**Stm32软件开发基础说明**

1. **Windows系统下开发stm32**
2. **必备软件**
3. **STM32CubeIDE**

STM32cubeIDE是ST官方推出的一款用于开发STM32的工具，整合了STM32CubeMX和TrueSTUDIO而成，对于STM32的开发这个工具在未来会逐渐成为主流，毕竟官方出品维护并且又商用免费。

下载地址：<https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-ides/stm32cubeide.html>

STM32cubeIDE可以在ST官网搜索下载，包含Win、Mac以及几种Linux发行版的版本，参考图1，点击箭头所指处获取window版本软件：

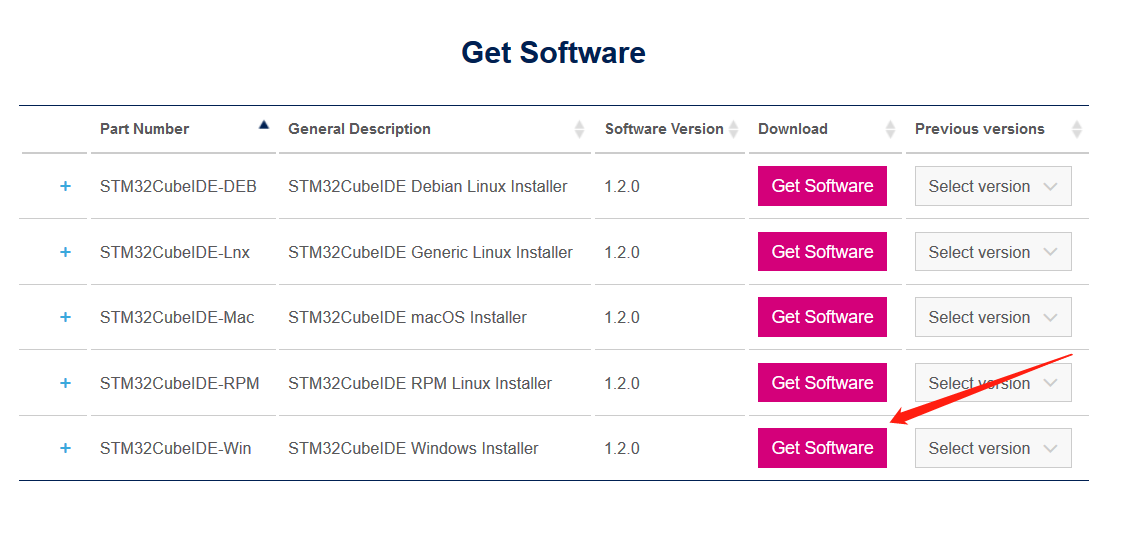


图1. STM32cubeIDE软件下载

需要注意的是，**安装时注意路径必须使用ASCII字符集中的（即不能出现中文）**，否则可能会报错。安装完成后首次打开软件会提示选择工作空间（Workspace），需要指定一个文件夹，软件会在该文件夹下生成.metadata文件夹（用来存放对于该工作空间中软件的各种设置、项目索引等），关于工作空间的更多内容会在之后的文章进行介绍。

1. **串口调试助手**

串口调试助手是串口调试的相关工具，我们在进行软件调试的过程中，常用的软件。在网络上有很多个版本的串口调试助手，没有串口调试助手的同学可以通过以下链接获取。

链接：https://pan.baidu.com/s/1W9YEWawSggjPOsSJiuUoiw

提取码：46wd

1. **STM32CubeProgrammer**

STM32CubeProgrammer烧录程序的软件。为任意环境下的STM32微控制器烧录提供了一个一体化的软件工具：多操作系统（Linux、Windows、macOS），图形用户界面或命令行界面，支持多种连接选择（JTAG、SWD、USB、UART……），采用手动操作或通过脚本自动操作。

下载地址：

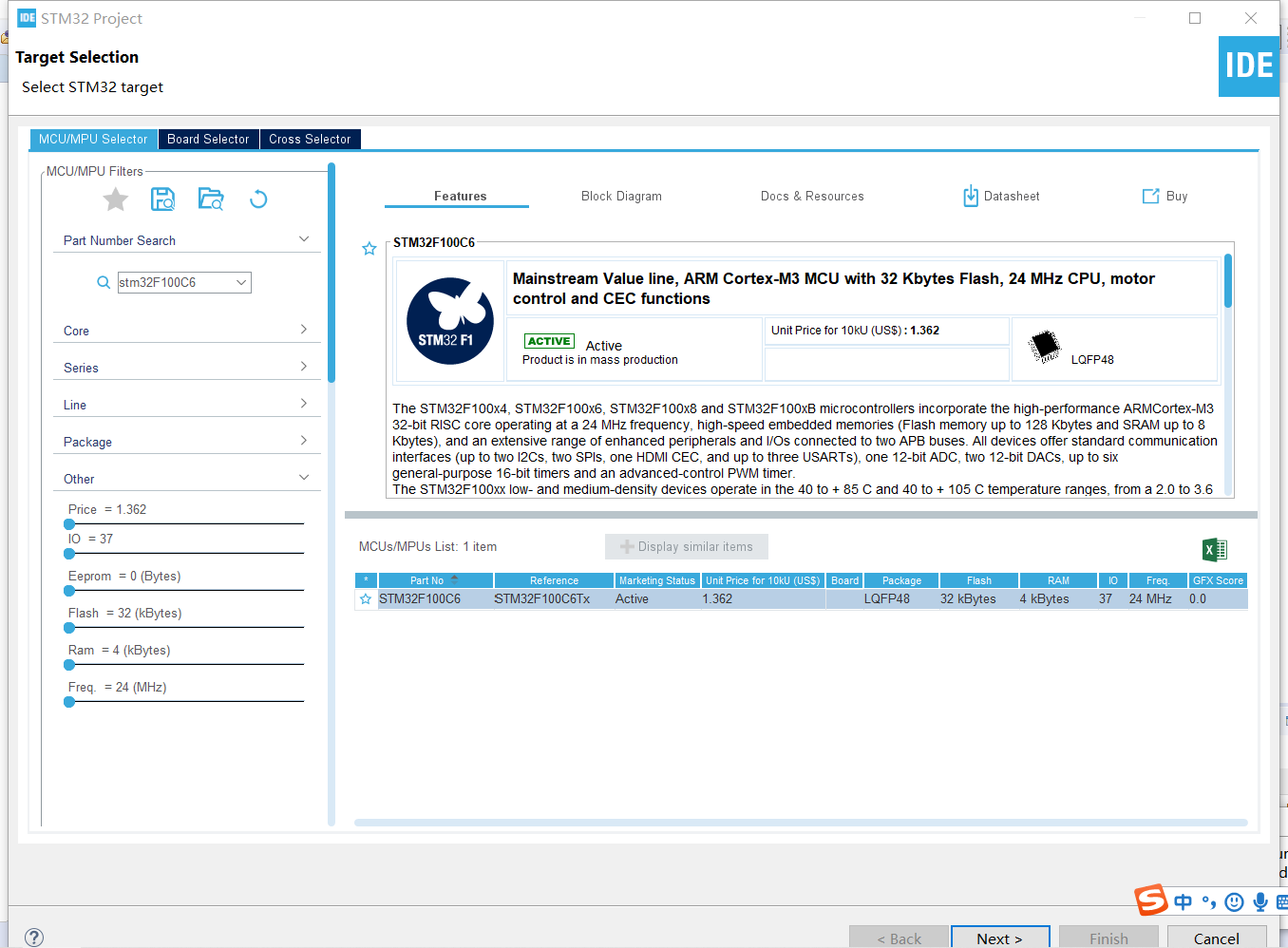
<https://www.st.com/zh/development-tools/stm32cubeprog.html#get-software>

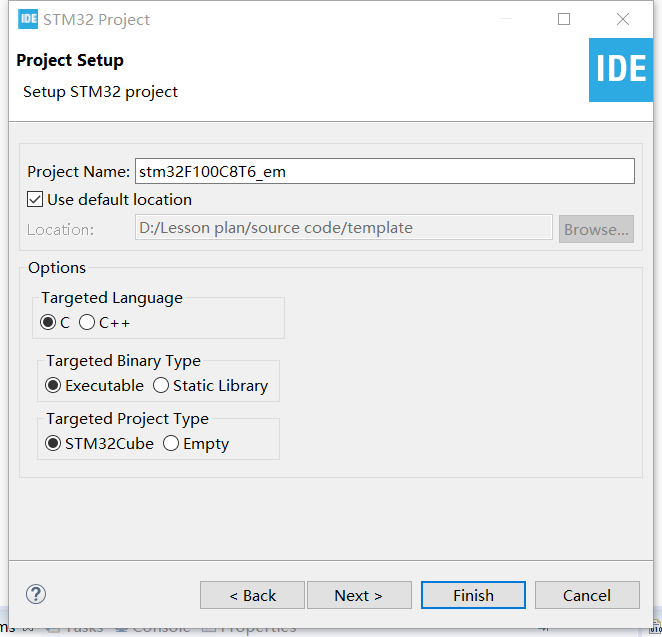
同样的，在安装过程中，不要出现中文路径。

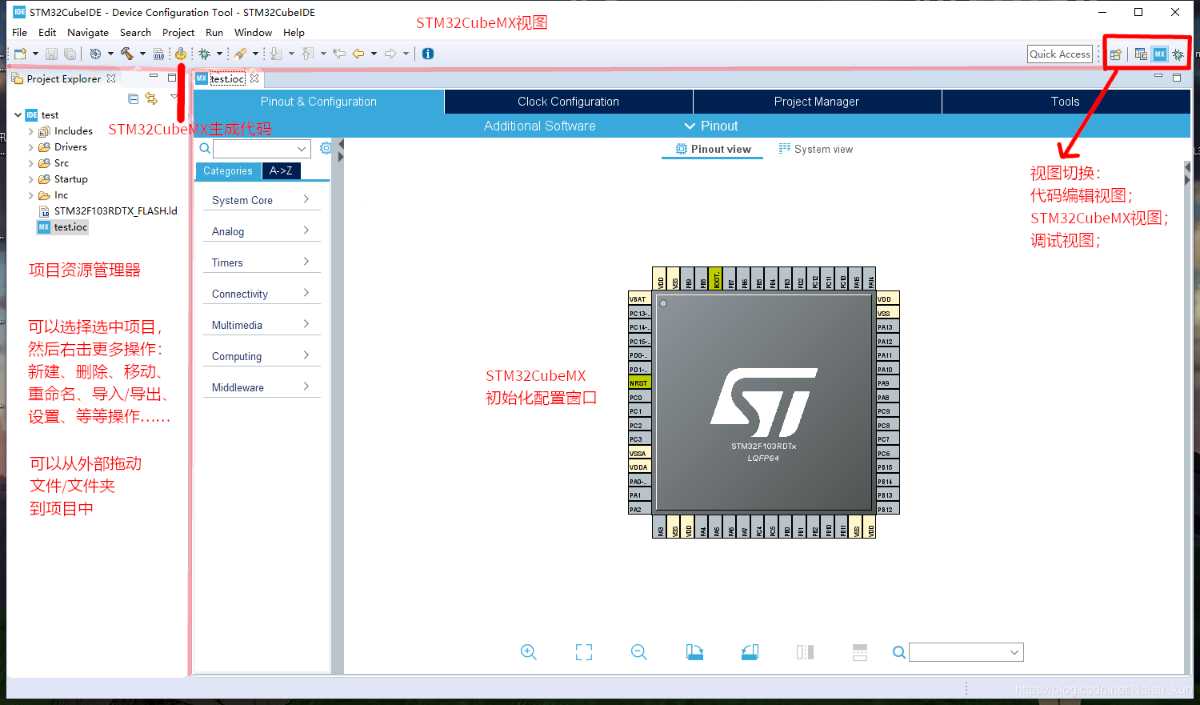
1. **Stm32开发的工程建立**

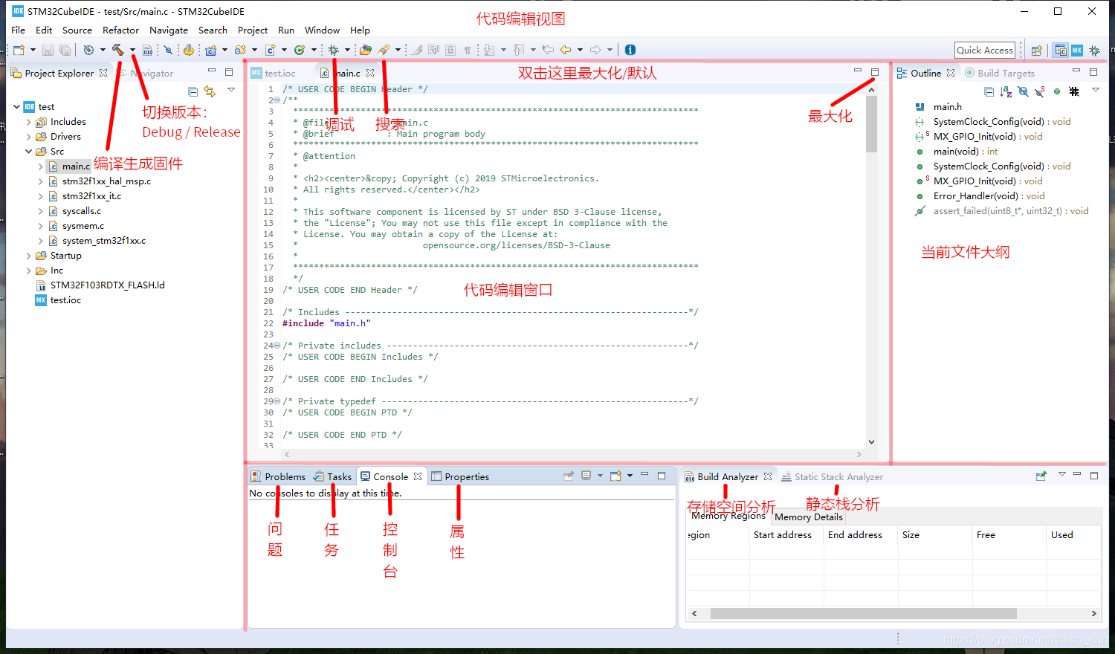
通过STM32CubeIDE建立一个空的工程，此工程不实现任何功能。由于pixhawk飞控板载两块stm32主控芯片，先对STMF100协处理器进行操作。

1. **新建项目（有多处可以新建项目，通用方式 File > New > STM32 Project ）**



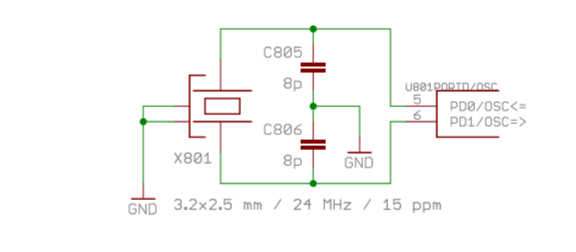






1. **配置时钟系统并生成初始化代码（操作的是： 项目名.ioc 文件）**

在完成上述步骤时候，我们需要根据板载系统的硬件，配置好芯片的时钟系统。下图所示，为原理图上协处理器的外部晶振。从电路板上也可以看到，是一个3225贴片封装的24MHz石英晶振。

****

回到STM32CubeIDE软件中，选中左侧边栏项目中的.ioc文件，进入STM32CubeMX视图，选中Pinout & Configuration选项卡，选中System core->RCC，按下图所示配置好。RCC即为Reset and clock control，时钟配置，控制提供给各模块时钟信号的通断。关于stm32F100芯片的时钟系统，另附“Stm32的时钟系统”文章中，详细进行介绍。

在此我们先对一些基本的名词进行解释，在 STM32中，一共有 5 个时钟源，分别是 HSI 、HSE、LSI 、LSE 、PLL 。

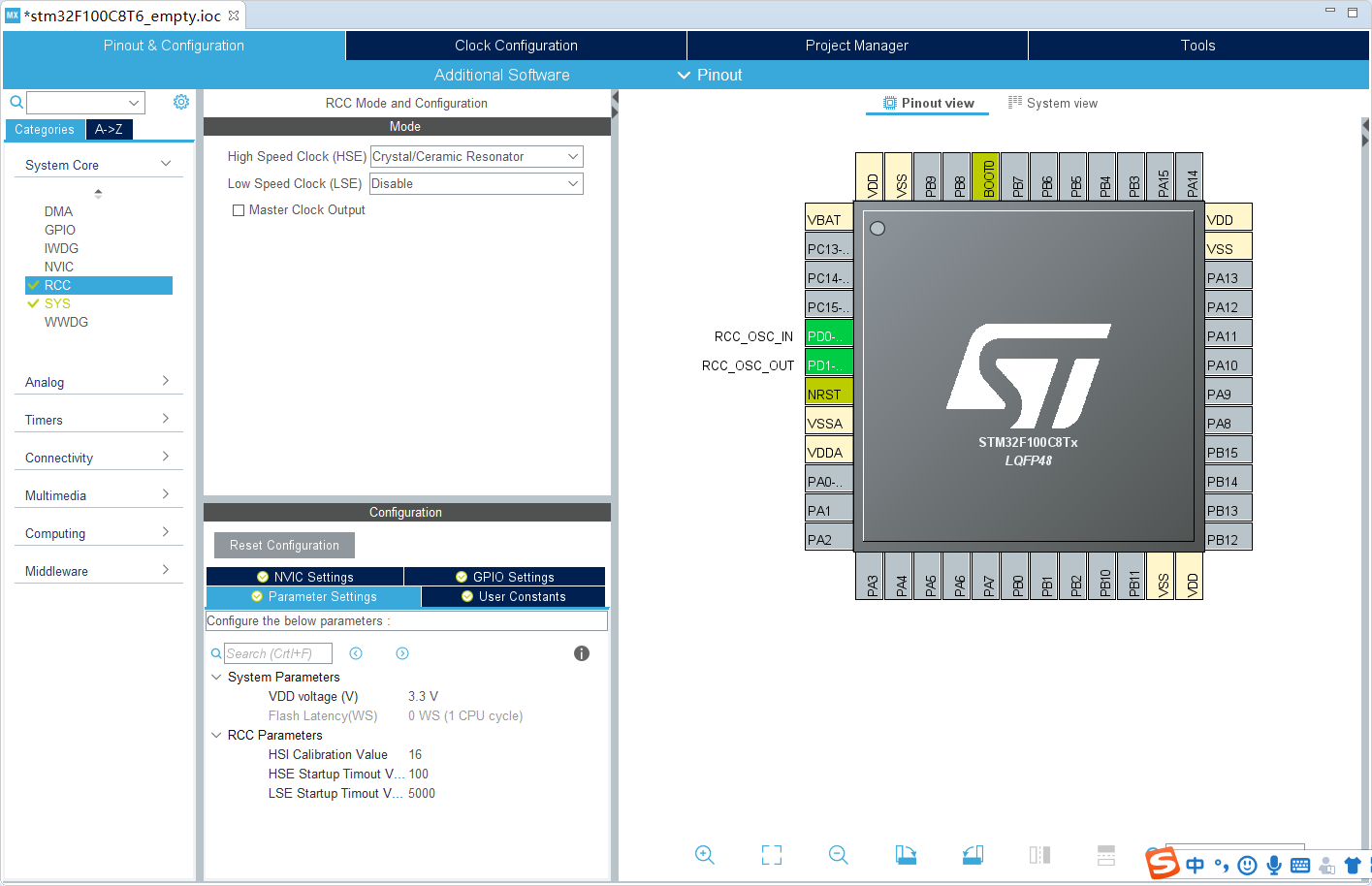
①HSI 是高速内部时钟， RC 振荡器；

②HSE 是高速外部时钟，可接石英 / 陶瓷谐振器，或者接外部时钟源；

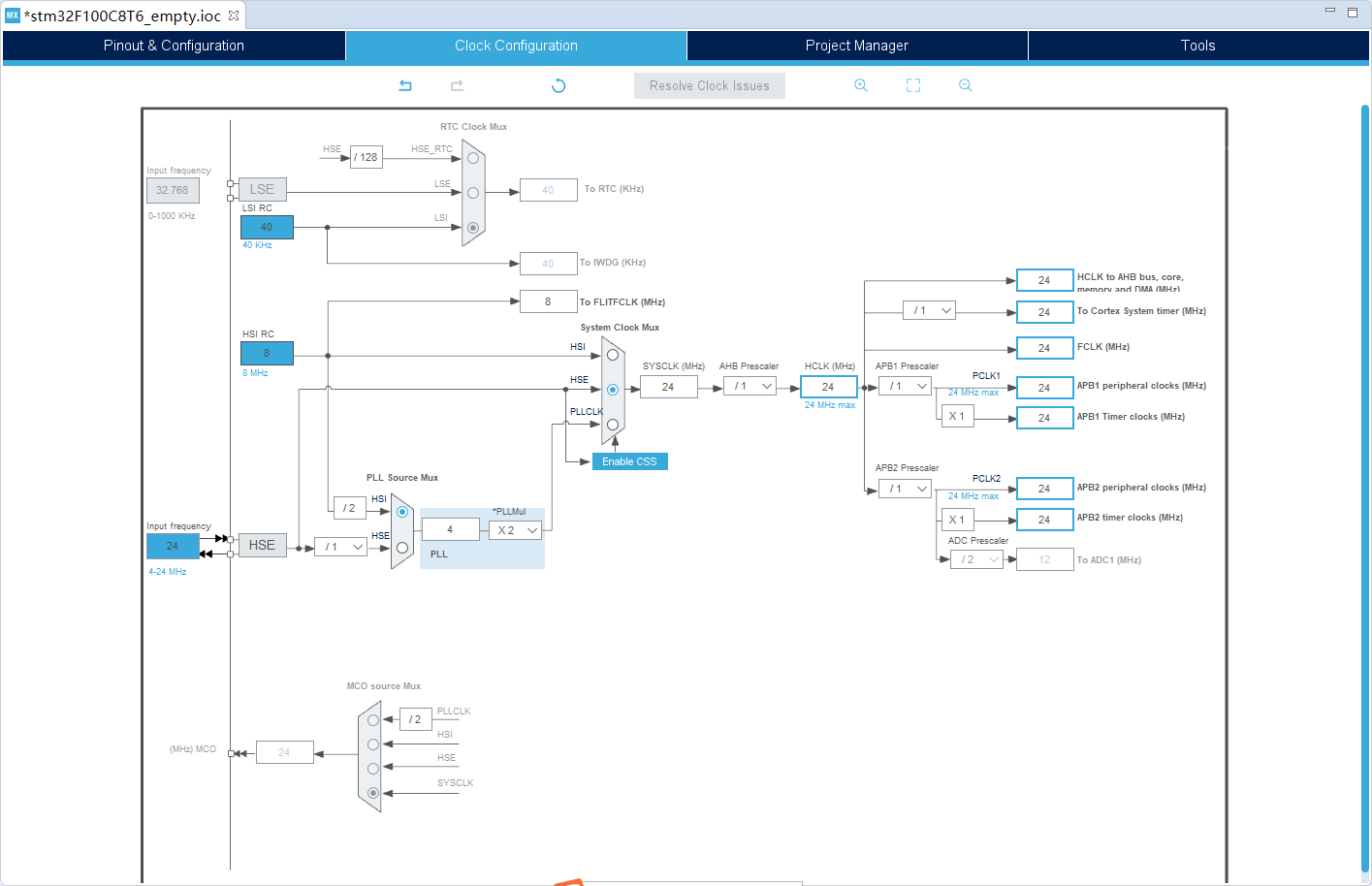
③LSI 是低速内部时钟， RC 振荡器；

④LSE 是低速外部时钟，接频率为 32.768KHz 的石英晶体；

⑤PLL 为锁相环倍频输出。

****

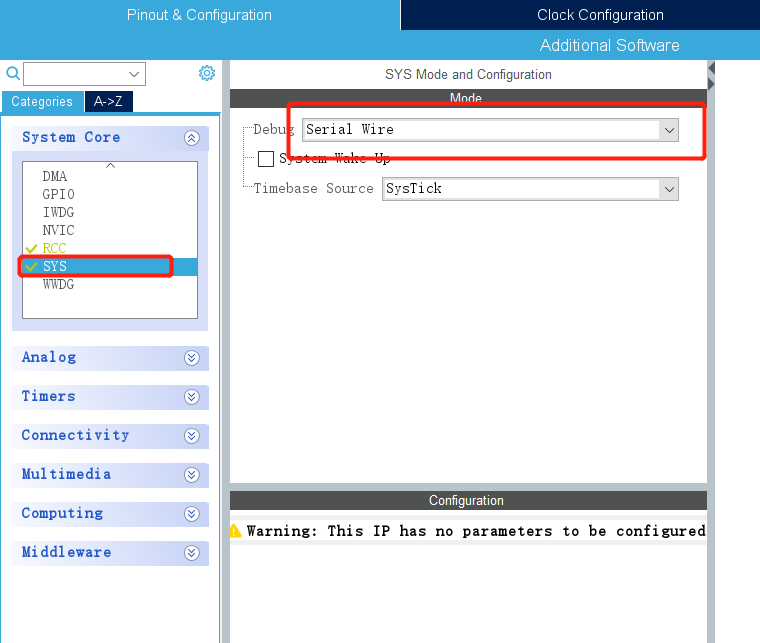
在完成了上述步骤之后，如下图所示，点击进入Clock Configuration选项卡，修改HSE输入频率为24MHz，因为pixhawk飞控板上的晶振为24MHz。之后配置System Clock Mux（系统时钟多路复用器）为HSE输入。

****

1. **Debug方法的配置**

JTAG和SWD通常是开发STM32常用的调试方式，在本实验课中，我们选择更加简洁的SWD方式，在STM32CubeIDE中的配置方式如图所示。详细了解JTAG与SWD的区别可以参考如下表格中的网址：

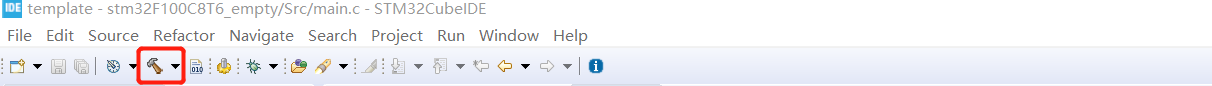
|  |
| --- |
| <https://blog.csdn.net/yuyangyg/article/details/77532485> |



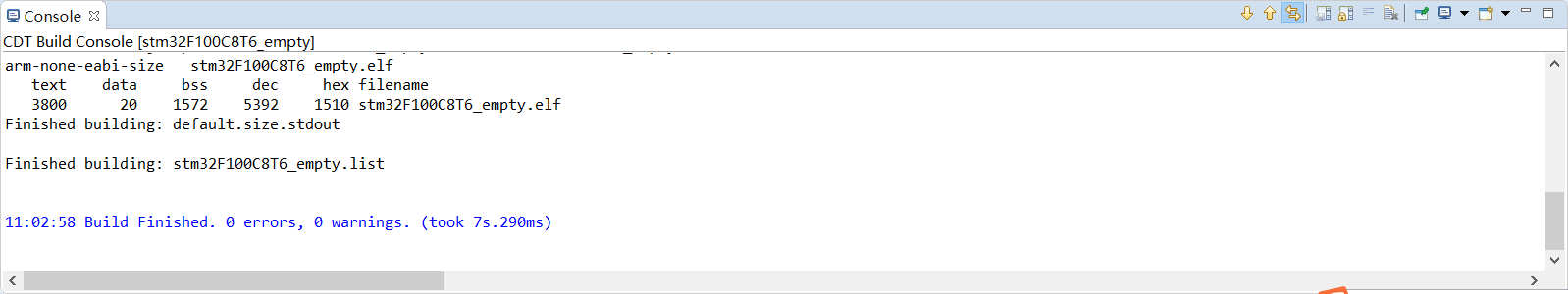
1. **代码编译**

完成上述的配置之后，可以开始看看代码了，在左侧的边栏中选中，src->main.c文件，可以看到主函数的代码。

点击下图所示的图标，进行代码编译：

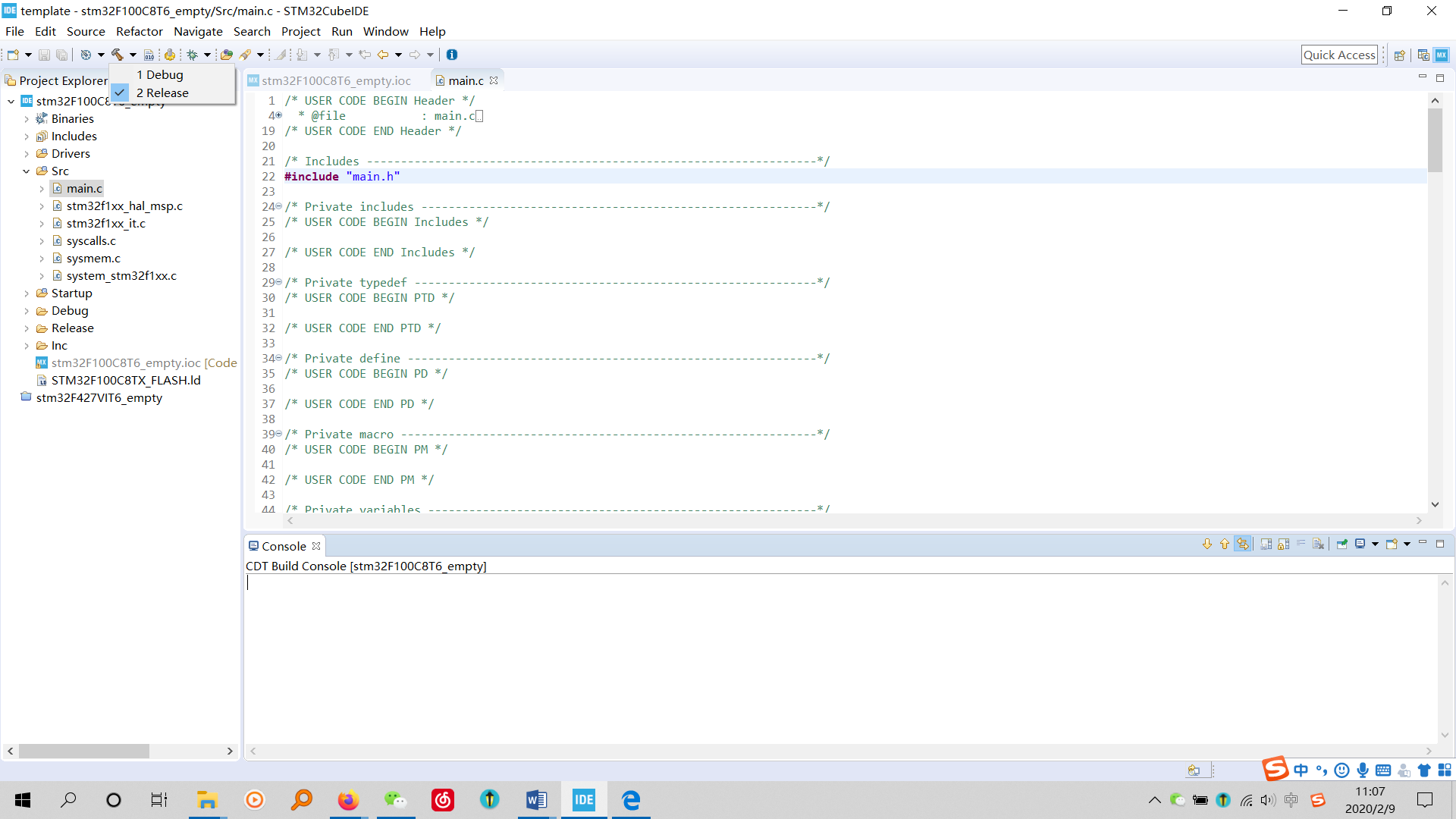


编译完成，如下图所示：

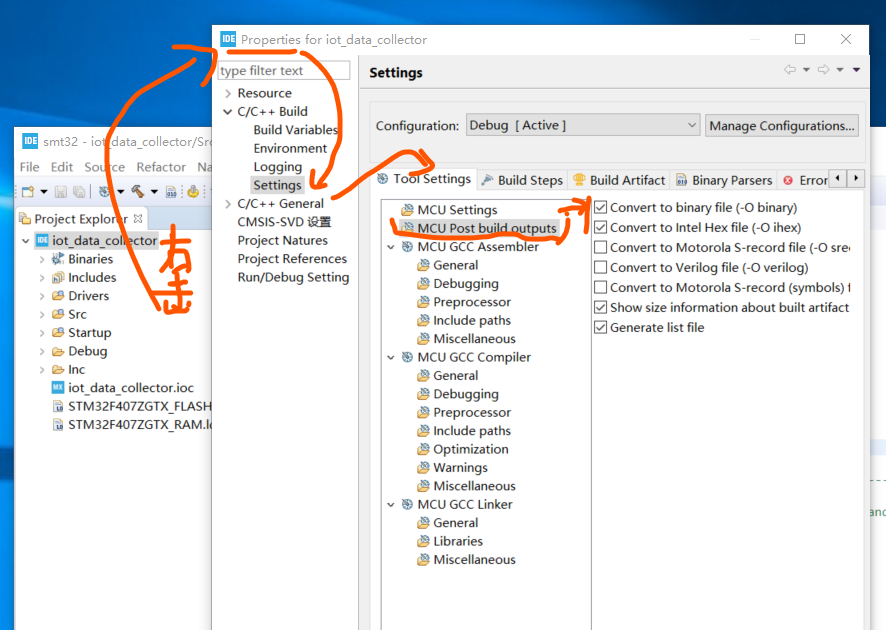


1. **生成烧录用的文件**

编译通过之后，按下图操作，生成可以被烧录到芯片中的文件：



查看自己工程文件的路径，可以发现该文件夹下出现了一个Release文件夹。项目文件夹 -> Release文件夹 -> 项目名.elf，即为生成的烧录文件。不仅仅只有.elf文件可以作为烧录文件，.hex和.bin文件也可以作为烧录文件使用，按照如下图所示的方式配置。

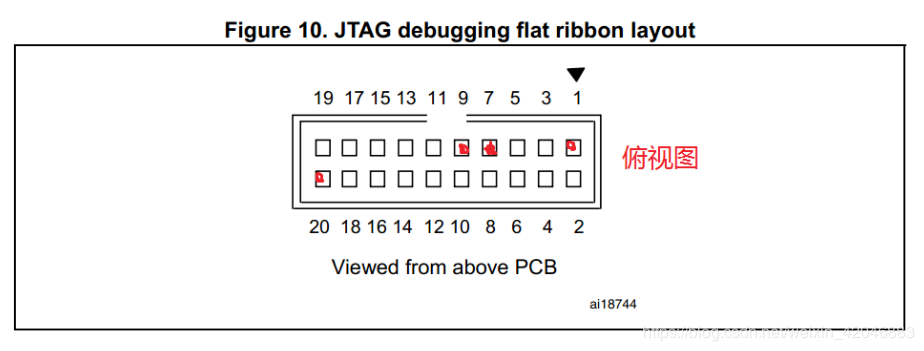


1. **程序烧写**
2. **硬件连接：**

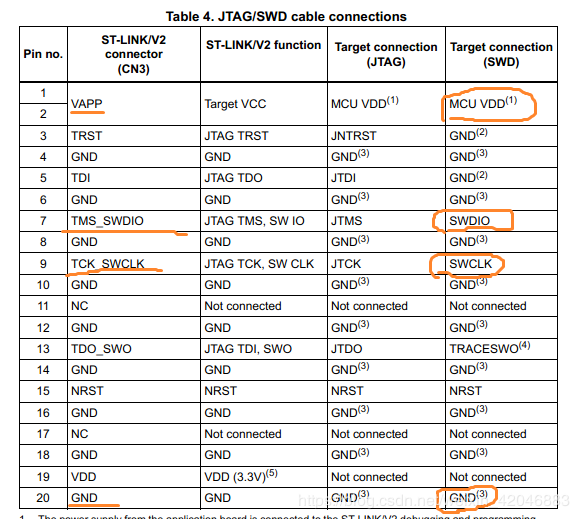
下图为ST-link烧写器的外形图，在本实验课中我们选择的烧写器版本是ST-link v2：



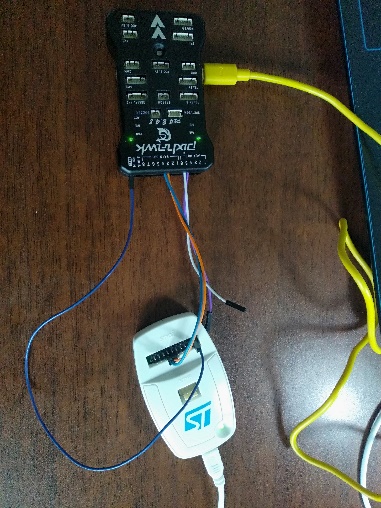
接口引脚顺序定义如下图所示，注意缺口位置。



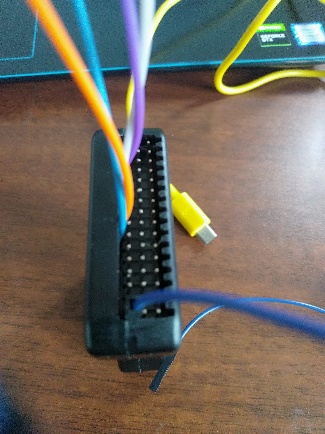
按照下表中所示的接口名字，一一与我们之前从pixhawk飞控电路板上焊接引出来的杜邦线相连接，需要注意的是，当时我们引出了两块芯片的swd，而我们现在用的是stm32F100芯片的swd接口。即为：**IO-SWCLK接ST-link的9 pin，IO-SWDIO接ST-link的7 pin。同时需要将两个硬件的共地，即为将飞控板的GND与ST-link的GND连接**



连接好板子的一端之后，把ST-link的另一端通过usb端口与计算机相连接。硬件接线图如下所示：

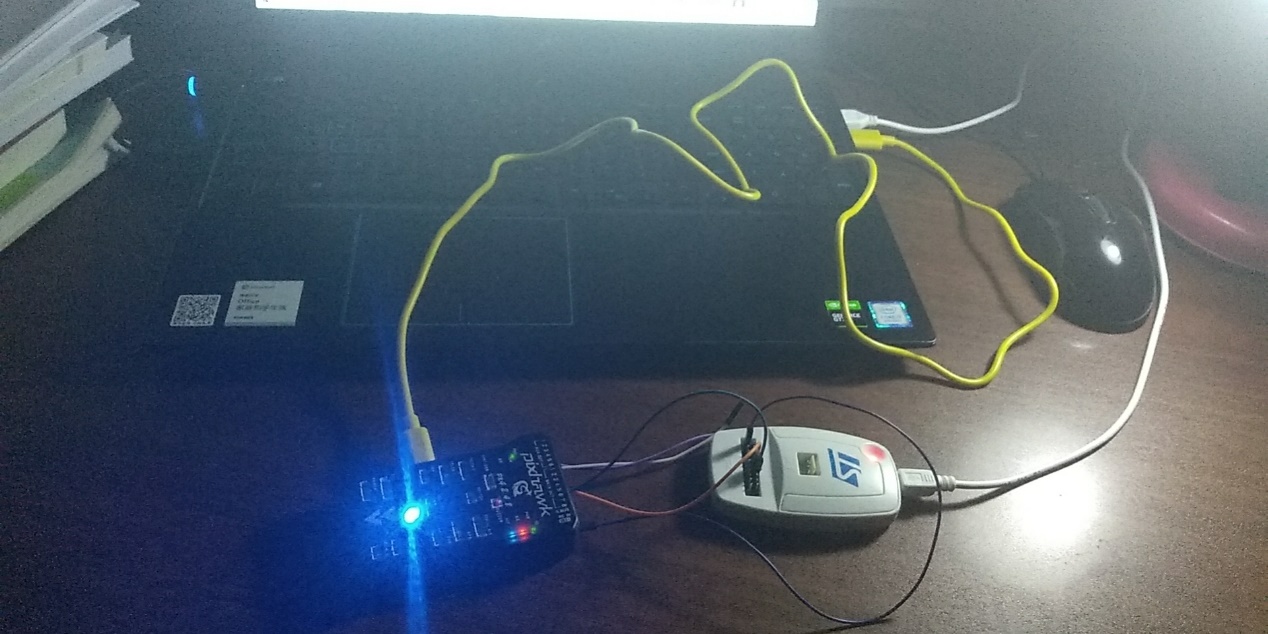


图中蓝线对应的是GND，从飞控板上引出，如下图所示：



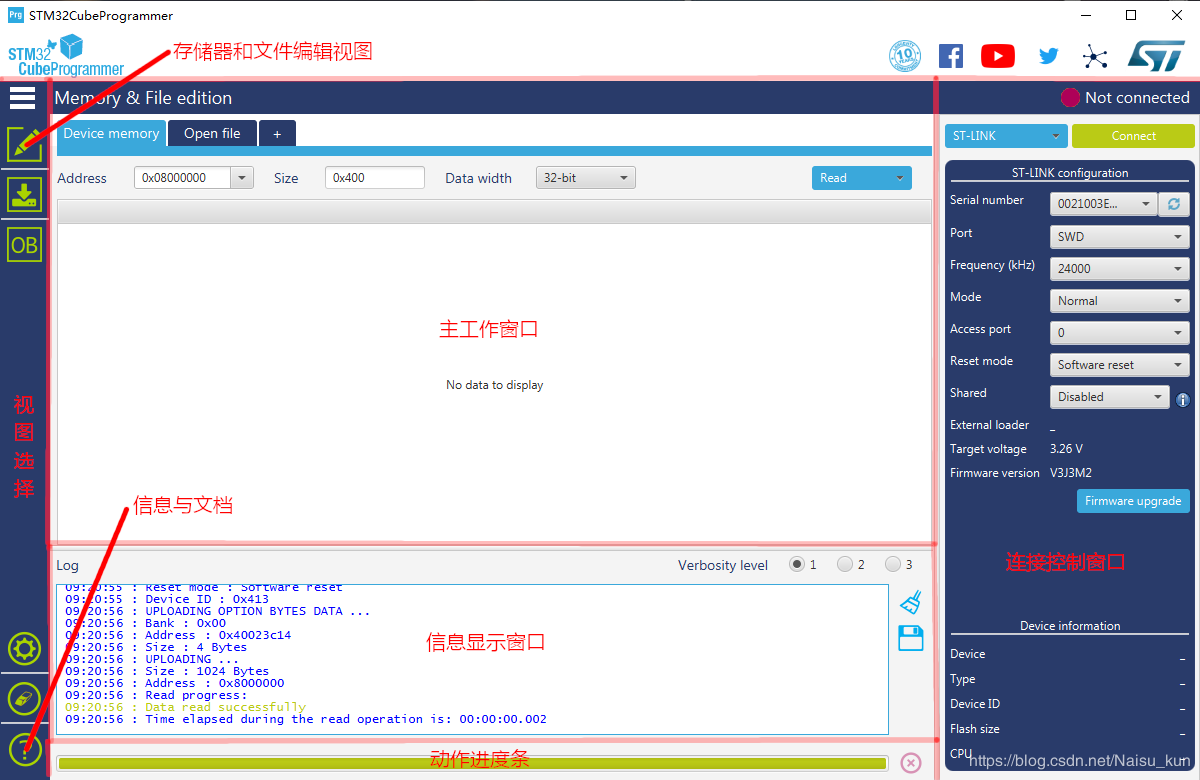
1. **软件烧录：**

首先按照下图所示，连接好烧写器，开发板和上位机，需要注意的是，需要给飞控板连接上usb线供电，具体原因可参考其原理图中的电源模块。



打开软件STM32CubeProgrammer，单击右侧边栏的Connect。如果连接失败，很大的可能是ST-Link V2的固件版本太低的缘故，需要更新固件。具体的操作方式，可类比下表格中的网址解决。在此不再做使用STM32CubeProgrammer进行固件升级的详细图解。

|  |
| --- |
| <https://blog.csdn.net/sinat_29891353/article/details/82778978> |



File path中要选择之前在工程路径下生成的 “项目名.elf”文件，之后单击Start Programming，待程序烧录完毕。



因为我们在程序中，并没有写任何代码，只是一个空的程序，所以烧录完毕之后看不到任何效果。本例程目的在于，熟悉整个工程建立的过程，和烧录程序的过程。

1. **linux下开发stm32**

linux系统下的嵌入式开发，应用也非常广泛，在此也对其做简单的介绍。

1. **linux双系统的安装**

鉴于目前大部分计算机都是window系统，要学习linux系统，常用的方法就是安装一个linux和window的双系统。建议安装Ubuntu 16.04版本的linux系统。

可参考博文：<https://blog.csdn.net/fanxueya1322/article/details/90205143>

1. **linux下的开发**

开发中同样需要安装，STM32CubeIDE，与STM32CubeProgrammer两个软件，但是需要下载linux下版本的软件，其他的操作与在windows系统下的操作类似，在此不再做详细记录，可自行摸索尝试。