

# TASKING 环境使用说明

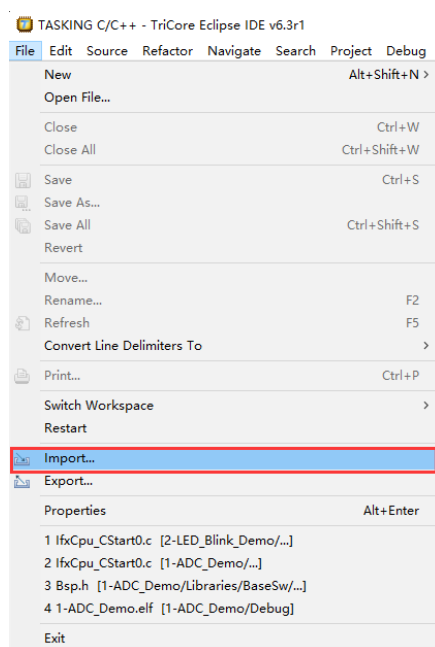
# 目录

目录.....	1
1. 工程的导入、拷贝.....	2
1.1. 导入现有工程 .....	2
1.2. 拷贝工程.....	4
2. 编译、调试工程.....	6
2.1. 编译工程.....	6
2.2. 调试工程.....	6
2.2.1. 断点以及单步调试功能 .....	7
2.2.2. 变量查看功能 .....	8
2.2.3. 内存查看功能 .....	9
2.2.4. log 输出功能.....	10
3. 工作空间管理.....	12
3.1. 打开、关闭或切换工程.....	12
3.1.1. 切换工程 .....	12
3.1.2. 打开、关闭工程.....	14
3.2. 删除工程.....	15
4. 文档版本 .....	16

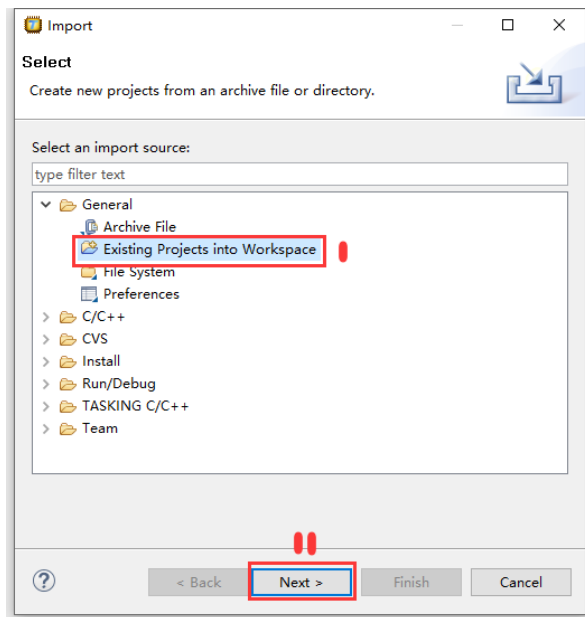
# 1.工程的导入、拷贝

## 1.1.导入现有工程

安装完成 TASKING 并打开 IDE(安装的时候请务必保证 tasking 安装路径没有中文与空格!), 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>, 点击工具栏: **File->Import..**选项, 进入到导入工程界面, 如下图所示。



进入到导入选项卡, 选择: **General->Existing Projects into Workspace** 进行导入现有工程, 如下图所示:

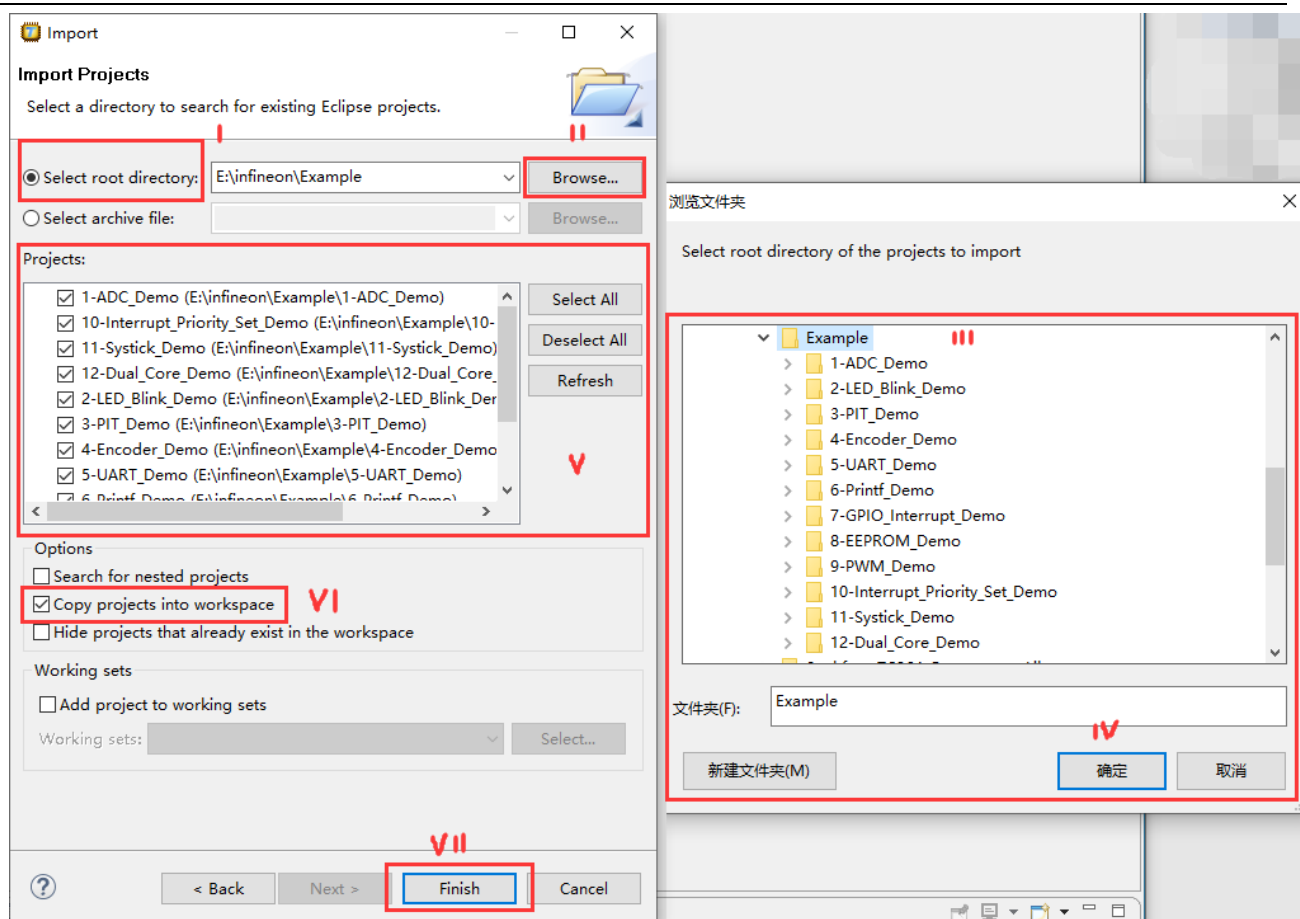


进入到工程导入选择界面，以我们提供的例程作为例子，导入工程。<先导步骤：下载逐飞 TC264 开源库及例程后，保存并解压，**请注意保存路径不要包含中文以及空格！同时请注意工作空间路径不要包含中文以及空格！**>由 **Select root directory** 方式选择例程所在文件夹，在 **projects** 列表中选择所需要导入的例程，这里可以一次性导入全部例程到你的工作区。

**注意：**推荐在 **Options** 选项中勾选 **Copy projects into workspace**。这里是为了防止后续在使用、调试、修改例程后无法恢复原本例程状态而进行的类似备份操作，将你需要的工程拷贝到你的工作空间，而不直接对原工程产生影响。同时也避免由于拷贝、打开工程时由于路径、包含文件的原因导致出错。

如果是导入之前从工作空间移除的工程则不需要勾选 **Copy projects into workspace**。

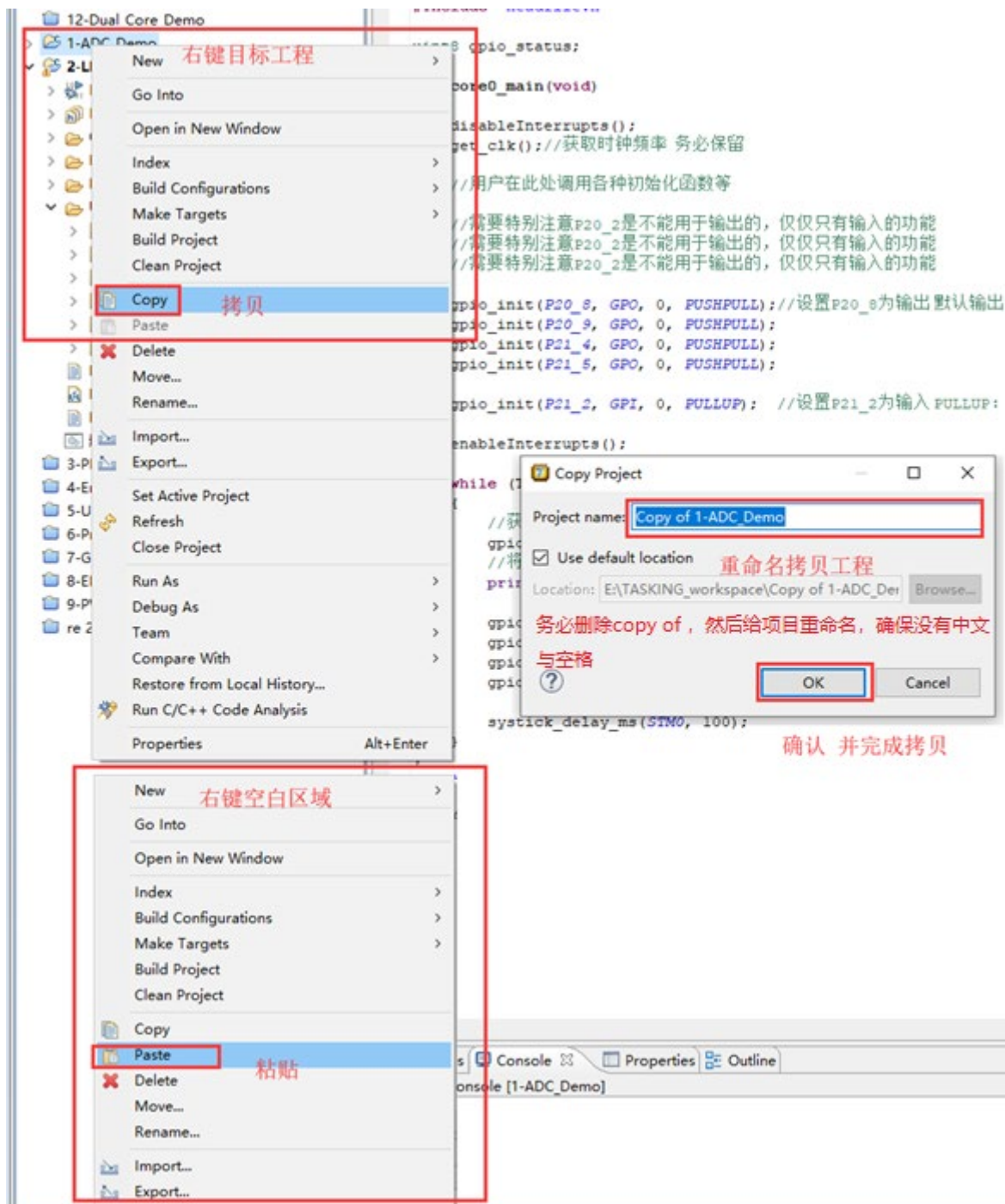
步骤如下如所示：



点击确定后，所有的例程已全数导入到你的工作空间内。如果不需要导入所有例程工程，请根据需要在 **projects** 列表中选择所需要导入的例程。

## 1.2.拷贝工程

不建议各位自行新建工程，各位可以将例程或模板拷贝后再进行代码的编写。拷贝工程的步骤为：在 C/C++ Projects 选项卡<界面左侧选项卡>中**右键目标工程**，选择 **Copy** 选项，在 C/C++ Projects 选项卡**空白处右键鼠标**，选择 **Paste** 选项，在 **Copy Project** 弹窗中**重命名工程并单击 OK** 完成拷贝<请注意重命名尽量不要包含中文以及空格>。步骤如下图所示：





## 2.编译、调试工程

### 2.1.编译工程

编译工程步骤与其他环境差别不大，方式为：

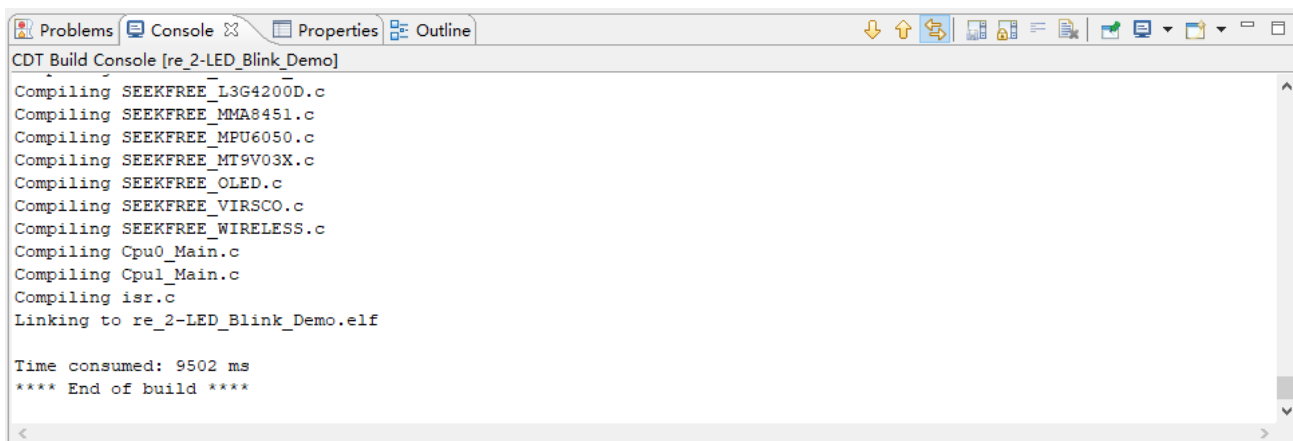
1. 通过右键工程选择 Build Project 选项；

2. 上方选项栏中  Build 或 Rebuild 选项；

3. 上方选项栏中  Build ALL 选项。

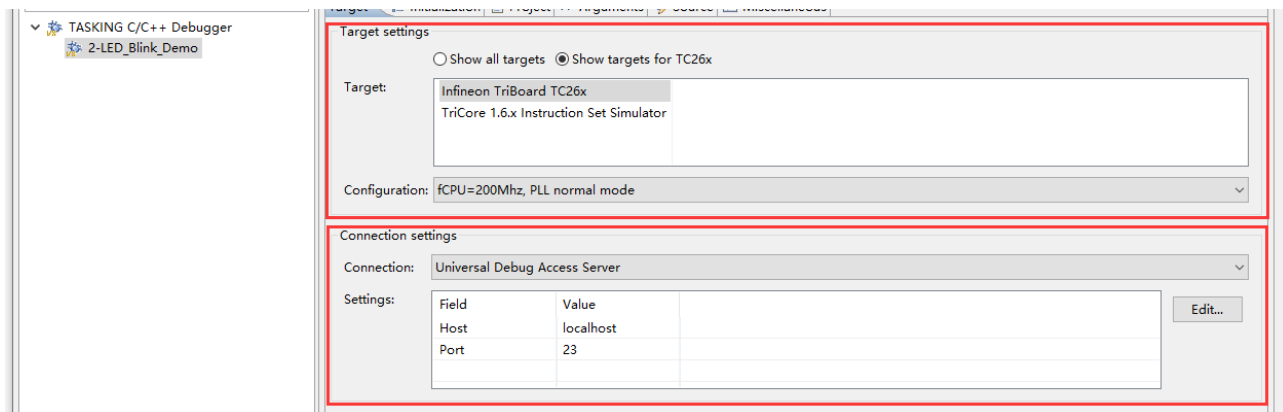
注意，如果当前工作区内打开了多个工程时，请谨慎选择第三个方式，该方式会将打开的所有工程编译，极其耗时。<打开工程与关闭工程请参照 3.1.2>

下方 Console 选项卡会输出编译文件步骤以及最终耗时，以及相关报错信息。

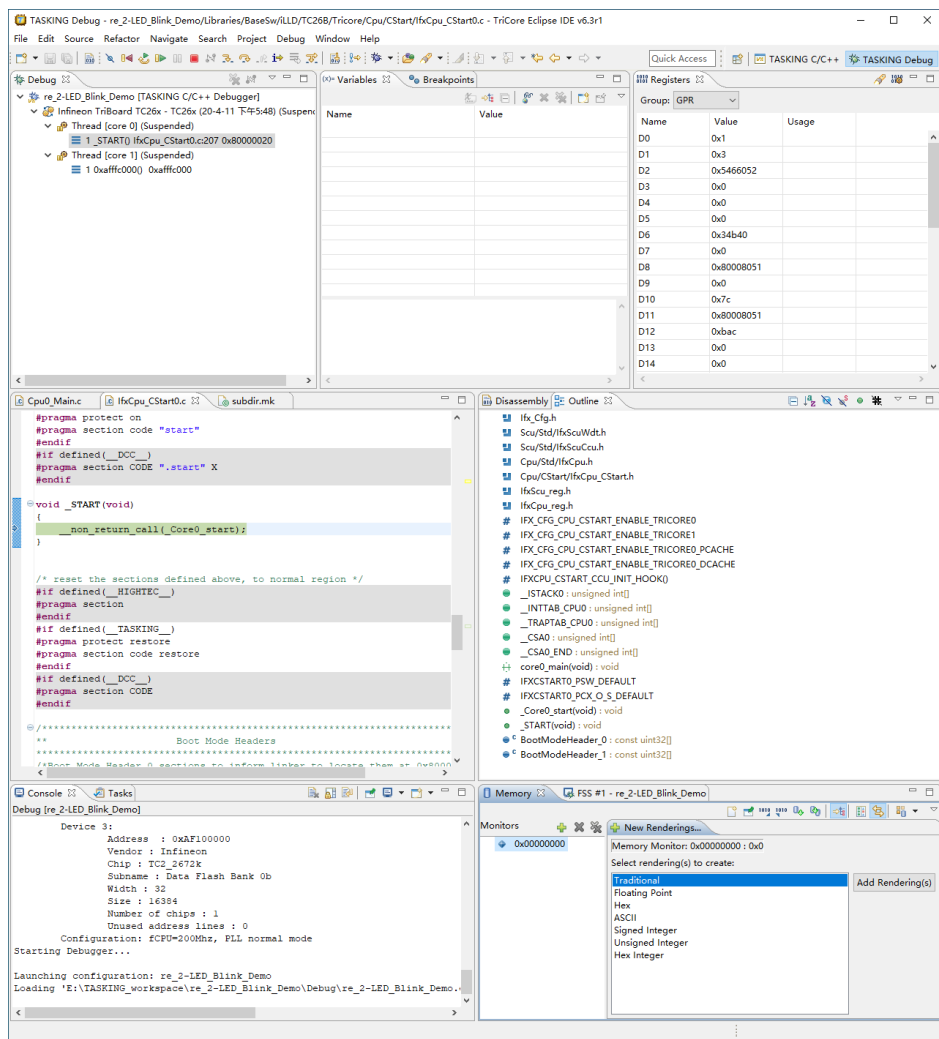


### 2.2.调试工程

调试工程可以通过上方选项栏  Debug 选项进行调试，首次点击时，会进入 Debug Configurations 窗口进行设置，按照下方默认设置<默认设置不需要更改>。



请确保在已连接调试下载器, 并连接正常的情况下, 进行调试, 下图为正常进入调试界面:



## 2.2.1.断点以及单步调试功能

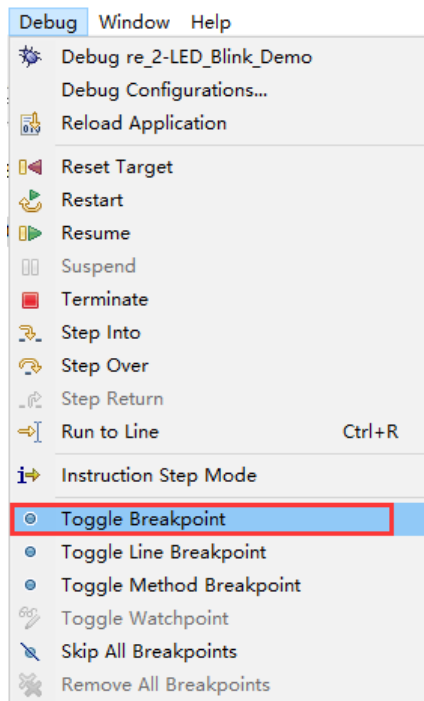
设置断点可以通过在需要设置断点的行数左侧双击设置断点: <Tips:右键蓝条部分勾选



Show Line Numbers 显示行数>


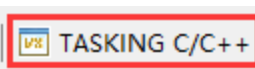


```
41  gpio_init(P21_2, GPI, 0, PULLUP); //设置P21_2为输入 PULLUP: 上拉输
42
43  enableInterrupts();
```

或可以单击所需要添加断点的行，在菜单栏中 **Debug->Toggle Breakpoint** 设置断点。



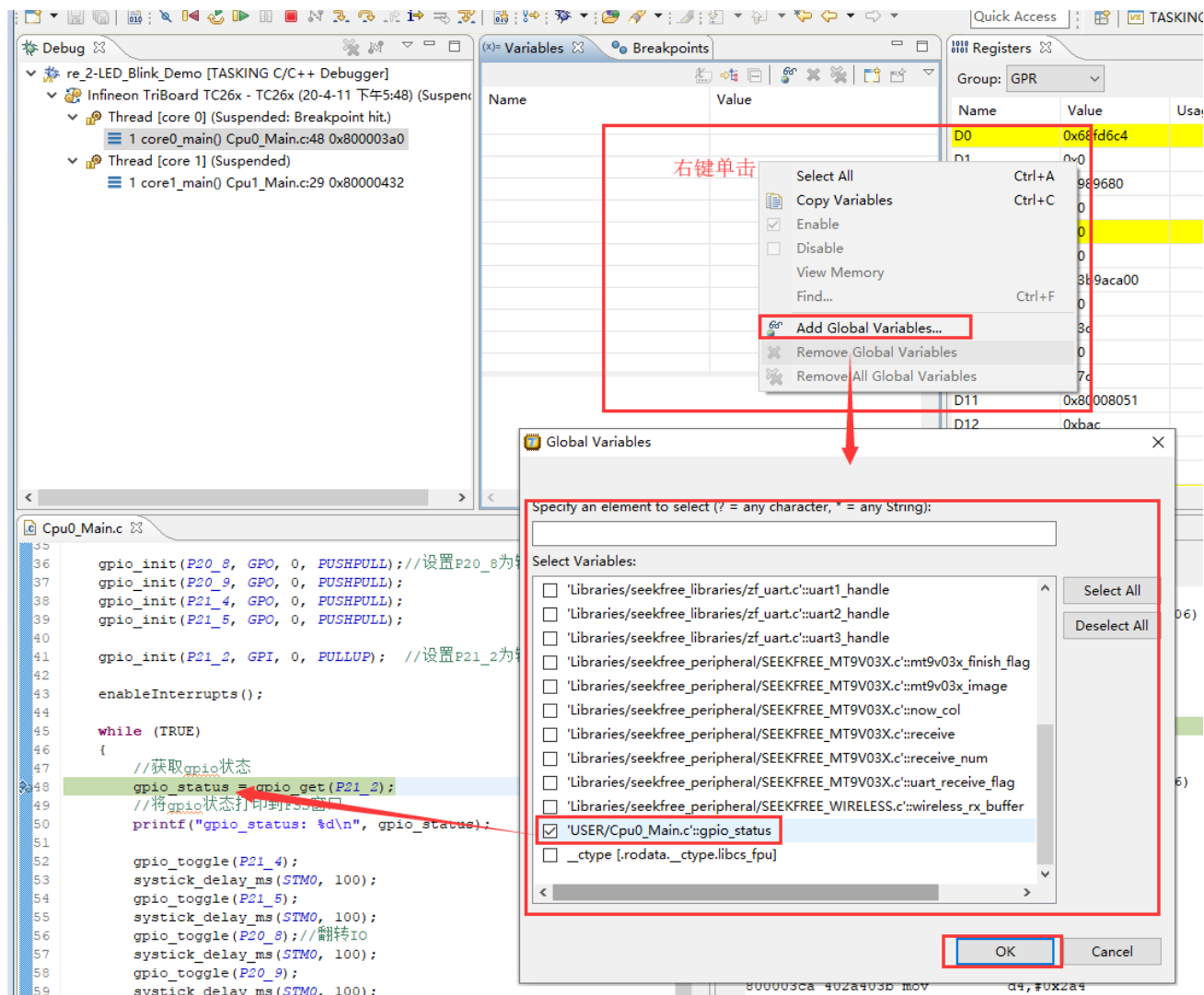
进入 Debug 模式后，点击工具栏  **Resume<快捷键 F8>**选项即可运行代码，运行到断点处即停止，此刻可以使用  **Step Into<快捷键 F5>**进行指令单步<每个操作为一步，可以跳转到所调用函数的定义>调试，或可以使用  **Step Over<快捷键 F6>**进行代码单步<当前函数单行为一步>调试。

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。

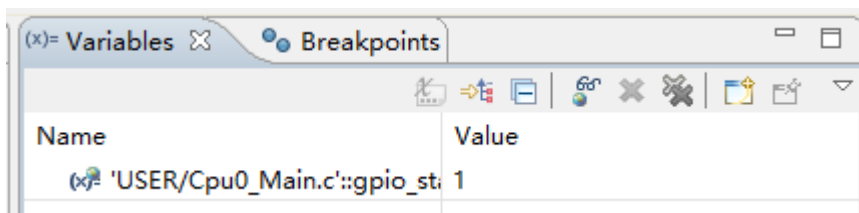
## 2.2.2.变量查看功能

在调试界面内，可以通过上方<默认在上方>的 **Variables** 选项卡查看全局变量，在选项卡内空白区域，**右键->Add Global Variables...**进入 Global Variables 窗口选择所需要查看的全局

变量，勾选并单击 OK 即可，如下图所示：



此时在 Variables 选项卡可查看到所选变量的状态：

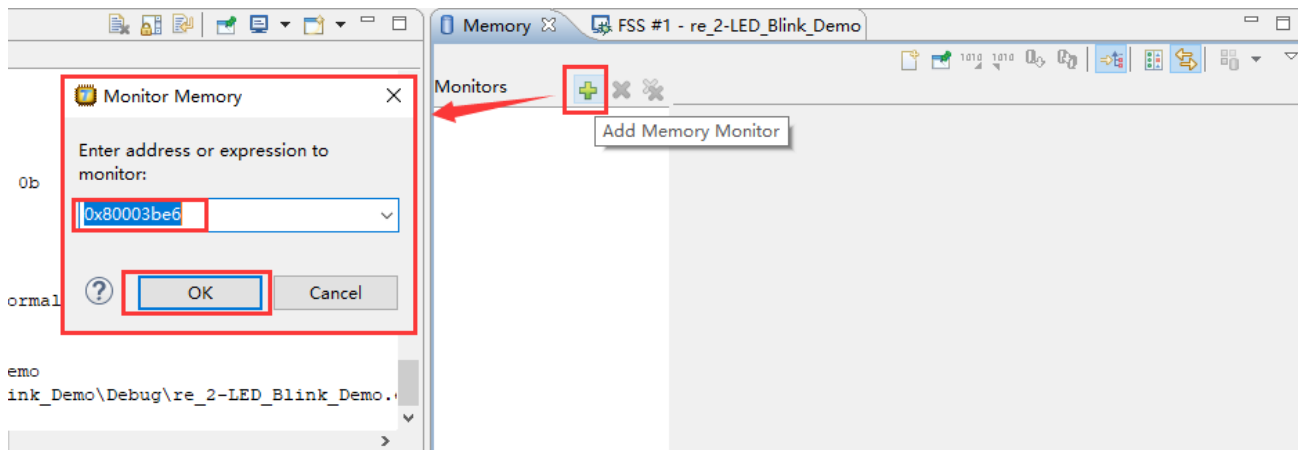


<Tips: 请注意只能添加全局变量>

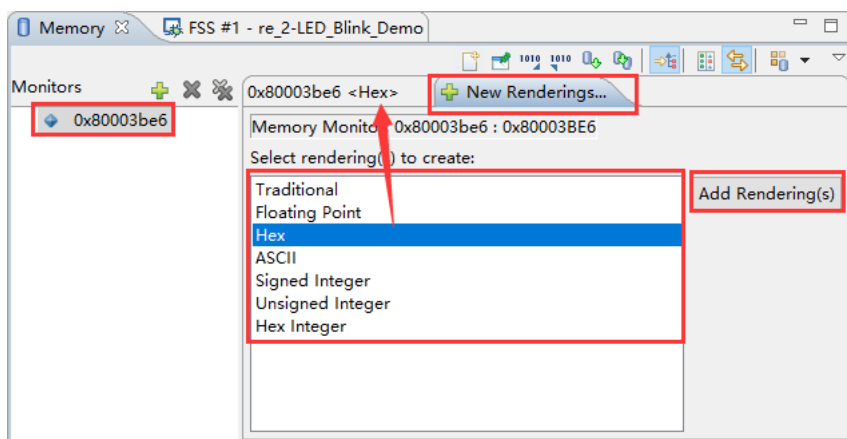
### 2.2.3.内存查看功能

TASKING 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

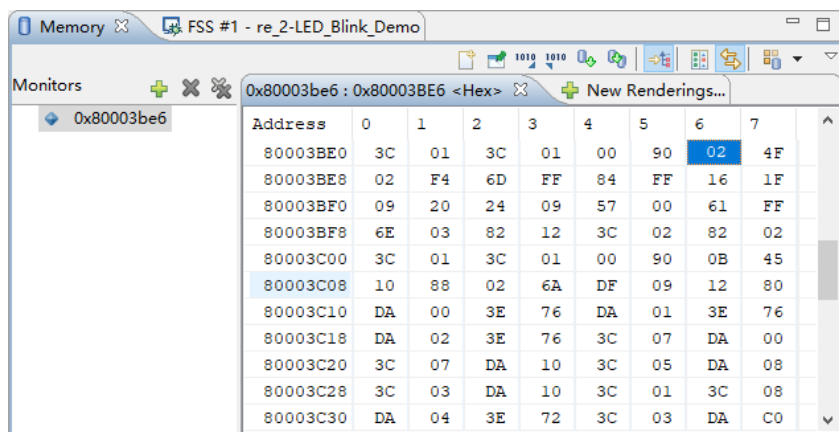
项卡可以添加不同的数据格式，具体操作如下图：



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：



添加完成后如下图：



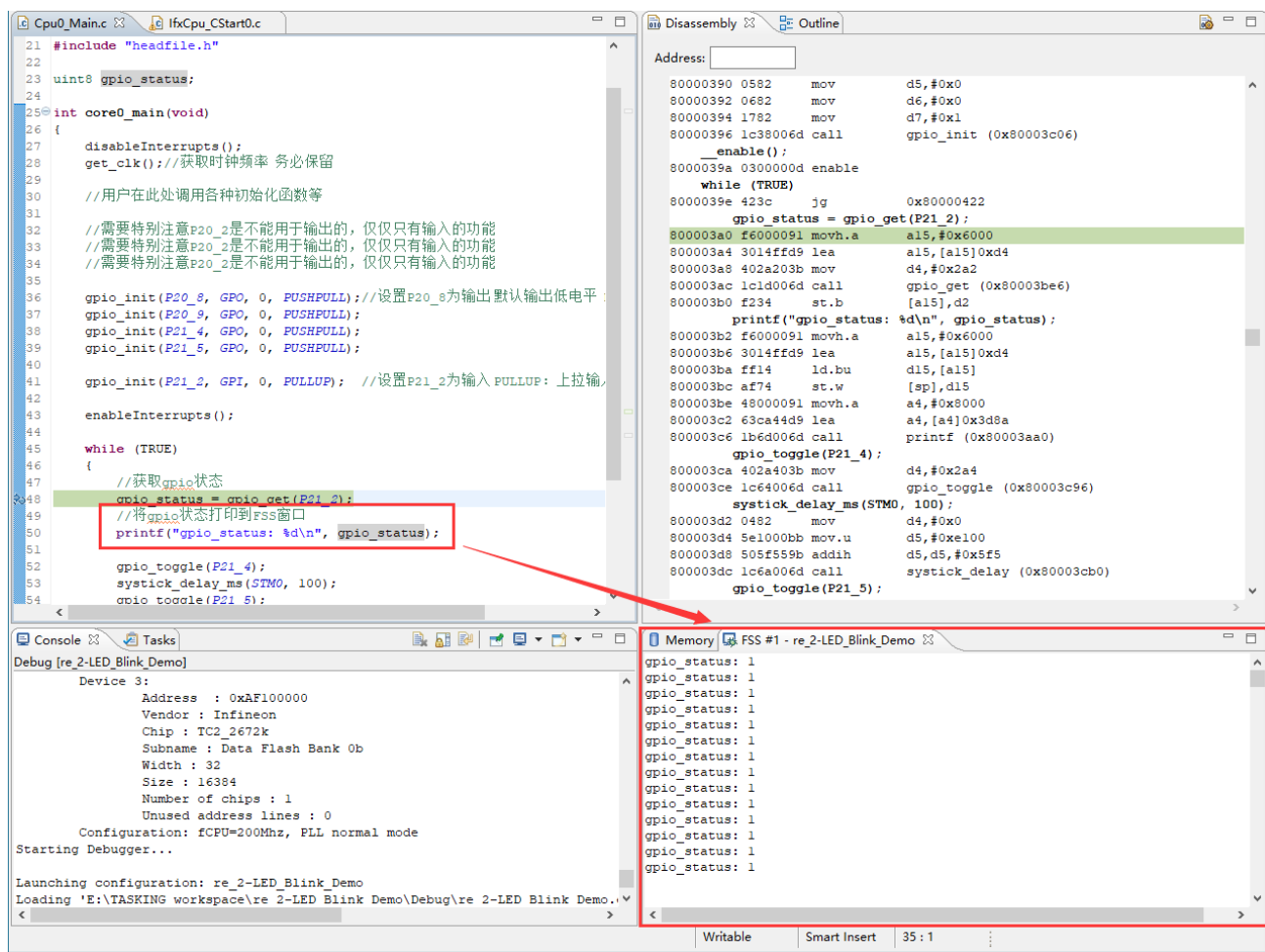
## 2.2.4.log 输出功能

TASKING 还提供了 log 输出的工具，可以通过重定向的 **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。版本可以后续在熟悉 TASKING 环境后再行尝试。

实际运行效果如下图：

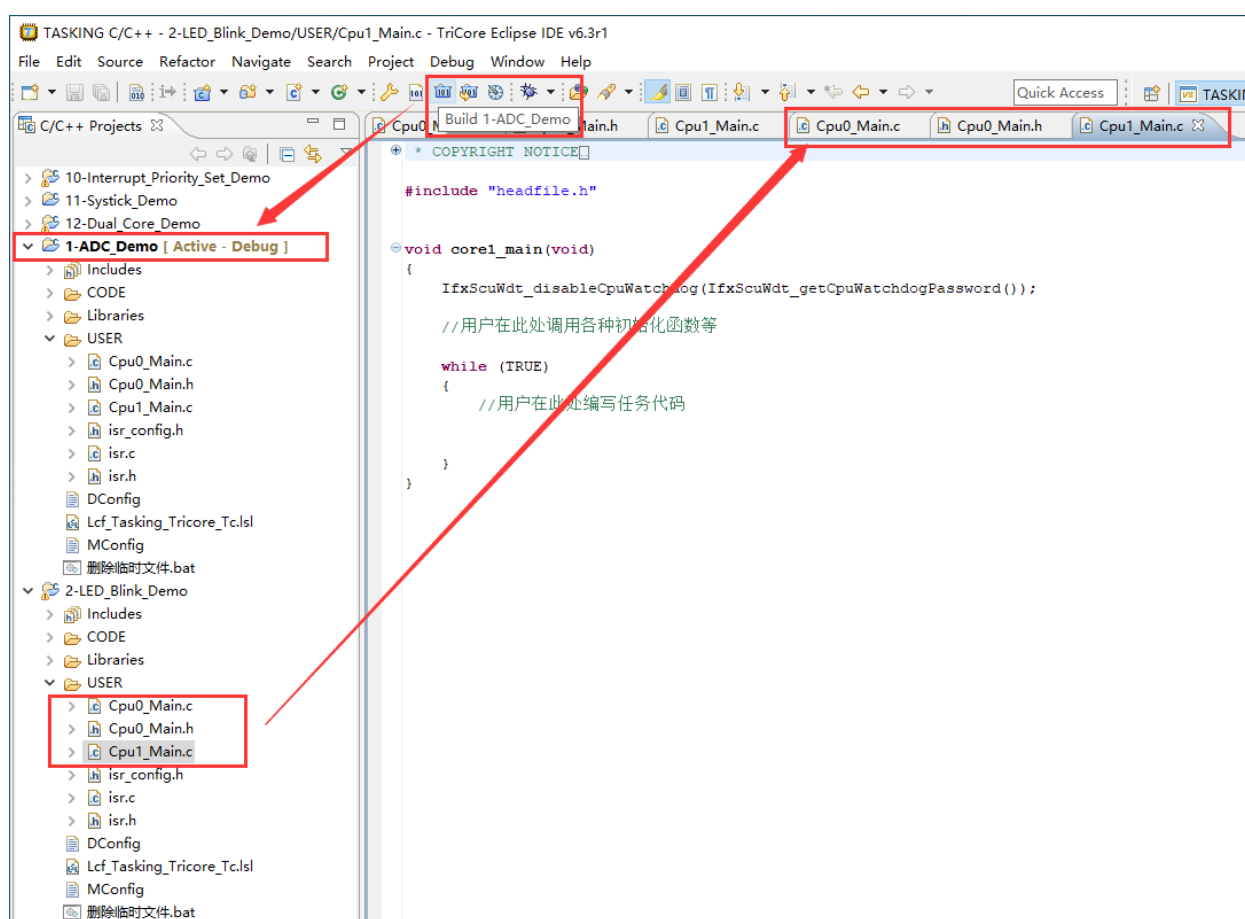


## 3.工作空间管理

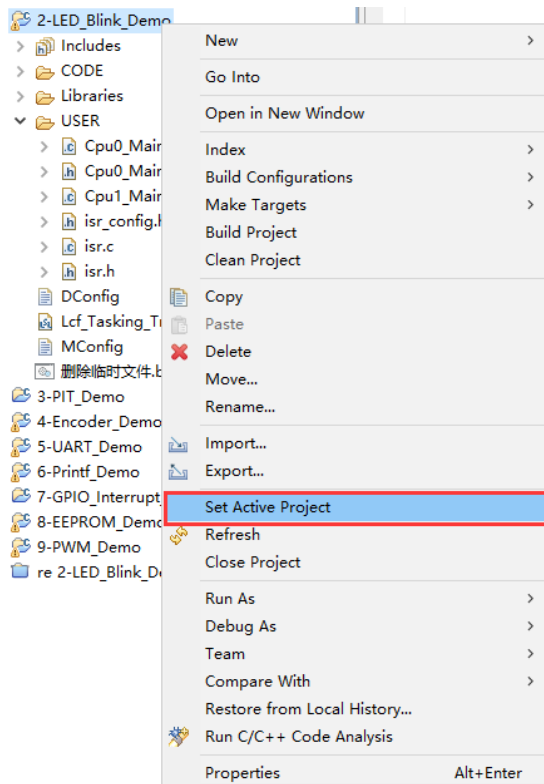
### 3.1.打开、关闭或切换工程

#### 3.1.1.切换工程

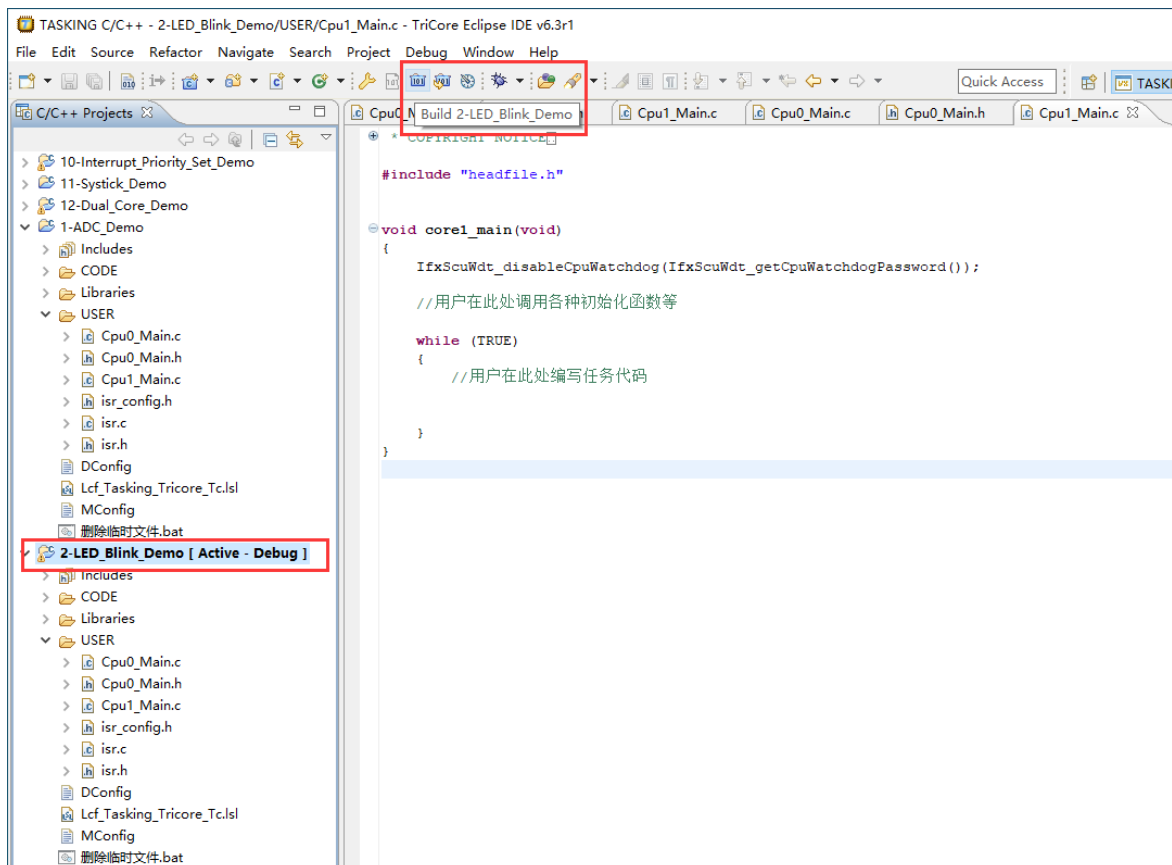
当你的工作空间中留存有许多的工程时，需要注意打开的源文件是否是 **Active Debug** 状态，Active Debug 状态下的工程会在 C/C++ Projects 选项卡中以粗体标示，表示当前该工程处于 Active Debug 状态，编译、调试等操作会针对于该工程进行。





如上图所示，我当前打开了例程<2-LED\_Blink\_Demo>的源文件，但实际 Active Project 为 <1-ADC\_Demo>，所以此时我点击编译时，会编译例程<1-ADC\_Demo>而不是<2-LED\_Blink\_Demo>，请务必注意这一点。此时，我们需要切换<2-LED\_Blink\_Demo>为 Active Debug 状态，在需要调试的工程上右键选择 **Set Active Project**，如下图所示：



切换后, <2-LED\_Blink\_Demo>变为 Active Debug 状态, 此时再进行编译、调试等操作时, 就会以<2-LED\_Blink\_Demo>工程为基础进行, 如下图所示:

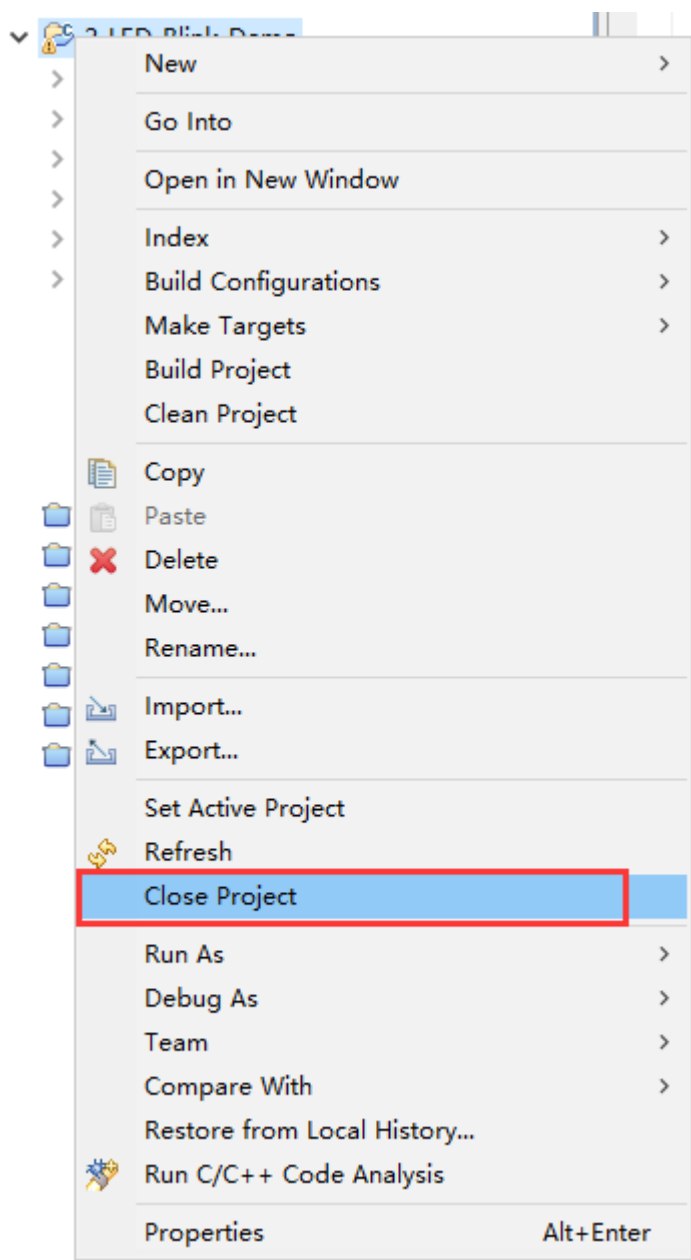


### 3.1.2.打开、关闭工程

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。



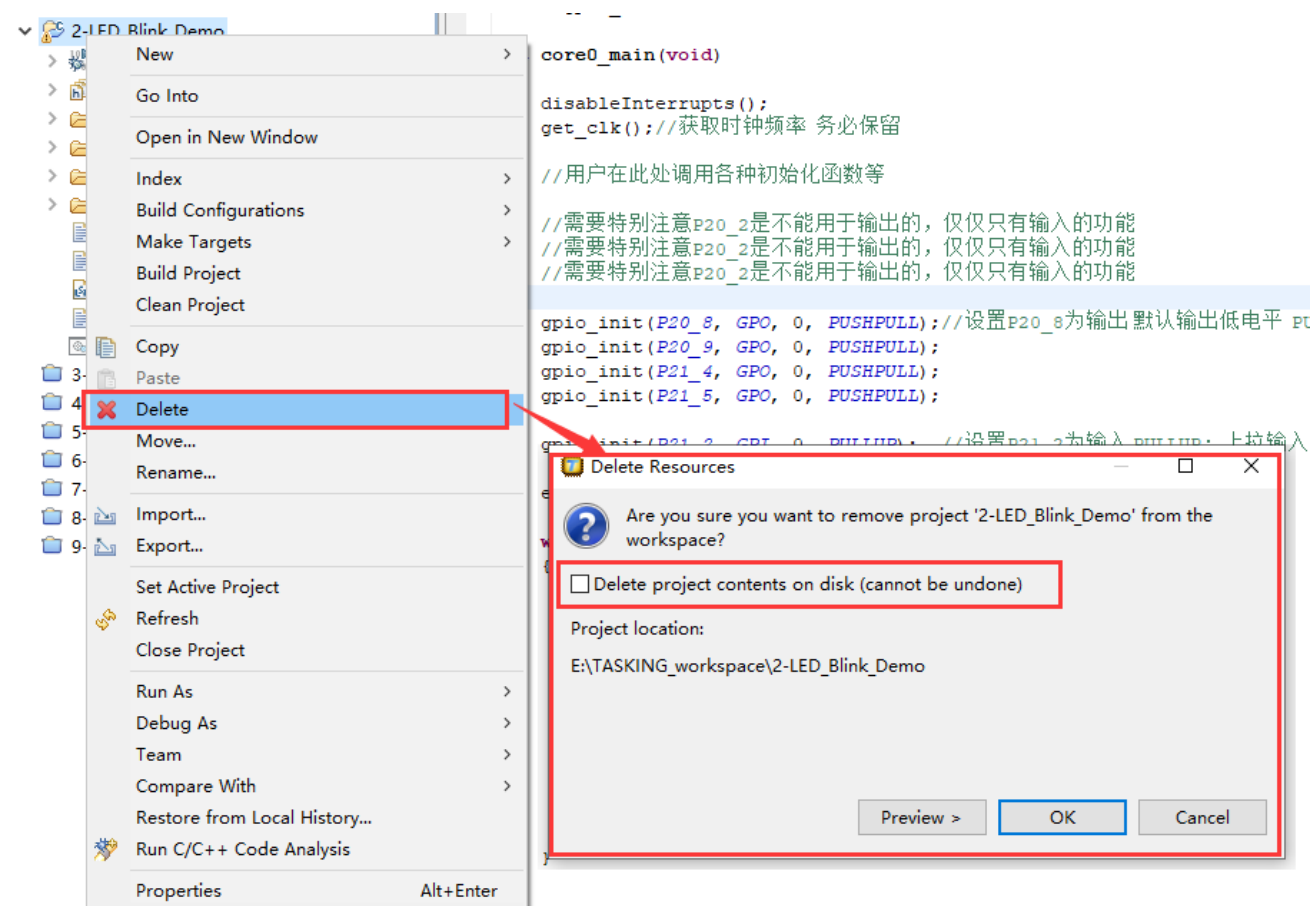
打开工程的操作为：双击工程。

## 3.2. 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 1.2>。





## 4.文档版本

版本号	日期	内容变更
V1.0	2020-04-11	初始版本。
V1.1	2020-04-14	删除 problems 描述
V1.2	2020-05-12	增加路径强调说明