

温湿度传感器 Modbus 协议

串口参数：
波特率 9600，
数据位 8，
校验位 无，
停止位 1

读取寄存器：03 (Read Register: 03)

The computer sends to the meter header: **device station number** **command** **start address** **number of reading required** **CRC check**

计算机向表头发送： 设备站号 命令 开始地址 需要读取数目 CRC 校验

Header return: **Device station number** **Command** **Data size** **Effective data** **CRC check**

表头返回： 设备站号 命令 数据大小 有效数据 CRC 校验

读寄存器 (Read register): Reg0-1

发出 (send) 0x01 0x03 0x00 0x00 0x00 0x02 0xC4 0x0B

接收正温 0x01 0x03 0x04 0x00 0xFA 0x03 0xC8 0xDB 0x64 (Receive positive temperature)

接收负温 0x01 0x03 0x04 0xFF 0x69 0x03 0xB8 0x1A 0xB9 (Receive negative temperature)

0x01 设备站号 (Equipment station number)

0x03 命令 (command)

0x04 数据大小 (Data size)

0x00 0xFA 0x03 0xC8 正有效数据

温度= $0x00FA = 250 = 25.0$ 度

湿度= $0x03C8 = 968 = 96.8$ %

0xDB 0x64 CRC 校验

0xFF 0x69 0x03 0xB8 负有效数据

温度= $0xFF69 = (0x10000 - 0xFF69) = 151 = -1.51$ 度

湿度= $0x03B8 = 968 = 95.2$ %

0x1A 0xB9 CRC 校验

Change the address: 06 (When changing the address, the device station number must be 0)

改地址: **06** (改地址时, 设备站号要为 0)

The computer sends to the header: device station number command start address modified data CRC check

计算机向表头发送: 设备站号 命令 开始地址 修改的数据 CRC 校验

表头返回: 设备站号 命令 开始地址 修改的数据 CRC 校验

改地址:

发出 0x00 0x06 0x00 0x00 0x00 0x02 0x09 0xDA ->将总线上的设备地址改为 0x00 0x02

接收 0x00 0x06 0x00 0x00 0x00 0x02 0x09 0xDA ->返回与接收数据一至

Note: When changing the address, there can only be one slave device on the bus, otherwise all slave devices will be changed

注: 改地址时, 总线上只能有一个从机设备, 否则所有的从机设备都会改掉

修正温湿度方法

温度加格式: **AT+CT+x.x**

AT+CT+xx.x

加回车发送

温度减格式: **AT+CT-x.x**

AT+CT-xx.x

加回车发送

例: **AT+CT+2.1** 在当前温度上加 **2.1** 度

AT+CT-1.5 在当前温度上减 **1.5** 度

湿度加格式: **AT+CH+x.x**

AT+CH+xx.x

加回车发送

湿度减格式: **AT+CH-x.x**

AT+CH-xx.x

加回车发送

例: **AT+CH+5.2** 在当前湿度上加 **5.2%**

AT+CH-1.0 在当前湿度上减 **1.0%**

CRC16 校验产生函数

```
/*
*****
* Function Name   :   crc16
* Input          :   数据缓冲区指针: puchMsg ,   数据长度: usDataLen
* Return         :   16 位 CRC 校验码
* Description    :   产生 16 位 CRC 校验码
*****
INT16U crc16(INT8U *puchMsg, INT8U usDataLen)
{
    INT8U  uchCRCHi=0xFF; /* 高 CRC 字节初始化 */
    INT8U  uchCRCLo=0xFF; /* 低 CRC 字节初始化 */
    INT16U uIndex;        /* CRC 循环中的索引 */

    while(usDataLen--)    /* 传输消息缓冲区 */
    { uIndex  =uchCRCHi^*puchMsg++; /* 计算 CRC */
      uchCRCHi=uchCRCLo^uchCRCHi[uIndex];
      uchCRCLo=uchCRCLo[uIndex];
    }
```

```

    }
    return (uchCRCHi<<8|uchCRCLo);
}

```

/* CRC 高位字节值表 */

```

const INT8U code auchCRCHi[] = {
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  
};
```

/* CRC 低位字节值表*/

```
const INT8U code auchCRCLo[] = {  
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
} ;