

## 摘要

由于某些客户受到 Keil/IAR 的使用限制，可能需要提供 GNU ARM GCC 开发调试支持，开发人员需要自己搭建 GCC 编译开发环境，需要学习 MakeFile/CMake 语法、GCC 编译器命令行参数、GCC 链接脚本、JLink 配置等，而且在推广和移植时也会遇到比较多的问题。

因此我们开发了基于 VS Code 的插件，目的是统一开发工具、简化用户操作、提高开发效率，不需要学习编译/链接等脚本语法，只需要简单配置参数、新建工程、增加代码文件后，就可以开始编译和调试了。如果要生成静态库文件也只需要修改输出格式，再编译就可能生成.a 的文件。

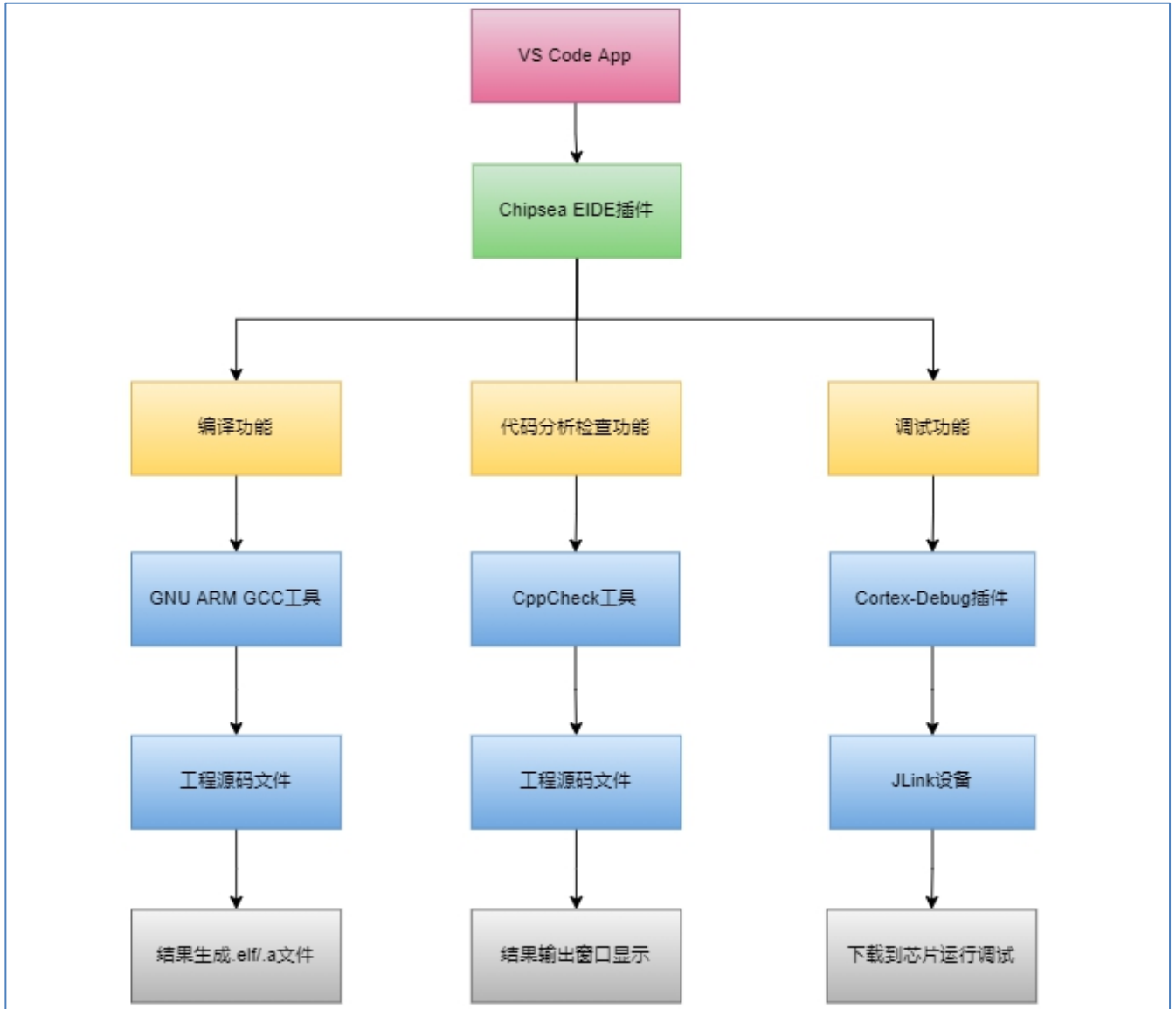
## 版本

历史版本	修改内容	日期
V1.0	初版生成	2022-06-01

一、插件介绍.....	11
1.1 插件开发架构.....	11
二、安装使用说明.....	12
2.1 安装 Chipsea IDE32 插件.....	12
2.2 安装 GNU ARM GCC（如本地已安装则跳过）.....	13
2.3 安装 Cortex-Debug 插件（如本地已经安装则跳过）.....	14
2.4 安装 Segger JLink 软件（如本地已经安装则跳过）.....	14
2.5 创建工程.....	15
2.5.1 通过内部模板创建工程.....	15
2.5.2 通过在线 Demo 模板创建工程.....	15
2.6 编译工程.....	16
2.6.1 设置编译工具路径.....	16
2.6.2 增加需要编译的源码目录和文件.....	16
2.6.3 排除不需要编译的文件或目录.....	17
2.6.4 编译工程源代码.....	17
2.6.5 编译生成库文件.....	18
2.7 运行调试.....	18
2.7.1 设置 Segger JLink 工具路径.....	18
2.7.2 运行调试.....	19
三、插件辅助功能.....	19
3.1 CppCheck 功能.....	19
3.2 代码注释功能.....	21

## 一、插件介绍

### 1.1 插件开发架构

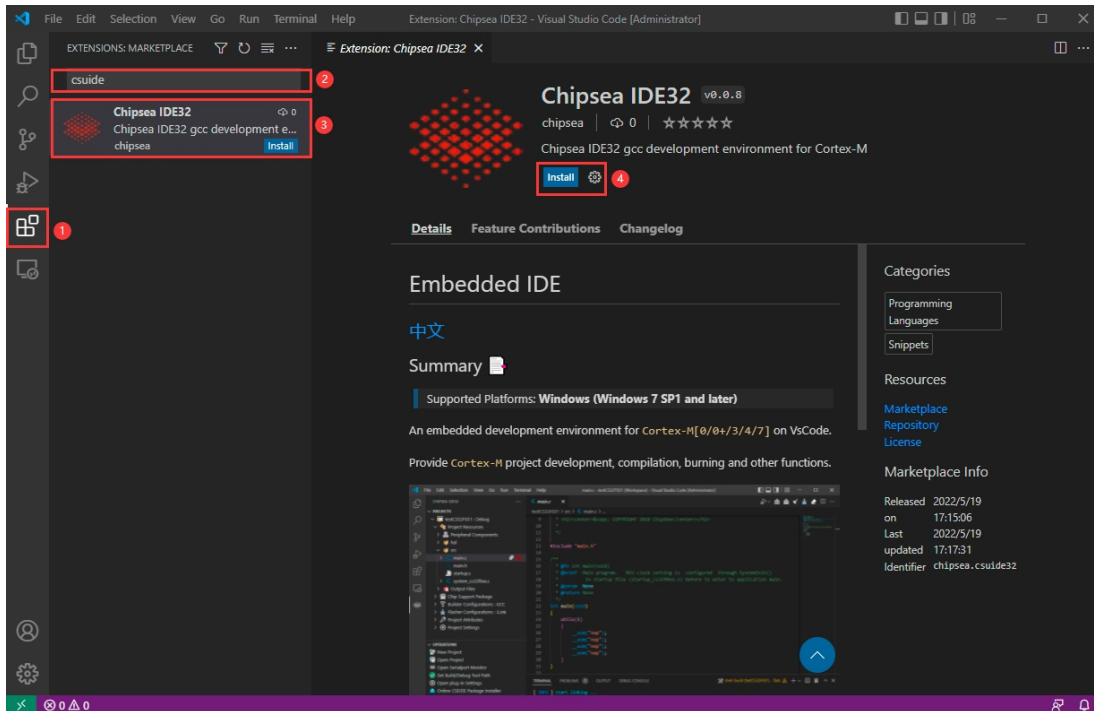


#### 主要功能特点:

- 1、基于开源插件 EIDE 进行二次开发
- 2、增加芯海的芯片型号
- 3、优化掉不必要的功能
- 4、增加在线型号包安装功能
- 5、增加在线 Demo 模板功能
- 6、增加代码注释和 CppCheck 功能
- 7、自动安装烧录算法到 JLink 并配置 Cortex-Debug 插件参数，简化用户操作

## 二、安装使用说明

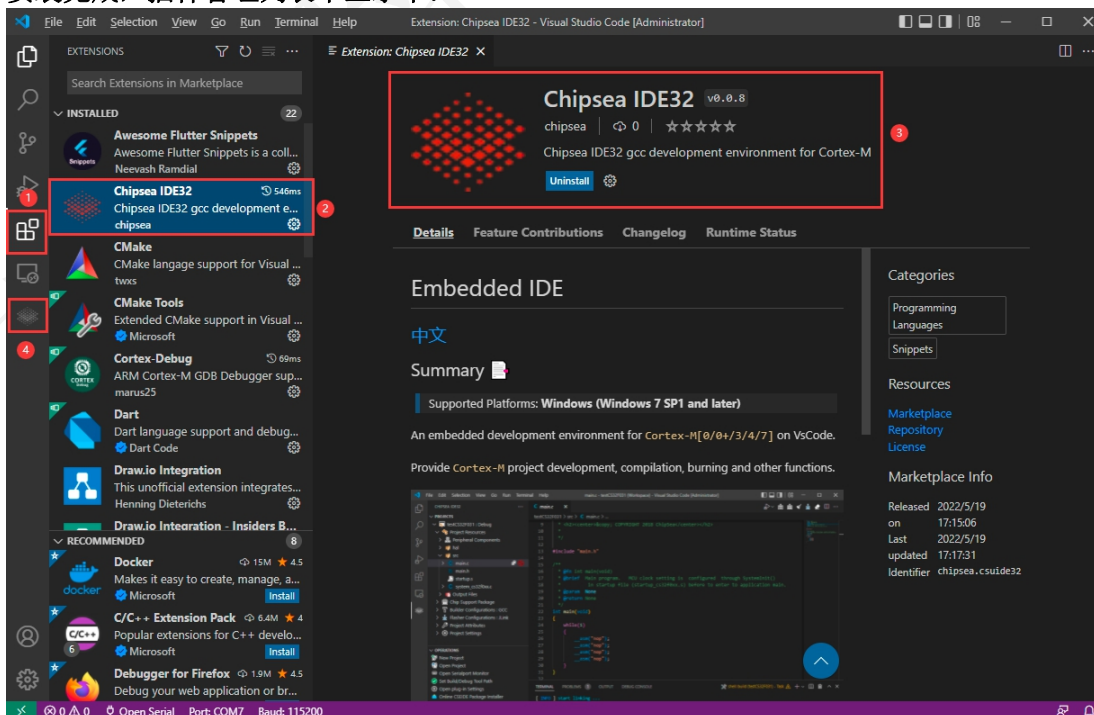
### 2.1 安装 Chipsea IDE32 插件



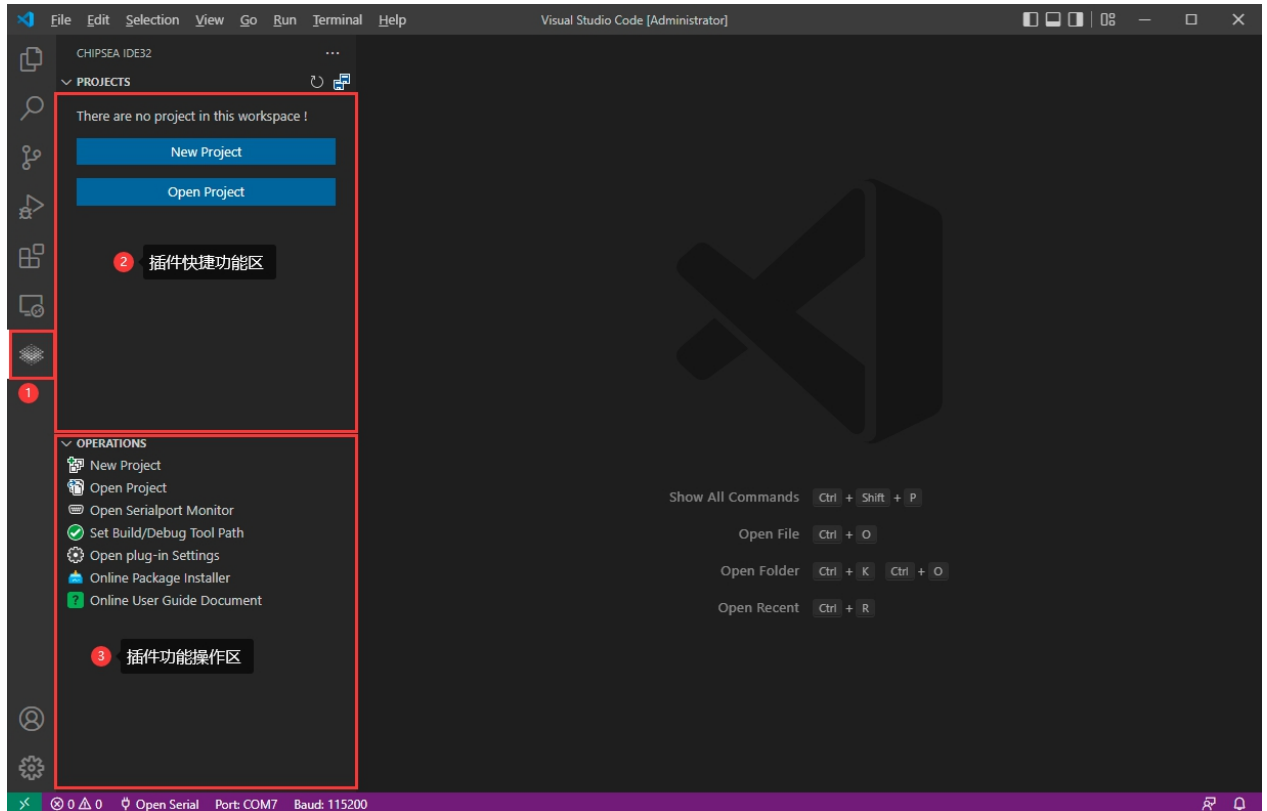
- 1、 点击插件管理器
- 2、 搜索栏输入 csuide
- 3、 点击 Chipsea IDE32 插件
- 4、 点击右侧插件的【Install】按钮进行安装

**注意：由于插件装依赖比较多，安装需要 1~2 分钟左右时间**

安装完成，插件管理列表中显示下：



安装完成，插件功能区显示如下：



## 2.2 安装 GNU ARM GCC (如本地已安装则跳过)

下载链接如下：

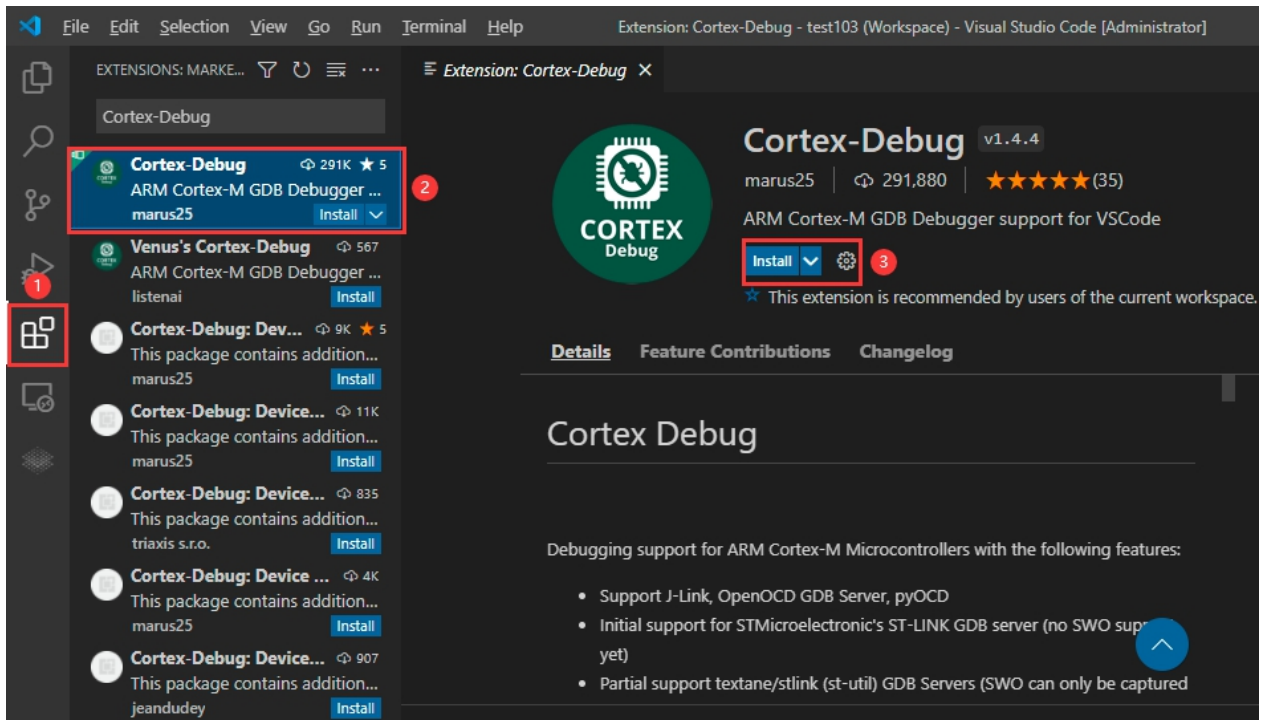
<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/downloads>

安装/解压后 **GNU ARM GCC 根目录**如下：

```
gcc-arm-none-eabi/  
|-- Tools  
|-- arm-none-eabi  
|   |-- bin  
|   |-- include  
|   |-- lib  
|   '-- share  
-- bin  
-- lib  
  '-- gcc  
-- share  
  |-- doc  
  '-- gcc-arm-none-eabi
```

根目录

### 2.3 安装 Cortex-Debug 插件（如本地已经安装则跳过）

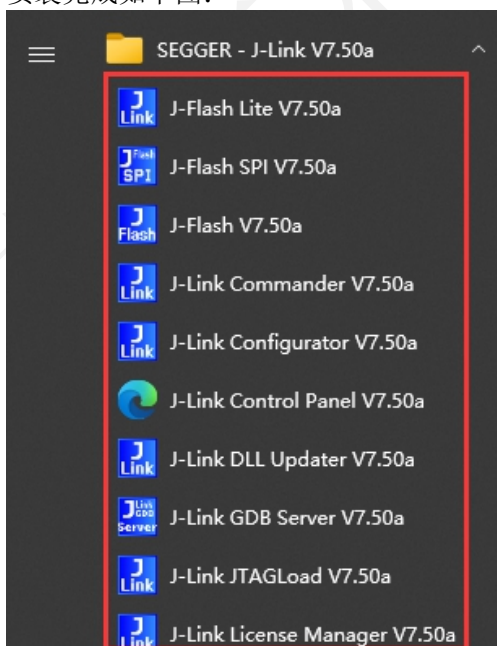


- 1、打开插件管理器
  - 2、搜索栏输入 Cortex-Debug 进行查找
  - 3、点击【Cortex-Debug】项后，点击【Install】按钮完成安装
- 注意：如果不需要调试功能，可以不用安装此插件**

### 2.4 安装 Segger JLink 软件（如本地已经安装则跳过）

下载链接：<https://www.segger.com/downloads/jlink/>

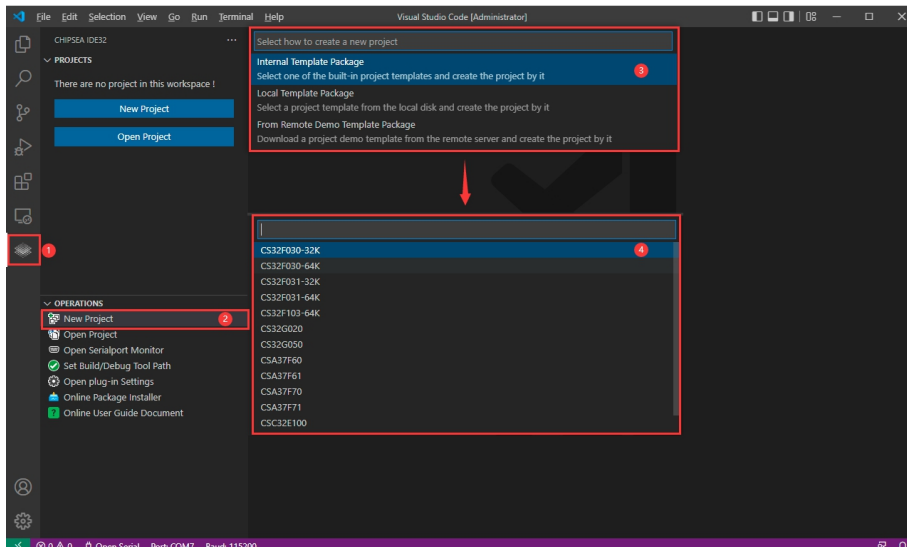
安装完成如下图：



**注意：如果不需要调试功能，可以不用安装此插件**

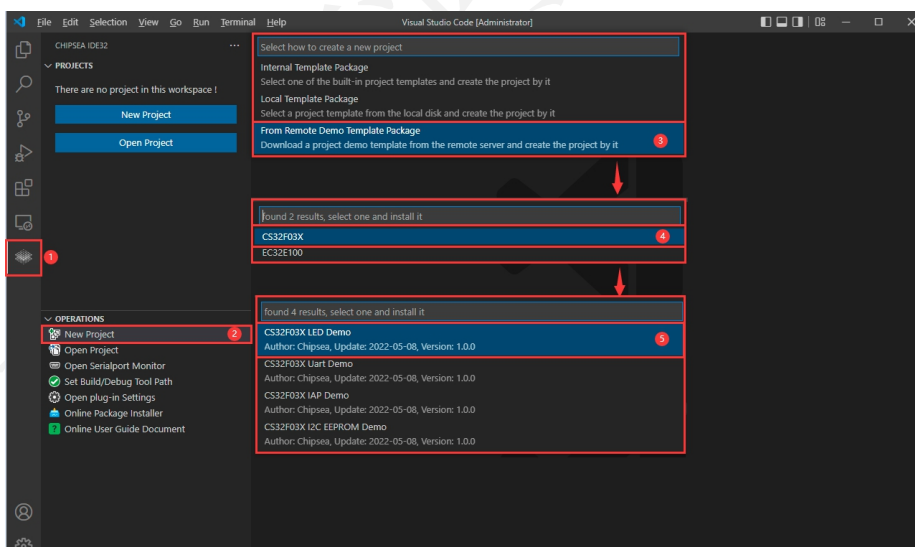
## 2.5 创建工程

### 2.5.1 通过内部模板创建工程



- 1、点击【New Project】菜单
- 2、点击【Internal Template Package】菜单
- 3、选择型号
- 4、输入工程名称
- 5、选择保存目录

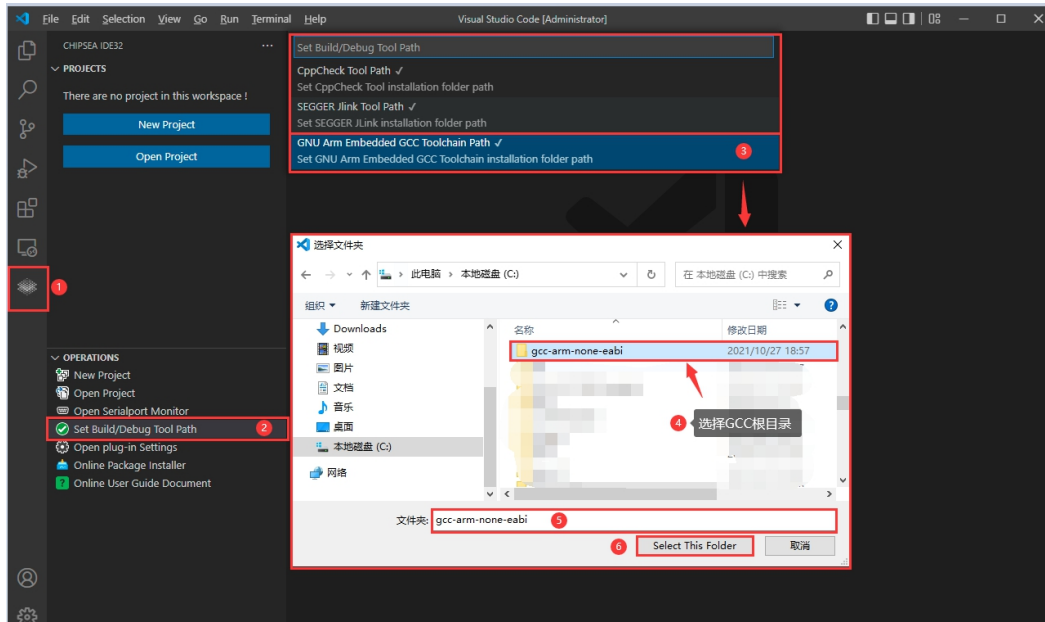
### 2.5.2 通过在线 Demo 模板创建工程



- 1、点击【New Project】菜单
- 2、点击【From Remote Demo Template Package】菜单
- 3、选择型号
- 4、选择 Demo 模板
- 5、输入工程名称
- 6、选择保存目录

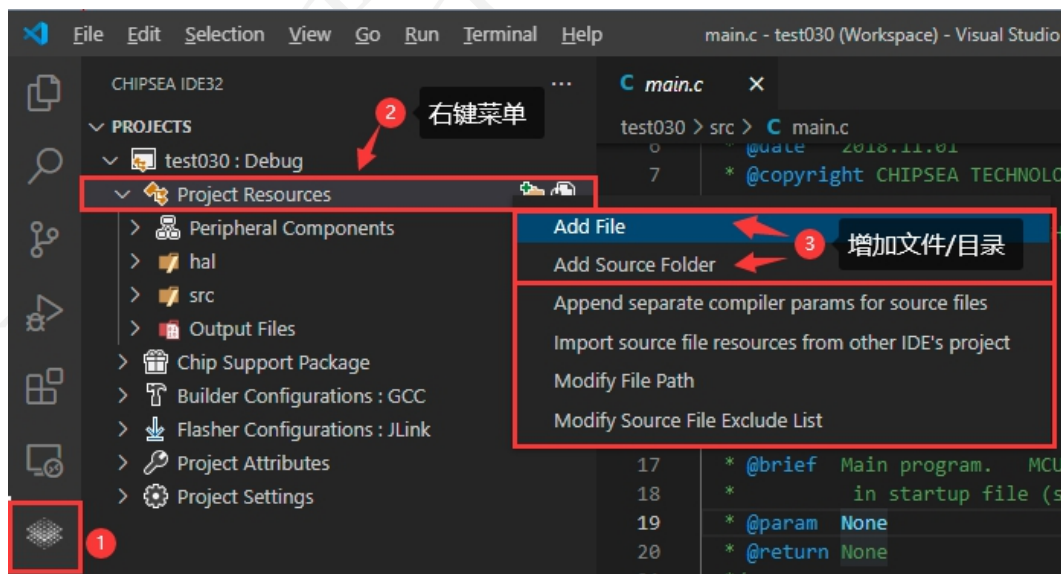
## 2.6 编译工程

### 2.6.1 设置编译工具路径



- 1、点击【Set Build/Debug Tool Path】菜单
- 2、点击【GNU Arm Embedded GCC Toolchain Path】菜单
- 3、选择 GNU ARM GCC 根目录
- 4、设置完成后，右下角会显示结果提示信息

### 2.6.2 增加需要编译的源码目录和文件

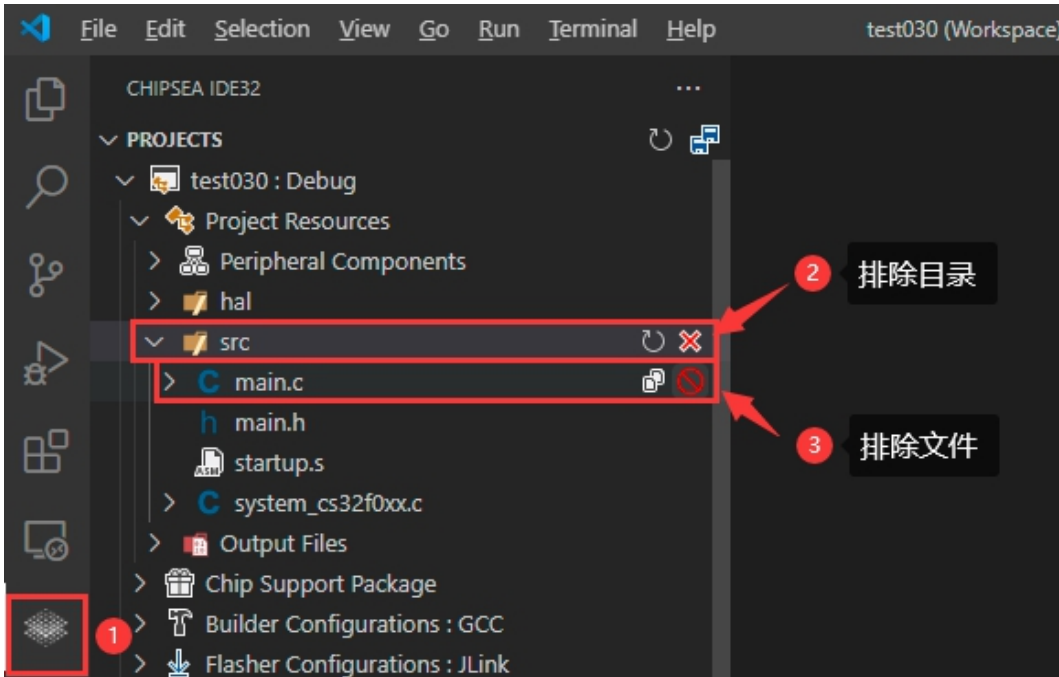


- 1、右击【Project Resources】结点，弹出菜单
- 2、选择【Add File】或【Add Source Folder】

注意：编译时会递归遍历查找目录下的所有.c/.cpp 文件进行编译，不需编译的文件需要设置为排除项

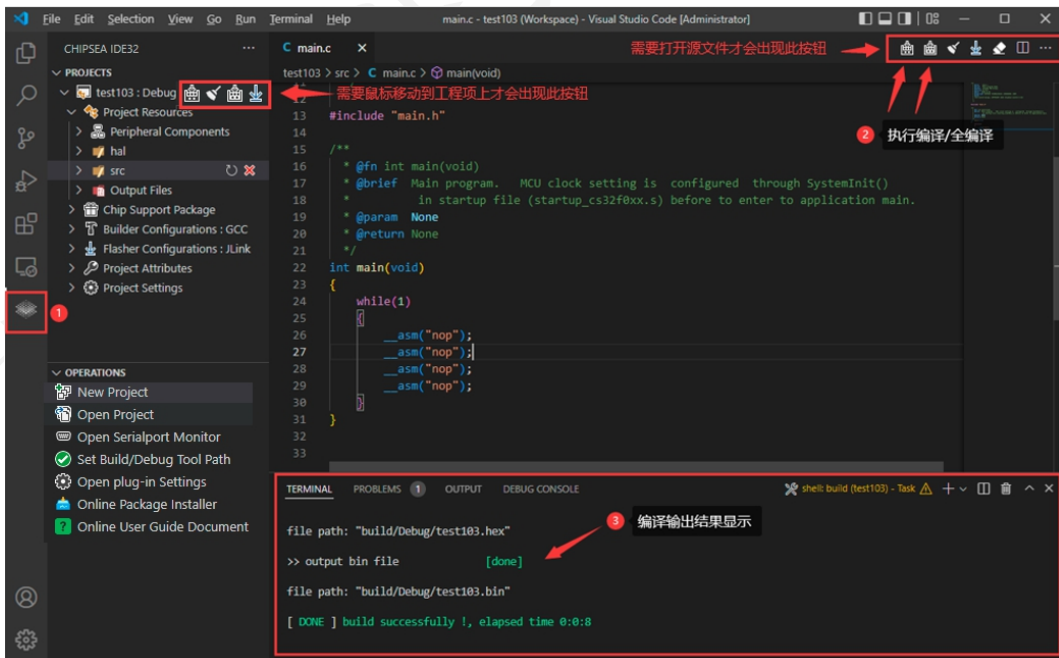


### 2.6.3 排除不需要编译的文件或目录



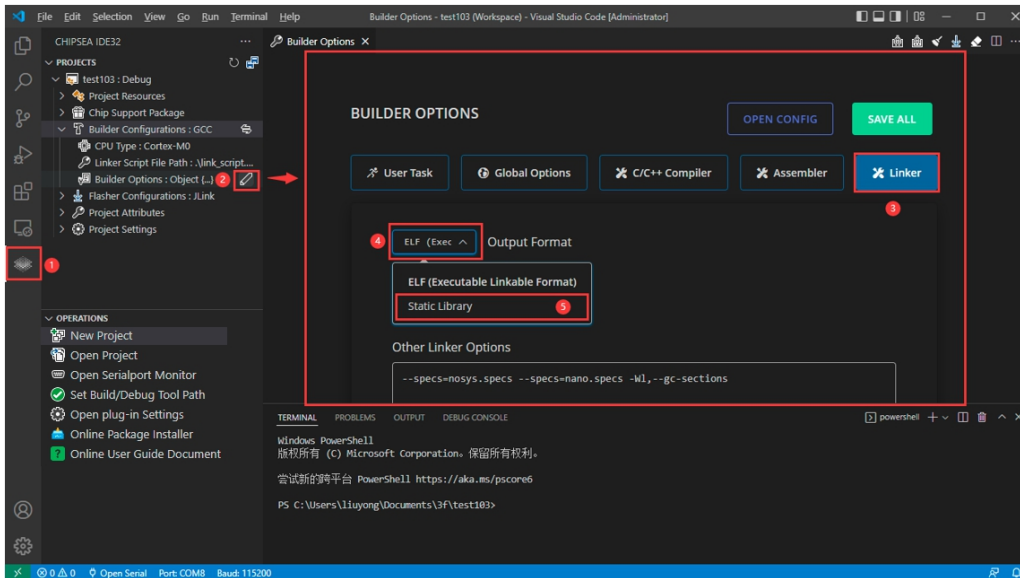
- 1、鼠标移动到目录上，点击 按钮，即可排除目录不进行编译
- 2、鼠标移动到文件上，点击 按钮，即可排除文件不进行编译

### 2.6.4 编译工程源代码



- 1、切换到插件功能面板
- 2、通过工程项或文件上方的工具条执行编译动作
- 3、编译结果显示在输出窗口中

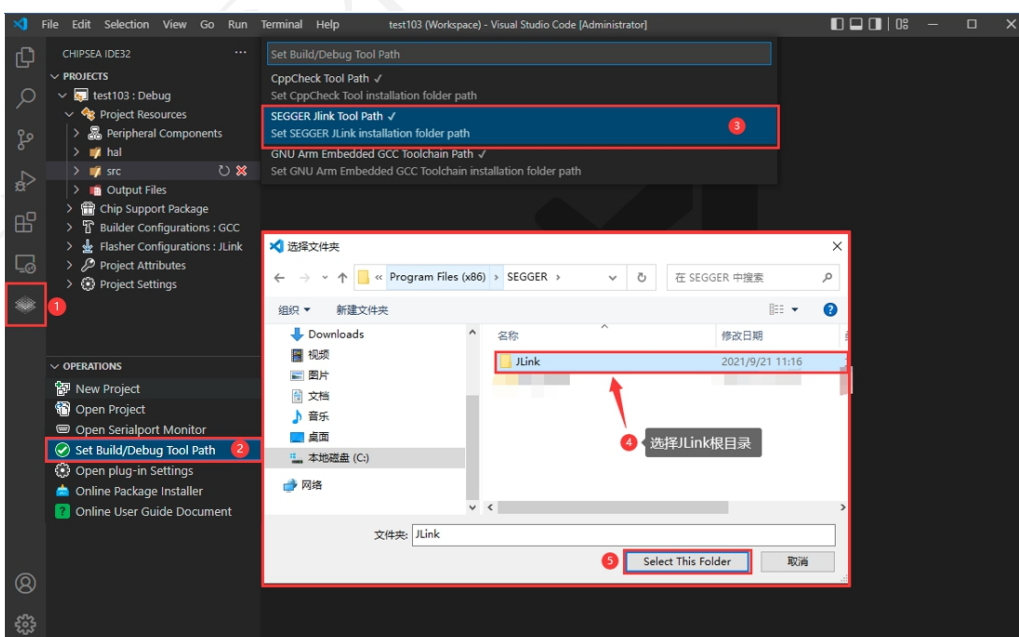
## 2.6.5 编译生成库文件



- 1、默认生成 elf 文件
- 2、切换到插件功能面板
- 3、点击【Builder Options】的编辑按钮
- 4、点击【Linker】按钮
- 5、点击【Output Format】按钮
- 6、选择【Static Library】按钮
- 7、执行编译动作，在输出目录生成.a 的文件

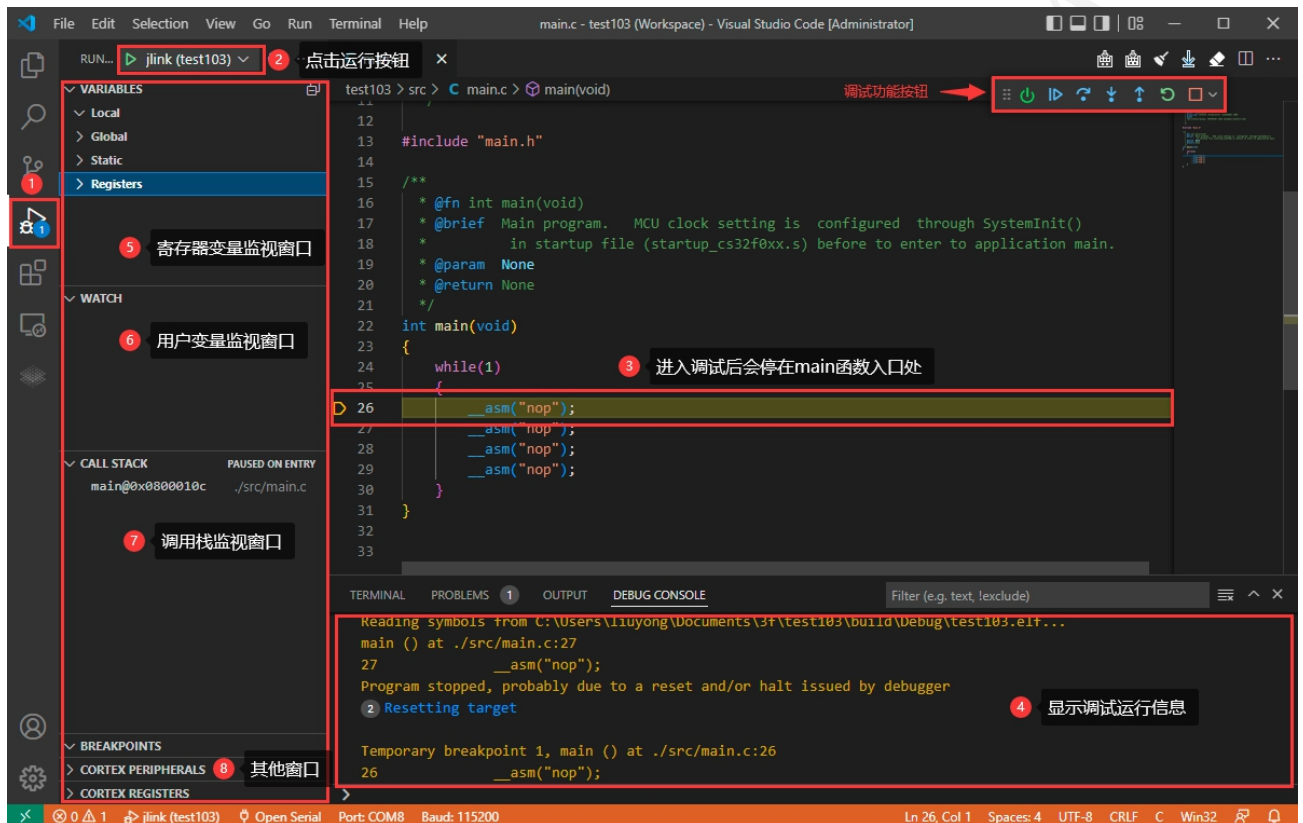
## 2.7 运行调试

### 2.7.1 设置 Segger JLink 工具路径



- 1、切换到插件功能界面
- 2、点击【Set Build/Debug Tool Path】按钮
- 3、点击【SEGGER JLink Tool Path】按钮
- 4、选择 JLink 根目录
- 5、插件自动安装烧录算法到 JLink 并配置 Cortex-Debug 插件参数
- 5、设置完成后，右下角会显示结果提示信息

## 2.7.2 运行调试



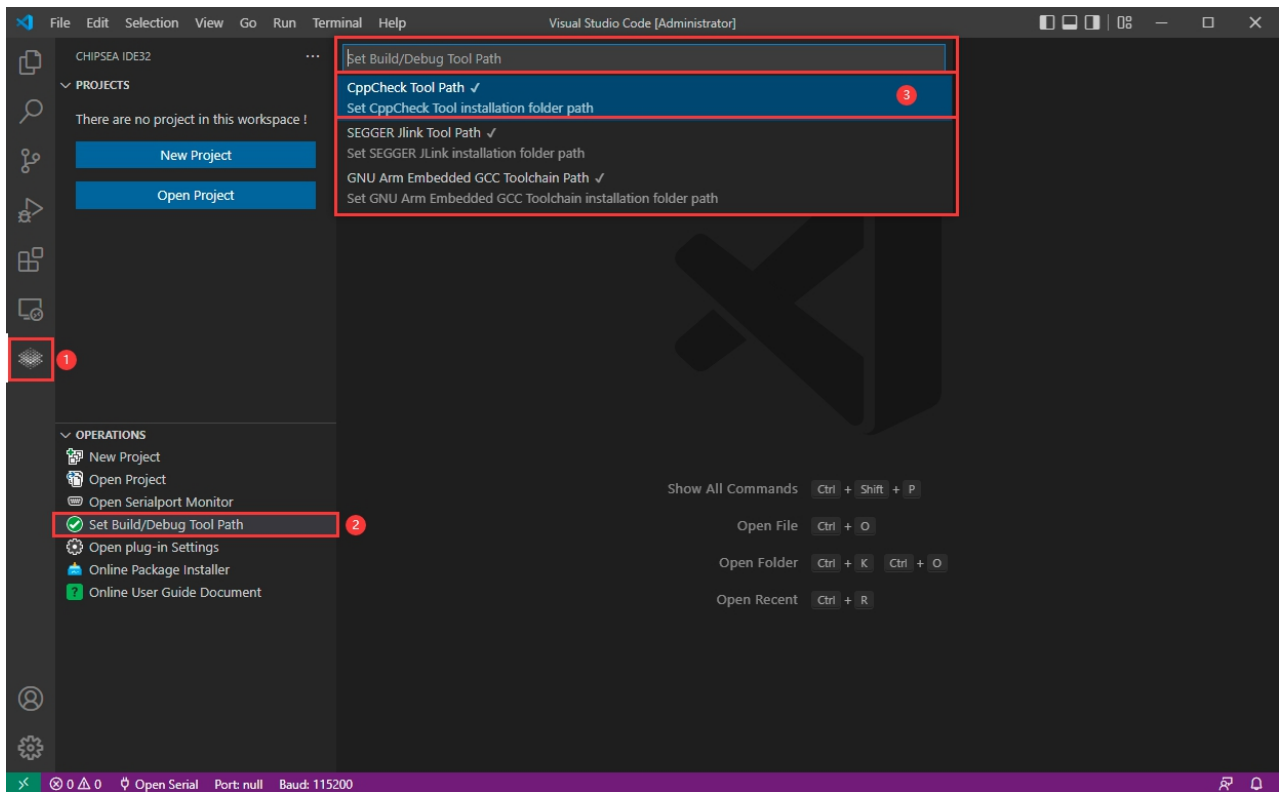
- 1、点击【Run And Debug】按钮
- 2、点击【Start Debugging】按钮
- 3、如果正确进入调试会停在 main 函数中

## 三、插件辅助功能

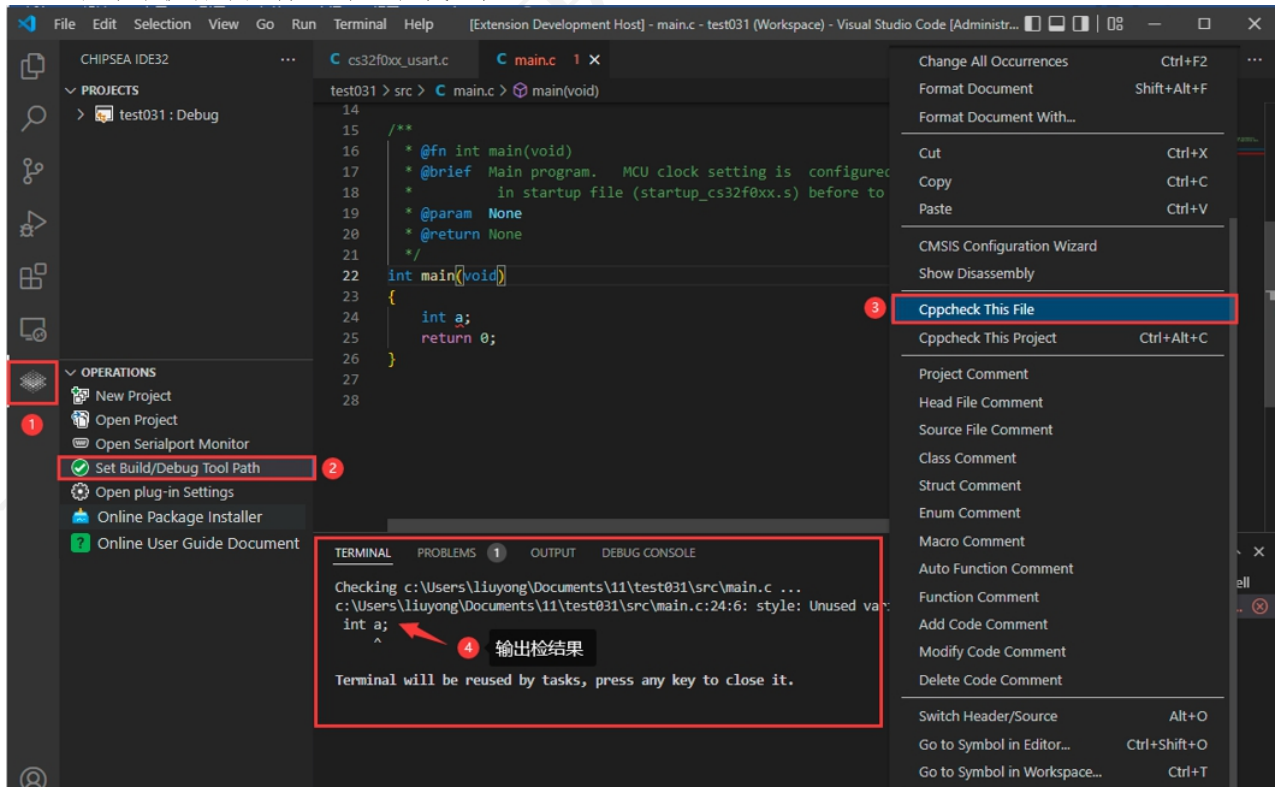
### 3.1 CppCheck 功能

增加 CppCheck 单文件和整个工程代码检查，只需要配置 CppCheck 路径，就可以通过菜单执行命令

- 1、设置 CppCheck 工具路径



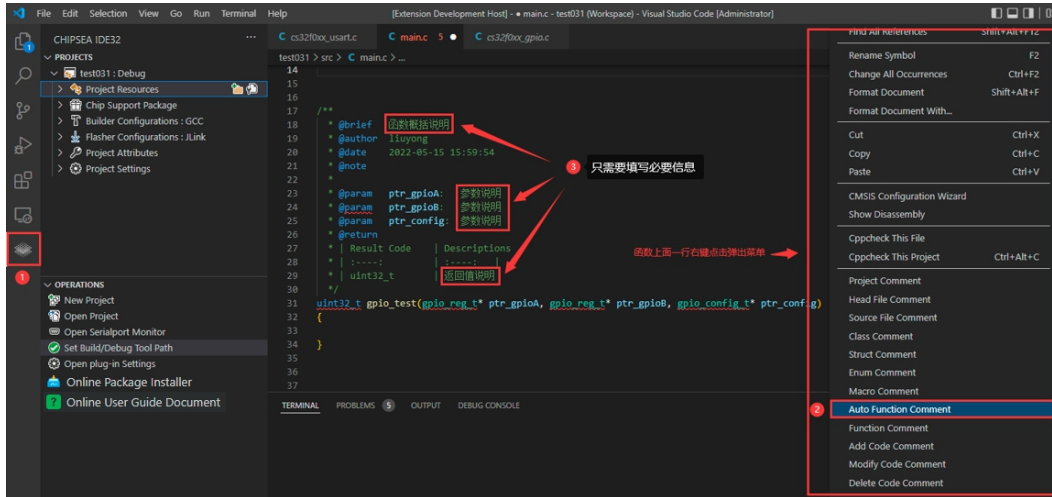
## 2、右键菜单执行单文件/整个工程代码检查



## 3.2 代码注释功能

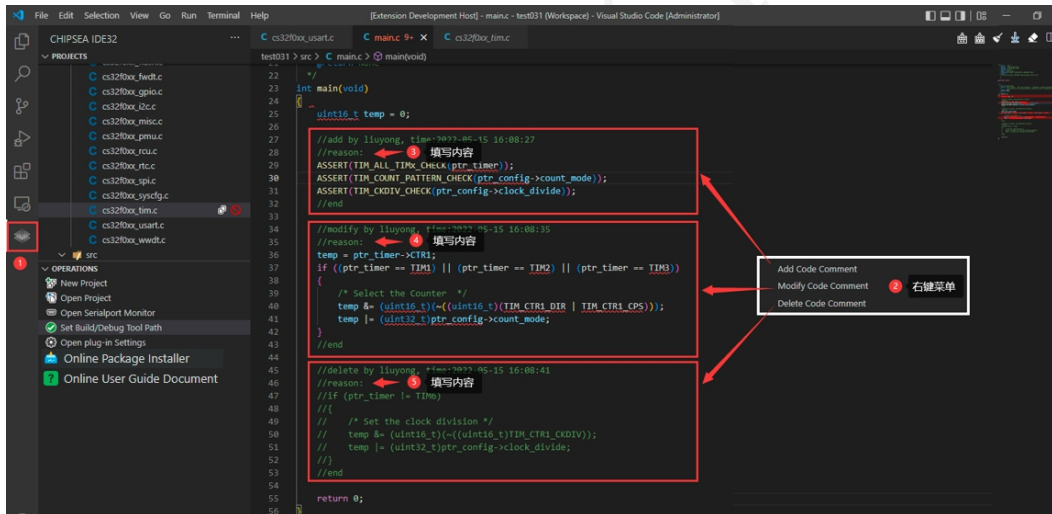
代码注释每天都会用到，很多时候都是从别的地方复制或自己写，必填的作者、日期时间不同的用户都要自己处理。增加的注释功能可以自动获取系统登录用户名和时间进行填充，提高工作效率。在编写复杂函数的注释时会更加耗时，也容易出错，我们提供了自动函数注释功能，可以分析提取函数参数信息自动生成注释，使用者只需要填写必要字段就可以了。

### 1、自动函数注释功能

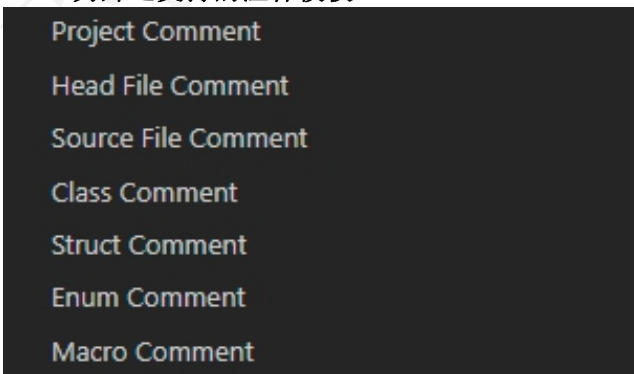


### 2、增/删/改注释功能

先选中需要增/删/改的代码，再点右键菜单中的对应菜单，会自动生成如下注释



### 3、另外还支持注释模板





芯海科技  
CHIPSEA

股票代码:688595

### 免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。