MW-SBL24D200S-U



User's Manual

©Copyright NTREXLAB

(주)엔티렉스
NTREX
Nano Technology Revolution eXecute



※ 사용하시기 전에

- MoonWalker SBL24D200S-U 드라이버를 구입해 주셔서 감사합니다.
- 사용자 설명서에는 SBL24D200S-U 드라이버의 주의사항, 제품사양, 취급방법 등이 기재되어 있습니다.
- 사용자 설명서를 잘 읽어보신 후 SBL24D200S-U 드라이버를 안전하게 사용하여 주십시오.
- 사용자 설명서는 사용하는 사람이 언제든지 볼 수 있는 위치에 잘 보관해 주기 바랍니다.

※ 일반 주의사항

- 사용자 설명서에 포함된 정보는 정확하고 신뢰성이 있는 내용입니다. 그러나 출판 당시 발견되지 않은 오류가 있을 수 있으니 사용자는 자신의 제품 검증을 수행하시기 바라며, 전적으로 사용자 설명서에 포함된 정보에 의존하지 마시기 바랍니다.
- 사용자가 임의로 제품을 개조하는 것은 당사의 보증 범위 밖이므로 책임지지 않습니다.

※ 안전 주의사항

- 드라이버의 설치, 점검, 보수 전에는 반드시 사용자 설명서를 숙지하신 후 실시하여 주시기 바랍니다.
- 드라이버의 설치, 점검, 보수시에는 적합한 자격을 가진 사람이 실시하여 주십시오.
- 드라이버가 손상되어 있거나 또는, 부품이 빠져 있는지 확인하십시오. 비정상적인 제품을 설치하거나 운전 시 기계 파손 또는, 사용자 부상의 위험이 있습니다.
- 운반 시에는 충분히 주의하시고 제품을 던지지 마십시오. 이런 행동은 제품의 오작동을 유발합니다.
- 드라이버를 취급할 장소에서는 불연물을 사용해 주십시오.
- 드라이버의 주변에 폭발성 물질이나, 인화성 가스가 있는 장소, 부식성 분위기, 물이 닿을 가능성이 있는 장소, 가연성 물질이 있는 부근에서는 사용을 피하여 주십시오.
- 드라어버의 정격 (전압 또는 전력) 이내의 범위에서 사용하시기 바랍니다.
- 드라이버의 전원 입력 전압이 OFF 되어 있는 것을 확인한 후 작업하여 주십시오.
- 드라이버의 배선 접속은 배선도에 따라 정확하게 실시하여 주십시오.
- 드라이버에 전원을 투입할 때에는 드라이버의 제어 입력을 모두 OFF로 한 후에 투입하여 주십시오.
- 드라이버를 비 정상적으로 전원을 차단 하였을 경우, 다시 전원 인가 시 전체적인 설정 상태를 재 확인 하시기 바랍니다. 갑작스런 모터 기동으로 인한 상해의 위험이 있습니다.
- 드라이버의 보호 기능이 작동하면 원인을 제거한 후 보호 기능을 해제하여 주십시오.
- 모든 배선장치의 연결 또는 해제 시 전원이 꺼져 있는지 확인 하시기 바랍니다.
- 통전 중에 배선 변경을 하지 마십시오.
- 보조 제동장치가 설치되지 않은 조건에서 수직 상하 운동(Z축)에 사용하지 마십시오.
- 여러대의 드라이버를 작고 좁은 밀폐된 공간에 설치할 경우, 냉각팬 등을 설치하여 주위 온도를 50°C 이하가 되도록 하십시오.

1 전원 연결시 주의사항 1

1.1 전원 스위치 및 비상 정비 버튼의 사용 1

1.2 파워서플라이 사용시 주의사항 1

1.3 배터리 충전시 주의사항 2

1.4 전기 노이즈 감소 방법 2

2 모터 사양 및 크기 3

2.1 BLDC 시리즈 3

2.1.1 BLDC 모터 시리즈 명명법 3

2.1.2 BLDC 모터 사양 3

2.1.3 BLDC 모터 크기 4

2.1.4 BLDC 모터 & 엔코더 배선 4

2.1.5 BLDC 모터 토크 특성 5

3 드라이버 주요 특징 6

3.1 드라이버 주요 특징 6

4 드라이버 사양 및 크기 7

4.1 드라이버 사양 7

4.2 드라이버 크기 8

5 드라이버 연결 및 배선 9

5.1 드라이버 연결시 주의사항 9

5.2 드라이버 외부 구성 10

5.3 드라이버 핀맵 구성 11

5.4 드라이버 외부 배선도 12

5.5 드라이버 외부 LED 13

5.6 엔코더 연결 구성 13

5.7 제어 신호 연결 구성 14

5.8 통신 연결 구성 14

6 시스템 구성 15

6.1 드라이버 & 모터 & 모션제어기 구성 15

6.2 드라이버 & 모터 & 액추에이터 구성 16

7 MOTOR CONTROL UI 17

7.1 소프트웨어 다운로드 및 실행 17

7.1.1 시스템 요구사항 17

7.1.2 다운로드 17

7.1.3 실행 17

7.2 메인 화면 구성 18

7.2.1 헤더 18

7.2.2 탭 윈도우 19

7.2.3 메시지 표시 20

7.3 Chart Recorder 창 20

7.4 연결 22

7.4.1 Connection 22

7.5 Motor Control Tab 23

7.5.1 Motor Control Status 그룹 23

7.5.2 Motor Fault Flags그룹 24

7.5.3 Motor Drive그룹 24

7.5.4 Jog그룹 26

7.5.5 Motion Test그룹 26

7.5.6 Power Stage Monitoring그룹 26

7.6 Configuration Tab 28

7.6.1 모터 드라이버 오토 튜닝 30

7.6.2 세부 튜닝 38
7.6.3 Product Information 39
7.6.4 Start-Up 40
7.6.5 Serial Connection 40
7.6.6 Position Sensor 41
7.6.7 Motor & Mechanics 41
7.6.8 Motor & Mechanics – Motion Profile 42
7.6.9 Motor & Mechanics – Electrical Parameters 43
7.6.10 Motor & Mechanics – Mechanical Parameters 43
7.6.11 Controller – Fault Protection – Fault Limits 43
7.6.12 Controller – Fault Protection – Detection 44
7.6.13 Controller – Feedback Controller 44
7.6.14 Controller – Status & I/O Parameters 45

7.7 Firmware Update 47

8 통신 프로토콜 50

8.1 Communication Protocol 50
8.1.1 용어의 정리 50

8.2 CAN 메시지 51
8.2.1 CAN 패킷의 기본 구조 52
8.2.2 오브젝트 읽기 요청 53
8.2.3 오브젝트 쓰기 요청 53
8.2.4 오브젝트 읽기/쓰기 요청에 대한 성공 응답 54
8.2.5 오브젝트 읽기/쓰기 요청에 대한 실패 응답 54

8.3 시리얼 바이너리 패킷 55
8.3.1 바이너리 패킷의 기본 구조 55
8.3.2 오브젝트 읽기 요청 56
8.3.3 오브젝트 쓰기 요청 56
8.3.4 오브젝트 읽기/쓰기 요청에 대한 성공 응답 56
8.3.5 오브젝트 읽기/쓰기 요청에 대한 실패 응답 56

8.4 시리얼 텍스트 패킷 57
8.4.1 오브젝트 읽기 요청 57

8.4.2 오브젝트 쓰기 요청 58
8.4.3 오브젝트 읽기/쓰기 요청에 대한 성공 응답 58
8.4.4 오브젝트 읽기/쓰기 요청에 대한 실패 응답 58
8.4.5 Device ID의 부여 59
8.4.6 여러 오브젝트의 읽기/쓰기 59

9 오브젝트 테이블 61

9.1 Object Table 61

9.2 Product Information 64
9.2.1 vendor_id 64
9.2.2 product_id 64
9.2.3 software_version 64
9.2.4 hardware_version 65

9.3 System Parameters 65
9.3.1 system_status 65
9.3.2 system_control 66
9.3.3 system_argument 68
9.3.4 power_source_voltage 69
9.3.5 power_source_current 69
9.3.6 units 69

9.4 Connection 70
9.4.1 device_name 70
9.4.2 device_id 70
9.4.3 serial_watchdog 71
9.4.4 serial_baudrate 71

9.5 Startup 71
9.5.1 startup_control_mode 71

9.6 Pulse/Direction Input 72
9.6.1 pulse_input_mode 72
9.6.2 pulse_count_edge 72
9.6.3 dir_signal_polarity 73

9.6.4 enable_signal_level 73
9.6.5 gear_numerator 73
9.6.6 gear_denominator 73

9.7 Error Code 73
9.7.1 error_code 73

9.8 Motor Status 77
9.8.1 control 77
9.8.2 status 78
9.8.3 fault 79
9.8.4 temperature 80

9.9 Motor Drive 80
9.9.1 target_position 80
9.9.2 target_velocity 80
9.9.3 target_current 81
9.9.4 target_voltage 81
9.9.5 torque_limit 81
9.9.6 position_demand 82
9.9.7 velocity_demand 82
9.9.8 current_demand 82
9.9.9 voltage_demand 82
9.9.10 position_actual 82
9.9.11 velocity_actual 82
9.9.12 current_actual 83
9.9.13 position_error 83
9.9.14 velocity_error 83
9.9.15 current_error 83

9.10 Position Limit 83
9.10.1 soft_limit_check 84
9.10.2 min_position 84
9.10.3 max_position 84

9.11 Rated Values 84
9.11.1 rated_voltage 84

9.11.2 rated_current 84
9.11.3 rated_velocity 85
9.11.4 max_current 85
9.11.5 max_velocity 85

9.12 Velocity Profile 85
9.12.1 profile_type 85
9.12.2 profile_velocity 86
9.12.3 profile_acceleration 86
9.12.4 profile_deceleration 88
9.12.5 profile_jerk 88

9.13 Fault Limits 88
9.13.1 overheat_limit 88
9.13.2 overcurrent_limit 88
9.13.3 overvoltage_limit 88
9.13.4 under0voltage_limit 89

9.14 Fault Detection 89
9.14.1 vibration_detect 89
9.14.2 stall_current_detect 89
9.14.3 position_track_error 90
9.14.4 velocity_track_error 90

9.15 Motor Properties 91
9.15.1 motor_type 91

9.16 Sensor Properties 91
9.16.1 position_sensor 91
9.16.2 encoder_direction 92
9.16.3 encoder_resolution 92
9.16.4 no_pole_pairs 92
9.16.5 hallsensor_phase 94

9.17 Electrical Parameters 94
9.17.1 bemf_constant 94
9.17.2 resistance 94

9.17.3 inductance 95

9.18 Mechanical Parameters 95
9.18.1 inertia_ratio 95
9.18.2 moment_of_inertia 95
9.18.3 viscous_friction 95
9.18.4 coulomb_friction 95

9.19 Controller's Gain 96
9.19.1 position_p_gain 96
9.19.2 velocity_p_gain 96
9.19.3 velocity_i_gain 96
9.19.4 current_p_gain 96
9.19.5 current_i_gain 97

9.20 Moving Average Filter 97
9.20.1 pmaf_window_size 97
9.20.2 vmaf_window_size 97

9.21 I/O Parameters 98
9.21.1 in_position_threshold 98
9.21.2 in_velocity_threshold 98
9.21.3 regen_voltage_clamp 99

10 관련 자료 99

1 전원 연결시 주의사항

드라이버에 전원(Power Source, 배터리 또는 파워서플라이)을 연결하는 방법에 대해 설명합니다.

1.1 전원 스위치 및 비상 정비 버튼의 사용

파워서플라이와 같이 전원 스위치가 없는 배터리를 전원으로 사용할 때는, 드라이버의 전원을 ON/OFF 할 수 있는 전원 스위치의 연결을 권장합니다. 또는, 비상시 전원을 차단하기 위한 비상 정지 스위치(Emergency Disconnect Switch)의 연결을 권장합니다.



그림 1-1 비상 정지 버튼

전원 스위치나 비상 정지 스위치 양단에는 1KΩ, 0.5W 저항을 연결하기를 권장합니다. 이 저항은 스위치가 켜질 때 스위치 내부에서 발생하는 전기적 아크(electric arc)를 방지하는 역할을 합니다.

모터가 고속으로 회전하는 동안에는 전원 스위치를 ON/OFF하지 마십시오. 드라이버가 손상될 수도 있으니 주의하시기 바랍니다.

1.2 파워서플라이 사용시 주의사항

스위칭 방식의 파워서플라이(Switching power supply)를 사용할 경우, 모터에서 발전되는 전력에 의해 전원으로 전류가 역류해서 손상을 일으킬 수 있기 때문에 주의해야 하며, 다음과 같은 보호 단계들이 고려되어야 합니다:

1. 출력 전압보다 높은 전압이 역으로 가해질 때에도 파손되지 않는 전원공급장치를 사용해야 합니다.
2. 전원공급장치의 출력과 병렬로 배터리를 연결합니다. 배터리는 회생 전류를 담을 수 있는 저수조 역할을 합니다. 배터리를 처음 연결할 때는 완전히 충전된 배터리를 사용해야 하며, 전원공급장치로 전류의 역류를 방지하기 위해 출력 단에 디커플링 다이오드를 연결합니다

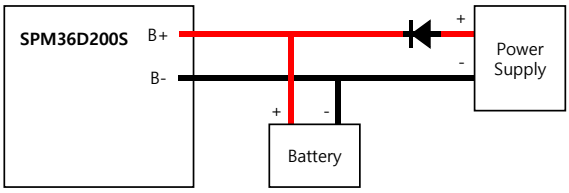


그림 1-2 파워서플라이와 배터리 병렬 연결

1.3 배터리 충전시 주의사항

충전식 배터리를 사용할 경우 드라이버 및 전원 보호 회로를 외부적으로 따로 설계한 후, 전원 스위치를 끈 상태에서 배터리를 충전해야 합니다. 만약 드라이버와 연결된 상태에서 충전할 경우, 드라이버 및 주변 회로에 배터리 전압 이상의 전압이 인가될 수 있습니다.

1.4 전기 노이즈 감소 방법

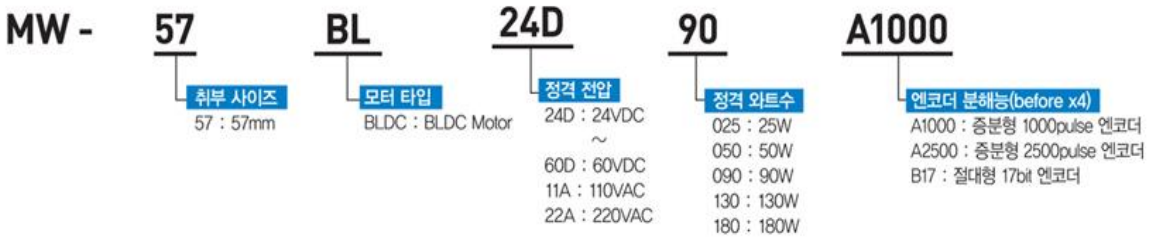
전기적 노이즈를 감소시키기 위해 드라이버 내에 여러 회로들이 설계되어 있지만, 드라이버를 모터와 전원에 연결하여 사용할 때는 노이즈를 저감하기 위한 추가적인 작업이 필요합니다. 다음은 전기적 노이즈를 감소시키는 방법들입니다:

- 전선은 가능한 짧게
- 전선을 페라이트 코어(Ferrite cores)에 감기
- 모터 단자에 스너버(Snubber) RC 회로 추가
- 드라이버와 전선, 배터리를 외부와 접촉이 없는 금속 프레임에 설치

2 모터 사양 및 크기

2.1 BLDC 시리즈

2.1.1 BLDC 모터 시리즈 명명법



2.1.2 BLDC 모터 사양

- **전기적 사양**

Parameters	57BL24D025	57BL24D050	57BL36D090	57BL36D130	57BL36D180	57BL36D180-C
Rated Voltage (VDC)	24	24	36	36	36	36
Rated Power (W)	25	50	90	130	180	180
Rated Tprque (N.M)	0.08	0.16	0.29	0.41	0.57	0.43
Peak Torque (N.M)	0.24	0.48	0.87	1.23	1.71	1.27
Rated Speed (RPM)	3000	3000	3000	3000	3000	4000
Rated Current (A)	1.6	3	3.45	5.3	6.7	7
Peak Current (A)	4.8	9	10.35	15.9	20	20.5
Torque Const. (N.M/A)	0.05	0.053	0.084	0.078	0.085	0.063
Back EMF Const (V/RPM)	5.2	5.55	8.8	8.2	8.9	6.6
Resistance (ohms)	1.73	0.88	1.35	0.63	0.9	0.35
Inductance (mH)	3.36	2.2	4.1	2.17	2	1
Inertia ($Kgm^2 \times 10^{-4}$)	30	75	119	173	230	230
Motor Length L (mm)	70	80	100	120	140	140
Mass (Kg)	0.25	0.5	0.75	1	1.25	1.25

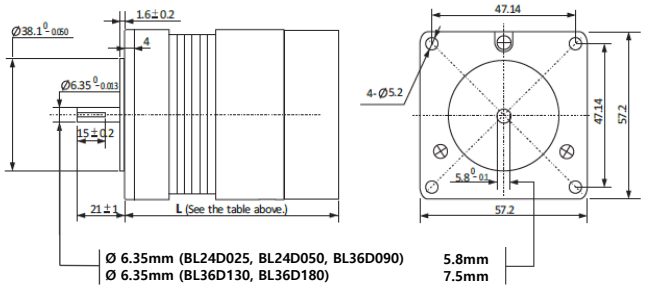
- **일반적 사양**

Parameters	
Winding connection	△ Delta connection
Hall sensor	120°
Pole Pairs	2
Phase	3
Shaft Radial Play	0.025 mm
Shaft Axial Play	0.025mm@460g
Allowable radial load	75N@20mm from th flange
Allowable axial load	15N
Isolation Level	Class B
Isolation Strength	500 VDC for one minute
Isolation Resistance	100 M ohms

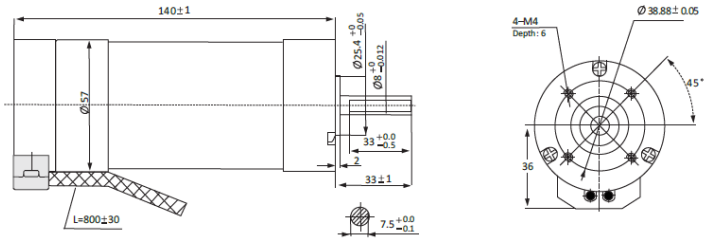
2.1.3 BLDC 모터 크기

Parameters	
Frame Size	57mm (NEMA23)

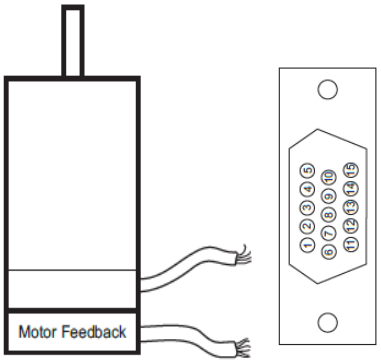
- **57BL24D025, 57BL24D050, 57BL36D090, 57BL36D130, 57BL36D180**



- **57BL36D180-C**



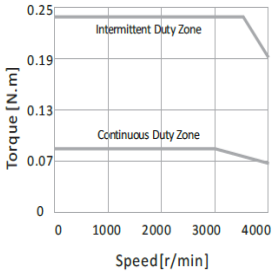
2.1.4 BLDC 모터 & 엔코더 배선



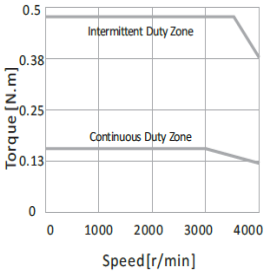
pin	Description	pin	Description	pin	Description
1	EA+	6	NC	11	EA-
2	EB+	7	NC	12	EB-
3	GND	8	NC	13	VCC
4	Hall C	9	Hall B	14	NC
5	Hall A	10	NC	15	NC

- **Motor Cable :** U (BRN), V (BLU), W (BLK)
- **Hall Sensor Cable :** Hall A (BRN), Hall B (GRY), Hall C (ORG), VCC+5V (RED), GND (BLK)
- **Encoder Cable :** EA+ (GRN), EA- (GRN/BLK), EB+ (WHT), EB- (WHT/BLK), VCC+5V (ORN/BLK), GND (BRN/BLK)

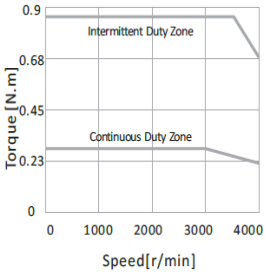
2.1.5 BLDC 모터 토크 특성



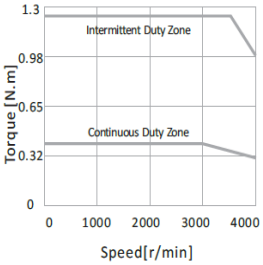
MW-57BL24D025



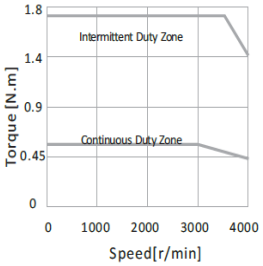
MW-57BL24D050



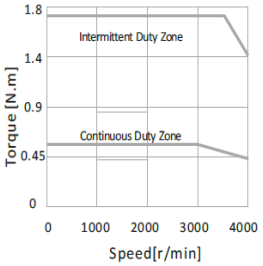
MW-57BL36D090



MW-57BL36D130



MW-57BL36D180



MW-57BL36D180-C

3 드라이버 주요 특징

3.1 드라이버 주요 특징

- DC/BLDC/PMSM 모터 제어용 드라이버
- PC 기반의 User Interface를 사용하여 제어 파라미터 설정 및 구동
- 시리얼 명령어를 통한 위치, 속도, 전류, 전압 지령
- 사다리꼴과 Sinusoidal 프로파일을 사용한 위치&속도 구동
- 위치, 속도, 전류에 대한 PID 피드백 위치 제어기 사용
- 모터의 전기적 파라미터 (L, R, Ke) 탐지 및 전류제어기 이득 자동 설정 기능 탑재
- 모터의 기계적 파라미터 (J, B, TL) 탐지 및 속도&위치 제어기 이득 자동 설정 기능 탑재
- 모터의 전기적 & 기계적 모델로부터 Feed Forward 제어 기능 사용
- FOC (자속 기반 제어) 방식 제어를 통한 저소음, 저리플, 고효율 구동
- 펄스 입력 방식과 USB 통신 방식에 의한 모터 제어
- 1펄스 (Pulse, Direction) 방식과, 2펄스 (CW, CCW) 방식 중 선택 구동
- 모터의 단락, 과전압, 저전압, 과온도 등 상태 감지에 의한 알람 발생 기능



4 드라이버 사양 및 크기

4.1 드라이버 사양

전기적 데이터	
정격 전압	+24VDC
최소 전압	+9 VDC
최대 전압	+30 VDC
연속 전류	10 A
최대 전류	20 A
정격 출력	200W
모터 타입	DC/BLDC/PMSM
PWM 구동 주파수	20 KHz
제어 방법	자속 기반 제어 (FOC)
전류제어 샘플링 주파수	10 KHz
속도제어 샘플링 주파수	1 KHz
위치제어 샘플링 주파수	1 KHz

전압/전류 출력	
Encoder	+5V, 120 mA
Hall Sensor	+5V, 150 mA

Position Sensor Feedback	
Encoder Type	Incremental Encoder, Differential Input(26LS32)
Encoder 최대 입력 주파수	10 Mpps
Hall Sensor Type	Line Driver or Open-Collector (U, V, W)

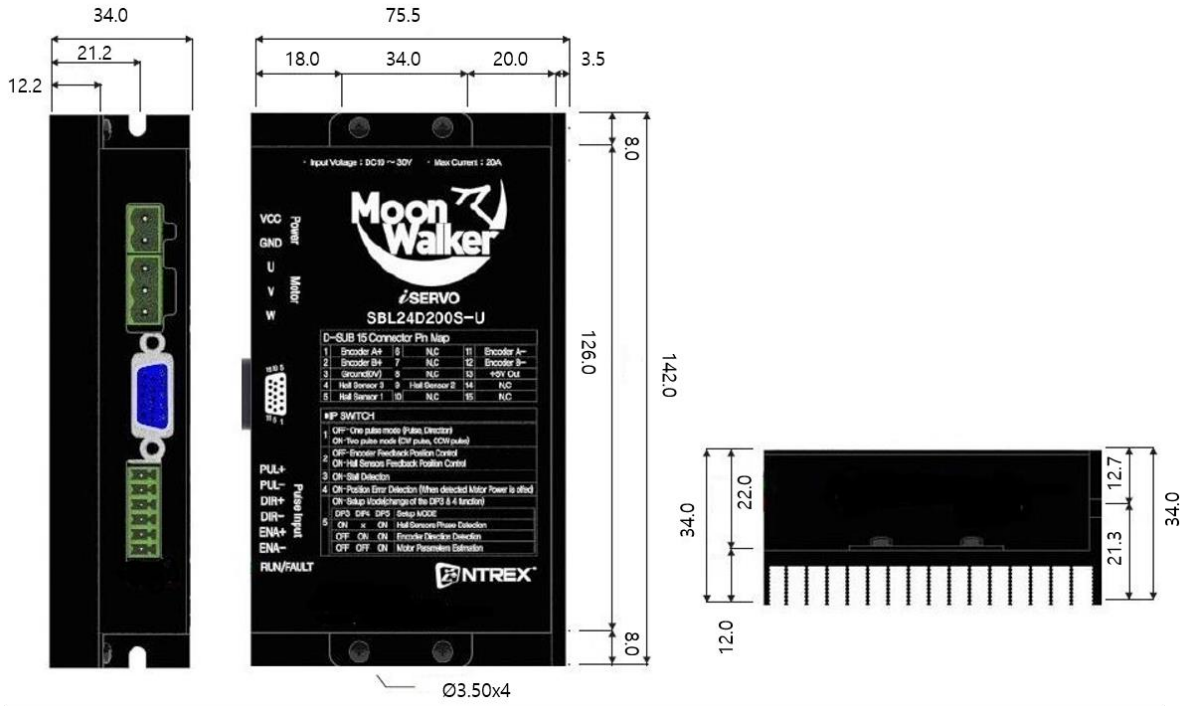
Communication	
USB	PC 와의 접속 – UI 운용

상태 표시	
BLUE LED	Running
RED LED	Error / Fault
GREEN LED	PC Communication

환경 조건	
보관 온도	0 ~ 50 ℃
습도	40 ~ 90 % RH
방열	Heat Sink Plane

4.2 드라이버 크기

기계적 데이터	
길이 (mm)	142
너비 (mm)	75.5
높이 (mm)	35
중량 (g)	320



5 드라이버 연결 및 배선

5.1 드라이버 연결시 주의사항

- 실내에서 사용해 주시기 바랍니다.
- 실내 주위 온도는 0~50 ℃에서 사용해 주시기 바랍니다. (여러대의 드라이버를 작고 좁은 밀폐된 공간에 설치할 경우, 냉각팬 등을 설치하여 주십시오.)
- 드라이버의 온도가 50 ℃ 이상이 되면 방열을 해주시기 바랍니다.
- 주변에 폭발성 물질이나, 인화성 가스가 있는 장소, 부식성 분위기, 물이 닿을 가능성이 있는 장소, 가연성 물질이 있는 부근에서는 사용을 피하여 주시기 바랍니다.
- 통전 중에 절대로 배선 변경을 하지 마십시오.
- 배선의 연결 또는 해제 시 전원이 꺼져 있는지 확인해주시기 바랍니다.
- 드라이버를 여러대를 나란히 설치 시에는 수직 방향으로 20mm, 수평 방향으로 50mm 이상 거리를 두고 설치해 주시기 바랍니다.

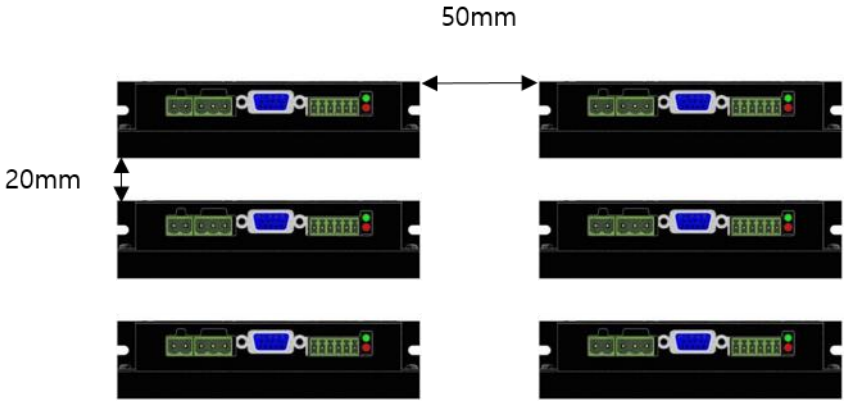
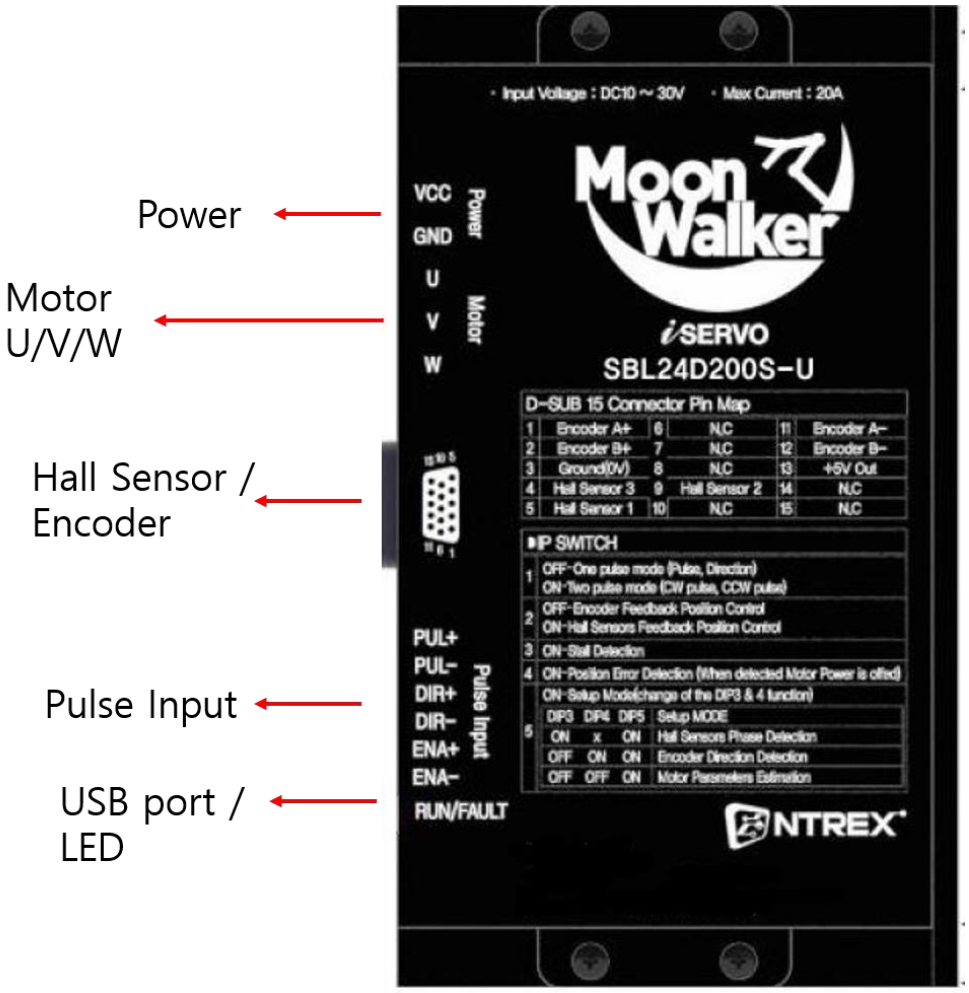
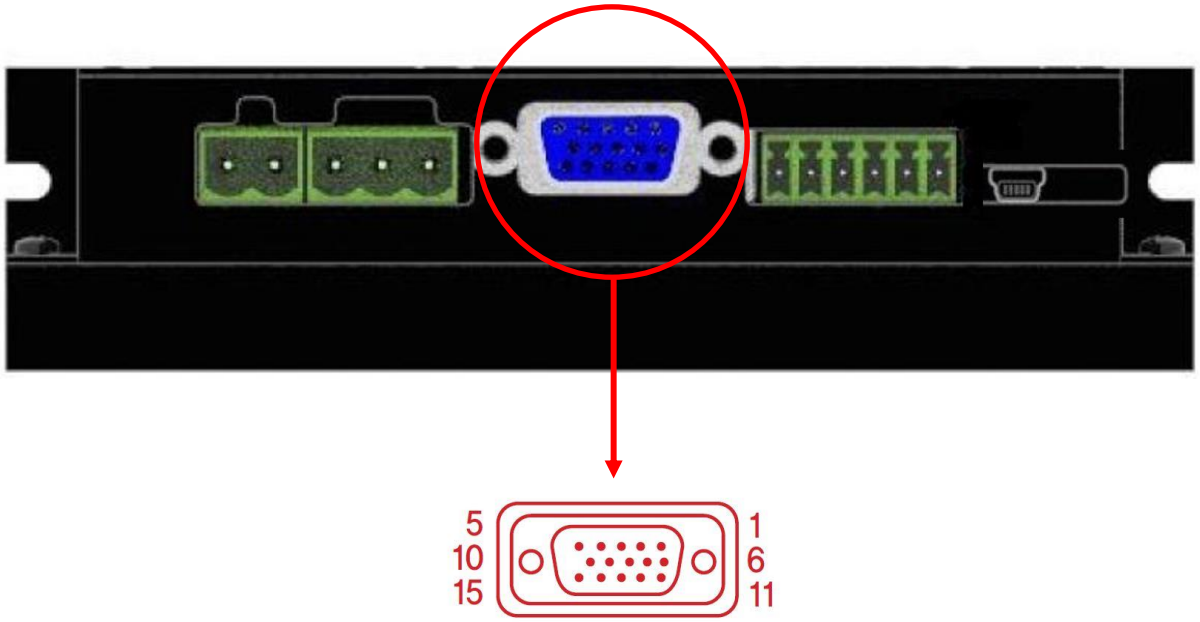


그림 5-1 드라이버 여러대 설치 시 간격

5.2 드라이버 외부 구성


Power
Motor U/V/W
Hall Sensor / Encoder
Pulse Input
USB port / LED
· Input Voltage : DC10 ~ 30V
· Max Current : 20A
Moon Walker
iSERVO
SBL24D200S-U
VCC
GND
Power
U
V
W
Motor
PUL+
PUL-
DIR+
DIR-
ENA+
ENA-
Pulse Input
RUN/FAULT
D-SUB 15 Connector Pin Map
1 Encoder A+ 6 N.C 11 Encoder A-
2 Encoder B+ 7 N.C 12 Encoder B-
3 Ground(0V) 8 N.C 13 +5V Out
4 Hall Sensor 3 9 Hall Sensor 2 14 N.C
5 Hall Sensor 1 10 N.C 15 N.C
DIP SWITCH
1 OFF-One pulse mode (Pulse, Direction)
ON-Two pulse mode (CW pulse, CCW pulse)
2 OFF-Encoder Feedback Position Control
ON-Hall Sensors Feedback Position Control
3 ON-Stall Detection
4 ON-Position Error Detection (When detected Motor Power is offed)
5 ON-Setup Mode(change of the DIP3 & 4 function)
DIP3 DIP4 DIP5 Setup MODE
ON x ON Hall Sensors Phase Detection
OFF ON ON Encoder Direction Detection
OFF OFF ON Motor Parameters Estimation
NTREX

5.3 드라이버 핀맵 구성



1	Encoder Positive A	Encoder A+ / Positive A 상 입력
2	Encoder Positive B	Encoder B+ / Positive B 상 입력
3	GND	Ground / 엔코더 전원 그라운드
4	Hall Sensor W	Hall Sensor3 / W 상 입력
5	Hall Sensor V	Hall Sensor2 / V 상 입력
6	N.C	Not Connect
7		Not Connect
8		Not Connect
9	Hall Sensor U	Hall Sensor1 / U 상 입력
10	N.C	Not Connect
11	Encoder Negative A	Encoder A- / Negative A 상 입력
12	Encoder Negative B	Encoder B- / Negative B 상 입력
13	+5V	+5V / 엔코더 전원 그라운드 / 최대 150mA
14	N.C	Not Connect
15		Not Connect

5.4 드라이버 외부 배선도

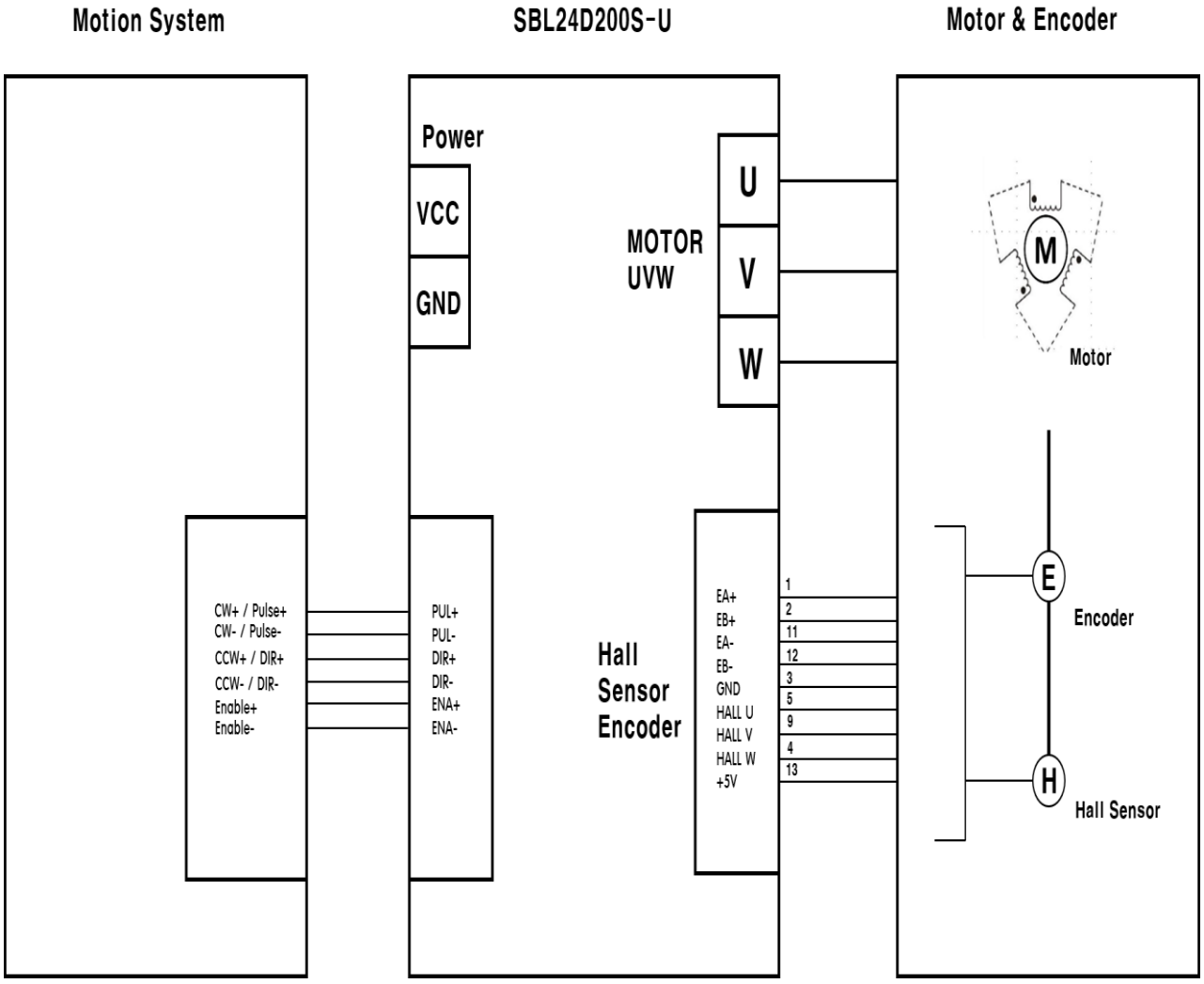


그림 5-2 SBL24D200S-U 외부 배선도

5.5 드라이버 외부 LED

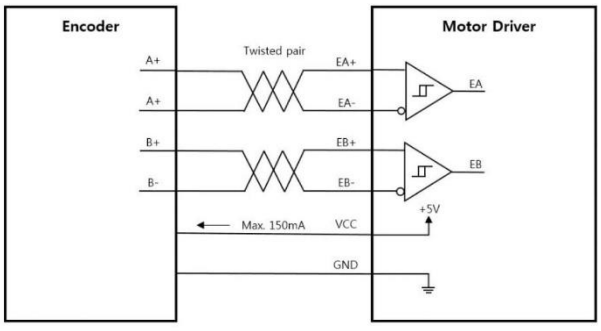
제어기는 동작 상태를 표시하는 청색, 적색, 녹색의 LED를 가지고 있습니다. 각각의 LED가 표시하는 상태는 다음 표10 에서 정리하고 있습니다.

제어기에 전원이 투입되면 3개의 LED가 동시에 0.5초 동안 켜졌다가 꺼지며, 이후 동작상태를 표시하는 청색 LED만 깜박이게 됩니다. 이는 정상적으로 제어기가 켜진 상태를 표시하는 것입니다. 제어기에 전원을 투입 후, 3개의 LED가 모두 꺼져있거나 켜져 있는 경우는 모터제어기가 정상적으로 초기화되지 않은 상태입니다. 만일 모터제어기가 정상적으로 초기화 되었다면 동작상태를 표시하는 LED는 항상 1초 주기로 깜박이게 됩니다.

3개의 LED가 모두 0.25초 주기로 깜박이는 경우는 제어기의 하드웨어나 소프트웨어가 오작동을 일으켜 소프트웨어 실행이 중단된 상황으로 제어기를 하드웨어적으로 리셋 하거나 전원을 껐다 켜야 합니다. 근본적으로 제어기의 소프트웨어나 하드웨어에 문제가 있는 것을 의미하며 개발자에게 상황을 리포트 하여 소프트웨어나 하드웨어 수준에서 문제를 수정해야 합니다.

Status	Pattern	Description
BLUE LED (Run)	0000000000	모터에 전력이 공급되지 않는(Power OFF) 상태이고 폴트가 발생한 상태
	1000000000	모터에 전력이 공급되지 않는 상태이고 폴트가 발생하지 않은 상태
	1111100000	모터에 전력이 공급되고(Power ON) 동작 가능한 상태
	1010101010	모터에 전력이 공급되고 모터가 구동 중인 상태
	1110011100	모터에 전력이 공급되고 모터가 목적 위치에 도달한 상태
RED LED (Fault)	0000000000	Fault off – 정상 동작 중
	1000000000	Fault on – Overcurrent error
	1010000000	Fault on – Undervoltage or Overvoltage error
	1010100000	Fault on – Over temperature error
	1010101000	Fault on – Short circuit error (내부 회로 단락 또는 쇼트)
	1010101010	Fault on – Stall detection, 과대 위치 오차 감지, 홀센서 단선 감지, 잘못된 역기전력 상수 입력
GREEN LED (Comm)	0	통신하고 있지 않음
	1	통신하고 있음

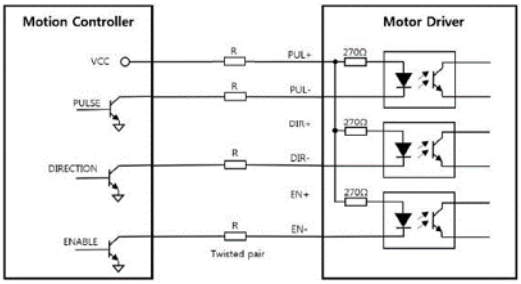
5.6 엔코더 연결 구성



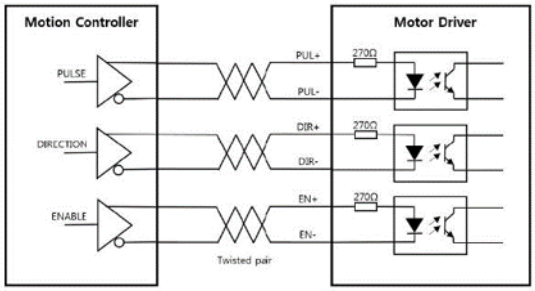
엔코더의 신호는 라인드라이버 방식의 차동 신호를 입력받는 형태입니다. 따라서 꼬임선 형태, 그리고 외부 노이즈로 부터의 쉴드 기능이 갖추어진 통신선(RS-422)의 사용을 권장 합니다. 엔코더 전원공급을 위한 모터 드라이버 측의 최대 출력전류는 150mA입니다. 엔코더의 수요전류가 150mA이상일 경우 외부적인 보조전원장치(+5V)를 연결하시기 바랍니다.

5.7 제어 신호 연결 구성

SBL24D200S-U의 제어신호는 Pulse, Direction, Enable 총 3개의 입력 라인이 제공되며, 외부 노이즈로부터의 보호를 위해 모두 전기적 절연이되어 있습니다. 아래의 예제는 두 가지의 입력방법을 그림을 통하여 설명하고 있습니다. 제어신호 라인은 10mA에서 20mA사이의 전류가 흐를 수 있도록 설계 되어야 합니다. 그리고 각 신호라인마다 270Ω의 저항이 내장 되어 있습니다. 최대 전류가 20mA가 넘지 않도록 외부적으로 저항값을 적절히 조절해야 합니다.



●그림 5-7.1 Open-Collector 연결의 경우 (common-anode)결구성



●그림 5-7.2 Line Driver 연결의 경우 (Differential Input)

5.8 통신 연결 구성

사용자가 PC를 사용하여 제어기의 구성(Configuration)을 설정하고 운용 하는 가장 간단한 방법은 제어기와 PC 간에 USB 연결을 구성하고 PC에서 Motor Control UI 유틸리티를 사용하는 것입니다. 제어기의 Mini-USB(B type) 단자에 USB 케이블을 연결합니다. 그러면 PC는 새로운 하드웨어를 감지하고 USB 장치 드라이버를 설치합니다. 그리고 PC에 가상 시리얼 포트(VCP)가 자동으로 설치됩니다. 정상적으로 설치가 끝나면 PC에서 Motor Control UI 유틸리티를 실행하여 제어기를 설정하고 운용 할 수 있습니다.

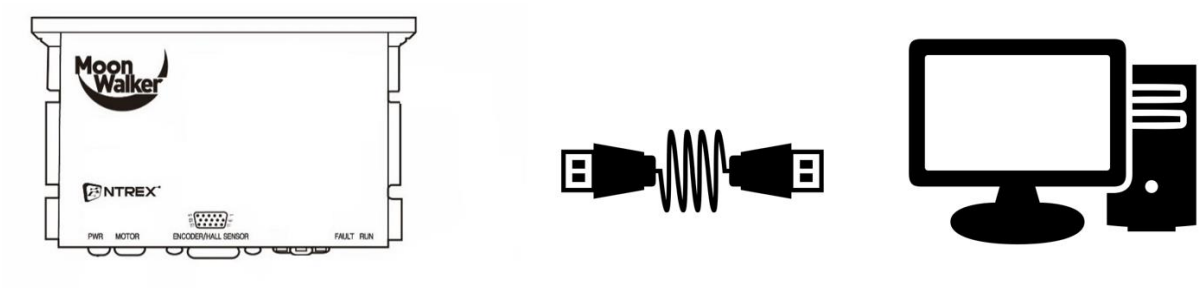


그림 5-3 SBL24D200S-U의 USB 연결

6 시스템 구성

6.1 드라이버 & 모터 & 모션제어기 구성

SBL24D200S-U 드라이버와 BLDC 모터 그리고 모션제어기를 아래 그림과 같이 연결 후 바로 사용이 가능합니다.

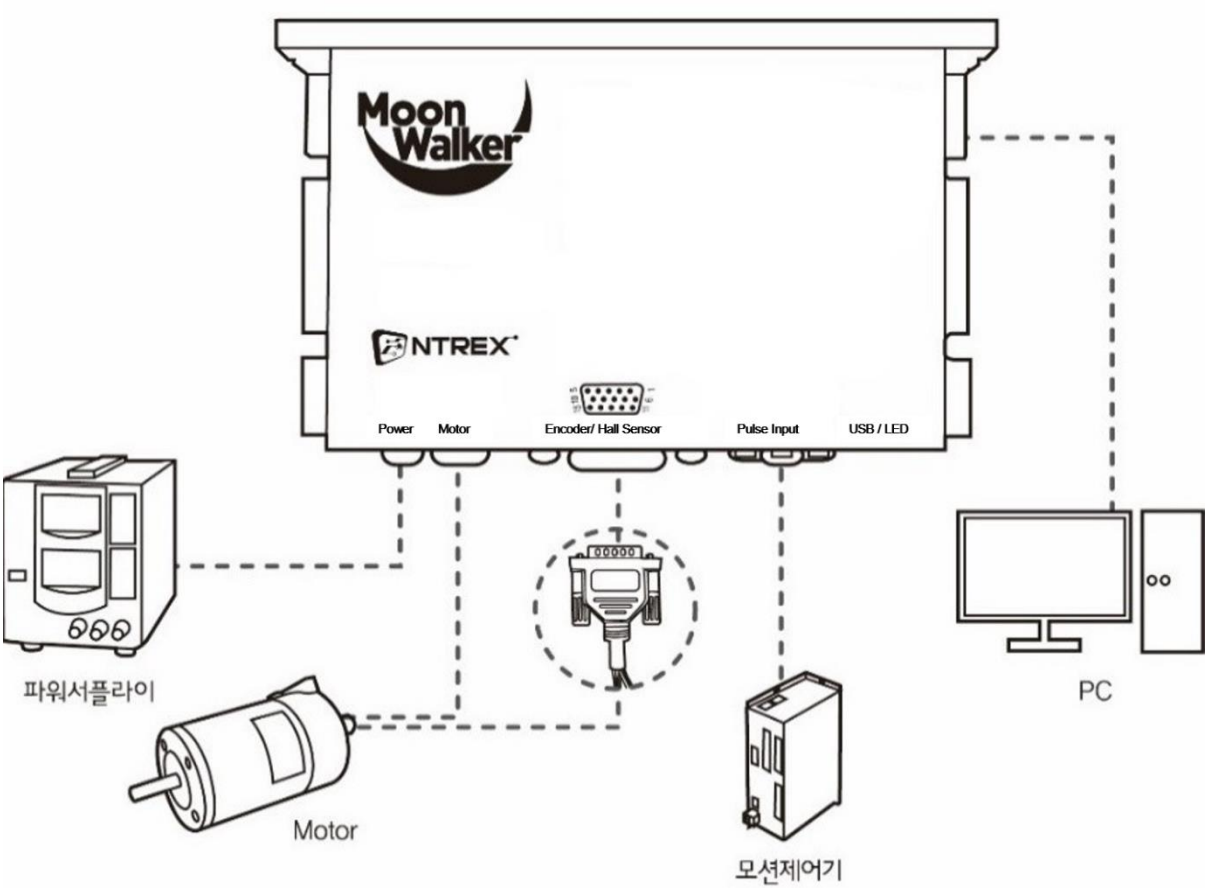


그림 6-1 드라이버 & 모터 & 모션제어기 구성의 예

6.2 드라이버 & 모터 & 액추에이터 구성

SBL24D200S-U 드라이버와 DC/BLDC/PMSM모터, 그리고 액추에이터를 가지고 아래 그림과 같이 연결 후 2 & 3축 직교로봇을 구성할 수 있습니다.

[액추에이터]



W40	NEMA23
W45	NEMA23
W50	NEMA23 / NEMA24
W60	NEMA23 / NEMA24

[모터]



BLDC	57BL24D025
BLDC	57BL24D050
BLDC	57BL36D090
BLDC	57BL36D130
BLDC	57BL36D180

[드라이버]



SBL24D200S-U









그림 6-2 드라이버 & 모터 & 액추에이터 구성의 예

7 Motor Control UI

7.1 소프트웨어 다운로드 및 실행

7.1.1 시스템 요구사항

이 유틸리티를 실행하기 위해서는 다음과 같은 PC 환경이 필요합니다.

- Window XP/7/10 32bit/64bit OS
- 10Mbyte의 HDD 여유공간
- 1GByte 이상의 RAM
- USB포트

***주의: Motor Control UI 유틸리티는 제품에 동봉되어 있지 않습니다. 이 유틸리티를 다운로드 하기 위해서 PC는 인터넷과 연결되어 있어야 합니다.**

7.1.2 다운로드

UI 유틸리티 프로그램은 디바이스마트내 제품 관련자료에서 다운받을 수 있습니다.

http://www.devicezine.com/

7.1.3 실행

Motor Control UI 유틸리티는 설치 과정이 필요없습니다. 압축을 푼 폴더 안에 있는 MotorControlUI.exe 파일을 실행하면 됩니다. 이 유틸리티의 첫 실행 화면은 그림 7-1에서 보여주고 있습니다.

이 유틸리티를 올바르게 사용하려면 모터 제어기가 PC에 연결되어 있어야 합니다. 그렇지 않으면 유틸리티의 모든 기능이 활성화 되지 않습니다.

※ Motor Control UI 유틸리티는 수시로 업데이트 될 수 있습니다. 따라서 사용자는 최신 버전을 확인하고 사용하기 바랍니다.



그림 7-1 SBL24D200S-U의 Motor Control UI 메인화면

7.2 메인 화면 구성

Motor Conrol UI 프로램의 메인 화면은 헤더와 탭 윈도우 부분으로 구성됩니다.

7.2.1 헤더

헤더에는 UI 유틸리티를 모터 드라이버에 연결하거나 펌웨어를 다운로드 할 수 있는 버튼들로 구성됩니다.



헤더의 제일 왼쪽에는 ㈜엔티렉스 상호와 MoonWalker 상표명, 프로그램 이름 (MW iServo UI)과 버전이 표시됩니다. 그리고 우측에 표시되는 버튼들은 다음과 같습니다.

- [Connection] 버튼 – 모터 드라이버와 연결을 위한 대화 상자가 표시
- [Firmware Update] 버튼 – 모터 드라이버에 펌웨어 다운로드를 진행하는 대화상자 표시
- [Chart] 버튼 – Chart Recorder 창를 화면에 보이거나 숨기도록 토글 함, 현재 창이 표시되지 않는 상태에서 버튼을 누르면 창이 표시되고, 반대로 창이 표시되는 상태에서 버튼을 누르면 창을 숨김
- [About] 버튼 – 프로그램에 대한 정보를 보여주는 대화상자 표시
- [Help] 버튼 – 도움말 표시
- [Emergency STOP] 버튼 – 비상 정지 버튼으로, 모터 드라이버를 Disable 상태로 만들고 모터는 Freen run 상태가 됨

7.2.2 탭 윈도우

탭 윈도우는 Motor Control, Configurations의 2개 탭으로 구성되어 있습니다.



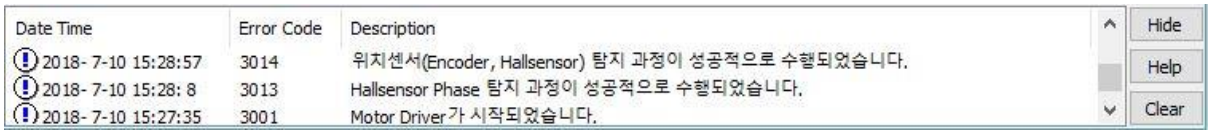
다음은 각각의 탭에 대한 간략한 설명입니다.

- Motor Control Tab – 모터 드라이버에 모터 제어 명령을 내리고 상태를 모니터링 함.
- Configurations Tab – 모터 드라이버의 각종 파라메터를 설정 함.

사용자는 필요에 따라 해당 탭을 선택하여 UI 의 기능을 사용할수 있습니다.

7.2.3 메시지 표시

메시지 표시부에는 모터 드라이버에서 발생한 information, Warning, Error 메시지가 순서대로 표시됩니다.



우측의 버튼들은 다음의 기능을 수행합니다.

- [Hide] 버튼 – 메시지 표시부를 접어 보이지 않게 함, 메시지가 발생하면 메시지 표시부가 열리게 됨
- [Help] 버튼 – 에러 메시지에 대한 설명과 조치 방법에 대한 문서를 오픈
- [Clear] 버튼 – 모터 드라이버에서 발생한 모든 메시지를 삭제

7.3 Chart Recorder 창

Chart Recorder 창은 모터에 내려지는 위치, 속도, 전류, 전압 명령과 센서 측정 그리고 입출력 채널 중 선택된 항목의 값을 실시간 그래프로 모니터링 할 수 있습니다.

이 창은 메인 화면의 헤더에서 [Chart] 버튼을 눌러 표시하거나 닫을 수 있습니다..



현재 출력되는 그래프를 파일로 저장하고자 한다면, [Save] 버튼을 눌러 파일 이름을 지정하고 *.csv 포멧으로 저장 가능합니다.

그래프의 왼쪽에 표시되는 스케일(Scale)은 **마우스 휠**을 위아래로 굴려 확대하거나 축소할 수 있습니다. 또한, 그래프의 아래에는 그래프로 표시되는 항목들에 대한 세부 정보가 표시됩니다.

[Channel] 버튼을 클릭하면 다음 그림7.3과 같이 Channel Selection 대화상자가 표시됩니다. 사용자는 최대 8개의 오브젝트를 선택할 수 있으며, 선택한 오브젝트의 Sub-index 값과 Scale, Color를 지정 가능합니다.

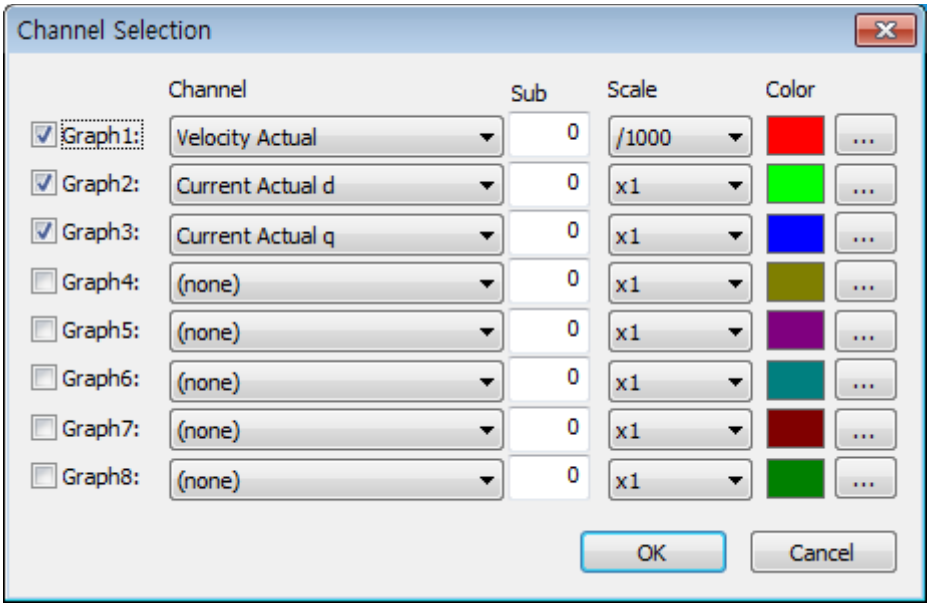


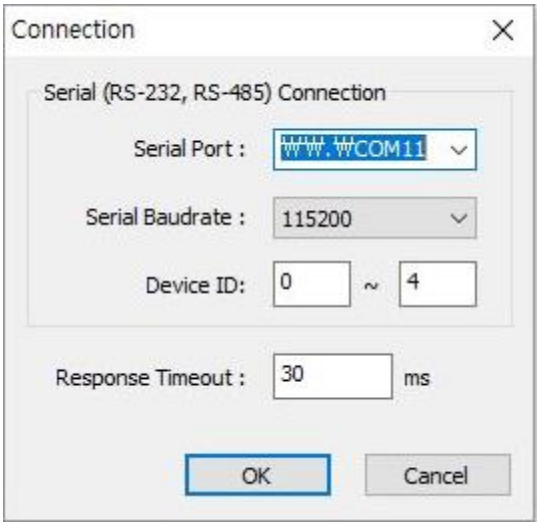
그림 7-3 Channel Selection 대화 상자

7.4 연결

7.4.1 Connection

메인 화면의 헤더에서 [Connection] 버튼을 클릭하면, UI 유틸리티를 모터 드라이버에 연결하기 위한 대화상자가 화면에 표시됩니다. 모터 드라이버는 종류에 따라 USB, RS-232, RS-485, Ethernet 연결을 제공합니다.

SBL24D200S-U 모터드라이버의 경우 USB포트로 통신을 하기 때문에 해당 UI와 통신이 가능합니다.



Serial Port에는 연결 가능한 USB, RS-232 또는 RS-485 통신 포트가 검색되어 표시됩니다. 표시되는 포트 이름에서 모터 드라이버가 연결된 통신 포트를 선택합니다. 만일 연결된 제어기의 가상 COM 포트가 표시되지 않는다면, COM 포트 번호를 직접 편집할 수 있습니다.

Serial Baudrate에서는 COM 포트의 통신 속도(9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 bits/s)를 선택합니다. 만일 'Auto detect'를 선택하면, 모든 통신 속도를 검색하여 모터 드라이버가 응답하는지 체크합니다.

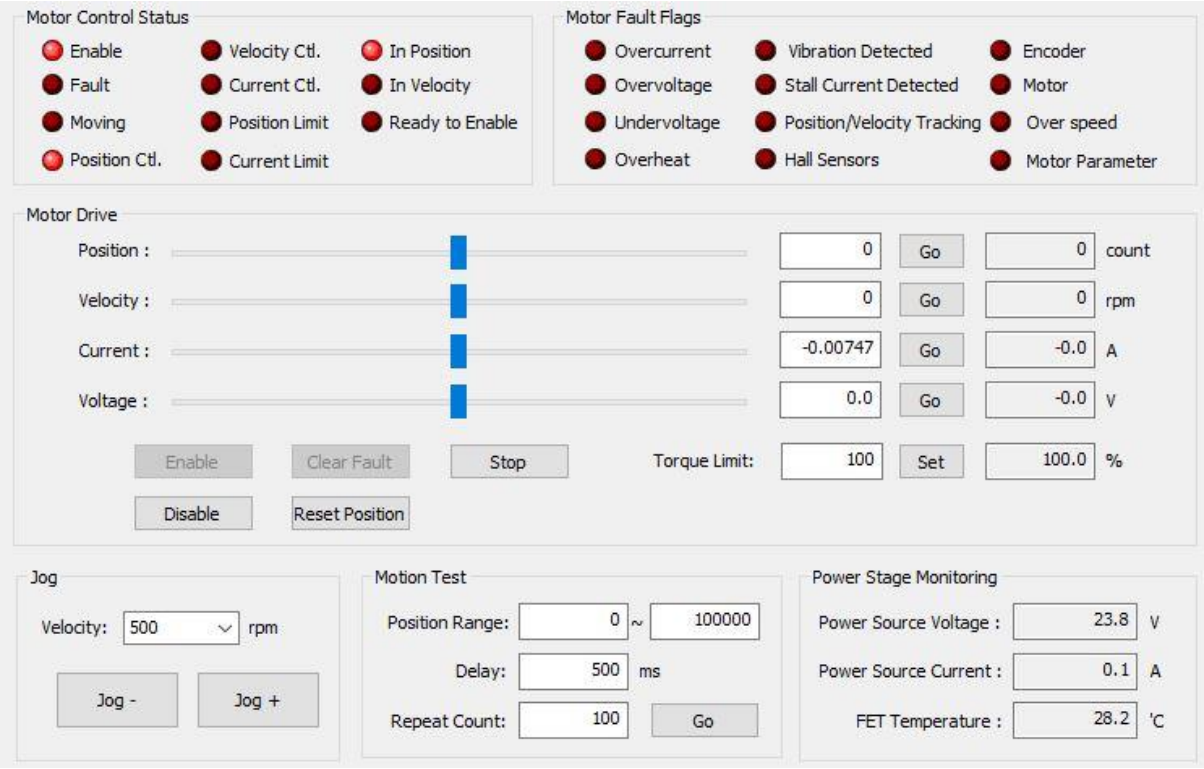
Device ID에서는 모터 드라이버 ID의 검색 범위를 설정합니다. 상기 그림에서와 같이 0에서 4 값으로 설정하면 모터 드라이버 ID를 0부터 4까지 검색하여 제일 먼저 응답하는 모터 드라이브와 연결하게 됩니다. 만일 검색 범위를 넓게 설정하게 되면 UI 프로그램은 많은 모터 드라이버 ID를 검색해야 하기 때문에 검색 시간이 오래 걸립니다.

Response Timeout에서는 제어기에 패킷을 보내고 되돌아올 때까지 기다리는 응답 시간을 밀리세컨드(ms)단위로 설정합니다.

연결과 관련된 정보 설정이 끝나면 [Connect] 버튼을 눌러 모터 드라이버와 연결을 시도합니다. 만일 [Cancel] 버튼을 누르면 연결을 시도하지 않고 않고 대화상자를 닫습니다.

7.5 Motor Control Tab

Motor Control 탭은 모터 드라이버의 상태를 표시하고, 모터의 위치, 속도, 전류, 전압 명령을 내리거나 현재 값들을 읽어올 수 있습니다.



Motor Control 탭에는 크게 Motor Control Status, Motor Fault Flags, Motor Drive, Jog, Motion test, Power Satge Monitoring 그룹으로 나누어 집니다.

7.5.1 Motor Control Status 그룹

Motor Control Status 그룹에서는 제어기의 모터 구동 상태 및 제어기의 I/O 상태 등을 확인 할 수 있습니다. 표시되는 상태를 정리하면 다음과 같습니다.

- Enable – 모터에 전원이 공급되고 구동 가능한 상태
- Fault – 모터에 폴트가 발생한 상황
- Moving – 모터가 회전하고 있는 상태
- In Position – 모터가 목적 위치에 도달하였음을 표시
- In Velocity – 모터가 목적 속도에 도달하였음을 표시
- Position Limit – 모터가 Low/High Limit 범위를 벗어남을 표시
- Current Limit – 모터에 흐르는 전류가 최대전류에 가깝게 흐르는 상태
- Position Ctl – 제어기가 위치 제어 상태일때를 표시
- Velocity Ctl – 제어기가 속도 제어 상태일때를 표시
- Current Ctl – 제어기가 전류 제어 상태일때를 표시

- Ready to Enable – 제어기가 Enable이 가능한 상태임을 표시

제어기의 모터 상태와 모터 폴트르 표시하는 LED는 그림 7.6과 같습니다.

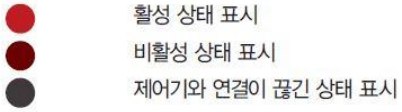


그림 7-6 UI 상태 LED

7.5.2 Motor Fault Flags 그룹

Motor Fault Flags는 모터에서 폴트가 발생한 세부 내역을 확인합니다. 모터의 폴트 플래그 목록은 다음과 같습니다. 제어기에서 Fault 가 발생하면 모터에 공급되는 전원은 차단(Power OFF)되며, Fault 조건이 해제된 이후에 모터는 Power ON 가능합니다.

- Overcurrent – 모터에 과전류가 흐름
- Overvoltage – 제어기에 과전압이 걸림
- Undervoltage – 제어기에 저전압 감지
- Overheat – 내부 FET와 방열판이 과열
- Vibration Detected – 모터의 전류가 설정 주기 이상으로 진동
- Stall Current Detected – 모터에 전력이 공급되지만, 회전하지 못하고 있는 상태 감지
- Position/Velocity Tracking – 위치/속도 제어모드에서 위치/속도 오차가 지정된 값 이상 감지
- Hall Sensors – 홀센서의 결선이나 입력 신호가 오류
- Encoder – 엔코더의 결선이나 입력 신호가 오류
- Motor – 모터의 결선이나 연결이 오류
- Over speed – 모터의 속도가 최대 속도의 150%를 초과
- Motor Parameter – 제어기가 역기전력(B-EMF)을 잘못 계산

7.5.3 Motor Drive 그룹

Motor Drive그룹에서는 Position, Velocity, Current, Voltage에 대한 슬라이드 바를 움직이거나 값을 직접 입력하고 [Go] 버튼을 누름으로 모터에 위치, 속도, 전류, 전압 명령을 내릴수 있습니다. 그리고 모터의 현재 위치, 속도, 전압과 전류 값을 표시합니다.



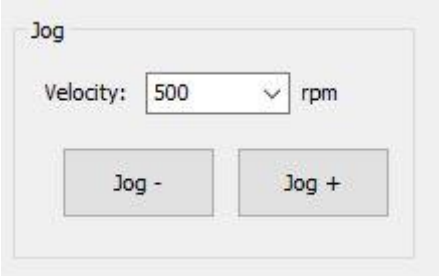
- Position – 목표 위치를 지정하고 [Go] 버튼을 눌러 목표 위치로 이동 명령을 내림, 슬라이드 바의 이동 범위는 min_position과 max_position에 의해 결정
- Velocity – 목표 속도를 지정하고 [Go] 버튼을 눌러 목표 속도로 이동 명령을 내림, 슬라이드 바의 이동 범위는 –rated_velocity와 +rated_velocity에 의해 결정
- Current – d축과 q축에 대한 목표 전류를 지정하고 [Go] 버튼을 눌러 목표 전류로 구동 명령을 내림, 슬라이드 바로는 q축 목표 전류를 조절하며 슬라이드 바의 이동 범위는 –rated_current와 +rated_current에 의해 결정
- Voltage – d축과 q축에 대한 출력 전압를 지정하고 [Go] 버튼을 눌러 전압 구동 명령을 내림, 슬라이드 바로는 q축 출력 전류를 조절하며 슬라이드 바의 이동 범위는 –rated_voltage와 +rated_voltage에 의해 결정

모터를 활성화 하거나 비활성화하고 구동을 정지하는 기능들은 다음과 같이 버튼들로 구성됩니다.

- [Enable] 버튼 – 모터가 Disable 상태일 때 Enable 상태로 만듦, Enable 상태에서는 모터에 전원이 공급되고 위치, 속도, 전류, 전압 명령에 의해 모터를 구동 가능
- [Disable] 버튼 – 모터가 Enable 상태일 때 Disable 상태로 만듦, Disable 상태에서는 모터에 전원이 차단되고 모터를 외력에 의해 구동 가능한 Free run 상태
- [Clear Fault] 버튼 – 모터 드라이브에 발생한 Fault 플래그를 모두 해제함, 만일 모든 Fault 플래그가 정상적으로 해제되었다면 Ready to Enable 상태
- [Reset Position] 버튼 – 모터의 현재 위치 값을 0으로 설정
- [Stop] 버튼 – 모터가 위치, 속도 명령에 의해 구동 중일 때는 모터의 속도가 0이 되도록 하거나, 모터가 전류, 전압 명령에 의해 구동 중일 때는 모터에 출력 되는 전압이 0이 됨

7.5.4 Jog 그룹

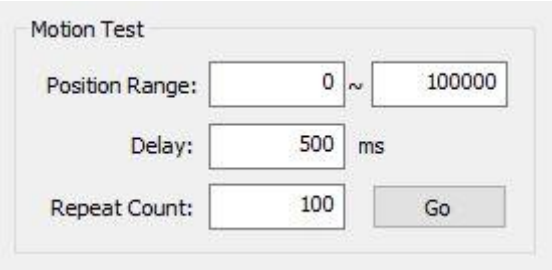
Jog 그룹은 조그 명령으로 모터를 정방향 / 역방향 구동합니다.



- Velocity – 조그 이동 속도를 설정
- [Jog -] 버튼 – 조그 이동 속도에 설정된 속도로 모터를 역방향 구동
- [Jog +] 버튼 – 조그 이동 속도에 설정된 속도로 모터를 정방향 구동

7.5.5 Motion Test 그룹

Motion Test 그룹에서는 모터가 지정된 두 위치 사이를 반복적으로 이동하도록 명령을 생성합니다. 이 명령은 UI에서 생성되기 때문에 정밀한 명령 생성 시간이 보장되지 않습니다.

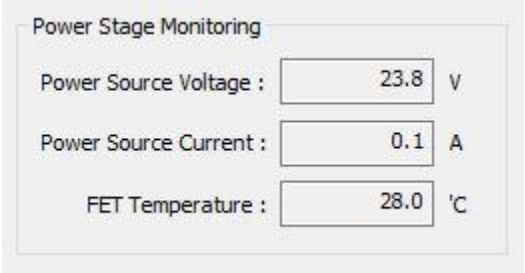


- Position Range – 모터가 왕복 이동하는 두 개의 목표 위치를 설정
- Delay – 모터가 목표 위치에 도달 후 멈춰 대기하는 시간 설정
- Repeat Count – 반복 횟수 설정

위 3가지 항목을 설정하고 [Go] 버튼을 눌러 모터의 구동을 시작합니다. 모터가 설정된 반복 횟수만큼 이동을 완료하면 정지합니다.

7.5.6 Power Stage Monitoring 그룹

Power Stage Monitoring 그룹에서는 모터에 공급되는 전력을 조절하는 파워 스테이지에서의 각종 측정 값을 표시합니다.



- Power Source Voltage – 제어기에 공급되는 전원의 전압 (VDC, GND 단자간 전압)
- Power Source Current – 제어기에 흐르는 전류 (모터 전류와 구분)
- FET Temperature – FET와 방열판의 온도

제어기는 전원에서 공급하는 전류가 아닌 모터를 통해 흐르는 전류를 측정하고 제어합니다. 일반적으로 모터를 통해 흐르는 전류는 전원 전류보다 높습니다. 이러한 현상은 모터의 인덕턴스에 의한 플라이백(Flyback) 전류 때문이며, 전원 전류가 낮게 흐르더라도 모터 전류는 극단적으로 높게 흘러 제어기에 잠재적인 손상을 가져올 수도 있습니다.

다음 그림과 같이 모터 전류는 플라이백 효과로 인해 PWM 스위칭에 의해 FET가 OFF 되었을 때에도 ON 되었을 때와 같은 수준의 전류 양이 연속으로 흐릅니다.

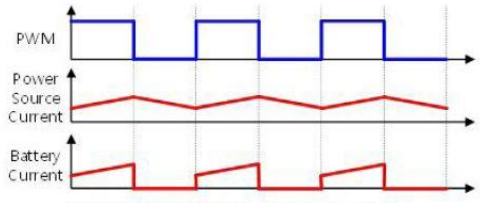


그림 17. SDC24D200S-U 의 Motor 상전류

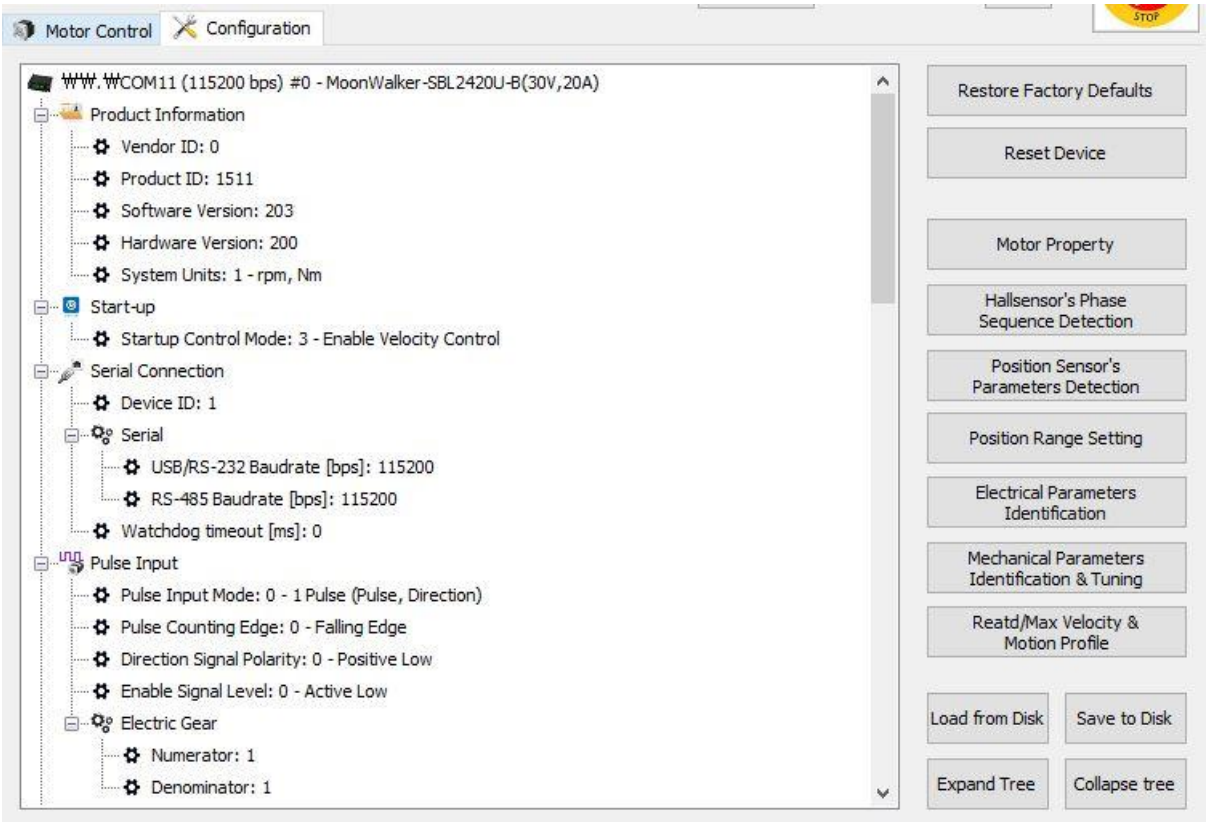
전원 전류(Power Source Current)와 모터 전류(Motor Current) 사이의 관계식은 다음과 같습니다:

$$PowerSourceCurrent = PWM_DutyRatio \times MotorCurrent$$

만약 제어기가 20% PWM 듀티비(Duty Ratio)에서 전원 전류가 1A로 측정되면 모터에 흐르는 전류는 5A가 될 것으로 예상할 수 있습니다.

7.6 Configuration Tab

Configurations 탭은 모터 드라이버의 각종 파라미터를 설정할 수 있습니다. 설정 값들은 모터 드라이버의 플래쉬 메모리에 저장되고, Factory default 값으로 초기화 할 수 있으며, PC의 파일로 저장하고 불러올 수도 있습니다.



UI가 모터 드라이버에 연결되면, 모터 드라이버의 파라미터 값을 읽고 왼편 창에 표시합니다. 사용자가 파라미터의 값을 변경하면, 변경된 값은 즉시 모터 드라이버에 다운로드 되고 적용됩니다. 사용자가 오른쪽 버튼들을 이용해서 값을 바꿀시 톱니바퀴 색은 파란색으로 변하고, 사용자가 직접 값을 입력해서 바꾼 경우에는 톱니바퀴 색은 붉은색으로 변합니다.



※ 값을 변경하실 경우 톱니바퀴 색이 바뀌었는지 체크하시기 바랍니다.

***주의: 모터 드라이버에 다운로드 된 파라미터 값은 모터 드라이버에 즉시 적용되지만 플래시 메모리에 바로 저장되지는 않습니다. 플래시 메모리에 저장되기 위해서는 모터 드라이브가 구동을 멈춘 상태에서 5초 정도의 시간이 필요합니다. 즉, 5초 이상 모터를 구동하지 않는 조건에서 변경된 파라미터를 플래시 메모리에 기록하기 때문에, 모터 드라이버의 파라미터 값을 바꾼 경우에는 즉시 전원을 끄지 않도록 주의하셔야 합니다.**

탭의 오른쪽 버튼들에 대한 기능을 요약하면 다음과 같습니다

- [Reset Factory Defaults] 버튼 - 제품을 출시할 때의 설정 값으로 모든 파라미터 설정을 초기화
- [Reset Device] 버튼 - 모터 드라이버를 소프트웨어 리셋 함, 모터 드라이버 펌웨어가 재시작
- [Motor Property] –모터에 대한 정격 전압과 정격전류, 최대전류 값을 설정하는 대화상자를 실행
- [Hallsensor's Phase Sequence Detection] 버튼 – 홀센서의 배열 순서와 위상을 탐지하는 대화상자를 실행
- [Position Sensor's Parameters Detection] 버튼 – 모터에 연결된 위치 센서(Hallsensor, Encoder) 정보를 탐지하는 대화상자를 실행
- [Position Range Setting] 버튼 – 모터의 구동 범위를 설정하는 대화상자를 실행
- [Electrical Parameters Identification & Tuning] 버튼 – 모터의 전기 파라미터를 탐지하고 d축과 q축 전류 제어기 이득을 설정하는 대화상자를 실행
- [Mechanical Parameters Identification & Tuning] 버튼 – 모터의 기계 파라미터를 탐지하고 위치 제어기와 속도 제어기 이득을 설정하는 대화상자를 실행
- [Motion Profile & Feed-forward Options] 버튼 – 모션 프로파일과 피드 포워드 옵션을 선택하는 대화상자를 실행
- [Load from Disk] 버튼 - PC에서 구성 파라미터 파일 읽어 모터 드라이버로 다운로드
- [Save to Disk] 버튼 – 모터 드라이버에서 구성 파라미터를 읽어 PC의 파일로 기록
- [Expand Tree] 버튼 – 트리 메뉴를 모두 펼침
- [Collapse Tree] 버튼 – 트리 메뉴를 모두 접음

7.6.1 모터 드라이버 오토 튜닝

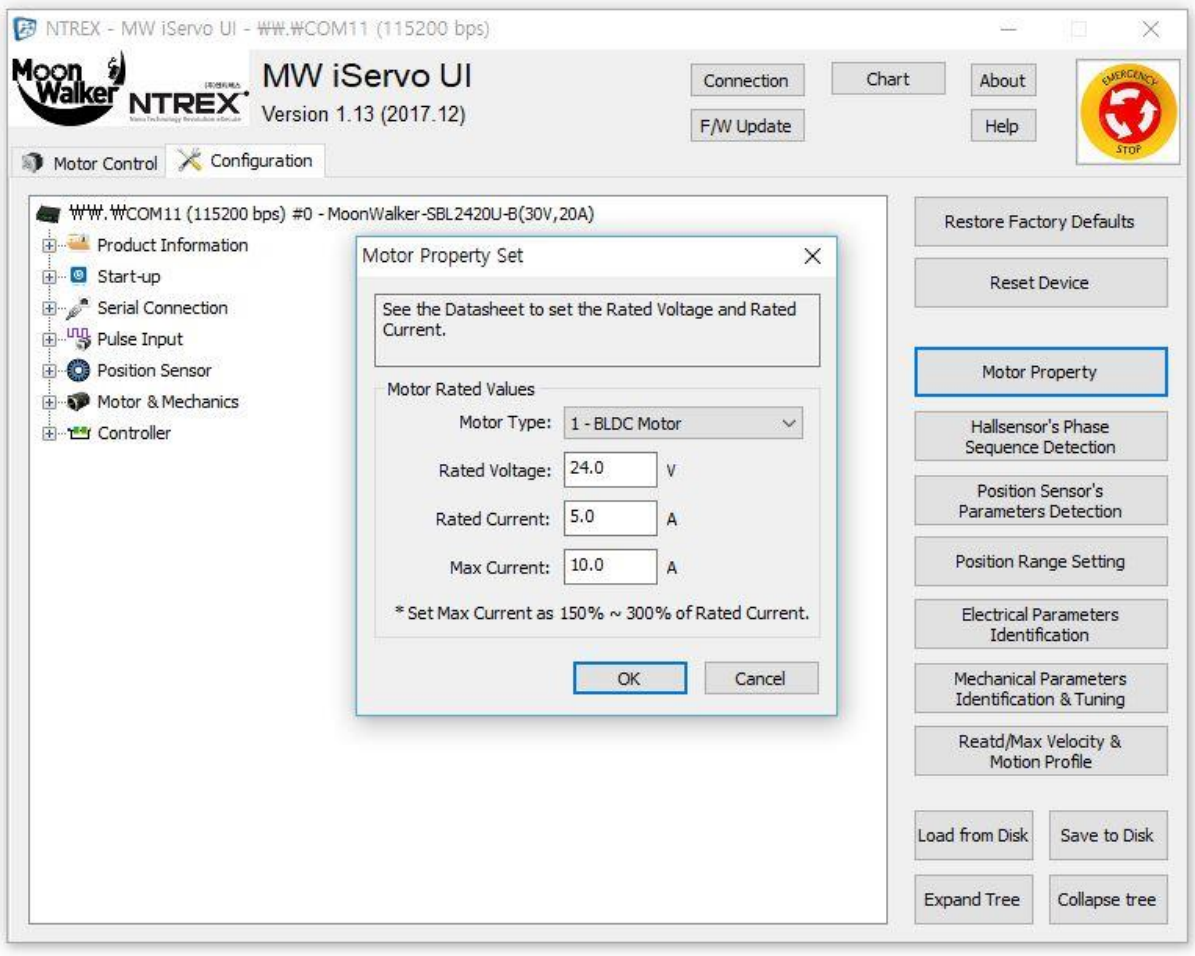
문워커 서보 모터드라이버를 사용하시려면 전원 – 모터드라이버 – 모터 연결을 끝낸 후 모터 드라이버 와 모터를 오토 튜닝 기능으로 매칭 시켜주어야 정상적으로 동작 합니다.

***주의: 오토튜닝 기능을 사용 하실 때는 모터가 무부하 상태로 있어야 합니다.**

***주의: Mechanical Parameters Identification & Tuning 버튼은 액추에이터등 기구부와 연결 한 후에 한번 더 설정하여 값을 맞춰야 합니다.**

7.6.1.1 Motor Property

[Motor Property] 버튼을 눌러 모터의 종류와 정격, 전압 및 전류를 설정 합니다.



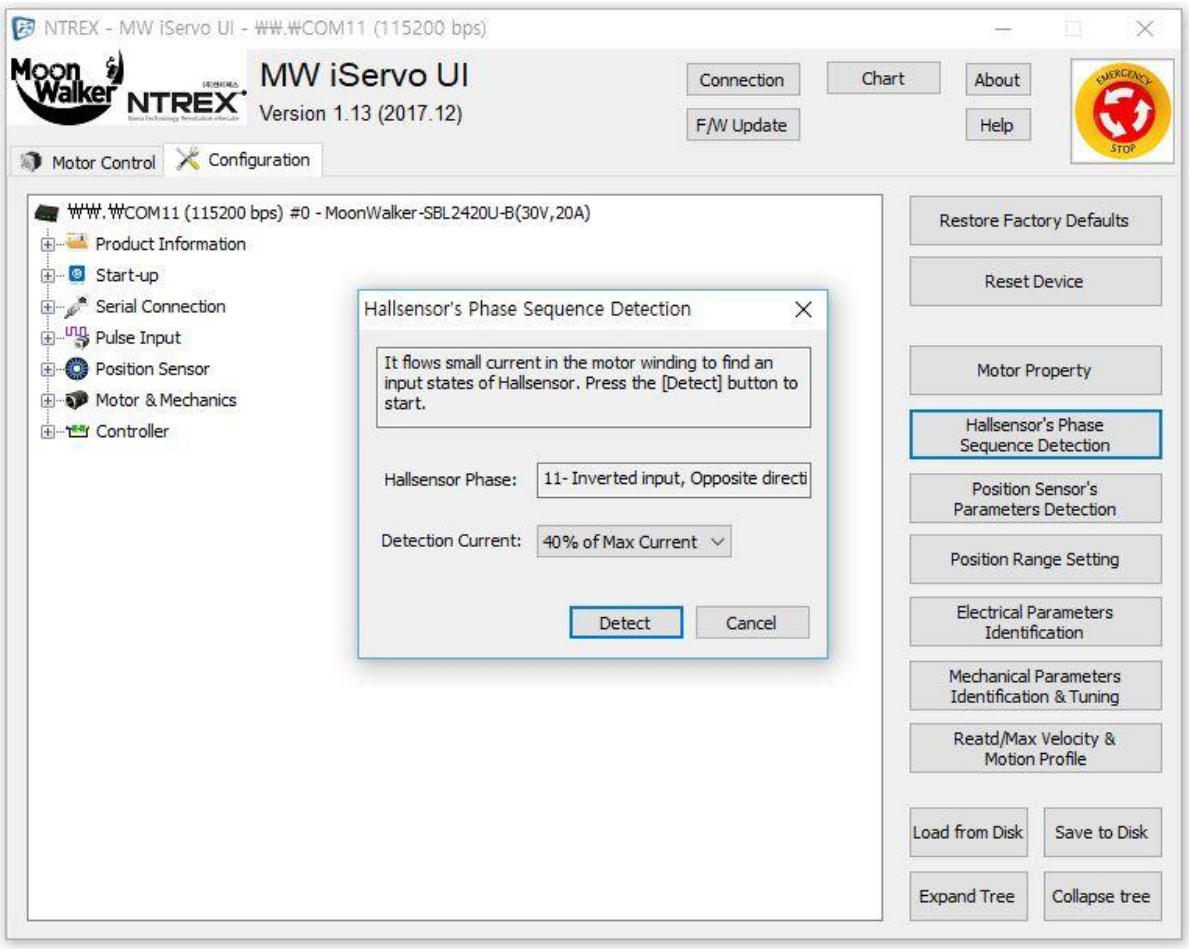
- Motor Type에서 모터의 종류를 선택
- Rated Voltage에서 정격 전압을 선택
- Rated Current에서 정격 전류를 선택
- Max Current에서 최대 전류를 선택

Rated Voltage에서는 모터에 가하는 정격 전압을, Rated Current에서는 모터에 연속으로 흘릴 수 있는 정격 전류(또는 데이터 시트의 continuous current)를 설정하고, Max Current(또는 데이터 시트의 peak current)는 보통 정격 전류의 300% 이내로 설정합니다.

※ Rated Current는 모터의 가감속시 모터에 평균적으로 흐르는 전류와 연관이 있습니다. 즉, 모터의 정격 전류 설정값에 의해 속도 프로파일의 가속도와 감속도가 결정됩니다.
※ Max Current는 속도 제어기가 생성하는 전류 명령의 최대치를 결정합니다. 모터에 흐르는 전류는 최대 전류 설정값을 넘어가지 않도록 제어되며, 모터가 멈춘 상태에서 일정 시간동안 최대 전류가 흐르면 Stall Current Fault를 발생합니다.

7.6.1.2 HallSensor's Phase Sequence Detection

[Hallsensor's Phase Sequence Detection] 버튼을 눌러 홀센서의 배열 순서를 찾습니다.



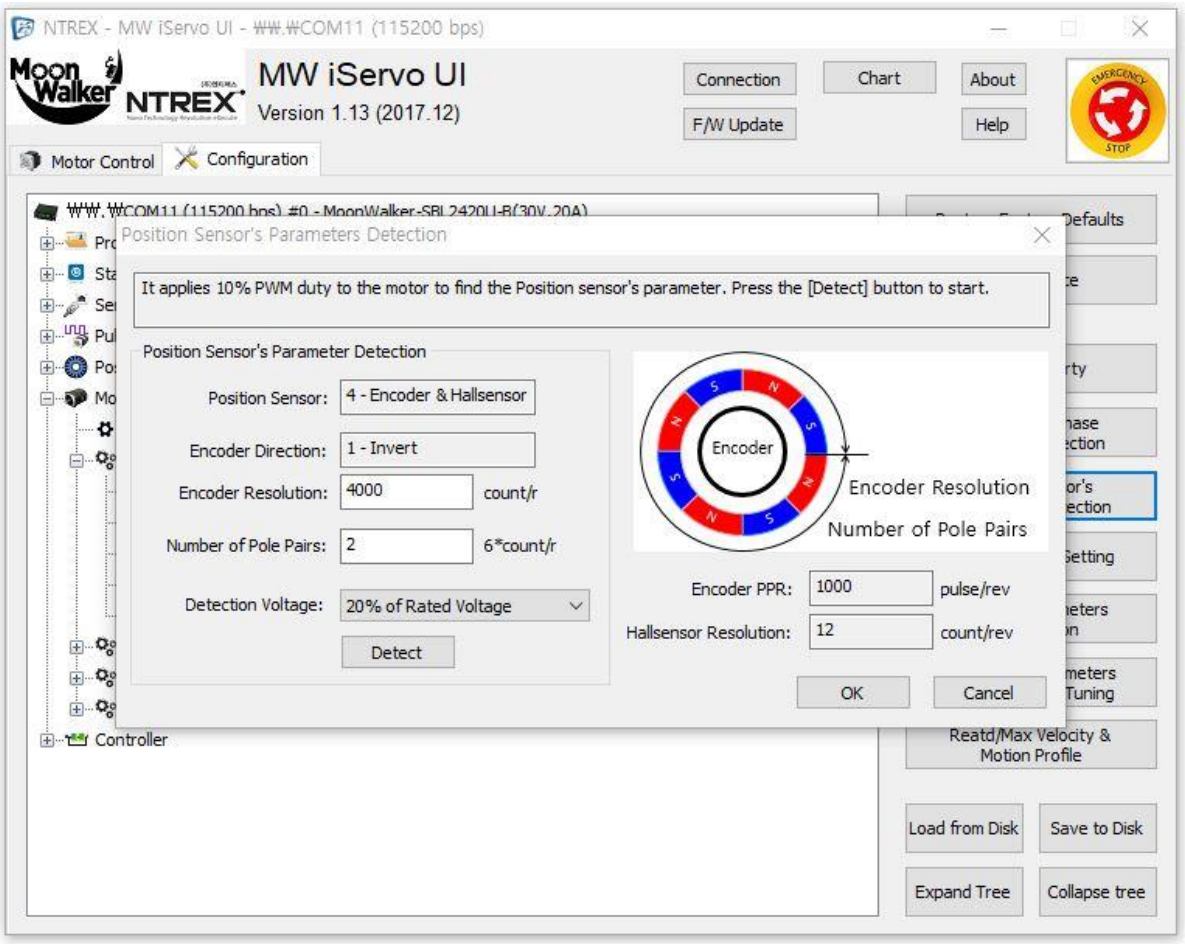
[Detect] 버튼을 누르면, 홀센서의 배열 순서를 찾기 위해 모터는 수 바퀴 회전합니다. 만일 모터가 일정한 속도로 회전하지 않거나 탐지에 실패하는 경우는 경우는 Detection Current를 높여서 다시 탐지 과정을 수행하면 됩니다.

***주의: DC 모터나 STEP 모터는 Hallsensor를 사용하지 않기 때문에 이 기능을 사용할 수 없습니**

다. BLDC모터와 PMSM 모터에서만 사용 가능하도록 버튼이 활성화 됩니다.

7.6.1.3 Position Sensor's Parameter Detection

[Position Sensor's Parameter Detection] 버튼을 눌러 Encoder와 Hallsensor 정보를 찾습니다



[Detect] 버튼을 누르면, Encoder와 Hallsensor 정보를 찾기 위해 모터는 수에서 수십 바퀴 회전하게 됩니다. 만일 모터가 일정한 속도로 회전하지 않거나 탐지에 실패하는 경우는 경우는 Detection Voltage를 높여 탐지 과정을 다시 수행하면 됩니다.

Encoder에서 Index 신호를 제공하는 경우는 Encoder Direction, Encoder Resolution, Number of Pole Pairs 값을 정확하게 찾습니다.

하지만 Index 신호를 사용할 수 없을 때는 Hallsensor 신호간의 균일하지 않은 시간차 데이터를 통계적으로 비교하여 Encoder Resolution, Number of Pole Pairs 값을 찾습니다. 이 경우에는 찾은 값의 비율은 맞을 수 있으나 실제 값과 다르게 찾을 수도 있습니다.

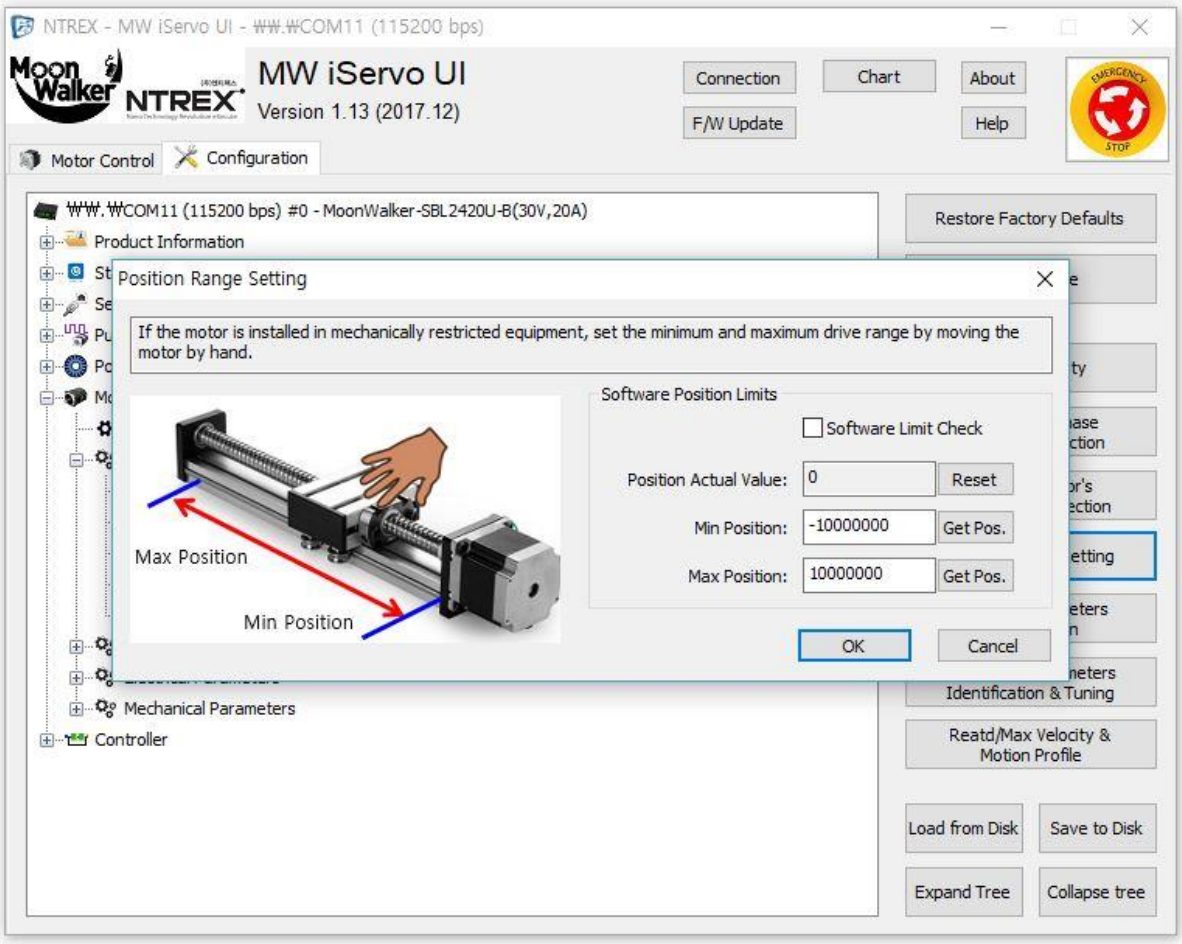
***주의: Encoder에 Index 신호가 없는 BLDC나 PMSM 모터의 경우, 통계적인 기법으로 Encoder와 Hallsensor 정보를 찾게 됩니다. 탐지 과정을 거쳐 찾은 값들은 모터의 데이터 시트를 참조하**

여 올바르게 설정되었는지 꼭 확인해야 한다. 만일 이 값들이 잘못 설정된 경우 모터의 회전 속도가 잘못 계산됩니다.

***주의: Index 신호가 없는 Encoder를 장착한 DC모터인 경우, 대략적인 Encoder Resolution을 추정합니다. 이 값은 모터의 데이터 시트를 참조하여 사용자가 올바르게 수정해야 합니다.**

7.6.1.4 Position Range Setting

[Position Range Setting] 버튼을 눌러 모터가 구동 가능한 최소/최대 범위를 설정합니다.



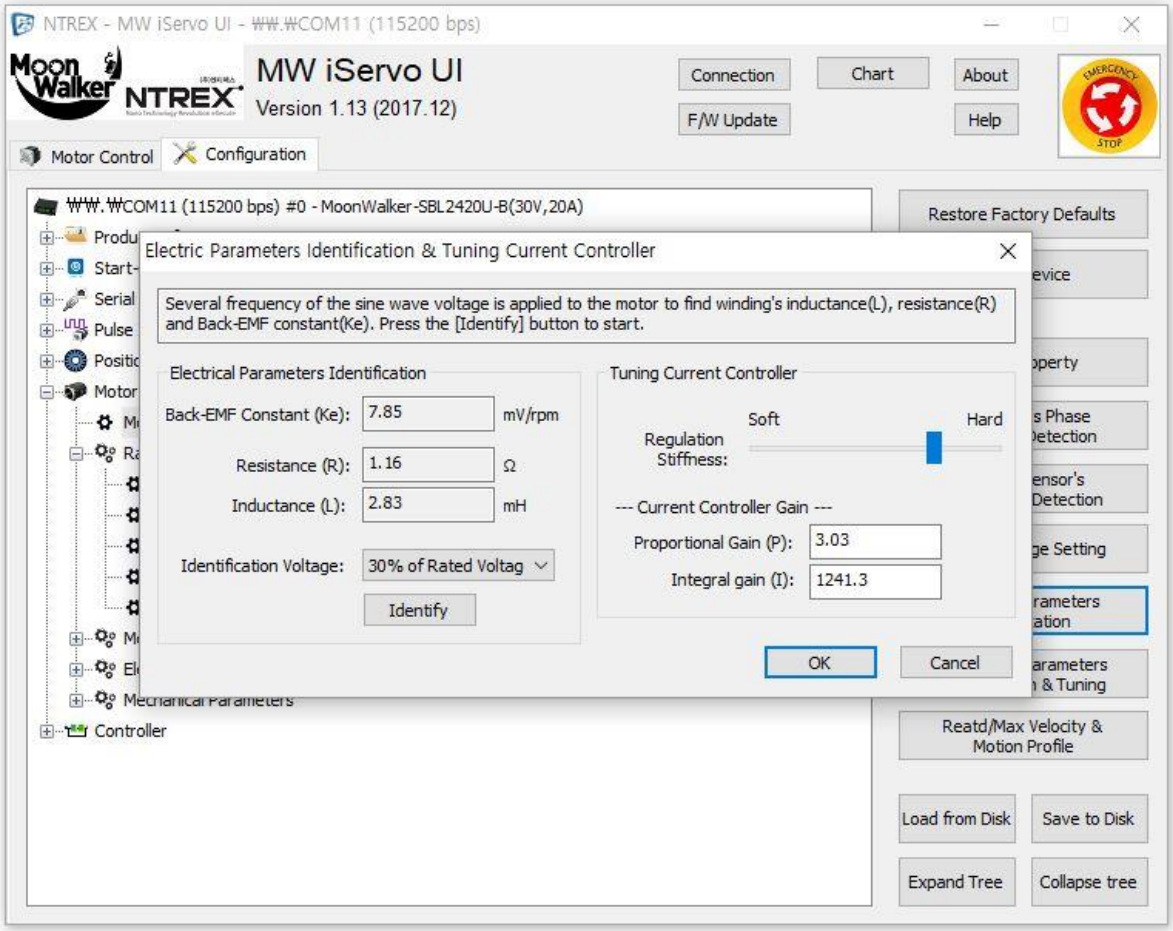
모터가 리니어 모듈에 연결되어 모터의 구동 범위가 제한되는 경우, 모터가 구동 가능한 최소/최대 범위를 설정하여 이 범위를 벗어나지 못하도록 합니다. 만일 이동로봇의 바퀴와 같이 모터가 무한 회전 가능한 경우는 이 대화상자의 내용을 설정하지 않아도 됩니다.

모터의 구동 범위를 제한해야 할 경우, Software Limit Check 체크박스를 체크하고 모터가 구동 가능한 Min Position 및 Max Position을 설정합니다. 이 때 손으로 구동부를 움직여 [Get Pos.] 버튼을 누름으로 현재 Encoder 값을 Min/Max Position 값으로 설정할 수 있습니다. 만일 [Reset] 버튼을 누르면 현재 위치를 Encoder 값 0으로 설정합니다.

***주의: 만일 Linear 타입 모터거나 모터가 리니어 모듈에 연결되어 이동 범위가 제한된 경우는 모터의 구동 범위를 꼭 올바르게 설정해야 합니다. 이 값이 올바르게 설정되지 않은 경우, 이후의 모터 파라미터 탐지 과정에서 모터가 스토퍼나 기타 기구부에 충돌할 수 있습니다.**

7.6.1.5 Electrical Parameters Identification & Tuning

[Electrical Parameters Identification & Tuning] 버튼을 눌러 모터의 전기적 파라미터를 찾고 이와 관련된 전류 제어기의 이득을 설정합니다.



[Identify] 버튼을 누르면 모터가 좌우로 회전하면서 모터의 전기 파라미터인 Back-EMF Constant, Resistance, d/q-axis Inductance 값을 찾게 됩니다. 만일 모터가 좌우로 움직이지 못하였거나 식별에 실패하는 경우는 경우는 Identification Voltage를 높여서 다시 식별 과정을 수행해 보도록 합니다.

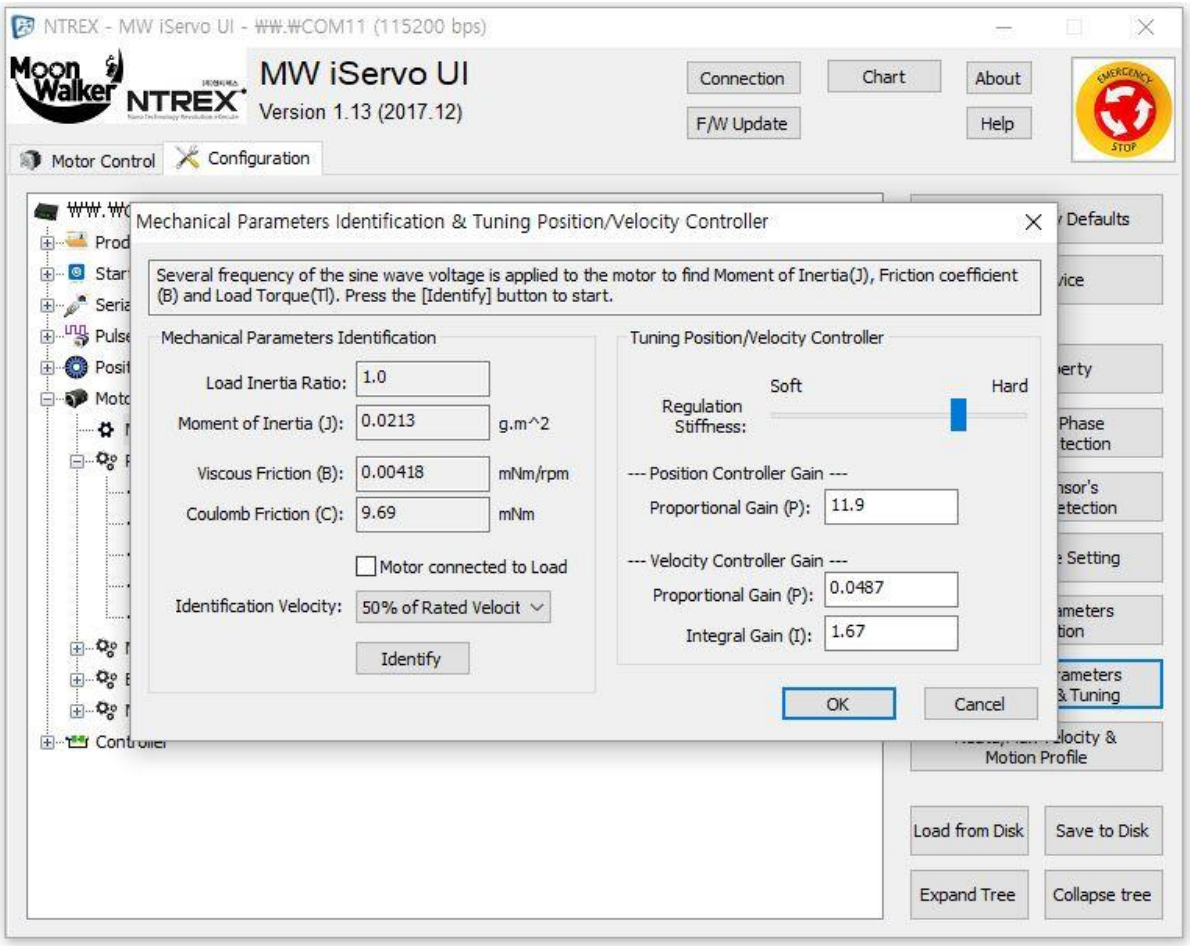
식별 과정을 성공적으로 수행한 경우, 오른 쪽의 Regulation Stiffness 슬라이드 바를 좌우로 움직여 전류 제어기의 이득을 설정합니다. Soft로 움직이면 이득이 낮게 설정되어 모터가 부드럽게 회전합니다. 반대로 Hard로 움직이면 이득이 높게 설정되어 모터의 전류 추종 성능이 좋아지지만 진동이 발생할 수 있습니다. 만일 전류제어기에서 이득을 너무 높게 설정하였다면, 모터는 가청

주파수 대(1kHz 근처)의 진동음을 내면서 진동하게 됩니다. 이 때는 소리가 나지 않을 때까지 슬라이드 바를 왼쪽으로 움직여 이득을 낮추도록 합니다.

***주의: 모터의 전기 파라미터를 찾을 때는 모터가 기구부와 분리되어 있어야 합니다. 만약 모터가 액추에이터등 기구부와 연결되어있다면 탐지된 파라미터가 부정확할 수 있으며 탐지 과정에서 모터가 스토퍼나 기타 기구부에 충돌할 수 있습니다. 또한 모터가 갑자기 움직일 수 있으므로 사용자가 다치지 않도록 주의해야 합니다.**

7.6.1.6 Mechanical Parameters Identification & Tuning

[Mechanical Parameters Identification & Tuning] 버튼을 눌러 모터의 기계적 파라미터를 찾고 이와 관련된 위치/속도 제어기의 이득을 설정합니다.



모터의 기계 파라미터를 찾을 때는 모터가 기구부와 연결되어 있는 상태인지 또는 분리되어 있는 상태인지를 구분해야 합니다. 만일 Motor connected to Load 체크박스를 체크한 상태라면 모터가 기구부와 연결되어 있는 상태로 보고, 현재 설정된 Moment of Inertia 값은 모터 회전자의 관성으로 고려하여 기구부(Load) 관성과 비율을 구하게 됩니다.

$$\text{Load Inertia Ratio} = \frac{\text{Load Inertia + Motor Inertial}}{\text{Motor Inertial}}$$

만일 체크가 해제된 상태라면 Load Inertia Ratio는 1이 되고, Moment of Inertial 값은 모터 회전자의 관성이 됩니다.

[Identify] 버튼을 누르면 모터가 가감속을 반복하면서 좌우로 이동하여 모터의 기계 파라미터인 Moment of Inertia, Viscous Friction, Coulomb Friction 값을 찾게 됩니다. 만일 모터가 좌우로 움직이지 못하였거나 식별에 실패하는 경우는 경우는 Identification Velocity를 높여서 다시 식별 과정을 수행해 보도록 합니다.

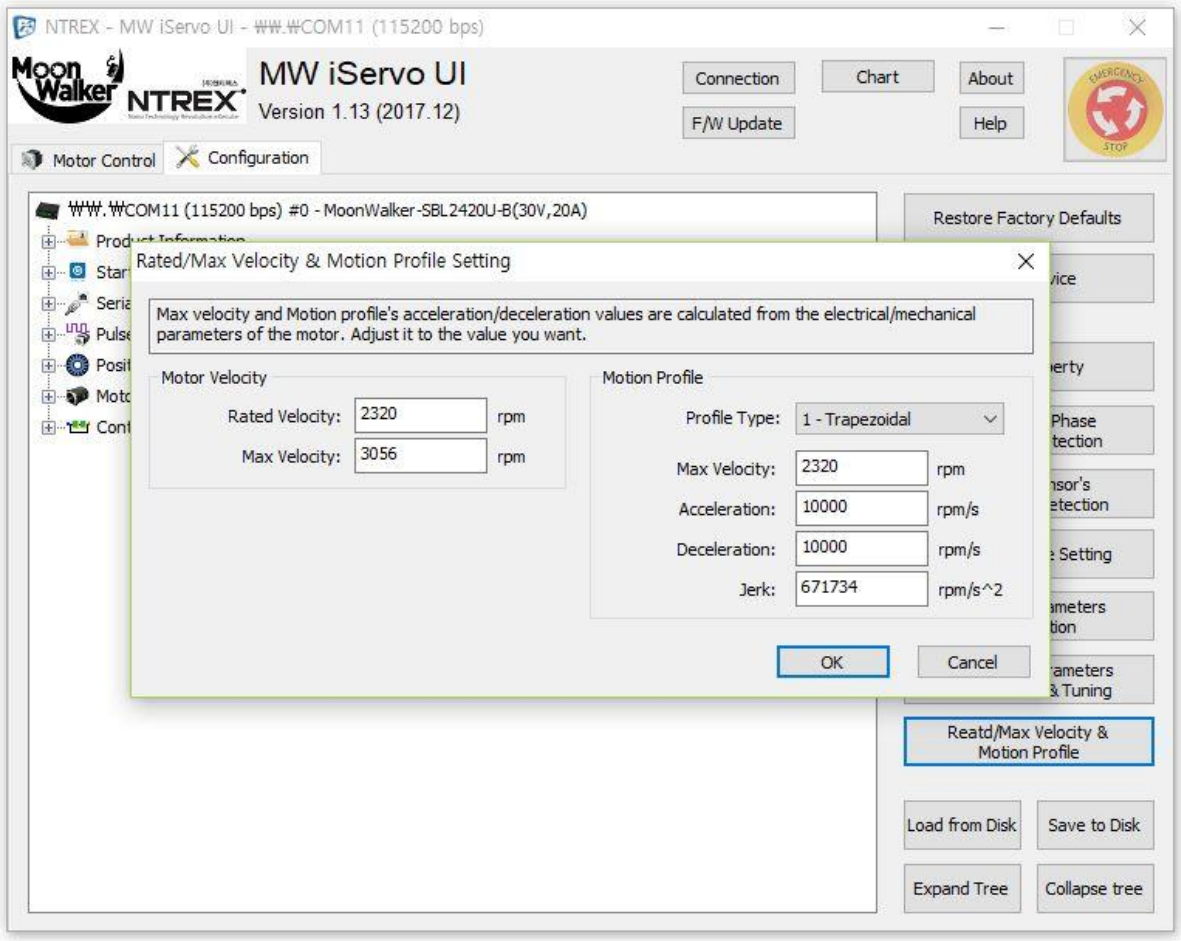
식별 과정을 성공적으로 수행한 경우, 오른 쪽의 Regulation Stiffness 슬라이드 바를 좌우로 움직여 위치/속도 제어기의 이득을 설정합니다. Soft로 움직이면 이득이 낮게 설정되어 모터가 부드럽게 회전합니다. 반대로 Hard로 움직이면 이득이 높게 설정되어 모터의 위치/속도 추종 성능이 좋아지지만 진동이 발생할 수 있습니다. 만일 위치/속도 제어기에서 이득을 너무 높게 설정하였다면, 모터는 저주파(수십~수백Hz)로 진동하게 됩니다. 이 때는 진동이 없어질 때까지 슬라이드 바를 왼쪽으로 움직여 이득을 낮추도록 합니다.

***주의: 이 값을 찾을 때 모터는 기구부와 연결되어 있어야 합니다. 모터는 진동하면서 값을 찾기 때문에 사람이 다치거나 기구부가 파손되지 않도록 주의해야 합니다.**

***주의: 위치 제어기와 속도 제어기의 이득을 설정할 때, PID값을 수동으로 각자 설정하는 것보다 Regulation Stiffness 슬라이드 바를 움직여 한꺼번에 설정하는 것을 권장합니다.**

7.6.1.7 Motion Profile & Feed-forward Options

[Motion Profile & Feed-forward Options] 버튼을 눌러 모터의 동작 프로파일을 설정하고 피드 포워드 옵션을 설정합니다.



Rated Velocity와 Max Velocity는 모터의 정격 전압과 Electrical Parameters Identification 과정에서 알아낸 Back-EMF constant에 의해 적절하게 계산된 값입니다.

Motion Profile 그룹에서, Profile Type을 설정하고 프로파일의 Max Velocity, Acceleration, Deceleration, Jerk 값을 설정합니다. [Electrical Parameters Identification & Tuning], [Mechanical Parameters Identification & Tuning] 과정을 올바르게 수행하였다면, 이 값들은 모터의 각종 파라미터를 기반으로 모터가 정격 범위 내에서 구동할 수 있도록 계산된 값으로 설정되어 있습니다.

※ Profile Type을 '1-Trapezoidal' 또는 '2-Sinusoidal' 중 하나로 설정하는 것을 권장하며, Max Velocity, Acceleration, Deceleration, Jerk 값들을 제어기가 계산한 값 이상으로 올리는 것은 모터에 무리를 가하게 되므로 권장하지 않지만, 이 값들을 낮추는 것은 문제가 되지 않습니다.

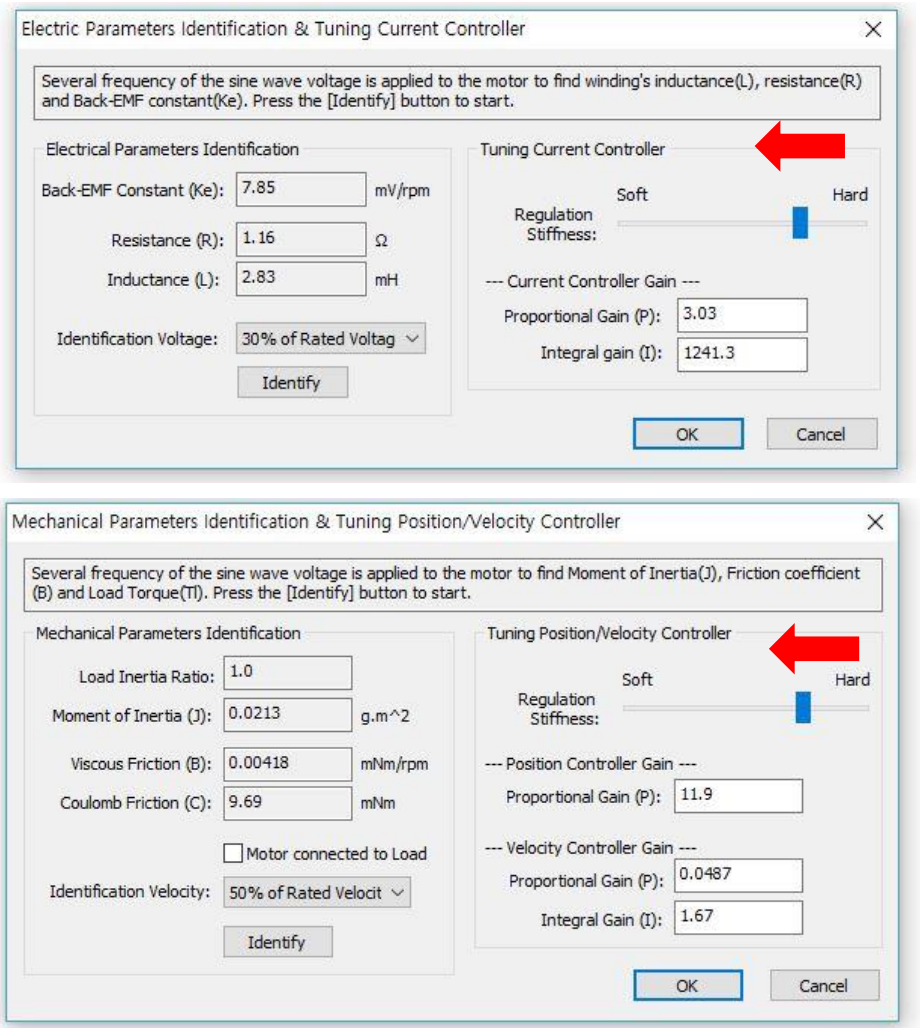
7.6.2 세부 튜닝

7.6.2.1 정지 상태에서 진동 발생하는 경우

UI의 Motor Control 탭에서 [Enable] 버튼을 눌러 모터를 Enable 하고 모터를 움직이지 않는 상태에서 진동이 발생하는 경우가 있을 수 있습니다.

진동은 종종 벨트 타입의 리니어 모듈을 구동하는 모터에서 Current Controller의 Regulation Stiffness를 최대한 Hard로 설정하였거나 Position/Velocity Controller의 Regulation Stiffness를 최대한 Hard로 설정하였을 때 발생합니다.

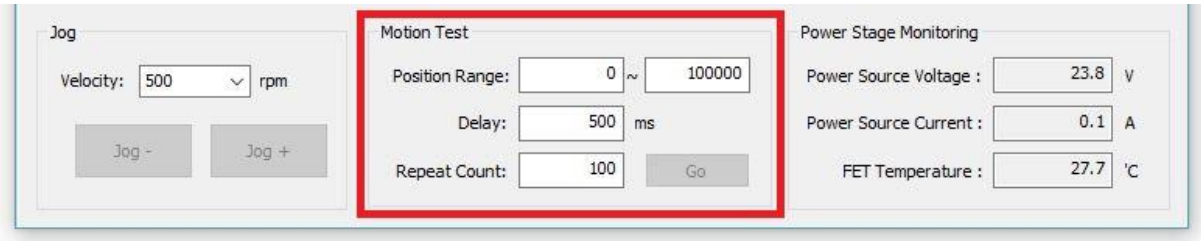
이 때는 Tuning Current Controller와 Tuning Position/Velocity Controller의 Regulation Stiffness를 Soft 방향으로 진동이 없어질 때까지 조금씩 낮춰 보도록 합니다.



7.6.2.2 구동 상태에서 진동 발생하는 경우

UI의 Motor Control 탭에서 [Enable] 버튼을 눌러 모터를 Enable 하고 모터가 특정 구간을 반복하

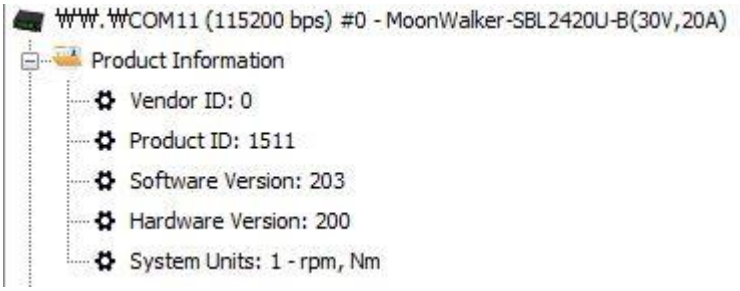
도록 Motion Test 그룹에서 Position Range, Delay, Repeat Count 값을 설정 합니다. 그리고 [Go] 버튼을 눌러 모터가 반복 구동하도록 합니다.



이후 "정지 상태에서 진동이 발생하는 경우" 절에서와 같은 방법으로 진동이 없어질 때까지 이득을 조절합니다.

7.6.3 Product Information

Product Information 그룹은 현재 연결된 제품공급자 ID(항상 0으로 표시됨), 제품 ID, 모터 드라이버와 펌웨어 버전, 제어기 하드웨어 버전을 확인할 수 있다. 이 값들은 사용자가 임의로 변경할 수 없습니다. 그리고 장치의 이름과 모터 드라이버에서 사용하는 단위를 설정할 수 있습니다.



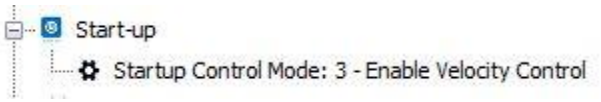
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Vendor ID – vendor_id
- Product ID – product_id
- Software Version – software_version
- Hardware Version – hardware_version
- System Units – units

7.6.4 Start-Up

Start-up/Stop 그룹은 모터 드라이버에 전원이 투입되어 처음 시작할 때의 동작을 결정합니다.



Startup Control Mode는 다음 중 하나를 선택할 수 있습니다:

- 0 – Disable
- 1 – Enable
- 2 – Enable and Current Control Mode
- 3 – Enable and Position Control Mode
- 4 – Enable and Velocity Control Mode

이 설정 값은 모터 제어기를 Enable 할 때 모터의 구동 모드를 결정할 때도 사용됩니다. 0이나 1로 설정된 경우 Voltage Output 모드로 Enable 되고, 2으로 설정된 경우 Current Control 모드가 됩니다. 3로 설정된 경우는 Velocity Control 모드가 되고, 4로 설정된 경우 Position Control 모드가 됩니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Startup Control Mode – startup_control_mode

7.6.5 Serial Connection

Serial Connection 그룹은 모터 드라이버 ID, Serial 통신 속도, 와치독 등을 설정하고 확인할 수 있습니다. 이 그룹에서 Device ID, RS-232/485 Baudrate 값을 변경한 경우에는 모터 드라이버를 재시작해야 변경된 값이 반영됩니다. Serial 통신속도는 115200bps 가 기본값입니다.



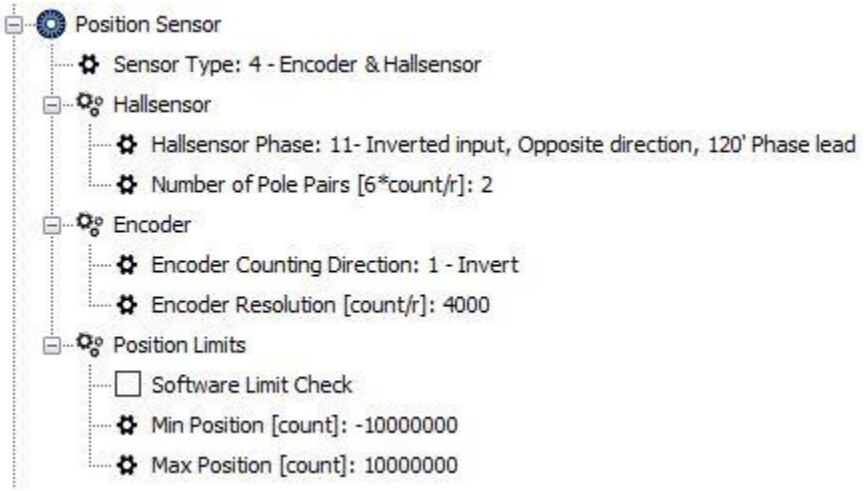
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Device ID - device_id
- Watchdog timeout - serial_watchdog
- RS-232/485 Baudrate - rs232_baudrate

7.6.6 Position Sensor

Position Sensor 그룹은 위치 센서에 대한 설정입니다. 위치 제어를 하기 위해 최소/최대 위치 값, 모터 1회전당 엔코더 펄스 수, Soft Limit 사용 여부를 설정할 수 있습니다.



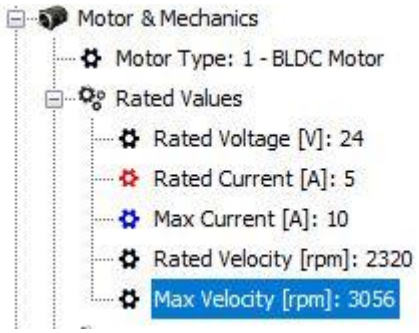
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Sensor Type - position_sensor
- Hallsensor – Hallsensor Phase - hallsensor_phase
- Hallsensor – Number of Pole Pairs - no_pole_pairs
- Encoder – Encoder Counting Direction - encoder_direction
- Encoder - Encoder Resolution - encoder_resolution
- Position Limits – Software Limit Check - soft_limit_check
- Position Limits – Min Position - min_position
- Position Limits – Max Position - max_position

7.6.7 Motor & Mechanics

Motor & Mechanics 그룹은 사용자가 사용하고자 하는 모터 자체의 특성치에 대한 설정입니다. 모터에 흐르는 최대 연속 전류, 모터를 구동하는 정격 전압, 모터의 최대 회전 속도, 모터의 회전 가속도, 모터의 회전 감속도를 설정할 수 있습니다.



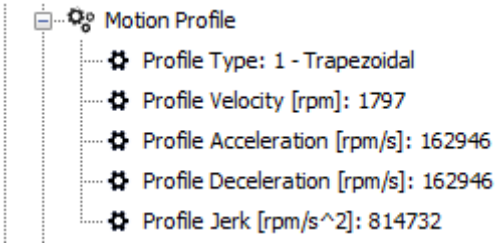
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Motor Type - motor_type
- Rated Values – Rated Voltage - rated_voltage
- Rated Values – Rated Current - rated_current
- Rated Values – Max Current - max_current
- Rated Values – Rated Velocity - rated_velocity
- Rated Values – Max Velocity - max_velocity

7.6.8 Motor & Mechanics – Motion Profile

Motors & Mechanics – Motion Profile 그룹은 모터의 위치나 속도 제어시 사용하는 프로파일의 형식과 프로파일의 모양을 결정하는 최고 속도, 가속도, 감속도, 저크 값을 설정합니다.



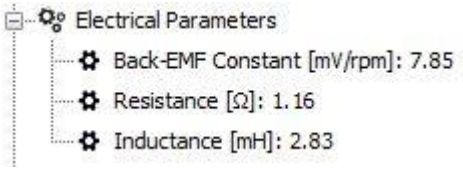
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Profile Type - profile_type
- Profile Velocity - profile_velocity
- Profile Acceleration - profile_acceleration
- Profile Deceleration - profile_deceleration
- Profile Jerk - profile_jerk

7.6.9 Motor & Mechanics – Electrical Parameters

Motors & Mechanics – Electrical Parameters 모터의 전기 파라미터인 역기전력 상수, 저항, 인덕턴스 값들을 설정합니다.



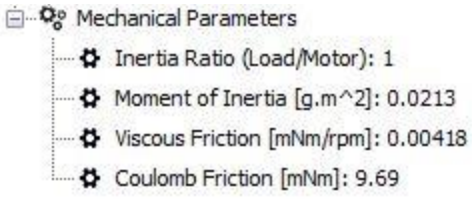
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Back-EMF Constant - bemf_constant
- Resistance[Ω] - resistance
- Inductance[mH] - inductance

7.6.10 Motor & Mechanics – Mechanical Parameters

Motors & Mechanics – Electrical Parameters 모터의 기계 파라미터인 관성모멘트, 정지 마찰계수, 운동마찰계수 값들을 설정합니다.



※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

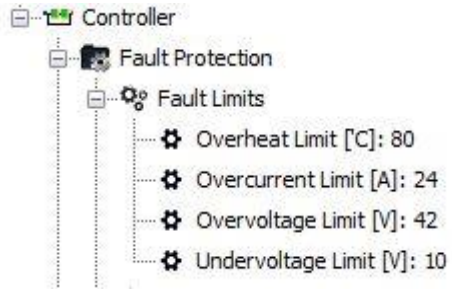
이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Inertia Ratio - inertia_ratio
- Moment of Inertia - moment_of_inertia
- Viscous Friction - viscous_friction
- Coulomb Friction - coulomb_friction

7.6.11 Controller – Fault Protection – Fault Limits

Controller – Fault Protection – Fault Limits 그룹에서는 모터 및 제어기를 보호하기 위한 과열, 과전

류, 과전압 한계 값을 설정합니다.



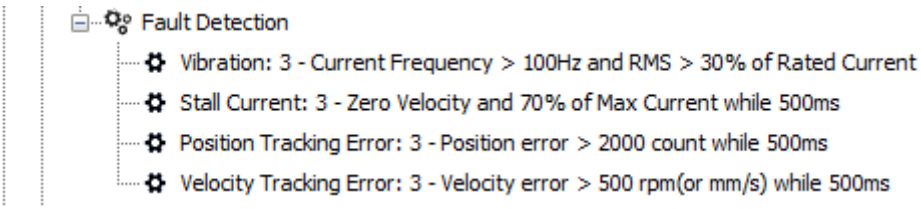
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Overheat Limit - overheat_limit
- Overcurrent Limit - overcurrent_limit
- Overvoltage Limit - overvoltage_limit
- Undervoltage Limit - undervoltage_limit

7.6.12 Controller – Fault Protection – Detection

Controller – Fault Protection - Fault Detection 그룹에서는 모터 및 모터에 연결된 장비를 보호하기 위한 각종 폴트 감지 조건을 설정합니다.



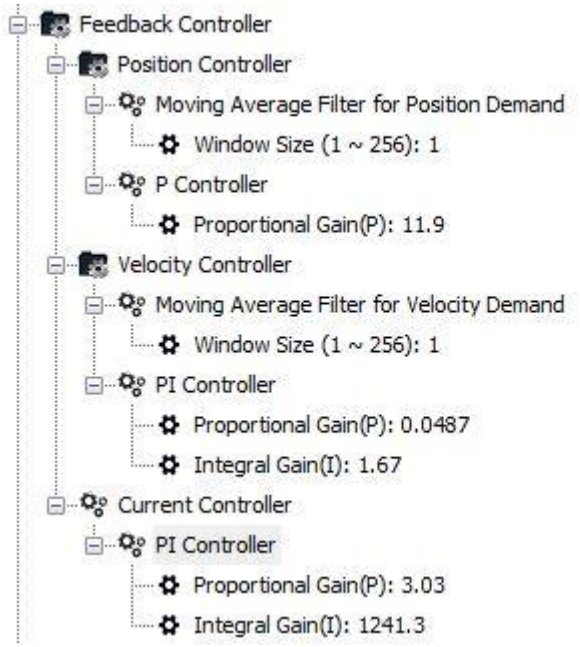
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Vibration - vibration_detect
- Stall Current - stall_current_detect
- Position Tracking Error - position_track_error
- Velocity Tracking Error - velocity_track_error

7.6.13 Controller – Feedback Controller

Feedback Controller는 모터 드라이버 피드백 제어기의 PI 이득과 필터 계수를 설정할 수 있습니다.



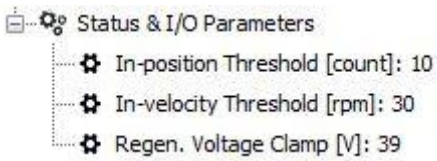
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

- Position Controller – Moving Average Filter – Window Size - pmaf_window_size
- Position Controller – P Controller – Proportional Gain - position_p_gain
- Velocity Controller – Moving Average Filter – Window Size - vmaf_window_size
- Velocity Controller – PI Controller – Proportional Gain - velocity_p_gain
- Velocity Controller – PI Controller – Integral Gain - velocity_i_gain
- Current Controller – PI Controller – Proportional Gain - current_p_gain
- Current Controller – PI Controller – Integral Gain - current_i_gain

7.6.14 Controller – Status & I/O Parameters

Controller – Status & I/O Parameters 그룹에서는 모터 드라이버에서 지원하는 입출력 포트의 동작과 관련된 파라미터를 설정 할 수 있습니다.



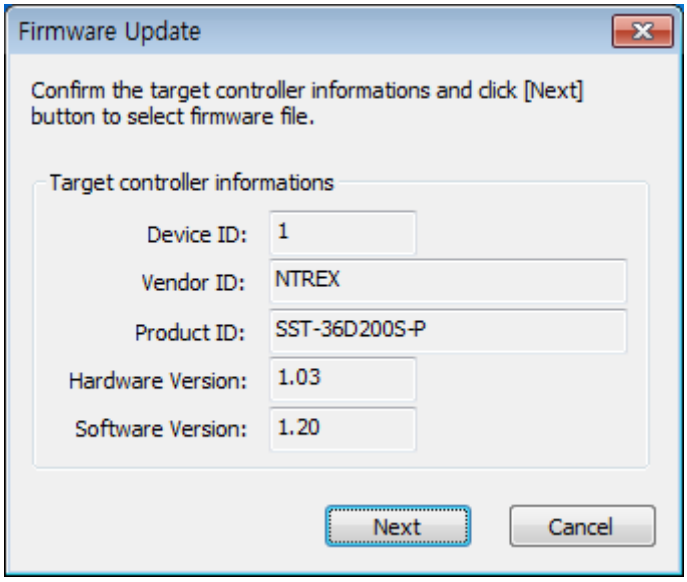
※ 자세한 내용은 오브젝트 테이블 항목에 있습니다.

이 그룹의 항목과 관련되는 오브젝트는 다음과 같습니다.

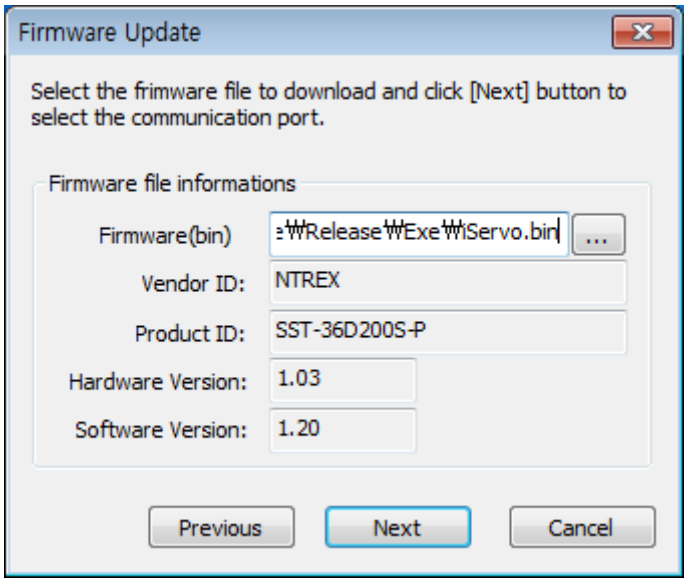
- In-position Threshold - in_position_threshold
- In-velocity Threshold - in_velocity_threshold
- Regen. Voltage Clamp - regen_voltage_clamp

7.7 Firmware Update

Firmware Update 대화상자는 UI 유틸리티 헤더의 [F/W Update] 버튼을 눌러 실행합니다. 만일 현재 모터 드라이버와 연결되어 있다면, 다음 그림과 같이 연결된 모터 드라이버의 Device ID, Vender ID, Product ID, H/W Version, S/W Version 정보를 표시합니다. 모터 드라이버 정보를 확인하고 [Next] 버튼을 눌러 다음 스탭으로 이동하거나 [Cancel] 버튼을 눌러 펌웨어 업데이트를 취소할 수 있습니다.

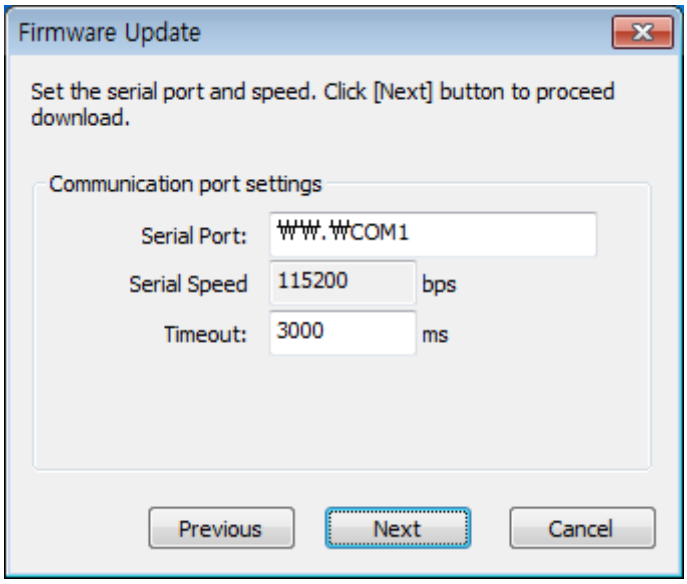


현재 모터 드라이버 버전을 확인한 후 [Next] 버튼을 누르면 다음 그림과 같이 업데이트할 펌웨어를 선택할 수 있는 대화상자가 표시됩니다.

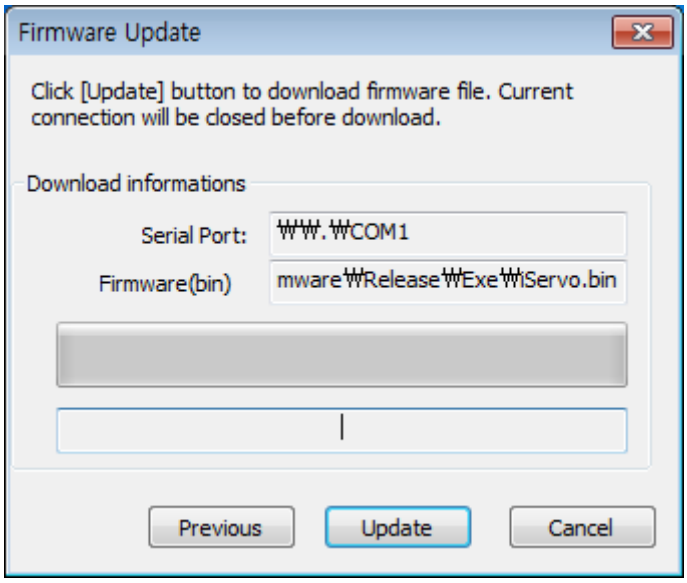


이 대화상자에서 Binary(bin) file 옆의 [...]버튼을 눌러서 엔티렉스 연구소 홈페이지, 혹은 디바이스 마트에서 다운받은 펌웨어 파일을 선택합니다. 그러면 펌웨어 파일의 Vender ID, Product ID, H/W

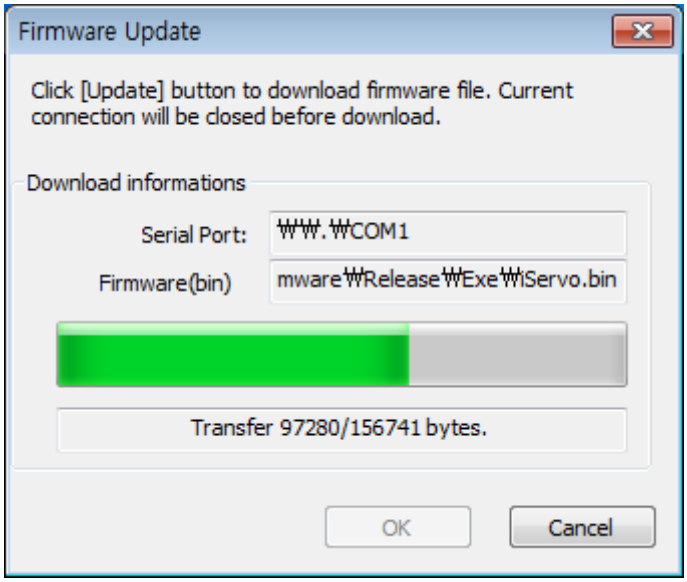
Version, S/W Version을 확인할 수 있습니다. 이 정보와 대상 모터 드라이버의 모델 정보가 일치하는지 확인해야 합니다. 그 후 [Next] 버튼을 누르면 다음 그림과 같이 펌웨어를 다운로드 할 포트 정보가 표시됩니다.



여기서는 모터 드라이버에 펌웨어를 다운로드 할 시리얼 포트와 와 통신속도, 패킷 전송 Timeout을 설정합니다. 그리고 [Next] 버튼을 누르면 다음과 같이 펌웨어를 다운로드 하기위한 대화상자가 표시됩니다.



여기서 다시한번 펌웨어를 다운로드 하기위한 시리얼 포트와 펌웨어 파일을 확인합니다. 마지막으로 [Update] 버튼을 누르면 기존의 모든 모터 드라이버 연결이 종료되고 다음 그림에서와 같이 선택한 펌웨어 다운로드가 시작됩니다.



펌웨어 다운로드 중 [Cancel] 버튼을 누르면 다운로드 과정을 취소하고 펌웨어 업데이트 대화상자를 닫습니다. 펌웨어 다운로드가 완료가 되면 [OK] 버튼이 표시되고, [OK] 버튼을 눌러 펌웨어 업데이트 과정을 종료합니다.

8 통신 프로토콜

8.1 Communication Protocol

이 장에서는 PC나 마이크로 컨트롤러에서 모터 드라이브의 오브젝트 값을 읽고 쓰기 위한 통신 프로토콜에 대해 설명합니다. 모터 드라이브에 동작을 명령하거나 모터 드라이브의 구성 파라미터를 설정하는 것은 해당 오브젝트에 특정 값을 쓰는 것을 의미하며, 모터 드라이브의 상태를 읽거나 모터 드라이브의 구성 파라미터를 읽는 것은 해당 오브젝트에서 특정 값을 읽는 것을 의미합니다.

통신 프로토콜은 크게 CAN 포트에서 사용되는 프로토콜과 시리얼(RS-232, RS-485) 포트에서 사용되는 프로토콜로 구분됩니다. 시리얼 포트 프로토콜은 다시 패킷의 구조에 따라 **바이너리(binary)** 형태와 **텍스트(text)** 형태로 구분됩니다.

8.1.1 용어의 정리

모터 드라이브의 객체를 통신 프로토콜로 읽고 쓰는데 사용되는 용어를 표 8-1에서 정의합니다.

Term	Description
Index	오브젝트의 참조 색인
Sub-index	오브젝트의 참조 부-색인
Short Name	Index와 호환되는 오브젝트의 짧은 이름
Long Name	Index와 호환되는 오브젝트의 긴 이름

표 8-1 통신 프로토콜에 사용되는 용어

상기 표에서 사용되는 용어 중 Index, Short Name, Long Name은 오브젝트를 표현하는 방식의 차이입니다. Index는 모터 드라이브 내부의 오브젝트를 참조하는 색인으로, CAN 및 시리얼 바이너리 패킷에 사용됩니다.

Sub-index는 오브젝트를 참조하는 부-색인으로, 통상 아날로그/디지털 I/O의 채널 번호를 의미합니다. 채널 1에 연결된 I/O에 대한 명령이나 설정이라면 Sub-index로 1을, 채널 2에 연결된 I/O의 경우는 2를 사용하면 됩니다. I/O가 아니라 모터 드라이브 자체에 대한 것이라면 통상 Sub-index는 0이 됩니다.

Short Name과 Long Name은 Index와 호환되는 오브젝트의 짧은 이름과 긴 이름으로, 시리얼 텍스트 패킷에서 사용됩니다. 이 이름은 모터 드라이브 내부에서 Index 값으로 변환되어 Index와 동일하게 사용됩니다.

만약 사용자가 모터 드라이브에 인가된 주 전원의 전압을 체크하는 **power_source_voltage** 오브젝트의 내용을 읽고자 할 때, 시리얼 텍스트 기반으로 통신하는 경우, 해당 오브젝트의 Short Name인 **pv**나 Long Name인 **power_source_voltage**를 사용하면 됩니다. CAN이나 시리얼 바이너

리 패킷 통신을 사용하는 경우는 해당 오브젝트의 Index인 **0x2008**을 사용하면 됩니다.

표 8-2에서는 오브젝트의 4가지 속성을 보여줍니다. 각각의 오브젝트 타입에 따라 패킷의 길이가 달라질 수 있습니다.

Object Type	Size	Description
INT8	1byte	부호를 가지는 8 bit 정수형
INT16	2bytes	부호를 가지는 16 bit 정수형
INT32	4bytes	부호를 가지는 32 bit 정수형
FLOAT	4bytes	부호를 가지는 32 bit 실수형

표 8-2 오브젝트의 타입

또한, 표 8-3에서는 오브젝트의 엑세스 속성을 나타냅니다. RO로 표시된 오브젝트는 읽기 전용으로 상수(Constant)와 상태(Status) 오브젝트가 여기에 해당하며, 값을 쓰려고 할 때 에러를 리턴할 것입니다. WO로 표시된 오브젝트는 쓰기 전용으로 명령(Command) 오브젝트가 여기에 해당하며, 값을 읽으려고 할 때 에러를 리턴할 것입니다. RW로 표시된 오브젝트는 구성 파라미터(Configuration Parameter)와 변수(Variable) 오브젝트가 여기에 해당합니다.

Access	Object	Description
RO	Constant, Status	읽기 전용 (Read Only)
WO	Command	쓰기 전용 (Write Only)
RW	Configuration Parameter, Variable	읽고 쓰기 가능 (Read Writeable)

표 8-3 오브젝트의 엑세스 속성

8.2 CAN 메시지

모터 드라이브의 구성 파라미터(Configuration Parameter)를 설정하고 명령(Command)을 내리거나 상태(Status)를 읽기 위해서 모터 드라이브에 존재하는 여러 오브젝트들을 CAN 통신으로 엑세스 할 수 있습니다. 이 절은 모터 드라이브에 존재하는 오브젝트들을 엑세스하기 위한 CAN 통신의 메시지 구조에 대해 기술합니다.

※ 마스터 PC와 모터 드라이브가 CAN 버스에 연결되어 통신하기 위해서는, 네트워크에 연결된 모든 노드의 통신 속도가 일치해야 하고 각각의 장치 ID는 모두 달라야 합니다.

8.2.1 CAN 패킷의 기본 구조

마스터 PC와 모터 드라이브 간에 CAN 통신으로 주고받는 패킷의 주요 구성 요소는 Device ID와 메시지의 길이(Length) 그리고 전송되는 메시지(Message)가 됩니다(아래 그림 8-2-1 참조).

Device ID	Length	Message
11bit	0~8	0 ~ 8byte

그림 8-2-1 CAN 통신 패킷 구성 요소

Device ID로는 최대 11bit를 설정할 수 있는데, 모터 드라이브의 **device_id**가 여기에 사용됩니다. 마스터에서 장치로 CAN 메시지를 보내거나 장치가 마스터로 회신 할 때, Device ID가 동일하게 지정되어야 합니다. Length는 메시지(Message)의 길이를 나타내는데, 메시지의 최대 크기가 8byte 이므로 0에서 8사이의 값이 됩니다. 메시지에는 전송되는 데이터를 담습니다.

그림 8-2-2는 메시지(Message)의 기본 구조를 나타냅니다. 메시지는 최대 8byte 크기를 가질 수 있으며, 1byte의 Command와 2byte의 Index, 1byte의 Sub-index와 나머지는 상황에 맞는 오브젝트의 값(Value)으로 구성되어 있습니다. 상황에 따라 5th ~ 8th byte 지점의 값들은 존재할 수도 있고 아닐 수도 있습니다. 만약 5th byte 지점까지만 Value가 존재한다면 5th byte까지의 내용만 전송하면 됩니다.

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Sub-index	Value			

그림 8-1-2 CAN 통신 패킷 구성 요소

여기서 Command는 표 8-4과 같은 액세스 형식(Access Code)과 오브젝트 형식(Object Type)의 조합으로 1byte를 구성하게 됩니다.

구분	코드	내용
Access Code	0x10	오브젝트 쓰기 요청
	0x20	오브젝트 쓰기 요청에 대한 응답
	0x30	오브젝트 읽기 요청
	0x40	오브젝트 읽기 요청에 대한 응답
	0x80	오브젝트 읽기/쓰기 요청에 대한 에러 응답
Object Type	0x00	INT8 - 8 bit signed integer
	0x04	INT16 - 16 bit signed integer
	0x08	INT32 - 32 bit signed integer
	0x0C	FLOAT - 32 bit IEEE Standard 754 floating-point

표 8-4 액세스 형식(Access Code)과 오브젝트 형식(Object Type)

Index나 Value와 같이 한 바이트 이상의 데이터가 메시지에 저장될 때는 데이터의 하위 바이트부터 왼쪽에 저장되는 **리틀 인디안(Little-Endian)방식**을 따릅니다.

8.2.2 오브젝트 읽기 요청

마스터 PC가 모터 드라이브의 오브젝트를 읽기 위해, 마스터 PC가 모터 드라이브에 보내는 쿼리 패킷을 구성합니다.

오브젝트의 값을 읽을 때는 표 8-4의 Access Code 0x30과 읽고자 하는 Object Type을 조합해서 Command를 먼저 구성해야 합니다. 읽고자 하는 오브젝트의 형이 INT16 이라면 Command에는 0x34를 사용합니다.

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Sub-index				

8.2.3 오브젝트 쓰기 요청

마스터 PC가 모터 드라이브의 오브젝트에 값을 쓰기 위해, 마스터 PC가 모터 드라이브에 보내는 쿼리를 구성합니다.

오브젝트에 값을 쓸 때는, 4th byte 지점 까지는 오브젝트 읽기 요청과 동일하게 구성합니다. 단, Command는 표 8-4의 Access Code에 의해 0x10과 Object Type에 맞는 조합으로 작성되어야 하며, 그 형식에 맞춰 오브젝트의 값(Value)을 작성합니다.

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Sub-index	Value(INT8)			
				Value(INT16)			
				Value(INT32)			
				Value(FLOAT)			

8.2.4 오브젝트 읽기/쓰기 요청에 대한 성공 응답

모터 드라이브가 마스터 PC의 오브젝트 읽기/쓰기 요청에 응답하기 위해, 모터 드라이브가 마스터 PC에 보내는 응답 패킷을 구성합니다.

오브젝트의 값을 읽거나 쓰기가 성공한 경우에는 오브젝트의 값(Value)이 응답으로 돌아옵니다. 패킷의 구성은 오브젝트 쓰기 요청에서와 같습니다. 단, Command는 표 8-4의 Access Code에 제시된 것처럼 0x20이나 0x40중 하나와 Object Type에 맞는 조합으로 작성됩니다.

<table>
<tr><th>1st byte</th><th>2nd byte</th><th>3rd byte</th><th>4th byte</th><th>5th byte</th><th>6th byte</th><th>7th byte</th><th>8th byte</th></tr>
<tr><td rowspan="4">Command</td><td colspan="2" rowspan="4">Index</td><td rowspan="4">Sub-index</td><td>Value(INT8)</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Value(INT16)</td><td></td><td></td></tr>
<tr><td colspan="4">Value(INT32)</td></tr>
<tr><td colspan="4">Value(FLOAT)</td></tr>
</table>

※ 모터 드라이브는 사용자가 쓴 값이 적용되지 않을 수 있기 때문에 사용자의 오브젝트를 읽는 요청뿐만 아니라 오브젝트에 쓰기 요청에 대해서도 해당 오브젝트의 값을 응답합니다.

8.2.5 오브젝트 읽기/쓰기 요청에 대한 실패 응답

모터 드라이브가 마스터 PC의 오브젝트 읽기/쓰기 요청에 실패를 알리기 위해, 모터 드라이브가 마스터 PC에 보내는 실패 응답 패킷을 구성합니다.

오브젝트의 내용을 읽거나 쓰기 위한 패킷을 모터 드라이브에 요청했을 때, 오류가 발생하는 경우는 다음 그림 8-2-3과 같은 형식의 오류 패킷이 돌아옵니다.

<table>
<tr><th>1st byte</th><th>2nd byte</th><th>3rd byte</th><th>4th byte</th><th>5th byte</th><th>6th byte</th><th>7th byte</th><th>8th byte</th></tr>
<tr><td>Command</td><td colspan="2">Error Code</td><td colspan="5"></td></tr>
</table>

그림 8-2-3 CAN 오류 패킷

오류 메시지는 표 8-5에 나타나 있습니다. CAN 패킷에서 오류 메시지 내용은 전달되지 않으며 Error Code만 반환됩니다. 그리고 Command는 0x80이 반환됩니다.

Error Code	Error Message	내용
1	undefined index	Index나 Sub-index로 지정한 오브젝트가 존재하지 않음
2	packet format error	패킷의 구성이 잘못 되었음
3	variable access error	읽기 전용 오브젝트에 쓰거나, 쓰기 전용 오브젝트의 읽기를 시도함

표 8-5 오브젝트의 읽기/쓰기 과정에서 발생하는 오류

※ 본 절의 오류 메시지는 패킷의 구성에 관련된 오류로 모터 드라이브의 오류 상황이나 오작동에 대한 부분이 아닙니다.

8.3 시리얼 바이너리 패킷

모터 드라이브의 오브젝트들은 시리얼(RS-232, RS-485) 통신으로도 읽고 쓰기가 가능합니다. 시리얼 통신은 텍스트 기반의 패킷과 함께 바이너리 기반의 패킷 통신을 함께 지원합니다.

이 절은 모터 드라이브에 존재하는 오브젝트들을 엑세스하기 위한 시리얼 통신의 바이너리 패킷 구조에 관해 설명합니다.

8.3.1 바이너리 패킷의 기본 구조

바이너리 패킷의 메시지 구조는 CAN 패킷에서 사용되는 메시지 구조와 동일합니다. 단 CAN에서는 Object Type에 따라 패킷의 길이가 바뀌지만, 바이너리 패킷 구조에서는 Value의 최대 길이를 4byte로 고정하고 패킷 전체 길이를 13byte로 고정합니다.

1st byte	2nd byte	3rd byte	4th ~ 11th byte	12th byte	13th byte
STX (0x02)	Length (=13)	Device ID	Message	Checksum	ETX (0x03)

그림 8-3-1 바이너리 패킷 구성 요소

상기 그림 8-3-1에서 처음과 끝의 STX, ETX는 시작과 끝을 의미하는 문자입니다. 그리고 Length는 13으로 고정되어 있습니다.

Checksum은 3rd ~ 11th byte 지점 까지를 바이트 단위로 더한 후 결과에서 1byte만 취한 값입니다. 다음은 C언어로 작성된 Checksum 계산 예제 코드입니다:

```
char Checksum (char *msg, int len)
{
	char cs = 0;

	for (int i=0; i<len; ++i)
		cs += msg[i];
	return cs;
}
```

4th ~ 11th byte 영역의 Message는 CAN 통신에서의 메시지 구조와 동일합니다. 단 CAN은 8byte의 메시지를 모두 채우지 않아도 되지만, 시리얼 바이너리 패킷에서는 8byte의 메시지를 모두 채워야 합니다. 만약 메시지의 크기가 8byte가 되지 않는다면, 남는 공간에 0을 채우면 됩니다.

시리얼 바이너리 패킷에서도 CAN 패킷에서와 같이 리틀 인디안(Little-Endian)방식으로 패킷을 구성합니다.

8.3.2 오브젝트 읽기 요청

오브젝트의 값을 읽을 때는, 패킷의 4th ~ 11th byte 지점에 위치하는 Message를 아래와 같이 구성하면 됩니다. Command는 표 8-4의 Access Code 0x30과 읽고자 하는 Object Type을 조합해서 Command를 먼저 구성해야 합니다.

4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte	11th byte
Command	Index		Sub-index	0	0	0	0

8.3.3 오브젝트 쓰기 요청

오브젝트에 값을 쓸 때는, 7th byte 지점 까지는 오브젝트 읽기 요청과 동일하게 구성합니다. 단 Command는 표 8-4의 Access Code에 의해 0x10과 오브젝트 형에 맞는 조합으로 작성되어야 하며, 그 형식에 맞춰 오브젝트의 값(Value)을 작성합니다.

4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte	11th byte
Command	Index		Sub-index	Value(INT8)	0	0	0
				Value(INT16)		0	0
				Value(INT32)			
				Value(FLOAT)			

8.3.4 오브젝트 읽기/쓰기 요청에 대한 성공 응답

오브젝트의 값을 읽거나 쓰기가 성공한 경우에는 오브젝트의 값(Value)이 응답으로 돌아옵니다. 패킷의 구성은 오브젝트 쓰기 요청에서와 같습니다. 단, Command는 표 8-4의 Access Code에 제시된 것처럼 0x20이나 0x40중 하나와 Object Type에 맞는 조합으로 작성됩니다.

4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte	11th byte
Command	Index		Sub-index	Value(INT8)	0	0	0
				Value(INT16)		0	0
				Value(INT32)			
				Value(FLOAT)			

모터 드라이브는 사용자가 쓴 값이 적용되지 않을 수 있기 때문에 사용자의 오브젝트를 읽는 요청뿐만 아니라 오브젝트에 쓰기 요청에 대해서도 해당 오브젝트의 값을 응답합니다.

8.3.5 오브젝트 읽기/쓰기 요청에 대한 실패 응답

오브젝트의 내용을 읽거나 쓰기 위한 패킷을 모터 드라이브에 요청했을 때, 오류가 발생하는 경우는 다음 그림 8-3-2와 같은 형식의 오류 패킷이 돌아옵니다.

4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte	11th byte
Command	Error Code		0	0	0	0	0

그림 8-3-2 오류 패킷

오류 메시지는 표 8-5에 나타나 있습니다. 시리얼 바이너리 패킷에서 오류 메시지 내용은 전달되지 않으며 Error Code만 반환됩니다. 그리고 Command는 0x80이 반환됩니다.

※ 오류 메시지는 패킷의 구성에 관련된 오류로 모터 드라이브의 오류 상황이나 오작동에 대한 부분이 아닙니다.

8.4 시리얼 텍스트 패킷

시리얼(RS-232, RS-485) 통신에서는 텍스트(text) 기반으로 모터 드라이브의 오브젝트를 읽고 쓸 수 있도록 합니다. 이는 사용자가 통신을 위한 전용 프로그램을 사용하지 않더라도 Hyperterminal과 같은 유틸리티를 사용하여 모터 드라이브의 오브젝트들을 쉽게 엑세스 할 수 있도록 합니다.

시리얼 텍스트 기반 패킷에서는 Index를 직접적으로 사용하지 않고 표 8-1의 Short Name이나 Long Name과 Sub-index를 사용합니다.

8.4.1 오브젝트 읽기 요청

오브젝트 값을 읽을 때는 텍스트 패킷을 다음과 같이 구성합니다.

Short/Long Name	CR/LF

Short/Long Name	Sub-index	CR/LF

Short/Long Name	[	Sub-index	]	CR/LF

Short/Long Name은 오브젝트의 Index에 해당하는 이름이며, Sub-index는 해당 Index에 따라 필요하면 I/O의 채널을 의미합니다. Sub-index가 0인 경우 생략 가능합니다.

CR/LF는 Carriage Return/Line Feed의 약자로 ASCII 코드상 각각 0x0D, 0x0A입니다. 일반적으로는 키보드의 Enter 키에 해당하며 하이퍼터미널과 같은 유틸리티로 PC에서 연결했다면 명령 입력 후 Enter 키를 입력하는 것입니다. (이후 CR/LF를 ↲ 기호로 대체)

예제) 모터 드라이버의 **user_variable**은 sub-index가 0~255 범위의 배열형 오브젝트입니다. 이 오브젝트의 sub-index 가 1인 곳의 값을 읽기 위해서는 다음과 같이 명령을 입력하고 엔터 키를 입력합니다.

```
user_variable[1]↵
user_variable1↵
```

8.4.2 오브젝트 쓰기 요청

오브젝트 값을 쓸 때는 텍스트 패킷을 다음과 같이 구성합니다.

Short/Long Name	=	Value	CR/LF

Short/Long Name	Sub-index	=	Value	CR/LF

Short/Long Name	[	Sub-index	]	=	Value	CR/LF

Short/Long Name은 오브젝트의 Index에 해당하는 이름이며, Sub-index는 해당 Index에 따라 필요하다면 I/O의 채널을 의미합니다. Sub-index가 0인 경우 생략 가능합니다.

예제) 모터 버에 연결된 모터를 1000rpm 속도로 회전하기 위해서는 다음과 같이 명령을 입력하고 엔터 키를 입력합니다.

```
tv=1000↵
target_velocity=1000↵
```

8.4.3 오브젝트 읽기/쓰기 요청에 대한 성공 응답

오브젝트의 값을 읽거나 쓰기가 성공한 경우에는 오브젝트의 값(Value)이 응답으로 돌아옵니다. 모터 드라이브가 응답하는 형식은 다음과 같습니다.

Short/Long Name	=	Value	CR/LF

Short/Long Name	Sub-index	=	Value	CR/LF

Short/Long Name	[	Sub-index	]	=	Value	CR/LF

사용자가 Short/Long Name 형식 중 하나로 질의를 하면, 같은 형식으로 응답합니다. 응답하는 문장의 끝에는 CR 문자와 LF 문자가 함께 붙습니다.

모터 드라이브는 사용자가 쓴 값이 적용되지 않을 수 있기 때문에 사용자의 오브젝트를 읽는 요청뿐만 아니라 오브젝트에 쓰기 요청에 대해서도 해당 오브젝트의 값을 응답합니다.

8.4.4 오브젝트 읽기/쓰기 요청에 대한 실패 응답

오브젝트의 내용을 읽거나 쓰기 위한 패킷을 모터 드라이브에 요청했을 때, 오류가 발생하는 경

우는 다음 그림과 같은 형식의 오류 패킷이 돌아옵니다.

!	Error Code	,	Error Message	CR/LF

오류 메시지는 표8-5에 나타나 있습니다.

※ 오류 메시지는 패킷의 구성에 관련된 오류로 모터 드라이브의 오류 상황이나 오작동에 대한 부분이 아닙니다.

8.4.5 Device ID 의 부여

만일 둘 이상의 모터 드라이브가 RS-485 버스에 연결된 경우, 시리얼 텍스트 패킷에는 각각의 모터 드라이브를 구분하는 장치의 ID를 부여해야 합니다.

이 때는 다음 그림과 같이 패킷을 수신할 장치의 ID를 패킷 앞에 붙이면 됩니다.

Device ID	;	Serial Text Packet

예제) 만약 **device_id** 1번을 가지는 모터 드라이브의 모터를 10rpm 속도 구동시키고자 할 때 아래와 같이 입력합니다.

```
1;tv=10↲
1;target_velocity=10↲
```

모터 드라이브가 장치 ID를 가지는 패킷을 수신하였을 때는, 응답 패킷에도 장치 ID를 붙여 응답합니다. 응답 패킷도 다음과 같은 형태가 됩니다.

Device ID	;	Serial Text Packet

***주의 RS-422이나 RS-485 버스에 둘 이상의 모터 드라이브를 연결할 때는, 각각의 모터 드라이브 간에 Device ID가 서로 충돌하지 않도록 설정하여야 합니다.**

8.4.6 여러 오브젝트의 읽기/쓰기

시리얼 텍스트 패킷 통신에서 하나의 요청으로 여러 오브젝트를 동시에 읽고 쓸 수 있습니다. 각각의 읽기와 쓰기 요청을 구분하는 문자는 ';' 입니다.

다음은 3개의 읽기/쓰기 요청을 구성하는 패킷의 모양입니다. 여기서 텍스트 패킷의 길이가 256 byte를 넘지 않도록 구성하여야 합니다.

읽기/쓰기 요청 1	;	읽기/쓰기 요청 2	;	읽기/쓰기 요청 3

모터 드라이브가 둘 이상의 읽기/쓰기 요청을 받아들이고 정상적으로 처리하였다면 다음과 같이 각각의 요청에 대한 응답을 구성하여 회신합니다. 각각의 응답을 구분하는 문자는 ';' 입니다.

읽기/쓰기 응답 1	;	읽기/쓰기 응답 2	;	읽기/쓰기 응답 3

만일 모터 드라이브가 구성한 응답 패킷의 길이가 256 byte보다 클 때는 256 byte를 초과하는 부분이 잘려나간 상태에서 오류를 표시하지 않고 응답하기 때문에 되도록 요청과 응답 패킷을 너무 길게 만들지 않는 것이 좋습니다.

예제) 모터 모터 드라이브의 **user_variable** 오브젝트의 sub-index 1,2,3 값을 동시에 쓰고자 할 때 아래와 같이 입력합니다.

```
u1=10;u2=20;u3=30↲
```

예제) 1번 체널의 디지털 입력과 2번 체널의 디지털 입력, 3번 체널의 디지털 입력을 동시에 읽고자 할 때는 다음과 같이 입력합니다.

```
div1;div2;div3↲
```

9 오브젝트 테이블

오브젝트는 모터 제어기를 외부에서 또는 내부의 Virtual Machine에서 엑세스 하는데 사용하도록 외부로 공개된 일종의 변수 목록입니다.

시리얼 통신(RS-232, RS-485)을 통해 오브젝트를 다음과 같이 TEXT 방식으로 엑세스 할 수 있습니다.

```
Ex) brake_on_delay = 100↲ // Brake on Delay 값을 100ms로 설정한다.
    brake_on_delay↲       // Brake on Delay 값을 읽어온다.
```

Script 언어에서는 오브젝트 이름 앞에 언더스코어('_')를 붙여 사용한다.

```
Ex) _brake_on_delay = 100 // Brake on Delay 값을 100ms로 설정한다.
    d = _brake_on_delay   // Brake on Delay 값을 변수 d로 읽어온다.
```

9.1 Object Table

Index	Sub-index	Type	Access	Short name	Long name
0x2001	0	I16	RO	vid	vendor_id
0x2002	0	I16	RO	pid	product_id
0x2003	0	I16	RO	swv	software_version
0x2004	0	I16	RO	hwv	hardware_version
0x2005	0	I32	RO	ss	system_status
0x2006	0	I32	WO	sc	system_control
0x2007	0	I32	RW	sa	system_argument
0x2008	0	F32	RO	pv	power_source_voltage
0x2009	0	F32	RO	pc	power_source_current
0x2010	0~7	S32	RW	na	device_name
0x2011	0	I32	RW	id	device_id
0x2012	0	I32	RW	sw	serial_watchdog
0x2014	0~1	I32	RW	sb	serial_baudrate
0x201A	0	I08	RW	un	units
0x201C	0	I08	RW	scm	startup_control_mode
0x2041	0	I08	RW	pi	pulse_input_mode

0x2042	0	I08	RW	pce	pulse_count_edge
0x2043	0	I08	RW	dp	dir_signal_polarity
0x2044	0	I08	RW	el	enable_signal_level
0x2048	0	I32	RW	gn	gear_numerator
0x2049	0	I32	RW	gd	gear_denominator
0x2080	0~15	I16	RW	ec	error_code
0x2101	0	I32	WO	c	control
0x2102	0	I32	RO	s	status
0x2103	0	I32	RO	f	fault
0x2105	0	F32	RO	te	temperature
0x2111	0	I32	WO	tp	target_position
0x2112	0	I32	WO	tv	target_velocity
0x2114	0	F32	WO	tc	target_current
0x2116	0	F32	WO	tvq	target_voltage
0x2119	0	F32	RW	tl	torque_limit
0x2121	0	I32	RO	pd	position_demand
0x2122	0	I32	RO	vd	velocity_demand
0x2124	0	F32	RO	cdq	current_demand
0x2126	0	F32	RO	vdq	voltage_demand
0x2131	0	I32	RW	p	position_actual
0x2132	0	I32	RO	v	velocity_actual
0x2134	0	F32	RO	cq	current_actual
0x2141	0	I32	RO	pe	position_error
0x2142	0	I32	RO	ve	velocity_error
0x2144	0	F32	RO	ceq	current_error
0x2151	0	I08	RW	sl	soft_limit_check
0x2152	0	I32	RW	n	min_position
0x2153	0	I32	RW	x	max_position

0x2161	0	F32	RW	rvo	rated_voltage
0x2162	0	F32	RW	rc	rated_current
0x2163	0	I32	RW	rv	rated_velocity
0x2164	0	F32	RW	mc	max_current
0x2165	0	I32	RW	mv	max_velocity
0x2167	0	I08	RW	pty	profile_type
0x2168	0	I32	RW	pve	profile_velocity
0x2169	0	I32	RW	pa	profile_acceleration
0x216A	0	I32	RW	pde	profile_deceleration
0x216B	0	I32	RW	pj	profile_jerk
0x2171	0	F32	RW	oh	overheat_limit
0x2172	0	F32	RW	oc	overcurrent_limit
0x2173	0	F32	RW	ov	overvoltage_limit
0x2174	0	F32	RW	uv	undervoltage_limit
0x2176	0	I08	RW	vb	vibration_detect
0x2177	0	I08	RW	sd	stall_current_detect
0x2178	0	I08	RW	pt	position_track_error
0x2179	0	I08	RW	vt	velocity_track_error
0x2181	0	I08	RW	mt	motor_type
0x2188	0	I08	RW	ps	position_sensor
0x2189	0	I08	RW	ed	encoder_direction
0x218A	0	I32	RW	er	encoder_resolution
0x218B	0	I08	RW	np	no_pole_pairs
0x218C	0	I08	RW	hs	hallsensor_phase
0x2191	0	F32	RW	b	bemf_constant
0x2192	0	F32	RW	r	resistance
0x2194	0	F32	RW	l	inductance
0x2198	0	F32	RW	ir	inertia_ratio
0x2199	0	F32	RW	i	moment_of_inertia
0x219A	0	F32	RW	vf	viscous_friction
0x219B	0	F32	RW	cf	coulomb_friction

0x21A1	0	F32	RW	pp	position_p_gain
0x21A4	0	F32	RW	vp	velocity_p_gain
0x21A5	0	F32	RW	vi	velocity_i_gain
0x21A9	0	F32	RW	cp	current_p_gain
0x21AA	0	F32	RW	ci	current_i_gain
0x21B1	0	I16	RW	pw	pmaf_window_size
0x21B2	0	I16	RW	vw	vmaf_window_size
0x21D1	0	I32	RW	ip	in_position_threshold
0x21D2	0	I32	RW	iv	in_velocity_threshold
0x21D5	0	I32	RW	rvc	regen_voltage_clamp

9.2 Product Information

9.2.1 vendor_id

0으로 고정된 값입니다.

```
ex)
vendor_id↵          // vendor_id를 입력하고 [Enter] 키를 입력
vendor_id=0         // 제어기에서 vendor_id=0으로 응답
```

9.2.2 product_id

모터 제어기 제품 ID입니다.

SBL24D200S-U의 제품 ID는 '1511' 입니다.

```
ex)
product_id↵
product_id=1511
```

9.2.3 software_version

모터 제어기 펌웨어 버전에 100을 곱한 값입니다.

펌웨어 버전은 다음과 같이 표시됩니다: XX.YY

- XX: major version
- YY: minor version

```
ex)
software_version↵
software_version=111        // Version 1.11 임을 나타냄
```

9.2.4 hardware_version

모터 제어기 하드웨어 버전에 100을 곱한 값입니다.

하드웨어 버전은 다음과 같이 표시됩니다: XX.YY

- XX: major version
- YY: minor version

9.3 System Parameters

9.3.1 system_status

모터 제어기의 전반적인 상태를 표시합니다. 모터 구동과 직접 관련된 상태는 status 오브젝트에서 표시합니다.

definition	code	description
SS_HS_PHASE_DETECT	0x0001	SC_HS_PHASE_DETECT 명령이 실행 중이거나 실행 되었음을 표시합니다.
SS_POS_SENSOR_PARAM	0x0002	SC_POS_SENSOR_PARAM 명령이 실행 중이거나 실행 되었음을 표시합니다.
SS_ELECTRIC_PARAM_EST	0x0003	SC_ELECTRIC_PARAM_EST 명령이 실행 중이거나 실행 되었음을 표시합니다.
SS_MECHANIC_PARAM_EST	0x0004	SC_MECHANIC_PARAM_EST 명령이 실행 중이거나 실행 되었음을 표시합니다.
SS_COMMAND_SUCCESS	0x0010	상기 SC_HS_PHASE_DETECT, SC_POS_SENSOR_PARAM, SC_ELECTRIC_PARAM_EST, SC_MECHANIC_PARAM_EST 명령이 성공적으로 실행 종료 되었음을 표시합니다.
SS_COMMAND_FAILED	0x0020	상기 SC_HS_PHASE_DETECT, SC_POS_SENSOR_PARAM, SC_ELECTRIC_PARAM_EST, SC_MECHANIC_PARAM_EST 명령이 수행 도중 또는 수행 완료 후 실패하였음을 표시합니다.
SS_COMMAND_ABORTED	0x0040	상기 SC_HS_PHASE_DETECT, SC_POS_SENSOR_PARAM, SC_ELECTRIC_PARAM_EST, SC_MECHANIC_PARAM_EST 명령이 사용자 요청에 의해 중단되었음을 표시합니다.

9.3.2 system_control

모터 제어기에 운용 명령을 내립니다. 모터 구동과 직접 관련된 명령은 control 오브젝트로 내립니다.

definition	code	description
SC_COMMAND_ABORT	10	현재 수행중인 SC_HS_PHASE_DETECT, SC_POS_SENSOR_PARAM, SC_ELECTRIC_PARAM_EST, SC_MECHANIC_PARAM_EST 명령을 중단합니다. system_status 오브젝트에는 SS_COMMAND_ABORTED 가 표시됩니다.
SC_HS_PHASE_DETECT	11	모터를 60도 각도 단위로 Micro-stepping 구동하여 Hallsensor의 연결 순서와 위상을 찾습니다. 찾은 값은 hallsensor_phase 오브젝트에 저장됩니다. 이 명령은 BLDC와 PMSM 모터에만 해당하며, 명령을 실행하였을 때 모터는 1바퀴 정도 회전하게 됩니다. 명령을 실행하기 전에 모터와 부하를 분리하여 모터만 구동할 것을 권장합니다. Hallsensor가 올바르게 연결되었는데도 탐지를 실패하는 경우가 발생하는데, 이 때는 몇 번 재시도 하면 찾게 됩니다. 이 명령을 실행하기 전, system_argument에 모터를 구동하기 위한 전류를 rated_current의 %값(10~100)으로 설정해야 합니다.
SC_POS_SENSOR_PARAM	12	모터를 Micro-stepping 구동 또는 Voltage 구동하여 위치 센서 (Hallsensor and/or Encoder) 정보를 찾습니다. 이 명령을 실행하면 position_sensor, encoder_direction, encoder_resolution, no_pole_pairs 오브젝트 값이 변경됩니다. 이 명령을 실행하면 모터는 일정 속도로 수~수십 바퀴 회전하게 되는데, 모터와 부하를 분리시킨 상태에서 구동할 것을 권장합니다. Linear motor에서는 soft_limit_check 옵션을 켜고 움직이는 최소 최대 범위(min_position, max_position)를 지정하여 모터가 스토퍼에 충돌하지 않도록 합니다. 이 명령을 실행하기 전, system_argument에 모터를 구동하기 위한 전압을rated_voltage의 %값(2~20)으로 설정해야 합니다.
SC_ELECTRIC_PARAM_EST	13	모터의 전기 파라미터(역기전력 상수(Ke), 권선 저항(R), 권선 d/q축 인덕턴스(Ld/Lq)) 추정을 시작합니다. 이 명령을 실행하면 bemf_constant, resistance, inductance_d,

		inductance_q, electric_angle_bias, torque_constant 오브젝트 값이 변경됩니다. 명령이 실행 중일 때 모터는 전기 파라미터를 찾기 위해 움직이게 되며, 모터와 부하를 분리시킨 상태에서 구동할 것을 권장합니다. Linear motor에서는 soft_limit_check 옵션을 켜고 움직이는 최소 최대 범위(min_position, max_position)를 지정하여 모터가 스토퍼에 충돌하지 않도록 합니다. 이 명령을 실행하기 전, system_argument에 모터를 구동하기 위한 전압을rated_voltage의 %값(10~30)으로 설정해야 합니다.
SC_MECHANIC_PARAM_EST	14	모터의 기계 파라미터(기구부의 관성 모멘트(J), 속도에 따른 마찰 계수(B, C)) 추정을 시작합니다. 이 명령을 실행하면 moment_of_inertia, viscous_friction, coulomb_friction 오브젝트 값이 변경됩니다. Linear motor에서는 soft_limit_check 옵션을 켜고 움직이는 최소 최대 범위(min_position, max_position)를 지정해야 합니다. 이 명령을 실행하기 전에 system_argument에 기구부를 움직이는 속도를 rated_velocity의 %값(10~100)으로 설정해야 합니다. 이 명령은 모터와 기구부에 대한 기계 파라미터를 찾기 때문에 모터와 기구부가 연결되어 있어야 합니다. 명령이 실행 중일 때 모터가 급격하게 가감속 운동하게 됨으로 기계 시스템의 강성을 고려하여 %값을 순차적으로 증가시키면서 수행할 것을 권장합니다. ***주의: 이 명령을 실행하면 모터가 급격히 가감속 운동을 하기 때문에 진동과 소음이 발생 합니다. 진동과 소음이 발생하지 않는 경우 다시 한번 실행해 주시면 됩니다.**
SC_CC_GAIN_STIFFNESS	41	d축과 q축 전류 제어기의 이득에 대한 stiffness를 0에서 100% 사이의 값으로 설정 합니다. 이 값은 d축과 q축 전류 제어기의 P, I 이득을 동시에 조정하게 되며, 높은 값을 설정하면 전류제어 응답성이 좋아지지만 모터가 진동하거나 발산할 수 있습니다. 이 명령을 실행하면 d축과 q축 전류 제어기 이득을 결정하는 cc_kp, cc_ki오브젝트의 값이 변경됩니다. 명령을 실행하기 전에 stiffness 값을 system_argument에 설정해야 합니다.

		***주의: 전류 제어기의 이득이 높으면 가청 주파수 대의 '삐~~'하는 소음이 들립니다. 전류 제어기의 이득은 소음이 들리지 않는 정도에서 설정합니다.**
SC_PV_GAIN_STIFFNESS	42	위치와 속도 제어기의 이득에 대한 stiffness를 0에서 100% 사이의 값으로 설정 합니다. 이 값은 위치 제어기의 P 이득과 속도 제어기의 P, I 이득을 동시에 조정하게 되며, 높은 값을 설정하면 전류제어 응답성이 좋아지지만 모터가 진동하거나 발산할 수 있습니다. 이 명령을 실행하면 pc_kp, vc_kp, vc_ki 오브젝트의 값이 변경됩니다. 명령을 실행하기 전에 stiffness 값을 system_argument에 설정해야 합니다. ***주의: 위치와 속도 제어기의 이득이 높으면 모터가 낮은 주파수로 진동합니다. 모터의 진동이 발생하지 않도록 설정해야 합니다.**
SC_CC_GAIN_STIF_READ	81	제어기에 설정된 전류 제어기의 이득에 대한 stiffness를 읽어옵니다. 읽은 값은 system_argument에 저장됩니다.
SC_PV_GAIN_STIF_READ	82	제어기에 설정된 위치와 속도 제어기의 이득에 대한 stiffness를 읽어옵니다. 읽은 값은 system_argument에 저장됩니다.
SC_FWC_SPEED_UP_READ	83	제어기에 설정된 speed up 값을 읽어옵니다. 읽은 값은 system_argument에 저장됩니다.
SC_CLEAR_ERROR_CODE	90	error_code 오브젝트 배열의 에러 코드를 모두 삭제합니다.
SC_FACTORY_DEFAULT	98	모터 제어기의 설정과 관련된 모든 오브젝트들을 공장 출하 시 기본 값으로 초기화 합니다.
SC_SYSTEM_RESET	99	모터 제어기를 소프트웨어 리셋 합니다. ***주의: 모터 파라미터를 변경한 경우 모터 구동을 멈추고 5초 정도 지나야 변경된 파라미터가 Flash Memory에 저장됩니다.**
SC_FIRMWARE_DOWNLOAD	9090	제어기를 Firmware Download 모드로 진입합니다. Firmware는 Y-mode 프로토콜로 전송할 수 있으며, 전송 도중 실패하거나 중단한 경우 제어기는 계속해서 Firmware Download 모드에 남아있게 됩니다. 이 상태는 제어기에 올바른 펌웨어 다운로드를 완료하였을 때 빠져나올 수 있습니다.

9.3.3 system_argument

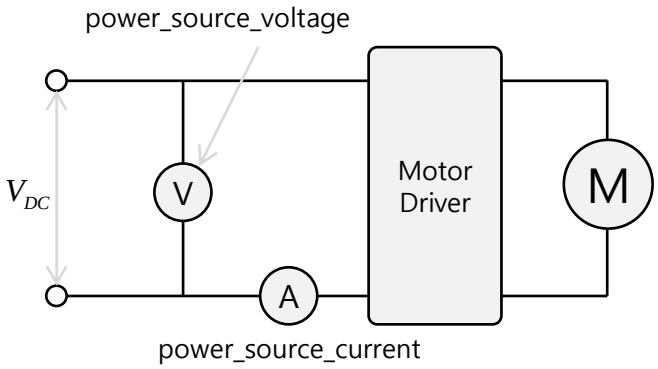
system_control 명령어 중 argument를 필요로 하는 명령어가 있습니다. SC_HS_PHASE_DETECT 명

령어의 경우, 명령어를 실행하기 전에 홀센서 탐지 전류 값을 지정해야 하는데, system_argument 에 홀센서 탐지 전류 값을 %로 지정하고 명령을 실행하게 됩니다.

```
ex)
system_argument = 40↵
system_control = 11↵
```

9.3.4 power_source_voltage

전원단에 공급되는 전압을 측정합니다.



9.3.5 power_source_current

전원단에 흐르는 전류를 계산합니다. 이 값은 실제 측정된 값이 아니라 모터에 흐르는 전류로부터 계산된 값입니다.

power_source_current = [제어기의 기본 소비 전력]/power_source_voltage + [PWM Duty ratio]*[모터에 흐르는 전류]

9.3.6 units

모터 제어기에서 사용할 단위의 종류를 선택합니다.

값	길이, 속도, 가속도 (velocity, acceleration, jerk)	각종 상수 (bemf_constant, viscous_friction)	힘, 토크 (coulomb_friction)	관성, 무게 (moment_of_inertia)
0	count/s, count/s^2, count/^3	mV/(count/s), mNm/(count/s)	mNm, mNm/A	g.m^2

1	rpm, rpm/s, rpm/s^2	mV/rpm, mNm/rpm	mNm, mNm/A	g.m^2

ex)
units↵
units=1

9.4 Connection

9.4.1 device_name

모터 제어기의 이름을 설정합니다. 설정 가능한 문자열은 최대 32byte입니다.

ex)
device_name=BLDC_2P↵ // Device Name을 BLDC_2P로 설정함
device_name↵ // Device Name을 읽어옴

device_name은 모터의 구동이나 연결에는 영향을 미치지 않으며, 단지 사용자가 모터 제어기를 쉽게 구분하도록 이름을 부여하는 기능입니다.

9.4.2 device_id

RS-485, CAN과 같이 여러 개의 제어기가 하나의 통신 bus에 연결된 상태에서 호스트와 제어기와 통신하는 경우, 각각의 제어기는 서로 다른 device_id를 가지도록 설정해야 합니다. device_id는 0에서 255사이 값으로 설정합니다. 기본 설정 값은 1입니다.

RS-485 통신으로 연결된 제어기에 명령을 내리는 경우, 다음과 같이 device_id를 사용하지 않고 명령을 내릴 수 있습니다. 하지만 RS-485 버스에 한 개 이상의 제어기가 연결되어 있다면 연결된 모든 제어기가 응답하게 됨으로 올바른 응답을 받을 수 없게 됩니다.

ex)
software_version↵ // 소프트웨어 버전 읽기 명령을 내림
????????????????????? // 여러 제어기가 명령에 응답하여 수신 문자열이 깨짐

상기와 같은 경우, 다음과 같이 device_id를 명령 앞에 사용하여 해당 제어기에만 명령을 내리고 응답 받을 수 있습니다.

ex)
1:software_version↵ // device_id가 1인 제어기에 명령을 내림

```
1:software_version=111      // device_id가 1인 제어기가 응답 함
```

9.4.3 serial_watchdog

RS-232, RS-485 통신에서 모터를 구동하는 명령에 대한 와치독 타임아웃 값을 설정합니다.

만일 설정된 시간 이내에 모터 구동 명령이 수신되지 않으면 모터를 정지하게 됩니다.

모터가 정지된 상태에서 새로운 구동 명령이 내려지면, 제어기는 구동 명령을 수행합니다.

이 값이 0으로 설정된 경우, 와치독 타이머는 작동하지 않습니다.

9.4.4 serial_baudrate

시리얼(RS-232, RS-485) 통신의 속도를 설정합니다. RS-232, RS-485의 경우 다음과 같은 통신 속도를 설정 가능합니다.

- 9600
- 19200
- 38400
- 57600
- 115200 (기본값)
- 230400
- 460800
- 921600

rs232_baudrate, rs485_baudrate 오브젝트는 다음과 같이 구분하여 사용할 수 있습니다.

```
ex)
rs232_baudrate=115200↵            // RS-232의 속도를 설정
rs485_baudrate=115200↵            // RS-485의 속도를 설정
```

9.5 Startup

9.5.1 startup_control_mode

제어기가 시작될 때 초기 제어 모드를 설정합니다.

- 0 – Disable
- 1 – Enable
- 2 – Enable and Current Control Mode

- 3 – Enable and Position Control Mode
- 4 – Enable and Velocity Control Mode

이 설정 값은 모터 제어기를 Enable 할 때 모터의 구동 모드를 결정할 때도 사용됩니다. 0이나 1로 설정된 경우 Voltage Output 모드로 Enable 되고, 2으로 설정된 경우 Current Control 모드가 됩니다. 3로 설정된 경우는 Velocity Control 모드가 되고, 4로 설정된 경우 Position Control 모드가 됩니다.

9.6 Pulse/Direction Input

Pulse Input 그룹에는 펄스 입력 신호(PUL+, PUL-, DIR+, DIR-, ENA+, ENA-)의 운영 방법을 설정 합니다.

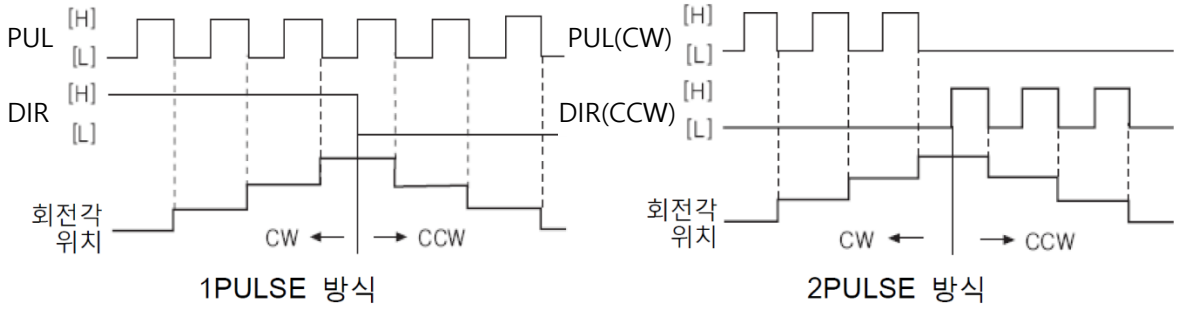
9.6.1 pulse_input_mode

펄스 입력 신호의 펄스 방식을 선택합니다.

- 0 – One Pulse mode (기본값)
- 1 – Two Pulse mode

(1) One pulse mode인 경우는 제어 입력으로 Pulse와 Direction 입력을 받아들입니다. Pulse는 구동 펄스로 카운트 되고 Direction은 카운터의 방향이 됩니다.

(2) Two Pulse mode인 경우는 제어 입력으로 CW pulse와 CCW pulse 입력을 받아들입니다. CW pulse는 정 방향 구동 펄스로 카운트 되고 CCW pulse는 역방향 구동 펄스로 카운트 됩니다.



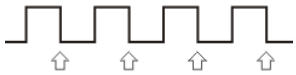
9.6.2 pulse_count_edge

Pulse 입력 신호의 카운트 시점을 결정합니다.

- 0 - Falling Edge: Pulse 입력의 하강 모서리에서 펄스 카운트 (기본값)
- 1 - Rising Edge: Pulse 입력의 상승 모서리에서 펄스 카운트

이 값을 0으로 설정한 경우, 아래 그림과 같이 펄스가 카운트 되는 시점은 Pulse 입력의 하강 모

서리에서 카운트 됩니다.



9.6.3 dir_signal_polarity

Direction 입력 신호의 상태에 따라, 카운트 방향에 대한 논리 레벨을 결정합니다.

- 0 - Positive Low: Direction 입력이 0일 때 정 방향 펄스 카운트 (기본값)
- 1 - Positive High: Direction 입력이 1일 때 정 방향 펄스 카운트

9.6.4 enable_signal_level

Enable 입력 신호의 상태에 따라, 모터의 Enable/Disable 논리 레벨을 결정합니다.

- 0 - Active Low: Enable 입력이 0일 때 모터 Enable (기본값)
- 1 - Active High: Enable 입력이 1일 때 모터 Enable

9.6.5 gear_numerator

펄스 입력에 대해 전자 기어비를 적용합니다.

이 값은 전자 기어의 분자 값이 됩니다. 다음 식을 이용하여 펄스 입력 신호의 펄스를 카운트 하게 됩니다.

$$PulseCounter = \frac{\text{gear_numerator}}{\text{gear_denominator}} \times InputPulse$$

9.6.6 gear_denominator

펄스 입력에 대해 전자 기어비를 적용합니다.

이 값은 전자 기어의 분모 값입니다.

9.7 Error Code

9.7.1 error_code

error_code의 Sub-index 0에는 모터 제어기가 시작되고 현재까지 발생한 Error Code의 개수가 설정됩니다. Sub-index 1~15 까지는 Error Code가 발생한 순서대로 기록됩니다.

만일 error_code[0]의 값이 20이라면 error_code[1 ~ 15]에는 최근에 발생한 에러코드 15개만 기록되며, 나머지 5개는 뒤로 밀려나 지워집니다.

다음은 Error Code의 각 코드 값의 범위에 대한 메시지의 종류입니다.

- 1000 ~ 1999: Error
- 2000 ~ 2999: Warning
- 3000 ~ 3999: Information

Error는 보통 모터 구동 중 모터를 더 이상 정상적으로 구동할 수 없는 상황에서 발생되며, Error가 발생하면 fault 플래그가 설정되고 제어기는 Disable 상태가 됩니다. 반면 Warning은 각종 제약이나 제어기 상태에 의해 사용자가 내린 명령을 수행할 수 없을 때 발생합니다. Information은 단지 사용자에게 정보를 전달하기 위해 사용됩니다.

다음 표는 각 코드 값에 대한 설명입니다.

Error Code	Description
1002	Hallsensor의 Sector 정보를 읽어올 수 없습니다.
1003	Encoder Resolution이 설정되지 않았습니다.
1004	Hallsensor Phase가 설정되지 않았습니다.
1005	Hallsensor의 Number of Pole Pairs가 설정되지 않았습니다.
1006	Stall Current(모터가 힘을 발생하여도 회전하지 않음)가 탐지되었습니다.
1007	Position Controller에서 Position Error가 설정된 값을 초과하였습니다.
1008	Velocity Controller에서 Velocity Error가 설정된 값을 초과하였습니다.
1009	Encoder를읽을수없습니다. Encoder 연결을확인하십시오.
1010	Motor에 전압을 가하여도 전류가 흐르지 않습니다. Motor의 결선을 확인하십시오.
1011	Motor의 단자가 전원과 단락 되었을 가능성이 있습니다.
1012	Motor의 q축 전류가 Overcurrent limit을 초과하였습니다.
1013	Motor의 d축 전류가 Overcurrent limit을 초과하였습니다.
1014	전원의 전압이 Undervoltage limit 아래로 떨어졌습니다.
1015	전원의 전압이 Overvoltage limit을 초과하였습니다.
1016	FET와 Heatsink의 온도가 Overheta limit을 초과하였습니다.
1017	모터에 흐르는 전류에서 진동이 탐지되었습니다.
1018	모터의 속도가 Max Velocity의 150%를 초과하였습니다.
1019	Hallsensor의 Number of Pole Pairs에 비해 Encoder Resolution이 잘못 설정되었습니다.
1020	Script Code 수행도중 Stack overflow가발생하였습니다.
1021	잘못된 Script Code가 실행되었습니다.

1022	모터의 위치가 Positive Limit 위치를 벗어났습니다.
1023	모터의 위치가 Negative Limit 위치를 벗어났습니다.
1024	Emergency Switch가 켜졌습니다.
1025	모터의 전기 파라미터(R, Ld, Lq, Ke)에 잘못된 값이 설정되었습니다.
1026	모터의 기계 파라미터(J, B, C, Tl)에 잘못된 값이 설정되었습니다.

Error Code	Description
2001	System Command가 수행중인 상태에서 Motor 제어 명령을 수행할 수 없습니다.
2002	Motor가 Disable 상태에서 제어 명령을 수행할 수 없습니다.
2003	Digital Input이나 Serial Command에 의한 Motor Stop 상태에서 제어 명령을 수행할 수 없습니다.
2004	Forward Limit 상태에서 제어 명령을 수행할 수 없습니다.
2005	Reverse Limit 상태에서 제어 명령을 수행할 수 없습니다.
2006	Serial 포트의 watchdog timer가 타임아웃 되었습니다. Motor에 Stop 명령을 내립니다.
2007	Factory default 값들을 제어기에 적용합니다.
2008	Motor의 position이 Software forward limit에 도달하였습니다. Motor를 stop 합니다.
2009	Motor의 position이 Software reverse limit에 도달하였습니다. Motor를 stop 합니다.
2010	Digital Input에의한Motor Disable(Emergency Stop) 상태에서제어명령을수행할수없습니다.
2012	Motor의 q축 전류가 Overcurrent limit을 초과한 상태에서 제어 명령을 수행할 수 없습니다.
2013	Motor의 d축 전류가 Overcurrent limit을 초과한 상태에서 제어 명령을 수행할 수 없습니다.
2014	전원의 전압이 Undervoltage limit 미만인 상태에서 제어 명령을 수행할 수 없습니다.
2015	전원의 전압이 Overvoltage limit을 초과한 상태에서 제어 명령을 수행할 수 없습니다.
2016	FET와 Heatsink의 온도가 Overheta limit을 초과한 상태에서 제어 명령을 수행할 수 없습니다.
2017	Quick stop 상태에서 제어 명령을 수행할 수 없습니다.
2019	Motor가 움직이고 있을 때는 Script를 다운로드 할 수 없습니다.
2020	Script가 실행중일 때는 Script를 다운로드 할 수 없습니다.
2021	Script의 다운로드 size가 잘못 설정되었습니다.
2022	Script의 address가 Script의 size를 벗어나 설정되었습니다.
2023	Script의 code가 size를 벗어나 다운로드 되었습니다.
2024	실행할 수 없는 Script가 적재되어 있습니다.
2030	Stall Current Detection 옵션이설정되지않았습니다. Home Search를시작할수없습니다.
2031	Digital Input에서Home Switch가설정되지않았습니다. Home Search를시작할수없습니다.
2032	Digital Input에서Forward Limit Switch가설정되지않았습니다. Home Search를시작

	할수없습니다.
2033	Digital Input에서Reverse Limit Switch가설정되지않았습니다. Home Search를시작할 수없습니다.
2034	Encoder가Index 신호를가지고있지않습니다. Home Search를시작할수없습니다.
2035	DC 모터에서는 Hallsensor Phase 찾기를 수행할 수 없습니다.
2036	Hallsensor의 Sector 정보를 읽어올 수 없습니다.

Error Code	Description
3001	Motor Driver가 시작되었습니다.
3002	Motor가 Home position 탐색을 완료하였습니다.
3003	Motor의position을Motion table에저장하였습니다.
3004	Motion table을시작합니다.
3005	Motion table을종료합니다.
3006	Script를시작합니다.
3007	Script를종료합니다.
3008	Motor의 position을 새로운 값으로 설정 합니다.
3010	Notch filter 1의 파라미터를 설정하였습니다.
3011	Notch filter 2의 파라미터를 설정하였습니다.
3012	Notch filter 3의 파라미터를 설정하였습니다.

9.8 Motor Status

9.8.1 control

제어기에 모터의 구동과 관련된 제어 명령을 내립니다.

definition	code	description
MC_DISABLE	0	모터에 공급되는 전원을 차단합니다.
MC_ENABLE	1	모터에 전원을 공급하여 구동 준비상태가 됩니다. Position/Velocity/Current/Voltage 명령을 수신하고 모터를 명령에 따라 구동합니다. Enable 될 때 startup_control_mode 의 설정에 따라 Position, Velocity, Current, Voltage 모드 중 하나의 제어 모드가 됩니다.
MC_CLEAR_FAULT	2	제어기에 발생한 fault 오브젝트 비트를 모두 지웁니다.
MC_STOP	3	모터를 감속하여 정지합니다.

9.8.2 status

제어기의 상태를 표시합니다.

definition	code	description
ST_ENABLE	0x0001	모터에 전원이 공급되고 명령을 수신하여 모터를 구동 가능한 상태입니다
ST_FAULT	0x0002	Fault가 발생하였으며, 모터는 Disable 됩니다. 발생한 폴트의 상세한 정보는 fault 오브젝트에서 확인 가능합니다
ST_MOVING	0x0004	모터가 구동 중인 상태입니다
ST_POSITION_CTL	0x0020	위치 구동 모드: 위치 제어 중인 상태입니다
ST_VELOCITY_CTL	0x0040	속도 구동 모드: 속도 제어 중인 상태입니다
ST_CURRENT_CTL	0x0080	전류 구동 모드: 전류 제어 중인 상태입니다
ST_POS_LIMIT	0x0100	모터의 위치가 정방향 또는 역방향 리미트에 도달했습니다.
ST_CURRENT_LIM	0x0200	모터의 전류가 Max Current 근처에 도달했습니다
ST_IN_POSITION	0x0400	모터의 위치가 목적 위치에 도달했습니다
ST_IN_VELOCITY	0x0800	모터가 목표 속도에 도달했습니다
ST_PULSE_DRIVE	0x00040000	Pulse/Direction 신호에 의해 제어기에 명령이 내려지는 중입니다
ST_SERIAL_DRIVE	0x00080000	시리얼 포트(RS-232, RS-485, CAN, Ethernet) 로부터 제어기에 명령이 내려지는 중입니다
ST_READY	0x20000000	모터 Disable 상태에서 Enable 준비가 되었음을 표시합니다.

9.8.3 fault

제어기에 발생한 폴트 상태를 표시합니다.

definition	code	description	error_code
FF_OVER_CURRENT	0x0001	모터에 허용된 한계전류 이상이 흐름 (관련 오브젝트: overcurrent_limit)	1012, 1013
FF_OVER_VOLTAGE	0x0002	제어기의 전원으로부터 허용된 한계전압 이상이 가해짐 (관련 오브젝트: overvoltage_limit)	1015
FF_UNDER_VOLTAGE	0x0004	제어기의 전원으로부터 허용된 한계전압 이하로 떨어짐 (관련 오브젝트: undervoltage_limit)	1014
FF_OVER_HEAT	0x0008	제어기의 FET와 방열판에서 허용된 한계온도 이상으로 온도가 올라감 (관련 오브젝트: overheat_limit)	1016
FF_SHORT_CIRCUIT	0x0010	모터 단자에 과전류가 흐름, 모터 단자와 전원 단자간 단락이 의심됨	1011
FF_VIBRATION	0x0020	모터 구동 중 진동이 탐지 됨 (관련 오브젝트: vibration_detect)	1017
FF_STALL_CURRENT	0x0040	모터가 멈춘 상태에서 설정된 시간 동안 설정된 값 이상의 전류가 흐름 (관련 오브젝트: stall_current_detect)	1006
FF_POSITION_TRACKING	0x0080	위치 제어기에서 설정된 시간 동안 위치 오차가 설정된 값을 초과 함 (관련 오브젝트: position_track_error)	1007
FF_VELOCITY_TRACKING	0x0100	속도 제어기에서 설정된 시간 동안 속도 오차가 설정된 값을 초과 함 (관련 오브젝트: velocity_track_error)	1008
FF_HALLSENSOR	0x0200	Hallsensor 입력이 들어오지 않거나 비정상적인 조합의 Hallsensor 입력이 들어옴	1002, 1004, 1005
FF_ENCODER	0x0400	Encoder 입력이 들어오지 않거나, 한 바퀴당 Hallsensor 카운트 값과 Encoder 카운트 값이 설정 값과 일치하지 않음	1003, 1009
FF_MOTOR	0x0800	모터 단자에 전류가 흐르지 않음, 모터 연결이 끊어진 상황이 의심됨	1001, 1010
FF_OVER_SPEED	0x1000	모터의 속도가 정격 속도(rated_velocity)의 200%	1018

		이상으로 회전함	
FF_ELECTRIC_PARAM	0x20000	Electric Parameter (R, L, Ke) 설정 오류	1025
FF_MECHANIC_PARAM	0x40000	Mechanic Parameter (J, B, C) 설정 오류	1026
FF_INTERNAL_VARIABLE	0x80000	내부 변수의 계산 오류	

9.8.4 temperature

FET와 방열판의 현재 온도 값입니다. 이 값이 overheat_limit 설정 값 이상으로 올라가면 제어기를 보호하기 위해 폴트를 발생하고 모터는 Disable 상태가 됩니다.

9.9 Motor Drive

모터 구동 명령은 다음 중 하나의 입력 포트를 통해 입력될 수 있습니다:

- Pulse Input mode - Pulse/Direction 신호에 의해 제어기에 명령이 내려짐
- Serial Input mode - 시리얼 포트(USB, RS-232, RS-485) 로부터 제어기에 명령이 내려짐

9.9.1 target_position

모터가 도달해야 할 목표 위치를 지정합니다. 제어기가 Enable 상태에 있다면, 이 값은 변경 즉시 적용됩니다.

속도 프로파일을 사용하지 않는 경우(profile_type = 0), Moving Average Filter를 거쳐 위치 명령이 완만하게 변하도록 셰이핑 할 수 있습니다. 속도 프로파일을 사용하는 경우, Velocity Profile Generation을 거쳐 속도 프로파일의 형태를 사다리꼴(Trapezoidal profile) 또는 사인파(Sinusoidal profile) 형태가 되도록 할 수 있습니다.

만일 제어기가 속도 제어나 전류 제어 상태에서 구동 중일 때 이 값을 변경하면, 제어기는 위치 제어 상태로 변경되고 현재 속도와 위치를 고려하여 target_position에 지정된 위치로 이동을 시작합니다.

9.9.2 target_velocity

모터가 도달해야 할 목표 속도를 지정합니다. 제어기가 Enable 상태에 있다면, 이 값은 변경 즉시 적용됩니다.

속도 프로파일을 사용하지 않는 경우(profile_type = 0), Moving Average Filter를 거쳐 속도 명령이 완만하게 변하도록 셰이핑 할 수 있습니다. 속도 프로파일을 사용하는 경우, Velocity Profile Generation을 거쳐 속도 프로파일의 형태를 사다리꼴(Trapezoidal profile) 또는 사인파(Sinusoidal

profile) 형태가 되도록 할 수 있습니다.

만일 제어기가 위치 제어나 전류 제어 상태에서 구동 중일 때 이 값을 변경하면, 제어기는 속도 제어 상태로 변경되고 현재 속도를 고려하여 target_velocity에 지정된 속도로 가속이나 감속하여 목표 속도에 도달하게 됩니다.

9.9.3 target_current

모터가 도달해야 할 q축 목표 전류를 지정합니다. 제어기가 Enable 상태에 있다면, 이 값은 변경 즉시 적용됩니다.

만일 제어기가 위치 제어나 속도 제어 상태에서 구동 중일 때 이 값을 변경하면, 제어기는 전류 제어 상태로 변경되고 target_current에 지정된 전류를 모터에 흐르도록 제어합니다.

q축 전류 제어기가 동작하는 동안 d축 전류 제어기는 0이 되도록 제어합니다.

9.9.4 target_voltage

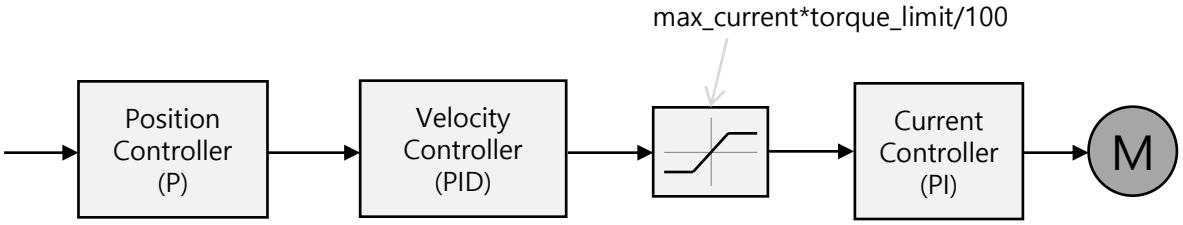
모터에 가해지는 q축 전압을 지정합니다. 제어기가 Enable 상태에 있다면, 이 값은 변경 즉시 적용됩니다.

만일 제어기가 위치 제어나 속도 제어, 전류 제어 상태에서 구동 중일 때 이 값을 변경하면, 제어기는 모든 제어 상태를 해제하고 target_voltage에 지정된 전압을 모터에 가합니다.

q축 전압이 직접적으로 출력되는 경우, d축 전압은 0으로 출력됩니다.

9.9.5 torque_limit

위치 제어나 속도 제어시 전류의 최대 값을 제한함으로 제어기의 토크 리미트를 %단위로 설정합니다.



torque_limit 오브젝트 값은 모터가 Enable 되면서 100%로 재설정 됩니다.

9.9.6 position_demand

위치 제어기가 추종해야 할 demand 값입니다. position_demand 값은 목표 위치(target_position)에 도달하기 위한 위치 경로 값으로 속도 프로파일 생성기에 의해 만들어집니다.

position_error = position_demand – position_actual

9.9.7 velocity_demand

속도 제어기가 추종해야 할 demand 값입니다. velocity_demand 값은 목표 속도(target_velocity)에 도달하기 위한 속도 경로 값으로 속도 프로파일 생성기에 의해 만들어지고, 여기에 위치 제어기의 출력 값이 더해집니다.

velocity_error = velocity_demand – velocity_actual

9.9.8 current_demand

q축 전류 제어기가 추종해야 할 demand 값입니다. 만일 위치 제어기나 속도 제어기가 동작 중이라면 속도 제어기의 출력 값이 되고, 전류 제어기만 동작 중이라면 목표 전류(target_current) 값으로 설정됩니다.

current_error = current_demand – current_actual

9.9.9 voltage_demand

모터에 가해지는 q축 전압 값입니다. 만일 전류 제어기가 동작 중이라면 전류 제어기의 출력 값이 되고, 목표 전압(target_voltage)이 직접 설정되었다면 목표 전압 값이 됩니다.

9.9.10 position_actual

Encoder나 Hallsensor로부터 읽은 카운트 값으로 모터의 현재 위치에 해당하는 값입니다.

9.9.11 velocity_actual

position_actual을 미분한 값입니다. 이 값은 Encoder나 Hallsensor로부터 읽은 위치 값을 속도 추정기를 거쳐 추정한 속도라서 실제 모터의 속도와 다를 수 있습니다.

velocity_actual의 절대값이 max_velocity의 150%를 초과하면 FF_OVER_SPEED 폴트를 발생시킵니다.

9.9.12 current_actual

모터에 흐르는 q축 전류 값입니다.

전류 값이 overcurrent_limit 설정 값을 초과하면 FF_OVER_CURRENT 폴트를 발생시킵니다.

9.9.13 position_error

위치 제어기에 입력되는 값입니다.

position_error = position_demand – position_actual

위치 제어 모드에서 position_error 값이 position_track_error 오브젝트로 설정한 조건을 벗어나게 되면 FF_POSITION_TRACKING 폴트를 발생시킵니다.

9.9.14 velocity_error

속도 제어기에 입력되는 값입니다.

velocity_error = velocity_demand – velocity_actual

속도 제어 모드에서 velocity_error 값이 velocity_track_error 오브젝트로 설정한 조건을 벗어나게 되면 FF_VELOCITY_TRACKING 폴트를 발생시킵니다.

9.9.15 current_error

q축 전류 제어기에 입력되는 값입니다.

current_error = current_demand – current_actual

9.10 Position Limit

모터 구동 중 사용자의 Position/Velocity/Current/Voltage 명령에 의해 모터의 위치가 최소/최대 리미트를 벗어나는 경우 모터는 정지하게 됩니다. 하지만 리미트 범위 내로 진입하는 명령은 수행 가능합니다.

9.10.1 soft_limit_check

소프트웨어 위치 리미트 사용 여부를 결정합니다.

- 0 – 소프트웨어 위치 리미트 사용하지 않음
- 1 – 소프트웨어 위치 리미트 사용 함

***주의: 소프트웨어 위치 리미트는 Pulse 입력 모드(ST_PULSE_DRIVE)에서는 적용되지 않습니다.**

9.10.2 min_position

소프트웨어 위치 리미트의 최소 값입니다. soft_limit_check가 1로 설정되고 position_actual 값이 min_position보다 작아진 경우 모터는 정지합니다.

이 값은 Pulse/Direction Input 모드에서는 적용되지 않습니다.

9.10.3 max_position

소프트웨어 위치 리미트의 최대 값입니다. soft_limit_check가 1로 설정되고 position_actual 값이 max_position보다 커진 경우 모터는 정지합니다.

이 값은 Pulse/Direction Input 모드에서는 적용되지 않습니다.

9.11 Rated Values

9.11.1 rated_voltage

모터의 정격 전압을 설정합니다. rated_voltage는 모터 구동을 위해 모터에 가하는 최대 전압입니다. 모터에 내려지는 전압 명령의 절대값은 rated_voltage 값보다 커질 수 없습니다.

※ Power supply에서 공급하는 전압은 rated_voltage 설정 값보다 약 20% 정도 높은 전압을 공급하는 것이 좋습니다. 만일 정격 24V 모터를 최고 속도에서 최대 토크로 구동하고자 할 때 전압 여유가 없다면 모터가 최고 속도에서 토크를 제대로 내지 못하게 됩니다.

9.11.2 rated_current

모터의 정격 전류를 설정합니다. 부하가 있는 상태에서 모터를 가감속 하여 구동할 때 모터에 연속으로 흘릴 수 있는 전류입니다. 모터에 내려지는 전류 명령의 절대값은 rated_current 값보다 커질 수 없습니다.

※ rated_current는 Velocity Profile의 profile_acceleration과 profile_deceleration을 계산하는데 사용됩니다. rated_current 값에 비례하여 가속도와 감속도 값이 설정됩니다.

9.11.3 rated_velocity

모터의 정격 회전 속도를 설정합니다. 부하가 있는 상태에서 모터를 연속 구동할 때의 속도가 됩니다. 모터에 내려지는 속도 명령의 절대값은 rated_velocity 값보다 커질 수 없습니다.

9.11.4 max_current

모터에 순간적으로 흘릴 수 있는 최대 전류입니다. 전류 제어기는 모터에 흐르는 q축 전류가 max_current를 넘지 않도록 제한합니다.

※ 일반적으로 max_current는 rated_current의 150% ~ 300% 값으로 설정합니다.

9.11.5 max_velocity

무부하 상태에서 모터에 rated_voltage가 가해질 때의 속도입니다.

모터의 회전 속도가 max_velocity의 150%를 초과하면 FF_OVER_SPEED fault를 발생시킵니다.

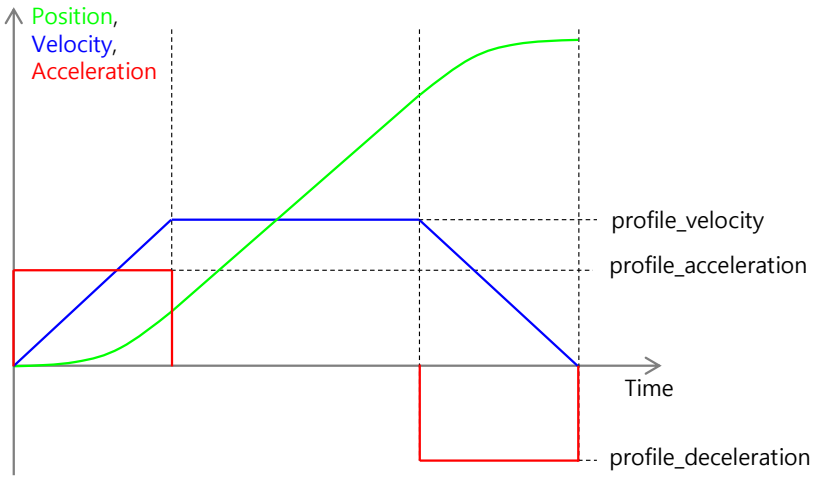
9.12 Velocity Profile

9.12.1 profile_type

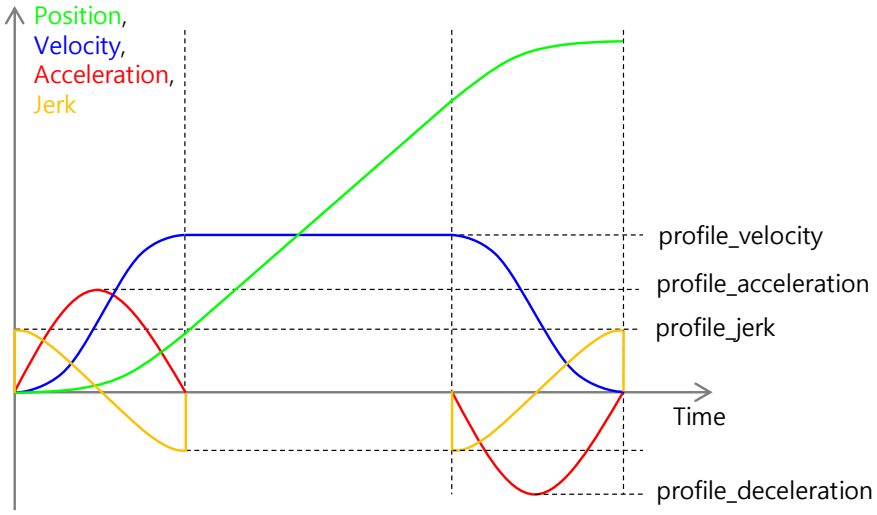
제어기의 위치와 속도 명령이 내려질 때 목적 위치나 속도로 이동하기 위한 프로파일 형상을 결정합니다.

- 0 - None
- 1 – Trapezoidal (기본값)
- 2 - Sinusoidal

다음은 사다리꼴 형태의 속도 프로파일을 사용할 때의 위치와 속도, 가속도 그래프입니다.



다음은 사인파 형태의 속도 프로파일을 사용할 때의 위치와 속도, 가속도, 저크 그래프입니다.



9.12.2 profile_velocity

속도 프로파일을 사용할 때 적용되는 최고 속도 값입니다.

이 값을 rated_velocity와 같거나 작게 설정해야 합니다.

9.12.3 profile_acceleration

속도 프로파일을 사용할 때 적용되는 가속도 값입니다.

이 값은 다음 식으로부터 계산된 값으로 설정하는 것을 권장합니다.

$$\text{profile_acceleration} = \frac{\text{torque_constant} \times \text{rated_current}}{\text{moment_of_inertia}}$$

하지만 가감속 구간의 시간이 짧고 큰 토크가 필요하다면 다음과 같이 설정합니다.

$$\text{profile_acceleration} = \frac{\text{torque_constant} \times \text{max_current}}{\text{moment_of_inertia}}$$

※ 계산시 단위 변환에 유의해야 합니다.

※ 상기 식은 다음 식으로부터 유도됩니다: $J\alpha = K_T i$

9.12.4 profile_deceleration

속도 프로파일을 사용할 때 적용되는 감속도 값입니다.

이 값도 profile_acceleration 과 마찬가지로 설정되어야 합니다.

9.12.5 profile_jerk

속도 프로파일을 사용할 때 적용되는 저크 값입니다.

이 값은 Sinusoidal 프로파일에서만 사용됩니다.

9.13 Fault Limits

9.13.1 overheat_limit

FET와 방열판의 온도(temperature)가 overheat_limit에 설정된 값을 넘어가면 FF_OVER_HEAT fault를 발생합니다. 보통 이 값은 제어기의 구동부 소자(FET)가 정상 동작 가능한 온도 범위 내에서 설정되며, FET와 방열판의 온도가 overheat_limit 전류 이상으로 올라가면 제어기를 보호하기 위하여 fault를 발생하고 모터에 전력 공급을 차단합니다.

9.13.2 overcurrent_limit

모터에 흐르는 전류(current)가 overcurrent_limit에 설정된 값을 넘어가면 Fault를 발생합니다. 보통 이 값은 제어기의 구동부 소자(FET)를 통해 흘릴 수 있는 최대 전류로 설정되며, 모터에 흐르는 전류가 overcurrent_limit 전류 이상으로 올라가면 제어기를 보호하기 위하여 fault를 발생하고 모터에 전력 공급을 차단합니다.

9.13.3 overvoltage_limit

전원단에서 제어기로 입력되는 전압(power_source_voltage)이 overvoltage_limit에 설정된 값을 넘어가면 FF_OVER_CURRENT fault를 발생합니다. 보통 이 값은 제어기를 구성하는 각종 전자 부품들이 견딜 수 있는 최대 전압으로 설정되며, 전원 전압이 overvoltage_limit 전압 위로 올라가면 제어기를 보호하기 위하여 fault를 발생하고 모터에 전력 공급을 차단합니다.

9.13.4 under0voltage_limit

전원단에서 제어기로 입력되는 전압(power_source_voltage)이 undervoltage_limit에 설정된 값 아래로 떨어지면 FF_UNDER_VOLTAGE fault를 발생합니다. 보통 이 값은 제어기가 정상적으로 동작하기 위한 최소 전압으로 설정되며, 전원 전압이 undervoltage_limit 전압 아래로 떨어지면 제어기를 보호하기 위하여 fault를 발생하고 모터에 전력 공급을 차단합니다.

9.14 Fault Detection

9.14.1 vibration_detect

전류 제어기에서 발생하는 진동을 탐지하여 설정된 조건에 부합되는 경우 FF_VIBRATION fault를 발생합니다.

- 0 – Disable
- 1 - Current Frequency > 20Hz and RMS > 30% of Rated Current
- 2 - Current Frequency > 50Hz and RMS > 30% of Rated Current
- 3 - Current Frequency > 100Hz and RMS > 30% of Rated Current
- 4 - Current Frequency > 20Hz and RMS > 60% of Rated Current
- 5 - Current Frequency > 50Hz and RMS > 60% of Rated Current
- 6 - Current Frequency > 100Hz and RMS > 60% of Rated Current
- 7 - Current Frequency > 20Hz and RMS > 90% of Rated Current
- 8 - Current Frequency > 50Hz and RMS > 90% of Rated Current
- 9 - Current Frequency > 100Hz and RMS > 90% of Rated Current

9.14.2 stall_current_detect

모터가 스토퍼에 부딪혀 멈춘 상태에서 설정된 시간 동안 설정된 값 이상의 전류가 흐르면 FF_STALL_CURRENT fault를 발생합니다.

- 0 - Disable
- 1 - Zero Velocity and 50% of Max Current while 100ms
- 2 - Zero Velocity and 60% of Max Current while 200ms
- 3 - Zero Velocity and 70% of Max Current while 500ms
- 4 - Zero Velocity and 80% of Max Current while 1000ms
- 5 - Zero Velocity and 90% of Max Current while 2000ms
- 6 - Zero Velocity and 90% of Max Current while 1000ms
- 7 - Zero Velocity and 90% of Max Current while 500ms
- 8 - Zero Velocity and 90% of Max Current while 200ms

- 9 - Zero Velocity and 90% of Max Current while 100ms

9.14.3 position_track_error

위치 제어기에서 설정된 시간 동안 위치 오차(position_error)가 설정 값 이상을 초과하면 FF_POSITION_TRACKING fault를 발생합니다.

- 0 - Disable
- 1 – Position error > 100 counts while 100ms
- 2 - Position error > 500 counts while 200ms
- 3 - Position error > 2000 counts while 500ms
- 4 - Position error > 5000 counts while 1000ms
- 5 - Position error > 20000 counts while 2000ms

9.14.4 velocity_track_error

속도 제어기에서 설정된 시간 동안 속도 오차(velocity_error)가 설정 값 이상을 초과하면 FF_VELOCITY_TRACKING fault를 발생합니다.

- 0 - Disable
- 1 - Velocity error > 100 rpm(or mm/s) while 100ms
- 2 - Velocity error > 200 rpm(or mm/s) while 200ms
- 3 - Velocity error > 500 rpm(or mm/s) while 500ms
- 4 - Velocity error > 1000 rpm(or mm/s) while 1000ms
- 5 - Velocity error > 2000 rpm(or mm/s) while 2000ms

9.15 Motor Properties

9.15.1 motor_type

제어기에 연결된 모터의 종류를 설정합니다.

- 0 - DC Motor
- 1 - BLDC Motor
- 2 - PMSM Motor

9.16 Sensor Properties

9.16.1 position_sensor

모터의 위치를 파악하기 위해 사용하는 센서를 선택합니다.

- 0 – (none)
- 1 – Encoder
- 2 – Encoder with Index
- 3 – Hallsensor
- 4 – Encoder & Hallsensor
- 5 – Encoder with Index & Hallsensor

각 모터별 사용 가능한 센서는 다음과 같습니다.

	DC	BLDC	PMSM
(none)	X	X	X
Encoder	O	X	O
Encoder w. Index	O	X	O
Hallsensor	X	O	X
Encoder & Hallsensor	-	O	O
Encoder w. Index & Hallsensor	-	O	O

* O – 사용 가능, X – 사용 불가, - - 고려안함

9.16.2 encoder_direction

엔코더의 카운트 방향을 모터의 회전 방향과 일치하도록 설정합니다.

- 0 – Normal (기본값)
- 1 – Invert.

이 값이 올바르게 설정되었는지 확인하기 위해서, 모터를 전압 출력으로 구동합니다. 이때 모터의 회전 방향과 반대로 엔코더 값이 카운트 되는 경우, 현재의 encoder_direction 설정 값을 다른 값으로 변경해야 합니다 (0인 경우 1로, 1인 경우 0으로).

***주의: 모터의 회전 방향을 바꾸기 위해 엔코더의 카운트 방향 설정을 바꾸면 안됩니다. 이 값은 모터의 회전 방향과 엔코더의 카운트 방향이 불일치할 때 사용합니다. 즉, 모터에 양의 전압을 가했는데 엔코더가 거꾸로 카운트 되는 경우 이 값을 1-Invert로 설정합니다.**

***주의: 이 값이 모터의 회전 방향과 반대 방향일 경우 제어기가 모터를 컨트롤 할 수 없는 상태가 됩니다.**

9.16.3 encoder_resolution

엔코더의 한 회전당 펄스 카운트 수로 엔코더의 펄스 수를 4체배 한 값입니다. 즉, 모터 축을 한 바퀴 돌렸을 때 카운트 되는 엔코더 펄스 수가 됩니다.

이 값을 직접 찾기 위해서는, position_sensor를 '1 – Encoder'로 설정하고 모터 축을 손으로 한 바퀴 돌립니다. 이때 위치 카운트 값의 변화량이 encoder_resolution 값이 됩니다. 이 값은 4의 배수가 되어야 합니다. 한 바퀴로 값을 찾기 어려운 경우, 여러 바퀴 돌린 후 평균을 내면 어느정도 정확한 값을 얻을 수 있습니다.

일예로, 모터 축을 손으로 10바퀴 회전하였을 때 카운트 되는 위치 변화량이 1012라고 가정하겠습니다. 그러면 1회전당 101.2 펄스가 되고, 이 값은 4의 배수가 되어야 하기때문에 100으로 보면 됩니다.

***주의: 이 값이 잘못 설정되면, 모터의 회전속도와 자동으로 탐지되는 모터의 각종 파라미터들이 잘못 계산됩니다. 데이터 시트를 참조하여 올바른 값을 입력해야합니다.**

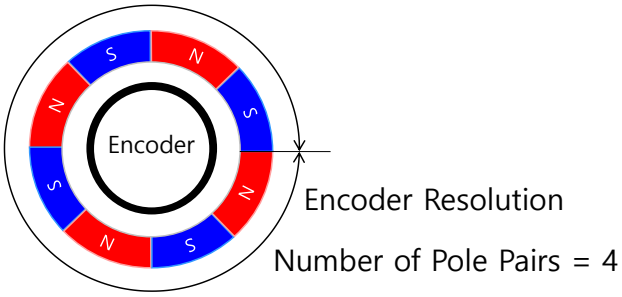
***주의: DC 모터에 Index 신호가 없는 엔코더가 장착된 경우, system_control의 SC_POS_SENSOR_PARAM 명령으로 자동 탐지한 encoder_resolution은 어림 값입니다. 데이터 시트를 참조하여 올바를 값을 입력해야합니다.**

9.16.4 no_pole_pairs

홀센서의 극쌍 수(number of pole pairs)를 설정합니다. no_pole_pairs 값은 모터 축을 한 바퀴 돌

렸을 때 카운트 되는 홀센서의 펄스 카운트 수를 6으로 나눈 값입니다.

일례로, 다음 그림에서는 모터를 한 바퀴 돌렸을 홀센서의 카운트 값은 24가 됩니다. 이때 no_pole_pairs 값은 4가 됩니다 (24/6).



no_pole_pairs 값을 직접 찾기 위해서는, position_sensor를 '3 – Hallsensor'로 설정하고 모터 축을 손으로 한 바퀴 돌립니다. 이때 위치 카운트 값의 변화량을 Δ라고 하면, Δ는 6의 배수가 되어야 합니다. 만일 Δ가 11, 12, 13 과 같다면 no_pole_pairs는 2라고 보면 됩니다.

9.16.5 hallsensor_phase

홀센서 입력이 들어오는 순서에 따른 홀센서 위상을 선택합니다.

- 0 - Normal,
- 1 - 60' Phase lead,
- 2 - 120' Phase lead,
- 3 - Opposite direction (Hall2 <--> Hall3),
- 4 - Opposite direction, 60' Phase lead,
- 5 - Opposite direction, 120' Phase lead,
- 6 - Inverted input,
- 7 - Inverted input, 60' Phase lead,
- 8 - Inverted input, 120' Phase lead,
- 9 - Inverted input, Opposite direction,
- 10 - Inverted input, Opposite direction, 60' Phase lead,
- 11 - Inverted input, Opposite direction, 120' Phase lead.

BLDC 또는 PMSM 모터에서 홀센서의 U, V, W 입력 신호가 들어오는 순서 및 위상에 따라 상기와 같이 12가지 조합이 있을 수 있습니다. 이 값을 수동으로 설정해야할 때는, position_sensor를 '3 – Hallsensor'로 설정하고 0 부터 11까지 hallsensor_phase 값을 바꾸어 가며 모터를 전압 출력으로 구동해 봅니다. 여기서 모터가 가장 매끄럽게 회전하는 경우의 값을 사용하면 됩니다.

이 값을 직접 설정하기 보다는 system_control의 SC_HS_PHASE_DETECT 명령으로 자동 탐지할 것을 권장합니다.

9.17 Electrical Parameters

9.17.1 bemf_constant

모터의 역기전력 상수는 모터에서 발생하는 역기전력을 전향보상하기 위해 Feed-forward 제어기에서 사용되며, 속도 제어기의 이득을 계산하는데 사용됩니다. 그리고 모터의 max_velocity, rated_velocity도 bemf_constant로부터 계산됩니다. 이 값은 Electric parameter estimation에서 계산됩니다.

9.17.2 resistance

모터의 권선 저항이며, 권선 저항은 모터의 정격 전류를 흘릴 때 측정한 값입니다. 모터의 권선 저항은 d축과 q축 전류 제어기의 이득을 계산하는데 사용됩니다. 이 값은 Electric parameter estimation에서 계산됩니다.

9.17.3 inductance

모터 권선의 인덕턴스는 d축과 q축 전류 제어기의 이득을 설정하는데 사용됩니다. 이 값은 Electric parameter estimation에서 계산됩니다.

사용자가 LRC메터로 모터의 권선으로부터 직접 값을 측정하여 입력할 수도 있지만, 가능하다면 제어기가 추정한 값을 사용할 것을 권장합니다.

9.18 Mechanical Parameters

9.18.1 inertia_ratio

모터와 부하의 관성 비 값입니다.

$$\text{inertia_ratio} = \frac{\text{부하 관성}}{\text{모터 관성(moment_of_inertia)}}$$

9.18.2 moment_of_inertia

모터 회전자의 관성 모멘트는 피드 포워드 제어기 및 속도 제어기의 이득을 계산하는데 사용됩니다. 또한 속도 프로파일의 가속도와 감속도를 결정하는 중요한 파라미터입니다. 이 값이 클수록 가속도와 감속도를 낮추어야 하고 작을 수로 가속도와 감속도를 높일 수 있습니다. 이 값은 Mechanic parameter estimation에서 계산됩니다.

9.18.3 viscous_friction

모터의 점성 마찰계수는 피드 포워드 제어기 및 속도 제어기의 이득을 계산하는데 사용됩니다. 이 값은 Mechanic parameter estimation에서 계산됩니다.

9.18.4 coulomb_friction

모터의 정지 마찰계수는 피드 포워드 제어기의 이득을 계산하는데 사용됩니다. 이 값은 Mechanic parameter estimation에서 계산됩니다.

9.19 Controller's Gain

9.19.1 position_p_gain

위치 제어기의 비례 이득으로, 이 값을 높이면 목표 위치를 더 빠르게 추종하게 됩니다. 하지만 과도하게 높인 경우 모터에 진동이 발생하게 됩니다.

이 값을 직접 조정하기 보다는 system_control의 SC_PV_GAIN_STIFFNESS 명령으로 조정할 것을 권장합니다.

9.19.2 velocity_p_gain

속도 제어기의 비례 이득으로, 이 값을 높이면 목표 속도를 더 빠르게 추종하게 됩니다. 하지만 과도하게 높인 경우 모터에 진동이 발생하게 됩니다.

이 값을 직접 조정하기 보다는 system_control의 SC_PV_GAIN_STIFFNESS 명령으로 조정할 것을 권장합니다.

***주의: SC_PV_GAIN_STIFFNESS 명령으로 속도 제어기의 이득을 설정하는 경우 모터의 올바른 Mechanic Parameter를 미리 입력해 두어야 합니다.**

9.19.3 velocity_i_gain

속도 제어기의 적분 이득으로, 이 값을 높이면 정상상태 오차를 빠르게 줄일 수 있습니다. 하지만 과도하게 높인 경우 모터에 저주파 진동이 발생하게 됩니다.

이 값을 직접 조정하기 보다는 system_control의 SC_PV_GAIN_STIFFNESS 명령으로 조정할 것을 권장합니다.

9.19.4 current_p_gain

d/q축 전류 제어기의 비례 이득으로, 이 값을 높이면 목표 전류를 더 빠르게 추종하게 됩니다. 하지만 과도하게 높인 경우 모터에 소음이 발생하게 됩니다.

이 값을 직접 조정하기 보다는 system_control의 SC_CC_GAIN_STIFFNESS 명령으로 조정할 것을 권장합니다.

***주의: SC_CC_GAIN_STIFFNESS 명령으로 전류 제어기의 이득을 설정하기 전에 모터의 올바른 Electric Parameter를 미리 설정해야 합니다.**

9.19.5 current_i_gain

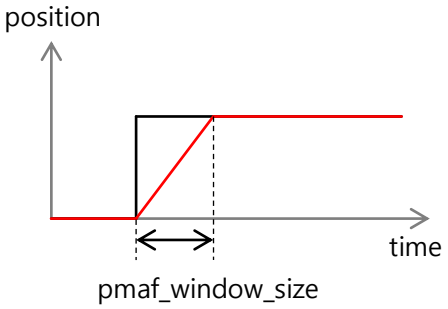
d/q축 전류 제어기의 적분 이득으로, 이 값을 높이면 정상상태 오차를 빠르게 줄일 수 있습니다. 하지만 과도하게 높인 경우 모터에 소음과 진동이 발생하게 됩니다.

이 값을 직접 조정하기 보다는 system_control의 SC_CC_GAIN_STIFFNESS 명령으로 조정할 것을 권장합니다.

9.20 Moving Average Filter

9.20.1 pmaf_window_size

위치제어기 입력단에 배치된 Moving Average 필터의 윈도우 사이즈를 설정합니다. 윈도우 사이즈는 1 ~ 256 사이의 값으로 설정합니다. 윈도우 사이즈가 1로 설정된 경우는 필터가 동작하지 않습니다.

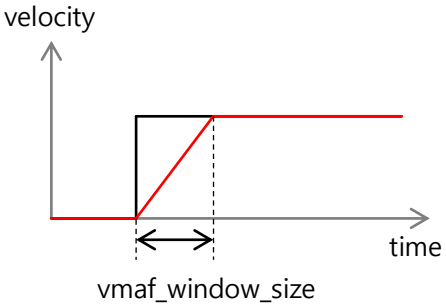


$$\text{position_demand} = \frac{1}{\text{n}} \sum_{i=0}^{\text{n}-1} \text{target_position}[M-i]$$

시간 축에서 검은색 그래프가 필터를 통과하기 전의 위치 값이고, 붉은색 그래프가 필터를 통과한 후의 위치 값입니다. 급격한 위치 이동 명령이 Moving Average 필터를 통과함으로 부드럽게 변하는 위치 이동 명령으로 바뀌게 됩니다.

9.20.2 vmaf_window_size

속도제어기 입력단에 배치된 Moving Average 필터의 윈도우 사이즈를 설정합니다. 윈도우 사이즈는 1 ~ 256 사이의 값으로 설정합니다. 윈도우 사이즈가 1로 설정된 경우는 필터가 동작하지 않습니다.



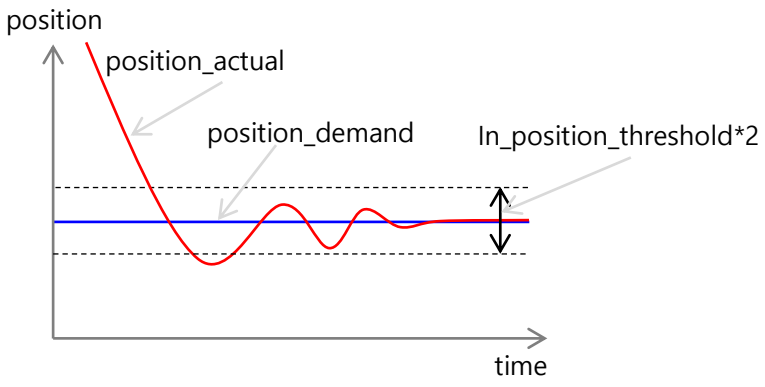
$$\text{velocity_demand} = \frac{1}{\text{n}} \sum_{i=0}^{\text{n}-1} \text{target_velocity}[M - i]$$

시간 축에서 검은색 그래프가 필터를 통과하기 전의 속도 값이고, 붉은색 그래프가 필터를 통과한 후의 속도 값입니다. 급격한 속도 명령이 Moving Average 필터를 통과함으로 부드럽게 변하는 속도 명령으로 바뀌게 됩니다.

9.21 I/O Parameters

9.21.1 in_position_threshold

목표 위치 도달을 판단하기 위한 threshold 값을 설정합니다. position_demand – position 값이 in_position_threshold 이하로 4ms 이상 유지되면 모터가 목표 위치에 도달하였다고 판단합니다.



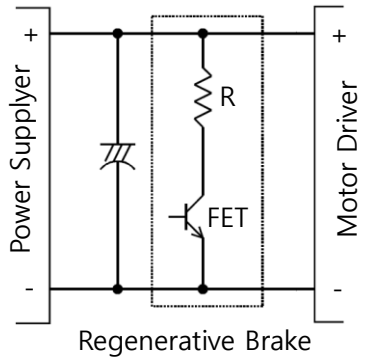
9.21.2 in_velocity_threshold

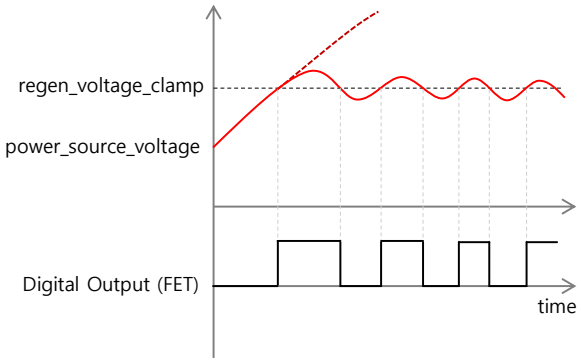
목표 속도 도달을 판단하기 위한 threshold 값을 설정합니다. velocity_demand – velocity 값이 in_velocity_threshold 이하로 4ms 이상 유지되면 모터가 목표 속도에 도달하였다고 판단합니다.

9.21.3 regen_voltage_clamp

전원단의 전압이 regen_voltage_clamp 설정 값 이상으로 올라가면 FET나 릴레이를 통해 연결된 저항으로 전원단을 단락 시켜 회생 에너지를 저항으로 소모하게 됩니다.

회생 브레이크: 관성이 큰 부하가 장착된 모터를 감속하거나 정지한 모터를 강제로 회전시킬 경우, 모터는 발전기 상태가 되어 전기 에너지를 전원단으로 되돌리게 됩니다. 충방전이 가능한 베터리와 같이 되돌아온 에너지를 수용할 수 있는 경우는 문제가 없지만, 파워 서플라이와 같이 되돌아온 에너지를 다시 수용하지 못할 경우 전원단의 전압이 상승하여 제어기를 파손시키게 됩니다. 이것을 방지하기 위해서 다음 그림과 같이 전원단의 +, - 단자 사이에 저항과 FET를 직렬로 연결하여, 전원 전압이 일정 값이 상으로 상승할 경우 FET를 켜 저항으로 전류를 흘림으로 회생 에너지를 열로 소비시킵니다.





10 관련 자료

아래 홈페이지에서 MoonWalker 관련 제품과 자료 그리고 예제, 동영상을 확인하실 수 있습니다.

- **엔티렉스:**
  http://www.ntrexgo.com/
- **디바이스마트:**
  http://www.devicemart.co.kr/
- **MoonWalker 제품:**
  http://mwbot.co.kr/
- **MoonWalker 액추에이터:**
  http://www.devicemart.co.kr/goods/list.php?category=006011008011

- **MoonWalker 판매페이지:**
  http://www.devicemart.co.kr/goods/list.php?category=006011008

제품의 보증

1. 본 제품은 엄정한 품질관리 및 검사과정을 거쳐서 만들어 진 제품입니다.
2. 제품 구입 후 1개월 이내에 제품 고장 발생 시에 무상으로 A/S를 해드립니다.
3. 정상적인 사용 상태에서 고장이 발생하였을 경우 보증기간 동안은 무상으로 A/S를 해드립니다.
4. 제품 보증기간이 경과한 후에 고장이 발생할 경우 유상으로 A/S를 해드립니다.
5. 보증기간 이내라 하더라도 본 보증 이내의 유상 서비스 안내에 해당되는 경우 서비스 따라 유상으로 A/S를 해드립니다.
6. 오용, 남용 및 인가되지 않은 인력에 의한 수리, 부적절한 보관상태 자연 재해로 인한 파손은 유상으로 A/S를 해드립니다.
7. 고객 변심 또는 구매 후 7일 이후에는 반품이 되지 않습니다.

회 사 명	(주)엔티렉스
연 구 소 주 소	인천 미추홀구 염전로 330 제이타워 701호 (주)엔티렉스
전 화 번 호	070 - 7019 - 8887
팩 스 번 호	02 - 6008 - 4953
E – Mail	기술문의 - lab@ntrex.co.kr 영업문의 - stock@ntrex.co.kr