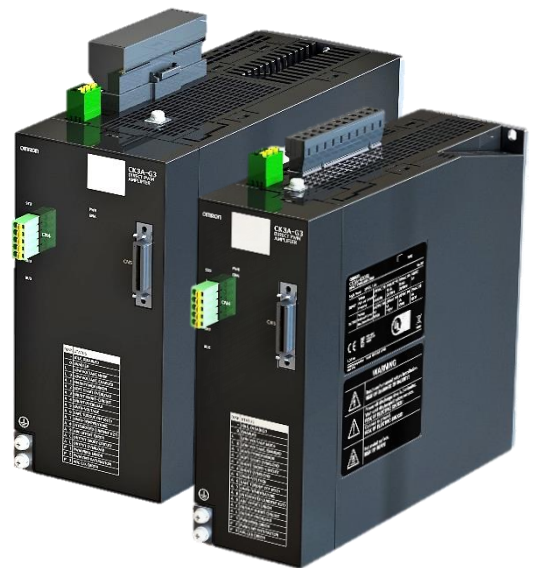


CK3A 시리즈 다이렉트 PWM 증폭기

사용 설명서

CK3A-G305L
CK3A-G310L



참고

- All rights reserved. OMRON의 사전 서면 허가 없이는 이 간행물의 어떤 부분도 기계적, 전자적, 복사, 녹음 등 어떤 형태나 수단으로 재생산하거나 검색 시스템에 저장하거나 전송할 수 없습니다.
- 여기에 포함된 정보의 사용과 관련하여 어떠한 특허 책임도 지지 않습니다. 또한 OMRON은 고품질 제품을 개선하기 위해 지속적으로 노력하고 있으므로 이 설명서에 포함된 정보는 예고 없이 변경될 수 있습니다.
- 이 설명서를 준비할 때 모든 예방 조치를 취했습니다. 그럼에도 불구하고 OMRON은 오류나 누락에 대해 책임을 지지 않습니다.
이 간행물에 포함된 정보의 사용으로 인해 발생하는 손해에 대해 어떠한 책임도 지지 않습니다.

상표

- Microsoft, Windows, Excel 및 Visual Basic은 미국 및 기타 국가에서 Microsoft Corporation의 등록 상표 또는 상표입니다.

이 문서에 나오는 다른 회사 이름과 제품 이름은 해당 회사의 상표 또는 등록 상표입니다.

저작권

- Microsoft 제품 스크린샷은 Microsoft Corporation의 허가를 받아 재인쇄되었습니다.

소개

CK3A 다이렉트 PWM 증폭기를 구입해 주셔서 감사합니다.

이 설명서에는 CK3A 다이렉트 PWM 증폭기의 올바른 장착, 설치, 배선, 구성 및 문제 해결에 필요한 정보가 포함되어 있습니다.

대상 독자

이 설명서는 전기 엔지니어 또는 이에 준하는 사람과 같이 전기 시스템에 대한 지식이 있어야 하는 다음과 같은 사람을 대상으로 합니다.

- FA 시스템 도입 담당자
- FA 시스템 설계 담당자
- FA 시스템 설치 및 유지관리 담당자
- FA 시스템 및 시설 관리 담당자

공지

이 사용 설명서에는 CK3A 증폭기의 올바른 사용에 중요한 정보가 포함되어 있습니다.

CK3A 증폭기를 사용하는 적절한 방법을 제대로 이해하려면 이 설명서를 충분히 읽으십시오.
이 설명서는 빠른 참조를 위해 편리한 장소에 보관하고 최종 사용자에게 제공되었는지 확인하십시오.

적용 대상 제품

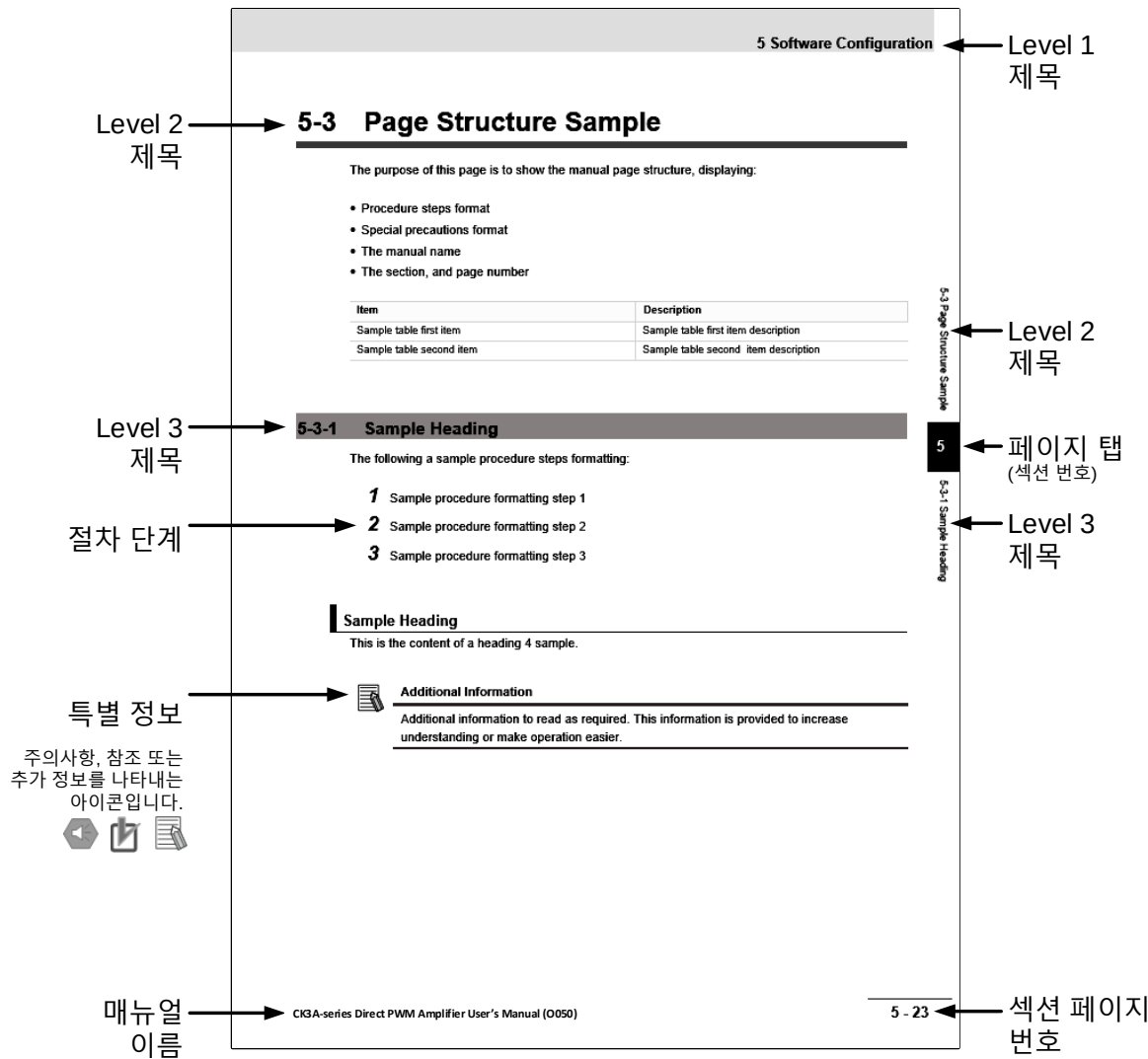
이 설명서에 해당하는 부품 번호는 다음과 같습니다.

- CK3A-G305L
- CK3A-G310L

설명서 구조

페이지 구조

이 설명서의 페이지 구조는 다음과 같습니다.



참고 이 페이지 그림은 샘플일 뿐입니다. 그 내용은 정보 자료가 아니며 설명서에 나와 있지 않습니다.

PMAC 스크립트

이 설명서에는 IDE 소프트웨어에서 사용할 수 있는 PMAC(언어) 스크립트 샘플이 포함되어 있습니다. 이러한 스크립트 샘플은 다음 형식으로 되어 있습니다.

```
GLOBAL MyVar1 // Global variable 1
GLOBAL MyVar2 = 0 // Global variable 2
```

특별 정보

이 설명서의 특별 정보는 다음과 같이 분류됩니다.



안전한 사용을 위한 주의사항

제품의 안전한 사용을 보장하기 위해 해야 할 일과 하지 말아야 할 일에 대한 주의사항입니다.



올바른 사용을 위한 주의사항

적절한 작동 및 성능을 보장하기 위해 해야 할 일과 하지 말아야 할 일에 대한 주의사항입니다.



추가 정보

필요에 따라 읽을 추가 정보입니다. 이 정보는 이해도를 높이거나 쉽게 작동하기 위해 제공됩니다.

이 설명서의 섹션

		1
		2
1	증폭기 소개	3
2	사양	4
3	장착 및 설치	5
4	배선	6
5	소프트웨어 구성	A
6	문제 해결	
A	부록	

목차

소개	1
대상 독자.....	1
공지	1
적용 대상 제품	1
설명서 구조	2
페이지 구조	2
PMAC 스크립트	2
특별 정보.....	3
이 설명서의 섹션	5
목차	6
이용약관 동의.....	10
보증, 책임의 제한.....	10
적용 고려 사항	11
면책 조항.....	11
안전 주의사항.....	12
주의사항 정보의 정의	12
기호 정의.....	12
주의사항 정보	13
안전한 사용을 위한 주의사항	17
올바른 사용을 위한 주의사항	20
포장을 뜯 후 확인해야 할 사항	25
명판 위치.....	25
부속품	26
규정 및 표준	27
EU 지침 준수.....	27
UL 지침 준수	27
KC 표준 준수.....	27
버전	28
하드웨어 버전	28
펌웨어 버전	28
관련 설명서	29
용어	30
설명서 개정 이력	31

섹션 1 증폭기 소개

1-1 개요.....	1-2
1-1-1 증폭기 기능	1-2
1-1-2 일반 구성	1-3
1-1-3 부품 이름 및 위치	1-4
1-1-4 부품 기능	1-8
1-2 작동 절차	1-10

1-2-1	준비	1-10
1-2-2	안전	1-10
1-2-3	장착 및 배선	1-10
1-2-4	안전 테스트 및 전원 켜기	1-10
1-2-5	증폭기용 컨트롤러 설정	1-11
1-2-6	테스트 실행	1-11

섹션 2 사양

2-1	증폭기 사양.....	2-2
2-1-1	일반/기계 사양.....	2-2
2-1-2	환경 사양.....	2-2
2-1-3	전기 사양.....	2-3
2-1-4	성능	2-4
2-1-5	회생 에너지 흡수.....	2-5
2-2	기능 및 데이터 보고.....	2-9
2-2-1	증폭기 기능	2-9
2-2-2	데이터 보고	2-10
2-2-3	상태 비트.....	2-10
2-3	부품 번호 지정	2-11
2-3-1	증폭기 부품 번호.....	2-11

섹션 3 장착 및 설치

3-1	외부 및 장착 치수.....	3-2
3-1-1	CK3A-G305L 치수 및 장착.....	3-2
3-1-2	CK3A-G310L 치수 및 장착.....	3-3
3-2	설치.....	3-4
3-2-1	환기	3-4
3-2-2	패널 간격	3-4
3-2-3	장착 방향.....	3-6
3-2-4	설치 조건.....	3-7
3-2-5	장치에 이물질 유입 방지	3-7

섹션 4 배선

4-1	주 회로 전원 및 셉트(CN1).....	4-3
4-1-1	커넥터 핀아웃.....	4-3
4-1-2	와이어 크기	4-4
4-1-3	배선 예	4-5
4-2	로직 전원 공급(CN2).....	4-9
4-2-1	커넥터 핀아웃.....	4-9
4-2-2	와이어 크기	4-9
4-2-3	배선 예	4-9
4-3	모터 연결(CN3).....	4-10
4-3-1	커넥터 핀아웃.....	4-10
4-3-2	와이어 크기	4-11
4-3-3	배선 예	4-12
4-4	안전 토크 꺼짐 STO(CN4)	4-13
4-4-1	커넥터 핀아웃.....	4-13
4-4-2	와이어 크기	4-13

4-4-3	배선 예	4-13
4-5	다이렉트 PWM 연결(CN5)	4-16
4-5-1	커넥터 핀아웃	4-16
4-6	전체 배선도 예	4-17
4-6-1	CK3A-G305L	4-17
4-6-2	CK3A-G310L	4-18
4-6-3	증폭기 결함 릴레이	4-19
4-7	EMC 지침을 준수하는 배선	4-20
4-7-1	주변 장비 연결 예	4-20
4-8	주변 구성 요소 선택	4-22
4-8-1	주 회로 AC 전원 공급	4-22
4-8-2	주 회로 DC 전원 공급 장치(저전압 모드)	4-22
4-8-3	로직 전원 공급	4-22
4-8-4	노이즈 필터	4-23
4-8-5	전압 서지 흡수기	4-23
4-8-6	배선용 차단기(MCCB, Molded Case Circuit Breaker)	4-23
4-8-7	외부 셉트 저항	4-24
4-8-8	다이렉트 PWM 케이블	4-24
4-8-9	호환 모터	4-24

섹션 5 소프트웨어 구성

5-1	소프트웨어 구성 개요	5-3
5-1-1	컨트롤러 증폭기 인터페이스	5-3
5-1-2	요약	5-3
5-1-3	대상 사용자	5-4
5-1-4	대상 하드웨어	5-4
5-1-5	Gate3 별칭 이름	5-4
5-2	기본 구성	5-5
5-2-1	PWM 주파수	5-5
5-2-2	Gate3 설정	5-5
5-2-3	채널 설정	5-5
5-2-4	모터 설정	5-6
5-3	ADC 스트로브 워드의 세부 정보	5-7
5-3-1	개요	5-7
5-3-2	명시적 주소	5-8
5-4	증폭기 기능의 세부 정보	5-9
5-4-1	기능	5-9
5-4-2	버스 방전 제어	5-9
5-4-3	결함 삭제	5-11
5-4-4	팬 제어	5-11
5-4-5	다이내믹 브레이크 제어	5-12
5-4-6	저전압(LV) 모드	5-13
5-5	ADC 레지스터의 세부 정보	5-14
5-5-1	ADC A의 정보	5-14
5-5-2	ADC B의 정보	5-14
5-5-3	ADC 레지스터 명시적 주소	5-15
5-6	상태 비트의 세부 정보	5-16

5-6-1	상태 비트 세트 1	5-16
5-6-2	상태 비트 세트 2	5-17
5-6-3	상태 비트 세트 3	5-18
5-7	증폭기 데이터의 세부 정보	5-19
5-7-1	개요 및 예	5-19
5-8	CK3A 샘플 PLC.....	5-20
5-8-1	샘플 PLC 구현	5-20
5-8-2	지연 타이머 서브루틴	5-21
5-8-3	사용자 공유 메모리에 대한 설명	5-22
5-8-4	CK3A 샘플 정의 4 모터	5-23
5-8-5	CK3A 샘플 PLC 4 모터	5-25
5-8-6	CK3A 샘플 정의 8 모터	5-27
5-8-7	CK3A 샘플 PLC 8 모터	5-31
5-8-8	전원을 켤 때 CK3A PLC 활성화	5-33
5-8-9	CK3A PLC 사용	5-34

섹션 6 문제 해결

6-1	LED 표시등	6-2
6-2	표시 코드	6-3
6-2-1	일반 모드 작동	6-3
6-2-2	오류 코드	6-3
6-2-3	문제 해결 오류 코드	6-5
6-2-4	결함 임계값	6-8
6-2-5	증폭기 작동 문제 해결	6-9

부록

A-1	IDE를 사용한 자동 모터 설정	A-2
A-1-1	IDE 시스템 설정	A-2
A-2	Power PMAC3 수동 모터 설정	A-11
A-2-1	IDE 프로젝트 생성	A-11
A-2-2	기본 최적화, 시스템 및 게이트 설정	A-15
A-2-3	인코더 진행 위치, 확인	A-16
A-2-4	모터 설정 및 시운전	A-21
A-2-5	모터 작동 방향 반전	A-45
A-3	Gate3 "스크립트" 모터 설정 샘플.....	A-46
A-3-1	직각위상 인코더가 있는 로터리 서보 모터	A-47
A-3-2	OMRON G5 서보 모터	A-49
A-3-3	OMRON 1S 서보 모터(ACC-84E).....	A-52
A-3-4	정현파 인코더가 있는 리니어 서보 모터	A-54
A-3-5	BiSS 인코더가 포함된 리니어 서보 모터(ACC-84E).....	A-56
A-4	Gate1 "스크립트" 모터 설정 샘플.....	A-58
A-4-1	Gate1이 있는 Power PMAC(ACC-24E2)	A-59
A-4-2	Gate1이 있는 Turbo PMAC(ACC-24E2).....	A-61
A-5	Gate1용 CK3A 샘플 PLC	A-63
A-5-1	Gate1이 있는 Power PMAC.....	A-63
A-5-2	Gate1이 있는 Turbo PMAC.....	A-66

이용약관 동의

보증, 책임의 제한

보증

● 독점 보증

Omron의 배타적 보증은 제품이 Omron에서 판매한 날로부터 12개월 동안(또는 Omron에서 서면으로 명시한 기타 기간) 부품 및 제작상의 결함에 대해 적용됩니다. Omron은 이 외 모든 명시적, 묵시적 보증을 인정하지 않습니다.

● 제한 사항

OMRON은 제품의 특정 목적 상 사용에 대해 권리 미 침해, 상품성 및 적합성에 대해 명시적이든 묵시적이든 어떠한 보증이나 책임을 지지 않습니다. 구매자는 제품이 의도된 용도의 요구 사항을 적절하게 충족할 것이라고 단독으로 결정했음을 인정합니다.

또한 Omron은 제품 또는 기타 지적 재산권 침해에 근거한 청구 또는 비용에 대한 모든 유형의 보증 및 책임을 인정하지 않습니다.

● 구매자 구제

Omron의 유일한 의무는 Omron의 선택에 따라 (i) 비 준수 제품을 교체(제거 또는 교체에 대한 인건비 구매자 부담), (ii) 비 준수 제품을 수리하는 것입니다. 또한 (iii) 비 준수 제품의 구매 가격과 동일한 금액을 구매자에게 상환하거나 크레딧을 제공합니다. 단, Omron의 분석 결과 제품이 적절하게 취급, 저장, 설치 및 유지되고 오염, 남용, 오용 또는 부적절한 수정의 대상이 아님이 확인되지 않는 한 Omron은 제품에 대한 보증, 수리, 배상 또는 기타 청구 또는 비용에 대해 어떠한 책임도 지지 않습니다. 구매자에 의한 모든 제품의 반품은 선적 전에 Omron에 의해 서면으로 승인되어야 합니다. Omron사는 전기 또는 전자 구성 요소, 회로, 시스템 어셈블리 또는 기타 부품이나 물질 사용 또는 특정 환경에서 제품을 사용하여 발생하는 적합성 또는 부적합 또는 결과에 대해 책임을 지지 않습니다. 구두 또는 서면으로 제공되는 모든 조언, 권고 또는 정보는 위의 보증에 대한 수정 또는 추가로 해석되어서는 안 됩니다.

게시된 정보는 <http://www.omron.com/global/>을 참조하거나 Omron 담당자에게 문의하십시오.

책임의 제한 등

OMRON 회사는 이러한 청구가 계약, 보증, 과실 또는 엄격한 책임에 근거하는지 여부에 관계없이 제품과 관련된 특수, 간접, 부수적 또는 결과적 손해, 이익 손실 또는 생산 또는 상업적 손실에 대해 책임을 지지 않습니다.

또한 어떠한 경우에도 Omron사의 책임은 책임이 주장되는 제품의 개별 가격을 초과하지 않습니다.

적용 고려 사항

사용의 적합성

Omron 회사는 구매자의 제품 적용 또는 사용에서 제품 조합에 적용되는 표준, 코드 또는 규정의 준수에 대해 책임을 지지 않습니다. 구매자의 요청에 따라 Omron은 제품에 적용되는 등급 및 사용 제한을 식별하는 해당 제3자 인증 문서를 제공합니다. 이 정보만으로는 최종 제품, 머신, 시스템 또는 기타 적용이나 사용과 함께 제품의 적합성을 완전히 판단하는 데 충분하지 않습니다. 구매자는 구매자의 용도, 제품 또는 시스템과 관련하여 특정 제품의 적합성을 결정할 책임이 전적으로 있습니다. 구매자는 제품 사용 및 적용 대한 모든 경우에 책임을 집니다.

전체 시스템이 위험에 대해 대응할 수 있도록 설계되지 않은 채 생명이나 재산에 심각한 위험을 가져올 수 있는 환경에서 제품을 사용하지 마십시오. OMRON 제품이 전체 장비 또는 시스템 내에서 의도된 용도에 적합한 등급을 받고 안전하게 설치되었는지 확인 후 사용하십시오.

프로그램형 제품

Omron 회사는 프로그램형 제품의 사용자 프로그래밍 또는 그 결과에 대해 책임을 지지 않습니다.

면책 조항

성능 데이터

Omron 회사의 웹 사이트, 카탈로그 및 기타 자료에 제시된 데이터는 사용자가 적합성을 결정하는 데 지침으로 제공하지만 보증을 하지는 않습니다. 이는 Omron의 테스트 조건 결과를 나타낼 수 있으며 사용자는 이를 실제 적용 요구 사항과 연관시켜야 합니다. 실제 성능은 Omron의 보증 및 책임 제한의 적용을 받습니다.

사양 변경

제품 사양 및 부속품은 개선 및 기타 이유로 언제든지 변경될 수 있습니다. 게시된 등급이나 기능이 변경되거나 상당한 구조 변경이 있을 때 부품 번호를 변경하는 것이 당사 관행입니다. 단, 제품의 일부 사양은 예고 없이 변경될 수 있습니다. 확실하지 않은 경우 용도에 맞게 주요 사양을 수정하거나 설정하기 위해 특수 부품 번호를 할당할 수 있습니다. 구매한 제품의 실제 사양을 확인하려면 언제든지 Omron 대리점에 문의하십시오.

오류 및 누락

Omron 회사에서 제공하는 정보는 확인되었으며 정확한 것으로 간주됩니다. 그러나 사무적, 인쇄적 또는 교정 오류 또는 누락에 대한 책임은 지지 않습니다.

안전 주의사항

CK3A 다이렉트 PWM 증폭기를 안전하고 올바르게 사용하려면 설치 및 배선 전에 안전 주의사항 섹션을 주의 깊게 충분히 읽어야 합니다. 이 사용 설명서는 최종 사용자에게 제공되어야 하며 참조를 위해 쉽게 접근할 수 있는 위치에 보관해야 합니다.

주의사항 정보의 정의

이 설명서에 설명된 안전 주의사항은 안전을 위해 매우 중요합니다. 모든 안전 주의사항에 제공된 정보를 항상 읽고 주의하십시오.

다음 표기법은 CK3A 다이렉트 PWM 증폭기의 안전한 사용을 보장하고 필요한 예방 조치를 제공하는 데 사용됩니다.

 WARNING	예방하지 않을 경우 사망 또는 심각한 부상을 초래할 수 있는 잠재적으로 위험한 상황을 나타냅니다. 또한 심각한 재산 피해가 발생할 수 있습니다.
--	---

기호 정의

이 설명서에는 다음 기호가 포함되어 있습니다.

	원과 슬래시 기호는 수행해서는 안 되는 작업을 나타냅니다. 구체적인 작업은 원에 표시되고 텍스트로 설명됩니다. 이 예는 분해가 금지됨을 나타냅니다.
	삼각형 기호는 주의사항(경고 포함)을 나타냅니다. 구체적인 작업은 삼각형에 표시되고 텍스트로 설명됩니다. 이 예는 감전 주의를 나타냅니다.
	삼각형 기호는 주의사항(경고 포함)을 나타냅니다. 구체적인 작업은 삼각형에 표시되고 텍스트로 설명됩니다. 이 예는 화재 주의를 나타냅니다.
	삼각형 기호는 주의사항(경고 포함)을 나타냅니다. 구체적인 작업은 삼각형에 표시되고 텍스트로 설명됩니다. 이 예는 화상 주의를 나타냅니다.
	삼각형 기호는 주의사항(경고 포함)을 나타냅니다. 구체적인 작업은 삼각형에 표시되고 텍스트로 설명됩니다. 이 예는 일반적인 주의를 나타냅니다.
	채워진 원 기호는 반드시 수행해야 하는 작업을 나타냅니다. 구체적인 작업은 원에 표시되고 텍스트로 설명됩니다. 이 예는 수행해야 하는 작업에 대한 일반적인 주의를 보여줍니다.

주의사항 정보

이 설명서에 포함된 그림은 때때로 세부 사항을 보여주기 위해 덮개와 안전 실드가 없는 상태를 나타냅니다. 이 제품을 사용할 때는 반드시 명시된 대로 덮개와 실드를 설치하고 이 설명서에 따라 제품을 사용하십시오.

운반 및 포장 풀기



WARNING

증폭기를 운반할 때 케이블을 잡고 운반하지 마십시오.
부상, 장치 손상 또는 오작동이 발생할 수 있습니다.



증폭기를 밟거나 무거운 물건을 올려놓지 마십시오.
부상, 장치 손상 또는 오작동이 발생할 수 있습니다.



설치, 배선 및 유지보수



WARNING

이 설명서의 지침에 따라 이 증폭기를 전원 공급 장치, 모터 및 기타 장비에 올바르게 연결하십시오. 그렇게 하지 않으면 심각한 부상을 입을 수 있습니다.



적절한 외부 셉트 저항을 사용하십시오. 외부 셉트 저항을 사용할 때 안전을 위해 온도 센서 등의 외부 보호 장치를 설치하십시오.
그렇게 하지 않으면 소손의 위험이 있습니다.



제품 및 주변 방비는 금속 등 불연성 재질 위에 설치하십시오.
그렇게 하지 않으면 화재가 발생할 수 있습니다.



단자대와 커넥터 배선 및 삽입을 올바르게 수행하려면 이 설명서의 지침을 따르십시오.
전원 공급 장치를 켜기 전에 모든 배선 및 커넥터 삽입을 다시 확인하십시오.
그렇게 하지 않으면 감전, 화재, 장비 손상, 오작동 또는 부상을 초래할 수 있습니다.



케이블을 손상시키거나 당기거나 세게 구부리거나 과도한 힘을 주거나 무거운 물건을 올려놓지 마십시오. 오작동이나 화상을 입을 수 있습니다.



증폭기 또는 주변 장비 근처에 가연성 물질을 두지 마십시오. 화재가 발생할 수 있습니다.	
저전압 단락 와이어(CN1)가 설치된 경우 증폭기에 AC 전원을 인가하지 마십시오. 손상, 오작동 또는 부상의 원인이 될 수 있습니다.	
증폭기(및 모터)가 100Ω 이하의 연결을 사용하여 접지에 연결되어 있는지 확인하십시오. 그렇게 하지 않으면 감전될 수 있습니다.	
증폭기 덮개를 제거하지 마십시오. 감전될 수 있습니다.	
적절한 공구를 사용하여 단자와 커넥터를 배선하십시오. 사용하기 전에 단락이 없는지 확인하십시오. 그렇게 하지 않으면 감전될 수 있습니다.	
모터 케이블의 프레임 접지선을 증폭기의 ⓪ 또는 FG에 단단히 연결하십시오. 그렇게 하지 않으면 감전될 수 있습니다.	
외부 배선의 합선 및 제품의 고장을 방지하기 위해 퓨즈, 배선용 차단기, 누전차단기 등의 안전 조치를 마련해 두십시오. 그렇게 하지 않으면 화재가 발생할 수 있습니다.	

소프트웨어 구성



WARNING

컨트롤러의 모터 관련 보호 매개변수, 특히 Motor[x].I2tSet, Motor[x].I2tTrip 및 Motor[x].MaxDac를 포함한 I2T 열 보호가 올바르게 설정되었는지 확인하십시오.
그렇게 하지 않으면 모터가 타거나 화재가 발생하거나 심각한 부상을 입을 수 있습니다.



주 전원이 인가된 상태에서



WARNING

배선 작업이나 검사 작업은 주 회로 전원을 차단하고 5분 이상 경과 후 실시하십시오.
그렇게 하지 않으면 감전되거나 화상을 입을 수 있습니다.



고장 안전 조치



WARNING

제품의 오작동이나 작동에 영향을 미치는 기타 외부 요인으로 인해 증폭기 오류 또는 이상이 발생할 경우 시스템의 안전을 보장하기 위해 외부 회로에서 주 회로 전원을 차단하는 등의 안전 조치를 취하십시오. 그렇게 하지 않으면 심각한 사고가 발생할 수 있습니다.



외부 제어 회로에는 비상 정지, 인터록, 제한 회로 및 이와 유사한 안전 조치를 취해야 합니다. 그렇게 하지 않으면 심각한 사고가 발생할 수 있습니다.



신호선 단선, 순간 정전 등의 원인으로 신호가 부정확하거나 누락되거나 이상 신호가 발생하는 경우 안전을 확보하기 위한 고장 안전 조치를 취해야 합니다. 그렇게 하지 않으면 심각한 사고가 발생할 수 있습니다.



수직 축의 경우 브레이크 또는 카운터 밸런스 메커니즘을 사용하여 서보 제어가 정지될 때 넘어지지 않도록 하십시오. 그렇게 하지 않으면 심각한 부상을 입을 수 있습니다.



테스트 실행



WARNING

이 설명서 및 관련 설명서의 지침에 따라 증폭기를 적절하게 프로그래밍하십시오. 그렇게 하지 않으면 장비가 손상될 수 있습니다.



Power PMAC 컨트롤러를 교체하거나 구성을 변경할 때(IDE 소프트웨어에서) 항상 안전을 확인하십시오. 장치 또는 장비는 제품의 작동 모드에 관계없이 예기치 않은 작동을 할 수 있습니다.
그렇게 하지 않으면 장비 손상, 오작동 또는 부상을 초래할 수 있습니다.



작동 시



WARNING

머신 작동 시 작동 영역에 들어가지 마십시오.
그렇게 하지 않으면 부상을 입을 수 있습니다.



전원이 공급되는 동안 또는 전원이 꺼진 후 5분 이상은 뜨겁기 때문에 증폭기 히트 싱크,
션트 저항 또는 모터를 만지지 마십시오.
화재가 발생하거나 화상을 입을 수 있습니다.



정격 전압의 지정된 전원이 공급되도록 적절한 조치를 취하십시오. 전원 공급이 불안정한
장소에서는 특히 주의하십시오.
그렇게 하지 않으면 고장이 발생할 수 있습니다.



순간 정전 후 전원이 복구되면 장비가 갑자기 다시 작동될 수 있습니다. 전원을 복구할
때는 장비에 접근하지 마십시오. 그렇게 하지 않으면 부상을 입을 수 있습니다.
장비가 다시 시작될 때 주변 사람들의 안전을 확보할 수 있는 조치를 취하십시오.



주 회로 전원이 켜져 있는 동안에는 증폭기의 연결을 연결하거나 분리하지 마십시오.
고장, 장비 손상 또는 심각한 부상을 초래할 수 있습니다.



모터 커넥터가 연결되지 않은 상태에서 컨트롤러가 모터 명령을 전송하는 경우 증폭기에
오류가 발생하지 않습니다. 컨트롤러가 모터 명령을 전송하지 않는지 확인하거나 주 회로
전원이 꺼질 때까지 모터 커넥터를 연결하지 마십시오.
고장, 장비 손상 또는 심각한 부상을 초래할 수 있습니다.



소프트 스타트 중에는 모터를 활성화하지 마십시오. 주 AC 전원을 켜 후 모터를
활성화하기 전에 최소 5초 정도 기다리십시오.
고장, 장비 손상 또는 심각한 부상을 초래할 수 있습니다.



안전한 사용을 위한 주의사항

운송

- 장치를 떨어뜨리거나 비정상적인 진동이나 충격을 가하지 마십시오.
장치 오작동, 화재 또는 감전이 발생할 수 있습니다.

장착

- 설치 방향 및 주변과의 간격을 반드시 준수하십시오.
- 사용하기 전에 단자대, 커넥터 및 기타 잠금 장치의 래치가 올바르게 고정되었는지 확인한 후 사용하십시오.

설치

- 흡입구 또는 배기구를 막지 마십시오. 오작동, 작동 중지 또는 장비 손상이 발생할 수 있습니다.
- 이물질이 증폭기에 들어가지 않도록 하십시오.
- 다음 장소에서 작동하거나 보관하지 마십시오.
 - 직사광선에 노출될 수 있는 장소
 - 규격에 지정된 범위를 벗어나는 온도에 노출될 수 있는 장소
 - 규격에 지정된 범위를 벗어나는 습도에 노출될 수 있는 장소
 - 결로가 발생할 수 있는 장소(예: 심각한 온도 변화)
 - 부식성 또는 인화성 가스가 발생할 수 있는 장소
 - 먼지(특히 철가루)나 염분에 노출될 수 있는 장소
 - 물, 기름, 화학 물질에 노출될 수 있는 장소
 - 충격 또는 진동이 발생할 수 있는 장소

이런 경우 감전, 화재, 장비 손상 또는 오작동을 초래할 수 있습니다.

배선

- 단자대와 커넥터 배선 및 삽입을 올바르게 수행하려면 이 설명서의 지침을 따르십시오.
전원 공급 장치를 켜기 전에 모든 배선 및 커넥터 삽입을 다시 확인하십시오.
- 케이블을 잡아당기거나 자연 한계를 초과하여 구부리지 마십시오.
- 케이블이나 다른 선 위에 무거운 물체를 올려놓지 마십시오. 그렇게 하면 케이블이 파손될 수 있습니다.
- 전선 조각, 부스러기 또는 기타 이물질이 증폭기에 들어가지 않도록 하십시오. 그렇지 않으면 증폭기 소손, 고장 또는 오작동이 발생할 수 있습니다. 특히 배선 작업을 수행할 때 증폭기를 덮거나 다른 적절한 조치를 취하십시오.
- 케이블 트랙 또는 적절한 메커니즘을 사용하여 시스템의 움직이는 부분과 고정된 부분을 분리하십시오.
- 접촉기나 다른 전기 릴레이 장치 없이 증폭기를 모터에 연결하십시오.

전원 공급 장치 설계

- 전원 공급 장치는 이 설명서에 지정된 정격 용량 범위 내에 있어야 합니다.
- 외부 차단기를 설치하고 단락 및 과전류에 대한 기타 안전 조치를 취하십시오.
- 이 설명서에 지정된 범위를 초과하는 전압을 인가하지 마십시오.

모터 선택

- 증폭기에 연결할 모터를 선택하기 전에 사양을 주의 깊게 확인하십시오.

소프트웨어 구성

- 이 설명서의 소프트웨어 구성 섹션에 설명된 컨트롤러의 증폭기별 매개변수가 올바르게 설정되어 있는지 확인하십시오.
- 컨트롤러의 모터별 보호 매개변수가 올바르게 설정되었는지 확인하십시오. 특히 I2T 열 보호 설정을 확인하십시오.

전원 켜기

- 전원을 켜기 전에 단자대, 커넥터 및 잠금 장치 또는 나사 조임 장치가 있는 기타 항목이 제자리에 올바르게 잠겨 있는지 확인하십시오.
- 다음은 증폭기를 켜는 올바른 절차입니다.
 - 1** 24VDC 로직 전원을 먼저 켭니다(항상 주 회로 전원보다 먼저).
 - 2** 증폭기가 부팅될 때까지 최소 10초 동안 기다립니다.
 - 3** 주 회로 전원을 켭니다(예: 200VAC).
 - 4** 최소 5초 동안 기다립니다.
- 증폭기가 부팅 중일 때 주 전원을 켜거나 컨트롤러에서 증폭기에 명령을 내리지 마십시오.
- 주 회로 전원을 켜기 전에 컨트롤러가 증폭기에 모터 명령을 보내지 않는지 확인하십시오.
- 주 회로 전원을 켜 후 컨트롤러에서 증폭기로 모터 명령을 보내기 전에 최소 5초 동안 기다리십시오.

위의 모든 작업을 수행하지 않으면 심각한 부상, 장비 손상 또는 오작동이 발생할 수 있습니다.

전원이 켜져 있는 동안

- 전원이 공급되는 동안 또는 전원이 꺼진 후 5분 이상은 뜨겁기 때문에 증폭기 히트 싱크, 셉트 저항 또는 모터를 만지지 마십시오. 화재가 발생하거나 화상을 입을 수 있습니다.
- 오류가 발생하면 오류의 원인을 제거하고 안전을 확인한 후 재설정 및 시스템 재시작을 수행하십시오. 그렇게 하지 않으면 손상, 오작동 또는 부상의 원인이 될 수 있습니다.

전원 끄기

- 다음은 증폭기를 끄는 올바른 절차입니다.
 - 1** 항상 24VDC 로직 전원보다 주 회로 전원(예: 200VAC)을 먼저 끕니다.
 - 2** 콘덴서 에너지를 방전하도록 증폭기가 프로그래밍된 경우(컨트롤러의 스트로브 워드로 설정) 최소 3초 동안 기다리십시오.
 - 3** 24VDC 로직 전원을 끕니다.
- 다음 작업을 수행하기 전에 항상 전원을 끄십시오.
 - a) 장치 장착 또는 제거.
 - b) 케이블 연결 또는 시스템 배선.
 - c) 단자대 또는 커넥터 연결 또는 분리.

위의 모든 작업을 수행하지 않으면 심각한 부상, 장비 손상 또는 오작동이 발생할 수 있습니다.

유지보수

- 부식성 화학 물질을 사용하여 증폭기를 청소하지 마십시오. 증폭기의 고장 또는 오작동이 발생할 수 있습니다.
- 현지 조례에 따라 제품을 폐기하십시오.

올바른 사용을 위한 주의사항

일반 주의사항

- 다음 위치에 증폭기를 설치할 때는 적절하고 충분한 조치를 취하십시오.
 - a) 강한 고주파 노이즈가 발생하는 위치
 - b) 정전기 또는 기타 노이즈가 발생하는 위치
 - c) 강한 전자기장의 영향을 받는 위치
 - d) 방사능에 노출될 수 있는 위치
 - e) 전원선에 가까운 위치

설치 및 장착

- 증폭기를 올바르게 설치하려면 이 설명서의 지침을 따르십시오.
- 증폭기 장착 시 지정된 토크로 조이십시오.
- 제품 및 주변 장비의 장착 나사를 규정된 토크로 조이십시오.
- 다음 위치에 증폭기를 설치할 때는 적절하고 충분한 조치를 취하십시오.
 - a) 강한 고주파 노이즈가 발생하는 위치
 - b) 정전기 또는 기타 노이즈가 발생하는 위치
 - c) 강한 전자기장의 영향을 받는 위치
 - d) 방사능에 노출될 수 있는 위치
 - e) 전원선에 가까운 위치
- 증폭기를 열원에서 멀리 설치하고 적절하게 환기시키십시오.
- 제품을 설치 또는 교체한 후에는 작동 상태를 충분히 확인한 후 실제 작동으로 넘어가십시오.

위의 모든 작업을 수행하지 않으면 부상, 장비 손상, 오작동 또는 화재가 발생할 수 있습니다.

배선

- 케이블을 올바르게 안전하게 배선하십시오.
- 배선 및 조립을 주의하여 수행하십시오.
- 압착 단자를 사용하여 나사식 단자대를 배선하십시오. 노출된 연선을 단자대에 직접 연결하지 마십시오.
- 항상 설명서에 명시된 전원 전압을 사용하십시오.
- 안전 기능을 포함하는 시스템을 구축할 때 관련 안전 표준 및 사용 설명서의 모든 관련 정보를 이해하고 표준에 맞게 시스템을 설계하십시오.
- 증폭기에서 메거 테스트(절연 저항 측정)를 시도하기 전에 증폭기의 모든 연결을 분리하십시오. 증폭기에서 절연 내력 테스트를 수행하지 마십시오.
- 나사, 금속 조각, 오일과 같은 전도성 또는 가연성 이물질을 증폭기 및 커넥터에 가까이 두지 마십시오. 증폭기 상단에 있는 커넥터에 특히 주의하십시오.
- 커넥터를 설치하거나 제거할 때 보호 장비를 착용하십시오. 특히 주 회로 커넥터에 주의하십시오. 커넥터 오프너의 돌출부가 하사점에 도달한 후에는 힘을 가하지 마십시오. (100N 이상의 힘을 가하지 마십시오).
- 배선 단자 및 커넥터에 무리한 힘을 가하지 마십시오.

위의 모든 작업을 수행하지 않으면 부상, 장비 손상, 오작동 또는 화재가 발생할 수 있습니다.

전원 켜기/끄기

- 로직 전원이 꺼져 있으면 주 회로 전원을 켜고 끄지 마십시오.
- 로직 전원을 켜 후 10초간 기다린 후 주 회로 전원을 켜십시오.

조정

- 테스트 작업 중 컨트롤러 또는 컴퓨터와의 통신에 문제가 발생하면 모터를 중지할 수 없습니다. 모터가 반드시 정지할 수 있도록 외부에 설치된 비상 정지 스위치 등을 범용 입력의 Error Stop Input에 연결하십시오.
- 컨트롤러에서 매개변수를 극단적인 값으로 조정하지 마십시오. 이로 인해 작동이 불안정해질 수 있습니다.

위의 모든 작업을 수행하지 않으면 부상, 장비 손상, 오작동 또는 화재가 발생할 수 있습니다.

작동

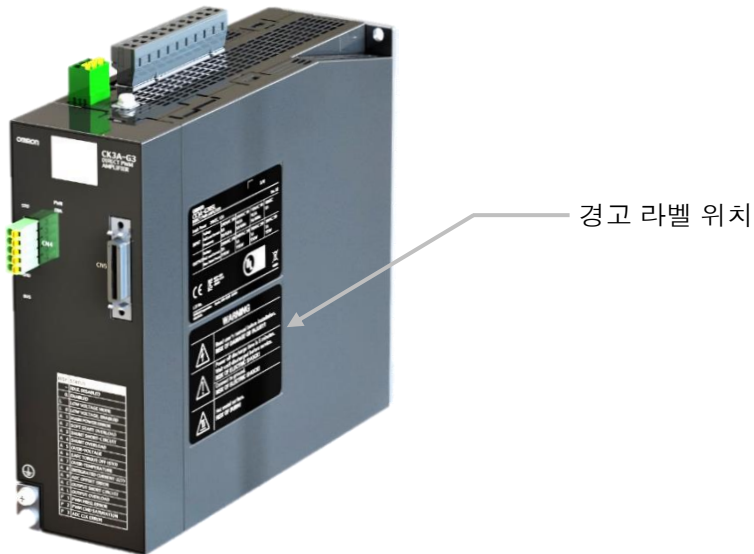
- 커넥터가 단단히 꽂혀 있고 단자대가 단단히 조여져 있는지 확인하십시오.
- 안전을 위해 장비에 적절한 정지 장치를 설치하십시오. 다이내믹 브레이크 메커니즘은 정지 장치가 아닙니다.
- 머신 외부에 즉시 정지 장치를 설치하여 즉시 작동을 정지하고 전원을 차단할 수 있도록 하십시오.
- 지진, 낙뢰, 홍수, 정전 후에는 반드시 안전 점검을 실시하십시오.
- 증폭기 근처에 가연성 물질을 두지 마십시오.
- 오일이나 물 속에 있었던 케이블은 사용하지 마십시오.
- 증폭기에 장애가 발생하면 전원에서 증폭기로의 전원 공급을 차단하십시오.
- 장기간 증폭기를 사용하지 않을 때는 전원 공급 장치를 끄십시오.
- 이상한 냄새, 소음, 연기, 비정상적인 발열 또는 진동이 발생하면 즉시 작동을 중지하고 전원 공급 장치를 끄십시오.
- 적절한 외부 셉트 저항을 사용하십시오. 외부 셉트 저항을 사용할 때 안전을 위해 온도 센서 등의 외부 보호 장치를 설치하십시오.
- 위의 모든 작업을 수행하지 않으면 부상, 장비 손상, 오작동 또는 화재가 발생할 수 있습니다.

유지보수 및 문제 해결

- 다음 작업을 수행하기 전에 항상 전원을 끄십시오.
 - a) 장치 장착 또는 제거.
 - b) 케이블 연결 또는 시스템 배선.
 - c) 단자대 또는 커넥터 연결 또는 분리.
- 위의 모든 작업을 수행하지 않으면 심각한 부상, 장비 손상 또는 오작동이 발생할 수 있습니다.
- 증폭기를 분해, 수리, 개조하지 마십시오. 손상, 오작동 또는 부상의 원인이 될 수 있습니다.

경고 라벨의 위치

증폭기에는 다음 위치에 경고 라벨이 부착되어 있습니다. 이 라벨에 제공된 지침을 반드시 준수하십시오.



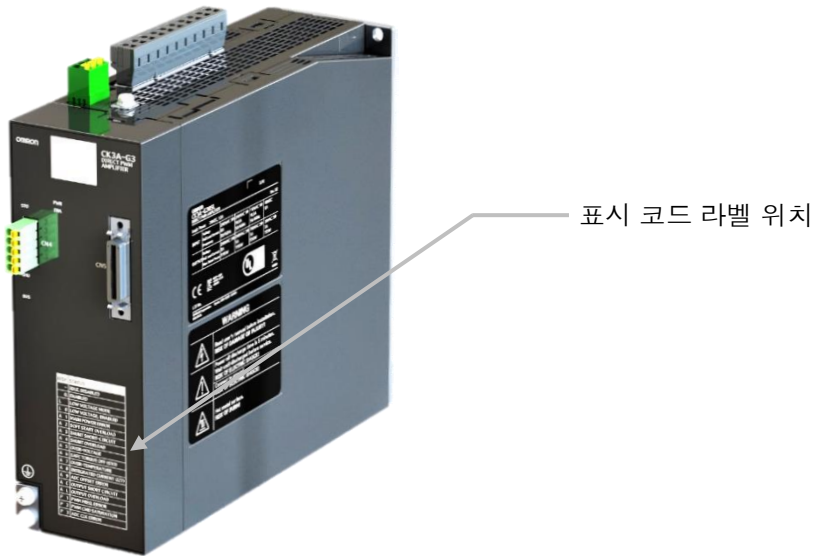
● 경고 표시에 대한 지침

WARNING	
	Read user's manual before installation. RISK OF DAMAGE OR INJURY!
	Power off discharge time is 5 minutes. Wait until discharged before service. RISK OF ELECTRIC SHOCK!
	Connect to ground. RISK OF ELECTRIC SHOCK!
	Hot metal surface. RISK OF BURN!

참고 위는 경고 표시의 예입니다.

표시 코드 위치

증폭기에는 다음 위치에 표시 코드 라벨이 부착되어 있습니다. 이것은 빠른 참조를 위한 것입니다. 표시 코드에 대한 자세한 내용은 이 설명서의 문제 해결 섹션을 참조하십시오.



● 표시 코드의 정보

DISP.	STATUS
-	IDLE, DISABLED
0.	ENABLED
L	LOW VOLTAGE MODE
L 0	LOW VOLTAGE, ENABLED
A 1	MAIN POWER ERROR
A 2	SOFT START OVERLOAD
A 3	SHUNT SHORT-CIRCUIT
A 4	SHUNT OVERLOAD
A 5	OVER-VOLTAGE
A 6	SAFE TORQUE OFF (STO)
A 7	OVER-TEMPERATURE
A 8	INTEGRATED CURRENT (I2T)
A 9	ADC OFFSET ERROR
A C	OUTPUT SHORT CIRCUIT
A L	OUTPUT OVERLOAD
P 1	PWM FREQ. ERROR
P 2	PWM CMD SATURATION
P 3	ADC CLK ERROR

폐기

WEEE 지침에 따라 폐기하십시오.

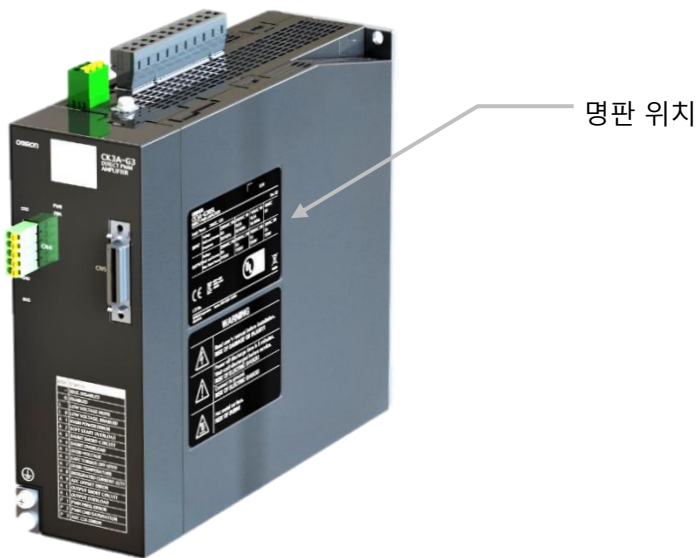
포장을 뜯은 후 확인해야 할 사항

제품의 포장을 뜯은 후 다음 사항을 확인하십시오.

- 주문한 모델인가?
- 배송 중 파손은 없었는가?

명판 위치

증폭기에는 다음 위치에 명판 라벨이 부착되어 있습니다.



명판의 정보

증폭기 모델 →

증폭기 정격 →

규정 및 표준 →

OMRON CK3A-G305L DIRECT PWM AMPLIFIER					
Logic Power 24VDC, 1.5A					Ver. 02
INPUT	Voltage	240VAC 3Φ	240VAC 1Φ	110VAC 1Φ	48VDC
	Full Load Current	6A	10.5A	10.5A	6A
	Frequency	50/60Hz	50/60Hz	50/60Hz	
OUTPUT	Voltage	240VAC 3Φ	240VAC 3Φ	110VAC 3Φ	39VAC 3Φ
	Full Load Current	5A	5A	5A	5A
	Max. Rated Power	1195W	1195W	550W	195W
LOT No. 					
OMRON Corporation Kyoto, 600-8350 JAPAN MADE IN USA					

로트 번호 및
시리얼 번호

참고 CK3A-G305L 모델의 명판 예.

부속품

증폭기에는 다음 부속품이 함께 제공됩니다.

- 사용 설명서 × 1부
- 경고 라벨 × 1장
- EU에 대한 일반 규정 준수 정보 및 지침 × 1부
- 연결된 커넥터(아래 표 참조)

UL/CSA 인증이 필요한 경우 증폭기 주변에 경고 라벨을 부착하십시오.

아래 표 이외의 커넥터, 장착 나사, 장착 브래킷 및 기타 부속품은 제공되지 않습니다.

이러한 부속품은 고객이 조달해야 합니다.

모델	주 전원 커넥터 (CN1)	제어 전원 커넥터 (CN2)	모터 커넥터 (CN3)	STO 커넥터 (CN4)	PWM 커넥터/케이블 (CN5)
CK3A-G305L	포함됨	포함됨	포함됨	포함됨	포함되지 않음
CK3A-G310L					

누락된 품목이 있거나 증폭기 손상과 같은 문제가 발견된 경우 제품을 구입한 지역 OMRON 대리점 또는 영업소에 문의하십시오.

규정 및 표준

이 섹션에서는 CK3A 다이렉트 PWM 증폭기에 적용되는 표준에 대해 설명합니다.

EU 지침 준수

항목	표준
EMC 지침	EN61800-3 second environment
저전압 지침	EN61800-5-1 C2 category
기능 안전	EN61800-5-2 SIL3 (STO)

UL 지침 준수

항목	표준
UL 표준	UL 61800-5-1
CSA 표준	CSA C22.2 No. 274

KC 표준 준수

한국에서 증폭기를 사용할 때는 다음 주의사항을 준수하십시오.

사용자 안내문

이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

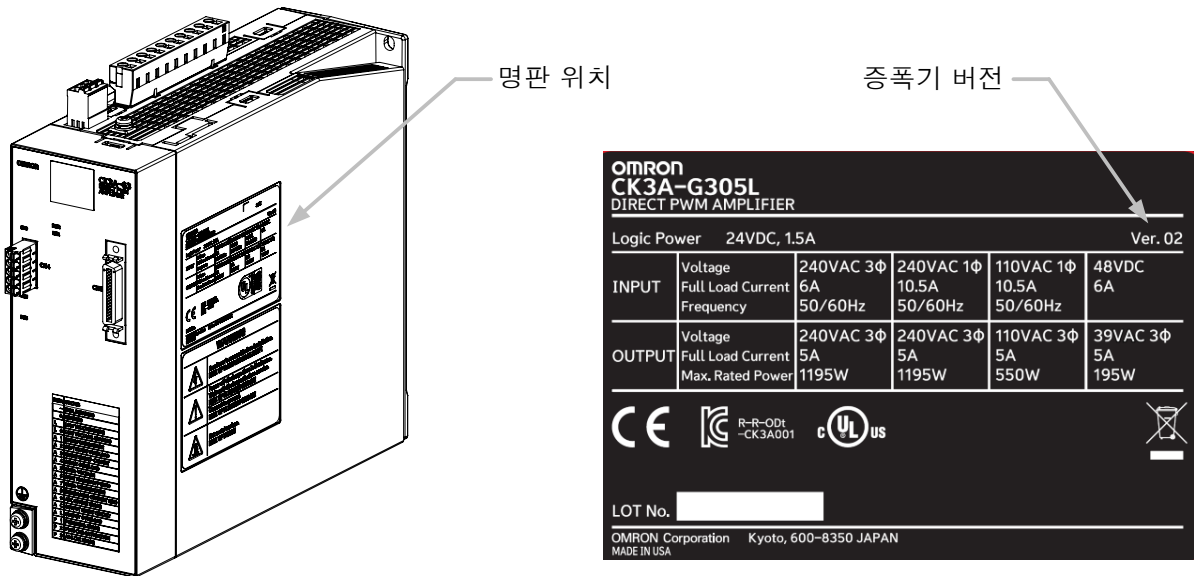
이 제품은 업무용 전자파 적합성 요구 사항을 충족합니다. 이 제품을 가정에서 사용할 경우 무선 간섭의 위험이 있습니다.

버전

CK3A 다이렉트 PWM 증폭기는 장치 버전을 사용합니다. 장치 버전은 제품 수정, 업그레이드 등의 차이점을 관리하는 데 유용합니다.

하드웨어 버전

CK3A 다이렉트 PWM 증폭기의 하드웨어 장치 버전은 명판에 표시되어 있습니다.



펌웨어 버전

증폭기 펌웨어 버전은 이 설명서의 소프트웨어 구성 섹션에 있는 지침에 따라 검색할 수 있습니다.

관련 설명서

다음은 참조용 관련 설명서입니다. 설명서에 대해서는 현지 OMRON 대리점에 문의하십시오.

설명서 이름	Cat. No.	용도	설명
CK3A 시리즈 다이렉트 PWM 증폭기 사용 설명서 (이 설명서)	O050	설치, 배선, 기본 소프트웨어 구성, 유지 보수 및 문제 해결을 포함한 사양에 대해 알 수 있습니다.	• 증폭기 소개 • 구성, 기능 및 사양 • 장착, 설치 및 배선 • 기본 소프트웨어 구성 • 유지보수 및 문제 해결
CK3M 시리즈 프로그램형 다축 컨트롤러 하드웨어 사용 설명서	O036	소개 정보, 설계, 설치 및 유지보수를 포함하여 CK3M의 기본 사양을 알 수 있습니다. 주로 하드웨어 정보입니다.	• 기능 및 시스템 구성 • 소개 • 부품 이름 및 기능 • 일반 사양 • 설치 및 배선 • 유지보수 및 검사
Power PMAC 소프트웨어 참조 설명서	O015	Power PMAC 컨트롤러의 명령 세트와 구조 요소를 알 수 있습니다.	• Power PMAC 데이터 구조 • 모든 명령의 목록과 설명 • CK3M 및 UMAC를 포함한 모든 ASIC, 좌표계 및 모터 구조 요소의 목록과 설명
Power PMAC 사용 설명서	O014	Power PMAC 컨트롤러의 기능과 사용 예를 알 수 있습니다.	• 증폭기와 관련된 매개변수 설정 • 모터 기본 기능 • 인코더 구성 예 • 모터 설정 예 • Power PMAC 프로그래밍 예
Power PMAC IDE 사용 설명서	O016	Power PMAC 컨트롤러의 통합 개발 환경 IDE를 사용하는 방법을 알 수 있습니다.	• Power PMAC IDE 소프트웨어의 작동 절차 • 시스템 설정을 사용한 다이렉트 PWM 증폭기 구성
ACC-24E3 하드웨어 참조 설명서	N/A	소개 정보, 설계, 설치 및 유지보수를 포함하여 UMAC 부속품 ACC-24E3의 기본 사양을 알 수 있습니다.	• 기능 및 시스템 구성 • 소개 • 부품 이름 및 기능 • 일반 사양 • 설치 및 배선

용어

용어	설명
Power PMAC	• PMAC는 Programmable Multi Axis Controller의 약자 • Power는 이 설명서를 작성할 당시의 CPU 세대를 의미
CK3M	CK3M 시리즈 프로그램형 다축 컨트롤러(Power PMAC CPU 내부)
UMAC	• Universal Machine and Automation Controller의 약어 • 일반적으로 Power PMAC CPU 및 기타 부속품이 들어있는 랙 • Delta Tau Data Systems, Inc.에서 설계
Gate3	다이렉트 PWM 옵션이 있는 CK3M/UMAC 축 서보 카드의 Delta Tau ASIC
컨트롤러	이 설명서에서는 Power PMAC Controller w/ Gate3 ASIC를 줄여서 사용
증폭기	이 설명서에서는 CK3A 다이렉트 PWM 증폭기를 줄여서 사용
IDE	• Integrated Development Environment(통합 개발 환경)의 약어 • Power PMAC 컨트롤러를 프로그래밍하는 데 사용되는 기본 소프트웨어
F.L.A.	Full Load Amperage의 약어
IPM	• Intelligent Power Module의 약어 • 이 설명서에서는 증폭기의 전력 전자 회로를 의미
FPGA	Field Programmable Gate Arrays
ADC	• Analog to Digital Converter • 이 설명서에서는 증폭기의 ADC 전류 센서를 의미
ADC 스트로브 워드	Power PMAC 구조 요소 AdcAmpStrobe를 의미
PWM	Pulse Width Modulation의 약어
다이렉트 PWM	이 설명서에서는 Controller-CK3A 인터페이스를 의미
주 회로 전원	증폭기의 AC 주 전원 공급 장치를 의미
로직 전원	증폭기의 24VDC 제어 전원 공급 장치를 의미
버스 전압	증폭기 내부의 DC 버스 링크를 의미
PLC	• Programmable Logic Controller의 약어 • 이 설명서에서는 Power PMAC PLC 프로그램을 의미
I2T	열 보호 모델로, 증폭기 내장 I2T 또는 PMAC 구성형 I2T를 의미
ECT	Encoder Conversion Table(인코더 변환표)의 약자

설명서 개정 이력

설명서 개정 코드는 설명서의 앞표지와 뒤표지에 있는 카탈로그 번호의 접미사 형태로 표시됩니다.

Cat. No.	O050-E1-01
----------	------------

↑ 개정 코드

개정 코드	날짜	개정된 내용
01	2022년 4월	최초 간행

증폭기 소개

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 기능, 전체 구성, 각 부품의 이름 및 작동 절차를 설명합니다.

1-1	개요.....	1-2
1-1-1	증폭기 기능	1-2
1-1-2	일반 구성.....	1-3
1-1-3	부품 이름 및 위치	1-4
1-1-4	부품 기능.....	1-8
1-2	작동 절차	1-10
1-2-1	준비	1-10
1-2-2	안전	1-10
1-2-3	장착 및 배선	1-10
1-2-4	안전 테스트 및 전원 켜기	1-10
1-2-5	증폭기용 컨트롤러 설정	1-11
1-2-6	테스트 실행	1-11

1-1 개요

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 기능 및 기본 구성에 대해 설명합니다.

1-1-1 증폭기 기능

정보

CK3A 다이렉트 PWM(이하 증폭기라고도 함)은 디지털 다이렉트 PWM 서보 카드와 인터페이스하도록 설계된 증폭기 제품군으로, 일반적으로 OMRON 및 Delta Tau Data, Inc.(이하 Delta Tau) 시리즈의 Turbo(레거시) 및 Power PMAC 컨트롤러와 통합되어 있습니다.

초저지연 서보 컨트롤

디지털 다이렉트 PWM을 사용하는 이 증폭기는 ASIC(DSPGATE) 머신 인터페이스를 통해 컨트롤러에서 직접 실제 전력 트랜지스터 켜기/끄기 신호를 가져오는 동시에 폐쇄 루프 서보 제어를 위한 디지털 위상 전류 피드백 및 상태를 제공합니다.

이 아키텍처는 산업 분야에서 가장 빠른 컨트롤러-증폭기 인터페이스 중 하나를 제공하여 초고속 서보 업데이트 속도를 허용합니다.

유연성

CK3A 증폭기는 Power PMAC 컨트롤러의 유연성 및 다양한 인코더 피드백 장치와의 인터페이스 기능과 결합되어 다음 유형의 모터를 지원할 수 있습니다.

- AC/DC 동기 브러시리스(로터리 또는 리니어) – 예: 서보 모터.
- DC 브러시드 – 예: 보이스 코일 액추에이터.
- AC 비동기 – 예: 인덕션 모터(이 유형의 모터 설정은 지원팀에 문의).

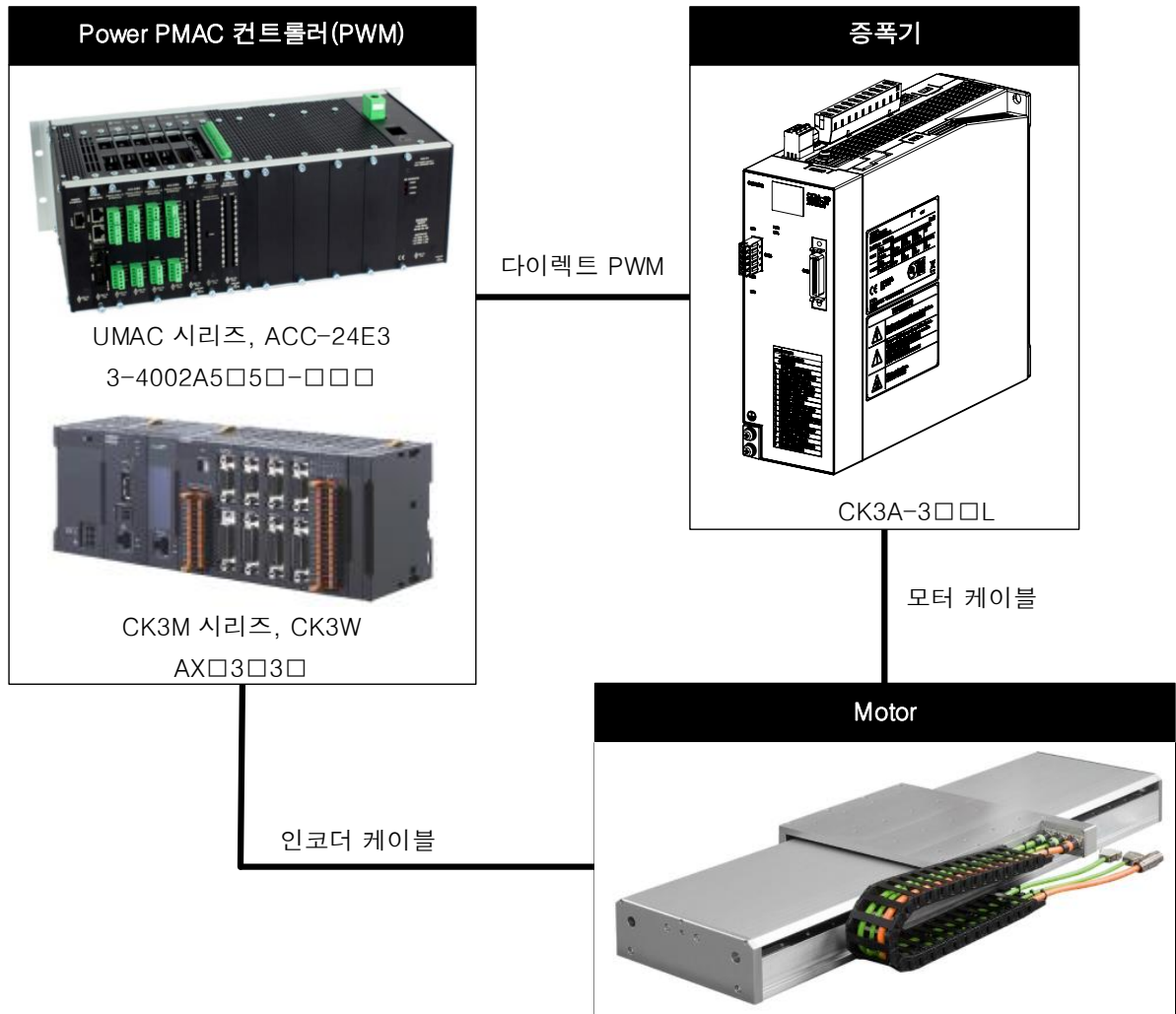
증폭기 기능

- 나노 스케일 리니어 서보 포지셔닝 정확도
- 높은 PWM 주파수 – 최대 20KHz
- 고분해능 전류 감지 – 16비트 ADC
- 고속 전류 ADC 샘플링 – 최대 6.125MHz
- 듀얼 STO 입력
- STO 상태 출력
- 기본 기능 – 에너지 방전, 다이내믹 제동, 팬 제어
- 기본 데이터 보고 – DC 버스 전압, 전원 모듈 온도, 펌웨어 버전
- 듀얼 7 세그먼트 LED 상태 디스플레이
- 내장 또는 외부 셉트 저항
- 저전압 주 전원 작동 지원

1-1-2 일반 구성

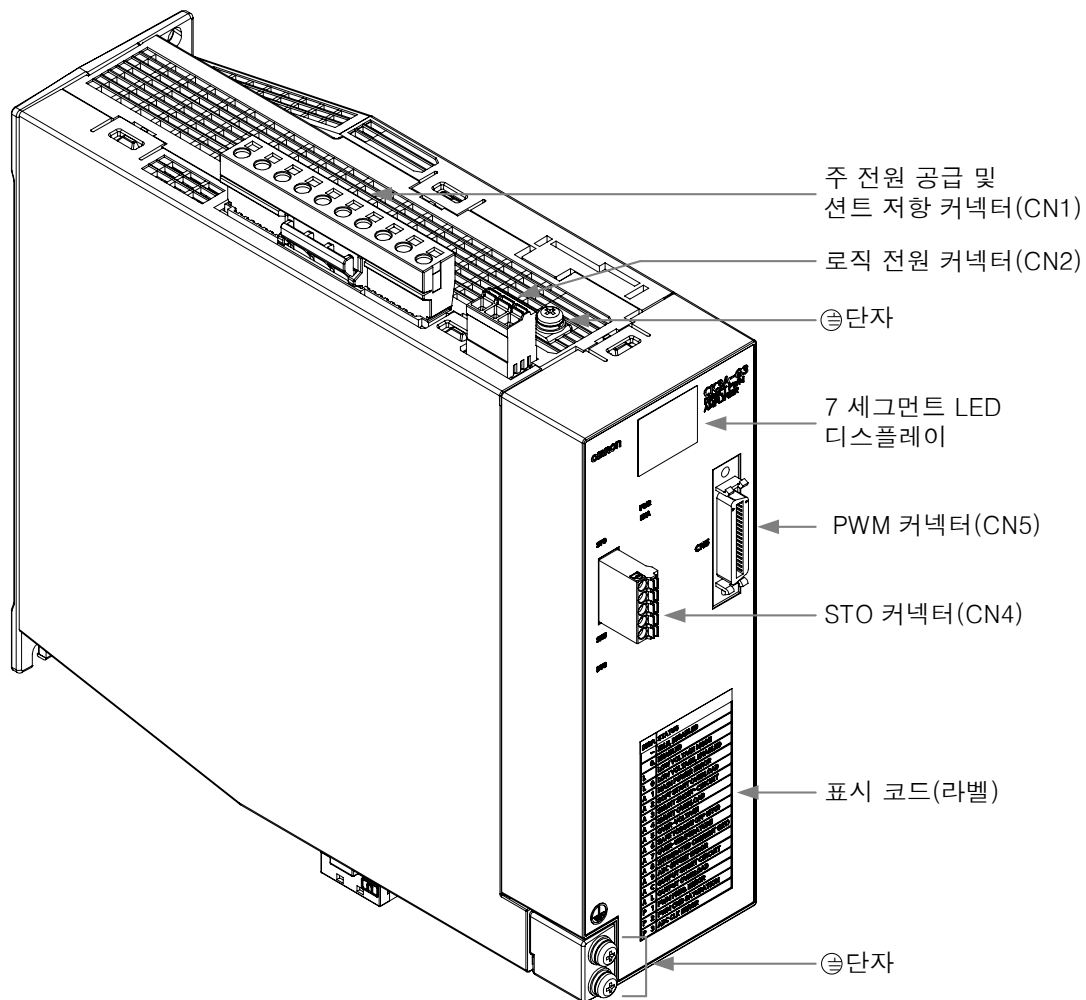
CK3A 다이렉트 PWM 증폭기는 PWM 케이블을 통해 CK3M 또는 UMAC 컨트롤러에 연결됩니다.

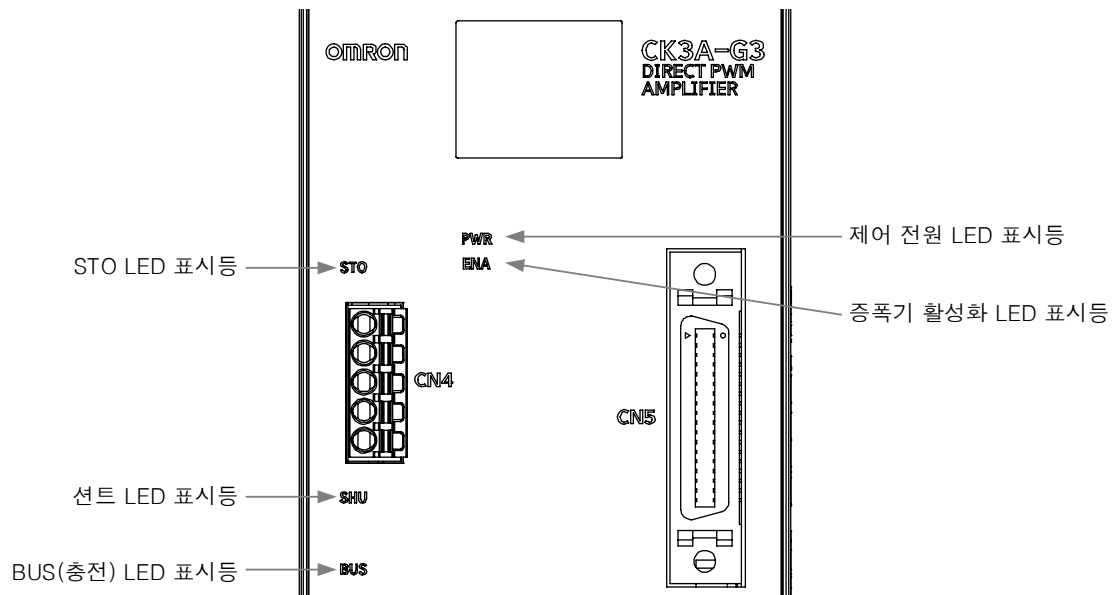
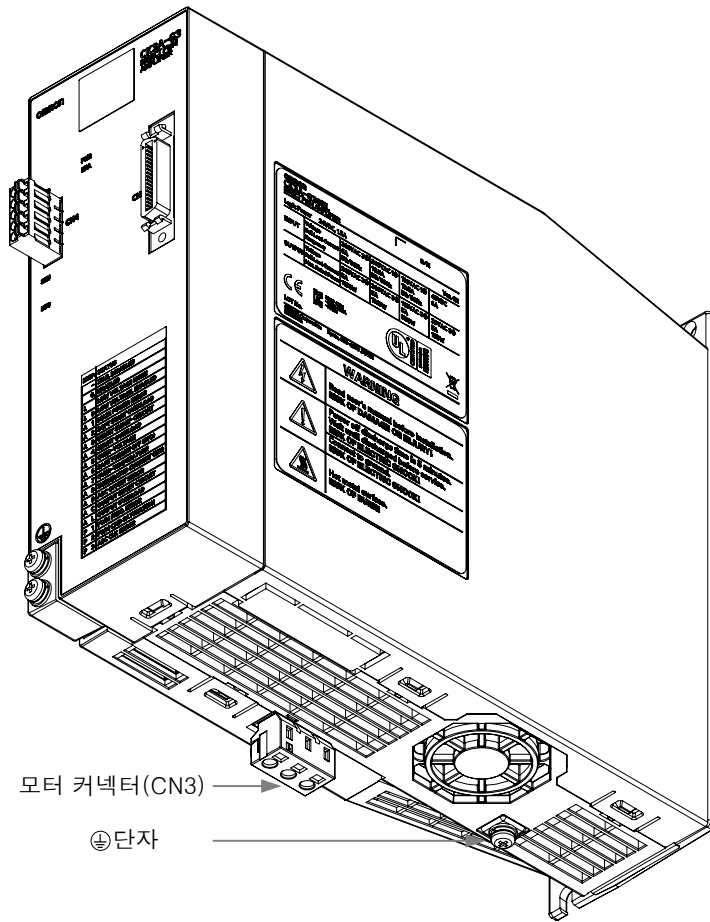
- 모터는 CK3A 다이렉트 PWM 증폭기에 직접 연결됩니다.
- 인코더는 컨트롤러에 직접 연결됩니다.



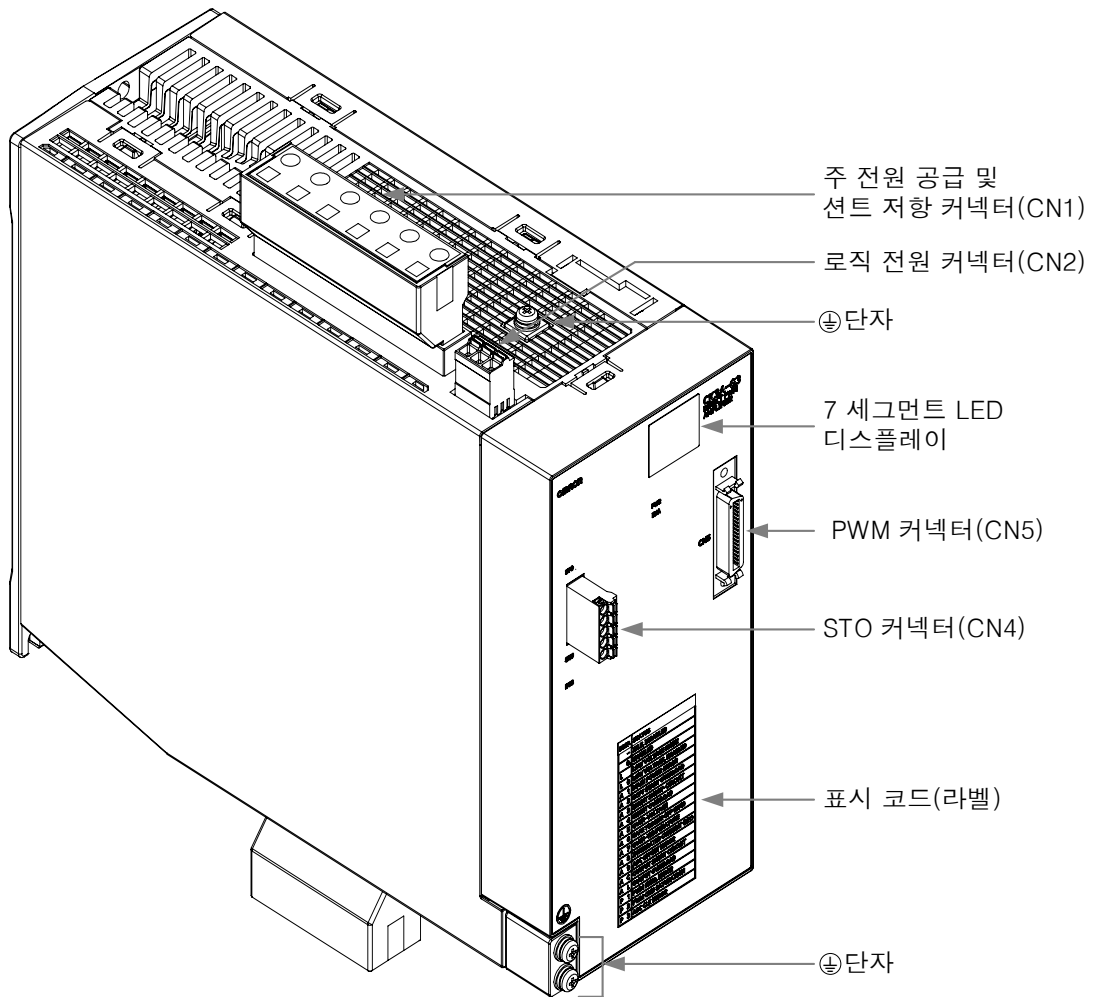
1-1-3 부품 이름 및 위치

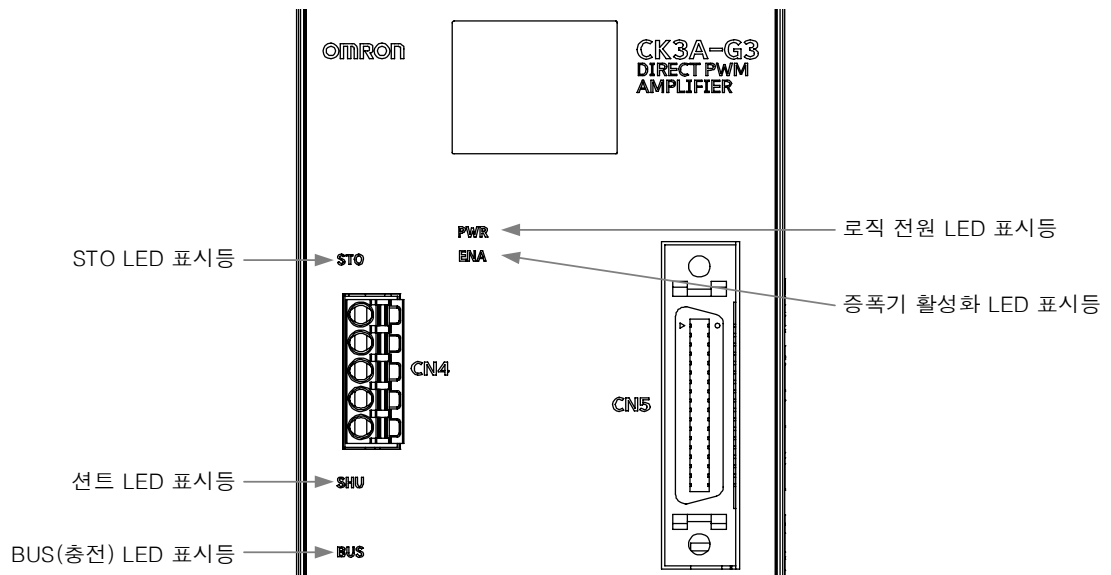
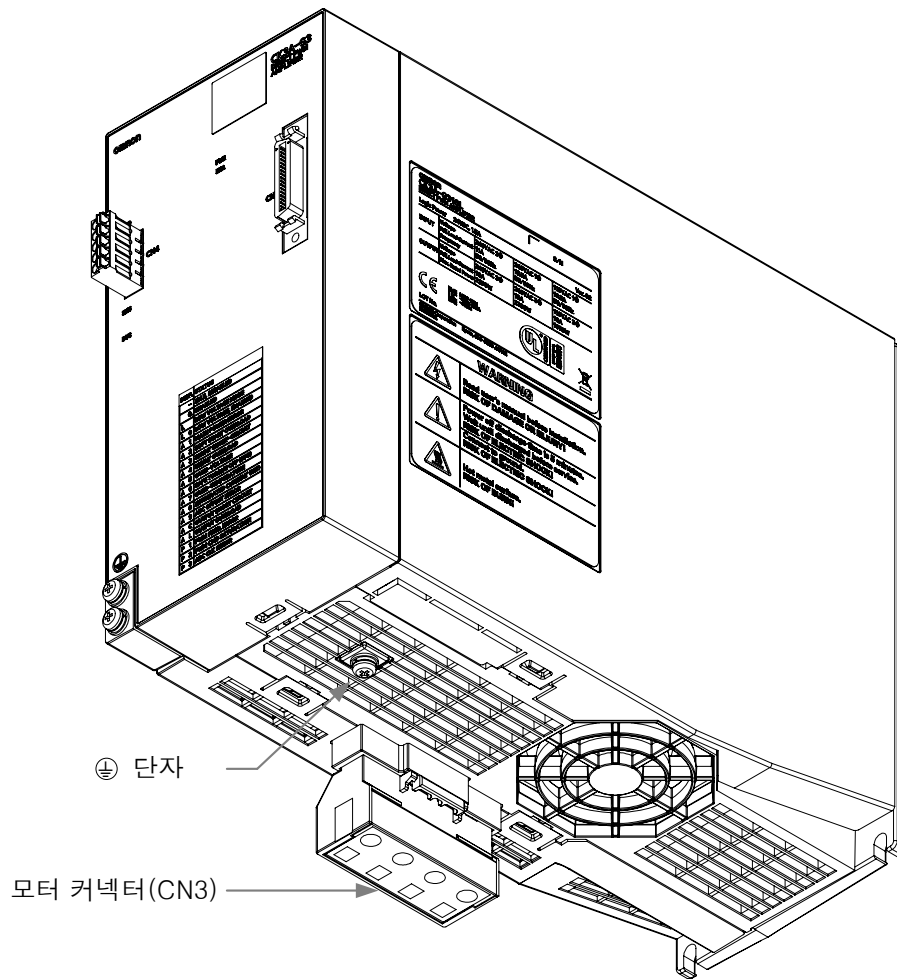
CK3A-G305L





CK3A-G310L





1-1-4 부품 기능

상태 LED 표시등

다음 LED 표시등은 증폭기 전면에 있습니다.

이름	색상	설명
PWR	녹색	로직 전원 공급 장치 상태를 나타냄
ENA	녹색	증폭기 활성화 상태를 나타냄
SHU	빨간색	셧 동작이 활성화인지 여부를 나타냄
STO	빨간색/녹색	STO 입력 상태를 나타냄
BUS	빨간색	버스 충전 상태를 나타냄

자세한 내용은 문제 해결 섹션을 참조하십시오.

7 세그먼트 LED 디스플레이

듀얼 7 세그먼트 LED 디스플레이는 증폭기 전면에 있습니다. 이 디스플레이는 증폭기 작동 및 오류 상태를 보고합니다. 전원을 켜면 7 세그먼트 디스플레이가 모든 표시등을 6회 스크롤합니다. 오류 코드에 대한 자세한 내용은 문제 해결 섹션에서 찾을 수 있습니다.



추가 정보

증폭기가 꺼져 있으면 표시등이 잘 보이지 않을 수 있습니다. 디스플레이가 흰색 면으로 나타날 수 있습니다.

주 회로 및 셧트 커넥터(CN1)

CN1 커넥터는 다음 기능에 사용됩니다.

- **CK3A-G305L(10핀)**
 - 주 회로 전원 연결
 - 내부 셧트 저항 연결(단락 와이어 사용) 또는 외부 셧트 저항 연결
 - 저전압 모드(단락 와이어 사용)
- **CK3A-G310L(6핀)**
 - 주 회로 전원 연결
 - 내부 셧트 저항 연결(단락 와이어 사용) 또는 외부 셧트 저항 연결

로직 전원 공급 커넥터(CN2)

CN2 커넥터(3핀)는 증폭기에 24VDC 로직 전원을 공급하는 데 사용됩니다.

모터 커넥터(CN3)

CN3 커넥터는 모터와 증폭기를 연결하는 데 사용됩니다. CK3A-G305L 모델은 3핀 커넥터, CK3A-G310L 모델은 4핀 커넥터입니다.

세이프 토크 오프 커넥터(CN4)

CN4 커넥터(5핀)는 STO 입력 및 STO 상태 출력을 비활성화하거나 연결하는 데 사용됩니다. STO를 비활성화하기 위한 단락 와이어는 공장에서 커넥터에 설치됩니다.

다이렉트 PWM 커넥터(CN5)

CN5 커넥터(36핀)는 증폭기를 컨트롤러에 연결하는 데 사용됩니다. 이는 사전 구성된 케이블 커넥터입니다.

접지 단자

다음은 증폭기의 접지 ㉡ 단자의 위치와 개수입니다.

위치	단자 수	권장 사용
위	1	주 회로 전원 입력의 보호 접지 PE
전면	2	제어판 내부의 프레임 접지 FG
아래	1	모터 케이블 및 실드의 프레임 접지 FG

1-2 작동 절차

이 섹션에서는 CK3A 다이렉트 PWM 증폭기를 시스템에 통합하는 데 필요한 단계를 설명합니다.

1-2-1 준비

항목	참조
Power PMAC 컨트롤러	시스템 구성에 표시된 PWM 옵션이 있어야 함
로직 전원 공급	이 설명서에 명시된 사양 내에 있어야 함
주 회로 전원 공급	이 설명서에 명시된 사양 내에 있어야 함
모터 사양	필요한 작동 조건과 증폭기 사양 내에 있어야 함
치수, 패널 크기 및 위치	이 설명서에 명시된 조건과 사양 내에 있어야 함

1-2-2 안전

항목	참조
전반적인 안전	위험 평가를 기반으로 안전 조치 결정
경고 및 주의 검토	이 설명서의 안전 주의사항 섹션에 명시된 알림
설계	전반적인 머신 안전 및 경고를 기반으로 안전 조치 설계

1-2-3 장착 및 배선

항목	참조
장착	이 설명서에 설명된 장착 지침을 따름 • 인체공학적 및 환경적 위치 • 장착 방향 • 단일 장치 간격 • 다중 장치 간격
배선	이 설명서에 설명된 배선도 및 지침 사용 • 안전 조치 • 올바른 핀아웃 • 와이어 게이지 • 인체공학적 케이블 관리

1-2-4 안전 테스트 및 전원 켜기

항목	참조
안전 테스트	테스트 켜기/끄기 안전 조치 구현(예: STO, 비상 중지)
안전한 사용을 위한 주의사항	이 설명서의 안전한 사용을 위한 주의사항 섹션 검토
전원 켜기	안전한 사용을 위한 주의사항의 전원 켜기 절차 참조

1-2-5 증폭기용 컨트롤러 설정



안전한 사용을 위한 주의사항

이 단계는 주 회로 전원을 끈 상태에서 수행해야 합니다.

항목	참조
준비	• 사용 중인 컴퓨터에 Power PMAC IDE 소프트웨어를 설치 • 이 설명서에 설명된 관련 설명서 목록을 식별
Power PMAC 컨트롤러	• Power PMAC와 IDE 소프트웨어 간의 통신 설정 • 기존 프로젝트를 로드하거나 공장 기본 설정에서 시작 • 컨트롤러 폼 팩터 또는 IDE 시스템 설정 관련 설명서 사용
인코더 설정	인코더 사양 시트 및 관련 설명서를 사용하여 인코더별 구성 및 설정 구현
증폭기 설정	이 설명서에 설명된 증폭기 특정 설정 구현
모터 설정	모터 사양서 및 관련 설명서를 사용하여 모터별 구성 및 설정 구현

1-2-6 테스트 실행

항목	참조
인코더 피드백 확인	관련 하드웨어 또는 사용 설명서 사용
켜기/끄기 활성화	• 0으로 설정된 컨트롤러의 전류 루프 게인으로 켜기 및 끄기(개방 루프) 활성화 • 이 설명서에 설명된 상태 표시별로 증폭기 결함이 없는지 확인
모터 설정	관련 하드웨어, 사용 설명서 또는 IDE 시스템 설정 사용
서보 커짐/꺼짐	서보 커짐/꺼짐 모터 작동 확인



추가 정보

IDE에서 자동 시스템 설정을 사용하여 이 단계를 수행할 수도 있습니다.

사양

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 일반 사양을 설명합니다.

2-1	증폭기 사양.....	2-2
2-1-1	일반/기계 사양.....	2-2
2-1-2	환경 사양.....	2-2
2-1-3	전기 사양.....	2-3
2-1-4	성능	2-4
2-1-5	회생 에너지 흡수	2-5
2-2	기능 및 데이터 보고	2-9
2-2-1	증폭기 기능	2-9
2-2-2	데이터 보고	2-10
2-2-3	상태 비트	2-10
2-3	부품 번호 지정	2-11
2-3-1	증폭기 부품 번호	2-11

2-1 증폭기 사양

이 섹션에서는 증폭기의 일반, 기계, 환경, 기능 및 전기 사양에 대해 설명합니다.

2-1-1 일반/기계 사양

항목		사양
축 수		1
인클로저		패널 장착
보호 케이스		IP20(IP54 패널 내장)
접지		200V 등급 D 접지, 100Ω 이하
내진동성		가속도 5.88m/s ² 이하에서 10 ~ 60Hz (공진 주파수에서 연속 운전하지 않음)
기류 간격		설치 섹션 참조
장착 나사 조임 토크		1.2Nm
냉각		자연 대류 및 내장 팬
무게	CK3A-G305L	1.81kg
	CK3A-G310L	2.67kg
치수	CK3A-G305L	212.5 x 65.0 x 180.0mm
	CK3A-G310L	238.0 x 90.0 x 180.0mm

2-1-2 환경 사양

항목	사양
작동 주변 온도	0 ~ 55°C
작동 주변 습도	10% ~ 90% RH(결로 또는 결빙 없음)
보관 주변 온도	-25 ~ 70°C
보관 주변 습도	10% ~ 90% RH(결로 또는 결빙 없음)
작동 및 보관 대기	부식성 가스가 없어야 함
최대 작동 고도	1,000m

2-1-3 전기 사양

괄호 안의 값은 허용 가능한 변동 범위를 나타냅니다. 괄호 밖에 있는 것이 정격 값입니다.

		항목	CK3A-G305L	CK3A-G310L
입력 전원		전압	24VDC(22.0 ~ 26.4)	
		전류 소비	1.5A	
		돌입 전류	2.5A	
		돌입 시간	5msec	
주 회로 전원	3상 AC	전압	240VAC(170 ~ 252VAC)	
		F.L.A.	6A _{rms}	11A _{rms}
		주파수	50 / 60Hz	
	단상 AC	전압	110 ~ 240VAC(85 ~ 252VAC)	
		F.L.A.	10.5 A _{rms}	19.5 A _{rms}
		주파수	50 / 60Hz	
	단상 DC ^{*1}	(저)전압	48VDC ^{*1,3} (44 ~ 60VDC)	N/A ^{*2,3}
		F.L.A.	6A _{rms}	N/A
출력	정격 전류		5A _{RMS}	10A _{RMS}
	최대(피크) 전류		10A _{RMS}	20A _{RMS}
	최대 정격 전원(3상 AC)		1195W	2390W
	최대 정격 전원(단상 240VAC)		1195W	2390W
	최대 정격 전원(단상 110VAC)		550W	1095W
	최대 정격 전원(단상 DC)		195W	N/A ^{*2}
	피크 전류에서의 시간		2sec	
PWM 인터페이스	전류 피드백 분해능		16비트	
	최대 전류 ADC 판독값		15.735A	31.470A
	최소 PWM 데드타임		2μsec	3μsec
	PWM 주파수		8 ~ 20KHz	
선티저항	내부 선티저항		30W	
	외부 선티저항		20Ω, 60W	17Ω, 60W

*1. 저전압 단락 와이어를 설치하고 ADC 스트로브 워드를 올바르게 설정해야 합니다.

*2. CK3A-G310L은 저전압 주 전원 입력으로 작동하려면 특수 부품 번호 및 공장 수정이 필요합니다. 이 옵션에 대해서는 현지 Omron 대리점에 문의하십시오.

*3. CK3A(두 모델 모두)는 단락 와이어 사용이나 특별한 고려 없이 100 ~ 350VDC 주 공급 입력 범위에서 기능적으로 작동할 수 있습니다. 작동 및 설정은 이 설명서와 동일하나, 이 경우 사양 및 성능을 규정하지 않습니다.

2-1-4 성능



안전한 사용을 위한 주의사항

안전을 위해 머신에 적절한 정지 장치를 설치하십시오. 다이내믹 브레이크 메커니즘은 정지 장치가 아닙니다.



올바른 사용을 위한 주의사항

- 다이내믹 브레이크 기능이 켜져 있으면 사용자는 증폭기를 활성화하고 모션을 명령하기 전에 최소 50밀리초를 기다려야 합니다.
- 다이내믹 제동이 수행된 경우(작동 중인 모터가 정지한 후) 사용자는 증폭기를 다시 활성화하기 전에 최소 500밀리초를 기다려야 합니다.

사양	값	참고
전원 드라이버에 대한 STO 입력 꺼짐	< 30msec	
IPM에 대한 과전류 I2T 꺼짐	< 10msec	A8 결함
IPM에 대한 위상 단락 꺼짐	< 3μsec	AC 결함
현재 루프 응답 시간	< 1msec	1mH 3상 브러시리스 모터 Y 권선
다이내믹 브레이크 릴레이 응답 시간	< 20msec	기계식 릴레이 시간 상수
증폭기에 대한 I2T 시간 꺼짐	< 2.5sec	200% 출력에서
소프트 시작 시간	< 650msec	소프트 스타트 중에는 증폭기를 활성화하지 말 것
순간 정전 시 홀드	10msec	정격 부하에서 3상 208VAC
DC 버스가 36VDC 미만으로 방전	< 2.5sec	션트 저항으로 강제 방전
DC 버스가 36VDC 미만으로 방전	< 5min	자연 방전
현재 ADC 클록 주파수 범위	2.450 ~ 6.250MHz	컨트롤러에서 설정
주 회로 전원 사이클 사이의 시간	최소 1min	방전이 켜진 경우
주 회로 전원 사이클 사이의 시간	최소 10sec	방전이 꺼진 경우

2-1-5 회생 에너지 흡수

증폭기에는 콘덴서가 내장되어 있으며, 이 콘덴서는 카운터 EMF로 인한 모터 감속 및 다이내믹 제동 시 발생하는 회생 에너지를 흡수합니다. 내장 콘덴서가 회생 에너지를 모두 흡수할 수 없는 경우 내부(또는 외부) 셉트 저항이 나머지 에너지를 흡수합니다.

모터로부터의 회생 에너지량이 너무 많으면 내부 셉트 저항의 소손을 방지하기 위해 회생 작동을 정지하고 과전압 이상(A5)으로 모터를 정지시킵니다.

회생 지속 시간이 2초를 초과하면 회생 작동을 정지하고, 셉트 과부하 오류(A3)로 모터를 비활성화시켜 과열 및 파손을 방지합니다. 이런 경우에는 5분간 냉각하는 것이 좋습니다.

이러한 경우가 발생하면 모터에서 발생하는 회생 에너지를 줄여야 합니다. 회생 에너지의 계산 및 권장 사항은 이 섹션에 설명되어 있습니다.

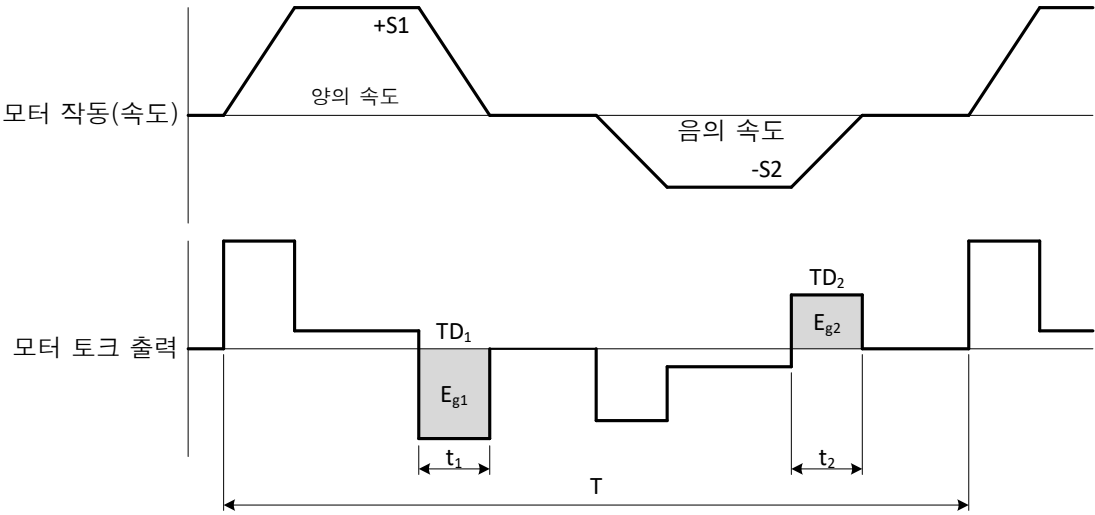
증폭기 내부 회생 흡수 용량

다음 표에는 증폭기 전원, 내부 셉트 저항 사양, 회생 전력 흡수 용량 및 최대 지속 시간이 나와 있습니다. 이 값은 200VAC 주 전원 공급 장치를 기준으로 합니다.

모델	CK3A-G305L	CK3A-G310L
정격 RMS 전원 [W]	1195W	2390W
내부 셉트 저항 사양	25Ω 30W	17Ω 80W
내장 콘덴서 흡수 에너지 [J]	46J	62J
내부 셉트 저항 평균 회생 에너지 [W]	18W	32W
연속 재생의 최대 지속 시간 [sec]	2sec	2sec
허용 가능한 최소 셉트 저항 [Ω]	20Ω	15Ω

회생 에너지 계산

● 가로 축



기호	설명
S1, S2	감속 시작 시 모터 속도(로터리 모터의 경우 [r/min], 리니어 모터의 경우 [m/s])
TD1, TD2	감속 토크 [N.m]
t1, t2	감속 시간 [s]

회생 에너지 흡수 E_{g1} 및 E_{g2} [J]는 다음과 같이 계산할 수 있습니다.

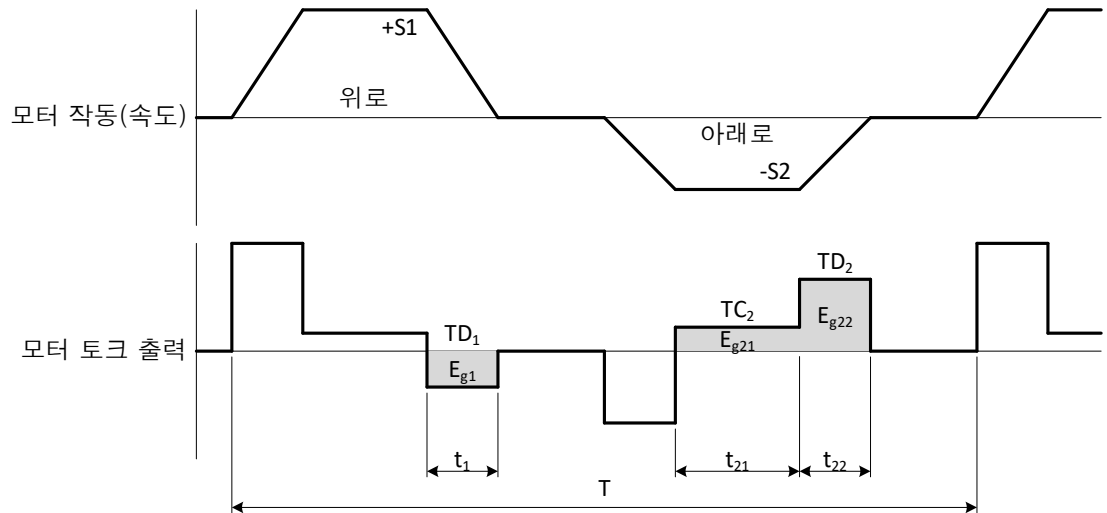
로터리 모터 회생 에너지 [J]	리니어 모터 회생 에너지 [J]
$E_{g1} = \frac{1}{2} \times \frac{2\pi}{60} \times S1 \times TD_1 \times t_1$	$E_{g1} = \frac{1}{2} \times S1 \times T_1 \times t_1$
$E_{g2} = \frac{1}{2} \times \frac{2\pi}{60} \times S2 \times TD_2 \times t_2$	$E_{g2} = \frac{1}{2} \times S2 \times T_2 \times t_2$



추가 정보

모터 권선 저항 때문에 실제 회생 에너지는 계산 값의 약 90%가 됩니다.

● 세로 축



기호	설명
S1, S2	감속 시작 시 모터 속도(로터리 모터의 경우 [r/min], 리니어 모터의 경우 [m/s])
TD ₁ , TD ₂	감속 토크 [N.m]
TC ₂	하향 정속 토크 [N.m]
t ₁ , t ₂₂	감속 시간 [s]
t ₂₁	하향 정속 시간 [s]

각 영역 Eg₁, Eg₂₁ 및 Eg₂₂ [J]에 대한 회생 에너지 흡수는 다음과 같이 계산할 수 있습니다.

로터리 모터 회생 에너지 [J]	리니어 모터 회생 에너지 [J]
$E_{g1} = \frac{1}{2} \times \frac{2\pi}{60} \times S1 \times TD_1 \times t_1$	$E_{g1} = \frac{1}{2} \times S1 \times TD_1 \times t_1$
$E_{g21} = \frac{2\pi}{60} \times S2 \times TC_2 \times t_{21}$	$E_{g21} = S2 \times TC_2 \times t_{21}$
$E_{g22} = \frac{1}{2} \times \frac{2\pi}{60} \times S2 \times TD_2 \times t_{22}$	$E_{g22} = \frac{1}{2} \times S2 \times TD_2 \times t_{22}$
$E_{g2} = E_{g21} + E_{g22}$	

하향 이동의 총 회생 에너지는 $E_{g2} = E_{g21} + E_{g22}$ 입니다.



추가 정보

모터 권선 저항 때문에 실제 회생 에너지는 계산 값의 약 90%가 됩니다.

필요한 회생 에너지 대 증폭기 흡수 용량

이전에 계산된 회생 에너지 E_{g1} 및 E_{g2} [J] 값이 모두 증폭기의 콘덴서 회생 흡수 용량 E_c [J]보다 작거나 같으면 할 일이 없습니다.

단, E_{g1} 또는 E_{g2} [J] 중 어느 하나가 증폭기의 콘덴서 회생 흡수 용량 E_c [J]보다 크면 다음 식으로 평균 회생 전력 P_r [W]을 구합니다.

$$E_g = (E_{g1} - E_c) + (E_{g2} - E_c) \text{ [J]}$$

$$P_r = E_g / T \text{ [W]}$$

기호	설명
P_r	한 사이클 작동 시 평균 회생 전력 흡수 [W]
E_g	한 사이클 작동 시 회생 에너지 흡수 [J]
E_c	내장 콘덴서 흡수 에너지 [J]
T	작동 사이클 시간 [s]

참고 $(E_{g1} - E_c)$ 또는 $(E_{g2} - E_c) \leq 0$ 인 경우, 0을 사용하십시오.

평균 회생 전력 P_r [W]이 증폭기 내장 저항의 평균 회생 전력보다 작거나 같으면 할 일이 없습니다.

하지만 평균 회생 전력 P_r [W]이 증폭기 내장 저항의 평균 회생 전력보다 큰 경우에는 다음 중 하나 이상의 조치를 취할 수 있습니다.

- 외부 셉트 저항을 연결하십시오.
- 모션 프로파일 속도를 낮추십시오. 회생 에너지는 모터 속도의 제곱에 비례합니다.
- 모션 프로파일 감속 시간을 낮추면 단위 시간당 회생 에너지가 줄어듭니다.
- 작동 사이클 시간을 늘려 연속 이동 사이에 더 많은 시간을 허용하십시오.

2-2 기능 및 데이터 보고

CK3A 다이렉트 PWM 증폭기는 PWM 케이블을 통해 컨트롤러와 인터페이스합니다.

증폭기에는 컨트롤러에서 구성할 수 있는 기능이 있습니다. 또한 컨트롤러에서 읽을 수 있는 데이터 및 상태 비트입니다.



올바른 사용을 위한 주의사항

PWM 케이블이 연결되어 있지 않거나 로직 전원이 꺼져 있으면 기능, 데이터 및 상태 정보에 액세스할 수 없습니다.



추가 정보

- 증폭기 데이터 보고는 실시간 사용을 위한 것이 아닙니다. 컨트롤러에서 수집하는 방법과 채널 수에 따라 경우에 따라 업데이트 속도가 수백 밀리초가 될 수 있습니다.
- 증폭기 데이터 보고는 프로그래밍, 문제 해결 또는 머신 운영자 상태 표시 중에 범용으로 사용하기 위한 것입니다.
- 증폭기 기능 및 데이터 보고는 컨트롤러의 ADC 스트로브 워드 설정에 의해 제어됩니다.

2-2-1 증폭기 기능

다음은 증폭기 기능입니다.

항목	기능 옵션
냉각 팬 제어	• 자동(증폭기 펌웨어에 의한) 또는 • 항상 켜기.
다이내믹 브레이크 제어	• 서보 꺼짐(또는 오류) 이벤트 후 모터가 관성 회전(프리 런)하게 하거나 • 서보 꺼짐(또는 오류) 이벤트 후 모터를 정지시킴
결함 기능 삭제	증폭기 결함을 삭제하는 프로세스(ADC 스트로브 워드 사용)
저전압 모드 컨트롤	정상 AC 입력(사양에 따름) 또는 • 저전압 44 ~ 60VDC 주 회로 전원 입력(CK3A-G305L만 해당)
버스 방전 제어	• 주 전원 회로가 꺼져 있을 때 잔류 에너지를 방출하지 않거나 • 주 전원 회로가 꺼져 있을 때 잔류 에너지를 방출함



올바른 사용을 위한 주의사항

다이내믹 제동은 결함 조건 발생 시 모터를 정지시키기 위한 용도로만 사용됩니다. 다이내믹 브레이크 기능이 켜진 경우에는 작동 중인 모터를 자주 정지시키지(서보 꺼짐) 마십시오.

2-2-2 데이터 보고

증폭기는 (컨트롤러에) 다음 데이터를 보고할 수 있습니다.

항목	사양
DC 버스 전압	DC 링크 버스 전압을 VDC로 보고합니다.
전원 모듈 온도	전원 모듈 온도를 °C로 보고합니다.
펌웨어 버전	증폭기 펌웨어 버전을 보고합니다.
전류 등급	증폭기의 전류 등급을 암페어 단위로 보고합니다.

2-2-3 상태 비트

증폭기는 (컨트롤러에) 다음 상태 비트를 보고할 수 있습니다.

상태 비트 세트 1	상태 비트 세트 2	상태 비트 세트 3
주 전원 회로 결함	주 전원 회로 상태	I2T 적분기 상태
소프트 시작 결함	과열 결함	ADC 오프셋 결함
STO 상태	션트 과부하 결함	증폭기 준비 상태
션트 단락 결함	PWM 주파수 결함	-
과전압 결함	PWM 명령 결함	-
I2T 결함	증폭기 활성화 상태	-
단락 결함	-	-
과부하 결함	-	-

2-3 부품 번호 지정

이 섹션은 부품 번호 지정을 보여줍니다.

2-3-1 증폭기 부품 번호

C	K	3	A	-	G	3			L
---	---	---	---	---	---	---	--	--	---

05: 5/10A 모델

10: 10/20A 모델



장착 및 설치

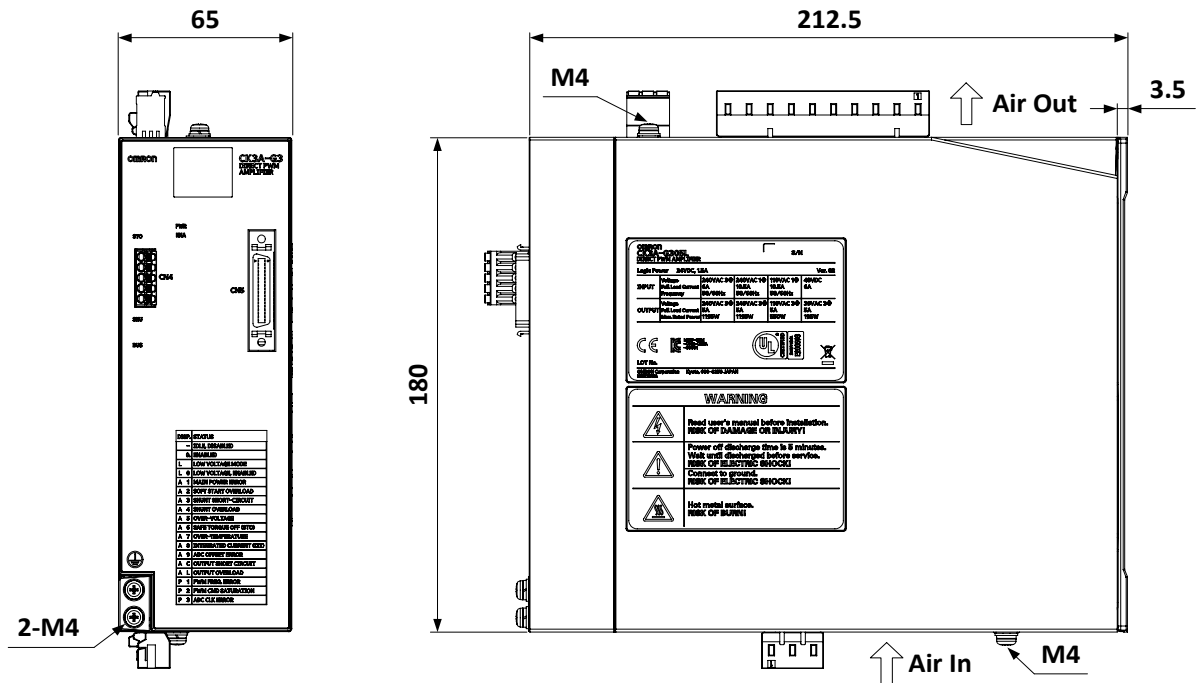
이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 장착 및 설치 방법에 대해 설명합니다. 모든 치수는 밀리미터로 표시됩니다.

3-1	외부 및 장착 치수.....	3-2
3-1-1	CK3A-G305L 치수 및 장착.....	3-2
3-1-2	CK3A-G310L 치수 및 장착.....	3-3
3-2	설치.....	3-4
3-2-1	환기	3-4
3-2-2	패널 간격	3-4
3-2-3	장착 방향.....	3-6
3-2-4	설치 조건	3-7
3-2-5	장치에 이물질 유입 방지	3-7

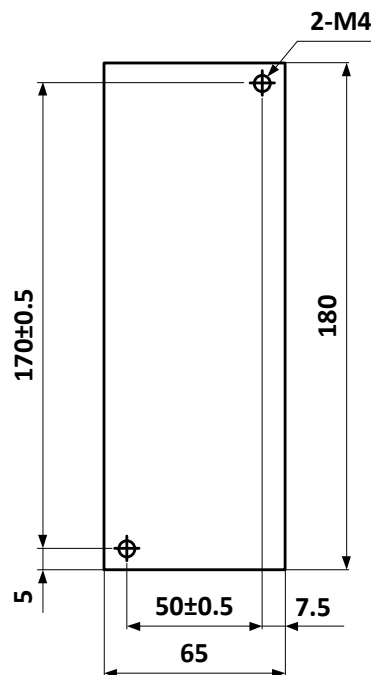
3-1 외부 및 장착 치수

3-1-1 CK3A-G305L 치수 및 장착

외부 치수

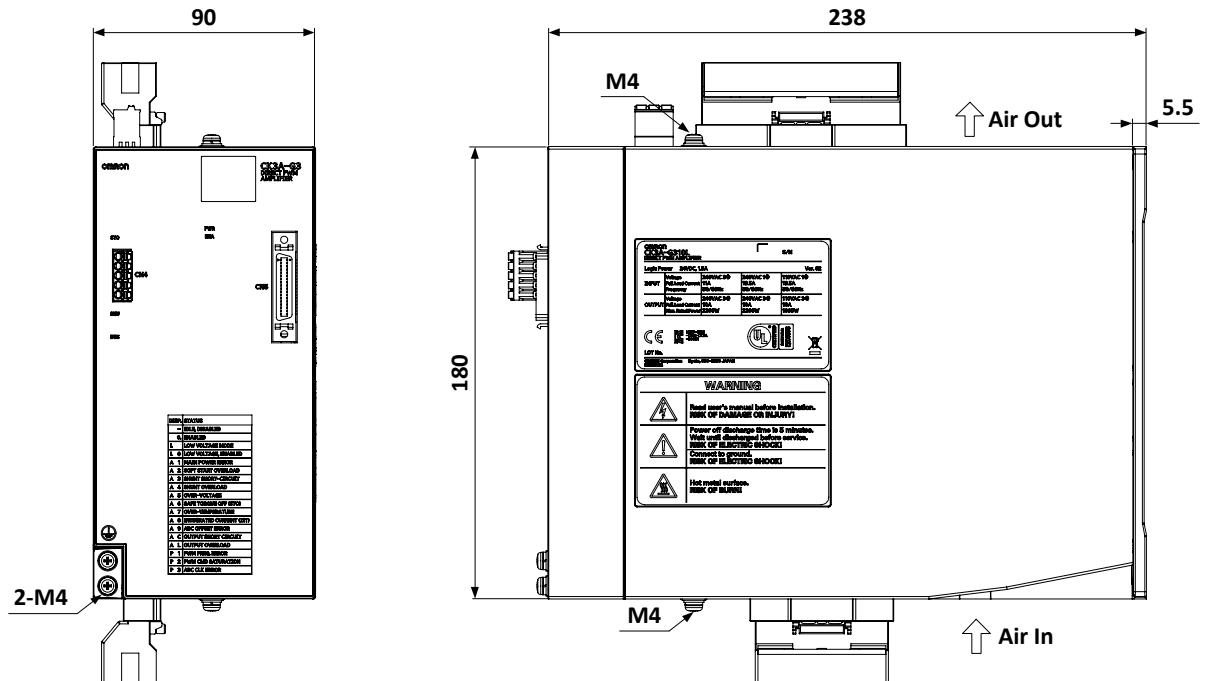


장착 치수

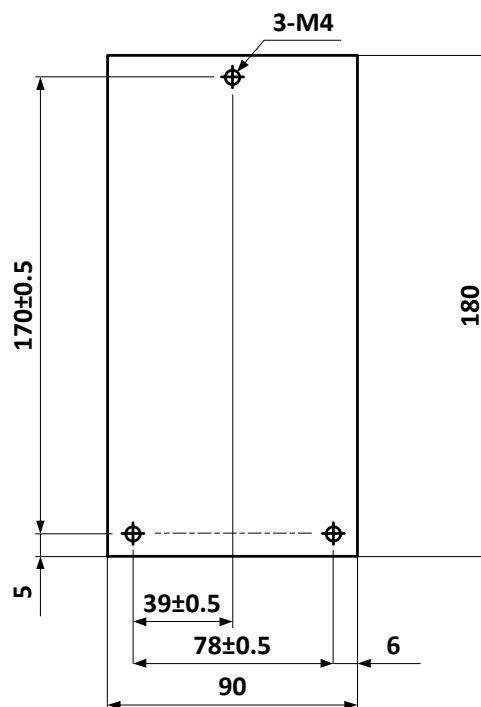


3-1-2 CK3A-G310L 치수 및 장착

외부 치수



장착 치수



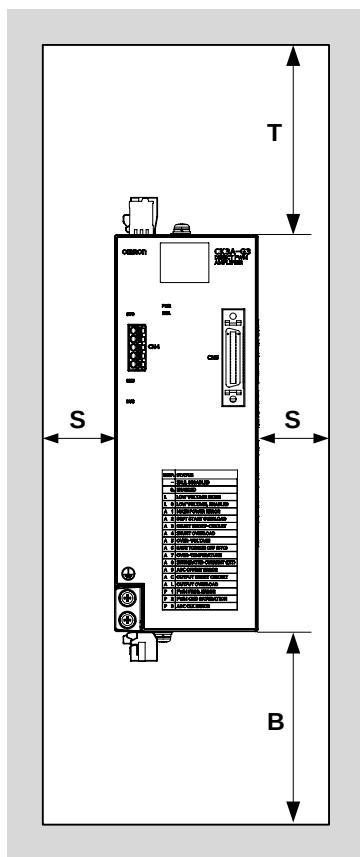
3-2 설치

3-2-1 환기

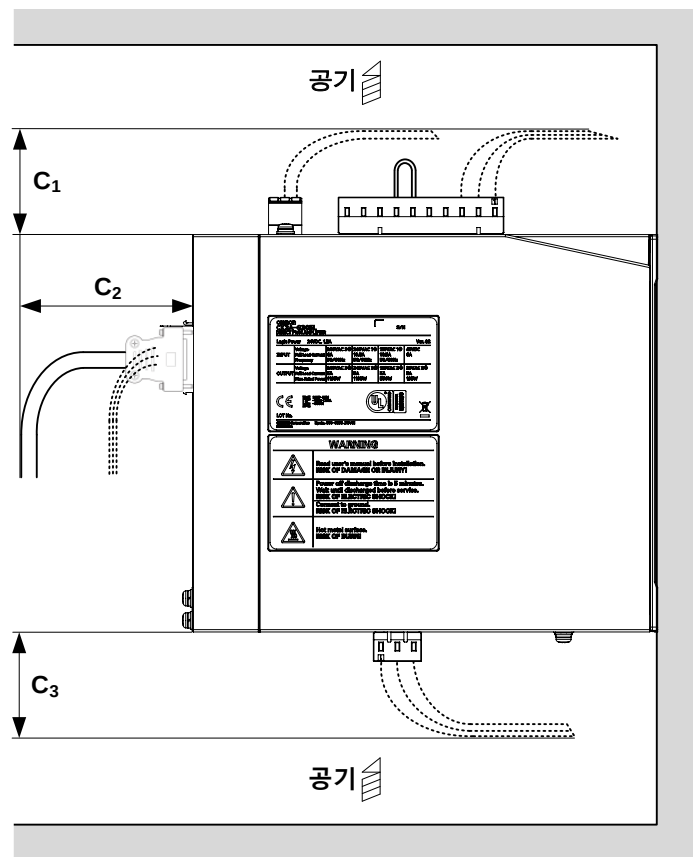
아래 표시된 치수 사양에 따라 증폭기를 설치합니다. 증폭기의 열이 적절히 방출되고 패널 내부에서 대류가 발생하는지 확인합니다. 증폭기를 나란히 설치한 경우 패널 내부의 온도가 불균일하지 않도록 공기 순환을 위한 환기 시스템 또는 팬을 설치합니다.

3-2-2 패널 간격

한 대 설치

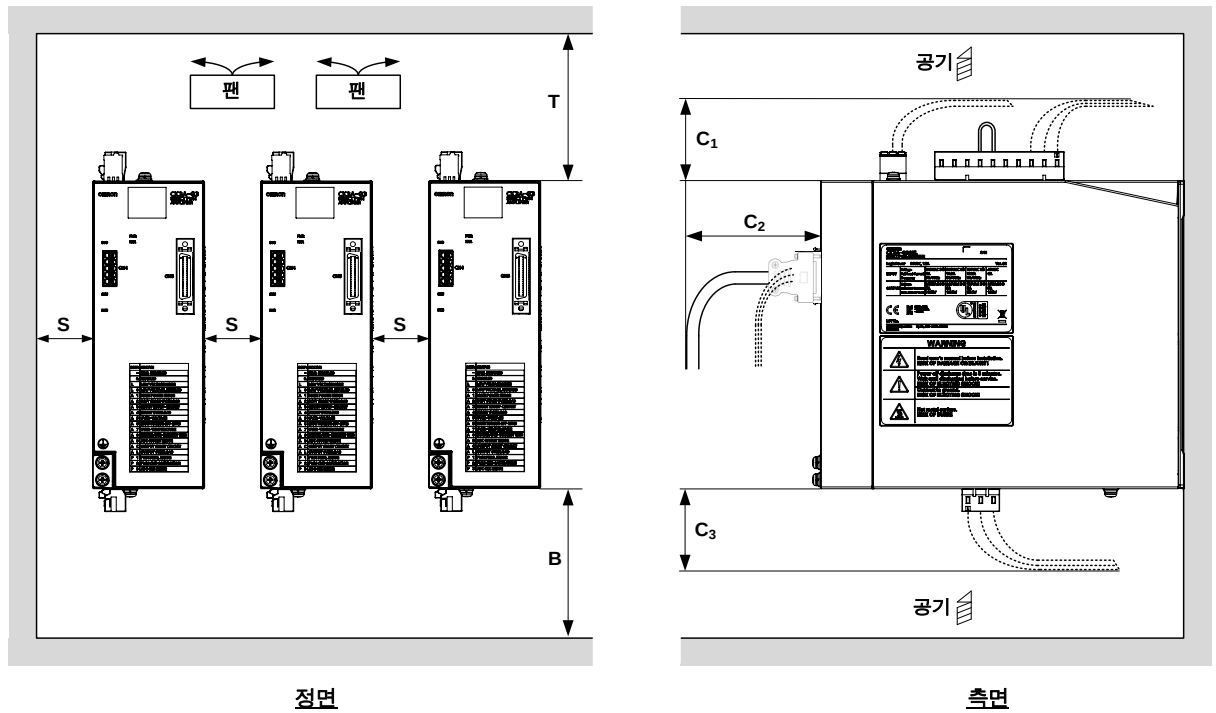


전면



측면

여러 대 나란히 설치



● CK3A-G305L

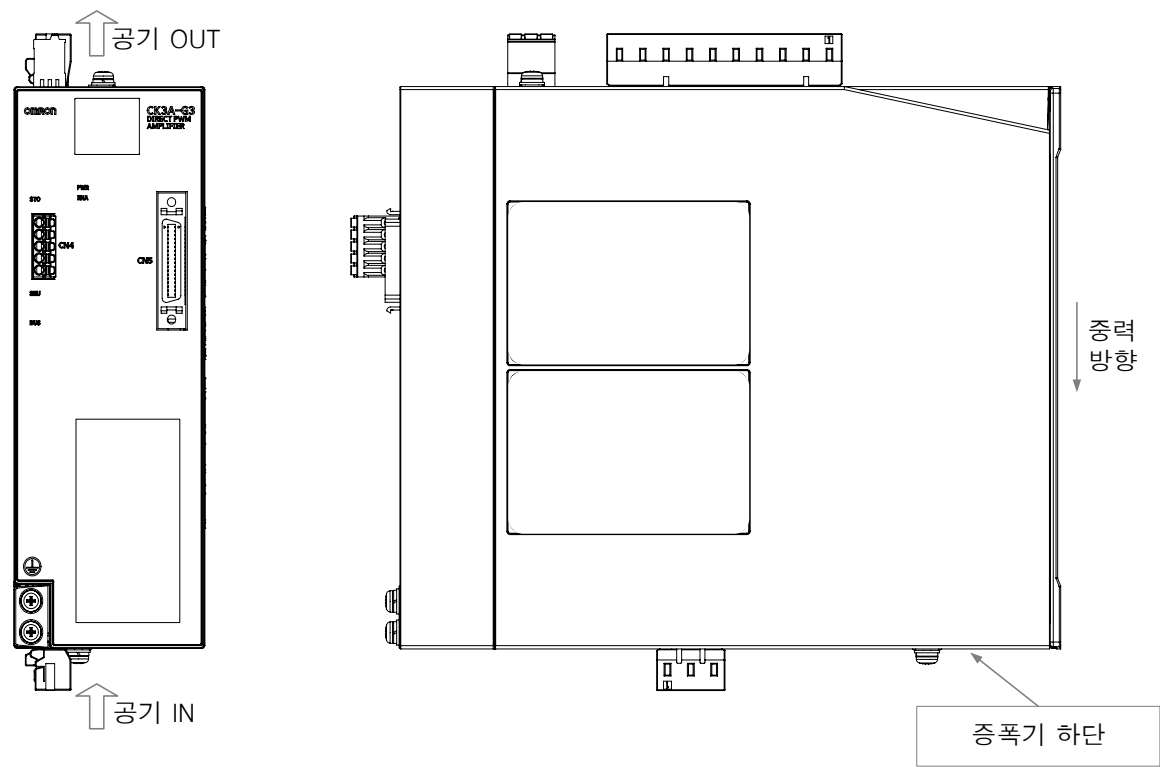
치수	최소 거리 [mm]
T(위)	110
B(아래)	110
S(옆)	10
C ₁	70
C ₂	100
C ₃	70

● CK3A-G310L

치수	최소 거리 [mm]
T(위)	140
B(아래)	140
S(옆)	10
C ₁	100
C ₂	100
C ₃	100

3-2-3 장착 방향

증폭기는 아래가 바닥과 평행하게 중력 방향으로 세워서 장착해야 합니다.



3-2-4 설치 조건

- 패널 환경(예: 작동 온도 및 습도) 조건은 이 설명서에 명시된 환경 사양을 준수해야 합니다. 이러한 조건을 벗어나 증폭기를 작동하면 오작동이 발생할 수 있습니다.

치수	거리
작동 주변 온도	0 ~ 55°C
작동 주변 습도	최대 90%(응결 없음)
작동 대기	부식성 가스 없음
작동 고도	최대 1,000m

- 증폭기를 서랍과 같은 밀폐된 공간에 나란히 설치하면 각 장치에서 나오는 열로 인해 주변 온도가 상승할 수 있습니다. 팬 또는 에어컨을 사용하여 지정된 작동 주변 온도를 유지하십시오.
- 높은 수준의 신뢰성을 유지하기 위해 온도 상승이 최소인 환경에서 작동하는 것이 좋습니다.
- 더운 환경에서 증폭기를 계속 사용하면 수명이 단축됩니다. 팬이나 에어컨을 사용하여 주변 온도를 40°C 이하로 유지하십시오.
- 수직 금속 면에 증폭기를 설치하십시오.
- 전기 전도성을 제공하기 위해 표면 페인트를 제거하십시오.
- 맞춤형 장착 브래킷을 사용할 때는 전도성 도금을 적용하십시오.
- 증폭기 장착에 권장되는 조임 나사는 1.2Nm입니다.
- 장착 나사의 나사산 부분이 증폭기의 무게와 권장 토크를 견딜 수 있는 충분한 깊이인지 확인하십시오.

3-2-5 장치에 이물질 유입 방지

- 증폭기에 금속 조각, 부스러기, 오일, 절삭유, 먼지, 물 등의 이물질이 들어가지 않도록 설치 및 운전 중에 조치를 취하십시오.
- 증폭기 위에 덮개를 씌우거나 다른 예방 조치를 취하여 설치하는 동안 드릴 파편과 같은 이물질이 증폭기에 들어가지 않도록 하십시오. 설치가 완료된 후에는 덮개를 제거하십시오. 작동 중에 덮개가 덮인 채로 있으면 증폭기의 열이 방출되지 않아 오작동이 발생할 수 있습니다.

4

배선

4

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 배선 방법을 설명합니다.

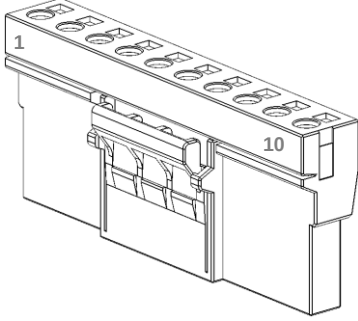
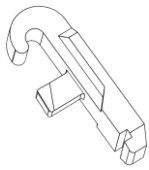
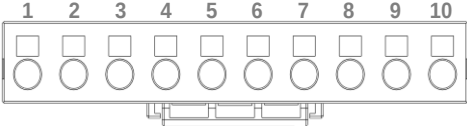
4-1	주 회로 전원 및 셉트(CN1).....	4-3
4-1-1	커넥터 핀아웃.....	4-3
4-1-2	와이어 크기	4-4
4-1-3	배선 예	4-5
4-2	로직 전원 공급(CN2).....	4-9
4-2-1	커넥터 핀아웃.....	4-9
4-2-2	와이어 크기	4-9
4-2-3	배선 예	4-9
4-3	모터 연결(CN3).....	4-10
4-3-1	커넥터 핀아웃.....	4-10
4-3-2	와이어 크기	4-11
4-3-3	배선 예	4-12
4-4	안전 토크 꺼짐 STO(CN4)	4-13
4-4-1	커넥터 핀아웃.....	4-13
4-4-2	와이어 크기	4-13
4-4-3	배선 예	4-13
4-5	다이렉트 PWM 연결(CN5)	4-16
4-5-1	커넥터 핀아웃.....	4-16
4-6	전체 배선도 예	4-17
4-6-1	CK3A-G305L.....	4-17
4-6-2	CK3A-G310L.....	4-18
4-6-3	증폭기 결함 릴레이	4-19
4-7	EMC 지침을 준수하는 배선.....	4-20
4-7-1	주변 장비 연결 예	4-20

4-8	주변 구성 요소 선택	4-22
4-8-1	주 회로 AC 전원 공급	4-22
4-8-2	주 회로 DC 전원 공급 장치(저전압 모드)	4-22
4-8-3	로직 전원 공급	4-22
4-8-4	노이즈 필터	4-23
4-8-5	전압 서지 흡수기	4-23
4-8-6	배선용 차단기(MCCB, Molded Case Circuit Breaker)	4-23
4-8-7	외부 셉트 저항	4-24
4-8-8	다이렉트 PWM 케이블	4-24
4-8-9	호환 모터	4-24

4-1 주 회로 전원 및 셉트(CN1)

4-1-1 커넥터 핀아웃

CK3A-G305L

연결 커넥터	스프링 오프너
10핀 소켓 TE 연결 PN 1-2134249-0 ODT PN 040-000006 	TE 연결 PN 1981045-1 ODT PN 100-000032 
	

핀	기호	설명 및 사양
1	L1	주 회로 전원 입력. - L1-L2-L3에서 3상 170 ~ 252VAC 50/60Hz - L1-L2에서 단상 85 ~ 252VAC 50/60Hz, L3를 부동(개방)으로 둠 - L1-L2에서 단상 44 ~ 60VDC, L3를 부동(개방)으로 둠
2	L2	
3	L3	
4	NC	연결되지 않음(부동으로 둠)
5	NC	
6	B1	셉트 저항 단자. - 내부 셉트 저항의 경우, B2-B3을 단락하고 B1 부동 상태로 둠 - 외부 셉트 저항의 경우, B1-B2 사이에 저항을 연결하고 B3은 부동 상태로 둠
7	B2	
8	B3	
9	LV1	저전압 작동. 44 ~ 60VDC 작동을 위해 LV1-LV2 단락*1 - 단상 또는 3상 AC 작동을 위해 부동(개방) 상태로 둠
10	LV2	

*1. 컨트롤러의 저전압 모드에 대해 ADC 스트로브 워드도 설정해야 합니다.



안전한 사용을 위한 주의사항

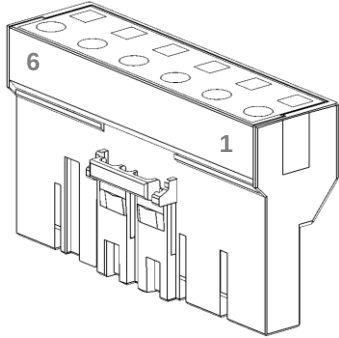
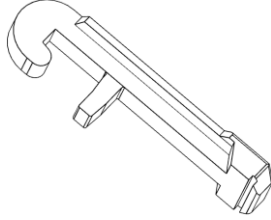
저전압 단락 와이어가 설치된 경우 AC 전원을 인가하지 마십시오.

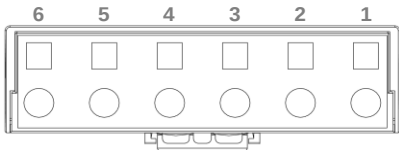


추가 정보

커넥터는 내부 셉트 저항용으로 설치된 단락 와이어 B2-B3과 함께 배송됩니다.

CK3A-G310L

연결 커넥터	스프링 오프너
6핀 소켓 TE 연결 PN 6-2260663-1 ODT PN 016-000006 	TE 연결 PN 2260828-1 ODT PN 100-000033 



핀	기호	설명 및 사양
1	L1	주 회로 전원 입력. - L1-L2-L3에서 3상 170 ~ 252VAC 50/60Hz - L1-L2에서 단상 85 ~ 252VAC 50/60Hz, L3를 부동(개방)으로 둠
2	L2	
3	L3	
4	B3	선트 저항 단자. - 내부 선트 저항의 경우, B2-B3을 단락하고 B1을 부동(개방) 상태로 둠 - 외부 선트 저항의 경우, B1-B2에 연결하고 B3을 부동(개방) 상태로 둠
5	B2	
6	B1	



추가 정보

커넥터는 내부 선트 저항용으로 설치된 단락 와이어 B2-B3과 함께 배송됩니다.

4-1-2 와이어 크기



올바른 사용을 위한 주의사항

단상 또는 3상 AC 주 전원에는 600VAC 이상의 케이블 배선을 사용하십시오.

주 회로 전원 공급 장치, 내부 저항 단락 와이어 및 저전압 모드 단락 와이어에 대한 권장 와이어 게이지가 아래에 나와 있습니다.

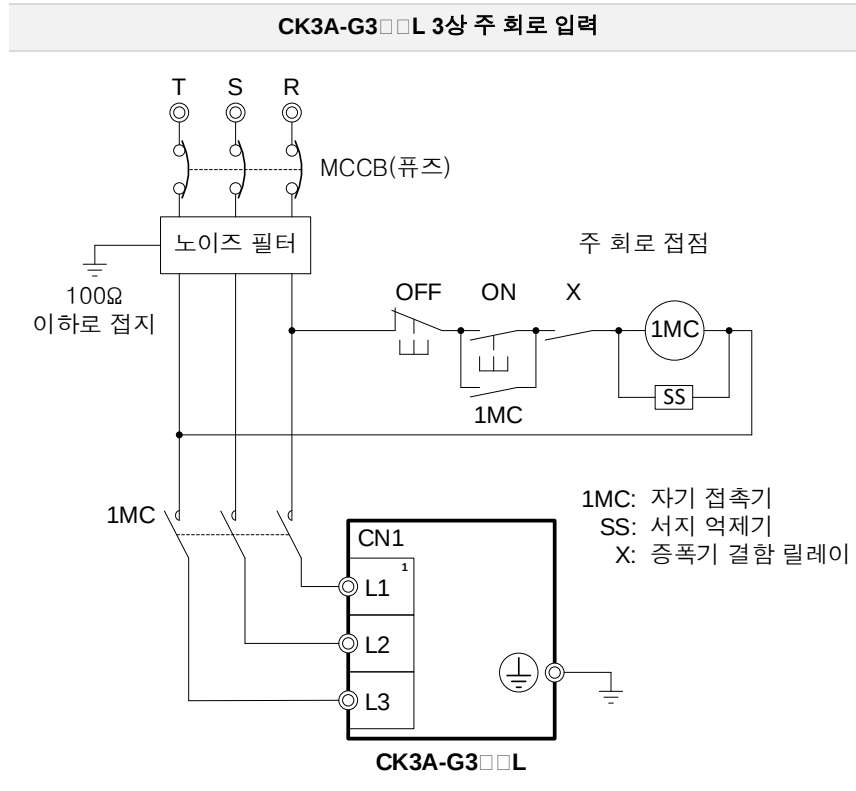
항목	CK3A-G305L	CK3A-G310L
주 회로 전원 및 내부 저항 단락 와이어	AWG 18 ~ 14 0.75 ~ 2.0mm ²	AWG 12 ~ 8 3.3 ~ 8.4mm ²
보호 접지 ⚡	AWG 12, 2.5mm ²	
나사 크기	M4	
조임 토크	1.2Nm	

참고 외부 선트 저항의 경우 저항과 함께 제공된 내장 와이어를 사용하십시오.

4-1-3 배선 예

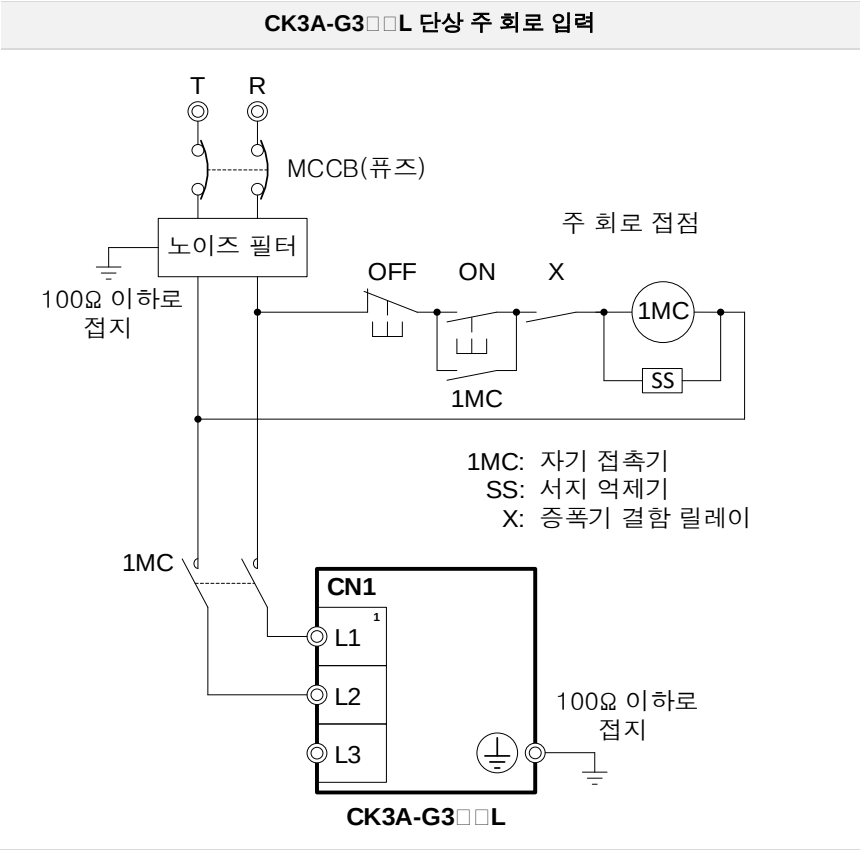
3상 AC

증폭기는 3상 AC 주 전원 입력으로 작동하도록 설계되었습니다. 다음 다이어그램은 이 배선 체계를 보여줍니다.



단상 AC

증폭기는 단상 AC 주 전원 입력으로 작동할 수 있습니다. 다음 다이어그램은 이 배선 체계를 보여줍니다.



DC 입력(저전압 모드)

CK3A-G305L은 저전압(44 ~ 60VDC) 주 회로 전원 입력으로 작동할 수 있습니다. 다음 다이어그램은 이 배선 체계를 보여줍니다.



안전한 사용을 위한 주의사항

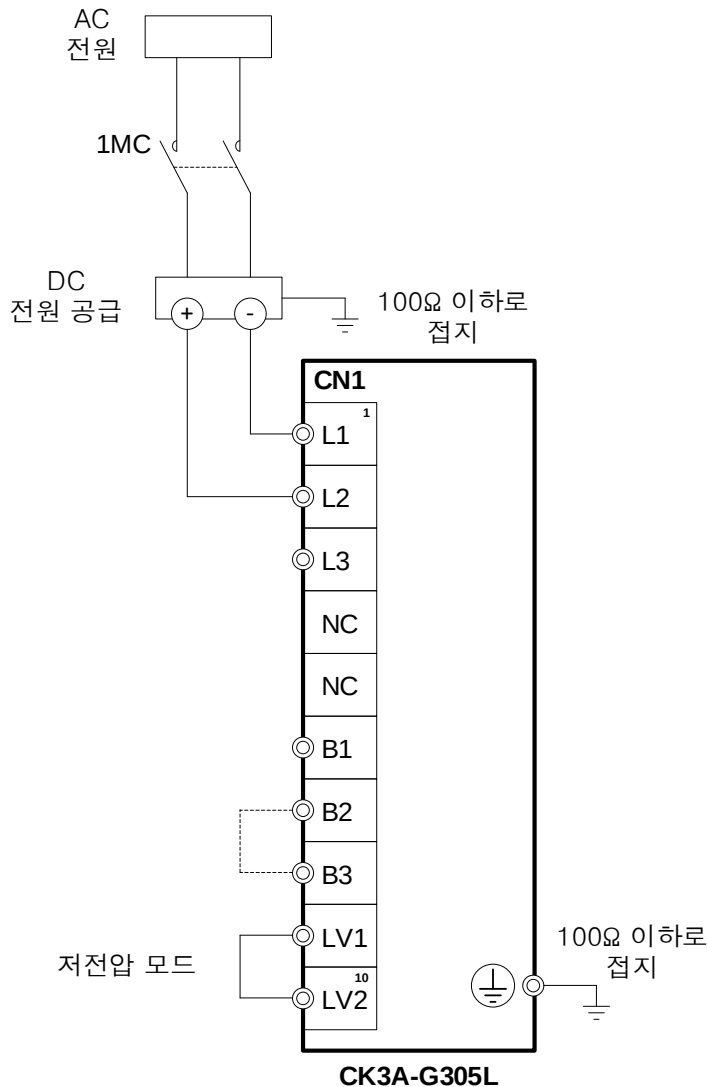
저전압 단락 와이어가 설치된 경우 AC 전원을 인가하지 마십시오.



올바른 사용을 위한 주의사항

저전압 작동을 성공적으로 수행하려면 컨트롤러의 ADC 스트로브 워드 설정을 적절하게 설정해야 합니다.

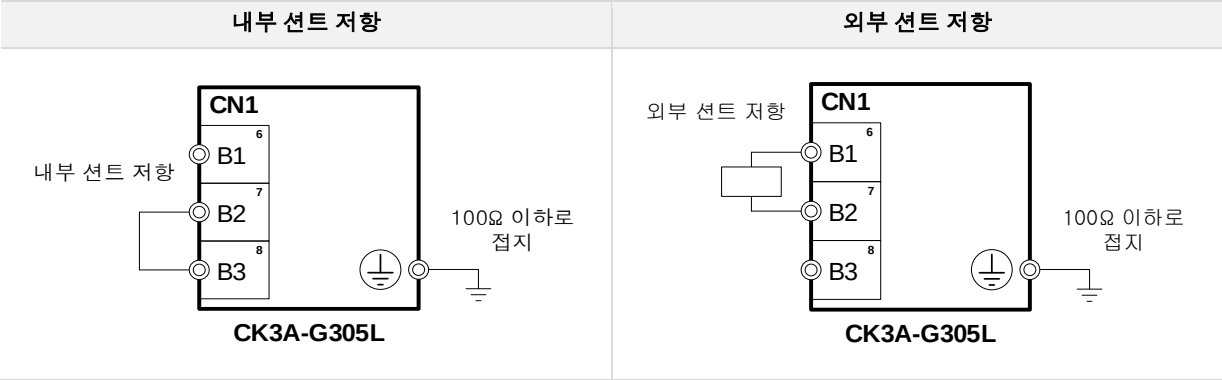
CK3A-G305L DC 주 회로 입력



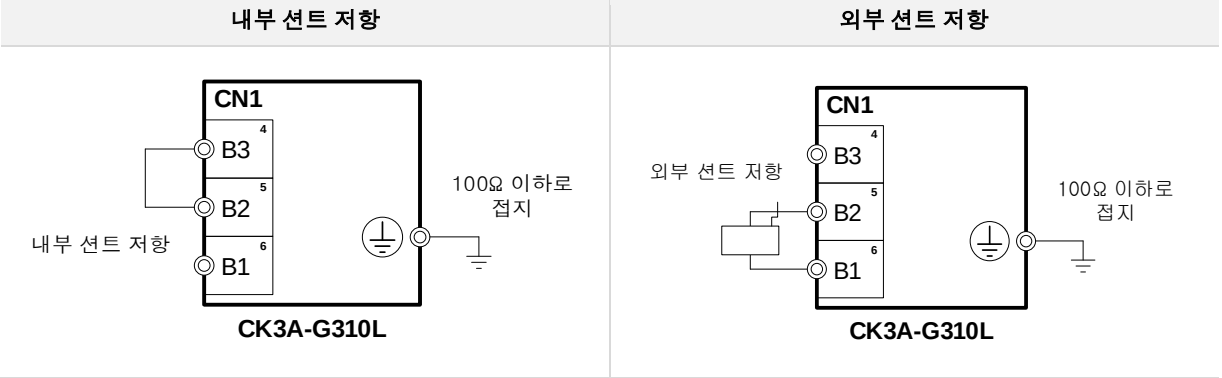
선티 저항 배선

다음 다이어그램은 내부 및 외부 선티 저항 배선 구성을 보여줍니다.

● CK3A-G305L



● CK3A-G310L

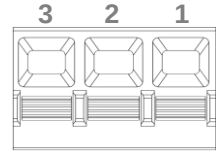
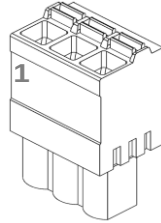


4-2 로직 전원 공급(CN2)

4-2-1 커넥터 핀아웃

연결 커넥터

3핀 소켓
Phoenix Contact PN 1754571
ODT PN 016-175457-1P3

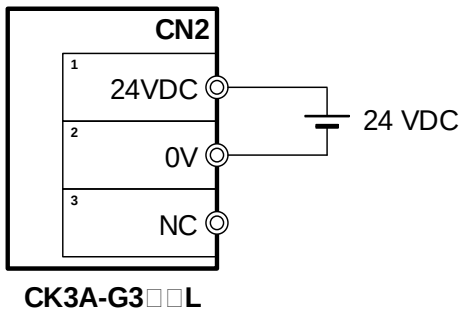


핀	기호	설명 및 사양
1	24VDC	로직 전원 공급 입력. • 22.0 ~ 26.4VDC • 전류 소비량 1.5A(5msec 동안 2.5A 돌입)
2	0V	
3	NC	

4-2-2 와이어 크기

로직 전원 공급 입력에 권장되는 와이어 게이지는 AWG 20 ~ 16, 0.5 ~ 1.5mm²입니다.

4-2-3 배선 예

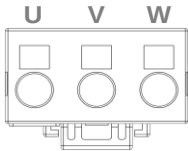


4-3 모터 연결(CN3)

4-3-1 커넥터 핀아웃

CK3A-G305L

연결 커넥터	스프링 오프너
3핀 소켓 TE 연결 PN 3-2229794-1 ODT PN 017-000002	TE 연결 PN 1981045-1 ODT PN 100-000032



핀	기호	설명 및 사양
1	U	모터 위상 출력 단자.
2	V	• 브러시리스 모터의 경우, U, V, W를 연결합니다.
3	W	• 브러시드 모터의 경우 U와 W를 연결하고 V를 부동(개방) 상태로 둡니다.



올바른 사용을 위한 주의사항

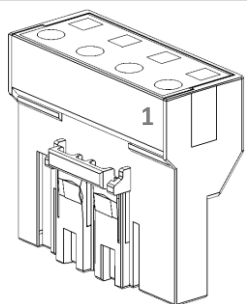
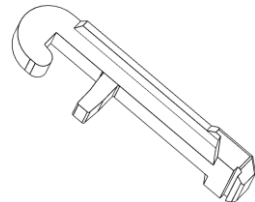
모터 프레임 접지를 증폭기의 보호 접지 ⚡ 장착대에 연결합니다.

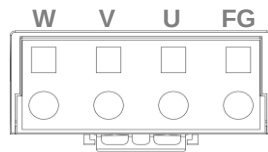


안전한 사용을 위한 주의사항

모터 커넥터가 연결되지 않은 상태에서 컨트롤러가 모터 명령을 전송하는 경우 증폭기에 오류가 발생하지 않습니다. 컨트롤러가 모터 명령을 전송하지 않는지 확인하거나 주 전원이 분리될 때까지 모터 커넥터를 연결하지 마십시오.

CK3A-G310L

연결 커넥터	스프링 오프너
4핀 소켓 TE 연결 PN 4-2260663-1 ODT PN 017-000001 	TE 연결 PN 2260828-1 ODT PN 100-000033 



핀	기호	설명 및 사양
1	FG	모터 위상 출력 단자. - FG는 프레임 접지이며 모터 프레임 접지선을 이 단자에 연결합니다. 3상 브러시리스 모터의 경우, U, V, W를 연결합니다. 2상 브러시드 모터의 경우 U와 W를 연결하고 V를 부동(개방) 상태로 둡니다.
2	U	
3	V	
4	W	

4-3-2 와이어 크기

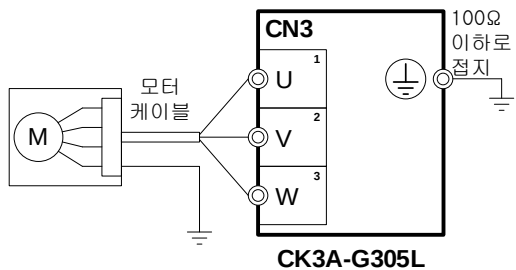
모터 위상 단자 및 프레임 접지에 권장되는 와이어 게이지는 다음과 같습니다.

항목	CK3A-G305L	CK3A-G310L
주 회로 전원 및 단락 와이어	AWG 18 ~ 14 0.75 ~ 2.0mm ²	AWG 14 ~ 8 2.0 ~ 8.4mm ²
보호 접지 Ⓢ	AWG 12, 2.5mm ²	
나사 크기	M4	
조임 토크	1.2Nm	

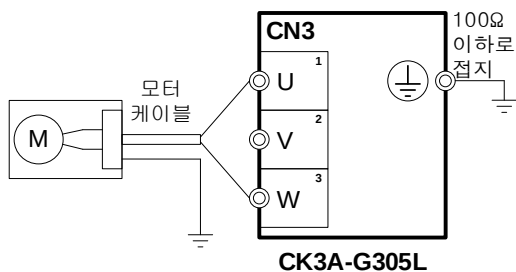
4-3-3 배선 예

CK3A-G305L

● 브러시리스 모터

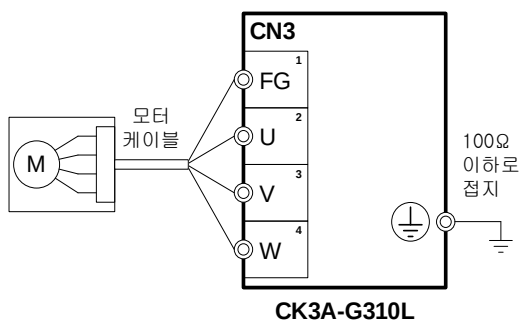


● 브러시드 모터

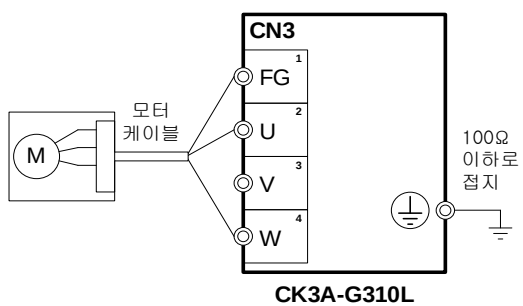


CK3A-G310L

● 브러시리스 모터



● 브러시드 모터

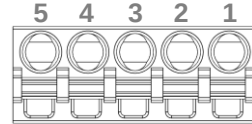
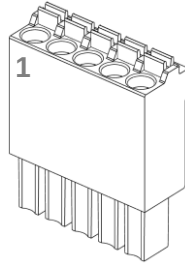


4-4 안전 토크 꺼짐 STO(CN4)

4-4-1 커넥터 핀아웃

연결 커넥터

5핀 소켓
Phoenix Contact PN 1745920
ODT PN 040-000005



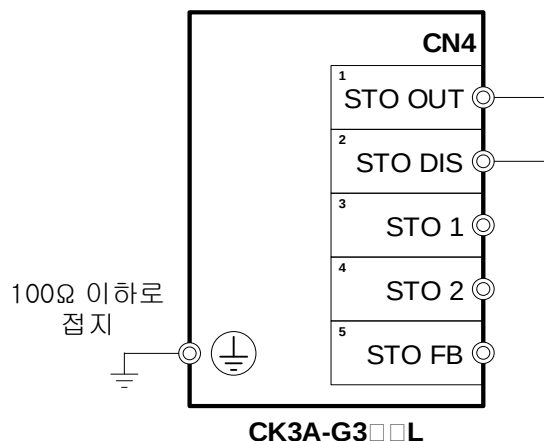
핀	기호	설명 및 사양
1	STO OUT	안전 토크 꺼짐 STO 제어 단자.
2	STO DIS	STO 기능을 비활성화하려면, STO OUT 및 STO DIS를 단락하고, 다른 모든 단자를 부동(개방) 상태로 둡니다.
3	STO IN1	STO 입력 1만 사용하려면, STO IN1과 STO OUT 사이에 릴레이 입력을 배선하고, STO IN2를 STO OUT에 연결한 후, STO DIS를 부동(개방) 상태로 둡니다.
4	STO IN2	- STO 입력 2만 사용하려면, STO IN2와 STO OUT 사이에 릴레이 입력을 배선하고, STO IN1을 STO OUT에 연결한 후, STO DIS를 부동(개방) 상태로 둡니다. - STO 입력 1과 2를 모두 사용하려면, STO IN1과 STO OUT 사이에 릴레이 입력 1을 배선하고, STO IN2와 STO OUT 사이에 릴레이 입력 2를 배선한 후, STO DIS를 부동(개방) 상태로 둡니다.
5	STO FB	안전 토크 꺼짐 STO 상태 단자. 이 단자는 STO 상태를 반영한 PNP 유형 출력입니다.

4-4-2 와이어 크기

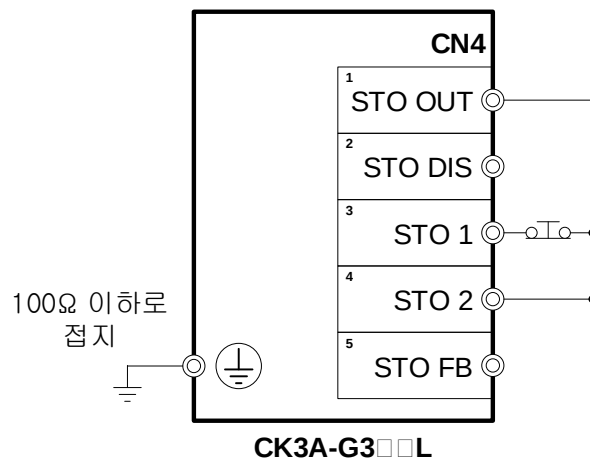
STO 기능에 권장되는 와이어 게이지는 AWG 20 ~ 16, 0.5 ~ 1.5mm²입니다.

4-4-3 배선 예

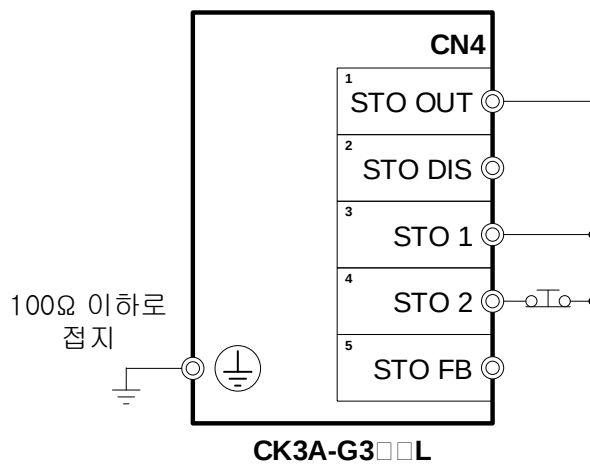
STO 비활성화



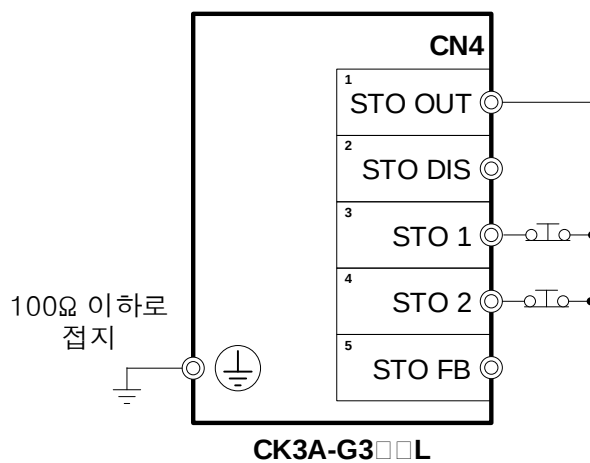
STO 1만 사용



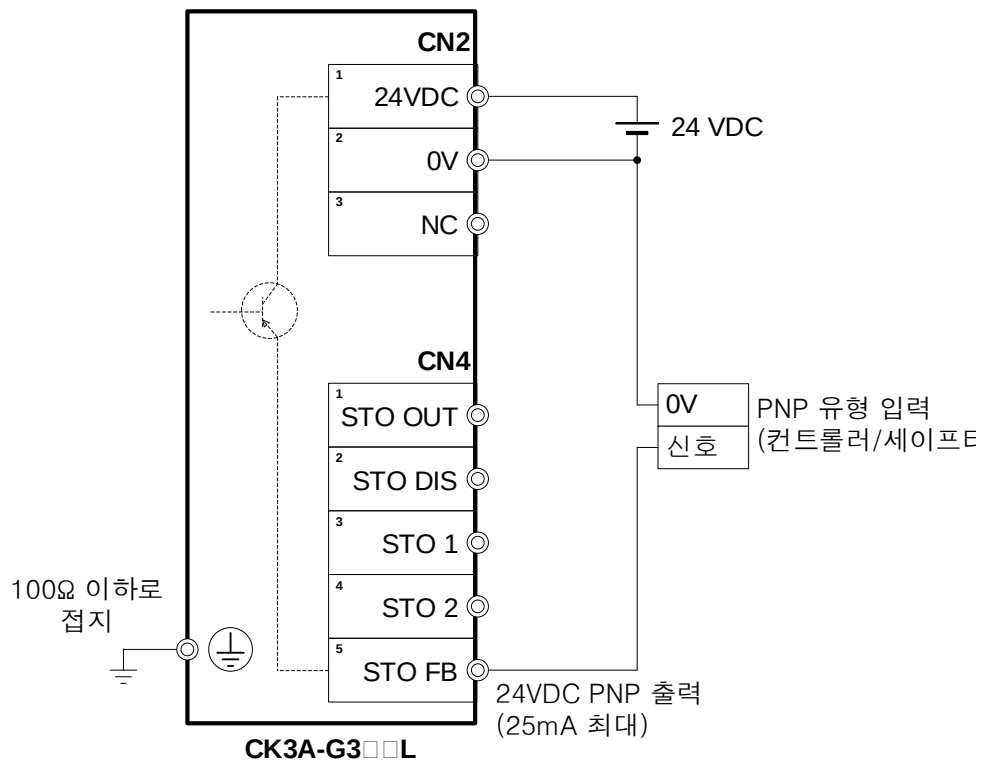
STO 2만 사용



STO 1 및 STO 2 모두 사용



STO 피드백



4-5 다이렉트 PWM 연결(CN5)

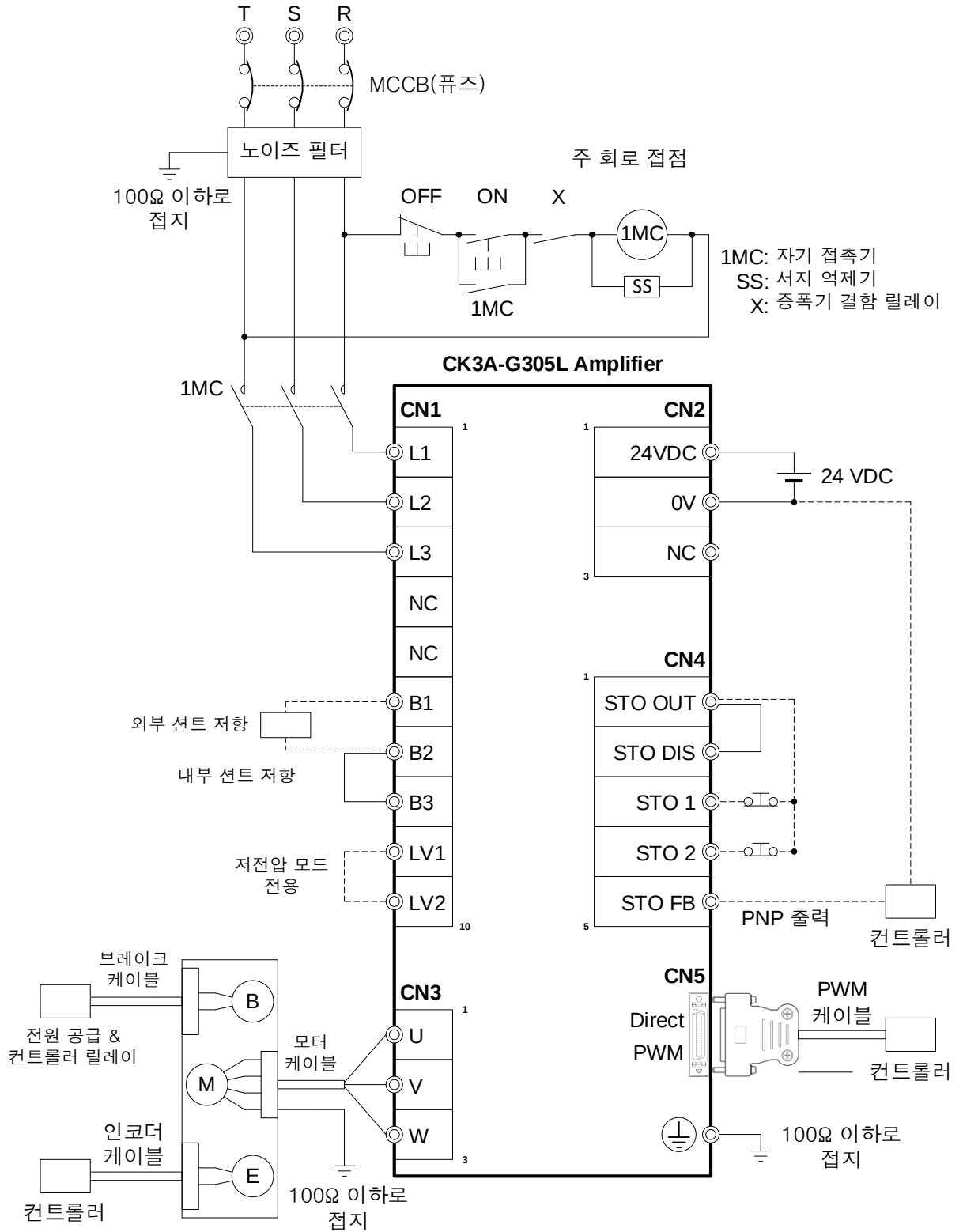
4-5-1 커넥터 핀아웃

핀	기호	설명	기능*1
1	PHACKL1+	위상 클록	입력
2	N.C.	연결 없음	-
3	ADCCLK1+	A/D 컨버터 클록	입력
4	ADCSTB1+	A/D 컨버터 스트로브	입력
5	ADCDAT1A+	A상 전류 데이터	출력
6	ADCDAT1B+	B상 전류 데이터	출력
7	AENA1+	증폭기 활성화	입력
8	FAULT1+	증폭기 결함	출력
9	PWMATOP1+	A상 위 cmd.	입력
10	PWMABOT1+	A상 아래 cmd.	입력
11	PWMBTOP1+	B상 위 cmd.	입력
12	PWMBBOT1+	B상 아래 cmd.	입력
13	PWMCTOP1+	C상 위 cmd.	입력
14	PWMCBOT1+	C상 아래 cmd.	입력
15	0V	기준 전압	일반
16	+5V	디지털 전력	입력
17	PWMDTOP1+	D상 위 cmd.	입력
18	PWMDBOT1+	D상 아래 cmd.	입력
19	PHACKL1-	위상 클록	입력
20	N.C.	연결 없음	-
21	ADCCLK1-	A/D 컨버터 클록	입력
22	ADCSTB1-	A/D 컨버터 스트로브	입력
23	ADCDAT1A-	A상 전류 데이터	출력
24	ADCDAT1B-	B상 전류 데이터	출력
25	AENA1-	증폭기 활성화	입력
26	FAULT1-	증폭기 결함	출력
27	PWMATOP1-	A상 위 cmd.	입력
28	PWMABOT1-	A상 아래 cmd.	입력
29	PWMBTOP1-	B상 위 cmd.	입력
30	PWMBBOT1-	B상 아래 cmd.	입력
31	PWMCTOP1-	C상 위 cmd.	입력
32	PWMCBOT1-	C상 아래 cmd.	입력
33	0V	기준 전압	일반
34	+5V	디지털 전력	입력
35	PWMDTOP1-	D상 위 cmd.	입력
36	PWMDBOT1-	D상 아래 cmd.	입력

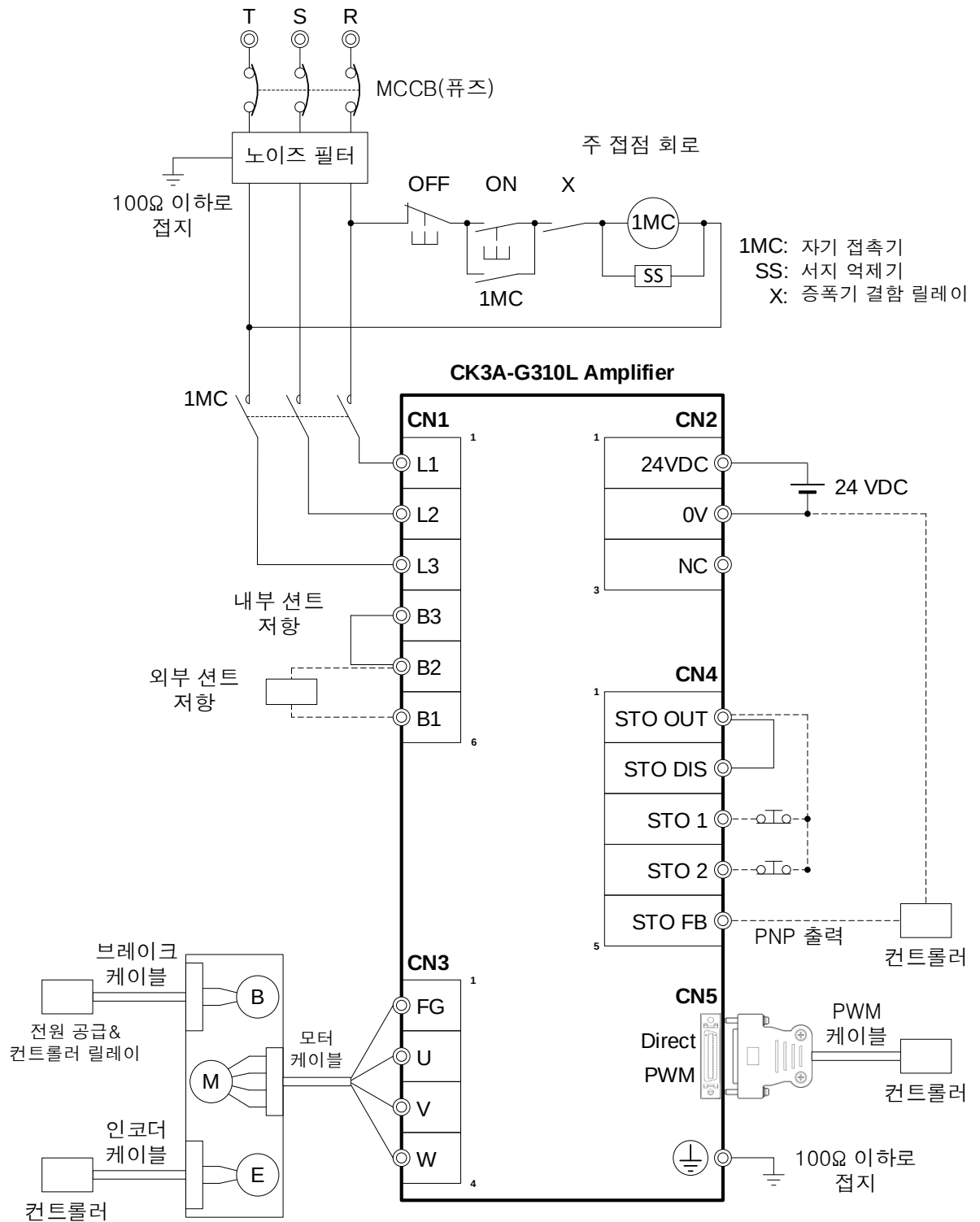
*1. 증폭기의 기능.

4-6 전체 배선도 예

4-6-1 CK3A-G305L



4-6-2 CK3A-G310L



4-6-3 증폭기 결함 릴레이

증폭기 결함이 있는 경우 주 전원 공급 장치를 분리하려면 릴레이 코일을 제어하는 출력 신호가 컨트롤러에서 나와야 합니다. 이는 해당 모터 증폭기 결함 비트를 미러링하는 PLC에서 수행할 수 있습니다.

샘플 PLC

```
PTR DigitalOutput1->Gate3[0].GpioData[0].16
```

```
OPEN PLC SamplePLC
```

```
DigitalOutput1 = Motor[1].AmpFault
```

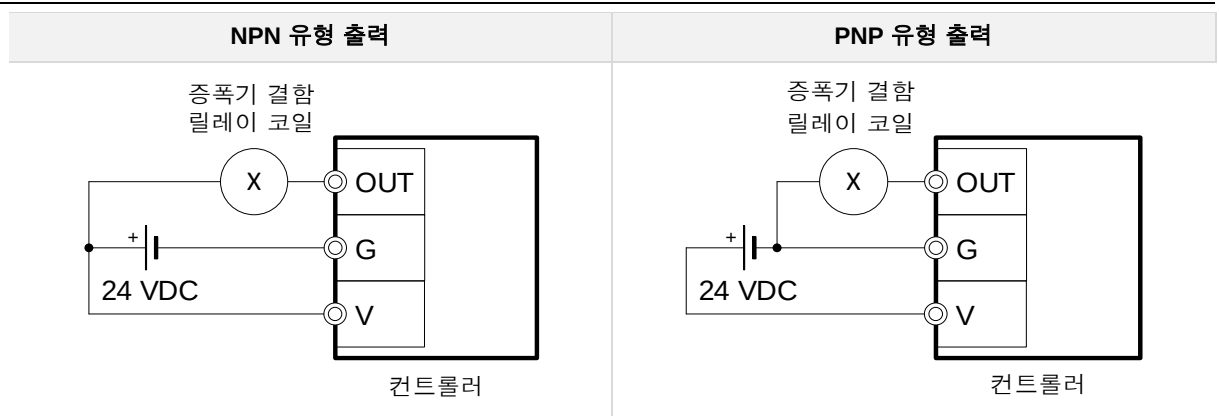
```
CLOSE
```



추가 정보

위 PLC의 코드 라인은 기존 활성 PLC와 병합될 수 있습니다. 자체 PLC일 필요는 없습니다.

배선 예



4-7 EMC 지침을 준수하는 배선

CK3A 다이렉트 PWM 증폭기는 이 섹션에 설명된 배선 조건에서 EMC 지침(EN 61800-3)을 준수합니다.

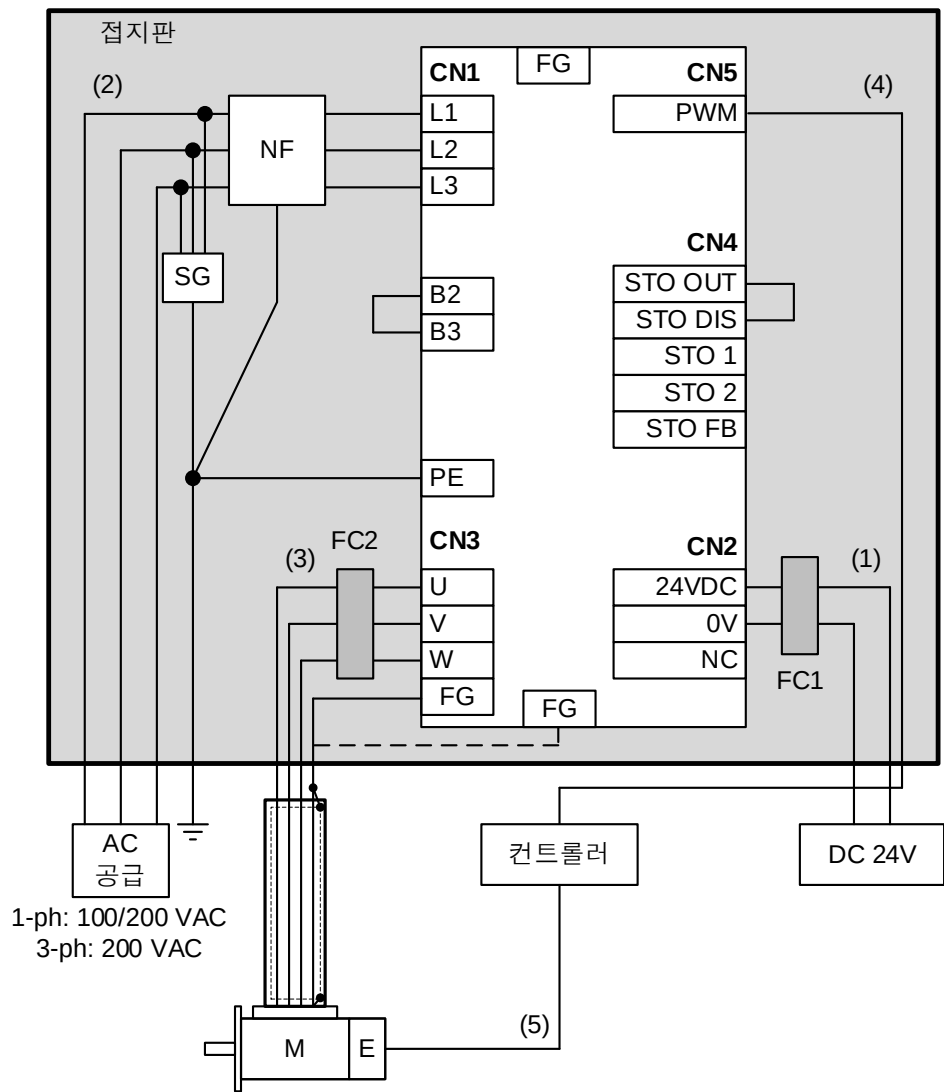
CK3A 시리즈가 EMC 지침을 준수할 수 있도록 다음 조건을 정했습니다.

장비에 제품을 장착할 때 고객은 머신 전체가 EMC 지침을 준수하는지 확인하기 위해 점검을 수행해야 합니다.

다음은 EMC 지침을 준수하기 위해 필요한 조건입니다.

- 증폭기를 접지판에 설치하십시오.
- 전력선에는 노이즈 필터와 낙뢰 서지 흡수 소자(서지 흡수기)를 설치하십시오.
- I/O 신호 및 인코더에는 편조 실드 케이블을 사용하십시오.
- 실드에는 주석 도금 연강 와이어를 사용해야 합니다.
- 각 케이블의 실드를 접지하십시오.

4-7-1 주변 장비 연결 예



참고 단상 입력의 경우 L1과 L2를 연결하십시오.

참고 CK3A-G305L의 경우 모터 커넥터에 FG가 없으므로 대신 FG 나사 단자를 사용하십시오.

- 위 다이어그램과 같이 장치 프레임 접지를 위한 접지판은 단일 지점으로 접지하십시오.
- 최소 두께 2.5mm²의 보호 접지선을 사용하고 보호 접지선이 가능한 한 짧도록 배선하십시오.
- 증폭기의 주 회로 커넥터 CN1 근처에 서지 흡수기와 노이즈 필터를 설치하십시오.

구성 요소 세부 정보

기호	이름	제조업체	모델	증폭기 모델	비고
SG	서지 흡수기	Soshin Electric Co., Ltd.	LT-C12G801WS	CK3A-G3□□L	단상 100/200VAC
			LT-C32G801WS		3상 200VAC
NF	노이즈 필터	OMRON	R88A-FI1S105	CK3A-G305L	단상 100/200VAC
			R88A-FI1S208		3상 200VAC
		Schaffner	FN 시리즈	CK3A-G310L	단상 100VAC
		OMRON	R88A-FI1S116		단상 200VAC
			R88A-FI1S216		3상 200VAC
FC1	페라이트 코어	NEC TOKIN	ESD-SR-250	CK3A-G3□□L	-
FC2	페라이트 코어	SEIWA ELECTRIC MFG	E04SR301334		-
M	모터	-	-	-	-
E	인코더	-	-	-	-
-	컨트롤러	OMRON	CK3M 시리즈	CK3A-G3□□L	다이렉트 PWM 포함
			UMAC 시리즈		다이렉트 PWM 포함

참고 자세한 사양은 타사 제조업체 또는 OMRON 판매 담당자에게 문의하십시오.

케이블 정보

번호	인터페이스	페라이트 코어	최대 길이	차폐 유무
(1)	로직 전원 공급 케이블	2회전	3m	아니요
(2)	주 회로 전원 공급 케이블	-	3m	아니요
(3)	모터 케이블	2회전	20m	예
(4)	다이렉트 PWM 케이블	-	3.6m	-
(5)	인코더 케이블	-	-	예

4-8 주변 구성 요소 선택

4-8-1 주 회로 AC 전원 공급

주 회로 AC 전원 공급 장치는 이 설명서의 사양 섹션에 나열된 사양을 준수하여 선택해야 합니다.

4-8-2 주 회로 DC 전원 공급 장치(저전압 모드)

주 회로 DC 전원 공급 장치는 이 설명서의 사양 섹션에 나열된 사양을 준수하여 선택해야 합니다.
다음은 권장되는 전원 공급 장치(48VDC)입니다.

전원 공급	제 조 업 체
S8VK-G 시리즈	OMRON
S8FS-G 시리즈	OMRON
S8FS-C 시리즈	OMRON



올바른 사용을 위한 주의사항

DC 전원 공급 장치 등급 선택은 적용 요구 사항에 따라 다릅니다.

- 동일한 전원 공급 장치를 사용하는 증폭기 수(총 유효 부하)
- 과도 또는 가속 전류(Amps @48VDC) 필요
- 정상 상태, 유휴 또는 정속 전류(Amps @48VDC) 필요

4-8-3 로직 전원 공급

로직 전원 공급 장치는 이 설명서의 사양 섹션에 나열된 사양을 준수하여 선택해야 합니다.
다음은 권장되는 전원 공급 장치(24VDC)입니다.

전원 공급	제 조 업 체
S8VK-G 시리즈	OMRON
S8VK-S 시리즈	OMRON

4-8-4 노이즈 필터

AC 전압 환경에서 라인 노이즈 필터는 양방향(시스템에서/시스템으로)으로 전자기 노이즈를 제거합니다. T 유형 필터는 권장되지 않습니다. PI 유형 필터가 권장됩니다.

- 노이즈 필터는 드라이브 및 전원과 동일한 패널에 장착해야 합니다.
- 노이즈 필터는 가능한 한 전원에 가깝게 장착해야 합니다.
- 노이즈 필터는 들어오는 캐비닛 전원에 최대한 가깝게 장착해야 합니다.
- 노이즈 필터는 이 설명서의 사양 섹션에 나열된 주 전원 입력 사양을 준수하여 선택해야 합니다.

다음은 권장되는 노이즈 필터입니다.

증폭기 모델	전압 입력	노이즈 필터	제조업체
CK3A-G305L	단상 100VAC	R88A-FI1S105	OMRON
	단상 200VAC	R88A-FI1S105	OMRON
	3상 200VAC	R88A-FI1S208	OMRON
CK3A-G310L	단상 100VAC	FN 시리즈	Schaffner
	단상 200VAC	R88A-FI1S116	OMRON
	3상 200VAC	R88A-FI1S216	OMRON

참고 자세한 사양은 해당 지역의 OMRON 영업 담당자에게 문의하십시오.

4-8-5 전압 서지 흡수기

서지 흡수기는 낙뢰 서지 전압과 전력 입력 라인의 비정상적인 전압으로부터 보호합니다.

다음은 권장되는 서지 흡수기입니다.

전압 입력	서지 흡수기	제조업체
단상 100VAC	LT-C12G801WS	Soshin Electric Co., Ltd.
단상 200VAC	LT-C12G801WS	Soshin Electric Co., Ltd.
3상 200VAC	LT-C32G801WS	Soshin Electric Co., Ltd.

참고 자세한 사양은 제조업체에 문의하십시오.

4-8-6 배선용 차단기(MCCB, Molded Case Circuit Breaker)

최대 입력 전류를 기준으로 배선용 회로 차단기를 선택하십시오.

- 증폭기의 순간 최대 출력은 정격 출력의 약 2배이며 최대 출력 지속 시간은 2초입니다.
따라서 정격 전류의 200%에서 10초 이상 동작할 수 있는 배선용 차단기를 선택하십시오.
- 정격 전류가 유효 부하 전류의 합보다 큰 배선용 차단기를 선택하십시오(증폭기가 여러 개인 경우).
- 배선용 회로 차단기를 선택할 때 다른 장치의 소비 전류를 추가해야 합니다.

추천 퓨즈

반드시 LISTED 및 ㉠ 마크가 있는 UL 인증 제품인 퓨즈를 사용하십시오. 온도 등급이 75℃ 이상인
구리 배선을 사용하십시오. 다음 표는 권장 퓨즈를 보여줍니다.

증폭기 모델	주 회로 공급	추천 퓨즈
CK3A-G305L	3상(170 ~ 252VAC)	UL CLASS RK5 10A
	단상(85 ~ 252VAC)	UL CLASS RK5 15A
	단상(44 ~ 60VDC)	UL CLASS RK5 10A
CK3A-G310L	3상(170 ~ 252VAC)	UL CLASS RK5 15A
	단상(85 ~ 252VAC)	UL CLASS RK5 25A

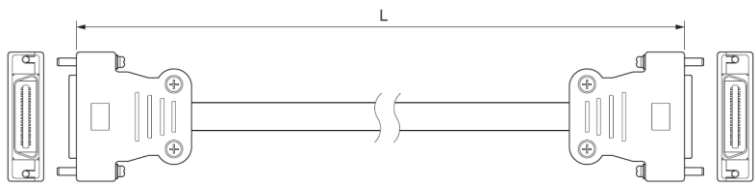
4-8-7 외부 셉트 저항

외부 셉트 저항은 이 설명서의 사양 섹션에 나열된 사양을 준수하여 선택해야 합니다. 다음은 권장되는
외부 셉트 저항입니다.

증폭기 모델	외부 셉트 저항	제조업체	사양
CK3A-G305L	R88A-RR30020	OMRON	60W, 20Ω
CK3A-G310L	R88A-RR30017	OMRON	60W, 17Ω

참고 자세한 사양은 해당 지역의 OMRON 영업 담당자에게 문의하십시오.

4-8-8 다이렉트 PWM 케이블



OMRON 모델	Delta Tau 모델	길이(L)
CK3W-CAAD009A	CABPWM-2(200-602739-036x)	900mm(36in)
CK3W-CAAD018A	CABPWM-4(200-602739-072x)	1.8m(72in)
CK3W-CAAD036A	CABPWM-6(200-602739-144x)	3.6m(144in)

4-8-9 호환 모터

CK3A 시리즈 제품 라인에는 다양한 모터에 연결할 수 있습니다. 증폭기는 AC/DC 브러시리스(동기식)
서보 모터리 또는 리니어 및 AC 유도(비동기식) 모터를 포함하여 거의 모든 유형의 3상 모터를 제어할
수 있습니다. 또한 증폭기는 보이스 코일 액추에이터(증폭기의 3상 중 2상 사용)와 같은 영구 자석 DC
브러시 모터를 제어할 수 있습니다.

모터 크기는 이 설명서의 범위를 벗어납니다. 모터 크기 조정은 다양한 적용 요구에 따라 달라지며 일반적으로 적절한 크기 조정 도구를 사용하여 수행됩니다.

모터 사양과 후보를 선택한 후 모터 사양서와 증폭기 사양서를 빠르게 비교하면 호환 여부를 알 수 있습니다. 기본적으로 고려해야 할 네 가지 주요 사양 매개변수가 아래 표에 나와 있습니다.

항목	CK3A-G305L	CK3A-G310L
정격 전압(주 회로 입력)	110 ~ 240VAC(85 ~ 252VAC)*1	
정격 전류	최대 5A _{rms}	최대 10A _{rms}
피크 전류	최대 2sec 동안 10A _{rms}	최대 2sec 동안 20A _{rms}
모터 인덕턴스	최소 1mH	

*1. 괄호 밖에 있는 것이 정격 값입니다. 괄호 안의 값은 허용 가능한 범위를 나타냅니다.

정격 전압

전압은 속도에 정비례합니다.

● 공급 전압보다 작거나 같은 모터 정격 전압

모터 정격 전압이 주 회로 공급 전압보다 작거나 같으면 모터 정격 속도를 달성할 수 있습니다(여기서는 문제 없음).



올바른 사용을 위한 주의사항

- 이 경우 컨트롤러의 Motor[x].PwmSf 설정은 다음과 같아야 합니다.
16,384 * (모터 전압 / 공급 전압).
- 손실을 감안하여 필요한 경우 10% 증가할 수 있습니다.

● 공급 전압보다 큰 모터 정격 전압

모터 정격 전압이 (최대) 주 회로 공급 전압보다 크면 모터는 작동하지만 정격 속도에 도달하지 못할 수 있습니다. 이 경우 모터 정격 속도가 용도에 필요한지 여부를 고려하십시오. 정격 속도가 필요하지 않은 경우 모터를 호환하여 사용할 수 있습니다. 정격 속도가 필요한 경우 모터를 호환하여 사용할 수 없습니다.



추가 정보

- 기본적으로 충분한 공급 전압과 적절한 전류 루프 튜닝을 통해 모터 정격/최고 속도를 달성할 수 있어야 합니다. 그러나 특정 용도에서는 피드백 장치 또는 과도한 부하와 같은 제한으로 인해 불가능할 수 있습니다.
- 일반적으로 최대 모터 속도는 선간 입력 전압을 모터의 역기전력 상수로 나눈 값으로 결정할 수 있습니다. 적절한 서보 적용을 위해 이 기능을 약간 저하시키는 것이 현명합니다.

정격 및 피크 전류

전류는 토크에 정비례합니다.

- 정격 전류는 일반적으로 정상 상태 위치 유지 또는 정속 모션과 관련이 있습니다.
- 피크 전류는 일반적으로 과도(가속) 모션과 관련이 있습니다.

● 증폭기 정격 전류보다 작거나 같은 모터 정격 전류

모터 정격 전류가 증폭기 정격 전류보다 작거나 같으면 모터가 호환 가능하고(여기서는 문제없음) 최적의 정상 상태 모션 성능을 달성할 수 있습니다.



안전한 사용을 위한 주의사항

이 경우 모터가 타는 것을 방지하려면 컨트롤러의 I2T 보호를 올바르게 설정해야 합니다.

● 증폭기 정격 전류보다 큰 모터 정격 전류

모터 정격 전류가 증폭기 정격 전류보다 크면 모터가 작동하지만 정상 상태 또는 일정한 모션에서 최적의 토크를 달성하지 못할 수 있습니다.



추가 정보

모션 애플리케이션에서는 약간 높은 등급의 모터를 사용하는 것이 상당히 일반적입니다. 대부분의 경우 위치 유지 또는 정속 중에는 정격 전류의 일부만 사용됩니다.

● 증폭기 피크 전류보다 작거나 같은 모터 피크 전류

모터 피크 전류가 증폭기 피크 전류보다 작거나 같으면 모터가 호환 가능하고(여기서는 문제없음) 최적의 과도(가속) 모션 성능을 달성할 수 있습니다.



안전한 사용을 위한 주의사항

이 경우 모터가 타는 것을 방지하려면 컨트롤러의 I2T 보호를 올바르게 설정해야 합니다.

● 증폭기 피크 전류보다 큰 모터 피크 전류

모터 피크 전류가 증폭기 피크 전류보다 크면 모터가 작동하지만 과도(가속) 모션에서 최적의 토크를 달성하지 못할 수 있습니다.

이 경우 최고 가속이 용도에 필요한지 여부를 고려하십시오. 최고 가속이 필요하지 않은 경우 모터를 호환하여 사용할 수 있습니다. 최고 가속이 필요한 경우 모터를 호환하여 사용할 수 없습니다.



추가 정보

피크 시간의 경우, 모션 애플리케이션에서의 최대 가속(피크 전류 인출) 프로파일은 일반적으로 500밀리초 미만임을 기억하십시오.

모터 인덕턴스

PWM 출력은 온-오프 전압 신호를 비교적 부드러운 전류 흐름으로 전환하고 적은 리플 유지를 위해 상당한 모터 인덕턴스를 필요로 합니다. 일반적으로 서보 모터의 모터 인덕턴스는 1 ~ 15mH입니다. CK3A 시리즈는 이 범위에서 쉽게 구동할 수 있습니다.

인덕턴스가 낮은 모터(1mH 미만)의 경우 모터에 큰 리플 전류가 흐르는 PWM 스위칭으로 인해 과도한 에너지 낭비와 발열을 유발하는 문제가 발생할 수 있습니다. 1mH 미만의 모터가 필요한 용도의 경우 외부 인덕터를 추가하여 인덕턴스를 높일 수 있습니다.

인덕턴스가 15mH를 초과하는 모터는 여전히 구동할 수 있지만 반응 속도가 느리며 일반적으로 고성능 서보 모터의 범위를 벗어납니다.

소프트웨어 구성

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 소프트웨어 구성에 대해 설명합니다.

5-1	소프트웨어 구성 개요	5-3
5-1-1	컨트롤러 증폭기 인터페이스	5-3
5-1-2	요약	5-3
5-1-3	대상 사용자	5-4
5-1-4	대상 하드웨어	5-4
5-1-5	Gate3 별칭 이름	5-4
5-2	기본 구성	5-5
5-2-1	PWM 주파수	5-5
5-2-2	Gate3 설정	5-5
5-2-3	채널 설정	5-5
5-2-4	모터 설정	5-6
5-3	ADC 스트로브 워드의 세부 정보	5-7
5-3-1	개요	5-7
5-3-2	명시적 주소	5-8
5-4	증폭기 기능의 세부 정보	5-9
5-4-1	기능	5-9
5-4-2	버스 방전 제어	5-9
5-4-3	결함 삭제	5-11
5-4-4	팬 제어	5-11
5-4-5	다이내믹 브레이크 제어	5-12
5-4-6	저전압(LV) 모드	5-13
5-5	ADC 레지스터의 세부 정보	5-14
5-5-1	ADC A의 정보	5-14
5-5-2	ADC B의 정보	5-14

5-5-3	ADC 레지스터 명시적 주소	5-15
5-6	상태 비트의 세부 정보	5-16
5-6-1	상태 비트 세트 1	5-16
5-6-2	상태 비트 세트 2	5-17
5-6-3	상태 비트 세트 3	5-18
5-7	증폭기 데이터의 세부 정보	5-19
5-7-1	개요 및 예	5-19
5-8	CK3A 샘플 PLC	5-20
5-8-1	샘플 PLC 구현	5-20
5-8-2	지연 타이머 서브루틴	5-21
5-8-3	사용자 공유 메모리에 대한 설명	5-22
5-8-4	CK3A 샘플 정의 4 모터	5-23
5-8-5	CK3A 샘플 PLC 4 모터	5-25
5-8-6	CK3A 샘플 정의 8 모터	5-27
5-8-7	CK3A 샘플 PLC 8 모터	5-31
5-8-8	전원을 켤 때 CK3A PLC 활성화	5-33
5-8-9	CK3A PLC 사용	5-34

5-1 소프트웨어 구성 개요

이 섹션에서는 더 나은 이해를 위해 소프트웨어 구성에 대한 개요를 제공합니다.

5-1-1 컨트롤러 증폭기 인터페이스

CK3A 다이렉트 PWM 증폭기는 PWM 케이블을 통해 컨트롤러와 인터페이스합니다. PC와 직접 통신하지 않으며 구성 설정을 포함하지 않습니다.

ADC 스트로브 워드(이후 섹션에서 설명)라는 컨트롤러의 구조 요소를 사용하여 증폭기 기능을 제어하고 컨트롤러에 보고되는 상태 비트 및 데이터 정보를 지정할 수 있습니다.

5-1-2 요약

다음 항목은 소프트웨어 구성 섹션에서 설명합니다.

항목	내용
기본 구성	증폭기의 기본 구성을 허용하는 ADC 스트로브 워드의 “기본” 설정을 포함한 요소 설정
ADC 스트로브 워드의 세부 정보	다음에 포함된 ADC 스트로브 워드 설명: • 증폭기 기능 필드 제어 • 상태 비트 필드 제어 • 데이터 보고 필드 제어
증폭기 기능의 세부 정보	방법 설명 및 예: • 디스플레이 오류 삭제 • 다이내믹 브레이크 구성 • 팬 작동 모드 제어 • 방전 모드 구성(주 전원 차단 시 잔류 에너지) • 저전압 작동 설정
ADC 레지스터의 세부 정보	상태 비트 및 데이터 정보가 상주하는 레지스터 목록
상태 비트의 세부 정보	모든 상태 비트의 목록 및 주소
데이터 정보의 세부 사항	데이터 정보 목록 및 주소
CK3A 샘플 PLC	사용자가 증폭기 기능을 쉽게 제어하고 상태 비트 및 데이터 정보를 읽을 수 있게 해주는 4개 및 8개 채널용 샘플 코드.



추가 정보

증폭기 기능, 상태 비트 및 데이터 보고는 CK3A 증폭기의 기본(기본) 작동에 필요하지 않습니다. 사용 필요성은 용도에 의해 결정됩니다.

5-1-3 대상 사용자

소프트웨어 구성 섹션에서는 Power PMAC Gate3 및 채널 아키텍처에 대한 일반적인 이해가 필요합니다.

5-1-4 대상 하드웨어

소프트웨어 구성 섹션은 다음 축 카드를 사용하는 Power PMAC 컨트롤러의 CK3M 및 UMAC 시리즈와 관련이 있습니다.

- CK3M AX□3□3□
- ACC-24E3 3-4002A5□5□-□□□-□□□

5-1-5 Gate3 별칭 이름

IDE 스크립트 환경에서 CK3WAX[i] 및 ACC-24E3[i]와 관련된 구조 요소는 하드웨어 별칭입니다. 이는 Gate3[i]와 동일합니다. 쉽게 설명하기 위해 소프트웨어 구성 섹션의 모든 예에서 Gate3[i] 명명법을 사용합니다.

5-2 기본 구성

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 기본 구성을 허용하는 최소 필요 설정에 대해 설명합니다.



추가 정보

- 모터 및 인코더 설정은 이 섹션에서 설명하지 않습니다. 이 부분은 부록 섹션을 참조하십시오.
- 전원을 켜다가 켜 후 CK3A 다이렉트 PWM 증폭기를 올바르게 시작하려면 이 섹션에서 설명하는 설정을 Power PMAC 메모리(및 IDE 프로젝트)에 저장해야 합니다.

5-2-1 PWM 주파수

CK3A 다이렉트 PWM 증폭기가 올바르게 작동하려면 PWM 주파수가 [8 ~ 20] KHz 범위에 있어야 합니다. PWM 주파수는 Power PMAC 위상 주파수(Gate3[i].PhaseFreq 설정) 및 PWM 주파수(Gate3[i].Chan[j].PwmFreqMult 설정)에 의해 결정되며 다음과 같은 관계가 있습니다.

$$\text{PWM Frequency} = \frac{\text{Gate3[i].Chan[j].PwmFreqMult} + 1}{2} \times \text{Phase Frequency}$$

5-2-2 Gate3 설정

```
Gate3[i].AdcAmpStrobe = $901001
Gate3[i].AdcAmpHeaderBits = 4
Gate3[i].AdcAmpClockDiv = 5
```



올바른 사용을 위한 주의사항

4개의 Gate3[i] 증폭기 중 하나가 저전압(48VDC) 모드에서 사용될 CK3A-G305L 모델인 경우 Gate3[i].AdcAmpStrobe = \$B01001로 설정합니다.

5-2-3 채널 설정

CK3A-G305L

```
Gate3[i].Chan[j].PwmDeadTime = 2 / 0.0533
Gate3[i].Chan[j].PackInData = 0
Gate3[i].Chan[j].PackOutData = 0
```

CK3A-G310L

```
Gate3[i].Chan[j].PwmDeadTime = 3.1 / 0.0533
Gate3[i].Chan[j].PackInData = 0
Gate3[i].Chan[j].PackOutData = 0
```

5-2-4 모터 설정

```
Motor[x].AdcMask = $FFFF0000
```

```
Motor[x].CurrentNullPeriod = 0
```

PMAC I2T 설정의 경우 각 모델의 최대 ADC 값은 다음과 같습니다.

모델	최대 ADC
CK3A-G305L	15.735
CK3A-G310L	31.470



안전한 사용을 위한 주의사항

Motor[x].I2tSet, Motor[x].I2tTrip 및 Motor[x].MaxDac를 포함한 PMAC I2T 설정을 올바르게 설정해야 합니다. 모터의 정격이 증폭기보다 낮은 경우에는 특히 그렇습니다. 그렇게 하지 않으면 모터가 탈 위험이 있습니다.

5-3 ADC 스트로브 워드의 세부 정보

이 섹션에서는 사용자가 소프트웨어 설정을 통해 다음을 수행할 수 있는 ADC 스트로브 워드의 세부 정보를 설명합니다.

- 증폭기 기능 제어
- 증폭기 상태 비트에 액세스
- 증폭기 데이터 정보에 액세스

ADC 스트로브 워드의 구조 요소는 Gate3[i].AdcAmpStrobe입니다.

5-3-1 개요

IDE 스크립트 환경에서 ADC 스트로브 워드는 CK3A 다이렉트 PWM 증폭기와 관련된 다음 필드로 구성된 24비트 요소입니다.

Hex	9				0				1				0				0				1			
Bit #	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Fixed

Dynamic brake ctrl: =1 OFF, =0 ON

Low voltage mode: =0 OFF, =1 ON

Clear faults: =1 then =0

Fan ctrl: =0 AUTO, =1 ON

Bus discharge ctrl: =0 OFF, =1 ON

Fixed

=0: Report status bits set 2
=1: Report status bits set 3
=2: Report voltage in VDC (÷4)
=3: Report temperature in °C
=6: Report firmware version
=7: Report current rating
Other settings are for internal use

5-3-2 명시적 주소

Gate3[i].AdcAmpStrobe 구조 요소는 비트별(개별 비트) 매핑에 사용할 수 없습니다. 필요한 경우 대신 명시적 주소 지정을 사용해야 합니다.

다음은 일반적으로 처음 32개의 증폭기 채널에 해당하는 처음 8개의 Gate3[i] 인덱스에 대한 주소 목록입니다.

인덱스 i	구조 요소	채널	주소
0	Gate3[0].AdcAmpStrobe	1 ~ 4	\$90020C
1	Gate3[1].AdcAmpStrobe	5 ~ 8	\$90420C
2	Gate3[2].AdcAmpStrobe	9 ~ 12	\$90820C
3	Gate3[3].AdcAmpStrobe	13 ~ 16	\$90C20C
4	Gate3[4].AdcAmpStrobe	17 ~ 20	\$91020C
5	Gate3[5].AdcAmpStrobe	21 ~ 24	\$91420C
6	Gate3[6].AdcAmpStrobe	25 ~ 28	\$91820C
7	Gate3[7].AdcAmpStrobe	29 ~ 32	\$91C20C



추가 정보

- 달러(\$) 기호는 PMAC 스크립트 언어의 16진수 값을 나타냅니다.
- 명시적 주소는 Gate3[i].AdcAmpStrobe.a에서 Sys.piom을 빼고 결과를 16진수로 읽어서 찾습니다.

5-4 증폭기 기능의 세부 정보

이 섹션에서는 컨트롤러에서 설정할 수 있는 증폭기 기능에 대해 설명합니다.

5-4-1 기능

ADC 스트로브 워드에 의해 제어되는 증폭기 기능을 통해 사용자는 다음을 수행할 수 있습니다.

- 버스 방전 모드 설정
- 디스플레이 오류 삭제(리셋)
- 다이내믹 브레이크 모드 설정
- 팬 작동 모드 설정
- 저전압 모드 설정



추가 정보

- 증폭기 기능은 동일한 Gate3[i]에 있는 4개의 증폭기 채널에 전역적으로 적용됩니다.
- 이 섹션의 예를 사용하여 ADC 스트로브 워드의 원하는 설정을 찾을 수 있습니다. 이 설정은 이후 전원을 켤 수 있도록 IDE 프로젝트 파일 또는 구성에 저장해야 합니다.

5-4-2 버스 방전 제어

ADC 스트로브 워드의 비트 #5는 주 회로 전원이 차단될 때 증폭기의 잔류 에너지 방전 모드를 제어합니다.

= 0 방출 끄기(권장)

ADC 스트로브 워드의 비트 #5가 0으로 설정되어 있고 주 회로 전원이 제거되면 증폭기가 잔류 에너지를 천천히 방출합니다. 주 전원 사이클 사이에 허용되는 시간이 짧기 때문에 이 모드를 사용하는 것이 좋습니다.



올바른 사용을 위한 주의사항

이 모드에서는 주 전원 사이클 사이에 최소 10초의 시간을 두어야 합니다.



추가 정보

이 모드에서 잔류 에너지(< 36VDC)를 방전하는 데는 약 5분이 걸립니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 버스 방전 끄기에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFDF) | $0
```

= 1 방출 켜기

ADC 스트로브 워드의 비트 #5가 1로 설정되어 있고 주 회로 전원이 제거되면 증폭기가 내부 또는 외부 셉트 저항에 잔류 에너지를 빠르게 방출합니다.



안전한 사용을 위한 주의사항

셉트 저항이 연결되지 않은 경우 증폭기가 충전된 상태로 남아있게 됩니다.

- 매번 성공적인 완전 방전을 위해 사용자는 주 회로 전원 사이클 사이에 최소 1분 이상 기다려야 합니다.
- 이 기능은 주로 수리 시 장치를 열고 안전하게 작업해야 하는 경우에 유용합니다.
- 이 기능은 안전을 위해 비상 정지 또는 유지보수 중에 머신 장비에서 모든 에너지를 제거해야 하는 일부 시스템에 유용할 수 있습니다.



올바른 사용을 위한 주의사항

이 모드에서:

- 사용자는 주 회로 전원 사이클 사이에 최소 1분 이상 기다려야 합니다.
- 정상적인 작동을 위해 셉트 저항(내부 또는 외부)을 연결해야 합니다.



추가 정보

이 모드에서 잔류 에너지(< 36VDC)를 방전하는 데는 약 2초가 걸립니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 버스 방전 켜짐에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFDF) | $20
```

5-4-3 결함 정상화

ADC 스트로브 워드의 비트 #16은 결함 조건이 제거된 후 증폭기 디스플레이 결함을 정상화합니다.



추가 정보

- 소프트 스타트(A2), ADC 오프셋(A9) 및 단락(AC) 결함을 삭제하려면 증폭기의 로직 전원 사이클이 필요합니다.
- 이 기능은 증폭기 디스플레이 오류만 삭제합니다. PMAC의 증폭기 결함 (Motor[x].AmpFault) 상태는 삭제되지 않습니다.
- PMAC에서 증폭기 결함 상태를 삭제하는 일반 명령은 없습니다. 결함 조건이 제거되고 모터가 활성화되면 자동으로 삭제됩니다.

결함 작동 정상화

4개의 Gate3[i] 증폭기 중 하나에서 표시 결함을 지우려면 ADC 스트로브 워드의 비트 #16을 최소 10밀리초 동안 1로 설정한 다음, 정상 작동을 위해 다시 0으로 설정합니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 ADC 스트로브 워드 비트 #16을 1로 설정한 다음 0으로 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FEFFFF) | $10000
```

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FEFFFF) | $0
```

5-4-4 팬 제어

ADC 스트로브 워드의 비트 #17은 증폭기 팬 작동을 제어합니다.

= 0 팬 자동

ADC 스트로브 워드의 비트 #17이 0으로 설정되면 팬은 4개의 Gate3[i] 증폭기 모두에 대해 자동 모드로 설정됩니다. 전원 모듈의 온도가 35°C에 도달하면 켜지고, 다시 32°C로 내려가면 꺼집니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 팬 자동 모드에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FDFFFF) | $0
```

= 1 팬 항상 켜기

ADC 스트로브 워드의 비트 #17이 1로 설정되면 팬은 4개의 Gate3[i] 증폭기 모두에 대해 항상 켜기로 설정됩니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 팬 항상 켜기에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FDFFFF) | $20000
```

5-4-5 다이내믹 브레이크 제어

ADC 스트로브 워드의 비트 #20은 다이내믹 브레이크 기능을 제어합니다.



안전한 사용을 위한 주의사항

- 다이내믹 브레이크 기능은 오류 발생 시 정지를 위한 것이므로 단시간 정격을 갖습니다. 정상적인 작동을 멈추기 위해 사용하지 마십시오.
- 전원 공급이 꺼져 있고 다이내믹 브레이크가 켜진 경우 외부 구동원을 사용하여 모터를 구동하지 마십시오.
- 일반적인 지침으로 다이내믹 브레이크 사용 사이에 최소 1분 이상의 시간을 두십시오.
- 카운터 밸런스 없이 수직 모터를 정지시키기 위해 다이내믹 브레이크에 의존하지 마십시오.
- 증폭기가 활성화된 경우 다이내믹 브레이크 설정이 적용되지 않습니다. 다이내믹 브레이크 기능을 켜기/끄기로 설정하기 전에 증폭기 활성화 신호를 꺼야 해야 합니다.

다이내믹 브레이크 기능이 켜짐으로 설정되면 정지(서보 꺼짐) 이벤트 후 모터가 관성 회전하는 것을 방지합니다. 이 상황은 다음의 경우에 해당합니다.

- 치명적 후속 오류 등 사용자 명령 또는 PMAC 기능에 의해 모터가 정지함(서보 꺼짐)
- 주 회로 전원 끄기 또는 로직 전원 끄기를 포함한 증폭기 오류

= 0 다이내믹 브레이크 켜기

ADC 스트로브 워드의 비트 #20이 0으로 설정되면 다이내믹 브레이크는 4개의 Gate3[i] 증폭기 모두에 대해 켜기로 설정됩니다.



올바른 사용을 위한 주의사항

- 다이내믹 브레이크 기능이 켜져 있으면 사용자는 증폭기를 활성화하고 모션을 명령하기 전에 최소 50밀리초를 기다려야 합니다.
- 다이내믹 제동이 수행된 경우(작동 중인 모터가 정지한 후) 사용자는 증폭기를 다시 활성화하기 전에 최소 500밀리초를 기다려야 합니다.
- 다이내믹 브레이크 기능이 켜지고 모터가 작동 중인 경우 사용자는 정지(서보 꺼짐) 명령을 내리기 전에 먼저 JOG/를 실행하여 모터를 감속하여 정지시켜야 합니다.
- 증폭기가 활성화된 경우 다이내믹 브레이크 설정 켜기/끄기를 전환하지 마십시오.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 다이내믹 브레이크 기능 켜기에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFFF) | $0
```

= 1 다이내믹 브레이크 끄기

ADC 스트로브 워드의 비트 #20이 1로 설정되면 다이내믹 브레이크는 4개의 Gate3[i] 증폭기 모두에 대해 끄기로 설정됩니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 다이내믹 브레이크 기능 끄기에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFFF) | $100000
```

5-4-6 저전압(LV) 모드

ADC 스트로브 워드의 비트 #21은 증폭기를 저전압(44 ~ 60VDC) 주 회로 전원 공급 장치와 함께 사용할지 여부를 지정합니다.

- 이 모드는 CK3A-G305L 모델에서만 사용할 수 있습니다.
- CK3A-G310L 모델의 경우 이 기능을 사용하려면 공장에서 수정해야 합니다.
- 4개의 Gate3[i] 증폭기 중 하나 이상이 저전압 범위에서 사용되는 경우 ADC 스트로브 워드를 반드시 저전압 모드로 설정해야 합니다. 동일한 Gate3[i]의 다른 증폭기는 여전히 AC 입력과 함께 사용할 수 있습니다.



추가 정보

저전압 모드가 설정되고 주 전원(예: 48VDC)이 인가되면 증폭기는 왼쪽 7 세그먼트 디스플레이에 L을 표시합니다.

= 0 일반 모드

ADC 스트로브 워드의 비트#21이 0으로 설정되면 4개의 Gate3[i] 증폭기 중 어느 것도 저전압 범위에서 작동할 수 없습니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 일반 모드에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $DFFFFFF) | $0
```

= 1 저전압 모드

ADC 스트로브 워드의 비트 #21이 1로 설정되면 4개의 Gate3[i] 증폭기 중 하나가 44 ~ 60VDC 주 전원 입력 범위에서 작동할 수 있습니다.

다음 예에서는 일반적으로 증폭기 채널 1 ~ 4에 연결된 Gate3 인덱스 0을 사용하여 IDE 터미널 창에서 저전압 모드에 대한 ADC 스트로브 워드를 설정하는 빠른 방법을 보여줍니다.

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $DFFFFFF) | $200000
```



올바른 사용을 위한 주의사항

저전압 범위 주 전원으로 올바르게 작동하려면 주 회로 전원 커넥터(CN1)의 저전압 단락 와이어를 설치해야 합니다.

5-5 ADC 레지스터의 세부 정보

이 섹션에서는 ADC 레지스터의 세부 정보를 설명합니다.

Power PMAC 컨트롤러(일반적으로 한 증폭기에 연결됨)의 각 채널에는 다음과 같이 정의된 두 개의 ADC 레지스터가 있습니다.

이름	구조 요소
ADC A	Gate3[i].Chan[j].AdcAmp[0]
ADC B	Gate3[i].Chan[j].AdcAmp[1]

5-5-1 ADC A의 정보

ADC A는 다음 정보를 전달하는 32비트 레지스터입니다.

비트	ADC A의 정보	ADC 스트로브 워드 설정
7:0	내부 사용	-
15:8	상태 비트 세트 1	"올바른" 스트로브 워드 설정
31:16	모터 위상 U 전류 피드백	"올바른" 스트로브 워드 설정

5-5-2 ADC B의 정보

ADC B는 다음 정보를 전달하는 32비트 레지스터입니다.

비트	ADC B의 정보	ADC 스트로브 워드 설정
7:0	내부 사용	-
15:08	상태 비트 세트 2	비트 [11:08] = 0
	상태 비트 세트 3	비트 [11:08] = 1
	DC 버스 전압	비트 [11:08] = 2
	전원 모듈 온도	비트 [11:08] = 3
	증폭기 펄스폭	비트 [11:08] = 6
	증폭기 전류 등록	비트 [11:08] = 7
31:16	모터 위상 V 전류 피드백	"올바른" 스트로브 워드 설정



추가 정보

비트 [11:08]의 다른 모든 설정은 내부적으로 사용되는 데이터를 보고합니다.

5-5-3 ADC 레지스터 명시적 주소

Gate3[i].Chan[j].AdcAmp[k] 구조 요소는 비트별(개별 비트) 매핑에 사용할 수 없습니다. 대신 명시적 주소 지정을 사용해야 합니다. 다음은 일반적으로 처음 32개의 증폭기 채널에 해당하는 처음 8개의 Gate3[i] 인덱스에 대한 주소 목록입니다.

	채널	ADC A 구조 요소	주소	ADC B 구조 요소	주소
Gate3[0]	1	Gate3[0].Chan[0].AdcAmp[0]	\$900020	Gate3[0].Chan[0].AdcAmp[1]	\$900024
	2	Gate3[0].Chan[1].AdcAmp[0]	\$9000A0	Gate3[0].Chan[1].AdcAmp[1]	\$9000A4
	3	Gate3[0].Chan[2].AdcAmp[0]	\$900120	Gate3[0].Chan[2].AdcAmp[1]	\$900124
	4	Gate3[0].Chan[3].AdcAmp[0]	\$9001A0	Gate3[0].Chan[3].AdcAmp[1]	\$9001A4
Gate3[1]	5	Gate3[1].Chan[0].AdcAmp[0]	\$904020	Gate3[1].Chan[0].AdcAmp[1]	\$904024
	6	Gate3[1].Chan[1].AdcAmp[0]	\$9040A0	Gate3[1].Chan[1].AdcAmp[1]	\$9040A4
	7	Gate3[1].Chan[2].AdcAmp[0]	\$904120	Gate3[1].Chan[2].AdcAmp[1]	\$904124
	8	Gate3[1].Chan[3].AdcAmp[0]	\$9041A0	Gate3[1].Chan[3].AdcAmp[1]	\$9041A4
Gate3[2]	9	Gate3[2].Chan[0].AdcAmp[0]	\$908020	Gate3[2].Chan[0].AdcAmp[1]	\$908024
	10	Gate3[2].Chan[1].AdcAmp[0]	\$9080A0	Gate3[2].Chan[1].AdcAmp[1]	\$9080A4
	11	Gate3[2].Chan[2].AdcAmp[0]	\$908120	Gate3[2].Chan[2].AdcAmp[1]	\$908124
	12	Gate3[2].Chan[3].AdcAmp[0]	\$9081A0	Gate3[2].Chan[3].AdcAmp[1]	\$9081A4
Gate3[3]	13	Gate3[3].Chan[0].AdcAmp[0]	\$90C020	Gate3[3].Chan[0].AdcAmp[1]	\$90C024
	14	Gate3[3].Chan[1].AdcAmp[0]	\$90C0A0	Gate3[3].Chan[1].AdcAmp[1]	\$90C0A4
	15	Gate3[3].Chan[2].AdcAmp[0]	\$90C120	Gate3[3].Chan[2].AdcAmp[1]	\$90C124
	16	Gate3[3].Chan[3].AdcAmp[0]	\$90C1A0	Gate3[3].Chan[3].AdcAmp[1]	\$90C1A4
Gate3[4]	17	Gate3[4].Chan[0].AdcAmp[0]	\$910020	Gate3[4].Chan[0].AdcAmp[1]	\$910024
	18	Gate3[4].Chan[1].AdcAmp[0]	\$9100A0	Gate3[4].Chan[1].AdcAmp[1]	\$9100A4
	19	Gate3[4].Chan[2].AdcAmp[0]	\$910120	Gate3[4].Chan[2].AdcAmp[1]	\$910124
	20	Gate3[4].Chan[3].AdcAmp[0]	\$9101A0	Gate3[4].Chan[3].AdcAmp[1]	\$9101A4
Gate3[5]	21	Gate3[5].Chan[0].AdcAmp[0]	\$914020	Gate3[5].Chan[0].AdcAmp[1]	\$914024
	22	Gate3[5].Chan[1].AdcAmp[0]	\$9140A0	Gate3[5].Chan[1].AdcAmp[1]	\$9140A4
	23	Gate3[5].Chan[2].AdcAmp[0]	\$914120	Gate3[5].Chan[2].AdcAmp[1]	\$914124
	24	Gate3[5].Chan[3].AdcAmp[0]	\$9141A0	Gate3[5].Chan[3].AdcAmp[1]	\$9141A4
Gate3[6]	25	Gate3[6].Chan[0].AdcAmp[0]	\$918020	Gate3[6].Chan[0].AdcAmp[1]	\$918024
	26	Gate3[6].Chan[1].AdcAmp[0]	\$9180A0	Gate3[6].Chan[1].AdcAmp[1]	\$9180A4
	27	Gate3[6].Chan[2].AdcAmp[0]	\$918120	Gate3[6].Chan[2].AdcAmp[1]	\$918124
	28	Gate3[6].Chan[3].AdcAmp[0]	\$9181A0	Gate3[6].Chan[3].AdcAmp[1]	\$9181A4
Gate3[7]	29	Gate3[7].Chan[0].AdcAmp[0]	\$91C020	Gate3[7].Chan[0].AdcAmp[1]	\$91C024
	30	Gate3[7].Chan[1].AdcAmp[0]	\$91C0A0	Gate3[7].Chan[1].AdcAmp[1]	\$91C0A4
	31	Gate3[7].Chan[2].AdcAmp[0]	\$91C120	Gate3[7].Chan[2].AdcAmp[1]	\$91C124
	32	Gate3[7].Chan[3].AdcAmp[0]	\$91C1A0	Gate3[7].Chan[3].AdcAmp[1]	\$91C1A4



추가 정보

- 달러(\$) 기호는 PMAC 스크립트 언어의 16진수 값을 나타냅니다.
- 명시적 주소는 Gate3[i].Chan[j].AdcAmp[k]에서 Sys.piom을 빼고 결과를 16진수로 읽어서 찾습니다.

5-6 상태 비트의 세부 정보

이 섹션에서는 CK3A 증폭기가 컨트롤러에 보고할 수 있는 세 가지 상태 비트 세트의 세부 정보를 설명합니다.

5-6-1 상태 비트 세트 1

ADC Strobe Word에서 "올바르게" 설정된 경우 **ADC A, Gate3[i].Chan[j].AdcAmp[0]**에서 다음 상태 비트를 찾을 수 있습니다.

비트 #	항목	사양	결함/표시 코드
8	주 전원	=0 결함 없음, =1 결함	A1
9	소프트 시작	=0 결함 없음, =1 결함	A2
10	STO 상태	=0 트리거되지 않음, =1 트리거됨	A6
11	섀트 단락	=0 결함 없음, =1 결함	A3
12	과전압	=0 결함 없음, =1 결함	A5
13	I2T 결함	=0 결함 없음, =1 결함	A8
14	단락	=0 결함 없음, =1 결함	AC
15	과부하	=0 결함 없음, =1 결함	AL

일반적으로 증폭기/모터 #1에 연결된 Gate3 인덱스 0 채널 0을 사용한 직접 매핑 액세스 예:

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1
PTR Ck3a1PwrFlt      ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt ->U.IO:$900020.9.1
PTR Ck3a1StoStatus    ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt   ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt        ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt      ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt   ->U.IO:$900020.15.1
```

5-6-2 상태 비트 세트 2

ADC 스트로브 워드의 비트 [11:08]이 0으로 설정된 경우 증폭기는 **ADC B**, **Gate3[i].Chan[j].AdcAmp[1]**에서 상태 비트 세트 2를 보고합니다.

비트 #	항목	사양	결함/표시 코드
8	전원 입력 상태	=1 AC 전원 인가됨, =0 제거됨	A1
10	과열	=0 결함 없음, =1 결함	A7
11	섀트 과부하	=0 결함 없음, =1 결함	A4
12	PWM 주파수 결함	=0 결함 없음, =1 결함	P1
13	PWM 명령 결함	=0 결함 없음, =1 결함	P2
14	증폭기 활성화 상태	=0 활성화되지 않음, =1 활성화됨	0(활성화된 경우)



추가 정보

다른 비트 값은 내부용으로 예약되어 있습니다.

일반적으로 증폭기/모터 #1에 연결된 Gate3 인덱스 0 채널 0을 사용한 직접 매핑 액세스 예:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $0
```

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[1] STATUS SET 2
PTR Ck3a1PwrStatus      ->U.IO:$900024.8.1
PTR Ck3a1OverTempFlt    ->U.IO:$900024.10.1
PTR Ck3a1ShuntOverLFlt  ->U.IO:$900024.11.1
PTR Ck3a1PwmFreqFlt     ->U.IO:$900024.12.1
PTR Ck3a1PwmCmdFlt      ->U.IO:$900024.13.1
PTR Ck3a1AmpEna         ->U.IO:$900024.14.1
```

5-6-3 상태 비트 세트 3

ADC 스트로브 워드의 비트 [11:08]이 1인 경우 증폭기는 **ADC B, Gate3[i].Chan[j].AdcAmp[1]**에서 상태 비트 세트 2를 보고합니다.

비트 #	항목	사양	결함/표시 코드
10	I2T 적분기 상태	=0 끄기, =1 켜기	-
12	ADC 오프셋	=0 결함 없음, =1 결함	A9
14	증폭기 준비	=0 준비, =1 준비 안 됨	-



추가 정보

- 다른 비트 값은 내부용으로 예약되어 있습니다.
- 증폭기 준비 비트는 증폭기를 초기화하는 동안(약 6초) 1로 설정됩니다. 증폭기가 준비되었음을 나타내는 부팅이 완료되면 0으로 설정됩니다.

일반적으로 증폭기/모터 #1에 연결된 Gate3 인덱스 0 채널 0을 사용한 직접 매핑 액세스 예:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $100
```

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[1] STATUS SET 3
```

```
PTR Ck3a1I2tint ->U.IO:$900024.10.1
```

```
PTR Ck3a1AdcOffsetFlt ->U.IO:$900024.12.1
```

```
PTR Ck3a1Ready ->U.IO:$900024.14.1
```

5-7 증폭기 데이터의 세부 정보

이 섹션에서는 CK3A 증폭기가 컨트롤러에 보고할 수 있는 데이터의 세부 정보를 설명합니다.

5-7-1 개요 및 예

ADC 스트로브 워드의 비트 [11:08]은 컨트롤러에 보고되는 증폭기 데이터도 지정합니다. 보고된 데이터는 ADC B 비트 [15:08]에서 찾을 수 있습니다.

Bit # [11:08] 설정	증폭기 데이터 보고됨
= 2	VDC의 버스 전압(4로 나눈 값)
= 3	증폭기 전력 모듈 온도(°C)
= 6	증폭기 펌웨어 버전
= 7	암페어 단위의 증폭기 전류 정격(CK3A-G305L의 경우 5, CK3A-G310L의 경우 10)

다음은 IDE 터미널 창에서 빠르게 액세스하는 예입니다. 일반적으로 증폭기/모터 #1에 연결된 Gate3 인덱스 0 채널 0을 사용합니다.

전역 정의

```
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
```

DC 버스 전압

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $200
```

```
Ck3a1DcBus = ((Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8) * 4
```

전원 모듈 온도

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $300
```

```
Ck3a1Temp = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

증폭기 펌웨어 버전

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $600
```

```
Ck3a1FwVer = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

증폭기의 전류 등급

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $700
```

```
Ck3a1CurRating = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

5-8 CK3A 샘플 PLC

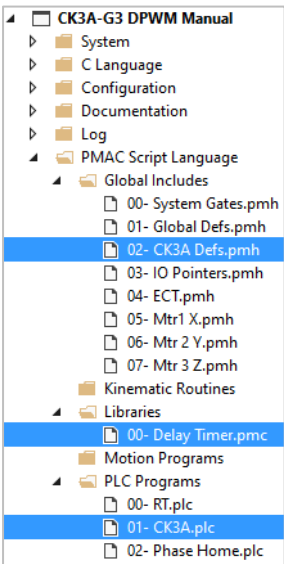
이 섹션에서는 제안된 변수, 포인터 정의 및 4 및 8 모터에 대한 샘플 PLC 프로그램의 Power PMAC 스크립트 목록을 제공하여 사용자가 쉽게 다음을 수행할 수 있도록 합니다.

- 증폭기 기능 제어
- 증폭기 데이터 읽기
- 증폭기 상태 비트 읽기

5-8-1 샘플 PLC 구현

이 섹션에서 샘플 PLC를 성공적으로 구현하려면 다음 단계가 필요합니다.

- 1 지연 타이머 서브루틴이 IDE 프로젝트에 있는지 확인합니다.
- 2 사용된 Sys.Udata[i] 구조가 기존 용도와 충돌하지 않는지 확인합니다.
- 3 IDE 프로젝트의 **Global Includes** 섹션에 있는 파일에 **CK3A 정의**를 복사합니다.
- 4 **CK3A PLC**를 IDE 프로젝트의 **PLC 프로그램**에 있는 파일로 복사합니다.
- 5 전원을 켤 때 CK3A PLC를 활성화합니다(선택 사항).
- 6 컴파일 오류 없이 다운로드가 성공했는지 확인합니다.



추가 정보

IDE 프로젝트 트리의 파일 이름은 작업에 중요하지 않습니다.

5-8-2 지연 타이머 서브루틴

지연 타이머 서브루틴은 Power PMAC 스크립트 PLC 프로그램에서 시간 지연을 도입하는 데 사용됩니다. 이 지연 타이머 서브루틴은 대부분의 IDE 프로젝트 템플릿에 있습니다. 아래는 참조용입니다.

```

OPEN SUBPROG DelayTimer
SUB: sec (DelayTimeSec)
LOCAL EndTimeSec
EndTimeSec = Sys.Time + DelayTimeSec
WHILE (EndTimeSec > Sys.Time){}
RETURN

SUB: msec (DelayTimeMsec)
LOCAL EndTimeMsec
EndTimeMsec = Sys.Time + DelayTimeMsec * 0.001
WHILE (EndTimeMsec > Sys.Time){}
RETURN
CLOSE

```

5-8-3 사용자 공유 메모리에 대한 설명

상태 비트 세트 1은 지정된 채널의 ADC A 레지스터에서 보고됩니다. 이러한 비트는 ADC 스트로브 워드의 특별한 설정 없이 항상 액세스할 수 있습니다.

그러나 상태 비트 세트 2와 3은 ADC B 레지스터의 동일한 위치를 공유합니다. ADC 스트로브 설정의 비트 [11:08]은 현재 보고되는 세트를 나타냅니다.

- 상태 세트 2는 ADC 스트로브 워드의 비트 [11:08]이 0인 경우 보고됩니다.
- 상태 세트 3은 ADC 스트로브 워드의 비트 [11:08]이 1인 경우 보고됩니다.

이러한 이유로 비트 [11:08] 설정이 PLC에서 변경될 때마다(0과 1 사이) 홀딩 레지스터로 "미러 워드"를 사용합니다.

"서명되지 않은" 사용자 공유 메모리 레지스터 Sys.Udata[i]는 이러한 절차 유형에 가장 적합합니다. 이 섹션의 샘플 코드에서 사용되는 레지스터는 다음과 같이 정의됩니다.

상태 세트 2	Sys.Udata[i]	주소	상태 세트 3	Sys.Udata[i]	주소
채널 1	Sys.Udata[256000]	\$FA000	채널 1	Sys.Udata[256040]	\$FA0A0
채널 2	Sys.Udata[256001]	\$FA004	채널 2	Sys.Udata[256041]	\$FA0A4
채널 3	Sys.Udata[256002]	\$FA008	채널 3	Sys.Udata[256042]	\$FA0A8
채널 4	Sys.Udata[256003]	\$FA00C	채널 4	Sys.Udata[256043]	\$FA0AC
채널 5	Sys.Udata[256004]	\$FA010	채널 5	Sys.Udata[256044]	\$FA0B0
채널 6	Sys.Udata[256005]	\$FA014	채널 6	Sys.Udata[256045]	\$FA0B4
채널 7	Sys.Udata[256006]	\$FA018	채널 7	Sys.Udata[256046]	\$FA0B8
채널 8	Sys.Udata[256007]	\$FA01C	채널 8	Sys.Udata[256047]	\$FA0BC

PLC에서 제안하는 방법은 다음과 같습니다.

- ADC 스트로브 워드의 비트 [11:08] = 0으로 설정
- 상태 세트 2를 Sys.Udata[25600x]에 복사
- ADC 스트로브 워드의 비트 [11:08] = 1로 설정
- 상태 세트 3을 Sys.Udata[25604x]에 복사

개별 비트를 Sys.Udata[i] 레지스터에 매핑하면 각 세트의 상태 비트가 반영됩니다.



추가 정보

- 이러한 Sys.Udata[i] 레지스터가 현재 적용 분야에서 사용되지 않는지 확인하십시오.
- 레지스터 인덱스가 변경된 경우 주의해야 합니다.
- "서명되지 않은" 사용자 공유 메모리 레지스터의 명시적 주소는 sys.Udata[i].a에서 sys.pushm을 빼고 그 결과를 16진수로 읽어서 찾을 수 있습니다.

5-8-4 CK3A 샘플 정의 4 모터

```

// GATE3[0], CH. 1 - 4 AMPLIFIER FUNCTIONS ===== //
#define Gate30WpDis Gate3[0].WpKey = $AAAAAAA

GLOBAL Ck3a0ClrF, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl

PTR Gate30DataCtrl      ->U.IO:$90020C.16.4
PTR Ck3a0BusDisState     ->U.IO:$90020C.13.1
PTR Ck3a0ClrFBit        ->U.IO:$90020C.24.1
PTR Ck3a0FanState       ->U.IO:$90020C.25.1
PTR Ck3a0DbState        ->U.IO:$90020C.28.1

// CH. 1 - 4 ADC B REGISTERS [15:08] ===== //
PTR Ck3a1AdcBData       ->U.IO:$900024.8.8 // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData       ->U.IO:$9000A4.8.8 // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData       ->U.IO:$900124.8.8 // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData       ->U.IO:$9001A4.8.8 // Gate3[0].Chan[3].AdcAmp[1]

// CH. 1 - 4 AMPLIFIER DATA ===== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a1PwrFlt         ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt   ->U.IO:$900020.9.1
PTR Ck3a1StoStatus      ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt  ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt    ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt         ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt       ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt    ->U.IO:$900020.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a2PwrFlt         ->U.IO:$9000A0.8.1
PTR Ck3a2SoftStartFlt   ->U.IO:$9000A0.9.1
PTR Ck3a2StoStatus      ->U.IO:$9000A0.10.1
PTR Ck3a2ShuntShortFlt  ->U.IO:$9000A0.11.1
PTR Ck3a2OverVoltFlt    ->U.IO:$9000A0.12.1
PTR Ck3a2I2tFlt         ->U.IO:$9000A0.13.1
PTR Ck3a2ShortFlt       ->U.IO:$9000A0.14.1
PTR Ck3a2OverLoadFlt    ->U.IO:$9000A0.15.1

// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a3PwrFlt         ->U.IO:$900120.8.1
PTR Ck3a3SoftStartFlt   ->U.IO:$900120.9.1
PTR Ck3a3StoStatus      ->U.IO:$900120.10.1
PTR Ck3a3ShuntShortFlt  ->U.IO:$900120.11.1
PTR Ck3a3OverVoltFlt    ->U.IO:$900120.12.1
PTR Ck3a3I2tFlt         ->U.IO:$900120.13.1
PTR Ck3a3ShortFlt       ->U.IO:$900120.14.1
PTR Ck3a3OverLoadFlt    ->U.IO:$900120.15.1

```

```

// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a4PwrFlt      ->U.IO:$9001A0.8.1
PTR Ck3a4SoftStartFlt ->U.IO:$9001A0.9.1
PTR Ck3a4StoStatus    ->U.IO:$9001A0.10.1
PTR Ck3a4ShuntShortFlt ->U.IO:$9001A0.11.1
PTR Ck3a4OverVoltFlt  ->U.IO:$9001A0.12.1
PTR Ck3a4I2tFlt       ->U.IO:$9001A0.13.1
PTR Ck3a4ShortFlt     ->U.IO:$9001A0.14.1
PTR Ck3a4OverLoadFlt  ->U.IO:$9001A0.15.1

// CH. 1, SYS.UDATA[256000] STATUS SET 2 ===== //
PTR Ck3a1PwrStatus    ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt  ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt   ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt    ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna       ->U.USER:$FA000.6.1

// CH. 2, SYS.UDATA[256001] STATUS SET 2 ===== //
PTR Ck3a2PwrStatus    ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt  ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLFlt ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt   ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt    ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna       ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ===== //
PTR Ck3a3PwrStatus    ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt  ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLFlt ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt   ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt    ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna       ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ===== //
PTR Ck3a4PwrStatus    ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt  ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLFlt ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt   ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt    ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna       ->U.USER:$FA00C.6.1

// CH. 1, SYS.UDATA[256040] STATUS SET 3 ===== //
PTR Ck3a1I2tint       ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready        ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ===== //
PTR Ck3a2I2tint       ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready        ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ===== //
PTR Ck3a3I2tint       ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready        ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ===== //
PTR Ck3a4I2tint       ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready        ->U.USER:$FA0AC.6.1

```

5-8-5 CK3A 샘플 PLC 4 모터

```

OPEN PLC Ck3aPLC
// CH. 1 - 4 STATUS BITS SET 2
Gate30WpDis Gate30DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 STATUS BITS SET 3
Gate30WpDis Gate30DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 DC BUS VOLTAGE
Gate30WpDis Gate30DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
CALL DelayTimer.msec(1)

// CH. 1 - 4 TEMPERATURE
Gate30WpDis Gate30DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 FIRMWARE VERSION
Gate30WpDis Gate30DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 CURRENT RATING
Gate30WpDis Gate30DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Gate30WpDis Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Gate30WpDis Ck3a0BusDisState = 0

```

```

// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Gate30WpDis Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Gate30WpDis Ck3a0FanState = 0

// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Gate30WpDis Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Gate30WpDis Ck3a0DbState = 1

// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0ClrF == 1)
{
    Gate30WpDis Ck3a0ClrFBit = 1
    CALL DelayTimer.msec(10)
    Gate30WpDis Ck3a0ClrFBit = 0
    Ck3a0ClrF = 0
}

// CH. 1 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
CLOSE

```

5-8-6 CK3A 샘플 정의 8 모터

```

// GATE3[0], CH. 1 - 8 AMPLIFIER FUNCTIONS ===== //
#define Gate30WpDis Gate3[0].WpKey = $AAAAA
#define Gate31WpDis Gate3[1].WpKey = $AAAAA

GLOBAL Ck3a0ClrF, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl
GLOBAL Ck3a1ClrF, Ck3a1FanCtrl, Ck3a1BusDisCtrl, Ck3a1DbCtrl

PTR Gate30DataCtrl ->U.IO:$90020C.16.4
PTR Ck3a0BusDisState ->U.IO:$90020C.13.1
PTR Ck3a0ClrFBit ->U.IO:$90020C.24.1
PTR Ck3a0FanState ->U.IO:$90020C.25.1
PTR Ck3a0DbState ->U.IO:$90020C.28.1

PTR Gate31DataCtrl ->U.IO:$90420C.16.4
PTR Ck3a1BusDisState ->U.IO:$90420C.13.1
PTR Ck3a1ClrFBit ->U.IO:$90420C.24.1
PTR Ck3a1FanState ->U.IO:$90420C.25.1
PTR Ck3a1DbState ->U.IO:$90420C.28.1

// CH. 1 - 8 ADC B REGISTERS [15:08] ===== //
PTR Ck3a1AdcBData ->U.IO:$900024.8.8 // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData ->U.IO:$9000A4.8.8 // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData ->U.IO:$900124.8.8 // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData ->U.IO:$9001A4.8.8 // Gate3[0].Chan[3].AdcAmp[1]

PTR Ck3a5AdcBData ->U.IO:$904024.8.8 // Gate3[1].Chan[0].AdcAmp[1]
PTR Ck3a6AdcBData ->U.IO:$9040A4.8.8 // Gate3[1].Chan[1].AdcAmp[1]
PTR Ck3a7AdcBData ->U.IO:$904124.8.8 // Gate3[1].Chan[2].AdcAmp[1]
PTR Ck3a8AdcBData ->U.IO:$9041A4.8.8 // Gate3[1].Chan[3].AdcAmp[1]

// CH. 1 - 8 AMPLIFIER DATA ===== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

GLOBAL Ck3a5DcBus, Ck3a5Temp, Ck3a5FwVer, Ck3a5CurRating
GLOBAL Ck3a6DcBus, Ck3a6Temp, Ck3a6FwVer, Ck3a6CurRating
GLOBAL Ck3a7DcBus, Ck3a7Temp, Ck3a7FwVer, Ck3a7CurRating
GLOBAL Ck3a8DcBus, Ck3a8Temp, Ck3a8FwVer, Ck3a8CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a1PwrFlt ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt ->U.IO:$900020.9.1
PTR Ck3a1StoStatus ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt ->U.IO:$900020.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a2PwrFlt ->U.IO:$9000A0.8.1
PTR Ck3a2SoftStartFlt ->U.IO:$9000A0.9.1
PTR Ck3a2StoStatus ->U.IO:$9000A0.10.1
PTR Ck3a2ShuntShortFlt ->U.IO:$9000A0.11.1
PTR Ck3a2OverVoltFlt ->U.IO:$9000A0.12.1
PTR Ck3a2I2tFlt ->U.IO:$9000A0.13.1
PTR Ck3a2ShortFlt ->U.IO:$9000A0.14.1
PTR Ck3a2OverLoadFlt ->U.IO:$9000A0.15.1

```

```

// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a3PwrFlt      ->U.IO:$900120.8.1
PTR Ck3a3SoftStartFlt ->U.IO:$900120.9.1
PTR Ck3a3StoStatus    ->U.IO:$900120.10.1
PTR Ck3a3ShuntShortFlt ->U.IO:$900120.11.1
PTR Ck3a3OverVoltFlt  ->U.IO:$900120.12.1
PTR Ck3a3I2tFlt       ->U.IO:$900120.13.1
PTR Ck3a3ShortFlt     ->U.IO:$900120.14.1
PTR Ck3a3OverLoadFlt  ->U.IO:$900120.15.1

// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a4PwrFlt      ->U.IO:$9001A0.8.1
PTR Ck3a4SoftStartFlt ->U.IO:$9001A0.9.1
PTR Ck3a4StoStatus    ->U.IO:$9001A0.10.1
PTR Ck3a4ShuntShortFlt ->U.IO:$9001A0.11.1
PTR Ck3a4OverVoltFlt  ->U.IO:$9001A0.12.1
PTR Ck3a4I2tFlt       ->U.IO:$9001A0.13.1
PTR Ck3a4ShortFlt     ->U.IO:$9001A0.14.1
PTR Ck3a4OverLoadFlt  ->U.IO:$9001A0.15.1

// CH. 5, Gate3[1].Chan[0].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a5PwrFlt      ->U.IO:$904020.8.1
PTR Ck3a5SoftStartFlt ->U.IO:$904020.9.1
PTR Ck3a5StoStatus    ->U.IO:$904020.10.1
PTR Ck3a5ShuntShortFlt ->U.IO:$904020.11.1
PTR Ck3a5OverVoltFlt  ->U.IO:$904020.12.1
PTR Ck3a5I2tFlt       ->U.IO:$904020.13.1
PTR Ck3a5ShortFlt     ->U.IO:$904020.14.1
PTR Ck3a5OverLoadFlt  ->U.IO:$904020.15.1

// CH. 6, Gate3[1].Chan[1].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a6PwrFlt      ->U.IO:$9040A0.8.1
PTR Ck3a6SoftStartFlt ->U.IO:$9040A0.9.1
PTR Ck3a6StoStatus    ->U.IO:$9040A0.10.1
PTR Ck3a6ShuntShortFlt ->U.IO:$9040A0.11.1
PTR Ck3a6OverVoltFlt  ->U.IO:$9040A0.12.1
PTR Ck3a6I2tFlt       ->U.IO:$9040A0.13.1
PTR Ck3a6ShortFlt     ->U.IO:$9040A0.14.1
PTR Ck3a6OverLoadFlt  ->U.IO:$9040A0.15.1

// CH. 7, Gate3[1].Chan[2].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a7PwrFlt      ->U.IO:$904120.8.1
PTR Ck3a7SoftStartFlt ->U.IO:$904120.9.1
PTR Ck3a7StoStatus    ->U.IO:$904120.10.1
PTR Ck3a7ShuntShortFlt ->U.IO:$904120.11.1
PTR Ck3a7OverVoltFlt  ->U.IO:$904120.12.1
PTR Ck3a7I2tFlt       ->U.IO:$904120.13.1
PTR Ck3a7ShortFlt     ->U.IO:$904120.14.1
PTR Ck3a7OverLoadFlt  ->U.IO:$904120.15.1

// CH. 8, Gate3[1].Chan[3].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a8PwrFlt      ->U.IO:$9041A0.8.1
PTR Ck3a8SoftStartFlt ->U.IO:$9041A0.9.1
PTR Ck3a8StoStatus    ->U.IO:$9041A0.10.1
PTR Ck3a8ShuntShortFlt ->U.IO:$9041A0.11.1
PTR Ck3a8OverVoltFlt  ->U.IO:$9041A0.12.1
PTR Ck3a8I2tFlt       ->U.IO:$9041A0.13.1
PTR Ck3a8ShortFlt     ->U.IO:$9041A0.14.1
PTR Ck3a8OverLoadFlt  ->U.IO:$9041A0.15.1

```

```

// CH. 1, SYS.UDATA[256000] STATUS SET 2 ===== //
PTR Ck3a1PwrStatus      ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt    ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt  ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt     ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt      ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna         ->U.USER:$FA000.6.1

// CH. 2, SYS.UDATA[256001] STATUS SET 2 ===== //
PTR Ck3a2PwrStatus      ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt    ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLFlt  ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt     ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt      ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna         ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ===== //
PTR Ck3a3PwrStatus      ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt    ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLFlt  ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt     ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt      ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna         ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ===== //
PTR Ck3a4PwrStatus      ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt    ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLFlt  ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt     ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt      ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna         ->U.USER:$FA00C.6.1

// CH. 5, SYS.UDATA[256004] STATUS SET 2 ===== //
PTR Ck3a5PwrStatus      ->U.USER:$FA010.0.1
PTR Ck3a5OverTempFlt    ->U.USER:$FA010.2.1
PTR Ck3a5ShuntOverLFlt  ->U.USER:$FA010.3.1
PTR Ck3a5PwmFreqFlt     ->U.USER:$FA010.4.1
PTR Ck3a5PwmCmdFlt      ->U.USER:$FA010.5.1
PTR Ck3a5AmpEna         ->U.USER:$FA010.6.1

// CH. 6, SYS.UDATA[256005] STATUS SET 2 ===== //
PTR Ck3a6PwrStatus      ->U.USER:$FA014.0.1
PTR Ck3a6OverTempFlt    ->U.USER:$FA014.2.1
PTR Ck3a6ShuntOverLFlt  ->U.USER:$FA014.3.1
PTR Ck3a6PwmFreqFlt     ->U.USER:$FA014.4.1
PTR Ck3a6PwmCmdFlt      ->U.USER:$FA014.5.1
PTR Ck3a6AmpEna         ->U.USER:$FA014.6.1

// CH. 7, SYS.UDATA[256006] STATUS SET 2 ===== //
PTR Ck3a7PwrStatus      ->U.USER:$FA018.0.1
PTR Ck3a7OverTempFlt    ->U.USER:$FA018.2.1
PTR Ck3a7ShuntOverLFlt  ->U.USER:$FA018.3.1
PTR Ck3a7PwmFreqFlt     ->U.USER:$FA018.4.1
PTR Ck3a7PwmCmdFlt      ->U.USER:$FA018.5.1
PTR Ck3a7AmpEna         ->U.USER:$FA018.6.1

// CH. 8, SYS.UDATA[256007] STATUS SET 2 ===== //
PTR Ck3a8PwrStatus      ->U.USER:$FA01C.0.1
PTR Ck3a8OverTempFlt    ->U.USER:$FA01C.2.1
PTR Ck3a8ShuntOverLFlt  ->U.USER:$FA01C.3.1
PTR Ck3a8PwmFreqFlt     ->U.USER:$FA01C.4.1
PTR Ck3a8PwmCmdFlt      ->U.USER:$FA01C.5.1
PTR Ck3a8AmpEna         ->U.USER:$FA01C.6.1

```

```

// CH. 1, SYS.UDATA[256040] STATUS SET 3 ===== //
PTR Ck3a1I2tint      ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready       ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ===== //
PTR Ck3a2I2tint      ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready       ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ===== //
PTR Ck3a3I2tint      ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready       ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ===== //
PTR Ck3a4I2tint      ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready       ->U.USER:$FA0AC.6.1

// CH. 5, SYS.UDATA[256044] STATUS SET 3 ===== //
PTR Ck3a5I2tint      ->U.USER:$FA0B0.2.1
PTR Ck3a5AdcOffsetFlt ->U.USER:$FA0B0.4.1
PTR Ck3a5Ready       ->U.USER:$FA0B0.6.1

// CH. 6, SYS.UDATA[256045] STATUS SET 3 ===== //
PTR Ck3a6I2tint      ->U.USER:$FA0B4.2.1
PTR Ck3a6AdcOffsetFlt ->U.USER:$FA0B4.4.1
PTR Ck3a6Ready       ->U.USER:$FA0B4.6.1

// CH. 7, SYS.UDATA[256046] STATUS SET 3 ===== //
PTR Ck3a7I2tint      ->U.USER:$FA0B8.2.1
PTR Ck3a7AdcOffsetFlt ->U.USER:$FA0B8.4.1
PTR Ck3a7Ready       ->U.USER:$FA0B8.6.1

// CH. 8, SYS.UDATA[256047] STATUS SET 3 ===== //
PTR Ck3a8I2tint      ->U.USER:$FA0BC.2.1
PTR Ck3a8AdcOffsetFlt ->U.USER:$FA0BC.4.1
PTR Ck3a8Ready       ->U.USER:$FA0BC.6.1

```

5-8-7 CK3A 샘플 PLC 8 모터

```

OPEN PLC Ck3aPLC
// CH. 1 - 8 STATUS BITS SET 2
Gate30WpDis Gate30DataCtrl = 0
Gate31WpDis Gate31DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
Sys.Udata[256004] = Ck3a5AdcBData
Sys.Udata[256005] = Ck3a6AdcBData
Sys.Udata[256006] = Ck3a7AdcBData
Sys.Udata[256007] = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 STATUS BITS SET 3
Gate30WpDis Gate30DataCtrl = 1
Gate31WpDis Gate31DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
Sys.Udata[256044] = Ck3a5AdcBData
Sys.Udata[256045] = Ck3a6AdcBData
Sys.Udata[256046] = Ck3a7AdcBData
Sys.Udata[256047] = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 DC BUS VOLTAGE
Gate30WpDis Gate30DataCtrl = 2
Gate31WpDis Gate31DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
IF(Ck3a5Ready == 0) Ck3a5DcBus = Ck3a5AdcBData * 4
IF(Ck3a6Ready == 0) Ck3a6DcBus = Ck3a6AdcBData * 4
IF(Ck3a7Ready == 0) Ck3a7DcBus = Ck3a7AdcBData * 4
IF(Ck3a8Ready == 0) Ck3a8DcBus = Ck3a8AdcBData * 4
CALL DelayTimer.msec(1)

// CH. 1 - 8 TEMPERATURE
Gate30WpDis Gate30DataCtrl = 3
Gate31WpDis Gate31DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5Temp = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6Temp = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7Temp = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8Temp = Ck3a8AdcBData
CALL DelayTimer.msec(1)

```

```

// CH. 1 - 8 FIRMWARE VERSION
Gate30WpDis Gate30DataCtrl = 6
Gate31WpDis Gate31DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5FwVer = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6FwVer = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7FwVer = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8FwVer = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 CURRENT RATING
Gate30WpDis Gate30DataCtrl = 7
Gate31WpDis Gate31DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5CurRating = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6CurRating = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7CurRating = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8CurRating = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Gate30WpDis Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Gate30WpDis Ck3a0BusDisState = 0
// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Gate30WpDis Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Gate30WpDis Ck3a0FanState = 0
// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Gate30WpDis Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Gate30WpDis Ck3a0DbState = 1

// GATE3[1] CH. 5-8 BUS DISCHARGE CONTROL
IF(Ck3a1BusDisCtrl == 1 && Ck3a1BusDisState == 0) Gate31WpDis Ck3a1BusDisState = 1
IF(Ck3a1BusDisCtrl == 0 && Ck3a1BusDisState == 1) Gate31WpDis Ck3a1BusDisState = 0
// GATE3[1] CH. 5-8 FAN CTRL
IF(Ck3a1FanCtrl == 1 && Ck3a1FanState == 0) Gate31WpDis Ck3a1FanState = 1
IF(Ck3a1FanCtrl == 0 && Ck3a1FanState == 1) Gate31WpDis Ck3a1FanState = 0
// GATE3[1] CH5-8 DYNAMIC BRAKE CONTROL
IF(Ck3a1DbCtrl == 1 && Ck3a1DbState == 1) Gate31WpDis Ck3a1DbState = 0
IF(Ck3a1DbCtrl == 0 && Ck3a1DbState == 0) Gate31WpDis Ck3a1DbState = 1

// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0ClrF == 1)
{
    Gate30WpDis Ck3a0ClrFBit = 1
    CALL DelayTimer.msec(10)
    Gate30WpDis Ck3a0ClrFBit = 0
    Ck3a0ClrF = 0
}

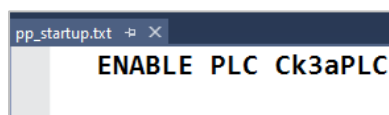
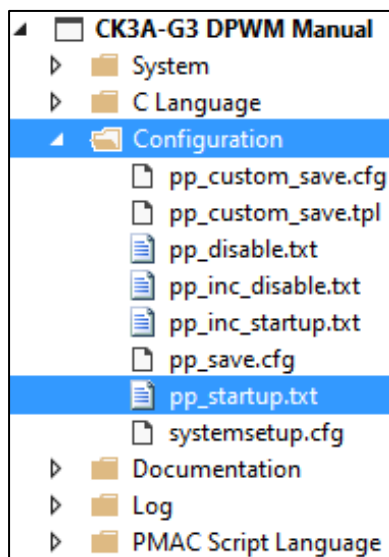
```

```
// GATE3[1] CH. 5-8 CLEAR FAULT(S)
IF(Ck3a1ClrF == 1)
{
    Gate31WpDis Ck3a1ClrFBit = 1
    CALL DelayTimer.msec(10)
    Gate31WpDis Ck3a1ClrFBit = 0
    Ck3a1ClrF = 0
}

// CH. 1 - 8 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
IF(Ck3a5Ready == 1) Ck3a5DcBus,4 = 0
IF(Ck3a6Ready == 1) Ck3a6DcBus,4 = 0
IF(Ck3a7Ready == 1) Ck3a7DcBus,4 = 0
IF(Ck3a8Ready == 1) Ck3a8DcBus,4 = 0
CLOSE
```

5-8-8 전원을 켤 때 CK3A PLC 활성화

IDE 프로젝트의 구성 아래에 있는 startup.txt 파일을 사용하여 전원을 켤 때 CK3A PLC를 활성화할 수 있습니다.



추가 정보

ENABLE PLC 명령은 파일이 아닌 PLC의 이름을 의미합니다.

5-8-9 CK3A PLC 사용

기능 제어

제안된 증폭기 기능 구조는 **Ck3a[i]{function name}**과 같습니다. 여기서,

- {i}는 Gate3 인덱스입니다.
- {function name}은 원하는 기능입니다.

다음은 사용 가능한 명령의 요약입니다.



올바른 사용을 위한 주의사항

기능 변경으로 인해 원하는 ADC 스트로브 워드를 업데이트하고 기본 프로젝트 파일이나 구성에 저장하여 후속 전원 사이클에 적용해야 합니다.

채널	오류 삭제	다이내믹 브레이크 제어	팬 제어	버스 방전 제어
Gate3[0] 채널 1 ~ 4	Ck3a0Clr=1	켜짐의 경우 Ck3a0DbCtrl=1 꺼짐의 경우 Ck3a0DbCtrl=0	Ck3a0FanCtrl=1 켜짐 Ck3a0FanCtrl=0 자동	Ck3a0BusDisCtrl=1 켜짐 Ck3a0BusDisCtrl=0 꺼짐
Gate3[1] 채널 5 ~ 8	Ck3a1Clr=1	켜짐의 경우 Ck3a1DbCtrl=1 꺼짐의 경우 Ck3a1DbCtrl=0	Ck3a1FanCtrl=1 켜짐 Ck3a1FanCtrl=0 자동	Ck3a1BusDisCtrl=1 켜짐 Ck3a1BusDisCtrl=0 꺼짐



추가 정보

Ck3a[i]Clr 매개변수는 PLC에 의해 자동으로 0으로 다시 설정됩니다.



올바른 사용을 위한 주의사항

증폭기가 활성화된 경우 다이내믹 브레이크 설정 켜기/끄기를 전환하지 마십시오.

데이터 읽기

제안된 증폭기 데이터 구조는 **Ck3a{x}{data name}**과 같습니다. 여기서,

- {x}는 모터/증폭기 번호입니다.
- {data name}은 읽을 데이터입니다.

이러한 변수는 모니터링을 위해 IDE 감시 창에 배치하거나 PLC 프로그램 또는 운영자 인터페이스 디스플레이에 사용할 수 있습니다. 다음은 읽기 예입니다.

채널 1

Watch	
Command/Query ^v	Response
Ck3a1DcBus	294
Ck3a1Temp	25
Ck3a1FwVer	60
Ck3a1CurRating	5

Watch	
Command/Query ^v	Response
Ck3a1DcBus,4	294 23 60 5

채널 1 ~ 4

Watch	
Command/Query ^v	Response
Ck3a1DcBus,4,4	294 298 300 294
Ck3a1Temp,4,4	23 28 25 24
Ck3a1FwVer,4,4	60 60 60 60
Ck3a1CurRating,4,4	5 5 5 10

Watch	
Command/Query ^v	Response
Ck3a1DcBus,4	294 23 60 5
Ck3a2DcBus,4	298 28 60 5
Ck3a3DcBus,4	300 25 60 5
Ck3a4DcBus,4	294 24 60 10



추가 정보

- {variable},{n} 구문은 다음 n개의 연속된 변수를 보고합니다.
- {variable},{n},{m} 구문은 다음 n개의 변수를 m개의 변수로 구분하여 보고합니다.

상태 비트 읽기

제안된 증폭기 상태 구조는 **Ck3a {x} {status name}**과 같습니다. 여기서,

- {x}는 모터/증폭기 번호입니다.
- {Status name}은 읽을 상태입니다.

이러한 변수는 모니터링을 위해 IDE 감시 창에 배치하거나 PLC 프로그램 또는 운영자 인터페이스 디스플레이에 사용할 수 있습니다. 다음은 읽기 예입니다.

채널 1

Command/Query ^v	Response
Ck3a1PwrFlt	0
Ck3a1SoftStartFlt	0
Ck3a1StoStatus	0
Ck3a1ShuntShortFlt	0
Ck3a1OverVoltFlt	0
Ck3a1I2tFlt	0
Ck3a1ShortFlt	0
Ck3a1OverLoadFlt	0
Ck3a1PwrStatus	1
Ck3a1OverTempFlt	0
Ck3a1ShuntOverLFlt	0
Ck3a1PwmFreqFlt	0
Ck3a1PwmCmdFlt	0
Ck3a1AmpEna	1
Ck3a1I2tint	0
Ck3a1AdcOffsetFlt	0
Ck3a1Ready	0

채널 1 ~ 4

Command/Query ^v	Response
Ck3a1PwrFlt,8	0 0 0 0 0 0 0 0
Ck3a1PwrStatus,6	1 0 0 0 0 1
Ck3a1I2tint,3	0 0 0
Ck3a2PwrFlt,8	0 0 0 0 0 0 0 0
Ck3a2PwrStatus,6	1 0 0 0 0 1
Ck3a2I2tint,3	0 0 0
Ck3a3PwrFlt,8	0 0 0 0 0 0 0 0
Ck3a3PwrStatus,6	1 0 0 0 0 1
Ck3a3I2tint,3	0 0 0
Ck3a4PwrFlt,8	0 0 0 0 0 0 0 0
Ck3a4PwrStatus,6	1 0 0 0 0 1
Ck3a4I2tint,3	0 0 0

Command/Query ^v	Response
Ck3a1PwrFlt,8	0 0 0 0 0 0 0 0
Ck3a1PwrStatus,6	1 0 0 0 0 1
Ck3a1I2tint,3	0 0 0



추가 정보

{variable},{n} 구문은 다음 n개의 연속된 변수를 보고합니다.



문제 해결

이 섹션에서는 CK3A 다이렉트 PWM 증폭기의 LED 표시, 오류 표시 코드 및 문제 해결 방법에 대해 설명합니다.

6-1	LED 표시등	6-2
6-2	표시 코드	6-3
6-2-1	일반 모드 작동	6-3
6-2-2	오류 코드	6-3
6-2-3	문제 해결 오류 코드	6-5
6-2-4	결함 임계값	6-8
6-2-5	증폭기 작동 문제 해결	6-9

6-1 LED 표시등

다음 LED 표시등은 증폭기 전면에 있습니다. 아래는 색상 지정 및 설명입니다.

이름	표시등	상태	색상	설명
로직 전원		켜짐	녹색	24V 로직 전원 적용 및 범위 내 (22 ~ 26.4VDC 사이여야 함)
		켜지지 않음	-	24V 로직 전원이 인가되지 않았거나 범위 내에 있지 않음 (증폭기 꺼짐)
증폭기 활성화됨		켜짐	녹색	증폭기가 활성화됨 (예: OUT 또는 JOG PMAC 명령, 서보 켜짐)
		켜지지 않음	-	증폭기가 활성화되지 않음 (PMAC 명령 정지, 서보 꺼짐 또는 증폭기 꺼짐)
STO		켜짐	녹색	전원 드라이버에 전원이 공급됨, 정상 작동 (모터를 서보할 수 있음)
		켜짐	빨간색	전원 드라이버에 전원이 공급되지 않음, STO 트리거됨 또는 비활성화되지 않음 (모터를 서보할 수 없음)
		켜지지 않음	-	증폭기가 꺼져 있음, 또는 내부 고장 (증폭기가 켜져 있으면 정상 아님)
션트		켜짐	녹색	션트 작동이 활성화 상태임(추적 에너지 또는 제동) (일반적으로 고부하 감속 시 매우 짧은 시간 동안 깜박임)
		켜지지 않음	-	션트 작동이 활성화 상태가 아님 (정상 작동, 문제없음)
버스		켜짐	빨간색	버스 링크 충전됨 (주 회로 전원 인가됨, 콘덴서 충전됨)
		희미함	빨간색	버스 링크가 에너지를 소모하고 있음 (주 회로 전원 제거됨)
		매우 희미함	-	방전 완료, 로직 전원이 켜짐(20V까지 항상 잔류) 또는 증폭기가 5분 미만 동안 꺼짐
		켜지지 않음	-	증폭기가 5분 이상 동안 꺼져 있음



추가 정보

- BUS LED의 강도는 DC 링크의 전압 수준에 정비례합니다.
- 로직 전원 입력이 22.0VDC 미만이거나 26.4VDC보다 크면 증폭기 로직 전원이 꺼지고 STO LED가 켜집니다(희미/빨간색).

6-2 표시 코드

이 섹션에서는 듀얼 7 세그먼트 디스플레이의 오류 코드에 대해 설명합니다.

6-2-1 일반 모드 작동

일반 모드 작동에서 증폭기는 다음 코드를 표시합니다.

표시	결함	설명
	결함 없음	증폭기 꺼짐
	결함 없음	증폭기가 활성화되지 않음 (모터 정지함, 서보 꺼짐)
	결함 없음	증폭기 활성화됨 (PMAC 명령 OUT 또는 JOG와 같은 폐쇄 루프, 서보 켜짐)
	결함 없음	저전압 모드 설정됨(스트로브 워드 설정 및 CN1에 설치된 LV 단락 와이어) 증폭기가 활성화되지 않음(모터 정지함, 서보 꺼짐)
	경고	주 전원 공급이 AC이고 전압이 80 ~ 100VDC로 떨어지는 경우(증폭기는 비활성화되지 않음)
	결함 없음	저전압 모드 설정됨(스트로브 워드 설정 및 CN1에 설치된 LV 단락 와이어) 증폭기 활성화됨(PMAC 명령 OUT 또는 JOG와 같은 폐쇄 루프, 서보 켜짐)

6-2-2 오류 코드

결함이 발생하면 증폭기에 결함 코드가 표시되고 전원 드라이버 회로가 꺼집니다.

표시	결함	설명	리셋
	PWM 주파수	감지된 PWM 주파수가 허용 범위를 벗어났습니다. ([8 ~ 20] KHz)	삭제 결함
	PWM 명령	PWM 명령이 너무 오랫동안 켜짐 상태로 유지되어 IPM 트랜지스터가 지속적으로 전도되었습니다.	삭제 결함
	ADC 클럭	감지된 ADC 클럭이 허용 범위를 벗어났습니다. ([2.450 ~ 6.250] MHz)	자동 리셋

표시	결함	설명	리셋
	주 전원 회로	주 회로 전원 입력(AC 또는 DC)이 없거나 제거되었습니다.	자동 리셋
	소프트 스타트 과부하	중간에 충분한 시간 없이 주 회로 전원 사이클이 너무 많음	로직 전원 사이클
	션트 단락	션트 회로에서 단락이 감지됨	삭제 결함
	션트 과부하	션트 또는 제동의 최대 연속 시간(2초) 초과	삭제 결함
	과전압	내부 버스 전압이 임계값을 초과함(403VDC)	삭제 결함
	STO 입력	STO 비활성화 단락 와이어가 설치되지 않았거나 STO 입력이 제거됨	활성화/삭제 결함
	과열	전원 모듈 온도가 임계값을 초과함(80°C)	삭제 결함
	I2T	200% 공칭 출력에서 허용된 시간이 임계값을 초과함(2초)	삭제 결함
	ADC 오프셋	허용된 값보다 큰 현재 ADC 오프셋(증폭기 펌웨어)	로직 전원 사이클
	단락	최대 전원 모듈 출력에서 허용되는 시간이 임계값을 초과함(3마이크로초)	삭제 결함
	과부하	증폭기의 지정된 최대 출력에서 허용된 시간이 임계값을 초과함(6밀리초)	삭제 결함



올바른 사용을 위한 주의사항

- 소프트 스타트 A2 결함 후 5분간 냉각시킵니다.
- 디스플레이 결함을 삭제하기 위해 소프트 스타트(A2), ADC 오프셋(A9) 및 단락(AC) 결함에는 로직 전원 사이클이 필요합니다.
- 디스플레이 결함은 결함 조건이 제거된 경우에만 삭제할 수 있습니다.



추가 정보

- 일부 디스플레이 결함은 결함 조건이 제거되는 즉시 자동으로 리셋됩니다.
- 일부 결함은 증폭기를 활성화하거나 ADC 스트로브 워드를 사용하여 결함을 제거하여 삭제할 수 있습니다.

6-2-3 문제 해결 오류 코드

표시	오류 및 문제 해결
	증폭기 꺼짐 로직 전원 공급 장치 및 배선을 확인하고 24VDC 입력을 측정하고 22.0 ~ 26.4VDC 사이인지 확인합니다.
	부족 전압 경고(AC 주 전원 공급 장치가 입력된 경우) - 저전압 모드인 경우 이는 정상적인 표시입니다. - 주 전원 공급이 AC이고 전압이 80 ~ 100VDC로 떨어지는 경우, 이를 경고합니다(증폭기는 비활성화되지 않음). - 주 회로(AC) 전원 공급 및 배선을 확인하십시오.
	주 전원 공급 결함 - 주 회로(AC 또는 DC) 전원 공급 및 배선을 확인하십시오. - 저전압 모드가 설정되면 CN1에 저전압 단락선이 설치되어 있는지 확인하십시오. - 저전압 모드가 설정된 경우 ADC 스트로브 워드 비트 #21(LV 모드)이 1로 설정되어 있는지 확인하십시오.
	소프트 스타트 과부하 - 머신 작동 시 주 회로 전원 순환 횟수 감소 - 방전 모드가 꺼져 있으면 주 전원 사이클 사이에 최소 10초의 간격을 두어야 합니다. - 방전 모드가 켜져 있으면 주 전원 사이클 사이에 최소 1분의 간격을 두어야 합니다. - 결함 후 5분간 냉각시키십시오.
	선트 단락 - 내부 선트를 사용하는 경우 B2-B3 단락(CN1)이 적절한 와이어 게이지이고 올바르게 설치되었는지 확인하십시오. - 외부 선트를 사용하는 경우 CN1 측 배선 연결을 점검하십시오. - 외부 선트 저항을 측정하여 무한이 아닌지 확인하십시오. - 외부 선트 와이어가 패널의 다른 와이어와 단락되지 않았는지 확인하십시오.
	선트 과부하 - 과전압 A5 결함이 발생하면 A4도 표시됩니다(이는 정상입니다). - 외부 저항을 사용하는 경우 사양이 권장 선트 저항을 준수하는지 확인하십시오. - 권장 선트 저항을 사용하십시오. - 선트 저항을 측정하여 무한이 아닌지 확인하십시오. - 모터 감속 프로필을 줄여 보십시오.
	과전압 - 내부 선트를 사용하는 경우 B2-B3 단락(CN1)이 적절한 와이어 게이지이고 올바르게 설치되었는지 확인하십시오. - 외부 분로를 사용하는 경우 권장 저항을 사용하고 CN1에 대한 연결을 확인하십시오. - 선트 저항을 측정하고 예상/사양보다 낮지 않은지 확인하십시오. - IDE에서 보고된 전압을 읽고 예상 범위 내에 있는지 확인하십시오. - 모터 감속 프로필을 줄여 보십시오.

표시	오류 및 문제 해결
	STO 입력 - STO 오류 표시 A6은 ADC 스트로브 워드 또는 증폭기 활성화(서보 켜짐)되어 삭제될 때까지 유지됩니다. - STO 입력을 사용하지 않는 경우 CN4.1과 CN4.2 사이의 STO 비활성화 단락이 장착되어 있는지 확인하십시오. - STO 입력을 사용하는 경우 CN4.2가 개방(부동)있는지 확인하십시오. - STO 입력을 사용하는 경우 배선을 확인하고 CN4.1에서 24VDC를 측정하십시오. - CN4.1에 24VDC가 없는 경우 로직 전원이 켜져 있는지 확인하십시오. - STO 입력을 사용하는 경우 STO1 및 STO2 입력에서 24VDC를 측정하십시오. - CN4 커넥터가 연결/연결 해제될 때 정지 LED 변화를 점검하십시오.
	과열 - 작동 환경이 이 설명서에 설명된 사양 내에 있는지 확인하십시오. - 캐비닛 및 장착 간격이 이 설명서에 설명된 사양 내에 있는지 확인하십시오. - 팬이 먼지나 이물질로 막히지 않았는지 확인하십시오. - 팬이 작동하는지 육안으로 확인하고 바람이 나오는지 확인하십시오. - IDE에서 팬 켜기/끄기 전환
	I2T 통합 전류 결함 - 연결이 느슨하거나 간헐적인 단락이 있는지 모터 배선을 확인하십시오. - 모터가 딱딱한 표면에 지속적으로 밀리지 않는지 확인하십시오. - 모터에 브레이크가 있는 경우 작동 중에 브레이크가 해제되었는지 확인하십시오. - CK3A-G305L을 사용하는 경우 더 많은 전력을 위해 CK3A-G310L을 사용하는 것을 고려하십시오. - 모터 가속/감속 프로필을 줄여 보십시오.
	ADC 오프셋 - ADC 스트로브 워드를 사용하여 결함을 삭제하십시오. - 로직 전원을 순환시키십시오.
	단락 - 증폭기가 활성화된 경우(서보 켜짐) 다이내믹 브레이크 켜기/끄기 설정을 전환하지 마십시오. - 모터 배선이 올바르게 적절한 와이어 게이지를 사용하는지 확인하십시오. - 모터 리드가 단락되거나 접지에 연결되어 있지 않은지 점검하십시오. - 모터 위상 저항을 측정하고 일관성이 있는지 확인하십시오. - 모터 가속/감속 프로필을 줄여 보십시오.
	과부하 - 다이내믹 브레이크 기능이 켜진 경우 JOG/를 실행하여 모터를 정지시키기 전에 모터를 정지시키지(서보 꺼짐) 마십시오. - 다이내믹 브레이크 기능이 켜진 경우 증폭기를 활성화한 후 최소 50밀리초 이상 기다린 후 모션을 명령하십시오. - 다이내믹 브레이크 기능이 켜진 경우 모터를 정지한(서보 꺼짐) 후 최소 500밀리초 이상 기다린 후 모터를 다시 서보하십시오. - 모터 배선이 적절한 와이어 게이지를 사용하고 올바르게 배선되었는지 확인하십시오. - 모터 새시가 접지되어 있고 전위 전압이 없는지 확인하십시오. - 모터 위상 저항을 측정하고 일관성이 있는지 확인하십시오. - 주 회로 연결이 적절한 와이어 게이지를 사용하고 올바르게 배선되어 있는지 확인하십시오.

표시	오류 및 문제 해결
	PWM 주파수가 범위를 벗어남 - 컨트롤러의 PWM 주파수 설정이 [8 ~ 20] KHz 범위 내에 있는지 확인하십시오. - IDE 소프트웨어에서 증폭기가 보고한 PWM 주파수를 읽으십시오. - PWM 연결을 확인하고 커넥터가 단단히 연결되어 있는지 확인하십시오. - PWM 케이블을 다른 서보 채널에 연결해 보십시오(불량 컨트롤러 카드인지 확인하기 위해). - PWM 케이블을 교체해 보십시오. - 연결이 느슨하거나 간헐적인 단락이 있는지 모터 배선을 확인하십시오.
	PWM 명령(포화도)이 너무 오래 켜짐 - 컨트롤러의 PWM 주파수 설정이 [8 ~ 20] KHz 범위 내에 있는지 확인하십시오. - Motor[x].PwmSf가 올바르게 설정되었는지 확인하십시오. - 모터 가속 프로필을 줄여 보십시오.



ADC 클럭이 범위를 벗어남

- 컨트롤러의 ADC 클럭 주파수 설정이 [2.450 ~ 6.250] MHz 범위 내에 있는지 확인하십시오.
- PWM 케이블 및 (느슨한) 연결을 확인하십시오.
- PWM 케이블을 다른 서보 채널에 연결해 보십시오(불량 컨트롤러 카드인지 확인하기 위해).
- PWM 케이블을 교체해 보십시오.



추가 정보

위에서 설명한 모든 문제 해결 단계를 수행한 후에도 결함 상태가 지속되고 다시 발생하면 증폭기를 교체하십시오.



안전한 사용을 위한 주의사항

CK3A 다이렉트 PWM 증폭기는 단상 또는 3상 AC 주 전원과 혼용하여 작동하도록 설계되어 있어 3상 작동에서 위상이 손실되더라도 오류가 발생하지 않습니다.

6-2-4 결함 임계값

항목		오류	임계값
과전압(A5)		A5	403VDC
선트 켜짐		-	390VDC
선트 꺼짐		-	370VDC
부족 전압 경고		L-	AC 주 전원 공급 및 전압이 80 ~ 100VDC 사이인 경우
부족 전압(A1) – 일반 AC 모드		A1	80VDC
부족 전압(A1) – 저전압 모드		A1	43VDC
과열 트립(A7)		A7	80°C
과열 릴리스		-	70°C
자동 팬 켜짐		-	35°C
자동 팬 꺼짐		-	32°C
과부하(AL)	CK3A-G305L	AL	15.55A _{peak} 에서 6msec 이상
	CK3A-G310L		28.85A _{peak} 에서 6msec 이상
I2T 통합 시작	CK3A-G305L	-	5.1A _{rms}
	CK3A-G310L		10.1A _{rms}
I2T 트립 시간	CK3A-G305L	A8	5A _{rms} 에서 2초 이상
	CK3A-G310L		10A _{rms} 에서 2초 이상

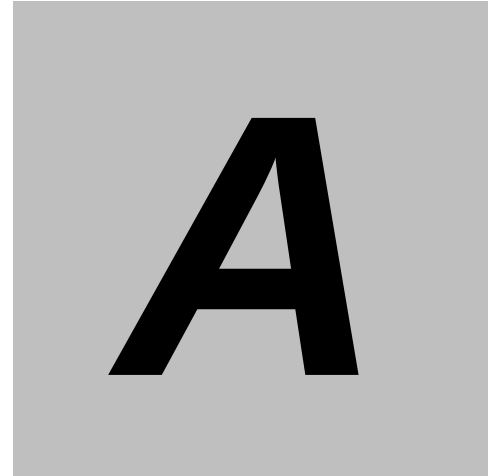
6-2-5 증폭기 작동 문제 해결

문제	가능한 근본 원인	점검 및 조치
7 세그먼트 디스플레이 상의 오류	• 결함 조건 발생 • 증폭기 손상	• 문제 해결 결함 코드 섹션을 참조하여 결함 조건을 제거하고 디스플레이 결함을 재설정하십시오.
STO LED가 켜지지 않음	• 증폭기 손상	
BUS LED가 켜지지 않음	• 주 회로 공급 장치가 꺼짐 • 증폭기 손상 / LED 소손	• A1 오류 코드 문제 해결을 참조하십시오.
증폭기가 켜지지 않습니다(7 세그먼트 디스플레이 표시 없음, PWR LED가 켜지지 않음)	• 로직 전원이 공급되지 않음 • 범위를 벗어난 로직 전원 • 와이어 게이지가 너무 얇음 • 배선 • 증폭기 손상	• 전원 공급 장치가 사양 내에 있습니다. • 전원 공급 장치가 작동 중입니다. • 배선은 올바른 와이어 게이지를 사용합니다. • 배선이 정확합니다.
ENA LED가 켜지지 않음	• 활성화되지 않음(서보 꺼짐) • 컨트롤러의 결함 • 불량 PWM 케이블 • 결함 조건 발생 • 증폭기 손상 / LED 소손	• 증폭기 활성화(서보 켜짐) • PMAC에서 Motor[x].AmpEna는 1이어야 합니다. • PWM 케이블을 교체하십시오. • 문제 해결 결함 코드 섹션을 참조하여 결함 조건을 제거하고 디스플레이 결함을 재설정하십시오.
방전 모드에서 버스가 방출하지 않음	• 내부 셉트 저항 단락 와이어가 설치되어 있지 않음 • 외부 셉트 저항이 설치되어 있지 않음 • 사용된 외부 셉트 저항이 사양을 벗어남 • 방전 켜짐 모드에 대한 ADC 스트로브 워드가 설정되지 않았습니다 • 주 전원 제거 후 로직 전원이 너무 빨리 제거됨	• 내부 셉트용 CN1 단락 와이어가 적절한 게이지를 사용하고 올바르게 배선되었는지 확인하십시오. • 외부 셉트 저항이 올바르게 장착되고 배선되었는지 확인하십시오. • 권장 Omron 외부 셉트 또는 올바른 사양의 셉트를 사용하십시오. • 방전 켜짐에 대한 ADC 스트로브 워드를 올바르게 설정하고 ADC 레지스터에서 상태 비트를 확인하십시오 • 주 전원을 제거한 후 로직 전원을 제거하기 전에 최소 5초 이상 기다리십시오.



추가 정보

증폭기가 켜져 있으면 STO LED가 항상 빨간색 또는 녹색으로 켜져야 합니다.



부록

이 섹션에는 CK3A 다이렉트 PWM 증폭기와 관련된 부록이 포함되어 있습니다.

A-1	IDE를 사용한 자동 모터 설정	A-2
A-1-1	IDE 시스템 설정	A-2
A-2	Power PMAC3 수동 모터 설정	A-11
A-2-1	IDE 프로젝트 생성	A-11
A-2-2	기본 최적화, 시스템 및 게이트 설정	A-15
A-2-3	인코더 진행 위치, 확인	A-16
A-2-4	모터 설정 및 시운전	A-21
A-2-5	모터 작동 방향 반전	A-45
A-3	Gate3 "스크립트" 모터 설정 샘플	A-46
A-3-1	직각위상 인코더가 있는 로터리 서보 모터	A-47
A-3-2	OMRON G5 서보 모터	A-49
A-3-3	OMRON 1S 서보 모터(ACC-84E)	A-52
A-3-4	정현파 인코더가 있는 리니어 서보 모터	A-54
A-3-5	BiSS 인코더가 포함된 리니어 서보 모터(ACC-84E)	A-56
A-4	Gate1 "스크립트" 모터 설정 샘플	A-58
A-4-1	Gate1이 있는 Power PMAC(ACC-24E2)	A-59
A-4-2	Gate1이 있는 Turbo PMAC(ACC-24E2)	A-61
A-5	Gate1용 CK3A 샘플 PLC	A-63
A-5-1	Gate1이 있는 Power PMAC	A-63
A-5-2	Gate1이 있는 Turbo PMAC	A-66

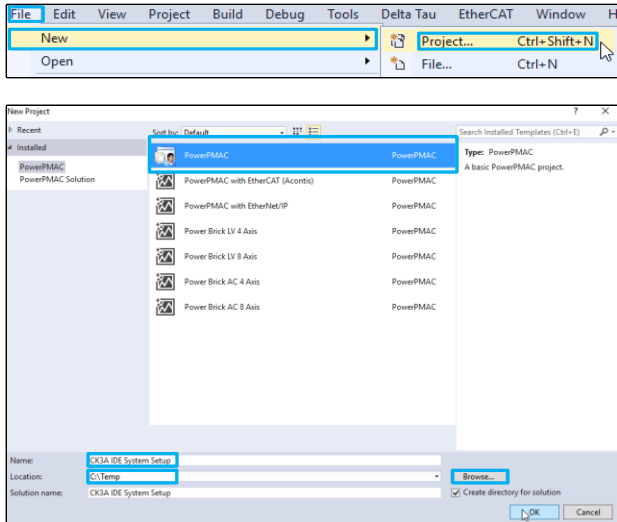
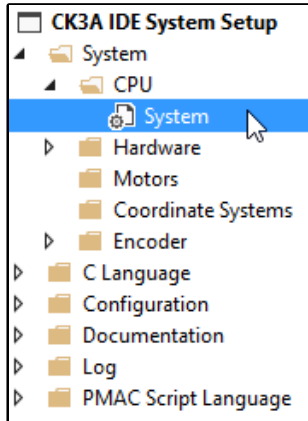
A-1 IDE를 사용한 자동 모터 설정

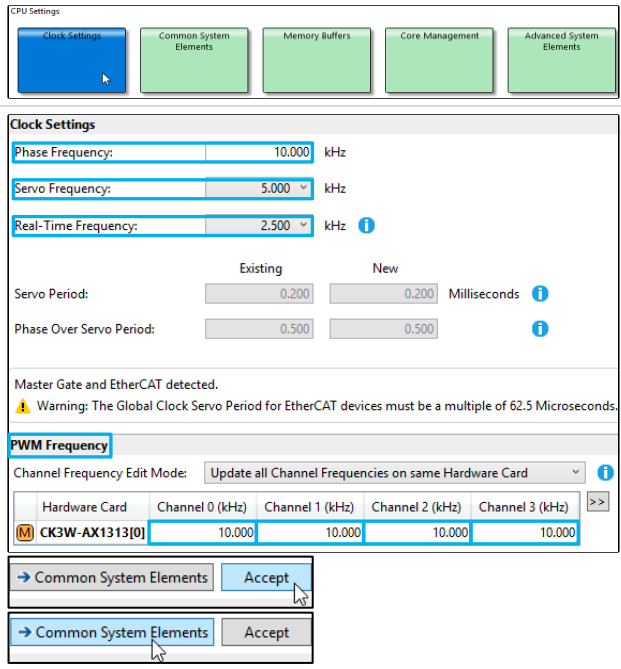
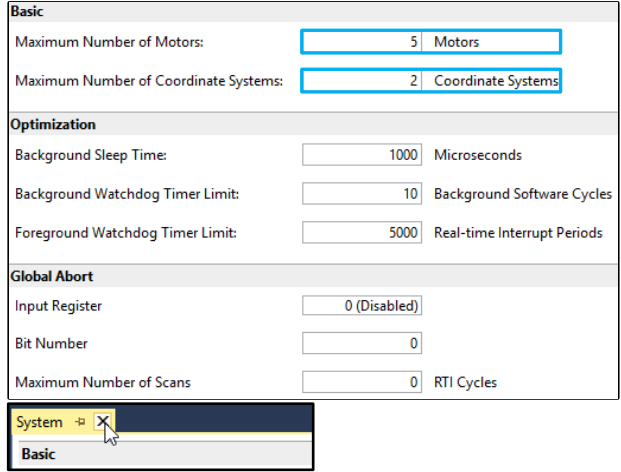
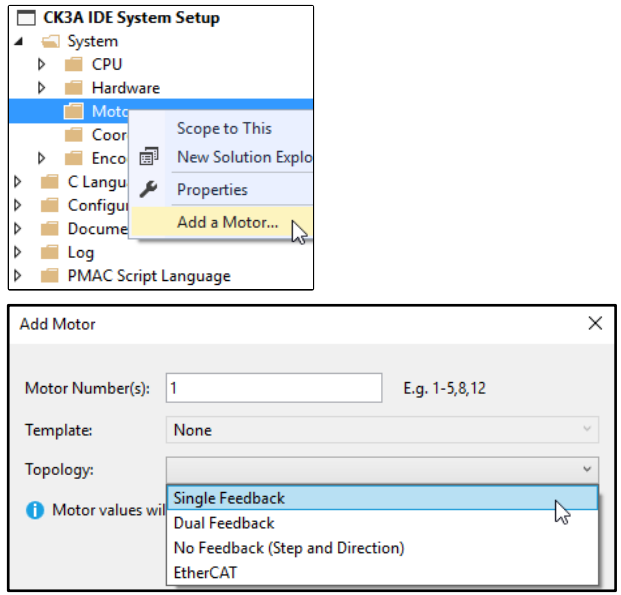
이 섹션에서는 IDE 시스템 설정 도구를 사용하여 CK3A 다이렉트 PWM 증폭기로 브러시리스 서보 모터를 설정하는 단계를 설명합니다.

구성

항목	참고
컨트롤러	이 예는 CK3M AX 또는 UMAC ACC-24E3에 적용됩니다.
증폭기	CK3A-G305L(110VAC 공급 전압)
모터	로터리 서보 모터, 200VAC, 2.4/10.2A, 5극 쌍
인코더	20비트 단일 회전 절대값(Panasonic 프로토콜)

A-1-1 IDE 시스템 설정

단계	그림
새 IDE 프로젝트 만들기 - File(파일) >>> New(새로 만들기) >>> Project(프로젝트)... - 템플릿 선택 - 프로젝트 이름 지정(유형) - PC 폴더 위치 선택(찾아보기)	
이전 단계에서는 IDE 솔루션 탐색기 창에 프로젝트 트리를 만듭니다. "System"을 두 번 클릭	

단계	그림
CPU(시스템 및 전역) 설정 "시계 설정" 클릭	
클럭 설정 - 위상 주파수 설정 - 서보 주파수 설정 - 실시간 주파수 설정 - PWM 주파수 설정 CK3A 채널의 경우 [8~20KHz]여야 함 - 완료되면 "Accept(수락)" - 다음 단계로 이동하려면 "Common System Elements(공통 시스템 요소)"	
공통 시스템 요소 - 최대 모터 수에 + 1 설정 (모터 #0 포함) - 최대 좌표계 수에 + 1 설정 (좌표계 0 포함) - 완료되면 "Accept(수락)" - 시스템 창 닫기, 지금은 완료됨	
모터 추가 - 프로젝트 트리에서 모터 폴더를 마우스 오른쪽 단추로 클릭 "Add a Motor(모터 추가)"를 클릭 - 모터 수 선택 - 토폴로지(이 예에서는 단일 피드백) 선택	
단계	그림

증폭기 지정

- 증폭기 세부 정보(데이터베이스 아이콘)를 클릭하거나 기본 또는 이전에 데이터베이스에서 저장한 증폭기(폴다운 메뉴)를 선택

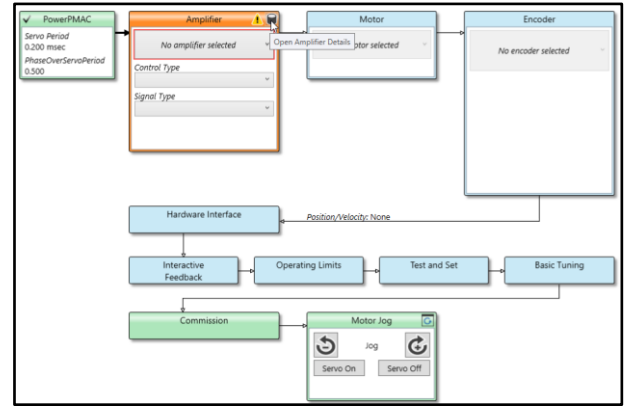
- 제조업체(Manufacturer) 선택 (OMRON 및 Delta Tau는 기본값입니다.)
- OMRON에서 CK3A 모델을 선택

- Select Amplifier(증폭기 선택)를 클릭합니다. 여기서는 변경할 필요가 없습니다.

- "Select Amplifier(증폭기 선택)"

- 입력(공급) 전압 선택
이 예에서는 110VAC = ~155.54VDC

- 저장(플로피 디스크 아이콘) 클릭



Start typing to filter items

Manufacturer: Omron

Manufacturer	Part Number
Omron	CK3A-G305L
Omron	CK3A-G310L

General

Manufacturer: * OMRON

Part Number: * CK3A-G305L

Control Type: * Velocity Torque Sinewave **Direct PWM** Direct Micro-Stepping

Signal Type: * Analog **PWM** Step&Direction

Power Ratings

Maximum Input Voltage: * 252 AC

Continuous Current: * 5 RMS Amps

Instantaneous Current: * 10 RMS Amps

Time Allowed: * 2 Seconds

Amplifier Fault Polarity: Low True **High True**

Current Feedback Information

Maximum ADC Current: * 15.735 Amps

ADC Header: 4 Bits

ADC Resolution: 16 Bits

PWM Dead-Time: * 2 Microseconds

Select Amplifier

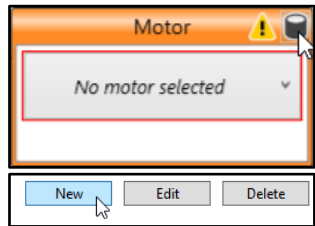
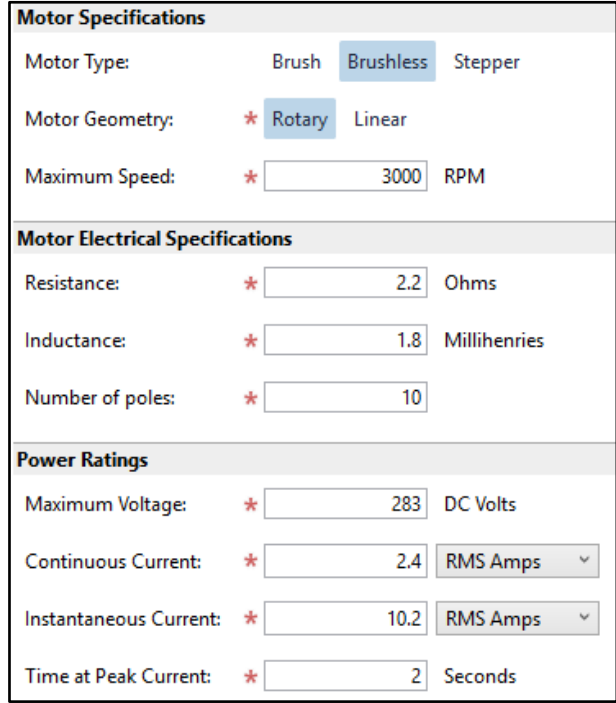
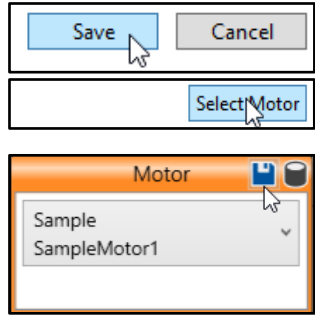
Amplifier

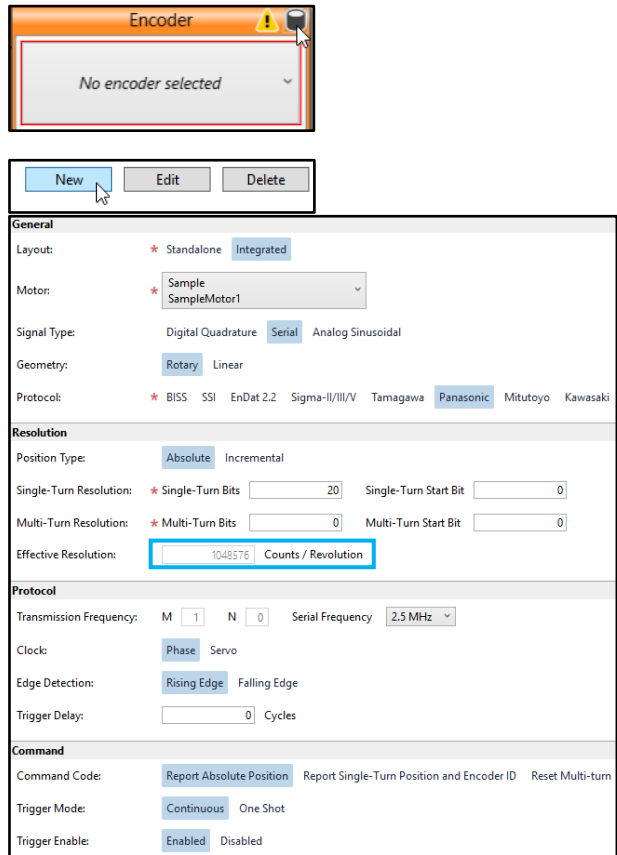
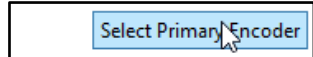
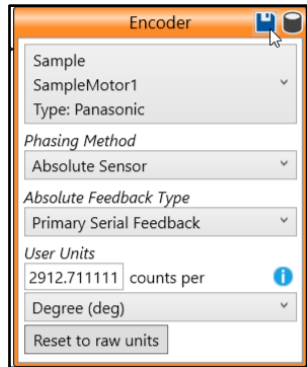
OMRON
CK3A-G305L

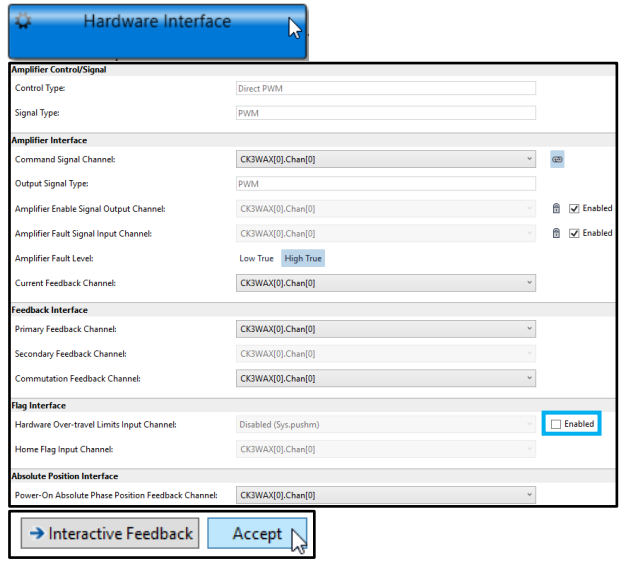
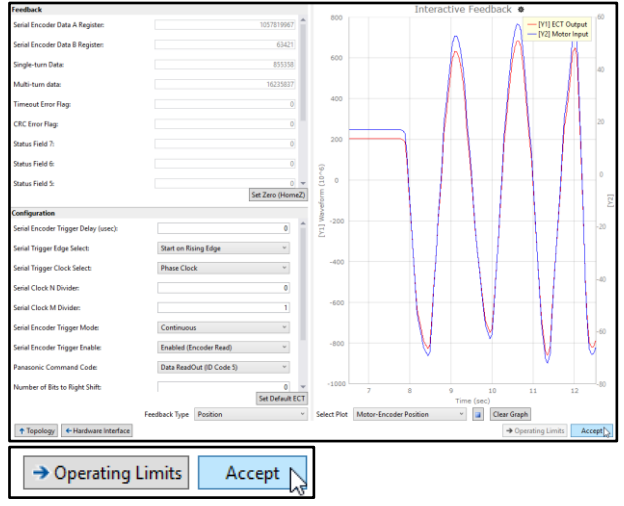
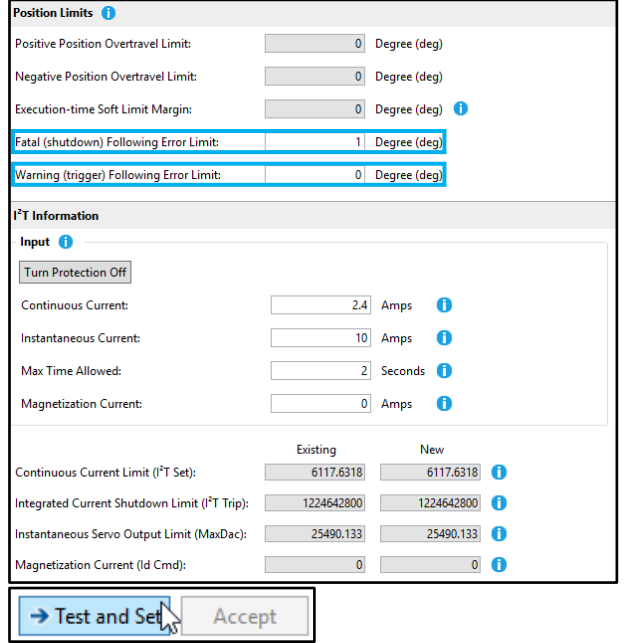
Control Type
Direct PWM

Signal Type
PWM

Input Voltage (VDC)
155.54

단계	그림
모터 지정 - 모터 세부 정보(데이터베이스 아이콘)를 클릭하거나 이전에 데이터베이스에서 저장한 모터(폴다운 메뉴)를 선택 - 새 모터의 경우 "New(신규)"를 클릭 "New(신규)" 모터 정보 입력	  
- 데이터베이스에 "Save(저장)" "Select Motor(모터 선택)" "Save(저장)"(데이터베이스 아이콘)	

단계	그림
인코더 지정 - 인코더 세부 정보(데이터베이스 아이콘)를 클릭하거나 이전에 데이터베이스에서 저장한 인코더(폴다운 메뉴)를 선택 - 새 인코더의 경우 "New(신규)"를 클릭 - 신규 인코더 정보를 입력합니다. 유효 분해능 결과를 기록합니다(다음 단계에서 사용자 단위 계산을 위해). - 데이터베이스에 "Save(저장)" "Select Primary Encoder(기본 인코더 선택)" - Phasing Method(위상 조정 방법)를 선택 - 위상 조정 방법에 대한 피드백 장치 선택(해당되는 경우) - 축 엔지니어링 사용자 단위(예: 도)를 선택 - 사용자 단위당 카운트 수를 입력합니다. = 이 예에서 도의 경우 1048576/360 - 완료되면 저장(플로피 디스크 아이콘)	   

단계	그림
하드웨어 인터페이스 "Hardware Interface(하드웨어 인터페이스)"를 클릭 - 일반적으로 이 페이지에서 고려해야 할 한 가지 항목은 초과 이동 제한입니다. 유선이 아닌 경우 비활성화해야 합니다. "Accept(수락)" - 다음 단계로 이동하려면 "Interactive Feedback(대화형 피드백)"	
대화형 피드백 - 이 페이지는 인코더 피드백을 확인하는 데 유용합니다. 일반적으로 가능하고 안전한 경우 손으로 모터를 옮겨 위치 변경을 모니터링합니다. - 보통 이 페이지에서는 변경 사항이 없습니다. "Accept(수락)" - 다음 단계로 이동하려면 "Operating Limits(작동 한계)"	
작동 한계 - 일반적으로 이 페이지에서 고려해야 할 항목은 "Fatal Following Error Limit(치명적인 후속 오류 제한)" 및 "Warning Following Error Limit(경고 후속 오류 제한)"입니다. - I2T 설정은 자동으로 계산됩니다. - 특히 모터의 정격이 CK3A보다 낮은 경우 절대로 I2T 보호를 끄지 않아야 합니다. "Accept(수락)" - 다음 단계로 이동하려면 "Test and Set(테스트 및 설정)"	

단계

그림

테스트 및 설정

- 이러한 테스트는 종종 높은 가속 속도로 모터를 움직입니다. 주의를 기울여야 합니다. 모터를 장착하고 단단히 고정해야 합니다.
- 사용자는 "Auto(자동)" 또는 "Manual(수동)"을 선택할 수 있습니다.
- "Auto(자동)"은 나열된 테스트를 순차적으로 수행합니다.
- "Auto(자동)" 모드에서 테스트가 실패하면 자동으로 "Manual(수동)"으로 전환됩니다.
- "Manual(수동)" 모드에서 사용자는 수행할 테스트를 선택할 수 있습니다.
- 테스트 중에 구조 요소 설정이 출력 창에 표시됩니다. 이는 문제 해결 연습에 유용할 수 있습니다.
- 또한 Power PMAC Messages 창에서 보다 자세한 메시지를 모니터링할 수 있습니다. 이는 문제 해결 연습에 유용할 수 있습니다.

- 계속하려면 모든 테스트를 통과해야 합니다.
- "Accept(수락)"
- "Basic Tuning(기본 튜닝)"

Auto Manual

Test No.	Description	Progress	Result
1	Detect current sensor direction	100%	Pass
2	Measure current sensor bias value	100%	Pass
3	Voltage six step test	100%	Pass
4	Tune current loop	100%	Pass
5	Current six step test	100%	Pass
6	Open loop test	100%	Pass
7	Phase reference search	100%	Pass

Test Start

Test Stop

Step No.	PowerPMAC Command	Value
3	Motor[1].PhaseOffset	683
3	Motor[1].PwmSf	13337
3	Motor[1].PhasePosSf	2048/(4096 * 1 * 209715.200000)

PowerPMAC Messages

0 Errors 315 Warnings 21 Messages 78 Outputs

Date	Location	Module	Description
12/5/2020 1:55:38 PM	Motor[1]		Sampling the current sensor reading with zero commands.
12/5/2020 1:55:38 PM	Motor[1]		Calculating Maximum ADC allowed based upon I2TSet and Checking for ADC Latch possibility.
12/5/2020 1:55:38 PM	Motor[1]		Maximum ADC allowed current = 1529 ADC bits
12/5/2020 1:55:38 PM	Motor[1]		Increasing the output to the selected excitation magnitude.
12/5/2020 1:55:38 PM	Motor[1]		The excitation magnitude will be limited to 780 (bits) to prevent an I2T fault.
12/5/2020 1:55:38 PM	Motor[1]		Sampling the current sensor readings with excitation values commanded.
12/5/2020 1:55:39 PM	Motor[1]		Disabling the amplifier.

Basic Tuning

Accept

Basic Tuning

Accept

기본(자동) 튜닝

- 이는 초기 사용자 입력이 없는 "Auto-Tuning(자동 튜닝)" 도구입니다.
- 프로세스를 시작하려면 "Start Tuning(튜닝 시작)"
- 이 식별 및 튜닝 프로세스 중에 모터는 종종 높은 가속도로 움직입니다. 주의를 기울여야 합니다. 모터를 장착하고 단단히 고정해야 합니다.

Tuning Criteria

Rigidity / Stiffness 0 0 0

Damping 0 1 0

Integral Ratio 0 1 0

Start Tuning

Commission

Accept

Test Start

The following test(s) will physically move the Motor. Do you want to continue?

OK

Cancel

- 결과는 단계 이동 응답의 플롯으로 나타납니다.

Basic Servo Loop Tuning

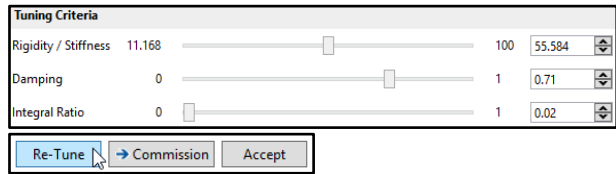
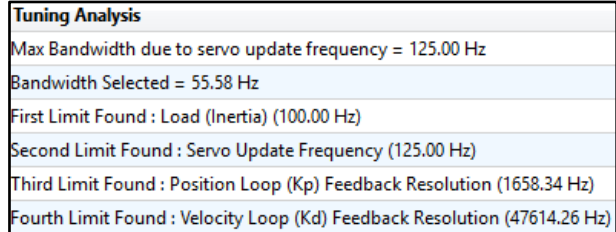
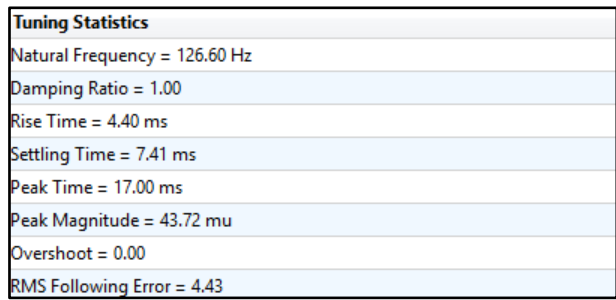
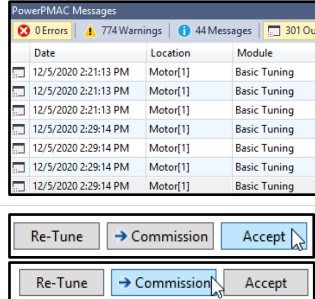
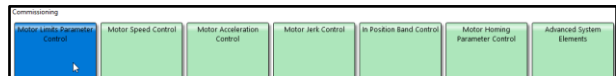
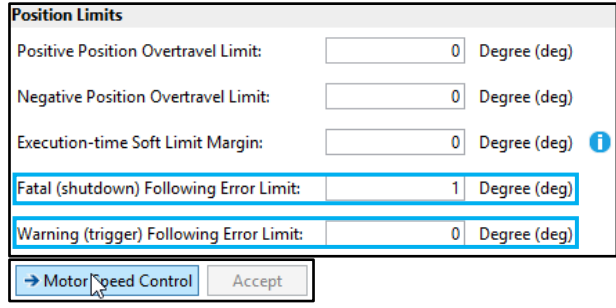
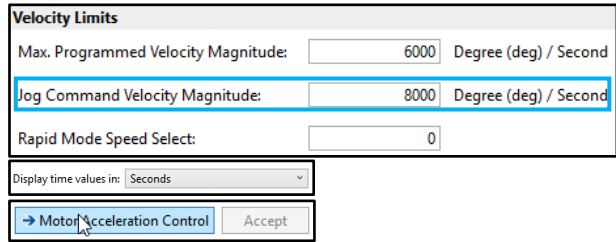
Position

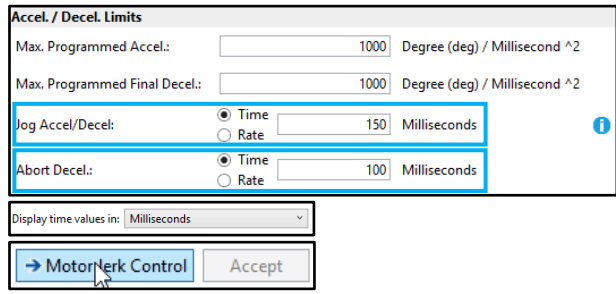
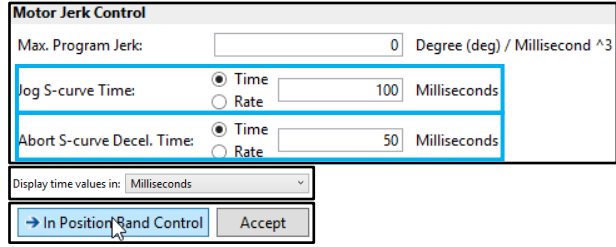
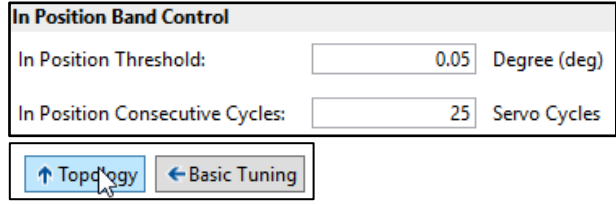
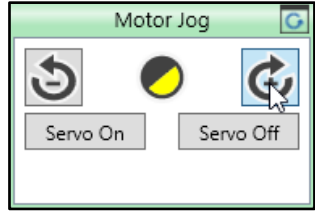
Time (msec)

Desired Position

Actual Position

Servo Command

단계	그림
- 결과를 자동 튜닝 알고리즘이 선택한 기준으로 나타냅니다. 이러한 매개변수(예: 적분 비율 0)를 수정하여 "Re-Tune(다시 튜닝)"할 수 있습니다. - 결과를 튜닝 분석으로 나타냅니다. 여기에는 대역폭 선택과 일부 시스템 제한 사항이 설명되어 있습니다. - 결과를 단계 이동 응답을 통계로 보여줍니다. - 참고 Power PMAC 메시지 출력 창은 자동 조정 프로세스 동안 정보를 제공합니다. - 완료되면 "Accept(수락)" - 다음 단계로 이동하려면 "Commission(시운전)"	   
시운전 시운전을 시작하려면 "Motor Limits Parameter control(모터 제한 매개변수 제어)"	
위치 제한 - 일반적으로 여기서 주요 고려 사항은 치명적인 후속 오류 제한과 경고 후속 오류 제한을 확인하는 것입니다. - 절대 원점 복귀 매개변수를 설정한 후 절대 인코더에 대한 포지티브 및 네거티브 소프트웨어 설정할 수 있습니다. 증분형 인코더의 경우 일반적으로 원점 복귀 루틴을 수행한 후 소프트웨어 제한이 설정됩니다. - 다음 항목으로 이동하려면 "Accept(수락)" 및 "Motor Speed Control(모터 속도 제어)"	
속도 제한 - 일반적으로 여기서 주요 고려 사항은 조그 속도(크기)입니다. 최대 프로그래밍 속도는 일반적으로 좌표계 설정과 함께 모션 프로그램에 대해 구성됩니다. - 사용자 입력 시간 단위를 선택할 수 있습니다.	

단계	그림
가속/감속 제한 - 일반적으로 여기서 주요 고려 사항은 조그 및 중단 선형 가속/감속 시간 또는 속도입니다. 최대 프로그래밍 가속/감속은 일반적으로 좌표계 설정과 함께 모션 프로그램에 대해 구성됩니다. - 다음 항목으로 이동하려면 "Accept(수락)" 및 "Motor Jerk Control(모터 저크 제어)"	
모터 저크 제어 - 일반적으로 여기서 주요 고려 사항은 조그 및 중단 선형 가속/감속 시간 또는 속도입니다. 최대 프로그래밍 가속/감속은 일반적으로 좌표계 설정과 함께 모션 프로그램에 대해 구성됩니다. - 다음 항목으로 이동하려면 "Accept(수락)" 및 "In Position Band Control(인 포지션 대역 제어)"	
모터 인 포지션 대역 제어 "In Position Threshold(인 포지션 임계값)"는 Motor[x].InPos 비트가 true로 설정되기 전에 "In Position Consecutive Cycles(인 포지션 연속 사이클)" 시간 동안 유지되는 후속 오류 "창"을 정의합니다. - 이것으로 필수 시운전을 마칩니다. - 종료하려면 "Topology(토폴로지)"	
모터 조그 이 제어판은 선택적 테스트(조그) 실행입니다.	

A-2 Power PMAC3 수동 모터 설정

이 섹션에서는 Gate3, ACC-84E 및 CK3A 다이렉트 PWM 증폭기가 있는 Power PMAC 컨트롤러로 브러시리스 모터를 수동으로 설정하는 단계별 절차를 설명합니다. 절차 단계는 다음과 같습니다.

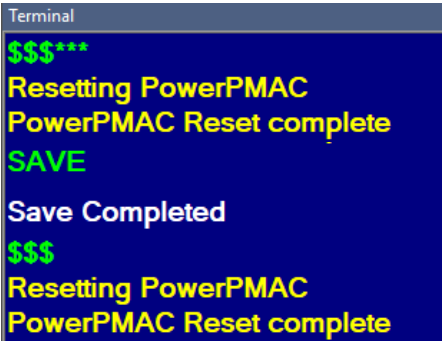
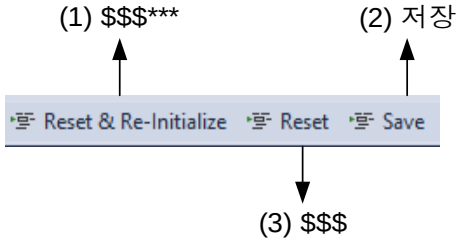
- 1 IDE 프로젝트 생성
- 2 기본 최적화, 시스템 및 게이트 설정
- 3 인코더 진행 위치, 확인
- 4 모터 설정 및 시운전

A-2-1 IDE 프로젝트 생성

공장 기본값 리셋

새 프로젝트의 경우 깨끗한 시작점을 보장하기 위해 공장 기본 설정에서 시작하는 것이 좋습니다. 공장 기본값 리셋은 전역 리셋 \$\$\$***, SAVE(저장) 후, 일반 리셋 \$\$\$을 실행하여 수행할 수 있습니다.

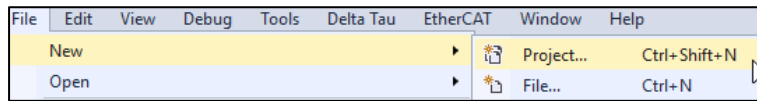
이 작업은 터미널 창에서 수행하거나 IDE 도구 모음 바로 가기를 사용하여 수행할 수 있습니다.

터미널 창	IDE 도구 모음
 <pre> Terminal \$\$\$*** Resetting PowerPMAC PowerPMAC Reset complete SAVE Save Completed \$\$\$ Resetting PowerPMAC PowerPMAC Reset complete </pre>	 <pre> (1) \$\$\$*** (2) 저장 ^ ^ [Reset & Re-Initialize] [Reset] [Save] v (3) \$\$\$ </pre>

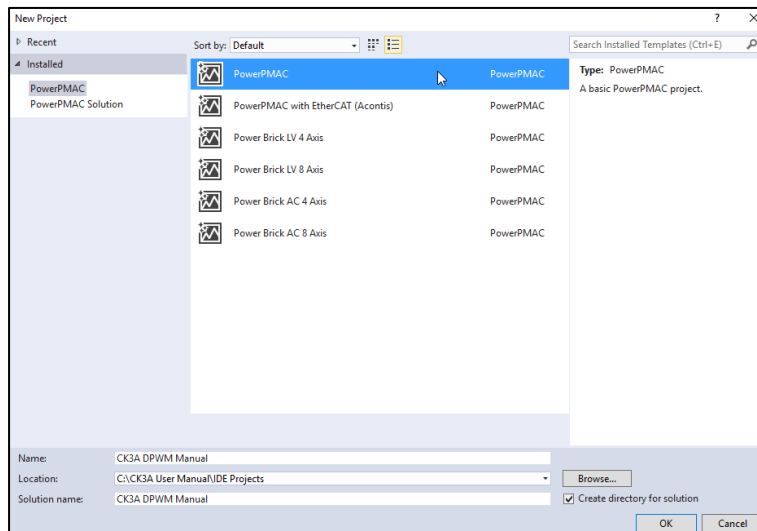
새 IDE 프로젝트

새 프로젝트 생성은 파일 메뉴에서 수행합니다.

- File(파일) > New(새로 만들기) > Project(프로젝트)



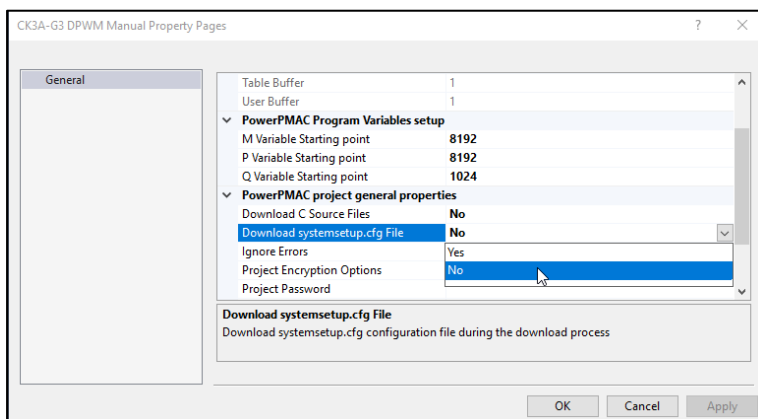
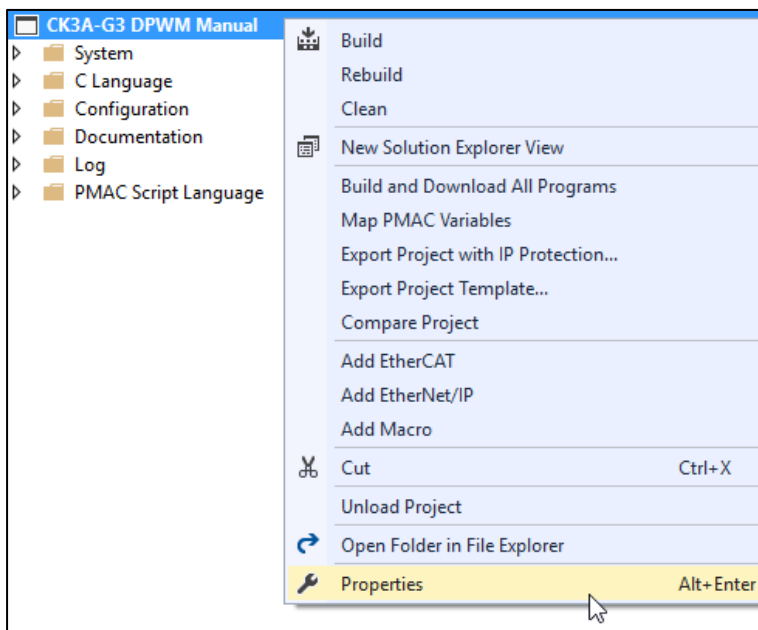
- 템플릿 선택
- 프로젝트 위치 선택
- 프로젝트 이름 입력



Systemsetup.cfg

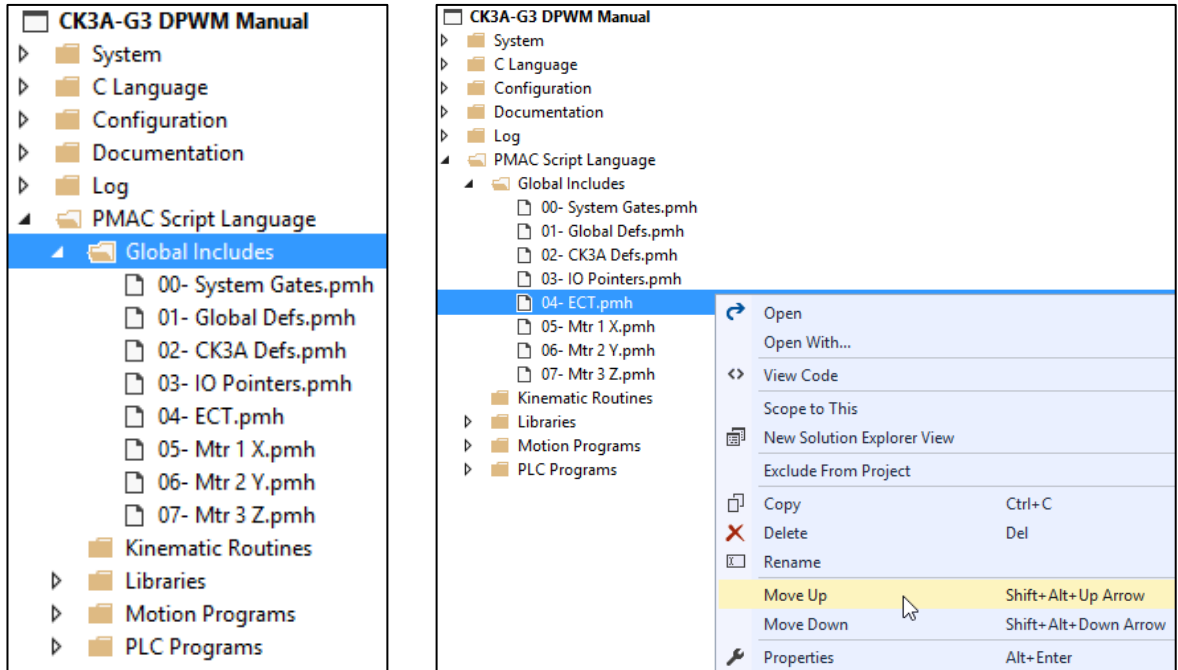
IDE 시스템 설정 도구를 사용하지 않는 경우 systemsetup.cfg 파일을 비활성화해야 합니다.

- 프로젝트 이름을 마우스 오른쪽 버튼으로 클릭
- Properties(속성)
- Download Systemsetup.cfg File(Systemsetup.cfg 파일 다운로드) > NO(아니요) > OK(확인)



프로젝트 조직

다음 섹션에서 설명하는 대부분의 매개변수는 일반적으로 **Global Includes** 아래에 배치됩니다. 이 폴더의 파일은 사용자 기본 설정별로 관리할 수 있습니다. 기존 프로젝트에서 추가, 삭제, 삽입, 이름 바꾸기, 정렬(위아래로 이동) 등을 할 수 있습니다. 이러한 조작에 대해 알아보려면 IDE 설명서를 참조하십시오. 제안된 분류 중 하나가 아래에 나와 있습니다.



다음은 제안된 이러한 파일에 대한 간략한 설명입니다.

파일	일반적인 내용	내용 예
System Gates	시스템 구조 요소	Sys.MaxMotors
	Gate3 구조 요소	Gate3[0].PhaseFreq
	채널 구조 요소	Gate3[0].Chan[0].PwmFreqMult
Global Defs	사용자 정의 변수	GLOBAL MyVariable
CK3A Defs	CK3A 특정 변수	GLOBAL Ck3a1DcBus
IO Pointers	디지털 I/O 포인터	PTR Input1->Gate3[0].GpioData[0].0.1
	아날로그 I/O 포인터	PTR AdcIn1->S:IO:\$900040.16.16
ECT	인코더 변환표	EncTable[1].type
Mtr1 X	모터 구조 요소	Motor[1].ServoCtrl

A-2-2 기본 최적화, 시스템 및 게이트 설정

이 섹션에서는 기본 최적화, 클럭 주파수 및 CK3A 특정 구조 요소에 대해 설명합니다.

CK3A-G3 DPWM Manual

- ▶ System
- ▶ C Language
- ▶ Configuration
- ▶ Documentation
- ▶ Log
- ▶ PMAC Script Language
 - ▶ Global Includes
 - 00- System Gates.pmh**
 - 01- Global Defs.pmh
 - 02- CK3A Defs.pmh
 - 03- IO Pointers.pmh
 - 04- ECT.pmh
 - 05- Mtr 1 X.pmh
 - 06- Mtr 2 Y.pmh
 - 07- Mtr 3 Z.pmh
 - ▶ Kinematic Routines
 - ▶ Libraries
 - ▶ Motion Programs
 - ▶ PLC Programs

```

00- System Gates.pmh
Sys.WpKey = $AAAAAAAA
Sys.MaxCoords = 2
Sys.MaxMotors = 5

Gate3[0].PhaseFreq = 10000
Gate3[0].ServoClockDiv = 1
Sys.RtIntPeriod = 1
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)

Gate3[0].Chan[0].PwmFreqMult = 1
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533
Gate3[0].AdcAmpStrobe = $901001
Gate3[0].AdcAmpClockDiv = 5
Gate3[0].AdcAmpHeaderBits = 4

Gate3[0].Chan[0].PackInData = 0
Gate3[0].Chan[1].PackInData = 0
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0
Gate3[0].Chan[1].PackOutData = 0
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

```

구조 요소	설명 및 예
Sys.WpKey	Gate3용 쓰기 방지 키 쓰기 방지 비활성화, 일반적으로, = \$AAAAAAAA
Sys.MaxCoords	사용할 좌표계의 최대 수 + 1 예, = 2(좌표계 0..1 사용, 0은 일반적으로 사용되지 않음)
Sys.MaxMotors	사용할 최대 모터 수 + 1 예, = 5(모터 #0..4 사용, 0은 일반적으로 사용되지 않음)
Gate3[0].PhaseFreq	위상 클럭 주파수([Hz]) 예, = 10000(10KHz)
Gate3[0].ServoClockDiv	서보 클럭 제수(서보 주파수 정의) 서보 주파수 = 위상 주파수 / (ServoClockDiv + 1) 예, = 1(5KHz)
Sys.ServoPeriod	보간을 위한 서보 업데이트 주기 올바른 주파수 설정에 필요합니다. 고정식으로 설정할 수 있습니다. = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq
Sys.PhaseOverServoPeriod	서보에 대한 위상 비율 올바른 주파수 설정에 필요합니다. 고정식으로 설정할 수 있습니다. = 1 / (Gate3[0].ServoClockDiv + 1)
Gate3[i].Chan[j].PwmFreqMult	PWM 주파수 승수(PWM 주파수 정의) PWM 주파수 = (PwmFreqMult + 1) * PhaseFreq / 2 예, = 1 (10 KHz ~ CK3A 범위는 8 ~ 20 KHz)
Gate3[i].Chan[j].PwmDeadTime	PWM 데드타임 항상 = 2 / 0.0533(CK3A-G305L의 경우 2μsec) 항상 = 3 / 0.0533(CK3A-G310L의 경우 3μsec)
Gate3[i].AdcAmpStrobe	증폭기 ADC 스트로브 워드(CK3A 전용) 예, = \$901001(이 설명서에 자세히 설명되어 있음)
Gate3[i].AdcAmpClockDiv	주파수 A/DC 컨버터 클럭 주파수(CK3A 전용) CK3A에 허용되는 Gate3 설정: = 5(3.125MHz), = 4(6.250MHz)

Gate3[i].AdcAmpHeaderBits	헤더 비트의 증폭기 ADC 수(CK3A 특징) 항상 = 4(CK3A의 경우)
Gate3[i].Chan[j].PackInData	ADC 입력 팩 데이터 활성화 항상 = 0(CK3A의 경우)
Gate3[i].Chan[j].PackOutData	ADC 출력 팩 데이터 활성화 항상 = 0(CK3A의 경우)



추가 정보

자세한 설명은 Power PMAC 소프트웨어 참조 설명서에서 찾을 수 있습니다.

A-2-3 인코더 진행 위치, 확인

인코더 변환표 ECT는 진행 중인 실제 위치를 생성하는 기본 인코더 입력 처리 구조입니다. ECT의 결과는 일반적으로 모터 서보 루프에서 사용됩니다.

```

04- ECT.pmh
EncTable[1].Type = 1
EncTable[1].pEnc = Gate3[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 12
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(12)

EncTable[2].Type = 0

```



추가 정보

- 자세한 설명은 Power PMAC 소프트웨어 참조 설명서를 참조하십시오.
- EncTable[n].Type=0은 ECT의 끝을 나타냅니다(효율성을 위해 강력히 권장됨).

아래는 가장 일반적인 인코더 피드백 장치 유형에 대한 ECT 항목 설정에 대한 일반적인 지침입니다.



추가 정보

EncTable[n].Index1 ~ 6은 일반적으로 생략된 경우 =0입니다.

디지털 직각위상용 ECT 항목

구조 요소	일반 설정
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1 / 256

표준 아날로그 정현파 x16384용 ECT 항목(UMAC 및 CK3M)

구조 요소	일반 설정
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1

● 이 유형의 처리에 필요한 기타 UMAC(ACC-24E3) 설정

구조 요소	설정
Motor[x].EncType	= 6
Gate3[i].Chan[j].AtanEna	= 1

● 이 유형의 처리에 필요한 기타 CK3M(CK3W-AX2323□) 설정

구조 요소	설정
Motor[x].EncType	= 6
Gate3[i].Chan[j].AtanEna	= 1
Gate3[i].EncClockDiv	= 3
Gate3[i].AdcEncClockDiv	= 3
Gate3[i].AdcEncCtrl	= \$3FFFC000

아날로그 정현파 ACI x65536(UMAC)용 ECT 항목

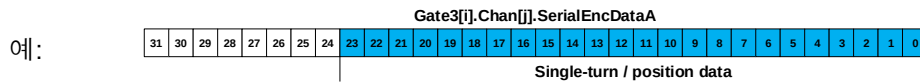
구조 요소	일반 설정
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Gate3[i].Chan[j].AtanSumOfSqr.a
EncTable[n].ScaleFactor	= 1

● 이 유형의 처리에 필요한 기타 설정

구조 요소	설정
Motor[x].EncType	= 7
Gate3[i].Chan[j].AtanEna	= 1
Gate3[i].AdcEncHeaderBits	= 0
Gate3[i].AdcEncStrobe	= \$800000

Gate3이 포함 직렬용 ECT 항목(CK3W-AX 및 ACC-24E3)

구조 요소	일반 설정
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= 비트 #31 + 인덱스2로 왼쪽으로 이동할 데이터 비트 수. 이것은 2차 이동 연산입니다. 아래 다이어그램 예의 경우, = 8
EncTable[n].Index2	= 비트 #0으로 오른쪽으로 이동할 데이터 비트 수(드문 경우). 이것은 1차 이동 연산입니다. 아래 다이어그램 예의 경우, = 0
EncTable[n].ScaleFactor	= $1 / 2^{\text{EncTable[n].Index1}}$ 아래 다이어그램 예의 경우, = $1 / 2^8$



● 이 유형의 처리에 필요한 기타 설정

구조 요소	설정
Gate3[i].Chan[j].SerialEncEna	= 1
Gate3[i].SerialEncCtrl	Power PMAC 소프트웨어 참조 설명서 참조
Gate3[i].Chan[j].SerialEncCmd	Power PMAC 소프트웨어 참조 설명서 참조

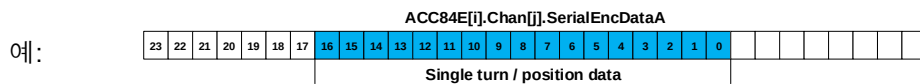


추가 정보

Gate3[i].SerialEncCtrl 및 Gate3[i].Chan[j].SerialEncCmd를 성공적으로 설정하면 Gate3[i].Chan[j].SerialEncEna=1 원시 직렬 인코더 데이터를 Gate3[i].Chan[j].SerialEncDataA 및 Gate3[i].Chan[j].SerialEncDataB에서 볼 수 있습니다. 프로토콜에 따라 일부 정보는 Gate3[i].Chan[j].SerialEncDataC 또는 Gate3[i].Chan[j].SerialEncDataD에 있을 수 있습니다.

ACC-84E(UMAC)가 포함된 직렬용 ECT 항목

구조 요소	일반 설정
EncTable[n].Type	= 1
EncTable[n].pEnc	= ACC84E[i].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= 비트 #31 + 인덱스2로 왼쪽으로 이동할 데이터 비트 수. 이것은 2차 이동 연산입니다. 아래 다이어그램 예의 경우, = 15
EncTable[n].Index2	= 8 + 비트 #0으로 오른쪽으로 이동할 데이터 비트 수(드문 경우). 이것은 1차 이동 연산입니다. 아래 다이어그램 예의 경우, = 8
EncTable[n].ScaleFactor	= $1 / 2^{\text{EncTable[n].Index1}}$ 아래 다이어그램 예의 경우, = $1 / 2^{15}$



● 이 유형의 처리에 필요한 기타 설정

구조 요소	설정
ACC84E[i].SerialEncCtrl	Power PMAC 소프트웨어 참조 설명서 참조
ACC84E[i].Chan[0].SerialEncCmd	Power PMAC 소프트웨어 참조 설명서 참조



추가 정보

ACC84E[i].SerialEncCtrl 및 ACC84E[i].Chan[j].SerialEncCmd를 성공적으로 설정하면 원시 직렬 인코더 데이터는 ACC84E[i].Chan[j].SerialEncDataA 및 ACC84E[i].Chan[j].SerialEncDataB에서 볼 수 있습니다. 프로토콜에 따라 일부 정보는 ACC84E[i].Chan[j].SerialEncDataC 또는 ACC84E[i].Chan[j].SerialEncDataD에 있을 수 있습니다.

인코더 피드백 완료 및 확인



안전한 사용을 위한 주의사항

- 인코더 데이터가 없으면 폐쇄 루프 제어에서 매우 위험한 상태가 될 수 있습니다. 왜냐하면 서보 루프는 모터의 실제 물리적 위치(보통 모터가 "붙어 있다"고 생각함)를 더 이상 인식하지 못하고 격렬하게 반응하여 종종 폭주 상태가 발생할 수 있기 때문입니다.
- 모터 폐쇄 루프 명령을 실행하기 전에 인코더 데이터가 업데이트되고 올바르게 카운팅되는지 확인하십시오.
- Motor[x].PosSf 및 Motor[x].Pos2Sf는 서보 루프를 포함한 Motor 구조 요소의 일부입니다. 초기 설정 후 변경하면 가속, 속도 설정 및 서보 루프 게인과 같은 다른 많은 요소를 조정해야 합니다.

ECT 항목의 결과 출력은 일반적으로 아래 구조 요소를 사용하여 위치 및 속도 서보 루프에서 사용되는 모터에 연결됩니다.



추가 정보

- ECT의 (처리된) 출력은 EncTable[n].PrevEnc에 나타납니다.
- EncTable[n].PrevEnc 데이터는 EncTable[n].ScaleFactor로 곱해지지 않습니다.

구조 요소	설명 및 예
Motor[x].ServoCtrl	모터 채널 활성화 일반적으로, = 1
Motor[x].pEnc	위치 피드백 포인터 일반, = EncTable[n].a(해당 ECT)
Motor[x].pEnc2	속도 피드백 포인터, 단일 피드백의 경우 항상 =Motor[x].pEnc 일반, = EncTable[n].a(해당 ECT)
Motor[x].PosSf	위치 환산 계수(사용자 단위로 변환 가능) 20비트 로터리 모터(도), = 360 / 1048576 1nm 스케일의 리니어 모터, = 1 / 1000000
Motor[x].Pos2Sf	속도 환산 계수, 단일 피드백의 경우 항상 =Motor[x].PosSf 20비트 로터리 모터(도), = 360 / 1048576 1nm 스케일의 리니어 모터, = 1 / 1000000
Motor[x].Posunit	IDE 위치 창 단위 설정 도의 경우, = 11 mm의 경우, = 3

● 양방향으로 인코더 카운팅 확인

IDE에서 위치 창을 모니터링하면서, 시계 방향, 시계 반대 방향, 정방향 또는 역방향 등 손으로 모터를 움직여 이를 수행할 수 있습니다.

또한 사용자는 정지 상태에서 위치가 안정적인지(본래의 디더링 양 내) 확인해야 합니다.

문제 해결을 위해 IDE 위치 창에 위치 변경이 표시되지 않으면 "원시 인코더 수"를 확인하는 것이 도움이 될 수 있습니다. 아래 표는 인코더 유형별로 이러한 레지스터를 보여줍니다.

인코더 "처리" 유형	원시 카운트 구조 요소
디지털 직각위상	Gate3[i].Chan[j].ServoCapt
아날로그 정현파	Gate3[i].Chan[j].ServoCapt
아날로그 정현파 ACI	Gate3[i].Chan[j].ServoCapt
Gate3이 포함된 직렬	Gate3[i].Chan[j].SerialEncDataA Gate3[i].Chan[j].SerialEncDataB(항상 데이터가 있는 것은 아님)
ACC-84E가 포함된 직렬	ACC84E[i].Chan[j].SerialEncDataA ACC84E[i].Chan[j].SerialEncDataB(항상 데이터가 있는 것은 아님)

● 정확한 거리 보고

사용자는 모터를 알려진 양만큼 움직이고 IDE 위치 창에 표시된 경과 거리를 기록하여 피드백 장치가 올바르게 계산하고 있는지 확인할 수 있습니다. 경우에 따라 #nHMZ(여기서 n은 모터 번호) 명령을 사용하여 위치 표시를 제로화할 수 있습니다.

카운팅이 잘못된 경우 EncTable[n].ScaleFactor, Motor[x].PosSf, Motor[x].Pos2Sf 및 Motor[x].EncType이 올바르게 설정되었는지 확인하십시오.



추가 정보

Motor[x].pLimits는 하드웨어가 인식하고 펌웨어에 의해 자동으로 설정됩니다.
Motor[x].pLimits는 하드웨어 초과 이동 제한의 소스(플래그 포인터)를 지정합니다.
연결되지 않았거나 비활성화하려는 경우 Motor[x].pLimits를 =0으로 설정할 수 있습니다.

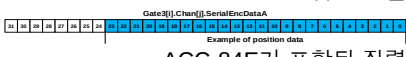
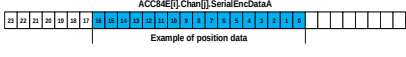
공통 증폭기 및 다이렉트 PWM 제어 요소

구조 요소	설명
Motor[x].AmpFaultLevel	증폭기 결함 로직 상태 항상 = 1(CK3A의 경우)
Motor[x].AdcMask	전류 피드백 분해능 지정(비트 단위) 항상 = \$FFFF0000(CK3A의 경우)
Motor[x].PhaseOffset	모터 위상 사이의 각도 거리 일반적으로, = 683
Motor[x].PwmSf	PWM 출력 환산 계수(분압기) 주 공급 전압보다 크거나 같은 모터 정격 전압, = 16384 * 0.95 주 공급 전압보다 낮은 모터 정격 전압, = 16384*모터 전압/주 전압

진행 중인 단계(정류) 위치

다음 정의는 아래 설명에서 사용됩니다.

이름	설명
ST	단일 회전 직렬 로터리 인코더 위치 데이터
NoOfPolePairs	로터리 브러시리스 모터의 쌍극 수
CtsPerRev	로터리 모터의 회전당 카운트 수(일반적으로 라인 x4)
LinesPerRev	정현파 로터리 인코더의 사인 사이클 수
ECLmm	리니어 모터 전기 사이클 길이 또는 자기 피치(mm)(예: 30.48mm)
RESmm	mm 단위의 리니어 인코더 분해능(예: 1µm = 0.001mm)

구조 요소	설명 및 예
Motor[x].PhaseCtrl	모터 정류 제어 일반적으로, = 4
Motor[x].pPhaseEnc	정류 진행 위치(포인터) 소스 디지털 직각위상 = Gate3[i].Chan[j].PhaseCapt.a 아날로그 정현파 = Gate3[i].Chan[j].PhaseCapt.a 아날로그 정현파 ACI = Gate3[i].Chan[j].PhaseCapt.a Gate3이 포함된 직렬 = Gate3[i].Chan[j].SerialEncDataA.a ACC-84E가 포함된 직렬 = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].PhaseEncRightShift	위상 위치 소스 데이터를 오른쪽으로(비트 #0으로) 이동하는 비트 수 디지털 직각위상 = 0 아날로그 정현파 = 0 아날로그 정현파 ACI = 0 Gate3이 포함된 직렬 이 예의 경우 = 0  (Gate3가 있는 대부분의 직렬 프로토콜) ACC-84E가 포함된 직렬  이 예의 경우 = 8 (ACC-84E 사용 시 일반적)

Motor[x].PhaseEncLeftShift	위상 위치 소스 데이터를 왼쪽으로(비트 #31으로) 이동하는 비트 수 디지털 직각위상 = 0 아날로그 정현파 = 0 아날로그 정현파 ACI = 0 Gate3이 포함된 직렬 이 예의 경우 = 8 ACC-84E가 포함된 직렬 이 예의 경우 = 15 (7 데이터 비트 + PhaseEncRightShift) Example of position data Example of position data
Motor[x].PhasePosSf	정류 각도 환산 계수 디지털 직각위상 아날로그 정현파 아날로그 정현파 ACI Gate3이 포함된 직렬 ACC-84E가 포함된 직렬 로터리 모터 = $2048 * \text{NoOfPolePairs} / \text{CtsPerRev}$ 리니어 모터 = $2048 * \text{RES}_{\text{mm}} / (256 * \text{ECL}_{\text{mm}})$ 로터리 모터 = $2048 * \text{NoOfPolePairs} / (\text{LinesPerRev} * 16384)$ 리니어 모터 = $2048 * \text{RES}_{\text{mm}} / (16384 * \text{ECL}_{\text{mm}})$ 로터리 모터 = $2048 * 4 * \text{NoOfPolePairs} / (\text{LinesPerRev} * 65536)$ 리니어 모터 = $2048 * 4 * \text{RES}_{\text{mm}} / (65536 * \text{ECL}_{\text{mm}})$ 로터리 모터 = $2048 * \text{NoOfPolePairs} / (2^{\text{Motor[x].PhaseEncLeftShift} + \text{ST}})$ 리니어 모터 = $2048 * \text{RES}_{\text{mm}} / (2^{\text{Motor[x].PhaseEncLeftShift} * \text{ECL}_{\text{mm}}})$ 로터리 모터 = $2048 * \text{NoOfPolePairs} / (2^{\text{Motor[x].PhaseEncLeftShift} + \text{ST}})$ 리니어 모터 = $2048 * \text{RES}_{\text{mm}} / (2^{\text{Motor[x].PhaseEncLeftShift} * \text{ECL}_{\text{mm}}})$

PMAC I2T 보호



안전한 사용을 위한 주의사항

- PMAC I2T를 올바르게 설정해야 합니다. 모터 전류가 증폭기보다 낮은 경우에는 특히 그렇습니다. 그렇게 하지 않으면 모터가 손상될 위험이 있습니다.
- 증폭기와 모터 사이의 더 낮은 전류 사양은 적절한 보호를 위해 항상 I2T 계산에 사용되어야 합니다.

CK3A 증폭기에는 과열로부터 보호하기 위해 자체 I2T 보호 기능이 내장되어 있으며 이는 구성할 수 없습니다. 이 섹션에서는 모터 과열을 방지하는 PMAC I2T 기능의 설정에 대해 설명합니다.

Motor[x].I2tSum(적분기 값)이 Motor[x].I2tTrip에 도달하면 PMAC I2T 결함이 발생합니다. 이 경우 증폭기 활성화 신호가 즉시 꺼짐(모터 정지)으로 설정되고 증폭기 결함 및 I2T 결함 상태 비트가 설정됩니다(IDE 모터 상태 창 참조).

이러한 상태 비트는 모터 구조 요소 Motor[x].AmpFault 및 Motor[x].I2tFault를 사용하여 액세스할 수 있습니다.

Status			
Motor Status		Global Status	
Motor 1		Motor activated	
Description	Status	Description	Status
AmpEna	False	I2tFault	True
AmpFault	True	InPos	False
AmpWarn	False	InterlockStop	False
AuxFault	False	LimitStop	False
BiDir	Plus	MinusLimit	False
BlockRequest	False	PhaseFound	True
ClosedLoop	False	PlusLimit	False
Csolve	False	SoftLimit	False
DacLimit	False	SoftLimitDir	Plus
DesVelZero	True	SoftMinusLimit	False
EncLoss	False	SoftPlusLimit	False
FeFatal	False	SpindleMotor	False
FeWarn	False	TraceCount	0
GantryHomed	False	TriggerMove	False
HomeComplete	True	TriggerNotFound	False
HomeInProgress	False	TriggerSpeedSel	MaxSpeed

PMAC I2T 설정에는 최대 전류, 정격 전류, 최대 전류에서 허용되는 시간 및 증폭기의 전류 스케일링에 대한 지식이 필요합니다. 다음 표에는 CK3A 앰프 사양이 나와 있습니다.

사양	CK3A-G305L	CK3A-G310L
최대 RMS 전류	5A _{rms}	10A _{rms}
정격 RMS 전류	10A _{rms}	20A _{rms}
전류 스케일링(최대 ADC)	15.735A _{peak}	31.470A _{peak}
최대 전류에서 허용되는 시간	2sec	2sec

증폭기와 모터 사이의 더 낮은 전류 사양은 적절한 보호를 위해 항상 I2T 계산에 사용되어야 합니다. 다음 표에는 이 선택 프로세스가 요약되어 있습니다.

정격 RMS 전류 조건	I2T 계산에 사용할 사양
모터 ≤ 증폭기	모터 정격 RMS 전류
모터 > 증폭기	증폭기 정격 RMS 전류
최대 RMS 전류 조건	I2T 계산에 사용할 사양
모터 ≤ 증폭기	- 모터 최대 RMS 전류 - 최대 전류에서 허용되는 모터 시간
모터 > 증폭기	- 증폭기 최대 RMS 전류 - 최대 전류에서 허용되는 증폭기 시간



추가 정보

CK3A 증폭기 정격 및 최대 전류 사양은 RMS로 제공됩니다. 모터 전류 사양이 "피크"로 지정된 경우 $\sqrt{2}$ 로 나누어야 합니다.

다음 정의를 가정하면 PMAC I2T를 설정하기 위한 표현은 다음과 같습니다.

이름	설명
MaxRmsCur	모터와 증폭기 사이의 최저 최대 RMS 전류
RatedRmsCur	모터와 증폭기 사이의 최저 정격 RMS 전류
TimeAtMaxCur	최대 전류 시간(초)
Ck3aMaxAdc	증폭기 전류 스케일링(암페어 피크로 정의)

구조 요소	설명 및 예
Motor[x].MaxDac	최대 서보 출력 $= \text{MaxRmsCur} * 32768 * \text{SQRT}(2) * \text{COSD}(30) / \text{Ck3aMaxAdc}$
Motor[x].I2tSet	연속 전류 제한 $= \text{RatedRmsCur} * 32768 * \text{SQRT}(2) * \text{COSD}(30) / \text{Ck3aMaxAdc}$
Motor[x].I2tTrip	최대(종료) 전류 제한 $= ((\text{Motor}[x].\text{MaxDac} * \text{Motor}[x].\text{MaxDac}) - (\text{Motor}[x].\text{I2tSet} * \text{Motor}[x].\text{I2tSet})) * \text{TimeAtMaxCur}$



추가 정보

SQRT() 및 COSD()는 PMAC 스크립트에서 제공된 및 코사인(도)입니다.

전류 루프 튜닝

이 섹션에서는 IDE의 수동 전류 루프 튜닝 도구에 대한 개요를 보여줍니다.

[50 – 100] % →

[25 – 50] % →

[10 – 100] msec →

보수적인 시작, = 0.1 이하 →

보수적인 시작, = liGain * 10 →

Auto
Interactive

Criteria

Step Size: % I2tSet

Rough Phasing Magnitude: % I2tSet ⓘ

Step Time: msec

Gains

Integral: liGain

Forward Path Proportional: lpfGain

Back Path Proportional: lpbGain

Phase A Current Offset: IaBias

Phase B Current Offset: IbBias

필드	설명 및 범위
크기	현재 단계 중에 인가된 전류 일반적인 범위, [Motor[x].I2tSet/2 ~ Motor[x].I2tSet]
러프 위상 조정 크기	전류 스텝 전에 모터를 제로 전기 사이클로 강제 적용하기 위해 인가된 전류 일반적인 범위, [Motor[x].I2tSet/4 ~ Motor[x].I2tSet/2]
드웰 시간	전류 스텝 시간 일반적인 범위, [10 - 100] msec
구조 요소	설명 및 예
Motor[x].liGain	전류 루프 적분 게인 보수적인 시작, = 0.1 이하
Motor[x].lpfGain	전류 루프 순방향 경로 비례 게인 보수적인 시작, = Motor[x].liGain * 10
Motor[x].lpbGain	전류 루프 역방향 경로 비례 게인 이 예제 가이드에서는 사용되지 않음

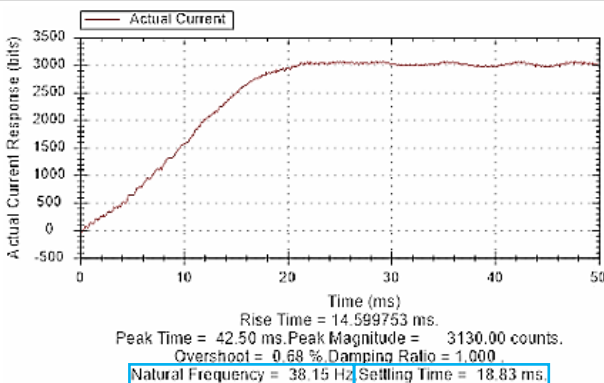
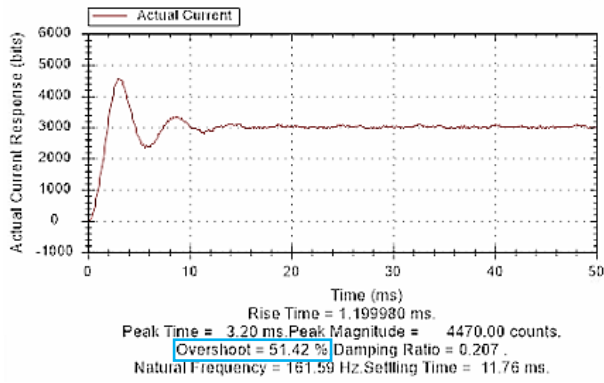
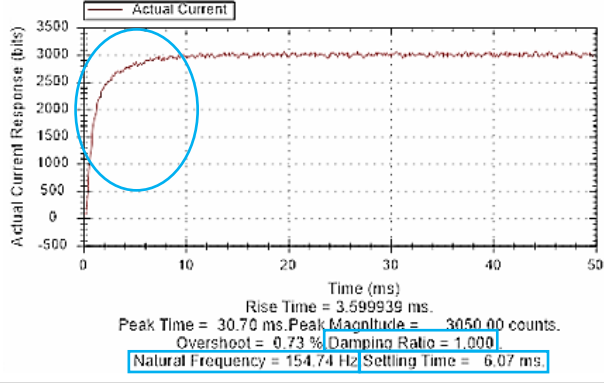
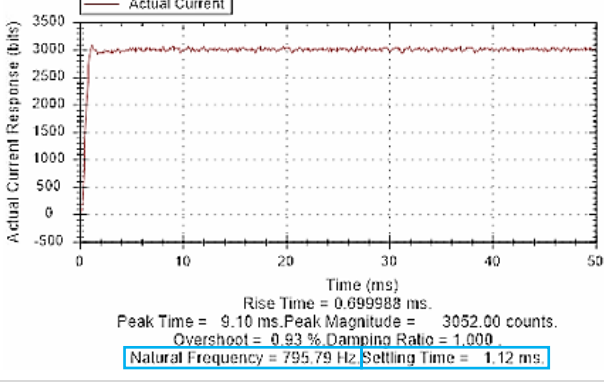
대부분의 최신 모터의 전류 루프는 Motor[x].liGain 및 Motor[x].lpfGain만 사용하여 튜닝할 수 있습니다. 목표는 200 ~ 1200Hz 범위(일반적으로 위치 루프보다 10배 빠름)의 단계 응답 고유 주파수, 1 ~ 3msec 이하의 정착 시간 및 3 ~ 5% 이하의 오버슈트를 달성하는 것입니다. 이러한 특성은 대부분의 서보 모터 용도에 적합합니다.



추가 정보

모터 및 증폭기 매개변수에 대한 기본 지식이 있으면 전류 루프 게인을 분석적으로 계산(시작)할 수 있습니다. 이 내용은 Power PMAC 사용 설명서에 설명되어 있습니다.

아래는 전류 루프 튜닝을 위한 일반적인 지침입니다.

계단 응답	관찰	대책
	"느린" 응답. 매우 낮은 고유 주파수와 긴 정착 시간.	Motor[x].liGain 및 Motor[x].lpfGain을 비례적으로 높이십시오.
	"감쇠 부족" 응답. 오버슈트가 너무 많습니다.	Motor[x].lpfGain을 높이십시오.
	"과도한 감쇠" 응답(곡선 모양). 고유 주파수가 낮아지고 정착 시간이 길어집니다.	Motor[x].liGain을 높이고 Motor[x].lpfGain을 낮추십시오.
	좋은 "매우 감쇠" 응답. 고유 주파수 ~800Hz, 정착 시간 ~1msec.	없음(종료).



추가 정보

전류 루프를 과도하게 조이면 특히 응답이 느린 모터에서 응답의 오버슈트 및 진동이 발생할 수 있습니다. 이는 위치 루프 성능에 악영향을 미칠 수 있으므로 바람직하지 않습니다.

위상 기준 설정(모터 위상 조정)



안전한 사용을 위한 주의사항

이 섹션에서 모터 위상을 조정하면 모터가 움직일 수 있습니다. 주의해야 합니다.

이 섹션에서는 가장 일반적인 두 가지 "초기" 위상 조정 방법에 대해 설명합니다. 두 방법 사이에는 근본적인 차이점이 없습니다. 두 방법 모두 모든 유형의 브러시리스 모터 및 인코더와 함께 사용할 수 있습니다. 아래 표는 제안 및 추가 정보를 제공합니다.

"초기" 위상 조정	권장 사용	최종 용도별 위상 조정 방법
수동 위상 조정	모터 어셈블리에 절대 센서가 있는 경우.	절대 센서를 사용하여 절대 위상 조정 설정(이후 섹션에서 설명)
자동 스텝퍼	모터 어셈블리에 절대 센서가 없는 경우.	스텝퍼 위상 조정 방법 매개변수를 PMAC에 저장하고 전원을 켜 후 위상 명령을 실행합니다.



추가 정보

- 위상 기준 설정에 대한 자세한 내용은 Power PMAC 사용 설명서를 참조하십시오.
- 모터 어셈블리에 홀 또는 절대 인코더와 같은 절대 센서가 있는 경우 절대 위상 조정 방법을 구현하여 컨트롤러에 저장할 수 있습니다. 이에 대해서는 이후 섹션에서 설명합니다.
- 초기 모터 위상 조정은 커플링되지 않은 모터에서 수행하는 것이 가장 좋습니다.

● 수동 위상 조정 방법

수동 위상 조정 방법은 전류를 `Motor[x].IbBias` 오프셋에 강제 적용하여 정류 전기 사이클(일반적으로 PMAC의 위상 B)의 0 위치에 모터를 단단히 잠그는 것입니다. 모터가 단단히 잠길수록 위상 기준이 향상됩니다. 이 절차는 일반적으로 IDE의 터미널 창에서 아래에 설명된 단계를 사용하여 수행됩니다.

단계	명령	설명
1	<code>#xK</code>	모터가 정지하고 멈추었는지 확인하십시오.
2	<code>Motor[x].IbBias = Motor[x].I2tSet/4</code>	전류량(토크) $[Motor[x].I2tSet/4 - Motor[x].I2tSet]$
3	<code>#xOUT0</code>	증폭기 활성화(모터가 이동함) 모터가 안정될 때까지 대기(이동 중지)
4	<code>Motor[x].PhasePos = 0</code>	PMAC에 이것이 제로 위상 위치임을 알립니다.
5	<code>Motor[x].PhaseFound = 1</code>	상태 비트 설정
6	<code>Motor[x].IbBias = 0 #xK</code>	<code>Motor[x].IbBias</code> 재설정 및 모터 정지

x는 모터 번호를 나타냅니다.



추가 정보

- 이 절차 중에 모터가 진동할 수 있으며 `Motor[x].IbBias`를 줄일 수 있습니다.
또는 안전하고 가능하면 손으로 모터를 안정화할 수 있습니다.
- 모터 어셈블리에 절대 센서가 없는 경우 주문형 위상 조정을 위해 PLC 프로그램에서 수동 위상 조정 방법을 구현(및 저장)할 수 있습니다.
- 개방형 루프 테스트(아래에서 자세히 설명)를 통해 이 방법의 성공 여부를 확인합니다.

● 자동 스테퍼 모터 위상 조정 방법

자동 스테퍼 위상 조정 방법은 Power PMAC 펌웨어에 내장된 두 가지 위치 기준 루틴 중 하나입니다. 다음 구조 요소의 설정이 필요합니다.

구조 요소	설명 및 예
Motor[x].PhaseFindingDac	위상 조정 검색 출력(전류/토크) 크기 일반적인 범위, [Motor[x].I2tSet/4 ~ Motor[x].I2tSet/2]
Motor[x].PhaseFindingTime	위상 조정 검색 시간 = {Time in msec} / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1))
Motor[x].AbsPhasePosOffset	유효한 위상 조정 검색 이동을 한정하는 최소 모션 일반적으로, = 2048 / 5

이러한 설정이 PMAC에 다운로드되면 터미널 또는 PLC 프로그램에서 위상 조정 검색 이동을 시작할 수 있습니다.

- 터미널 창에서: #x\$(여기서 x는 모터 번호)
- PLC 또는 서브루틴에서: Motor[x].PhaseFindingStep = 1

위상 조정 검색 이동이 성공하면 Motor[x].PhaseFound 상태 비트가 true로 설정됩니다.

위상 조정 검색 이동이 성공하지 못한 경우 다음 팁 중 하나를 시도하십시오.

- Motor[x].PhaseFindingDac를 높이십시오.
- Motor[x].PhaseFindingTime을 높이십시오.
- 리버스 인코더 디코드를 수행합니다(증분형 인코더의 경우).
Gate3[i].Chan[j].EncCtrl = 7 <-> 3
- 직렬 인코더의 경우 카운트 방향을 바꾸는 것은 간단하지 않습니다. 정류 방향을 반대로 하려면 Motor[x].PwmSf 및 Motor[x].PhaseOffset을 동시에 반대 부호로 설정해 봅니다.
- 두 개의 모터 리드를 교환하십시오.
- 모터를 부하에서 분리하고 다시 시도해 보십시오.

Status			
Motor Status		Coordinate Status	
Motor	1	● Motor activated	
Description	Status	Description	Status
AmpFna	False	I2tFault	False
AmpFault	False	InPos	False
AmpWarn	False	InterlockStop	False
AuxFault	False	LimitStop	False
BiDir	Plus	MinusLimit	False
BlockRequest	False	PhaseFound	True
ClosedLoop	False	PlusLimit	False
Csolve	False	SoftLimit	False
DacLimit	False	SoftLimitDir	Plus
DesVelZero	False	SoftMinusLimit	False
EncLoss	False	SoftPlusLimit	False
FeFatal	False	SpindleMotor	False
FeWarn	False	TraceCount	0
GantryHomed	False	TriggerMove	False
HomeComplete	False	TriggerNotFound	False
HomeInProgress	False	TriggerSpeedSel	MaxSpeed

개방 루프 테스트



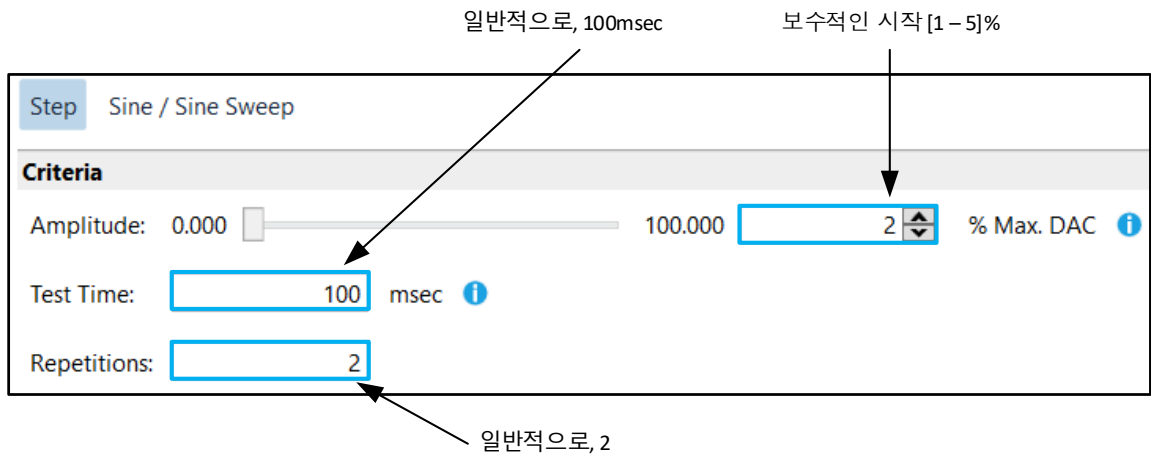
안전한 사용을 위한 주의사항

- 개방 루프 테스트를 통과하지 못한 모터의 위치 루프를 닫으려 하지 마십시오. 이로 인해 위험한 폭주 상황이 발생할 수 있습니다.
- 개방 루프 테스트는 모터를 움직입니다. 주의해야 합니다.

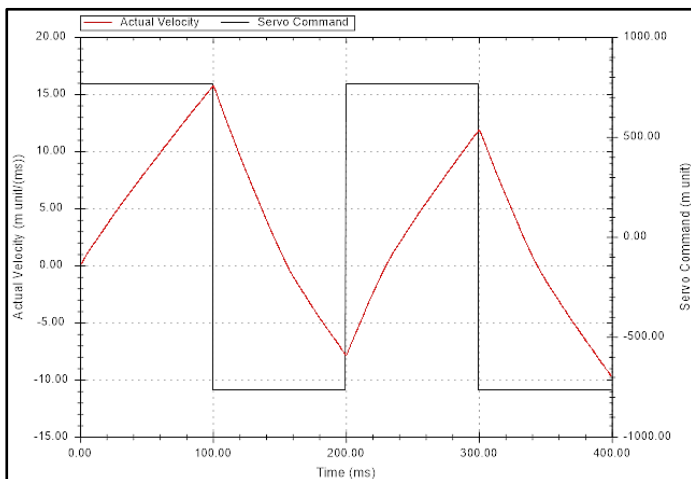
개방 루프 테스트는 정류 "브러시리스" 모터 설정에서 중요한 단계입니다. 주로 다음 기능에 대한 확인을 제공합니다.

- 초기 위상 조정 성공
- 증폭기 출력은 인코더 카운팅 센서와 같은 방향

개방 루프 테스트는 일반적으로 아래 표시된 IDE 튜닝 도구를 사용하여 수행됩니다.



개방 루프 테스트의 목표는 포지티브(개방 루프) 명령이 내려질 때 포지티브(속도) 움직임을 얻는 것이며, 그 반대도 마찬가지입니다. 예상되는 "합격" 결과는 다음과 같은 "톱니" 모양의 그래프입니다.



● 제한된 이동량

일부 모터는 이동이 제한되어 개방 루프 테스트를 수행하기 어려울 수 있습니다. 이 경우 IDE의 터미널 창에서 수동 개방 루프 테스트를 수행할 수 있습니다.

- 항상 #xK kill Motor x (서보 꺼짐)를 기억하십시오.
- #xOUT1, 모터가 정방향으로 움직이고 있습니까?
- #xOUT-1, 모터가 역방향으로 움직이고 있습니까?

이는 IDE에서 모터 속도 창을 보면 확인할 수 있습니다. 백분을 출력은 필요에 따라 증가할 수 있습니다.

● 개방 루프 테스트 문제 해결

일반적으로 개방 루프 테스트는 다음 두 가지 방법 중 하나에서 실패할 수 있습니다.

- 모터가 움직이지 않음(잠긴 느낌)
- 모터가 개방 루프 명령의 반대 방향으로 움직입니다(응답은 "톱니" 모양이 뒤집어져 나타남).

이는 다음 원인 중 하나 이상 있음을 나타냅니다. 하나의 항목이 변경된 경우, 다른 개방 루프 테스트를 시도하기 전에 모터를 다시 위상 조정해야 합니다.

- 잘못된 정류 사이클 크기, Motor[x].PhasePosSf 설정
- 초기 위상 조정이 성공적으로 수행되지 않았음
- 증분형 인코더 방향이 반대임, Gate3[i].Chan[j].EncCtrl = 7 <-> 3
(직렬 인코더에는 적용되지 않음)
- 직렬 인코더 방향이 증폭기 출력과 일치하지 않습니다. 이 경우 모터 리드 두 개를 교체하여 정류 카운팅 방향을 반대로 하는 것이 좋습니다. 또는 Motor[x].PwmSf 및 Motor[x].PhaseOffset을 동시에 반대 부호로 설정합니다.

위치 루프(기본) 튜닝



안전한 사용을 위한 주의사항

개방 루프 테스트를 통과하기 전에는 위치 루프 튜닝을 시도하지 마십시오. 이로 인해 위험한 폭주 상황이 발생할 수 있습니다.

기본 자동 튜닝 도구를 사용하여 초기 계인을 얻을 수 있습니다.

보수적인 시작 [10–20]Hz →

보수적인 시작 [0.5–0.7] →

보수적인 시작 [0.1–0.3] →

Auto Interactive

Basic Advanced

선형 모터의 경우 회전, 인치 또는 밀리미터에 대한 회전당 원시 인코더 수입니다

Criteria

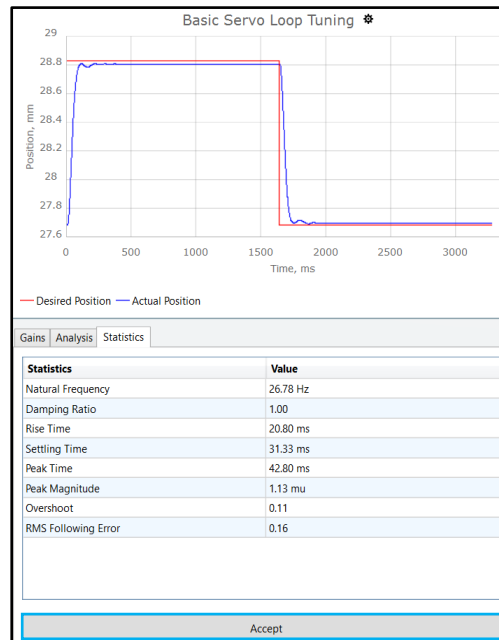
Feedback Resolution: 2000 cts/rev ⓘ

Bandwidth: 0.001 100.000 15 Hz

Damping Ratio: 0.000 1.000 0.7

Integral Ratio: 0.000 1.000 0.1

자동 튜닝이 성공하면 스텝 응답 및 통계를 보여줍니다.



자동 튜닝 도구에 의해 생성된 계인을 수락하거나 가장 일반적인 스텝 및 포물선 튜닝 이동을 사용하여 대화형 미세 튜닝을 수행할 수 있습니다.

Auto Interactive

Move Profile

Step

Parabolic

Point-to-point

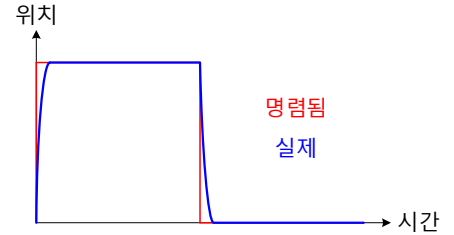
Sine / Sine Sweep

● 스텝 이동(기본) 튜닝

스텝 이동 튜닝의 목표는 명령된 "스텝" 위치 위에 실제 위치를 "중첩"하는 것입니다.

즉, 다음과 같은 구조 요소 게인을 최적화함으로써 얻을 수 있습니다.

- Motor[x].Servo.Kp
- Motor[x].Servo.Kvfb



스텝 이동 튜닝 준비:

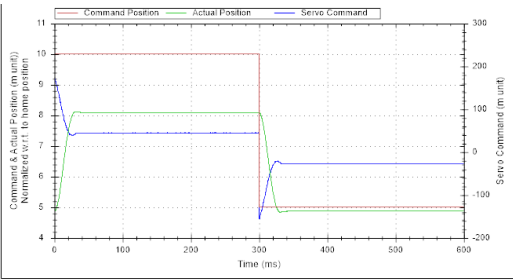
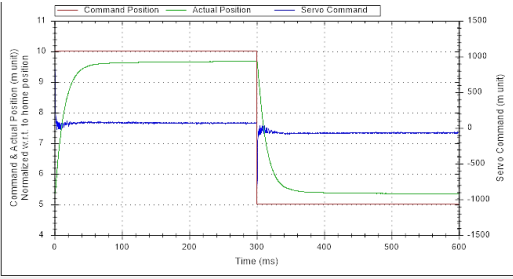
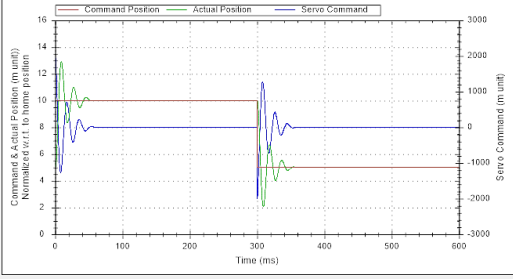
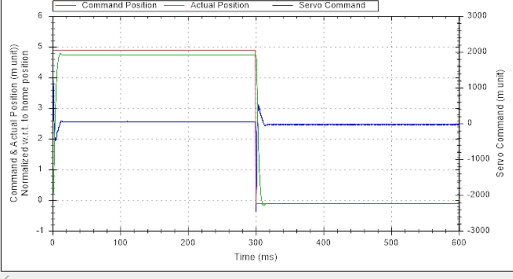
- 다른 모든 게인이 0으로 설정되어 있는지 확인
- (부하를 이동하는) 이동 크기 선택
로터리 모터의 일반적인 범위 [1 ~ 5도]
리니어 모터의 일반적인 범위 [0.1 ~ 0.5mm]
- 이동 시간 선택, 일반 범위 [300 ~ 2000msec]
- 치명적인 후속 오류는 스텝 이동보다 커야 합니다. 스텝 이동 튜닝 연습을 위해 일시적으로 약간 증가할 수 있습니다.
- 서보 출력 제한(Motor[x].MaxDac)을 옆두에 두고 스텝 이동 응답의 과도 부분에서 서보 "effort" 명령과 비교하십시오.



추가 정보

처음에는 (단일) 스텝 이동을 시도하십시오. 안전하고 안정적인 경우 Live Tune 기능을 사용한 다음, 플롯에서 응답 변화를 모니터링하면서 즉석에서 게인을 변경하십시오.

다음은 스텝 이동 튜닝에 대한 일반적인 지침입니다.

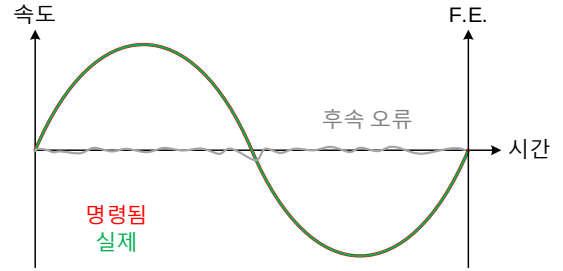
계단 응답	관찰	대책																		
<table><tr><th colspan="2">Stats Decimal Places 2</th></tr><tr><td>Natural Frequency</td><td>23.70 Hz</td></tr><tr><td>Damping Ratio</td><td>0.82</td></tr><tr><td>Settling Time</td><td>24.83 ms</td></tr><tr><td>Peak Time</td><td>30.60 ms</td></tr><tr><td>Overshoot</td><td>1.20%</td></tr><tr><td>Rise Time</td><td>17.20 ms</td></tr><tr><td>Peak Magnitude</td><td>3.24 mu</td></tr><tr><td>RMS Following Error</td><td>1.57</td></tr></table>	Stats Decimal Places 2		Natural Frequency	23.70 Hz	Damping Ratio	0.82	Settling Time	24.83 ms	Peak Time	30.60 ms	Overshoot	1.20%	Rise Time	17.20 ms	Peak Magnitude	3.24 mu	RMS Following Error	1.57	• "느린" 응답. 모터가 헐거워질 수 있습니다. • 서보 "effort" 명령이 너무 적습니다. 이 예에서는 Motor[x].MaxDac의 약 1%입니다. • 상대적으로 긴 정착 시간	Motor[x].Servo.Kp 및 Motor[x].Servo.Kvfb를 비례적으로 높이십시오.
Stats Decimal Places 2																				
Natural Frequency	23.70 Hz																			
Damping Ratio	0.82																			
Settling Time	24.83 ms																			
Peak Time	30.60 ms																			
Overshoot	1.20%																			
Rise Time	17.20 ms																			
Peak Magnitude	3.24 mu																			
RMS Following Error	1.57																			
<table><tr><th colspan="2">Stats Decimal Places 2</th></tr><tr><td>Natural Frequency</td><td>18.69 Hz</td></tr><tr><td>Damping Ratio</td><td>1.00</td></tr><tr><td>Settling Time</td><td>43.14 ms</td></tr><tr><td>Peak Time</td><td>278.60 ms</td></tr><tr><td>Overshoot</td><td>0.05%</td></tr><tr><td>Rise Time</td><td>29.80 ms</td></tr><tr><td>Peak Magnitude</td><td>4.33 mu</td></tr><tr><td>RMS Following Error</td><td>0.88</td></tr></table>	Stats Decimal Places 2		Natural Frequency	18.69 Hz	Damping Ratio	1.00	Settling Time	43.14 ms	Peak Time	278.60 ms	Overshoot	0.05%	Rise Time	29.80 ms	Peak Magnitude	4.33 mu	RMS Following Error	0.88	• "과도한 감쇠" 응답	Motor[x].Servo.Kp를 높이고 Motor[x].Servo.Kvfb를 낮추십시오.
Stats Decimal Places 2																				
Natural Frequency	18.69 Hz																			
Damping Ratio	1.00																			
Settling Time	43.14 ms																			
Peak Time	278.60 ms																			
Overshoot	0.05%																			
Rise Time	29.80 ms																			
Peak Magnitude	4.33 mu																			
RMS Following Error	0.88																			
<table><tr><th colspan="2">Stats Decimal Places 2</th></tr><tr><td>Natural Frequency</td><td>59.56 Hz</td></tr><tr><td>Damping Ratio</td><td>0.17</td></tr><tr><td>Settling Time</td><td>39.92 ms</td></tr><tr><td>Peak Time</td><td>9.20 ms</td></tr><tr><td>Overshoot</td><td>58.71%</td></tr><tr><td>Rise Time</td><td>3.20 ms</td></tr><tr><td>Peak Magnitude</td><td>7.93 mu</td></tr><tr><td>RMS Following Error</td><td>0.65</td></tr></table>	Stats Decimal Places 2		Natural Frequency	59.56 Hz	Damping Ratio	0.17	Settling Time	39.92 ms	Peak Time	9.20 ms	Overshoot	58.71%	Rise Time	3.20 ms	Peak Magnitude	7.93 mu	RMS Following Error	0.65	• "부족한 감쇠" 응답 • 오버슈트가 너무 많습니다.	Motor[x].Servo.Kvfb를 높이십시오.
Stats Decimal Places 2																				
Natural Frequency	59.56 Hz																			
Damping Ratio	0.17																			
Settling Time	39.92 ms																			
Peak Time	9.20 ms																			
Overshoot	58.71%																			
Rise Time	3.20 ms																			
Peak Magnitude	7.93 mu																			
RMS Following Error	0.65																			
<table><tr><th colspan="2">Stats Decimal Places 2</th></tr><tr><td>Natural Frequency</td><td>99.47 Hz</td></tr><tr><td>Damping Ratio</td><td>1.00</td></tr><tr><td>Settling Time</td><td>8.06 ms</td></tr><tr><td>Peak Time</td><td>11.60 ms</td></tr><tr><td>Overshoot</td><td>0.89%</td></tr><tr><td>Rise Time</td><td>5.60 ms</td></tr><tr><td>Peak Magnitude</td><td>4.89 mu</td></tr><tr><td>RMS Following Error</td><td>0.50</td></tr></table>	Stats Decimal Places 2		Natural Frequency	99.47 Hz	Damping Ratio	1.00	Settling Time	8.06 ms	Peak Time	11.60 ms	Overshoot	0.89%	Rise Time	5.60 ms	Peak Magnitude	4.89 mu	RMS Following Error	0.50	• "심각한 감쇠" 응답 • 결합된 모터의 경우 Motor[x].MaxDac가 80%까지 달성되어야 합니다. • 빠른 정착 시간(<10msec) • 오버슈트가 거의 없거나 전혀 없음(<1%) • 높은 고유 주파수(~100Hz)	양호한 스텝 이동 응답
Stats Decimal Places 2																				
Natural Frequency	99.47 Hz																			
Damping Ratio	1.00																			
Settling Time	8.06 ms																			
Peak Time	11.60 ms																			
Overshoot	0.89%																			
Rise Time	5.60 ms																			
Peak Magnitude	4.89 mu																			
RMS Following Error	0.50																			

● 포물선 이동(기본) 튜닝

포물선 이동 조정의 목표는 명령된 "포물선" 위치 위에 실제 속도를 최소 및 0 부근을 중심으로 한 후속 오류로 "중첩"하는 것입니다.

즉, 다음과 같은 구조 요소 계인을 최적화함으로써 얻을 수 있습니다.

- Motor[x].Servo.Kvff
- Motor[x].Servo.Kaff
- Motor[x].Servo.Ki



포물선 이동 튜닝 준비:

- Motor[x].Servo.Kvfb를 Motor[x].Servo.Kvff로 복사합니다(이 두 계인은 거의 항상 동일함).
- (부하를 이동하는) 이동 크기 선택
로터리 모터의 일반적인 범위 [1 ~ 4회전]
리니어 모터의 일반적인 범위 [10 ~ 80mm]
- 이동 시간 선택, 일반 범위 [600 ~ 2000msec]
- 일반적으로 예상되는 모터 최대 속도의 20~40%까지 달성하는 것이 바람직합니다.

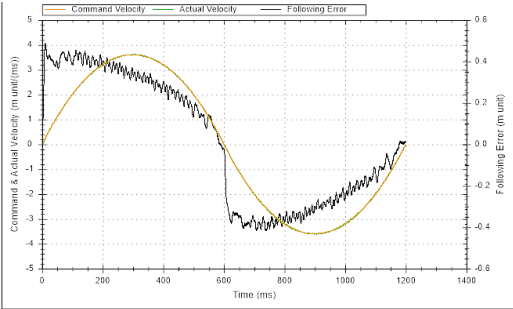
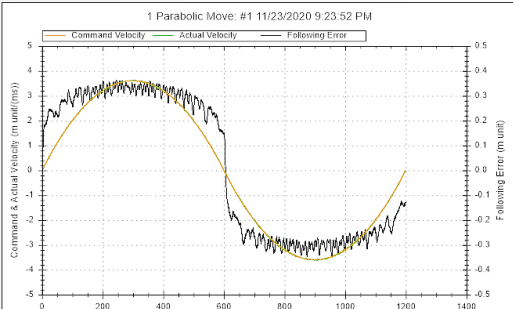
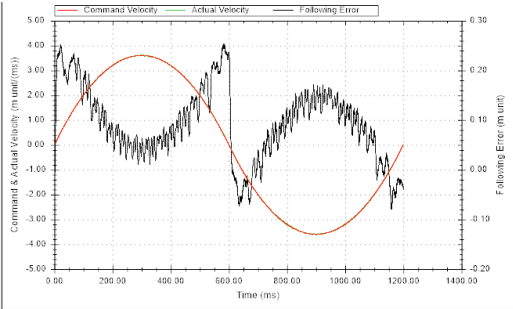
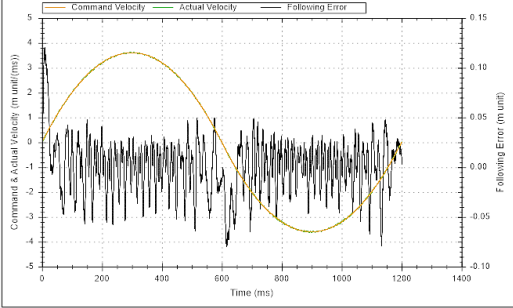
Step	Parabolic	Point-to-point	Sine / Sine Sweep
Move Size:	1080	deg	
Move Time:	600	msec	
Dwell Time:	500	msec	☐
Kill Motor After Move			
Gains Servo Loop Filters Trajectory Prefilter			
Gains			
Proportional:	500	Kp	
Derivative 1:	9300	Kvfb	
Derivative 2:	0	Kvifb	
Integral:	0.016000001	Ki	
Velocity Feedforward 1:	9300	Kvff	
Velocity Feedforward 2:	0	Kviff	
Acceleration Feedforward Gain:	85000	Kaff	
Friction Feedforward Gain:	0	Kfff	
Other Servo Settings			
Integrator Mode:	0	SwZvInt	
Fatal Following Error Limit:	5	FatalFELimit	
Servo Output Limit:	25505.205	MaxDac	
			Single Move Live Tune



추가 정보

처음에는 (단일) 포물선 이동을 시도하십시오. 안전하고 안정적인 경우 Live Tune 기능(PA-33)을 사용한 다음, 플롯의 변화를 모니터링하면서 즉석에서 계인을 변경하십시오.

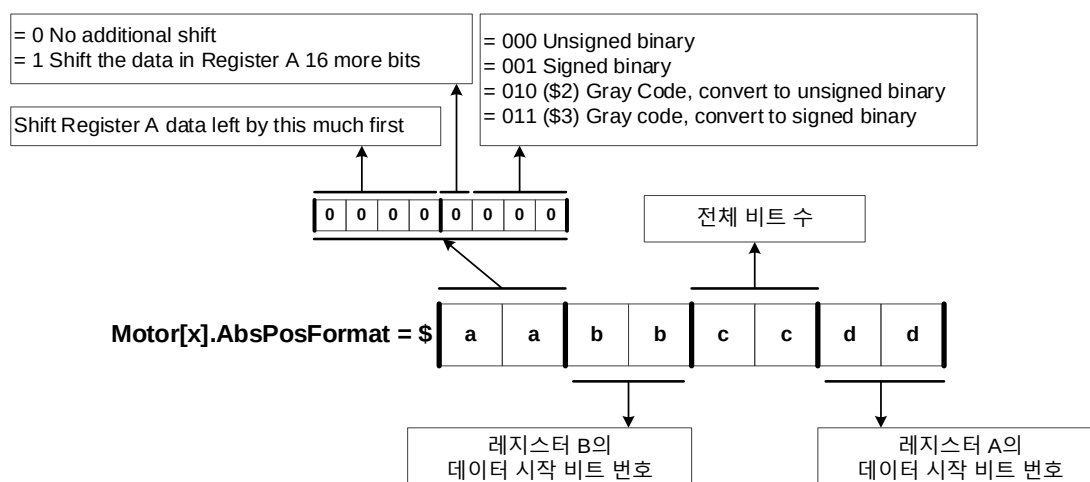
Motor[x].Servo.Ki보다 먼저 Motor[x].Servo.Kaff를 최적화하는 것이 좋습니다. 다음은 포물선 이동 튜닝에 대한 일반적인 지침입니다.

계단 응답	관찰	대책								
 Stats Decimal Places 4 <table><tr><td>Vel. Corr</td><td>0.9086</td><td>Max Fe</td><td>0.4892</td></tr><tr><td>Acc. Corr</td><td>0.3274</td><td>RMS Following Error</td><td>0.3076</td></tr></table>	Vel. Corr	0.9086	Max Fe	0.4892	Acc. Corr	0.3274	RMS Following Error	0.3076	높은 가속 상관 관계(후속 오류는 V 모양을 띠)	Motor[x].Servo.Kaff를 높이십시오.
Vel. Corr	0.9086	Max Fe	0.4892							
Acc. Corr	0.3274	RMS Following Error	0.3076							
 1 Parabolic Move: #1 11/23/2020 9:23:52 PM Stats Decimal Places 4 <table><tr><td>Vel. Corr</td><td>0.9651</td><td>Max Fe</td><td>0.3628</td></tr><tr><td>Acc. Corr</td><td>0.0075</td><td>RMS Following Error</td><td>0.2821</td></tr></table>	Vel. Corr	0.9651	Max Fe	0.3628	Acc. Corr	0.0075	RMS Following Error	0.2821	- 높은 속도 상관 관계(후속 오류는 계단 모양을 띠) - 이동 및 정착(지점 간) 애플리케이션의 경우 Motor[s].Servo.Kvff를 조정할 필요가 없으며 항상 Motor[x].Servo.Kvfb와 동일하게 유지하십시오.	Motor[x].Servo.Kvff를 높이십시오.
Vel. Corr	0.9651	Max Fe	0.3628							
Acc. Corr	0.0075	RMS Following Error	0.2821							
 Stats Decimal Places 4 <table><tr><td>Vel. Corr</td><td>0.0238</td><td>Max Fe</td><td>0.2528</td></tr><tr><td>Acc. Corr</td><td>0.0387</td><td>RMS Following Error</td><td>0.1184</td></tr></table>	Vel. Corr	0.0238	Max Fe	0.2528	Acc. Corr	0.0387	RMS Following Error	0.1184	- 가속 및 속도 상관관계가 거의 없거나 전혀 없음 - 후속 오류 스케일 확인 0 부근을 중심으로 한 후속 오류 형태	Motor[x].Servo.Ki를 높이십시오.
Vel. Corr	0.0238	Max Fe	0.2528							
Acc. Corr	0.0387	RMS Following Error	0.1184							
 Stats Decimal Places 4 <table><tr><td>Vel. Corr</td><td>0.0024</td><td>Max Fe</td><td>0.1204</td></tr><tr><td>Acc. Corr</td><td>0.2278</td><td>RMS Following Error</td><td>0.0278</td></tr></table>	Vel. Corr	0.0024	Max Fe	0.1204	Acc. Corr	0.2278	RMS Following Error	0.0278	0 부근에 최적화된 후속 오류 형태 - 플롯 중간에 후속 오류(방향 변경)의 급증에 대해 소량의 Motor[x].Servo.Kfff를 추가하는 것이 가능합니다(윤곽선 애플리케이션의 경우).	좋은 포물선 이동 응답
Vel. Corr	0.0024	Max Fe	0.1204							
Acc. Corr	0.2278	RMS Following Error	0.0278							

절대 위치 설정(절대 호밍)

이 섹션에서는 절대 직렬 인코더에 대한 "절대" 위치 읽기를 설정하기 위한 기본 작업을 설명합니다. 이 기능을 설정하는 데 필요한 요소는 다음과 같습니다.

구조 요소	설명 및 예
Motor[x].pAbsPos	전원 켜기 절대 위치 포인터 일반적으로, Gate3의 경우, = Gate3[i].Chan[j].SerialEncDataA.a 일반적으로, ACC-84E의 경우, = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPosSf	절대 위치 읽기 환산 계수 = Motor[x].PosSf
Motor[x].AbsPosFormat	전원 켜기 절대 데이터 형식 아래의 설명 및 예(자세한 내용)를 참조하십시오.
Motor[x].HomeOffset	포지셔닝 기준 오프셋 원하는 0 위치를 나타내기 위해 사용자가 설정(모터 단위)



이러한 설정은 PMAC에 다운로드됩니다. 이 기능은 다음과 같이 사용할 수 있습니다.

- 터미널 창에서: #xHMZ는 모터 x의 절대 위치를 판독합니다.
- 프로그램에서(PLC 또는 모션): HOMEZ x는 모터 x의 절대 위치를 판독합니다.
- 또는 Motor[x].PowerOnMode(= \$2)의 비트 #1이 전원을 켤 때 읽은 절대 위치를 지정합니다. 전원을 켤 때 인코더 전원을 사용할 수 있는지 확인하십시오(전원이 축 카드 외부에 있는 경우).



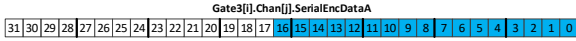
추가 정보

회색 코드 변환은 이미 직렬 인코더 명령어 Gate3[i].Chan[j].SerialEncCmd 또는 ACC84E[i].Chan[j].SerialEncCmd에서 구현된 경우 여기에서 생략해야 합니다.

예

직렬 데이터 A 레지스터의 하위 필드에 위치한 17비트 이진 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더(다중 회전 없음).

• Gate3

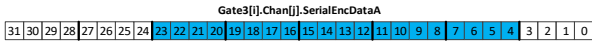


• ACC-84E

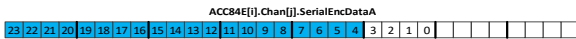


직렬 데이터 A 레지스터의 비트 #4에서 시작하는 20비트 이진 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더(다중 회전 없음).

• Gate3

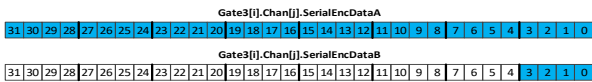


• ACC-84E



직렬 데이터 A 및 B 레지스터에 연속적으로 위치한 36비트 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더.

• Gate3

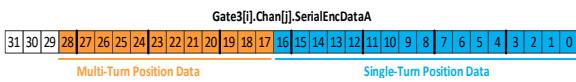


• ACC-84E

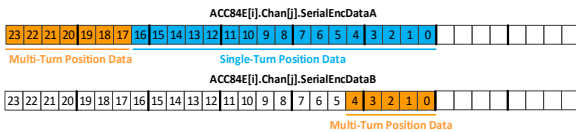


직렬 데이터 A 레지스터의 비트 #0에서 시작하여 비트 #28까지 연속적으로 확장되는 17비트의 단일 회전 위치 데이터와 12비트의 다중 회전 위치 데이터가 있는 29비트 이진 절대 직렬 인코더.

• Gate3

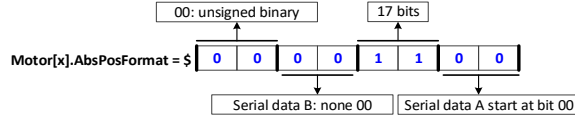


• ACC-84E

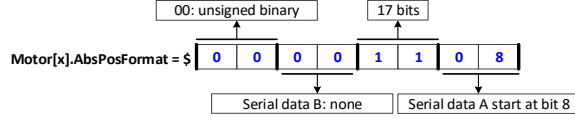


Motor[x].AbsPosFormat 설정

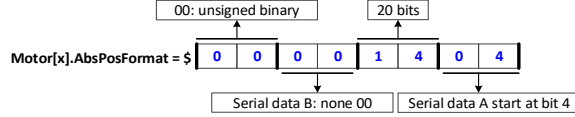
• Gate3



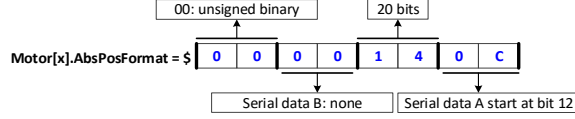
• ACC-84E



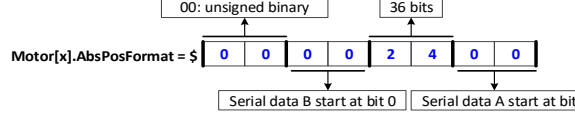
• Gate3



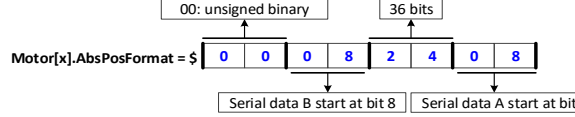
• ACC-84E



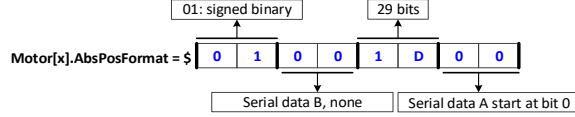
• Gate3



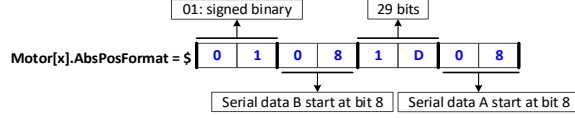
• ACC-84E



• Gate3



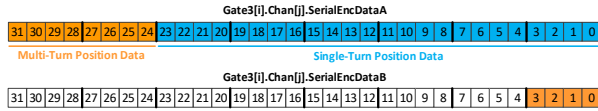
• ACC-84E



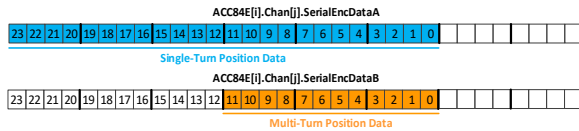
예

24비트의 단일 회전 위치 데이터와 12비트의 다중 회전 위치 데이터가 있는 36비트 이진 절대 직렬 인코더.

• Gate3

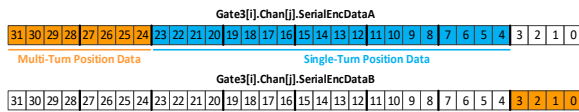


• ACC-84E

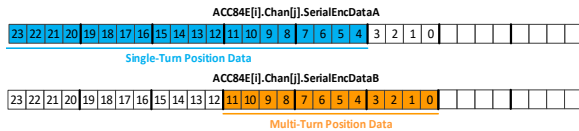


20비트의 단일 회전 위치 데이터(비트 #4에서 시작)와 12비트의 다중 회전 위치 데이터가 있는 32비트 회색 코드 절대 직렬 인코더.

• Gate3



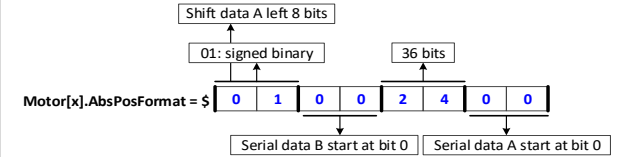
• ACC-84E



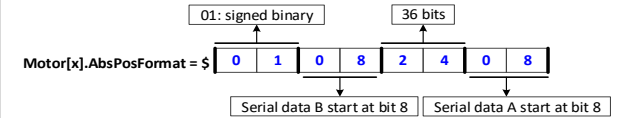
참고 이미 전역 명령어에 회색 코드가 설정되어 있다면 여기서 설정하면 안 됩니다.

Motor[x].AbsPosFormat 설정

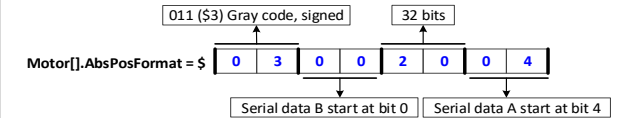
• Gate3



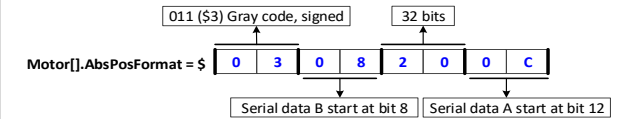
• ACC-84E



• Gate3



• ACC-84E



추가 정보

동일한 직렬 인코더 데이터 레지스터에서 단일 회전 데이터와 다중 회전 데이터 사이에 간격이 있는 경우(연속되지 않는 경우) 위치 데이터를 수동으로 모아(예: PLC에서) 사용자 메모리 레지스터(예: Sys.Udata[i])에 배치하여 간격을 없애야 합니다. 그런 다음 절대 호밍 구조 요소를 이 레지스터에 적용할 수 있습니다.

절대 위상 기준 설정



올바른 사용을 위한 주의사항

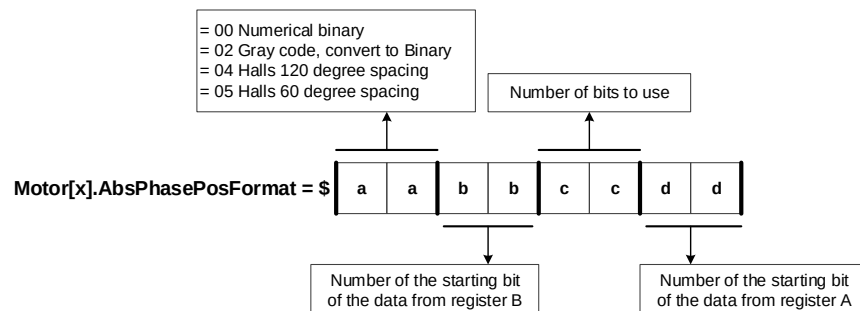
- 모터는 절대 위상 기준을 설정하기 전에 위상 조정되어야 합니다.
- Motor[x].PhaseFindingDac 및 Motor[x].PhaseFindingTime은 0과 같아야 합니다.

이 섹션에서는 홀 효과 센서와 절대 직렬 인코더가 있는 모터의 절대 위상(정류) 기준 설정에 대해 설명합니다. 이 섹션에서는 다음 정의가 사용됩니다.

이름	설명
ST	단일 회전 직렬 로터리 인코더 위치 데이터
NoOfPolePairs	로터리 브러시리스 모터의 쌍극 수
ECLmm	리니어 모터 전기 사이클 길이 또는 자기 피치(mm)(예: 30.48mm)
RESmm	mm 단위의 리니어 인코더 분해능(예: 1 μ m = 0.001mm)

절대 위상 기준을 설정하기 위한 기본 구조 요소는 다음과 같습니다.

구조 요소	설명 및 예
Motor[x].pAbsPhasePos	전원 켜기 절대 위상 위치 포인터 홀의 경우, =Gate3[i].Chan[j].Status.a 절대 직렬 인코더의 경우, =Gate3[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPosSf	절대 위치 읽기 환산 계수 홀의 경우, = $\pm 2048 / 12$ (UVW 전환에 따라 다름) 직렬 절대 로터리 인코더의 경우, = $2048 * \text{NoOfPolePairs} / 2^{\text{ST}}$ 직렬 절대 리니어 인코더의 경우, = $2048 * \text{RESmm} / \text{ECLmm}$
Motor[x].AbsPhasePosFormat	전원 켜기 절대 위상 데이터 형식 홀 간격이 120°인 경우, =\$400030C 절대 직렬 인코더에 대해서는 아래 설명 및 예(자세한 내용)를 참조하십시오.
Motor[x].AbsPhasePosOffset	절대 위상 위치 기준 홀의 경우, = UVW의 1 <-> 3 전환과 교차하는 위상 위치 값 절대 직렬 인코더의 경우, = -Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf



이러한 설정은 PMAC에 다운로드됩니다. 이 기능은 다음과 같이 사용할 수 있습니다.

- 터미널 창에서: #x\$는 모터 x의 절대 위치를 판독합니다.
- PLC 프로그램에서: Motor[x].PhaseFindingStep = 1은 모터 x의 절대 위치를 판독합니다.
- 또는 Motor[x].PowerOnMode(= \$4)의 비트 #2는 전원을 켤 때 절대 위상 기준을 지정합니다. 전원을 켤 때 인코더 전원을 사용할 수 있는지 확인하십시오(전원이 축 카드 외부에 있는 경우).

● 홀 효과 센서를 사용한 절대 위상 기준

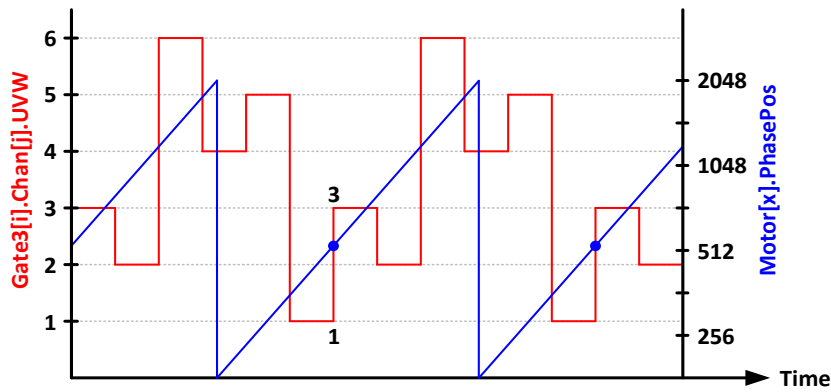
이 섹션에서는 기존의 120° 홀 효과 센서 간격을 사용하는 절대 위상 기준 설정에 대해 설명합니다.



올바른 사용을 위한 주의사항

- 모터는 절대 위상 기준을 설정하기 전에 위상 조정되어야 합니다.
- Motor[x].PhaseFindingDac 및 Motor[x].PhaseFindingTime은 0과 같아야 합니다.

모터를 **정방향으로** 이동하면서(일반적으로 느림) IDE 플롯을 사용하여 Gate3[i].Chan[j].UVW 및 Motor[x].PhasePos 구조 요소를 수집합니다.



이 플롯은 두 가지 중요한 홀 설정을 나타냅니다.

- $\text{Motor}[x].\text{PhasePosSf} = 2048 / 12$ (UVW이 1에서 3으로 전환되는 경우)
 $= -2048 / 12$ (UVW이 3에서 1로 전환되는 경우)
- 1<-> 3 전환 시 $\text{Motor}[x].\text{AbsPhasePosOffset} = \{\text{value of Motor}[x].\text{PhasePos}\}$

● 절대 인코더를 사용한 절대 위상 기준



올바른 사용을 위한 주의사항

- 모터는 절대 위상 기준을 설정하기 전에 위상 조정되어야 합니다.
- Motor[x].PhaseFindingDac 및 Motor[x].PhaseFindingTime은 0과 같아야 합니다.

절대 위상 기준에 사용할 수 있는 최대 비트 수는 32입니다. 전체 위치 데이터 스트림을 사용할 필요는 없습니다. 단순하게 하기 위해 단일 회전 데이터만 사용하는 것이 좋습니다. 그러나 한 가지 중요한 점은 사용할 위치 데이터의 MSB(최상위 비트)에 항상 포함시켜야 한다는 것입니다.

Motor[x].AbsPhasePosForce는 홀딩 레지스터일 뿐입니다. 이 값은 다음 절대 위상 기준 테스트 절차를 수행하여 확인할 수 있습니다.

단계	명령	설명
1	#xK	모터가 정지하고 멈추었는지 확인하십시오.
2	Motor[x].IbBias = Motor[x].I2tSet/4	전류량(토크) [Motor[x].I2tSet/4 - Motor[x].I2tSet]
3	#xOUT0	증폭기 활성화(모터가 이동함) 모터가 안정될 때까지 대기(이동 중지)
4	Read Motor[x].AbsPhasePosForce	={value}, 아래 예를 참조하십시오.
6	Motor[x].IbBias = 0 #xK	Motor[x].IbBias 재설정 및 모터 정지

x는 모터 번호를 나타냅니다.



추가 정보

이 절차 중에 모터가 진동할 수 있으며 Motor[x].IbBias를 줄일 수 있습니다. 또는 안전하고 가능하면 손으로 모터를 안정화할 수 있습니다.

예

절대 위상 기준 설정

1

직렬 데이터 A 레지스터의 하위 필드에 위치한 17비트 이진 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더(다중 회전 없음).

• Gate3

Gate3[i].Chan[j].SerialEncDataA

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{NoOfPolePairs} / 2^{17}$
리니어: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{RESmm} / \text{ECLmm}$
 $\text{Motor}[x].\text{pAbsPhasePos} = \text{Gate3}[i].\text{Chan}[j].\text{SerialEncDataA.a}$
 $\text{Motor}[x].\text{AbsPhasePosFormat} = \$00001100$
 $\text{Motor}[x].\text{AbsPhasePosForce} = \text{테스트 중 } \{\text{Gate3}[i].\text{Chan}[j].\text{SerialEncDataA} \& \$1FFFF\}$ 의 값
 $\text{Motor}[x].\text{AbsPhasePosOffset} = -\text{Motor}[x].\text{AbsPhasePosForce} * \text{Motor}[x].\text{AbsPhasePosSf}$

• ACC-84E

ACC84E[i].Chan[j].SerialEncDataA

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{NoOfPolePairs} / 2^{17}$
리니어: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{RESmm} / \text{ECLmm}$
 $\text{Motor}[x].\text{pAbsPhasePos} = \text{ACC84E}[i].\text{Chan}[j].\text{SerialEncDataA.a}$
 $\text{Motor}[x].\text{AbsPhasePosFormat} = \$00001108$
 $\text{Motor}[x].\text{AbsPhasePosForce} = \text{테스트 중 } \{\text{ACC84E}[i].\text{Chan}[j].\text{SerialEncDataA} \& \$1FFFF\}$ 의 값
 $\text{Motor}[x].\text{AbsPhasePosOffset} = -\text{Motor}[x].\text{AbsPhasePosForce} * \text{Motor}[x].\text{AbsPhasePosSf}$

2

직렬 데이터 A 레지스터의 비트 #4에서 시작하는 20비트 이진 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더(다중 회전 없음).

• Gate3

Gate3[i].Chan[j].SerialEncDataA

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{NoOfPolePairs} / 2^{20}$
리니어: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{RESmm} / \text{ECLmm}$
 $\text{Motor}[x].\text{pAbsPhasePos} = \text{Gate3}[i].\text{Chan}[j].\text{SerialEncDataA.a}$
 $\text{Motor}[x].\text{AbsPhasePosFormat} = \$00001404$
 $\text{Motor}[x].\text{AbsPhasePosForce} = \text{테스트 중 } \{(\text{Gate3}[i].\text{Chan}[j].\text{SerialEncDataA} \& \$FFFFFF0) \gg 4\}$ 의 값
 $\text{Motor}[x].\text{AbsPhasePosOffset} = -\text{Motor}[x].\text{AbsPhasePosForce} * \text{Motor}[x].\text{AbsPhasePosSf}$

• ACC-84E

ACC84E[i].Chan[j].SerialEncDataA

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{NoOfPolePairs} / 2^{20}$
리니어: $\text{Motor}[x].\text{AbsPhasePosSf} = 2048 * \text{RESmm} / \text{ECLmm}$
 $\text{Motor}[x].\text{pAbsPhasePos} = \text{ACC84E}[i].\text{Chan}[j].\text{SerialEncDataA.a}$
 $\text{Motor}[x].\text{AbsPhasePosFormat} = \$0000140C$
 $\text{Motor}[x].\text{AbsPhasePosForce} = \text{테스트 중 } \{(\text{ACC84E}[i].\text{Chan}[j].\text{SerialEncDataA} \& \$FFFFFF0) \gg 4\}$ 의 값
 $\text{Motor}[x].\text{AbsPhasePosOffset} = -\text{Motor}[x].\text{AbsPhasePosForce} * \text{Motor}[x].\text{AbsPhasePosSf}$

예

절대 위상 기준 설정

3

직렬 데이터 A 및 B 레지스터에 연속적으로 위치한 36비트 단일 회전(또는 선형 스케일) 위치 데이터가 있는 절대 직렬 인코더.

• Gate3

Gate3[i].Chan[j].SerialEncDataA

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Gate3[i].Chan[j].SerialEncDataB

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2³²
리니어: Motor[x].AbsPhasePosSf = 2048 * RESmm * 2⁴ / ECLmm
Motor[x].pAbsPhasePos = Gate3[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = \$00002004(하위 4비트 무시, MSB를 포함하려면 상위 32비트 사용)
Motor[x].AbsPhasePosForce = 테스트 중
{Gate3[i].Chan[j].SerialEncDataA>>4+(Gate3[i].Chan[j].SerialEncDataB&\$F)<<28}의 값
Motor[x].AbsPhasePosOffset = −Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

• ACC-84E

ACC84E[i].Chan[j].SerialEncDataA

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

ACC84E[i].Chan[j].SerialEncDataB

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

로터리: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2³²
리니어: Motor[x].AbsPhasePosSf = 2048 * RESmm * 2⁴ / ECLmm
Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = \$0008200C(하위 4비트 무시, MSB를 포함하려면 상위 32비트 사용)
Motor[x].AbsPhasePosForce = 테스트 중
{ACC84E[i].Chan[j].SerialEncDataA>>4+(ACC84E[i].Chan[j].SerialEncDataB&\$FFF)<<20}의 값
Motor[x].AbsPhasePosOffset = −Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

4

직렬 데이터 A 레지스터의 비트 #0에서 시작하여 비트 #28까지 연속적으로 확장되는 17비트의 단일 회전 위치 데이터와 12비트의 다중 회전 위치 데이터가 있는 29비트 이진 절대 직렬 인코더.

• Gate3

Gate3[i].Chan[j].SerialEncDataA

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Multi-Turn Position Data

Single-Turn Position Data

Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2¹⁷
Motor[x].pAbsPhasePos = Gate3[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = \$00001100(레지스터 A에서 17비트 단일 회전 데이터만 사용)
Motor[x].AbsPhasePosForce = 테스트 중 {Gate3[i].Chan[j].SerialEncDataA & \$1FFFF}의 값
Motor[x].AbsPhasePosOffset = −Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

• ACC-84E

ACC84E[i].Chan[j].SerialEncDataA

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Multi-Turn Position Data

Single-Turn Position Data

ACC84E[i].Chan[j].SerialEncDataB

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

Multi-Turn Position Data

로터리: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2¹⁷
Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = \$00001108(레지스터 A에서 17비트 단일 회전 데이터만 사용)
Motor[x].AbsPhasePosForce = 테스트 중 {ACC84E[i].Chan[j].SerialEncDataA & \$1FFFF}의 값
Motor[x].AbsPhasePosOffset = −Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

A-2-5 모터 작동 방향 반전



안전한 사용을 위한 주의사항

- 모터에 전원이 공급되는 동안에는 다음과 같은 설정 변경 사항을 적용하면 안 됩니다.
- 작동 방향 변경을 수행한 후에는 모터의 위상을 다시 조정해야 합니다.
- 모터 작동 방향이 변경되면 초과 이동 및 소프트 제한을 적절하게 수정해야 합니다.

종종 특정 애플리케이션에서는 설정이 완료된 후 모터 작동 방향을 반대로 하는 것이 바람직합니다. 브러시리스 모터의 경우 이 변경 사항을 적용하는 데 필요한 구조 요소의 "새로운" 설정이 아래에 나와 있습니다. 인코더 유형에 따라 다릅니다.

디지털 직각위상 및 아날로그 정현파 인코더가 있는 모터

구조 요소	새 값
Motor[x].PhaseOffset	= - 현재 값
Motor[x].PwmSf	= - 현재 값
Gate3[i].Chan[j].EncCtrl	= 3 또는 7(반대 디코딩 값)
Motor[x].AbsPhasePosSf(홀을 사용하는 경우)	= - 현재 값
Motor[x].AbsPhasePosOffset(홀을 사용하는 경우)	= 2048 - 현재 값

직렬 인코더가 있는 모터

구조 요소	새 값
Motor[x].PhaseOffset	= - 현재 값
Motor[x].PwmSf	= - 현재 값
Motor[x].PhasePosSf	= - 현재 값
EncTable[n].ScaleFactor	= - 현재 값
Motor[x].AbsPosSf(절대 인코더의 경우)	= - 현재 값
Motor[x].AbsPhasePosSf(절대 인코더의 경우)	= - 현재 값
Motor[x].AbsPhasePosOffset(절대 인코더의 경우)	= 2048 - 현재 값

A-3 Gate3 "스크립트" 모터 설정 샘플

이 섹션에서는 Power PMAC ACC-24E3, ACC-84E 및 CK3M AX 장치가 있는 브러시리스 모터 설정 샘플이 참조용으로 제공됩니다.



안전한 사용을 위한 주의사항

이 섹션에서 제공되는 샘플은 참조용일 뿐이며 모터가 즉시 작동하지 않을 수 있습니다. 다음 항목에 주의해야 합니다.

- 인코더 및 모터 사양 매개변수
- 전류 루프 튜닝을 수행해야 함
- 초기 위상 조정을 수행해야 함
- 개방 루프 테스트를 수행해야 함
- 위치 루프 튜닝을 수행해야 함
- 절대 위치 설정(해당되는 경우)을 검토해야 함
- 절대 위상 기준 설정(해당되는 경우)을 검토해야 함
- 일반 모션 매개변수(예: 조그 속도)를 검토해야 함



추가 정보

쉽게 설명하기 위해 이 섹션의 샘플은 모터 #1 및 Gate3, 인덱스 0, 채널 0을 참조합니다.

A-3-1 직각위상 인코더가 있는 로터리 서보 모터

하드웨어 구성

항목	참고
컨트롤러	이 샘플은 UMAC ACC-24E3 및 CK3M AX에 적용됩니다.
증폭기	CK3A-G310L(3상 208VAC 주 공급 전압)
모터	로터리 서보 모터, 200VAC, 11/23A, 4극 쌍
인코더	디지털 직각위상, 2000라인(x4 디코딩으로 회전당 8,000카운트)

시스템 게이트

Sys.WpKey = \$AAAAAAA	DISABLE WRITE-PROTECTION
Sys.MaxCoords = 2	MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5	MAX. NO. OF MOTORS+1 (EFFICIENCY)
Gate3[0].PhaseFreq = 10000	10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1	5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1	2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq	TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)	TYPICAL CALCULATION
Gate3[0].Chan[0].PwmFreqMult = 1	CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1	
Gate3[0].Chan[2].PwmFreqMult = 1	
Gate3[0].Chan[3].PwmFreqMult = 1	
Gate3[0].Chan[0].PackInData = 0	CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0	MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0	
Gate3[0].Chan[3].PackInData = 0	
Gate3[0].Chan[0].PackOutData = 0	CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0	MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0	
Gate3[0].Chan[3].PackOutData = 0	
Gate3[0].AdcAmpStrobe = \$901001	CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5	CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4	CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533	CH1 MIN. 3μSEC FOR G310

ECT(인코더 변환표)

EncTable[1].Type = 1	TYPICAL QUADRATURE ENCODER ENTRY
EncTable[1].pEnc = Gate3[0].Chan[0].ServoCapt.a	
EncTable[1].pEnc1 = Sys.Pushm	
EncTable[1].index1 = 0	
EncTable[1].index2 = 0	
EncTable[1].index3 = 0	
EncTable[1].index4 = 0	
EncTable[1].index5 = 0	
EncTable[1].index6 = 0	
EncTable[1].ScaleFactor = 1 / 256	
EncTable[1].MaxDelta = 0	
EncTable[2].Type = 0	END OF ECT (FOR EFFICIENCY)

모터 설정 및 시운전

GLOBAL Ck3a1MaxAdc = 31.470	CK3A MAX ADC
GLOBAL Mtr1CtsPerRev = 8000	COUNTS PER REVOLUTION
GLOBAL Mtr1PolePairs = 4	NO. OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 20	MAX RMS CUR.(MTR>CK3A)
GLOBAL Mtr1RatedRmsCur = 10	RATED RMS CUR.(MTR>CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2	TIME ALLOWED AT MAX CUR.
Gate3[0].Chan[0].EncCtrl = 7	ENC. COUNTING DIRECTION
Motor[1].ServoCtrl = 1	ACTIVATE CHANNEL
Motor[1].PosUnit = 11	IDE WINDOW UNITS, DEGREES
Motor[1].PosSf = 360 / Mtr1CtsPerRev	SCALE POSITION TO DEGREES
Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev	SCALE VELOCITY TO DEGREES
Motor[1].pLimits = Gate3[0].Chan[0].Status.a	ENABLE LIMITS, =0 TO DISABLE
Motor[1].AbortTa = 150	ABORT DECELERATION, 150 MSEC
Motor[1].AbortTs = 0	ABORT S-CURVE, NONE
Motor[1].JogTa = 300	JOG ACCELERATION, 300 MSEC
Motor[1].JogTs = 0	JOG S-CURVE, NONE
Motor[1].JogSpeed = 4.5	JOG SPEED, 4.5 DEG/MSEC
Motor[1].FatalFeLimit = 5	FE LIMIT, 5 DEGREES
Motor[1].WarnFeLimit = 0	WARNING FE LIMIT, NONE
Motor[1].InPosBand = 0.05	INPOS BAND, 0.05 DEGREES
Motor[1].InPosTime = 20 / Sys.ServoPeriod	INPOS TIME, 20 MSEC
Motor[1].pPhaseEnc = Gate3[0].Chan[0].PhaseCapt.a	TYPICAL FOR QUADRATURE
Motor[1].PhaseCtrl = 4	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 0	NONE FOR QUADRATURE
Motor[1].PhaseEncRightShift = 0	NONE FOR QUADRATURE
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / (256 * Mtr1CtsPerRev)	TYPICAL FOR QUADRATURE
Motor[1].AmpFaultLevel = 1	=1 FOR CK3A
Motor[1].AdcMask = \$FFFF0000	=\$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384	TYPICAL IF MTR VOLTAGE>INPUT
Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	...
Motor[1].I2tTrip = (POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur	...
Motor[1].CurrentNullPeriod = 0	=0 FOR CK3A
Motor[1].IaBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IbBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IiGain = 0	CURRENT LOOP GAINS
Motor[1].IpfGain = 0	(MUST TUNE)
Motor[1].IpbGain = 0	...
Motor[1].Servo.Kp = 0	POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0	(MUST TUNE)
Motor[1].Servo.Kvifb = 0	...
Motor[1].Servo.Kvff = 0	...
Motor[1].Servo.Kviff = 0	...
Motor[1].Servo.Ki = 0	...
Motor[1].Servo.Kaff = 0	...
Motor[1].Servo.Kfff = 0	...
Motor[1].Servo.SwZvInt = 0	...
Motor[1].PhaseFindingTime = 8000 / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1))	STEPPER PHASING (8 SECONDS)
Motor[1].PhaseFindingDac = 0.8 * Motor[1].I2tSet	80% OF I2TSET
Motor[1].AbsPhasePosOffset = 2048 / 5	TYPICAL SETTING
Motor[1].PowerOnMode = 0	NO ABS. PHASE ON POWER-UP

A-3-2 OMRON G5 서보 모터

하드웨어 구성

항목	참고
컨트롤러	이 설정 샘플은 CK3M AX 또는 UMAC ACC-24E3에 적용됩니다.
증폭기	CK3A-G305L(110VAC 주 공급 전압)
모터	G5 서보 모터 R88M-K40030H-S2, 200VAC, 2.4/10.2A, 5극 쌍
인코더	1회전당 G5 20비트 절대값(초기 표시 후)

시스템 게이트

Sys.WpKey = \$AAAAAAA Sys.MaxCoords = 2 Sys.MaxMotors = 5	DISABLE WRITE-PROTECTION MAX. COORD SYSTEMS+1 (EFFICIENCY) MAX. NO. OF MOTORS+1 (EFFICIENCY)
Gate3[0].PhaseFreq = 10000 Gate3[0].ServoClockDiv = 1 Sys.RtIntPeriod = 1 Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)	10 KHZ PHASE FREQUENCY 5 KHZ SERVO FREQUENCY 2.5 KHZ RTI FREQUENCY TYPICAL CALCULATION TYPICAL CALCULATION
Gate3[0].Chan[0].PwmFreqMult = 1 Gate3[0].Chan[1].PwmFreqMult = 1 Gate3[0].Chan[2].PwmFreqMult = 1 Gate3[0].Chan[3].PwmFreqMult = 1	CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[0].PackInData = 0 Gate3[0].Chan[1].PackInData = 0 Gate3[0].Chan[2].PackInData = 0 Gate3[0].Chan[3].PackInData = 0	CH1-4 IN DATA MUST BE 0 FOR CK3A
Gate3[0].Chan[0].PackOutData = 0 Gate3[0].Chan[1].PackOutData = 0 Gate3[0].Chan[2].PackOutData = 0 Gate3[0].Chan[3].PackOutData = 0	CH1-4 OUT DATA MUST BE 0 FOR CK3A
Gate3[0].AdcAmpStrobe = \$901001 Gate3[0].AdcAmpClockDiv = 5 Gate3[0].AdcAmpHeaderBits = 4 Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533	CH1-4 STROBE WORD (FOR CK3A) CH1-4 3.125 MHZ ADC CLOCK CH1-4 MUST BE 4 (FOR CK3A) CH1 MIN. 2μSEC FOR G305
Gate3[0].SerialEncCtrl = \$01000008 Gate3[0].Chan[0].SerialEncCmd = \$521000 Gate3[0].Chan[0].SerialEncEna = 1	CH1-4 SERIAL ENC. CTRL 20-BIT G5 CH1 SERIAL ENC. CMD CH1 SERIAL ENC. ENABLE

ECT(인코더 변환표)

EncTable[1].Type = 1 EncTable[1].pEnc = Gate3[0].Chan[0].SerialEncDataA.a EncTable[1].pEnc1 = Sys.Pushm EncTable[1].index1 = 12 EncTable[1].index2 = 0 EncTable[1].index3 = 0 EncTable[1].index4 = 0 EncTable[1].index5 = 0 EncTable[1].index6 = 0 EncTable[1].ScaleFactor = 1 / EXP2(EncTable[1].index1) EncTable[1].MaxDelta = 0	32-BIT READ 12-BIT LEFT SHIFT TO MSB FOR ROLLOVER 1/2 ^{INDEX1} SCALE TO LSB
EncTable[2].Type = 0	END OF ECT (FOR EFFICIENCY)

모터 설정 및 시운전

GLOBAL Ck3a1MaxAdc = 15.735	CK3A MAX ADC
GLOBAL Mtr1SingleTurn = 20	SINGLE TURN ST BITS
GLOBAL Mtr1CtsPerRev = EXP2(Mtr1SingleTurn)	COUNTS PER REV. =2 ST
GLOBAL Mtr1PolePairs = 5	NUMBER OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 10	MAX RMS CUR.(CK3A<MTR)
GLOBAL Mtr1RatedRmsCur = 2.5	RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2	TIME ALLOWED AT MAX CUR.
Motor[1].ServoCtrl = 1	ACTIVATE CHANNEL
Motor[1].PosUnit = 11	IDE WINDOW UNITS (DEGREES)
Motor[1].PosSf = 360 / Mtr1CtsPerRev	SCALE POSITION TO DEG.
Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev	SCALE VELOCITY TO DEG.
Motor[1].pLimits = Gate3[0].Chan[0].Status.a	ENABLE LIMITS, =0 TO DISABLE
Motor[1].AbortTa = 100	ABORT DECELERATION, 100 MSEC
Motor[1].AbortTs = 0	ABORT S-CURVE, NONE
Motor[1].JogTa = 100	JOG ACCELERATION, 100 MSEC
Motor[1].JogTs = 0	JOG S-CURVE, NONE
Motor[1].JogSpeed = 3	JOG SPEED, 3 DEG/MSEC
Motor[1].FatalFeLimit = 1	FE LIMIT, 1 DEG
Motor[1].WarnFeLimit = 0	WARNING FE LIMIT, NONE
Motor[1].InPosBand = 0.050	INPOS BAND, [DEG]
Motor[1].InPosTime = 5 / Sys.ServoPeriod	INPOS TIME, 5 MSEC
Motor[1].pPhaseEnc = Gate3[0].Chan[0].SerialEncDataA.a	SERIAL ENCODER
Motor[1].PhaseCtrl = 4	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 12	SHIFT LEFT TO MSB
Motor[1].PhaseEncRightShift = 0	NO RIGHT SHIFT
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / EXP2(32)	=2048*POLE PAIRS/2 ^{ST+LeftShift}
Motor[1].AmpFaultLevel = 1	=1 FOR CK3A
Motor[1].AdcMask = \$FFFF0000	= \$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384	TYPICAL IF MTR VOLTAGE>INPUT
Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	...
Motor[1].I2tTrip = (POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur	...
Motor[1].CurrentNullPeriod = 0	=0 FOR CK3A
Motor[1].IaBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IbBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IiGain = 0.144	CURRENT LOOP GAINS
Motor[1].IpGain = 2.7	...
Motor[1].IpbGain = 0	...
Motor[1].Servo.Kp = 1200	POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 13000	...
Motor[1].Servo.Kvifb = 0	...
Motor[1].Servo.Kvff = 13000	...
Motor[1].Servo.Kviff = 0	...
Motor[1].Servo.Ki = 0.016	...
Motor[1].Servo.Kaff = 80000	...
Motor[1].Servo.Kfff = 0	...
Motor[1].Servo.SwZvInt = 0	...
Motor[1].pAbsPos = Gate3[0].Chan[0].SerialEncDataA.a	TYPICAL FOR SERIAL ABS. ENC.
Motor[1].AbsPosFormat = \$1400	UNSIGNED ST BITS
Motor[1].AbsPosSF = Motor[1].PosSf	SAME AS POSITION SCALE
Motor[1].HomeOffset = 0	USER CONFIGURABLE POS. OFFSET
Motor[1].pAbsPhasePos = Gate3[0].Chan[0].SerialEncDataA.a	TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = \$1400	USE ST DATA FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1PolePairs / Mtr1CtsPerRev	=2048*MOTOR POLE PAIRS/2 ST
//Motor[1].AbsPhasePosForce = Gate3[0].Chan[0].SerialEncDataA & \$FFFF	MASK ST BITS, TEST VALUE
Motor[1].AbsPhasePosForce = 69614	STORE TEST VALUE
Motor[1].AbsPhasePosOffset = -Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf	TYPICAL SETTING
Motor[1].PowerOnMode = 0	NO POWER ON ABS READS

이 모터/인코더 어셈블리를 위한 특수 PLC

이 G5 인코더에 전원이 처음 공급될 때 절대 위치(및 위상 기준)를 판독하기 있기 전에 인덱스 표시로 이동하는 특별한 절차가 필요합니다. 다음은 이 절차를 수행하는 샘플 PLC입니다.

```
GLOBAL PhaseHomeFlag = 0

OPEN PLC PhaseHomePLC
IF(PhaseHomeFlag == 1)
{
    LOCAL EncStatusWord
    LOCAL StepAngle = 1
    LOCAL StepMag = 1000

    // READ ENCODER STATUS WORD
    EncStatusWord = (Gate3[0].Chan[0].SerialEncDataA & $F00000) >> 20
    CALL DelayTimer.msec(10)

    // NOT INITIALIZED?
    IF(EncStatusWord > 0)
    {
        WHILE(EncStatusWord > 0 && Motor[1].AmpFault == 0)
        {
            IF(Motor[1].AmpEna == 0)
            {
                COUT 1:0
                WHILE(Motor[1].AmpEna == 0 && Motor[1].AmpFault == 0){}
            }
            Motor[1].IaBias = StepMag * SIND(StepAngle + 120)
            Motor[1].IbBias = StepMag * SIND(StepAngle)
            EncStatusWord = (Gate3[0].Chan[0].SerialEncDataA & $F00000) >> 20
            StepAngle++
        }
        CALL DelayTimer.msec(50)

        Motor[1].IaBias = 0
        Motor[1].IbBias = 0
        KILL 1
        CALL DelayTimer.msec(50)
    }

    // ABSOLUTE PHASE READ
    Motor[1].PhaseFindingStep = 1
    CALL DelayTimer.msec(100)
    WHILE(Motor[1].PhaseFound == 0){}

    // ABSOLUTE POSITION READ
    HOMEZ 1
    CALL DelayTimer.msec(100)
    WHILE(Motor[1].HomeComplete == 0){}

    PhaseHomeFlag = 0
}
CLOSE
```

A-3-3 OMRON 1S 서보 모터(ACC-84E)

하드웨어 구성

항목	참고
컨트롤러	이 설정 샘플은 ACC-84E(1S 프로토콜)가 포함된 UMAC ACC-24E3에 적용됩니다
증폭기	CK3A-G305L(110VAC 주 공급 전압)
모터	1S 서보 모터 R88M-1M10030S, 100VAC, 1.5/4.7A
인코더	1S 직렬 절대 23비트 단일 회전, 16비트 다중 회전

시스템 게이트

<pre> Sys.WpKey = \$AAAAAAAA Sys.MaxCoords = 2 Sys.MaxMotors = 5 Gate3[0].PhaseFreq = 10000 Gate3[0].ServoClockDiv = 1 Sys.RtIntPeriod = 1 Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1) Gate3[0].Chan[0].PwmFreqMult = 1 Gate3[0].Chan[1].PwmFreqMult = 1 Gate3[0].Chan[2].PwmFreqMult = 1 Gate3[0].Chan[3].PwmFreqMult = 1 Gate3[0].Chan[0].PackInData = 0 Gate3[0].Chan[1].PackInData = 0 Gate3[0].Chan[2].PackInData = 0 Gate3[0].Chan[3].PackInData = 0 Gate3[0].Chan[0].PackOutData = 0 Gate3[0].Chan[1].PackOutData = 0 Gate3[0].Chan[2].PackOutData = 0 Gate3[0].Chan[3].PackOutData = 0 Gate3[0].AdcAmpStrobe = \$901001 Gate3[0].AdcAmpClockDiv = 5 Gate3[0].AdcAmpHeaderBits = 4 Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533 ACC84E[0].SerialEncCtrl = \$E ACC84E[0].Chan[0].SerialEncCmd = \$1400 </pre>	DISABLE WRITE-PROTECTION MAX. COORD SYSTEMS+1 (EFFICIENCY) MAX. NO. OF MOTORS+1 (EFFICIENCY) 10 KHZ PHASE FREQUENCY 5 KHZ SERVO FREQUENCY 2.5 KHZ RTI FREQUENCY TYPICAL CALCULATION TYPICAL CALCULATION CH1-4 10 KHZ PWM FREQUENCY CH1-4 IN DATA MUST BE 0 FOR CK3A CH1-4 OUT DATA MUST BE 0 FOR CK3A CH1-4 STROBE WORD (FOR CK3A) CH1-4 3.125 MHZ ADC CLOCK CH1-4 MUST BE 4 (FOR CK3A) CH1 MIN. 2μSEC FOR G305 ACC-84E CH1-4 FOR 1S ACC-84E CH1 SERIAL ENC. CMD FOR 1S
---	--

ECT(인코더 변환표)

<pre> EncTable[1].Type = 1 EncTable[1].pEnc = Gate3[0].Chan[0].SerialEncDataA.a EncTable[1].pEnc1 = Sys.Pushm EncTable[1].index1 = 8 EncTable[1].index2 = 8 EncTable[1].index3 = 0 EncTable[1].index4 = 0 EncTable[1].index5 = 0 EncTable[1].index6 = 0 EncTable[1].ScaleFactor = 1 / EXP2(EncTable[1].index1) EncTable[1].MaxDelta = 0 EncTable[2].Type = 0 </pre>	32-BIT READ ACC-84 (LEFT SHIFT) ACC-84 (RIGHT SHIFT) 1/2^{INDEX1} SCALE TO LSB END OF ECT (FOR EFFICIENCY)
--	--

모터 설정 및 시운전

GLOBAL Ck3a1MaxAdc = 15.735	CK3A MAX ADC
GLOBAL Mtr1SingleTurn = 23	SINGLE TURN ST BITS
GLOBAL Mtr1MultiTurn = 16	MULTI TURN MT BITS
GLOBAL Mtr1CtsPerRev = EXP2(Mtr1SingleTurn)	COUNTS PER REV. =2 ST
GLOBAL Mtr1PolePairs = 5	NUMBER OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 4.7	MAX RMS CUR.(MTR<CK3A)
GLOBAL Mtr1RatedRmsCur = 1.5	RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2	TIME ALLOWED AT MAX CUR.
Motor[1].ServoCtrl = 1	ACTIVATE CHANNEL
Motor[1].PosUnit = 11	IDE WINDOW UNITS (DEGREES)
Motor[1].PosSf = 360 / Mtr1CtsPerRev	SCALE POSITION TO DEGREES
Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev	SCALE VELOCITY TO DEGREES
Motor[1].pLimits = Gate3[0].Chan[0].Status.a	ENABLE LIMITS, =0 TO DISABLE
Motor[1].AbortTa = 100	ABORT DECELERATION, 100 MSEC
Motor[1].AbortTs = 0	ABORT S-CURVE, NONE
Motor[1].JogTa = 100	JOG ACCELERATION, 100 MSEC
Motor[1].JogTs = 0	JOG S-CURVE, NONE
Motor[1].JogSpeed = 3	JOG SPEED, 3 DEG/MSEC
Motor[1].FatalFeLimit = 1	FE LIMIT, 1 DEG
Motor[1].WarnFeLimit = 0	WARNING FE LIMIT, NONE
Motor[1].InPosBand = 0.050	INPOS BAND, [DEG]
Motor[1].InPosTime = 5 / Sys.ServoPeriod	INPOS TIME, 5 MSEC
Motor[1].pPhaseEnc = ACC84E[0].Chan[0].SerialEncDataA.a	ACC-84 SERIAL ENCODER
Motor[1].PhaseCtrl = 4	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 9	SHIFT LEFT TO MSB
Motor[1].PhaseEncRightShift = 8	TYPICAL ACC-84
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / EXP2(32)	=2048*POLE PAIRS/2 ^{ST+LeftShift}
Motor[1].AmpFaultLevel = 1	=1 FOR CK3A
Motor[1].AdcMask = \$FFFF0000	=\$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683	TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384	TYPICAL IF MTR VOLTAGE>INPUT
Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc	...
Motor[1].I2tTrip = (POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur	...
Motor[1].CurrentNullPeriod = 0	=0 FOR CK3A
Motor[1].IaBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IbBias = 0	=0 DURING NORMAL OPERATION
Motor[1].IiGain = 0	CURRENT LOOP GAINS
Motor[1].IpGain = 0	(MUST TUNE)
Motor[1].IpbGain = 0	...
Motor[1].Servo.Kp = 0	POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0	(MUST TUNE)
Motor[1].Servo.Kvifb = 0	...
Motor[1].Servo.Kvff = 0	...
Motor[1].Servo.Kviff = 0	...
Motor[1].Servo.Ki = 0	...
Motor[1].Servo.Kaff = 0	...
Motor[1].Servo.Kfff = 0	...
Motor[1].Servo.SwZvInt = 0	...
Motor[1].pAbsPos = ACC84E[0].Chan[0].SerialEncDataA.a	TYPICAL FOR SERIAL ABS. ENC.
Motor[1].AbsPosFormat = \$01082708	SIGNED ST+MT BITS
Motor[1].AbsPosSf = Motor[1].PosSf	SAME AS POSITION SCALE
Motor[1].HomeOffset = 0	USER CONFIGURABLE POS. OFFSET
Motor[1].pAbsPhasePos = ACC84E[0].Chan[0].SerialEncDataA.a	TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = \$1708	USE ST DATA FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1PolePairs / Mtr1CtsPerRev	=2048*MOTOR POLE PAIRS/2 ST
//Motor[1].AbsPhasePosForce = ACC84E[0].Chan[0].SerialEncDataA & \$7FFFFFFF	MASK ST BITS, TEST VALUE
Motor[1].AbsPhasePosForce = 1269945	STORE TEST VALUE
Motor[1].AbsPhasePosOffset = -Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf	TYPICAL SETTING
Motor[1].PowerOnMode = \$6	POWER-ON ABS. PHASE AND POS

A-3-4 정현파 인코더가 있는 리니어 서보 모터

하드웨어 구성

항목	참고
컨트롤러	이 샘플은 UMAC ACC-24E3(표준 정현파) 및 CK3M AX(정현파)에 적용 가능합니다.
증폭기	CK3A-G310L(3상 208VAC 주 공급 전압)
모터	리니어 서보 모터, 200VAC, 2.6/8.3A, 60.96mm 전기 사이클 길이
인코더	정현파, 20μm(분해능) 사인 사이클 길이

시스템 게이트

Sys.WpKey = \$AAAAAAA Sys.MaxCoords = 2 Sys.MaxMotors = 5	DISABLE WRITE-PROTECTION MAX. COORD SYSTEMS+1 (EFFICIENCY) MAX. NO. OF MOTORS+1 (EFFICIENCY)
Gate3[0].PhaseFreq = 10000 Gate3[0].ServoClockDiv = 1 Sys.RtIntPeriod = 1 Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)	10 KHZ PHASE FREQUENCY 5 KHZ SERVO FREQUENCY 2.5 KHZ RTI FREQUENCY TYPICAL CALCULATION TYPICAL CALCULATION
Gate3[0].Chan[0].PwmFreqMult = 1 Gate3[0].Chan[1].PwmFreqMult = 1 Gate3[0].Chan[2].PwmFreqMult = 1 Gate3[0].Chan[3].PwmFreqMult = 1	CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[0].PackInData = 0 Gate3[0].Chan[1].PackInData = 0 Gate3[0].Chan[2].PackInData = 0 Gate3[0].Chan[3].PackInData = 0	CH1-4 IN DATA MUST BE 0 FOR CK3A
Gate3[0].Chan[0].PackOutData = 0 Gate3[0].Chan[1].PackOutData = 0 Gate3[0].Chan[2].PackOutData = 0 Gate3[0].Chan[3].PackOutData = 0	CH1-4 OUT DATA MUST BE 0 FOR CK3A
Gate3[0].AdcAmpStrobe = \$901001 Gate3[0].AdcAmpClockDiv = 5 Gate3[0].AdcAmpHeaderBits = 4 Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533	CH1-4 STROBE WORD (FOR CK3A) CH1-4 3.125 MHZ ADC CLOCK CH1-4 MUST BE 4 (FOR CK3A) CH1 MIN. 3μSEC FOR G310
Gate3[0].EncClockDiv = 3 Gate3[0].AdcEncClockDiv = 3 //Gate3[0].AdcEncCtrl = \$3FFFC000	ENC. SAMPLING 12.5MHZ ENC. ADC SAMPLING 12.5MHZ UNCOMMENT FOR CK3M AX

ECT(인코더 변환표)

EncTable[1].Type = 1 EncTable[1].pEnc = Gate3[0].Chan[0].ServoCapt.a EncTable[1].pEnc1 = Sys.Pushm EncTable[1].index1 = 0 EncTable[1].index2 = 0 EncTable[1].index3 = 0 EncTable[1].index4 = 0 EncTable[1].index5 = 0 EncTable[1].index6 = 0 EncTable[1].ScaleFactor = 1 EncTable[1].MaxDelta = 0	일반적인 정현파 인코더 항목
EncTable[2].Type = 0	ECT 끝(효율성)

모터 설정 및 시운전

<pre> GLOBAL Ck3a1MaxAdc = 31.470 GLOBAL Mtr1Res = 0.020 GLOBAL Mtr1CtsPerMm = 16384 / Mtr1Res GLOBAL Mtr1Ec1 = 60.96 GLOBAL Mtr1MaxRmsCur = 8.3 GLOBAL Mtr1RatedRmsCur = 2.6 GLOBAL Mtr1TimeAtMaxCur = 2 Motor[1].EncType = 6 Gate3[0].Chan[0].AtanEna = 1 Gate3[0].Chan[0].EncCtrl = 7 Motor[1].ServoCtrl = 1 Motor[1].PosUnit = 3 Motor[1].PosSf = 1 / Mtr1CtsPerMm Motor[1].Pos2Sf = 1 / Mtr1CtsPerMm Motor[1].pLimits = Gate3[0].Chan[0].Status.a Motor[1].AbortTa = 100 Motor[1].AbortTs = 0 Motor[1].JogTa = 100 Motor[1].JogTs = 0 Motor[1].JogSpeed = 0.050 Motor[1].FatalFeLimit = 0.500 Motor[1].WarnFeLimit = 0 Motor[1].InPosBand = 0.000100 Motor[1].InPosTime = 10 / Sys.ServoPeriod Motor[1].pPhaseEnc = Gate3[0].Chan[0].PhaseCapt.a Motor[1].PhaseCtrl = 4 Motor[1].PhaseEncLeftShift = 0 Motor[1].PhaseEncRightShift = 0 Motor[1].PhasePosSf = 2048 * Mtr1Res / (Mtr1Ec1 * 16384) Motor[1].AmpFaultLevel = 1 Motor[1].AdcMask = \$FFFF0000 Motor[1].PhaseOffset = 683 Motor[1].PwmSf = 0.95 * 16384 Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2tTrip = (POW(Motor[1].MaxDac, 2) - POW(Motor[1].I2TSet, 2)) * Mtr1TimeAtMaxCur Motor[1].CurrentNullPeriod = 0 Motor[1].IaBias = 0 Motor[1].IbBias = 0 Motor[1].IiGain = 0 Motor[1].IpfGain = 0 Motor[1].IpbGain = 0 Motor[1].Servo.Kp = 0 Motor[1].Servo.Kvfb = 0 Motor[1].Servo.Kvifb = 0 Motor[1].Servo.Kvff = 0 Motor[1].Servo.Kviff = 0 Motor[1].Servo.Ki = 0 Motor[1].Servo.Kaff = 0 Motor[1].Servo.Kfff = 0 Motor[1].Servo.SwZvInt = 0 Motor[1].pAbsPhasePos = Gate3[0].Chan[0].Status.a Motor[1].AbsPhasePosFormat = \$400030C Motor[1].AbsPhasePosSf = 2048 / 12 Motor[1].AbsPhasePosOffset = 0 Motor[1].PowerOnMode = 2 </pre>	CK3A 최대 ADC 분해능, 인코더 피치 [mm] MM당 카운트 전기 사이클 길이 [mm] 최대 RMS 전류(MTR<CK3A) 정격 RMS 전류(MTR<CK3A) 최대 전류에서 허용되는 시간 정현파에 일반적 정현파에 일반적 인코더 카운팅 방향 채널 활성화 IDE 창 단위 [mm] 위치를 mm로 환산 속도를 mm로 환산 제한 활성화, 비활성화하려면 =0 중단 감속, 100MSEC 중단 S-곡선, 없음 조그 가속, 100MSEC 조그 S-곡선, 없음 조그 속도, 50 mm/sec FE 제한, 500μm 경고 FE 제한, 없음 INPOS 대역, 100nm INPOS 시간, 10MSEC 정현파에 일반적 브러시리스 모터에 일반적 정현파의 경우 없음 정현파의 경우 없음 정현파에 일반적 CK3A의 경우 =1 CK3A의 경우 = \$FFFF0000 브러시리스 모터에 일반적 MTR 전압>입력인 경우 일반적 브러시리스에 일반적인 I2T CK3A의 경우 =0 정상 작동 시 =0 정상 작동 시 =0 전류 루프 게인 (튜닝 필수) ... 위치 서보 루프 게인 (튜닝 필수) 홀 절대 위상 조정 홀 절차에 따라 ±를 설정해야 함 홀 절차에 따라 찾아야 함 전원을 켤 때 절대 홀 위상
--	--

A-3-5 BiSS 인코더가 포함된 리니어 서보 모터(ACC-84E)

하드웨어 구성

항목	참고
컨트롤러	이 샘플은 ACC-84E(BISS)가 포함된 UMAC ACC-24E3에 적용 가능합니다.
증폭기	CK3A-G310L(3상 208VAC 주 공급 전압)
모터	리니어 서보 모터, 200VAC, 2.6/8.3A, 60.96mm 전기 사이클 길이
인코더	BiSS, 32비트, 1nm 분해능

시스템 게이트

```

Sys.WpKey = $AAAAAAA
Sys.MaxCoords = 2
Sys.MaxMotors = 5

Gate3[0].PhaseFreq = 10000
Gate3[0].ServoClockDiv = 1
Sys.RtIntPeriod = 1
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)

Gate3[0].Chan[0].PwmFreqMult = 1
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0
Gate3[0].Chan[1].PackInData = 0
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0
Gate3[0].Chan[1].PackOutData = 0
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001
Gate3[0].AdcAmpClockDiv = 5
Gate3[0].AdcAmpHeaderBits = 4
Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533

ACC84E[0].SerialEncCtrl = $31000B
ACC84E[0].Chan[0].SerialEncCmd = $2114A0

```

쓰기 방지 비활성화
최대 조정 시스템+1(효율성)
최대 모터 수+1(효율성)

10KHZ 위상 주파수
5KHZ 서보 주파수
2.5KHZ RTI 주파수
일반적인 계산
일반적인 계산

CH1-4 10KHZ PWM 주파수

CH1-4 인 데이터
CK3A의 경우 0이어야 함

CH1-4 아웃 데이터
CK3A의 경우 0이어야 함

CH1-4 스트로브 워드(CK3A의 경우)
CH1-4 3.125MHZ ADC 클럭
CH1-4는 4여야 함(CK3A의 경우)
CH1 최소 G310의 경우 3MSEC

직렬 인코더 CTRL(BISS)
직렬 인코더 명령(32비트)

ECT(인코더 변환표)

```

EncTable[1].Type = 1
EncTable[1].pEnc = ACC84E[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 8
EncTable[1].index2 = 8
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = EXP2(EncTable[1].index1)
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0

```

일반적인 직렬 ACC-84 항목

ECT 끝(효율성)

모터 설정 및 시운전

GLOBAL Ck3a1MaxAdc = 31.470 GLOBAL Mtr1Res = 0.000001 GLOBAL Mtr1CtsPerMm = 1 / Mtr1Res GLOBAL Mtr1Ec1 = 60.96 GLOBAL Mtr1MaxRmsCur = 8.3 GLOBAL Mtr1RatedRmsCur = 2.6 GLOBAL Mtr1TimeAtMaxCur = 2	CK3A 최대 ADC 1nm 분해능 [mm] MM당 카운트 전기 사이클 길이 [mm] 최대 RMS 전류(MTR<CK3A) 정격 RMS 전류(MTR<CK3A) 최대 전류에서 허용되는 시간
Motor[1].ServoCtrl = 1 Motor[1].PosUnit = 3 Motor[1].PosSf = 1 / Mtr1CtsPerMm Motor[1].Pos2Sf = 1 / Mtr1CtsPerMm Motor[1].pLimits = Gate3[0].Chan[0].Status.a	채널 활성화 IDE 창 단위 [mm] 위치를 mm로 환산 속도를 mm로 환산 제한 활성화, 비활성화하려면 =0
Motor[1].AbortTa = 100 Motor[1].AbortTs = 0	중단 감속, 100MSEC 중단 S-곡선, 없음
Motor[1].JogTa = 100 Motor[1].JogTs = 0 Motor[1].JogSpeed = 0.050	조그 가속, 100MSEC 조그 S-곡선, 없음 조그 속도, 50 mm/sec
Motor[1].FatalFeLimit = 0.500 Motor[1].WarnFeLimit = 0	FE 제한, 500μm 경고 FE 제한, 없음
Motor[1].InPosBand = 0.000100 Motor[1].InPosTime = 10 / Sys.ServoPeriod	INPOS 대역, 100nm INPOS 시간, 10MSEC
Motor[1].pPhaseEnc = ACC84E[0].Chan[0].SerialEncDataA.a Motor[1].PhaseCtrl = 4 Motor[1].PhaseEncLeftShift = 8 Motor[1].PhaseEncRightShift = 8 Motor[1].PhasePosSf = 2048 * Mtr1Res / (EXP2(Motor[1].PhaseEncLeftShift) * Mtr1Ec1)	직렬에 일반적 브러시리스 모터에 일반적 ACC-84 ACC-84 직렬에 일반적
Motor[1].AmpFaultLevel = 1 Motor[1].AdcMask = \$FFFF0000 Motor[1].PhaseOffset = 683 Motor[1].PwmSf = 0.95 * 16384	CK3A의 경우 =1 CK3A의 경우 = \$FFFF0000 브러시리스 모터에 일반적 MTR 전압>입력인 경우 일반적
Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2tTrip = (POW(Motor[1].MaxDac, 2) - POW(Motor[1].I2TSet, 2)) * Mtr1TimeAtMaxCur	브러시리스에 일반적인 I2T
Motor[1].CurrentNullPeriod = 0 Motor[1].IaBias = 0 Motor[1].IbBias = 0	CK3A의 경우 =0 정상 작동 시 =0 정상 작동 시 =0
Motor[1].IiGain = 0 Motor[1].IpfGain = 0 Motor[1].IpbGain = 0	전류 루프 게인 (튜닝 필수) ...
Motor[1].Servo.Kp = 0 Motor[1].Servo.Kvfb = 0 Motor[1].Servo.Kvifb = 0 Motor[1].Servo.Kvff = 0 Motor[1].Servo.Kviff = 0 Motor[1].Servo.Ki = 0 Motor[1].Servo.Kaff = 0 Motor[1].Servo.Kfff = 0 Motor[1].Servo.SwZvInt = 0	위치 서보 루프 게인 (튜닝 필수)
Motor[1].pAbsPos = ACC84E[0].Chan[0].SerialEncDataA.a Motor[1].AbsPosFormat = \$00082008 Motor[1].AbsPosSf = Motor[1].PosSf Motor[1].HomeOffset = 0	직렬 절대 인코더에 일반적 할당되지 않은 32비트 위치 스케일과 동일 사용자 구성 가능한 위치 오프셋
Motor[1].pAbsPhasePos = ACC84E[0].Chan[0].SerialEncDataA.a Motor[1].AbsPhasePosFormat = \$00082008 Motor[1].AbsPhasePosSf = 2048 * Mtr1Res / Mtr1Ec1 //Motor[1].AbsPhasePosForce = //ACC84E[0].Chan[0].SerialEncDataA+(ACC84E[0].Chan[0].SerialEncDataB&\$FF)*16777216 Motor[1].AbsPhasePosForce = 253109861 Motor[1].AbsPhasePosOffset = -Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf Motor[1].PowerOnMode = \$6	절대 직렬 인코더에 일반적 절대 위상의 경우 32비트 사용 절대 직렬 인코더에 일반적 32비트 판독, 테스트 값 테스트 값 저장 일반 설정 전원을 켤 때 절대 위상 및 위치

A-4 Gate1 "스크립트" 모터 설정 샘플

이 섹션에서는 Gate1이 포함된 브러시리스 모터 설정 샘플(UMAC ACC-24E2)이 참조용으로 제공됩니다.



안전한 사용을 위한 주의사항

이 섹션에서 제공되는 샘플은 참조용일 뿐이며 모터가 즉시 작동하지 않을 수 있습니다. 다음 항목에 주의해야 합니다.

- 인코더 및 모터 사양 매개변수
- 전류 루프 튜닝을 수행해야 함
- 초기 위상 조정을 수행해야 함
- 개방 루프 테스트를 수행해야 함
- 위치 루프 튜닝을 수행해야 함
- 절대 위치 설정(해당되는 경우)을 검토해야 함
- 절대 위상 기준 설정(해당되는 경우)을 검토해야 함
- 일반 모션 매개변수(예: 조그 속도)를 검토해야 함



추가 정보

쉽게 설명하기 위해 이 섹션의 샘플은 모터 #1 및 Gate1, 인덱스 4, Power PMAC 섹션의 채널 0, Turbo PMAC 섹션의 기본 주소 78200을 참조합니다.

A-4-1 Gate1이 있는 Power PMAC(ACC-24E2)

하드웨어 구성

항목	참고
컨트롤러	이 샘플은 ACC-24E2가 있는 Power PMAC UMAC에 적용됩니다.
증폭기	CK3A-G305L(단상 110VAC 주 공급 전압)
모터	로터리 서보 모터, 48VDC, 3/9A, 2극 쌍
인코더	디지털 직각위상, 500라인(x4 디코딩으로 회전당 2,000카운트)

시스템 게이트

```

Sys.WpKey = $AAAAAAA
Sys.MaxCoords = 2
Sys.MaxMotors = 5

Gate1[4].PwmPeriod = 2948
Gate1[4].PhaseClockDiv = 1
Gate1[4].ServoClockDiv = 1
Sys.RtIntPeriod = 1
Sys.ServoPeriod = 0.2
Sys.PhaseOverServoPeriod = 0.5

Gate1[4].AdcStrobe = $901001
Gate1[4].HardwareClockCtrl = 2258
Gate1[4].Chan[0].PwmDeadTime = 2 / 0.135

```

쓰기 방지 비활성화
최대 조정 시스템+1(효율성)
최대 모터 수+1(효율성)

20KHZ 최대 위상 주파수
10KHZ 위상 및 PWM 주파수
5KHZ 서보 주파수
2.5KHZ RTI 주파수
서보 주기 [MSEC]
위상 오버 서보 비율

CH1-4 스트로브 워드(CK3A의 경우)
CH1-4 2.4576MHz ADC 클럭(기본값)
CH1-4 2MSEC(G310의 경우)

ECT(인코더 변환표)

```

EncTable[1].Type = 3
EncTable[1].pEnc = Gate1[4].Chan[0].ServoCapt.a
EncTable[1].pEnc1 = Gate1 [4].Chan[0].TimeBetweenCts.a
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / 512
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0

```

일반적인 직각위상 인코더 항목

ECT 끝(효율성)

모터 설정 및 시운전

<pre> GLOBAL Ck3a1MaxAdc = 15.735 GLOBAL Mtr1CtsPerRev = 2000 GLOBAL Mtr1PolePairs = 4 GLOBAL Mtr1MaxRmsCur = 9 GLOBAL Mtr1RatedRmsCur = 3 GLOBAL Mtr1TimeAtMaxCur = 2 Gate1[4].Chan[0].EncCtrl = 7 Motor[1].ServoCtrl = 1 Motor[1].PosUnit = 11 Motor[1].PosSf = 360 / Mtr1CtsPerRev Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev Motor[1].pLimits = Gate1[4].Chan[0].Status.a Motor[1].AbortTa = 150 Motor[1].AbortTs = 0 Motor[1].JogTa = 300 Motor[1].JogTs = 0 Motor[1].JogSpeed = 3 Motor[1].FatalFeLimit = 5 Motor[1].WarnFeLimit = 0 Motor[1].InPosBand = 0.01 Motor[1].InPosTime = 20 / Sys.ServoPeriod Motor[1].pPhaseEnc = Gate1[4].Chan[0].PhaseCapt.a Motor[1].PhaseCtrl = 4 Motor[1].PhaseEncLeftShift = 0 Motor[1].PhaseEncRightShift = 0 Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / (256 * Mtr1CtsPerRev) Motor[1].AmpFaultLevel = 1 Motor[1].AdcMask = \$FFFF0000 Motor[1].PhaseOffset = -683 Motor[1].PwmSf = -Gate1[4].PwmPeriod * 48 / 155 Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc Motor[1].I2tTrip = (POW(Motor[1].MaxDac, 2) - POW(Motor[1].I2TSet, 2)) * Mtr1TimeAtMaxCur Motor[1].CurrentNullPeriod = 0 Motor[1].IaBias = 0 Motor[1].IbBias = 0 Motor[1].IiGain = 0 Motor[1].IpfGain = 0 Motor[1].IpbGain = 0 Motor[1].Servo.Kp = 0 Motor[1].Servo.Kvfb = 0 Motor[1].Servo.Kvifb = 0 Motor[1].Servo.Kvff = 0 Motor[1].Servo.Kviff = 0 Motor[1].Servo.Ki = 0 Motor[1].Servo.Kaff = 0 Motor[1].Servo.Kfff = 0 Motor[1].Servo.SwZvInt = 0 Motor[1].PhaseFindingTime = 4000 / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1)) Motor[1].PhaseFindingDac = 0.5 * Motor[1].I2tSet Motor[1].AbsPhasePosOffset = 2048 / 5 Motor[1].PowerOnMode = 0 </pre>	CK3A 최대 ADC 회전당 카운트 극 쌍 수 최대 RMS 전류(MTR>CK3A) 정격 RMS 전류(MTR>CK3A) 최대 전류에서 허용되는 시간 인코더 카운팅 방향 채널 활성화 IDE 창 단위, 도 위치를 도로 환산 속도를 도로 환산 제한 활성화, 비활성화하려면 =0 중단 감속, 150MSEC 중단 S-곡선, 없음 조그 가속, 300MSEC 조그 S-곡선, 없음 조그 속도, 4.5도/MSEC FE 제한, 5도 경고 FE 제한, 없음 INPOS 대역, 0.05도 INPOS 시간, 20MSEC 직각위상에 일반적 브러시리스 모터에 일반적 직각위상의 경우 없음 직각위상의 경우 없음 직각위상에 일반적 CK3A의 경우 =1 CK3A의 경우 = \$FFFF0000 브러시리스 모터에 일반적 MTR 전압>입력인 경우 일반적 브러시리스에 일반적인 I2T ... CK3A의 경우 =0 정상 작동 시 =0 정상 작동 시 =0 전류 루프 게인 (튜닝 필수) ... 위치 서보 루프 게인 (튜닝 필수) 스테퍼 위상 조정(4초) I2TSET의 50% 일반 설정 전원을 켤 때 절대 위상 없음
---	---

A-4-2 Gate1이 있는 Turbo PMAC(ACC-24E2)

하드웨어 구성

항목	참고
컨트롤러	이 샘플은 ACC-24E2가 있는 Turbo PMAC UMAC에 적용됩니다.
증폭기	CK3A-G305L(단상 110VAC 주 공급 전압)
모터	로터리 서보 모터, 48VDC, 3/9A, 2극 쌍
인코더	디지털 직각위상, 500라인(x4 디코딩으로 회전당 2,000카운트)

시스템 게이트

```

END GAT
DEL GAT
CLOSE

I7200 = 2947
I7201 = 1
I7202 = 1
I10 = 1677368

I7206 = $901001
I7203 = 2258
I7204 = 2 / 0.135

#define MaxPhaseFreq P8000 ; Max Phase Clock [KHz]
#define PWMClk P8001 ; PWM Clock [KHz]
#define PhaseClk P8002 ; Phase Clock [KHz]
#define ServoClk P8003 ; Servo Clock [KHz]
MaxPhaseFreq = 117964.8 / (2 * I7200 + 3)
PWMClk = 117964.8 / (4 * I7200 + 6)
PhaseClk = MaxPhaseFreq / (I7201 + 1)
ServoClk = PhaseClk / (I7202 + 1)

```

수집 종료

수집 버퍼 삭제

열린 버퍼 닫기

20KHZ 최대 위상 주파수

10KHZ 위상 및 PWM 주파수

5KHZ 서보 주파수

서보 인터럽트 시간

CH1-4 스트로브 워드(CK3A의 경우)

CH1-4 2.4576MHz ADC 클럭(기본값)

CH1-4 2MSEC(G310의 경우)

ECT(인코더 변환표)

```

I8000 = $78200
I8001 = 0

```

일반적인 직각위상 인코더 항목

ECT 끝(효율성)

모터 설정 및 시운전

```

I7210 = 3
I100 = 1
I101 = 1
I124 = $800001

I184 = $FFFF00
I172 = 683
I166 = I7200 * 48 / 155

I170 = 2
I171 = 2000

I119 = 0.5
I120 = 100
I121 = 0
I122 = 32
I123 = 16
I115 = 0.5
I111 = 55 * 16
I112 = 0
I128 = 1 * 16

#define ContCurrent      3
#define PeakCurrent      9
#define MaxADC           15.735
#define I2TOnTime        2
I157 = INT(32767 * (ContCurrent * 1.414 / MaxADC) * COS(30))
I169 = INT(32767 * (PeakCurrent * 1.414 / MaxADC) * COS(30))
I158 = INT((I169*I169-I157*I157)*ServoClk*1000*I2TOnTime/(32767*32767))

I182 = $78216
I129 = 0
I179 = 0
I161 = 0
I162 = 0
I176 = 0

I130 = 0
I131 = 0
I132 = 0
I133 = 0
I134 = 0
I135 = 0
I136 = 0
I137 = 0
I138 = 0
I139 = 0
I168 = 0

I180 = 6
I173 = 1000
I174 = 60

```

인코더 카운팅 방향
SERVOCTRL 활성화
PHASECTRL 활성화
플래그 제어

ADC 마스크(CK3A의 경우)
위상 오프셋, 브러시리스에 일반적
PWM 환산 계수, 모터<공급 전압

극 쌍 수
회전당 카운트

기본 모션 설정
...

I2T 계산

전류 루프 주소

전류 루프 튜닝

위치 루프 튜닝

4-GUESS 위상 조정 방법

A-5 Gate1용 CK3A 샘플 PLC

A-5-1 Gate1이 있는 Power PMAC

CK3A 정의 4 모터

```
// GATE3[0], CH. 1 - 4 AMPLIFIER FUNCTIONS ===== //
GLOBAL Ck3a0ClrF, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl
PTR Gate30DataCtrl ->U.IO:$6000B0.16.4
PTR Ck3a0BusDisState ->U.IO:$6000B0.13.1
PTR Ck3a0ClrFBit ->U.IO:$6000B0.24.1
PTR Ck3a0FanState ->U.IO:$6000B0.25.1
PTR Ck3a0DbState ->U.IO:$6000B0.28.1

// CH. 1 - 4 ADC B REGISTERS [15:08] ===== //
PTR Ck3a1AdcBData ->U.IO:$600018.8.8 // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData ->U.IO:$600058.8.8 // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData ->U.IO:$600098.8.8 // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData ->U.IO:$6000D8.8.8 // Gate3[0].Chan[3].AdcAmp[1]

// CH. 1 - 4 AMPLIFIER DATA ===== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a1PwrFlt ->U.IO:$600014.8.1
PTR Ck3a1SoftStartFlt ->U.IO:$600014.9.1
PTR Ck3a1StoStatus ->U.IO:$600014.10.1
PTR Ck3a1ShuntShortFlt ->U.IO:$600014.11.1
PTR Ck3a1OverVoltFlt ->U.IO:$600014.12.1
PTR Ck3a1I2tFlt ->U.IO:$600014.13.1
PTR Ck3a1ShortFlt ->U.IO:$600014.14.1
PTR Ck3a1OverLoadFlt ->U.IO:$600014.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a2PwrFlt ->U.IO:$600054.8.1
PTR Ck3a2SoftStartFlt ->U.IO:$600054.9.1
PTR Ck3a2StoStatus ->U.IO:$600054.10.1
PTR Ck3a2ShuntShortFlt ->U.IO:$600054.11.1
PTR Ck3a2OverVoltFlt ->U.IO:$600054.12.1
PTR Ck3a2I2tFlt ->U.IO:$600054.13.1
PTR Ck3a2ShortFlt ->U.IO:$600054.14.1
PTR Ck3a2OverLoadFlt ->U.IO:$600054.15.1

// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a3PwrFlt ->U.IO:$600094.8.1
PTR Ck3a3SoftStartFlt ->U.IO:$600094.9.1
PTR Ck3a3StoStatus ->U.IO:$600094.10.1
PTR Ck3a3ShuntShortFlt ->U.IO:$600094.11.1
PTR Ck3a3OverVoltFlt ->U.IO:$600094.12.1
PTR Ck3a3I2tFlt ->U.IO:$600094.13.1
PTR Ck3a3ShortFlt ->U.IO:$600094.14.1
PTR Ck3a3OverLoadFlt ->U.IO:$600094.15.1

// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ===== //
PTR Ck3a4PwrFlt ->U.IO:$6000D4.8.1
PTR Ck3a4SoftStartFlt ->U.IO:$6000D4.9.1
PTR Ck3a4StoStatus ->U.IO:$6000D4.10.1
PTR Ck3a4ShuntShortFlt ->U.IO:$6000D4.11.1
PTR Ck3a4OverVoltFlt ->U.IO:$6000D4.12.1
PTR Ck3a4I2tFlt ->U.IO:$6000D4.13.1
PTR Ck3a4ShortFlt ->U.IO:$6000D4.14.1
PTR Ck3a4OverLoadFlt ->U.IO:$6000D4.15.1

// CH. 1, SYS.UDATA[256000] STATUS SET 2 ===== //
PTR Ck3a1PwrStatus ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna ->U.USER:$FA000.6.1
```

\$6000B0은 Gate1[i].AdcStrobe이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].AdcStrobe.a - Sys.piom

\$600xx8은 ADC B
Gate1[i].Chan[j].Adc[1]이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].Chan[j].Adc[1].a - Sys.piom

\$600014는 ADC A
Gate1[i].Chan[j].Adc[0]이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].Chan[j].Adc[0].a - Sys.piom

\$600054는 ADC A
Gate1[i].Chan[j].Adc[0]이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].Chan[j].Adc[0].a - Sys.piom

\$600094는 ADC A
Gate1[i].Chan[j].Adc[0]이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].Chan[j].Adc[0].a - Sys.piom

\$6000D4는 ADC A
Gate1[i].Chan[j].Adc[0]이며, 주소는 다음과 같이 찾을 수 있습니다.
Gate1[i].Chan[j].Adc[0].a - Sys.piom

\$FA000은 Sys.Udata[256000]이며, 주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a - Sys.pushm

```
// CH. 2, SYS.UDATA[256001] STATUS SET 2 ===== //
PTR Ck3a2PwrStatus      ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt    ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLF1t   ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt     ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt      ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna         ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ===== //
PTR Ck3a3PwrStatus      ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt    ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLF1t   ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt     ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt      ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna         ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ===== //
PTR Ck3a4PwrStatus      ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt    ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLF1t   ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt     ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt      ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna         ->U.USER:$FA00C.6.1

// CH. 1, SYS.UDATA[256040] STATUS SET 3 ===== //
PTR Ck3a1I2tInt         ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt   ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready          ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ===== //
PTR Ck3a2I2tInt         ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt   ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready          ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ===== //
PTR Ck3a3I2tInt         ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt   ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready          ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ===== //
PTR Ck3a4I2tInt         ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt   ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready          ->U.USER:$FA0AC.6.1
```

\$FA004는 Sys.Udata[256001]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA008은 Sys.Udata[256002]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA00C는 Sys.Udata[256003]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA0A0는 Sys.Udata[256040]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA0A4는 Sys.Udata[256041]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA0A8은 Sys.Udata[256042]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

\$FA0AC는 Sys.Udata[256043]이며,
주소는 다음과 같이 찾을 수 있습니다.
Sys.Udata[i].a – Sys.pushm

CK3A 샘플 PLC 4 모터

```

OPEN PLC Ck3aPLC
// CH. 1-4 STATUS BITS SET 2
Gate30DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 STATUS BITS SET 3
Gate30DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 DC BUS VOLTAGE
Gate30DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
CALL DelayTimer.msec(1)
// CH. 1-4 TEMPERATURE
Gate30DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 FIRMWARE VERSION
Gate30DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 CURRENT RATING
Gate30DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Ck3a0BusDisState = 0
// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Ck3a0FanState = 0
// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Ck3a0DbState = 1
// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0ClrF == 1)
{
    Ck3a0ClrFBit = 1
    CALL DelayTimer.msec(10)
    Ck3a0ClrFBit = 0
    Ck3a0ClrF = 0
}
// CH. 1-4 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
CLOSE

```

A-5-2 Gate1이 있는 Turbo PMAC

CK3A 정의 4 모터

```

// CH. 1-4 AMPLIFIER FUNCTIONS
#define Ck3a0ClrF          P1000
#define Ck3a0BusDisCtrl    P1001
#define Ck3a0FanCtrl       P1002
#define Ck3a0DbCtrl        P1003

#define Ck3a0BusDisState    P1004
#define Ck3a0FanState       P1005
#define Ck3a0DbState        P1006

// CH. 1-4 ADC B DATA REGISTERS
#define Ck3a1AdcBData       M1000
#define Ck3a2AdcBData       M1001
#define Ck3a3AdcBData       M1002
#define Ck3a4AdcBData       M1003

Ck3a1AdcBData->Y:$078206,0,8
Ck3a2AdcBData->Y:$07820E,0,8
Ck3a3AdcBData->Y:$078216,0,8
Ck3a4AdcBData->Y:$07821E,0,8

// CH. 1-4 DATA
#define Ck3a1DcBus          P1010
#define Ck3a1Temp           P1011
#define Ck3a1FwVer          P1012
#define Ck3a1CurRating      P1013

#define Ck3a2DcBus          P1014
#define Ck3a2Temp           P1015
#define Ck3a2FwVer          P1016
#define Ck3a2CurRating      P1017

#define Ck3a3DcBus          P1018
#define Ck3a3Temp           P1019
#define Ck3a3FwVer          P1020
#define Ck3a3CurRating      P1021

#define Ck3a4DcBus          P1022
#define Ck3a4Temp           P1023
#define Ck3a4FwVer          P1024
#define Ck3a4CurRating      P1025

// Ch. 1-4 STATUS BITS SET 1, ADC A (ANY STROBE)
#define Ck3a1PwrFlt         M1004
#define Ck3a1SoftStartFlt   M1005
#define Ck3a1StoStatus       M1006
#define Ck3a1ShuntShortFlt   M1007
#define Ck3a1OverVoltFlt    M1008
#define Ck3a1I2tFlt         M1009
#define Ck3a1ShortFlt       M1010
#define Ck3a1OverLoadFlt    M1011
Ck3a1PwrFlt->Y:$78205,0,1
Ck3a1SoftStartFlt->Y:$78205,1,1
Ck3a1StoStatus->Y:$78205,2,1
Ck3a1ShuntShortFlt->Y:$78205,3,1
Ck3a1OverVoltFlt->Y:$78205,4,1
Ck3a1I2tFlt->Y:$78205,5,1
Ck3a1ShortFlt->Y:$78205,6,1
Ck3a1OverLoadFlt->Y:$78205,7,1

#define Ck3a2PwrFlt         M1012
#define Ck3a2SoftStartFlt   M1013
#define Ck3a2StoStatus       M1014
#define Ck3a2ShuntShortFlt   M1015
#define Ck3a2OverVoltFlt    M1016
#define Ck3a2I2tFlt         M1017
#define Ck3a2ShortFlt       M1018
#define Ck3a2OverLoadFlt    M1019
Ck3a2PwrFlt->Y:$7820D,0,1
Ck3a2SoftStartFlt->Y:$7820D,1,1
Ck3a2StoStatus->Y:$7820D,2,1
Ck3a2ShuntShortFlt->Y:$7820D,3,1
Ck3a2OverVoltFlt->Y:$7820D,4,1
Ck3a2I2tFlt->Y:$7820D,5,1
Ck3a2ShortFlt->Y:$7820D,6,1
Ck3a2OverLoadFlt->Y:$7820D,7,1

```

```

#define Ck3a3PwrFlt           M1020
#define Ck3a3SoftStartFlt     M1021
#define Ck3a3StoStatus        M1022
#define Ck3a3ShuntShortFlt    M1023
#define Ck3a3OverVoltFlt      M1024
#define Ck3a3I2tFlt           M1025
#define Ck3a3ShortFlt         M1026
#define Ck3a3OverLoadFlt      M1027
Ck3a3PwrFlt->Y:$78215,0,1
Ck3a3SoftStartFlt->Y:$78215,1,1
Ck3a3StoStatus->Y:$78215,2,1
Ck3a3ShuntShortFlt->Y:$78215,3,1
Ck3a3OverVoltFlt->Y:$78215,4,1
Ck3a3I2tFlt->Y:$78215,5,1
Ck3a3ShortFlt->Y:$78215,6,1
Ck3a3OverLoadFlt->Y:$78215,7,1

#define Ck3a4PwrFlt           M1028
#define Ck3a4SoftStartFlt     M1029
#define Ck3a4StoStatus        M1030
#define Ck3a4ShuntShortFlt    M1031
#define Ck3a4OverVoltFlt      M1032
#define Ck3a4I2tFlt           M1033
#define Ck3a4ShortFlt         M1034
#define Ck3a4OverLoadFlt      M1035
Ck3a4PwrFlt->Y:$7821D,0,1
Ck3a4SoftStartFlt->Y:$7821D,1,1
Ck3a4StoStatus->Y:$7821D,2,1
Ck3a4ShuntShortFlt->Y:$7821D,3,1
Ck3a4OverVoltFlt->Y:$7821D,4,1
Ck3a4I2tFlt->Y:$7821D,5,1
Ck3a4ShortFlt->Y:$7821D,6,1
Ck3a4OverLoadFlt->Y:$7821D,7,1

// Ch. 1-4 STATUS BITS SET 2, ADC B (STROBE [11:8] = 0)
#define Ck3a1PwrStatus        M1036
#define Ck3a1OverTempFlt      M1037
#define Ck3a1ShuntOverLFlt    M1038
#define Ck3a1PwmFreqFlt       M1039
#define Ck3a1PwmCmdFlt        M1040
#define Ck3a1AmpEna           M1041
Ck3a1PwrStatus->X:$10F0,0,1
Ck3a1OverTempFlt->X:$10F0,2,1
Ck3a1ShuntOverLFlt->X:$10F0,3,1
Ck3a1PwmFreqFlt->X:$10F0,4,1
Ck3a1PwmCmdFlt->X:$10F0,5,1
Ck3a1AmpEna->X:$10F0,6,1

#define Ck3a2PwrStatus        M1042
#define Ck3a2OverTempFlt      M1043
#define Ck3a2ShuntOverLFlt    M1044
#define Ck3a2PwmFreqFlt       M1045
#define Ck3a2PwmCmdFlt        M1046
#define Ck3a2AmpEna           M1047
Ck3a2PwrStatus->X:$10F1,0,1
Ck3a2OverTempFlt->X:$10F1,2,1
Ck3a2ShuntOverLFlt->X:$10F1,3,1
Ck3a2PwmFreqFlt->X:$10F1,4,1
Ck3a2PwmCmdFlt->X:$10F1,5,1
Ck3a2AmpEna->X:$10F1,6,1

#define Ck3a3PwrStatus        M1048
#define Ck3a3OverTempFlt      M1049
#define Ck3a3ShuntOverLFlt    M1050
#define Ck3a3PwmFreqFlt       M1051
#define Ck3a3PwmCmdFlt        M1052
#define Ck3a3AmpEna           M1053
Ck3a3PwrStatus->X:$10F2,0,1
Ck3a3OverTempFlt->X:$10F2,2,1
Ck3a3ShuntOverLFlt->X:$10F2,3,1
Ck3a3PwmFreqFlt->X:$10F2,4,1
Ck3a3PwmCmdFlt->X:$10F2,5,1
Ck3a3AmpEna->X:$10F2,6,1

```

```

#define Ck3a4PwrStatus          M1054
#define Ck3a4OverTempFlt       M1055
#define Ck3a4ShuntOverLFlt     M1056
#define Ck3a4PwmFreqFlt       M1057
#define Ck3a4PwmCmdFlt         M1058
#define Ck3a4AmpEna            M1059
Ck3a4PwrStatus->X:$10F3,0,1
Ck3a4OverTempFlt->X:$10F3,2,1
Ck3a4ShuntOverLFlt->X:$10F3,3,1
Ck3a4PwmFreqFlt->X:$10F3,4,1
Ck3a4PwmCmdFlt->X:$10F3,5,1
Ck3a4AmpEna->X:$10F3,6,1

// Ch. 1-4 STATUS BITS SET 3, ADC B (STROBE [11:8] = 1)
#define Ck3a1I2tint             M1060
#define Ck3a1AdcOffsetFlt      M1061
#define Ck3a1Ready              M1062
Ck3a1I2tint->Y:$10F0,2,1
Ck3a1AdcOffsetFlt->Y:$10F0,4,1
Ck3a1Ready->Y:$10F0,6,1

#define Ck3a2I2tint             M1063
#define Ck3a2AdcOffsetFlt      M1064
#define Ck3a2Ready              M1065
Ck3a2I2tint->Y:$10F1,2,1
Ck3a2AdcOffsetFlt->Y:$10F1,4,1
Ck3a2Ready->Y:$10F1,6,1

#define Ck3a3I2tint             M1066
#define Ck3a3AdcOffsetFlt      M1067
#define Ck3a3Ready              M1068
Ck3a3I2tint->Y:$10F2,2,1
Ck3a3AdcOffsetFlt->Y:$10F2,4,1
Ck3a3Ready->Y:$10F2,6,1

#define Ck3a4I2tint             M1069
#define Ck3a4AdcOffsetFlt      M1070
#define Ck3a4Ready              M1071
Ck3a4I2tint->Y:$10F3,2,1
Ck3a4AdcOffsetFlt->Y:$10F3,4,1
Ck3a4Ready->Y:$10F3,6,1

// CH. 1-4 ADC B MIRROR WORDS
#define Ch1Mirror2               M1072
#define Ch1Mirror3               M1073
Ch1Mirror2->X:$10F0,0,8
Ch1Mirror3->Y:$10F0,0,8

#define Ch2Mirror2               M1074
#define Ch2Mirror3               M1075
Ch2Mirror2->X:$10F1,0,8
Ch2Mirror3->Y:$10F1,0,8

#define Ch3Mirror2               M1076
#define Ch3Mirror3               M1077
Ch3Mirror2->X:$10F2,0,8
Ch3Mirror3->Y:$10F2,0,8

#define Ch4Mirror2               M1078
#define Ch4Mirror3               M1079
Ch4Mirror2->X:$10F3,0,8
Ch4Mirror3->Y:$10F3,0,8

```

CK3A 샘플 PLC 4 모터

```
// MOTORS 1-4 CK3A PLC
OPEN PLC 1 CLEAR
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
// STATUS BITS SET 2
I7206 = (I7206 & $FFF0FF) | $0
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
Ch1Mirror2 = Ck3a1AdcBData
Ch2Mirror2 = Ck3a2AdcBData
Ch3Mirror2 = Ck3a3AdcBData
Ch4Mirror2 = Ck3a4AdcBData
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// STATUS BITS SET 3
I7206 = (I7206 & $FFF0FF) | $100
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
Ch1Mirror3 = Ck3a1AdcBData
Ch2Mirror3 = Ck3a2AdcBData
Ch3Mirror3 = Ck3a3AdcBData
Ch4Mirror3 = Ck3a4AdcBData
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// DC BUS VOLTAGE
I7206 = (I7206 & $FFF0FF) | $200
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1DcBus = Ck3a1AdcBData * 4 ENDIF
IF(Ck3a2Ready = 0) Ck3a2DcBus = Ck3a2AdcBData * 4 ENDIF
IF(Ck3a3Ready = 0) Ck3a3DcBus = Ck3a3AdcBData * 4 ENDIF
IF(Ck3a4Ready = 0) Ck3a4DcBus = Ck3a4AdcBData * 4 ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// TEMPERATURE
I7206 = (I7206 & $FFF0FF) | $300
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1Temp = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2Temp = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3Temp = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4Temp = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// FIRMWARE VERSION
I7206 = (I7206 & $FFF0FF) | $600
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1FwVer = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2FwVer = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3FwVer = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4FwVer = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// CURRENT RATING
I7206 = (I7206 & $FFF0FF) | $700
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1CurRating = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2CurRating = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3CurRating = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4CurRating = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// BUS DISCHARGE CONTROL
Ck3a0BusDisState = (I7206 & $20) / 32
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0BusDisCtrl = 1 AND Ck3a0BusDisState = 0)
    I7206 = (I7206 & $FFFFDF) | $20
ENDIF
IF(Ck3a0BusDisCtrl = 0 AND Ck3a0BusDisState = 1)
    I7206 = (I7206 & $FFFFDF) | $0
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// FAN CONTROL
Ck3a0FanState = (I7206 & $20000) / 131072
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0FanCtrl = 1 AND Ck3a0FanState = 0)
    I7206 = (I7206 & $FDFFFF) | $20000
ENDIF
IF(Ck3a0FanCtrl = 0 AND Ck3a0FanState = 1)
    I7206 = (I7206 & $FDFFFF) | $0
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
```

```

// DYNAMIC BRAKE CONTROL
Ck3a0DbState = (I7206 & $100000) / 1048576
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0DbCtrl = 1 AND Ck3a0DbState = 1)
    I7206 = (I7206 & $FFFFFF) | $0
ENDIF
IF(Ck3a0DbCtrl = 0 AND Ck3a0DbState = 0)
    I7206 = (I7206 & $FFFFFF) | $100000
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// CLEAR FAULT
IF(Ck3a0ClrF = 1)
    I7206 = (I7206 & $FFFFFF) | $10000
    I5111 = 10 * 8388608 / I10 WHILE (I5111 > 0) ENDW
    I7206 = (I7206 & $FFFFFF) | $0
    I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
    Ck3a0ClrF = 0
ENDIF

// NOT READY (POWER OFF)?
IF(Ck3a1Ready = 1)
    Ck3a1DcBus = 0
    Ck3a1Temp = 0
    Ck3a1FwVer = 0
    Ck3a1CurRating = 0
ENDIF

IF(Ck3a2Ready = 1)
    Ck3a2DcBus = 0
    Ck3a2Temp = 0
    Ck3a2FwVer = 0
    Ck3a2CurRating = 0
ENDIF

IF(Ck3a3Ready = 1)
    Ck3a3DcBus = 0
    Ck3a3Temp = 0
    Ck3a3FwVer = 0
    Ck3a3CurRating = 0
ENDIF

IF(Ck3a4Ready = 1)
    Ck3a4DcBus = 0
    Ck3a4Temp = 0
    Ck3a4FwVer = 0
    Ck3a4CurRating = 0
ENDIF
ENDIF
CLOSE

```


OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ASIA PACIFIC PTE. LTD.

No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON ELECTRONICS LLC

2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2022 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. 0050-E1-01