# MPS-X00XH-1 系列 Modbus 协议通信协议

寄存器地址与对应功能码表如下：

| 寄存器地址 | 名称 | 功能 | 功能码 0X03 | 功能码 0X06 | 功能码 0X10 | 赋值范围 |
|---|---|---|---|---|---|---|
| 0X0000 | Remote Mode | 远程操作模式 | Y | Y | Y | 0:本地摸 1:远程模式 |
| 0X0001 | CH1 V_Set | CH1 电压参数设定 | Y | Y | Y | 0~Vmax 步进 1mV |
| 0X0002 | CH1 A_Set | CH1 电流参数设定 | Y | Y | Y | 0~Amax 步进 1mA |
| 0X0003 | CH1 OVP_Set | CH1 过压保护参数设定 | Y | Y | Y | 0~Vmax 步进 1mV |
| 0X0004 | CH1 OCP_Set | CH1 过流保护参数设定 | Y | Y | Y | 0~Amax 步进 1mA |
| 0X0005 | CH1 OVP_State | CH1 过压保护状态设定 | Y | Y | Y | 0:禁用 1: 启用 |
| 0X0006 | CH1 OCP_State | CH1 过流保护状态设定 | Y | Y | Y | 0:禁用 1: 启用 |
| 0X0007 | CH1 OUT_State | CH1 输出状态设定 | Y | Y | Y | 0:关闭 1: 输出 |
| 0X0008 | | | Y | Y | Y | |
| 0X0009 | | | Y | Y | Y | |
| 0X000A | | | Y | Y | Y | |
| 0X000B | | | Y | Y | Y | |
| 0X000C | | | Y | Y | Y | |
| 0X000D | | | Y | Y | Y | |
| 0X000E | | | Y | Y | Y | |
| 0X000F | CH1 V_Out | CH1 电压实际输出值 | Y | N | N | 0~Vmax 步进 1mV |
| 0X0010 | CH1 A_Out | CH1 电流实际输出值 | Y | N | N | 0~Amax 步进 1mA |
| 0X0011 | CH1 WorkMode | CH1 工作状态 | Y | N | N | 0: CV 模式 1:CC 模式 |
| 0X0012 | | | Y | N | N | |
| 0X0013 | | | Y | N | N | |
| 0X0014 | | | Y | N | N | |

注：所有操作基于远程操作模式下（寄存器 0X0000 值为 0X0001）生效！

*电源重新上电后基于本地操作模式!

功能码表及间隔时间如下：

| 功能码 | 对应功能 | 两次操作间隔时间 |
|---|---|---|
| 0X03 | 连读一个或多个寄存器命令 | N * 5ms |
| 0X06 | 单写一个寄存器命令 | 10ms |
| 0X10 | 连写一个或多个寄存器命令 | N * 5ms |

注：N 为寄存器个数

通信协议格式如下：

## 功能码 0X03

PC 发送:8 Byte

| 地址 | 功能码 | 起始地址高字节 | 起始地址低字节 | 寄存器个数高字节 | 寄存器个数低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|
| 0X01 | 0X03 | | | | | | |

电源返回： 5+N*2 Byte

| 地址 | 功能码 | 数据长度字节 | 返回数据高字节 | 返回数据低字节 | 返回数据 N+1 高字节 | 返回数据 N+1 低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|---|
| 0X01 | 0X03 | | | | | | | |

举例一：读单个寄存器 (0X0000)

PC 发送：01 03 00 00 00 01 84 0A （开始地址为 0X0000，长度为 1 个寄存器，CRC16 校验结果为 0X0A84）

电源返回：01 03 02 00 01 79 84 （0X0001 标示远程操作模式，0X8479 为 CRC16 校验结果）

举例二：读多个寄存器 (0X0001~0X0002)

PC 发送：01 03 00 01 00 02 95 CB （开始地址为 0X0001，长度为 2 个寄存器，CRC16 校验结果为 0XCB95）

电源返回：01 03 04 01 F4 03 E8 79 84　　(0X01F4 表示 0.500V，0X03E8 表示 1.000A，0X8479 为 CRC16 校验结果)

## 功能码 0X06

PC 发送：　8 Byte

| 地址 | 功能码 | 写入地址高字节 | 写入地址低字节 | 写入数据高字节 | 写入数据低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|
| 0X01 | 0X06 | | | | | | |

电源返回：　8Byte

| 地址 | 功能码 | 写入地址高字节 | 写入地址低字节 | 写入数据高字节 | 写入数据低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|
| 0X01 | 0X06 | | | | | | |

举例一：写寄存器状态(0X0000)

PC　发送：01 06 00 00 00 01 48 0A　　(写入地址为 0X0000，写入数据为 0X0001，CRC16 校验结果为 0X0A48)

电源返回：01 06 00 00 00 01 48 0A　　(写入地址为 0X0000，写入数据为 0X0001，CRC16 校验结果为 0X0A48)

## 功能码 0X10

PC 发送：　9+N*2 Byte

| 地址 | 功能码 | 起始地址高字节 | 起始地址低字节 | 寄存器个数高字节 | 寄存器个数低字节 | 数据长度字节 | 写入数据高字节 | 写入数据低字节 | 写入数据 N+1 高字节 | 写入数据 N+1 低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0X01 | 0X10 | | | | | | | | | | | |

电源返回：　8 Byte

| 地址 | 功能码 | 起始地址高字节 | 起始地址低字节 | 数据长度高字节 | 数据长度低字节 | CRC16 校验码低字节 | CRC16 校验码高字节 |
|---|---|---|---|---|---|---|---|
| 0X01 | 0X10 | | | | | | |

举例一：写多个寄存器状态(0X0001~0X0002)

PC  发送：01 10 00 01 00 02 04 01 F4 03 E8 72 D3  （开始地址为 0X0001，长度为 2 个寄存器,写入数据 0.500V/1.000A,CRC16 校验结果为 0XD372）
电源返回：01 10 00 01 00 04 90 0A  （开始地址为 0X0001，数据长度为 4 字节,CRC16 校验结果为 0X0A90）

## CRCR16 计算代码：

```
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;
}
```