

AT32 MCU Develop with VSCode

前言

本使用指南描述了使用VSCode开发、编译、下载、调试AT32 MCU的解决方法，支持AT32F全系列芯片。

支持型号列表：

支持型号	AT32 全系列
------	----------

目录

1	概述	5
2	软件安装	6
2.1	VSCode 安装.....	6
2.2	交叉编译工具链与 MinGW 安装.....	6
2.2.1	交叉编译工具链安装.....	6
2.2.2	MinGW 安装	7
2.3	Openocd 使用.....	7
2.4	环境变量配置.....	7
3	Template 使用	9
3.1	硬件环境.....	9
3.2	工程导入.....	9
3.3	配置修改.....	10
3.4	工程使用.....	11
4	IC 型号修改	13
4.1	Makefile 文件修改.....	13
4.2	c_cpp_properties.json 脚本修改	13
5	版本历史	14

表目录

表 4. 文档版本历史 14

图目录

图 1. VSCode 官方下载.....	6
图 2. VSCode 界面.....	6
图 3. MinGW 设置.....	7
图 4. 环境变量设置.....	8
图 5. 模板工程.....	9
图 6. 工程目录结构.....	9
图 7. c_cpp_properties.json 脚本.....	10
图 8. launch.json 脚本.....	10
图 9. tasks.json 脚本.....	11
图 10. 编译、下载和清除.....	11
图 11. 调试界面.....	12
图 12. Makefile 修改.....	13
图 13. c_cpp_propertie 修改.....	13

1 概述

VSCode（全称：Visual Studio Code）是一款由微软开发且跨平台的免费源代码编辑器。该软件支持语法高亮、代码自动补全（又称IntelliSense）、代码重构、查看定义功能，并且内置命令行工具和Git版本控制系统。用户可以更改主题和键盘快捷方式实现个性化设置，也可以通过内置的插件安装功能扩展软件功能。在2019年的Stack Overflow组织的开发者调查中，Visual Studio Code被认为是最受开发者欢迎的开发环境。

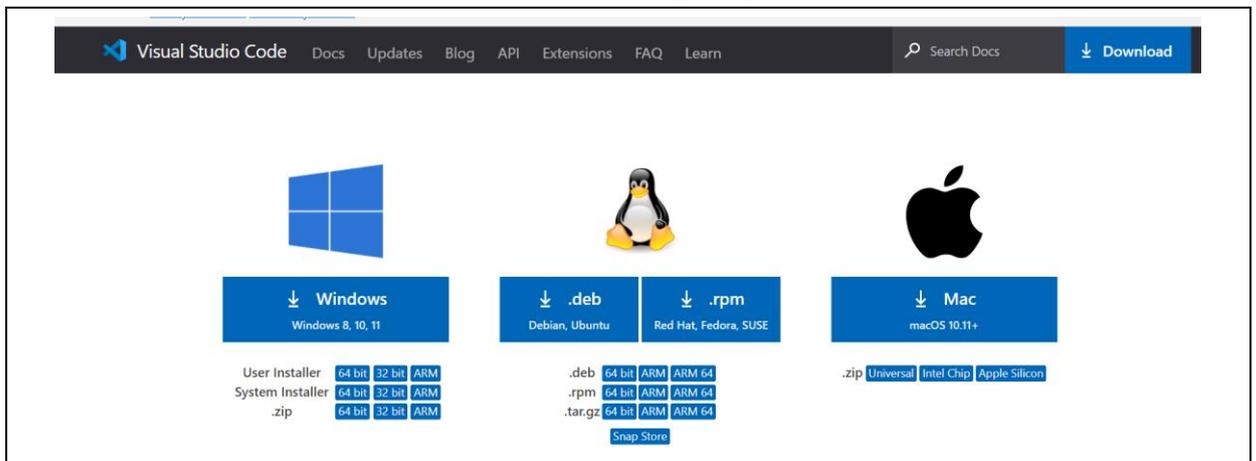
VSCode因为其轻量化、丰富的插件以及炫酷的界面等优点，当前成为了开发者热捧的开发工具。AT32提供在VSCode下开发、编译、下载、调试解决方法。只需安装所需要的软件工具并简单配置即可使用。本指南所使用的所有软件，目前都是开源的。

2 软件安装

2.1 VSCode 安装

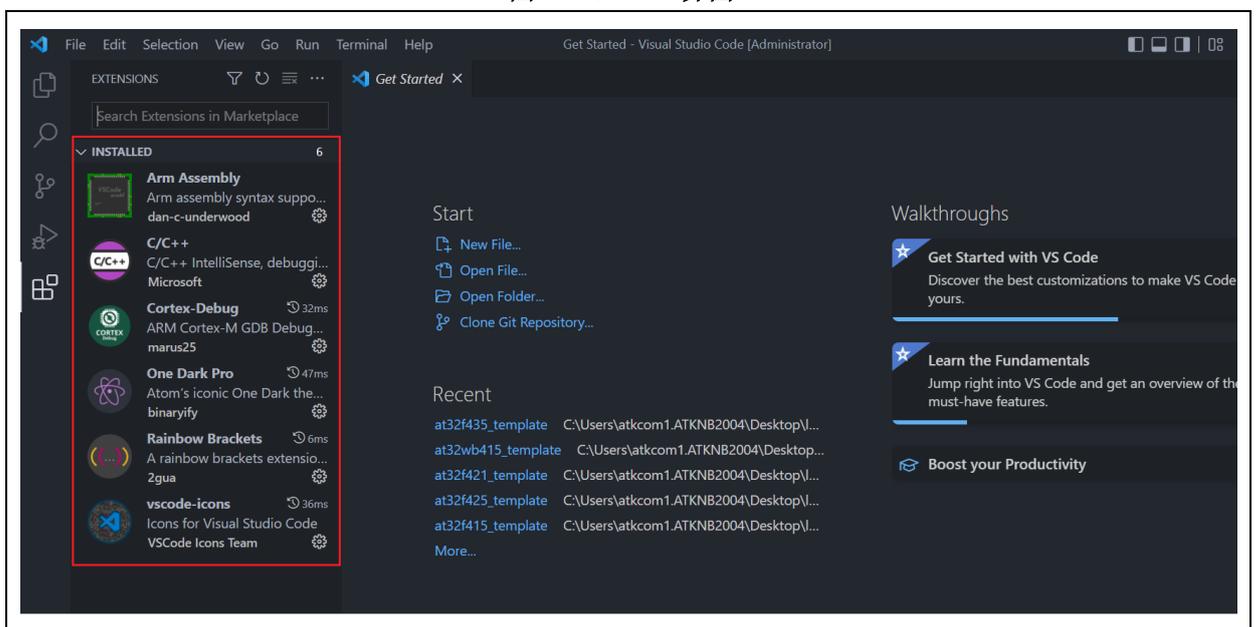
VSCode 是微软开发的代码编辑器软件，可免费下载和使用，登录 VSCode 官网即可下载安装文件，注意在 windows 环境下需要下载对应的版本。

图 1. VSCode 官方下载



下载安装后打开 VSCode 界面如下图所示

图 2. VSCode 界面



注意安装完成后需要将 VSCode 安装路径加入到环境变量中，后续使用会更加方便。上图中的红色框内为安装的插件，C/C++和 Cortex-debug 必须安装，其他可选择性安装即可。

2.2 交叉编译工具链与 MinGW 安装

本小节说明如何安装 arm 交叉编译工具链和 MinGW，以及 openocd 的使用。

2.2.1 交叉编译工具链安装

为了能使在 windows 平台下编译的代码在 AT32 MCU 上运行，需要安装 ARM 提供的交叉编译工具链。

在本文档配套的代码包内可找到此软件(gcc-arm-none-eabi-10.3-2021.10-win32.exe)，然后双击运行安装即可。

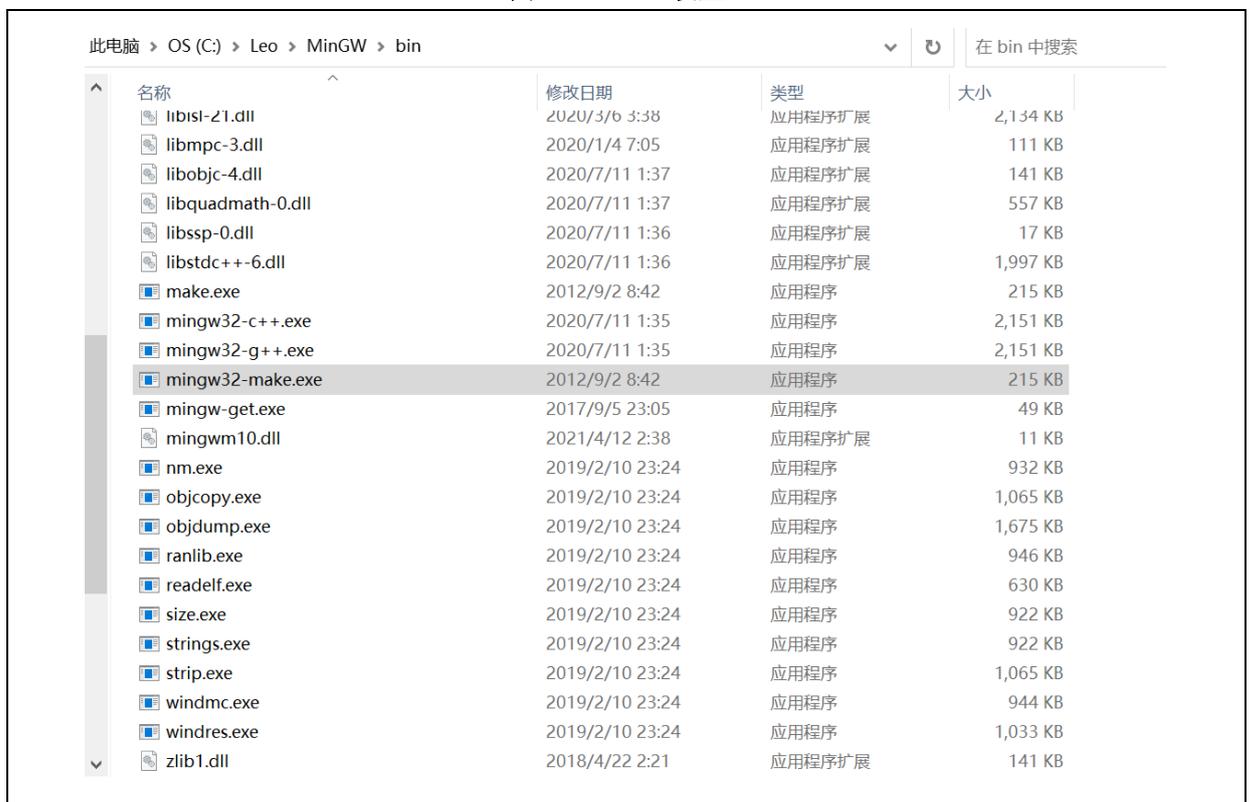
注意安装完成后需要添加系统环境变量。

2.2.2 MinGW 安装

MinGW是Minimalist GNU for windows的缩写。它是可以自由使用自由发布的windows特定头文件和使用GNU工具集导入库的集合，可在windows平台生成本地的windows程序。其包括GCC编译器，所以本应用所使用的编译器为GCC。

在本文档配套的代码包内可找到此软件(mingw-get-setup.exe)，双击运行即可安装。需要注意安装完成后需要添加系统环境变量。

图 3. MinGW 设置



到MinGW安装目录下，进入/bin目录，找到其中的mingw32-make.exe并复制一份放到相同目录下，修改命名为make.exe。

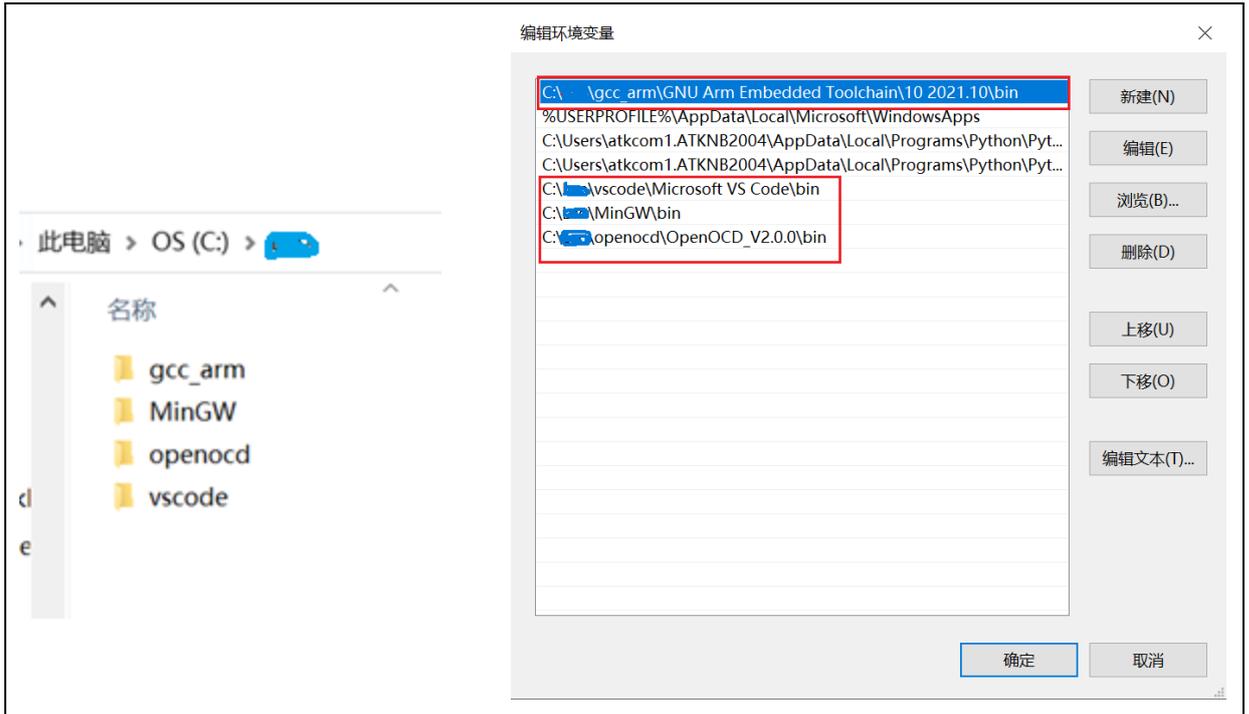
2.3 Openocd 使用

Openocd可执行文件由雅特力提供，在本文档配套的代码包内可找到。使用时只需将其添加到系统环境变量即可，无需其他安装与配置。

2.4 环境变量配置

在系统中搜索“环境变量”，进入环境变量配置界面，确保前面所安装的软件都已添加到环境变量中；例如如下所示：

图 4. 环境变量设置



上图为笔者系统下软件安装路径和环境变量配置，可作为参考。

需要特别注意，如若环境变量安装路径不对，会导致VSCode编译下载代码等出错。

3 Template 使用

3.1 硬件环境

对应AT-START-BOARD搭配AT-Link

3.2 工程导入

代码包内模板工程如下：

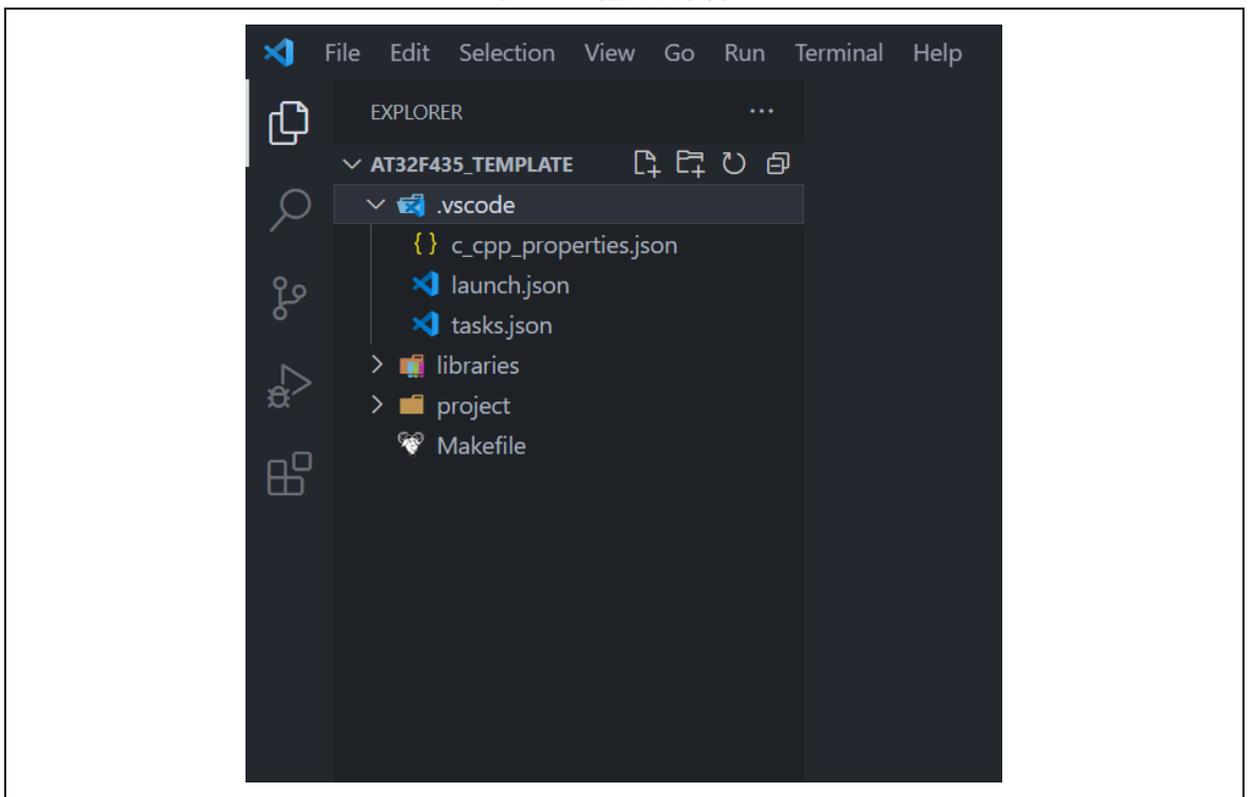
图 5. 模板工程

名称	修改日期	类型
at32f403_template	2022/5/27 13:31	文件夹
at32f403a_template	2022/5/27 13:31	文件夹
at32f407_template	2022/5/27 13:31	文件夹
at32f413_template	2022/5/27 13:31	文件夹
at32f415_template	2022/5/27 13:31	文件夹
at32f421_template	2022/5/27 13:31	文件夹
at32f425_template	2022/5/27 13:31	文件夹
at32f435_template	2022/5/27 13:31	文件夹
at32f437_template	2022/5/27 13:31	文件夹
at32wb415_template	2022/5/27 13:32	文件夹

如若环境安装正确，下面只需要将本文档配套的代码包内的模板工程文件夹使用VSCode打开即可。

打开后工程目录如下所示：

图 6. 工程目录结构



Libraries: 主要存放artery芯片库文件;

Project: 工程相关配置文件以及用户代码;

Makefile: 编译器需要使用;

.vscode: 工程配置脚本。

3.3 配置修改

由于环境不同，软件安装的路径存在差异，所以需要进行环境适配。只需修改.vscode目录下的3个配置脚本即可。

1. c_cpp_properties.json脚本修改

图 7. c_cpp_properties.json 脚本

```
1 {
2   "configurations": [
3     {
4       "name": "Win32",
5       "includePath": [
6         "C:/Program Files/GNU Arm Embedded Toolchain/10.2021.10/arm-none-eabi/include",
7         "${workspaceFolder}/project/inc",
8         "${workspaceFolder}/project/board",
9         "${workspaceFolder}/libraries/cmsis/cm4/core_support",
10        "${workspaceFolder}/libraries/cmsis/cm4/device_support",
11        "${workspaceFolder}/libraries/drivers/inc",
12        "${workspaceFolder}",
13        "${workspaceFolder}/libraries/cmsis/cm4/device_support"
14      ],
15      "defines": [
16        "_DEBUG",
17        "UNICODE",
18        "_UNICODE",
19        "AT_START_F435_V1",
20        "AT32F435ZM7",
21        "USE_STDPERIPH_DRIVER"
22      ],
23      "compilerPath": "C:/Program Files/GNU Arm Embedded Toolchain/10.2021.10/bin/arm-none-eabi-gcc.exe",
24      "cStandard": "gnu17",
25      "cppStandard": "gnu++14",
26      "intelliSenseMode": "gcc-x64"
27    }
28  ],
29  "version": 4
30 }
```

上图中框起来的路径需要根据实际路径进行修改。

2. launch.json脚本修改

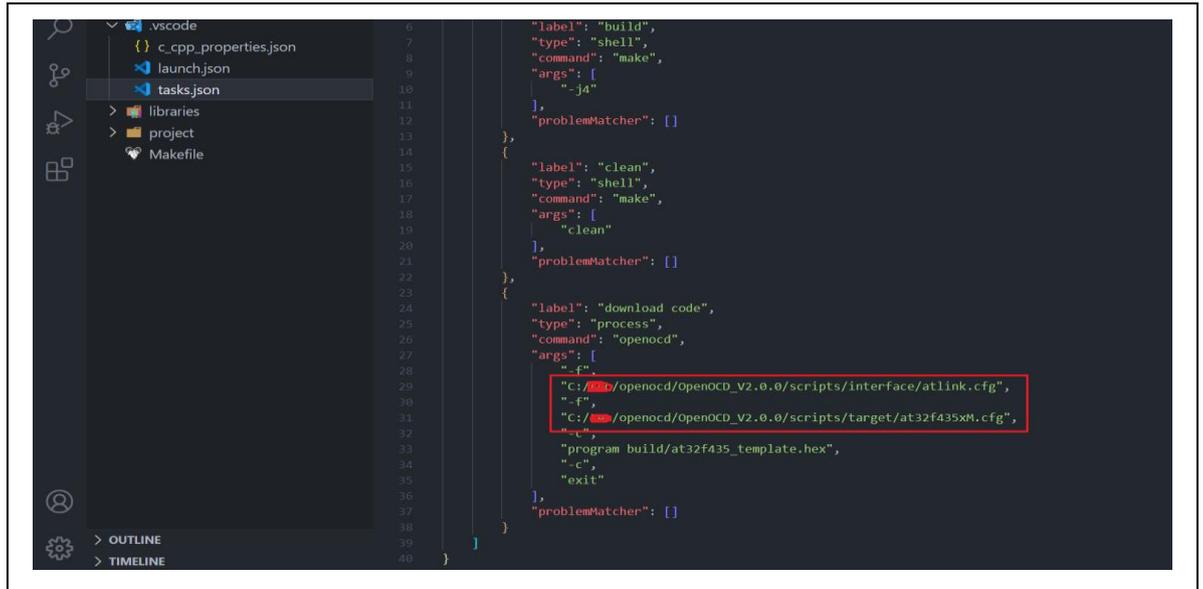
图 8. launch.json 脚本

```
1 {
2   // "version": "0.2.0",
3   "configurations": [
4     {
5       "cwd": "${workspaceRoot}",
6       "type": "cortex-debug",
7       "request": "launch",
8       "name": "openocd-launch",
9       "armToolchainPath": "C:/Program Files/GNU Arm Embedded Toolchain/10.2021.10/bin",
10      "gdbPath": "C:/Program Files/GNU Arm Embedded Toolchain/10.2021.10/bin/arm-none-eabi-gdb.exe",
11      "interface": "swd",
12      "serverType": "openocd",
13      "configFiles": [
14        "C:/Program Files/OpenOCD/OpenOCD_V2.0.0/scripts/interface/atlink.cfg",
15        "C:/Program Files/OpenOCD/OpenOCD_V2.0.0/scripts/target/at32f435xm.cfg"
16      ],
17      "executable": "build/at32f435_template.elf",
18      "svdFile": "./project/misc/AT32F435xx_v2.svd",
19      "runToMain": true,
20    }
21  ],
22  "version": 4
23 }
```

上图中框起来的路径需要根据实际路径进行修改。

3. tasks.json脚本修改

图 9. tasks.json 脚本



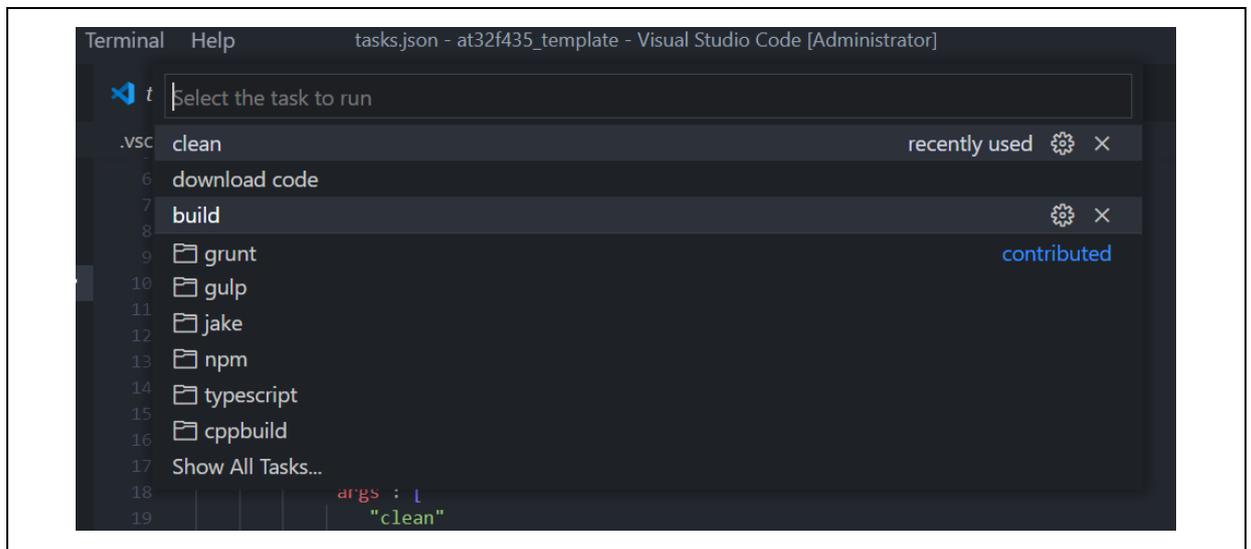
上图中框起来的路径需要根据实际路径进行修改。

3.4 工程使用

通过以上配置，现在可以使用VSCode开发、编译、下载、调试AT32 MCU了。

点击VSCode菜单栏Terminal→Run Task会出现如下界面：

图 10. 编译、下载和清除



界面有clean、download code和build，点击这三个按钮分别执行如下操作：

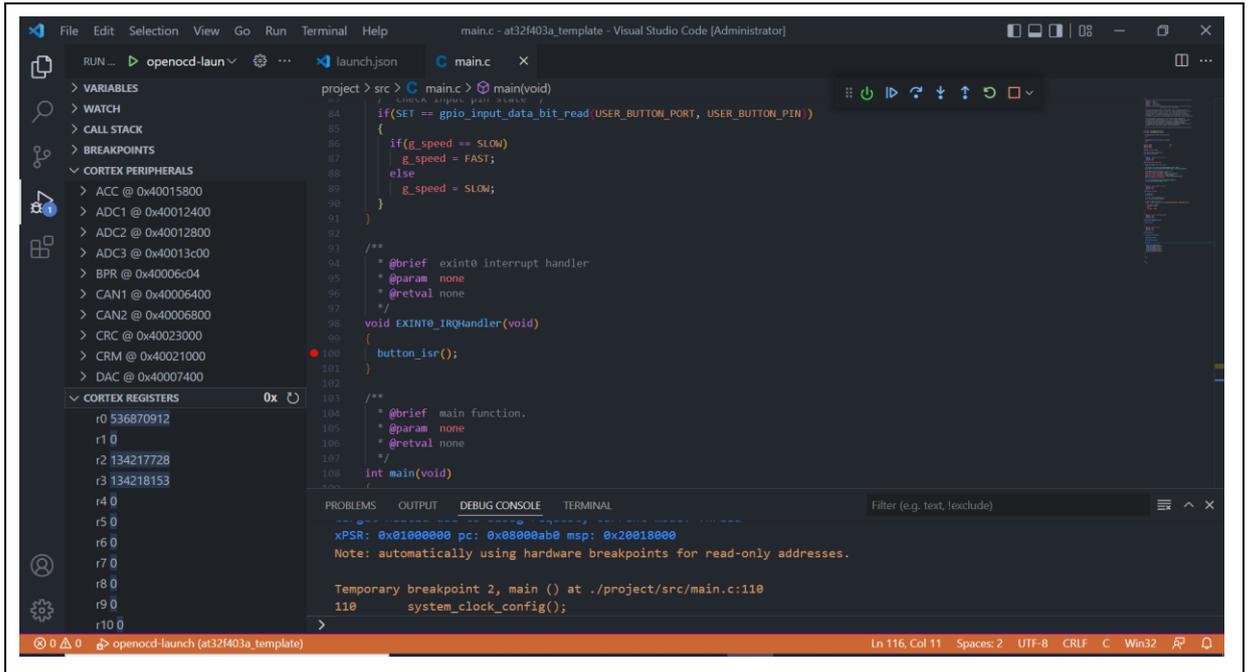
Build: 编译代码；

Clean: 删除编译产生的文件；

download code: 下载代码到目标板。

编译下载代码后，点击VSCode菜单栏Run→Start Debugging即可进入调试状态：

图 11. 调试界面



进入调试界面后可进行单步执行等操作，也可查看外设寄存器值、变量值等操作。

4 IC 型号修改

Template工程为各系列中单一型号，如果使用到其他型号需要按照如下步骤修改即可。

4.1 Makefile 文件修改

Makefile文件如下红框内容需根据实际型号进行修改

图 12. Makefile 修改

```
111 # C defines
112 C_DEFS = \
113 -DAT_START_F435_V1 \
114 -DAT32F435ZMT7 \
115 -DUSE_STDPERIPH_DRIVER
...
143 #####
144 # LDFLAGS
145 #####
146 # link script
147 LDSOSCRIPT = ./project/misc/AT32F435xM_FLASH.ld
```

.ld文件可在artery BSP内获取，路径如下：

AT32F435_437_Firmware_Library_V2.x.x\libraries\cmsis\cm4\device_support\startup\gcc\linker

理论上修改完makefile文件就可以顺利编译下载调试代码了，但在VSCode上阅读代码时会出现波浪线等警告提示，这是因为VSCode解释代码关系出错，还需要修改一下c_cpp_properties.json脚本

4.2 c_cpp_properties.json 脚本修改

文件如下红框内容需根据实际型号进行修改：

图 13. c_cpp_propertie 修改

```
15  ✓
16
17
18
19
20
21
22
"defines": [
  "_DEBUG",
  "UNICODE",
  "_UNICODE",
  "AT_START_F435_V1",
  "AT32F435ZMT7",
  "USE_STDPERIPH_DRIVER"
],
```

5 版本历史

表 1. 文档版本历史

日期	版本	变更
2022.5.12	2.0.0	最初版本
2022.5.27	2.0.1	1.OpenOCD进版到V2.0.2 2.template增加at32wb415支持

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 (重庆) 有限公司 保留所有权利