应用手册

IAP 实现原理

适用范围：

- Cortex m0+全系列

文档说明

本应用手册将详细介绍 SYM32 Cortex-m0+系列微控制器的 IAP 的实现原理。

1 IAP 是什么

IAP 是 In Application Programming 的首字母缩写，通常称作在应用编程。其典型应用场景是产品在正常工作时对自己的固件进行编程以实现对产品功能的修改或升级。ICP、ISP 需要通过特定的工具才能修改产品内的固件；IAP 则仅需预留的通信接口即可修改产品内的固件。

具有 IAP 功能的产品，其闪存区一般存在两个相对独立的固件：Bootloader 和 APP；均需要通过编程工具写入到产品闪存中。BootLoader 负责启动 APP 或更新 APP；APP 负责实现产品的用户级功能。

芯片复位后首先执行 Bootloader，Bootloader 通过一定的逻辑判断以决定启动 APP 执行用户级功能还是执行更新 APP 的流程。

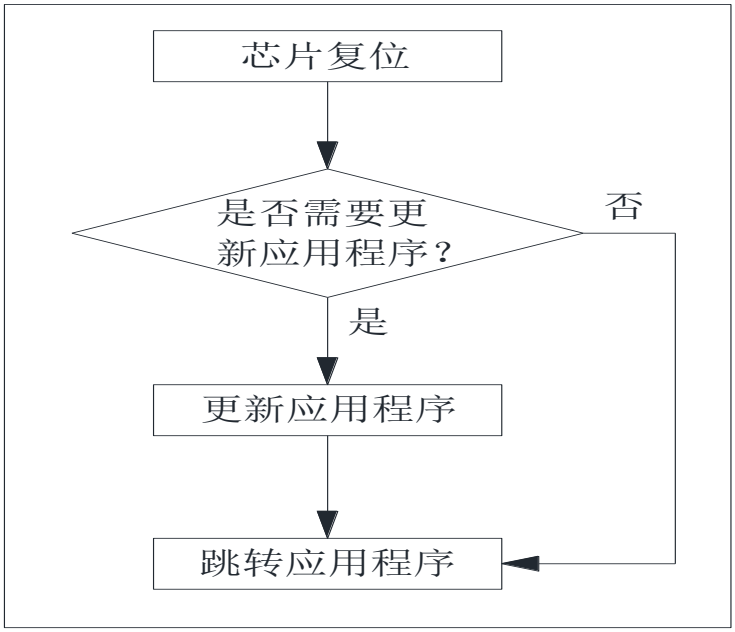

图 1-1 Bootloader 代码执行流程

2 IAP/APP 执行原理

本应用笔记以嵌入式 64KB 的闪存为例，详述内存内程序分布

2.1 无 IAP 功能的闪存空间分布

无 IAP 功能的程序在闪存内的分布如图 2-1 所示。

其中，0x0000 0000 地址存放堆栈指针的值，从 0x0000 0004 ~ 0x0000 00BF 地址存放中断向量表，从 0x0000 00C0 ~ 0x0000 FFFF 存放用户程序。即用户程序可以占用中断向量表后方的所有空间。

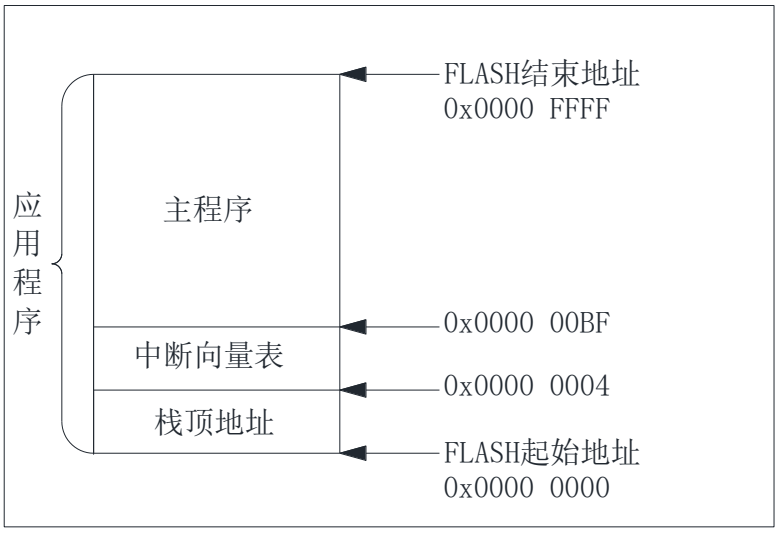

图 2-1 无 IAP 功能时 Flash 的存储分布图

2.2 有 IAP 功能的闪存空间分布

有 IAP 功能的程序分布如图 2-2 所示，假定 BootLoader 占用空间为 8KB。

0x0000 0000 地址存放 BootLoader 程序的堆栈指针的值，从 0x0000 0004～0x0000 00BF 地址存放 BootLoader 程序的中断向量表，从 0x0000 00C0～0x0000 1FFF 存放 BootLoader 的功能代码。

0x0000 2000 地址存放 APP 程序的堆栈指针的值，从 0x0000 2004～0x0000 20BF 地址存放 APP 程序的中断向量表，从 0x0000 20C0～0x0000 FFFF 存放 APP 的功能代码。

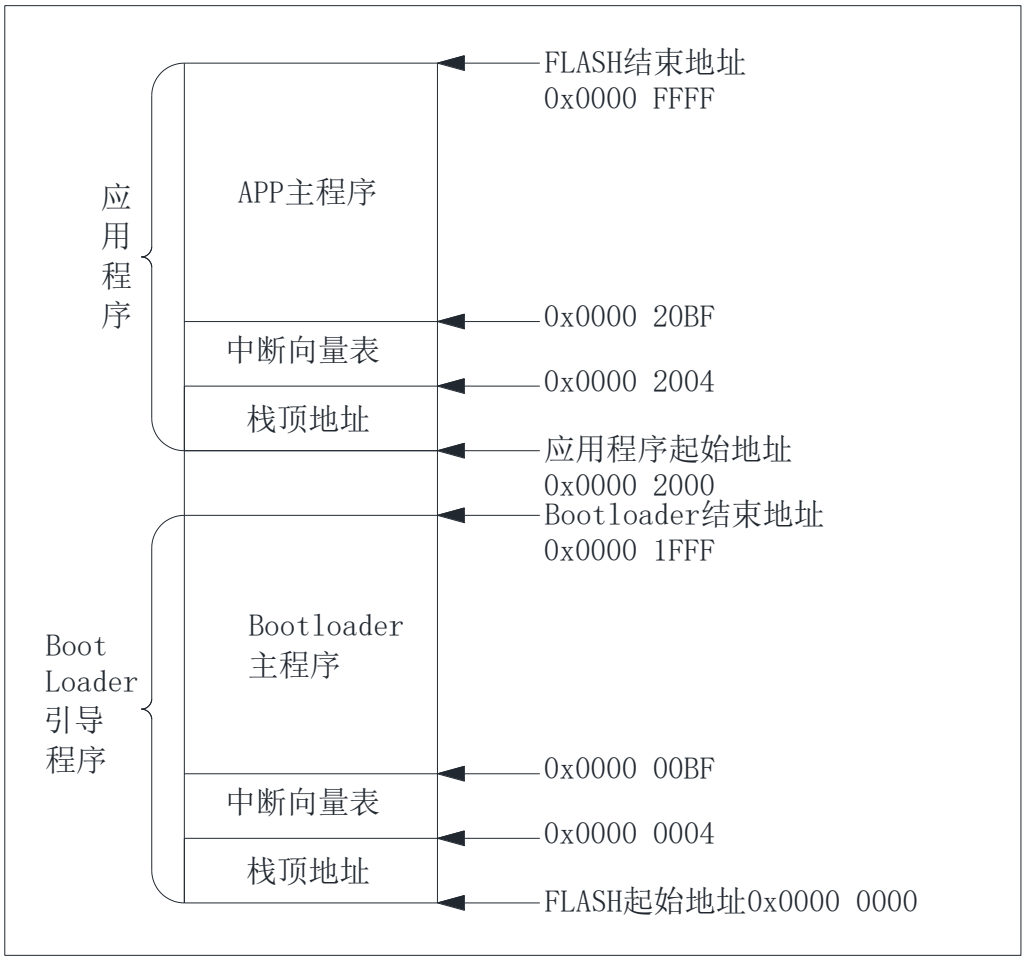

图 2-2 具有 IAP 功能时 Flash 的存储分布图

2.3 MCU 程序的执行原理

SYM32 芯片复位后，硬件电路从 0x0000 0000 地址处读取堆栈值并写入相应寄存器；然后从 0x0000 0004 地址处读取复位函数的地址并写入 PC 指针寄存器，芯片开始执行程序；当发生中断时，硬件从 SCB->VTOR 寄存器所定义的偏移地址读取中断向量地址并执行相应中断。

当 BootLoader 检测到需要执行用户程序时，需要依次执行如下步骤：

Step1. 向 SCB->VTOR 寄存器写入 APP 的起始地址（例：0x0000 2000）；

Step2. 从 APP 的起始地址读出 APP 的堆栈地址，然后写入到堆栈寄存器；

Step3. 从 APP 的复位向量地址读出 APP 的复位函数地址，并跳转到该地址。

完成以上步骤后，MCU 由完全由 APP 程序控制；可正常调用函数并执行中断服务程序。

3 编译程序注意事项

本手册以 MDK 编译环境为例，详细说明 BootLoader 程序和 APP 应用程序的工程配置。

3.1 Bootloader 程序工程配置

1、Bootloader 代码区域和 APP 应用程序代码区域不能重叠，Bootloader 代码是从 0x0000 开始的，配置如下：

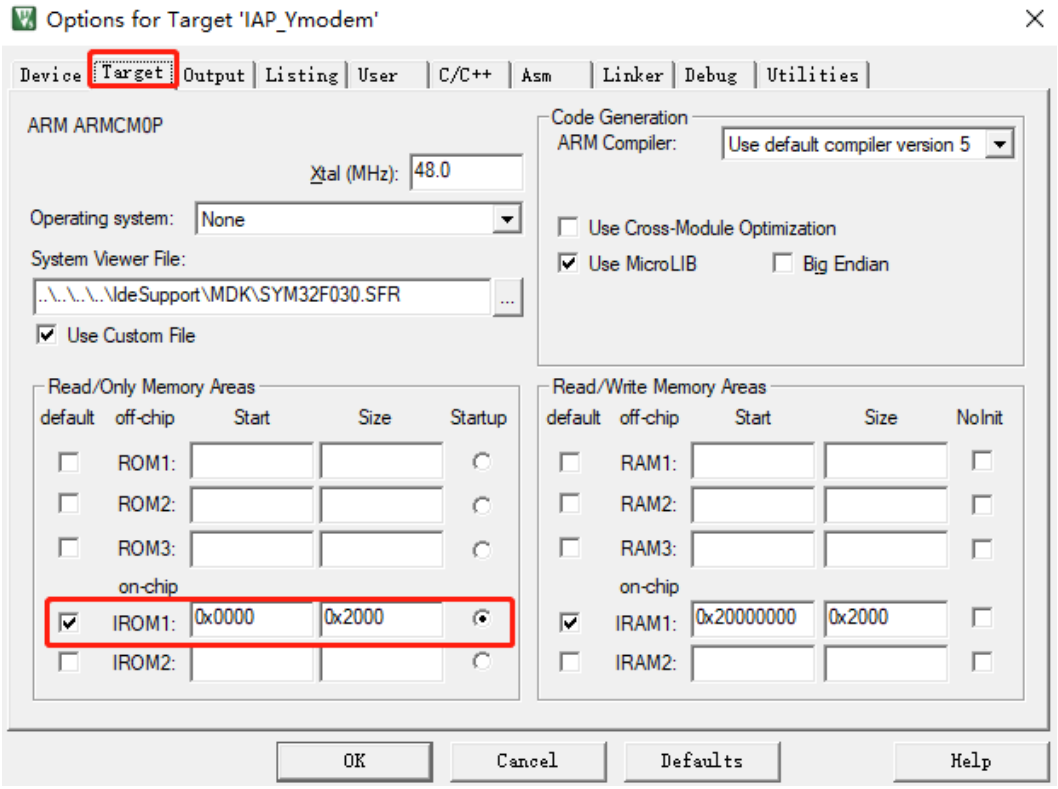

图 3-1Bootloader 程序工程配置

2、确认 Bootloader 代码的大小。

Bootloader 代码大小可以从编译结果来查看，其大小就是 Code + RO Data + RW Data，也可以从生成的 map 文件里查看，如下图：

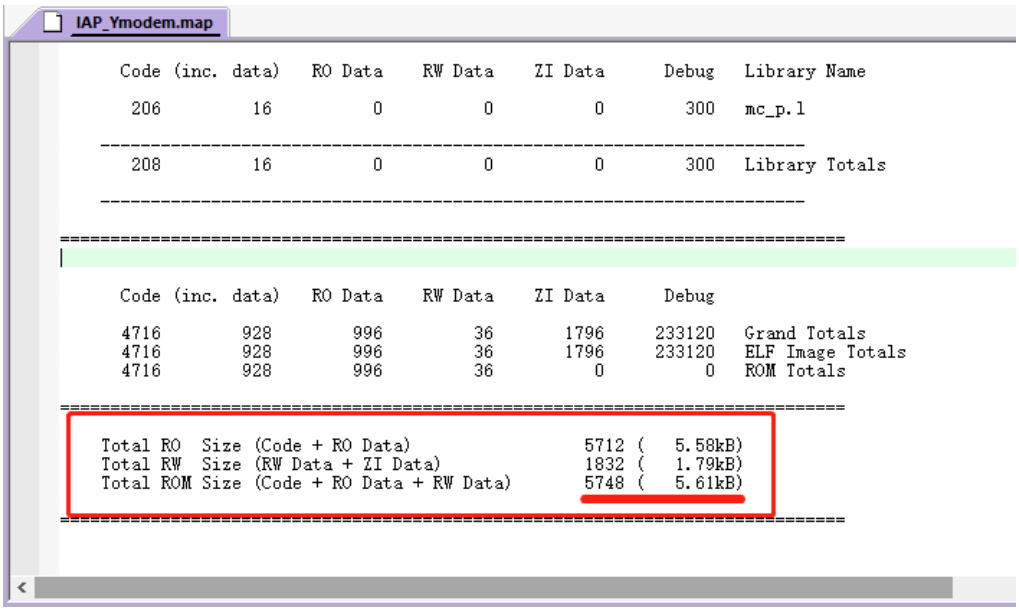

图 3-2 map 文件里查看代码大小

3.2 APP 应用程序工程配置

1、设置 Flash 起始地址

假设 Bootloader 代码大小不足 8K 字节，为了保险起见，给 Bootloader 程序分配 8K 字节的 FLASH 空间，由于 SYM32 的一个 PAGE 是 512 个字节， 所以 0x0000 0000 ~ 0x0000 1FFF 的 FLASH 空间用于存放 Bootloader 程序， 0x0000 2000 ~ 0x0000 FFFF 的 FLASH 空间用于存放 APP 应用程序，设置如下图

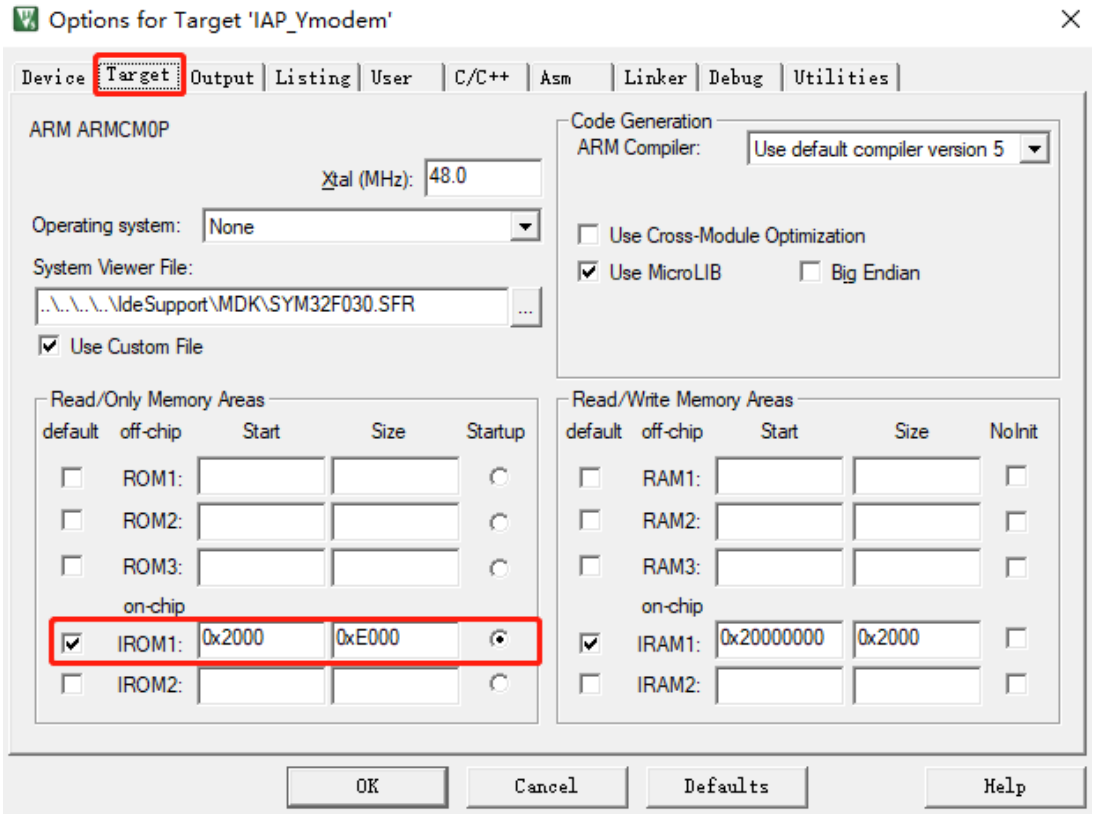

图 3-3APP 代码工程设置

2、设置 MDK 编译输出 bin 文件

APP 应用程序需要编译生成 bin 文件才能通过 IAP 烧写到 APP 应用程序 FLASH 区中，点开 Options for Target -> User 选项卡，勾选 Run#1，输入 fromelf.exe --bin -o "$L@L.bin" "#L"，如图 3-4，然后编译，可以生成 bin 格式的文件。

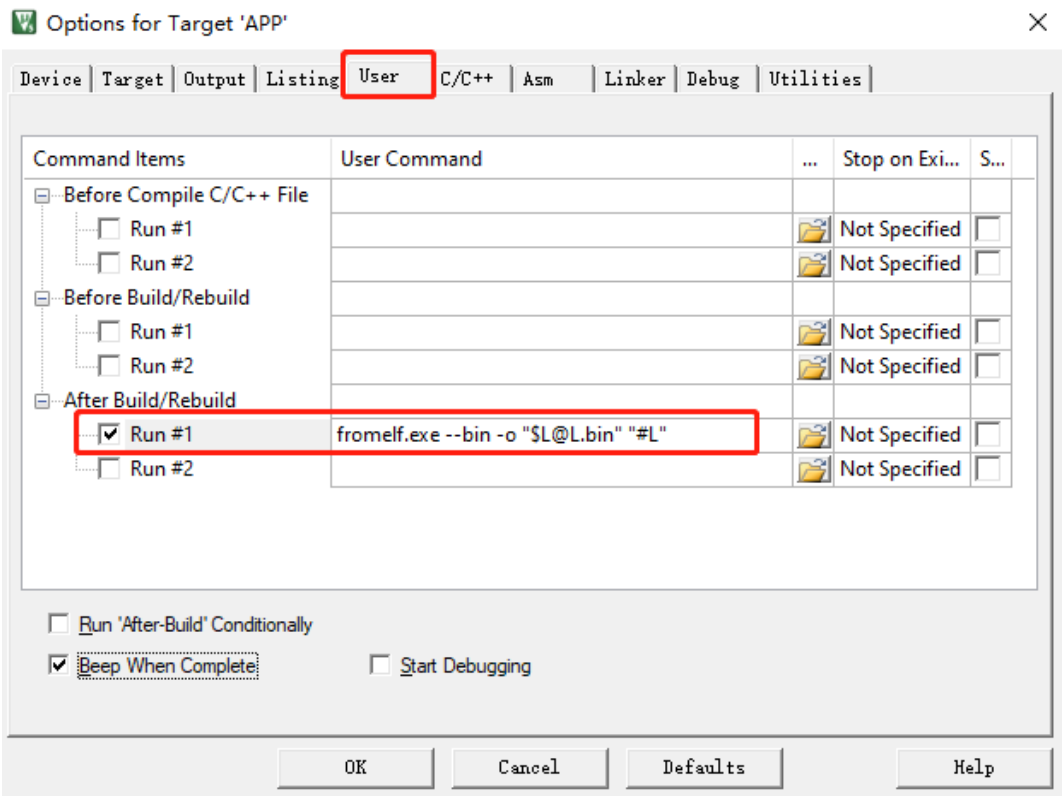

图 3-4 设置生成 bin 文件选项

4 关键代码详解

以下代码，是程序从 Bootloader 跳转到 APP 程序的关键。ApplicationAddress 是应用程序的起始地址，也是中断向量的偏移地址，可根据 APP 的起始地址修改。

1、if (((*(__IO uint32_t*)ApplicationAddress) & 0x2FFFE000) == 0x20000000)

APP 程序的起始地址存储的就是栈顶地址，查看栈顶地址是否在 SRAM 地址范围内来判断是否下载了 APP 程序，SRAM 地址范围可以通过查阅相应 MCU 的数据手册，以 SYM32L083 为例，SRAM 地址范围是 0x2000 0000～0x2000 5FFF。栈顶地址与 0x2FFFE000 相与后为 0x20000000，在 SRAM 范围内，认为已经下载了 APP 程序，可以继续执行跳转任务。

2、JumpAddress = *(__IO uint32_t*) (ApplicationAddress + 4); 和 Jump_To_Application = (pFunction) JumpAddress;

ApplicationAddress + 4 位置存储的 Reset_Handler 向量，该向量为 Reset_Handler 处理函数的入口地址。由于已经将 pFunction 重定义为 void 类型的函数指针，所以下一句将 Jump_To_Application 指针指向 Reset_Handler 函数的入口地址。

3、__set_MSP(*(__IO uint32_t*) ApplicationAddress);

执行 APP 程序的第一条指令，将主堆栈指针设置为 APP 起始地址，在运行 APP 程序之前先初始化好 MSP，因为可能 Reset_Handler 的第一条指令还没来得及执行，就发生了 NMI 或者其他 fault，此时就需要 MSP 来提供堆栈。

4、Jump_To_Application();

执行 Jump_To_Application 指针指向的函数，即 Reset_Handler 函数，在 Reset_Handler 函数中执行完__main 函数后，将自动跳转到 main()函数，也就是 APP 主程序部分。

5 IAP 示例

以 IAP Demo 工程为例，连接好 SYM32 和 PC 端接线。

Step1.上电或复位时检测到板载按键处于按下状态，进入 IAP 模式，执行 Step2，否则执行 Step3。

Step2.接收到 PC 端上位机软件按下“下载程序”键，执行通过 Ymodem 传输协议，更新应用程序。

接收到 PC 端上位机软件按下“上传程序”键，执行通过 Ymodem 传输协议，上传应用程序。

接收到 PC 端上位机软件按下“执行应用程序”键，执行 Step3。

Step3.执行跳转到应用程序。

下面详细介绍一下 Ymodem 协议。

YModem 协议可达每帧数据传输 1024 字节，使用冗余循环校验，是一个可靠、高效的文件数据传输协议。

表 5-1 字符说明

字符说明	
字符	十六进制
SOH	0x01
STX	0x02
ACK	0x06
NACK	0x15
EOT	0x04
C	0x43

表 5-2 帧格式说明

帧格式					
帧头 （1字节）	发送序号 （1字节）	发送序号补码 （1字节）	数据区 （128/1024字节）	CRCH (1字节)	CRCL (1字节)
SOH/STX	index	~index	…	…	…

SOH/STX：发送 128 字节/1024 字节

CRCH：16 位 CRC 高字节

CRCL：16 位 CRC 低字节

起始帧（133 字节）

表 5-3 起始帧

起始帧					
SOH	00	FF	filename[]+filesize[]+NULL[]	CRCH	CRCL

			（共128字节）		

filename[]:存放文件名（存放文件名的十六进制），文件名后一位存放 0x00 作为结束标志。

filesize[]:存放文件的大小，后加 0x00 作为结束标志。

NULL[]:表示剩下的字节填充 0x00

数据帧（133 字节/1029 字节）

表 5-4 数据帧

数据帧					
SOH/STX	[帧序号]	[帧序号反码]	data[0]+data[1]+data[2]+ … .. （128字节/1024字节）	CRCH	CRCL

data：传输的数据，如果不足 128 字节，其余全部填充 1A。

结束帧（133 字节）

表 5-5 结束帧

结束帧					
SOH	00	FF	NULL+NULL+NULL+…..+NULL （128字节）	CRCH	CRCL

NULL：结束帧 NULL 全部填入 0x00.

PC 端和 SYM32 之间的传输协议流程如下图：

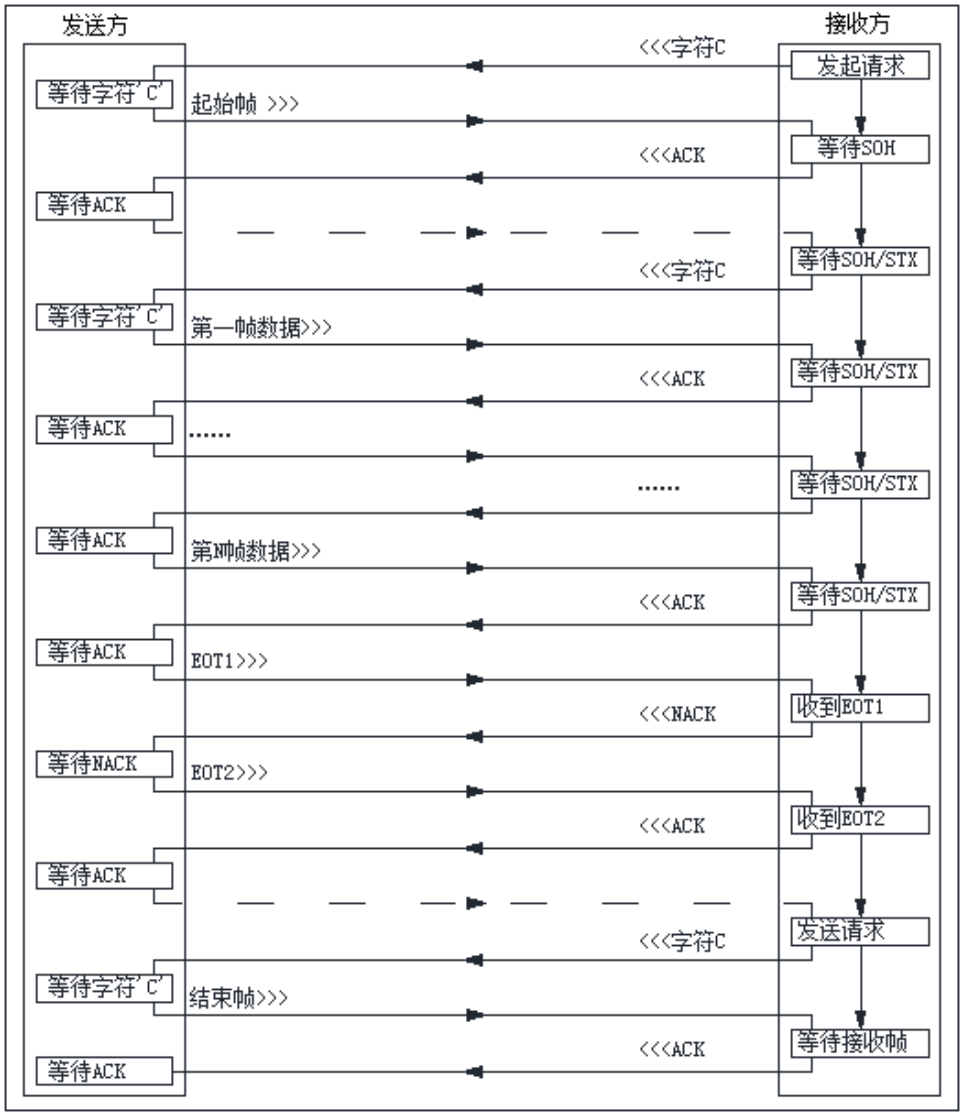

图 5-1 Ymodem 协议传输流程

流程说明：

Step1. 接收方开启传输，接收方发送一个字符‘c’，进入等待（SOH）状态，没回应，超时退出；

Step2. 发送方开始时处于等待状态，等待字符‘c’。若发送方收到‘c’后，发送第一帧（起始帧）发送完毕后，进入等待(ACK)状态；

Step3. 接收方收到第一帧数据包后，进行 CRC 校验，校验通过，则发送 ACK；

Step4. 发送方收到 ACK，又进入等待“文件传输开启”信号，进入等待‘c’状态；

Step5. 接收方又发出一个字符‘c’，开始准备接收文键。进入等待 SOH 或 STX 状态；

Step6. 发送方收到字符‘c’后，开始发送数据帧；

Step7. 接收方收到数据后，发送一个 ACK，若要传输数据包含多帧数据，继续 ACK 应答，直到所有数据传输完毕；

Step8. 数据传输完毕后，发送方发送 EOT，第一次以 NACK 应答，进行二次确认。发送方收到 NAK 后，重发 EOT，接收方第二次收到结束符，就发送 ACK 应答。最后接收方在发送一个字符‘c’开启另一次传输，发送方在没有第二个文件要传输的情况下，发送结束帧，正式结束数据传输。

6 PC 端上位机操作示例——以 超级终端 为例

6.1 新建连接

Step1.打开软件，点击“文件-新建连接”，如图 6-1 所示；

Step2.不安装调制解调器，如图 6-2 所示；

Step3.输入名称，点击“确定”，如图 6-3 所示；

Step4.输入区号，选择连接方式为端口，点击“确定”，如图 6-4 所示；

Step5.配置端口参数，与 Bootloader 程序工程配置保持一致，点击“确定”，如图 6-5 所示。

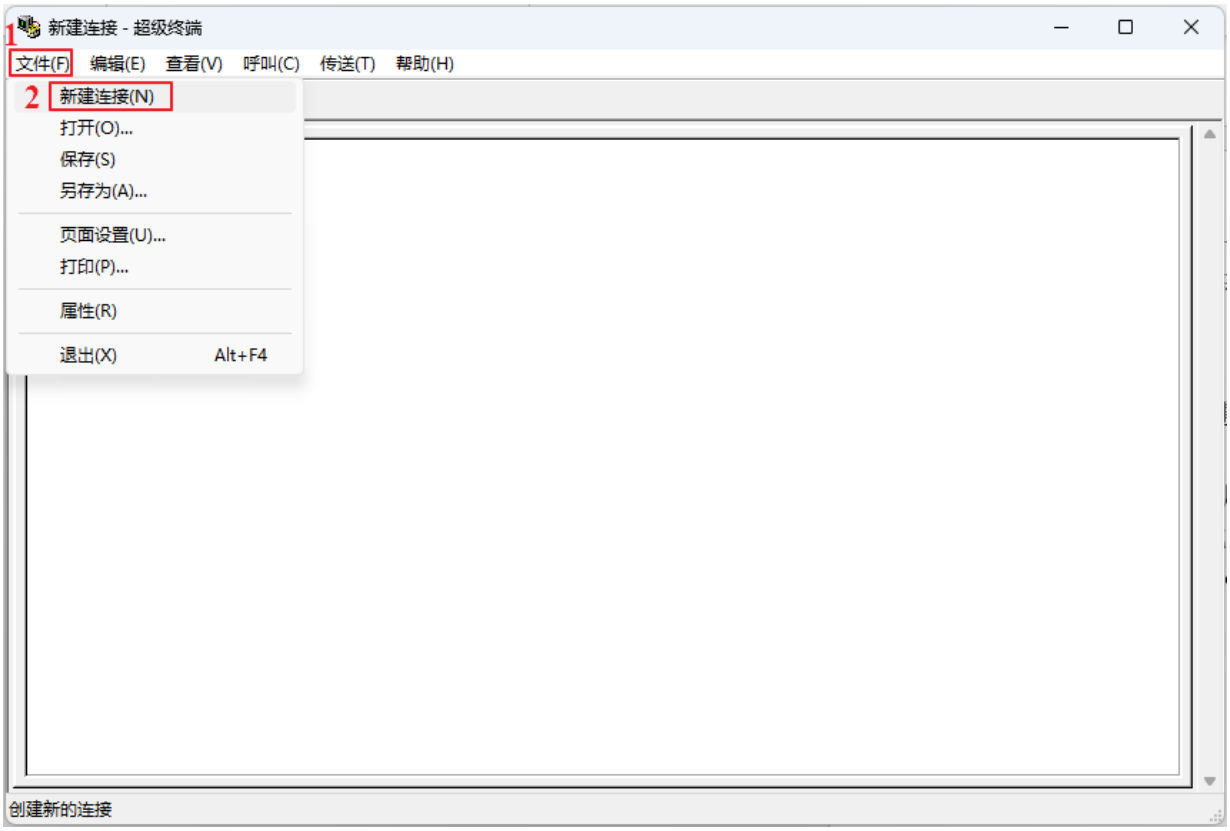

图 6-1 新建连接

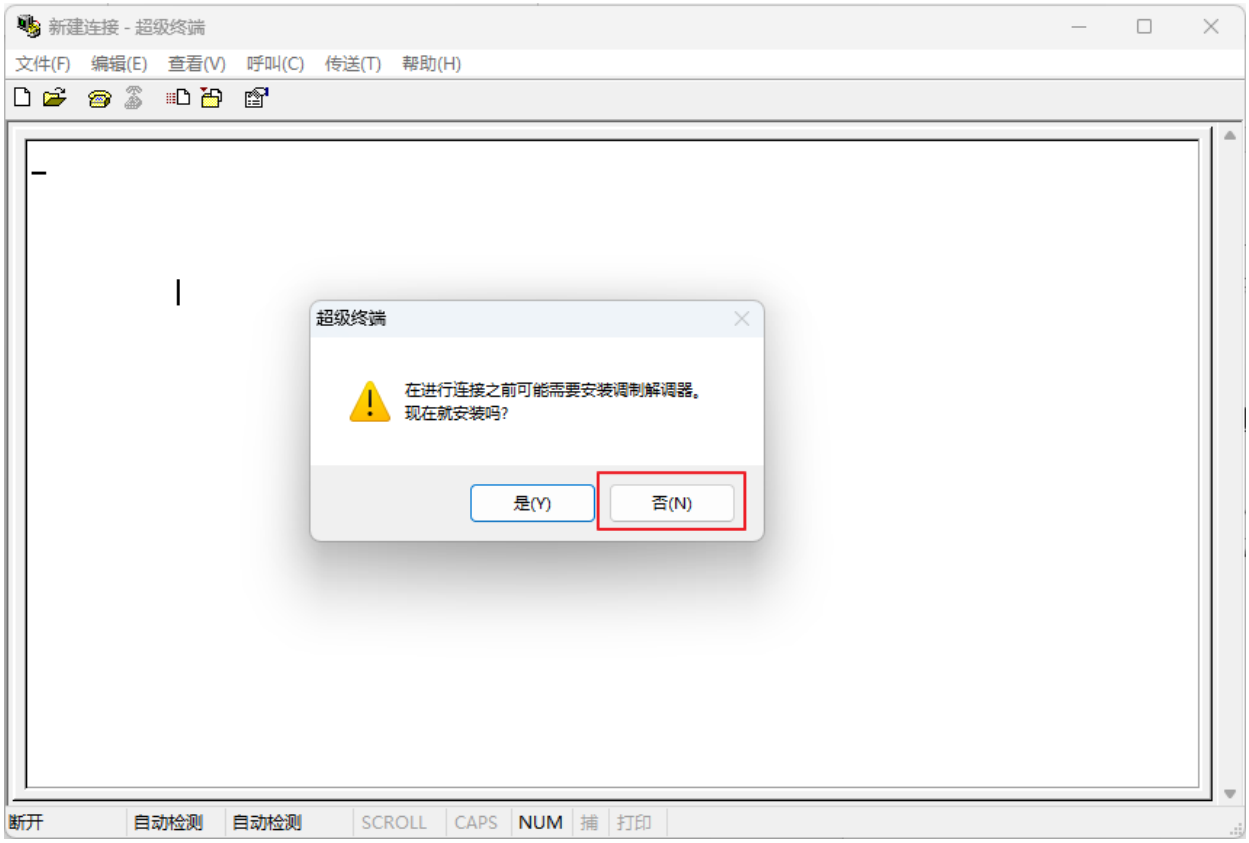

图 6-2 无需安装解调器

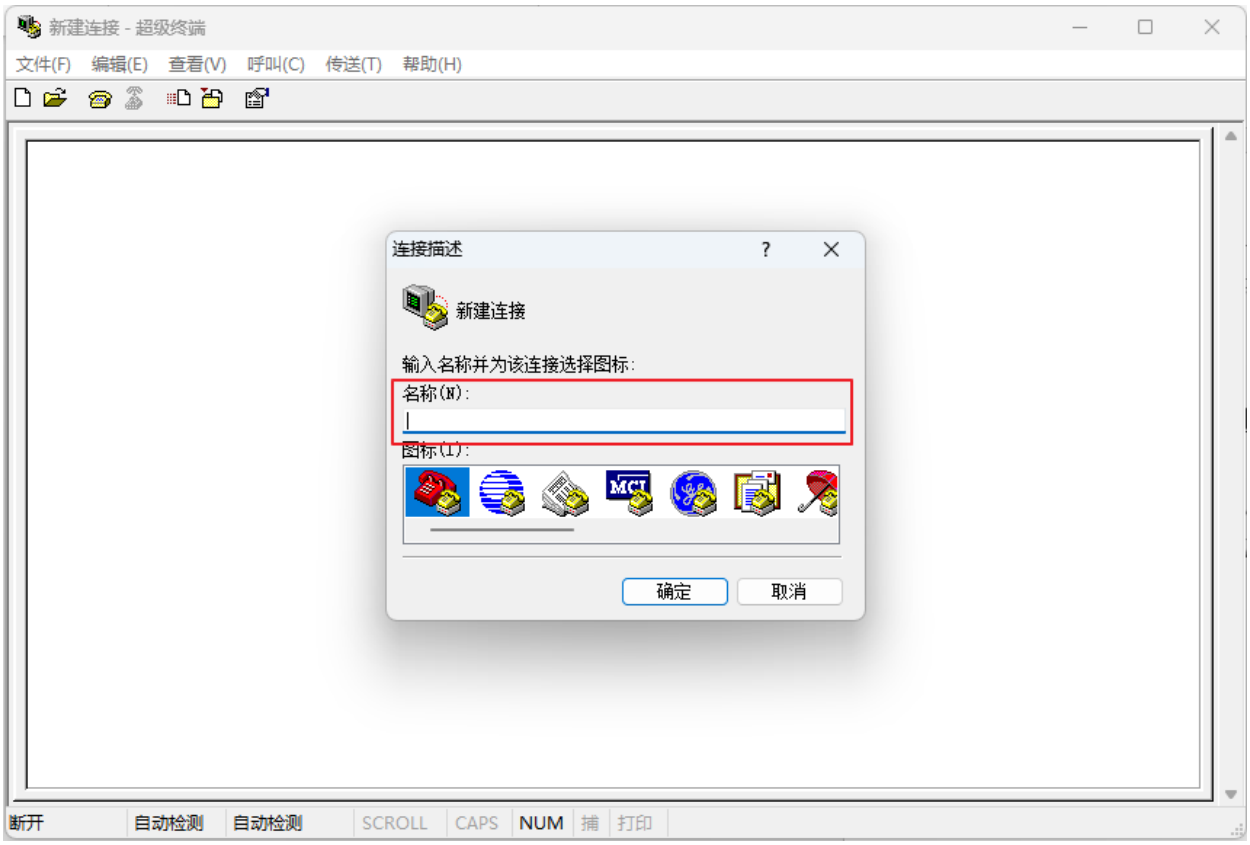

图 6-3 输入名称

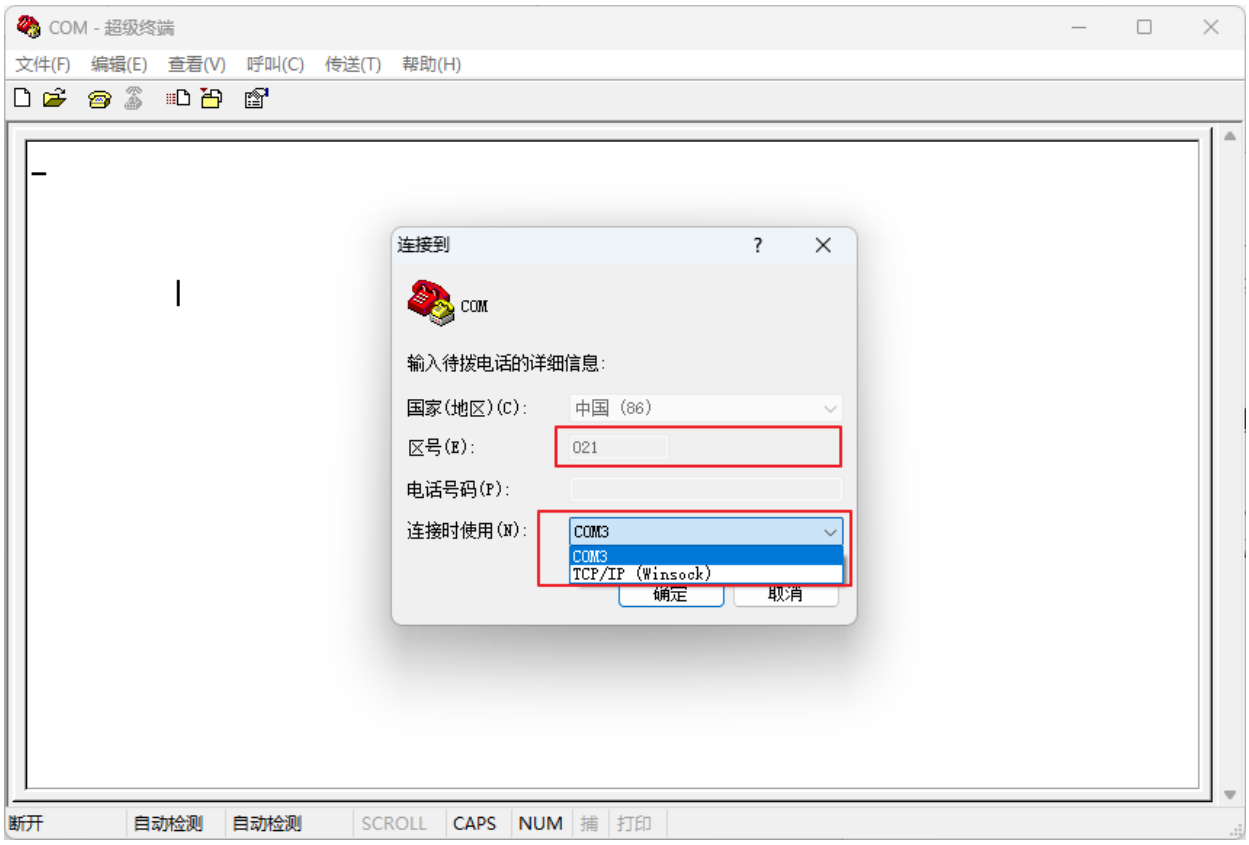

图 6-4 输入区号并选择连接方式

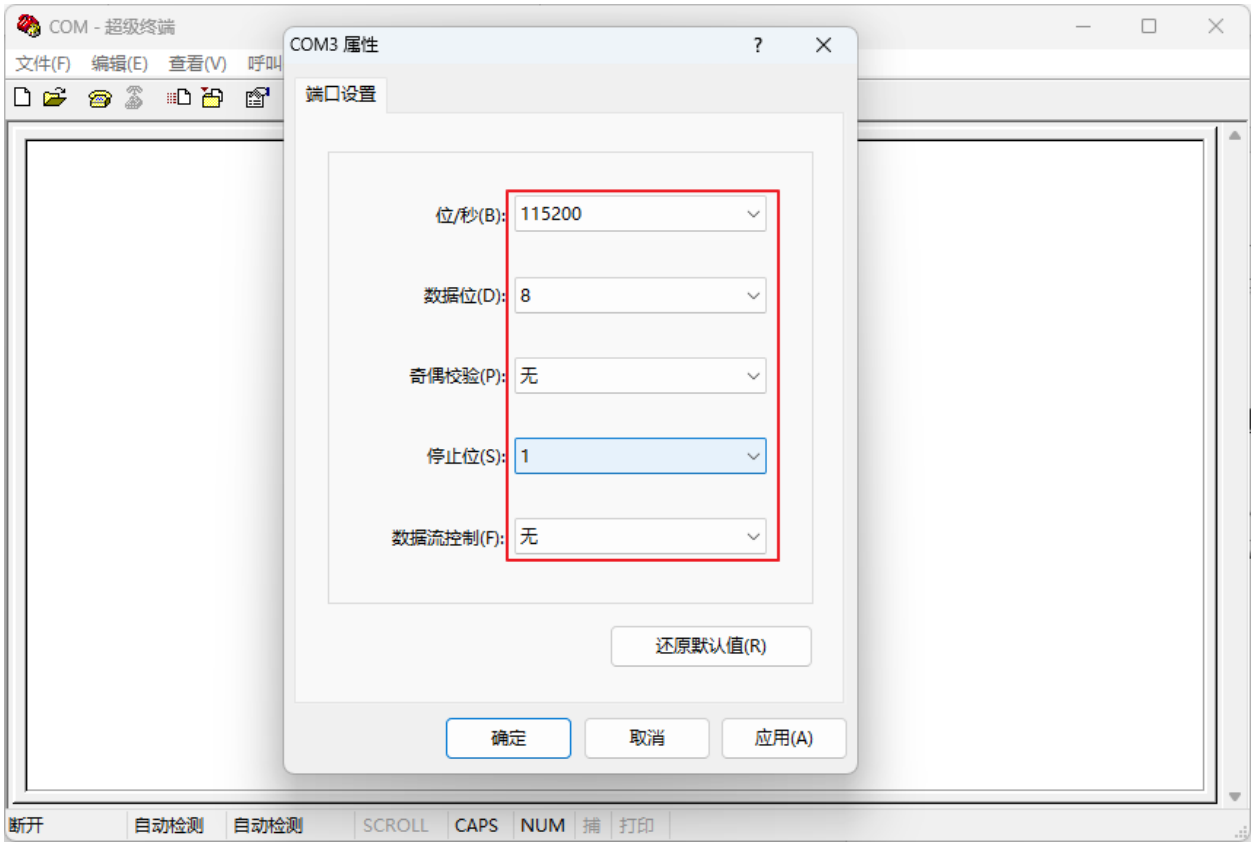

图 6-5 端口设置

6.2 端口的连接与断开

在断开状态下，点击“呼叫-呼叫”进行连接，如图 6-6 所示；

在连接状态下，点击“呼叫-断开”进行断开，如图 6-7 所示。

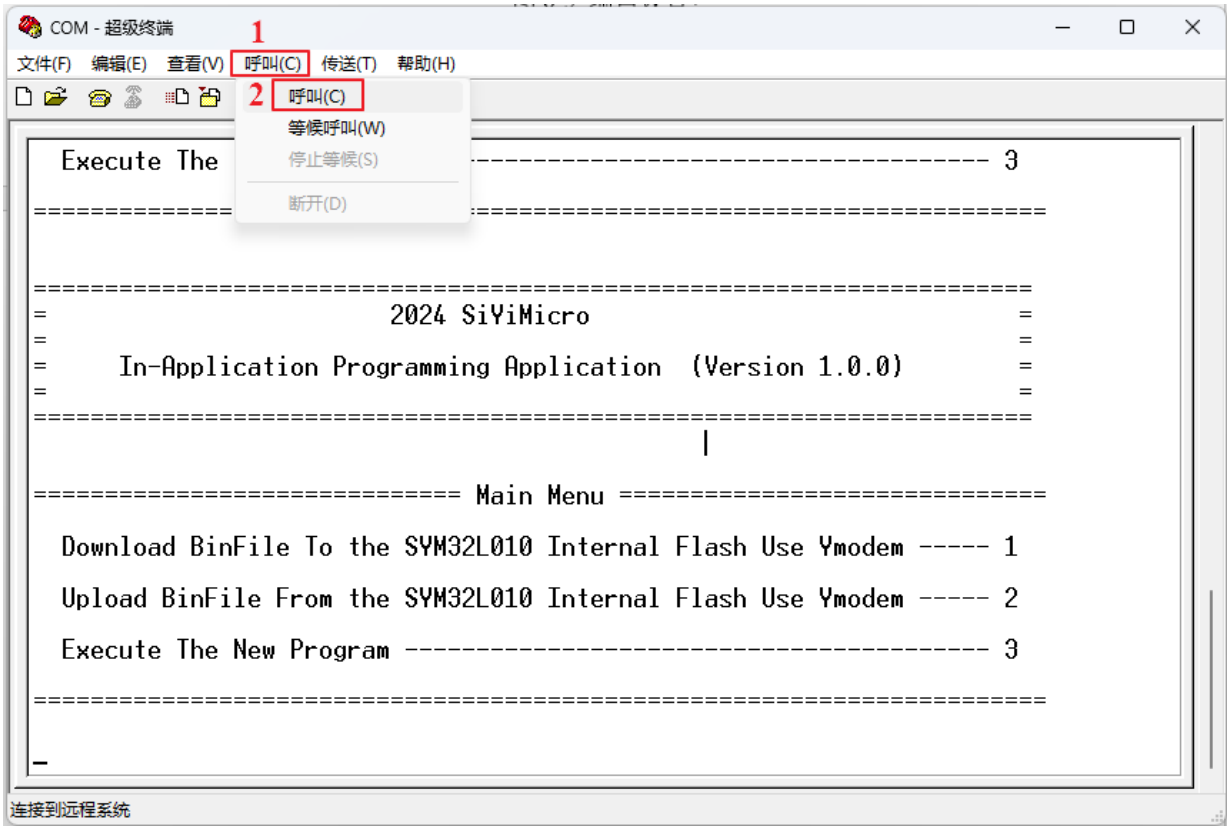

图 6-6 端口的连接

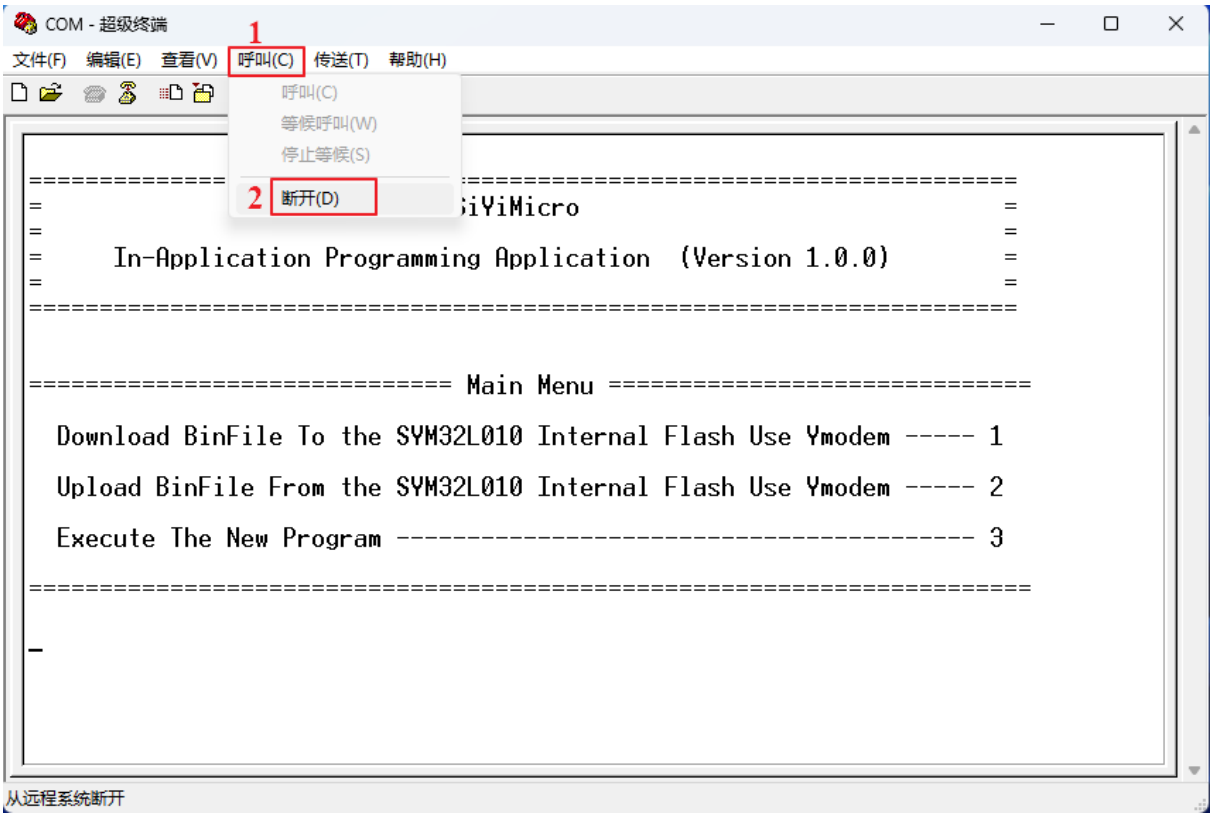

图 6-7 端口的断开

6.3 发送字符

通过键盘按键发送字符。注意：已发送的字符不会显示在桌面。

6.4 发送文件

Step1.保持板载按键按下的状态进行一次复位，然后松开板载按键，使下位机进入 IAP 模式，如图 6-8 所示；

Step1.根据提示信息，发送字符“1”使下位机处于接收状态，如图 6-9 所示；

Step2.点击“传送-发送文件”，如图 6-10 所示；

Step3.选择发送的文件（本例中选择 APP 应用程序工程编译输出的文件），协议选择“Ymodem”，点击“发送”，如图 6-11 所示；

Step4.等待发送完成，如图 6-12 所示。

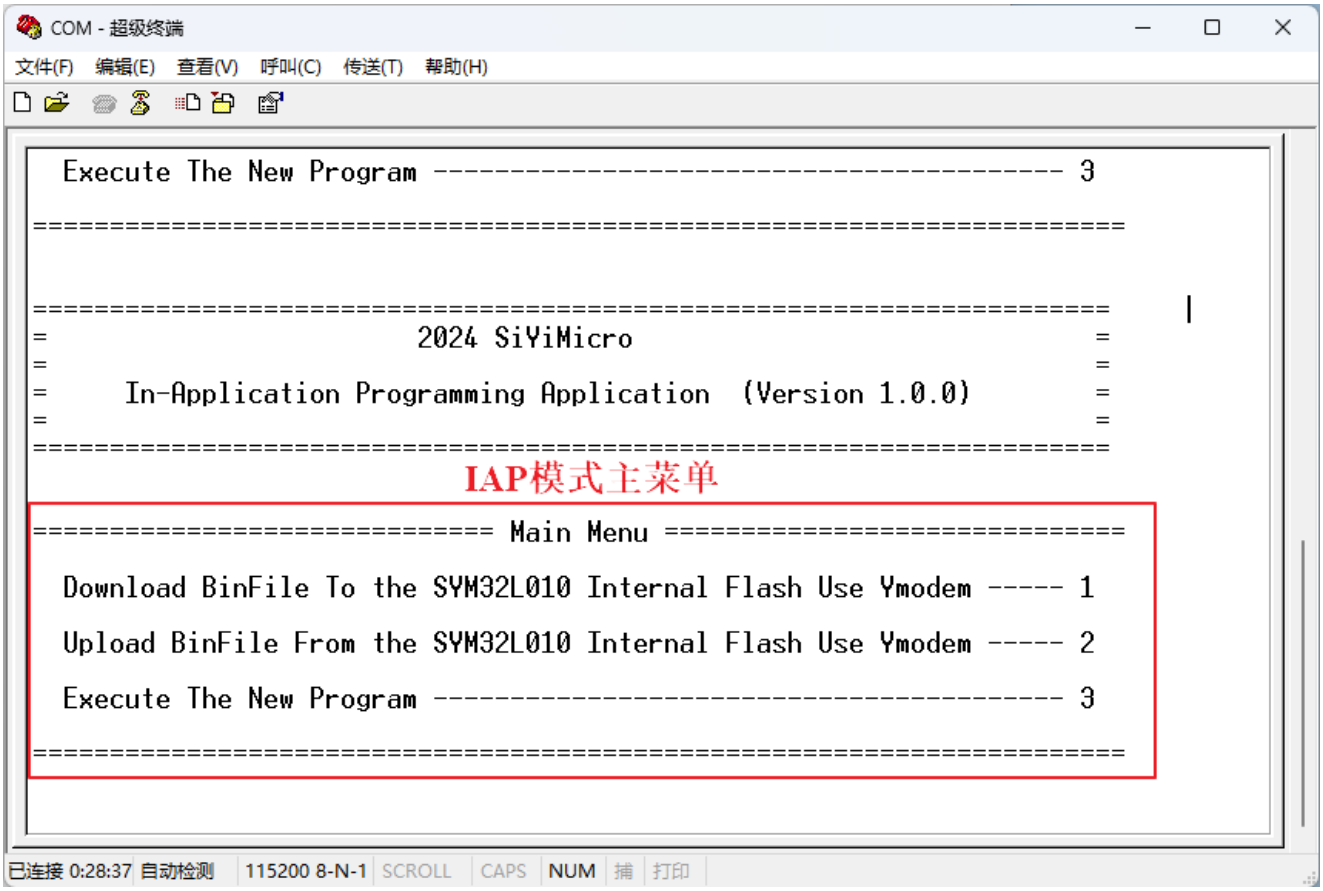

图 6-8 下位机处于 IAP 模式时的提示信息

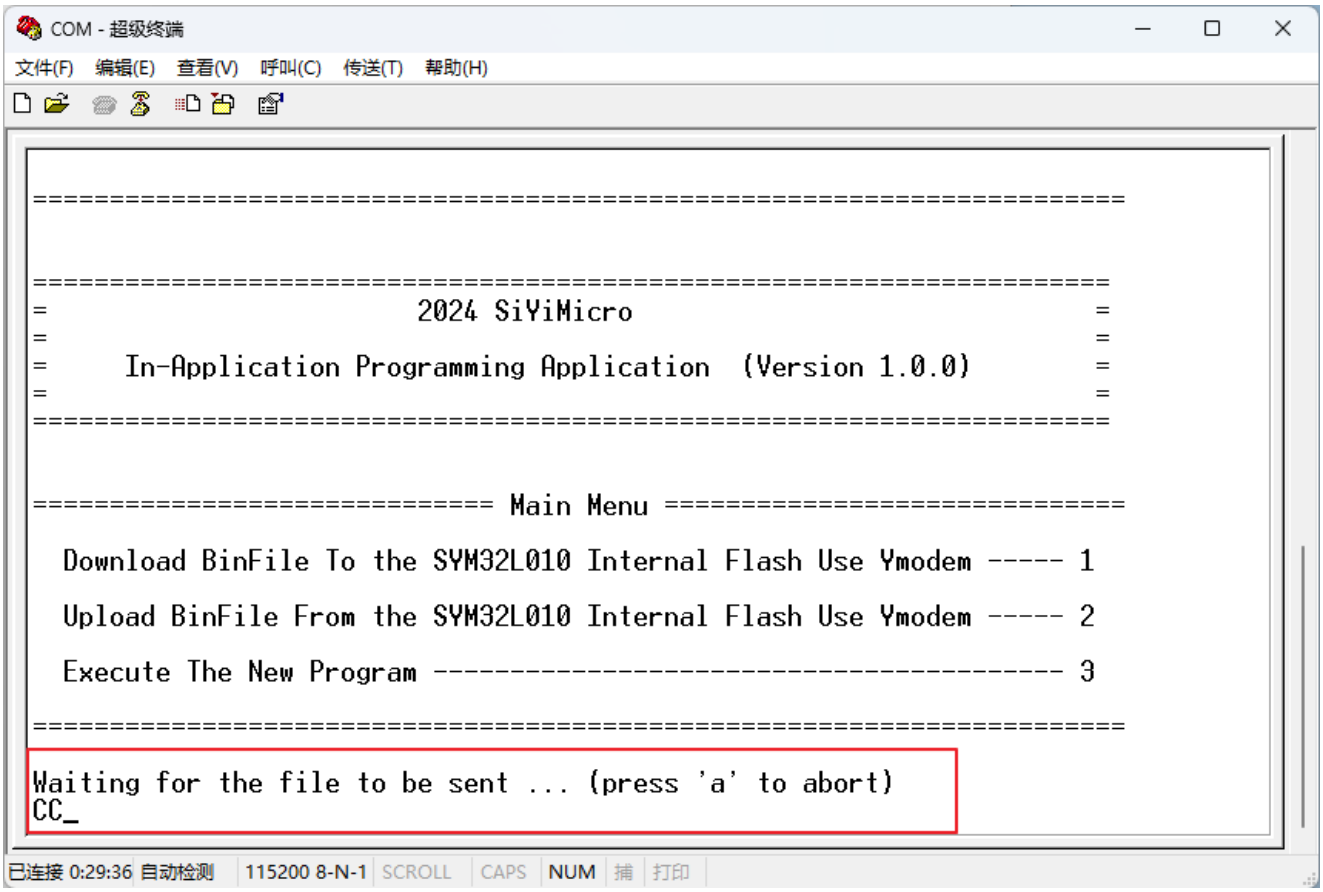

图 6-9 下位机处于接收状态时的提示信息

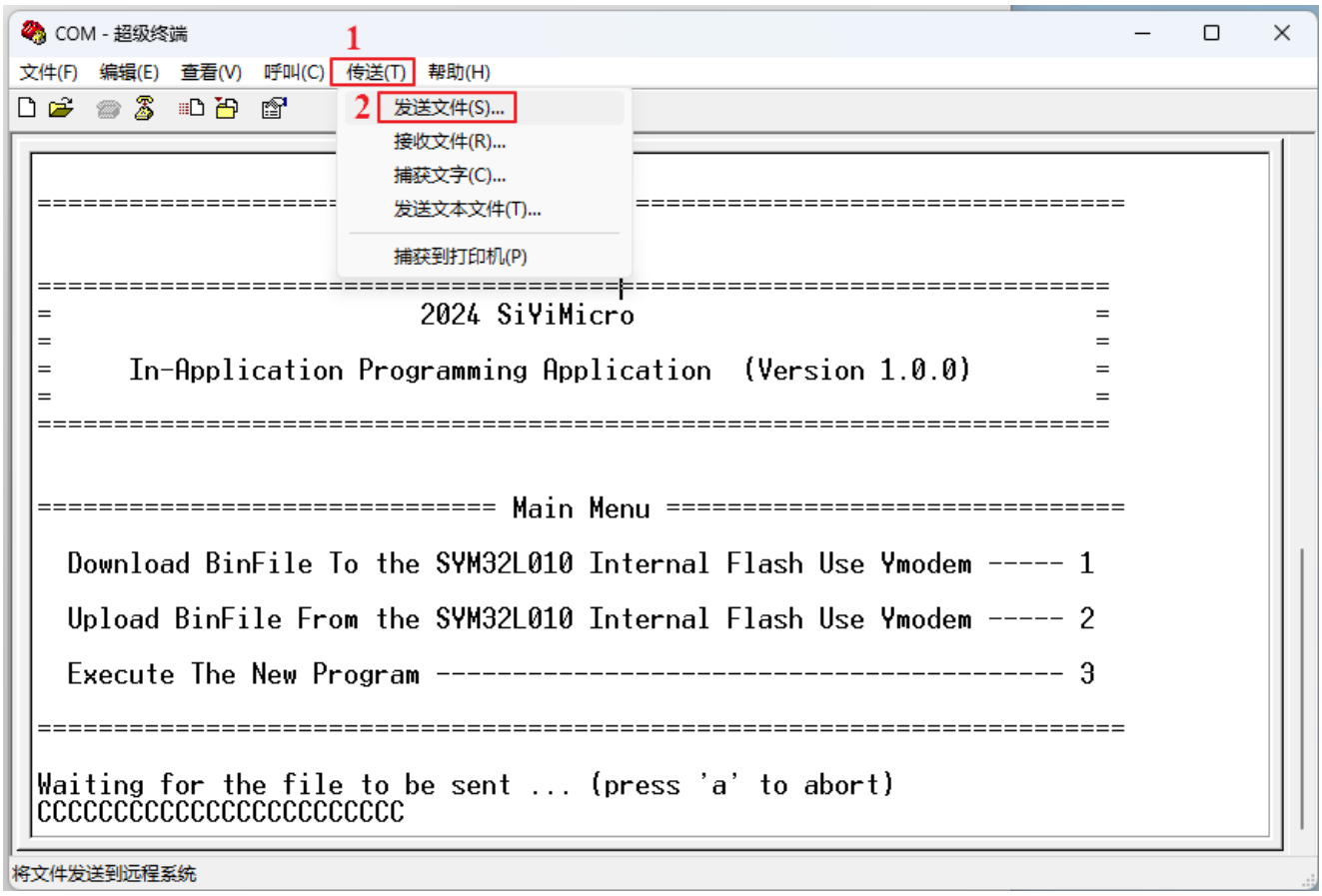

图 6-10 发送文件

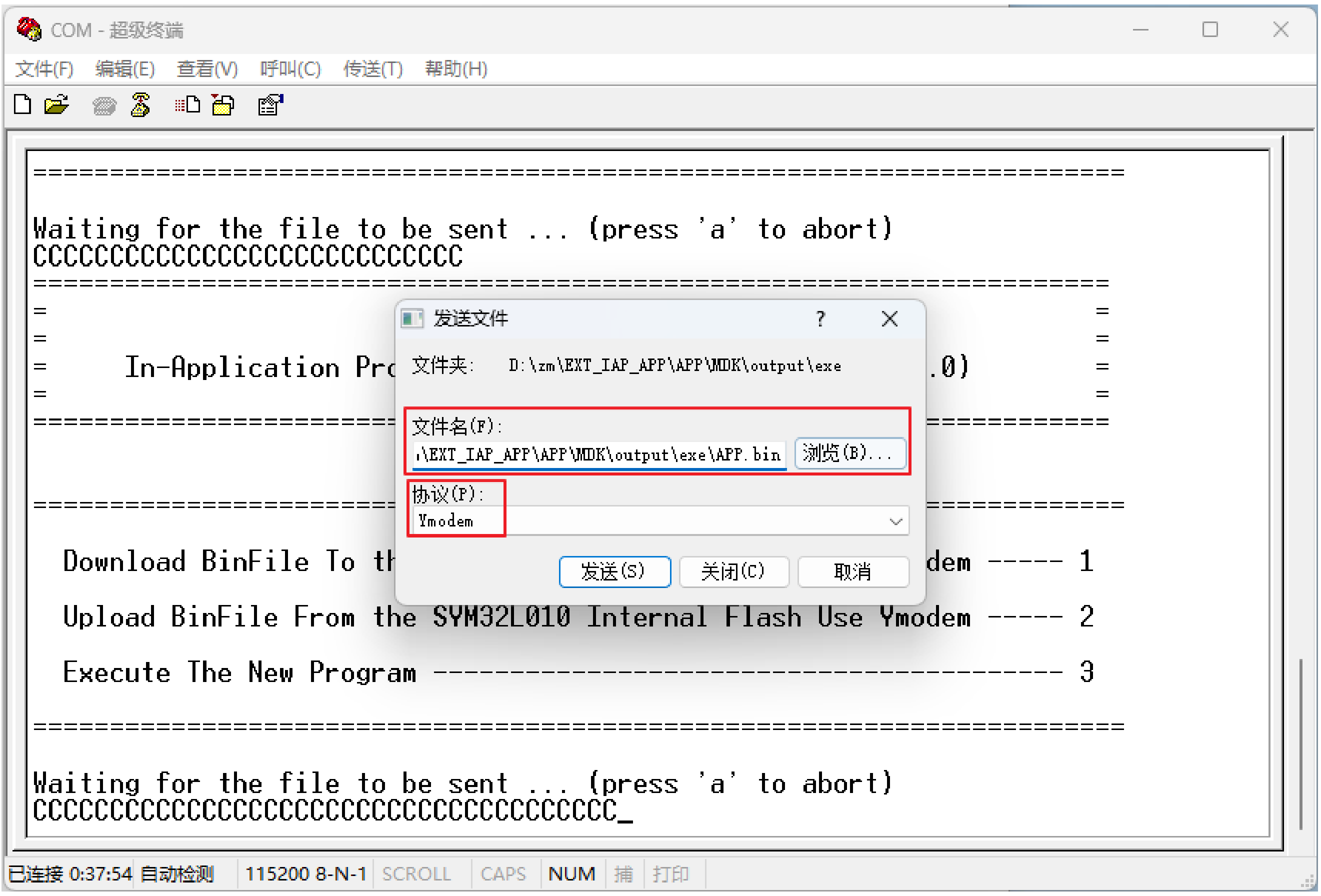

图 6-11 选择发送的文件和协议

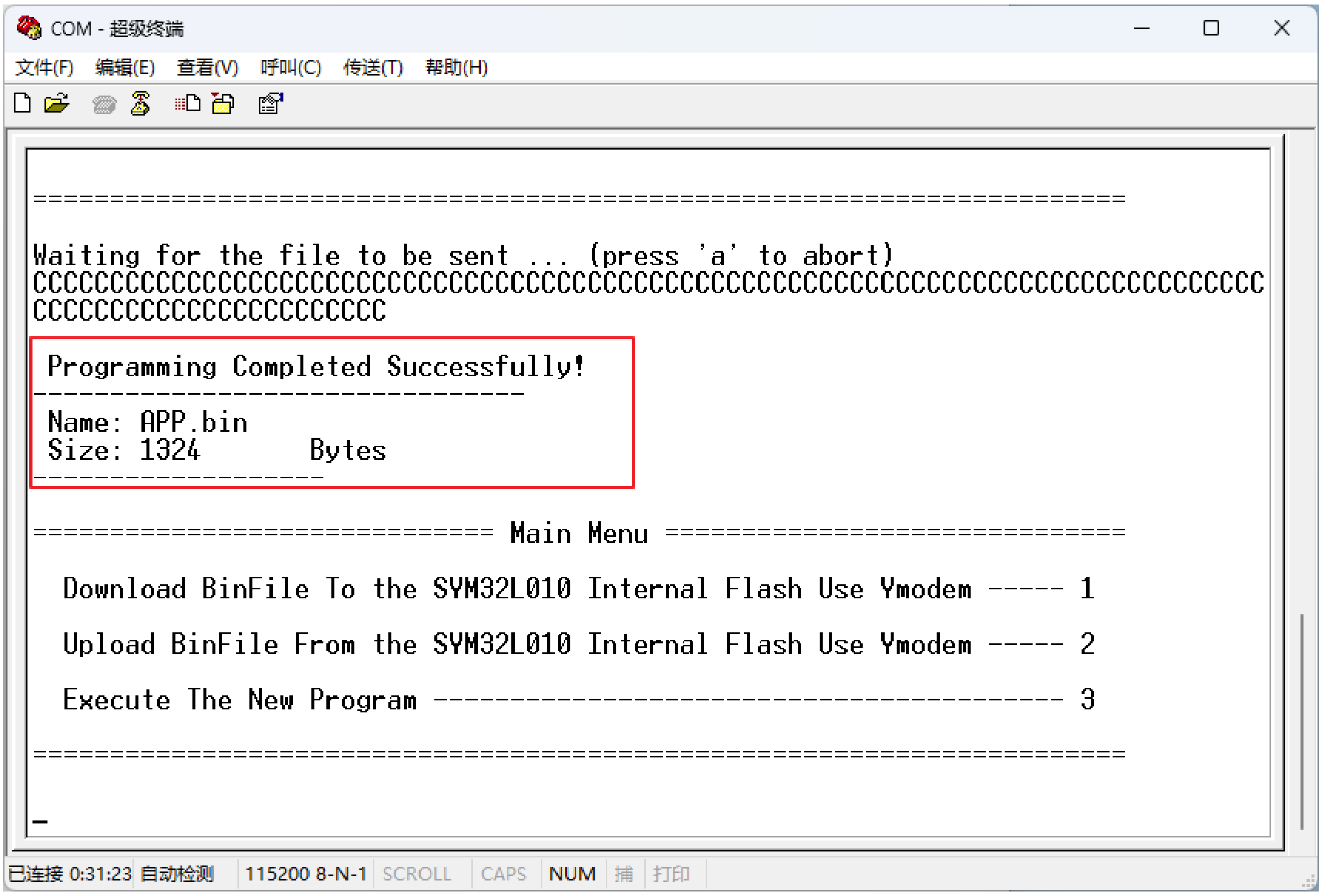

图 6-12 发送完成的提示信息

6.5 接收文件

Step1.在保持板载按键按下的状态时进行一次复位，然后松开板载按键，使下位机进入 IAP 模式，如图 6-8 所示；

Step1.根据提示信息，发送字符“2”使下位机处于发送状态，如图 6-13 所示；

Step2.点击“传送-接收文件”，如图 6-14 所示；

Step3.选择接收文件的保存位置，接收协议选择“Ymodem”，点击“接收”，如图 6-15 所示；

Step4.等待接收结束，如图 6-16 所示。

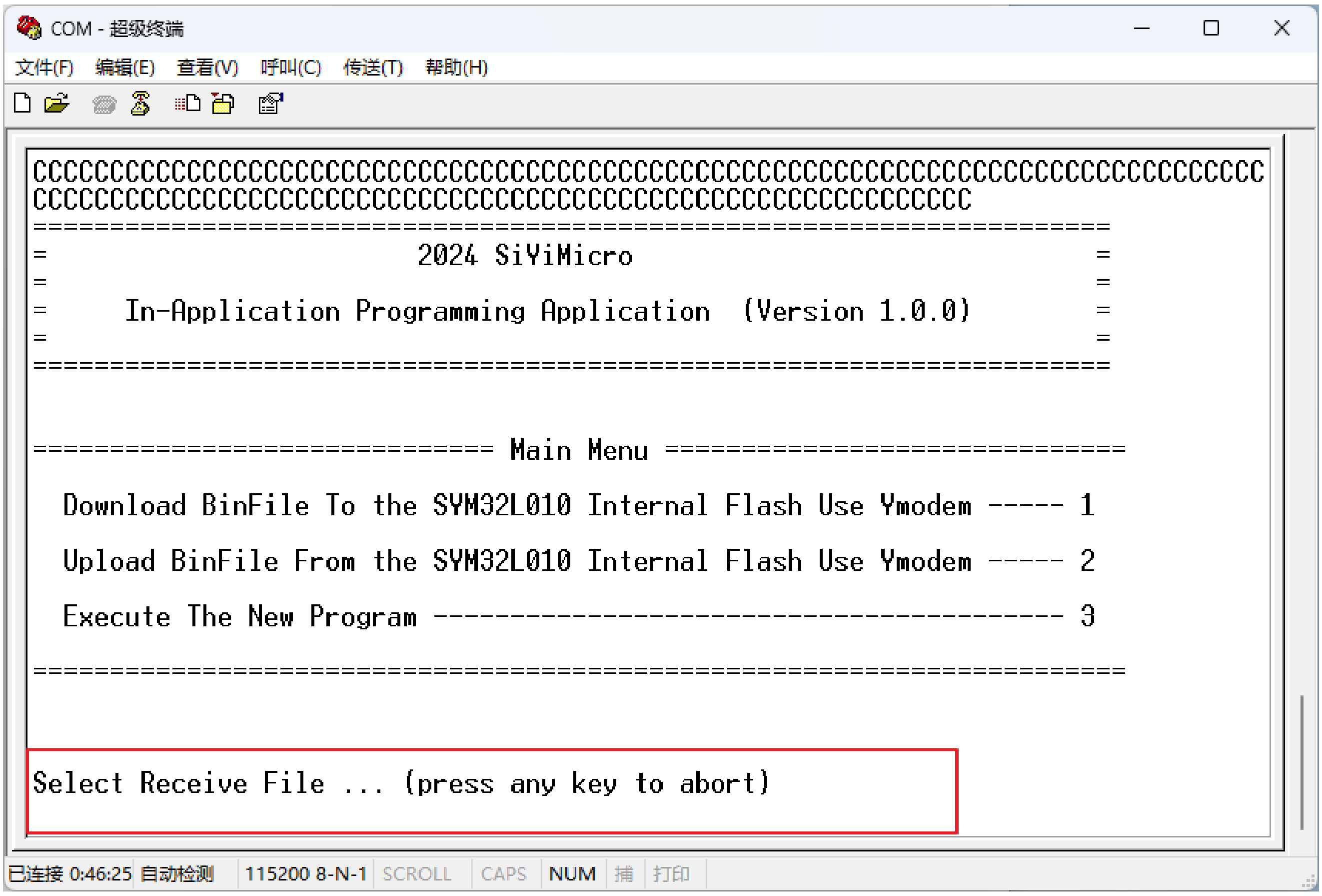

图 6-13 下位机处于发送状态时的提示信息

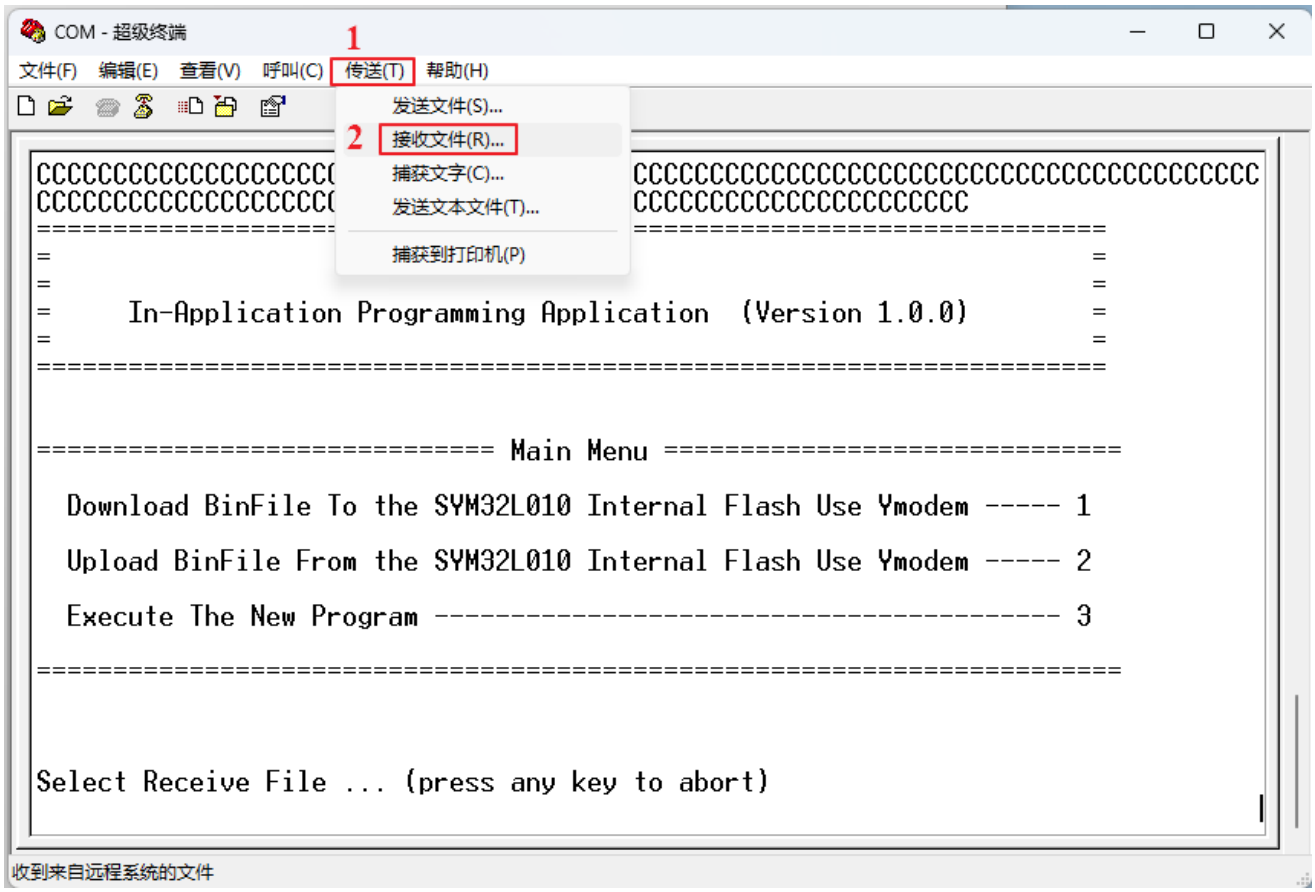

图 6-14 接收文件

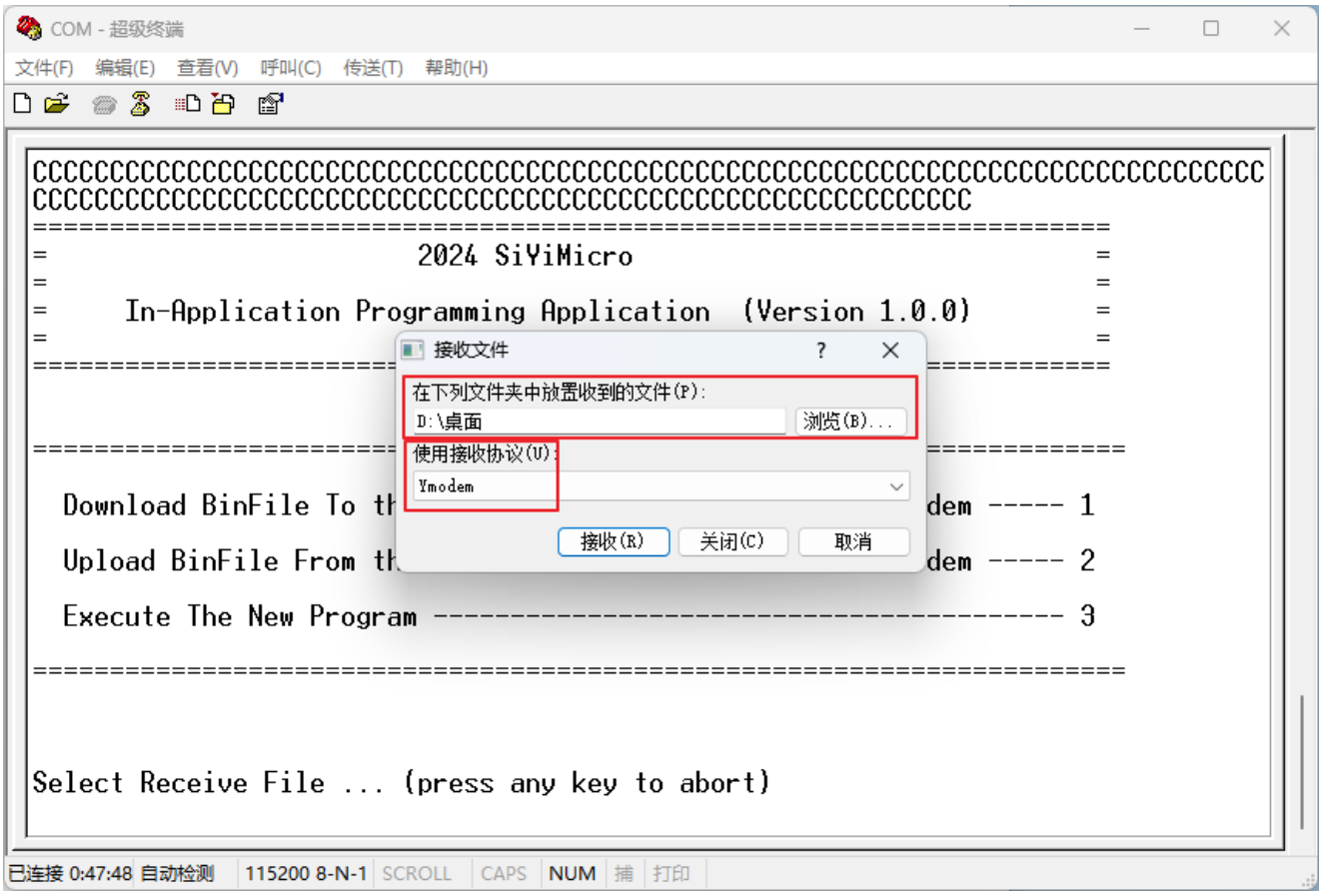

图 6-15 选择接收文件存放位置和接受协议

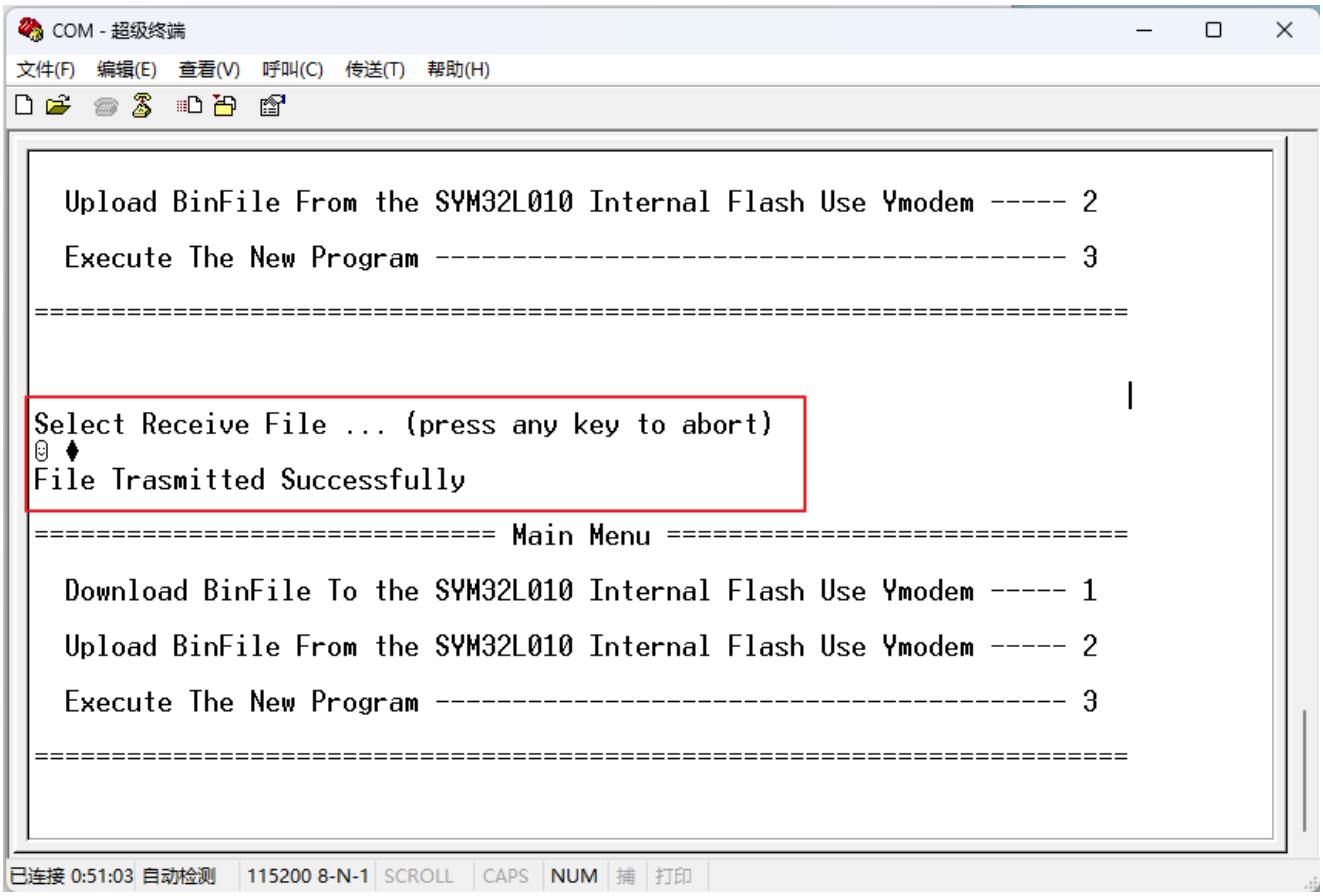

图 6-16 上位机接收完成的提示信息

版本记录

版本	修订日期	修订说明
Rev1.0	2022-05-20	初始版本
Rev1.1	2024-07-08	新增上位机的介绍与操作