基于Makefile的STM32跨平台开发指南

环境配置

系统要求：Windows Mac Linux均可

作者尝试过Windows 10 20H2和Ubuntu 20.04系统，都是可以运行的。

软件安装：

STM32Cube 运行环境：Java

- **Windows**：前往https://www.oracle.com/java/technologies/downloads/下载最新版Java（这里是Java17。具体在Java SE Development Kit 17.x.x downloads项目下找到Windows，然后下载x64 Installer，一路下一步安装即可。
- **Linux**：打开终端，输入

```
sudo apt-get insall openjdk-17-jdk
```

安装即可。或者也可以去上方官网下载。

请尽量安装Java最新版。不要安装Java 8。

STM32CubeMX：

前往ST官网https://www.st.com/en/development-tools/stm32cubemx.html下载，有Linux、Mac和Windows版本。注意需要注册并登陆一个ST账号。请务必下载最新版（修复了CAN的BUG）。

Linux安装好之后可以自己创建一个快捷方式。建议安装在home目录下。

编译器：gcc-arm-none-eabi

- **Windows**：前往https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads 下载最新版并安装。请选择后缀win32的exe安装包。

安装过程中，在结束页面请务必勾选“Add path to environment variable”复选框。（建议全部勾选）。

完成后打开cmd或者powershell，输入

```
arm-none-eabi-gcc -v
```

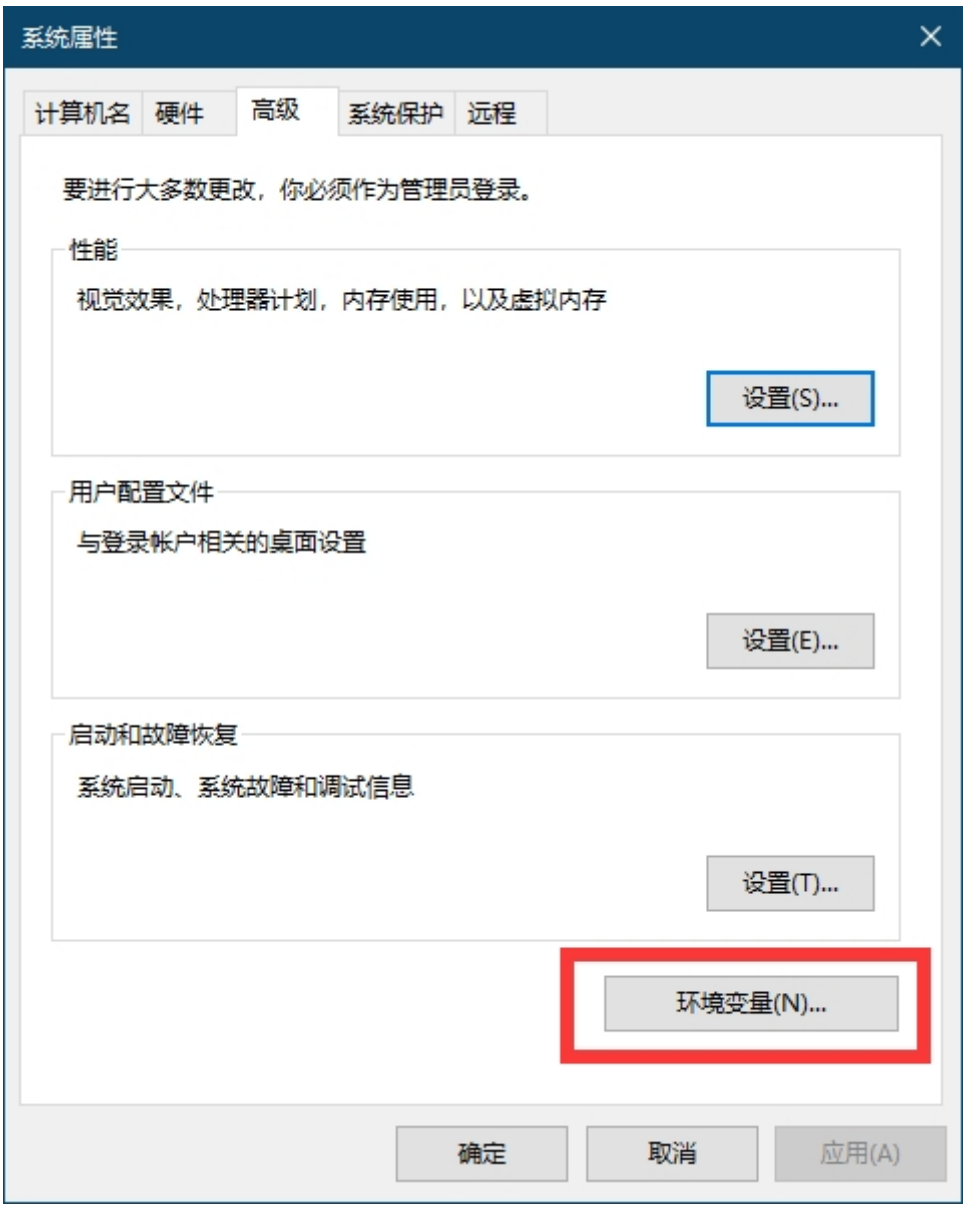
如果最后一行显示了gcc version xx.x.x 这样的就表示成功添加了环境变量。

如果提示找不到则需要手动添加，如果找到了，可直接跳转至下一环节

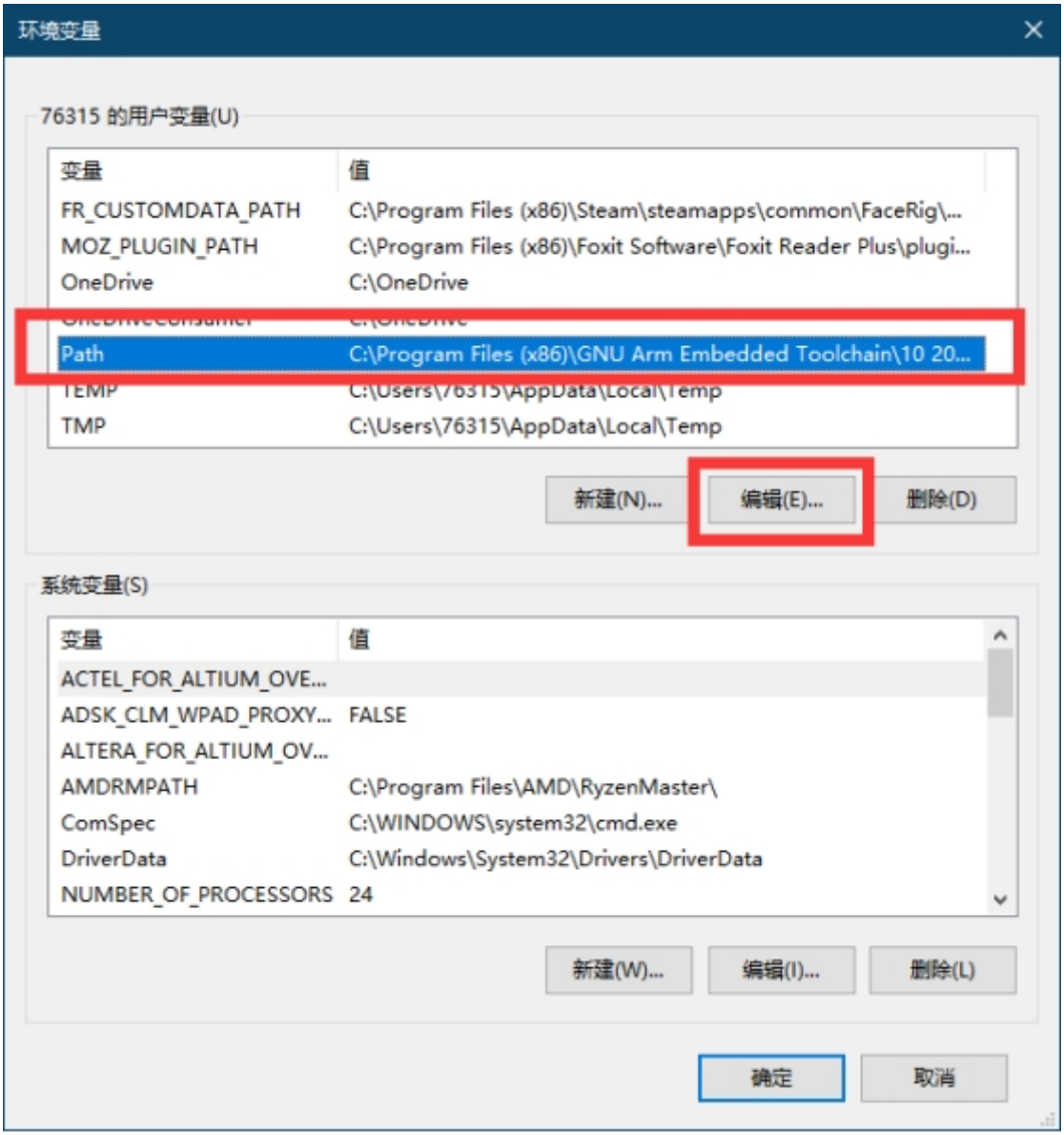
手动添加步骤如下：

打开设置-系统-关于，找到高级系统设置：

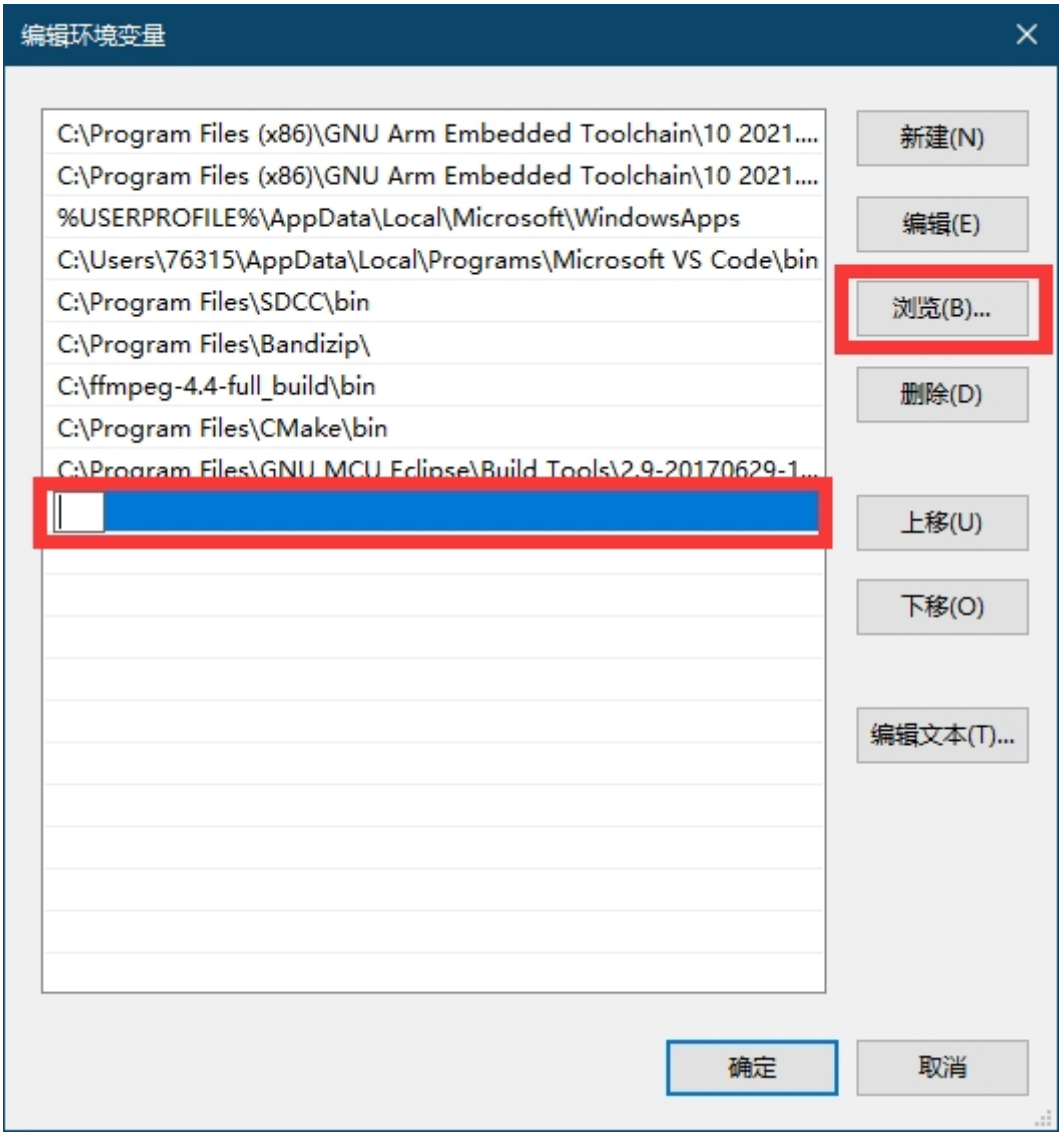
点击“环境变量”：


系统属性
计算机名
硬件
高级
系统保护
远程
要进行大多数更改，你必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与登录帐户相关的桌面设置
设置(E)...
启动和故障恢复
系统启动、系统故障和调试信息
设置(T)...
环境变量(N)...
确定
取消
应用(A)

找到path，点击“编辑”：

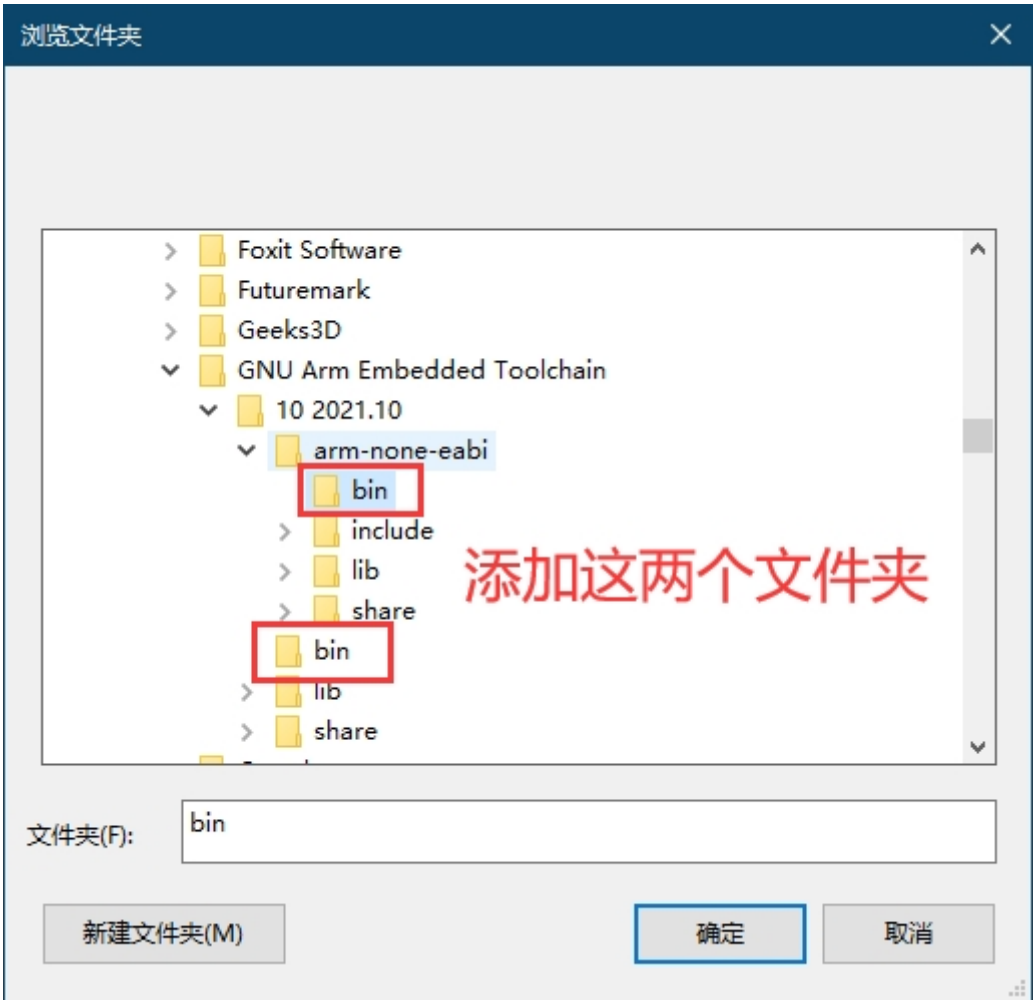


双击击最下面的空白行，显示文本框后，点击“浏览”：



找到你的安装目录（一般是C:\Program Files (x86)\GNU Arm Embedded Toolchain(版本号)\），选中bin文件夹，并点确定。

重复上述，再添加（安装目录）\arm-none-eabi\bin文件夹。



然后全部点确定关闭窗口。

完成后再次打开cmd或者powershell，输入

```
arm-none-eabi-gcc -v
```

检查是否添加成功。

- **Linux**：打开终端，输入

```
sudo apt-get install gcc-arm-none-eabi
```

执行即可。可以使用

```
arm-none-eabi-gcc -v
```

指令检查是否安装成功。（方法请参考Windows）

Make工具（仅限Windows）：gnu-mcu-eclipse-build-tools

前往GitHub下载：https://github.com/xpack-dev-tools/windows-build-tools-xpack/releases

找到最新版（不是Pre-Release，请务必选择Latest），点击标题进入下载页面，下载x64.zip压缩包，之后解压到C盘根目录下。

请参照上述手动添加环境变量的方法将你解压的文件夹中的bin目录添加到环境变量（一般路径是C:\xpack-windows-build-tools-(版本号)\bin）。

完成后再次打开cmd或者powershell，输入

```
make -v
```

检查是否添加成功。

Linux自带GNU Make工具，不需要这一步骤。

IDE：VScode

前往官网https://code.visualstudio.com/下载。网页会自动根据你的操作系统确定版本。

安装好之后需要安装以下必要插件（拓展，Extensions）：

- C/C++
- ARM
- Makefile Tools

其他的插件请根据自身需要安装。这里不给出建议。

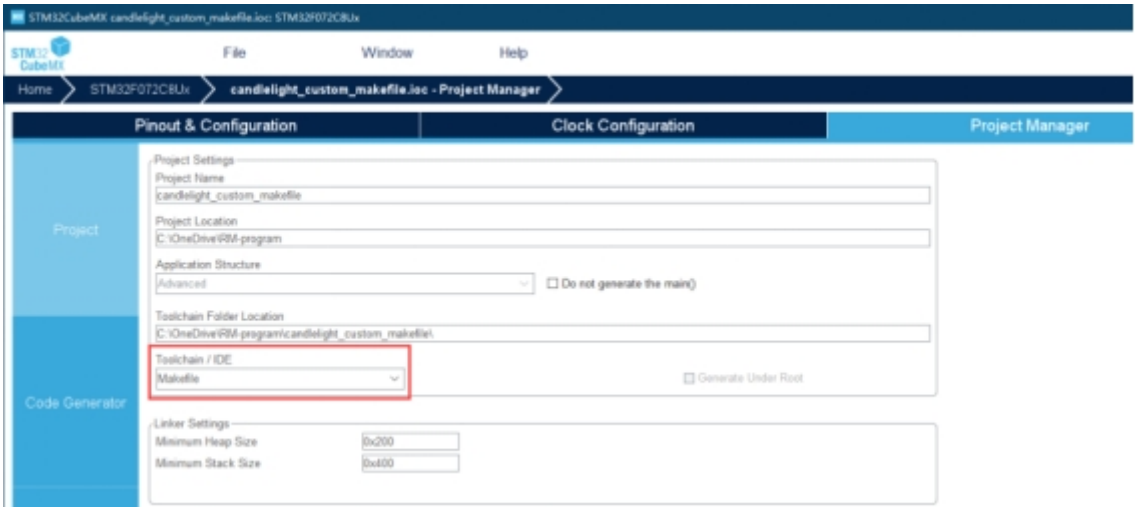
调试工具：Ozone

前往官网https://www.segger.com/downloads/jlink/，找到Ozone - The J-Link Debugger，选择对应操作系统下载安装即可。

工程配置：

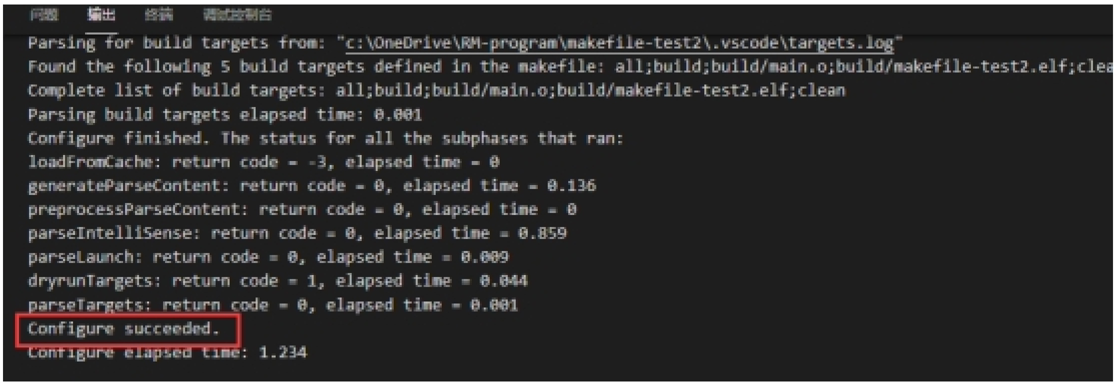
使用STM32Cube新建一个工程，按照通常的方法配置。

在Project Manager选项卡中，参照下图配置：



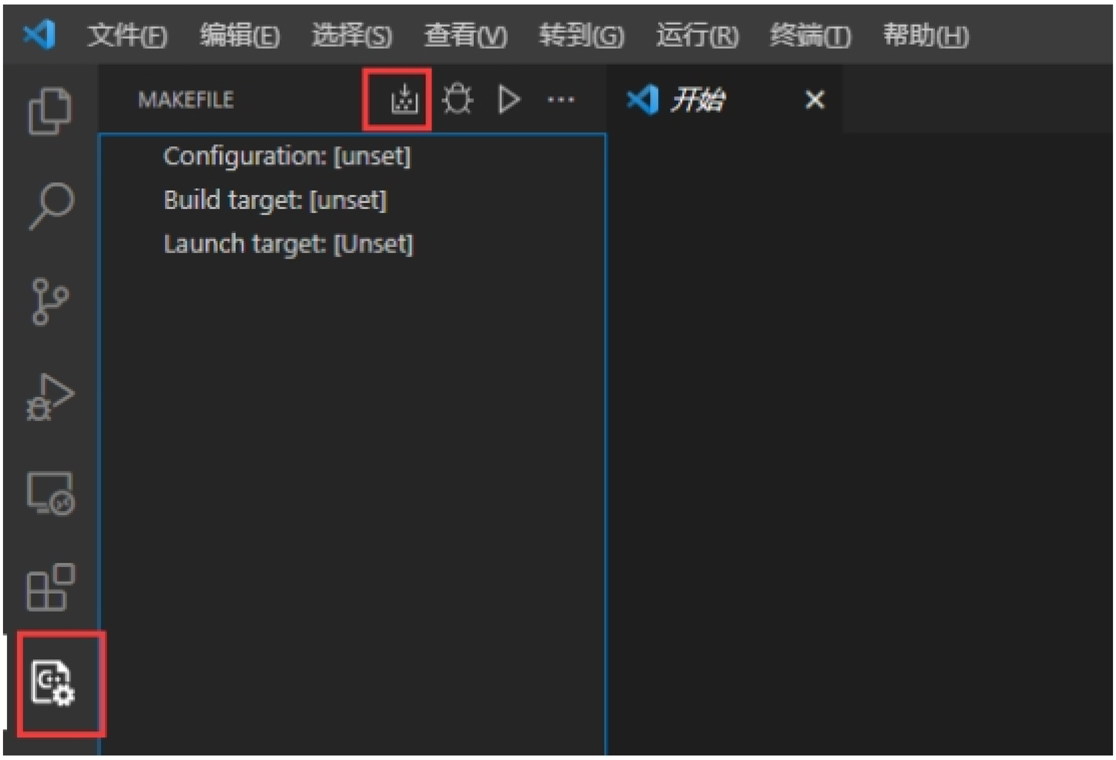
然后生成工程。

使用VScode打开工程文件夹。在打开之前请先确认你已经安装了必要插件，并配置好了环境变量。新版本的VScode需要你选择信任此文件夹。如果没有问题的话，VScode的输出界面应该会出现以下提示：

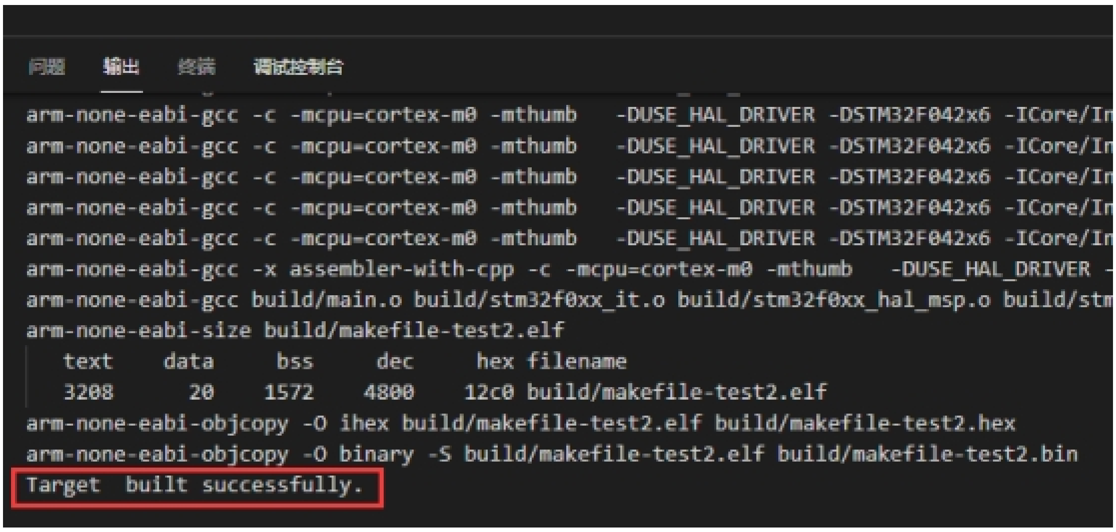


出现这个提示代表工程已经完成了。

在左侧的侧边栏中找到Makefile选项，可以找到build按钮，点击即可编译。



编译完成后会显示以下输出：



此时在工程目录下即可找到build文件夹，文件夹内有后缀elf的调试用可执行文件和后缀bin的二进制烧录文件。

- 添加新源代码和头文件目录：

使用VScode打开工程目录下的Makefile文件：

```
#######################################
# source
#######################################
# C sources
C_SOURCES =  \
Core/Src/main.c \
Core/Src/stm32f0xx_it.c \
Core/Src/stm32f0xx_hal_msp.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_tim.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_tim_ex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_rcc.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_rcc_ex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_i2c.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_i2c_ex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_gpio.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_dma.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_cortex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_pwr.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_pwr_ex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_flash.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_flash_ex.c \
Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_exti.c \
Core/Src/system_stm32f0xx.c
```

在这里参照上面的格式可以添加源代码文件。

```
# C includes
C_INCLUDES =  \
-ICore/Inc \
-IDrivers/STM32F0xx_HAL_Driver/Inc \
-IDrivers/STM32F0xx_HAL_Driver/Inc/Legacy \
-IDrivers/CMSIS/Device/ST/STM32F0xx/Include \
-IDrivers/CMSIS/Include
```

在这里参照上面的格式可以添加头文件目录。请在目录前加上-I前缀。

使用STM32Cube重新生成工程并不会影响你已经添加的文件和目录。