

SBC-10R4 사용설명서 & RS485 Modbus 프로토콜

1. 제품소개

SBC-10R4제품은 RS485통신, 가변저항기제어가 가능한 스텝모터 전용 드라이버입니다.

2. 제어방식

>> 가변저항기제어:

가변저항기로 속도 제어, 스위치로 브레이크 방향제어 가능합니다. 외부제어 최고속도는 600rpm입니다.

>> RS485통신제어:

스위치를 Int로 설정. RS485의 프로토콜 중 RS485기동 지령을 송신후에 유효합니다. 브레이크와 방향은 RS485를 통해 제어가능합니다.

>>> 직렬통신: 통신속도 기본값 9600, 패리티 확인 안함, 정지비트 1비트.

>>> 입력전압: VCC 12~24V,

>>> 외형사이즈: 42*42mm

3. RS485통신 프로토콜

통신 프레임은 Modbus Rtu 표준프로토콜을 사용하고 있습니다. 해당 드라이버에서 사용가능한 Function code는 3가지가 있으며 0x03, 0x06, 0x10입니다. 하단은 3가지 Function code에 대한 송신 및 피드백 내용입니다.

Function code "03" 레지스터 읽기

>> 마스터 송신: 장치주소(1바이트) + 0x03 + 레지스터 주소(2바이트) + 레지스터 개수(2바이트) + CRC 체크비트(2바이트)

>> 슬레이브 응답: 장치주소(1바이트) + 0x03 + 레지스터 길이(1바이트) + 레지스터 데이터 1(2바이트) + ... + 레지스터 데이터 n(2바이트) + CRC 체크비트(2바이트)

Function code"06" 한개 데이터 쓰기:

>> 마스터 송신: 장치주소(1바이트) + 0x06 + 레지스터 주소(2바이트) + 레지스터 데이터(2바이트) + CRC 체크비트(2바이트)

>> 슬레이브 응답: 장치주소(1바이트) + 0x06 + 레지스터 주소(2바이트) + 레지스터 데이터(2바이트) + CRC 체크비트(2바이트)

Function code"10" 여러 데이터 쓰기

>> 마스터 송신: 장치주소(1바이트) + 0x10 + 레지스터 주소(2바이트) + 레지스터 데이터 개수(2바이트) + 데이터 바이트(1바이트) + 레지스터 데이터1(2바이트) + ... + 레지스터 데이터 n(2바이트) + CRC 체크비트(2바이트)

>> 슬레이브 응답: 장치주소 (1바이트) + 0x10 + 레지스터 주소 (2바이트) + 레지스터 데이터 수량 (2바이트) + CRC 체크 비트 (2바이트)

레지스터 주소 및 기능

레지스터 주소	Read/write	Function code	데이터	기능
0x0020	R/W	0x03/0x06	16bit	모듈 주소 읽기/쓰기(주소수정 및 주소읽기)
0x0021	R/W	0x03/0x10	32bit	통신속도 읽기/쓰기(통신속도설정 및 통신속도읽기)
0x0023	R/W	0x03/0x06	16bit	운전 및 구속전류 읽기/쓰기 통신속도 설정하고 읽는데 사용.
0x0024	R/W	0x03/0x10	32bit	상대위치 읽기/쓰기쓰기
0x0026	R/W	0x03/0x10	32bit	초기속도 읽기/쓰기
0x0028	R/W	0x03/0x10	32bit	운전속도 읽기/쓰기
0x002A	W	0x06	16bit	속도모드로 운전 쓰기(리셋지령) 주요로 정역회전 실행, 위치를 정하지 않은 상태에서 사용, 원점찾기로도 사용
0x002B	R/W	0x03/0x10	32bit	좌표모드 읽기/쓰기(목표위치쓰기, 현재위치읽기) 특정스텝수로 위치를 설정하는데 사용
0x002D	W	0x06	16bit	모터정지
0x002E	R	0x03	16bit	모터상태읽기
0x0030	R/W	0x03/0x06	16bit	분주비읽기

4. 레지스터 주소 설명

레지스터주소에 대한 설명을 주소 0x01을 예시로 설명하겠습니다. 참고로 제어기의 출고설정 기본값은 0x01입니다.

1. 주소 레지스터 0x0020

송신 : 01 06 0020 0002 09C1 제어기주소 1을 주소 2로 변경	
응답 : 02 06 0020 0002 09F2 주소설정성공	
송신지령	설명
0x00	현재 제어기 주소
0x06	Function code 쓰기
0x0020	제어기 주소 레지스터
0x0002	새 주소(주소범위 0x00-0x7F, 해당 범위 초과시 무효)
0x09C1	CRC 체크비트

2. 통신속도 레지스트 0x0021

송신 : 01 10 0021 0002 04 0000 2580 2B4B 설정주소가 1인 모듈의 통신속도는 9600	
응답 : 01 10 0021 0002 11C2 통신속도 설정성공	
송신지령	설명
0x01	현재 제어기 주소
0x10	Function code 데이터 여러 개 쓰기
0x0021	통신속도 레지스트 주소
0x0002	데이터 2개 쓰기 (16비트)
0x04	4 바이트 데이터 쓰기
0x0000 2580	통신속도 9600 (0x2580십진수: 9600). 2400/4800/9600/14400/19200/38400/56000/57600/115200 지원
0x2B4B	CRC 체크비트

3. 운전 및 구속전류 레지스터 0x0023

송신 : 01 06 0023 1002 F401 설정주소가 1인 모듈의 운전전류는 16, 구속전류는 2. 응답 : 01 06 0023 1002 F401 전류설정 성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 쓰기
0x0023	운전전류 및 구속전류 레지스터
0x1002	최대 전류 I _{max} : 2A, high 8비트: 운전 전류 값, low 8비트 구속전류 값. IRUN = (16+1)/32 * I _{max} IHOLD = (2+1)/32 * I _{max} 구속 전류는 1 이상으로 설정하는 것을 권장합니다. 만약 모터 구속이 필요하지 않을 경우 이네이블(Enable) 신호를 클로즈 하시면 됩니다.
0XF401	CRC 체크비트

비고: 운전전류와 구속전류 값은 0-31(0-0x1f) 범위에서 설정할 수 있으며, 0은 최소값으로 (0+1)/32I_{max}, 31은 (31+1)/32I_{max}입니다. 운전전류는 실제 상황에 맞춰 설정하고, 구속전류는 동작 안정성을 유지하는 범위 내에서 가능한 작게 설정하는 것이 좋습니다. 이렇게 하면 모터가 과열되지 않습니다.

4. 상대위치 레지스터 0x0024

송신 : 01 10 0024 0002 04 0000 6400 DA84 설정주소가 1인 모듈 정회전 25600step 응답 : 01 10 0024 0002 01C3 주소1 모듈 설정 성공	
송신지령	설명
0x01	현재 제어기 주소
0x10	Function code 데이터 여러 개 쓰기
0x0024	상대위치 레지스터 주소
0x0002	데이터 2개 쓰기 (16비트)
0x04	4 바이트 데이터 쓰기
0x0000 6400	정회전 25600펄스수(역회전 0x8000 6400)
0xDA84	CRC 체크비트

5. 초기 속도레지스터 0x0026

송신: 01 10 0026 0002 04 0000 C800 265D 설정주소가 1인 모듈의 초기속도는 51200step/s 응답: 01 10 00 2600 02 A003 주소 1 모듈로 설정성공	
송신지령	설명
0x01	현재 제어기 주소
0x10	Function code 쓰기
0x0026	초기속도 레지스터
0x0002	데이터 2개 쓰기 (16비트)
0x04	4 바이트 데이터 쓰기
0x0000 C800	해당모듈의 초기속도는 51200펄스수. 해당 속도는 전원인가후 운전속도를 설정하지 않을 때의 속도입니다, 전원차단후 저장됩니다.
0x265D	CRC 체크비트

6. 운전속도 레지스터 0X0028

송신 : 01 10 0028 0002 04 0000 C800 A7D1 주소1 인 모듈의 운전속도는 51200step/s 응답 : 01 10 0028 0002 C1C0 주소1 모듈로 설정성공	
송신지령	설명
0x01	현재 제어기 주소
0x10	Function code 읽기
0x0028	운전속도 레지스터
0x0002	데이터 2개 쓰기 (16비트)
0x04	4 바이트 데이터 쓰기
0x0000 C800	해당모듈의 운전속도는 51200펄스수.
0xA7D1	CRC 체크비트

7. 속도모드 레지스터 0X002A

송신 : 01 06 002A 0010 A9CE 주소1 인 모듈 정회전 응답 : 01 06 002A 0010 A9CE 주소 1 모듈로 지령 접수성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 읽기
0x002A	속도 레지스터,
0x0010	모터정회전(리셋지령은 하단 표 확인)
0xA9CE	CRC 체크비트

비트	기능	설명(high 비트 앞, low 비트 뒤)
15-13	보류	현 단계 기능 없음
12	원점설정	1: 모터 현재위치를 원점으로 설정 0: 원점설절 하지 않고 정역만 하기
11-5	보류	현 단계 기능 없음
4	모터방향	1: 정회전 0: 역회전
0-3	보류	현 단계 기능 없음

8. 좌표모드 레지스터 0X002B

송신 : 01 10 002B 0002 04 0000 C800 E7C4 주소1인 모터가 원점에서 51200 스텝 이동 응답 : 01 10 002B 0002 31C0 주소 1 모듈로 지령 접수성공	
송신지령	설명
0x01	현재 제어기 주소
0x10	Function code 읽기. 0x03 실시간 좌표위치 읽기
0x002B	좌표모드 레지스터
0x0002	데이터 2개 쓰기 (16비트)
0x04	4 바이트 데이터 쓰기
0x0000 C800	모터가 원점에서부터 51200스텝까지 이동하기(하단표참조)
0xE7C4	CRC 체크비트

비트	기능	설명(high 비트 앞, low 비트 뒤)
31	좌표	1: 원점의 -방향좌표값으로 운전 0: 원점의 +좌표값으로 운전
30-24	보류	현 단계 기능 없음
23-0	모터가 지정된 좌표값으로 이동	스텝모터가 단일적으로 원점에서 이동할수 있는 범위는 최대 0x00FFFFFF스텝임.

9. 정지 레지스터 0X002D

송신 : 01 06 002D 0001 D803 주소1인 모듈 운전 정지 응답 : 01 06 002D 0001 D803 주소 1 모듈로 지령 접수성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 읽기.
0x002D	정지 레지스터
0x0001	0x0001 이네이블 비활성화하고 모터를 구속하지 않을경우 인위적으로 돌릴수 있음 0x0000 이네이블 활성화하면 모터는 구속됨
0xD803	CRC 체크비트

10. 상태 레지스터 0X002E

송신 : 01 03 002E 0001 E403 주소1인 모듈 상태 읽기, 데이터값이 0x0001가 아닌 경우 무효. 응답 : 01 03 02 8020 D85C 주소1인 모듈 모터 정지, 구속전류는 2 , 좌우광커플러 low레벨	
송신지령	설명
0x01	현재 제어기 주소
0x03	Function code 읽기.
0x02	2 바이트 데이터 받기
0x8020	상태값 읽기(하단표참조)
0xD85C	CRC 체크비트

비트	기능	설명(high 비트 앞, low 비트 뒤)
15	정지지시	모터이 표시는 모터가 정지 상태임을 나타냅니다. 이 지시는 마지막 펄스 이후 2^20 클럭에서 적용됩니다. 1: 모터 정지; 0: 모터 작동
14	과열 경고 표시	1: 과열 경고 온도 임계값(120℃)에 도달한 적이 있음
13	과열표시	1: 과열 한계(136℃)에 도달한 적이 있음. 드라이버는 IC가 냉각될 때까지 비활성화됨.
12-9	보류	현 단계 기능 없음
8-4	실제모터전류	주로 현재 감지된 전류 크기는 0-0x1F로, 정지 상태에서는 구속 전류 값을 표기
3-0	보류	현 단계 기능 없음

11. 분주비 레지스터 0X0030

송신 : 01 06 0030 0100 8855 주소1인 모듈의 분주비는 256(0x100)으로 설정. 응답 : 01 06 0030 0100 8855 주소 1 모듈로 지령 접수성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 쓰기.
0x0030	분주비 레지스터
0x0100	0x100를 십진수로 환산하면 256분주비, 분주비는 8가지 지원.
0x8855	CRC 체크비트

5. C언어 CRC 참고용 코드

```
uint16_t crc16bitbybit(uint8_t *data,uint8_t num)
{
    char i;
    int crc = 0xFFFF;
    for ( j = 0; j < num; j++)
    {
        crc ^= data [j];          // XOR byte into least sig. byte of crc
        for (i = 8; i != 0; i--)  // Loop over each bit
        {
            if ((crc & 0x0001) != 0) // If the LSB is set
            {
                crc >>= 1;    // Shift right and XOR 0xA001
                crc ^= 0xA001;
            }
            else                // Else LSB is not set
            {
                crc >>= 1;    // Just shift right
            }
        }
    }
    //    // HIGH,LOW바이트 전환

    crc = ((crc & 0x00ff) << 8) | ((crc & 0xff00) >> 8);
    return  crc
}
```