

Modbus Parameters

Mode: RTU (MSB First)

Baud: 4800 9600 14400 19200 38400 bps (default: 19200bps)

Start Bits: 1

Data Bits: 8

Stop Bits: 1 / 2 (default 2)

Parity: None / Odd / Even (default None)

Modbus Address: 1~255 (default 1)

CDD 24 Registers Map

Support Function: 3 4 6 16

Starting Register (Dec.)	Data Description	R/W	Length	Units	Valid Response
3x0001	Co2 Measurement	R	2	Float inverse	0~2000
3x0003	Temp. Measurement	R	2	Float inverse	0.0~50.0
3x0005	Humi. Measurement	R	2	Float inverse	0.1~99.9
4x0001	Modbus Address	R/W	1	UINT16	1~255
4x0002	Modbus BPS	R/W	1	UINT16	1-4800;2-9600;3-14400; 4-19200;5-38400
4x0003	Modbus Stop Parity Bit	R/W	1	UINT16	1-None 1Stop;2-None 2Stop; 3-Odd 1Stop;4-Even 1Stop
4x0030	Co2 Adjust	R/W	1	UINT16	-200~200
4x0031	Temp. Adjust	R/W	2	Float inverse	-3.0~3.0
4x0033	Hum Adjust Setpoint	R/W	2	Float inverse	-5.0~5.0

Function 1: Read Coils Status (0x) (Max. Support 8 Coils)

Send Frame:

[ADDR]+[01]+[DATA START PT HO]+[DATA START PT LO]+
[DATA #OF PTS HO]+ [DATA #OF PTS LO]+[CRC16 lo]+[CRC16 hi]

Response Frame:

[ADDR]+[01]+[BYTE COUNT]+[DATA COIL STATUS]+ [CRC16 lo]
+ [CRC16 hi]

Function 2: Read Discrete Input Status (1x) (Max. Support 8 Discrete Input)

Send Frame:

[ADDR]+[02]+[DATA START PT HO]+[DATA START PT LO]
+[DATA #OF PTS HO]+ [DATA #OF PTS LO]+[CRC16 lo]+[CRC16 hi]

Response Frame:

[ADDR]+[02]+[BYTE COUNT]+[DATA DISCRETE INPUT]
+ [CRC16 lo]+ [CRC16 hi]

Function 3: Read Holding Register (4x)

Send Frame:

[ADDR]+[03]+[DATA START REG HO]+[DATA START REG LO]+
[DATA #OF REGS HO]+ [DATA #OF REGS LO]+[CRC16 lo]+[CRC16 hi]

Response Frame:

[ADDR]+[03]+[BYTE COUNT]+[DATA OUTPUT RES HO]
+ [DATA OUTPUT RES LO]+ [DATA OUTPUT REG HO]
+ [DATA OUTPUT REG LO]+ [CRC16 lo]+ [CRC16 hi]

Function 4: Read Input Register (3x)

Send Frame:

[ADDR]+[04]+[DATA START REG HO]+[DATA START REG LO]
+[DATA #OF REGS HO]+ [DATA #OF REGS LO]+[CRC16 lo]+[CRC16 hi]

Response Frame:

[ADDR]+[04]+[BYTE COUNT]+[DATA INPUT RES HO]+
[DATA INPUT RES LO]+ [DATA INPUT RES HO]+ [DATA INPUT RES LO]
+ [CRC16 lo]+ [CRC16 hi]

Function 6: Preset Single Register(0x4)

Send Frame:

[ADDR]+[06]+[DATA REG HO]+[DATA REG LO]+[DATA VALUE HO]
+ [DATA VALUE LO]+[CRC16 lo]+[CRC16 hi]

Response Frame:

[ADDR]+[06]+[DATA REG HO]+[DATA REG LO]+[DATA VALUE HO]
+ [DATA VALUE LO]+[CRC16 lo]+[CRC16 hi]

Function 16: Preset Multiple Register(0x4) (Only Send 4 Bytes At One Time)

Send Frame:

[ADDR]+[10]+[DATA REG HO]+[DATA REG LO]+
[QUANTITY HI]+[QUANTITY LO]
+[BYTE COUNT]+[DATA VALUE 4]+ [DATA VALUE 3]
+[DATA VALUE 2]+ [DATA VALUE 1] +[CRC16 lo]
+[CRC16 hi]

Response Frame:

[ADDR]+[10]+[DATA REG HO]+[DATA REG LO]+
[QUANTITY HI]+[QUALITY LO]
+[CRC16 lo]+[CRC16 hi]

Error Code: (Only Support 01 and 02)

01: ILLEGAL FUNCTION

02: LLEGAL DATA ADDRESS

CRC Hight Byte

```
__flash unsigned char auchCRCHI[256] = {  
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, 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, 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, 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, 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 Low Byte

```
__flash unsigned char auchCRCLo[256] = {  
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  
};
```