版本：V0.3 密级：公开

**瘦西湖操作系统文件**

瘦西湖操作系统

IDE工具手册

中科物栖（南京）科技有限责任公司

二〇二三年十一月

文档修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修改内容描述** | **修改人** | **日期** |
| V0.1 | 新建文档。 | 张梦瑶、袁哲 | 2023-09-07 |
| V0.2 | 精简驱动、TAL开发，添加填写main | 张梦瑶、袁哲 | 2023-09-22 |
| V0.3 | 补充配置安装环境的常见问题解决方法 | 张梦瑶 | 2023-11-13 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1 目的 1](#_Toc4880)

[2 配置安装环境 1](#_Toc2967)

[3 瘦西湖IDE工具安装 3](#_Toc4474)

[4 下载或导入开发环境 4](#_Toc19807)

[4.1 下载开发环境 4](#_Toc7478)

[4.2 导入已有开发环境 7](#_Toc7817)

[5 开发环境功能介绍 10](#_Toc5563)

[6 创建项目 11](#_Toc5530)

[7 开发驱动 14](#_Toc8397)

[8 开发TAL 15](#_Toc27954)

[8.1 编写IDL 15](#_Toc21177)

[8.2 编写TAL 21](#_Toc17388)

[9 编写main 22](#_Toc17706)

[10 Kconfig 22](#_Toc21600)

[11 编译、烧录、清除 23](#_Toc31136)

[11.1 编译 23](#_Toc11815)

[11.2 烧录固件 24](#_Toc3305)

[11.3 清除 26](#_Toc18)

[11.4 清除、编译、烧录 27](#_Toc11007)

[12 烧录密钥 28](#_Toc17643)

[13 开启日志 30](#_Toc28476)

[14 TAL调试 33](#_Toc28543)

[14.1 TAL本地调试 33](#_Toc31425)

[14.2 TAL物栖云调试(功能暂未上线) 36](#_Toc22471)

[15 插件快捷键介绍 37](#_Toc7043)

# 目的

本文档用于介绍瘦西湖操作系统VSCode Jeejio-SlenderWestLakeOS-IDE插件工具（下文简称瘦西湖IDE工具）的使用。

# 配置安装环境

瘦西湖IDE工具支持Windows10、Windows11和Linux操作系统环境。在Windows环境下使用瘦西湖IDE须先配置符号链接，方法如下：

1、运行gpedit.msc，打开“本地组策略编辑器”。

2、依次展开“计算机配置”-->“Windows设置”-->“安全设置”-->“本地策略”-->“用户权限分配”，在右侧窗口双击“创建符号链接”，在弹出的窗口中点击“添加用户或组”按钮，填入当前的用户名，确认。

3、重启电脑生效。

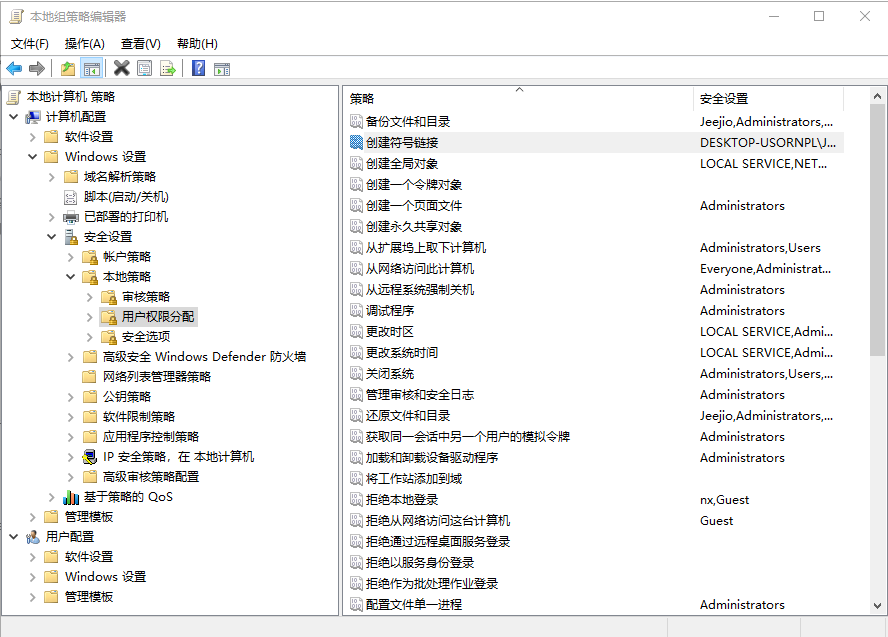


图2-1 配置符号链接

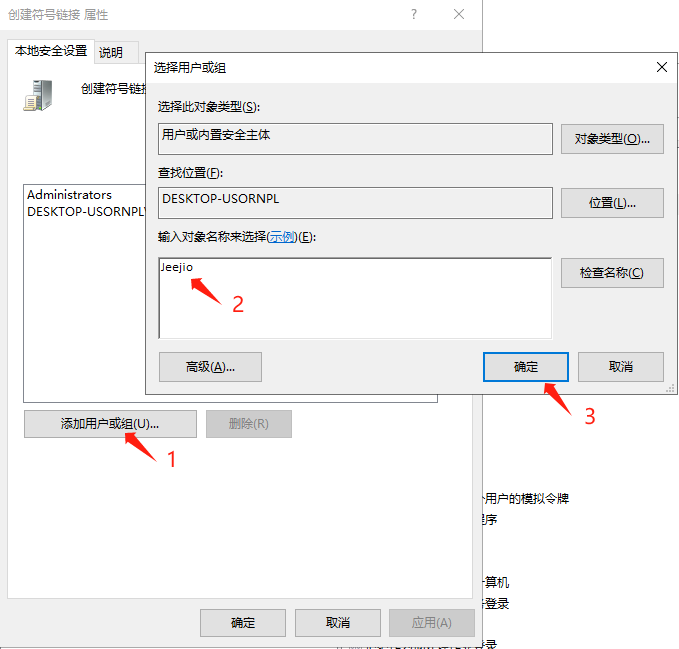


图2-2 添加用户名到符号链接

4、常见问题解决

（1）运行 gpedit.msc，若报错"windows找不到文件gpedit.msc"，则按照以下步骤：

①新建一个记事本，然后把下面的内容复制到记事本中。

@echo off

pushd "%~dp0"

dir /b C:\Windows\servicing\Packages\Microsoft-Windows-GroupPolicy-ClientExtensions-Package~3\*.mum >List.txt

dir /b C:\Windows\servicing\Packages\Microsoft-Windows-GroupPolicy-ClientTools-Package~3\*.mum >>List.txt

for /f %%i in ('findstr /i . List.txt 2^>nul') do dism /online /norestart /add-package:"C:\Windows\servicing\Packages\%%i"

pause

②把后缀名改为.bat，右键以管理员身份运行。

③继续执行步骤1-3。

（2）运行 gpedit.msc，若报错"试图引用不存在的令牌"，则按照以下步骤：

①检查C:\Windows\SysWOW64目录下是否有secpol.msc和gpedit.msc，若无则把这两个文件复制到此目录即可。

②继续执行步骤1-3。

# 瘦西湖IDE工具安装

1、从网上下载最新的Visual Studio Code软件并安装到电脑。

2、打开VSCode，点击“扩展”按钮，进入插件搜索界面。

3、输入“jeejio”，出现Jeejio-SlenderWestLakeOS-IDE应用，点击右侧“安装”图标进行瘦西湖IDE工具的安装。

瘦西湖IDE工具的安装如下图所示：

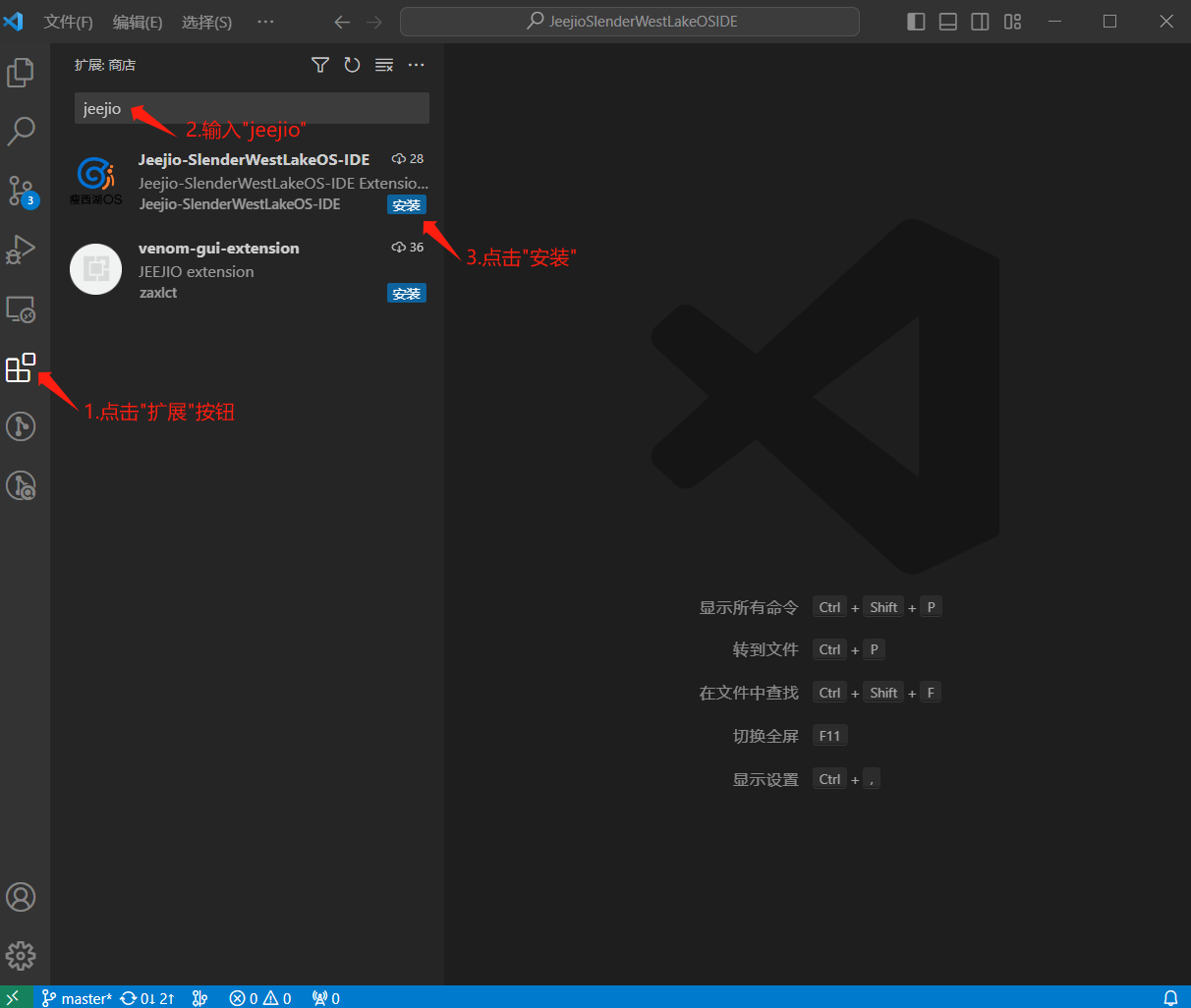


图3-1 安装瘦西湖IDE工具

安装完毕后出现下图界面。

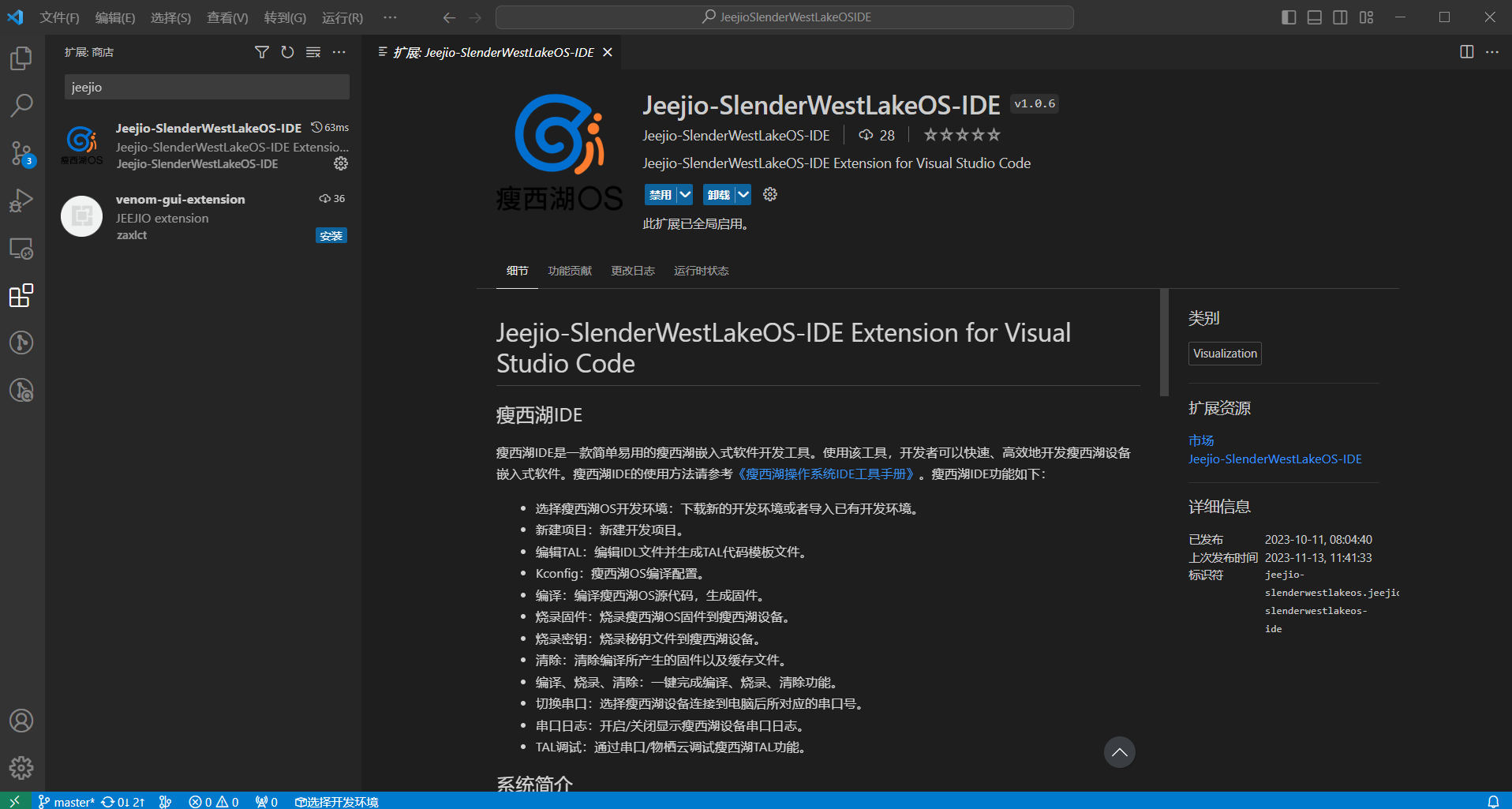


图3-2 安装IDE工具完成

# 下载或导入开发环境

## 下载开发环境

1、点击IDE窗口左下角“选择开发环境”按钮，在IDE窗口顶端出现“选择JEEJIO芯片开发环境”下拉菜单，选择“下载新的开发环境”。

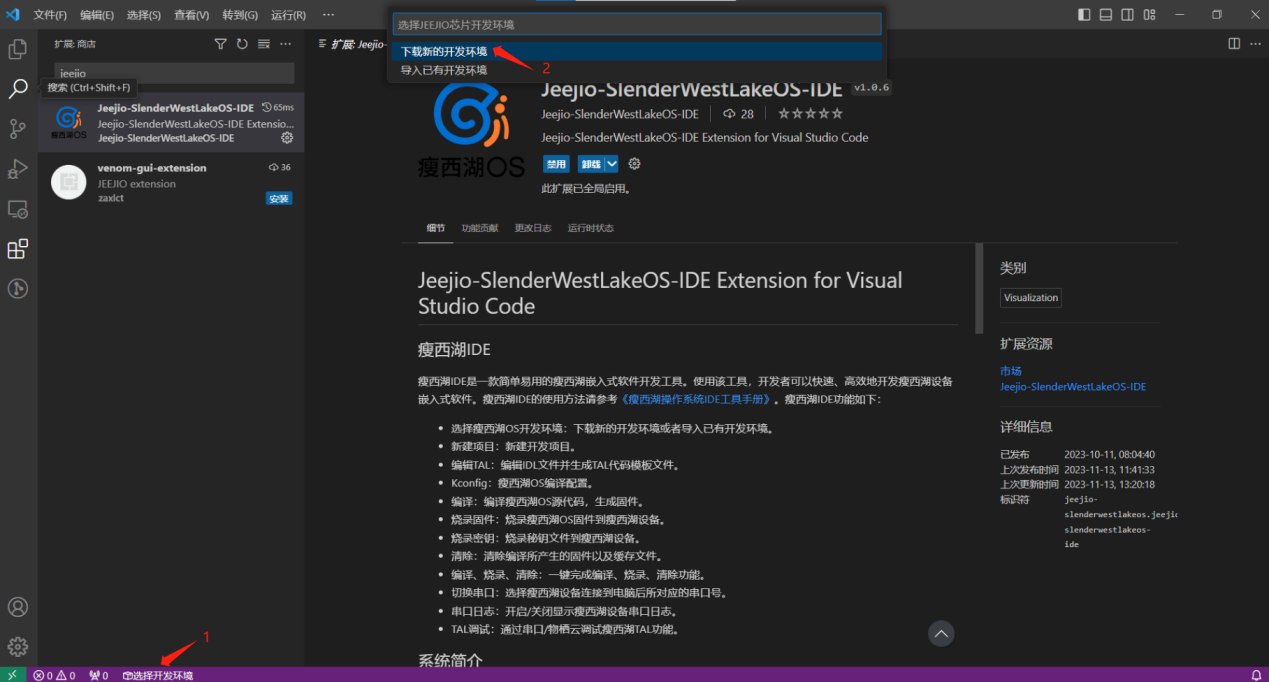


图4-1 下载开发环境操作入口

2、在弹出的“下载开发环境”窗口进行开发环境的初始化。

（1）点击“选择芯片型号”下拉菜单选择芯片.

（2）点击“选择瘦西湖OS版本”下拉菜单选择芯片的版本。

（3）点击“选择Jeejio-SlenderWestLakeOS-IDE开发环境目录”文件夹按钮选择PC中已有的文件目录。

（4）最后点击“下载环境”按钮开始下载开发环境。

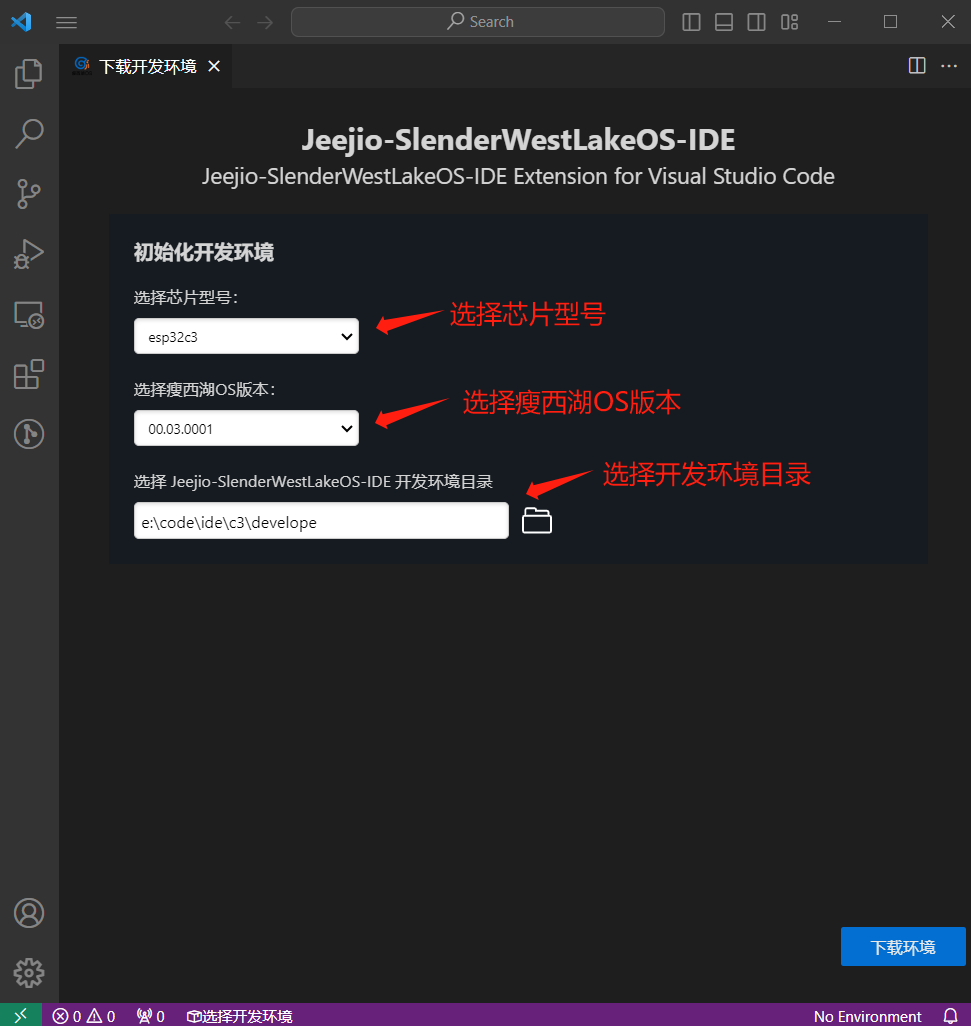


图4-2 初始化下载环境



图4-3 开始下载开发环境

3、完成开发环境下载后将自动弹出新建项目窗口。若不需要新建项目，则忽略该窗口。

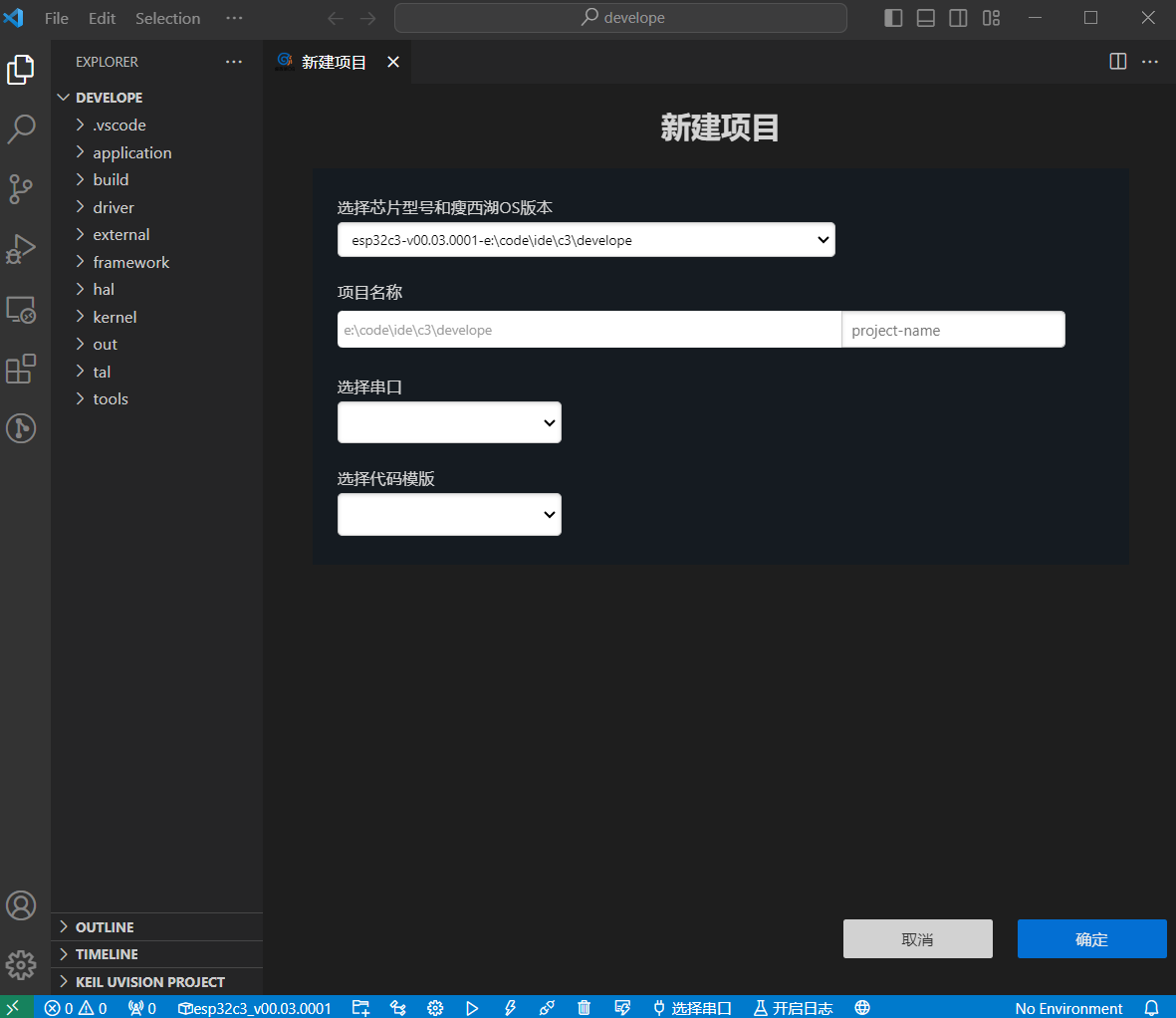


图4-4 开发环境下载完毕后的界面

## 导入已有开发环境

如果本地电脑中已经下载了开发环境，可以通过“导入已有开发环境”功能导入到IDE中。

1、点击IDE窗口左下角“选择开发环境”按钮，在IDE窗口顶端出现“选择JEEJIO芯片开发环境”下拉菜单，选择“导入已有开发环境”。

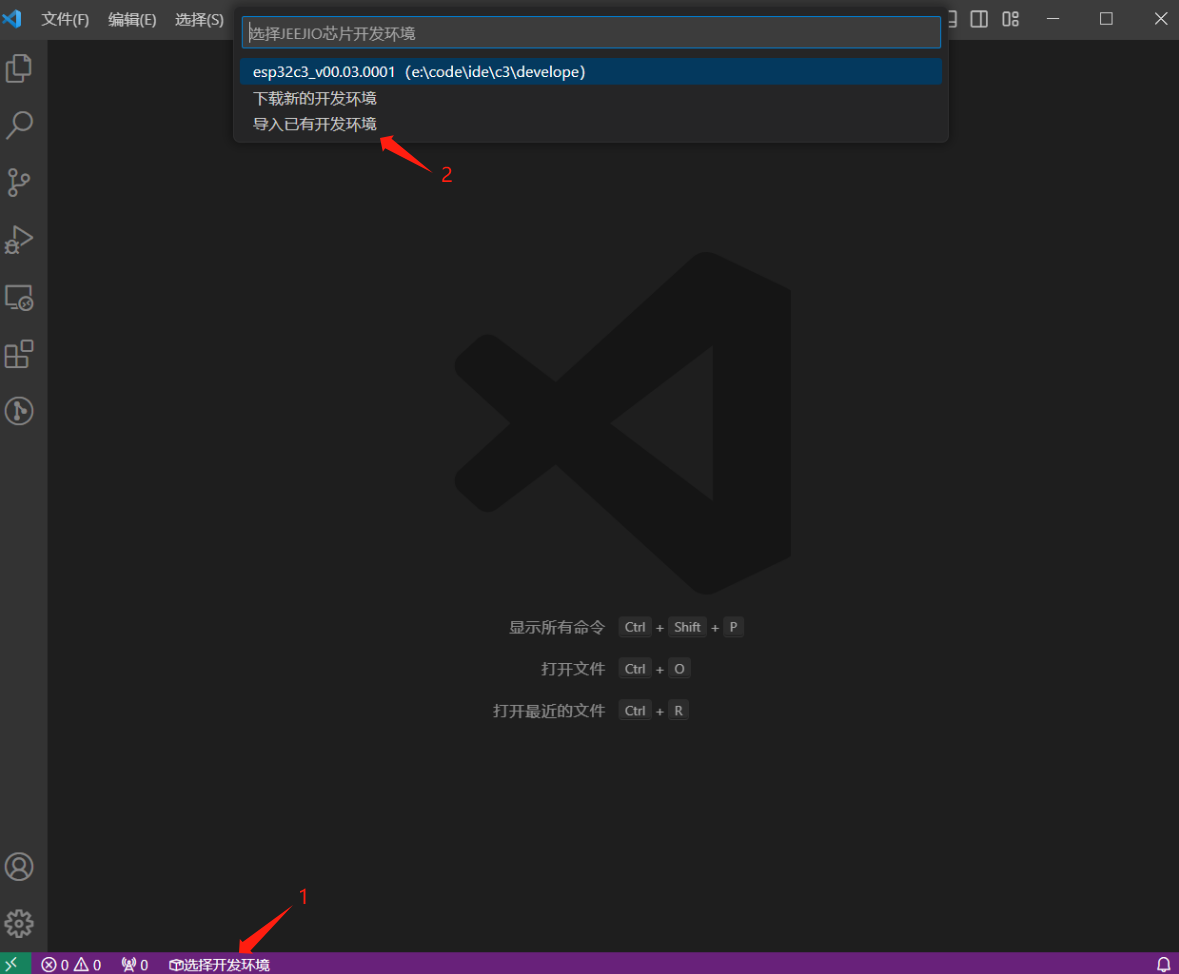


图4-5 导入已有开发环境操作入口

2、在弹出的导入开发环境窗口进行选择。

（1）点击“选择芯片型号”下拉菜单选择芯片.

（2）点击“选择瘦西湖OS版本”下拉菜单选择芯片的版本。

（3）点击“选择Jeejio-SlenderWestLakeOS-IDE开发环境目录”文件夹按钮选择本地电脑中已经下载的开发环境存放目录。

（4）最后点击“导入已有环境”按钮。



图4-6 初始化导入的开发环境

3、完成导入开发环境后将自动弹出新建项目窗口。若不需要新建项目，则忽略该窗口。

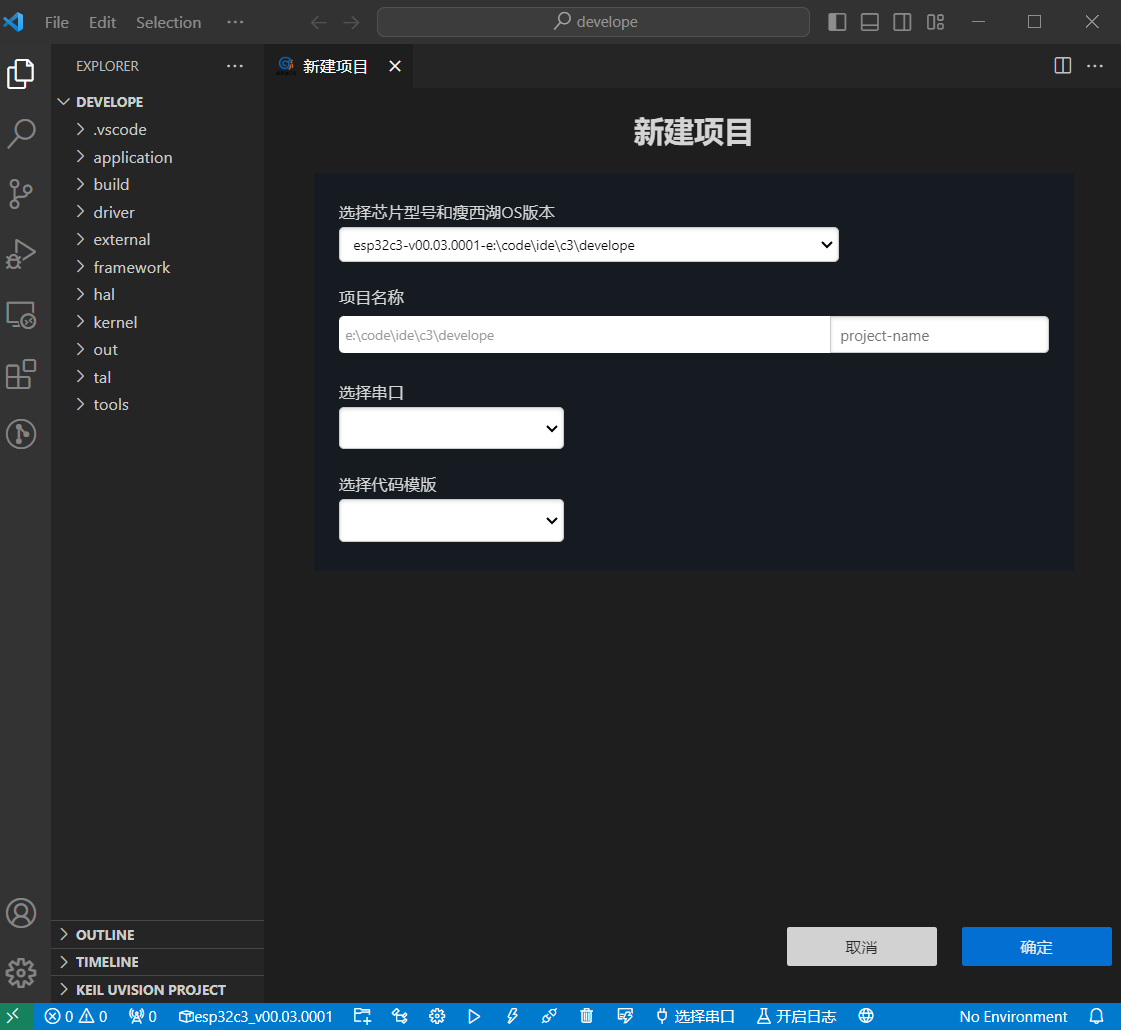


图4-7 导入开发环境完成后的界面

# 开发环境功能介绍

完成下载或导入开发环境后，在IDE窗口底部将出现瘦西湖IDE工具栏。



图5-1 IDE工具栏

1、选择瘦西湖OS开发环境开发环境：下载新的开发环境或者导入已有开发环境。

2、新建项目：新建开发项目。

3、编辑TAL：编辑IDL文件并生成TAL代码模板文件。

4、Kconfig：瘦西湖OS编译配置。

5、编译：编译瘦西湖OS源代码，生成固件。

6、烧录固件：烧录瘦西湖OS固件到瘦西湖设备。

7、烧录密钥：烧录密钥文件到瘦西湖设备。

8、清除：清除编译所产生的固件以及缓存文件。

9、编译、烧录、清除：一键完成编译、烧录、清除功能。

10、切换串口：选择瘦西湖设备连接到电脑后所对应的串口号。

11、串口日志：开启/关闭显示瘦西湖设备串口日志。

12、TAL调试：通过串口/物栖云调试瘦西湖TAL功能。

# 创建项目

1、点击IDE窗口底部“新建项目”工具按钮。



图6-1 点击“新建项目”按钮

2、在弹出的新建项目窗口设置项目参数。

（1）在“项目名称”下方“project-name”文本框中输入自定义的项目名称。

（2）在“选择串口”下拉菜单中选择所调试设备对应的串口号（设备串口号查询方法见下文）。

（3）在“选择代码模板”下拉菜单选择平台提供的代码模板。

（4）最后点击确定。

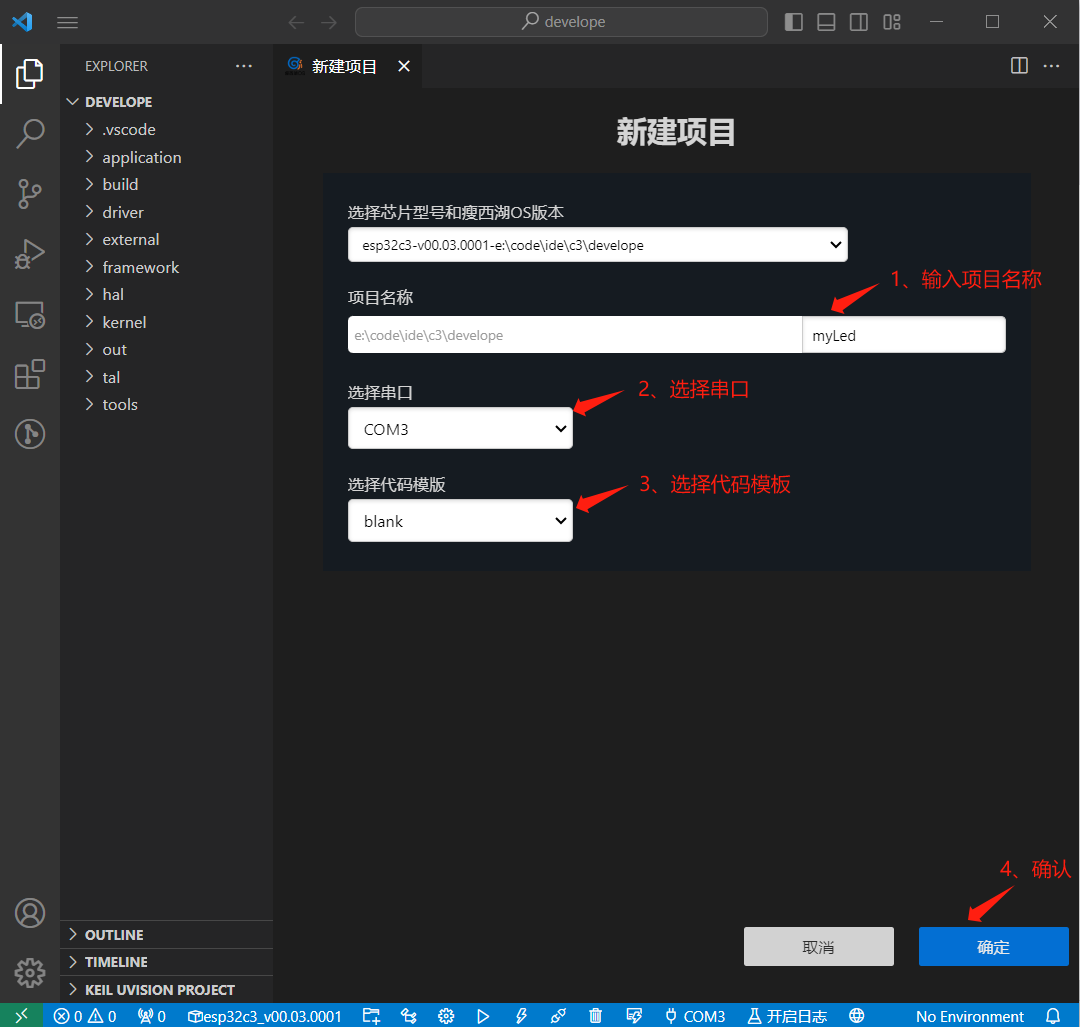


图6-2 设置新建项目

3、以新建台灯myLed项目为例，将弹出基于代码模板的项目开发窗口。

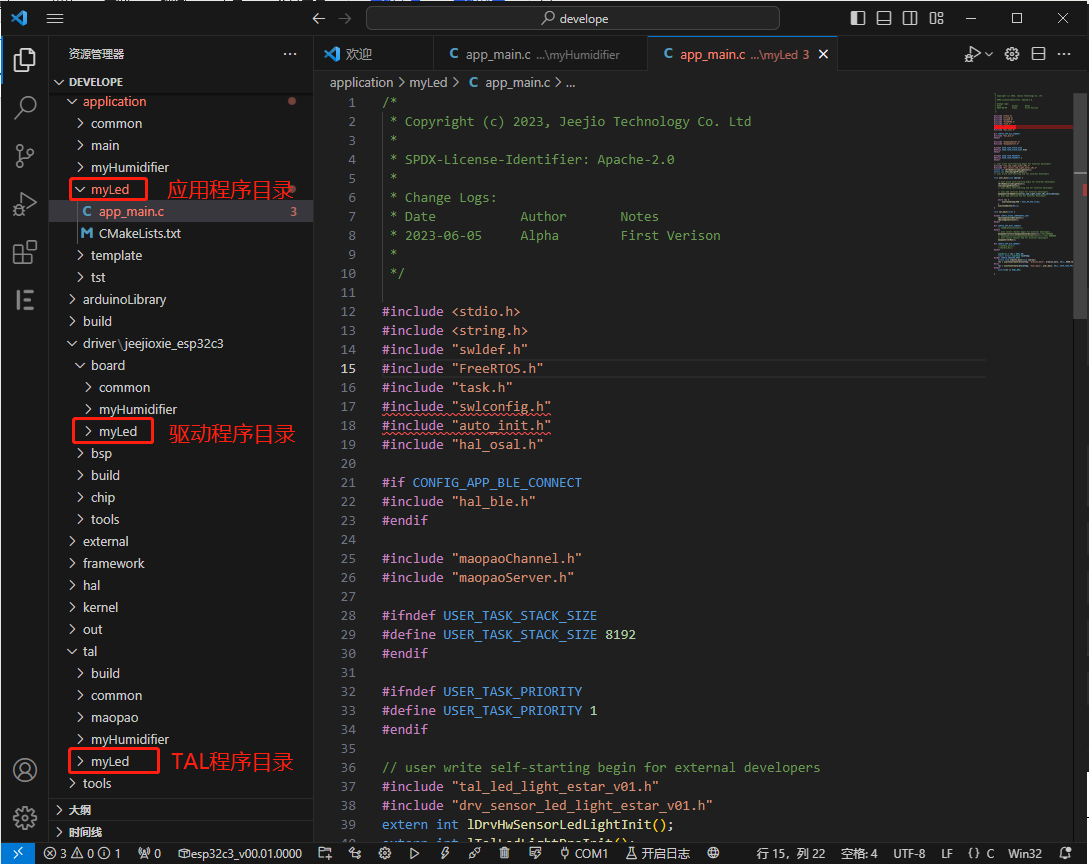


图6-3 基于代码模板的项目开发窗口

新建项目生成的代码目录结构如下：

${sdk}

├─application

│ ├─myLed

├─driver

│ └─jeejioxie\_esp32c3

│ ├─board

│ │ ├─common

│ │ └─myLed

└─tal

└─myLed

创建项目完成后即可开始开发项目。

# 开发驱动

下面继续以台灯myLed项目为例介绍使用IDE开发设备驱动。

**1、新建驱动空白源文件**

（1）在IDE左侧源码目录找到driver/jeejioxie\_esp32c3/board/myLed。

（2）鼠标选中“myLed”目录名，单击鼠标右键，在弹窗口选择“新建文件”，在myLed目录下新建led.c、led.h、drv\_sensor\_led.c、drv\_sensor\_led.h 4个空白文件。

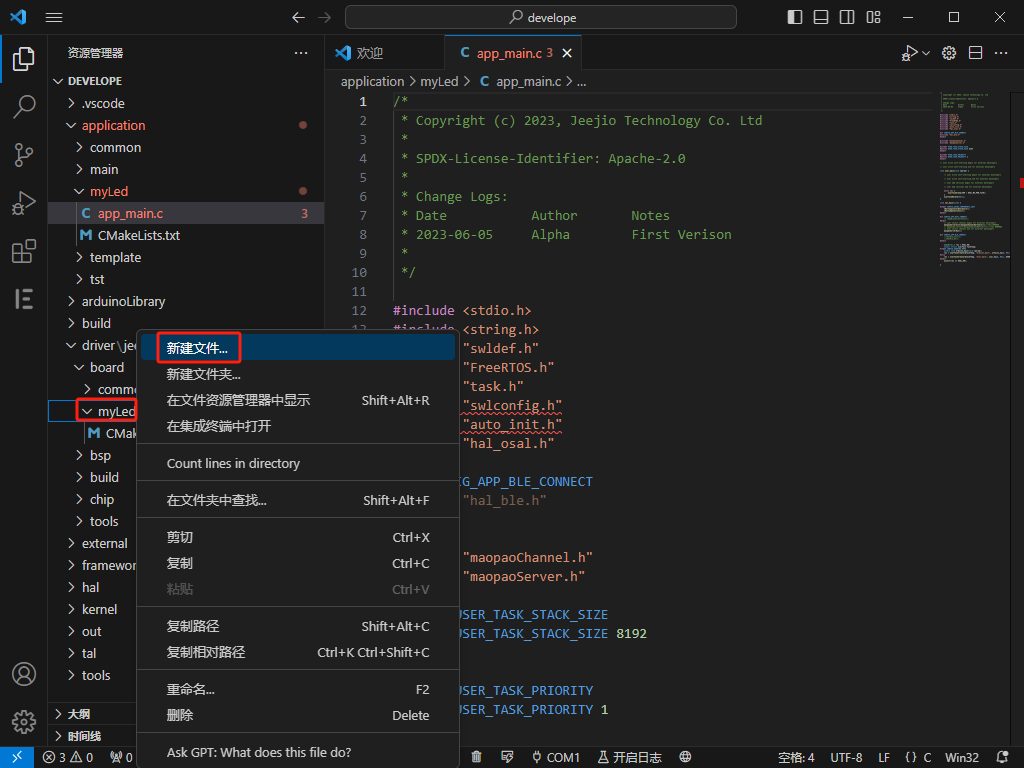


图7-1 新建驱动文件

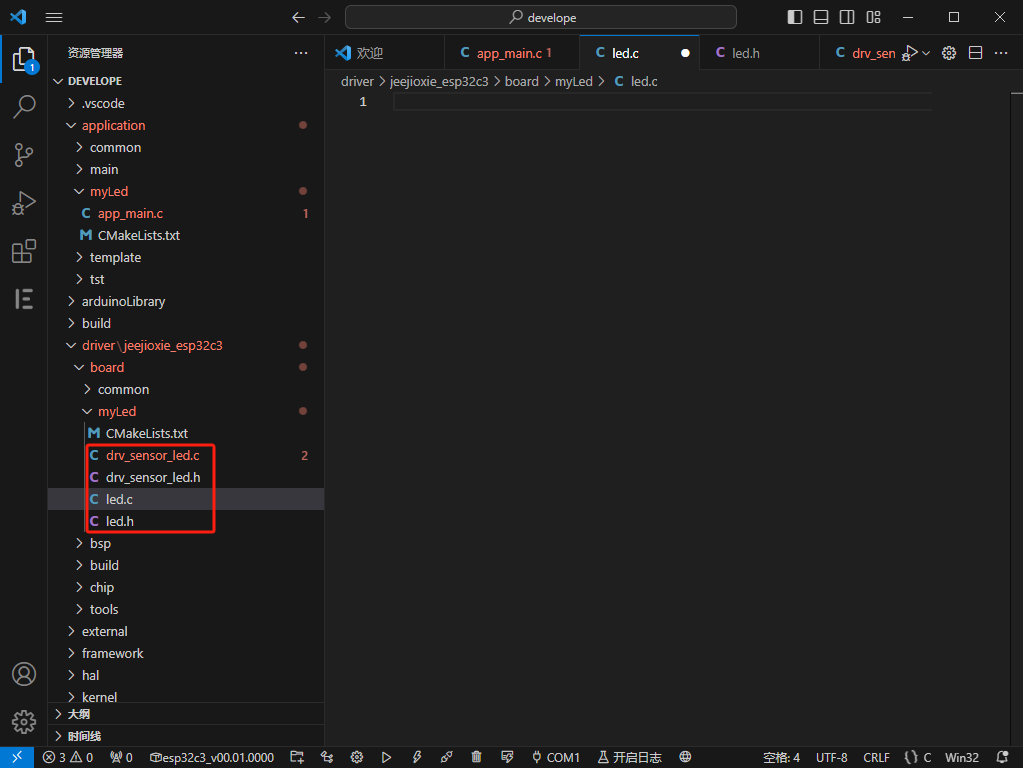


图7-2 新建驱动文件完成

台灯驱动程序目录结构如下所示。

driver

└── jeejioxie\_esp32c3

└── board

└── myLed

├── drv\_sensor\_led.c //驱动框架源文件

├── drv\_sensor\_led.h //驱动框架头文件

├── led.c //驱动源文件

└── led.h //驱动头文件

**2、编写台灯驱动**

在driver/jeejioxie\_esp32c3/board/myLed/led.c中编写台灯驱动，在led.h中编写台灯驱动头文件，在drv\_sensor\_led.c中编写台灯驱动框架程序，在drv\_sensor\_led.h中编写台灯驱动框架头文件。

瘦西湖OS驱动详细编写方法请参考[《火花计划-嵌入式开发岛开发者手册》](https://os.qajeejio.com:21080/swl/ide_usage_instructions/火花计划-嵌入式开发岛开发者手册.docx)。

# 开发TAL

## 编写IDL

1、点击IDE窗口左下角“编辑TAL”按钮，在IDE窗口顶端出现“选择项目”下拉菜单。选择拟开发的项目。以myLed项目为例，如下图所示。

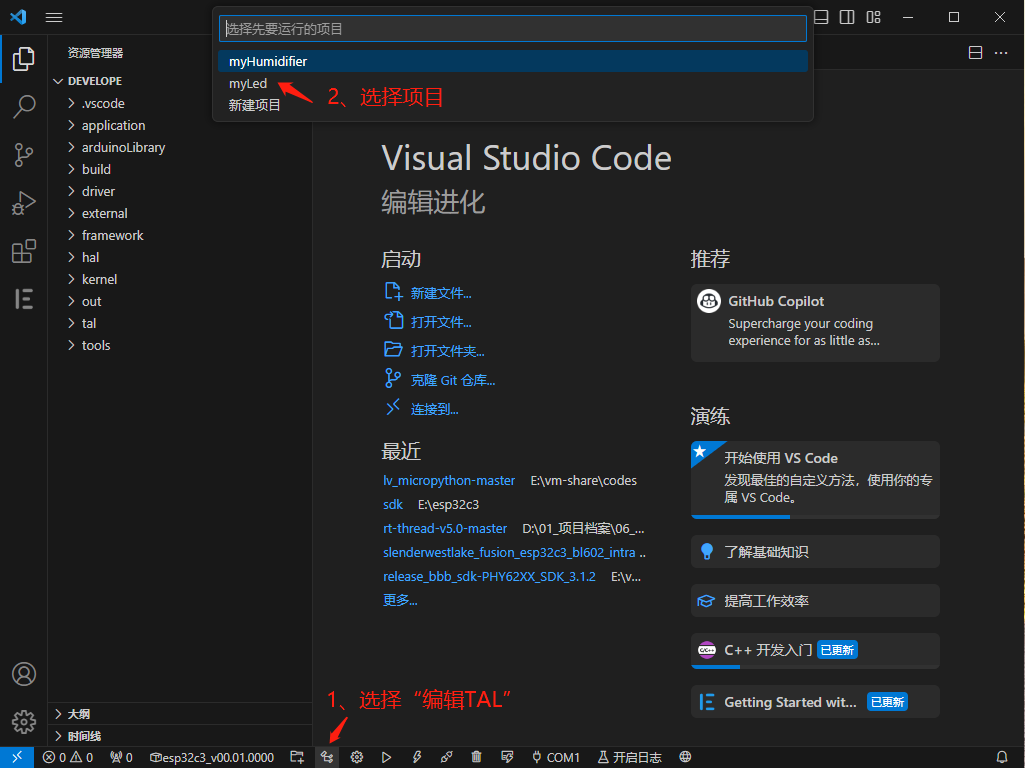


图8-1 开始编写IDL

2、选择项目后将弹出“编辑TAL”窗口，为瘦西湖IDE工具提供的IDL模板，在此模板中填写TAL详细信息。

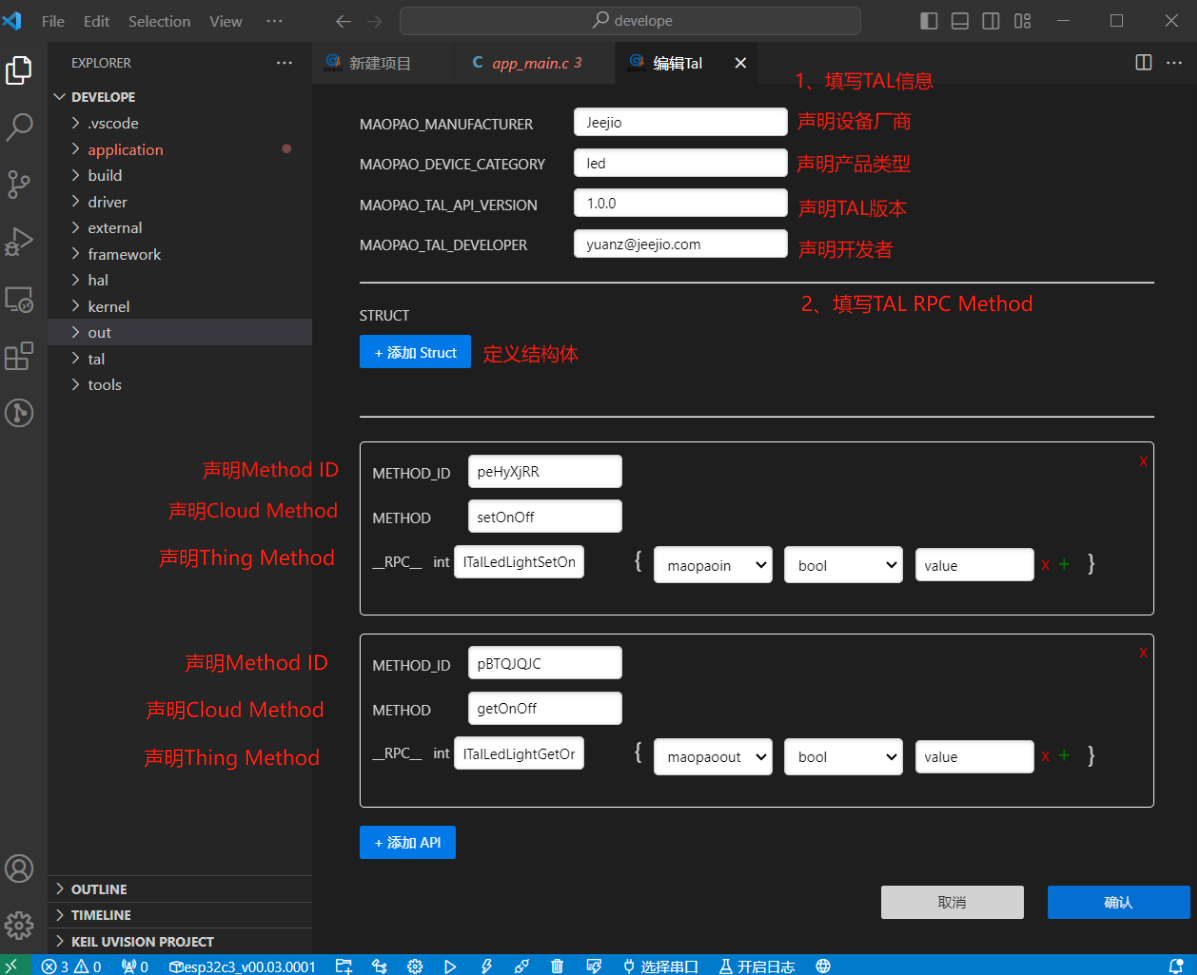


图8-2 编辑TAL界面

（1）填写TAL信息

在文本框填写本版本TAL程序的基本信息，包括该TAL设备的厂商、产品类型、TAL版本号、TAL开发者。

* MAOPAO\_MANUFACTURER：设备厂商名称
* MAOPAO\_DEVICE\_CATEGORY：产品类型
* MAOPAO\_TAL\_API\_VERSION：TAL程序的版本号
* MAOPAO\_TAL\_DEVELOPE：开发者名字或者开发者邮箱

示例：

// 声明设备厂商

MAOPAO\_MANUFACTURER("jeejio");

// 声明产品类型

MAOPAO\_DEVICE\_CATEGORY("led");

// 声明TAL版本

MAOPAO\_TAL\_API\_VERSION("1.0.0");

// 声明开发者

MAOPAO\_TAL\_DEVELOPER("yuanz@jeejio.com");

（2）填写TAL RPC Method

IDL中包含一个或多个TAL RPC Method定义，每个定义均对应TAL设备的一个功能API。TAL RPC Method定义由声明Method ID、声明Cloud Method、定义Thing Method组成。

示例：

// 声明Method ID

@METHOD\_ID("peHyXjRR")

// 声明Cloud Method

@METHOD("setOnOff")

//定义Thing Method

\_\_RPC\_\_ int lTalLedLightSetOnOff(maopaoin bool value);

// 声明Method ID

@METHOD\_ID("pXZuSByA")

// 声明Cloud Method

@METHOD("getRelativeBrightness")

//定义Thing Method

\_\_RPC\_\_ int lTalLedLightGetRelativeBrightness(maopaoout int value);

1）点击“添加API”按钮添加TAL API。

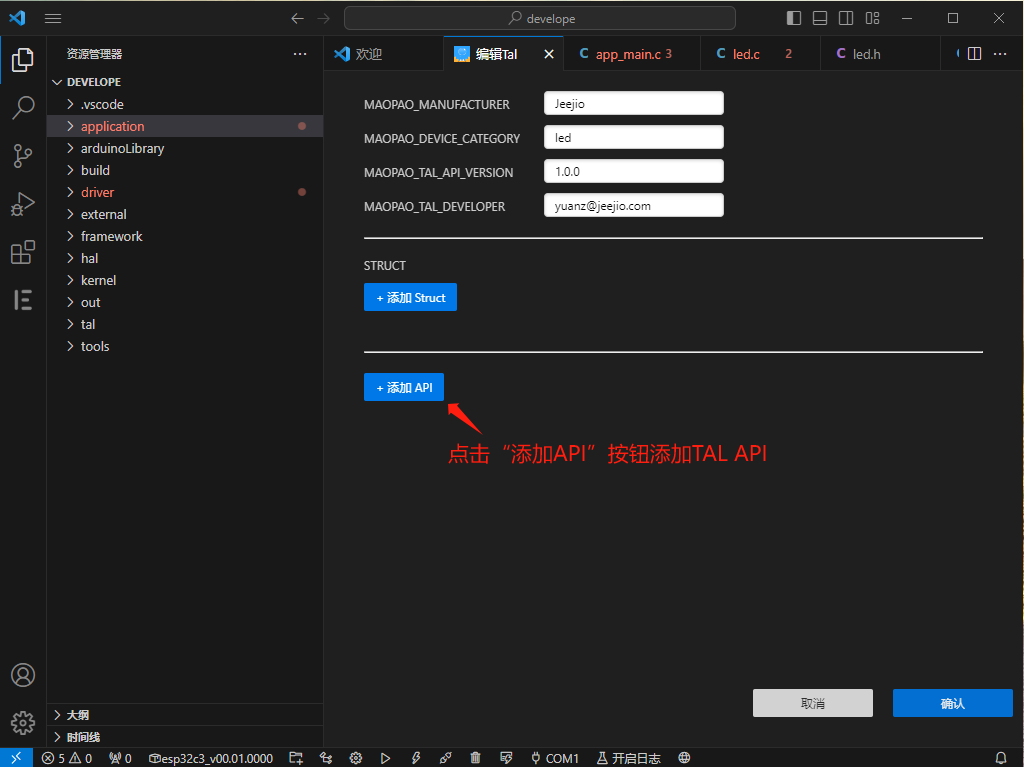


图8-3 添加TAL API

2）填写Method ID

Method ID是TAL RPC Method的标识码，每个TAL RPC Method对应唯一一个Method ID，每个Method ID对应唯一一个TAL RPC Method。Method ID直接从物栖TAL云平台获取，不可以修改。

3）填写Cloud Method

Cloud Method是物栖云平台定义的TAL设备功能API。物栖云平台通过Method ID和Cloud Method组合（RPC协议包）查询或者控制瘦西湖OS设备。Cloud Method直接从物栖TAL云平台获取，不可以修改。

4）定义Thing Method

Thing Method是瘦西湖OS设备端运行的设备功能函数，与Method ID及Cloud Method对应。IDL对Thing Method的定义参照C语言语法，格式如下：

\_\_RPC\_\_ 返回值类型 接口名称(maopaoout 参数1类型 参数1名称, maopaoout 参数2类型 参数2名称...maopaoout 参数n类型 参数n名称);

* \_\_RPC\_\_：标记该Method可以被物栖云平台远程调用，不添加该标识符则仅仅生成本地的接口模板，无法被远程调用。
* 返回值类型：固定为int型。Thing Method在执行完毕后会返回int型的错误码。
* 接口名称：定义Thing Method的函数名称，命名规则与C语言相同。
* 接口参数定义规则：
* 参数入口类型：TAL的参数入口类型分为输入参数和输出参数。输入参数是从云平台传给设备TAL API的数据，用maopaoin标记；输出参数是设备发给云平台的数据，用maopaoout标记。
* 参数类型：定义Thing Method的参数类型，规则与C语言相同。
* TAL接口可以有一个或多个输入参数，支持的基础参数类型有bool, int, char \*, float、结构体类型。
* TAL接口可以有一个或多个输出参数，支持的基础参数类型有bool, int, char \*, float、结构体类型，以及上述类型的数组。
* 参数名称：定义Thing Method的参数名称，命名规则与C语言相同。
* 在一个IDL中，Method ID、Cloud Method、Thing Method是唯一对应关系，不可以重复组合。

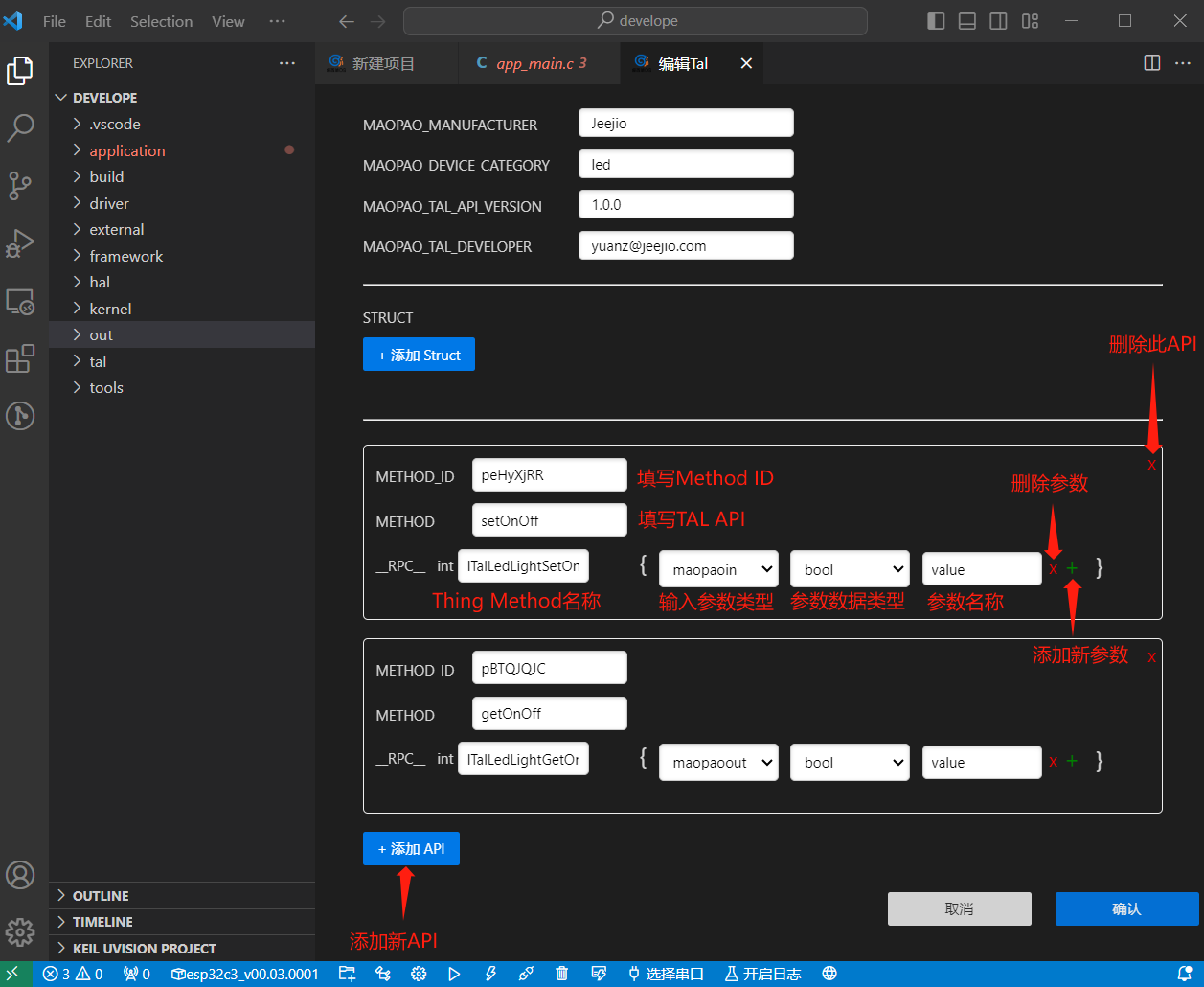


图8-4 填写TAL RPC Method

3、点击“确认”，将自动生成该项目的TAL代码模板，并保存到项目tal/${project\_name}/目录下。

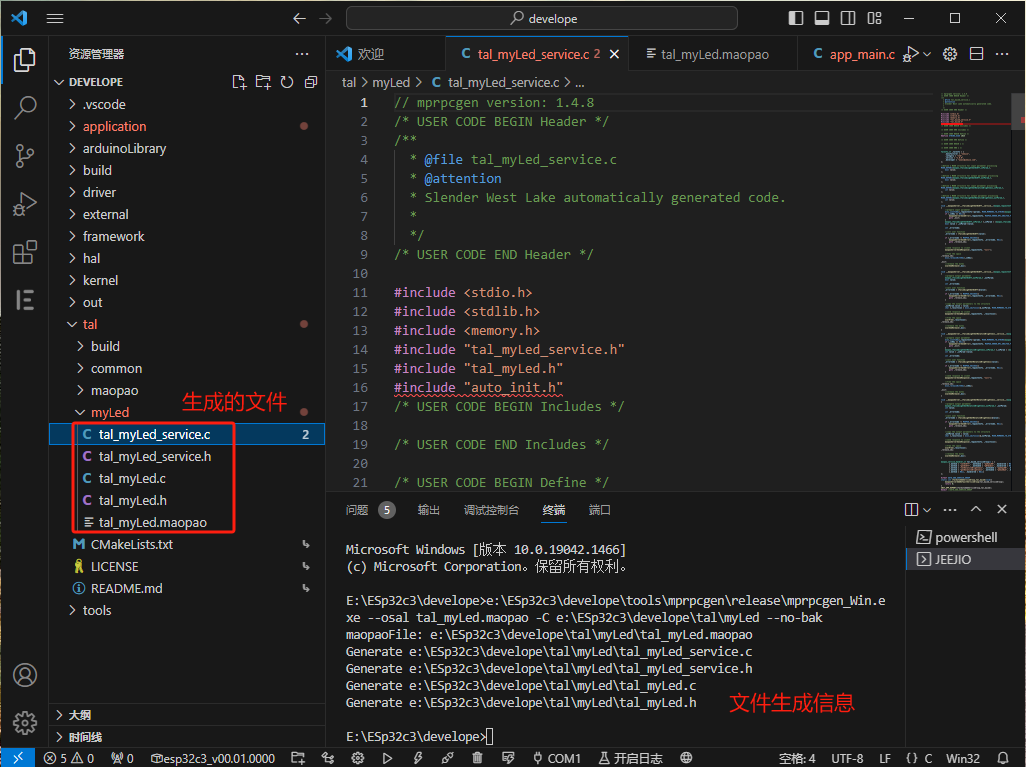


图8-5 生成TAL文件

生成的TAL代码目录结构如下：

tal

└─ myLed

├─ tal\_myLed.c //Led TAL c文件，需要开发者修改

├─ tal\_myLed.h //Led TAL 头文件，开发者无需修改

├─ tal\_myLed\_service.c //Led系统服务c文件，开发者无需修改

├─ tal\_myLed\_service.h //Led系统服务头文件，开发者无需修改

└─ tal\_myLed.maopao //Led IDL源文件，开发者无需修改

## 编写TAL

在<编写IDL>章节中介绍了编写IDL的方法，并通过IDE工具生成了TAL代码模板。接下来，只要在TAL代码模板中将Thing Method函数一一实现，即可完成TAL的编写。以myLed项目为例，在生成台灯TAL源文件过程中，台灯的功能函数API会按照模板自动填写到源文件tal\_myLed.c中。在tal\_myLed.c中编写程序完善TAL函数内容实现实际功能即可。

瘦西湖OS TAL详细编写方法请参考[《火花计划-嵌入式开发岛开发者手册》](https://os.qajeejio.com:21080/swl/ide_usage_instructions/火花计划-嵌入式开发岛开发者手册.docx)。

# 编写main

项目的主程序入口位于application/${project\_name}/app\_main.c，开发者可以在该文件中进行设备应用程序的开发。以myLed项目为例，必要的开发工作有：

1、包含tal头文件。

#include "tal\_myLed.h"

2、在user\_main()函数中添加服务tal\_\*\_serviceGroup。

maopaoServerAddServices(tal\_myLed\_serviceGroup);

tal\_myLed\_serviceGroup定义在tal\_led\_service.c中，在tal\_led\_service.h中声明，在编写IDL后由IDE工具自动生成，开发者不需要编写。其他项目与myLed项目类似。

3、在swl\_main()函数中选择启动RPC通信通道。

maopaoServerInit(maopaoChannelUartInit()); //TAL本地调试通道

maopaoServerInit(maopaoChannelMqttInit()); //TAL 冒泡通信通道

注意：上面两个通道二选一，只能同时开启一个。

4、编写自定义的应用程序。

# Kconfig

1、点击IDE工具底部工具栏“Kconfig”按钮。

2、IDE窗口将显示瘦西湖OS的配置界面。

3、在配置界面左侧目录可查看瘦西湖OS的所有配置选项。

4、点击所需配置项，在目录右侧界面设置对应参数。

5、点击“保存”按钮完成Kconfig配置。

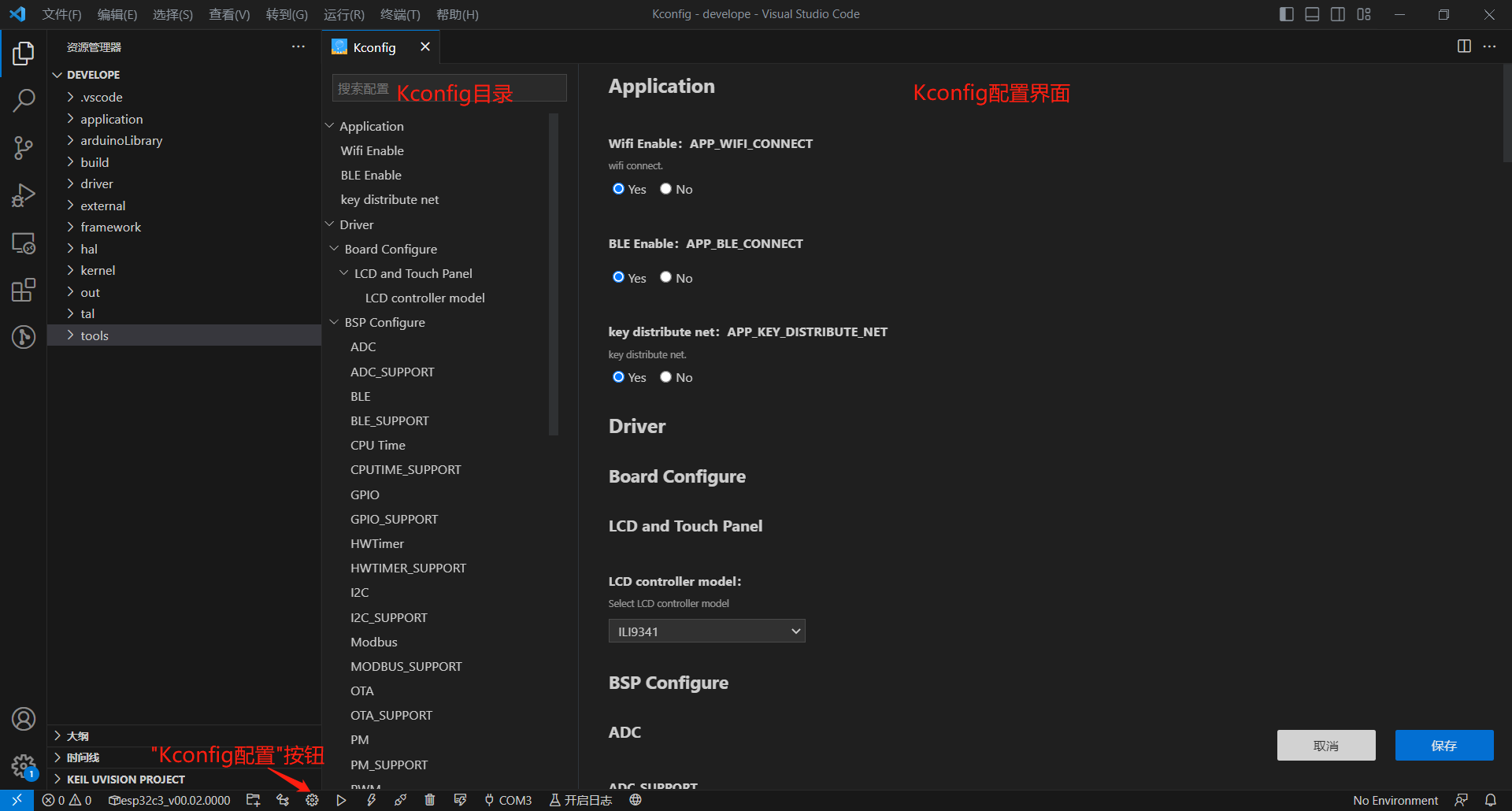


图10-1 Kconfig配置界面

仍以myLed项目为例，本项目没有特殊Kconfig配置需求，采用系统默认配置即可。

# 编译、烧录、清除

## 编译

（1）点击IDE底部工具栏中“编译”按钮开始编译代码。

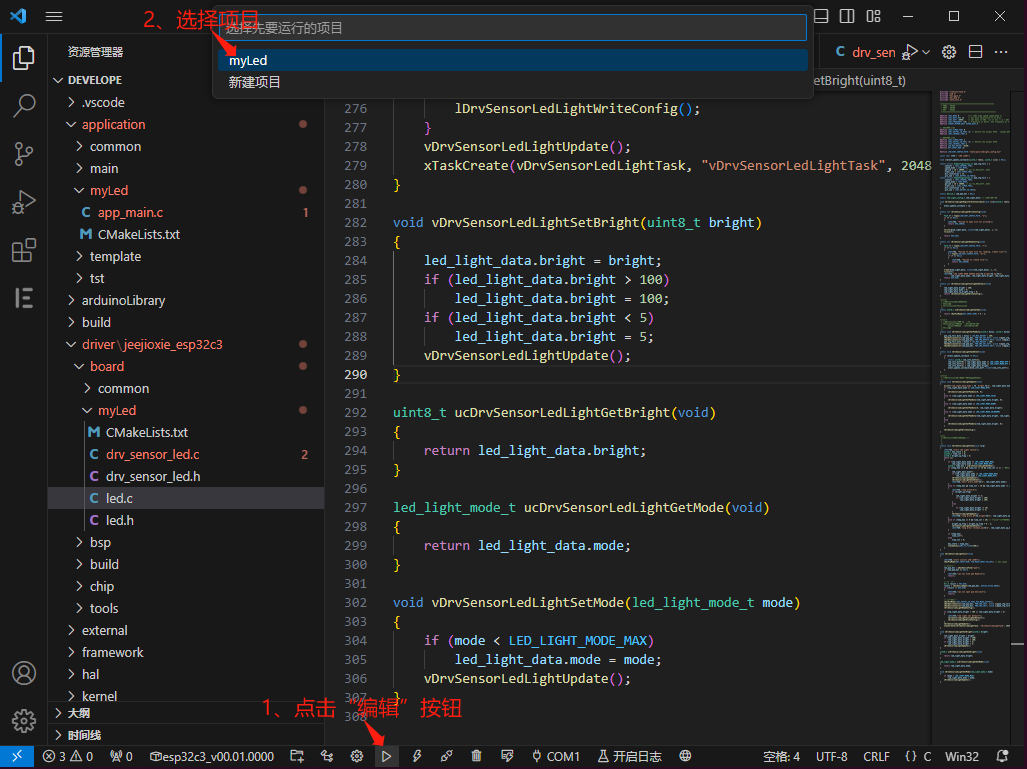


图11-1 开始编译

（2）编译成功后将出现成功提醒。

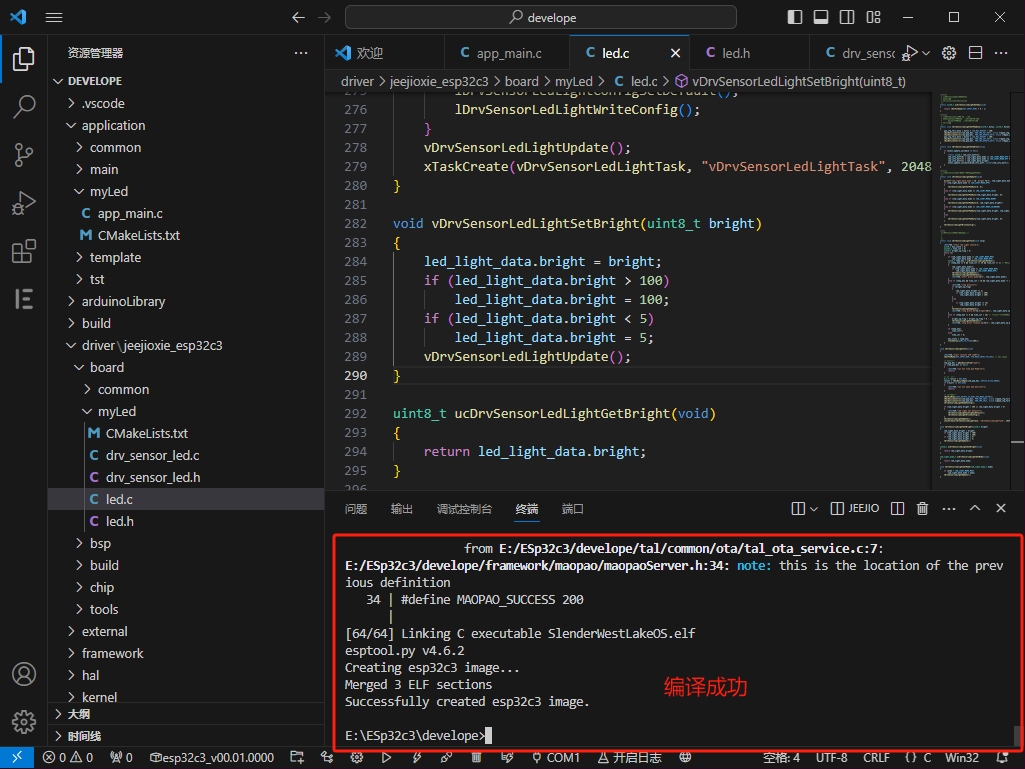


图11-2 编译成功

## 烧录固件

（1）选择设备对应的串口号。点击IDE底部工具栏中“切换串口”按钮，在IDE顶部“选择串口”下拉菜单中选择设备对应的串口号。

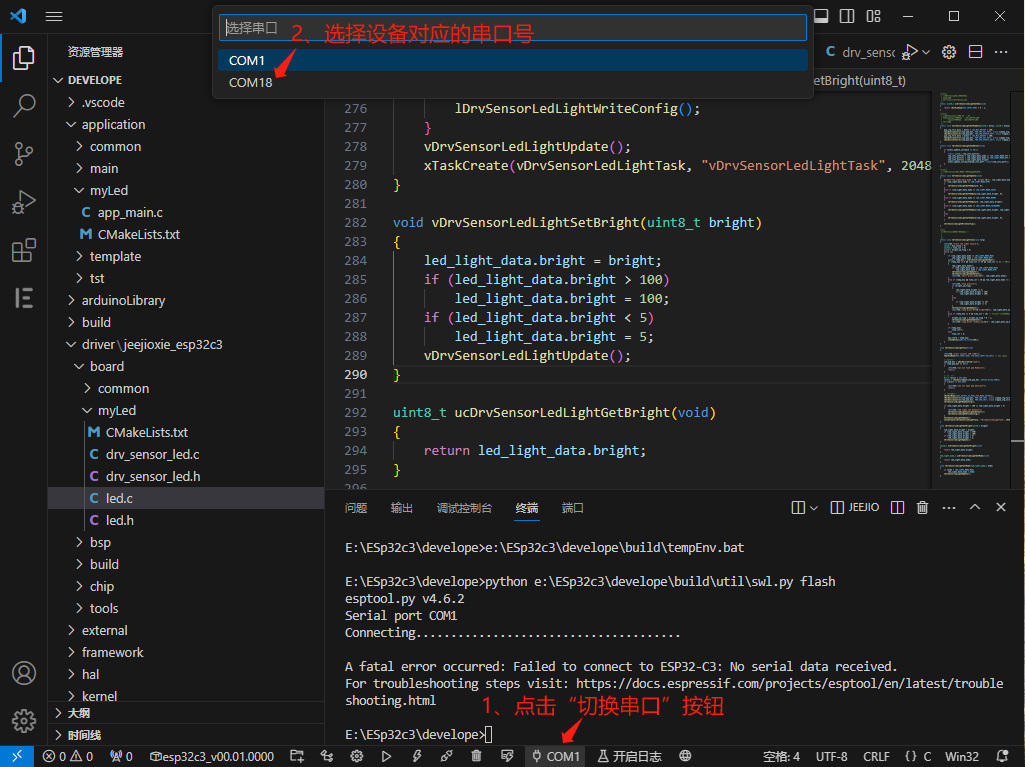


图11-3 切换到设备对应的串口

（2）点击“烧录固件”按钮开始烧录固件。

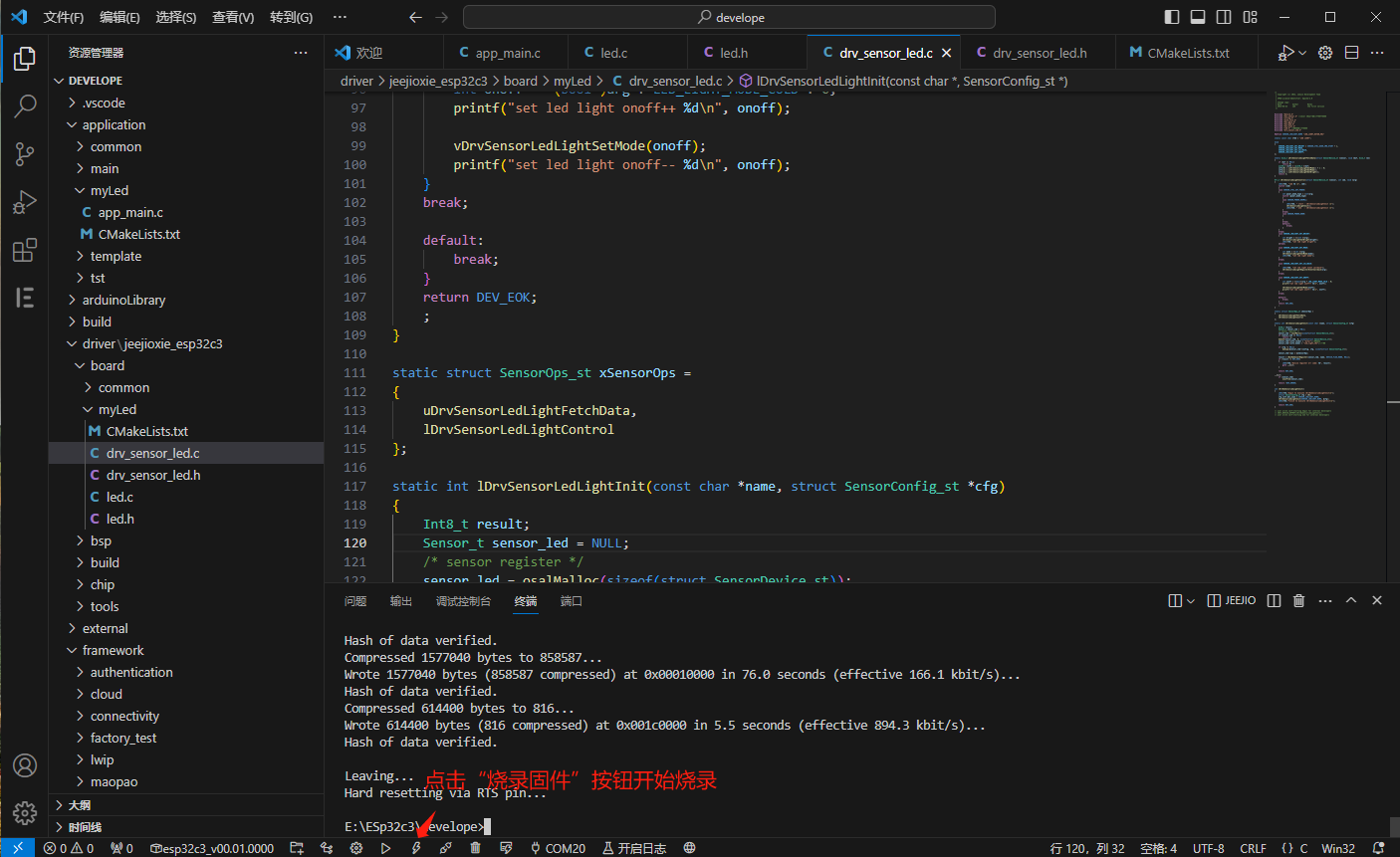


图11-4 开始烧录

（3）固件烧录完毕后将出现如下类似界面。

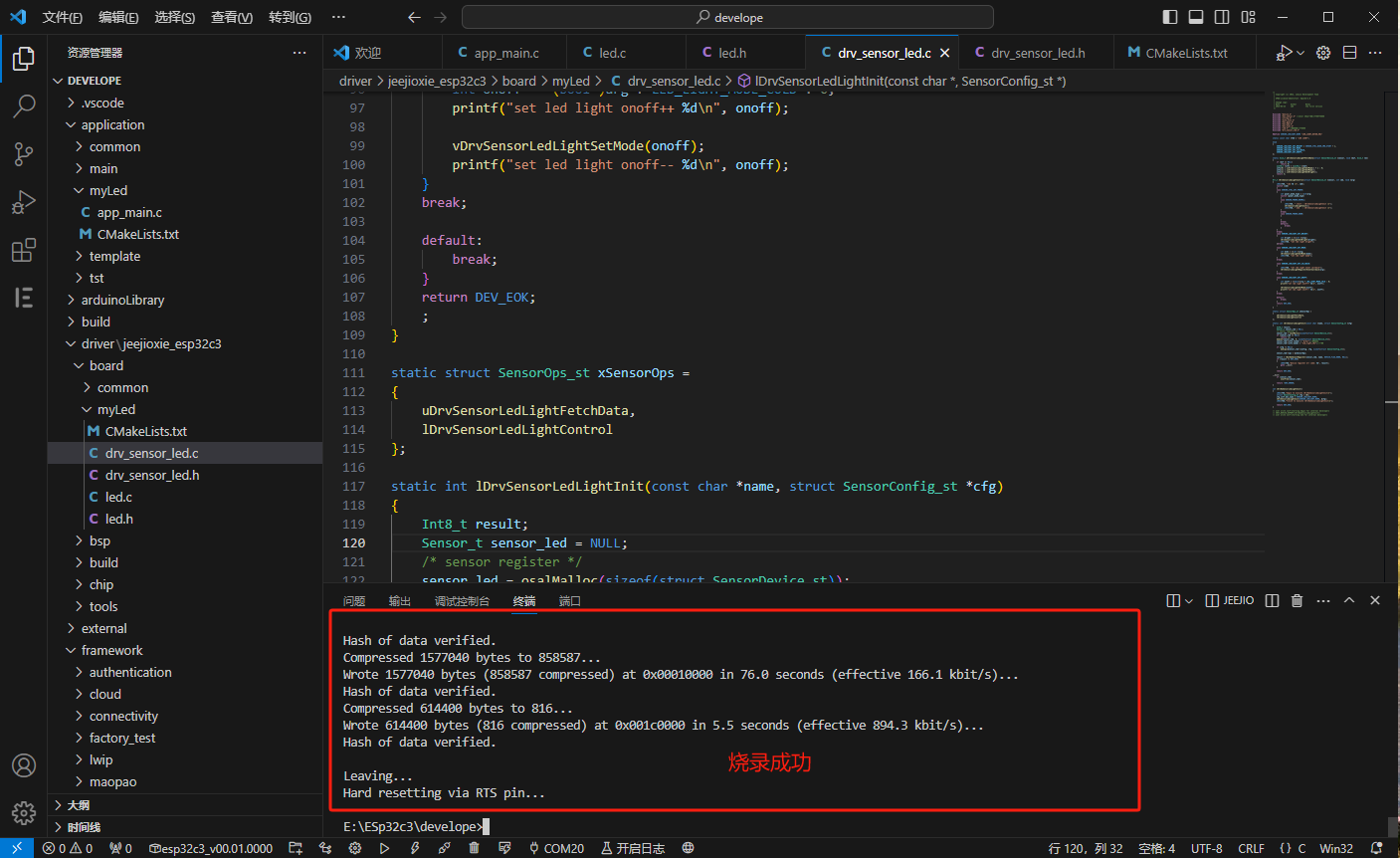


图11-5 烧录完成

## 清除

（1）点击IDE底部工具栏中“清除”按钮开始清除编译记录。

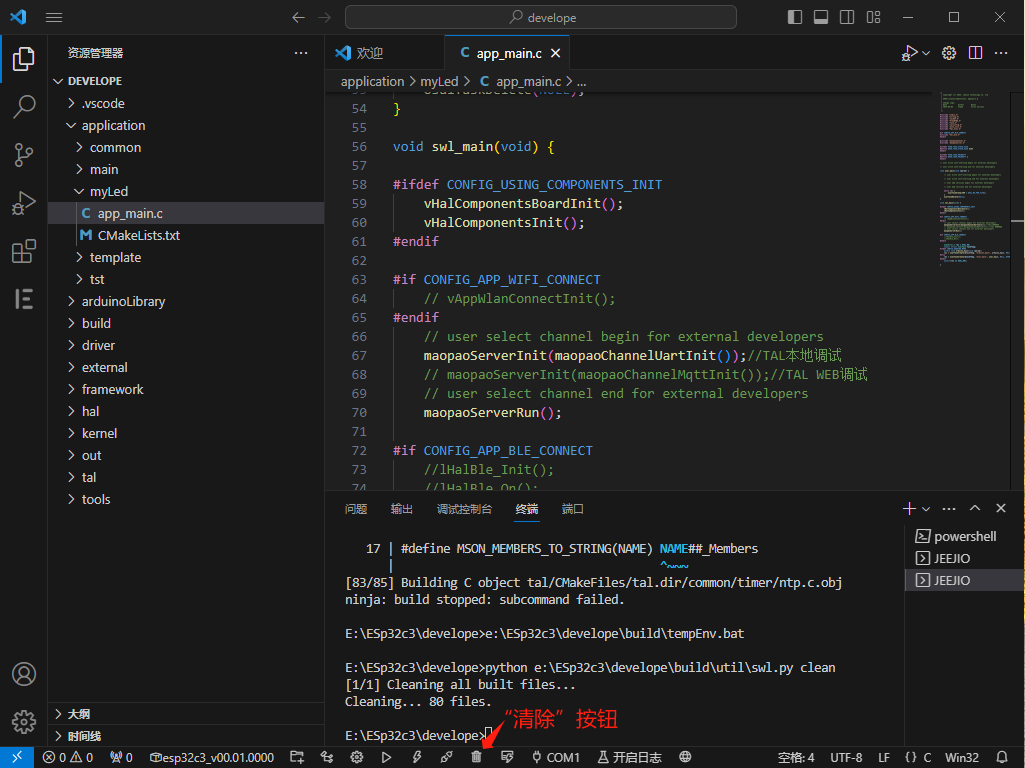


图11-6 开始清除

（2）IDE将清除编译记录、编译生成的文件，清除完毕后界面如下。

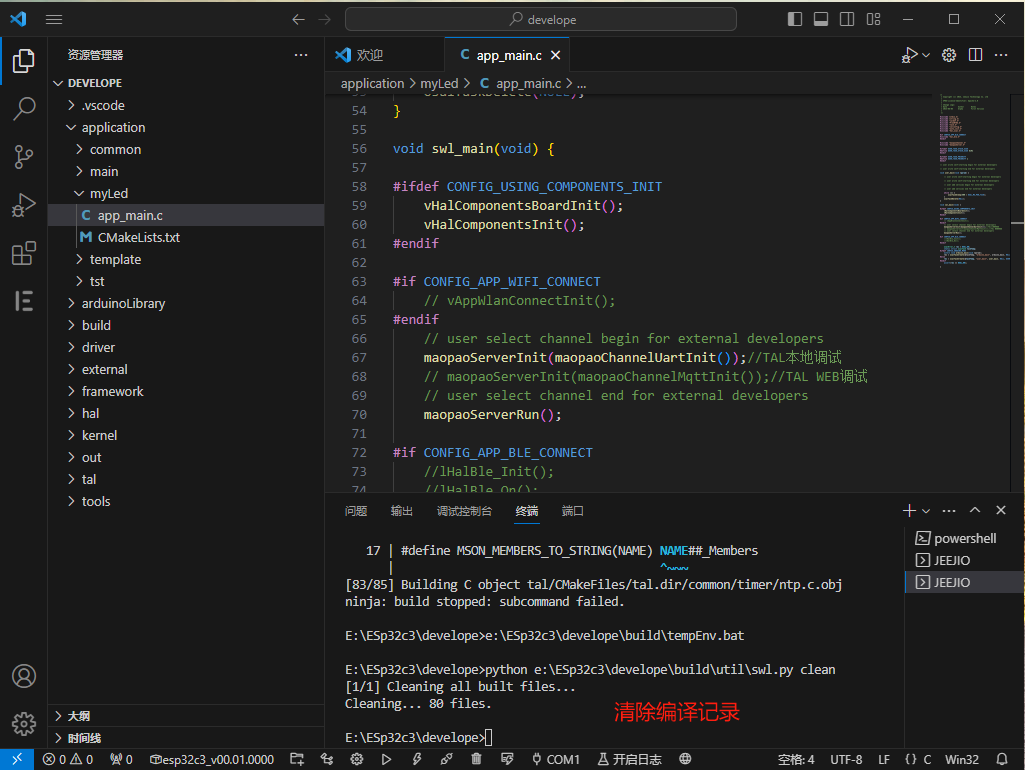


图11-7 清除完成

## 清除、编译、烧录

点击IDE底部工具栏中“清除、编译、烧录”按钮将一键完成清除、编译、烧录工作。

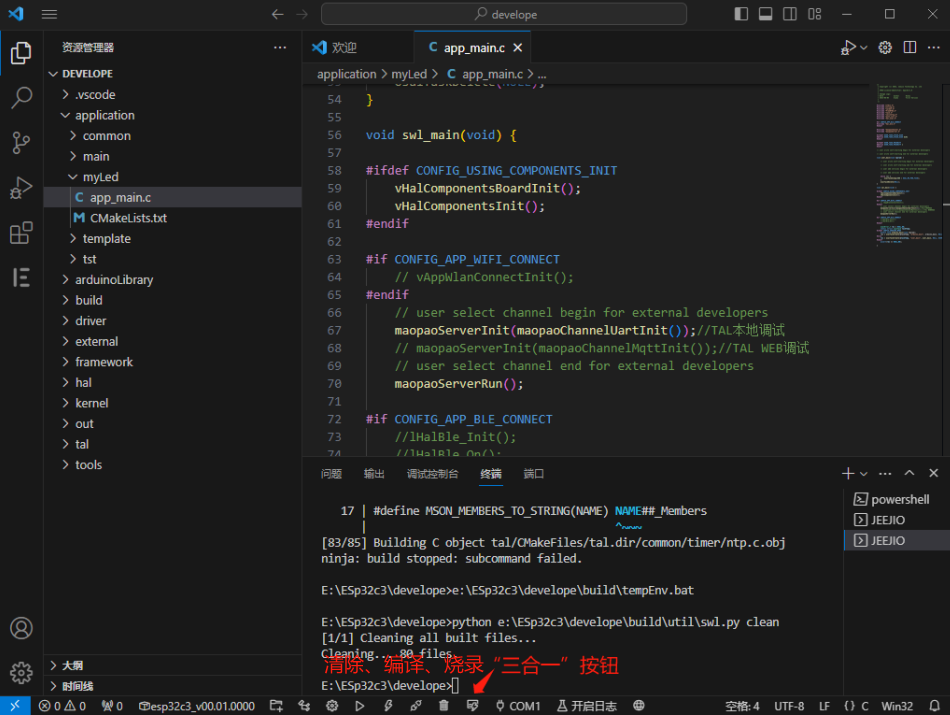


图11-8 清除、编译、烧录按钮

# 烧录密钥

1、密钥文件包含瘦西湖OS设备的身份信息和网络鉴权信息。瘦西湖设备在烧录密钥文件后才能注册到冒泡云平台。密钥文件须向中科物栖申请。密钥文件以.jk作为文件后缀，比如ZKWQAABA000A.jk。

2、选择设备对应的串口号。点击IDE底部工具栏中“切换串口”按钮，在IDE顶部“选择串口”下拉菜单中选择设备对应的串口号。

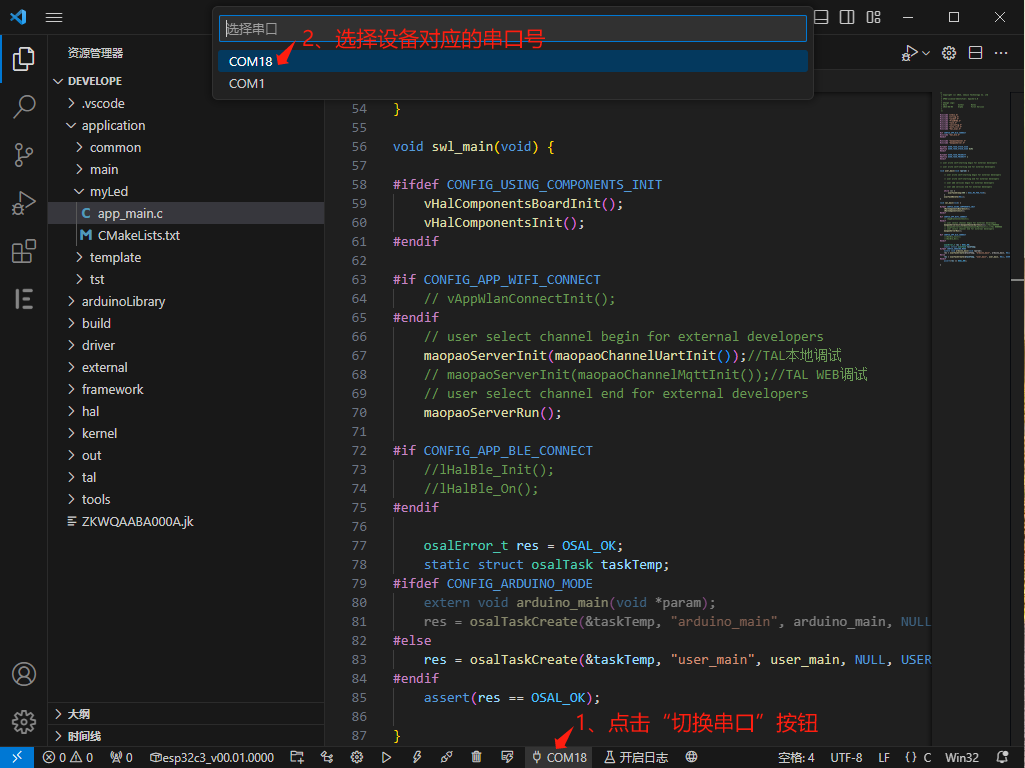


图12-1 切换到设备对应的串口

3、选择密钥并烧录。点击IDE底部工具栏中“烧录密钥”按钮，在IDE弹出的窗口中“选择密钥文件”，选中密钥文件后，IDE将自动开始烧录密钥到设备。

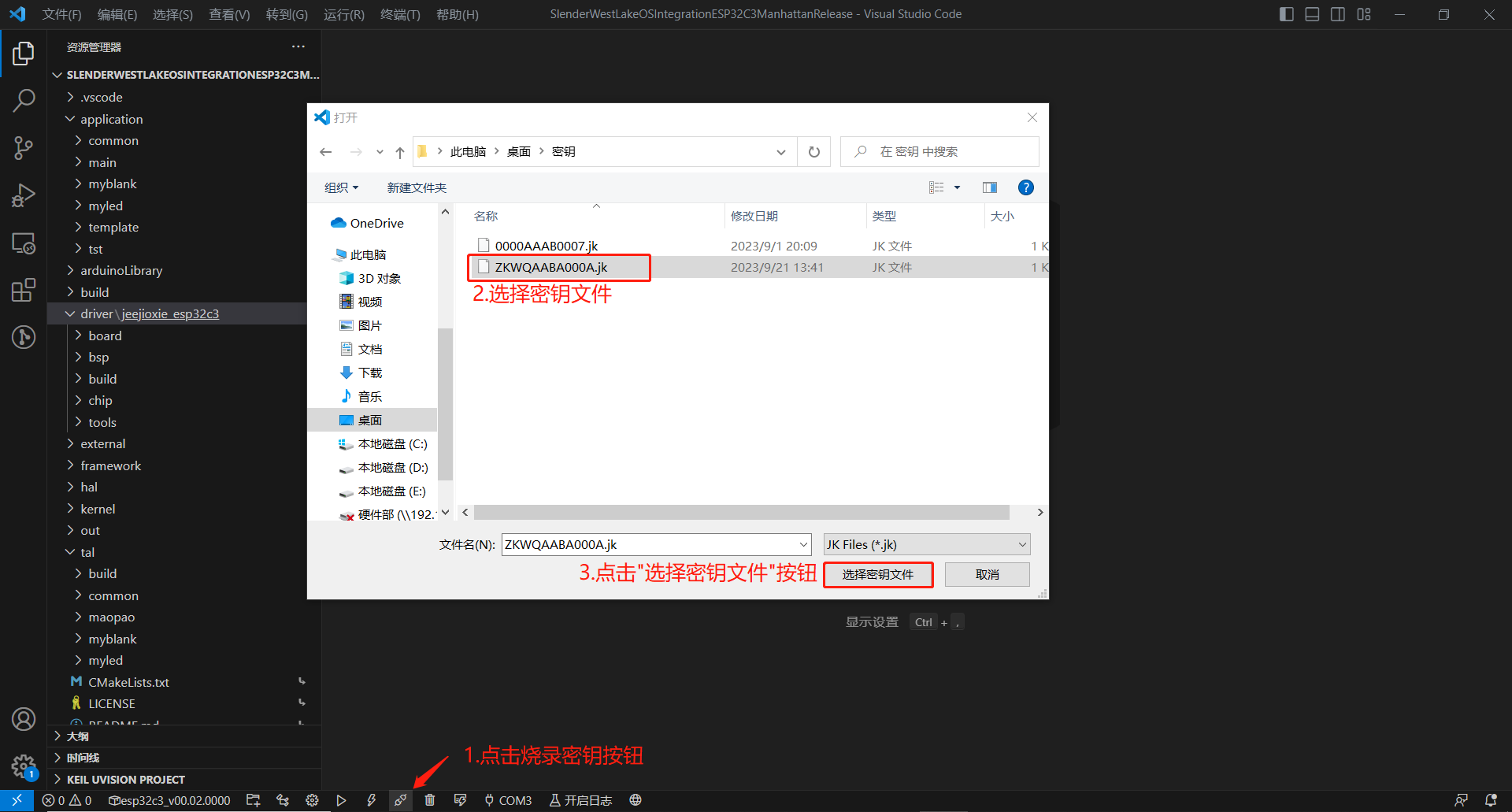


图12-2 开始烧录密钥

4、在密钥烧录过程中可以在IDE“终端”窗口查看日志信息。

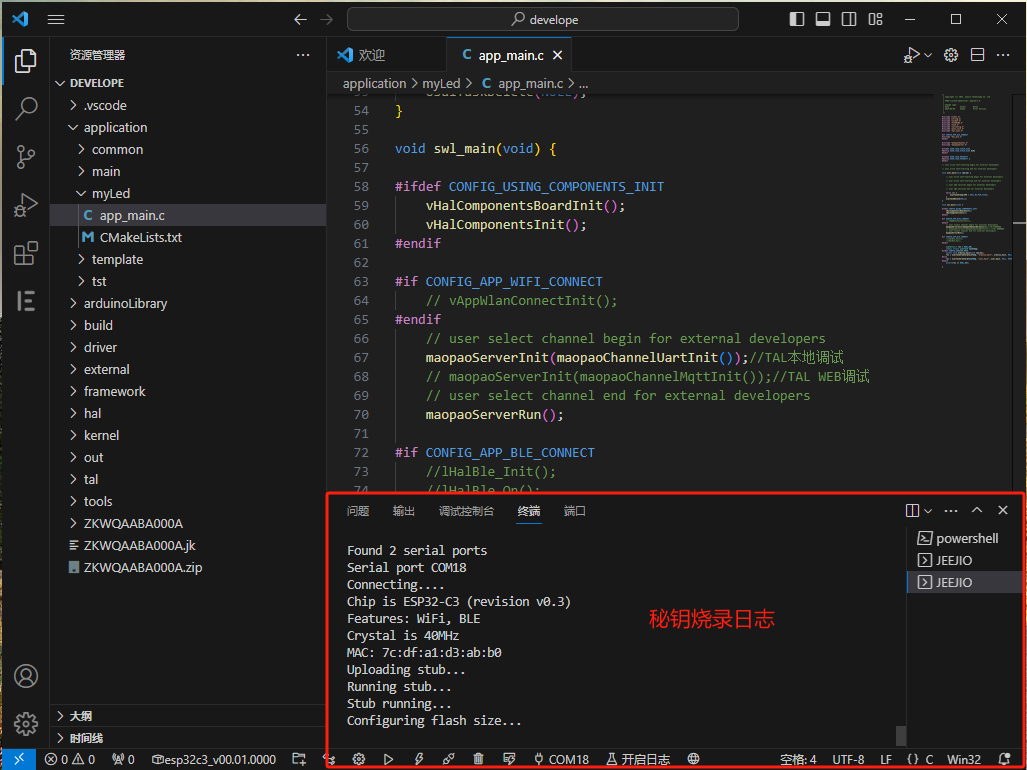


图12-3 烧录密钥日志

注：使用串口通道调试设备时候不需要烧录密钥。在设备联网到冒泡云平台时候需要烧录密钥。

# 开启日志

1、选择设备对应的串口号。点击IDE底部工具栏中“切换串口”按钮，在IDE顶部“选择串口”下拉菜单中选择设备对应的串口号。

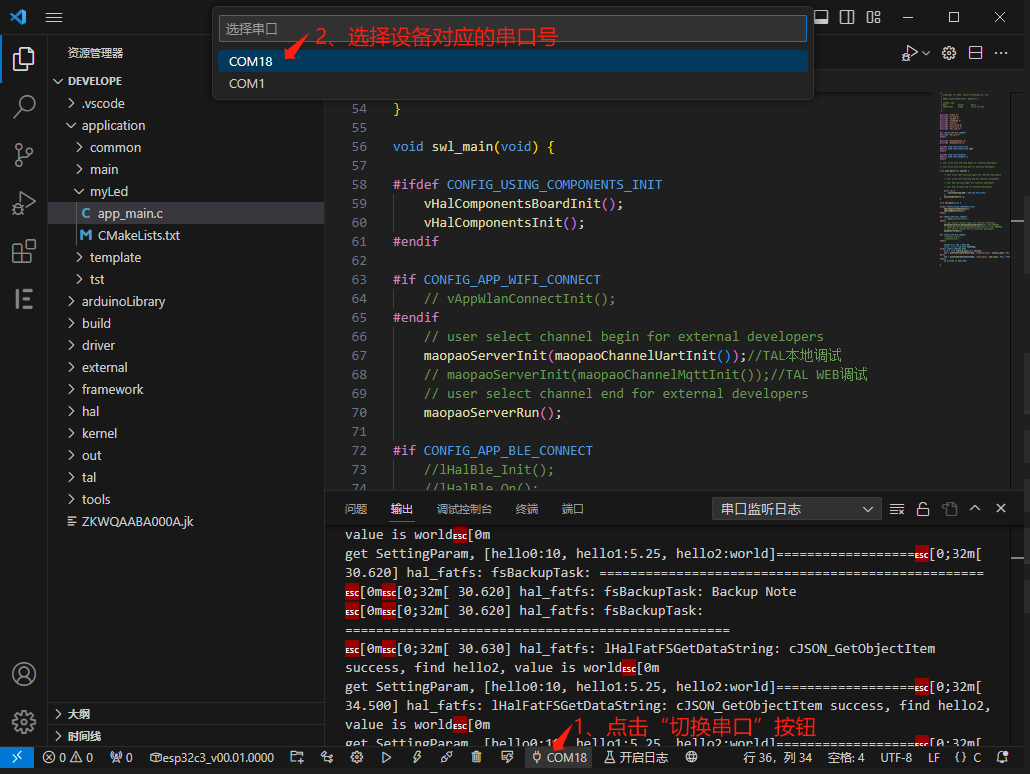


图13-1 切换设备对应的串口

选择成功后，将弹出成功提醒“您选择了：COM”，同时“切换串口”按钮将显示切换后的串口号。

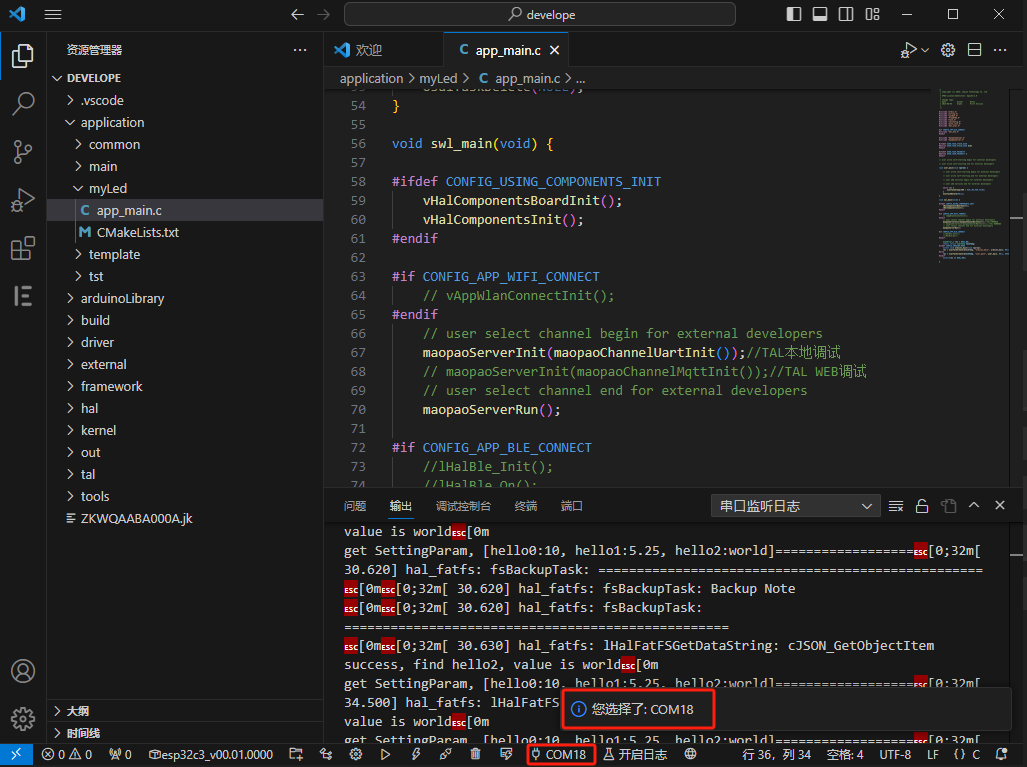


图13-2 切换串口成功

2、点击“开启日志”按钮，IDE将显示设备串口日志信息。

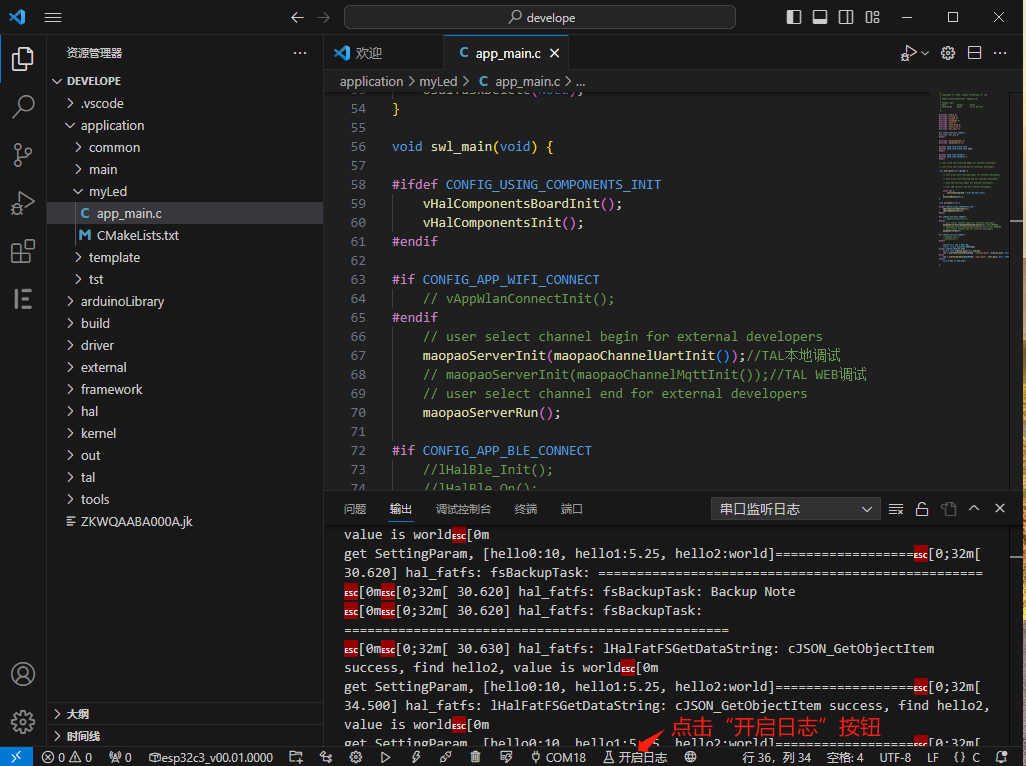


图13-3 开启日志

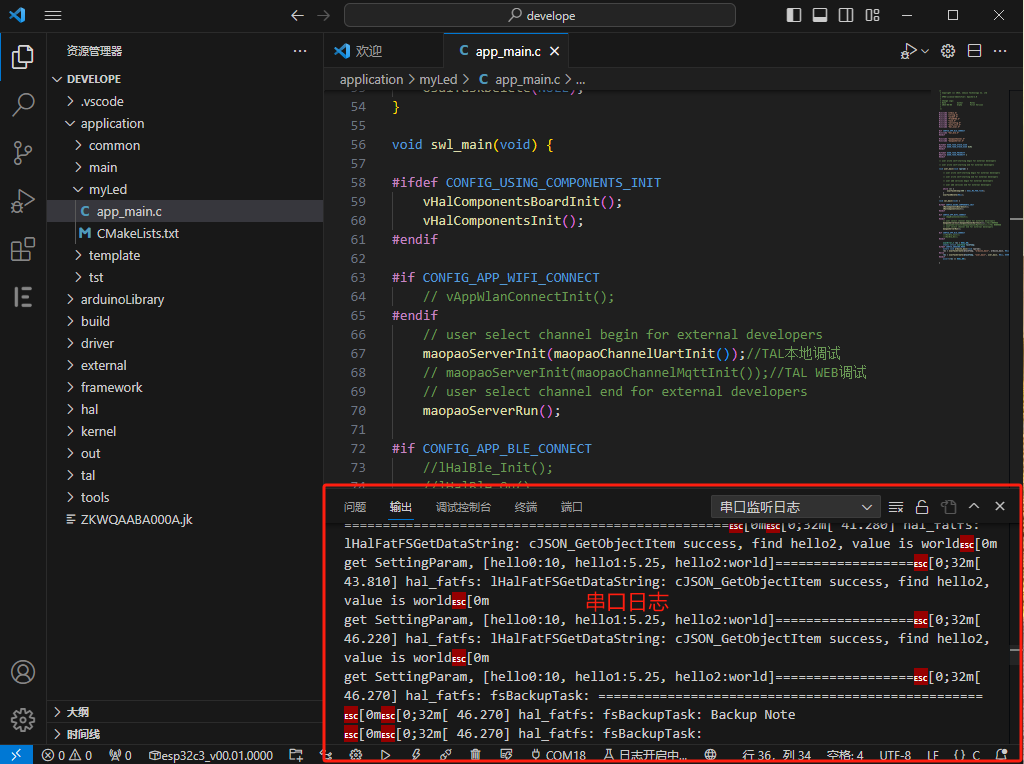


图13-4 串口日志界面

3、再次点击“串口日志”按钮将关闭串口日志功能，IDE不再显示设备串口日志。

# TAL调试

## TAL本地调试

1、点击“TAL调试”按钮，在IDE窗口顶部选择“TAL本地调试”。

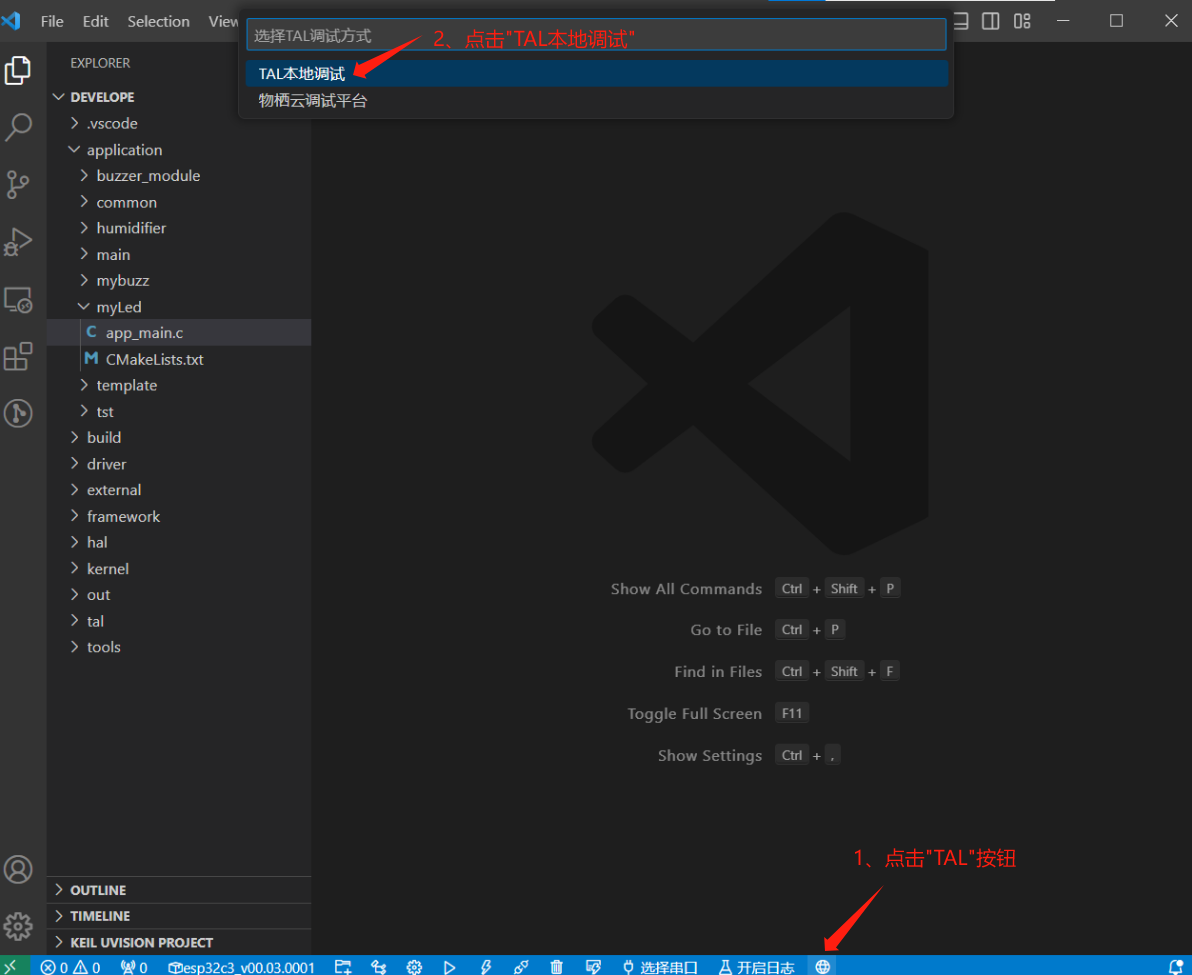


图14-1 选择TAL本地调试

2、在“TAL本地调试”界面配置串口信息，配置完毕后点击“打开串口”。

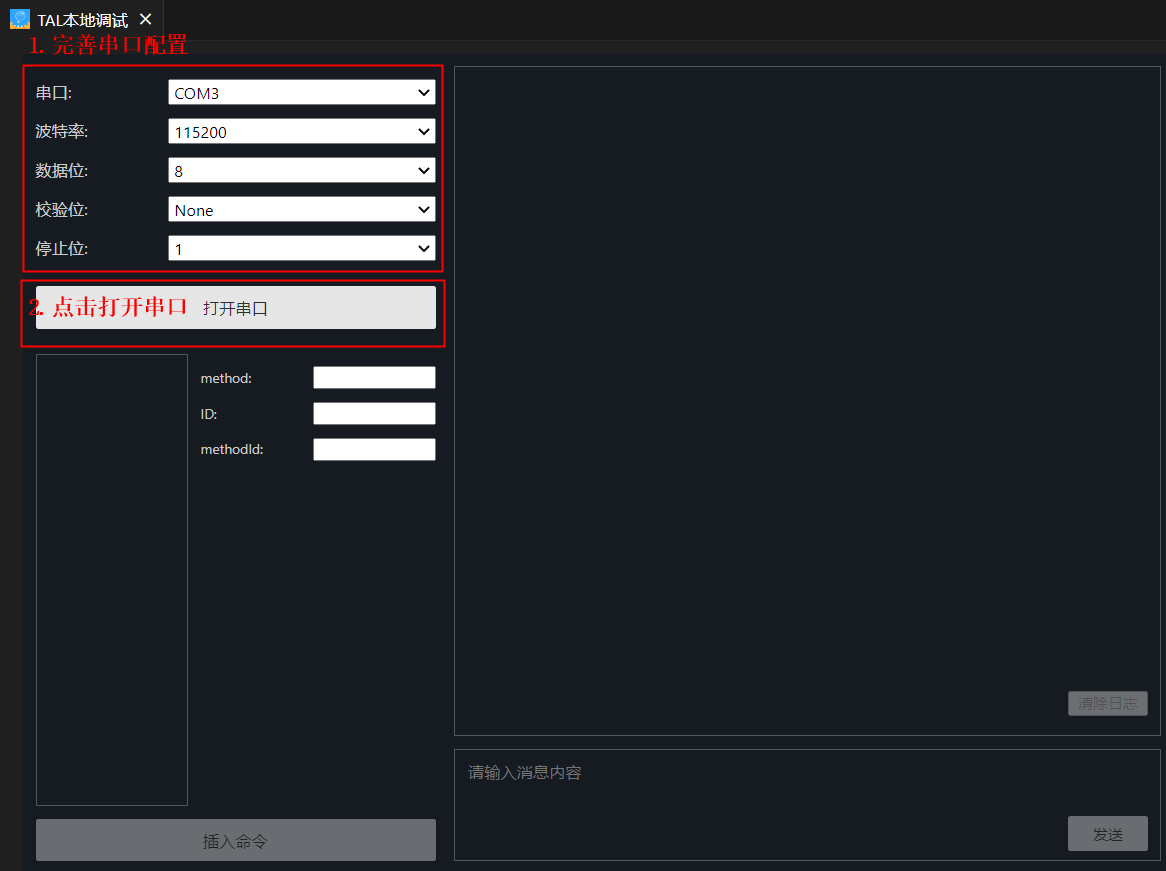


图14-2 TAL本地调试 - 配置串口

3、打开串口后，IDE工具将自动向台灯发送API查询指令。台灯收到指令后会发送支持的API功能到IDE并在窗口显示。

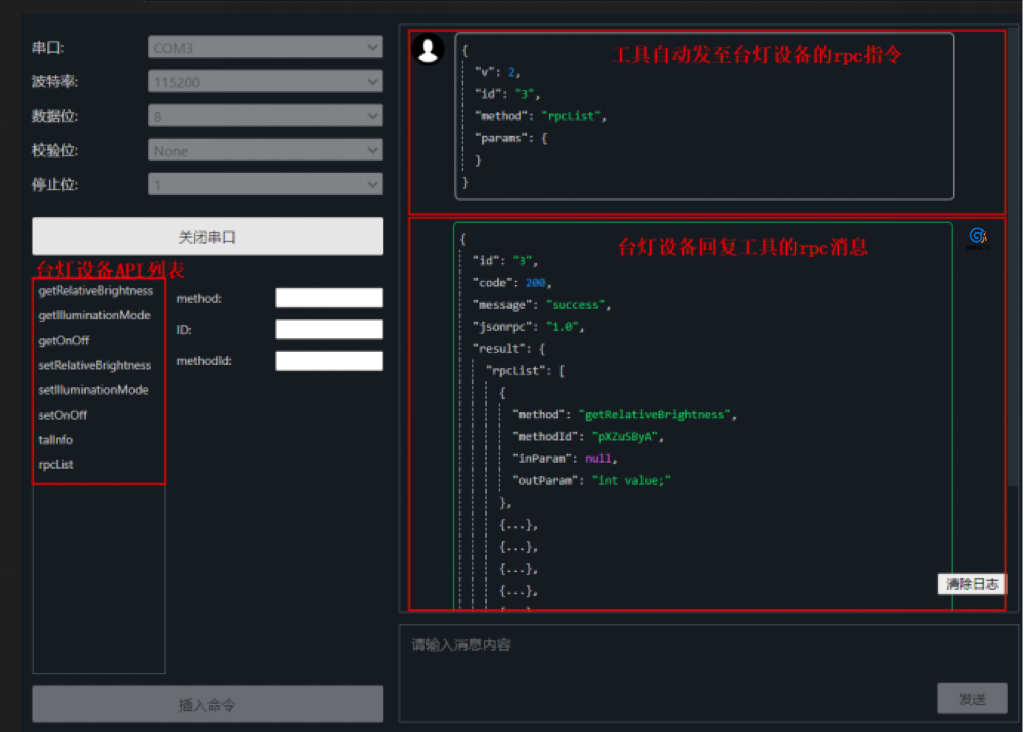


图14-3 TAL本地调试 - 获取RPC API列表

4、现在可以通过TAL本地调试工具来调试台灯。

（1）获取台灯开关状态。

1. 在工具左边的API功能列表中找到getOnOff并选中，此时在“method”文本框将显示“getOnOff”。
2. 点击“插入命令”，工具将生成RPC命令；
3. 点击“发送”，工具将把getOnOff封装成RPC协议包发送到台灯。
4. 台灯在读取到开关状态后将结果信息返回给IDE。

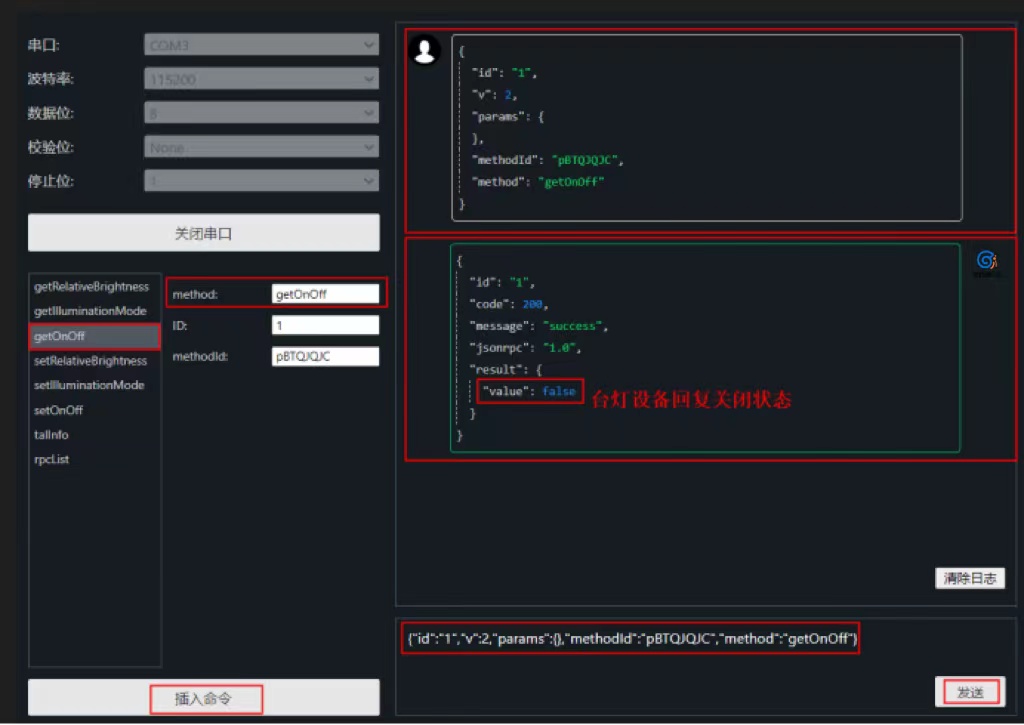


图14-4 TAL本地调试 - 获取台灯开关状态

（2）设置台灯开关状态。

1. 在工具左边的API功能列表中找到setOnOff并选中。
2. 此时在“method”文本框将显示“setOnOff”。
3. 在“bool value”文本框输入“true”。
4. 点击“插入命令”，工具将自动生成命令；
5. 点击“发送”，工具将把setOnOff和true封装成RPC协议包发送到台灯。
6. 台灯在设置台灯开关状态为true后，将执行结果信息返回给IDE。

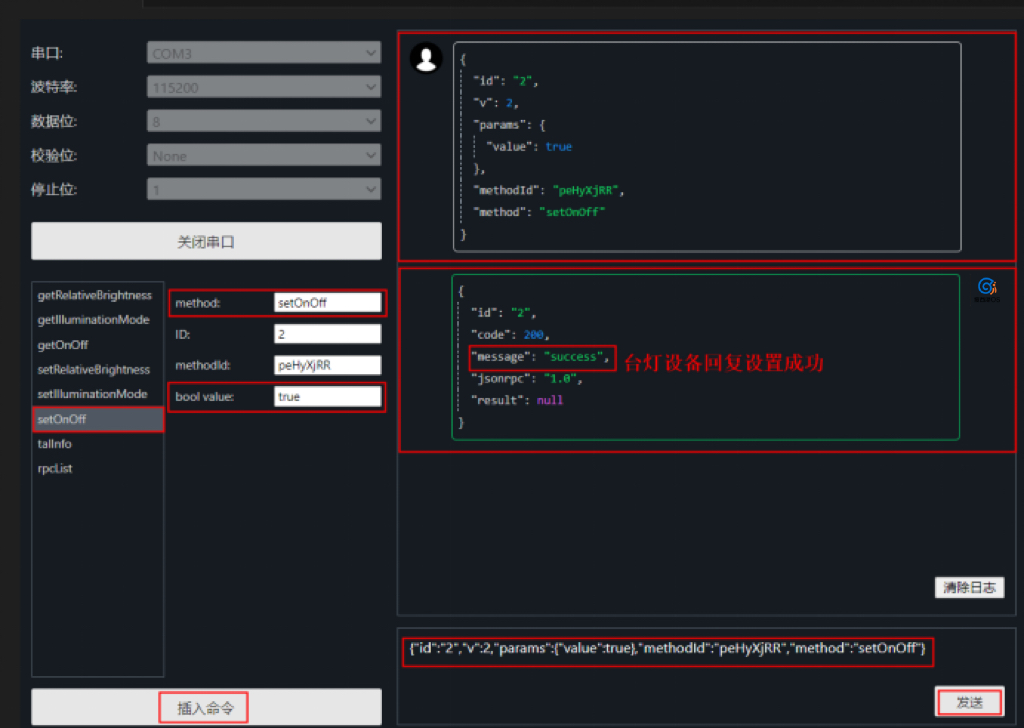


图14-5 TAL本地调试 - 设置台灯开关状态

## TAL物栖云调试(功能暂未上线)

1、点击“TAL调试”按钮，选择“TAL 物栖云调试”。

2、登录web平台-->选择设备对应的应用-->设备上电-->完成配网-->确定设备已上线，即可进行MQTT通道调试。

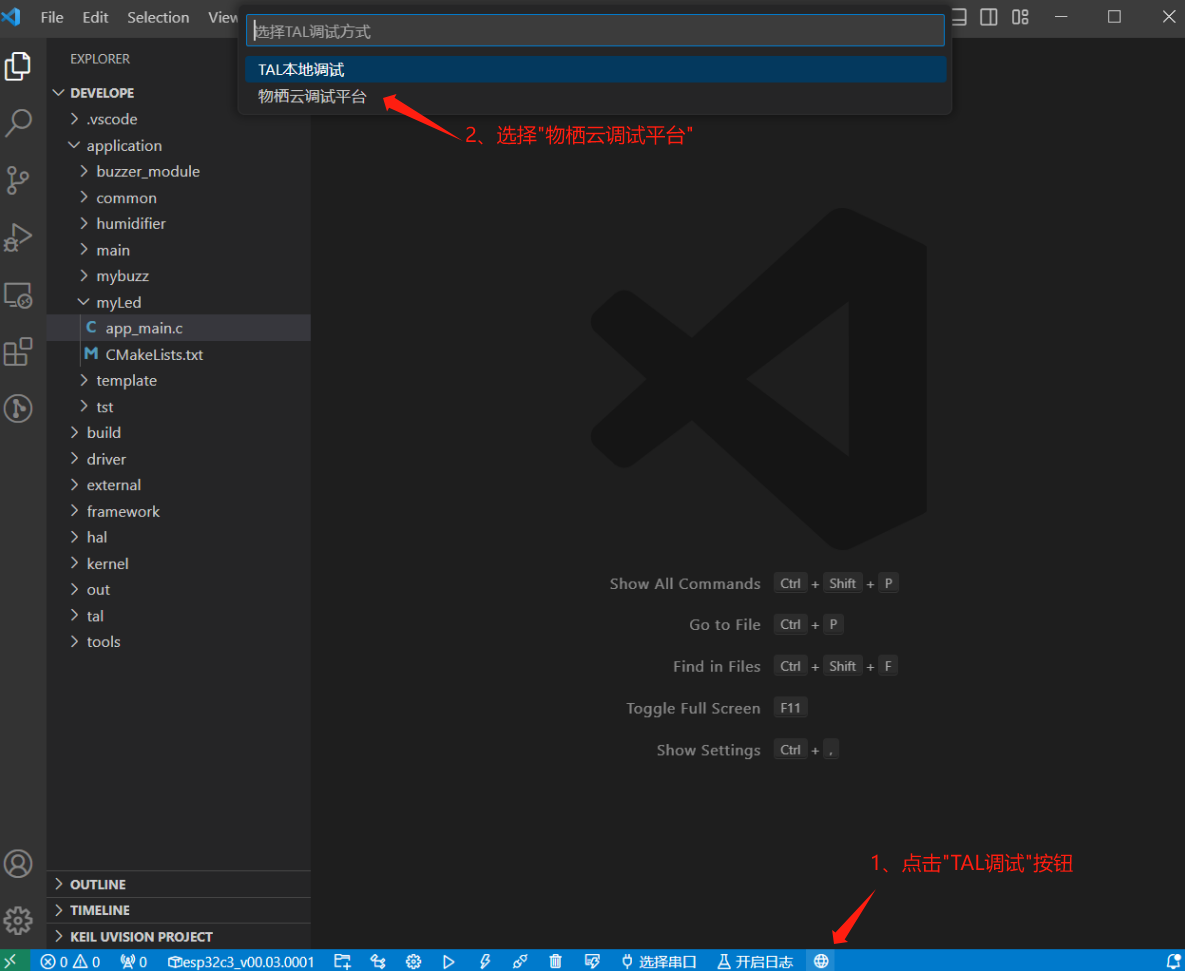


图14-5 选择物栖云调试

# 插件快捷键介绍

表15-1 快捷键介绍

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 快捷键 |
| 1 | 选择开发环境 | Ctrl+Alt+E |
| 2 | 新建项目 | Ctrl+Alt+P |
| 3 | 编辑TAL | Ctrl+Alt+T |
| 4 | kconfig | Ctrl+Alt+G |
| 5 | 编译 | Ctrl+Alt+C |
| 6 | 烧录固件 | Ctrl+Alt+F |
| 7 | 三合一 | Ctrl+Alt+M |
| 8 | 烧录密钥 | Ctrl+Alt+K |
| 9 | 清除 | Ctrl+Alt+U |
| 10 | 开启日志 | Ctrl+Alt+D |
| 11 | tal本地调试 | Ctrl+Alt+L |
| 12 | tal 物栖云调试 | Ctrl+Alt+R |