

BLC-20R4 사용설명서 & RS485 Modbus 프로토콜

1. 제품소개

BLC-20R4제품은 RS485통신,외부PWM제어,가변저항기제어가 가능한 구동 드라이버 내장형 BLDC모터의 전용 드라이버입니다.

2. 제어방식

>> 외부PWM제어:

스위치를 Ext로 설정. 아두이노로 제어가능. 전압범위 0~5V

>> 가변저항기제어:

스위치를 Int로 설정. 가변저항기로 속도 제어, 스위치로 브레이크 방향제어 가능합니다.

>> 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, 전류<5A

>>> 외형사이즈: 40*40mm

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	RS485 통신 모드 기동. RS485 통신시 꼭 송신해야 함.
0x0024	R/W	0x03/0x06	16bit	속도설정. 듀티비 0~100% 주파수 20khz
0x0025	R/W	0x03/0x06	16bit	방향설정 1CW / 2CCW
0x0026	R/W	0x03/0x06	16bit	브레이크설정. Low 레벨 0, High 레벨 1.

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 0x0000-0x007F, 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. RS485통신 모드 기동, 0x0023

송신 : 01 06 0023 0000 7800 RS485통신모드 기동, 초기브레이크는 Low레벨, 속도 0% 응답 : 01 06 0023 0000 7800 RS485통신모드 기동 성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 쓰기
0x0023	RS485통신모드 기동 레지스터
0x0000	0x0000 : 초기브레이크Low레벨,속도PWM 0% 0x0001 : 초기브레이크Low레벨,속도PWM 100% 0x0002 : 초기브레이크Hige레벨,속도PWM 0% 0x0003 : 초기브레이크Hige레벨,속도PWM 100%
0x7800	CRC 체크비트

비고: 초기상태는 모터의 사양에 따라 다릅니다. 전원인가후 바로 구동이 되는 모터도 있고 전원인가후 기동지령을 송신한 후에야 기동하는 모터도 있습니다. 구동드라이버 내장형의 모터일 경우 메이커에 따라 설정이 다르기에 상황에 따라 사용하시기 바랍니다.

4. 속도 레지스터 0x0024

송신 : 01 06 0024 0064 C82A 설정주소가 1인 모터 속도의 PWM은 100% 응답 : 01 06 0024 0064 C82A 주소1 모듈 설정 성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 데이터 여러 개 쓰기
0x0024	속도 레지스터 주소
0x0064	PWM 듀티비 0~100% 주파수 20khz
0x7800	CRC 체크비트

5. 방향 레지스터 0x0025

송신: 01 06 0025 0001 59C1 설정주소가 1인 모터방향 CW 응답: 01 06 0025 0001 59C1 주소 1 모듈로 설정성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 쓰기
0x0025	방향 레지스터
0x0001	0x0001: CCW / 0x0002: CW
0x52C1	CRC 체크비트

6. 브레이크 레지스터 0X0026

송신 : 01 06 0026 0001 A9C1 주소1 모터의 브레이크 High레벨 응답 : 01 06 0026 0001 A9C1 주소 1 모듈로 설정성공	
송신지령	설명
0x01	현재 제어기 주소
0x06	Function code 읽기
0x002F	브레이크 레지스터,
0x0005	0x0000: Low레벨 / 0x0001: Hige레벨
0x7800	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
}
```

6. 아두이노 테스트 코드

```
//Arduino Uno
//PWM, Direction, Brake
//V3.1

#include <TimerOne.h>
#include <MsTimer2.h>

int PWM = 9; //PWM, Pin 9
int Direction = 4; //Direction, Pin 4
int Brake = 2; //Brake, Pin 2
int Slope = 5; //Slope step 1~10
volatile int DutyRef = 0; //PWM duty reference -1024~1024(-100%~100%)
volatile int Duty = 0;

void flash() {
    if(Duty < DutyRef) Duty++;
    else if(Duty > DutyRef) Duty--;

    if(Duty > 0) {
        Timer1.setPwmDuty(PWM, 1024-Duty); //PWM, Duty
        digitalWrite(Direction, LOW); //Direction, '0'
    }
    else if(Duty < 0) {
        Timer1.setPwmDuty(PWM, 1024+Duty); //PWM, Duty
        digitalWrite(Direction, HIGH); //Direction, '1'
    }
}

void setup() {
    //PWM
    Timer1.initialize(); //Timer1
    Timer1.pwm(PWM, 0); //PWM, Duty 0%
    Timer1.setPeriod(50); //PWM period 20kHz
    Timer1.setPwmDuty(PWM, 0); //PWM, Duty 0%

    //Direction
    pinMode(Direction, OUTPUT); //Direction, Output
```

```

digitalWrite(Direction, LOW); //Direction, '0'

//Brake
pinMode(Brake, OUTPUT); //Brake, Output
digitalWrite(Brake, HIGH); //Brake, '0'

//Timer interrupt
MsTimer2::set(10/Slope, flash); //Timer interrupt period
MsTimer2::start(); //Timer interrupt start
}

void loop() {
    DutyRef = 1000; //PWM duty reference -1024~1024(-100%~100%)
    delay(5000); //Delay 5s

    DutyRef = -1000; //PWM duty reference -1024~1024(-100%~100%)
    delay(5000); //Delay 5s
}

```

비고:

**테스트 시 int Slope = 5; //Slope step 1~10 Slope는 연결하지 않고 테스트를 진행한점 참고바랍니다.