目 录

使用 vscode 开发 stm32 实战

Vscode+clang+openocd+gcc-arm-none-eabi+cubemx+stlink+gdb

开源越来越强，传统开发软件授权壁垒，是这个向导的主因。

学习新知识、天天向上，生命在于折腾，是次因。

习惯了 windows 下傻瓜式的 IED 环境，对 linux 有恐惧，想战胜这份恐惧。所以结合网上丰富的非正规教程，就成了这篇备忘录。

一、 工具集下载

1.1、 vs code 下载

源于微软的一款文本编辑器，由于免费和丰富的插件生态，主要用来程序编写、调试。

软件链接 https://code.visualstudio.com/Download

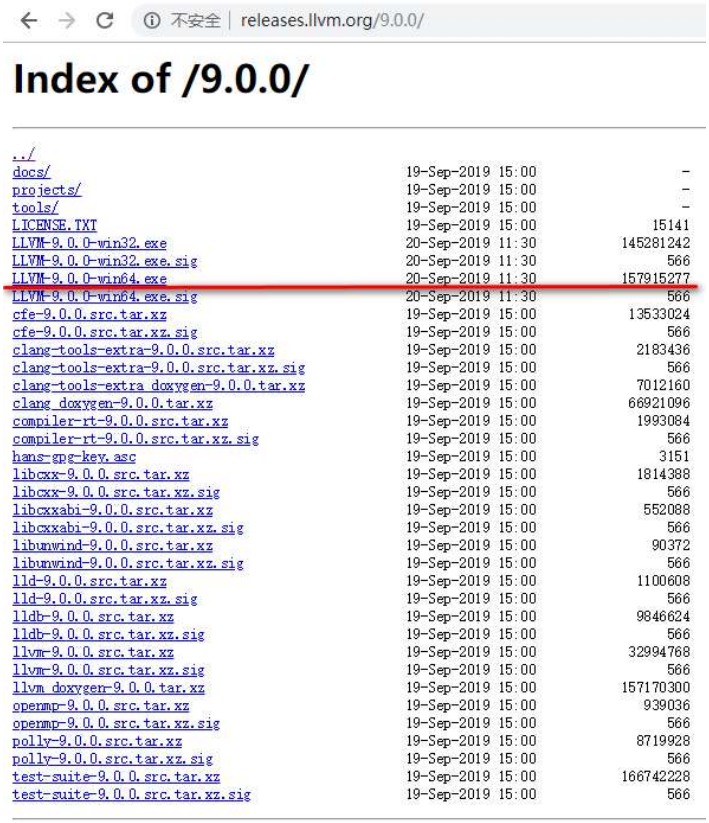
安装过程 windows 风格，注意有个复选框，添加“右键打开文件和文件夹”选项即可。

1.2、 clang 下载

其另一个名字交 LLVM,用于提供语法补全代码格式化等功能，正常完成安装后将软件的安装目录下的 bin 文件夹设置进入环境变量 PATH 中。

链接：http://releases.llvm.org/9.0.0/

后边是版本号，直接修改版本号即可进入对应版本的下载链接。进入网页后根据自己系统和需求下载对应的文件。我直接下载了 exe 安装包。安装后添加 bin 目录到系统 path 环境变量。



不安全 | releases.llvm.org/9.0.0/

Index of /9.0.0/

../		
docs/	19-Sep-2019 15:00	-
projects/	19-Sep-2019 15:00	-
tools/	19-Sep-2019 15:00	-
LICENSE.TXT	19-Sep-2019 15:00	15141
LLVM-9.0.0-win32.exe	20-Sep-2019 11:30	145281242
LLVM-9.0.0-win32.exe.sig	20-Sep-2019 11:30	566
LLVM-9.0.0-win64.exe	20-Sep-2019 11:30	157915277
LLVM-9.0.0-win64.exe.sig	20-Sep-2019 11:30	566
cfe-9.0.0.src.tar.xz	19-Sep-2019 15:00	13533024
cfe-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
clang-tools-extra-9.0.0.src.tar.xz	19-Sep-2019 15:00	2183436
clang-tools-extra-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
clang-tools-extra_doxygen-9.0.0.tar.xz	19-Sep-2019 15:00	7012160
clang_doxygen-9.0.0.tar.xz	19-Sep-2019 15:00	66921096
compiler-rt-9.0.0.src.tar.xz	19-Sep-2019 15:00	1993084
compiler-rt-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
hans-gpg-key.asc	19-Sep-2019 15:00	3151
libcxx-9.0.0.src.tar.xz	19-Sep-2019 15:00	1814388
libcxx-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
libcxxabi-9.0.0.src.tar.xz	19-Sep-2019 15:00	552088
libcxxabi-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
libunwind-9.0.0.src.tar.xz	19-Sep-2019 15:00	90372
libunwind-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
lld-9.0.0.src.tar.xz	19-Sep-2019 15:00	1100608
lld-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
lldb-9.0.0.src.tar.xz	19-Sep-2019 15:00	9846624
lldb-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
llvm-9.0.0.src.tar.xz	19-Sep-2019 15:00	32994768
llvm-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
llvm_doxygen-9.0.0.tar.xz	19-Sep-2019 15:00	157170300
openmp-9.0.0.src.tar.xz	19-Sep-2019 15:00	939036
openmp-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
polly-9.0.0.src.tar.xz	19-Sep-2019 15:00	8719928
polly-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566
test-suite-9.0.0.src.tar.xz	19-Sep-2019 15:00	166742228
test-suite-9.0.0.src.tar.xz.sig	19-Sep-2019 15:00	566

1.3、 git 下载

提供 Git 支持和 MINGW64 指令终端。我主要用 git 带的终端了，名叫 bash.exe

官网下载链接：https://gitforwindows.org/

1.4、 openocd 下载

为什么牛 X 的都是老外，是一个学校的开源项目，十来年了，它的进化版本叫 visualGDB，好像是，收费。提供一个 pc 和调试器之间的一个接口--gdb 服务。用 keil 哪有这么多事儿。这个文件下载下来不需要安装，直接放到一个文件夹下，bin 目录添加系统 path 环境变量就行。

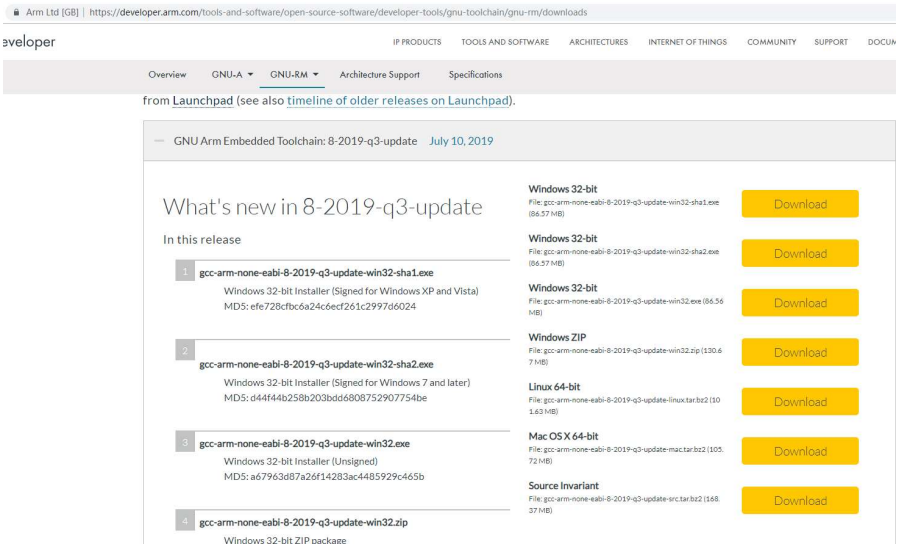
下载链接：http://gnutoolchains.com/arm-eabi/openocd/（还有好多其他的工具也在这里）（有的给的是这个链接，这个里边的是源码，还需要自己编译。https://sourceforge.net/就是这里）

1.5、 arm-none-eabi-gcc 下载

1.6、 arm 的交叉编译工具集，GNU toolchain 下载

下载链接：https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads

我下载的 gcc-arm-none-eabi-8-2019-q3-update-win32.exe 这个。



1.7、 cubemx 下载

stm32 程序初始化神器，ST 官网直接找吧。

1.8、 st cubeprogrammer 下载

用于 stm32 程序下载，ST 官网直接找吧。

1.9、 gnu-mcu-eclipse 下载

stm32 make 工具 gnu-mcu-eclipse，是一个 eclipse 的插件，可以在 windows 下构建 makefile 环境，适合直接在终端界面 make 工程。可以作为 vscode 的辅助使用。

下载链接：https://github.com/gnu-mcu-eclipse/windows-build-tools/releases

1.10、 参考教程

其他参考教程：https://www.jianshu.com/u/b1ffe963c188

https://blog.csdn.net/zhengyangliu123

写的比较好的两位博主。

https://www.brobwind.com/archives/1291（有关 gdb 和 openocd 的）

二、 工具安装注意事项

2.1 vscode

添加“右键打开文件和文件夹”选项，安装好后添加一些常用的插件。

```
Chinese (Simplified) Language Pack for Visual Studio Code    中文界面
c/c++
c/c++ clang command adapter
c/c++ snippets
c++ intellisense
Chinese language… vsc 汉化
Cortex-debug （不在本教程范围，后边再研究，也可以用于调试）
Cortex-debug: device support
Gbk2utf8
Visual studio intellicode 语法支持、智能补全、颜色
```

2.2 clang/LLVM

bin 目录添加到环境变量

2.3 git

正常安装 在 vsc 中使用它的终端

2.4 Openocd

解压后放入自定义目录，然后其 bin 目录添加到环境变量

2.5 arm-none-eabi-gcc

添加“右键打开文件和文件夹”选项

2.6 cubemx

安装好后，下载相应 STM32 芯片的 pack 包

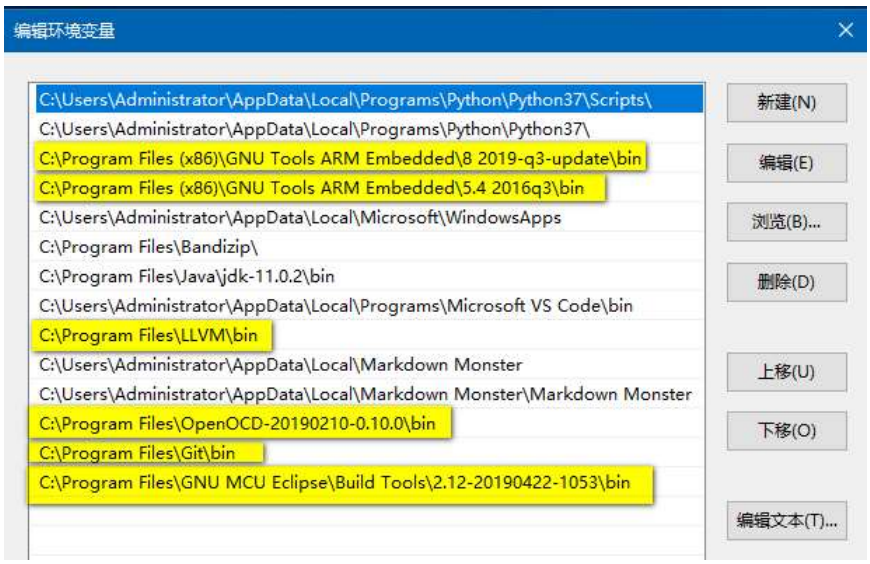
2.7 gun mcu eclipse 插件

解压到自定义文件夹，并添加 bin 目录到系统环境变量

2.8 stm32cubeprogrammer

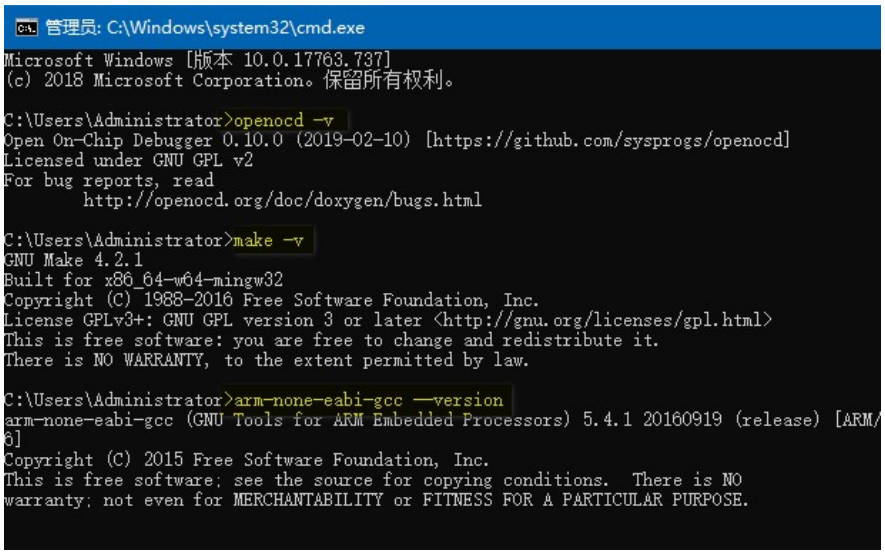
安装好后，将 *STM32_Programmer_CLI.exe* 所在目录添加至系统环境变量。

仅供参考，系统环境变量如下：



三、 软件安装检查

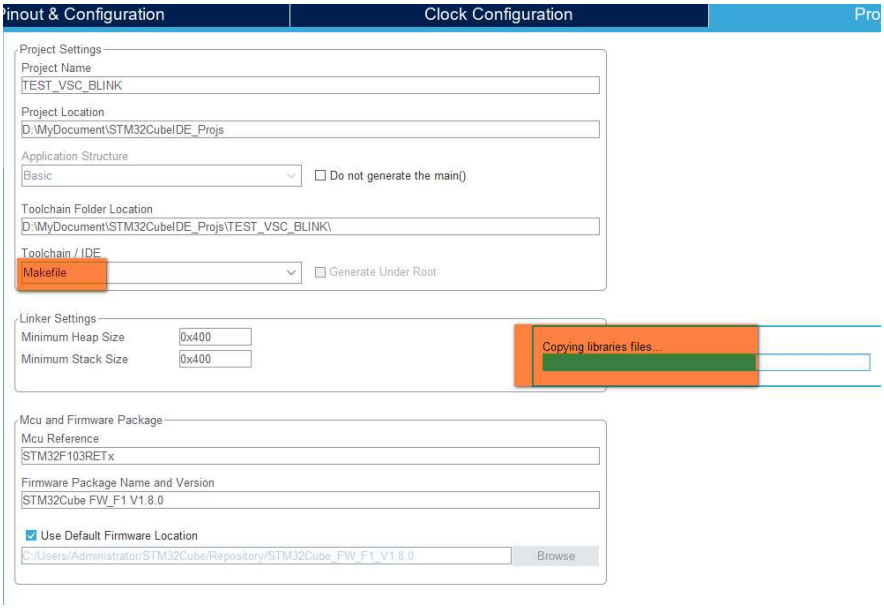
Win+R 输入 cmd，打开终端窗口，验证 make、arm-none-eabi-gcc、openocd 功能。正常查出版本号，说明正确安装。随便一个终端工具都可以用。



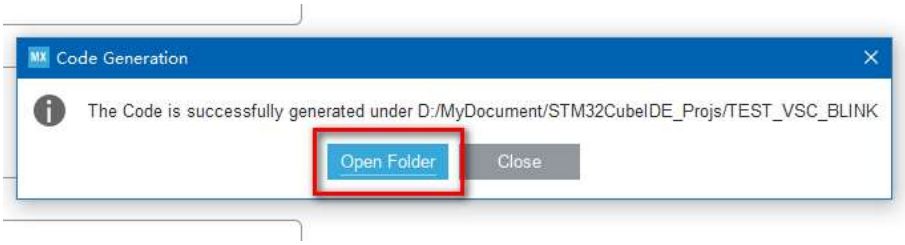
四、 Cubemx 生成独立工程

4.1 建立工程

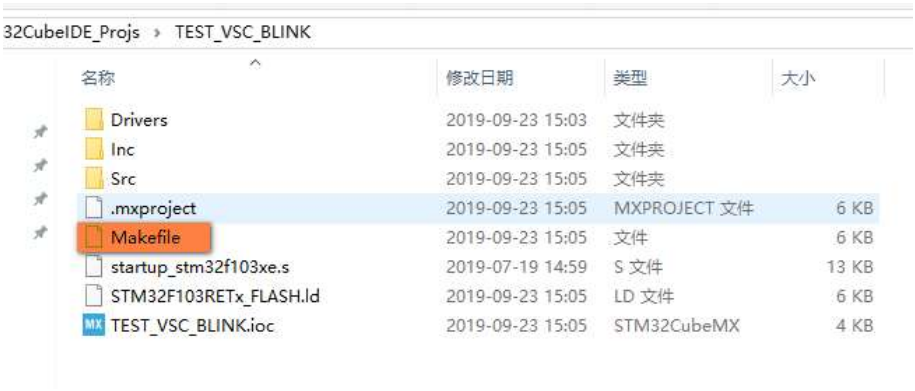
建工程的过程就略了，选择生成的工程类型：**makefile**



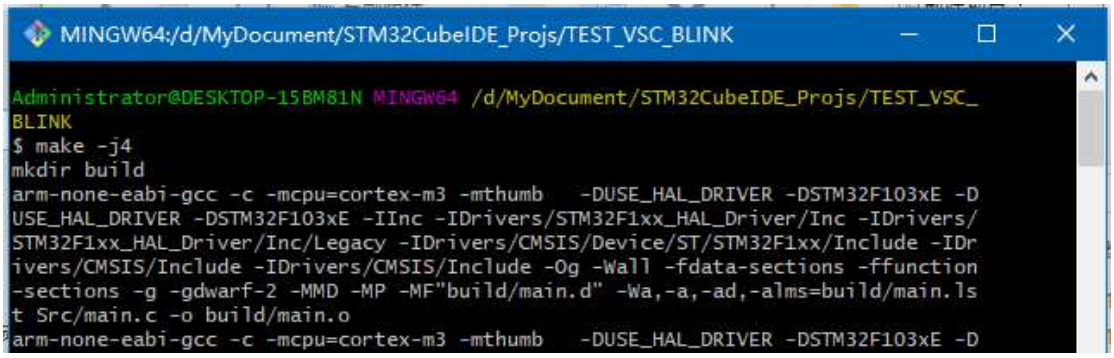
完成之后，选择 openfolder

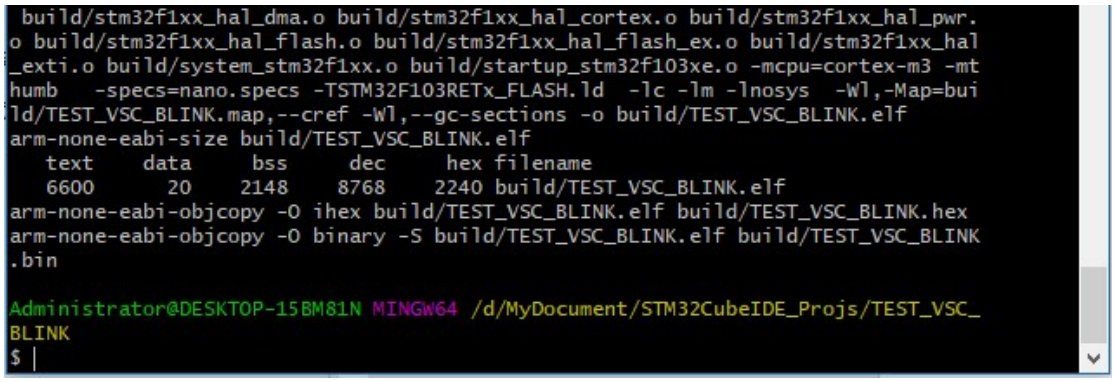


生成的文件夹+文件如下(自动生成了 makefile):



然后再该文件夹下，shift+右键，在弹出的菜单中选择 git bash here 或者在此处打开 powershell 窗口，输入 make -j4

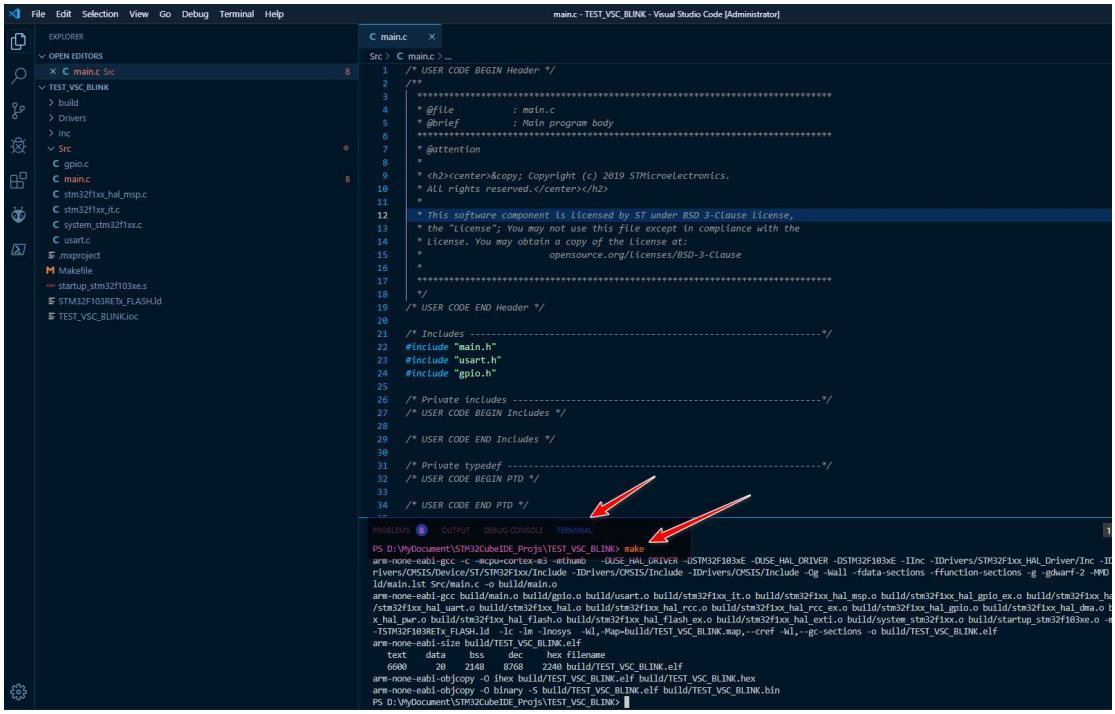




见到如上的信息，说明已配置好 make 环境，并生成了正常的可用的程序。

4.2 使用 vscode 打开工程

在工程文件夹空白处右键，选择“open with code”



在下边箭头所指向的窗口选择 terminal，然后输入 make，正常的话，回正确输出编译过程和最后输出 elf 文件、hex 文件、bin 文件。

那么使用 vscode 开发 stm32 已经迈出了成功的第一步，剩下还有 99 步。

五、 Vscode 配置

Vscode 配置 json 文件实现编译、下载、硬件复位、调试等功能

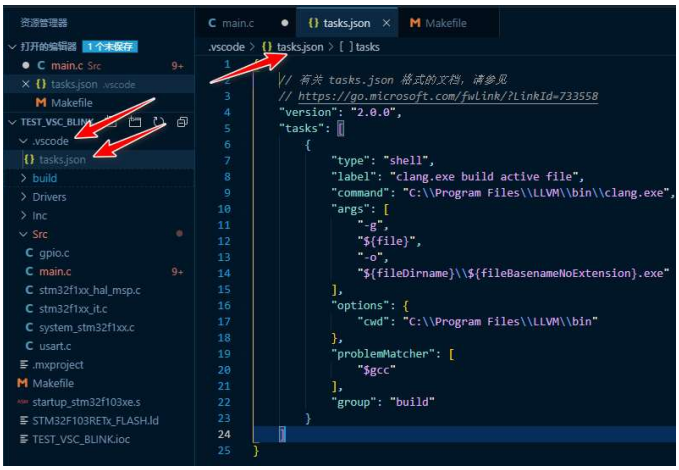
5.1 添加配置文件

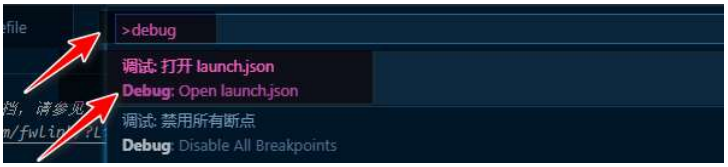
默认情况下，工程下是不含.vscode 的文件夹的，需要在 vscode 下生成这样的文件或者直接从别处拷贝一份过来。

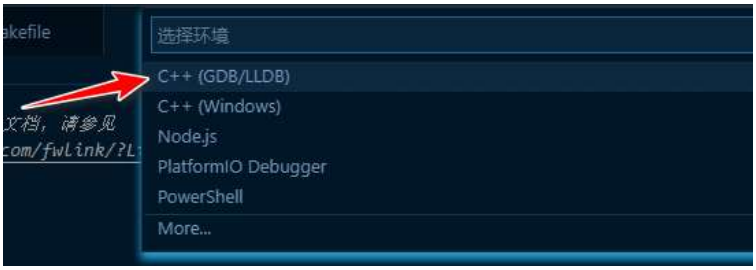
步骤：ctrl+shift+p，输入 tasks，选择 configure task，然后选择 C/C++:clang.exe……，这时候 vsc 会自动创建文件夹，并生成 tasks.json 模板文件，后期进行修改即可。

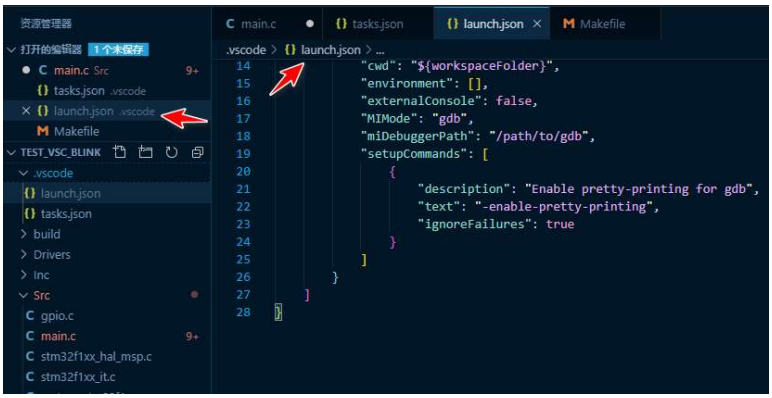
再次使用 ctrl+shift+p, 输入 debug, 选择 open launch.json, 然后选择 C++…, 结果会生成一个 launch.json 的文件。

同样的方式输入 c c++，找到图示的选择或者设置都可以，最终生成一个 c_cpp_properties.json 的文件。

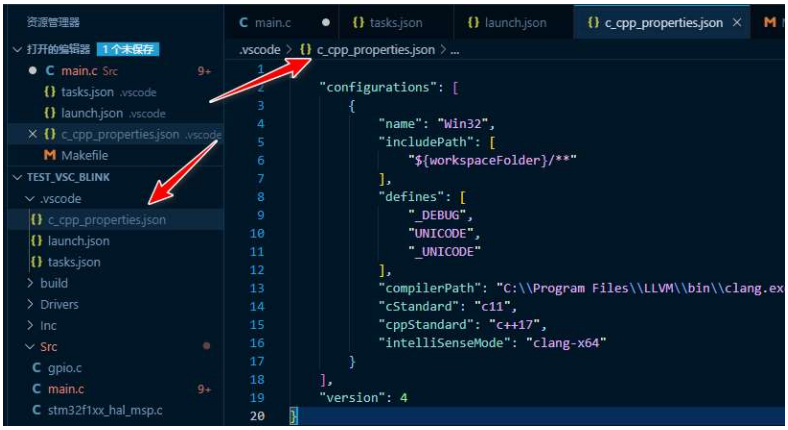

main.c
tasks.json
Makefile
.vscode > tasks.json > [ ] tasks
// 有关 tasks.json 格式的文档，请参见
// https://go.microsoft.com/fwlink/?LinkId=733558
"version": "2.0.0",
"tasks": [
{
"type": "shell",
"label": "clang.exe build active file",
"command": "C:\\Program Files\\LLVM\\bin\\clang.exe",
"args": [
"-g",
"${file}",
"-o",
"${fileDirname}\\${fileBasenameNoExtension}.exe"
],
"options": {
"cwd": "C:\\Program Files\\LLVM\\bin"
},
"problemMatcher": [
"$gcc"
],
"group": "build"
}
]
}


>debug
调试: 打开 launch.json
Debug: Open launch.json
调试: 禁用所有断点
Debug: Disable All Breakpoints


选择环境
C++ (GDB/LLDB)
C++ (Windows)
Node.js
PlatformIO Debugger
PowerShell
More...


main.c
tasks.json
launch.json
Makefile
.vscode > launch.json > ...
"cwd": "${workspaceFolder}",
"environment": [],
"externalConsole": false,
"MIMode": "gdb",
"miDebuggerPath": "/path/to/gdb",
"setupCommands": [
{
"description": "Enable pretty-printing for gdb",
"text": "-enable-pretty-printing",
"ignoreFailures": true
}
]
}
]
}


>c c++
C/C++: 选择配置...
最近使用
C/C++: Select a Configuration...
C/C++: 编辑配置 (UI)
C/C++: Edit Configurations (UI)
C/C++: 编辑配置 (JSON)
C/C++: Edit Configurations (JSON)
合并冲突: 比较当前冲突
其他命令
Merge Conflict: Compare Current Conflict
C/C++: 导航...
Alt + N
C/C++: Navigate...
C/C++: 调查问卷
C/C++: Take Survey
C/C++: 访问vcpkg帮助页面
C/C++: Visit the vcpkg help page
C/C++: 更换配置提供程序...



六、 连接测试

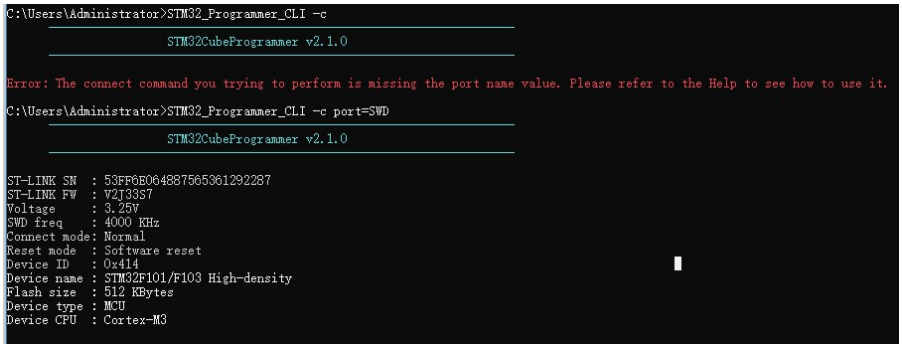
6.1 在终端使用 stm32cubeprogrammer 连接 stm32

主要是使用它的 *STM32_Programmer_CLI.exe* 这个命令行工具。当然，安装 jlink、st-link utility 等也可以。

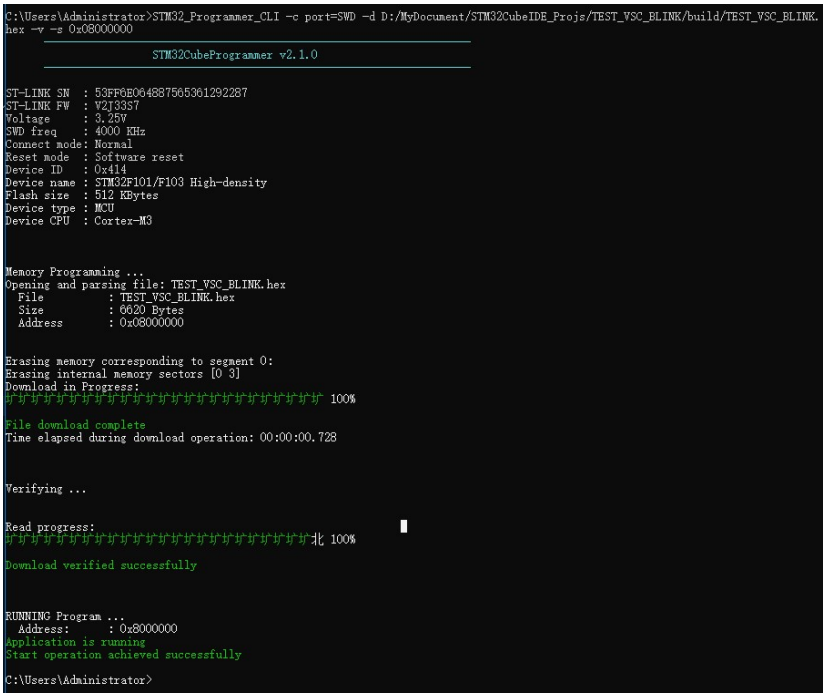
下图为 st-link utility 和 STM32cubeProgrammer 的 CLI 工具帮助信息，相对而言，Programmer 的功能更丰富，所以以这个为例来演示。

6.2 Stlink 连接 stm32 芯片测试：

终端输入命令：`STM32_Programmer_CLI -c port=SWD`，结果如图所示，可以显示连接到的芯片的信息，说明已正确连接。前提是硬件正常、正确连接、供电正常。



使用终端成功下载程序



使用 vsc 修改 makefile 后再终端下载程序

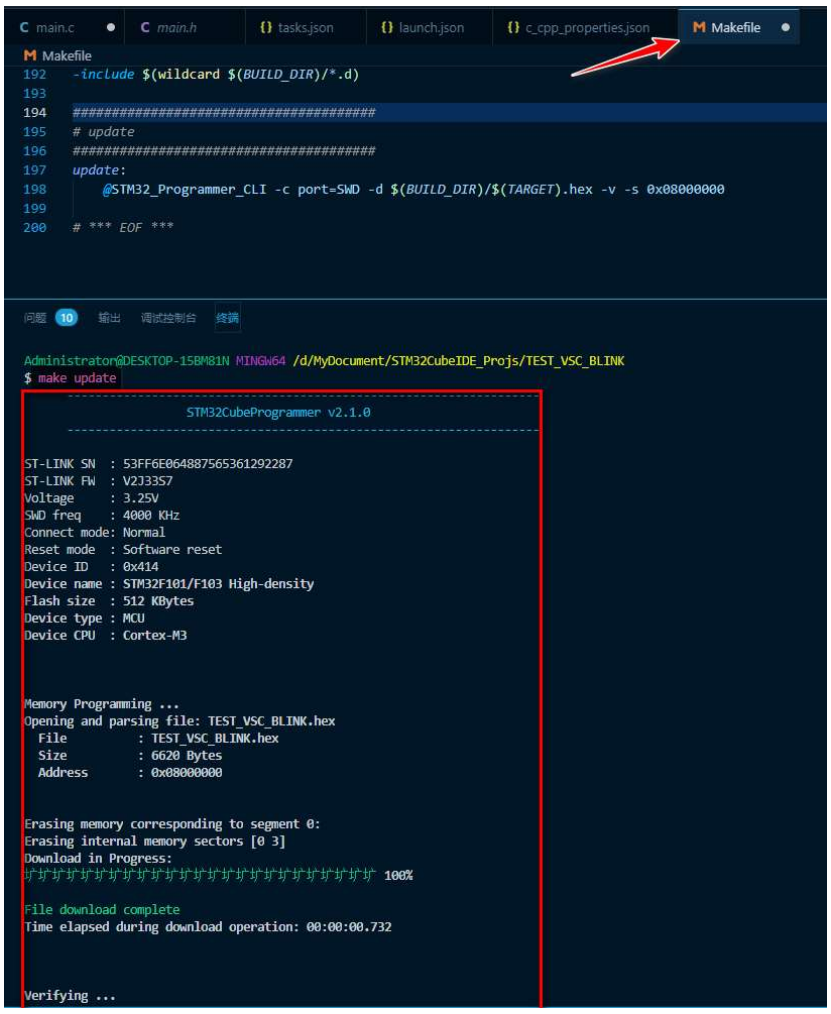
6.3 Makefile 修改

Makefile 添加代码如下：

```
update:
```

```
	@STM32_Programmer_CLI -c port=SWD -d $(BUILD_DIR)/$(TARGET).hex
-v -s 0x08000000
```

添加 update 段，功能是-c 连接 device，port=SWD，使用 swd 接口，-d 下载，后边为下载的文件，-v Verify，-s start，后边是下载首地址。



七、 Vscode 下 json 文件配置

7.1 c_cpp_properties.json 文件

主要有添加 include 路径，编译器路径，宏定义等

```
{
    "configurations": [
        {
            "name": "Win32",
            "includePath": [
                "${workspaceFolder}Drivers/STM32F1xx_HAL_Driver/Inc",
```

```
                "${workspaceFolder}Drivers/STM32F1xx_HAL_Driver/Inc/Leg
acy",
                "${workspaceFolder}Drivers/CMSIS/Device/ST/STM32F1xx/In
clude",
                "${workspaceFolder}Drivers/CMSIS/Include",
                "${workspaceFolder}Drivers/CMSIS/Include",
                "C:/Program Files (x86)/GNU Tools ARM Embedded/8 2019-
q3-update/arm-none-eabi/include"
            ],
            "defines": [
                "USE_HAL_DRIVER ",
                "STM32F103xE ",
                "USE_HAL_DRIVER ",
                "STM32F103xE"
            ],
            "compilerPath": "C:\\Program Files (x86)\\GNU Tools ARM Emb
edded\\8 2019-q3-update\\bin\\arm-none-eabi-gcc.exe",
            "cStandard": "c11",
            "cppStandard": "c++17",
            "intelliSenseMode": "gcc-x86"
        }
    ],
    "version": 4
    }
```

7.2 launch.json

```
{
    // 使用 IntelliSense 了解相关属性。
    // 悬停以查看现有属性的描述。
    // 欲了解更多信息，请访
问: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "stm32 Debug",
            "cwd": "${workspaceRoot}",
            "executable": "${workspaceFolder}/build/TEST_VSC_BLINK.elf"
,
            "request": "launch",
            "type": "cortex-debug",
            "servertype": "stutil",
            "device": "STM32F103RE",
            "preLaunchTask": "生成并下载",
            "postDebugTask": "复位设备"
```

```
        }
    ]
    }
```

7.3 tasks.json

```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "label": "生成",
            "type": "shell",
            "command": "make -j6",
            "problemMatcher": [],
            "group": {
                "kind": "build",
                "isDefault": true
            }
        },
        {
            "label": "生成并下载",
            "type": "shell",
            "command": "make -j6 && make update",
            "problemMatcher": []
        },
        {
            "label": "重新生成",
            "type": "shell",
            "command": "make clean && make -j6",
            "problemMatcher": []
        },
        {
            "label": "复位设备",
            "type": "shell",
            "command": "STM32_Programmer_CLI -c port=SWD -hardRst",
            "problemMatcher": []
        }
    ]
    }
```

附件（命令行下载程序代码）：

命令行下使用各种软件下载方法

```
1.    STM32_Programmer _CLI -c port=SWD -d $(BUILD_DIR)/$(TARGET).hex -
   v -s 0x08000000
2.  st-flash write TEST_VSC_BLINK.bin 0x8000000
3. ST-LINK_CLI  -c  SWD  -
   P  E:/wo4fisher/Documents/keil_proj__stm32/GCC_VSCODE_TEST/build/GCC
   _VSCODE_TEST.hex  -V  "after_programming"
```

第一种：STM32_Programmer，第二种 git-hub 开源软件 st-link（release 地址：

https://github.com/texane/stlink/releases/tag/1.3.0），第三种：st link utility

Launch 原版

```
{
    // 使用 IntelliSense 了解相关属性。
    // 悬停以查看现有属性的描述。
    // 欲了解更多信息，请访
问: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "stm32 Debug",
            "cwd": "${workspaceRoot}",
            "executable": "${workspaceFolder}/build/TEST_VSC_BLINK.elf"
,
            "request": "launch",
            "type": "cortex-debug",
            "servertype": "stutil",
            "device": "STM32F103RE",
            "preLaunchTask": "生成并下载",
            "postDebugTask": "复位设备"
        }
    ]
    }
```