



OZBV Ver2.0 시리즈 BLDC 드라이버 통신 프로토콜

목차

1. 소개

2. ModBus RTU 프로토콜

- 2.1 Read Coil (Func 01-01H)
- 2.2 Read Discretes Input (Func 02-02H)
- 2.3 Read Holding Registers (Func 03-03H)
- 2.4 Read Input Registers (Func 04-04H)
- 2.5 Write Single Coil (Func 05-05H)
- 2.6 Write Single Holding Registers (Func 06-06H)
- 2.7 예외 처리(Exception Response-Error Code)

3. OZBP Modbus RTU Mapping Table

- 3.1 출력 IO 상태 및 출력 (Func: 01/05, RW: R/W)
- 3.2 입력 IO 상태 (Func: 02, RW: R)
- 3.3 드라이버 현재 상태 (Func: 04, RW: R)
- 3.4 파라미터 (Func: 03/06, RW: R/W)
 - 3.4.1 Control Setting 그룹
 - 3.4.2 Command 그룹

4. CRC-16 Calculation

1. 소개

OZBV 시리즈 드라이버 통신 프로토콜은 개방형 프로토콜인 Modicon사의 MODBUS-RTU 방식을 지원합니다.

상세 프로토콜은 <http://www.modbus.org/> 를 참조 하십시오.

통신 설정

- Communication Port : RS-232 or RS-422 방식
- Baud rate : 9600BPS
- Data Bit : 8Bit
- Stop Bit : 1Bit
- Parity : None

드라이버는 아래와 같이 4개의 Primary tables 영역을 가지고 있습니다.

Function code	Primary tables	Objecct type	Type of access	Comments
02 H	Discretes Input	Single bit	Read-Only	드라이버 입력 IO 상태
01 H 05 H	Coil	Single bit	Read-Write	드라이버 출력 IO 상태 및 출력
04 H	Input Registers	16-bit Word	Read-Only	드라이버 상태
03 H 06 H	Holding Registers	16-bit Word	Read-Write	드라이버 Parameter

4개의 Primary tables 영역은 각각의 Function code 로 Read-Write 할 수 있습니다.

OZBV Drive는 **브로딩캐스팅**을 지원하지 않습니다. (국번 0)

OZBV Drive는 Function : **0x0F**(Write Multi Coil), **0x10**(Write Multi Registers) 을 지원하지 않습니다.

Master/Slave 방식으로서 Master에 의해서 전송된 모든 Packet은 Request가 되고 Slave에 의해서 전송된 모든 Packet은 Response가 된다.

Master : PC or 상위제어기

Slave : OZBV 드라이버

Modbus RTU PC 용 소스는 당사 홈페이지 자료실에서 다운 받을 실 수 있습니다.

Note : 상세 파라미터 에서 설명하지 않은 파라미터는 사용자가 임의로 변경하지 마십시오.

2. ModBus RTU 프로토콜

2.1 Read Coil (Func 01-01H)

슬레이브 디바이스 내, 출력IO(OX 레퍼런스)의 ON/OFF 상태를 읽습니다.

드라이버의 출력 IO 상태를 읽을 때 사용합니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)	Data (데이터)	Data (데이터)	CRC16	
						Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

Master측에서 Slave(Address 1)측의 코일 00001(0000) ~ 00010(0009)내 10EA의 출력상태(ON : 1, OFF :0)를 읽고자 할 경우의 예입니다

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	01 H	00 H	00 H	00 H	0A H	## H	## H

Slave측의 코일00008(0007) ~ 00001(0000)번의 값 "ON-ON-OFF-OFF-ON-ON-OFF-ON"이고 00010(0009)~00009(0008)번의 값이 "OFF-ON"일 경우의 예입니다

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (00008 ~ 00001)	Data (00010 ~ 00009)	CRC16	
					Lo(하위)	Hi(상위)
01 H	01 H	02 H	CD H	01 H	## H	## H

2.2 Read Discretes Input (Func 02-02H)

슬레이브 디바이스 내, 입력IO(1X 레퍼런스)의 ON/OFF 상태를 읽습니다.

드라이버의 입력 IO 상태를 읽을 때 사용합니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)	Data (데이터)	Data (데이터)	CRC16	
						Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

Master측에서 Slave(Address 1)측의 10001(0000)~10010(0009)내 10EA의 입력상태(ON: 1, OFF: 0)를 읽고자 할 경우의 예입니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	02 H	00 H	00 H	00 H	0A H	## H	## H

Slave측의 10008(0007) ~ 10001(0000)번의 값이 "ON-ON-OFF-OFF-ON-ON-OFF-ON"이고 10010(0009)~10009(0008)번의 값이 "OFF-ON"일 경우의 예입니다.

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (00008 ~ 00001)	Data (00010 ~ 00009)	CRC16	
					Lo(하위)	Hi(상위)
01 H	02 H	02 H	CD H	01 H	## H	## H

2.3 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 (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)		Data (데이터)		CRC16	
			Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	
CRC16								

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

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	03 H	00 H	00 H	00 H	02 H	## H	## H

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

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)		Data (데이터)		CRC16	
			Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	03 H	04 H	02 H	2B H	00 H	64 H	## H	
CRC16								

2.4 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 (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)		Data (데이터)		CRC16	
			Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	
CRC16								

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

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	04 H	00 H	00 H	00 H	02 H	## H	## H

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

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Byte Count (데이터 byte 수)	Data (데이터)		Data (데이터)		CRC16	
			Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	04 H	04 H	00 H	0A H	00 H	14 H	## H	
CRC16								

2.5 Write Single Coil (Func 05-05H)

슬레이브 디바이스 내, 출력 IO(0X 레퍼런스)의 ON/OFF 데이터를 씁니다.

드라이버 출력 IO 에 On/Off 값을 쓸 때 사용 합니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
CRC16							

Master측에서 Slave(Address 1)측의 코일 00001(0000) 을 ON 으로 설정 할 경우의 예입니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	05 H	00 H	00 H	FF H	00 H	## H	## H

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	05 H	00 H	00 H	FF H	00 H	## H	## H

Master측에서 Slave(Address 1)측의 코일 00001(0000) 을 OFF 으로 설정 할 경우의 예입니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	05 H	00 H	00 H	00 H	00 H	## H	## H

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	05 H	00 H	00 H	00 H	00 H	## H	## H

2.6 Write Single Holding Registers (Func 06-06H)

슬레이브 디바이스 내, 단일 Holding Registers(4X 레퍼런스)의 Binary 데이터를 씁니다.

드라이버 Parameter 에 값을 쓸 때 사용 합니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16		
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)	
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	
CRC16								

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16		
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)	
1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	
CRC16								

Master측에서 Slave(Address 1)측의 Holding Register 40001(0000)에 "10(A H)"을 쓰고자 할 경우의 예입니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	06 H	00 H	00 H	00 H	0A H	## H	## H

• Response(Slave 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		Data (데이터)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	06 H	00 H	00 H	00 H	0A H	## H	## H

2.7 예외 처리(Exception Response-Error Code)

통신에러가 발생한 경우, 수신한 명령(Function)의 최상의 비트를 세트(1) 후, 응답 명령을 보내고 해당 Exception Code를 전송합니다.

Slave Address (국번)	Function (명령) + 80 H	Exception Code	CRC16	
			Lo(하위)	Hi(상위)
1Byte	1Byte	1Byte	1Byte	1Byte
CRC16				

- ILLEGAL FUNCTION (Exception Code: 01 H): 지원하지 않는 명령일 경우.
- ILLEGAL DATA ADDRESS (Exception Code: 02 H): 요청한 데이터의 시작번지가 장치에서 전송할 수 있는 번지와 불일치할 경우.
- ILLRGAL DATA VALUE (Exception Code: 03 H): 요청한 데이터의 개수가 장치에서 전송할 수 있는 개수와 불일치할 경우.
- SLAVE DEVICE FAILURE (Exception Code : 04 H): 요청 받은 명령을 정상적으로 처리하지 못할 경우.

Master측에서 Slave(Address 1)측의 존재하지 않는 코일 01001(03E8 H)의 출력 상태(ON: 1, OFF: 0)를 읽고자 할 경우의 예입니다.

• Request (Master 측)

Slave Address (국번)	Function (명령)	Starting Address(시작번지)		No. of Points(데이터 개수)		CRC16	
		Hi(상위)	Lo(하위)	Hi(상위)	Lo(하위)	Lo(하위)	Hi(상위)
01 H	01 H	03 H	E8 H	00 H	01 H	## H	## H

• Response(Slave 측)

Slave Address (국번)	Function (명령) + 80 H	Exception Code	CRC16	
			Lo(하위)	Hi(상위)
01 H	81 H	02 H	## H	## H

3. OZBV Modbus RTU Mapping Table

3.1 출력 IO 상태 및 출력 (Func: 01/05, RW: R/W)

드라이버의 출력IO 상태를 읽거나, 쓰는 영역입니다.

Slave Address	Name	설정 범위	단위	설명
00001(0000)	OUT 1	0:OFF, 1:ON	-	OUT 1 출력상태 및 출력
00002(0001)	OUT 2	0:OFF, 1:ON	-	OUT 2 출력상태 및 출력
00003(0002)	OUT 3	0:OFF, 1:ON	-	OUT 3 출력상태 및 출력
00004(0003)	Reserved	-	-	-
00005(0004)	Reserved	-	-	-
...				
00016(0015)	Reserved	-	-	-

3.2 입력 IO 상태 (Func: 02, RW: R)

드라이버의 입력IO 상태를 읽는 영역입니다.

Slave Address	Name	설정 범위	단위	설명
10001(0000)	IN 1	0:OFF, 1:ON	-	IN 1 입력상태
10002(0001)	IN 2	0:OFF, 1:ON	-	IN 2 입력상태
10003(0002)	IN 3	0:OFF, 1:ON	-	IN 3 입력상태
10004(0003)	IN 4	0:OFF, 1:ON	-	IN 4 입력상태
10005(0004)	Reserved	-	-	-
10006(0005)	Reserved	-	-	-
10007(0006)	Reserved	-	-	-
10008(0007)	Reserved	-	-	-
10009(0008)	Reserved	-	-	-
...				
10016(0015)	Reserved	-	-	-

3.3 드라이버 현재 상태 (Func: 04, RW: R)

드라이버의 현재 상태를 읽는 영역입니다.

Slave Address	Name	설정 범위	단위	설명
30001(0000)	드라이버 상태		-	표1 참조
30002(0001)	드라이버 알람		-	표2 참조
30003(0002)	지령속도		rpm	
30004(0003)	현재속도		rpm	
30005(0004)	지령전류		mA	
30006(0005)	현재전류		ma	
30007(0006)	외부볼륨		-	
30008(0007)	Reserved	-	-	
30009(0008)	Reserved	-	-	
30010(0009)	Reserved	-	-	
30011(0010)	Reserved	-	-	
30012(0011)	Reserved	-	-	
30013(0012)	Reserved	-	-	
30014(0013)	Reserved	-	-	
30015(0014)	Reserved	-	-	
30016(0015)	Reserved	-	-	
30017(0016)	Reserved	-	-	
...				
10032(0031)	Reserved	-	-	

표 1) 드라이버 상태

Bit	Name	설명
Bit 0	BRK	모터브레이크
Bit 1	FRE	모터프리
Bit 2	ALM	알람상태
Bit 3	EMG	비상정지
Bit 4	DEC	감속중
Bit 5	ACC	가속중
Bit 6	DIR	0 : CW, 1 : CCW
Bit 7	RUN	0 : STOP, 1 : RUN

표 2) 드라이버 알람

알람 번호	설명	알람 번호	설명
0	알람 없음	8	모터 단시간운전 에러
1	비상 정지	-	-
2	과전류 에러	-	-
3	온도 에러	-	-
4	Hall 센서 에러	-	-
5	과전압 에러	-	-
6	저전압 에러	-	-
7	모터 과부하	-	-

3.4 파라미터 (Func: 03/06, RW: R/W)

드라이버의 파라미터를 읽고 쓰는 영역입니다.

파라미터의 자세한 설명은 User Manual 의 상세 파라미터 를 참조 하십시오.

3.4.1 Control Setting 그룹

Slave Address	Name	설정 범위	단위	설명
40001(0000)	Reserved	-	-	-
40002(0001)	Reserved	-	-	-
40003(0002)	Reserved	-	-	-
40004(0003)	Reserved	-	-	-
40005(0004)	Reserved	-	-	-
40006(0005)	Reserved	-	-	-
40007(0006)	Reserved	-	-	-
40008(0007)	Reserved	-	-	-
40009(0008)	Reserved	-	-	-
40010(0009)	Reserved	-	-	-
40011(0010)	전류 루프 사용	0 ~ 1	-	0: 사용안함, 1: 사용 함
40012(0011)	속도 루프 사용	0 ~ 1	-	0: 사용안함, 1: 사용 함
40013(0012)	Reserved	-	-	-
40014(0013)	전류 P Gain	0 ~ 32000	-	-
40015(0014)	전류 I Gain	0 ~ 32000	-	-
40016(0015)	속도 P Gain	0 ~ 32000	-	-
40017(0016)	속도 I Gain	0 ~ 32000	-	-
40018(0017)	Reserved	-	-	-
40019(0018)	Reserved	-	-	-
40020(0019)	Reserved	-	-	-
40021(0020)	가속 시간	1 ~ 20000	msec	-
40022(0021)	감속 시간	1 ~ 20000	msec	-
40023(0022)	최저 속도	1 ~ 3000	rpm	-
40024(0023)	최고 속도	1000 ~ 30000	rpm	-
40025(0024)	Reserved	-	-	-
40026(0025)	전류 제한	100 ~ 10000	mA	-
40027(0026)	전류 제한 한계 시간	10 ~ 10000	msec	-
40028(0027)	과온도 한계	50 ~ 100	도(C)	-
40029(0028)	과온도 한계 시간	10 ~ 10000	msec	-
40030(0029)	전원 변동 한계 시간	10 ~ 10000	msec	-
40031(0030)	정지 방법	0 ~ 3	-	상세 파라미터 참조
40032(0031)	긴급정지방법	0 ~ 3	-	상세 파라미터 참조
40033(0032)	Reserved	-	-	-
40034(0033)	Reserved	-	-	-
40035(0034)	Reserved	-	-	-
40036(0035)	Reserved	-	-	-
40037(0036)	Reserved	-	-	-
40038(0037)	Reserved	-	-	-

OZBV 통신 프로토콜

Slave Address	Name	Value	Unit	Explanation
40039(0038)	속도 도달 허용 폭	0 ~ 1000	rpm	-
40040(0039)	Reserved	-	-	-
40041(0040)	IN 극성 선택	0 ~ 15	-	상세 파라미터 참조
40042(0041)	OUT 극성 선택	0 ~ 7	-	"
40043(0042)	운전모드 선택	0 ~ 3	-	"
40044(0043)	IN 1 기능 선택	0 ~ 11	-	"
40045(0044)	IN 2 기능 선택	0 ~ 11	-	"
40046(0045)	IN 3 기능 선택	0 ~ 11	-	"
40047(0046)	IN 4 기능 선택	0 ~ 11	-	"
40048(0047)	OUT 1 기능 선택	0 ~ 7	-	"
40049(0048)	OUT 2 기능 선택	0 ~ 7	-	"
40050(0049)	OUT 3 기능 선택	0 ~ 7	-	"
40051(0050)	속도 인덱스 0	-	rpm	상세 파라미터 참조
40052(0051)	속도 인덱스 1	-	rpm	"
40053(0052)	속도 인덱스 2	-	rpm	"
40054(0053)	속도 인덱스 3	-	rpm	"
40055(0054)	속도 인덱스 4	-	rpm	"
40056(0055)	속도 인덱스 5	-	rpm	"
40057(0056)	속도 인덱스 6	-	rpm	"
40058(0057)	속도 인덱스 7	-	rpm	"
40059(0058)	Reserved	-	-	-
40060(0059)	Reserved	-	-	-
40061(0060)	아날로그 지령방법	1 ~ 2	-	상세 파라미터 참조
40062(0061)	아날로그 입력최소	0 ~ 5000	mA	"
40063(0062)	FND Display Default	0 ~ 4	-	"
40064(0063)	모터 토크 체크 사용여부	0 ~ 1	-	"
40065(0064)	모터 토크 0_0 Rpm	0 ~ 30000	rpm	"
40066(0065)	모터 토크 0_0 전류 Base	100 ~ 10000	mA	"
40067(0066)	모터 토크 0_0 전류 Limit	100 ~ 10000	mA	"
40068(0067)	모터 토크 0_1 Rpm	0 ~ 30000	rpm	"
40069(0068)	모터 토크 0_1 전류 Base	100 ~ 10000	mA	"
40070(0069)	모터 토크 0_1 전류 Limit	100 ~ 10000	mA	"
40071(0070)	모터 토크 1_0 Rpm	0 ~ 30000	rpm	"
40072(0071)	모터 토크 1_0 전류 Base	100 ~ 10000	mA	"
40073(0072)	모터 토크 1_0 전류 Limit	100 ~ 10000	mA	"
40074(0073)	모터 토크 1_1 Rpm	0 ~ 30000	rpm	"
40075(0074)	모터 토크 1_1 전류 Base	100 ~ 10000	mA	"
40076(0075)	모터 토크 1_1 전류 Limit	100 ~ 10000	mA	"
40077(0076)	모터 토크 2_0 Rpm	0 ~ 30000	rpm	"
40078(0077)	모터 토크 2_0 전류 Base	100 ~ 10000	mA	"
40079(0078)	모터 토크 2_0 전류 Limit	100 ~ 10000	mA	"
40080(0079)	모터 토크 2_1 Rpm	0 ~ 30000	rpm	"
40081(0080)	모터 토크 2_1 전류 Base	100 ~ 10000	mA	"
40082(0081)	모터 토크 2_1 전류 Limit	100 ~ 10000	mA	"
40083(0082)	모터 토크 3_0 Rpm	0 ~ 30000	rpm	"

OZBV 통신 프로토콜

Slave Address	Name	Value	Unit	Explanation
40084(0083)	모터 토크 3_0 전류 Base	100 ~ 10000	mA	상세 파라미터 참조
40085(0084)	모터 토크 3_0 전류 Limit	100 ~ 10000	mA	"
40086(0085)	모터 토크 3_1 Rpm	0 ~ 30000	rpm	"
40087(0086)	모터 토크 3_1 전류 Base	100 ~ 10000	mA	"
40088(0087)	모터 토크 3_1 전류 Limit	100 ~ 10000	mA	"
40089(0088)	모터 토크 4_0 Rpm	0 ~ 30000	rpm	"
40090(0089)	모터 토크 4_0 전류 Base	100 ~ 10000	mA	"
40091(0090)	모터 토크 4_0 전류 Limit	100 ~ 10000	mA	"
40092(0091)	모터 토크 4_1 Rpm	0 ~ 30000	rpm	"
40093(0092)	모터 토크 4_1 전류 Base	100 ~ 10000	mA	"
40094(0093)	모터 토크 4_1 전류 Limit	100 ~ 10000	mA	"
40095(0094)	Speed Offset 0	0 ~ 100	rpm	"
40096(0095)	Speed Offset 1	0 ~ 100	rpm	"
40097(0096)	Speed Offset 2	0 ~ 100	rpm	"
40098(0097)	모터토크 전류제한 시간	50 ~ 10000	ms	"
40099(0098)	Reserved	-	-	-
40100(0099)	Reserved	-	-	-
40101(0100)	모터 동작 방향	0 ~ 1	-	상세 파라미터 참조
40102(0101)	모터 극수	2 ~ 16	-	-
40103(0102)	Reserved	-	-	-
40104(0103)	전류 필터	1 ~ 1024	-	-
40105(0104)	속도 필터	1 ~ 1024	-	-
40106(0105)	외부볼륨필터	1 ~ 1024	-	-
40107(0106)	전류센서용량	1000 ~ 10000	mA	-
40108(0107)	Reserved	-	-	-
40109(0108)	Reserved	-	-	-
40110(0109)	Reserved	-	-	-
4120(0119)	Reserved	-	-	-

**Note : 40064(0063) ~ 40098(0097) 파라미터는 모터 토크 제한에 관련된 파라미터입니다.
특별한 이유 없이 변경 하지 마십시오.**

3.4.2 Command 그룹

Slave Address	Name	Value	Unit	Explanation
40121(0120)	명령	-	-	표 3 참조
40122(0121)	지령 속도	-	rpm	Note 1
40123(0122)	Reserved	-	-	-
40124(0123)	Reserved	-	-	-
40125(0124)	Reserved	-	-	-
40126(0125)	Reserved	-	-	-
40127(0126)	Reserved	-	-	-
40128(0127)	Reserved	-	-	-
40129(0128)	Reserved	-	-	-
⋮				
40143(0142)	H/W 버전	-	-	
40144(0143)	S/W 버전	-	-	

Note 1 :

1. 지령속도 값 +- 부호로 모터 회전 방향이 바뀝니다. 예) -1000, 1000
2. 드라이버가 RUN 중 일 때는 지령속도 값 만 변경 하여도 속도가 변경 됩니다.
(펌웨어 버전 : 12.0.07 이후 버전 부터)

표 3) 명령 List

명령	설명	명령	설명
0	None	10	-
1	운전 시작	11	모터프리 정지
2	운전 정지	12	모터브레이크 정지
3	비상 정지	13	파라미터 저장
4	알람 리셋	-	-
5	-	-	-
6	-	-	-
7	-	-	-
8	-	-	-
9	-	-	-

4. CRC-16 Calculation

```
unsigned short CRC16(unsigned char *addr, int num)
{
    unsigned short CRC = 0xFFFF;
    int i;

    while (num--)
    {
        CRC ^= *addr++;
        for (i = 0; i < 8; i++)
        {
            if (CRC & 1)
            {
                CRC >>= 1;
                CRC ^= 0xA001;
            }
            else
            {
                CRC >>= 1;
            }
        }
    }

    return CRC;
}
```