

STM32CubeIDE 入门教程

1.简介：

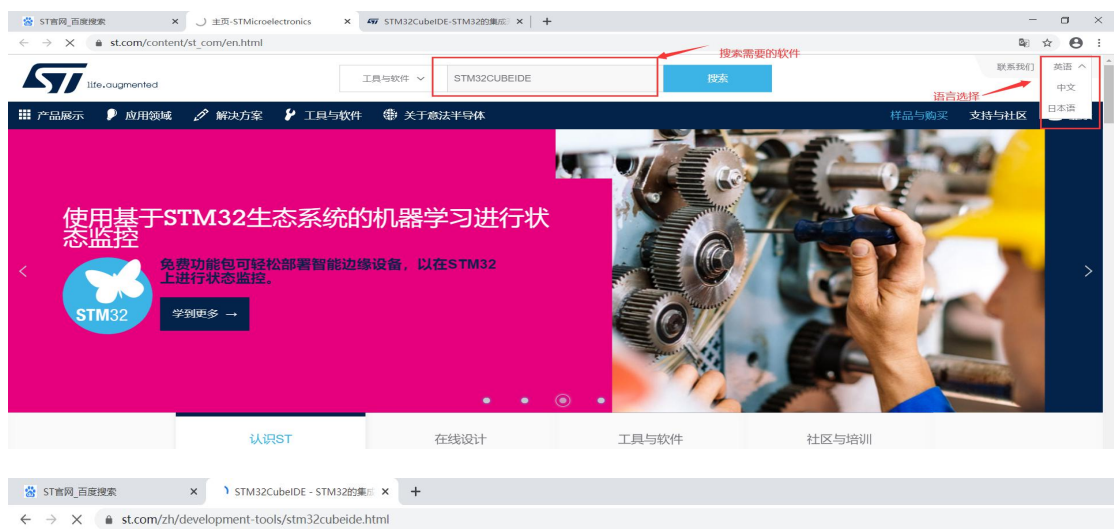
STM32CubeIDE 是一个高级 C / C ++开发平台，具有用于 STM32 微控制器和微处理器的外设配置，代码生成，代码编译和调试功能。它基于 Eclipse® / CDT 框架和 GCC 工具链进行开发，并基于 GDB 进行调试。它允许现有几百插件是完成了 Eclipse 的功能整合® IDE。

STM32CubeIDE 集成了 STM32CubeMX 的 STM32 配置和项目创建功能，以提供多合一的工具体验，并节省安装和开发时间。从选择板子或示例中选择空的 STM32 MCU 或 MPU 或预配置的微控制器或微处理器后，将创建项目并生成初始化代码。在开发过程中的任何时候，用户都可以返回外围设备或中间件的初始化和配置，并重新生成初始化代码，而不会影响用户代码。

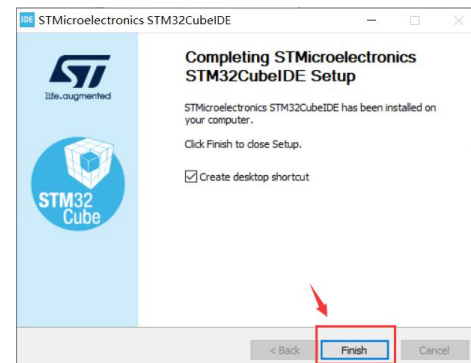
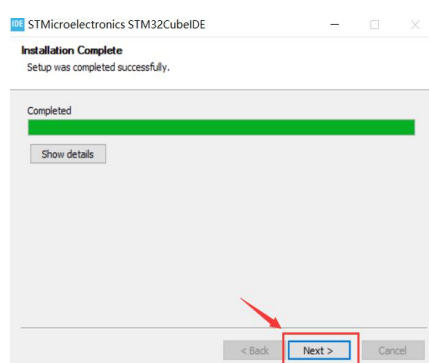
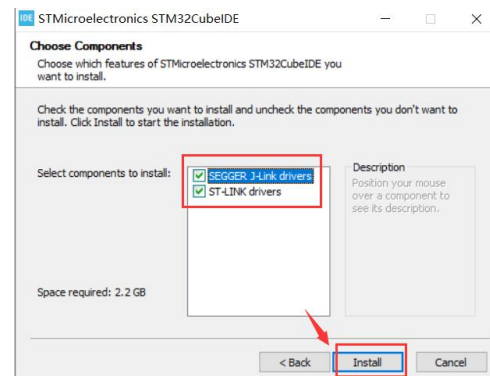
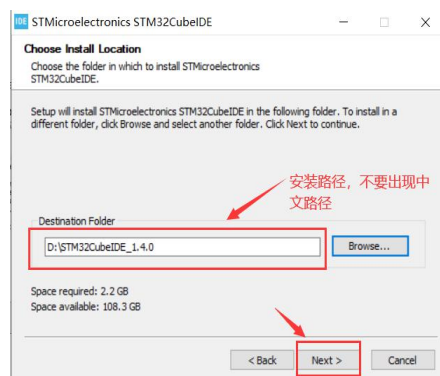
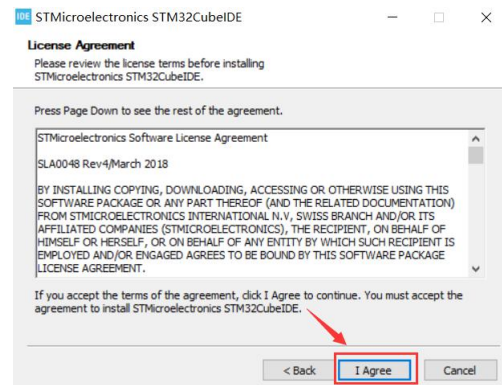
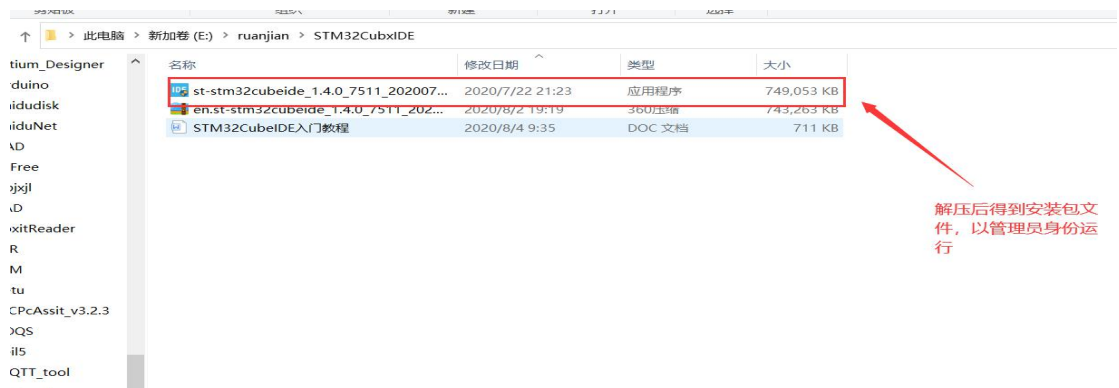
STM32CubeIDE 包括构建和堆栈分析器，可为用户提供有关项目状态和内存要求的有用信息。STM32CubeIDE 还包括标准和高级调试功能，包括 CPU 内核寄存器，存储器和外设寄存器的视图，以及实时变量监视，Serial Wire Viewer 接口或故障分析器。

2.安装教程


1.去 ST 官网下载软件安装包：https://www.st.com/content/st_com/en.html

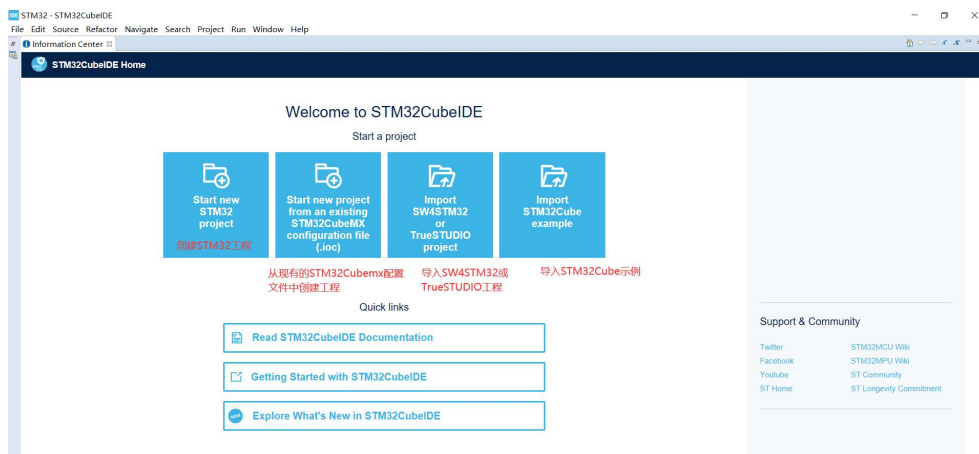
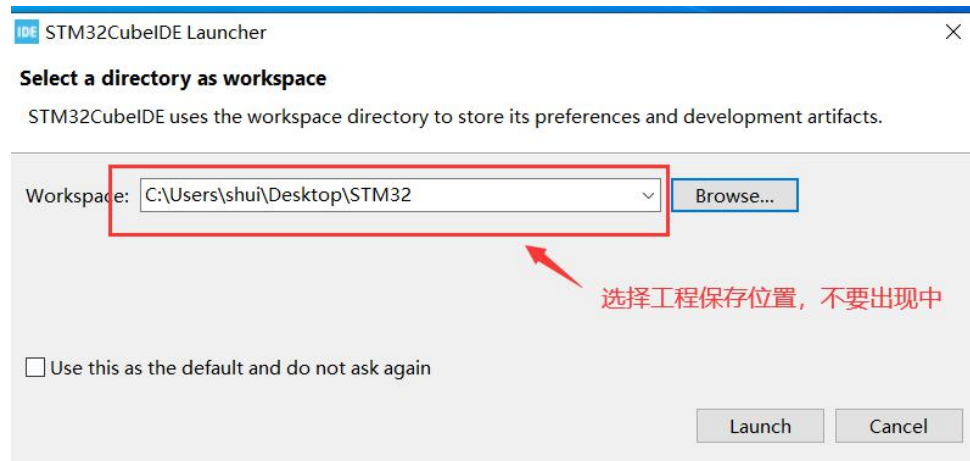


2. 软件安装:

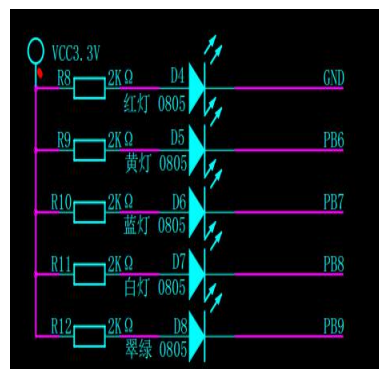


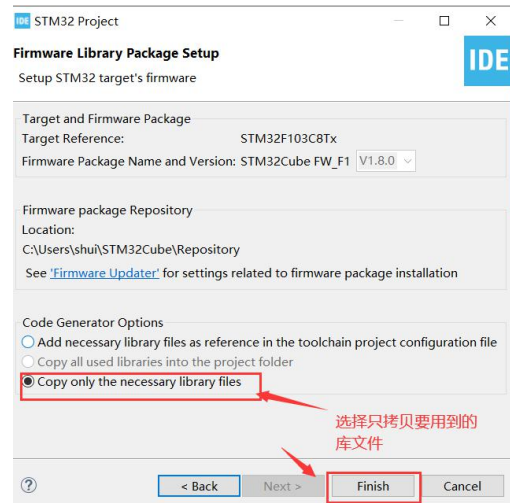
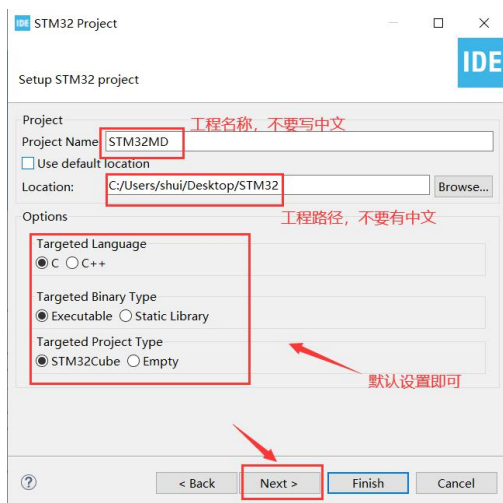
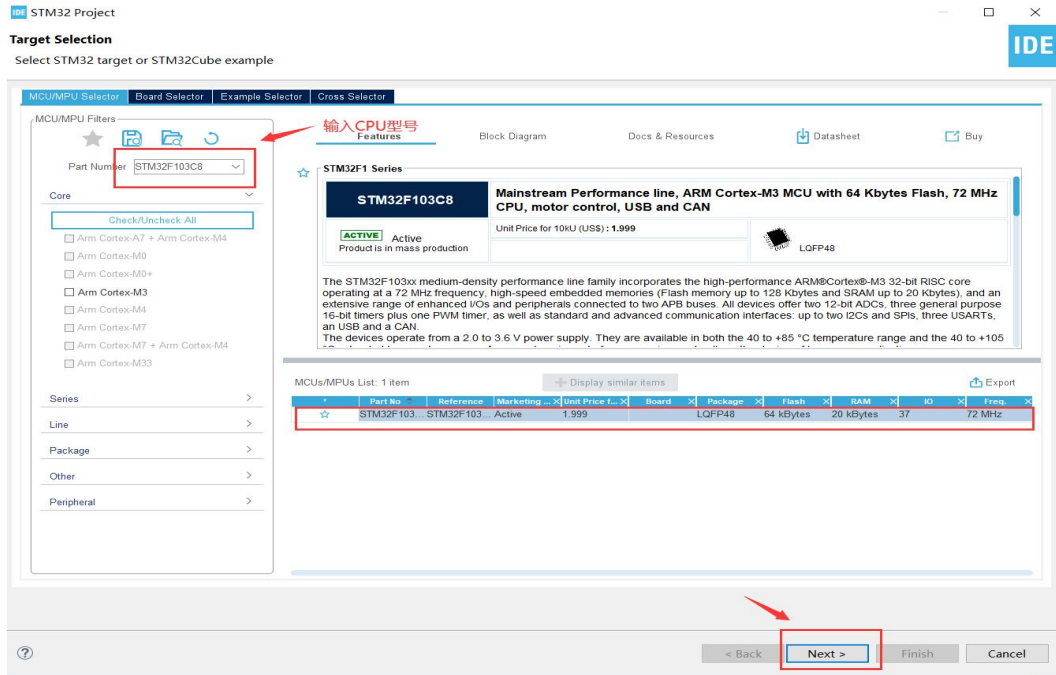
3. 创建工程

3.1 双击  打开软件，选择工作区路径，不要出现中文名称

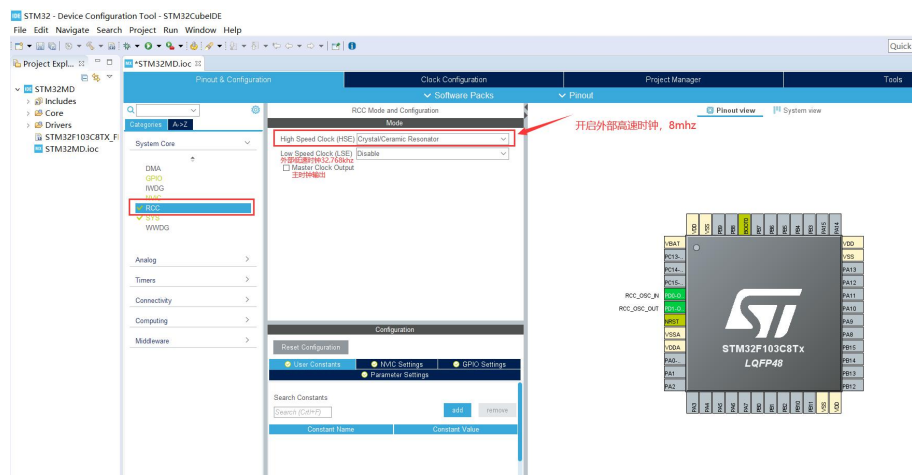


3.2 选择直接创建 STM32 工程，我们这里以 STM32F103C8 为例驱动 LED 灯。

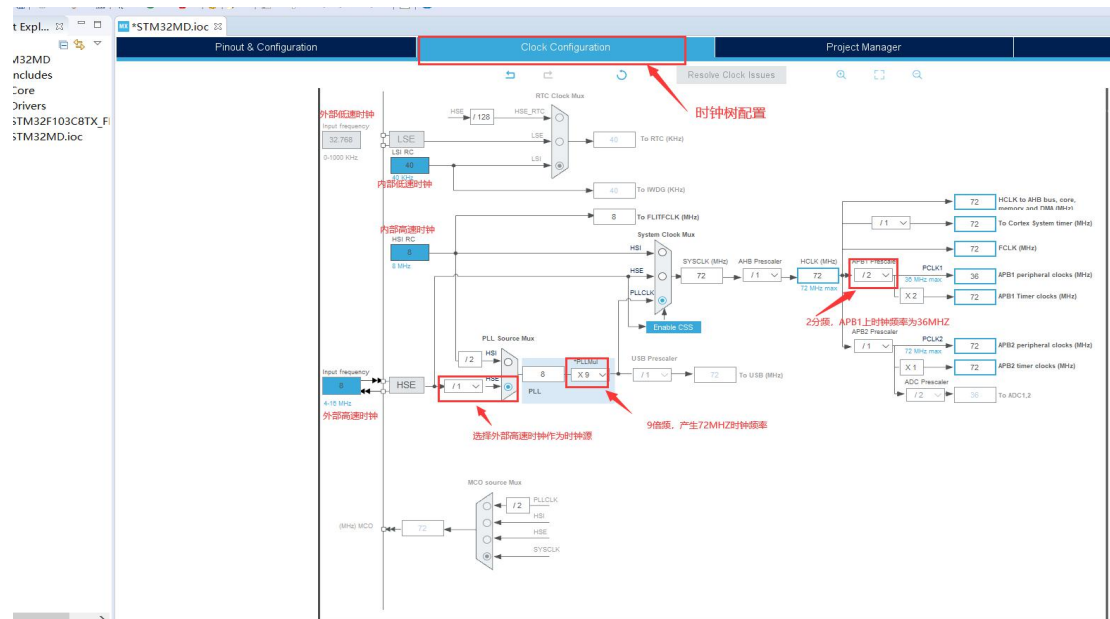




3.3 参数配置时钟源选择:

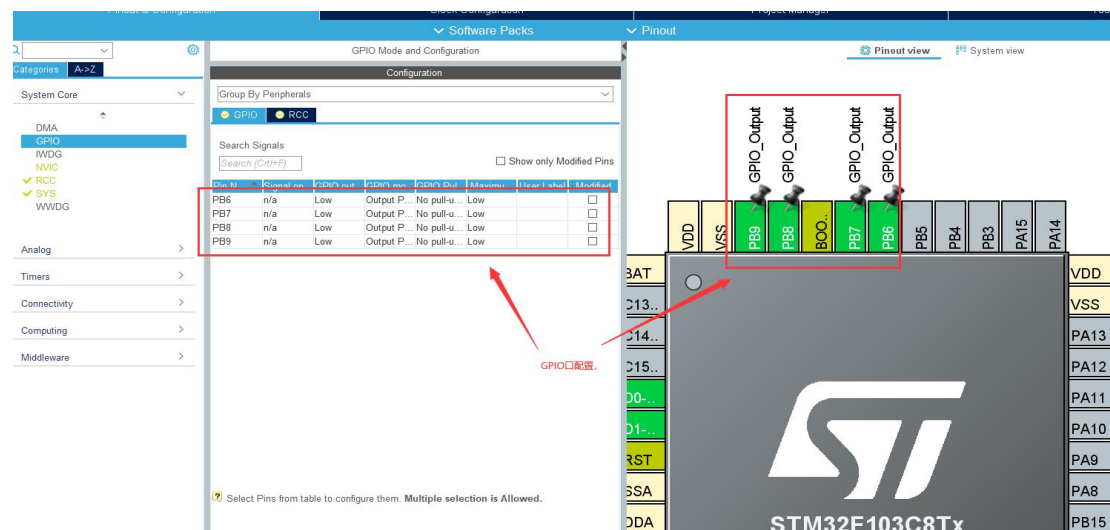


3.4 时钟树配置:



3.5 配置 GPIO 口,

LED1--PB6; LED2--PB7; LED3--PB8; LED4--PB9



3.6 GPIO 口基本参数配置, 以 PB6 为例, 其她三个配置相同。

Group By Peripherals: GPIO, RCC

Search Signals: Search (Ctrl+F)

Pin N. Signal on. GPIO out. GPIO mo. GPIO Pul. Maximum. User Label. Modified

Pin N.	Signal on.	GPIO out.	GPIO mo.	GPIO Pul.	Maximum.	User Label.	Modified
PB6	n/a	Low	Output P...	Pull-up	High	LED1	<input checked="" type="checkbox"/>
PB7	n/a	Low	Output P...	No pull-u...	Low		<input type="checkbox"/>
PB8	n/a	Low	Output P...	No pull-u...	Low		<input type="checkbox"/>
PB9	n/a	Low	Output P...	No pull-u...	Low		<input type="checkbox"/>

单击选中PB6,进行IO口参数配置

PB6 Configuration:

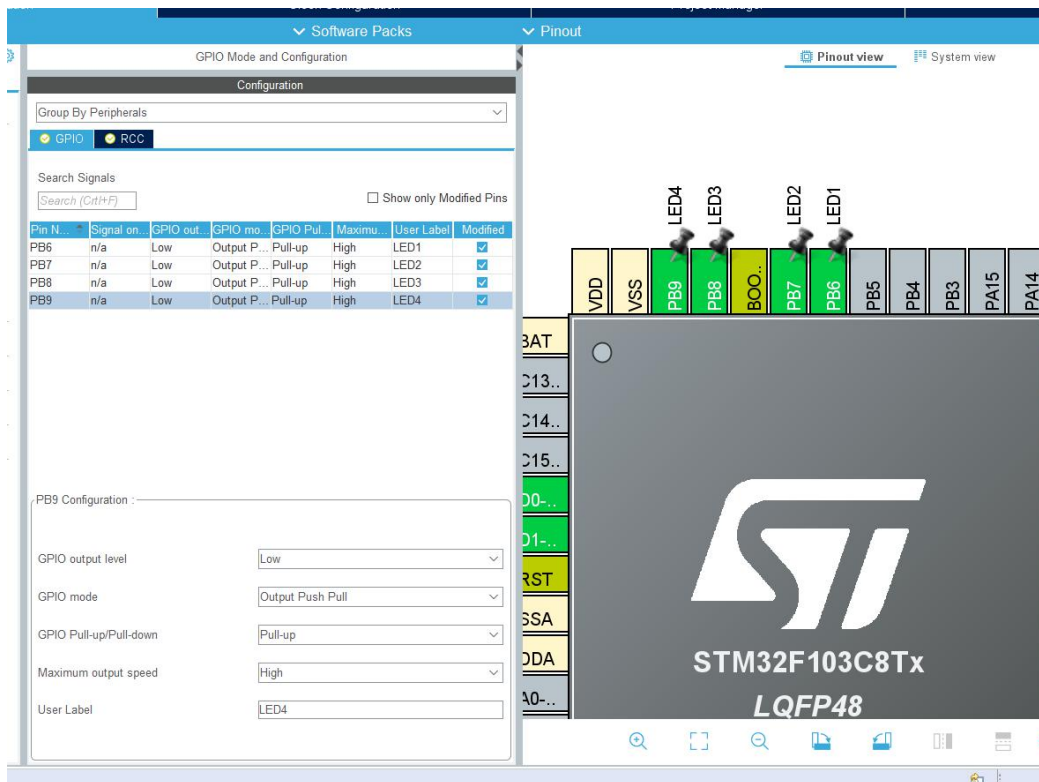
GPIO output level: Low 默认输出电平为低电平

GPIO mode: Output Push Pull 通用推挽输出模式

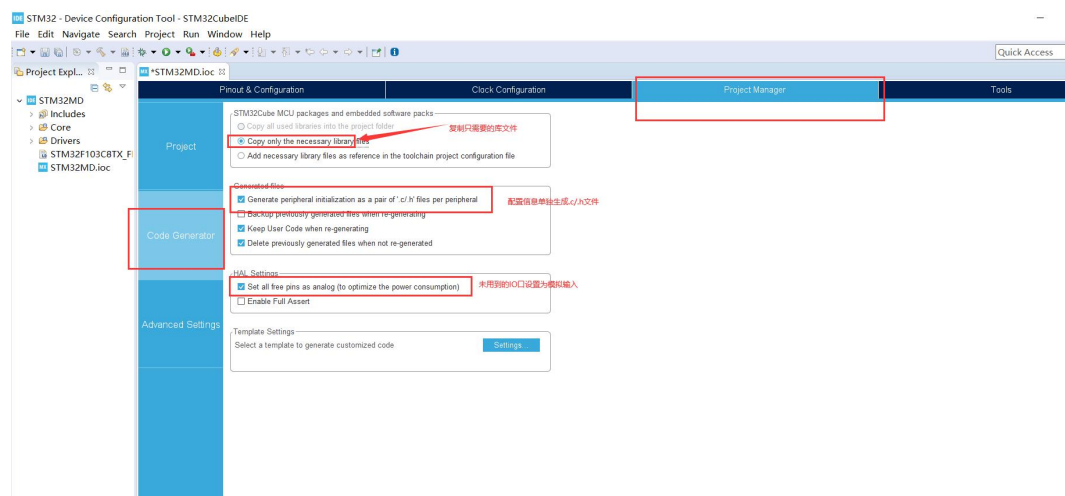
GPIO Pull-up/Pull-down: Pull-up 上拉

Maximum output speed: High IO口响应速度设置为高速

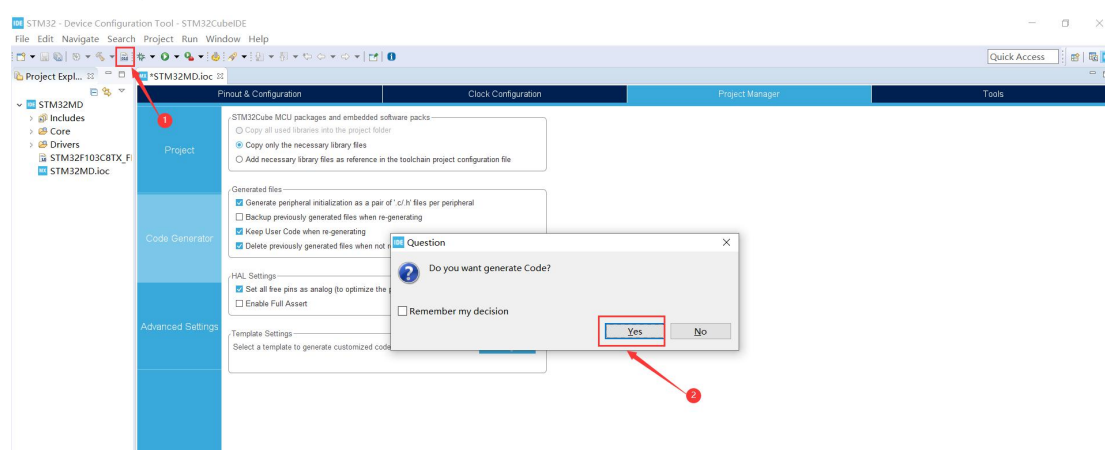
User Label: LED1 PB6自定义名称为LED1



3.7 工程参数设置:



3.8 生成代码:



3.9 实现 500ms 闪烁一次:

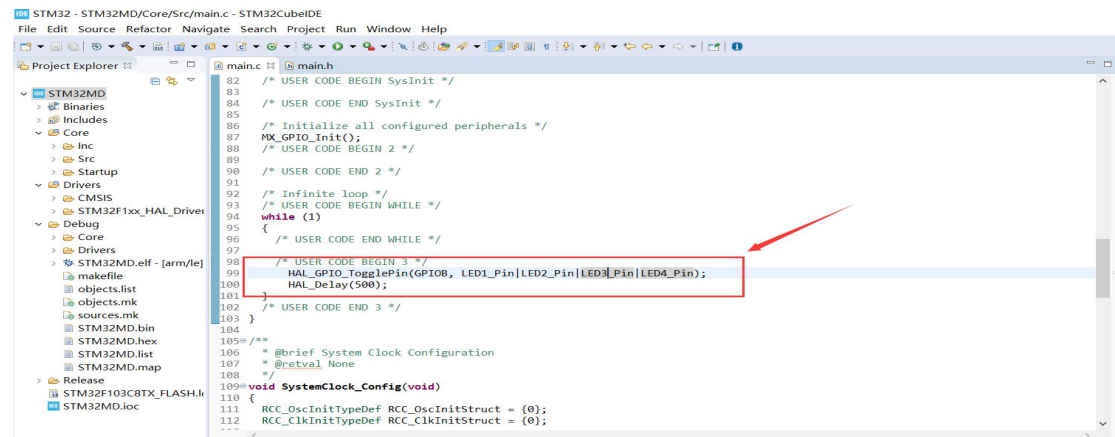
Alt+/ : 补全代码

CTRL+“-”:缩小字体

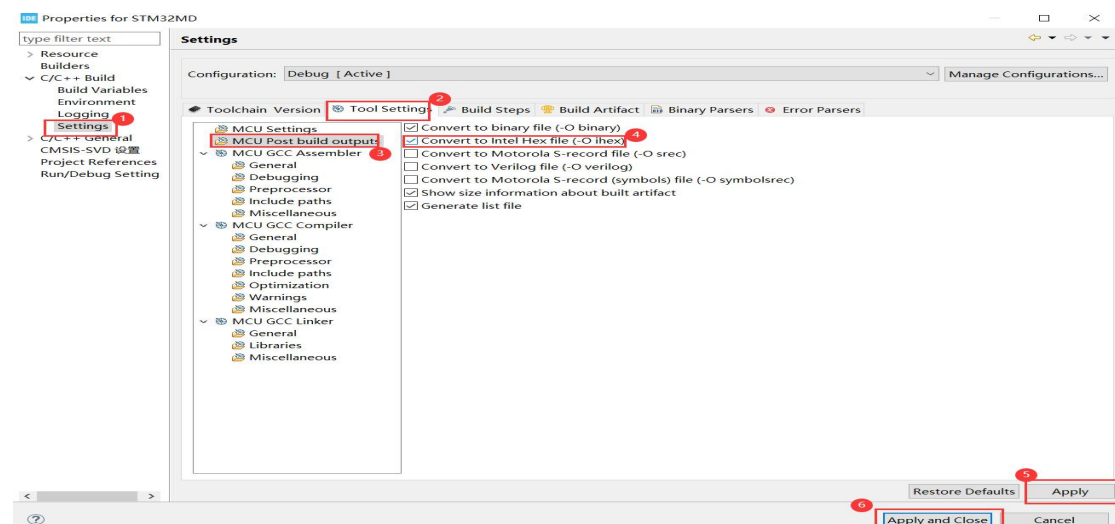
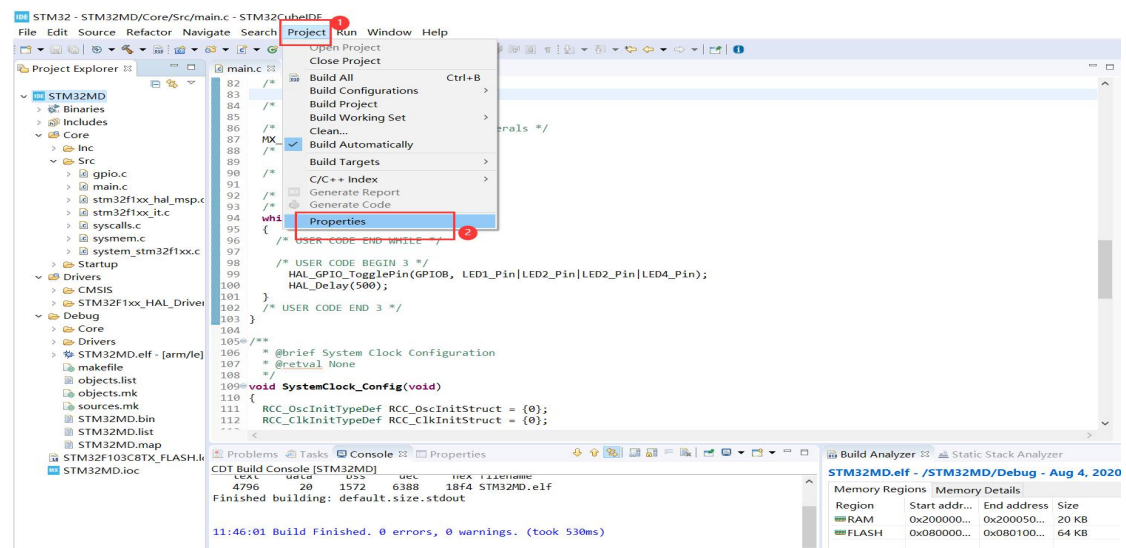
CTRL+SHIFT+“+”:放大字体

CTRL+TAB: .c 和.h 文件切换

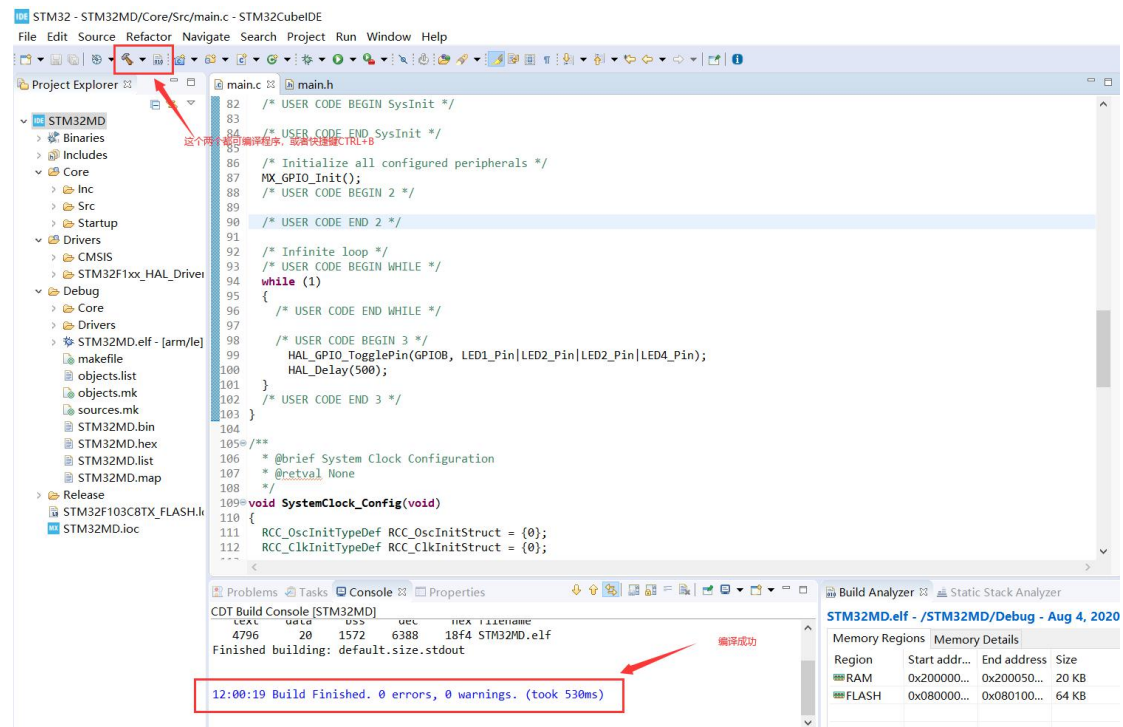
CTRL+/:注释



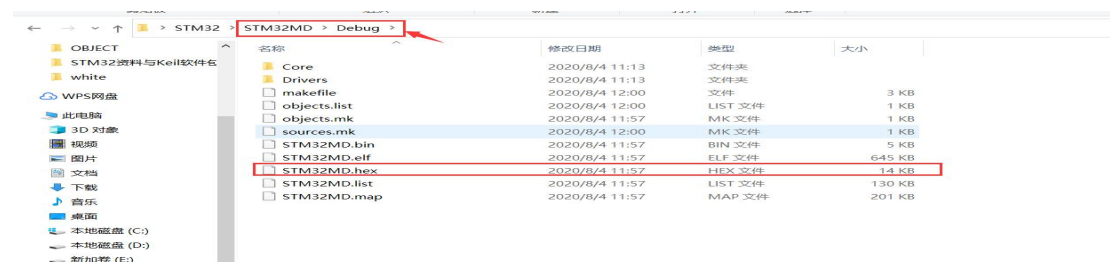
3.10 生成.hex 文件设置:



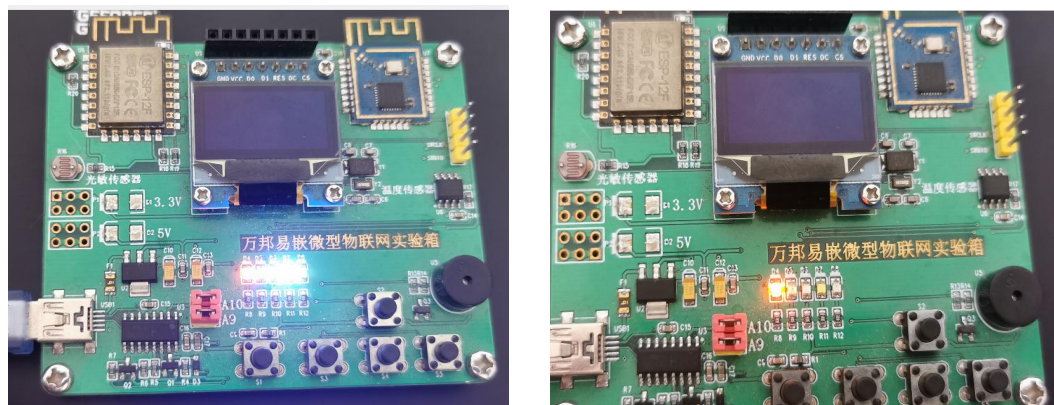
3.11 编译工程：



3.12 生成可执行文件路径：在你所保存的工程路径下的 Debug 目录下

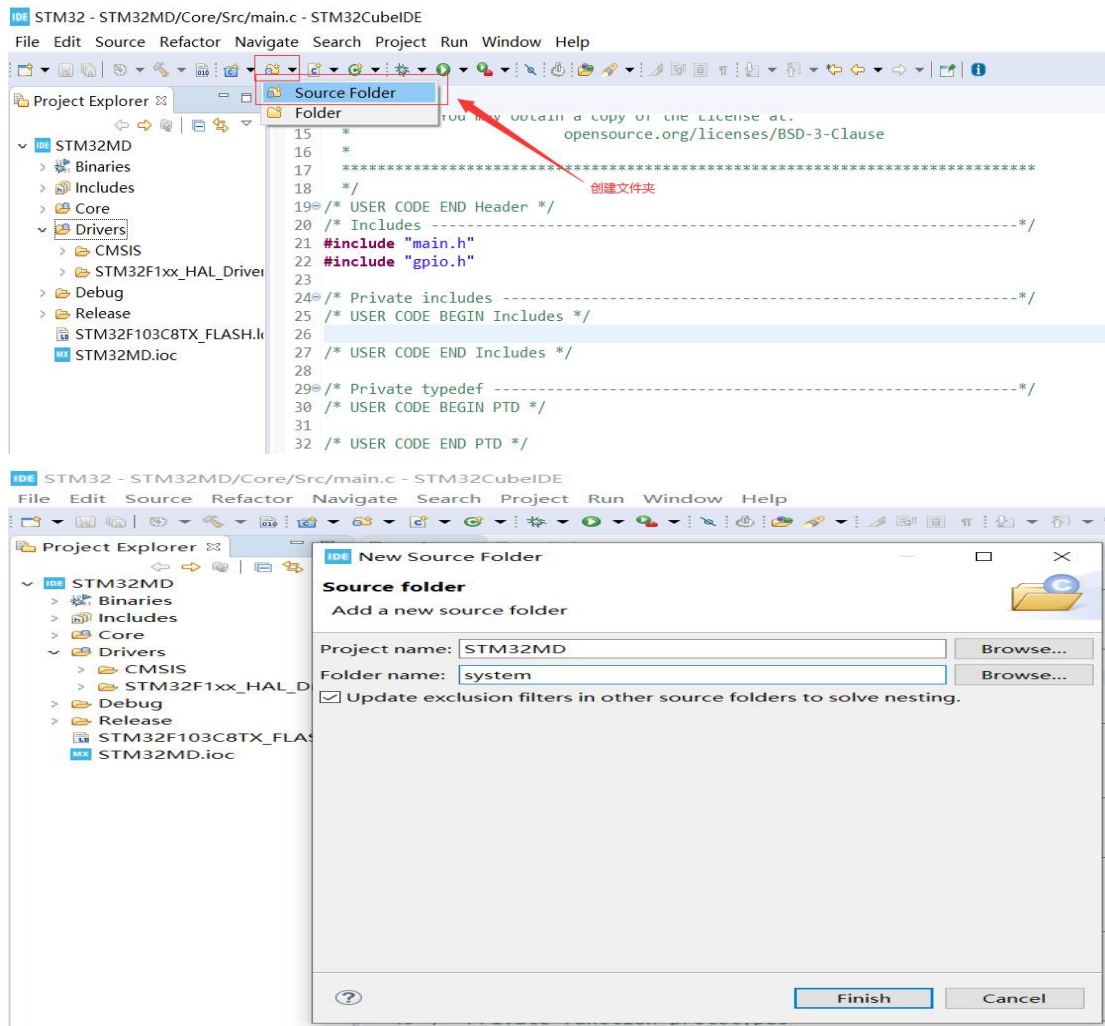


3.13 效果展示：

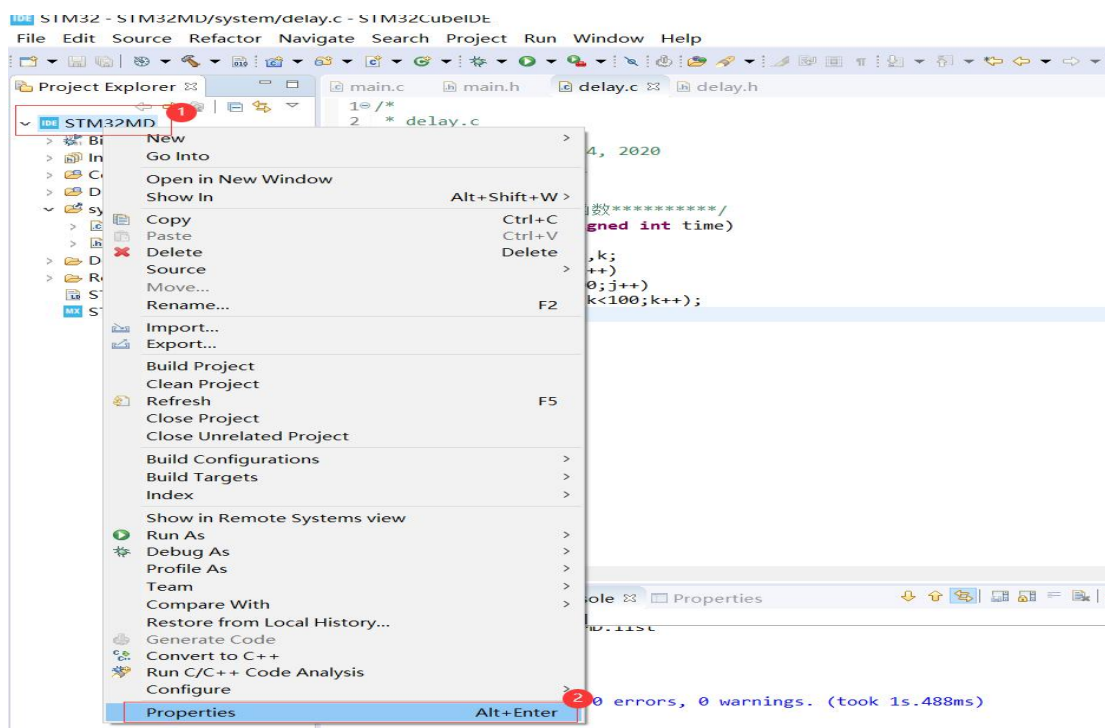


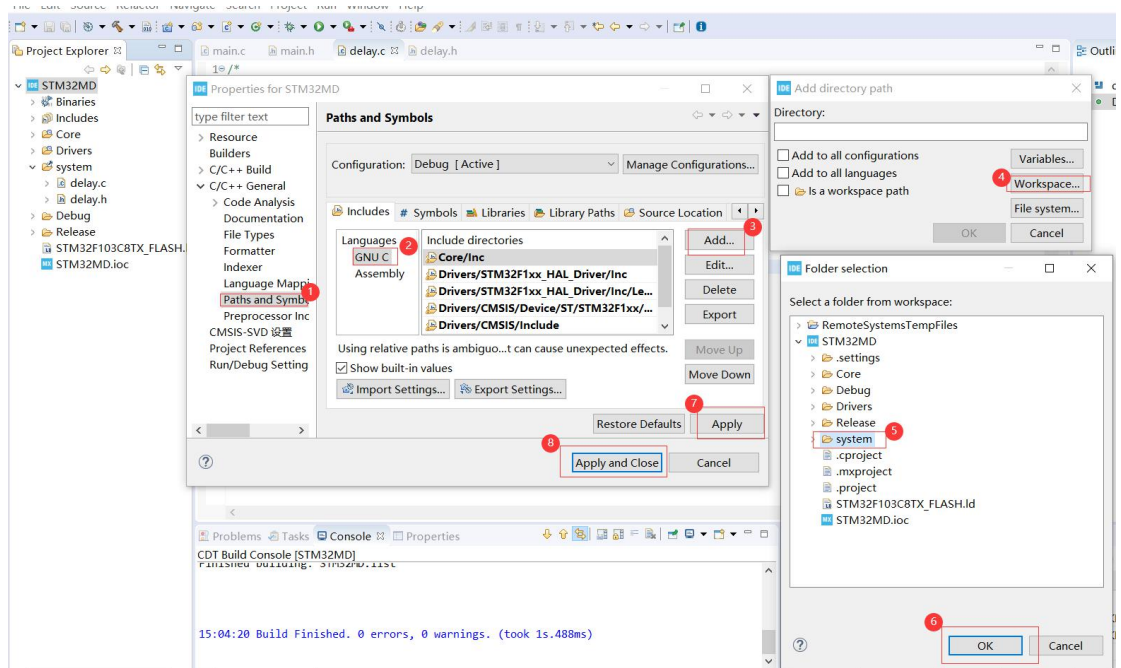
4. 自定义.c 和.h 文件

4.1 创建文件夹

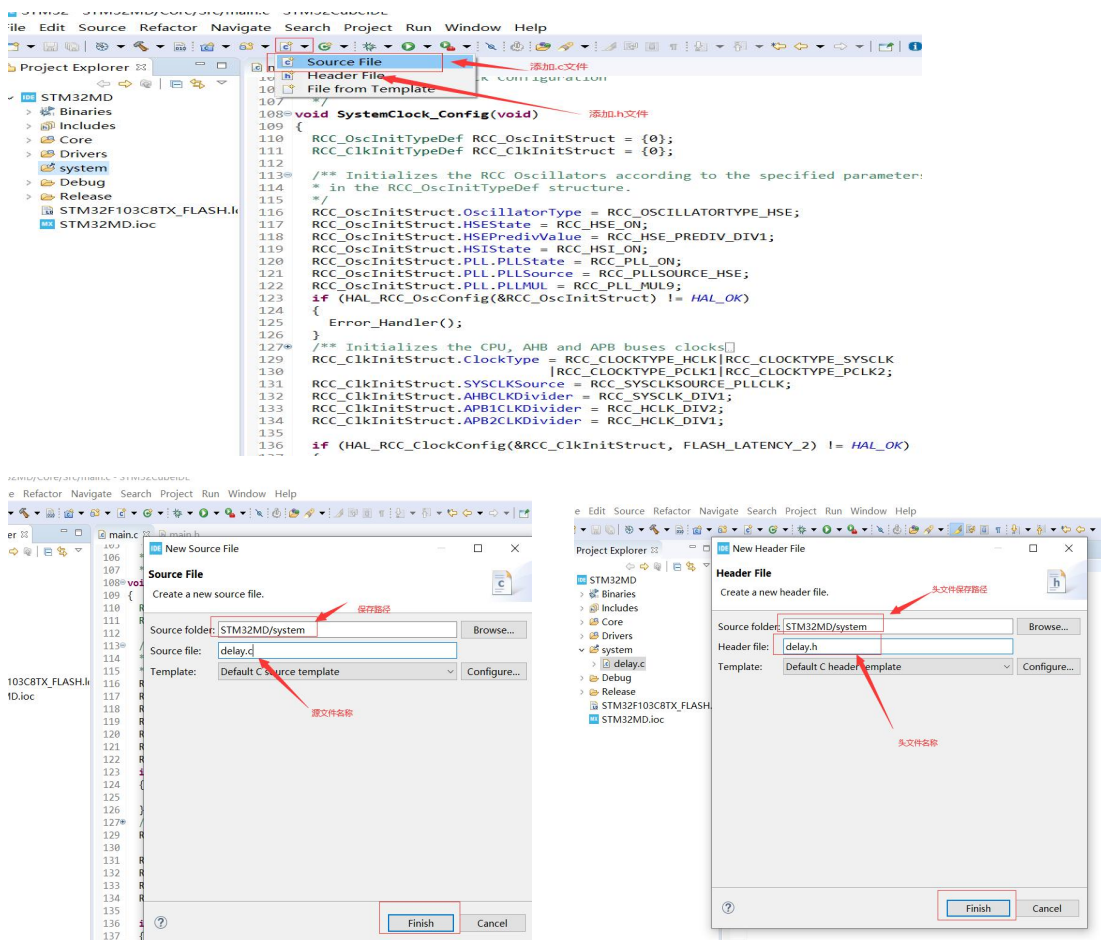


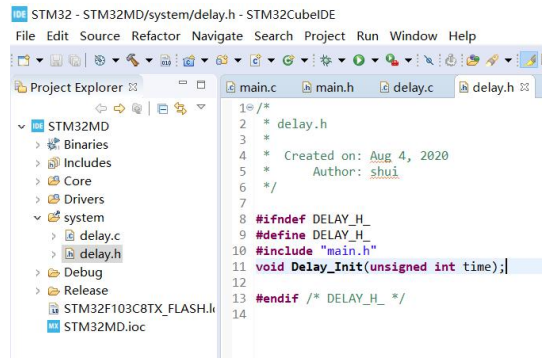
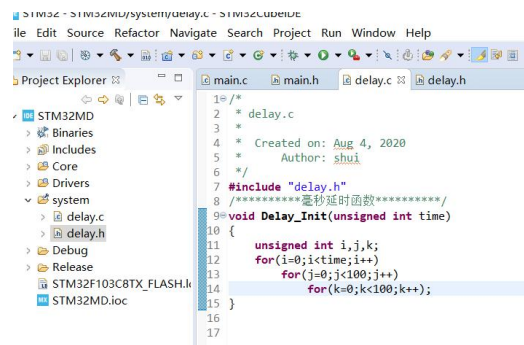
4.2 将文件夹添加到工作区



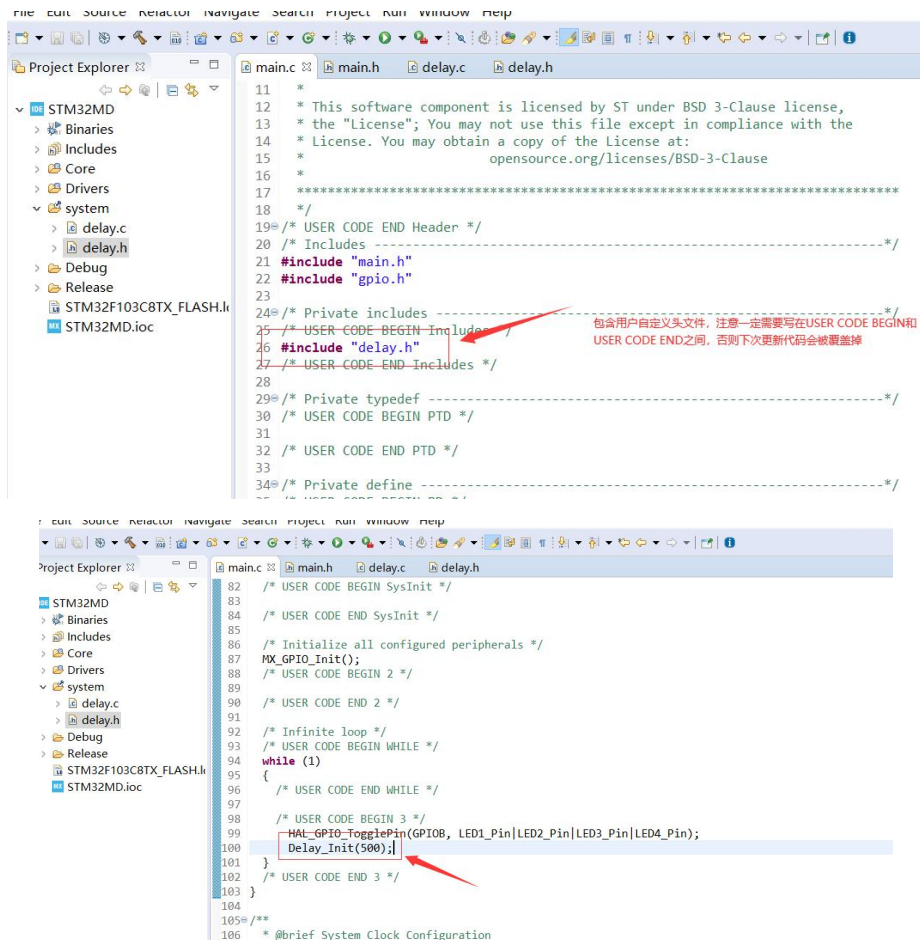


4.3 添加.c和.h文件：这里以添加 delay.c 和 delay.h 为例：





4.4 调用用户自定义函数：



4.5 程序编译，下载到开发板

