



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



# CCS开发环境使用以及技巧

— DSP培训系列之三

上海交大-TI 联合DSP实验室

版权所有

## 2.1 CCS简介

## 2.2 安装

CCS的安装

BIOS实时操作系统的安装

## 2.3 CCS的基本功能

创建新工程

构建程序

加载程序

基本调试

# TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP，MSP430等技术培训与技术服务，项目合作等。培训内容有

- 1) CCS开发环境精解与实例；
- 2) DSP/SYS BIOS 实例；
- 3) C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；
- 4) HPI，EMIF，EDMA，Timer等外设；
- 5) C6416、DM642，C6678多核EVM开发平台实例；
- 6) Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），

邮件报名：[jhniu@sjtu.edu.cn](mailto:jhniu@sjtu.edu.cn)，



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



## 颁发TI授权的培训证书



SHANGHAI JIAO TONG UNIVERSITY

the Office of the President



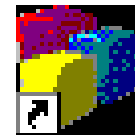
## 2.1 CCS简介

- ❶ CCS (Code Composer Studio) 是用于 TI DSP、微处理器和应用处理器的集成开发环境。
- ❷ CCS 包含一整套用于开发和调试嵌入式应用的工具。它包含适用于每个 TI 器件系列的编译器、源码编辑器、项目构建环境、调试器、描述器、仿真器以及多种其它功能。

## 2.2 安装

### 1. CCS的安装

- ④ 用户可以用软件提供的**标准配置文件产生一个系统配置**，也可以用自己的配置文件产生一个客户化的配置。
- ④ 1. 双击桌面上的Setup Code Composer Studio 图标，出现系统配置配置对话框。
- ④ 2. 从available factory board中选择与系统匹配的标准设置。



Setup CCStudio v3.3.1nk

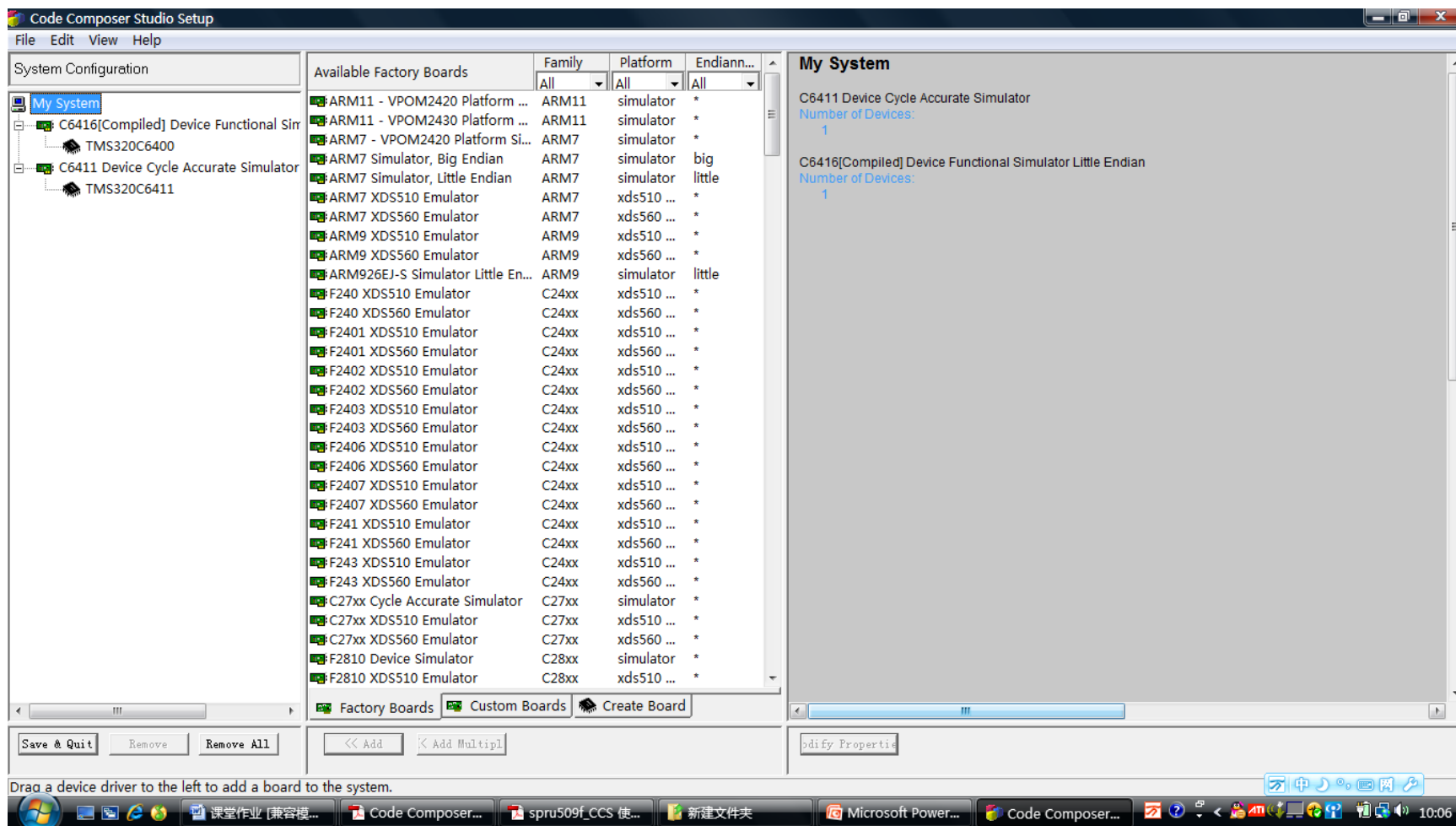




# CCS的安装



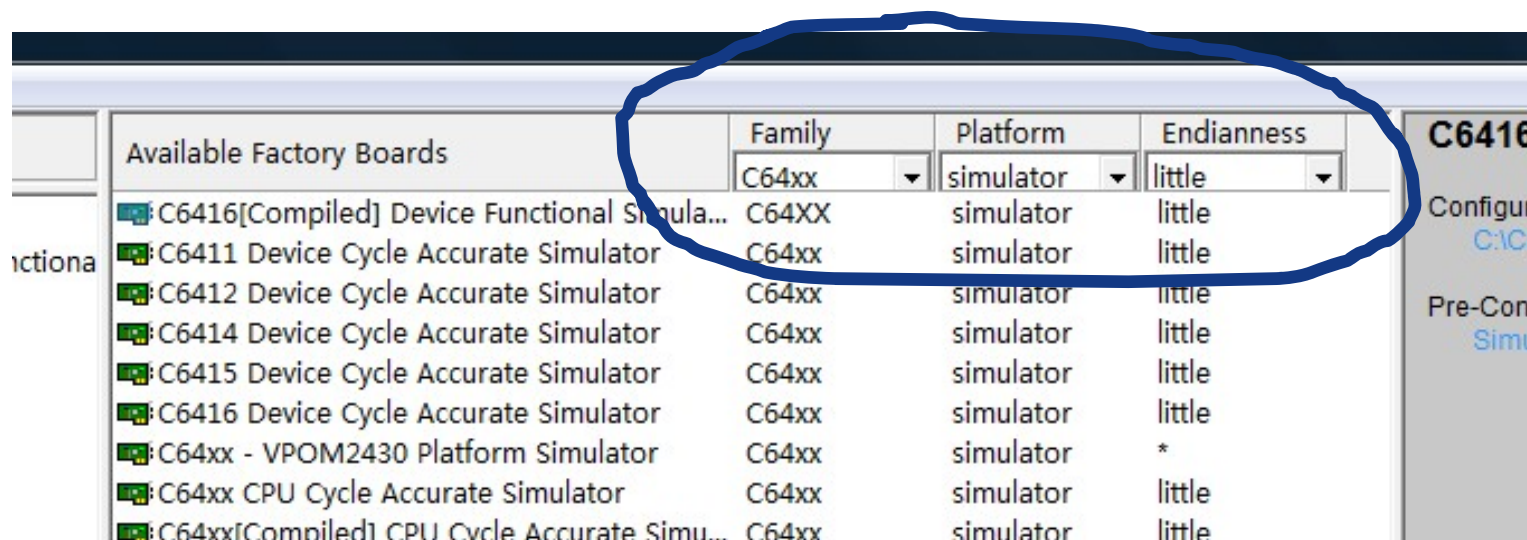
## CCSsetup的配置





# CCS的安装

## CCSsetup的配置



注意事项：请尽可能选择默认的安装路径，并严禁使用中文路径。

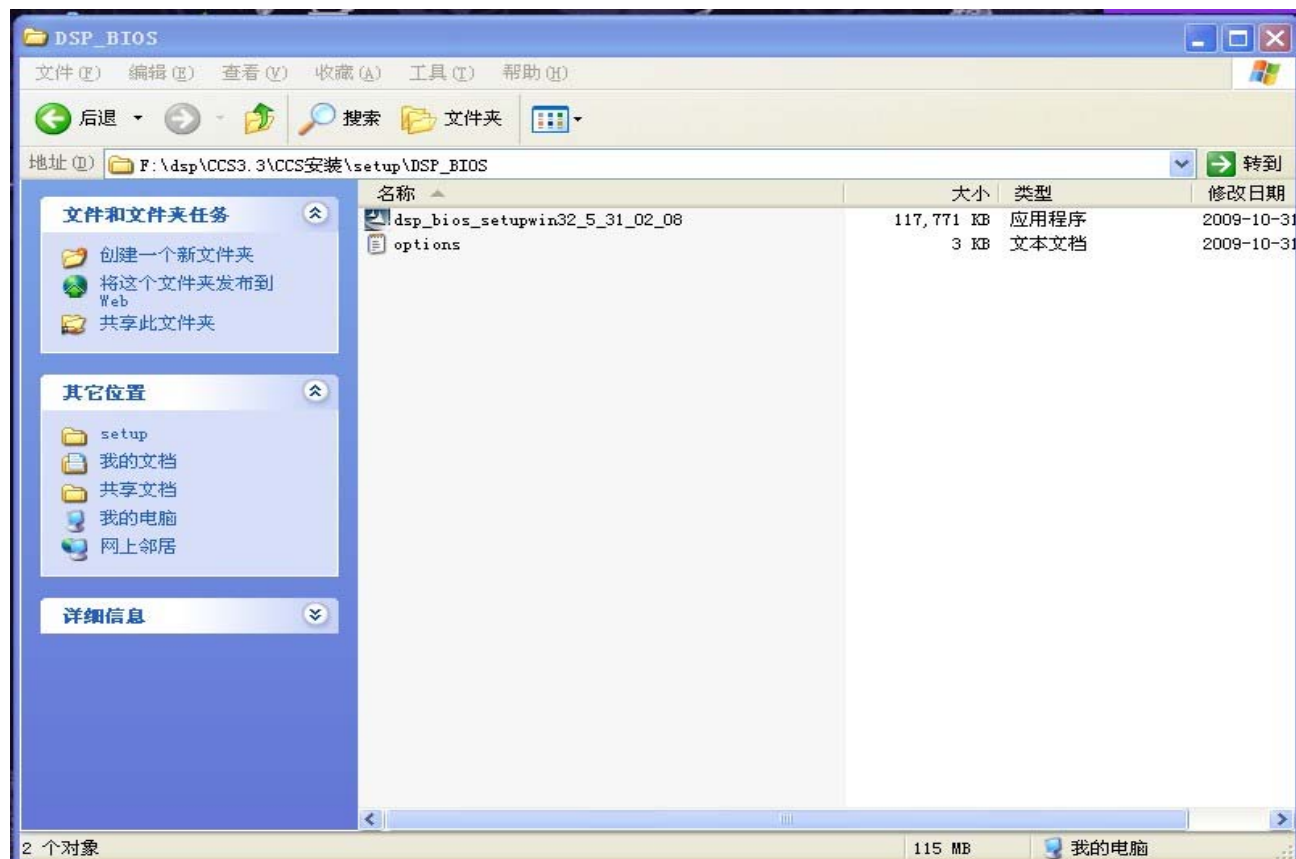


- ④ 3. 单击选择的配置，然后单击ADD按钮将选择的配置添加到system configuration中。  
(如果你的配置包含多于一个的目标板，重复以上步骤直到为每一个目标板选择了一个配置。)
- ④ 4. 单击 Save & Quit 按钮保存配置。
- ④ 5. 单击 Yes 按钮，启动配置好的CCS集成开发环境。现在可以打开一个工程。



# BIOS实时操作系统的安装

- CCS3.3 的bios 可以独立安装，安装文件名：  
dsp\_bios\_setupwin32\_5\_31\_02\_08.exe。





## 2.3 CCS的基本功能

- ④ 创建新工程 (Creating a New Project)
- ④ 构建程序 (Building Your Program)
- ④ 加载程序 (loading your program)
- ④ 基本调试 (basic debugging)



## 2.3 CCS的基本功能

CCS所使用的重要的图标:



Launches Code Composer Studio



重新构建工程 (Rebuilds the project)



增量构建 (Builds the project incrementally)



暂停执行 (Halts execution)



## 2.3 CCS的基本功能



断点 (Toggles breakpoint)



运行 (Runs project)



进入 (Single steps project)



跳出 (Step out)



跳过 (Step over)



## 2.3 CCS的基本功能

- 如果这些图标在工具栏中无法显示，请选择  
View→Debug Toolbars→ASM/Source Stepping。  
在这个调试工具栏选项表中，你可以看到许多  
调试工具的列表，并且你可以将想要的调试工具  
设置为可视。在菜单栏中，可视的工具名字旁有  
个校验标记。



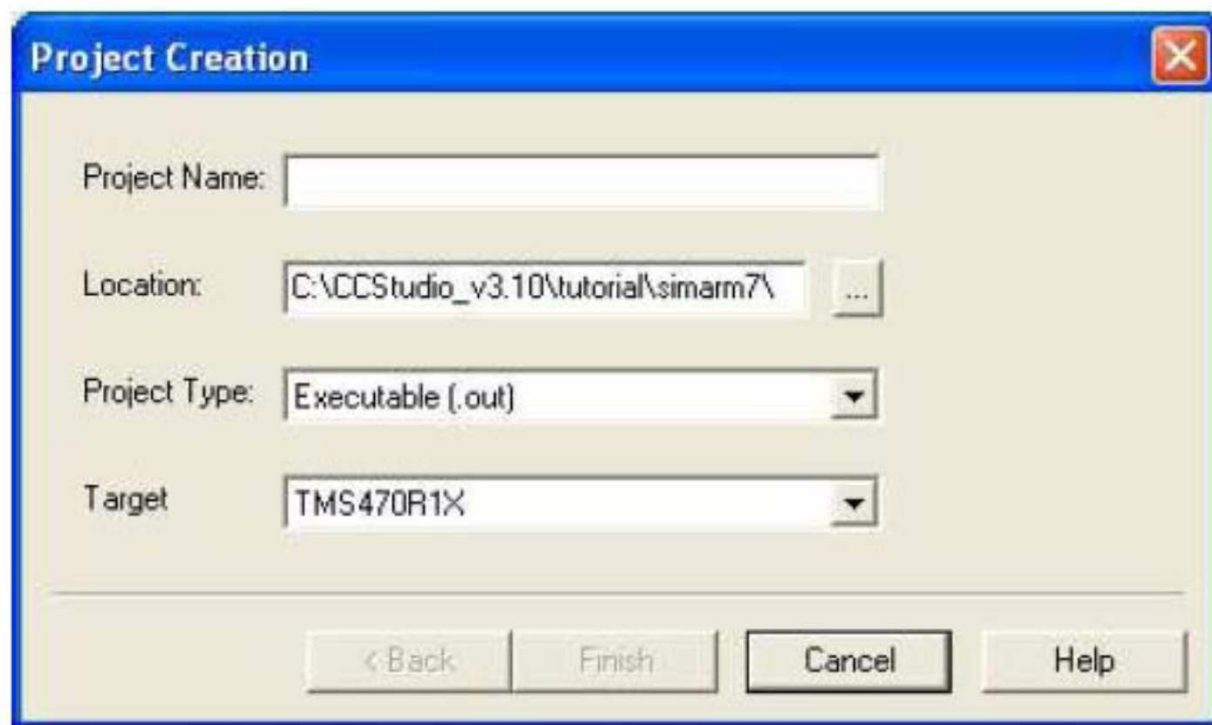


# 1. 创建新工程

- ④ 1. 如果CCS安装在C:\CCStudio\_v3.1，在C:\CCStudio\_v3.1\myprojects文件下新建一个practice文件夹。
- ④ 2. 把C:\CCStudio\_v3.1\tutorial\target\consultant文件夹下的内容复制到新建的文件夹里。目标与当前CCS的配置有关。
- ④ 3. 选择Project->New。
- ④ 4. 在Project Name框里输入工程名：（例如:practice）。

## 创建新工程

- 5. 在Location框里输入或者浏览第一步创建的文件夹。



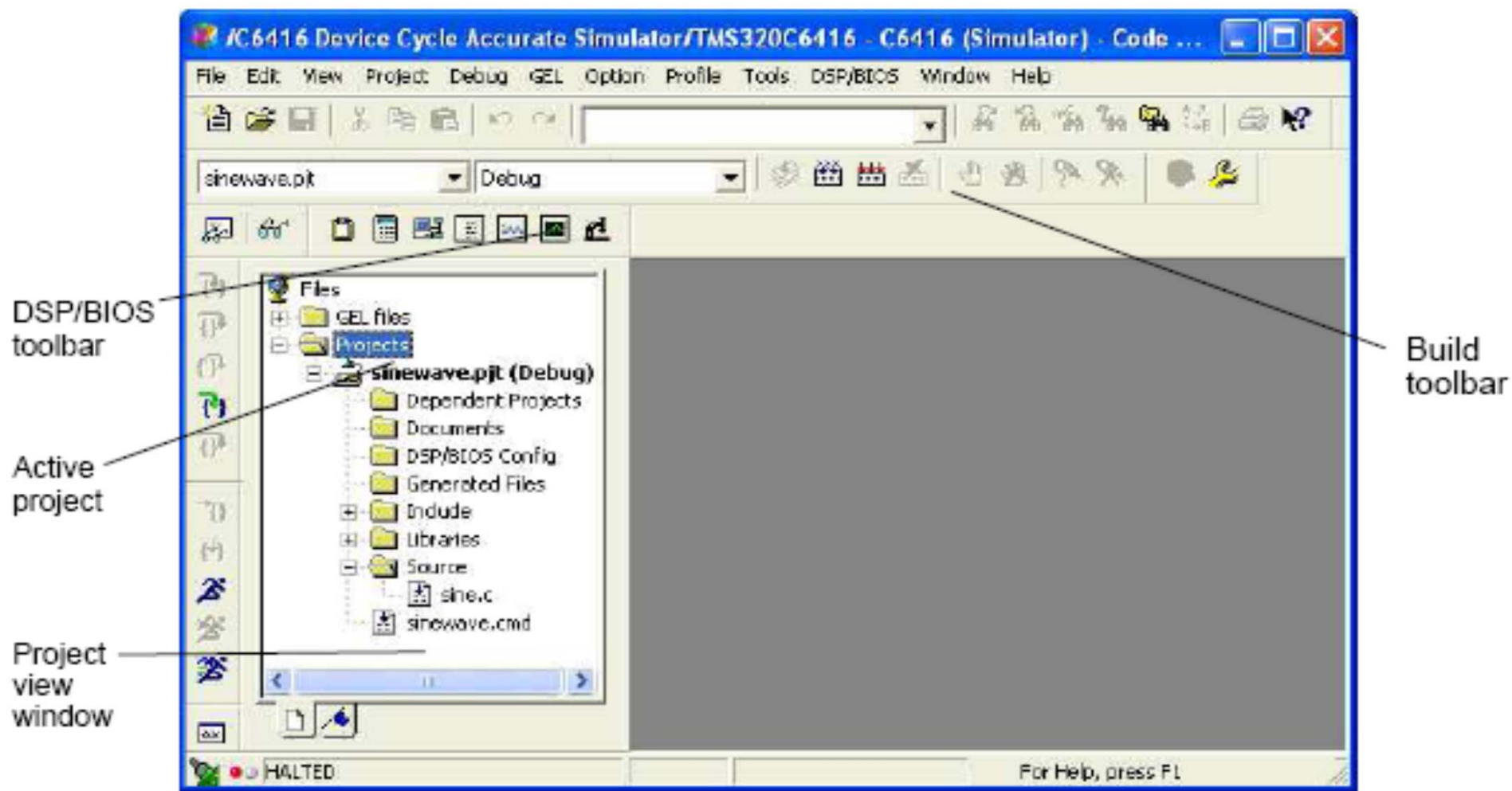
## 创建新工程

- ⑥ 6. 在Project Type一栏中，从下拉列表中选择工程文件的类型。这里选择执行文件（.out），也可以选择库文件（.lib）。在Target一栏中，为CPU选择目标板，当工具安装多个目标板上时，这个选择是必要的。
- ⑦ 7. 点击 Finish ，CCS创建一个叫做 practice.prj的工程文件。文件里存储了工程配置和工程所需要的各种相关文件。



# 创建新工程

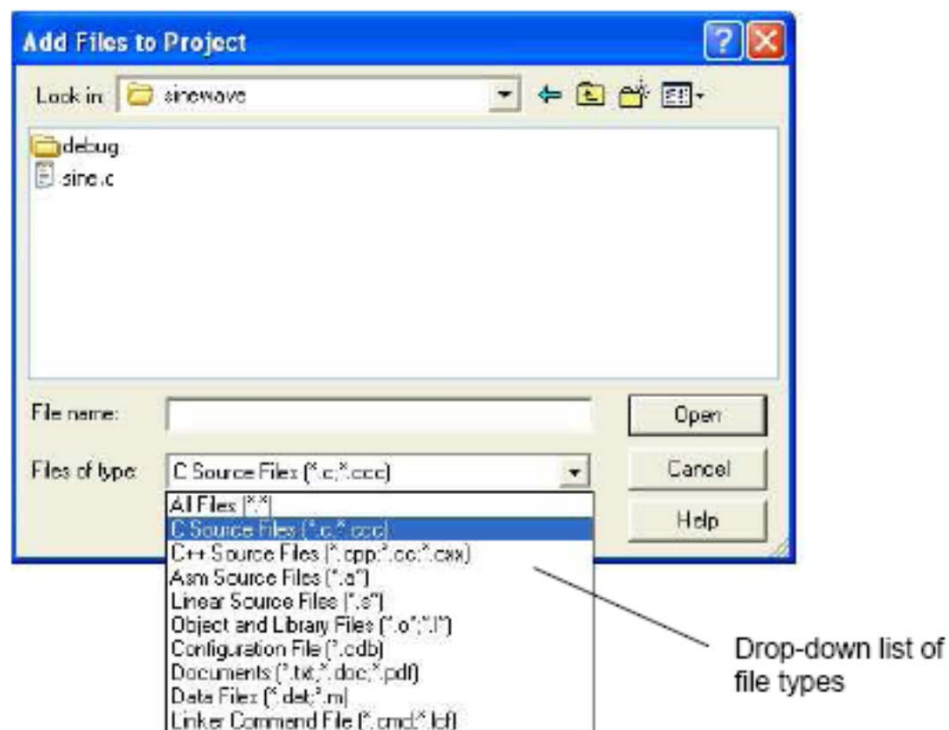
## CCStudio IDE 界面





# 创建新工程

8. 选择Project→Add Files to Project，或者工程视图（Project View）中的工程名上点击右键，选择Add Files to Project。显示加入文件到工程的对话框。





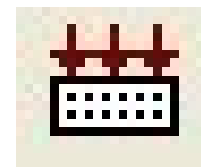
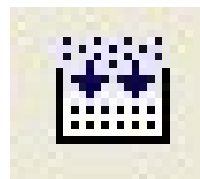
## 创建新工程

- 9. 从你所创建的文件夹里添加 `main.c`, `DoLoop.c`, 和 `Ink.c` (映射内存的连接命令文件)。浏览 `C:\CCStudio_v3.1\c6000\cgtools\lib\` 目录, 为所配置的目标添加相应的 `rts.lib`。
- 如果你想从工程中删除一个文件, 在工程视图 (Project View) 中右击文件名, 选择从工程中移除。

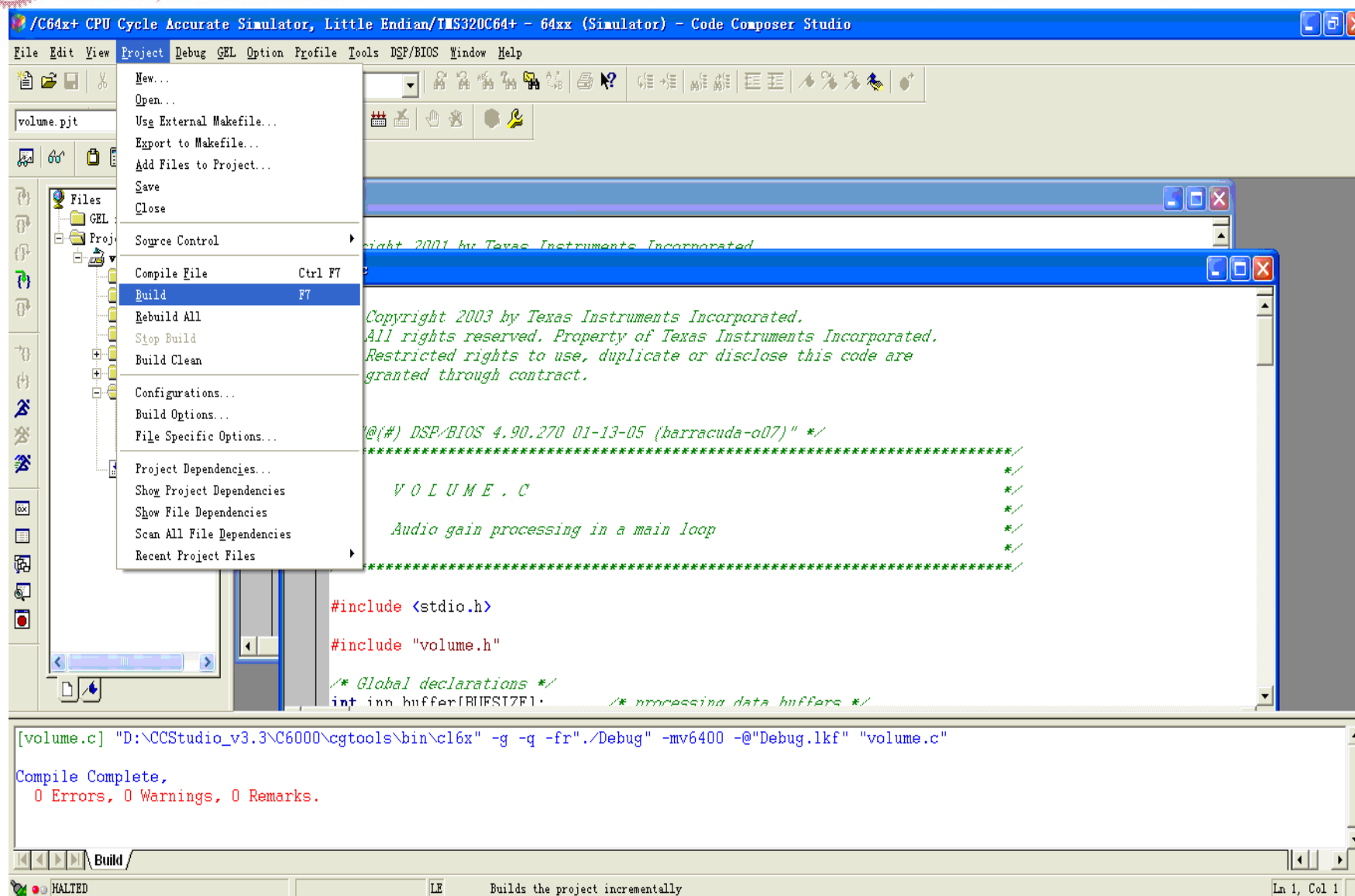


## 2. 构建程序

- 创建了一个功能程序后，你就可以构建（build）它。构建主要完成编译（compile）与连接（link）。第一次使用全构建（Build All）功能便可以构建工程，以后可以使用增量构建（Build the project incrementally）。一个输出窗口将会显示构建过程和状态。当构建完成后，输出窗口将会显示Build complete 0 errors, 0 warnings。



# 构建程序



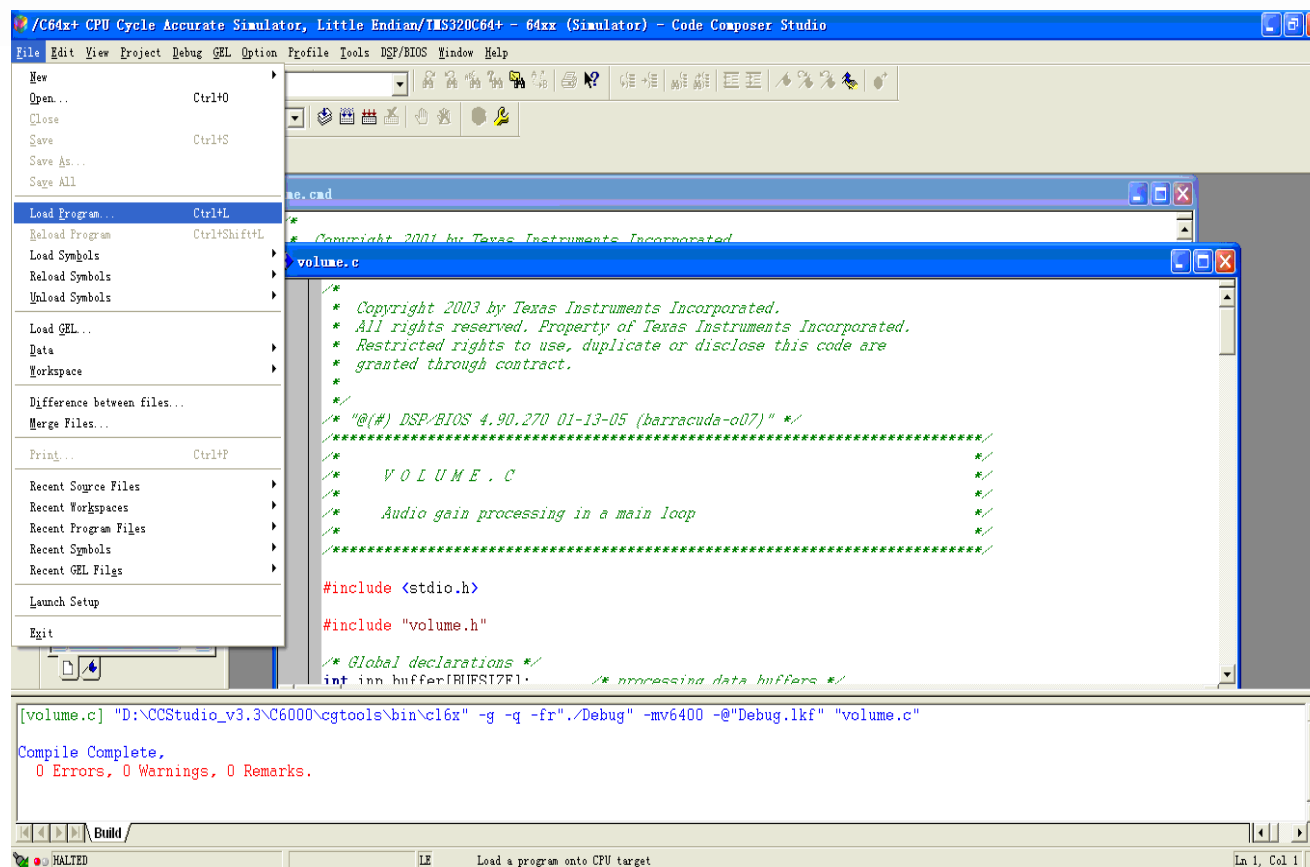
- ④ 当工程选项或所有文件发生改变后，执行 Rebuild All 命令重建工程。
- ④ 这些过程的信息将显示在底部的窗口中。在默认情况下，.out 文件将在你当前工程的 debug 目录下生成。可以在选择配置工具栏中选择不同的目录改变路径。

### 3. 加载程序

- ④ 程序成功构建后，执行File->Load Program加载程序。（也可以点击option-> custom->load program after build设置为自动加载构建后的程序）
- ④ 加载过程是将上述构建成功，生成的可执行文件加载到目标板，目标板可以是软件仿真环境，也可以是硬件目标板。默认情况下，CCS集成开发环境将会在你的工程路径下创建一个Debug子目录，把生成的.out文件放在里面。点击Open加载程序。



# 加载程序

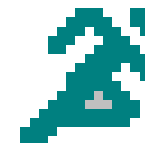


注：如果你修改并且重新构建了工程，切记通过 file→Reload 重新加载程序。

## 4. 基本调试

### 1. 运行

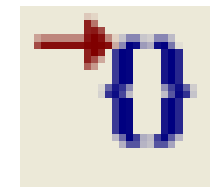
- ④ 可以运用这些命令来运行程序：
- ④ 主程序(Main) 可以通过先择Debug→Go Main, 来开始对主程序的调试。这个执行命令将会执行主程序函数。
- ④ 运行(Run) 在执行停止后, 可以通过点击Run按钮来继续运行程序。





# 运行

- 运行到光标处 (Run to Cursor) 如果想要程序运行到一个指定的位置，可以先把光标移到该位置，然后按下这个按钮。

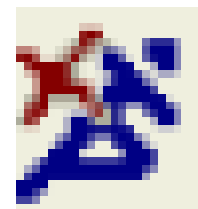


- 驱动 (Animate) 这个执行命令将一直运行程序直到运行到断点处。在断点处，执行停止并且将更新所有与任何试探点 (probe point) 没有联系的窗口。试探点 (probe point) 停止执行并更新所有图表及与之有关的窗口，然后继续运行程序。



# 单步调试

- ❶ 停止 (Halt) 最后，可以在任意时候按下停止按钮来终止程序执行。

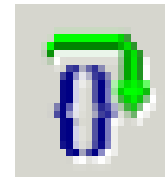
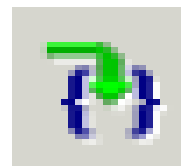
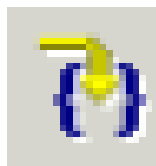


## 2. 单步调试

- ❷ 只有在执行程序的时候源程序和汇编程序的单步调试才可以使用。源程序的单步调试是通过单步执行源程序编辑器中所显示的代码行，而汇编程序的单步调试是通过单步执行反汇编窗口中显示的指令行。

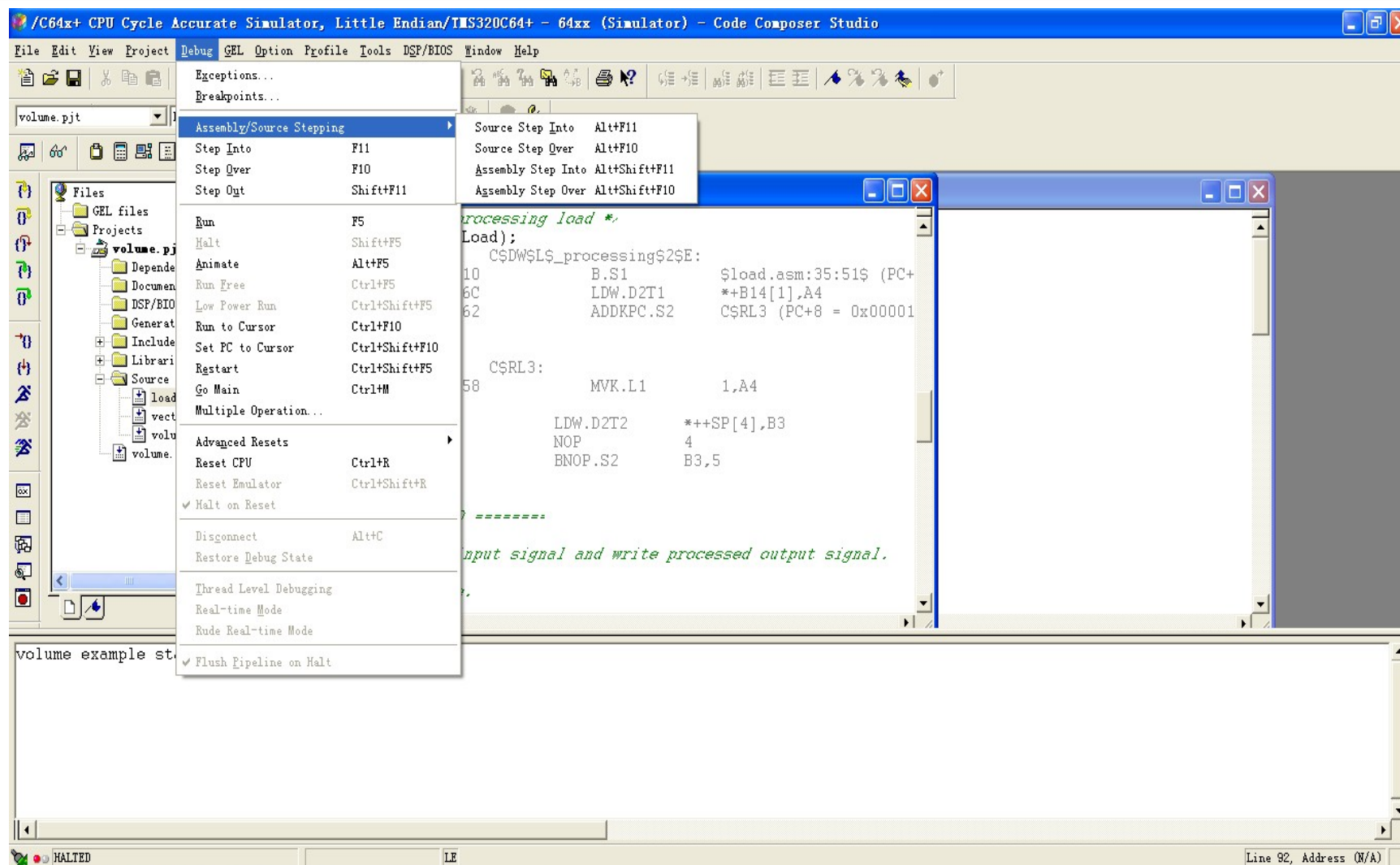
# 单步调试

- 通过View→Mixed Source/ASM来切换源程序/汇编程序混合模式，可以同时查看源代码的汇编代码。
- 执行一个单步调试命令，先在工具栏中选择合适单步调试图标。另一种方法是先选择Debug→Assembly/Source Stepping（然后选择合适的命令）。





# 单步调试



单步调试共有三种：

- ① 单步调试或者只执行一个表达式然后就终止程序执行。
- ② 跳过整个函数的执行然后当函数返回时终止程序。
- ③ 跳出执行当前的子程序并返回到调用函数入口。  
当返回到调用函数入口时，程序就终止了。

## 3. 断点

- ❶ 断点会停止程序的执行。当程序停止时，可以检查程序的状态，检查或修改变量，检查调用堆栈等等。断点可以设置在编辑窗口中的任意一行源代码中或者设置在反汇编窗口的任意一个反汇编指令上。在设置完一个断点后，可以启用断点也可以禁用断点。





## 软件断点:

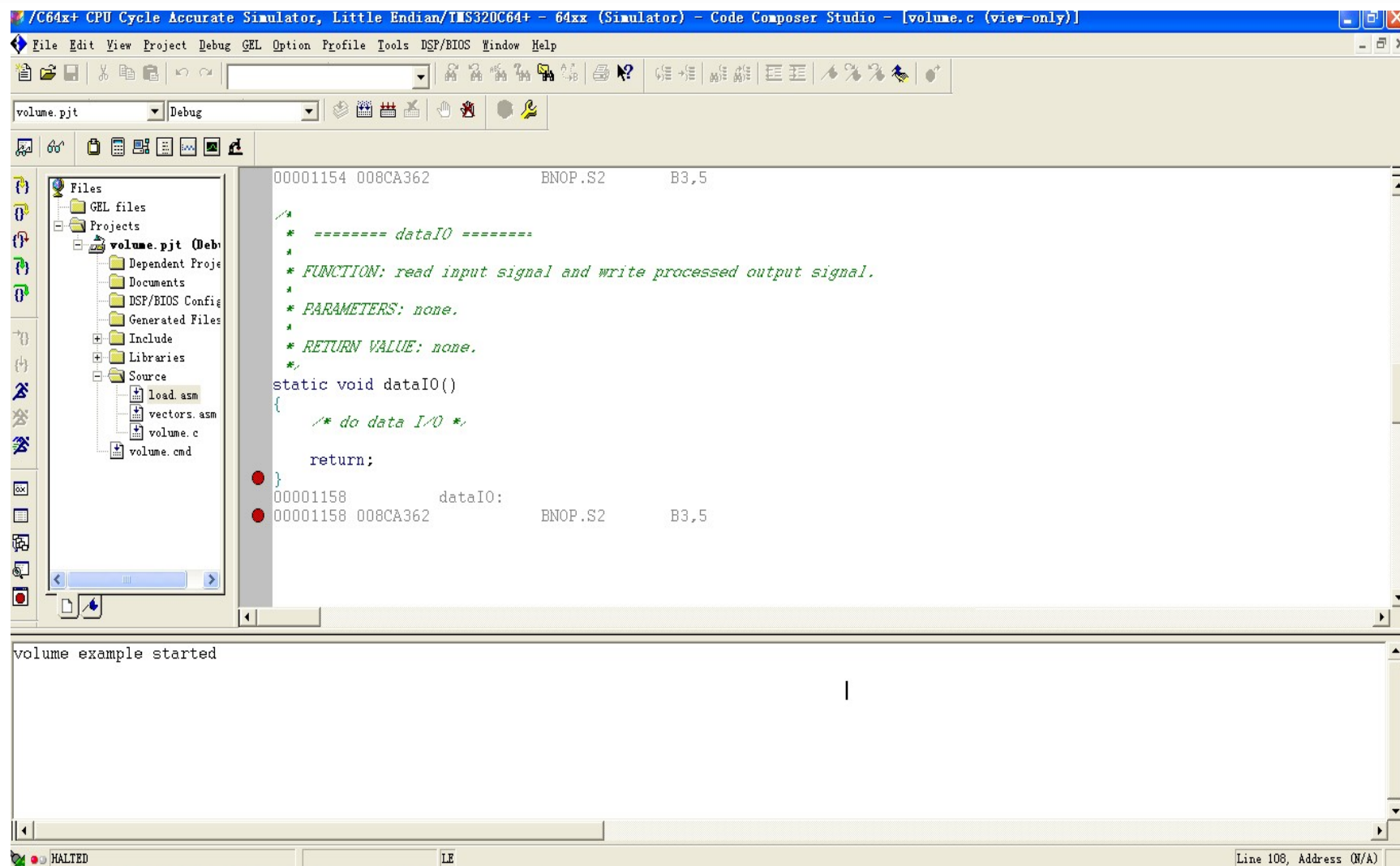
- ④ 可以在任意一个反汇编窗口或者含有C/C++源代码的文档窗口设置断点。只要断点设置的位置合适，对于断点的数量便没有限制。软件断点通过改变目标程序使之在需要的位置增加一条断点指令。

## 设置软件断点的方法：

- ① 1. 在一个文档窗口或者反汇编窗口，移动指针到你想要设置断点的那一行。
- ① 2. 当你在文档窗口设置断点时，只需在选定行的前面的页边空白处迅速双击即可。若是在反汇编窗口，则只需在选定行双击。
- ① 在选定行的页边空白处的一个实心红点即为断点标志，它表示在所需要的位置已经设定了一个断点。



# 软件断点



- ④ 我们也可以使用切换断点命令和切换断点按钮来迅速的设置和清楚断点。
- ④ 1. 在一个文档窗口或者反汇编窗口，移动指针到你想要设置断点的那一行。
- ④ 2. 点击鼠标右键并选择切换断点，或者在软件工具栏中点击切换断点标志按钮。

# 硬件断点

硬件断点与软件断点不同的是它们并不改变目标程序，而是利用芯片上可以利用的硬件资源。硬件中断的用途是在只读存储器或者存储进程中设置断点，而不是获取指令。可以使用的硬件断点的数量取决于所采用的DSP型号。在仿真目标上不能实现硬件断点。

## 设置硬件断点的方法：

- ① 1. 选择Debug->Breakpoints。在选择断点这一栏后，便会出现Break/Probe Points对话框。
- ② 2. 在Breakpoint type一栏，选择H/W Break作为指令获取断点，或者在特定位置选择Break on <bus> <Read|Write|R/W>作为存储读取断点。
- ③ 3. 在程序或存储器中你想设置断点的某个位置，按以下方法中的一种操作：

# 硬件断点

- ④ — 对于一个绝对地址，你可以输入任意C语言中的表达方式：C的函数名或者一个标志符号。
- ④ — 输入断点的位置基于你的C源文件。当你不知道C指令在可执行文件中的位置时，这就很方便了。在基于C源文件的位置输入的格式是：文件名第几行 总行数。



# 硬件断点

- ④ 4. 在计数这一栏，输入断点产生前，该处指令需要运行的次数。如果计数设为1，则每次到该位置便产生断点。
- ④ 5. 点击添加按钮可以产生一个新的断点。这样便可创造一个新的断点并对其激活。
- ④ 6. 点击OK。

## 4. 观察窗口:

- 观察窗口允许用户观察局部变量和全局变量还有C/C++表达式的值。（关于观察窗口主题的在线帮助：Help→Contents→Debugging→Viewing Debug Information→Watch Window。）

### 打开观察窗口:

- 1. 选择View→Watch Window，或者点击观察工具栏上的观察窗口图标按钮。观察到窗口包含两个统计表：Watch Locals和Watch 1。

## 观察窗口

- ④ -在Watch Locals统计表中，调试器自动显示当前正在执行函数的局部变量的名称、值的大小、类型和基的选择（Radix option）。
- ④ -在Watch 1统计表中，调试器显示局部变量、全局变量和用户指定表达式的名称、值的大小、类型和基的选择（Radix option）。
- ④ 2. 选择File→Load Program。
- ④ 3. 双击在Project View 中的filename.c 文件（filename为文件名）。

- ④ 4. 把光标移动到允许打断点的行。
- ④ 5. 点击 Toggle Breakpoint 工具栏按钮或按下 F9。被选择的页面空白处会显示出断点已经建立好了（红色图标）。
- ④ 6. 选择 View->Watch Window。在窗口的右下角会出现一个单独的区域，在运行过程中这个区域显示的就是被观察的变量的值。默认情况下，显示的是 Watch Locals 统计表，显示内容是执行过的函数的局部变量的值。



## 观察窗口

- 7. 如果不是在主函数，选择Debug->Go Main。
- 8. 选择Debug->Run, 或按下F5, 或按下运行图标。观察窗口会更新局部的值。

Name	Value	Type	Radix
input	0x00002538	int *	hex
output	0x000026C8	int *	hex
size	21	int	dec

Watch Locals

Watch 1

## 观察窗口

- 9. 选择 Watch 1 统计表.
- 10. 在Name column点击Expression图标并且输入需要观察的变量的名称。
- 11. 点击窗口的空白处可以保存所做的改动。值会立即显示出来，就像下面这个例子。



- 12. 点击Step Over工具栏按钮或按下F10来跳过对要观察的变量的调用。

除了观察一个简单的变量的值之外, 用户还可以观察一个结构体中的元素的值。

方法如下:

- 1. 选择Watch 1 统计表。
- 2. 点击Name 栏中表达式图表并且输入需要观察的expressions的名称。



- ④ 3. 点击窗口的空白处用来保存所做的改动。
- ④ 4. 一旦点击“+”标记，该目录会展开并列出色体中所有的元素以及他们对应的值。  
(所示的连接地址可能会有所不同。)
- ④ 5. 双击在结构体中任意一个元素的值，就可以对这个值进行编辑。
- ④ 6. 改变这个变量的值。

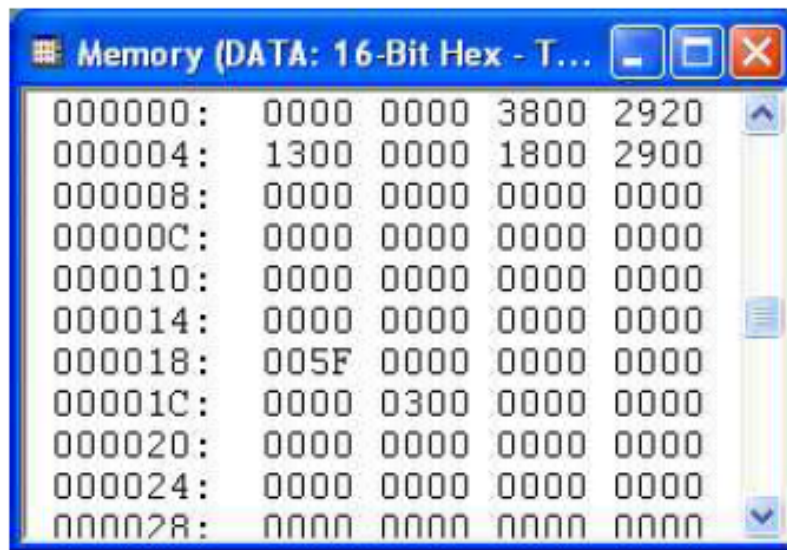
# 观察窗口

- 需要注意的是在观察窗口中的值如果发生了改变，这个值的颜色也会变成红色用来表示它已经被手动进行了修改。



## 5. 内存窗口（Memory Window）

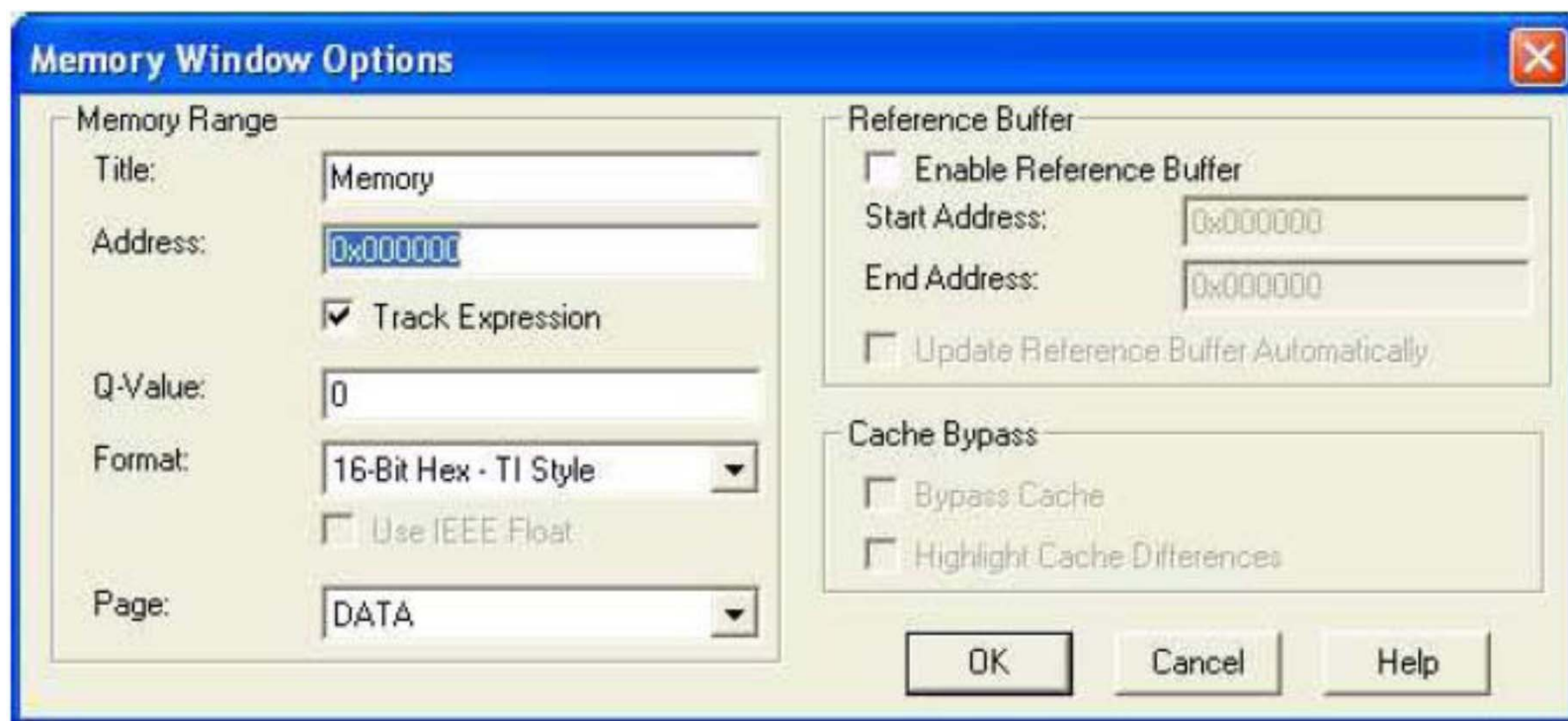
- 内存窗口允许用户观察由指定地址开始的存储单元中的内容。用户可以通过选项对内存窗口的显示进行格式化，也可以编辑被选择的存储单元的内容。





# 内存窗口

- 可以在内存窗口选项对话框中定义内存窗口不同的特性。



# 内存窗口

- ④ 该对话框提供了以下内存窗口选项：
- ④ Title 标题 为内存窗口输入一个有意义的名字。  
当打开内存窗口时，这个名字会显示在标题栏上。  
当有多个内存窗口打开时，标题会起到十分重要的作用。
- ④ Address地址 输入需要观察的存储单元的起始地址。



# 内存窗口

- ④ Track Expression 跟踪表达式 点击这个选项会使内存窗口自动地重新评估并且改变它基于与起始地址相关联的表达式的起始地址。
- ④ Q值 用户可以用Q值来显示整数。这个值将整数值表示成更精确的二进制值。小数点被插到二进制值中，最低有效位(LSB) 产生的偏移量由Q值决定。
- ④ Format 格式 从下拉菜单中选择内存显示的格式。

# 内存窗口

- ④ Enable Reference Buffer 参考缓冲器有效 为指定的内存区域保存一个快照，可以用来为后面的比较做准备。
  - ④ Start Address 起始地址 输入想要保存在参考缓冲器中的存储单元的起始地址。
  - ④ End Address 终止地址 输入想要保存在参考缓冲器中的存储单元的终止地址。
- (以上两个区域只有当“激活参考缓冲器”选择以后才会激活。)



# 内存窗口

- ④ Update Reference Buffer Automatically 自动更新参考缓冲器 选择这个复选框可以自动地用指定地址区域的当前内存内容覆盖参考缓冲器的内容。这个选项只有当选择了“Enable Reference Buffer”以后才会激活。
- ④ Bypass Cache 旁路高速缓存 该选项使得内存总是从物理内存中读取内存内容。 如果这个选项被激活，CCS将忽略或者绕过高速缓存的内容。

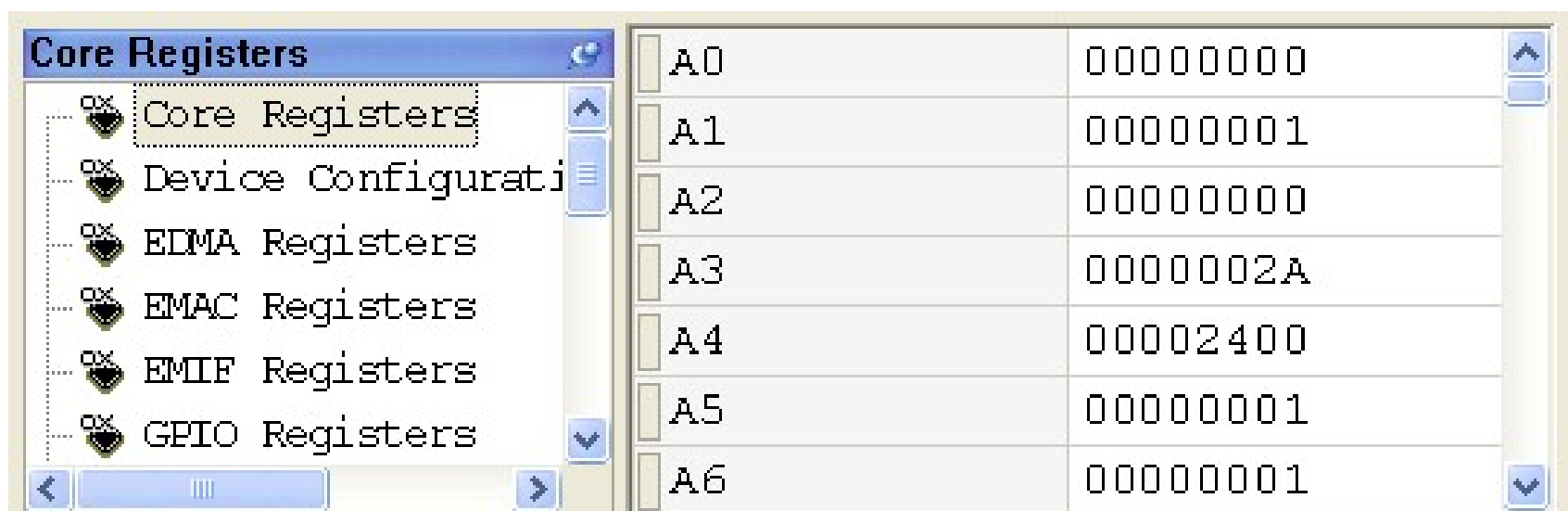


# 内存窗口

- ④ Highlight Cache Differences 突出高速缓存的差异 当高速缓存的值和物理值不一致时，这个选项突出强调了的存储单元的值。也会用色彩来加强突出高速缓存的差异。选择Option→Customize→Color 并且选择在Screen Element下拉框中的Cache Bypass Differences 选项。

## 6. 寄存器窗口 (Register Window)

- 用户可以在寄存器窗口观察并编辑选中的不同寄存器的内容。



## 寄存器窗口

- 要访问寄存器窗口，选择View→Registers并且选择需要观察/编辑的寄存器组。
- 要访问寄存器的内容，选择Edit→Edit Register，或者从寄存器窗口双击一个寄存器，或者右击一个寄存器并选择Edit Register（编辑寄存器）。





## 7. 反汇编模式/混合模式 ( Disassembly/Mixed Mode )

- ① 1 . 反汇编模式 ( Disassembly Mode )
- ② 当你加载程序到实际的或模拟的目标板时，调试器会自动打开一个反汇编窗口。
- ③ 反汇编窗口显示反汇编的指令和符号信息以供调试需要。反汇编与汇编过程相反，并允许内存的内容以汇编语言代码的形式显示。符号信息包括符号和代表目标板上地址或值的文字数字式字符串。



## 反汇编模式

- 如果以单步命令运行程序，程序计数器将跟随这些指令向下执行。

```
Disassembly (c_int00)
00002260 c_int00:
➔ 00002260 0798FE2A
00002264 0780006A
00002268 07BF09F2
0000226C 0715962A
00002270 0700006A
00002274 0208B02A
00002278 0200006A
0000227C 00100362
00002280 021B0028
```

## ② 2. 混合模式 ( Mixed Mode )

- ② 除了在分解窗口里查看分解的指令，调试器允许以C源代码和分解代码交叉显示方式察看。点击 View Mixed Source/ASM，或在源文件窗口中右击，根据你目前的模式选择Mixed Mode或Source Mode。





# 混合模式



## Mixed Mode

```
volume.c (view-only)
*output++ = *input++ * gain;
000010E4      C$DW$LS_processing$2$B:
000010E4 033C22E6      LDW.D2T2      *+SP[1],B6
000010E8 0200006E      LDW.D2T2      *+B14[0],B4
000010EC 03BC42E6      LDW.D2T2      *+SP[2],B7
000010F0 00002000      NOP          2
000010F4 029836E6      LDW.D2T2      *B6++[1],B5
000010F8 033C22F6      STW.D2T2      B6,*+SP[1]
000010FC 0F9C805A      ADD.L2        4,B7,B31
00001100 0FBC42F6      STW.D2T2      B31,*+SP[2]
00001104 00000000      NOP
00001108 04948882      MPYLH.M2      B4,B5,B9
0000110C 0410A882      MPYLH.M2      B5,B4,B8
00001110 02948F82      MPYU.M2      B4,B5,B5
00001114 0221207A      ADD.L2        B9,B8,B4
00001118 02120CA2      SHL.S2        B4,0x10,B4
0000111C 0210A07A      ADD.L2        B5,B4,B4
00001120 021C02F6      STW.D2T2      B4,*+B7[0]
00001124 00002000      NOP          2
```

## Source Mode

```
volume.c (view-only)
/* additional processing load */
load(processLoad);
0000113C      C$DW$LS_processing$2$E:
0000113C 00024010      B.S1          $load.asm:35:51$ (PC+
00001140 0200016C      LDW.D2T1      *+B14[1],A4
00001144 01826162      ADDKPC.S2     C$R13 (PC+8 = 0x00001

return(TRUE);
00001148      C$R13:
00001148 0204A358      MVK.L1        1,A4
}
0000114C 01BC92E6      LDW.D2T2      *++SP[4],B3
00001150 00006000      NOP          4
00001154 008CA362      BNOP.S2       B3,5

/*
 * ===== dataIO =====
 *
 * FUNCTION: read input signal and write processed output signal.
 *
 * PARAMETERS: none.
 */
```

# TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP，MSP430等技术培训与技术服务，项目合作等。培训内容有

- 1) CCS开发环境精解与实例；
- 2) DSP/SYS BIOS 实例；
- 3) C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；
- 4) HPI，EMIF，EDMA，Timer等外设；
- 5) C6416、DM642，C6678多核EVM开发平台实例；
- 6) Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），

邮件报名：[jhniu@sjtu.edu.cn](mailto:jhniu@sjtu.edu.cn)，

[niu\\_jinhai@yahoo.com.cn](mailto:niu_jinhai@yahoo.com.cn)

## 关于帮助

- ④ 可以进入help → contents来一步步指导你学习一些主题，进入Tutorial来查阅在线帮助站点，这些网站提供了最新的主题帮助，或者在PDF中浏览user manuals来获得一些重要的细节信息或方法。你可以通过help →update advisor来获得最新的一些功能特征。
- ④ Code Composer Studio IDE指南包含了帮助你快速启动该软件的教程。选择Help→Tutorial 来打开Code Composer Studio IDE的在线指南工具。



The screenshot shows the Code Composer Studio (CCS) 3.3 IDE. The main window displays a build error for the project 'volume1.pjt'. The error message is: "Build error: cannot run C:\CCStudio\_v3.3\bin\cc3\_32\bin\cc3\_32.exe [volume.c] \"C:\CCStudio\_v3.3\bin\cc3\_32\bin\cc3\_32.exe\", line 23: fatal error: cc3\_32.exe: 1 fatal error detected in the compiler. Compilation terminated." Below the error message, it says ">> Compilation failure" and "[load.s62] \"C:\CCStudio\_v3.3\bin\cc3\_32\bin\cc3\_32.exe\"". At the bottom, it says "Build Complete, 2 Errors, 0 Warnings, 0 Remarks.".

Overlaid on the IDE is the "Code Composer Studio Tutorial" window. The tutorial is titled "Developing a Simple Program" and is part of the "CCS - L1" series. It contains the following sections:

- Developing a Simple Program**
  - In this lesson, you develop a simple program that performs basic signal processing.
  - Learning Objectives:**
    - Create a simple program using Code Composer Studio™ tools
    - Use basic debug techniques to analyze the program
    - Use the tools to facilitate development of programs
  - Examples used in this lesson: volume1**
  - Application Objective:**

This lesson uses the volume1 example to develop and run a simple program. You will create a project from scratch, add files to it, and review the code. After you build and run the program, you will change build options using the build options dialog, and fix syntax errors using the editor. Finally, you will utilize basic debug techniques like breakpoints, watch windows, and the command window.

The tutorial window also includes a table of contents with the following items:

- How to Print
- Tutorial Example List
- Code Composer Studio IDE
- Introduction
- Developing a Simple Program
  - Overview
  - Creating a New Project
  - Adding Files to a Project
  - Reviewing the Source Code
  - Building and Running the Program
  - Changing Program Options and Fixing Syntax Errors
  - Using Breakpoints and the Watch Window
  - Using the Watch Window with Structures
  - Adding a Breakpoint for File I/O
  - Displaying Graphs
  - Animating the Program and Graphs
  - Adjusting the Gain
- Project Management
- Editing Techniques
- Using Debug Tools
- Data Visualization
- Profiling Code Execution
- Using GEL Language
- Configuring Target Devices
- Application Code Tuning
- Advanced Event Triggering
- Optimizing C
- RTDX
- Real-Time Emulation
- Trace Tutorial

the Office of the President



# TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP，MSP430等技术培训与技术服务，项目合作等。培训内容有

- 1) CCS开发环境精解与实例；
- 2) DSP/SYS BIOS 实例；
- 3) C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；
- 4) HPI，EMIF，EDMA，Timer等外设；
- 5) C6416、DM642，C6678多核EVM开发平台实例；
- 6) Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），

邮件报名：[jhniu@sjtu.edu.cn](mailto:jhniu@sjtu.edu.cn)，

[niuujinhai@yahoo.com.cn](mailto:niuujinhai@yahoo.com.cn)



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



## 颁发TI授权的培训证书



SHANGHAI JIAO TONG UNIVERSITY

the Office of the President



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



End

Thanks