

AURIX Development Studio 环境使用说明

目录

目录.....	1
1. 使用 AURIX Development Studio 导入工程.....	3
1.1. 导入工程.....	3
1.2. 编译工程并进行调试.....	5
2. 编译、调试与下载.....	8
2.1. 编译工程.....	8
2.2. 仅下载程序.....	8
2.3. 调试工程.....	8
2.4. 单步调试与断点功能.....	9
2.5. 变量查看功能	10
2.6. 内存查看功能	11
2.7. printf 输出功能.....	12
2.8. 新建窗口面板	13
3. 工作空间管理.....	15
3.1. 切换工作空间	15
3.2. 切换工程.....	15
3.3. 打开或关闭工程	16
3.4. 删除工程.....	17

3.5. 重命名工程.....	18
3.6. 添加、新建文件	18
3.6.1. 文件管理器添加文件	18
3.6.2. ADS 新建文件.....	19
3.6.3. 直接添加文件	21
3.7. 包含路径.....	22
4. 环境设置	23
4.1. 字体设置.....	23
4.2. 拼写报错屏蔽设置	24
5. 常见问题及解决方案	25
5.1. 硬件未连接报错	25
5.2. 宏定义未生效问题	26
5.3. 每次调试时都会重新编译的问题	26
5.4. 重命名工程后编译、调试报错的问题.....	26
6. ADS 下载方式	27
6.1. 百度云盘下载	27
6.2. 官网下载.....	27
7. 文档版本	28

1.使用 AURIX Development Studio 导入工程

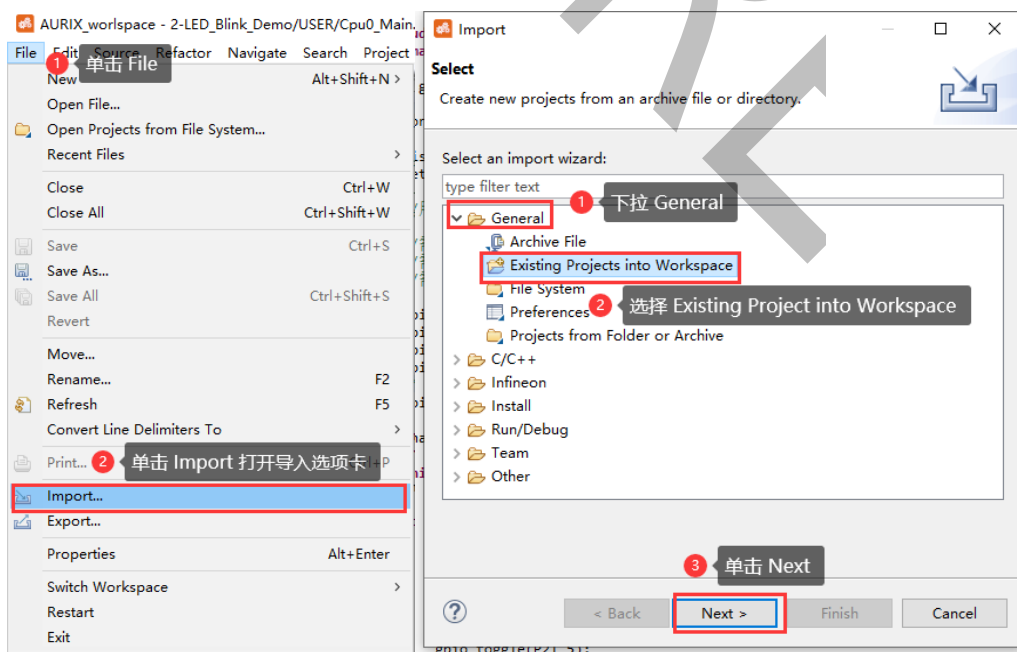
安装完成 AURIX Development Studio（在后面的描述中简称为 ADS）后并打开，**安装的时候请务必保证 ADS 安装路径没有中文与空格，只允许使用英文、数字与下划线！**初次打开需要选定工作空间路径，**请注意工作空间路径不要包含中文以及空格，只允许使用英文、数字与下划线！**

本章节介绍如何使用 ADS 导入现有工程。本文仅作为参考，仅针对 TC264 芯片，以及逐飞科技 TC264 开源库。**请严格按照手册执行，避免出现额外的问题。**

1.1.导入工程

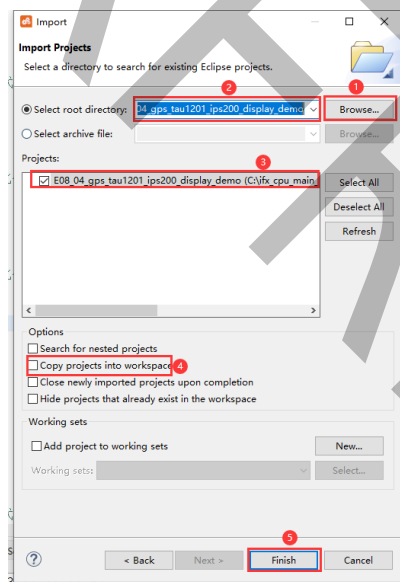
单击 ADS 的菜单栏 File 选项，单击 Import...选项，打开导入选项卡。或者在工作空间选项卡区域单击鼠标右键，找到 Import...选项，打开导入选项卡。

在导入选项卡中打开 General 下拉选项，选择 Existing Project into Workspace 选项，单击 Next 进入下一步。



这里简单的对导入工程的界面进行说明：

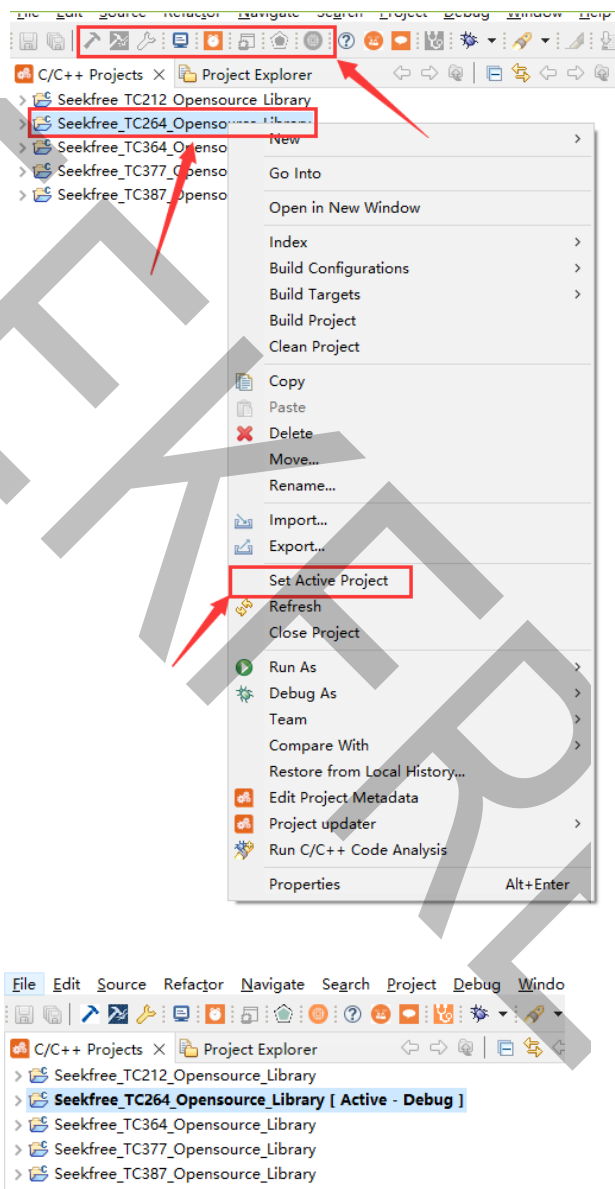
1. 选择工程路径，单击 Browse 按钮可以弹出文件选择窗口，可以找到自己的工程再确定添加。
2. 直接输入工程路径，相对于 Browse 选择会更快，一般的操作是在文件夹直接复制工程路径，然后粘贴到此处即可。
3. 选择需要添加的工程，ADS 会扫描选择路径下的所有工程文件，如果只有一个就默认勾选，如果有多个可以由用户自行选择。
4. 是否复制到工作空间，如果 ADS 版本号低于 1.5.2，那么我们建议勾选复制到工作空间，否则会出现一些意料之外的 bug。如果版本号高于 1.5.2，就可以由用户自行决定了，复制到工作空间可以增加代码的安全性，不复制可以节约添加工程的时间。
5. 选择完成后点击 Finish 即可完成添加。



注意：如果 ADS 当前工作空间存在同名工程，那么将无法添加。如果 ADS 工作空间路径内存在同名工程，那么将无法复制工程到工作空间。

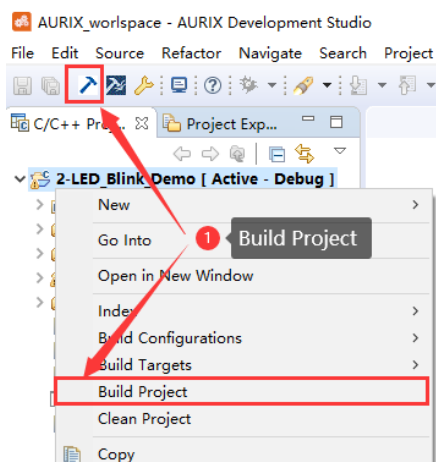
1.2.编译工程并进行调试

初次导入工程会发现部分操作按钮是灰色的，这里我们还需要一个步骤，鼠标右击目标工程，然后选择“Set Active Project”，单击后会发现工程名称加粗，并且上面的按钮也亮起来了。



右键工程并且单击 Clean Project 清理工程！

此时再对工程进行编译操作，推荐右键工程单击 Build Project。

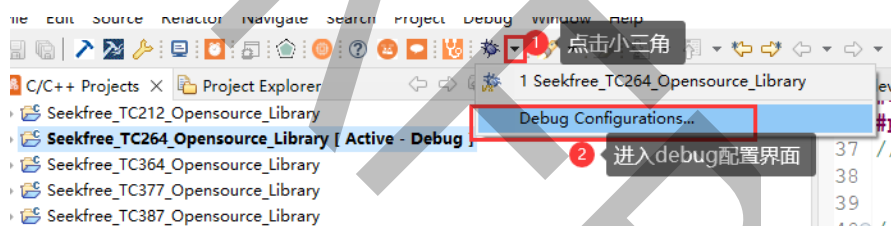


在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

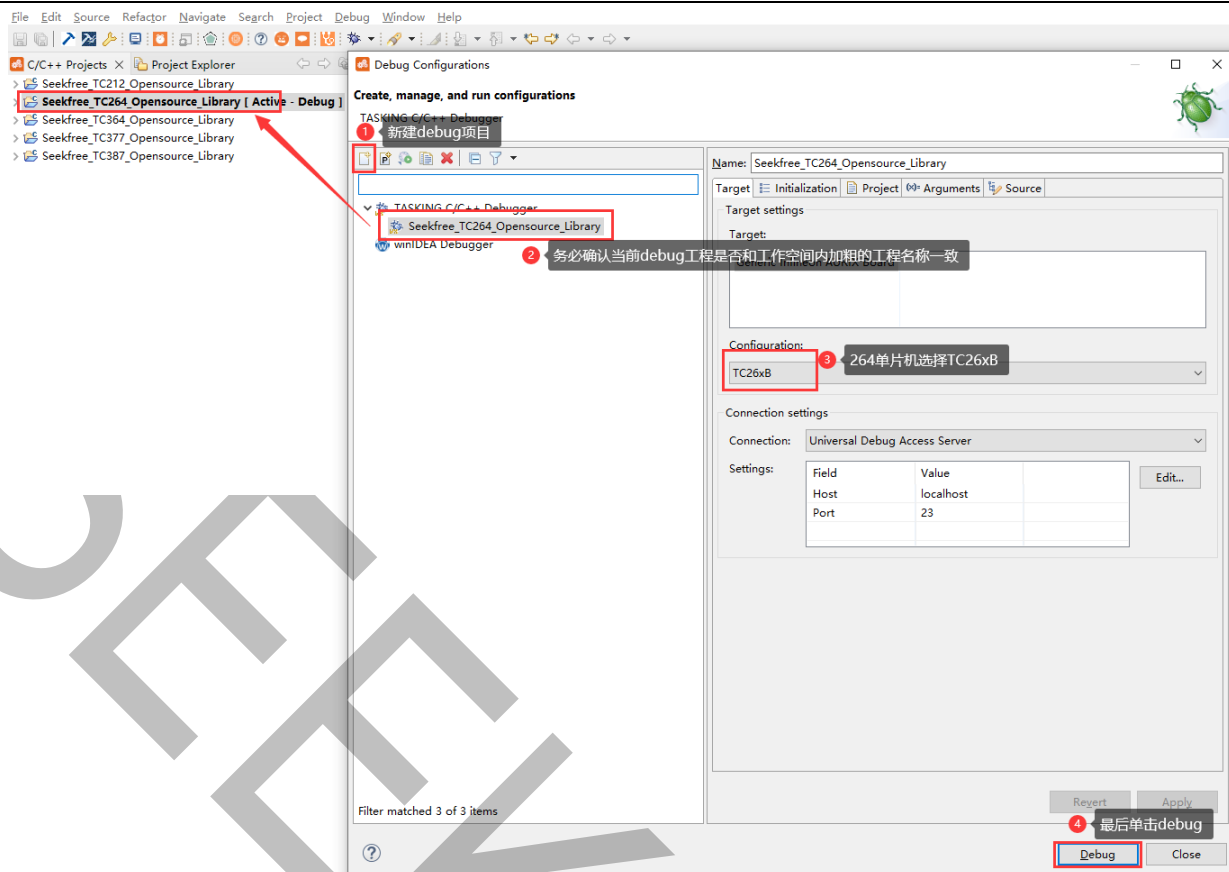
在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

工程编译完成且无报错后，找到上方瓢虫图案旁边的小三角，点击会弹出选择窗口，**选择进入 Debug Configuration 选项卡。**



在 Debug Configuration 选项卡中，如果 TASKING C/C++ Debugger 下没有 debug 项目，那么就需要单击新建 launch，然后选中新建的 launch 进行设置。这里如果已经有可选的 launch，那么只需要确认选项即可。

将 Configuration 选项设置为 TC26xB（这里导入的是我们 TC264 的开源库所以这里设置为 TC26xB）。然后单击 Debug 进行调试。



完成初次调试后，再次调试就不会进行调试的编译过程(工程未作修改的情况下)，直接进入调试界面。推荐使用我们提供的说明中的调试步骤进行调试。

至此导入以及编译调试整个流程完成。

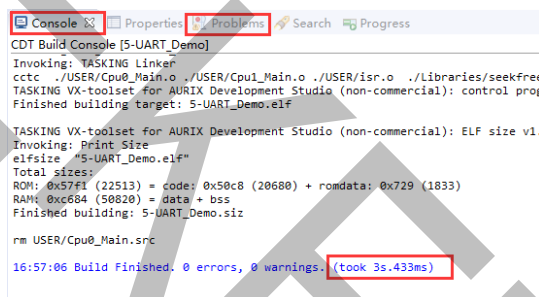
2.编译、调试与下载

2.1.编译工程

编译工程步骤与其他环境差别不大，方式为：

1. 通过右键工程选择 Build Project 选项；
2. 上方选项栏中 Build 或 Rebuild 选项 ；

下方 Console 选项卡会输出编译步骤以及最终耗时，Problems 会输出错误以及警告信息。



```

CDT Build Console [5-UART_Demo]
Invoking: TASKING Linker
cttc ./USER/Cpu0_Main.o ./USER/Cpu1_Main.o ./USER/isr.o ./libraries/seekfree
TASKING VX-toolset for AURIX Development Studio (non-commercial): control prog
Finished building target: 5-UART_Demo.elf

TASKING VX-toolset for AURIX Development Studio (non-commercial): ELF size v1.
Invoking: Print Size
elfsize "5-UART_Demo.elf"
Total sizes:
ROM: 0x57f1 (22513) = code: 0x50c8 (20680) + romdata: 0x729 (1833)
RAM: 0xc684 (50820) = data + bss
Finished building: 5-UART_Demo.siz

rm USER/Cpu0_Main.src

16:57:06 Build Finished. 0 errors, 0 warnings. (took 35.433ms)
    
```

2.2.仅下载程序

1.8.0 版本的 ADS 有仅下载程序到 Flash 的功能，可点击工具栏的 Flash current project

 按键下载当前的工程到单片机。

2.3.调试工程

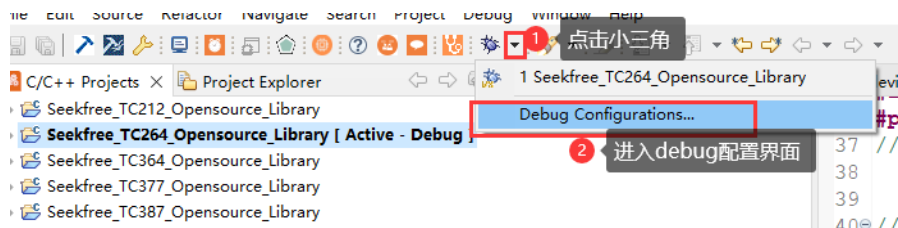
在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

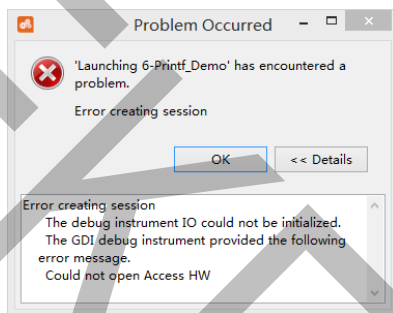
在进行调试工程前请务必确认已经正确连接调试下载器！并且连接好了核心板！

工程编译完成且无报错后，找到上方瓢虫图案旁边的小三角，点击会弹出选择窗口，或者

右键工程选择 Debug As 进入 Debug Configuration 窗口进入调试。



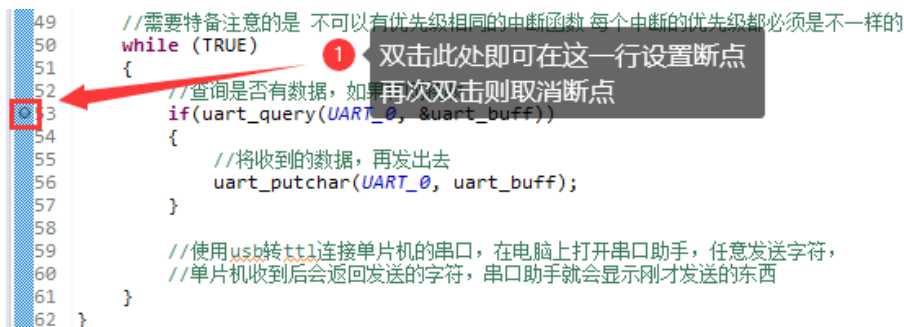
在 Debug Configuration 选项卡中，如果 TASKING C/C++ Debugger 下没有 debug 项目，那么就需要单击新建 launch，然后选中新建的 launch 进行设置。这里如果已经有可选的 launch，那么只需要确认选项即可。将 Configuration 选项设置为 TC26xB（这里导入的是我们 TC264 的开源库所以这里设置为 TC26xB）。然后单击 Debug 进行调试。



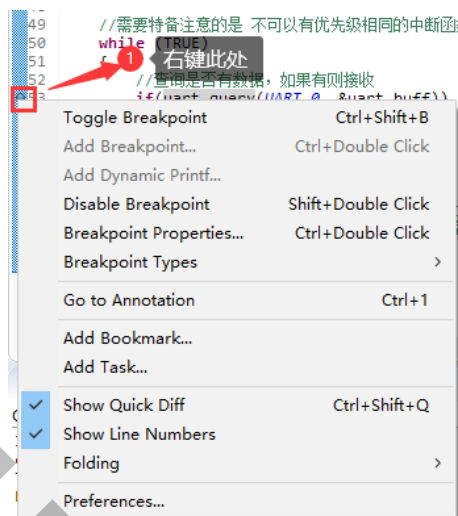
报这个错证明没有检测到调试下载器、没有检测到芯片（核心板）。请检查调试下载器是否正确连接，核心板硬件连接是否正确。调试下载器是否正常被电脑识别，DAS Device Scanner 是否能够扫描到核心。详见 5.1 硬件未连接报错。

2.4.单步调试与断点功能

设置断点可以通过在需要设置断点的行数左侧双击设置断点。

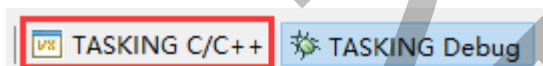


右键该位置可以进行断点的相关操作，例如取消、启用、屏蔽等。



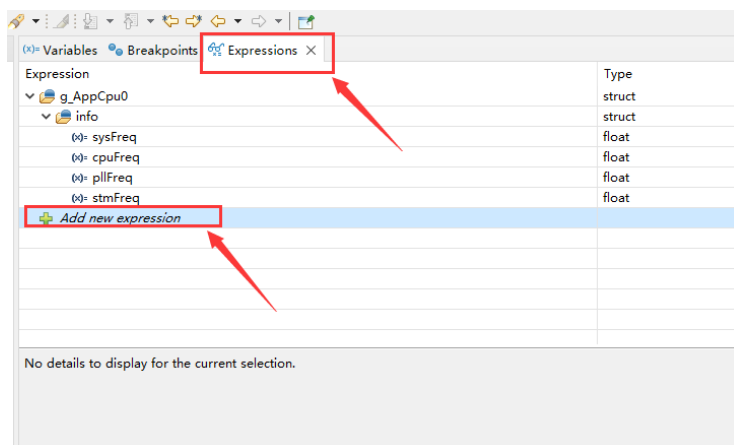
进入 Debug 模式后，点击工具栏 **Resume**<快捷键 F8>选项即可运行代码，运行到断点处即停止，此刻可以使用 **Step Into**<快捷键 F5>进行指令单步<每个操作为一步，可以跳转到所调用函数的定义>调试，或可以使用 **Step Over**<快捷键 F6>进行代码单步<当前函数单行为一步>调试。

可以通过 **Terminate** <快捷键 CTR+F2>结束调试，结束调试后 TASKING 并不会自行跳转回到代码编辑界面，请在窗口上方工具栏右侧，单击 **TASKING C/C++** 切换回代码编辑界面。



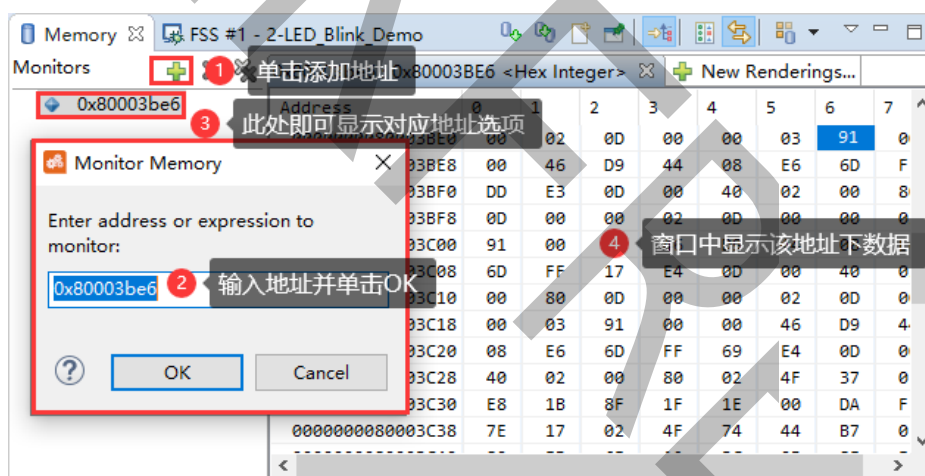
2.5.变量查看功能

在调试界面内，可以通过上方<默认在上方>的 **Expressions** 选项卡查看全局变量。单击 “Add new expression”，然后输入想要查看的变量即可。当然，**Expressions** 并不局限于查看全局变量，当运行到指定函数时也能查看局部变量，也可以查看结构体或者一个计算式子。唯一得不足就是无法实时查看，需要暂停运行后才能查看值。

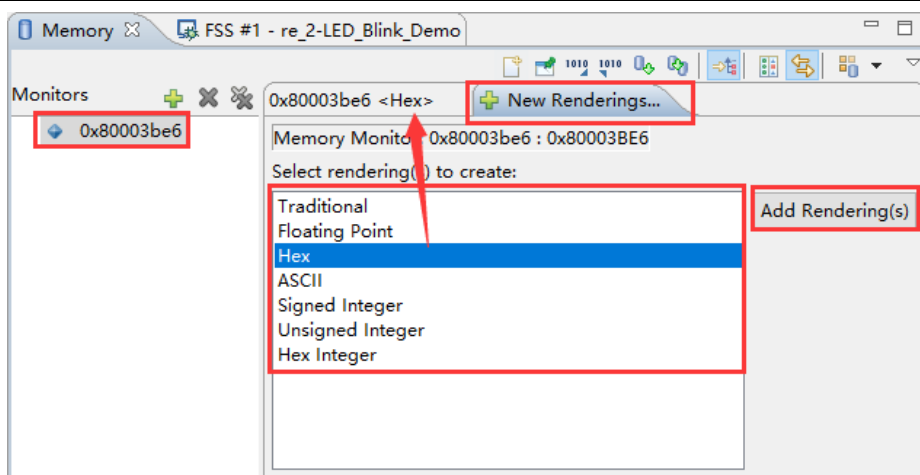


2.6.内存查看功能

ADS 同样提供了内存查看的工具，可以直接查看指定地址下的数据情况，在下方<默认下方>的 Memory 选项卡可以通过 Monitors 功能添加映射地址，在 New Renderings...选项卡可以添加不同的数据格式，具体操作如下图：



在 New Renderings...选项卡可以针对一个地址添加不同的数据格式，具体操作如下图：



2.7.printf 输出功能

从 1.1.6 版本的开源库起，printf 重定向为使用串口输出，不再输出至调试的 FSS 窗口。

开源库版本信息在每个工程的 Libraries->doc 文件夹下的 version.txt 文件中有详细记录，该文件记录当前工程使用的开源库底层的版本。

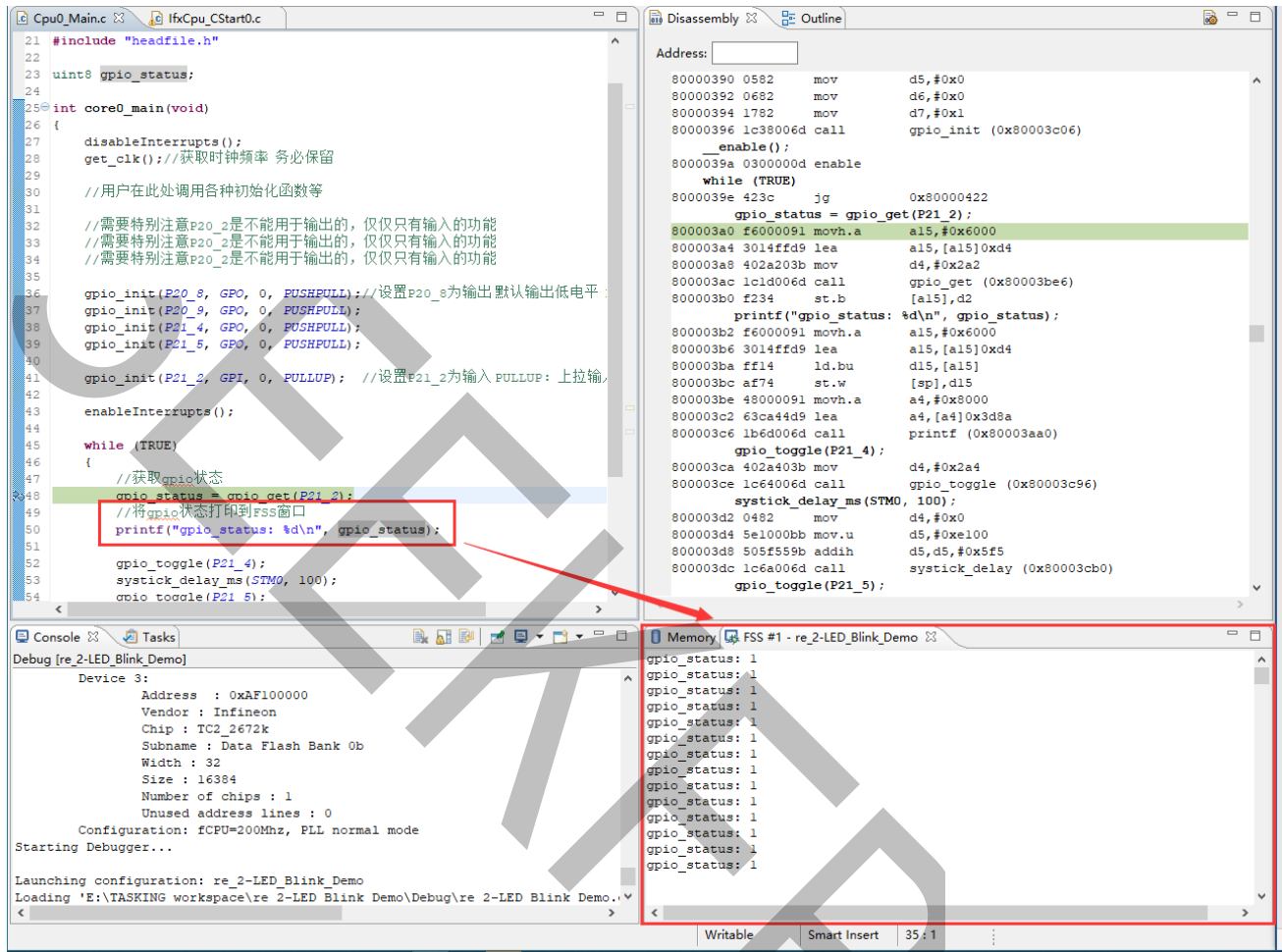
在 1.1.6-2.9.9 之内的开源库版本工程中，printf 会通过 get_clk 函数中初始化的 Debug UART 输出，Debug UART 使用的串口号可以在“TC264_config.h”头文件中看到定义。

```
TC264_config.h
35 //printf串口打印函数所使用的串口、串口波特率、串口引脚宏定义
36 //-----
37 #define DEBUG_UART          UART_0
38 #define DEBUG_UART_BAUD     115200
39 #define DEBUG_UART_TX_PIN   UART0_TX_P14_0
40 #define DEBUG_UART_RX_PIN   UART0_RX_P14_1
```

在 3.0.0 以上的开源库版本工程中，printf 会通过 debug_init 函数中初始化的 Debug UART 输出，Debug UART 使用的串口号可以在“zf_common_debug.h”头文件中看到定义。

```
#define DEBUG_UART_INDEX      (UART_0)           // 指定 debug uart 所使用的的串口
#define DEBUG_UART_BAUDRATE   (115200)          // 指定 debug uart 所使用的的串口波特率
#define DEBUG_UART_TX_PIN     (UART0_TX_P14_0)   // 指定 debug uart 所使用的的串口引脚
#define DEBUG_UART_RX_PIN     (UART0_RX_P14_1)   // 指定 debug uart 所使用的的串口引脚
#define DEBUG_UART_USE_INTERRUPT (0)             // 是否启用 debug uart 接收中断
```

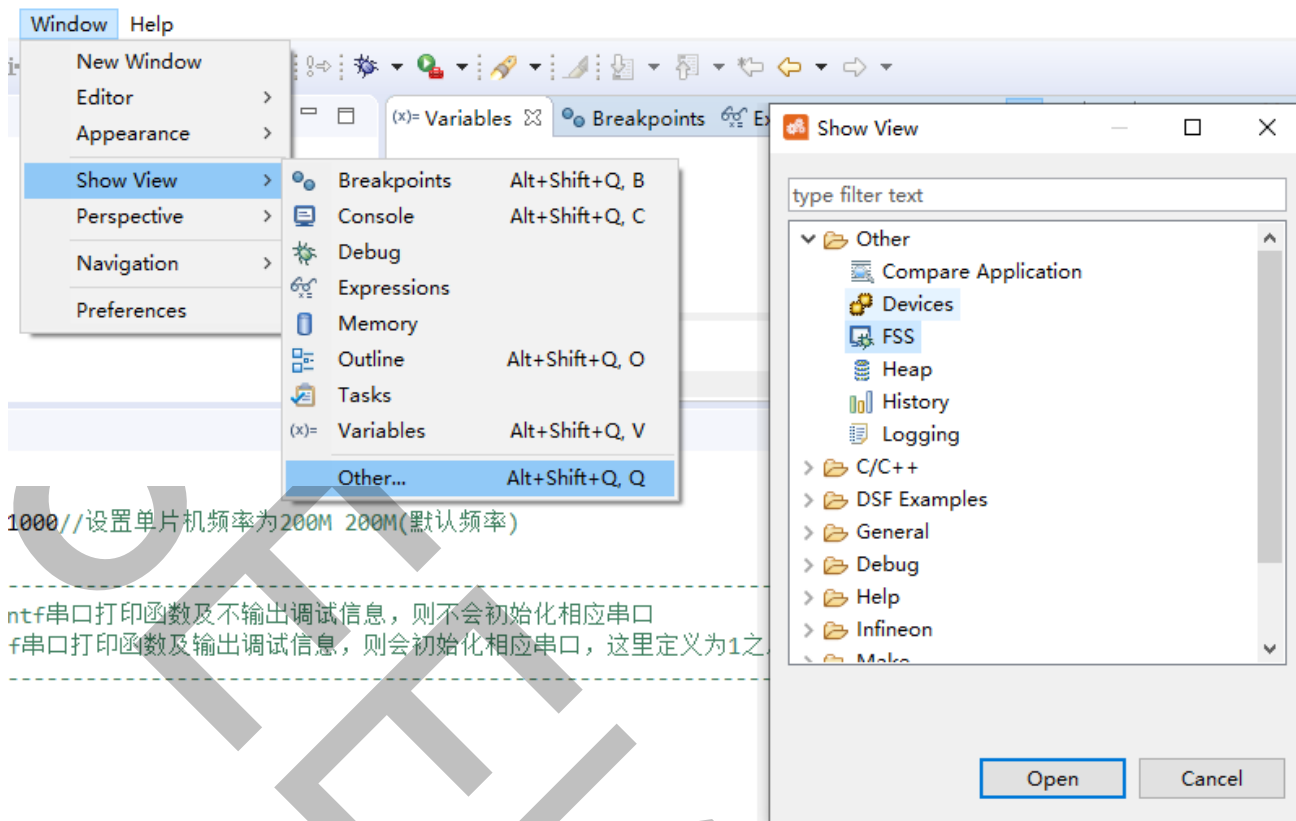
在 1.1.6 以下的开源库版本工程中，通过 ADS 本身提供的 log 输出的工具，可以使用 printf 函数将 log 信息输出到在下方<默认下方>的 FSS 选项卡。



为了方便调试，可以在 debug 版本代码中添加 log 输出，相对会更加方便调试，更加友好，而在 release 版本代码中可以将 log 输出删除以提高运行效率。如果找不到 FSS 窗口或者不小心将 FFS 串口关闭了，请看下一章节的添加面板的介绍。

2.8.新建窗口面板

在调试时可能会需要用到某些窗口，如果之前关闭了或者没有需要的信息窗口，可以通过新建调试窗口面板添加，这里以 FSS 窗口为例：



1000//设置单片机频率为200M 200M(默认频率)

ntf串口打印函数及不输出调试信息，则不会初始化相应串口
f串口打印函数及输出调试信息，则会初始化相应串口，这里定义为1之

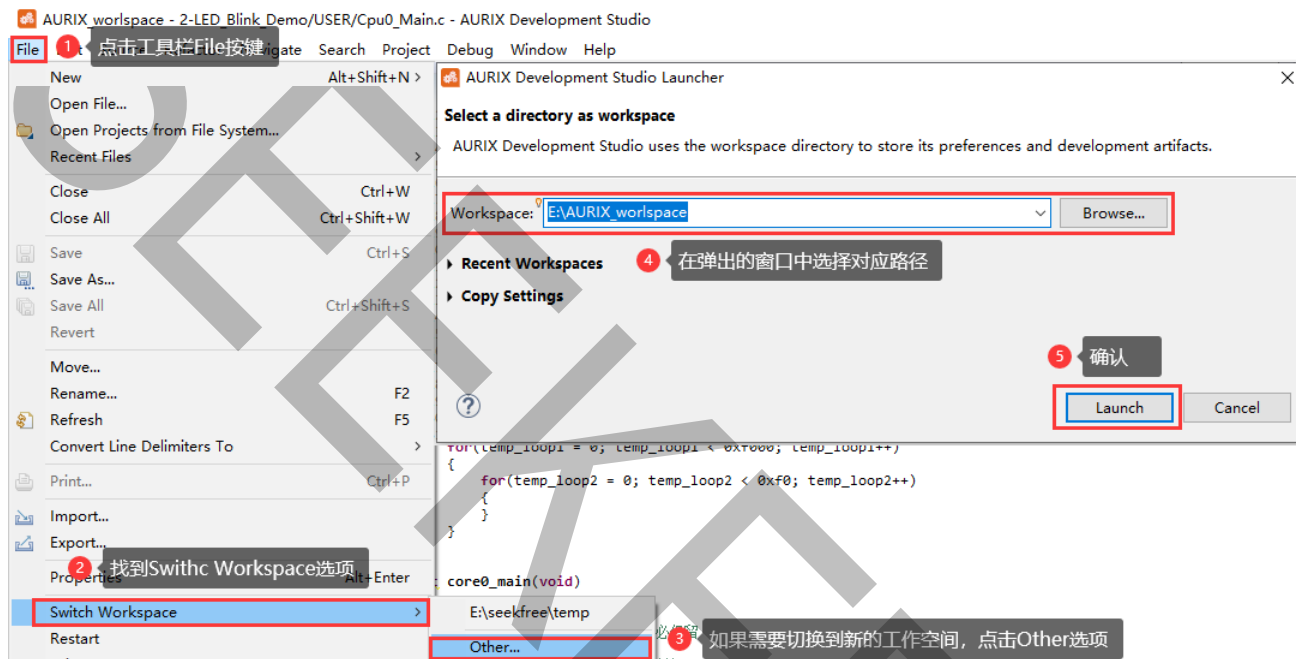
单击菜单栏 Window 选项，在 Show View 下拉菜单中选择 Other...选项，选中 FSS 并单击 Open 即可新建一个 FSS 窗口。

如果需要其他窗口，以同样的方式找到对应的窗口 Open 即可。

3. 工作空间管理

3.1. 切换工作空间

如果需要切换工作空间，请务必**注意工作空间路径不要包含中文以及空格！**



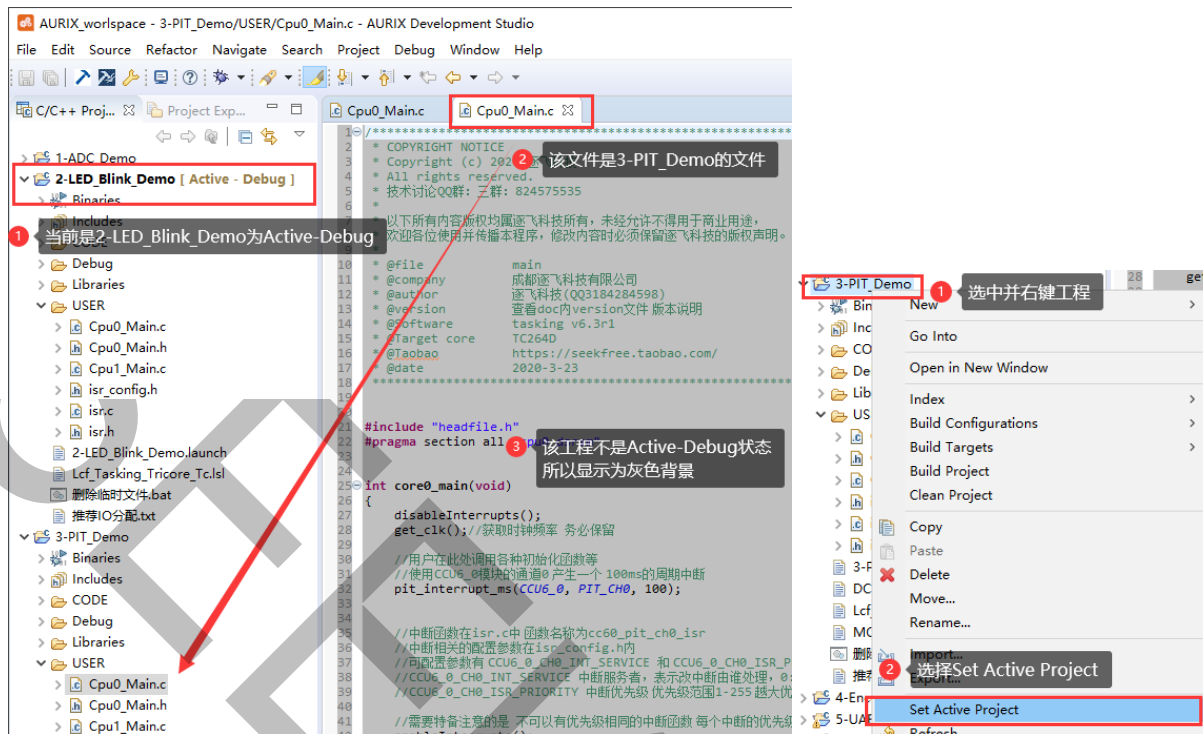
按照上述步骤，即可切换工作空间路径，确认后 ADS 会重新启动一次并切换到新的工作空间路径。

请不要随意改动当前工作空间路径以及路径中的文件夹名称！如果当前工作空间路径包含中文，也不要直接对工作空间路径中中文部分进行直接修改，请按照上述步骤更改工作空间路径！

3.2. 切换工程

当你的工作空间中留存有许多的工程时，需要注意打开的源文件是否是 **Active Debug** 状态，Active Debug 状态下的工程会在 C/C++ Projects 选项卡中以粗体标示，表示当前该

工程处于 Active Debug 状态，编译、调试等操作会针对于该工程进行。



如左图所示，我当前打开了例程<3-PIT_Demo>的源文件，但实际 Active Project 为<2-LED_Blink_Demo>，所以此时我点击编译时，会编译例程<2-LED_Blink_Demo>而不是<3-PIT_Demo>，请务必注意这一点。此时，我们需要切换<3-PIT_Demo>为 Active Debug 状态，在需要调试的工程上右键选择 Set Active Project，如右图所示。

切换后，<3-PIT_Demo>变为 Active Debug 状态，此时再进行编译、调试等操作时，就会以<3-PIT_Demo>工程为基础进行。

3.3.打开或关闭工程

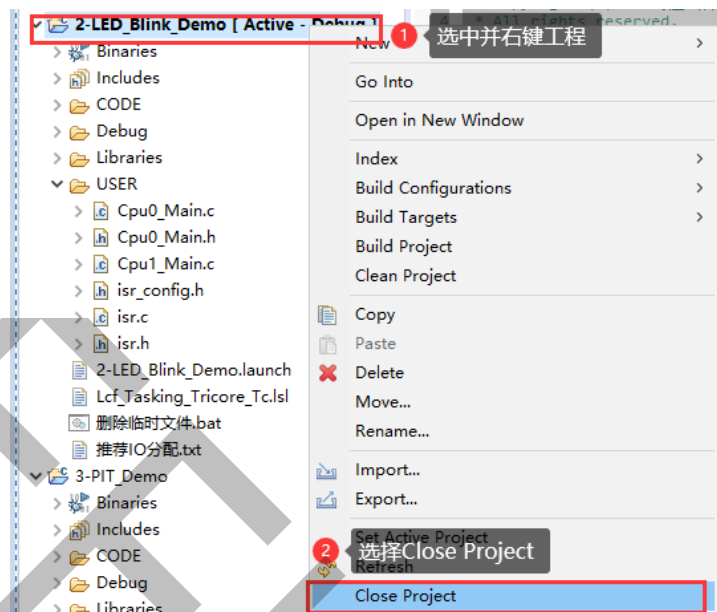
工程处于打开状态时，图标显示为带折叠箭头的打开的文件夹，工程处于关闭状态时，图标显示为不带折叠箭头的关闭的文件夹。



工作空间内工程处于打开状态时，使用 Build ALL 操作时会对所有打开工程进行编译操

作，请务必注意。

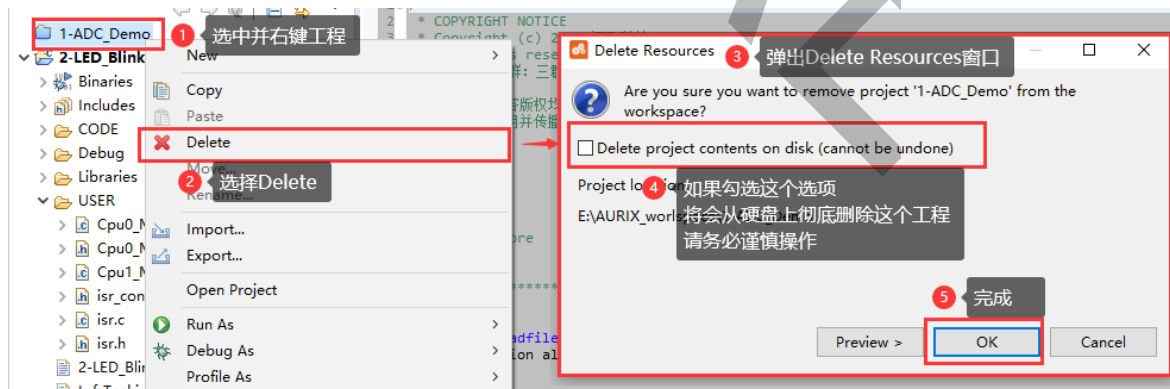
关闭工程的操作为：**右键工程->Close Project**。打开工程的操作为：**双击工程**。



3.4.删除工程

如果当前工作空间内有工程不需要进行修改、调试时，可以将其移除工程，操作为：**右键工程->Delete**。此时会弹出 **Delete Resources** 窗口。

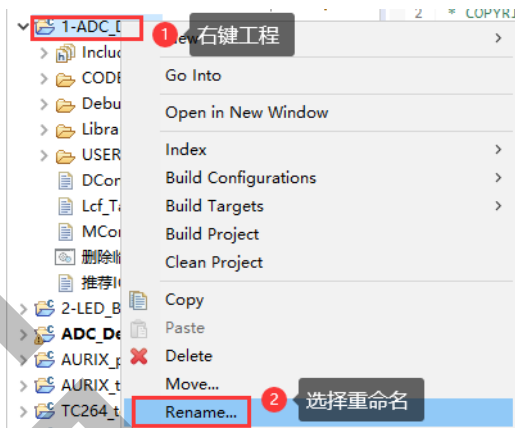
如果勾选 **Delete project contents on disk (cannot be undone)** 选项会从工作空间中移除该工程并且从磁盘<硬盘>中彻底删除该工程，请谨慎操作！



如果不勾选该选项，则只从工作空间删除该工程，可以在需要该工程时通过 Import 从工作空间所在目录添加该工程回到工作空间<参考 1.1>。

3.5.重命名工程

我们选中需要重命名的工程，右键，选中 Rename...选项。



虽然使用此方法可以重命名工程，但我们并不建议如此使用，因为 ADS 目前版本还存在重命名的 bug——仅会修改.project 文件内的工程名字，而不修改.cproject 文件。这将导致用户无法正常进行 debug 调试。我们的建议是手动修改工程名字：

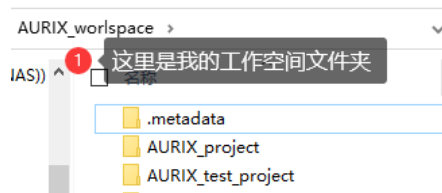
- 1.修改工程文件夹名字。
- 2.使用记事本打开.project 文件，使用 Ctrl+H 替换掉原来的工程名。
3. 使用记事本打开.cproject 文件，使用 Ctrl+H 替换掉原来的工程名。

3.6.添加、新建文件

如果需要向工程中添加文件，需要注意包含路径的问题，我们的开源库本身的包含路径已经设置完善，并且我们推荐各位在我们开源库下的 CODE 文件夹下新建文件。

3.6.1.文件管理器添加文件

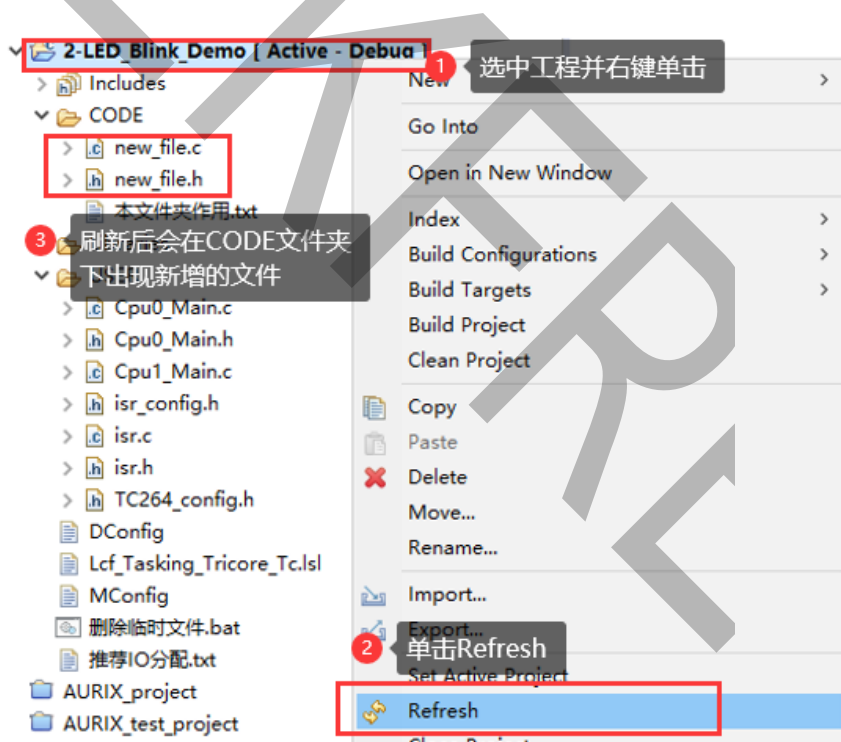
首先是通过文件管理器直接添加文件的方式，我们默认各位按照我们之前的导入工程步骤操作，那么此时的工程对应的实际位置，就在工作空间的文件夹下。



那么我们打开文件管理器，找到对应的工作空间路径，打开对应的工程文件夹，进入我们推荐的添加文件的文件夹 CODE，这里我们以 2-LED_Blink_Demo 工程为例：

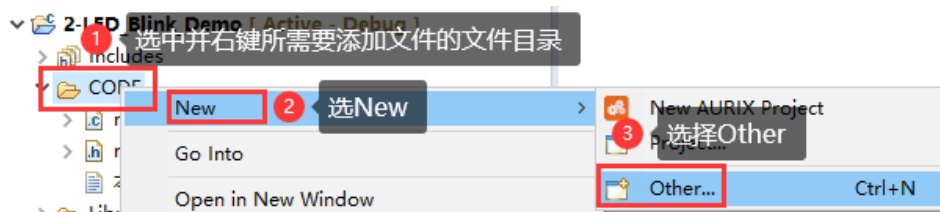


然后回到 ADS 中，选中并右键 2-LED_Blink_Demo 工程，单击 Refresh 刷新工程，此时即可在 CODE 目录下看到新增的文件：

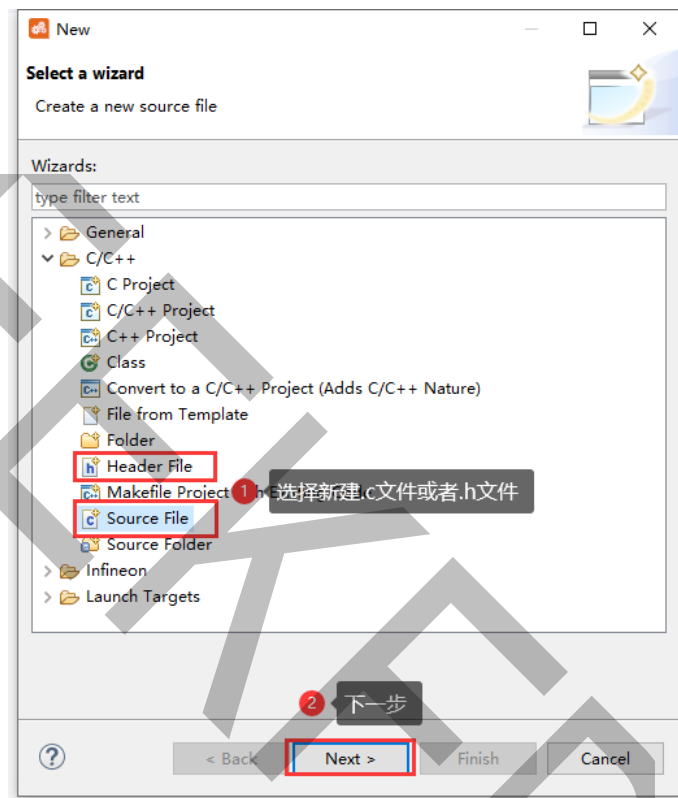


3.6.2.ADS 新建文件

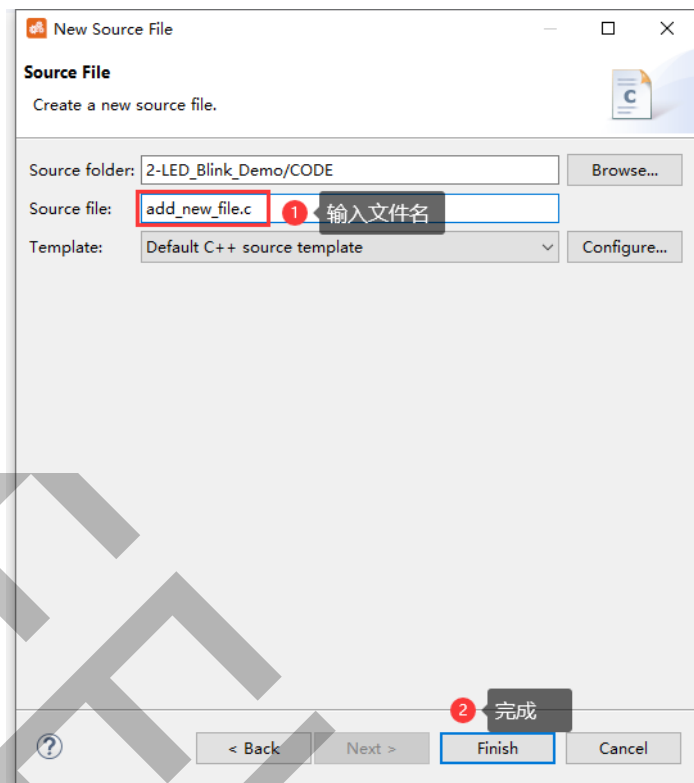
然后是直接使用 ADS 自身的新建功能新建文件，首先选中目录并右键选择 New->Other:



然后在弹窗中选择相应的文件类型，单击下一步：



然后输入文件名，单击完成即可，请注意只允许使用英文、数字与下划线：

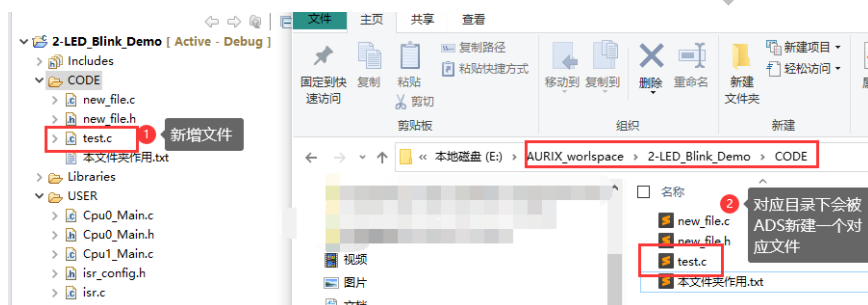


3.6.3.直接添加文件

最后一个是最直接简单粗暴的方法，直接添加文件：

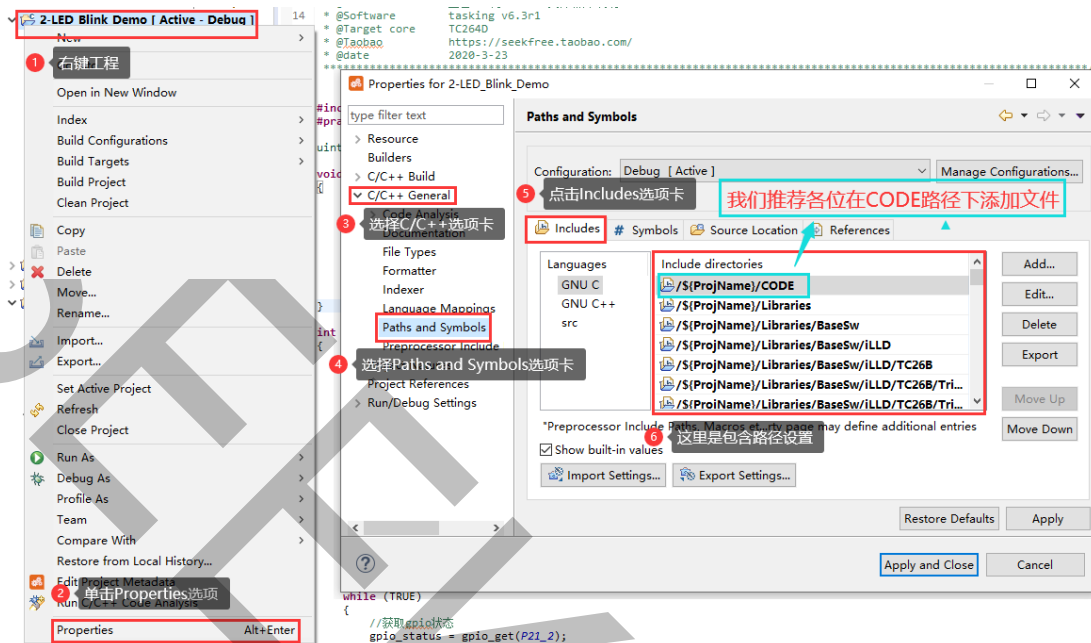


直接将文件拖动到 ADS 中, 此时会直接拷贝一份该文件在对应目录下, 并添加到工程内。

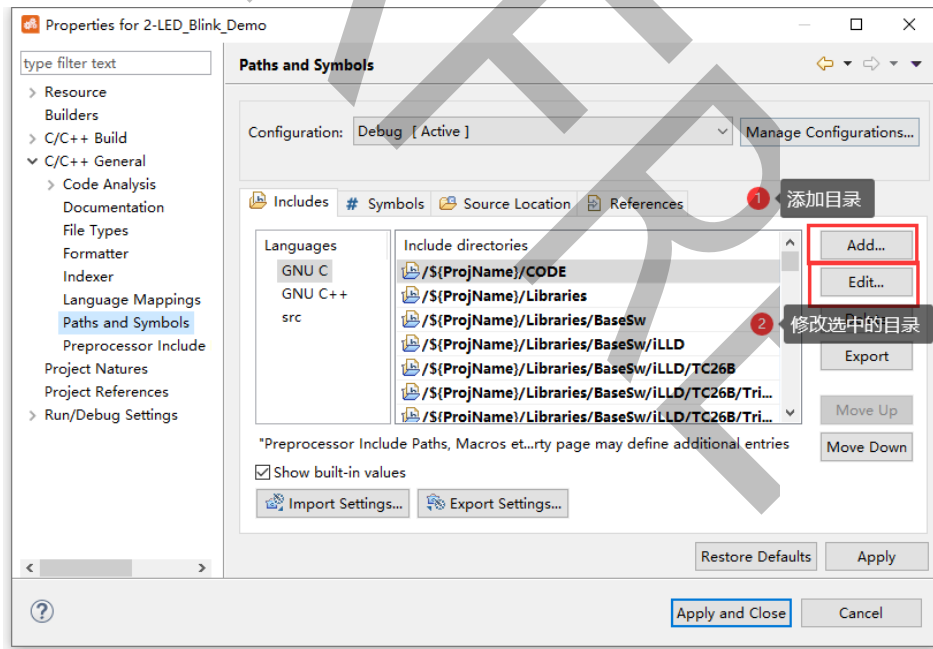


3.7.包含路径

我们的库中已经设置好了相应的包含路径，包含路径查看如下：



如果需要添加自己的文件夹目录，需要在这里设置添加对应的包含路径。



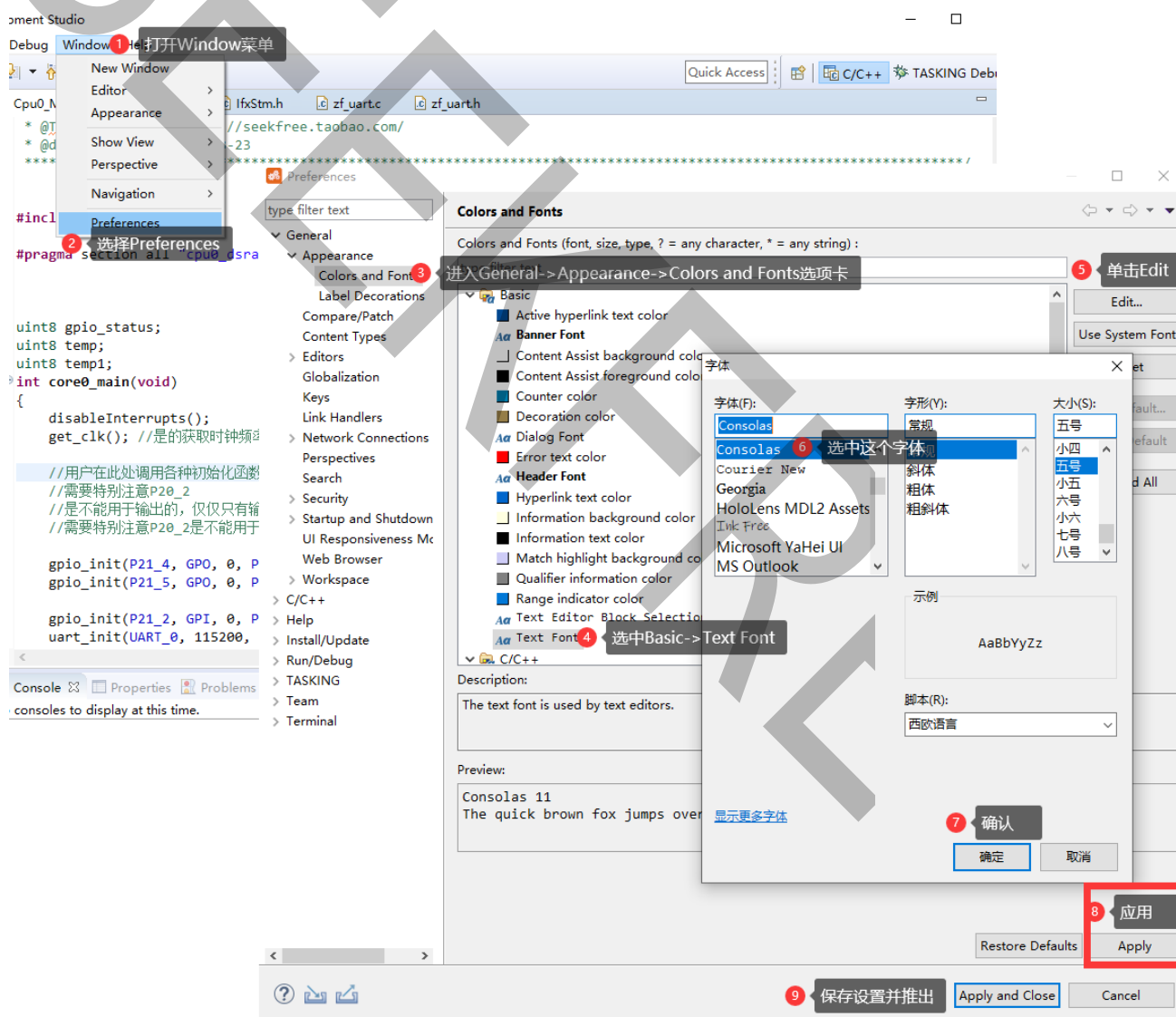
注意：1.5.2 以上版本的 ADS 默认设置为自动包含路径，工程文件夹内的所有文件都会主动进行包含，无需用户再进行路径添加的操作。

4.环境设置

4.1.字体设置

ADS 默认的字体是非等宽的字体，这样会导致代码对齐变得困难，并且格式变得混乱，显得代码格式杂乱，不美观，不方便查找与编写。

所以我们需要重新设置字体，使用等宽字体保证代码的美观性与可阅读性。

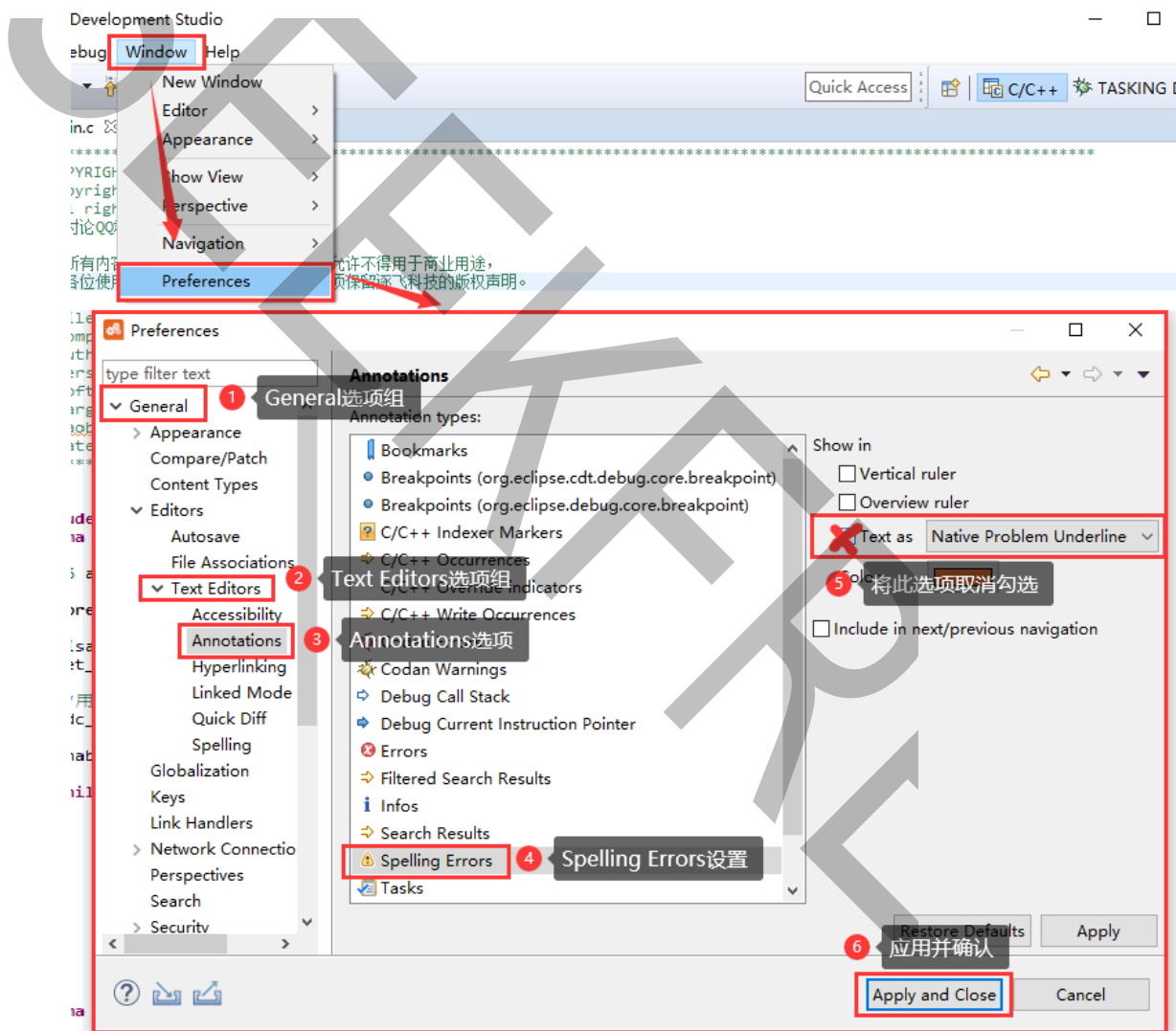


4.2.拼写报错屏蔽设置

在 ADS 中默认对注释也有拼写检查，会在注释里提示拼写错误：

```
58 while (TRUE)
59 {
60     //获取gpio状态
61     gpio_status = gpio_get(P21_2);
62     //将gpio状态打印到FSS窗口
63     printf("gpio_status: %d\n", gpio_status);
64 }
```

按照下述方法将拼写报错关闭即可。



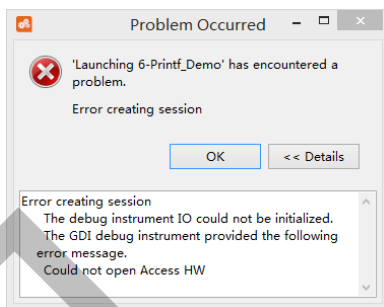
完成后不再提示拼写错误：

```
60 //获取gpio状态
61 gpio_status = gpio_get(P21_2);
62 //将gpio状态打印到FSS窗口
63 printf("gpio_status: %d\n", gpio_status);
64
```

5.常见问题及解决方案

5.1.硬件未连接报错

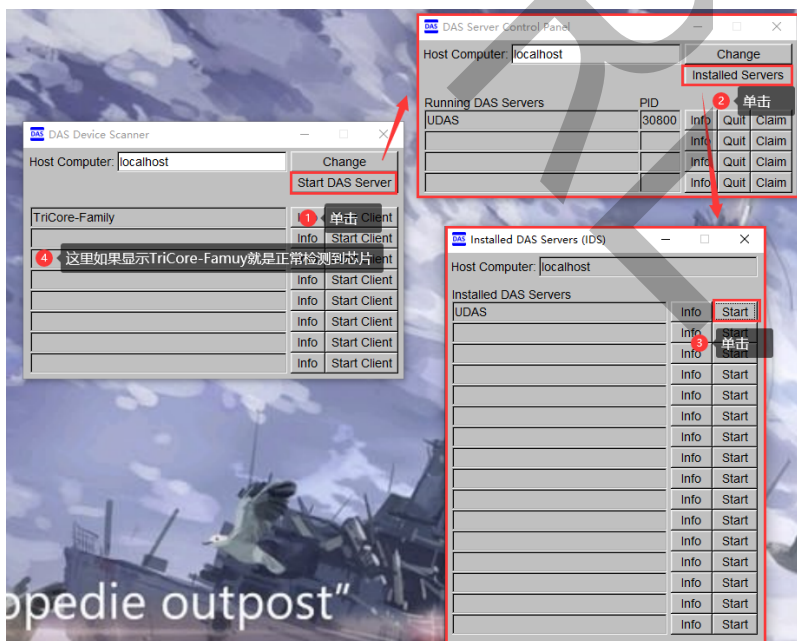
报这个错证明没有检测到调试下载器、没有检测到芯片（核心板）。



请检查调试下载器是否正确连接，核心板硬件连接是否正确。调试下载器是否正常被电脑识别，DAS Device Scanner 是否能够扫描到核心。在开始菜单中找到 DAS Device Scanner。



打开后按照下图所示操作：



如果显示 No device 的话证明没有检测到芯片，请检查连接。

如果显示 TriCore-Family 的话证明检测到芯片，回到 ADS 继续调试，如果依旧报同样错误则 clean 工程后删除 launch 调试配置重新编译构建，参见 2.3 章节。

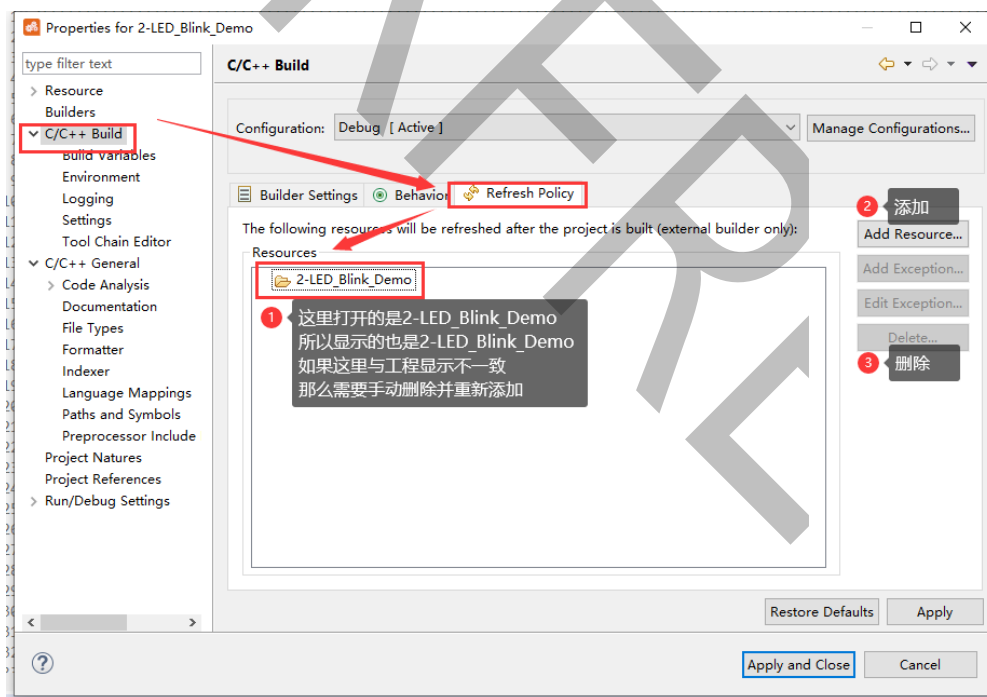
5.2.宏定义未生效问题

修改宏定义后直接编译调试，发现工程异常，程序运行没有按照宏定义正常生效，该问题是 ADS 本身的问题，需要 Clean 工程后重新编译工程即可。

5.3.每次调试时都会重新编译的问题

在导入以往工程，或者别人的工程时，有部分工程设置不会跟随导入改变，会出现一些问题，其中包括在每次调试时，都会重新编译工程，十分浪费时间。

如果出现该现象，右键工程打开工程设置，查看工程的关联设置：



5.4.重命名工程后编译、调试报错的问题

与 5.3 章节一样，如果出现该现象，右键工程打开工程设置，查看工程的关联设置。

6.ADS 下载方式

6.1.百度云盘下载

在电脑上安装百度云盘的软件后，在浏览器中输入一下链接，页面打开后输入提取码，即可进入下载页面。链接中包含了 ADS 软件 1.8.0 版本以及使用 ADS 的详细教程。如果 ADS 有更新我们将会在这个链接上进行同步更新。

链接： <https://pan.baidu.com/s/1s340z2pADOTttxliqaqwgA>

提取码：kscj

6.2.官网下载

第一步先打开英飞凌官网并注册账号。

第二步打开这个链接

<https://www.infineon.com/cms/en/product/promopages/aurix-development-studio/#>。

第三步点击 Download here，然后填写信息等待官方将下载地址发到注册账号时填写的邮箱。

第四步收到下载链接后，打开下载链接即可开始下载 ADS。

7. 文档版本

版本号	日期	内容变更
V1.0	2020-10-09	初始版本
V1.1	2020-10-12	更新 project_file 文件夹，简化步骤
V1.2	2020-10-26	增加下载说明
V1.3	2020-11-02	增加切换工作空间描述、增加文件添加描述、增加调试硬件连接提示
V1.4	2020-11-05	新增工程重命名，添加拼写报错设置
V1.5	2020-11-25	新增 Flash 下载描述、开源库版本描述以及修改排版
V1.6	2020-12-01	删除导入修改工程部分
V1.7	2021-01-12	新增常见错误章节 修改文件添加描述
V1.8	2021-01-21	更新 printf 描述 新增调试窗口添加 添加重命名工程报错问题
V1.9	2022-12-07	简化部分操作步骤