

第4章 如何用串口下载程序

秉火 STM32F103VET6—“指南者”自带串口下载电路，配合上位机可实现一键 ISP 下载，不需要修改开发板上的 BOOT 设置。与仿真器 Fire-Debugger 相比，ISP 只能下载程序，不能在线调试且下载速度慢。

4.1 安装 USB 转串口驱动

秉火的 STM32 开发板用的 USB 转串口的驱动芯片是 CH340，要使用串口得先在电脑中安装 USB 转串口驱动—CH340 版本。驱动可在网上搜索下载或者使用我们光盘里面提供的。WIN7 用户请用管理员身份安装。如果不能安装成功，请先百度查找原因自行解决。

如果 USB 转串口驱动安装成功，USB 线跟板子连接没有问题，在计算机->管理->设备管理器->端口中可识别到串口。

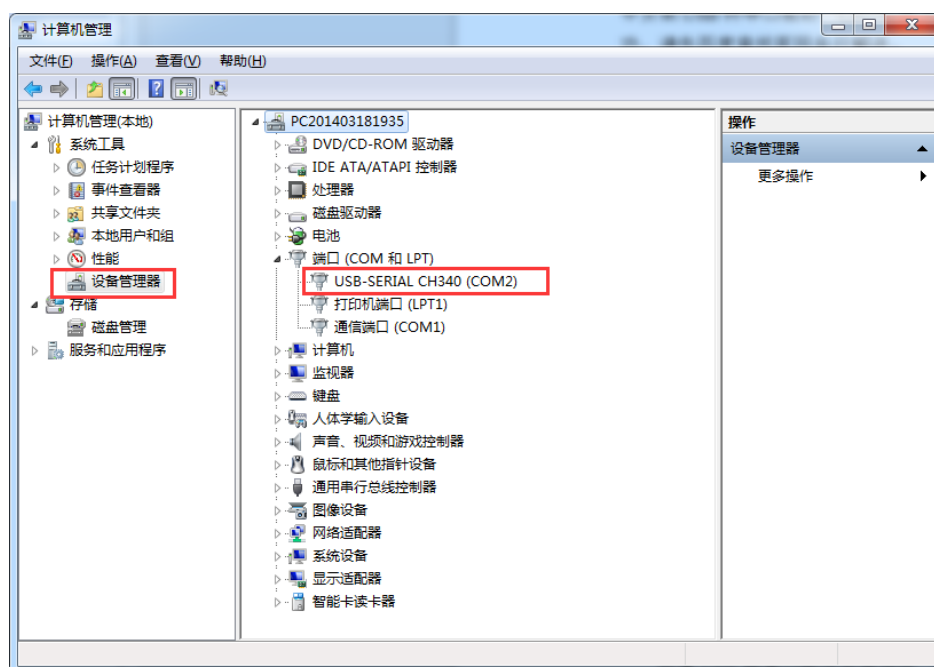


图 4-1 USB 转串口驱动安装成功

如果识别不了串口，请检查 USB 线是否完好，可换一根 USB 线试试。

4.2 硬件连接

用 USB 线连接电脑和开发板的 USB 转串口接口：USB TO UART，给开发板上电。

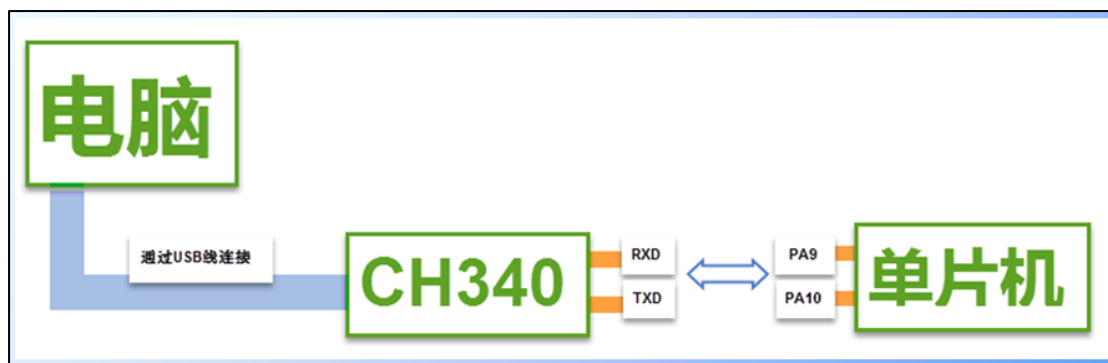


图 4-2 串口下载连接图

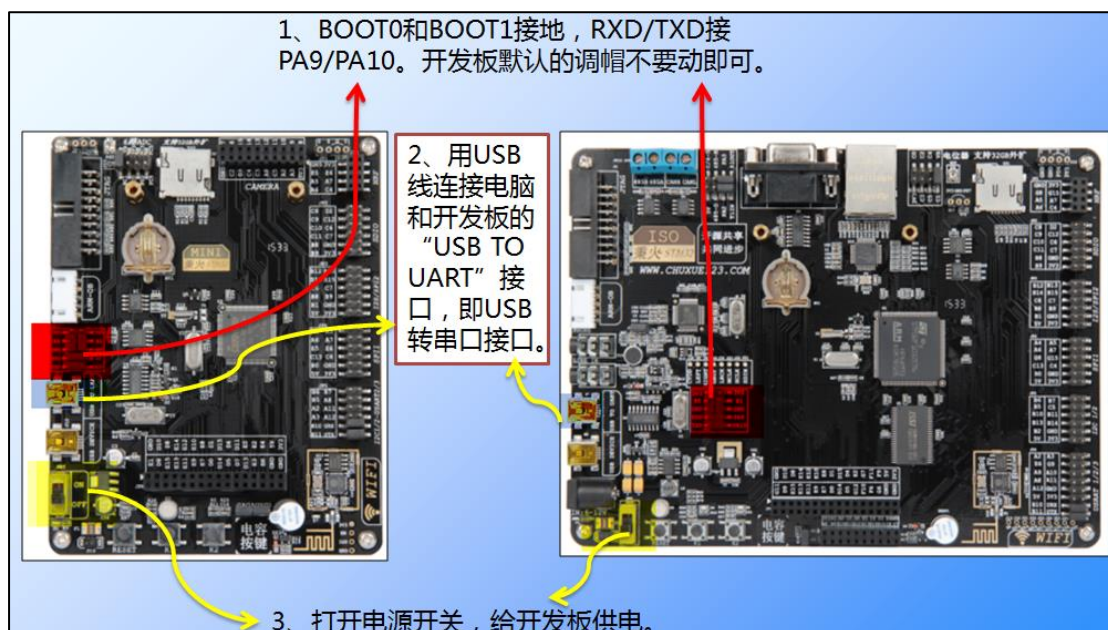


图 4-3 指南者/霸道串口下载接线图

4.3 开始下载

打开 mcuisp 软件，配置如下：①搜索串口，设置波特率 115200（尽量不要设置的太高）、②选择要下载的 HEX 文件、③校验、编程后执行、④DTR 低电平复位，RTS 高电平进入 bootloader、⑤开始编程。如果出现一直连接的情况，按一下开发板的复位键即可。

零死角玩转 STM32F103—指南者

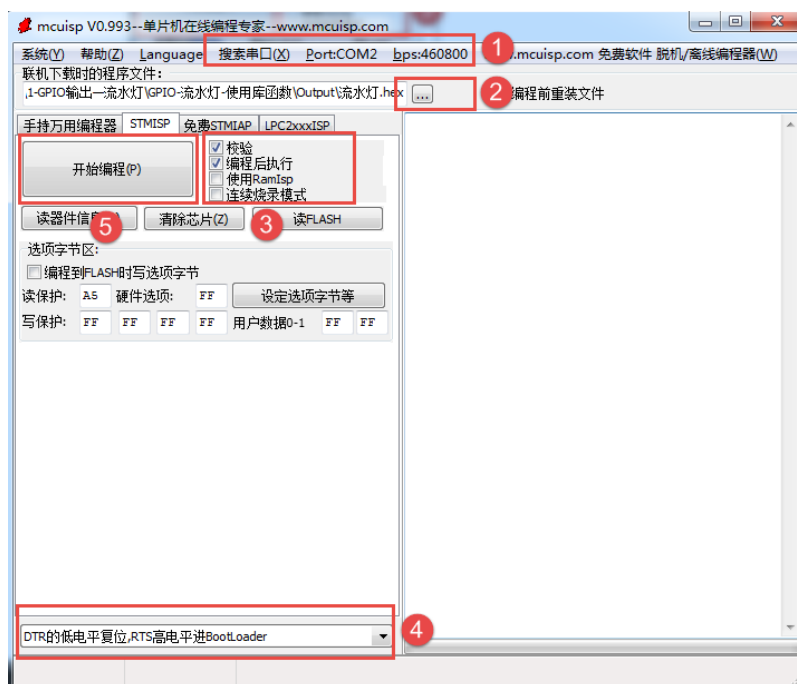


图 4-4 ISP 下载配置

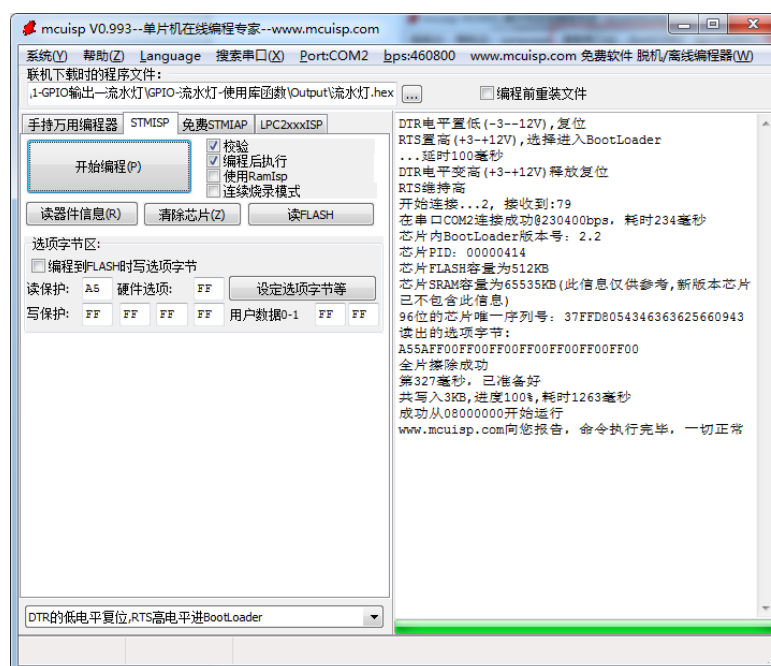


图 4-5 ISP 下载成功

4.4 ISP 一键下载原理分析

4.4.1 ISP 简介

ISP (In-System Programming) 在系统可编程，指电路板上的空白器件可以编程写入最终用户代码，而不需要从电路板上取下器件，已经编程的器件也可以用 ISP 方式擦除或再编程。

ISP 的时候需要用到(bootloader)自举程序，自举程序存储在 STM32 器件的内部自举 ROM 存储器（系统存储器）中。其主要任务是通过一种可用的串行外设（USART、CAN、USB、I2C 等）将应用程序下载到内部 Flash 中。每种串行接口都定义了相应的通信协议，其中包含兼容的命令集和序列。

4.4.2 ISP 普通下载

现在我们针对 USART1 的 ISP 进行分析，通常的 ISP 的步骤如下：

1. 电脑通过 USB 转串口线连接 STM32 的 USART1，并打开电脑端的上位机；
2. 设置跳线保持 BOOT0 为高电平，BOOT1 为低电平；
3. 复位单片机使其进入 bootloader 模式，通过上位机下载程序；
4. 下载完毕，设置跳线保持 BOOT0 为低电平，BOOT1 为低电平；
5. 复位单片机即可启动用户代码，正常运行。

以上步骤有个不好的地方就是下载程序需要跳线及复位操作，很繁琐。通过对 ISP 的原理认识，一键 ISP 就诞生了，它需要做的事情就是用上位机去控制 BOOT0 脚和单片机的复位脚，原理图如下：

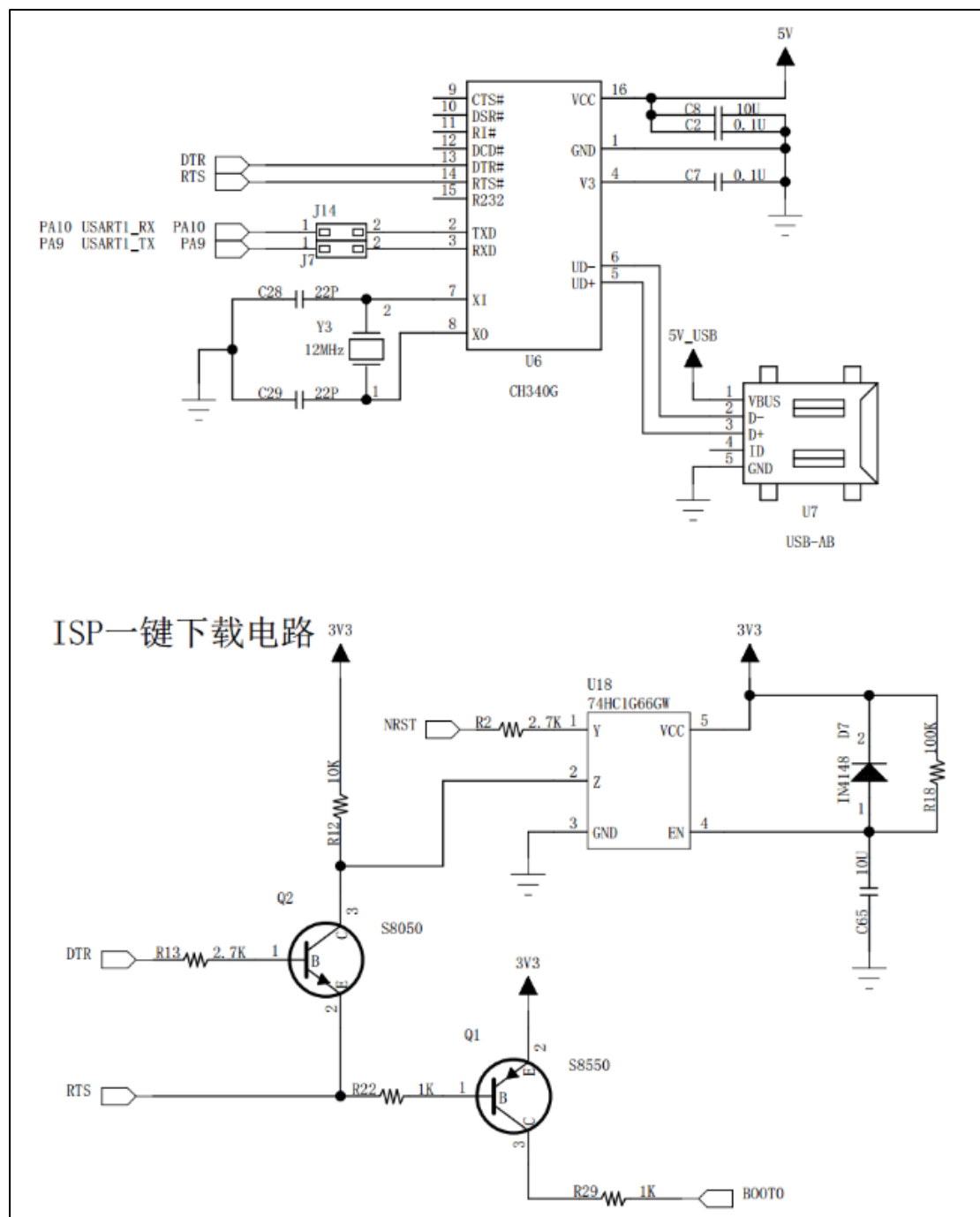


图 4-6 ISP 一键下载电路

4.4.3 BOOT 配置

在 ISP 下载电路中，我们需要配置 BOOT 引脚，有关 BOOT 引脚不同的配置会产生不同的启动方式，具体见表格 4-1 BOOT 配置。

表格 4-1 BOOT 配置

BOOT0	BOOT1	启动方式
0	X	内部 FLASH
1	0	系统存储器

1	1	内部 SRAM
---	---	---------

4.4.4 ISP 一键下载

USB 转串口估计大家都很熟悉，一般都是用到 RXD 和 TXD 这两个口，一键 ISP 电路中我们需要用 USB 转串口的芯片的 DTR 口和 RTS 口来控制单片机的 BOOT0 和 NRST，原理如下：

1. 通过上位机控制 U6(CH340G)的 RTS 脚为低电平，Q1 导通，BOOT0 的电平上拉为高电平。
2. 通过上位机控制 U6(CH340G)的 DTR 脚为高电平，由于 RTS 为低电平，Q2 导通，U8 的 2 脚为低电平，U18 为一个模拟开关，使能端由 4 脚控制，默认高电平，U18 的 1 脚和 2 脚导通，所以 NRST 为低电平系统复位。
3. 单片机进入 ISP 模式，此时可以将 DTR 脚设置为低电平，RTS 设置为高电平。Q1 和 Q2 为截至状态，BOOT0 和 NRST 还原默认电平。
4. 上位机将程序下载到单片机，下载完毕之后，程序自动运行。
5. 至此，很多人还会认为 U18、Q1、Q2 是多余的，用 U6 的 RTS 和 DTR 直接控制也可以。正常情况下，这样理解没有问题，但是我们忽略了一点，就是单片机上电瞬间如果 USB 转串口连接了电脑，DTR 和 RTS 的电平是变化的，如果不处理好，单片机会一直进入 ISP 模式，或者系统会复位多次，这种情况是不允许的。
6. 于是，就有了我们全新的一键 ISP 电路。我们主要是分析上电瞬间的逻辑关系，单片机上电时我们通过示波器观察波形得知 DTR 和 RTS 的电平是变化的，但是也有一个规律就是：只要 RTS 为低电平的时候，DTR 的电平也是低，因此一般情况 Q2 不会导通，但由于这两个 IO 口的电平存在“竞争冒险”，会出现 RTS 的下降沿的时候刚好遇到 DTR 的上升沿，这个时候 Q2 导通，导致系统复位，而 BOOT0 此时有可能也为高电平，就会进入 ISP 模式。这个是不受我们控制的，我们不想系统出现这样的情况。因此加入了模拟开关来切断这种干扰。
7. 加入模拟开关 U18，通过控制 U18 的 4 脚的开关来达到隔离干扰电平的目的。下面我们分析一下延时开关电路，上电瞬间，电容 C65 通过电阻 R18 来充电，由于电阻 100k 很大，电容的充电电流很小，等电容充电达到 U18 的 4 脚的有效电平 2V 时，大概耗时 1S，在这个 1S 时间内 U18 的模拟开关是断开的，因此 RTS 和 DTR 的干扰电平不会影响到系统复位。系统正常运行。