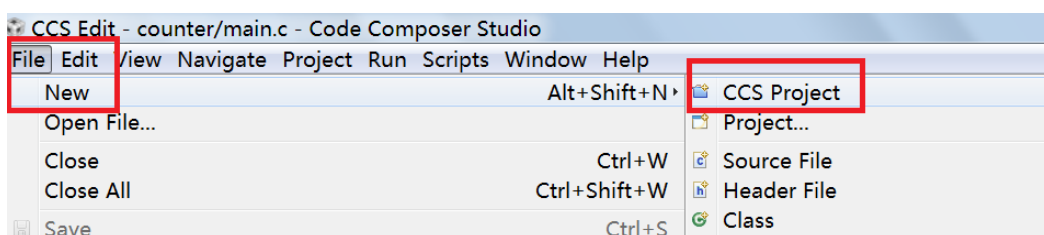

CCS 基本操作

1 新建工程

1.1 创建一个空白工程

(1) 欢迎屏幕关闭之后，将会显示厦门的工作区，此时可以创建新项目。转到菜单“File->New->CCS Project (文件->新建->CCS 项目)”如下图所示：



(2) 在“Project Name (项目名称)”字段中，键入新项目的名称。若选中“Use default location (使用默认位置)”选项（默认启用），将会在工作区文件夹中创建项目。默认位置勾选项为 Workspace 路径下，一般选择之前下载的 TM4C123G_LaunchPad_Workshop 的路径地址。取消选中该项目可以选择一个新位置（使用“Browse... (浏览...)”按钮）。为项目命名，例如“LAB”。

(3) 在“Target”菜单中选择要使用的芯片类型。Family 选择 ARM 是因为这款 LaunchPad 主芯片是内置了 ARM Cortex-M4F 内核的 TM4C 芯片。芯片序列号选择 123g, Tiva TM4C123GE6PM。

(4) 在“Connection”里选择调试器。选择内置调试器 Stellaris In_Circuit Debug Interface。

(5) 单击“Finish (完成)”创建一个空白 Project 项目。如下图所示：

New CCS Project

CCS Project

Create a new CCS Project.

Target: Tiva C Series Tiva TM4C123GH6PM

Connection: Stellaris In-Circuit Debug Interface Verify...

Cortex M [ARM]

Project name: LAB

☒ Use default location

Location: C:\Users\admin\workspace_v6_2\LAB Browse...

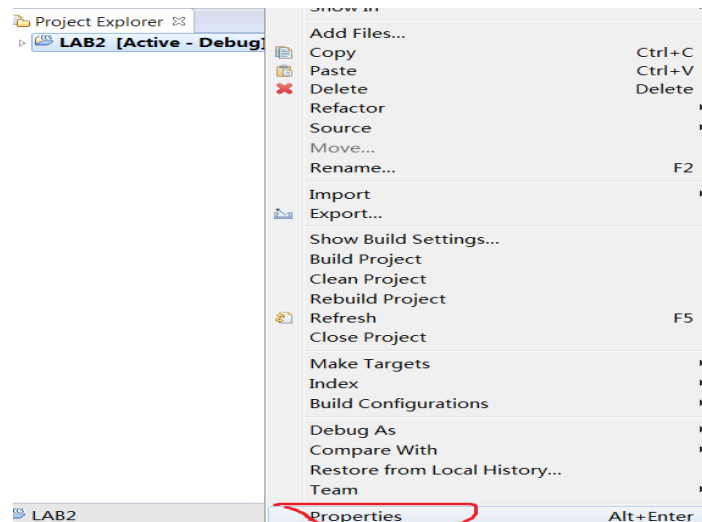
Compiler version: TI v15.12.3.LTS More...

Advanced settings

Project templates and examples

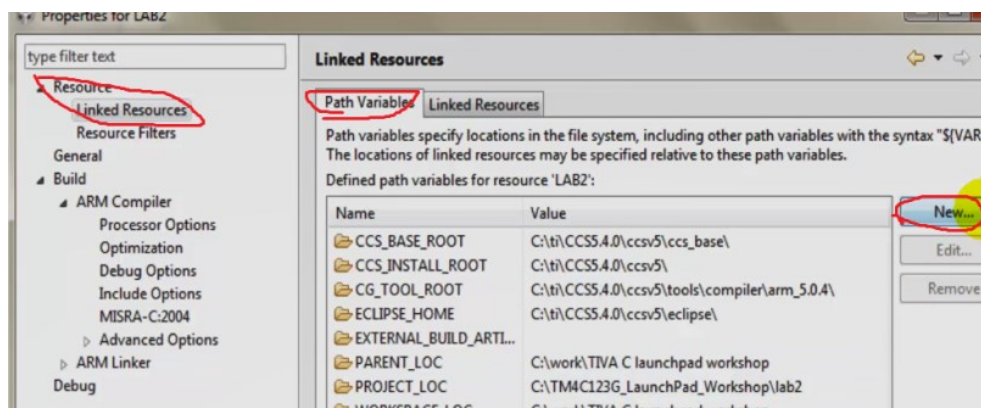
type filter text Creates an empty project fully initialized for the selected device

< Back Next > Finish Cancel



1.2 添加 Path Variables 路径变量

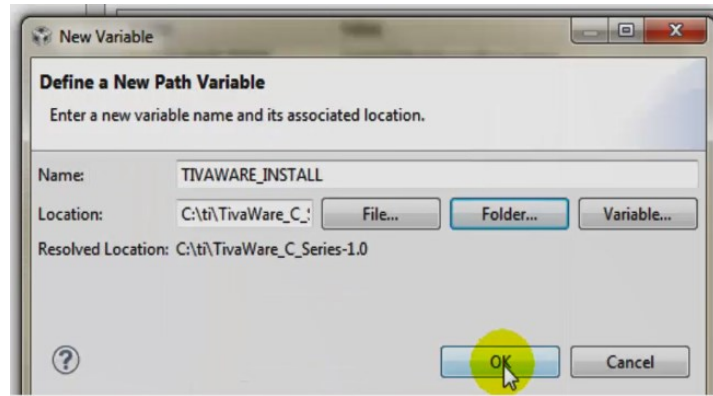
Path Variables 用于链接源文件和库，路径变量指定文件系统的位置。链接的资源可以指定相对于这些路径变量的位置。



(1) 单击鼠标右键，选择 Properties，首先添加 Path Variables 路径变量。

找到 Resource->Linked Resource,选择 new, 新建一个链接的路径, 指向 TIVAWARE 的文件夹下, 并取名为 TIVAWARE_INATALL。

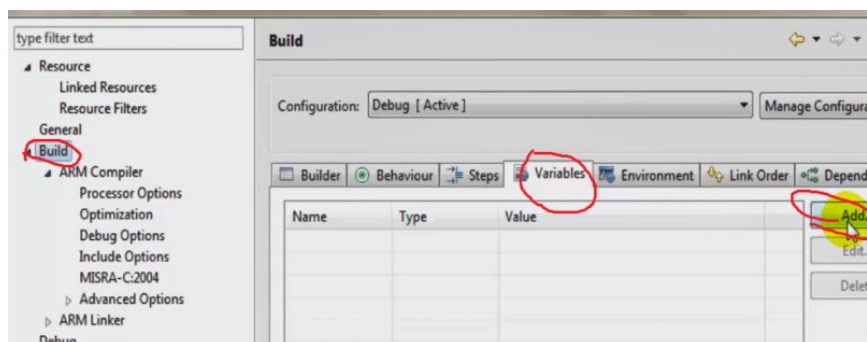
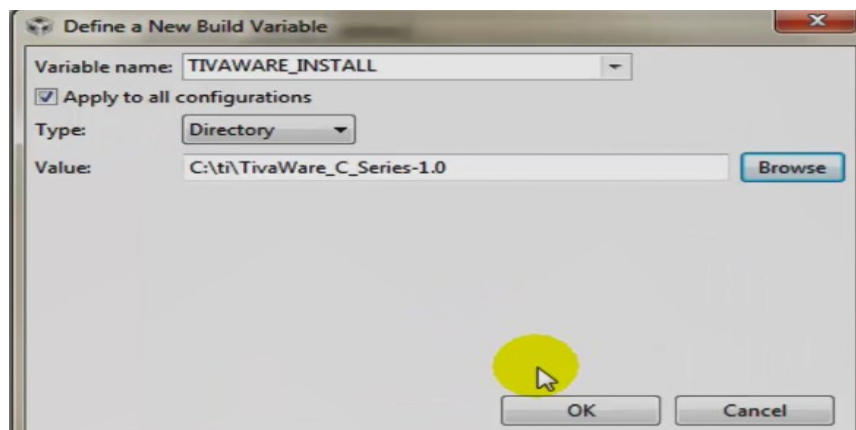
(2) 在出现的对话框中, 新建一个名为 TIVAWARE_INATALL 且路径在 TivaWare 下的 Path Variables。



1.3 添加 Build Variables 宏定义路径变量

Build Variables 是编译的环境变量，只有编译变量才能用于 include path 中。由于项目中需要使用 TivaWare 中包括的 Driver 链接库文件，因此需要添加 Build Variables。

- (1) 单击鼠标右键，选择 Properties，找到 Build->Variables，点击 ADD，添加路径。

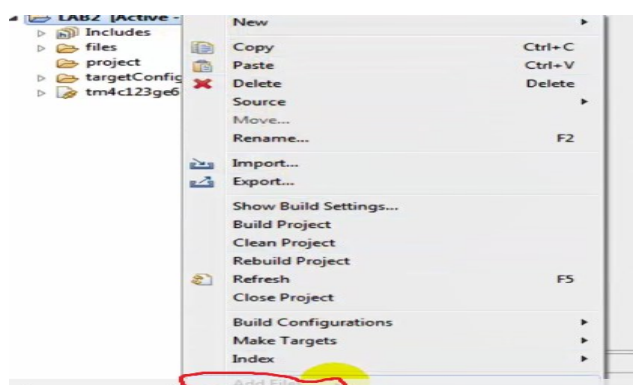


- (2) 同样新建一个名为 TIVAWARE_INATALL 且路径在 TivaWare 下的 Bulid Variables。

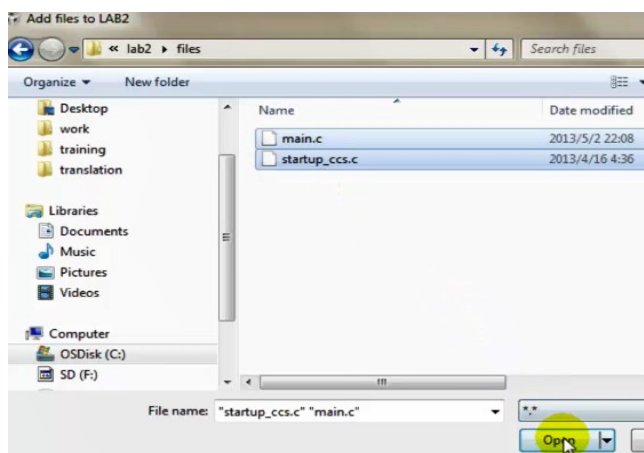
1.4 新建一个函数源文件

(1) 要为项目创建文件，请在“C/C++ Projects (C/C++项目)”（路径：工具栏 Window->Show View->Other 下的 C/C++ ->C/C++ Projects）视图中右键单击项目名称，并选择“New->Source File (新建->源文件)”。在打开的文本框中，键入包含与源代码类型对应的有效扩展名（.c、.C、.cpp、.c++、.asm、.s64、.s55 等）的文件名称。单击“Finish (完成)”。

(2) 要向项目添加现有源文件，请在“C/C++ Projects (C/C++项目)”选项卡中右键单击项目名称，并选择“Add Files to Project (将文件添加到项目)”，将源文件复制到项目目录。也可以选择“Link Files to Project (将文件链接到项目)”来创建文件引用，这样可以将文件保留在其原始目录中。如果源代码将文件包含在非常特定的目录结构中，则这是十分必要的。



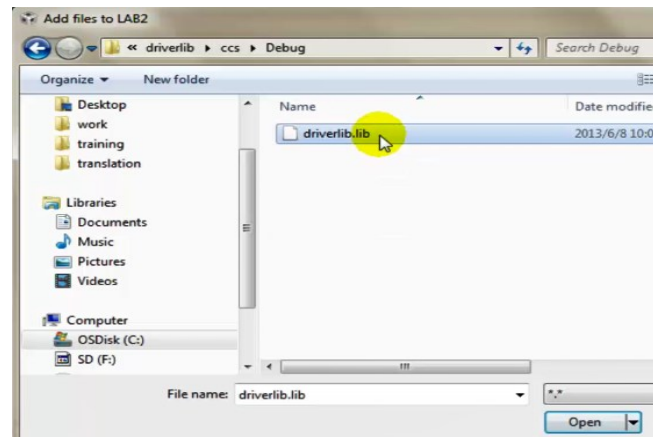
(3) 在出现的对话框中，选择文件，以 Lab2 为例，文件位于 TM4C123G_LaunchPad_Workshop 中 Lab2 下。



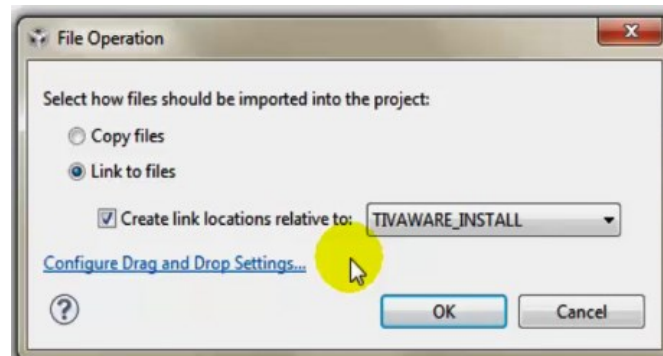
(4) 出现添加文件的对话框，由于文件是项目中要使用到的文件，所以一般选择 Copy files。

1.5 拷贝 driverlib 库的标准文件

(1) 将与 Tiva C 芯片外设相关的外设驱动库 Lib 文件添加到 Project 中,在 TIVAWARE 下的这个路径下，如 C:\ti\TivaWare_C_Series-2.1.4.178\driverlib\ccs\Debug。

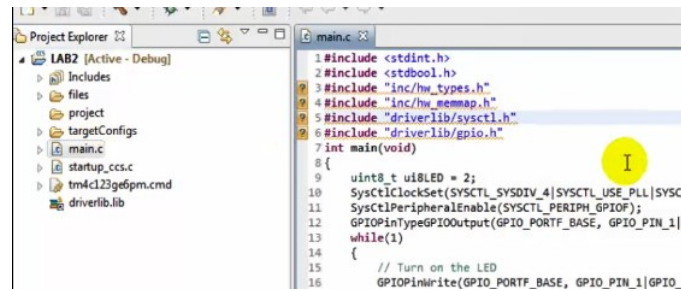


(2) 同样出现添加文件的对话框，由于驱动库文件仅被使用不会被修改，且驱动库文件不仅是为当前项目所使用，因此选择 Link 文件。选择相对地址来寻找 Lib 文件，用刚才添加的新的 TIVAWARE_INSTALL 的相对路径来寻找此 Lib 文件。

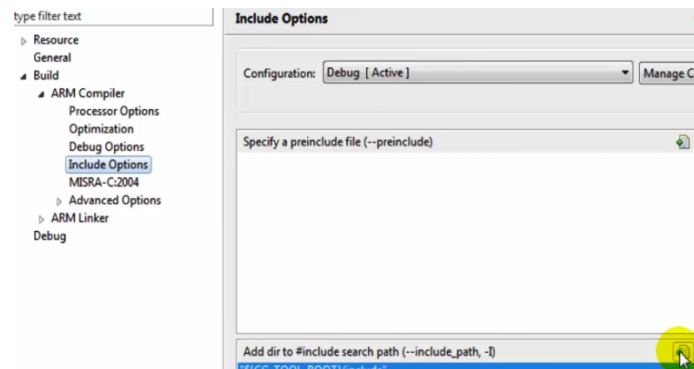


1.6 设置头文件 (include) 的搜索路径

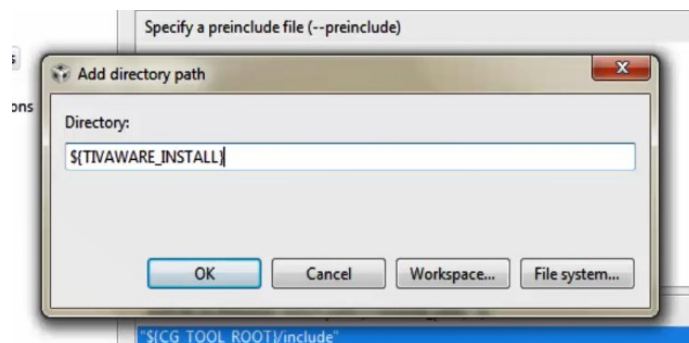
(1) 双击打开 main.c 文件，在文件的 include 命令行中出现了问号，这是由于项目并不知道从哪里找到这些头文件，因此需要设置路径告知编译器这些头文件的位置。



(2) 单击鼠标右键选择 Properties, 点击 Build 下 Include Options, 点击 + 号



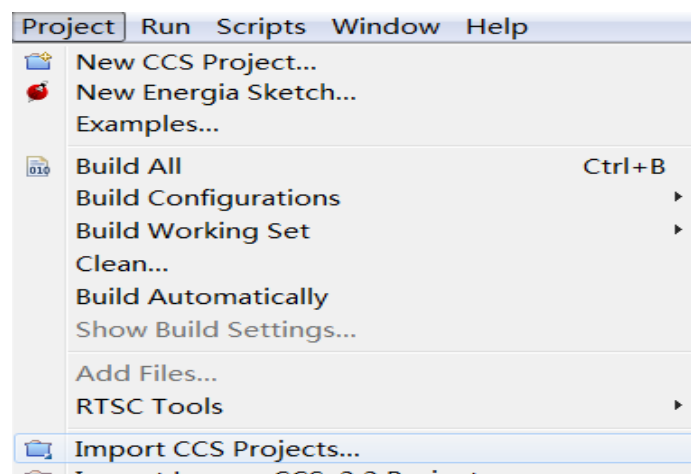
(3) 在出现的添加目录路径对话框, 加入带问号头文件所在的路径地址, 编译器就能找到这些头文件。添加头文件有两种方法, 一种是使用 File system 来找到位于 C 盘中的 TIVAWARE 然后找到这些头文件, 这种方法添加的路径是绝对路径, 若修改了 TIVAWARE 所在的目录, 项目就需要随之做出修改, 否则编译器找不到被修改路径的 TIVAWARE 头文件的地址; 另一种方法是使用相对地址, 用 TIVAWARE 的相对地址来指向目录, 这种情况下一旦修改了 TIVAWARE 的安装目录, 只要修改 TIVAWARE_INSTALL 的 Path Variables 变量所指向的地址, 那项目就不需做任何修改仍然能找到这些头文件。



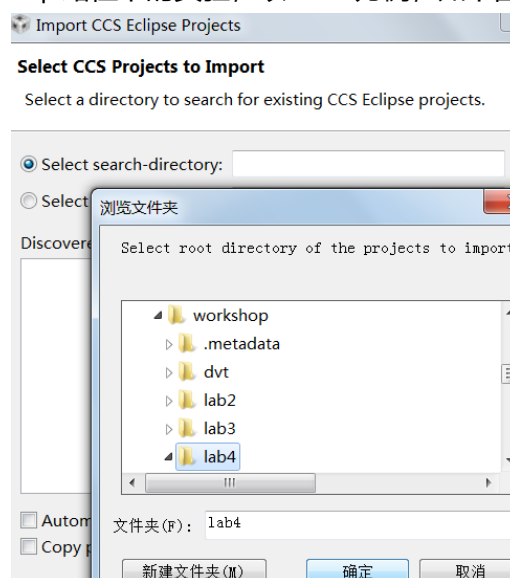
(4) 再次回到 main.c 文件, 之前的问号就消失了。因为编译器经过检查找到了这些头文件。可以进行调试编译了。

2 导入已有工程

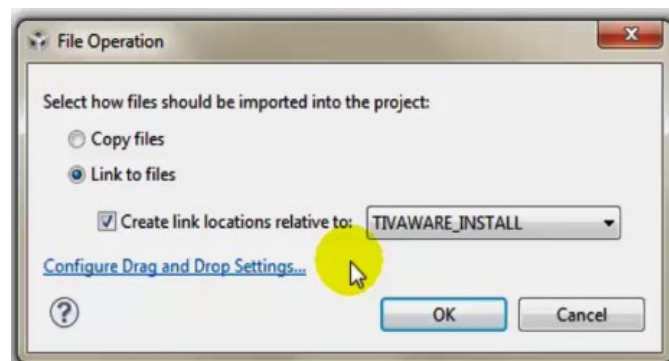
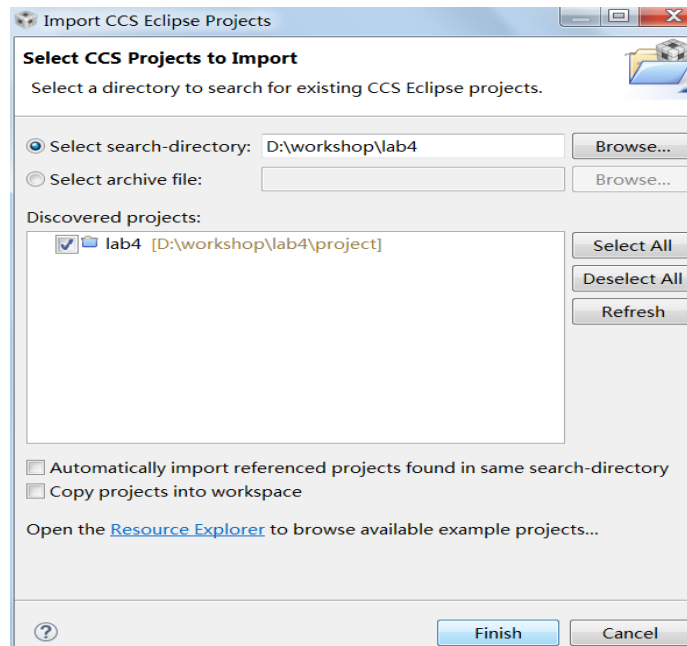
- 1) 点击 Project-Import CCS Projects, 如下图:



- 2) 选择下载 workshop 路径下的实验, 以 Lab4 为例, 如下图:



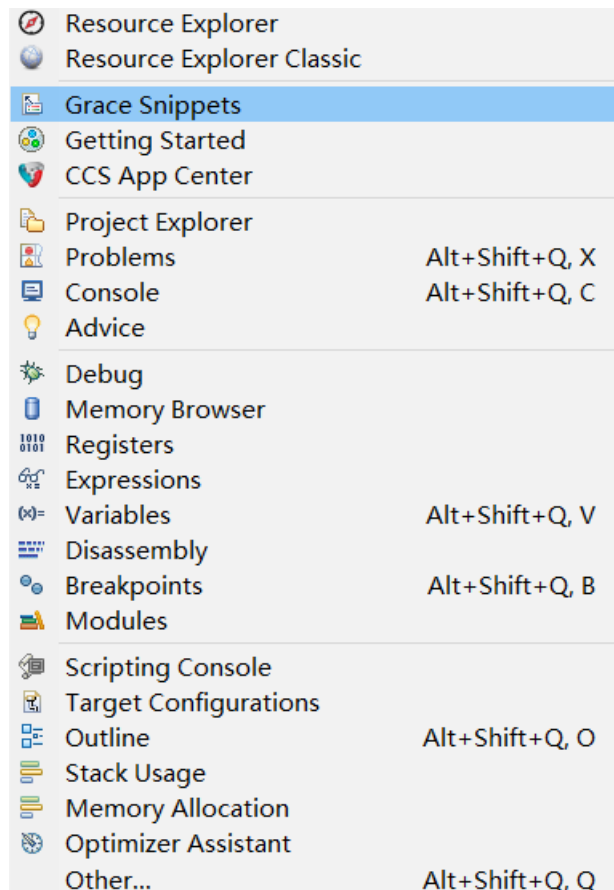
- 3) 确定-Finish, 已有工程成功导入。



3 编译及调试

3.1 窗口介绍

在菜单 VIEW 下，可以调出下列窗口。



Project Explorer: 显示打开工程文件窗口

Outline: 显示当前文件所有变量、宏定义、函数等，需要查找时非常方便

Problems: 如图所示，在点击 build 了之后，若有问题，会在此处窗口显示，然后双击错误处，CCS 可以定位到提示错误的地方

Console: 文件 build 成功了就会提示 “Finished building target: xxxxxx.out”，下一步就可以烧录了。

Advice: CCS 对于你的程序给的建议。

Modules: 显示当前的烧录文件: xxxx.out

Registers: 显示所有的寄存器的值，可以在这里看到程序是否按照你配置的寄存器值去配置，这里看的寄存器配置的值是最正确的。

Expression: 查看变量的数值与改变。

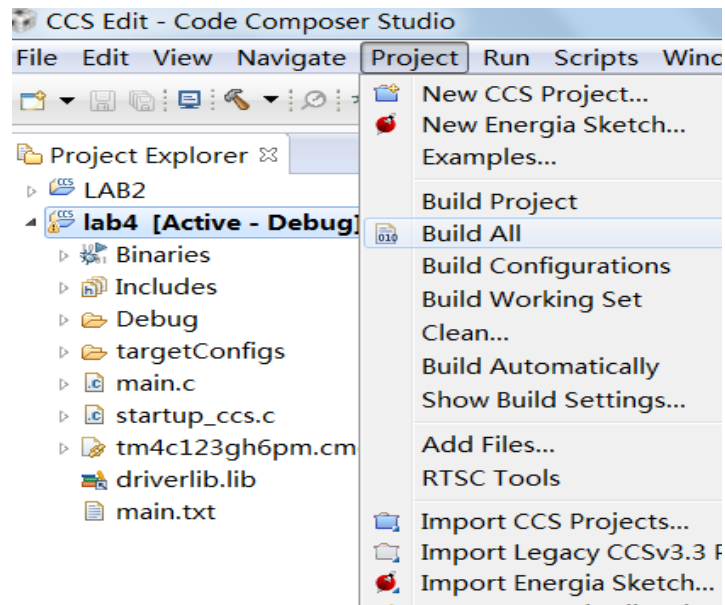
Memory Brower: 显示存储器的值，可以根据地址查看存储器的值。

Breakpoints: 显示文件中所有设置的断点，可以批量处理断点，进行单步调试的时候很方便快捷。

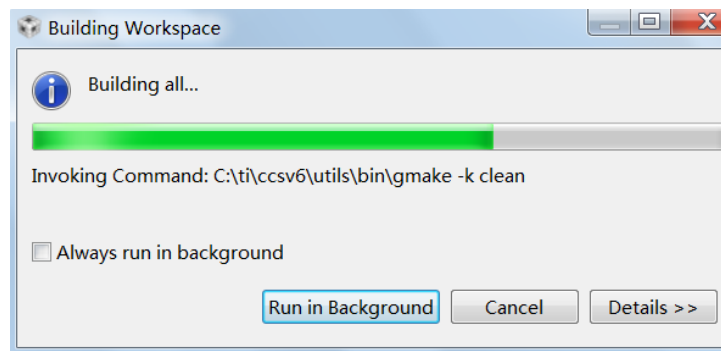
Disassembly: 反汇编窗口，显示现在文件单步调试时，每一行代码对于汇编程序，可以看看汇编程序是怎么样走的，逻辑对不对。

3.2 编译

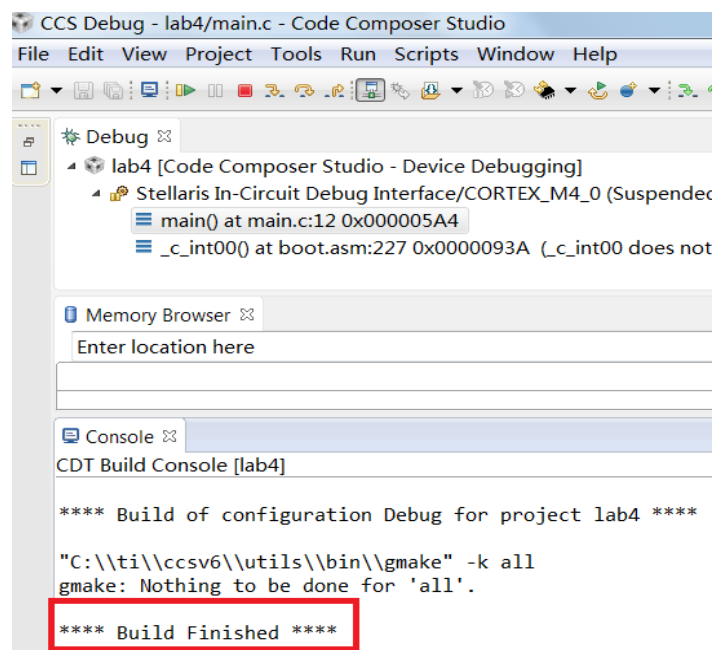
- (1) 所有设置完成后，就是对源文件进行编译了 Project-Build All，如下图。



- (2) 点击 Build All，编译所有源文件，出现如下对话框：



- (3) 编译结束后，会在 Console 窗口显示编译结果信息。



3.3 调试

调试程序就是发现程序中的错误并且修正这些错误。调试的基本方法是跟踪程序的运行，观察程序执行状态以及中间结果，来发现错误。例如：让程序逐条语句执行，观测每条语句的执行结果，检查结果是否正确，从而找到出错的位置。调试一般通过下面的操作来实现：

1) 控制程序的执行


程序执行的控制是指让程序运行到感兴趣的语句（指令）处并停下来，因为只有将程序停下来才能实现对程序执行中间过程进行观测，方法包括对程序的**单步执行**控制和设置**断点**。

值得注意的是，在具体程序的单步执行过程中，一条 c 语句可能对应若干条汇编语句，所以在具体的调试过程中应尽量选择最低的调试优化级，这样可以比较直观地观察到整个程序的运行状态。

2) 检查程序运行和处理器的状态

在调试器中可以对处理器内部的寄存器进行读 / 写操作，可以看到并修改这些寄存器的值。调试器提供对系统存储器的访问，可以查看并改变特定存储器地址的内容，这对系统的调试是非常重要的、可以实现对局部变量和全局变量的变化进行跟踪。

3.3.1 进入调试

选择菜单 Run-Debug (F11)，或者图标  会进入调试模式，**注意：在调试之前需将开发板连接到 PC。** 进入调试模式会将代码烧写到 LaunchPad 上。

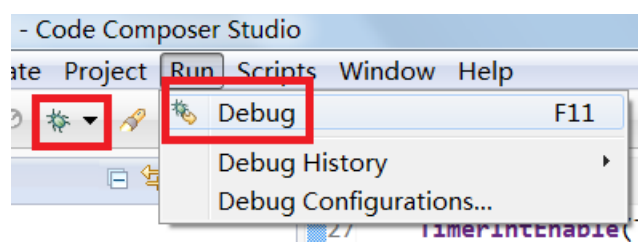



图 进入调试



图 调试界面

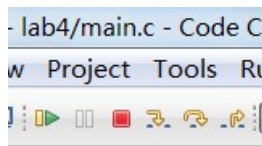
点击下图中的第一个按钮 Run ，程序会开始运行。

点击按钮 Suspend ，会暂停程序的运行。

单击按钮 Restart ，会重新定位到 main() 函数。


单击按钮 CPU Reset ，会复位微控制器。


可通过 Terminate  按钮返回到编辑界面。




3.3.2 单步运行

单步(step)是每次执行一条语句，然后暂停。调试器一般都会有好几种单步模式。



进入式单步(Step Into) ：每次运行一条指令，当遇到函数时，将进入函数中逐条执行。

跨越式单步(Step Over) ：每次运行一条指令，当遇到函数时，直接把整个函数一步执行完。

跳出式单步(Step Return) ：在调试过程中，有时会进入某个函数内执行。当执行完函数的一部分时，可能不再关心该函数其它部分的运行情况，这时就可以使用跳出式单步，一步执行完函数剩余部分，直接返回到调用该函数的位置。

3.3.3 设置断点

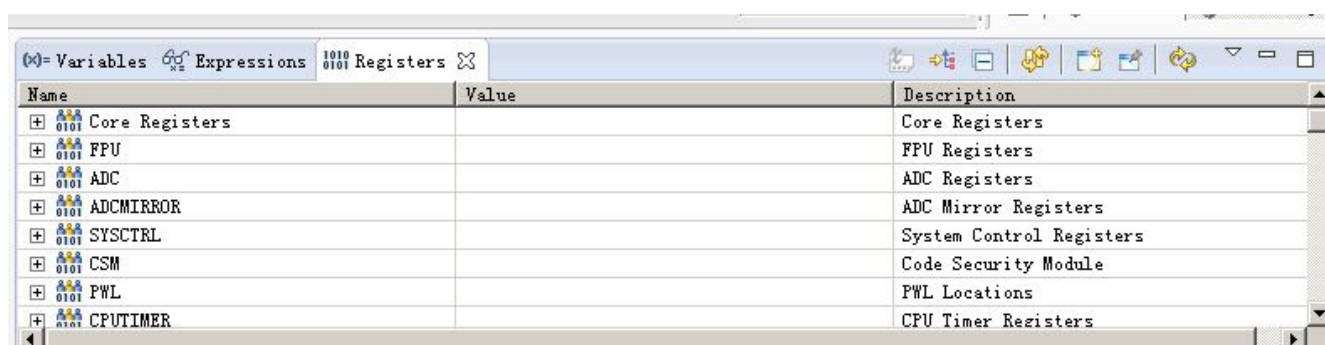
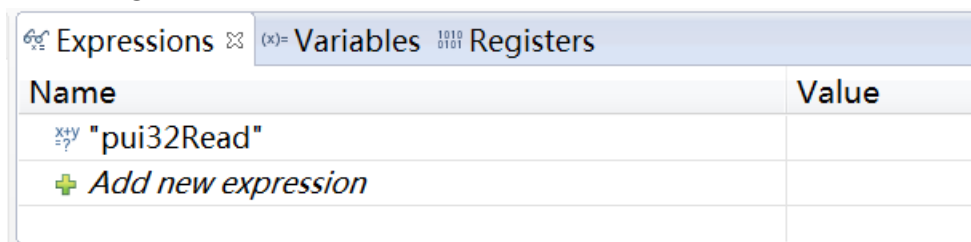
在程序调试的过程中，可通过设置断点来调试程序。在需要的地方设置断点，当程序运行到此处时，程序会暂停运行（执行挂起）。

选择需要的位置，双击代码行即可设置断点，或者点击按钮 。断点设置成功后将显示图标 ，可以通过双击该图标来取消该断点。

注意：在代码中，有时无法将断点设置到 C 源代码中确切的某一行。同样是因为一条 C 语句可能对应若干条汇编语句，从而无法确定对应哪一条汇编语句。

3.3.4 查看变量和寄存器

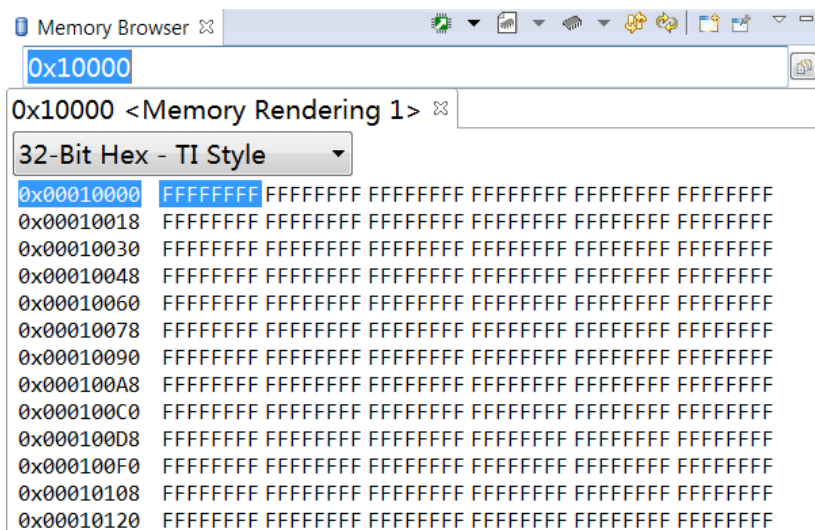
在程序加载时打开 “Variables”、“Expressions” 和 “Registers” 视图。Variables 可查看局部变量，Expressions 查看全局变量，Registers 查看寄存器，寄存器包括内核寄存器（Core Registers）和各外设的寄存器。



3.3.5 内存查看

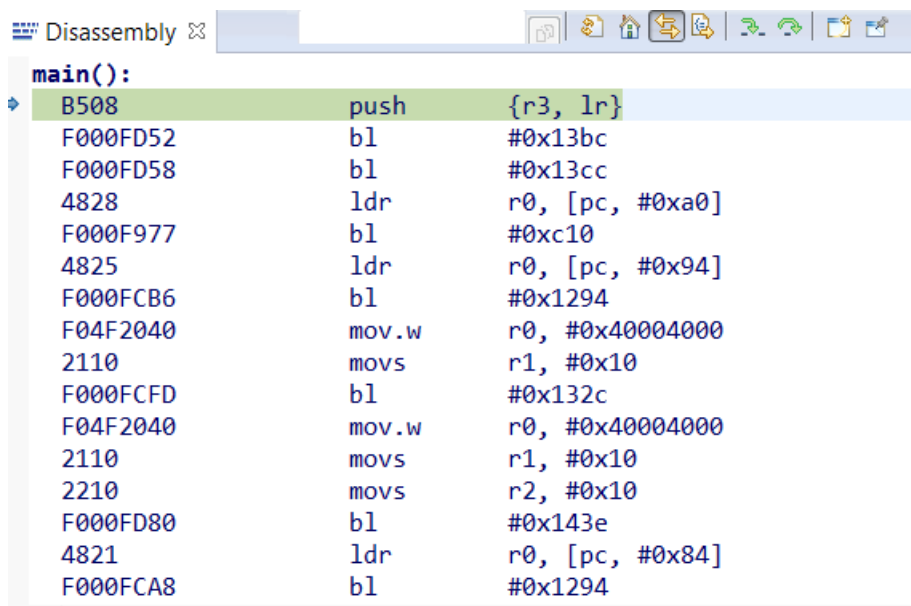
默认情况下不会打开内存视图，但是可通过选择菜单 “View->Memory Browser (查看->内存浏览)” 查看。

通过此屏幕可访问一些有用的功能：内存可通过多种格式进行查看，可填充任意值，也可保存至 PC 主机中的二进制文件或从中加载，此外还可以查看所有变量和函数，而且每个内存位置都有上下文相关的信息框，如下图所示：

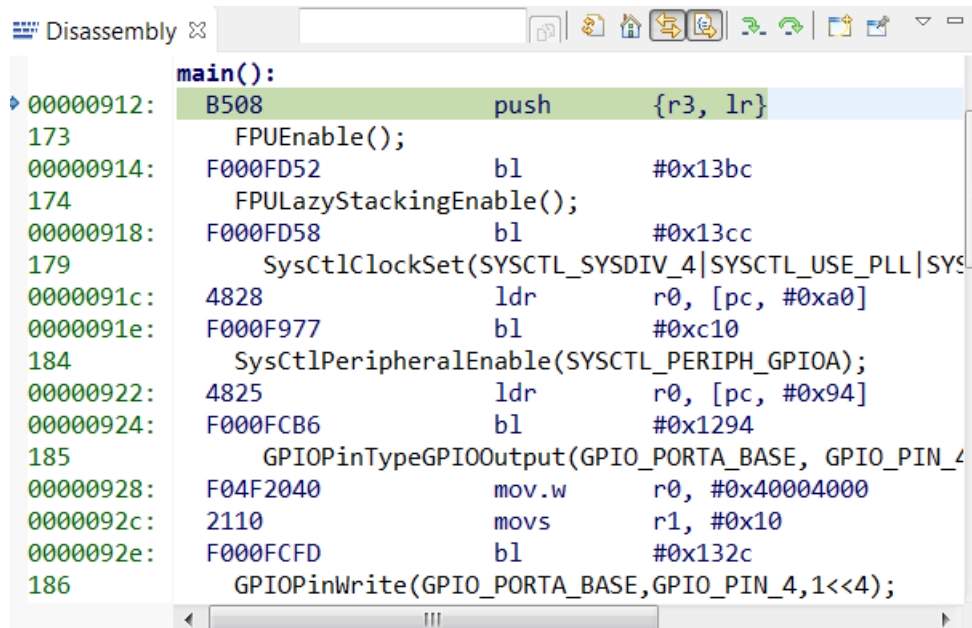


3.3.6 反汇编以及源代码与汇编代码混合模式

默认情况下不会打开反汇编视图，但是可通过转到菜单“View-> Disassembly（查看-反汇编）”查看，如下图所示：



反汇编窗口中一个极其有用的功能是源代码与汇编代码混合模式查看器，要使用此功能，只需在“Disassembly（反汇编）”视图中右键单击并选择“View Source（查看源代码）”，如下图所示：



```
Disassembly
main():
00000912: B508      push     {r3, lr}
173      FPUEnable();
00000914: F000FD52  bl      #0x13bc
174      FPULazyStackingEnable();
00000918: F000FD58  bl      #0x13cc
179      SysCtlClockSet(SYSCTL_SYSDIV_4|SYSCTL_USE_PLL|SYS
0000091c: 4828      ldr      r0, [pc, #0xa0]
0000091e: F000F977  bl      #0xc10
184      SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
00000922: 4825      ldr      r0, [pc, #0x94]
00000924: F000FCB6  bl      #0x1294
185      GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_4
00000928: F04F2040  mov.w    r0, #0x40004000
0000092c: 2110      movs     r1, #0x10
0000092e: F000FCFD  bl      #0x132c
186      GPIOPinWrite(GPIO_PORTA_BASE,GPIO_PIN_4,1<<4);
```

4 常见问题解决

把一些常见问题汇总如下，欢迎大家继续补充。

5.1 找不到 Tivaware

请先到官网下载 Tivaware 软件包，下载完成后安装到电脑上；最后，要记得把 Tivaware 导入 CCS。

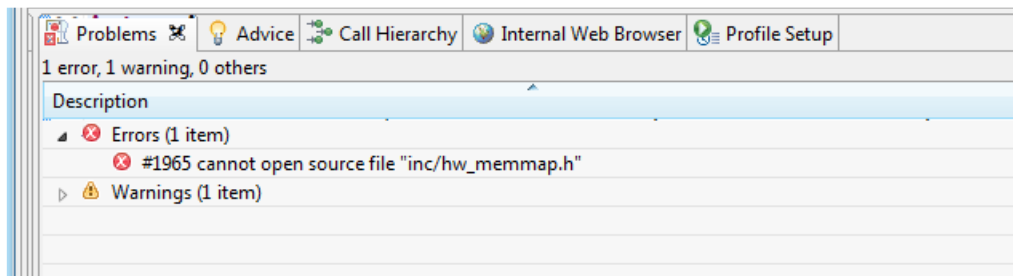
5.2 Tivaware 的函数定义

Tivaware 外设驱动库有哪些函数可以调用，每个函数的输入输出参数等说明，都可以参考 Tivaware 外设驱动库用户指南 Tivaware™ Peripheral Driver Library for Series Users Guide (<http://www.ti.com/cn/lit/pdf/spmu298>)，同时，外设驱动库 DriverLib 的函数实现（即源代码）导入 CCS 中也可以直接查看。

5.3 编译时提示找不到头文件

在编译工程时，经常会出现找不到头文件的错误情形。错误现象描述：

- (1) Problem 窗口提示：“#1965 cannot open source file” inc/hw_memmap.h”



- (2) 在源代码也会提示 X 或者问号提示:

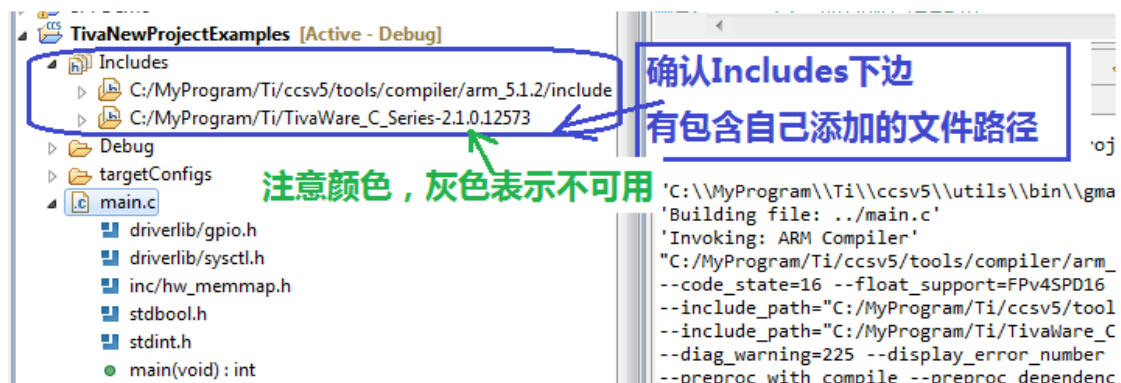


解决方法:

请参考 1.6 中的描述添加头文件路径;

如果头文件路径均包含正确:

- (3) 工程目录 Include 下包含有自己添加的文件路径:



- (4) 源代码包含头文件的地方, 前边不带问号:



5.4 找不到变量

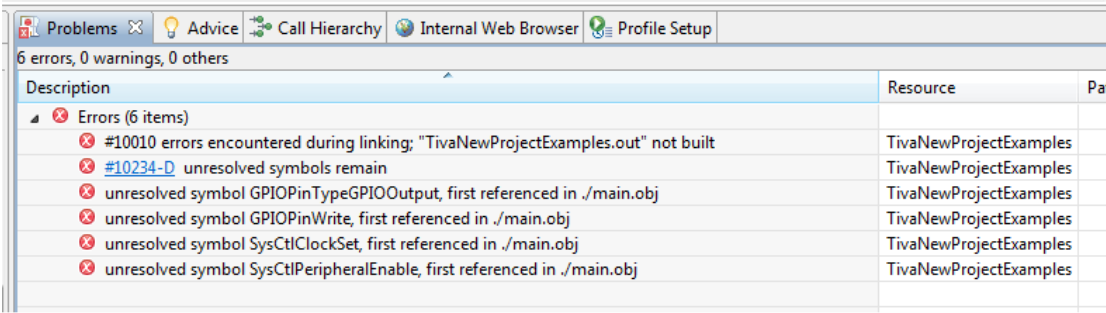
也是常见的问题，一般是因为没有包含头文件。具体要包含哪些头文件，要靠自己多练习了。

5.5 unresolved symbol 错误

错误现象描述：

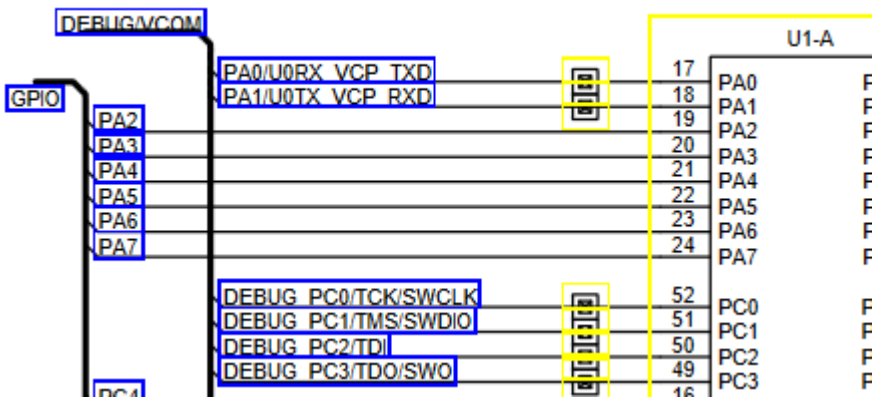
unresolved symbol GPIOPinWrite,first referenced in ./main.obj

unresolved symbol GPIOPinTypeGPIOOutput,first referenced in ./main.obj

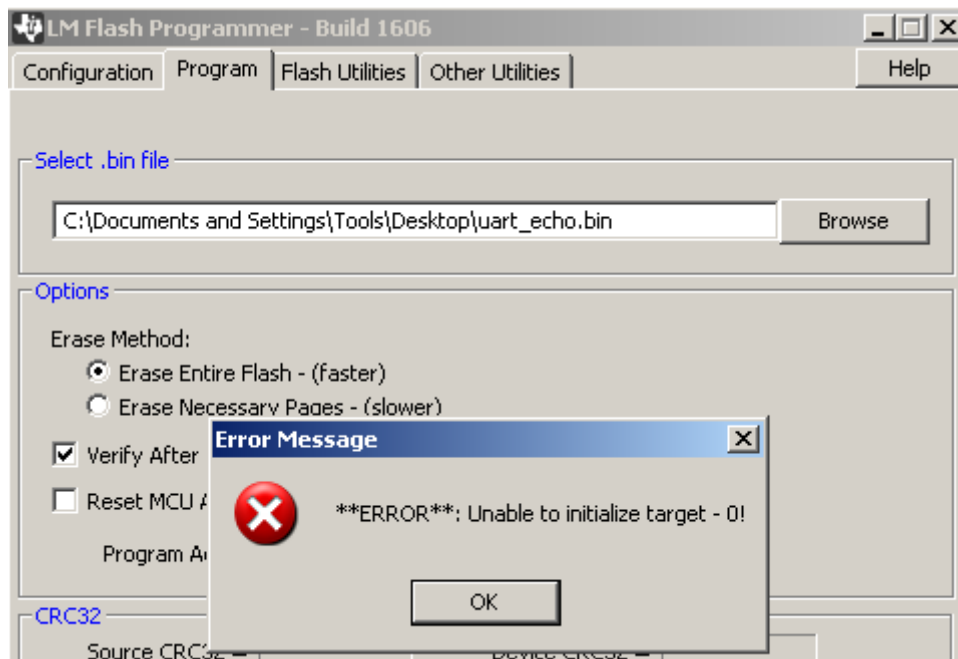


解决方法：只申明了函数，找不到原型。引用了 Tivaware 的 DriverLib 库函数，但是工程里边没有把库函数添加进来。

5.6 Flash Programmer 提示 Unable to initialize target



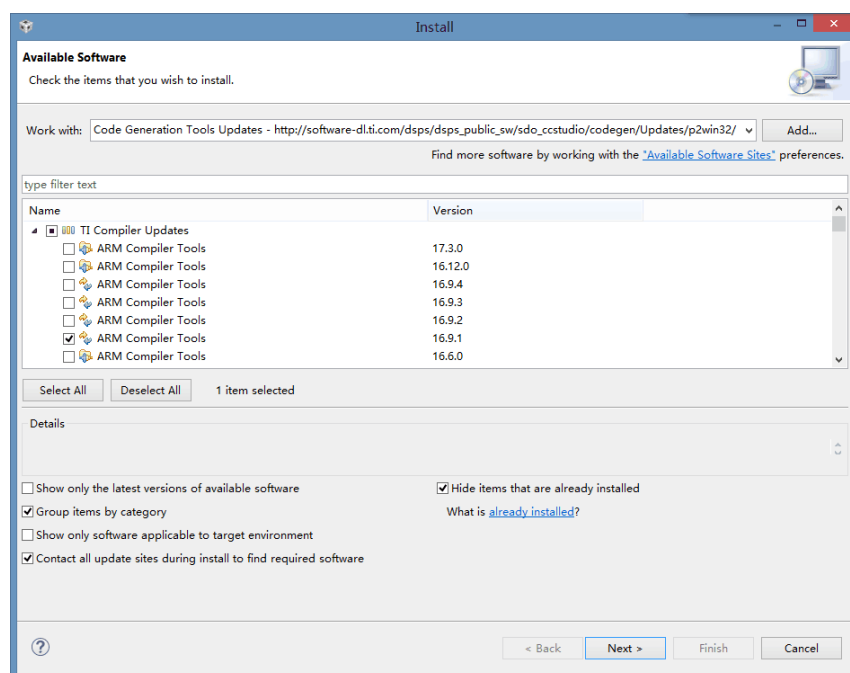
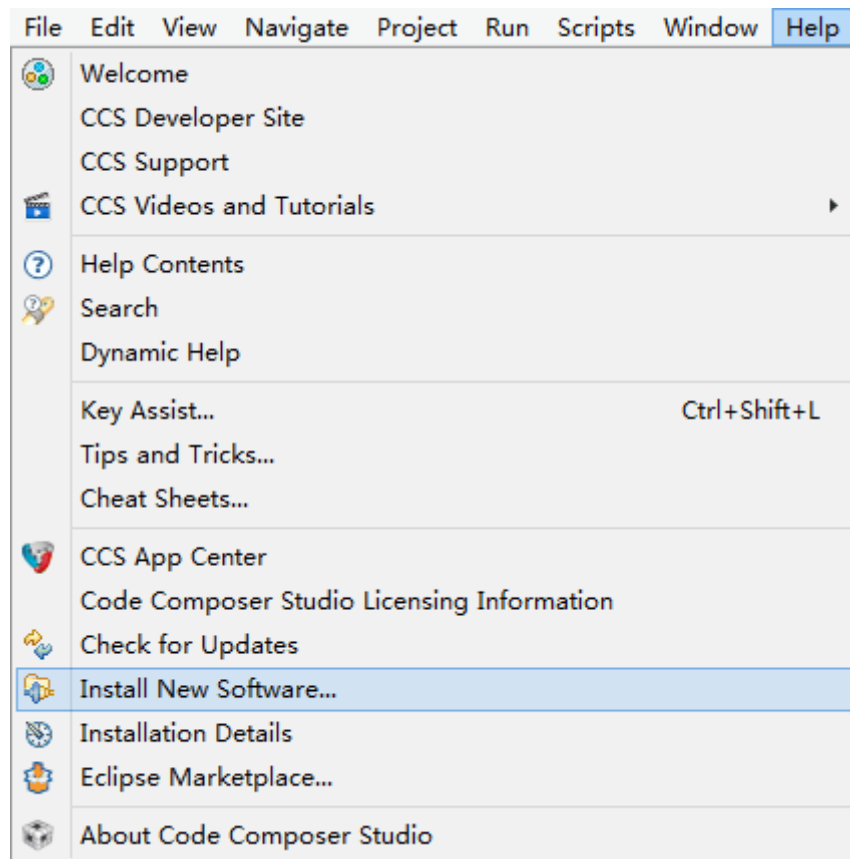
PC0~PC3、PA0、PA1 用于调试接口，如果编程时对这几个 IO 进行操作，会导致使用 Flash Programmer 烧录程序时提示：



解决方法是： 选择 Other Utilities --> Debug port unlock --> Fury, Dustdevil, TM4C123 and TM4C129 Classes, 点击 Unlock, 按窗口提示操作（断电，按住复位后上电，看到提示后松手，再根据提示重新上电）。

5.7 提示缺少 armv16.9.0

解决办法如下：



☐ Show only the latest versions of available software

去掉上图勾选项，就可以找到 16.9.0

Code Generation Tools Updates - http://software-dl.ti.com/dsp/dsp_public_sw/sdo_ccstudio/codegen/Updates/p2win32/

这个后面 win32 是 win 系统的，其他的还有 linux 和 mac 的注意区分。