



CCS6.0及XDS100v2仿真器 用户手册



声 明

南京研旭电气科技有限公司保留随时对其产品进行修改、改进和完善的权利，同时也保留在不作任何通告的情况下，终止其任何一款产品的供应和服务的权利。用户在下订单前应获取相关信息的最新版本，并验证这些信息是当前的和完整的。

南京研旭电气科技有限公司

版权©2014，南京研旭电气科技有限公司

第一章 CCSv6 简介

Code Composer Studio[™] (CCS 或 CCStudio) 是一种针对 TI 的 DSP、微控制器和应用处理器的集成开发环境。CCStudio 包括一套用于开发和调试嵌入式应用程序的工具。它包括用于各种 TI 设备系列的编译器、源代码编辑器、项目生成环境、调试程序、探查器、模拟器和其他许多功能。CCStudio 提供一个单一用户界面, 指导用户完成应用程序开发流程的每一步骤。类似的工具和界面使用户能够比以前更快地开始使用, 并且能够向他们的应用程序添加功能, 这些都归功于成熟的生产能力工具。

CCStudio 版本 6 (CCSv6) 以 Eclipse 开源软件框架为基础。CCSv6 之所以以 Eclipse 为基础, 是因为 Eclipse 为开发环境提供了一个优异的软件框架, 是众多嵌入式软件供应商所使用的标准框架。CCSv6 将 Eclipse 软件框架的优势和来自 TI 的高级嵌入式调试功能相结合, 为嵌入式程序开发人员生成一个功能丰富的吸引人的开发环境。

1.1 调试程序

CCStudio 的集成调试程序具有用于简化开发的众多功能和高级断点。条件断点或硬件断点以全 C 表达式、本地变量或寄存器为基础。高级内存窗口允许您检查内存的每一级别, 以便您可以调试复杂的缓存一致性问题。CCStudio 支持复杂的多处理器或多核系统的开发。全局断点和同步操作提供了对多个处理器和多核的控制。

1.2 分析

CCStudio 的交互式探查器使快速测量代码性能并确保在调试和开发过程中目标资源的高效使用变得更容易。探查器使开发人员能够轻松分析其应用程序中指令周期内或其他事件内的所有 C/C++ 函数, 例如缓存未命中/命中率、管道隔栏和分支。分析范围可用于在优化期间将精力集中在代码的高使用率方面, 帮助开发人员开发出经过优化的代码。分析可用于任何

组合的汇编、C++或C代码范围。为了提高生产能力，所有分析设备在整个开发周期中都可供使用。

1.3 脚本

某些任务，例如测试，需要运行数小时或数天而不需要用户交互。要完成此类任务，IDE应能自动执行一些常见任务。CCStudio 拥有完整的脚本环境，允许自动进行重复性任务，例如测试和性能基准测试。一个单独的脚本控制台允许您在 IDE 内键入命令或执行脚本。

1.4 图像分析和虚拟化

CCStudio 拥有许多图像分析及图形虚拟化功能。其中包括以图形方式在能够自动刷新的屏幕上查看变量和数据的能力。CCStudio 还能以本机格式（YUV、RGB）查看主机 PC 或在目标电路板中加载的图像和视频数据。

1.5 编译器

TI 已经开发了专门为了最大程度地提高处理器的使用率和性能而优化的 C/C++ 编译器。TI 编译器使用各种各样经典的、面向应用的、成熟的、因设备而异的优化，专为所有支持的结构而优化。其中部分优化包括：

- 消除公共子表达式
- 软件流水
- 强度折减

- 自动增量寻址
- 基于成本的寄存器分配
- 指令预测
- 硬件循环
- 函数内联
- 矢量化

TI 编译器还执行程序级别优化，在应用程序级别评估代码性能。通过程序级别视图，编译器能够像具有完整系统视图的汇编程序开发人员一样生成代码。编译器充分利用此应用程序级别视图，找出能够显著提升处理器性能的折衷。

TI ARM 和 Microcontroller C/C++编译器经过专门针对代码大小和控制代码效率的优化。它们具备行业领先的性能和兼容性。

1.6 模拟

模拟器向用户提供一种在能够使用开发板之前开始开发的方式。模拟器还具有更加透彻地了解应用程序性能和行为的优势。提供了几种模拟器，让用户能够权衡周期精确性、速度和外围设备模拟，一些模拟器特别适合算法基准测试，而另一些特别适合更加详细的系统模拟。

1.7 硬件调试（仿真）

TI 设备包含高级硬件调试功能。这些功能包括：

- IEEE 1149.1 (JTAG) 和边界扫描

- 对寄存器和内存的非侵入式访问
- 实时模式，用于调试与不得禁用的中断进行交互的代码。实时模式允许您在中断事件挂起后台代码，同时继续执行时间关键中断服务例程。
- 多核操作，例如同步运行、步进和终止。其中包括跨核触发，该功能可以让一个核触发另一个核终止。

高级事件触发 (AET)，可在选定设备上使用，允许用户依据复杂事件或序列，例如无效数据或程序内存访问，终止 CPU 或触发其他事件。它能够以非侵入式方式测量性能及统计系统事件数量（例如缓存事件）。

CCStudio 提供有关选定设备的处理器跟踪，帮助客户发现以前“看不到的”复杂实时缺陷。跟踪能够探测很难发现的缺陷-事件之间的争用情况、间歇式实时干扰、堆栈溢出崩溃、失控代码和不停用处理器的误中断。跟踪是一种完全非侵入式调试方法，依赖处理器内的调试单元，因此不会干扰或更改应用程序的实时行为。跟踪可以微调复杂开关密集型多通道应用程序的代码性能和缓存优化。处理器跟踪支持程序、数据、计时和所选处理器与系统事件/中断的导出。可以将处理器跟踪导出到 XDS560 跟踪外部 JTAG 仿真器或选定设备上，或导出到芯片缓存嵌入式跟踪缓存 (ETB) 上。

1.8 实时操作系统支持

CCSv6 具有两个版本的 TI 实时操作系统：

- DSP/BIOS5.4x 是一种为 DSP 设备提供预清空多任务服务的实时操作系统。其服务包括 ISR 调度、软件中断、信号灯、消息、设备 I/O、内存管理和电源管理。此外，DSP/BIOS5.x 还包括调试诊断和加工，包括低系统开销打印和统计数据收集。
- BIOS6.x 是一种高级可扩展实时操作系统，支持 ARM926、ARM Cortex M3、C674x、C64x+、C672x 和基于 28x 的设备。它提供 DSP/BIOS 5.x 没有的若干内核和调试增强，包括更快、更灵活的内存管理、事件和优先级继承互斥体。

注意: BIOS6.x 包括 DSP/BIOS5.x 兼容层, 从而使应用程序源代码的迁移非常轻松。

1.9 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。以后随时可以在 CCSv6 中更改工作区, 如下图 1-9-1 所示:

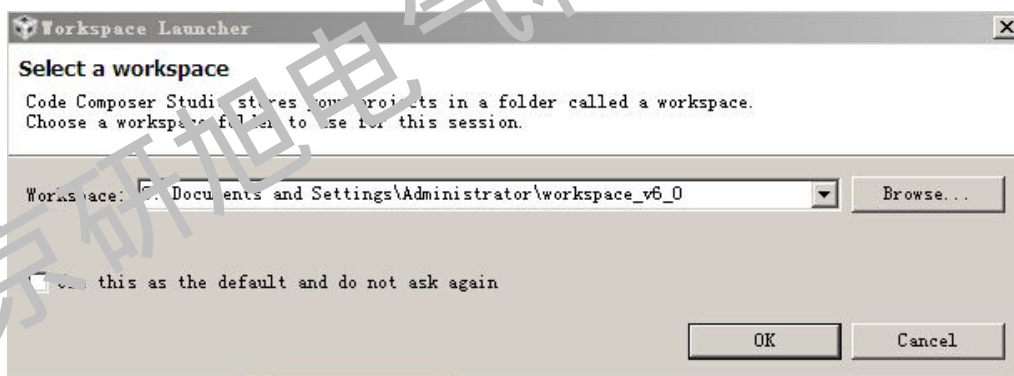


图 1-9-1

Code Composer Studio 上的所有工作都基于项目——一个源文件、include 文件和配置文件的集合。

1.10 创建项目

1.10.1 简介

本部分说明了在 CCSv6 中创建项目的一般步骤。标记为教程:的句子会创建示例程序,帮助您了解 CCSv6 的功能。

1.10.2 创建项目

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。

- ① 转到菜单“File -> New -> CCS Project (文件->新建->CCS 项目)”如下图 1-10-1 所示:

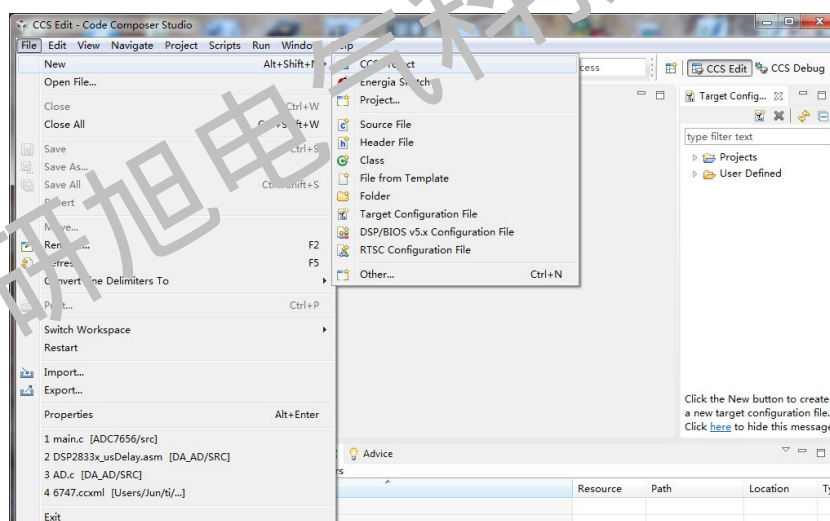


图 1-10-1

- 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)。

教程: 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。所创建的项目将显示在“C/C++ Projects（C/C++ 项目）”选项卡中，可随时用于创建或添加源文件，如下图 1-10-2 所示：

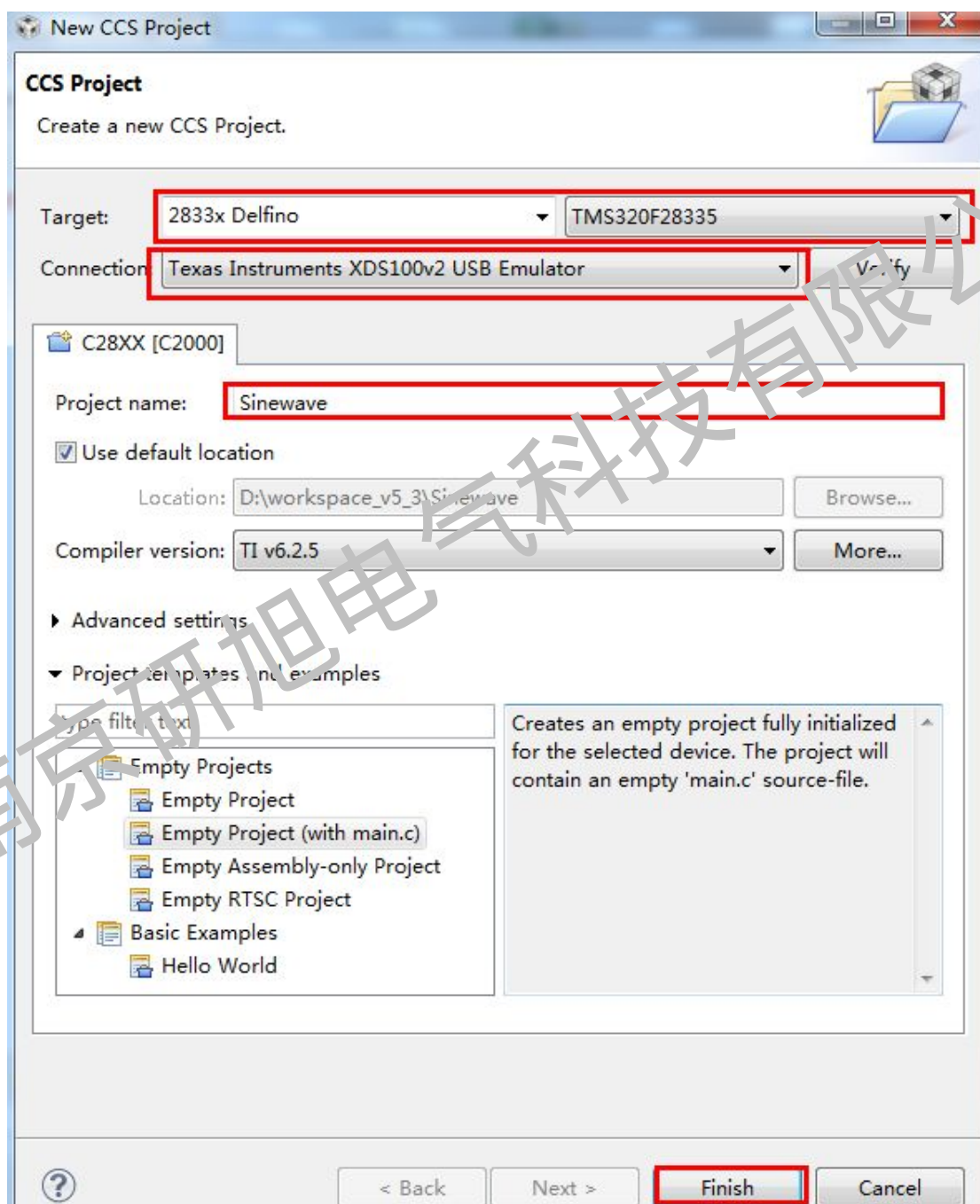


图 1-10-2

⑥ 要为项目创建文件,请在“C/C++ Projects (C/C++ 项目)”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。

1.10.3 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。

1.10.4 配置生成设置

要配置生成设置,请在“C/C++ Projects (C/C++ 项目)”视图中右键单击项目,并选择“Build Properties... (生成属性...)”。有多个适用于编译器、汇编器和链接器的选项。

1.11 项目调试

本部分说明了在 CCSv6 中创建目标配置和调试项目所需的一般步骤。标记为**教程:**的句子说明了对上一部分中创建的示例程序进行调试的具体步骤。

1.11.1 启动调试器之前

在启动调试器之前, 需要选择并配置代码将要执行的目标位置。目标可以是软件模拟器或与开发板相连的仿真器。

- 软件模拟器不需要外部硬件, 对于执行基准和算法验证十分有用。有关模拟技术的其他信息, 请参阅以下链接:

<http://processors.wiki.ti.com/index.php/Category:Simulation>

- 仿真器是用于直接对硬件进行调试的硬件设备, 可以内置到开发板 (DSK、eZdsp、EVM 等), 也可以采取独立形式 (XD100v2、XD560 等)。有关仿真技术的其他信息, 请参阅以下链接:

<http://processors.wiki.ti.com/index.php/Category:Emulation>

注意: 如果您熟悉 CCSv3.3, 应该知道目标配置是通过外部程序 CCSetup 来完成的。而在 CCSv6 中, 这种配置在 IDE 内部完成, 它不仅创建整个系统范围的配置, 还可以创建各个项目的单独配置。这样做还有一个好处, 就是在每个目标配置更改后无需重新启动 CCS。

教程: 本例中将使用模拟器。

以下介绍如何创建目标配置文件:

CCSv6 提供了一个十分简单易用的图形目标配置编辑器, 它提供多个预配置的设备 and 开发板, 而且还可以在自定义硬件中使用。

每个项目可以拥有一个或多个目标配置, 但只能有一个处于活动状态。

可选: CCSv6 还允许创建一个系统范围的目标配置, 以便可以在各个项目之间进行共享。

① 右键单击项目名称，并选择“New -> Target Configuration File（新建 ->目标配置文件）”如下图所示：

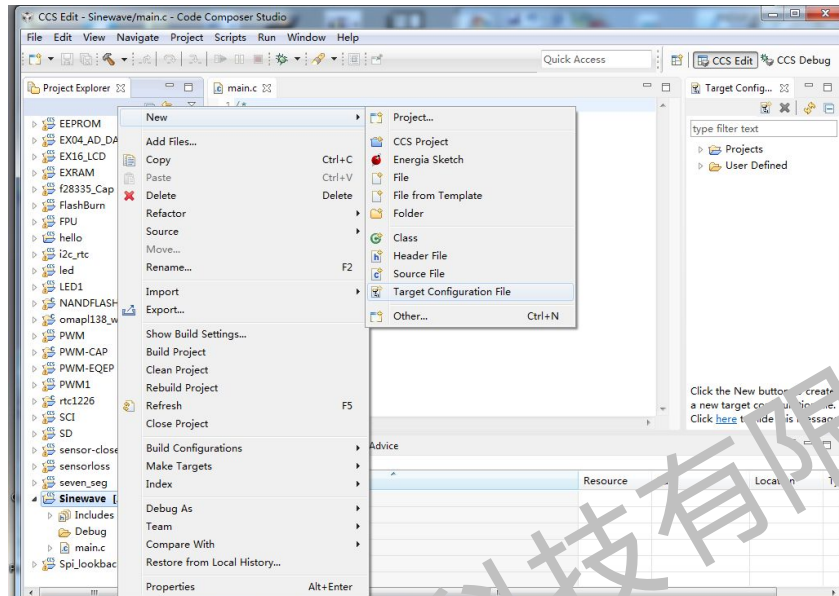


图 1-11-1

② 为配置文件命名-将会添加扩展名.ccxml。建议根据所使用的目标和仿真器指定一个有意义的名称，如下图所示。

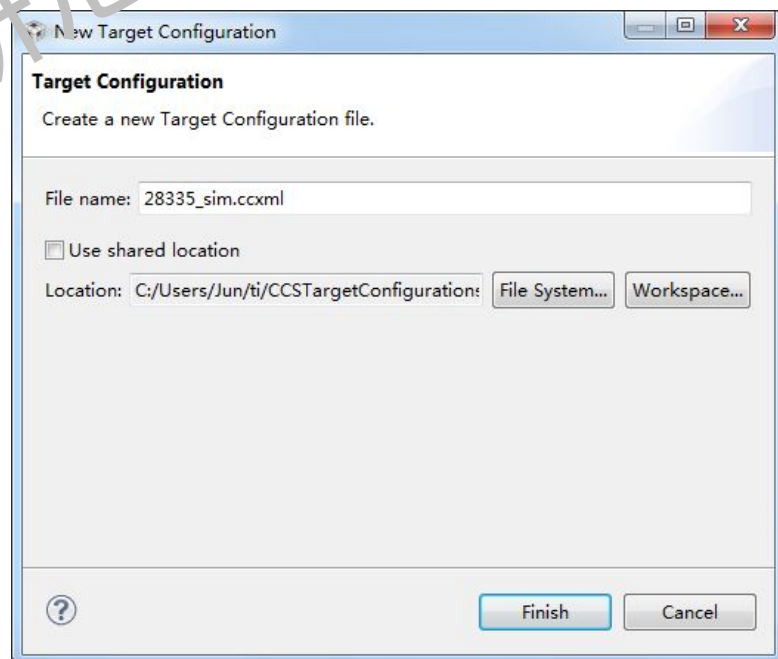


图 1-11-2

可选。如果选中“Use shared location（使用共享位置）”选项，新的目标配置将在所有项目之间共享，并存储在默认的 CCSv6 目录下。

③ 单击“Finish（完成）”。此时将打开目标配置编辑器，如下图 1-11-3 所示：

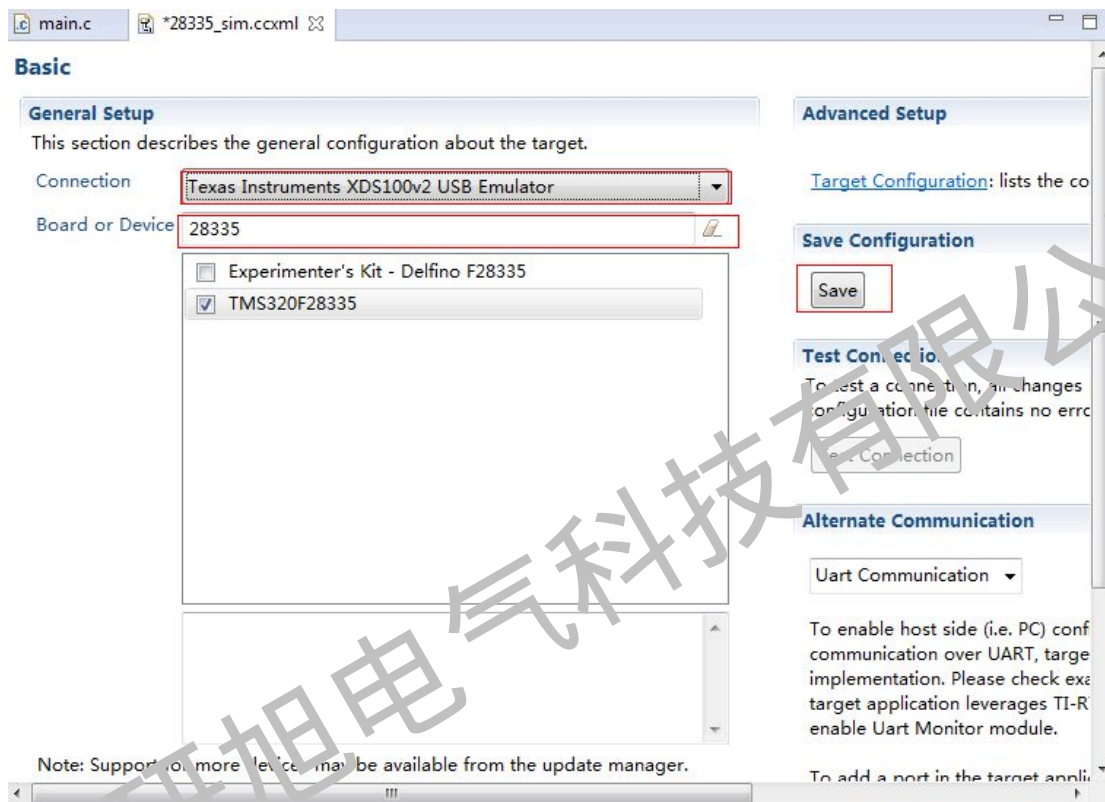


图 1-11-3

④ 目标中有两项元素是必须配置的：

- 通过“Connection（连接）”下拉菜单可以选择是使用一个软件模拟器，还是使用多种内置或独立仿真器。有关如何配置常见目标的说明，请查 [GSG：常见目标配置](#) 部分。
- “Board or Device（设备）”部分包含与所选连接兼容的所有设备。上部的框是筛选器，可以帮助在下部框中的浏览表中选择正确设备。

⑤ 选择设备后，单击“Save（保存）”按钮。该配置将自动设置为“Active（活动）”。

每个项目可以拥有多个目标配置，但只能有一个处于活动状态，该配置将会自动启动。

注意：要查看系统现有的所有目标配置，只需转到菜单“View -> Target Configurations（查看 -> 目标配置）”。

1.11.2 启动调试器

创建配置之后, 可通过转到菜单“Target -> Debug Active Project (目标 -> 调试活动项目)”启动调试器。将会打开“Debug Perspective (调试透视)”, 专为调试定制的一组专用窗口和菜单。

注意: 如果对源代码或生成选项进行了修改, 启动调试器可能会导致 CCSv6 生成活动项目。

(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)”, 如下图 1-11-1 和图 1-11-2 所示:

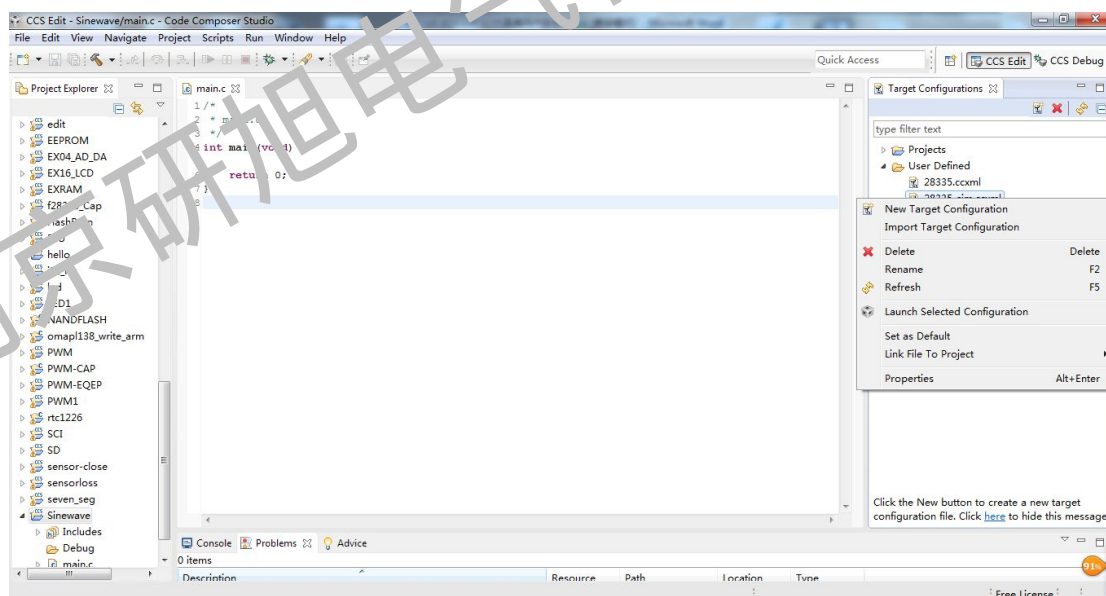


图 1-11-1

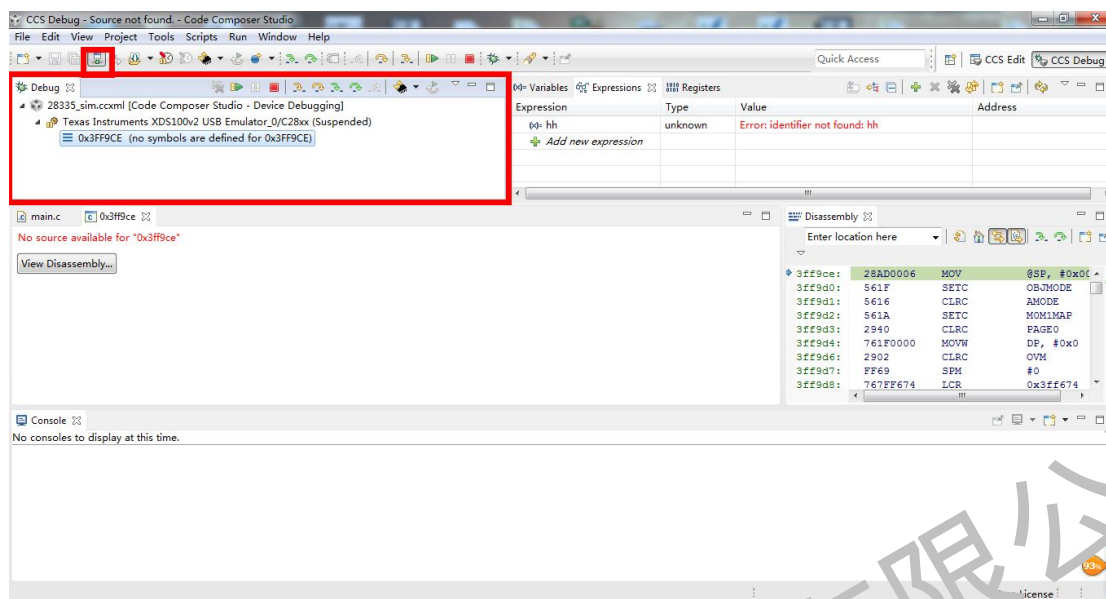


图 1-11-2

- “Debug（调试）”视图包含每个芯片核的目标配置和调用堆栈。
 - 源代码视图显示了在 `main()` 处停止的程序。
 - 基本调试功能（运行、停止、步入/步出、复位）位于“Debug（调试）”视图的顶部栏中。
- “Target（目标）”菜单还有其他几种调试功能。

教程: 通过转到菜单“Target -> Debug Active Project（目标 -> 调试活动项目）”启动调试器。

注意: 如果目标配置需要先运行脚本再加载代码，将打开“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。

(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 1-11-3 和图 1-11-4 所示：

Expression	Type	Value	Address
(*)= y1	float	5.9	0x0000C00A@Data
(*)= y2	float	13.28	0x0000C008@Data
+ Add new expression			

图 1-11-3

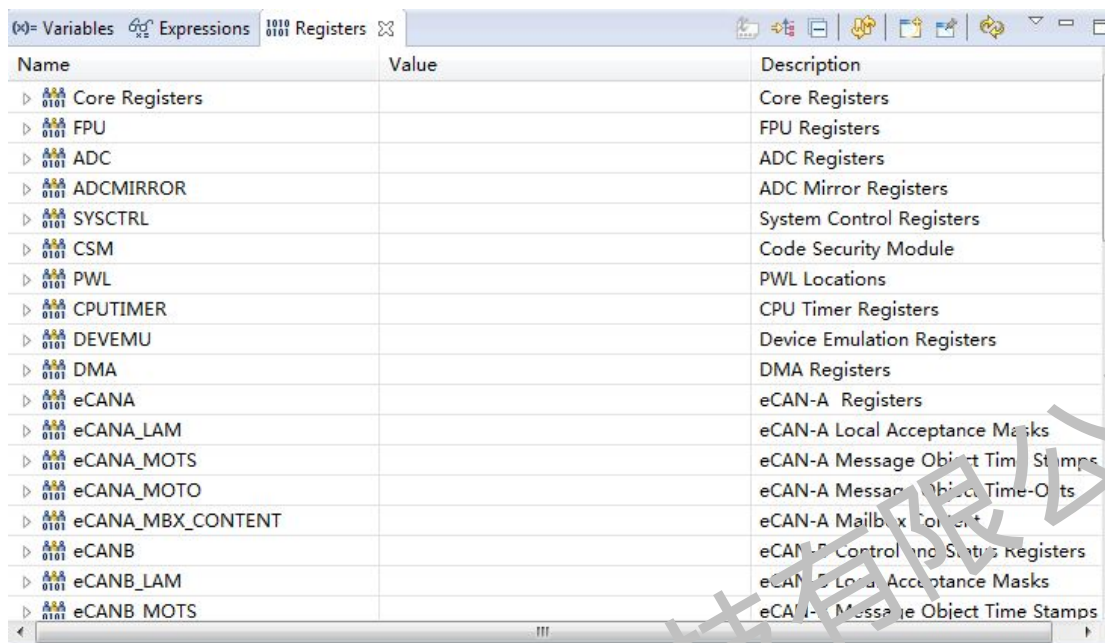


图 1-11-4

教程: 上面的屏幕截图显示了典型 TMS320C28335 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。

(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。

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 1-11-5 所示：

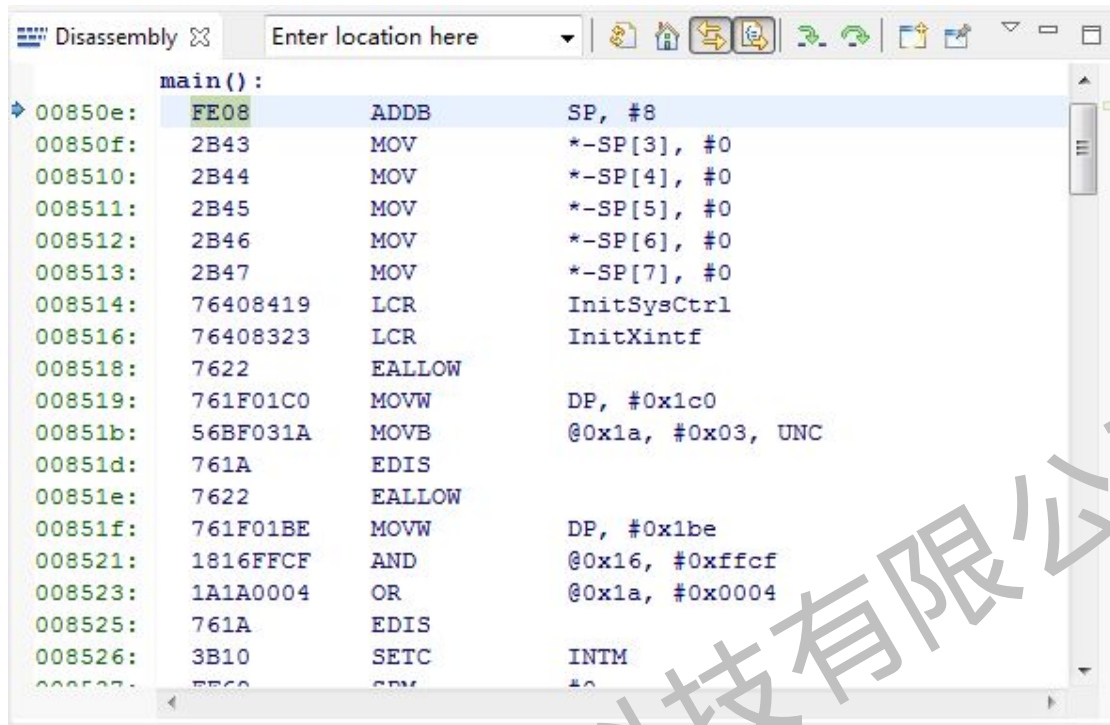


图 1-11-5

(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（查看 -> 内存）”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 1-11-6 所示：

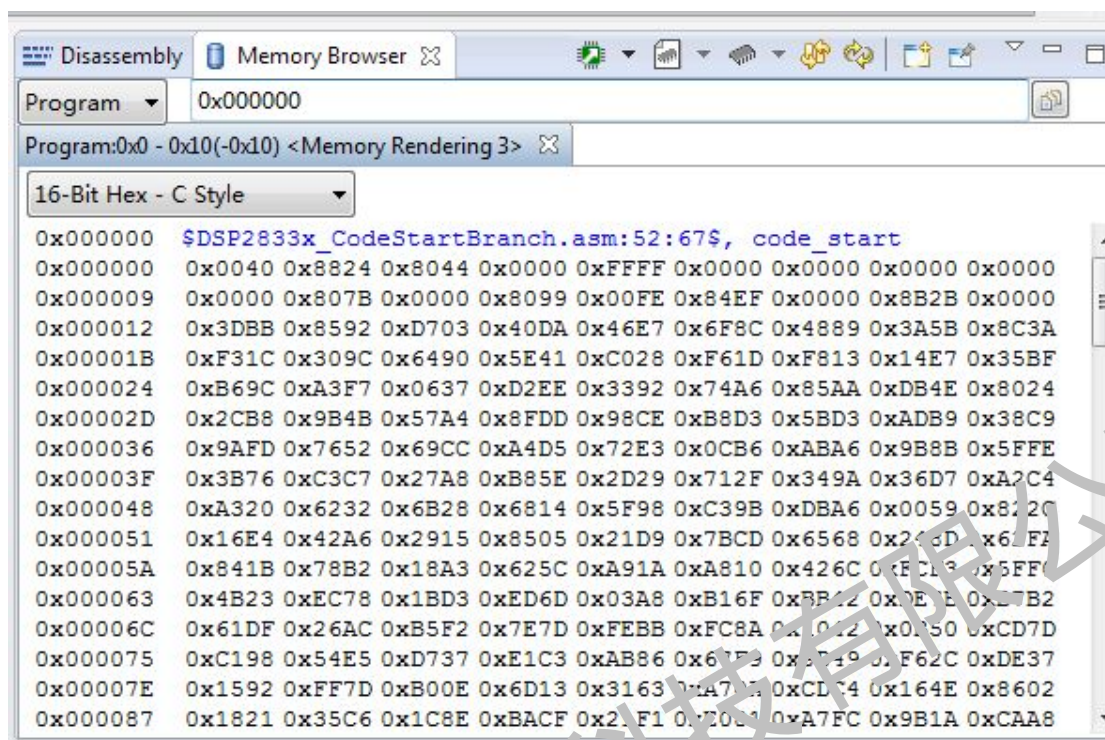


图 1-11-6

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。

(5) 管理断点

作为任何调试器都会拥有的最基本功能，CCSv5 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 1-11-7 所示：

- 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。

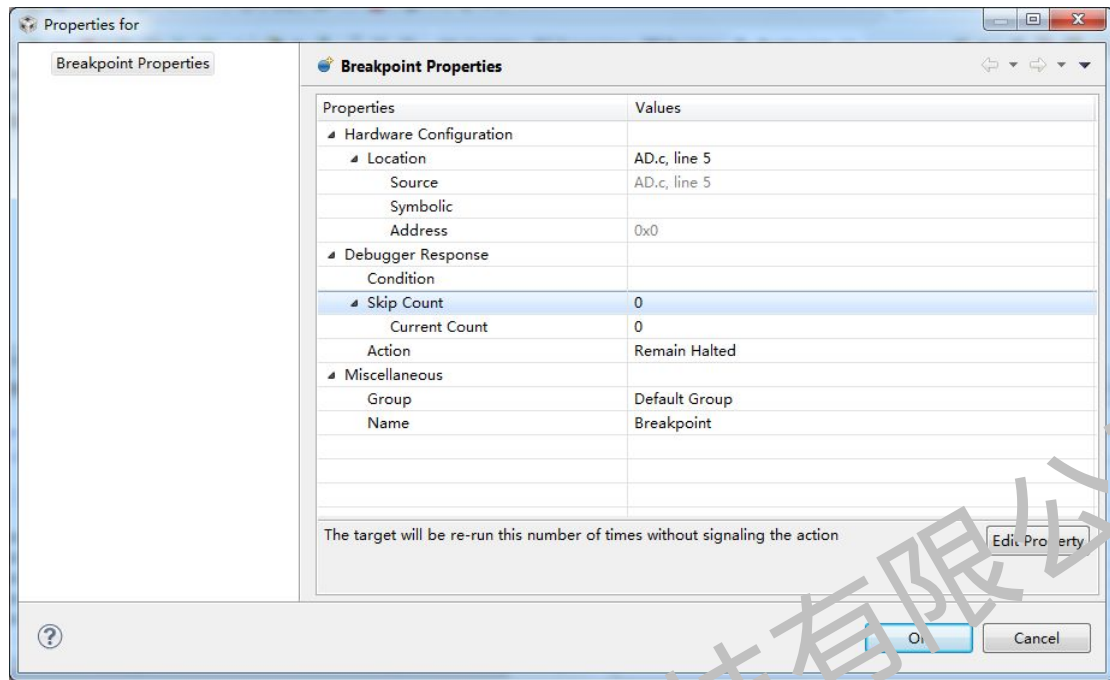


图 1-11-7

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 1-11-8 断点查看器中看到。

Identity	Name	Condition	Count	Action
<input type="checkbox"/> 0xFFFFD45E0 (D ---		---	---	---
<input type="checkbox"/> 0xFFFFD45E8 (D ---		---	---	---
<input type="checkbox"/> AD.c, line 124 ---		---	---	---
<input checked="" type="checkbox"/> AD.c, line 135 Breakpoint			0 (0)	Remain Halted
<input type="checkbox"/> AD.c, line 5 [S/ Breakpoint			0 (0)	Remain Halted
<input type="checkbox"/> EXRAM.c, line ---		---	---	---
<input type="checkbox"/> EXRAM.c, line ---		---	---	---
<input type="checkbox"/> nandwriter.c, li ---		---	---	---
<input type="checkbox"/> wuweizhi.c, line ---		---	---	---

图 1-11-8

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) 图形显示工具

CCSv4 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“Real 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 1-11-9 所示：



图 1-11-9

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。

- 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...）”。
- 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。

- 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 1-11-10 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 1-11-10

- 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。
- 单击 “Target -> Run (目标 -> 运行)”, 该图形应该以 16 个样本为一组分批更新。
- 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 1-11-11 所示:

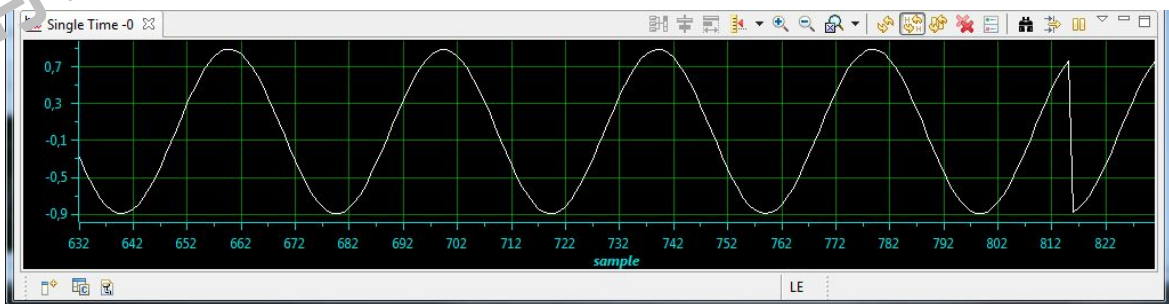


图 1-11-11

(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：

- 转到菜单“View -> Memory (查看 ->内存)”打开内存视图。
- 在地址框中键入有效的目标地址：0xC0000000
- 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\ccsv5\c6000\examples
- 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。
- 按下图 1-11-12 所示设置属性：

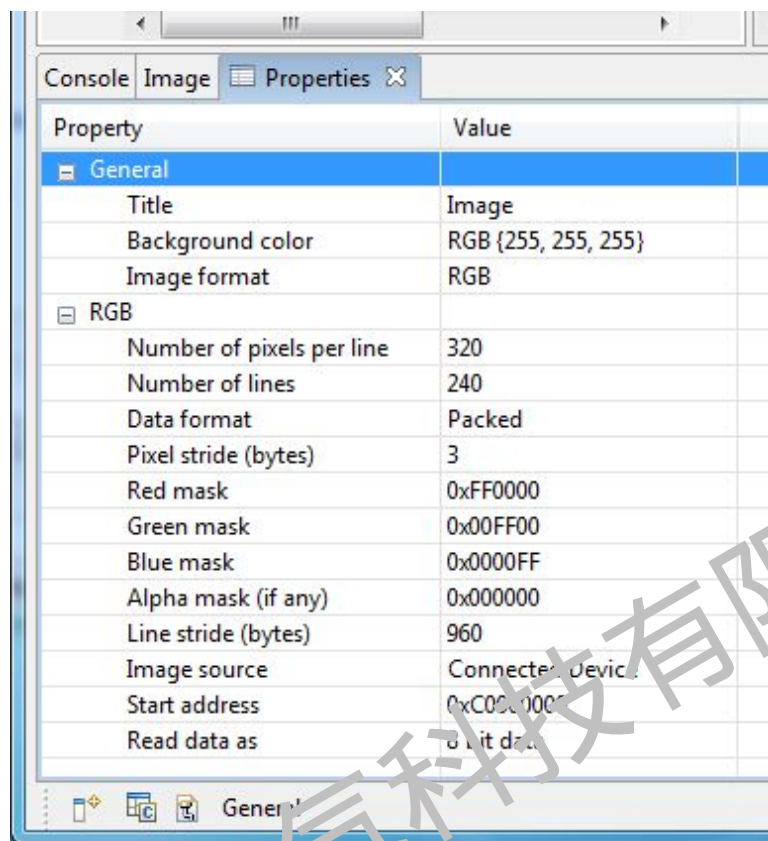


图 1-11-12

- 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 1-11-13 的图像。

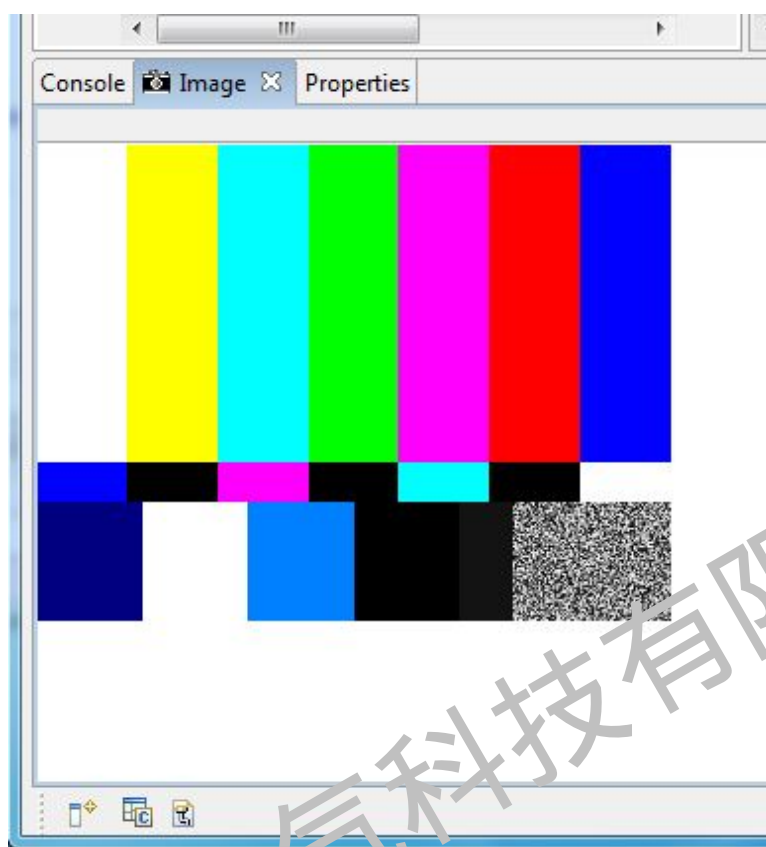


图 1-11-13

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。

第二章 CCSv6 安装

CCS 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)。如果您曾使用过其他公司的仿真器产品。

我们推荐使用 CCS6.0 版本, 因为暂时合纵达公司的 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 版本以上的开发环境。

双击 ccs_setup_6.0.0.00190.exe 文件, 出现如下图 2-1 界面



图2-1

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

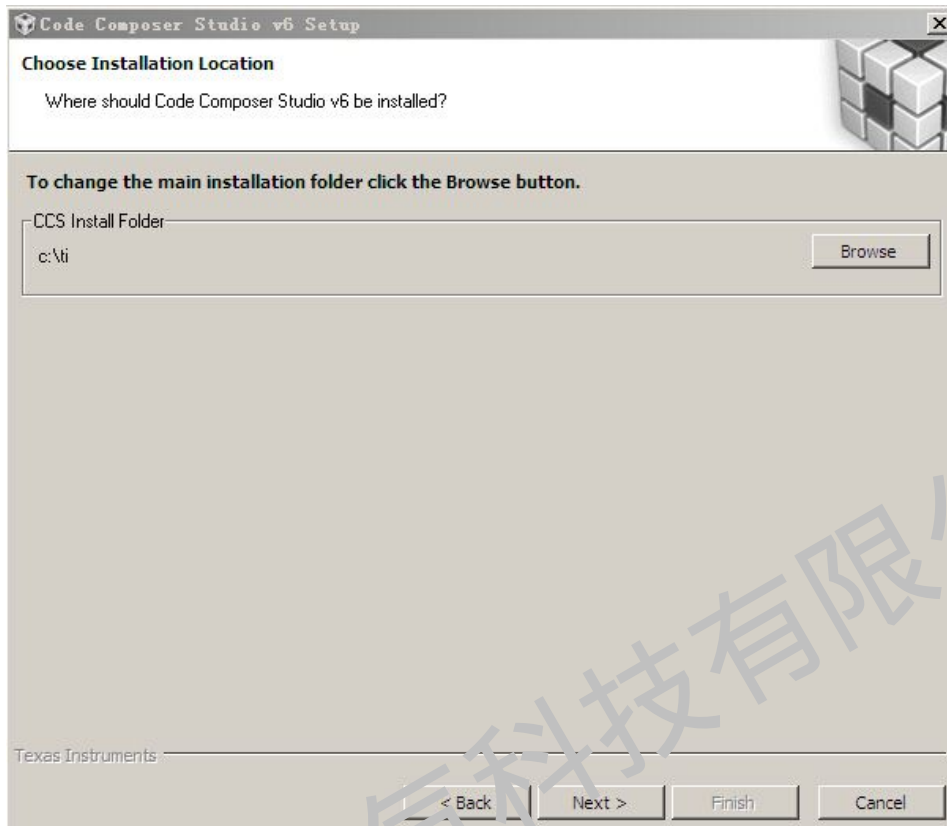


图 2-2

点击“Browse”选择安装路径，但推荐默认路径，点击“Next”见下图 2-3 所示：

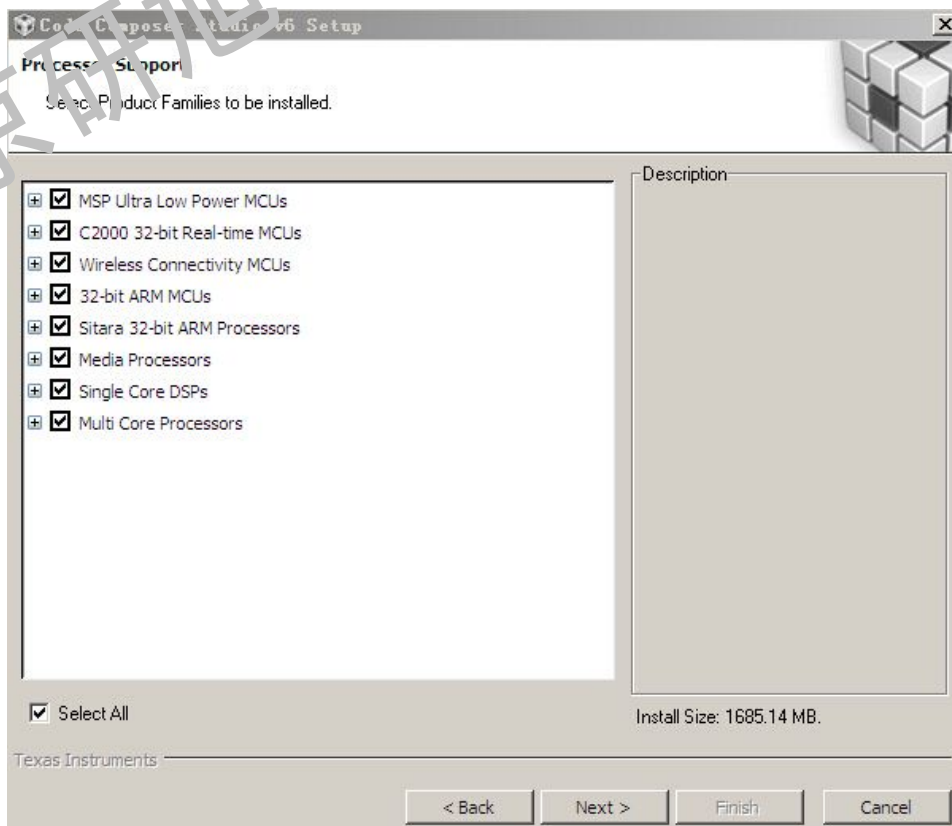


图 2-3

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 2-4 所示：

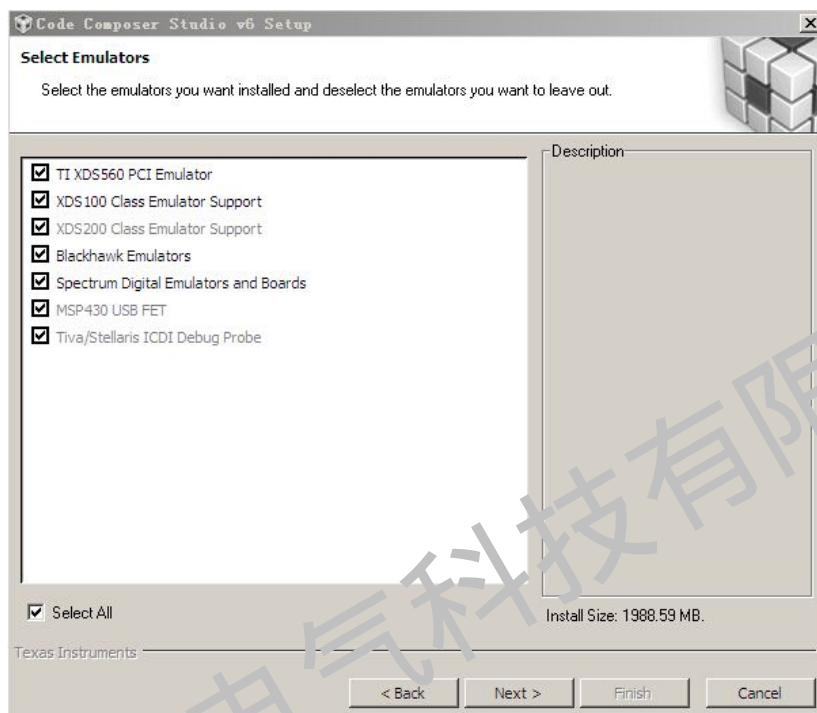


图 2-4

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 2-5 所示：



图2-5

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2-6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;

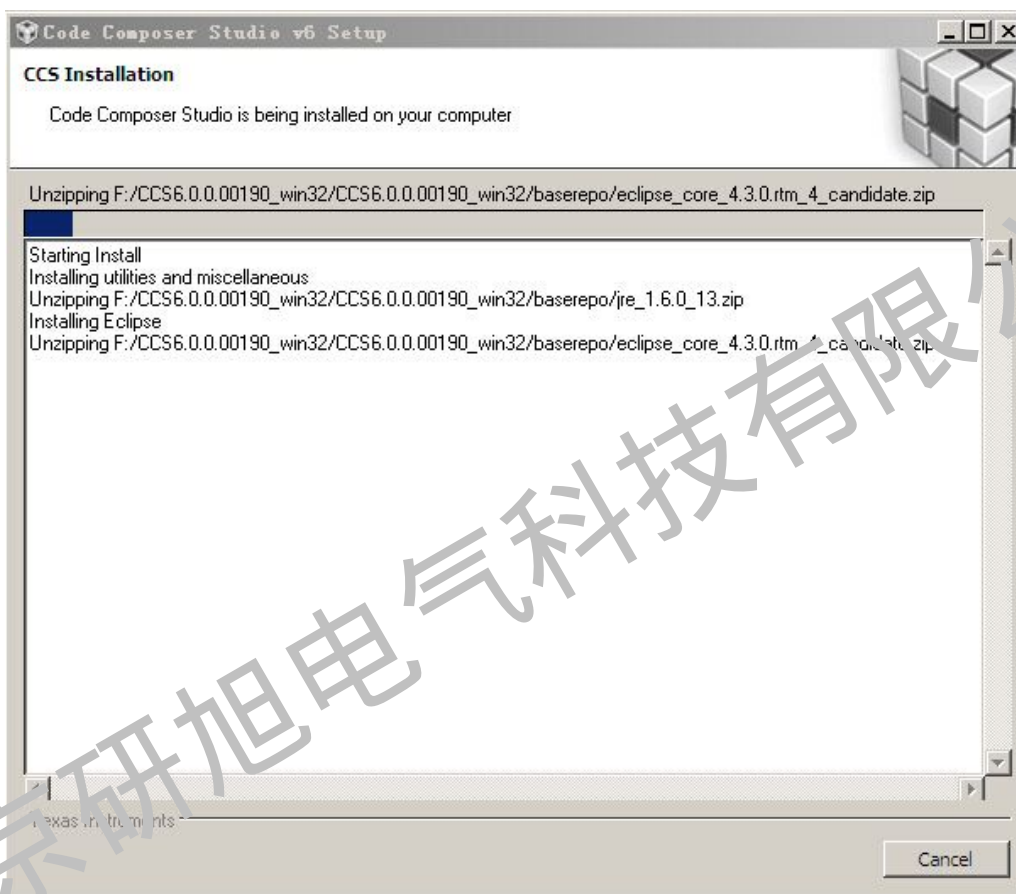


图 2-6

安装完成, 点击“Finish”见下图 2-7 所示:

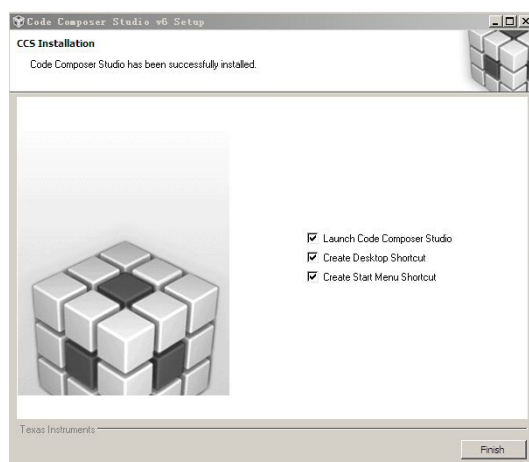


图 2-7

第三章 仿真器驱动与识别

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。

3.1 仿真器的使用

3.1.1 建立仿真目标板配置环境

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 CCSETUP 进行建立, 在此不多做介绍, 下面主要介绍如何在 CCSv6 中建立:

- 在菜单 Target 一栏中, 选择 New Target configuration, 如下图 3-1-1 所示:

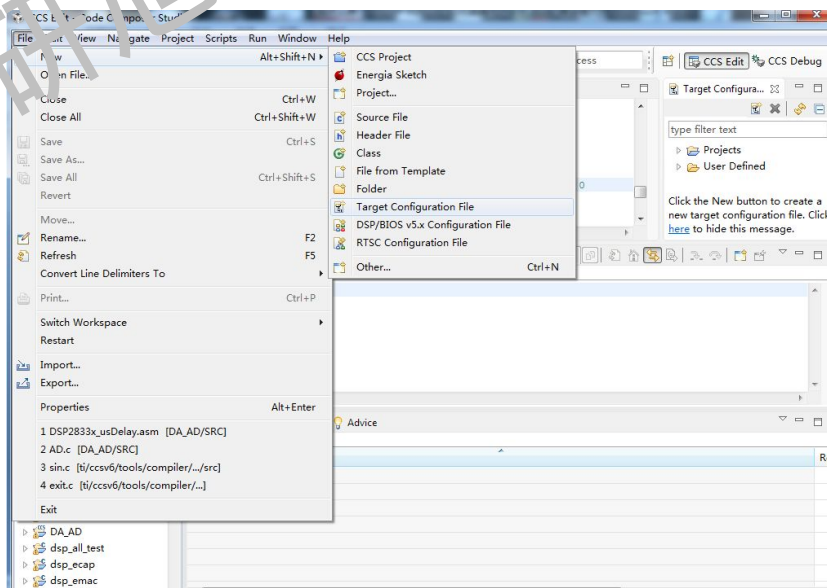


图 3-1-1

- 为此配置命名 f28335_xds100_v2 .ccxml (可以自行命名), 点击 Finish, 如下图 3-1-2 所示:

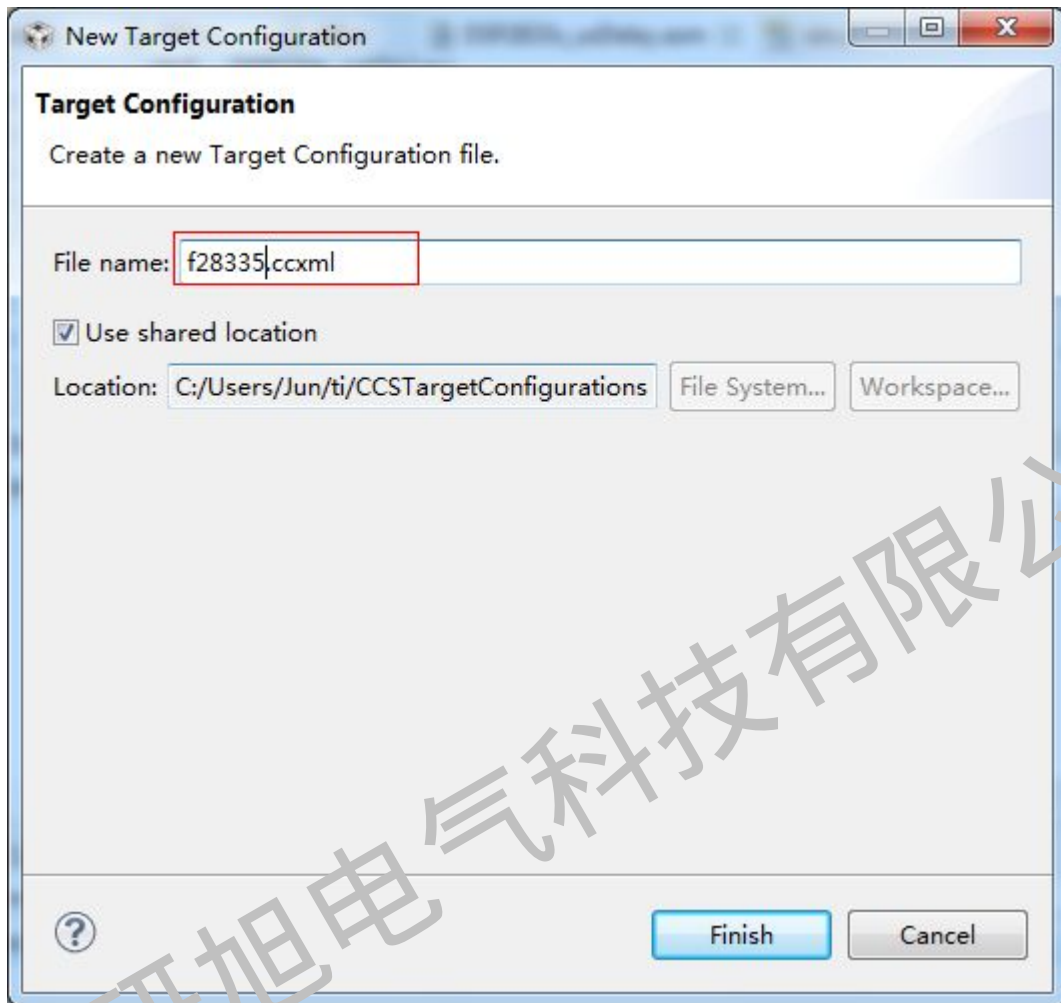


图 3-1-2

- 在 connection 一栏中选择 Texas Instruments XDS100 v2 USB Emulator, Device 一栏中选择 TMS320F28335, 如下图 3-1-3 所示:

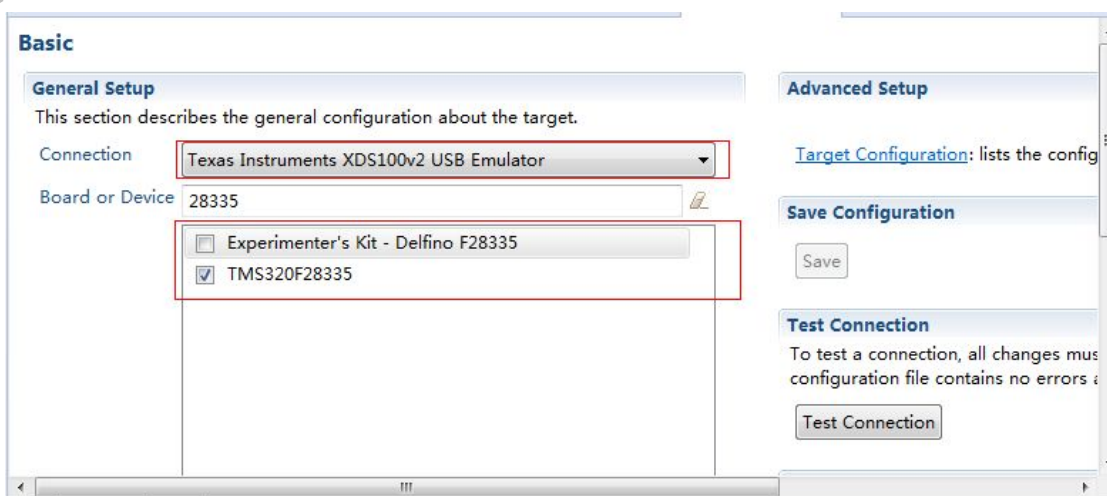


图 3-1-3

- 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 3-1-4 界面：

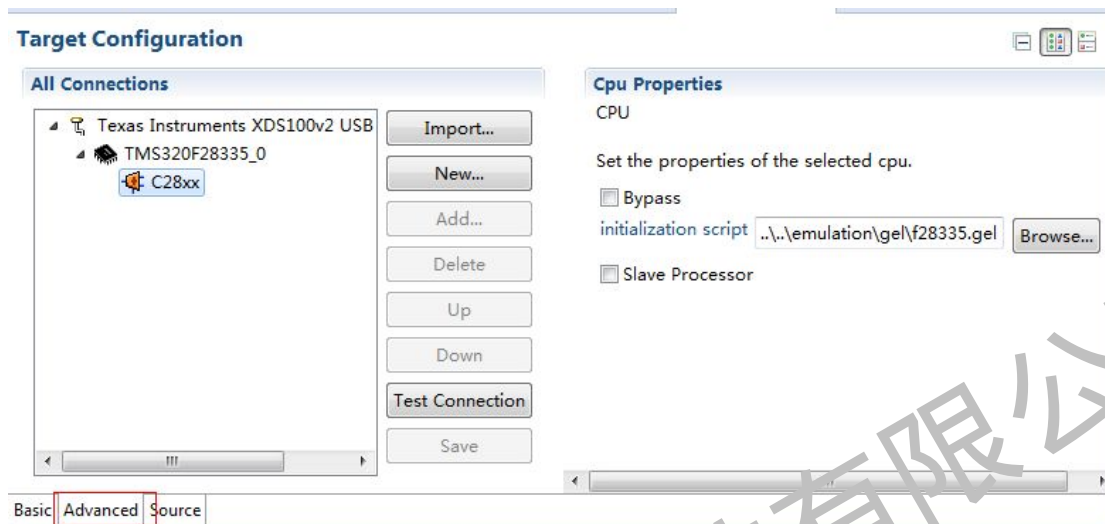


图 3-1-4

- 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。
- 在 Target configuration 中，将刚刚建立的 Texas Instruments XDS100 v2 USB Emulator 设置为默认状态；其操作步骤如下图 3-1-5 所示：

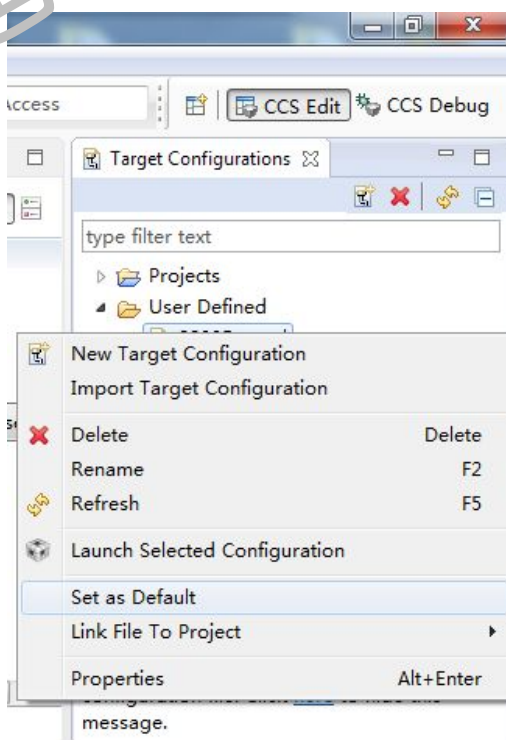


图 3-1-5

3.1.2 连接目标板

- 启动调试, 右击选择已配置的项目的 Launch Selected Configuration; 见下图 3-1-6 所示:

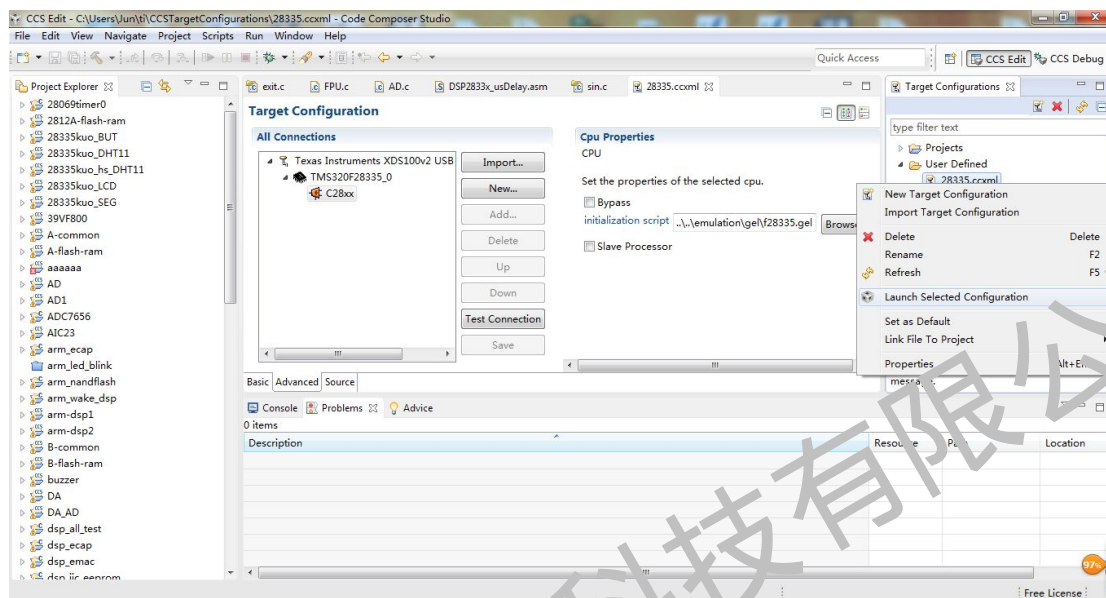


图 3-1-6

- 当启动成功后, “连接标志”就会有效, 点击即可连接目标板; 出现以下图 3-1-7 界面, 证明连接成功:

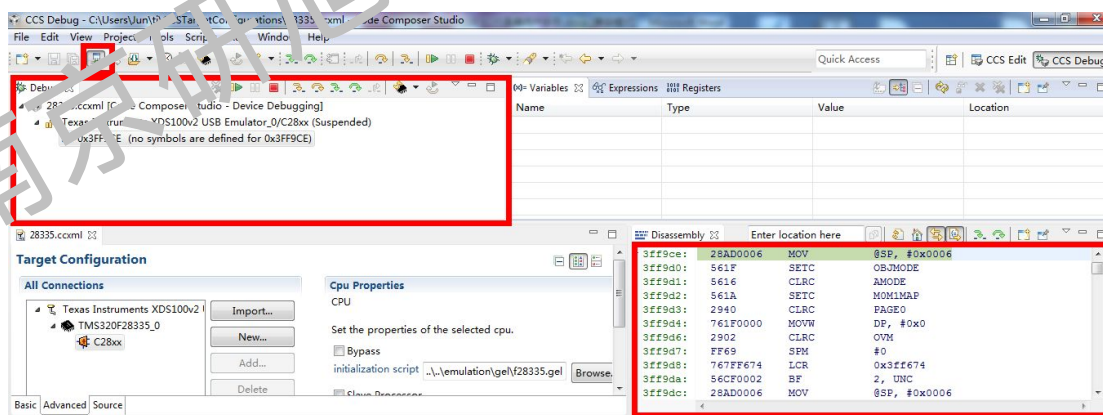


图 3-1-7

- 接下去, 用户可以自行仿真实验。