

MS-60S 시리즈 사용설명서

모터 사양

항목	단위	MS-60S-200W 사용된 모터	MS-60S-400W 사용된 모터	MS-60S-600W 사용된 모터
정격 출력	W	200	400	600
정격 전압	V	24	48	48
정격 전류	A	12	10	15
정격 토크	N·m	0.64	1.27	1.91
정격 속도	r/min	3000	3000	3000
순간 최대 전류	A	36	30	30
순간 최대 토크	N·m	1.91	3.81	3.81
순간 최고 속도	r/min	3200	3200	3200
Back EMF	V/1000 r/min	3.85	7.8	8.5
토크 상수	N·m/A	0.053	0.127	0.127
권선 저항	Ω	0.17	0.39	0.25
권선 인덕턴스	mH	0.27	0.72	0.43
전기적 시정수	ms	1.58	1.8	1.7
로터 관성 모멘트	kg·m ²	0.29	0.53	0.68
중량	kg	1	1.4	1.7
극수		5	5	5
내열 등급		Class F	Class F	Class F
보호 등급		IP65	IP65	IP65
주변 온도	°C	-20~40	-20~40	-20~40

모터 전원 단자 정의

단자 기호	명칭	기능
U	모터 U상 권선	U 블루
V	모터 V상 권선	V 레드
W	모터 W상 권선	W 블랙

엔코더 / 통신 단자 정의

단자 번호	명칭	기능
1	485A	RS-485 엔코더 A 신호 그린
2	485B	RS-485 엔코더 B 신호 블루
3	GND	전원 GND 블랙
4	NC	
5	5V	전원 5 V 출력 레드
6	NC	
7	NC	
8	NC	
9	NC	

1. 드라이버 특성

1. 15비트 절대형 엔코더 적용, 1회전당 최대 32,768 펄스 지원.
2. 저소음 · 저진동, 고속 위치 결정 및 고신뢰성 실현.
3. FOC 벡터 제어 방식 적용, 위치/속도 폐루프 제어 지원.
4. 영(0) 지연 펄스 지령 모드에서 동작 가능, 무지연 추종 제어 지원.
5. 원점 신호 자동 검색 및 Z상 신호 자동 검출 기능 지원.
6. 16비트 전자 기어 기능 지원.
7. 내부 절연 Modbus RTU 통신 지원. (통신 속도 설정 가능, 내장 절연 전원 적용)
8. 위치 제어 모드: 펄스 + 방향 신호, 정사각형 펄스 신호 지원.
9. 속도 제어 모드: PWM 듀티비 신호, 4~20 mA 전류 신호, 0.6~3 V 전압 신호 지원.
10. 전원 전압: DC +20V ~ 50V, 50~500W급 서보 모터 지원.
11. 절연 출력 신호 제공: 알람 출력 신호, 엔코더 원점 신호.
12. 보호 기능 내장: 저전압, 과전압, 스톨, 과열.

2. 드라이버 결선

1. 전원 (드라이버상 표기: Power Motor)

명칭	기능
+V	직류 전원 입력 (+), DC +24V ~ +48V 전압이 설정 범위보다 낮을 경우 드라이버 알람 발생 및 정지될 수 있음
GND	직류 전원 GND
U	모터 U상 권선
V	모터 V상 권선
W	모터 W상 권선

2. 제어 신호(드라이버상 표기: Signal)

명칭	기능
PU+ (+5V) PU- (PU)	펄스 제어 신호 입력: 펄스 상승 에지 유효 PU- 기준 고레벨: 4~5V, 저레벨: 0~0.5V 펄스 신호의 안정적인 인식을 위해 펄스 폭은 1.2 μ s 이상 권장 +12V 또는 +24V 신호 사용 시 직렬 저항 삽입 필요
DIR+ (+5V) DIR- (DIR)	회전 방향 신호 입력: High / Low 레벨 제어 모터의 안정적인 방향 전환을 위해 방향 신호는 펄스 신호보다 최소 5 μ s 이상 선행하여 인가되어야 함 DIR- 기준 고레벨: 4~5V, 저레벨: 0~0.5V
EN+ (+5V) EN-(EN)	Enable(구동 허용) 신호 입력: 드라이버의 구동 허용 또는 차단 을 제어 ENA+에 +5V, ENA-에 저레벨(또는 내부 포토커플러 ON) 입력 시 드라이버는 각 상 전류를 차단하여 모터를 자유 상태로 전환 해당 상태에서는 펄스 신호가 인식되지 않음 Enable 기능을 사용하지 않을 경우, 해당 단자는 미접속 가능 드라이버 알람 발생 시, Enable 차단을 통해 알람 해제 가능

BLD-400R4 드라이버는 차동 방식 인터페이스 회로를 채택하여 차동 신호, 단일단 공통 음 및 공통 양 방식 인터페이스를 모두 지원합니다.

또한 내장형 고속 포토커플러를 적용하여 라인 드라이버 출력, 오픈 컬렉터 및 PNP 출력 회로의 신호 입력을 허용합니다.

전기적 노이즈가 많고 환경 조건이 열악한 경우에는 장거리 전송에 적합한 라인 드라이버 방식 사용을 권장하며, 이를 통해 우수한 내노이즈 성능과 신뢰성 있는 신호 전송을 확보할 수 있습니다.

3. 모터 신호 파트 연결

단자 번호	명칭	기능
1	485A	485 엔코더 485A 신호
2	485B	485 엔코더 485B 신호
3	GND	전원 GND
4	NC	
5	5V	전원 5 V 출력
6	NC	
7	NC	
8	NC	
9	NC	

4. 통신 및 출력 신호

전용 485 케이블을 통해 PC와 연결할 수 있습니다.

제공되는 **상위 제어 소프트웨어(PC 프로그램)**를 이용하여 드라이버 파라미터 설정, 동작 테스트 및 각종 진단 정보 확인이 가능하며, 이를 통해 드라이버 이상 상태를 신속하게 점검할 수 있습니다.

주의: 1번 핀: 하단 좌측 첫 번째 핀

6번 핀: 상단 좌측 첫 번째 핀

단자 번호	기호	기능
1	NC	
2	485A	RS-485 통신 A 단자 (드라이버 내장 절연 전원)
3	485B	RS-485 통신 B 단자 (드라이버 내장 절연 전원)
4	NC	
5	V_in	아날로그 전압 입력
6	COM	출력 신호 공통 및 통신 공통
7	WR	알람 출력 신호 (알람 발생 시 Low 레벨)
8	GND	아날로그 전압 입력 공통
9	ZO	엔코더 원점(Zero) 신호 출력 (원점 도달 시 Low 레벨)
10	5V_OUT	+5 V 출력 (최대 100 mA) 포텐서미터 연결 가능: 5V_OUT / GND / V_in

5. DIP 스위치 설정

SW1	SW2	동작 모드	제어 방식
OFF	OFF	위치 모드	펄스 + 방향
OFF	ON	위치 모드	AB 펄스, 정사각형 펄스, 엔코더 추종
ON	OFF	속도 모드	PWM 듀티비
ON	ON	속도 모드	4~20 mA 또는 0.6~3 V

6. 상태 표시 및 알람

전원 인가 시 레드 LED와 그린 LED가 1회 동시에 점등되며, 이는 LED 정상 동작 여부를 확인하기 위한 동작입니다.

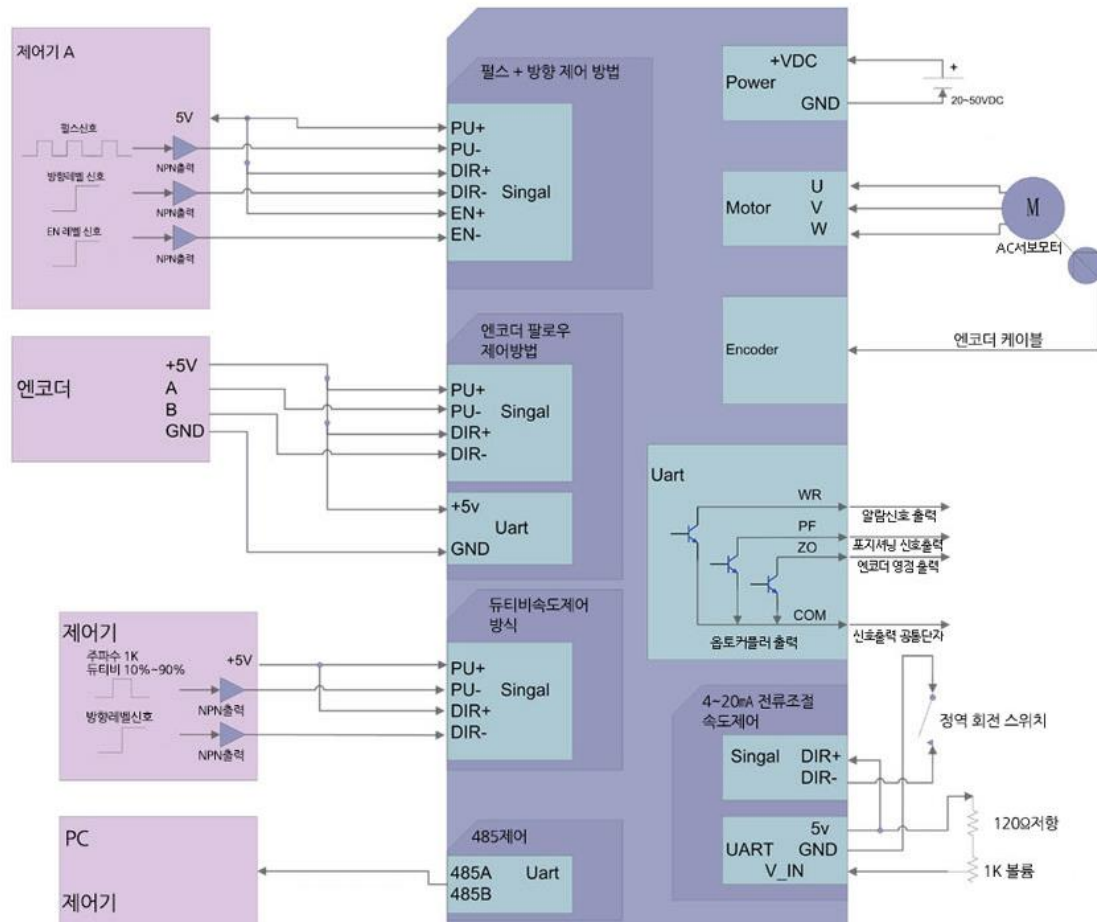
이후 그린 LED 점등 / 레드 LED 소등 상태가 정상 상태입니다.

알람 발생 시에는 레드 LED 점멸 패턴을 통해 원인을 확인할 수 있으며, 또는 Modbus 통신을 통해 알람 코드를 직접 읽어 확인할 수 있습니다.

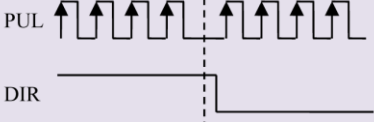
알람 코드	레드 LED 점멸	알람 원인	처리 방법
0x12	1회 장점멸+2회 단점멸	시스템 과전류 알람	정지
0x13	1회 장점멸+3회 단점멸	시스템 저전압 알람 (< 20 V)	정지 / 전압이 20 V 이상 회복 시 재동작
0x14	1회 장점멸+4회 단점멸	스톨(구속) 알람, 과부하	정지
0x15	1회 장점멸+5회 단점멸	과전압 알람	정지 전원 입력 과전압 또는 관성 부하 감속 시 회생 전압 상승

3. 드라이버 결선도 및 제어 방법

1. 드라이버 결선도



2. 펄스커맨드 + 방향위치 제어 모드

펄스 커맨드 형식	CCW	CW	파라미터 세팅값
펄스 기호			펄스 커맨드 + 방향

1회전당 3200 펄스가 필요한 경우:

전자 기어는 32768(엔코더 1회전당 펄스 수) : 3200(설정이 필요한 1회전당 펄스 수)로 설정하며, 약분하면 256 : 25가 됩니다.

1회전당 8192 펄스가 필요한 경우(기본 설정):

전자 기어는 32768 : 8192로 설정하고, 약분 후 비율은 4 : 1입니다.

주의: 전자 기어 비율은 가능한 한 약분하여 설정해야 하며, 전자 기어 분자 값이 32768과 같이 지나치게 클 경우 추종 성능에 영향을 줄 수 있습니다.

지령 펄스 주파수는 다음과 같이 계산됩니다.

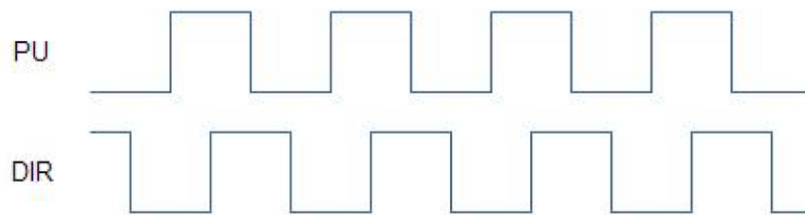
지령 펄스 주파수 = (필요한 모터 회전 속도 / 60) × 1회전당 펄스 수

예를 들어, 목표 회전 속도가 1000 RPM이고 1회전당 펄스 수가 8192인 경우, 펄스 주파수 = $1000 / 60 \times 8192 = 136,533 \text{ Hz}$ 입니다.

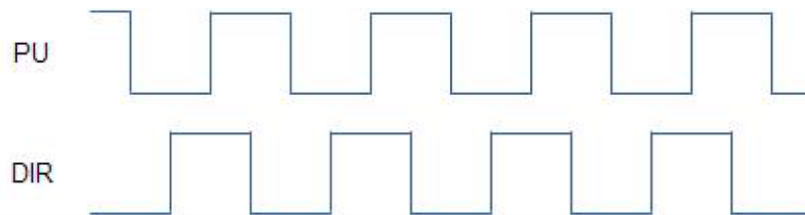
3. 구형파 커맨드 펄스 위치 제어 방법

특수 기능(주소 0x19)을 2로 설정한 후 전원을 다시 인가하면, 드라이버는 엔코더 추종 모드로 동작합니다. 이 모드는 엔코더 추종 제어에 사용되며, 하나의 축에 엔코더를 장착하고 해당 엔코더 출력 신호를 드라이버에 입력하면(배선 방식은 드라이버 기본 배선도 참조), 드라이버는 입력된 엔코더 신호에 따라 서보 모터를 동기 추종 제어합니다. 또한 전자 기어(Electronic Gear) 설정을 통해 제어 엔코더의 회전 각도와 모터 회전 각도 간의 비율을 조정할 수 있습니다.

정방향 펄스



역방향 펄스



모터 회전 방향은 PU 신호와 DIR 신호의 상승 에지 타이밍 관계에 의해 결정됩니다. PU 상승 에지가 DIR 상승 에지보다 선행할 경우 정회전, PU 상승 에지가 DIR 상승 에지보다 지연될 경우 역회전으로 동작합니다.

4. 파라미터 조정

모터에 연결되는 부하 조건에 따라, 최적의 성능을 얻기 위해서는 파라미터 조정이 필요합니다.

1. 내부 가감속 곡선

컨트롤러 출력 신호 방식에 따라 내부 가감속 곡선 사용 여부를 선택할 수 있습니다.

내부 가속 곡선 사용

모터 가속도 설정 값이 60000 미만일 경우, 드라이버는 내부 가감속 곡선을 활성화하며 실제 가속도는 설정된 값과 동일하게 적용됩니다.

적용 환경: 내부 가속 곡선을 사용할 경우 펄스 지연 현상이 발생하므로, 실시간 추종이 필요하지 않은 시스템에 적합합니다. 일부 컨트롤러는 가감속 없이 목표 속도에 해당하는 펄스를 직접 출력하므로, 이 경우 내부 가감속 곡선을 사용하면 컨트롤러 프로그래밍 난이도를 낮출 수 있습니다.

내부 가속 곡선 비활성화

모터 가속도 설정 값이 60000 이상일 경우, 드라이버는 외부 펄스의 가감속을 그대로 따르며 내부 가속도 설정은 무효가 됩니다.

적용 환경: 예를 들어 **조각기(CNC)**와 같이 컨트롤러에서 이미 가감속이 포함된 펄스를 출력하는 경우, 드라이버 내부 가속 곡선을 사용하면 실제 펄스보다 지연이 발생하므로 사용하지 않는 것이 바람직합니다.

2. 볼스크류(리드스크류) 부하

토크 개념을 설명하기 위해 400W 모터, 정격 토크 1.3 N·m를 기준으로 설명합니다. 부하가 리드 5 mm인 볼스크류일 경우, 모터가 1회전하면 부하는 5 mm 이동합니다. 이때 부하의 등가 레버 암은 다음과 같습니다.

- 등가 레버 암 = $5 \text{ mm} / \pi = 1.592 \text{ mm}$

모터가 제공할 수 있는 추력은 다음과 같습니다.

- 스크류 전달 추력 = $1.3 \text{ N} \cdot \text{m} / (1.592 \text{ mm} \times 0.001) = 816 \text{ N}$

이는 약 80 kg의 하중을 수직으로 지탱할 수 있음을 의미하며, 수평 구동 시에는 더 큰 하중도 가능합니다.

볼스크류 부하는 모터 1회전당 이동 거리가 짧기 때문에, 드라이버 파라미터는 **가속도 값을 크게 설정(예: 20000)**할 수 있으며, **위치 루프 KP 값도 크게 설정(예: 3000)**할 수 있습니다. 서보 모터는 이러한 볼스크류 부하에 가장 적합합니다.

3. 타이밍 벨트(풀리) 부하

서보 모터는 일반적으로 벨트 풀리 부하에 적합하지 않습니다.

예를 들어 풀리 직경이 30 mm인 경우, 모터 1회전 시 부하 이동 거리는 $30 \text{ mm} \times \pi = 94.2 \text{ mm}$ 로, 앞서 설명한 5 mm 볼스크류보다 매우 큽니다.

이 경우 모터가 제공할 수 있는 추력은 다음과 같습니다.

- 벨트 전달 추력 = $1.3 \text{ N}\cdot\text{m} / (30 \text{ mm} \times 0.001) \approx 43.3 \text{ N}$
즉, 약 4.3 kg의 하중만 구동할 수 있습니다.

이는 풀리 구동 방식에서 레버 암이 길어지기 때문입니다. 이러한 경우 서보 모터를 사용하려면 가능한 한 작은 풀리를 모터 측에 장착하거나, 모터 측에는 작은 풀리, 부하 측에는 큰 풀리를 사용하여 감속비를 확보하는 것이 바람직합니다.

이와 같은 조건에서는 부하의 등가 관성이 크므로, 드라이버 파라미터는 **가속도를 작게 설정(예: 5000)**하여 가감속 충격을 줄여야 합니다.

4. 원판(디스크) 부하

이와 같은 부하는 서보 모터가 직접 구동하기 어렵기 때문에, 일반적으로 감속기를 사용해야 합니다.

예를 들어 직경 200 mm, 중량 10 kg의 원판 부하는 반경이 100 mm이며, 등가 반경은 약 50 mm로 레버 암이 매우 큽니다.

서보 모터를 이러한 부하에 사용할 경우, 감속기를 연결한 후 부하를 구동하는 것을 권장합니다.

원판(디스크) 하중이 크지 않은 경우에는 일부 위치 정밀도와 강성을 희생하여도 제어가 가능합니다.

이때의 권장 설정 방법은 다음과 같습니다.

- 모터 가속도: 비교적 낮은 값으로 설정
→ 예: 약 1000
- 속도 루프 KI: 2000으로 설정하여
→ 적분 동작을 비활성화
- 위치 루프 KP: 1000으로 조정

상기와 같은 파라미터 설정 시, 일반적인 원판형 부하에서는 안정적으로 사용이 가능합니다.

5. 자동 단일 회전 원점 찾기 기능

자동 원점 찾기 기능은 레지스터 주소 0x19(특수 기능) 설정을 통해 선택합니다.
전원 인가 시 자동으로 원점을 찾도록 설정하는 방법은 다음과 같습니다.

- Modbus Enable: 1 전송
- 특수 기능(주소 0x19): 10 ~ 32768 전송 (32768은 모터 **360**에 해당)
- 파라미터 저장: 1 전송

설정 후 전원을 다시 인가하면 자동으로 원점 검색을 수행합니다.

절대값 엔코더를 사용하므로, 전원 인가 시 1회전 내 임의 위치를 원점으로 설정할 수 있습니다. (DIR 극성 1 또는 0 설정을 통해 원점 검색 방향 설정 가능)

6. 자동 기계 원점 찾기 기능

자동 기계 원점 기능 역시 **레지스터 주소 0x19(특수 기능)**을 통해 설정합니다.
전원 인가 시 자동으로 기계 원점을 찾는 설정 방법은 다음과 같습니다.

- Modbus Enable: 1 전송
- 특수 기능(주소 0x19): 1 전송 (즉시 자동 기계 원점 검색 수행)
- 파라미터 저장: 1 전송 (저장 시 재기동 후 자동 실행)

전원을 다시 인가하면 모터는 역회전하여 스톱(구속) 상태에 도달한 후,
다시 36° 역회전한 위치를 원점으로 설정합니다. (DIR 극성 1 또는 0 설정으로 원점
검색 방향 변경 가능)

7. 자동 원점 찾기 기능 (원점 스위치 사용)

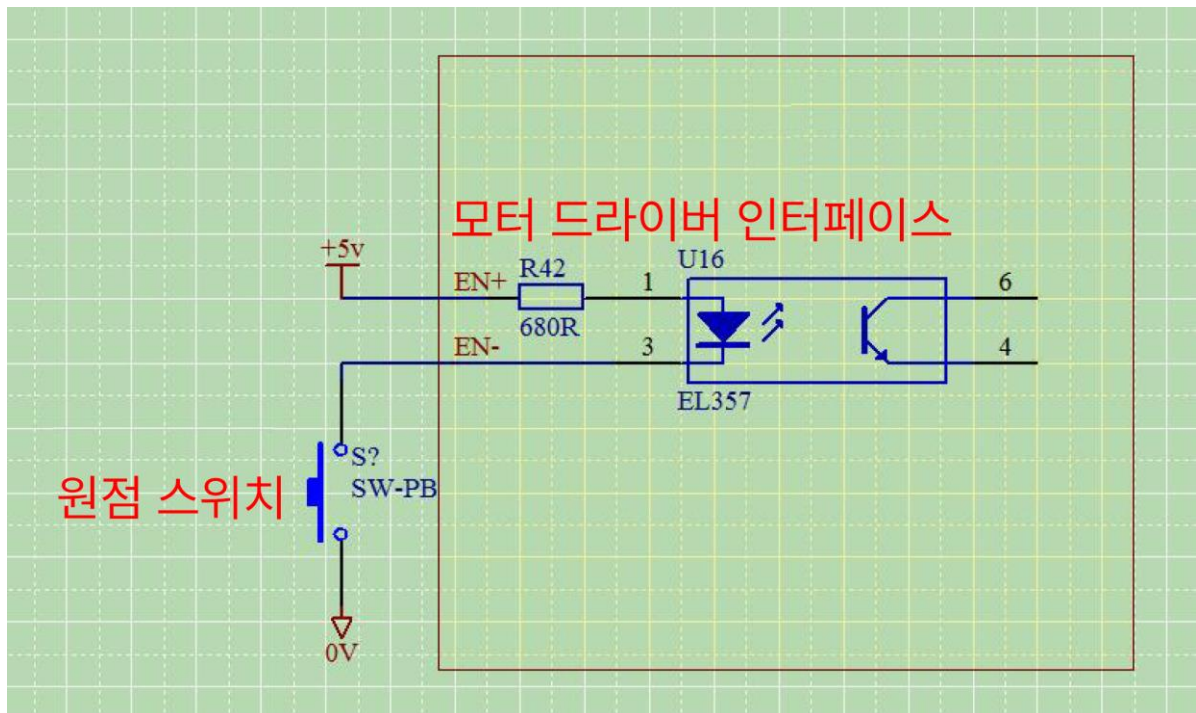
원점 스위치를 이용한 자동 원점 기능은 **레지스터 주소 0x19(특수 기능)**을 통해
설정합니다.

전원 인가 시 자동 원점 검색 설정 방법은 다음과 같습니다.

- Modbus Enable: 1 전송
- 특수 기능(주소 0x19): 8 전송 (즉시 원점 검색 수행)
- 파라미터 저장: 1 전송 (저장 시 재기동 후 자동 실행)

전원을 다시 인가하면 모터는 역회전하여 EN 신호가 입력될 때까지 이동한 후,
정회전하여 엔코더 원점 위치를 원점으로 설정합니다. (DIR 극성 1 또는 0 설정으로
원점 검색 방향 변경 가능)

원점 스위치는 반드시 EN 신호 단자에 연결해야 합니다.
배선 방식은 다음 도면을 참조하십시오.



8. 통신 방식 위치 초기화

절대 위치 초기화

운전 중 절대 위치를 0으로 초기화해야 하는 경우, 다음 절차를 따르십시오.

전자기어 분자에 0 전송

(통신 모드에서는 전자기어가 무효이나, 본 특수 기능 수행을 위해 사용됨)

통신 제어 방식에서는 전자기어 분자를 직접 0으로 저장 가능)

절대 위치 레지스터 (0x16)에 0 전송

→ 절대 위치가 즉시 0으로 초기화됨

급정지

통신 모드에서 아직 많은 펄스 이동량이 남아 있는 상태에서 즉시 정지가 필요한 경우 사용합니다.

전자기어 분자에 0 전송

(통신 모드에서는 전자기어가 무효이나, 급정지용 특수 기능으로 사용
통신 제어 시 전자기어 분자를 0으로 저장 가능)

표 위치 레지스터 (0x0C)에 0 전송

→ 급정지 수행

급정지 시에도 소량의 감속 거리가 존재하며, 해당 감속 거리의 크기는 위치 루프 KP
값에 의해 결정됩니다

9. 전원 인가 시 기본 통신 제어

전자기어 분자를 0으로 설정한 후 저장하면, 전원을 다시 인가했을 때 Modbus 통신
Enable 값은 기본적으로 1로 설정됩니다.

5. Modbus 제어 방식

1. 하드웨어 연결

컨트롤러와 모터 연결 시, RS-485A / RS-485B 두 가닥만 연결하면 됩니다.

다수의 모터는 서로 다른 주소로 설정하여 데이터 체인 방식으로 연결 가능

드라이버 내부의 RS-485 회로는 광절연(Opto-isolation) 처리되어 있음

다수의 슬레이브 연결 시 발생할 수 있는 노이즈 간섭 및 손상 문제를 효과적으로 방지

2. 레지스터 설명

드라이버는 Modbus RTU 모드를 통해 제어됩니다.

마스터는 레지스터 읽기/쓰기 기능을 통해 파라미터 설정 및 운전 제어가 가능합니다.

지원 기능 코드

0x03 : 레지스터 읽기

0x06 : 레지스터 쓰기

0x78 : 목표 위치 쓰기

0x7A : 장치 주소 변경

레지스터 주소	파라미터 명칭	읽기/쓰기	설정 범위	설명
0x00	Modbus 사용 설정	읽기/쓰기	0~1	0: Modbus 비활성 / 1: Modbus 활성
0x01	드라이버 출력 Enable	읽기/쓰기	0~1	0: 출력 금지 / 1: 출력 허용
0x02	모터 목표 속도	읽기/쓰기	0~3000 r/min	속도 모드: 목표 속도 / 위치 모드: 최대 속도
0x03	모터 가속도	읽기/쓰기	0~60098 (r/min)/s	60000 미만: 내부 가감속 곡선 적용 / 60000: 가속 없음 / 60000~60098: 위치 전방보상(0~98%)
0x04	약자속 각도	읽기/쓰기	0~306 r/min	내부 파라미터 (별도 설정 불필요)
0x05	속도 루프 비례 계수	읽기/쓰기	0~10000	0.0~10.0 대응, 값이 클수록 강성 증가
0x06	속도 루프 적분 시간	읽기/쓰기	2~2000 ms	값이 작을수록 강성 증가
0x07	위치 루프 비례 계수	읽기/쓰기	60~30000	위치 KP, 값이 클수록 강성 증가
0x08	속도 전방 보상	읽기/쓰기	0~12.0 V/KRPM	327 = 1V/KRPM (기본값)
0x09	DIR 극성	읽기/쓰기	0~1	0: DIR OFF 시 CW / 1: DIR ON 시 CW
0x0A	전자기어 분자	읽기/쓰기	0~65535	16 비트 전자기어 분자, 0 설정 시 특수 기능
0x0B	전자기어 분모	읽기/쓰기	1~65535	16 비트 전자기어 분모
0x0C	목표 위치 하위 16 비트	읽기 전용	-	이동 스텝 수 상위 16 비트
0x0D	목표 위치 상위 16 비트	읽기 전용	-	이동 스텝 수 하위 16 비트
0x0E	알람 코드	읽기 전용	-	-
0x0F	시스템 전류	읽기 전용	0~32767	실제 전류 = 값 / 2000 (A)
0x10	모터 현재 속도	읽기 전용	- 30000~30000 r/min	실제 회전속도 = 값 / 10

0x11	시스템 전압	읽기 전용	0~32767	실제 전압 = 값 / 327 (V)
0x12	시스템 온도	읽기 전용	0~100 °C	섭씨 온도
0x13	시스템 PWM 출력	읽기 전용	- 32768~32767	-100% ~ +100%
0x14	파라미터 저장 상태	읽기/쓰기	0~2	0: 미저장 / 1: 저장 중 / 2: 저장 완료
0x15	장치 주소	읽기 전용	0~255	디바이스 주소
0x16	절대 위치 하위 16 비트	읽기/쓰기	-	누적 이동 스텝 상위 16 비트
0x17	절대 위치 상위 16 비트	읽기/쓰기	-	누적 이동 스텝 하위 16 비트
0x18	정지 시 최대 허용 출력	읽기/쓰기	0~609	0~60.9% 출력 제한 / 하위 0: 스톱 알람 없음 / 3 초 후 출력 1/2 감소
0x19	특수 기능	읽기/쓰기	0~32768	0 : 펄스 + 방향 제어 모드 1 : 자동 기계 원점 탐색 후 정방향 36° 이동 (전원 인가 시 자동으로 역 회전하여 기계적 원점으로 이동한 후, 정방향으로 36° 이동 후 정 지) 2 : 엔코더 추종 모드 3 : 속도 모드 (PWM 듀티 제어 방식) → 듀티비 10% ~ 90%에 대 응하여 0 ~ 1000 RPM 속도 제어 10 ~ 32768 : 전원 인가 시 자동 이동 각도 설정 → 계산식: $X \times 360^\circ / 32768$ 4 : 자동 기계 원점 탐색 후 엔코더 원점까지 이동 (전원 인가 시 역회전하여 기계적 원점 탐색 후, 정방향으로 이동하여 엔코

				더 원점 위치에서 정지) 5 : Z_OUT 원점 신호 시뮬레이션 모드 (전원 인가 후 모터에 펄스를 인가하여 저속 역회전 → 스톨 발생 시 Z_OUT 출력 ON → 이후 정회전하여 Z_OUT 신호가 OFF 되는 지점을 시스템 원점으로 설정 ※ 엔코더 원점 신호는 1회 발생 후 소멸되어 원점 위치의 정확도를 보장함) 8 : 자동 원점 탐색 모드 (원점 신호 EN 입력 사용) (전원 인가 시 EN 신호가 ON 될 때까지 역회전 → 이후 정회전하여 EN 신호가 OFF 되는 지점에서 정지)
--	--	--	--	---

3. Modbus 통신 포

a. Modbus 마스터 데이터 읽기 및 슬레이브 응답 포맷 (기능 코드: 0x03)

마스터 → 슬레이브							
장치 주소	기능 코드	시작 레지스터 상위 주소	시작 레지스터 하위 주소	레지스터 개수상위	레지스터 개수하위	CRC High	CRC Low
0x01	0x03	0x00	0x00	0x00	0x01	0x84	0x0a
슬레이브 → 마스터							
장치 주소	기능 코드	데이터 길이	데이터 상위 바이트	데이터 하위 바이트	CRC High	CRC Low	
0x01	0x03	0x02	0x00	0x01	0x79	0x84	

시리얼 통신(Modbus)을 통해 수신되는 데이터는 모두 무부호(unsigned) 값이다.
만약 해당 레지스터가 부호 있는 값(signed) 인 경우, 전송 데이터는 2의 보수(Binary Two's Complement) 형식으로 전달된다.

아래는 이를 부호 있는 정수로 변환하는 알고리즘 예시(VB 코드) 이다.

```
If modbus.data(11) > 32767 Then
```

```
    disp_modbus_data.PU = (modbus.data(11) - 32768) * 65536 + modbus.data(10)
```

```
    disp_modbus_data.PU = -(&H7FFFFFFF - disp_modbus_data.PU) + 1)
```

```
Else
```

```
    disp_modbus_data.PU = dmodbus.data(11) * 65536 + modbus.data(10)
```

```
End If
```

주의: modbus.data(11)은 목표 위치(Target Position)의 상위 16비트

modbus.data(10)은 목표 위치(Target Position)의 하위 16비트

b. Modbus 마스터 데이터 쓰기 및 슬레이브 응답 (기능 코드: 0x06)

마스터 → 슬레이브							
장치 주소	기능 코드	시작 레지스터 상위 주소	시작 레지스터 하위 주소	데이터 상위 바이트	데이터 하위 바이트	CRC High	CRC Low
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

슬레이브 → 마스터							
장치 주소	기능 코드	시작 레지스터 상위 주소	시작 레지스터 하위 주소	데이터 상위 바이트	데이터 하위 바이트	CRC High	CRC Low
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

c. Modbus 마스터 32비트 펄스 수 쓰기 (기능 코드: 0x10)

마스터 → 슬레이브 (PU 펄스 수 기록)						
장치 주소	기능 코드	시작 레지스터 상위 주소	시작 레지스터 하위 주소	레지스터 개수 상위 바이트	레지스터 개수 하위 바이트	데이터 길이
0x01	0x10	0x00	0x0c	0x00	0x02	0x04
PU[15~8]	PU[7~0]	PU[31~24]	PU[23~16]	CRC High	CRC Low	
0x27	0x10	0x00	0x00	0xf8	0x8b	

펄스 수(PU)는 **부호 있는 정수(signed integer)**이다.

음수 값(가정: X)을 32비트 16진수 데이터로 변환하는 알고리즘은 다음과 같다.

(VB 언어 기준 예제)

If X < 0 Then

X = &H7FFFFFFF + (X + 1)

PU24_31 = Fix(X / (256 * 65536)) + &H80

Else

PU24_31 = Fix(X / (256 * 65536))

End If

PU16_23 = Fix(X / 65536) mod 256

PU8_15 = Fix(X / 256) mod 256

PU0_7 = X mod 256

주의 : fix()함수는 소수점을 버리고 정수 부분만 취하는 함수이다.

슬레이브 → 마스터							
장치 주소	기능 코드	시작 레지스터 상위 주소	시작 레지스터 하위 주소	레지스터 개수 상위 바이트	레지스터 개수 하위 바이트	CRC High	CRC Low
0x01	0x10	0x00	0x00	0x00	0x02	0x81	0xcb

d. Modbus 증분 펄스 쓰기

마스터 → 슬레이브							
장치 주소	기능 코드	PU[31~24]	PU[23~16]	PU[15~8]	PU[7~0]	CRC High	CRC Low
0x01	0x78	0x00	0x00	0x27	0x10	0xbb	0xfc

슬레이브 → 마스터							
장치 주소	기능 코드	PU[15~8]	PU[7~0]	PU[31~24]	PU[23~16]	CRC High	CRC Low
0x01	0x78	0x27	0x0e	0x00	0x00	0xca	0xb7

e. Modbus 절대 위치 쓰기 (특수 기능 코드: 0x78)

마스터 → 슬레이브							
장치 주소	기능 코드	PU[31~24]	PU[23~16]	PU[15~8]	PU[7~0]	CRC High	CRC Low
0x01	0x7b	0x00	0x00	0x27	0x10	0xff	0xfc

슬레이브 → 마스터							
장치 주소	기능 코드	PU[15~8]	PU[7~0]	PU[31~24]	PU[23~16]	CRC High	CRC Low
0x01	0x7b	0x27	0x10	0x00	0x00	0xee	0xb1

4. Modbus를 이용한 다축 동기 제어

Modbus RTU를 통해 최대 100대의 서보 모터 동기 제어 가능

Modbus-RTU 통신 쓰기 명령

명령									
1BYTE	1 BYTE	2BYTE		2BYTE		1BYTE	레지스터 개수*8 BYTE	2BYRE	
장치 주소	기능 코드	레지스터 시작 주소		레지스터 개수		데이터 길이	데이터 내용	CRC	
0x00	0x10	0x00	0x16	0x00	0x10	0xXX	见下表	HI	LO

장치 주소: 기본값이 0, 이 경우 모든 주소의 모터가 해당 명령을 수신

기능 코드: 0x10으로 표준 Modbus 기능 코드와 호환되어 다양한 표준 장치 레지스터 주소를 지원

기본 레지스터 시작 주소: 0x16으로 절대 위치 레지스터에 해당

레지스터 개수: 모터 1대를 제어할 경우 0x04, 2대를 제어할 경우 0x08. (이후 모터 수에 비례하여 증가)

데이터 길이: 필요에 따라 임의의 값으로 전송 가능

데이터 내용											
4byte				2byte		2byte		4byte	2byte	2byte	...
1 축 위치				1 축 속도		1 축 속도		2 축 위치	2 축 속도	2 축 가속도	...
32 비트				16 비트		16 비트		32 비트	16 비트	16 비트	...
[15~8]	[7~0]	[31~24]	[23~16]	[15~8]	[7~0]	[15~8]	[7~0]				

데이터 내용 :

각 모터는 자신의 장치 주소에 해당하는 데이터만 자동으로 추출하여 처리한다.

예를 들어 주소 1번 모터의 경우, 데이터 프레임의 앞쪽 8바이트를 사용한다.

데이터 구성은 위표와 같다.

1축 위치: 주소 1번 모터의 절대 위치

1축 속도: 주소 1번 모터의 목표 속도

1축 가속도: 주소 1번 모터의 가속도

```

unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg ; /* CRC 검사를 수행할 메시지 */
unsigned short usDataLen ; /* 메시지의 바이트 수 */
{
    unsigned char uchCRCHi = 0xFF ; /* CRC 상위 바이트 초기값 */
    unsigned char uchCRCLo = 0xFF ; /* CRC 하위 바이트 초기값 */
    unsigned ulIndex ; /* CRC 계산 루프 인덱스 */
    while (usDataLen--) /* 메시지 버퍼 처리 */
    {
        ulIndex = uchCRCHi ^ *puchMsgg++ ; /* CRC 계산 */
        uchCRCHi = uchCRCLo ^ auchCRCHi[ulIndex] ;
        uchCRCLo = auchCRCLo[ulIndex] ;
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}

/* CRC 상위 바이트 값 테이블 */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
};

```

/* CRC 하위 바이트 값 테이블 */

```
static char auchCRCLo[] = {  
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,  
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,  
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,  
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,  
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,  
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,  
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,  
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,  
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,  
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,  
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,  
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,  
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,  
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,  
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,  
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,  
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,  
0x40 }  
}
```

5. Modbus 방식 마스터 제어 절차

a. 위치 모드(Position Mode)

딥스위치 SW1을 OFF로 설정한 후 전원을 인가하면 위치 모드로 동작한다.

전원 인가 후, 제공된 상위 제어 소프트웨어를 사용하여 다음 파라미터를 설정한다.

1. Modbus Enable : 1 전송

(Modbus Enable이 1일 때만 다른 파라미터 설정 가능하며, 외부 펄스 입력은 무효화됨)

HEX 명령: 01 06 00 00 00 01 48 0A

2. 모터 가속도 값 : 5000

(실제 필요에 따라 설정, 미설정 시 기본값 20000 사용)

HEX 명령: 01 06 00 03 13 88 74 9C

3. 목표 속도 : 1500

(미설정 시 기본값 2800 사용)

HEX 명령: 01 06 00 02 05 DC 2A C3

4. 전자기어 분자 : 0

(0으로 저장 시, 다음 전원 인가 시 Modbus Enable 기본값은 1)

HEX 명령: 01 06 00 0A 00 00 A9 C8

5. 파라미터 저장 플래그 : 1

(이전에 설정한 파라미터를 내부 메모리에 저장)

HEX 명령: 01 06 00 14 00 01 08 0E

6. 재부팅 확인

전원을 다시 인가하여 파라미터가 정상적으로 저장되었는지 확인한다.

위 설정은 제공된 상위 제어 소프트웨어로만 설정하면 되며, HEX 명령을 직접 송신할 필요는 없다.

위치 명령 송신

파라미터 설정 완료 후, PLC / MCU / 자체 상위 제어 프로그램을 통해 위치 명령을 송신할 수 있다.

위치 명령은 기능 코드 0x10을 사용하여 전송한다.

1) 증분 위치 명령 (Incremental Position)

증분 위치란, 송신한 데이터만큼 현재 위치 기준으로 전·후진 이동하는 방식이다.

예 1) 정방향 1회전 이동 (엔코더: 1000라인 → 1회전 = 4000펄스)

HEX 명령 : 01 10 00 0C 00 02 04 0F A0 00 00 F0 CC

예 2) 역방향 1회전 이동 (엔코더: 1000라인 → 1회전 = -4000펄스)

-4000의 2의 보수(32비트) 계산 과정은 다음과 같다.

+4000의 16진수 값은 00 00 0F A0이다. (주: 0 = FF FF FF FF + 1)

HEX 명령 : 1 10 00 0C 00 02 04 F0 60 FF FF C1 54

2) 절대 위치 명령 (Absolute Position)

절대 위치란, 전원 인가 직후, 절대 위치 초기화(0 클리어) 수행 후, 또는 자동 원점 복귀 (Home) 완료 후의 위치를 0으로 정의하고,
해당 기준점(0)으로부터 지정한 목표 위치까지 이동하는 방식을 의미한다.

최초로 4000을 전송하면 → 모터는 1회전 이동

이후 이미 위치가 4000인 상태에서 동일한 명령(4000)을 다시 전송하면 → 모터는 움직이지 않는다.

예 1) 모터를 2회전 위치로 이동시키는 경우

(엔코더가 1000라인일 때, 2회전의 펄스 수 = 8000)

HEX 원시 명령: 01 10 00 16 00 02 04 1F 40 00 00 74 89

예 2) 모터를 원점으로 복귀시키는 경우

전자 기어 분자(Electronic Gear Numerator)가 0일 때,
0 값을 전송하면 현재 위치 클리어 명령이 되므로,
원점으로 이동시키기 위해서는 1을 전송한다.
이때 1펄스는 위치 정밀도에 영향을 주지 않는다.

HEX 원시 명령: 01 10 00 16 00 02 04 00 01 00 00 23 49

제어 시 참고 사항:

모터 제어 시에는 먼저 이동할 목표 위치(절대 위치)를 전송하고 절대 위치 읽기 명령을 통해 현재 위치와 목표 위치를 비교하여 다음 명령 실행 여부를 판단한다.

절대 위치 명령은 동일한 명령을 여러 번 반복 전송해도 항상 같은 위치로 이동하므로, 위치 제어에서는 절대 위치 명령 사용을 권장한다.

위치 도달 여부 판단 시에는 ± 2 펄스 오차 허용 범위를 두어야 한다.

또는 PF 신호를 사용할 수 있으며, 목표 위치 도달 시 드라이버에서 포토커플러 출력(ON/OFF 신호)을 제공한다.

절대 위치 읽기 명령: 01 03 00 16 00 02 25 CF

6. 통신 속도(Baud Rate) 변경

통신 속도 변경은 제공된 상위 제어 소프트웨어를 통해서만 가능하며, 아래 순서를 반드시 따른다.

설정 절차:

A. Modbus Enable (주소 0) → 1

B. 모터 가속도 (주소 3) → 803 (803 : 115200 802 : 38400 801 : 19200 800 : 9600)

C. 약자속 각도 (주소 4) → 129

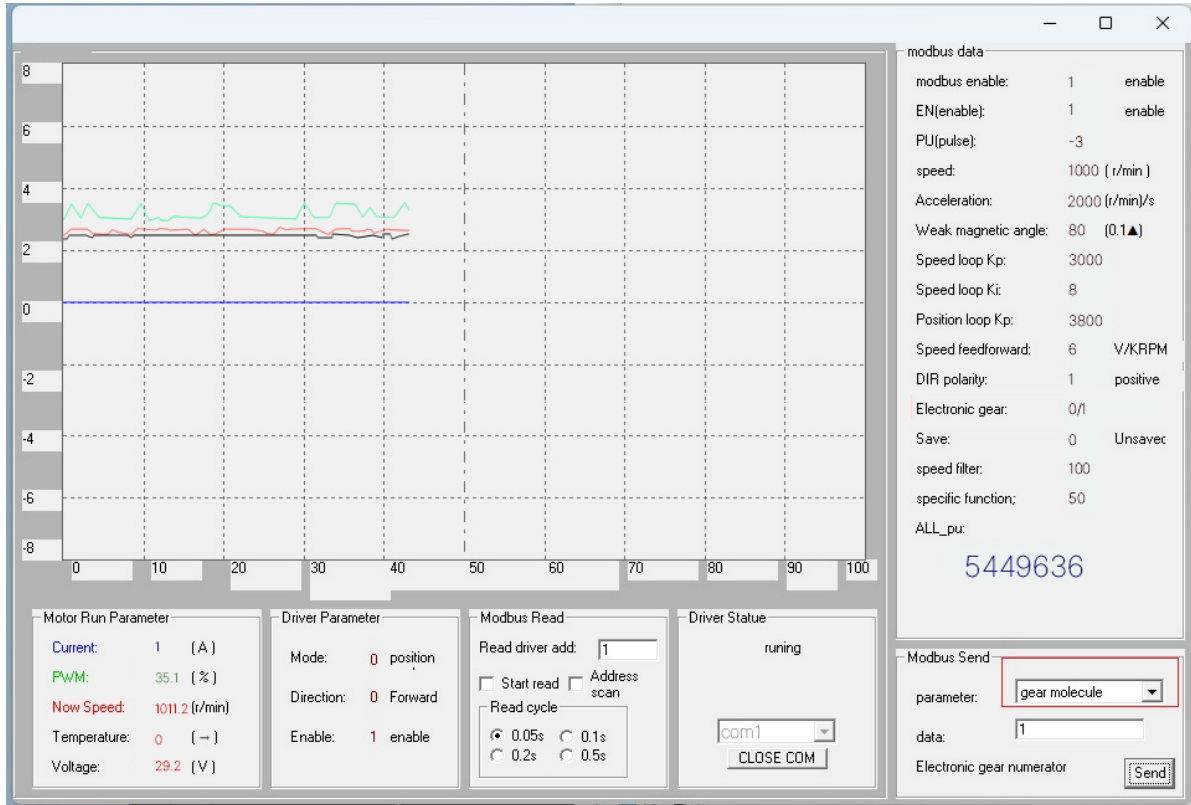
D. Modbus Enable (주소 0) → 506

전원 재인가 후 적용됨

주의: 파라미터 저장 명령은 필요 없음 (내부 파라미터 변경)

6. 상위 제어 소프트웨어 사용 설명

본 드라이버는 전용 상위 제어 소프트웨어를 제공하며,
이를 통해 드라이버 상태 모니터링 및 내부 파라미터의 확인·설정이 가능하다.



그림과 같이, 본 소프트웨어는 파형 표시, 모터 운전 파라미터 등 여러 영역으로 구성되어 있습니다. 아래에서는 각 영역의 기능과 역할에 대해 설명합니다.

파형 표시: 총 4개의 채널로 구성되어 있으며, 각 채널은 서로 다른 색상으로 표시됩니다. 각 색상은 모터 운전 파라미터 영역의 글자 색상과 동일하며, 의미는 다음과 같습니다.

블루: 전류

그린: 출력 펄스 폭

레드: 현재 회전 속도

블랙: 전압

모터 운전 파라미터: 실시간 운전 데이터를 표시하는 영역

드라이버 설정 파라미터: DIP 스위치 설정과 방향 및 인에이블(Enable) 설정을 표시합니다.
Modbus 모드로 동작할 경우, 본 영역은 비활성화됩니다.

드라이버 운전 상태: 드라이버의 알람 상태를 표시.

알람이 발생하지 않은 경우, 정상 운전 상태로 표시

Modbus 제어 파라미터:

본 영역의 파라미터는 드라이버 내부에 저장된 Modbus 제어 파라미터입니다.
해당 파라미터를 변경하려면, 먼저 Modbus 쓰기 인에이블을 1로 설정해야 합니다.
각 파라미터의 상세한 의미는 레지스터 설명을 참고하십시오.

Modbus 읽기:

드라이버의 통신 주소, 데이터 읽기 주기, 그리고 읽기 활성화 여부를 설정하는 영역

Modbus 전송:

드라이버 파라미터를 수정하기 위한 영역

먼저 파라미터 유형을 선택한 후, 파라미터 값을 설정하고 전송(Send) 버튼을 클릭하면 설정이 적용