作者：粤嵌.温子祺

1 ymodem协议简介

ymodem协议是一个文件传输协议，由Chuck Forsberg于上世纪90年代开发完成，通常用于资源受限的设备。xmodem、ymodem和zmodem协议是最常用的三种通信协议。

ymodem协议是由xmodem协议演变而来的，是一种发送并等待的协议，即发送方发送一个数据包以后，都 要等待接收方的确认。如果是 ACK 信号，则可以发送新的包。如果是 NAK 信号， 则重发或者错误退出。

ymodem-1k用1024字节信息块传输取代标准的128字节传输，每包数据可以达到1024字节，是一个非常高效的文件传输协议，所用到的符号如下。

```
#define MODEM_SOH 0x01 //数据块起始字符
#define MODEM_STX 0x02 //1028字节开始
#define MODEM_EOT 0x04 //文件传输结束
#define MODEM_ACK 0x06 //确认应答
#define MODEM_NAK 0x15 //出现错误
#define MODEM_CAN 0x18 //取消传输
#define MODEM_C 0x43 //大写字母C
```

2 ymodem帧格式

2.1 ymodem协议的传输过程

如下图所示:

```
Figure 3.  YMODEM Batch Transmission Session (1 file)

SENDER                                          RECEIVER
                                                "sb foo.*<CR>"
"sending in batch mode etc."
                                                C (command:rb)
SOH 00 FF foo.c NUL[123] CRC CRC
                                                ACK
                                                C
SOH 01 FE Data[128] CRC CRC
                                                ACK
SOH 02 FC Data[128] CRC CRC
                                                ACK
SOH 03 FB Data[100] CPMEOF[28] CRC CRC
                                                ACK
EOT
                                                NAK
EOT
                                                ACK
                                                C
SOH 00 FF NUL[128] CRC CRC
                                                ACK
```

开启是由接收方开启传输，它发一个大写字母 C（0x43） 开启传输。然后进入 等待 SOH（0x01）)状态，如果没有回应，就会超时退出。

发送方开始时处于等待过程中，等待 C。收到 C 以后，发送(SOH)数 据包开始信号，发送序号(00)，补码(FF)，“文件名”，“\0” “文件大小” “除去序号外，补满 128 字节”，16位CRC 校验两个字节，高字节在前，低字节在后。进入等待(ACK)状态。

```
1 内容示例： SOH 00 FF Foo.bin NUL[123] CRC CRC
```

接收方收到以后，CRC 校验满足，则发送 ACK。发送方接收到 ACK，又 进入等待“文件传输开启”信号，即重新进入等待“C”的状态。

发送接收到“C”以后，发送第一个数据包，(SOH)(01序号)(F E补码)(128位数据)(CRC校验)，或者(STX)(01序号)(F E补码)(1024位数据)(CRC校验)，不满

128或者1024，用0x00补齐，等待接收方“ACK”。

```
- 内容示例：STX 01 FE data[1024] CRC CRC
```

文件发送完以后，发送方发出一个“EOT”信号，接收方也以“ACK”回应。然后接收方会再次发出“C”开启另一次传输，若接着发送方会发出一个“全0数据包”，接收方“ACK”以后，本次通信正式结束。

2.2 起始帧的数据格式

ymodem的起始帧并不直接传输文件的数据，而是将文件名与文件的大小放在数据帧中传输，它的帧长=3字节数据首部+128字节数据+2字节CRC16校验码=133字节。它的数据结构如下：。

```
SOH 00 FF foo.c 3232 NUL[118] CRCH CRCL
```

- SOH：表示本帧数据块大小为128字节
- 00： 表示数据帧序号，初始是0，依次向下递增，FF是帧序号的取反
- foo.c：是要传输的文件名，是ASCII字符串（以空字符结尾）
- 3232：表示文件的大小，是ASCII字符串（以空字符结尾）
- NUL[118]：剩余部分用空字符填充
- CRCH/L： 表示16位CRC校验码的高8位与低8位

2.3 数据帧的数据格式

ymodem的数据帧的数据块大小可以是128字节或者1024字节。

```
// 128字节的数据块
SOH 01 FE data[128] CRCH CRCL

// 1024字节的数据块
STX 01 FE data[1024] CRCH CRCL
```

一般会使用1024字节的数据块进行传输，这样可以加快传输速度，如果最后文件数据不足1024字节，则将其拆分为128字节的数据块进行传输，如果拆分后有不足128字节的数据依然按照128字节的数据块进行传输，但是剩余空间全部用0x1A填充，以表示文件结束。

2.4 结束帧数据结构

当文件传输结束时，除了发送EOT传输结束指令外，还需要发送一个结束帧。ymodem的结束帧与起始帧的数据格式相同，数据块大小为128字节，但是结束帧的数据块要全用空字符填充。

```
SOH 3A C5 NUL[128] CRCH CRCL
```

3 crc16函数

```
/**
* @bieaf CRC-16 校验
*
* @param addr 开始地址
* @param num 长度
* @param num CRC
* @return crc 返回CRC的值
*/
#define POLY 0x1021
 uint16_t crc16(uint8_t *addr, int32_t num, uint16_t crc)
 {
  int32_t i;
  for (; num > 0; num--) /* Step through bytes in memory */
  {
  crc = crc ^ (*addr++ << 8); /* Fetch byte from memory, XOR into CRC top byte*/
  for (i = 0; i < 8; i++) /* Prepare to rotate 8 bits */
  {
  if (crc & 0x8000) /* b15 is set... */
  crc = (crc << 1) ^ POLY; /* rotate and XOR with polynomic */
  else /* b15 is clear... */
  crc <<= 1; /* just rotate */
  } /* Loop for 8 bits */
  crc &= 0xFFFF; /* Ensure CRC remains 16-bit value */
  } /* Loop until num=0 */
  return(crc); /* Return updated CRC */
 }
```

4 ymodem传输大小选择

在SecureCRT中，ymodem默认为128字节数据包大小，如下图。当然亦可选择为1024字节（Xmodem-1k/Ymodem-1k）