

# DTS 温度数据网络传输协议

山东科华电力技术有限公司\_2018

## 1 端口设置

通过设置软件中输出设置中的网络传输端口，可以设置数据传输的端口，数据传输端口为设置值及后面的3个端口，例如设置2404，则2404为默认传输端口，2405、2406、2407为备用端口，也可以用于数据传输。

## 2 通信协议

21 、连接方式：采用点对点方式，DTS 主机作为数据服务器端，客户端连接成功后，主机对通道光缆进行循环监测，主机将采集数据实时发送，大约 6s 进行一次数据传输，一次传输当前通道的数据（一次数据传输可能分多个数据包发送）。

22 、包协议

表 1 中定义了具体的包协议，

表 1 包协议

字节数	描述	内容
1	包头	0xCC
2、3	字节数	见3.2.1
4	标示字	见3.2.2
5	类型字	见3.2.3
6到M+5	参数(M)	见3.2.4
M+5	CRC8	见3.2.5
M+6	包尾	0xED

### 2.2.1 字节数

2 个字节，高位在前、低位在后，字节数是从标示字开始到 CRC 结束的字节数量，包括命令字及 CRC 校验字，假如参数字节数为 M，则字节数 N 就等于 M+3；

### 2.2.2 标示字（传输数据所在的光纤通道）

1 个字节，表示“主方”发送的数据所对应的通道编号，与主机的通道编号相同，4 通道主机对应的标示字为：0x01、0x02、0x03、0x04。

### 2.2.3 类型字（传输数据的类型）

1 个字节，表示“主方”发送的数据类型，0x01：温度数据、0x02：故障报警、0x03 温度报警；

### 2.2.4 参数（传输的主要数据内容）

参数部分是主要的传输数据，根据类型字的不同，参数的数据结构也不同；  
类型字为 0x01 温度数据时：参数部分的内容为实时采集温度数据，传输数据从测温电缆的开始位置记为开始 0 点，每隔一米一个温度值，每个温度数据 2 个字节传输，高位在前，低位在后，且传输值=温度值×10+1000（温度值为 1 位小数），如 20.5℃的传输数据：20.5×10+1000=1205=0x04B5，传输值：04 B5；  
例如：通道 1 中共有 10 长米光纤每一米的值为：0 米处 20.7℃、1 米处 20.5℃、

2 米处 20.6℃、3 米处 20.3℃、4 米处 20.3℃、5 米处 20.8℃、6 米处 20℃、7 米处 20.8℃、8 米处 20.8℃、9 米处 20.7℃、10 米处 20℃，则传输数据为：CC 00 19 01 01 04 B7 04 B5 04 B6 04 B3 04 B3 04 B8 04 B0 04 B8 04 B8 04 B7 04 B0 BD ED

类型字为 0x02 故障报警时：参数部分的内容为当前出现的故障报警，

参数数据格式：故障类型 故障位置高位 故障位置低位 故障状态 报警时间（6 个字节）

故障类型：0x01 通道无数据 0x02 光纤断裂；

故障位置：当类型为 0x02 时，即为实际的光纤断裂位置（高位在前，低位在后）其他情况为 0x00 0x00；

故障状态：0x00 故障消失 0x01 故障报警；

报警时间：6 个字节，字节顺序为：年 月 日 时 分 秒；

年（年值=当前时间年-2010）、月份值（1-12）、日（1-31）、时（24 小时制 0-23）、分钟（0-59）、秒（0-59）。

类型字为 0x03 温度报警时：参数部分的内容为当前出现的温度报警，

参数数据格式：温度报警类型 报警位置高位 报警位置低位 报警温度值高位 报警温度值低位 报警状态 报警时间（6 个字节）

报警类型：0x01 高温橙色报警 0x02 高温红色报警 0x03 温升报警；

报警位置：为实际的温度报警位置（高位在前，低位在后）其他情况为 0x00 0x00；

报警温度值：传输值=温度值×10+1000（温度值为 1 位小数），如 20.5℃的传输数据：20.5×10+1000=1205=0x04B5，传输值：04 B5；

报警状态：0x00 报警消失 0x01 报警中；

报警时间：6 个字节，字节顺序为：年 月 日 时 分 秒；

年（年值=当前时间年-2010）、月份值（1-12）、日（1-31）、时（24 小时制 0-23）、分钟（0-59）、秒（0-59）。

## 2.2.5 CRC

1 个字节，CRC 循环冗余校验采用的生成多项式为  $g(x)=x^8+x^5+x^4+1$ ，校验内容为 CRC 校验位前所有字节（包括包头）。

附：CRC8 查表法（ $x^8+x^5+x^4+1$ ）

```
unsigned char crc_array[256] = {  
  
    0x00, 0x5e, 0xbc, 0xe2, 0x61, 0x3f, 0xdd, 0x83,  
    0xc2, 0x9c, 0x7e, 0x20, 0xa3, 0xfd, 0x1f, 0x41,  
    0x9d, 0xc3, 0x21, 0x7f, 0xfc, 0xa2, 0x40, 0x1e,  
    0x5f, 0x01, 0xe3, 0xbd, 0x3e, 0x60, 0x82, 0xdc,  
    0x23, 0x7d, 0x9f, 0xc1, 0x42, 0x1c, 0xfe, 0xa0,  
    0xe1, 0xbf, 0x5d, 0x03, 0x80, 0xde, 0x3c, 0x62,  
    0xbe, 0xe0, 0x02, 0x5c, 0xdf, 0x81, 0x63, 0x3d,
```

```

0x7c, 0x22, 0xc0, 0x9e, 0x1d, 0x43, 0xa1, 0xff,
0x46, 0x18, 0xfa, 0xa4, 0x27, 0x79, 0x9b, 0xc5,
0x84, 0xda, 0x38, 0x66, 0xe5, 0xbb, 0x59, 0x07,
0xdb, 0x85, 0x67, 0x39, 0xba, 0xe4, 0x06, 0x58,
0x19, 0x47, 0xa5, 0xfb, 0x78, 0x26, 0xc4, 0x9a,
0x65, 0x3b, 0xd9, 0x87, 0x04, 0x5a, 0xb8, 0xe6,
0xa7, 0xf9, 0x1b, 0x45, 0xc6, 0x98, 0x7a, 0x24,
0xf8, 0xa6, 0x44, 0x1a, 0x99, 0xc7, 0x25, 0x7b,
0x3a, 0x64, 0x86, 0xd8, 0x5b, 0x05, 0xe7, 0xb9,
0x8c, 0xd2, 0x30, 0x6e, 0xed, 0xb3, 0x51, 0x0f,
0x4e, 0x10, 0xf2, 0xac, 0x2f, 0x71, 0x93, 0xcd,
0x11, 0x4f, 0xad, 0xf3, 0x70, 0x2e, 0xcc, 0x92,
0xd3, 0x8d, 0x6f, 0x31, 0xb2, 0xec, 0x0e, 0x50,
0xaf, 0xf1, 0x13, 0x4d, 0xce, 0x90, 0x72, 0x2c,
0x6d, 0x33, 0xd1, 0x8f, 0x0c, 0x52, 0xb0, 0xee,
0x32, 0x6c, 0x8e, 0xd0, 0x53, 0x0d, 0xef, 0xb1,
0xf0, 0xae, 0x4c, 0x12, 0x91, 0xcf, 0x2d, 0x73,
0xca, 0x94, 0x76, 0x28, 0xab, 0xf5, 0x17, 0x49,
0x08, 0x56, 0xb4, 0xea, 0x69, 0x37, 0xd5, 0x8b,
0x57, 0x09, 0xeb, 0xb5, 0x36, 0x68, 0x8a, 0xd4,
0x95, 0xcb, 0x29, 0x77, 0xf4, 0xaa, 0x48, 0x16,
0xe9, 0xb7, 0x55, 0x0b, 0x88, 0xd6, 0x34, 0x6a,
0x2b, 0x75, 0x97, 0xc9, 0x4a, 0x14, 0xf6, 0xa8,
0x74, 0x2a, 0xc8, 0x96, 0x15, 0x4b, 0xa9, 0xf7,
0xb6, 0xe8, 0x0a, 0x54, 0xd7, 0x89, 0x6b, 0x35,
};

```

```

unsigned char CRC8_Table(unsigned char *p, char counter)

```

```

{

    unsigned char crc8 = 0;

    for( ; counter > 0; counter--)

    {

        crc8 = CRC8Table[crc8^*p]; //查表得到 CRC 码
        p++;

    }

    return crc8;

}

```