

中盛数码管显示屏

使用手册

V4.2

1.产品简介

1.1 产品功能

- 多种通讯方式可选:RS485、RS232、TTL(需订货时指定其中一种),支持 MODBUS RTU 协议。
- 可显示整数、负号、小数点、 ABCDEFPI 等各种 ASCII 字符,支持字符全面的一款数码管显示器。
- 支持电子标签功能,上电初始显示内容可设置。广泛用于仓储位号标签。比如显示“A001”、“B002”、“C003”等。
- 支持笔画亮度调节,具有 8 档亮度。
- 支持修改波特率(4800bps-115200bps),覆盖常用的几种波特率。
- 支持修改停止位、校验位。
- 可选择上电初始显示内容,显示 0、显示 485 地址、显示指定内容等
- 一条 485 总线可接 32 个数码管(无中继),加中继可接 254 个。
- 提供配套的 PC 机软件,方便测试、修改参数。
- 具有闪烁显示功能,可设置其中某一位、某几位、全部闪烁。

1.2产品特点

- 安装方便。设备面板上只需要开方孔即可嵌入,无需安装螺丝。
- 内置开关电源电路,输入电压范围宽,转换效率高,可以长时间连续工作。
- 电源输入端具有反接保护,过流保护。
- 工业级 RS485 芯片,总线接口具有防雷防浪涌、防静电、过压、过流保护
- 芯片全部为原装全新正品,工业级温度范围。

1.3 产品参数

| | |
|------|---|
| 供电电压 | DC 5~37V |
| 接口 | 端子型号:H2.54-4P 引脚定义: <ul style="list-style-type: none">➤ +:电源正极➤ -:电源负极➤ A/T:RS485 总线 A/ TTL_TXD➤ B/R:RS485 总线 B/TTL_RXD |
| 串口参数 | <ul style="list-style-type: none">➤ 串口参数波特率:4800、9600（出厂默认）、19200、38400、57600、115200bps➤ 起始位:1 位➤ 停止位:1 位(出厂默认)、2 位、1.5 位➤ 数据位:8➤ 校验位:无校验(出厂默认)、奇校验、偶校验 |
| 通讯协议 | Modbus RTU |

2. 通讯协议

2.1 寄存器类型

本模块使用 Modbus-RTU 中的保持寄存器。。寄存器中的数值均为16 位无符号数值。

2.2 保持寄存器功能定义

| 地址 | 类型 | 复位值 | 功能描述 |
|----|----|------|---|
| 0 | 整型 | 0x30 | 第 1 位数码管显内容，ASCII 码值。 |
| 1 | 整型 | 0x30 | 第 2 位数码管显内容，ASCII 码值 |
| 2 | 整型 | 0x30 | 第 3 位数码管显内容，ASCII 码值 |
| 3 | 整型 | 0x30 | 第 4 位数码管显内容，ASCII 码值 |
| 4 | 整型 | 0x30 | 第 5 位数码管显内容，ASCII 码值 |
| 5 | 整型 | 0x30 | 第 6 位数码管显内容，ASCII 码值 |
| 6 | 整型 | 0 | <p>数据格式。</p> <p>高字节的高 4 位：</p> <ul style="list-style-type: none">➤ 0：正数➤ 1：负数 <p>高字节的低 4 位：小数点个数，范围 0~5</p> <p>低字节：</p> <p>与寄存器 7 一起表示要显示的数据（显示 65535 以上的数据，需使用 3 个字节表示），此字节表示数据的最高 8 位。（0xFFFFFF）</p> <p>注：此寄存器需与寄存器 7 同时使用，显示数据时使用写多个保持寄存器往这 2 个寄存器中写入数据。</p> |
| 7 | 整型 | 0 | <p>显示数据。</p> <p>（1）与寄存器 6 一起表示要显示的数据（数据使用 3 个字节表示），此寄存器的高字节表示数据的中间 8 位，低字节表示数据的最低 8 位。高字节在前，低字节在后。（0xFFFFFF）</p> <p>注：此寄存器需与寄存器 6 同时使用，显示数据时使用写多个保持</p> |

| | | | |
|----|----|---|---|
| | | | <p>寄存器往这 2 个寄存器中写入数据。</p> <p>(2) 独立使用时，往寄存器里写入十进制整数，数码管显示相应的数据。可以与寄存器 16（小数点个数）和寄存器 17（正负数设置）一起使用，显示浮点数和负数</p> |
| 8 | 整型 | 0 | <p>闪烁控制寄存器。每位代表一位数码管，最低位代表第一位数码管，依此类推。</p> <ul style="list-style-type: none"> ➤ 0: 不闪烁 ➤ 1: 闪烁 <p>注：此参数掉电不保存</p> |
| 9 | 整型 | 0 | <p>显示内容保存。</p> <ul style="list-style-type: none"> ➤ 0: 不保存 ➤ 1: 保存所有数码管显示内容 <p>注：此参数掉电保存</p> |
| 10 | 整型 | 1 | <p>模块的 485 总线地址，范围 1~254，0 是广播地址</p> <p>注：此参数掉电保存</p> |
| 11 | 整型 | 1 | <p>波特率设置。</p> <ul style="list-style-type: none"> ➤ 0:4800 ➤ 1: 9600 ➤ 2:14400 ➤ 3:19200 ➤ 4:38400 ➤ 5: 56000 ➤ 6: 57600 ➤ 7:115200 ➤ 8:128000 ➤ 9:256000 <p>注：此参数掉电保存</p> |
| 12 | 整型 | 0 | <p>停止位设置。</p> <ul style="list-style-type: none"> ➤ 0: 1 个停止位 ➤ 1: 1.5 个停止位 ➤ 2: 2 个停止位 <p>注：此参数掉电保存</p> |

| | | | |
|----|----|---|---|
| 13 | 整型 | 0 | <p>校验位设置。</p> <ul style="list-style-type: none"> ➤ 0: 无校验 ➤ 1: 奇校验 ➤ 2: 偶校验 <p>注：此参数掉电保存</p> |
| 14 | 整型 | 3 | <p>亮度调节。范围 0~7，0 表示最暗，7 表示最亮。</p> <p>注：此参数掉电保存</p> |
| 15 | 整型 | 0 | <p>数码管上电初始显示模式设置。</p> <ul style="list-style-type: none"> ➤ 0: 显示 “0” ➤ 1: 显示模块的 485 总线地址 ➤ 2: 显示保存的数据 <p>注：此参数掉电保存</p> |
| 16 | 整型 | 0 | <p>小数点个数</p> <p>注：此参数掉电保存</p> |
| 17 | 整型 | 0 | <p>正负数设置</p> <ul style="list-style-type: none"> ➤ 0: 正数 ➤ 1: 负数，显示负号 <p>注：此参数掉电保存</p> |

3 通信示例

3.1 读保持寄存器（功能码 0x03）

读从机保持寄存器，不支持设备广播地址。

控制器（上位机）发送：

报文中包含了要读的寄存器起始地址及寄存器的数量。例如要读取设备地址为 0x01 的保持寄存器（地址 0x0001~0x0003），则发送报文如下表：

| 字节名称 | Hex(16 进制) |
|-------------|------------|
| 设备地址 | 01 |
| 功能码 | 03 |
| 寄存器起始地址高字节 | 00 |
| 寄存器起始地址低字节 | 01 |
| 寄存器数量高字节 | 00 |
| 寄存器数量低字节 | 03 |
| CRC16 校验高字节 | -- |
| CRC16 校验低字节 | -- |

代码示例：

```
unsigned char TxCommand[8];  
TxCommand[0] = 0x01;  
TxCommand[1] = 0x03;  
TxCommand[2] = 0x00;  
TxCommand[3] = 0x01;  
TxCommand[4] = 0x00;  
TxCommand[5] = 0x03;  
TxCommand[6] = CRC 校验低字节;  
TxCommand[7] = CRC 校验高节;
```

向串口发送 TxCommand 数组可读取对应寄存器的数据。

模块返回：

每个寄存器分别对应 2 个字节，第一个字节为高 8 位数据，第二个字节为低 8 位数据。

如下表：

| 字节名称 | Hex(16 进制) |
|--------------|------------|
| 设备地址 | 01 |
| 功能码 | 03 |
| 寄存器数据字节数 | 06 |
| 寄存器 1 数据高字节 | -- |
| 寄存器 1 数据低字节 | -- |
| 寄存器 2 数据高字节 | -- |
| 寄存器 2 数据低字节 | -- |
| 寄存器 3 数据高字节 | -- |
| 寄存器 3 数据低字节) | -- |
| CRC16 校验高字节 | -- |
| CRC16 校验低字节 | -- |

3.2 写单个保持寄存器（功能码 0x06）

写单个保持寄存器。广播时，该功能把值设置到所有从机的寄存器中。

控制器（上位机）发送：

报文中规定了要写的保持寄存器地址。例如要写设备地址为 0x01 中地址为 0x0000 的保

持寄存器的值为 0x0003，则发送报文如下表所示：

| 字节名称 | Hex(16 进制) |
|-------------|------------|
| 设备地址 | 01 |
| 功能码 | 06 |
| 寄存器地址高字节 | 00 |
| 寄存器地址低字节 | 00 |
| 数据高字节 | 00 |
| 数据低字节 | 03 |
| CRC16 校验高字节 | --- |
| CRC16 校验低字节 | --- |

代码示例:

```
unsigned char TxCommand[8];  
TxCommand[0] = 0x11;  
TxCommand[1] = 0x06;  
TxCommand[2] = 0x00;  
TxCommand[3] = 0x00;  
TxCommand[4] = 0x00;  
TxCommand[5] = 0x03;  
TxCommand[6] = CRC16 校验低字节;  
TxCommand[7] = CRC16 校验高字节;  
向串口发送 TxCommand 数组可把数据 0x0003 写入寄存器中。
```

模块返回:

| 字节名称 | Hex(16 进制) |
|-------------|------------|
| 设备地址 | 01 |
| 功能码 | 06 |
| 寄存器地址高字节 | 00 |
| 寄存器地址低字节 | 00 |
| 数据位高字节 | 00 |
| 数据位低字节 | 03 |
| CRC16 校验高字节 | --- |
| CRC16 校验低字节 | --- |

3.3 写多个保持寄存器（功能码 0x10）

把数据按顺序写到多个保持寄存器中，广播时该功能代码可把数据预置到全部从机中的相同类型的寄存器中。

控制器（上位机）发送：

报文中规定了要写的保持寄存器首地址和寄存器数量。例如要写设备地址为 0x01 中的保持寄存器（地址为 0x0000~0x0001，分别写入数字 0x0001 和 00002），则发送报文如下表所示：

| 字段名称 | Hex(16 进制) |
|-------------|------------|
| 设备地址 | 01 |
| 功能码 | 10 |
| 寄存器起始地址高字节 | 00 |
| 寄存器起始地址低字节 | 00 |
| 寄存器数量高字节 | 00 |
| 寄存器数量低字节 | 02 |
| 寄存器数据字节个数 | 04 |
| 数据位高字节 | 00 |
| 数据位低字节 | 01 |
| 数据位高字节 | 00 |
| 数据位低字节 | 02 |
| CRC16 校验高字节 | --- |
| CRC16 校验低字节 | --- |

代码示例：

```
unsigned char TxCommand[13];
TxCommand[0] = 0x01;
TxCommand[1] = 0x10;
TxCommand[2] = 0x00;
TxCommand[3] = 0x00;
TxCommand[4] = 0x00;
TxCommand[5] = 0x02;
TxCommand[6] = 0x04;
TxCommand[7] = 0x00;
TxCommand[8] = 0x01;
TxCommand[9] = 0x00;
TxCommand[10] = 0x02;
```

TxCommand[11] = CRC 校验位低字节;

TxCommand[12] = CRC 校验位高字节;

向串口发送 TxCommand 数组可把数据 0x0001 和 0x0001 写入到指定的寄存器中。

模块返回:

正常响应返回从机地址、功能代码、寄存器起始地址和要寄存器数量。

| 字节名称 | Hex(16 进制) |
|-------------|------------|
| 设备地址 | 01 |
| 功能码 | 10 |
| 寄存器起始地址高字节 | 00 |
| 寄存器起始地址低字节 | 00 |
| 寄存器数量高字节 | 00 |
| 寄存器数量低字节 | 02 |
| CRC16 校验高字节 | --- |
| CRC16 校验低字节 | --- |

3.4 CRC 校验计算程序

```
static const unsigned char 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,
    0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    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,
    0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,
    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
};

static const unsigned char 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
};

/**
 * @brief Get CRC16
 * @par Full description:
 * None
 * @retval void None
 * @par Required preconditions:
 * None
 * @par Called functions:
 * None
 */
void Modbus_GetCRC16(unsigned char *RxBuffer,unsigned short
usLength,unsigned char *destCRC16Hi,unsigned char *destCRC16Lo)
{
    unsigned char uchCRCHi=0xFF;
    unsigned char uchCRCLo=0xFF;
    unsigned short uIndex ;
    while(usLength--)
    {
        uIndex=(unsigned char)(uchCRCHi^*RxBuffer++);
        uchCRCHi=(unsigned char)(uchCRCLo^auchCRCHi[uIndex]);
        uchCRCLo=(unsigned char)(auchCRCLo[uIndex]);
    }
    *destCRC16Hi=uchCRCLo;
    *destCRC16Lo=uchCRCHi;
}

```

4 ASCII 码及字形表

ASCII 码中 00H~1FH 之间的数据为控制字符,属于不可显示的字符, 20H 表示空格字符。2H-7EH 之间的数为可见的 ASCII 字符。其中能正确被数码管屏所显示的字符如下表。

| | | | | | | | | | |
|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| | | | | | | | | | |
| (20H) | !(21H) | " (22H) | #(23H) | '(27H) | ,(2cH) | -(2dH) | .(2eH) | /(2fH) | =(3dH) |
| | | | | | | | | | |
| ?(3fH) | [(5bH) | \(5cH) |](5dH) | _(5fH) | `(60H) | {(7bH) | (7cH) | }(7dH) | ~(7eH) |
| | | | | | | | | | |
| 0(30H) | 1(31H) | 2 (32H) | 3(33H) | 4(34H) | 5(35H) | 6(36H) | 7(37H) | 8(38H) | 9(39H) |
| | | | | | | | | | |
| A(41H) | B(42H) | C(43H) | D(44H) | E(45H) | F(46H) | G(47H) | H(48H) | I(49H) | J(4aH) |
| | | | | | | | | | |
| K(4bH) | L(4cH) | M(4dH) | N(4eH) | O(4fH) | P(50H) | Q(51H) | R(52H) | S(53H) | T(54H) |
| | | | | | | | | | |
| U(55H) | W(57H) | Y(59H) | Z(5aH) | a(61H) | b(62H) | c(63H) | d(64H) | e(65H) | f(66H) |
| | | | | | | | | | |
| g(67H) | h(68H) | i(69H) | j(6aH) | k(6bH) | l(6cH) | m(6d) | n(6eH) | o(6fH) | p(70H) |
| | | | | | | | | | |
| q(71H) | r(72H) | s(73H) | t(74H) | u(75H) | w(78H) | y(79H) | z(7aH) | | |

中盛科技（东莞）有限公司 版权所有
技术支持热线：138 2574 1827