EN32F016 IAP UART 协议

目录

1. Ymodem 帧格式 2
1.1 帧头 2
1.2 包序号 2
1.3 帧长度 2
1.4 校验 2
2.Ymodem 握手信号 2
3.Ymodem 起始帧 2
4.Ymodem 数据帧 2
5.Ymodem 结束帧 3
6.Ymodem 命令 3
6.1、起始帧的数据格式 3
6.2、数据帧的数据格式 4
6.3、结束帧数据结构 4
6.4、文件传输过程 4

1. Ymodem 帧格式

Ymodem 有两种帧格式，主要区别是信息块长度不一样。

名称	帧头	包号	包号反码	信息块	校验
简写	SOH/STX	PN	XPN	DATA	CRC
字节数	1	1	1	1024/128	2

1.1 帧头

帧头表示两种数据帧长度，主要是信息块长度不同。

帧头	SOH（0x01）	STX（0x02）
信息块长度	128字节	1024字节

1.2 包序号

数据包序号只有 1 字节，因此计算范围是 0~255；对于数据包大于 255 的，序号归零重复计算。

1.3 帧长度

1) 以 SOH(0x01)开始的数据包，信息块是 128 字节，该类型帧总长度为 133 字节。

2) 以 STX(0x02)开始的数据包，信息块是 1024 字节，该类型帧总长度为 1029 字节。

1.4 校验

Ymodem 采用的是 CRC-16/CCITT-FALSE 校验算法，校验值为 2 字节，传输时 CRC 高八位在前，低八位在后；CRC 计算数据为信息块数据，不包含帧头、包号、包号反码。

2. Ymodem 握手信号

握手信号由接收方发起，在发送方开始传输文件前，接收方需发送 YMODEM_C （字符 C，ASII 码为 0x43）命令，发送方收到后，开始传输起始帧。

3. Ymodem 起始帧

Ymodem 起始帧并不直接传输文件内容，而是先将文件名和文件大小置于数据帧中传输；起始帧是以 SOH 133 字节长度帧传输，格式如下。

帧头	包号	包号反码	文件名称	文件大小	填充区	校验高位	校验低位
SOH	0x00	0xff	File name+0x00	File size+0x00	NULL（0x00）	CRC-H	CRC-L

其中包号为固定为 0;Filename 为文件名称，文件名称后必须加 0x00 作为结束;Filesize 为文件大小值，文件大小值后必须加 0x00 作为结束；余下未满 128 字节数据区域，则以 0x00 填充。

4. Ymodem 数据帧

Ymodem 数据帧传输，在信息块填充有效数据。

帧头	包号	包号反码	有效数据	校验高位	校验低位
SOH/STX	PN	XPN	DATA	CRC-H	CRC-L

传输有效数据时主要考虑的是最后一包数据的是处理，SOH 帧和 STR 帧有不同的处理。

1）对于 SOH 帧，若余下数据小于 128 字节，则以 0x1A 填充，该帧长度仍为 133 字节。

2）对于 STX 帧需考虑几种情况：

●余下数据等于 1024 字节，以 1029 长度帧发送；

●余下数据小于 1024 字节，但大于 128 字节，以 1029 字节帧长度发送，无效数据以 0x1A 填充。

●余下数据等于 128 字节，以 133 字节帧长度发送。

●余下数据小于 128 字节，以 133 字节帧长度发送，无效数据以 0x1A 填充。

5. Ymodem 结束帧

Ymodem 的结束帧采用 SOH 133 字节长度帧传输，该帧不携带数据（空包），即数据区、校验都以 0x00 填充。

帧头	包号	包号反码	数据区	校验高位	校验低位
SOH	0x00	0xff	0x00	0x00	0x00

6. Ymodem 命令

命令	命令码	备注
YMODEM_SOH	0x01	133字节长度帧
YMODEM_STX	0x02	1024字节长度帧
YMODEM_EOT	0x04	文件传输结束命令
YMODEM_ACK	0x06	接收正确应答命令
YMODEM_NAK	0x15	重传当前数据包请求命令
YMODEM_CAN	0x18	取消传输命令，连续发送5个该命令
YMODEM_C	0x43	字符C

YModem-1K 用 1024 字节信息块传输取代标准的 128 字节传输，数据的发送回使用 CRC 校验，保证数据传输的正确性。它每传输一个信息块数据时，就会等待接收端回应 ACK 信号，接收到回应后，才会继续传输下一个信息块，保证数据已经全部接收。

6.1、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=33 字节。它的数据结构如下：

SOH 00 FF filename[ ] filezise[ ] NUL[ ] CRCH CRCL

其中 SOH=0x01，表示这个数据帧中包含着 128 字节的数据部分；在 SOH 后面的 00 FF，00 表示数据帧序号，因为是起始帧，所以它的帧序为 00，至于 FF，它是帧序的取反，YModem 特地这么做是为了给数据是否正确提供一种判断依据，通过判断这两个字节是否为取反关系，就可以知道数据是否传输出错；filename[ ]就是文件名，如文件名 foo.c，它在数据帧中

存放格式为：66 6F 6F 2E 63 00，一定要在文件名最后跟上一个 00，表示文件名结束；filesize[]就是文件大小，如上面的 foo.c 的大小为 1KByte，即 1024Byte，需要先将它转化成 16 进制，即 0x400，所以它在数据帧的存放格式为：34 30 30 00，即“400”，同样的文件大小最后需要跟上 00，表示结束；NUL[]表示剩下的字节都用 00 填充，数据部分大小为 128 字节,除去文件名与文件大小占用的空间外,剩余的字节全部用 00 填充;CRCH CRCL 分别表示 16 位 CRC 校验码的高 8 位与低 8 位。

6.2、数据帧的数据格式

YModem 的数据帧中会预留 1024 字节空间用来传输文件数据,它跟起始帧接收差不多,如下：

STX 01 FE data[1024] CRCH CRCL

其中 STX=0x02，表示这帧数据帧后面包含着 1024 字节的数据部分；STX 后面的 01 FE，01 表示第一帧数据帧,FE 则是它的取反,当然如果是第二帧数据的话就是：02 FD；data[1024] 表示存放着 1024 字节的文件数据；CRCH 与 CRCL 是 CRC16 检验码的高 8 位与低 8 位。

如果文件数据的最后剩余的数据在 128~1024 之前，则还是使用 STX 的 1024 字节传输，但是剩余空间全部用 0x1A 填充，如下结构：

STX [num] [~num] data[ ] 1A …1A CRCH CRCL

有一种特殊的情况：如果文件大小小于等于 128 字节或者文件数据最后剩余的数据小于 128 字节，则 YModem 会选择 SOH 数据帧用 128 字节来传输数据，如果数据不满 128 字节，剩余的数据用 0x1A 填充这是数据正的结构就变成了：

文件大小小于 128 字节： SOH 01 FE data[ ] 1A …1A CRCH CRCL

文件最后剩余数据小于 128 字节：SOH [num] [~~num] data[ ] 1A…1A CRCH CRCL

6.3、结束帧数据结构

YModem 的结束帧数据也采用 SOH 的 128 字节数据帧，它的结构如下：

SOH 00 FF NUL[128] CRCH CRCL

结束帧同样以 SOH 开头，表示后面跟着 128 字节大小的数据；结束帧的帧序也认为是 00 FF；结束帧的 128 字节的数据部分不存放任何信息，即 NUL[128]全部用 00 填充。

6.4、文件传输过程

文件的传输过程，以具体的例子说明。把 foo.c，大小为 4196Byte（16 进制为 0x1064）的文件作为传输的对象，则它的传输过程如下：

发送端---接收端

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< C

SOH 00 FF “foo.c” "1064’’ NUL[118] CRC CRC >>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< C

STX 01 FE data[1024] CRC CRC>>>>>>>>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

STX 02 FD data[1024] CRC CRC>>>>>>>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

STX 03 FC data[1024] CRC CRC>>>>>>>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

STX 04 FB data[1024] CRC CRC>>>>>>>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

SOH 05 FA data[100] 1A[28] CRC CRC>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

EOT >>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< NAK

EOT>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< C

SOH 00 FF NUL[128] CRC CRC >>>>>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ACK

YModem 的传输过程就是上面所示。但是上面传输过程中存在许多通信信号，它们的数值与意义如下表所示：

符号	数值	含义
SOH	0x01	128字节数据包
STX	0x02	1024字节数据包
EOT	0x04	结束传输
ACK	0x06	回应
NAK	0x15	不回应
CA	0x18	传输中止
C	0x43	请求数据包

还是有几点需要说明下：

1）EOT 信号由发送端发送

2）CA 中止传输信号也有发送端发送

5.CRC16 校验函数

```
uint16_t CRC_16_CCITT_FALSE(char* pData, int length)
{
    uint8_t i;
    uint16_t wCrc = 0xffff;
    while (length--) {
        wCrc ^= *(unsigned char *)pData++ << 8;
        for (i=0; i < 8; i++)
            wCrc = wCrc & 0x8000 ? (wCrc << 1) ^ 0x1021 : wCrc << 1;
    }
    return wCrc & 0xffff;
}
```