1..符合 MODBUS 标准（16 进制方式）。

主机查询，变送器应答的主从方式

查询温湿度数据

地址	03	00	00	00	02	CRCH	CRCL

例：对地址位为 01 的变送器读温湿度操作为：010300000002C40B

应答

地址	03	04	温度 H	温度 L	湿度 H	湿度 L	CRC H	CRC L

注：CRCH为CRC校验高字节，CRCL为CRC校验低字节。

2. 数据 H（高位字节）和数据 L（低位字节）为各自对应的当前温湿度值：

- 上传数据需除十，如湿度上传 16 进制 0311，对应十进制 00785，表示 78.5%。
- 零下温度换算,如温度上传 16 进制 FF8C，对十制为(0XFFFF-0XFF8C=0X73)115,表示-11.5℃。

3.节格式 8 位数据位，无校验，1 位停止位，波特率 1200，2400，4800,9600 可以设定。

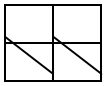
例：如对地址位 01（对应变送器 7 位拨码开关为 0000001）的变送器直接查询，在串口调试程序中进行如上通讯设置后输入：010300000002C40B 即可

4.和波特率的设定方法（打开产品的外壳，有两组设定参数的拨码开关）

4.1 波特率：出厂缺省：9600 bit/s

OFF(0)

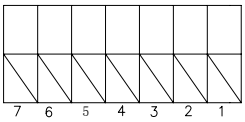
2 1



ON(1)

Bit2	Bit1	波特率
1	1	9600
0	1	4800
1	0	2400
0	0	1200

4.2 地址(缺省：01)



OFF(0)

ON(1)

BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7	地址
0	0	0	0	0	0	1	01
0	0	0	0	0	1	0	02
0	0	0	0	0	1	1	03
0	0	0	0	1	0	0	04

……							
……							
……							
1	1	1	1	1	0	0	124
1	1	1	1	1	0	1	125
1	1	1	1	1	1	0	126
1	1	1	1	1	1	1	127

5. CRC 校验的算法

下面是 CRC 算法的 C 语言的程序,用户编程可以直接应用或进行相应的移植.

```
//本程序为网络型温湿度校验方式的示范程序，适当更改就可使用；
#define CRC_CONSTANT 0xa001
unsigned int crc_result=0;
void crc_check (unsigned char crc_data) // crc_data is the nummber of check
    {
     bit xor_flag=1;
     unsigned char m;
     unsigned int crc_num;
     crc_result^=crc_data;
     crc_num=crc_result;
     crc_num&=0x0001;
     for (m=0;m<8;m++)
          {
           if (crc_num) xor_flag=1;
               else xor_flag=0;
           crc_result>>=1;
           if (xor_flag) crc_result^=CRC_CONSTANT;
           crc_num=crc_result;
           crc_num&=0x0001;
          }
    }
//应用示例
main()
    {
    unsigned char i,j,k;
    unsigned char r_buffer[10];////定义发送数据数组
    unsigned int int_crc;
        r_buffer[0]=0x01;//////请求应答的变送器地址
        r_buffer[1]=0x03;///功能码，固定
        r_buffer[2]=0x00;///固定
        r_buffer[3]=0x00;//固定
```

```
        r_buffer[4]=0x00;//固定
        r_buffer[5]=0x02;//固定
        r_buffer[6]=0xc4;////crc  高
        r_buffer[7]=0x0b;////crc  低
    crc_result=0xffff;
            for (i=0;i<6;i++)
                {
                j=r_buffer[i];
                crc_check (j);
                }
  int_crc=crc_result;     //     01    03    00    00    00    02 CRC 的高位 C4，低位 OB    ;和 int_crc 高低位相反
  int_crc=0;
}
```