

## Xmodem 文件传输协议

### 1.1 Xmodem 简介

Xmodem 协议是一种使用拨号调制解调器的个人计算机通信中广泛使用的异步文件传输协议。标准分为 Xmodem 和 1k-Xmodem 两种,前者以 128 字节块的形式传输数据,后者字节块为 1k 即 1024 字节,并且支持一般校验和、CRC 两种校验方式,在出现数据包错误的情况下支持多次重传(一般为 10 次)。Xmodem 协议传输由接收程序和发送程序完成。先由接收程序发送协商字符,协商校验方式,协商通过之后发送程序就开始发送数据包,接收程序接收到完整的一个数据包之后按照协商的方式对数据包进行校验。校验通过之后发送确认字符,然后发送程序继续发送下一包;如果校验失败,则发送否认字符,发送程序重传此数据包。由于 Xmodem 需要对每个块都进行认可,这将导致性能有所下降,特别是延时比较长的场合,这种协议显得效率更低。

除了 Xmodem,还有 Ymodem, Zmodem 协议。他们的协议内容和 Xmodem 类似,不同的是 Ymodem 允许批处理文件传输,效率更高;Zmodem 则是改进的了 Xmodem,它只需要对损坏的块进行重发,其它正确的块不需要发送确认字节。减少了通信量。

### 1.2 Xmodem 协议

#### 1.2.1 相关定义说明

<SOH>	01H	// Xmodem 数据头
<STX>	02H	// 1K-Xmodem 数据头
<EOT>	04H	// 发送结束
<ACK>	06H	// 认可响应
<NAK>	15H	// 不认可响应
<CAN>	18H	// 撤销传送
<EOF>	1AH	// 填充数据包

#### 1.2.2 协议简介

Xmodem 协议的传输数据单位为信息包,包含一个标题开始字符<SOH>或者<STX>,一个单字节包序号,一个包序号的补码,128/1024 个字节数据和两个双字节的 CRC16 校验。它把数据划分成 128/1024 个字符的小包进行发送,每发送一个小包都要检查是否正确,如果信息包正确接收方发送一个字节<ACK>的应答;有错重发则发送一个字节<NAK>应答,要求重发。因此 1K-Xmodem 是一种发送等待协议,具有流量控制功能。优点:简单通用,几乎所有通信软件都支持该协议。缺点:慢。Xmodem 包格式如下表所示。

表 1 Xmodem 包格式

Byte1	Byte2	Byte3	Byte4~131	Byte132~133
Start Of Header	Packet Number	~(Packet Number)	Packet Data	16-Bit CRC

表 2 1K-Xmodem 包格式

Byte1	Byte2	Byte3	Byte4~1027	Byte1028~1029
Start Of Header	Packet Number	~(Packet Number)	Packet Data	16-Bit CRC

### 1.2.3 数据包说明

对于标准 Xmodem 协议来说, 如果传送的文件不是 128 或者 1024 的整数倍, 那么最后一个数据包的有效内容肯定小于帧长, 不足的部分需要用 CTRL-Z(0x1A)来填充。如果传送的是 bootloader 工程生成的.bin 文件, mcu 收到后遇到 0x1A 字符会怎么处理? 其实如果传送的是文本文件, 那么接收方对于接收的内容是很容易识别的, 因为 CTRL-Z 不是前 128 个 ascii 码, 不是通用可见字符, 如果是二进制文件, mcu 其实也不会把它当作代码来执行。哪怕是 excel 文件等, 由于其内部会有些结构表示各个字段长度等, 所以不会读取多余的填充字符。否则 Xmodem 太弱了。对于 1k-Xmodem, 同上理。

### 1.2.4 如何启动传输

传输由接收方启动, 方法是向发送方发送"C"或者 NAK(注意哦, 这里提到的"C"是用来启动传输的。以下我们会看到"C"还可以用来对数据产生重传的机制)。接收方发送 NAK 信号表示接收方打算用累加和校验; 发送字符"C"则表示接收方想打算使用 CRC 校验。

### 1.2.5 传输过程

当接收方发送的第一个"C"或者 NAK 到达发送方, 发送方认为可以发送第一个数据包, 传输已经启动。发送方接着应该将数据以每次 128 字节的数据加上包头, 包号, 包号补码, 末尾加上校验和, 打包成帧格式传送。

发送方发了第一包后就等待接收方的确认字节 ACK, 收到接收方传来的 ACK 确认, 就认为数据包被接收方正确接收, 并且接收方要求发送方继续发送下一个包; 如果发送方收到接收方传来的 NAK(这里, NAK 用来告诉发送方重传, 不是用来启动传输)字节, 则表示接收方请求重发刚才的数据包; 如果发送方收到接收方传来的 CAN 字节, 则表示接收方请求无条件停止传输。

### 1.2.6 如何结束传输

如果发送方正常传输完全部数据, 需要结束传输, 正常结束需要发送方发送 EOT 字节通知接收方。接收方回以 ACK 进行确认。当然接收方也可强制停止传输, 当接收方发送 CAN 字节给发送方, 表示接收方想无条件停止传输, 发送方收到 CAN 后, 不需要再发送 EOT 确认。

### 1.2.7 特殊处理

虽然数据包是以 SOH 来标志一个信息包的起始的, 但在 SOH 位置上如果出现 EOT 则表示数据传输结束, 再也没有数据传过来。接收方首先应确认数据包序号的完整性, 通过对数据包序号取补, 然后和数据包序号的补码异或, 结果为 0 表示正确, 结果不为 0 则发送 NAK 请求重传。

接收方确认数据包序号正确后, 然后检查是否期望的序号。如果不是期望得到的数据包序号, 说明发生严重错误, 应该发送一个 CAN 来中止传输。如果接收到的数据包的包序号和前一包相同, 那么接收方会忽略这个重复包, 向发送方发出 ACK, 准备接收下一个包。

接收方确认了信息包序号的完整性和是正确期望的后, 只对 128 字节的数据区段进行算术和校验, 结

果与帧中最后一个字节（算术校验和）比较，相同发送 ACK，不同发送 NAK。

### 1.2.8 校验和的说明

Xmodem 协议支持 2 种校验和，它们是累加和与 CRC 校验。

当接收方一开始启动传输时发送的是 NAK，表示它希望以累加和方式校验。

当接收方一开始启动传输时发送的是字符“C”，表示它希望以 CRC 方式校验。

### 1.2.9 传输逻辑

1、收发双方拨号连通后，发送方等待接收方传来 NAK 信号。当第一个 NAK 到达，发送方解释为开始发送第一个包。

2、发送方一旦收到第一个 NAK，启动了传输，发送方就将数据以每次 128/1024 字节打包成帧格式传送，再等待接收方的确认信号。

3、发送方收到接收方传来的 ACK 信号，解释为信息包被正确接收，并有发送下一个包的含义。

4、发送方收到接收方传来的 NAK 信号，解释为请求重发同一数据包。

5、发送方收到接收方传来的 CAN 信号，解释为请求无条件停止传输过程。

6、发送方正常传输完全部数据，需要正常结束，发送 EOT 信号通知接收方。接收方用 ACK 进行确认。

7、接收方发送 CAN 无条件停止传输过程，发送方收到 CAN 后，不发送 EOT 确认。

8、虽然信息包是以 SOH 来标志一个信息包的起始的，但在 SOH 位置上出现的 EOT 则表示数据传输结束，再也没有数据传过来。

9、接收方首先应确认信息包序号的完整性，通过对信息包序号取补，然后和信息包序号的补码异或，结果为 0 表示正确，结果不为 0 则发送 NAK 请求重传。

10、接收方确认信息包序号正确后，然后检查是否期望的序号。如果不是期望得到的信息包序号，说明发生严重错误，应该发送一个 CAN 来中止传输。

11、对于 10 情况的唯一例外，是收到的包的信息包序号与前一个信息包序号相同，此中情况，接收方简单忽略这个重复的包，向发送方发出 ACK，准备接收下一个包。

12、接收方确认了信息包序号的完整性和是正确期望的后，对数据区段进行算术和校验，结果与帧中最后一个字节（算术校验和）比较，相同发送 ACK，不同发送 NAK。

### 1.2.10 超时处理

1、接收方等待一个信息包的到来所具有的超时时限为 10 秒，每个超时后发送 NAK。

2、当收到包时，接收过程中每个字符的超时间隔为 1 秒。

3、为保持“接收方驱动”，发送方在等待一个启动字节时不应该采用超时处理。

4、一旦传输开始，发送方采用单独的 1 分钟超时时限，给接收方充足的时间做发送 ACK，NAK，CAN 之前的必须处理。

5、所有的超时及错误事件至少重试 10 次。

## 1.2.11 Xmodem 协议发送流程图

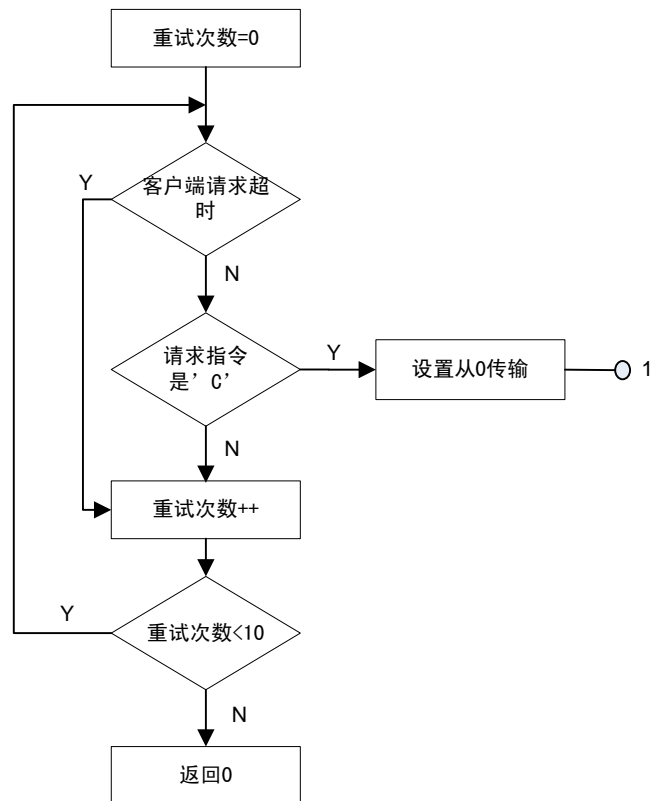


图 1 Xmodem 服务器接收启动发送指令

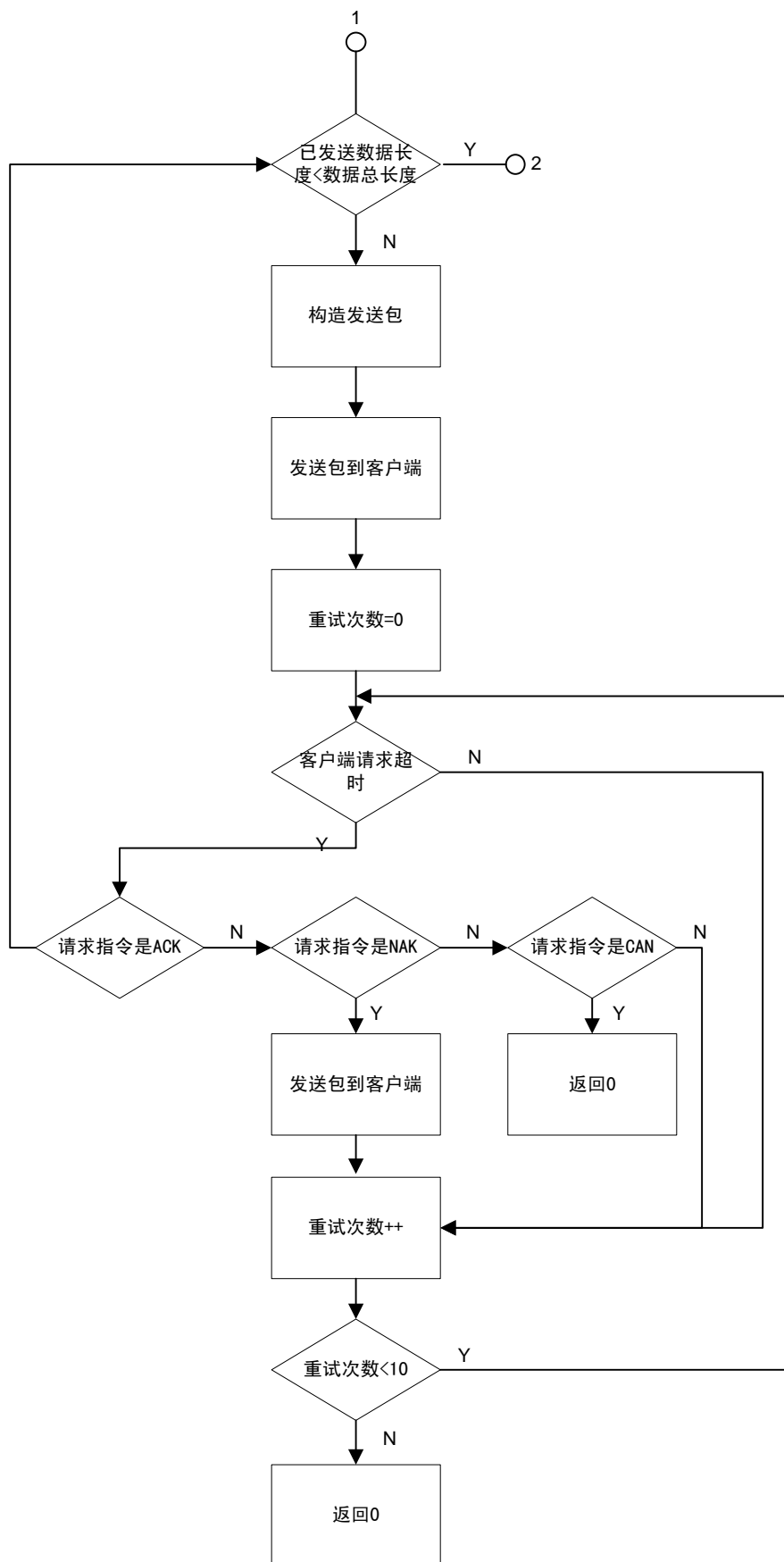


图 2 Xmodem 发送数据

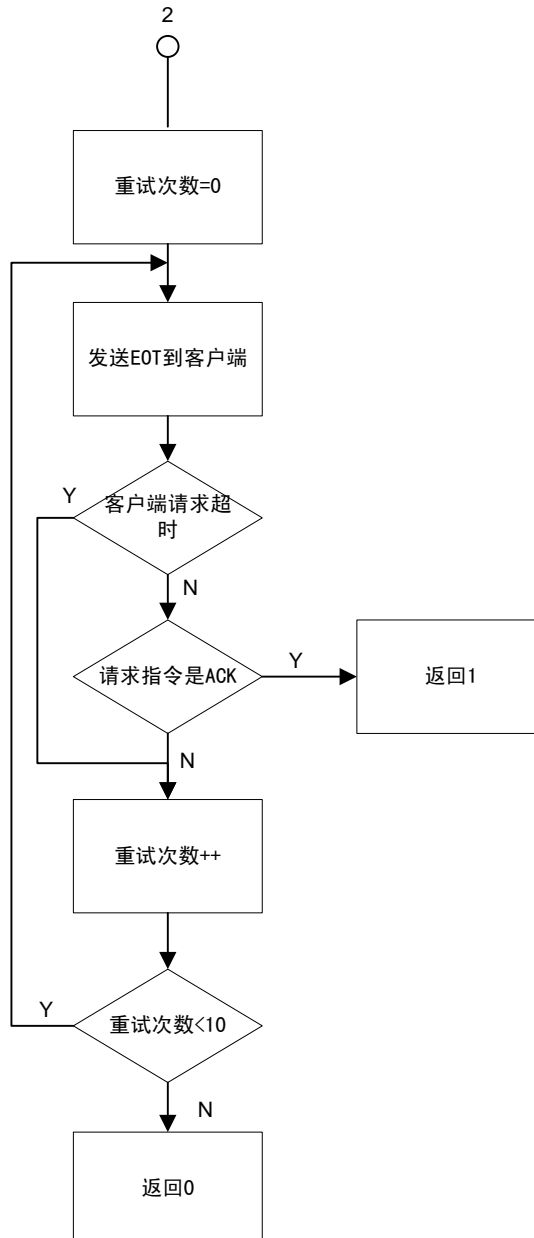


图 3 Xmodem 发送结束处理

### 1.2.12 CRC 校验信息包

#### 1. CRC 校验信息包

```
<STX><blk #><255-blk #><--1024 data bytes--><CRC hi><CRC lo>
```

其中:

< STX > = 02 hex

<blk #> = 信息包序号, 从 01 开始以发送一包将加 1, 加到 FF hex 将循环。

<255-blk #> = 信息包序号的补码。

<CRC hi> = CRC16 高字节。

<CRC lo> = CRC16 低字节。

## 2. CRC 描述

计算 16-Bit CRC 校验的除数多项式为  $X^{16} + X^{12} + X^5 + 1$ , 信息包中的 1024 数据字节将参加 CRC 校验的计算。在发送端 CRC16 的高字节在先, 低字节在后。

## 3. CRC 校验方式数据传输流程

接收方要求发送方以 CRC 校验方式发送时以‘C’来请求, 发送方将对此做出应答。如表 3 所示传输 3 包数据的示意过程。

表 3 CRC 校验传输过程

Sender					Flow	Receiver
					←	‘C’
						Time out
					←	‘C’
STX	0x01	0xFE	Data[1-1024]	CRC16	→	Packet OK
					←	ACK
STX	0x02	0xFD	Data[1025-2048]	CRC16	→	Line hit during transmission
					←	NAK
STX	0x02	0xFD	Data[1025-2048]	CRC16	→	Packet OK
					←	ACK
STX	0x03	0xFC	Data[2049-3072]	CRC16	→	Packet OK
					←	ACK
EOT					→	Packet OK
ACK get garbaged					←	ACK
EOT					→	Packet OK
Finished					←	ACK

## 1.3 支持 Xmodem 服务器软件

### 1.3.1 超级终端

超级终端是 Win 2000 系统自带的调试工具, 支持串口, 以太网等多种调试方式, 支持的串口传输协议也非常的丰富, 可惜 windows7 默认已经不再提供“超级终端”组件。

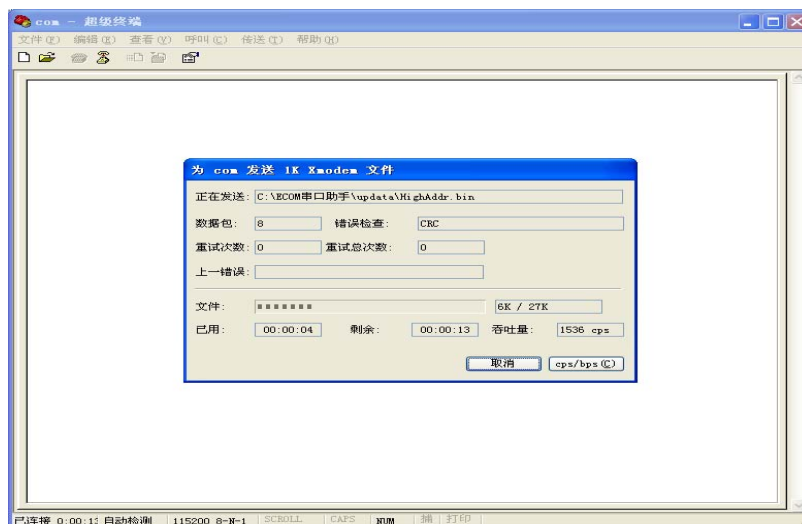


图 4 超级终端

### 1.3.2 Ecom 串口助手

ecom 串口助手是一款串口(RS232)调试软件。由我爱 IC 导航网(<http://www.52ic.net>)工作室开发。ecom 串口助手支持常用的 1200 ~ 921600bps 波特率, 能设置校验、数据位和停止位, 能以 ASCII 码或十六进制接收或发送任何数据或字符(包括中文), 能发送任意大小的文本文件, 可以任意设定自动发送周期, 并能将接收数据保存成文本文件。文件传输协议支持 Xmodem, 1K-Xmodem, 同时支持命令控制 1K-Xmodem 自动发送文件。

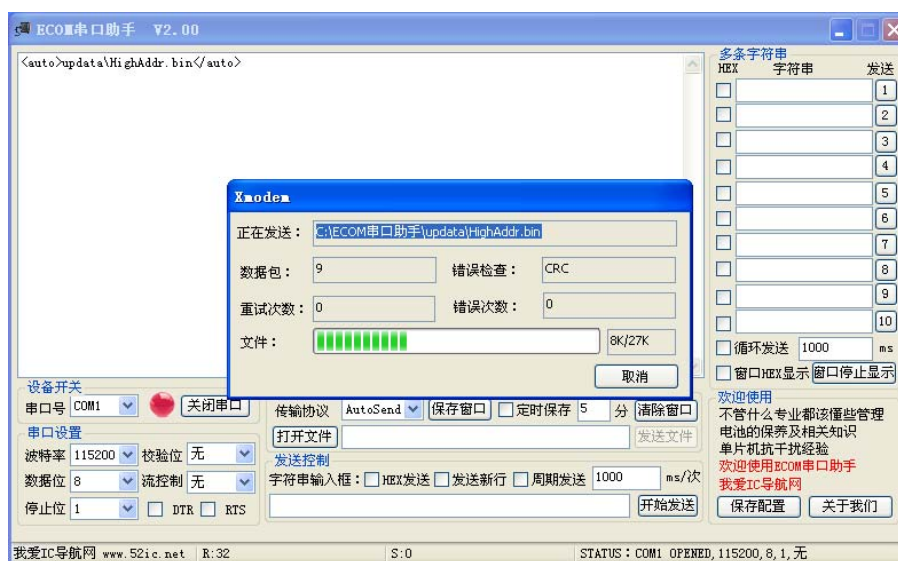


图 5 ECOM 串口助手