

F28377D 在 CCS9.2 下的使用基础

序言

本教程旨在体现更普遍的反映高版本 CCS 与仿真器使用情况，**由于 F28377D 为最新的芯片，只有 CCS6.1.3 以上软件环境可以支持**，故这里使用 XDS100V3 仿真器和 CCS9.2 软件在 WIN7 和 WIN10 系统下演示完整的操作流程，**本公司推荐用户使用最新版本的 CCS 软件（我们提供了 CCS9.2 版本），功能更全，基本使用方法与教程中 CCS9.2 应用方法一致，可直接套用。**

注意在最新版本 CCS（CCS8.0 以上版本）上第二章“CCS 与仿真器连接”的第 3 节“添加注册文件”不需要做，新版本完全开放，不需要这一步。

目录

F28377D 在 CCS9.2 下的使用基础.....	1
序言.....	1
第一章 CCS8.0.0 软件的安装.....	4
第二章 CCS 与仿真器的连接.....	9
2.1 定义工作区目录.....	9
2.2 建立目标板配置环境.....	11
2.3 添加注册文件.....	14
2.4 连接目标板.....	15
第三章 创建 CCS 工程.....	19
3.1 创建工程.....	19
3.2 工程编写.....	21
3.3 生成项目.....	23
第四章 工程导入.....	25
4.1 CCS 低版本工程的导入.....	25
4.2 CCS 高版本工程导入低版本.....	27
第五章 CCS8.0.0 的仿真与烧写.....	29
5.1 F28377D 的仿真操作.....	29
5.2 F28377D 的烧写操作.....	32
5.3 F28377D 的双核烧写操作.....	33
第六章 CCS8.0.0 中一些常见的功能按钮.....	36
6.1 加载代码.....	38
6.2 监视变量和寄存器.....	38
6.3 反汇编以及源代码与汇编代码混合模式.....	39
6.4 内存查看器.....	40

6.5 管理断点.....	41
6.6 图形显示工具.....	43
6.7 图像显示工具.....	45

第一章 CCS9.2软件的安装

首先我们需要来安装 TI DSP 的软件开发环境 CCS (Code Composer Studio)。由于 CCS7.0 以后 TI 开放了 CCS 软件, 且为 WIN10 专门做了适配, 为了方便, 我们推荐使用 CCS7.0 以上版本, 例程配套使用的是 CCS9.2。当然, DSP28377D 也支持 CCS6.1.3-CCS7.0 之间的版本, 不过 CCS7.0 以下版本需要破解, 这里为了教程的普遍性, 我们安装 CCS8.0 版本进行演示, 9.2 与之类似。

注意: 安装前先关闭杀毒软件和 360、电脑管家等安全防护软件, 否则点击安装程序会出现警告, 强行安装会出现文件丢失。

软件安装步骤:

1、双击 ccs_setup_8.0.0.00016.exe 文件, 在弹出的对话框中选择 “I accept the terms of the license agreement”, 然后点击 “next”, 如下图 1-1 所示:

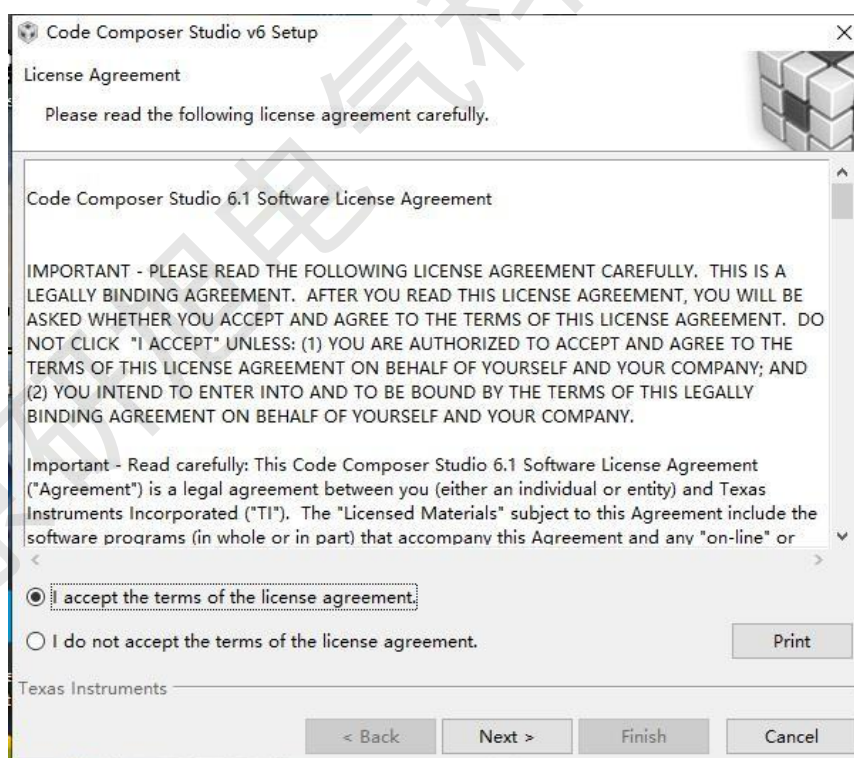


图 1-1

2、点击 “Browse” 可以自行选择安装路径 (**注意: 路径不可以有中文**), 点击 “Next” 见下图 1-2 所示:

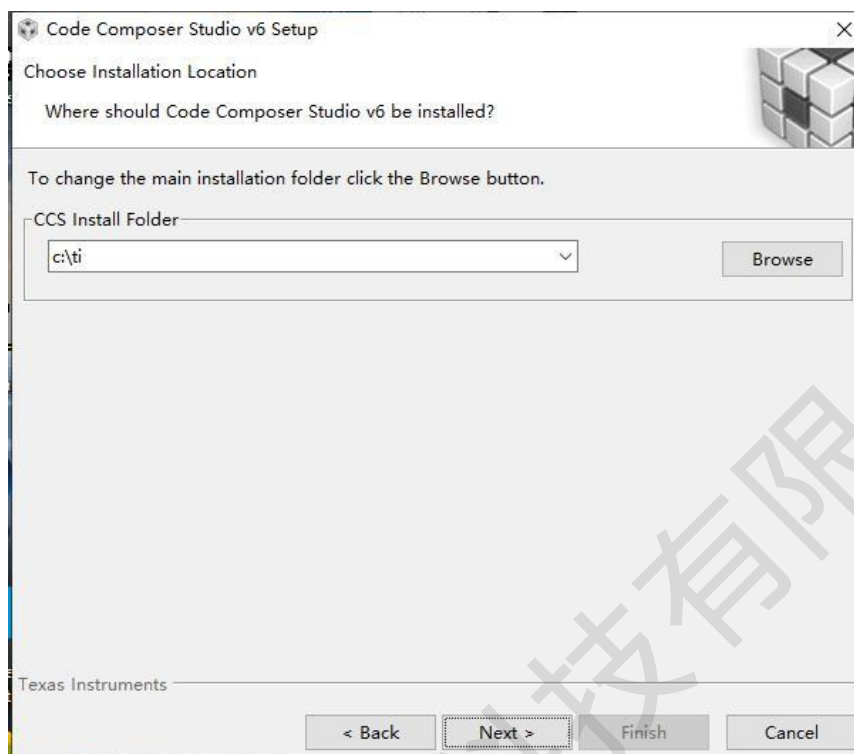


图 1-2

3、用户可根据自己的需求选择所要安装的内容，建议直接选择“Select All”，点击“Next”见下图 1-3 所示：

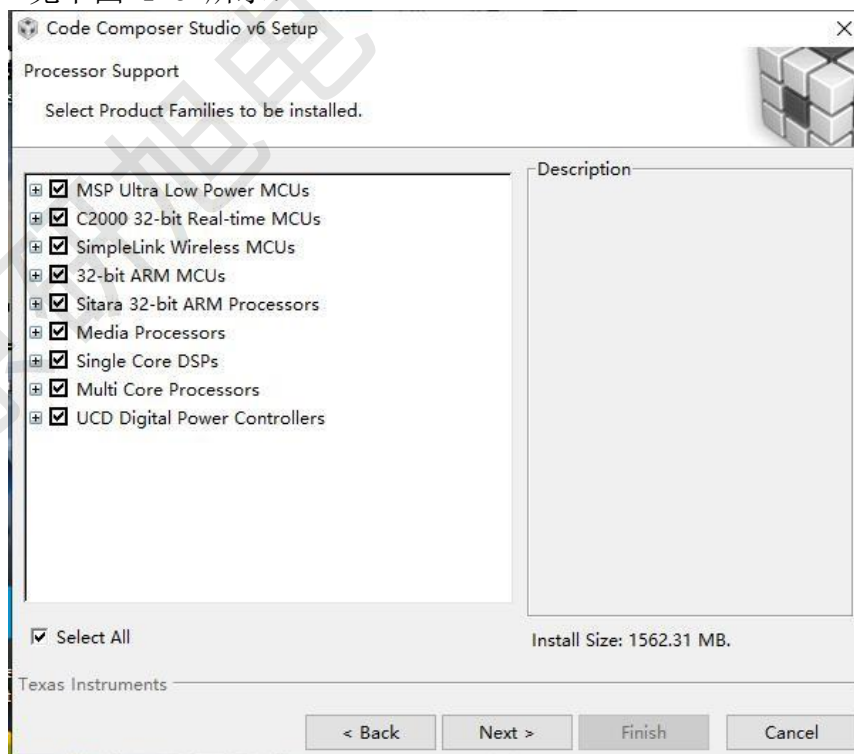


图 1-3

4、依旧根据自己的需求选择仿真设备驱动类型，这里选择“Select All”，然后点击“Next”见下图 1-4 所示：

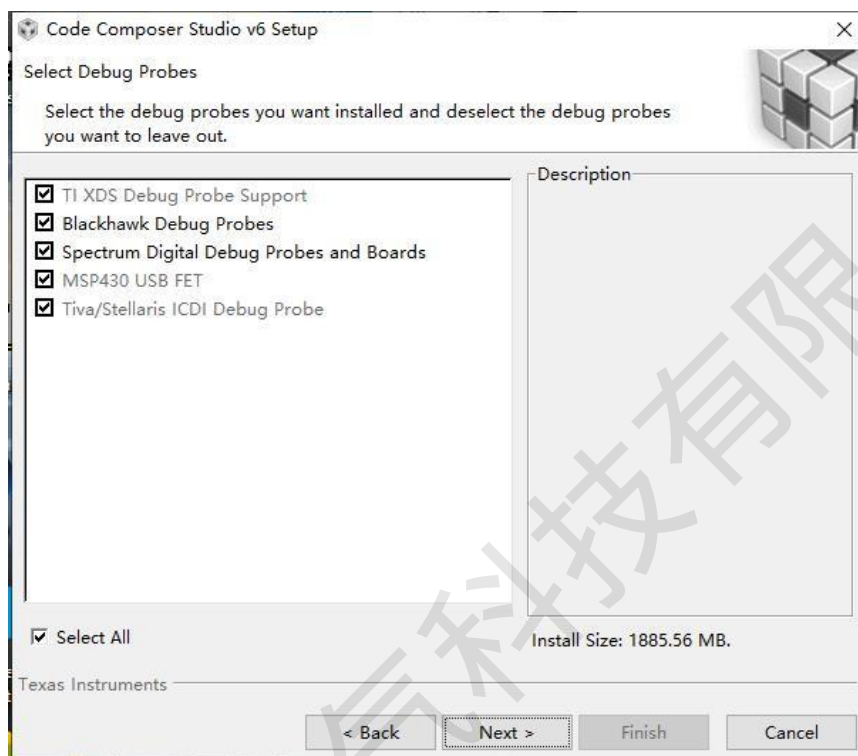


图 1-4

5、根据自己的需求选择，这里依然全选，然后点击“Finish”，见下图 1-5 所示：

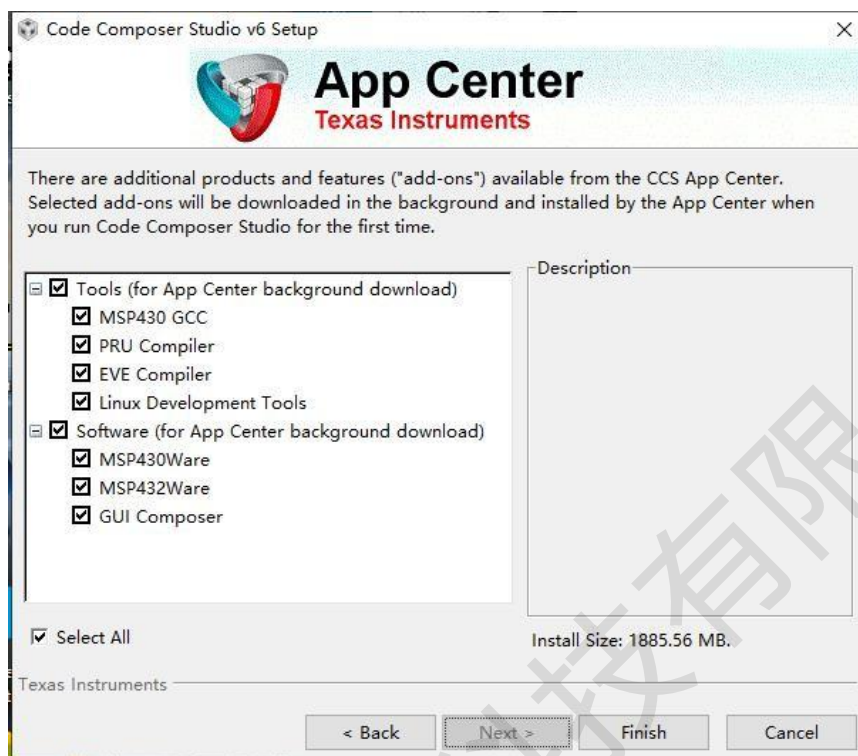


图 1-5

6、安装过程中会弹出如下图 1-6 所示一些安装功能的对话框，请勿单击 Cancel 按钮，否则在安装过程中就不会安装此功能。

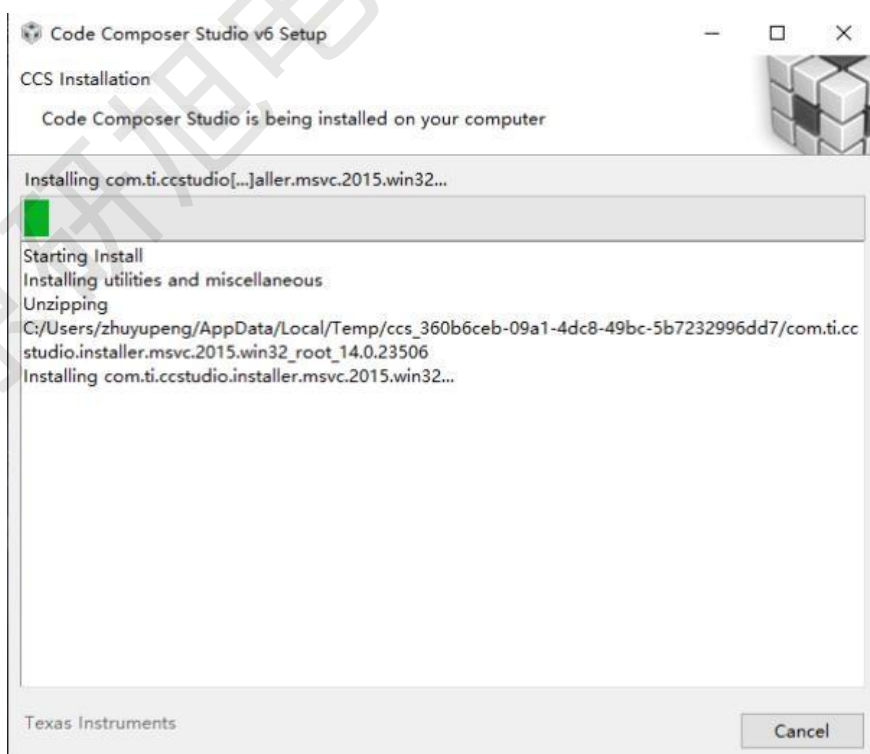


图 1-6

7、安装完成，点击“Finish”见下图 1-7 所示：



图 1-7

第二章 CCS 与仿真器的连接

由于 XDS100V3 仿真器的驱动已经做到了 CCS9.2 中了, 所以当用户安装 CCS9.2 版本的软件之后, 仿真器的驱动就已经安装好了。接下来, 用户只需要将仿真器的 USB 与 PC 机的 USB 接口连接即可。驱动也是自动识别安装的, 当提示驱动安装完毕在**设备管理器**下正确识别后, 用户就可以使用仿真器对目标板进行仿真操作。



图 2-0

下面我们以 YX-28377D 一体板为例为大家说明如何使用 XDS100V3 仿真器对目标板进行仿真。

2.1 定义工作区目录

启动 CCSv9.2 首先要求的是定义一个工作区, 即用于保存开发过程中用到的所有元素 (项目和指向项目的链接, 可能还有源代码) 的目录。

打开软件后, 会有一个工作空间 workspace 的创建步骤, 建议放在 C 盘外的磁盘根目录下, 如 D:\workspace_28377, 点击“OK”确认。默认情况下, 会在 C:\Users\<用户>\Documents 或 C:\Documents and Settings\<用户>\My Documents 目录下创建工作区, 不建议选择默认, 路径过于复杂, 不利于后续工作。点击 Browse 可以任意选择其他位置。如下图 2-1-1 所示。每次执行 CCSv8.0 都要选择工作区目录 (默认情况下选择上一次使用的工作区)。

当 CCS 打开时, 如需切换工作空间, 可以点击“file->Switch Workspace->other”, 自行变更。左下角的英文不要勾选, 否则每次打开将默认第一次设置的路径, 直接跳过

这一步。如图 2-1-2 所示。

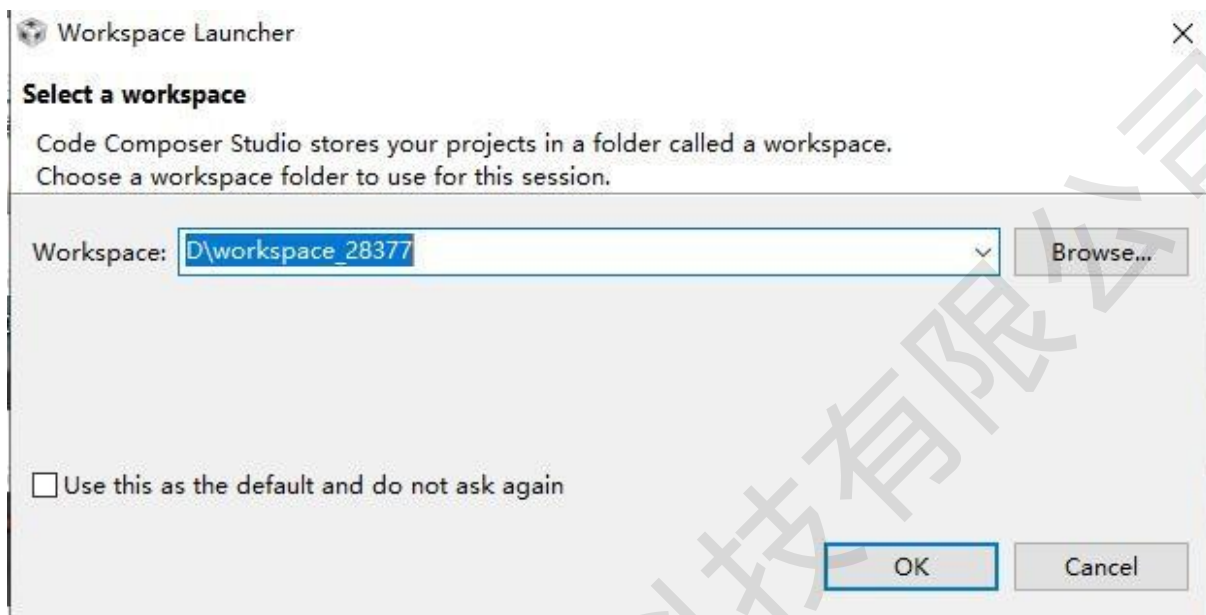


图 2-1-1

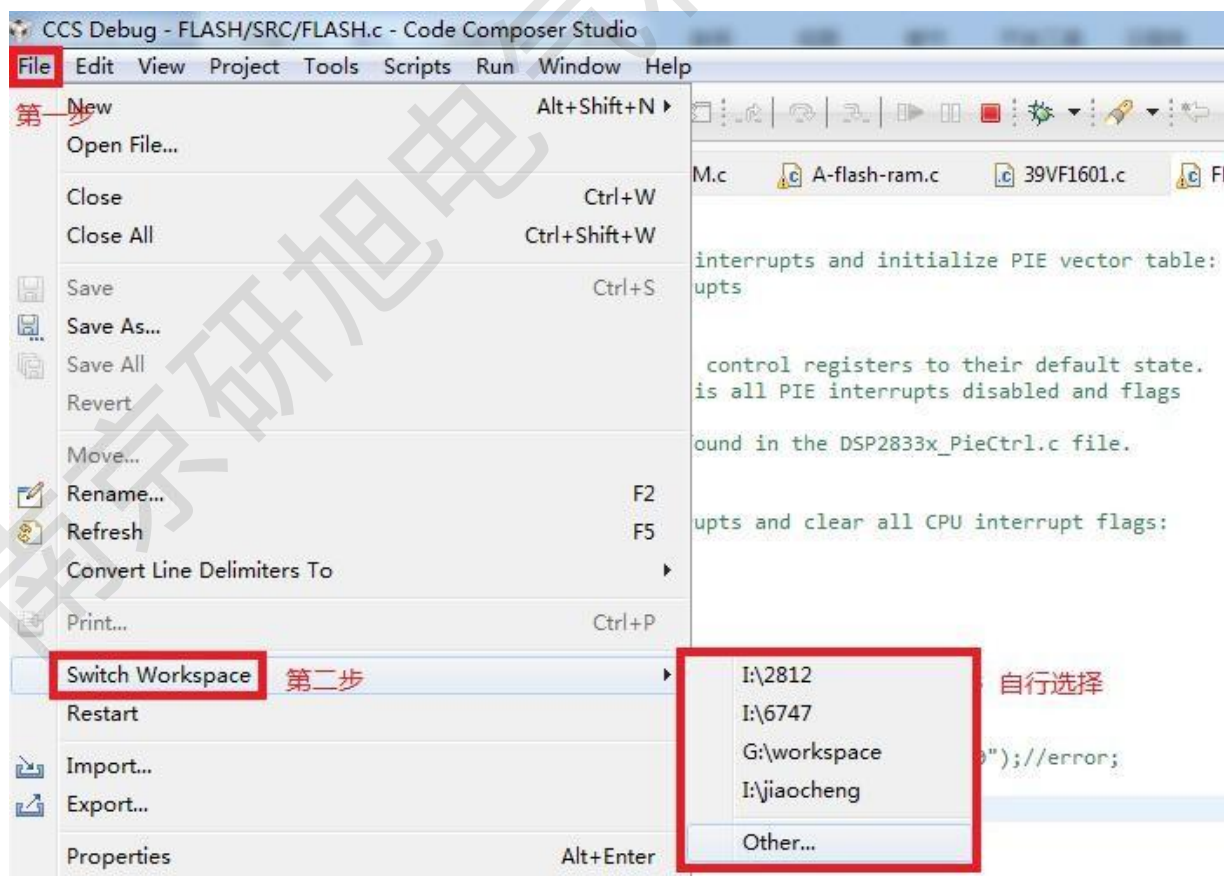


图 2-1-2

2.2 建立目标板配置环境

如果使用过 CCS 低版本的用户对此并不陌生，同样在 CCSv6.1.3 版本环境中，也需要建立仿真配置环境，只不过建立的人机界面与之前版本有所区别。在 CCS 低版本中，用 CCS SETUP 进行建立，在此不多做介绍，下面主要介绍如何在 CCSv8.0 中建立环境：

在工具栏中点击“File ->New->Target Configuration File”，如下图 2-2-1 所示：

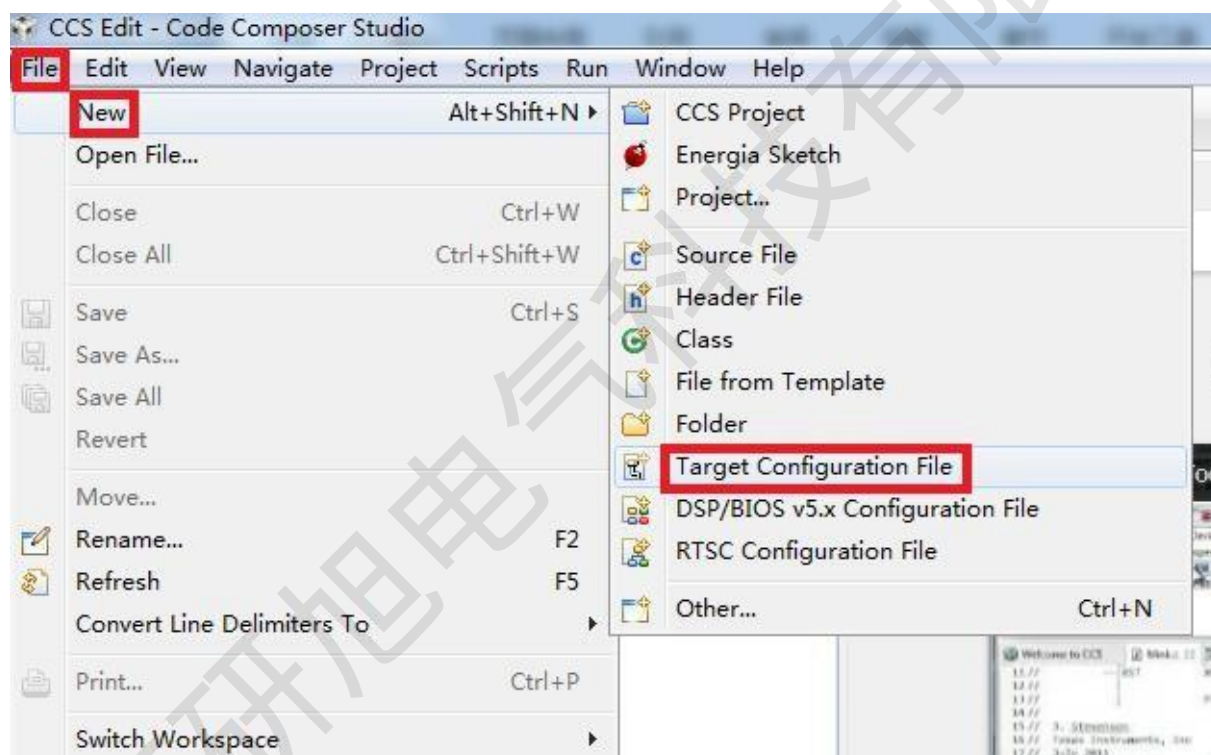


图 2-2-1

为此配置命名“f28377 xds100v3 .ccxml”（注：可自行命名，但后缀 .ccxml 不可更改），勾掉“use share location”用户可自行选择配置环境所要建立的位置，但是容易导致错误，此处要求选择默认位置，点击“Finsh”，如下图 2-2-2 所示：

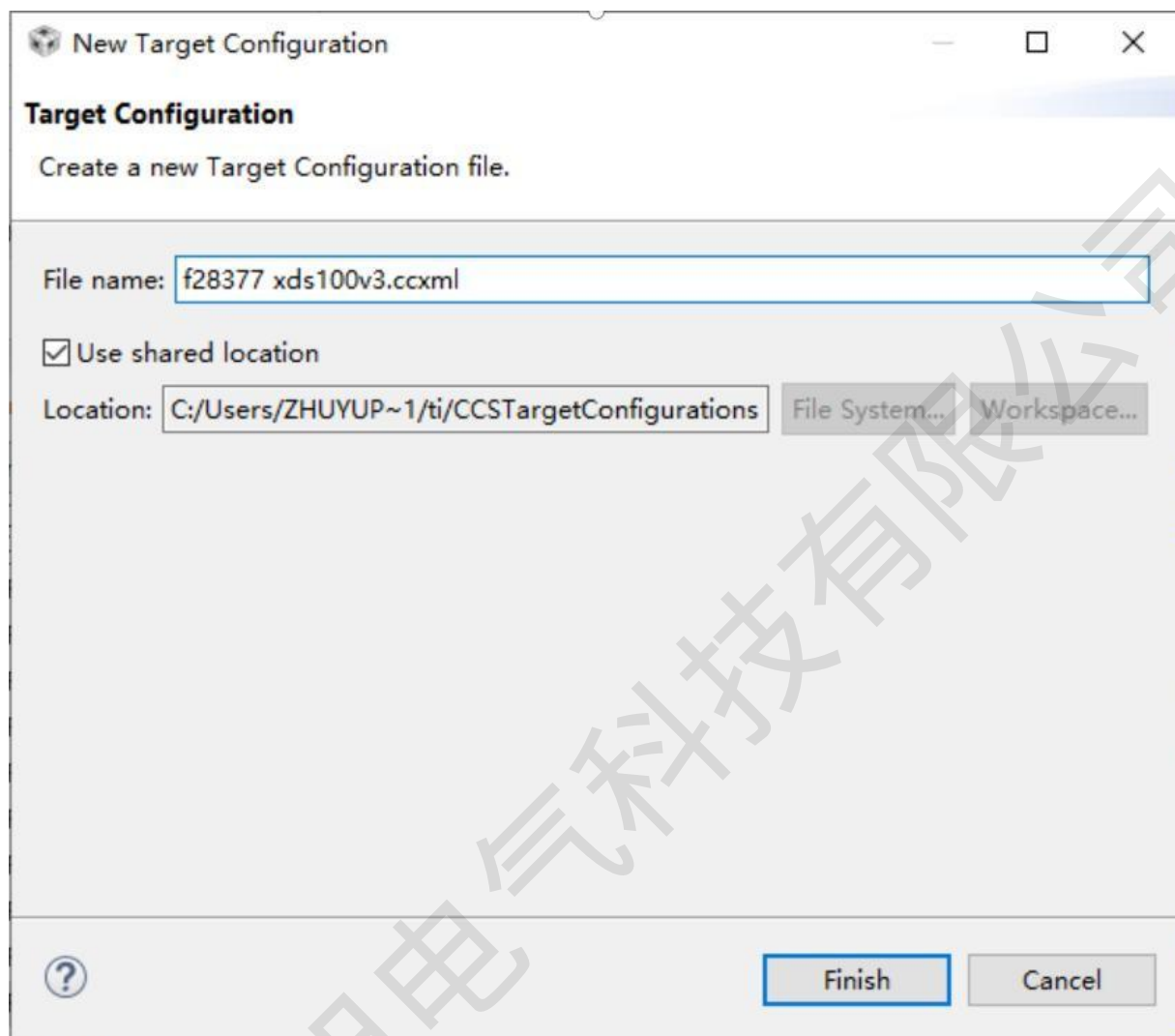


图 2-2-2

在“connection（调试器类型）”一栏中选择“Texas Instruments XDS100v3 USB Emulator”，“Device（芯片类型）”一栏中选择“TMS320F28377D”，如下图 2-2-3 所示：

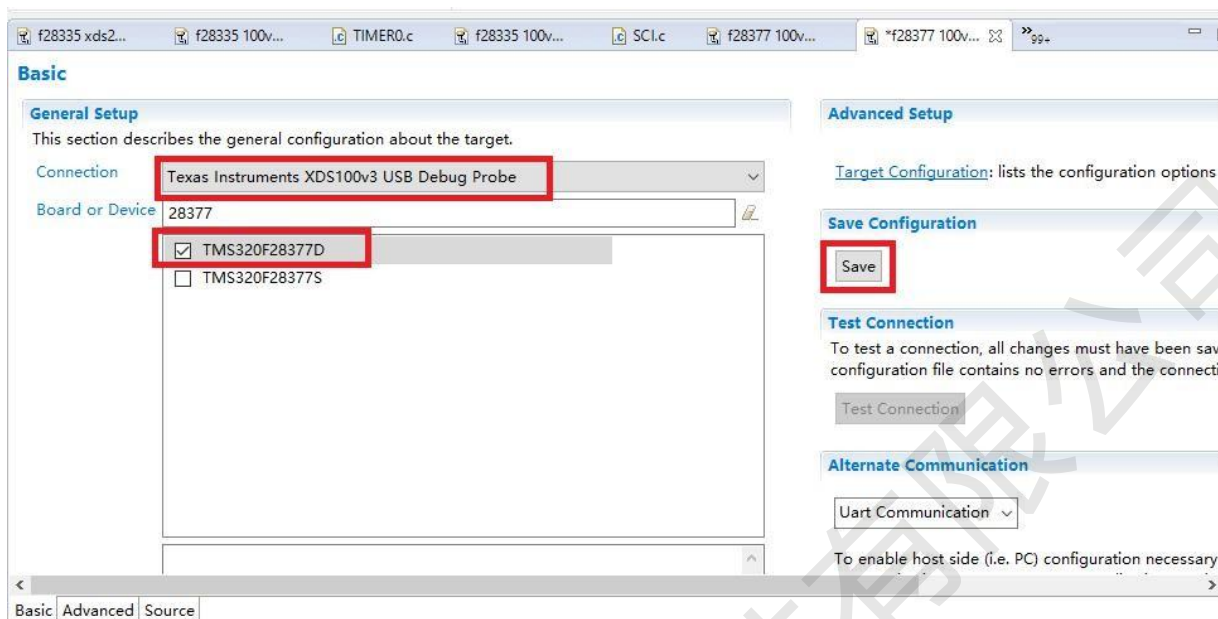


图 2-2-3

一般我们使用默认的 GEL 文件即可，直接点击“Save”。如果需要使用自己的 GEL 文件（GEL 文件主要是对芯片的初始化，一般不需要更改），那么点击蓝色字体“Target Configuration”就会出现如图 2-2-4 界面：

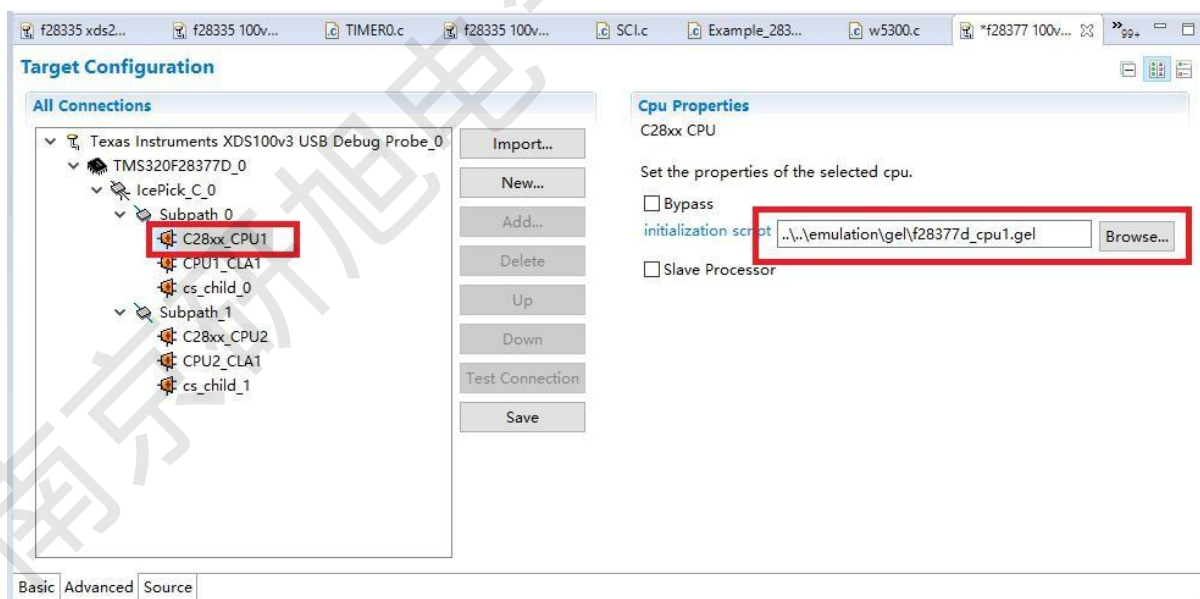


图 2-2-4

选中 C28xx，在右面的 initialization script 中选择自己的 GEL 文件，之后点击“Save”，到目前为止，此配置环境建立完毕。

2.3 注册文件升级

注意：CCS7.0 以上版本 TI 已经完全开放，无需添加该文件。

以下仅供CCS7.0以下版本参考，CCS9.2不用添加。

首先打开 CCS 安装目录，默认在 C:/ti 下，打开
ccsv6.1.3->ccs_base->DebugServer->license，将我们提供的 ccslicense.lic 文件拷贝到该文件夹下。（必须先补充注册文件，否则后面无法找到文件进行添加）

然后点击 CCS 软件顶部菜单 Help 选择下面的“Code Composer Studio Licensing Information”一栏，如下图 2-3-1 所示：

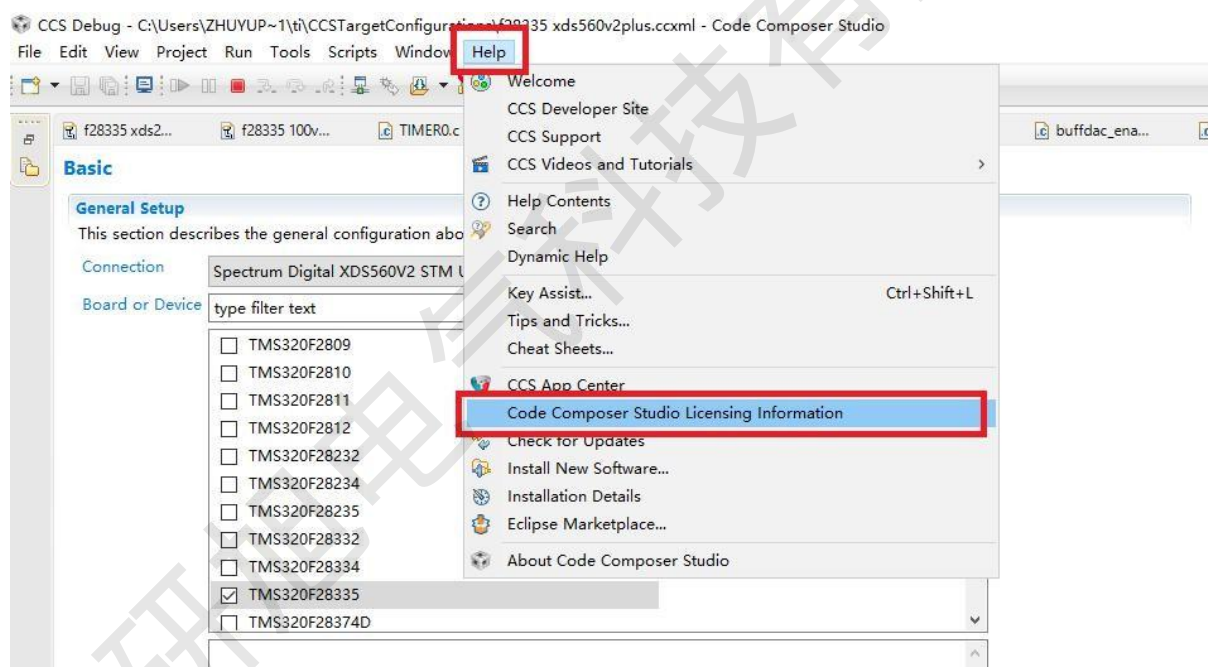


图 2-3-1

接着选择第一栏第三个菜单 Manage，然后点击 Add 添加 License 文件，路径为上一步复制到的路径，点击 OK 保存，如下图 2-3-2 所示（不需要勾选）：

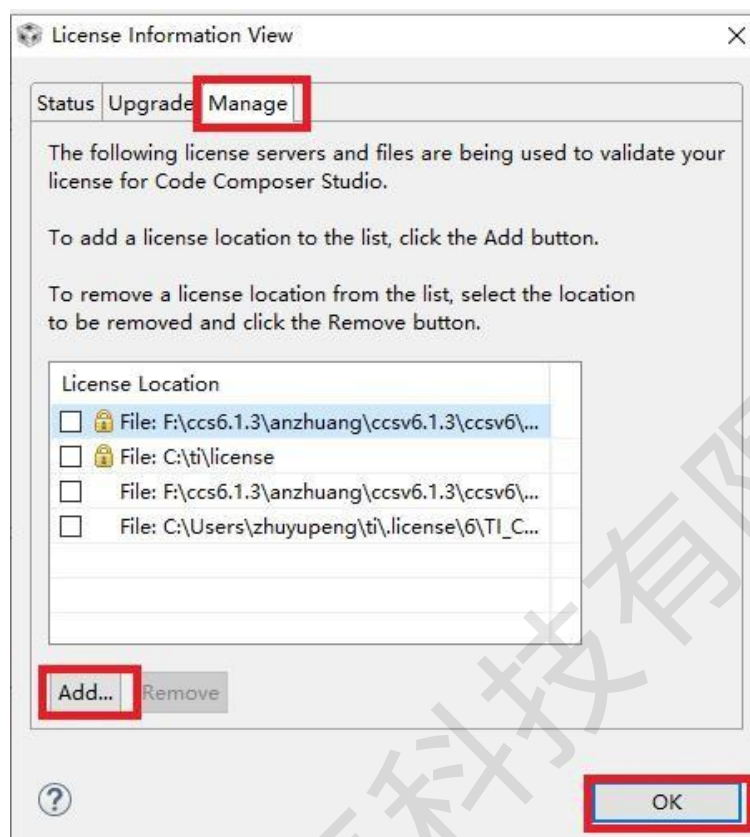


图 2-3-2

2.4 连接目标板

首先把仿真器通过 USB 线接到电脑上，然后另一端连接目标板，给板子上电，完成硬件连接。

接着找到工具栏的“View->Target configurations”按钮并单击切换出配置界面，然后在配置界面中右键点击配置文件选择“Set as Default”将刚刚建立的配置文件设置为默认状态,启动调试，右击选择已配置的项目的“Launch Selected Configuration”就会自动跳转到“CCS Debug”界面。详细操作见下图 2-4-1 所示：

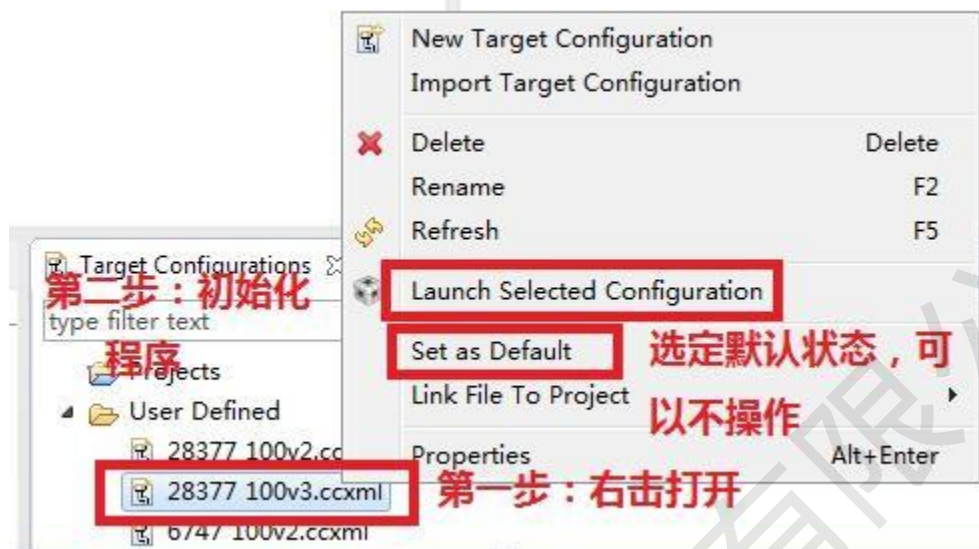


图 2-4-1

启动配置成功后弹出如图 2-4-2 所示界面:

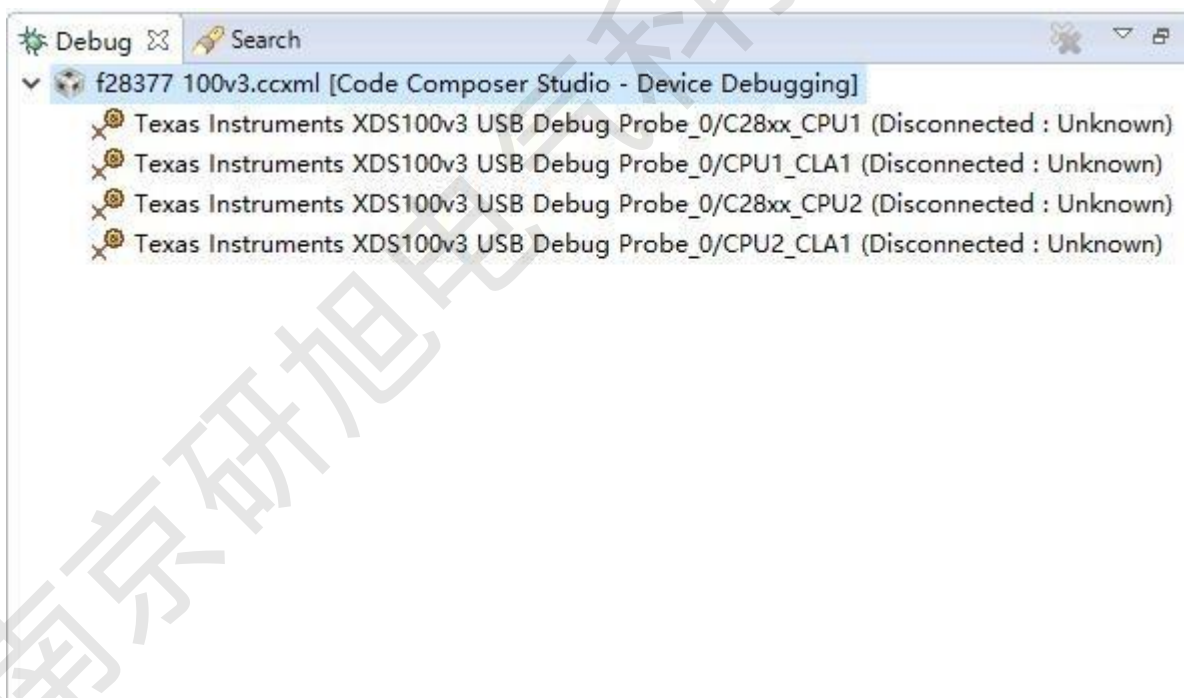



图 2-4-2

当启动成功后, 由于 28377D 为双核芯片, 还有两个 CLA 运算核心, 需要单机鼠标左键先选定核心才能进行连接。此时, 连接 LOGO  才会点亮, 如下图 2-4-3 所示:

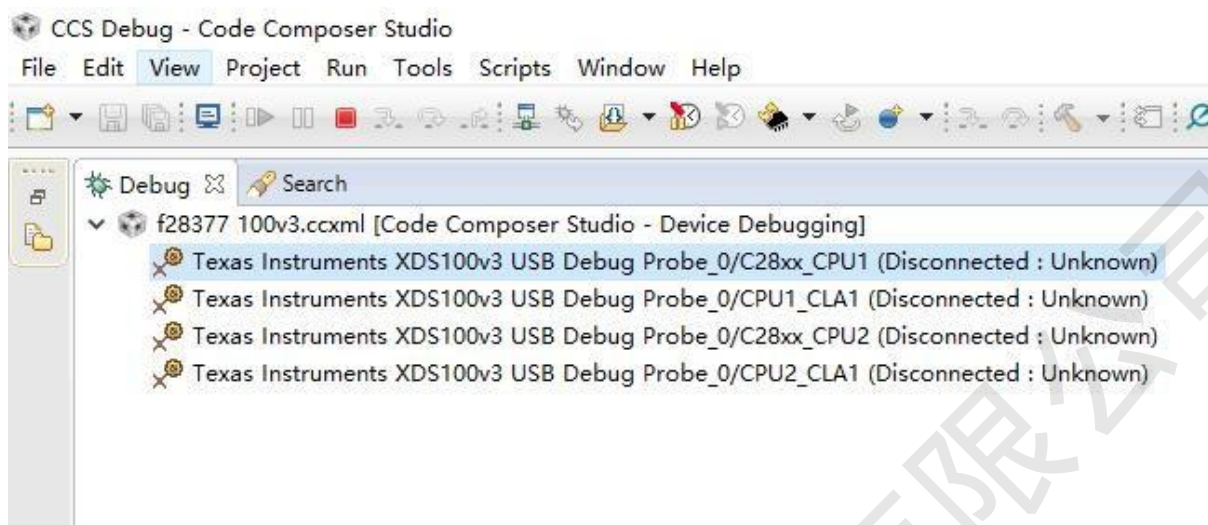


图 2-4-3

然后接入仿真器和目标板实物（实物连接应先将仿真器与目标板连接后再给目标板上电），在工具栏中点击“Run->Connect Target”或直接点击连接 LOGO 即可连接目标板，见下图 2-4-4 所示界面：

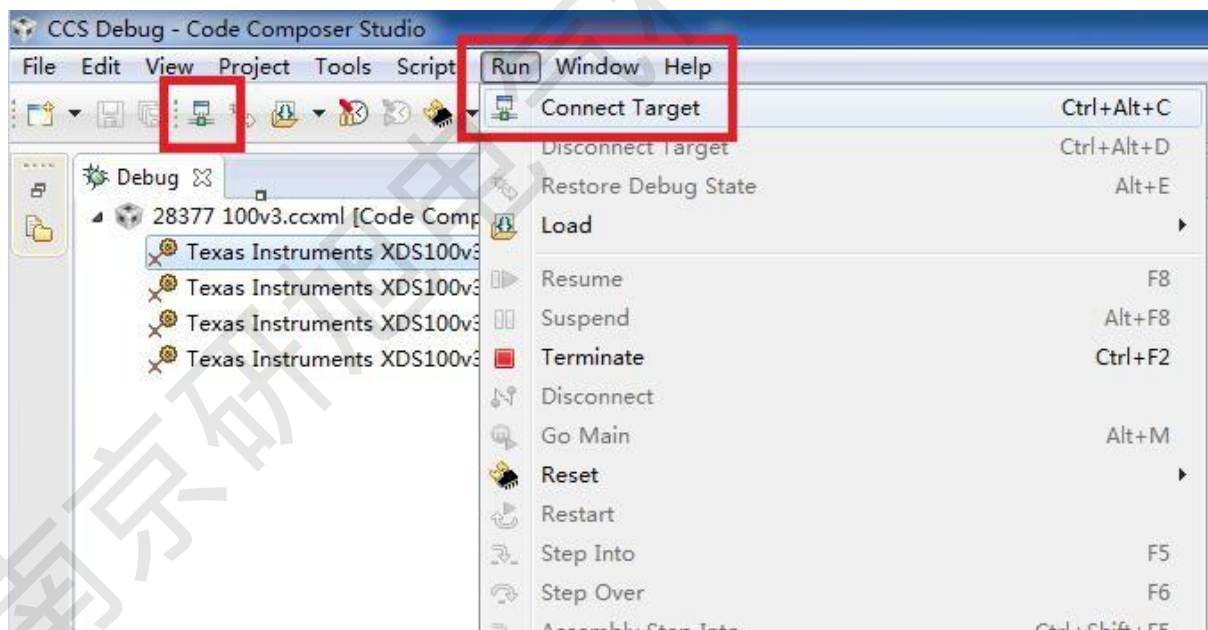


图 2-4-4

出现下图 2-3-5 所示的界面即表明连接成功，接下来，可以实现仿真或烧写。

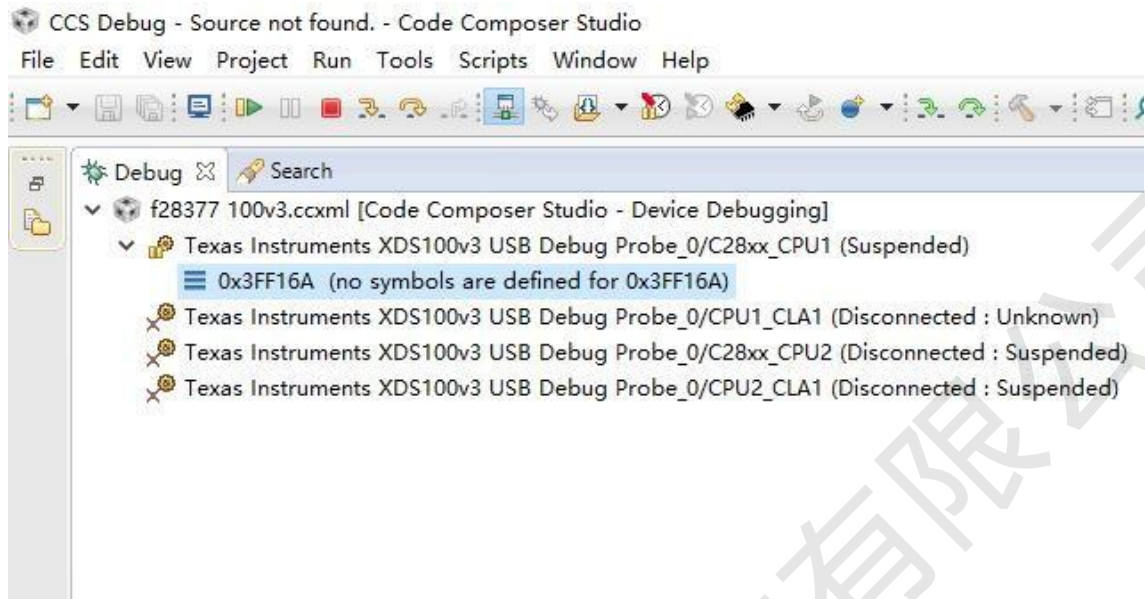


图 2-3-5

第三章 创建 CCS 工程

3.1 创建工程以及升级编译器

①在编译环境“CCS Edit”下, 点击“File -> New -> CCS Project (文件->新建-> CCS 项目)” 如下图 3-1-1 所示:

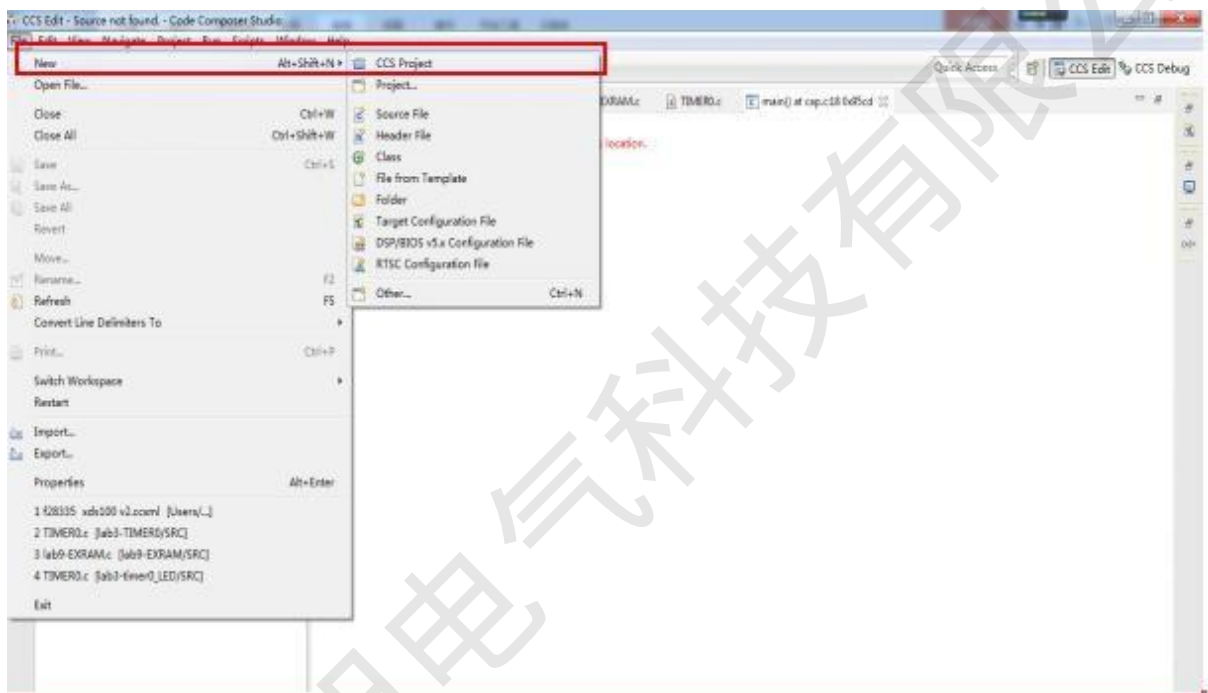


图 3-1-1

② 如图 3-1-2 所示, 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。取消选中该选项可以选择一个新位置(使用“Browse... (浏览...)”按钮)。此处将项目命名为“test1”。

③ 在“Target”菜单中选择要使用的芯片类型, 选择2837xD
Delfino->TMS320F28377D。

④ 在“Connetion”里选择调试器, 此处选择 Texas Instruments XDS100v3 USB Debug Probe。

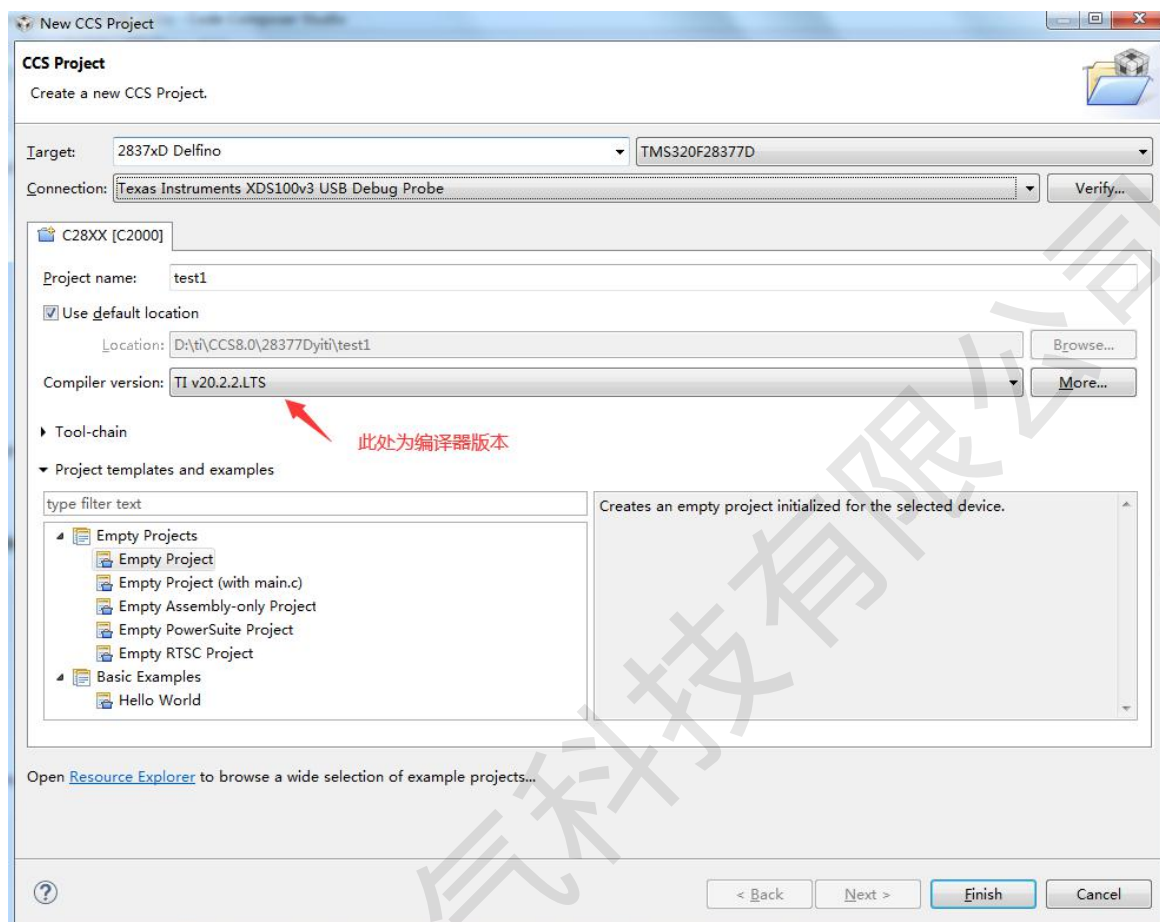


图 3-1-2

⑤ 其余选择默认，完成点击FINISH即可。**注意，此时编译器版本为V20.2.2，如显示的不是此版本，建议升级编译器，否则后续程序编译过程中可能会报错。**用户可以选择在线升级，在help菜单下->选择Install New Software->在TI Compiler Updates下我们可以选择相应的编译器版本，如图3-1-3所示，然后点击NEXT，自行安装即可。

当然我们也提供了20.2.2编译器版本的离线安装包，双击我们资料包里自带的ti_cgt_c2000_20.2.2.LTS_windows_installer文件，在Destination Directory 中，选择CCS的安装路径，这里建议装在CC安装目录下。如：D:\ti\ccsv8\tools\compiler。每个用户可能有所区别，但最终都建议装在CCS文件目录的tools文件的compiler中。直接把编译器复制到该路径下安装可不用自行选择，安装时一直点击NEXT直至安装结束。

此时，我们在新建工程的面板下，在Compiler version一栏下拉或者直接应该就可以看到20.2.2版本。

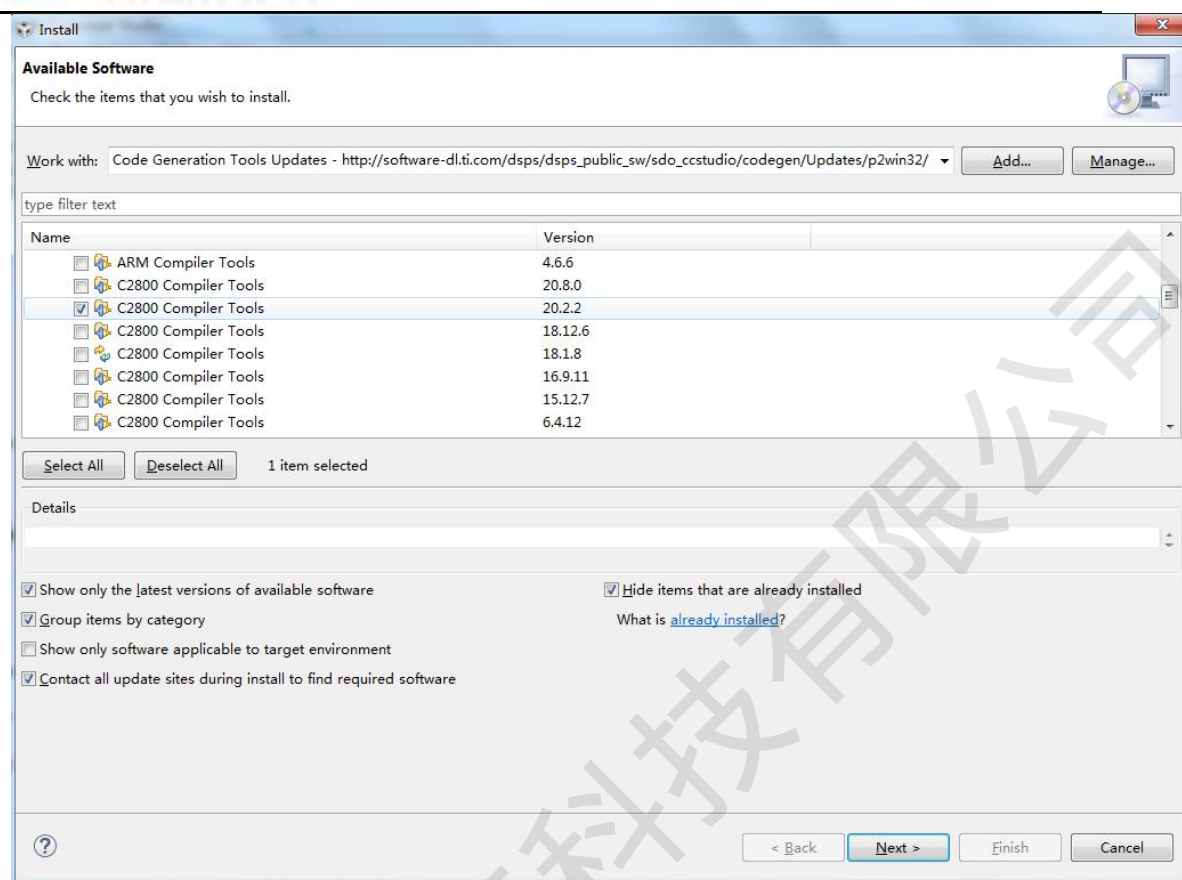


图3-1-3 编译器版本在线安装

⑥因为工程创建时在第④步选择了仿真器类型，这将导致新建工程只适配于一种仿真器，对工作造成不便，我们需要手动删除该文件。（注：如果不做第④步的配置 CCS 可能报错，建议按教程顺序创建工程。）在工程目录下（路径“View->Project Explorer”），右键单击 targetConfigs, 选择 delete 删除。如图 3-1-4 所示：

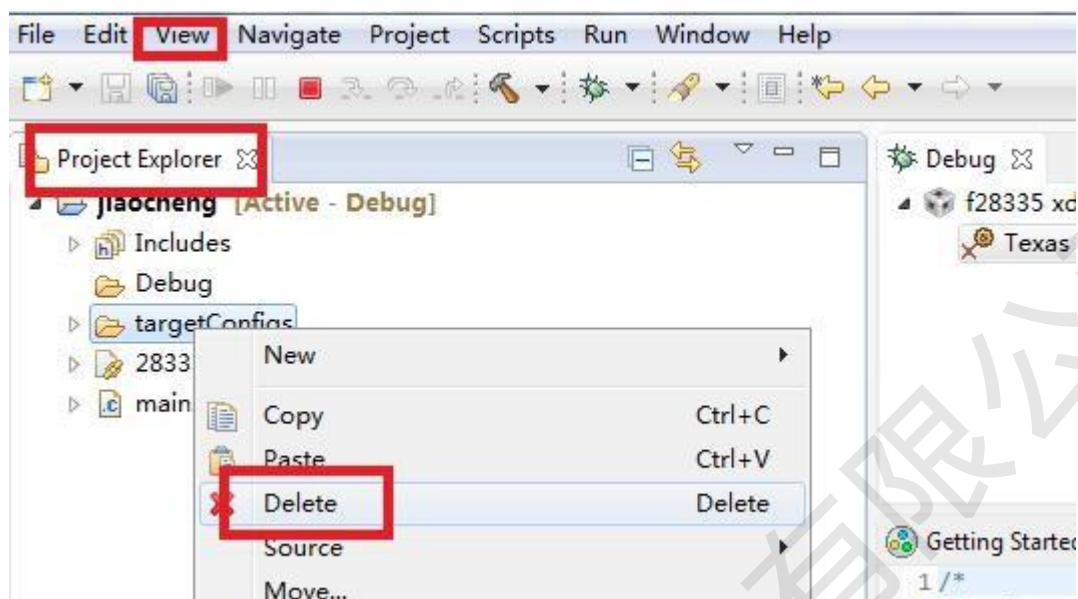


图 3-1-4

3.2 工程编写

①工程一般由命令文件“.cmd”（CMD）、头文件“.h”（INCLUDE）和源文件“.c”（source）文件,以及“main.c”文件组成。编写工程时,我们首先删除新建工程下的“2837x_FLASH_lnk_cpu1.cmd”和“main.c”（避免出现重复）,然后在正在使用的workspace里找到新建的工程,在工程文件夹里创建三个新的文件夹 CMD、INCLUDE、source。

注:编译时所需的文件(.c、.h、.cmd...)可以直接引用 TI 官方文件,本公司例程均是使用的官方文件,所以可以直接复制例程中的对应文件。具体过程是,打开例程文件夹找到需要用到的文件,复制粘贴到 workspace 下新建工程中的对应文件夹里。

②CMD 文件负责分配 DSP 的内存空间,将例程中的“.cmd”文件拷贝到新建工程 CMD 文件夹下,根据仿真和烧写,自行选择屏蔽(参见第五章内容)。方法是右击要屏蔽的文件,选择“Exclude from build”。例如“2837xD_FLASH_lnk_cpu1.cmd”是要屏蔽的文件,则情况如图 3-2-1 所示:

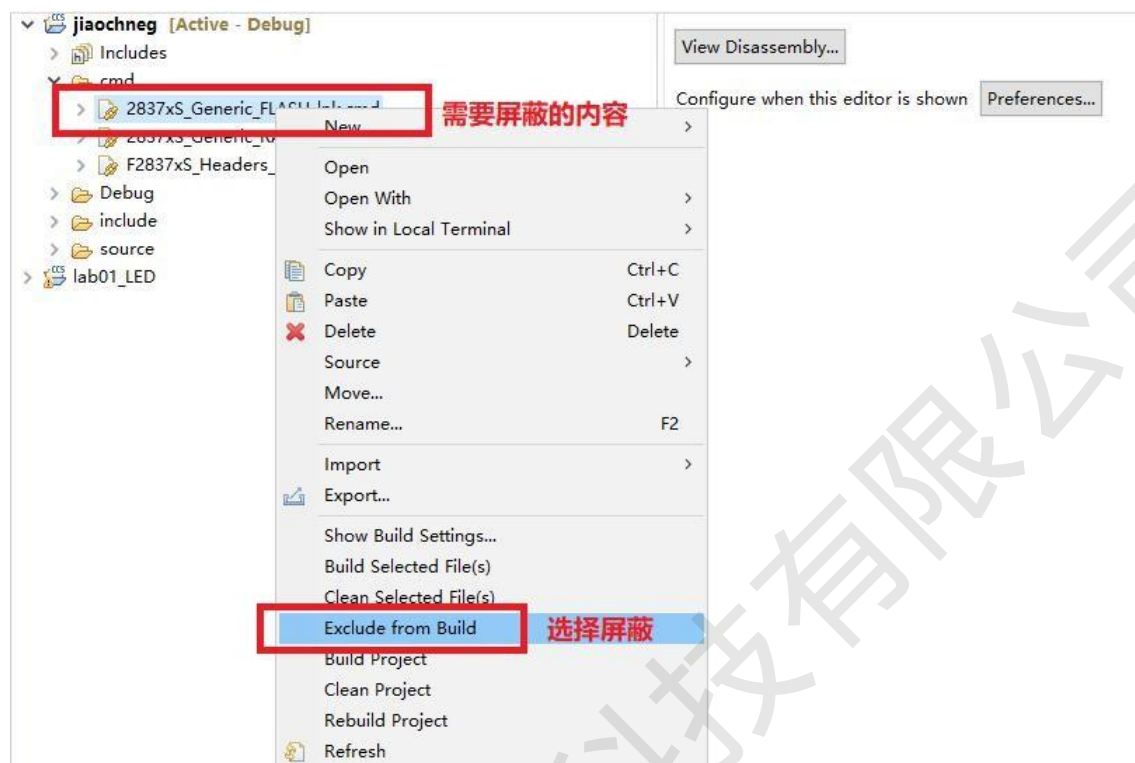


图 3-2-1

③INCLUDE 中负责放置工程的“.h”文件，直接拷贝例程中的文件到 INCLUDE。

④source 中放置工程的所有“.c”文件，包括“main.c”。用户可以根据需求拷贝同种类型例程的 source 文件，然后重新编译“main.c”文件或在其基础上进行修改。需要注意的是，main 函数中用到的所有定义函数都要包含在 source 中，如没有，请在别的例程中找到定义文件，补充到 source 中。

注：当官方文件无法满足我们的需要时，我们可以自行编译新的定义文件，在“View -> Project Explorer”视图下，右键单击项目名称（如 jiaocheng），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）”，完成源文件添加，然后在新建源文件中进行编译即可。如图 3-2-2 所示：

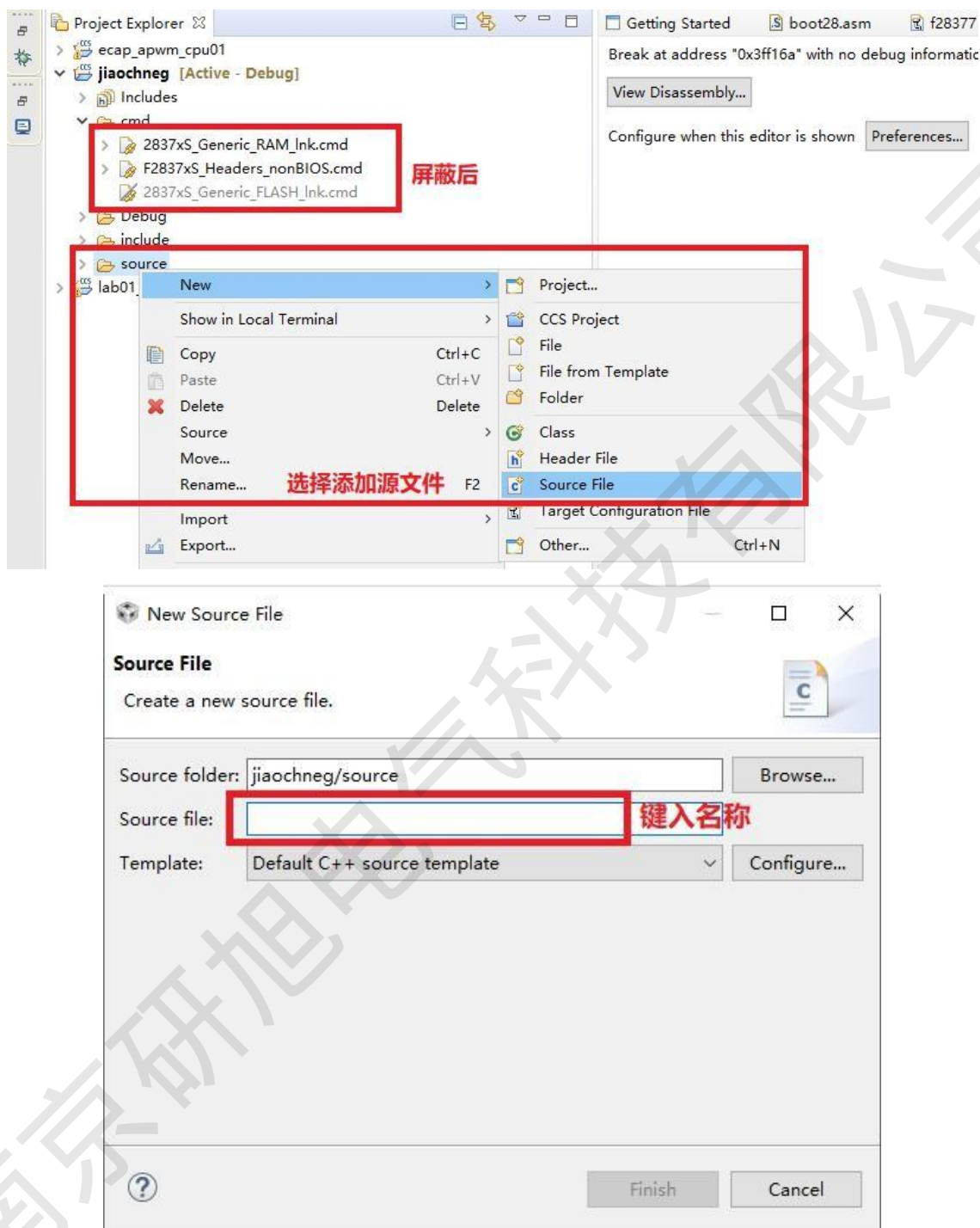


图 3-2-2

3.3 生成项目

在创建了项目并且添加或创建了所有文件之后，需要生成项目。只需右键工程名，选择“Build Project”即可。如图 3-3-1 所示。

注意：如果遇到生成错误，而且没有创建可执行文件，屏幕底部的控制台窗口将会显示一条错误或警告消息，并且不会启动调试会话。

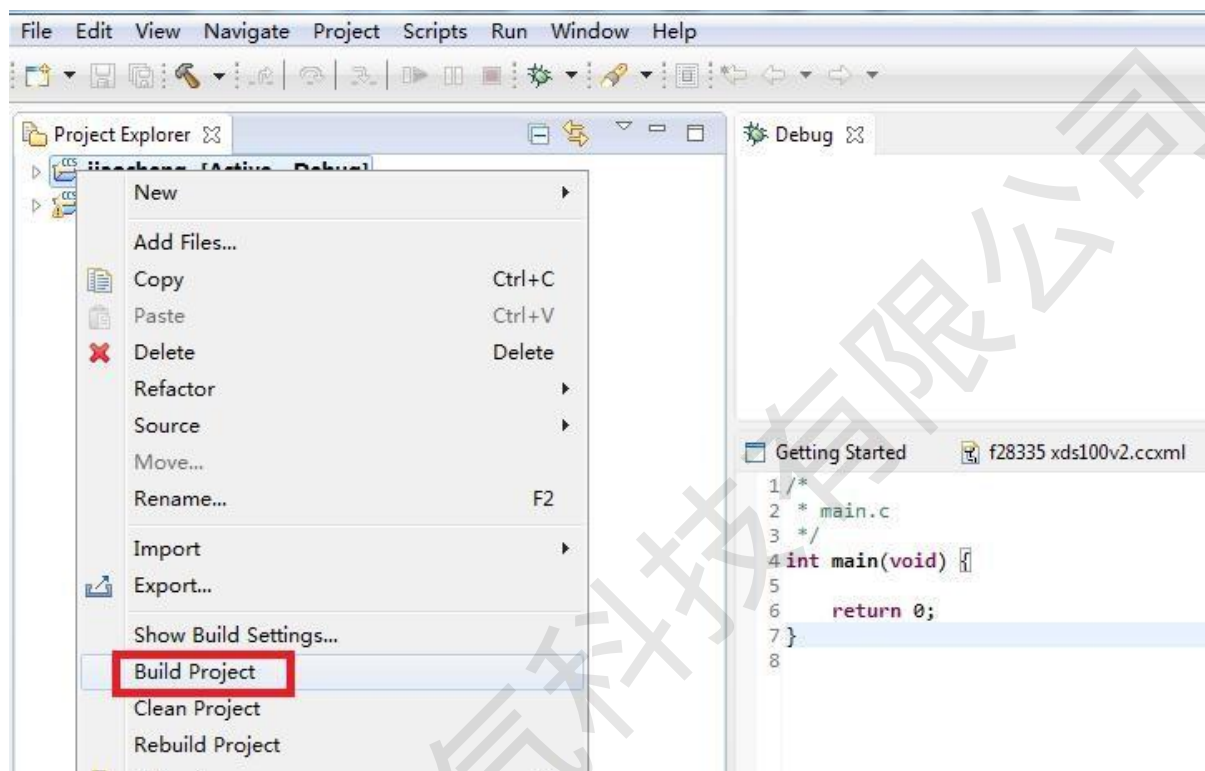


图 3-3-1

第四章 工程导入

4.1 CCS 低版本工程的导入

CCS 软件低版本编写的工程可以直接导入高版本，但是高版本的工程不能直接导入低版本 CCS 下使用，这里，我们先介绍常见的低版本工程的导入。

首先我们需要切换到 CCS Edit 模式下，在 CCS 界面的右上角有一个“Open Perspective”快捷按钮，点击它在打开的窗口选中 CCS Edit 选项，点击 OK 就会打开 CCS Edit 工作环境，如下图 4-1-1 所示：

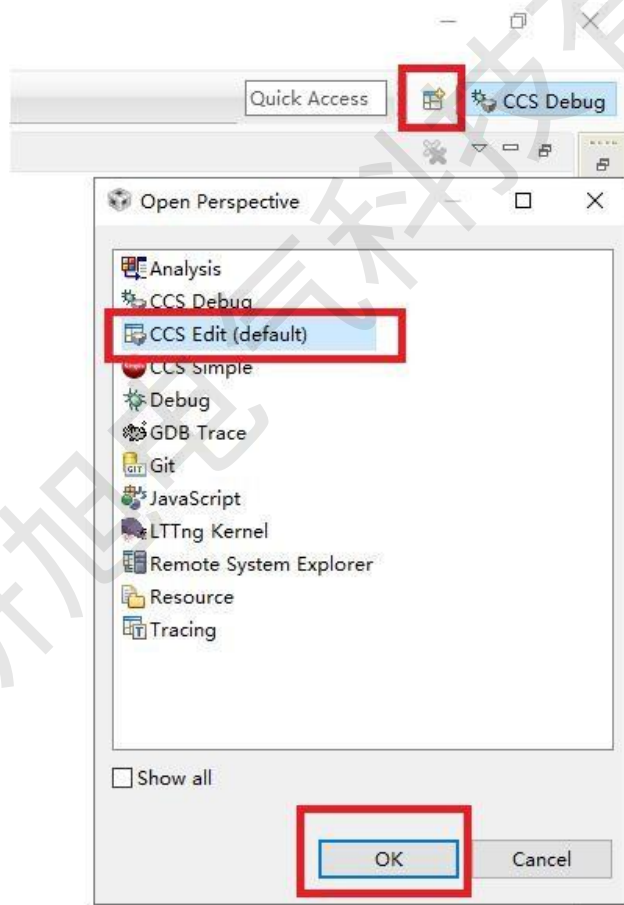


图 4-1-1

在打开的 CCS Edit 环境下选择菜单栏“Project”下的“Import CCS Project”，见下图 4-2-1 所示：

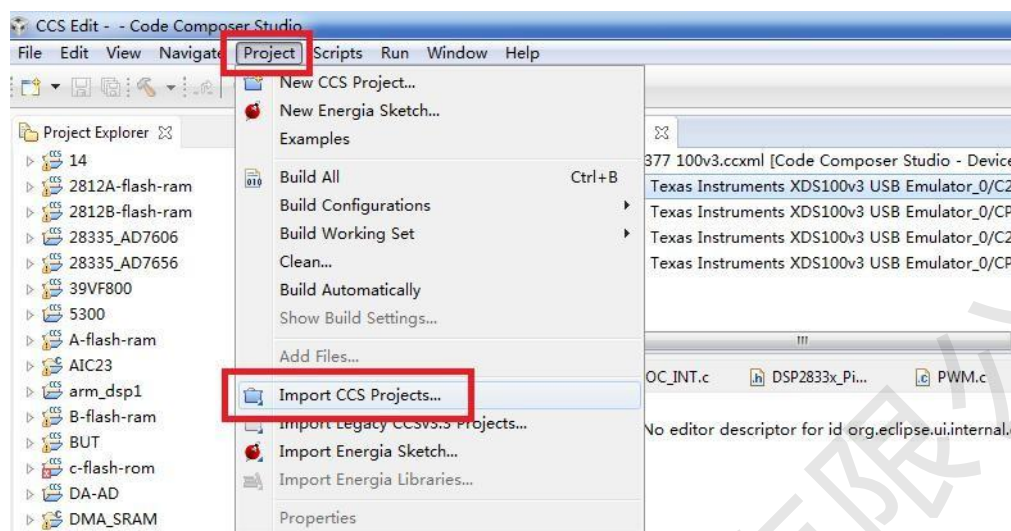


图 4-1-3

在弹出的窗口中作如下操作（注意：如果操作过程中出现警告，点击“OK”忽略），
如图 4-1-3 所示：



图 4-1-3

完成后, 右击工程选择“Build Project”进行编译, 编译完成后, 整个导入过程就全部结束。

4.2 CCS 高版本工程导入低版本

由于高版本工程不能直接导入低版本, 所以我们需要新建一个与原工程一模一样的本地工程来规避 CCS 的不兼容现象, 我们叫它程序搬移。

① 创建一个空白工程, 具体操作见第三章。(注: 在第三章第④步时可以选择 empty project, 使得工程更加简洁, 否则需要删除多出的 main.c 文件等, 否则会产生冲突。)

② 在工程目录下选中新建的工程, 双击找到仿真器配置文件“targetConfigs”和“F2837xD_Generic_FLASH_Lnk.cmd”文件, 选择“delete”删除。如图 4-2-1 所示:

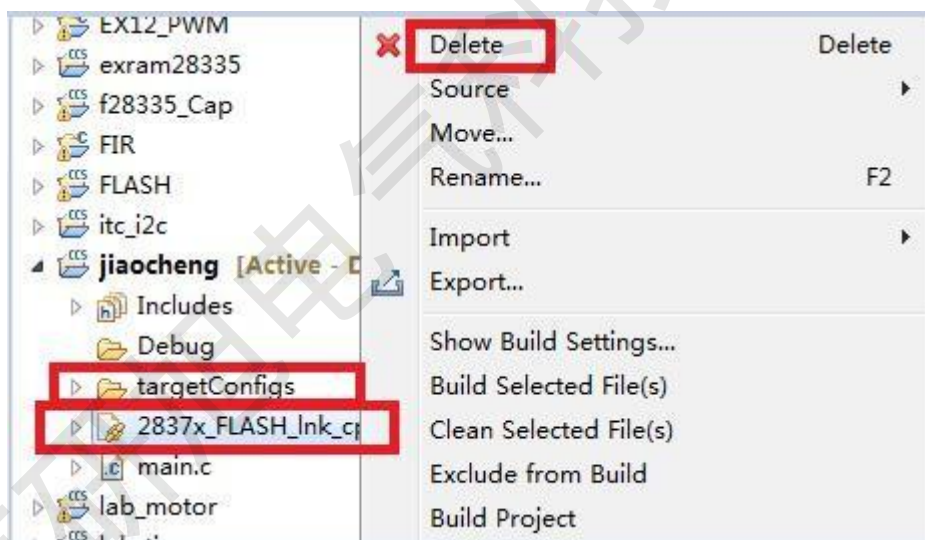


图 4-2-1

③ 在 workspace 里找到原工程文件夹, 复制 CMD、INCLUDE、SRC 文件夹将其粘贴到新建工程文件夹里。然后在 CCS 工程目录中右击新建工程名, 选择“Properties”添加新的文件地址。如图 4-2-2 所示:

④ 然后检查 CMD 文件下的“F2837xD_Generic_FLASH_Lnk.cmd”和“F2837xD_Generic_RAM_Lnk.cmd”是否是仿真设置配置正确后编译工程。

(注: 如果编译错误, 一般是文件重复定义“redefine”, 检查 INCLUDE 与 SRC 下的文件与原工程文件是否一致, 原工程有无屏蔽的文件, 有则屏蔽掉在进行编译。)

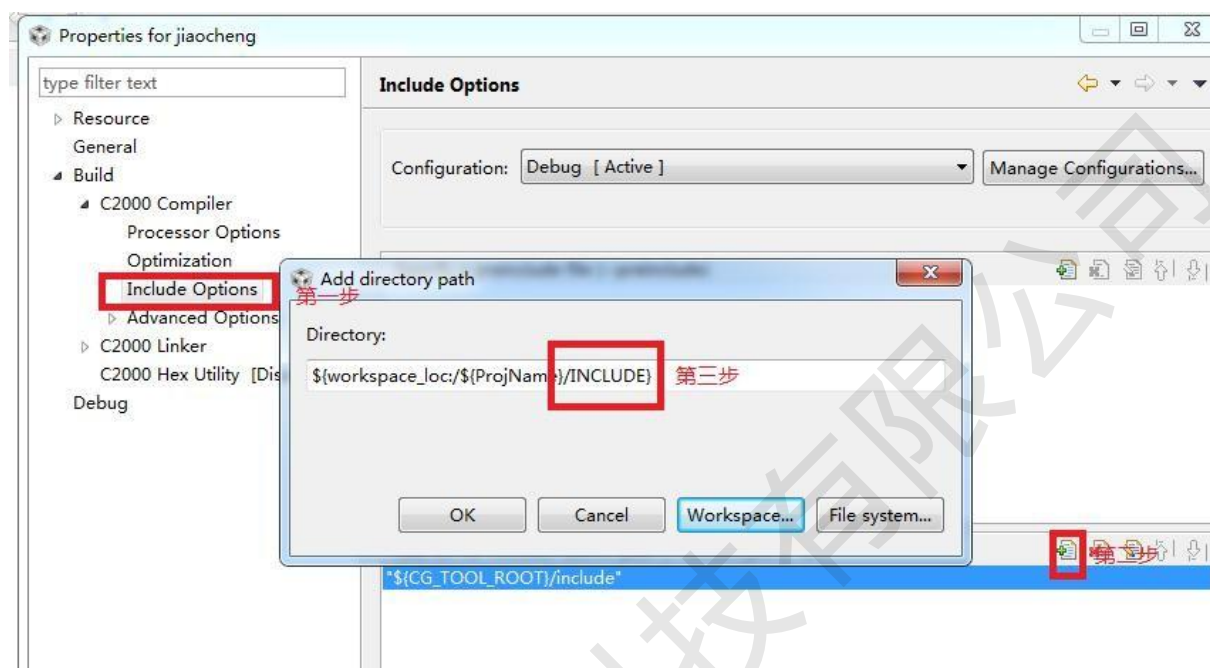


图 4-2-2

编译完成后就完成了工程的搬移。

第五章CCS9.2的仿真与烧写

5.1 F28377D 的仿真操作

仿真表示可以在线调试程序,但是仿真器断开连接后实验板芯片内程序被清除。在例程导入后,下面就该进行 CCS 与开发板的联调,这里介绍 TMS320F28377 一体板为例进行仿真测试。

在导入工程之后,在 Project Explorer 窗口下找到工程的 CMD 文件(双击工程名),屏蔽“2837xD_FLASH_lnk_cpu1.cmd(烧写所用 CMD 文件)”同时保留“2837xD_RAM_lnk_cpu1.cmd(仿真所用 CMD 文件)”,此时 2837xD_FLASH_lnk_cpu1.cmd 呈灰色,2837xD_RAM_lnk_cpu1.cmd 正常(注意:2837xD_FLASH_lnk_cpu1.cmd和 2837xD_RAM_lnk_cpu1.cmd 两者只能选其一参与编译,否则编译器将无法识别具体的操作空间而出错)。详细操作见下图 5-1-1 和 5-1-2 所示:

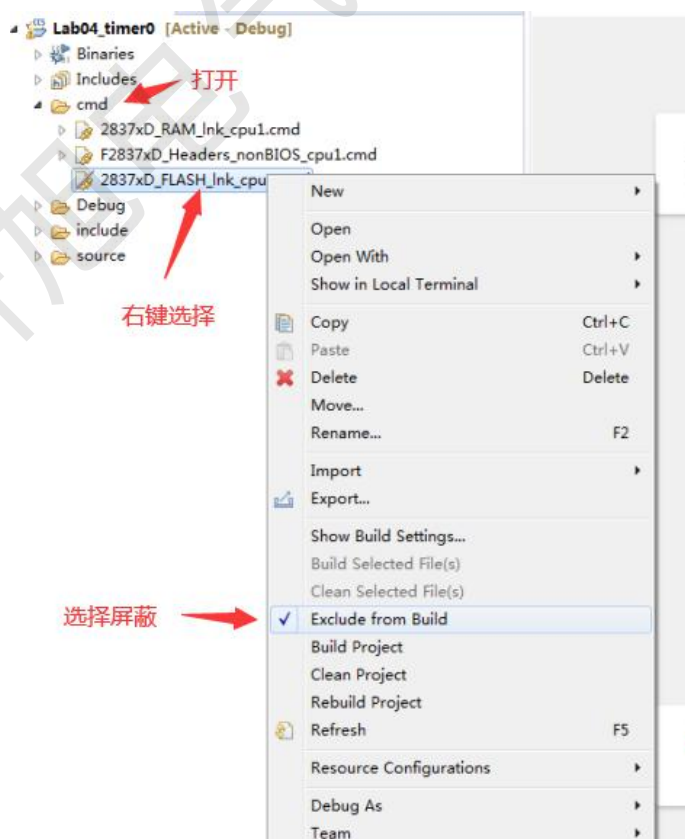
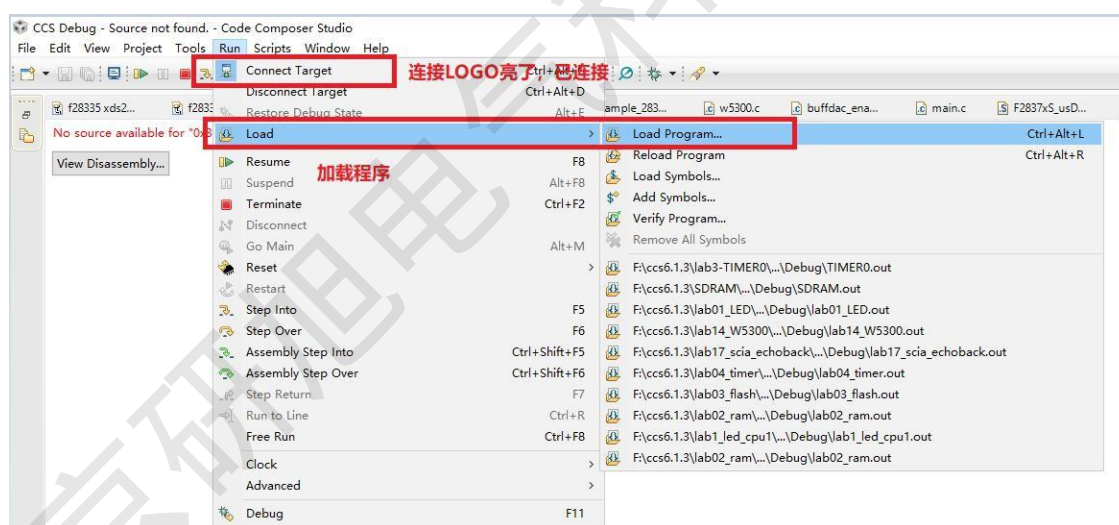


图 5-1-1



图 5-1-2

CMD 文件更改过后右击工程选择 Build Project, 编译成功后, 将视图切换到“CCS Debug”视图下, 先在连接实验板, 再对工程进行加载(步骤: 点击“工具栏 Run->Load->Load Program”或者直接使用图标快捷键), 详见下图 5-1-3 所示:



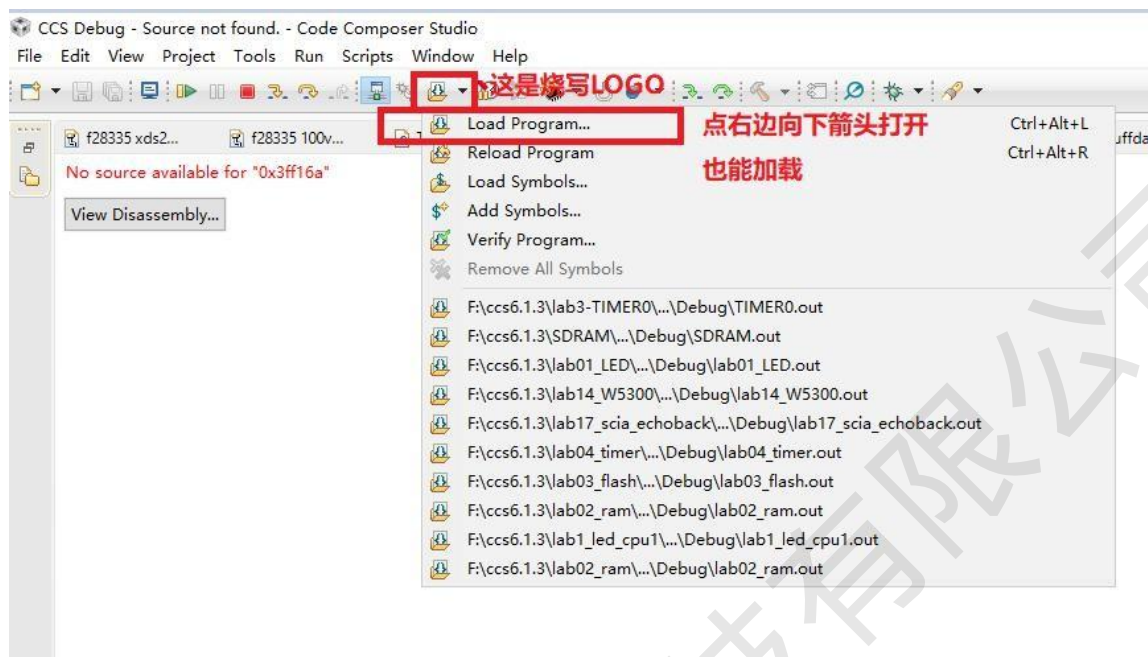


图 5-1-3

在出现的对话框里选择需要加载工程的.out 文件，如下图 5-1-4 的操作所示：

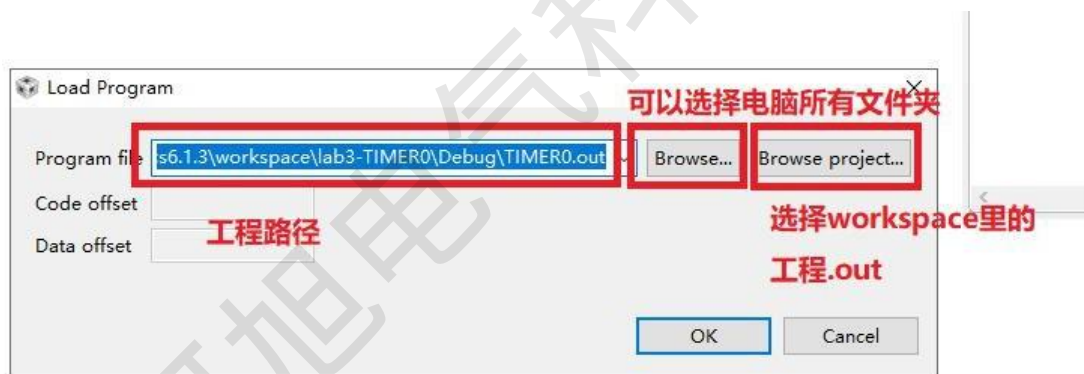


图 5-1-4

加载完成后点击运行，即可观察到开发板上的现象，操作如下图 5-1-5 所示：

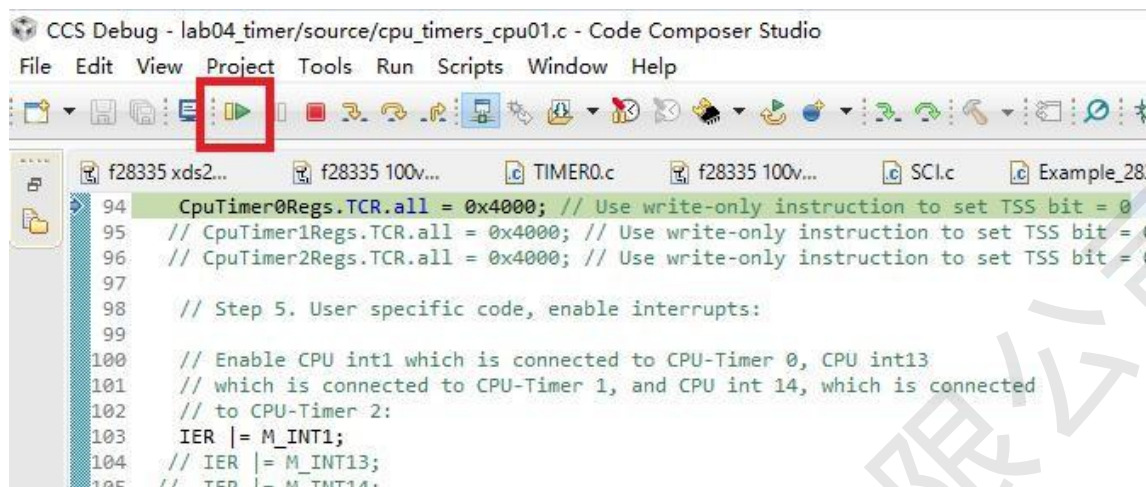


图 5-1-5

至此，仿真操作进行完毕。

5.2 F28377D 的烧写操作

烧写表示将程序下载到芯片，去掉仿真器后芯片上电仍可以单独运行。DSP28377D 的烧写操作与仿真操作类似，首先要将导入的工程的 CMD 文件从“2837xD_RAM_lnk_cpu1.cmd（仿真所用 CMD 文件）”替换成“2837xD_FLASH_lnk_cpu1.cmd（烧写所用 CMD 文件）”（注意：2837xD_RAM_lnk_cpu1.cmd 和 2837xD_FLASH_lnk_cpu1.cmd 两者只能选其一参与编译，否则编译器将无法识别具体的操作空间而出错），如图 5-2-1 所示：

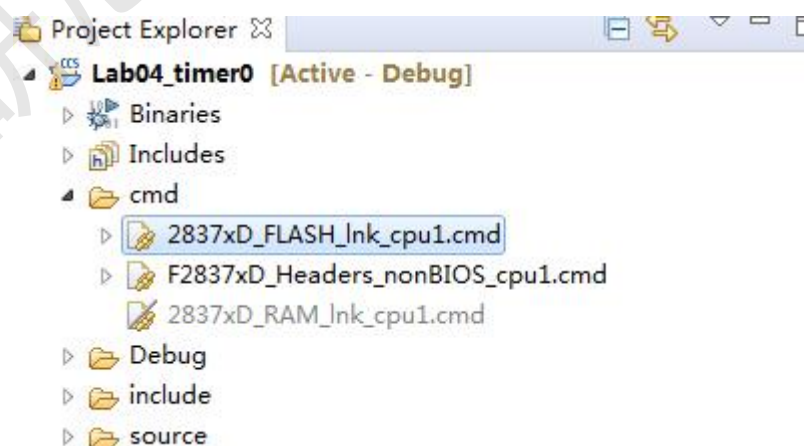


图 5-2-1

然后右击工程选择最下方的“Properties”选项，在打开的窗口找到“Predefined Symbol”一栏，打开如下图 5-2-2 窗口：

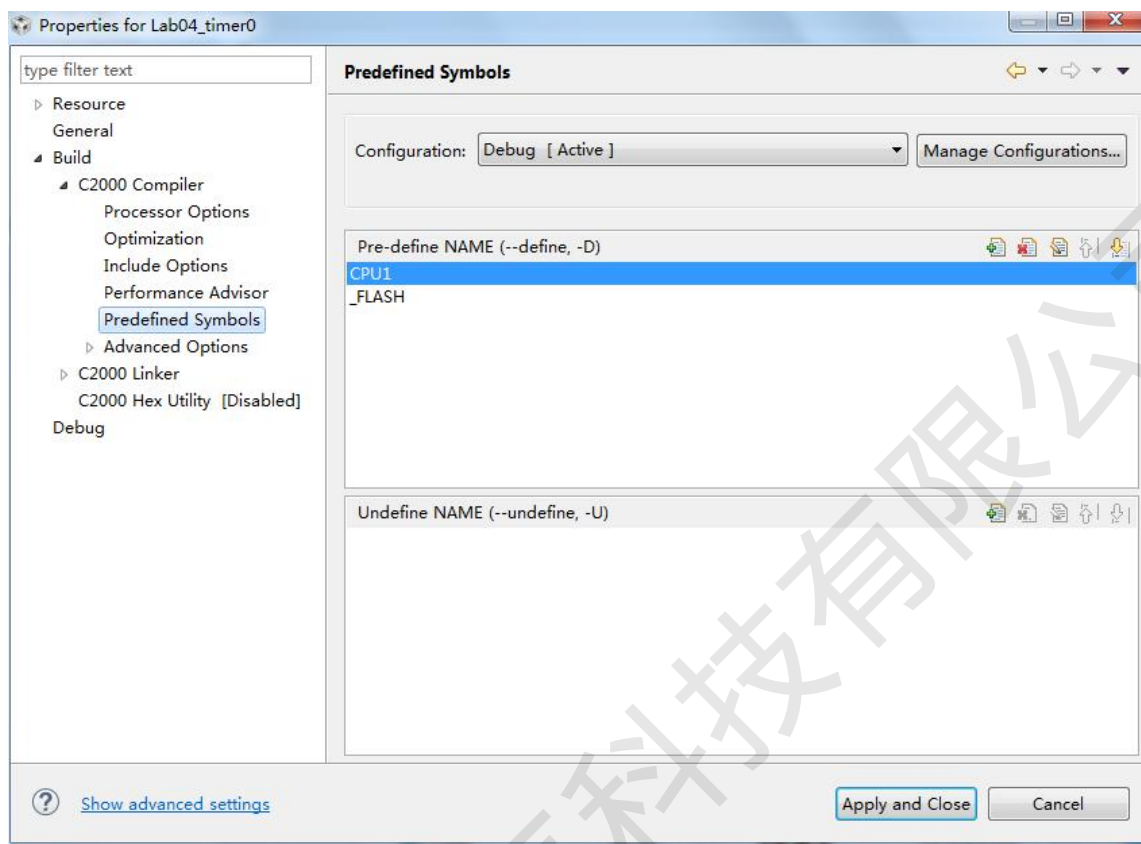


图 5-2-2

在打开的窗口下点击右上方加号图标添加如上图 5-2-2 的“CPU1”和“_FLASH”字符，单后点击下方的 OK 完成配置。

最后重复图 5-1-3、5-1-4、5-1-5 操作即可，即右击工程选择“Build Project”进行编译，编译没有错误后会在 Workspace（工作区间）的工程文件夹下的 Debug 文件夹里产生一个.out 文件），加载这个“.out”文件即可，点击运行，或者先断开 CCS 连接然后拔掉仿真器再单独给实验板供电，即可看到实验现象。

5.3 F28377D 的双核烧写操作

不同于常用的 C2000 核心 DSP2812 或者 28335，DSP28377D 拥有两个独立的核心，所以它除了单核程序外还拥有双核程序，这里我们单独介绍一下 28377D 芯片的双核程序烧写。

我们在第二章仿真器连接时候就介绍了 Launch 完连接文件后要单击鼠标左键选择 DSP 的核心，我们要烧写双核程序就要逐个选择 CPU1 与 CPU2，分别进行程序的烧写。

双核烧写和单核烧写的步骤基本一致，只有一处不同，在上一节的单核烧写中，我们右击工程打开了“Properties”选项窗口下的“Predefined Symbol”栏目，对该窗口进行了配置，烧写双核程序也一样，我们需要对该窗口进行相应的配置。

CPU1 核心烧写，配置窗口如下图 5-3-1 所示：

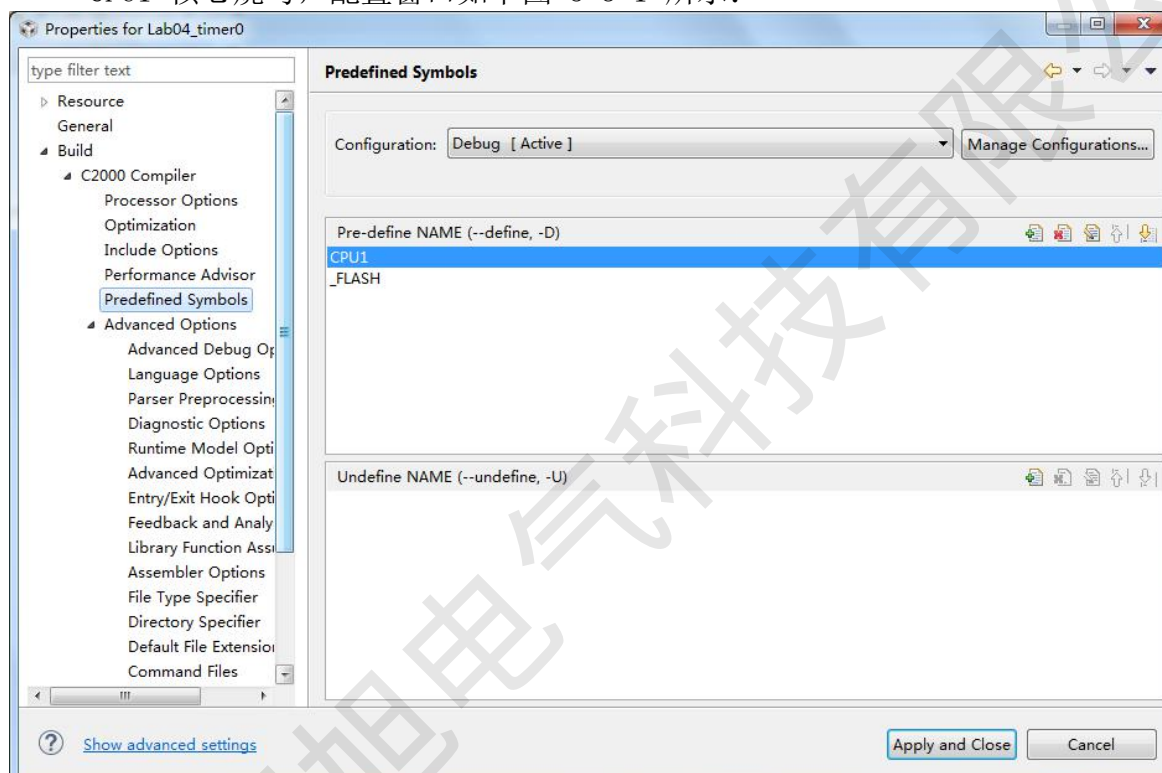


图 5-3-1

CPU2 核心烧写，配置窗口如下图 5-3-2 所示：把CPU1改成CPU2

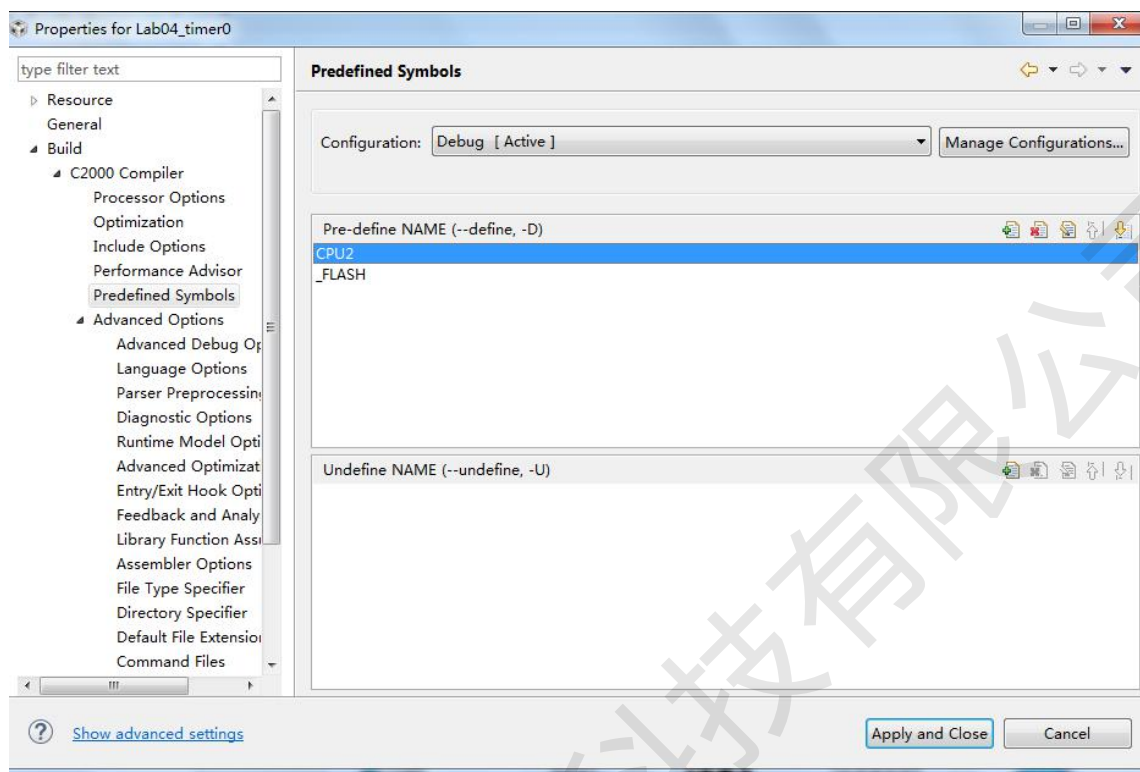


图 5-3-2

其余烧写过程只需参考5.2 节的操作方法即可，只需注意烧写CPU1 核必须使用CPU1 核心的程序，烧写 CPU2 核必须使用 CPU2 核对应的程序，然后给开发板单独上电，CPU1 与 CPU2 核均会单独开始工作，形成实验现象。

第六章 CCS9.2中一些常见的功能按钮

CCS8.0 中所有功能按钮都可以在工具栏中调出来，这里将一些比较常用的功能按钮拎出并加以介绍：

常见的功能按钮可在工具栏中的“View”、“Tools”和“Window->Show View”中调出来，如下图 6-1、6-2、6-3 所示：



图 6-1

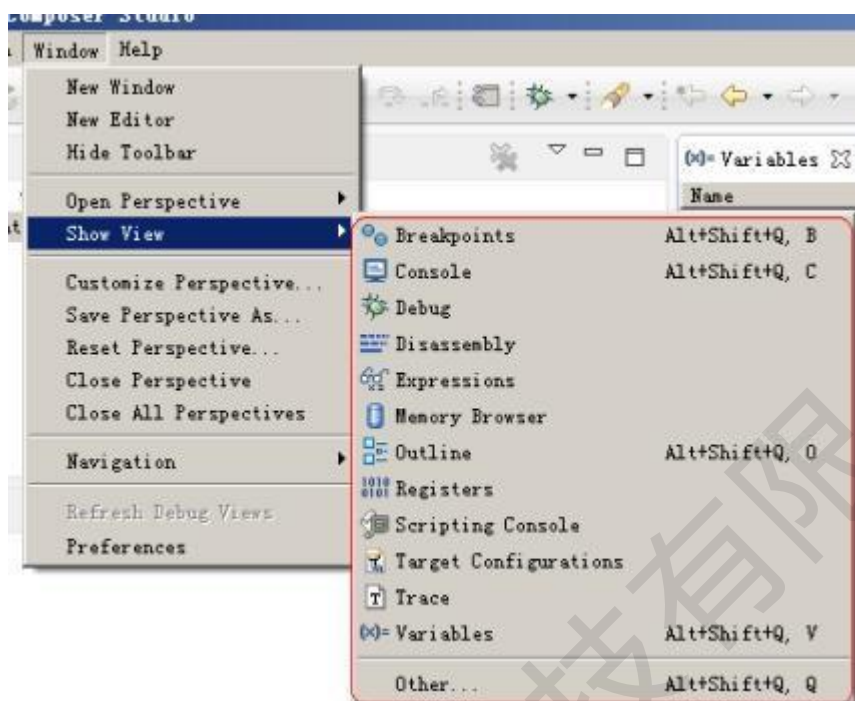


图 6-2

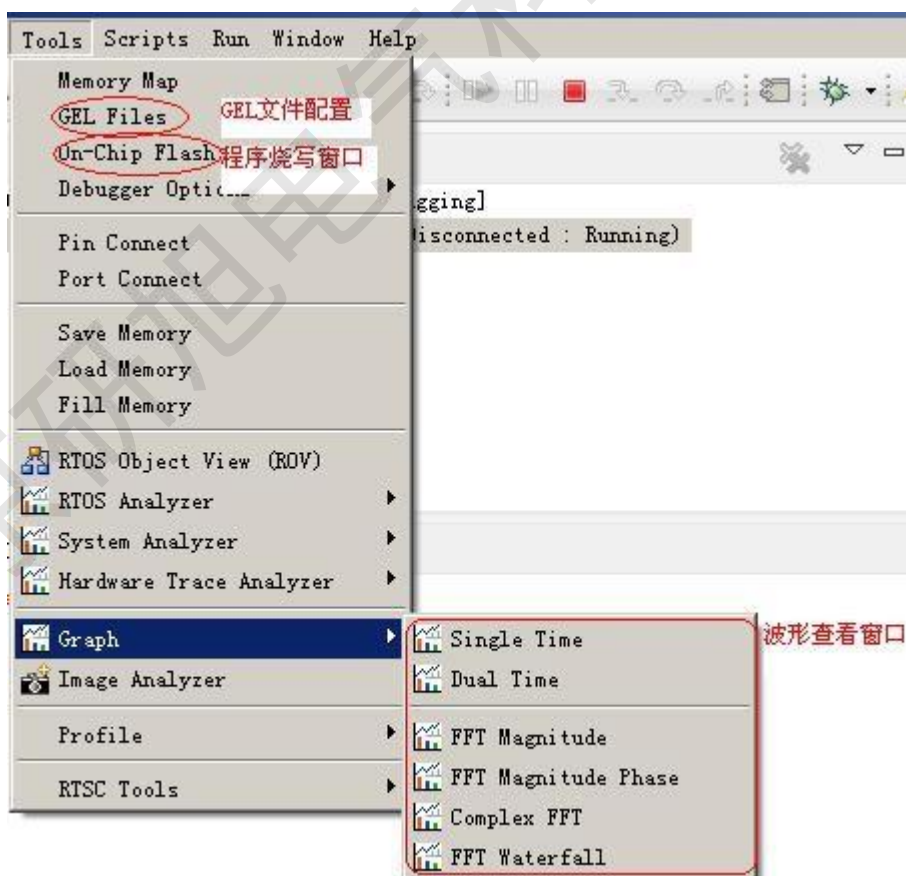


图 6-3

常见功能窗口介绍:

6.1 加载代码

调试器完成目标初始化之后, 项目的输出文件. OUT 将自动加载到活动目标, 并且默认情况下代码将在 main() 函数处停止。

注意: 代码将自动写入 MSP430、F28x 和 Stellaris 设备闪存中。要配置闪存加载程序属性, 请启动调试器并转到菜单 “Tools -> On-chip Flash (工具->片内闪存)”

① “CCS Debug (调试)” 视图包含每个芯片核的目标配置和调用堆栈。

②源代码视图显示了在 main() 处停止的程序。

③基本调试功能 (运行、停止、步入/步出、复位) 位于 “CCS Debug (调试)” 视图的顶部栏中。

④ “Target (目标)” 菜单还有其他几种调试功能。

注意: 如果目标配置需要先运行脚本再加载代码, 将打开 “View->Console (控制台)” 视图。

这些脚本采用 GEL (通用扩展语言) 编写而成, 在对包含复杂外部内存时序和电源配置的设备进行配置时尤其需要此类脚本。

6.2 监视变量和寄存器

在程序加载时还会打开 “Variables”、“Expressions” 和 “Registers” 视图, 并显示本地和全局变量, 如下图 6-2-2 和图 6-2-3 所示:



图 6-2-2

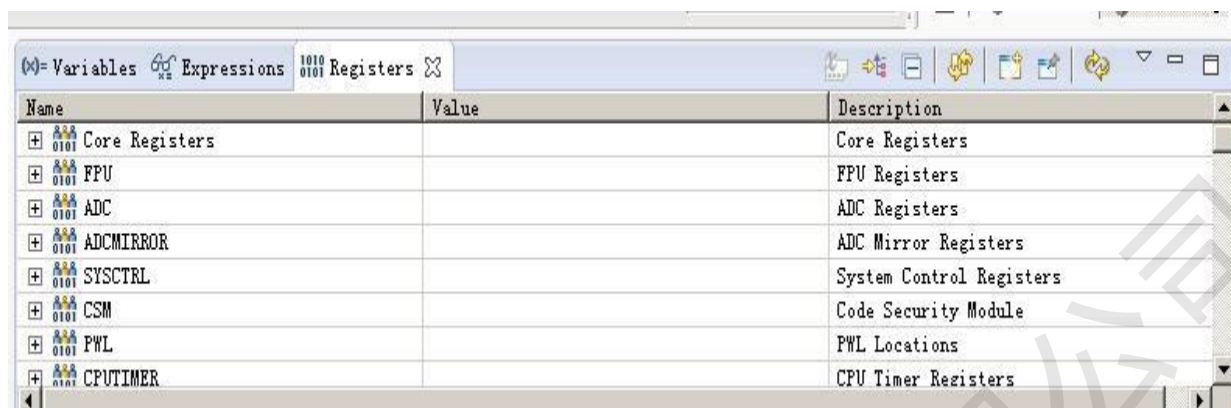


图 6-2-3

教程: 上面的屏幕截图显示了典型 F2833x 设备的所有外设寄存器，但是本例中所选的模拟器不模拟外设。

6.3 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，见下图 6-3-1 所示：

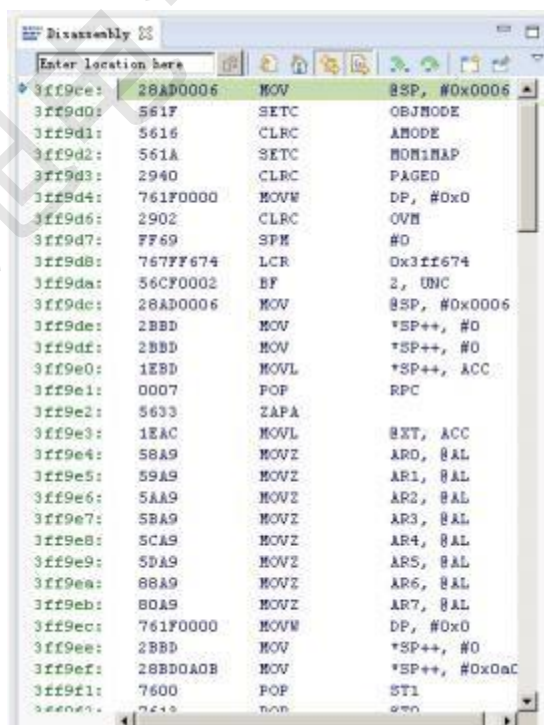


图 6-3-1

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，如上面的屏幕截图所示。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 6-3-2 所示：

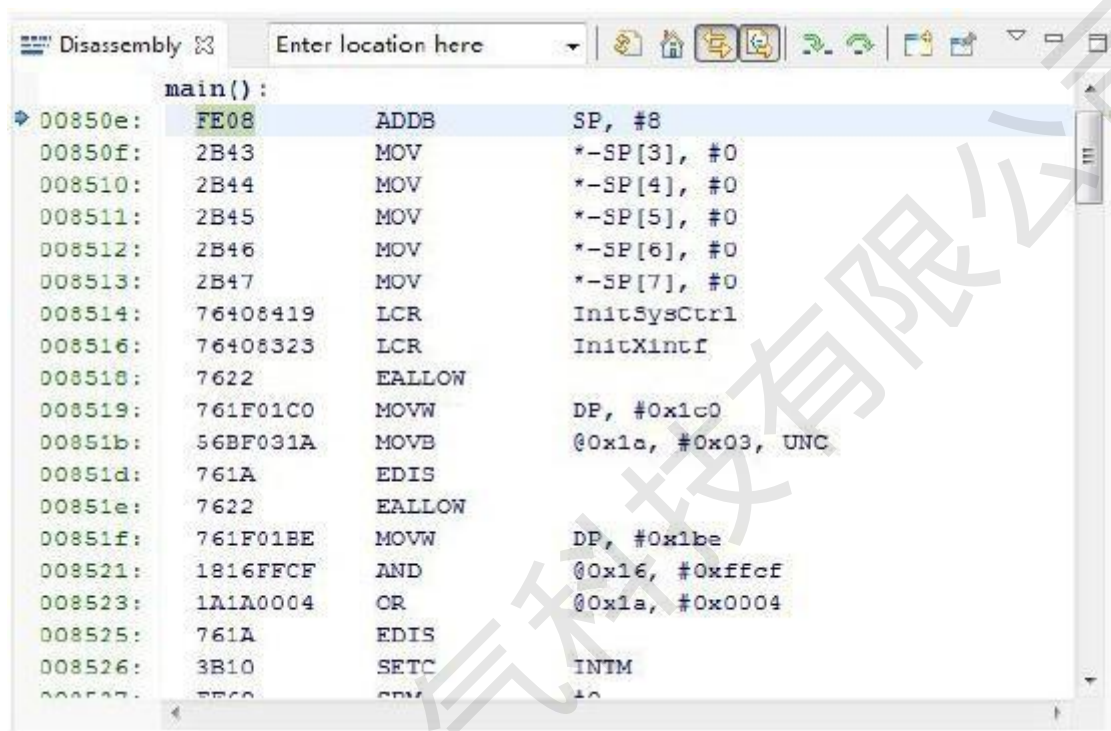


图 6-3-2

6.4 内存查看器

默认情况下不会打开内存视图，但是可通过转到菜单“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 6-4-1 所示：

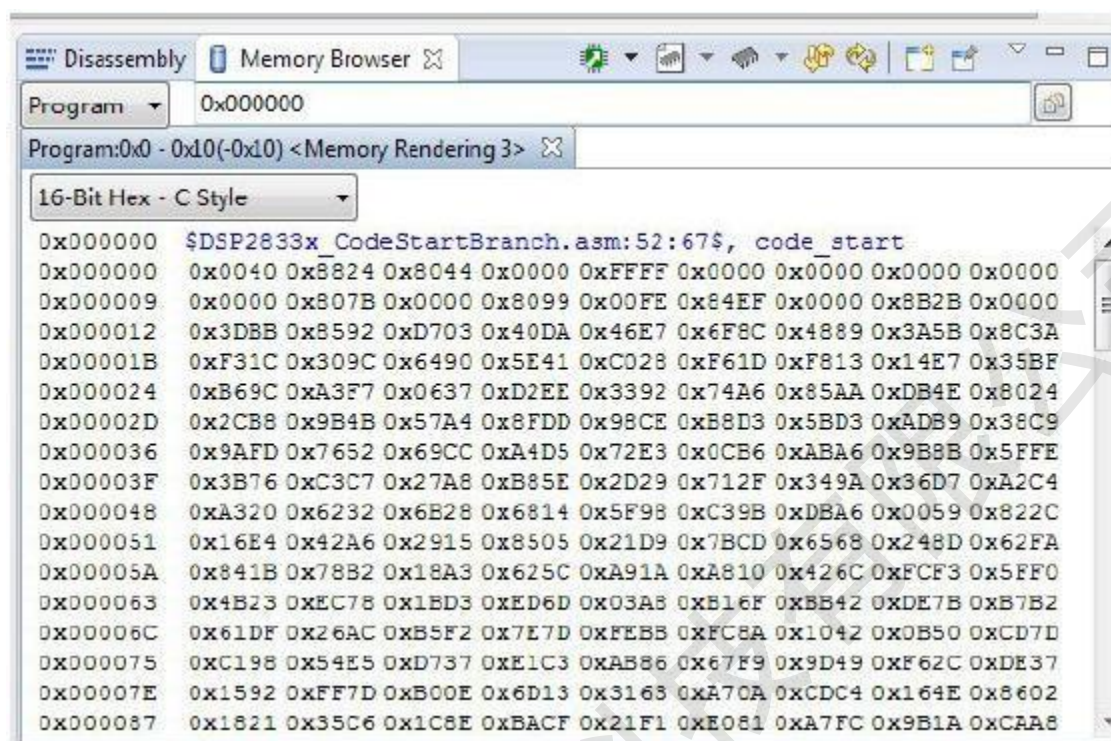


图 6-4-1

教程:该屏幕截图显示了分配在内存中的变量 `volatile short output[16]`。其他所有变量均为本地变量，因此分配在堆栈中。

6.5 管理断点

作为任何调试器都会拥有的最基本功能，CCSv8 中的断点添加了一系列选项，帮助增加调试进程的灵活性，如下图 6-5-1 所示：

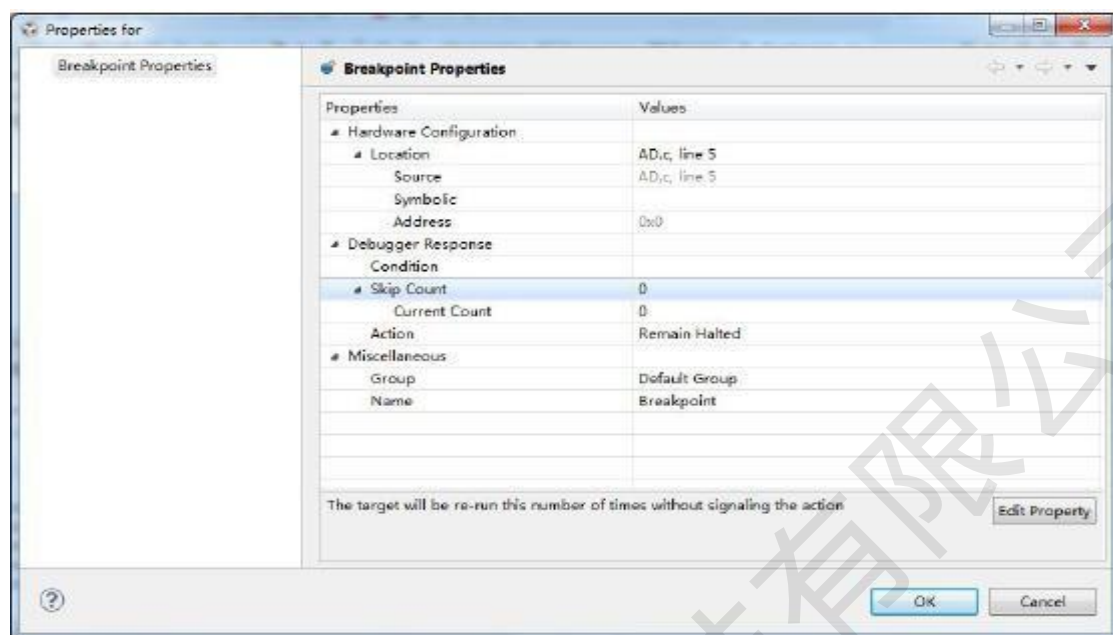


图 6-5-1

硬件断点可从 IDE 直接进行设置；

软件断点仅受到设备可用内存的限制；

软件断点可设置为无条件或有条件停止；

除了停止目标之外，软件断点还可执行其他功能：文件 I/O 传输、屏幕更新等。

要设置断点，只需在源代码或反汇编视图中双击代码行即可。硬件 或软件 断点的图标会指示其状态和放置位置。

注意：在优化代码中，有时无法将断点设置到 C 源代码中确切的某一行。这是因为优化器可能会将代码紧缩起来，从而影响汇编指令和 C 源代码之间的相关性。

所有断点（软件、硬件、已启用、已禁用）都可在下图 6-5-2 断点查看器中看到。

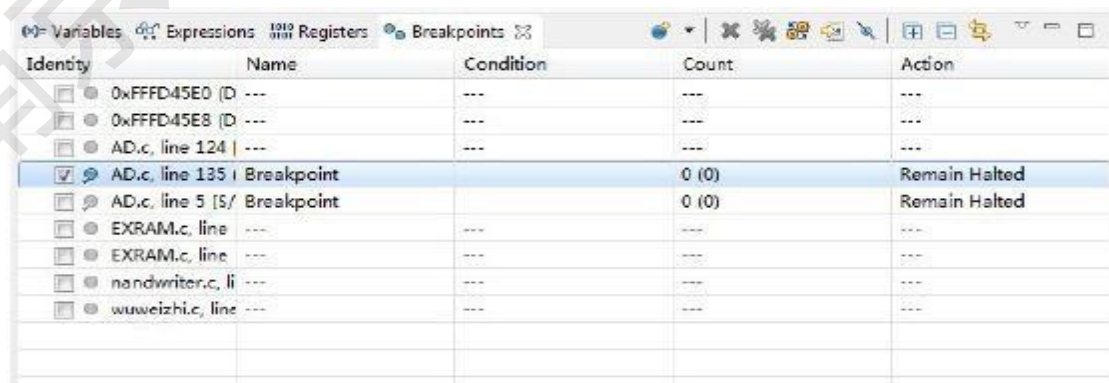


图 6-5-2

要配置断点, 只需右键单击蓝点, 或者在断点视图中右键单击并选择 “Breakpoint Properties... (断点属性...)”。

使用 “Action (操作)” 可以设置断点的行为, 例如保持停止、更新一个或所有调试器视图、从文件中读取数据或将数据写入其中、激活或停用断点组等。

使用 “Skip Count (跳过计数)” 可以设置执行断点操作之前通过的数目。

使用 “Group (分组)” 可以对断点进行分组以进行高级控制。

6.6 图形显示工具

CCSv8 中提供了一个高级图形和图像可视化工具。它可通过图形形式显示数组, 并且可采用多种格式。

要添加图形, 只需转到菜单 “Tools -> Graph (工具 -> 图形)”, 然后从各种显示选项中选择一个。

基于时间的图形: “Single Time (单曲线图)” 和 “Dual Time (双曲线图)”;

基于频率的图形: 所有 FFT 选项;

图形窗口中的顶部工具栏可控制多种功能, 例如更新速率 (冻结、连续、目标停止时或手动)、缩放、配置属性等, 如下图 6-6-1 所示:



图 6-6-1

默认情况下, 图形窗口会在目标停止时立即更新、使用自动缩放并以样本数显示 X 轴, 以整数值显示 Y 轴。所有这些选项都可进行设置。

注意: 请记住, 图形更新时所传输的数据量可能会影响目标硬件的实时操作。

教程: 下面的过程显示了包含正弦波发生器输出内容的图形。


①在源代码窗口中, 右键单击断点蓝点 (已在上一部分设置) 并选择 “Breakpoint Properties... (断点属性...)”。

②在 “Action (操作)” 属性中, 单击该属性值并选择 “Refresh All Windows (刷新所有窗口)”。这样将刷新所有窗口, 而不是将程序完全停止在该点。

③变量 `output[]` 包含 16 个正弦波发生器输出样本，因此整个缓冲区必须立即显示在图形窗口中。单击 “Tools → Graph → Single Time (工具 → 图形 → 单曲线图)”，然后将选项配置如下图 6-6-2 所示：

属性	值
采集缓冲区大小	16
Dsp 数据类型	16 位带符号整数
Q_value	15
开始地址	output

图 6-6-2

⑤屏幕底部应该出现一个图形窗口。如果需要，可通过单击  按钮更改图形属性。

⑥单击 “Run → Resume (目标 → 运行)”。该图形应该以 16 个样本为一组分批更新。

⑦要查看 `output` 数组的实际值，请单击 “Watch (监视)” 选项卡（应当在屏幕右上角部分），然后单击 “New (新建)”。键入 `output` 并展开此数组以显示其中的所有值。这些值以 16 位带符号整数输出，因此可通过调整 Q 值使其标准化：在 “Watch (监视)” 窗口中选择所有值，右键单击并选择 “Q-values → Q-value(15) (Q 值 → Q 值(15))” 如下图 6-6-3 所示：

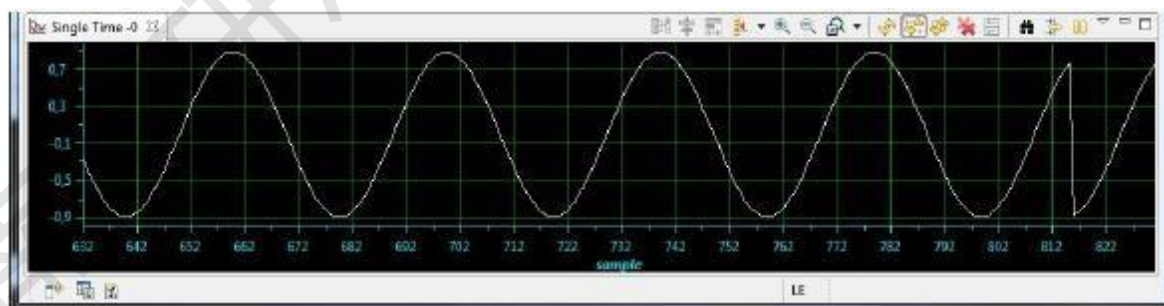


图 6-6-3

6.7 图像显示工具

要显示图像, 只需转到菜单 “Tools -> Image Analyzer (工具 -> 图像)”。屏幕底部将打开两个视图: “Image (图像)” 和 “Properties (属性)”。

CCSv6 显示的信息既可以是来自 PC 主机中的文件, 也可以是目标开发板中加载的图像。在属性页面中, 只需将 “Image source (图像源)” 选项设置为 “File (文件)” 或 “Connected Device (连接的设备)” 即可。

与图形查看器类似, 需要设置其他所有属性才能使显示内容有意义。彩色障板、线条尺寸和数据宽度等几种选项会影响图像的正确显示。

教程: 要显示加载至目标的图像, 请执行以下操作:

① 转到菜单 “View -> Memory Browser (查看 -> 内存)” 打开内存视图;

② 在地址框中键入有效的目标地址: 0xC0000000;

③ 将图像文件 <sample_24bpp.dat> 加载至 0xC0000000: 单击内存操作图标旁边的三角形, 然后单击 “Load (加载)”。浏览至下面的目录, 然后单击 “Next (下一步)”。

C:\Program Files\Texas Instruments\ccsv6\c6000\examples

④ 键入与内存窗口中相同的起始地址, 并将 “Type-size (类型大小)” 设置为 32 位;

⑤ 按下图 6-7-1 所示设置属性:

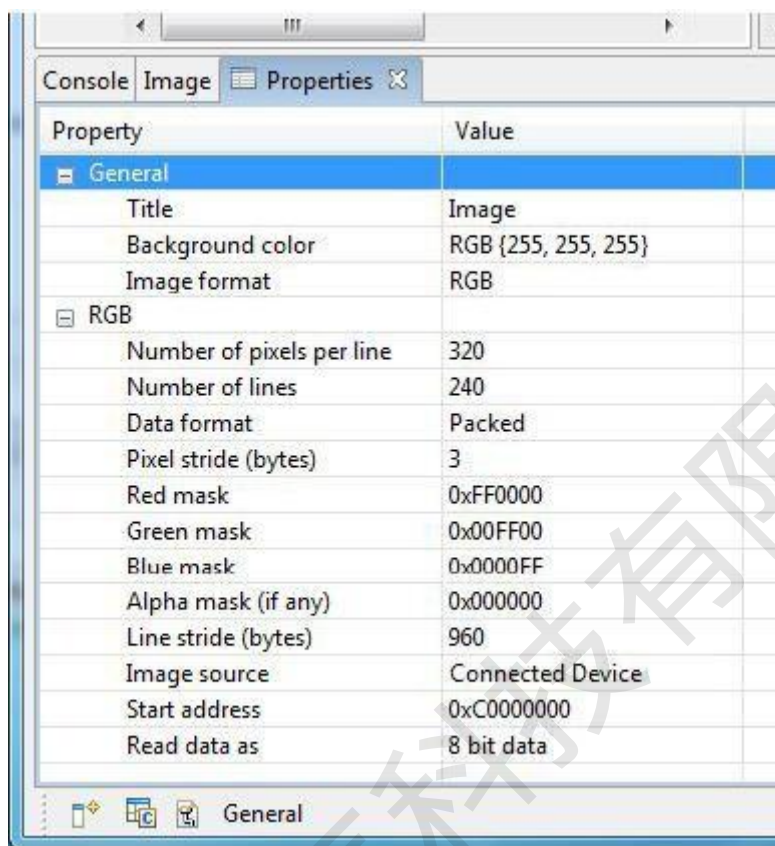


图 6-7-1

⑥选择“Image（图像）”选项卡，然后右键单击并选择“Refresh（刷新）”。应该会显示下图 6-7-2 的图像。

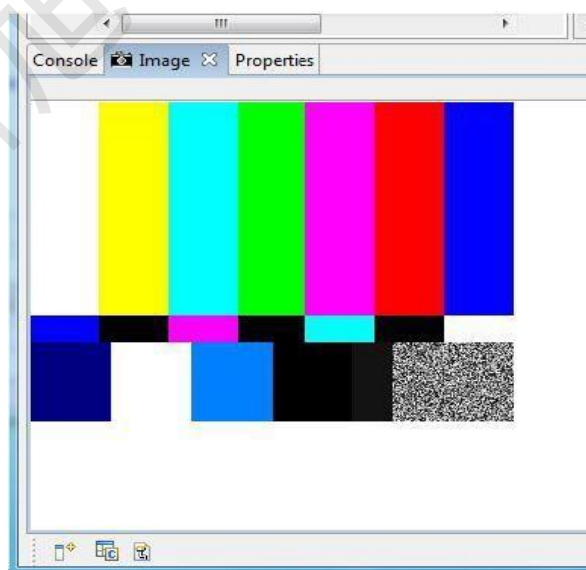


图 6-7-2

另外还提供了其他一些调试器功能和视图。强烈建议您尝试一下所有这些选项，了解调试器的所有功能。

南京研旭电气科技有限公司