

CCS6.0 教程

目录

CCS6.0 教程.....	1
第一章 CCS6.0 软件的安装.....	2
第二章 CCS 与仿真器的连接.....	7
2.1 定义工作区目录.....	7
2.2 建立目标板配置环境.....	8
2.3 连接目标板.....	11
第三章 创建 CCS6.0 工程.....	13
3.1 创建工程.....	13
3.2 生成项目.....	15
第四章 工程导入.....	15
4.1 CCS3.3 工程的导入.....	15
4.2 CCS 高版本工程的导入.....	23
第五章 CCS6.0 的仿真与烧写.....	24
5.1 CCS6.0 的仿真操作.....	24
5.2 CCS6.0 的烧写操作.....	28
第六章 CCS6.0 中一些常见的功能按钮.....	30
6.1 加载代码.....	32
6.2 监视变量和寄存器.....	28
6.3 反汇编以及源代码混合模式.....	33
6.4 内存查看器.....	34
6.5 管理断点.....	35
6.6 图形显示工具.....	37
6.7 图像显示工具.....	39

第一章 CCS6.0 软件的安装

首先我们需要来安装 TI DSP 的开发环境 CCS (Code Composer Studio)。如果您曾经使用过其他公司的仿真器产品。我们推荐使用 CCS6.0 版本, 因为暂时合众达公司的 XDS510PLUS 和 XDS560PLUS 仿真器只支持 CCS6.0, 再高版本兼容性不是很好。另外我们选用 XDS100 V2 仿真器也只能适用于 CCS4.0 及以上版本的开发环境, XDS100V3 仿真器只适用于 CCS5.0 及以上版本的开发环境。

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。双击 `ccs_setup_6.0.0.00190.exe` 文件, 出现如下图 1-1 所示界面:

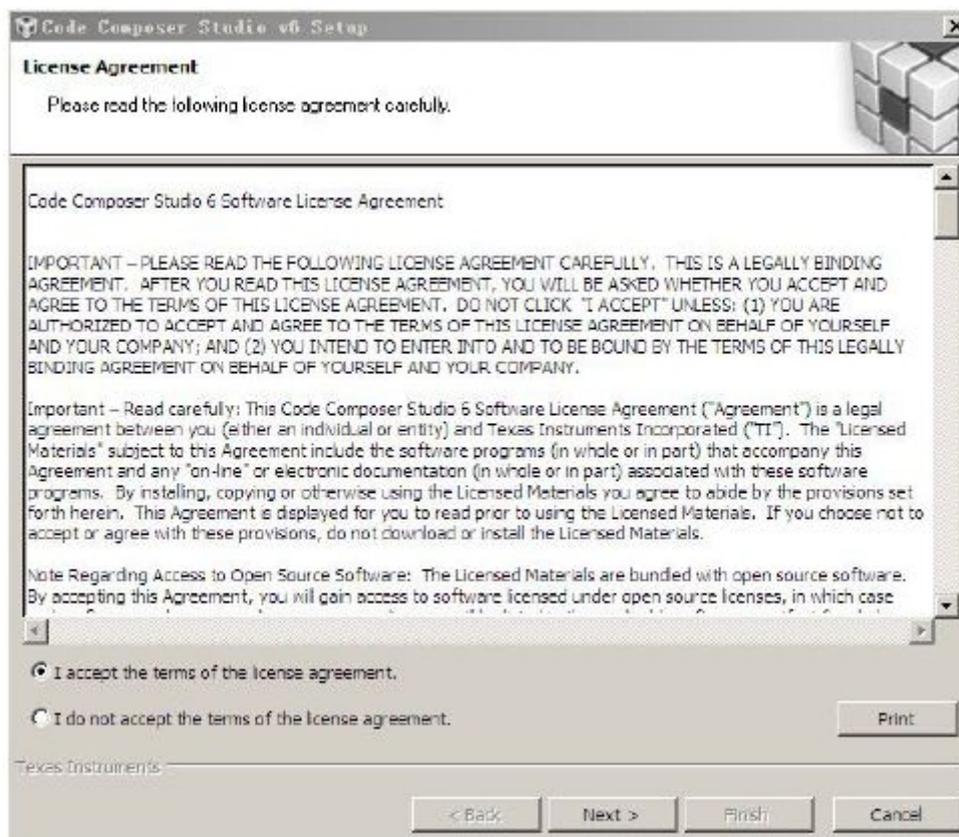


图 1-1

选择 “I accept the terms of the license agreement”, 点击 “Next” 如下图 1-2

所示:

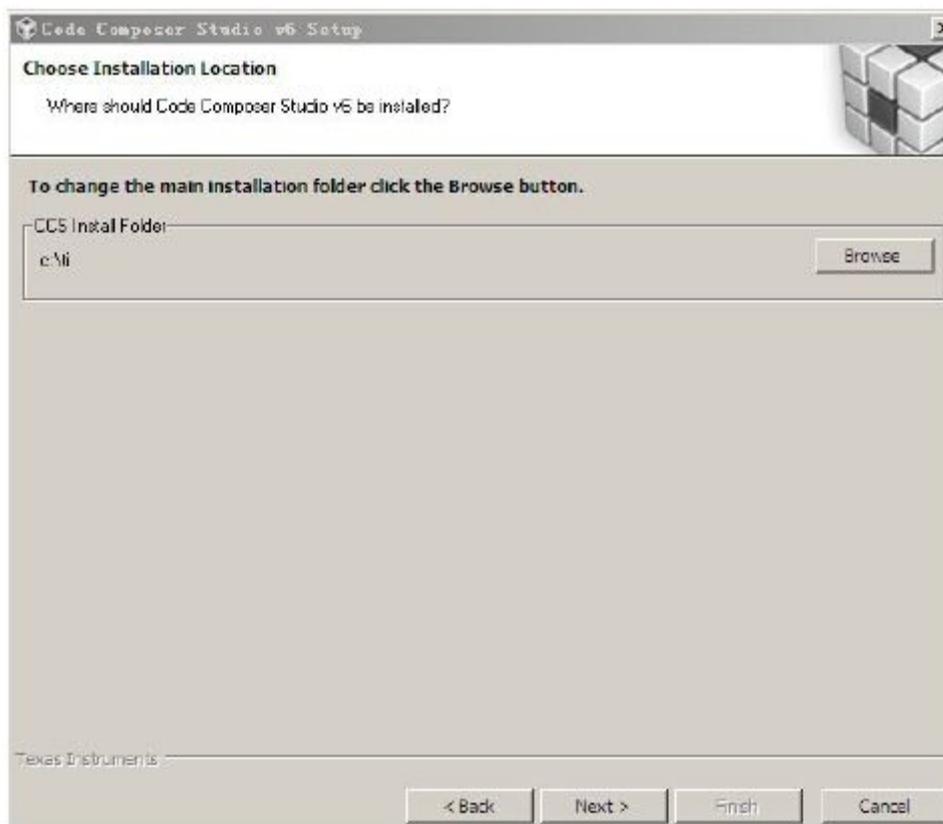


图 1-2

点击“Browse”选择安装路径（注意：路径不可以有中文），但推荐默认路径，点击“Next”见下图 1-3 所示：

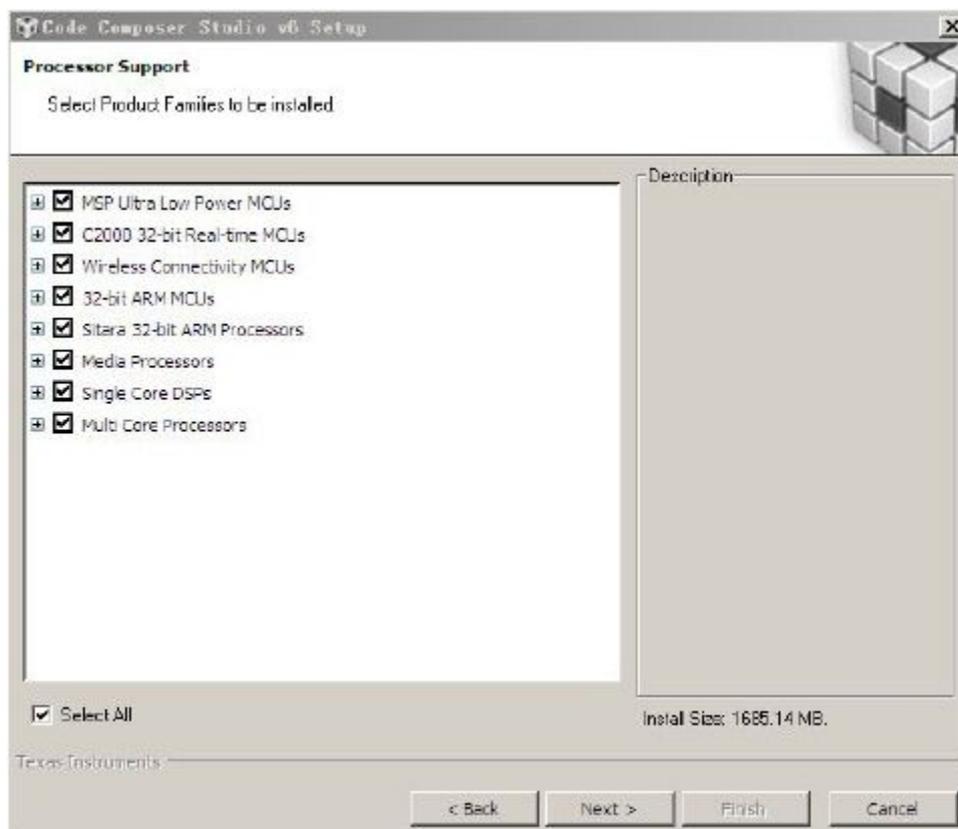


图 1-3

根据自己的需求选择索要安装的内容，这里选择“Select All”，然后点击“Next”见下图 1-4 所示：

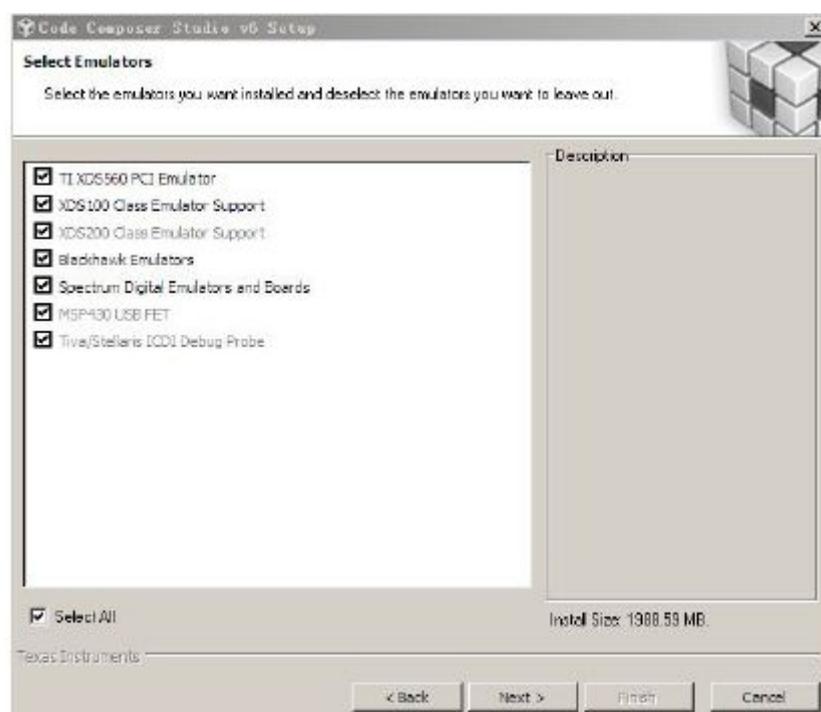


图 1-4

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 1-5 所示：



图 1-5

根据自己的需求选择，这里全不选，然后点击“Finish”。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。

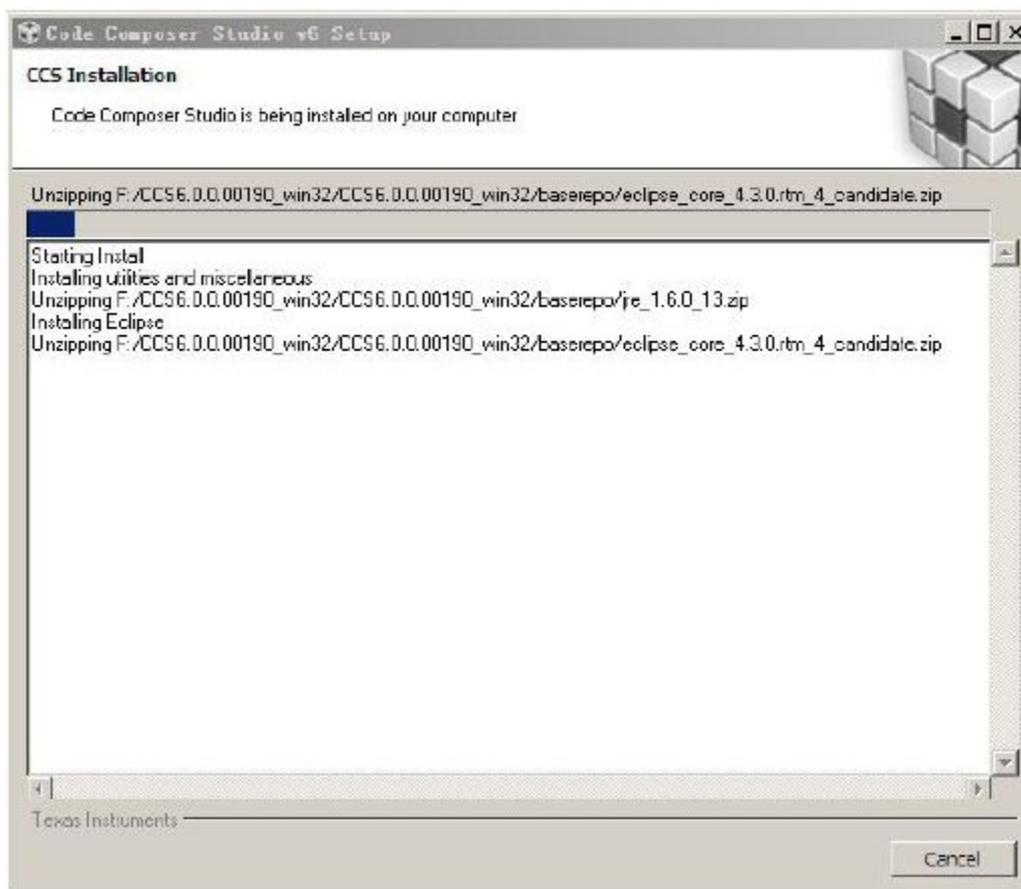


图 1-6

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 XDS100 V2 仿真器的驱动已经做到了 CCSV6 中了,所以当用户安装 CCSV6 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而且可以使用后,用户就可以使用仿真器对目标板进行仿真操作。

下面我们以 YX-28335 开发板为例为大家说明如何使用 XDS100 V2 对目标板进行仿真。至于其他型号 DSP,其操作过程都是一样的。

2.1 定义工作区目录

CCSv6 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。

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,但可以任意选择其位置。每次执行 CCSv6 都会要求工作区目录。如果计划对所有项目使用一个目录,只需选中“Use this as the default and do not ask again (默认使用此目录且不再询问)”选项,如下图 2-1-1 所示:

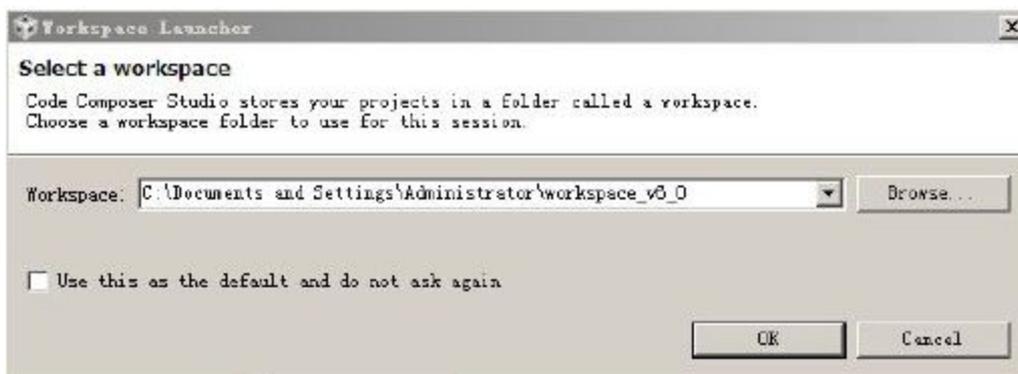


图 2-1-1

更改工作区目录详见下图 2-1-2,在工具栏中点击“File->Switch

Workspace→Others” 然后自己重新定义(注意: 目录不可以有中文) :

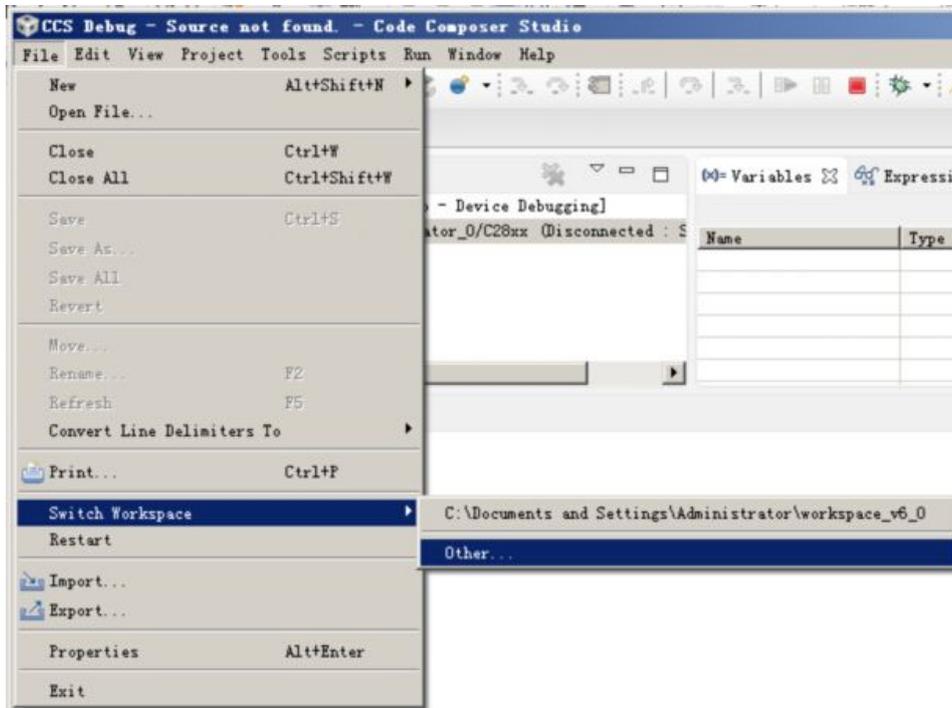


图 2-1-2

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 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 CCSv6 中建立:

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

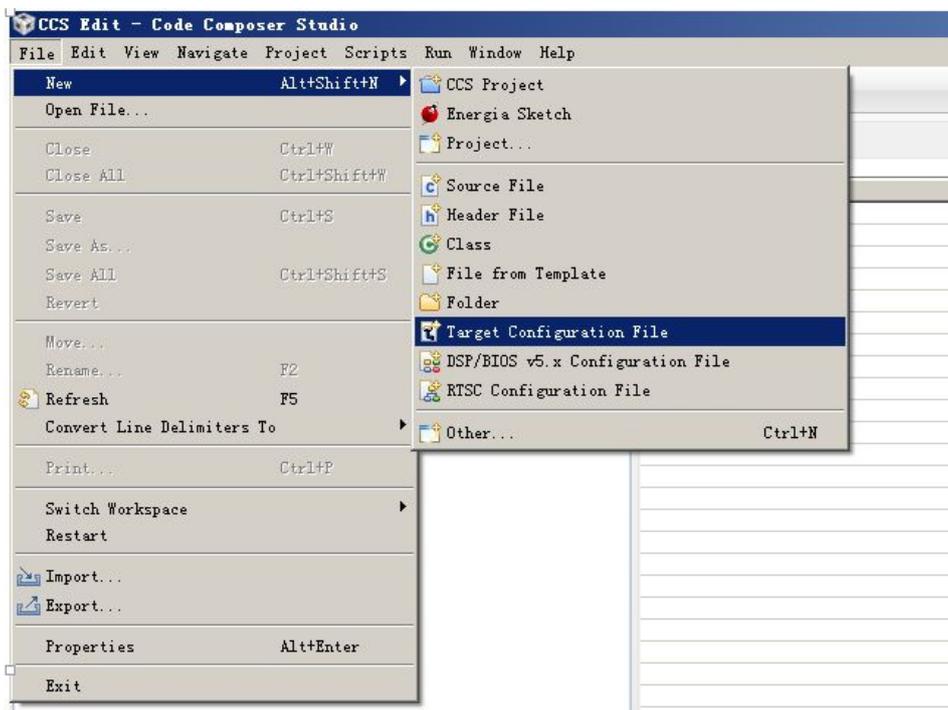


图 2-2-1

为此配置命名 “f28335_xds100_v2.ccxml” (可自行命名), 点击 “Finish”, 如下图 2-2-2 所示:

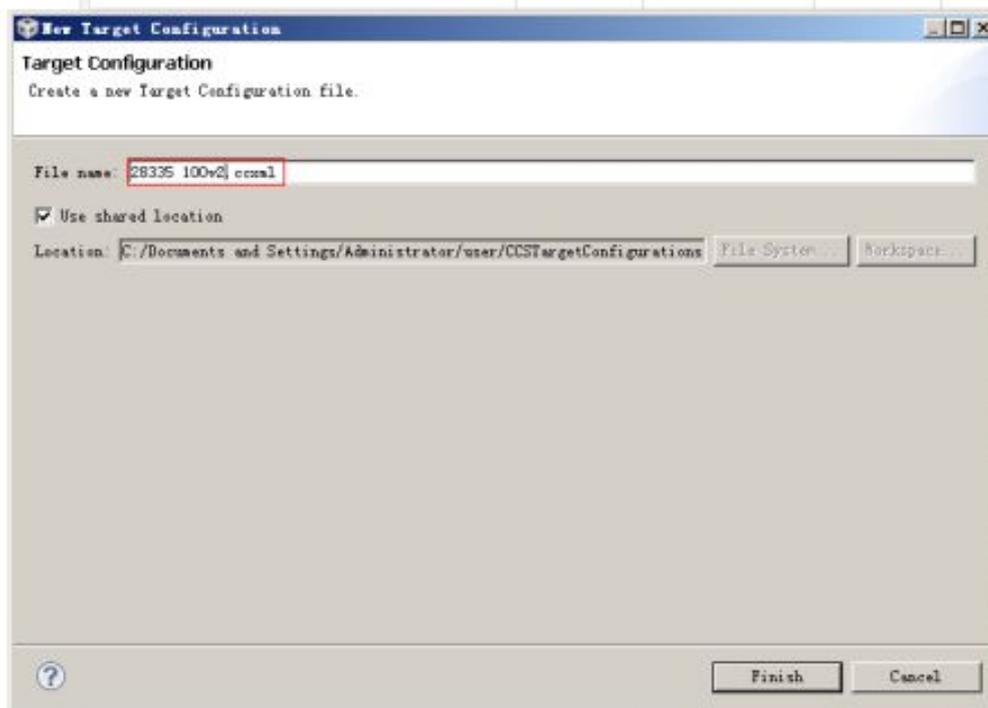


图 2-2-2

在“connection（调试器类型）”一栏中选择“Texas Instruments XDS100v2 USB Emulator”或“Texas Instruments XDS100v3 USB Emulator”，“Device（芯片类型）”一栏中选择“TMS320F28335”，如下图 2-2-3 所示：

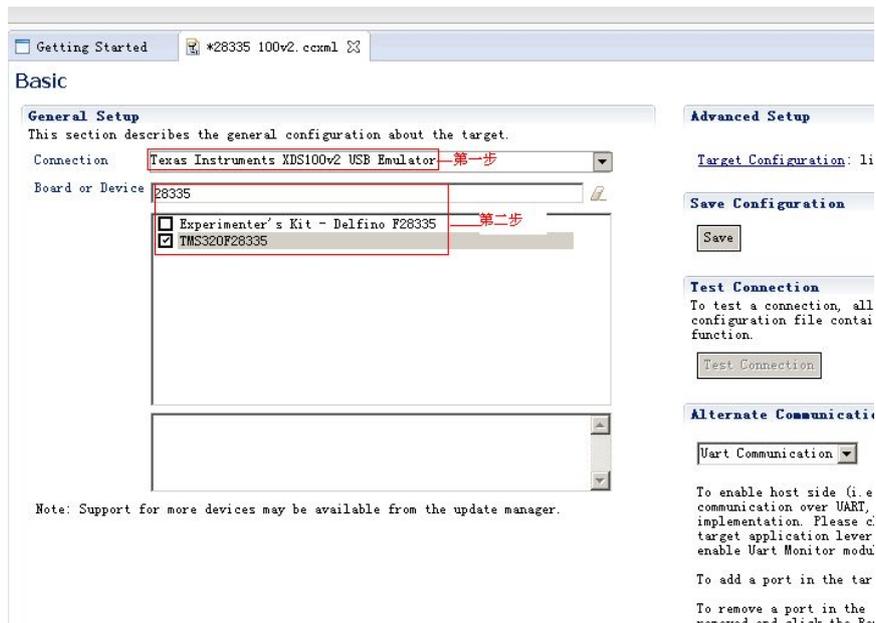


图 2-2-3

如果使用默认的 GEL 文件，那么请点击“Save”即可；如果使用自己的 GEL 文件，那么请点击蓝色字体的“Target Configuration”出现以下图 2-2-4 界面：

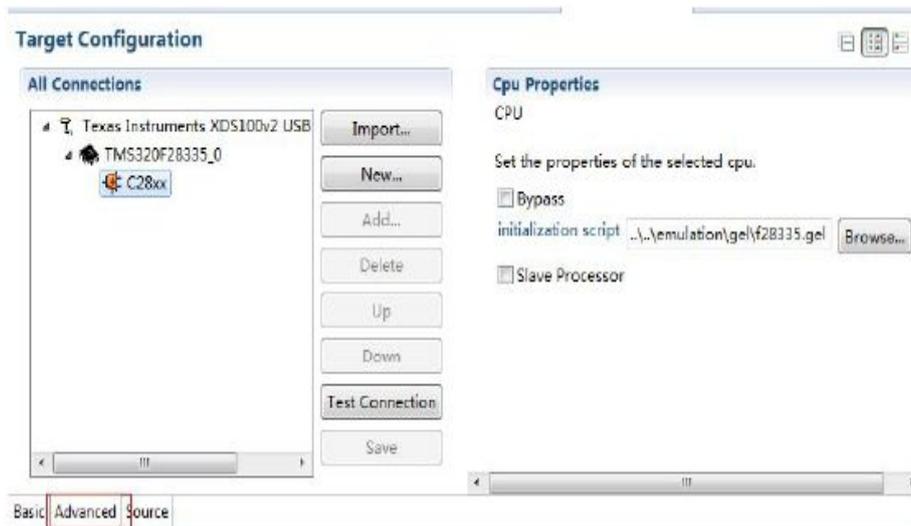


图 2-2-4

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 连接目标板

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”详细操作见下图 2-3-1 所示:

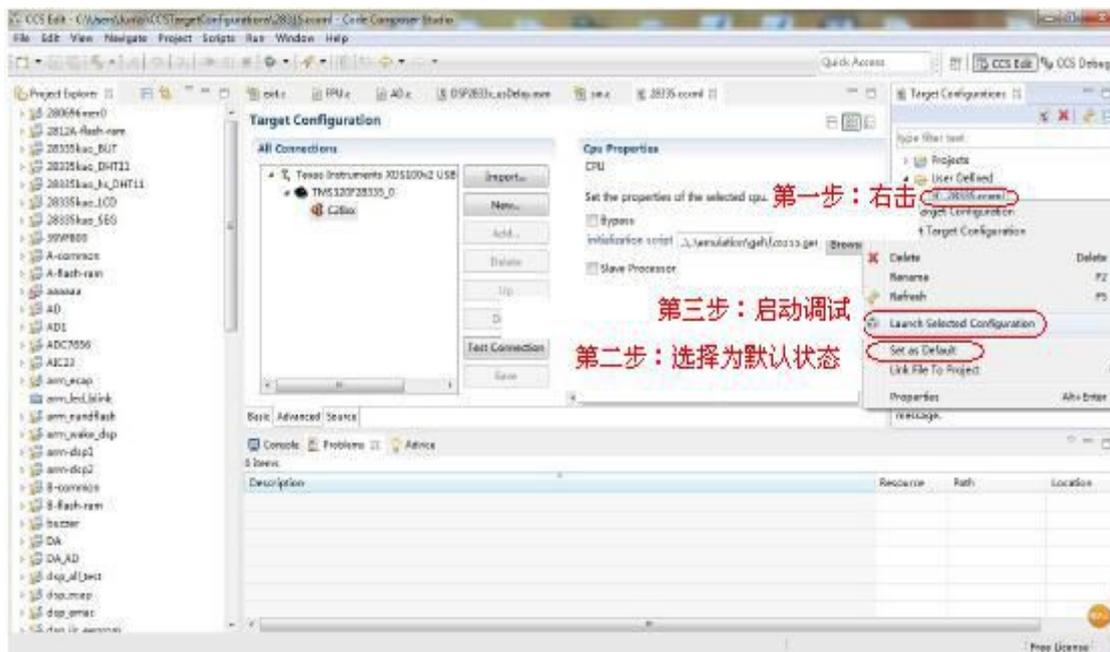


图 2-3-1

当启动成功后,在工具栏中点击“Run->Connect Target”即可连接目标板,见下图 2-3-2 所示界面:

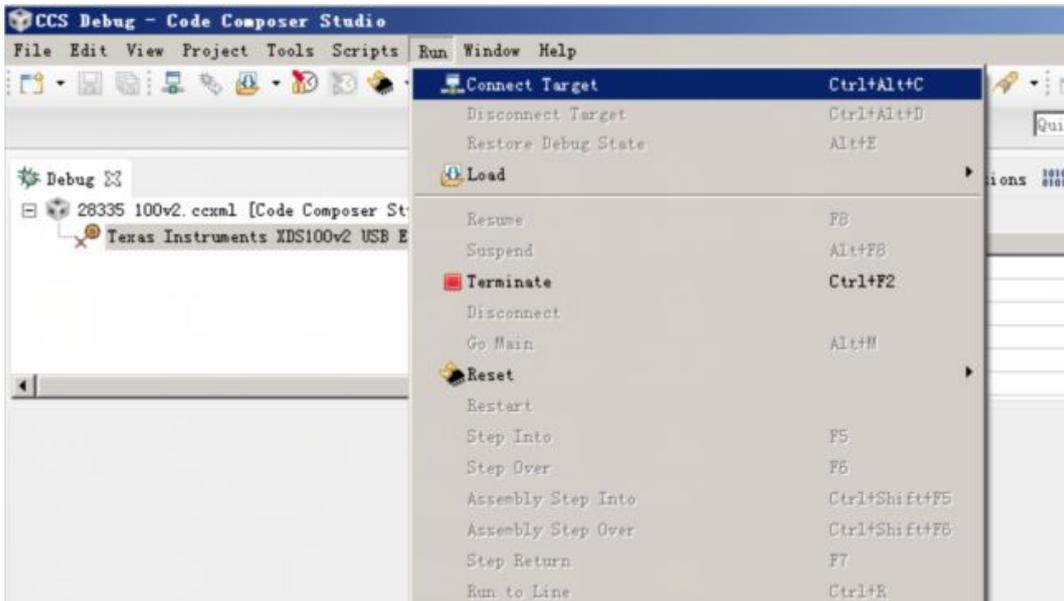


图 2-3-2

出现下图 2-3-3 所示的界面即表明连接成功，接下去，用户可以自行仿真实验。

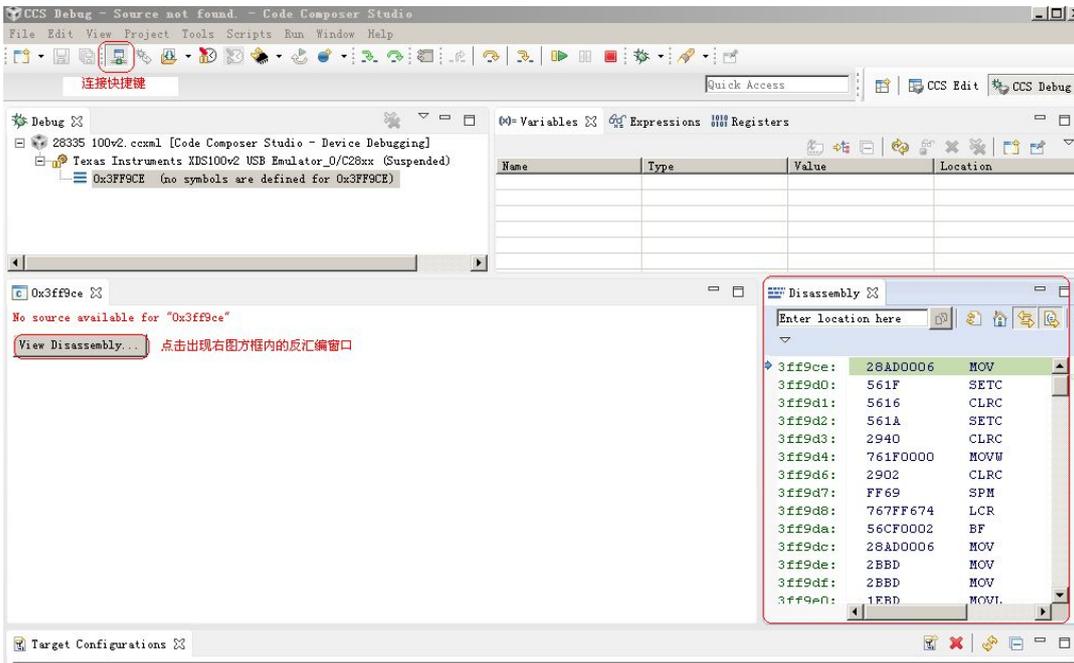


图 2-3-3

第三章 创建 CCS6.0 工程

3.1 创建工程

欢迎屏幕关闭之后,将会显示下面的工作区,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 3-1 所示:

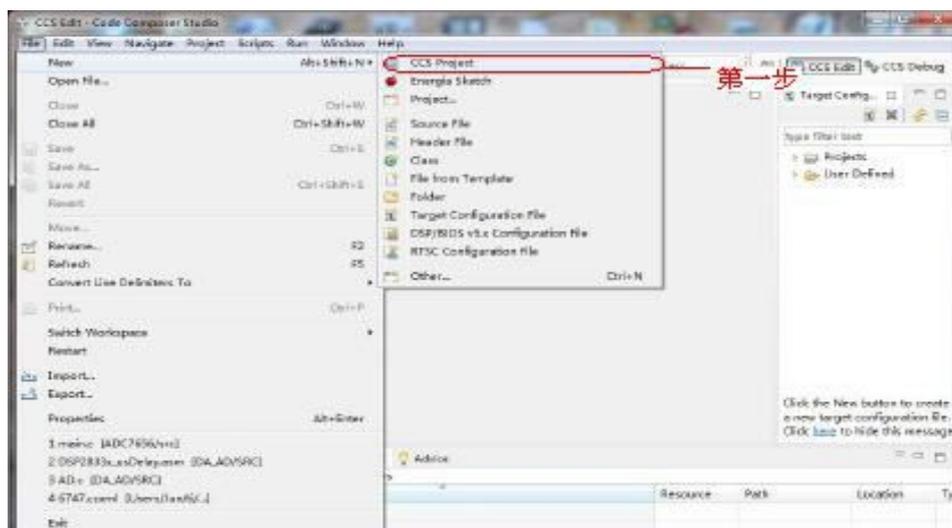


图 3-1

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。取消选中该选项可以选择一个新位置(使用“Browse... (浏览...)”按钮)。将项目命名为“Sinewave”。

③ 在“Target”菜单中选择要使用的芯片类型。

④ 在“Connetion”里选择调试器。

⑤ 单击“Finish (完成)”创建项目。 , 如下图 3-2 所示:

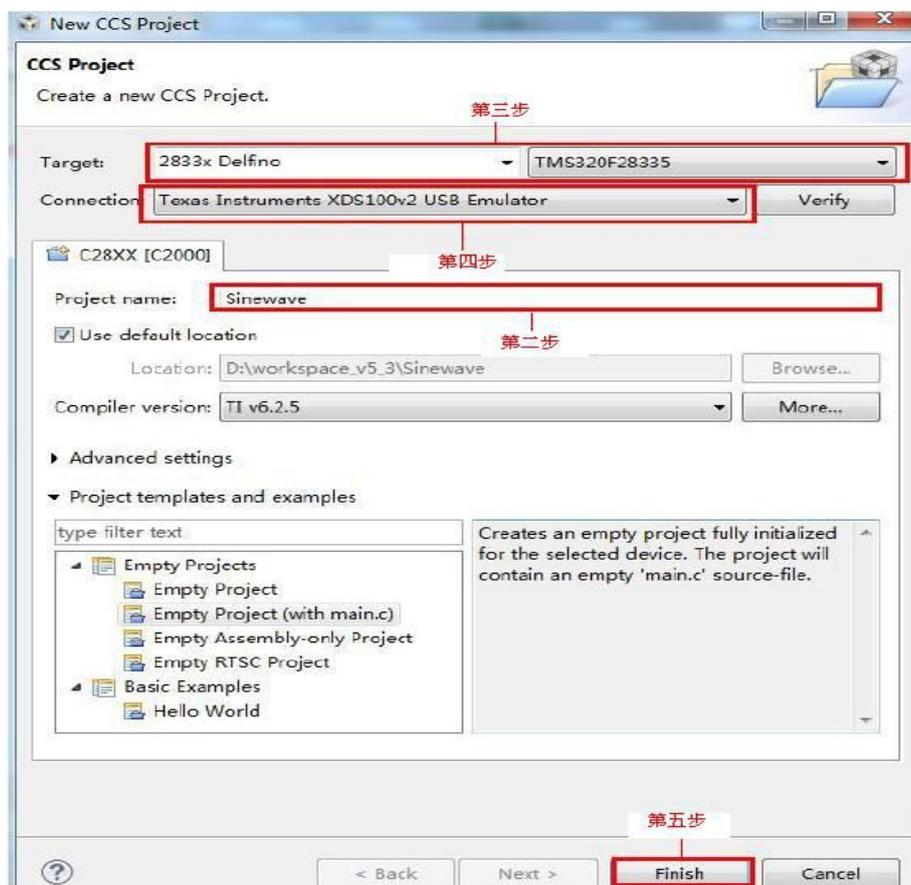


图 3-2

⑥ 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)”。

⑦ 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。

3.2 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。只需转到菜单“Project -> Build Active Project (项目->生成活动项目)”。“Rebuild Active Project (重新生成活动项目)”选项可重新生成所有源文件和引用的项目。不过如果项目较大，这可能是一个漫长的过程。

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。

第四章 工程导入

4.1 CCS3.3 工程的导入

在“C/C++ Projects (C/C++ 项目)” (路径：工具栏 Window->Show View->Other 下的 C/C++->C/C++ Projects) 即“CCS Edit”视图下，单击工具栏的“Project->Import Legacy CCSv3.3 Projects...” (工程->导入 CCS3.3 工程)” 见下图 4-1-1 所示：

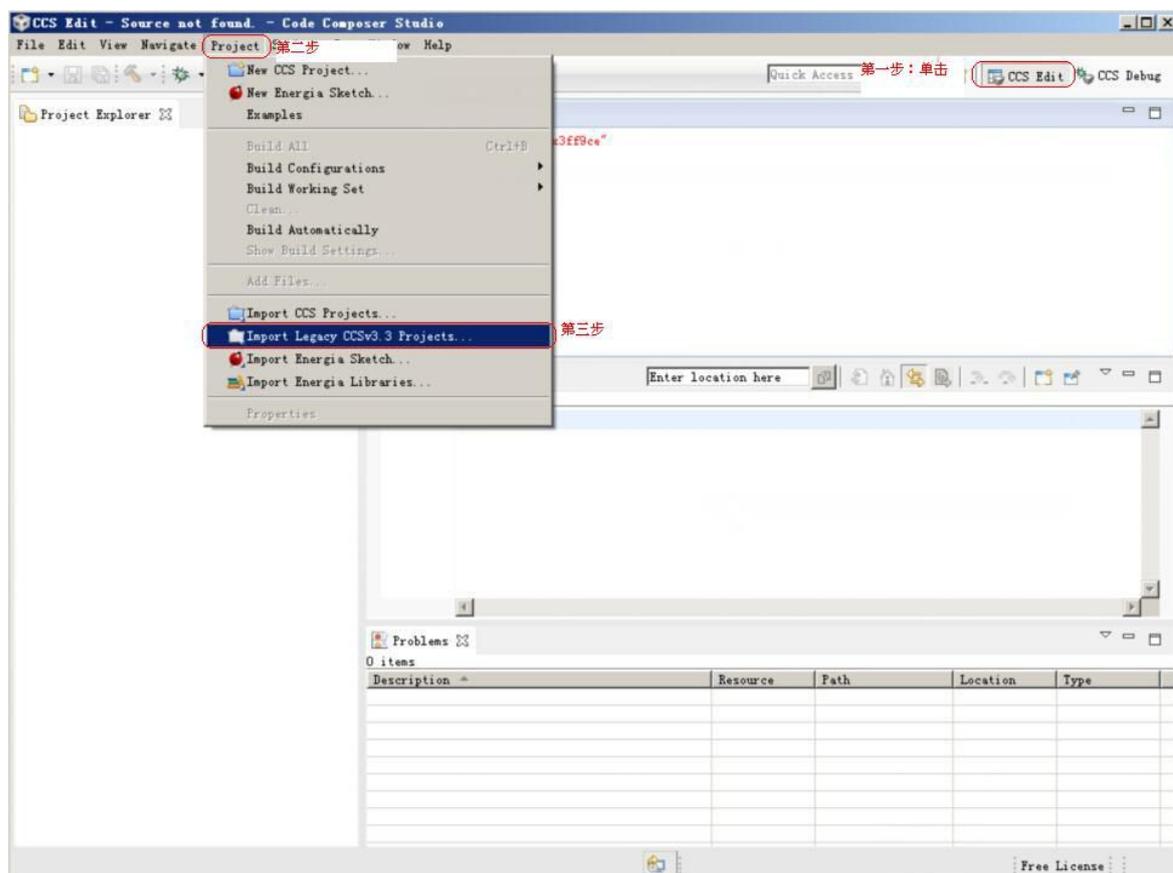


图 4-1-1

然后如下图 4-1-2 所示操作:

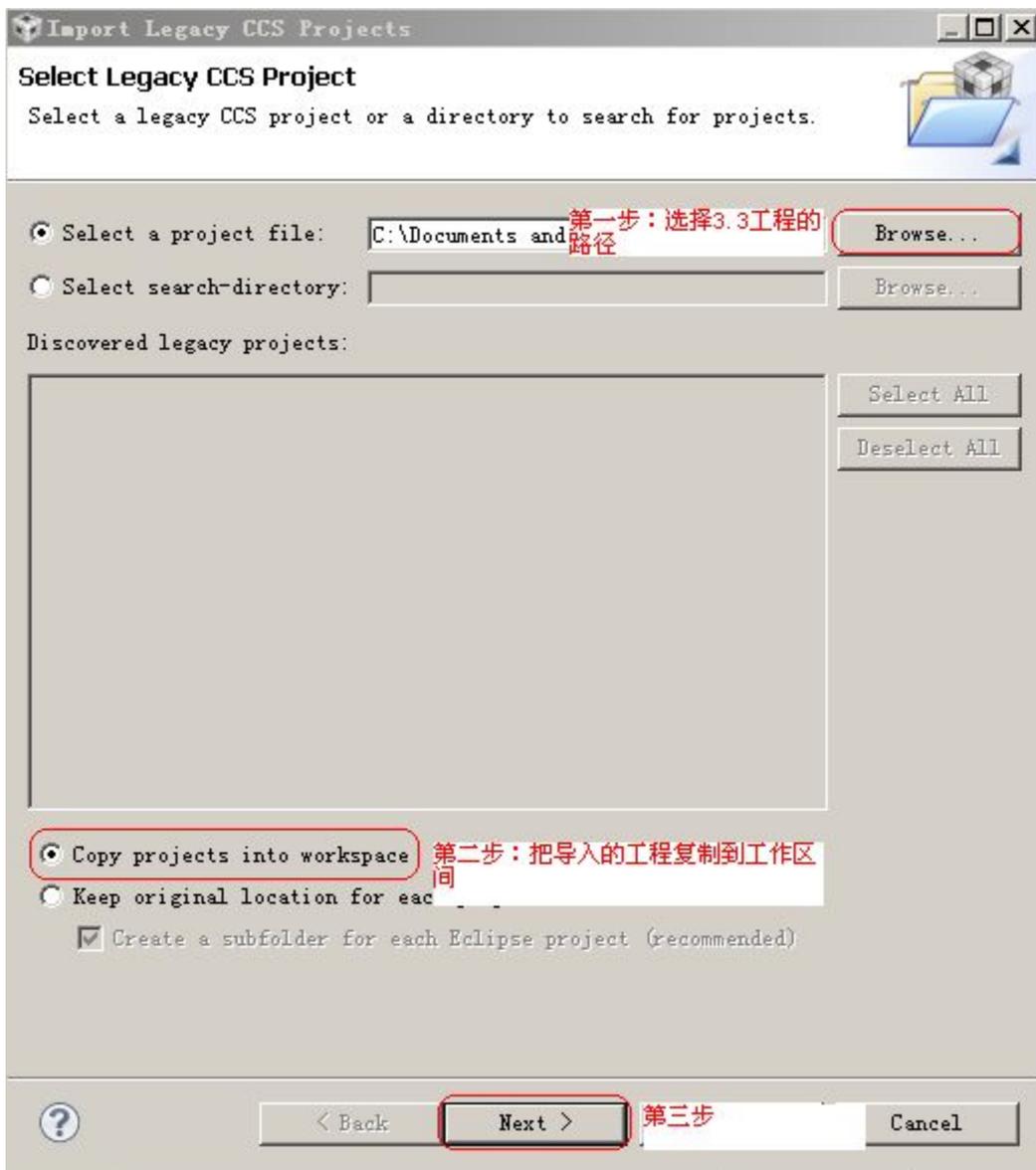


图 4-1-2

然后点击“Finish”，如图 4-1-3；

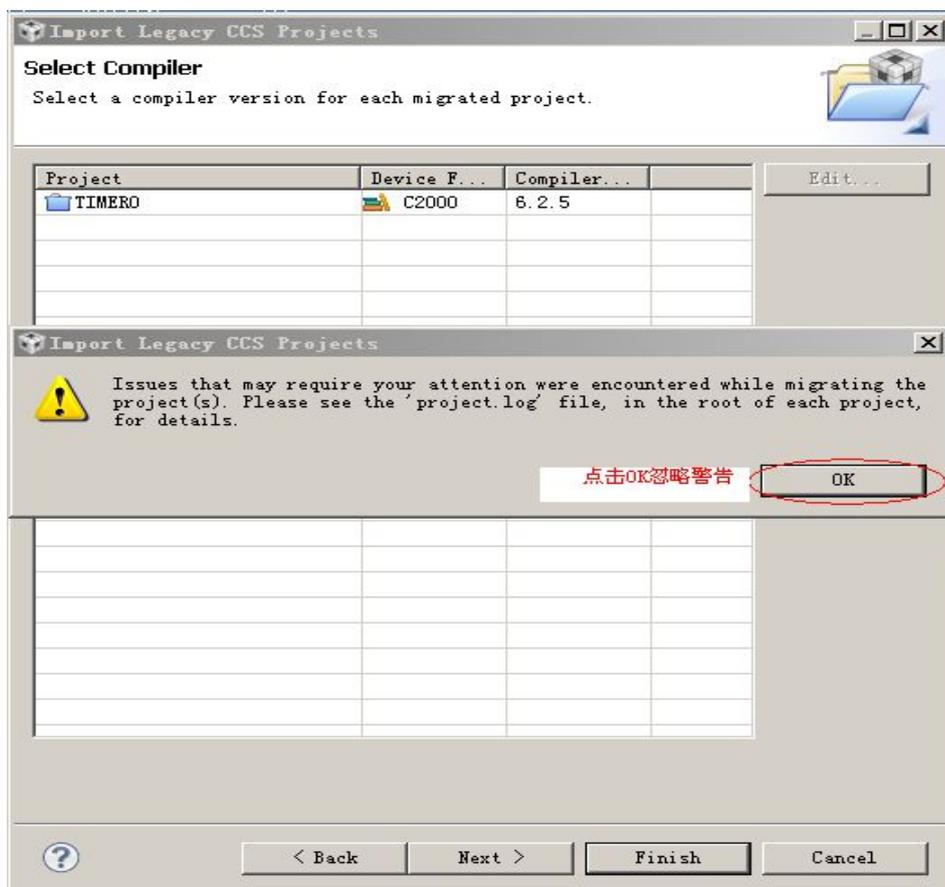


图 4-1-4

这样 CCS3.3 的工程就已经导入完成了，下面就是设置导入后工程的属性（路径：工具栏->Window->Show View->Project Explorer, 切换出界面后右击工程选择最后一项 Properties），具体操作如下图 4-1-5 所示：

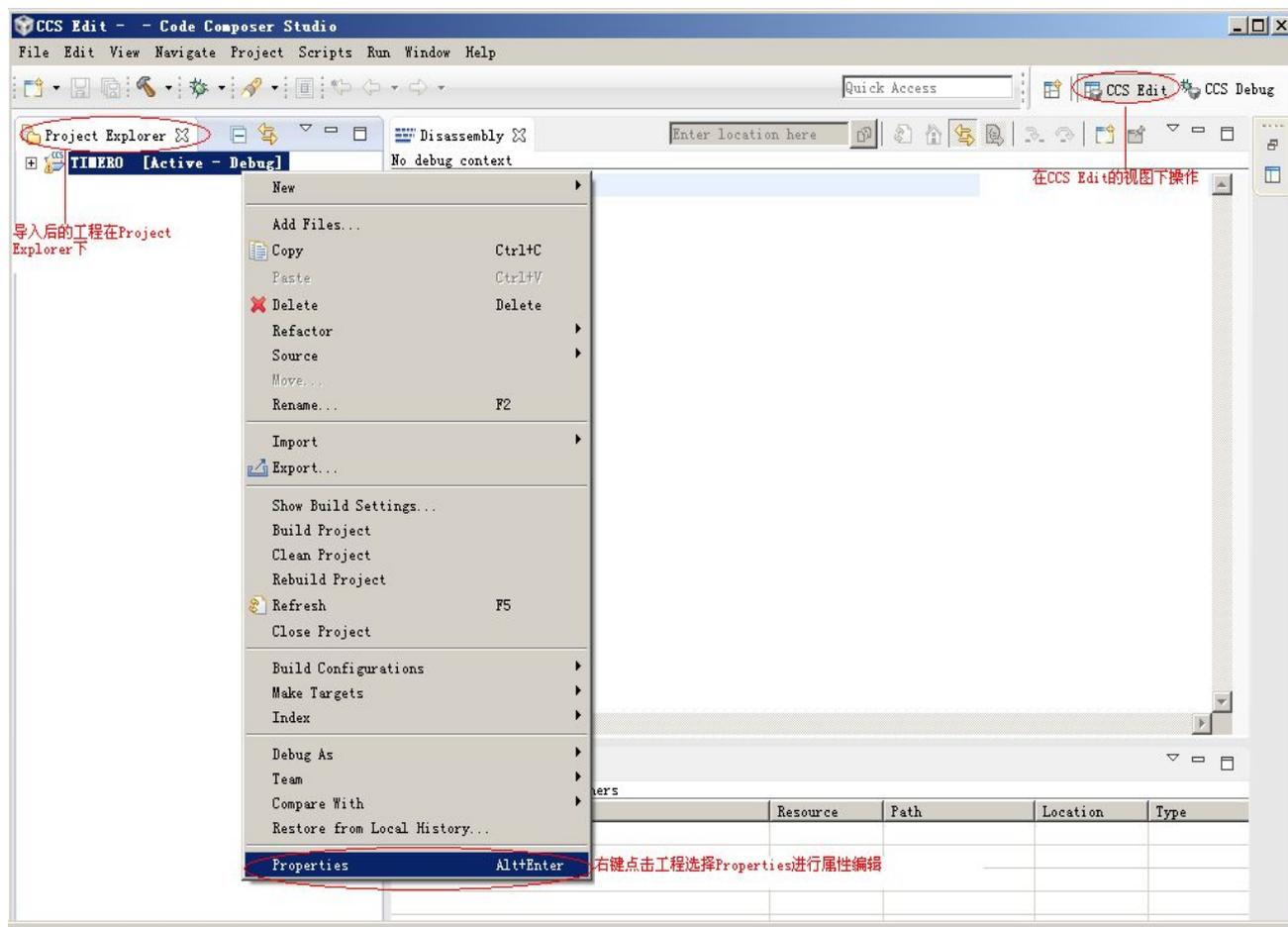


图 4-1-5

属性编辑需要设置两种属性，一种是修改头文件路径，这里我们把路径修改成绝对路径，见下图 4-1-6 所示操作：

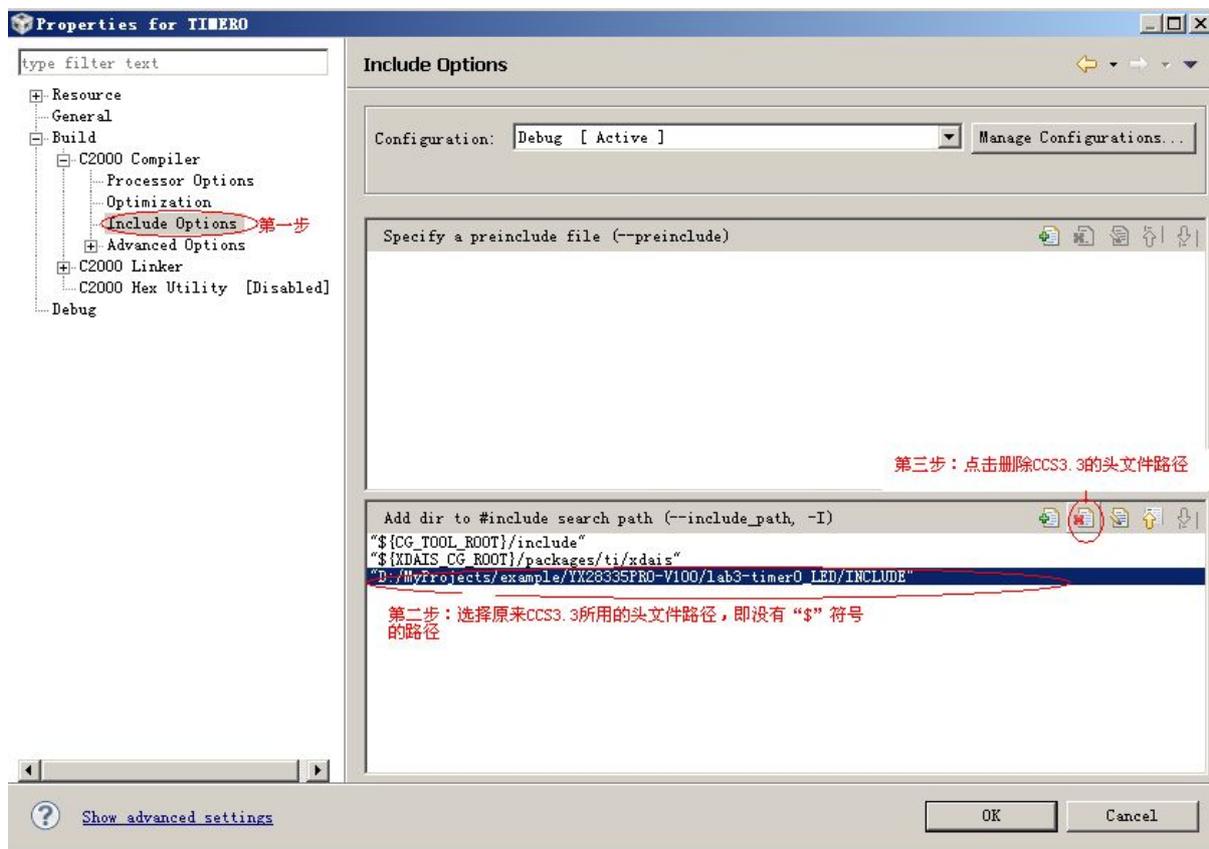


图 4-1-6

然后重新添加 CCS6.0 的头文件路径点击 “” 按钮，在弹出的窗口中点击 “Workspace” 按钮，如下图 4-1-7 所示：

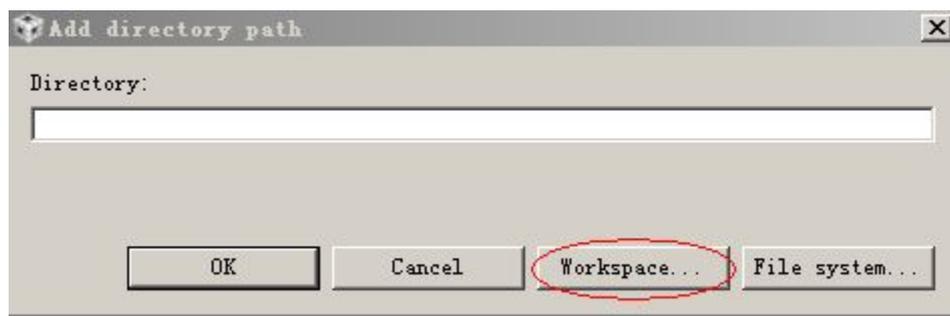


图 4-1-7

在弹出的窗口中将导入的工程展开，选中 “Include” 文件夹，再单击 “OK”，即可完成添加头文件路径，如下图 4-1-8 所示：



图 4-1-8

添加完成后，在原来的窗口也点击“OK”，完成添加；

头文件路径添加后，还要再设置个属性，具体操作如下图 4-1-9 所示：

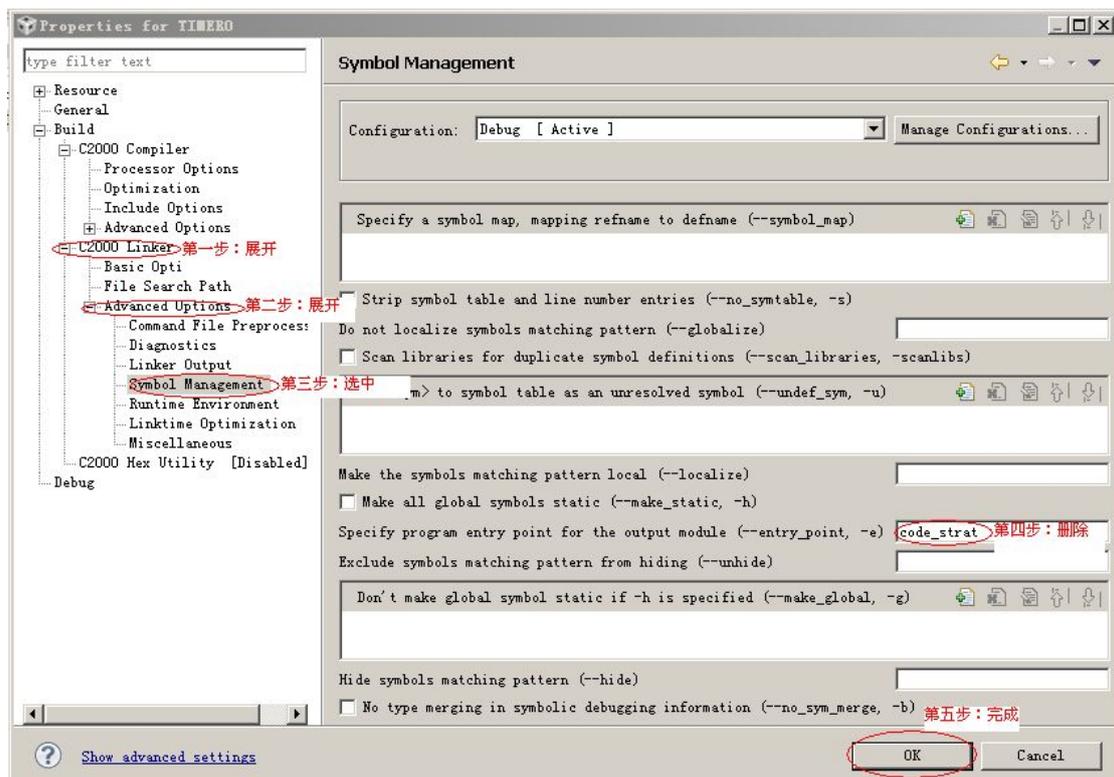


图 4-1-9

最后属性设置完成，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 高版本工程的导入

由于我们光盘附赠的例程虽然已经是 CCS3.3 以上的版本，但不可以直接拷贝到“workspace（工作区间）”使用，还是需要导入到工作区间才可以使用，下面将介绍如何导入 CCS 高版本的工程。

在“C/C++ Projects（C/C++ 项目）”（路径：工具栏 Window→Show View→Other 下的 C/C++→C/C++ Projects）即“CCS Edit”视图下，单击工具栏的“Project→Import CCS Projects...”（工程→导入 CCS 工程）见下图 4-2-1 所示：

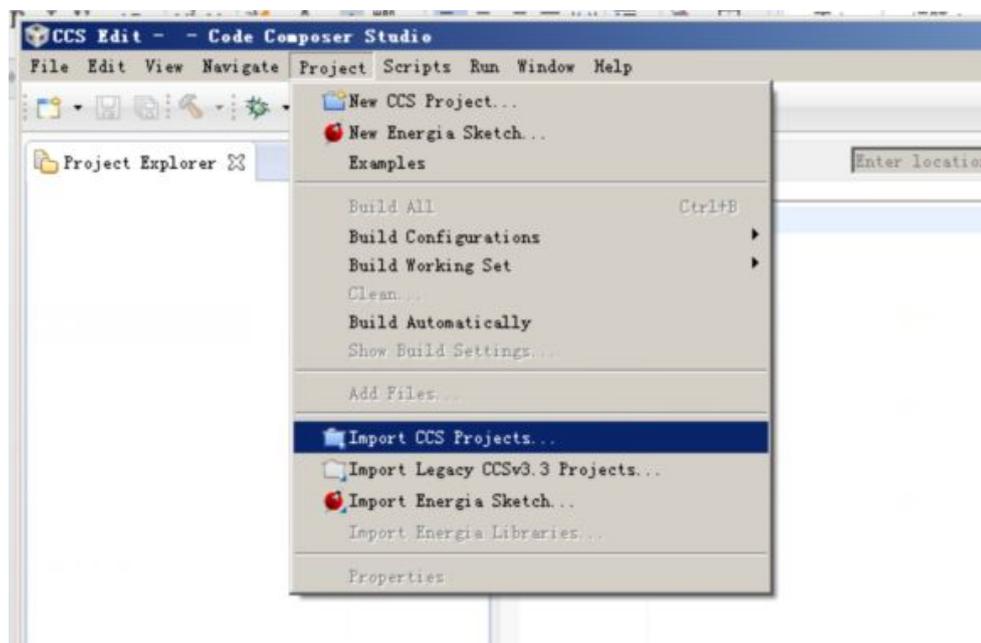


图 4-2-1

在弹出的窗口中作如下操作（注意：如果操作过程中出现和 CCS3.3 导入时一样的警告时，点击“OK”忽略），如图 4-2-2 所示：



图 4-2-2

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。

第五章 CCS6.0 的仿真与烧写

5.1 CCS6.0 的仿真操作

在例程导入后，下面就该进行 CCS 与开发板的联调，这里以 TMS320F28335 芯片为例进行仿真和烧写测试。

首先照第二章所做的说明，建立一个 TMS320F28335 芯片和 XDS100V2（或者 XDS100V3）的配置文件，启动并连接上，然后将视图切换到“CCS Edit”下，将导入的工程的 CMD 文件从“28335.cmd（烧写所用 CMD 文件）”替换成“28335_RAM_lnk.cmd（仿真所用 CMD 文件）”（注意：28335.cmd 和 28335_RAM_lnk.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），然后右击工程选择“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”文件后即可进行仿真操作，详细操作见下图 5-1-1 和 5-1-2 所示：

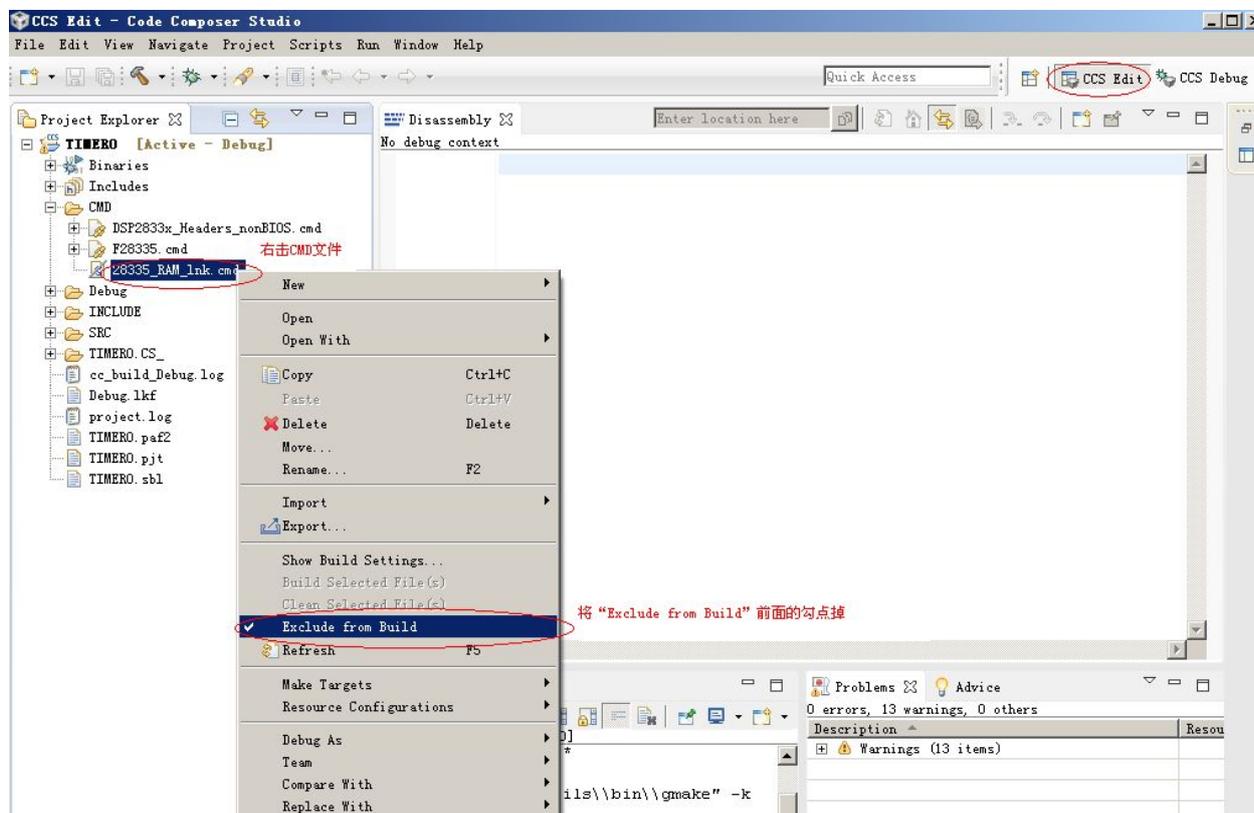


图 5-1-1

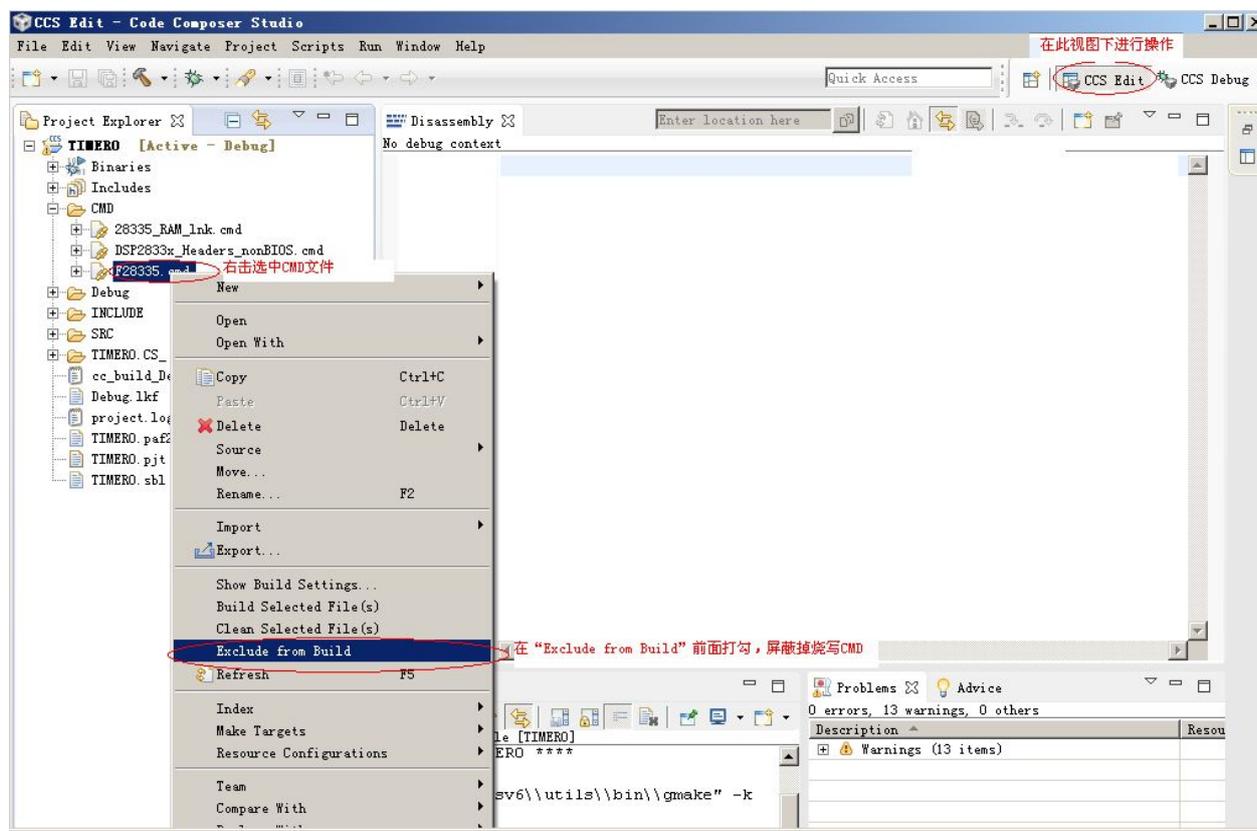


图 5-1-2

CMD 文件更改过后，将视图切换到“CCS Debug”视图下进行工程的加载（步骤：点击“工具栏 Run→Load→Load Program”），详见下图 5-1-3 所示：

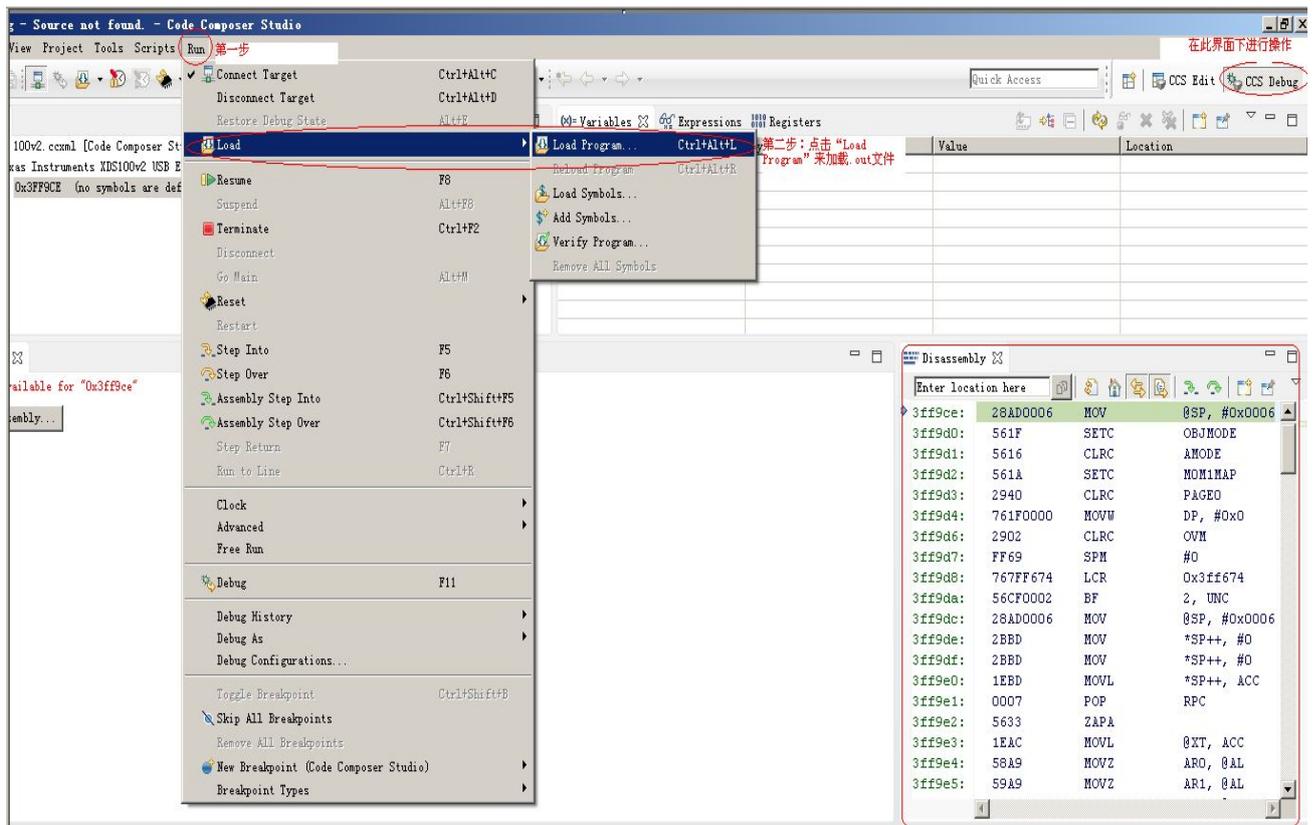


图 5-1-3

在出现的对话框里进行如下图 5-1-4 所示的操作:

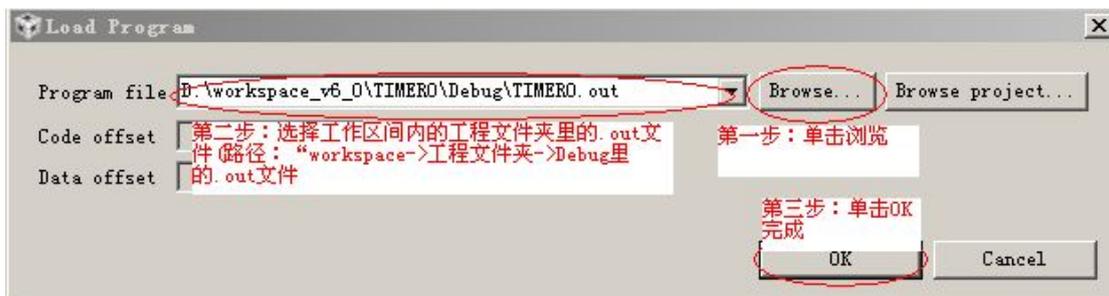


图 5-1-4

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:

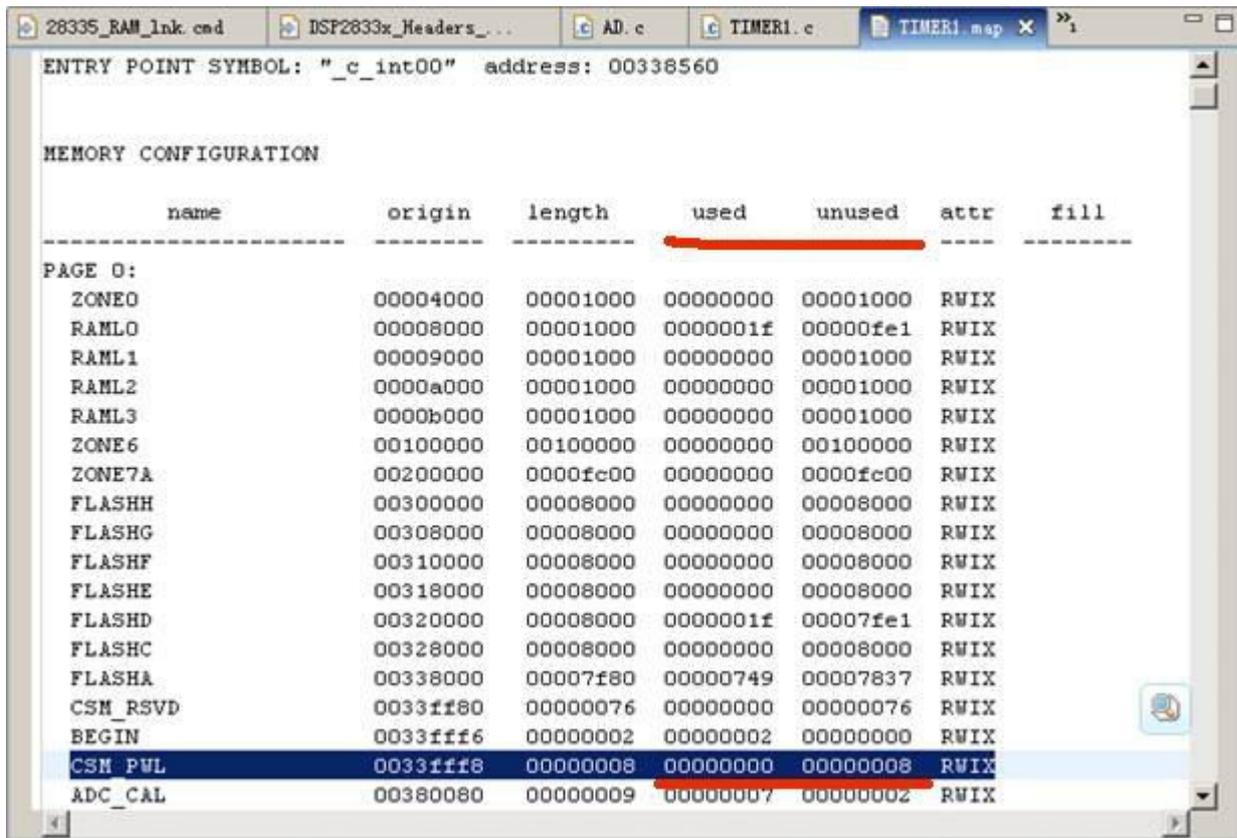


图 5-1-5

至此，仿真操作已进行完毕。

5.2 CCS6.0 的烧写操作

在烧写之前，展开工程下的 Debug 文件，双击打开 .map 文件，出现如下图 5-2-1 所示的内容：



ENTRY POINT SYMBOL: "_c_int00" address: 00338560

MEMORY CONFIGURATION

name	origin	length	used	unused	attr	fill
PAGE 0:						
ZONE0	00004000	00001000	00000000	00001000	RWIX	
RAML0	00008000	00001000	0000001f	00000fe1	RWIX	
RAML1	00009000	00001000	00000000	00001000	RWIX	
RAML2	0000a000	00001000	00000000	00001000	RWIX	
RAML3	0000b000	00001000	00000000	00001000	RWIX	
ZONE6	00100000	00100000	00000000	00100000	RWIX	
ZONE7A	00200000	0000fc00	00000000	0000fc00	RWIX	
FLASHH	00300000	00008000	00000000	00008000	RWIX	
FLASHG	00308000	00008000	00000000	00008000	RWIX	
FLASHF	00310000	00008000	00000000	00008000	RWIX	
FLASHE	00318000	00008000	00000000	00008000	RWIX	
FLASHD	00320000	00008000	0000001f	00007fe1	RWIX	
FLASHC	00328000	00008000	00000000	00008000	RWIX	
FLASHA	00338000	00007f80	00000749	00007837	RWIX	
CSM_RSVD	0033ff80	00000076	00000000	00000076	RWIX	
BEGIN	0033ffff	00000002	00000002	00000000	RWIX	
CSM_PWL	0033fff8	00000008	00000000	00000008	RWIX	
ADC_CAL	00380080	00000009	00000007	00000002	RWIX	

图 5-2-1

图中蓝底色一行为密码区域使用情况,如果在 used 于 unused 一栏下分别为 0 和 08 表示密码区域未使用,否则密码区域可能已经被使用,不可以烧写,若强行烧写将导致芯片锁死,此时可以自己重新新建工程编译。

CCS6.0 的烧写操作与仿真操作一样,只不过将导入的工程的 CMD 文件从“28335_RAM_lnk.cmd (仿真所用 CMD 文件)”替换成“28335.cmd (烧写所用 CMD 文件)”(注意: 28335.cmd 和 28335_RAM_lnk.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),然后右击工程选择“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,不需要点击运行。

第六章 CCS6.0 中一些常见的功能按钮

CCS6.0 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

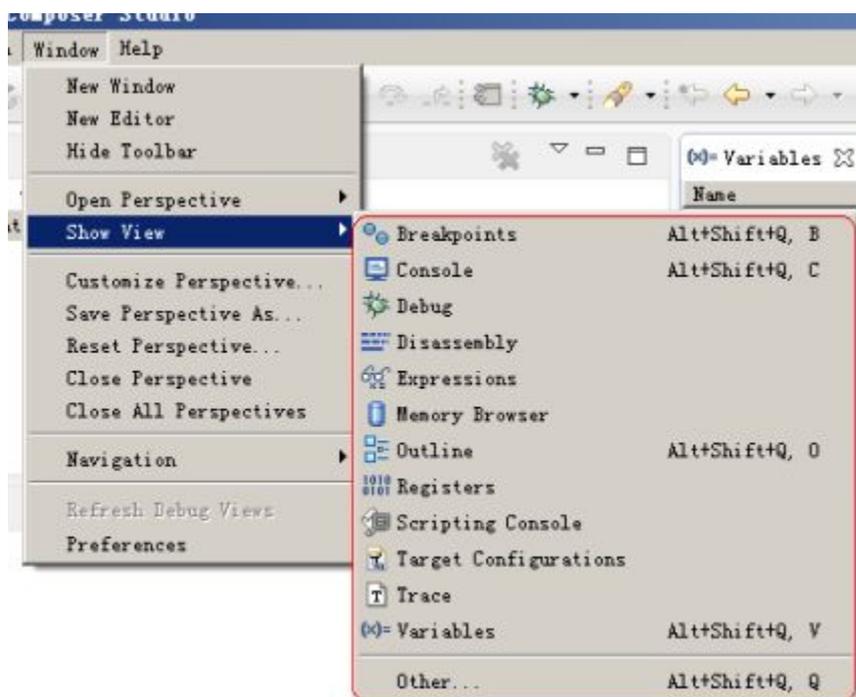


图 6-2

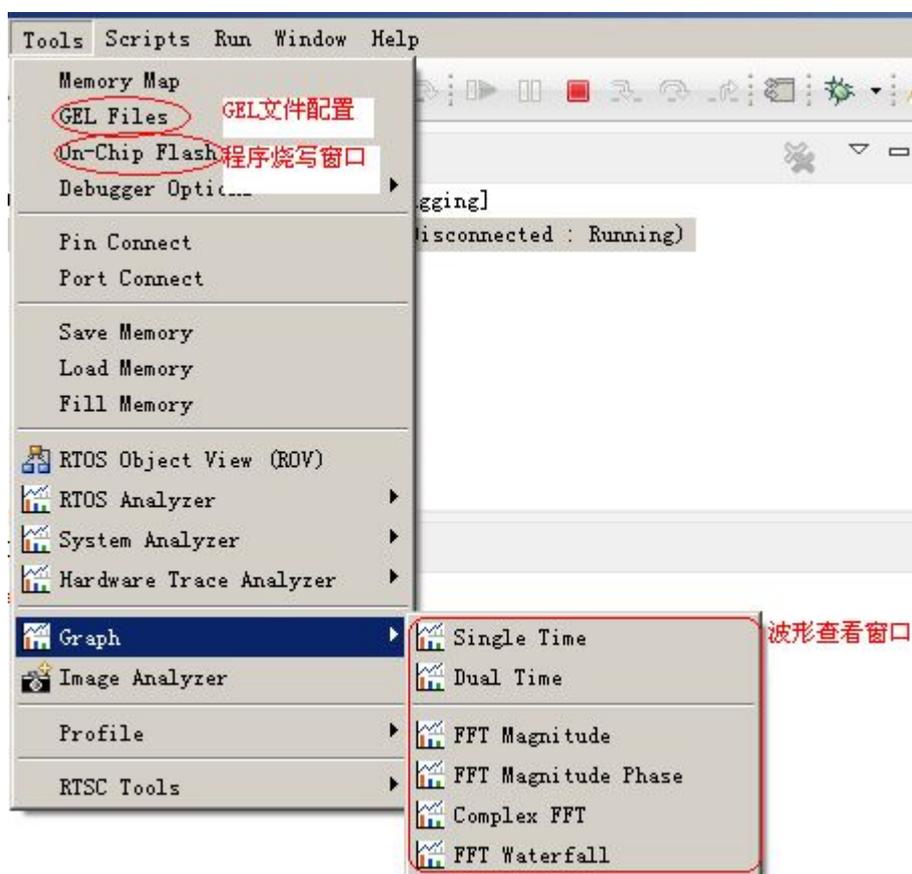


图 6-3

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

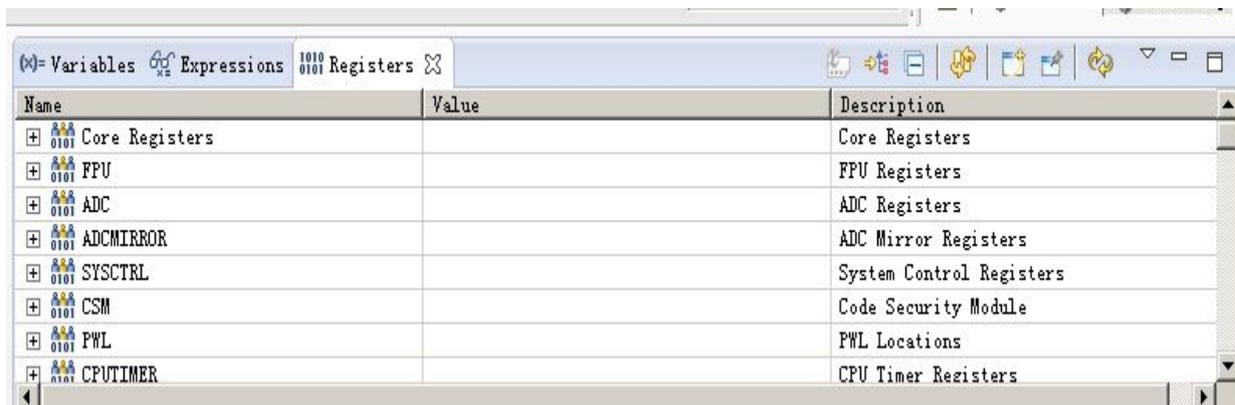


图 6-2-3

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：

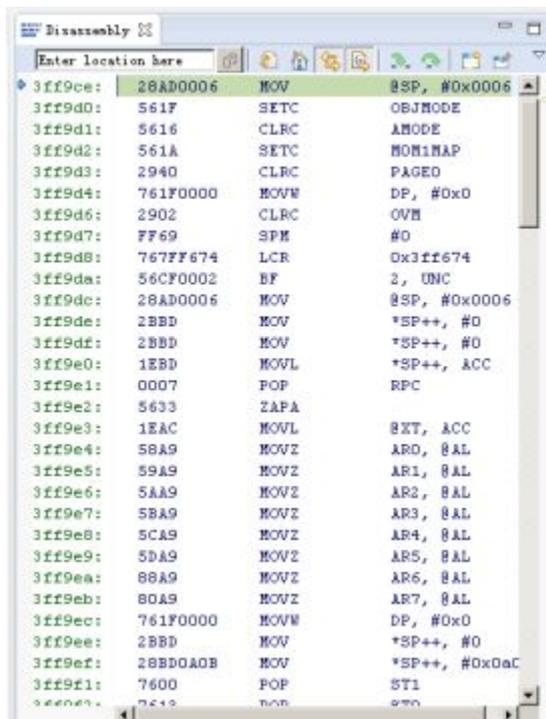


图 6-3-1

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：

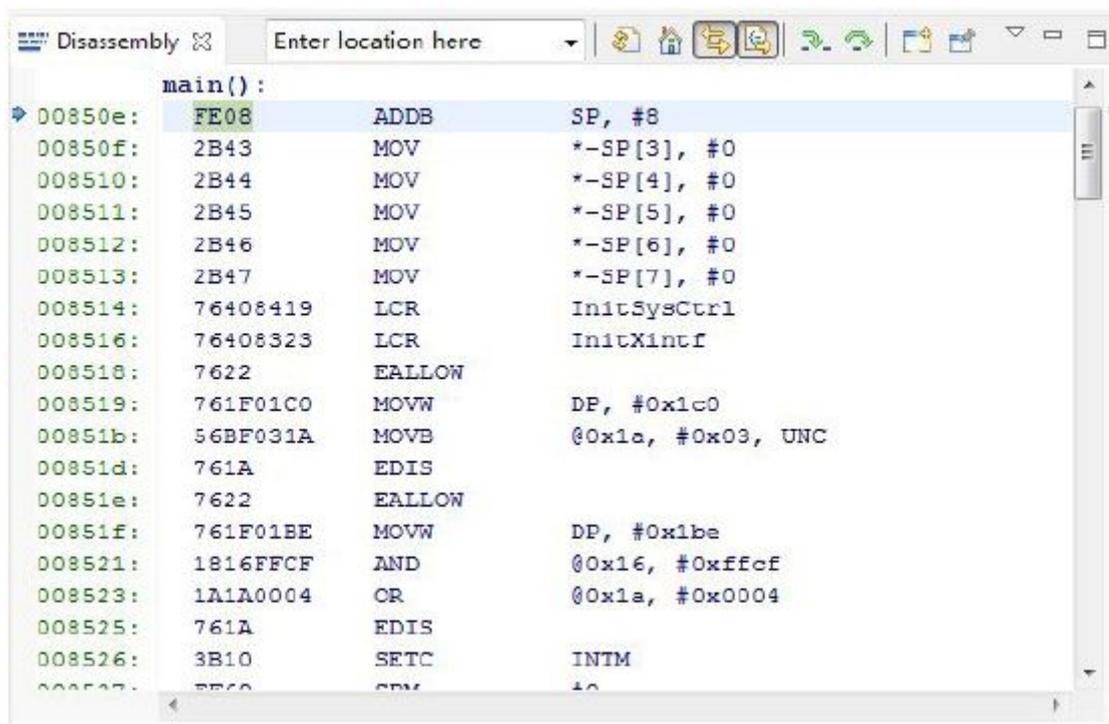


图 6-3-2

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：

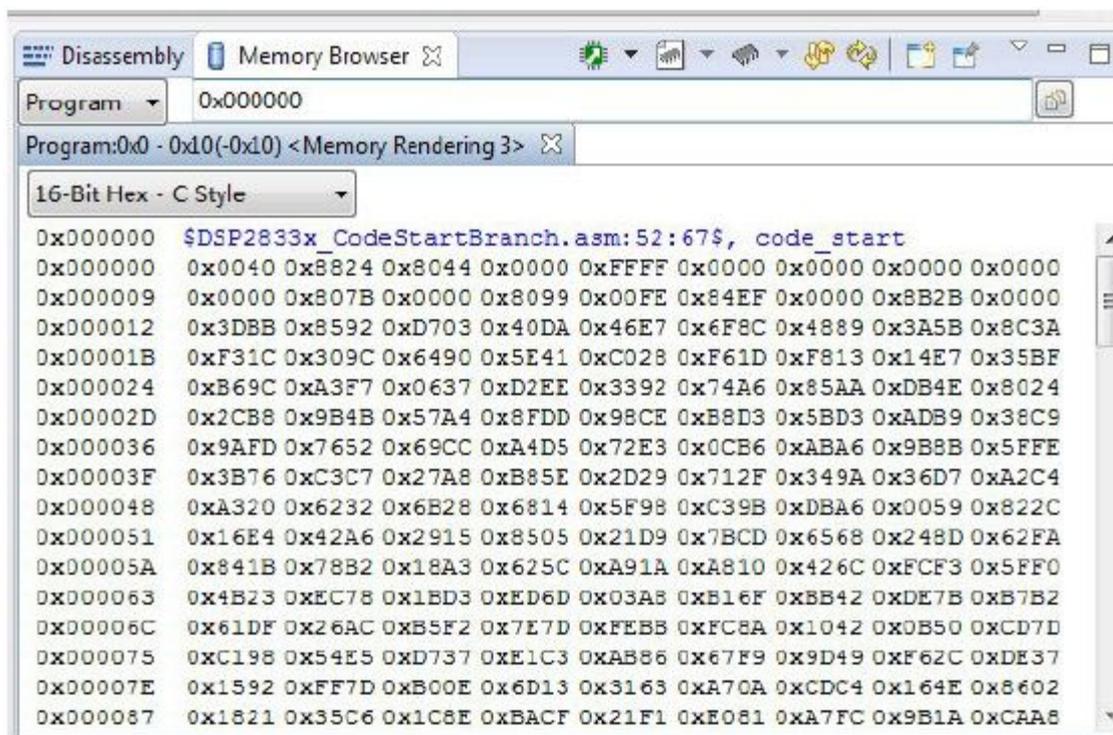


图 6-4-1

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，CCSv6 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：

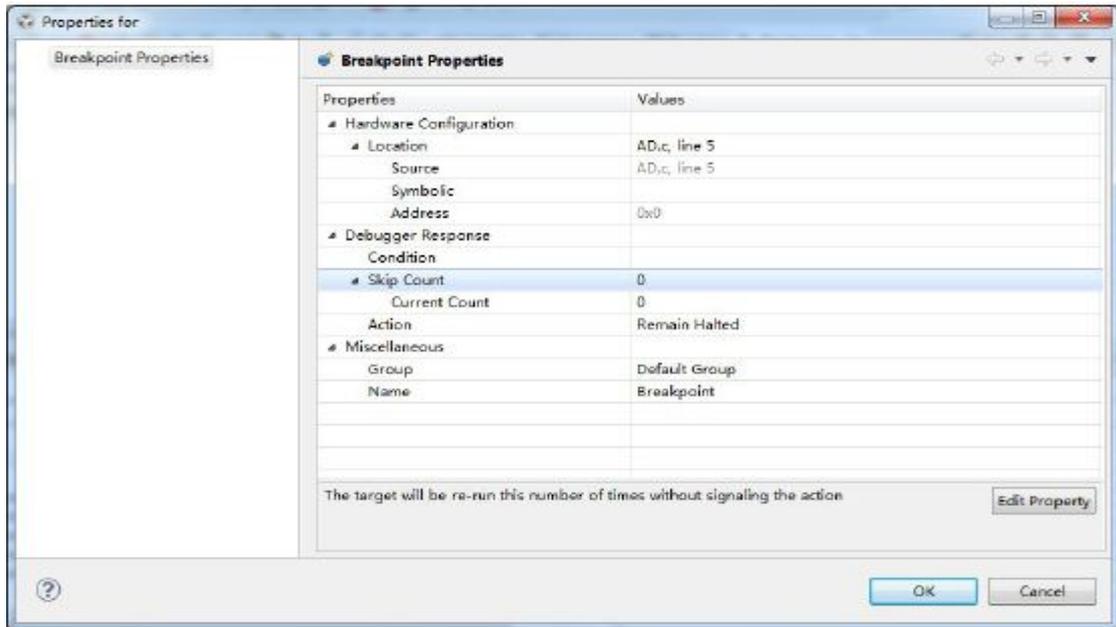


图 6-5-1

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。

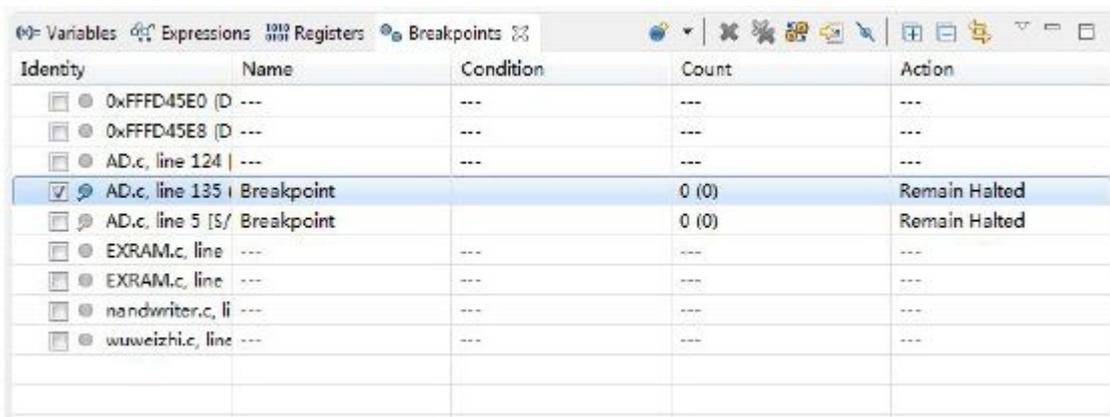


图 6-5-2

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

CCSv6 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個。

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:

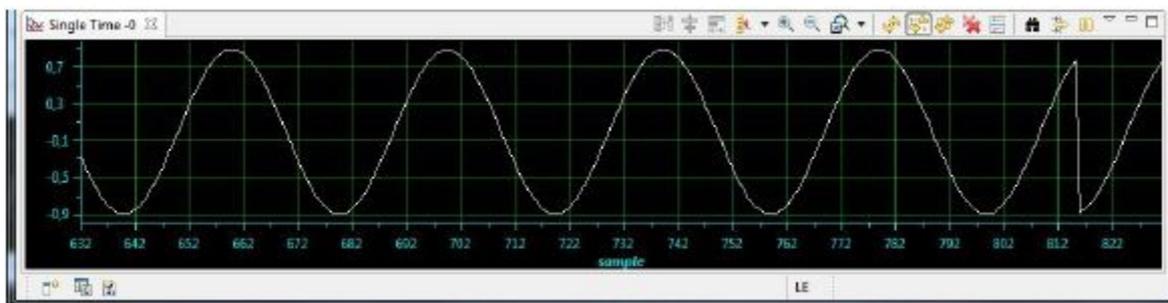


图 6-6-3

6.7 图像显示工具

要显示图像，只需转到菜单“Tools -> Image（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：

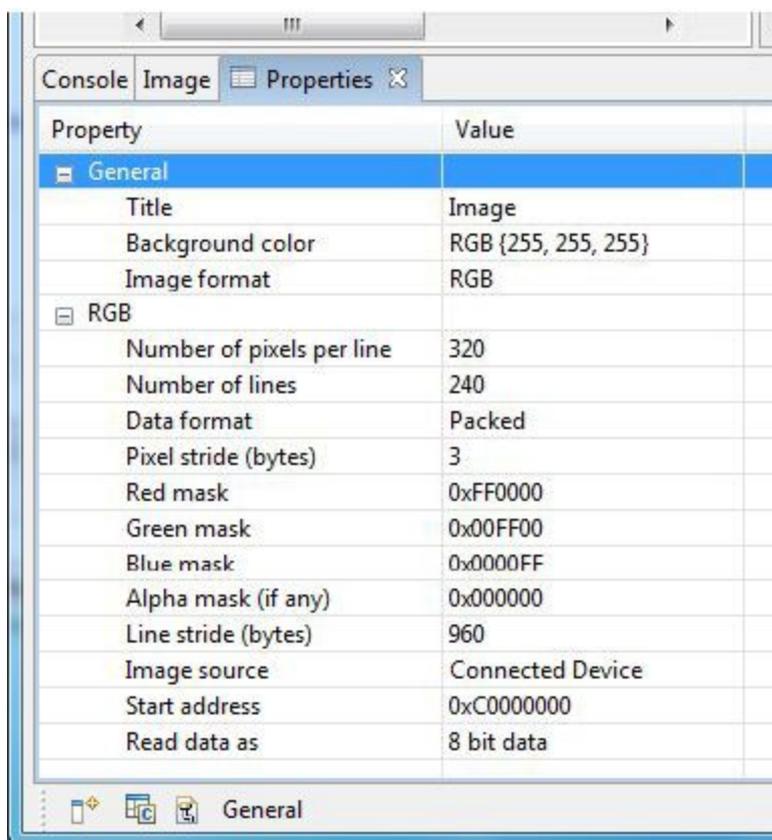


图 6-7-1

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。

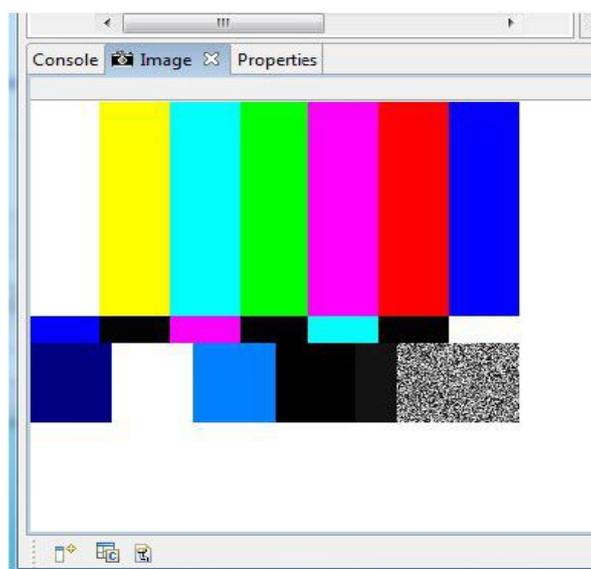


图 6-7-2

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。