

## 2、STM32开发环境搭建

- 2、STM32开发环境搭建
  - 2.1、STM32CubeIDE简介
  - 2.2、下载安装包
  - 2.3、开始安装
  - 2.4、新建工程
  - 2.5、引脚配置
  - 2.6、编写代码
  - 2.7、编译程序
  - 2.8、烧录程序
  - 2.9、程序现象

### 2.1、STM32CubeIDE简介

STM32CubeIDE是一体式多操作系统开发工具，是一种高级C/C++开发平台，具有STM32微控制器和微处理器的外设配置、代码生成、代码编译和调试功能。它基于Eclipse®/CDT™框架和用于开发的GCC工具链，以及用于调试的GDB，可选择一个STM32微处理器然后创建项目并生成初始化代码，支持图形化配置STM32的时钟和引脚等内容，支持Windows、Linux、Mac主流平台，功能非常强大实用。

### 2.2、下载安装包

用电脑浏览器打开以下链接：

<https://www.st.com/zh/development-tools/stm32cubeide.html>

根据电脑系统下载，这里以Win10系统安装1.9.0版本为例，选择最新版本下载。

#### 获取软件

产品型号	一般描述	Latest version	下载	All versions
+ STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	1.9.0	Get latest	选择版本
+ STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	1.9.0	Get latest	选择版本
+ STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	1.9.0	Get latest	选择版本
+ STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	1.9.0	Get latest	选择版本
+ STM32CubeIDE-Win	STM32CubeIDE Windows Installer	1.9.0	Get latest	选择版本

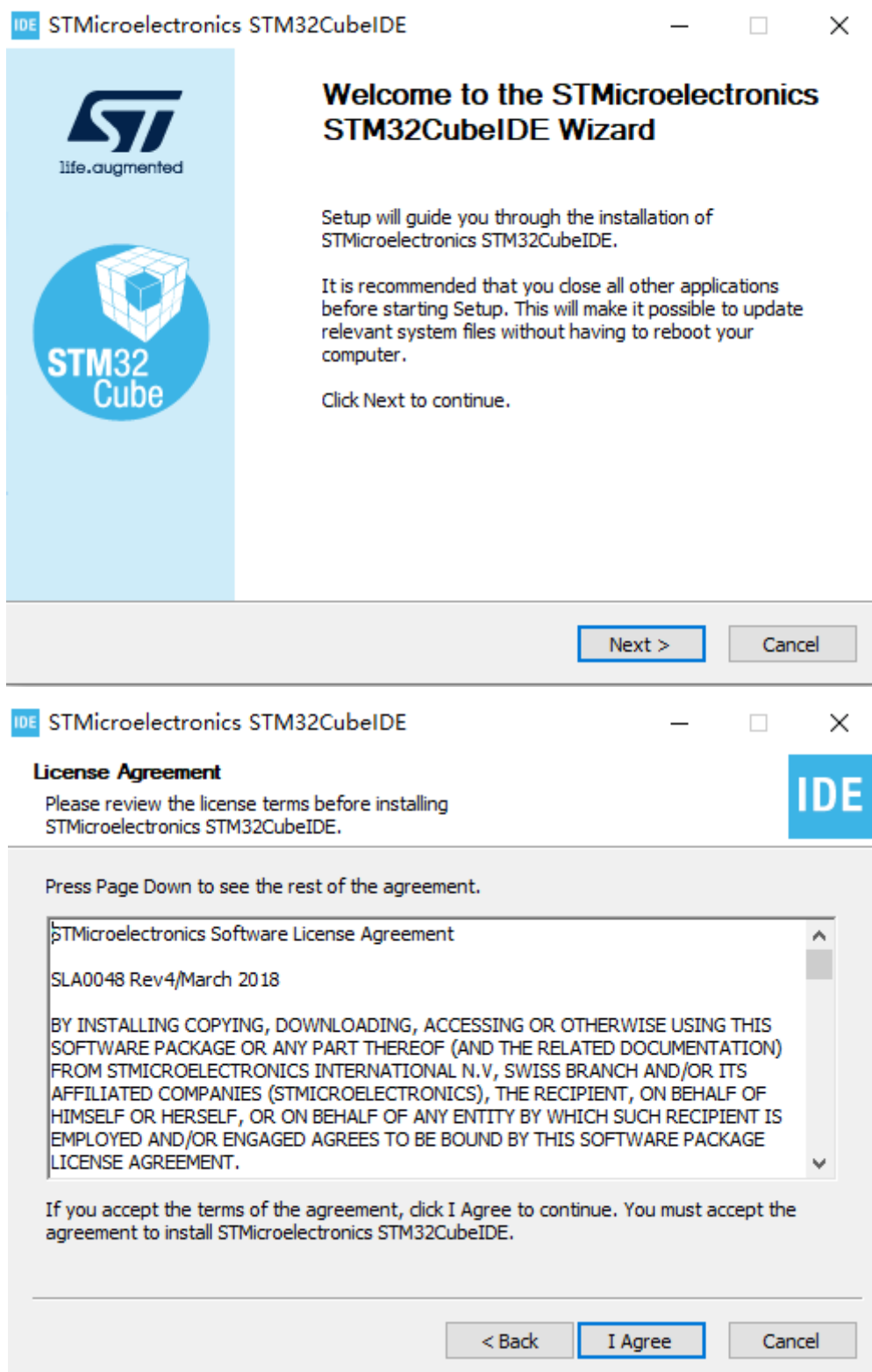
如果您在my.st.com上有帐户，即可直接登录并下载软件。如果您现在不想现在登录，只需要在下面的表单中提供您的姓名和电子邮件地址，就可以下载软件。

将下载到的文件解压出来，解压路径不要有中文。

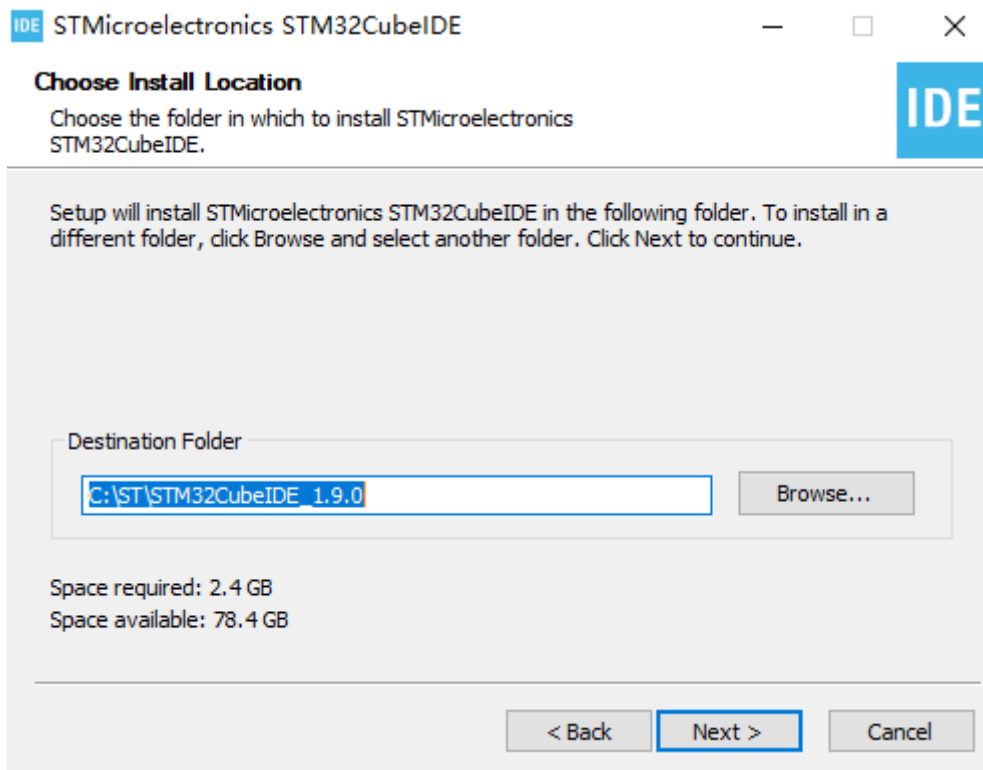
st-stm32cubeide\_1.9.0\_12015\_202203..

## 2.3、开始安装

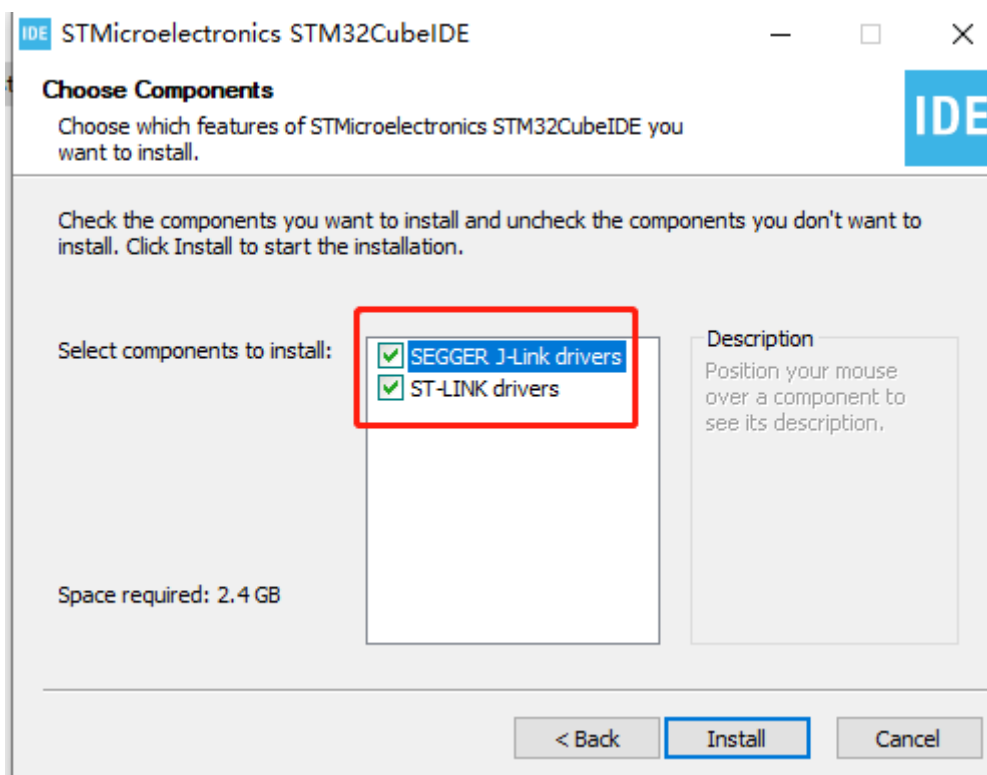
双击打开安装包。然后跟着教程操作就可以。



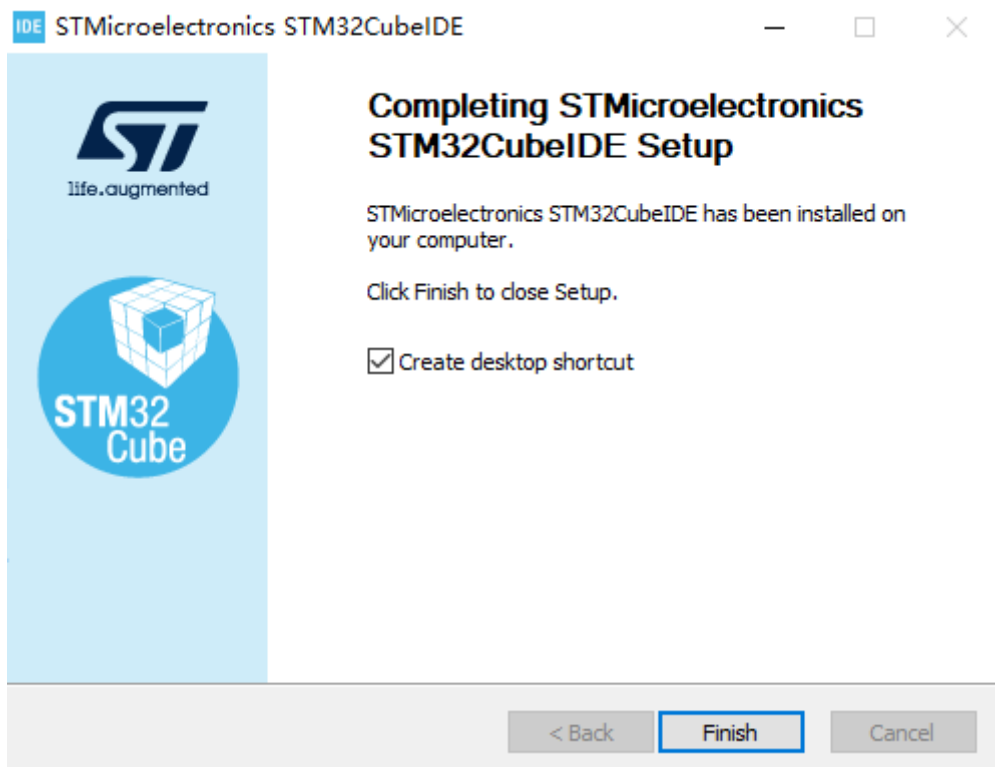
安装路径可根据实际自行修改，注意不要有中文。



勾选驱动，然后再点击Install安装。

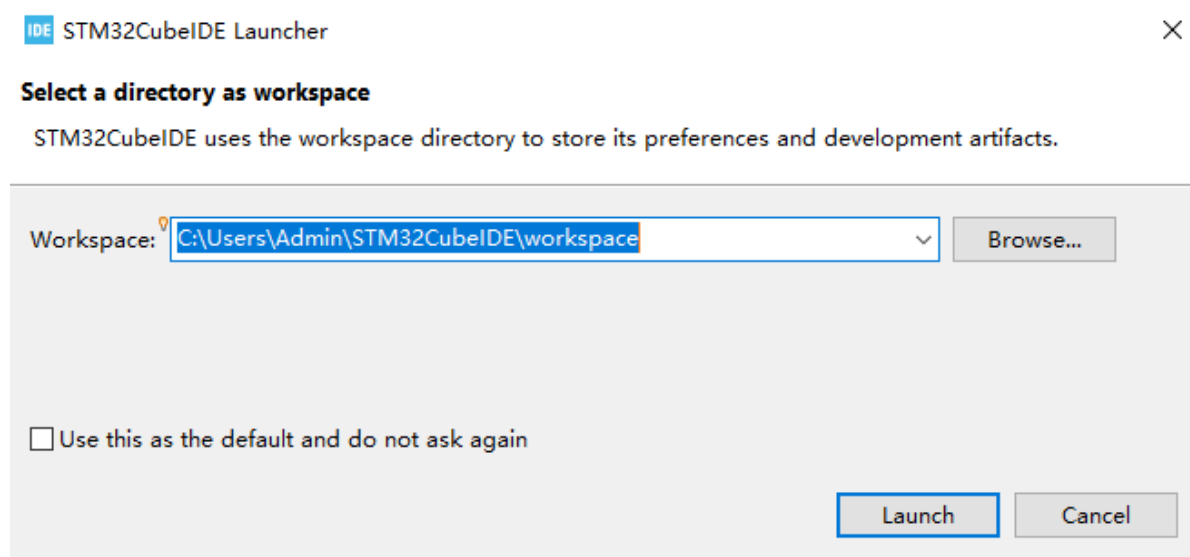
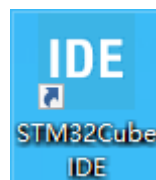


接下来等待安装完成即可。



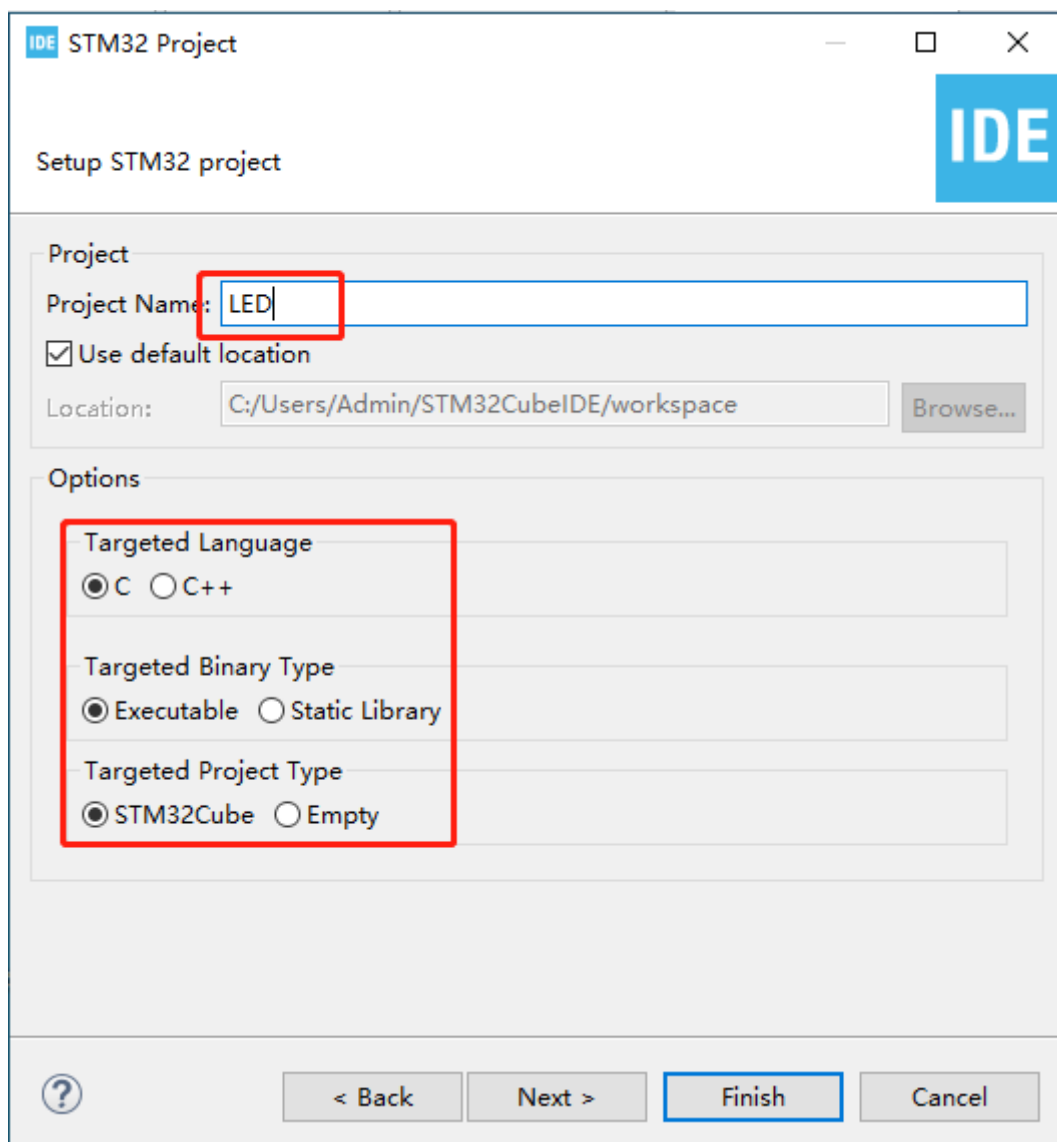
## 2.4、新建工程

1.双击桌面的快捷方式打开STM32CubeIDE，需要选择一下工作空间，保存路径可以根据实际选择其他路径（不要带中文）。

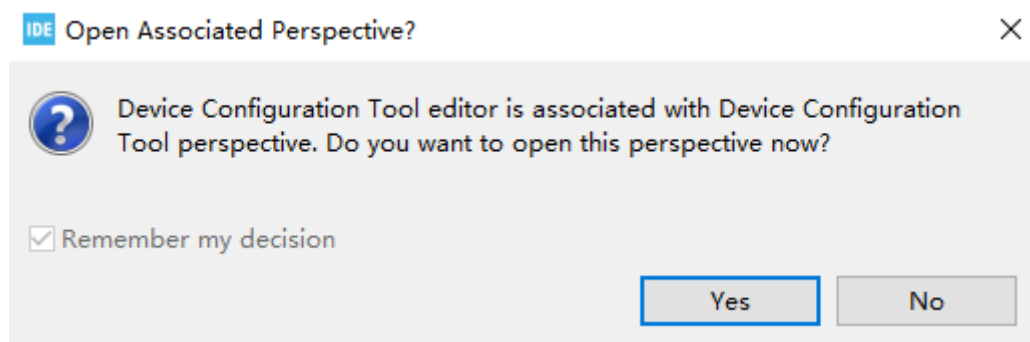


2.点击File->New->STM32 Project.

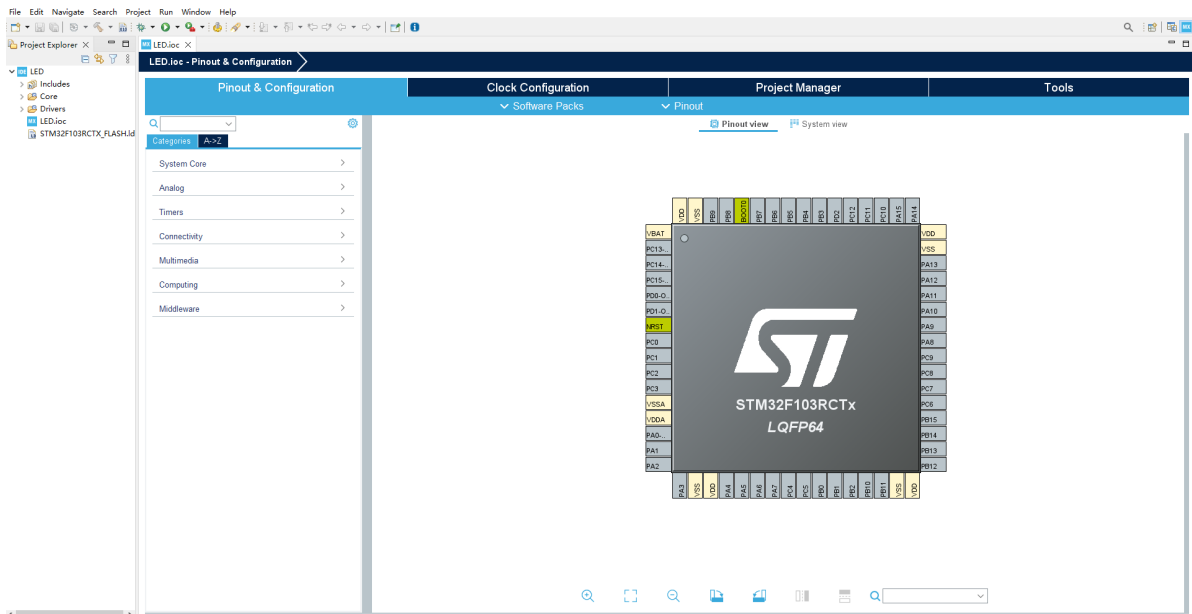




5.点击Yes，此时会加载图形化内容。

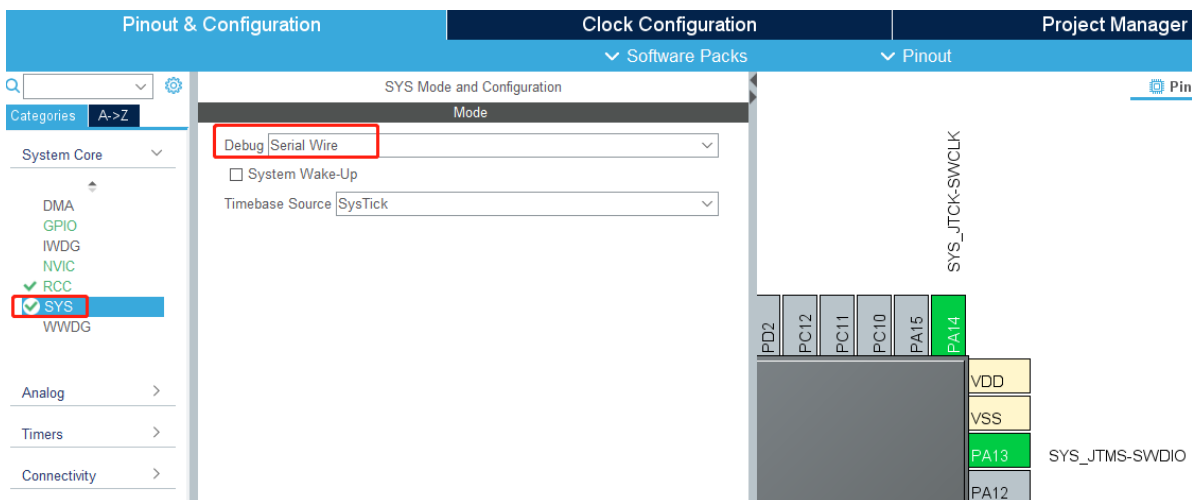


完成后如下图所示：

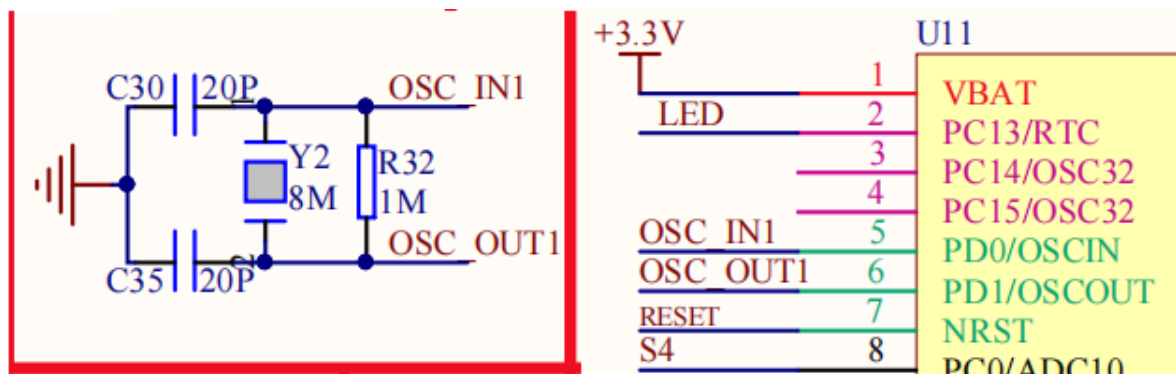


## 2.5、引脚配置

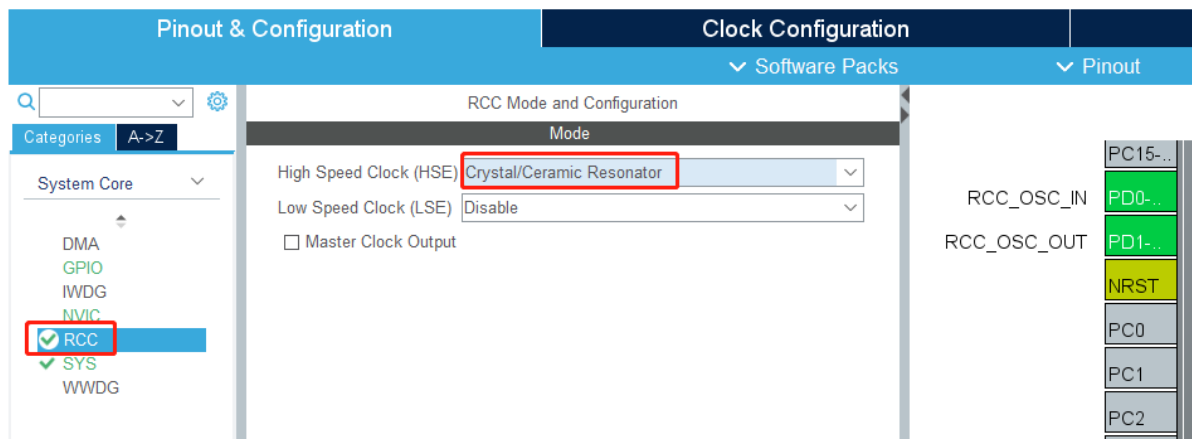
1.首先需要Debug信息，在Pinout&Configuration下点击SYS->Debug选择Serial Wire。



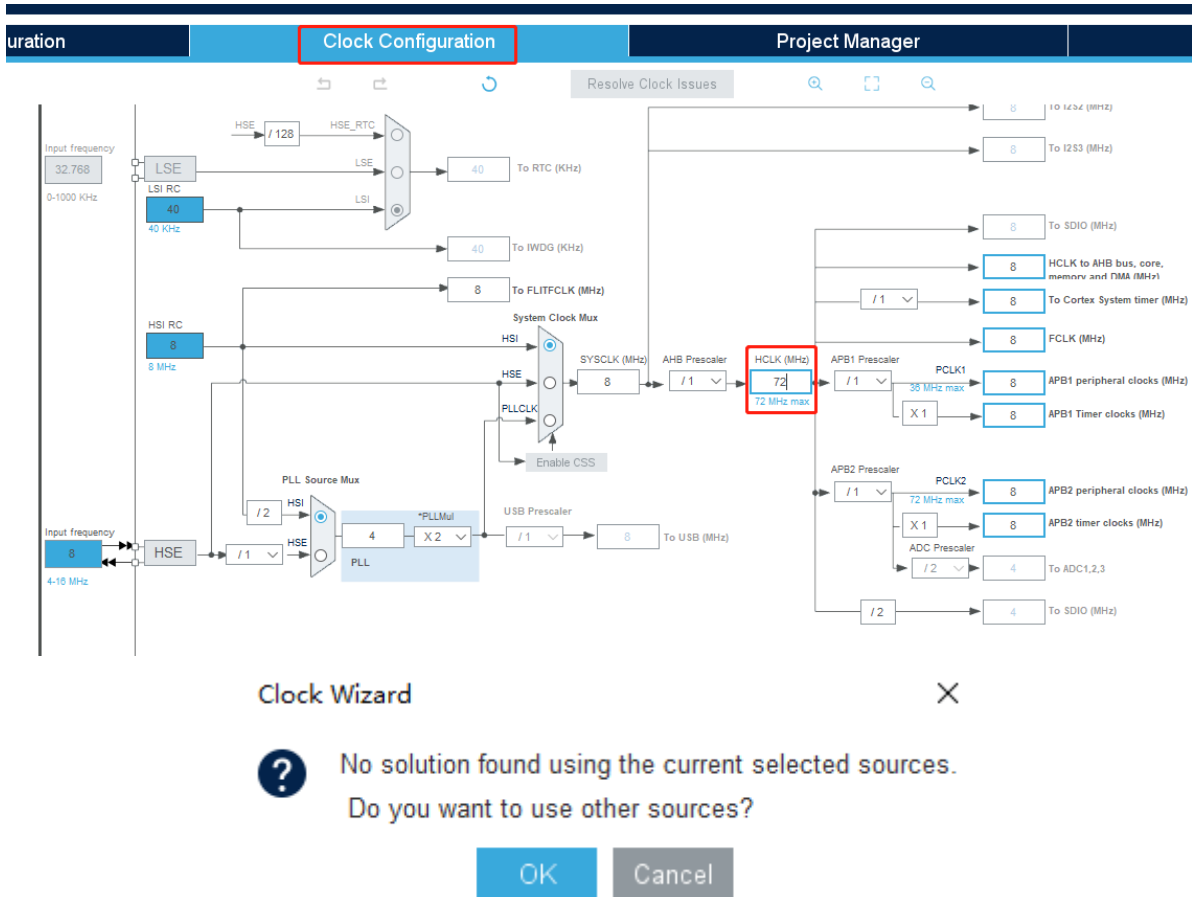
2.修改STM32的系统时钟，根据原理图可知，外部晶振为8M频率。



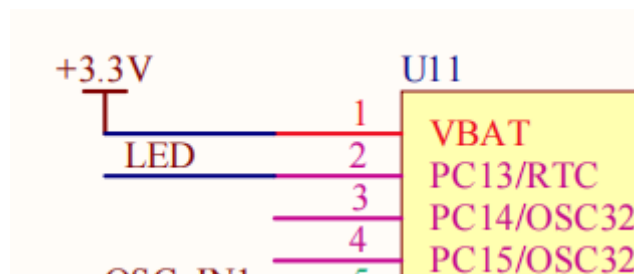
在Pinout&Configuration选择RCC->HSE选择Crystal/Ceramic Resonator。其中HSE为外部时钟，LSE为内部时钟，使用外部时钟能比内部时钟更加稳定高效。



切换到Clock Configuration修改HCLK的频率为72，按回车确认。

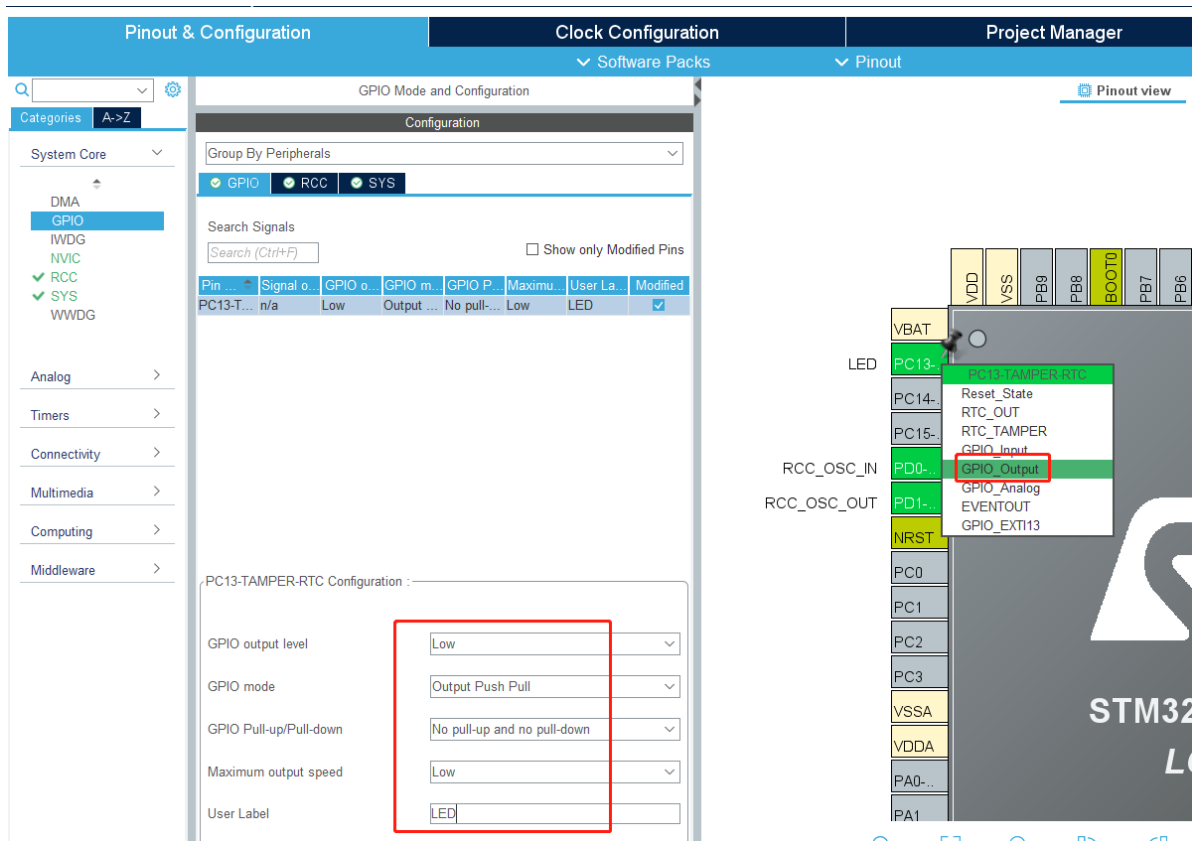


3.增加LED引脚配置，从原理图可知LED连接到PC13引脚。

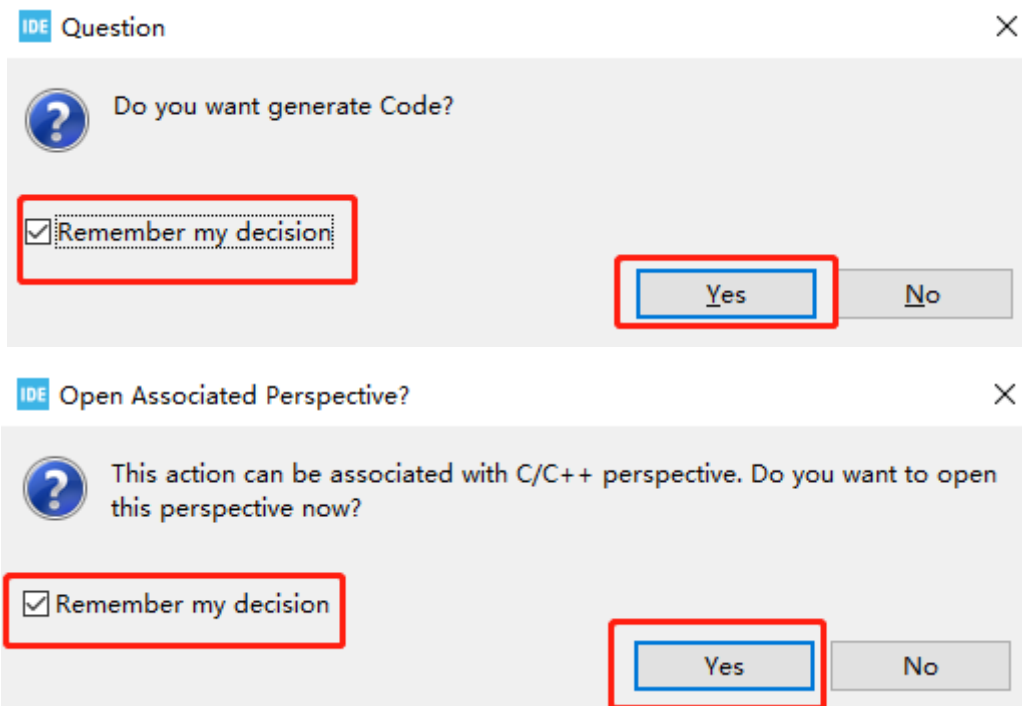


将PC13引脚设置为GPIO\_Output，为了方便这里修改Label为LED。





然后按Ctrl+S保存，将Remember my decision打钩，然后点击Yes。这样每次保存都会自动生成代码。



## 2.6、编写代码

1.由于上一步图形化配置已经生成系统的初始化代码，所以我们只需要添加要实现的功能就可以了。

找到main.c文件里的main函数，在while（1）下添加控制LED灯的内容。功能是LED灯每隔两百毫秒闪一次。按Ctrl+S保存代码。

```

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, SET);
    HAL_Delay(200);
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, RESET);
    HAL_Delay(200);
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

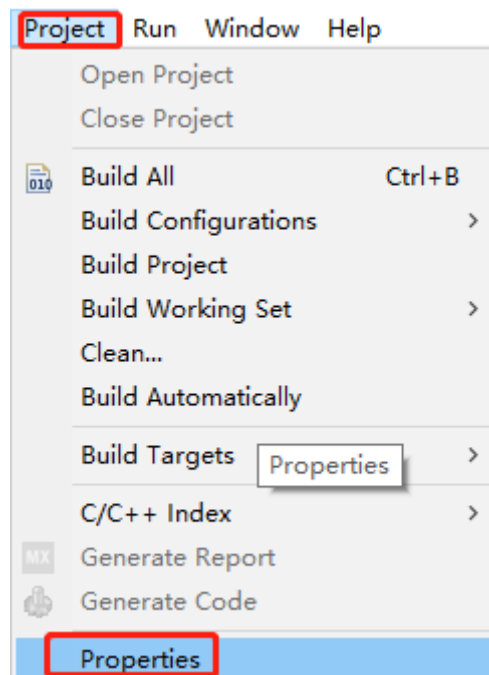
```

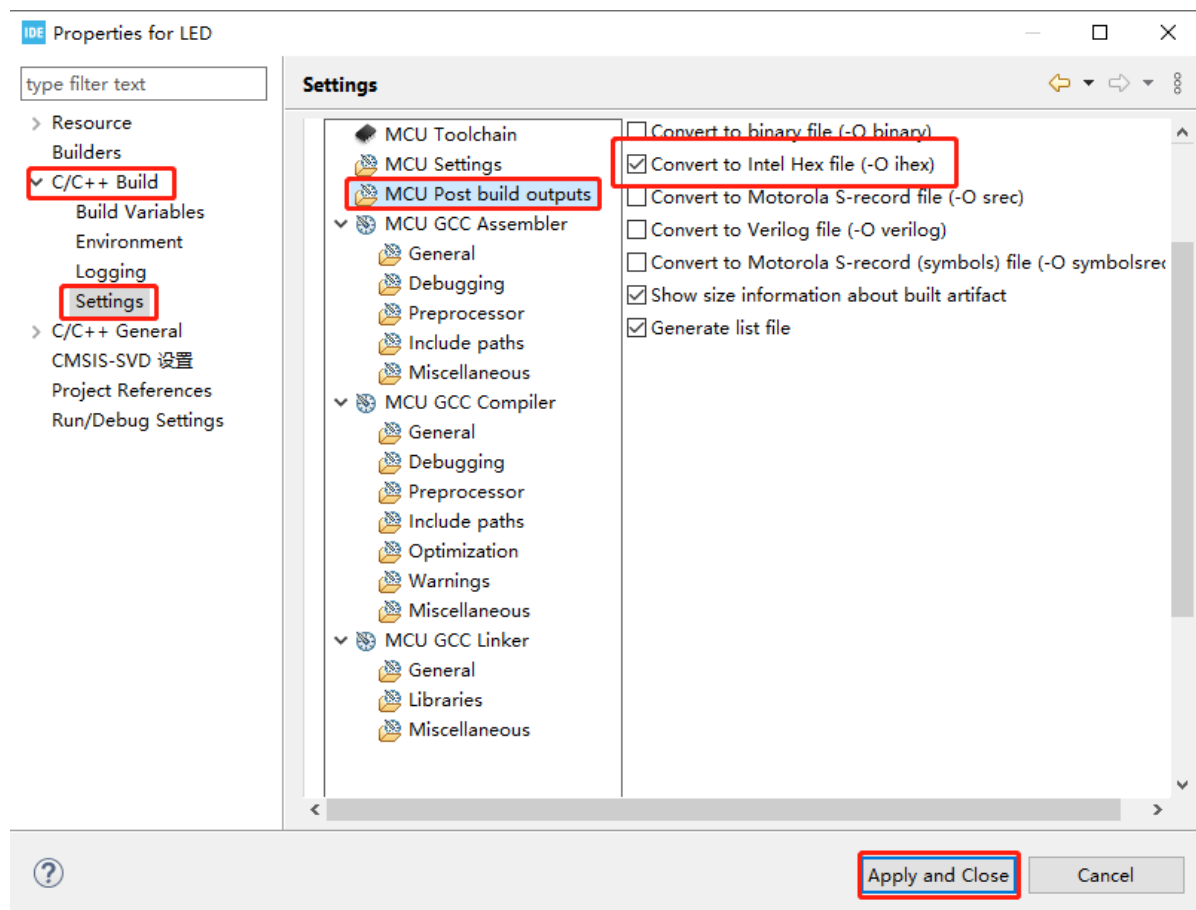
注意：代码内容需要添加在USER CODE BEGIN 和USER CODE END之间的。否则下次通过图形化工具生成代码后，代码内容会被覆盖掉，添加在USER CODE BEGIN 和USER CODE END之间的则不会被覆盖掉。不要写中文注释在里面，可能会出现乱码。

## 2.7、编译程序

1.添加生成HEX文件的功能。

依次点击Project->Properties->C/C++ Build->Settings->MCU Post build outputs, 然后在Convert to Intel Hex file(-O ihex)前打勾，如下图所示。

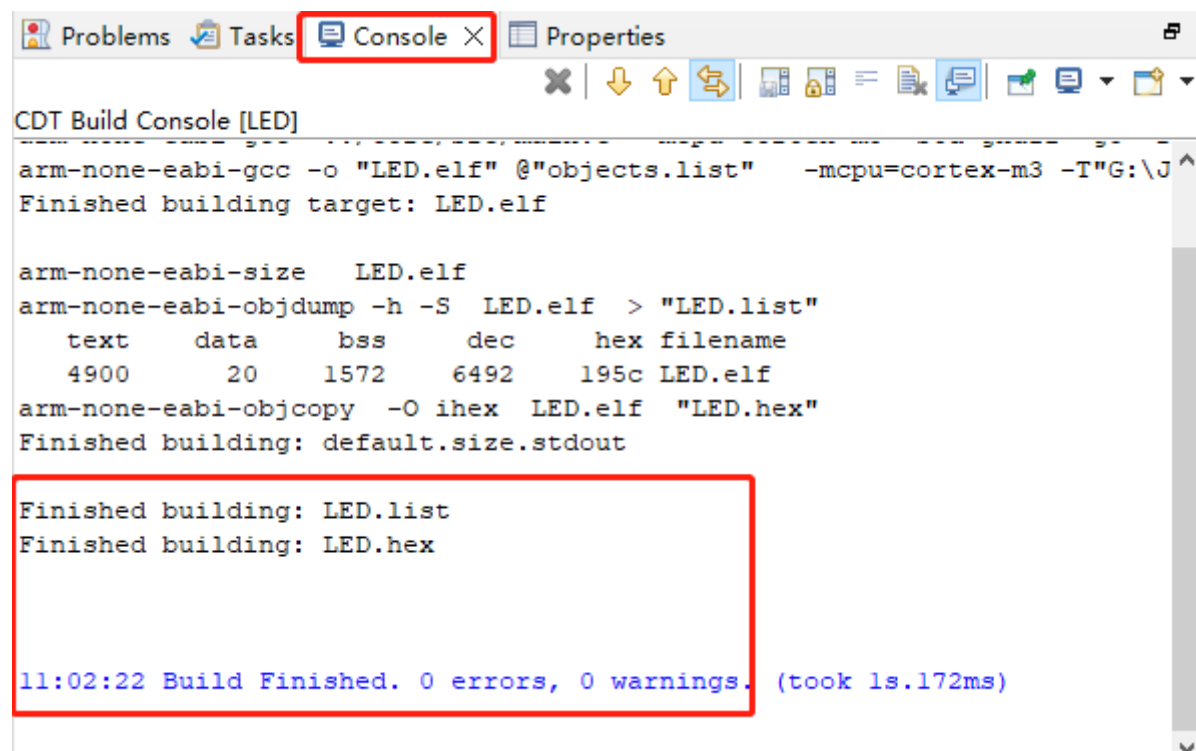




2. 点击工具栏中的锤子，开始编译工程。



STM32CubeIDE会弹出Console控制台，看到编译0错误，0警告就表示编译成功。如下图所示，项目生成的文件名为LED.hex，此文件保存在工程目录的Debug文件夹中。

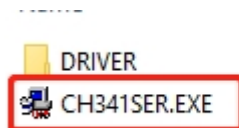


## 2.8、烧录程序

### 1.安装CH340驱动

由于Rosmaster扩展板的USB通讯使用了CH340芯片，所以需要安装CH340芯片的驱动，如果电脑已经安装过CH340驱动则不需要重复安装。

解压本课程资料中的【Uart drive (CH340) .zip】，双击打开CH341SER.EXE程序



点击安装，安装完成后会提示安装成功。

### 2.下载烧录软件

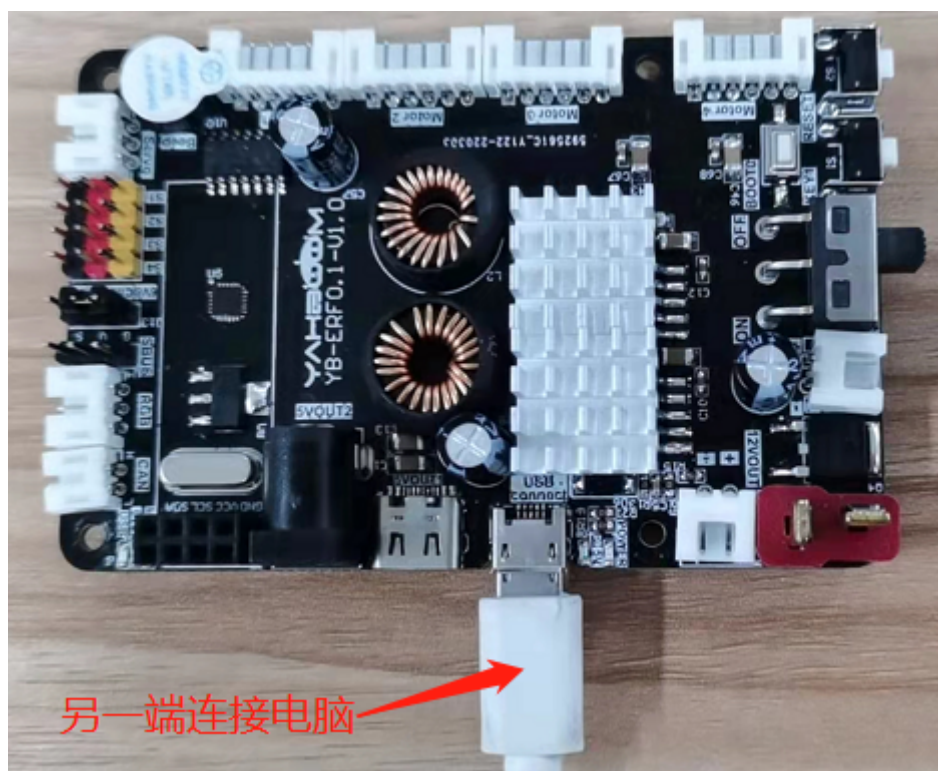
本次烧录Rosmaster扩展板单片机固件需要用到mcuisp（或flymcu）烧录软件，请到<http://www.mcuisp.com>网站下载mcuisp（或flymcu）烧录软件；也可以直接使用资料中提供的mcuisp软件。

mcuisp软件是绿色版软件，不需要安装，双击打开即可使用。

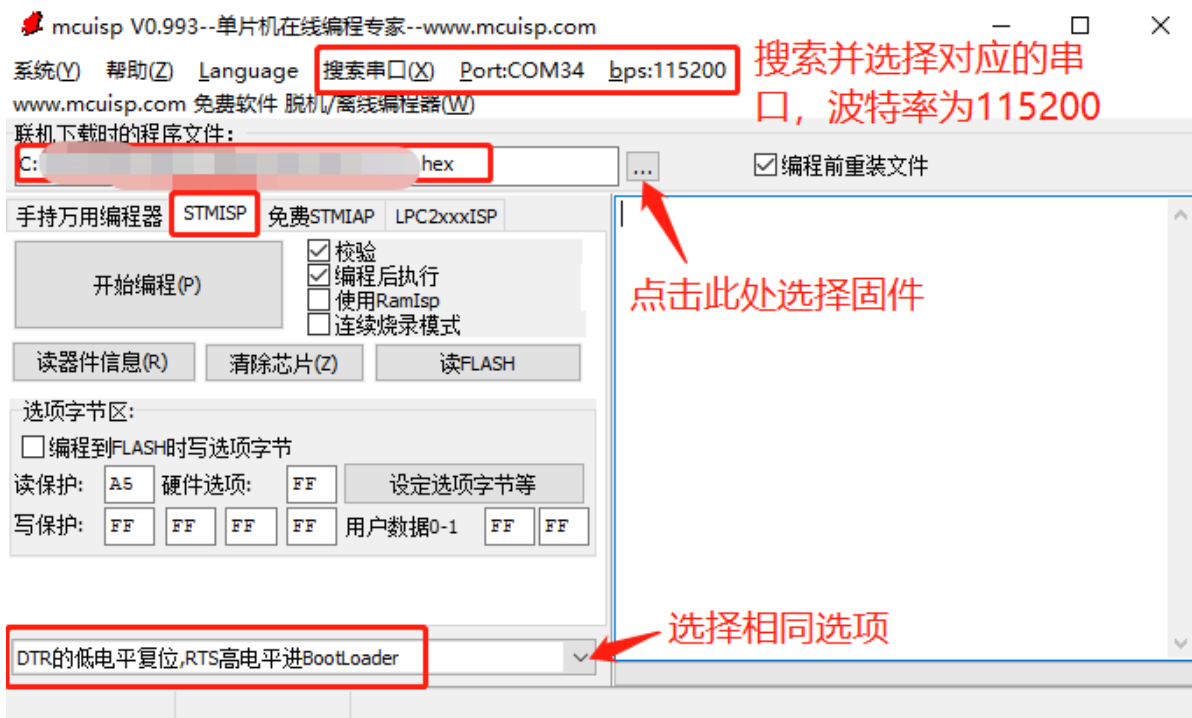
### 3.连接设备

在电脑连接Rosmaster之前，请将扩展板连接Jetson Nano的Micro USB数据线和电源线拔掉。

将USB数据线的一端插入电脑USB接口，另一端插入Rosmaster扩展板的Micro USB接口。



### 4.配置烧录软件



搜索串口的时候如果出现多个串口号, 又不确认哪个才是Rosmaster的。解决方法一: 拔掉其他USB接口再搜索; 解决方法二: 先拔掉Rosmaster的USB数据线, 点击搜索串口, 记下搜索到的串口号, 插入Rosmaster的USB数据线, 再次搜索串口, 对比前后两次, 新增加的串口号就是Rosmaster的串口号。

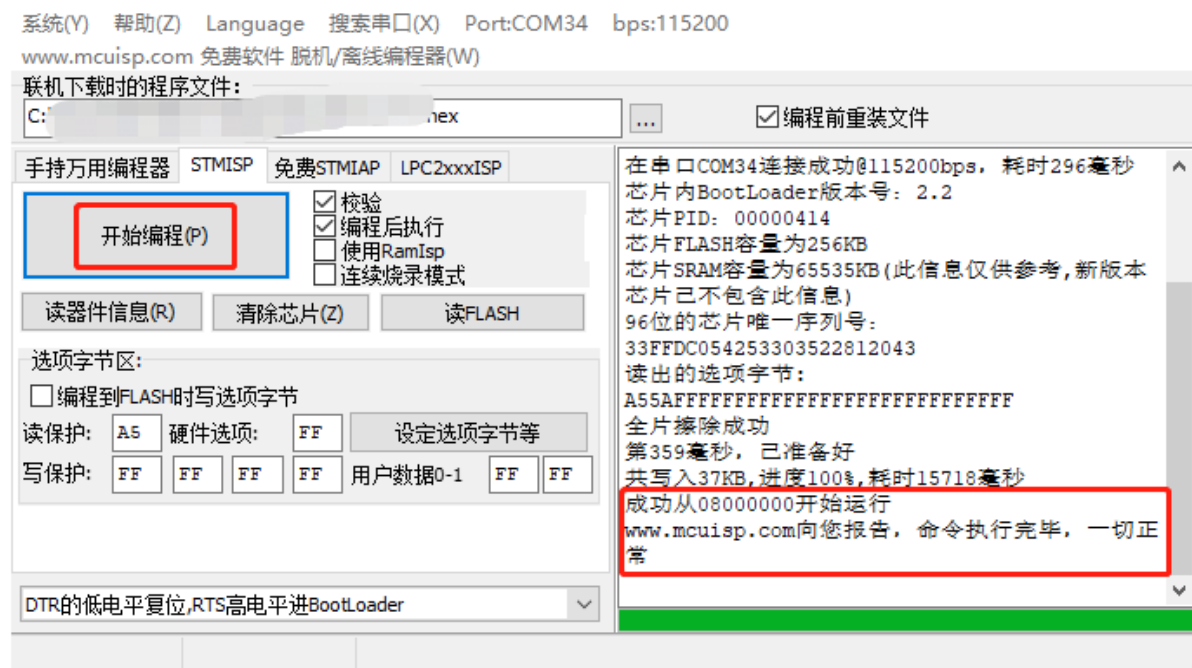
选择固件时需要选择工程目录的Debug文件夹下的LED.hex文件。

最后是底部的配置选择, 一定要选择【DTR的低电平复位, RTS高电平进BootLoader】选项, 否则可能导致下载失败。

### 3.烧录程序

请先将扩展板上的单片机进入烧录模式:

先按住扩展板上的BOOT0键, 再按一下RESET键, 最后松开BOOT0键。



点击【开始编程】, mcuisp烧录软件会将我们上一步选择的固件烧录到Rosmaster的扩展板上的单片机上。当右边出现【成功从08000000开始运行[www.mcuisp.com](http://www.mcuisp.com)向您报告, 命令执行完毕, 一切正常】的提示就表示下载成功。

注意:

①开始烧录前，请确认Rosmaster的串口号是可访问的，即没有串口助手占用它。

②Rosmaster进入烧录模式的操作，先按住扩展板上的BOOT0键，再按一下RESET键，最后松开BOOT0键。

## 2.9、程序现象

扩展板上的LED灯每隔200毫秒闪一次。