第一章：搭建STM32开发环境

工具链：CubeMX + Clion + arm-none-eabi-gcc + Jlink

一个简单的介绍：

CubeMX是代码生成器，可以基于我们的配置生成C语言源代码

Clion及其内置的Cmake、makefile等功能可以协助我们处理多文件工程，将C代码编译成机器码

arm-none-eabi-gcc是交叉编译器。虽然Clion内置了gcc，但内置的gcc编译出来的机器码只能在电脑上运行。但实际我们需要在单片机上跑，所以才需要交叉编译器。

jlink用于将机器码烧录进单片机

📌 **CubeMX**

https://www.st.com/stm32cubemx/

Clion

（UIUC邮箱/学信网认证可免费使用）

https://www.jetbrains.com/clion/

*安装时要勾选“open folder as clion project”，其他选项也建议勾上

arm-none-eabi-gcc

（建议使用最新版本13）

https://developer.arm.com/downloads/-/arm-gnu-toolchain-downloads/

jlink

https://www.segger.com/downloads/jlink/

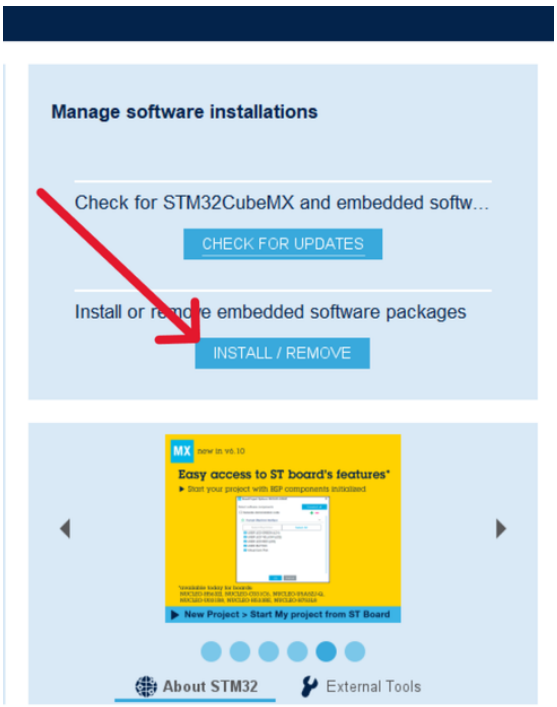
Ozone：和jlink源生兼容，拥有有丰富图形化界面的强大调试软件

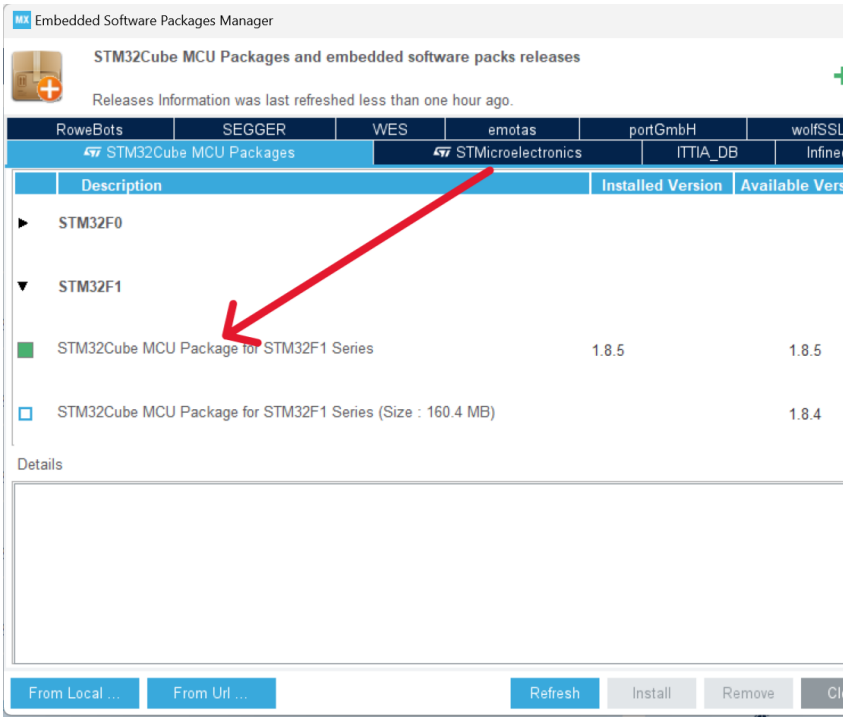
https://www.segger.com/downloads/jlink/#Ozone

点亮stm32f103c8t6

信息收集

打开CubeMX。第一次使用需要下载一些支持包。如图所示。

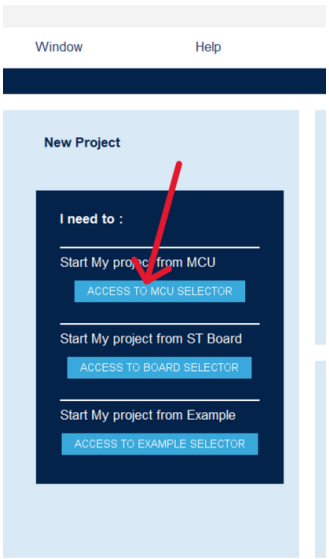


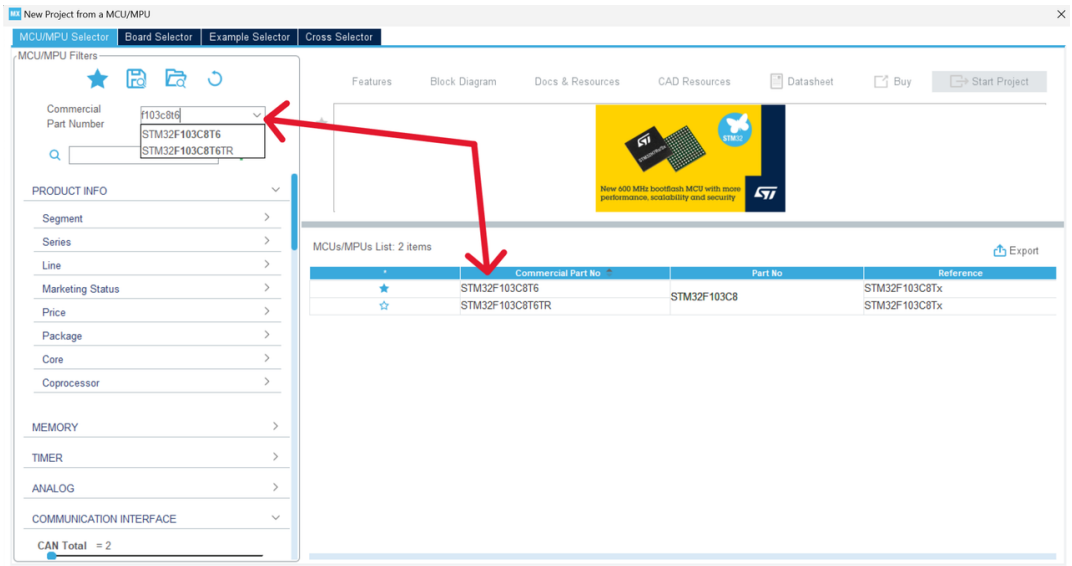


第一步 选择板子型号

在主界面，点击**ACCESS TO MCU SELECTOR**

选择正确的型号：**STM32F103C8T6**，双击新建工程

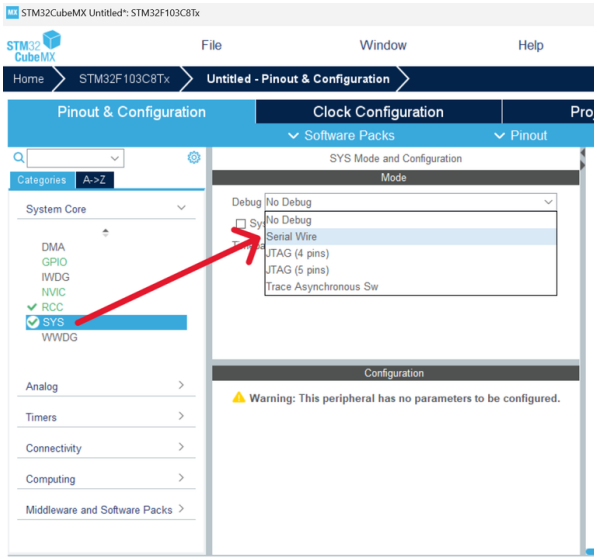


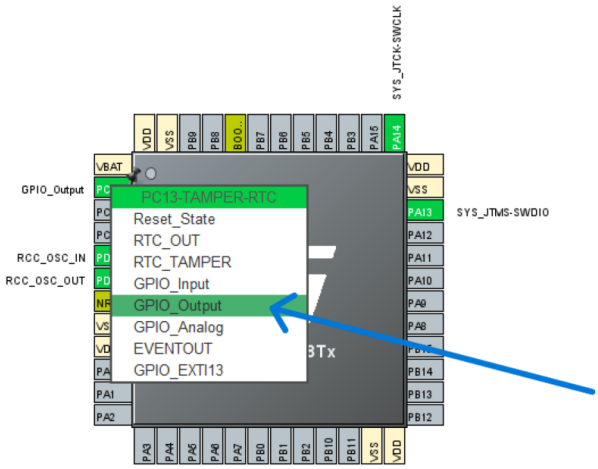


第二步 基本点灯配置

✅ 找到**Pinout & Configuration** → **System Core** → **SYS** → **Debug**，设置为：**Serial Wire**（这步必须做，否则板子无法再次烧录）

在右侧**Pinout veiw**里找到LED的引脚：**PC13**（和最小系统板上标注的是一致的），将其设置为：**GPIO_Output**





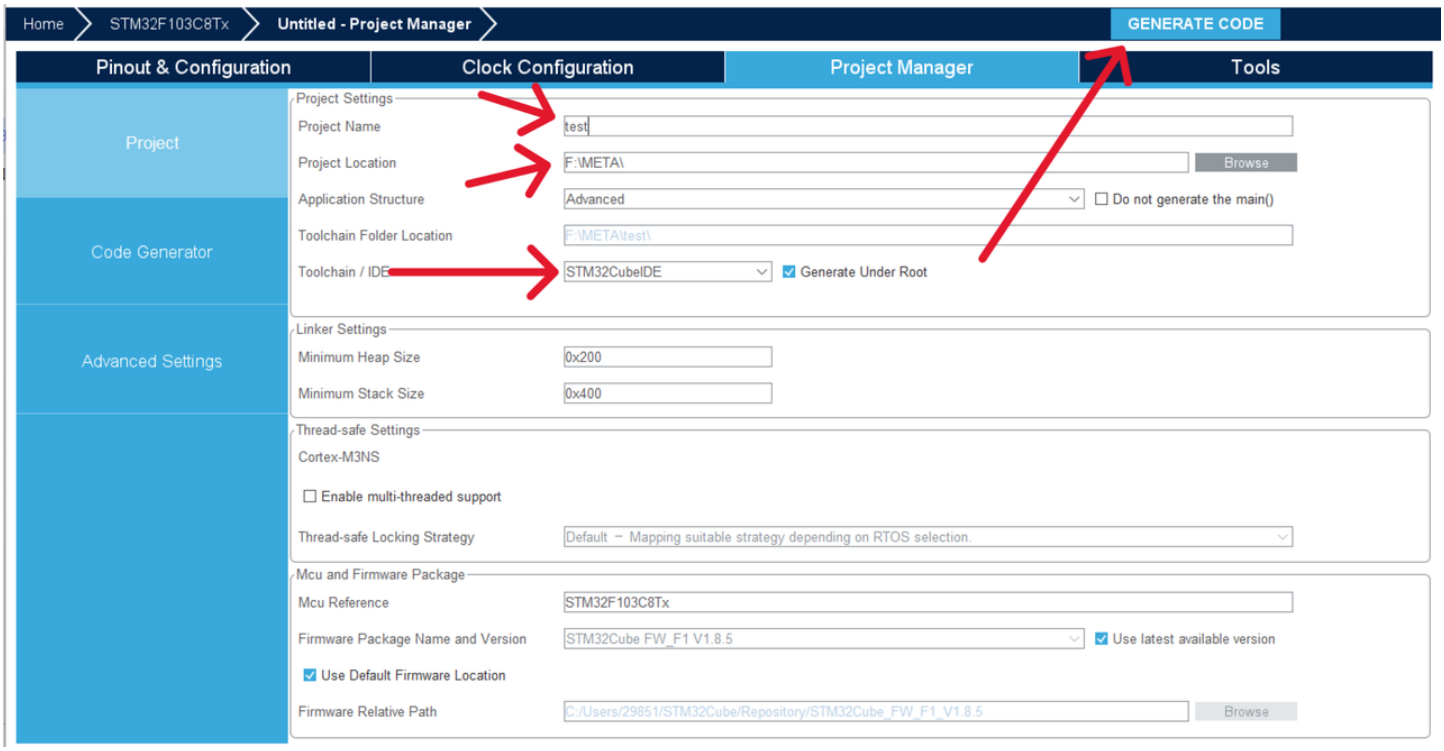
第三步 生成工程

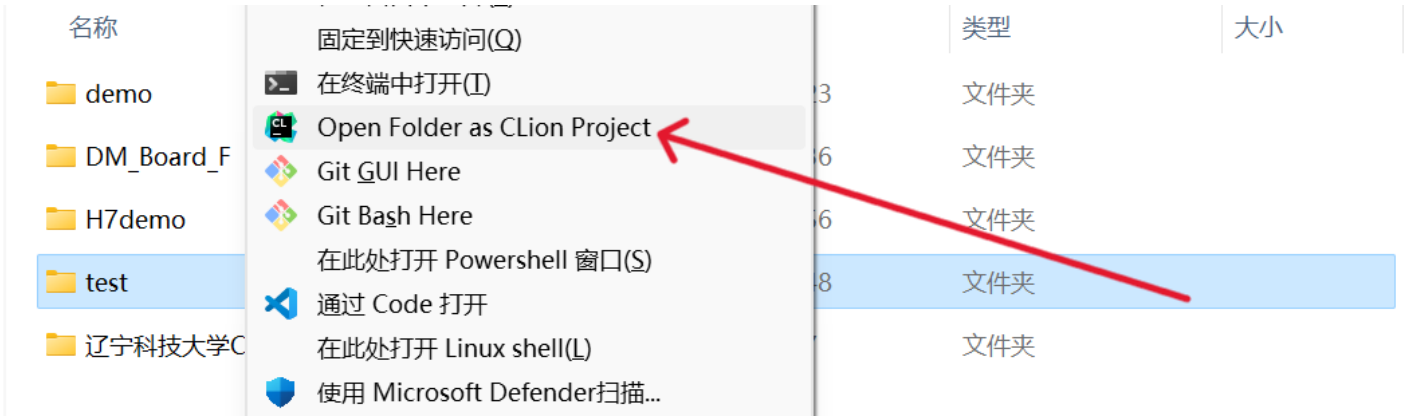
自行配置工程地址。Toolchain / IDE选择：**STM32CubeIDE**

其他配置保持默认即可

点击右上角的GENERATE CODE

用Clion打开



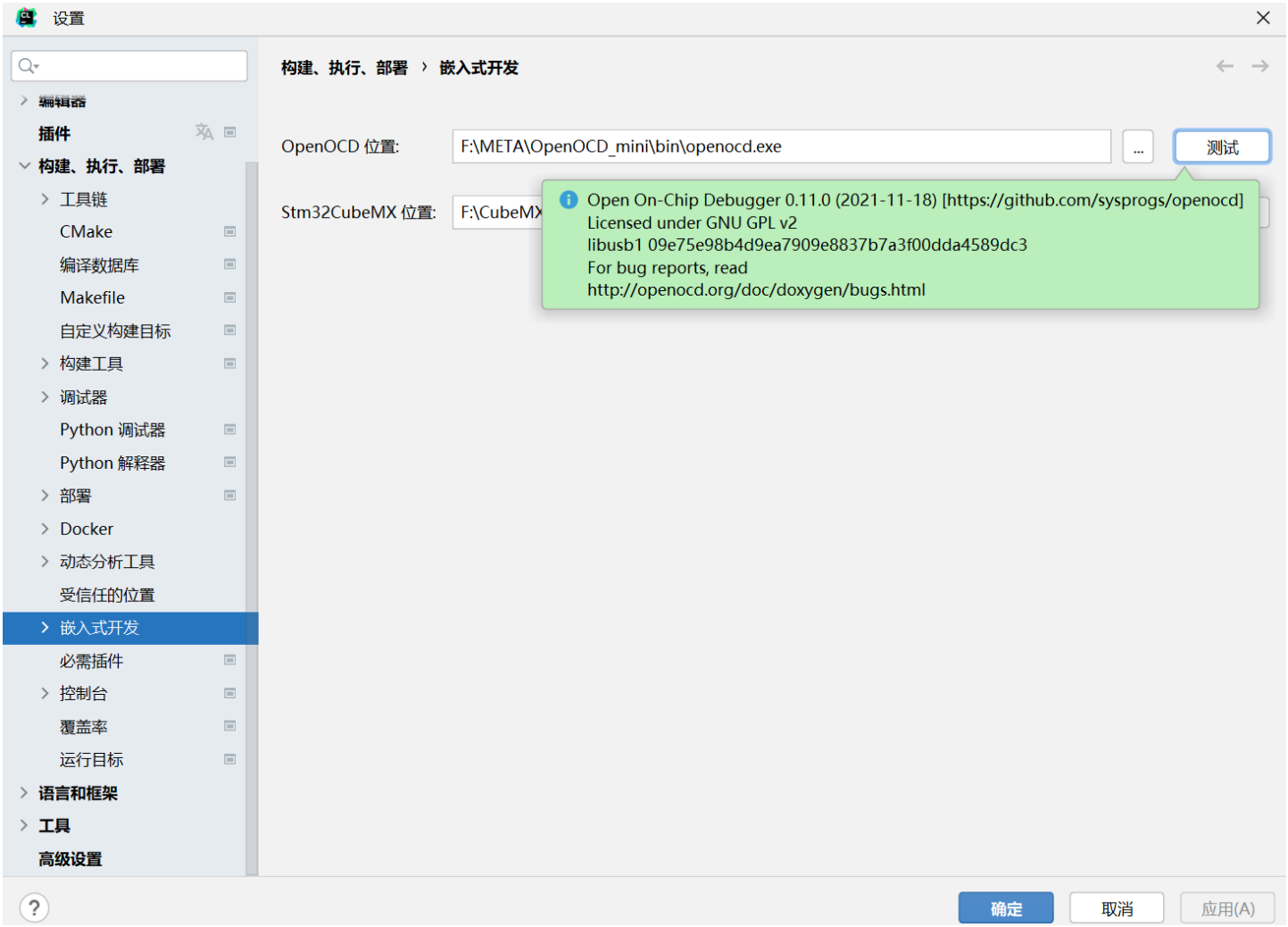


第四步 CLion内部设置

Clion默认英文，可以下载汉化插件改成中文

打开 **设置** → **构建、执行、部署** → **嵌入式开发**，配置CubeMX位置

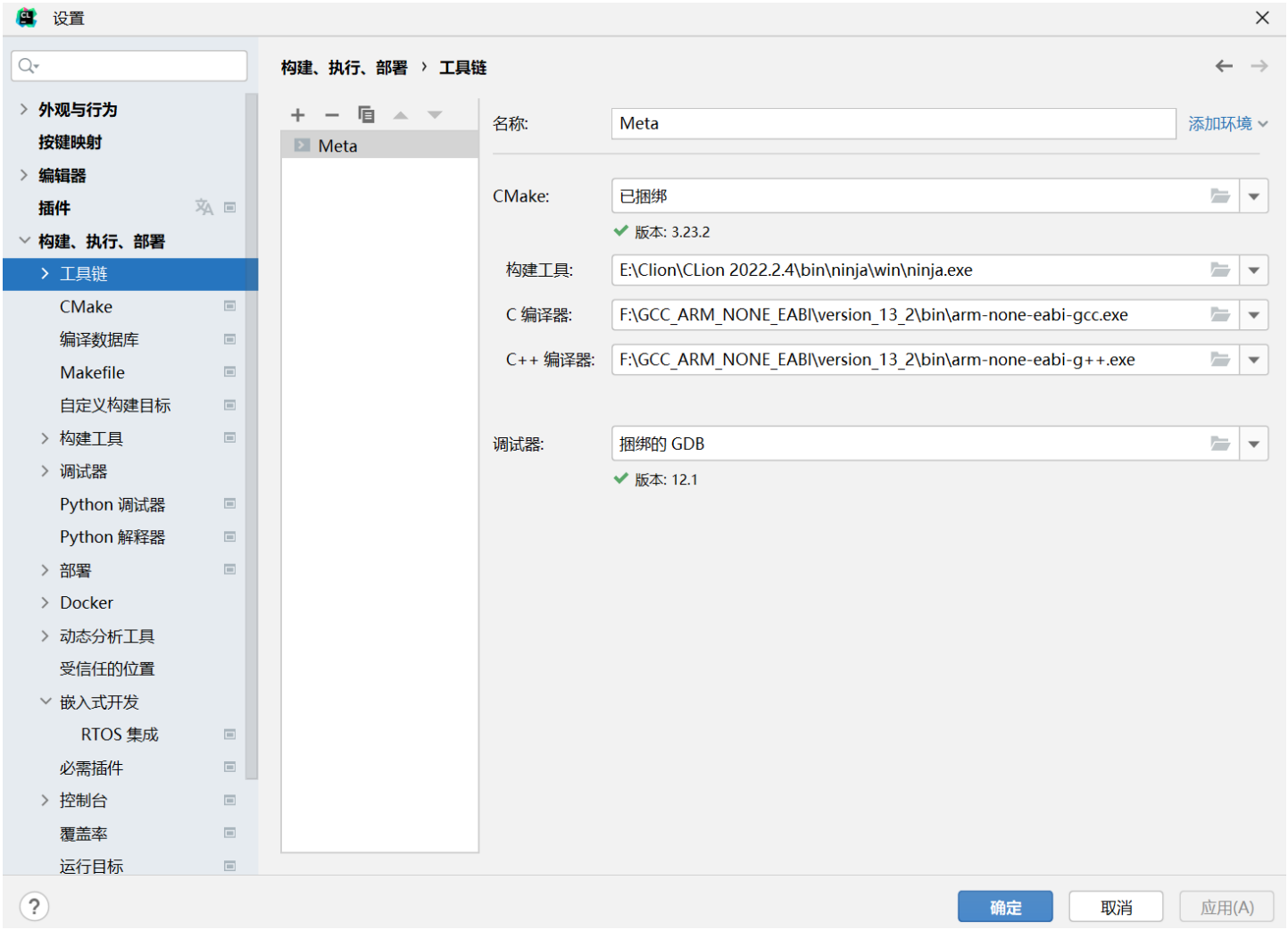
OpenOCD_mini.zip OpenOCD和Jlink GDB Server是差不多的东西，非必需



打开 **设置** → **构建、执行、部署** → **工具链**

新建一个系统工具链，取一个名字，比如Meta

> 📌 **Cmake：**Clion自带，可以在Clion的子目录中找到
>
> **构建工具：**用ninja，clion自带，**相对位置**参考下图，如果没有请自行下载：Releases · ninja-build/ninja
>
> **C/C++编译器：**arm-none-eabi-gcc.exe，arm-none-eabi-g++.exe，**相对位置**可以参考下图
>
> **调试器：**clion自带（不好用，容易出bug，一般也用不到，调试基本用的都是**Jlink+Ozone**）

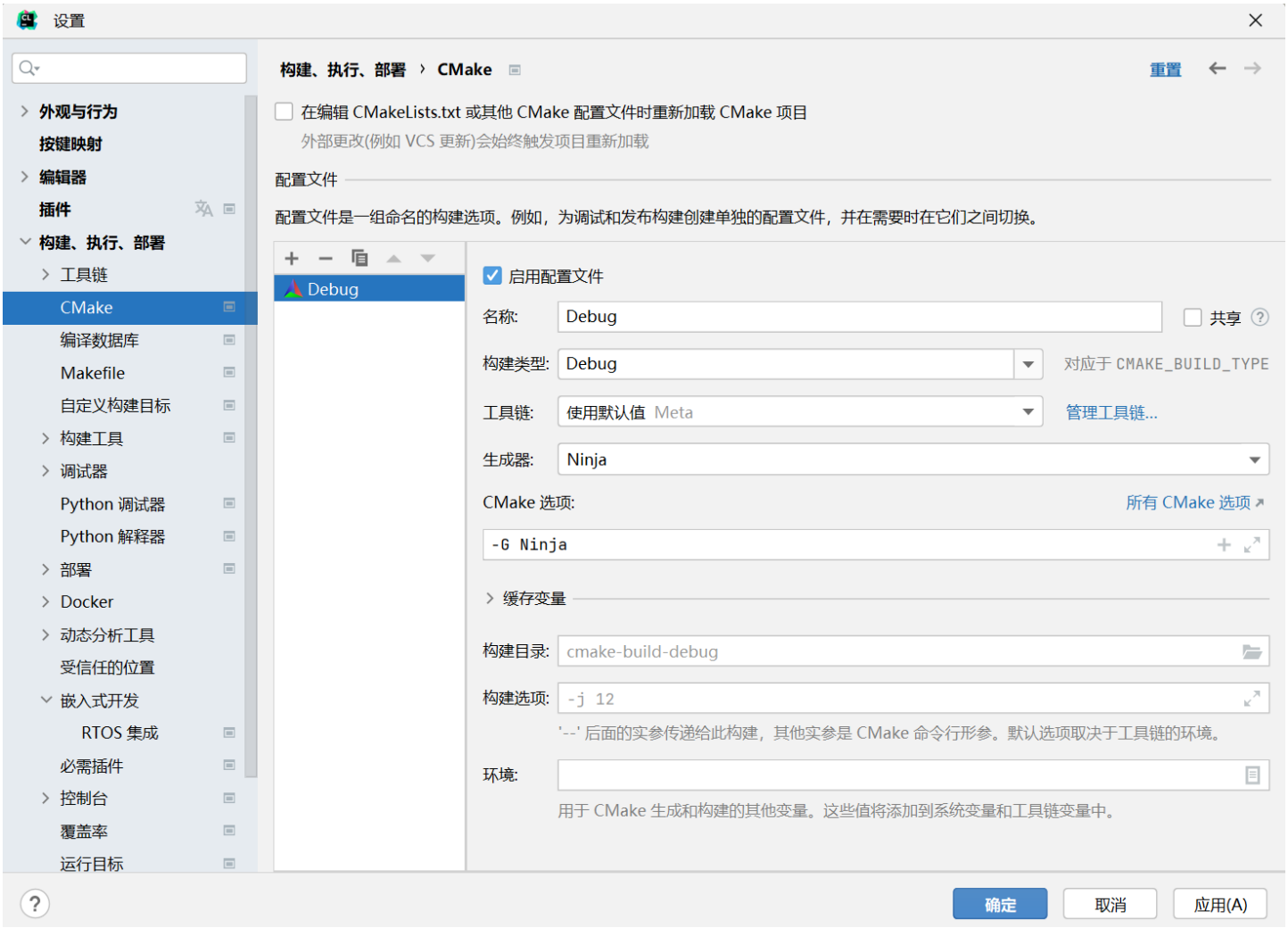


设置 → 构建、执行、部署 → Cmake

新建配置文件，模式为Debug，工具链采用你上面配置的名称，**生成器选择Ninja**

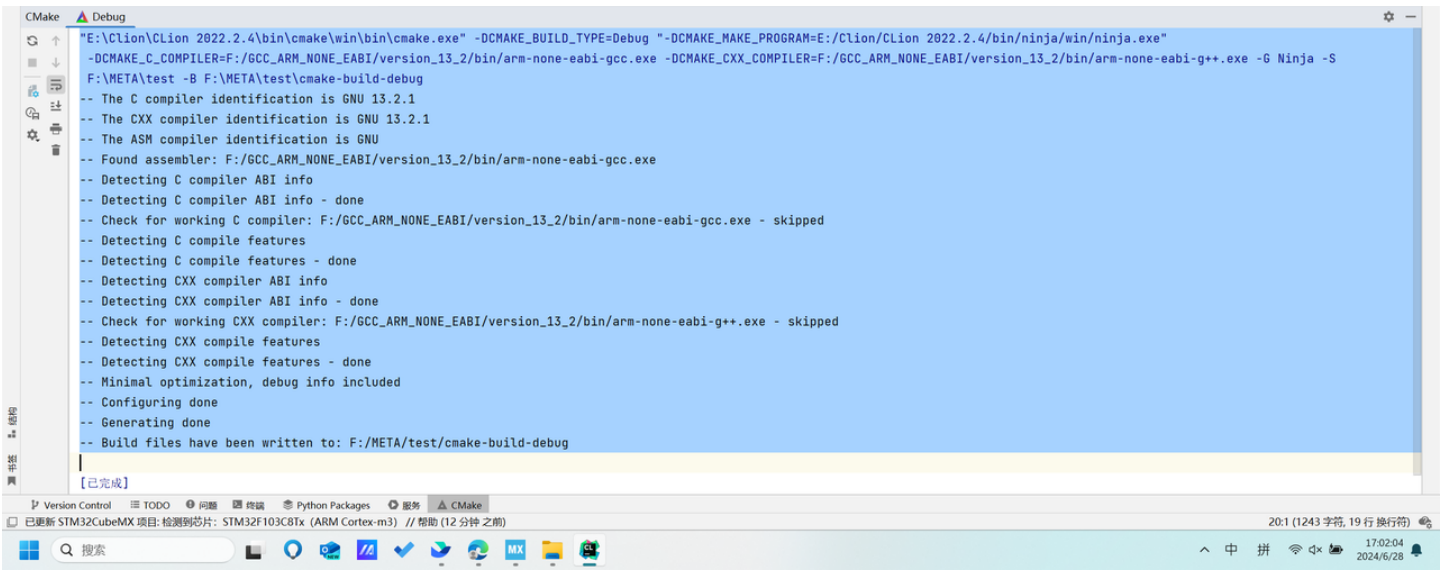
ps：其他的构建方式如Unix makefile都是可以使用的，这个看个人喜好

“构建目录”就是存放各种编译出来的临时文件的地方；构建选项 "-j N" 意为使用N个线程进行编译



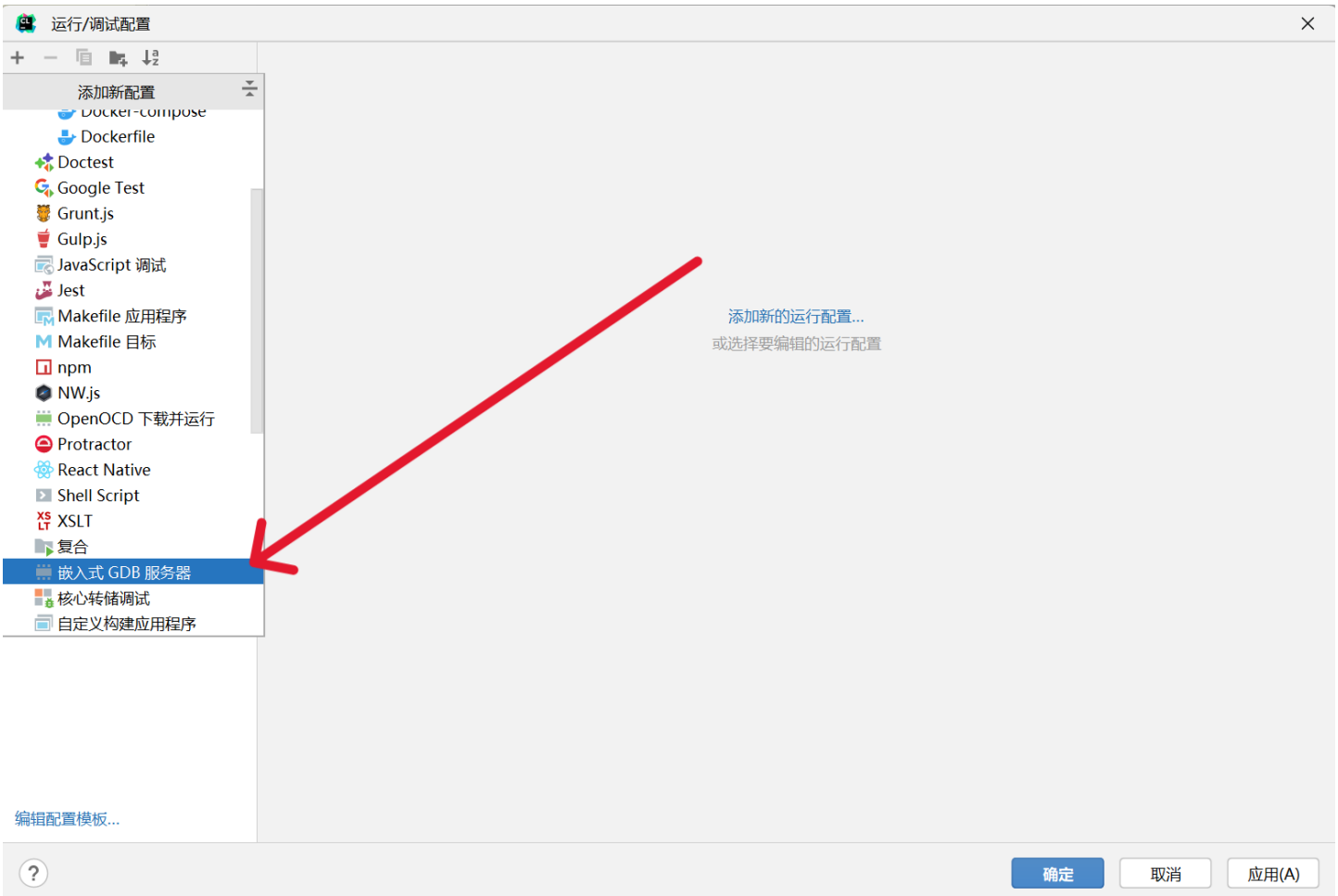
配置完后点击“应用”、“确定”

推出后应当有如下反馈信息（下图蓝色区块）



此时我们点击右上角下拉框

添加配置“**嵌入式GDB服务器**”

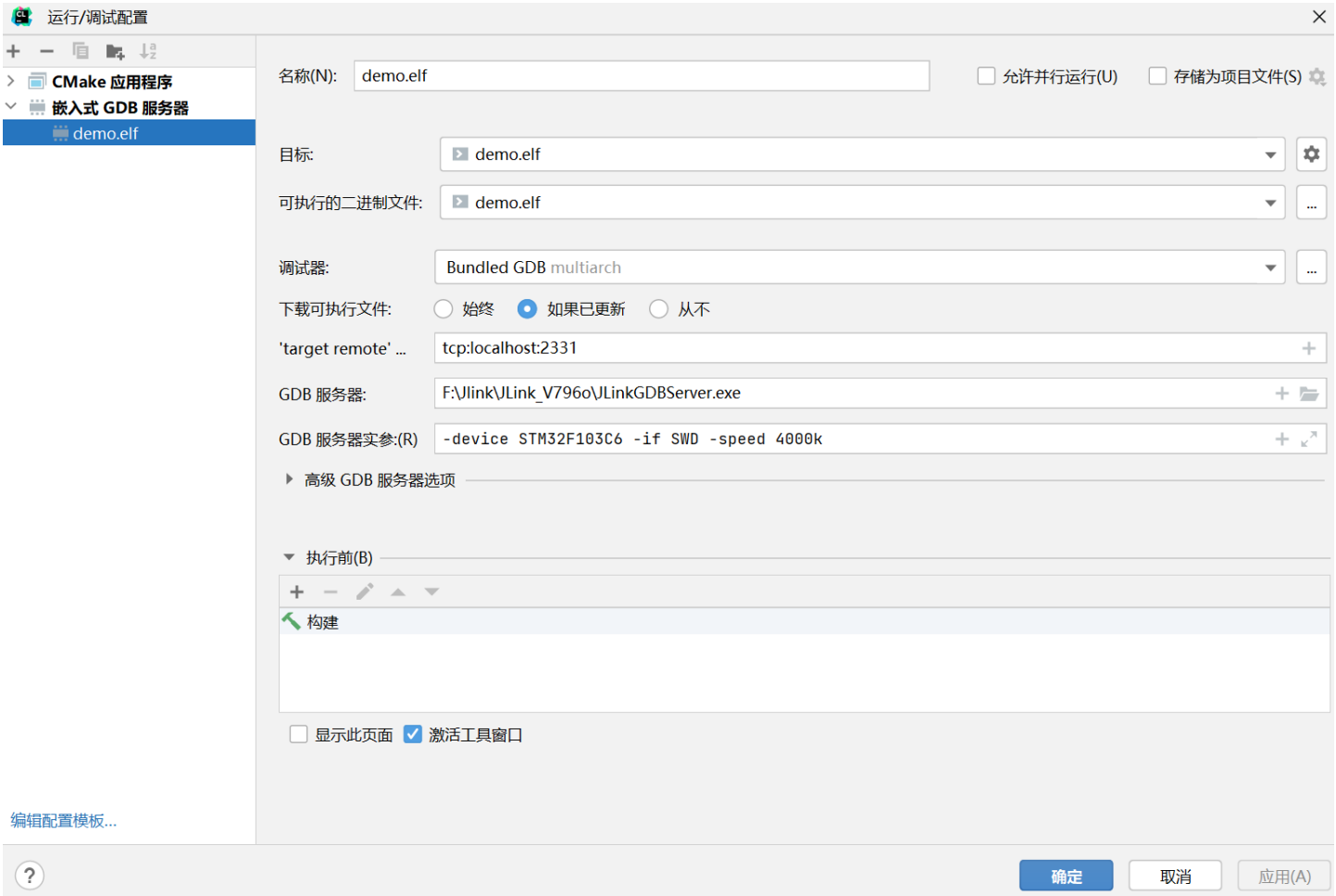


📌 可执行二进制文件：和目标同名的.elf文件

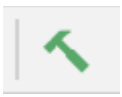
'target remote' args: **tcp:localhost:2331**

GDB服务器：在jlink下载目录里，相对位置参考下图

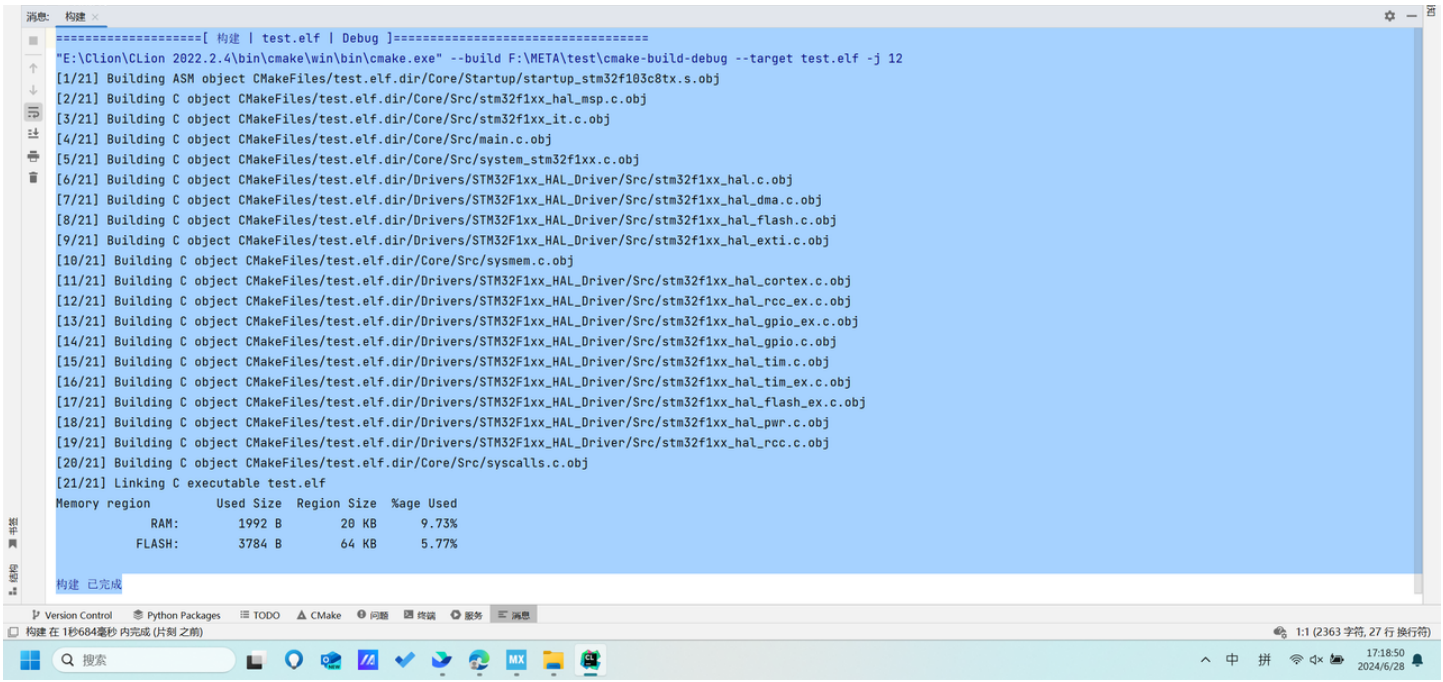
GDB服务器实参：**-device STM32F103C6 -if SWD -speed 4000k**



完成后退出，**下拉框选择你刚刚完成的配置**，点击下拉框左侧的锤子



成功后下方**消息**一栏会显示编译过程和结果



此时可以看到 工程根目录/cmake-build-debug 中出现了.elf和.bin，这些就是用来烧录的

将板子通过Jlink连接到电脑，3v3(VCC) GND CLK SWDIO 一一对应即可。点击下拉框右侧的调试键就可以直接下载

PS：如果Jlink GDB Server烧录一直不成功，可以尝试使用OpenOCD+stlink/daplink进行烧录，配置过程这里不展开，网上到处都是。

第五步 点灯

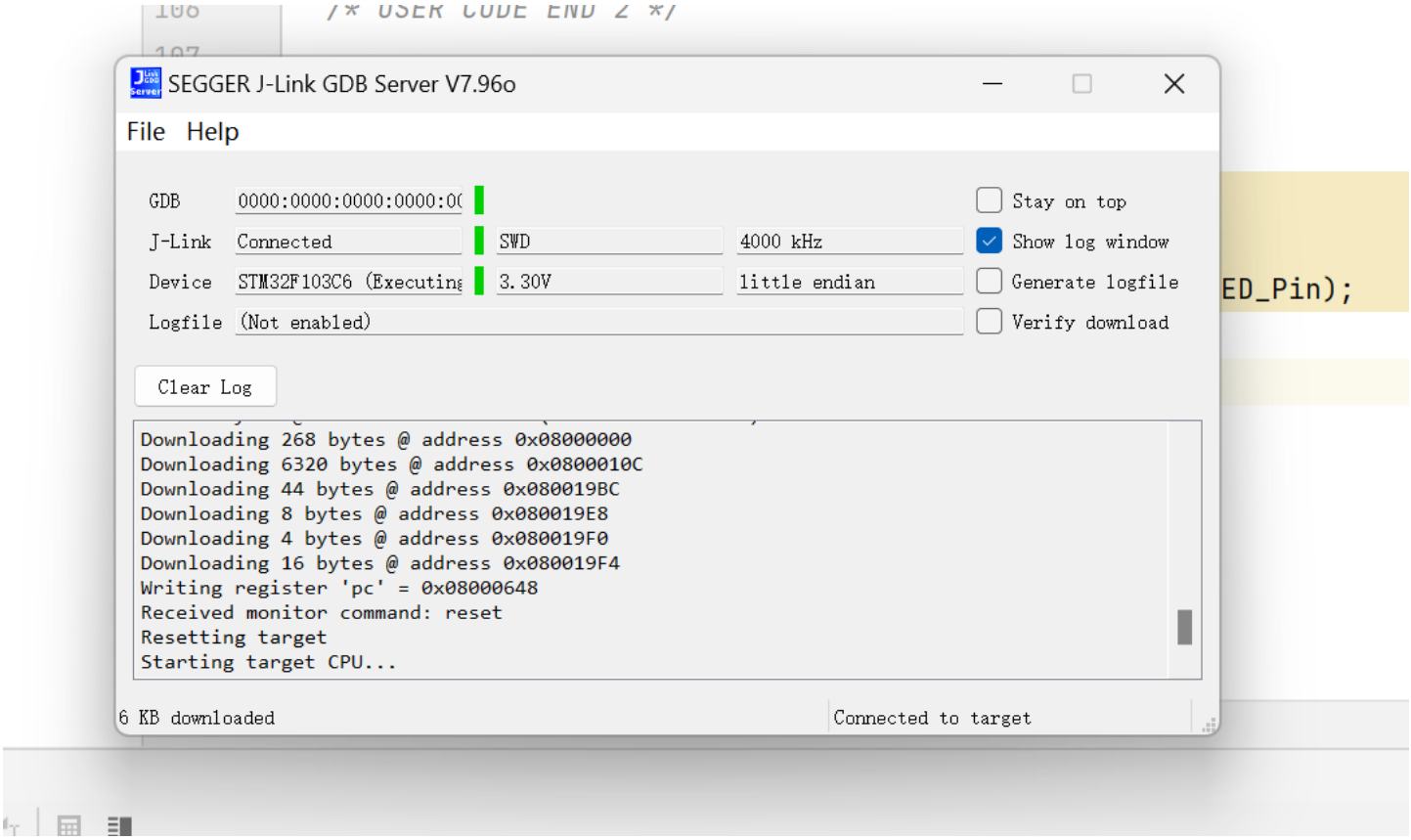
在Clion左侧的工程目录里找到Core/Src/main.c

找到main函数中的while循环，加上这两行：

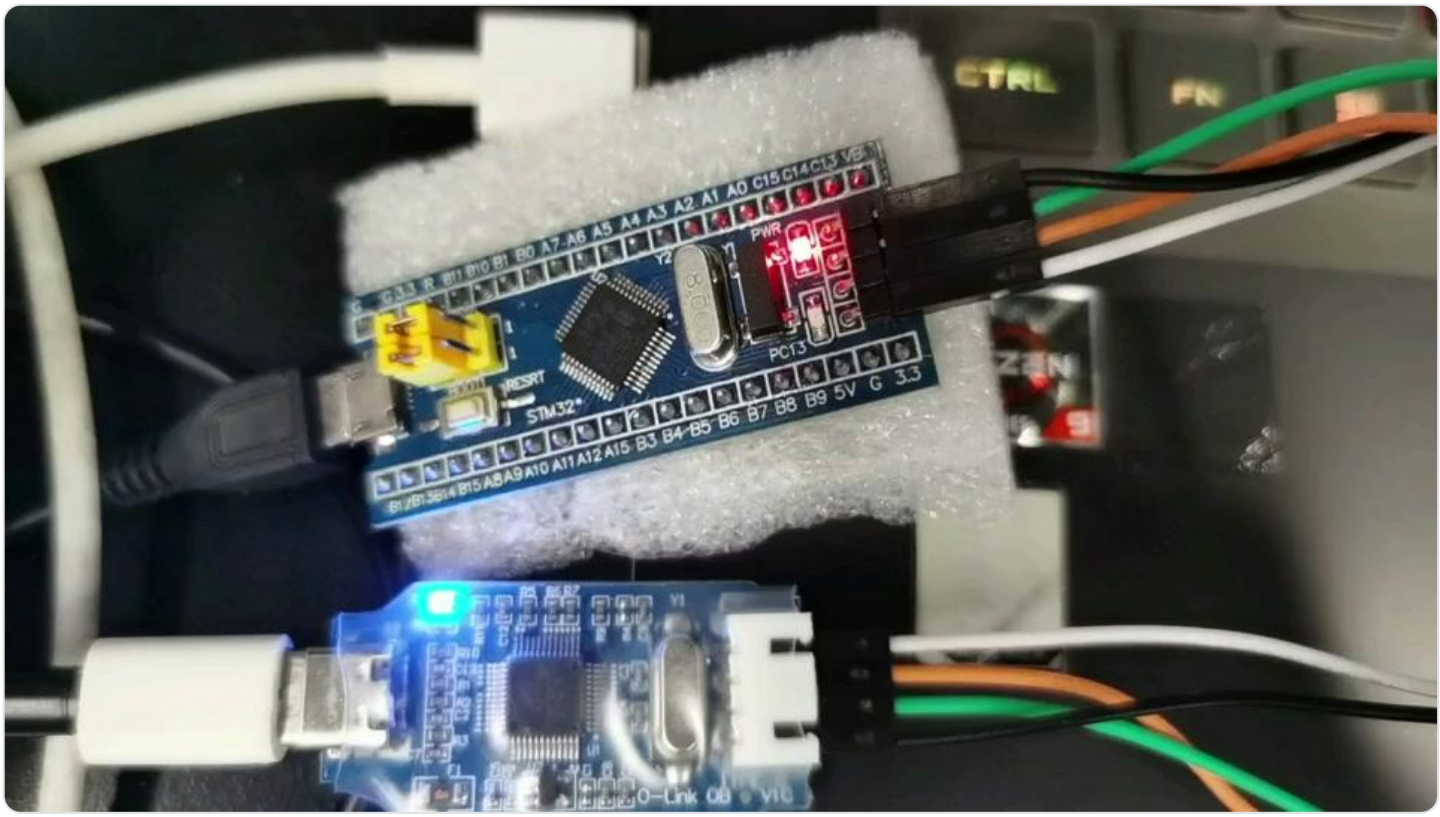
```
1 HAL_Delay(500); //延时500毫秒
2 HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_13); //反转PC13引脚
```

> 💡 **注意要写在 /* USER CODE BEGIN */和 /* USER CODE END*/之间，否则再次生成代码时会被CubeMX覆盖。**

点击调试键执行烧录操作：



可以看到LED以1Hz频率闪烁：



关于GPIO，可以看下面的视频，了解概念即可，不需要完全掌握电路原理

https://www.bilibili.com/video/BV1fu411a74Q

8分钟动画视频带你直观了解STM32 GPIO接口工作原理，内容很干！GPIO是什么？能用来做什么？八种工作模式，推挽输出和开漏输出的区别_哔哩哔哩_bilibili

8分钟动画视频带你全面了解STM32 GPIO接口工作原理，内容很干!GPIO是什么?能用来做什么?八种工作模式?不同模式的应用…

*本章节无需验收，但请务必跟上进度，80%的问题都可以自行上网查找解决。你可以在评论区里提问，如果有共性问题我会统一解答。

下一章节预习

https://www.bilibili.com/video/BV1ph4y1e7Ey

【STM32】超清晰STM32时钟树动画讲解_哔哩哔哩_bilibili

本期视频我们一起来看看STM32中的时钟源与时钟树.----------学习套件获取: https://b23.tv/sblUuvI 或者直接某宝搜索& quot;keysking stm32& quot;, 认准店铺& quot;波特律动&