第二天笔记

1. STM32的开发方式

STM32的开发方式有三种：函数库开发 + 寄存器开发 + 图形界面开发

* 函数库开发（**别人做饭，自己动手吃**）

ST公司针对STM32开发了两套库（标准外设库 + HAL库），对于目前的市场趋势而言，ST公司主推HAL库，已经对标准外设库停止维护，但是国内大多数的公司还是在采用标准外设库开发（更稳定、资料更多、容易维护）。

库是ST公司的开发人员为开发者提供的一系列函数接口，使用库函数开发，开发人员不需要对底层的寄存器有过多的了解，只要学习过C语言就可以使用函数来配置对应的寄存器。

优点：

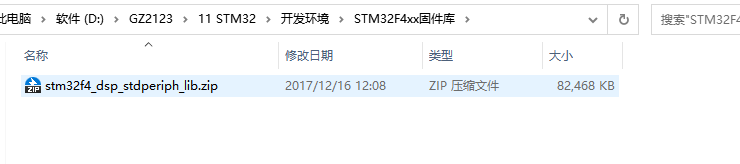
程序容易维护

代码可读性好

提高开发效率

缺点：

相比于寄存器开发，使用库函数开发的程序的运行效率稍微低一些，大概低10%~15%左右，但是对于一些实时性要求不高的产品来说，可以忽略不计。



* 寄存器开发（**自己做饭，自己动手吃**）

STM32属于32位的MCU，STM32内部的资源十分丰富，就导致寄存器的数量和复杂度都增大了，就要求开发人员对底层的知识掌握的非常扎实。

优点：

运行效率高

节约存储空间

缺点：

开发效率低

程序不容易维护

程序可读性差

一般来说，在项目开发的过程中绝大部分的代码都是采用函数库开发，如频繁调用的中断服务函数就需要采用寄存器开发（实时性强）。

* 图形界面开发（**别人做饭，别人喂你吃**）

为了开发人员考虑，ST公司近几年推出了一款图形界面软件（STM32CubeMx），也被称为初始化代码生成器，用户只需要根据实际情况来选择对应的按钮和功能就可以自动生成代码，可以极大程度的提高开发效率，但是一旦程序出现问题，就不容易维护，所以适合有经验的工程师使用，图形界面软件自动生成的代码也是基于HAL库。

优点：

不需要写代码

开发效率较高

缺点：

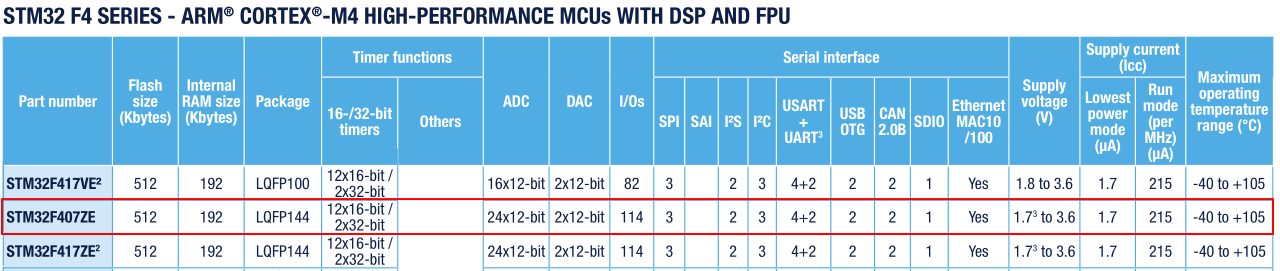
不容易维护

运行效率稍低

1. STM32的开发平台

* STM32开发平台的芯片

本次阶段使用的开发平台为GECM4开发板，M4开发板搭载的芯片是**ST公司**设计的芯片，型号为**STM32F407ZET6**，芯片采用的内核为**ARM公司**设计的**Cortex M4**内核（主频高达168MHZ，带DSP和**FPU 浮点运算单元**），对于芯片型号的说明可以参考ST公司提供的芯片选型手册。



STM32 32位MCU

F 基础型

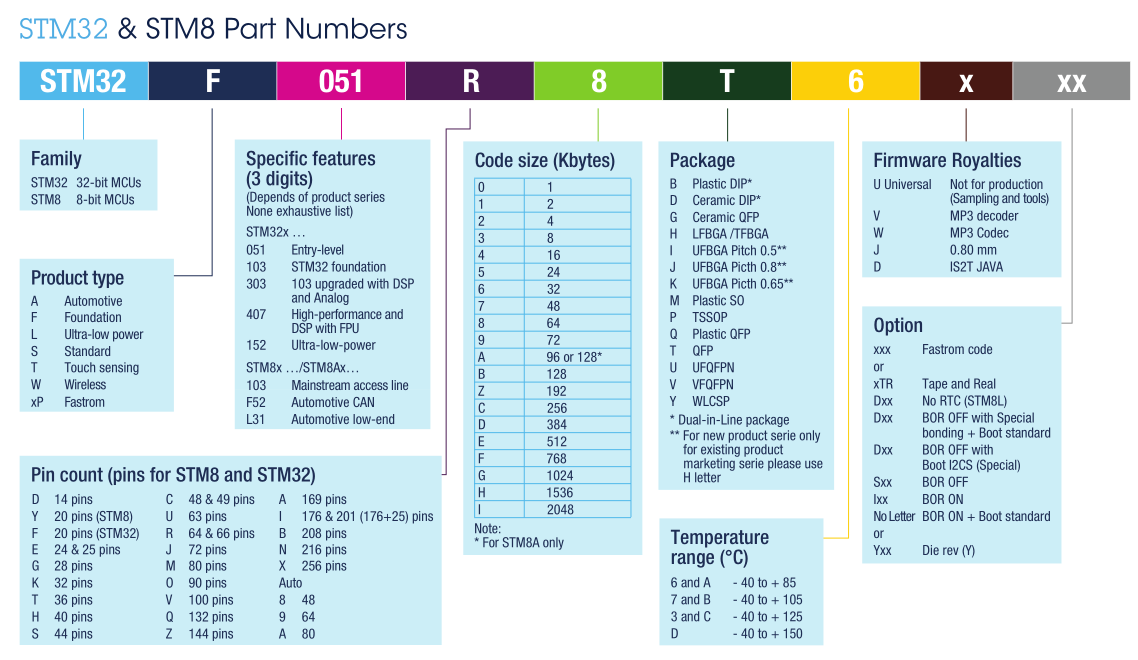
407 高性能

Z 引脚数目144

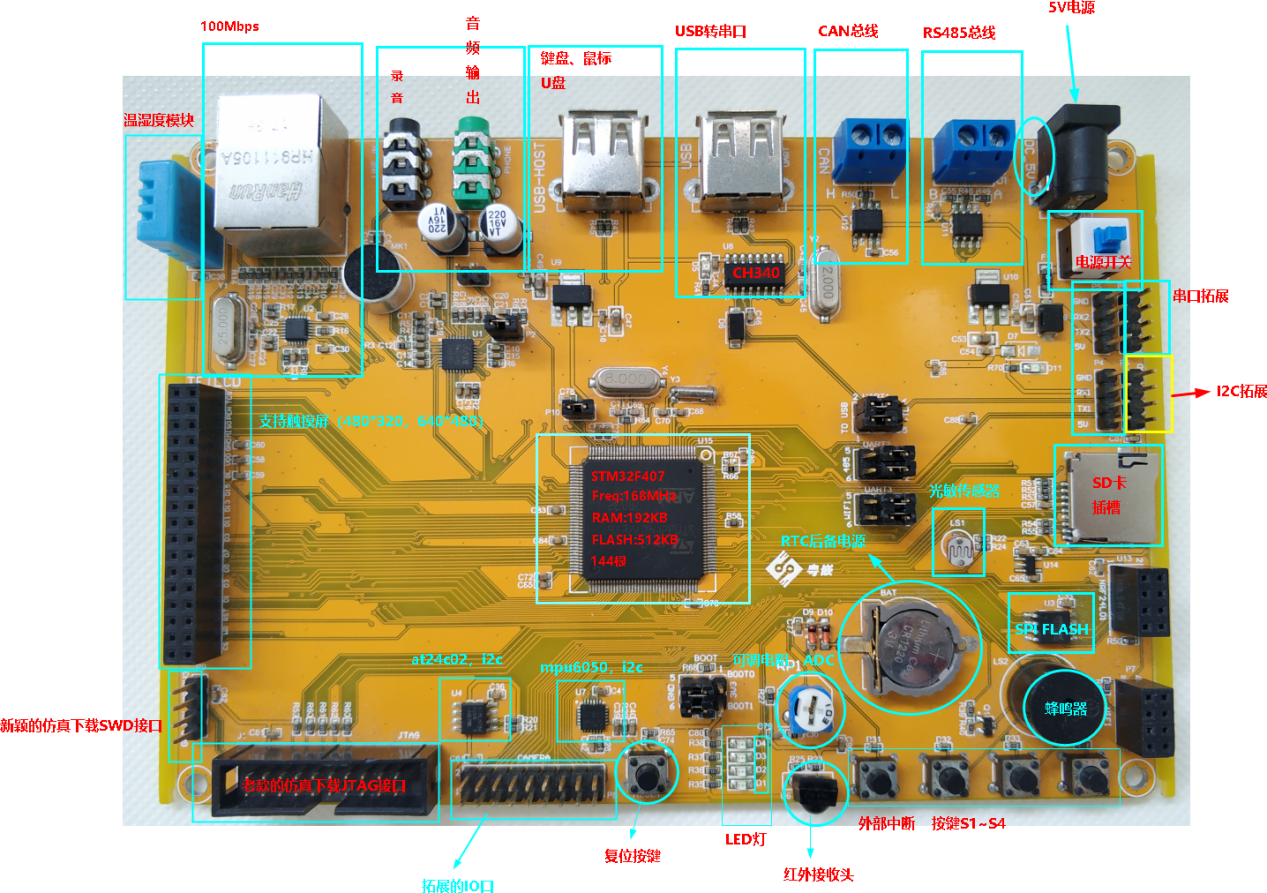
E 闪存容量 512K

T 封装QFP （硬件工程师）

6 温度范围 -40 ~ +85



* STM32开发平台的资源



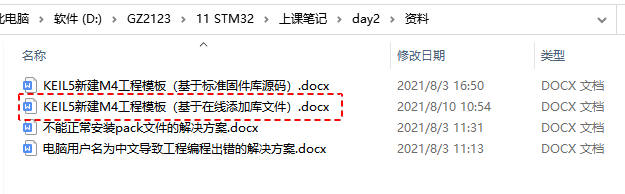
1. STM32的工程建立

一般在实际开发过程中，大家可能只是负责其中的一个模块，所以就必须提前搭建好一个可以方便移植的工程模板，方便大家对程序进行维护和交接，所以就需要把写好的代码进行模块化编程（封装源文件+头文件）。以标准外设库开发为例，搭建工程的方式如下：

使用函数库开发，搭建工程的方式有两种：**在线搭建** + **离线搭建** 两种的区别如下

* 在线搭建

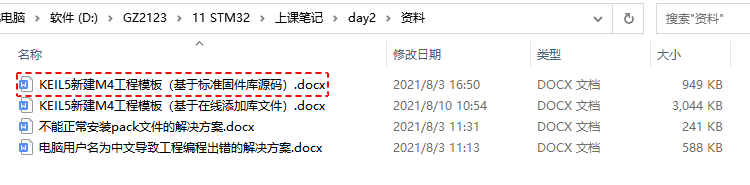
要确保在安装开发环境的时候已经安装了芯片支持包（芯片型号、芯片外设的驱动程序）



需要注意：如果安装的芯片支持包的版本较新，则内部并不会集成标准外设库的接口，所以如果打算使用标准外设库并且打算在线添加，则需要安装一个版本较低的芯片支持包。

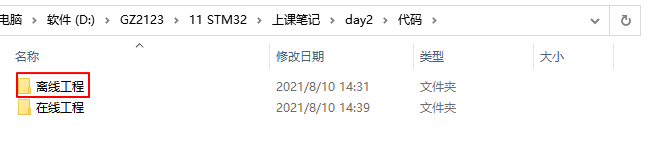
* 离线搭建（**推荐**）

离线搭建的方式是指去官网下载标准外设库的库压缩包，通过调用库压缩包中的文件来实现工程的建立。





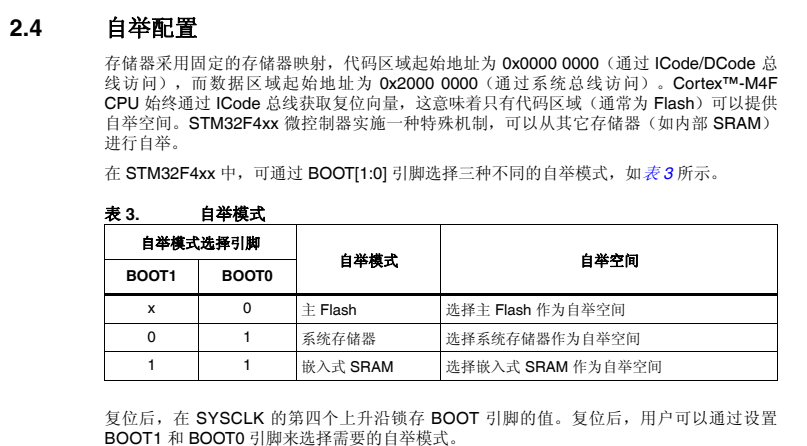
**练习：按照文档要求，搭建一个工程模板 （只需要搭建这一次即可，以后再也不用搭建）**



1. STM32的程序下载

STM32想要下载程序的方法有两种：串口下载 + 下载器下载（JTAG、JLINK、STLINK.....），这里就涉及到STM32启动方式（自举模式）。这里可以参考STM32F4中文参考手册的2.4章节。

STM32启动方式（自举模式）有三种：从用户闪存启动（Flash）、从系统存储器启动、从内部SRAM启动。

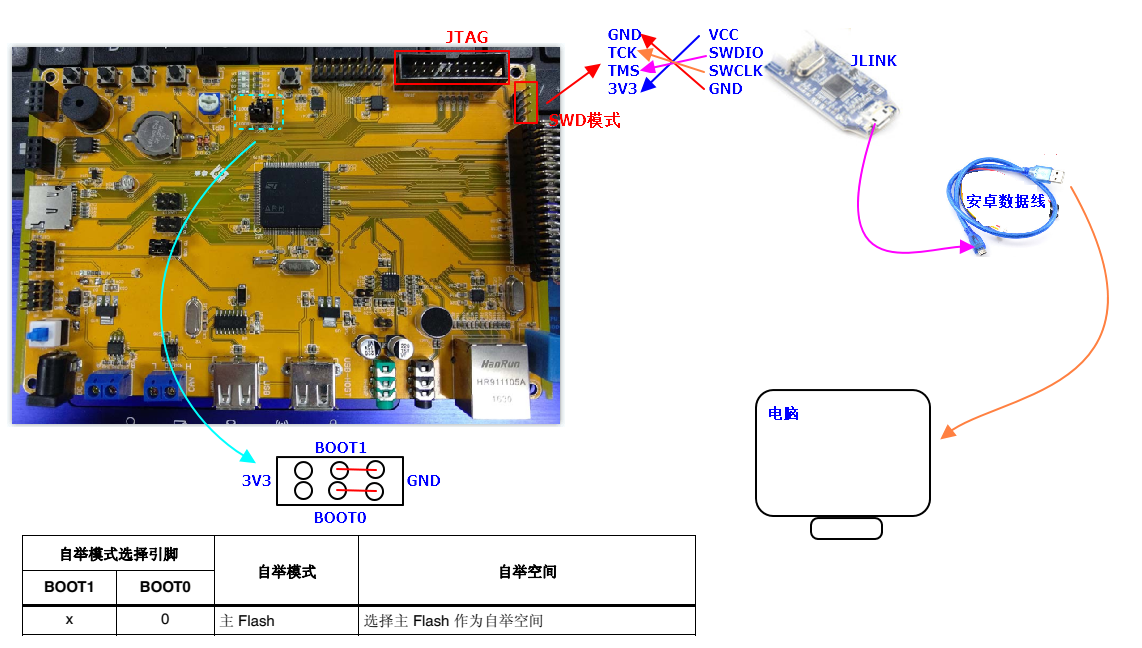


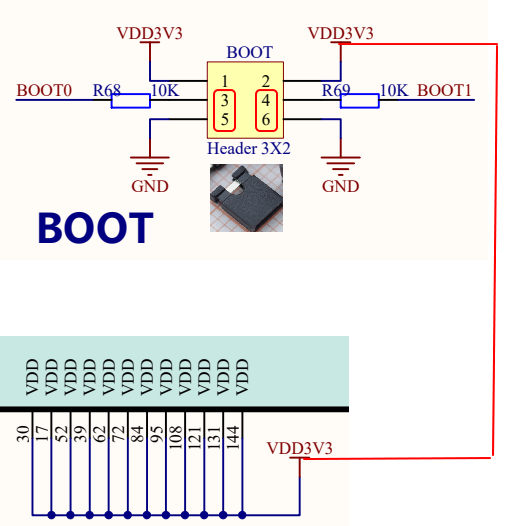
STM32的自举模式可以通过芯片的两个引脚BOOT0和BOOT1进行配置，这两个引脚在芯片复位时的电平状态决定了芯片复位之后从哪个区域运行程序。

* 主Flash启动 （**推荐**）

指的是从用户闪存空间启动（FLash空间 512K），也是**最常用的工作模式**，一般用JTAG接口和**SW**接口下载程序的时候意味着程序要下载到Flash中，在复位之后也是从Flash运行程序。一般用户闪存（Flash）可以反复擦除10万次左右，所以不用担心芯片锁死或者被擦爆。

1. 硬件接线（JLINK下载器）  **BOOT0必须接地 BOOT0 = 0 BOOT1=X**





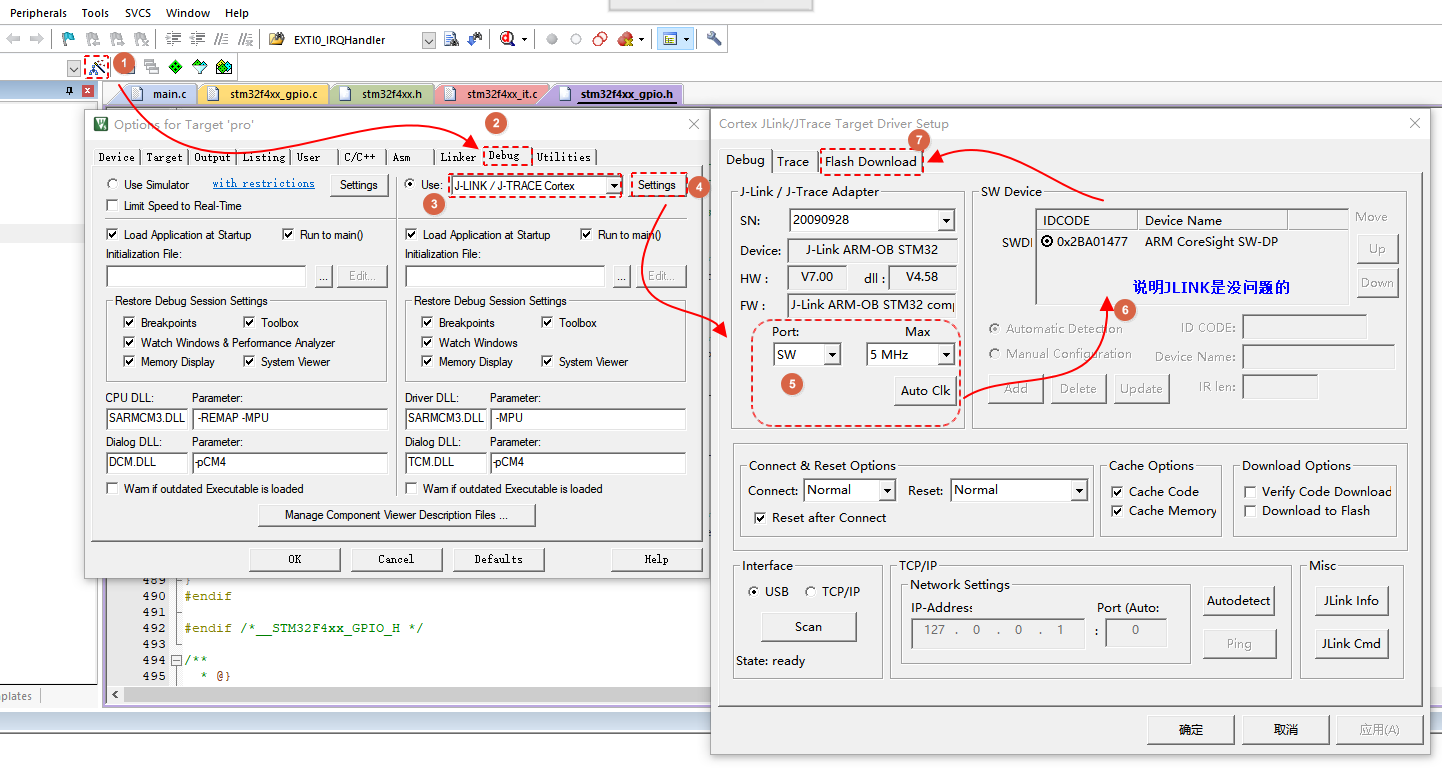
1. 安装JLINK的驱动程序 注意：如果安装最后弹出一个窗口 记得勾选MKD5

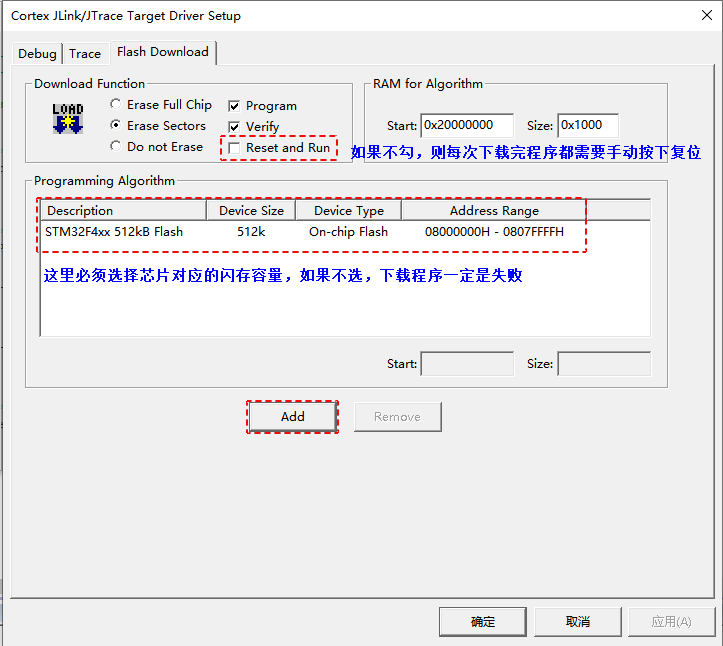


1. 检查JLINK驱动是否安装成功 我的电脑-设备管理器-通用串行总线控制器-JLINK-Drvier

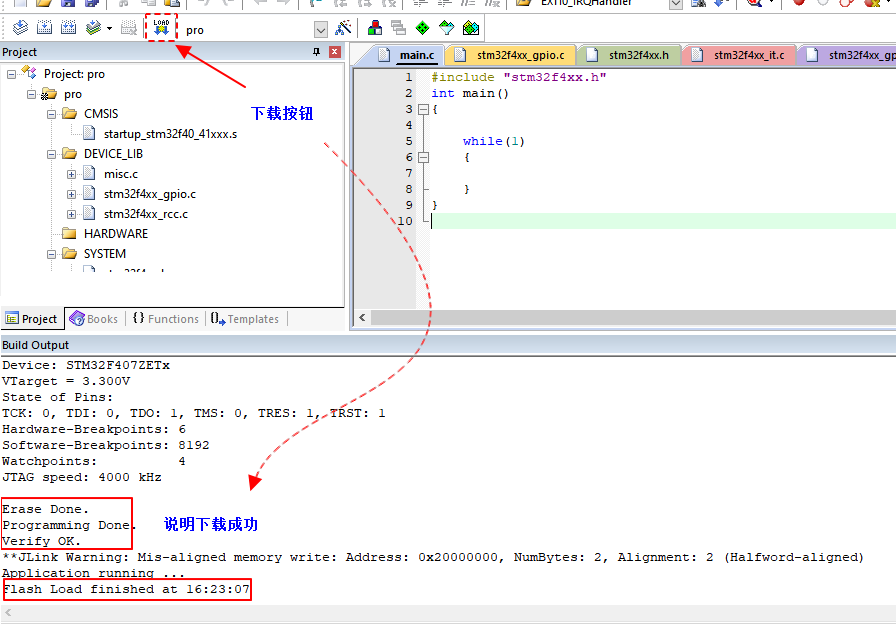


1. 打开MKD5，进行下载器的配置 点击魔术棒---DEBUG---选择JLINK---配置为SW模式





1. 点击MKD5的下载按钮 Download，等待程序下载完成即可

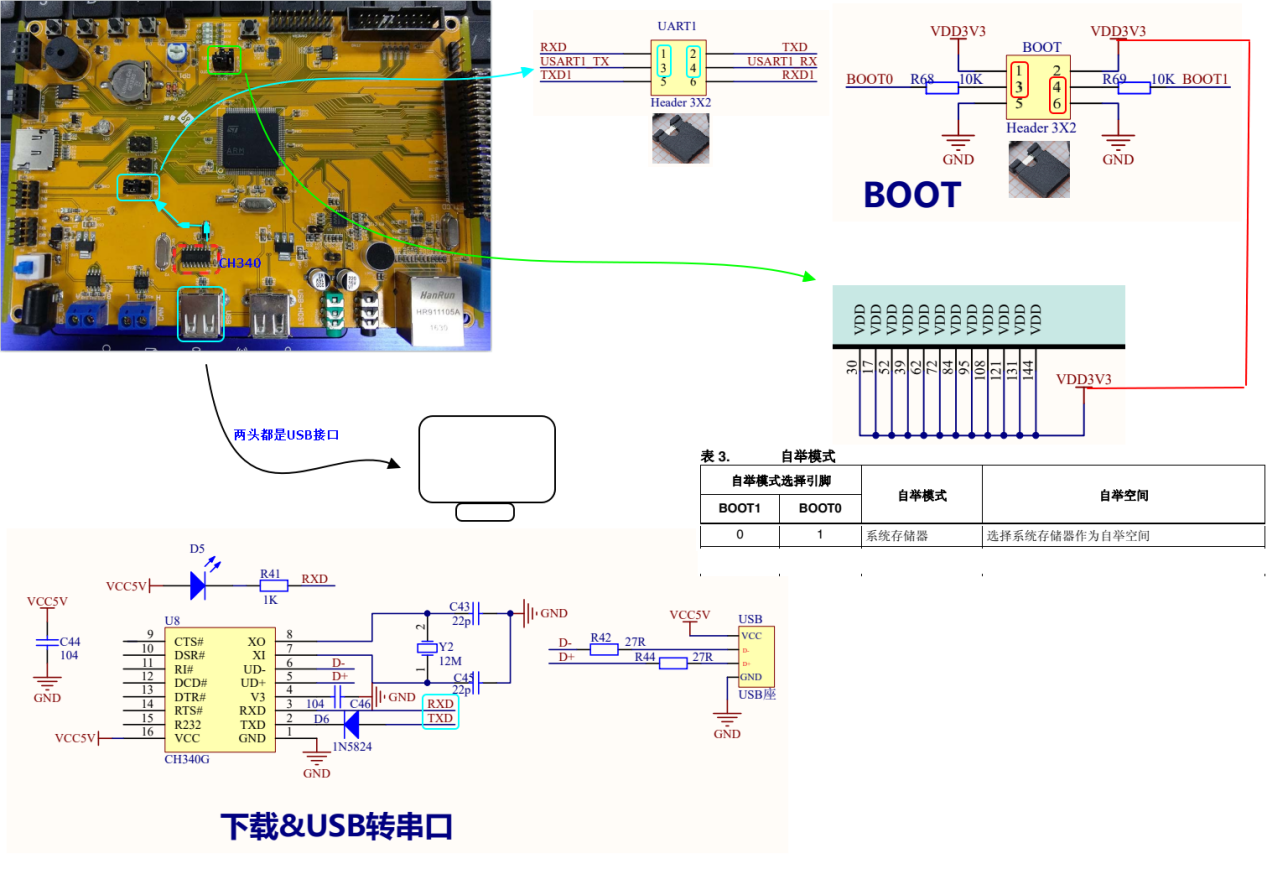


练习：利用Flash自举模式下载程序，熟悉下载流程

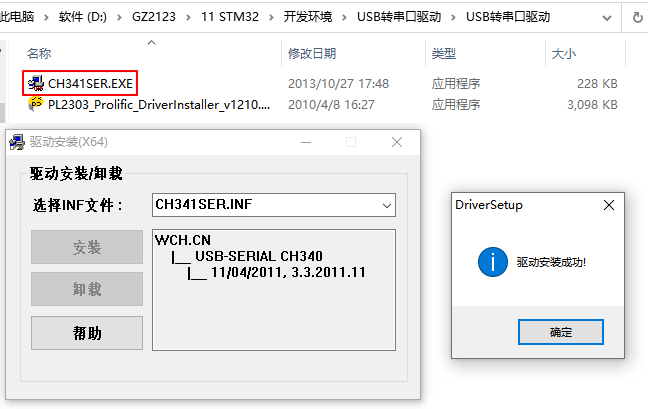
* 系统存储器启动

系统存储器指的是芯片中一块固定的区域，这块区域的程序的功能是由芯片厂商在出厂的时候就写入的一段程序，也被称为启动引导程序（BootLoader），是无法修改的，所以系统存储器其实就是一块ROM（只读存储器），一般通过系统存储器启动都是打算采用串口下载的方式烧写程序（ST公司的固件中包含了串口下载的功能），但是一般不常用，因为配置起来比较麻烦。

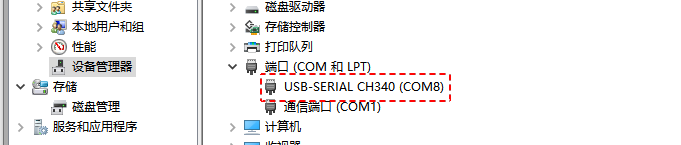
1. 硬件接线（串口线） BOOT1 = 0 BOOT0 = 1 **按一下复位键** （可以执行BootLoader）



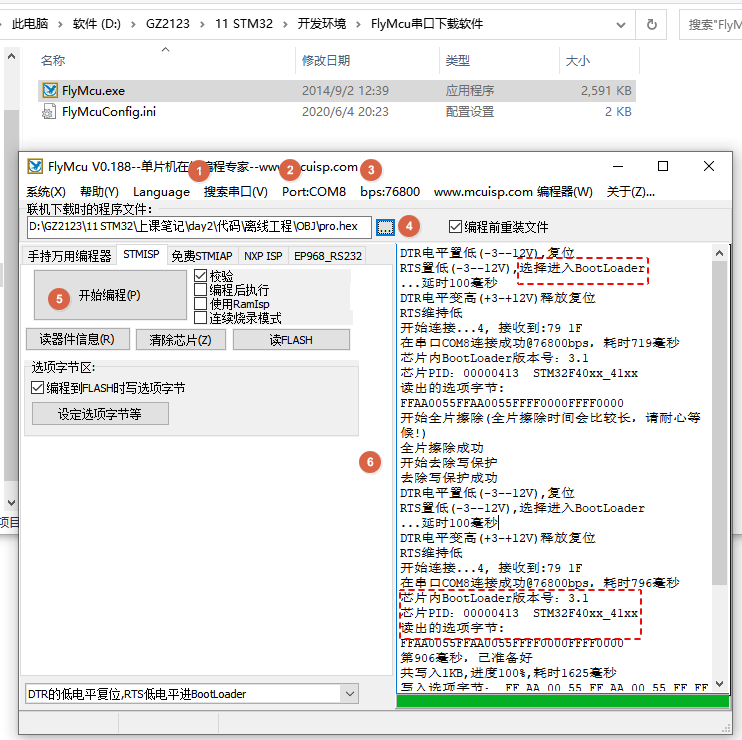
1. 安装CH340驱动 安装驱动之前需要把线接好



1. 检查CH340驱动是否安装成功 并知道CH340提供的端口 我的电脑-设备管理器-端口



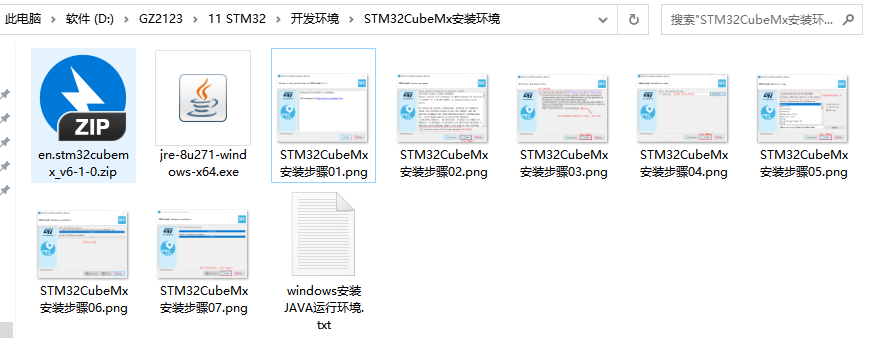
1. 打开串口下载软件FlyMCU，进行简单的配置 端口 + 波特率 + 可执行文件



1. 再次设置**BOOT0=0，BOOT1=0**（选择Flash自举），按一下复位（程序开始运行）

练习：利用系统存储器自举模式来完成程序的下载和运行（熟悉串口下载程序的流程）

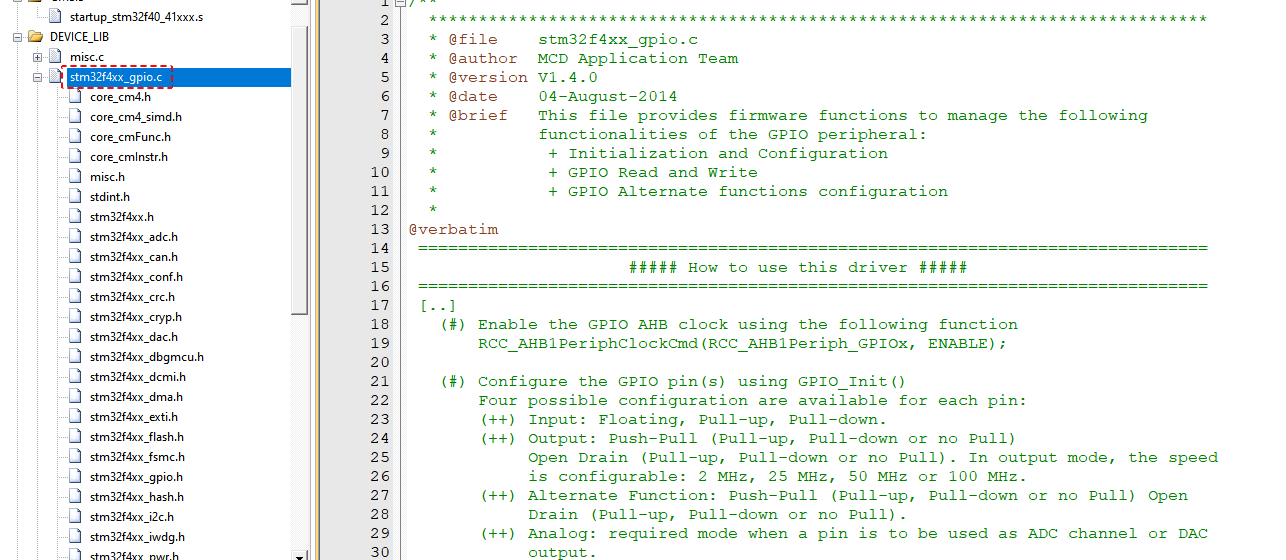
**作业：完成STM32CubeMx软件的安装，并学习软件的配置（如何利用该软件搭建工程）**



预习：阅读ST公司提供的函数库帮助手册（外设的使用和配置、库函数的命名规则、内核 结构）



预习：阅读ST公司编写关于GPIO的外设源文件+头文件 （函数的参数+函数的使用+注释）



晨讲：请简述CM3和CM4内核中的寄存器组R0-R15的作用，尽量详细+举例 （一杯奶茶）



会从线上或者线下抽 线上的同学可以在晚上录制一个简单的视频

午讲：能不能说一下入职之后的职业规划是什么？