로 동작합니다.
Ver.1.13 이하의 자동 전송 명령 파일과 Ver.1.14 이상의 자동 전송 명령 파일의 호환성에 대해서는 '자동 전송 명령 파일의 호환성(9-68페이지)'을 참조하십시오.

사용상의 주의 사항

자동 전송 명령 파일을 편집하는 경우에는 선택 가능한 데이터 그룹의 'yes/no'만 기술하십시오. 그 이외의 기술을 편집하여 자동 전송을 하면 컨트롤러가 의도하지 않은 동작을 하는 경우가 있습니다.

자동 전송 명령 파일의 호환성

Ver.1.13 이하의 자동 전송 명령 파일과 Ver.1.14 이상의 자동 전송 명령 파일의 호환성을 나타냅니다.

자동 전송 명령 파일과 백업 파일을 작성한 CPU 유니트의 유니트 버전	자동 전송대상인 CPU 유니트의 유니트 버전	
	Ver.1.13 이하	Ver.1.14 이상
Ver.1.13 이하	자동 전송 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 자동 전송됨	자동 전송 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 자동 전송됨
Ver.1.14 이상	자동 전송 불가능 Ver.1.13 이하의 CPU 유니트에서는 '내장 EtherNet/IP 포트 설정의 IP 주소' 항목을 해석할 수 없기 때문에 전체 정지 이상의 '자동 전송 시작 에러'가 발생, 이 때의 이상의 세부 내용은 0104Hex: 자동 전송 명령 파일의 기술 내용이 잘못됨'이 됨	자동 전송 가능 '내장 EtherNet/IP 포트 설정의 IP 주소'는 IPAdr의 yes/no에 따름

9-13-5 컨트롤러 비교 결과 파일의 사양

컨트롤러 비교 결과 파일의 파일명, 작성 타이밍, 작성 디렉토리, 비교 결과의 확인 방법은 다음과 같습니다.

파일명

파일	파일명
컨트롤러 비교 결과 파일	VerifyResult.log

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	Sysmac Studio SD 메모리 카드 화면	비교 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	비교 실행 시	백업 파일과 동일한 디렉토리
	CPU 유니트 전면 스위치	비교 실행 시	백업 파일과 동일한 디렉토리

(주의) 단, SD 메모리 카드가 쓰기 방지된 경우에는 작성되지 않습니다.

비교 결과 확인 방법

비교 결과 파일은 데이터 그룹별로 컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터의 비교 결과를 저장했습니다.
데이터 그룹별 확인 결과를 저장한 부분에서 비교 결과를 확인할 수 있습니다.
'Result=Matched'라고 쓰여 있으면 비교 일치, 'Result=Not matched'라고 쓰여 있으면 비교 불일치입니다.
파일 내용이 다음과 같은 경우에 사용자 프로그램·설정이 비교 일치, 유니트/슬레이브 설정이 비교 불일치입니다.

파일 내용	설명
[UserProgram] ; --- User Program and Configuration. --- Result=Matched [UnitConfig] ; --- Unit/Slave Parameters. --- Result=Not matched	사용자 프로그램·설정 Matched: 비교 일치, Not matched: 비교 불일치 유니트/슬레이브 설정 Matched: 비교 일치, Not matched: 비교 불일치

1 비교 실시한 데이터 그룹의 비교 결과만 기술됩니다.
2 '내장 EtherNet/IP 포트 설정의 IP 주소'의 비교 결과는 Ver.1.14 이상의 CPU 유니트에서도 '사용자 프로그램·설정' 항목에 포함됩니다.

9-13-6 EtherCAT 비교 결과 파일의 사양

EtherCAT 비교 결과 파일의 파일명, 작성 타이밍, 작성 디렉토리, 비교 결과의 확인 방법은 다음과 같습니다.

파일명

파일	파일명
EtherCAT 비교 결과 파일	VerifyResult_ECAT.log

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	Sysmac Studio SD 메모리 카드 화면	비교 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	비교 실행 시	백업 파일과 동일한 디렉토리
	CPU 유니트 전면 스위치	비교 실행 시	백업 파일과 동일한 디렉토리

(주의) 단, SD 메모리 카드가 쓰기 방지된 경우에는 작성되지 않습니다.

비교 결과 확인 방법

비교 결과 파일은 데이터 그룹별로 컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터의 비교 결과를 저장했습니다.

EtherCAT 슬레이브별 확인 결과를 저장한 부분에서 비교 결과를 확인할 수 있습니다.

'Result=Matched'라고 쓰여 있으면 비교 일치, 'Result=Not matched'라고 쓰여 있으면 비교 불일치입니다.

다음과 같은 파일 내용인 경우에 비교 결과는 다음과 같습니다.

- 비교 일치: EtherCAT 슬레이브 Master, EtherCAT 슬레이브 터미널 E002
- 비교 불일치: EtherCAT 슬레이브 E001

파일 내용	설명
[Verification Results] ; --- EtherCAT Parameters. --- : --- See the VerifyResult_ECAT_NX.log about detail result if NX mark is included in square brackets. [Master] Result=Matched [E001] Result=Not matched Factor=Verification error [E002:NX] Result=Matched	슬레이브는 사용자가 설정한 장치명으로 표시합니다. EtherCAT 슬레이브 터미널인 경우에는 장치명 뒤에 ":NX"가 붙습니다. *1 비교 결과는 다음과 같은 표기로 나타냅니다. Result=Matched 비교 일치 Result=Not matched 비교 불일치

*1. EtherCAT 슬레이브 터미널이 비교 대상인 경우에 EtherCAT 슬레이브 터미널 비교 결과 파일이 작성됩니다. EtherCAT 슬레이브 터미널의 비교 결과의 세부 내용은 EtherCAT 슬레이브 터미널 비교 결과 파일에 기술됩니다.

(주의) 비교 실시한 EtherCAT 슬레이브의 비교 결과만 기술됩니다.

9-13-7 EtherCAT 슬레이브 터미널 비교 결과 파일의 사양

EtherCAT 슬레이브 터미널 비교 결과 파일의 파일명, 작성 타이밍, 작성 디렉토리, 비교 결과의 확인 방법은 다음과 같습니다.

파일명

파일	파일명
EtherCAT 슬레이브 터미널 비교 결과 파일	VerifyResult_ECAT_NX.log

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	Sysmac Studio SD 메모리 카드 화면	비교 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	비교 실행 시	백업 파일과 동일한 디렉토리
	CPU 유니트 전면 스위치	비교 실행 시	백업 파일과 동일한 디렉토리

(주의) 단, SD 메모리 카드가 쓰기 방지된 경우에는 작성되지 않습니다.

비교 결과 확인 방법

비교 결과 파일은 데이터 그룹별로 컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터의 비교 결과를 저장했습니다.

EtherCAT 커플러 유니트 및 NX 유니트별 확인 결과를 저장한 부분에서 비교 결과를 확인할 수 있습니다. 'Result=Matched'라고 쓰여 있으면 비교 일치, 'Result=Not matched'라고 쓰여 있으면 비교 불일치입니다.

다음과 같은 파일 내용인 경우에 비교 결과는 다음과 같습니다.

- 비교 일치 : EtherCAT 커플러 유니트 E002, NX 유니트 N1, NX 유니트 N2
- 비교 불일치: EtherCAT 커플러 유니트 E005, NX 유니트 N3

파일 내용	설명
[Verification Results] ; --- NX Parameters. --- [E002:UnitNo.0 NX-ECC201] Result=Matched [N1:UnitNo.1 NX-AD2203] Result=Matched [N2:UnitNo.2 NX-DA2203] Result=Matched [N3:UnitNo.3 NX-TS3201] Result=Not matched Factor=Verification error [E005:UnitNo.0 NX-ECC201] Result=Not matched Factor=Verification error	유니트는 다음과 같은 사양으로 표기합니다. {장치명}: UnitNo.{유니트 번호}[공백]{유니트 형식} {장치명} 사용자가 설정한 장치명 {유니트 번호} 10진수의 문자열, 값의 범위는 0-125 {유니트 형식} 유니트 형식을 나타내는 문자열 형식의 뒷부분에서부터 연속되는 공백은 삭제합니다. 비교 결과는 다음과 같은 표기로 나타냅니다. Result=Matched 비교 일치 Result=Not matched 비교 불일치

9-13-8 NX 유니트 비교 결과 파일의 사양

NX 유니트 비교 결과 파일의 파일명, 작성 타이밍, 작성 디렉토리, 비교 결과의 확인 방법은 다음과 같습니다.

사용상의 주의 사항

CPU 유니트 상에서 NX 유니트를 사용할 수 있는 것은 NX102 및 NX1P2 CPU 유니트뿐입니다.

파일명

파일	파일명
NX 유니트 비교 결과 파일	VerifyResult_NXUnit.log

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	Sysmac Studio SD 메모리 카드 화면	비교 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	비교 실행 시	백업 파일과 동일한 디렉토리
	CPU 유니트 전면 스위치	비교 실행 시	백업 파일과 동일한 디렉토리

(주의) 단, SD 메모리 카드가 쓰기 방지된 경우에는 작성되지 않습니다.

비교 결과 확인 방법

비교 결과 파일은 데이터 그룹별로 컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터의 비교 결과를 저장했습니다.
NX 유니트별 확인 결과를 저장한 부분에서 비교 결과를 확인할 수 있습니다.
'Result=Matched'라고 쓰여 있으면 비교 일치, 'Result=Not matched'라고 쓰여 있으면 비교 불일치입니다.
다음과 같은 파일 내용인 경우에 비교 결과는 다음과 같습니다.

- 비교 일치: NX 유니트 N1
- 비교 불일치: NX 유니트 N3
- 비교 대상 외: NX 유니트 N2

파일 내용	설명
[Verification Results] ; --- NX Parameters. --- [N1:UnitNo.1 NX-AD2203] Result=Matched [N2:UnitNo.2 NX-DA2203] Result=Not verified [N3:UnitNo.3 NX-TS3201] Result=Not matched Factor=Verification error	유니트는 다음과 같은 사양으로 표기합니다. {장치명}: UnitNo.{유니트 번호}[공백]{유니트 형식} {장치명} 사용자가 설정한 장치명 {유니트 번호} 10진수의 문자열, 값의 범위는 0-125 {유니트 형식} 유니트 형식을 나타내는 문자열 형식의 뒷부분에서부터 연속되는 공백은 삭제합니다. 비교 결과는 다음과 같은 표기로 나타냅니다. Result=Matched 비교 일치 Result=Not matched 비교 불일치 Result=Not verified 비교 대상 외

9-13-9 CJ 유니트 비교 결과 파일의 사양

CJ 유니트 비교 결과 파일의 파일명, 작성 타이밍, 작성 디렉토리, 비교 결과의 확인 방법은 다음과 같습니다.

사용상의 주의 사항

CJ 유니트를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트뿐입니다.

파일명

파일	파일명
CJ 유니트 비교 결과 파일	VerifyResult_CJUnit.log

파일 작성 타이밍과 작성 디렉토리

기능	조작 방법	작성 타이밍	작성 디렉토리
SD 메모리 카드 백업 기능	Sysmac Studio SD 메모리 카드 화면	비교 실행 시	백업 파일과 동일한 디렉토리
	시스템 정의 변수	비교 실행 시	백업 파일과 동일한 디렉토리
	CPU 유니트 전면 스위치	비교 실행 시	백업 파일과 동일한 디렉토리

(주의) 단, SD 메모리 카드가 쓰기 방지된 경우에는 작성되지 않습니다.

비교 결과 확인 방법

비교 결과 파일은 데이터 그룹별로 컨트롤러의 데이터와 CPU 유니트에 장착한 SD 메모리 카드의 백업 파일 데이터의 비교 결과를 저장했습니다.
CJ 유니트별 확인 결과를 저장한 부분에서 비교 결과를 확인할 수 있습니다.
'Result=Matched'라고 쓰여 있으면 비교 일치, 'Result=Not matched'라고 쓰여 있으면 비교 불일치입니다.
파일 내용이 다음과 같은 경우에 CJ1W-CRM21(MODE0)과 CJ1W-EIP21이 비교 일치, CJ1W-DRM21과 CJ1W-PRM21-DPV1이 비교 불일치입니다.

파일 내용	설명
[Verification Results] ; --- CJ Unit Parameters. --- [Rack0 Slot0: CJ1W-CRM21(MODE0) UnitNo.10] Result=Matched [Rack0 Slot9: CJ1W-DRM21 UnitNo.0] Result=Not matched [Rack1 Slot0: CJ1W-EIP21 UnitNo.10] Result=Matched [Rack1 Slot1: CJ1W-PRM21-DPV1 UnitNo.1] Result=Not matched	유니트는 다음과 같은 사양으로 표기합니다. Rack{랙 번호}[공백]Slot{슬롯 번호}:[공백]{유니트 형식}[공백]UnitNo.{유니트 호기 번호} {랙 번호} 10진수의 문자열, 값의 범위는 0-3 {슬롯 번호} 10진수의 문자열, 값의 범위는 0-9 {유니트 형식} 유니트 형식을 나타내는 문자열 유니트 형식은 사이클릭 초기 데이터에서 취득한 것을 사용합니다. 형식의 뒷부분에서부터 연속되는 공백은 삭제합니다. {유니트 호기 번호} 10진수의 문자열, 제로 억제, 값의 범위는 CPU 고기능 유니트라면 0-15이고 고기능 I/O 유니트라면 0-95 비교 일치/비교 불일치는 다음과 같은 표기로 나타냅니다. Result=Matched 비교 일치 Result=Not matched 비교 불일치

(주의) 비교 실시한 EtherCAT 슬레이브의 비교 결과만 기술됩니다.

9-14 백업 기능의 파일 간 호환성

백업을 실행할 때와 복원할 때의 조건이 다르면 파일 간의 호환성을 고려해야 합니다.
호환성을 고려해야 하는 것은 다음과 같은 3가지입니다.

- 백업한 기능과 복원하는 기능이 다른 경우
- 백업원과 복원 대상인 CPU 유니트의 형식이 다른 경우
- 백업원과 복원 대상인 CPU 유니트 및 유니트, 슬레이브의 유니트 버전이 다른 경우

또한, 여기에서는 각 백업 기능의 '복원', '자동 전송', '프로그램 전송', '읽기'를 '복원'이라고 총칭합니다.

9-14-1 백업 기능 간 호환성

백업한 기능과 복원하는 기능이 다른 경우의 파일의 호환성은 다음과 같습니다.

(○: 호환성 있음, ×: 호환성 없음)

백업한 기능	복원하는 기능				
	SD 메모리 카드 백업 기능(복원) SD 메모리 카드에서 컨트롤러	자동 전송 기능, 프로그램 전송 기능	Sysmac Studio 컨트롤러백업 기능(복원) PC에서 컨트롤러	Sysmac Studio 변수/메모리 백업 기능 (복원) PC에서 컨트롤러	Sysmac Studio 백업 파일 가져오기 PC에서 프로젝트
SD 메모리 카드 백업 기능 (백업) 컨트롤러에서 SD 메모리 카드	○	○	○	×	○*1
Sysmac Studio 컨트롤러 백업 기능(백업) 컨트롤러에서 PC	○	○	○	×	○*1
Sysmac Studio 변수/메모리 백업 기능(백업) 컨트롤러에서 PC	×	×	×	○	×
Sysmac Studio 백업 파일 내보내기 프로젝트에서 PC	○*1	○*1	○*1	×	○

*1. 단, 다음 데이터는 대상에서 제외됩니다.

- 컨트롤러 설정의 내장 EtherNet/IP 포트명, 내장 EtherNet/IP용 태그 데이터 링크 설정
- 유니트 구성의 CPU 고기능 유니트 영역
- 조작 권한 인증
- 데이터 트레이스 설정
- 표준 시간대 설정
- 변수의 현재값
- CJ 유니트용 메모리의 현재값
- 앱솔루트 엔코더 원점 위치 오프셋

참고

Sysmac Studio 변수/메모리 백업 기능에서 취급하는 파일은 기타 백업 기능에서 취급하는 파일과 달라서 호환성이 없습니다.

이 기능의 자세한 내용은 '9-8 Sysmac Studio 변수/메모리 백업 기능(9-46페이지)'을 참조하십시오.

9-14-2 CPU 유니트의 형식 간의 호환성

백업원과 복원 대상인 CPU 유니트의 형식 그룹이 다른 경우의 파일의 호환성은 다음과 같습니다.

(○: 호환성 있음, ×: 호환성 없음)

백업원인 CPU 유니트 형식	복원 대상인 CPU 유니트 형식								
	NX701-	NX102-		NX1P2-		NJ501-	NJ301-	NJ101-	
	1700 1600	1200 1100 1000	9000	1140DT 1140DT1 1040DT 1040DT1	9024DT 9024DT1	1500 1400 1300	1200 1100	1000	9000
NX701-1700 NX701-1600	○	×*1	×*1	×*1	×*1	×*1	×*1	×*1	×*1
NX102-1200 NX102-1100 NX102-1000	×*1	○	×*1	×*1	×*1	×*1	×*1	×*1	×*1
NX102-9000	×*1	×*1	○	×*1	×*1	×*1	×*1	×*1	×*1
NX1P2-1140DT NX1P2-1140DT1 NX1P2-1040DT NX1P2-1040DT1	×*1	×*1	×*1	○	×*1	×*1	×*1	×*1	×*1
NX1P2-9024DT NX1P2-9024DT1	×*1	×*1	×*1	×*1	○	×*1	×*1	×*1	×*1
NJ501-1500 NJ501-1400 NJ501-1300	×*1	×*1	×*1	×*1	×*1	○	×*1	×*1	×*1
NJ301-1200 NJ301-1100	×*1	×*1	×*1	×*1	×*1	×*1	○	×*1	×*1
NJ101-1000	×*1	×*1	×*1	×*1	×*1	×*1	×*1	○	×
NJ101-9000	×*1	×*1	×*1	×*1	×*1	×*1	×*1	×	○

*1. Sysmac Studio 변수/메모리 백업 기능은 호환성이 있습니다. 단, CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합인 경우로 제한됩니다. Sysmac Studio 변수/메모리 백업 기능에서의 CPU 유니트 형식 간의 호환성에 대해서는 '9-8-3 CPU 유니트 형식 간의 호환성(9-47페이지)'을 참조하십시오.

참고

데이터베이스 연결 CPU 유니트, SECS/GEM 탑재 CPU 유니트, NJ Robotics CPU 유니트 및 NC 통합 컨트롤러 등과는 호환성이 없습니다.

이러한 CPU 유니트의 호환성에 대해서는 각 CPU 유니트의 설명서를 참조하십시오.

CPU 유니트의 형식의 호환성이 있는 경우에도 각각의 CPU 유니트 형식 간에 제약이 있는 경우가 있습니다.
CPU 유니트의 형식 간에 호환성이 있는 경우와 없는 경우에 복원하는 기능에 따라 다음과 같이 동작합니다.

(○: 복원을 실행함 ×: 복원을 실행 안 함)

CPU 유니트 형식 간의 호환성	복원하는 기능				
	SD 메모리 카드 백업 기능(복원) SD 메모리 카드에서 컨트롤러	자동 전송 기능, 프로그램 전송 기능	Sysmac Studio컨트롤러 백업 기능(복원) PC에서 컨트롤러*1	Sysmac Studio 변수/메모리 백업 기능(복원) PC에서 컨트롤러	Sysmac Studio 백업 파일 가져오기 기능 PC에서 프로젝트
호환성 있음	○*2	○*2	○	○	○
호환성 없음	×*3	×*4	×	×	×

*1. 이 기능으로 백업한 파일만 복원할 수 있습니다.
*2. 백업 파일의 내용이 복원 대상인 사양 범위 외인 경우에는 정상적으로 동작하지 않고 전체 정지 이상 레벨의 컨트롤러 이상 또는 부분 정지 이상 레벨의 컨트롤러 이상이 발생합니다. 예를 들어 사용하는 제어축 수가 사양 이외일 때 이 동작을 합니다.
*3. 감시 정보 '복원 시작 실패'가 발생합니다.
*4. 자동 전송 기능에서는 전체 정지 이상 레벨의 컨트롤러 이상의 '자동 전송 시작 에러', 프로그램 전송 기능에서는 감시 정보 레벨의 컨트롤러 이상의 'SD 메모리 카드 프로그램 전송 시작 실패'가 발생합니다.

9-14-3 CPU 유니트의 유니트 버전 호환성

백업원인 CPU 유니트의 유니트 버전과 복원 대상의 유니트 버전이 다른 경우의 백업 파일의 호환성은 다음과 같습니다.
또한, 백업, 복원 전후에 유니트 버전이 동일한 경우에는 제약 없이 복원을 실행할 수 있습니다.

(○: 복원을 실행함 ×: 복원을 실행 안 함)

CPU 유니트의 유니트 버전	복원하는 기능			
	SD 메모리 카드 백업 기능(복원) SD 메모리 카드에서 컨트롤러	자동 전송 기능, 프로그램 전송 기능	Sysmac Studio 컨트롤러 백업 기능 (복원) PC에서 컨트롤러	Sysmac Studio 변수/메모리 백업 기능 (복원) PC에서 컨트롤러
복원 대상의 버전이 높음	○	○	○	○
복원 대상의 버전이 낮음	×	×*1	×	○

*1. 자동 전송 기능에서는 전체 정지 이상 레벨의 컨트롤러 이상의 '자동 전송 시작 에러', 프로그램 전송 기능에서는 감시 정보 레벨의 컨트롤러 이상의 'SD 메모리 카드 프로그램 전송 시작 실패'가 발생합니다.

9-15 백업 기능과 동시에 실행할 수 없는 기능

다음과 같은 기능은 백업 기능과 동시에 실행할 수 없습니다. CPU 유니트가 이러한 기능을 실행 중인 경우에는 백업 기능을 실행하지 마십시오. 또한, 백업 기능을 실행 중인 경우에는 이러한 기능을 실행하지 마십시오.

- 백업 기능을 이미 실행 중
- 동기 전송(PC→컨트롤러)
- 온라인 편집 실행
- 메모리 전체 클리어 실행
- 표준 시간대 변경 실행
- 캠 테이블 저장 명령(MC_SaveCamTable) 실행
- CPU 유니트 이름 쓰기 실행
- 슬레이브 터미널의 파라메터 설정 전송 실행

10

통신 설정

이 장에서는 CPU 유니트의 온라인 연결, 다른 기기와의 연결에 대해 설명합니다.

10-1 통신 시스템의 개요 10-2
10-1-1 개요 10-3
10-2 Sysmac Studio 연결 10-7
10-2-1 온라인 연결이 가능한 구성 10-7
10-2-2 온라인 연결할 수 없는 구성 10-8
10-3 다른 컨트롤러 및 슬레이브와의 연결 10-10
10-3-1 컨트롤러-컨트롤러 간의 연결 구성 10-10
10-3-2 컨트롤러-슬레이브 간의 연결 구성 10-13
10-4 표시기 및 시리얼 통신 기기와의 연결 10-14
10-4-1 표시기와의 연결 10-14
10-4-2 시리얼 통신 기기와의 연결 10-15

10-1 통신 시스템의 개요

여기에서는 NJ/NX 시리즈의 통신 시스템의 개요에 대해 설명합니다.

전체 순서 중에서 다음과 같이 회색으로 표시한 위치에 대응합니다.

STEP 1. 소프트웨어 설계

- STEP 1-1 I/O 및 처리의 설계
- STEP 1-2 태스크 설계
- STEP 1-3 프로그램 설계

STEP 2. 소프트웨어 설정 및 프로그래밍

- STEP 2-1 슬레이브/유니트 구성
- STEP 2-2 컨트롤러 설정
- STEP 2-3 프로그래밍
- STEP 2-4 오프라인 디버그

STEP 3. 장착 및 하드웨어 설정

STEP 4. 배선

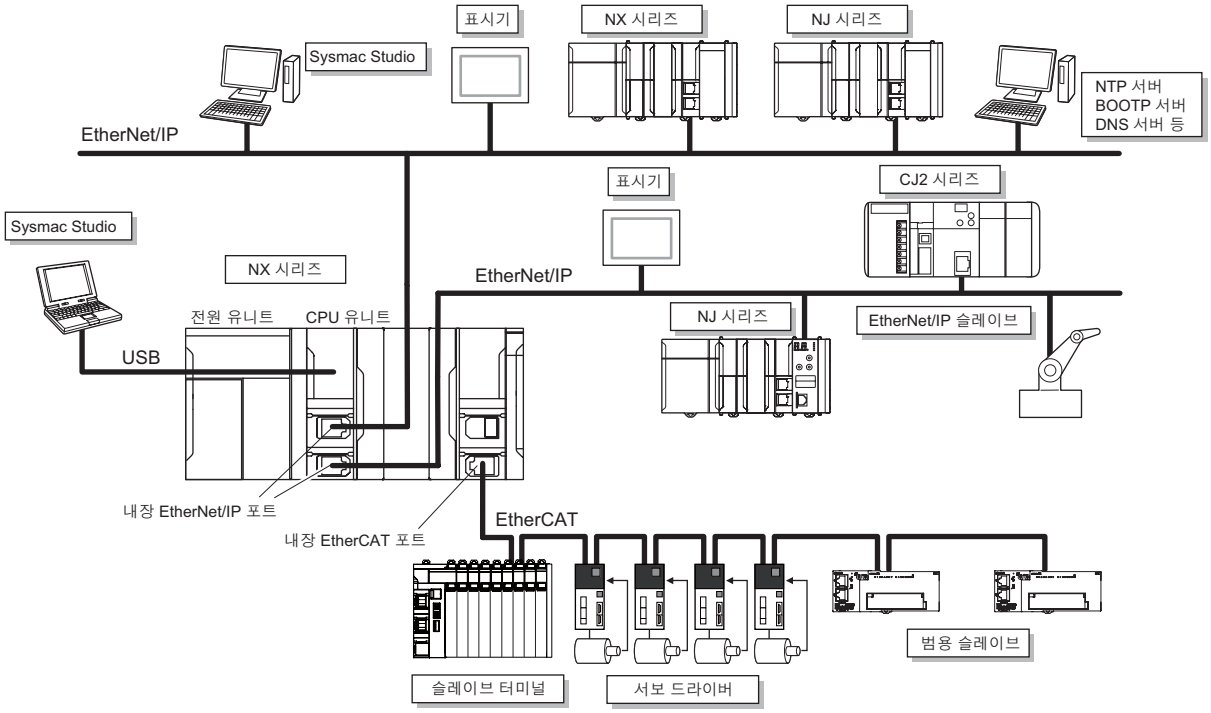
STEP 5. 실제 기기에 의한 동작 확인과 운전

자세한 내용은 📖'1-3 NJ/NX 시리즈의 사용 순서 전체(1-16페이지)'를 참조하십시오.

10-1-1 개요

- **NX701 시스템**

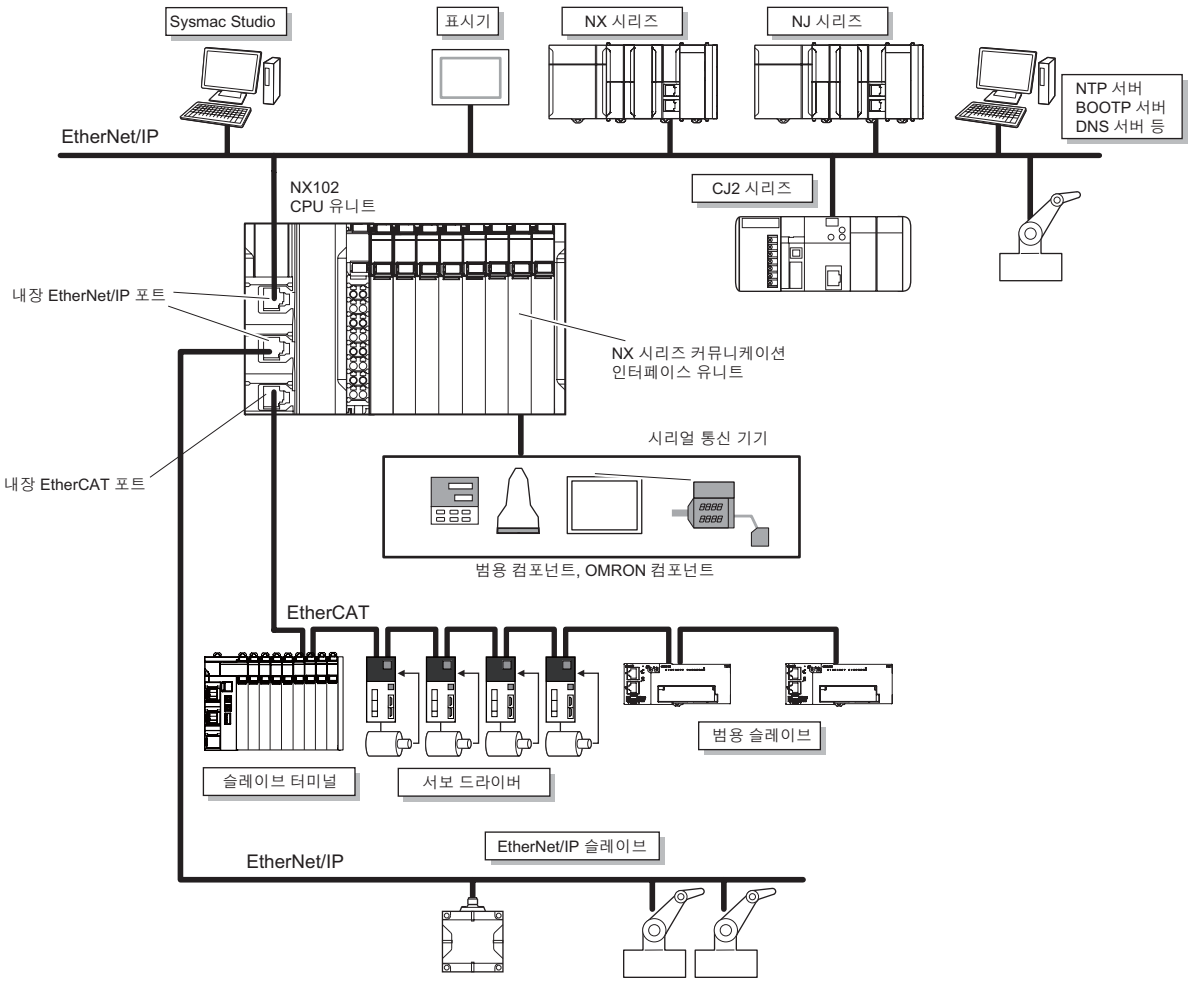
  NX701 시스템은 다음과 같은 통신 시스템을 구축할 수 있습니다.



연결 대상		연결 방법
Sysmac Studio 연결		USB 또는 내장 EtherNet/IP 포트를 사용
컨트롤러 간의 연결	NJ/NX 시리즈 컨트롤러 또는 CJ2 시리즈 PLC와 연결	내장 EtherNet/IP 포트를 사용
컨트롤러와 슬레이브 간의 연결	서보 드라이버, 범용 슬레이브를 연결	내장 EtherCAT 포트를 사용
표시기 연결		내장 EtherNet/IP 포트를 사용
각종 서버와 연결	FTP 서버, BOOTP 서버, DNS 서버, NTP 서버를 연결	내장 EtherNet/IP 포트를 사용

● **NX102 시스템**

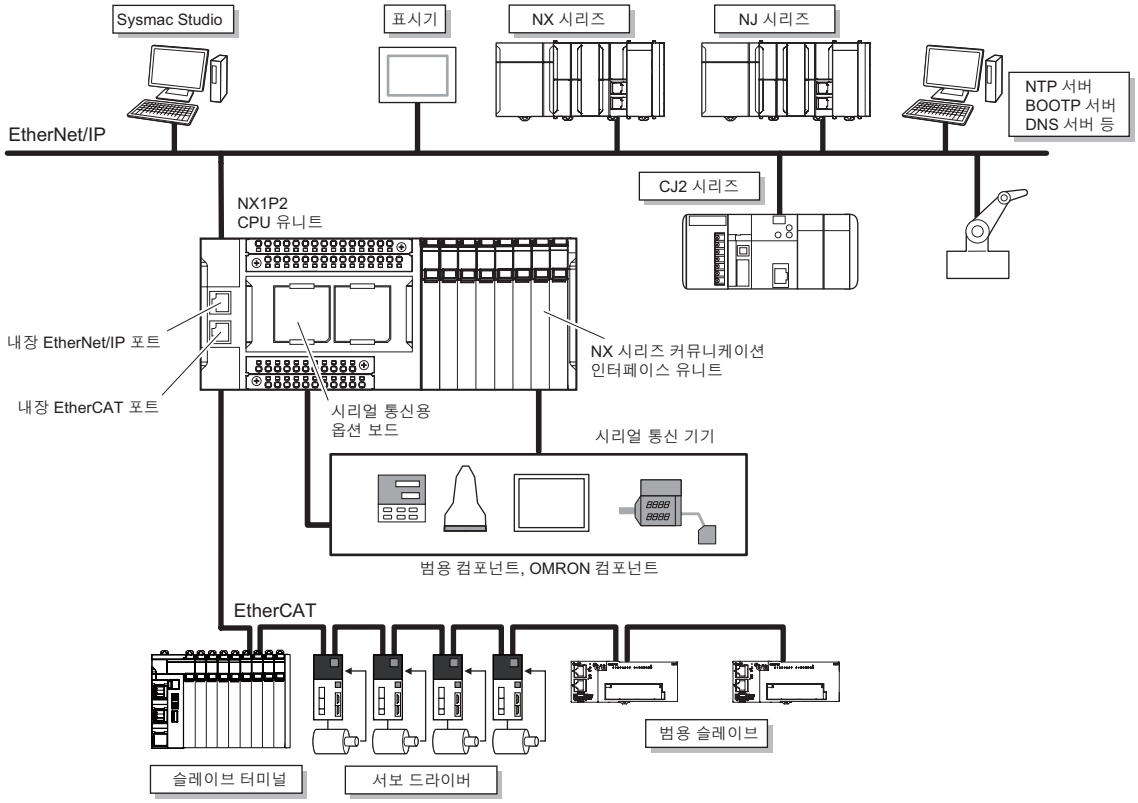
NX102 시스템은 다음과 같은 통신 시스템을 구축할 수 있습니다.



연결 대상		연결 방법
Sysmac Studio 연결		내장 EtherNet/IP 포트를 사용
컨트롤러 간의 연결	NJ/NX 시리즈 컨트롤러 또는 CJ2 시리즈 PLC와 연결	내장 EtherNet/IP 포트를 사용
컨트롤러와 슬레이브 간의 연결	서보 드라이버, 범용 슬레이브를 연결	내장 EtherCAT 포트를 사용
표시기 연결		내장 EtherNet/IP 포트를 사용
각종 서버와 연결	FTP 서버, BOOTP 서버, DNS 서버, NTP 서버를 연결	내장 EtherNet/IP 포트를 사용

● NX1P2 시스템

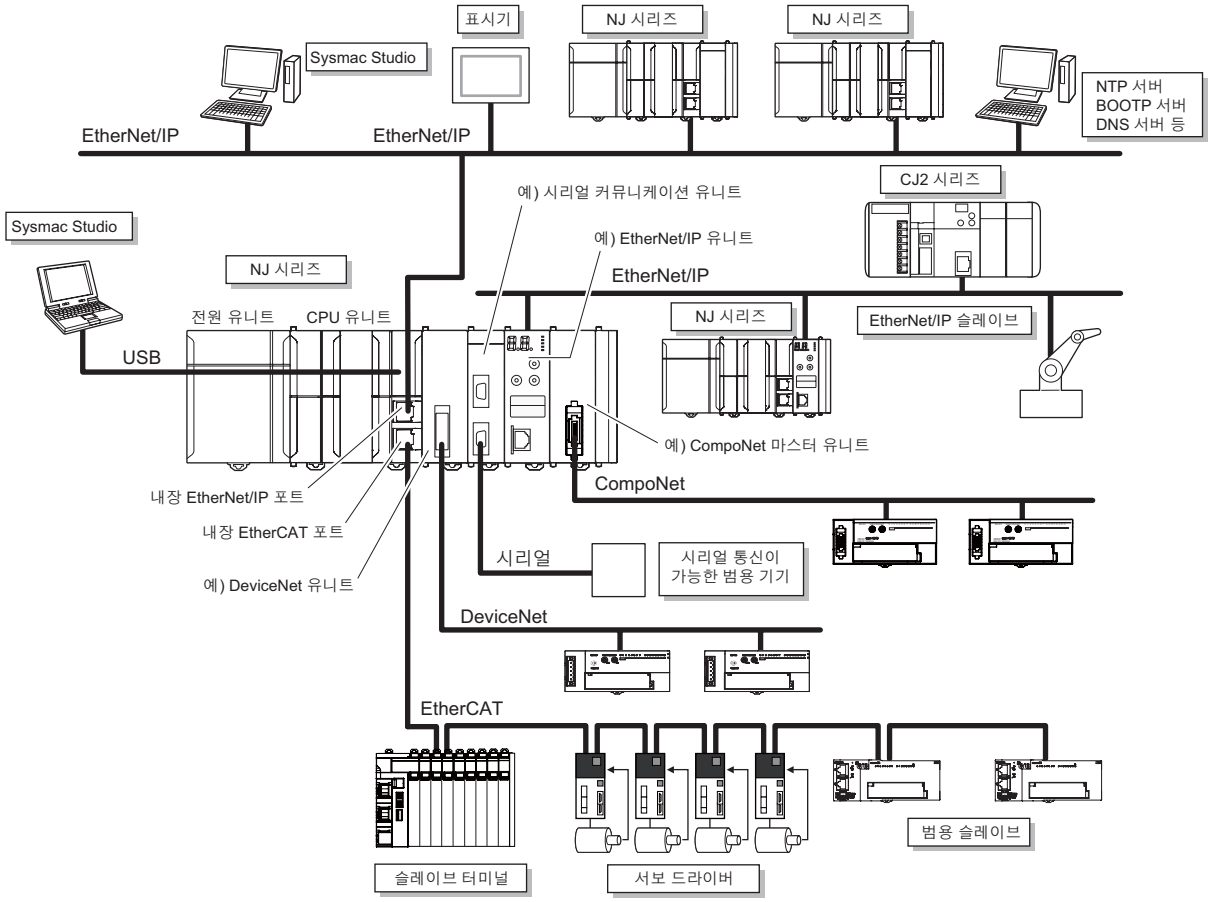
NX1P2 시스템은 다음과 같은 통신 시스템을 구축할 수 있습니다.



연결 대상		연결 방법
Sysmac Studio 연결		내장 EtherNet/IP 포트를 사용
컨트롤러 간의 연결	NJ/NX 시리즈 컨트롤러 또는 CJ2 시리즈 PLC와 연결	내장 EtherNet/IP 포트를 사용
컨트롤러와 슬레이브 간의 연결	서보 드라이버, 범용 슬레이브를 연결	내장 EtherCAT 포트를 사용
표시기 연결		내장 EtherNet/IP 포트를 사용
각종 서버와 연결	FTP 서버, BOOTP 서버, DNS 서버, NTP 서버를 연결	내장 EtherNet/IP 포트를 사용

● **NJ 시리즈 시스템**

NJ 시리즈 시스템은 다음과 같은 통신 시스템을 구축할 수 있습니다.



연결 대상		연결 방법
Sysmac Studio 연결		USB 또는 내장 EtherNet/IP 포트를 사용
컨트롤러 간의 연결	NJ/NX 시리즈 컨트롤러 또는 CJ2 시리즈 PLC와 연결	내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트*1 의 포트를 사용
컨트롤러와 슬레이브 간의 연결	서보 드라이버, 범용 슬레이브를 연결	내장 EtherCAT 포트를 사용
	I/O 제어	DeviceNet 유니트를 장착하고 DeviceNet을 사용하거나 또는 CompoNet 마스터 유니트를 장착하고 CompoNet을 사용
표시기 연결		내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트*1, *2 의 포트를 사용
시리얼 통신에서의 연결		시리얼 커뮤니케이션 유니트를 장착
각종 서버와 연결	FTP 서버*3 , BOOTP 서버, DNS 서버, NTP 서버를 연결	내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트*1의 포트를 사용

*1. 유니트 버전이 Ver.2.1 이상인 EtherNet/IP 유니트를 사용하십시오.
또한, CPU 유니트는 Ver.1.01 이상, Sysmac Studio 는 Ver.1.02 이상을 사용하십시오.
버전 업그레이드 관련 정보는 'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

*2. NA 시리즈 표시기는 CPU 유니트의 내장 EtherNet/IP 포트로 연결하십시오. NS 시리즈 표시기의 문제 해결사를 사용하는 경우에는 NS 시리즈 표시기를 CPU 유니트의 내장 EtherNet/IP 포트로 연결하십시오.

*3. FTP 서버는 CPU 유니트 Ver.1.08 이상과, Sysmac Studio Ver.1.09 이상과의 조합에서 사용할 수 있습니다.

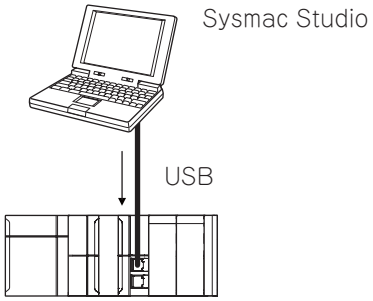
10-2 Sysmac Studio 연결

여기에서는 Sysmac Studio의 NJ/NX 시리즈에 대한 연결 구성에 대해 설명합니다.

10-2-1 온라인 연결이 가능한 구성

Sysmac Studio에서는 NJ/NX 시리즈 CPU 유니트의 페리페럴 USB 포트 또는 내장 EtherNet/IP 포트에 연결하여 온라인 연결이 가능합니다.

● USB로 연결



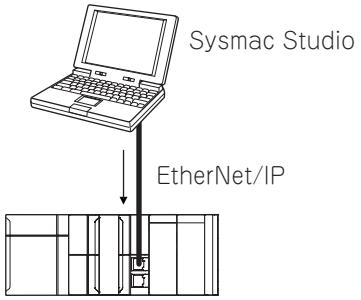
Sysmac Studio에서 직접 연결합니다. 연결 장치를 지정할 필요는 없습니다.

(주의) PC와 CPU 유니트 간은 USB2.0 인증의 케이블을 사용하여 USB 허브를 통하지 않고 연결하십시오.

(주의) NX102 및 NX1P2 CPU 유니트는 페리페럴 USB 포트가 없으므로 USB로 연결할 수 없습니다.

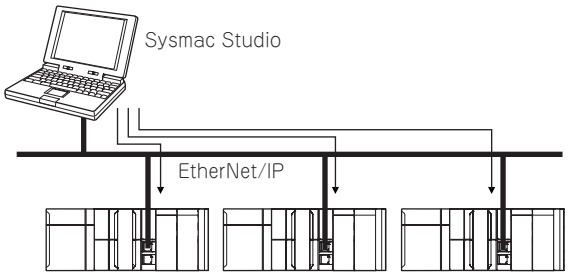
● EtherNet/IP로 연결

1:1 연결:



- Sysmac Studio에서 직접 연결합니다. IP 주소 및 연결 장치를 지정할 필요는 없습니다.
- 스위칭 허브의 유무에 관계없이 연결할 수 있습니다.
- 호환 케이블, 스트레이트 케이블 모두 가능합니다.
- NX701 및 NX102 CPU 유니트인 경우에 포트 1만 1:1 연결이 가능합니다.

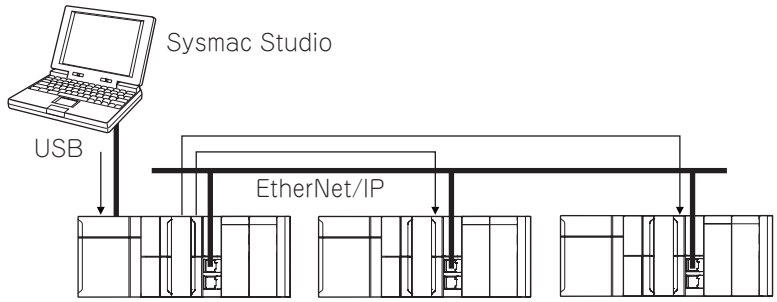
1:N 연결:



- Sysmac Studio에서 연결 상대의 IP 주소를 지정합니다.
- 호환 케이블, 스트레이트 케이블 모두 가능합니다.

- **USB 경유 EtherNet/IP로 연결**

  NJ 시리즈 또는 NX701 CPU 유니트의 USB 포트를 경유하여 내장 EtherNet/IP 포트에 연결할 수 있습니다.



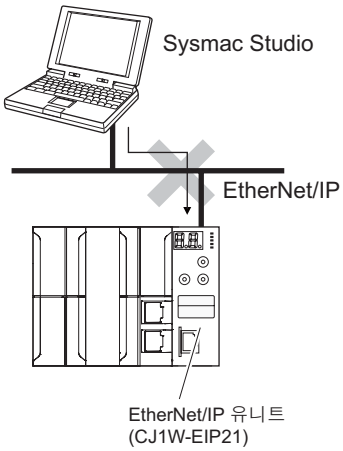
- Sysmac Studio에서 연결 상대의 IP 주소를 지정합니다.
- 호환 케이블, 스트레이트 케이블 모두 가능합니다.

(주의) PC와 CPU 유니트 간은 USB2.0 인증의 케이블을 사용하여 USB 허브를 통하지 않고 연결하십시오.
(주의) NX102 및 NX1P2 CPU 유니트는 페리페럴 USB 포트를 장비하지 않았습니다.

10-2-2 온라인 연결할 수 없는 구성

- **EtherNet/IP 유니트 경유로 연결**

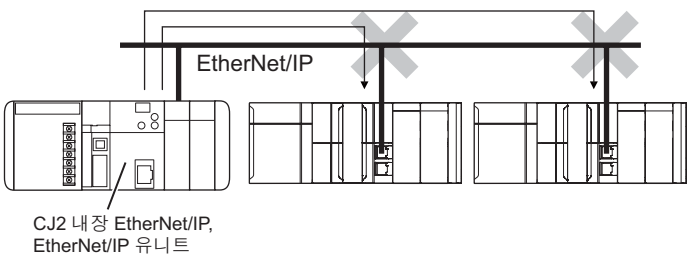
  NJ 시리즈 CPU 유니트에 연결한 EtherNet/IP 유니트(CJ1W-EIP21) 경유로 연결할 수 없습니다.



(주의) NX 시리즈 CPU 유니트에서는 EtherNet/IP(CJ1W-EIP21) 유니트를 사용할 수 없습니다.

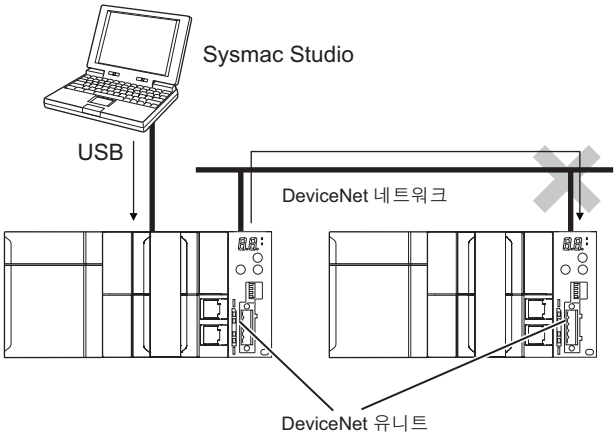
- **CS/CJ 시리즈의 EtherNet/IP 유니트/포트를 라우팅**

  CS/CJ 시리즈의 EtherNet/IP 유니트/포트(CS1W-EIP21, CJ1W-EIP21, CJ2 내장 EtherNet/IP 포트, CJ2M 내장 EtherNet/IP 포트)를 라우팅하여 NJ/NX 시리즈 CPU 유니트에 연결할 수 없습니다.



- **DeviceNet 등의 EtherNet/IP 이외를 라우팅**

  EtherNet/IP 네트워크 이외를 라우팅할 수 없습니다. (Controller Link 네트워크, DeviceNet 네트워크 등)



(주의) NX 시리즈 CPU 유니트에서는 DeviceNet 유니트를 연결할 수 없습니다.

10-3 다른 컨트롤러 및 슬레이브와의 연결

여기에서는 컨트롤러와 컨트롤러 간, 컨트롤러와 슬레이브 간의 연결 구성에 대해 설명합니다.

10-3-1 컨트롤러-컨트롤러 간의 연결 구성

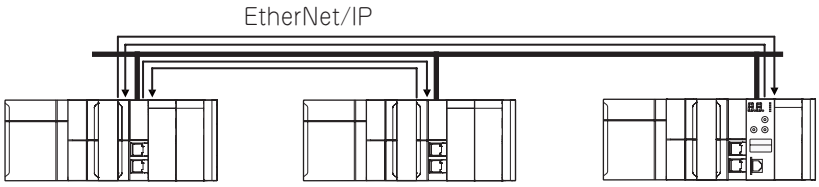
EtherNet/IP

내장 EtherNet/IP 포트 또는 EtherNet/IP 유니트(CJ1W-EIP21)의 포트를 사용합니다.

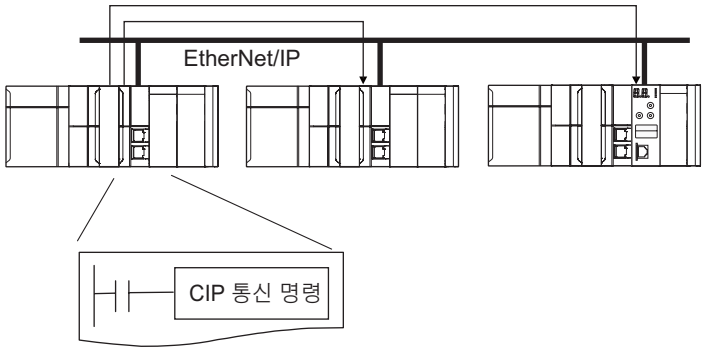
내장 EtherNet/IP 포트에 대해서는 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"를 EtherNet/IP 유니트(CJ1W-EIP21)에 대해서는 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-355)"을 참조하십시오.

(주의) NX 시리즈 CPU 유니트에서는 EtherNet/IP(CJ1W-EIP21) 유니트를 사용할 수 없습니다.

- **태그 데이터 링크**
  EtherNet/IP의 NJ/NX 시리즈 CPU 유니트와 태그 데이터 링크를 할 수 있습니다.

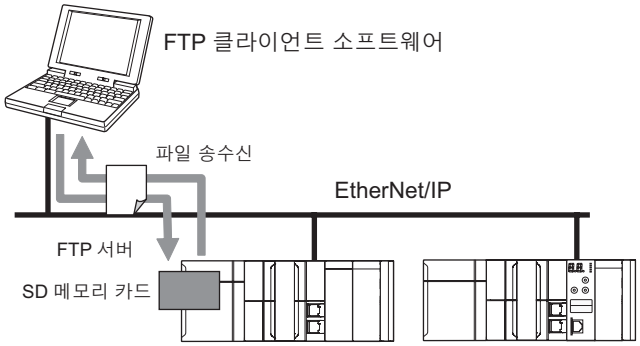


- **메시지 통신**
  CIP 메시지를 프로그램에서 전송할 수 있습니다.



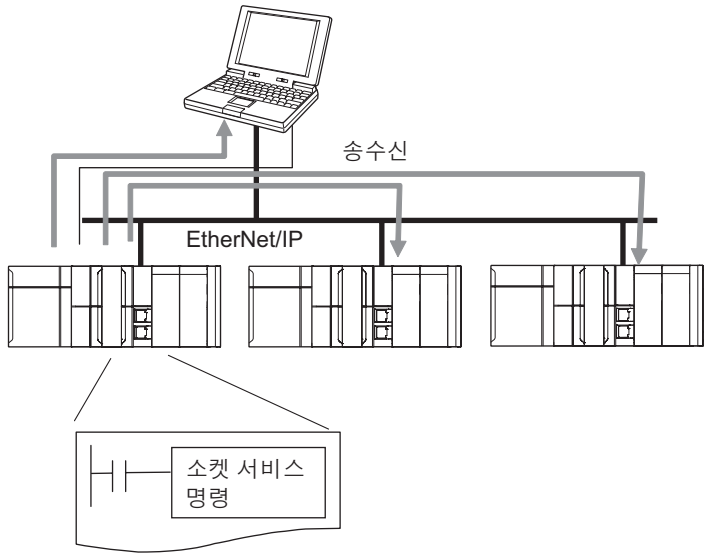
- **파일 송수신**

  FTP 클라이언트 소프트웨어에서 NJ/NX 시리즈 CPU 유니트에 장착된 SD 메모리 카드 내의 파일을 송수신을 할 수 있습니다.



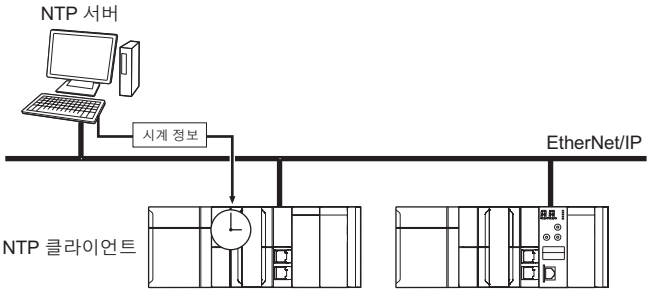
- **소켓 서비스**

  TCP 또는 UDP의 기능을 사용자 프로그램에서 직접 이용하면 호스트 컴퓨터-PLC 사이 또는 컨트롤러 사이에서 임의의 데이터를 상대 노드와 송수신할 수 있습니다.
  이 기능은 내장 EtherNet/IP 포트에서만 사용할 수 있습니다.



- **시계 정보 갱신**

  어떤 조건 성립 시 NTP 서버에서 시각 정보를 취득하여 내장 시계 시각을 갱신할 수 있습니다.



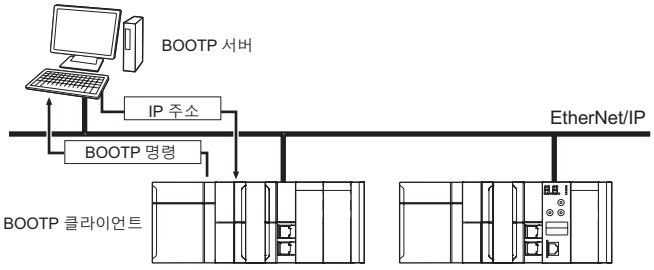
- **호스트명으로 지정**

  NTP 서버 및 SNMP 관리자의 IP 주소 또는 소켓 명령 및 CIP 통신 명령에서의 송신 대상에 대해 IP 주소가 아닌 호스트명으로 지정할 수 있습니다(DNS 클라이언트 기능 또는 Hosts 설정).
  예: DNS 서버에서 호스트명을 설정



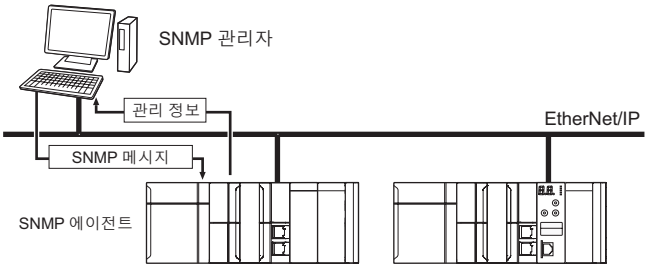
- **전원 투입 시 IP 주소 취득**

  전원 투입 시에 내장 EtherNet/IP 포트의 IP 주소를 BOOTP 서버에서 취득할 수 있습니다.



- **SNMP 에이전트 지정**

  SNMP 관리자를 사용한 네트워크 관리 소프트웨어에 내장 EtherNet/IP 포트의 내부 상태 정보를 제공합니다.

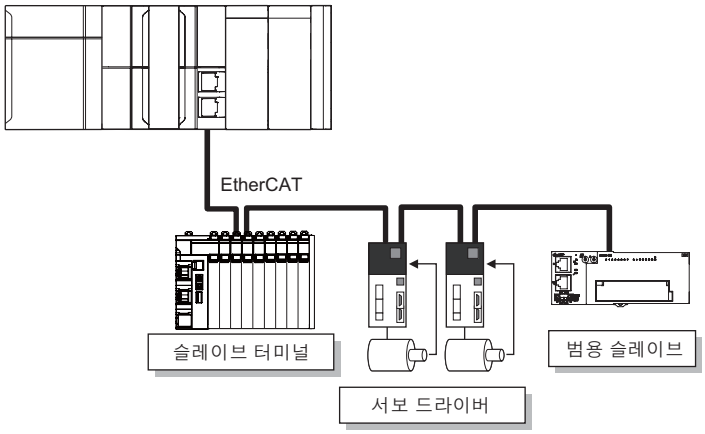


10-3-2 컨트롤러-슬레이브 간의 연결 구성

EtherCAT

서보 드라이버, 범용 슬레이브와의 고속·고정밀도 통신이 가능합니다.

자세한 내용은 📖"NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"를 참조하십시오.



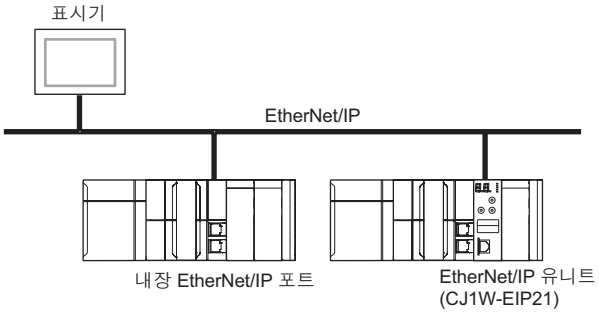
10-4 표시기 및 시리얼 통신 기기와의 연결

여기에서는 표시기, 시리얼 통신 기기와 NJ/NX 시리즈 컨트롤러의 연결 구성에 대해 설명합니다.

10-4-1 표시기와의 연결

- **EtherNet/IP**

  내장 EtherNet/IP포트 또는 EtherNet/IP 유니트(CJ1W-EIP21)를 사용합니다.



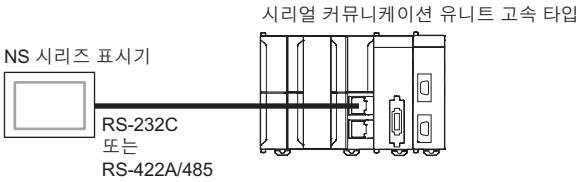
(주의) NX 시리즈 CPU 유니트에서는 EtherNet/IP(CJ1W-EIP21) 유니트를 사용할 수 없습니다.

NA 시리즈 표시기는 CPU 유니트의 내장 EtherNet/IP 포트로 연결하십시오.
NS 시리즈 표시기의 문제 해결사를 사용하는 경우에는 NS 시리즈 표시기를 CPU 유니트의 내장 EtherNet/IP 포트로 연결하십시오.

내장 EtherNet/IP 포트에 대해서는 "NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)", EtherNet/IP 유니트(CJ1W-EIP21)에 대해서는 "CJ 시리즈 EtherNet/IP 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-355)"을 참조하십시오.

- **시리얼 통신**

  시리얼 커뮤니케이션 유니트를 사용합니다(NS 시리즈 표시기).



(주의) NX 시리즈 CPU 유니트에서는 시리얼 커뮤니케이션 유니트를 사용할 수 없습니다.

시리얼 커뮤니케이션 유니트에 대해서는 "CJ 시리즈 시리얼 커뮤니케이션 유니트 사용 설명서 NJ시리즈 연결편(SBCD-354)"을 참조하십시오.

10-4-2 시리얼 통신 기기와의 연결

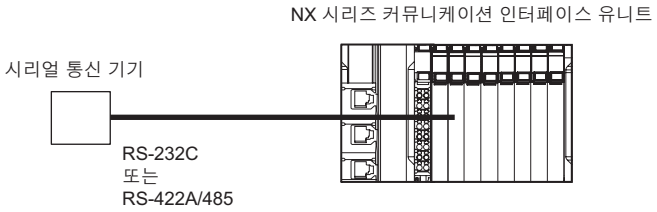
- **NX701 CPU 유니트**

  시리얼 커뮤니케이션 유니트 및 시리얼 통신용 옵션 보드는 사용할 수 없습니다.

- **NX102 CPU 유니트**

  NX 시리즈 커뮤니케이션 인터페이스 유니트를 사용합니다.

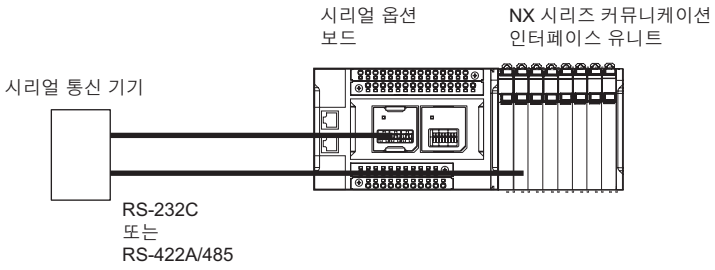
  NX 시리즈 커뮤니케이션 인터페이스 유니트의 자세한 내용은 "NX 시리즈 커뮤니케이션 인터페이스 유니트 사용 설명서(SBCA-422)"를 참조하십시오.



- **NX1P2 CPU 유니트**

  시리얼 통신용 옵션 보드 또는 NX 시리즈 커뮤니케이션 인터페이스 유니트를 사용합니다.

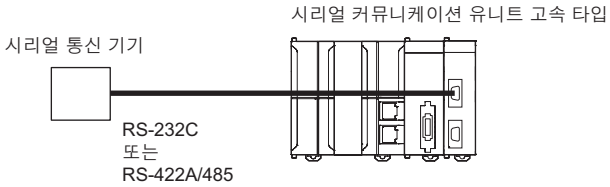
  시리얼 통신용 옵션 보드의 자세한 내용은 "NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)", NX 시리즈 커뮤니케이션 인터페이스 유니트의 자세한 내용은 "NX 시리즈 커뮤니케이션 인터페이스 유니트 사용 설명서(SBCA-422)"를 참조하십시오



- **NJ 시리즈 CPU 유니트**

  시리얼 커뮤니케이션 유니트를 사용합니다.

  자세한 내용은 "CJ 시리즈 시리얼 커뮤니케이션 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-354)"을 참조하십시오.



11

실제 사용 순서 예

이 장에서는 NJ 시리즈 컨트롤러를 실제로 사용하는 경우의 사용 순서 예에 대해 설명합니다.

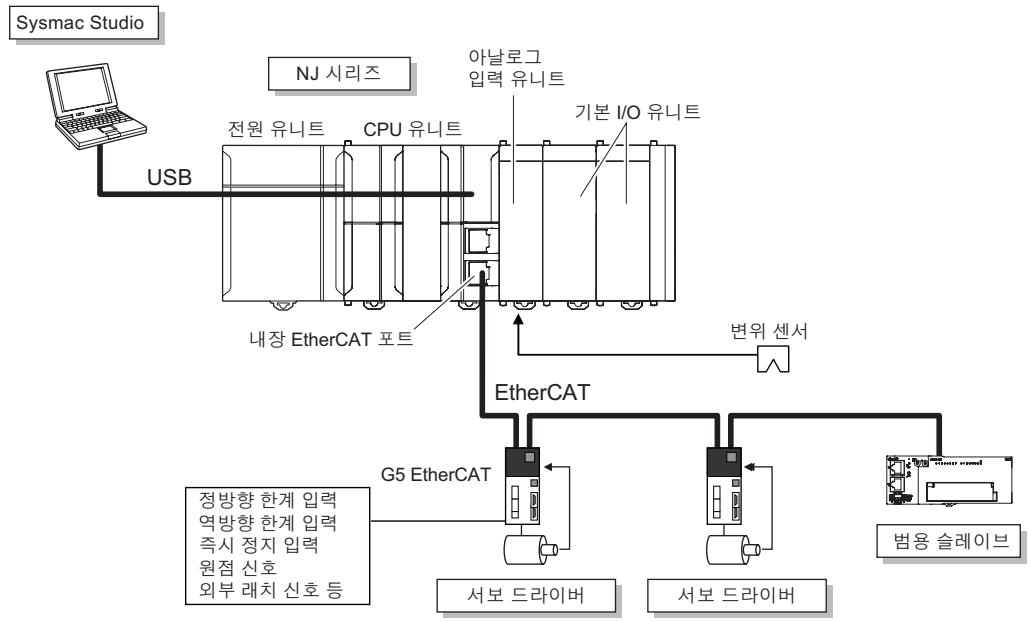
11-1 실제 사용 예 11-2
11-1-1 시스템 구성 11-2
11-1-2 동작 11-2
11-2 조작 순서 예의 개요 11-3
11-2-1 배선·설정 11-3
11-2-2 소프트웨어 설계 11-3
11-2-3 Sysmac Studio에서의 소프트웨어 설정 11-4
11-2-4 Sysmac Studio에서의 프로그래밍 11-8
11-2-5 Sysmac Studio에서의 시뮬레이션 11-9
11-2-6 실제 기기에 의한 동작 확인과 운전 11-9

11-1 실제 사용 예

여기에서는 NJ 시리즈의 실제 사용 예에 대해 설명합니다.

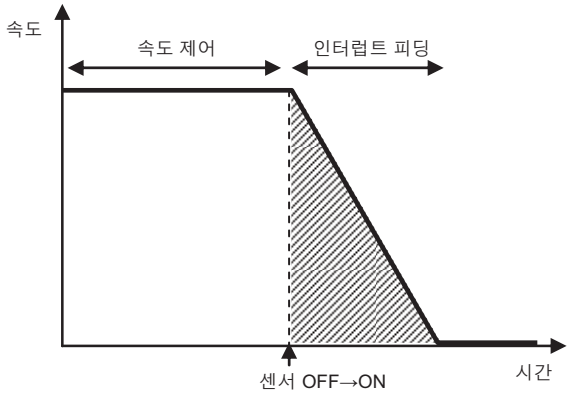
11-1-1 시스템 구성

유니트 이름		대수	연결 기기
전원 유니트		1대	–
CPU 유니트		1대	–
CJ 기본 I/O 유니트		2대	–
CJ 아날로그 입력 유니트		1대	변위 센서
EtherCAT 슬레이브	서보 드라이버 G5 EtherCAT 타입	2대	–
	I/O 터미널	1대	–



11-1-2 동작

속도 제어 중에 센서의 신호 OFF→ON에서 인터럽트 피딩을 합니다.



변위 센서에서의 아날로그 입력을 바탕으로 세로 방향의 위치를 변경합니다.

11-2 조작 순서 예의 개요

여기에서는 NJ 시리즈의 실제 조작 순서 예에 대해 설명합니다.

11-2-1 배선·설정

배선 및 하드웨어적인 설정을 합니다.

11-2-2 소프트웨어 설계

I/O 설계, 태스크 설계, POU 설계 및 변수 설계를 실행합니다.

I/O 설계

- 외부 I/O와 유니트의 구성 관계를 설계합니다.
- 어느 외부 I/O를 어느 정도의 간격으로 리프레시할지 설계합니다.

태스크 및 POU의 설계

다음 내용을 검토합니다.

- 태스크 구성을 어떻게 할 것인가?
- 어느 프로그램을 어떤 주기의 태스크에 할당할 것인가?
- 어느 유니트를 어느 태스크에 할당할 것인가?
- 어느 처리를 프로그램으로 하고, 어느 처리를 펑션 블록, 펑션으로 할 것인가?

변수 설계

다음 내용을 검토합니다.

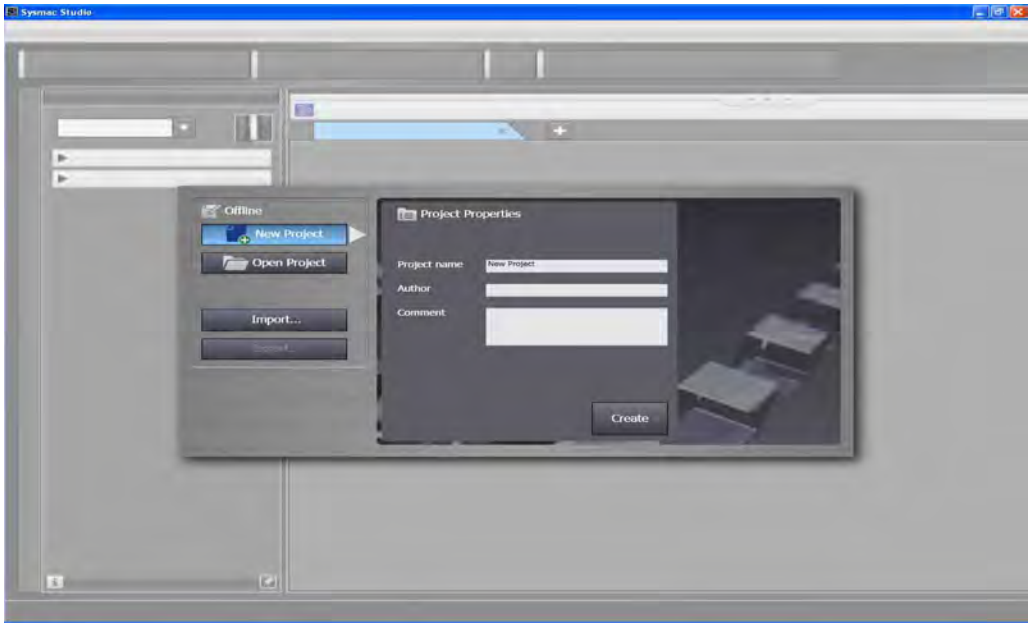
- POU 공통으로 사용하는 변수(글로벌 변수)와 각 POU에서만 사용하는 변수(로컬 변수)의 분류
- 슬레이브/유니트에 액세스하기 위한 디바이스 변수의 변수명 설계
- 변수의 이름, 유지 등의 각 속성 설계
- 변수의 데이터 형식 설계

11-2-3 Sysmac Studio에서의 소프트웨어 설정

Sysmac Studio에서 유니트/슬레이브 구성, 글로벌 변수/디바이스 변수의 등록, 축(축 변수) 작성 및 컨트롤러/고기능 유니트 설정을 합니다.

Sysmac Studio를 기동합니다

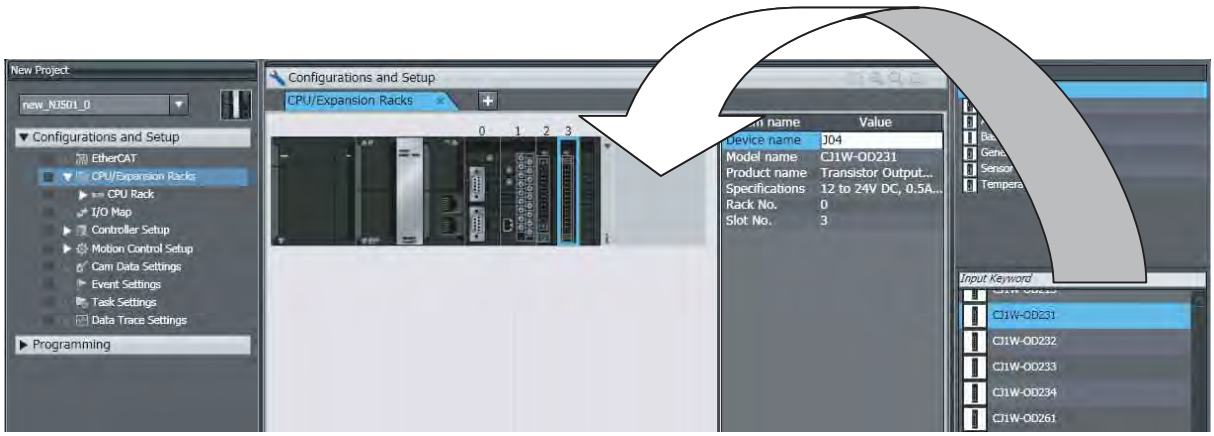
새 프로젝트를 작성합니다.



유니트를 구성합니다.

1 **[구성·설정]의 [CPU·증설 랙]을 두 번 클릭합니다.**

2 **유니트를 드래그 앤 드롭하여 유니트 구성을 작성합니다.**

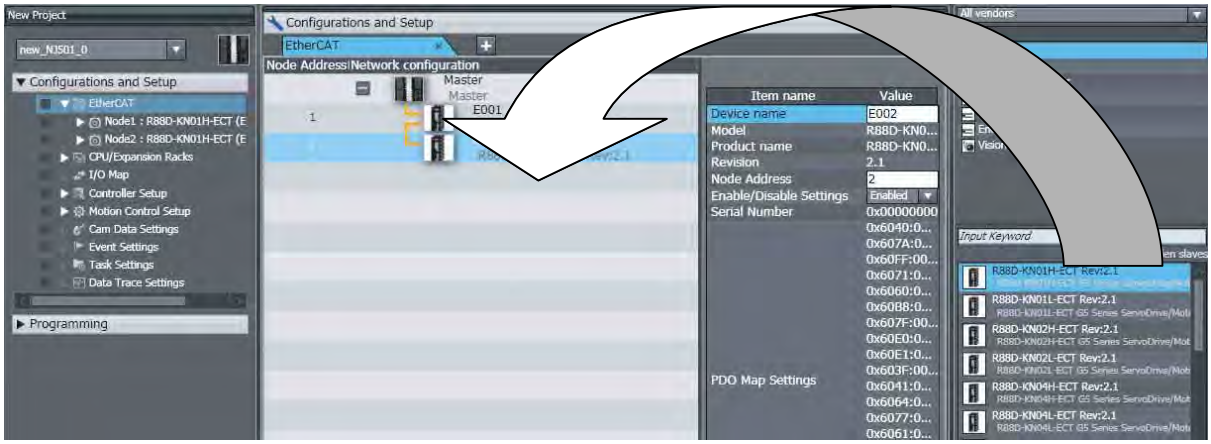


3 **각 유니트를 선택하여 유니트 설정을 합니다.**

EtherCAT 슬레이브를 구성합니다

1 **[구성·설정]의 [EtherCAT]을 두 번 클릭합니다.**

2 **슬레이브를 드래그 앤 드롭하여 슬레이브 구성을 작성합니다.**



3 **마스터를 선택하고 마스터 파라메터를 설정합니다.**

4 **각 슬레이브를 선택하고 슬레이브의 파라메터를 설정합니다.**

참고

이 시점에서 I/O 맵에서 강제값 리프레시를 실행하면 배선 확인을 실행할 수 있습니다.

글로벌 변수/디바이스 변수를 등록합니다

- **글로벌 변수의 등록**

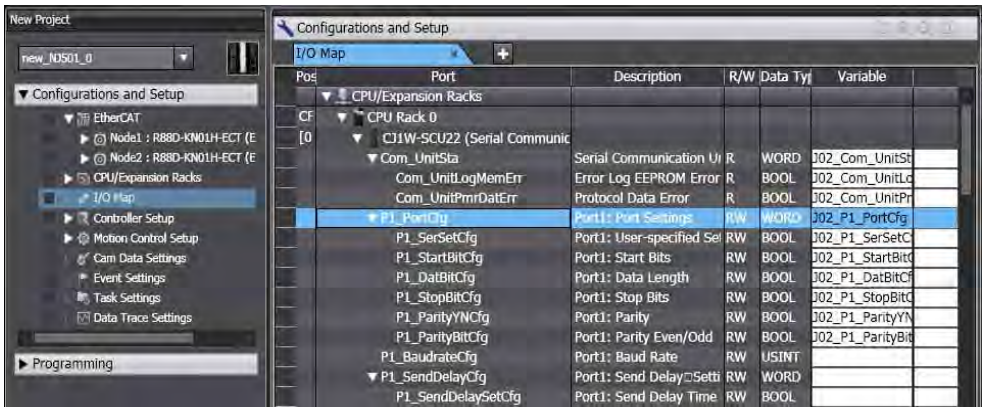
  1 **[프로그래밍]의 [데이터]의 [글로벌 변수]를 두 번 클릭합니다.**

  2 **글로벌 변수 테이블에 글로벌 변수를 등록합니다.**

- **디바이스 변수 등록**

  1 **[구성·설정]의 [IO 맵]을 두 번 클릭합니다.**

  2 **I/O 맵에서 I/O 포트(유니트/슬레이브 구성에 의해 자동으로 작성된)에 대해 변수를 할당합니다.**
  또한, Sysmac Studio에서 디바이스 변수명을 자동 생성할 수도 있습니다. 그 경우에 I/O 포트를 선택하여 마우스 오른쪽 버튼으로 클릭하고 [디바이스 변수 생성]을 선택합니다.

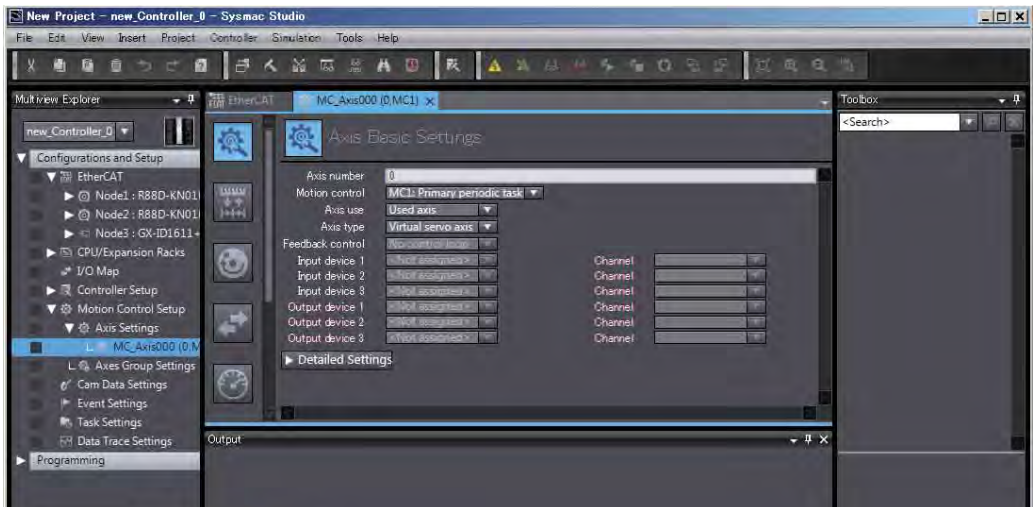


기본값에서는 생성한 디바이스 변수는 글로벌 변수 테이블에 등록됩니다.
필요에 따라 [변수 유형]을 '글로벌 변수'에서 각 POU의 로컬 변수(내부 변수)로 변경합니다.

'축'(축 변수)을 작성합니다

1 **[구성·설정]의 [모션 제어 설정]-[축 설정]을 선택하여 마우스 오른쪽 버튼으로 클릭하고 [추가]-[축 설정]을 선택합니다.**

2 **작성한 축(축 변수)에 대해 EtherCAT 구성상의 서보 드라이버를 할당합니다.**



- '축 사용'을 '사용 축'으로 선택합니다.
- NX 시리즈 CPU 유니트인 경우에 '모션 제어'를 '프라이머리 정주기 태스크'로 선택합니다.
- '축 유형'을 '서보 축'으로 선택합니다.
- '입력 장치', '출력 장치'에 슬레이브 구성에 등록한 EtherCAT 슬레이브를 선택합니다.

기타, 단위 변환, 동작 설정 등의 축 파라메터를 설정합니다.

컨트롤러 설정 및 고기능 유니트 설정을 합니다

- **PLC 모듈 관련 초기 설정:**
  컨트롤러 설정: 전원 투입 시 모드 등
- **고기능 유니트의 초기 설정:**
  유니트 구성·설정: 아날로그 입력 유니트의 초기 설정

11-2-4 Sysmac Studio에서의 프로그래밍

Sysmac Studio에서 프로그램 작성, 태스크 설정, 빌드를 합니다.

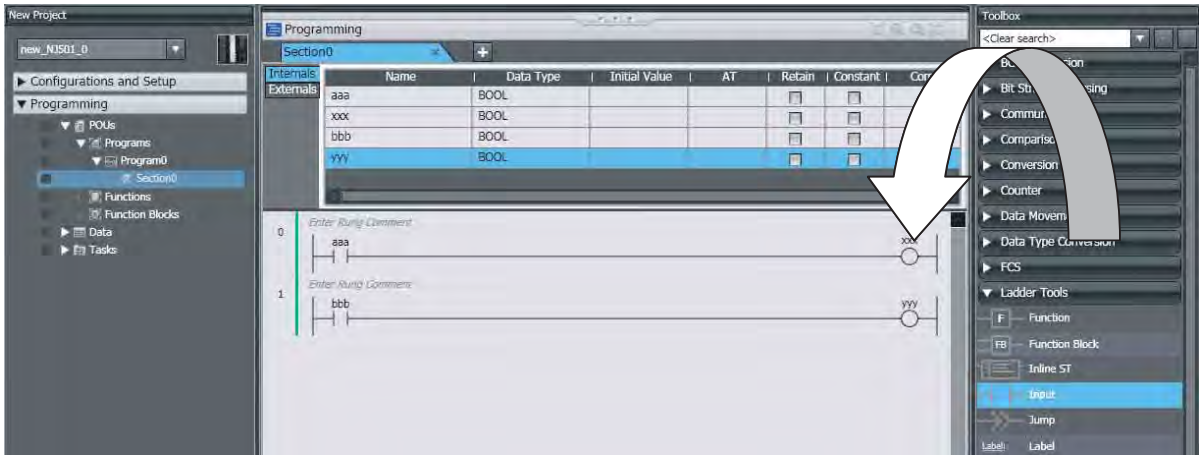
프로그래밍

1 [프로그래밍]의 [POU]의 [프로그램]을 선택해서 마우스 오른쪽 버튼으로 클릭하고 [추가]–[래더] 또는 [ST]를 선택합니다.

2 작성한 'プログラム'의 'Section□'를 두 번 클릭합니다.

3 각 프로그램의 로컬 변수를 등록합니다.

4 프로그램을 작성합니다.



다음 명령을 기술한 프로그램을 작성합니다.

- 원점 결정: MC_Home 명령
- 속도 제어: MC_MoveVelocity 명령
- 인터럽트 피딩: MC_MoveFeed 명령
- 위치 결정: MC_Move 명령

5 필요에 따라 [프로그래밍]의 [POU]의 [펑션] 또는 [펑션 블록]를 선택하고, [추가]–[래더] 또는 [ST]를 선택합니다.

작성한 'Function□' 또는 'FunctionBlock□'를 두 번 클릭합니다.
각 펑션 또는 펑션 블록의 로컬 변수를 등록합니다.
각 알고리즘을 작성합니다.
(주의) 래더인 경우 [R] 키를 눌러서 다음 회로를 작성합니다.

태스크 설정

[구성·설정]의 [태스크 설정]을 두 번 클릭합니다.

- [태스크 설정]에서 프라이머리 정주기 태스크의 '주기/실행 조건'의 태스크 주기를 풀다운 목록에서 선택합니다.
- [I/O를 제어하는 태스크 설정]에서 각 유니트/슬레이브를 할당할 태스크명을 선택합니다.
- [프로그램 할당 설정]에서 프로그램을 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당합니다.

빌드

[프로젝트] 메뉴에서 [빌드]를 선택합니다.

11-2-5 Sysmac Studio에서의 시뮬레이션

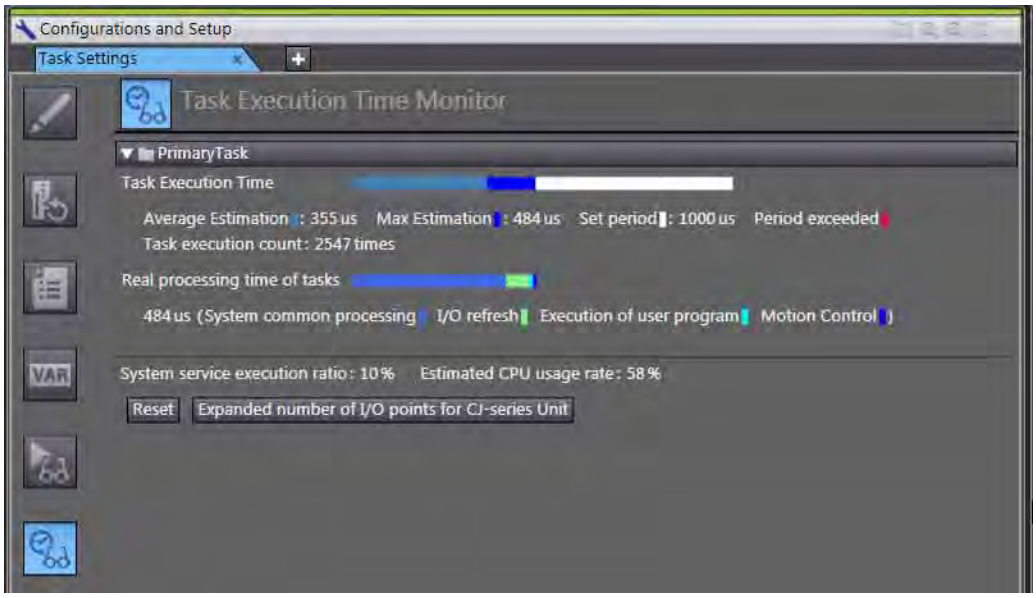
데스크탑 디버그를 실행합니다.
태스크 실행 시간 및 태스크 실제 처리 시간의 내역을 확인하고, 필요에 따라 태스크 설계를 재검토합니다.

시뮬레이터 기동·연결

[시뮬레이션] 메뉴에서 [실행 시간 예측 모드로 실행]을 선택합니다.
하드웨어 개정 번호가 표시되는 유니트에서는 해당하는 하드웨어 개정 번호를 선택합니다.
시뮬레이터(가상 컨트롤러)가 기동하고, 자동으로 온라인 연결됩니다.

시뮬레이터에서의 태스크 실행 시간 확인

[구성·설정]의 [태스크 설정]을 두 번 클릭합니다.
[태스크 실행 시간 모니터링]에서 태스크 실행 시간이 태스크 주기를 초과하는지 여부를 확인할 수 있습니다.



필요에 따라 태스크 구성, 프로그램 할당, 태스크 주기를 재검토합니다.

프로젝트 저장

[파일] 메뉴에서 [다른 이름으로 저장]을 선택합니다.

11-2-6 실제 기기에 의한 동작 확인과 운전

온라인 연결하여 프로젝트 다운로드, 배선 확인, 시운전한 후, 본 운전을 합니다.

온라인 연결

1. **NJ 시리즈 CPU 유니트의 전원을 투입합니다.**
2. **PC와 CPU 유니트를 USB 케이블로 연결합니다.**
3. **[컨트롤러] 메뉴에서 [통신 설정]을 선택합니다. [연결 방법]에서 연결 형태에 맞는 연결 방법을 선택합니다.**
4. **[컨트롤러] 메뉴에서 [온라인]을 선택합니다.**

프로젝트 다운로드('동기' 기능에 따라 다름)

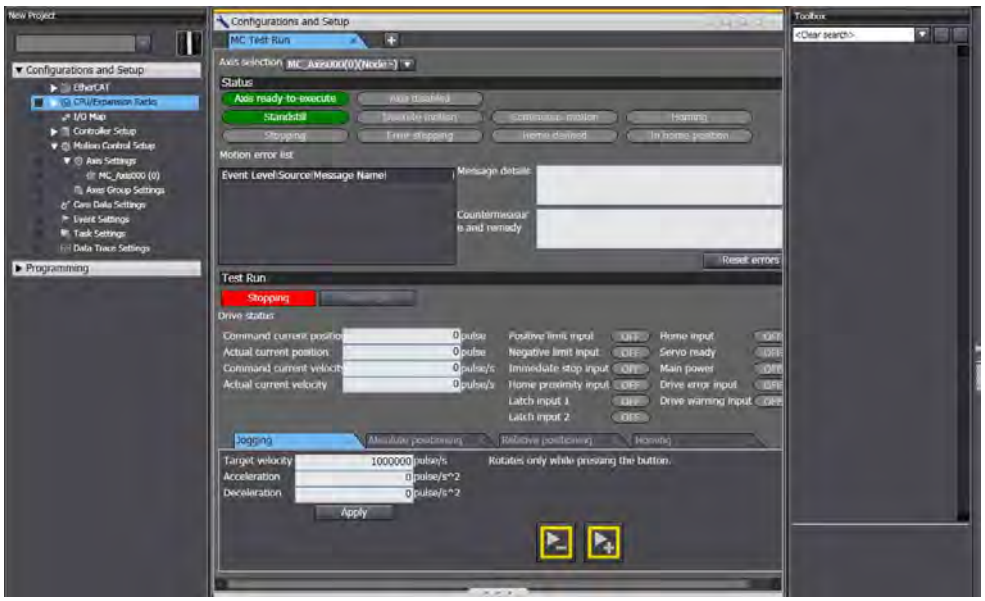
[컨트롤러] 메뉴에서 [동기]를 선택하여 프로젝트를 컨트롤러에 다운로드합니다.
(주의) '다운로드' 및 '업로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

배선 확인

I/O 맵 또는 래더 편집기에서 실제 I/O로의 강제값 리프레시 등을 실시하여 배선을 확인합니다.

MC 시운전

1. **MC 시운전 화면을 엽니다.**
2. **CPU 유니트를 프로그램 모드로 합니다.**
3. **화면에서 입력 신호를 모니터링하여 배선 확인을 실행합니다.**
4. **화면에서 조그 이송시킵니다.**



수동 운전

CPU 유니트를 운전 모드로 합니다.

- 서보 ON/OFF: 모션 제어 명령의 MC_Power를 실행
- 조그 이송: 모션 제어 명령의 MC_MoveJog를 실행

원점 결정

원점 결정: MC_Home 명령을 실행

본 운전

[컨트롤러] 메뉴에서 [동작 모드]-[운전 모드]를 선택합니다.
이상이 발생한 경우 원인를 조사하고 사용자 프로그램을 수정합니다.

12

이상의 확인과 처리

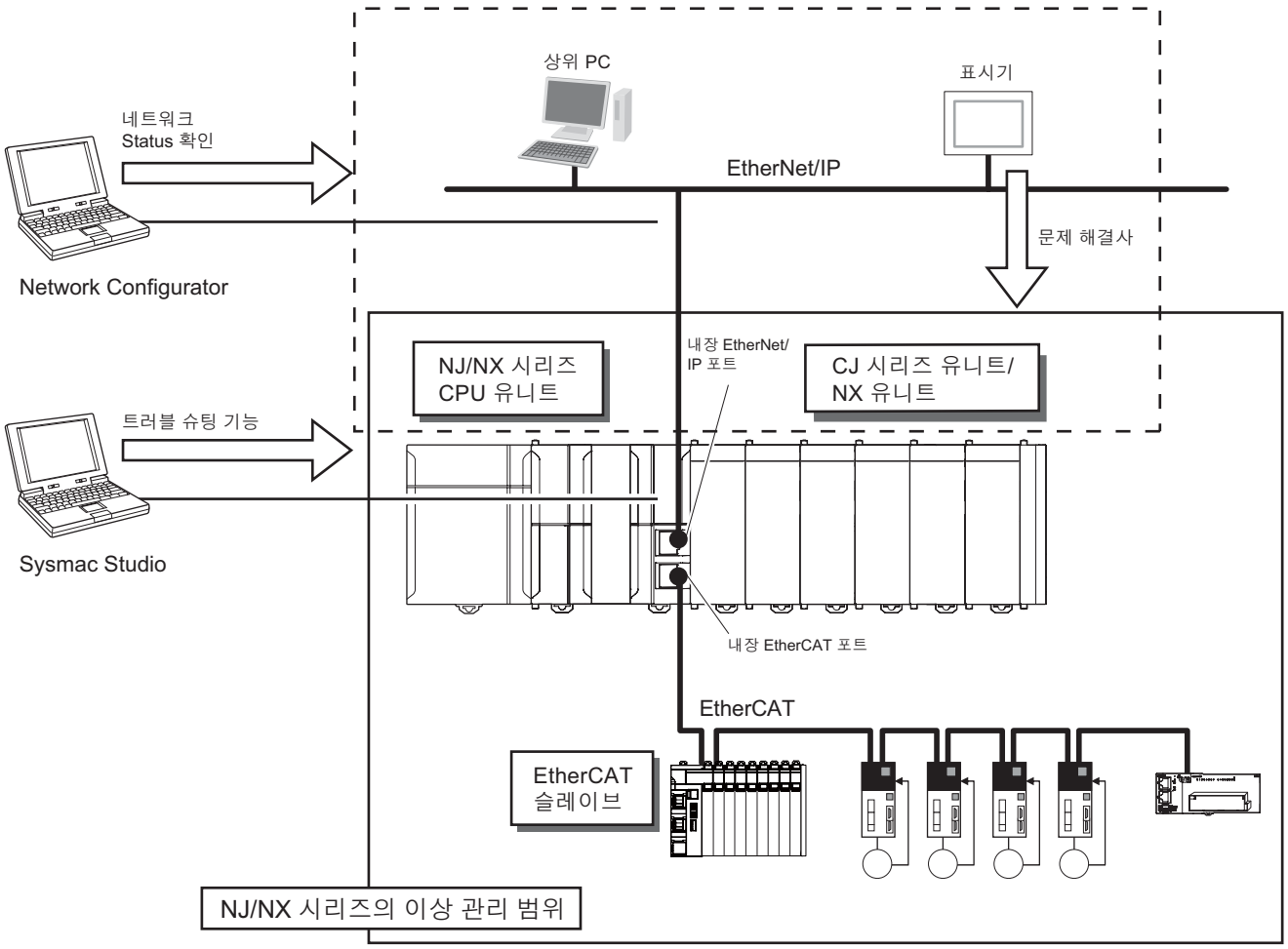
이상 확인 방법의 개요를 설명합니다.

12-1 이상의 확인과 처리의 개요 12-2

12-1 이상의 확인과 처리의 개요

NJ/NX 시리즈는 컨트롤러에서 발생한 이상을 이벤트로서 공통의 방법으로 일원적으로 관리합니다. 이것으로 관리 범위(CPU 유니트, NX유니트, NX 시리즈 슬레이브 터미널, EtherCAT 슬레이브*1, CJ 시리즈유니트)의 이상에 대해 공통의 방법으로 이상 내용과 대처 방법을 확인할 수 있습니다.

*1. Sysmac 장치이어야 합니다.



Sysmac Studio의 트러블 슈팅 기능 또는 표시기의 문제 해결사를 사용하면 이상 내용에서 대처 방법까지 확인할 수 있으므로 신속한 대응이 가능합니다.

이상의 종류 및 내용, 이상 발생 시의 구체적인 대처 방법 및 NJ/NX 시리즈 전체의 트러블 슈팅에 대해서는 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361V 이상)"를 참조하십시오.

A

부록

이 장에서는 CPU 유니트의 상세 사양, 태스크 실제 처리 시간, 시스템 정의 변수의 개별 사양 등 본문에 대한 보충 정보를 기술합니다.

A-1 사양 A-3
A-1-1 일반 사양 A-3
A-1-2 성능 사양 A-3
A-1-3 기능 사양 A-13

A-2 NX701에서의 태스크 실제 처리 시간의 기준 계산 A-19
A-2-1 태스크 실제 처리 시간의 평균값 계산 방법 A-20
A-2-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-27

A-3 NX102에서의 태스크 실제 처리 시간의 기준 계산 A-30
A-3-1 태스크 실제 처리 시간의 평균값 계산 방법 A-31
A-3-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-37

A-4 NX1P2에서의 태스크 실제 처리 시간의 기준 계산 A-40
A-4-1 태스크 실제 처리 시간의 평균값 계산 방법 A-41
A-4-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-47

A-5 NJ 시리즈에서의 태스크 실제 처리 시간의 기준 계산 A-50
A-5-1 태스크 실제 처리 시간의 평균값 계산 방법 A-51
A-5-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예 A-61

A-6 시스템 정의 변수 목록 A-64
A-6-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음) A-65
A-6-2 PLC 기능 모듈(카테고리명: _PLC) A-74
A-6-3 PLC 기능 모듈(카테고리명: _CJB) A-78
A-6-4 NX 버스 기능 모듈(카테고리명: _NXB) A-80
A-6-5 모션 제어 기능 모듈(카테고리명: _MC) A-82
A-6-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC) A-84
A-6-7 EtherNet/IP 기능 모듈(카테고리명: _EIP) A-89
A-6-8 이상 Status 각 비트의 의미 A-112

A-7 시스템 정의 변수의 개별 사양 A-114
A-7-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음) A-115
A-7-2 PLC 기능 모듈(카테고리명: _PLC) A-131
A-7-3 PLC 기능 모듈(카테고리명: _CJB) A-135
A-7-4 NX 버스 기능 모듈(카테고리명: _NXB) A-139
A-7-5 모션 제어 기능 모듈(카테고리명: _MC) A-142
A-7-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC) A-145
A-7-7 EtherNet/IP 기능 모듈(카테고리명: _EIP) A-154

A-8 CPU 유니트 내 데이터 속성 A-176

A-9 CJ 유니트용 메모리의 세부 내용 A-180
A-9-1 CIO 영역 A-180
A-9-2 내부 보조 릴레이 A-183
A-9-3 유지 릴레이 A-183
A-9-4 데이터 메모리 A-183
A-9-5 확장 데이터 메모리 A-184

A-10 변수의 메모리 확보 방법 A-185
A-10-1 변수의 메모리 확보 규칙 A-185
A-10-2 유의해야 하는 케이스 예 A-194

A-11 CX-Designer의 변수 테이블 등록 방법 A-198

A-12 EtherCAT 슬레이브 및 축 유효/무효 전환 A-201
A-12-1 EtherCAT 슬레이브 및 축을 사용하는 경우의 프로젝트 설정 A-201
A-12-2 EtherCAT 슬레이브 및 축 유효/무효 전환을 실행하는 명령 A-201
A-12-3 EtherCAT 슬레이브 및 축의 상태를 나타내는 시스템 정의 변수 . . . A-202
A-12-4 프로그램의 실행/정지 명령 A-203
A-12-5 프로그램의 실행/정지 확인 A-203
A-12-6 Sysmac Studio에서의 설정 A-204
A-12-7 EtherCAT 슬레이브 및 축 유효/무효 전환의 어플리케이션 예 A-205

A-13 사용자 프로그램의 크기에 관한 제한 A-208
A-13-1 사용자 프로그램 각 부품의 제한 A-208
A-13-2 사용자 프로그램 각 부품의 수 A-212

A-14 Ver.1.02 이하 CPU 유니트의 교환 방법 A-214
A-14-1 CPU 유니트 내의 데이터 업로드 A-214
A-14-2 새로운 CPU 유니트 연결 A-217
A-14-3 CPU 유니트로의 데이터 다운로드 A-218

A-15 NX 시리즈의 버전 관련 정보 A-222
A-15-1 유니트 버전과 Sysmac Studio 버전의 관계 A-222
A-15-2 유니트 버전 갱신에 따른 지원 기능의 추가와 변경 A-224

A-16 NJ 시리즈의 버전 관련 정보 A-226
A-16-1 유니트 버전과 Sysmac Studio 버전의 관계 A-226
A-16-2 하드웨어 개정 번호와 Sysmac Studio 버전의 관계 A-228
A-16-3 유니트 버전 갱신에 따른 지원 기능의 추가와 변경 A-229
A-16-4 유니트 버전의 갱신에 의한 성능 개선 A-233

A-17 설명서 번호 변경 전의 개정 이력 A-235

A-1 사양

여기에서는 NJ/NX 시리즈의 사양에 대해 설명합니다.

A-1-1 일반 사양

일반 사양에 대해서는 각 시리즈의 하드웨어 편을 참조하십시오.

- 🕮"NX 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-418)"
- 🕮"NX 시리즈 NX102 CPU 유니트 사용 설명서 하드웨어편(SBCA-462)"
- 🕮"NX 시리즈 NX1P2 CPU 유니트 사용 설명서 하드웨어편(SBCA-448)"
- 🕮"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"

A-1-2 성능 사양

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
처리 시간	명령 실행 시간	LD 명령		0.37ns~	3.3ns		1.2ns (~1.9ns)*1	2.0ns (~3.0ns)	3.3ns (~5.0ns)
		산술 명령 (배정도 실수형)		3.2ns~	70ns~		26ns~*2	42ns~	70ns~
프로그래밍	프로그램 용량*3	크기		80MB	5MB	1.5MB	20MB	5MB	3MB
		수량	POU 정의 수	6,000	3,000	450	3,000	750	450
			POU 인스턴스 수	48,000	9,000	1,800	9,000(*)	3,000(*)	1,800
	변수 용량	유지 속성 있음*4	크기	4MB	1.5MB	32KB*5	2MB	0.5MB	
			변수의 수	40,000	10,000	5,000	10,000	5,000(*)	
		유지 속성 없음*6	크기	256MB	32MB	2MB	4MB	2MB	
			변수의 수	360,000	90,000	90,000	90,000	22,500	
	데이터 형식	데이터 형식의 수		8,000	1,000	1,000	2,000	1,000	
	CJ 유니트용 메모리(변수의 AT 지정에 의해 지정 가능)	채널 I/O(CIO)		–	6144채널(0~6143)*7		6144채널(0~6143)		
		워크 릴레이(WR)		–	512채널(W0~ W511)*7		512채널(W0~W511)		
		유지 릴레이(HR)		–	1536채널(H0~H1535)*8		1536채널(H0~H1535)		
		데이터 메모리(DM)		–	32768채널 (D0~D32767)*8	16000채널 (D0~ D15999)*8	32768채널(D0~D32767)		
		확장 데이터 메모리 (EM)		–	32768채널x 25뱅크 (E0_00000~ E18_32767)*8	–	32768채널x25뱅크(E0_00000 ~ E18_32767)	32768채널x4뱅크 (E0_00000 ~ E3_32767)	
모션 제어				'CPU 유니트별 모션 제어의 성능 사양(A-10페이지)'을 참조하십시오.					
페리페럴(USB) 포트	지원 서비스			Sysmac Studio 연결	–	–	Sysmac Studio 연결		
	물리층			USB2.0 준수 B 커넥터	–	–	USB2.0 준수 B 커넥터		
	전송 거리			최대 5m	–	–	최대 5m		

항목		NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
내장 EtherNet/IP 포트	포트 수	2	2	1			
	물리층	10BASE-T/100BASE-TX/1000BASE-T	10BASE-T/100BASE-TX				
	프레임 길이	최대 1,514바이트					
	매체 액세스 방식	CSMA/CD					
	변조 방식	베이스밴드					
	토폴로지	스타형					
	전송 속도	1Gbps(1000BASE-T)	100Mbps(100BASE-TX)				
	전송 매체	트위스트 페어 케이블(실드 가공: STP): 카테고리 5, 5e 이상					
	전송 거리(허브와 노드 사이의 거리) 최대값	100m					
	캐스케이드 연결 최대 수	스위칭 허브 사용에 제한 없음					

항목			NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
내장 EtherNet/IP 포트	CIP 서비스: 태그 데이터 링크(사이클릭 통신)	최대 연결 수	256/포트 합계 512	32/포트 합계 64	32			
		패킷 간격*9	연결마다 설정 가능 0.5~10000 ms(0.5ms 단위)	연결마다 설정 가능 1~10000ms (1ms 단위)	연결마다 설정 가능 2~10000ms (1ms 단위)	연결마다 설정 가능 1~10000ms(1ms 단위)(*)		
		유니트 허용 통신 대역	40,000 pps*10 (하트비트 포함)	12,000pps*10(하트비트 포함)	3,000pps*10 (하트비트 포함)(*)			
		최대 태그 세트 수	256/포트 합계 512	32/포트 합계 40*11	32			
		태그 종류	네트워크 변수	네트워크 변수, CIO/WR/HR/DM		네트워크 변수, CIO/WR/HR/DM/EM		
		1연결(=1태그 세트)당 태그 수	8(태그 세트에 컨트롤러 Status가 포함되는 경우는 7)					
		최대 태그 수	256/포트 합계 512		256			
		1노드당 최대 링크 데이터 크기(전체 태그의 합계 크기)	369,664 바이트	19,200바이트				
		1연결당 최대 데이터 크기	1,444 바이트	600바이트				
		등록 가능한 최대 태그 세트 수	256/포트 합계 512 (1연결=1 태그 세트)	32/포트 합계 40*11 (1연결=1태그 세트)	32(1연결=1태그 세트)			
		1태그 세트의 최대 크기	1,444바이트(태그 세트에 컨트롤러 Status가 포함되는 경우는 2바이트 분량 사용)	600바이트(태그 세트에 컨트롤러 Status가 포함되는 경우는 2바이트 분량 사용)				
		멀티 캐스트 패킷 필터 기능*12	가능					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
내장 EtherNet/IP 포트	CIP 메시지 서비스 : Explicit 메시지	Class3(연결 수)		128/포트 합계 256 (클라이언트+서버)	32/포트 합계 64 (클라이언트+서버)	32(클라이언트+서버)			
		UCMM (비연결형)	동시 통신 가능한 최대 클라이언트 수	32/포트 합계 64		32			
			동시 통신 가능한 최대 서버 수	32/포트 합계 64		32			
	TCP 소켓 수			30(*)	60	30(*)			
	OPC UA 서버*13	지원 프로파일·모델		–	UA 1.02 Micro Embedded Device Server Profile PLCopen Information Model	–	UA 1.02 Micro Embedded Device Server Profile PLCopen Information Model	–	
		기본값 엔드 포인트·포트		–	opc.tcp://192.168.250.1:4840/	–	opc.tcp://192.168.250.1:4840/	–	
		최대 세션(클라이언트) 수		–	5	–	5	–	
		서버 전체의 최대 모니터링 항목 수		–	2000	–	2000	–	
		모니터링 항목의 Sampling Rate(ms)		–	0*14 , 50, 100, 250, 500, 1000, 2000, 5000, 10000	–	0*14, 50, 100, 250, 500, 1000, 2000, 5000, 10000	–	
		서버 전체의 최대 Subscription 수		–	100	–	100	–	
		공개할 수 있는 최대 변수 수		–	10,000	–	10,000	–	
		공개 변수의 Value Attribute 수		–	10,000	–	10,000	–	
		공개 가능한 구조체 정의 수		–	100	–	100	–	
		공개할 수 없는 변수의 제약		–	*15	–	*15	–	
		보안 정책·모드		–	*16	–	*16	–	
		어플리케이션 인증	인증 방법	–	X.509 준수	–	X.509 준수	–	
			저장 가능한 증명서 수	–	신뢰할 수 있는 증명서: 32 발행자 증명서: 32 거부 목록: 32	–	신뢰할 수 있는 증명서: 32 발행자 증명서: 32 거부 목록: 32	–	
		사용자 인증	인증 방법	–	다음 설정이 가능 사용자 이름/비밀번호 익명 (Anonymous)	–	다음 설정이 가능 사용자 이름/비밀번호 익명 (Anonymous)	–	

항목		NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
내장 EtherCAT 포트	통신 규격	IEC 61158 Type12					
	EtherCAT 마스터 사양	Class B 지원(Feature Pack Motion Control 지원)					
	물리층	100BASE-TX					
	변조 방식	베이스밴드					
	전송 속도	100Mbps(100BASE-TX)					
	Duplex 모드	Auto					
	토폴로지	라인, 데이지 체인, 분기 배선					
	전송 매체	카테고리 5 이상, 트위스트 페어 케이블(알루미늄 테이프와 편조 이중 차폐 실드 케이블, 스트레이트 권장)					
	노드 간 거리 최대값	100m					
	최대 슬레이브 수	512	64	16	192		64
	설정할 수 있는 노드 주소 범위	1~512	1~192				
	프로세스 데이터의 최대 크기	IN: 11,472바이트 OUT: 11,472바이트 단, 8프레임(최대 프로세스 데이터 프레임 수)의 범위 이내일 것	IN: 5,736바이트 OUT: 5,736바이트 단, 4프레임(최대 프로세스 데이터 프레임 수)의 범위 이내일 것	IN: 1,434바이트 OUT: 1,434바이트 단, 1프레임(최대 프로세스 데이터 프레임 수)의 범위 이내일 것	IN: 5,736바이트 OUT: 5,736바이트 단, 4프레임(최대 프로세스 데이터 프레임 수)의 범위 이내일 것		
	1슬레이브 최대 크기	IN: 1,434바이트 OUT: 1,434바이트					
	통신 주기	• 프라이머리정주기 태스크 125µs 250µs~8ms(250µs 단위로 설정 가능) • 정주기 태스크(실행 우선도 5) 125µs 250µs~100ms(250µs 단위로 설정 가능)	1,000µs~32,000µs(250µs 단위로 설정 가능)	2,000µs~8,000µs(250µs 단위로 설정 가능)	500/1,000/2,000/4,000µs(*)		1,000/2,000/4,000µs
	동기 지터(Jitter)	1µs 이하					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
시리얼 통신	통신 방식			–		반이중(시리얼 통신용 옵션 보드 연결 시)	–		
	동기 방식			–		비동기(시리얼 통신용 옵션 보드 연결 시)	–		
	전송 속도			–		1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2kbps(시리얼 통신용 옵션 보드 연결 시)	–		
유니트 구성	장착할 수 있는 유니트 수	1랙(CPU 랙 또는 증설 랙)에 장착할 수 있는 CJ 유니트의 최대 수		–			10		
		1랙(CPU 랙)당 최대 NX 유니트 수		–	32	8	–		
		시스템 전체의 최대 CJ 유니트 수		–			40		
		시스템 전체의 최대 NX 유니트 수		4,096 (EtherCAT 슬레이브 터미널 상)	400 (EtherCAT 슬레이브 터미널 상)	24 (CPU 랙 상: 8, EtherCAT 슬레이브 터미널 상: 16)	4,096 (EtherCAT 슬레이브 터미널 상)		400 (EtherCAT 슬레이브 터미널 상)
	증설 랙 최대 수			0			3		
	입출력 점수	CJ 유니트 최대 입출력 점수		–			2,560		
	전원(CPU 랙 또는 증설 랙)	형식		NX-PA9001 NX-PD7001	DC 입력의 비절연 전원을 CPU 유니트에 내장		NJ-P□3001		
		전원 OFF 확정 시간	AC 전원	30~45ms		–	30~45ms		
			DC 전원	5~20ms	2~8ms		22~25ms		
	NX 버스의 I/O 데이터의 최대 크기			–	IN: 8,192바이트 OUT: 8,192바이트		–		
옵션 보드	슬롯 수			–		*17	–		
내장 I/O	입력	점수		–		*18	–		
	출력	점수		–		*19	–		
		부하 단락 보호 기능		–		*20	–		
내장 시계	정밀도			주위 온도 55℃: 월차 -3.5~+0.5분 주위 온도 25℃: 월차 -1.5~+1.5분 주위 온도 0℃: 월차 -3분~+1분					
	내장 커패시터 백업 시간			–	주위 온도 40℃: 10일		–		

*1. 하드웨어 개정이 'A'인 유니트의 경우에는 '1.1ns(~1.7ns)'가 됩니다.
*2. 하드웨어 개정이 'A'인 유니트의 경우에는 '24ns~'가 됩니다.
*3. 실행 개체, 변수 테이블(변수명 등)의 용량입니다.
*4. CJ 유니트용 메모리의 HR, DM, EM은 포함되지 않습니다.
*5. CJ 유니트용 메모리가 포함됩니다.
*6. CJ 유니트용 메모리의 CIO, WR은 포함되지 않습니다.
*7. 유지 속성이 없는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.

*8. 유지 속성이 있는 변수를 사용합니다. 1채널 단위로 설정할 수 있습니다.
*9. 노드 수에 좌우되지 않고 설정된 주기에 따라 회선상의 데이터를 갱신합니다.
*10.pps는 Packet Per Second를 의미하며, 1초간 처리 가능한 송수신 패킷 수를 나타냅니다.
*11. 합계 수가 40을 초과하는 태그 세트가 설정된 경우에는 '태그 데이터 링크 태그 세트 설정 수 초과'(840E0000Hex) 이벤트가 발생합니다.
*12. EtherNet/IP 포트에는 IGMP 클라이언트가 장착되어 있으므로 IGMP Snooping 호환 스위칭 허브를 사용하여 불필요한 멀티 캐스트 패킷을 필터링할 수 있습니다.
*13.NJ 시리즈에서 OPC UA를 지원하는 것은 NJ501-1□00의 Ver.1.17 이상입니다.
*14.0은 50ms가 지정된 것으로 간주합니다.
*15.다음 변수를 공개할 수 없습니다.
변수 크기가 1024바이트를 초과하는 변수
이차원 이상의 구조체 배열
이차원 이상의 배열을 포함하는 구조체
중첩이 4 이상인 구조체
공용체
배열의 첨자가 0으로 시작하지 않는 배열
요소 수가 1024를 초과하는 배열
멤버 수가 100을 초과하는 구조체
*16.다음을 선택할 수 있습니다.
None
Sign - Basic128Rsa15
Sign - Basic256
Sign - Basic256Sha256
SignAndEncrypt - Basic128Rsa15
SignAndEncrypt - Basic256
SignAndEncrypt - Basic256Sha256
*17.NX1P2-□□40□□□: 2, NX1P2-□□24□□□: 1
*18.NX1P2-□□40□□□: 24, NX1P2-□□24□□□: 14
*19.NX1P2-□□40□□□: 16, NX1P2-□□24□□□: 10
*20.NX1P2-□□□□□□: 없음(NPN), NX1P2-□□□□□□1: 있음(PNP)

(주의) 표에 (*)가 표시된 항목은 버전 업그레이드로 성능 개선된 기능입니다. 버전 업그레이드에 대한 정보는 'A-15 NX 시리즈의 버전 관련 정보(A-222페이지)' 및 'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

● CPU 유니트별 모션 제어의 성능 사양

항목			NX701-	
			17□□	16□□
제어축 수*1	최대 제어축 수		256축	128축
		모션 제어축	256축	128축
		단축 위치 제어축	−	
	사용하는 실제 축의 최대 수		256축	128축
		사용 모션 제어 서보 축	256축	128축
		사용 단축 위치 제어 서보 축	−	
	직선 보간 제어 최대 수		1축 그룹당 4축	
	원호 보간 제어 축 수		1축 그룹당 2축	
축 그룹 최대 수			64그룹	
모션 제어 주기			EtherCAT 통신의 프로세스 데이터 통신 주기와 동일	
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점	
		전체 캠 테이블의 최대 점수	1,048,560점	
	캠 테이블 최대 테이블 수		640테이블	
위치 단위			펄스, mm, μm, nm, degree, inch	
오버라이드			0.00, 0.01~500.00%	

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

항목			NX102-			
			12□□	11□□	10□□	90□□
제어축 수*1	최대 제어축 수		15축			4축
		모션 제어축	11축			−
		단축 위치 제어축	4축			
	사용하는 실제 축의 최대 수		12축	8축	6축	4축
		사용 모션 제어 서보 축	8축	4축	2축	−
		사용 단축 위치 제어 서보 축	4축			
	직선 보간 제어 최대 수		1축 그룹당 4축			−
	원호 보간 제어 축 수		1축 그룹당 2축			−
축 그룹 최대 수			8그룹			−
모션 제어 주기			EtherCAT 통신의 프로세스 데이터 통신 주기와 동일			
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점			
		전체 캠 테이블의 최대 점수	262,140점			
	캠 테이블 최대 테이블 수		160테이블			
위치 단위			펄스, mm, μm, nm, degree, inch			
오버라이드			0.00, 0.01~500.00%			

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

항목			NX1P2-		
			11□□□□□	10□□□□□	90□□□□□
제어축 수*1	최대 제어축 수		12축	10축	4축
		모션 제어축	8축	6축	−
		단축 위치 제어축	4축		
	사용하는 실제 축의 최대 수		8축	6축	4축
		사용 모션 제어 서보 축	4축	2축	−
		사용 단축 위치 제어 서보 축	4축		
	직선 보간 제어 최대 수		1축 그룹당 4축		−
	원호 보간 제어 축 수		1축 그룹당 2축		−
축 그룹 최대 수			8그룹		−
모션 제어 주기			프라이머리 정주기 태스크의 주기와 동일함		
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점		−
		전체 캠 테이블의 최대 점수	262,140점		−
	캠 테이블 최대 테이블 수		80테이블		−
위치 단위			펄스, mm, μm, nm, degree, inch		
오버라이드			0.00, 0.01~500.00%		

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

항목			NJ501-		
			□5□□	□4□□	□3□□
제어축 수*1	최대 제어축 수		64축	32축	16축
		모션 제어축	64축	32축	16축
		단축 위치 제어축	−		
	사용하는 실제 축의 최대 수		64축	32축	16축
		사용 모션 제어 서보 축	64축	32축	16축
		사용 단축 위치 제어 서보 축	−		
	직선 보간 제어 최대 수		1축 그룹당 4축		
	원호 보간 제어 축 수		1축 그룹당 2축		
축 그룹 최대 수			32그룹		
모션 제어 주기			EtherCAT 통신의 프로세스 데이터 통신 주기와 동일		
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점		
		전체 캠 테이블의 최대 점수	1,048,560점		
	캠 테이블 최대 테이블 수		640테이블		
위치 단위			펄스, mm, μm, nm, degree, inch		
오버라이드			0.00, 0.01~500.00%		

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

항목			NJ301-	
			12□□	11□□
제어축 수*1	최대 제어축 수		15축(*)	
		모션 제어축	15축(*)	
		단축 위치 제어축	–	
	사용하는 실제 축의 최대 수		8축	4축
		사용 모션 제어 서보 축	8축	4축
		사용 단축 위치 제어 서보 축	–	
	직선 보간 제어 최대 수		1축 그룹당 4축	
	원호 보간 제어 축 수		1축 그룹당 2축	
축 그룹 최대 수			32그룹	
모션 제어 주기			EtherCAT 통신의 프로세스 데이터 통신 주기와 동일	
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점	
		전체 캠 테이블의 최대 점수	262,140점	
	캠 테이블 최대 테이블 수		160테이블	
위치 단위			펄스, mm, μm, nm, degree, inch	
오버라이드			0.00, 0.01~500.00%	

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

(주의) 표에 (*)가 표시된 항목은 버전 업그레이드로 성능 개선된 기능입니다. 버전 업그레이드에 대한 정보는 📖'A-15 NX 시리즈의 버전 관련 정보(A-222페이지)' 및 📖'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

항목			NJ101-	
			10□□	90□□
제어축 수*1	최대 제어축 수		6축	–
		모션 제어축	6축	–
		단축 위치 제어축	–	
	사용하는 실제 축의 최대 수		2축	–
		사용 모션 제어 서보 축	2축	–
		사용 단축 위치 제어 서보 축	–	
	직선 보간 제어 최대 수		1축 그룹당 4축	–
	원호 보간 제어 축 수		1축 그룹당 2축	–
축 그룹 최대 수			32그룹	–
모션 제어 주기			EtherCAT 통신의 프로세스 데이터 통신 주기와 동일	–
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점	–
		전체 캠 테이블의 최대 점수	262,140점	–
	캠 테이블 최대 테이블 수		160테이블	–
위치 단위			펄스, mm, μm, nm, degree, inch	–
오버라이드			0.00, 0.01~500.00%	–

*1. 각 축에 대한 설명은 📖"NJ/NX 시리즈 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

A-1-3 기능 사양

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
태스크 기능	기능			실행 조건과 실행 우선도를 지정하는 '태스크' 단위로 I/O 리프레시 및 사용자 프로그램을 실행하는 기능					
		정주기로 실행하는 태스크	최대 프라이머리 정주기 태스크 수	1					
			최대 정주기 태스크 수	4	2		3		
		조건 성립 시 실행할 태스크(*)	최대 이벤트 태스크 수	32					
			실행 조건	이벤트 태스크 기동 명령 실행 시, 변수의 조건식 일치 시					
	설정	시스템 서비스 감시 설정		–			시스템 서비스(CPU 유니트가 태스크 처리와 별도로 실행하는 처리)의 실행 간격 및 사용자 프로그램 실행 시간 전체에 대한 비율을 감시하는 기능		
프로그래밍 기능	POU (Program Organization Unit)	프로그램		태스크에 할당하는 단위의 POU					
		펑션 블록		상태가 있는 부품을 작성할 때 사용하는 POU					
		펑션		연산 처리 등 입력에 대해 출력이 전적으로 결정되는 부품을 작성할 때 사용하는 POU					
	프로그램 언어	종류		래더도[*1], 스트럭쳐 텍스트(ST)					
	이름 공간(*)			이름으로 POU 정의의 그룹 분류를 수행하는 기능					
	변수	변수의 외부 참조 기능	네트워크 변수	표시기나 상위 PC, 다른 컨트롤러 등에서의 액세스를 허가하는 기능					
	데이터 형식	기본 데이터 형식	부울 형식	BOOL					
			비트 열 형식	BYTE, WORD, DWORD, LWORD					
			정수 형식	INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT					
			실수 형식	REAL, LREAL					
			지속 시간 형식	TIME					
			날짜 형식	DATE					
			시각 형식	TIME_OF_DAY					
			날짜 시각 형식	DATE_AND_TIME					
			문자열 형식	STRING					
		파생 데이터 형식		구조체 형식, 공용체 형식, 열거 형식					
		구조체 형식	기능	데이터 형식이 다른 여러 개의 데이터를 하나로 묶어서 취급하는 기능					
			최대 멤버 수	2,048					
			최대 중첩 수	8					
			멤버의 데이터 형식	기본 데이터 형식, 구조체 형식, 공용체 형식, 열거 형식, 배열 변수					
			멤버의 오프셋 지정(*)	구조체 멤버를 임의의 메모리 위치에 배치하는 기능					
		공용체 형식	기능	동일한 데이터에 대해 여러 개의 다른 데이터 형식으로 액세스할 수 있도록 하는 기능					
			최대 멤버 수	4					
			멤버의 데이터 형식	BOOL, BYTE, WORD, DWORD, LWORD					
		열거 형식	기능	변수의 값을 '열거자'라고 하는 레이블(문자열)로 표현한 기능					
	데이터 형식의 속성	배열 지정	기능	동일한 데이터 형식의 요소를 묶어서 선두부터의 번호(첨자)로 지정하는 기능					
			최대 차원 수	3					
			최대 요소 수	65,535					
			FB 인스턴스의 배열 지정	가능					
		범위 지정		미리 정해진 범위 내의 값만 취할 수 있도록 명시하는 기능					
	라이브러리(*)			(사용자)라이브러리 사용 가능					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
모션 제어 기능*2	제어 모드			위치 제어, 속도 제어, 토크 제어					
	축 종류			서보 축, 가상 서보 축, 엔코더 축, 가상 엔코더 축					
	관리할 수 있는 위치			명령 위치, 피드백 위치					
	단축	단축 위치 제어	절대값 위치 결정	절대 좌표의 목표 위치를 지정하여 위치를 결정하는 기능					
			상대값 위치 결정	명령 현재 위치에서의 이동 거리를 지정하여 위치를 결정하는 기능					
			인터럽트 피딩 위치 결정	외부 입력에 의한 인터럽트 입력이 발생한 위치에서의 이동 거리를 지정하여 위치를 결정하는 기능					
			주기적 동기 절대 위치 제어(*)	위치 제어 모드에서 제어 주기마다 명령 위치를 출력하는 기능					
		단축 속도 제어	속도 제어	위치 제어 모드에서 속도를 제어하는 기능					
			주기적 동기 속도 제어	속도 제어 모드에서 제어 주기마다 속도 명령을 출력하는 기능					
		단축 토크 제어	토크 제어	모터의 토크를 제어하는 기능					
		단축 동기 제어	캠 동작 시작	지정한 캠 테이블을 사용하여 캠 동작을 시작하는 기능					
			캠 동작 해제	입력 파라메터로 지정한 축의 캠 동작을 종료하는 기능					
			기어 동작 시작	주축과 종축 간의 기어비를 설정하여 기어 동작을 수행하는 기능					
			위치 지정 기어 동작	주축과 종축 간의 기어비와 동기하는 위치를 설정하여 기어 동작을 수행하는 기능					
			기어 동작 해제	실행 중인 기어 동작, 위치 지정 기어 동작을 중지하는 기능					
			사다리꼴 패턴 캠	지정한 주축에 동기하여 위치를 결정하는 기능					
			주축 상대값 위상 보정	동기 제어 중인 주축의 위상을 보정하는 기능					
			가감산 위치 결정	2축의 명령 위치를 가산한 값 또는 감산한 값을 명령 위치로 하여 출력하는 기능					
		단축 수동 조작	운전 가능	서보 드라이버의 상태를 서보 ON 상태로 전환하여 축 동작을 가능하게 하는 기능					
			조그 이송	지정한 목표 속도에 따라 조그 이송을 수행하는 기능					
		단축 제어 보조	축 에러 리셋	축의 이상을 해제하는 기능					
			원점 복귀	모터를 작동시키고, 한계 신호, 원점 근방 신호, 원점 신호를 사용하여 기계 원점을 결정하는 기능					
			파라메터 지정 원점 복귀(*)	파라메터를 지정하여 모터를 작동시키고, 한계 신호, 원점 근방 신호, 원점 신호를 사용하여 기계 원점을 결정하는 기능					
			고속 원점 복귀	절대 좌표의 '0'을 목표 위치로 하여 위치 결정을 실행하고 원점으로 복귀하는 기능					
			강제 정지	축을 감속 정지하는 기능					
			즉시 정지	축을 즉시 정지하는 기능					
			오버라이드 값 설정	축의 목표 속도를 변경하는 기능					
			현재 위치 변경	축의 명령 현재 위치와 피드백 현재 위치를 임의의 값으로 변경하는 기능					
			외부 래치 유효	트리거 발생에 의해 축의 위치를 기록하는 기능					
			외부 래치 무효	실행 중인 래치를 무효로 하는 기능					
			존 감시	축의 명령 위치 또는 피드백 현재 위치가 지정된 범위(존) 안에 존재하는지 판정하는 기능					
			디지털 캠 스위치 유효(*)	축의 위치에 따라 디지털 출력을 ON 또는 OFF하는 기능					
			축간 편차 감시	지정한 2축의 명령 위치 또는 피드백 위치의 차이가 허용값을 초과하는지 감시하는 기능					
			편차 카운터 리셋	명령 현재 위치와 피드백 현재 위치 사이의 편차를 0으로 하는 기능					
			토크 제한	서보 드라이버의 토크 제한 기능의 유효/무효의 전환과 토크 제한값을 설정하여 출력 토크를 제한하는 기능					
			명령 위치 보정	동작 중인 축에 대해 위치를 보정하는 기능					
			기동 속도(*)	축 동작을 시작할 때의 초기 속도를 설정하는 기능					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
모션 제어 기능*2	축 그룹	다축 협조 제어	절대값 직선 보간	절대 위치를 지정하여 직선 보간을 수행하는 기능					
			상대값 직선 보간	상대 위치를 지정하여 직선 보간을 수행하는 기능					
			2축 원호 보간	2축의 원호 보간을 수행하는 기능					
			축 그룹 주기적 동기 절대 위치 제어(*)	위치 제어 모드에서 제어 주기마다 위치 명령을 출력하는 기능					
		다축 협조 제어 보조	축 그룹 에러 리셋	축 그룹 및 축의 이상을 해제하는 기능					
			축 그룹 유효	축 그룹의 동작을 유효로 설정하는 기능					
			축 그룹 무효	축 그룹의 동작을 무효로 설정하는 기능					
			축 그룹 강제 정지	보간 동작 중인 모든 축을 감속 정지하는 기능					
			축 그룹 즉시 정지	보간 동작 중인 모든 축을 즉시 정지하는 기능					
			축 그룹 오버라이드 값 설정	보간 동작 중의 합성 목표 속도를 변경하는 기능					
			축 그룹 위치 취득(*)	축 그룹의 명령 현재 위치와 피드백 현재 위치를 취득하는 기능입니다.					
			축 그룹 구성축 쓰기(*)	축 그룹 파라메터의 [구성축]을 일시적으로 재기입하는 기능					
	공통	캠	캠 테이블 속성 갱신	입력 파라메터에서 지정한 캠 테이블의 종점 인덱스를 갱신하는 기능					
			캠 테이블 저장	입력 파라메터에서 지정한 캠 테이블을 CPU 유니트 내의 비휘발성 메모리에 저장하는 기능					
			캠 테이블 생성(*)	입력 파라메터에서 지정한 캠 속성과 캠 노드로 캠 테이블을 생성하는 기능					
		파라메터	MC 설정 쓰기	축 파라메터 및 축 그룹 파라메터의 일부를 일시적으로 재기입하는 기능					
			축 파라메터의 변경(*)	축 파라메터를 사용자 프로그램에서 참조·변경하는 기능					
	보조 기능	카운트 모드		리니어 모드(유한 길이) 또는 로터리 모드(무한 길이) 중에서 선택 가능					
		단위 변환		각 축의 표시 단위를 기계에 맞춰서 설정 가능					
		가감속 제어	자동 가감속 제어	축이나 축 그룹 동작 시의 가감속 곡선을 저크로 설정하는 기능					
			가감속 속도 변경	가감속 동작 중에도 가감속 속도를 변경하는 기능					
		인포지션 확인		위치 결정 완료를 확인하기 위한 인포지션 폭과 인포지션 확인 시간을 설정하는 기능					
		정지 방법 선택		즉시 정지 입력 신호나 한계 입력 신호가 유효로 된 경우의 정지 방법을 설정하는 기능					
		모션 제어 명령의 재기동		실행 중인 모션 제어 명령의 입력 변수를 변경해서 재기동하여 동작 중에 목표값을 변경하는 기능					
		모션 제어 명령의 다중 기동(버퍼 모드)		동작 중에 다른 모션 제어 명령을 기동한 경우의 실행 시작 타이밍과 동작 간의 속도 연결 방법을 지정하는 기능					
		축 그룹 동작의 연속 동작 (트랜지션 모드)		축 그룹 동작의 다중 기동을 사용한 연속 동작 방법을 지정하는 기능					
		감시 기능	소프트웨어 리미트	축의 동작 범위를 감시하는 기능					
			위치 편차	축의 명령 현재값과 피드백 현재값 사이의 위치 편차를 감시하는 기능					
			속도/가감속 속도/토크/보간 속도/보간 가감속 속도	축마다 또는 축 그룹마다 경고값을 설정하여 감시하는 기능					
		앱솔루트 엔코더 대응		OMRON 서보 드라이버 1S 시리즈 또는 G5 시리즈의 앱솔루트 엔코더 장착 모터를 사용하여 전원 투입 시의 원점 복귀가 필요하지 않게 하는 기능					
		입력 신호의 논리 반전(*)		즉시 정지 입력 신호, 정방향 한계 입력 신호, 역방향 한계 입력 신호, 원점 근방 입력 신호의 논리를 반전하는 기능					
	외부 I/F 신호			서보 드라이버 측의 다음 입력 신호를 사용 가능 원점 신호, 원점 근방 신호, 정방향 한계 신호, 역방향 한계 신호, 즉시 정지 신호, 인터럽트 입력 신호					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
유니트(입출력) 관리 기능	EtherCAT 슬레이브	최대 슬레이브 수		512	64	16	192		64
	CJ 시리즈 유니트	최대 유니트 수		–			40		
통신 기능	페리페럴(USB) 포트			PC상의 각종 지원 소프트웨어와 통신하는 기능	–		PC상의 각종 지원 소프트웨어와 통신하는 기능		
	EtherNet/IP 포트	통신 프로토콜		TCP/IP, UDP/IP					
		CIP 통신 서비스	태그 데이터 링크	프로그램 없이 EtherNet/IP 네트워크의 장치와 주기적으로 데이터를 교환하는 기능					
			메시지 통신	EtherNet/IP 네트워크의 장치와 임의의 CIP 명령을 송수신하는 기능					
		TCP/IP 어플리케이션	소켓 서비스	UDP 또는 TCP 프로토콜로 Ethernet의 임의의 노드와 임의의 데이터를 송수신하는 기능, 소켓 통신용 명령에 의해 실행하는 기능					
			FTP 클라이언트(*)	CPU 유니트에서 Ethernet의 다른 컴퓨터나 컨트롤러에 FTP를 사용하여 데이터를 전송하는 기능, FTP 클라이언트 통신 명령에 의해 실행					
			FTP 서버	Ethernet의 다른 컴퓨터에서 컨트롤러의 CPU 유니트의 SD 메모리 카드 내의 파일을 읽고 쓰는 기능					
			시계 자동 조정	지정한 시각 또는 CPU 유니트의 전원이 투입된 후 지정한 일정 시간 간격으로 NTP 서버에서 시간 정보를 취득하여 CPU 유니트의 내부 시계 정보를 갱신하는 기능					
			SNMP 에이전트	SNMP 관리자를 사용한 네트워크 관리 소프트웨어에 내장 EtherNet/IP 포트의 내부 상태 정보를 제공하는 기능					
		OPC UA*3	서버 기능	–	OPC UA 네트워크상의 클라이언트로부터의 요구에 대해 응답하는 기능	–	OPC UA 네트워크상의 클라이언트로부터의 요구에 대해 응답하는 기능	–	
통신 기능	EtherCAT 포트	지원 서비스	프로세스 데이터 통신	CoE에서 정의된 EtherCAT 마스터와 슬레이브 간에서 제어 정보를 일정 주기의 사이클릭 통신으로 데이터 교환하는 통신 방식					
			SDO 통신	CoE에서 정의된 EtherCAT 마스터와 슬레이브 간에서 제어 정보를 일정하지 않은 주기의 이벤트 통신으로 데이터 교환하는 통신 방식					
		네트워크 스캔		연결된 슬레이브 기기의 정보를 읽어서 슬레이브 구성을 자동 생성하는 기능					
		DC(Distributed Clock)		모든 EtherCAT 장치(마스터 포함)에서 동일한 'EtherCAT System Time'을 공유하여 시각 동기를 실행하는 기능					
		패킷 모니터링		마스터가 송신한 프레임과 수신한 프레임을 저장하기 위한 기능 저장한 데이터는 WireShark 등으로 볼 수 있음(*)					
		슬레이브 유효/무효 설정		슬레이브를 통신 대상으로 하여 유효로 할지, 무효로 할지 여부를 설정하는 기능					
		슬레이브의 이탈/재가입		슬레이브의 교환 등의 보수를 위해 EtherCAT 네트워크에서 해당 슬레이브를 일시적으로 이탈/재가입시키는 기능					
		지원 어플리케이션 프로토콜	CoE	슬레이브에 EtherCAT에서 CAN 어플리케이션의 메시지(SDO)를 송신하는 기능					
	시리얼 통신	프로토콜		–		상위 링크(FINS), 무수순, Modbus-RTU 마스터 (시리얼 통신용 옵션 보드 연결 시)	–		
	통신용 명령			FTP 클라이언트 명령, CIP 통신 명령, 소켓 통신용 명령, SDO 메시지 명령, 무수순 통신 명령, Modbus RTU 프로토콜 명령(*), Modbus TCP 프로토콜 명령 *4			FTP 클라이언트 명령(*). CIP 통신 명령, 소켓 통신용 명령, SDO 메시지 명령, 무수순 통신 명령, 프로토콜 매크로 명령, Modbus RTU 프로토콜 명령(*)		

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
운전 관리 기능	운전 중 출력 접점			운전 모드 시, 전원 유니트의 접점이 ON이 되는 기능	–		운전 모드 시, 전원 유니트의 접점이 ON이 되는 기능		
시스템 관리 기능	이벤트 로그	기능		이벤트 발생을 기록하는 기능					
		최대 건수	시스템 이벤트 로그	2,048	768*5	576*6	1,024	512	
			액세스 이벤트 로그	1,024	576*7	528*8	1,024	512	
			사용자 이벤트 로그	1,024	512	512	1,024	512	
디버그 기능	온라인 편집			프로그램, 펑션 블록, 펑션, 글로벌 변수를 온라인 편집을 통해 변경하는 기능, 네트워크를 경유하여 여러 작업자가 개별 POU를 변경할 수 있음					
	강제값 리프레시			특정 접점을 강제값으로 리프레시하는 기능					
		최대 점수	EtherCAT 슬레이브 디바이스 변수	64					
			CJ 유니트 디바이스 변수, AT(할당대상) 지정된 변수	–			64		
	MC 시운전*2			모터의 동작이나 배선 확인을 Sysmac Studio에서 실행하는 기능					
	동기			온라인 연결 시 Sysmac Studio의 프로젝트 파일과 CPU 유니트의 데이터를 동일하게 만드는 기능					
	미분 모니터링(*)			접점의 상승 또는 하강을 모니터링하는 기능					
		최대 점수		8					
	데이터 트레이스	종류	트리거 트레이스	트리거가 성립된 후, 설정한 샘플링 수에 도달하면 트레이스를 자동으로 정지하는 기능					
			연속 트레이스	데이터 트레이스의 실행을 계속하고 트레이스 데이터를 Sysmac Studio가 상시 수집하는 기능					
		최대 동시 기동 수		4	2	2	4	2	
		최대 레코드 수		10,000					
		최대 샘플링 점수		192개 변수	48개 변수	48개 변수	192개 변수	48개 변수	
		샘플링 타이밍		지정 태스크 주기, 지정 시간, 샘플링 명령 실행 시					
		트리거 트레이스		트리거 조건을 설정하여 사건이 발생한 전후의 데이터를 기록하는 기능					
			트리거 조건	• BOOL 형식 변수의 상승/하강 • BOOL 형식 변수 이외의 상수값 비교 비교 방법: =, >, ≧, <, ≦, ≠					
			딜레이값	트리거 성립 전/성립 후의 샘플링 수 비율을 설정하는 기능					
	시뮬레이션 기능			Sysmac Studio에서 CPU 유니트의 동작을 시뮬레이션하는 기능					
높은 신뢰 기능	자체 진단	컨트롤러 이상	중요도	전체 정지 이상, 부분 정지 이상, 가벼운 이상, 감시 정보, 일반 정보					
			최대 메시지 언어 수	9(Sysmac Studio) 2(NS 시리즈 표시기)					
		사용자 이상	기능	임의의 이상을 설계해서 사전에 등록하고 명령을 실행하여 기록을 남기는 기능					
			중요도	8단계					
			최대 메시지 언어 수	9					

항목				NX701-□□□□	NX102-□□□□	NX1P2-□□□□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
보안 기능	고객 자산 보호/잘못된 조작 방지	CPU 유니트 이름 기능 및 시리얼 ID 기능		Sysmac Studio에서 온라인으로 연결할 때 프로젝트의 CPU 유니트 이름과 연결대상 CPU 유니트의 CPU 유니트 이름이 일치하는지 여부를 확인하는 기능					
		보호 기능	사용자 프로그램 복원 정보 없는 전송 기능	Sysmac Studio에서 CPU 유니트 내에 존재하는 데이터를 읽을 수 없도록 하는 기능					
			CPU 유니트에 대한 쓰기 방지 기능	Sysmac Studio/SD 메모리 카드에서 CPU 유니트 내에 존재하는 데이터를 쓸 수 없도록 하는 기능					
			프로젝트 파일 전체 보호 기능	Sysmac Studio에서 비밀번호를 사용하여 .smc 파일을 여는 것을 금지하는 기능					
			데이터 보호 기능(*)	Sysmac Studio에서 비밀번호를 사용하여 POU를 보호하는 기능					
		조작 권한 인증 기능		조작 실수에 의해 장치나 사람에게 위해를 가할 가능성이 있는 경우에 조작 권한에 맞춰 온라인에서의 조작 기능을 제한하는 기능					
			그룹 수(*)	5					
		사용자 프로그램 실행용 ID 인증 기능		특정 하드웨어(CPU 유니트)에 대해 Sysmac Studio에서 사용자 프로그램 실행용 ID를 입력하지 않는 한 사용자 프로그램을 실행할 수 없도록 하는 기능					
SD 메모리 카드 기능	스토리지 종류			SD 카드, SDHC 카드					
	어플리케이션	SD 메모리 카드 자동 전송 기능(*)		컨트롤러의 전원 투입 시, SD 메모리 카드의 autoload 디렉토리에 저장된 데이터를 컨트롤러에 다운로드하는 기능					
		SD 메모리 카드 프로그램 전송 기능(*)		시스템 정의 변수에서의 지시에 따라 SD 메모리 카드에 저장된 사용자 프로그램을 컨트롤러로 전송하는 기능					
		SD 메모리 카드 조작 명령		사용자 프로그램의 명령으로 SD 메모리 카드에 액세스하는 기능					
		Sysmac Studio에서의 파일 조작		사용자는 SD 메모리 카드에 컨트롤러용 파일 뿐만 아니라 범용 문서 파일 등도 저장하고 읽을 수 있음					
		SD 메모리 카드의 수명 감지 기능		SD 메모리 카드의 수명을 시스템 정의 변수 또는 이벤트 로그로 통지하는 기능					
백업 기능(*)	SD 메모리 카드 백업 기능	조작 방법	CPU 유니트 전면 스위치	CPU 유니트의 전면 스위치를 조작하여 백업, 비교, 복원을 수행하는 기능					
			시스템 정의 변수를 통한 지시	시스템 정의 변수를 조작하여 백업이나 비교를 수행하는 기능					
			Sysmac Studio 메모리 카드 화면	Sysmac Studio의 SD 메모리 카드 조작 화면에서 백업 또는 비교를 수행하는 기능					
			전용 명령(*)	전용 명령으로 백업을 수행하는 기능					
		보호 기능	SD 메모리 카드에 대한 백업 금지	SD 메모리 카드에 대한 백업을 금지하는 기능					
	Sysmac Studio 컨트롤러 백업 기능			Sysmac Studio를 사용하여 장치의 백업, 복원, 비교를 수행하는 기능					

*1. 인라인 ST(래더도 중에 ST 언어를 기술하는 래더도 언어의 요소)가 가능합니다.
*2. NJ101-90□□에서는 사용할 수 없습니다.
*3. NJ 시리즈에서 OPC UA를 지원하는 것은 NJ501-1□00의 Ver.1.17 이상입니다.
*4. NX102-□□□□에서만 사용할 수 있습니다.
*5. CPU 유니트분 512건과 NX 유니트분 256건의 합계입니다.
*6. CPU 유니트분 512건과 NX 유니트분 64건의 합계입니다.
*7. CPU 유니트분 512건과 NX 유니트분 64건의 합계입니다.
*8. CPU 유니트분 512건과 NX 유니트분 16건의 합계입니다.

(주의) 표의 (*)가 표시된 항목은 버전 업그레이드로 추가된 기능입니다. 버전 업그레이드에 대한 정보는 'A-15 NX 시리즈의 버전 관련 정보(A-222페이지)' 및 'A-16 NJ 시리즈의 버전 관련 정보(A-226페이지)'를 참조하십시오.

A-2 NX701에서의 태스크 실제 처리 시간의 기준 계산

여기서는 NX701에서의 태스크 실제 처리 시간의 평균값 기준 계산 방법에 대해 설명합니다.

태스크 실제 처리 시간과 태스크 실행 시간은 실제 기기 컨트롤러에서 확인해야 합니다. 세부 내용은 '5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'을 참조하십시오.

안전상의 중요 사항

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간이 변동됩니다.
최종적으로는 반드시 실제 기기를 사용하여 모든 조건에서 성능 확인을 실시하고, 태스크 주기 초과가 발생하지 않는지, 적절한 통신 응답성이 확보되는지 확인하십시오.

사용상의 주의 사항

정주기 태스크(실행 우선도 5)의 태스크 실제 처리 시간의 평균값을 계산하는 경우에 Sysmac Studio를 사용하여 EtherCAT 구성을 작성한 후, EtherCAT 구성 편집 화면에서 표시한 [PDO 통신 주기 2: 전송 지연 시간]의 값이 필요합니다.

참고

정주기 태스크는 실행 우선도가 그보다 높은 태스크의 실행 중에는 일시 정지될 수 있습니다. '태스크 실제 처리 시간'에 일시 정지한 시간은 포함하지 않습니다. 일시 정지한 시간을 포함하여 태스크 시작부터 종료까지 실제로 걸린 시간을 나타내는 것이 '태스크 실행 시간'입니다. '태스크 실제 처리 시간'과 '태스크 실행 시간'의 차이에 대한 세부 내용은 ' 태스크 실행 시간과 태스크 실제 처리 시간의 의미(5-107페이지)'를 참조하십시오.

A-2-1 태스크 실제 처리 시간의 평균값 계산 방법

태스크 실제 처리 시간의 평균값은 I/O 리프레시 처리 시간, 사용자 프로그램 실행 시간, 모션 제어 처리 시간, 공통 처리 시간의 합계입니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 처리 시간+사용자 프로그램 실행 시간
+모션 제어 처리 시간+공통 처리 시간

각 처리 내용은 다음과 같습니다.

○: 있음, −: 없음

처리		처리 내용	프라이머리 정주기 태스크	정주기 태스크 (실행 우선도 5)	정주기 태스크 (실행 우선도 16)	정주기 태스크 (실행 우선도 17, 18)
I/O 리프레시 처리		EtherCAT 슬레이브와의 리프레시를 실행합니다.	○	○	−	−
사용자 프로그램 실행		태스크에 할당된 프로그램을 할당 순으로 실행합니다.	○	○	○	○
모션 제어 처리		• 사용자 프로그램으로부터의 모션 제어 명령을 실행합니다. • 모션 출력 처리	○	○	−	−
공통 처리 시간	시스템 공통 처리 1	• 변수의 갱신 처리(참조 태스크의 경우) • 모션 입력 처리 • 데이터 트레이스 처리	○	○	○	○
	시스템 공통 처리 2	• 변수의 갱신 처리(갱신 태스크의 경우) • 태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서 변수 액세스 처리	○	○	○	○
	시스템 오버헤드 시간	상기 이외의 시스템 공통 처리	○	○	○	○

아래에 각 처리 시간 계산의 기준을 표시합니다.

I/O 리프레시 처리 시간

I/O 리프레시 처리 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간=EtherCAT 슬레이브 처리 시간

EtherCAT 슬레이브 처리 시간은 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에서 계산식이 다릅니다.
다음으로 각각의 태스크에서의 EtherCAT 슬레이브 처리 시간을 구하는 방법을 설명합니다.

사용상의 주의 사항

정주기 태스크(실행 우선도 5)를 구하는 경우에 먼저 프라이머리 정주기 태스크의 태스크 주기를 결정해두어야 합니다.

참고

EtherCAT 슬레이브를 할당하지 않은 태스크에서는 EtherCAT 슬레이브 처리 시간이 '0'입니다.

● 프라이머리 정주기 태스크에서의 EtherCAT 슬레이브 처리 시간

프라이머리 정주기 태스크에서의 EtherCAT 슬레이브 처리 시간은 다음 식으로 구할 수 있습니다.

EtherCAT 슬레이브 처리 시간[μs] = 0.0006×pDout + 0.0001×pDin + 0.082×pDinout
+ (1.24×Snum + 0.01×Clen) + 25.58

pDout	:	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
pDin	:	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브의 입력 프로세스 데이터 크기 합계[바이트]
pDinout	:	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브별 출력 프로세스 데이터 크기와 입력 프로세스 데이터의 크기 중 큰 쪽의 합계[바이트]
Snum	:	내장 EtherCAT 포트에 연결한 EtherCAT 슬레이브의 전체 대수[대]
Clen	:	내장 EtherCAT 포트에 연결한 케이블의 총 길이[m]

● 정주기 태스크(실행 우선도 5)에서의 EtherCAT 슬레이브 처리 시간

정주기 태스크(실행 우선도 5)에서의 EtherCAT 슬레이브 처리 시간은 다음 식으로 구할 수 있습니다.

EtherCAT 슬레이브 처리 시간[μs] = 0.0006×pDout + (다음 A와 B 중 큰 쪽)
+ sTsend + 0.0001×sDin + 16

A[μs] = ((0.0623×sDout + 37)×pTcycle)÷(pTcycle - (0.0243×pDout+35))
B[μs] = 0.082×pDinout

pDout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
sTsend*1	:	Sysmac Studio에서 표시한 PDO 통신 주기 2의 전송 지연 시간[μs]
sDin	:	정주기 태스크(실행 우선도 5)에 할당한 EtherCAT 슬레이브의 입력 프로세스 데이터 크기 합계[바이트]
sDout	:	정주기 태스크(실행 우선도 5)에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
pTcycle	:	프라이머리 정주기 태스크의 태스크 주기[μs]
pDinout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브별 출력 프로세스 데이터 크기와 입력 프로세스 데이터의 크기 중 큰 쪽의 합계[바이트]

*1. sTsend(전송 지연 시간)는 계산식으로 구할 수 없습니다. Sysmac Studio를 사용하여 EtherCAT 구성을 작성한 후에 EtherCAT 구성 편집 화면에서 표시한 [PDO 통신 주기 2: 전송 지연 시간]의 값을 sTsend(전송 지연 시간)에 대입하십시오. Sysmac Studio에서의 [PDO 통신 주기 2: 전송 지연 시간]의 표시 방법에 대해서는 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

사용자 프로그램 실행 시간

사용자 프로그램 실행 시간은 명령 종류×명령 수에 좌우됩니다.

기준으로는 크게 다음 3가지로 분류한 명령 수를 계측 또는 예상합니다.
- 표준적인 명령
- 배정도 실수(LREAL) 형식 사칙연산 명령
- 배정도 실수(LREAL) 형식 삼각 함수 명령

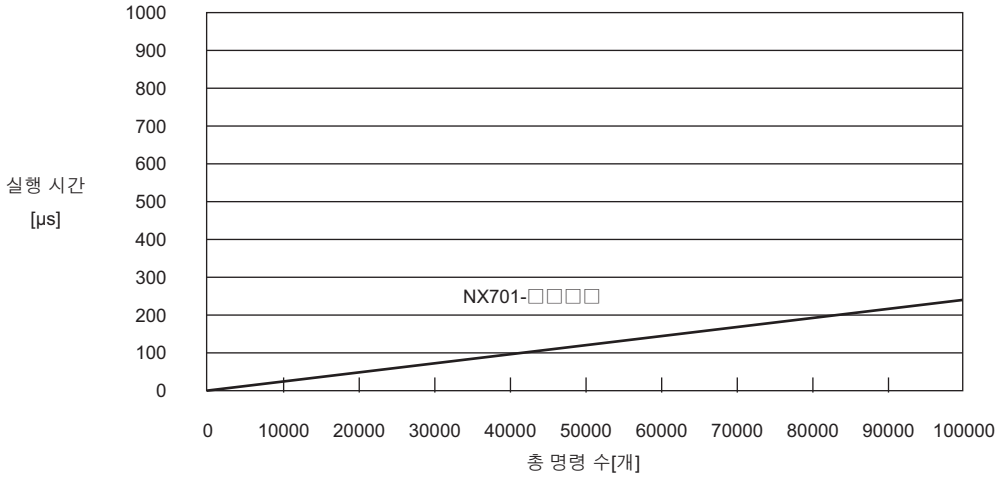
또한, 래더도 언어와 ST 언어에서는 사용하는 명령이 다릅니다. 각각의 명령 구성에 대해서는 "래더도 언어의 표준적인 명령'의 명령 종류 구성(A-23페이지)' 및 "ST 언어의 표준적인 명령'의 명령 종류 구성(A-25페이지)'을 참조하십시오.

● 간단한 계산

각 분류의 명령 수를 기준으로 다음 그래프에서 다음 각 명령의 실행 시간을 읽고 합계를 구합니다.
- 표준적인 명령의 실행 시간
- 배정도 실수(LREAL) 형식 사칙연산 명령의 실행 시간
- 배정도 실수(LREAL) 형식 삼각 함수 명령의 실행 시간

이를 통해 사용자 프로그램의 실행 시간을 추측할 수 있습니다.

래더도 언어의 표준적인 명령의 실행 시간

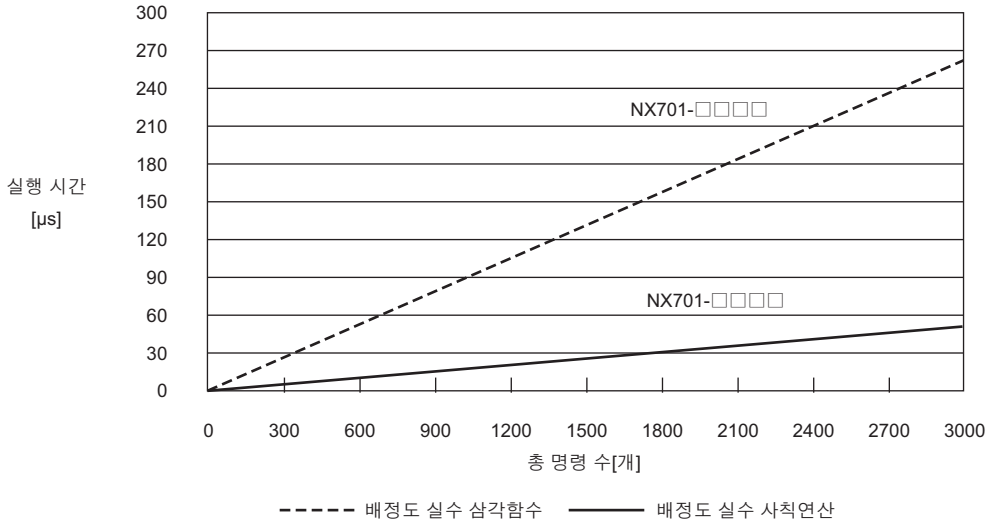


- '래더도 언어의 표준적인 명령'의 명령 종류 구성

이 구성에서의 명령 실행률은 20%입니다.

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
래더도 명령	LD, AND, OUT, SET, RESET	81.0	40.2
비교 명령	EQ, LT	4.1	8.3
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	1.6	7.3
산술 명령	+, -, *, /, ADD, SUB, MUL, DIV	2.4	6.5
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.2
비트 열 연산 명령	AND, OR	6.2	13.0
데이터 전송 명령	MOVE	4.6	23.5
합계		100.0	100.0

래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



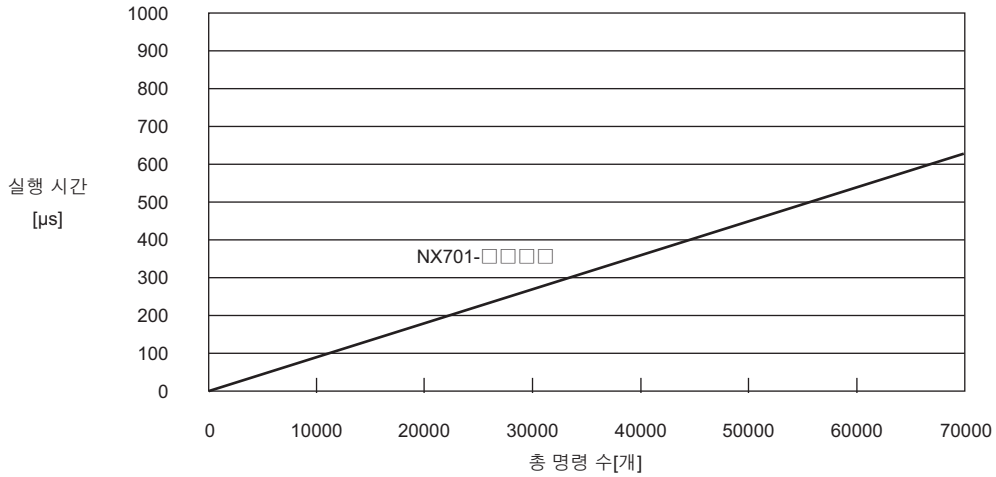
- '배정도 실수 사칙연산'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

- '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율 [%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

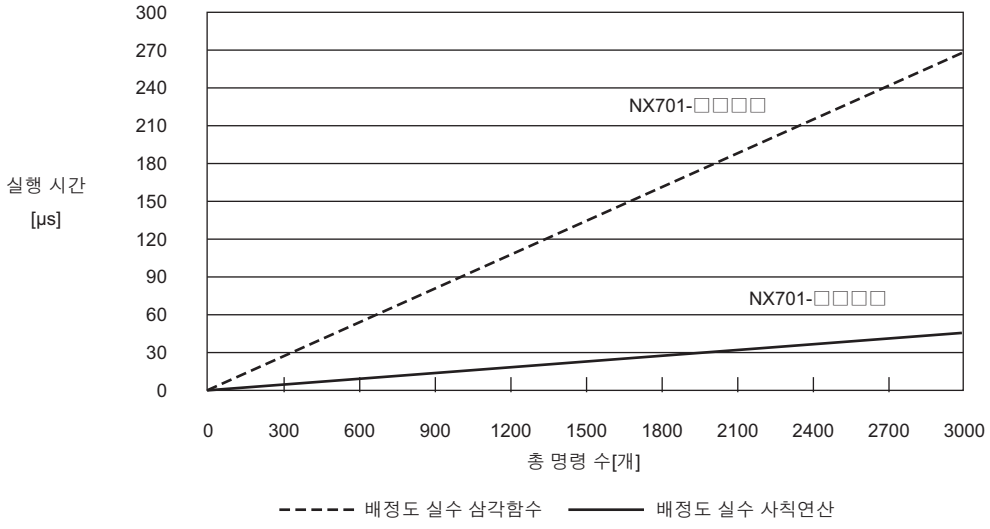
ST 언어의 표준적인 명령의 실행 시간



• 'ST 언어의 표준적인 명령'의 명령 종류 구성

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
ST 구문 명령	IF ELSIF END_IF	75.4	41.6
비교 명령	EQ, LT	5.2	8.7
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	2.1	18.8
산술 명령	+, -, *, /	3.1	10.2
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.6
비트 열 연산 명령	AND, OR	8.0	11.7
데이터 전송 명령	:=	5.9	7.3
합계		100.0	100.0

ST 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



• '배정도 실수 사칙연산'의 구성

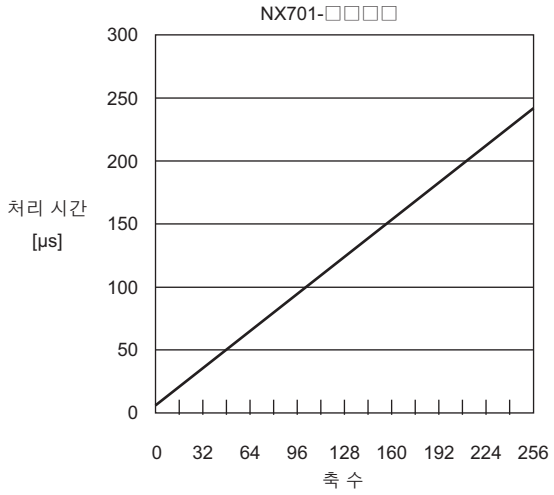
명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

• '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

모션 제어 처리 시간

모션 제어 처리 시간은 사용하는 서보의 축 수(가상 축 포함)에 좌우됩니다.
사용하는 서보 축 수를 기준으로 다음 그래프에서 모션 제어 처리 시간을 읽습니다.



공통 처리 시간

공통 처리 시간은 시스템 오버헤드, 시스템 공통 처리 1 및 시스템 공통 처리 2를 합하여 다음과 같은 값이 됩니다. 공통 처리 시간은 태스크 종류에 따라 다릅니다.

태스크 종류	공통 처리 시간[μs](참고값)
	NX701-□□□□
프라이머리 정주기 태스크	15
정주기 태스크(실행 우선도 5)	55
정주기 태스크(실행 우선도 16, 17, 18)	10

A-2-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예

태스크 실제 처리 시간의 평균값 계산 예

NX701-□□□□의 유니트 버전 Ver.1.10을 사용하는 경우 다음 조건일 때의 태스크 실제 처리 시간의 평균값을 구합니다.
태스크 종류는 프라이머리 정주기 태스크로 합니다.

항목		조건	
사용하는 슬레이브/유니트	EtherCAT 슬레이브	입력 슬레이브	GX-ID1611(Ver.1.1): 1대
		출력 슬레이브	GX-OD1611(Ver.1.1): 1대
		서보 드라이버	R88D-1SN□□□-ECT: 4대
		EtherCAT 슬레이브 터미널: 1대	
		EtherCAT 커플러 유니트	NX-ECC203: 1대
		DC 입력 유니트	NX-ID5342: 1대
		트랜지스터 출력 유니트	NX-OD5121: 1대
		아날로그 입력 유니트	NX-AD3608: 1대
		아날로그 출력 유니트	NX-DA2605: 1대
		커뮤니케이션 인터페이스 유니트	NX-CIF101: 1대 NX-CIF105: 1대
사용자 프로그램:	사용 언어	래더도 언어	
	표준적인 명령 구성	명령 수: 5,000개	
	배정도 실수 사칙연산	명령 수: 200개	
	배정도 실수 삼각 함수	명령 수: 100개	
모션 제어 처리	축 수	4축	

(주의) 내장 EtherCAT 포트에 연결한 케이블의 총 길이는 10[m]로 합니다.

● I/O 리프레시 시간

EtherCAT 슬레이브 처리 시간:

입력 슬레이브GX-ID1611(Ver.1.1), 출력 슬레이브GX-OD1611(Ver.1.1), 서보 드라이버 R88D-1SN□□□-ECT 및 상기 표시한 구성으로의 EtherCAT 슬레이브 터미널의 출력 프로세스 데이터 크기 pDout, 입력 프로세스 데이터 크기 pDin, 출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout의 값은 다음과 같습니다.

EtherCAT 슬레이브	출력 프로세스 데이터 크기 pDout [바이트]	입력 프로세스 데이터 크기 pDin [바이트]	출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout
GX-ID1611(Ver.1.1)	0	3	3
GX-OD1611(Ver.1.1)	2	1	2
R88D-1SN□□□-ECT	23	26	26
EtherCAT 슬레이브 터미널 합계	62	88	88
NX-ECC203	0	18	–
NX-ID5342	0	2	
NX-OD5121	2	0	
NX-AD3608	0	8	
NX-DA2605	4	0	
NX-CIF101	28	30	
NX-CIF105	28	30	

pDout, pDin 및 pDinout 각각의 합계는 다음과 같은 값입니다.

pDout = 2 + 23×4 + 62 = 156[바이트]

pDin = 3 + 1 + 26×4 + 88 = 196[바이트]

pDinout = 3 + 2 + 26×4 + 88 = 197[바이트]

이 값들을 이용해 I/O 리프레시 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간
=EtherCAT 슬레이브 처리 시간
=0.0006×pDout + 0.0001×pDin + 0.082×pDinout + (1.24×Snum + 0.01×Clen) + 25.58
=0.0006×156 + 0.0001×196 + 0.082×197 + (1.24×7 + 0.01×10) + 25.58
=0.0936 + 0.0196 + 16.154 + (8.68 + 0.1) + 25.58
≒51[μs]

● 사용자 프로그램 실행 시간

각 그래프에서 다음 수치를 읽습니다.

- 표준적인 명령 구성
  래더도 언어의 표준적인 명령의 실행 시간 그래프에서 NX701-□□□□의 명령 수 5000개인 사용자 프로그램 실행 시간은 12μs
- 배정도 실수 사칙연산
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX701-□□□□의 명령 수 200개인 사용자 프로그램 실행 시간은 4μs
- 배정도 실수 삼각 함수
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX701-□□□□의 명령 수 100개인 사용자 프로그램 실행 시간은 9μs

따라서 사용자 프로그램 실행 시간은 상기의 값을 합산하여 다음과 같은 값이 됩니다.

사용자 프로그램 실행 시간=12+4+9
=25[μs]

● 모션 제어 처리 시간

모션 제어 처리의 실행 시간 그래프에서 NX701-□□□□의 4축 모션 제어 처리의 실행 시간을 10μs로 읽습니다.

● 공통 처리 시간

태스크 종류가 프라이머리 정주기 태스크이므로 NX701-□□□□의 공통 처리 시간은 15μs입니다.

따라서 태스크 실행 시간의 평균값은 다음 식으로 구할 수 있습니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 시간+사용자 프로그램 실행 시간
+모션 제어 처리+공통 처리 시간
=51 + 25 + 10 + 15
=101[μs]

태스크 주기의 설정 예

위에서 구한 태스크 실제 처리 시간의 평균값을 바탕으로 태스크 주기를 설정합니다. 태스크 종류는 프라이머리 정주기 태스크입니다.
태스크 주기의 값은 계산으로 구한 태스크 실제 처리 시간의 평균값보다 크게 합니다. 그 때 태스크 주기 값을 여유 있게 태스크 실제 처리 시간의 평균값의 1.1배보다 크게 하십시오.

태스크 주기≧태스크 실제 처리 시간의 평균값×1.1

위에서 구한 태스크 실제 처리 시간의 평균값은 101μs이므로 태스크 주기는 그 1.1배보다 큰 값인 125μs로 설정합니다.

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간은 변동됩니다. 또한, 정주기 태스크인 경우는 그보다 실행 우선도가 더 높은 태스크의 동작 유무에 따라서도 태스크 실행 시간이 변동됩니다.
실제 기기의 컨트롤러를 사용하여 태스크 실행 시간 모니터링을 바탕으로 반드시 태스크 실행 시간을 검증하십시오.

A-3 NX102에서의 태스크 실제 처리 시간의 기준 계산

여기에서는 NX102를 사용하는 시스템에서의 태스크 실제 처리 시간의 평균값 기준 계산 방법에 대해 설명합니다.

태스크 실제 처리 시간과 태스크 실행 시간은 실제 기기 컨트롤러에서 확인해야 합니다. 세부 내용은 📖'5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'을 참조하십시오.



안전상의 중요 사항

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간이 변동됩니다.
최종적으로는 반드시 실제 기기를 사용하여 모든 조건에서 성능 확인을 실시하고, 태스크 주기 초과가 발생하지 않는지, 적절한 통신 응답성이 확보되는지 확인하십시오.



참고

정주기 태스크는 실행 우선도가 그보다 높은 태스크의 실행 중에는 일시 정지합니다. '태스크 실제 처리 시간'에 일시 정지한 시간은 포함하지 않습니다. 일시 정지한 시간을 포함하여 태스크 시작부터 종료까지 실제로 걸린 시간을 나타내는 것이 '태스크 실행 시간'입니다. '태스크 실제 처리 시간'과 '태스크 실행 시간'의 차이에 대한 세부 내용은 📖'태스크 실행 시간과 태스크 실제 처리 시간의 의미(5-107페이지)'를 참조하십시오.

A-3-1 태스크 실제 처리 시간의 평균값 계산 방법

태스크 실제 처리 시간의 평균값은 I/O 리프레시 처리 시간, 사용자 프로그램 실행 시간, 모션 제어 처리 시간, 공통 처리 시간의 합계입니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 처리 시간+사용자 프로그램 실행 시간
+모션 제어 처리 시간+공통 처리 시간

각 처리 내용은 다음과 같습니다.

○: 있음, −: 없음

처리		처리 내용	프라이머리 정주기 태스크	정주기 태스크 (실행 우선도 17, 18)
I/O 리프레시 처리		CPU 유니트 상의 NX 유니트, EtherCAT 슬레이브와의 리프레시를 실행합니다.	○	−
사용자 프로그램 실행		태스크에 할당된 프로그램을 할당순으로 실행합니다.	○	○
모션 제어 처리		• 사용자 프로그램으로부터의 모션 제어 명령을 실행합니다. • 모션 출력 처리	○	−
공통 처리 시간	시스템 공통 처리 1	• 변수의 갱신 처리(참조 태스크의 경우) • 모션 입력 처리 • 데이터 트레이스 처리	○	○
	시스템 공통 처리 2	• 변수의 갱신 처리(갱신 태스크의 경우) • 태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서 변수 액세스 처리	○	○
	시스템 오버헤드 시간	상기 이외의 시스템 공통 처리	○	○

아래에 각 처리 시간 계산의 기준을 표시합니다.

I/O 리프레시 처리 시간

I/O 리프레시 처리 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간=EtherCAT 슬레이브 처리 시간+NX 유니트 처리 시간

다음으로 상기 식의 EtherCAT 슬레이브 처리 시간, NX 유니트 처리 시간을 구하는 방법을 설명합니다.

● EtherCAT 슬레이브 처리 시간

EtherCAT 슬레이브 처리 시간은 다음 식으로 구할 수 있습니다.

EtherCAT 슬레이브 처리 시간[μs] = 0.017×pDout + 0.010×pDin + 0.082×pDinout
+ (1.24×Snum + 0.01×Clen) + 151.11

pDout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
pDin	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브의 입력 프로세스 데이터 크기 합계[바이트]
pDinout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브별 출력 프로세스 데이터 크기와 입력 프로세스 데이터의 크기 중 큰 쪽의 합계[바이트]
Snum	:	내장 EtherCAT 포트에 연결한 EtherCAT 슬레이브의 전체 대수[대]
Clen	:	내장 EtherCAT 포트에 연결한 케이블의 총 길이[m]

참고

EtherCAT 슬레이브를 연결하지 않은 경우에 EtherCAT 슬레이브 처리 시간은 '0'입니다.

● NX 유니트 처리 시간

NX 유니트 처리 시간은 다음 식으로 구할 수 있습니다.
CPU 유니트 상의 모든 NX 유니트에 대해 계산하십시오.

NX 유니트 처리 시간[μs] = 0.149×Dnxout + 0.064×Dnxin + 0.549×Dnxnum + 97.48

Dnxout	: 출력 데이터 크기[바이트]*1
Dnxin	: 입력 데이터 크기[바이트]*1
Dnxnum	: NX 유니트 연결 대수[대]

*1. 2바이트 이하 NX 유니트인 경우에는 2바이트로 계산하십시오.

참고

CPU 유니트 상에 NX 유니트를 연결하지 않은 경우에 NX 유니트 처리 시간은 '0'입니다.

사용자 프로그램 실행 시간

사용자 프로그램 실행 시간은 명령 종류×명령 수에 좌우됩니다.

기준으로는 크게 다음 3가지로 분류한 명령 수를 계측 또는 예상합니다.
- 표준적인 명령
- 배정도 실수(LREAL) 형식 사칙연산 명령
- 배정도 실수(LREAL) 형식 삼각 함수 명령

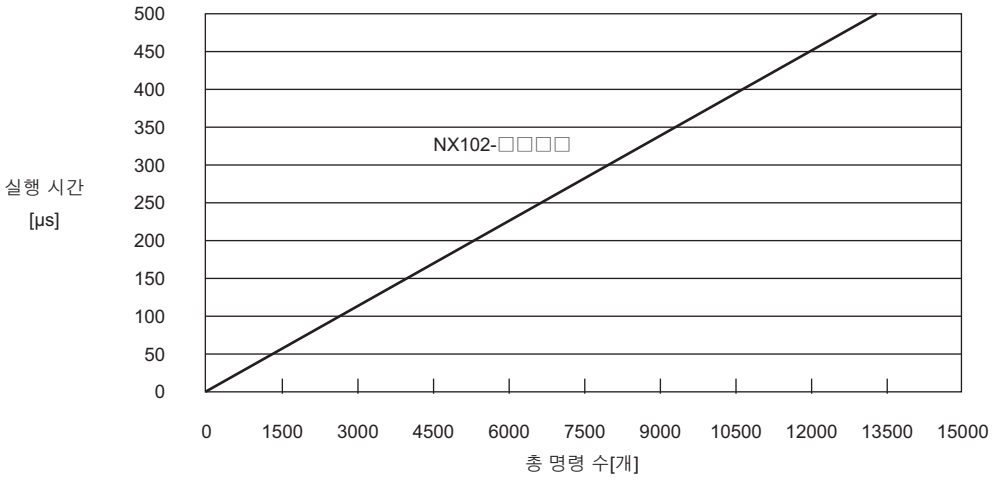
또한, 래더도 언어와 ST 언어에서는 사용하는 명령이 다릅니다. 각각의 명령 구성에 대해서는 "래더도 언어의 표준적인 명령'의 명령 종류 구성(A-56페이지)' 및 "ST 언어의 표준적인 명령'의 명령 종류 구성(A-58페이지)'을 참조하십시오.

● 간단한 계산

각 분류의 명령 수를 기준으로 다음 그래프에서 다음 각 명령의 실행 시간을 읽고 합계를 구합니다.
- 표준적인 명령의 실행 시간
- 배정도 실수(LREAL) 형식 사칙연산 명령의 실행 시간
- 배정도 실수(LREAL) 형식 삼각 함수 명령의 실행 시간

이를 통해 사용자 프로그램의 실행 시간을 추측할 수 있습니다.

래더도 언어의 표준적인 명령의 실행 시간

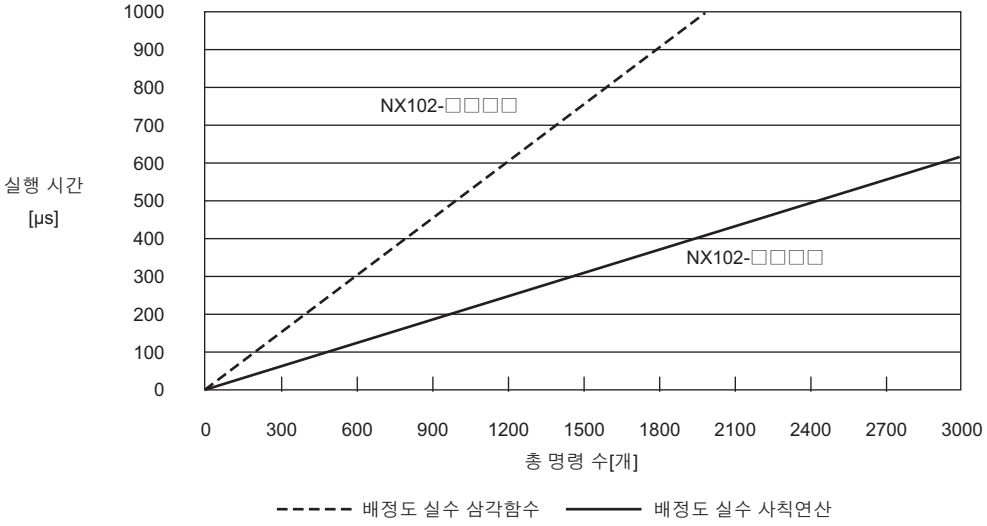


- '래더도 언어의 표준적인 명령'의 명령 종류 구성

이 구성에서의 명령 실행률은 20%입니다.

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
래더도 명령	LD, AND, OUT, SET, RESET	81.0	40.2
비교 명령	EQ, LT	4.1	8.3
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	1.6	7.3
산술 명령	+, -, *, /, ADD, SUB, MUL, DIV	2.4	6.5
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.2
비트 열 연산 명령	AND, OR	6.2	13.0
데이터 전송 명령	MOVE	4.6	23.5
합계		100.0	100.0

래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



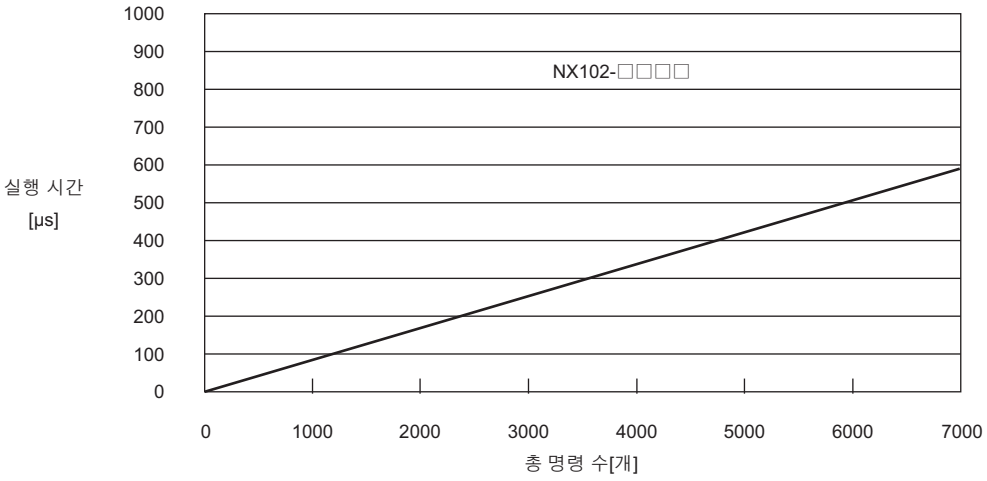
- '배정도 실수 사칙연산'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

- '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율 [%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

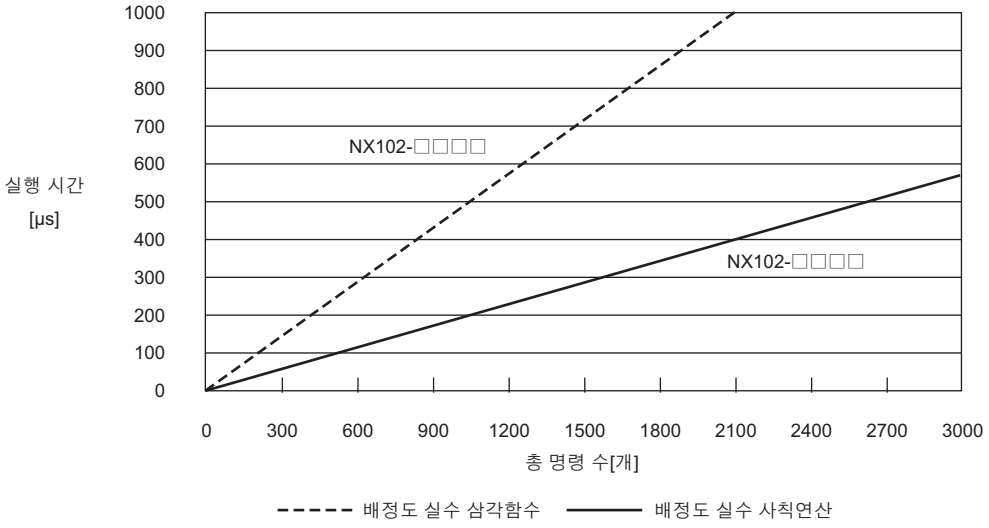
ST 언어의 표준적인 명령의 실행 시간



• 'ST 언어의 표준적인 명령'의 명령 종류 구성

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
ST 구문 명령	IF ELSIF END_IF	75.4	41.6
비교 명령	EQ, LT	5.2	8.7
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	2.1	18.8
산술 명령	+, -, *, /	3.1	10.2
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.6
비트 열 연산 명령	AND, OR	8.0	11.7
데이터 전송 명령	:=	5.9	7.3
합계		100.0	100.0

ST 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



• '배정도 실수 사칙연산'의 구성

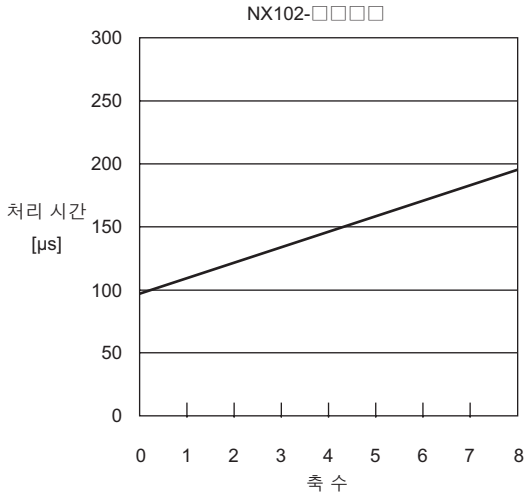
명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

• '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

모션 제어 처리 시간

모션 제어 처리 시간은 사용하는 서보의 축 수(가상 축 포함)에 좌우됩니다.
사용하는 서보 축 수를 기준으로 다음 그래프에서 모션 제어 처리 시간을 읽습니다.



공통 처리 시간

공통 처리 시간은 시스템 오버헤드, 시스템 공통 처리 1 및 시스템 공통 처리 2를 합하여 다음과 같은 값이 됩니다. 공통 처리 시간은 태스크 종류에 따라 다릅니다.

태스크 종류	공통 처리 시간[μs](참고값)
	NX102-□□□□
프라이머리 정주기 태스크	133
정주기 태스크	175

A-3-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예

태스크 실제 처리 시간의 평균값 계산 예

NX102-□□□□의 유니트 버전 Ver.1.30을 사용하는 경우 다음 조건일 때의 태스크 실제 처리 시간의 평균값을 구합니다.
태스크 종류는 프라이머리 정주기 태스크로 합니다.

항목		조건
사용하는 슬레이브/유니트	EtherCAT 슬레이브	• 입력 슬레이브 GX-ID1611(Ver.1.1): 1대 • 출력 슬레이브 GX-OD1611(Ver.1.1): 1대 • 서보 드라이버 R88D-1SN□□□-ECT: 4대
	NX 유니트 (CPU 랙 상)	• I/O 전원 추가 공급 유니트 NX-PF0630: 1대 • 디지털 입력 유니트 NX-ID5342: 3대 • 디지털 출력 유니트 NX-OD3153: 2대 • 아날로그 입력 유니트 NX-AD4608: 1대 • 아날로그 입력 유니트 NX-DA3605: 1대
사용자 프로그램:	사용 언어	래더도 언어
	표준적인 명령 구성	명령 수: 5,000개
	배정도 실수 사칙연산	명령 수: 200개
	배정도 실수 삼각 함수	명령 수: 100개
모션 제어 처리	축 수	4축

(주의) 내장 EtherCAT 포트에 연결한 케이블의 총 길이는 10[m]로 합니다.

● I/O 리프레시 시간

EtherCAT 슬레이브 처리 시간:

입력 슬레이브GX-ID1611(Ver.1.1), 출력 슬레이브GX-OD1611(Ver.1.1), 서보 드라이버 R88D-1SN□□□-ECT의 출력 프로세스 데이터 크기 pDout, 입력 프로세스 데이터 크기 pDin, 출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout의 값은 다음과 같습니다.

EtherCAT 슬레이브	출력 프로세스 데이터 크기 pDout [바이트]	입력 프로세스 데이터 크기 pDin [바이트]	출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout
GX-ID1611(Ver.1.1)	0	3	3
GX-OD1611(Ver.1.1)	2	1	2
R88D-1SN□□□-ECT	23	26	26

pDout, pDin 및 pDinout 각각의 합계는 다음과 같은 값입니다.

pDout = 2 + 23×4 = 94[바이트]
pDin = 3 + 1 + 26×4 = 108[바이트]
pDinout = 3 + 2 + 26×4 = 109[바이트]

NX 유니트 처리 시간:

CPU 랙 상의 I/O 전원 추가 공급 유니트 NX-PF0630, 디지털 입력 유니트 NX-ID5342, 디지털 출력 유니트 NX-OD3153, 아날로그 입력 유니트 NX-AD4608, 아날로그 출력 유니트 NX-DA3605의 출력 데이터 크기 Dnxout, 입력 데이터 크기 Dnxin의 값은 다음과 같습니다.

NX 유니트	출력 데이터 크기 Dnxout	입력 데이터 크기 Dnxin
NX-PF0630	0	0
NX-ID5342	0	2
NX-OD3153	2	0
NX-AD4608	0	16
NX-DA3605	8	0

Dnxout, Dnxin 각각의 합계는 다음과 같은 값이 됩니다.

Dnxout = 2×2 + 8 = 12[바이트]

Dnxin = 2×3 + 16 = 22[바이트]

이 값들을 이용해 I/O 리프레시 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간
= EtherCAT 슬레이브 처리 시간 + NX 유니트 처리 시간
= 0.017×pDout + 0.010×pDin + 0.082×pDinout + (1.24×Snum + 0.01×Clen) + 151.11
+ 0.149×Dnxout + 0.064×Dnxin + 0.549×Dnxnum + 97.48
= 0.017×94 + 0.010×108 + 0.082×109 + (1.24×6 + 0.01×10) + 151.11
+ 0.149×12 + 0.064×22 + 0.549×8 + 97.48
=1.598 + 1.080 + 8.938 + (7.24 + 0.1) + 151.11
+ 1.788 + 1.408 + 4.392 + 97.48
≒275[μs]

● 사용자 프로그램 실행 시간

각 그래프에서 다음 수치를 읽습니다.

- 표준적인 명령 구성
  래더도 언어의 표준적인 명령의 실행 시간 그래프에서 NX102-□□□□의 명령 수 5000개인 사용자 프로그램 실행 시간은 189μs
- 배정도 실수 사칙연산
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX102-□□□□의 명령 수 200개인 사용자 프로그램 실행 시간은 41μs
- 배정도 실수 삼각 함수
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX102-□□□□의 명령 수 100개인 사용자 프로그램 실행 시간은 51μs

따라서 사용자 프로그램 실행 시간은 상기의 값을 합산하여 다음과 같은 값이 됩니다.

사용자 프로그램 실행 시간 = 189 + 41 + 51
= 281[μs]

● 모션 제어 처리 시간

모션 제어 처리의 실행 시간 그래프에서 NX102-□□□□ Ver.1.30의 4축 모션 제어 처리의 실행 시간을 148μs로 읽습니다.

● 공통 처리 시간

태스크 종류가 프라이머리 정주기 태스크이므로 NX102-□□□□의 공통 처리 시간은 133μs입니다.

따라서 태스크 실행 시간의 평균값은 다음 식으로 구할 수 있습니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 시간+사용자 프로그램 실행 시간
+모션 제어 처리+공통 처리 시간
=275 + 281 + 148 + 133
=837[μs]

태스크 주기의 설정 예

위에서 구한 태스크 실제 처리 시간의 평균값을 바탕으로 태스크 주기를 설정합니다. 태스크 종류는 프라이머리 정주기 태스크입니다.
태스크 주기의 값은 계산으로 구한 태스크 실제 처리 시간의 평균값보다 크게 합니다. 그 때 태스크 주기 값을 여유 있게 태스크 실제 처리 시간의 평균값의 1.1배보다 크게 하십시오.

태스크 주기≧태스크 실제 처리 시간의 평균값×1.1

위에서 구한 태스크 실제 처리 시간의 평균값은 837μs이므로 태스크 주기는 그 1.1배보다 큰 값인 1000μs로 설정합니다.

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간은 변동됩니다. 또한, 정주기 태스크인 경우는 그보다 실행 우선도가 더 높은 태스크의 동작 유무에 따라서도 태스크 실행 시간이 변동됩니다.
실제 기기의 컨트롤러를 사용하여 태스크 실행 시간 모니터링을 바탕으로 반드시 태스크 실행 시간을 검증하십시오.

A-4 NX1P2에서의 태스크 실제 처리 시간의 기준 계산

여기서는 NX1P2를 사용하는 시스템에서의 태스크 실제 처리 시간의 평균값 기준 계산 방법에 대해 설명합니다.

태스크 실제 처리 시간과 태스크 실행 시간은 실제 기기 컨트롤러에서 확인해야 합니다. 세부 내용은 📖'5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'을 참조하십시오.



안전상의 중요 사항

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간이 변동됩니다.
최종적으로는 반드시 실제 기기를 사용하여 모든 조건에서 성능 확인을 실시하고, 태스크 주기 초과가 발생하지 않는지, 적절한 통신 응답성이 확보되는지 확인하십시오.



참고

정주기 태스크는 실행 우선도가 그보다 높은 태스크의 실행 중에는 일시 정지합니다. '태스크 실제 처리 시간'에 일시 정지한 시간은 포함하지 않습니다. 일시 정지한 시간을 포함하여 태스크 시작부터 종료까지 실제로 걸린 시간을 나타내는 것이 '태스크 실행 시간'입니다. '태스크 실제 처리 시간'과 '태스크 실행 시간'의 차이에 대한 세부 내용은 📖'태스크 실행 시간과 태스크 실제 처리 시간의 의미(5-107페이지)'를 참조하십시오.

A-4-1 태스크 실제 처리 시간의 평균값 계산 방법

태스크 실제 처리 시간의 평균값은 I/O 리프레시 처리 시간, 사용자 프로그램 실행 시간, 모션 제어 처리 시간, 공통 처리 시간의 합계입니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 처리 시간+사용자 프로그램 실행 시간
+모션 제어 처리 시간+공통 처리 시간

각 처리 내용은 다음과 같습니다.

○: 있음, −: 없음

처리		처리 내용	프라이머리 정주기 태스크	정주기 태스크 (실행 우선도 17, 18)
I/O 리프레시 처리		CPU 유니트 상의 NX 유니트, 내장 I/O, EtherCAT 슬레이브와의 리프레시를 실행합니다.	○	−
사용자 프로그램 실행		태스크에 할당된 프로그램을 할당순으로 실행합니다.	○	○
모션 제어 처리		• 사용자 프로그램으로부터의 모션 제어 명령을 실행합니다. • 모션 출력 처리	○	−
공통 처리 시간	시스템 공통 처리 1	• 변수의 갱신 처리(참조 태스크의 경우) • 모션 입력 처리 • 데이터 트레이스 처리	○	○
	시스템 공통 처리 2	• 변수의 갱신 처리(갱신 태스크의 경우) • 태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서 변수 액세스 처리	○	○
	시스템 오버헤드 시간	상기 이외의 시스템 공통 처리	○	○

아래에 각 처리 시간 계산의 기준을 표시합니다.

I/O 리프레시 처리 시간

I/O 리프레시 처리 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간=EtherCAT 슬레이브 처리 시간+NX 유니트 처리 시간

다음으로 상기 식의 EtherCAT 슬레이브 처리 시간, NX 유니트 처리 시간을 구하는 방법을 설명합니다.

● EtherCAT 슬레이브 처리 시간

EtherCAT 슬레이브 처리 시간은 다음 식으로 구할 수 있습니다.

EtherCAT 슬레이브 처리 시간[μs] = 0.017×pDout + 0.010×pDin + 0.082×pDinout
+ (1.24×Snum + 0.01×Clen) + 151.11

pDout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
pDin	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브의 입력 프로세스 데이터 크기 합계[바이트]
pDinout	:	프라이머리 정주기 태스크에 할당한 EtherCAT 슬레이브별 출력 프로세스 데이터 크기와 입력 프로세스 데이터의 크기 중 큰 쪽의 합계[바이트]
Snum	:	내장 EtherCAT 포트에 연결한 EtherCAT 슬레이브의 전체 대수[대]
Clen	:	내장 EtherCAT 포트에 연결한 케이블의 총 길이[m]

참고

EtherCAT 슬레이브를 연결하지 않은 경우에 EtherCAT 슬레이브 처리 시간은 '0'입니다.

● NX 유니트 처리 시간

NX 유니트 처리 시간은 다음 식으로 구할 수 있습니다.
CPU 유니트 상의 모든 NX 유니트에 대해 계산하십시오.

NX 유니트 처리 시간 = 0.248×Dnxout + 0.241×Dnxin + 0.549×Dnxnum + 125.36

Dnxout	: 출력 데이터 크기[바이트]*1
Dnxin	: 입력 데이터 크기[바이트]*1
Dnxnum	: NX 유니트 연결 대수[대]

*1. 2바이트 이하 NX 유니트인 경우에는 2바이트로 계산하십시오.

참고

CPU 유니트 상에 NX 유니트를 연결하지 않은 경우에 NX 유니트 처리 시간은 '0'입니다.

사용자 프로그램 실행 시간

사용자 프로그램 실행 시간은 명령 종류×명령 수에 좌우됩니다.

기준으로는 크게 다음 3가지로 분류한 명령 수를 계측 또는 예상합니다.
- 표준적인 명령
- 배정도 실수(LREAL) 형식 사칙연산 명령
- 배정도 실수(LREAL) 형식 삼각 함수 명령

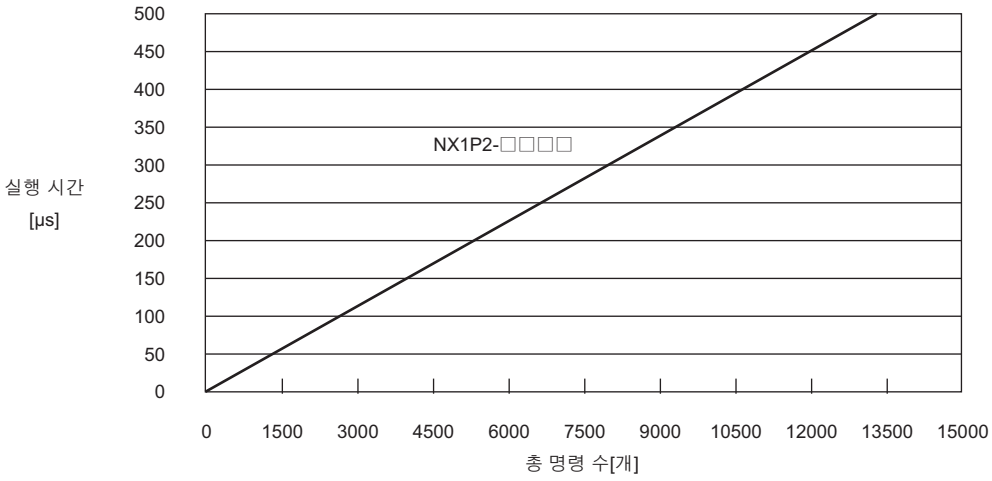
또한, 래더도 언어와 ST 언어에서는 사용하는 명령이 다릅니다. 각각의 명령 구성에 대해서는 "래더도 언어의 표준적인 명령'의 명령 종류 구성(A-56페이지)' 및 "ST 언어의 표준적인 명령'의 명령 종류 구성(A-58페이지)'을 참조하십시오.

● 간단한 계산

각 분류의 명령 수를 기준으로 다음 그래프에서 다음 각 명령의 실행 시간을 읽고 합계를 구합니다.
- 표준적인 명령의 실행 시간
- 배정도 실수(LREAL) 형식 사칙연산 명령의 실행 시간
- 배정도 실수(LREAL) 형식 삼각 함수 명령의 실행 시간

이를 통해 사용자 프로그램의 실행 시간을 추측할 수 있습니다.

래더도 언어의 표준적인 명령의 실행 시간

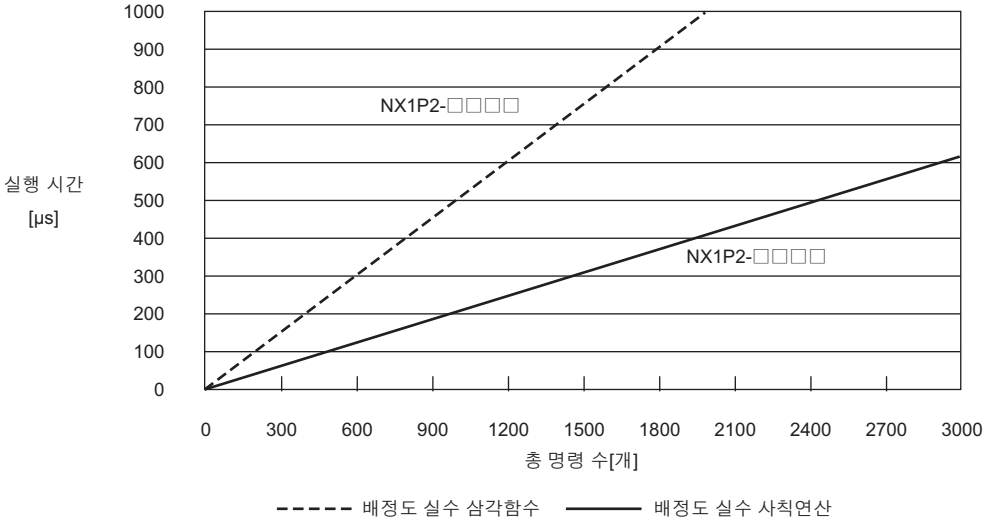


- '래더도 언어의 표준적인 명령'의 명령 종류 구성

이 구성에서의 명령 실행률은 20%입니다.

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
래더도 명령	LD, AND, OUT, SET, RESET	81.0	40.2
비교 명령	EQ, LT	4.1	8.3
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	1.6	7.3
산술 명령	+, -, *, /, ADD, SUB, MUL, DIV	2.4	6.5
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.2
비트 열 연산 명령	AND, OR	6.2	13.0
데이터 전송 명령	MOVE	4.6	23.5
합계		100.0	100.0

래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



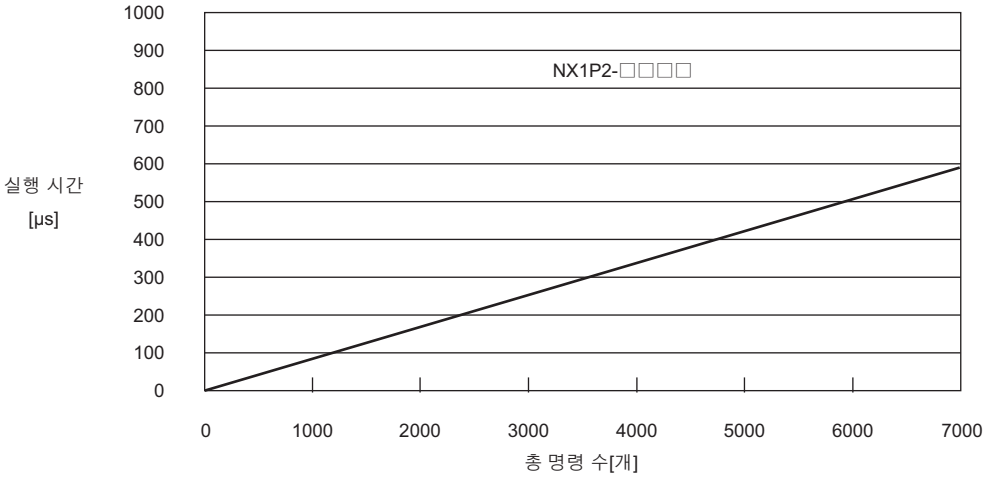
• '배정도 실수 사칙연산'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

• '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

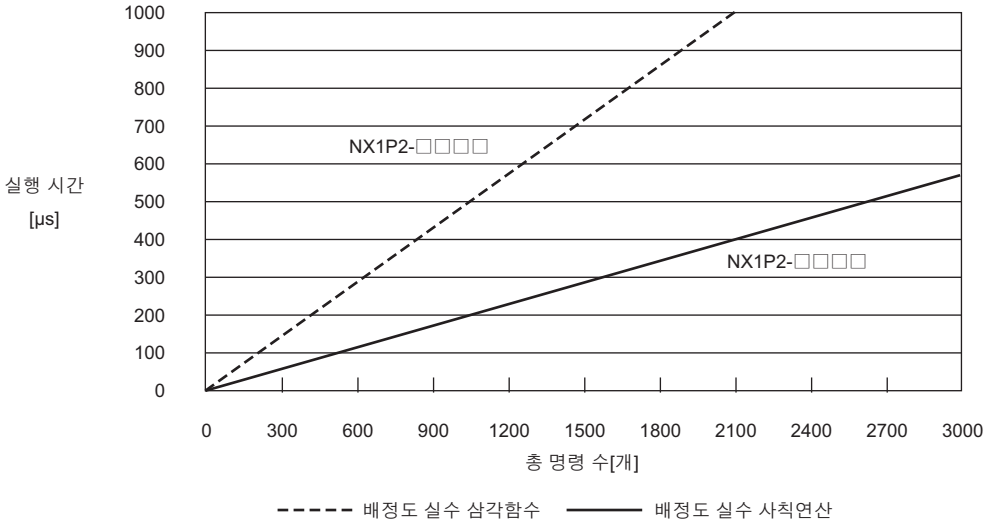
ST 언어의 표준적인 명령의 실행 시간



• 'ST 언어의 표준적인 명령'의 명령 종류 구성

명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
ST 구문 명령	IF ELSIF END_IF	75.4	41.6
비교 명령	EQ, LT	5.2	8.7
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	2.1	18.8
산술 명령	+, -, *, /	3.1	10.2
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.6
비트 열 연산 명령	AND, OR	8.0	11.7
데이터 전송 명령	:=	5.9	7.3
합계		100.0	100.0

ST 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간



• '배정도 실수 사칙연산'의 구성

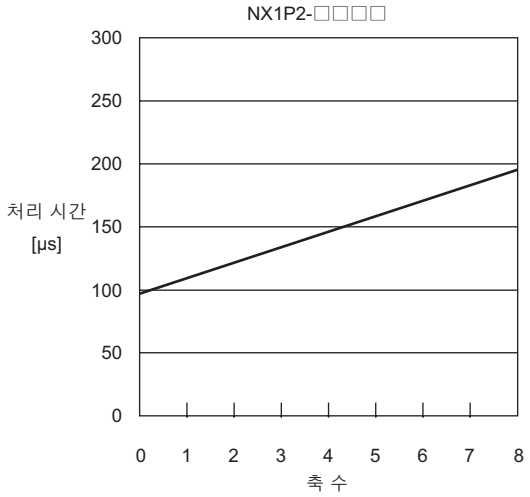
명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

• '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

모션 제어 처리 시간

모션 제어 처리 시간은 사용하는 서보의 축 수(가상 축 포함)에 좌우됩니다.
사용하는 서보 축 수를 기준으로 다음 그래프에서 모션 제어 처리 시간을 읽습니다.



공통 처리 시간

공통 처리 시간은 시스템 오버헤드, 시스템 공통 처리 1 및 시스템 공통 처리 2를 합하여 다음과 같은 값이 됩니다. 공통 처리 시간은 태스크 종류에 따라 다릅니다.

태스크 종류	공통 처리 시간[μs](참고값)
	NX1P2-□□□□
프라이머리 정주기 태스크	133
정주기 태스크	175

A-4-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예

태스크 실제 처리 시간의 평균값 계산 예

NX1P2-□□□□의 유니트 버전 Ver.1.13을 사용하는 경우 다음 조건일 때의 태스크 실제 처리 시간의 평균값을 구합니다.
태스크 종류는 프라이머리 정주기 태스크로 합니다.

항목		조건
사용하는 슬레이브/유니트	**EtherCAT 슬레이브**	• 입력 슬레이브 GX-ID1611(Ver.1.1): 1대 • 출력 슬레이브 GX-OD1611(Ver.1.1): 1대 • 서보 드라이버 R88D-1SN□□□-ECT: 4대
	NX 유니트 (CPU 랙 상)	• I/O 전원 추가 공급 유니트 NX-PF0630: 1대 • 디지털 입력 유니트 NX-ID5342: 3대 • 디지털 출력 유니트 NX-OD3153: 2대 • 아날로그 입력 유니트 NX-AD4608: 1대 • 아날로그 입력 유니트 NX-DA3605: 1대
사용자 프로그램:	**사용 언어**	래더도 언어
	표준적인 명령 구성	명령 수: 5,000개
	배정도 실수 사칙연산	명령 수: 200개
	배정도 실수 삼각 함수	명령 수: 100개
모션 제어 처리	**축 수**	4축

(주의) 내장 EtherCAT 포트에 연결한 케이블의 총 길이는 10[m]로 합니다.

● I/O 리프레시 시간

EtherCAT 슬레이브 처리 시간:

입력 슬레이브 GX-ID1611(Ver.1.1), 출력 슬레이브 GX-OD1611(Ver.1.1), 서보 드라이버 R88D-1SN□□□-ECT의 출력 프로세스 데이터 크기 pDout, 입력 프로세스 데이터 크기 pDin, 출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout의 값은 다음과 같습니다.

EtherCAT 슬레이브	출력 프로세스 데이터 크기 pDout [바이트]	입력 프로세스 데이터 크기 pDin [바이트]	출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 pDinout
GX-ID1611(Ver.1.1)	0	3	3
GX-OD1611(Ver.1.1)	2	1	2
R88D-1SN□□□-ECT	23	26	26

pDout, pDin 및 pDinout 각각의 합계는 다음과 같은 값입니다.

pDout = 2 + 23×4 = 94[바이트]
pDin = 3 + 1 + 26×4 = 108[바이트]
pDinout = 3 + 2 + 26×4 = 109[바이트]

NX 유니트 처리 시간:

CPU 랙 상의 I/O 전원 추가 공급 유니트 NX-PF0630, 디지털 입력 유니트 NX-ID5342, 디지털 출력 유니트 NX-OD3153, 아날로그 입력 유니트 NX-AD4608, 아날로그 출력 유니트 NX-DA3605의 출력 데이터 크기 Dnxout, 입력 데이터 크기 Dnxin의 값은 다음과 같습니다.

NX 유니트	출력 데이터 크기 Dnxout	입력 데이터 크기 Dnxin
NX-PF0630	0	0
NX-ID5342	0	2
NX-OD3153	2	0
NX-AD4608	0	16
NX-DA3605	8	0

Dnxout, Dnxin 각각의 합계는 다음과 같은 값이 됩니다.

Dnxout = 2×2 + 8 = 12[바이트]

Dnxin = 2×3 + 16 = 22[바이트]

이 값들을 이용해 I/O 리프레시 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간
= EtherCAT 슬레이브 처리 시간 + NX 유니트 처리 시간
= 0.017×pDout + 0.010×pDin + 0.082×pDinout + (1.24×Snum + 0.01×Clen) + 151.11
+ 0.248×Dnxout + 0.241×Dnxin + 0.549×Dnxnum + 125.36
= 0.017×94 + 0.010×108 + 0.082×109 + (1.24×6 + 0.01×10) + 151.11
+ 0.248×12 + 0.241×22 + 0.549×8 + 125.36
=1.598 + 1.080 + 8.938 + (7.44 + 0.1) + 151.11
+ 2.976 + 5.302 + 4.392 + 125.36
≒308[μs]

● 사용자 프로그램 실행 시간

각 그래프에서 다음 수치를 읽습니다.

- 표준적인 명령 구성
  래더도 언어의 표준적인 명령의 실행 시간 그래프에서 NX1P2-□□□□의 명령 수 5000개인 사용자 프로그램 실행 시간은 189μs
- 배정도 실수 사칙연산
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX1P2-□□□□의 명령 수 200개인 사용자 프로그램 실행 시간은 41μs
- 배정도 실수 삼각 함수
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NX1P2-□□□□의 명령 수 100개인 사용자 프로그램 실행 시간은 51μs

따라서 사용자 프로그램 실행 시간은 상기의 값을 합산하여 다음과 같은 값이 됩니다.

사용자 프로그램 실행 시간=189 + 41 + 51
=281[μs]

● 모션 제어 처리 시간

모션 제어 처리의 실행 시간 그래프에서 NX1P2-□□□□ Ver.1.13의 4축 모션 제어 처리의 실행 시간을 148μs로 읽습니다.

● 공통 처리 시간

태스크 종류가 프라이머리 정주기 태스크이므로 NX1P2-□□□□의 공통 처리 시간은 133μs입니다.

따라서 태스크 실행 시간의 평균값은 다음 식으로 구할 수 있습니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 시간+사용자 프로그램 실행 시간
+모션 제어 처리+공통 처리 시간
=308 + 281 + 148 + 133
=870[μs]

태스크 주기의 설정 예

위에서 구한 태스크 실제 처리 시간의 평균값을 바탕으로 태스크 주기를 설정합니다. 태스크 종류는 프라이머리 정주기 태스크입니다.
태스크 주기의 값은 계산으로 구한 태스크 실제 처리 시간의 평균값보다 크게 합니다. 그 때 태스크 주기 값을 여유 있게 태스크 실제 처리 시간의 평균값의 1.1배보다 크게 하십시오.

태스크 주기≧태스크 실제 처리 시간의 평균값×1.1

위에서 구한 태스크 실제 처리 시간의 평균값은 870μs이므로 태스크 주기는 그 1.1배보다 큰 값인 2000μs로 설정합니다.

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간은 변동됩니다. 또한, 정주기 태스크인 경우는 그보다 실행 우선도가 더 높은 태스크의 동작 유무에 따라서도 태스크 실행 시간이 변동됩니다.
실제 기기의 컨트롤러를 사용하여 태스크 실행 시간 모니터링을 바탕으로 반드시 태스크 실행 시간을 검증하십시오.

A-5 NJ 시리즈에서의 태스크 실제 처리 시간의 기준 계산

여기서는 NJ 시리즈 시스템에서의 태스크 실제 처리 시간의 평균값 기준 계산 방법에 대해 설명합니다.

태스크 실제 처리 시간과 태스크 실행 시간은 실제 기기 컨트롤러에서 확인해야 합니다. 세부 내용은 '5-11 태스크 설계 방법 및 입출력 응답 시간(5-108페이지)'을 참조하십시오.



안전상의 중요 사항

- 실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간이 변동됩니다.
  최종적으로는 반드시 실제 기기를 사용하여 모든 조건에서 성능 확인을 실시하고, 태스크 주기 초과가 발생하지 않는지, 적절한 통신 응답성이 확보되는지 확인하십시오.
- 하드웨어 개정이 다르면 성능이 다를 수 있습니다. 하드웨어 개정 번호가 다른 CPU 유니트에 사용자 프로그램 및 각종 데이터, 설정값을 전송할 때에는 동작을 충분히 확인한 후에 본 운전을 실시하십시오.



사용상의 주의 사항

하드웨어 개정 번호가 다른 경우에 사용자 프로그램 및 각종 데이터, 설정값이 같더라도 이벤트 태스크, 태그 데이터 링크 서비스 및 시스템 서비스의 실제 실행 타이밍이 다른 경우가 있습니다.



참고

정주기 태스크는 실행 우선도가 그보다 높은 태스크의 실행 중에는 일시 정지합니다. '태스크 실제 처리 시간'에 일시 정지한 시간은 포함하지 않습니다. 일시 정지한 시간을 포함하여 태스크 시작부터 종료까지 실제로 걸린 시간을 나타내는 것이 '태스크 실행 시간'입니다. '태스크 실제 처리 시간'과 '태스크 실행 시간'의 차이에 대한 세부 내용은 '태스크 실행 시간과 태스크 실제 처리 시간의 의미(5-107페이지)'를 참조하십시오.

A-5-1 태스크 실제 처리 시간의 평균값 계산 방법

태스크 실제 처리 시간의 평균값은 I/O 리프레시 처리 시간, 사용자 프로그램 실행 시간, 모션 제어 처리 시간, 공통 처리 시간의 합계입니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 처리 시간+사용자 프로그램 실행 시간
+모션 제어 처리 시간+공통 처리 시간

각 처리 내용은 다음과 같습니다.

○: 있음, −: 없음

처리		처리 내용	프라이머리 정주기 태스크	정주기 태스크 (실행 우선도 16)	정주기 태스크 (실행 우선도 17, 18)
I/O 리프레시 처리		CJ 유니트(기본 I/O 유니트, 고기능 I/O 유니트, CPU 고기능 유니트), EtherCAT 슬레이브와의 리프레시를 실행합니다.	○	○	−
사용자 프로그램 실행		태스크에 할당된 프로그램을 할당순으로 실행합니다.	○	○	○
모션 제어 처리		• 사용자 프로그램으로부터의 모션 제어 명령을 실행합니다. • 모션 출력 처리	○	−	−
공통 처리 시간	시스템 공통 처리 1	• 변수의 갱신 처리(참조 태스크의 경우) • 모션 입력 처리 • 데이터 트레이스 처리	○	○	○
	시스템 공통 처리 2	• 변수의 갱신 처리(갱신 태스크의 경우) • 태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서 변수 액세스 처리	○	○	○
	시스템 오버헤드 시간	상기 이외의 시스템 공통 처리	○	○	○

아래에 각 처리 시간 계산의 기준을 표시합니다.

I/O 리프레시 처리 시간

I/O 리프레시 처리 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간=I/O 리프레시 오버헤드 시간
+(EtherCAT 슬레이브 처리 시간과 CJ 유니트 처리 시간 중 큰 쪽)

다음으로 상기 식의 I/O 리프레시 오버 헤드 시간, EtherCAT 슬레이브 처리 시간, CJ 유니트 처리 시간을 구하는 방법을 설명합니다.

● I/O 리프레시 오버헤드 시간

I/O 리프레시 오버헤드 시간은 EtherCAT 슬레이브의 유무, CJ 유니트의 유무 및 CPU 유니트의 형식에 따라 다음 표와 같이 결정됩니다.

EtherCAT 슬레이브	CJ 유니트	I/O 리프레시 오버헤드 시간[μs]		
		NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
있음	있음	65*1	95	135
있음	없음	35*2	50	65
없음	있음	30	45	70

*1. CPU 유니트 Ver.1.01 이하는 '90'입니다.
*2. CPU 유니트 Ver.1.01 이하는 '60'입니다.

● EtherCAT 슬레이브 처리 시간

EtherCAT 슬레이브 처리 시간은 다음 식으로 구할 수 있습니다.

EtherCAT 슬레이브 처리 시간[μs] = Tout×Dout + Tin×Din + Tref×Dinout
+ (1.24×Snum + 0.01×Clen - Tec)

Tout	: 1바이트당 출력 처리 시간[μs]
Dout	: 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브의 출력 프로세스 데이터 크기 합계[바이트]
Tin	: 1바이트당 입력 처리 시간[μs]
Din	: 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브의 입력 프로세스 데이터 크기 합계[바이트]
Snum	: EtherCAT 슬레이브 연결 대수[대]
Tref	: 1바이트당 리프레시 처리 시간[μs]
Dinout	: 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 16)에 할당한 EtherCAT 슬레이브별 출력 프로세스 데이터 크기와 입력 프로세스 데이터의 크기 중 큰 쪽의 합계[바이트]
Clen	: 케이블 총 길이[m]
Tec	: EtherCAT 통신 조정 시간[μs]

위 식 내의 출력 처리 시간, 입력 처리 시간, 리프레시 처리 시간 및 EtherCAT 통신 조정 시간은 고정값입니다. CPU 유니트의 형식에 따라 다음과 같이 결정됩니다.

CPU 유니트	1바이트당 출력 처리 시간 Tout[μs]	1바이트당 입력 처리 시간 Tin[μs]	1바이트당 리프레시 처리 시간 Tref[μs]	EtherCAT 통신 조정 시간 Tec[μs]
NJ501-□□□□	0.004	0.011	0.082	55*1
NJ301-□□□□	0.005	0.013	0.082	90
NJ101-□□□□	0.010	0.004	0.082	145

*1. CPU 유니트 Ver.1.01 이하는 '70'입니다.

또한, 상기 식 내의 '1.24×Snum + 0.01×Clen - Tec'의 계산 결과가 음수인 경우는 '0'으로 간주합니다.

● CJ 유니트 처리 시간

CJ 유니트 처리 시간은 다음 식으로 구할 수 있습니다.

Σ(각 CJ 유니트의 I/O 리프레시 시간×유니트 대수)-Tcj[μs]

단, 위 식의 Σ는 모든 CJ 유니트에 대한 합계를 나타냅니다.
또한, 위 식의 계산 결과가 음수인 경우에는 CJ 유니트 처리 시간을 0μs로 간주합니다.
각 CJ 유니트의 I/O 리프레시 시간을 구하는 방법은 뒤에서 설명합니다.
Tcj는 CPU 유니트의 형식과 유니트 버전에 따라 다릅니다.

형식	유니트 버전	Tcj[μs]
NJ501-□□□□	Ver.1.01 이하	230
	Ver.1.02 이상	110
NJ301-□□□□	Ver.1.01 이하	230
	Ver.1.02 이상	165
NJ101-□□□□	Ver.1.10 이상*1	225

*1. NJ101-□□□□에는 Ver.1.09 이하가 없습니다.

다음 CJ 유니트를 1대 이상 사용하는 경우는 유니트 대수에 관계 없이 위 식의 계산 결과에 +Tcj[μs]를 더하십시오.

- 절연형 풀 멀티 입력 유니트 CJ1W-PH41U
- 절연형 풀 멀티 입력 유니트 CJ1W-AD04U
- 절연형 직류 입력 유니트 CJ1W-PDC15
- ID 센서 유니트 CJ1W-V680C11
- ID 센서 유니트 CJ1W-V680C12
- CompoNet 마스터 유니트 CJ1W-CRM21

CJ 유니트의 I/O 리프레시 시간

각 CJ 유니트의 I/O 리프레시 시간은 다음과 같습니다.

● 기본 I/O 유니트

기본 I/O 유니트를 CPU 랙 내에서 사용한 경우의 I/O 리프레시 시간은 다음과 같습니다. 증설 랙 내에서 사용한 경우의 I/O 리프레시 시간은 표에 기재된 값의 약 1.5배입니다.

유니트 이름	형식	1유니트당 I/O 리프레시 시간[μs]
DC 입력 유니트(8/16점)	CJ1W-ID201/211/212	1
DC 입력 유니트(32점)	CJ1W-ID231/232/233	2
DC 입력 유니트(64점)	CJ1W-ID261/262	4
AC 입력 유니트(8점/16점)	CJ1W-IA201/111	1
인터럽트 입력 유니트(16점)	CJ1W-INT01	1
펄스 캐치 입력 유니트(16점)	CJ1W-IDP01	1
릴레이 접점 출력 유니트	CJ1W-OC201/OC211	1
트라이액 출력 유니트	CJ1W-OA201	1
트랜지스터 출력 유니트 (8점/16점)	CJ1W-OD201/202/203/ 204/211/212/213	1
트랜지스터 출력 유니트(32점)	CJ1W-OD231/232/233/234	2
트랜지스터 출력 유니트(64점)	CJ1W-OD261/262/263	4
DC24V 입력/트랜지스터 출력 유니트(16점 입력/16점 출력)	CJ1W-MD231/232/233	1
DC24V 입력/트랜지스터 출력 유니트(32점 입력/32점 출력)	CJ1W-MD261/263	2
TTL 입출력 유니트 (16점 입력/16점 출력)	CJ1W-MD563	4
B7A 인터페이스 유니트	CJ1W-B7A04	4
	CJ1W-B7A14	4
	CJ1W-B7A22	4

● 고기능 I/O 유니트

고기능 I/O 유니트를 CPU 랙 내에서 사용한 경우의 I/O 리프레시 시간은 다음과 같습니다. 증설 랙 내에서 사용한 경우의 I/O 리프레시 시간은 표에 기재된 값의 약 1.5배입니다.

유니트 이름	형식		1유니트당 I/O 리프레시 시간[μs]
절연형 풀 멀티 입력 유니트	CJ1W-AD04U		66
아날로그 입력 유니트	CJ1W-AD041-V1/081-V1/042		24
아날로그 출력 유니트	CJ1W-DA021/041/042V/08V/08C		24
아날로그 입출력 유니트	CJ1W-MAD42		24
절연형 풀 멀티 입력 유니트	CJ1W-PH41U		80(180)*1
절연형 직류 입력 유니트	CJ1W-PDC15		60(100)*1
온도 조절 유니트	CJ1W-TC□□□		114
ID 센서 유니트	CJ1W-V680C11		76
	CJ1W-V680C12		86
고속 카운터 유니트	CJ1W-CT021		54
CompoNet 마스터 유니트	CJ1W-CRM21	통신 모드 번호 0	34
		통신 모드 번호 1	52
		통신 모드 번호 2	88
		통신 모드 번호 3	84
		통신 모드 번호 8	14 + (1.0×점유 CH 수)*2

*1. 괄호 안의 수치는 확장 할당 메모리 영역 사용 시의 I/O 리프레시 시간입니다.
*2. 점유 CH 수는 전체 슬레이브가 점유하는 I/O 영역의 CH 수입니다.

● CPU 고기능 유니트

CPU 고기능 유니트를 사용한 경우의 I/O 리프레시 시간은 다음과 같습니다. CPU 랙 내 또는 증설 랙 내 중 어느 쪽에서 사용하더라도 같은 값입니다.

유니트 이름	형식	1유니트당 I/O 리프레시 시간[μs]
시리얼 커뮤니케이션 유니트	CJ1W-SCU42 CJ1W-SCU32 CJ1W-SCU22	계수*1 ×25*2
DeviceNet 유니트	CJ1W-DRM21	계수*1×(점유 CH 수*3 +25)
EtherNet/IP 유니트	CJ1W-EIP21	7.0 + 0.8×태그 수*4

*1. CPU 유니트의 형식에 따라 다음 계수를 사용합니다.
NJ501-□□□□: 0.1
NJ301-□□□□: 0.2
NJ101-□□□□: 0.3
*2. 프로토콜 매크로를 실행하는 경우는 최대 다음 값이 더해집니다.
계수×리프레시 CH 수[μs]
*3. 점유 CH 수는 전체 슬레이브가 점유하는 I/O 영역의 CH 수입니다.
*4. 유니트 버전 Ver.1.03 이상인 CPU 유니트를 사용하는 경우에 EtherNet/IP 유니트의 I/O 리프레시 시간은 더해지지 않습니다.

사용자 프로그램 실행 시간

사용자 프로그램 실행 시간은 명령 종류×명령 수에 좌우됩니다.

기준으로는 크게 다음 3가지로 분류한 명령 수를 계측 또는 예상합니다.
- 표준적인 명령
- 배정도 실수(LREAL) 형식 사칙연산 명령
- 배정도 실수(LREAL) 형식 삼각 함수 명령

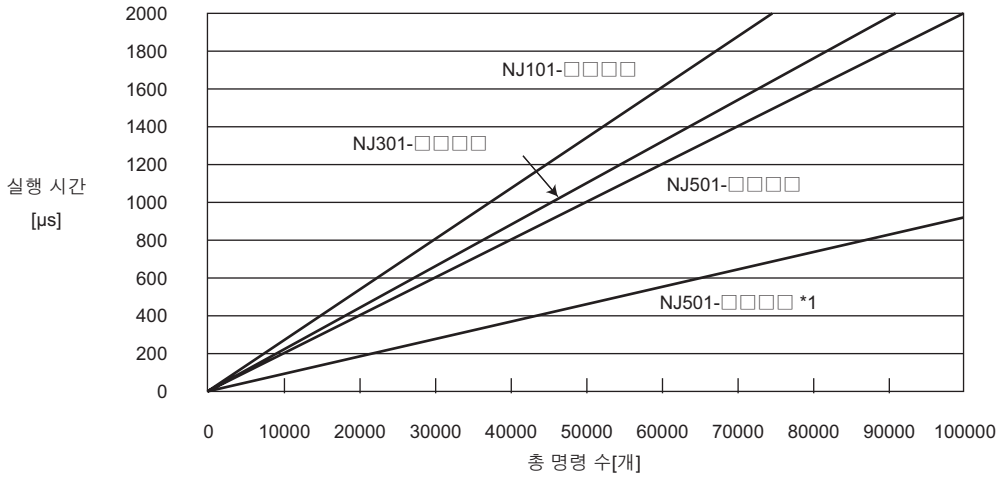
또한, 래더도 언어와 ST 언어에서는 사용하는 명령이 다릅니다. 각각의 명령 구성에 대해서는 "래더도 언어의 표준적인 명령'의 명령 종류 구성(A-56페이지)' 및 "ST 언어의 표준적인 명령'의 명령 종류 구성(A-58페이지)'을 참조하십시오.

● 간단한 계산

각 분류의 명령 수를 기준으로 다음 그래프에서 다음 각 명령의 실행 시간을 읽고 합계를 구합니다.
- 표준적인 명령의 실행 시간
- 배정도 실수(LREAL) 형식 사칙연산 명령의 실행 시간
- 배정도 실수(LREAL) 형식 삼각 함수 명령의 실행 시간

이를 통해 사용자 프로그램의 실행 시간을 추측할 수 있습니다.

래더도 언어의 표준적인 명령의 실행 시간



*1. 하드웨어 개정 번호가 'A'인 유니트의 경우입니다. 그 외에는 하드웨어 개정 번호가 '없음'인 유니트의 경우입니다.

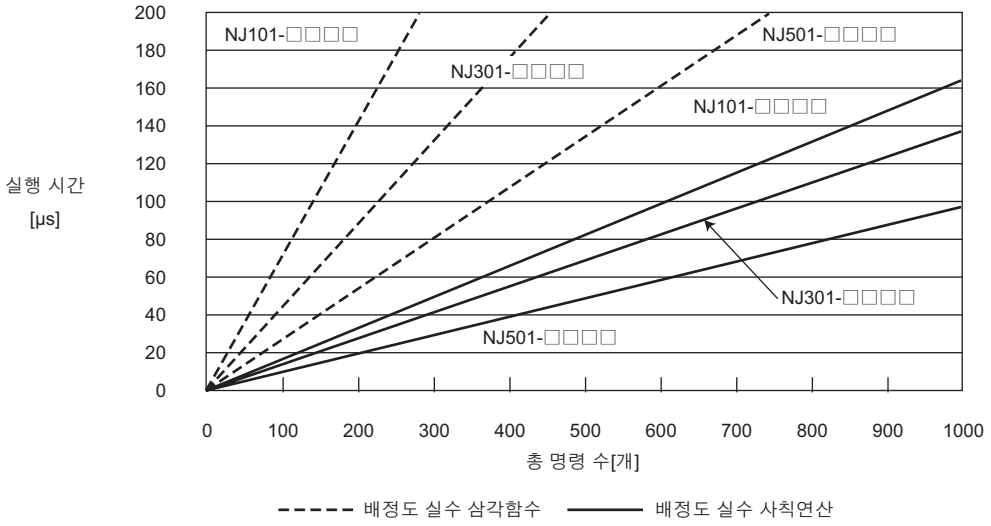
- '래더도 언어의 표준적인 명령'의 명령 종류 구성

이 구성에서의 명령 실행률은 20%입니다.

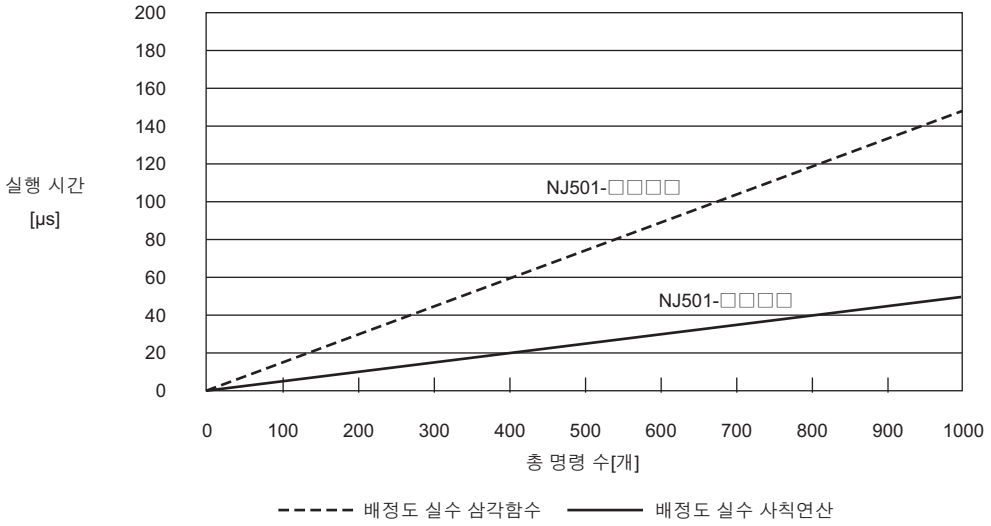
명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
래더도 명령	LD, AND, OUT, SET, RESET	81.0	40.2
비교 명령	EQ, LT	4.1	8.3
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	1.6	7.3
산술 명령	+, -, *, /, ADD, SUB, MUL, DIV	2.4	6.5
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.2
비트 열 연산 명령	AND, OR	6.2	13.0
데이터 전송 명령	MOVE	4.6	23.5
합계		100.0	100.0

래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간

- 하드웨어 개정 번호가 '없음'인 유니트의 경우



- 하드웨어 개정 번호가 'A'인 유니트의 경우



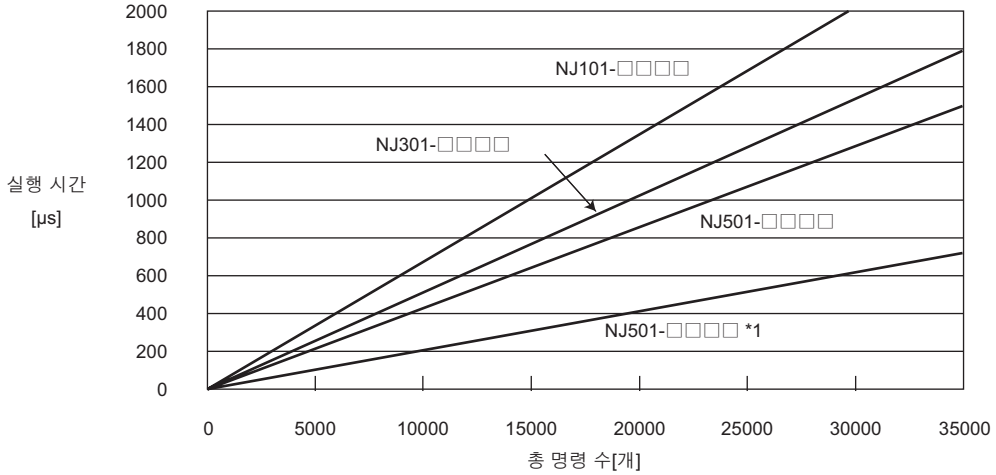
- '배정도 실수 사칙연산'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

- '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

ST 언어의 표준적인 명령의 실행 시간



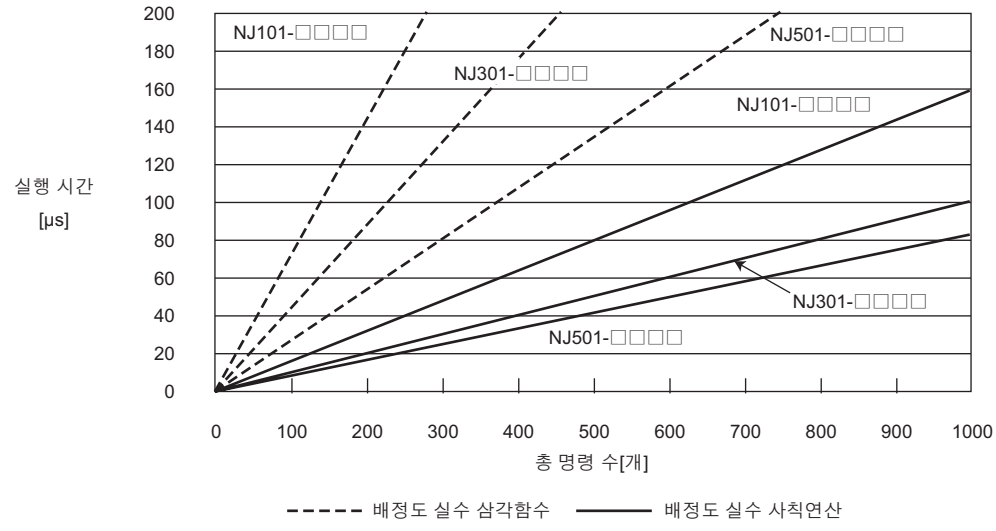
*1. 하드웨어 개정 번호가 'A'인 유니트의 경우입니다. 그 외에는 하드웨어 개정 번호가 '없음'인 유니트의 경우입니다.

- 'ST 언어의 표준적인 명령'의 명령 종류 구성

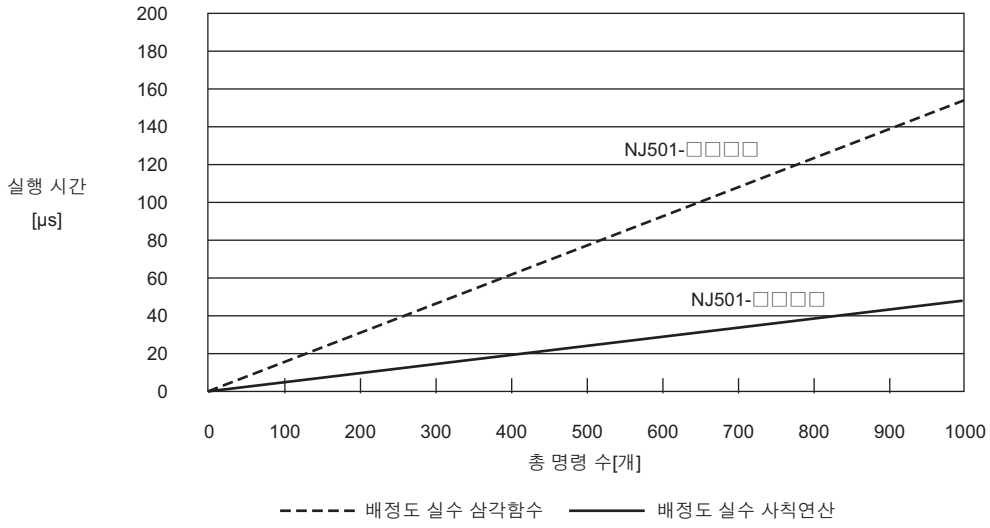
명령의 종류	명령명	명령 개수의 비율 [%]	명령 종류별 실행 시간 비율[%]
ST 구문 명령	IF ELSIF END_IF	75.4	41.6
비교 명령	EQ, LT	5.2	8.7
타이머 명령·카운터 명령	Timer, TON/TOF, CTU/CTD	2.1	18.8
산술 명령	+, -, *, /	3.1	10.2
BCD 변환 명령·데이터 변환 명령	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.6
비트 열 연산 명령	AND, OR	8.0	11.7
데이터 전송 명령	:=	5.9	7.3
합계		100.0	100.0

ST 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간

- 하드웨어 개정 번호가 '없음'인 유니트의 경우



• 하드웨어 개정 번호가 'A'인 유니트의 경우



• '배정도 실수 사칙연산'의 구성

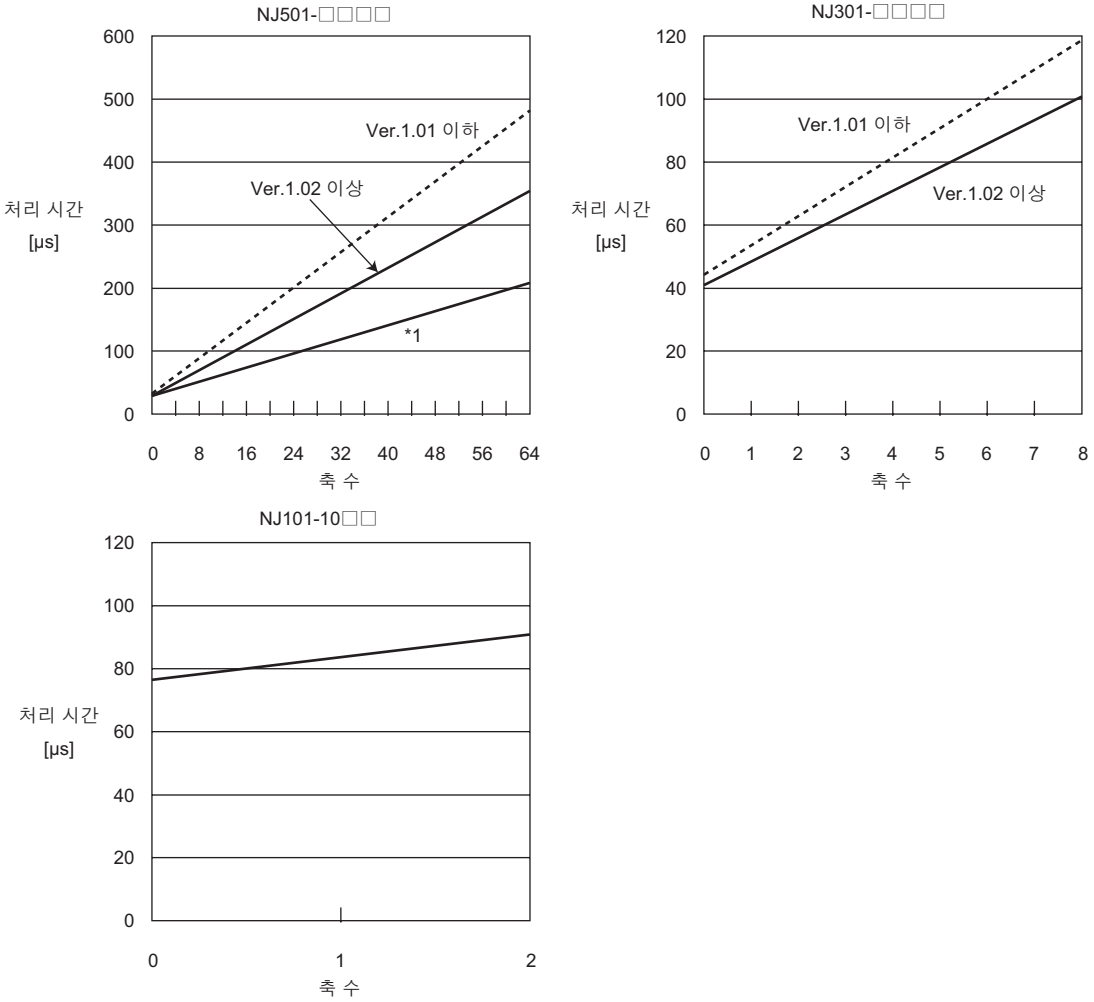
명령명	명령 개수의 비율[%]
배정도 실수 덧셈	20.0
배정도 실수 뺄셈	20.0
배정도 실수 곱셈	30.0
배정도 실수 나눗셈	30.0
합계	100.0

• '배정도 실수 삼각 함수'의 구성

명령명	명령 개수의 비율[%]
배정도 실수 sin	16.7
배정도 실수 Cos	16.7
배정도 실수 Tan	16.7
배정도 실수 Sin^{-1}	16.7
배정도 실수 Cos^{-1}	16.7
배정도 실수 Tan^{-1}	16.7
합계	100.0

모션 제어 처리 시간

모션 제어 처리 시간은 사용하는 서보의 축 수(가상 축 포함)에 좌우됩니다.
사용하는 서보 축 수를 기준으로 다음 그래프에서 모션 제어 처리 시간을 읽습니다.



*1. 하드웨어 개정 번호가 'A'인 유니트의 경우입니다. 그 외에는 하드웨어 개정 번호가 '없음'인 유니트의 경우입니다.

(주의) NJ101-90□□에서는 모션 제어 기능을 사용할 수 없습니다.

공통 처리 시간

공통 처리 시간은 시스템 오버헤드, 시스템 공통 처리 1 및 시스템 공통 처리 2를 합하여 다음과 같은 값이 됩니다. 공통 처리 시간은 태스크 종류에 따라 다릅니다.

태스크 종류	공통 처리 시간[μs](참고값)		
	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
프라이머리 정주기 태스크	150*1	240*2	300
정주기 태스크	10	32	82

*1. CPU 유니트의 유니트 버전 Ver.1.01 이하인 경우에는 265μs입니다.
*2. CPU 유니트의 유니트 버전 Ver.1.01 이하인 경우에는 360μs입니다.

A-5-2 태스크 실제 처리 시간의 평균값 계산과 태스크 주기 설정의 예

태스크 실제 처리 시간의 평균값 계산 예

하드웨어 개정 번호가 '없음'인 NJ501-□□□□의 유니트 버전 Ver.1.02를 사용하는 경우에 다음 조건일 때의 태스크 실제 처리 시간의 평균값을 구합니다.
태스크 종류는 프라이머리 정주기 태스크로 합니다.

항목		조건
사용하는 슬레이브/유니트	EtherCAT 슬레이브	• 입력 슬레이브 GX-ID1611(Ver.1.1): 1대 • 출력 슬레이브 GX-OD1611(Ver.1.1): 1대 • 서보 드라이버 R88D-1SN□□□-ECT: 4대
	CJ 유니트 (CPU 랙 내)	• DC 입력 유니트 CJ1W-ID211: 1대 • 트랜지스터 출력 유니트 CJ1W-OD211: 1대 • 아날로그 입력 유니트 CJ1W-AD042: 1대 • 아날로그 출력 유니트 CJ1W-DA021: 1대 • 시리얼 커뮤니케이션 유니트 CJ1W-SCU42: 1대(프로토콜 매크로 사용 없이)
사용자 프로그램:	사용 언어	래더도 언어
	표준적인 명령 구성	명령 수: 5,000개
	배정도 실수 사칙연산	명령 수: 200개
	배정도 실수 삼각 함수	명령 수: 100개
모션 제어 처리	축 수	4축

(주의) 내장 EtherCAT 포트에 연결한 케이블의 총 길이는 10[m]로 합니다.

● I/O 리프레시 시간

I/O 리프레시 오버헤드 시간:

I/O 리프레시 오버헤드 시간은 EtherCAT 슬레이브와 CJ 유니트가 연결되어 있으므로 65μs입니다.

EtherCAT 슬레이브 처리 시간:

CPU 유니트 NJ501-□□□□를 사용한 경우의 1바이트당 출력 처리 시간 Tout, 1바이트당 입력 처리 시간 Tin, 1바이트당 리프레시 처리 시간 Tref의 값은 다음과 같습니다.

1바이트당 출력 처리 시간 Tout[μs]	1바이트당 입력 처리 시간 Tin[μs]	1바이트당 리프레시 처리 시간 Tref[μs]	EtherCAT 통신 조정 시간 Tec[μs]
0.004	0.011	0.082	55

입력 슬레이브GX-ID1611(Ver.1.1), 출력 슬레이브GX-OD1611(Ver.1.1), 서보 드라이버 R88D-1SN□□□-ECT의 출력 프로세스 데이터 크기 Dout, 입력 프로세스 데이터 크기 Din, 출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 Dinout의 값은 다음과 같습니다.

EtherCAT 슬레이브	출력 프로세스 데이터 크기 Dout[바이트]	입력 프로세스 데이터 크기 Din[바이트]	출력 데이터 크기와 입력 데이터 크기 중 큰 쪽 Dinout
GX-ID1611(Ver.1.1)	0	3	3
GX-OD1611(Ver.1.1)	2	1	2
R88D-1SN□□□-ECT	23	26	26

Dout, Din 및 Dinout 각각의 합계는 다음과 같은 값입니다.

Dout = 2 + 23×4 = 94[바이트]

Din = 3 + 1 + 26×4 = 108[바이트]

Dinout = 3 + 2 + 26×4 = 109[바이트]

이 값들을 이용해 I/O 리프레시 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 처리 시간

=EtherCAT 슬레이브 처리 시간

=Tout×Dout + Tin×Din + Tref×Dinout + (1.24×Snum + 0.01×Clen - Tec)

=0.004×94 + 0.011×108 + 0.082×109 + (1.24×6 + 0.01×10 - 55)

=0.376 + 1.188 + 8.938 + (7.44 + 0.1 - 55)

=10.502 + 0*1

≒11[μs]

*1. 괄호 안의 계산 결과가 음수이므로 0μs로 간주합니다.

CJ 유니트 처리 시간:

CPU 랙 내의 DC 입력 유니트 CJ1W-ID211, 트랜지스터 출력 유니트 CJ1W-OD211, 아날로그 입력 유니트 CJ1W-AD042, 아날로그 출력 유니트 CJ1W-DA021, 시리얼 커뮤니케이션 유니트 CJ1W-SCU42의 1유니트당 I/O 리프레시 시간은 다음과 같습니다.

형식	1유니트당 I/O 리프레시 시간[μs]
CJ1W-ID211	1
CJ1W-OD211	1
CJ1W-AD042	24
CJ1W-DA021	24
CJ1W-SCU42	2.5

연결 대수는 모두 1이므로 CJ 유니트 처리 시간은 다음 식으로 구할 수 있습니다.

CJ 유니트 처리 시간 = Σ(각 CJ 유니트의 I/O 리프레시 시간×연결 대수)-110

= 1×1 + 1×1 + 24×1 + 24×1 + 2.5×1 - 110

= - 57.5[μs]

위 식의 계산 결과가 음수이므로 CJ 유니트 처리 시간은 0μs로 간주합니다.

여기까지의 계산에서 I/O 리프레시 오버헤드 시간, EtherCAT 슬레이브 처리 시간, CJ 유니트 처리 시간은 각각 다음과 같은 결과가 나옵니다.

항목	값[μs]
I/O 리프레시 오버헤드 시간	65
EtherCAT 슬레이브 처리 시간	11
CJ 유니트 처리 시간	0

이 값들을 이용해 I/O 리프레시 시간은 다음 식으로 구할 수 있습니다.

I/O 리프레시 시간 = I/O 리프레시 오버헤드 시간+

(EtherCAT 슬레이브 처리 시간과 CJ 유니트 처리 시간 중 큰 쪽)

=65 + 11

=76[μs]

● 사용자 프로그램 실행 시간

각 그래프에서 다음 수치를 읽습니다.

- 표준적인 명령 구성
  래더도 언어의 표준적인 명령의 실행 시간 그래프에서 NJ501-□□□□의 명령 수 5000개인 사용자 프로그램 실행 시간은 100μs
- 배정도 실수 사칙연산
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NJ501-□□□□의 명령 수 200개인 사용자 프로그램 실행 시간은 20μs
- 배정도 실수 삼각 함수
  래더도 언어의 배정도 실수(LREAL) 형식 사칙연산 명령과 삼각 함수 명령의 실행 시간 그래프에서 NJ501-□□□□의 명령 수 100개인 사용자 프로그램 실행 시간은 27μs

따라서 사용자 프로그램 실행 시간은 상기의 값을 합산하여 다음과 같은 값이 됩니다.

사용자 프로그램 실행 시간=100 + 20 + 27
=147[μs]

● 모션 제어 처리 시간

모션 제어 처리의 실행 시간 그래프에서 NJ501-□□□□ Ver.1.02의 4축 모션 제어 처리의 실행 시간을 46μs로 읽습니다.

● 공통 처리 시간

태스크 종류가 프라이머리 정주기 태스크이므로 NJ501-□□□□의 공통 처리 시간은 150μs입니다.

따라서 태스크 실행 시간의 평균값은 다음 식으로 구할 수 있습니다.

태스크 실제 처리 시간의 평균값=I/O 리프레시 시간+사용자 프로그램 실행 시간
+모션 제어 처리+공통 처리 시간
=76 + 147 + 46 + 150
=419[μs]

태스크 주기의 설정 예

위에서 구한 태스크 실제 처리 시간의 평균값을 바탕으로 태스크 주기를 설정합니다. 태스크 종류는 프라이머리 정주기 태스크입니다.
태스크 주기의 값은 계산으로 구한 태스크 실제 처리 시간의 평균값보다 크게 합니다. 그 때 태스크 주기 값을 여유 있게 태스크 실제 처리 시간의 평균값의 1.1배보다 크게 하십시오.

태스크 주기≧태스크 실제 처리 시간의 평균값×1.1

위에서 구한 태스크 실제 처리 시간의 평균값은 419μs이므로 태스크 주기는 그 1.1배보다 큰 값인 500μs로 설정합니다.

실제 컨트롤러에서는 사용자 프로그램에 기재된 로직의 동작 유무, 통신 명령의 발생 및 데이터 링크의 동작 유무, 데이터 트레이스 등 각종 기능의 사용 유무 등에 따라 태스크 실행 시간은 변동됩니다. 또한, 정주기 태스크인 경우는 그보다 실행 우선도가 더 높은 태스크의 동작 유무에 따라서도 태스크 실행 시간이 변동됩니다.
실제 기기의 컨트롤러를 사용하여 태스크 실행 시간 모니터링을 바탕으로 반드시 태스크 실행 시간을 검증하십시오.

A-6 시스템 정의 변수 목록

시스템에서 특정 기능이 할당된 변수입니다.
사전에 글로벌 변수 테이블 또는 각 POU의 로컬 변수 테이블에 등록되어 있습니다.
변수명은 고정입니다. "_(언더 바)"로 시작하는 것과 "P_"로 시작하는 것으로 이루어집니다.

읽기 전용인 시스템 정의 변수와 읽기/쓰기 가능한 시스템 정의 변수가 있습니다.

사용자 프로그램, 외부로부터의 통신 또는 Sysmac Studio, NS/NA 시리즈 표시기에서 읽기/쓰기가 가능합니다.

시스템 정의 변수는 기본적으로 기능 모듈별로 분류합니다.
변수명은 다음 카테고리명으로 시작합니다.

기능 모듈	카테고리명
NJ/NX 시리즈 컨트롤러 시스템 전체	_(없음)
PLC 기능 모듈	_PLC
	_CJB
NX 버스 기능 모듈	_NXB
모션 제어 기능 모듈	_MC, _MC1, _MC2
EtherCAT 마스터 기능 모듈	_EC
EtherNet/IP 기능 모듈	_EIP, _EIP1, _EIP2

목록표를 보는 방법은 다음과 같습니다.

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
[시스템 정의 변수명입니다. 앞에 카테고리명이 붙습니다.]	[이름입니다.]	[기능을 설명합니다.]	[변수의 데이터 형식을 나타냅니다.]	[변수의 값의 범위를 나타냅니다.]	[개별 사양표의 페이지를 나타냅니다.]

변수명 란의 괄호 안에는 해당 시스템 정의 변수가 추가된 CPU 유니트의 유니트 버전을 나타냅니다.

사용상의 주의 사항

컨트롤러의 시리즈에 따라 지원하지 않거나, 배열 수 등 사양이 다른 시스템 정의 변수가 있습니다. 세부 내용은 'A-7 시스템 정의 변수의 개별 사양(A-114페이지)'에서 개별 사양을 확인하십시오.

A-6-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음)

● 기능 분류: 시계 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_CurrentTime	시스템 시각	CPU 유니트의 내부 시계 데이터가 저장됩니다.	DATE_AND_TIME	• NX 시리즈 CPU 유니트의 경우 DT#1970-01-01-00:00:00~DT#2069-12-31-23:59:59 • NJ 시리즈 CPU 유니트의 경우 DT#1970-01-01-00:00:00~DT#2106-02-06-23:59:59	A-115

● 기능 분류: 태스크 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_<태스크 이름>_Active	태스크 실행 중 플래그	태스크 실행 중일 때 TRUE가 됩니다. 태스크 실행 중이 아닐 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-115
_<태스크 이름>_LastExecTime	이전 태스크 실행 시간	이전에 실행한 태스크 실행 시간(0.1μs 단위)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	TIME	데이터 형식에 따름	A-115
_<태스크 이름>_MaxExecTime	태스크 실행 시간 최대값	태스크 실행 시간의 최대값(0.1μs)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	TIME	데이터 형식에 따름	A-115
_<태스크 이름>_MinExecTime	태스크 실행 시간 최소값	태스크 실행 시간의 최소값(0.1μs)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	TIME	데이터 형식에 따름	A-115
_<태스크 이름>_ExecCount	태스크 실행 횟수	태스크의 실행 횟수를 표시합니다. 4294967295회를 초과하면 0으로 돌아가고, 카운트를 계속합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	UDINT	데이터 형식에 따름	A-116

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_<태스크 이름>_Exceeded	태스크 주기 초과 플래그	태스크 주기를 초과하면 TRUE가 됩니다. 태스크 주기 이내에 태스크 실행이 완료되면 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-116
_<태스크 이름>_ExceedCount	태스크 주기 초과 횟수	주기를 초과한 횟수가 저장됩니다. 현재값이 데이터 형식의 최대값을 초과하면 현재값을 0으로 되돌려서 카운트를 계속합니다. 4294967295회를 초과하면 0으로 돌아가고, 카운트를 계속합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.	UDINT	데이터 형식에 따름	A-116

● 기능 분류: 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_ErrSta	컨트롤러 이상 Status	컨트롤러 이상 발생 중에는 TRUE가 됩니다. 컨트롤러 이상 미발생 중에는 FALSE가 됩니다. (주의) 실시간성과 각 기능 모듈의 이상 요인을 나타내는 Status와의 동시성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#C0F0	A-117
_AlarmFlag	사용자 이상 Status	사용자 이상 발생 중에는 이벤트 중요도에 해당하는 비트가 TRUE가 됩니다. User fault Level 1~8이 비트 00~07에 대응합니다. 사용자 이상 미발생 중에는 16진수 0000이 됩니다.	WORD	16#0000~16#00FF	A-117

● 기능 분류: SD 메모리 카드 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_Card1Ready	SD 메모리 카드 사용 가능 플래그	SD 메모리 카드가 인식되었을 때 TRUE가 됩니다. 인식하지 못했을 때 FALSE가 됩니다. TRUE: 사용 가능 FALSE: 사용불가	BOOL	TRUE, FALSE	A-117
_Card1Protect	SD 메모리 카드 쓰기 방지 플래그	SD 메모리 카드에 LOCK 스위치에 의한 쓰기 방지가 걸려있을 때 TRUE가 됩니다. TRUE: 쓰기 방지 있음 FALSE: 쓰기 방지 없음	BOOL	TRUE, FALSE	A-117
_Card1Err	SD 메모리 카드 에러 플래그	사용이 불가능한 SD 메모리 카드가 장착됐을 때 또는 포맷에 이상이 발생했을 때 TRUE가 됩니다. TRUE: 에러 있음 FALSE: 에러 없음	BOOL	TRUE, FALSE	A-118
_Card1Access	SD 메모리 카드 액세스 중 플래그	SD 메모리 카드에 액세스 중에 TRUE가 됩니다. TRUE: 액세스 중 FALSE: 액세스 중이 아님 이 플래그는 100ms마다 시스템이 갱신합니다. 이로 인해 최대 100ms, SD 메모리 카드에 대한 액세스 상태인 이 플래그로의 반영이 지연됩니다. 따라서 사용자 프로그램 상에서 사용하는 것은 권장하지 않습니다.	BOOL	TRUE, FALSE	A-118
_Card1Deteriorated	SD 메모리 카드 수명 경고 플래그	SD 메모리 카드의 수명을 감지했을 때 TRUE가 됩니다. TRUE: 수명 감지 있음 FALSE: 수명 감지 없음	BOOL	TRUE, FALSE	A-118
_Card1PowerFail	SD 메모리 카드 액세스 중 전원 OFF 플래그	SD 메모리 카드 액세스 중에 CPU 유니트의 전원 OFF가 발생했을 때 TRUE가 됩니다. TRUE: SD 메모리 카드 액세스 중에 전원 OFF가 있었음 FALSE: 정상	BOOL	TRUE, FALSE	A-118

변수명 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
_Card1BkupCmd (Ver.1.03)	SD 메모리 카드 백업 지시		_sBKUP_ CMD		A-118
ExecBkup	백업 기동 플래그	컨트롤러의 데이터를 SD 메모리 카드로 백업할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-118
CancelBkup	백업 취소 플래그	SD 메모리 카드로의 백업을 중지할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-119
ExecVefy	비교 기동 플래그	컨트롤러와 SD 메모리 카드 내의 백업 파일을 비교할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-119
CancelVefy	비교 취소 플래그	컨트롤러와 SD 메모리 카드 내의 백업 파일 비교를 중지할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-119
DirName	디렉토리명	백업 대상인 SD 메모리 카드의 디렉토리명을 지정합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	STRING(64)	데이터 형식에 따름	A-119
_Card1BkupSta (Ver.1.03)	SD 메모리 카드 백업 Status		_sBKUP_ STA		A-119
Done	완료 플래그	백업 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-119
Active	실행 중 플래그	백업 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120
Err	에러 플래그	백업 처리가 이상 종료했을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120

변수명 / 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
_Card1VefySta (Ver.1.03)	SD 메모리 카드 비교 Status		_sVEFY_STA		A-120
Done	완료 플래그	비교 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120
Active	실행 중 플래그	비교 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120
VefyRslt	비교 결과 플래그	비교 일치일 때 TRUE, 비교 불일치일 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120
Err	에러 플래그	비교 처리가 이상 종료되었을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.	BOOL	TRUE, FALSE	A-120
_Card1PrgTransferCmd(Ver.1.11)	SD 메모리 카드 프로그램 전송 지시		_sPRGTRANSFER_CMD		A-121
Exec	프로그램 전송 기동 플래그	SD 메모리 카드 프로그램 전송 기능에 의해 SD 메모리 카드의 백업 파일 데이터를 컨트롤러로 전송할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-121
DirName	디렉토리명	프로그램 전송 대상인 백업 파일이 저장된 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING(64)	데이터 형식에 따름	A-121
Password	비밀번호	프로그램 전송 시작 시에 비교되는 비밀번호를 지정합니다. SD 메모리 카드 프로그램 전송을 시작할 때마다 비밀번호가 초기화됩니다.	STRING(33)	데이터 형식에 따름	A-121
TargetUserProgram	사용자 프로그램·설정 전송 플래그	프로그램 전송에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. SD 메모리 카드 프로그램 전송 기능에서는 반드시 TRUE로 하십시오.	BOOL	TRUE, FALSE	A-122
TargetIPAdr	IP 주소 전송 플래그	프로그램 전송에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 종류, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.	BOOL	TRUE, FALSE	A-122
TargetVariable	변수의 현재값(유지 속성만 해당) 전송 플래그	프로그램 전송에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-122
TargetMemory	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그	프로그램 전송에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-122
_Card1PrgTransferSta(Ver.1.11)	SD 메모리 카드 프로그램 전송 Status		_sPRGTRANSFER_STA		A-123

변수명 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
Done	완료 플래그	프로그램 전송 실행 완료일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-123
Active	실행 중 플래그	프로그램 전송 실행 중일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-123
Err	에러 플래그	프로그램 전송이 이상 종료됐을 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-123
_Card1RestoreCmd (Ver.1.14)	SD 메모리 카드 복원 지시		_sRESTORE_CMD		A-123
Exec	복원 기동 플래그	SD 메모리 카드 내의 백업 파일의 데이터를 컨트롤러로 복원할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-123
DirName	디렉토리명	시스템 정의 변수에 의한 복원 대상인 백업 파일이 저장된 SD 메모리 카드의 디렉토리명을 지정합니다.	STRING(64)	데이터 형식에 따름	A-123
Password	비밀번호	시스템 정의 변수에 의한 복원 시작 시에 비교될 비밀번호를 지정합니다. 시스템 정의 변수에 의한 복원을 시작할 때마다 비밀번호가 초기화됩니다.	STRING(33)	데이터 형식에 따름	A-124
_Card1RestoreSta (Ver.1.14)	SD 메모리 카드 복원 Status		_sRESTORE_STA		A-124
Done	완료 플래그	복원 실행 완료일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-124
Active	실행 중 플래그	복원 실행 중일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-124
Err	에러 플래그	복원이 이상 종료됐을 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-124

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_Card1RestoreCmdTargetUserProgram (Ver.1.14)	사용자 프로그램·설정 전송 플래그	시스템 정의 변수에 의한 복원에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. 시스템 정의 변수에 의한 복원일 때는 반드시 TRUE로 하십시오.	BOOL	TRUE, FALSE	A-124
_Card1RestoreCmdTargetIPAdr (Ver.1.14)	IP 주소 전송 플래그	시스템 정의 변수에 의한 복원에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 종류, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다.	BOOL	TRUE, FALSE	A-125
_Card1RestoreCmdTargetVariable (Ver.1.14)	변수의 현재값(유지 속성만 해당) 전송 플래그	시스템 정의 변수에 의한 복원에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-125
_Card1RestoreCmdTargetMemory (Ver.1.14)	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그	시스템 정의 변수에 의한 복원에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-125
_Card1RestoreCmdTargetUnitConfig (Ver.1.14)	유니트·슬레이브 설정 전송 플래그	시스템 정의 변수에 의한 복원에서 유니트·슬레이브 설정을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-125
_Card1RestoreCmdTargetAbsEncoder (Ver.1.14)	앱솔루트 엔코더 원점 위치 오프셋 전송 플래그	시스템 정의 변수에 의한 복원에서 앱솔루트 엔코더 원점 위치 오프셋을 전송 대상으로 할 때 TRUE로 합니다.	BOOL	TRUE, FALSE	A-126

● 기능 분류: 백업 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_BackupBusy (Ver.1.03)	백업 관련 기능 실행 중 플래그	백업, 복원, 비교 실행 중일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-126

● 기능 분류: 전원 관리 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_PowerOnHour	통전 시간	통전 시간을 표시합니다. CPU 유니트의 통전 시간을 1시간 단위로 저장합니다. 리셋하려면 0을 저장하십시오. 4294967295가 되면, 그 이상은 갱신되지 않습니다. 전원 ON 시에 초기화되지 않습니다.	UDINT	0~4294967295	A-126
_PowerOnCount	전원 OFF 발생 횟수	전원 OFF 발생 횟수를 표시합니다. CPU 유니트의 최초 전원 투입부터 전원 OFF 발생마다 누적(+1)됩니다. 리셋하려면 0을 저장하십시오. 4294967295가 되면, 그 이상은 갱신되지 않습니다. 전원 ON 시에 초기화되지 않습니다.	UDINT	0~4294967295	A-126
_RetainFail	전원 OFF 유지 실패 플래그	다음의 경우에 TRUE(전원 OFF 유지 실패)가 됩니다. • 전원 투입 시, 배터리 백업 메모리의 확인에서 에러일 때 다음의 경우에 FALSE(전원 OFF 유지 실패 없음)가 됩니다. • 전원 투입 시, 배터리 백업 메모리의 메모리 확인에서 OK일 때 • 사용자 프로그램이 다운로드 되었을 때 • 메모리 전체 클리어를 실행했을 때 (주의) 앱솔루트 엔코더 원점 위치 오프셋 데이터를 유지하지 못했을 때는 이 플래그가 아니라 축 변수의 이상 Status에 반영됩니다.	BOOL	TRUE, FALSE	A-127

● 기능 분류: 프로그래밍 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
P_On	상시 ON 플래그	항상 TRUE를 나타내는 플래그입니다.	BOOL	TRUE	A-127
P_Off	상시 OFF 플래그	항상 FALSE를 나타내는 플래그입니다.	BOOL	FALSE	A-127
P_CY	캐리 플래그	일부 명령으로 갱신하는 플래그입니다.	BOOL	TRUE, FALSE	A-127
P_First_RunMode	운전 시작 시 1주기 ON 플래그	컨트롤러의 동작 모드를 프로그램 모드에서 운전 모드로 변경했을 때 해당 프로그램이 실행 중인 경우에 1태스크 주기만 TRUE가 됩니다. 해당 프로그램이 실행 중이 아니면 FALSE로 유지됩니다. CPU 유니트가 운전을 시작했을 때 초기 처리를 수행하는 경우 등에 사용합니다. (주의) 이 시스템 정의 변수는 펑션 내에서는 사용할 수 없습니다.	BOOL	TRUE, FALSE	A-127
P_First_Run (Ver.1.08)	프로그램 기동 시 1주기 ON 플래그	해당 프로그램이 실행을 시작했을 때 1태스크 주기만 TRUE가 됩니다. 프로그램이 실행을 시작할 때마다 초기 처리를 수행하는 경우 등에 사용합니다. (주의) 이 시스템 정의 변수는 펑션 내에서는 사용할 수 없습니다.	BOOL	TRUE, FALSE	A-128
P_PRGER	명령 에러 발생 플래그	해당 프로그램 또는 해당 프로그램에서 호출한 FB나 FUN에서 명령 에러가 발생했을 때 계속 TRUE가 되는 플래그입니다. 한번 TRUE가 되면 사용자 프로그램에 의해 FALSE가 될 때까지 TRUE로 유지됩니다.	BOOL	TRUE, FALSE	A-128

● 기능 분류: 통신 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_Port_numUsingPort	사용 포트 수	현재 사용 중인 내부 논리 포트 수를 표시합니다. 사용자 프로그램의 디버그 시 등에 사용합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	USINT	0~32	A-128
_Port_isAvailable	네트워크 통신 명령 실행 가능 플래그	사용 가능한 내부 논리 포트의 유무를 표시합니다. 사용 가능한 내부 논리 포트가 있을 때 TRUE, 없을 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-128
_FINSTCPConnSta	FINS/TCP 연결 Status	FINS/TCP 연결 Status를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~16#FFFF	A-128

● 기능 분류: 버전 관리 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_UnitVersion (Ver.1.08)	유니트 버전	CPU 유니트의 유니트 버전이 저장됩니다. 요소 번호 0에 유니트 버전의 정수 부분이 저장됩니다. 요소 번호 1에 유니트 버전의 소수 부분이 저장됩니다. 예1) 유니트 버전 Ver.1.08의 경우에 요소 번호 0에 1, 요소 번호 1에 8이 저장됩니다. 예2) 유니트 버전 Ver.1.10의 경우에 요소 번호 0에 1, 요소 번호 1에 10이 저장됩니다.	ARRAY[0..1] OF USINT	0~99	A-129
_HardwareRevision (Ver.1.11)	하드웨어 개정	CPU 유니트의 하드웨어 개정이 저장됩니다. 하드웨어 개정이 '없음'인 경우에는 '-', 그 이외의 경우는 'A'~'Z'가 저장됩니다.	STRING[2]	'-', 'A'~'Z'	A-129

● 기능 분류: 자체 진단 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_SelfTest_HighTemperature (Ver.1.10)	CPU 유니트 내부 온도 초과 플래그	CPU 유니트 내부 온도가 고온일 때 TRUE가 됩니다. (주의) NX102 CPU 유니트, NX1P2 CPU 유니트의 경우에는 항상 FALSE입니다.	BOOL	TRUE, FALSE	A-129
_SelfTest_LowBattery (Ver.1.10)	배터리 전압 저하 플래그	배터리가 분리되어 있거나, 배터리 전압이 저하되어 있을 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-129
_SelfTest_LowFanRevolution (Ver.1.10)	FAN 회전 수 저하 플래그	FAN이 분리되어 있거나, FAN 회전 수가 저하되어 있을 때 TRUE가 됩니다. (주의) NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에는 항상 FALSE입니다.	BOOL	TRUE, FALSE	A-129

● 기능 분류: PLC 내장 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_DeviceOutHoldCfg (Ver.1.13)	장치 출력 유지 설정	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력을 유지하는 경우에 16#A5A5로 합니다. 16#A5A5 이외에서는 동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력은 초기화됩니다.	WORD	16#0000~16#FFFF	A-130
_DeviceOutHoldStatus (Ver.1.13)	장치 출력 유지 상태	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력이 유지된 경우에 TRUE가 됩니다. 장치 출력 유지 설정이 16#A5A5 이외일 때 또는 전체 정지 이상 레벨의 컨트롤러 이상 발생 시에 대상 장치 출력이 초기화된 경우에 FALSE가 됩니다.	BOOL	TRUE, FALSE	A-130

A-6-2 PLC 기능 모듈(카테고리명: _PLC)

● 기능 분류: 디버그 관련

변수명 / 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
_PLC_TraceSta[0..3]			_sTRACE_STA		A-131
.IsStart	트레이스 실행 중 ON 플래그	트레이스를 시작했을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-131
.IsComplete	트레이스 완료 시 ON 플래그	트레이스 실행이 완료됐을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-131
.IsTrigger	트레이스 트리거 모니터링 플래그	트레이스 트리거 조건이 성립됐을 때 TRUE가 됩니다. 다음으로 트레이스를 시작했을 때, FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-131
.ParamErr	트레이스 파라메터 이상 플래그	트레이스 시작 시에 트레이스 설정이 이상일 때 TRUE가 됩니다. 정상일 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-131

● 기능 분류: 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_PLC_ErrSta	PLC 기능 모듈 이상 Status	PLC 기능 모듈과 관련된 컨트롤러 이상 발생 중에는 TRUE가 됩니다. PLC 기능 모듈과 관련된 컨트롤러 이상이 발생 중이지 않을 때는 FALSE가 됩니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~ 16#00F0	A-132

● 기능 분류: 옵션 보드 관련

변수명 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
_PLC_OptBoardSta (Ver.1.13)	옵션 보드 Status	옵션 보드 관련 Status를 저장합니다. 옵션 보드 형식에 관계 없이 공통으로 사용합니다. 배열 요소 1이 옵션 보드 슬롯 1, 배열 요소 2가 옵션 보드 슬롯 2에 대응합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY[1..2] OF _sOPTBOARD_STA		A-132
isDetect	옵션 보드 장착	옵션 보드 슬롯에 옵션 보드가 장착되어 있음을 표시합니다. TRUE: 장착함 FALSE: 장착하지 않음 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-132
Run	옵션 보드 정상 동작	옵션 보드가 정상적으로 동작 중인지 여부를 표시합니다. 옵션 보드의 디바이스 변수를 사용할 때나 옵션 보드용 통신 명령을 사용할 때는 이 멤버를 인터록 조건으로 합니다. TRUE: 정상적으로 동작 중 FALSE: 초기 처리 중 또는 설정 변경 중 또는 에러 발생 중 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-132
Error	옵션 보드 에러 발생	옵션 보드의 에러가 발생 여부를 표시합니다. TRUE: 에러가 발생함 FALSE: 에러가 발생하지 않음 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-133
_PLC_OptSerialErrSta (Ver.1.13)	시리얼 옵션 보드 에러 Status	시리얼 통신용 옵션 보드의 전송 에러의 에러 Status를 저장합니다. 시리얼 통신용 옵션 보드의 시리얼 통신 모드를 상위 링크(FINS)로 한 경우에만 각 멤버의 값을 갱신합니다. 위 이외의 경우에는 모든 멤버의 값을 FALSE로 합니다. 배열 요소 1이 옵션 보드 슬롯 1, 배열 요소 2가 옵션 보드 슬롯 2에 대응합니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY[1..2] OF _sOPTSERIALERR_STA		A-133

변수명 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
Error	전송 에러 발생	전송 에러 발생 유무를 표시합니다. TRUE: 패리티 에러 또는 프레이밍 에러 또는 오버런 에러가 발생 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 패리티 에러 또는 프레이밍 에러 또는 오버런 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-133
ParityErr	패리티 에러 발생	패리티 에러 발생 유무를 나타냅니다. 이 에러가 발생했다면 시리얼 통신 설정이 연결 상대 기기와 맞지 않을 가능성이 있습니다. TRUE: 패리티 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 패리티 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-134
FramingErr	프레이밍 에러 발생	프레이밍 에러 발생 유무를 표시합니다. 이 에러가 발생했다면 시리얼 통신 설정이 연결 상대 기기와 맞지 않을 가능성이 있습니다. TRUE: 프레이밍 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 프레이밍 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-134

변수명 멤버명	이름	기능	데이터 형식	값의 범위	참조 페이지
OverRun	오버런 에러 발생	오버런 에러 발생 유무를 표시합니다. 이 에러가 발생했다면 옵션 보드의 전송 속도가 너무 빠를 가능성이 있습니다. TRUE: 오버런 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 오버런 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-134

A-6-3 PLC 기능 모듈(카테고리명: _CJB)

● 기능 분류: I/O 버스의 Status 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_CJB_MaxRackNo	최대 랙 번호	컨트롤러가 인식하고 있는 증설 랙의 최대 랙 번호를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	UINT	0~3 0인 경우에는 CPU 랙만 있음	A-135
_CJB_MaxSlotNo	최대 슬롯 번호	컨트롤러가 인식하고 있는 CJ 유니트의 랙마다 (최대 슬롯 번호+1)을 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..3] OF UINT	0~10 0의 경우에 CJ 유니트 장착 없음	A-135

● 기능 분류: I/O 버스의 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_CJB_ErrSta	I/O 버스 이상 Status	I/O 버스의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~ 16#C0F0	A-135
_CJB_MstrErrSta	I/O 버스 마스터 이상 Status	I/O 버스 마스터의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~ 16#00F0	A-135
_CJB_UnitErrSta	I/O 버스 유니트 이상 Status	I/O 버스 유니트의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..3, 0..9] OF WORD	16#0000~ 16#80F0	A-136
_CJB_InRespTm	기본 입력 유니트의 입력 시상수	기본 입력 유니트의 입력 시상수를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..3, 0..9] OF UINT	0~320	A-136

● 기능 분류: CJ 유니트용 메모리의 특수 보조 릴레이 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_CJB_IOUnitInfo	기본 I/O 유니트 정보 영역	기본 I/O 유니트(부하 단락 방지 기능 장착)의 알람 출력을 표시합니다. TRUE: 부하 단락 있음 FALSE: 부하 단락 없음 (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..3, 0..9, 0..7] OF BOOL	TRUE, FALSE	A-136
_CJB_CBU00InitSta · · · _CJB_CBU15InitSta	CPU 고기능 유니트 초기화 중 플래그	CPU 고기능 유니트가 초기화 중일 때 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 유니트 번호가 대응됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-136
_CJB_SIO00InitSta · · · _CJB_SIO95InitSta	고기능 I/O 유니트 초기화 중 플래그	고기능 I/O 유니트가 초기화 중에 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 호기 번호가 대응됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-137
_CJB_CBU00Restart · · · _CJB_CBU15Restart	CPU 고기능 유니트 재시작 플래그	상승 시 CPU 고기능 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 유니트 번호에 대응합니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-137
_CJB_SIO00Restart · · · _CJB_SIO95Restart	고기능 I/O 유니트 재시작 플래그	상승 시 고기능 I/O 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 호기 번호에 대응합니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-137
_CJB_SCU00P1ChgSta _CJB_SCU00P2ChgSta · · ·	시리얼 커뮤니케이션 유니트 0호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그	해당 포트의 설정 변경 중에 TRUE가 됩니다. 설정 변경이 완료된 시점에서 FALSE가 됩니다. 명령 실행 또는 사용자 조작으로 이 비트를 TRUE로 하면, 시리얼 포트 설정 변경을 통지할 수도 있습니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-138
· · · _CJB_SCU15P1ChgSta _CJB_SCU15P2ChgSta	시리얼 커뮤니케이션 유니트 1-15호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그		BOOL	TRUE, FALSE	A-138

A-6-4 NX 버스 기능 모듈(카테고리명: _NXB)

● 기능 분류: NX 버스 기능 모듈의 Status 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_NXB_MaxUnitNo (Ver.1.13)	최대 NX 유니트 번호	NX 버스 기능 모듈이 인식하고 있는 CPU 유니트 상의 NX 유니트의 NX 유니트 번호의 최대값을 표시합니다. Sysmac Studio에 의해 유니트 구성 정보가 등록된 경우에 등록한 유니트 구성의 NX 유니트 번호의 최대값이 됩니다. 미장착 유니트로 설정된 유니트도 대상입니다. Sysmac Studio에 의해 유니트 구성 정보가 등록되지 않은 경우에 실제 유니트 구성의 유니트 번호의 최대값이 됩니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	UINT	0~32 *1 0의 경우에 NX 유니트 장착 없음	A-139
_NXB_UnitIOActiveTbl (Ver.1.13)	NX 유니트 I/O 데이터 통신 중 Status	CPU 유니트 상의 NX 유니트의 I/O 데이터가 유효한지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, 항상 TRUE가 됩니다. TRUE: NX 유니트의 I/O 데이터가 유효 FALSE: NX 유니트의 I/O 데이터가 무효 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..32] OF BOOL *2	TRUE, FALSE	A-139
_NXB_UnitMsgActiveTbl (Ver.1.13)	NX 유니트 메시지 통신 가능 Status	CPU 유니트 상의 NX 유니트가 메시지 통신 가능한지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, 항상 TRUE가 됩니다. TRUE: 메시지 통신 가능 FALSE: 메시지 통신 불가능 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..32] OF BOOL *2	TRUE, FALSE	A-139
_NXB_UnitRegTbl (Ver.1.13)	NX 유니트 등록 Status	CPU 유니트 상의 NX 유니트가 유니트 구성 정보에 등록되어 있는지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미합니다. TRUE: 등록 있음 FALSE: 등록 없음 Sysmac Studio에 의해 유니트 구성 정보가 등록되지 않은 경우에 모든 유니트가 'FALSE'가 됩니다. 미장착 유니트로 설정된 NX 유니트는 'TRUE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..32] OF BOOL *2	TRUE, FALSE	A-140

*1. NX1P2 CPU 유니트의 경우에 값의 범위는 0~8입니다.
*2. NX1P2 CPU 유니트의 경우에 데이터 형식은 ARRAY [0..8] OF BOOL입니다.

● 기능 분류: NX 버스 기능 모듈의 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_NXB_ErrSta (Ver.1.13)	NX 버스 기능 모듈 이상 Status	NX 버스 기능 모듈의 이상 Status입니다. 'NX 버스 기능 모듈 마스터 이상 Status'와 모든 NX 유니트의 'NX 버스 기능 모듈 유니트 이상 Status'를 집약한 시스템 정의 변수입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~ 16#40F2	A-140
_NXB_MstrErrSta (Ver.1.13)	NX 버스 기능 모듈 마스터 이상 Status	CPU 유니트의 NX 버스 기능 모듈에서 감지한 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~ 16#40F2	A-140
_NXB_UnitErrStaTbl (Ver.1.13)	NX 버스 기능 모듈 유니트 이상 Status	CPU 유니트 상의 NX 유니트에서 감지한 이상 Status입니다. 이 Status는 WORD 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..32] OF BOOL *1	16#0000~ 16#40F2	A-141
_NXB_UnitErrFlagTbl (Ver.1.13)	NX 유니트 이상 Status	CPU 유니트 상의 NX 유니트에서 이상이 발생했는지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, NX 버스 기능 모듈이 감지하는 이벤트가 발생했는지를 표시합니다. TRUE: 이상 있음 FALSE: 이상 없음 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..32] OF BOOL*1	TRUE, FALSE	A-141

*1. NX1P2 CPU 유니트의 경우에 데이터 형식은 ARRAY [0..8] OF BOOL입니다.

A-6-5 모션 제어 기능 모듈(카테고리명: _MC)

● 기능 분류: 모션 제어 기능 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_MC_ErrSta	모션 제어 기능 모듈의 이상 Status	모션 제어 기능 모듈에서 감지한 이상의 상태를 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#40F0	A-142
_MC_ComErrSta	공통 이상 Status	모션 제어에서 공통 처리로 감지한 이상의 상태를 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-142
_MC_AX_ErrSta	축 이상 Status	축 단위로 감지한 이상의 상태를 표시합니다. 최대 256축[*1] 을 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	ARRAY [0..255] OF WORD[*1]	16#0000~16#00F0	A-142
_MC_GRP_ErrSta	축 그룹 이상 Status	축 그룹 단위로 감지한 이상의 상태를 표시합니다. 최대 64축 그룹[*2] 을 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	ARRAY [0..63] OF WORD[*2]	16#0000~16#00F0	A-142
_MC_COM	공통 변수	모션 제어 기능 모듈 공통의 상태 모니터링을 합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오.	_sCOMMON_REF	–	A-143
_MC_GRP	축 그룹 변수	NX701 CPU 유니트의 경우에 모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 모션 제어 명령에서의 축 그룹 지정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오.	ARRAY[0..63] OF _sGROUP_REF[*3]	–	A-143

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_MC1_GRP	축 그룹 변수	모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다. '_MC1_GRP'와 '_MC_GRP'는 배열 요소 번호가 같으면 같은 값을 참조합니다.	ARRAY[0..63] OF _sGROUP_REF	–	A-143
_MC2_GRP	축 그룹 변수	모션 제어 2에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다.	ARRAY[0..63] OF _sGROUP_REF	–	A-143
_MC_AX	축 변수	NX701 CPU 유니트의 경우에 모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오.	ARRAY[0..255] OF _sAXIS_REF *4	–	A-144
_MC1_AX	축 변수	모션 제어 1에서 사용하는 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다. '_MC1_AX'와 '_MC_AX'는 배열 요소 번호가 같으면 같은 값을 참조합니다.	ARRAY[0..255] OF _sAXIS_REF	–	A-144

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_MC2_AX	축 변수	모션 제어 2에서 사용하는 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션 편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다.	ARRAY[0..255] OF _sAXIS_REF	–	A-144

*1. NX102 및 NX1P2 CPU 유니트의 경우에 최대 16축, 데이터 형식은 ARRAY [0..15] OF WORD입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 64축, 데이터 형식은 ARRAY [0..63] OF WORD입니다.
*2. NX102 및 NX1P2 CPU 유니트의 경우에 최대 8축 그룹, 데이터 형식은 ARRAY [0..7] OF WORD입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 32축 그룹, 데이터 형식은 ARRAY [0..31] OF WORD입니다.
*3. NX102 및 NX1P2 CPU 유니트의 경우에 최대 8축 그룹, 데이터 형식은 ARRAY [0..7] OF _sGROUP_REF입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 32축 그룹, 데이터 형식은 ARRAY[0..31] OF _sGROUP_REF입니다.
*4. NX102 및 NX1P2 CPU 유니트의 경우에 최대 16축, 데이터 형식은 ARRAY[0..15] OF _sAXIS_REF입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 64축, 데이터 형식은 ARRAY[0..63] OF _sAXIS_REF입니다.

A-6-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC)

● 기능 분류: EtherCAT 통신의 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EC_ErrSta	내장 EtherCAT 이상	EtherCAT 마스터 기능 모듈의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#40F0	A-145
_EC_PortErr	통신 포트 이상	EtherCAT 마스터용 통신 포트의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-145
_EC_MstrErr	마스터 이상	EtherCAT 마스터의 이상과 EtherCAT 마스터가 감지한 슬레이브의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-145
_EC_SlavErr	슬레이브 이상	EtherCAT 슬레이브 전체의 이상 Status를 집약한 시스템 정의 변수입니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-145
_EC_SlavErrTbl	슬레이브 이상 테이블	EtherCAT 슬레이브 단위의 이상 Status를 통지합니다. 이 이상 Status는 실제 시스템 구성의 슬레이브 단위 이상 상태를 표시합니다. 이 변수의 배열은 EtherCAT 슬레이브의 노드 주소(1~512)*1 에 대응하며, 이상이 발생 중인 슬레이브를 표시합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	ARRAY [1..512] OF WORD*1	16#0000~16#00F0	A-146
_EC_MacAdrErr	MAC 주소 이상	MAC 주소가 잘못된 경우 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-146
_EC_LanHwErr	통신 컨트롤러 이상	통신 컨트롤러의 하드웨어 이상을 감지한 경우 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-146
_EC_LinkOffErr	링크 오프 이상	통신 컨트롤러의 링크 미확립을 검출한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-146
_EC_NetCfgErr	네트워크 구성 정보 이상	잘못된 네트워크 구성 정보를 검출한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-146
_EC_NetCfgCmpErr	네트워크 구성 비교 이상	네트워크 구성 정보와 네트워크 실제 구성이 일치하지 않는 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147
_EC_NetTopologyErr	네트워크 구성 이상	네트워크 구성 이상(연결 대수 초과, 링 연결)이 발생한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EC_PDCommErr	프로세스 데이터 통신 이상	프로세스 데이터 통신 중에 의도하지 않은 슬레이브 이탈, 가입 또는 슬레이브 WDT 이상을 검출한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147
_EC_PDTimeoutErr	프로세스 데이터 수신 타임아웃 이상	프로세스 데이터 수신 시에 타임아웃이 발생한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147
_EC_PDSendErr	프로세스 데이터 송신 이상	프로세스 데이터 송신 이상(프로세스 데이터 통신 주기 내에 송신할 수 없음, 송신 지터 초과)이 발생한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147
_EC_SlavAdrDupErr	슬레이브 노드 주소 중복 이상	슬레이브 노드 주소가 중복된 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-147
_EC_SlavInitErr	슬레이브 초기화 이상	슬레이브 대상의 초기화 명령에서 이상이 발생한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-148
_EC_SlavAppErr	슬레이브 어플리케이션 이상	슬레이브의 어플리케이션 Status 레지스터에서 이상을 감지한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-148
_EC_MsgErr	EtherCAT 메시지 이상	메시지 기능을 지원하지 않는 슬레이브 대상으로 메시지를 전송했거나, 슬레이브로 전송한 메시지의 응답이 이상한 형식인 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-148
_EC_SlavEmergErr	비상 메시지 감지	슬레이브가 비상 메시지를 전송했음을 마스터가 감지한 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-148
_EC_IndataInvalidErr (Ver.1.13)	프로세스 데이터 입력 값 무효 이상	EtherCAT 마스터가 Operational 상태이고, 프로세스 데이터 통신을 정상적으로 실행할 수 없고, 입력 데이터 무효인 상태가 다음 중 한 기간 이상 계속된 경우에 TRUE가 됩니다. • 태스크 주기가 10ms 이 내인 경우: 100ms • 태스크 주기가 10ms를 초과하는 경우: 태스크 주기의 10배	BOOL	TRUE, FALSE	A-148
_EC_CommErrTbl	통신 이상 슬레이브 테이블	슬레이브 노드 주소 순서를 테이블로 표시합니다. 마스터에서 검출한 이상이 발생 중인 슬레이브의 해당 위치는 TRUE가 됩니다.	ARRAY [1..512] OF BOOL*2	TRUE, FALSE	A-149
_EC_CycleExceeded	EtherCAT 통신 주기 설정 이상	기동 시에 설정한 통신 주기로 통신할 수 없는 경우에 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-149

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 노드 주소는 1~192, 데이터 형식은 ARRAY [1..192] OF WORD입니다.

*2. NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.



참고

내장 EtherCAT 이상 플래그의 대표 관계

변수명	이름	변수명	이름	변수명	이름	이벤트 중요도
_EC_ErrSta	내장 EtherCAT 이상	_EC_PortErr	통신 포트 이상	_EC_MacAdrErr	MAC 주소 이상	부분 정지 이상 레벨
				_EC_LanHwErr	통신 컨트롤러 이상	
				_EC_LinkOffErr	링크 오프 이상	
		_EC_MstrErr	마스터 이상	_EC_NetCfgErr	네트워크 구성 정보 이상	가벼운 이상 레벨
				_EC_NetCfgCmpErr	네트워크 구성 비교 이상	
				_EC_NetTopologyErr	네트워크 구성 이상	
				_EC_PDCommErr	프로세스 데이터 통신 이상	
				_EC_PDTimeoutErr	프로세스 데이터 수신 타임아웃 이상	
				_EC_PDSendErr	프로세스데이터 송신 이상	
				_EC_SlavAdrDupErr	슬레이브 노드 주소 중복 이상	
				_EC_SlavInitErr	슬레이브 초기화 이상	
				_EC_SlavAppErr	슬레이브 어플리케이션 이상	
				_EC_CommErrTbl	통신 이상 슬레이브 테이블	
				_EC_CycleExceeded	EtherCAT 통신 주기 설정 이상	
				_EC_MsgErr	EtherCAT 메시지 이상	감시 정보
				_EC_SlavEmergErr	비상 메시지 감지	
		_EC_SlavErr	슬레이브 이상	_EC_SlavErrTbl	슬레이브 이상 테이블	슬레이브 측에서 정의

(주의) EtherCAT 통신의 이상과 관련된 모든 시스템 정의 변수는 이상 요인을 제거하고 발생 중인 컨트롤러 이상을 해제(Sysmac Studio의 트러블 슈팅/ResetECError 명령)할 때까지 값이 변화하지 않습니다.

● 기능 분류: EtherCAT 통신의 Status 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EC_RegSlavTbl	등록 슬레이브 테이블	네트워크 구성 정보에 등록된 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 등록된 경우에 해당 위치는 TRUE가 됩니다.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-149
_EC_EntrySlavTbl	네트워크 가입 슬레이브 테이블	네트워크 가입 슬레이브 목록을 통지합니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 네트워크 가입했을 때, 해당 위치는 TRUE가 됩니다.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-149
_EC_MBXSlavTbl	메시지 통신 가능 슬레이브 테이블	메시지 통신이 가능한 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 메시지 통신 가능 상태(Pre-Operational, Safe-Operational, Operational)로 이동한 경우에 해당 위치는 TRUE가 됩니다. (주의) EtherCAT 슬레이브와 메시지 통신하는 경우에 이 데이터를 참조하여 해당 슬레이브와 메시지 통신이 가능한지 여부를 확인하십시오.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-150
_EC_PDSlavTbl	프로세스 데이터 통신 중 슬레이브 테이블	프로세스 데이터 통신 중인 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 IN, OUT 모두 프로세스 데이터 유효 상태(Operational)로 이동한 경우에 해당 위치는 TRUE가 됩니다. (주의) EtherCAT 슬레이브를 제어하는 경우에 이 데이터를 참조하여 해당 슬레이브의 데이터 유효성을 확인하십시오.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-150
_EC_DisconnSlavTbl	이탈 명령 중 슬레이브 테이블	슬레이브 노드 주소 순서를 테이블로 표시합니다. 이탈 명령 중인 슬레이브의 해당 위치는 TRUE가 됩니다.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-150
_EC_DisableSlavTbl	무효 설정 슬레이브 테이블	슬레이브 노드 주소 순서를 테이블로 표시합니다. 무효로 설정된 슬레이브의 해당 위치는 TRUE가 됩니다.	ARRAY [1..512] OF BOOL*1	TRUE, FALSE	A-150
_EC_PDActive	프로세스 데이터 통신 Status	전체 슬레이브(*)와의 프로세스 데이터 통신 중일 때 TRUE가 됩니다. * 무효로 설정된 슬레이브는 포함하지 않습니다.	BOOL	TRUE, FALSE	A-151
_EC_PktMonStop	패킷 모니터링 정지	패킷 모니터링 정지 중일 때 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-151
_EC_LinkStatus	링크 Status	통신 컨트롤러의 링크 Status가 Link ON인 경우에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-151
_EC_PktSaving	패킷 데이터 파일 저장 중	패킷 데이터 파일 저장 처리 중인지 여부를 표시합니다. TRUE: 패킷 데이터 파일 저장 처리 중 FALSE: 패킷 데이터 파일 저장 미처리 중	BOOL	TRUE, FALSE	A-151
_EC_InDataInvalid	입력 데이터 무효	프라이머리 정주기 태스크에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다.	BOOL	TRUE, FALSE	A-151
_EC_InData1Invalid	입력 데이터 1 무효	프라이머리 정주기 태스크에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-152

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EC_InData2Invalid	입력 데이터 2 무효	정주기 태스크(실행 우선도 5)에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다. (주의) NX102 및 NX1P2 CPU 유니트의 경우에 항상 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-152

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

(주의) EtherCAT 통신 Status에 관련된 모든 시스템 정의 변수는 현재 Status를 표시합니다.

● 기능 분류: EtherCAT 통신의 진단 통계 정보 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EC_StatisticsLogEnable (Ver.1.11)	진단 통계 정보 로그 허가	진단 통계 정보 로그 기능을 시작할 때 FALSE에서 TRUE로 합니다. 진단 통계 정보 로그 기능을 종료할 때 TRUE에서 FALSE로 합니다.	BOOL	TRUE, FALSE	A-153
_EC_StatisticsLogCycleSec (Ver.1.11)	진단 통계 정보 로그 저장 주기	진단 통계 정보의 로그 기능에 대해 통계 진단 정보의 쓰기를 실시하는 주기를 초 단위로 지정합니다. '0'을 지정한 경우에 진단 통계 정보 로그 기능을 종료할 때에 1회만 쓰기를 실시합니다. (주의) 진단 통계 정보 로그 기능의 실행 중에 이 시스템 정의 변수의 값을 변경해도 실행 중인 쓰기 주기는 변경되지 않습니다.	UINT	0, 30~1800	A-153
_EC_StatisticsLogBusy (Ver.1.11)	진단 통계 정보 로그 실행 상태 플래그	진단 통계 정보 로그 기능을 실행 중에 TRUE가 됩니다.	BOOL	TRUE, FALSE	A-153
_EC_StatisticsLogErr (Ver.1.11)	진단 통계 정보 로그 이상 종료 플래그	진단 통계 정보 로그 기능 시작에 실패했거나, 실행 중에 로그를 쓸 수 없었던 경우에 TRUE가 됩니다. 진단 통계 정보 로그 기능 시작 후에 _EC_StatisticsLogBusy(진단 통계 정보 로그 실행 상태 플래그)가 FALSE가 됐을 때, 이 플래그의 값이 확정됩니다. 이상 종료에는 다음 요인이 포함됩니다. • SD 메모리 카드의 용량이 상한에 도달하여, 레코드를 로그 파일에 추가할 수 없음 • SD 메모리 카드가 쓰기 방지됨 • SD 메모리 카드가 장착되지 않음 • _EC_StatisticsLogCycleSec(진단 통계 정보 로그 저장 주기)에 지정된 값이 잘못되어 기능을 시작할 수 없음	BOOL	TRUE, FALSE	A-153

A-6-7 EtherNet/IP 기능 모듈(카테고리명: _EIP)

● 기능 분류: EtherNet/IP 통신의 이상 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_ErrSta	내장 EtherNet/IP 이상	내장 EtherNet/IP 포트의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_PortErr(통신 포트 1 이상) • _EIP2_PortErr(통신 포트 2 이상) • _EIP1_CipErr(CIP 통신 1 이상) • _EIP2_CipErr(CIP 통신 2 이상) • _EIP_TcpAppErr(TCP 어플리케이션 통신 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_PortErr(통신 포트 이상) • _EIP_CipErr(CIP 통신 이상) • _EIP_TcpAppErr(TCP 어플리케이션 통신 이상) (주의) 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-154
_EIP_PortErr	통신 포트 이상	통신 포트의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_MacAdrErr(통신 포트 1 MAC 주소 이상) • _EIP1_LanHwErr(통신 포트 1 통신 컨트롤러 이상) • _EIP1_EtnCfgErr(통신 포트 1 Ethernet 기본 설정 이상) • _EIP1_IPAdrCfgErr(통신 포트 1 IP 주소 설정 이상) • _EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) • _EIP1_BootpErr(통신 포트 1 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_MacAdrErr(MAC 주소 이상) • _EIP_LanHwErr(통신 컨트롤러 이상) • _EIP_EtnCfgErr(Ethernet 기본 설정 이상) • _EIP_IPAdrCfgErr(IP 주소 설정 이상) • _EIP_IPAdrDupErr(IP 주소 중복 이상) • _EIP_BootpErr(BOOTP 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 처리 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-154

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP1_PortErr	통신 포트 1 이상	통신 포트의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP1_MacAdrErr(통신 포트 1 MAC 주소 이상) • _EIP1_LanHwErr(통신 포트 1 통신 컨트롤러 이상) • _EIP1_EtnCfgErr(통신 포트 1 Ethernet 기본 설정 이상) • _EIP1_IPAdrCfgErr(통신 포트 1 IP 주소 설정 이상) • _EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) • _EIP1_BootpErr(통신 포트 1 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 처리 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~16#00F0	A-155
_EIP2_PortErr	통신 포트 2 이상	통신 포트의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP2_MacAdrErr(통신 포트 2 MAC 주소 이상) • _EIP2_LanHwErr(통신 포트 2 통신 컨트롤러 이상) • _EIP2_EtnCfgErr(통신 포트 2 Ethernet 기본 설정 이상) • _EIP2_IPAdrCfgErr(통신 포트 2 IP 주소 설정 이상) • _EIP2_IPAdrDupErr(통신 포트 2 IP 주소 중복 이상) • _EIP2_BootpErr(통신 포트 2 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 처리 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~16#00F0	A-155

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_CipErr	CIP 통신 이상	CIP 통신의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_IdentityErr(CIP 통신 1 Identity 정보 잘못됨) • _EIP1_TDLinkCfgErr(CIP 통신 1 태그 데이터 링크 설정 이상) • _EIP1_TDLinkOpnErr(CIP 통신 1 태그 데이터 링크 연결 확립 실패) • _EIP1_TDLinkErr(CIP 통신 1 태그 데이터 링크 통신 이상) • _EIP1_TagAdrErr(CIP 통신 1 태그 해결 이상) • _EIP1_MultiSwONErr(CIP 통신 1 복수 스위치 동시 ON 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_IdentityErr(Identity 정보가 잘못됨) • _EIP_TDLinkCfgErr(태그 데이터 링크 설정 이상) • _EIP_TDLinkOpnErr(태그 데이터 링크 연결 확립 실패) • _EIP_TDLinkErr(태그 데이터 링크 통신 이상) • _EIP_TagAdrErr(태그 해결 이상) • _EIP_MultiSwONErr(복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고, 이 이상을 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~ 16#00F0	A-156
_EIP1_CipErr	CIP 통신 1 이상	CIP 통신 1의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP1_IdentityErr(CIP 통신 1 Identity 정보가 잘못됨) • _EIP1_TDLinkCfgErr(CIP 통신 1 태그 데이터 링크 설정 이상) • _EIP1_TDLinkOpnErr(CIP 통신 1 태그 데이터 링크 연결 확립 실패) • _EIP1_TDLinkErr(CIP 통신 1 태그 데이터 링크 통신 이상) • _EIP1_TagAdrErr(CIP 통신 1 태그 해결 이상) • _EIP1_MultiSwONErr(CIP 통신 1 복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고 이 이상을 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~ 16#00F0	A-156

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_CipErr	CIP 통신 2 이상	CIP 통신 2의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP2_IdentityErr(CIP 통신 2 Identity 정보가 잘못됨) • _EIP2_TDLinkCfgErr(CIP 통신 2 태그 데이터 링크 설정 이상) • _EIP2_TDLinkOpnErr(CIP 통신 2 태그 데이터 링크 연결 확립 실패) • _EIP2_TDLinkErr(CIP 통신 2 태그 데이터 링크 통신 이상) • _EIP2_TagAdrErr(CIP 통신 2 태그 해결 이상) • _EIP2_MultiSwONErr(CIP 통신 2 복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고 이 이상을 ON합니다. 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	WORD	16#0000~16#00F0	A-156
_EIP_TcpAppErr	TCP 어플리케이션 통신 이상	TCP 어플리케이션 통신의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP_TcpAppCfgErr(TCP 어플리케이션 설정 이상) • _EIP_NTPSrvErr(NTP 서버 연결 이상) (주의) 각 비트의 의미는 뒤에 설명하는 '이상 Status 각 비트의 의미'를 참조하십시오.	WORD	16#0000~16#00F0	A-157
_EIP_MacAdrErr	MAC 주소 이상	NX 시리즈 CPU 유니트의 경우에 전원 ON 시에 통신 포트 1의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 전원 ON 시에 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상	BOOL	TRUE, FALSE	A-157
_EIP1_MacAdrErr	통신 포트 1 MAC 주소 이상	전원 ON 시에 통신 포트 1의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-157
_EIP2_MacAdrErr	통신 포트 2 MAC 주소 이상	전원 ON 시에 통신 포트 2의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-157
_EIP_LanHwErr	통신 컨트롤러 이상	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상	BOOL	TRUE, FALSE	A-158

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP1_LanHwErr	통신 포트 1 통신 컨트롤러 이상	통신 포트 1에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-158
_EIP2_LanHwErr	통신 포트 2 통신 컨트롤러 이상	통신 포트 2에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-158
_EIP_EtnCfgErr	Ethernet 기본 설정 이상	NX 시리즈 CPU 유니트의 경우에 통신 포트 1의 Ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 Ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상	BOOL	TRUE, FALSE	A-158
_EIP1_EtnCfgErr	통신 포트 1 Ethernet 기본 설정 이상	통신 포트 1의 ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-159
_EIP2_EtnCfgErr	통신 포트 2 Ethernet 기본 설정 이상	통신 포트 2의 ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-159
_EIP_IPAdrCfgErr	IP 주소 설정 이상	NX 시리즈 CPU 유니트의 경우에 통신 포트 1의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 • 기본 게이트웨이 설정이 잘못됨 FALSE: 정상	BOOL	TRUE, FALSE	A-159
_EIP1_IPAdrCfgErr	통신 포트 1 IP 주소 설정 이상	통신 포트 1의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-159

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_IPAdrCfgErr	통신 포트 2 IP 주소 설정 이상	통신 포트 2의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-160
_EIP_IPAdrDupErr	IP 주소 중복 이상	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외	BOOL	TRUE, FALSE	A-160
_EIP1_IPAdrDupErr	통신 포트 1 IP 주소 중복 이상	통신 포트 1에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-160
_EIP2_IPAdrDupErr	통신 포트 2 IP 주소 중복 이상	통신 포트 2에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-160
_EIP_DNSCfgErr*1	DNS 설정 이상	DNS 설정값 또는 hosts 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상	BOOL	TRUE, FALSE	A-161
_EIP_BootpErr	BOOTP 서버 연결 이상	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. NJ 시리즈 CPU 유니트의 경우에 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다.	BOOL	TRUE, FALSE	A-161
_EIP1_BootpErr	통신 포트 1 BOOTP 서버 연결 이상	통신 포트 1에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-161

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_BootpErr	통신 포트 2 BOOTP 서버 연결 이상	통신 포트 2에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-161
_EIP_IPRTblErr	IP 라우트 테이블 이상	NX 시리즈 CPU 유니트의 경우에 기본 게이트웨이 설정 또는 IP 라우터 테이블 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 IP 라우터 테이블 설정 또는 hosts 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상	BOOL	TRUE, FALSE	A-162
_EIP_IdentityErr	Identity 정보가 잘못됨	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 Identify 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 Identify 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상	BOOL	TRUE, FALSE	A-162
_EIP1_IdentityErr	CIP 통신 1 Identity 정보가 잘못됨	CIP 통신 1 Identity 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-162
_EIP2_IdentityErr	CIP 통신 2 Identity 정보가 잘못됨	CIP 통신 2 Identity 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-162
_EIP_TDLinkCfgErr	태그 데이터 링크 설정 이상	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상	BOOL	TRUE, FALSE	A-163
_EIP1_TDLinkCfgErr	CIP 통신 1 태그 데이터 링크 설정 이상	CIP 통신 1의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-163

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_TDLinkCfgErr	CIP 통신 2 태그 데이터 링크 설정 이상	CIP 통신 2의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-163
_EIP_TDLinkOpnErr	태그 데이터 링크 연결 확립 실패	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외	BOOL	TRUE, FALSE	A-163
_EIP1_TDLinkOpnErr	CIP 통신 1 태그 데이터 링크 연결 확립 실패	CIP 통신 1의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-164
_EIP2_TDLinkOpnErr	CIP 통신 2 태그 데이터 링크 연결 확립 실패	CIP 통신 2의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-164

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_TDLinkErr	태그 데이터 링크 통신 이상	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외	BOOL	TRUE, FALSE	A-164
_EIP1_TDLinkErr	CIP 통신 1 태그 데이터 링크 통신 이상	CIP 통신 1의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-164
_EIP2_TDLinkErr	CIP 통신 2 태그 데이터 링크 통신 이상	CIP 통신 2의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-165
_EIP_TagAdrErr	태그 해결 이상	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외	BOOL	TRUE, FALSE	A-165

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP1_TagAdrErr	CIP 통신 1 태그 해결 이상	CIP 통신 1의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-165
_EIP2_TagAdrErr	CIP 통신 2 태그 해결 이상	CIP 통신 2의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-165
_EIP_MultiSwONErr	복수 스위치 동시 ON 이상	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외	BOOL	TRUE, FALSE	A-166
_EIP1_MultiSwONErr	CIP 통신 1 복수 스위치 동시 ON 이상	CIP 통신 1에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-166
_EIP2_MultiSwONErr	CIP 통신 2 복수 스위치 동시 ON 이상	CIP 통신 2에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-166

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_TcpAppCfgErr	TCP 어플리케이션 설정 이상	TRUE: 적어도 1개 이상의 TCP 어플리케이션 기능(FTP, NTP, SNMP)의 설정값이 잘못되었거나 읽기에 실패했습니다. FALSE: 정상	BOOL	TRUE, FALSE	A-166
_EIP_NTPSrvErr	NTP 서버 연결 이상	TRUE: NTP 클라이언트가 서버 연결에 실패(타임아웃)했습니다. FALSE: NTP의 설정이 되어있지 않거나 NTP의 설정이 되어 있고 연결에 성공했습니다.	BOOL	TRUE, FALSE	A-166
_EIP_DNSSrvErr	DNS 서버 연결 이상	TRUE: DNS 클라이언트가 서버 연결에 실패(타임아웃)했습니다. FALSE: DNS의 설정이 되어있지 않거나 DNS의 설정이 되어 있고 연결에 성공했습니다.	BOOL	TRUE, FALSE	A-167

*1. NJ 시리즈 CPU 유니트에서는 CPU 유니트의 유니트 버전 Ver.1.11 이상에서 사용 가능합니다.

NJ 시리즈에서의 EtherNet/IP 이상 관련 시스템 정의 변수의 계층 관계

EtherNet/IP의 이상 관련 시스템 정의 변수는 다음 표와 같이 계층 관계입니다. 예를 들어, 제2계층의 _EIP_PortErr, _EIP_CipErr, _EIP_TcpAppErr 중 어느 값이 0이 아닌 경우에 그 상위 계층의 _EIP_ErrSta 값이 0이 아니게 됩니다. 즉, 상위 계층의 시스템 정의 변수 값을 참조하면, 그보다 하위 계층의 이상이 발생했는지 여부를 알 수 있습니다.

제1계층		제2계층		제3계층	
변수명	이름	변수명	이름	변수명	이름
_EIP_ErrSta	내장 EtherNet/IP 이상	_EIP_PortErr	통신 포트 이상	_EIP_MacAdrErr	MAC 주소 이상
				_EIP_LanHwErr	통신 컨트롤러 이상
				_EIP_EtnCfgErr	Ethernet 기본 설정 이상
				_EIP_IPAdrCfgErr	IP 주소 설정 이상
				_EIP_IPAdrDupErr	IP 주소 중복 이상
				_EIP_BootpErr	BOOTP 서버 연결 이상
				_EIP_DNSSrvErr	DNS 서버 연결 이상
				_EIP_IPRTblErr	IP 라우트 테이블 이상
		_EIP_CipErr	CIP 통신 이상	_EIP_IdentityErr	Identity 정보가 잘못됨
				_EIP_TDLinkCfgErr	태그 데이터 링크 설정 이상
				_EIP_TDLinkOpnErr	태그 데이터 링크 연결 확립 실패
				_EIP_TDLinkErr	태그 데이터 링크 통신 이상
				_EIP_TagAdrErr	태그 해결 이상
				_EIP_MultiSwONErr	복수 스위치 동시 ON
		_EIP_TcpAppErr	TCP 어플리케이션 통신 이상	_EIP_TcpAppCfgErr	TCP 어플리케이션 설정 이상
				_EIP_NTPSrvErr	NTP 서버 연결 이상

NX 시리즈에서의 EtherNet/IP 이상 관련 시스템 정의 변수의 계층 관계

EtherNet/IP의 이상 관련 시스템 정의 변수는 다음 표와 같이 계층 관계입니다. 예를 들어, 제2계층의 _EIP1_PortErr, _EIP2_PortErr, _EIP1_CipErr, _EIP2_CipErr, _EIP_TcpAppErr 중 어느 값이 0이 아닌 경우에 그 상위 계층의 _EIP_ErrSta 값이 0이 아니게 됩니다. 즉, 상위 계층의 시스템 정의 변수 값을 참조하면, 그보다 하위 계층의 이상이 발생했는지 여부를 알 수 있습니다.

제1계층		제2계층		제3계층	
변수명	이름	변수명	이름	변수명	이름
_EIP_ErrSta	내장 EtherNet/IP 이상	_EIP1_PortErr	통신 포트 1 이상	_EIP1_MacAdrErr	통신 포트 1 MAC 주소 이상
				_EIP1_LanHwErr	통신 포트 1 통신 컨트롤러 이상
				_EIP1_EtnCfgErr	통신 포트 1 Ethernet 기본 설정 이상
				_EIP1_IPAdrCfgErr	통신 포트 1 IP 주소 설정 이상
				_EIP1_IPAdrDupErr	통신 포트 1 IP 주소 중복 이상
				_EIP1_BootpErr	통신 포트 1 BOOTP 서버 연결 이상
				_EIP_DNSCfgErr	DNS 설정 이상
				_EIP_DNSSrvErr	DNS 서버 연결 이상
				_EIP_IPRTblErr	IP 라우트 테이블 이상
		_EIP2_PortErr	통신 포트 2 이상	_EIP2_MacAdrErr	통신 포트 2 MAC 주소 이상
				_EIP2_LanHwErr	통신 포트 2 통신 컨트롤러 이상
				_EIP2_EtnCfgErr	통신 포트 2 Ethernet 기본 설정 이상
				_EIP2_IPAdrCfgErr	통신 포트 2 IP 주소 설정 이상
				_EIP2_IPAdrDupErr	통신 포트 2 IP 주소 중복 이상
				_EIP2_BootpErr	통신 포트 2 BOOTP 서버 연결 이상
				_EIP_DNSCfgErr	DNS 설정 이상
				_EIP_DNSSrvErr	DNS 서버 연결 이상
				_EIP_IPRTblErr	IP 라우트 테이블 이상
		_EIP1_CipErr	CIP 통신 1 이상	_EIP1_IdentityErr	CIP 통신 1 Identity 정보가 잘못됨
				_EIP1_TDLinkCfgErr	CIP 통신 1 태그 데이터 링크 설정 이상
				_EIP1_TDLinkOpnErr	CIP 통신 1 태그 데이터 링크 연결 확립 실패
				_EIP1_TDLinkErr	CIP 통신 1 태그 데이터 링크 통신 이상
				_EIP1_TagAdrErr	CIP 통신 1 태그 해결 이상
				_EIP1_MultiSwONErr	CIP 통신 1 복수 스위치 동시 ON
		_EIP2_CipErr	CIP 통신 2 이상	_EIP2_IdentityErr	CIP 통신 2 Identity 정보가 잘못됨
				_EIP2_TDLinkCfgErr	CIP 통신 2 태그 데이터 링크 설정 이상
				_EIP2_TDLinkOpnErr	CIP 통신 2 태그 데이터 링크 연결 확립 실패
				_EIP2_TDLinkErr	CIP 통신 2 태그 데이터 링크 통신 이상
				_EIP2_TagAdrErr	CIP 통신 2 태그 해결 이상
				_EIP2_MultiSwONErr	CIP 통신 2 복수 스위치 동시 ON
		_EIP_TcpAppErr	TCP 어플리케이션 통신 이상	_EIP_TcpAppCfgErr	TCP 어플리케이션 설정 이상
				_EIP_NTPSrvErr	NTP 서버 연결 이상

(주의) 변수명이 '_EIP1'로 시작하는 시스템 정의 변수는 변수명이 '_EIP'로 시작하는 동일한 이름의 시스템 정의 변수와 같은 값을 참조합니다. 예를 들어, _EIP1_PortErr(통신 포트 1 이상)는 _EIP_PortErr(통신 포트 이상)와 같은 값을 참조합니다.

(주의) 변수명이 '_EIP2'로 시작하는 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.

● 기능 분류: EtherNet/IP 통신의 Status 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_EtnOnlineSta	온라인	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP 포트의 통신 포트 1의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP 포트의 통신 포트의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다.	BOOL	TRUE, FALSE	A-167
_EIP1_EtnOnlineSta	통신 포트 1 온라인	내장 EtherNet/IP 포트의 통신 포트 1의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-167
_EIP2_EtnOnlineSta	통신 포트 2 온라인	내장 EtherNet/IP 포트의 통신 포트 2의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-167
_EIP_TDLinkRunSta	태그 데이터 링크 통신 중	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외	BOOL	TRUE, FALSE	A-168
_EIP1_TDLinkRunSta	CIP 통신 1 태그 데이터 링크 통신 중	CIP 통신 1에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-168

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_TDLinkRunSta	CIP 통신 2 태그 데이터 링크 통신 중	CIP 통신 2에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-168
_EIP_TDLinkAllRunSta	전체 태그 데이터 링크 통신 중	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. NJ 시리즈 CPU 유니트의 경우에 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다.	BOOL	TRUE, FALSE	A-168
_EIP1_TDLinkAllRunSta	CIP 통신 1 전체 태그 데이터 링크 통신 중	CIP 통신 1의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-169
_EIP2_TDLinkAllRunSta	CIP 통신 2 전체 태그 데이터 링크 통신 중	CIP 통신 2의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-169
_EIP_RegTargetSta[255]	등록 대상 노드 정보	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP의 CIP 통신 1에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있지 않습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있지 않습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-169

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP1_RegTargetSta[255]	CIP 통신 1 등록 대상 노드 정보	내장 EtherNet/IP의 CIP 통신 1에 연결이 등록되어 있는 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있지 않습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-169
_EIP2_RegTargetSta[255]	CIP 통신 2 등록 대상 노드 정보	내장 EtherNet/IP의 CIP 통신 2에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결이 등록되어 있지 않습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-170
_EIP_EstbTargetSta[255]	정상 대상 노드 정보	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP의 CIP 통신 1이 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-170
_EIP1_EstbTargetSta[255]	CIP 통신 1 정상 대상 노드 정보	내장 EtherNet/IP의 CIP 통신 1이 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-170

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_EstbTargetSta[255]	CIP 통신 2 정상 대상 노드 정보	내장 EtherNet/IP의 CIP 통신 2가 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x' 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-170
_EIP_TargetPLCModeSta[255]	대상 PLC 동작 모드	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-171
_EIP1_TargetPLCModeSta[255]	CIP 통신 1 대상 PLC 동작 모드	내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-171

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_TargetPLCModeSta[255]	CIP 통신 2 대상 PLC 동작 모드	내장 EtherNet/IP가 오리지네이터인 CIP 통신 2의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-171
_EIP_TargetPLCErr[255]	대상 PLC 이상 정보	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-172
_EIP1_TargetPLCErr[255]	CIP 통신 1 대상 PLC 이상 정보	내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-172

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_TargetPLCErr [255]	CIP 통신 2 대상 PLC 이상 정보	내장 EtherNet/IP가 오리지네이터인 CIP 통신 2의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-172

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_TargetNodeErr [255]	대상 노드 이상 정보	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나 (등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되어 있으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었지만(등록 대상 노드 정보가 TRUE이며, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다 (대상 PLC 이상 정보가 TRUE). NJ 시리즈 CPU 유니트의 경우에 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나 (등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되어 있으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었지만(등록 대상 노드 정보가 TRUE이며, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다 (대상 PLC 이상 정보가 TRUE).	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-173

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP1_TargetNodeErr[255]	CIP 통신 1 대상 노드 이상 정보	CIP 통신 1의 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나 (등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되어 있으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었지만(등록 대상 노드 정보가 TRUE이며, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다 (대상 PLC 이상 정보가 TRUE). (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-173
_EIP2_TargetNodeErr[255]	CIP 통신 2 대상 노드 이상 정보	CIP 통신 2의 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나 (등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되어 있으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었지만(등록 대상 노드 정보가 TRUE이며, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다 (대상 PLC 이상 정보가 TRUE). (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	ARRAY [0..255] OF BOOL	TRUE, FALSE	A-173
_EIP_NTPResult	NTP의 동작 정보	사용자 프로그램에서 NTP의 동작 정보를 참조하는 경우에는 GetNTPStatus 명령을 사용하십시오. 직접 액세스할 수는 없습니다.	_sNTP_RESULT		A-174

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
.ExecTime	NTP의 최종 동작 시각	NTP 정상 종료 시각을 등록합니다. NTP 서버에서 정상적으로 시각을 취득한 경우에 NTP 서버에서 취득한 시각을 등록합니다. NTP 서버에서 정상적으로 시각을 취득할 수 없는 경우에는 등록하지 않습니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다.	DATE_AND_TIME	데이터 형식에 따름	A-174
.ExecNormal	NTP의 동작 결과	TRUE: NTP 정상 종료를 나타냅니다. FALSE: NTP 이상 종료 또는 한번도 NTP를 실행하지 않았음을 나타냅니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다.	BOOL	TRUE, FALSE	A-174

참고

대상 노드와의 통신 상태

NJ/NX 시리즈 컨트롤러와 대상 노드와의 통신 상태는 다음 4개의 시스템 정의 변수 값의 조합으로 다음 표와 같이 결정됩니다.

- _EIP_RegTargetSta(등록 대상 노드 정보)
- _EIP_EstbTargetSta(정상 대상 노드 정보)
- _EIP_TargetPLCErr(대상 PLC 이상 정보)
- _EIP_TargetNodeErr(대상 노드 이상 정보)

_EIP_RegTargetSta의 값	_EIP_EstbTargetSta의 값	_EIP_TargetPLCErr의 값	_EIP_TargetNodeErr의 값	대상 노드와의 통신 상태
TRUE	TRUE	FALSE	FALSE	대상 노드와의 연결이 정상적으로 확립되었으며, 대상 PLC에 이상이 발생하지 않았습니다.
		TRUE	TRUE	대상 노드와의 연결은 확립되었으나, 대상 PLC에 이상이 발생했습니다.
	FALSE	−	TRUE	대상 노드와의 연결이 정상적으로 확립되어 있지 않습니다.
FALSE	−	−	−	대상 노드가 등록되지 않았으므로, 정보는 무효입니다.

NX 시리즈 컨트롤러의 경우에 CIP 통신 1과 CIP 통신 2의 통신 상태는 각각 다음 4개의 시스템 정의 변수의 조합으로 위 표와 동일하게 결정됩니다.

- CIP 통신 1의 경우
  - _EIP1_RegTargetSta(CIP 통신 1 등록 대상 노드 정보)
  - _EIP1_EstbTargetSta(CIP 통신 1 정상 대상 노드 정보)
  - _EIP1_TargetPLCErr(CIP 통신 1 대상 PLC 이상 정보)
  - _EIP_TargetNodeErr(대상 노드 이상 정보)
- CIP 통신 2의 경우
  - _EIP2_RegTargetSta(CIP 통신 2 등록 대상 노드 정보)
  - _EIP2_EstbTargetSta(CIP 통신 2 정상 대상 노드 정보)
  - _EIP2_TargetPLCErr(CIP 통신 2 대상 PLC 이상 정보)
  - _EIP_TargetNodeErr(대상 노드 이상 정보)

● 기능 분류: EtherNet/IP 통신의 스위치 관련

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP_TDLinkStartCmd	태그 데이터 링크 통신 시작 스위치	NX 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. NJ 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오.	BOOL	TRUE, FALSE	A-174
_EIP1_TDLinkStartCmd	CIP 통신 1 태그 데이터 링크 통신 시작 스위치	FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-174
_EIP2_TDLinkStartCmd	CIP 통신 2 태그 데이터 링크 통신 시작 스위치	FALSE에서 TRUE가 되면, CIP 통신 2의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-175
_EIP_TDLinkStopCmd	태그 데이터 링크 통신 정지 스위치	NX 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. NJ 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오.	BOOL	TRUE, FALSE	A-175
_EIP1_TDLinkStopCmd	CIP 통신 1 태그 데이터 링크 통신 정지 스위치	FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-175

변수명	이름	기능	데이터 형식	값의 범위	참조 페이지
_EIP2_TDLinkStopCmd	CIP 통신 2 태그 데이터 링크 통신 정지 스위치	FALSE에서 TRUE가 되면, CIP 통신 2의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.	BOOL	TRUE, FALSE	A-175

A-6-8 이상 Status 각 비트의 의미

다음의 이상 Status 각 비트의 의미는 공통입니다.

- '_ErrSta'(컨트롤러 이상 Status)
- '_PLC_ErrSta'(PLC 기능 모듈 이상 Status)
- '_CJB_ErrSta'(I/O 버스 이상 Status)
- '_CJB_MstrErrSta'(I/O 버스 마스터 이상 Status)
- '_CJB_UnitErrSta'(I/O 버스 유니트 이상 Status)
- '_NXB_ErrSta'(NX 버스 기능 모듈 이상 Status)
- '_NXB_MstrErrSta'(NX 버스 기능 모듈 마스터 이상 Status)
- '_NXB_UnitErrStaTbl'(NX 버스 기능 모듈 유니트 이상 Status)
- '_MC_ErrSta'(MC 이상 Status)
- '_MC_ComErrSta'(MC 공통 이상 Status)
- '_MC_AX_ErrSta'(축 이상 Status)
- '_MC_GRP_ErrSta'(축 그룹 이상 Status)
- '_EC_ErrSta'(내장 EtherCAT 이상)
- '_EC_PortErr'(통신 포트 이상)
- '_EC_MstrErr'(마스터 이상)
- '_EC_SlavErr'(슬레이브 이상)
- '_EC_SlavErrTbl'(슬레이브 이상 테이블)
- '_EIP_ErrSta'(내장 EtherNet/IP 이상)
- '_EIP_PortErr'(통신 포트 이상), '_EIP1_PortErr'(통신 포트 1 이상), '_EIP2_PortErr'(통신 포트 2 이상)
- '_EIP_CipErr'(CIP 통신 이상), '_EIP1_CipErr'(CIP 통신 1 이상), '_EIP2_CipErr'(CIP 통신 2 이상)
- '_EIP_TcpAppErr'(TCP 어플리케이션 통신 이상),

다음과 같습니다. 단, '_ErrSta'(컨트롤러 이상 Status), '_CJB_ErrSta'(I/O 버스 이상 Status), '_CJB_MstrErrSta'(I/O 버스 마스터 이상 Status), '_CJB_UnitErrSta'(I/O 버스 유니트 이상 Status), '_NXB_ErrSta'(NX 버스 기능 모듈 이상 Status), '_NXB_MstrErrSta'(NX 버스 기능 모듈 마스터 이상 Status), '_NXB_UnitErrStaTbl'(NX 버스 기능 모듈 유니트 이상 Status)는 실시간성, 그리고 각 기능 모듈의 이상 요인을 나타내는 Status와의 동시성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다.

비트:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD			-	-	-	-	-	-					-	-	-	-

비트	내용
15	마스터 감지: 마스터가 컨트롤러 이상의 이상 Status 대상인 유니트/슬레이브에 대해 컨트롤러 이상을 감지했는지 여부를 나타냅니다. TRUE: 마스터가 감지한 컨트롤러 이상 있음 FALSE: 마스터가 감지한 컨트롤러 이상 없음 (_CJB_UnitErrSta에서 유효)
14	슬레이브 집약: 이벤트 발생원(각 기능 모듈)의 하위 계층(유니트/슬레이브/축/축 그룹 등)에 대해 컨트롤러 이상이 발생했는지 여부를 나타냅니다. TRUE: 어느 하위 계층에서 컨트롤러 이상 있음 FALSE: 하위 계층에 컨트롤러 이상 없음 (_CJB_ErrSta/_MC_ErrSta/_EC_ErrSta에서 유효)
13~8	예약
7	전체 정지 이상(Major fault) 레벨의 컨트롤러 이상이 발생 중인지 여부를 나타냅니다. TRUE: 전체 정지 이상 레벨의 컨트롤러 이상이 발생 중 FALSE: 전체 정지 이상 레벨의 컨트롤러 이상 발생 중이 아님
6	부분 정지 이상(Partial fault) 레벨의 컨트롤러 이상이 발생 중인지 여부를 나타냅니다. TRUE: 부분 정지 이상 레벨의 컨트롤러 이상이 발생 중 FALSE: 부분 정지 이상 레벨의 컨트롤러 이상 발생 중이 아님

비트	내용
5	가벼운 이상(Minor fault) 레벨의 컨트롤러 이상이 발생 중인지 여부를 나타냅니다. TRUE: 가벼운 이상 레벨의 컨트롤러 이상이 발생 중 FALSE: 가벼운 이상 레벨의 컨트롤러 이상 발생 중이 아님
4	감시 정보(Observation) 레벨의 컨트롤러 이상이 발생 중인지 여부를 나타냅니다. TRUE: 감시 정보 레벨의 컨트롤러 이상이 발생 중 FALSE: 감시 정보 레벨의 컨트롤러 이상 발생 중이 아님
3~0	예약

A-7 시스템 정의 변수의 개별 사양

시스템 정의 변수의 개별 사양표를 보는 방법은 다음과 같습니다.

<table>
<tr><td>변수명</td><td colspan="2">[시스템 정의 변수명입니다. 앞에 카테고리명이 붙습니다.]</td><td>멤버(구조체 형식의 경우)</td><td>[구조체 형식의 경우에만 멤버명을 표시합니다.]</td></tr>
<tr><td>이름</td><td colspan="2">[이름입니다.]</td><td>글로벌/로컬</td><td>'글로벌: 글로벌 변수, 로컬: 로컬 변수'</td></tr>
<tr><td>기능</td><td colspan="4">[기능을 설명합니다.]</td></tr>
<tr><td>데이터 형식</td><td colspan="2">[변수의 데이터 형식을 나타냅니다.]</td><td>값의 범위</td><td>[변수의 값의 범위를 나타냅니다.]</td></tr>
<tr><td>R/W 액세스</td><td>[R: 읽기 전용
RW: 읽기/쓰기 가능]</td><td>유지
[변수의 유지 속성을 나타냅니다.]</td><td>네트워크 공개</td><td>[변수의 네트워크 공개 속성을 나타냅니다.]</td></tr>
<tr><td>사용자 프로그램에서의 사용 가능 여부</td><td>[사용자 프로그램 상에서 직접 지정하여 사용할 수 있는지 여부를 나타냅니다.]</td><td>관련 명령</td><td colspan="2">[변수와 관련된 명령을 나타냅니다.
'사용자 프로그램에서 사용'이 불가능일 경우에는 액세스하기 위한 전용 명령을 나타냅니다.]</td></tr>
</table>

A-7-1 NJ/NX 시리즈 컨트롤러의 시스템 전체(카테고리명: 없음)

● 기능 분류: 시계 관련

변수명	_CurrentTime				
이름	시스템 시각			글로벌/로컬	글로벌
기능	CPU 유니트의 내부 시계 데이터가 저장됩니다.				
데이터 형식	DATE_AND_TIME			값의 범위	• NX 시리즈 CPU 유니트의 경우 DT#1970-01-01-00:00:00~ DT#2069-12-31-23:59:59 • NJ 시리즈 CPU 유니트의 경우 DT#1970-01-01-00:00:00~ DT#2106-02-06-23:59:59
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	시계 기능용 명령		

● 기능 분류: 태스크 관련

변수명	_<태스크 이름>_Active				
이름	태스크 실행 중 플래그			글로벌/로컬	글로벌
기능	태스크 실행 중일 때 TRUE가 됩니다. 태스크 실행 중이 아닐 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	• ActEventTask 사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • Task_IsActive		

변수명	_<태스크 이름>_LastExecTime				
이름	이전 태스크 실행 시간			글로벌/로컬	글로벌
기능	이전에 실행한 태스크 실행 시간(0.1μs 단위)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	TIME			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

변수명	_<태스크 이름>_MaxExecTime				
이름	태스크 실행 시간 최대값			글로벌/로컬	글로벌
기능	태스크 실행 시간의 최대값(0.1μs)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	TIME			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

변수명	_<태스크 이름>_MinExecTime				
이름	태스크 실행 시간 최소값			글로벌/로컬	글로벌
기능	태스크 실행 시간의 최소값(0.1μs)을 표시합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	TIME			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

변수명	_<태스크 이름>_ExecCount				
이름	태스크 실행 횟수			글로벌/로컬	글로벌
기능	태스크의 실행 횟수를 표시합니다. 4294967295회를 초과하면 0으로 돌아가고, 카운트를 계속합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	UDINT			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

변수명	_<태스크 이름>_Exceeded				
이름	태스크 주기 초과 플래그			글로벌/로컬	글로벌
기능	태스크 주기를 초과하면 TRUE가 됩니다. 태스크 주기 이내에 태스크 실행이 완료되면 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

변수명	_<태스크 이름>_ExceedCount				
이름	태스크 주기 초과 횟수			글로벌/로컬	글로벌
기능	주기를 초과한 횟수가 저장됩니다. 현재값이 데이터 형식의 최대값을 초과하면 현재값을 0으로 되돌려서 카운트를 계속합니다. 4294967295회를 초과하면 0으로 돌아가고, 카운트를 계속합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio의 데이터 트레이스 기능으로 태스크의 상태를 참조할 때에만 사용합니다.				
데이터 형식	UDINT			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetMyTaskStatus		

● 기능 분류: 이상 관련

변수명	_ErrSta				
이름	컨트롤러 이상 Status			글로벌/로컬	글로벌
기능	컨트롤러 이상 발생 중에는 TRUE가 됩니다. 컨트롤러 이상 미발생 중에는 FALSE가 됩니다. (주의) 실시간성과 각 기능 모듈의 이상 요인을 나타내는 Status와의 동시성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#C0F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	• ResetPLCError • ResetCJBError • ResetECError • ResetMCError • MC_Reset • MC_GroupReset 사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetPLCError • GetCJBError • GetECError • GetMCError • GetEIPError		

변수명	_AlarmFlag				
이름	사용자 이상 Status			글로벌/로컬	글로벌
기능	사용자 이상 발생 중에는 이벤트 중요도에 해당하는 비트가 TRUE가 됩니다. User fault Level 1~8이 비트 00~07에 대응합니다. 사용자 이상 미발생 중에는 16진수 0000이 됩니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00FF
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• SetAlarm • ResetAlarm • GetAlarm		

● 기능 분류: SD 메모리 카드 관련

변수명	_Card1Ready				
이름	SD 메모리 카드 사용 가능 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드가 인식되었을 때 TRUE가 됩니다. 인식하지 못했을 때 FALSE가 됩니다. TRUE: 사용 가능 FALSE: 사용 불가능				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_Card1Protect				
이름	SD 메모리 카드 쓰기 방지 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드에 LOCK 스위치에 의한 쓰기 방지가 걸려있을 때 TRUE가 됩니다. TRUE: 쓰기 방지 있음 FALSE: 쓰기 방지 없음				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_Card1Err				
이름	SD 메모리 카드 에러 플래그			글로벌/로컬	글로벌
기능	사용이 불가능한 SD 메모리 카드가 장착됐을 때 또는 포맷에 이상이 발생했을 때 TRUE가 됩니다. TRUE: 에러 있음 FALSE: 에러 없음				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_Card1Access				
이름	SD 메모리 카드 액세스 중 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드에 액세스 중에 TRUE가 됩니다. TRUE: 액세스 중 FALSE: 액세스 중이 아님 이 플래그는 100ms마다 시스템이 갱신합니다. 이로 인해 최대 100ms, SD 메모리 카드에 대한 액세스 상태인 이 플래그로의 반영이 지연됩니다. 따라서 사용자 프로그램 상에서 사용하는 것은 권장하지 않습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_Card1Deteriorated				
이름	SD 메모리 카드 수명 경고 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드의 수명을 감지했을 때 TRUE가 됩니다. 이 플래그가 TRUE가 되면 SD 메모리 카드를 교환하십시오. 교환하지 않으면 읽기/쓰기에 실패할 가능성이 있습니다. TRUE: 수명 감지 있음 FALSE: 수명 감지 없음				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_Card1PowerFail				
이름	SD 메모리 카드 액세스 중 전원 OFF 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드 액세스 중에 CPU 유니트의 전원 OFF가 발생했을 때 TRUE가 됩니다. TRUE: SD 메모리 카드 액세스 중에 전원 OFF가 있었음 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*1	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1BkupCmd*1			멤버명	.ExecBkup
이름	백업 기동 플래그			글로벌/로컬	글로벌
기능	컨트롤러의 데이터를 SD 메모리 카드로 백업할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.				
데이터 형식	구조체: _sBKUP_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupCmd*1			멤버명	.CancelBkup
이름	백업 취소 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드로의 백업을 중지할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.				
데이터 형식	구조체: _sBKUP_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupCmd*1			멤버명	.ExecVefy
이름	비교 기동 플래그			글로벌/로컬	글로벌
기능	컨트롤러와 SD 메모리 카드 내의 백업 파일을 비교할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.				
데이터 형식	구조체: _sBKUP_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupCmd*1			멤버명	.CancelVefy
이름	비교 취소 플래그			글로벌/로컬	글로벌
기능	컨트롤러와 SD 메모리 카드 내의 백업 파일 비교를 중지할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.				
데이터 형식	구조체: _sBKUP_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupCmd*1			멤버명	.DirName
이름	디렉토리명			글로벌/로컬	글로벌
기능	백업 대상인 SD 메모리 카드의 디렉토리명을 지정합니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 명령을 내릴 때 사용합니다.				
데이터 형식	구조체: _sBKUP_CMD, 멤버: STRING(64)			값의 범위	데이터 형식에 따름
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupSta*1			멤버명	.Done
이름	완료 플래그			글로벌/로컬	글로벌
기능	백업 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sBKUP_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupSta*1			멤버명	.Active
이름	실행 중 플래그			글로벌/로컬	글로벌
기능	백업 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sBKUP_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1BkupSta*1			멤버명	.Err
이름	에러 플래그			글로벌/로컬	글로벌
기능	백업 처리가 이상 종료했을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sBKUP_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1VefySta*1			멤버명	.Done
이름	완료 플래그			글로벌/로컬	글로벌
기능	비교 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sVEFY_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1VefySta*1			멤버명	.Active
이름	실행 중 플래그			글로벌/로컬	글로벌
기능	비교 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sVEFY_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1VefySta*1			멤버명	.VefyRslt
이름	비교 결과 플래그			글로벌/로컬	글로벌
기능	비교 일치일 때 TRUE, 비교 불일치일 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sVEFY_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1VefySta*1			멤버명	.Err
이름	에러 플래그			글로벌/로컬	글로벌
기능	비교 처리가 이상 종료되었을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. CIP 메시지 통신으로 표시기 또는 상위 PC에서 읽을 때 사용합니다.				
데이터 형식	구조체: _sVEFY_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.Exec
이름	프로그램 전송 기동 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드 프로그램 전송 기능에 의해 SD 메모리 카드의 백업 파일 데이터를 컨트롤러로 전송할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.DirName
이름	디렉토리명			글로벌/로컬	글로벌
기능	프로그램 전송 대상인 백업 파일이 저장된 SD 메모리 카드의 디렉토리명을 지정합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: STRING(64)			값의 범위	데이터 형식에 따름
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.Password
이름	비밀번호			글로벌/로컬	글로벌
기능	프로그램 전송 시작 시에 비교되는 비밀번호를 지정합니다. SD 메모리 카드 프로그램 전송을 시작할 때마다 비밀번호가 초기화됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: STRING(33)			값의 범위	데이터 형식에 따름
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.TargetUserProgram
이름	사용자 프로그램·설정 전송 플래그			글로벌/로컬	글로벌
기능	프로그램 전송에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.TargetIPAdr
이름	IP 주소 전송 플래그			글로벌/로컬	글로벌
기능	프로그램 전송에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 종류, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.TargetVariable
이름	변수의 현재값(유지 속성만 해당) 전송 플래그			글로벌/로컬	글로벌
기능	프로그램 전송에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferCmd*1			멤버명	.TargetMemory
이름	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그			글로벌/로컬	글로벌
기능	프로그램 전송에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1PrgTransferSta*1			멤버명	.Done
이름	완료 플래그			글로벌/로컬	글로벌
기능	프로그램 전송 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_Card1PrgTransferSta*1			멤버명	.Active
이름	실행 중 플래그			글로벌/로컬	글로벌
기능	프로그램 전송 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_Card1PrgTransferSta*1			멤버명	.Err
이름	에러 플래그			글로벌/로컬	글로벌
기능	프로그램 전송이 이상 종료됐을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sPRGTRANSFER_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_Card1RestoreCmd*1			멤버명	.Exec
이름	복원 기동 플래그			글로벌/로컬	글로벌
기능	SD 메모리 카드 내의 백업 파일의 데이터를 컨트롤러로 복원할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_CMD, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmd*1			멤버명	.DirName
이름	디렉토리명			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원 대상인 백업 파일이 저장된 SD 메모리 카드의 디렉토리명을 지정합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_CMD, 멤버: STRING(64)			값의 범위	데이터 형식에 따름
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmd*1			멤버명	.Password
이름	비밀번호			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원 시작 시에 비교될 비밀번호를 지정합니다. 시스템 정의 변수에 의한 복원을 시작할 때마다 비밀번호가 초기화됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_CMD, 멤버: STRING(33)			값의 범위	데이터 형식에 따름
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.

변수명	_Card1RestoreSta*1			멤버명	.Done
이름	완료 플래그			글로벌/로컬	글로벌
기능	복원 실행 완료일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.

변수명	_Card1RestoreSta*1			멤버명	.Active
이름	실행 중 플래그			글로벌/로컬	글로벌
기능	복원 실행 중일 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.

변수명	_Card1RestoreSta*1			멤버명	.Err
이름	에러 플래그			글로벌/로컬	글로벌
기능	복원이 이상 종료됐을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	구조체: _sRESTORE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.

변수명	_Card1RestoreCmdTargetUserProgram*1				
이름	사용자 프로그램·설정 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 사용자 프로그램이나 설정을 전송 대상으로 할 때 TRUE로 합니다. 시스템 정의 변수에 의한 복원일 때는 반드시 TRUE로 하십시오. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmdTargetIPAdr*1				
이름	IP 주소 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 내장 EtherNet/IP 포트의 IP 주소를 전송 대상으로 할 때 TRUE로 합니다. IP 주소에는 설정 종류, IP 주소, 서브넷 마스크, 기본 게이트웨이가 포함됩니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmdTargetVariable*1				
이름	변수의 현재값(유지 속성만 해당) 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 변수의 현재값(유지 속성만 해당)을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmdTargetMemory*1				
이름	CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 CJ 유니트용 메모리의 현재값(유지 메모리만 해당)을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmdTargetUnitConfig*1				
이름	유니트·슬레이브 설정 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 유니트·슬레이브 설정을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_Card1RestoreCmdTargetAbsEncoder*1				
이름	앱솔루트 엔코더 원점 위치 오프셋 전송 플래그			글로벌/로컬	글로벌
기능	시스템 정의 변수에 의한 복원에서 앱솔루트 엔코더 원점 위치 오프셋을 전송 대상으로 할 때 TRUE로 합니다. (주의) 이 시스템 정의 변수는 NJ/NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	유지*2	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.14에서 추가된 시스템 정의 변수입니다.
*2. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

● 기능 분류: 백업 관련

변수명	_BackupBusy*1				
이름	백업 관련 기능 실행 중 플래그			글로벌/로컬	글로벌
기능	백업, 복원, 비교 실행 중일 때 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 시스템 정의 변수입니다.

● 기능 분류: 전원 관리 관련

변수명	_PowerOnHour				
이름	통전 시간			글로벌/로컬	글로벌
기능	통전 시간을 표시합니다. CPU 유니트의 통전 시간을 1시간 단위로 저장합니다. 리셋하려면 0을 저장하십시오. 4294967295가 되면, 그 이상은 갱신되지 않습니다. 전원 ON 시에 초기화되지 않습니다.				
데이터 형식	UDINT			값의 범위	0~4294967295
R/W 액세스	RW	유지	유지*1	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_PowerOnCount				
이름	전원 OFF 발생 횟수			글로벌/로컬	글로벌
기능	전원 OFF 발생 횟수를 표시합니다. CPU 유니트의 최초 전원 투입부터 전원 OFF 발생마다 누적(+1)됩니다. 리셋하려면 0을 저장하십시오. 4294967295가 되면, 그 이상은 갱신되지 않습니다. 전원 ON 시에 초기화되지 않습니다.				
데이터 형식	UDINT			값의 범위	0~4294967295
R/W 액세스	RW	유지	유지*1	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. 유지 속성이 있는 시스템 정의 변수이지만, 백업 기능의 대상이 아닙니다.

변수명	_RetainFail				
이름	전원 OFF 유지 실패 플래그			글로벌/로컬	글로벌
기능	다음의 경우에 TRUE(전원 OFF 유지 실패)가 됩니다. • 전원 투입 시, 배터리 백업 메모리의 확인에서 에러일 때 다음의 경우에 FALSE(전원 OFF 유지 실패 없음)가 됩니다. • 전원 투입 시, 배터리 백업 메모리의 메모리 확인에서 OK일 때 • 사용자 프로그램이 다운로드 되었을 때 • 메모리 전체 클리어를 실행했을 때 (주의) 앱솔루트 엔코더 원점 위치 오프셋 데이터를 유지하지 못했을 때는 이 플래그가 아니라 축 변수의 이상 Status에 반영됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: 프로그래밍 관련

변수명	P_On				
이름	상시 ON 플래그			글로벌/로컬	글로벌
기능	항상 TRUE를 나타내는 플래그입니다.				
데이터 형식	BOOL			값의 범위	TRUE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	P_Off				
이름	상시 OFF 플래그			글로벌/로컬	글로벌
기능	항상 FALSE를 나타내는 플래그입니다.				
데이터 형식	BOOL			값의 범위	FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	P_CY				
이름	캐리 플래그			글로벌/로컬	로컬
기능	일부 명령으로 갱신하는 플래그입니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	P_First_RunMode				
이름	운전 시작 시 1주기 ON 플래그			글로벌/로컬	로컬
기능	컨트롤러의 동작 모드를 프로그램 모드에서 운전 모드로 변경했을 때 해당 프로그램이 실행 중인 경우에 1태스크 주기만 TRUE가 됩니다. 해당 프로그램이 실행 중이 아니면 FALSE로 유지됩니다. CPU 유니트가 운전을 시작했을 때 초기 처리를 수행하는 경우 등에 사용합니다. (주의) 이 시스템 정의 변수는 펑션 내에서는 사용할 수 없습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	P_First_Run*1				
이름	프로그램 기동 시 1주기 ON 플래그			글로벌/로컬	로컬
기능	해당 프로그램이 실행을 시작했을 때 1태스크 주기만 TRUE가 됩니다. 프로그램이 실행을 시작할 때마다 초기 처리를 수행하는 경우 등에 사용합니다. (주의) 이 시스템 정의 변수는 펑션 내에서는 사용할 수 없습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.08에서 추가된 시스템 정의 변수입니다.

변수명	P_PRGER				
이름	명령 에러 발생 플래그			글로벌/로컬	로컬
기능	해당 프로그램 또는 해당 프로그램에서 호출한 FB나 FUN에서 명령 에러가 발생했을 때 계속 TRUE가 되는 플래그입니다. 한번 TRUE가 되면 사용자 프로그램에 의해 FALSE가 될 때까지 TRUE로 유지됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: 통신 관련

변수명	_Port_numUsingPort				
이름	사용 포트 수			글로벌/로컬	글로벌
기능	현재 사용 중인 내부 논리 포트 수를 표시합니다. 사용자 프로그램의 디버그 시 등에 사용합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	USINT			값의 범위	0~32
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	통신계 명령 (ExecPMCR, SerialSend, SerialRcv, 통신 명령, SendCmd)		

변수명	_Port_isAvailable				
이름	네트워크 통신 명령 실행 가능 플래그			글로벌/로컬	글로벌
기능	사용 가능한 내부 논리 포트의 유무를 표시합니다. 사용 가능한 내부 논리 포트가 있을 때 TRUE, 없을 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	통신계 명령 (ExecPMCR, SerialSend, SerialRcv, 통신 명령, SendCmd)		

변수명	_FINSTCPConnSta				
이름	FINS/TCP 연결 Status			글로벌/로컬	글로벌
기능	FINS/TCP 연결 Status를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#FFFF
R/W 액세스	R	유지	비유지	네트워크 공개	공개*1
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.07 이하에서는 네트워크 비공개입니다.

● 기능 분류: 버전 관리 관련

변수명	_UnitVersion*1				
이름	유니트 버전			글로벌/로컬	글로벌
기능	CPU 유니트의 유니트 버전이 저장됩니다. 요소 번호 0에 유니트 버전의 정수 부분이 저장됩니다. 요소 번호 1에 유니트 버전의 소수 부분이 저장됩니다. 예1) 유니트 버전 Ver.1.08의 경우에 요소 번호 0에 1, 요소 번호 1에 8이 저장됩니다. 예2) 유니트 버전 Ver.1.10의 경우에 요소 번호 0에 1, 요소 번호 1에 10이 저장됩니다.				
데이터 형식	ARRAY[0..1] OF USINT			값의 범위	0~99
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.08에서 추가된 시스템 정의 변수입니다.

변수명	_HardwareRevision*1				
이름	하드웨어 개정			글로벌/로컬	글로벌
기능	CPU 유니트의 하드웨어 개정이 저장됩니다. 하드웨어 개정이 '없음'인 경우에는 '-', 그 이외의 경우는 'A'~'Z'가 저장됩니다.				
데이터 형식	STRING[2]			값의 범위	'-', 'A'~'Z'
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

● 기능 분류: 자체 진단 관련

변수명	_SelfTest_HighTemperature*1				
이름	CPU 유니트 내부 온도 초과 플래그			글로벌/로컬	글로벌
기능	CPU 유니트 내부 온도가 고온일 때 TRUE가 됩니다. (주의) NX102 CPU 유니트 및 NX1P2 CPU 유니트의 경우에 항상 FALSE입니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.10에서 추가된 시스템 정의 변수입니다.

변수명	_SelfTest_LowBattery*1				
이름	배터리 전압 저하 플래그			글로벌/로컬	글로벌
기능	배터리가 분리되어 있거나, 배터리 전압이 저하되어 있을 때 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.10에서 추가된 시스템 정의 변수입니다.

변수명	_SelfTest_LowFanRevolution*1				
이름	FAN 회전 수 저하 플래그			글로벌/로컬	글로벌
기능	FAN이 분리되어 있거나, FAN 회전 수가 저하되어 있을 때 TRUE가 됩니다. (주의) NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 항상 FALSE입니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.10에서 추가된 시스템 정의 변수입니다.

● 기능 분류: PLC 내장 관련

변수명	_DeviceOutHoldCfg*1				
이름	장치 출력 유지 설정			글로벌/로컬	글로벌
기능	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력을 유지하는 경우에 16#A5A5로 합니다. 16#A5A5 이외에서는 동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력은 초기화됩니다.				
데이터 형식	WORD			값의 범위	16#0000~16#FFFF
R/W 액세스	RW	유지	유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_DeviceOutHoldStatus*1				
이름	장치 출력 유지 상태			글로벌/로컬	글로벌
기능	동작 모드 변경 시 또는 다운로드 시에 대상 장치 출력이 유지된 경우에 TRUE가 됩니다. 장치 출력 유지 설정이 16#A5A5 이외일 때 또는 전체 정지 이상 레벨의 컨트롤러 이상 발생 시에 대상 장치 출력이 초기화된 경우에 FALSE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

A-7-2 PLC 기능 모듈(카테고리명: _PLC)

● 기능 분류: 디버그 관련

변수명	_PLC_TraceSta[0..3]			멤버명	.IsStart
이름	트레이스 실행 중 ON 플래그			글로벌/로컬	글로벌
기능	트레이스를 시작했을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.				
데이터 형식	구조체: _sTRACE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	• TraceTrig • TraceSamp 사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetTraceStatus		

변수명	_PLC_TraceSta[0..3]			멤버명	.IsComplete
이름	트레이스 완료 시 ON 플래그			글로벌/로컬	글로벌
기능	트레이스 실행이 완료됐을 때 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.				
데이터 형식	구조체: _sTRACE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	• TraceTrig • TraceSamp 사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetTraceStatus		

변수명	_PLC_TraceSta[0..3]			멤버명	.IsTrigger
이름	트레이스 트리거 모니터링 플래그			글로벌/로컬	글로벌
기능	트레이스 트리거 조건이 성립됐을 때 TRUE가 됩니다. 다음으로 트레이스를 시작했을 때, FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.				
데이터 형식	구조체: _sTRACE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	• TraceTrig • TraceSamp 사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetTraceStatus		

변수명	_PLC_TraceSta[0..3]			멤버명	.ParamErr
이름	트레이스 파라메터 이상 플래그			글로벌/로컬	글로벌
기능	트레이스 시작 시에 트레이스 설정이 이상일 때 TRUE가 됩니다. 정상일 때 FALSE가 됩니다. (주의) 이 시스템 정의 변수는 사용자 프로그램 상에서는 사용할 수 없습니다. Sysmac Studio에서 데이터 트레이스 상태를 감시할 때에만 사용합니다.				
데이터 형식	구조체: _sTRACE_STA, 멤버: BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetTraceStatus		

● 기능 분류: 이상 관련

변수명	_PLC_ErrSta				
이름	PLC 기능 모듈 이상 Status			글로벌/로컬	글로벌
기능	PLC 기능 모듈과 관련된 컨트롤러 이상 발생 중에는 TRUE가 됩니다. PLC 기능 모듈과 관련된 컨트롤러 이상이 발생 중이지 않을 때는 FALSE가 됩니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• GetPLCError 다음 전용 명령으로 해제 가능 • ResetPLCError		

● 기능 분류: 옵션 보드 관련

변수명	_PLC_OptBoardSta*1				
이름	옵션 보드 Status			글로벌/로컬	글로벌
기능	옵션 보드 관련 Status를 저장합니다. 옵션 보드 형식에 관계 없이 공통으로 사용합니다. 배열 요소 1이 옵션 보드 슬롯 1, 배열 요소 2가 옵션 보드 슬롯 2에 대응합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY[1..2] OF _sOPTBOARD_STA			값의 범위	
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptBoardSta*1			멤버명	IsDetect
이름	옵션 보드 장착			글로벌/로컬	글로벌
기능	옵션 보드 슬롯에 옵션 보드가 장착되어 있음을 표시합니다. TRUE: 장착함 FALSE: 장착하지 않음 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptBoardSta*1			멤버명	Run
이름	옵션 보드 정상 동작			글로벌/로컬	글로벌
기능	옵션 보드가 정상적으로 동작 중인지 여부를 표시합니다. 옵션 보드의 디바이스 변수를 사용할 때나 옵션 보드용 통신 명령을 사용할 때는 이 멤버를 인터록 조건으로 합니다. TRUE: 정상적으로 동작 중 FALSE: 초기 처리 중 또는 설정 변경 중 또는 에러 발생 중 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptBoardSta*1			멤버명	Error
이름	옵션 보드 에러 발생			글로벌/로컬	글로벌
기능	옵션 보드의 에러가 발생 여부를 표시합니다. TRUE: 에러가 발생함 FALSE: 에러가 발생하지 않음 (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptSerialErrSta*1				
이름	시리얼 옵션 보드 에러 Status			글로벌/로컬	글로벌
기능	시리얼 통신용 옵션 보드의 전송 에러의 에러 Status를 저장합니다. 시리얼 통신용 옵션 보드의 시리얼 통신 모드를 상위 링크(FINS)로 한 경우에만 각 멤버의 값을 갱신합니다. 위 이외의 경우에는 모든 멤버의 값을 FALSE로 합니다. 배열 요소 1이 옵션 보드 슬롯 1, 배열 요소 2가 옵션 보드 슬롯 2에 대응합니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY[1..2] OF _sOPTSERIALERR_STA			값의 범위	
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptSerialErrSta*1			멤버명	Error
이름	전송 에러 발생			글로벌/로컬	글로벌
기능	전송 에러 발생 유무를 표시합니다. TRUE: 패리티 에러 또는 프레이밍 에러 또는 오버런 에러가 발생 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 패리티 에러 또는 프레이밍 에러 또는 오버런 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptSerialErrSta*1			멤버명	ParityErr
이름	패리티 에러 발생			글로벌/로컬	글로벌
기능	패리티 에러 발생 유무를 나타냅니다. 이 에러가 발생했다면 시리얼 통신 설정이 연결 상대 기기와 맞지 않을 가능성이 있습니다. TRUE: 패리티 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 패리티 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptSerialErrSta*1			멤버명	FramingErr
이름	프레이밍 에러 발생			글로벌/로컬	글로벌
기능	프레이밍 에러 발생 유무를 표시합니다. 이 에러가 발생했다면 시리얼 통신 설정이 연결 상대 기기와 맞지 않을 가능성이 있습니다. TRUE: 프레이밍 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 프레이밍 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_PLC_OptSerialErrSta*1			멤버명	OverRun
이름	오버런 에러 발생			글로벌/로컬	글로벌
기능	오버런 에러 발생 유무를 표시합니다. 이 에러가 발생했다면 옵션 보드의 전송 속도가 너무 빠를 가능성이 있습니다. TRUE: 오버런 에러가 발생함 FALSE: 다음 중 어느 조건에서 FALSE가 됨 • 오버런 에러가 발생하지 않음 • 포트 재시작 • 시리얼 통신 모드가 상위 링크(FINS)가 아님 • 시리얼 통신용 이외의 옵션 보드를 장착 • 옵션 보드 장착 안 함 (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. (주의) 이 시스템 정의 변수는 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령			

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

A-7-3 PLC 기능 모듈(카테고리명: _CJB)

● 기능 분류: I/O 버스의 Status 관련

변수명	_CJB_MaxRackNo				
이름	최대 랙 번호			글로벌/로컬	글로벌
기능	컨트롤러가 인식하고 있는 증설 랙의 최대 랙 번호를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	UINT			값의 범위	0~3 0은 증설 랙 없음
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_CJB_MaxSlotNo				
이름	최대 슬롯 번호			글로벌/로컬	글로벌
기능	컨트롤러가 인식하고 있는 CJ 유니트의 랙마다 (최대 슬롯 번호+1)을 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..3] OF UINT			값의 범위	0~10 0의 경우에 CJ 유니트 장착 없음
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: I/O 버스의 이상 관련

변수명	_CJB_ErrSta				
이름	I/O 버스 이상 Status			글로벌/로컬	글로벌
기능	I/O 버스의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#C0F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetCJBError 다음 전용 명령으로 해제 가능 • ResetCJBError		

변수명	_CJB_MstrErrSta				
이름	I/O 버스 마스터 이상 Status			글로벌/로컬	글로벌
기능	I/O 버스 마스터의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetCJBError 다음 전용 명령으로 해제 가능 • ResetCJBError		

변수명	_CJB_UnitErrSta				
이름	I/O 버스 유니트 이상 Status			글로벌/로컬	글로벌
기능	I/O 버스 유니트의 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 📖 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..3, 0..9] OF WORD			값의 범위	16#0000~16#80F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetCJBError 다음 전용 명령으로 해제 가능 • ResetCJBError		

변수명	_CJB_InRespTm				
이름	기본 입력 유니트의 입력 시상수			글로벌/로컬	글로벌
기능	기본 입력 유니트의 입력 시상수를 표시합니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..3, 0..9] OF UINT			값의 범위	0~320
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: CJ 유니트용 메모리의 특수 보조 릴레이 관련

변수명		_CJB_IOUnitInfo				
이름		기본 I/O 유니트 정보 영역			글로벌/로컬	글로벌
기능		기본 I/O 유니트(부하 단락 방지 기능 장착)의 알람 출력을 표시합니다. TRUE: 부하 단락 있음 FALSE: 부하 단락 없음 (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		ARRAY [0..3, 0..9, 0..7] of BOOL			값의 범위	TRUE, FALSE
R/W 액세스		R	유지	비유지	네트워크 공개	비공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	–		
특수 보조 릴레이 주소	CH	A50~A69				
	비트	A50.00~A69.15				

변수명		_CJB_CBU00InitSta . . . _CJB_CBU15InitSta				
이름		CPU 고기능 유니트 초기화 중 플래그			글로벌/로컬	글로벌
기능		CPU 고기능 유니트가 초기화 중일 때 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 유니트 번호가 대응됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		BOOL			값의 범위	TRUE, FALSE
R/W 액세스		R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	• ResetUnit		
특수 보조 릴레이 주소	CH	A302				
	비트	A302.00~A302.15				

변수명		_CJB_SIO00InitSta · · · _CJB_SIO95InitSta				
이름		고기능 I/O 유니트 초기화 중 플래그			글로벌/로컬	글로벌
기능		고기능 I/O 유니트가 초기화 중에 TRUE가 됩니다. 초기화 완료 시에 FALSE가 됩니다. 각 비트와 호기 번호가 대응됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		BOOL			값의 범위	TRUE, FALSE
R/W 액세스		R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	• ResetUnit		
특수 보조 릴레이 주소	CH	A330~A335				
	비트	A330.00~A335.15				

변수명		_CJB_CBU00Restart · · · _CJB_CBU15Restart				
이름		CPU 고기능 유니트 재시작 플래그			글로벌/로컬	글로벌
기능		상승 시 CPU 고기능 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 유니트 번호에 대응합니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		BOOL			값의 범위	TRUE, FALSE
R/W 액세스		RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	• ResetUnit		
특수 보조 릴레이 주소	CH	A501				
	비트	A501.00~A501.15				

변수명		_CJB_SIO00Restart · · · _CJB_SIO95Restart				
이름		고기능 I/O 유니트 재시작 플래그			글로벌/로컬	글로벌
기능		상승 시 고기능 I/O 유니트가 재시작됩니다(재시작 처리 후, 시스템에서 FALSE가 됩니다). 각 비트가 호기 번호에 대응합니다. 명령 실행 시에 재시작 플래그를 TRUE로 한 경우에는 다음 태스크 주기의 리프레시 처리 시부터 재시작 처리가 시작됩니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		BOOL			값의 범위	TRUE, FALSE
R/W 액세스		RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	• ResetUnit		
특수 보조 릴레이 주소	CH	A502~A507				
	비트	A502.00~A507.15				

변수명		_CJB_SCU00P1ChgSta _CJB_SCU00P2ChgSta · · · _CJB_SCU15P1ChgSta _CJB_SCU15P2ChgSta				
이름		시리얼 커뮤니케이션 유니트 0호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그 · · · 시리얼 커뮤니케이션 유니트 1-15호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그			글로벌/로컬	글로벌
기능		해당 포트의 설정 변경 중에 TRUE가 됩니다. 설정 변경이 완료된 시점에서 FALSE가 됩니다. 명령 실행 또는 사용자 조작으로 이 비트를 TRUE로 하면 시리얼 포트 설정 변경을 통지할 수도 있습니다. (주의) 이 시스템 정의 변수는 NJ 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식		BOOL			값의 범위	TRUE, FALSE
R/W 액세스		RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부		가능	관련 명령	–		
특수 보조 릴레이 주소	CH	시리얼 커뮤니케이션 유니트 0호기 포트의 경우: A620 시리얼 커뮤니케이션 유니트 1-15호기 포트의 경우: A621~A635				
	비트	시리얼 커뮤니케이션 유니트 0호기 포트의 경우: A620.01~A620.02 시리얼 커뮤니케이션 유니트 1-15호기 포트의 경우: A621.01~A635.02				

A-7-4 NX 버스 기능 모듈(카테고리명: _NXB)

● 기능 분류: NX 버스 기능 모듈의 Status 관련

변수명	_NXB_MaxUnitNo*1				
이름	최대 NX 유니트 번호			글로벌/로컬	글로벌
기능	NX 버스 기능 모듈이 인식하고 있는 CPU 유니트 상의 NX 유니트의 NX 유니트 번호의 최대값을 표시합니다. Sysmac Studio에 의해 유니트 구성 정보가 등록된 경우에 등록한 유니트 구성의 NX 유니트 번호의 최대값이 됩니다. 미장착 유니트로 설정된 유니트도 대상입니다. Sysmac Studio에 의해 유니트 구성 정보가 등록되지 않은 경우에 실제 유니트 구성의 유니트 번호의 최대값이 됩니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	UINT			값의 범위	0~32 *2 0의 경우에 NX 유니트 장착 없음
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.
*2. NX1P2 CPU 유니트의 경우에 값의 범위는 0~8입니다.

변수명	_NXB_UnitIOActiveTbl*1				
이름	NX 유니트 I/O 데이터 통신 중 Status			글로벌/로컬	글로벌
기능	CPU 유니트 상의 NX 유니트의 I/O 데이터가 유효한지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, 항상 TRUE가 됩니다. TRUE: NX 유니트의 I/O 데이터가 유효 FALSE: NX 유니트의 I/O 데이터가 무효 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..8] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_NXB_UnitMsgActiveTbl*1				
이름	NX 유니트 메시지 통신 가능 Status			글로벌/로컬	글로벌
기능	CPU 유니트 상의 NX 유니트가 메시지 통신 가능한지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, 항상 TRUE가 됩니다. TRUE: 메시지 통신 가능 FALSE: 메시지 통신 불가능 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..8] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_NXB_UnitRegTbl*1				
이름	NX 유니트 등록 Status			글로벌/로컬	글로벌
기능	CPU 유니트 상의 NX 유니트가 유니트 구성 정보에 등록되어 있는지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미합니다. TRUE: 등록 있음 FALSE: 등록 없음 Sysmac Studio에 의해 유니트 구성 정보가 등록되지 않은 경우에 모든 유니트가 'FALSE'가 됩니다. 미장착 유니트로 설정된 NX 유니트는 'TRUE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..8] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

● 기능 분류: NX 버스 기능 모듈의 이상 관련

변수명	_NXB_ErrSta*1				
이름	NX 버스 기능 모듈 이상 Status			글로벌/로컬	글로벌
기능	NX 버스 기능 모듈의 이상 Status입니다. 'NX 버스 기능 모듈 마스터 이상 Status'와 모든 NX 유니트의 'NX 버스 기능 모듈 유니트 이상 Status'를 집약한 시스템 정의 변수입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#40F2
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetNXBError 다음 전용 명령으로 해제 가능 • ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_NXB_MstrErrSta*1				
이름	NX 버스 기능 모듈 마스터 이상 Status			글로벌/로컬	글로벌
기능	CPU 유니트의 NX 버스 기능 모듈에서 감지한 이상 Status입니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#40F2
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetNXBError 다음 전용 명령으로 해제 가능 • ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_NXB_UnitErrStaTbl*1				
이름	NX 버스 기능 모듈 유니트 이상 Status			글로벌/로컬	글로벌
기능	CPU 유니트 상의 NX 유니트에서 감지한 이상 Status입니다. 이 Status는 WORD 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서의 사용은 권장하지 않습니다. 통신을 통해 표시기 등 외부에서 상태를 참조할 때에만 사용합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [1..8] OF WORD			값의 범위	16#0000~16#40F2
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetNXBError 다음 전용 명령으로 해제 가능 • ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_NXB_UnitErrFlagTbl*1				
이름	NX 유니트 이상 Status			글로벌/로컬	글로벌
기능	CPU 유니트 상의 NX 유니트에서 이상이 발생했는지를 표시합니다. 이 Status는 BOOL 형식 데이터 배열입니다. 배열의 첨자는 NX 유니트 번호에 대응합니다. 첨자가 '0'인 경우는 NX 버스 기능 모듈을 의미하고, NX 버스 기능 모듈이 감지하는 이벤트가 발생했는지를 표시합니다. TRUE: 이상 있음 FALSE: 이상 없음 미장착 유니트로 설정된 NX 유니트는 'FALSE'입니다. (주의) 이 시스템 정의 변수는 NX102 및 NX1P2 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..8] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로만 액세스 할 수 있습니다. • GetNXBError 다음 전용 명령으로 해제 가능 • ResetNXBError • RestartNXUnit		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

A-7-5 모션 제어 기능 모듈(카테고리명: _MC)

● 기능 분류: 모션 제어 기능 관련

변수명	_MC_ErrSta				
이름	모션 제어 기능 모듈의 이상 Status			글로벌/로컬	글로벌
기능	모션 제어 기능 모듈에서 감지한 이상의 상태를 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미 (A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#40F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• GetMCError • ResetMCError • MC_Reset • MC_GroupReset		

변수명	_MC_ComErrSta				
이름	공통 이상 Status			글로벌/로컬	글로벌
기능	모션 제어에서 공통 처리로 감지한 이상의 상태를 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미 (A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• GetMCError • ResetMCError		

변수명	_MC_AX_ErrSta				
이름	축 이상 Status			글로벌/로컬	글로벌
기능	축 단위로 감지한 이상의 상태를 표시합니다. 최대 256축[*1] 을 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미 (A-112페이지)'를 참조하십시오.				
데이터 형식	ARRAY [0..255] OF WORD[*1]			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• GetMCError • ResetMCError • MC_Reset		

*1. NX102 및 NX1P2 CPU 유니트의 경우에 최대 16축, 데이터 형식은 ARRAY [0..15] OF WORD입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 64축, 데이터 형식은 ARRAY [0..63] OF WORD입니다.

변수명	_MC_GRP_ErrSta				
이름	축 그룹 이상 Status			글로벌/로컬	글로벌
기능	축 그룹 단위로 감지한 이상의 상태를 표시합니다. 최대 64축 그룹[*1] 을 표시합니다. 사용자 프로그램 상에서 사용(직접 참조)할 수 있습니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미 (A-112페이지)'를 참조하십시오.				
데이터 형식	ARRAY [0..63] OF WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	• GetMCError • ResetMCError • MC_GroupReset		

*1. NX102 및 NX1P2 CPU 유니트의 경우에 최대 8축 그룹, 데이터 형식은 ARRAY [0..7] OF WORD입니다.
NJ 시리즈 CPU 유니트의 경우에 최대 32축 그룹, 데이터 형식은 ARRAY [0..31] OF WORD입니다.

변수명	_MC_COM				
이름	공통 변수			글로벌/로컬	글로벌
기능	모션 제어 기능 모듈 공통의 상태 모니터링을 합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.				
데이터 형식	_sCOMMON_REF			값의 범위	–
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_MC_GRP				
이름	축 그룹 변수			글로벌/로컬	글로벌
기능	NX701 CPU 유니트의 경우에 모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 모션 제어 명령에서의 축 그룹 지정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.				
데이터 형식	ARRAY[0..63] OF _sGROUP_REF*1			값의 범위	–
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. NX102 및 NX1P2 CPU 유니트의 경우에 ARRAY[0..7] OF _sGROUP_REF가 됩니다.
NJ 시리즈 CPU 유니트의 경우에 ARRAY[0..31] OF _sGROUP_REF가 됩니다.

변수명	_MC1_GRP				
이름	축 그룹 변수			글로벌/로컬	글로벌
기능	모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다. '_MC1_GRP'와 '_MC_GRP'는 배열 요소 번호가 같으면 같은 값을 참조합니다.				
데이터 형식	ARRAY[0..63] OF _sGROUP_REF			값의 범위	–
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_MC2_GRP				
이름	축 그룹 변수			글로벌/로컬	글로벌
기능	모션 제어 2에서 사용하는 모션 제어 명령에 대한 축 그룹 설정, 다축 협조 제어의 상태 모니터링, 다축 협조 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축 그룹'을 작성하면, 사용자 정의 별칭 축 그룹 변수가 생성됩니다. 통상은 별칭 축 그룹 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY[0..63] OF _sGROUP_REF			값의 범위	–
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_MC_AX				
이름	축 변수			글로벌/로컬	글로벌
기능	NX701 CPU 유니트의 경우에 모션 제어 1에서 사용하는 모션 제어 명령에 대한 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.				
데이터 형식	ARRAY[0..255] OF _sAXIS_REF*1			값의 범위	−
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

*1. NX102 및 NX1P2 CPU 유니트의 경우에 ARRAY[0..15] OF _sAXIS_REF가 됩니다.
NJ 시리즈 CPU 유니트의 경우에 ARRAY[0..63] OF _sAXIS_REF가 됩니다.

변수명	_MC1_AX				
이름	축 변수			글로벌/로컬	글로벌
기능	모션 제어 1에서 사용하는 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다. '_MC1_AX'와 '_MC_AX'는 배열 요소 번호가 같으면 같은 값을 참조합니다.				
데이터 형식	ARRAY[0..255] OF _sAXIS_REF			값의 범위	−
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

변수명	_MC2_AX				
이름	축 변수			글로벌/로컬	글로벌
기능	모션 제어 2에서 사용하는 모션 제어 명령에서의 축 지정, 단축 제어의 상태 모니터링, 단축 제어의 설정 모니터링을 합니다. Sysmac Studio에서 '축'을 작성하면, 사용자 정의의 별칭 축 변수가 생성됩니다. 통상적으로 별칭 축 변수를 사용합니다. 구조체 멤버에 대한 세부 내용은 📖 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY[0..255] OF _sAXIS_REF			값의 범위	−
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

A-7-6 EtherCAT 마스터 기능 모듈(카테고리명: _EC)

● 기능 분류: EtherCAT 통신의 이상 관련

변수명	_EC_ErrSta				
이름	내장 EtherCAT 이상			글로벌/로컬	글로벌
기능	EtherCAT 마스터 기능 모듈의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#40F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 Status 취득 명령 • GetECError EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_PortErr				
이름	통신 포트 이상			글로벌/로컬	글로벌
기능	EtherCAT 마스터용 통신 포트의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 Status 취득 명령 • GetECError EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_MstrErr				
이름	마스터 이상			글로벌/로컬	글로벌
기능	EtherCAT 마스터의 이상과 EtherCAT 마스터가 감지한 슬레이브의 이상을 집약한 시스템 정의 변수입니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 Status 취득 명령 • GetECError EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_SlavErr				
이름	슬레이브 이상			글로벌/로컬	글로벌
기능	EtherCAT 슬레이브 전체의 이상 Status를 집약한 시스템 정의 변수입니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 Status 취득 명령 • GetECError EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_SlavErrTbl				
이름	슬레이브 이상 테이블			글로벌/로컬	글로벌
기능	EtherCAT 슬레이브 단위의 이상 Status를 통지합니다. 이 이상 Status는 실제 시스템 구성의 슬레이브 단위 이상 상태를 표시합니다. 이 변수의 배열은 EtherCAT 슬레이브의 노드 주소(1~512)*1 에 대응하며, 이상이 발생 중인 슬레이브를 표시합니다. 각 비트의 의미는 📖 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	ARRAY [1..512] OF WORD*1			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 Status 취득 명령 • GetECError EtherCAT 이상 해제 명령 • ResetECError		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 노드 주소는 1~192, 데이터 형식은 ARRAY [1..192] OF WORD입니다.

변수명	_EC_MacAdrErr				
이름	MAC 주소 이상			글로벌/로컬	글로벌
기능	MAC 주소가 잘못된 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_LanHwErr				
이름	통신 컨트롤러 이상			글로벌/로컬	글로벌
기능	통신 컨트롤러의 하드웨어 이상을 감지한 경우 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_LinkOffErr				
이름	링크 오프 이상			글로벌/로컬	글로벌
기능	통신 컨트롤러의 링크 미확립을 검출한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_NetCfgErr				
이름	네트워크 구성 정보 이상			글로벌/로컬	글로벌
기능	잘못된 네트워크 구성 정보를 검출한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_NetCfgCmpErr				
이름	네트워크 구성 비교 이상			글로벌/로컬	글로벌
기능	네트워크 구성 정보와 네트워크 실제 구성이 일치하지 않는 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_NetTopologyErr				
이름	네트워크 구성 이상			글로벌/로컬	글로벌
기능	네트워크 구성 이상(연결 대수 초과, 링 연결)이 발생한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_PDCommErr				
이름	프로세스 데이터 통신 이상			글로벌/로컬	글로벌
기능	프로세스 데이터 통신 중에 의도하지 않은 슬레이브 이탈, 가입 또는 슬레이브 WDT 이상을 검출한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_PDTimeoutErr				
이름	프로세스 데이터 수신 타임아웃 이상			글로벌/로컬	글로벌
기능	프로세스 데이터 수신 시에 타임아웃이 발생한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_PDSendErr				
이름	프로세스 데이터 송신 이상			글로벌/로컬	글로벌
기능	프로세스 데이터 송신 이상(프로세스 데이터 통신 주기 내에 송신할 수 없음, 송신 지터 초과)이 발생한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_SlavAdrDupErr				
이름	슬레이브 노드 주소 중복 이상			글로벌/로컬	글로벌
기능	슬레이브 노드 주소가 중복된 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_SlavInitErr				
이름	슬레이브 초기화 이상			글로벌/로컬	글로벌
기능	슬레이브 대상의 초기화 명령에서 이상이 발생한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_SlavAppErr				
이름	슬레이브 어플리케이션 이상			글로벌/로컬	글로벌
기능	슬레이브의 어플리케이션 Status 레지스터에서 이상을 감지한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_MsgErr				
이름	EtherCAT 메시지 이상			글로벌/로컬	글로벌
기능	메시지 기능을 지원하지 않는 슬레이브 대상으로 메시지를 전송했거나, 슬레이브로 전송한 메시지의 응답이 이상한 형식인 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	CoE 메시지(CoE SDO 읽기) • EC_CoESDORead CoE 메시지(CoE SDO 쓰기) • EC_CoESDOWrite		

변수명	_EC_SlavEmergErr				
이름	비상 메시지 감지			글로벌/로컬	글로벌
기능	슬레이브가 비상 메시지를 전송했음을 마스터가 감지한 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

변수명	_EC_IndataInvalidErr*1				
이름	프로세스 데이터 입력값 무효 이상			글로벌/로컬	글로벌
기능	EtherCAT 마스터가 Operational 상태이고, 프로세스 데이터 통신을 정상적으로 실행할 수 없고, 입력 데이터 무효인 상태가 다음 중 한 기간 이상 계속된 경우에 TRUE가 됩니다. • 태스크 주기가 10ms 이 내인 경우: 100ms • 태스크 주기가 10ms를 초과하는 경우: 태스크 주기의 10배				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

*1. CPU 유니트의 유니트 버전 Ver.1.13에서 추가된 시스템 정의 변수입니다.

변수명	_EC_CommErrTbl				
이름	통신 이상 슬레이브 테이블			글로벌/로컬	글로벌
기능	슬레이브 노드 주소 순서를 테이블로 표시합니다. 마스터에서 검출한 이상이 발생 중인 슬레이브의 해당 위치는 TRUE가 됩니다.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 이상 해제 명령 • ResetECError		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

(주의) EtherCAT 통신의 이상과 관련된 모든 시스템 정의 변수는 이상 요인을 제거하고 발생 중인 컨트롤러 이상을 해제(Sysmac Studio의 트러블 슈팅/ResetECError 명령)할 때까지 값이 변화하지 않습니다.

변수명	_EC_CycleExceeded				
이름	EtherCAT 통신 주기 설정 이상			글로벌/로컬	글로벌
기능	기동 시에 설정한 통신 주기로 통신할 수 없는 경우에 TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: EtherCAT 통신의 Status 관련

변수명	_EC_RegSlavTbl				
이름	등록 슬레이브 테이블			글로벌/로컬	글로벌
기능	네트워크 구성 정보에 등록된 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 등록된 경우에 해당 위치는 TRUE가 됩니다.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_EntrySlavTbl				
이름	네트워크 가입 슬레이브 테이블			글로벌/로컬	글로벌
기능	네트워크 가입 슬레이브 목록을 통지합니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 네트워크 가입했을 때, 해당 위치는 TRUE가 됩니다.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_MBXSlavTbl				
이름	메시지 통신 가능 슬레이브 테이블			글로벌/로컬	글로벌
기능	메시지 통신이 가능한 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 메시지 통신 가능 상태(Pre-Operational, Safe-Operational, Operational)로 이동한 경우에 해당 위치는 TRUE가 됩니다. (주의) EtherCAT 슬레이브와 메시지 통신하는 경우에 이 데이터를 참조하여 해당 슬레이브와 메시지 통신이 가능한지 여부를 확인하십시오.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 슬레이브 이탈 • EC_DisconnectSlave EtherCAT 슬레이브 재가입 • EC_ConnectSlave		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_PDSlavTbl				
이름	프로세스 데이터 통신 중 슬레이브 테이블			글로벌/로컬	글로벌
기능	프로세스 데이터 통신 중인 슬레이브 목록입니다. 슬레이브 노드 주소 순서를 테이블로 표시합니다. 슬레이브가 IN, OUT 모두 프로세스 데이터 유효 상태(Operational)로 이동한 경우에 해당 위치는 TRUE가 됩니다. (주의) EtherCAT 슬레이브를 제어하는 경우에 이 데이터를 참조하여 해당 슬레이브의 데이터 유효성을 확인하십시오.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 슬레이브 이탈 • EC_DisconnectSlave EtherCAT 슬레이브 재가입 • EC_ConnectSlave		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_DisconnSlavTbl				
이름	이탈 명령 중 슬레이브 테이블			글로벌/로컬	글로벌
기능	슬레이브 노드 주소 순서를 테이블로 표시합니다. 이탈 명령 중인 슬레이브의 해당 위치는 TRUE가 됩니다.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 슬레이브 이탈 • EC_DisconnectSlave EtherCAT 슬레이브 재가입 • EC_ConnectSlave		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_DisableSlavTbl				
이름	무효 설정 슬레이브 테이블			글로벌/로컬	글로벌
기능	슬레이브 노드 주소 순서를 테이블로 표시합니다. 무효로 설정된 슬레이브의 해당 위치는 TRUE가 됩니다.				
데이터 형식	ARRAY [1..512] OF BOOL*1			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트의 경우에 ARRAY [1..192] OF BOOL입니다.

변수명	_EC_PDActive				
이름	프로세스 데이터 통신 Status			글로벌/로컬	글로벌
기능	전체 슬레이브*1 와의 프로세스 데이터 통신 중일 때 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	EtherCAT 슬레이브 이탈 • EC_DisconnectSlave EtherCAT 슬레이브 재가입 • EC_ConnectSlave		

*1. 무효로 설정된 슬레이브는 포함하지 않습니다.

변수명	_EC_PktMonStop				
이름	패킷 모니터링 정지			글로벌/로컬	글로벌
기능	패킷 모니터링 정지 중일 때 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	패킷 모니터링 정지 • EC_StopMon 패킷 모니터링 기동 • EC_StartMon		

변수명	_EC_LinkStatus				
이름	링크 Status			글로벌/로컬	글로벌
기능	통신 컨트롤러의 링크 Status가 Link ON인 경우에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EC_PktSaving				
이름	패킷 데이터 파일 저장 중			글로벌/로컬	글로벌
기능	패킷 데이터 파일 저장 처리 중인지 여부를 표시합니다. TRUE: 패킷 데이터 파일 저장 처리 중 FALSE: 패킷 데이터 파일 저장 미처리 중				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	패킷 데이터의 파일 저장 • EC_SaveMon		

변수명	_EC_InDataInvalid				
이름	입력 데이터 무효			글로벌/로컬	글로벌
기능	프라이머리 정주기 태스크에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

(주의) 1. EtherCAT 통신 Status에 관련된 모든 시스템 정의 변수는 현재 Status를 표시합니다.

변수명	_EC_InData1Invalid				
이름	입력 데이터 1 무효			글로벌/로컬	글로벌
기능	프라이머리 정주기 태스크에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

(주의) 1. EtherCAT 통신 Status에 관련된 모든 시스템 정의 변수는 현재 Status를 표시합니다.

변수명	_EC_InData2Invalid				
이름	입력 데이터 2 무효			글로벌/로컬	글로벌
기능	정주기 태스크(실행 우선도 5)에서 실행하는 프로세스 데이터 통신이 정상적으로 실행할 수 없어서 입력값이 잘못되었을 때, TRUE가 됩니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다. (주의) NX102 및 NX1P2 CPU 유니트의 경우에 항상 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

(주의) 1. EtherCAT 통신 Status에 관련된 모든 시스템 정의 변수는 현재 Status를 표시합니다.

● 기능 분류: EtherCAT 통신의 진단 통계 정보 관련

변수명	_EC_StatisticsLogEnable*1				
이름	진단 통계 정보 로그 허가			글로벌/로컬	글로벌
기능	진단 통계 정보 로그 기능을 시작할 때 FALSE에서 TRUE로 합니다. 진단 통계 정보 로그 기능을 종료할 때 TRUE에서 FALSE로 합니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_EC_StatisticsLogCycleSec*1				
이름	진단 통계 정보 로그 저장 주기			글로벌/로컬	글로벌
기능	진단 통계 정보의 로그 기능에 대해 통계 진단 정보의 쓰기를 실시하는 주기를 초 단위로 지정합니다. '0'을 지정한 경우에 진단 통계 정보 로그 기능을 종료할 때에 1회만 쓰기를 실시합니다. (주의) 진단 통계 정보 로그 기능의 실행 중에 이 시스템 정의 변수의 값을 변경해도 실행 중인 쓰기 주기는 변경되지 않습니다.				
데이터 형식	UINT			값의 범위	0, 30~1800
R/W 액세스	RW	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_EC_StatisticsLogBusy*1				
이름	진단 통계 정보 로그 실행 상태 플래그			글로벌/로컬	글로벌
기능	진단 통계 정보 로그 기능을 실행 중에 TRUE가 됩니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

변수명	_EC_StatisticsLogErr*1				
이름	진단 통계 정보 로그 이상 종료 플래그			글로벌/로컬	글로벌
기능	진단 통계 정보 로그 기능 시작에 실패했거나, 실행 중에 로그를 쓸 수 없었던 경우에 TRUE가 됩니다. 진단 통계 정보 로그 기능 시작 후에 _EC_StatisticsLogBusy(진단 통계 정보 로그 실행 상태 플래그)가 FALSE가 됐을 때, 이 플래그의 값이 확정됩니다. 이상 종료에는 다음 요인이 포함됩니다. • SD 메모리 카드의 용량이 상한에 도달하여, 레코드를 로그 파일에 추가할 수 없음 • SD 메모리 카드가 쓰기 방지됨 • SD 메모리 카드가 장착되지 않음 • _EC_StatisticsLogCycleSec(진단 통계 정보 로그 저장 주기)에 지정된 값이 잘못되어 기능을 시작할 수 없음				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	−		

*1. CPU 유니트의 유니트 버전 Ver.1.11에서 추가된 시스템 정의 변수입니다.

A-7-7 EtherNet/IP 기능 모듈(카테고리명: _EIP)

● 기능 분류: EtherNet/IP 통신의 이상 관련

변수명	_EIP_ErrSta				
이름	내장 EtherNet/IP 이상			글로벌/로컬	글로벌
기능	내장 EtherNet/IP 포트의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_PortErr(통신 포트 1 이상) • _EIP2_PortErr(통신 포트 2 이상) • _EIP1_CipErr(CIP 통신 1 이상) • _EIP2_CipErr(CIP 통신 2 이상) • _EIP_TcpAppErr(TCP 어플리케이션 통신 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_PortErr(통신 포트 이상) • _EIP_CipErr(CIP 통신 이상) • _EIP_TcpAppErr(TCP 어플리케이션 통신 이상) (주의) 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP_PortErr				
이름	통신 포트 이상			글로벌/로컬	글로벌
기능	통신 포트의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_MacAdrErr(통신 포트 1 MAC 주소 이상) • _EIP1_LanHwErr(통신 포트 1 통신 컨트롤러 이상) • _EIP1_EtnCfgErr(통신 포트 1 Ethernet 기본 설정 이상) • _EIP1_IPAdrCfgErr(통신 포트 1 IP 주소 설정 이상) • _EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) • _EIP1_BootpErr(통신 포트 1 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_MacAdrErr(MAC 주소 이상) • _EIP_LanHwErr(통신 컨트롤러 이상) • _EIP_EtnCfgErr(Ethernet 기본 설정 이상) • _EIP_IPAdrCfgErr(IP 주소 설정 이상) • _EIP_IPAdrDupErr(IP 주소 중복 이상) • _EIP_BootpErr(BOOTP 서버 연결 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP1_PortErr				
이름	통신 포트 1 이상			글로벌/로컬	글로벌
기능	통신 포트의 1의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP1_MacAdrErr(통신 포트 1 MAC 주소 이상) • _EIP1_LanHwErr(통신 포트 1 통신 컨트롤러 이상) • _EIP1_EtnCfgErr(통신 포트 1 Ethernet 기본 설정 이상) • _EIP1_IPAdrCfgErr(통신 포트 1 IP 주소 설정 이상) • _EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) • _EIP1_BootpErr(통신 포트 1 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP2_PortErr				
이름	통신 포트 2 이상			글로벌/로컬	글로벌
기능	통신 포트 2의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP2_MacAdrErr(통신 포트 2 MAC 주소 이상) • _EIP2_LanHwErr(통신 포트 2 통신 컨트롤러 이상) • _EIP2_EtnCfgErr(통신 포트 2 Ethernet 기본 설정 이상) • _EIP2_IPAdrCfgErr(통신 포트 2 IP 주소 설정 이상) • _EIP2_IPAdrDupErr(통신 포트 2 IP 주소 중복 이상) • _EIP2_BootpErr(통신 포트 2 BOOTP 서버 연결 이상) • _EIP_DNSCfgErr(DNS 설정 이상) • _EIP_DNSSrvErr(DNS 서버 연결 이상) • _EIP_IPRTblErr(IP 라우트 테이블 이상) (주의) '링크 오프 감지' 또는 '내장 EtherNet/IP 이상' 시에 이벤트 로그에 등록하고, 이 이상 비트를 ON합니다. 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP_CipErr				
이름	CIP 통신 이상			글로벌/로컬	글로벌
기능	CIP 통신의 이상 Status 변수입니다. NX 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP1_IdentityErr(CIP 통신 1 Identity 정보가 잘못됨) • _EIP1_TDLinkCfgErr(CIP 통신 1 태그 데이터 링크 설정 이상) • _EIP1_TDLinkOpnErr(CIP 통신 1 태그 데이터 링크 연결 확립 실패) • _EIP1_TDLinkErr(CIP 통신 1 태그 데이터 링크 통신 이상) • _EIP1_TagAdrErr(CIP 통신 1 태그 해결 이상) • _EIP1_MultiSwONErr(CIP 통신 1 복수 스위치 동시 ON 이상) NJ 시리즈 CPU 유니트의 경우에 다음 이상 플래그를 집약합니다. • _EIP_IdentityErr(Identity 정보가 잘못됨) • _EIP_TDLinkCfgErr(태그 데이터 링크 설정 이상) • _EIP_TDLinkOpnErr(태그 데이터 링크 연결 확립 실패) • _EIP_TDLinkErr(태그 데이터 링크 통신 이상) • _EIP_TagAdrErr(태그 해결 이상) • _EIP_MultiSwONErr(복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고 이 이상을 ON합니다. 각 비트의 의미는 📖'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP1_CipErr				
이름	CIP 통신 1 이상			글로벌/로컬	글로벌
기능	CIP 통신 1의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP1_IdentityErr(CIP 통신 1 Identity 정보가 잘못됨) • _EIP1_TDLinkCfgErr(CIP 통신 1 태그 데이터 링크 설정 이상) • _EIP1_TDLinkOpnErr(CIP 통신 1 태그 데이터 링크 연결 확립 실패) • _EIP1_TDLinkErr(CIP 통신 1 태그 데이터 링크 통신 이상) • _EIP1_TagAdrErr(CIP 통신 1 태그 해결 이상) • _EIP1_MultiSwONErr(CIP 통신 1 복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고 이 이상을 ON합니다. 각 비트의 의미는 📖'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP2_CipErr				
이름	CIP 통신 2 이상			글로벌/로컬	글로벌
기능	CIP 통신 2의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP2_IdentityErr(CIP 통신 2 Identity 정보가 잘못됨) • _EIP2_TDLinkCfgErr(CIP 통신 2 태그 데이터 링크 설정 이상) • _EIP2_TDLinkOpnErr(CIP 통신 2 태그 데이터 링크 연결 확립 실패) • _EIP2_TDLinkErr(CIP 통신 2 태그 데이터 링크 통신 이상) • _EIP2_TagAdrErr(CIP 통신 2 태그 해결 이상) • _EIP2_MultiSwONErr(CIP 통신 2 복수 스위치 동시 ON 이상) (주의) '태그 해결 이상' 시에 이벤트 로그에 등록하고 이 이상을 ON합니다. 각 비트의 의미는 📖'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP_TcpAppErr				
이름	TCP 어플리케이션 통신 이상			글로벌/로컬	글로벌
기능	TCP 어플리케이션 통신의 이상 Status 변수입니다. 다음 이상 플래그를 집약합니다. • _EIP_TcpAppCfgErr(TCP 어플리케이션 설정 이상) • _EIP_NTPSrvErr(NTP 서버 연결 이상) (주의) 각 비트의 의미는 'A-6-8 이상 Status 각 비트의 의미(A-112페이지)'를 참조하십시오.				
데이터 형식	WORD			값의 범위	16#0000~16#00F0
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	사용자 프로그램에서는 다음과 같은 전용 명령으로 액세스 할 수 있습니다. • GetEIPError		

변수명	_EIP_MacAdrErr				
이름	MAC 주소 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 전원 ON 시에 통신 포트 1의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 전원 ON 시에 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_MacAdrErr				
이름	통신 포트 1 MAC 주소 이상			글로벌/로컬	글로벌
기능	전원 ON 시에 통신 포트 1의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_MacAdrErr				
이름	통신 포트 2 MAC 주소 이상			글로벌/로컬	글로벌
기능	전원 ON 시에 통신 포트 2의 MAC 주소 읽기 이상이 발생했음을 나타냅니다. TRUE: 이상 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_LanHwErr				
이름	통신 컨트롤러 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_LanHwErr				
이름	통신 포트 1 통신 컨트롤러 이상			글로벌/로컬	글로벌
기능	통신 포트 1에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_LanHwErr				
이름	통신 포트 2 통신 컨트롤러 이상			글로벌/로컬	글로벌
기능	통신 포트 2에서 통신 컨트롤러 고장이 발생했음을 나타냅니다. TRUE: 고장 발생 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_EtnCfgErr				
이름	Ethernet 기본 설정 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 통신 포트 1의 Ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 Ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_EtnCfgErr				
이름	통신 포트 1 Ethernet 기본 설정 이상			글로벌/로컬	글로벌
기능	통신 포트 1의 ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_EtnCfgErr				
이름	통신 포트 2 Ethernet 기본 설정 이상			글로벌/로컬	글로벌
기능	통신 포트 2의 ethernet 통신 속도 설정(Speed/Duplex)이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_IPAdrCfgErr				
이름	IP 주소 설정 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 통신 포트 1의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 • 기본 게이트웨이 설정이 잘못됨 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_IPAdrCfgErr				
이름	통신 포트 1 IP 주소 설정 이상			글로벌/로컬	글로벌
기능	통신 포트 1의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_IPAdrCfgErr				
이름	통신 포트 2 IP 주소 설정 이상			글로벌/로컬	글로벌
기능	통신 포트 2의 IP 주소 설정 이상을 나타냅니다. TRUE: • IP 주소 설정값이 잘못됨 • 읽기 실패 • BOOTP 서버에서 취득한 IP 주소가 불일치 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_IPAdrDupErr				
이름	IP 주소 중복 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_IPAdrDupErr				
이름	통신 포트 1 IP 주소 중복 이상			글로벌/로컬	글로벌
기능	통신 포트 1에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_IPAdrDupErr				
이름	통신 포트 2 IP 주소 중복 이상			글로벌/로컬	글로벌
기능	통신 포트 2에서 IP 주소 중복이 발생했음을 나타냅니다. TRUE: 중복 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_DNSCfgErr*1				
이름	DNS 설정 이상			글로벌/로컬	글로벌
기능	DNS 설정값 또는 hosts 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

*1. NJ 시리즈 CPU 유니트에서는 CPU 유니트의 유니트 버전 Ver.1.11 이상에서 사용 가능합니다.

변수명	_EIP_BootpErr				
이름	BOOTP 서버 연결 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 통신 포트 1에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. NJ 시리즈 CPU 유니트의 경우에 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_BootpErr				
이름	통신 포트 1 BOOTP 서버 연결 이상			글로벌/로컬	글로벌
기능	통신 포트 1에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_BootpErr				
이름	통신 포트 2 BOOTP 서버 연결 이상			글로벌/로컬	글로벌
기능	통신 포트 2에서 BOOTP 서버 연결에 실패했음을 나타냅니다. TRUE: BOOTP 서버 연결에 실패(타임아웃)했습니다. FALSE: BOOTP 설정이 되지 않았거나, BOOTP 설정이 되어 있고 정상적으로 BOOTP 서버에서 IP 주소를 취득했습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_IPRTblErr				
이름	IP 라우트 테이블 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 기본 게이트웨이 설정 또는 IP 라우터 테이블 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 IP 라우터 테이블 설정 또는 hosts 설정이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_IdentityErr				
이름	Identity 정보가 잘못됨			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 Identify 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 Identify 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_IdentityErr				
이름	CIP 통신 1 Identity 정보가 잘못됨			글로벌/로컬	글로벌
기능	CIP 통신 1 Identity 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_IdentityErr				
이름	CIP 통신 2 Identity 정보가 잘못됨			글로벌/로컬	글로벌
기능	CIP 통신 2 Identity 정보(사용자에 의한 재기입 불가능)가 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkCfgErr				
이름	태그 데이터 링크 설정 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkCfgErr				
이름	CIP 통신 1 태그 데이터 링크 설정 이상			글로벌/로컬	글로벌
기능	CIP 통신 1의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkCfgErr				
이름	CIP 통신 2 태그 데이터 링크 설정 이상			글로벌/로컬	글로벌
기능	CIP 통신 2의 태그 데이터 링크 설정값이 잘못되었거나 읽기에 실패했음을 나타냅니다. TRUE: 잘못되었거나 읽기 실패 FALSE: 정상 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkOpnErr				
이름	태그 데이터 링크 연결 확립 실패			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkOpnErr				
이름	CIP 통신 1 태그 데이터 링크 연결 확립 실패			글로벌/로컬	글로벌
기능	CIP 통신 1의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkOpnErr				
이름	CIP 통신 2 태그 데이터 링크 연결 확립 실패			글로벌/로컬	글로벌
기능	CIP 통신 2의 태그 데이터 링크 연결 확립이 실패했음을 나타냅니다. TRUE: 태그 데이터 링크 연결 확립이 다음 요인에 의해 실패했습니다. • 태그 데이터 링크 파라메터에 등록되어 있는 상대 노드 정보와 실제 노드 정보가 다름 • 상대 노드로부터 응답이 없음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkErr				
이름	태그 데이터 링크 통신 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkErr				
이름	CIP 통신 1 태그 데이터 링크 통신 이상			글로벌/로컬	글로벌
기능	CIP 통신 1의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkErr				
이름	CIP 통신 2 태그 데이터 링크 통신 이상			글로벌/로컬	글로벌
기능	CIP 통신 2의 태그 데이터 링크 연결에 타임아웃이 발생했음을 나타냅니다. TRUE: 타임아웃 발생 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TagAdrErr				
이름	태그 해결 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TagAdrErr				
이름	CIP 통신 1 태그 해결 이상			글로벌/로컬	글로벌
기능	CIP 통신 1의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TagAdrErr				
이름	CIP 통신 2 태그 해결 이상			글로벌/로컬	글로벌
기능	CIP 통신 2의 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했음을 나타냅니다. TRUE: 태그 해결(태그 이름에서 주소를 특정하는 것)에 실패했습니다. 다음과 같은 요인이 있습니다. • 네트워크 변수와 태그 설정 간에 크기가 다름 • 태그 데이터 링크의 설정 입출력 방향과 CPU 유니트 내의 변수 입출력 방향이 일치하지 않음 • CPU 유니트 내에 태그 설정에 해당하는 네트워크 변수가 존재하지 않음 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_MultiSwONErr				
이름	복수 스위치 동시 ON 이상			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_MultiSwONErr				
이름	CIP 통신 1 복수 스위치 동시 ON 이상			글로벌/로컬	글로벌
기능	CIP 통신 1에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_MultiSwONErr				
이름	CIP 통신 2 복수 스위치 동시 ON 이상			글로벌/로컬	글로벌
기능	CIP 통신 2에서 복수의 스위치가 동시에 ON되었음을 나타냅니다. TRUE: 복수의 데이터 링크 시작/데이터 링크 정지 스위치가 동시에 TRUE가 되었습니다. FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TcpAppCfgErr				
이름	TCP 어플리케이션 설정 이상			글로벌/로컬	글로벌
기능	TRUE: 적어도 1개 이상의 TCP 어플리케이션 기능(FTP, NTP, SNMP)의 설정값이 잘못되었습니다. 읽기에 실패했습니다. FALSE: 정상				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_NTPSrvErr				
이름	NTP 서버 연결 이상			글로벌/로컬	글로벌
기능	TRUE: NTP 클라이언트가 서버 연결에 실패(타임아웃)했습니다. FALSE: NTP의 설정이 되지 않았거나 NTP의 설정이 되어 있고 연결에 성공했습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_DNSSrvErr				
이름	DNS 서버 연결 이상			글로벌/로컬	글로벌
기능	TRUE: DNS 클라이언트가 서버 연결에 실패(타임아웃)했습니다. FALSE: DNS의 설정이 되지 않았거나 DNS의 설정이 되어 있고 연결에 성공했습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

● 기능 분류: EtherNet/IP 통신의 Status 관련

변수명	_EIP_EtnOnlineSta				
이름	온라인			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP 포트의 통신 포트 1의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP 포트의 통신 포트의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_EtnOnlineSta				
이름	통신 포트 1 온라인			글로벌/로컬	글로벌
기능	내장 EtherNet/IP 포트의 통신 포트 1의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_EtnOnlineSta				
이름	통신 포트 2 온라인			글로벌/로컬	글로벌
기능	내장 EtherNet/IP 포트의 통신 포트 2의 통신 기능을 이용할 수 있다는 것(링크 ON이면서 IP 주소 결정, 그리고 이상이 발생하지 않았을 것)을 나타냅니다. TRUE: 이용 가능 FALSE: 초기 처리 중, 재기동 중, 이상 발생, 링크 오프에 의해 내장 EtherNet/IP 포트의 통신 기능을 이용할 수 없습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkRunSta				
이름	태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkRunSta				
이름	CIP 통신 1 태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	CIP 통신 1에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkRunSta				
이름	CIP 통신 2 태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	CIP 통신 2에서 적어도 1개 이상의 연결이 정상 동작하고 있음을 나타냅니다. TRUE: 정상 동작 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkAllRunSta				
이름	전체 태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. NJ 시리즈 CPU 유니트의 경우에 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkAllRunSta				
이름	CIP 통신 1 전체 태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	CIP 통신 1의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkAllRunSta				
이름	CIP 통신 2 전체 태그 데이터 링크 통신 중			글로벌/로컬	글로벌
기능	CIP 통신 2의 전체 태그 데이터 링크가 통신 중임을 나타냅니다. TRUE: 오리지네이터로서 모든 연결에서 태그 데이터 링크가 통신 중임을 나타냅니다. FALSE: 적어도 1개 이상의 연결에 이상이 발생했습니다. (주의) 이 시스템 정의 변수는 NX701 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_RegTargetSta[255]				
이름	등록 대상 노드 정보			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP의 CIP 통신 1에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 노드와의 연결이 등록되지 않았습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 노드와의 연결이 등록되지 않았습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_RegTargetSta[255]				
이름	CIP 통신 1 등록 대상 노드 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP의 CIP 통신 1에 연결이 등록되어 있는 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 노드와의 연결이 등록되지 않았습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_RegTargetSta[255]				
이름	CIP 통신 2 등록 대상 노드 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP의 CIP 통신 2에 연결이 등록된 노드 목록을 나타냅니다. 이 변수는 내장 EtherNet/IP가 오리지네이터로 된 경우에만 유효가 됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 등록되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 노드와의 연결이 등록되지 않았습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_EstbTargetSta[255]				
이름	정상 대상 노드 정보			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP의 CIP 통신 1이 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_EstbTargetSta[255]				
이름	CIP 통신 1 정상 대상 노드 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP의 CIP 통신 1이 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_EstbTargetSta[255]				
이름	CIP 통신 2 정상 대상 노드 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP의 CIP 통신 2가 연결을 정상적으로 확립한 노드 목록을 나타냅니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 노드와의 연결이 정상적으로 확립되어 있습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x' 노드와의 연결에 미확립 또는 이상이 발생한 것이 있습니다. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TargetPLCModeSta[255]				
이름	대상 PLC 동작 모드			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TargetPLCModeSta[255]				
이름	CIP 통신 1 대상 PLC 동작 모드			글로벌/로컬	글로벌
기능	내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TargetPLCModeSta[255]				
이름	CIP 통신 2 대상 PLC 동작 모드			글로벌/로컬	글로벌
기능	내장 EtherNet/IP가 오리지네이터인 CIP 통신 2의 연결에 대해 해당 대상 노드의 컨트롤러 운전 정보를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값이 유지됩니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전하고 있는 상태입니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TargetPLCErr[255]				
이름	대상 PLC 이상 정보			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 NJ 시리즈 CPU 유니트의 경우에 내장 EtherNet/IP가 오리지네이터인 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TargetPLCErr[255]				
이름	CIP 통신 1 대상 PLC 이상 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP가 오리지네이터인 CIP 통신 1의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TargetPLCErr[255]				
이름	CIP 통신 2 대상 PLC 이상 정보			글로벌/로컬	글로벌
기능	내장 EtherNet/IP가 오리지네이터인 CIP 통신 2의 연결에 대해 해당 대상 노드의 컨트롤러 본체의 이상 상태(운전 정지, 계속 이상의 논리합)를 나타냅니다. 이 영역의 정보는 정상 대상 노드 정보가 TRUE일 때에만 유효합니다. FALSE인 경우는 직전의 값을 유지합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 컨트롤러가 운전 계속 또는 정지 이상이 발생했습니다. 배열 [x]가 FALSE: 상기 이외 (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TargetNodeErr[255]				
이름	대상 노드 이상 정보			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 CIP 통신 1의 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나(등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE) 대상 노드와의 연결은 확립되었으나(등록 대상 노드 정보가 TRUE이고, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다(대상 PLC 이상 정보가 TRUE). NJ 시리즈 CPU 유니트의 경우에 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나(등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE) 대상 노드와의 연결은 확립되었으나(등록 대상 노드 정보가 TRUE이고, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다(대상 PLC 이상 정보가 TRUE).				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TargetNodeErr[255]				
이름	CIP 통신 1 대상 노드 이상 정보			글로벌/로컬	글로벌
기능	CIP 통신 1의 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나(등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE) 대상 노드와의 연결은 확립되었으나(등록 대상 노드 정보가 TRUE이고, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다(대상 PLC 이상 정보가 TRUE). (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TargetNodeErr[255]				
이름	CIP 통신 2 대상 노드 이상 정보			글로벌/로컬	글로벌
기능	CIP 통신 2의 등록 대상 노드 정보의 해당하는 연결이 확립되지 않았거나, 대상 컨트롤러에 이상이 발생했음을 나타냅니다. 이 영역은 등록 대상 노드 정보가 TRUE인 경우에만 유효합니다. 배열 [x]가 TRUE: 대상 노드 ID 'x'의 대상 노드와의 연결이 정상적으로 확립되지 않았거나(등록 대상 노드 정보가 TRUE이고 정상 대상 노드 정보가 FALSE), 대상 노드와의 연결은 확립되었으나 대상 컨트롤러에 이상이 발생했습니다. 배열 [x]가 FALSE: 대상 노드 ID 'x'의 대상 노드가 등록되지 않았거나(등록 대상 노드 정보가 FALSE) 대상 노드와의 연결은 확립되었으나(등록 대상 노드 정보가 TRUE이고, 정상 대상 노드 정보가 TRUE) 대상 컨트롤러에 이상이 발생했습니다(대상 PLC 이상 정보가 TRUE). (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	ARRAY [0..255] OF BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_NTPResult			멤버명	.ExecTime
이름	NTP의 최종 동작 시각			글로벌/로컬	글로벌
기능	NTP 정상 종료 시각을 등록합니다. NTP 서버에서 정상적으로 시각을 취득한 경우에 NTP 서버에서 취득한 시각을 등록합니다. NTP 서버에서 정상적으로 시각을 취득할 수 없는 경우에는 등록하지 않습니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다.				
데이터 형식	구조체: _sNTP_RESULT 멤버: DATE_AND_TIME			값의 범위	데이터 형식에 따름
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	GetNTPStatus 명령으로 읽기 가능		

변수명	_EIP_NTPResult			멤버명	.ExecNormal
이름	NTP의 동작 결과			글로벌/로컬	글로벌
기능	NTP가 정상적으로 종료했는지 여부를 나타냅니다. TRUE: NTP 정상 종료를 나타냅니다. FALSE: NTP 이상 종료 또는 한번도 NTP를 실행하지 않았음을 나타냅니다. (주의) 실시간성이라는 관점에서 사용자 프로그램 상에서는 사용할 수 없습니다. 통신을 통해 외부에서 상태를 참조할 때에만 사용합니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	불가능	관련 명령	GetNTPStatus 명령으로 읽기 가능		

● 기능 분류: EtherNet/IP 통신의 스위치 관련

변수명	_EIP_TDLinkStartCmd				
이름	태그 데이터 링크 통신 시작 스위치			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. NJ 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkStartCmd				
이름	CIP 통신 1 태그 데이터 링크 통신 시작 스위치			글로벌/로컬	글로벌
기능	FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkStartCmd				
이름	CIP 통신 2 태그 데이터 링크 통신 시작 스위치			글로벌/로컬	글로벌
기능	FALSE에서 TRUE가 되면, CIP 통신 2의 태그 데이터 링크를 시작합니다. 태그 데이터 링크 동작이 시작되면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP_TDLinkStopCmd				
이름	태그 데이터 링크 통신 정지 스위치			글로벌/로컬	글로벌
기능	NX 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. NJ 시리즈 CPU 유니트의 경우에 FALSE에서 TRUE가 되면, 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP1_TDLinkStopCmd				
이름	CIP 통신 1 태그 데이터 링크 통신 정지 스위치			글로벌/로컬	글로벌
기능	FALSE에서 TRUE가 되면, CIP 통신 1의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX 시리즈 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

변수명	_EIP2_TDLinkStopCmd				
이름	CIP 통신 2 태그 데이터 링크 통신 정지 스위치			글로벌/로컬	글로벌
기능	FALSE에서 TRUE가 되면, CIP 통신 2의 태그 데이터 링크가 정지합니다. 태그 데이터 링크 동작이 정지하면, 자동으로 FALSE가 됩니다. (주의) 자동적으로 FALSE가 될 때까지는 이 스위치를 사용자 프로그램 또는 Sysmac Studio 등으로 강제 조작하지 마십시오. (주의) 이 시스템 정의 변수는 NX701 및 NX102 CPU 유니트에서만 사용할 수 있습니다.				
데이터 형식	BOOL			값의 범위	TRUE, FALSE
R/W 액세스	R/W	유지	비유지	네트워크 공개	공개
사용자 프로그램에서의 사용 가능 여부	가능	관련 명령	–		

A-8 CPU 유니트 내 데이터 속성

CPU 유니트 내 데이터의 전원 OFF 시, 전원 투입 시, 동작 모드 변경 시, 전체 정지 이상 레벨의 컨트롤러 이상 발생 시의 유지/비유지 등의 속성은 다음과 같습니다.

CPU 유니트 내 데이터 항목			전원 OFF 시의 유지 유무와 방법 ○:전원 OFF 유지 ×:전원 OFF 유지 안 함	전원 투입 시	상태 변경		덮어쓰기 금지 방지 중의 쓰기	Sysmac Studio의 전송 기능	쓰기 가능한 동작 모드	운전 모드에서의 재기입
					프로그램 모드←→운전 모드 변경 시	전체 정지 이상 레벨의 컨트롤러 이상 발생 시		동기 기능의 대상		
사용자 프로그램	POUs, 사용자 프로그램 실행용 ID(사용자 프로그램 내)		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드/운전 모드(온라인 편집)	○ 온라인 편집
태스크 설정	태스크 설정		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
변수	변수 테이블 (변수의 값 제외)	디바이스 변수	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
		사용자 정의 변수	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드/운전 모드(온라인 편집)	○ 온라인 편집
데이터 형식		사용자 정의 데이터 형식	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드/운전 모드(온라인 편집)	○ 온라인 편집
컨트롤러명		CPU 유니트 이름	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드/운전 모드	○
		내장 EtherNet/IP 포트명	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	가능	×	프로그램 모드/운전 모드	○
컨트롤러 설정	동작 설정	동작 설정 이상 설정	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	CPU 유니트 이름: 운전 모드/프로그램 모드 그 외: 프로그램 모드	×
	보안 설정	전원 투입 시의 보호 설정	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	쓰기 방지, 그 외: 프로그램 모드	○

CPU 유니트 내 데이터 항목			전원 OFF 시의 유지 유무와 방법 ○:전원 OFF 유지 ×:전원 OFF 유지 안 함	전원 투입 시	상태 변경		덮어쓰기금지 방지 중의 쓰기	Sysmac Studio의 전송 기능	쓰기 가능한 동작 모드	운전 모드에서의 재기입
					프로그램 모드←→운전 모드 변경 시	전체 정지 이상 레벨의 컨트롤러 이상 발생 시		동기 기능의 대상		
컨트롤러 설정	내장 EtherNet/IP 포트 설정	TCP/IP 내장 EtherNet/IP 포트 LINK 설정 서비스 설정 SNMP 설정 SNMP 트랩 설정 NTP 설정 FTP 설정 IP 라우터 테이블	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
		내장 EtherNet/IP 포트용 태그 데이터 링크 설정	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	가능	×	프로그램 모드/운전 모드	×
	FINS 설정	노드 주소 설정 FINS/UDP FINS/TCP FINS 라우팅 테이블 FINS 쓰기 방지	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
	내장 I/O 설정		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
	옵션 보드 설정		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
	CJ 유니트용 메모리 설정		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×

CPU 유니트 내 데이터 항목			전원 OFF 시의 유지 유무와 방법 ○:전원 OFF 유지 ×:전원 OFF 유지 안 함	전원 투입 시	상태 변경		덮어쓰기 금지 방지 중의 쓰기	Sysmac Studio의 전송 기능	쓰기 가능한 동작 모드	운전 모드에서의 재기입
					프로그램 모드←→운전 모드 변경 시	전체 정지 이상 레벨의 컨트롤러 이상 발생 시		동기 기능의 대상		
OPC UA 설정	OPC UA 서버 설정	서버 설정 네트워크 설정 엔드 포인트 설정 실행 로그 설정	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
	클라이언트 증명서 설정	증명서 신뢰 목록 증명서 신뢰 목록(실효) 증명서 거부 목록 루트 증명서/중간 증명서 목록 루트 증명서/중간 증명서 목록(실효)	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	가능	×	프로그램 모드/운전 모드	○
	사용자 지정	사용자 인증정의 파일	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	가능	×	프로그램 모드/운전 모드	○
모션 설정	축 할당 축 파라메터 설정 축 그룹 파라메터 설정 MC 공통 파라메터 설정		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
캠 데이터			○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
이벤트 설정 테이블	이벤트 설정 테이블	사용자 이상 메시지	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	운전 모드/프로그램 모드	×
버스 구성	NX 유니트		○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
	CJ 버스 구성	I/O 테이블	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
고기능 I/O 유니트 설정/고기능 CPU 유니트 설정	CJ 유니트 설정	CJ 유니트 내 예) 프로토콜 매크로	○ CJ 유니트에 좌우됨	−	유지	유지	가능	×	유니트에 따라 다름	
		CPU 고기능 유니트 영역 예) Controller Link 데이터 링크 테이블	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	가능	×	운전 모드/프로그램 모드	○
		할당 DM 영역	○ (배터리에 따름)	전원 OFF 전과 동일	유지	유지	가능	○	운전 모드/프로그램 모드	○
EtherCAT 구성	EtherCAT 네트워크 구성	네트워크 구성 정보	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×

CPU 유니트 내 데이터 항목			전원 OFF 시의 유지 유무와 방법 ○:전원 OFF 유지 ×:전원 OFF 유지 안 함	전원 투입 시	상태 변경		덮어쓰기금지 방지 중의 쓰기	Sysmac Studio의 전송 기능	쓰기 가능한 동작 모드	운전 모드에서의 재기입
					프로그램 모드←→운전 모드 변경 시	전체 정지 이상 레벨의 컨트롤러 이상 발생 시		동기 기능의 대상		
EtherCAT 설정	EtherCAT 설정	마스터	○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	○	프로그램 모드	×
		슬레이브 자체 설정	○ 슬레이브에 따름	−	유지	유지	가능	○	운전 모드/프로그램 모드	○
조작 권한 인증			○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	×	프로그램 모드	×
사용자 프로그램 실행용 ID (CPU 유니트 내)			○ (비휘발성 메모리에 의함)	전원 OFF 전과 동일	유지	유지	불가능	×	프로그램 모드	×
변수의 값	비유지 변수의 값	사용자 정의 변수/디바이스 변수	×	초기값	초기값	초기값	가능	×	운전 모드/프로그램 모드	○
	유지 변수의 값	사용자 정의 변수/디바이스 변수	○ (배터리에 따름*1)	전원 OFF 전과 동일	유지	유지	가능	×	운전 모드/프로그램 모드	○
CJ 유니트용 메모리의 값	CIO/WR		×	16#0000	16#0000	16#0000	가능	×	운전 모드/프로그램 모드	○
	HR/DM/EM		○ (배터리에 따름*1)	전원 OFF 전과 동일*2	유지	유지	가능	×	운전 모드/프로그램 모드	○
이벤트 로그	이력	시스템 로그 사용자 로그	○ (배터리에 따름*1)	전원 OFF 전과 동일	유지	유지	가능	×		○
내장 시계	시스템 시계의 현재 시각		○ (배터리에 따름)	배터리가 있는 경우: 유지 (계속) 배터리가 없는 경우: 일정하지 않음(정지하는 경우도 있음)	유지 (계속)	유지 (계속)	가능	×	운전 모드/프로그램 모드	○
앱솔루트 엔코더 원점 위치 오프셋			○ (배터리에 따름*1)	전원 OFF 전과 동일	유지 (계속)	유지 (계속)	가능	×		×

*1. NX102 및 NX1P2 CPU 유니트인 경우에 비휘발성 메모리에 저장됩니다.
*2. NX102 및 NX1P2 CPU 유니트에서 [배터리레스 관련 이상 감지]를 [안 함]으로 설정한 경우에 16#0000이 됩니다.

A-9 CJ 유니트용 메모리의 세부 내용

CJ 유니트용 메모리는 변수의 AT(할당대상) 지정의 대상이 됩니다.
영역 종류별 세부 내용은 다음과 같습니다.

사용상의 주의 사항

- CJ 유니트용 메모리를 사용할 수 있는 것은 NJ 시리즈 CPU 유니트, NX102 CPU 유니트 및 NX1P2 CPU 유니트뿐입니다.
- NX1P2 CPU 유니트에서의 사용 방법은 "NX 시리즈 NX1P2 CPU 유니트 내장 I/O, 옵션 보드 기능편 사용 설명서(SBCA-449)"를 참조하십시오.

A-9-1 CIO 영역

입출력 릴레이

● **설명**

CJ 시리즈 기본 I/O 유니트의 입출력 단자를 할당할 수 있는 릴레이 영역입니다.
연결 위치에 따라(CPU 유니트로부터 가장 가까운 위치부터 좌측에서 우측으로) 각 CJ 시리즈 기본 I/O 유니트의 점유 채널 수만큼 일련 번호로 16점 단위로 할당됩니다.
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● **주소**

주소	채널 주소	비트 주소
범위	0~159CH	0.00~159.15

참고

NJ 시리즈 CPU 유니트에서는 이 입출력 릴레이 영역을 I/O 포트에 할당한 디바이스 변수에서 액세스합니다. 따라서 AT(할당대상) 지정하여 사용하는 것은 권장하지 않습니다.
일부 고기능 유니트의 할당 임의 영역의 지정 시에만 이 입출력 릴레이 영역을 AT(할당대상) 지정하십시오.

CPU 고기능 유니트 릴레이

● **설명**

CJ 시리즈 CPU 고기능 유니트의 Status 정보 등을 할당할 수 있는 릴레이 영역입니다. 1유니트당 25CH이 유니트 번호에 따라 할당됩니다.
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● 주소

주소	채널 주소	비트 주소	1유니트당
범위	1500~1899CH	1500.00~1899.15	25CH

할당 영역은 다음과 같습니다.

채널 번호	유니트 번호
1500~1524CH	0
1525~1549CH	1
≀	≀
1875~1899CH	F

이 영역의 사용 방법의 세부 내용은 각 CPU 고기능 유니트의 사용 설명서를 참조하십시오.

참고

NJ 시리즈 CPU 유니트에서는 이 CPU 고기능 유니트 릴레이 영역을 I/O 포트에 할당한 디바이스 변수에서 액세스합니다. 따라서 AT(할당대상) 지정하여 사용하는 것은 권장하지 않습니다.
일부 고기능 유니트의 할당 임의 영역의 지정 시에만 이 입출력 릴레이 영역을 AT(할당대상) 지정하십시오.

고기능 I/O 유니트 릴레이

● 설명

CJ 시리즈 I/O 고기능 유니트의 Status 정보 등을 할당할 수 있는 릴레이 영역입니다. 1유니트당 10CH, 총 96유니트(0~95호기)분이 호기 번호에 따라 할당됩니다.
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● 주소

주소	채널 주소	비트 주소	1유니트당
범위	2000~2959CH (10CH×96호기분)	2000.00~2959.15	10CH

할당 영역은 다음과 같습니다.

채널 번호	유니트 번호
2000~2009CH	0호기
2010~2019CH	1호기
≀	≀
2950~2959CH	95호기

이 영역의 사용 방법의 세부 내용은 각 고기능 I/O 유니트의 사용 설명서를 참조하십시오.

참고

NJ 시리즈 CPU 유니트에서는 이 고기능 I/O 유니트 릴레이 영역을 I/O 포트에 할당한 디바이스 변수에서 액세스합니다. 따라서 AT(할당대상) 지정하여 사용하는 것은 권장하지 않습니다.

DeviceNet 릴레이

● 설명

DeviceNet 유니트의 리모트 I/O 마스터 기능을 사용할 때, 각 슬레이브가 할당되는 릴레이 영역입니다(고정 할당 시).
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● 주소

주소	채널 주소	비트 주소
범위	3200~3799CH	3200.00~3799.15

할당 릴레이 영역의 소프트웨어 스위치 내의 고정 할당 영역 설정 1/2/3 스위치에 의해 다음 고정 할당 영역 1/2/3 중 하나를 선택할 수 있습니다.

주소	마스터→슬레이브 출력 (OUT) 영역	슬레이브→마스터 입력 (IN) 영역
고정 할당 영역 1	3200~3263CH	3300~3363CH
고정 할당 영역 2	3400~3463CH	3500~3563CH
고정 할당 영역 3	3600~3663CH	3700~3763CH

DeviceNet 유니트의 리모트 I/O 슬레이브 기능을 고정 할당하여 사용하는 경우도 이 DeviceNet 릴레이에 할당됩니다.

주소	마스터→슬레이브 출력 (OUT) 영역	슬레이브→마스터 입력 (IN) 영역
고정 할당 영역 1	3370CH	3270CH
고정 할당 영역 2	3570CH	3470CH
고정 할당 영역 3	3770CH	3670CH

자세한 내용은 🕮"CJ 시리즈 DeviceNet 유니트 사용 설명서 NJ 시리즈 연결편(SBCD-357)"을 참조하십시오.

CIO 영역의 내부 보조 릴레이

● 설명

프로그램 상에서만 사용할 수 있는 릴레이입니다. 외부 입출력 단자와의 입출력은 할 수 없습니다.
내부 보조 릴레이는 기본적으로 이 영역에서 사용하십시오.
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● 주소

주소	채널 주소	비트 주소
범위	1300~1499CH, 3800~6143CH	1300.00~1499.15, 3800.00~6143.15

A-9-2 내부 보조 릴레이

● **설명**

프로그램 상에서만 사용할 수 있는 릴레이입니다. 외부 입출력 단자와의 입출력은 할 수 없습니다. 내부 보조 릴레이는 기본적으로 이 영역에서 사용하십시오.
전원 OFF 복귀(ON→OFF→ON), 동작 모드 변경(프로그램 모드←→운전 모드) 시, 클리어됩니다.

● **주소**

주소	채널 주소	비트 주소
범위	W0~W511CH	W0.00~W511.15

A-9-3 유지 릴레이

● **설명**

프로그램 상에서만 사용할 수 있으며, 전원 OFF 복귀(ON→OFF→ON) 또는 동작 모드 변경(프로그램 모드←→운전 모드) 시에도 그 전원 OFF 직전 또는 모드 변경 직전의 ON/OFF 상태를 유지하는 릴레이입니다.

● **주소**

주소	채널 주소	비트 주소
범위	H0~H511CH	H0.00~H511.15

A-9-4 데이터 메모리

● **설명**

워드(16비트) 단위로 읽고 쓰는 범용 데이터 영역입니다.
또한, 주소에 비트 번호를 부가하면 비트 단위로의 지정(읽기/쓰기)도 가능합니다.
전원 OFF 복귀(ON→OFF→ON) 또는 동작 모드 변경(프로그램 모드←→운전 모드) 시에도 그 전원 OFF 직전 또는 모드 변경 직전의 데이터를 유지합니다.

● **주소**

주소	채널 주소	비트 주소
범위	D0~D32767	D0.00~D32767.15

고기능 유니트용 DM 영역

● 설명

데이터 메모리 영역 중에 다음 영역은 고기능 유니트의 초기 설정 데이터 등의 할당에 사용됩니다.

● 주소

주소	CJ 시리즈의 고기능 유니트 종류	채널 주소	1유니트당
범위	CJ 시리즈 고기능 I/O 유니트	D20000~29599 (100CH×96호기분)	100CH
	CJ 시리즈 CPU 고기능 유니트	D30000~31599 (100CH×16 유니트 번호분)	100CH

할당 영역은 다음과 같습니다.

CJ 시리즈 고기능 I/O 유니트

채널 번호	호기 번호
D20000~D20099	0호기
D20100~D20199	1호기
↓	↓
D29500~D29599	95호기

CJ 시리즈 CPU 고기능 유니트

채널 번호	유니트 번호
D30000~D30099	0
D30100~D30199	1
↓	↓
D31500~D31599	F

이 영역의 사용 방법의 세부 내용은 각 고기능 유니트의 사용 설명서를 참조하십시오.

참고

NJ 시리즈 CPU 유니트에서는 이 고기능 유니트용 DM 영역을 I/O 포트에 할당한 디바이스 변수에서 액세스합니다. 따라서 AT(할당대상) 지정하여 사용하는 것은 권장하지 않습니다.

A-9-5 확장 데이터 메모리

● 설명

워드(16비트) 단위로 읽고 쓰는 범용 데이터 영역입니다.
또한, 주소에 비트 번호를 부가하면 비트 단위로의 지정(읽기/쓰기)도 가능합니다.
전원 OFF 복귀(ON→OFF→ON) 또는 동작 모드 변경(프로그램 모드←→운전 모드) 시에도 그 전원 OFF 직전 또는 모드 변경 직전의 데이터를 유지합니다.

● 주소

주소	채널 주소	비트 주소
범위	NJ501-□□□□: E0_0~E18_32767 NJ301-□□□□: E0_0~E3_32767 NJ101-□□□□: E0_0~E3_32767	NJ501-□□□□: E0_0.00~E18_3276.15 NJ301-□□□□: E0_0.00~E3_32767.15 NJ101-□□□□: E0_0.00~E3_32767.15

(주의) 뱅크 수는 16진수로 표현합니다.

A-10 변수의 메모리 확보 방법

변수의 메모리 확보 방법은 구조체 형식이나 공용체 형식 변수 멤버의 메모리 상 위치를 다른 기기와 맞추는 경우에 의식해야 합니다. 주로 구조체 형식이나 공용체 형식 변수를 사용하여 다른 기기와 다음과 같은 통신을 실시하는 경우에 조정해야 합니다.

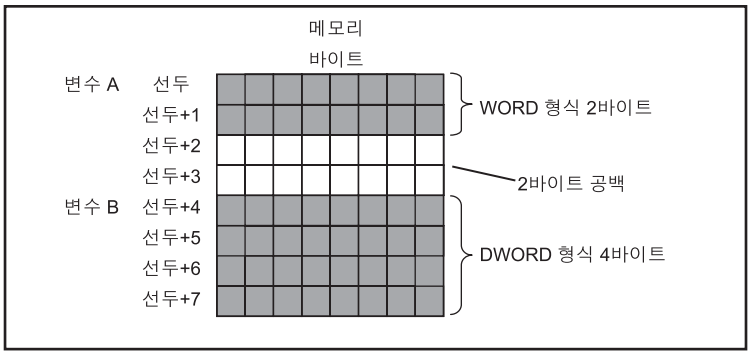
- NJ/NX 시리즈 CPU 유니트와 다른 CPU 유니트 간에 EtherNet/IP의 태그 데이터 링크 또는 CIP 메시지에 의한 변수 액세스를 실시할 때
- ID 태그 등의 CPU 유니트 이외의 기기와 구조체 형식이나 공용체 형식 변수에서 데이터 변환을 실시할 때

A-10-1 변수의 메모리 확보 규칙

변수를 위해 확보되는 메모리 양과 메모리 상의 위치는 그 변수의 데이터 형식에 따라 달라집니다. 또한, 배열의 요소, 구조체의 멤버, 공용체의 멤버를 위해 확보되는 메모리 양과 메모리 상의 위치는 각 데이터 형식에 더해 배열, 구조체, 공용체의 선언 내용에 따라 달라집니다.

데이터 형식 얼라인먼트와 확보되는 메모리 양

데이터 형식은 각각의 데이터 크기가 정해져 있습니다. 데이터 크기란, 그 데이터 형식의 값을 저장하기 위해 최소한으로 필요한 메모리 양을 의미합니다.
한편, 변수에 대응하는 메모리는 가장 효율적으로 액세스할 수 있도록 컨트롤러에 의해 자동적으로 배치됩니다. 따라서 변수를 위해 확보되는 메모리의 총량과 그 변수들의 데이터 크기의 합계가 반드시 일치하지는 않습니다. 예를 들어, WORD 형식과 DWORD 형식 변수를 선언한 경우에 데이터 크기의 합계는 6바이트이지만, 메모리는 다음 그림과 같이 8바이트 확보됩니다.



변수 테이블

이름	데이터 형식
A	WORD
B	DWORD

이와 같은 메모리상의 변수 배치가 결정되기 위한 정보가 얼라인먼트입니다. 얼라인먼트는 데이터 형식별로 정해져 있으며, 그 변수를 위해 확보되는 메모리 양과 메모리 상의 위치는 다음과 같습니다.

항목	사양
확보되는 메모리 양	얼라인먼트의 정수배, 단, 데이터 크기 이상
메모리 상의 위치	메모리 선두로부터 얼라인먼트의 정수배의 위치

기본 데이터 형식과 열거 형식의 얼라인먼트와 확보되는 메모리 양은 다음과 같습니다.

데이터 형식	얼라인먼트[바이트]	확보되는 메모리 양[바이트]
BOOL	2	2
BYTE, USINT, SINT	1	1
WORD, UINT, INT	2	2
DWORD, UDINT, DINT	4	4
LWORD, ULINT, LINT	8	8
REAL	4	4
LREAL	8	8
TIME, DATE, TIME_OF_DAT, DATE_ANT_TIME	8	8
STRING[N+1]*1	1	N+1
열거 형식	4	4

*1. N은 취급하는 최대 문자수를 나타냅니다. 예를 들어, 반각 영숫자의 최대 문자 수가 10인 경우에 문자열 종단의 NULL을 더한 11문자 분량의 메모리를 확보해야 합니다.

배열, 구조체 형식, 공용체 형식, 그 각 요소 또는 각 멤버도 각각 가장 효율적으로 액세스할 수 있도록 메모리 상에 배치됩니다. 배열, 구조체 형식, 공용체 형식의 얼라인먼트와 확보되는 메모리 양은 그 선언에 따라 다음과 같이 결정됩니다.

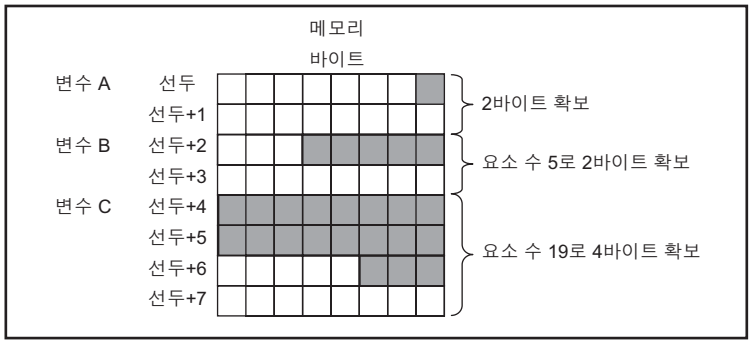
데이터 형식	얼라인먼트	확보되는 메모리 양
배열	요소의 데이터 형식의 얼라인먼트와 같음	(요소의 데이터 형식을 위해 확보되는 메모리 양)×요소 수*1
구조체 형식	모든 멤버의 얼라인먼트 중의 최대값	각 멤버를 그 멤버의 데이터 형식 얼라인먼트의 정수배인 위치에 순서대로 배치했을 때 확보되는 메모리 총량 이상이며, 얼라인먼트의 정수배
공용체 형식	모든 멤버의 얼라인먼트 중의 최대값	모든 멤버의 확보되는 메모리 양 중의 최대값

*1. BOOL 형식 배열인 경우는 예외입니다. BOOL 형식 배열을 위해 확보되는 메모리 양에 대해서는 뒤의 사용상의 주의 사항을 참조하십시오.

사용상의 주의 사항

BOOL 형식 배열을 위해 확보되는 메모리 양

BOOL 형식의 단독 변수, 구조체 멤버, 공용체 멤버를 위해 확보되는 메모리 양은 2바이트입니다. 그러나 BOOL 형식 배열은 메모리가 1요소에 2바이트씩 확보되는 것은 아닙니다. 각 요소에 대응하는 메모리가 1비트씩 순서대로 확보됩니다. 그리고 배열 전체적으로는 여분도 포함하여 2바이트의 정수배의 메모리가 확보됩니다.



변수 테이블

이름	데이터 형식
A	BOOL
B	ARRAY[1..5]OF BOOL
C	ARRAY[0..18]OF BOOL

따라서 BOOL 형식 배열을 위해 확보되는 메모리 양은 다음 식으로 구할 수 있습니다. 요소 수가 1~16인 경우는 2바이트, 17~32인 경우는 4바이트의 메모리가 확보됩니다.

$$\text{메모리량} = 2\left[\frac{\text{요소 수}-1}{16}\right]+2$$

괄호 안의 계산 결과에서 소수점 이하는 버림

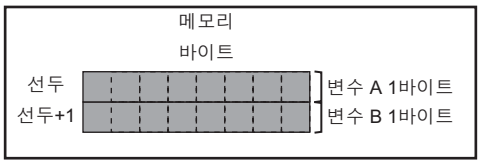
다음의 각 데이터 형식 변수의 메모리 확보 규칙을 구체적인 예를 들어 설명합니다.

기본 데이터 형식

● BYTE 형식 등의 얼라인먼트가 1바이트인 변수

1바이트 단위의 얼라인먼트에서 1바이트 메모리가 확보됩니다.

예) BYTE 형식 변수가 2개 연속된 경우



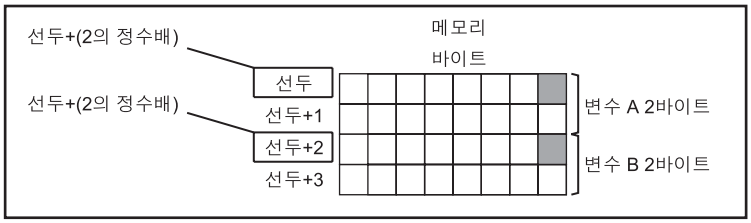
변수 테이블

이름	데이터 형식
A	BYTE
B	BYTE

● BOOL 형식, WORD 형식 등 얼라인먼트가 2바이트인 변수

2바이트 단위의 얼라인먼트에서, 2바이트 메모리가 확보됩니다.

예) BOOL 형식 변수가 2개 연속된 경우



변수 테이블

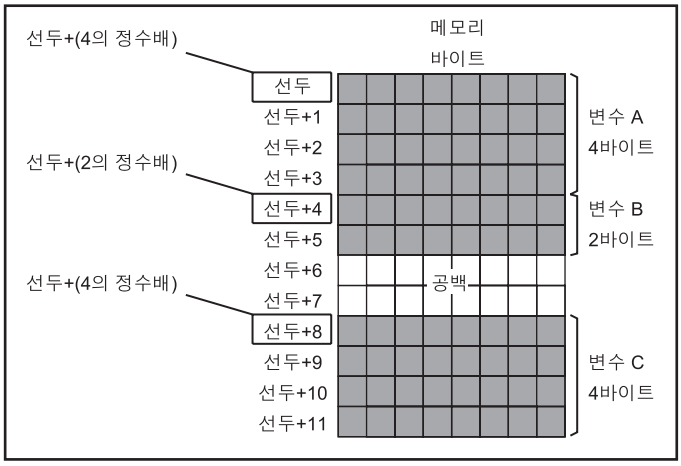
이름	데이터 형식
A	BOOL
B	BOOL

● DWORD 형식 등의 얼라인먼트가 4바이트인 변수

4바이트 단위의 얼라인먼트에서 4바이트 메모리가 확보됩니다.

데이터가 메모리에 배치되는 선두 위치가 4바이트의 정수배가 되므로 WORD 형식 등의 얼라인먼트가 2바이트인 변수가 삽입되면, 2바이트만큼 여분이 발생합니다.

예) DWORD 형식, WORD 형식, DWORD 형식 순으로 변수가 연속된 경우

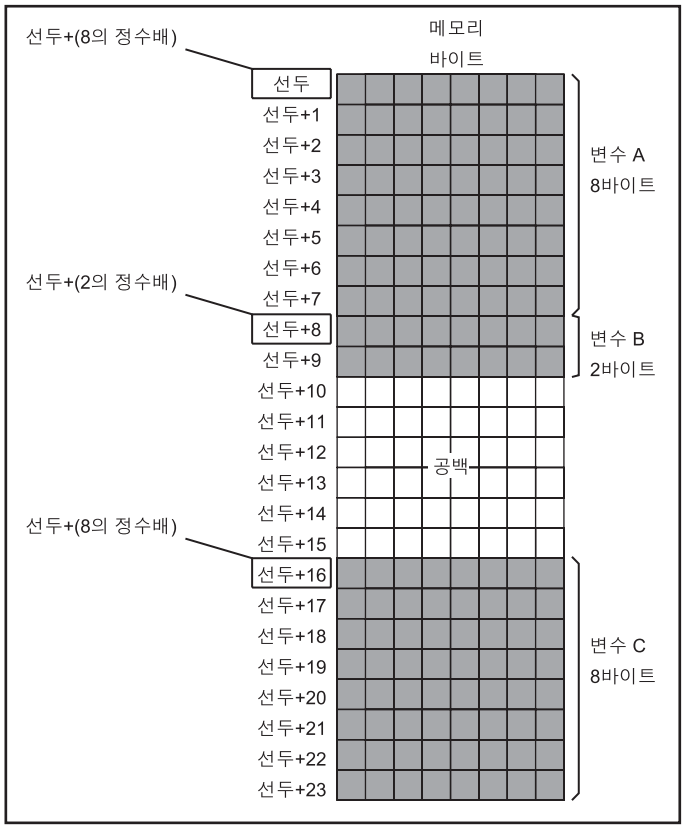


변수 테이블

이름	데이터 형식
A	DWORD
B	WORD
C	DWORD

● LWORD 형식 등의 얼라인먼트가 8바이트인 변수

8바이트 단위의 얼라인먼트에서 8바이트 메모리가 확보됩니다.
데이터가 메모리에 배치되는 선두 위치가 8바이트의 정수배가 되므로, WORD 형식 등의 얼라인먼트가 2바이트인 변수가 삽입되면 6바이트만큼, DWORD 형식 등의 얼라인먼트가 4바이트인 변수가 삽입되면 4바이트만큼 여분이 발생합니다.
예) LWORD 형식, WORD 형식, LWORD 형식 순으로 변수가 연속된 경우



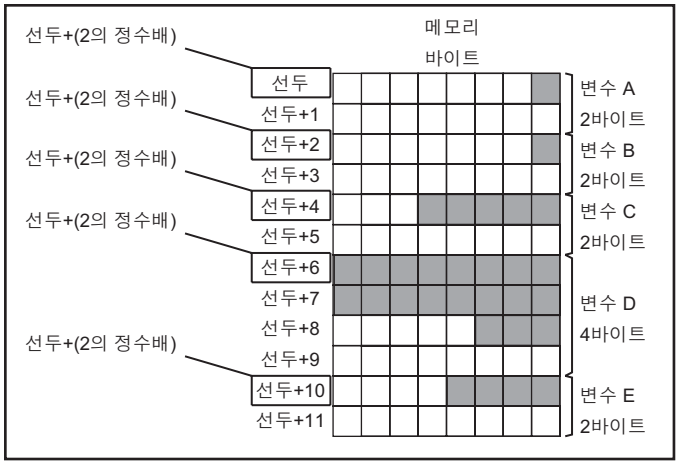
변수 테이블

이름	데이터 형식
A	LWORD
B	WORD
C	LWORD

배열

배열 변수는 배열 요소의 데이터 형식의 데이터 크기를 단위로 요소 수만큼 연속하여 메모리가 확보됩니다. 배열의 얼라인먼트는 요소의 데이터 형식의 얼라인먼트와 같습니다.

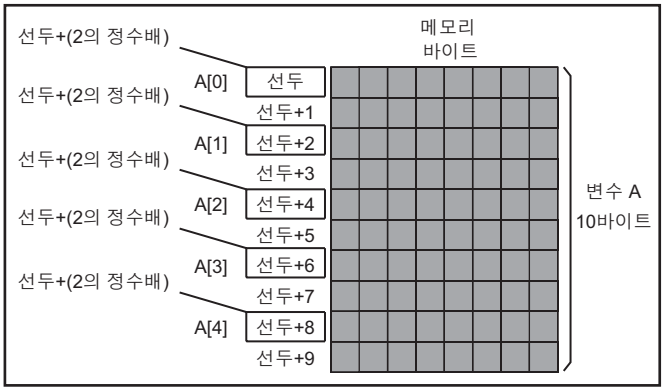
예) BOOL 형식 변수가 2개, 요소 수 5인 BOOL 형식 배열, 요소 수 19인 BOOL 형식 배열, 요소 수 4인 BOOL 형식 배열 순으로 변수가 연속된 경우



변수 테이블

이름	데이터 형식
A	BOOL
B	BOOL
C	ARRAY[1..5]OF BOOL
D	ARRAY[0..18]OF BOOL
E	ARRAY[5..8]OF BOOL

예) 요소 수 5인 INT 형식 배열의 경우



변수 테이블

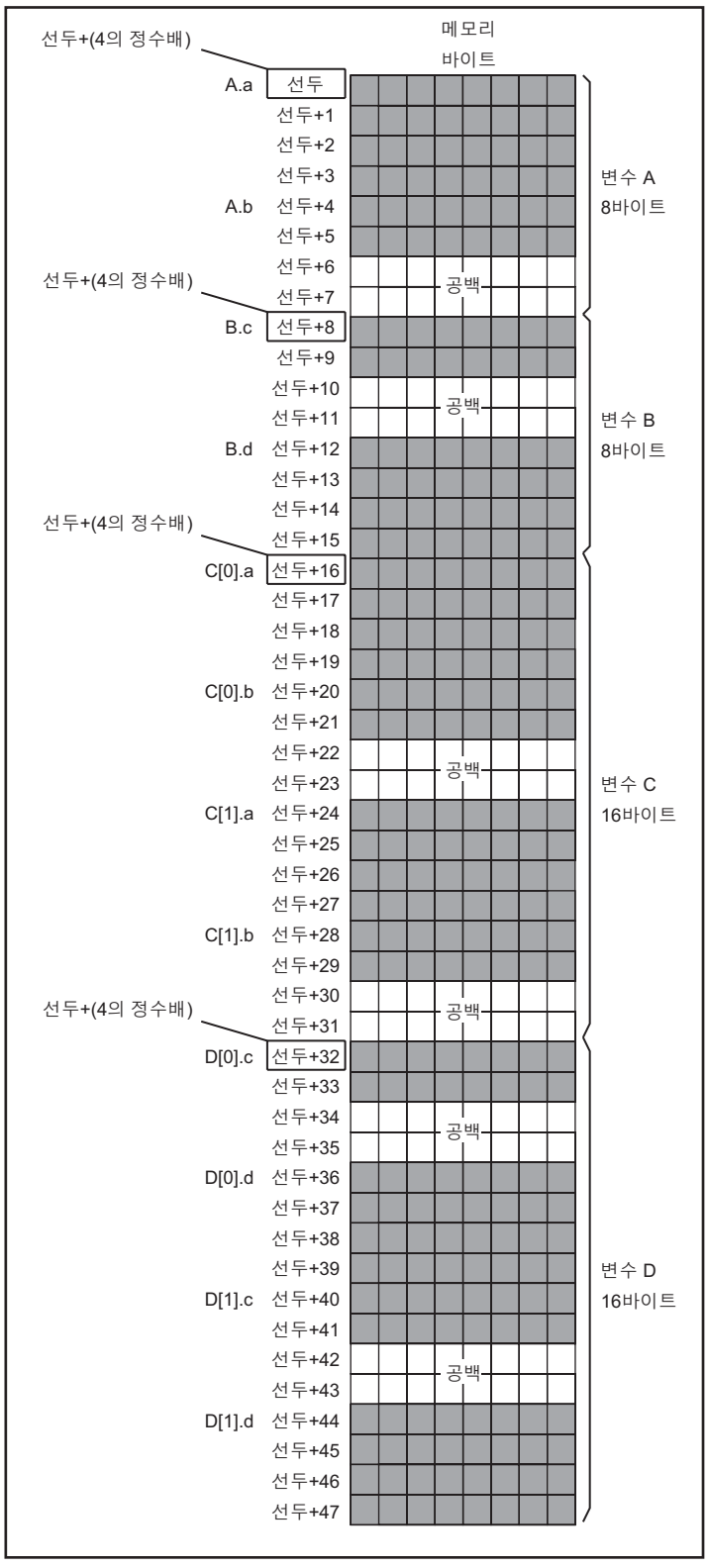
이름	데이터 형식
변수 A	ARRAY[0..4] OF INT

구조체 형식

구조체 변수는 각 멤버가 선언된 순서대로 메모리에 배치됩니다. 또한, 각 멤버가 그 멤버의 데이터 형식 얼라인먼트의 정수배의 위치에 배치됩니다. 따라서 멤버 간 또는 멤버 종단에 메모리의 여분이 삽입될 가능성이 있습니다. 구조체 형식의 얼라인먼트는 모든 멤버의 얼라인먼트 중의 최대값입니다. 확보되는 메모리 양은 각 멤버를 그 멤버 데이터 형식의 얼라인먼트의 정수배인 위치에 순서대로 배치할 때 확보되는 메모리 총량 이상이며, 얼라인먼트의 정수배입니다.

예) 다음 그림의 4개 변수 선언의 경우에 각각의 얼라인먼트와 확보되는 메모리 양은 다음과 같은 값이 됩니다.

변수	얼라인먼트[바이트]	확보되는 메모리 양[바이트]
A	4	8
B	4	8
C	4	16
D	4	16



데이터 형식 정의

이름	데이터 형식
구조체 STR_A	STRUCT
a	DINT
b	INT

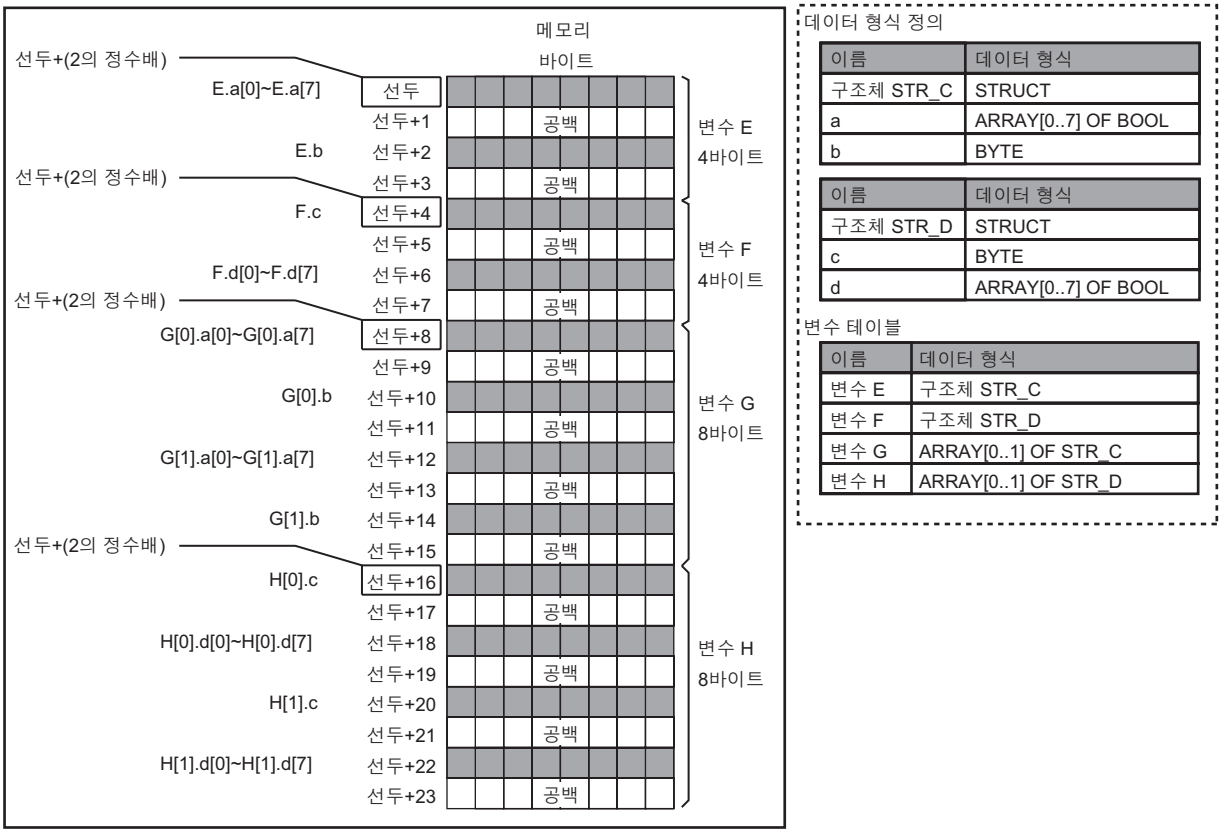
이름	데이터 형식
구조체 STR_B	STRUCT
c	INT
d	DINT

변수 테이블

이름	데이터 형식
변수 A	구조체 STR_A
변수 B	구조체 STR_B
변수 C	ARRAY[0..1] OF STR_A
변수 D	ARRAY[0..1] OF STR_B

예) 다음 그림의 4개 변수 선언의 경우에 각각의 얼라인먼트와 확보되는 메모리 양은 다음과 같은 값이 됩니다.

변수	얼라인먼트[바이트]	확보되는 메모리 양[바이트]
E	2	4
F	2	4
G	2	8
H	2	8



데이터 형식 정의

이름	데이터 형식
구조체 STR_C	STRUCT
a	ARRAY[0..7] OF BOOL
b	BYTE

이름	데이터 형식
구조체 STR_D	STRUCT
c	BYTE
d	ARRAY[0..7] OF BOOL

변수 테이블

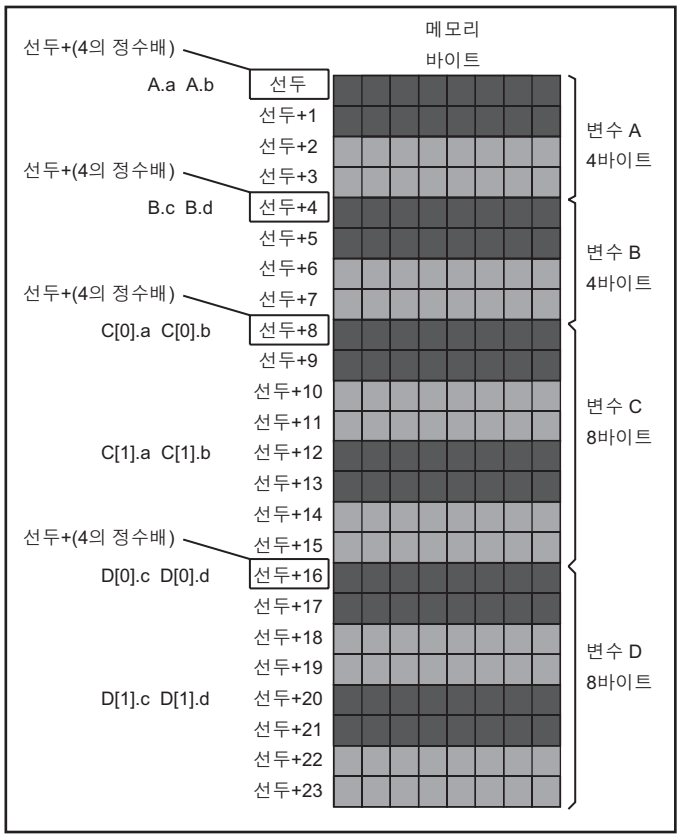
이름	데이터 형식
변수 E	구조체 STR_C
변수 F	구조체 STR_D
변수 G	ARRAY[0..1] OF STR_C
변수 H	ARRAY[0..1] OF STR_D

공용체 형식

공용체 형식은 각 멤버가 메모리 상의 같은 위치에 겹쳐 배치됩니다.
공용체 형식의 얼라인먼트는 모든 멤버의 얼라인먼트 중의 최대값입니다. 확보되는 메모리 양은 모든 멤버의 확보되는 메모리 양 중의 최대값입니다.

예) 다음 그림의 4개 변수 선언의 경우에 각각의 얼라인먼트와 확보되는 메모리 양은 다음과 같은 값이 됩니다.

변수	얼라인먼트[바이트]	확보되는 메모리 양[바이트]
A	4	4
B	4	4
C	4	8
D	4	8



데이터 형식 정의

이름	데이터 형식
공용체 UNI_A	UNION
a	DWORD
b	WORD

이름	데이터 형식
공용체 UNI_B	UNION
c	WORD
d	DWORD

변수 테이블

이름	데이터 형식
변수 A	공용체 UNI_A
변수 B	공용체 UNI_B
변수 C	ARRAY[0..1] OF UNI_A
변수 D	ARRAY[0..1] OF UNI_B

A-10-2 유의해야 하는 케이스 예

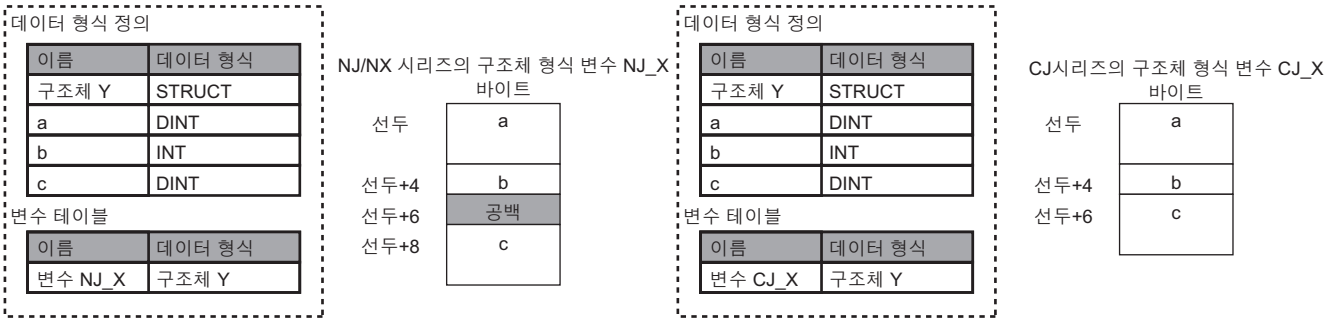
NJ/NX 시리즈 CPU 유니트의 구조체 형식 변수에서 다른 기기와의 통신 등으로 데이터를 교환할 때에는 구조체 형식 변수의 멤버의 메모리 배치를 통신하는 상대 기기와 일치시켜야 합니다.
그 경우에는 NJ/NX 시리즈 CPU 유니트 측, 또는 통신 상대 기기 측 어느 쪽에서 이 항목을 설명할 대책을 실시하십시오.

참고

NJ/NX 시리즈 CPU 유니트끼리 데이터를 교환하는 경우에는 특별히 멤버의 메모리 배치를 의식할 필요는 없습니다.

통신하는 상대 기기와 메모리 배치를 일치시키는 방법

통신하는 상대 기기와 메모리 배치를 일치시키려면 2가지 방법이 있습니다.
예를 들어, NJ/NX 시리즈 CPU 유니트와 CJ 시리즈 CPU에서는 구조체 형식 변수의 메모리 구성이 다음과 같이 다릅니다.
이 항에서는 그 메모리 배치를 일치시키는 방법을 예로 설명합니다.



- **대책 1: NJ/NX 시리즈 CPU 유니트의 구조체 형식 변수의 멤버의 메모리 배치를 변경함**

  NJ/NX 시리즈 CPU 유니트에서는 멤버의 오프셋 지정을 통해 구조체 형식 변수별로 멤버의 메모리 배치를 설정할 수 있습니다. NJ/NX 시리즈 CPU 유니트 측에서 구조체 형식 변수 멤버의 메모리 배치를 통신하는 상대 기기에 맞춰 변경하고, 조정합니다.
  구조체 형식 변수 멤버의 오프셋 지정은 구조체 형식 등록 시에 실시합니다.
  통신하는 상대 기기가 CJ 시리즈 CPU 유니트인 경우에 오프셋 종별을 'CJ'로 선택하면 자동으로 CJ 메모리 배치가 됩니다.
  또한, 오프셋 종별을 '임의'로 선택하면, 오프셋을 자유롭게 결정할 수 있습니다.

버전 관련 정보

멤버의 오프셋 지정 사용 여부는 CPU 유니트 의 유니트 버전과 Sysmac Studio의 버전 조합에 따라 다음과 같이 달라집니다.

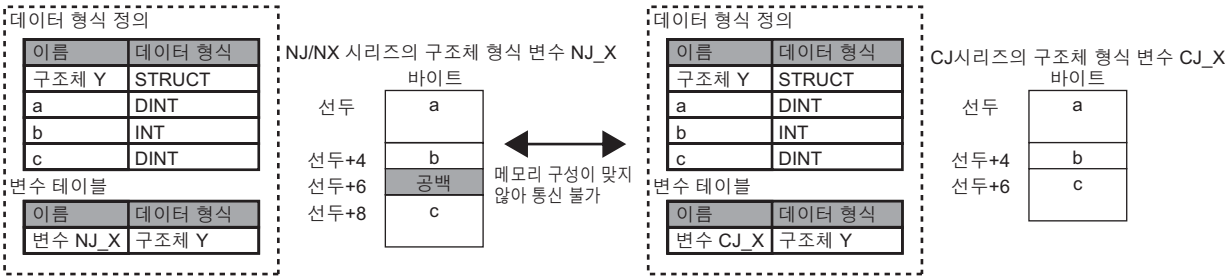
CPU 유니트의 유니트 버전	Sysmac Studio의 버전		
	Ver.1.01 이하	Ver.1.02	Ver.1.03 이상
Ver.1.01 이상	불가능	가능*1	가능
Ver.1.00	불가능	불가능	불가능

*1. 오프셋 종별의 선택 기능은 없습니다. 멤버의 오프셋은 임의로 설정합니다.

이 설정에서 구조체 형식 변수의 메모리 배치를 조정한 경우에는 네트워크 상에 있는 다른 NJ/NX 시리즈 CPU 유니트에도 마찬가지로 설정 변경을 실시하십시오.

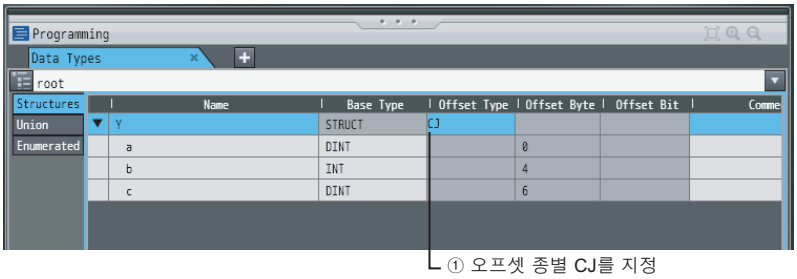
구조체 형식 변수의 메모리 배치 조작 방법에 대해서는 "Sysmac Studio Version 1 조작 설명서 (SBCA-470)"를 참조하십시오.

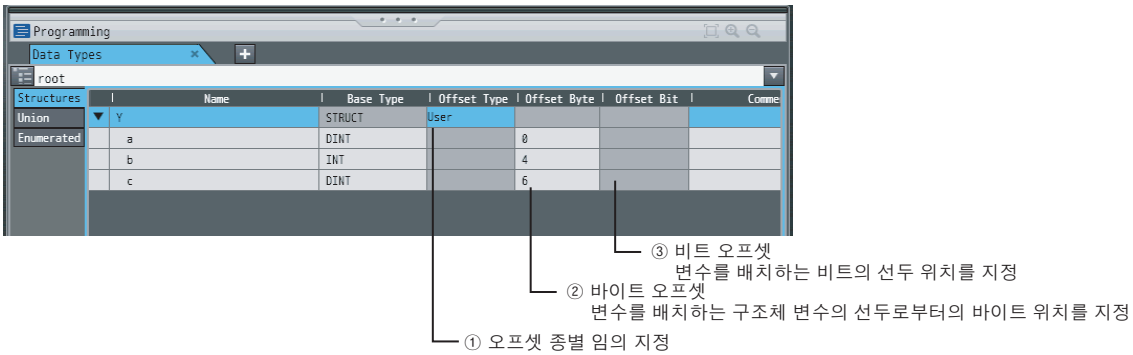
예) NJ/NX 시리즈 CPU 유니트의 구조체 형식 변수 멤버의 메모리 배치를 CJ 시리즈 CPU 유니트의 구조체 형식 변수의 멤버에 맞춰 변경함

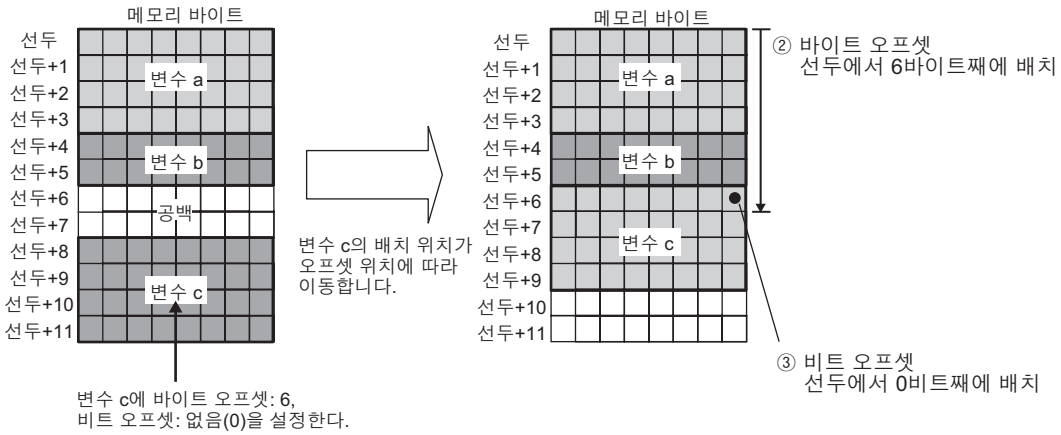


NJ/NX 시리즈 CPU 유니트와 CJ 시리즈 CPU 모듈 사이의 메모리 구성을 맞추기 위해 Sysmac Studio로 오프셋 지정을 합니다.

구조체 형식 변수 NJ_X 데이터 형식 Y의 멤버 c에 다음 오프셋 값을 지정합니다.



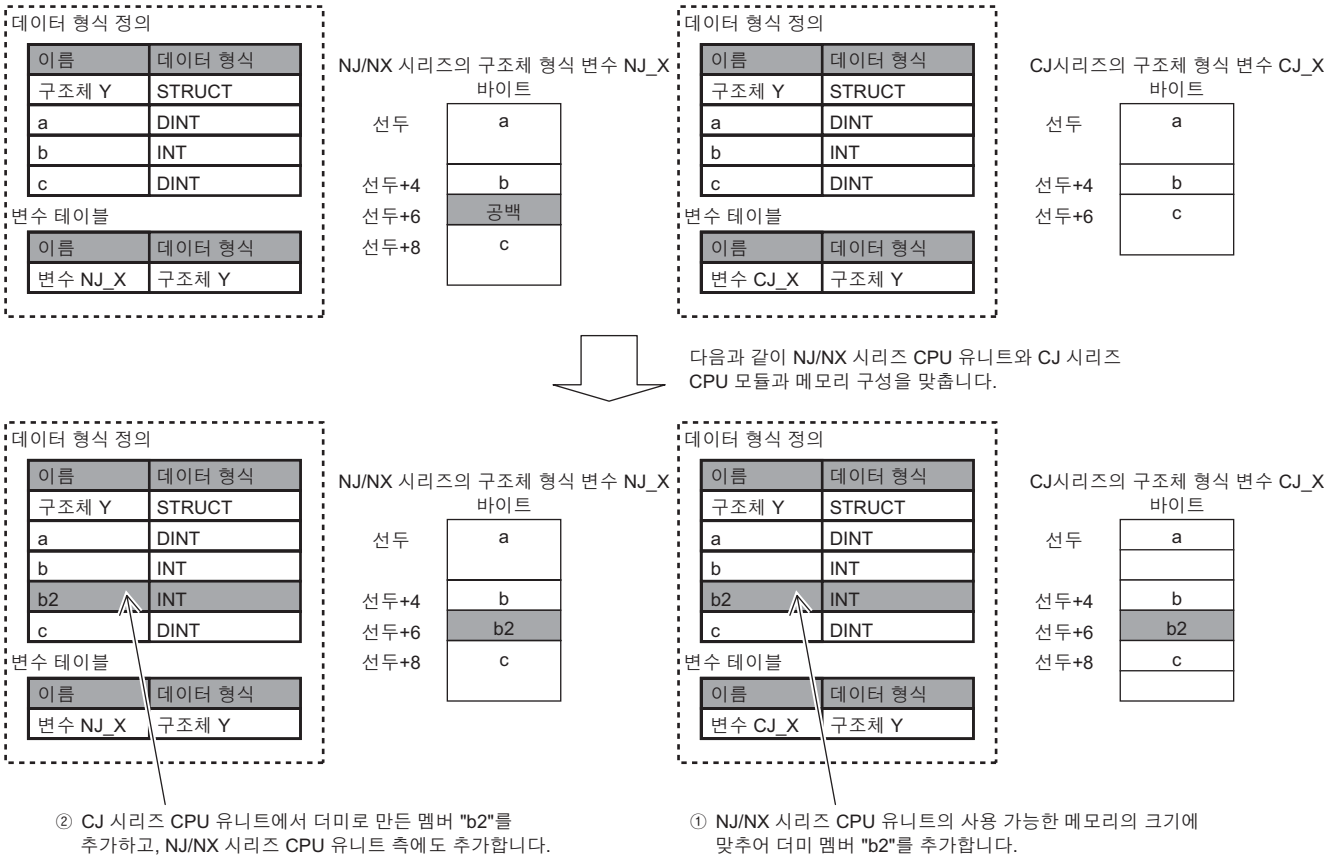




● 대책 2: 통신하는 상대 기기의 구조체 형식 변수의 메모리 배치를 변경함

NJ/NX 시리즈 CPU 유니트의 구조체 형식 변수의 메모리 배치에 맞도록 통신할 상대 기기의 구조체 형식 변수의 메모리 배치를 배치 조정용 멤버 등을 삽입하여 조정합니다.
또한, 메모리 배치 뿐만 아니라 데이터 형식도 일치시켜야 하므로 배치 조정용 멤버는 통신하는 상대 기기, NJ/NX 시리즈 CPU 유니트 모두 작성해야 합니다.

예) CJ 시리즈 CPU 유니트의 구조체 형식 변수의 메모리 배치를 NJ/NX 시리즈 CPU 유니트의 구조체 형식 변수의 메모리 배치에 맞춰 변경함



A-11 CX-Designer의 변수 테이블 등록 방법

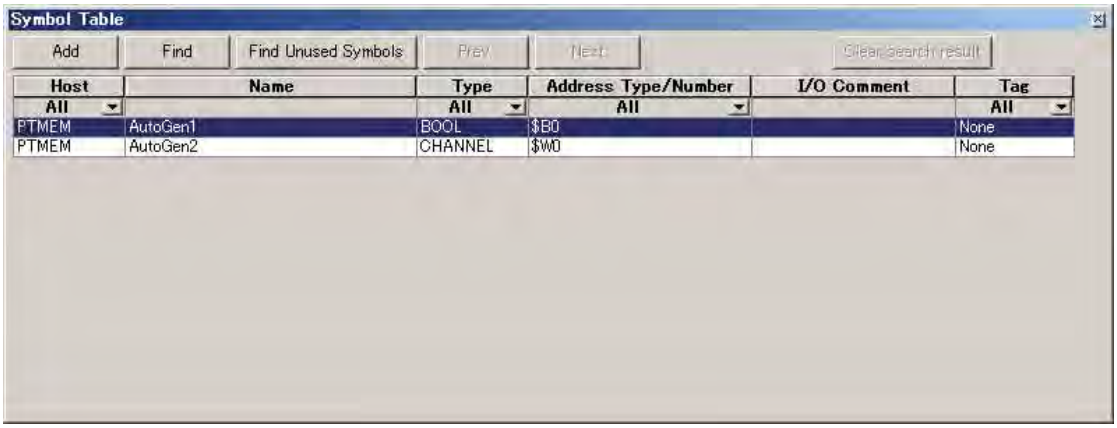
NJ/NX 시리즈 컨트롤러와 NS 시리즈 표시기를 연결하여 사용하는 경우에 CX-Designer에서는 각 기기 부품 등에 설정하는 주소로 변수를 사용할 수 있습니다. 변수는 [변수 테이블]로 관리합니다. 여기서는 Microsoft Excel에서 변수표를 복사하여 복수의 변수를 일괄적으로 변수 테이블에 등록하는 방법을 설명합니다. CX-Designer의 세부 내용은 "CX-Designer 사용 설명서(SBSA-532)"를 참조하십시오.

1 **Microsoft Excel에서 다음 서식으로 변경표를 작성합니다.**

표의 열 수나 배치를 다음 서식대로 하십시오. [주소 종류/번호] 또는 [I/O 코멘트]와 같이 공백 셀만 있는 열도 생략하지 마십시오.

호스트	이름	타입	주소 종류/번호	I/O 코멘트	태그
HOST3	_Card1BkupCmd.ExecBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.ExecVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.DirName	STRING(64)			TRUE
HOST3	_Card1BkupSta.Done	BOOL			TRUE
HOST3	_Card1BkupSta.Active	BOOL			TRUE
HOST3	_Card1BkupSta.Err	BOOL			TRUE
HOST3	_Card1VefySta.Done	BOOL			TRUE
HOST3	_Card1VefySta.Active	BOOL			TRUE
HOST3	_Card1VefySta.VefyRslt	BOOL			TRUE
HOST3	_Card1VefySta.Err	BOOL			TRUE
HOST3	_BackupBusy	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.Exec	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.DirName	STRING(64)			TRUE
HOST3	_Card1PrgTransferCmd.Password	STRING(33)			TRUE
HOST3	_Card1PrgTransferCmd.TargetUserProgram	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetIPAdr	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetVariable	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetMemory	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Done	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Active	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Err	BOOL			TRUE
HOST3	_Card1RestoreCmd.Exec	BOOL			TRUE
HOST3	_Card1RestoreCmd.DirName	STRING(64)			TRUE
HOST3	_Card1RestoreCmd.Password	STRING(33)			TRUE
HOST3	_Card1RestoreSta.Done	BOOL			TRUE
HOST3	_Card1RestoreSta.Active	BOOL			TRUE
HOST3	_Card1RestoreSta.Err	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetUserProgram	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetIPAdr	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetVariable	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetMemory	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetUnitConfig	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetAbsEncoder	BOOL			TRUE

2 CX-Designer를 기동하여 변수 테이블 화면을 엽니다.

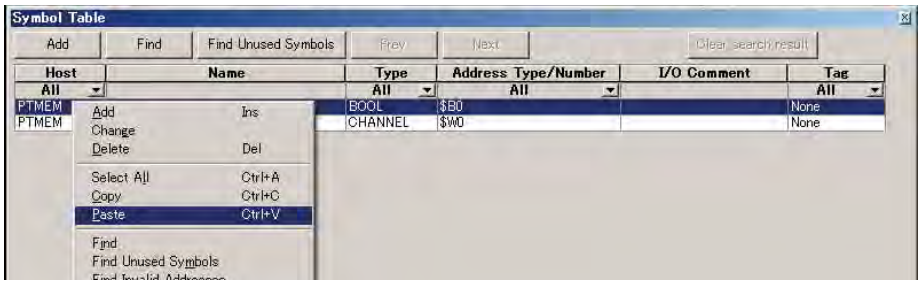


3 Microsoft Excel 변수표의 회색 부분을 복사합니다.

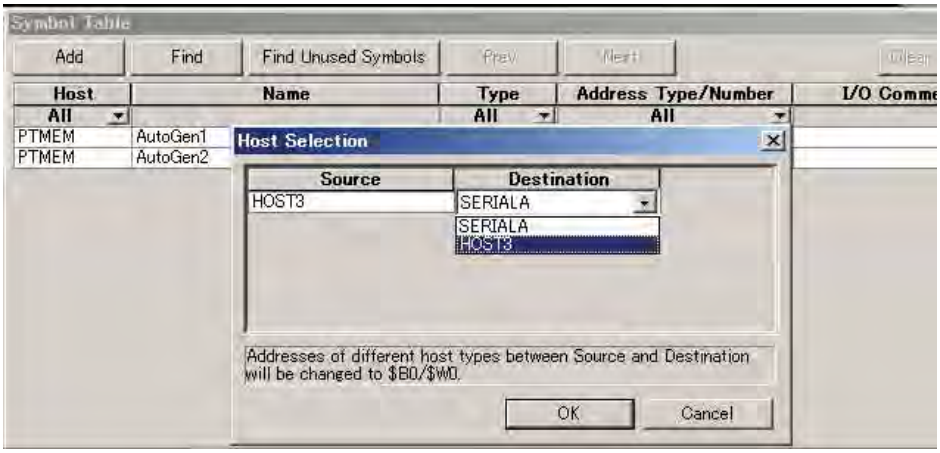
반드시 모든 열을 복사하십시오.

호스트	이름	타입	주소 종류/번호	I/O 코멘트	태그
HOST3	_Card1BkupCmd.ExecBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.ExecVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.DirName	STRING(64)			TRUE
HOST3	_Card1BkupSta.Done	BOOL			TRUE
HOST3	_Card1BkupSta.Active	BOOL			TRUE
HOST3	_Card1BkupSta.Err	BOOL			TRUE
HOST3	_Card1VefySta.Done	BOOL			TRUE
HOST3	_Card1VefySta.Active	BOOL			TRUE
HOST3	_Card1VefySta.VefyRslt	BOOL			TRUE
HOST3	_Card1VefySta.Err	BOOL			TRUE
HOST3	_BackupBusy	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.Exec	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.DirName	STRING(64)			TRUE
HOST3	_Card1PrgTransferCmd.Password	STRING(33)			TRUE
HOST3	_Card1PrgTransferCmd.TargetUserProgram	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetIPAdr	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetVariable	BOOL			TRUE
HOST3	_Card1PrgTransferCmd.TargetMemory	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Done	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Active	BOOL			TRUE
HOST3	_Card1PrgTransferSta.Err	BOOL			TRUE
HOST3	_Card1RestoreCmd.Exec	BOOL			TRUE
HOST3	_Card1RestoreCmd.DirName	STRING(64)			TRUE
HOST3	_Card1RestoreCmd.Password	STRING(33)			TRUE
HOST3	_Card1RestoreSta.Done	BOOL			TRUE
HOST3	_Card1RestoreSta.Active	BOOL			TRUE
HOST3	_Card1RestoreSta.Err	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetUserProgram	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetIPAdr	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetVariable	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetMemory	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetUnitConfig	BOOL			TRUE
HOST3	_Card1RestoreCmdTargetAbsEncoder	BOOL			TRUE

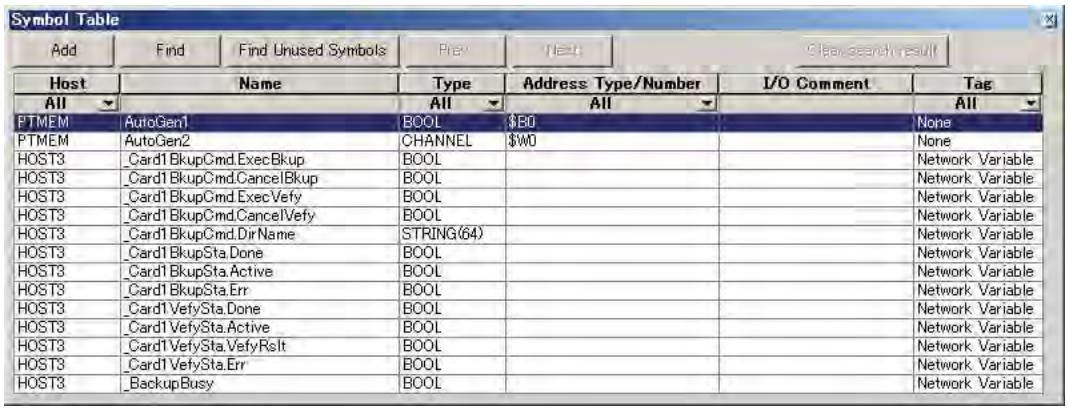
4 **CX-Designer의 변수 테이블 화면에서 마우스 오른쪽 버튼을 클릭하고, [붙여넣기]를 실행합니다.**



5 **CX-Designer의 호스트 선택 화면에서 NJ/NX 시리즈 호스트를 선택하고 [OK] 버튼을 누릅니다.**



이렇게 하면 CX-Designer의 변수 테이블 화면에 변수가 등록됩니다.



A-12 EtherCAT 슬레이브 및 축 유효/무효 전환

EtherCAT 슬레이브 및 축 유효/무효 전환을 명령을 사용하여 실행할 수 있습니다. 이 기능에 의해 다음과 같은 어플리케이션에 대응할 수 있습니다.

- EtherCAT 슬레이브 구성이나 축 구성이 다른 복수의 장치를 Sysmac Studio의 하나의 프로젝트로 관리할 수 있습니다.
- 어느 한쪽의 생산 라인을 가동시킨 채로 다른 쪽 생산 라인의 EtherCAT 슬레이브 구성이나 축 구성을 변경할 수 있습니다.

여기서는 이 기능에 사용하는 명령, 시스템 정의 변수 및 어플리케이션 예를 설명합니다.

버전 관련 정보

EtherCAT 슬레이브 및 축 유효/무효 전환 명령에 의한 실행은 CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상 조합으로 가능합니다.

A-12-1 EtherCAT 슬레이브 및 축을 사용하는 경우의 프로젝트 설정

전원 투입 시 또는 프로젝트를 다운로드 할 때에 EtherCAT 네트워크에 설치되지 않았을 가능성이 있는 EtherCAT 슬레이브는 사전에 미리 무효 설정을 해두십시오. 마찬가지로 축 사용도 미사용 축으로 설정해두십시오. EtherCAT 네트워크에 설치되지 않은 EtherCAT 슬레이브가 유효 설정되었거나 축 사용이 사용 축으로 설정되어 있으면 운전 실행 시에 이상이 발생할 가능성이 있습니다.

참고

- EtherCAT 슬레이브의 유효/무효 전환을 Sysmac Studio의 [구성·설정]-[EtherCAT]-[네트워크 구성]-[유효/무효 설정]의 설정으로 실시할 수도 있습니다. 단, 그 방법을 사용할 시에는 뒤에 설명하는 '어플리케이션 1 EtherCAT 슬레이브 구성이나 축 구성이 다른 장치의 일괄 관리(A-205페이지)'의 경우에 장치에 따라 Sysmac Studio의 설정을 매번 수동으로 변경하거나, 프로젝트 파일을 변경해야 합니다.
- 무효 설정한 EtherCAT 슬레이브는 EtherCAT 네트워크에서 분리하거나 설치할 수 있게 됩니다.

A-12-2 EtherCAT 슬레이브 및 축 유효/무효 전환을 실행하는 명령

EtherCAT 슬레이브 및 축 유효/무효 전환은 사용자 프로그램 내의 명령을 사용하여 실행합니다.
EtherCAT 슬레이브의 유효/무효 전환과 축 유효/무효 전환은 서로 다른 명령을 사용합니다. 각각의 명령명은 다음과 같습니다.

전환 대상	명령명
EtherCAT 슬레이브	EC_ChangeEnableSetting(EtherCAT 슬레이브 유효/무효 전환) 명령
축	MC_ChangeAxisUse(축 사용 변경) 명령

버전 관련 정보

EC_ChangeEnableSetting 명령, MC_ChangeAxisUse 명령은 둘 다 CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합에서만 사용할 수 있습니다.

EC_ChangeEnableSetting 명령

EtherCAT 슬레이브의 유효/무효 전환에는 EC_ChangeEnableSetting(EtherCAT 슬레이브 유효 무효 전환) 명령을 사용합니다. EC_ChangeEnableSetting 명령은 지정한 노드 주소의 EtherCAT 슬레이브를 유효로 하거나 무효로 할 수 있는 명령입니다. 이 명령을 실행한 후에 컨트롤러의 전원을 재투입하면, 명령 실행 전의 설정으로 되돌아갑니다.

EC_ChangeEnableSetting 명령의 사양에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편 (SBCA-360)"을 참조하십시오.

MC_ChangeAxisUse 명령

축의 유효/무효 전환에는 MC_ChangeAxisUse(축 사용 변경) 명령을 사용합니다. MC_ChangeAxisUse 명령은 지정한 축 파라메터의 [축 사용]을 사용 축으로 하거나 미사용 축으로 할 수 있는 명령입니다. 이 명령을 실행한 후에 컨트롤러의 전원을 재투입하면, 명령 실행 전의 설정으로 되돌아갑니다.

MC_ChangeAxisUse 명령의 사양에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 모션편 (SBCE-364)"을 참조하십시오.

A-12-3 EtherCAT 슬레이브 및 축의 상태를 나타내는 시스템 정의 변수

EtherCAT 슬레이브 및 축의 현재 상태는 시스템 정의 변수의 값을 참조하면 알 수 있습니다. 각각의 시스템 정의 변수는 다음과 같습니다.

상태를 참조하는 대상	시스템 정의 변수명
EtherCAT 슬레이브	_EC_DisableSlavTbl[ ](무효 설정 슬레이브 테이블)
축	_MC_AX[].Cfg.AxEnable(축 사용)*1

*1. NX701 CPU 유니트에서는 _MC1_AX[].Cfg.AxEnable, _MC2_AX[].Cfg.AxEnable도 사용할 수 있습니다.

_EC_DisableSlavTbl[](무효 설정 슬레이브 테이블)

_EC_DisableSlavTbl[](무효 설정 슬레이브 테이블)은 EtherCAT 슬레이브의 현재 상태가 무효인지 여부를 나타내는 시스템 정의 변수입니다. 배열의 첨자에는 노드 주소를 지정합니다. _EC_DisableSlavTbl[ ](무효 설정 슬레이브 테이블) 값의 의미는 다음과 같습니다.

값	의미
TRUE	지정한 노드 주소의 EtherCAT 슬레이브가 무효
FALSE	지정한 노드 주소의 EtherCAT 슬레이브가 유효

_MC_AX[].Cfg.AxEnable(축 사용)

_MC_AX[].Cfg.AxEnable(축 사용)은 축의 미작성 축 여부 및 축의 사용 여부를 나타내는 시스템 정의 변수입니다. 배열의 첨자에는 축 번호를 지정합니다. _MC_AX[].Cfg.AxEnable(축 사용) 값의 의미는 다음과 같습니다.

값	의미
0:_mcNoneAxis	지정한 축이 미작성 축
1:_mcUnusedAxis	지정한 축이 미사용 축
2:_mcUsedAxis	지정한 축이 사용 축

A-12-4 프로그램의 실행/정지 명령

유효/무효 전환을 실시하는 EtherCAT 슬레이브 및 축에는 특정 프로그램이 부속된 경우가 있습니다. 그 경우에 EtherCAT 슬레이브 및 축 유효/무효 전환에 따라 부속되는 프로그램을 실행시키거나 정지시켜야 합니다. 특정 프로그램의 실행/정지 명령은 사용자 프로그램 내에서 다음과 같은 명령을 사용하여 실시할 수 있습니다.

기능	명령명
프로그램 실행 명령	PrgStart 명령
프로그램 정지 명령	PrgStop 명령

PrgStart 명령, PrgStop 명령의 사양에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편 (SBCA-360K 이상)"을 참조하십시오.

사용상의 주의 사항

프로그램을 정지시키는 경우에는 그 프로그램이 부속되는 EtherCAT 슬레이브 또는 축의 무효화를 프로그램 정지보다 먼저 실행하십시오.

버전 관련 정보

PrgStart 명령, PrgStop 명령은 둘 다 CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

A-12-5 프로그램의 실행/정지 확인

유효/무효 전환을 실시하는 EtherCAT 슬레이브 및 축에 부속된 프로그램에 대해 실행이 명령되었는지 정지가 명령되었는지를 PrgStatus 명령을 사용하여 확인할 수 있습니다. PrgStatus 명령의 사양에 대한 세부 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360K 이상)"을 참조하십시오.

버전 관련 정보

PrgStatus 명령은 CPU 유니트 Ver.1.08 이상과 Sysmac Studio Ver.1.09 이상의 조합에서 사용할 수 있습니다.

A-12-6 Sysmac Studio에서의 설정

EtherCAT 슬레이브의 유효/무효, 축의 유효/무효 및 프로그램의 운전 시작 시 실행/정지의 각 설정은 Sysmac Studio에서도 변경 가능합니다. 어플리케이션에 따라서는 EtherCAT 슬레이브, 축 및 운전 시작 시 프로그램의 상태를 Sysmac Studio에서 설정해두는 경우가 있습니다.

Sysmac Studio에서의 EtherCAT 슬레이브의 유효/무효 설정

Sysmac Studio에서 EtherCAT 슬레이브를 유효로 하려면 다음 순서에 따라 설정하십시오.

1. **[구성·설정]-[EtherCAT]을 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [편집]을 선택합니다.**
   EtherCAT 구성 편집 화면이 표시됩니다.
2. **툴박스에서 연결하는 EtherCAT 슬레이브를 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [삽입]을 선택합니다.**
   EtherCAT 구성 편집 화면에서 EtherCAT 마스터의 아래에 선택한 EtherCAT 슬레이브가 표시됩니다. 또한, EtherCAT 구성 편집 화면의 우측에 EtherCAT 슬레이브의 파라메터 설정 화면이 표시됩니다.
3. **EtherCAT 슬레이브의 파라메터 설정 화면에서 [유효/무효 설정]의 설정값을 '유효'로 합니다.**

Sysmac Studio에서의 축 유효/무효 설정

Sysmac Studio에서 축을 유효로 하려면 다음 순서에 따라 설정하십시오.

1. **[구성·설정]-[무션 제어 설정]-[축 설정]을 마우스 오른쪽 버튼으로 클릭하고 메뉴에서 [추가]-[축 설정]을 선택하십시오.**
   [축 설정]의 아래에 'MC_Axis000(0)' 등의 축 변수가 추가됩니다.
2. **'MC_Axis000(0)' 등의 축 변수명을 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [편집]을 선택합니다.**
   축 파라메터 설정의 [축 기본 설정] 화면이 표시됩니다.
3. **[축 사용]을 '사용 축'으로 설정합니다.**

Sysmac Studio에서의 운전 시작 시의 프로그램 기동/정지 설정

Sysmac Studio에서 운전 시작 시에 프로그램을 실행시키려면 다음 순서에 따라 설정하십시오.

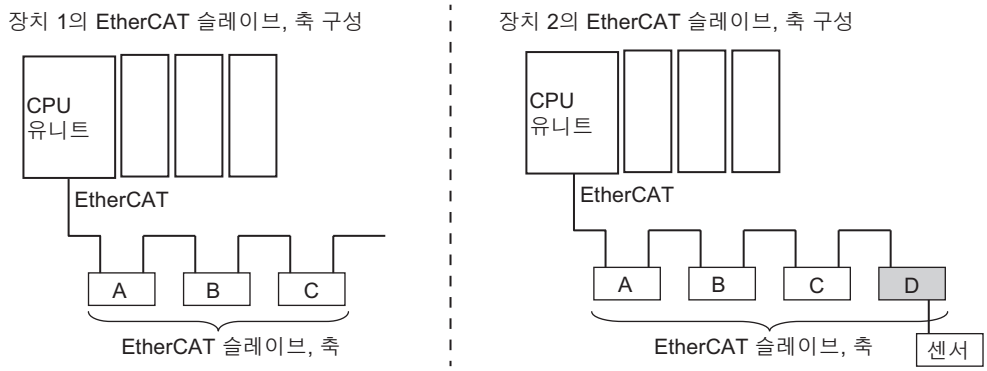
1. **[구성·설정]-[태스크 설정]을 마우스 오른쪽 버튼으로 클릭하고, 메뉴에서 [편집]을 선택합니다.**
   태스크 설정 화면이 표시됩니다.
2. **[프로그램 할당 설정] 버튼을 클릭합니다.**
   태스크 설정의 프로그램 할당 설정 화면이 표시됩니다.
3. **프로그램 할당 설정 화면에서 해당 프로그램의 [초기 상태]를 '기동'으로 합니다.**

A-12-7 EtherCAT 슬레이브 및 축 유효/무효 전환의 어플리케이션 예

EtherCAT 슬레이브 및 축 유효/무효 전환 사용 방법을 구체적인 어플리케이션을 예로 들어 설명합니다.

어플리케이션 1 EtherCAT 슬레이브 구성이나 축 구성이 다른 장치의 일괄 관리

NJ 컨트롤러의 EtherCAT 슬레이브 및 축 구성이 장치 1과 장치 2에서 다음 그림과 같이 다르다고 가정합니다. 이것을 Sysmac Studio의 하나의 프로젝트로 일괄 관리합니다.



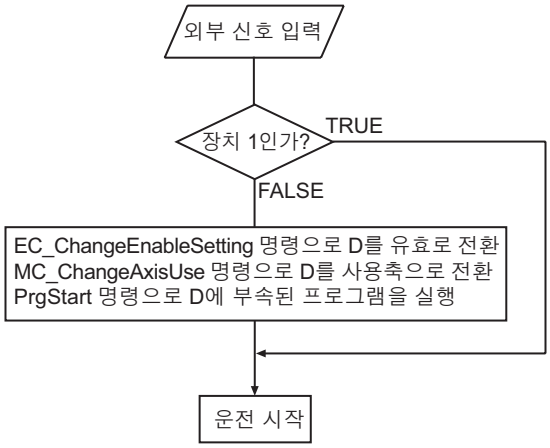
Sysmac Studio의 프로젝트에서는 그림 안의 A, B, C, D 4개의 모든 EtherCAT 슬레이브 및 축에 대해 EtherCAT 슬레이브 구성을 작성해둡니다. 그리고 각 EtherCAT 슬레이브의 유효/무효, 축 사용의 사용축/미사용 축 및 부속되는 프로그램의 운전 시작 시의 기동/정지 상태는 Sysmac Studio에서 다음과 같이 장치 1에 맞춰 설정해둡니다.

EtherCAT 슬레이브	EtherCAT 네트워크로의 설치	유효/무효	축 사용	부속되는 프로그램
A,B,C	있음	유효	사용 축	운전 시작 시, 기동
D	없음	무효	미사용 축	운전 시작 시, 정지

장치 2에 대응하는 경우는 명령에 따라 EtherCAT 슬레이브의 유효/무효, 축 사용의 사용 축/미사용 축 및 부속되는 프로그램의 실행/정지 설정을 다음과 같이 전환합니다.

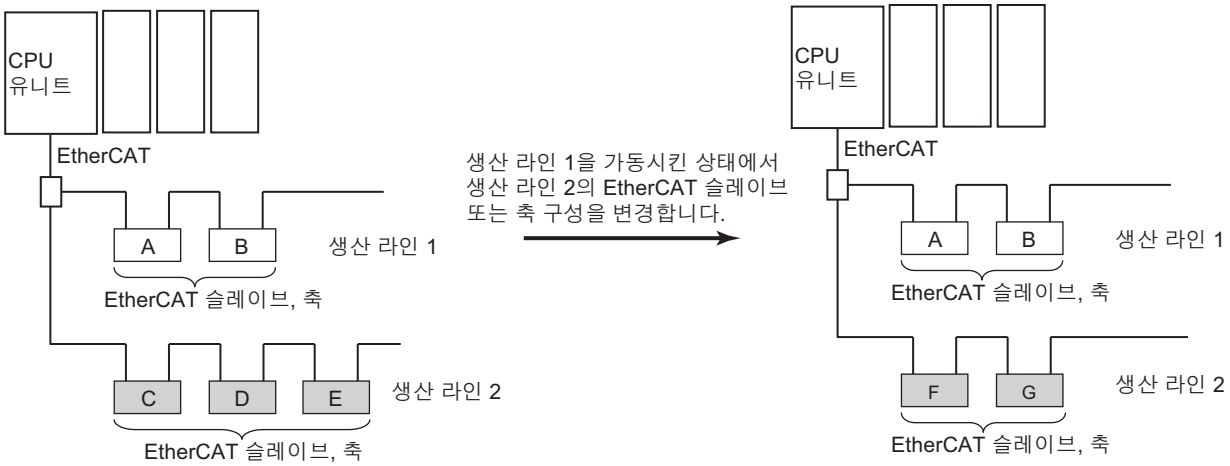
EtherCAT 슬레이브	EtherCAT 네트워크로의 설치	유효/무효	축 사용	부속되는 프로그램
A,B,C	있음	유효	사용 축	실행
D	있음	유효	사용 축	실행

사용자 프로그램의 알고리즘은 다음 그림과 같이 합니다. 컨트롤러 외부에서 운전하는 장치가 장치 1인지 장치 2인지를 나타내는 신호가 입력된다고 가정합니다.



어플리케이션 2 EtherCAT 슬레이브 구성이나 축 구성의 가동 중 변경

다음 그림의 생산 라인 1을 가동시킨 상태에서 생산 라인 2의 EtherCAT 슬레이브 및 축 구성을 변경합니다.



Sysmac Studio의 프로젝트에서는 그림 안의 A~G 7개의 모든 EtherCAT 슬레이브 및 축에 대해 EtherCAT 슬레이브 구성을 작성해둡니다.
구성 변경 전의 A~G의 EtherCAT 슬레이브의 유효/무효, 축 사용의 사용 축/미사용 축 및 부속되는 프로그램의 운전 시작 시의 기동/정지 상태는 Sysmac Studio를 사용하여 다음과 같이 설정해둡니다.

EtherCAT 슬레이브	EtherCAT 네트워크로의 설치	유효/무효	축 사용	부속되는 프로그램
A, B	있음	유효	사용 축	운전 시작 시, 기동
C,E,D	있음	유효	사용 축	운전 시작 시, 기동
F, G	없음	무효	미사용 축	운전 시작 시, 정지

사용하는 EtherCAT 슬레이브 및 축을 C, D, E에서 F, G로 변경하는 순서는 다음과 같습니다.

1. **생산 라인 2를 정지시킵니다.**
2. **MC_ChangeAxisUse 명령을 사용하여 C, D, E의 [축 사용]을 미사용 축으로 합니다.**
3. **EC_ChangeEnableSetting 명령을 사용하여 C,D,E의 EtherCAT 슬레이브 설정을 무효로 합니다.**
4. **PrgStop 명령을 사용하여 C, D, E에 부속되는 프로그램을 정지시킵니다.**
5. **EtherCAT 슬레이브 C,D,E를 생산 라인 2에서 분리합니다.**
6. **EtherCAT 슬레이브 F,G를 생산 라인 2에 설치합니다.**
7. **EC_ChangeEnableSetting 명령을 사용하여 F,G의 EtherCAT 슬레이브 설정을 무효로 합니다.**
8. **MC_ChangeAxisUse 명령을 사용하여 F, G의 [축 사용]을 사용 축으로 합니다.**
9. **PrgStart 명령을 사용하여 F, G에 부속되는 프로그램을 실행합니다.**
10. **생산 라인 2를 재가동시킵니다.**

상기 결과로 A~G의 EtherCAT 슬레이브의 유효/무효, 축 사용의 사용 축/미사용 축 및 부속되는 프로그램의 실행/정지의 설정은 다음과 같이 변합니다.

EtherCAT 슬레이브	EtherCAT 네트워크로의 설치	유효/무효	축 사용	부속되는 프로그램
A, B	있음	유효	사용 축	실행
C,E,D	없음	무효	미사용 축	정지
F, G	있음	유효	사용 축	실행

사용상의 주의 사항

프로그램을 정지시키는 경우에는 그 프로그램이 부속되는 EtherCAT 슬레이브 또는 축의 무효화를 프로그램 정지보다 먼저 실행하십시오.

A-13 사용자 프로그램의 크기에 관한 제한

사용자 프로그램에는 CPU 유니트의 메모리 용량 등의 제약에 의해 크기에 관한 제한이 있습니다. 제한을 초과하면 운전 시에 이상이 발생합니다. 여기에서는 CPU 유니트 1대에서 작성할 수 있는 사용자 프로그램의 크기에 관한 제한을 항목별로 설명합니다.
사용자 프로그램이나 변수 크기의 개략적인 값은 Sysmac Studio의 메모리 표시 기능을 사용하여 확인할 수 있습니다.
사용자 프로그램을 작성할 때에는 제한에 주의하십시오. 단, 기재된 제한값은 어디까지나 기준으로 참고만 하십시오. 추후 사용자 프로그램을 확장할 가능성도 포함하여 제한값보다 여유있게 사용자 프로그램을 작성할 것을 권장합니다.

사용상의 주의 사항

온라인 편집을 사용한 경우에 사용자 프로그램의 크기가 제한값 이하라도 이상이 발생할 수 있습니다. 이는 온라인 편집으로 사용자 프로그램을 변경하더라도 CPU 유니트 내의 메모리에 확보된 채로 남아있는 데이터가 있기 때문입니다. 이상이 발생한 경우는 컨트롤러를 프로그램 모드로 하고 사용자 프로그램을 컨트롤러에 재전송하면 이상 해제할 수 있는 경우가 있습니다.

A-13-1 사용자 프로그램 각 부품의 제한

사용자 프로그램 각 부품의 제한을 설명합니다. 제한이 있는 부품은 다음과 같습니다.

- POU
- 변수
- 데이터 형식 정의
- 상수(리터럴)

POU의 제한

POU의 제한에는 POU 정의의 제한과 POU 인스턴스의 제한이 있습니다.

● POU 정의의 제한

POU 정의에는 다음과 같은 제한이 있습니다.

제한 항목	CPU 유니트의 형식					
	NX701-□□□□	NX102-□□□□	NX1P2-□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
프로그램 수의 최대값	1,000	1,000	500	500	500	500
(FB 정의 수+FUN 정의 수+래더도 프로그램의 섹션 수)의 최대값	6,000	3,000	450	3,000	750	450
(FB 정의와 FUN 정의의 입력/출력/입출력 변수의 합계 개수)의 최대값	64	64	64	64	64	64

● POU 인스턴스의 제한

POU 인스턴스에는 다음과 같은 제한이 있습니다. CPU 유니트의 형식 및 CPU 유니트와 Sysmac Studio의 버전에 따라 POU 인스턴스 수의 최대값이 다릅니다.

NX701-□□□□의 POU 인스턴스 수의 최대값

제한 항목	CPU 유니트의 형식
	NX701-□□□□
POU 인스턴스 수의 최대값	48,000

NX102-□□□□의 POU 인스턴스 수의 최대값

제한 항목	CPU 유니트의 형식
	NX102-□□□□
POU 인스턴스 수의 최대값	9,000

NX1P2-□□□□의 POU 인스턴스 수의 최대값

제한 항목	CPU 유니트의 형식
	NX1P2-□□□□
POU 인스턴스 수의 최대값	1,800

NJ501-□□□□의 POU 인스턴스 수의 최대값

제한 항목	버전	CPU 유니트의 형식
	Sysmac Studio	NJ501-□□□□
POU 인스턴스 수의 최대값	1.05 이하	6,000
	1.06 이상	9,000

NJ301-□□□□의 POU 인스턴스 수의 최대값

제한 항목	버전		CPU 유니트의 형식
	CPU 유니트	Sysmac Studio	NJ301-□□□□
POU 인스턴스 수의 최대값	−	1.04 이하	1,500
	1.03 이하	1.05 이상	2,400
	1.04 이상		3,000

NJ101-□□□□의 POU 인스턴스 수의 최대값

제한 항목	CPU 유니트의 형식
	NJ101-□□□□
POU 인스턴스 수의 최대값	1,800

POU의 인스턴스를 세는 방법에 대한 세부 내용은 🕮'POU 인스턴스 수(A-212페이지)'를 참조하십시오.

변수의 제한

변수의 제한에는 변수 사용량의 제한과 변수 정의의 제한이 있습니다.

● 변수 사용량의 제한

변수의 사용량에는 다음과 같은 제한이 있습니다.

제한 항목	CPU 유니트의 형식					
	NX701-□□□□	NX102-□□□□	NX1P2-□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
유지 속성이 없는 변수의 합계 크기[MB]의 최대값*1	256	32	2*2	4	2	2
유지 속성이 있는 변수의 합계 크기[MB]의 최대값*1	4	1.5	0.032*2	2	0.5	0.5
유지 속성이 없는 변수 수*3 의 최대값	360,000	90,000	90,000	90,000	22,500	22,500
유지 속성이 있는 변수 수*4 의 최대값	40,000	10,000	5,000	10,000	*5	5,000
네트워크 변수 수의 최대값	40,000	40,000	27,500	40,000	27,500	27,500

*1. 변수 1개당 데이터 크기는 데이터 형식에 따라 다릅니다. 각 데이터 형식은 '6-3-5 데이터 형식(6-31페이지)'을 참조하십시오.
*2. CJ 유니트용 메모리가 포함됩니다.
*3. 세는 방법에 대한 세부 내용은 '유지 속성이 없는 변수 수(A-212페이지)'를 참조하십시오.
*4. 세는 방법에 대한 세부 내용은 '유지 속성이 있는 변수의 수(A-213페이지)'를 참조하십시오.
*5. Sysmac Studio의 버전과 CPU 유니트의 유니트 버전에 따라 제한값이 다음과 같이 달라집니다.
Sysmac Studio Ver.1.05 이상이면서 CPU 유니트 Ver.1.04 이상인 경우: 5000개
상기 이외의 조합: 2500개

● 변수 정의의 제한

변수 정의에는 다음과 같은 제약이 있습니다.

제한 항목	CPU 유니트의 형식					
	NX701-□□□□	NX102-□□□□	NX1P2-□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
배열 1개당 요소 수의 최대값	65,535	65,535	65,535	65,535	65,535	65,535
배열 차원 수의 최대값	3	3	3	3	3	3
배열 1개당 크기[MB]의 최대값	8	4	1	4	2	2
배열 첨자(요소 번호) 값의 최대값	65,535	65,535	65,535	65,535	65,535	65,535
문자열 형식 변수의 크기[바이트]*1 의 최대값	1,986	1,986	1,986	1,986	1,986	1,986

*1. 종단 NULL 문자를 포함합니다. 따라서 크기가 1986바이트인 경우의 문자 수는 반각 영숫자로 1985문자입니다.

데이터 형식 정의의 제한

데이터 형식 정의에는 다음과 같은 제한이 있습니다.

제한 항목	CPU 유니트의 형식					
	NX701-□□□□	NX102-□□□□	NX1P2-□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
데이터 형식 정의 수*1 의 최대값	8,000	1,000	1,000	2,000	1,000	1,000
구조체 형식 정의 1개당 계층 수의 최대값	8	8	8	8	8	8
구조체 형식 정의 1개당 멤버 수의 최대값	2,048	2,048	2,048	2,048	2,048	2,048
구조체 형식 변수 1개당 크기[MB]의 최대값	8	4	1	4	2	2
공용체 형식 정의 1개당 멤버 수의 최대값	4	4	4	4	4	4
열거 형식 정의 1개당 열거자 수의 최대값	2,048	2,048	2,048	2,048	2,048	2,048

*1. 세는 방법에 대한 세부 내용은 '데이터 형식 정의 수(A-213페이지)'를 참조하십시오.

상수(리터럴)의 제한

상수(리터럴)에는 다음과 같은 제한이 있습니다.

제한 항목	CPU 유니트의 형식					
	NX701-□□□□	NX102-□□□□	NX1P2-□□□□	NJ501-□□□□	NJ301-□□□□	NJ101-□□□□
문자열 형식 상수(리터럴)의 크기[바이트]의 최대값	1,985	1,985	1,985	1,985	1,985	1,985

A-13-2 사용자 프로그램 각 부품의 수

POU 인스턴스, 유지 속성이 없는 변수, 유지 속성이 있는 변수 및 데이터 형식 정의를 세는 방법을 설명합니다. 단, 이곳에 기재된 내용은 어디까지나 기준입니다. CPU 유니트의 유니트 버전의 차이 등에 다라 부품을 세는 방법이 조금씩 다른 경우가 있습니다. 따라서 사용자 프로그램의 크기가 적절한지 여부는 반드시 Sysmac Studio에서 확인하십시오.

POU 인스턴스 수

POU 인스턴스는 다음과 같이 셉니다.

● POU 인스턴스의 계수 대상

POU 인스턴스의 계수 대상은 다음과 같습니다.

- 프로그램
- FB 인스턴스(사용자 작성과 명령 둘 다 포함)
- FUN(사용자 작성과 명령 둘 다 포함)

● POU 인스턴스를 셀 때의 주의 사항

POU 인스턴스를 셀 때는 다음과 같은 점에 주의하십시오.

- 동일한 FB 정의에 대한 FB 인스턴스를 n개 사용하는 경우는 n개로 셉니다.
- 동일한 FUN을 동일한 태스크 내에서 사용하는 경우에 그 FUN이 몇 개가 있더라도 1개로 셉니다.
- 동일한 FUN을 다른 태스크에서 사용하는 경우에 태스크별로 각각 1개로 셉니다.

유지 속성이 없는 변수 수

유지 속성이 없는 변수는 다음과 같이 셉니다.

● 유지 속성이 없는 변수의 계수 대상

유지 속성이 없는 변수의 계수 대상은 다음과 같습니다.

- 유지 속성이 없는 글로벌 변수
- 프로그램 및 FB 인스턴스(사용자 작성과 명령 둘 다 포함)의 유지 속성이 없는 로컬 변수

● 유지 속성이 없는 변수를 셀 때의 주의 사항

유지 속성이 없는 변수를 셀 때에 다음과 같은 점에 주의하십시오.

- 배열은 요소 수에 관계 없이 1개로 셉니다.
- FB 인스턴스는 1개로 셉니다. 단, FB 인스턴스에는 사용자 작성과 명령 둘 다 포함합니다.
- FB 인스턴스를 요소로 하는 배열은 요소 수에 관계 없이 1개로 셉니다. 단, 그 FB 중에서 사용하고 있는 유지 속성이 없는 변수 수는 배열의 요소 수만큼 셉니다.

유지 속성이 있는 변수의 수

유지 속성이 있는 변수는 다음과 같이 셉니다.

● 유지 속성이 있는 변수의 계수 대상

유지 속성이 있는 변수의 계수 대상은 다음과 같습니다.

- 유지 속성이 있는 글로벌 변수
- 프로그램 및 FB 인스턴스(사용자 작성과 명령 둘 다 포함)의 유지 속성이 있는 로컬 변수

● 유지 속성이 있는 변수를 셀 때의 주의 사항

유지 속성이 있는 변수를 셀 때에 다음과 같은 점에 주의하십시오.

- 배열은 요소 수에 관계 없이 1개로 셉니다.
- FB 인스턴스를 요소로 하는 배열은 수에 포함하지 않습니다. 단, 그 FB 중에서 사용하고 있는 유지 속성이 있는 변수의 수는 배열의 요소 수만큼 셉니다.

데이터 형식 정의 수

데이터 형식 정의는 다음과 같이 셉니다.

● 데이터 형식 정의의 계수 대상

데이터 형식 정의의 계수 대상은 다음과 같습니다.

- 사용자가 작성한 구조체 형식 정의
- 사용자가 작성한 공용체 형식 정의
- 사용자가 작성한 열거 형식 정의

A-14 Ver.1.02 이하 CPU 유니트의 교환 방법

유니트 버전 Ver.1.02 이하의 NJ 시리즈 CPU 유니트는 SD 메모리 카드 백업 기능, Sysmac Studio 컨트롤러 백업 기능에 대응하지 않습니다. 따라서 CPU 유니트 고장 또는 버전 업그레이드 등의 이유로 CPU 유니트를 교환하는 경우에 다음과 같은 작업이 필요합니다.

작업	내용
CPU 유니트 내의 데이터 업로드	교환 전 CPU 유니트 내의 데이터를 다음 3종류의 데이터로 나누어 PC에 업로드합니다. • 프로젝트 • 변수/메모리의 현재값 • 태그 데이터 링크 테이블
새로운 CPU 유니트 연결	기존 CPU 유니트를 컨트롤러에서 분리하고, 새로운 CPU 유니트를 연결합니다.
CPU 유니트로의 데이터 다운로드	PC에 저장해둔 3종류의 데이터를 교환 후의 CPU 유니트에 다운로드 합니다.

여기에서는 상기의 작업의 세부 내용을 설명합니다.

A-14-1 CPU 유니트 내의 데이터 업로드

교환 전 CPU 유니트 내의 PC에 다음 3종류의 데이터로 나누어 업로드합니다. 지원 소프트웨어는 Sysmac Studio와 Network Configurator를 사용합니다.

업로드하는 데이터	데이터의 내용	지원 소프트웨어
프로젝트	• 유니트 구성·유니트 설정 • I/O 맵 • 컨트롤러 설정(동작 설정, 내장 EtherNet/IP 포트 설정) • 모션 제어 설정 • 캠 데이터 설정 • 이벤트 설정 • 태스크 설정 • POU • 데이터 형식 정의, 글로벌 변수 정의	Sysmac Studio
변수/메모리의 현재값	• 유지 속성이 있는 변수의 값 • CJ 유니트용 메모리 안의 HR, DM, EM 영역의 값 • 앱솔루트 엔코더 원점 위치 오프셋	Sysmac Studio
태그 데이터 링크 테이블*1	• EtherNet/IP에서의 태그 데이터 링크 설정	Sysmac Studio Network Configurator*2

*1. 태그 데이터 링크 테이블의 업로드가 필요한 것은 태그 데이터 링크 설정이 되어 있는 경우 뿐입니다.
*2. Ver.1.09 이하 Sysmac Studio를 사용하는 경우에 Network Configurator를 사용합니다.

사용상의 주의 사항

프로젝트, 변수/메모리의 현재값, 태그 데이터 링크 테이블에는 CPU 유니트 내의 다음 데이터는 포함되지 않습니다. 따라서 CPU 유니트의 교환 후에 재설정이 필요합니다.

- 데이터 트레이스 설정
- 컨트롤러 이름
- 조작 권한 인증
- 내장 시계의 표준 시간대 설정

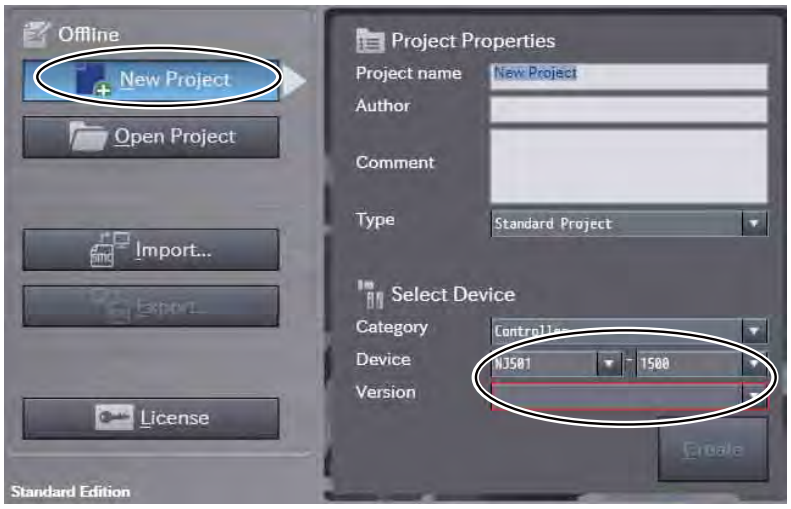
프로젝트 업로드

프로젝트 업로드의 순서는 다음과 같습니다.

1 **NJ 시리즈 컨트롤러와 연결한 PC에서 Sysmac Studio를 기동합니다.**

2 **[새 프로젝트] 버튼을 클릭하고, 새 프로젝트를 작성합니다.**

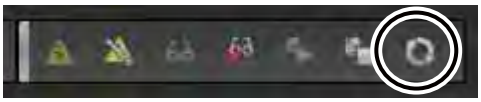
3 **교환 전 CPU 유니트의 [장치]와 [버전]을 선택합니다.**



4 **툴바의 [온라인] 버튼을 클릭합니다.**



5 **툴바의 [동기] 버튼을 클릭합니다.**



6 **[EtherNet/IP 연결 설정(내장 포트, 유니트를 전송하지 않음)]의 체크를 해제하고, [전송[컨트롤러→PC]] 버튼을 클릭합니다.**

컨트롤러 내의 프로젝트가 PC에 업로드됩니다.



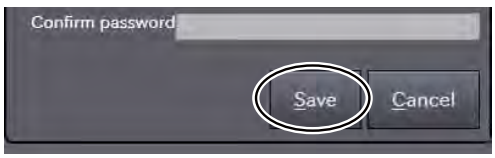
7 **[재비교] 버튼을 클릭합니다.**

업로드한 프로젝트와 컨트롤러 내의 프로젝트가 비교됩니다.



8 **[저장] 버튼을 클릭합니다.**

프로젝트가 PC에 저장됩니다.

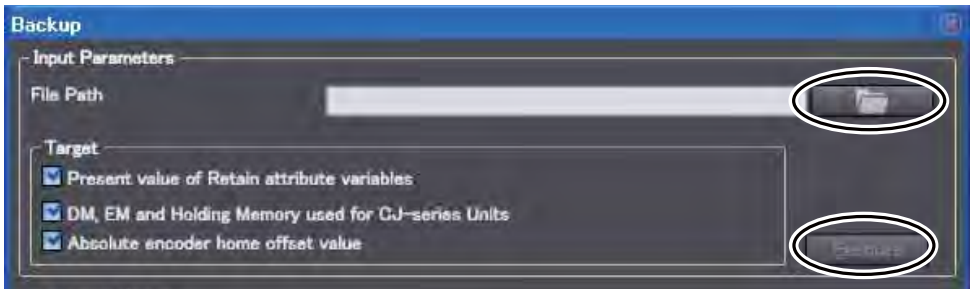


변수/메모리의 현재값 업로드

변수/메모리의 현재값 업로드는 'Sysmac Studio 변수/메모리 백업 기능'을 사용합니다. 'Sysmac Studio 변수/메모리 백업 기능'의 세부 내용은 '9-8 Sysmac Studio 변수/메모리 백업 기능(9-46페이지)'을 참조하십시오.
순서는 다음과 같습니다.

1 **Sysmac Studio 툴바의 [툴]-[백업]-[변수·메모리 백업]을 선택합니다.**

2 **대상의 '유지 속성 변수의 현재값', 'CJ 유니트용 메모리의 DM, EM, HR 영역의 현재값', '앱솔루트 엔코더 원점 위치 오프셋 데이터'를 선택하고, [실행] 버튼을 클릭합니다.**
변수/메모리의 각 데이터가 PC에 업로드됩니다.



태그 데이터 링크 테이블 업로드

Ver.1.09 이하 Sysmac Studio를 사용하는 경우에 Network Configurator를 사용합니다.
태그 데이터 링크 테이블의 업로드 순서는 다음과 같습니다.

1 **NJ 시리즈 컨트롤러와 연결한 PC에서 Network Configurator를 기동합니다.**

2 **툴바의 [네트워크]-[연결]을 선택합니다.**

3 **툴바의 [네트워크]-[업로드]를 선택합니다.**
'현재 문서에서 업로드를 시작합니다. 계속하시겠습니까? '라고 표시됩니다.

4 **[예] 버튼을 클릭합니다.**

5 **'대상 장치'는 연결된 CPU 유니트의 IP 주소만을 선택해서 [OK] 버튼을 클릭합니다.**



6 **[파일]-[다른 이름으로 저장]를 클릭합니다.**
태그 데이터 링크 테이블이 PC에 업로드됩니다.

A-14-2 새로운 CPU 유니트 연결

교환 전의 CPU 유니트를 분리하고, 새로운 CPU 유니트를 연결합니다. 연결 방법의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"을 참조하십시오.

새로운 CPU 유니트에 대해 다음을 실행하십시오.

- 교환 전의 CPU에 장착되어 있던 SD 메모리 카드를 장착함
- 딥 스위치 설정을 교환 전의 CPU와 동일하게 함

A-14-3 CPU 유니트로의 데이터 다운로드

PC에 업로드해 둔 프로젝트, 변수/메모리의 현재값 및 태그 데이터 링크 테이블을 교환 후의 CPU 유니트에 다운로드합니다.

안전상의 중요 사항

다운로드한 프로젝트는 동작을 충분히 확인한 다음 본 운전을 실시하십시오.

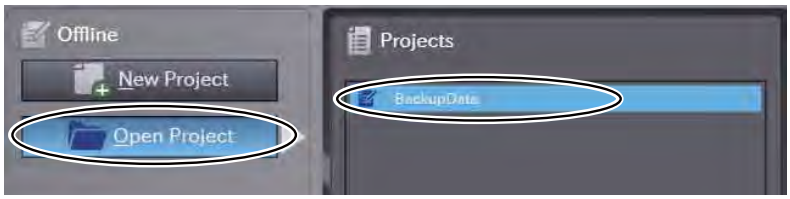
프로젝트 다운로드

프로젝트 다운로드의 순서는 다음과 같습니다.

1 **NJ 시리즈 컨트롤러와 연결한 PC에서 Sysmac Studio를 기동합니다.**

2 **[프로젝트 열기] 버튼을 클릭합니다.**

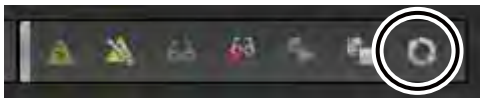
3 **프로젝트 목록 중에서 CPU 유니트 교환 전에 업로드한 프로젝트를 선택합니다.**
다음 그림은 CPU 유니트 교환 전에 업로드한 프로젝트 이름이 '백업 데이터'인 경우의 예입니다.



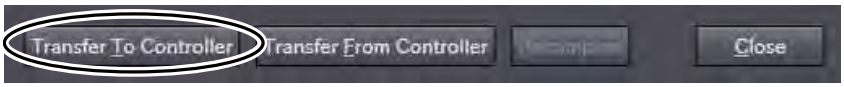
4 **툴바의 [온라인] 버튼을 클릭합니다.**



5 **툴바의 [동기] 버튼을 클릭합니다.**



6 **[EtherNet/IP 연결 설정(내장 포트, 유니트를 전송하지 않음)]의 체크를 해제하고, [전송[PC→컨트롤러]] 버튼을 클릭합니다.**
PC 내의 프로젝트가 컨트롤러에 다운로드됩니다.

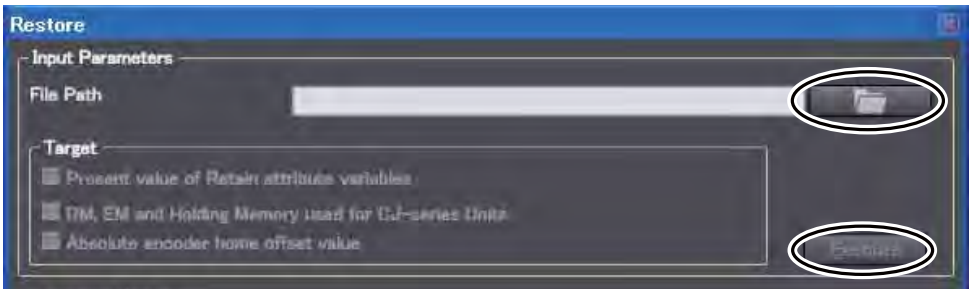


변수/메모리의 현재값 다운로드

변수/메모리의 현재값 다운로드는 'Sysmac Studio 변수/메모리 백업 기능'을 사용합니다. 'Sysmac Studio 변수/메모리 백업 기능'의 세부 내용은 '9-8 Sysmac Studio 변수/메모리 백업 기능(9-46페이지)'을 참조하십시오.
순서는 다음과 같습니다.

1 **Sysmac Studio 툴바의 [툴]-[백업]-[변수·메모리 복원]을 선택합니다.**

2 **CPU 유니트의 교환 전에 업로드한 데이터 파일을 선택하고, [실행] 버튼을 클릭합니다.**
변수/메모리의 각 데이터가 컨트롤러에 다운로드됩니다.



태그 데이터 링크 테이블 다운로드

Ver.1.09 이하 Sysmac Studio를 사용하는 경우에 Network Configurator를 사용합니다.
태그 데이터 링크 테이블의 다운로드 순서는 다음과 같습니다.

1 **NJ 시리즈 컨트롤러와 연결한 PC에서 Network Configurator를 기동합니다.**

2 **툴바의 [네트워크]-[연결]을 선택합니다.**

3 **[파일]-[열기]를 선택하고, CPU 유니트의 교환 전에 업로드한 태그 데이터 링크 테이블 파일을 엽니다.**

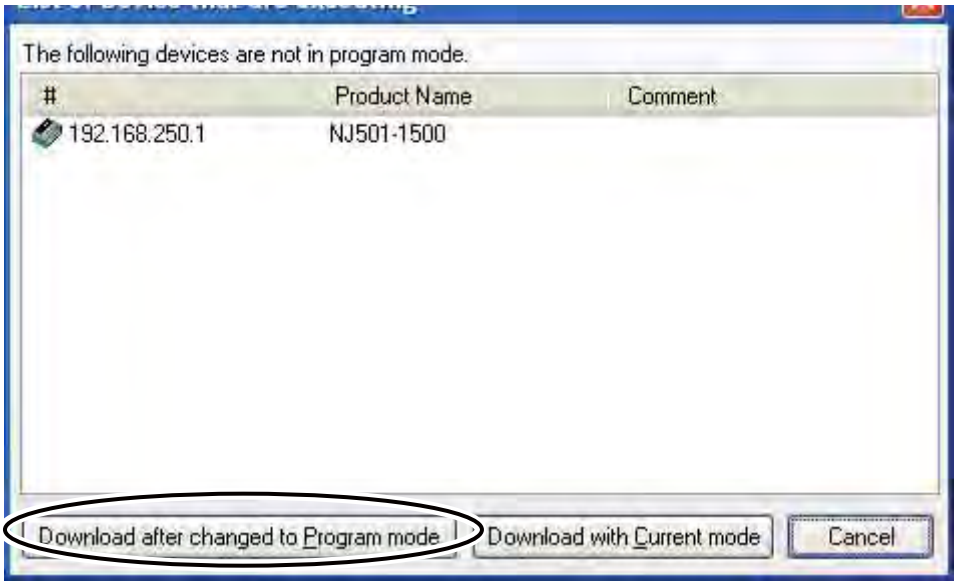
4 **유니트 버전 Ver.1.02 이하의 CPU 유니트에서 업로드한 태그 데이터 링크 테이블을 유니트 버전 Ver.1.03 이상의 CPU 유니트로 다운로드하는 경우에는 [장치]-[장치 변경]을 선택하고, '새로운 장치'를 'NJ□□□-1□□□ Rev2'로 설정합니다.**



5 **툴바의 [네트워크]-[다운로드]를 선택합니다.**
'전체 장치의 파라메터 쓰기를 실시하고, 새로운 설정을 유효로 합니다. 계속하시겠습니까? '라고 표시됩니다.

6 **[예] 버튼을 클릭합니다.**
동작 중 장치 목록이 표시됩니다.

7 **[프로그램 모드로 변경한 다운로드] 버튼을 클릭합니다.**
태그 데이터 링크 테이블이 컨트롤러에 다운로드됩니다.



사용상의 주의 사항

상기의 순서 4에서 '새로운 장치' 선택지에 'NJ□□□-1□□□ Rev2'가 표시되지 않는 경우에는 Network Configurator를 최신 버전으로 하십시오.

A-15 NX 시리즈의 버전 관련 정보

NX 시리즈 CPU 유니트와 Sysmac Studio의 버전 대응 및 유니트 버전별 지원 기능에 대해 설명합니다.

A-15-1 유니트 버전과 Sysmac Studio 버전의 관계

NX 시리즈 CPU 유니트의 유니트 버전 및 Sysmac Studio의 버전에는 대응 관계가 있습니다. 기본적으로 대응하는 버전끼리 조합하여 사용하십시오.

유니트 버전과 Sysmac Studio의 대응 버전

NX 시리즈 CPU 유니트의 유니트 버전과 Sysmac Studio의 대응 버전 표는 다음과 같습니다.

CPU 유니트의 유니트 버전	Sysmac Studio의 대응 버전
Ver.1.30*1	Ver.1.23
Ver.1.18*2	Ver.1.22
Ver.1.16*3	Ver.1.20
Ver.1.14	Ver.1.18
Ver.1.13*4	Ver.1.17*5
Ver.1.12	Ver.1.16
Ver.1.11	Ver.1.15
Ver.1.10	Ver.1.13

*1. CPU 유니트 NX102-□□□□의 유니트 버전입니다. CPU 유니트 NX102-□□□□에는 Ver.1.29 이하는 없습니다. CPU 유니트 NX701-□□□□ 및 NX1P2-□□□□□□□□에는 Ver.1.30은 없습니다.
*2. CPU 유니트 NX701-□□□□ 및 NX1P2-□□□□□□□□에는 Ver.1.17은 없습니다.
*3. CPU 유니트 NX701-□□□□ 및 NX1P2-□□□□□□□□에는 Ver.1.15는 없습니다.
*4. CPU 유니트 NX1P2-□□□□□□□□에는 Ver.1.12 이하는 없습니다.
*5. CPU 유니트 NX1P2-□□□□□□□□에는 Sysmac Studio Ver.1.17 이상을 사용하십시오. Sysmac Studio Ver.1.16 이전에서는 NX1P2-□□□□□□□□를 사용할 수 없습니다.

유니트 버전과 Sysmac Studio의 버전이 맞지 않는 경우의 사양

NX 시리즈 CPU 유니트의 유니트 버전과 Sysmac Studio의 버전이 맞지 않는 경우의 사양은 다음과 같습니다.

● Sysmac Studio 버전이 Ver.1.12 이하인 경우

Ver.1.12 이하의 Sysmac Studio에서는 NX 시리즈 CPU 유니트를 사용할 수 없습니다.

● Sysmac Studio의 버전이 낮은 경우

Sysmac Studio의 버전이 낮은 경우에 사용할 수 있는 기능은 Sysmac Studio의 버전에 대응하는 CPU 유니트의 유니트 버전이 탑재된 기능입니다.

예) CPU 유니트의 유니트 버전: Ver.1.11
Sysmac Studio 버전: Ver.1.13
Syamac Studio Ver.1.13에 대응하는 CPU 유니트의 유니트 버전은 Ver.1.10입니다. 따라서 사용할 수 있는 기능은 CPU 유니트 Ver.1.10이 탑재된 기능입니다. CPU 유니트 Ver.1.11 이상에서 추가된 기능은 사용할 수 없습니다.

● CPU 유니트의 유니트 버전이 낮은 경우

NX 시리즈CPU 유니트의 유니트 버전이 낮은 경우에는 Sysmac Studio의 프로젝트의 장치 선택 화면에서 연결된 CPU 유니트의 유니트 버전 이하로 선택하십시오. 사용할 수 있는 기능은 연결된 CPU 유니트의 유니트 버전이 탑재된 기능입니다.

예) CPU 유니트의 유니트 버전: Ver.1.10

Sysmac Studio 버전: Ver.1.15

Syamac Studio Ver.1.15에 대응하는 CPU 유니트의 유니트 버전은 Ver.1.11입니다. 그러나 연결된 CPU 유니트의 유니트 버전이 Ver.1.10이므로 프로젝트의 장치 선택 화면의 '버전'을 'Ver.1.10'으로 선택하십시오.

장치 선택 화면의 '버전'을 'Ver.1.10'으로 선택한 경우에 사용할 수 있는 기능은 CPU 유니트 Ver.1.10이 탑재 기능입니다. CPU 유니트 Ver.1.11 이상에서 추가된 기능은 사용할 수 없습니다.



참고

프로젝트의 장치에 설정하는 유니트 버전

- Sysmac Studio Ver.1.02 이상에서는 유니트 버전 선택을 장치 선택 화면에서 설정합니다.
- 설정할 수 있는 유니트 버전은 사용하는 CPU 유니트의 유니트 버전 이하입니다. 예를 들어, CPU 유니트의 유니트 버전이 Ver.1.11인 경우에는 '1.10' 또는 '1.11'을 선택하여 설정하십시오.
- Sysmac Studio에서는 프로젝트의 장치에 설정한 유니트 버전을 CPU 유니트의 유니트 버전으로 취급합니다. 예를 들어, 프로젝트의 장치에 설정하는 유니트 버전이 Ver.1.10인 경우에는 Sysmac Studio에서 유니트 버전 Ver.1.10까지의 기능을 사용할 수 있습니다.
- 프로젝트의 장치에 설정한 유니트 버전이 전송대상 CPU 유니트의 유니트 버전 이하인 경우에 Sysmac Studio에서 프로젝트를 전송할 수 있습니다.
- 장치 선택 화면의 세부 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

A-15-2 유니트 버전 갱신에 따른 지원 기능의 추가와 변경

NX 시리즈 CPU 유니트의 유니트 버전 갱신에 따라 추가 또는 변경된 지원 기능을 소개합니다.

● 기능 사양의 추가와 변경

추가 또는 변경된 기능 사양별로 CPU 유니트의 유니트 버전과 Sysmac Studio 버전을 나타냅니다.

기능				추가/변경	유니트 버전	Sysmac Studio의 버전	참조할 항목
프로그래밍 기능	변수	배열	가변 길이 배열	추가	Ver.1.18	Ver.1.22	📖'6-3-7 데이터 형식의 배열/범위 지정(6-51페이지)'
통신 기능	Ethernet/IP 포트	TCP/IP 기능	Packet Filter *1	추가	Ver.1.30	Ver.1.23	📖"NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"
		OPC UA	서버 기능*1	추가	유니트 버전과 Sysmac Studio의 대응 버전에 대해서는 "NJ/NX 시리즈 CPU 유니트 사용 설명서 OPC UA편(SBCD-374)"을 참조		
		기타	Modbus/TCP (클라이언트)*1	추가	Ver.1.30	Ver.1.23	📖"NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서 (SBCD-359)"
			FINS 통신 서비스*1	추가	Ver.1.30	Ver.1.23	📖"NX 시리즈 CPU 유니트 사용 설명서 FINS 기능편 (SBCD-375)"
	통신용 명령			변경	Ver.1.11 Ver.1.30*1	Ver.1.15 Ver.1.23	📖"NJ/NX 시리즈 명령 참조 설명서 기본편 (SBCA-360)"
SD 메모리 카드 기능	어플리케이션	SD 메모리 카드 프로그램 전송 기능		추가	Ver.1.11	Ver.1.15	📖'9-5 SD 메모리 카드 프로그램 전송 기능 (9-34페이지)'

*1. CPU 유니트 NX102-□□□□에서만 추가됐습니다.

(주의) CPU 유니트의 유니트 버전 갱신에 따른 각 기능 모듈의 기능 추가, 변경에 대해서는 각 기능 모듈의 설명서를 참조하십시오.

● 기본 명령 및 모션 제어 명령의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 사용할 수 있는 기본 명령 및 모션 제어 명령이 추가 또는 변경될 수 있습니다.

자세한 내용은 "NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)" 및 "NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

● 컨트롤러 이벤트의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 발생하는 이벤트가 추가 또는 변경될 수 있습니다. 또한, 이상 발생 시 대처 방법이 변경될 수도 있습니다.

자세한 내용은 "NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

● 시스템 정의 변수의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 사용할 수 있는 시스템 정의 변수가 추가 또는 변경될 수 있습니다.

자세한 내용은 'A-6 시스템 정의 변수 목록(A-64페이지)'을 참조하십시오.

A-16 NJ 시리즈의 버전 관련 정보

NJ 시리즈 CPU 유니트와 Sysmac Studio의 버전 대응 및 유니트 버전별 지원 기능에 대해 설명합니다.

A-16-1 유니트 버전과 Sysmac Studio 버전의 관계

NJ 시리즈 CPU 유니트의 유니트 버전 및 Sysmac Studio의 버전에는 대응 관계가 있습니다. 기본적으로 대응하는 버전끼리 조합하여 사용하십시오.

유니트 버전과 Sysmac Studio의 대응 버전

NJ 시리즈 CPU 유니트의 유니트 버전과 Sysmac Studio의 대응 버전 표는 다음과 같습니다.

CPU 유니트의 유니트 버전	Sysmac Studio의 대응 버전
Ver.1.18	Ver.1.22
Ver.1.17	Ver.1.21
Ver.1.16	Ver.1.20
Ver.1.15	Ver.1.19
Ver.1.14	Ver.1.18
Ver.1.13	Ver.1.17
Ver.1.12	Ver.1.16
Ver.1.11	Ver.1.15
Ver.1.10*1	Ver.1.13*2
	Ver.1.12
Ver.1.09	Ver.1.10
Ver.1.08	Ver.1.09
Ver.1.07	Ver.1.08
Ver.1.06	Ver.1.07
Ver.1.05	Ver.1.06
Ver.1.04	Ver.1.05
Ver.1.03	Ver.1.04
Ver.1.02	Ver.1.03
Ver.1.01	Ver.1.02
Ver.1.00*3	Ver.1.01
	Ver.1.00

*1. CPU 유니트 NJ101-□□□□에는 Ver.1.09 이하는 없습니다.
*2. CPU 유니트 NJ101-□□□□에는 Sysmac Studio Ver.1.13 이상을 사용하십시오. Sysmac Studio Ver.1.12 이하에서는 NJ101-□□□□을 사용할 수 없습니다.
*3. CPU 유니트 NJ301-□□□□에는 Ver.1.00이 없습니다. 따라서 Sysmac Studio Ver.1.01 이하에서는 NJ301-□□□□을 사용할 수 없습니다.

유니트 버전과 Sysmac Studio의 버전이 맞지 않는 경우의 사양

NJ 시리즈 CPU 유니트의 유니트 버전과 Sysmac Studio의 버전이 맞지 않는 경우의 사양은 다음과 같습니다.

● Sysmac Studio의 버전이 낮은 경우

Sysmac Studio의 버전이 낮은 경우에 사용할 수 있는 기능은 Sysmac Studio의 버전에 대응하는 CPU 유니트의 유니트 버전이 탑재된 기능입니다.

예) CPU 유니트의 유니트 버전: Ver.1.04
Sysmac Studio 버전: Ver.1.04
Syamac Studio Ver.1.04에 대응하는 CPU 유니트의 유니트 버전은 Ver.1.03입니다. 따라서 사용할 수 있는 기능은 CPU 유니트가 Ver.1.03이 탑재된 기능입니다. CPU 유니트 Ver.1.04 이상에서 추가된 기능은 사용할 수 없습니다.

● CPU 유니트의 유니트 버전이 낮은 경우

NJ 시리즈 CPU 유니트의 유니트 버전이 낮은 경우는 Sysmac Studio의 프로젝트의 장치 선택 화면에서 연결된 CPU 유니트의 유니트 버전 이하로 선택하십시오. 사용할 수 있는 기능은 연결된 CPU 유니트의 유니트 버전이 탑재된 기능입니다.

예) CPU 유니트의 유니트 버전: Ver.1.03
Sysmac Studio 버전: Ver.1.05
Syamac Studio Ver.1.05에 대응하는 CPU 유니트의 유니트 버전은 Ver.1.04입니다. 그러나 연결된 CPU 유니트의 유니트 버전이 Ver.1.03이므로 프로젝트의 장치 선택 화면의 '버전'을 'Ver.1.03' 이하로 선택하십시오.
장치 선택 화면의 '버전'을 'Ver.1.03'으로 선택한 경우에 사용할 수 있는 기능은 CPU 유니트 Ver.1.03이 탑재된 기능입니다. CPU 유니트 Ver.1.04 이상에서 추가된 기능은 사용할 수 없습니다.

사용상의 주의 사항

다음 조건에 해당하는 조작을 하면 이상이 발생하므로 주의하십시오.

- Sysmac Studio Ver.1.02 이상에서 프로젝트의 장치 선택 화면의 '버전'을 '1.01' 이상으로 선택하여 프로젝트를 작성함
- 그 프로젝트를 Sysmac Studio를 Ver.1.01에 업로드함

참고

프로젝트의 장치에 설정하는 유니트 버전

- Sysmac Studio Ver.1.02 이상에서는 유니트 버전 선택을 장치 선택 화면에서 설정합니다.
- 설정할 수 있는 유니트 버전은 사용하는 CPU 유니트의 유니트 버전 이하입니다. 예를 들어, CPU 유니트의 유니트 버전이 Ver.1.01인 경우에는 '1.00' 또는 '1.01'을 선택하여 설정하십시오.
- Sysmac Studio에서는 프로젝트의 장치에 설정한 유니트 버전을 CPU 유니트의 유니트 버전으로 취급합니다. 예를 들어, 프로젝트의 장치에 설정하는 유니트 버전이 Ver.1.00인 경우에는 Sysmac Studio에서 유니트 버전 Ver.1.00까지의 기능을 사용할 수 있습니다.
- 프로젝트의 장치에 설정한 유니트 버전이 전송대상 CPU 유니트의 유니트 버전 이하인 경우에 Sysmac Studio에서 프로젝트를 전송할 수 있습니다.
- 장치 선택 화면의 세부 내용은 "Sysmac Studio Version 1 조작 설명서(SBCA-470)"를 참조하십시오.

A-16-2 하드웨어 개정 번호와 Sysmac Studio 버전의 관계

NJ 시리즈 CPU 유니트의 하드웨어 개정 번호와 Sysmac Studio의 버전은 대응 관계입니다. Sysmac Studio의 시뮬레이터를 실행 시간 예측 모드로 실행하는 경우에 대응하는 버전 이상의 Sysmac Studio를 사용하십시오. Sysmac Studio의 버전이 더 오래된 경우에 해당하는 하드웨어 개정 번호를 선택할 수 없습니다.

형식	CPU 유니트의 하드웨어 개정 번호	Sysmac Studio의 대응 버전
NJ501-□□□□	A	Ver.1.14

A-16-3 유니트 버전 갱신에 따른 지원 기능의 추가와 변경

NJ 시리즈 CPU 유니트의 유니트 버전 갱신에 따라 추가 또는 변경된 지원 기능을 소개합니다.

● 기능 사양의 추가와 변경

추가 또는 변경된 기능 사양별로 CPU 유니트의 유니트 버전과 Sysmac Studio 버전을 나타냅니다.

기능				추가/변경	유니트 버전	Sysmac Studio의 버전	참조할 항목
태스크 기능	기능	조건 성립 시 실행할 태스크		추가	Ver.1.03	Ver.1.04	📖'5-2 태스크의 개요(5-5페이지)'
프로그래밍 기능	이름 공간			추가	Ver.1.01	Ver.1.02	📖'6-7 이름 공간(6-143페이지)'
	데이터 형식	구조체 형식	멤버의 오프셋 지정	추가	Ver.1.01	Ver.1.02	📖'구조체 멤버의 오프셋 지정(6-43페이지)'
				변경		Ver.1.03	
	변수	배열	가변 길이 배열	추가	Ver.1.18	Ver.1.22	📖'6-3-7 데이터 형식의 배열/범위 지정(6-51페이지)'
	라이브러리			추가	Ver.1.01	Ver.1.02	📖'6-8 라이브러리(6-148페이지)'
모션 제어 기능	단축	단축 위치 제어	주기적 동기 절대 위치 결정	추가	Ver.1.03	Ver.1.04	📖"NJ/NX 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"
		단축 제어 보조	파라메터 지정 원점 복귀	추가	Ver.1.03	Ver.1.04	
			디지털 캠 스위치 유효	추가	Ver.1.06	Ver.1.07	
			명령 위치 보정	추가	Ver.1.10	Ver.1.12	
			기동 속도	추가	Ver.1.05	Ver.1.06	
	축 그룹	다축 협조 제어	축 그룹 주기적 동기 절대 위치 제어	추가	Ver.1.01	Ver.1.02	
		다축 협조 제어 보조	축 그룹 위치 취득	추가	Ver.1.01	Ver.1.02	
			축 그룹 구성축 쓰기	추가	Ver.1.01	Ver.1.02	
	공통	캠	캠 테이블 생성	추가	Ver.1.08	Ver.1.09	
		파라메터	축 파라메터의 변경	추가	Ver.1.08	Ver.1.09	
	보조 기능	입력 신호의 논리 반전		추가	Ver.1.05	Ver.1.06	

기능				추가/변경	유니트 버전	Sysmac Studio의 버전	참조할 항목
유니트(입출력) 관리 기능	NX 유니트			추가	Ver.1.05	Ver.1.06	📖"NX 시리즈 EtherCAT 커플러 유니트 사용 설명서(SBCD-361)"
통신 기능	Ethernet/IP 포트	TCP/IP 어플리케이션	FTP 클라이언트	추가	Ver.1.08	Ver.1.09	📖"NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용 설명서(SBCD-359)"
		OPC UA	서버 기능	추가	유니트 버전과 Sysmac Studio의 대응 버전에 대해서는 📖"NJ/NX 시리즈 CPU 유니트 사용 설명서 OPC UA편(SBCD-374)"을 참조		
	EtherCAT 포트	패킷 모니터링*1 (NJ301-□□□□)		추가	Ver.1.10	Ver.1.12	📖"NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용 설명서(SBCD-358)"
	통신용 명령			변경	Ver.1.08 Ver.1.11	Ver.1.09 Ver.1.15	📖"NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)"
디버그 기능	미분 모니터링			추가	Ver.1.03	Ver.1.04	📖'8-6-5 미분 모니터링(8-61페이지)'
높은 신뢰 기능	자체 진단	컨트롤러 이상	중요도 변경	추가	Ver.1.03	Ver.1.04	📖'8-8 이벤트 중요도 변경 기능(8-80페이지)'
보안 기능	고객 자산 보호/잘못된 조작 방지	보호 기능	데이터 보호 기능	추가	Ver.1.01	Ver.1.02	📖'8-5-4 데이터 보호 기능(8-40페이지)'
		조작 권한 인증 기능	그룹 수	변경	Ver.1.01	Ver.1.02	📖'8-5-5 조작 권한 인증 기능(8-42페이지)'
SD 메모리 카드 기능	어플리케이션	SD 메모리 카드 자동 전송 기능		추가	Ver.1.03	Ver.1.04	📖'9-4 SD 메모리 카드 자동 전송 기능(9-32페이지)'
		SD 메모리 카드 프로그램 전송 기능		추가	Ver.1.11	Ver.1.15	📖'9-5 SD 메모리 카드 프로그램 전송 기능(9-34페이지)'

기능				추가/변경	유니트 버전	Sysmac Studio의 버전	참조할 항목
백업 기능	SD 메모리 카드 백업 기능	조작 방법	CPU 유니트 전면 스위치	추가	Ver.1.03	Ver.1.04	📖'9-2 SD 메모리 카드백업 기능(9-13페이지)'
			시스템 정의 변수를 통한 지시	추가	Ver.1.03	Ver.1.04	
			Sysmac Studio 메모리 카드 화면	추가	Ver.1.03	Ver.1.04	
			전용 명령	추가	Ver.1.08	Ver.1.09	
		보호 기능	SD 메모리 카드에 대한 백업 금지	추가	Ver.1.03	Ver.1.04	📖'9-3 SD 메모리 카드에 대한 백업 금지 기능(9-31페이지)'
	Sysmac Studio 컨트롤러 백업 기능			추가	Ver.1.03	Ver.1.04	📖'9-6 Sysmac Studio 컨트롤러 백업 기능 (9-40페이지)'

*1. CPU 유니트 NJ301-□□□□에서만 추가됐습니다. NJ501-□□□□, NJ101-□□□□는 모든 버전에서 패킷 모니터링을 사용할 수 있습니다.

(주의) CPU 유니트의 유니트 버전 갱신에 따른 각 기능 모듈의 기능 추가, 변경에 대해서는 각 기능 모듈의 설명서를 참고하십시오.

● 장착할 수 있는 CJ 유니트 추가

CPU 유니트의 유니트 버전 갱신에 의해 장착할 수 있는 CJ 시리즈 유니트가 추가되는 경우가 있습니다.

자세한 내용은 📖"NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"을 참조하십시오.

● 기본 명령 및 모션 제어 명령의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 사용할 수 있는 기본 명령 및 모션 제어 명령이 추가 또는 변경될 수 있습니다.

자세한 내용은 📖"NJ/NX 시리즈 명령 참조 설명서 기본편(SBCA-360)" 및 📖"NJ/NX 시리즈 명령 참조 설명서 모션편(SBCE-364)"을 참조하십시오.

● 컨트롤러 이벤트의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 발생하는 이벤트가 추가 또는 변경될 수 있습니다. 또한, 이상 발생 시 대처 방법이 변경될 수도 있습니다.

자세한 내용은 📖"NJ/NX 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

● 시스템 정의 변수의 추가와 변경

CPU 유니트의 유니트 버전 갱신에 의해 사용할 수 있는 시스템 정의 변수가 추가 또는 변경될 수 있습니다.

자세한 내용은 🕮'A-6 시스템 정의 변수 목록(A-64페이지)'을 참조하십시오.

A-16-4 유니트 버전의 갱신에 의한 성능 개선

NJ 시리즈 CPU 유니트의 유니트 버전, 및 Sysmac Studio의 버전 갱신에 따라 성능 개선된 기능을 소개합니다.

기능				성능값	유니트 버전	SysmacStudio의 버전
프로그래밍	프로그램 용량	수량	POU 인스턴스 수 (NJ501-□□□□)	9,000	−	Ver.1.06 이상
				6,000	−	Ver.1.05 이하
			POU 인스턴스 수 (NJ301-□□□□)	3,000	Ver.1.04 이상	Ver.1.05 이상
				1,500		Ver.1.04 이하
				2,400	Ver.1.03 이하	Ver.1.05 이상
				1,500		Ver.1.04 이하
	변수 용량	유지 속성 있음	변수의 수*1 (NJ301-□□□□)	5,000	Ver.1.04 이상	Ver.1.05 이상
				2,500		Ver.1.04 이하
				2,500	Ver.1.03 이하	−
모션 제어	제어축 수	최대 제어축 수*2*3*4 (NJ301-□□□□)		15축	Ver.1.06 이상	Ver.1.07 이상
				8축 (NJ301-1200) 4축 (NJ301-1100)	상기 이외의 조합	
		단축 제어 최대 수*4*5 (NJ301-□□□□)		단축 15축	Ver.1.06 이상	Ver.1.07 이상
				단축 8축 (NJ301-1200) 단축 4축 (NJ301-1100)	상기 이외의 조합	
내장 EtherNet/IP 포트	CIP 서비스 : 태그 데이터 링크 (사이클릭 통신)	패킷 간격		연결마다 설정 가능 1~10000ms (1ms 단위)	Ver.1.03 이상	−
				연결마다 설정 가능 10~10000ms (1ms 단위)	Ver.1.02 이하	
		유니트 허용 통신 대역		3000pps*6 (하트비트 포함)	Ver.1.03 이상	−
				1000pps (하트비트 포함)	Ver.1.02 이하	
	TCP 소켓 수			30	Ver.1.03 이상	−
				16	Ver.1.02 이하	
내장 EtherCAT 포트	통신 주기*7 (NJ301-□□□□)			500/1000/2000/4000μs	Ver.1.03 이상	−
				1000/2000/4000μs	Ver.1.02 이하	

*1. CPU 유니트 NJ301-□□□□만의 성능 개선입니다. NJ501-□□□□의 유지 속성이 있는 변수 수의 최대값은 10,000입니다.
*2. 모든 축 종류의 합계입니다.
*3. CPU 유니트 NJ301-□□□□만의 성능 개선입니다. NJ501-□□□□의 최대 제어축 수는 다음과 같습니다.
NJ501-□5□□: 64축, NJ501-□4□□: 32축, NJ501-□3□□: 16축
*4. 사용하는 실제 축의 최대 수에 변경은 없습니다.
*5. CPU 유니트 NJ301-□□□□만의 성능 개선입니다. NJ501-□□□□의 단축 제어 최대 수는 다음과 같습니다.
NJ501-□5□□: 단축 64축, NJ501-□4□□: 단축 32축, NJ501-□3□□: 단축 16축

*6. Packet Per Second를 의미하며, 1초간 처리 가능한 송수신 패킷 수를 나타냅니다.
*7. CPU 유니트 NJ301-□□□□만의 성능 개선입니다. NJ501-□□□□는 500/1000/2000/4000μs의 통신 주기, NJ101-□□□□는 1000/2000/4000μs의 통신 주기를 사용할 수 있습니다.

A-17 설명서 번호 변경 전의 개정 이력

이 설명서는 설명서 번호를 다음과 같이 변경했습니다.

현재 설명서 번호	변경 전의 설명서 번호
SBCA-467	SBCA-359

변경 전의 설명서 번호에서의 개정 이력을 표시합니다.

개정 기호	개정 연월	개정 이유 및 개정 페이지
A	2011년 7월	초판
B	2012년 3월	• NJ301-□□□□ 추가 • CPU 유니트 Ver.1.01 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
C	2012년 5월	• CPU 유니트 Ver.1.02 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
D	2012년 8월	• CPU 유니트 Ver.1.03 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
E	2013년 2월	• CPU 유니트 Ver.1.04 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
F	2013년 4월	• CPU 유니트 Ver.1.05 이상의 지원 기능에 대한 기술을 추가 기재함 • NX 시리즈에 관한 기술을 추가 기재 • 잘못 기재된 내용 수정
G	2013년 6월	• CPU 유니트 Ver.1.06 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
H	2013년 9월	• CPU 유니트 Ver.1.07 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
J	2013년 12월	• CPU 유니트 Ver.1.08 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
K	2014년 7월	• 잘못 기재된 내용 수정
L	2015년 1월	• CPU 유니트 Ver.1.10 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
M	2015년 4월	• NX701-□□□□, NJ101-□□□□ 추가 • 잘못 기재된 내용 수정
N	2015년 10월	• 하드웨어 개정에 관한 기술을 추가 • 잘못 기재된 내용 수정
P	2016년 4월	• CPU 유니트 Ver.1.11 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
R	2016년 7월	• CPU 유니트 Ver.1.12 이상의 지원 기능에 대한 기술을 추가 기재함
S	2016년 10월	• NX1P2-□□□□□□ 추가 • CPU 유니트 Ver.1.13 이상의 지원 기능에 대한 기술을 추가 기재함
T	2017년 1월	• 잘못 기재된 내용 수정
U	2017년 4월	• CPU 유니트 Ver.1.14 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
V	2017년 6월	• 잘못 기재된 내용 수정
W	2017년 6월	• CPU 유니트 Ver.1.15 이상의 지원 기능에 대한 기술을 추가 기재함
X	2017년 10월	• CPU 유니트 Ver.1.16 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정
Y	2018년 1월	• CPU 유니트 Ver.1.17 이상의 지원 기능에 대한 기술을 추가 기재함
Z	2018년 4월	• CPU 유니트 Ver.1.18 이상의 지원 기능에 대한 기술을 추가 기재함 • 잘못 기재된 내용 수정

I

색인

색인

기호

'래더도 언어의 표준적인 명령'의 명령 종류 구성 (A-56페이지) A-33

A

__Active(태스크 실행 중 플래그)..... 5-7, A-65, A-115
_AlarmFlag(사용자 이상 Status)..... 8-75, 8-79, A-67, A-117
AT(할당대상) 지정 6-59

B

_BackupBusy(백업 관련 기능 실행 중 플래그)... A-71, A-126
BOOL..... 6-32, 6-45
BOOTP 서버 연결 이상..... A-94, A-161
BYTE 6-32, 6-45

C

_Card1Access(SD 메모리 카드 액세스 중 플래그) 8-30, A-67, A-118
_Card1BkupCmd(SD 메모리 카드 백업 지시).... A-68, A-118
_Card1BkupSta(SD 메모리 카드 백업 Status) ... A-68, A-119
_Card1Deteriorated(SD 메모리 카드 수명 경고 플래그)..... 8-29, 8-30, A-67, A-118
_Card1Err(SD 메모리 카드 에러 플래그).... 8-26, 8-30, A-67, A-118
_Card1PowerFail(SD 메모리 카드 액세스 중 전원 OFF 플래그)..... 8-30, A-67, A-118
_Card1PrgTransferCmd(SD 메모리 카드 프로그램 전송 지시)..... A-69, A-121
_Card1PrgTransferSta(SD 메모리 카드 프로그램 전송 Status)..... A-69, A-123
_Card1Protect(SD 메모리 카드 쓰기 방지 플래그)..... 8-30, A-67, A-117
_Card1Ready(SD 메모리 카드 사용 가능 플래그)..... 8-30, A-67, A-117
_Card1RestoreCmd(SD 메모리 카드 복원 지시) A-70, A-123
_Card1RestoreCmdTargetAbsEncoder(앱솔루트 엔코더 원점 위치 오프셋 전송 플래그)..... A-70, A-126
_Card1RestoreCmdTargetIPAdr(IP 주소 전송 플래그)..... A-70, A-125
_Card1RestoreCmdTargetMemory(CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그) A-70, A-125
_Card1RestoreCmdTargetUnitConfig(유니트·슬레이브 전송 플래그)..... A-70, A-125
_Card1RestoreCmdTargetUserProgram(사용자 프로그램·설정 전송 플래그) A-70, A-124
_Card1RestoreCmdTargetVariable(변수의 현재값(유지 속성만 해당) 전송 플래그)..... A-70, A-125
_Card1RestoreSta(SD 메모리 카드 복원 Status)..... A-70, A-124
_Card1VefySta(SD 메모리 카드 비교 Status)..... A-69, A-120
CIP 통신 1 Identity 정보가 잘못됨 A-95
CIP 통신 1 대상 PLC 동작 모드..... A-104, A-171
CIP 통신 1 대상 PLC 이상 정보..... A-105, A-172
CIP 통신 1 대상 노드 이상 정보 A-108, A-173
CIP 통신 1 등록 대상 노드 정보 A-103, A-169
CIP 통신 1 복수 스위치 동시 ON 이상..... A-98, A-166
CIP 통신 1 이상 A-91, A-156
CIP 통신 1 전체 태그 데이터 링크 통신 중 A-102, A-169
CIP 통신 1 태그 데이터 링크 설정 이상..... A-95, A-163
CIP 통신 1 태그 데이터 링크 연결 확립 실패 A-96, A-164
CIP 통신 1 태그 데이터 링크 통신 시작 스위치..... A-110, A-174
CIP 통신 1 태그 데이터 링크 통신 이상..... A-97, A-164
CIP 통신 1 태그 데이터 링크 통신 정지 스위치..... A-110, A-175
CIP 통신 1 태그 데이터 링크 통신 중..... A-101, A-168
CIP 통신 1 태그 해결 이상..... A-98, A-165
CIP 통신 1 정상 대상 노드 정보 A-103, A-170
CIP 통신 2 Identity 정보가 잘못됨 A-95
CIP 통신 2 대상 PLC 동작 모드..... A-105, A-171
CIP 통신 2 대상 PLC 이상 정보..... A-106, A-172
CIP 통신 2 대상 노드 이상 정보 A-108, A-173
CIP 통신 2 등록 대상 노드 정보 A-103, A-170
CIP 통신 2 복수 스위치 동시 ON 이상..... A-98, A-166
CIP 통신 2 이상 A-92, A-156
CIP 통신 2 전체 태그 데이터 링크 통신 중 A-102, A-169
CIP 통신 2 정상 대상 노드 정보 A-104, A-170
CIP 통신 2 태그 데이터 링크 설정 이상..... A-96, A-163
CIP 통신 2 태그 데이터 링크 연결 확립 실패 A-96, A-164
CIP 통신 2 태그 데이터 링크 통신 시작 스위치..... A-110, A-175
CIP 통신 2 태그 데이터 링크 통신 이상..... A-97, A-165
CIP 통신 2 태그 데이터 링크 통신 정지 스위치..... A-111, A-175
CIP 통신 2 태그 데이터 링크 통신 중..... A-102, A-168
CIP 통신 2 태그 해결 이상..... A-98, A-165
CIP 통신 이상..... A-91, A-156
CJ 유니트 구성..... 1-5, 1-11
CJ 유니트용 메모리..... A-180
CJ 유니트용 메모리의 현재값(유지 메모리만 해당) 전송 플래그 A-70, A-125
_CJB_CBU□□InitSta(CPU 고기능 유니트 초기화 중 플래그) 8-24, A-79, A-136
_CJB_CBU□□Restart(CPU 고기능 유니트 재시작 플래그)..... 8-24, A-79, A-137
_CJB_ErrSta(I/O 버스 이상 Status)..... A-78, A-135
_CJB_InRespTm(기본 입력 유니트의 입력 시상수)..... 8-23, A-78, A-136
_CJB_IOUnitInfo(기본 I/O 유니트 정보 영역)..... A-79, A-136

_CJB_MaxRackNo(최대 랙 번호)..... A-78, A-135
_CJB_MaxSlotNo(최대 슬롯 번호)..... A-78, A-135
_CJB_MstrErrSta(I/O 버스 마스터 이상 Status) A-78, A-135
_CJB_SCU00P□ChgSta(시리얼 커뮤니케이션 유니트 0호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그)..... A-79, A-138
_CJB_SCU15P□ChgSta(시리얼 커뮤니케이션 유니트 1-15호기 포트 1/2용 시리얼 포트 설정 변경 중 플래그)..... A-79, A-138
_CJB_SIO□□InitSta(고기능 I/O 유니트 초기화 중 플래그)..... 8-24, A-79, A-137
_CJB_SIO□□Restart(고기능 I/O 유니트 재시작 플래그)..... 8-24, A-79, A-137
_CJB_UnitErrSta(I/O 버스 유니트 이상 Status).... A-78, A-136
CPU 고기능 유니트 초기화 중 플래그..... 8-24, A-79, A-136
CPU 고기능 유니트 재시작 플래그..... 8-24, A-79, A-137
CPU 유니트 내부 온도 초과 플래그..... A-73, A-129
CPU 유니트 내의 데이터..... 2-35
CPU 유니트 상의 NX 유니트와의 액세스..... 2-18
CPU 유니트 이름..... 8-45
CPU 유니트 이름 기능..... 8-45
CPU 유니트에 대한 쓰기 방지 기능..... 8-44
CPU 유니트의 상태..... 2-36
_CurrentTime(시스템 시각)..... 8-5, A-65, A-115
CX-Designer의 변수 테이블 등록 방법..... A-198

D

DATE..... 6-32, 6-45
DATE_ AND_ TIME 6-33, 6-45
_DeviceOutHoldCfg(장치 출력 유지 설정)..... A-73, A-130
_DeviceOutHoldStatus(장치 출력 유지 상태) A-73, A-130
DINT..... 6-32
DNS 서버 연결 이상..... A-99, A-167
DNS 설정 이상 A-94, A-161
DWORD..... 6-32

E

_EC1_InDataInvalid(입력 데이터 1 무효) A-87
_EC2_InDataInvalid(입력 데이터 2 무효) A-88
_EC_CommErrTbl(통신 이상 슬레이브 테이블)..... A-85, A-149
_EC_CycleExceeded(EtherCAT 통신 주기 설정 이상)..... A-85, A-149
_EC_DisableSlavTbl(무효 설정 슬레이브 테이블) A-87, A-150
_EC_DisconnSlavTbl(이탈 명령 중 슬레이브 테이블)..... A-87, A-150
_EC_EntrySlavTbl(네트워크 가입 슬레이브 테이블)..... A-87, A-149
_EC_ErrSta(내장 EtherCAT 이상) A-84, A-145
_EC_InData1Invalid(입력 데이터 1 무효) A-152
_EC_InData2Invalid(입력 데이터 2 무효) A-152
_EC_InDataInvalid(입력 데이터 무효)..... A-87, A-151
_EC_IndataInvalidErr(프로세스 데이터 입력값 무효 이상)..... A-85, A-148
_EC_LanHwErr(통신 컨트롤러 이상) A-84, A-146
_EC_LinkOffErr(링크 오프 이상)..... A-84, A-146
_EC_LinkStatus(링크 Status)..... A-87, A-151
_EC_MacAdrErr(MAC 주소 이상)..... A-84, A-146
_EC_MBXSlavTbl(메시지 통신 가능 슬레이브 테이블)..... A-87, A-150
_EC_MsgErr(EtherCAT 메시지 이상)..... A-85, A-148
_EC_MstrErr(마스터 이상)..... A-84, A-145
_EC_NetCfgCmpErr(네트워크 구성 비교 이상)..... A-84, A-147
_EC_NetCfgErr(네트워크 구성 정보 이상)..... A-84, A-146
_EC_NetTopologyErr(네트워크 구성 이상)..... A-84, A-147
_EC_PDActive(프로세스 데이터 통신 Status)... A-87, A-151
_EC_PDCommErr(프로세스 데이터 통신 이상) A-85, A-147
_EC_PDSendErr(프로세스 데이터 송신 이상) .. A-85, A-147
_EC_PDSlavTbl(프로세스 데이터 통신 중 슬레이브 테이블)..... A-87, A-150
_EC_PDTimeoutErr(프로세스 데이터 수신 타임아웃 이상)..... A-85, A-147
_EC_PktMonStop(패킷 모니터링 정지)..... A-87, A-151
_EC_PktSaving(패킷 데이터 파일 저장 중)..... A-87, A-151
_EC_PortErr(통신 포트 이상)..... A-84, A-145
_EC_RegSlavTbl(등록 슬레이브 테이블)..... A-87, A-149
_EC_SlavAdrDupErr(슬레이브 노드 주소 중복 이상)..... A-85, A-147
_EC_SlavAppErr(슬레이브 어플리케이션 이상)..... A-85, A-148
_EC_SlavEmergErr(비상 메시지 감지) A-85, A-148
_EC_SlavErr(슬레이브 이상) A-84, A-145
_EC_SlavErrTbl(슬레이브 이상 테이블)..... A-84, A-146
_EC_SlavInitErr(슬레이브 초기화 이상)..... A-85, A-148
_EC_StatisticsLogBusy(진단 통계 정보 로그 실행 상태 플래그) A-88, A-153
_EC_StatisticsLogCycleSec(진단 통계 정보 로그 저장 주기) A-88, A-153
_EC_StatisticsLogEnable(진단 통계 정보 로그 허가)..... A-88, A-153
_EC_StatisticsLogErr(진단 통계 정보 로그 이상 종료 플래그)..... A-88, A-153
_EIP1_BootpErr(통신 포트 1 BOOTP 서버 연결 이상)..... A-94, A-161
_EIP1_CipErr(CIP 통신 1 이상) A-91, A-156
_EIP1_EstbTargetSta(CIP 통신 1 정상 대상 노드 정보)..... A-103, A-170
_EIP1_EtnCfgErr(통신 포트 1 Ethernet 기본 설정 이상)..... A-93, A-159
_EIP1_EtnOnlineSta(통신 포트 1 온라인)..... A-101, A-167
_EIP1_IdentityErr(CIP 통신 1 Identity 정보가 잘못됨)..... A-95, A-162
_EIP1_IPAdrCfgErr(통신 포트 1 IP 주소 설정 이상) A-93, A-159
_EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) .. A-160
_EIP1_IPAdrDupErr(통신 포트 1 IP 주소 중복 이상) A-94

_EIP1_LanHwErr(통신 포트 1 통신 컨트롤러 이상) A-93, A-158
_EIP1_MacAdrErr(통신 포트 1 MAC 주소 이상) A-92, A-157
_EIP1_MultiSwONErr(CIP 통신 1 복수 스위치 동시 ON 이상)..... A-98, A-166
_EIP1_PortErr(통신 포트 1 이상)..... A-90, A-155
_EIP1_RegTargetSta(CIP 통신 1 등록 대상 노드 정보) A-103, A-169
_EIP1_TagAdrErr(CIP 통신 1 태그 해결 이상)... A-98, A-165
_EIP1_TargetNodeErr(CIP 통신 1 대상 노드 이상 정보) A-108, A-173
_EIP1_TargetPLCErr(CIP 통신 1 대상 PLC 이상 정보) A-105, A-172
_EIP1_TargetPLCModeSta(CIP 통신 1 대상 PLC 동작 모드)..... A-104, A-171
_EIP1_TDLinkAllRunSta(CIP 통신 1 전체 태그 데이터 링크 통신 중) A-102, A-169
_EIP1_TDLinkCfgErr(CIP 통신 1 태그 데이터 링크 설정 이상)..... A-95, A-163
_EIP1_TDLinkErr(CIP 통신 1 태그 데이터 링크 통신 이상)..... A-97, A-164
_EIP1_TDLinkOpnErr(CIP 통신 1 태그 데이터 링크 연결 확립 실패) A-96, A-164
_EIP1_TDLinkRunSta(CIP 통신 1 태그 데이터 링크 통신 중)..... A-101, A-168
_EIP1_TDLinkStartCmd(CIP 통신 1 태그 데이터 링크 통신 시작 스위치)..... A-110, A-174
_EIP1_TDLinkStopCmd(CIP 통신 1 태그 데이터 링크 통신 정지 스위치)..... A-110, A-175
_EIP2_BootpErr(통신 포트 2 BOOTP 서버 연결 이상) A-95, A-161
_EIP2_CipErr(CIP 통신 2 이상)..... A-92, A-156
_EIP2_EstbTargetSta(CIP 통신 2 정상 대상 노드 정보) A-104, A-170
_EIP2_EtnCfgErr(통신 포트 2 Ethernet 기본 설정 이상) A-93, A-159
_EIP2_EtnOnlineSta(통신 포트 2 온라인) A-101, A-167
_EIP2_IdentityErr(CIP 통신 2 Identity 정보가 잘못됨)..... A-95, A-162
_EIP2_IPAdrCfgErr(통신 포트 2 IP 주소 설정 이상) A-94, A-160
_EIP2_IPAdrDupErr(통신 포트 2 IP 주소 중복 이상).. A-160
_EIP2_IPAdrDupErr(통신 포트 2 IP 주소 중복 이상).... A-94
_EIP2_LanHwErr(통신 포트 2 통신 컨트롤러 이상) A-93, A-158
_EIP2_MacAdrErr(통신 포트 2 MAC 주소 이상)..... A-92
_EIP2_MultiSwONErr(CIP 통신 2 복수 스위치 동시 ON 이상)..... A-98, A-166
_EIP2_PortErr(통신 포트 2 이상)..... A-90, A-155
_EIP2_RegTargetSta(CIP 통신 2 등록 대상 노드 정보) A-103, A-170
_EIP2_TagAdrErr(CIP 통신 2 태그 해결 이상)... A-98, A-165
_EIP2_TargetNodeErr(CIP 통신 2 대상 노드 이상 정보) A-108, A-173
_EIP2_TargetPLCErr(CIP 통신 2 대상 PLC 이상 정보) A-106, A-172
_EIP2_TargetPLCModeSta(CIP 통신 2 대상 PLC 동작 모드)..... A-105, A-171
_EIP2_TDLinkAllRunSta(CIP 통신 2 전체 태그 데이터 링크 통신 중) A-102, A-169
_EIP2_TDLinkCfgErr(CIP 통신 2 태그 데이터 링크 설정 이상)..... A-96, A-163
_EIP2_TDLinkErr(CIP 통신 2 태그 데이터 링크 통신 이상) A-97, A-165
_EIP2_TDLinkOpnErr(CIP 통신 2 태그 데이터 링크 연결 확립 실패) A-96, A-164
_EIP2_TDLinkRunSta(CIP 통신 2 태그 데이터 링크 통신 중)..... A-102, A-168
_EIP2_TDLinkStartCmd(CIP 통신 2 태그 데이터 링크 통신 시작 스위치) A-110, A-175
_EIP2_TDLinkStopCmd(CIP 통신 2 태그 데이터 링크 통신 정지 스위치) A-111, A-175
_EIP_BootpErr(BOOTP 서버 연결 이상) A-94, A-161
_EIP_CipErr(CIP 통신 이상) A-91, A-156
_EIP_DNSCfgErr(DNS 설정 이상) A-94, A-161
_EIP_DNSSrvErr(DNS 서버 연결 이상)..... A-99, A-167
_EIP_ErrSta(내장 EtherNet/IP 이상)..... A-89, A-154
_EIP_EstbTargetSta(정상 대상 노드 정보) A-103, A-170
_EIP_EtnCfgErr(Ethernet 기본 설정 이상) A-93, A-158
_EIP_EtnOnlineSta(온라인)..... A-101, A-167
_EIP_IdentityErr(Identity 정보가 잘못됨)..... A-95, A-162
_EIP_IPAdrCfgErr(IP 주소 설정 이상)..... A-93, A-159
_EIP_IPAdrDupErr(IP 주소 중복 이상)..... A-94, A-160
_EIP_IPRTblErr(IP 라우트 테이블 이상)..... A-95, A-162
_EIP_LanHwErr(통신 컨트롤러 이상)..... A-92, A-158
_EIP_MacAdrErr(MAC 주소 이상) A-92, A-157
_EIP_MultiSwONErr(복수 스위치 동시 ON 이상) A-98, A-166
_EIP_NTPResult(NTP의 동작 정보) A-108
_EIP_NTPResult.ExecNormal(NTP의 동작 결과) A-109, A-174
_EIP_NTPResult.ExecTime(NTP의 최종 동작 시각) A-109, A-174
_EIP_NTPSrvErr(NTP 서버 연결 이상) A-99, A-166
_EIP_PortErr(통신 포트 이상) A-89, A-154
_EIP_RegTargetSta(등록 대상 노드 정보)..... A-102, A-169
_EIP_TagAdrErr(태그 해결 이상) A-97, A-165
_EIP_TargetNodeErr(대상 노드 이상 정보)..... A-107, A-173
_EIP_TargetPLCErr(대상 PLC 이상 정보)..... A-105, A-172
_EIP_TargetPLCModeSta(대상 PLC 동작 모드) A-104, A-171
_EIP_TcpAppCfgErr(TCP/IP 설정 이상) A-99, A-166
_EIP_TcpAppErr(TCP 어플리케이션 통신 이상) A-92, A-157
_EIP_TDLinkAllRunSta(전체 태그 데이터 링크 통신 중)..... A-102, A-168
_EIP_TDLinkCfgErr(태그 데이터 링크 설정 이상) A-95, A-163
_EIP_TDLinkErr(태그 데이터 링크 통신 이상) ... A-97, A-164
_EIP_TDLinkOpnErr(태그 데이터 링크 연결 확립 실패) A-96, A-163
_EIP_TDLinkRunSta(태그 데이터 링크 통신 중)..... A-101, A-168

_EIP_TDLinkStartCmd(태그 데이터 링크 통신 시작 스위치) A-110, A-174
_EIP_TDLinkStopCmd(태그 데이터 링크 통신 정지 스위치) A-110, A-175
EN 6-20
ENO 6-21
_ErrSta(컨트롤러 이상 Status) 5-100, A-67, A-117
EtherCAT 네트워크 구성 1-5, 1-7, 1-9, 1-11
EtherCAT 마스터 기능 모듈의 초기 설정 4-24
EtherCAT 메시지 이상 A-85, A-148
EtherCAT 슬레이브 구성 3-4
EtherCAT 슬레이브 및 축 유효/무효 전환 A-201
EtherCAT 슬레이브와의 액세스 2-16
EtherCAT 통신 주기 설정 이상 A-85, A-149
Ethernet 기본 설정 이상 A-93, A-158
EtherNet/IP 기능 모듈의 초기 설정 4-25
__ExceedCount(태스크 주기 초과 횟수) 5-7, 5-100, A-66, A-116
__Exceeded(태스크 주기 초과 플래그) 5-7, 5-100, 5-106, A-66, A-116
__ExecCount(태스크 실행 횟수) 5-7, 5-106, A-65, A-116

F

FAN 회전 수 저하 플래그 A-73, A-129
FB명 또는 명령명 6-10
FINS/TCP 연결 Status A-72, A-128
_FINSTCPConnSta(FINS/TCP 연결 Status) A-72, A-128
FTP 서버 기능 8-29
FTP 클라이언트 통신 명령 8-28
FUN 정의명 또는 명령명 6-18

H

_HardwareRevision(하드웨어 개정) A-73, A-129

I

I/O 리프레시 주기 초과 5-102
I/O 버스 마스터 이상 Status A-78, A-135
I/O 버스 유니트 이상 Status A-78, A-136
I/O 버스 이상 Status A-78, A-135
I/O 포트 3-8
I/O 포트명 3-9
I/O를 제어하는 태스크 설정 4-9
CIP 통신 1 Identity 정보가 잘못됨 A-162
CIP 통신 2 Identity 정보가 잘못됨 A-162
Identity 정보가 잘못됨 A-95, A-162
INT 6-32
IP 라우트 테이블 이상 A-95, A-162
IP 주소 전송 플래그 A-70, A-125
통신 포트 1 IP 주소 중복 이상 A-160
통신 포트 2 IP 주소 중복 이상 A-160
IP 주소 설정 이상 A-93, A-159
IP 주소 중복 이상 A-94, A-160

L

__LastExecTime(이전 태스크 실행 시간) 5-7, A-65, A-115
LD 6-90
LINT 6-32
LREAL 6-32
LWORD 6-32

M

MAC 주소 이상 A-84, A-92, A-146, A-157
__MaxExecTime(태스크 실행 시간 최대값) 5-7, A-65, A-115
MC 시운전 7-9
_MC1_AX(축 변수) A-83, A-144
_MC1_GRP(축 그룹 변수) A-83, A-143
_MC2_AX(축 변수) A-84, A-144
_MC2_GRP(축 그룹 변수) A-83, A-143
_MC_AX(축 변수) A-83, A-144
_MC_AX_ErrSta(축 이상 Status) A-82, A-142
_MC_COM(공통 변수) A-82, A-143
_MC_ComErrSta(공통 이상 Status) A-82, A-142
_MC_ErrSta(모션 제어 기능 모듈의 이상 Status) A-82, A-142
_MC_GRP(축 그룹 변수) A-82, A-143
_MC_GRP_ErrSta(축 그룹 이상 Status) A-82, A-142
__MinExecTime(태스크 실행 시간 최소값) 5-7, A-65, A-115

N

NTP 8-5
NTP 서버 연결 이상 A-99, A-166
NTP의 동작 결과 A-109, A-174
NTP의 동작 정보 A-108
NTP의 최종 동작 시각 A-109, A-174
NX 버스 기능 모듈 마스터 이상 Status A-81, A-140
NX 버스 기능 모듈 유니트 이상 Status A-81, A-141
NX 버스 기능 모듈 이상 Status A-81, A-140
NX 유니트 I/O 데이터 통신 중 Status A-80, A-139, A-141
NX 유니트 구성 1-7, 1-9
NX 유니트 등록 Status A-80, A-140
NX 유니트 이상 Status A-81, A-141
_NXB_ErrSta(NX 버스 기능 모듈 이상 Status) A-81, A-140
_NXB_MaxUnitNo(최대 NX 유니트 번호) A-80, A-139
_NXB_MstrErrSta(NX 버스 기능 모듈 마스터 이상 Status) A-81, A-140
_NXB_UnitErrFlagTbl(NX 유니트 이상 Status) .. A-81, A-141
_NXB_UnitErrStaTbl(NX 버스 기능 모듈 유니트 이상 Status) A-81, A-141
_NXB_UnitIOActiveTbl(NX 유니트 I/O 데이터 통신 중 Status) A-80, A-139, A-141
_NXB_UnitRegTbl(NX 유니트 등록 Status) A-80, A-140

P

P_CY(캐리 플래그)..... 6-8, 6-136, A-72, A-127
P_First_Run(프로그램 기동 시 1주기 ON 플래그)..... A-72, A-128
P_First_Run(프로그램 기동 시 주기 ON 플래그)..... 6-8
P_First_RunMode(운전 시작 시 1주기 ON 플래그)..... 6-8, A-72, A-127
PLC 기능 모듈..... 4-4
PLC 기능 모듈 이상 Status..... A-74, A-132
_PLC_ErrSta(PLC 기능 모듈 이상 Status)..... A-74, A-132
_PLC_OptBoardSta(옵션 보드 Status)..... A-75, A-132
_PLC_OptBoardSta.Error(옵션 보드 에러 발생)..... A-75, A-133
_PLC_OptBoardSta.isDetect(옵션 보드 장착).... A-75, A-132
_PLC_OptBoardSta.Run(옵션 보드 정상 동작).. A-75, A-132
_PLC_OptSerialErrSta(시리얼 옵션 보드 에러 Status)..... A-75, A-133
_PLC_OptSerialErrSta.Error(전송 에러 발생).... A-76, A-133
_PLC_OptSerialErrSta.FramingErr(프레이밍 에러 발생)..... A-76, A-134
_PLC_OptSerialErrSta.OverRun(오버런 에러 발생)..... A-77, A-134
_PLC_OptSerialErrSta.ParityErr(패리티 에러 발생)..... A-76, A-134
_PLC_TraceSta.IsComplete(트레이스 완료 시 ON 플래그)..... 8-61, A-74, A-131
_PLC_TraceSta.IsStart(트레이스 실행 중 ON 플래그)..... 8-61, A-74, A-131
_PLC_TraceSta.IsTrigger(트레이스 트리거 모니터링 플래그)..... 8-61, A-74, A-131
_PLC_TraceSta.ParamErr(트레이스 파라메터 이상 플래그)..... 8-61, A-74, A-131
P_Off(상시 OFF 플래그)..... A-72, A-127
P_On(상시 ON 플래그)..... A-72, A-127
_Port_isAvailable(네트워크 통신 명령 실행 가능 플래그)..... A-72, A-128
-Port_numUsingPort(사용 포트 수)..... A-72, A-128
POU
  펑션 블록(FB)..... 6-6
  펑션(FUN)..... 6-6
  프로그램..... 6-6
POU(프로그램 구성 단위)..... 6-5
POU명 변경 및 POU 새로 작성 등에 따른 유지 변수의 값..... 6-72
POU의 제한 사항..... 6-26
_PowerOnCount(전원 OFF 발생 횟수)..... A-71, A-126
_PowerOnHour(통전 시간)..... A-71, A-126
P_PRGER(명령 에러 발생 플래그)..... 6-8, 6-138, 6-141, A-72, A-128

R

REAL..... 6-32, 6-45
_RetainFail(전원 OFF 유지 실패 플래그)..... A-71, A-127
RUN 접점..... 8-6

S

SD 메모리 카드
  기능 목록..... 8-25
  사양..... 8-26
  수명 감지 기능..... 8-29
  자체 진단 기능 SD 메모리 카드 자체 진단 기능]..... 8-31
  조작 명령..... 8-28
  파일 액세스의 배타 제어..... 8-33
SD 메모리 카드 백업 Status..... A-68, A-119
SD 메모리 카드 백업 기능..... 9-13
SD 메모리 카드 백업 지시..... A-68, A-118
SD 메모리 카드 복원 Status..... A-70, A-124
SD 메모리 카드 복원 지시..... A-70, A-123
SD 메모리 카드 비교 Status..... A-69, A-120
SD 메모리 카드 사용 가능 플래그..... 8-30, A-67, A-117
SD 메모리 카드 설정..... 4-5
SD 메모리 카드 수명 경고 플래그.... 8-29, 8-30, A-67, A-118
SD 메모리 카드 쓰기 방지 플래그..... 8-30, A-67, A-117
SD 메모리 카드 액세스 중 전원 OFF 플래그..... 8-30, A-67, A-118
SD 메모리 카드 액세스 중 플래그..... 8-30, A-67, A-118
SD 메모리 카드 에러 플래그..... 8-26, 8-30, A-67, A-118
SD 메모리 카드 자동 전송 기능..... 9-32
SD 메모리 카드 프로그램 전송 Status..... A-69, A-123
SD 메모리 카드 프로그램 전송 기능..... 4-5, 9-34
SD 메모리 카드 프로그램 전송 지시..... A-69, A-121
SD 메모리 카드에 대한 백업 금지 기능..... 9-31
_SelfTest_HighTemperature(CPU 유니트 내부 온도 초과 플래그)..... A-73, A-129
_SelfTest_LowBattery(배터리 전압 저하 플래그)..... A-73, A-129
_SelfTest_LowFanRevolution(FAN 회전 수 저하 플래그)..... A-73, A-129
SINT..... 6-32, 6-45
ST 언어..... 6-96
  CASE..... 6-109
  EXIT..... 6-120
  FB 호출..... 6-121
  FUN 호출..... 6-124
  FOR..... 6-112
  IF(복수의 조건)..... 6-107
  IF(하나의 조건)..... 6-105
  REPEAT..... 6-118
  RETURN..... 6-104
  WHILE..... 6-116
  각 연산자의 피연산자에 사용 가능한 데이터 형식..... 6-102
  구문의 키워드..... 6-99
  대입..... 6-103
  연산자..... 6-100
STRING..... 6-33, 6-45
Sysmac Studio 백업 파일 가져오기/내보내기 기능..... 9-45
Sysmac Studio 변수/메모리 백업 기능..... 9-46
Sysmac Studio 컨트롤러 백업 기능..... 9-40

T

TCP 어플리케이션 통신 이상 A-92, A-157
TCP/IP 설정 이상 A-99, A-166
TIME 6-32, 6-45
TIME_ OF_DAY 6-32, 6-45

U

UDINT 6-32
UINT 6-32
ULINT 6-32
_UnitVersion(유니트 버전) A-73, A-129
USINT 6-32

W

WORD 6-32

가

가변 길이 배열 지정 6-55
각 상태의 동작 2-38
강제값 리프레시 8-47
강제값을 프로그램에서 사용하는 경우의 유의 사항 8-50
갱신 태스크 5-90
고기능 I/O 유니트 초기화 중
  플래그 8-24, A-79, A-137
고기능 I/O 유니트 재시작 플래그 8-24, A-79, A-137
고기능 유니트와의 액세스 2-22
고기능 유니트의 초기 설정 4-20
공용체 형식 6-46
공통 변수 A-82, A-143
공통 이상 Status A-82, A-142
구문 에러 6-140
구조체 멤버의 오프셋 지정 6-43
구조체 형식 6-41
글로벌 변수 6-29
기능 모듈
  EtherCAT 마스터 기능 모듈 2-4
  EtherNet/IP 기능 모듈 2-4
  NX 버스 기능 모듈 2-4
  PLC 기능 모듈 2-4
  모션 제어 기능 모듈 2-4
기본 I/O 유니트 정보 영역 A-79, A-136
기본 I/O 유니트와의 액세스 2-21
기본 구성 1-5, 1-7, 1-9, 1-11
기본 데이터 형식 6-31
기본 입력 유니트의 입력 시상수 8-23, A-78, A-136

나

날짜 시각 형식 6-33, 6-45
날짜 형식 6-32, 6-45
내부 변수 6-20
내장 EtherCAT 이상 A-84, A-145
내장 EtherNet/IP 이상 A-89, A-154
내장 I/O와의 액세스 2-20
네트워크 가입 슬레이브 테이블 A-87, A-149
네트워크 공개 6-65
네트워크 구성 1-6, 1-8, 1-10, 1-12
네트워크 구성 비교 이상 A-84, A-147
네트워크 구성 이상 A-84, A-147
네트워크 구성 정보 이상 A-84, A-146
네트워크 통신 명령 실행 가능 플래그 A-72, A-128

다

단순명 6-144
대상 PLC 동작 모드 A-104, A-171
대상 PLC 이상 정보 A-105, A-172
대상 노드 이상 정보 A-107, A-173
데이터 보호 기능 8-40
데이터 유지/비유지 속성 목록 A-176
데이터 트레이스 8-56
데이터 트레이스의 동작 8-59
데이터 트레이스의 사양 8-57
데이터 형식 6-31
데이터 형식 변환 6-40
동작 모드 2-40
동작 모드 전환 2-41
동작 모드 전환 시의 변수 값 유지 2-41
동작 모드 확인 2-42
동작 설정 4-5
동작 설정 탭 4-4
동작 옵션 6-135, 6-136
등록 대상 노드 정보 A-102, A-169
등록 슬레이브 테이블 A-87, A-149
디바이스 변수 2-13, 3-11
디버그 프로그램 7-6

라

라이브러리 6-148
래더도 언어 6-90
래더도 언어의 표준적인 명령'의 명령 종류
  구성(A-56페이지) A-33
래더도에서의 FB, FUN 연결 6-92
로컬 변수 6-29
  내부 변수 6-29
  외부 변수 6-29
  입력 변수 6-29
  입출력 변수 6-29
  출력 변수 6-29
로컬 변수 테이블 6-10, 6-19
리터럴 6-85
링크 Status A-87, A-151
링크 오프 이상 A-84, A-146

마

마스터 이상 A-84, A-145
마스터 컨트롤 6-153
메모리 전체 클리어 8-3

메시지 통신 가능 슬레이브 테이블 A-87, A-150
명령 6-134
명령 에러 6-137
명령 에러 발생 플래그 6-8, 6-138, 6-141, A-72, A-128
모션 제어 기능 모듈의 이상 Status A-82, A-142
모션 제어 기능 모듈의 초기 설정 4-22
모션 제어 시스템 변수 2-14
모션 제어 주기 초과 모션 제어 주기 초과] 5-101
모션 제어의 구조 2-32
무효 설정 슬레이브 테이블 A-87, A-150
문자열 형식 6-33, 6-45
미분 모니터링 8-61

바

반환값 6-21
배열 지정(ARRAY OF) 6-51
배터리 전압 저하 플래그 A-73, A-129
백업 9-3
백업 관련 기능 실행 중 플래그 A-71, A-126
버스바 6-90
버전 관련 정보 A-226
범위 지정((,)) 6-57
변수
  개요 6-28
  종류 6-28
변수 액세스 시간 4-8
변수 액세스 시간 설정 5-98
변수명 6-59
변수명 등의 프로그램 관련 이름의 제한 6-83
변수와 I/O의 액세스 2-15
변수의 메모리 확보 방법 A-185
변수의 속성 6-30
변수의 종류
  사용자 정의 변수 2-12
  시스템 정의 변수 2-12
  준 사용자 정의 변수 2-12
변수의 태스크 간 배타 제어 설정 4-10
변수의 현재값(유지 속성만 해당) 전송 플래그 ... A-70, A-125
보안 설정 4-5
복수 스위치 동시 ON 이상 A-98, A-166
복원 9-3
부울 형식 6-32, 6-45
비교 9-3
비상 메시지 감지 A-85, A-148
비트 열 형식 6-32, 6-45
비트 열 형식, 실수 형식, 문자열 형식의 데이터 형식 6-36

사

사양
  기능 사양 A-13
  성능 사양 A-3
  일반 사양 A-3
사양의 개요 1-13
사용 포트 수 A-72, A-128
사용자 이벤트 8-67, 8-74
사용자 이상 Status 8-75, 8-79, A-67, A-117
사용자 정의 변수 6-28
  종류 6-29
사용자 프로그램 복원 정보 없는 전송 기능 8-38
사용자 프로그램 실행용 ID 8-35
사용자 프로그램 작성상의 유의 사항 6-152
사용자 프로그램·설정 전송 플래그 A-70
사용자 프로그램·설정 전송 플래그) A-124
상수 6-64, 6-85
  날짜 시각 형식 6-87
  날짜 형식 6-87
  문자열 형식 6-88
  부울 형식 6-85
  비트 열 형식 6-85
  시각 형식 6-87
  실수 형식 6-86
  열거 형식 6-89
  정수 6-86
  지속 시간 형식 6-86
상시 OFF 플래그 A-72, A-127
상시 ON 플래그 A-72, A-127
소프트웨어 구성 2-4
소프트웨어의 동작 2-5
슬레이브 노드 주소 중복 이상 A-85, A-147
슬레이브 어플리케이션 이상 A-85, A-148
슬레이브 이상 A-84, A-145
슬레이브 이상 테이블 A-84, A-146
슬레이브 초기화 이상 A-85, A-148
시각 형식 6-32, 6-45
시계 기능 8-3
시리얼 ID 기능 8-46
시리얼 옵션 보드 에러 Status A-75, A-133
시리얼 커뮤니케이션 유니트 0호기 포트 1/2용
  시리얼 포트 설정 변경 중 플래그 A-79, A-138
시리얼 커뮤니케이션 유니트 1-15호기 포트 1/2용
  시리얼 포트 설정 변경 중 플래그 A-79, A-138
시뮬레이션 기능 7-3
시뮬레이션 속도 7-7
시뮬레이션 프로그램 7-6
시뮬레이터의 기동과 정지 7-3
시스템 서비스 감시 설정 4-5
시스템 서비스 시간 부족 5-102
시스템 시각 8-5, A-65, A-115
시스템 정의 변수 2-14, 6-28
시스템 정의 변수 목록 A-64
시스템 정의 변수의 개별 사양 A-114
시퀀스 제어와 모션 제어의 구조 2-28
시퀀스 제어와 모션 제어의 동기 2-34
시퀀스 제어의 구조 2-30
실수 형식 6-32, 6-45

아

알고리즘 6-10, 6-18
암시적 캐스트 6-127
앱솔루트 엔코더 원점 위치 오프셋 전송 플래그. A-70, A-126
엣지 6-66

연결선 6-91
연속 트레이스 8-56
열거 형식(ENUM) 6-49
오버런 에러 발생 A-77, A-134
오프라인 디버그 7-3
온라인 A-101, A-167
온라인 연결이 가능한 구성 10-7
온라인 연결할 수 없는 구성 10-8
온라인 편집 8-54
옵션 보드 Status A-75, A-132
옵션 보드 에러 발생 A-75, A-133
옵션 보드 장착 A-75, A-132
옵션 보드 정상 동작 A-75, A-132
옵션 보드와의 액세스 2-20
완전 한정명 6-144
외부 변수 6-20
운전 모드 2-40
운전 모드 재기입 4-6
운전 시작 시 1주기 ON 플래그 6-8, A-72, A-127
운전 중 출력 기능 8-6
유니트 구성 3-5
유니트 버전 A-73, A-129
유니트·슬레이브 전송 플래그 A-70, A-125
유지 6-61
유지 변수의 데이터 형식을 변경했을 때의 변수 값 6-75
유지 조건 6-72
이름 공간 6-143
이름 공간의 사용 선언 6-145
이벤트 12-2
이벤트 로그 기능 8-66
이벤트 로그 설정 4-5
이벤트 발생원 8-68
이벤트 설정 테이블 8-76
이벤트 중요도 8-70
이벤트 중요도 변경 기능 8-80
이벤트 코드 8-69
이벤트 태스크 5-18, 5-36, 5-55
ActEventTask 명령 5-19, 5-37, 5-56
변수의 조건식 일치 5-21, 5-38, 5-57
설정 항목 5-89
이상 Status A-112
이전 태스크 실행 시간 5-7, A-65, A-115
이탈 명령 중 슬레이브 테이블 A-87, A-150
인라인 ST 6-95
인스턴스 6-14
인스턴스명 6-10
입력 데이터 1 무효 A-87, A-152
입력 데이터 2 무효 A-88, A-152
입력 데이터 무효 A-87, A-151
입력 변수 6-20
입출력 변수 6-20
입출력 응답 시간 5-112

자

장치 출력 유지 상태 A-73, A-130
장치 출력 유지 설정 6-70, A-73, A-130
전송 에러 발생 A-76, A-133
전원 OFF 발생 횟수 A-71, A-126
전원 OFF 유지 실패 플래그 A-71, A-127
전체 태그 데이터 링크 통신 중 A-102, A-168
접점 6-91
정상 대상 노드 정보 A-103, A-170
정수 형식 6-32, 6-45
정주기 태스크 5-53
정주기 태스크(실행 우선도 16, 17, 18)의 설정 항목 5-88
정주기 태스크(실행 우선도 5)의 설정 항목 5-88
정주기 태스크(실행 우선도 16, 17, 18) 5-17, 5-35
정주기 태스크(실행 우선도 5) 5-16
조건 플래그 6-136
조작 권한 8-42
조작 권한 인증 기능 8-42
주기/실행 조건 4-8
준 사용자 정의 변수 2-13, 6-28
중첩 계층 6-27
지속 시간 형식 6-32, 6-45
지원 소프트웨어 1-5, 1-6, 1-7, 1-8, 1-9, 1-10, 1-11, 1-12
진단 통계 정보 로그 실행 상태 플래그 A-88, A-153
진단 통계 정보 로그 이상 종료 플래그 A-88, A-153
진단 통계 정보 로그 저장 주기 A-88, A-153
진단 통계 정보 로그 허가 A-88, A-153

차

참조 태스크 5-90
초기 설정
EtherCAT 마스터 기능 모듈의 초기 설정 4-24
EtherNet/IP 기능 모듈의 초기 설정 4-25
고기능 유니트의 초기 설정 4-20
모션 제어 기능 모듈의 초기 설정 4-22
초기값 6-62
최대 NX 유니트 번호 A-80, A-139
최대 랙 번호 A-78, A-135
최대 슬롯 번호 A-78, A-135
축 3-18
축 그룹 변수 3-18, A-82, A-83, A-143
축 그룹 이상 Status A-82, A-142
축 및 축 변수의 작성과 사용 방법 3-20
축 변수 3-18, A-83, A-84, A-144
축 이상 Status A-82, A-142
출력 변수 6-20

카

카테고리 8-68
캐리 플래그 6-8, 6-136, A-72, A-127
캠 데이터 변수 2-13
컨트롤러 설정 4-4
컨트롤러 이벤트 8-67, 8-73
컨트롤러 이상 Status 5-100, A-67, A-117
코일 6-91

타

태그 데이터 링크 설정 이상 A-95, A-163
태그 데이터 링크 연결 확립 실패 A-96, A-163
태그 데이터 링크 통신 시작 스위치..... A-110, A-174
태그 데이터 링크 통신 이상 A-97, A-164
태그 데이터 링크 통신 정지 스위치..... A-110, A-175
태그 데이터 링크 통신 중 A-101, A-168
태그 해결 이상 A-97, A-165
태스크
I/O 리프레시에 대한 할당..... 5-78
NJ 시리즈의 사양..... 5-46
NX 시리즈의 사양..... 5-9, 5-29
프로그램의 할당..... 5-86
태스크 간 배타 제어 명령 5-93
태스크 간 배타 제어 설정 5-97
태스크 관련 명령..... 5-7
태스크 관련 시스템 정의 변수 5-7
태스크 사이에서 변수 값의 동시성을 확보 5-90
태스크 설계의 예..... 5-111
태스크 설정 4-7
태스크 실제 처리 시간..... 5-107
태스크 실행 상태 모니터링 4-11
태스크 실행 시간..... 5-107
태스크 실행 시간 모니터링 5-104
태스크 실행 시간 최대값..... 5-7, A-65, A-115
태스크 실행 시간 최소값..... 5-7, A-65, A-115
태스크 실행 시간 확인..... 5-108
태스크
실행 우선도 5-11, 5-31, 5-47
태스크 실행 중 플래그..... 5-7, A-65, A-115
태스크 실행 타임아웃..... 5-101
태스크 실행 타임아웃 시간 4-8
태스크 실행 횟수..... 5-7, 5-106, A-65, A-116
태스크 이름 4-7, 4-9
태스크 종류 5-15, 5-34, 5-52
태스크 주기 초과..... 5-100
태스크 주기 초과 감지 4-8
태스크 주기 초과 플래그..... 5-7, 5-100, 5-106, A-66, A-116
태스크 주기 초과 횟수..... 5-7, 5-100, A-66, A-116
태스크 타입 4-7
태스크에 할당할 수 있는 POU..... 5-86
태스크의 실행 상태 모니터링 5-103
통신 이상 슬레이브 테이블 A-85, A-149
통신 컨트롤러 이상..... A-84, A-92, A-146, A-158
통신 포트 1 BOOTP 서버 연결 이상 A-94, A-161
통신 포트 1 Ethernet 기본 설정 이상 A-93, A-159
통신 포트 1 IP 주소 설정 이상 A-93, A-159
통신 포트 1 IP 주소 중복 이상 A-94
통신 포트 1 MAC 주소 이상 A-92, A-157
통신 포트 1 온라인 A-101, A-167
통신 포트 1 이상 A-90, A-155
통신 포트 1 통신 컨트롤러 이상 A-93, A-158
통신 포트 2 BOOTP 서버 연결 이상 A-95, A-161
통신 포트 2 Ethernet 기본 설정 이상 A-93, A-159
통신 포트 2 IP 주소 설정 이상 A-94, A-160
통신 포트 2 IP 주소 중복 이상 A-94
통신 포트 2 MAC 주소 이상 A-92
통신 포트 2 온라인 A-101, A-167
통신 포트 2 이상 A-90, A-155
통신 포트 2 통신 컨트롤러 이상 A-93, A-158
통신 포트 이상 A-84, A-89, A-145, A-154
통전 시간 A-71, A-126
트레이스 실행 중 ON 플래그 8-61, A-74, A-131
트레이스 완료 시 ON 플래그 8-61, A-74, A-131
트레이스 트리거 모니터링 플래그..... 8-61, A-74, A-131
트레이스 파라메터 이상 플래그..... 8-61, A-74, A-131
트리거 트레이스..... 8-56

파

파라메터 6-10, 6-19
파생 데이터 형식..... 6-41
패리티 에러 발생..... A-76, A-134
패킷 데이터 파일 저장 중 A-87, A-151
패킷 모니터링 정지..... A-87, A-151
펑션 블록
ST 언어에서의 호출의 표현 6-11
기동 조건 6-16
내부 변수의 사양..... 6-12
변수 외부에서의 참조 6-17
인스턴스 6-14
펑션 블록(FB)
EN, ENO 사용/생략 가능 여부 6-24
파라메터 생략 시의 동작 6-24
파라메터 이상 시의 동작 6-25
펑션 블록(FB) 구성 6-9
펑션 블록(FB)의 세부 내용 6-9
펑션(FUN)
EN, ENO 사용/생략 가능 여부 6-24
ST 언어에서의 표현..... 6-19
구성..... 6-18
기동 조건 6-23
내부 변수의 사양..... 6-20
세부 내용 6-18
파라메터 생략 시의 동작 6-24
파라메터 이상 시의 동작 6-25
프라이머리 정주기 태스크 5-15, 5-34, 5-52
설정 항목 5-87
프레이밍 에러 발생..... A-76, A-134
프로그램 6-8
프로그램 기동 시 1주기 ON 플래그..... 6-8, A-72, A-128
프로그램 모드 2-40
프로그램 실행 순서..... 4-9, 5-86
프로그램 언어 6-90
프로그램 할당 설정..... 4-9
프로그램의 운전 시작 시 초기 상태..... 5-86
프로그램의 초기 상태 4-9
프로세스 데이터 송신 이상 A-85, A-147
프로세스 데이터 수신 타임아웃 이상..... A-85, A-147
프로세스 데이터 입력값 무효 이상..... A-85, A-148
프로세스 데이터 통신 Status A-87, A-151
프로세스 데이터 통신 이상 A-85, A-147

프로세스 데이터 통신 중 슬레이브 테이블..... A-87, A-150
프로젝트 파일 전체 보호 기능..... 8-39

하

하드웨어 개정..... A-73, A-129
현재값 변경..... 8-52
현재값 변경으로 외부로의 출력 상태를 변경하는 경우의
유의 사항 8-53

- 본 카탈로그에 기재되지 않은 조건이나 환경에서 사용하거나, 원자력 제어, 철도, 항공, 차량, 연소 장치, 의료용 기기, 오락 기계, 안전 기기, 기타 인명이나 재산에 큰 영향을 미칠 것이 예측되는 경우 등 특히 안전성이 요구되는 용도에 사용하는 경우에는 당사가 의도한 특별한 제품 용도인 경우나 특별한 합의가 있는 경우를 제외하고는 당사는 당사 제품에 대해 일체 보증하지 않습니다.
- 본 제품 중 외환 및 해외 무역법의 규정에 따라 수출 허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우에는 동법에 규정된 수출 허가, 승인(또는 직무 거래 허가)이 필요합니다.

Industrial Web ▶ http://www.ia.omron.co.kr

한국 오므론 제어기기 주식회사

00611 서울특별시 서초구 강남대로 465, B동 18층(서초동, 교보타워) TEL : 1522-8994 FAX : 02-3483-7787

카탈로그 번호 SBCA-467A-K6 2018년 8월 현재

© OMRON Corporation 2011-2018 All Rights Reserved.
예고없이 사양 등을 변경하는 경우가 있으므로 양해 부탁드립니다.