2 축 컨트롤러 SNC-2000

Firmware V1.0

(주)세주에프에이

서울시 금천구 가마산로 96, 310 (가산동, 대륭테크노타운 8 차)

*알 림

본 사용 설명서의 내용과 제품의 기능은 사전 통보 없이 변경될 수 있습니다.
본 제품을 이 자료에서 설명한 용도 외에 사용할 경우, 폐사에서는 어떠한 책임도 지지 않으므로 주의하시기 바랍니다.
본 제품은 ㈜세주에프에이 고유 기술을 사용하여 개발된 제품으로 저작권법에 의한 보호를 받고 있습니다.
따라서 본 제품 (제품에 대한 아이디어나 설명서 및 기타 포함)의 어떠한 부분도 사전에 폐사와의 문서 동의 없이 복사되거나 변경, 재생산 할 수 없으며, 또한 다른 언어로도 번역될 수 없습니다.

*사용 전에 안전을 위한 주의사항을 반드시 읽고 사용하여 주십시오

★ 인명이나 재산상에 영향이 큰 기기(예:원자력 제어, 의료기기, 차량, 철도, 항공, 연소장치, 오락기기 등 또는 안전장치)에 사용할 경우에는 반드시 2중으로 안전장치를 부착한 후 사용하여 주십시오.
화재, 인사사고, 재산상의 막대한 손실이 발생할 수 있습니다.
★ 기계와 조합하여 운전을 할 경우에는, 언제라도 비상정지 할 수 있는 상태로 해주십시오.
상해를 입을 수 있습니다.
★ 콘트롤러 내부에는 절대 손을 대지 마십시오. 감전의 우려가 있습니다.
★ 자사 수리 기술자 이외에는 제품을 개조하지 마십시오. 감전이나 화재의 위험이 있습니다.
★ 실외에서 사용하지 마십시오. 제품의 수명이 짧아지는 원인이 되며, 감전의 우려가 있습니다.
★ 반드시 정격/성능 범위 내에서 사용하여 주십시오. 제품의 수명이 짧아지는 원인이 되며 화재의 우려가 있습니다.
★ 입/출력 포트 용량 정격값을 초과하는 부하를 사용하지 마십시오. 절연불량, 접점불량, 접촉불량, 화재 등의 원인이 됩니다.
★ 청소 시 물, 유기용제를 사용하지 마시고, 물기가 없는 마른 수건으로 청소하십시오. 감전이나 화재의 우려가 있습니다.
★ 가연성 가스, 폭발성 가스, 습기, 직사광선, 복사열, 진동, 충격이 있는 장소에서 사용하지 마십시오.
제품이 손상되거나 오동작할 우려가 있습니다.
★ 본 제품의 내부로 먼지나 배선 찌꺼기가 유입되지 않도록 하여 주십시오.
화재나 장치 고장의 우려가 있습니다.

★ 케이블을 손상하거나, 강하게 당기거나, 무리한 힘을 주거나 또는 무거운 것을 올려 놓는 등의 행위는 절대 금하여 주십시오. 감전, 제품의 동작정지, 화재발생 등의 우려가 있습니다.

★ 배선은 바르고 정확하게 설치해 주십시오. 모터 폭주, 부상, 고장 등의 우려가 있습니다.

보증정보(Warranty Information)

당사에서 제품 또는 라이선스를 구매한 원 구매자에 대한 보증(Warranty)은 아래와 같습니다.

보증조건

㈜세주에프에이(이하 당사) 제품의 고객보증기간은 1 년으로 그 기간 내에 제품 자체 문제에 대한 지원을 받을 수 있습니다.

당사는 다음의 경우에 야기된 제품 훼손에 대하여는 보증기간 내에도 보증지원을 책임지지 않습니다.
"제품매뉴얼에서 명기된 설치안내 사항과 디지털 입/출력 정격을 고려하지 않고 사용한 경우"
"외부 인위적 요인이나 제품이 설치된 환경적 요인에 의해 제품에 이상이 생긴 경우"
원 구매자는 제품보증기간 내에 발생한 제품문제사항을 당사로 즉시 연락 바랍니다.
보증기간 이내에 원 구매자로부터 제품문제가 제기되면 구매자 지역에서 제품문제를 진단하거나 당사로 제품을 배송 받아 직접 확인하고 제품에 대한 수리 및 교체서비스를 지원합니다. 만약 구매한 제품이 보증기간을 초과하거나 제품문제가 지원조건에 해당되지 않는 경우 수리/교체 및 배송에 대한 관련비용을 원 구매자가 부담해야 합니다.

당사는 아래에 명기된 "보증조건이행 제한사항"이 현행 응용법에 위배되지 않는 한 그 어떤 경우의 법적인 요구와 주장[계약 유무에 관계없이, 배상, 보증, 불법행위(과실 및 무과실책임포함)]에 대하여 손실에 대한 책임을 지지 않으며, 원 구매자의 사업중단, 사용상의 손실, 수익문제를 포함한 구매제품에 대한 특례적, 간접적, 우발적, 법적, 회사정리로 인한 결과적인 피해나 손실에 대한 책임을 지지 않습니다

보증조건이행 제한사항

당사는 상기된 "보증조건" 불이행에 대한 고객의 요구사항을 제외하고 판매된 본제품과 관련되거나 초래된 손실, 피해, 또는 지출에 대하여 원 구매자와 그 관련자, 대리인, 또는 계약자가 주장하는 어떠한 요구에 대해서도 책임을 지지 않습니다.
상기된 "보증조건"은 원 구매자의 독점적 권리입니다. 당사는 상기된 보증조건 외에 명시 또는 묵시적인 여타 다른 보증조건(특정목적을 위한 제품수정 및 제품매매상의 묵시적인 보증조건, 법적 침해가 없는 보증조건도 포함) 에 대한 이행을 거부합니다.
당사 제품 동작 및 유지에 대한 지침사항을 정확하게 따르지 않고 교체, 사고, 오용, 남용, 부주의 등으로 인한 제품문제에는 "보증조건"에 적용되지 않습니다. 원 구매자의 시스템 디자인에서 당사의 인력과 대리인에 의해 제공된 기술적인 도움은 하나의 제안이며 추천사항은 아닙니다. 그 제안의 실행결정에 대한 책임은 원 구매자에게 있고 원 구매자에 의해 테스트 되어야 합니다. 고객의 목적에 맞는 제품과 그 사용의 적합성을 결정하는 것은 원 구매자의 책임입니다.
"보증조건"에서 기술된 내용에 따라 실제로 적용돼야 합니다. 대리점 또는 다른 독립체, 당사 또는 여타 회사의 개인이나 직원은 그 어떤 이유로도 "보증조건"의 내용을 개정, 수정, 또는 확장할 수 있는 권한을 가지지 않습니다.

1. 특 징 - 7 -

2. 시스템 구성 - 8 -

2.1. 모듈구성 - 8 -

2.2. 기본시스템 구성 - 11 -

3. 기본 사양 - 12 -

3.1. 전원 - 12 -

3.2. 디지털 입력 - 12 -

3.3. 디지털 출력 - 12 -

3.4. 릴레이 출력(확장보드 추가시) - 12 -

3.5. 아날로그 입력(확장보드 추가시) - 12 -

3.6. 아날로그 출력(확장보드 추가시) - 13 -

3.7. 통신 사양(RS485, RS232) - 13 -

3.8. USB, LAN - 13 -

3.9. 설치 환경 - 13 -

4. 입출력 사양 - 14 -

4.1. 제어 전원 입력 - 14 -

4.2. 통신 신호(RS-485, HMI 용) - 14 -

4.3. 통신 신호(RS-232C) - 14 -

4.4. X 축 서보 컨넥터 - 15 -

4.5. Y 축 서보 컨넥터 - 16 -

4.6. 메인 입/출력 컨넥터 - 17 -

4.7. 축관련 입/출력 컨넥터 - 18 -

4.8. 엔코더 입/출력 컨넥터 - 19 -

5. 접속 TYPE - 20 -

5.1. IN TYPE-1 - 20 -

5.2. IN TYPE-2 - 20 -

5.3. IN TYPE-3 - 20 -

5.4. IN TYPE-4 - 20 -

5.5. OUT TYPE-1 - 21 -

5.6. OUT TYPE-2 - 21 -

5.7. OUT TYPE-3 - 21 -

6. 프로그램 코드 및 데이터 - 22 -

7. 파라미터 범위 - 61 -

8. 모드버스 인터페이스 - 66 -

8.1. 초기설정 - 66 -

8.2. 주소범위 - 66 -

8.3. 항목별 주소 - 68 -

1. 특 징

SNC-2000 2 축 제어기는 1 대의 제어기로 동시에 모션컨트롤과 시퀀스제어가 가능하며, 외부 입/출력과 연동한 고속, 고정도 위치제어시스템이 가능합니다.
또한, 각종 I/O(디지털, 아날로그 입출력)를 추가모듈로 확장할 수가 있으며,
모드버스 제어가 가능하므로, Flexible 한 화면(HMI)구성이 가능합니다.

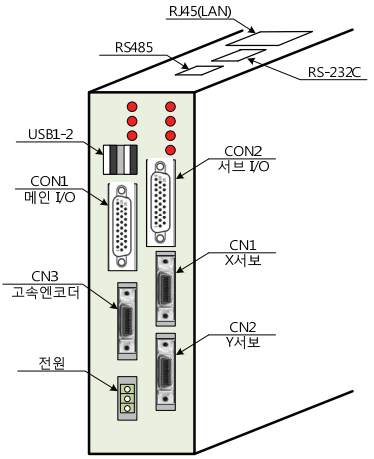
SNC-2000 2 축 제어기의 특징은 다음과 같습니다.

- 부동소수점 연산기능 내장된 고속 CORTEX-M7 마이컴사용.
- G-Code, M-Code 를 이용하여 자유로운 이송위치 프로그램가능.
- FPGA 를 이용한 고속카운터 및 Flying Cut 기능.
- 외부 입/출력 제어가 가능.
- USB, LAN 을 이용하여 손쉬운 프로그램 다운로드/업로드.
- 상용 HMI 와 연결이 가능토록 MODBUS 표준 어드레스 제공.
- 프로그램의 연속/스텝 구동으로 손쉬운 디버깅.

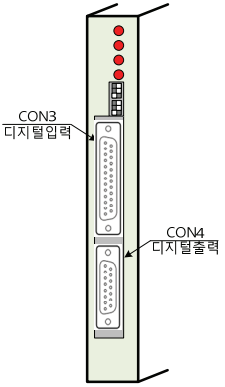
2. 시스템 구성

2.1. 모듈구성

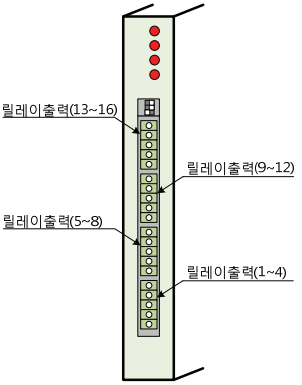
각종 모듈의 외관표시입니다.



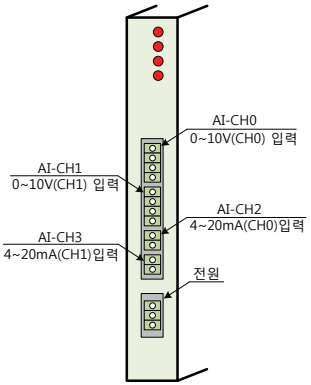
【메인보드】



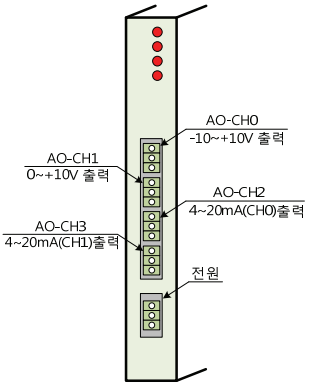
【DIO 보드】



【릴레이보드】

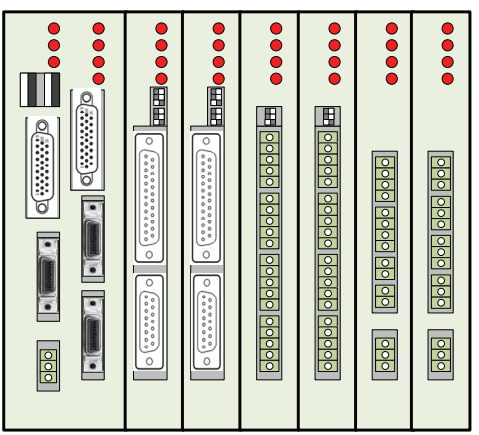


【AI 보드】



【AO 보드】

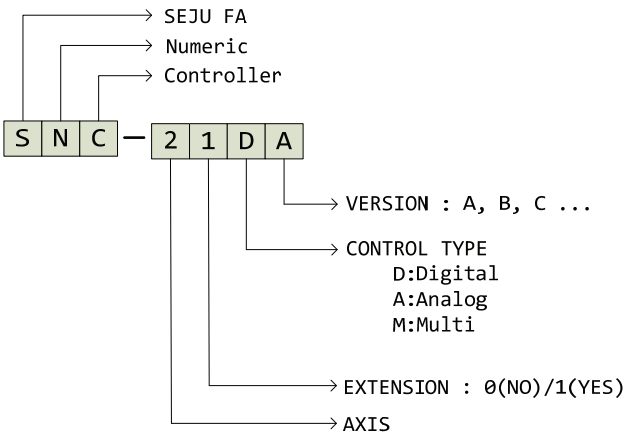
메인보드 1 개 만로도 동작시킬 수 있으며, 확장 시에는 아래 그림과 같이 DIO 보드(2 개) + 릴레이보드(2 개) + AI 보드(1 개) + AO 보드(1 개) 까지 확장시킬 수 있습니다.
확장보드는 순서와 관계없이 임의의 순서로 장착/탈착이 가능합니다.



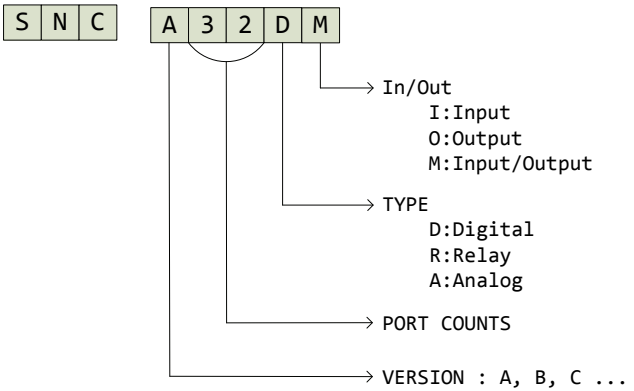
【최대 확장 시】

2.2. 모듈형식

[본체 형식]



[확장 형식]



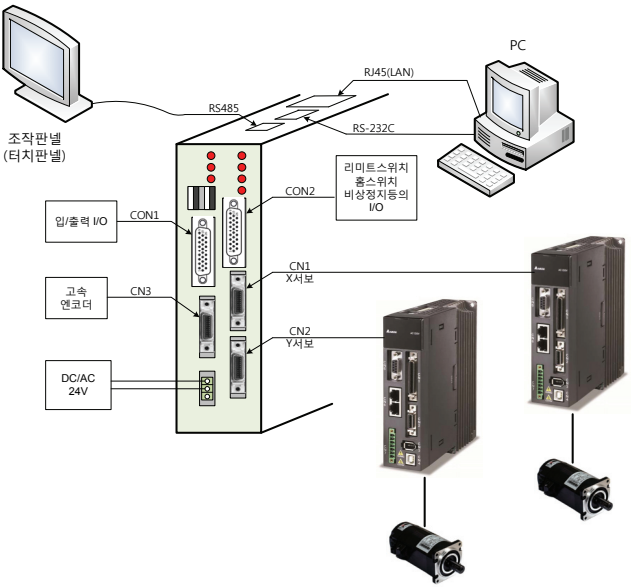
(1) SNC-A32DM : 디지털입력(20), 디지털출력(12)

(2) SNC-A16RO : 릴레이 출력(16)

(3) SNC-A04AI : 아날로그 4ch 입력

(4) SNC-A04AO : 아날로그 4ch 출력

2.3. 기본시스템 구성



【기본시스템】

메인보드 1 대 만으로 운용되는 시스템입니다.
프로그램 코드의 다운로드/업로드는 메인보드 1 대 만인 독립시스템의 경우, USB 메모리카드로 가능하며, PC 를 연결하면 모니터링 기능(예정)및 프로그램디버깅기능(예정)이 가능합니다.

3. 기본 사양

3.1. 전원

전원 사양	
입력 전원	AC24V 50/60Hz, DC24V
소비 전력	최대 VA

3.2. 디지털 입력

디지털 입력	
입력 방식	Opto-Isolation
입력 수	8 개 (확장보드 추가 시 : 20 개/1 보드)
신호 전원	AC24V or DC24V

3.3. 디지털 출력

디지털 출력	
출력 방식	오픈 컬렉터
출력 수	8 개 (확장보드 추가 시 : 12 개/1 보드)
허용 전류	최대 120mA

3.4. 릴레이 출력(확장보드 추가 시)

디지털 출력	
출력 방식	릴레이 접점
출력 수	16 개
릴레이 접점 정격	250V, 5A

3.5. 아날로그 입력(확장보드 추가 시)

아날로그 입력	
10V	전압입력 2 개
4~20mA	2 개(센서용 전원 24V 내장)
편차 보정	펌웨어

3.6. 아날로그 출력(확장보드 추가 시)

아날로그 출력	
0~+10V	전압
-10V~+10V	전압
4~20mA	전류
설정방식	펌웨어

3.7. 통신 사양(RS485, RS232)

통 신	
통신 방법	RS485 (반이중 방식) 1 채널, RS232(전이중 방식) 1 채널
통신 속도	4800, 9600, 19200, 38400 BPS Parity None, Data 8bit, Stop 1bit
통신 거리	최대 1.2kM
권장 케이블	BELDEN 9842 or 8761

3.8. USB, LAN

통 신	
USB	Full Speed(12Mbit/sec) TYPE-A(2 CH)
LAN	100Base-T(RJ-45) 커넥터 Fast Ethernet(100Mbit/sec)

3.9. 설치 환경

설치 환경	
설치 위치	옥내용
동작 온도	0 ~ 40℃
보관 온도	-20 ~ 75℃
동작 습도	무결로 상태 20~80%

4. 입출력 사양

4.1. 제어 전원 입력

명칭	Pin Name	Function	Type
Power	FG	FIELD GROUND (CASE SHIELD)	
	L2(-)	AC24V(-) or DC 0V	
	L1(+)	AC24V(+) or DC+24V	

4.2. 통신 신호(RS-485, HMI 용)

명칭	Pin Name	Function	Type
COMM1	RS485-	SYSTEM BUS TRX-	RS-485
	RS485+	SYSTEM BUS TRX+	

※ 통신사양 (반이중, HMI 용)

① 통신방식 : RS-485

② 통신속도 : 4800, 9600, 19200, 38400 BPS, N, 8, 1

③ 통신프로토콜 : MODBUS ASCII MODE

④ 권장 통신 케이블 : BELDEN 9842 or 8761

4.3. 통신 신호(RS-232C)

명칭	Pin Name	Function	Type
COMM2.	TX	송신신호	RS-232C
	RX	수신신호	
	GND	0V	

※ 통신사양 (전이중)

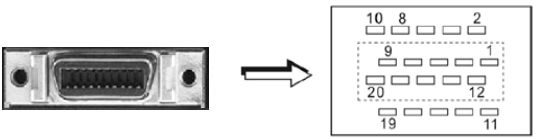
① 통신방식 : RS-232C

② 통신속도 : 4800, 9600, 19200, 38400 BPS, N, 8, 1

③ 통신프로토콜 : TBD

④ 권장 통신 케이블 : BELDEN 9842 or 8761

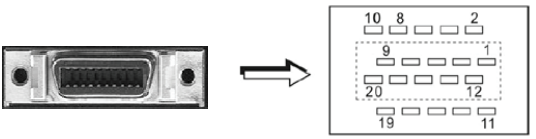
4.4. X 축 서보 커넥터



명칭	Pin No	Name	기능	I/O
X 축 서보	1	DIR P	서보 디렉션 +	O③
	2	PULSE P	서보 펄스 +	O③
	3	DIR N	서보 디렉션 -	O③
	4	PULSE N	서보 펄스 -	O③
	5	Z ENCODER P	Z 엔코더 피드백 +	I③
	6	A ENCODER P	A 엔코더 피드백 +	I③
	7	Z ENCODER N	Z 엔코더 피드백 -	I③
	8	A ENCODER N	A 엔코더 피드백 -	I③
	9	B ENCODER N	B 엔코더 피드백 -	I③
	10	B ENCODER P	B 엔코더 피드백 +	I③
	11	+24V	외부 24V 입력	PI
	12	+24V	외부 24V 입력	PI
	13	READY	서보 READY 상태	I②
	14	SERVO ON	서보 온	O①
	15	ALARM	서보 알람 상태	I②
	16	ALARM RESET	서보 알람 리셋	O①
	17	IN POSITION	서보 인 포지션	I②
	18	OUT_COM	14, 16 공통접지	COM
	19	GND	0V	G
	20	GND	0V	G

I : INPUT, O : OUTPUT [①②③은 4.접속 TYPE 별 분류 참조]
PI : 24V 입력, G : GROUND
COM : SERVO ON, ALARM RESET 출력 공통단자

4.5. Y 축 서보 커넥터



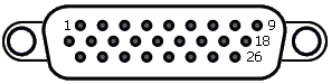
명칭	Pin No	Name	기능	I/O
Y 축 서보	1	DIR P	서보 디렉션 +	O③
	2	PULSE P	서보 펄스 +	O③
	3	DIR N	서보 디렉션 -	O③
	4	PULSE N	서보 펄스 -	O③
	5	Z ENCODER P	Z 엔코더 피드백 +	I③
	6	A ENCODER P	A 엔코더 피드백 +	I③
	7	Z ENCODER N	Z 엔코더 피드백 -	I③
	8	A ENCODER N	A 엔코더 피드백 -	I③
	9	B ENCODER N	B 엔코더 피드백 -	I③
	10	B ENCODER P	B 엔코더 피드백 +	I③
	11	+24V	외부 24V 입력	PI
	12	+24V	외부 24V 입력	PI
	13	READY	서보 READY 상태	I②
	14	SERVO ON	서보 온	O①
	15	ALARM	서보 알람 상태	I②
	16	ALARM RESET	서보 알람 리셋	O①
	17	IN POSITION	서보 인 포지션	I②
	18	OUT_COM	14, 16 공통접지	COM
	19	GND	0V	G
	20	GND	0V	G

I : INPUT, O : OUTPUT [①②③은 4.접속 TYPE 별 분류 참조]

PI : 24V 입력, G : GROUND

COM : SERVO ON, ALARM RESET 출력 공통단자

4.6. 메인 입/출력 커넥터



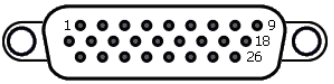
명칭	Pin No	Name	기능	I/O
메인 I/O	1	INPUT_COM1	1~4 입력 공통단자	COM
	2	INPUT 1	디지털입력 1	I①
	3	INPUT 2	디지털입력 2	I①
	4	INPUT 3	디지털입력 3	I①
	5	INPUT 4	디지털입력 4	I①
	6	INPUT_COM2	5~8, 비상정지입력 공통단자	COM
	7	INPUT 5	디지털입력 5	I①
	8	INPUT 6	디지털입력 6	I①
	9	INPUT 7	디지털입력 7	I①
	10	INPUT 8	디지털입력 8	I①
	11	EMG_INP	비상정지	I①
	12	OUTPUT 1	디지털출력 1	O①
	13	OUTPUT 2	디지털출력 2	O①
	14	OUTPUT 3	디지털출력 3	O①
	15	OUTPUT 4	디지털출력 4	O①
	16	OUTPUT_COM1	1~4 출력 공통단자	COM
	17	OUTPUT 5	디지털출력 5	O①
	18	OUTPUT 6	디지털출력 6	O①
	19	OUTPUT 7	디지털출력 7	O①
	20	OUTPUT 8	디지털출력 8	O①
	21	OUTPUT_COM2	5~8 출력 공통단자	COM
	22	NC		
	23	NC		
	24	NC		
	25	NC		
	26	NC		

I : INPUT, O : OUTPUT [①②③은 4.접속 TYPE 별 분류 참조]

COM : 해당 입력/출력 공통단자

NC : NOT CONNECT(연결금지)

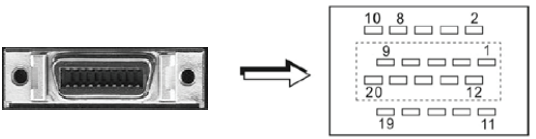
4.7. 축관련 입/출력 커넥터



명칭	Pin No	Name	기능	I/O
축관련 I/O	1	X_COM	X 입력 공통단자	COM
	2	X + LIMIT	X 축 + 리미트 스위치	I①
	3	X - LIMIT	X 축 - 리미트 스위치	I①
	4	X HOME	X 축원점 스위치	I①
	5	Y_COM	Y 입력 공통단자	COM
	6	Y + LIMIT	Y 축 + 리미트 스위치	I①
	7	Y - LIMIT	Y 축 - 리미트 스위치	I①
	8	Y HOME	Y 축 원점 스위치	I①
	9	X ESTOP	X 비상정지	I①
	10	Y ESTOP	Y 비상정지	I①
	11	NC		
	12	NC		
	13	NC		
	14	NC		
	15	NC		
	16	NC		
	17	NC		
	18	GND	0V	G
	19	GND	0V	G
	20	+5V	+5V	PO
	21	+5V	+5V	PO
	22	Y MPG_POWER	Y MPG 풀업 전원	*PI
	23	Y MPG A	Y MPG 엔코더 A	I③or④
	24	Y MPG /A	Y MPG 엔코더 /A	I③or④
	25	Y MPG B	Y MPG 엔코더 B	I③or④
	26	Y MPG /B	Y MPG 엔코더 /B	I③or④

I : INPUT, O : OUTPUT [①②③④는 4.접속 TYPE 별 분류 참조]
COM : 해당 입력/출력 공통단자
*PI : 오픈 컬렉터 사용시 24V 입력
PO : 5V 출력, G : GROUND
NC : NOT CONNECT(연결금지)

4.8. 엔코더 입/출력 커넥터



명칭	Pin No	Name	기능	I/O
엔코더 I/O	1	ENCODER1 A	엔코더 1 입력 A	I③or④
	2	ENCODER1 /A	엔코더 1 입력 /A	I③or④
	3	ENCODER1 B	엔코더 1 입력 B	I③or④
	4	ENCODER1 /B	엔코더 1 입력 /B	I③or④
	5	ENC1 POWER	엔코더 1 전원	*PI
	6	GND	0V	G
	7	+24V	+24V	PO
	8	+24V	+24V	PO
	9	GND	0V	G
	10	+5V	+5V	PO
	11	ENC2 POWER	엔코더 2 전원	*PI
	12	ENCODER2 A	엔코더 2 입력 A	I③or④
	13	ENCODER2 /A	엔코더 2 입력 /A	I③or④
	14	ENCODER2 B	엔코더 2 입력 B	I③or④
	15	ENCODER2 /B	엔코더입력 /B	I③or④
	16	EN1	FPGA 출력 1(TBD)	O②
	17	EN2	FPGA 출력 2(TBD)	O②
	18	EN3	FPGA 출력 3(TBD)	O②
	19	EN4	FPGA 출력 4(TBD)	O②
	20	GND	0V	G

I : INPUT, O : OUTPUT [①②③④는 4.접속 TYPE 별 분류 참조]

*PI : 오픈 컬렉터 사용시 24V 입력

PO : 24V 출력, G : GROUND

TBD(To Be Determined, 향후 결정)

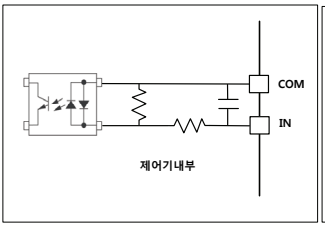
5. 접속 TYPE

5.1. IN TYPE-1

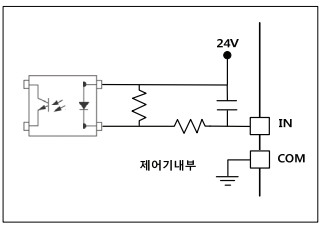
극성에 관계없이 COM(공통단자)사이에 외부 24V 전압을 인가하면 ON, 그렇지 않으면 OFF.

5.2. IN TYPE-2

GND 와 연결하면 ON, 그렇지 않으면 OFF.



IN TYPE-1



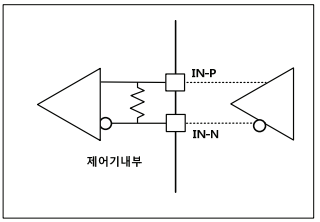
IN TYPE-2

5.3. IN TYPE-3

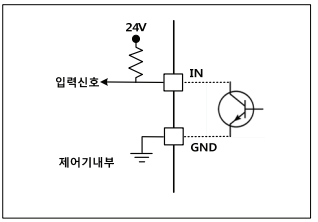
5V 라인드라이버 입력입니다.(피드백 엔코더 등)

5.4. IN TYPE-4

오픈 컬렉터 입력입니다.(엔코더, MPG 등)



IN TYPE-3



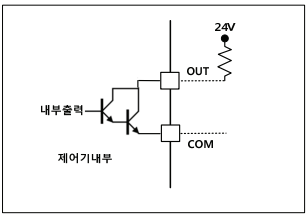
IN TYPE-4

5.5. OUT TYPE-1

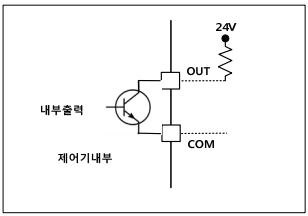
달링턴 오픈 컬렉터 타입으로, 최대 150mA (통상 120mA) 까지 가능합니다.

5.6. OUT TYPE-2

보통타입의 오픈 컬렉터 타입으로, 최대 10mA (통상 5mA) 까지 가능합니다.



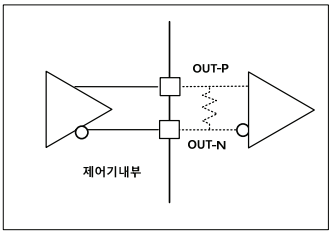
OUT TYPE-1



OUT TYPE-2

5.7. OUT TYPE-3

5V 라인드라이버 출력입니다.(DIR, PULSE 등)



OUT TYPE-3

6. 프로그램 코드 및 데이터

이송 데이터 mm 값의 이송 펄스는, 파라미터 서보 1 회전펄스, 기어비 분자, 분모값에 따라 다음과 같이 계산됩니다.

0.01mm 당 펄스수 = 서보 1 회전펄스 / 감속비(분자/분모) / 볼스크류

(예)

서보 1 회전펄스 : 10000

감속비 분자 : 1

감속비 분모 : 2

볼스크류 : 1000(10.00mm)

10000 / (1/2) / 1000 = **20** 펄스/0.01mm

따라서, 2mm 이송펄스는, 20 X 100 X 2 = 4000 펄스가 됩니다.

프로그램에 사용되는 데이터는 mm ➔ 펄스계산에 맞게 변환되며, 펄스값 자체로도 사용될 수 있습니다. 즉, mm 입력모드와 펄스입력모드 두가지를 사용할 수 있으며, 프로그램 데이터로는 최종 펄스 데이터를 사용합니다.

프로그램에 쓰이는 레지스터는, 일반 레지스터 10 개(REG0~9), 카운트 레지스터 2 개(C-REG0, C-REG1), REG_X, REG_Y 를 사용하며, 레지스터간 대입, 비교 등의 연산이 가능합니다.

5-1 표는 G-CODE, M-CODE 의 개략이며, 자세한 설명은, 5-2 표의 코드별 리스트에 기술되어있습니다.

5-1 코드

코드	번호	내용
G	00	위치이동
	01	위치이동
	02	포인트이동
	03	화면목표이동
	04	현재위치 0 설정
	05	화면목표이동
	06	이동완료면 JUMP
	08	시간지연
	09	시간지연
	20	가감속 시간설정
	21	속도(PPS)구동
	22	정지
	23	속도(PPS)구동
	30	원점이동
	31	속도(PPS)설정
	40	운전방향설정
	80	X 축 ABS/INC

코드	번호	내용
G	81	Y 축 ABS/INC
	90	서브루틴 CALL

코드	번호	내용
M	00	프로그램 일시정지
	01	서브루틴 개시점
	02	서브루틴 RET
	03	JUMP
	04	LOOP START
	05	LOOP STOP
	10	REGn = X 축 비교
	11	REGn > X 축 비교
	12	REGn < X 축 비교
	13	REGn ≥ X 축 비교
	14	REGn ≤ X 축 비교
	15	REGn = Y 축 비교
	16	REGn > Y 축 비교
	17	REGn < Y 축 비교
	18	REGn ≥ Y 축 비교
	19	REGn ≤ Y 축 비교
	20	X 축 = N 비교
	21	X 축 > N 비교
	22	X 축 < N 비교
	23	X 축 ≥ N 비교
	24	X 축 ≤ N 비교
	25	Y 축 = N 비교
	26	Y 축 > N 비교
	27	Y 축 < N 비교
	28	Y 축 ≥ N 비교
	29	Y 축 ≤ N 비교
	30	X 축 N 대입
	31	Y 축 N 대입
	32	X, Y 축 상수 대입
	33	REG0 ← REGn
	34	REGn ← REG0

코드	번호	내용
M	35	REGn ← N
	36	X 축 ← REGn
	37	Y 축 ← REGn
	38	X, Y 축 ← REGn
	40	REGn ← X 축
	41	REGn ← Y 축
	42	REGn ← X, Y 축
	43	REGn 가감산
	44	REGn 가감산
	45	카운트 레지스터 ← N
	46	REGn 가산
	47	REGn 감산
	48	REGn N 가감산
	49	REGn+1
	50	REGn = C0 비교
	51	REGn > C0 비교
	52	REGn < C0 비교
	53	REGn ≥ C0 비교
	54	REGn ≤ C0 비교
	55	REGn = C1 비교
	56	REGn > C1 비교
	57	REGn < C1 비교
	58	REGn ≥ C1 비교
	59	REGn ≤ C1 비교
	60	X 이동
	61	Y 이동
	62	X, Y 이동
	65	REGn X 이동
	66	REGn Y 이동
	67	REGn X,Y 이동
	70	입력포트 ON 상태 확인
	71	입력포트 OFF 상태 확인
	72	입력포트 ON 대기
	73	입력포트 OFF 대기
	75	출력포트 ON
	76	출력포트 OFF
	77	버튼대기

코드	번호	내용
M	78	버튼 상태 확인
	80	고속카운터 대기
	81	고속카운터 목표
	82	고속카운터 스타트
	83	고속카운터 스톱
	84	고속카운터 DIR
	85	고속카운터 X Reg
	86	고속카운터 Y Reg
	87	고속카운터 XY Reg
	90	아날로그 입력
	91	아날로그 출력

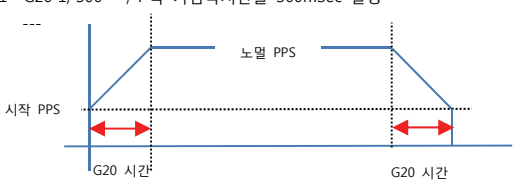
5-2 표

코드	내 용 설 명	1OP	2OP
G00	**위치이동(이동완료까지 대기)** 지정된 데이터만큼 이동한다. (사용예) 현재위치가 **X=10000, Y=10000** 일 때, **G00 2000, -3000** 이송명령은 이송모드에 따라 다음과 같이 진행한다. [ABS-모드] **X** : CCW(-)방향으로 8000 펄스가 출력되며, 결과적으로 위치 **2000** **Y** : CCW(-)방향으로 13000 펄스가 출력되며, 결과적으로 위치 **-3000** [INC-모드] **X** : CW(+)방향으로 2000 펄스가 출력되며, 결과적으로 위치 **12000** **Y** : CCW(-)방향으로 3000 펄스가 출력되며, 결과적으로 위치 **7000** X, Y 축별로 모드선택이 가능하다.(즉, X:ABS, Y:INC) **프로그램 예제** **G00 10000 10000** >> X10000 Y10000 으로 이동 **M75 0** >> 이송완료 후 0 번출력 ON **G08 200** >> 2 초 딜레이 **M76 0** >> 0 번출력 OFF **END**	X 축이송 -8388000 ~ +8388000	Y 축이송 -8388000 ~ +8388000

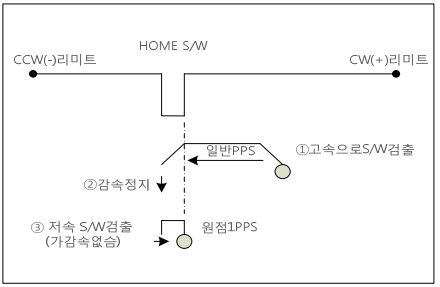
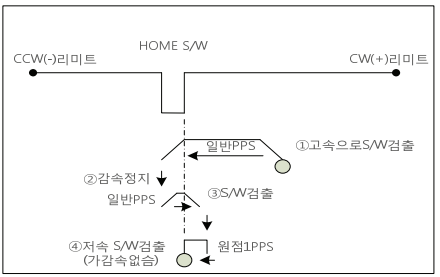
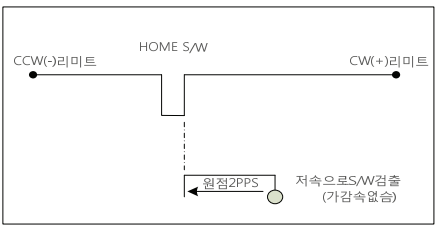
코드	내 용 설 명	1OP	2OP
G01	**위치이동(프로그램 다음스텝 진행)** 위 G00 와 같은 동작이나, 구동완료 시까지 대기하지 않고 다음스텝으로 진행한다. (사용예) --- 10 G00 2000, -3000 ← 구동완료 때까지 대기 ---. 20 G01 2000, -3000 --- ← I/O 시퀀스제어동작 가능 **프로그램 예제** **G01 10000 10000** >> X10000 Y10000 으로 이동합니다 **M75 0** >> 이송시작 후 0 번출력 ON 합니다. **G08 200** >> 2 초 딜레이합니다. **M76 0** >> 0 번출력 OFF 합니다. **END**	X 축이송 -8388000 ~ +8388000	Y 축이송 -8388000 ~ +8388000
G02	**포인트이동** 지정된 포인트로 이동한다. (사용예) ABS 모드일 때, P0 = 2000, 0 P1 = 2000, 2000 P2 = 0, 2000 P4 = 0,0 일경우, 10 G02 0 11 G02 1 12 G02 2 13 G02 3 이송명령은 사각형 움직임이 된다. ABS, INC 모드가능, 완료까지 대기. **프로그램 예제** **G02 0** >> 0 번위치로 이동합니다 **G02 1** >> 0 번위치 이동 완료 후 1 번위치로 이동합니다. **G08 200** >> 1 번위치 이동 완료 후 2 초간 딜레이합니다. **G02 2** >> 1 번위치 이동 완료 후 2 번위치로 이동합니다. **END**	포인트 번호 0~99	미사용

코드	내 용 설 명	1OP	2OP
G03	**화면위치이동(이동완료까지 대기)** 유저 화면상에 입력된 데이터로 이동 목표는 HMI 상의 MODBUS 주소 X(40001), Y(41001)에 입력되는 데이터 파라미터 0~2 는, 0 : X 축, Y 축 동시이동 1 : X 축 이동(Y 는 정지) 2 : Y 축 이동(X 는 정지) **프로그램 예제** **G03 0** >> X, Y 가 Modbus 주소 X(40001), Y(41001)에 입력된 위치로 이동합니다. **G03 1** >> X 가 Modbus 주소 X(40001)에 입력된 위치로 이동합니다. **G03 2** >> Y 가 Modbus 주소 Y(41001)에 입력된 위치로 이동합니다. **END**	XY,X,Y 지정 0~2	미사용
G04	**현재위치 0 설정** 현재위치를 0 으로 설정한다. 파라미터 0~2 는, 0 : X 축, Y 축 현재위치를 각각 0 으로 설정 1 : X 축 현재위치를 0 으로 설정 2 : Y 축 현재위치를 0 으로 설정 **프로그램 예제** **G00 10000 10000** >> X, Y 위치를 이동한다 **G04 0** >> X, Y 현재위치를 0 으로 변환한다 **G00 10000 10000** >> X, Y 위치를 이동한다 **G04 1** >> X 현재위치를 0 으로 변환한다 **G08 100** >> 1 초간 딜레이 **G04 2** >> Y 현재위치를 0 으로 변환한다 **END**	XY,X,Y 지정 0~2	미사용
G05	**화면위치이동(프로그램 다음스텝 진행)** 위 G03 과 같은 동작이나, 구동완료 시까지 대기하지 않고 다음스텝으로 진행한다. 파라미터 0~2 는, 0 : X 축, Y 축 동시이동 1 : X 축 이동(Y 는 정지) 2 : Y 축 이동(X 는 정지)	XY,X,Y 지정 0~2	미사용

코드	내 용 설 명	1OP	2OP
G06	**구동완료** X,Y 구동이 완료되었으면 지정된 번호로 프로그램 스텝을 JUMP 한다. 완료되지 않았으면 다음번호로 진행. (사용예) --- 10 G00 2000, 3000 11 M76 3 ; 3 출력 OFF 12 G06 100 ; 구동완료면 JUMP 100 13 M75 3 ; 3 출력 ON 14 M03 11 ; JUMP 11 --- 100 G00 1000, 1000 --- **프로그램 예제** **G01 2000 3000** >> X, Y 가 각각의 위치로 이동합니다. 이송시작과 동시에 다음스텝을 진행합니다. **M76 0** >> 0 번출력을 off 합니다 **G06 6** >> X, Y 가 이송중이라면 다음스텝으로, 그렇지 않다면, 6 번 스텝으로 이동합니다. **M75 0** >> 0 번출력을 ON 합니다 **M03 2** >> 2 번 스텝으로 Jump 합니다. **G00 1000 1000** >> X, Y 가 각각의 위치로 이동합니다. **END**	스텝 0~249	미사용
G08	**시간지연** 지정시간만큼 시간을 지연시킴. 단위는 10mSec(0.01 초) (사용예) --- 10 G08 100 ; 100 X 10mSec = 1 초간 지연 --- **프로그램 예제** **M75 0** >> 0 번출력을 on 합니다. **G08 200** >> 2 초동안 딜레이 합니다. (10mSec 단위) **M76 0** >> 0 번출력을 OFF 합니다. **END**	시간지연 1 ~ 1000000	미사용

코드	내 용 설 명	1OP	2OP
G09	**시간지연** 파라미터 G09 설정값에 설정되어있는 시간만큼 지연시킴. 단위는 10mSec(0.01 초) (사용예) --- 10 G09 ; 10(파라미터값) X 10mSec = 0.1 초간 지연 --- **프로그램 예제** **M75 0** >> 0 번 출력을 ON 합니다. **G09** >> 파라미터의 G09 설정값에 설정되어 있는 값만큼, 시간이 지연됩니다. **M76 0** >> 0 번출력을 OFF 합니다. **END**	미사용	미사용
G20	**가감속시간** 가감속 시간을 지정. 단위는 mSec. 파라미터 1, 0~2 는, 0 : X 축, Y 축 동시설정 1 : X 축 설정 2 : Y 축 설정 (사용예) --- 10 G20 0, 200 ; X 축 가감속시간을 200mSec 설정 11 G20 1, 300 ; Y 축 가감속시간을 300mSec 설정 ---  **프로그램 예제** **G20 0 100** >> X, Y 양축의 가감속시간을 100mSec 로 변경합니다. **G00 10000 10000** >> X, Y 가 지정된 위치만큼 이동합니다. **G20 1 1000** >> X 축의 가감속 시간을 1000mSec 로 변경합니다. **G00 0 0** >> X, Y 가 지정된 위치만큼 이동합니다. **G20 2 1000** >> Y 축의 가감속 시간을 1000mSec 로 변경합니다. **G00 10000 10000** >> X, Y 가 지정된 위치만큼 이동합니다. **END**	XY,X,Y 지정 0~2	시간설정 1 ~ 10000

코드	내 용 설 명	1OP	2OP
G21	**속도이동** 지정 PPS(Pulse/Sec)로 계속 이동하며, G22 명령에 의하여 정지한다. (-)는 CCW, (+)는 CW. 파라미터 0~2 는, 0 : X 축, Y 축 동시이동 1 : X 축 이동(Y 는 정지) 2 : Y 축 이동(X 는 정지) **프로그램 예제** **G21** 0 1000 >> 1000Hz 의 속도로 X, Y 두축을 속도제어 합니다. **G08** 300 >> 3 초간 딜레이 합니다. **G22** 0 >> 속도제어하던 X, Y 두축을 정지합니다. END	XY,X,Y 지정 0~2	펄스설정 -500000 ~ +500000
G22	**속도정지** G21 의 속도이동을 정지한다. 파라미터 0~2 는, 0 : X 축, Y 축 동시정지 1 : X 축 정지 2 : Y 축 정지 **프로그램 예제** **G21** 0 1000 >> 1000Hz 의 속도로 X, Y 두축을 속도제어 합니다. **G08** 300 >> 3 초간 딜레이 합니다. **G22** 0 >> 속도제어하던 X, Y 두축을 정지합니다. END	XY,X,Y 지정 0~2	미사용
G23	**속도이동** n 레지스터 PPS(Pulse/Sec)로 계속 이동하며, G22 명령에의하여 정지한다. (-)는 CCW, (+)는 CW. 파라미터 0~2 는, 0 : X 축, Y 축 동시이동 1 : X 축 이동(Y 는 정지) 2 : Y 축 이동(X 는 정지) **프로그램 예제** **G23 0 1** >> 1 번 레지스터에 입력된 값으로 X, Y 축을 속도제어 한다. **G08 300** >> 3 초간 딜레이 한다. **G22 0** >> 속도제어하던 X, Y 두축을 정지한다. END	XY,X,Y 지정 0~2	레지스터 0~9

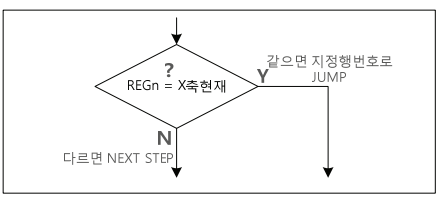
코드	내 용 설 명	1OP	2OP
G30	**원점찾기** 2OP 지정에 의한 원점이동으로 찾는 방식은 3 가지를 제공한다. 방식[0]  방식 [1]  방식 [2]  파라미터 0~2 는, 0 : X 축, Y 축 동시 원점이동 1 : X 축 원점이동 2 : Y 축 원점이동 원점방식 위 3 가지방식에 있어서 사용되는 SW(스위치)의 종류를 지정	XY,X,Y 지정 0~2	원점방식 0~8

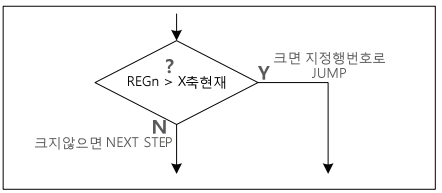
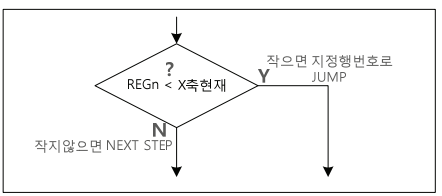
	0, 1, 2 : 홈 SW ON 되고서 처음 Z 엔코더. 3, 4, 5 : 홈 SW ON 6, 7, 8 : Z 엔코더 **프로그램 예제** **G30 0 0** >>X, Y 축을 동시에 0 번 옵션으로 원점찾기를 진행합니다. 각 옵션은 부가 설명창에 예시되어 있습니다.		
G31	**속도 설정** 속도설정 PPS(100~500000) 파라미터 0~2 는, 0 : X 축, Y 축 속도설정 1 : X 축 속도설정 2 : Y 축 속도설정 **프로그램 예제** **G31 0 1000** >> X, Y 양축의 이동속도를 1000Hz 로 변경합니다. **G00 10000 10000** >> X, Y 가 지정된 위치만큼 이동합니다. **G31 1 2000** >> X 축의 이동속도를 2000Hz 로 변경합니다. **G00 0 0** >> X, Y 가 지정된 위치만큼 이동합니다. (X 축 속도 값 변경 확인) **G31 2 2000** >> Y 축의 이동속도를 2000Hz 로 변경합니다. **G00 10000 10000** >> X, Y 가 지정된 위치만큼 이동합니다. (Y 축 속도 값 변경 확인)	XY,X,Y 지정 0~2	속도 100 ~ 500000
G40	**방향 설정** 방향을 설정(0:CW, 1:CCW) 파라미터 0~2 는, 0 : X 축, Y 축 방향 1 : X 축 방향 2 : Y 축 방향 **프로그램 예제** **G40 0 0** >> X, Y 축의 방향을 CW 로 지령합니다 **G00 1000 1000** >> X, Y 축을 각각의 지령값 만큼 이동합니다. **G40 1 1** >> X 축의 방향을 CCW 로 지령합니다 **G00 2000 2000** >> X, Y 축을 각각의 지령값 만큼 이동합니다. **G40 2 1** >> Y 축의 방향을 CCW 로 지령합니다. **G00 3000 3000** >> X, Y 축을 각각의 지령값 만큼 이동합니다. **END**	XY,X,Y 지정 0~2	방향 0~1

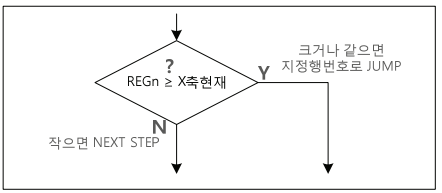
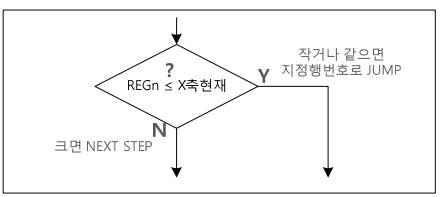
코드	내 용 설 명	1OP	2OP
G80	**X 절대지령/증분지령** X 축에 대한 지령모드 설정. 0:절대지령(ABS), 1:증분지령(INC) **프로그램 예제** **G80 0** >> X 축 지령방식을 절대지령으로 설정합니다. **G00 1000** >> X 축을 절대위치 1000 으로 이동합니다. **G08 100** >> 1 초간 딜레이 합니다. **G00 2000** >> X 축을 절대위치 2000(상대위치 +1000)으로 이동합니다. **G80 1** >> X 축 지령방식을 상대지령으로 설정합니다. **G00 1000** >> X 축을 상대위치 +1000(절대위치 3000)으로 이동합니다. **END**	모드 0~1	미사용
G81	**Y 절대지령/증분지령** Y 축에대한 지령모드설정. 0:절대지령(ABS), 1:증분지령(INC) **프로그램 예제** **G81 0** >> Y 축 지령방식을 절대지령으로 설정합니다. **G00 0 1000** >> Y 축을 절대위치 1000 으로 이동합니다. **G08 100** >> 1 초간 딜레이 합니다. **G00 0 2000** >> Y 축을 절대위치 2000(상대 +1000)으로 이동합니다. **G81 1** >> Y 축 지령방식을 상대지령으로 설정합니다. **G00 1000** >> Y 축을 상대위치 +1000(절대위치 3000)으로 이동합니다. **END**	모드 0~1	미사용
G90	**서브루틴** 행번호로 시작되는 서브루틴 CALL 이며, M02(RETURN)명령으로 G90 의 다음 행으로 복귀한다. (사용예) --- 10 G90 100 11 --- 12 --- 100 --- 101 --- 102 M02 서브루틴은 최대 3 회 중첩 가능(서브루틴 속에서 서브루틴 콜)	행번호 0~249	미사용

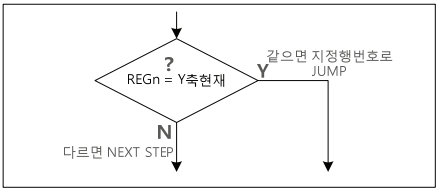
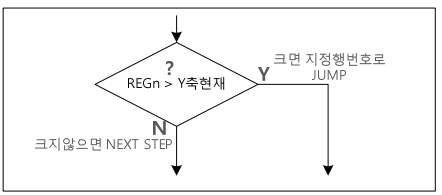
	프로그램 예제 **G90 4** >> 4 번행으로 점프하며 서브루틴을 시작합니다. **G00 1000 1000** >> X, Y 축을 각각의 설정값 만큼 이동합니다. 하지만 서브루틴 작동으로 추후에 동작합니다. **M76 0** >> 0 번출력을 OFF 합니다. **M75 0** >> 0 번 출력을 ON 합니다, 서브루틴이 시작되는 지점입니다. **G08 200** >> 2 초를 딜레이 합니다. **M02** >> 서브루틴을 종료하고 2 번행으로 돌아갑니다. **END**		

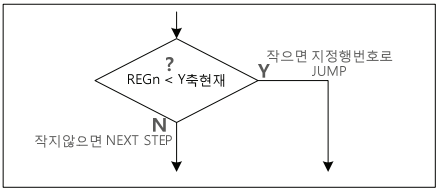
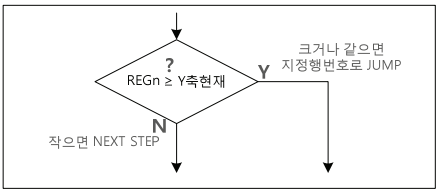
코드	내 용 설 명	1OP	2OP
M00	**프로그램 PAUSE** 프로그램 진행이 일시 정지하며, 화면의 RESUME 버튼을 누르면 재개한다. **프로그램 예제** **M70 0 6** >> 0 번 버튼이 입력되면 6 번 스탭으로 이동합니다. **M70 1 8** >> 1 번 버튼이 입력되면 8 번 스탭으로 이동합니다. **M70 2 10** >> 2 번 버튼이 입력되면 10 번 스탭으로 이동합니다. **M70 3 12** >> 3 번 버튼이 입력되면 12 번 스탭으로 이동합니다. **M03 1** >> 1 번 스탭으로 복귀 합니다. **G00 1000** >> X 축이 지령한 값 1000 으로 이동합니다. **M03 1** >> 1 번 스탭으로 복귀 합니다. **G00 2000** >> X 축이 지령한 값 2000 으로 이동합니다. **M03 1** >> 1 번 스탭으로 복귀 합니다. **G00 3000** >> X 축이 지령한 값 3000 으로 이동합니다. **M03 1** >> 1 번 스탭으로 복귀 합니다. **M00** >> 현재 스탭에서 프로그램을 일시정지 합니다. Resume 명령으로 재개할 수 있습니다. **M03 1** >> 1 번 스탭으로 복귀 합니다. **END**	미사용	미사용
M01	**서브루틴 스타트** 서브루틴의 시작점을 알리는 명령어	미사용	미사용
M02	**서브루틴 RET** 서브루틴 실행 중 복귀명령	미사용	미사용
M03	**JUMP** 입력된 행번호로 JUMP 하는 명령 **프로그램 예제** **G00 1000** >> x 축이 지령한 값 1000 으로 이동합니다. **M03 4** >> 4 번 스탭으로 점프합니다. **G00 2000** >> X 축이 지령한 값 2000 으로 이동합니다. (이전스탭의 점프명령어 때문에 동작하지 않습니다.) **M75 0** >> 0 번 출력을 ON 합니다. **G08 200** >> 2 초간 딜레이 합니다. **M76 0** >> 0 번 출력을 OFF 합니다. **END**	행번호 0~249	미사용

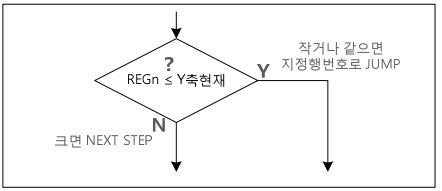
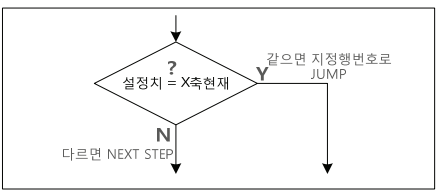
코드	내 용 설 명	1OP	2OP
M04	**LOOP START** LOOP 를 시작하는 명령어. 지정된 설정수만큼 LOOP 를 실행 **프로그램 예제** **M04 3** >> 현재 스탭부터 M05 번이 실행되는 스탭까지 3 회 반복합니다. **M75 0** >> 0 번 출력을 ON 합니다. **G08 100** >> 1 초간 딜레이 합니다. **M76 0** >> 0 번 출력을 OFF 합니다. **G08 100** >> 1 초간 딜레이 합니다. **M05** >> 루프의 마지막 명령입니다. **END**	횟수 1~100000	미사용
M05	**LOOP END** LOOP 의 끝. M04 예제 참조	미사용	미사용
M10	**레지스터 n 과 X 축 비교** n 레지스터와 X 축현재 값을 비교하여, 값이 같으면 이동. 같지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M10 0 6** >> X 축 현재값이 0 번 레지스터와 비교하여 같으면 6 번 스텝으로 이동합니다. **G80 1** >> X 축의 이동방법을 상대 위치값 지령으로 변경합니다. **G00 10000** >> X 축을 +10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249

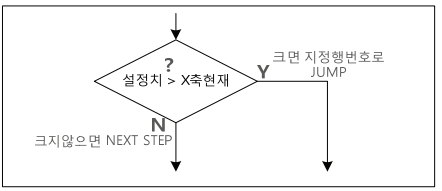
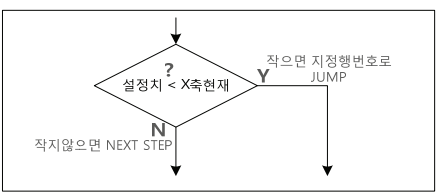
코드	내 용 설 명	1OP	2OP
M11	**레지스터 n 과 X 축 비교** n 레지스터와 X 축현재 값을 비교하여, 값이 크면 이동. 크지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M11 0 6** >> X 축 현재값이 0 번 레지스터값과 비교하여 값이 크면 6 번스텝으로 이동합니다. **G80 1** >> X 축의 이동방법을 상대위치값 지령으로 변경합니다. **G00 10000** >> X 축을 10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249
M12	**레지스터 n 과 X 축 비교** n 레지스터와 X 축현재 값을 비교하여, 값이 작으면 이동. 작지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 -30000** >> 0 번 레지스터에 -30000 의 값을 입력합니다. **M12 0 6** >> X 축 현재값이 0 번 레지스터값과 비교하여 값이 작으면 6 번스텝으로 이동합니다. **G80 1** >> X 축의 이동방법을 상대위치값 지령으로 변경합니다. **G00 -10000** >> X 축을 -10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249

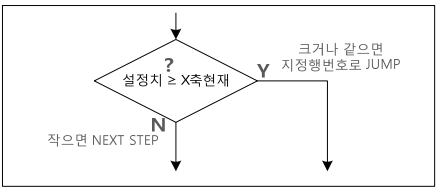
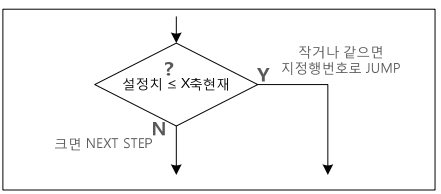
코드	내 용 설 명	1OP	2OP
M13	**레지스터 n 과 X 축 비교** n 레지스터와 X 축현재 값을 비교하여, 값이 크거나 같으면 이동. 작으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 -30000 의 값을 입력합니다. **M13 0 6** >> X 축 현재값이 0 번 레지스터값과 비교하여 값이 같거나 크면 6 번스텝으로 이동합니다. **G80 1** >> X 축의 이동방법을 상대위치값 지령으로 변경합니다. **G00 10000** >> X 축을 -10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249
M14	**레지스터 n 과 X 축 비교** n 레지스터와 X 축현재 값을 비교하여, 값이 작거나 같으면 이동. 크면 다음 스텝 진행.  **프로그램 예제** **M35 0 -30000** >> 0 번 레지스터에 -30000 의 값을 입력합니다. **M14 0 6** >> X 축 현재값이 0 번 레지스터값과 비교하여 값이 작거나 같으면 6 번스텝으로 이동합니다. **G80 1** >> X 축의 이동방법을 상대위치값 지령으로 변경합니다. **G00 -10000** >> X 축을 -10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249

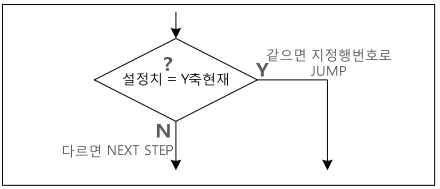
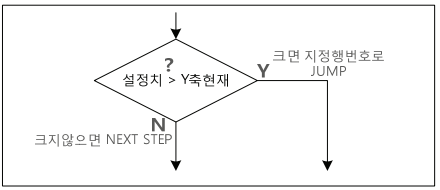
코드	내 용 설 명	1OP	2OP
M15	**레지스터 n 과 Y 축 비교** n 레지스터와 Y 축현재 값을 비교하여, 값이 같으면 이동. 같지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M15 0 6** >>y 축 현재값이 0 번 레지스터값과 비교하여 같으면 6 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> y 축을 +10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249
M16	**레지스터 n 과 Y 축 비교** n 레지스터와 Y 축현재 값을 비교하여, 값이 크면 이동. 크지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M16 0 6** >>y 축 현재값이 0 번 레지스터값과 비교하여 크면 6 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> y 축을 +10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249

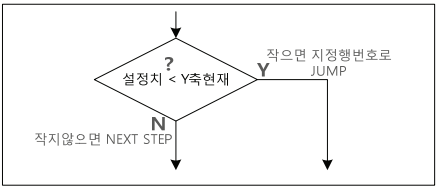
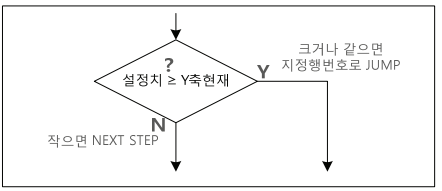
코드	내 용 설 명	1OP	2OP
M17	**레지스터 n 과 Y 축 비교** n 레지스터와 Y 축현재 값을 비교하여, 값이 작으면 이동. 작지 않으면 다음 스텝 진행.  **프로그램 예제** **M35 0 -30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M17 0 6** >>y 축 현재값이 0 번 레지스터값과 비교하여 작면 6 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 -10000** >> y 축을 -10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249
M18	**레지스터 n 과 Y 축 비교** n 레지스터와 Y 축현재 값을 비교하여, 값이 크거나 같으면 이동. 작으면 다음 스텝 진행.  **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 3000 의 값을 입력합니다. **M18 0 6** >>y 축 현재값이 0 번 레지스터값과 비교하여 크거나 같으면 6 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> y 축을 +1000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249

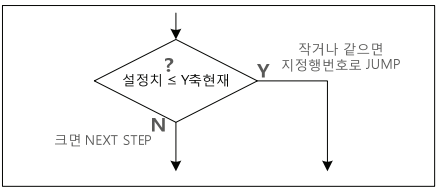
코드	내 용 설 명	1OP	2OP
M19	**레지스터 n 과 Y 축 비교** n 레지스터와 Y 축 현재 값을 비교하여, 값이 작거나 같으면 이동. 크면 다음 스텝 진행.  **프로그램 예제** **M35 0 -30000** >> 0 번 레지스터에 -30000 의 값을 입력합니다. **M19 0 6** >>y 축 현재값이 0 번 레지스터값과 비교하여 작거나 같으면 6 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 -10000** >> y 축을 -10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	행번호 0~249
M20	**설정치와 X 축 비교** 설정치와 X 축 현재 값을 비교하여, 값이 같으면 이동. 같지 않으면 다음 스텝 진행.  **프로그램 예제** **M20 30000 5** >>x 축 현재값이 설정값(30000)과 비교하여 같으면 5 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> x 축을 +10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30** 1 0 >> x 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249

코드	내 용 설 명	1OP	2OP
M21	**설정치와 X 축 비교** 설정치와 X 축현재 값을 비교하여, 값이 크면 이동. 크지 않으면 다음 스텝 진행.  **프로그램 예제** **M21 30000 5** >>x 축 현재값이 설정값(30000)과 비교하여 크면 5 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> x 축을 +10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30** 1 0 >> x 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
M22	**설정치와 X 축 비교** 설정치와 X 축현재 값을 비교하여, 값이 작으면 이동. 작지 않으면 다음 스텝 진행.  **프로그램 예제** **M22 -30000 5** >>x 축 현재값이 설정값(-30000)과 비교하여 작으면 5 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 -10000** >> x 축을 -10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30** 1 0 >> x 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
코드	내 용 설 명	1OP	2OP

M23	**설정치와 X 축 비교** 설정치와 X 축현재 값을 비교하여, 값이 크거나 같으면 이동. 작으면 다음 스텝 진행.  **프로그램 예제** **M21 30000 5** >>x 축 현재값이 설정값(30000)과 비교하여 크거나 같으면 5 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> x 축을 +10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30** 1 0 >> x 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
M24	**설정치와 X 축 비교** 설정치와 X 축현재 값을 비교하여, 값이 작거나 같으면 이동. 크면 다음 스텝 진행.  **프로그램 예제** **M21 -30000 5** >>x 축 현재값이 설정값(-30000)과 비교하여 작거나 같으면 5 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 -10000** >> x 축을 -10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30** 1 0 >> x 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
코드	**내 용 설 명**	1OP	2OP

M25	**설정치와 Y 축 비교** 설정치와 Y 축현재 값을 비교하여, 값이 같으면 이동. 같지 않으면 다음 스텝 진행.  **프로그램 예제** **M25 30000 5** >>y 축 현재값이 설정값(30000)과 비교하여 같으면 5 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> y 축을 +10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
M26	**설정치와 Y 축 비교** 설정치와 Y 축현재 값을 비교하여, 값이 크면 이동. 크지 않으면 다음 스텝 진행.  **프로그램 예제** **M26 30000 5** >>y 축 현재값이 설정값(30000)과 비교하여 크면 5 번스텝으로 이동합니다. **G81 1** >> y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> y 축을 +10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> y 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249

코드	내 용 설 명	1OP	2OP
M27	**설정치와 Y 축 비교** 설정치와 Y 축현재 값을 비교하여, 값이 작으면 이동. 작지 않으면 다음 스텝 진행.  **프로그램 예제** **M27 -30000 5** >>Y 축 현재값이 설정값(-30000)과 비교하여 작으면 5 번스텝으로 이동합니다. **G81 1** >> Y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 -10000** >> Y 축을 -10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> Y 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
M28	**설정치와 Y 축 비교** 설정치와 Y 축현재 값을 비교하여, 값이 크거나 같으면 이동. 작으면 다음 스텝 진행.  **프로그램 예제** **M28 30000 5** >>Y 축 현재값이 설정값(30000)과 비교하여 크거나 같으면 5 번스텝으로 이동합니다. **G81 1** >> Y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 10000** >> Y 축을 10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> Y 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
코드	내 용 설 명	1OP	2OP

M29	**설정치와 Y 축 비교** 설정치와 Y 축현재 값을 비교하여, 값이 작거나 같으면 이동. 크면 다음 스텝 진행.  **프로그램 예제** **M29 -30000 5** >>Y 축 현재값이 설정값(-30000)과 비교하여 작거나 같으면 5 번스텝으로 이동합니다. **G81 1** >> Y 축의 이송방법을 상대위치값으로 변경합니다. **G00 0 -10000** >> Y 축을 -10000 만큼 이동합니다. **M03 1** >> 2 번 스텝으로 복귀합니다. **G30 2 0** >> Y 축만 원점찾기 동작을 진행합니다. **END**	설정치 -8388000 ~ +8388000	행번호 0~249
M30	**X 축 대입** X 현재 위치에 임의의 설정값 대입. (동작 중 대입은 에러발생) **프로그램 예제** **G80 0** >> x 축을 절대위치 지령으로 변경합니다. **G00 10000** >> x 축을 지령값 10000 만큼 이동합니다. **M30 0** >> x 축의 현재값을 0 으로 변경합니다. **M03 2** >> 2 번스텝으로 이동합니다. **END**	설정치 -8388000 ~ +8388000	미사용
M31	**Y 축 대입** Y 현재 위치에 임의의 설정값 대입. (동작 중 대입은 에러발생) **프로그램 예제** **G81 0** >> Y 축을 절대위치 지령으로 변경합니다. **G00 0 10000** >> Y 축을 지령값 10000 만큼 이동합니다. **M31 0** >> Y 축의 현재값을 0 으로 변경합니다. **M03 2** >> 2 번스텝으로 이동합니다. **END**	설정치 -8388000 ~ +8388000	미사용

코드	내 용 설 명	1OP	2OP
M32	**X,Y 축 대입** X,Y 현재 위치에 임의의 설정값 대입. (동작 중 대입은 에러발생) **프로그램 예제** **G80 0** >> X 축을 절대위치 지령으로 변경합니다. **G81 0** >> Y 축을 절대위치 지령으로 변경합니다. **G00 10000 10000** >> X,Y 축을 지령값 10000 만큼 이동합니다. **M32 0 0** >> X,Y 축의 현재값을 0 으로 변경합니다. **M03 2** >> 2 번스텝으로 이동합니다. **END**	X 설정치 -8388000 ~ +8388000	Y 설정치 -8388000 ~ +838800
M33	**레지스터 0 대입** n 레지스터값을 레지스터 0 에 복사 REG0 ← REGn (M34 의 대입 방향이 반대) **프로그램 예제** **M35 1 30000** >> 1 번 레지스터에 3000 의 값을 입력합니다. **M33 1** >> 1 번 레지스터의 값을 0 번 레지스터로 복사 합니다. **M10 0 7** >>X 축 현재값이 0 번 레지스터값과 비교하여 같으면 7 번스텝으로 이동합니다. **G80 1** >> X 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> X 축을 +10000 만큼 이동합니다. **M03 3** >> 3 번 스텝으로 복귀합니다. **M30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	미사용
M34	**레지스터 n 대입** 레지스터 0 값을 n 레지스터에 복사 REGn ← REG0 (M33 의 대입 방향이 반대) **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 3000 의 값을 입력합니다. **M34 1** >> 0 번 레지스터의 값을 1 번 레지스터로 복사 합니다. **M10 0 7** >>X 축 현재값이 0 번 레지스터값과 비교하여 같으면 7 번스텝으로 이동합니다. **G80 1** >> X 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> X 축을 +1000 만큼 이동합니다. **M03 3** >> 3 번 스텝으로 복귀합니다. **M30 1 0** >> X 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	미사용
코드	내 용 설 명	1OP	2OP

M35	**레지스터 n 대입** 설정값을 n 레지스터에 복사 REGn ← 설정치 **프로그램 예제** **M35 0 30000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M10 0 6** >>x 축 현재값이 0 번 레지스터값과 비교하여 같으면 6 번스텝으로 이동합니다. **G80 1** >> x 축의 이송방법을 상대위치값으로 변경합니다. **G00 10000** >> x 축을 +10000 만큼 이동합니다. **M03 2** >> 2 번 스텝으로 복귀합니다. **M30 1 0** >> x 축만 원점찾기 동작을 진행합니다. **END**	레지스터 0~9	설정치 -8388000 ~ +8388000
M36	**X 축 위치설정** n 레지스터값을 X 현재위치에 복사 X 위치 ← REGn **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M30 1 0** >> x 축을 원점찾기 합니다. **M36 0** >> 0 번 레지스터의 값을 x 축 현재위치에 대입합니다. **G80 0** >> x 축의 이동방식을 절대위치 이동방식으로 설정합니다. **G00 20000** >> x 축을 20000(+10000)위치로 이동합니다. **END**	레지스터 0~9	미사용
M37	**Y 축 위치설정** n 레지스터값을 Y 현재위치에 복사 Y 위치 ← REGn **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 30000 의 값을 입력합니다. **M30 2 0** >> y 축을 원점찾기 합니다. **M37 0** >> 0 번 레지스터의 값을 y 축 현재위치에 대입합니다. **G81 0** >> y 축의 이동방식을 절대위치 이동방식으로 설정합니다. **G00 0 20000** >> y 축을 20000(+10000)위치로 이동합니다. **END**	레지스터 0~9	미사용
코드	**내 용 설 명**	1OP	2OP

M38	**X,Y 축 위치설정** n 레지스터 값을 X,Y 현재 위치 복사 X 위치 ← REGn Y 위치 ← REGn **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 10000 의 값을 입력합니다. **M35 1 20000** >> 1 번 레지스터에 20000 의 값을 입력합니다. **M30 0 0** >> X,Y 축을 원점찾기 합니다. **M38 0 1** >> 0 번 레지스터의 값을 x 축 현재위치에, 1 번 레지스터 값을 Y 축 현재위치에 대입합니다. **G80 0** >> x 축의 이동방식을 절대위치 이동방식으로 설정합니다. **G81 0** >> y 축의 이동방식을 절대위치 이동방식으로 설정합니다. **G00 20000 30000** >> X 축을 20000(+10000) Y 축을 30000(+10000) 위치로 이동합니다. **END**	레지스터 0~9	레지스터 0~9
M40	**X 축 위치복사** X 현재 값을 n 레지스터에 복사 REGn ← X 현재 **프로그램 예제** **G00 10000** >> X 축을 10000 만큼 이동합니다. **M40 0** >> X 축의 현재값을 0 번레지스터에 복사합니다. **M37 0** >> 0 번 레지스터를 Y 축 현재위치로 복사합니다. **END**	레지스터 0~9	미사용
M41	**Y 축 위치복사** Y 현재 값을 n 레지스터에 복사 REGn ← Y 현재 **프로그램 예제** **G00 0 10000** >> Y 축을 10000 만큼 이동합니다. **M41 0** >> Y 의 현재값을 0 번레지스터에 복사합니다. **M36 0** >> 0 번 레지스터를 X 축 현재위치로 복사합니다. **END**	레지스터 0~9	미사용
코드	**내 용 설 명**	1OP	2OP

M42	**X,Y 축 위치복사** X,Y 현재 값을 n 레지스터에 복사 REGn ← X 현재 REGn ← Y 현재 **프로그램 예제** **G00 20000 10000** >> X 축을 20000, Y 축을 10000 만큼 이동합니다. **M42 0 1** >> X 의 현재값을 0 번 레지스터에, Y 의 현재값을 1 번레지스터에 복사합니다. **M48 0 10000** >> 0 번 레지스터의 값에 10000 을 더합니다. **M48 1 10000** >> 1 번 레지스터의 값에 10000 을 더합니다. **M36 0** >> X 축의 현재위치값에 0 번 레지스터 값을 대입합니다. **M37 1** >> Y 축의 현재위치값에 1 번 레지스터 값을 대입합니다. **END**	레지스터 0~9	레지스터 0~9
M43	**레지스터 가감산** n 레지스터에 설정치 가감산 REGn ← REGn + 설정치(X 비율) **프로그램 예제** **M43 20000 0** >> 0 번 레지스텨에 20000 을 X 의 길이 비율로 가산한다 **M36 0** >> 0 번 레지스터의 값을 X 축 현재위치에 입력합니다. **END**	설정치 -8388000 ~ +8388000	레지스터 0~9
M44	**레지스터 가감산** n 레지스터에 설정치 가감산 REGn ← REGn + 설정치(Y 비율) **프로그램 예제** **M44 20000** 0 >> 0 번 레지스텨에 2000 을 y 의 길이 비율로 가산한다 **M37 0** >> 0 번 레지스터의 값을 y 축 현재위치에 입력합니다. **END**	설정치 -8388000 ~ +8388000	레지스터 0~9
코드	**내 용 설 명**	1OP	2OP

M45	**카운터 레지스터 대입** 카운터레지스터에 설정치 대입 C0 REG ← 설정치 C1 REG ← 설정치 파라미터 0~2 는, 0 : C0, C1 동시대입 1 : C0 대입 2 : C1 대입 **프로그램 예제** **M80 1** >> X 축을 상대값 이송으로 변경합니다. **M45 0 5** >> 카운터 0 번과 1 번을 5 로 설정합니다. **G00 10000** >> X 축을 10000 이송합니다. **M49 0** >> 0 번 레지스터에 1 을 가산합니다. **M50 0 7** >> 0 번 레지스터가 카운터 0 번값과 같으면 7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다. **M03 3** >> 3 번스텝으로 점프합니다. **G00 -10000** >> X 축을 -10000 이송합니다. **M49 1** >> 1 번 레지스터에 1 을 가산합니다. **M55 1 11** >> 1 번 레지스터가 카운터 1 번값과 같으면 11 번스텝으로 이동, 아닐시 다음스텝을 진행합니다. **M03 7** >> 7 번스텝으로 점프합니다. **END**	C0,1 지정 0~2	설정치 -8388000 ~ +8388000
M46	**REGn 가산** 레지스터 n 에 레지스터 m 을 더함 REGn ← REGn + REGm **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 10000 을 대입한다. **M35 1 20000** >> 1 번 레지스터에 20000 을 대입한다. **M46 0 1** >> 0 번 레지스터의 값에 1 번 레지스터 값을 더한다. **M65 0** >> 0 번 레지스터의 위치로 X 축이 이동한다. **END**	REGn 0~9	REGm 0~9
코드	**내 용 설 명**	1OP	2OP

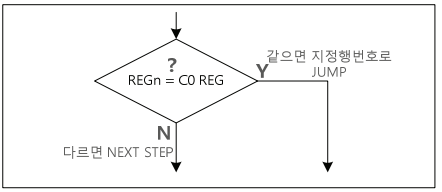
M47	**REGn 감산** 레지스터 n 에 레지스터 m 을 감산 REGn ← REGn – REGm **프로그램 예제** **M35 0 20000** >> 0 번 레지스터에 20000 을 대입한다. **M35 1 10000** >> 1 번 레지스터에 10000 을 대입한다. **M47 0 1** >> 0 번 레지스터의 값에 1 번 레지스터 값을 뺀다. **M65 0** >> 0 번 레지스터의 위치로 X 축이 이동한다. **END**	REGn 0~9	REGm 0~9
M48	**REGn 가감산** 레지스터 n 에 설정치 가감산 REGn ← REGn + 설정치 **프로그램 예제** **M35 0 20000** >> 0 번 레지스터에 20000 을 대입한다. **M35 1 -10000** >> 1 번 레지스터에 -10000 을 대입한다. **M48 0 1** >> 0 번 레지스터의 값에 1 번 레지스터 값을 더한다. **M65 0** >> 0 번 레지스터의 위치로 X 축이 이동한다. **END**	REGn 0~9	설정치 -8388000 ~ +8388000
M49	**REGn 증가** REGn 에 1 증가 REGn ← REGn + 1 **프로그램 예제** **M70 0 4** >> 0 번 입력을 누르면 4 번 스텝으로 이동한다. **M70 1 6** >> 1 입력을 누르면 6 번 스텝으로 이동한다. **M03 1** >> 1 번 스텝으로 이동한다. **M49 0** >> 0 번 레지스터에 1 을 더한다. **M03 1** >> 1 번 스텝으로 이동한다. **M45 0 5** >> 카운터 레지스터 0 번에 5 를 입력한다. **M51 0 9** >> 0 번 카운터 레지스터와 0 번 레지스터의 값을 비교하여 크면 9 번 스텝으로 이동한다. **M03 1** >> 1 번스텝으로 이동한다. **G00 10000 0** >> X 축을 10000 만큼 이동한다. **M35 0 0** >> 0 번레지스터에 0 을 대입한다. END	레지스터 0~9	미사용

코드	내 용 설 명	1OP	2OP

<table>
<tr><td>M50</td><td>

REGn 과 C0 비교

REGn 과 카운트 0 REG 값을 비교하여, 값이 같으면 이동.
같지 않으면 다음 스텝 진행.

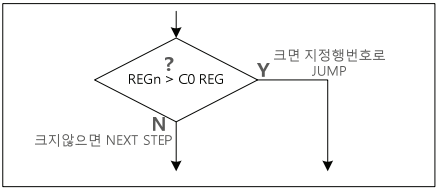


프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 1 5 >> 카운터 0 번 값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M50 0 7 >> 0 번 레지스터가 카운터 0 번값과 같으면 7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>M51</td><td>

REGn 과 C0 비교

REGn 과 카운트 0 REG 값을 비교하여, 값이 크면 이동.
크지 않으면 다음 스텝 진행.



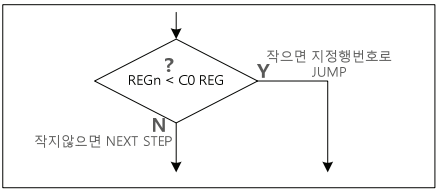
프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 1 5 >> 카운터 0 번값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M51 0 7 >> 0 번 레지스터가 카운터 0 번값과 크면 7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>코드</td><td>내 용 설 명</td><td>1OP</td><td>2OP</td></tr>
</table>

<table>
<tr><td>M52</td><td>

REGn 과 C0 비교

REGn 과 카운트 0 REG 값을 비교하여, 값이 작으면 이동.
작지 않으면 다음 스텝 진행.

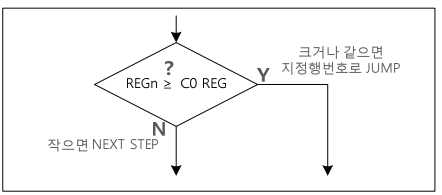


프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 1 5 >> 카운터 0 번 값을 5 로 설정합니다.
M52 0 5 >> 0 번카운터의 값이 0 번레지스터 값보다 작으면 5 번 스텝으로 이동, 아닐시 다음스텝 진행합니다.
M03 8 >>8 번스텝으로 이동합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M03 1 >> 1 번 스텝으로 이동합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>M53</td><td>

REGn 과 C0 비교

REGn 과 카운트 0 REG 값을 비교하여, 값이 크거나 같으면 이동.
작으면 다음 스텝 진행.



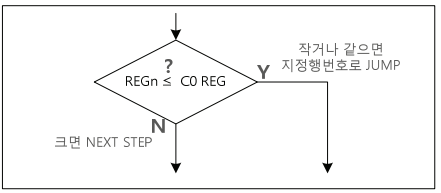
프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 1 5 >> 카운터 0 번 값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M53 0 7 >> 0 번 레지스터가 카운터 0 번값과 크거나 같으면 7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><th>코드</th><th>내 용 설 명</th><th>1OP</th><th>2OP</th></tr>
</table>

<table>
<tr><td>M54</td><td>

REGn 과 C0 비교

REGn 과 카운트 0 REG 값을 비교하여, 값이 작거나 같으면 이동.
크면 다음 스텝 진행.

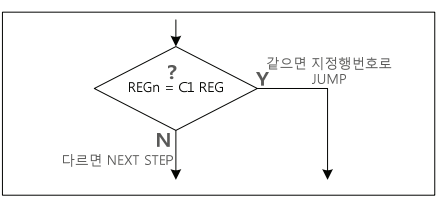


프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 1 5 >> 카운터 0 번 값을 5 로 설정합니다.
M54 0 5 >> 0 번카운터의 값이 0 번레지스터 값보다 작거나 같으면 5 번 스텝으로 이동, 아닐시 다음스텝 진행합니다.
M03 8 >>8 번스텝으로 이동합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M03 1 >> 1 번 스텝으로 이동합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>M55</td><td>

REGn 과 C1 비교

REGn 과 카운트 1 REG 값을 비교하여, 값이 같으면 이동.
같지 않으면 다음 스텝 진행.



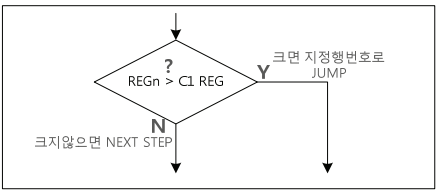
프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 2 5 >> 카운터 1 번 값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M55 0 7 >> 0 번 레지스터가 카운터 1 번값과 같으면 7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>코드</td><td>내 용 설 명</td><td>1OP</td><td>2OP</td></tr>
</table>

<table>
<tr><td>M56</td><td>

REGn 과 C1 비교

REGn 과 카운트 1 REG 값을 비교하여, 값이 크면 이동.
크지 않으면 다음 스텝 진행.

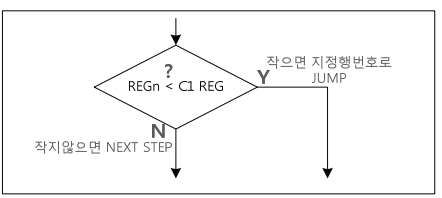


프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 2 5 >> 카운터 1 번값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M56 0 7 >> 0 번 레지스터가 카운터 1 번값과 크면 7 번스텝으로 이동,
아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>M57</td><td>

REGn 과 C1 비교

REGn 과 카운트 1 REG 값을 비교하여, 값이 작으면 이동.
작지 않으면 다음 스텝 진행.



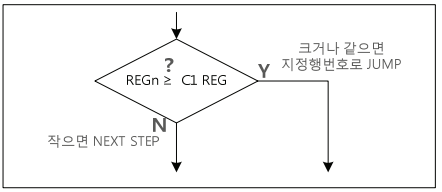
프로그램 예제
M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 2 5 >> 카운터 1 번 값을 5 로 설정합니다.
M52 0 5 >> 1 번카운터의 값이 0 번레지스터 값보다 작으면 5 번
스텝으로 이동, 아닐시 다음스텝 진행합니다.
M03 8 >>8 번스텝으로 이동합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M03 1 >> 1 번 스텝으로 이동합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>코드</td><td>내 용 설 명</td><td>1OP</td><td>2OP</td></tr>
</table>

<table>
<tr><td>M58</td><td>

REGn 과 C1 비교

REGn 과 카운트 1 REG 값을 비교하여, 값이 크거나 같으면 이동.
작으면 다음 스텝 진행.



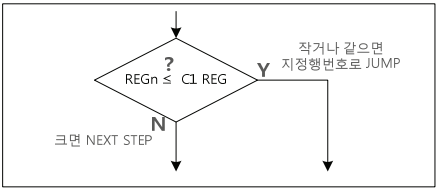
프로그램 예제

M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 2 5 >> 카운터 1 번 값을 5 로 설정합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M58 0 7 >> 0 번 레지스터가 카운터 1 번값과 크거나 같으면
7 번스텝으로 이동, 아닐시 다음스텝을 진행합니다.
M03 3 >> 3 번스텝으로 점프합니다.
END

</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>M59</td><td>

REGn 과 C1 비교

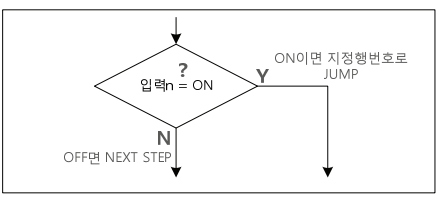
REGn 과 카운트 1 REG 값을 비교하여, 값이 작거나 같으면 이동.
크면 다음 스텝 진행.



프로그램 예제

M80 1 >> X 축을 상대값 이송으로 변경합니다.
M45 2 5 >> 카운터 1 번 값을 5 로 설정합니다.
M59 0 5 >> 1 번카운터의 값이 0 번레지스터 값보다 작거나 같으면 5 번
스텝으로 이동, 아닐시 다음스텝 진행합니다.
M03 8 >>8 번스텝으로 이동합니다.
G00 10000 >> X 축을 10000 이송합니다.
M49 0 >> 0 번 레지스터에 1 을 가산합니다.
M03 1 >> 1 번 스텝으로 이동합니다.
END

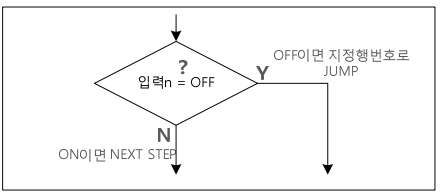
</td><td>레지스터
0~9</td><td>행번호
0~249</td></tr>
<tr><td>코드</td><td>내 용 설 명</td><td>1OP</td><td>2OP</td></tr>
</table>

M65	**REGn ➔ X 이동** REGn 지점으로 X 축을 이송한다. **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 10000 을 입력한다 **M65 0** >> 0 번 레지스터 값으로 X 축을 이송한다 **END**	레지스터 0~9	미사용
M66	**REGn ➔ Y 이동** REGn 지점으로 Y 축을 이송한다. **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 10000 을 입력한다 **M66 0** >> 0 번 레지스터 값으로 Y 축을 이송한다 **END**	레지스터 0~9	미사용
M67	**REGn ➔ X, Y 이동** REGn 지점으로 X, Y 축을 이송한다. **프로그램 예제** **M35 0 10000** >> 0 번 레지스터에 10000 을 입력한다 **M67 0** >> 0 번 레지스터 값으로 X,Y 두축을 이송한다 **END**	레지스터 0~9	레지스터 0~9
M70	**입력 ON ?** 입력포트 n 이 ON 이면 JUMP, OFF 이면 다음 스텝 진행.  **프로그램 예제** **M70 0 2** >> 0 번 입력이 on 일시 2 번스텝으로 이동, 아닐시 다음스텝 진행합니다 **M03 1** >> 1 번스텝으로 이동합니다. **G00 10000** >> X 축을 10000 이동합니다. **END**	입력번호 0~47	행번호 0~249
코드	**내 용 설 명**	1OP	2OP

<table>
<tr>
<td>M71</td>
<td>

입력 OFF ?

입력포트 n 이 OFF 이면 JUMP, ON 이면 다음 스텝 진행.



프로그램 예제

M71 0 2 >> 0 번 입력이 off 일시 2 번스텝으로 이동, 아닐시 다음스텝 진행합니다

M03 1 >> 1 번스텝으로 이동합니다.

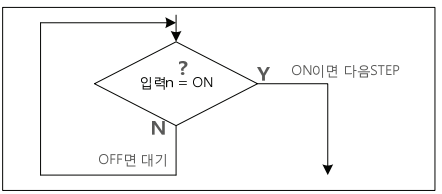
G00 10000 >> x 축을 10000 이동합니다.

END

</td>
<td>입력번호
0~47</td>
<td>행번호
0~249</td>
</tr>
<tr>
<td>M72</td>
<td>

입력 ON 대기

입력포트 n 이 ON 되기까지 대기.



프로그램 예제

M72 0 >> 0 번 입력이 on 될 때까지 대기합니다.

M72 1 >> 1 번 입력이 on 될 때까지 대기합니다.

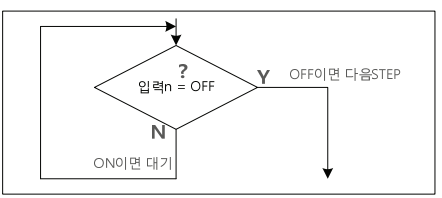
G00 10000 >> x 축을 10000 이동합니다.

END

</td>
<td>입력번호
0~47</td>
<td>미사용</td>
</tr>
<tr>
<td>M73</td>
<td>

입력 OFF 대기

입력포트 n 이 OFF 되기까지 대기.



프로그램 예제

</td>
<td>입력번호
0~47</td>
<td>미사용</td>
</tr>
</table>

	M72 0 >> 0 번 입력이 on 될때까지 대기합니다. **G00 10000** >> X 축을 10000 이동합니다. **M73 0** >> 0 번입력이 off 될때까지 대기합니다. **G00 0 10000** >> Y 축을 10000 이동합니다. **END**		
M75	**출력 ON** 출력포트 n ➔ ON 출력. **프로그램 예제** **M75 0** >> 0 번출력을 on 합니다. **END**	출력번호 0~47	미사용
M76	**출력 OFF** 출력포트 n ➔ OFF 출력. **프로그램 예제** **M76 0** >> 0 번출력을 off 합니다. **END**	출력번호 0~47	미사용
M77	**버튼대기** 숫자 0~9 버튼이 ON 될때까지 대기 **프로그램 예제** **M77 0** >> 0 번 키패드가 눌릴 때까지 대기합니다. **M75 0** >> 0 번 출력을 on 합니다. **END**	버튼번호 0~9	미사용
M78	**버튼 ON?** 숫자 0~9 버튼이 ON 이면 JUMP **프로그램 예제** **M78 0** >> 0 키패드가 on 일시 설정된 스텝으로 이동, 아닐시 다음 스텝을 실행합니다. **M76 0 >>** 0 번 출력을 off 합니다. **M03 1 >>** 1 번 스텝으로 이동합니다. **M75 0 >>** 0 번 출력을 on 합니다.. **END**	버튼번호 0~9	행번호 0~249

파라미터 범위

파라미터는 모드버스 프로토콜을 이용하여 수정할 수가 있습니다.(8.모드버스 인터페이스 참조)
수정 시, 수정하는 값의 범위(최소값/최대값)를 다음에 표시합니다.
(이 범위를 벗어나는 값으로 수정하면, 최소값으로 수정됩니다.)

항 목	최 소	최 대	기본설정 (Default)	내 용
X,Y 축 갯수	1	2	2	
[X]축 회전방향	0	1	0	CW:0, CCW:1
[Y]축 회전방향	0	1	0	CW:0, CCW:1
[X]소수점 자릿수	0	3	2	0.00(소수점두자리)
[Y]소수점 자릿수	0	3	2	0.00(소수점두자리)
[X]이송(ABS/INC)	0	1	0	절대좌표:0, 상대좌표:1
[Y]이송(ABS/INC)	0	1	0	절대좌표:0, 상대좌표:1
파일번호	0	9	0	
[X]홈센서 위치	0	1	0	CCW 로 서치:0, CW 로 서치:1
[Y]홈센서 위치	0	1	0	CCW 로 서치:0, CW 로 서치:1
[X]원점 모드	0	2	0	프로그램 코드 G30 설명참조 0: 원점 서치 2 단(CCW ➔ CW) 1: 원점 서치 2 단(back 하여 CCW) 2: 원점 서치 1 단
[Y]원점 모드	0	2	0	프로그램 코드 G30 설명참조 0: 원점 서치 2 단(CCW ➔ CW) 1: 원점 서치 2 단(back 하여 CCW) 2: 원점 서치 1 단
카운터 REG(0)	1	100000	100	
카운터 REG(1)	1	100000	100	
[X]원점 1 PPS	100	500000	5000	원점 서치 모드 0,1 일경우 사용
[Y]원점 1 PPS	100	500000	5000	원점 서치 모드 0,1 일경우 사용
[X]원점 2 PPS	100	500000	5000	원점 서치 모드 2 일경우 사용
[Y]원점 2 PPS	100	500000	5000	원점 서치 모드 2 일경우 사용
[X]시작 PPS	100	500000	5000	이송 시 스타트 속도
[Y]시작 PPS	100	500000	5000	이송 시 스타트 속도
[X]조그 PPS	100	500000	5000	MPG 사용 시 이송속도
[Y]조그 PPS	100	500000	5000	MPG 사용 시 이송속도
[X]일반 PPS	100	500000	50000	이송 시 속도
[Y]일반 PPS	100	500000	50000	이송 시 속도

항 목	최 소	최 대	기본설정 (Default)	내 용
[X]원점 후 이동(mm)	-99999	99999	0	원점 서치 후 이송
[Y]원점 후 이동(mm)	-99999	99999	0	원점 서치 후 이송
[X]원점 설정(mm)	-99999	99999	0	원점 서치 후 현재(원점)위치로 설정
[Y]원점 설정(mm)	-99999	99999	0	원점 서치 후 현재(원점)위치로 설정
[X]백래시(mm)	-9999	9999	0	*1
[Y]백래시(mm)	-9999	9999	0	*1
[X]보정거리(mm)	0	99999	0	*2
[Y]보정거리(mm)	0	99999	0	*2
[X]보정 Offset(mm)	-9999	9999	0	*2
[Y]보정 Offset(mm)	-9999	9999	0	*2
[X]가감속 (msec)	1	9999	200	
[Y]가감속 (msec)	1	9999	200	
[X]정지모드	0	1	0	STOP 버튼에 의한 정지 시 감속정지:0, 급정지:1
[Y]정지모드	0	1	0	STOP 버튼에 의한 정지 시 감속정지:0, 급정지:1
[X]서보 1 회전 펄스	100	1000000	10000	*3
[Y]서보 1 회전 펄스	100	1000000	10000	*3
[X]볼스크류(0.01mm)	50	100000	500	*3
[Y]볼스크류(0.01mm)	50	100000	500	*3
[X]기어비 분자	1	10000	10	*3
[Y]기어비 분자	1	10000	10	*3
[X]기어비 분모	1	10000	10	*3
[Y]기어비 분모	1	10000	10	*3
G09 용 지연시간	0	100000	0	mSec
통신 ID	1	250	1	
원격 (Inp) 조절 사용	0	1	0	*4
PPS(0),mm/S(1)	0	1	0	*5
비밀번호	0	9999	550	*6
[X]소프트리미트 사용	0	1	0	*7
[X] – 리미트 (mm)	-9999	9999		*7
[X] + 리미트 (mm)	-9999	9999		*7

항 목	최 소	최 대	기본설정 (Default)	내 용
[Y]소프트리미트 사용	0	1	0	*7
[Y] – 리미트 (mm)	-9999	9999		*7
[Y] +리미트 (mm)	-9999	9999		*7

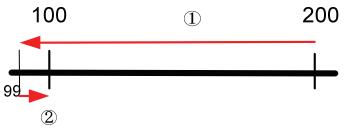
【*1】

백래시의 보상 운전 값입니다.

(예) 값이 -1.00mm 일 경우, (200 에서 100 이동)

① 200 에서 99 로 이동 후,

② 99 에서 100 으로 이동.

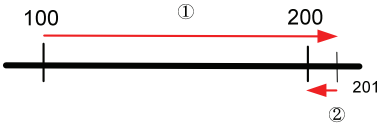


100 에서 200 이동은 그대로 이동.

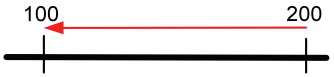
(예) 값이 +1.00mm 일 경우, (100 에서 200 이동)

① 100 에서 201 로 이동 후,

② 201 에서 200 으로 이동.



200 에서 100 이동은 그대로 이동.



값이 0 일때는, 양방향 모두 그대로 이동합니다.

【*2】

거리에 따르는 보상값입니다.

거리 1000mm 당 0.1mm 를 보상해야 할 경우,

보정거리 = 100000

보정옵셋 = 10

으로, 0.01mm 단위로 설정하면 됩니다.

예를 들어, 1000mm 이송을 시켰는데 결과가 1002mm 일 경우

보정거리 = 100000

보정옵셋 = -200

으로 설정하면 됩니다.

【*3】

피치(펄스/0.01mm) 설정 관련 파라미터입니다.

mm 당 펄스 계산은,

$$\frac{\text{(1)서보 1 회전당 펄스수 / (3)기어비}}{\text{(2)볼 스크류 피치}}$$

단, (3)기어비는 기어비분자/기어비분모

결과는 0.01mm 당 펄스수, 즉, 피치가 계산되어, 모든 이송에 쓰이는 중요한 파라미터입니다.
정확한 데이터를 입력시키지 않으면, 이송거리가 틀릴 수 있습니다.

1) 볼스크류 사용일 때

1 회전 5mm 인 볼스크류이고, 감속비가 1/2 인 시스템은

서보 1 회전당 펄스수 : 10000

볼 스크류 피치 : 500

기어비 분자 : 1

기어비 분모 : 2

$$\frac{10000 / (1 / 2)}{500}$$

로, 40 이됩니다.

즉, 0.01mm 당 40 펄스가 출력됩니다.

2) 볼스크류 미사용일 때

감속비가 1/10 인 시스템은

서보 1 회전당 펄스수 : 10000
볼 스크류 피치 : 1000 ➔ 으로 놓고 계산
기어비 분자 : 1
기어비 분모 : 10

$$\frac{10000 / (1 / 10)}{1000}$$

로, 100 이 됩니다.
즉, 0.01mm 당 100 펄스가 출력됩니다. ➔ 펄스 수와 거리를 조정할 때, 즉, 0.01mm 당 10 펄스로 하고 싶으면, 볼스크류 피치를 10000 으로 하거나, 분자를 10 으로 하면 됩니다.

★ 피치계산이 정수로 되지 않을 경우에는, 서보 1 회전당 펄스수와 볼스크류값 기어비등을 조정하여, 정수가 되도록 하여야 합니다.

【*4】

일반 Input 으로 자동 프로그램의 시작,정지 기능을 사용할 수 있는 파라미터 입니다.

이 파라미터를 활성화하게 되면 DI0 번을 시작 1 번을 정지 버튼으로 사용할 수 있습니다.

【*5】

속도 지령 방식을 변경합니다.

0 번을 입력할 시 속도를 PPS 로 지령하고 1 번으로 입력할 시 속도를 mm/S 로 지령합니다.

mm/S 의 경우는 기어비와 서보 펄스수를 계산하여 지령합니다.

【*6】

비밀번호를 지정합니다.

기본 550(0550)으로 저장되어 있으며 기본 운전화면 >> 설정화면 으로 전환시에 사용되는 비밀번호입니다.

【*7】

X,Y 축에 소프트 리미트 기능을 추가합니다.

사용여부를 결정할 수 있고 위치값 입력하여 리미트 스위치 사용 없이 리미트를 사용할 수 있도록 설정합니다.

최소값은 최대값보다 클 수 없고 최대값은 최소값보다 작을 수 없습니다. 이를 어길시 오동작할 여지가 있습니다.

7. 모드버스 인터페이스

7.1. 초기설정

BAUDRATE : 38400, 8, N, 1 (ASCII 모드)
콘트롤러 ID : 01

7.2. 주소범위

레지스터	형식	워드번호	비트번호	데이터 길이
출력 레지스터	Wn	40001~50000	없음	워드(2 바이트)
입력 레지스터	Wn	30001~40000	없음	워드(2 바이트)

레지스터	형식	워드번호	비트번호	데이터 길이
출력 비트	Bn	없음	1~10000	1 바이트
입력 비트	Bn	없음	10001~20000	1 바이트

비트는 모드버스상에서 1 바이트단위로 구분되며,
0x00(HEX DATA) ➔ OFF
0xFF(HEX DATA) ➔ ON

워드는 2 바이트 처리, 더블워드는 4 바이트 단위로 처리됩니다.

【출력비트 (X 축 서보 ON/OFF) 사용예】
X 축 서보 ON/OFF 의 출력 비트 주소는 1 이며, 모드버스 프로토콜 상에서는 0

(전체적인 항목별주소는 7-3 에 있습니다)

【X 축 서보 ON】

모드버스 FIELD	예(HEX DATA)
Heading	3A
Slave Address	01
Command Code	05
Bit Address HI	00
Bit Address LO	00
Force Data HI	FF
Force Data LO	00
Error Check(LRC)	FB

【X 축 서보 OFF】

모드버스 FIELD	예(HEX DATA)
Heading	3A
Slave Address	01
Command Code	05
Bit Address HI	00
Bit Address LO	00
Force Data HI	00
Force Data LO	00
Error Check(LRC)	FA

7.3. 항목별 주소

출력 W@40001	X	Y	Z	A	B	C
	1 축	2 축	3 축	4 축	5 축	6 축
pulse 목표	**40001**	**41001**	**42001**	**43001**	**44001**	**45001**
mm 목표	40003	41003	42003	43003	44003	45003
카운트(Work)	40005	41005	42005	43005	44005	45005
카운트 목표	40007	41007	42007	43007	44007	45007
CW(0)/CCW(1)	40009	41009	42009	43009	44009	45009
이송 ABS(0)INC(1)	40010	41010	42010	43010	44010	45010
소수점 자릿수	40011	41011	42011	43011	44011	45011
펄스(0)/mm(1) 표시	40012	41012	42012	43012	44012	45012
홈센서 좌 0/우 1	40013	41013	42013	43013	44013	45013
원점 모드	40014	41014	42014	43014	44014	45014
정지(감속 0/급 1)	40015	41015	42015	43015	44015	45015
원점 1 PPS	40016	41016	42016	43016	44016	45016
원점 2 PPS	40018	41018	42018	43018	44018	45018
시작 PPS	40020	41020	42020	43020	44020	45020
조그 PPS	40022	41022	42022	43022	44022	45022
일반 PPS	40024	41024	42024	43024	44024	45024
원점 후 이동(mm)	40026	41026	42026	43026	44026	45026
원점 설정(mm)	40028	41028	42028	43028	44028	45028

출력 W@40001	X	Y	Z	A	B	C
	1 축	2 축	3 축	4 축	5 축	6 축
백래시(mm)	40030	41030	42030	43030	44030	45030
가감속 (msec)	40032	41032	42032	43032	44032	45032
서보 1 회전 펄스	40034	41034	42034	43034	44034	45034
볼스크류(0.01mm)	40036	41036	42036	43036	44036	45036
기어비 분자	40038	41038	42038	43038	44038	45038
기어비 분모	40040	41040	42040	43040	44040	45040
FPGA 카운터 목표	40042	41042	42042	43042	44042	45042
G21 PPS	40044	41044	42044	43044	44044	45044
원점 1 RPM	40046	41046	42046	43046	44046	45046
원점 2 RPM	40048	41048	42048	43048	44048	45048
시작 RPM	40050	41050	42050	43050	44050	45050
조그 RPM	40052	41052	42052	43052	44052	45052
일반 RPM	40054	41054	42054	43054	44054	45054
백래시 2(mm)	40056	41056	42056	43056	44056	45056
보정거리(mm)	40058	41058	42058	43058	44058	45058
보정 Offset(mm)	40060	41060	42060	43060	44060	45060

공통/축 개수	**46001**
USB File 번호	46002
파일번호	46003
초기화면	46004
통신 ID	46005
	46006
G09 지연시간	46007
파일(Write)	**46009**
	46010
	46011
	46012
	46013
	46014
	46015
Analog OUT Ch0	46016
Analog OUT Ch1	46017
Analog OUT Ch2	46018
Analog OUT Ch3	46019
Analog OUT Ch4	46020
Analog OUT Ch5	46021
Analog OUT Ch6	46022
Analog OUT Ch7	46023
프로그램 NO.	46024
G 코드	46025
	46026
	46027
	46028

X	46029

Y	46031
Z	46033
R	46035
A	46037
B	46039
C	46041
D	46043

입력 W@30001	X	Y	Z	A	B	C
	1 축	2 축	3 축	4 축	5 축	6 축
현재(pulse)	**30001**	**31001**	**32001**	**33001**	**34001**	**35001**
현재(mm)	30003	31003	32003	33003	34003	35003

카운트 레지스터	**36001**
스텝	36003
PGM 스텝 NO.	36004

파일 1	**36009**
	36010
	36011
	36012
	36013
	36014
	36015
파일 2	**36016**
	36017
	36018
	36019
	36020
	36021
파일 3	**36023**
	36024
	36025
	36026
	36027
	36028
파일 4	**36030**
	36031
	36032

	36033
	36034
	36035
파일 5	**36037**
	36038
	36039
	36040
	36041
	36042

Analog IN Ch0	36044
Analog IN Ch1	36045
Analog IN Ch2	36046
Analog IN Ch3	36047
Analog IN Ch4	36048
Analog IN Ch5	36049
Analog IN Ch6	36050
Analog IN Ch7	36051

출력 B@1	X	Y	Z	A	B	C
	1 축	2 축	3 축	4 축	5 축	6 축
서보 ON/OFF	**1**	**1001**	**2001**	**3001**	**4001**	**5001**
알람 리셋	2	1002	2002	3002	4002	5002
원점 실행	3	1003	2003	3003	4003	5003
수동 목표운전	4	1004	2004	3004	4004	5004
mm 목표운전	5	1005	2005	3005	4005	5005
(+) 조그	6	1006	2006	3006	4006	5006
(-) 조그	7	1007	2007	3007	4007	5007
속도 제어 (+)	8	1008	2008	3008	4008	5008
속도 제어 (-)	9	1009	2009	3009	4009	5009
정지	10	1010	2010	3010	4010	5010

알람이력초기화	**6001**
파라미터 초기화	6002
포인트 초기화	6003
공정파일 초기화	6004
내부메모리 초기화	6005
자동 스타트	6006
원점 동시	6007
수동동시	6008
속도(+)동시	6009
속도(-)동시	6010
정지 동시	6011
mm 목표동시	6012
프로그램 입력	6013
PGM UP	6014
PGM DN	6015
	6016
	6017
	6018
	6019
USB 파일 1 읽기	6020
USB 파일 2 읽기	6021
USB 파일 3 읽기	6022
USB 파일 4 읽기	6023
USB 파일 5 읽기	6024
USB 쓰기	6025
USB UP	6026
USB DN	6027

Ext_Out0	6038
Ext_Out1	6039
Ext_Out2	6040
Ext_Out3	6041
Ext_Out4	6042
Ext_Out5	6043
Ext_Out6	6044
Ext_Out7	6045
Ext_Out8	6046
Ext_Out9	6047
Ext_Out10	6048
Ext_Out11	6049
Ext_Out12	6050
Ext_Out13	6051
Ext_Out14	6052
Ext_Out15	6053
Ext_Out16	6054
Ext_Out17	6055
Ext_Out18	6056
Ext_Out19	6057
Ext_Out20	6058
Ext_Out21	6059
Ext_Out22	6060
Ext_Out23	6061
Ext_Out24	6062
Ext_Out25	6063
Ext_Out26	6064
Ext_Out27	6065
Ext_Out28	6066
Ext_Out29	6067
Ext_Out30	6068
Ext_Out31	6069

릴레이 1-0	6070
릴레이 1-1	6071
릴레이 1-2	6072
릴레이 1-3	6073
릴레이 1-4	6074
릴레이 1-5	6075
릴레이 1-6	6076
릴레이 1-7	6077
릴레이 1-8	6078
릴레이 1-9	6079
릴레이 1-10	6080
릴레이 1-11	6081
릴레이 1-12	6082
릴레이 1-13	6083
릴레이 1-14	6084
릴레이 1-15	6085
릴레이 2-0	6086
릴레이 2-1	6087
릴레이 2-2	6088
릴레이 2-3	6089
릴레이 2-4	6090
릴레이 2-5	6091
릴레이 2-6	6092
릴레이 2-7	6093
릴레이 2-8	6094
릴레이 2-9	6095
릴레이 2-10	6096
릴레이 2-11	6097
릴레이 2-12	6098
릴레이 2-13	6099
릴레이 2-14	6100
릴레이 2-15	6101

입력 B@10001	X	Y	Z	A	B	C
	1 축	2 축	3 축	4 축	5 축	6 축
Ready	**10001**	**11001**	**12001**	**13001**	**14001**	**15001**
InPos	10002	11002	12002	13002	14002	15002
Alarm	10003	11003	12003	13003	14003	15003
(+) Limit	10004	11004	12004	13004	14004	15004
(-) Limit	10005	11005	12005	13005	14005	15005
HOME	10006	11006	12006	13006	14006	15006
운전 중	10007	11007	12007	13007	14007	15007

종합 에러	**16001**
USB 삽입	16002
운전 중	16003
LED ON	16004
PGM ERR	16005
USB ERR	16006
	16007
	16008
	16009
Ext_In0	16010
Ext_In1	16011
Ext_In2	16012
Ext_In3	16013
Ext_In4	16014
Ext_In5	16015
Ext_In6	16016
Ext_In7	16017
Ext_In8	16018
Ext_In9	16019
Ext_In10	16020
Ext_In11	16021
Ext_In12	16022
Ext_In13	16023
Ext_In14	16024
Ext_In15	16025
Ext_In16	16026
Ext_In17	16027
Ext_In18	16028
Ext_In19	16029

Ext_In20	16030
Ext_In21	16031
Ext_In22	16032
Ext_In23	16033
Ext_In24	16034
Ext_In25	16035
Ext_In26	16036
Ext_In27	16037
Ext_In28	16038
Ext_In29	16039
Ext_In30	16040
Ext_In31	16041
Ext_In32	16042
Ext_In33	16043
Ext_In34	16044
Ext_In35	16045
Ext_In36	16046
Ext_In37	16047
Ext_In38	16048
Ext_In39	16049
Ext_In40	16050
Ext_In41	16051
Ext_In42	16052
Ext_In43	16053
Ext_In44	16054
Ext_In45	16055
Ext_In46	16056
Ext_In47	16057