MPS_200_300S 系列 Modbus 協議通信協議

寄存器定義：

寄存器地址	名稱	03 功能碼	06 功能碼	10 功能碼	説明
0X0000	Remote Mode	Y	Y	Y	遠程控制狀態　　0:本地模式　1:遠程模式
0X0001	V_SET	Y	N	Y	電壓設定寄存器　　float 型
0X0003	A_SET	Y	N	Y	電流設定寄存器　　float 型
0X0005	V_MIN	Y	N	Y	最小電壓設定寄存器　　float 型
0X0007	V_MAX	Y	N	Y	最大電壓設定寄存器　　float 型
0X0009	A_MIN	Y	N	Y	最小電流設定寄存器　　float 型
0X000B	A_MAX	Y	N	Y	最大電流設定寄存器　　float 型
0X000D	OVP_SET	Y	N	Y	過壓參數設定寄存器　　float 型
0X000F	OCP_SET	Y	N	Y	過流參數設定寄存器　　float 型
0X0011	OVP_STATE	Y	Y	Y	過壓狀態設定寄存器　　0:關閉保護　1:打開保護
0X0012	OCP_STATE	Y	Y	Y	過流狀態設定寄存器　　0:關閉保護　1:打開保護
0X0013	OUTPUT	Y	Y	Y	輸出狀態設定寄存器　　0:關閉輸出　1:打開輸出
0X0014	STATE	Y	Y	Y	狀態讀取清除寄存器　　u16 型，參考狀態寄存器位定義表
0X0015	V_OUT	Y	N	N	輸出實際電壓寄存器　　float 型
0X0017	A_OUT	Y	N	N	輸出實際電流寄存器　　float 型
0X0019	CV/CC	Y	N	N	輸出工作狀態寄存器　　0:CV 狀態　1:CC 狀態

狀態寄存器(0X0014)位定義表：

Bit	名稱	屬性	説明
2	OTP_FLAG	W/R	1 爲過温保護標記，　寫 1 清除過温標記（温度回到安全值時）
1	OCP_FLAG	W/R	1 爲過流保護標記，　寫 1 清除過流標記
0	OVP_FLAG	W/R	1 爲過壓保護標記，　寫 1 清除過壓標記

注意：所有 float 型參數都占用兩個寄存器地址

功能碼表及間隔時間如下：

功能碼	對應功能	兩次操作間隔時間
0X03	連讀一個或多個寄存器命令	N * 5ms
0X06	單寫一個寄存器命令	10ms
0X10	連寫一個或多個寄存器命令	N * 5ms

注：N 爲寄存器個數

通信協議格式如下：

功能碼 0X03

PC 發送:8 Byte

地址	功能碼	起始地址高字節	起始地址低字節	寄存器個數高字節	寄存器個數低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X03						

電源返回： 5+N*2 Byte

地址	功能碼	數據長度字節	返回數據高字節	返回數據低字節	返回數據 N+1 高字節	返回數據 N+1 低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X03							

舉例一：讀輸出電壓值 (0X001D)

PC 發送：01 03 00 1D 00 02 54 0D (開始地址爲 0X001D，長度爲 2 個寄存器，CRC16 校驗結果爲 0X0D54)

電源返回：01 03 04 40 A0 00 00 EF D1 (0X40A00000 爲 5.000V，0XD1EF 爲 CRC16 校驗結果)

舉例二：讀輸出電壓電流值(0X001D~0X001F)

PC 發送：01 03 00 1D 00 04 D4 0F (開始地址爲 0X001D，長度爲 4 個寄存器，CRC16 校驗結果爲 0X0FD4)

電源返回：01 03 08 40 A0 00 00 40 00 00 00 24 2D (0X40A00000 表示 5.00V，0X40000000 表示 2.0000A，0X2D24 爲 CRC16 校驗結果)

功能碼 0X06

PC 發送： 8 Byte

地址	功能碼	寫入地址高字節	寫入地址低字節	寫入數據高字節	寫入數據低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X06						

電源返回： 8Byte

地址	功能碼	寫入地址高字節	寫入地址低字節	寫入數據高字節	寫入數據低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X06						

注意：所有 float 型參數寄存器都不支持 06 碼,06 碼只支持單個寄存器操作

舉例一：打開電源輸出(0X001B)

PC 發送：01 06 00 1B 00 01 38 0D (寫入地址爲 0X001B，寫入數據爲 0X0001，CRC16 校驗結果爲 0X0D38)

電源返回：01 06 00 1B 00 01 38 0D (寫入地址爲 0X001B，寫入數據爲 0X0001，CRC16 校驗結果爲 0X0A48)

功能碼 0X10

PC 發送： 9+N*2 Byte

地址	功能碼	起始地址高字節	起始地址低字節	寄存器個數高字節	寄存器個數低字節	數據長度字節	寫入數據高字節	寫入數據低字節	寫入數據 N+1 高字節	寫入數據 N+1 低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X10											

電源返回： 8 Byte

地址	功能碼	起始地址高字節	起始地址低字節	數據長度高字節	數據長度低字節	CRC16 校驗碼低字節	CRC16 校驗碼高字節
0X01	0X10						

舉例一：同時設定電壓電流值(0X0001~0X0003)

PC 發送：01 10 00 01 00 04 08 40 A0 00 00 40 00 00 00 FA 43 (開始地址爲 0X0001，長度爲 4 個寄存器，0X40A00000 爲 5.00V/40 00 00 00 爲 1.0000A，

CRC16 校驗結果爲 0X43FA)
電源返回：01 10 00 01 00 04 90 0A (開始地址爲 0X0001，4 個寄存器,CRC16 校驗結果爲 0X0A90)

CRCR16 計算代碼：

```
const unsigned char  CRCHTalbe[] =
{
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,
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,
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
}


const unsigned char  CRCLTalbe[] =
{
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
};
```

```
unsigned int crc16(unsigned char *DData,unsigned char len)
{
   unsigned char CRCHi = 0XFF;
   unsigned char CRCLo = 0XFF;
   unsigned int wIndex;
   unsigned int CRC_DData;
   while(len--)
    {
    wIndex = CRCLo ^ *DData++;
    CRCLo = CRCHi ^ CRCHTalbe[wIndex];
    CRCHi = CRCLTalbe[wIndex];
    }
CRC_DData = CRCHi;
CRC_DData<<= 8;
CRC_DData |=CRCLo;
   return CRC_DData;
}
```