

## 协议特性

- 1、TCP 连接
- 2、使用CRC16数据帧错误检查，多项式码0xA 0 01.
- 3、通讯数据为16进制数据。
- 4、若服务器收到错误命令，则不会响应终端的命令。
- 5、除CRC16字段是低字节在前，高字节在后外，其他字段的多字节以高字节在前，低字节在后处理。

### 二、协议说明

#### 1)通用帧格式

序号	内容	字节数	说明	备注
1	设备ID	4	设备ID号	唯一，平台进行ID注册
2	命令域	1	命令码	
3	数据长度	1	数据域的长度	发送或应答有效数据域长度
4	数据域	N	数据	
5	校验域	2	CRC16校验码（从设备ID到数据域，包括数据域）	低前高后

#### 2)协议命令码

序号	命令码	含义	备注
1	0XA2	发送实时数据	上传要素数量和数据长度可变。服务器依据数据长度读取实时数据，并不是固定的0x40，只是说最大为0x40个字节。一般为0x40

#### 3)通讯协议详细说明

温度数据都以有符号16进制表示，如数据为0x 00 0x FE，表示为十进制数即 254 (0x 00FE)，温度含 1 位小数，则实际温度值为 25.4℃。即数据字节的高位在前，低位在后，此顺序不能错，否则数据值将错误。

土壤湿度数据都以有符号16进制表示，如数据为0x 01 0 Xc 3，表示十进制数 45.1 (0x 01c 3)，土壤湿度含 1 位小数点，则实际土壤湿度为 45.1%。十六进制数据高位在前，低位在后，此顺序不能错，否则数据值将错误。

### 终端向服务器发送数据

终端发送：

```
10 06 1C 01 A2 40 2E EE 01 46 00 FE 01 50 00 F4 01 47 00 F6 01 48 00 F5 7F FF 7F FF
7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF
```

FF 7F FF 7F FF 12 09 13 0D 02 28 7F FF 7F FF 42 83

说明：40 代表终端发送了 64 个字节有效数据给服务器（最大字节数的发包），0x00FE (=254) 表示水温为 25.4℃；其他通道数据为无效数据 0x7FFF。终端会发送十六个标准通道数据（加 16 个扩展通道，共 32 个通道，每个通道占 2 个字节）到服务器。

16 个标准通道分别对应（每个通道对应的哪种要素最好可配置，当然如果只用于导管土湿可按如下表格对应。）：

CH0	电池电量（单位 V，3 位小数，若单位是 mv 没有小数点） 12.014 V	CH 8	土温 4 24.5
CH 1	土湿 1 32.6	CH 9	气温
CH 2	土温 1 25.4°	CH 10	保留
CH 3	土湿 2 33.6	CH 1 1	保留
CH 4	土温 2 24.4	CH 12	保留
CH 5	土湿 3 32.7	CH 13	保留
CH 6	土温 3 24.6	CH 14	保留
CH 7	土湿 4 32.8	CH 15	保留

16 个扩展通道对应：

CH 16	保留	CH 24	保留
CH 1 7	保留	CH 25	保留
CH 18	保留	CH 26	保留
CH 19	保留	CH 27	当前时间 年月 12 18年 09 9月
CH 20	保留	CH 28	当前时间 日时 13 19日 0D 13点
CH 21	保留	CH 29	当前时间 分秒 02 2分 28 40秒
CH 22	保留	CH 30	保留
CH 23	保留	CH 31	保留

服务器返回：

10 06 1C 01 A2 08 10 06 1C 00 00 00 02 58 C3 46

说明：若数据接收成功，则服务器返回实时时间。（10 06 1C 00 00 00 指年，月，日，时，分，秒）。02 58 为上传间隔时间（600秒，高字节在前，低字节在后。平台可修改，在收到上传的实时数据后下发给终端。最低 300 秒）。C3 46 CRC 校验码

### 三 计算CRC码的步骤:

- 1、预置16位寄存器为十六进制FFFF（即全为1）。称此寄存器为CRC寄存器；
- 2、把第一个8位数据与16位CRC寄存器的低8位相异或，把结果放于CRC寄存器；
- 3、把寄存器的内容右移一位（朝低位），用0填补最高位，并检查右移后的移出位；
- 4、如果最低位为0：重复第3步（再次右移一位）  
如果最低位为1：CRC寄存器与多项式A001(1010 0000 0000 0001)进行异或；
- 5、重复步骤3和4，直到右移8次，这样整个8位数据全部进行了处理；
- 6、重复步骤2到步骤5，进行下一字节8位数据的处理；
- 7、将该通讯帧所有字节(除CRC字段)按上述步骤计算完成后，得到的16位CRC寄存器高低字节进行交换；
- 8、将CRC结果放入信息帧时。

```

//*****
/**名      称: CRC16
/**说      明: CRC效验函数
/**形      参: *p效验帧的指针   帧长 datalen
/**返 回  值: 效验字
//*****
unsigned int CRC16(unsigned char * p, uint16 datalen )
{
    unsigned char CRC16Lo,CRC16Hi,CL,CH,SaveHi,SaveLo;
    int i,Flag;
        CRC16Lo = 0xFF;      CRC16Hi = 0xFF;
        CL = 0x01;          CH = 0xA0;
    for(i=0;i<datalen;i++)
    {
        CRC16Lo ^=*(p+i); //每一个数据与CRC寄存器进行异或
    for(Flag=0;Flag<8;Flag++)
        {
            SaveHi = CRC16Hi; SaveLo = CRC16Lo;
            CRC16Hi >>= 1 ;CRC16Lo >>= 1 ; //高位右移一位，低位右移一位
            if ((SaveHi & 0x01) == 0x01) //如果高位字节最后一位为1
                CRC16Lo |=0x80 ; //则低位字节右移后前面补1否则自动补0
            if ((SaveLo & 0x01) == 0x01) //如果LSB为1，则与多项式码进行异或
            { CRC16Hi ^= CH; CRC16Lo ^= CL; }
        }
    }
    return (CRC16Hi<<8)|CRC16Lo; (返回后再进行高低位转换)
}

```