**、 STM32CubeIDE使用入门**

ST公司提供的STM32CubeIDE版本跟新较快，从2020年的1.3.0版本到现在的1.10.1版本，各版本STM32CubeIDE在界面上会略有不同，但基本相似。本文以STM32CubeIDEVersion 1.9.0 和 ST公司的图0-1 Nucleo-G431RB开发板为例，介绍STM32CubeIDE的基本使用。其他型号的STM32单片机，除了建立项目时，选择的单片机型号不同外，其他操作相同。

图0-1是实验板的实物结构图，该实验板上有一个STM32G431RB单片机和一个板载调试器(SWD接口)。板上有一个蓝色按键B1，接在单片机引脚PC13上(按下为1，释放为0) ；一个LED（LD2）接在PA5上参见原理图见图0-3；黑色按键B2，是单片机的复位按键；板载调试器, 带有USB虚拟串口功能，USB虚拟串口的收/发信号线与STM32G431RB单片机的LPUSART1和USART2的发/收信号线，在电路板上已通过PCB线互联，故与计算机进行串口通信实验时，可不用再另外接线。

如图0-2，请用扁口USB数据线，将实验板和计算机连接。**注意正确操作，不要用力过猛，避免造成实验板USB口的损坏。**

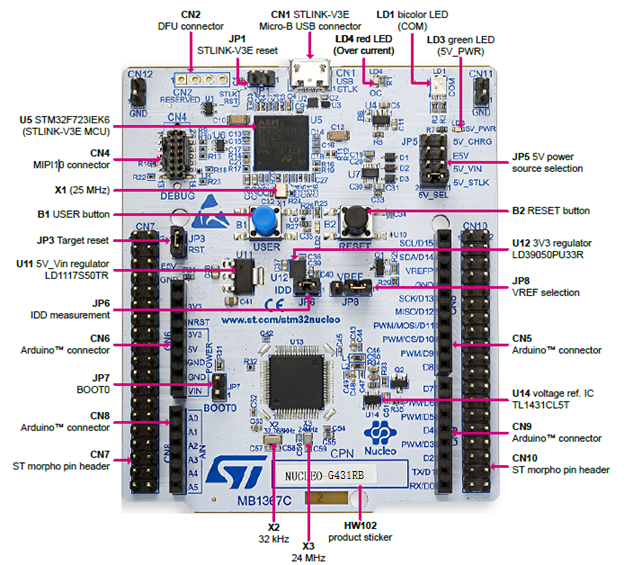


图0-1 Nucleo-G431RB实验板(大图见文末)

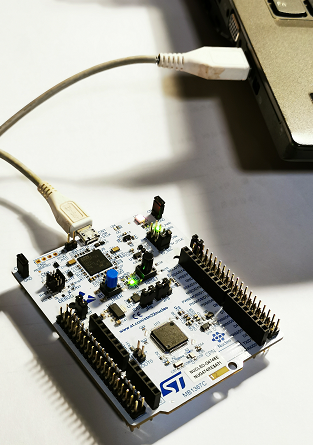


图0-2 实验板与计算机相连

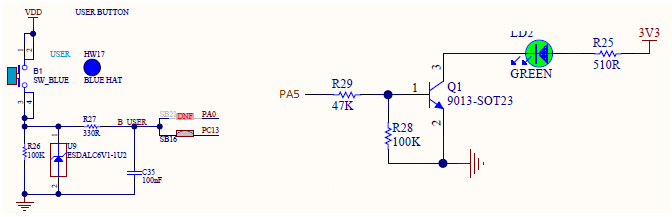


图0-3 Nucleo-G431RB开发板上按键和LED原理图

## 创建工程空间文件夹

在电脑硬盘上，创建一个存放STM32项目的文件夹，文件夹名字可自取，本例用D:\ 000stm32，作为STM32的工作文件夹。注意：文件夹路径不可出现中文文字，否则STM32CubeIDE不能正常工作。

如果使用实验室的电脑，只能将文件建在F: 盘符下，C~E盘均设有保护，重启系统，C~E盘用户的内容(程序)不被保存，将丢失。

## 运行STM32CubeIDE

点击STM32CubeIDE桌面快捷键，或点击桌面左下角“开始”，在所有程序中找到STM32CubeIDE运行。进入STM32CubeIDE欢迎界面, 如图0，可关闭该欢迎界面。

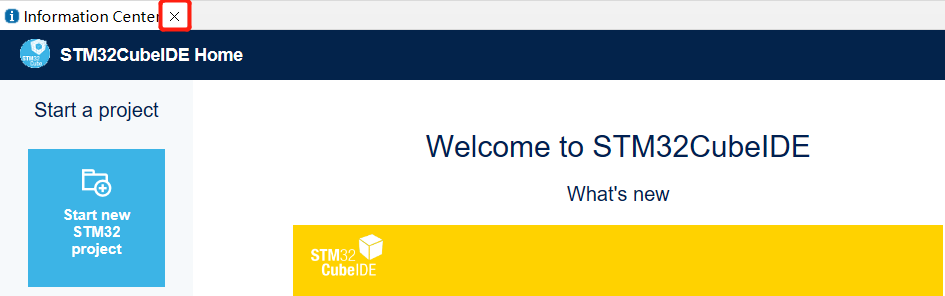


图0 STM32CubeIDE欢迎界面

## 创建新项目

如图1, 点击file>New>STM32 Project, 开始创建项目。创建项目过程中，如图2会显示项目在自动更新中, 有时会下载和更新一下库文件，会花一点时间，然后进入图3所示的选择目标MCU界面。

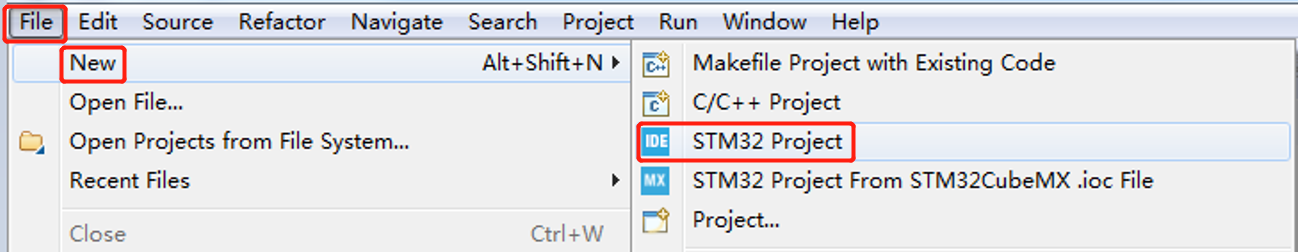


图1 创建一个新项目

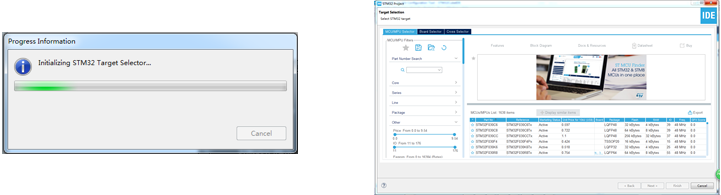


图2 显示在项目创建信息 图3 选择目标MCU界面

## 选择目标MCU

如图4，在Part Number 框中输入“G431RBT6”，在图右列出现的列表中，选择目标MCU型号**STM32G431RBT6**，并点击Next。注意型号不要输出，**STM32G431RBT6是实验板上焊接好的单片机型号。**

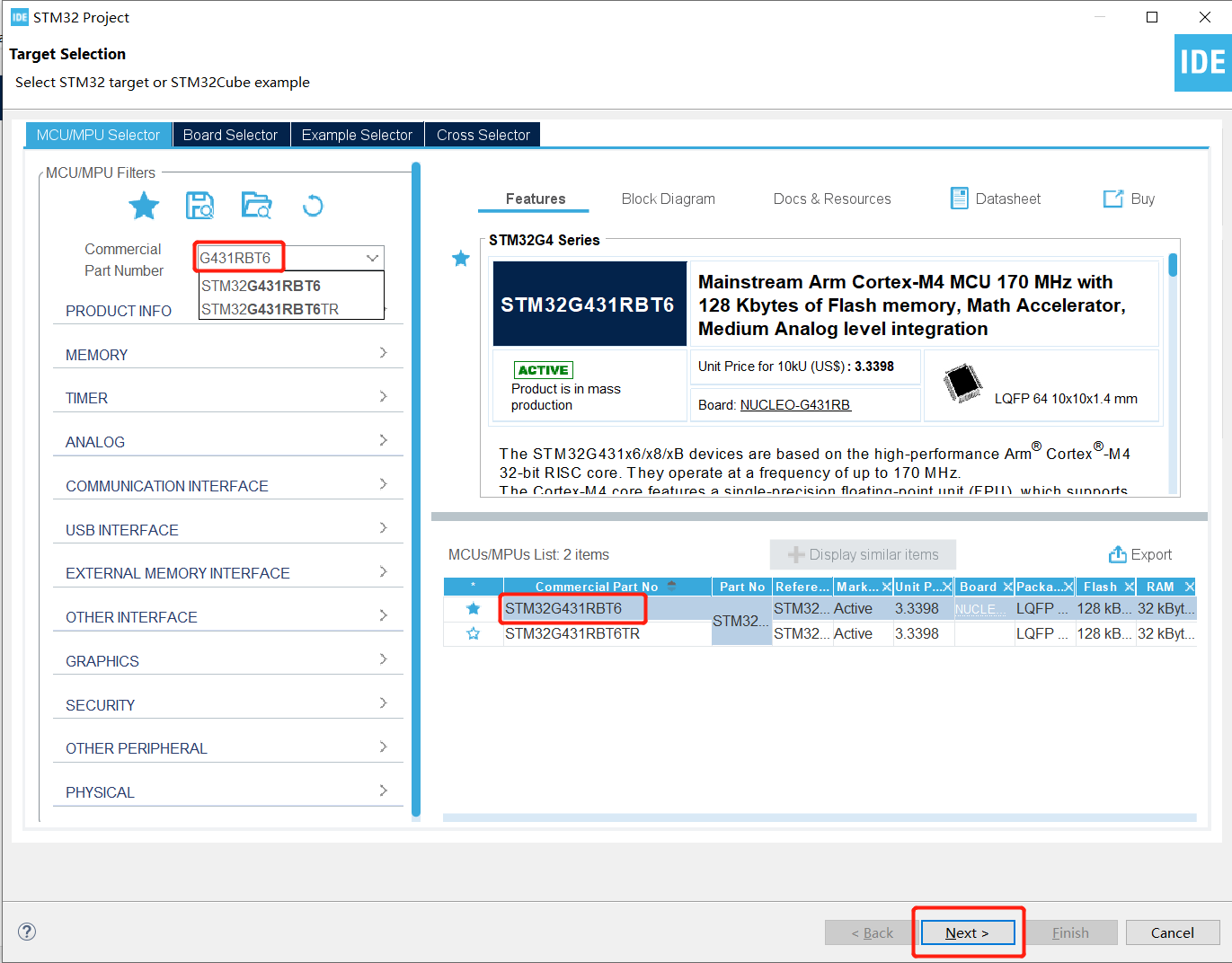


图4 选择与实验板一致的MCU型号

## 给项目命名和设置存放项目文件夹

在出现的STM32 Project界面中，可以如图5，点Browser选择自己设置的文件夹作为项目存放文件夹，如已创建好的D:/000STM32；也可以如图6，选用安装系统时默认的工程文件夹，作为项目文件夹。然后输入项目名testGPIO，然后点击finish, 开始创建STM32项目。

图7-1 给出启动STM32CubeMX的提示，可勾选提示信息中的“Remember my decision”, 然后如图7-2显示正在创建状态。创建完，进入图8的STM32CubeMX图形化配置界面。双击图中TestGPIO.ioc文件，可以最大化该窗口。

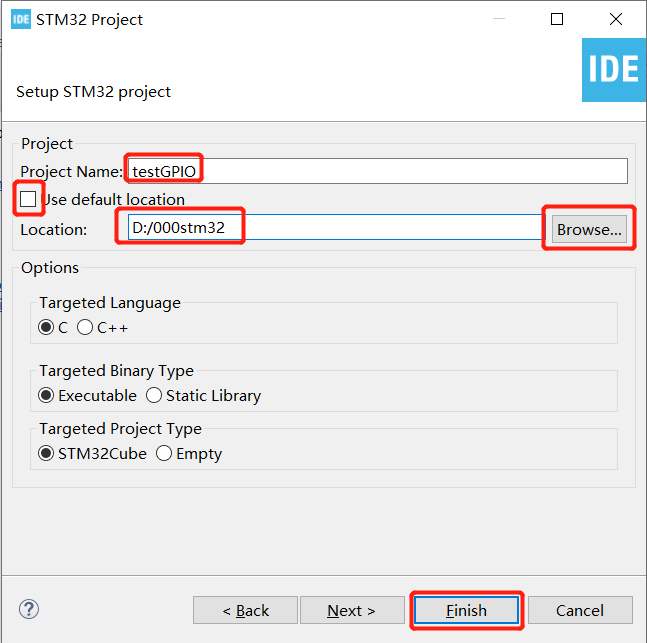


图5 选用自己设置的文件夹作为项目文件夹



图6 选用系统默认的文件夹作为项目文件夹

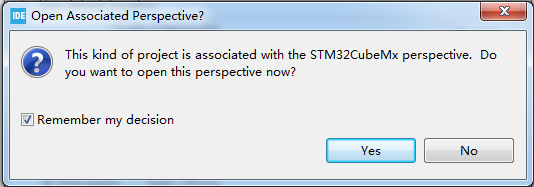


图7-1 项目在创建中状态

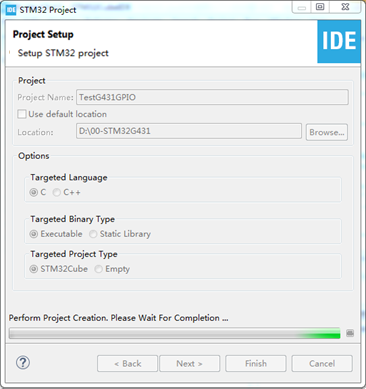


图7-2 项目在创建中状态

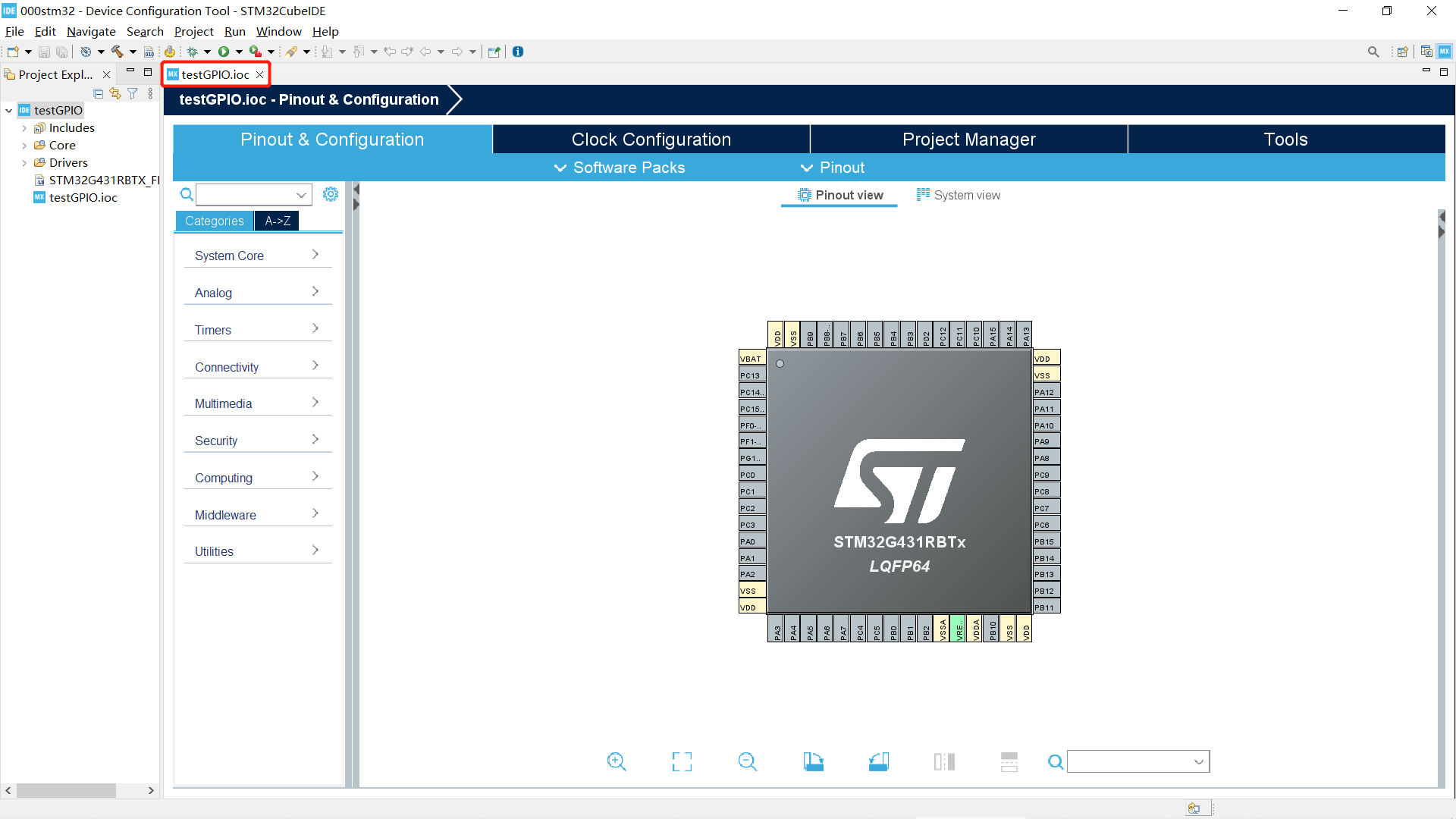


图8 STM32CubeMX图形化配置界面

## 配置RCC(Reset and Clock Control)

如图9，点击Pinout & Configurations>Categories >System Core >RCC , 然后在右侧的RCC Mode and Configuration 中，点击High Speed Clock(HSE)右侧下拉图标，选择 Crystal/Ceramic Resonator，即选用外部晶振。

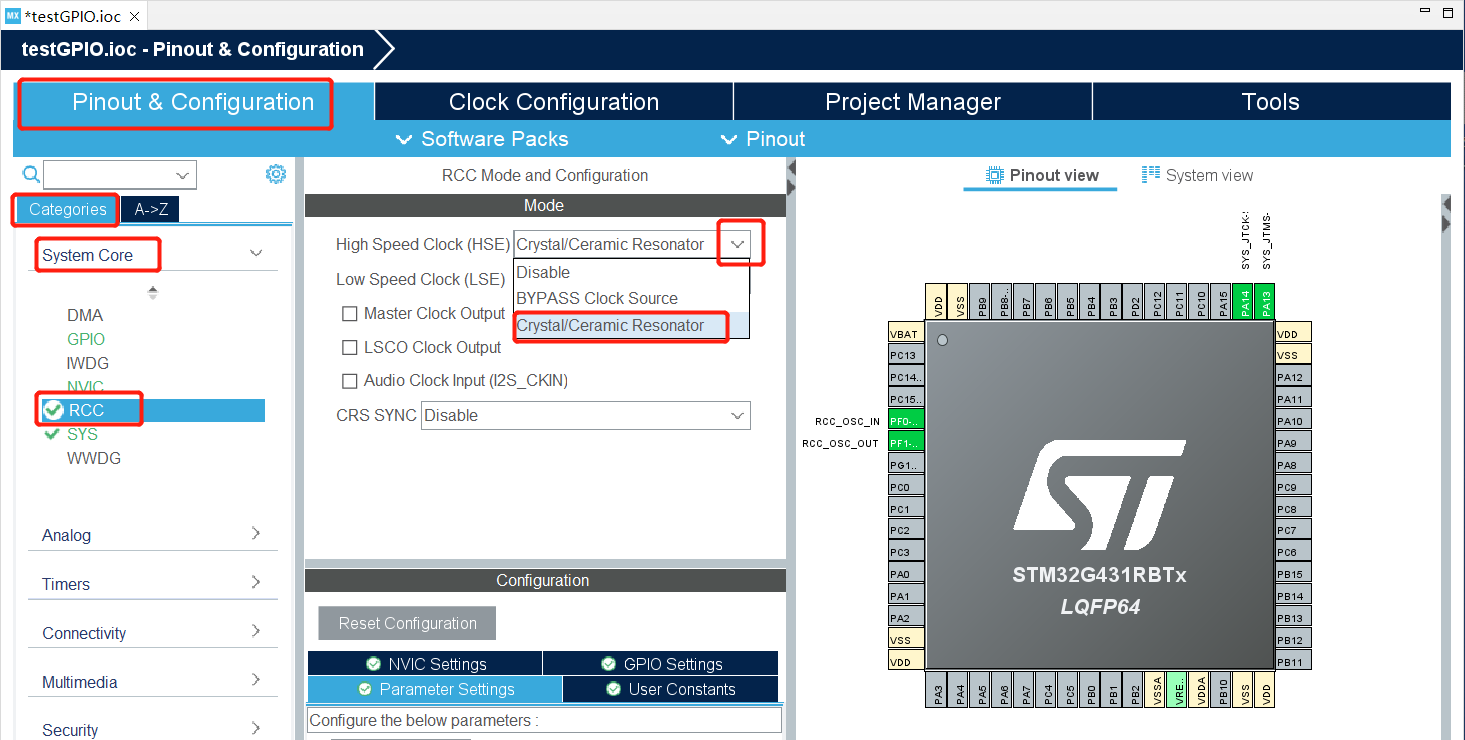


图9 配置RCC

## 配置SYS

如图10，点击Categories >System Core >SYS , 然后在右侧的SYS Mode and Configuration 中，点击Debug右侧下拉图标，选择 Serial Ware, 即选用SW模式。

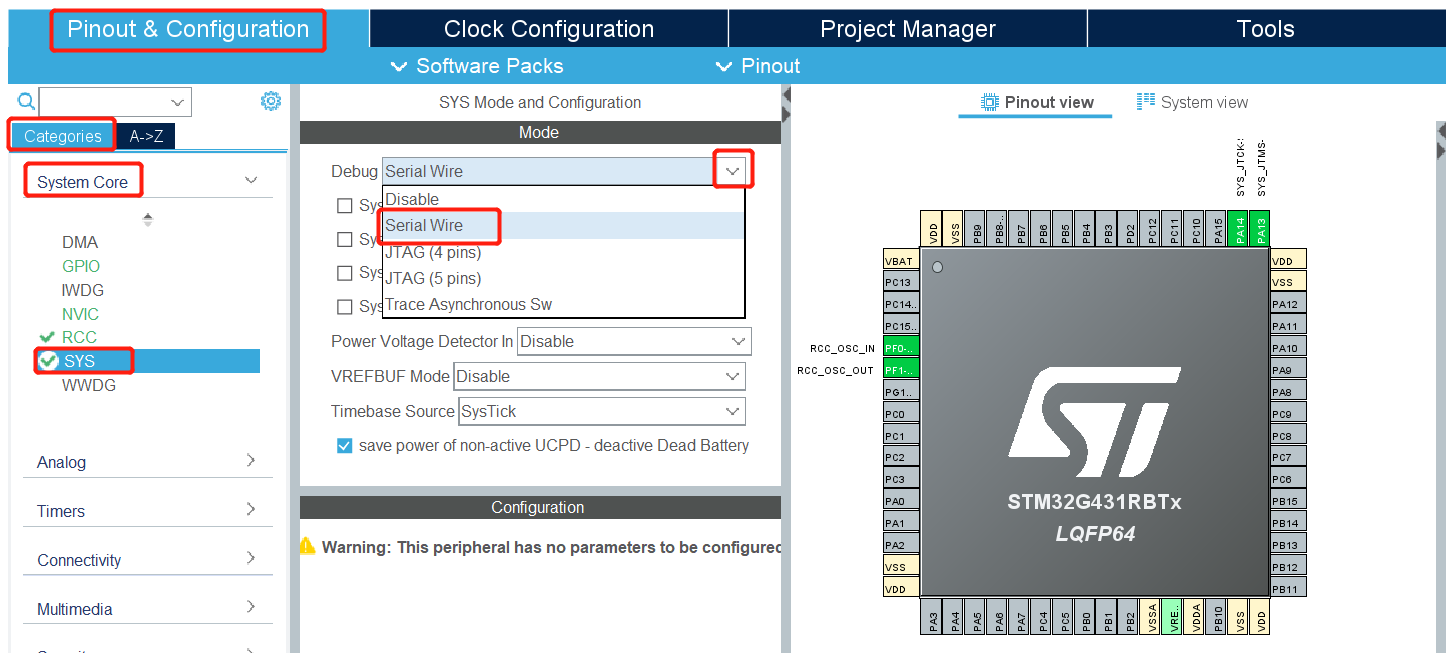


图10 配置SYS

8. **配置引脚功能**

**1) 配置引脚PA5为GPIO输出**：如图11，在Pinout View窗口的查找框，输入“PA5”，点击界面中闪动的引脚PA5，在出现的选项中，点击选择GPIO\_Output。



图11 配置引脚PA5功能

**2) 配置引脚PC13为GPIO输入**：如图12，在Pinout View窗口的查找框，输入“PC13”，点击界面中闪动的引脚PC13，在出现的选项中，点击选择GPIO\_Intput。

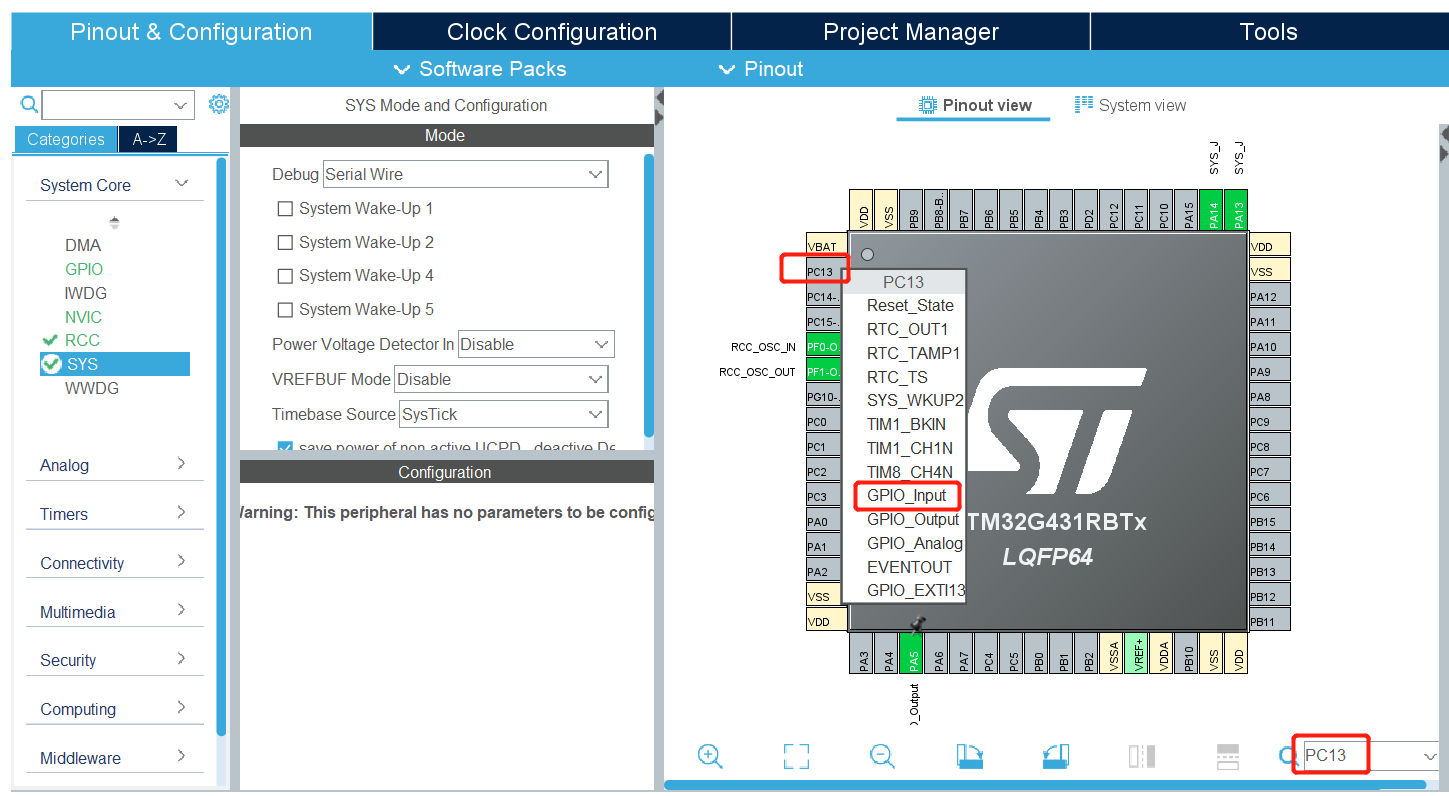


图12 配置引脚PC13功能

**9. 配置GPIO模块**

如图13，点击左侧Pinout & Configurations>Categories >Systerm Core >GPIO , 然后在右侧的PA5，出现与引脚PA5有关的GPIO配置，目前选择默认配置即可；如图14，点击右侧的PC13，出现与引脚PC13有关的GPIO配置，目前选择默认配置即可。

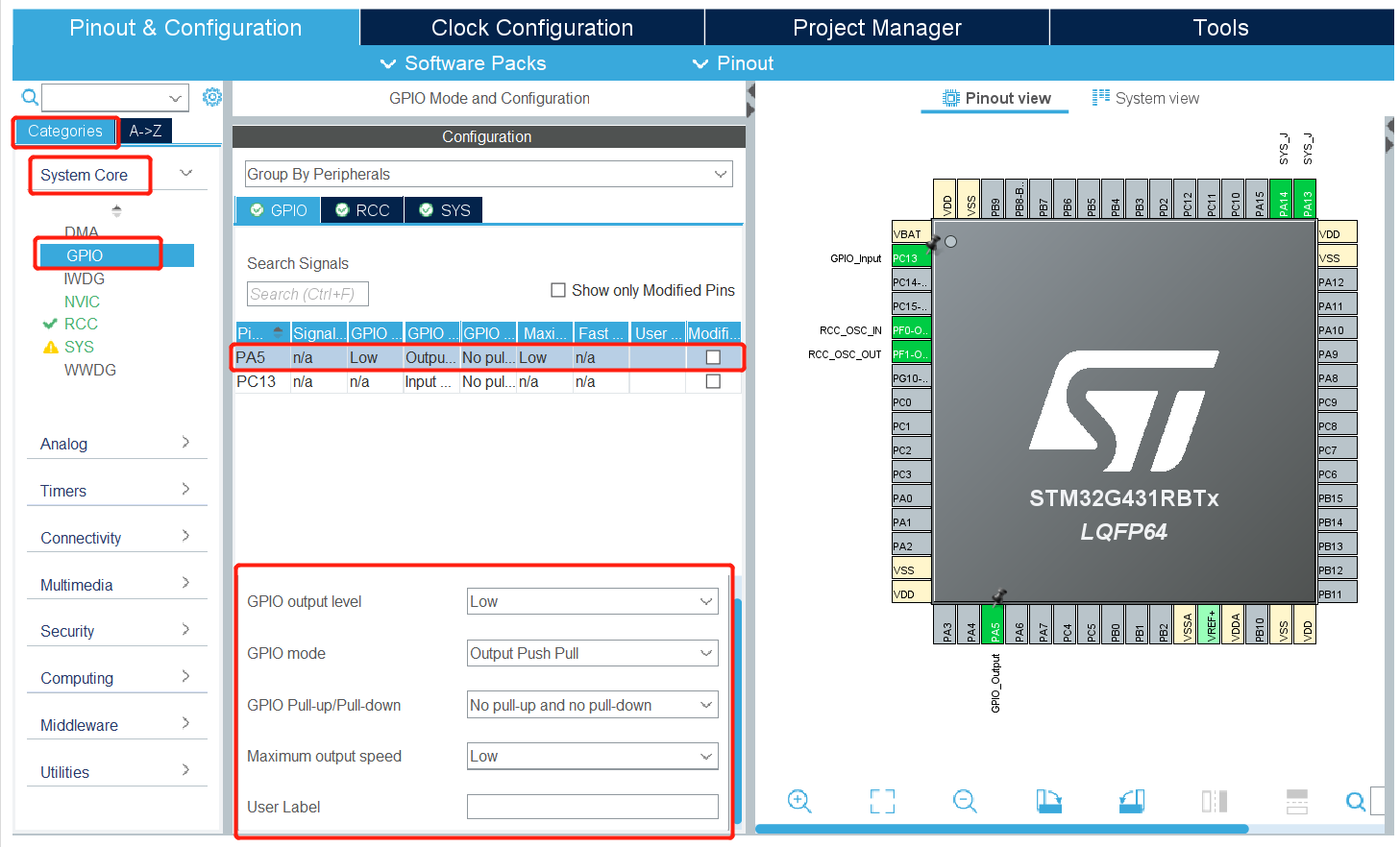


图13 配置GPIO引脚PA5

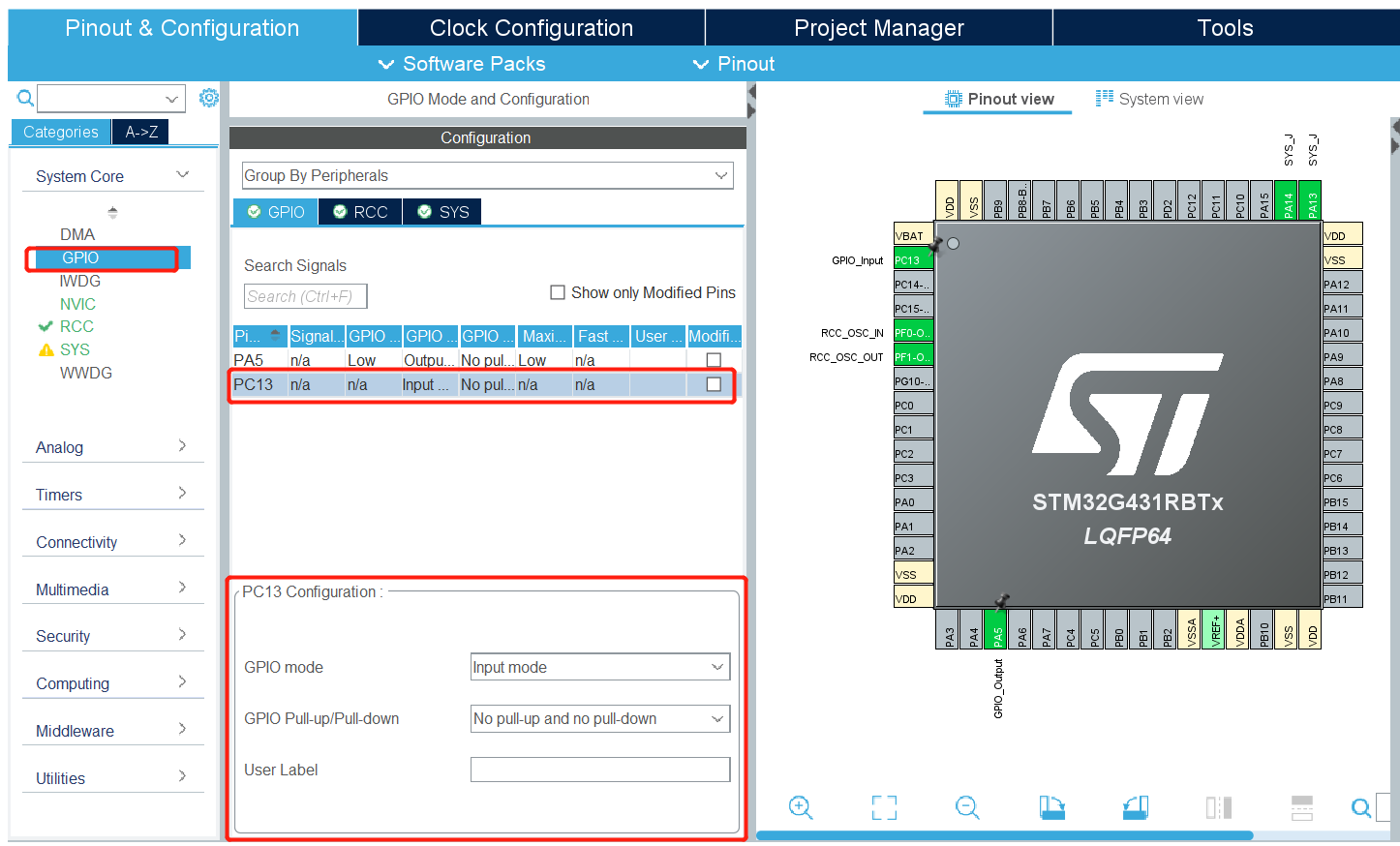


图14 配置GPIO引脚PC13

10. **配置时钟**

如图15，点击Clock Configuration , 在出现的时钟树中，1）在Input Frequency框中，输入24，即外部晶振为24MHz；2）在PLL Source MUX中选择HSE；3）Sytem Clock MUX中选择PLLCLK；4）然后在HCLK(MHz)框中，输入 170，即170MHz，**然后回车**，系统自动按照所配置的选项，自动配置相关的PLLM、PLL、和AHB Prescaler中的参数。如果自动配置不成功，可以按图15中绿色方框内的选项，做设置，分别设置/ 6、/85、/2，完成手动配置HCLK为170Mhz。

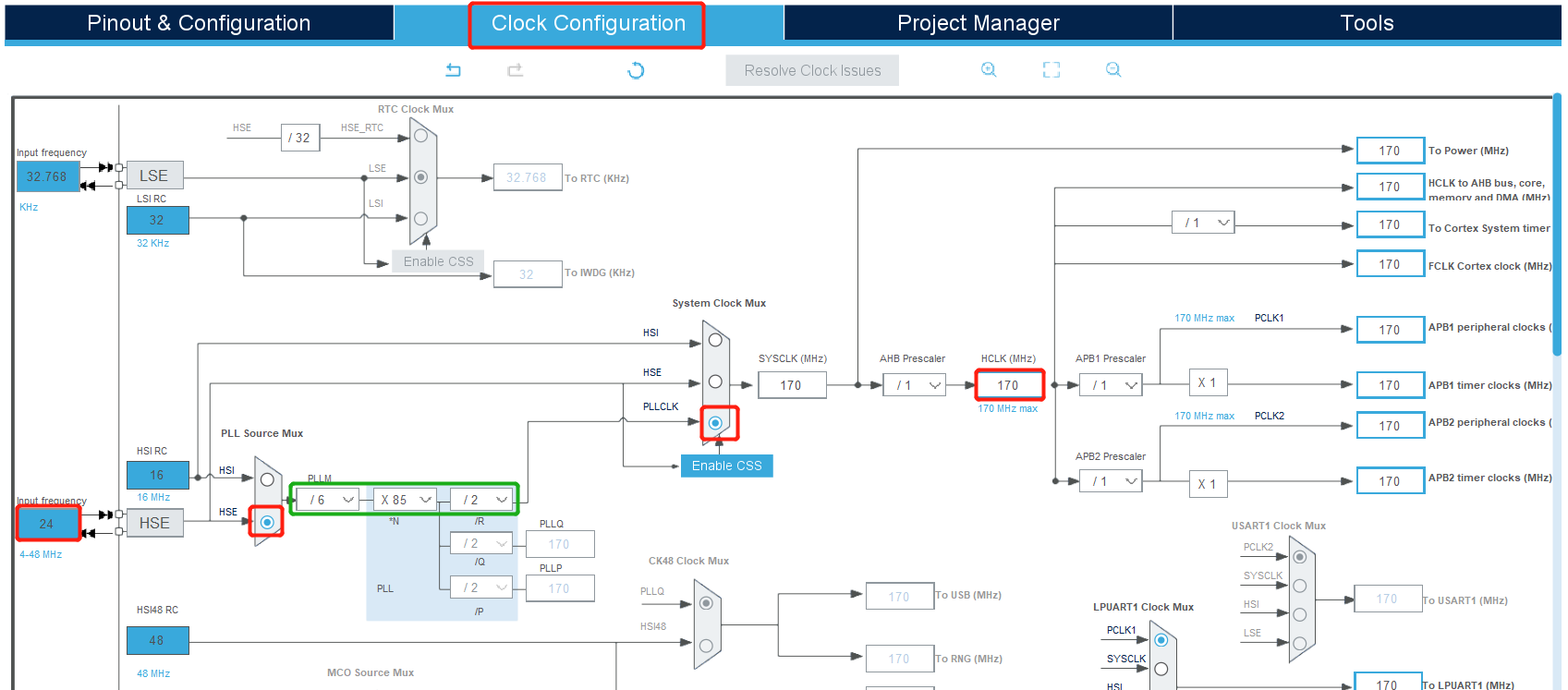


图15 配置时钟

**11．设置Project Manager**

如图16和图17，可设置和查看一下项目管理中的Project、Code Generator ，目前不要修改，保留默认设置即可。

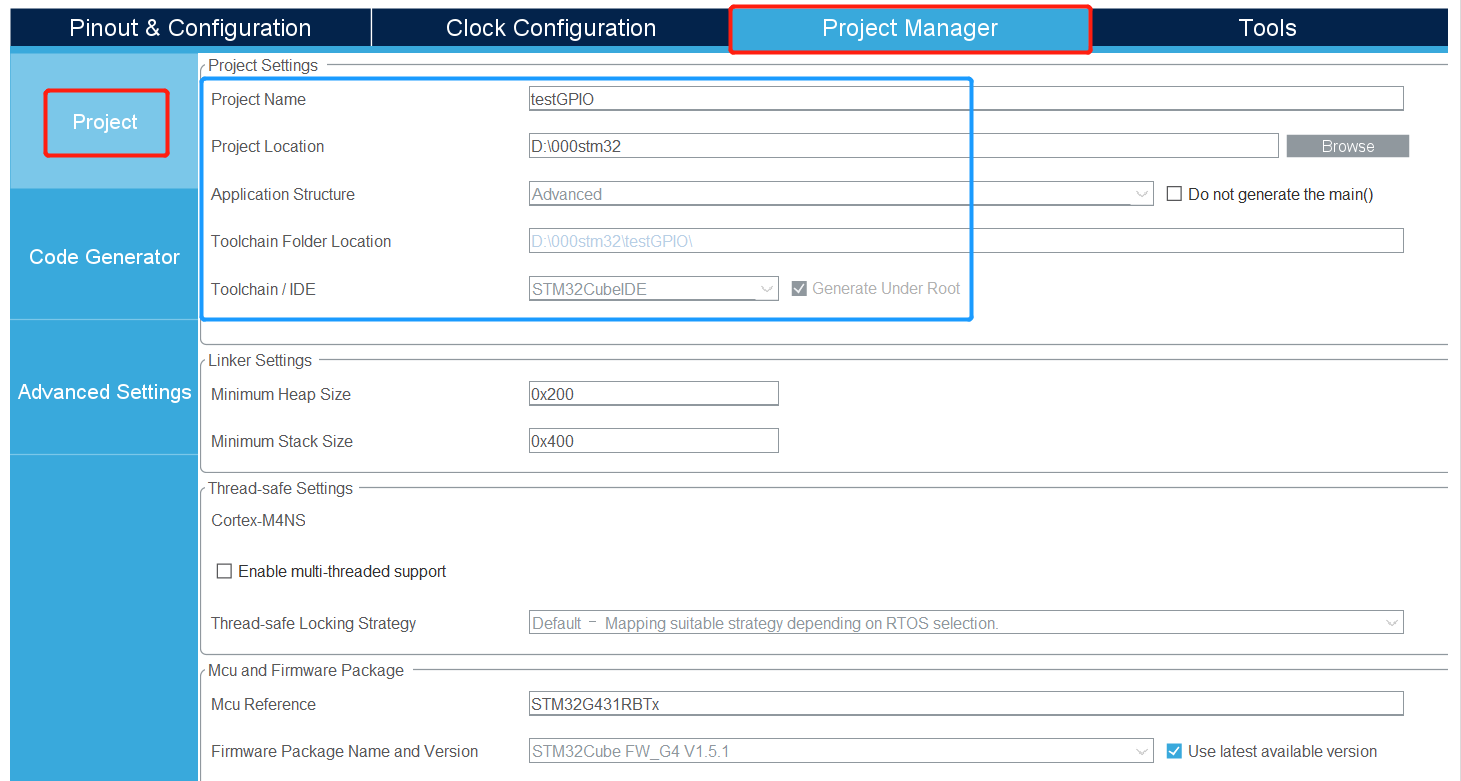


图16 Project设置

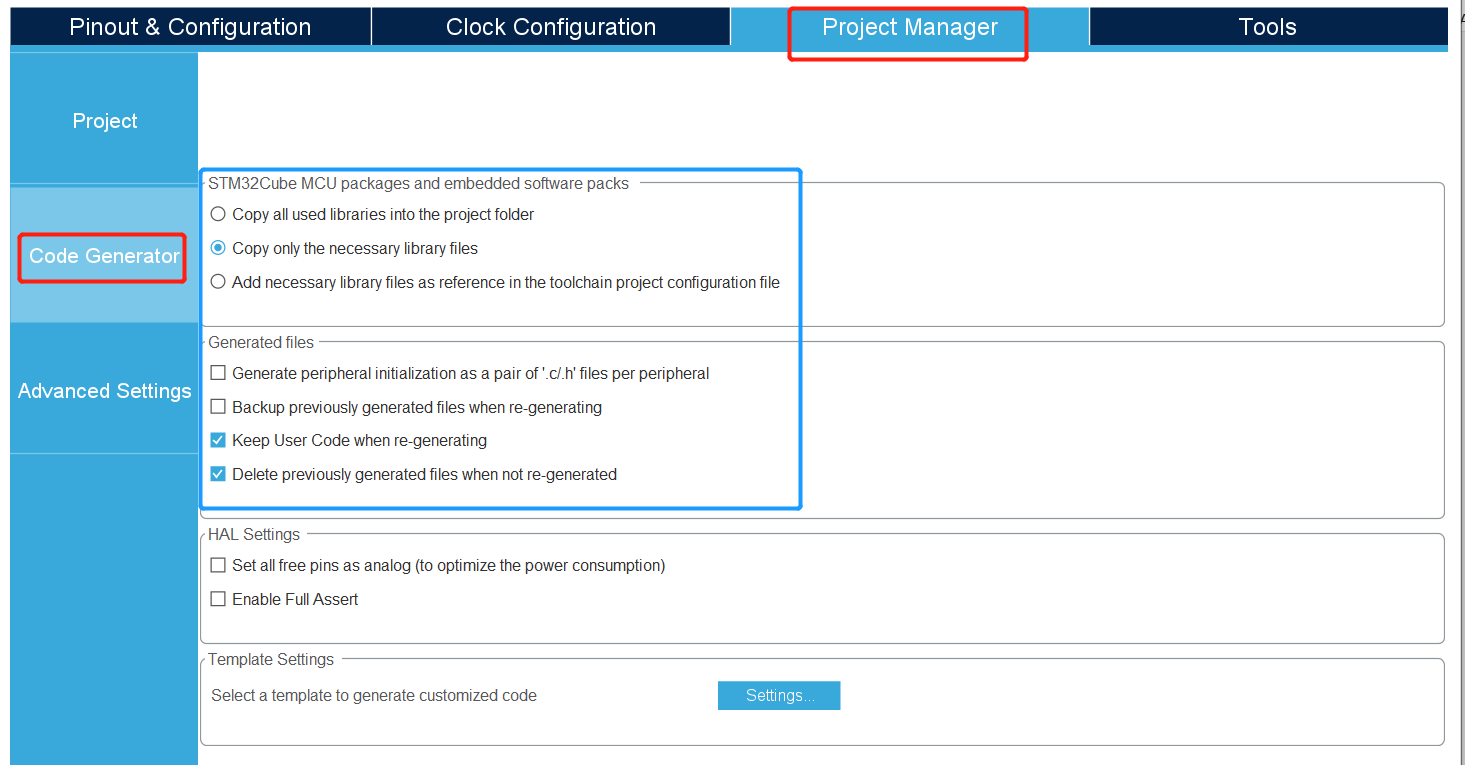
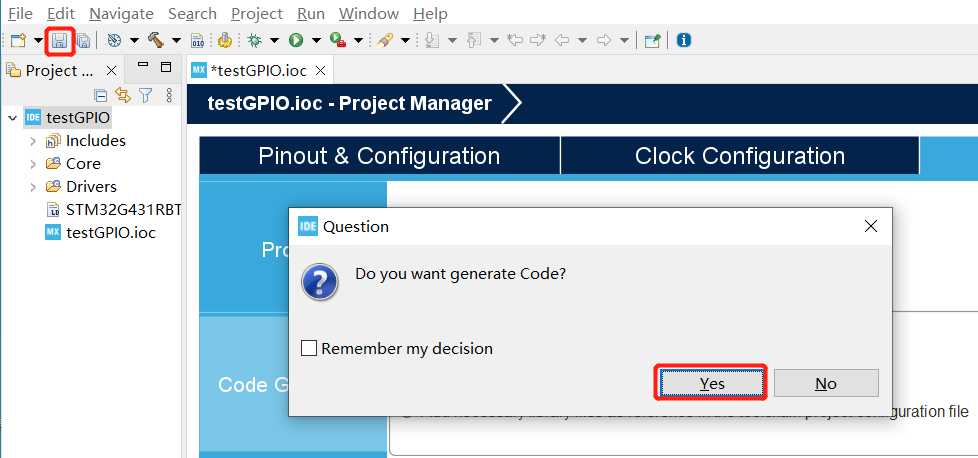


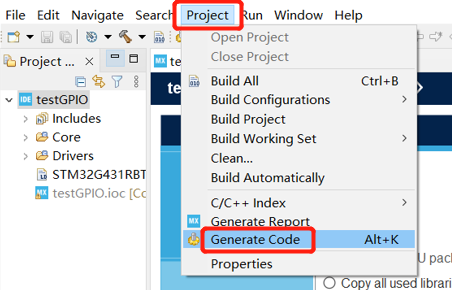
图17 Code Generator 设置

**12. 生成代码（Generate Code）**

如图18，点击菜单栏中的保存；或如图19，点击Projec>Generate Code,在弹出的“Do you want to generate Code？”确认框中，点确认yes。系统开始形成带有初始化的程序代码，并出现图20 打开编辑窗口的提示，点击确认。如果首次创建项目，如图21，系统会自动联网，下载和解压所需要的HAL和ARM CMSIS库文件。如果项目配置有问题，会弹出图22所示的代码生成确认框。建议选择No，查看和修改时钟树等的配置。然后重新生成代码。



如图18 点击保存，开始生成代码



如图19 Projec>Generate Code，开始生成代码

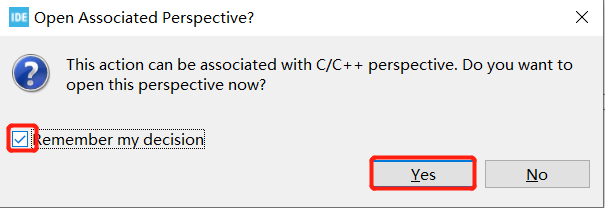


图20 提示将打开编辑窗口

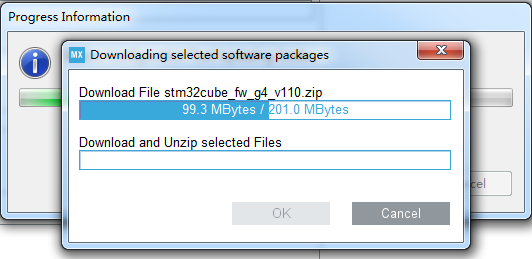


图21 自动下载和安装STM32和CSII库文件

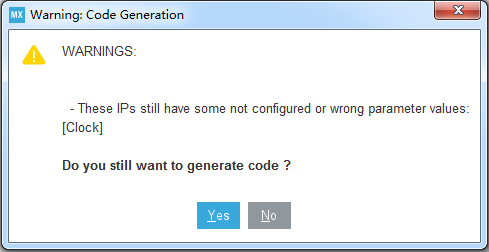


图22 配置出错提示信息

**13. 编写项目程序**

1) 代码生成后，如图23，点击右上角的编辑图标，**进入到图23-1 Porject Explorer界面**。

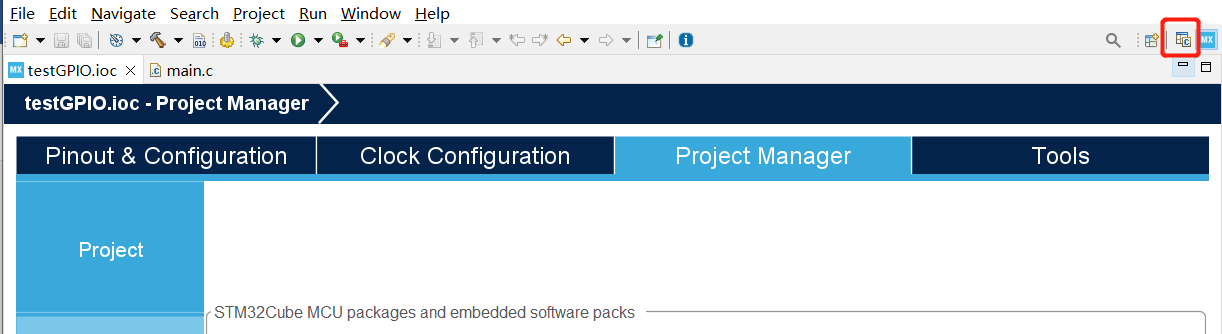


图23 点击右上角Porject Explorer图标

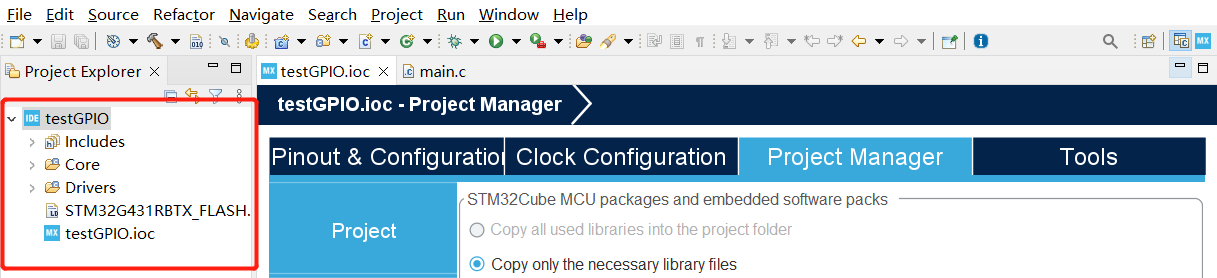


图24 Porject Explorer界面

**2) 打开项目的main.c文件：**如图25，在Porject Explorer界面，点击testGPIO>Core>Src，双击Src下的main.c，打开main.c文件。

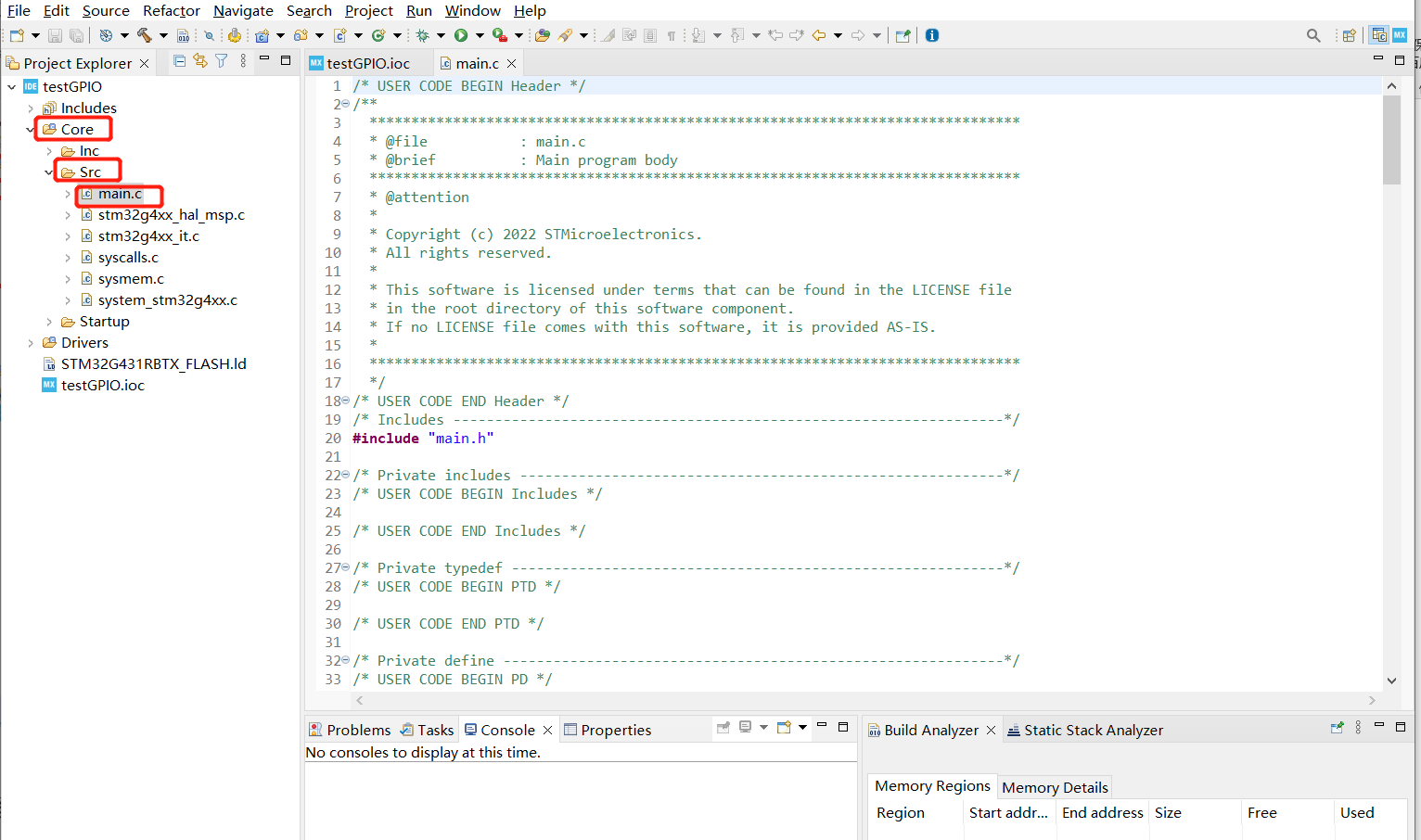


图25 打开main.c文件

**3) 修改main.c 文件：**

如图26，在main.c中，在main( )函数内while(1)主循环 /\* USER CODE BEGIN 3 \*/后，添加如下代码，可以直接拷贝。注意这里用到了两个函数，一个是读引脚状态函数，一个是写引脚函数，实现对引脚的输入、输出控制。

if ( HAL\_GPIO\_ReadPin(GPIOC, GPIO\_PIN\_13)) // 判断按键是否按下:1按下,0释放

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5,GPIO\_PIN\_SET); //按下亮

else HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5,GPIO\_PIN\_RESET); //释放灭

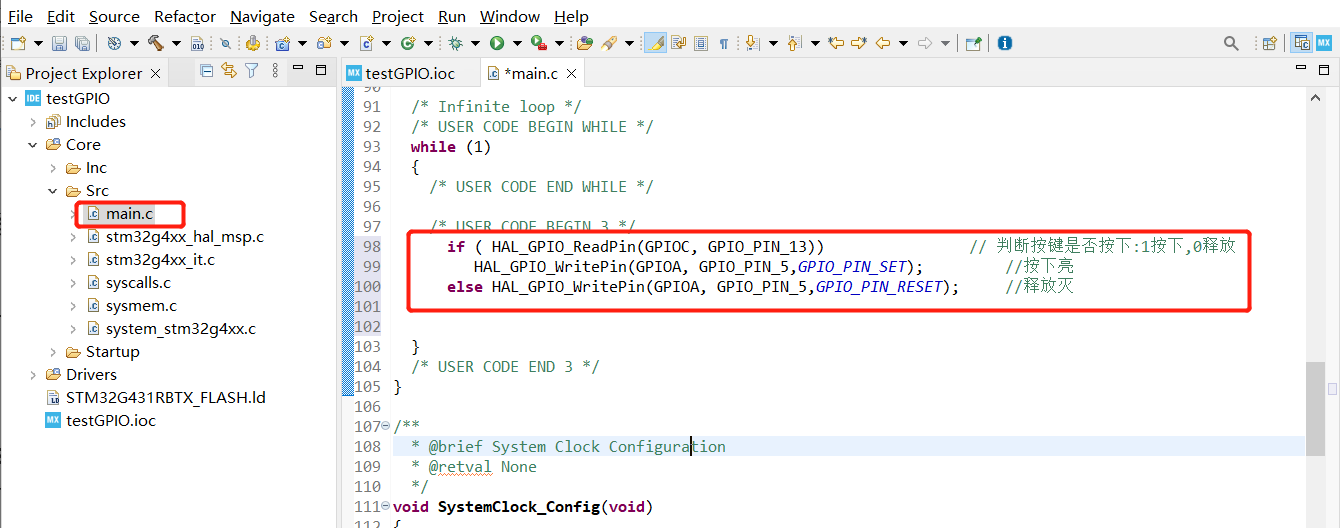


图26 修改main( )函数

**14. 编译项目(Build Project)**

如图27，点击图标 ，或点击Project>Build Project。编译后，会在控制台console窗口显示相关信息，无错则形成.elf文件，可下载到MCU运行、调试。如果有错，会在console串口显示编译出现的错误提示。

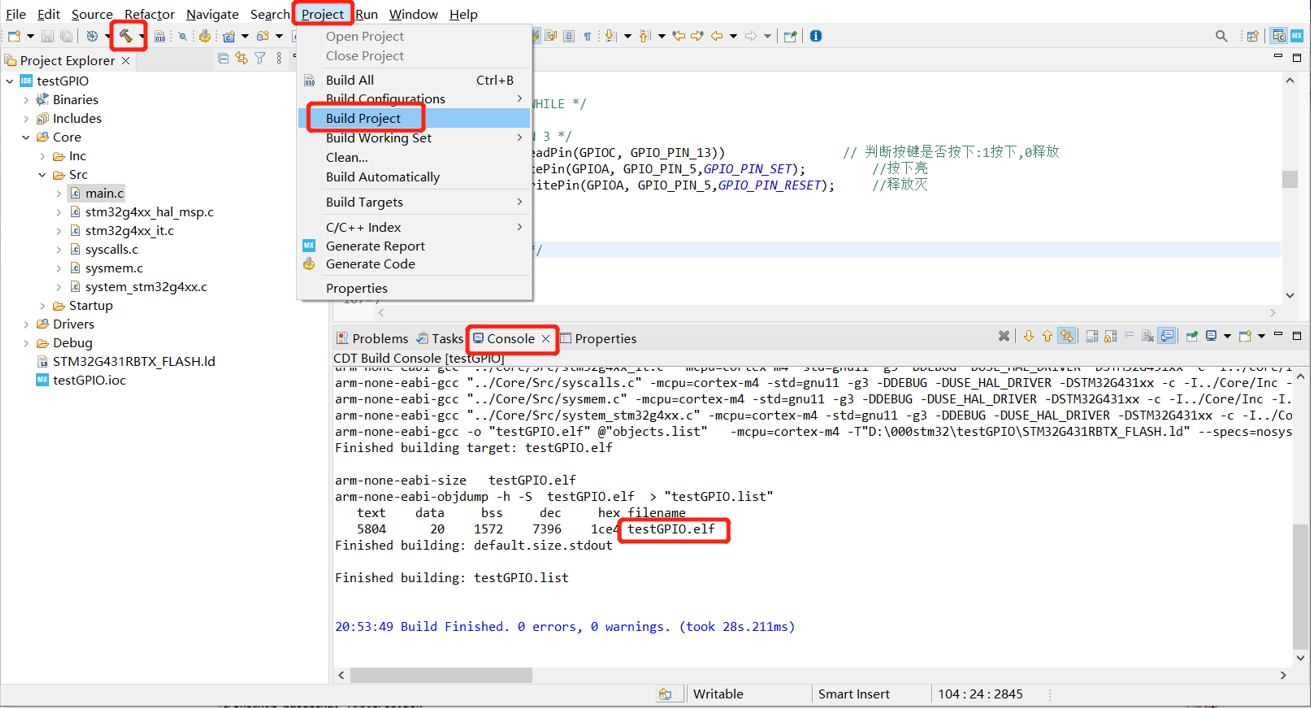


图27 编译项目

**15. 设置调试配置(Debug Configurations)**

1）启动Debug Configurations：如图28，点击菜单栏上Run> Debug Configurations。

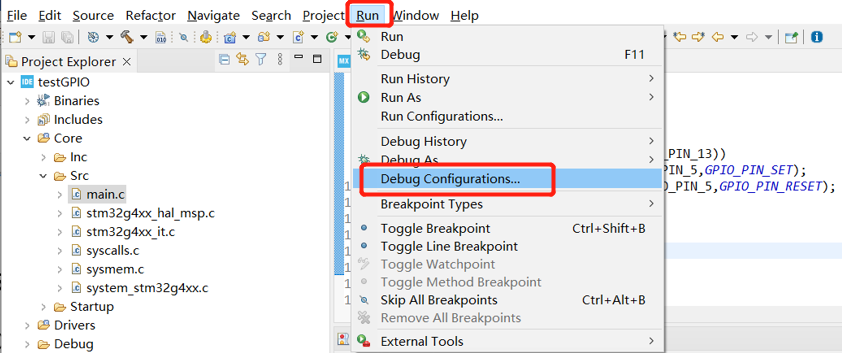


图28 启动Debug Configurations

2) 如图29-1，在出现调试配置界面，用鼠标双击点STM32 Cortext-M C/C++ Application，刷新其中的项目，出现图29-2可调试的项目，选中testGPIO项目，右侧出现Debug\testGPIO.elf 。

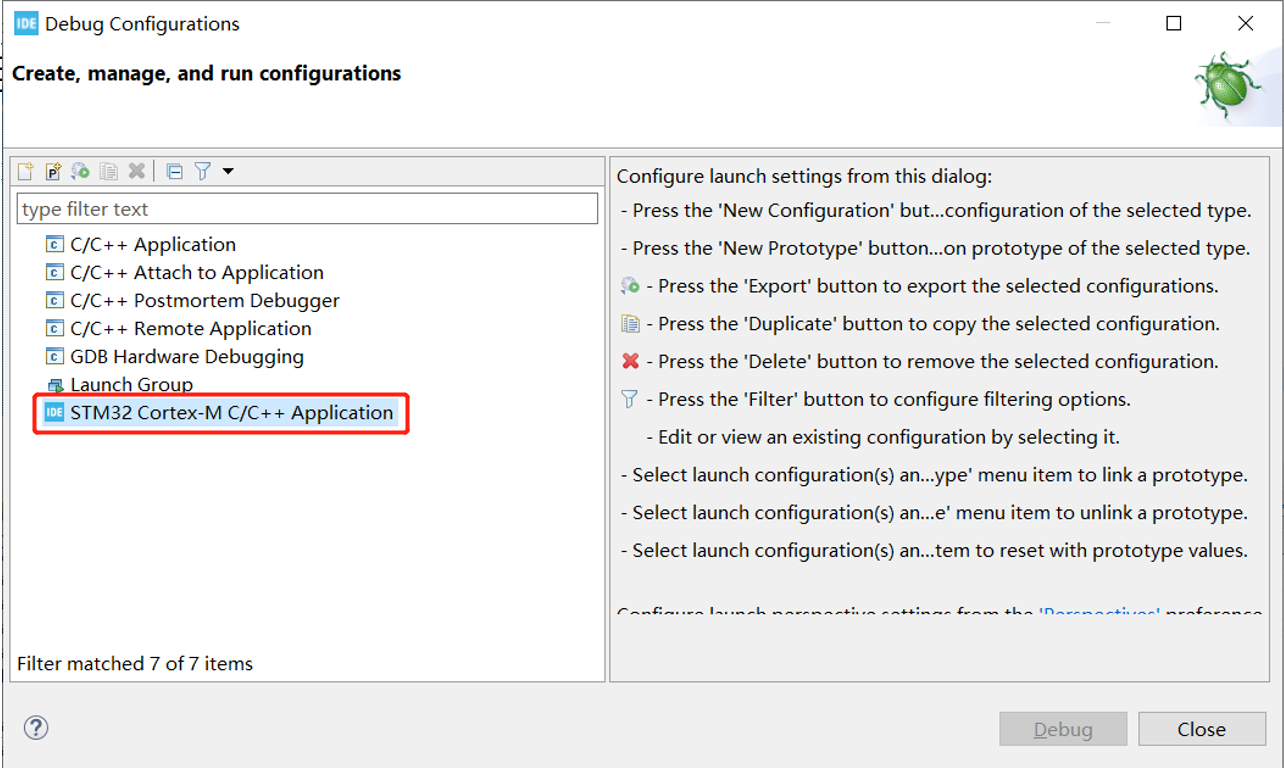


图29-1 调试配置界面

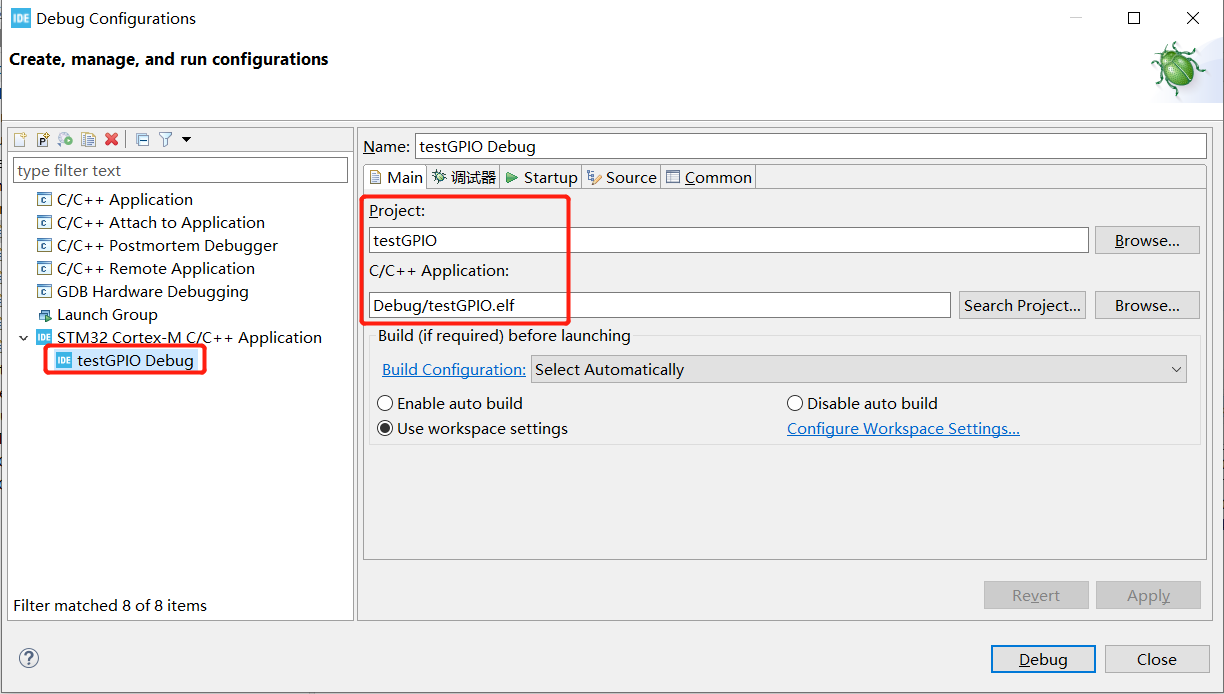


图29-2 Debug Configurations界面

3) 如图30，根据连接的调试器类型，确认调试器配置，准备下载和调试。

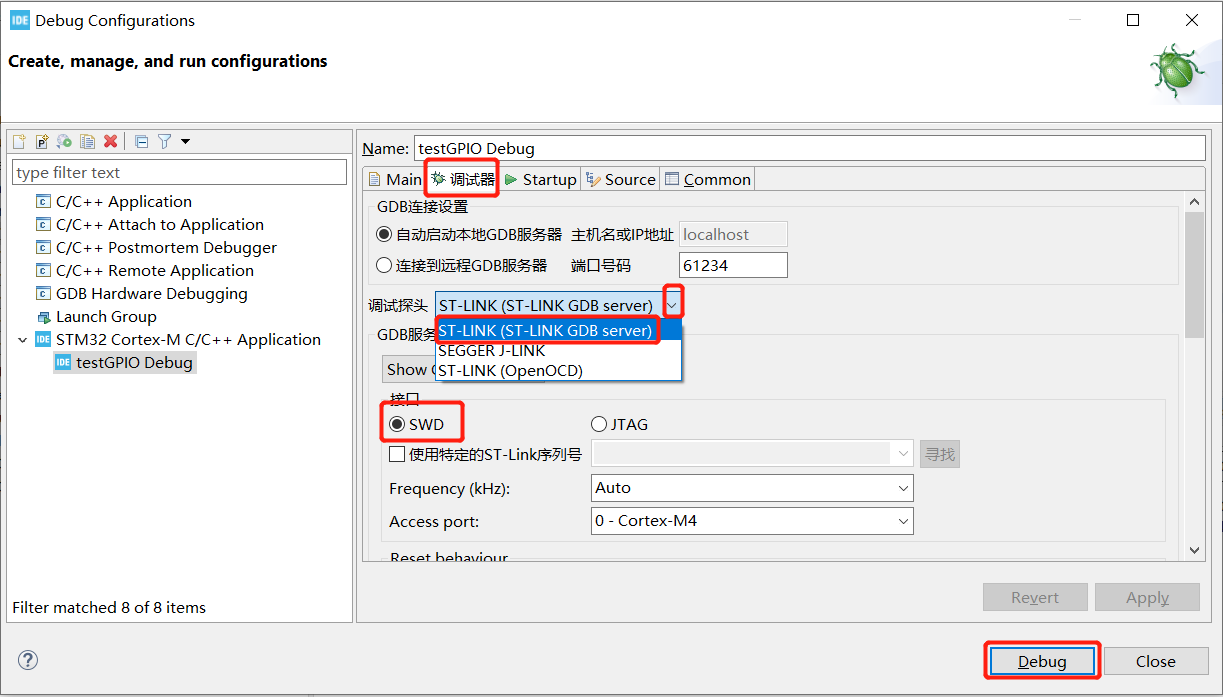


图30 设置调试器接口

4) 确认实验板已通过USB数据线和电脑连接好，点击图30右下角的Debug，会出现图31-1保存和发送信息窗口，可以勾选“Always save resources before launching”， 点击OK，在出现的图31-2 Confirm Perspective Switch界面，可以勾选“Remember my decision”。然后点击图30-2 Confirm Perspective Switch窗口中的Switch。程序开始下载到MCU中，没有出错的话，进入到图32的DEBUG调试界面，可进入下面的步骤16。

如果出现图31-3中的错误提示，需点击OK。然后查看九三级设备管理器下，实验板的驱动程序安装好否。图31-4是实验板在设备管理器安装好的显示。如出现图31-5固件升级提示，点击Yes，进入到图31-6的固件升级界面。在图31-6中，先点击“Open in update mode”，待操作完成，再点击Update，开始升级。直到出现图31-7中的“Update successful”，固件升级完成，关闭该窗口。点击图31-8中的Debug图标，或菜单栏中的Run>Debug，进入图32中的调试界面。

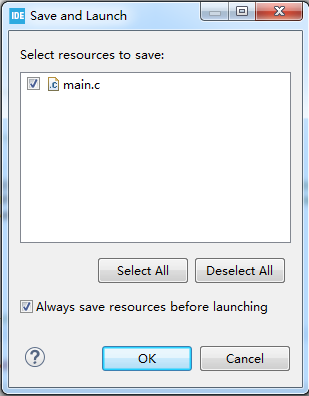


图31-1 Save and Launch确定界面

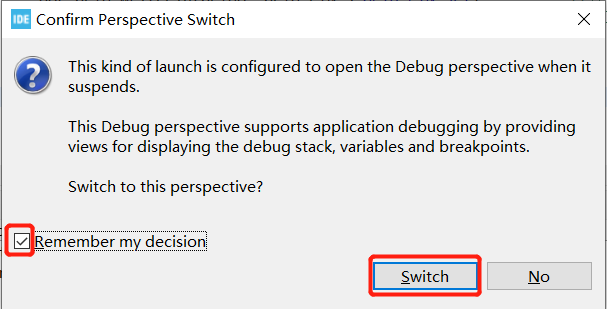


图31-2 Confirm Perspective Switch确定界面

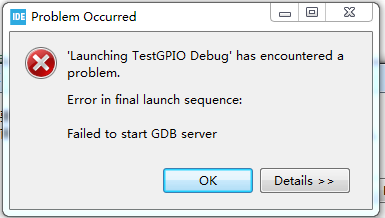


图31-3 下载时出现错误提示信息

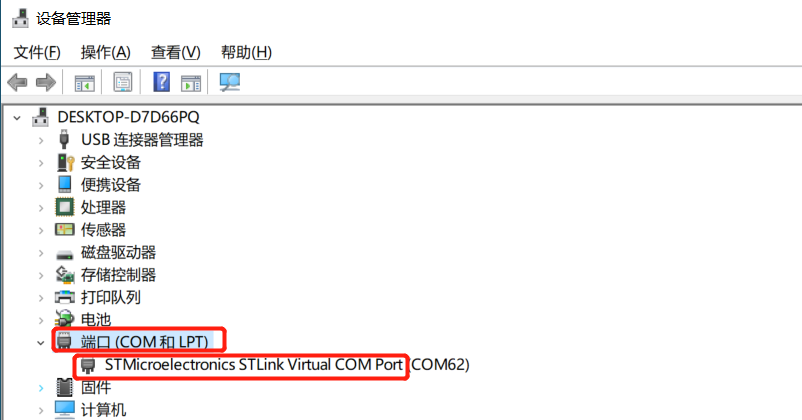


图31-4 实验板在设备管理器下的显示

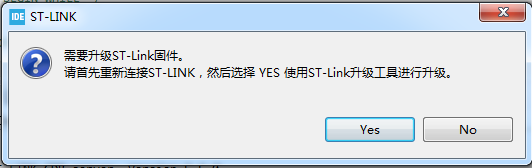


图31-5 ST-Link固件升级提示

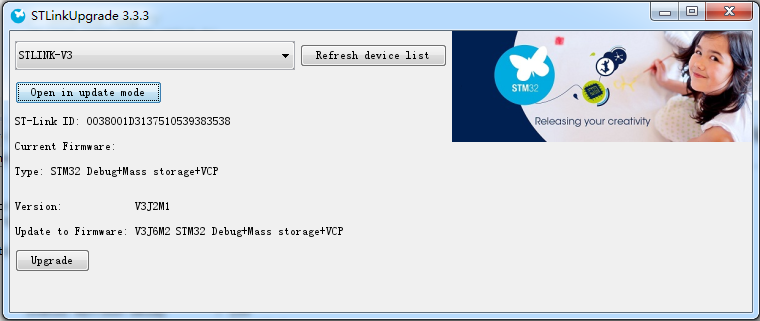


图31-6 固件升级界面

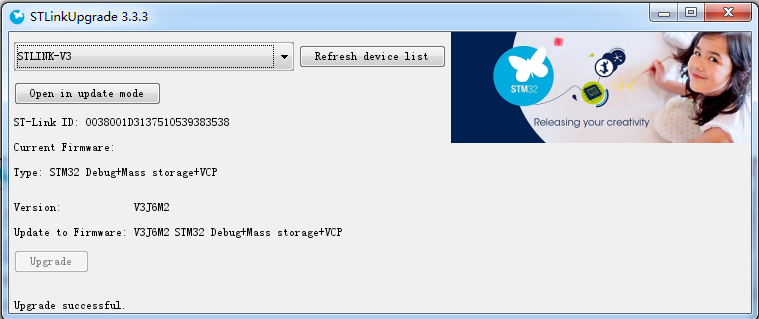


图31-7 固件升级完成

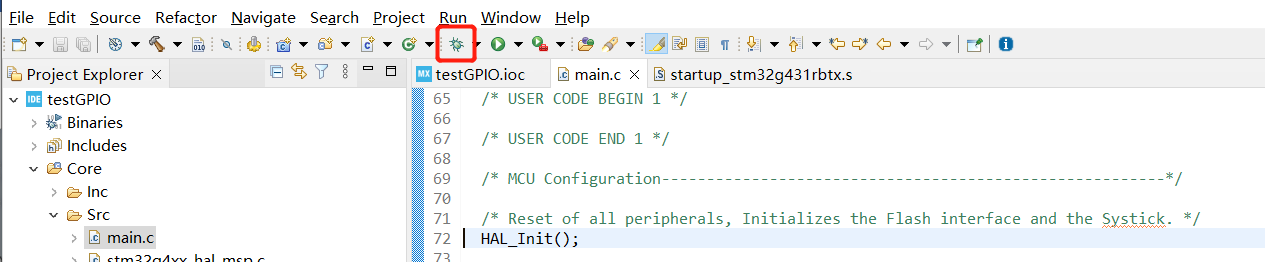


图31-8 菜单栏中的Debug图标

**特别注意：在更新实验板固件、以及下载程序到单片机的过程中，不能插拔单片机板**，否则会导致实验板烧写出错，实验板出现异常，不能被电脑识别，从而不能使用。

**16. 运行和调试程序**

**1) 运行程序：**

如图32，点击运行图标，操作实验板上的**蓝色按键B1**，可看到随着**蓝色按键的按下和释放**，蓝色按键旁一个很小的发光二极管LD2，将产生亮、灭变化，注意操作的**不是**实验板上的**黑色**按键。

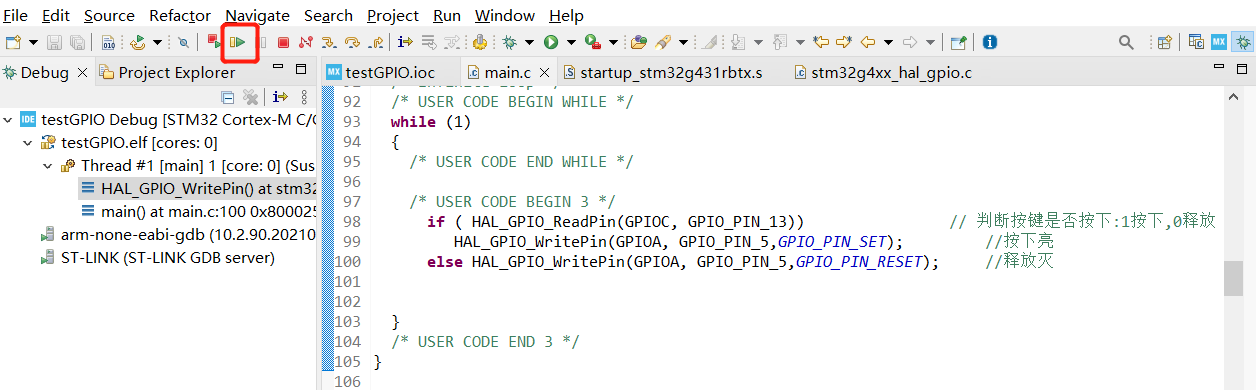


图32 点击运行图标运行程序

**2) 查看内部模块寄存器：**

如图33，点击暂停图标，点击Windows>Show View>SFRs，可以查看MCU内各模块内寄存器。

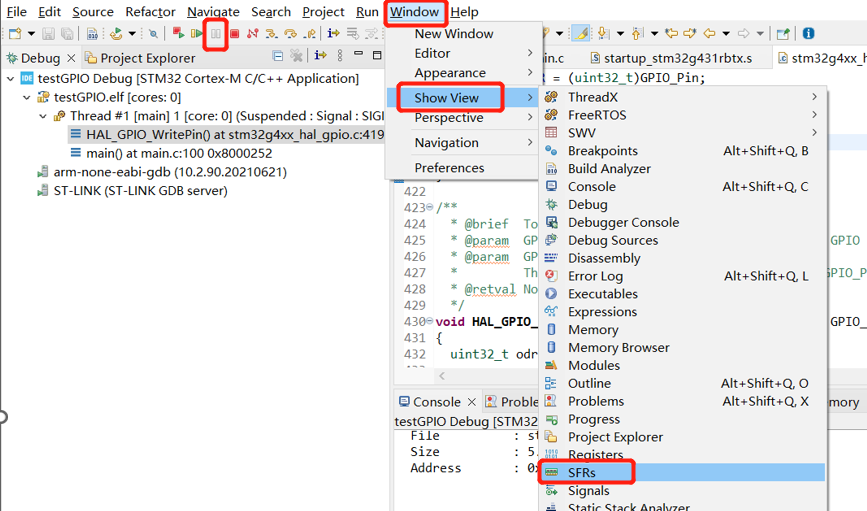


图33 查看MCU内部模块寄存器

**3) 修改模块寄存器**

如图34，在SFRs窗口找到GPIOA模块中的输出数据寄存器ODR的ODR5（D5位），在MSB和LSB的位图中，操作其中的D5位，则可以看到LED随着变化。

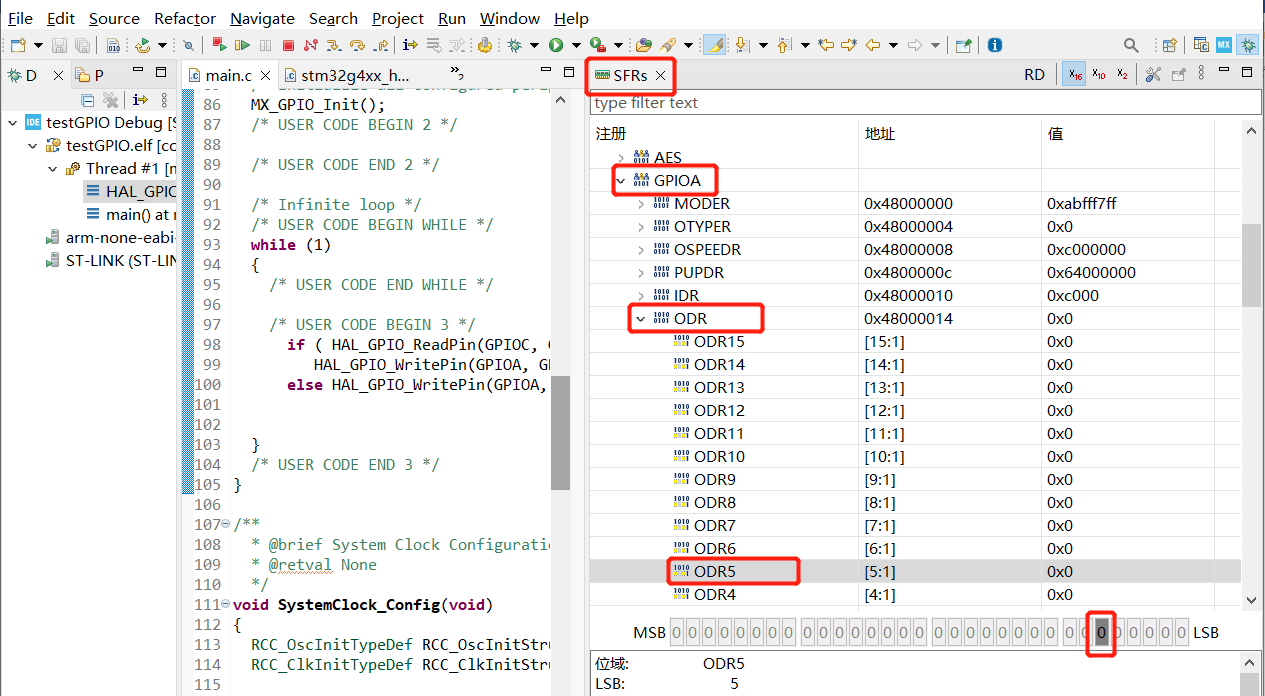


图34 修改MCU内部模块寄存器

**4) 在程序中设置断点、查看程序中的变量等操作，与CCS类似。**（省略，可以自己试试）

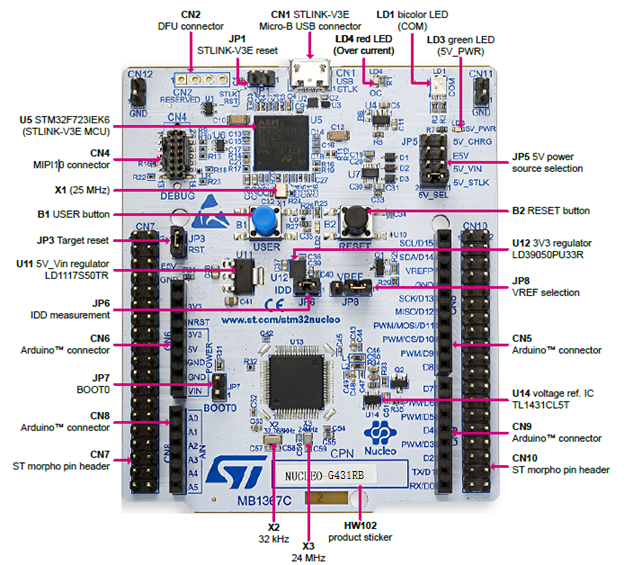
**17. 退出DEBUG**

如图35，点击图标，退出Debug。

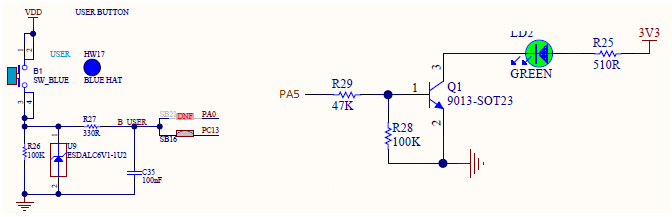


图35 退出图标

**至此，看到如附件“使用入门-单片机现象.mp4”的现象，说明实验平台搭建好了。**



Nucleo-G431RB开发板



Nucleo-G431RB开发板上按键和LED原理图