

# Network Based Digital Motor Controller

STEP 모터 드라이버 사용설명서

Model : CUBE-STxxxx-xxx



[www.robocube.co.kr](http://www.robocube.co.kr)

Revision : Dec/27/2010

목 차

1.	사용하기 전에	4
1.1	제품의 확인	4
1.2	주의사항	4
2.	QUICK GUIDE	5
2.1	시스템 구성	5
2.2	통신 설정	6
2.3	모터파라미터 설정	7
2.4	위치제어	7
2.5	속도제어	7
3.	제품 소개	8
3.1	FEATURES	8
3.2	제품 사양	9
4.	배선	13
4.1	단자대 결선도	13
4.2	전원연결	15
4.3	Motor Winding 연결	16
4.4	Encoder 연결	16
4.5	RS232 통신 연결	17
4.6	RS485 또는 CAN 통신 연결	18
4.7	DIGITAL/ANALOG INPUT/OUTPUT 연결(LIMIT 스위치 연결)	19
5.	통신 프로토콜	21
5.1	통신 개요	21
5.2	RS232 통신	22
5.3	RS485 통신	23
5.4	CAN 통신	24
5.5	RS485/CAN 통신 선택방법	24
5.6	HOST 프로그램 설치 및 실행	27
6.	모터 구동 파라미터 설정	37
6.1	SX 명령으로 기본 환경 설정하기	37
6.2	ST 명령으로 모터 타입 설정하기	39
6.3	SA 명령으로 제어기 주소 설정하기	40
6.4	SEA55A 명령으로 엔코더 펄수 설정하기	40
6.5	SGA55A 명령으로 제어 변수 설정하기	41
6.6	SI 명령으로 전류제한값 설정하기	42
6.7	Sw 명령으로 전류제한값 설정하기	43
6.8	SQ/Sq/SR/Sr 명령으로 제어파라미터 설정하기	44
6.9	SH명령으로 구동전류 진폭 설정하기	46
6.10	Sh명령으로 전류명령값의 최대치 설정하기	47
7.	모터 구동	49
7.1	제어모드 설정	49
7.2	서보 Enable/Disable	49
7.3	PA/PB/Pa/PI/P1/P2 명령을 이용한 위치제어	50
7.4	SV 명령을 이용한 속도제어	52

7.5	PV 명령을 이용한 속도제어	53
7.6	PP/Pp/PQ/Pq/p 명령을 이용한 실시간 위치제어	54
7.7	PP/Pp/PQ/Pq/p 명령을 이용한 속도제어	56
7.8	PM 명령을 이용한 위치제어	56
7.9	Ps 명령으로 동작중 정지시키기	57
7.10	PI 명령으로 위치제어하기	58
7.11	GH/Gh 명령을 이용한 원점복귀	59
7.12	GI/Gi 명령을 이용한 원점복귀	61
7.13	GJ/Gj 명령을 이용한 원점복귀	63
7.14	GK/Gk 명령을 이용한 원점복귀	65
7.15	J/K/j/k 명령을 이용한 조그동작	66
7.16	사인파 위치제어 동작	68
8.	FAULT 처리	69
8.1	개요	69
8.2	FAULT 상태표	70
9.	시퀀스 자동화 프로그램	75
9.1	개요	75
9.2	샘플 프로그램	77
9.3	예약어	78
10.	제품 DIMENSION	86
※	매뉴얼 변경사항 HISTORY	87

PRELIMINARY

# 1. 사용하기 전에

## 1.1 제품의 확인

- 본 제품은 Controller, 초기세팅용 시리얼 통신케이블 1개로 구성되어 있습니다. (케이블은 추가구매 가능합니다)
- 제품을 수령 후 기본 방열판에 표기된 Model Number를 점검하여 주문하신 제품과 일치하는가를 확인하시기 바랍니다.

CUBE	-DC	xx	xx	-x	I	X	xxx
	모터의 종류	입력전압	정격전류	모터 채널 수	용도	MOSFET 종류	옵션
BL	BLDC 모터	24 ~30V	01 ~1A	S	1 채널	I 산업용	R IRMCK201
DC	DC 모터	36 ~45V	02 ~2A	D	2 채널	M 의료용	E Encoder
ST	STEP 모터	48 ~60V	04 ~4A			D 국방용	U USB
			08 ~8A			A Auto motive	L LAN
			16 ~16A			Q Small Integrated	
			32 ~32A				

## 1.2 주의 사항

- ⚠ 제품을 처음 사용하는 고객은 꼭 매뉴얼 숙지 후 사용하십시오.
- ⚠ 모든 선의 연결 및 제거를 하기 전에 전원이 꺼져 있는지 확인 하십시오.
- ⚠ 최초 전원 공급 전에 모든 연결선의 결선 위치, 방향을 꼭 확인 하십시오. 전원 공급 단자의 (+),(-) 선을 거꾸로 연결하거나 정격전압이상의 과도한 전압을 인가하는 경우 제어기가 파손될 수 있습니다.
- ⚠ 최초 모터의 회전 시험은 모터의 모든 부하를 제거한 후 실행 하십시오.



- 커넥터의 1번 핀은 Firmware Download를 위해 할당된 핀으로서, 일반적인 통신용으로 사용시에 1번 핀은 연결되지 않도록 하여 주십시오.
- 1번 핀이 GROUND에 연결된 상태로 전원이 인가되면 제어기는 Firmware Download 모드로 전환 되며, 이 때 녹색 LED는 점멸하지 않습니다.
- RXD, TXD는 모터제어기 기준으로 볼 때의 신호명입니다. 따라서 제어기의 TXD는 PC의 RXD에 연결하여야합니다.
- 통신 케이블의 결선은 4.5 통신연결의 Schematic을 참고하십시오.
- Encoder케이블 결선시 핀수가 다를 수 있으니 자세한 내용은 4.4장의 Schematic을 참조하십시오.
- 전원 투입시 운동장치 전류리미트 확인합니다. 전원을 켜면 녹색 LED가 깜박거리기 시작합니다.
- 정상 동작시 0.5s주기로 녹색 LED가 깜박거립니다. 이상이 있을 시 녹색LED의 깜박거림 주기로 error code를 확인 가능하고 통신으로도 error code확인 가능합니다.  
\*error code는 8.2장 표에서 확인 가능합니다.
- 제어전원과 모터구동전원 분리형의 경우는 옵션사항입니다. 일반적인 경우는 공통전원사용의 연결을 하시기 바랍니다.

CUBE - ST Series

## 2.2 통신 설정

- 제어기에 통신 케이블을 연결하고 전원을 인가한 후 제어기와 같이 배포된 CD의 CubeMonLT 프로그램을 설치 후 CubeMonLT.exe를 실행 합니다.

5.6장 참조



-  통신설정 버튼을 실행합니다

- 제품 출하시 제어기의 RS232 설정은 115200bps, ID는 1번으로 설정 되어있습니다.



- 올바른 통신 설정을 확인 하기 위하여 명령어 입력창에 ":"을 반복하여 입력 합니다. ":"을 입력 할 때마다 ";;"이 리턴되어 표시된다면 설정이 올바른 것입니다. ":"에 대하여 아무런 반응이 없다면 통신 설정과 RS485/CAN 통신의 경우 제어기의 ID를 확인 후 재 입력하고 위의 과정을 반복하십시오.

## 2.3 모터 파라미터 설정

- 모터 스펙이 2-PHASE, 1.8°, 0.95A일 때 다음 명령을 순서대로 입력하면 모터 1과 모터 2의 모터 파라미터 설정을 할 수 있다.

※6.2장 참조

	명령어	설명
1	STFFFF;	제어 파라미터 변경이 가능한 타입으로 설정합니다.
2	EDA55A;	파라미터를 factory default 값으로 바꿉니다.
3	SEA55A,0400,0400,0032,0032;	90°/1.8°(스텝각) 50을 핵사값 0032로 설정합니다.
4	EDA55A;	이 과정에서 바뀐 파라미터에 따른 초기화를 실시합니다.
5	SAFE01;	사용하려는 주소로 재설정합니다.
6	STFFFE;	제어 파라미터가 변경되지 않도록 모터 타입을 FFFE로 설정합니다.
7	SH400,400,0,0,0,0;	구동전류의 진폭을 400bit/A로 설정합니다.
8	Sh400,400,8,8;	전류명령값의 최대치를400bit/A로 전류치 비율을 8로 설정합니다.
9	EsA55A;	설정한 파라미터를 저장합니다.

## 2.4 위치 제어

- 아래 표와 같이 순서대로 명령을 입력하면 모터 1과 모터 2를 각각 30000/-100000만큼 위치제어를 실시하며 이때 정속 구간의 속도는 1000/1500rpm이고, 가/감속은 500/800ms 동안 수행 됩니다.

	명령어	설명
1	SS1000,1500;	정속 구간의 이동속도를 모터1은 1000rpm, 모터2는 1500rpm으로 설정합니다.
2	Ss500,800;	가/감속 구간의 시간을 모터1 500ms, 모터2 800ms로 설정합니다.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 위치제어 모드로 설정합니다.
5	PA5030000,4900000;	위치 명령을 모터1 5030000, 모터2 4900000을 전송합니다. 5000000 이 원점입니다. (위치명령의 16384값이 모터의 1회전 값입니다. 명령을 5016384로 주었을 경우 모터가 1바퀴 돕니다.)
6	QP;	현재 포지션을 확인하는 명령어입니다.

## 2.5 속도 제어

- 아래 표와 같이 순서대로 명령을 입력하면 Sa로 설정된 가/감속률로 SV로 설정한 속도[rpm]로 속도제어를 실시합니다.

	명령어	설명
1	Sa400,400,500,400;	100ms당 모터1의 가속도 400rpm 감속도 400rpm 모터2의 가속도 500rpm 감속도 400rpm의 기울기를 갖는 가/감속률을 설정합니다. [rpm/100ms] (※이 설정을 1초단위로 환산하면 4000,4000, 5000, 4000 rpm/sec가 됩니다.)
2	PE0001;	Servo Enable [ID=1로 가정]
3	SM0909;	STEP모터 전용 속도제어 모드로 설정합니다.
4	SV1000,-700;	모터1의 속도를 1000rpm, 모터2의 속도를 -700rpm으로 설정합니다.

## 3. 제품 소개

### 3.1 FEATURES



CUBE-ST2402-DIIE



CUBE-ST2402-DII

## 로봇 및 산업용 STEP 모터 제어기

CUBE-STxxxx-xxx는 STEP 서보모터를 구동하기 위한 제어기로 로봇구동에 적용하도록 초소형으로 설계된 Network 기반의 고성능 실시간 제어기입니다.

- 초소형 사이즈
  - 80mm × 50mm × 15mm (DIIE 버전)
  - 65mm × 40mm × 15mm (DII 버전)
- 다양한 통신방식 지원
  - RS232(~115.2Kbps), RS485(1.25Mbps),
  - CAN(1Mbps)
- Daisy Chain 방식의 다중모터 제어기능
  - RS485, CAN
- 실시간 위치/속도제어 기능
  - 32bit 위치값 / ~30,000[rpm] / ~정격전류
- 위치/속도/전류 명령 Update 주기
  - 2~100ms
- 넓은 전원입력 범위
  - 10V ~ 30V(DIIE 버전)
- 연속최대전류 (추가 대용량 방열판 부착시)
  - 2A
- 자체 고속 Interpolation 기능으로 과도진동 저감
- Micro Stepping 기능(4096/Step)
- Encoder Interface 지원
- 다양한 Protection 기능
  - Over & Under Voltage Protection
  - Over Current Protection
  - Motor Stall Protection
  - Excessive Control Error Protection
- 연속궤적 저장 및 Play 기능과 단독 시퀀스 제어 기능
- 통신주기의 변화에 둔감한 실시간 위치/속도/전류 제어기능
- 100KHz PWM 사용
- 외부 입출력
  - 8 Analog/Digital 겸용 IO Ports (DIIE 버전)
- 임의의 순간에 동작모드, 동작명령, 동작변수의 변경 가능
- 파라미터 확인, 변경, 저장(EEPROM) 기능
- Pulse/Direction 또는 통신방식 구동

3.2 FEATURES & SPECIFICATION



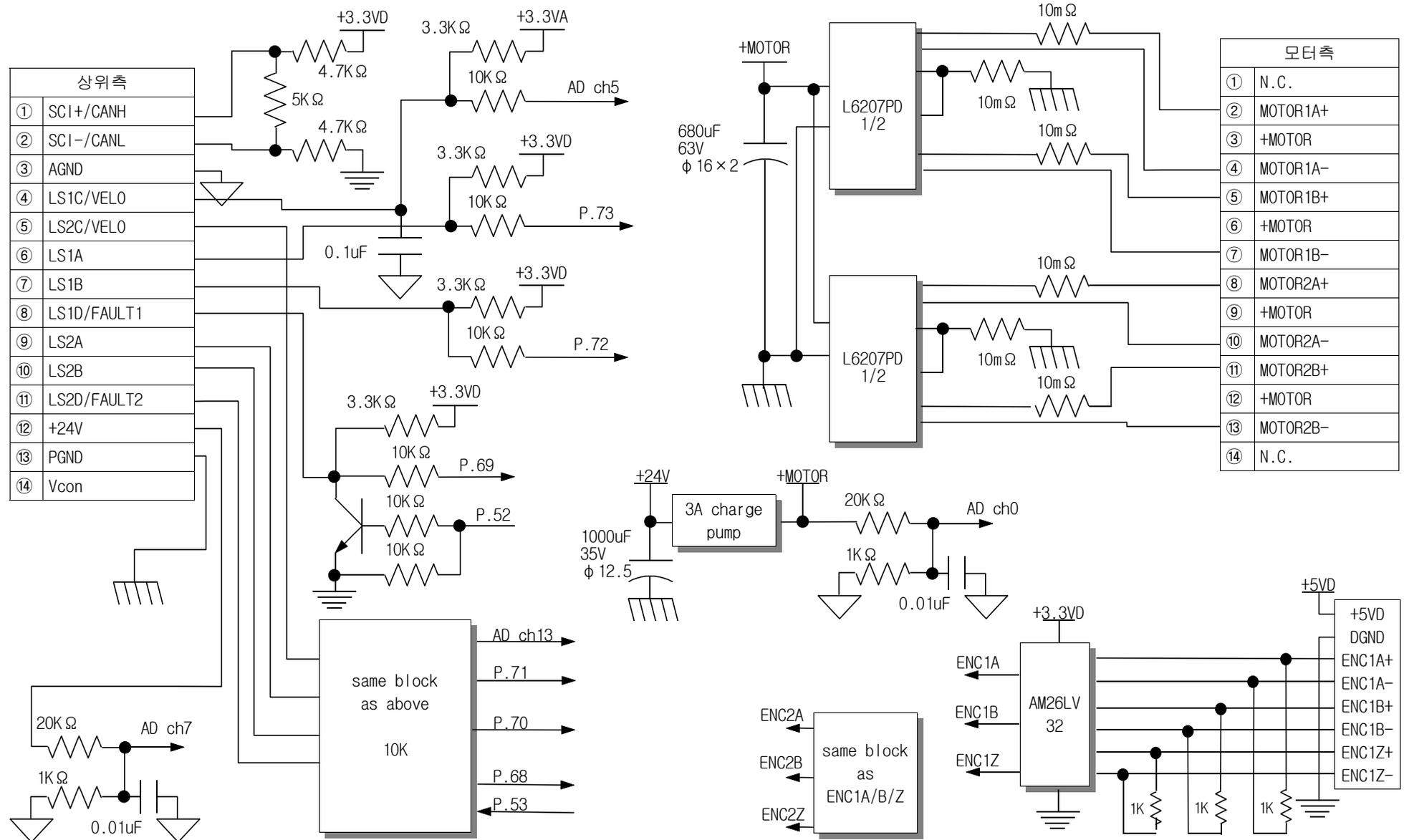
CUBE-ST2402-DIIE

- ➔ Encoder type 순간최대 50W급 Bipolar STEP 모터 2채널 구동
- ➔ Real-Time 위치/속도제어 기능
- ➔ 모터구동칩 L6207PD 채용으로 초소형화 실현
- ➔ 다중모터제어 통신방식(RS232/485/CAN)
- ➔ User Programmable Sequence 제어 기능
- ➔ Micro Stepping 기능(4096/Step)
- ➔ Pulse/Direction 또는 통신방식 구동
- ➔ Encoder Interface 지원
- ➔ 외부입출력 : 8 Analog/Digital I/O port
- ➔ 넓은 전원입력 범위

SECTION	DETAILS	SPECIFICATION	UNIT	
MOTOR	number of motor driver	2	EA	
ELECTRICAL	Minimum supply voltage	10	V	
	Nominal supply voltage	24/36	V	
	Maximum supply voltage	30/45	V	
	continuous current (impulse max current) <sup>(1)</sup>	2(4)	A / ch	
	continuous power	37/54 × 2	W	
	PWM frequency	20 ~ 100	KHz	
	motor coil driving MOSFET	L6207PD		
OPERATION MODE	position control	maximum midway points of continuous pass	192	points
		position counter	32	bit
		period of real-time position command	2~100	ms
		speed set & modify	○	
		acceleraton/deceleration set	○	
	speed control	speed range	0 ~ 3000	RPM
		period of real-time speed command	2~100	ms
		acceleraton/deceleration set	○	
	current control	×		
	SENSOR	encoder(Line drive, Push-pull, Open collector)	○	
analog Hall sensor		×		
INPUT/OUTPUT	analog input or digital I/O or fault output	8	EA	
COMMUNICATION	RS232 speed	9600~115200	bps	
	RS485 speed	upto 1.25	Mbps	
	CAN speed	upto 1	Mbps	
	maximum nodes (RS485 / CAN)	126/2048	nodes	
PROTECTION	Over voltage protection	○		
	Under voltage protection	○		
	Over current protection	○		
	Over speed protection	○		
	Motor stall prorection	○		
	excessive control error protection	○		
TEMPERATURE	operating ambient temperature	-20~50 <sup>(2)</sup>	℃	
MECHANICAL	dimension (without case)	80×50×15	mm (L×W×H)	
	weight (without case)	45	g	

(1). 허용전류는 방열판 용량에 따라 달라질 수 있습니다.  
전류의 크기는 모터권선에 흐르는 전류의 진폭값을 의미합니다.  
(2). 제어기 동작온도는 방열판 용량에 따라 달라질 수 있습니다.

CUBE - ST Series



CUBE-ST2402-DIIE의 입출력 내부회로



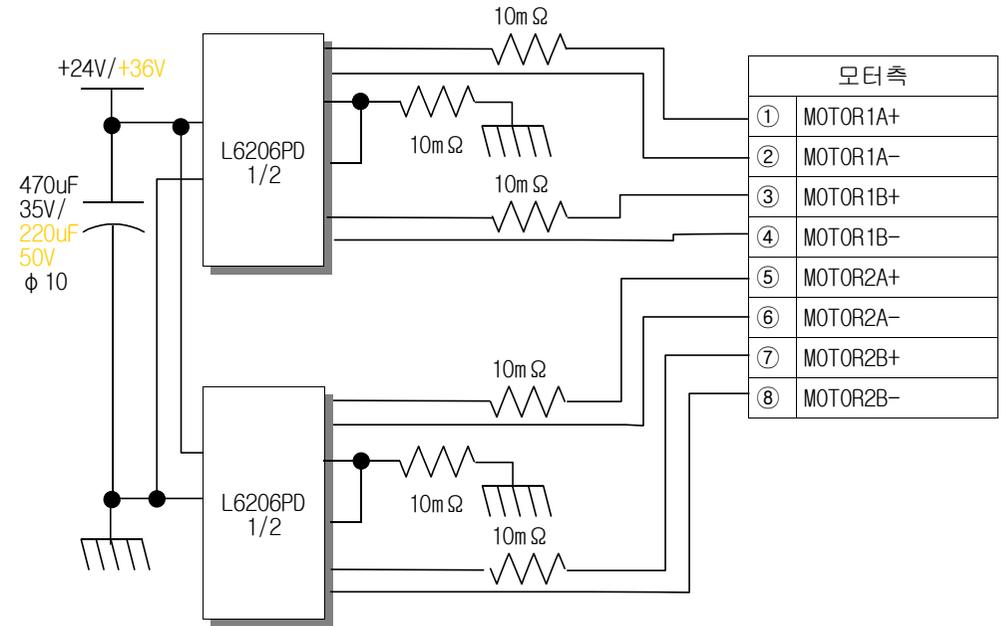
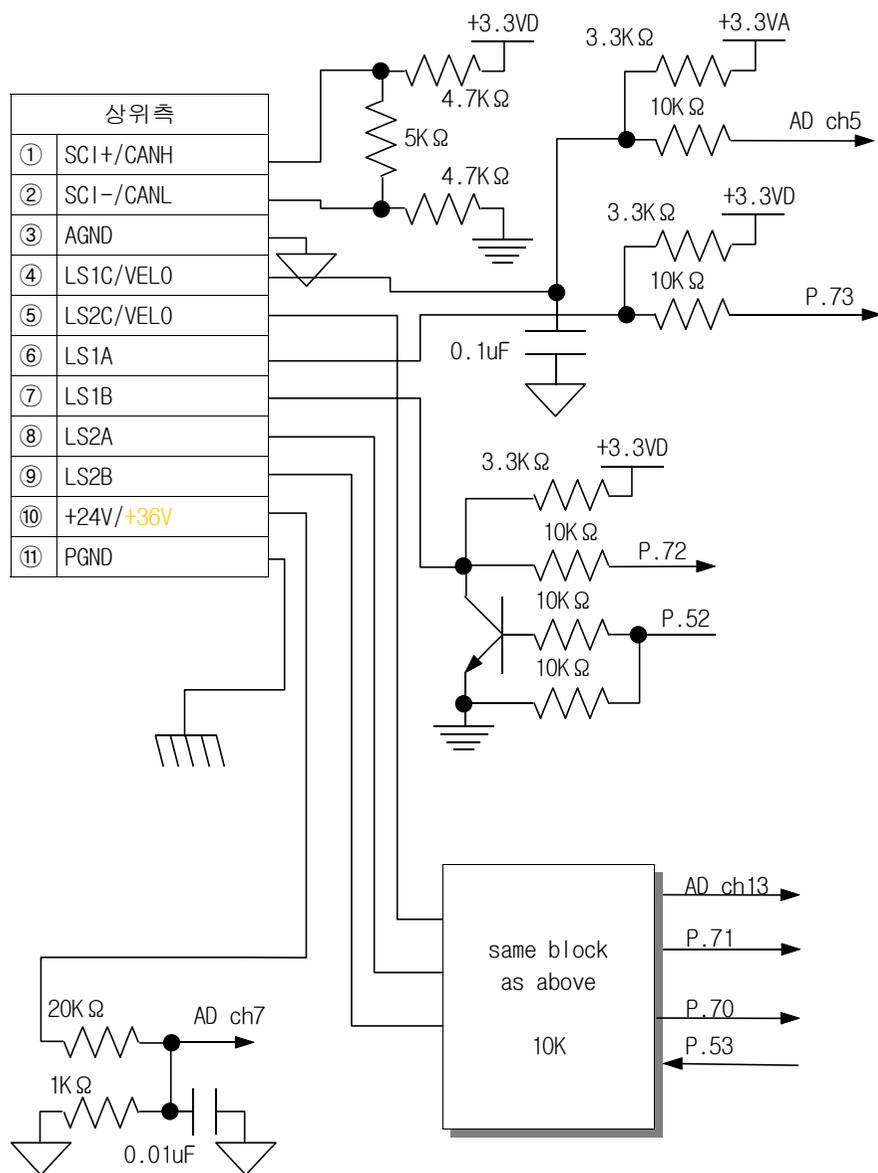
**CUBE-ST2402-DII**

- 순간최대 50W급 Bipolar STEP 모터 2채널 구동
- Real-Time 위치/속도제어 기능
- 모터구동칩 L6206PD 채용으로 초소형화 실현
- 다중모터제어 통신방식(RS232/485/CAN)
- User Programmable Sequence 제어 기능
- Micro Stepping 기능(4096/Step)
- Pulse/Direction 또는 통신방식 구동
- 외부입출력 : 6 Analog/Digital I/O port
- 넓은 전원입력 범위

SECTION	DETAILS		SPECIFICATION	UNIT
MOTOR	number of motor driver		2	EA
ELECTRICAL	Minimum supply voltage		10	V
	Nominal supply voltage		24/36	V
	Maximum supply voltage		30/45	V
	continuous current (impulse max current) <sup>(1)</sup>		2(4)	A / ch
	continuous power		37/54 × 2	W
	PWM frequency		20 ~ 100	KHz
	motor coil driving MOSFET		L6206PD	
OPERATION MODE	position control	maximum midway points of continuous pass	192	points
		position counter	32	bit
		period of real-time position command	2~100	ms
		speed set & modify	○	
		acceleraton/deceleration set	○	
	speed control	auto homing	○	
		speed range	0 ~ 3000	RPM
		period of real-time speed command	2~100	ms
	acceleraton/deceleration set		○	
	current control		×	
SENSOR	encoder(Line drive, Push-pull, Open collector)		×	
	analog Hall sensor		×	
INPUT/OUTPUT	analog input or digital I/O or fault output		6	EA
COMMUNICATION	RS232 speed		9600~115200	bps
	RS485 speed		upto 1.25	Mbps
	CAN speed		upto 1	Mbps
	maximum nodes (RS485 / CAN)		126/2048	nodes
PROTECTION	Over voltage protection		○	
	Under voltage protection		○	
	Over current protection		○	
	Over speed protection		○	
	Motor stall prorection		○	
	excessive control error protection		○	
TEMPERATURE	operating ambient temperature		-20~50 <sup>(2)</sup>	℃
MECHANICAL	dimension (without case)		65×40×15	mm (L×W×H)
	weight (without case)		25	g

(1). 허용전류는 방열판 용량에 따라 달라질 수 있습니다.  
전류의 크기는 모터권선에 흐르는 전류의 진폭값을 의미합니다.  
(2). 제어기 동작온도는 방열판 용량에 따라 달라질 수 있습니다.

CUBE - ST Series



모터측	
①	MOTOR1A+
②	MOTOR1A-
③	MOTOR1B+
④	MOTOR1B-
⑤	MOTOR2A+
⑥	MOTOR2A-
⑦	MOTOR2B+
⑧	MOTOR2B-

CUBE - ST Series

CUBE-ST2402-DII의 입출력 내부회로

# 4. 배선

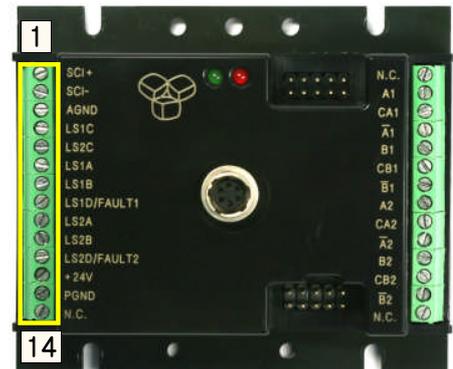
## 4.1 단자대 결선도

### 배선시 유의사항

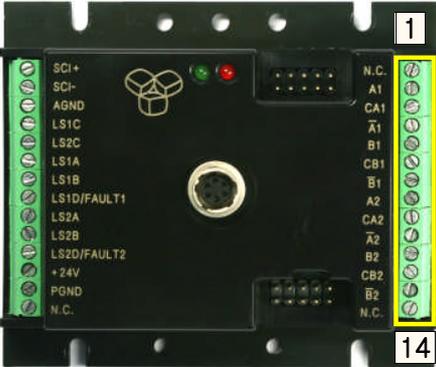
- 전원용 전선은 사용할 모터의 전류용량을 고려하여 충분히 **굵은 선**을 사용해야 합니다.
- 전원용 전선은 전자파 노이즈를 줄이기 위하여 두 선을 **Twist** 하여 사용하는 것을 권장합니다.
- **전원전압이 작용하는 상태에서 전원선을 연결 또는 분리를 하거나, 전원을 거꾸로 연결하거나, 정격전압 이상의 과도한 전압을 인가하는 경우 제어기와 모터가 파손될 수 있으니 주의** 하시기 바랍니다.

### ■ CUBE-ST2402-DIIE

단자 번호	단자 명칭	기능 설명	
1	SCI+/CANH	RS485/CAN 통신의 Positive Signal	
2	SCI-/CANL	RS485/CAN 통신의 Negative Signal	
3	AGND	Analog Ground	
4	LS1C/VELO	-리미트센서입력 or Analog Input [0-3V]	
5	LS2C/VELO	-리미트센서입력 or Analog Input [0-3V]	
6	LS1A	+리미트센서 입력 or Digital Input	
7	LS1B	홀센서 입력 or Digital Input	
8	LS1D/FAULT1	입력	Digital Input
		출력	Open Collector Fault Output
9	LS2A	+리미트센서 입력 or Digital Input	
10	LS2B	홀센서 입력 or Digital Input	
11	LS2D/FAULT2	입력	Digital Input
		출력	Open Collector Fault Output
12	+24V	+30V 이하의 모터구동용 및 +30V 이하의 제어용 전원입력	
13	PGND	Power Ground	
14	N.C/Vcon	N.C	Reserved
		Vcon	+30V 이하의 제어용 전원입력



CUBE - ST Series

사진	단자 번호	단자 명칭	기능 설명
	1	N.C.	Reserved
	2	MOT A1	Motor 1 A상 Positive
	3	N.C.	사용하지 않음
	4	MOT $\bar{A}$ 1	Motor 1 A상 Negative
	5	MOT B1	Motor 1 B상 Positive
	6	N.C.	사용하지 않음
	7	MOT $\bar{B}$ 1	Motor 1 B상 Negative
	8	MOT A2	Motor 2 A상 Positive
	9	N.C.	사용하지 않음
	10	MOT $\bar{A}$ 2	Motor 2 A상 Negative
	11	MOT B2	Motor 2 B상 Positive
	12	N.C.	사용하지 않음
	13	MOT $\bar{B}$ 2	Motor 2 B상 Negative
	14	N.C.	Reserved

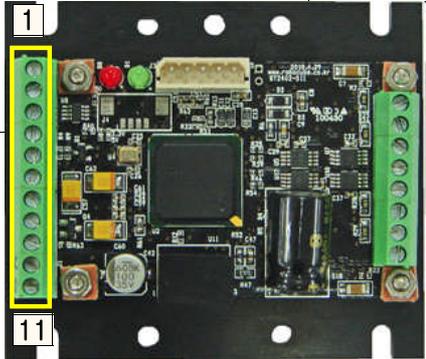
■ CUBE-ST2402-DII			
단자 번호	단자 명칭	기능 설명	사진
1	SCI+/CANH	RS485/CAN 통신의 Positive Signal	
2	SCI-/CANL	RS485/CAN 통신의 Negative Signal	
3	AGND	Analog Ground	
4	LS1C/VELO	-리미트센서 입력 or Analog Input [0-3V]	
5	LS2C/VELO	-리미트센서 입력 or Analog Input [0-3V]	
6	LS1A	+리미트센서 입력 or Digital Input	
7	LS1B	홀센서 입력 or Digital Input	
8	LS2A	+리미트센서 입력 or Digital Input	
9	LS2B	홀센서 입력 or Digital Input	
10	+24V/+36V	+30V/45V 이하의 모터구동용 및 +30V 이하의 제어용 전원입력	
11	PGND	Power Ground	

사진	단자 번호	단자 명칭	기능 설명
	1	MOT A1	Motor 1 A상 Positive
	2	MOT $\bar{A}$ 1	Motor 1 A상 Negative
	3	MOT B1	Motor 1 B상 Positive
	4	MOT $\bar{B}$ 1	Motor 1 B상 Negative
	5	MOT A2	Motor 2 A상 Positive
	6	MOT $\bar{A}$ 2	Motor 2 A상 Negative
	7	MOT B2	Motor 2 B상 Positive
	8	MOT $\bar{B}$ 2	Motor 2 B상 Negative

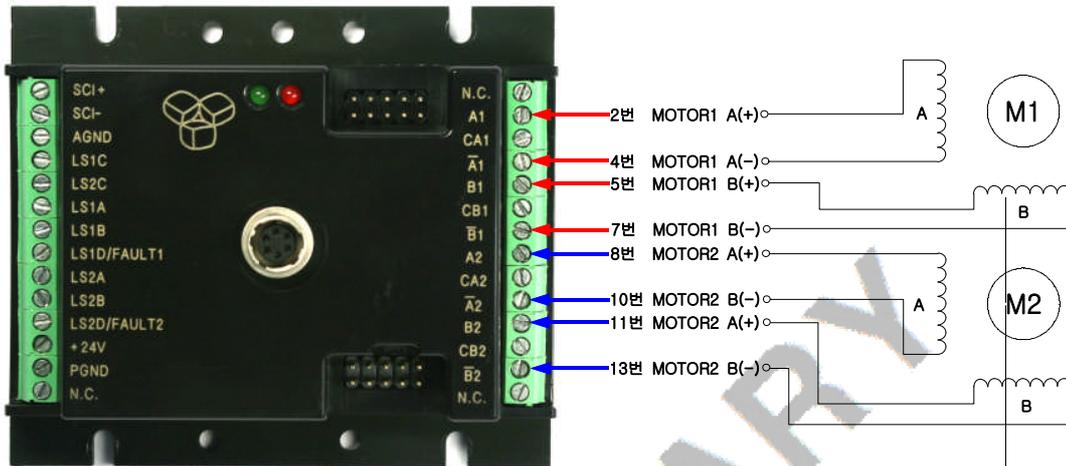
## 4.2 전원 연결

- **모터 제어전원과 구동전원을 같이 사용한 경우** (단자 14번 핀에 N.C 표기가 있어야 가능)
  - 이 경우에는 모터구동 및 모터제어용 전원은 동일한 전원을 사용해야 합니다. 좌측단자대의 12번 핀에 +10V ~ +30V 크기의 전압을 공급하고, 13번 핀에 Ground 선을 연결 하십시오.
  
- **모터 제어전원과 구동전원을 분리 사용한 경우** (단자 14번 핀에 Vcon 표기가 있어야 가능)
  - 이 경우에는 모터구동 및 모터제어용 전원은 다르게 사용해야 합니다. 좌측단자대의 12번핀에 모터구동용 +10V ~ +30V 크기의 전압을 공급해야 하고, 12번 핀에는 모터제어용 +10V ~ +30V 크기의 전압을 공급하며, 13번 핀에 Ground 선을 연결 하십시오.

### 4.3 Motor Winding 연결

■ 모터의 (+), (-) 단자의 순서를 확인하십시오. 잘못된 결선은 제어기와 모터의 파손 원인이 됩니다.

#### ■ CUBE-ST2402-DIIE

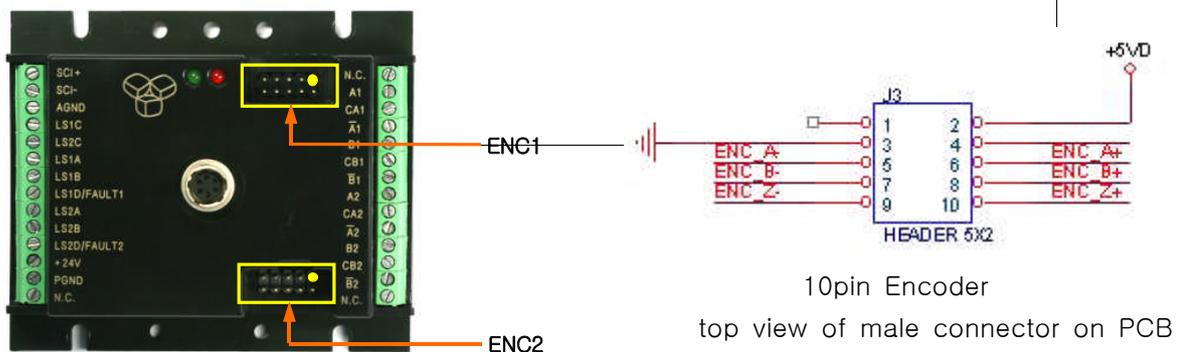


### 4.4 Encoder 연결

- Encoder의 정확한 결선을 위해 아래 Schematic을 참고하여 주십시오. ENC\_Z 상은 연결하지 않아도 됩니다. 잘못된 결선은 제어기와 모터의 파손 원인이 됩니다.
  - Push-Pull 출력형이나 Open collector 출력형인 경우는 ENC\_A+, ENC\_B+, (ENC\_Z+) 단자를 사용하며, 전류가 많이 흐르는 경우는 엔코더 신호에 노이즈가 발생할 수 있어서 Differential 타입의 엔코더를 권장합니다.
- 아래 엔코더 그림의 핀중에 노란점이 1번핀이니 연결에 착오없으시길 바랍니다.

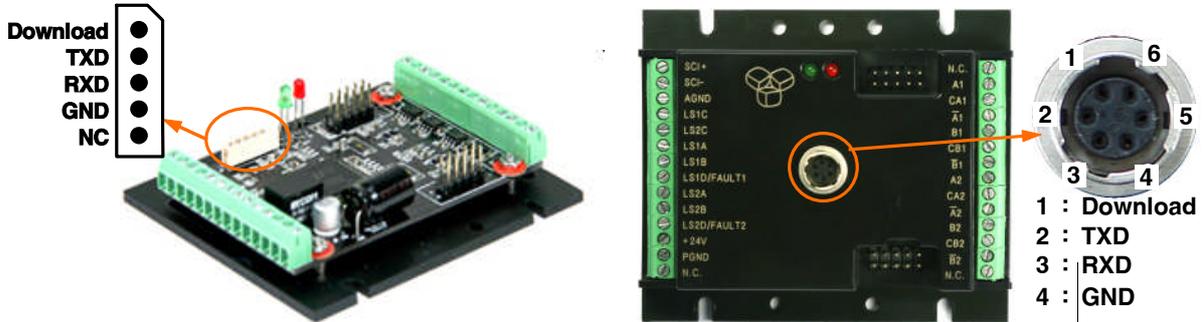
#### ■ CUBE-ST2402-DIIE

#### ■ Encoder Schematic

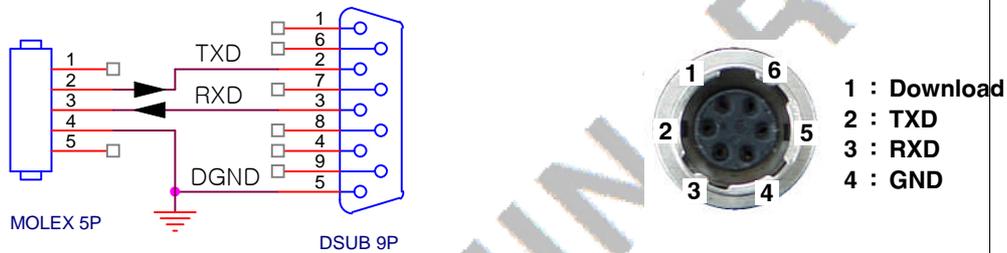


4.5 RS232 통신 연결

■ CUBE-ST24xx-xxxx without CASE ■ CUBE-ST24xx-xxxx with CASE



■ RS232 Schematic

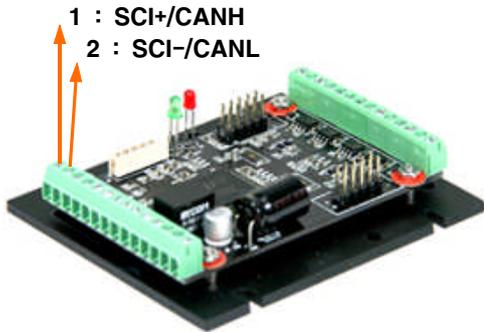


- 5Pin RS232 커넥터의 1번 핀은 Firmware Download를 위해 할당된 핀으로서, 일반적인 통신용으로 사용시에 1번 핀은 연결되지 않도록 하여 주십시오.
- 1번 핀이 GROUND에 연결된 상태로 전원이 인가되면 제어기는 Firmware Download 모드로 전환 되며, 이 때 녹색 LED는 점멸하지 않습니다.
- RXD, TXD는 모터제어기 기준으로 볼 때의 신호명입니다. 따라서 제어기의 TXD는 PC의 RXD에 연결하여야합니다.
- 통신 케이블의 결선은 위 Schematic을 참고하십시오.
- 시리얼 케이블은 초기 세팅용으로 1개만 보내드리고 있고 추가로 필요하신 경우 본사측에서 추가 구매하시거나 부속품을 직접 구매, 제작하시는 방법이 있습니다. (SN86PM or MOLEX 5pin + 케이블 + D-SUB 9pin FEMALE 솔더타입 + D-SUB CASE 9pin용)

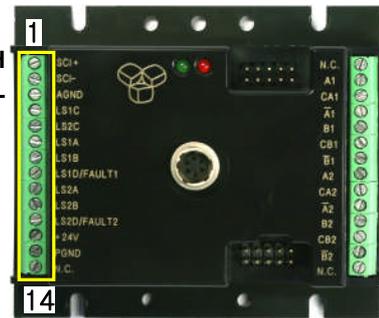
CUBE - ST Series

4.6 RS485 또는 CAN 통신 연결

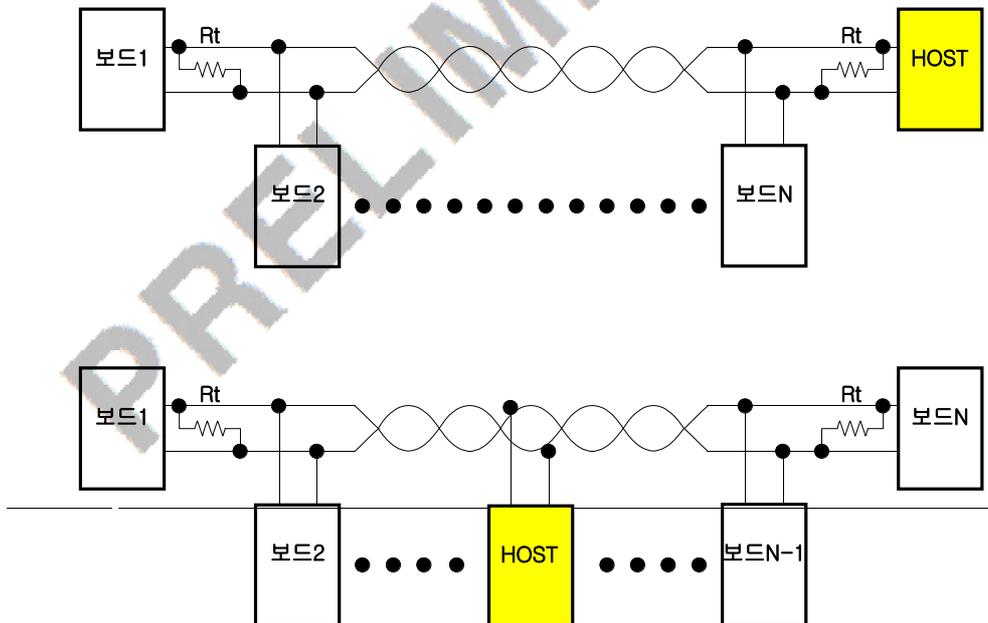
■ CUBE-ST24xx-xxxx without Case ■ CUBE-ST24xxx-xxx with Case



1 :SCI+/CANH  
2 :SCI-/CANL



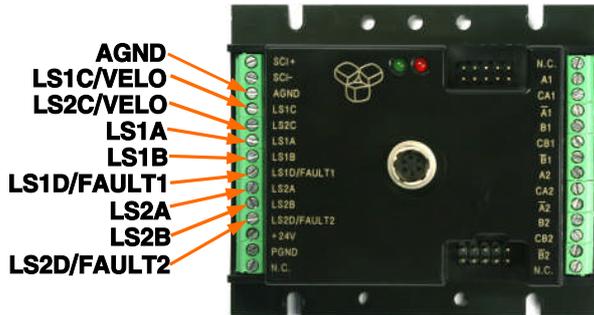
- 2개 이상의 제어기를 Daisy-Chain 방식으로 호스트 컨트롤러(PC포함)에 연결하여 사용하는 경우, 통신선의 양쪽 끝단에 종단저항(Rt)을 달아 사용하시기 바랍니다.
- 종단저항(Rt)의 값은 120Ω을 권장합니다.
- RS485/CAN 통신용 케이블은 Twisted & Shield Cable을 권장합니다.
- 통신 케이블의 결선은 아래 그림을 참고하십시오.
- 모터제어기를 10개 이상 연결하고자 하면 종단저항 설정과 관련하여 문의를 바랍니다.
- 각 모터제어기 및 Host Controller간의 전위차가 12V이상이 되는 경우 해당 모터제어기 및 Host Controller의 CAN통신 IC손상의 원인이 되므로 Device간의 Ground를 Common하여 주십시오.



CUBE - ST Series

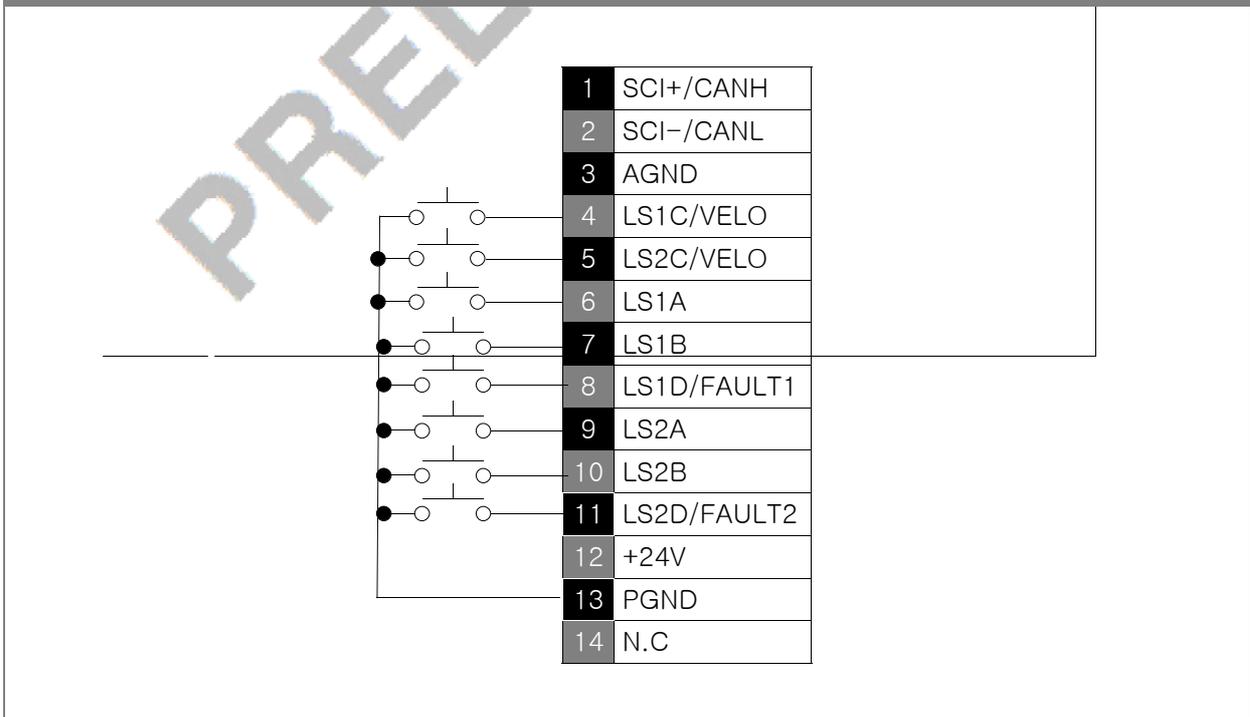
4.7 DIGITAL/ANALOG INPUT/OUTPUT 연결(LIMIT 스위치 연결)

CUBE-ST24xx-xxxx



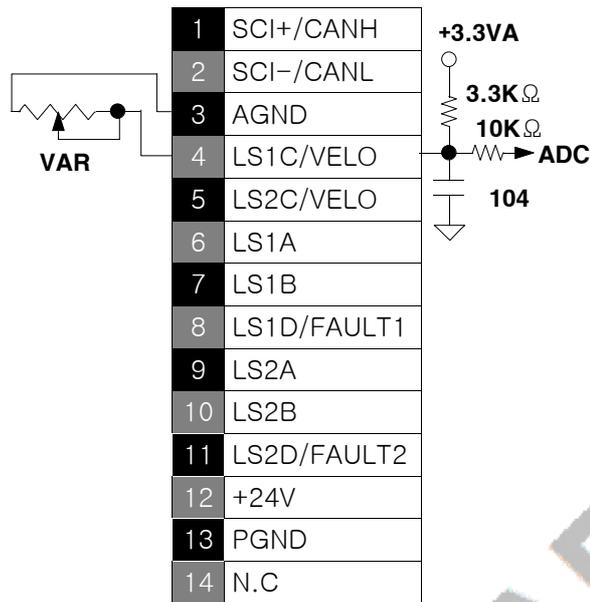
- CUBE-ST24xx-xxx 제어기는 2개의 입력 포트와 1개의 입/출력 겸용 포트를 가지고 있습니다.
- LSxD/FAULTx는 Digital 입력 또는 Fault status를 출력하는 출력 포트에 사용 할 수 있습니다. Fault status 출력은 Fault가 발생하였을 때 자동으로 Fault 상태를 출력하는 용도로 쓰고 있지 않으며, XO 명령어로 강제 출력하거나 또는 시퀀스 제어시에 사용할 수 있습니다.
- LSxC/VELO는 -리미트센서입력 또는 아날로그 입력을 위한 포트입니다
- LSxB는 홀센서입력 또는 디지털 입력 포트에 사용 할 수 있습니다.
- LSxA는 +리미트센서입력 또는 디지털 입력 포트에 사용 할 수 있습니다.
- LSxA, LSxB, LSxC는 제어기의 홈 포지션 탐색용 센서 입력으로 사용 할 수 있습니다. 홈 동작에 관련된 자세한 내용은 7.11절을 참고하여 주십시오.
- 아래의 사용 예를 참고하여 주십시오.

CUBE-ST24xx-Dxxx SERIES (모든 포트를 디지털 입력으로 사용할 때)

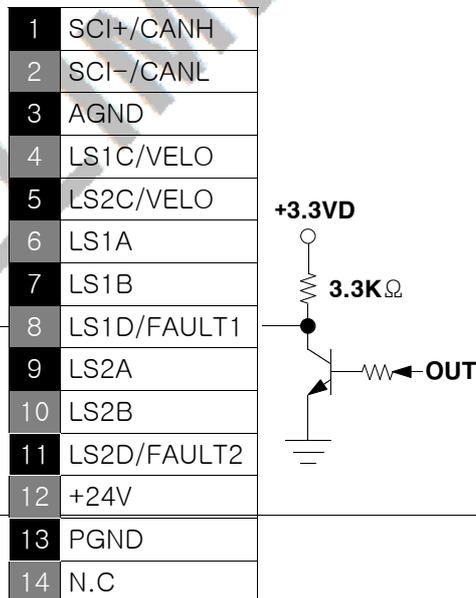


CUBE - ST Series

CUBE-ST24xx-Dxxx SERIES (LSC를 아날로그 입력으로 사용할 때)



CUBE-ST24xx-Dxxx SERIES (LSxD를 디지털 출력으로 사용할 때)



## 5. 통신 프로토콜

### 5.1 통신 개요

#### ■ 프로토콜

- 로보큐브테크의 제어기와 통신하기 위해서는 제어기의 프로토콜을 준수하여 통신을 해야 합니다. 로보큐브테크의 모든 제어기는 하나의 프로토콜을 사용하지만 명령어마다 그 포맷이 다르고 Binary 형태의 데이터와 ASCII 형태의 데이터를 혼합하여 사용하기 때문에 각 명령에 대하여 첨부된 별도의 명령어 리스트 파일을 확인 후 사용 하시기 바랍니다.

#### ■ 통신 패킷의 구성 [상위제어기가 모터제어기로 보내는 명령어 형식]

구분	ID + COMMAND + DATA + ; or !	
ID	제어기의 고유 식별 번호입니다. RS232 통신에서는 ID가 필요하지 않으며, CAN 통신의 경우는 CAN 통신 controller에서 ID를 별도 처리하므로 ID는 CAN통신의 데이터 영역을 점유하지는 않습니다. ID는 1~126(0X01~0X7E)까지 126개를 사용할 수 있습니다.	
COMMAND	미리 약속된 명령어입니다. 명령어는 ASCII 코드문자를 사용합니다.	
DATA	명령어에 수반되는 데이터입니다. 데이터는 ASCII 코드문자와 Binary 코드를 모두 사용하며 콤마(',')로 구분되기도 합니다.	
; or !	통신 패킷의 마지막을 의미하는 값입니다.	

- RS485통신의 경우 각 제어기의 응답이 필요 없는 경우에는 통신시간의 절감을 위해 해당 제어기의 ID에 128(0x80)을 더해 129~254(0x81~0xFE)의 범위에서 사용할 수 있습니다. 예를 들어 ID를 0x82를 사용하여 명령어를 송신하면 ID 2번 제어기가 명령어를 수신하지만, 제어기로부터의 리턴값은 없습니다.
- RS485/CAN통신의 경우 호스트(모션제어기/PC)에서 제어기 ID의 값으로 255(0xFF)를 사용하여 명령어를 전송하는 경우에는 모든 제어기가 명령어를 수신하게 되어, 결과적으로 명령어를 Broadcasting 한 것이 됩니다.
- RS232 통신의 경우 1:1 통신만이 가능하므로 ID의 구분없이 사용하지만, RS485/CAN 통신의 경우에는 ID를 지정하여 통신을 해야만 합니다. 아래의 사용 예는 같은 명령에 대한 RS232 통신과 RS485/CAN 통신의 차이점을 설명합니다.

## 5.2 RS232 통신

- RS232 통신은 8bit, 1 Stop bit, None Parity, Asynchronous Serial 통신을 사용합니다.
- 공장 출하시의 Baudrate은 115200bps 입니다. Baudrate을 변경 하려면 "SB" 명령을 사용하면 됩니다. [9600,19200,38400,57600,115200] "EsA55A;" 명령어를 사용하면 EEPROM에 저장할 수 있으며, 현재의 Baudrate를 알고 싶으면 명령창에(5.6 절의 메뉴설명 참조) "SB?;" 입력하여 에코 수신창에서(5.6절의 메뉴설명 참조) 확인할 수 있습니다.
- 통신 패킷의 마지막을 의미하는 값이 ';'이면 host에서 제어기로 보낸 명령어의 처리에 따른 return값이 에코 수신창에 보이며, '!' 이면 명령어의 처리에 따른 return값이 없습니다.

### ■ RS232 통신 예

- 다음은 호스트와 RS232 통신 케이블로 연결된 제어기의 ID를 확인하는 송수신의 예입니다.

	송신 명령	응답	설명
ASCII	SA?;	SAFE01;	제어기 자신의 ID가 01번임을 의미 합니다.
HEX	53 41 3F 3B	53 41 46 45 30 31 3B	

#### 참 고

- S로 시작하는 명령어 2글자 뒤에 물음표('?')를 붙이면 해당 명령어로 설정된 값을 확인할 수 있습니다. P, G, W, X로 시작되는 명령어도 2글자 뒤에 물음표('?')를 붙이면 해당 명령어로 설정된 값을 확인할 수 있습니다. 자세한 내용은 명령어 리스트를 참조하시기 바랍니다.

### ■ RS232 통신 패킷 구성 예 : **SV2000,3000;**

내용	구분	사이즈	설 명
SV	ASCII	2byte	속도제어시 속도 설정 명령.
2000,3000	ASCII	9byte	1채널 모터속도 2000[rpm], 2채널 모터속도 3000[rpm]
;	ASCII	1byte	통신패킷의 마지막 값.

### 5.3 RS485 통신

- 공장 출하시의 RS485 Baudrate은 1.25Mbps입니다. "Sb" 명령을 사용하여 Baudrate을 변경할 수 있습니다. EsA55A; 명령어를 사용하면 EEPROM에 저장할 수 있습니다.
- RS485의 경우 모든 명령어의 앞에 1BYTE의 제어기 ID를 추가 해야 하며, 제어기의 ID는 "SA" 명령어를 사용하여 설정할 수 있습니다. 공장 출하시의 제어기 ID는 0x01 이며, "SA?;"를 명령어 입력창에(5.6절 메뉴설명 참조) 입력하시면 현재 제어기의 ID를 확인할 수 있습니다.
- RS485 통신에서는 하나의 패킷 내에서 각 데이터 바이트와 바이트 사이의 지연시간이 한바이트 송신시간 이상인 경우, 잘못된 데이터로 간주하여 무시되므로 **Semicolon[;]** 이전의 모든 데이터 바이트 사이에는 한바이트 송신시간 이상의 지연이 발생하지 않도록 해야 합니다.
- 존재하지 않는 제어기의 ID로 시작하는 데이터는 모두 무시 됩니다.
- RS485의 경우 제어기 ID를 0xFF로 주면 모든 제어기가 명령패킷을 수신합니다. [Broadcasting]
- RS485의 경우 제어기 ID를 0x80 + 제어기 ID로 주면 리턴이 없는 명령어가 됩니다.
- 호스트가 보낸 명령패킷의 마지막 값을 ';' 대신에 '!'를 사용하면 제어기로부터의 리턴값은 없습니다.

송신	RETURN	설명
(0x01)PA5001000,5001000;	(0x00)PA5001000,5001000;	ID 1번 제어기의 각 채널 포지션을 5001000으로 설정하고 응답이 있음.
(0x81)PA5001000,5001000;	NONE	ID 1번 제어기의 각 채널 포지션을 5001000으로 설정하고 응답이 없음.

#### ■ RS485 통신을 위한 준비

- RS485 통신을 위해서는 아래와 같은 준비가 필요합니다.  
RS485 통신용 젠더 케이블(RS232 to RS485 또는 USB to RS485 또는 PCI to RS485)
- HOST(PC)용 Program은 당사 홈페이지 자료실을 참고하시기 바랍니다.

#### ■ RS485 통신 예

- 호스트 ID는 "Sm" 명령을 사용하여 설정할 수 있습니다. 다음은 제어기의 ID가 0x01, 호스트 ID가 0x00 인 경우의 RS485 송수신의 예입니다.

	송신 명령	응답	설명
ASCII	(0x01)SA?;	(0x00)SAFE01;	제어기 자신의 ID가 01번임을 의미 합니다.
HEX	01 53 41 3F 3B	00 53 41 46 45 30 31 3B	

#### ■ RS485 통신 패킷 구성 예 : (0x01)SV2000,3000;

내용	구분	사이즈	설명
01	binary	1byte	명령을 수행할 대상 제어기의 ID
SV	ASCII	2byte	속도제어시 속도 설정 명령.
2000,3000	ASCII	9byte	1채널 모터 속도 2000[rpm], 2채널 모터속도 3000[rpm]
;	ASCII	1byte	통신 패킷의 마지막 값.

## 5.4 CAN 통신

### ■ CAN 요약

- 공장 출하시의 CAN 통신의 속도는 1Mbps 입니다. "Sb" 명령어를 사용하여 Baudrate을 변경할 수 있습니다. EsA55A; 명령어를 사용하면 EEPROM에 저장할 수 있습니다.
- 표준 CAN(버전 2.0A) 11비트 ID를 사용합니다.
- 호스트가 보낸 명령패킷의 ID와 일치하는 제어기가 존재하지 않는 경우에는 호스트가 통신에러 복구를 해주어야 합니다.
- 호스트가 보낸 명령패킷의 ID가 0xFF이면 모든 제어기가 명령패킷을 수신합니다. [Broadcasting]
- 호스트가 보낸 명령패킷의 마지막 값을 ';' 대신에 '!'를 사용하면 제어기로부터의 리턴값은 없습니다.
- 제어기로부터 전송되는 리턴값의 ID는 호스트 ID를 나타내며, 제품출하시의 호스트 ID는 00 입니다. 호스트 ID는 "Sm" 명령어를 사용하여 설정 할 수 있습니다.

명령어	파라미터	설명
Sm	HLHL	호스트 ID를 설정하는 명령어. 상위 바이트는 하위 바이트의 보수. SmFE01:(ID=1), SmFD02:(ID=2), SmF50A:(ID=10), SmEE11:(ID=17)

### ■ CAN 통신을 위한 준비

- CAN 통신을 위해서는 아래와 같은 준비가 필요합니다.  
CAN 통신용 젠더 케이블(RS232 to CAN 또는 USB to CAN 또는 PCI to CAN)
- HOST(PC)용 Program은 당사 홈페이지 자료실을 참고하시기 바랍니다.

### ■ CAN 통신 패킷 구성 예 : (0x01)SV2000,3000;

Message Frame	내용	사이즈	설명
Arbit Field	1	11bit	CAN 통신패킷의 ID영역에서 별도 처리
Data Field	SV2000,3	8byte	속도제어시 속도 설정 명령. 8byte 이상 다음 Data Field에 보냄.
Data Field	000;	8byte	세미콜론[;] 종료문자로 제어기에서 명령어 처리

## 5.5 RS485/CAN 통신 선택방법

- RS232 통신은 RS485/CAN통신과 무관하게 독립적으로 사용이 가능합니다.
- RS485/CAN 통신은 선택적으로 사용 가능합니다.
- 공장 출하 시 기본 통신모드는 RS485입니다.
- RS485, CAN통신은 "SX" 명령어를 이용하여 모드를 설정할 수 있습니다. "SX" 명령어 파라미터의 자세한 내용은 6.1절을 참고하시기 바랍니다.
  1. RS485 "SX0071;"
  2. CAN "SX0070;"

주의	
■	모든 파라미터는 설정 후 "EsA55A;" 명령어로 내장된 EEPROM에 저장할 수 있습니다.
■	"SX" 명령은 통신 방식 이외에도 LIMIT 스위치 설정 등 다양한 기능이 포함되어 있으니 꼭 확인 후 사용하시기 바랍니다.

■ 통신 관련 명령어

명령어	파라미터	설명	초기값 <sup>[주2]</sup>	
SB	dddd <sup>[주1]</sup>	RS232통신 속도를 설정함. (9600, 19200, 38400, 57600, 115200)	115200	
Sb	dddd	RS485통신 속도를 설정함. (9600, 19200, 38400, 57600, 115200, 625000, 1250000)	1250000	
SX	HLHL HLHL	bit0	0 : CAN 통신을 선택함, 1 : RS485 통신을 선택함	00000071
		bit1	0 : 직선보간 기능 사용안함, 1 : 사용함	
		bit2	0 : 모터권선의 온도보호 기능의 설정 안함, 1 : 설정함	
		bit3	0 : 모터의 입력/출력 power, 출력토크, 효율, 모터권선/하우징의 온도계산의 설정 안함, 1 : 설정함	
		bit4 ~ 6	010 : 통신테스트 모드설정	
		bit7	0 : LSA, LSC를 리미트센서로 사용함, 1 : LSA, LSB를 리미트센서로 사용함.	
		bit8	0 : Dumy Motor 구동을 안함, 1 : 구동함 <sup>[주3]</sup>	
		bit9	0 : 속도제어시 정지중 적분항의 Zero 처리를 안함, 1 : 실시함	
		bit10	0 : 가상 위치 리미트 제한을 사용함, 1 : 사용 안함 <sup>[주4]</sup>	
		bit11	0 : 홀센서 파라미터 재계산 안함, 1 : 재계산하여 반영함	
		bit12	0 : 속도제어시 리미트센서를 사용 안함, 1 : 사용함	
		bit13	0 : LSxC입력단자의 전압에 따라서 속도명령 또는 전류명령을 설정안함. 엔코더가 있는 스텝모터의 경우는 위치명령도 엔코더에 의하여 설정됨, 1 : 설정함	
		bit14	0 : 연속 시퀀스 제어를 실시함, 1 : 스텝 시퀀스제어 실시함	
		bit15	0 : 시퀀스제어를 안함, 1 : 실시함	
		bit16	0 : 모터1엔코더 A/B상신호 안바꿈, 1 : 바꿈	
		bit17	0 : 모터2엔코더 A/B상신호 안바꿈, 1 : 바꿈	
		bit18	0 : 스텝핑모터 승압회로 사용안함, 1 : 사용함	
		bit19	0 : 스텝핑모터1의 Lead Angle 제어 안함, 1 : 실시함	
		bit20	0 : 스텝핑모터2의 Lead Angle 제어 안함, 1 : 실시함	
		bit21	0 : 저속에서 전류스위칭 소음을 줄이기 위하여 전류 제어 게인을 모터 속도에 따라서 가변하는 기능을 사용함 1 : 사용안함	
bit24	0 : 위치제어에서 설정속도가 과도하여 위치제어 오차가 매우 커질 때 발생하는 문제를 처리함 1 : 처리안함			
bit25	0 : 1ms main loop의 기동을 내부타이머로 실시함 1 : 1ms main loop의 기동을 CAN 데이터가 수신될때 실시함 (DC 모터로서 CAN 통신에서만 가능)			
bit26	0 : 속도제어에서 감속에 따른 역기전력을 감소시키는 기능을 사용함 1 : 사용안함			
bit27	0 : 위치제어에서 PA/PB/Pa 명령수행시에 감속에 따른 역기전력을 감소시키는 기능을 사용하지 않음 1 : 사용함			

		bit28	0 : BLDC 모터제어에서 BLOCK COMMUTATION 방식을 사용안함 1 : 사용함	
		bit31	0 : 데모작동 안함, 1 : 실시함.	
Sm	HLHL	호스트 ID를 설정함, 상위 바이트는 하위 바이트의 보수임		FF00
		예	SmFE01;(ID=1), SmFD02;(ID=2), SmF50A;(ID=10)	
SA	HLHL	제어기 ID를 설정함, 상위 바이트는 하위 바이트의 보수임		807F
		예	SAFE01;(ID=1), SAFD02;(ID=2), SAF50A;(ID=10)	
Es	A55A	설정된 파라미터 값을 EEPROM으로 저장함.		

- [주1] dddd는 십진수이며, HLHL은 16bit 값을 HEX ASCII로 표현한 값입니다.
- [주2] 초기값은 EDA55A; 에 의하여 설정되는 기본값입니다.
- [주3] DC/BLDC 2채널용 제어기에만 해당되며, 1번째 채널의 PWM 명령을 2번째 채널 출력 PWM으로 Copy 하여 사용하는 기능으로서 2대의 모터를 병렬 운전할 수 있습니다.
- [주4] 호스트의 포지션 명령이 "SL" 명령으로 설정된 모터위치 범위를 벗어나는 경우, 제한 여부를 설정합니다.

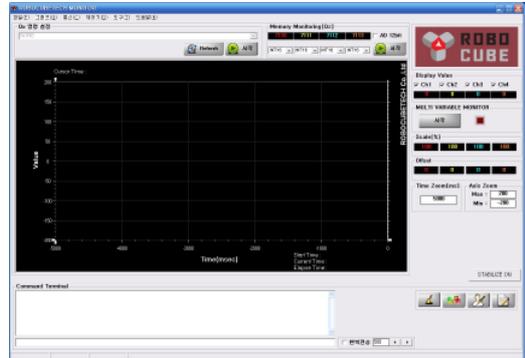
PRELIMINARY

CUBE - ST Series

## 5.6 HOST 프로그램 설치 및 실행

### ■ HOST 프로그램 설치 및 실행

- 제어기에 통신 케이블을 연결하고 전원을 인가한 후 제어기와 같이 배포된 CD의 CubeMonLT 프로그램을 설치 후 CubeMonLT.exe를 실행 합니다.



- Vista일 경우 CubeMonLT.Ver.1.0.0.17.GEN의 셋업 파일에서 마우스의 오른쪽 버튼을 눌러 관리자 권한으로 실행합니다.



■ 메뉴 설명



■ 기능

1. 통신 설정 기능 ( 메뉴설명 ⑧ )
2. 명령 typing과 명령어 전송에 따른 제어기의 응답 표시기능. ( 메뉴설명 ⑤, ⑥ )
3. Qz (변수의 주소 지정) 명령에 의한 실시간 모니터 기능. ( 메뉴설명 ③, ① )
4. Qx (예약된 변수) 명령에 의한 실시간 모니터 기능. ( 메뉴설명 ④, ②, ① )
5. 모터 제어기 SINE파 구동과 모니터 기능
6. PP 명령에 의한 실시간 제어기능

■ 통신 연결

-  통신설정 버튼을 실행합니다
- 제품 출하시 제어기의 RS232 설정은 115200bps, ID는 1번으로 설정 되어있습니다.

\* 통신 설정과 통신 컨넥터 연결의 좀 더 자세한 내용은 4장의 4.5절을 참고 하시기 바랍니다.

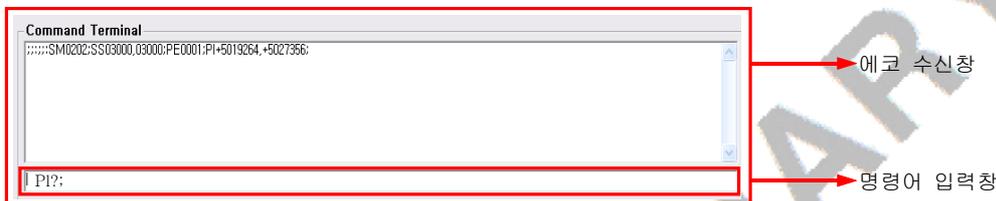


- 올바른 통신 설정을 확인 하기 위하여 명령어 입력창에 ";"을 반복하여 입력 합니다. ";"을 입력 할 때마다 ";;"이 리턴되어 표시된다면 설정이 올바른 것입니다. ";"에 대하여 아무런 반응이 없다면 다음과 같이 확인 합니다. 485/CAN 통신의 경우 보드 ID를 확인합니다.

1. ComPort Set창에서 통신 속도 설정 확인.
2. ComPort Set창에서 통신 포트와 통신 방식 설정 확인.
3. 통신 컨넥터 연결상태 확인.
4. 제어기의 전원 투입여부 확인.
5. 제어기의 초록색 LED 깜박거림 여부 확인.
6. ";"입력 후 리턴 되는지 여부 확인.

### ■ 명령어 전송 후 전송상태 확인하기

- 명령어 입력창에 명령어를 입력하고 ;을 누르면 에코 수신창에 명령어를 리턴받아 확인할 수 있습니다. 아래의 그림은 위치제어 모드를 명령어 입력창에 입력하고, 에코 수신창에 리턴받은 예입니다.



- 모터를 제어하기 위한 각종 명령어는 명령어 리스트를 따로 첨부합니다.

### ■ Qz 명령에 의한 실시간 모니터링 하기

- Qz 명령은 사용자가 정의한 어드레스의 변수를 모니터링 하는 방법입니다.
- 관련 명령어 (Qz / QZ)
- 사용 예

1. 아래 그림에서처럼 사용자가 원하는 어드레스를 지정합니다.



\*어드레스는 ROBOCUBE TECH에서 제공하는 robot.map을 참고하시기 바랍니다.

2. 각 변수의 데이터 타입을 지정합니다.
  - a. INT16 : 16bit 정수형 변수인 경우.
  - b. WORD : 16bit 부호없는 정수형 변수인 경우
  - c. INT32 : 32bit 정수형 변수인 경우
  - d. DWORD : 32bit 부호없는 정수형 변수인 경우

참 고

※ DWORD로 설정하면 다음 WORD를 사용할 수 없으며, 다음 WORD 설정부분이 회색으로 변합니다.



3. <시작> 버튼을 누르면 Qz명령이 적용되어 바로 모니터링 할 수 있습니다.



4. 아래의 그림과 같이 각 어드레스의 변수 값과 변수 값의 그래프를 실시간 모니터링 할 수 있습니다.

실시간 변수값과 그래프

Display Value  
 Ch1  Ch2  Ch3  Ch4  
 17500 0 15000 0

ROBOCUBETECH MONITOR  
 Memory Monitoring [Qz]  
 7110 7111 7112 7113  AD 12bit  
 INT16 INT16 INT16 INT16 시작

Signal Values  
 1500  
 1000  
 500  
 0  
 -500  
 -1000

Time(msec)  
 59000 60000 61000 62000 63000

Command Terminal  
 ST100000,05000:PE0001:SMOFOP;

Scale [%]  
 100 100 100 100

Offset  
 0 0 0 0

Time Zoom [ms] 2500  
 Axis Zoom  
 Max : 2000  
 Min : -2000

STABILIZE ON

5. 모니터링 정지는 아래의 그림처럼 <정지> 버튼을 누르면 됩니다.



## ■ Qx 명령에 의한 실시간 모니터링 하기

■ 미리 예약되어 있는 어드레스의 변수를 모니터링 하는 방법입니다.

**Qx + Number Mode** 이용하여 1msec 마다 프로그램에서 설정된 4word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data\_logging\_select 변수값을 설정합니다.

■ 관련 명령어 ( Qx, Qxy )

■ CubeMonLT 프로그램의 Qx+Number 명령어는 다음과 같습니다.

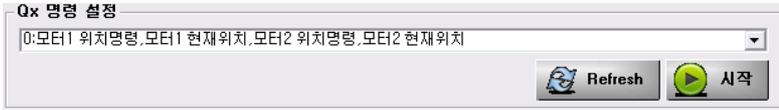
명령어	설명
Qxddd1;	ddd1의 값에 따라 제어기 프로그램에서 미리 정해 놓은 데이터 세트를 지정합니다.

ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0	모터1의 순간위치명령값 (position1_set)을 4로 나눈값	모터1의 현재 위치값 (position1)을 4로 나눈값	모터2의 순간위치명령값 (position2_set)을 4로 나눈값	모터2의 현재 위치값 (position2)을 4로 나눈값
1	모터1의 순간속도명령값 (speed1_cmd_old / SPEED_SCALE_FACTOR)	모터1의 현재 속도값 (speed1_RPM)	모터2의 순간속도명령값 (speed2_cmd_old / SPEED_SCALE_FACTOR)	모터2의 현재 속도값 (speed2_RPM)
2	모터1의 A상 전류명령값 (current1_cmd)	모터1의 A상 권선전류의 A/D 변환값(current1_AD)	모터1의 B상 전류명령값 (current2_cmd)	모터1의 B상 권선전류의 A/D 변환값(current2_AD)
3	모터1의 A상 권선전류의 A/D 변환값(current1_AD)	모터1의 A상 권선전류의 A/D 변환값을 filtering한 값(current1_AD_filtered)	모터1의 B상 권선전류의 A/D 변환값(current2_AD)	모터1의 B상 권선전류의 A/D 변환값을 filtering한 값(current2_AD_filtered)
4	모터1의 현재 총전류의 A/D 변환값(current1_AD)	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (current1_AD_filtered)	모터2의 현재 총전류의 A/D 변환값(current2_AD)	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (current2_AD_filtered)
7	모터2의 A상(RJM_VER6는 모터1의 C상) 전류명령 (current3_cmd)	모터2의 A상(RJM_VER6는 모터1의 C상) 권선전류의 A/D 변환값(current3_AD)	모터2의 B상(RJM_VER6는 모터2의 A상) 전류명령값 (current4_cmd)	모터2의 B상(RJM_VER6는 모터2의 A상) 권선전류의 A/D 변환값(current4_AD)
8	모터2의 B상 전류명령값 (current5_cmd, RJM_VER6 only)	모터2의 B상 권선전류의 A/D 변환값(current5_AD, RJM_VER6 only)	모터2의 C상 전류명령값 (current6_cmd, RJM_VER6 only)	모터2의 C상 권선전류의 A/D 변환값(current6_AD, RJM_VER6 only)
10	모터1의 순간위치명령값 (position1_set, RJM_VER5 only)을 4로 나눈값	모터1의 엔코더에 의한 현재 위치값(position1_ENC, RJM_VER5 only)을 한바퀴당 16384로 환산한 후에 4로 나눈값	모터2의 순간위치명령값 (position2_set, RJM_VER5 only)을 4로 나눈값	모터2의 엔코더에 의한 현재 위치값(position2_ENC, RJM_VER5 only)을 한바퀴당 16384로 환산한 후에 4로 나눈값
14	공급전압 65 가 1V 에 해당함.	모터1의 현재 총전류의 A/D 변환값(total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 변환전류 2/5/10mΩ의 경우 81.9/204.75/409.5 가 변환전류 1A에 해당
15	공급전압 65 가 1V 에 해당함.	모터2의 현재 총전류의 A/D 변환값(total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 변환전류 2/5/10mΩ의 경우 81.9/204.75/409.5 가 변환전류 1A에 해당

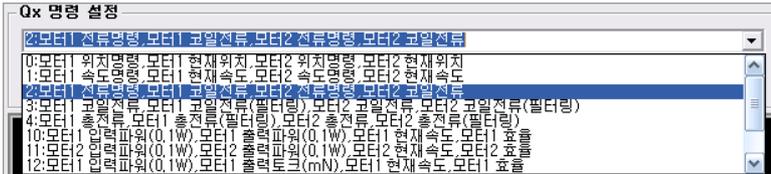
\* 자세한 내용은 명령어 리스트 파일을 참고하십시오

■ Qx 명령어의 사용법은 다음과 같습니다.

1. 아래 그림처럼 Refresh버튼을 누르면 설정메뉴가 활성화,초기화됩니다.



2. 그 후 원하는 명령의 번호를 선택합니다. 번호는 위의 명령어 리스트를 참고 하십시오.



3. 적용이 되었으면 <시작> 버튼을 눌러 모니터링을 시작합니다.

Qx명령에 의한 실시간 변수 값과 그래프 모니터링

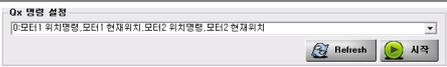
Ch1	Ch2	Ch3	Ch4
0.56	0	241	0

4. 모니터링 정지는 아래의 그림처럼 <정지> 버튼을 누르시면 됩니다.



■ 실시간 모니터링 도중에 명령어의 전송

- Qz, Qx의 모니터링 도중에도 제어기에 명령어 전송이 가능합니다.
- 사용 예

	명령어	설명
1		Qx '1' 속도모드로 선택합니다.
2		시작 버튼을 눌러 모니터링을 시작합니다.
3	Sa100,100,50,50;	100ms당 모터1의 가속도 100rpm 감속도 100rpm 모터2의 가속도 50rpm 감속도 50rpm의 기울기를 갖는 가/감속률을 설정합니다.
4	PE0001;	Servo Enable [ID=1로 가정]
5	SM0909;	STEP모터 전용 속도제어 모드로 설정합니다.
6	SV0700,0800;	모터1의 속도를 700rpm, 모터2의 속도를 800rpm으로 설정합니다.
7	SV1000,1500;	모터1의 속도를 1000rpm, 모터2의 속도를 1500rpm으로 설정합니다.

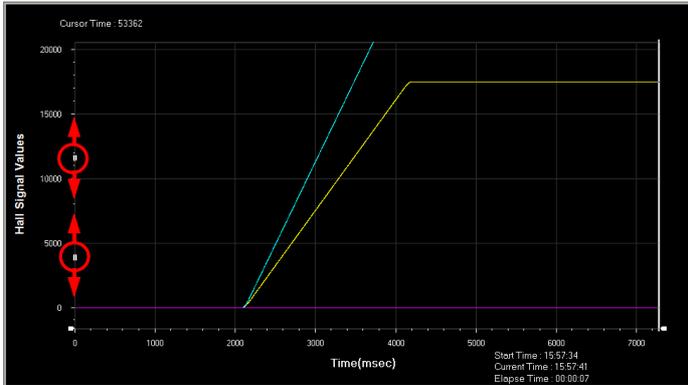
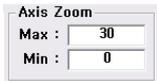


6번 적용시점

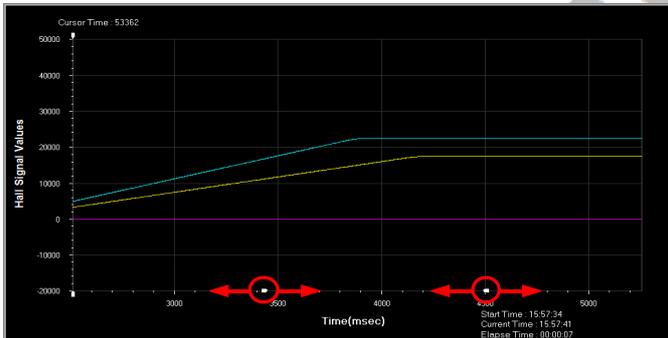
7번 적용시점

■ 그래프의 컬러, Scale, 확대 축소 기능

- 그래프상 Y축 값을 확대/축소 하시려면 우측 상단의 Axis Zoom의 값을 바꾸거나 그래프의 Zoom Pointer를 상하로 Drag하여 조절하면 됩니다.



- 그래프상 Time값을 확대/축소 하시려면 우측 상단의 Time Zoom의 값을 바꾸거나 그래프의 Zoom Pointer를 좌우로 Drag하여 조절하면 됩니다.

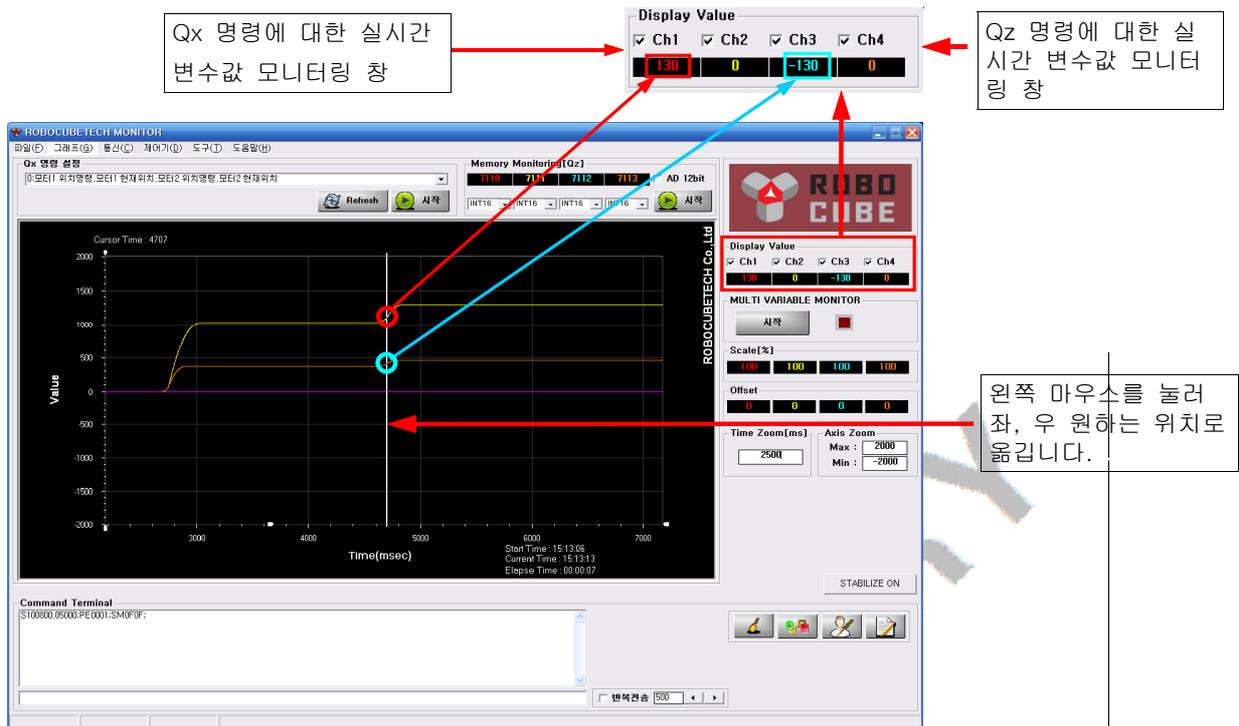


- 그래프상 Time축 또는 Y축을 Scroll하려면 Zoom Pointer의 사이에 마우스를 위치시켜 좌우 또는 상하로 움직이면 됩니다.

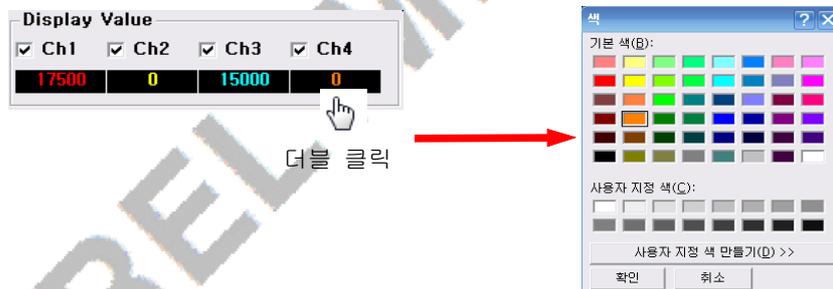


마우스를 Zoom Pointer 사이에 위치시킵니다.

- 그래프상 Time에 대한 변수 값을 알고 싶으면 그래프 표시창에서 왼쪽 마우스를 눌러 좌, 우 원하는 위치로 이동하면 됩니다.



- 그래프상에 표시되는 변수의 색을 변경하려면 해당 변수값 표시 창을 더블클릭하여 변경하십시오.



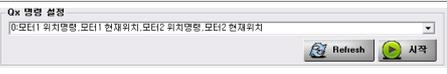
- 스케일 변수를 조절하여 그래프에 반영할 수 있습니다.



## ■ 사인파로 모터 구동 및 모니터링 하기

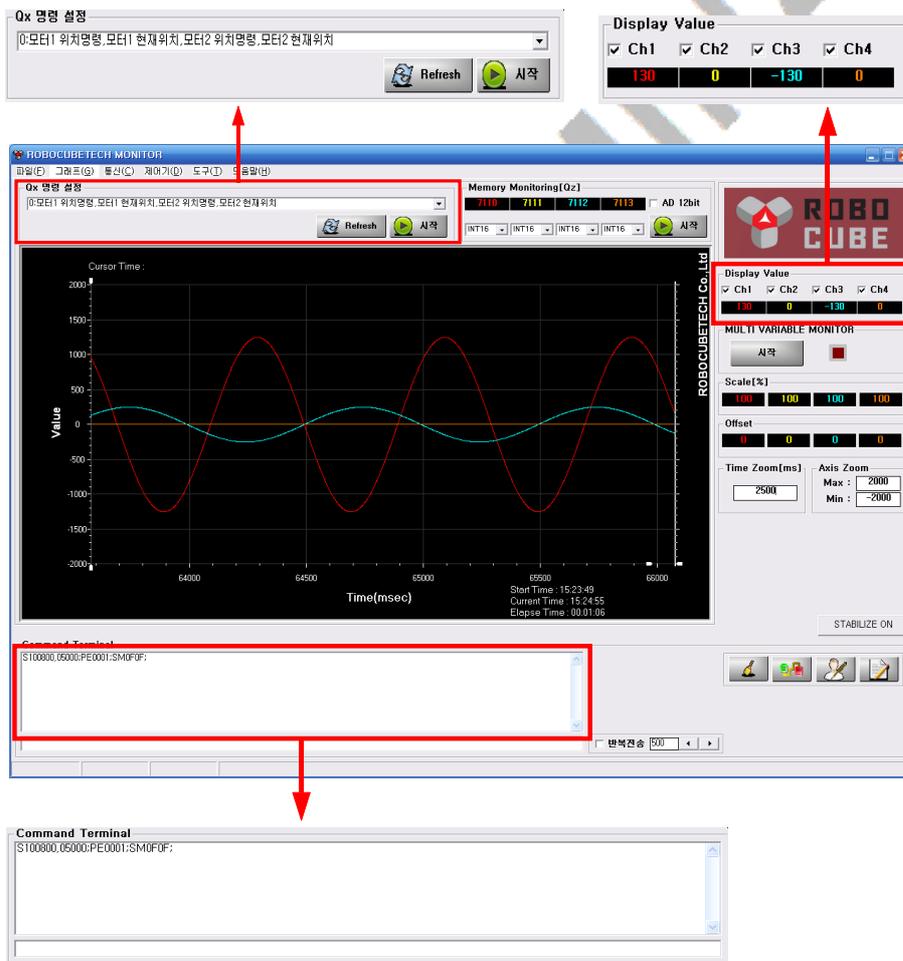
- 작동개요  
 설정된 주파수와 설정된 진폭으로 모터의 변위가 변하도록 하는 기능입니다.  
 서보의 성능을 판단하는데 유용한 기능입니다.
- 관련 명령어 ( SM / S1 / S2 )

■ 아래의 과정은 1번 모터에 대하여 주기 800ms, 진폭 5000펄스로 사인파 변위를 제어하는 예제입니다.

	명령어	설명
1	PE0001;	Servo Enable [ID=1로 가정]
2	SM0F0F;	STEP모터 전용 제어 모드로 설정합니다.
3	S1800,5000;	모터1에 대하여 주기 800ms이며 진폭 5000펄스의 사인파 변위를 정의합니다.
4		Qx의 '0'번을 적용하여 위치 값을 모니터링하도록 설정합니다.
5.		시작 버튼을 눌러 모니터링을 시작합니다.

**참고**  
 ※ S2 명령어는 모터 2에 대한 사인파 변위를 정의합니다.

■ 아래의 그림은 Qx0:의 명령을 이용하여 위치 명령 값과 실시간 위치 값을 모니터링한 결과입니다.



# 6. 모터 구동 전 파라미터 설정

## 6.1 SX 명령으로 기본 환경 설정하기

### ■ 개요

- SX 명령은 통신모드(RS485/CAN/통신테스트) 및 모터의 작동환경(포지션 리미트/계인처리/외부명령모드), 독립적인 시퀀스 제어(작동여부/스텝동작)등의 모드를 설정합니다.

### ■ 명령어 포맷

- SXHLHLHLHL; 아래 2진수의 Status Parameter 값을 16진수로 변환해서 명령어 입력합니다.

### ■ Status Parameter(TMS320F2811 Version)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SCF	SCS	EIM	SLM	HSR	PLF	IRF	DCM	LS		CTM		MCM	MTP	LIM	CM
0 : CM	Communication Mode 0 : CAN 1 : RS485														
1 : LIM	Linear Interpolation Mode 0 : 직선보간 기능 실시 안함 1 : 실시함														
2 : MTP	MotorCoil Thermal Protect 0 : 모터권선의 온도보호 기능의 설정 안함, 1 : 실시함														
3 : MCM	Motor Condition Monitor 0 : 모터의 입력/출력 power, 출력토크, 효율, 모터권선/하우징의 온도계산의 설정 안함 1 : 실시함														
4~6 : CTM	Communication Test Mode 010 : 통신 테스트 모드 <sup>[주2]</sup>														
7 : LS	Limit-Switch Selection 0 : LSA, LSC를 리미트센서로 사용함. 1 : LSA, LSB를 리미트센서로 사용함.														
8 : DCM	Dummy Control Mode 0 : 개별 제어 1 : 2채널 제어기의 경우 두 번째 채널의 제어를 첫 번째 채널의 명령으로 사용함														
9 : IRF	Integral Gain Reset Flag 0 : 적분게인을 유지함 1 : 속도명령이 0 일 때, 적분게인을 0 으로 리셋함(digital hall IC만 사용하는 경우 필수)														
10 : PLF	Position Limit Flag 0 : SL 명령으로 설정한 가상 포지션 리미트 적용함 1 : 포지션 리미트 적용 안함														
11 : HSR	HALL SENSOR Refresh 0 : 홀센서 재계산 안함 1 : 아날로그 홀센서 모터의 경우 SM0606모드에서 속도가 0이되는 순간 홀센서 파라미터를 재계산하여 반영함														
12 : SLM	Speed-Control Limit-Switch Mode 0 : 속도제어시 리미트센서를 사용 안함. 1 : 속도제어시에 리미트센서를 만나면 정지하고, 반대방향으로만 속도값 명령설정 가능														
13 : EIM	External Input Control Mode 0 : LSxC 포트를 입력(Digital/Analog)으로만 사용 1 : 외부 전압 입력(LSxC 포트)에 따라 속도제어 <sup>[주1]</sup> 실행함														

CUBE - ST Series

14 : SCS	<b>Sequence Control Step Mode</b> 0 : 연속으로 Sequence 제어를 실행함 1 : Step by Step으로 Sequence 제어를 실행함
15 : SCF	<b>Sequence Control Flag</b> 0 : Sequence 제어를 실행안함 1 : Sequence 제어를 실행함

[주1] 엔코더가 있는 스텝모터의 경우는 엔코더 입력을 통해서도 위치 제어를 실시합니다.

[주2] 상위제어기로부터 받은 데이터를 2msec 후에 리턴하며, 수신에러 발생시에는 수신에러 상태값 (SCIRXST)을 리턴합니다. 상태값에 대한 자세한 정보는 TMS320F2811 datasheet를 참조하십시오.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DFM	N/A	N/A	BCM	CDM	CDM	SCC	VOP	N/A	N/A	CGV	SLC2	SLC1	SBM	ESC2	ESC1

16 : ESC1	<b>Encoder Signal Change 1</b> 0 : 모터1 엔코더의 A/B상 신호를 서로 안바꿈. 1 : 모터1 엔코더의 A/B상 신호를 서로 바꿈.	2009/08/16일자 이후 펌웨어	어부터 적용
17 : ESC2	<b>Encoder Signal Change 2</b> 0 : 모터2 엔코더의 A/B상 신호를 서로 안바꿈. 1 : 모터2 엔코더의 A/B상 신호를 서로 바꿈.	2009/08/16일자 이후 펌웨어	어부터 적용
18 : SBM	<b>Stepping motor Booster Mode</b> 0 : Stepping 모터의 승압회로를 사용안함 1 : Stepping 모터의 승압회로를 사용함	2010/02/05일자 이후 펌웨어	어부터 적용
19 : SLC1	<b>Stepping motor Lead angle Controll Mode 1</b> 0 : Stepping 모터1의 Lead Angle 제어를 실시안함 1 : Stepping 모터1의 Lead Angle 제어를 실시함	2010/02/05일자 이후 펌웨어	어부터 적용
20 : SLC2	<b>Stepping motor Lead angle Controll Mode 2</b> 0 : Stepping 모터2의 Lead Angle 제어를 실시안함 1 : Stepping 모터2의 Lead Angle 제어를 실시함	2010/02/05일자 이후 펌웨어	어부터 적용
21 : CGV	<b>Current Gain Variation</b> 0 : Stepping 모터의 전류제어게인을 속도에 따라 가변하는 기능 실시함 1 : 실시 안함	2010/02/19일자 이후 펌웨어	어부터 적용
24 : VOP	<b>Velocity Over Process</b> 0 : 위치제어시 설정속도가 과도할 때 발생 문제처리 실시함 1 : 실시 안함	2010/04/04일자 이후 펌웨어	어부터 적용
25 : SCC	<b>Synchronization and CAN Command</b> 0 : 메인 루프 동기화와 위치/전류 명령값의 수신을 CAN으로 실시 안함 1 : 실시함	2010/04/04일자 이후 펌웨어	어부터 적용
26 : CDM	<b>Counter Electromotive Force Decrease</b> 0 : 속도제어에서 감속시 역기전력을 감소시키는 기능 사용함 1 : 사용 안함	2010/05/24일자 이후 펌웨어	어부터 적용
27 : CDM	<b>Counter Electromotive Force Decrease</b> 0 : 위치제어에서 감속시 역기전력을 감소시키는 기능 사용 안함 1 : 사용함	2010/05/24일자 이후 펌웨어	어부터 적용
28 : BCM	<b>Block Commutation Mode</b> 0 : BLDC모터에서 Block Commutation 방식을 사용 안함 1 : 사용함	2010/05/24일자 이후 펌웨어	어부터 적용
31 : DFM	<b>Demo Function Mode</b> 0 : 데모작동을 안함. 1 : 데모작동을 실시함.	2010/02/05일자 이후 펌웨어	어부터 적용

CUBE - ST Series

## 6.2 ST 명령으로 모터 타입 설정하기

### ■ 개요

- ST 명령은 제어기에 연결될 모터의 타입을 설정합니다.
- STEP모터 제어기의 경우 STFFFF 로 초기 설정이 되어있고 이 설정에서 제어기 주소, 엔코더 펄스 수 등을 설정해서 그대로 사용하면 됩니다.

### ■ 관련 명령어

- SEA55A / SGA55A / EDA55A / EsA55A

### ■ 명령어 포맷

ST	HL	HL	;	
		L	—	1번째 채널의 모터 타입
		L	—	2번째 채널의 모터 타입
L	—	—	—	명령어

### ■ STEP 모터 타입 설정하기

- 아래와 같은 방법으로 제어 파라미터를 설정할 수 있습니다. 아래는 스텝각 18°인 경우의 설정에 대한 예제입니다.
- **모터에 전원을 투입하고 모터를 작동시키지 않은 상태에서 설정을 개시하여야합니다.**

	명령어	설명
1	STFFFF;	ST모터 제어기 초기 설정값.
2	EDA55A;	모든 파라미터를 factory default 값으로 바꿉니다.
3	SEA55A,0FA0,0FA0,0005,0005;	90°/18°(스텝각) 5를 헥사값 0005로 설정합니다.
4	EDA55A;	이 과정에서 바뀐 파라미터에 따른 초기화를 실시합니다.
5	SAFE01;	사용하려는 주소로 재설정합니다.
6	STFFFFE: —	제어 파라미터가 변경되지 않도록 모터 타입을 FFFF로 설정합니다.
7	EsA55A;	설정된 파라미터를 저장합니다.

### 6.3 SA 명령으로 제어기 주소 설정하기

#### ■ 개요

- SA 명령은 제어기에 Address를 설정하는 명령으로 Daisy-Chain 방식으로 연결된 각 제어기의 ID를 부여하게 됩니다.

#### ■ 관련 명령어

- EsA55A

#### ■ 명령어 포맷

SA	HL	HL	;	
		L	—	제어기의 주소(ID)이며 16진수 값으로 표시되어야 합니다.
	L	—	—	제어기 주소의 보수(INVERSE Hex)입니다.
L	—	—	—	명령어

#### ■ 명령어 사용 예

- 아래는 제어기의 주소를 3번으로 설정한 후 저장하는 과정의 예입니다.

	모델 구분	명령어	설명
1	ALL	SAFC03;	제어기의 ID를 3번으로 설정합니다.
2	ALL	EsA55A;	파라미터를 저장합니다.

### 6.4 SEA55A 명령으로 엔코더 펄스 설정하기

#### ■ 개요

- SEA55A 명령은 모터가 1회전할 때 엔코더의 펄스수와 스텝각을 설정하는 명령입니다. 엔코더의 펄스수는 QEP(4체배 펄스)로 스텝각설정은 90°/(스텝각)을 헥사값으로 설정합니다.
- SEA55A 명령은 EDA55A; 명령을 실행한 이후에만 적용이 가능합니다.

#### ■ 관련 명령어

- ST / EDA55A / EsA55A

#### ■ 명령어 포맷

SEA55A	,	HLHL	,	HLHL	,	HLHL	,	HLHL	;	
				L	—	—	—	—	—	모터 2의 90°/STEP각
			L	—	—	—	—	—	—	모터 1의 90°/STEP각
		L	—	—	—	—	—	—	—	모터 2의 1회전당 4체배 엔코더 펄스수
	L	—	—	—	—	—	—	—	—	모터 1의 1회전당 4체배 엔코더 펄스수
L	—	—	—	—	—	—	—	—	—	명령어

## 6.5 SGA55A 명령으로 제어 변수 설정하기

### ■ 개요

- SGA55A 명령은 감속비, 모터의 최대 회전속도를 설정하는 명령어입니다. 최대 회전속도의 경우 DC 모터와 BLDC 모터에 대한 PWM 주기 파라미터의 Scale이 다르므로 주의하시기 바랍니다.
- SGA55A 명령은 EDA55A; 명령을 실행한 이후에만 적용이 가능합니다.

### ■ 관련 명령어

- ST / EDA55A / EsA55A

### ■ 명령어 포맷

SGA55A , d1 , d2 , d3 , d4 ;					
				L	모터 2의 최대속도(rpm) 2406 : 1000 ~ 32000 2811 : 1000 ~ 60000
			L	—	모터 1의 최대속도(rpm) 2406 : 1000 ~ 32000 2811 : 1000 ~ 60000
		L	—	—	모터2의 감속비[1 ~ 2000], 100일 때 100 : 1의 감속비를 의미
	L	—	—	—	모터1의 감속비[1 ~ 2000], 100일 때 100 : 1의 감속비를 의미
L	—	—	—	—	명령어

### ■ 명령어 사용 예

- SGA55A 명령의 사용에는 6.2절의 "STEP 모터 타입 설정하기"를 참고하십시오.
- SG에서 감속비는 정확히 맞추지 않아도 되며, 잘 모르면 100으로 설정하면 됩니다.

## 6.6 SI 명령으로 전류제한값 설정하기

### ■ 개요

- SI 명령은 구동회로와 기구를 보호하기 위한 Logic에서 사용하는 전류리미트값과 time threshold를 설정하는 명령어입니다.
- 전원측에서 제어기로 입력되는 총전류의 값을 제한할 수 있습니다.

### ■ 관련 명령어

- Sw / EsA55A

### ■ 명령어 포맷

SI	d1	d2	d3	d4	d5	d6	d7	d8	
								└	모터2의 전류초과시에 위치명령을 hold하기까지의 시간(ms)
								└	—
							└	—	모터2의 전류초과를 허용하는 시간 대비 끄는 비율(1~10)
					└			—	—
				└				—	—
			└					—	—
		└						—	—
	└							—	—
└								—	—
└	└							—	—
└	└	└						—	—
└	└	└	└					—	—
└	└	└	└	└				—	—
└	└	└	└	└	└			—	—
└	└	└	└	└	└	└		—	—
└	└	└	└	└	└	└	└	—	명령어

### ■ 참고 사항

- 당사 제어기는 모델에 따라 크기가 다른 전류감지 저항을 사용하기 때문에 SI, Sw 명령을 사용할 때에는 1A에 해당하는 파라미터 값을 계산하여 사용해야 합니다. 명령어 리스트 부분을 참고하여 착오 없으시길 바랍니다.
- 모델별 1A에 해당하는 값  
1A = 409.5 bit/A (ST2402-DII/E)

## 6.7 Sw 명령으로 전류제한값 설정하기

### ■ 개요

- Sw 명령은 제어시 사용하는 전류의 리미트를 설정하는 명령어입니다. 전류를 사용하여 전류나 위치, 속도제어를 하는 경우에 사용하는 전류의 최대치를 설정합니다.
- 제어기측에서 모터로 흐르는 전류값을 제한할 수 있습니다.
- 리미트값의 스케일은 SI명령 참고부분을 참조하십시오.

### ■ 관련 명령어

- SI / EsA55A

### ■ 명령어 포맷

Sw	d1	,	d2	;	
			L		모터 2 제어전류리미트 (0~1638)
			L		모터 1 제어전류리미트 (0~1638)
L			-		명령어

### ■ 명령어 사용 예

	명령어	설명
1	Sw800,800;	모터 권선에 흐르는 전류리미트값을 약 10A로 설정한다
2	EsA55A;	현재설정값을 저장한다

## 6.8 SQ/Sq/SR/Sr 명령으로 제어파라미터 설정하기

### ■ 개요

- SQ/SR 명령은 각각 모터1/모터2의 제어파라미터를 설정할 수 있습니다.
- 위치/속도 제어시의 P,I,D 게인을 설정할 수 있습니다.
- Sq/Sr 명령일때 게인값들은 50000/inverse gain(SQ/SR값)을 의미합니다.

### ■ 관련 명령어

- EsA55A

### ■ 명령어 포맷

SQ	HLHLHLHL , HLHLHLHL , HLHLHLHL , HLHL , HLHLHLHL , HLHL , HLHL , HLHLHLHL , HLHL , HLHL , HLHLHLHL ;
1st	- long word MOTOR1_ERROR_SIZE_LIMIT
2nd	- long word MOTOR1_INTEGRAL_START_ERROR_SIZE
3rd	- long word MOTOR1_INTEGRAL_LIMIT
4th	- word 위치제어시 MOTOR1_INVERSE_OF_P_GAIN
5th	- long word 위치제어시 MOTOR1_INVERSE_OF_I_GAIN
6th	- word 위치제어시 MOTOR1_INVERSE_OF_D_GAIN
7th	- word 속도제어시 MOTOR1_INVERSE_OF_P_GAIN
8th	- long word 속도제어시 MOTOR1_INVERSE_OF_I_GAIN
9th	- word 속도제어시 MOTOR1_INVERSE_OF_D_GAIN
10th	- word 전류제어시 MOTOR1_INVERSE_OF_P_GAIN
11th	- long word 전류제어시 MOTOR1_INVERSE_OF_I_GAIN

CUBE - ST Series

### ■ 참고 사항

- ST2402-DIIE(엔코더 사용하는 버전) 사용시 위치제어(02번)에서는 SQ/Sq 또는 SR/Sr의 4/5/6번째 값을 변경해 주면 됩니다. (10/11번째 값은 제어시 사용되는 전류 게인값입니다)
- ST2402-DII(엔코더 미사용) 사용시 위치(08)/속도(09)제어에서는 SQ/SR의 10/11번째 값인 전류게인값만 변경하여 주면 됩니다.
- 각각의 설정값은 INVERSE 값으로, 설정값을 1/2으로 줄이면 해당 게인이 2배로 커짐을 뜻합니다.
- 게인 설정값은 최소한 1보다 커야하고, SQ?; , SR?; 등을 사용하여 현재값을 읽어내어 조금씩 바꾸는 형식으로 실시하시기 바랍니다.
- 무부하 상태에서 게인을 너무 높이면 모터진동이 심할 수 있으니 주의하시기 바랍니다.

## 명령어 사용 예

	명령어	설명
1	SQ?;	현재 게인값 확인
2	SQ00005000,00002000,003E8000,0100,00000200,0080,0080,00000200,0080,0080,00000004;	리턴되어 온 현재 게인값
3	SQ00005000,00002000,003E8000,0100,00000200,0080,0040,00000200,0080,0080,00000004;	속도제어시의 P-GAIN을 2배 만큼 증가시킨다
4	SQ00005000,00002000,003E8000,0100,00000200,0200,0080,00000200,0080,0080,00000004;	위치제어시의 I-GAIN을 1/4배만큼 감소시킨다
5	EsA55A;	현재 게인값을 저장한다

- 위치제어시 PID 제어 관계식 (제어주기는 1ms)

$$current\_cmd = \frac{400}{KP^{-1}} e_k + \frac{1}{2 * KI^{-1}} \sum_{i=0}^{i=k} e_i + \frac{3200}{KD^{-1}} (e_k - e_{k-4})$$

current\_cmd는 -current\_control\_limit ~ +current\_control\_limit 범위의 값으로 제한되며, 기본값은 -1229 ~ +1229 인데, 2mΩ의 전류감지저항을 쓰는 경우 -15A ~ +15A에 해당함. current\_control\_limit의 크기는 Sw 명령어로 설정하고 저장할 수 있음. KP<sup>-1</sup>/KI<sup>-1</sup>/KD<sup>-1</sup>는 SQ/SR 명령어로 설정하고 저장할 수 있음.

- ①  $e_k = position\_set - position$
- ②  $KP^{-1} = (PPR * 50) / PWM$
- ③  $KI^{-1} = (PPR * 50) / PWM$
- ④  $KD^{-1} = (PPR * 4) * (V_{max} / 60) / PWM$

- ① 기본단위는 엔코더의 펄스를 4채배한 값임
- ② 이 기본값을 사용하면 위치오차가 모터 한바퀴일 때, 59A의 전류를 적용하는 것에 해당함.
- ③ 이 기본값을 사용하면 위치오차가 200ms동안 모터 한바퀴일 때, 15A의 전류를 적용하는 것에 해당함.
- ④ 이 기본값을 사용하면 최대 RPM( $V_{max}$ )으로 회전할 때, 23A의 전류를 적용하는 것에 해당함.

PWM=600, PPR=모터 1회전시의 4채배된 엔코더 펄스수,  $V_{max}$ =무부하최대회전속도

## 6.9 SH 명령으로 구동전류 진폭 설정하기

### 개 요

- SH 명령으로 구동전류의 진폭과 전류 OFFSET을 설정하는 기능입니다.

### 명령어 포맷

명령어	파라미터	설명	초기값
SH	i1,i2,i3,i4 i5,i6,i7,i8;	1st : 모터1의 구동전류 진폭 (1 ~ 1000)	200
		2nd : 모터2의 구동전류 진폭 (1 ~ 1000)	200
		3rd : 코일1의 구동전류 offset (-200 ~ 200)	0
		4th : 코일2의 구동전류 offset (-200 ~ 200)	0
		5th : 코일3의 구동전류 offset (-200 ~ 200)	0
		6th : 코일4의 구동전류 offset (-200 ~ 200)	0
		7th : 코일5의 구동전류 offset (-200 ~ 200)	0
		8th : 코일6의 구동전류 offset (-200 ~ 200)	0
SH	?	현재 구동전류 진폭값을 확인 합니다.	

### 참고 사항

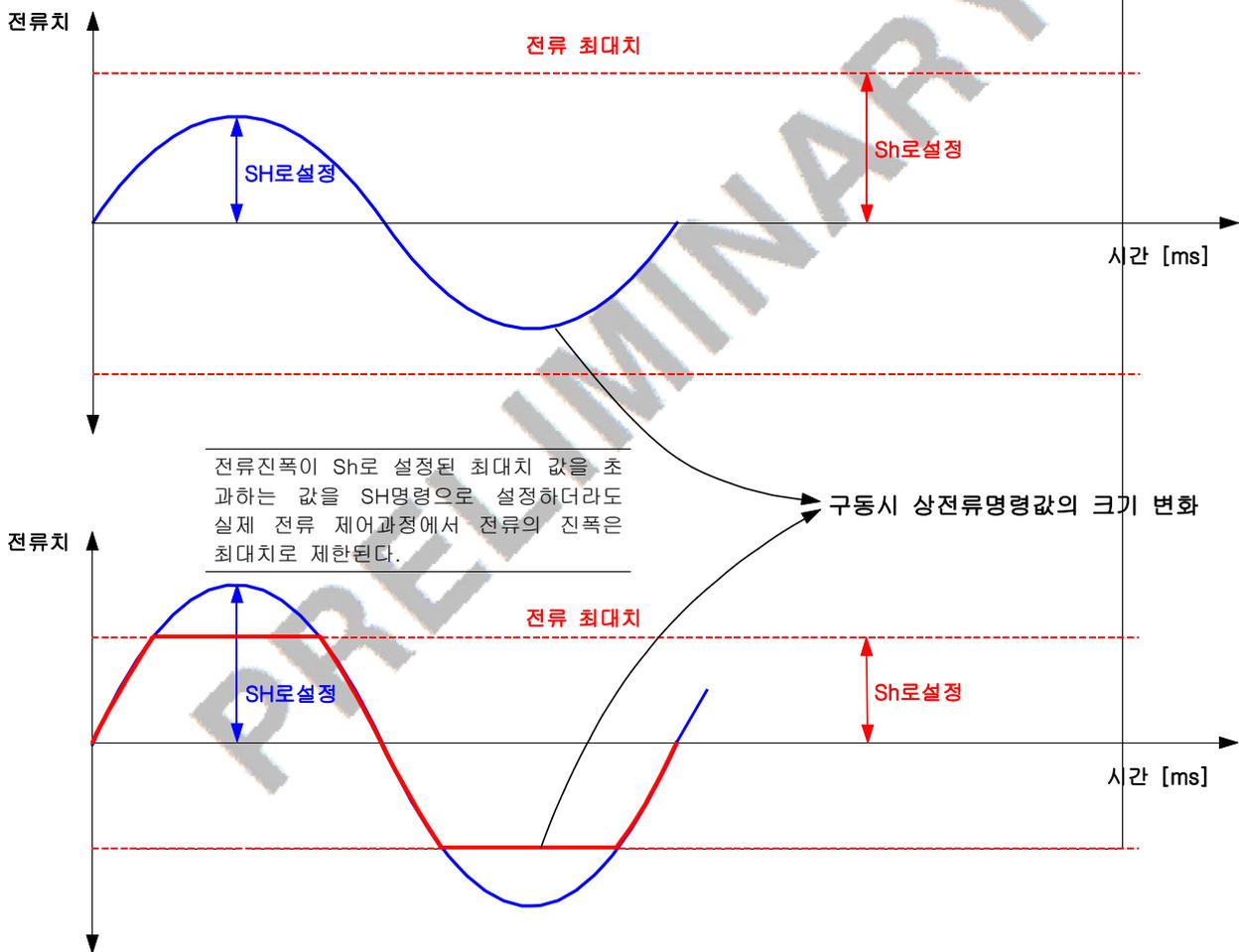
- SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (RJM\_VER5/6 STEP형만 해당됨)  
SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; (RJM\_VER6 3상STEP형만 해당됨)
- F2811/F28334를 사용한 STEP MOTOR 제어기(RJM\_VER5)의 경우는 10mΩ으로 409.5 bit/A의 스케일을 갖습니다.  
RJM\_VER5 STEP형을 2상 스테핑 모터의 구동기로 사용하는 경우는 coil1/2 가 한개의 모터가 되고 coil3/4 가 또 한개의 모터가 되어 총 2개의 스테핑모터를 구동할 수 있습니다.  
RJM\_VER5 STEP형을 3상 스테핑 모터의 구동기로 사용하는 경우는 coil1/2/3 가 해당코일이 되며, coil4는 단독 코일로서 구동전류 offset에 해당하는 전류가 흐르도록 제어됩니다.  
RJM\_VER6 3상STEP형인 경우는 첫번째 모터는 coil1/2/3 가 해당코일이 되며 두번째 모터는 coil4/5/6 이 해당코일이 됩니다.

## 6.10 Sh 명령으로 전류명령값의 최대치 설정하기

### 개요

- Sh 명령으로 전류명령값의 최대치와 정지시의 전류치 비율을 설정할 수 있다.

명령어	파라미터	설명	초기값
Sh	d1,d2,d3,d4;	전류 명령값의 최대치와 모터가 정지하고 있을 때의 전류치 비율을 설정한다.	400,400 ,8,8
Sh	?	현재 전류최대치와 전류치 비율을 확인.	



■ 명령어 포맷

Sh	d1	d2	d3	d4	
				└	전류2의 감소율 [0 - 128] (8)
			└	—	전류1의 감소율 [0 - 128] (8)
		└	—	—	전류2의 최대치 [0 - 4000] (400)
	└	—	—	—	전류1의 최대치 [0 - 4000] (400)
└	—	—	—	—	명령어

■ 참고 사항

- 전류진폭이 전류1/2 최대치 값을 초과하는 전류진폭을 SH명령어로 설정하더라도 실제 전류제어과정에서 전류의 진폭은 최대치로 제한됩니다. 따라서 진폭이 제한되는 경우에는 전류의 명령값이 사인파가 아니고 saturation 된 사인파가 됩니다. 그러므로 리밋에 걸리지 않기 위해서는 SH값을 Sh값보다 작거나 같게 설정해야 한다.
- 값이 128이면 정지시 전류는 구동시전류의 100%, 64이면 구동시전류의 50%, 8이면 6.25%이다.

PRELIMINARY

## 7. 모터 구동

### 7.1 제어모드 설정

#### ■ 제어모드 설정 설명

- 제어기의 제어 방식을 위치제어 / 속도제어 / 전류제어 설정

명령어	파라미터	설명	초기값
SM	HLHL	하위 HL [모터1] 상위 HL [모터2] 08 : STEP모터 위치제어 모드 09 : STEP모터 속도제어 모드 0F : 사인파 위치제어 00 : 제어모드 OFF	0000
SM	?	현재 모드 상태를 확인	

#### ■ 명령어 포맷

SM	HL	HL	;	
		L	—	모터1 모드
	L	—	—	모터2 모드
L	—	—	—	명령어

### 7.2 서보 Enable / Disable

#### ■ 서보 동작 개시

- 제어기의 모터 출력을 ON / OFF 시킴

명령어	파라미터	설명	초기값
PE	HLHL	제어기의 모터 출력을 ON 시킴 제어기의 ID로 설정값 입력 [ex]ID가 1인 제어기 PE0001;	0000
PE	?	현재 서보 상태를 확인 (모터 출력이 ON되어 있을 때는 board address 임)	
PD	HLHL	제어기의 모터 출력을 OFF 시킴	

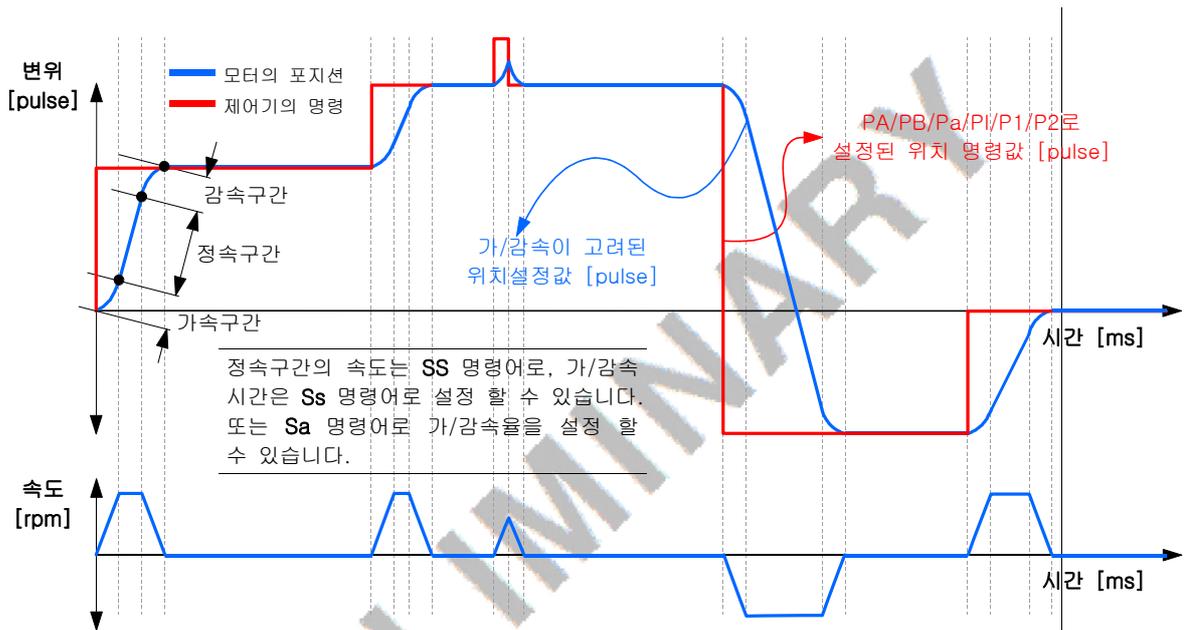
#### ■ 명령어 포맷

PE	HLHL	;	
	L	—	제어기의 주소(ID)이며 16진수 값으로 표시되어야 합니다.
L	—	—	명령어

### 7.3 PA/PB/Pa/PI/P1/P2 명령을 이용한 위치제어

#### ■ 작동 개요

- PA/PB/Pa/PI/P1/P2 명령은 아래 그림과 같이 SS 명령으로 설정된 속도와 Ss 명령으로 설정된 가/감속시간 또는 Sa 명령으로 설정된 가/감속률에 따라 Point to Point로 이동하는 명령입니다.
- PA/PB/Pa/PI/P1/P2 명령은 모터 회전 여부에 관계 없이 어느 때라도 임의의 순간에 원하는 가속도/감속도/정속도의 크기 및 목표위치 값을 임의로 설정할 수 있는 막강한 기능을 제공합니다.
- 만약 제어기가 속도나 전류제어 모드로 설정이 되어있다면, PP/PQ/Pp/Pq/p 명령으로 전송되는 값은 위치 명령 값이 아니고 속도, 전류 명령 값으로 제어기가 인식합니다.



#### ■ 관련 명령어

- SM / SS / Ss / Sa / PE / QP

#### ■ 명령어 사용 예

- 아래 표와 같이 순서대로 명령을 입력하면 모터 1과 모터 2를 각각 50000/100000만큼 위치제어를 실시하며 이때 정속 구간의 속도는 1000/1500rpm이고, 가/감속은 500/400ms 동안 수행 됩니다.

	명령어	설명
1	SS1000,1500; [주1]	정속 구간의 이동속도를 모터1 1000rpm, 모터2 1500rpm으로 설정합니다.
2	Ss500,400;	가/감속 구간의 시간을 모터1 500ms, 모터2 400ms로 설정합니다.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808; [주1]	위치제어 모드로 설정합니다.
5	PA5050000,5100000;	위치 명령을 모터1 5050000, 모터2 5100000을 전송하고 즉시 운동이 개시됩니다. 5000000 이 원점입니다. (위치명령의 16384값이 모터의 1회전 값입니다. 명령을 5016384로 주었을 경우 모터가 1바퀴 돕니다.)
6	QP;	현재 위치를 확인하는 명령어입니다.

CUBE - ST Series

[주1] SM 명령의 경우 만약 SM0008; 이었다면 모터1이 08번 STEP제어모드로 설정되고, 모터2가 00번으로 설정됩니다. SS 명령의 경우 만약 SS1000,2000; 이었다면 모터1의 정속구간 속도가 1000rpm이며 모터2의 정속구간 속도가 2000rpm입니다. 이처럼 2채널용 제어기의 경우 각 명령어마다 파라미터의 순서가 다르므로 명령어 설명을 꼭 확인 후 사용하십시오.

## ■ 참고 사항

- 용도에 따라 "QP;" "QV;" "QC;" 명령으로 각각 위치와 속도, 전류를 확인할 수 있습니다.
- 경우에 따라 "QEA55A;" "QEA5A5;" "QE5A5A;"<sup>[주3]</sup> 명령 등으로 Encoder 펄스 파라미터를 초기화할 수 있습니다.
- "SS"와 "Ss" 명령으로 설정된 파라미터는 "EsA55A;" 명령으로 EEPROM에 저장할 수 있습니다.
- 제어시 가/감속 프로파일은 "SS" + "Ss" 또는 "Sa"명령으로 설정할 수 있습니다. PA/PB/Pa/PI/P1/P2 명령시에 사용되는 가/감속 프로파일은 가장 마지막에 설정된 가/감속 설정 명령에 따릅니다.
- 제어모드를 종료하려면 SMHLHL; 명령어로 00의 제어 모드를 선택하고, PDHLHL; 명령어로 해당 모터제어보드의 출력을 disable 시키면 됩니다.

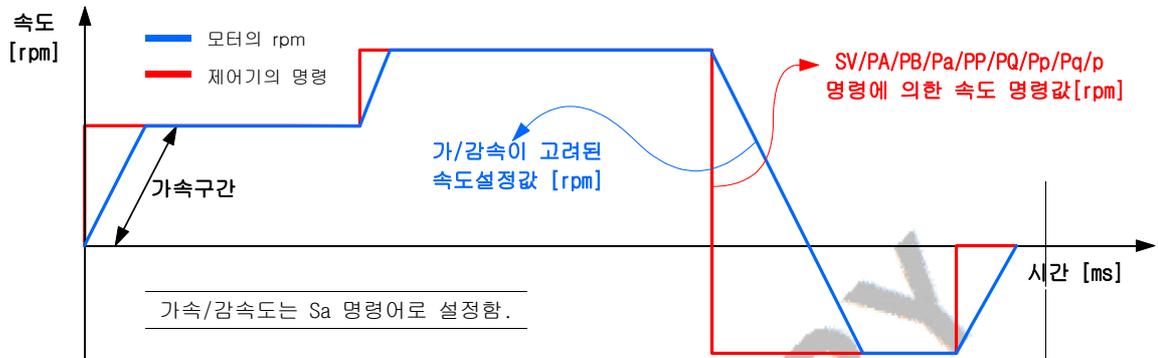
[주3] QEA55A; 모터(1)(2) 모두 RESET  
 QEA5A5; 모터(1)만 RESET  
 QE5A5A; 모터(2)만 RESET

PRELIMINARY

## 7.4 SV 명령을 이용한 속도제어

### ■ 작동 개요

- SV 명령은 Sa 명령으로 설정된 가/감속 프로파일에 따라서 속도를 제어 하는 명령입니다.



### ■ 관련 명령어

- SM / Sa / PE

### ■ 명령어 사용 예

- 아래 표와 같이 순서대로 명령을 입력하면 Sa로 설정된 가/감속률로 SV로 설정한 속도[rpm]로 회전하게 됩니다.

	명령어	설명
1	Sa400,300,200,400;	100ms당 모터1의 가속도 400rpm 감속도 300rpm 모터2의 가속도 200rpm 감속도 400rpm의 기울기를 갖는 가/감속률을 설정합니다. [rpm/100ms] (※이 설정을 1초단위로 환산하면 4000,3000, 2000, 4000 rpm/sec가 됩니다.)
2	PE0001;	Servo Enable [ID=1로 가정]
3	SM0909;	STEP모터 전용 속도제어 모드로 설정합니다.
4	SV0800,-0900;	모터1의 속도를 800rpm, 모터2의 속도를 -900rpm으로 설정하고, 즉시 운동개시 합니다.

## 7.5 PV 명령을 이용한 속도제어

### ■ 작동 개요

- PV 명령은 기본적으로 SV 명령에 의한 속도 제어 방식과 동일합니다.

### ■ 관련 명령어

- SV / SM / Sa / PE

### ■ 명령어 사용 예

- 아래 표와 같이 순서대로 명령을 입력하면 Sa로 설정된 가/감속률로 PV로 설정한 속도[rpm]로 회전하게 됩니다.

	명령어	설명	
1	Sa100,100,50,500;	100ms당 모터1의 가속도 100rpm 감속도 100rpm 모터2의 가속도 50rpm 감속도 500rpm의 기울기를 갖는 가/감속률을 설정합니다. [rpm/100ms] (※이 설정을 1초단위로 환산하면 1000,1000, 500, 5000 rpm/sec가 됩니다.)	
2	PE0001;	Servo Enable [ID=1로 가정]	
3	SM0909;	STEP모터 전용 속도제어 모드로 설정합니다.	
4	PV100,100,334; <sup>[주2]</sup>	모터1,2의 속도를 100rpm으로 설정합니다.	

[주2] checksum은 PV 명령어 뒤의 두 속도값과 속도값 사이의 ',' 만 포함하며 해당 문자의 값을 모두 더한 값이 됩니다. 위 표의 예는 아래와 같습니다.  
[100,100] = 49(1) + 48(0) + 48(0) + 44(,) + 49(1) + 48(0) + 48(0) = 334

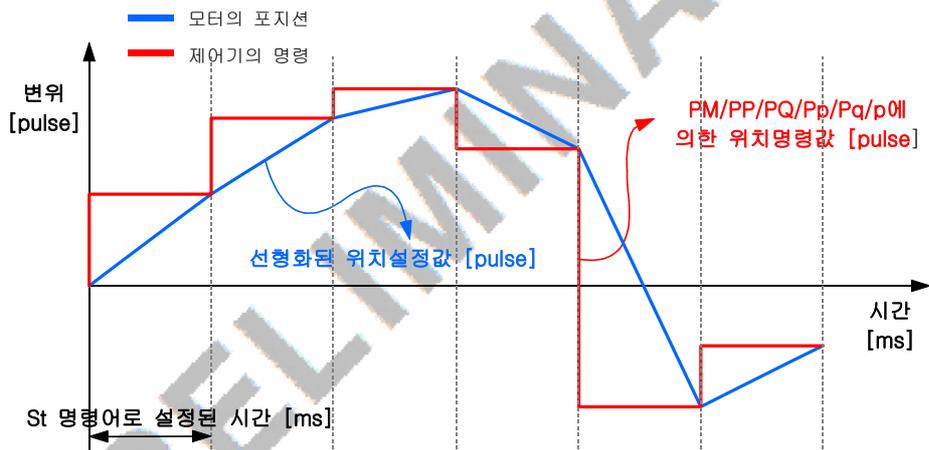
## 7.6 PP/PQ/Pp/Pq/p 명령을 이용한 실시간 위치제어

### 작동 개요

- PP/PQ/Pp/Pq/p 명령을 이용한 위치제어 방식은 최소 2ms ~ 최대 1000ms 범위의 주기로 명령을 전송하는 제어 방식입니다. 주기적으로 수신된 위치명령 값은 모터제어기에서 1ms 단위로 interpolation 하여 위치제어에 사용합니다.
- 호스트로부터 전송되는 명령의 주기는 "St" 명령어를 사용하여 설정할 수 있습니다.
- PP/PQ/Pp/Pq/p<sup>[주1]</sup> 명령의 경우에는 별도의 가속/감속이나 속도제한이 적용되지 않기 때문에 호스트에서 이를 감안한 명령값을 전송해야만 합니다.
- 만약 제어기가 속도나 전류제어 모드로 설정이 되어있다면, PP/PQ/Pp/Pq/p 명령으로 전송되는 값은 위치 명령 값이 아니고 속도, 전류 명령 값으로 제어기가 인식합니다.
- 제어기에서 Pp/Pq/p 명령의 return값은 "Sr"명령어로 지정된 주소의 변수값이 return됩니다.

[주1] p 명령은 CAN 통신 전용 명령어입니다.

PP/PQ의 파라미터는 10진수, Pp/Pq/p의 파라미터는 binary 값입니다.



### 관련 명령어

- SM / St / PE / Sr / PP / PQ / Pp / Pq / p

명령어 사용 예

■ 아래의 1 ~ 4번 과정을 순차적으로 전송 후 5번 과정을 반복 송신합니다.

	명령어	설명
1	St10,10,3; <sup>[주1]</sup>	호스트로부터의 명령어 수신 주기를 10ms로 설정합니다.
2	Sr7108,7109;	Pp/Pq/p 명령 전송시 리턴 받고자 하는 데이터의 주소를 설정합니다. 각 데이터의 주소는 Map Code 편을 참고하십시오.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	PP5001000,5001000; 또는 Pp4C4F284C4F2879;	실시간으로 이동해야할 위치값을 전송하고 즉시 운동을 개시합니다. Pp 명령의 경우에 4C4F28은 24bit binary number를 각각 가리키며 마지막 1byte 79는 앞 파라미터 6byte 합 의 Inverse Check Sum입니다. 따라서 Pp 명령의 경우에 Pp;를 포함한 총 데이터수는 10 byte입니다.
6		St로 설정된 주기마다 원하는 포지션 값을 파라미터로 하여 5번 항목의 명령을 반복합니다.

[주1]	St 10 , 10 , 3 ;		
	L	명령어 미수신 연속 허용횟수입니다. 설정된 횟수 이상 연속으로	통신장애 또는 미수신시 마지막 정상 수신 명령의 위치로 복귀합니다.
	L —	PQ/Pq/p 명령의 Update 주기(ms)	
	L — —	PP/Pp/p 명령의 Update 주기(ms)	
	L — — —	PP/PQ/Pp/Pq/p 명령의 Update 주기와 명령어 미수신의 연속 허용횟수를 설정함	

CUBE - ST Series

## 7.7 PP/PQ/Pp/Pq/p 명령을 이용한 속도제어

### 작동 개요

- PP/PQ/Pp/Pq/p 명령을 이용한 속도제어 방식은 최소 2ms ~ 최대 1000ms 범위의 주기로 명령을 전송하는 제어 방식입니다.
- 호스트로부터 전송되는 명령의 주기는 "St" 명령어를 사용하여 설정할 수 있습니다.

### 관련 명령어

- SM / St / PE / Sr / PP / PQ / Pp / Pq / p

### 명령어 사용 예

- 아래의 1 ~ 4번 과정을 순차적으로 전송 후 5번 과정을 반복 송신합니다.

	명령어	설명
1	St10,10,3;	호스트로부터의 명령어 수신 주기를 10ms로 설정합니다.
2	Sr7108,7109;	Pp/Pq/p 명령 전송시 리턴 받고자 하는 데이터의 주소를 설정합니다. 각 데이터의 주소는 Map Code 편을 참고하십시오.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	PP5001000,5001000; 또는 Pp4C4F284C4F2879;	실시간으로 이동해야할 속도값 <sup>[주1]</sup> 을 전송하며, 즉시 운동을 개시합니다. Pp 명령의 경우에 4C4F28는 24bit binary number를 각각 가리키며 마지막 1byte 79는 앞 파라미터 6byte 합인 Inverse Check Sum입니다. 따라서 Pp 명령의 경우에 Pp;를 포함한 총 데이터수는 10 byte입니다.
6		St로 설정된 주기마다 원하는 포지션 값을 파라미터로 하여 5번 항목의 명령을 반복합니다.

[주1] SM0808: 제어 모드인 경우 PP/PQ/Pp/Pq/p 명령의 단위는 rpm이며, 파라미터의 값이 5000000일 때 속도는 0rpm이고 방향은 아래와 같습니다.

- 속도 명령 > 5000000 : +방향으로 회전
- 속도 명령 < 5000000 : -방향으로 회전

## 7.8 PM 명령을 이용한 위치제어

### 작동 개요

- PM 명령은 SP 명령으로 EEPROM에 저장해 놓은 Position으로 이동하는 명령으로 반복적인 모션제어에 용이한 명령어입니다.
- EEPROM에 저장할 수 있는 포인트의 개수는 최대 192개입니다. 한개의 포인트는 2개의 32bit 위치값 (모터 1/2의 위치값)으로 구성됩니다.

■ 관련 명령어

- SP / EsA55A / SM / PE / PM / PME

■ 명령어 사용 예

- 아래의 과정은 포지션을 미리 저장한 후 원하는 포인트에서 포인트까지 정해진 시간 간격으로 이동하게 됩니다.

	명령어	설명
1	SP0,5000000,5000000; SP1,5001000,5002000; SP2,5003000,5006000; SP3,5009000,5008000; :	SP 명령으로 포지션 어레이를 정의합니다. SP0에 모터1의 포지션 5000000, 모터2의 포지션 5000000을 정의합니다. SP1에 모터1의 포지션 5001000, 모터2의 포지션 5002000을 정의합니다. SP2에 모터1의 포지션 5003000, 모터2의 포지션 5006000을 정의합니다.
2	ES0C00,0300;	SP로 설정된 포인트 데이터를 EEPROM에 저장합니다.
3	전원 RESET	EsA55A; 명령으로 저장된 데이터는 전원 Reset 된 경우에 적용됩니다.
4	PE0001;	Servo Enable [ID=1로 가정]
5	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
6	PM0,5,1000;	현재 위치에서 "SP0"로 이동하여 1초의 시간간격으로 SP1, SP2...SP5까지 이동합니다.
7	PM7,20,1000;	현재 위치에서 "SP7"로 이동하여 1초의 시간간격으로 SP8, SP9...SP20까지 이동
8	PME;	PM 명령 모드를 해제합니다.

7.9 Ps 명령으로 동작 중 정지 시키기

■ 작동 개요

- PA/SV/PP/PQ/Pp/Pq/p 명령에 의한 모션 제어 중 모터를 정지 시킵니다.
- Ss 또는 Sa 명령어에 의해 설정된 감속도로 정지 됩니다.
- 위치제어 중 정지 : 현재의 위치부터 정해진 감속률로 감속하여 정지합니다.  
속도제어 중 정지 : 현재의 속도부터 정해진 감속률로 감속하여 정지합니다.

■ 관련 명령어

- PA / SV / PP / PQ / Pp / Pq / p

■ 명령어 사용 예

	명령어	설명
1	PsA55A;	모터 1,2를 모두 현재의 위치에서 감속을 실시한 후에 정지시킵니다.
2	PsA5A5;	모터 1을 현재의 위치에서 감속을 실시한 후에 정지시킵니다.
3	Ps5A5A;	모터 2를 현재의 위치에서 감속을 실시한 후에 정지시킵니다.

## 7.10 PI 명령으로 위치제어 하기

### ■ 작동 개요

- PI 명령은 위치 제어시 증분량을 설정하여 제어하는 명령어입니다.
- 2채널용 제어기의 경우 P1, P2 명령으로 각 모터를 개별 제어 할 수 있습니다.
- PI, P1, P2 명령은 "SL" 명령으로 설정된 범위 내에서만 정상 작동 합니다.

### ■ 관련 명령어

- PI / P1 / P2 / PE / SS

### ■ 명령어 사용 예

- 아래의 과정은 모터 1,2 모두 100펄스 씩 "SS" 명령어로 설정 된 속도로 이동하는 예제입니다.

	명령어	설명
1	SS1000,1000;	SS 명령으로 Point to Point 간 이동속도를 설정합니다.
2	PE0001;	Servo Enable [ID=1로 가정]
3	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
4	PI100,100;	1,2채널 모두 1000rpm의 속도로 100펄스 이동합니다.

#### 참 고

- PI 명령의 경우 로직 처리 시 추가적인 연산이 필요하여 모터 1채널 당 약 150 $\mu$ s 정도의 지연이 발생할 수 있습니다.

## 7.11 GH/Gh 명령을 이용한 원점복귀

### ■ 작동 개요

- 2개의 리미트 센서와 1개의 홈 센서를 사용한 홈 동작을 수행합니다.
- 리미트센서를 LSxA(+), LSxC(-)에 홈 센서를 LSxB에 연결하고(1채널은 LSA, LSC,LSB) 모터 홈 동작을 실행 합니다.

### ■ 관련 명령어

- SM / SS / SI / PE / Q1 / PR / GT / Gt

### ■ 명령어 사용 예

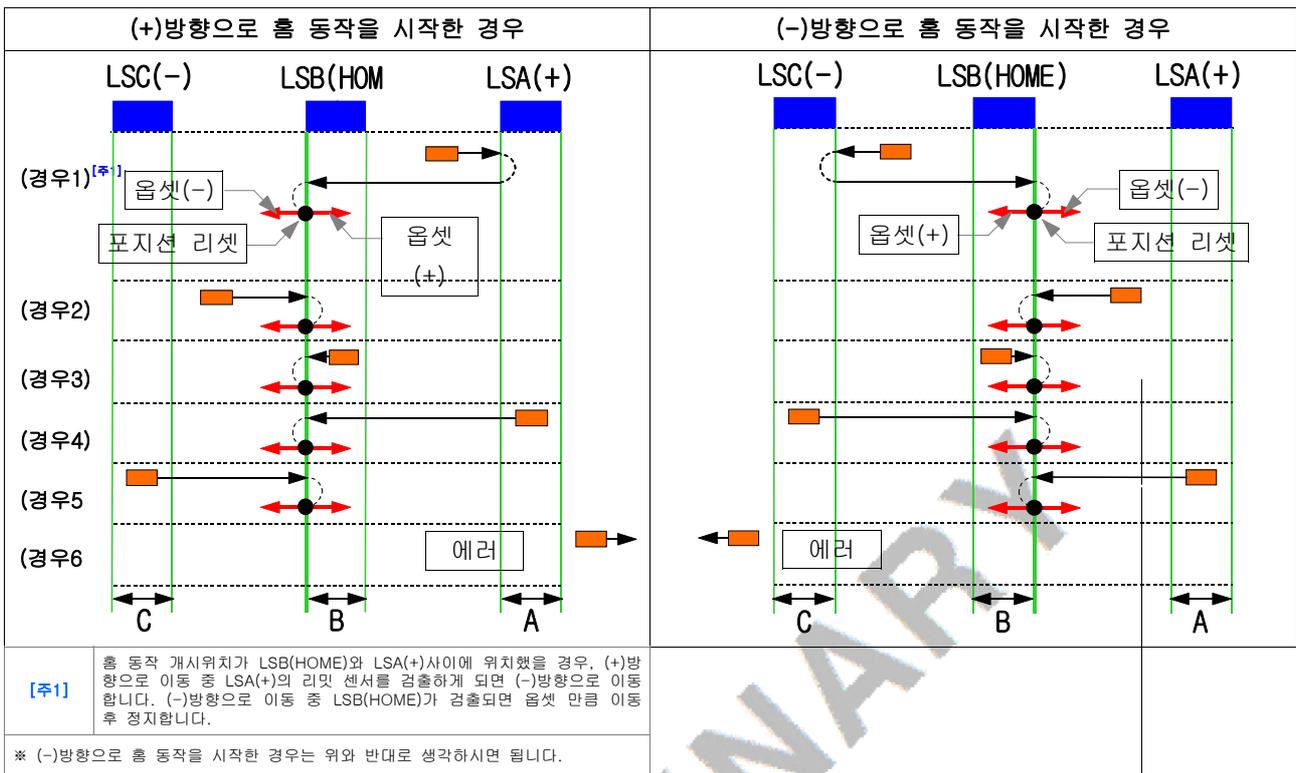
- 아래의 과정은 모터 1,2 모두 (+)방향으로 홈 동작을 시작하여 (+)방향의 리미트 센서를 검출한 후 (-) 방향으로 움직여 홈 센서를 검출하고 그 자리에 정지하도록 하는 예제입니다.

	명령어	설명
1	SS1000,2000;	SS 명령으로 Point to Point 간 이동속도를 설정합니다. 여기서 모터1의 속도명령값은 1000rpm, 모터2의 속도 명령값은 2000rpm입니다.
2	GT225,75;	GT 명령으로 리미트/홈 스위치의 OFF/ON threshold를 모터1에는 2.25V로 모터2는 0.75V로 설정합니다. (기본값을 바꾸고자 할 때만 필요하며, threshold값은 EsA55A; 명령어로서 EEPROM에 저장될 수 있습니다.)
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	GH03,11,0,0; <sup>[주1]</sup>	GH 명령으로 홈 동작을 개시합니다. 1,2채널 모두 +방향으로 회전하며 홈 위치 판단하고 홈 위치 검출 후 오피셋 이동 없이 정지 합니다.
6	Q1; 반복	Q1 명령으로 홈 동작의 상태를 판단합니다.

GH 03 , 11 , 0 , 0 ;	
	└─ 모터2 홈 센서 확인 후 오피셋(GH : 0.01° resolution, Gh : pulse)
	└─ 모터1 홈 센서 확인 후 오피셋(GH : 0.01° resolution, Gh : pulse)
	└─ 홈 동작 개시 방향 → 0:(-)방향, 1:(+)방향
	└─ 홈 동작 모터 설정 → 01:모터1, 02:모터2, 03:모터1,2
└─	└─ 홈 동작 명령어

참 고	
■	홈 동작 중에 홈 동작을 중지하기 위해서는 PR0001; 명령을 사용하면 됩니다.
■	홈 동작은 SI 명령어로 설정된 범위 내에서만 수행 가능하며 유효 범위를 벗어나는 경우 즉시 홈 동작을 중지하고 정지하게 됩니다.

■ GH/Gh 홈 명령의 순서도



■ : 홈 동작 개시위치  
 A,B,C : 각 센서의 ON 구간

CUBE - ST Series

## 7.12 GI/Gi 명령을 이용한 원점복귀

### 작동 개요

- GI 명령은 2개의 리미트 센서만을 사용하여 홈 동작을 수행합니다.
- 리미트센서를 LSxA(+), LSxB(-)에 연결하고(1채널에서는 LSA, LSB) 모터 홈 동작을 실행 합니다.

### 관련 명령어

- SM / SS / SI / PE / Q1 / PR / Gi / GT / Gt

### 명령어 사용 예

- 아래의 과정은 모터 1,2 모두 (+)방향으로 홈 동작을 시작하여 (+)방향의 리미트 센서를 검출한 후 (-) 방향으로 감속기 출력기준으로 1도 이동한 후 그 자리에 정지하도록 하는 예제입니다.

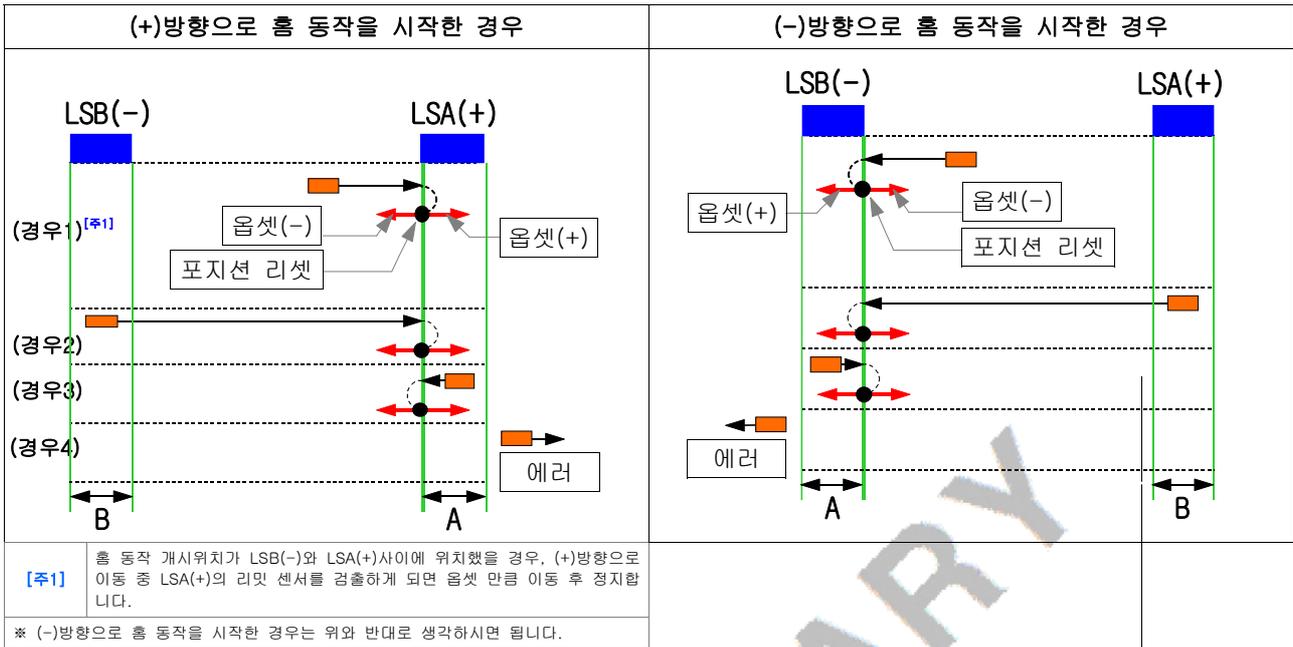
	명령어	설명
1	SS1000,2000;	SS 명령으로 Point to Point 간 이동속도를 설정합니다. 모터1의 이동속도 1000rpm, 모터2의 이동속도 2000rpm입니다.
2	GT225,75;	GT 명령으로 리미트/홈 스위치의 OFF/ON threshold를 2.25V와 0.75V로 설정합니다. (기본값을 바꾸고자 할 때만 필요하며, threshold값은 EsA55A: 명령어리로 EEPROM에 저장될 수 있습니다.)
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	GI03,11,100,100;	GI 명령으로 홈 동작을 개시합니다. 1,2채널 모두 (+)방향으로 회전하며 (+)방향 리미트 센서 검출 후 (-)방향으로 1도 회전 후 정지합니다.
6	Q1; 반복	Q1 명령으로 홈 동작의 상태를 판단합니다.

GI	03	,	11	,	100	,	100	;	
									모터2 홈 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
									모터1 홈 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
									홈 동작 개시 방향 → 0:(-)방향, 1:(+)방향
									홈 동작 모터 설정 → 01:모터1, 02:모터2, 03:모터1,2
									홈 동작 명령어

### 참고

- 홈 동작 중에 홈 동작을 중지하기 위해서는 PR0001; 명령을 사용하면 됩니다.
- 홈 동작은 SI 명령어로 설정된 범위 내에서만 수행 가능하며 유효 범위를 벗어나는 경우 즉시 홈 동작을 중지하고 정지하게 됩니다.

■ GI/Gi 홈 명령의 순서도



- : 홈 동작 개시위치
- A, B : 각 센서의 ON 구간

CUBE - ST Series

## 7.13 GJ/Gj 명령을 이용한 원점복귀

### 작동 개요

- GJ 명령은 1개의 홈 센서만을 사용하여 홈 동작을 수행합니다.
- 홈 센서를 LSxB에 연결하고(1채널에수는 LSB) 모터 홈 동작을 실행 합니다.
- GJ 명령은 홈센서가 ON된 상태나 OFF된 상태에서 모두 홈 동작이 가능합니다. 다만 홈 센서가 OFF된 상태에서 홈 동작을 개시할 경우 홈 센서의 방향으로 홈 동작을 개시해야만 합니다.

### 관련 명령어

- SM / SS / SI / PE / Q1 / PR / Gj / GT / Gt

### 명령어 사용 예

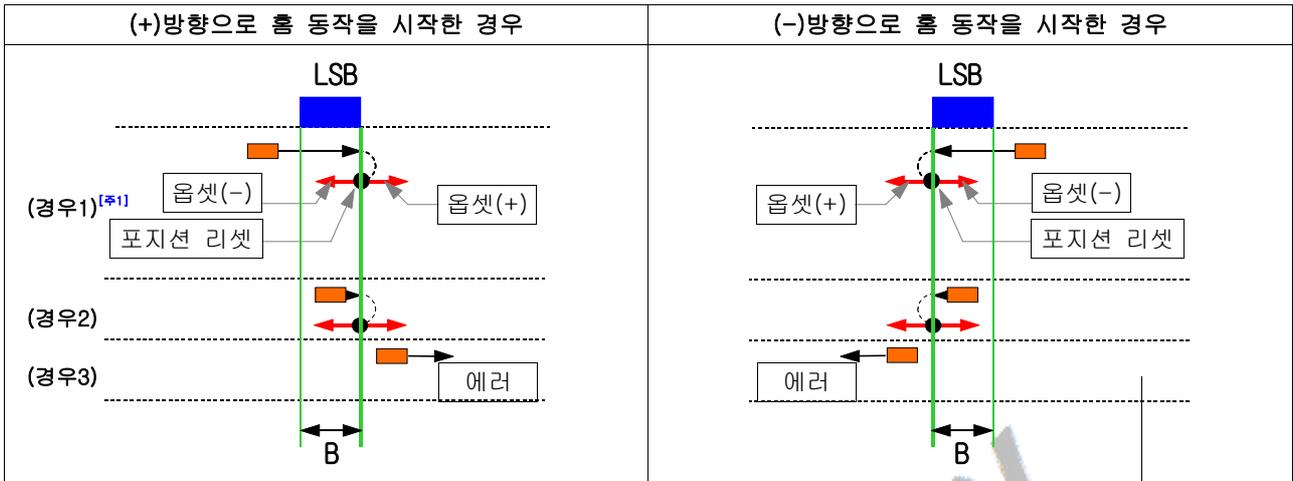
- 아래의 과정은 모터 1,2 모두 (+)방향으로 홈 동작을 시작했을 때 홈 센서가 감지된다는 가정하에 (+) 방향으로 동작을 시작하여 홈 센서를 감지한 후 (+)방향으로 계속 움직여 홈 센서가 OFF 되는 순간 정지하고 (-)방향으로 감속기 출력기준으로 1도 이동한 후 그 자리에 정지하도록 하는 예제입니다.

	명령어	설명
1	SS1000,2000;	SS 명령으로 Point to Point 간 이동속도를 설정합니다. 모터1의 이동속도 1000rpm, 모터2의 이동속도 2000rpm입니다.
2	GT225,75;	GT 명령으로 리밋/홈 스위치의 OFF/ON threshold를 2.25V와 0.75V로 설정합니다. (기본값을 바꾸고자 할 때만 필요하며, threshold값은 EsA55A; 명령어로서 EEPROM에 저장될 수 있습니다.)
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	GJ03,11,100,100;	GJ 명령으로 홈 동작을 개시합니다. 1,2채널 모두 (+)방향으로 회전하며 홈 센서 검출 후 (+)방향으로 계속 회전하다가 홈 센서 OFF시 (-)방향으로 1도 회전 후 정지합니다.
6	Q1; 반복	Q1 명령으로 홈 동작의 상태를 판단합니다.

GJ 03 , 11 , 100 , 100 ;			
			└─ 모터2 홈 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
			└─ 모터1 홈 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
		└─	─ 홈 동작 개시 방향 → 0:(-)방향, 1:(+)방향
	└─	─	─ 홈 동작 모터 설정 → 01:모터1, 02:모터2, 03:모터1,2
└─	─	─	─ 홈 동작 명령어

참고	
■	홈 동작 중에 홈 동작을 중지하기 위해서는 PR0001; 명령을 사용하면 됩니다.
■	센서가 ON된 상태에서 홈 동작을 시작할 때에는 홈 센서 OFF 후 정해진 옵셋만큼 이동 후 정지합니다.
■	홈 동작은 SI 명령어로 설정된 범위 내에서만 수행 가능하며 유효 범위를 벗어나는 경우 즉시 홈 동작을 중지하고 정지하게 됩니다.

■ GJ/Gj 홈 명령의 순서도



[주1] (+)방향으로 이동 중 홈 센서가 On → Off 가 검출되면 옴셋 만큼 이동 후 정지합니다.

\* (-)방향으로 홈 동작을 시작한 경우는 위와 반대로 생각하시면 됩니다.

- : 홈 동작 개시위치
- B : 홈 센서의 ON 구간

CUBE - ST Series

PRELIMINARY

## 7.14 GK/Gk 명령을 이용한 원점복귀

### ■ 작동 개요

- GK 명령은 1개의 홀 센서만을 사용하여 홀 동작을 수행합니다.
- 홀 센서를 LSxB에 연결하고 모터 홀 동작을 실행 합니다.
- GK 명령은 홀 센서가 ON된 상태에서만 홀 동작이 가능합니다. 만약 홀 센서가 OFF된 상태라면 GK 명령은 무시됩니다.

### ■ 관련 명령어

- SM / SS / SI / PE / Q1 / PR / Gk / GT / Gt

### ■ 명령어 사용 예

- 아래의 과정은 모터 1,2 모두 홀 센서가 감지된 상태(ON)에서 (+)방향으로 계속 움직여 홀 센서가 OFF 되는 순간 정지하고 (-)방향으로 감속기 출력기준으로 1도 이동한 후 그 자리에 정지하도록 하는 예제입니다.

	명령어	설명
1	SS1000,1000;	SS 명령으로 Point to Point 간 이동속도를 설정합니다.
2	GT225,75;	GT 명령으로 리밋/홀 스위치의 OFF/ON threshold를 2.25V와 0.75V로 설정합니다. (기본값을 바꾸고자 할 때만 필요하며, threshold값은 EsA55A: 명령어로 EEPROM에 저장될 수 있습니다.)
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	GK03,11,100,100;	GK명령으로 홀 동작을 개시합니다. 1,2채널 모두 (+)방향으로 회전하다가 홀 센서가 OFF 되는 순간 (-)방향으로 1도 회전 후 정지합니다.
6	Q1; 반복	Q1 명령으로 홀 동작의 상태를 판단합니다.

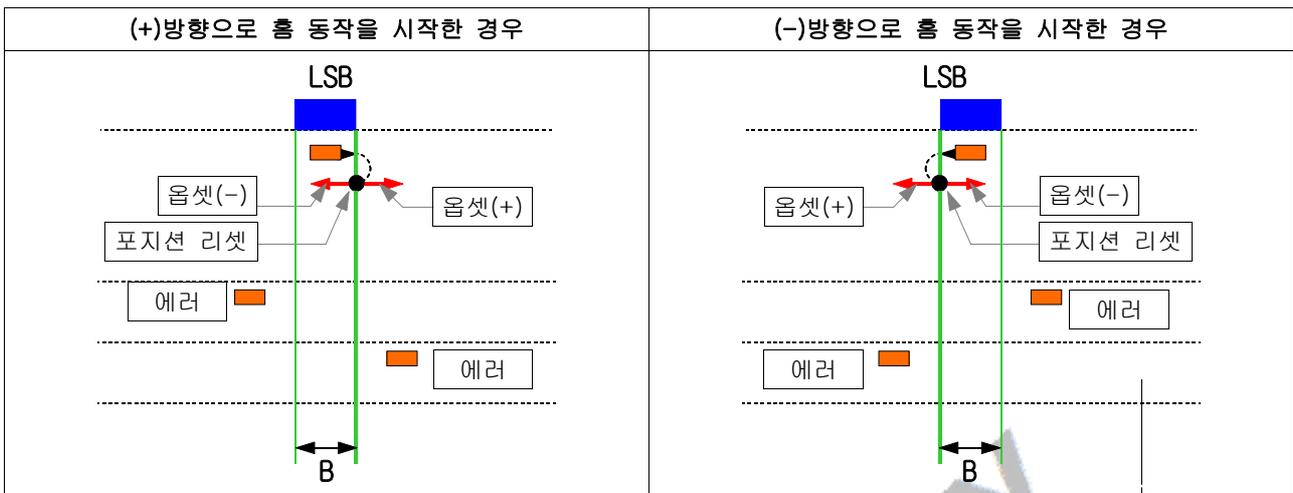
[주1] GK 03 , 11 , 100 , 100 ;

				└	모터2 홀 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
			└	—	모터1 홀 센서 확인 후 옵셋, Gi는 각도, Gi는 펄스 수(QEP)
		└	—	—	홀 동작 개시 방향 → 0:(-)방향, 1:(+)방향
	└	—	—	—	홀 동작 모터 설정 → 01:모터1, 02:모터2, 03:모터1,2
└	—	—	—	—	홀 동작 명령어

### 참 고

- 홀 동작 중에 홀 동작을 중지하기 위해서는 PR0001; 명령을 사용하면 됩니다.
- 홀 동작은 SI 명령어로 설정된 범위 내에서만 수행 가능하며 유효 범위를 벗어나는 경우 즉시 홀 동작을 중지하고 정지하게 됩니다.

■ GK/Gk 홈 명령의 순서도



[주1] LSB가 On되었을때 즉, 그림에서 경우(1)일때에만 홈 동작이 가능합니다. 경우(1)에서 (+)방향으로 이동 중 센서의 Off가 검출되면 오프셋 만큼 이동 후 정지합니다. 그외 경우(2), 경우(3)일 경우 홈 동작을 하지 않습니다.

\* (-)방향으로 홈 동작을 시작한 경우는 위와 반대로 생각하시면 됩니다.

- : 홈 동작 개시위치
- B** : 홈 센서의 ON 구간

CUBE - ST Series

PRELIMINARY

## 7.15 J/K/j/k 명령을 이용한 조그 동작

### ■ 작동 개요

- ▶ 속도제어시 조그동작 명령 : J; 또는 K;
- ▶ 위치제어시 조그동작 명령 : j; 또는 k;
- 조그 명령어 J; 또는 j; 또는 K; 또는 k; 가 설정 주기 안에 계속하여 전송되면 모터는 설정된 방향과 속도로 회전합니다. 조그 동작중에 조그 명령이 중지되면 모터는 회전을 멈춥니다.

### ■ 관련 명령어

- SM / SJ / Sa / PE / J / K / j / k

### ■ 속도제어 모드에서의 조그 동작

	명령어	설명
1	SJ11,1000,1000,30;	SJ 명령으로 회전방향, 회전속도, 지속시간을 설정합니다. 모터 1,2 모두 1000rpm으로 +방향으로 회전, 30ms 안에 J; 또는 K; 명령이 전송되지 않으면 모터 회전을 정지합니다.
2	Sa200,400,400,300;	Sa 명령으로 100ms당 모터1의 가속도 200rpm, 감속도 100rpm, 모터2의 가속도 400rpm, 감속도 300rpm의 가/감속을 설정합니다.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0909;	SM 명령으로 09번 속도제어 모드 선택합니다.
5	J; 또는 K;	1채널용 제어기의 경우 J; 명령으로 조그 동작을 실행합니다. 2채널용 제어기의 경우 모터1은 J; 모터2는 K; 명령으로 조그 동작을 실행합니다.

### ■ 위치제어 모드에서의 조그 동작

	명령어	설명
1	SJ11,2000,2000,30;	SJ 명령으로 회전방향, 회전속도, 지속시간을 설정합니다. 모터 1,2 모두 2000rpm으로 +방향으로 회전, 30ms 안에 j; 또는 k; 명령이 전송되지 않으면 모터 회전을 정지합니다.
	Sj11,2000,2000,200;	SJ 명령과 Sj 명령의 차이점은 마지막 파라미터에 있습니다. Sj 명령의 마지막 파라미터는 조그 명령시 조그 동작의 지속거리(pulse)를 의미합니다.
2	Ss500,500;	Ss 명령으로 가/감속 시간을 설정합니다.
3	PE0001;	Servo Enable [ID=1로 가정]
4	SM0808;	STEP모터 전용 제어 모드로 설정합니다.
5	j; 또는 k;	1채널용 제어기의 경우 j; 명령으로 조그 동작을 실행합니다. 2채널용 제어기의 경우 모터1은 j; 모터2는 k; 명령으로 조그 동작을 실행합니다.

CUBE - ST Series

## 7.16 사인파 위치제어 동작

### ■ 작동 개요

- 설정된 주파수와 설정된 진폭으로 모터의 변위가 변하도록 하는 기능입니다.
- 서보의 성능을 판단하는데 유용한 기능입니다.

### ■ 관련 명령어

- SM / S1 / S2 / PE

### ■ 명령어 사용 예

- 아래의 과정은 1번 모터에 대하여 주기 800ms, 진폭 5000펄스로 사인파 변위를 제어하는 예제입니다.

	명령어	설명
1	S1800,5000;	모터1에 대하여 주기 800ms이며 진폭 5000펄스의 사인파 변위를 정의합니다.
2	PE0001;	Servo Enable [ID=1로 가정]
3	SM0F0F;	사인파 위치제어 모드로 설정합니다.
4	S1400,5000;	모터1에 대하여 주기 400ms이며 진폭 5000펄스의 사인파 변위로 바꿉니다.

#### 참고

- S2 명령어는 모터 2에 대한 사인파 변위를 정의합니다.

## 8. FAULT 처리

### 8.1 개요

#### ■ 관련 명령어

- Q2 / PR / PPR / EC / ECR

#### ■ FAULT Code의 표시

- 제어 도중 Fault가 검출이 되면 해당 Fault Code가 제어기에 내장된 녹색 LED를 통해 표시됩니다. 이를 Flash code라고 하며, 녹색 LED가 On된 시간의 길이에 따라 긴 경우는 10자리를 의미하며, 짧은 경우 1자리를 의미합니다.  
예를 들어 Fault Code가 12인 경우에는 길게 한 번 짧게 두 번 깜박이게 됩니다. 만약 여러개의 Fault가 동시에 발생한 경우에는 순차적으로 반복해서 표시됩니다.
- Fault 발생시 그 내용을 확인하기 위해 Q2 명령을 사용할 수 있습니다.

#### ■ FAULT 발생시 복구

- Fault가 발생한 경우 제어기를 복구하려면 PR 명령을 사용하여 Fault Flag를 Clear 한 후 PE 명령을 통해 Servo를 다시 Enable 시킨 다음 사용하셔야 합니다.
- 제어기 Fault 중에는 PR 명령을 이용해서 복구가 되지 않는 Fault가 있습니다. 이 경우 제어기 전원을 OFF 한 후 Fault의 원인을 제거한 후 사용하여야 합니다.

#### ■ FAULT 처리 예

	명령어	설명
1	Q2;	Q2: 명령을 전송, 리턴되는 데이터를 분석하여 현재 Fault 상태를 판단함
2	PR0005;	Fault가 발생한 경우 복구가 가능한 Fault에 한하여 PR 명령으로 해당 5번 제어기의 Fault를 Clear함. PR 명령의 파라미터는 해당 제어기의 ID입니다.
3	PE0005;	PE 명령으로 해당 5번 모터제어보드의 출력을 enable 하여 제어를 개시합니다.

## 8.2 Fault 상태 표

### Fault와 관련된 Return code의 구조

- Q2 명령으로 Fault Code를 확인할 경우 16진수 형태를 가진 2 Word의 Fault Code가 Return 되며, 각 데이터의 의미는 아래와 같습니다.

Q2	HLHL	, HLHL	;	
		└		Fault Code(2nd Word)
	└	—		Fault Code(1st Word)
└	—	—		Command

### Fault Code Table [Q2 Command]

1ST WORD [Q2HLHL,HLHL]					
명령어	bit	Fault Code (LED)	Fault 발생조건	제어기에서의 조치	복구/점검/조치
Q2:	0	21	-모터1(21),모터2(31)의 구동 MOSFET gate를 차단하는 fault	-별도의 조치 없음.	-PRHLHL; 명령어로 복구 가능
	1	31	-저전압/과전압 fault에 의하여 세트됩니다.		
	2	11	-저전압 Fault -제어기의 초기화가 끝난 다음으로, 현재의 filtering된 공급전압이 전원에 투입이 되었을 때, filtering된 공급전압이 설정된 전압강하량(Sv 명령어로 설정가능, 기본값은 8V) 보다 전압 강하가 설정된 시간(Sv 명령어로 설정가능, 기본값 10ms) 동안 연속으로 발생하였을 때 세트됩니다.	-모터 1/2의 MOSFET 게이트를 off 시킴.  -모터 1/2의 MOTORx_gate_kill_fault_flg를 세트함.  -작동모드를 0으로 clear함.	-쇼트에 의한 전압강하이면 회로/배선 점검 -pulse성 전압강하이면 공급전압에 연결된 콘덴서의 용량을 보충해야함 -PRHLHL; 명령어로 복구 가능
	3	12	-과전압 Fault -제어기 초기화가 끝난 다음으로서, 모터의 게이트가 enable 되어있지 않을 때에만 적용하며, filtering된 현재 공급전압이 설정된 최대공급전압(Sv 명령어로 설정가능, 기본값은 34V) 보다 설정된 시간(Sv 명령어로 설정가능, 기본값 10ms)동안 연속으로 클 때 세트됩니다.	-모터 1/2의 MOSFET 게이트를 off 시킴.  -모터1/2의 MOTORx_gate_kill_fault_flg를 세트함.  -작동모드를 0으로 clear함.	-지나치게 높은 공급전압이므로 전원을 차단하고 규정공급전압을 재공급함 -PRHLHL; 명령어로 복구 가능

4	22	<p>-모터1(22), 모터2(32)의 과전류 Fault -구형 DC 제어기에서만 적용되며, MOSFET 게이트가 enable 되어있을 때로서 최대허용전류 초과에 의한 게이트 disable 신호가 1.024초 동안에 설정된 횟수(Si 명령어로 설정가능, 기본값 100회) 이상 발생할 때 세트됩니다.</p>	<p>-모터 1/2의 MOSFET 게이트를 off 시킴.  - 모터 1/2의 MOTORx_gate_kill_fault_flg 를 세트함.</p>	<p>-쇼트에 의한 최대허용 전류 초과라면 회로/배선 점검 - 최대허용전류 설정치가 너무 낮다면 SW 명령어로 최대허용전류 설정값을 높게 함 - PRHLHL: 명령어로 복구 가능</p>
5	32			
6	23	<p>-모터1(23), 모터2(33) Excessive Error -MOSFET 게이트가 enable 되어있을 때로서, 위치모드에서는 순간위치오차, 속도모드에서는 순간속도오차, 전류모드에서는 전류오차가 허용크기(Sx 명령어로 설정가능, 기본값 : 위치는 감속기 출력기준으로 10도, 속도는 1000 또는 SG 명령어로 설정되는 MOTOR1_SPEED_MAX/5, 전류오차는 200 또는 SI 명령어로 설정되는 current1_limit/2 bit)보다 서보 enable 된 상태에서 일정시간(Sx 명령어로 설정가능, 기본값 10초) 계속초과 될 때 세트됩니다.</p>	<p>-모터 1/2의 MOSFET 게이트를 off 시킴.  - 모터 1/2의 MOTORx_gate_kill_fault_flg 를 세트함.</p>	<p>-모터의 stall 때문이라면 기구적으로 해결하고 재가동함. 그것이 아니라면 오차, 허용크기나 시간을 늘려야함. -PRHLHL: 명령어로 에러 상태를 벗어날 수 있는데 이때 직전의 위치 모드였다면 명령위치와 현재위치가 다르므로 순간적으로 이동이 개시될 수있으니 주의를 요함. -속도나 전류모드의 경우는 PRHLHL: 명령어를 주지 않더라도 다시 기동할 수 있으나, 에러 플래그는 그대로 있으며 에러검출동작을 하지 않음.</p>
7	33		<p>-01/02/03/04 위치제어모드에서는 현 위치를 유지하도록 함. -05/06 속도모드에서는 정지 시킴. -07 전류모드에서는 전류값을 0이 되게함.</p>	

CUBE - ST Series

■ Fault Code Table[Q2 Command]

1ST WORD [Q2HLHL,HLHL]						
명령어	bit	Fault Code (LED)	Fault 발생조건	제어기에서의 조치	복구/점검/조치	
Q2:	8	24	-모터1 Over Current Fault(24), Over Current Fault 시간 체크 개시 flag -DC 모터의 경우는 해당모터 코일에 흐르는 전류, BLDC의 경우는 공급전압의 +단자를 통해 들어가는 전류가 설정 전류(SI 명령어로 설정가능, 기본값은 제어기마다 다르게 설정되어 있음) 보다 큰전류가 일정시간(SI 명령어로 설정가능, 기본값은 1000ms) 이상 경과하였을 때 세트됩니다.	-PM 명령을 수행중일 때는 현 위치를 고수하도록 함. -DC 모터의 경우는 PWM 출력값을 1/4로 줄이거나 전류제어를 하고 있는 경우는 전류명령값을 1/10로 줄임. -BLDC 모터의 경우는 속도명령값이나 전류명령값을 1/10로 줄임. -IRKCK201을 사용하지 않은 BLDC 모터제어기의 경우는 current1_over_K_flg==1 인 기간동안 전류리밋값에 해당하는 전류제어를 실시함.	-마지막 fault 발생 후 일정시간(SI 명령어로 설정가능, 기본값은 5000ms)이 지나면 해당 flag가 모두 clear됨. -IRKCK201을 사용하지 않은 BLDC 모터제어기의 경우는RJM_VER7의 과전류보호 로직에 대한 별도의 자료를 참조할 것.	
	10	25	-모터1 Stall Fault(25), Stall Fault 시간체크 개시 flag -일정시간(SI 명령어로 설정가능, 기본값은 1000ms)동안 과전류상태에서 감속 후 기준으로 변위가 1도이상 발생하지 않았을 경우에 세트됩니다.	-제어모드 01/02/03/04에서는 현 위치에서 감속후 기준으로 2도 후퇴하고 정지함. -제어모드 05/06에서는 속도설정값을 0으로 세트함.	-PPR 명령으로 Stall 된 상태에서 에러 해제 가능.	
	11					
	12	34	-모터2 Over Current Fault(34), Over Current Fault 시간 체크 개시 flag -DC 모터의 경우는 해당모터 코일에 흐르는 전류, BLDC의 경우는 공급전압의 +단자를 통해 들어가는 전류가 설정 전류(SI 명령어로 설정가능, 기본값은 제어기마다 다르게 설정되어 있음) 보다 일정시간(SI 명령어로 설정가능, 기본값은 1000ms) 이상 경과하였을 때 세트됩니다.	-PM 명령을 수행중일 때는 현 위치를 고수하도록 함. - DC 모터의 경우는 PWM 출력값을 1/4로 줄이거나 전류제어를 하고 있는 경우는 전류명령값을 1/10로 줄임. - BLDC 모터의 경우는 속도명령값이나 전류명령값을 1/10로 줄임. -IRKCK201을 사용하지 않은 BLDC 모터제어기의 경우는 current2_over_K_flg==1 인 기간동안 전류리밋값에 해당하는 전류제어를 실시함.	-마지막 fault 발생 후 일정시간(SI 명령어로 설정가능, 기본값은 5000ms)이 지나면 해당 flag가 모두 clear됨. -IRKCK201을 사용하지 않은 BLDC 모터제어기의 경우는 RJM_VER7의 과전류보호 로직에 대한 별도자료를 참조할 것.	
	13					
	14	35	-모터2 Stall Fault(35), Stall Fault 시간체크 개시 flag -일정시간(SI 명령어로 설정가능, 기본값은 1000ms) 동안의 과전류상태에서 감속후 기준으로 변위가 1도이상 발생하지 않았을 경우에 세트됩니다.	-제어모드 01/02/03/04에서는 현 위치에서 감속후 기준으로 2도 후퇴하고 정지함. -제어모드 05/06에서는 속도설정값을 0으로 세트함.	-PPR 명령으로 Stall 된 상태에서 에러를 해제 가능.	
	15					

2ND WORD [Q2HLHL,HLHL]					
명령어	bit	Fault Code (LED)	Fault 발생조건	제어기에서의 조치	복구/점검/조치
Q2;	0		-모터1/2 Positive Position Limit Over Fault -위치제어모드(02/03)에서 현 위치가 (+)Limit (SL 명령어로 설정 가능) 설정 값보다 크게 된 경우에 세트됩니다. -이 상태가 아니면 0 으로 clear 됩니다.	-별도의 조치 없음.	-리미트 안쪽으로 옮겨오도록 위치명령을 내리거나, 엔코더 값을 QE 명령어로 초기화 함.
	1				
	2		-모터1/2 Negative Position Limit Over Fault -위치제어모드(02/03)에서 현 위치가 (-)Limit (SL 명령어로 설정 가능) 설정 값보다 작게 된 경우에 세트됩니다. -이 상태가 아니면 0 으로 clear 됩니다.	-별도의 조치 없음.	-리미트 안쪽으로 옮겨오도록 위치명령을 내리거나, 엔코더 값을 QE 명령어로 초기화 함.
	3				
	4	13	-IRMCK201 error -CUBE-BL Series에서 에러가 발생한 경우에 세트됩니다.	-별도의 조치 없음.	-Q3; 명령어를 사용하여 에러 상태를 알 수 있으며 이를 이용하여 고장을 처리함. -PRHLHL; 명령어로 복구 가능.
	5		-RS232 통신과정에서 에러가 발생하였음을 알려줍니다.	-별도의 조치 없음.	-에러의 상태를 알려면 EC?; 명령어를 보내면 되며, 에러 플래그를 없애고 에러를 복구하려면 ECR; 명령어를 보내면 됨.
	6		-RS485 통신과정에서 에러가 발생하였음을 알려줍니다.	-별도의 조치 없음.	
	7		-CAN 통신과정에서 에러가 발생하였음을 알려줍니다.	-별도의 조치 없음.	
	8	26	-모터1(26),모터2(36)의 권선과온도에러 -SX 명령어로 설정되는 작동모드의 bit2와 bit3가 모두 1이고, 모터의 특성 파라미터가 ZQ/ZR 명령어로 적절히 입력되었을때만 유효하며, 모터의 권선온도를 추정한 값이 ZT 명령어로 설정되는 허용온도를 초과하면 에러를 발생함.	-전류제어에 사용하는 전류크기의 제한값(Sw명령어로 설정됨)을 점차 작게하며, 허용온도를 초과하지 않으면 점차 회복 시킴.	-전류제어에 사용하는 전류크기 제한값의 크기가 Sw명령어로 설정된 값으로 회복되면 에러 flag를 clear 시킴.
	9	36			
10	17	-공급전압이 MOSFET BREAKDOWN VOLTAGE에 접근한 에러(17) -제어기 초기화가 끝난 다음으로서, 모터의 게이트가 enable 되어있을 때에만 적용하며, filtering된 현재 공급전압이 설정된 MOFSET breakdown voltage(Z6 명령어로 설정된 전압값의 1.25배) 보다 크면 세트됨. (2010년 5월 22일자 이후의 펌웨어에서만 적용됨)	-모터 1/2의 MOSFET 게이트를 off시킴. -모터1/2의 MOTORx_gate_kill_fault_flag를 세트함. -작동모드를0으로 clear 함.	-감속시의 역기전력이나, 전원장치의 불안정으로 발생한 것이므로, 감속관련 파라미터를 설정하는 SD 및 Sd명령을 사용하거나, 전원장치를 점검하여야함. -PRHLHL; 명령어로 복구 가능.	

Q2:	11	18	<p>-모터제어기의 모터구동 bridge1 과발열 에러(18)</p> <p>-모터제어기 모터구동 bridge1에서 발생하는 발열량이 방열판에서 감당 가능한 범위를 벗어났을 때 세트됨. (ZW 명령어로 설정하는 방열판 용량의 5배를 발열허용범위의 초기치로 하여 방열판용량의 1배를 초과하는 시간이 경과할수록 발열허용범위를 낮춤)(2010년 9월22일자 이후 펌웨어에서만 적용됨)</p>	<p>-전류제어에 사용하는 전류크기의 제한값 (Sw 명령어로 설정됨)을 점차 작게하며, 발열량이 발열 허용범위 이내이면 회복 시킴.</p>	<p>-전류제어에 사용하는 전류크기의 제한값의 크기가 Sw 명령어로 설정된 값으로 회복되면 에러 flag를 clear 시킴.</p>
	12	19	<p>-모터제어기의 모터구동 bridge2 과발열 에러(19)</p> <p>-모터제어기 모터구동 bridge2에서 발생하는 발열량이 방열판에서 감당 가능한 범위를 벗어났을 때 세트됨. (ZW 명령어로 설정하는 방열판 용량의 5배를 발열허용범위의 초기치로 하여 방열판용량의 1배를 초과하는 시간이 경과할수록 발열허용범위를 낮춤)(2010년 9월22일자 이후 펌웨어에서만 적용됨)</p>	<p>-전류제어에 사용하는 전류크기의 제한값 (Sw 명령어로 설정됨)을 점차 작게하며, 발열량이 발열 허용범위 이내이면 회복 시킴.</p>	<p>-전류제어에 사용하는 전류크기의 제한값의 크기가 Sw 명령어로 설정된 값으로 회복되면 에러 flag를 clear 시킴.</p>

CUBE - ST Series

PRELIMINARY

# 9. 시퀀스 자동화 프로그램

## 9.1 개요

### ■ 시퀀스 자동화 프로그램이란

- 시퀀스 자동화 프로그램이란 사용자가 직접 당사가 제공하는 명령어를 사용하여 미리 정의된 동작을 수행하도록 프로그래밍할 수 있는 기능을 의미합니다.

### ■ 관련 명령어

- XP / XPP / XPsA55A / XPT / XPt / XPP / XPF / XPS

### ■ 시퀀스 자동화 프로그램의 개요

- 시퀀스 자동화 프로그램은 XP 명령어를 사용하여 입력합니다.
- 작성된 프로그램은 XPsA55A; 명령어를 사용하여 내장된 EEPROM에 저장할 수 있습니다.
- 시퀀스 자동화 프로그램의 실행은 전원 투입시 SX 명령어로 설정된 operation\_mode\_SWITCH의 bit15를 체크하여 bit15가 1일 때 개시됩니다.
- 시퀀스 자동화 프로그램은 매 10ms마다 저장된 시퀀스 제어코드를 해독하여 실행합니다.
- 시퀀스 자동화 프로그램을 Step-by-Step 모드(디버깅모드)로 실행하고자 할 때에는 operation\_mode\_SWITCH의 bit14를 1로 SET 한 후 EsA55A; 명령어로 저장합니다. 전원을 리셋한 후 XPS; 명령어로 Step 모드 실행을 시작할 수 있습니다.
- 시퀀스 자동화 프로그램의 크기는 1280 글자까지 입력이 가능합니다.

### ■ 시퀀스 자동화 프로그램의 주요변수

변수명	설명
■ PORT_STATUS_FOR_X_CONTROL	Port의 ON/OFF 상태를 가지고 있는 16bit 변수로서 그 값은 매 10ms마다 갱신됩니다.
■ X_PORT_STATUS	매 10ms 마다 PORT_STATUS_FOR_X_CONTROL 값으로 갱신되며 시퀀스 제어 프로그램 내에서 조작 가능합니다.
■ X_process_cnt	시퀀스 제어 프로그램의 실행과정을 표시하는 변수로서 시퀀스 제어 프로그램 내에서 조작이 가능합니다.
■ X_STATUS_FLAG	시퀀스 제어 프로그램에 의하여 조작이 가능한 16bit의 변수입니다.
■ X_timer0	매 10ms 마다 1씩 증가하는 타이머입니다.
■ X_timer0_period	X_timer0의 주기값으로서 기본값은 100(1초)입니다.
■ X_counter0	시퀀스 제어 프로그램에서 사용하기 위한 카운터입니다.

PORT_STATUS_FOR_X_CONTROL		
bit0	LS1A	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.
bit1	LS1B	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.
bit2	LS1C	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.
bit3	LS2A	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.
bit4	LS2B	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.

bit5	LS2C	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다.
bit6	MOT_EN_N_port	0 이면 모터가 enable 된 상태이고 1이면 disable된 상태입니다.
bit7	OFF12V_port	0 이면 charge pump가 작동중이며 1이면 off된 상태입니다.
bit8	DISABLE1_IN_port	0 이면 L6207이 정상이며 1이면 fault 상태입니다. (CUBE-DC2402-D1x, CUBE-BL2402-D1x, CUBE-ST2402-D1x 인 경우에만 적용됩니다.)
bit9	DISABLE2_IN_port	0 이면 L6207이 정상이며 1이면 fault 상태입니다. (CUBE-DC2402-D1x, CUBE-BL2402-D1x, CUBE-ST2402-D1x 인 경우에만 적용됩니다.)
bit10	LS1D_port	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다. (LS1D port가 있는 모델에만 적용됩니다.)
bit11	LS2D_port	0 이면 GND 단자와 short 상태이고 1이면 open 상태입니다. (LS2D port가 있는 모델에만 적용됩니다.)
bit12	EXT_FAULT1_port	0 이면 출력이 OFF 된 상태이며 1이면 ON 상태입니다.
bit13	EXT_FAULT2_port	0 이면 출력이 OFF 된 상태이며 1이면 ON 상태입니다.
bit14	-	
bit15	-	

■ 시퀀스 자동화 프로그램을 입력하려면

- SX 명령을 사용하여 operation\_mode\_SWITCH bit15를 0으로 Clear 합니다.
- XP 명령을 사용하여 프로그램을 입력합니다.
- 프로그램 입력이 완료되었으면 "XPsA55A;" 명령을 사용하여 EEPROM에 저장합니다.

■ 시퀀스 자동화 프로그램을 실행시키려면

- SX 명령을 사용하여 operation\_mode\_SWITCH bit15를 1로 Set 합니다.
- EsA55A; 명령으로 operation\_mode\_SWITCH 값을 저장한 후 전원을 끕니다.
- 전원을 다시 켜면 시퀀스 자동화 프로그램이 실행됩니다.
- XPt; 명령을 사용하여 현재의 X\_timer0, X\_timer0\_period, X\_counter0의 값을 볼 수 있습니다.
- XPP; 명령을 사용하여 현재와 직전의 시퀀스제어 프로세스 카운트 (X\_process\_cnt)를 볼 수 있습니다.

■ 시퀀스 자동화 프로그램을 디버깅 하려면

- SX 명령을 사용하여 operation\_mode\_SWITCH bit14를 1로 SET 한 후 "EsA55A;"명령으로 저장하고 전원을 끕니다.
- 전원을 재 투입한 후 "XPS;" 명령을 입력할 때마다 Step-by-Step 모드로 시퀀스제어 프로그램이 실행되므로 시퀀스제어 프로그램을 디버깅할 수 있습니다.

## 9.2 샘플 프로그램

## ■ 시퀀스 자동화 샘플 프로그램

- 다음 시퀀스 제어 프로그램은 LS1A가 ON되면 2000rpm으로 회전, LS1B가 ON되면 -2000rpm으로 회전하고 LS2B가 ON되면 정지하는 예제 프로그램입니다. 모터의 방향 전환시 LS2B로 정지 후 실행시켜야 합니다.
- 시퀀스 자동화 프로그램을 입력하기 전에 아래 절차를 먼저 수행하여 주십시오.
  1. 제어기의 전원을 인가합니다.
  2. SX 명령을 입력하여 파라미터의 bit15(MSB)를 0 으로 설정합니다.
  3. "EsA55A;" 명령을 입력하여 저장합니다.
  4. 전원을 껐다가 켜 다음 프로그램을 입력합니다.

```

XPXPLT0100IN; // 0x100(256) * 10ms 동안은 X_process_cnt를 증가 시키기만 합니다.
XPXPEQ0100INSM0505; // SM0505; 을 실행하고 X_process_cnt를 증가 시킵니다.
XPXPEQ0101INPE0001; // PE0001; 을 실행하고 X_process_cnt를 증가 시킵니다.
XPXPLT0111IN; // 0x100(256) * 10ms 동안은 X_process_cnt를 증가 시키기만 합니다.
XPXPEQ0111AN0001; // X_PORT_STATUS의 LSB만 남기고 모두 0으로 만듭니다.
XPXPEQ0111EQ0000JP0200SV2000,2000; // LS1A가 ON되면 SV2000,2000; 를 실행하고 X_process_cnt를 0x0200으로 강제 설정 합니다.
XPXPEQ0111R1JP0200SV-2000,-2000; // LS1B가 ON되면 SV-2000,-2000; 를 실행하고 X_process_cnt를 0x0200으로 강제 설정 합니다.
XPXPEQ0111ES; // 뒷부분의 프로그램을 확인하지 않고 빠져 나갑니다.
XPXPEQ0200R4JP0111SV0,0; // LS2B가 ON되면 SV0,0; 를 실행하고 X_process_cnt를 0x0111으로 강제설정 합니다.
XPXPEQ0200JP0200; // X_process_cnt를 0x0200으로 강제 설정 합니다.
XPXE; // 프로그램의 끝을 설정 합니다.
XPsa55A; // 프로그램을 EEPROM에 저장 합니다.

```

- 시퀀스 자동화 프로그램을 실행시키려면 다음을 수행하여 주십시오.
  - SX 명령을 입력하여 파라미터의 bit15(MSB)를 1 로 설정합니다.
  - "EsA55A;" 명령을 입력하여 저장합니다.
  - 전원을 껐다가 켜면 시퀀스 제어 프로그램이 수행됩니다.

9.3 예약어

■ 시퀀스 자동화 프로그램의 예약어

OR / AN / XO / EQ / Sn / Rn / LD / or / an / xo / eq / sn / rn / ld / IN / JP / ES

예약어	설명
XPLTHLHLORHLHL;	X_process_cnt가 HLHL보다 작으면 X_PORT_STATUS 값에 HLHL을 OR 시킵니다.
XPLTHLHLANHLHL;	X_process_cnt가 HLHL보다 작으면 X_PORT_STATUS 값에 HLHL을 AND 시킵니다.
XPLTHLHLXOHLHL;	X_process_cnt가 HLHL보다 작으면 X_PORT_STATUS 값에 HLHL을 XOR 시킵니다.
XPLTHLHLLEQHLHL명령어 <sup>[주1]</sup> ;	X_process_cnt가 HLHL보다 작으면 X_PORT_STATUS 값이 HLHL과 같을 때 명령어를 수행합니다.
XPLTHLHLsn명령어;	X_process_cnt가 HLHL보다 작으면 PORT_STATUS_FOR_X_CONTROL 값의 n 번째 비트가 1이면 명령어를 수행합니다.
XPLTHLHLrn명령어;	X_process_cnt가 HLHL보다 작으면 PORT_STATUS_FOR_X_CONTROL 값의 n 번째 비트가 0이면 명령어를 수행합니다.
XPLTHLHLLD;	X_process_cnt가 HLHL보다 작으면 PORT_STATUS_FOR_X_CONTROL 값을 X_PORT_STATUS 값에 복사합니다.
XPLTHLHLorHLHL;	process counter가 HLHL보다 작으면 X_STATUS_FLAG 값에 HLHL을 OR 시킵니다.
XPLTHLHLanHLHL;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값에 HLHL을 AND 시킵니다.
XPLTHLHLxoHLHL;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값에 HLHL을 XOR 시킵니다.
XPLTHLHLleqHLHL명령어;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값이 HLHL과 같을 때 명령어를 수행합니다.
XPLTHLHLsn명령어;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값의 n 번째 비트가 1이면 명령어를 수행합니다.
XPLTHLHLrn명령어;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값의 n 번째 비트가 0이면 명령어를 수행합니다.
XPLTHLHLldHLHL;	X_process_cnt가 HLHL보다 작으면 X_STATUS_FLAG 값에 HLHL값을 복사합니다.
XPLTHLHLIN명령어;	X_process_cnt가 HLHL보다 작으면 X_process_cnt 값을 증가시키고 명령어를 수행합니다.
XPLTHLHLJPHLHL명령어;	X_process_cnt가 HLHL보다 작으면 X_process_cnt 값을 HLHL으로 세트하고 명령어를 수행합니다.
XPLTHLHLES;	X_process_cnt가 HLHL보다 작으면 이후의 명령을 해독하여 처리하지 않고 빠져나갑니다.

[주1] 명령어는 IN, JPHLHL을 앞에 포함하거나 안 하거나 할 수 있으며, 이어서 뒤에는 blank일 수도 있고, 제어기가 처리 가능한 명령어일 수도 있습니다. IN, JPHLHL 이외의 명령어가 있을 경우에 process counter는 1이 자동 증가되며, 뒷부분에 있는 나머지 시퀀스 제어 프로그램은 처리하지 않습니다. TN, JPHLHL의 명령어가 있을 경우에는 process counter를 변화시키며, 뒷부분에 있는 나머지 시퀀스 제어 프로그램은 처리하지 않습니다.



■ 시퀀스 자동화 프로그램의 예약어

CL / CG / C< / C> / C= / C+ / C- / CR

예약어	설명
XPLTHLHLCLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (X_counter0의 값 <= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLCGHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (X_counter0의 값 >= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLCLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (X_counter0의 값 < (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (X_counter0의 값 > (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (X_counter0의 값 = (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 X_counter0의 값을 1 증가시키고 명령어를 수행합니다.
XPLTHLHLCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 X_counter0의 값을 1 감소시키고 명령어를 수행합니다.
XPLTHLHLCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 X_counter0의 값을 0으로 clear 시키고 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

BL / BG / B< / B> / B= / bL / bG / b< / b> / b=

예약어	설명
XPLTHLHLBLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS1C의 AD 변환값 <= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLBGHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS1C의 AD 변환값 >= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLBCLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS1C의 AD 변환값 < (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLBCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS1C의 AD 변환값 > (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLBCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS1C의 AD 변환값 = (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLbLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS2C의 AD 변환값 <= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLbGHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS2C의 AD 변환값 >= (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLbCLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS2C의 AD 변환값 < (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLbCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS2C의 AD 변환값 > (UINT)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLbCHHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (LS2C의 AD 변환값 = (UINT)HLHL)일 때 명령어를 수행합니다.



■ 시퀀스 자동화 프로그램의 예약어

WL / WG / W< / W> / W= / WI / WO / Wi / Wo

예약어	설명
XPLTHLHLWLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 값 <= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLWGHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 값 >= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLW<HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 값 < (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLW>HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 값 > (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLW=HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 값 = (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLWiHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 절대값 <= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLW0HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 절대값 >= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLWiHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 절대값 < (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLW0HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed2_RPM의 절대값 > (int)HLHL)일 때 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

vL / vG / v< / v> / v= / vI / vO / vI / vO

예약어	설명
XPLTHLHLvLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 값 <= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLvGHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 값 >= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLv<HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 값 < (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLv>HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 값 > (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLv=HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 값 = (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLvIHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 절대값 <= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLv0HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 절대값 >= (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLvIHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 절대값 < (int)HLHL)일 때 명령어를 수행합니다.
XPLTHLHLv0HLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (speed1_cmd_old의 절대값 > (int)HLHL)일 때 명령어를 수행합니다.



■ 시퀀스 자동화 프로그램의 예약어

PL / PG / P< / P> / P=

예약어	설명
XPLTHLHLPLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1 <= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLPGHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1 >= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLP<HLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1 < (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLP>HLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1 > (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLP=HLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1 == (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

QL / QG / Q< / Q> / Q=

예약어	설명
XPLTHLHLQLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position2 <= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLQGHHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position2 >= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLQ<HLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position2 < (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLQ>HLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position2 > (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLQ=HLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position2 == (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

pL / pG / p< / p> / p=

예약어	설명
XPLTHLHLpHLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1_set <= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLpGHHLHLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1_set >= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLp<HLHLHLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1_set < (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLp>HLHLHLHLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1_set > (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLp=HLHLHLHLHLHLHLHLHLHLHL 명령어;	X_process_cnt가 HLHL보다 작으면 (position1_set == (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

qL / qG / q< / q> / q=

예약어	설명
XPLTHLHLqLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (position2_set <= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLqGHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (position2_set >= (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLq<HLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (position2_set < (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLq>HLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (position2_set > (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLq=HLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (position2_set == (ULONG)HLHLHLHL)일 때 명령어를 수행합니다.

■ 시퀀스 자동화 프로그램의 예약어

nL / nG / n< / n> / n= / nI / nO / ni / no

예약어	설명
XPLTHLHLnLHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 값 <= (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLnGHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 값 >= (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLn<HLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 값 < (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLn>HLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 값 > (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLn=HLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 값 == (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLnIHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 절대값 <= (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLnOHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 절대값 >= (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLniHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 절대값 < (long)HLHLHLHL)일 때 명령어를 수행합니다.
XPLTHLHLnoHLHLHLHLHL명령어;	X_process_cnt가 HLHL보다 작으면 (HLHL번지의 절대값 > (long)HLHLHLHL)일 때 명령어를 수행합니다.

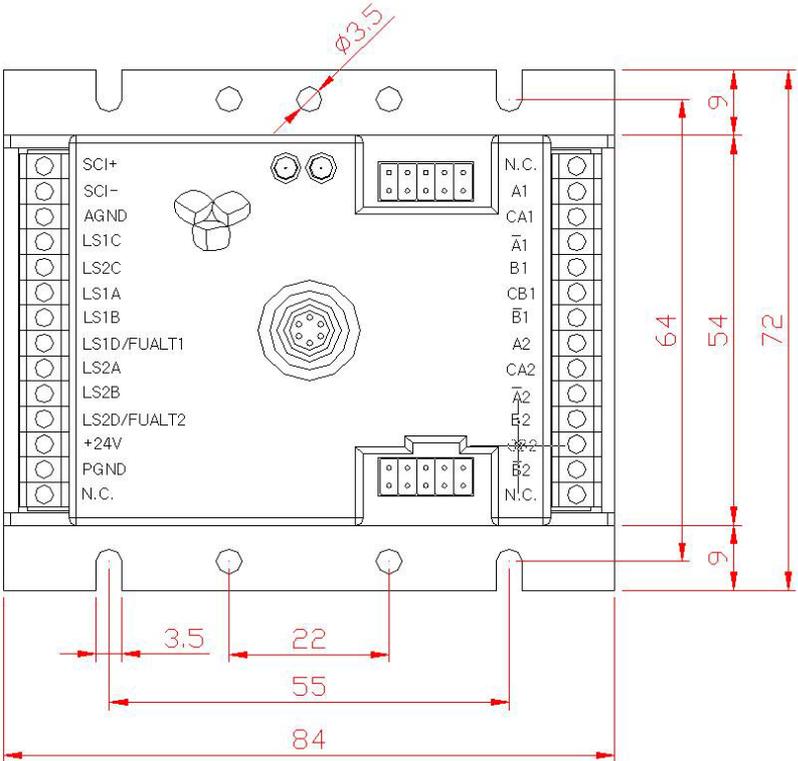
■ 시퀀스 자동화 프로그램의 예약어

XPLE / XPEQ / XC / XE

예약어	설명
XPLEHLHLxxxxxxx;	X_process_cnt가 HLHL보다 작거나 같으면 xxxxxxxx를 수행함.
XPEQHLHLxxxxxxx;	X_process_cnt가 HLHL과 같으면 xxxxxxxx를 수행함.
XC;	use RS485 or CAN for communication
XE;	프로그램의 끝을 나타냄.

# 10. 제품 DIMENSION

## ■ CUBU-ST2402-DIIE SIZE



CUBE - ST Series

PRE

