

Modbus Protocol 개요

본 문서는 GUB BLDC Motor Driver에서 사용되는 Modbus RTU protocol의 설명을 진행한다.
Modbus protocol과 Modbus RTU protocol에 대한 상세한 정의는 www.modbus.org에
설명되어 있다.

Modbus RTU protocol은 RS-485나 RS-232등과 같이 serial 통신 환경에서 동작하기 위한
Modbus protocol의 한 종류이다. 이 protocol은 장치 ID를 통하여 각 장치를 구분하고 CRC를
이용하여 에러를 확인한다.

GUB는 User Input/Output 기능(Run스위치, CW/CCW입력 등)과 RS-485기능을 지원한다. RS-485기능으로 사용하는 경우는 내부에 통신모듈(옵션)이 부착되어 있어야 하고, Select 스위치를 RS-485통신으로 선택을 한 후, 전원을 재 투입하여야 한다. Select 스위치로 기능을 지정하는데 전원 오프->온 시 스위치 설정을 확인하고 User Input/Output로 동작을 할 것인지 RS-485통신 모드로 동작할 것인지를 결정한다. 단, 장치 ID (Slave Address) 설정은 User Input/Output 모드에서 설정할 수 있다.

Modbus RTU 통신설정

1. 접속규격(Interface) : RS-485
2. 전송속도(Baud Rate) : 19200 bps
3. 데이터 형식 (Data Format) : 1 Start bit, 8 Data bits, 1 Stop bits(None Parity)
4. 전송방식(Transmit Method) : 반 2중 조보 동기(Half Duplex Asynchronous)
5. 에러검출(Error Checking) : CRC16(Cyclical Redundancy Check) 방식

Modbus RTU Protocol

1. 전송메시지의 구성

Slave Address	Function	Data	CRC 코드
8 bits	8 bits	n × 8 Bits	16 bits

Slave Address : 질의 메시지를 수신하고 응답메시지를 송신할 슬레이브의 주소를 나타내며 숫자 1~99의 값을 가지는 데이터가 전송된다. 출하시 Slave Address는 1로 설정되어 있다.

기능 코드 : 전송되는 메시지의 기능을 나타내는 코드로 숫자 1 ~ 255 범위 내로 표현된다. 질의(명령) 메시지를 수신하여 슬레이브 계기가 에러 없이 수행하고 응답할 때 같은 기능코드가 전송된다. GUB가 지원하는 기능코드는 다음과 같다.

기능 코드	설 명	레퍼런스 타입 (레지스터 주소범위)	코드 사용
03	Read Holding Regiaters	0x (40001-09999)	GUB Parameter Read
04	Read Input Registers	3x (30001-39999)	GUB 현재상태 모니터링
06	Write Single Register	0x (40001-39999)	Parameter Setting 및 Command

<기능 코드 표>

Data : 메시지의 기능코드 코드에 따라 내용과 길이가 달라지는 데이터 값들로서 여러 개의 8bits 그룹이다. 0~255의 값을 가지는 데이터가 전송되며, 0~65535(16비트)의 값을 가지는 데이터는 먼저 상위 8비트가 전송되고 나머지 하위 8비트가 전송된다.

CRC 코드 : 통신이 에러 없이 전송되었는지 여부를 확인하기 위한 에러 검출용 코드로서 전송 메시지의 끝에 추가되어 수신측이 수신 데이터를 1바이트씩 수신할 때마다 계산한 값과 비교하여 수신 에러여부를 판단한다.

2. Read Holding Registers (Func 03-03H)

슬레이브 디바이스 내, Holding Registers(4X 레퍼런스)의 Binary 데이터를 읽을 수 있습니다. 드라이버의 Parameter 값을 읽을 때 사용 합니다

➤ Request(Master측)

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

➤ Response(Slave측)

Slave Address	Function	N (Data Byte Count)	N x Data		CRC16	
			High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

Master측에서 Slave(Address 1)측의 Holding Register 40001(0000) ~ 40002(0001)내, 2EA의 값을 읽고자 할 경우의 예입니다

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x03	0x00	0x00	0x00	0x02	0xC4	0x0B

Slave측의 40001(0000)번의 값이 "555(22B H)"이고 40002(0001)번의 값이 "100(64 H)"일 경우의 예입니다.

Slave Address	Function	N	Data		Data		CRC16	
			High	Low	High	Low	Low	High
0x01	0x03	0x04	0x02	0x2B	0x00	0x64	0x8A	0x68

3. Read Input Registers (Func 04-04H)

슬레이브 디바이스 내, Input Registers(3X 레퍼런스)의 Binary 데이터를 읽을 수 있습니다.
드라이버의 상태 값을 읽을 때 사용 합니다

➤ Request(Master측)

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

➤ Response(Slave측)

Slave Address	Function	N	Data		CRC16	
			High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

Master측에서 Slave(Address 1)측의 Input Register 30001(0000) ~ 30002(0001)내, 2EA의 값을 읽고자 할 경우의 예입니다

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x04	0x00	0x00	0x00	0x02	0x71	0xCB

Slave측의 30001(0000)번의 값이 "10(A H)"이고 30002(0001)번의 값이 "20(14 H)"일 경우의 예입니다.

Slave Address	Function	N	Data		Data		CRC16	
			High	Low	High	Low	Low	High
0x01	0x04	0x04	0x00	0x0A	0x00	0x14	0xDB	0x89

4. Write Single Holding Registers (Func 06-06H)

슬레이브 디바이스 내, 단일 Holding Registers(4X 레퍼런스)의 Binary 데이터를 씁니다.
드라이버의 Parameter에 값을 쓸 때 사용합니다

➤ Request(Master측)

Slave Address	Function	Address		Data		CRC16	
		High	Low	High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

➤ Response(Slave측)

Slave Address	Function	Address		Data		CRC16	
		High	Low	High	Low	Low	High
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

Master측에서 Slave(Address 1)측의 Holding Register 40001(0000) ~ 30002(0001)에,
"10(A H)" 값을 쓰는 경우의 예입니다

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x06	0x00	0x00	0x00	0x0A	0x09	0xCD

Slave측의 Response입니다.

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x06	0x00	0x00	0x00	0x0A	0x09	0xCD

Modbus RTU Mapping Table

1. 드라이버 현재상태 (Func:04, RW:R)

드라이버의 현재상태를 읽는 영역입니다.

Address	Name	설정범위	단위	설명
30001(0000)	드라이버 상태			표1참고
30002(0001)	드라이버 알람			표2참고
30003(0002)	지령속도		rpm	
30004(0003)	현재속도		rpm	
30005(0004)				
30006(0005)	현재전류			Current A/D
30007(0006)			-	
30008(0007)	온도		-	Power Module NTC
30009(0008)	전압		-	DC Link Voltage
30010(0009)	명령		-	표3참고
30011(0010)	지령속도		Rpm	Note1

표1) 드라이버 상태

Bit	Name	설명
Bit 0	BRK	모터 브레이크
Bit 1	FRE	모터 프리
Bit 2	ALM	알람 상태
Bit 3	EMG	비상정지
Bit 4	DEC	감속 중
Bit 5	ACC	가속 중
Bit 6	DIR	0:CW, 1:CCW
Bit 7	RUN	0:STOP, 1:RUN

표2) 드라이버 알람

알람번호	설명	알람번호	설명
0	알람 없음	4	과부하
1	-	5	-
2	과전류	6	과전압
3	홀센서 이상	7	Over Temperature

표3) 명령 리스트

명령	설명	명령	설명
0	None	7	-
1	운전 시작	8	-
2	운전 정지	9	-
3	-	10	-
4	알람 리셋	11	모터 프리 정지
5	-	12	모터 브레이크 정지
6	-	13	-

2. Parameter (Func:03/06, RW:R/W)

드라이버의 파라미터를 읽고 쓰는 영역입니다.

Address	Name	설정범위	단위	설명
40001(0000)	모터용량			
40002(0001)	극수			Read Only
40003(0002)				
40004(0003)				
40005(0004)	정격속도	3000~4000	rpm	
40006(0005)				
40007(0006)				
40008(0007)				
40009(0008)				
40010(0009)				
40011(0010)	전류제한			Read Only
40012(0011)				
40013(0012)				
40014(0013)				
40015(0014)				
40016(0015)				
40017(0016)				
40018(0017)				
40019(0018)				
40020(0019)				
40021(0020)				
40022(0021)				
40023(0022)				
40024(0023)				
40025(0024)				
40026(0025)				
40027(0026)				
40028(0027)				
40029(0028)	드라이버 주소	1~99		MODBUS Slave 주소
40030(0029)	모터 정지 타입	0~3		0:감속-프리, 1:프리, 2:감속-브레이크, 3:브레이크
40031(0030)				
40121(0120)	명령			
40122(0121)	지령속도		rpm	Note1

Note1:지령속도 값은 +-부호로 회전방향을 바꿉니다. 예)-1000:CCW, 1000:CW

통신속도 : 19200bps

3. CRC16 Calculation

```

int Crc16Table[256] = {
    0x0000, 0xC0C1, 0x0181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
    0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
    0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
    0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
    0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

unsigned short uiCRC16(unsigned char *buf, int len)
{
    unsigned short CRC = 0xFFFF;
    int i,tmp;
    for(i=0; i<len; i++) {
        tmp = CRC ^ (0x00ff & buf[i]);
        CRC = (CRC>>8) ^ Crc16Table[tmp & 0xff];
    }
    return CRC;
}

```


Modbus RTU 설정방법 및 예제

1. 통신제어 활성화 방법

드라이버의 커버를 제거하고 내부에 통신모듈(옵션)이 있는지 확인



2. 통신제어 활성화 방법

GUB 내부 스위치를 좌측 통신쪽으로 설정 후 드라이버 전원 리셋



통신 제어 ← → I/O 제어

● 드라이버 번호 (Slave Address) 셋팅

드라이버 전원을 Off하고 GUB 내부 스위치를 우측 I/O제어 쪽으로 설정 후 드라이버 전원을 ON 한다. 485통신 Message C9 06 00 1C 00 NN CRC16 로 설정이 가능하다.

NN : Slave Address (1~99)

Parameter 40029(0028) 변경 (Default : 01) : 통신 제어 모드에서는 설정이 불가능 함

3. 통신 환경설정

- 통신 방식 : RS485 Half Duplex
- Data Bits : 8bit
- Stop Bits : 1bit
- Parity : None
- Baud Rate : 19200bps

4. 현재 모터 속도 읽기

➔ 아래 예제 참조

5. 드라이버 알람 읽기 및 알람 리셋

➔ 아래 예제 참조

6. 예제

- 모터 속도 읽기

Command : Master → Slave

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x04	0x00	0x03	0x00	0x01	0xC1	0xCA

Response : Slave → Master

Slave Address	N	Data		CRC16	
		High	Low	Low	High
0x01	0x02	0x03	0xE8	0xA0	0xA6

*모터 속도가 1000인 경우

- 알람 상태 읽기

Command : Master → Slave

Slave Address	Function	Starting Address		No. of Points		CRC16	
		High	Low	High	Low	Low	High
0x01	0x04	0x00	0x01	0x00	0x01	0x60	0x0A

Response : Slave → Master

Slave Address	N	Data		CRC16	
		High	Low	Low	High
0x01	0x02	0x00	0x00	0xA0	0x18

[알람 데이터 값 정의]

0x00 : 알람 없음

0x01 : Reserved

0x02 : 과전류

0x03 : 홀센서 이상

0x04 : 과부하

0x05 : Reserved

0x06 : 과전압

0x07 : 과열

- 모터 제어 명령

구분	DRV ID	Func	Address		Data		CRC16	
드라이버 ON	0x01	0x06	0x00	0x78	0x00	0x01	0xC8	0x13
드라이버 OFF	0x01	0x06	0x00	0x78	0x00	0x00	0x09	0xD3
모터브레이크 ON	0x01	0x06	0x00	0x78	0x01	0x01	0xC9	0x83
모터브레이크 OFF	0x01	0x06	0x00	0x78	0x01	0x00	0x08	0x43
알람 리셋	0x01	0x06	0x00	0x78	0x02	0x01	0xC9	0x73

- 정방향 속도 명령(CW)

구분	DRV ID	Func	Address		Data		CRC16	
0 rpm	0x01	0x06	0x00	0x79	0x00	0x00	0x58	0x13
1000 rpm	0x01	0x06	0x00	0x79	0x03	0xE8	0x58	0xAD
2000 rpm	0x01	0x06	0x00	0x79	0x07	0xD0	0x5B	0xBF
3000 rpm	0x01	0x06	0x00	0x79	0x0B	0xB8	0x5F	0x51

- 역방향 속도 명령(CW)

구분	DRV ID	Func	Address		Data		CRC16	
-1000 rpm	0x01	0x06	0x00	0x79	0xFC	0x18	0x19	0x19
-2000 rpm	0x01	0x06	0x00	0x79	0xF8	0x30	0x1B	0xC7
-3000 rpm	0x01	0x06	0x00	0x79	0xF4	0x48	0x1E	0xE5