

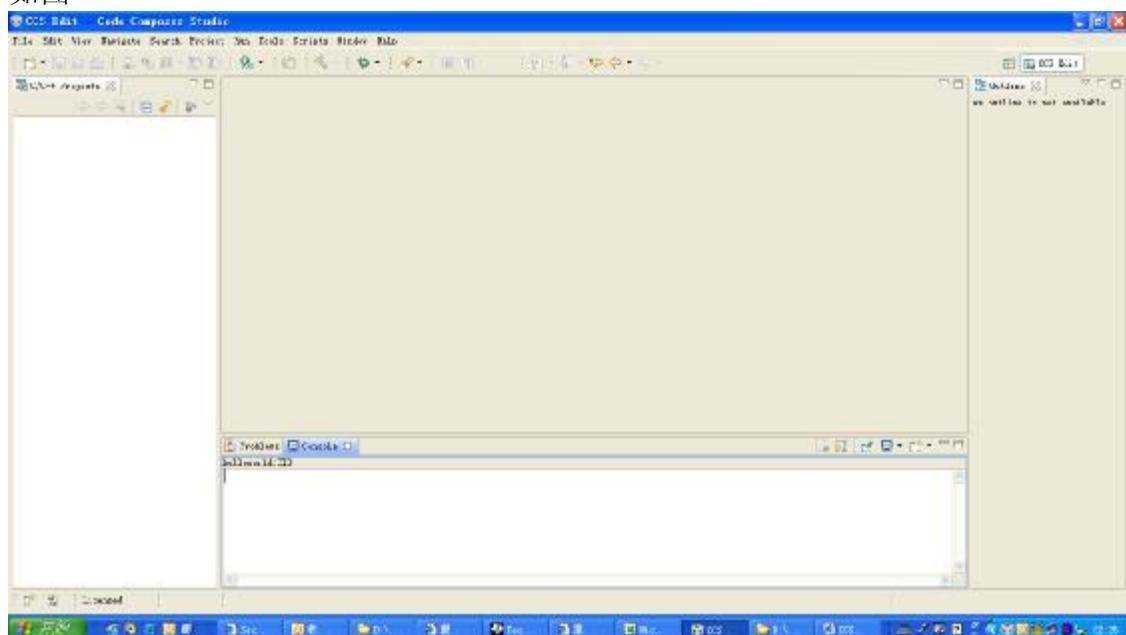
## TMS320C6678入门笔记

艾睿电子 蒋新刚

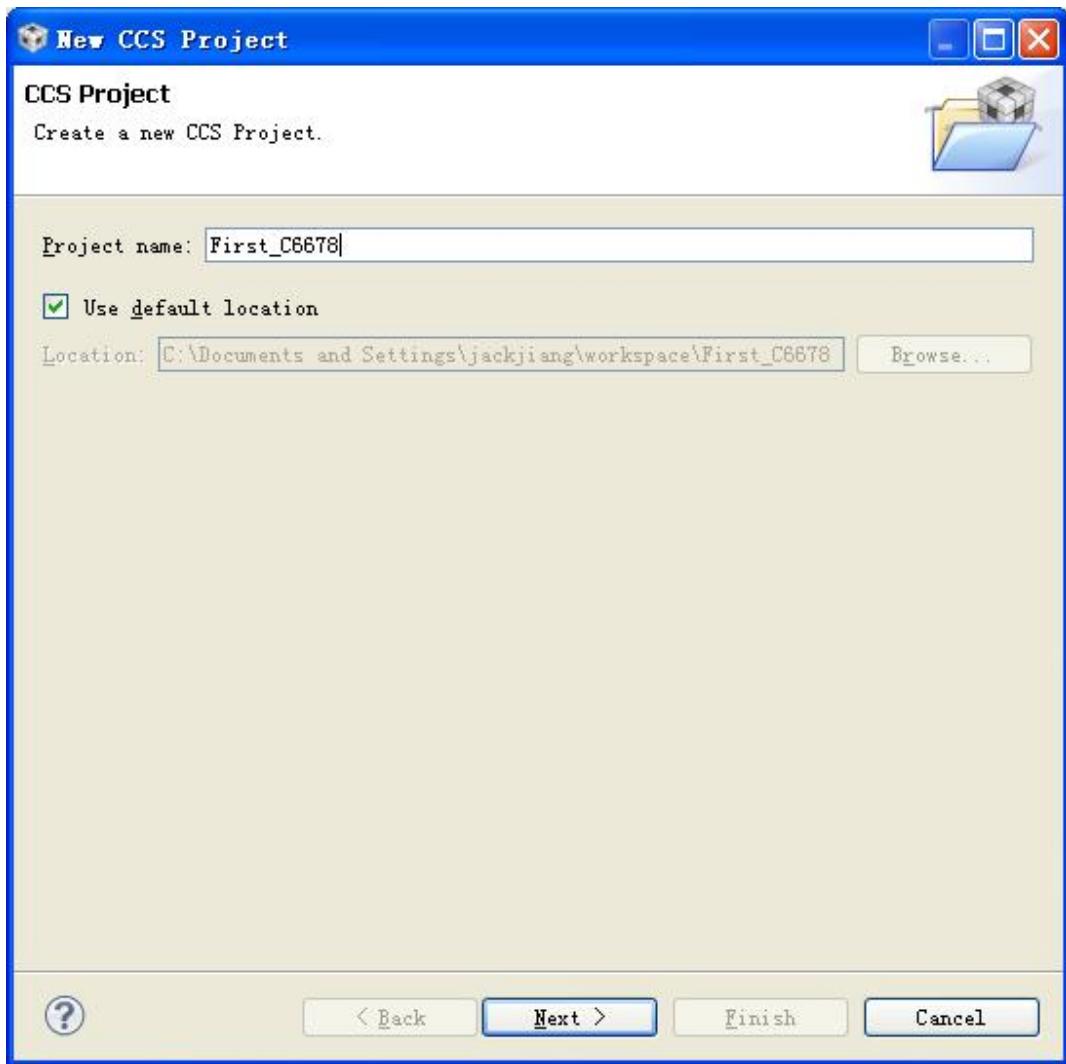
本入门笔记，通过建立一个最简单 TMS320C6678 的工程文件，详细记录了如何建立工程，如何配置工程文件，如何进行工程调试，其目的是给予初次学习多核器件 TMS320C6678 的网友以指导！不足之处，请见谅！

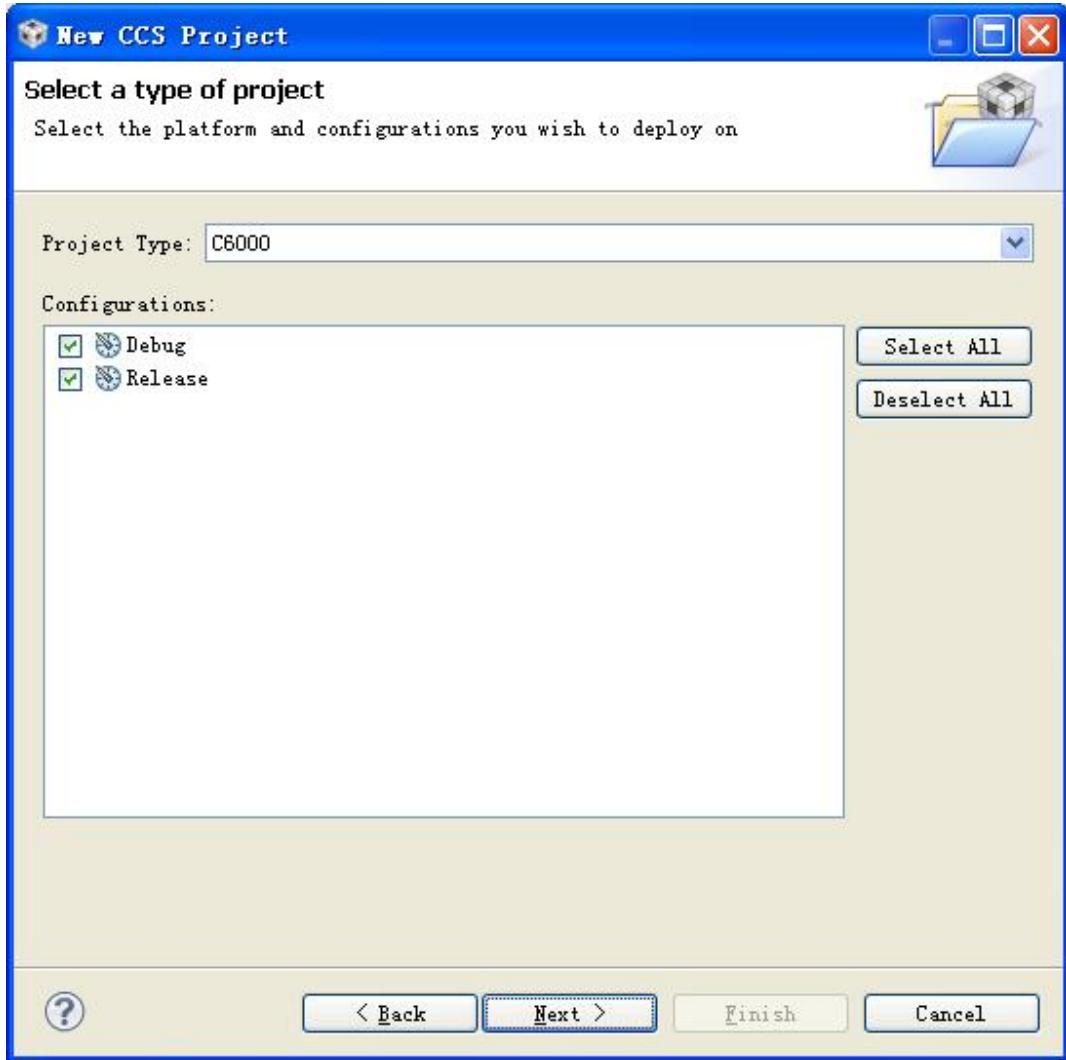
第一步，安装 CCSV5，很简单，此处略过

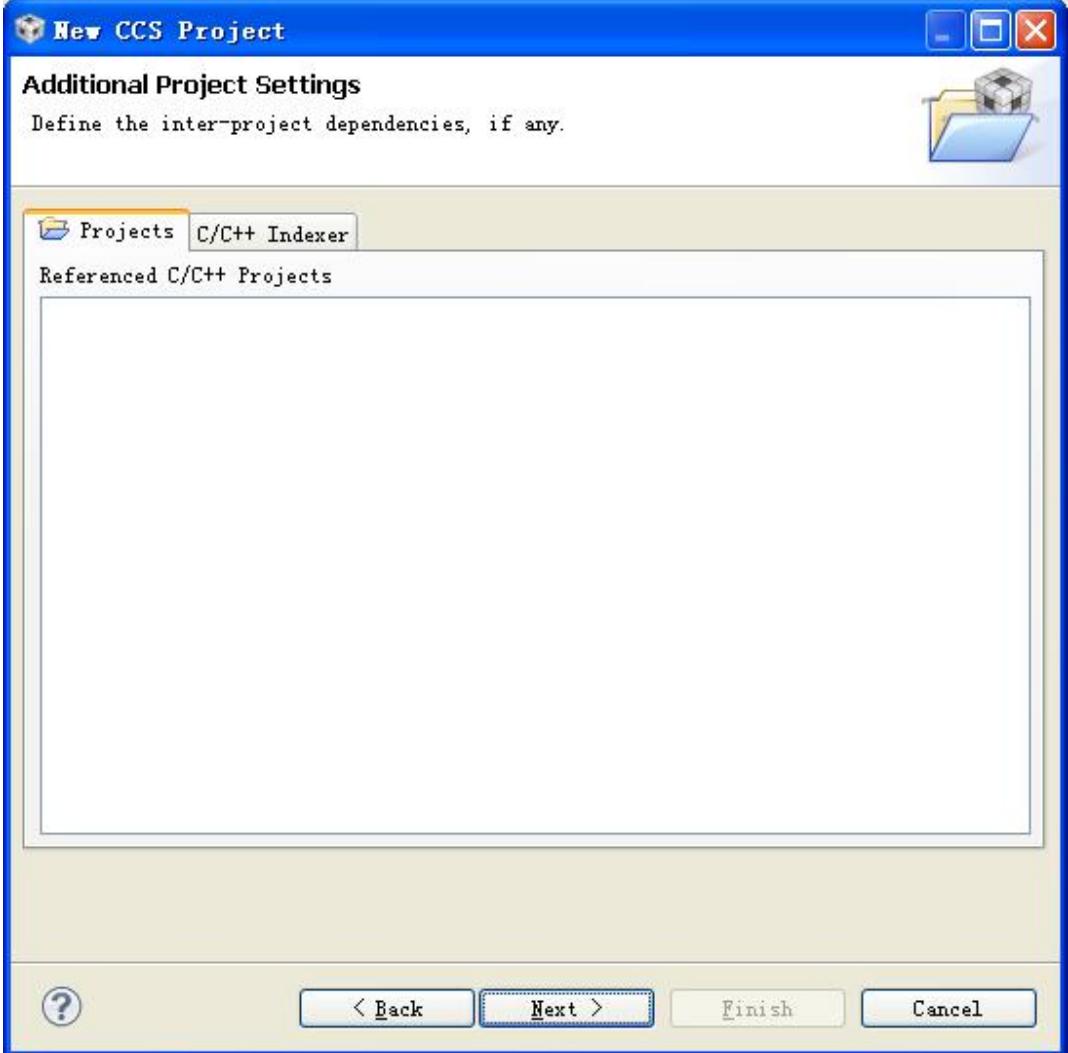
第二步，运行 CCSV5，起动过程中会要求设置工作区路径，CCSV5 起动后，界面如图

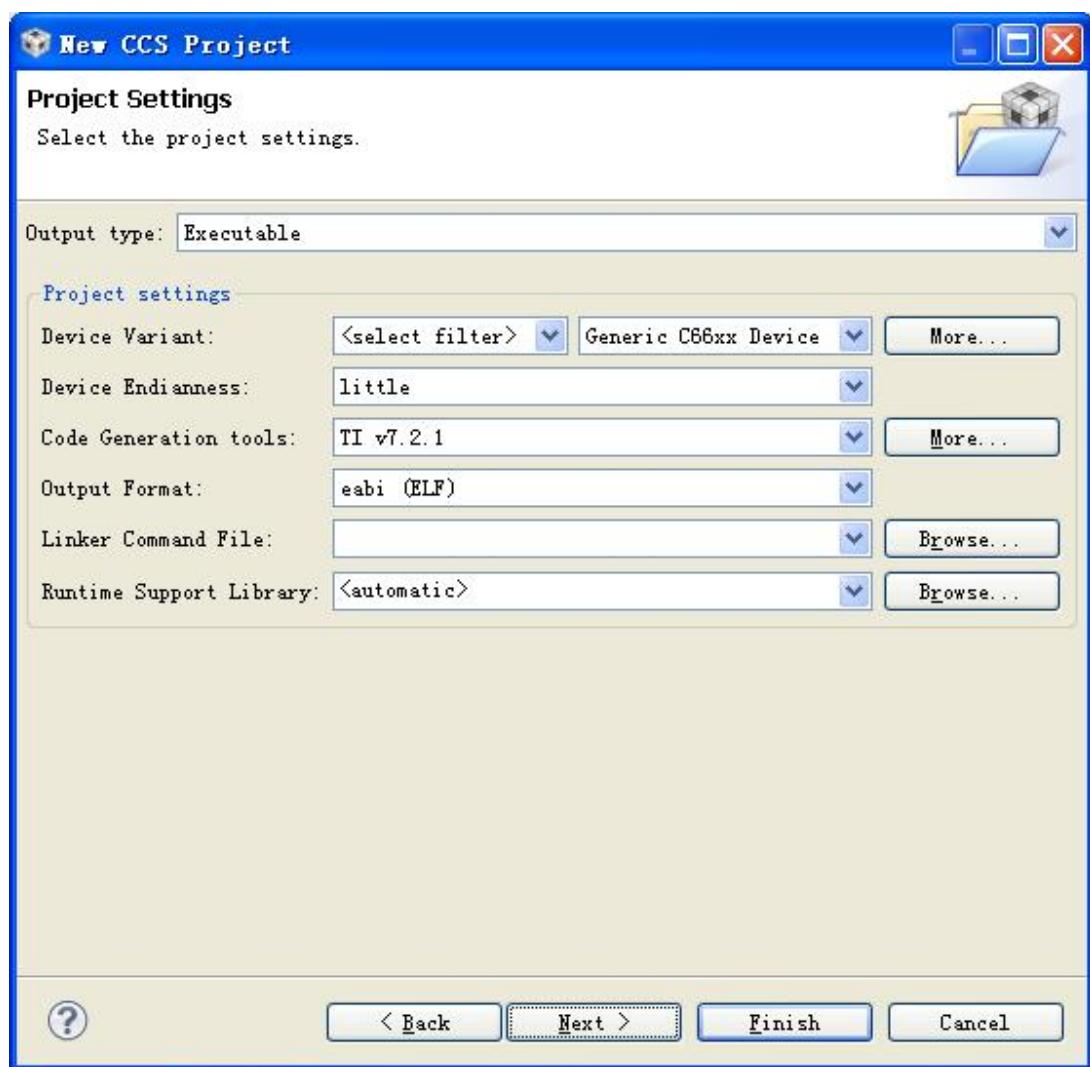


第三步，新建工程，选择 file->new->ccs project,输入工程名，点击 next

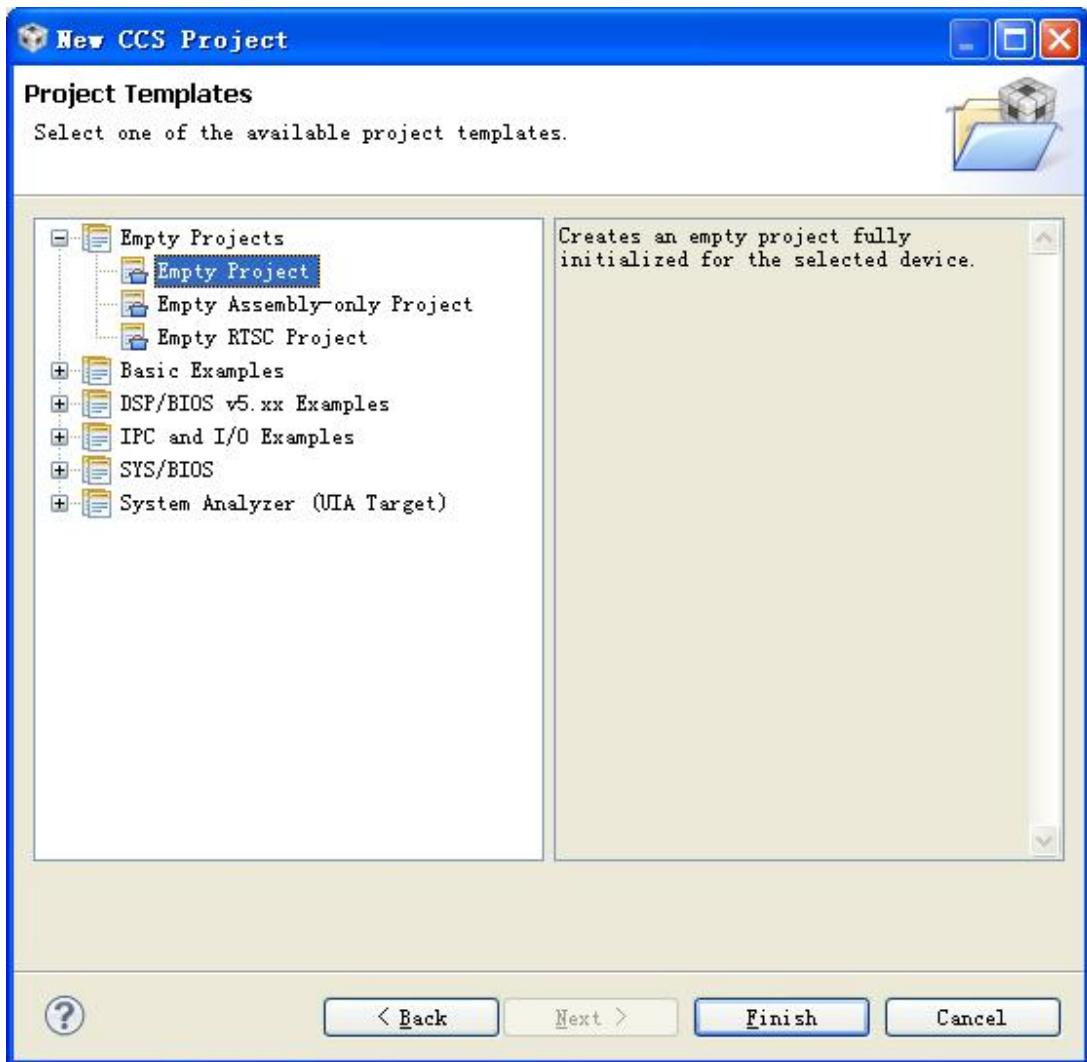




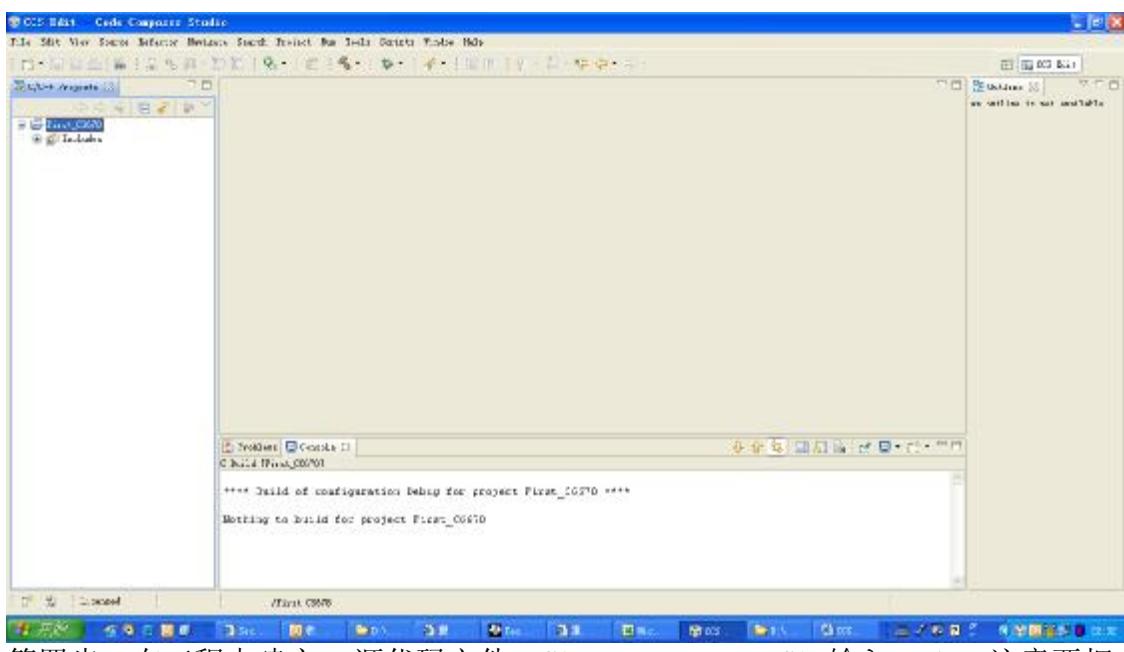




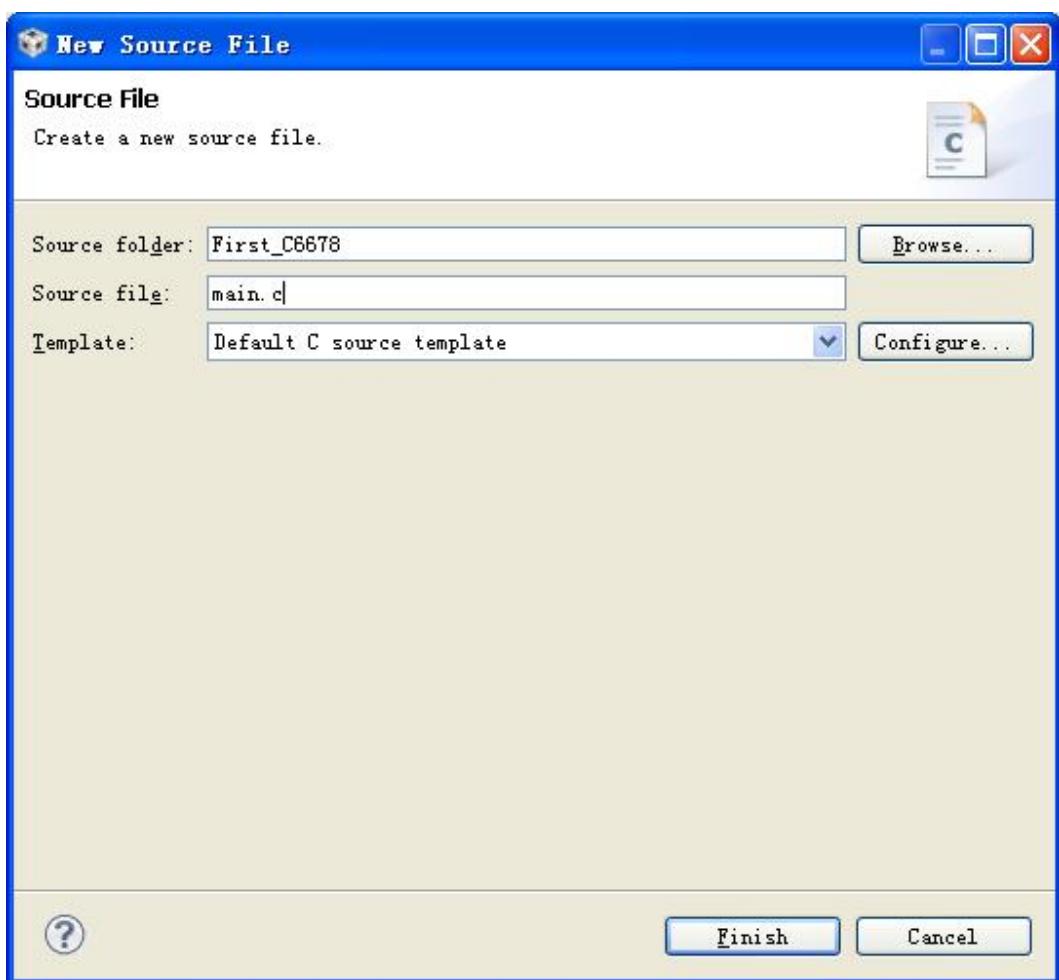
请注意工程中的大小端设置要与 TMS320C6678 评估板上的设置一致



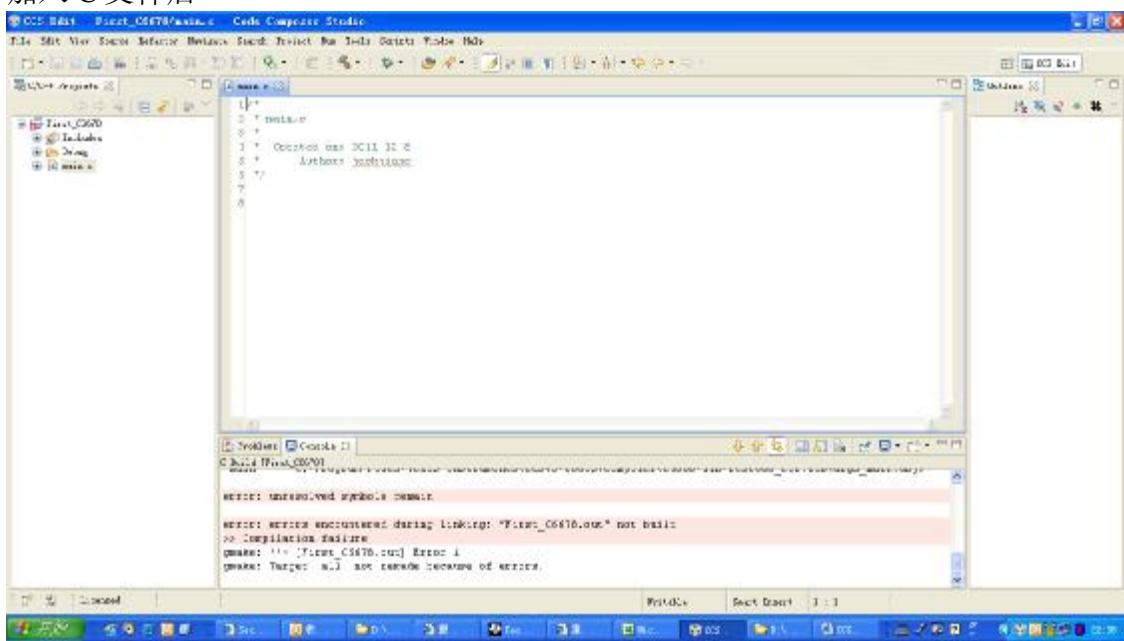
点击 finish, 完成工程建立



第四步，在工程中建立 C 源代码文件，file->new->source file, 输入 main.c, 注意要把文件名后缀带上，要不 CCSV5 会提示，不知道文件类型



加入 C 文件后



## 在 C 文件中输入源代码

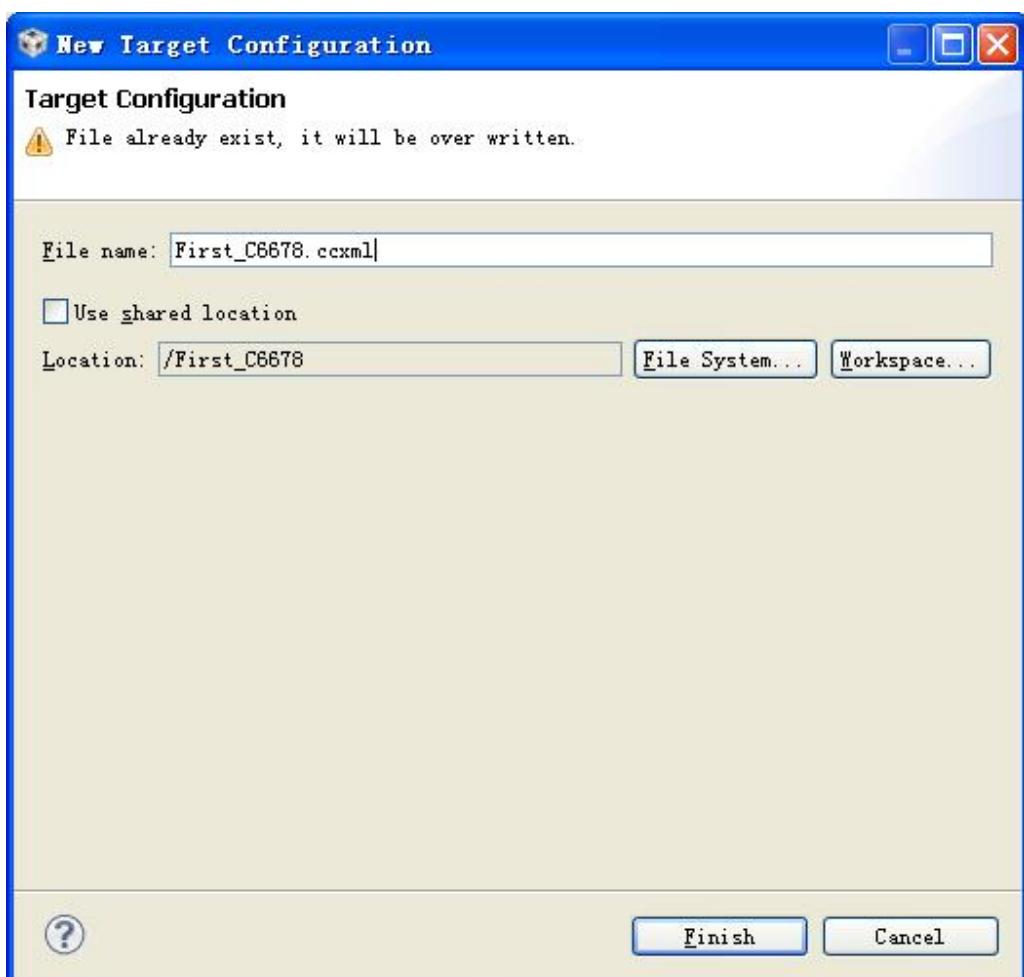
The screenshot shows the CCS V5 IDE interface. The main window displays a C source code file named 'main.c'. The code contains a function 'Debug' which prints the value of variable 'a' to the console. The 'main' function initializes 'a' to 0, increments it to 10, and then prints its value. The terminal window at the bottom shows the build log:

```
CCSV5 First_C6678
First_C6678.out
Linking
warning: creating ".stack" section with default size of 0x400; use the -sstack
option to change the default size
warning: creating ".bss" section with default size of 0x400; use the -sbss
option to change the default size
Finished building target: First_C6678.out
```

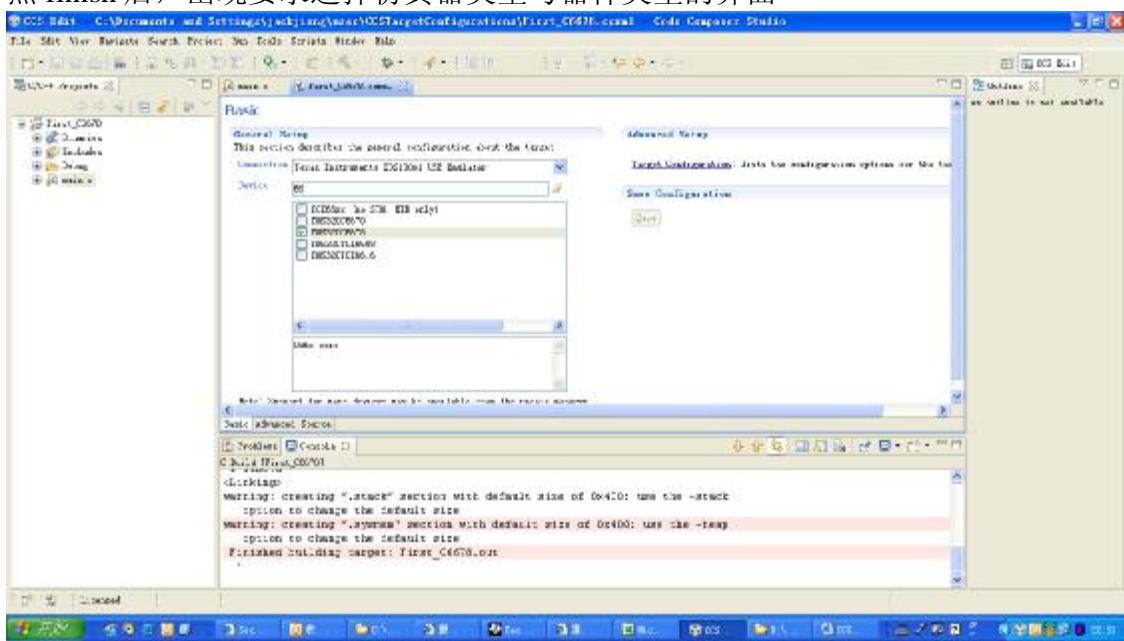
编译后，程序如果没有问题，最后在 console 中输出 'Finished building target:  
First\_C6678.out'

第五步，建立配置文件，在CCSV5中，配置文件的后缀名是点 ccxml

在左侧工程文件栏中，点击工程名称，再右键，new->target configuration  
file



点 finish 后，出现要求选择仿真器类型与器件类型的界面



选择对应的仿真器与器件型号后，点 save。再将位于屏幕中间位置的 advanced 选中，此处，是给程序要运行的 DSP 核配置 gel 文件。

C:\program files\Texas

Instruments\ccsv5\ccs\_base\_5.0.3.00023\emulation\boards\evmc6678l\gel\evmc6678l.gel  
此程序，我们只运行 0 核，给 0 核配置 gel 文件后，点 save

配置文件设置完毕，可以关闭该文件

**注意：**可能有网友会发现自己的目录下没有 evmc6678l 这个目录，请从配套光盘上拷贝

第六步，加入链接文件，拷贝配套程序中的 linker.cmd 到工程目录，然后编译，得到输出的点 out 文件

**注意：**如果不加入 linker.cmd 文件，工程依然可以通过编译，得到输出点 out 文件，但是这个文件在调试时 load 到 DSP 时，会报错误

C66xx\_0: Loader: One or more sections of your program falls into a memory region that is not writable. These regions will not actually be written to the target. Check your linker configuration and/or memory map.

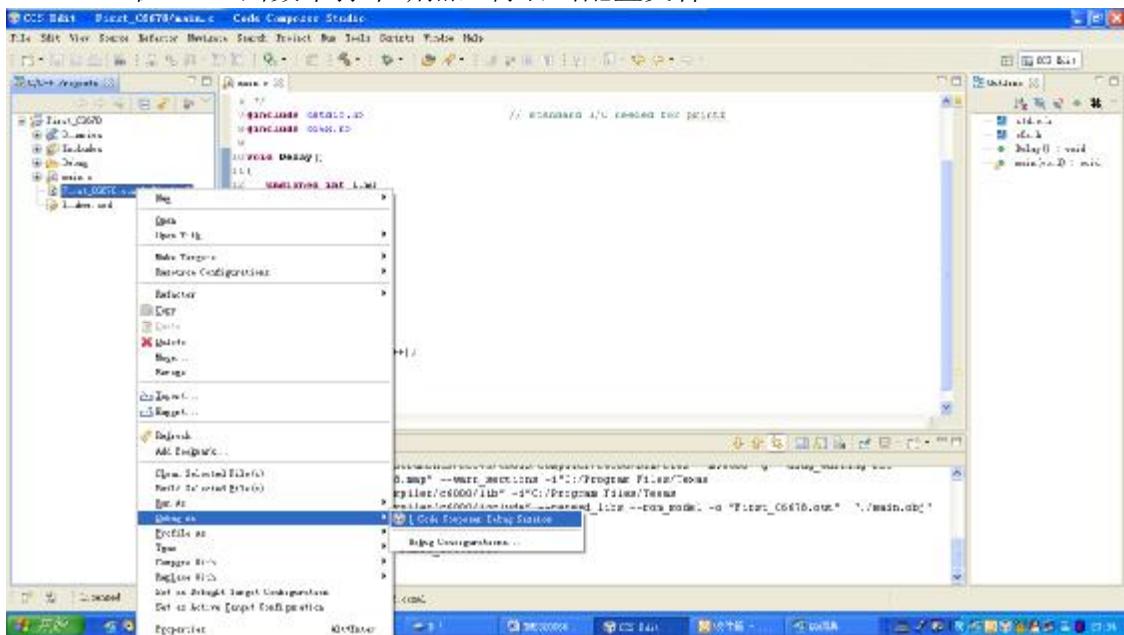
C66xx\_0: Trouble Writing Register PC: The IM memory request is not serviced by the HW memory system. This could be caused by the memory address specified does not exist in the specified memory/cache level. (Error -1176) @ 55712 (0xD9A0)

C66xx\_0: GEL Output: Disable all EDMA3 interrupts and events.

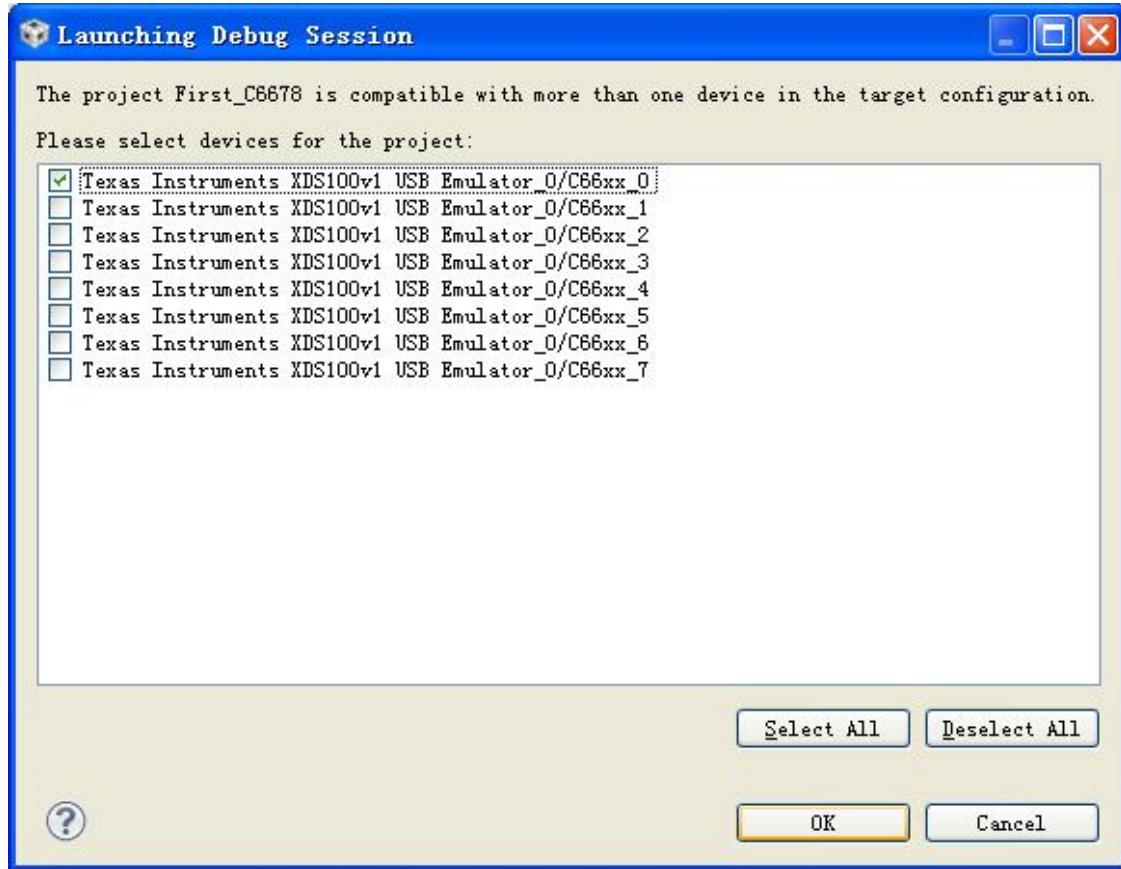
C66xx\_0: GEL: Error while executing OnRestart(0): Could not read the register DNUM: Execution state prevented access.

第七步，调试，连接好硬件，并上电

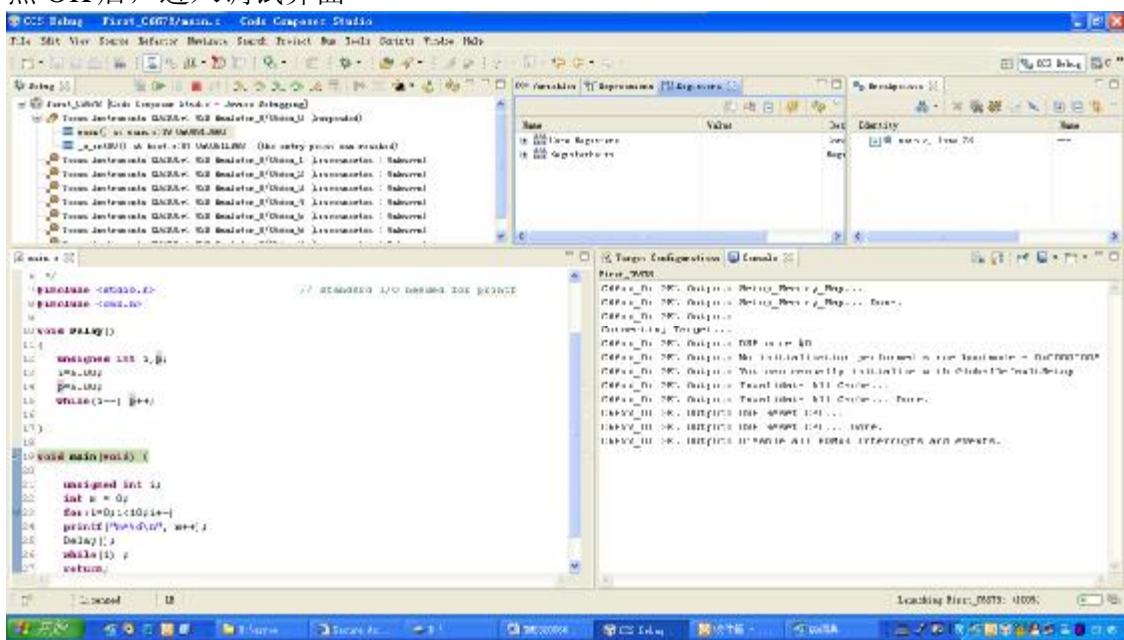
在 main 函数中打个断点，再右击配置文件



我们先只在 core0 上运行程序



点OK后，进入调试界面



正确进入调试界面后，你可以看到 console 的输出信息

左上角是 core 信息，中间是局部变量，全局变量，寄存器信息，右上角是断点信息，左下角是源代码文件。下图是调试控制按钮



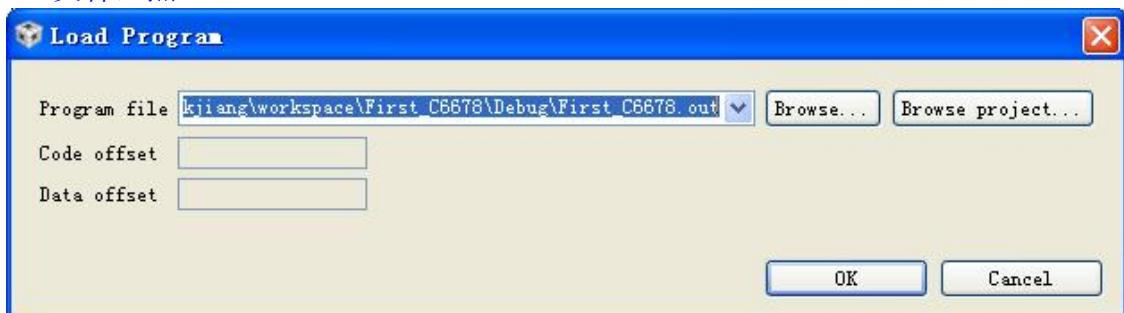
点击 resume 按钮，程序会运行到断点处

注意：此处我在调试时，程序并没有停止在断点处，而是直接运行完毕，断点没有起作用。遇到如此情况，请按以下操作

点暂停按钮，使 DSP 核停止运行，取消原行设置的断点

重新打一个断点，并在右上角的断点界面，钩上此断点

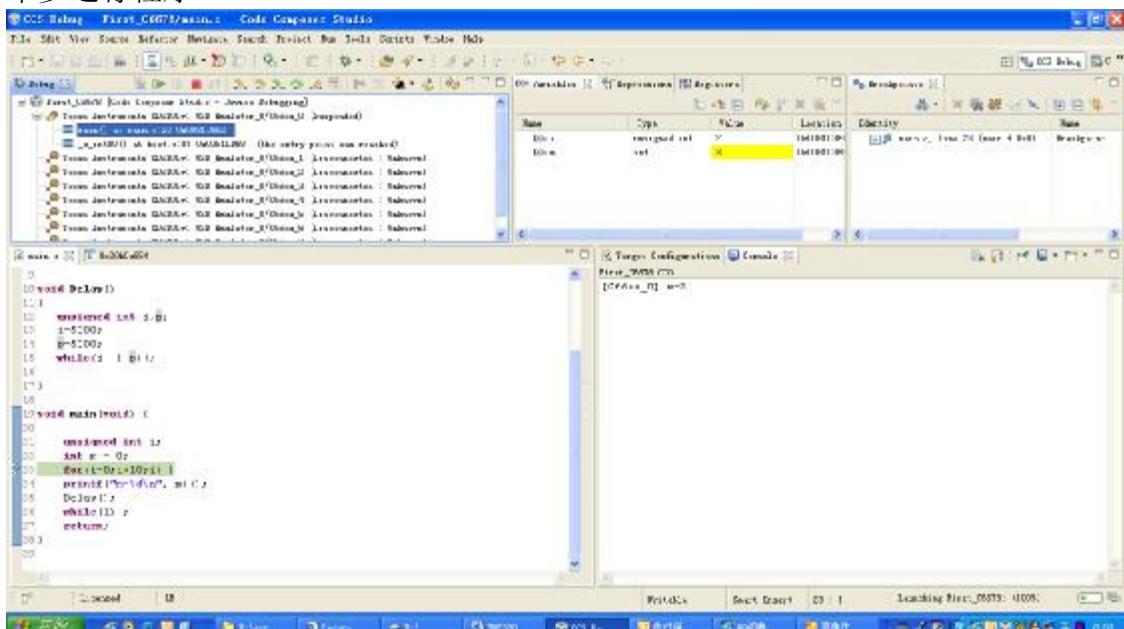
再点 cpu reset 按钮，再点界面上的 load 按钮，找到工程 debug 目录下的点 out 文件，点 OK



再点击 resume 按钮，程序会运行到断点处

可以 clear console 的输出信息

单步运行程序



可以看到，当输出 m=2 后，在局部变量的窗口中，已经执行 m++,m 的值为 3 了

## 继续单步执行程序

The screenshot shows the Code Composer Studio interface during a debug session. The assembly code window displays the following code:

```
3
10 void Delay()
11 {
12     unsigned int i;
13     i=5000;
14     i=5000;
15     while(i < 5000)
16     {
17         i++;
18     }
19 }
20 void main(void)
21 {
22     unsigned int i;
23     int m = 0;
24     for(i=0;i<10;i++)
25     {
26         printf("m=%d\n", m);
27         Delay();
28         m++;
29     }
30 }
```

The Registers window shows the value of the **i** register is 3. The Target Configuration window shows the processor is running at 12MHz.

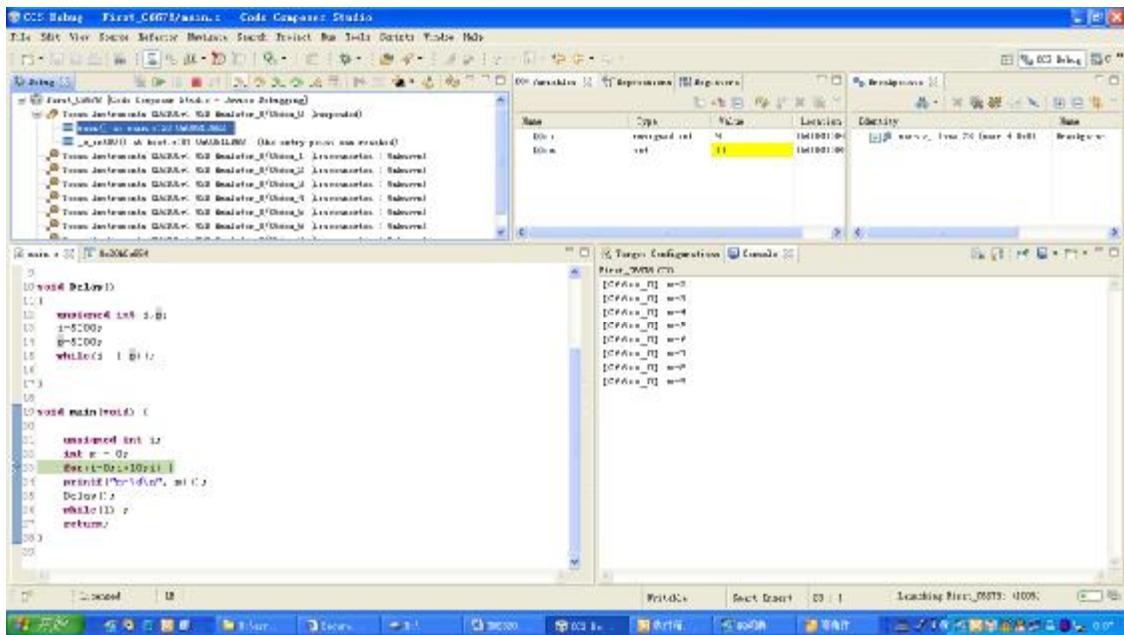
变量 i 的值变为 3,继续单步

The screenshot shows the Code Composer Studio interface during a debug session. The assembly code window displays the same code as before:

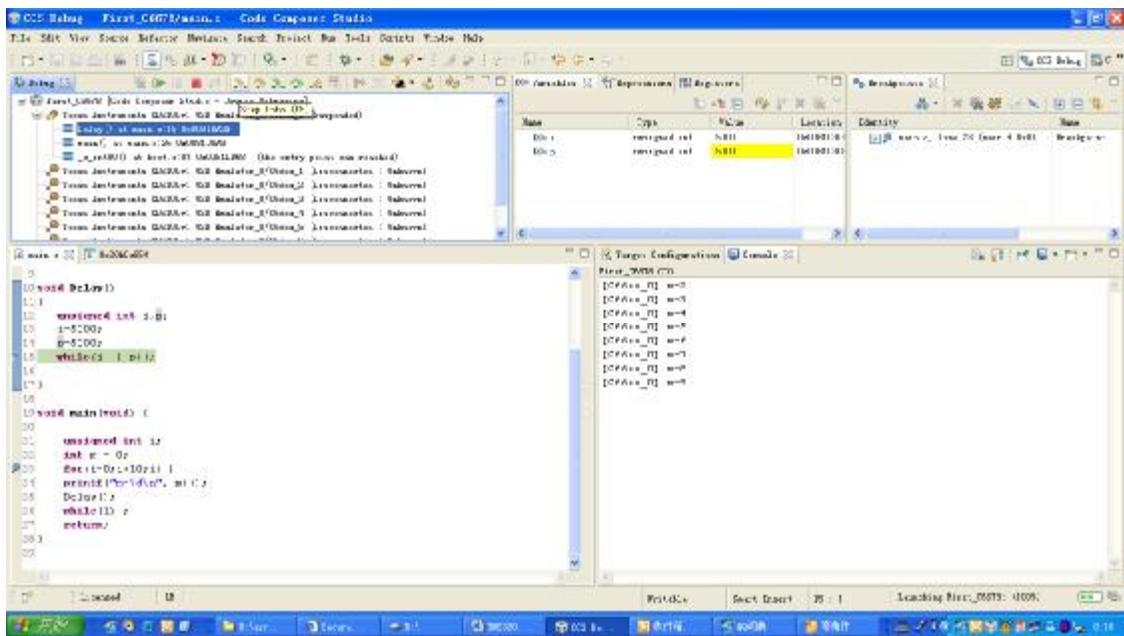
```
3
10 void Delay()
11 {
12     unsigned int i;
13     i=5000;
14     i=5000;
15     while(i < 5000)
16     {
17         i++;
18     }
19 }
20 void main(void)
21 {
22     unsigned int i;
23     int m = 0;
24     for(i=0;i<10;i++)
25     {
26         printf("m=%d\n", m);
27         Delay();
28         m++;
29     }
30 }
```

The Registers window shows the value of the **i** register is 4. The Target Configuration window shows the processor is running at 12MHz.

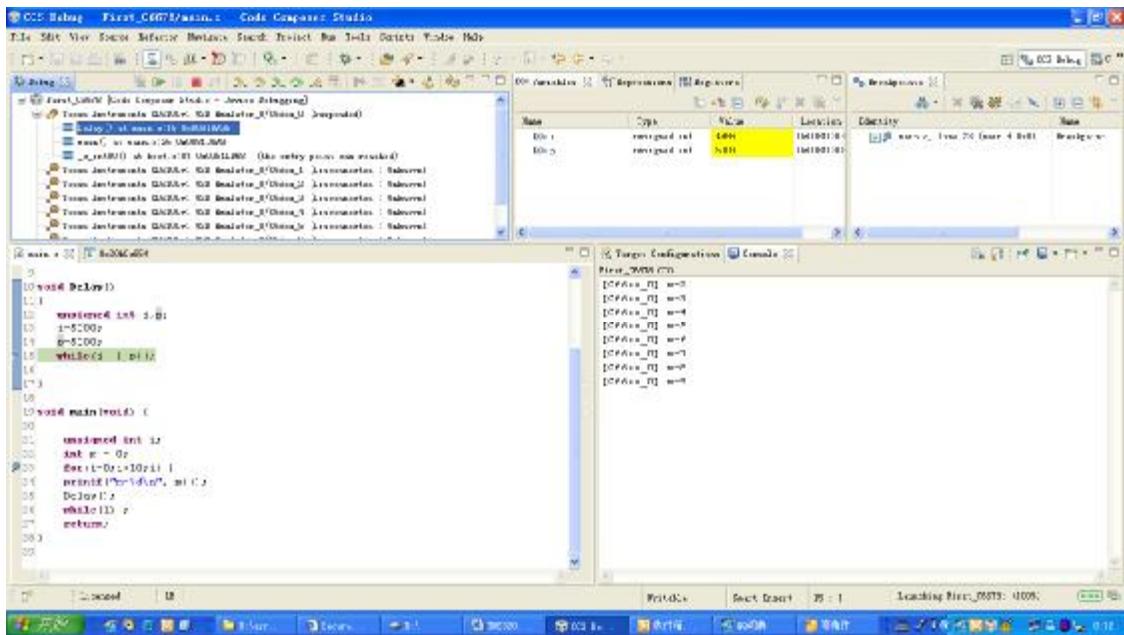
输出 m=3,m 的值为 4, 继续单步.....



当单步进入 Delay 子函数时，局部变量里的变量会自动更新为子函数中的局部变量



执行 i 与 p 赋值语句后的变量值，再继续单步，可以看到 i 与 p 的值变化

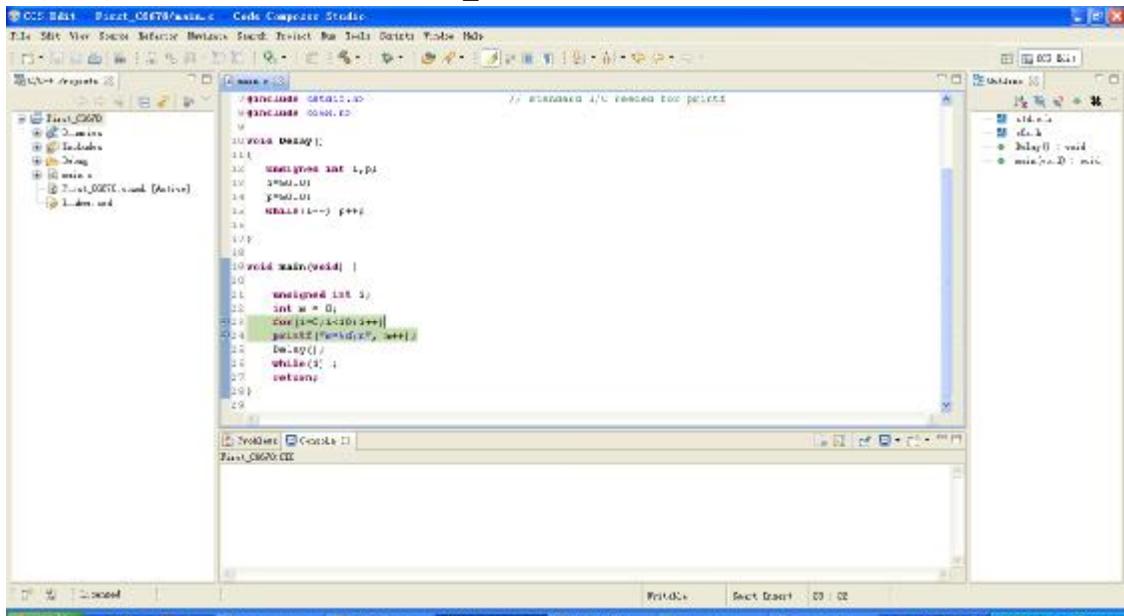


此处，发现 i 的值被多减了 1，此问题有待网友进一步验证

至此，单核调试笔记整理完毕，希望对初入门网友有用，不足之处，请见谅！

下面，我们继续如何进行在多个 DSP 中调试同一个程序。

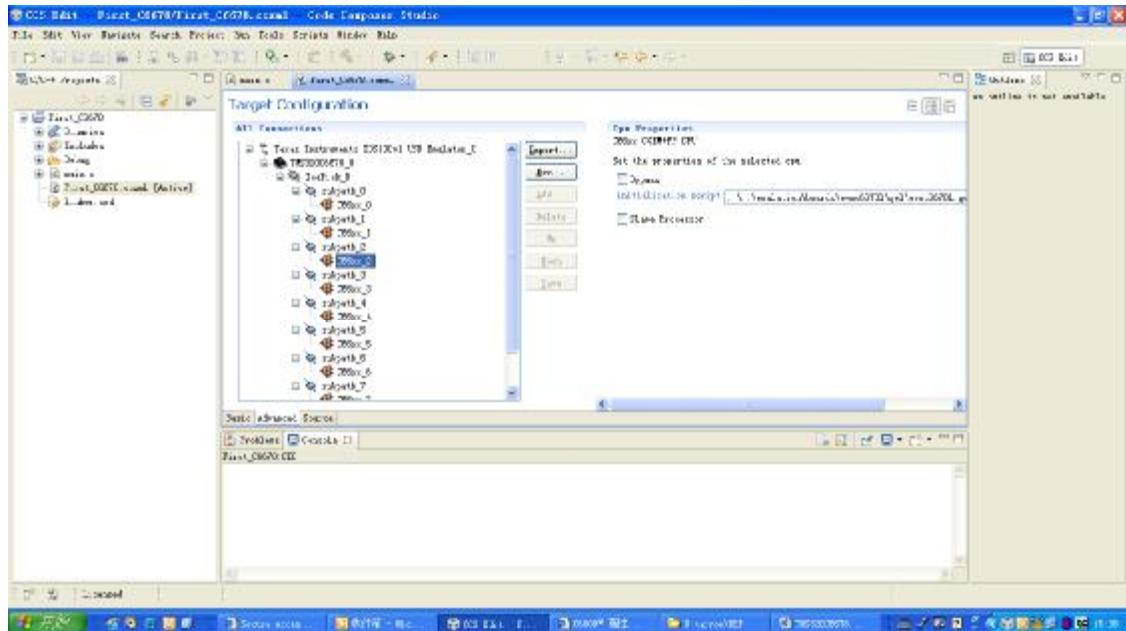
同样，在 CCSV5 中打开工程 First\_C6678



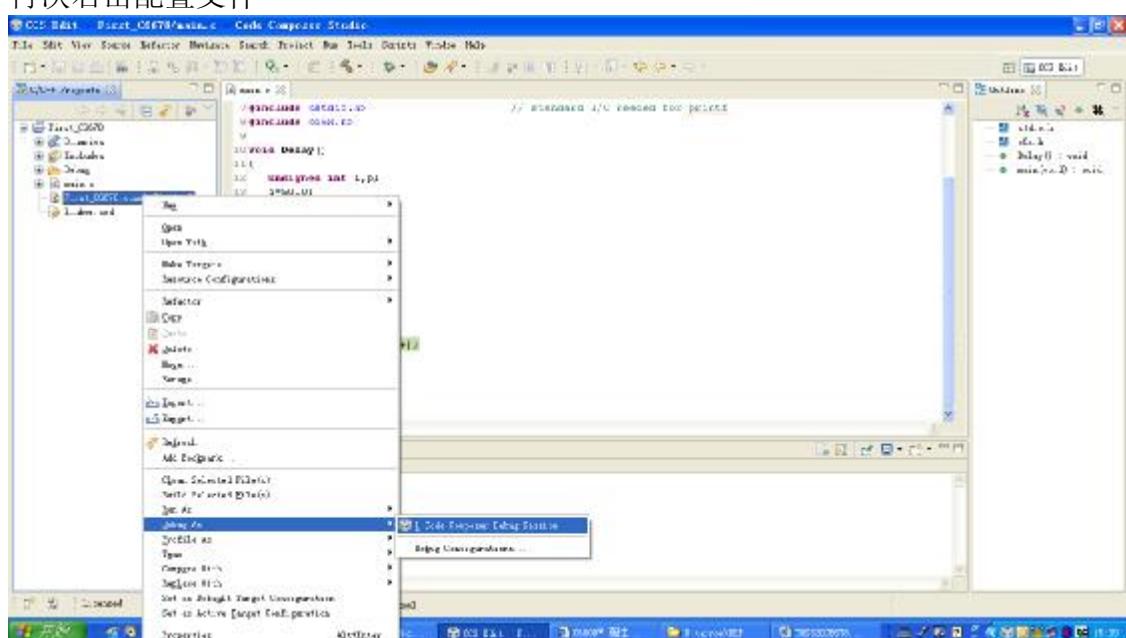
双击左侧工程栏中的配置文件 fisrt\_c6678.ccxml  
为每个核配置 gel 文件。

将位于屏幕中间位置的 advanced 选中，此处，是给程序要运行的 DSP 核配置 gel 文件。

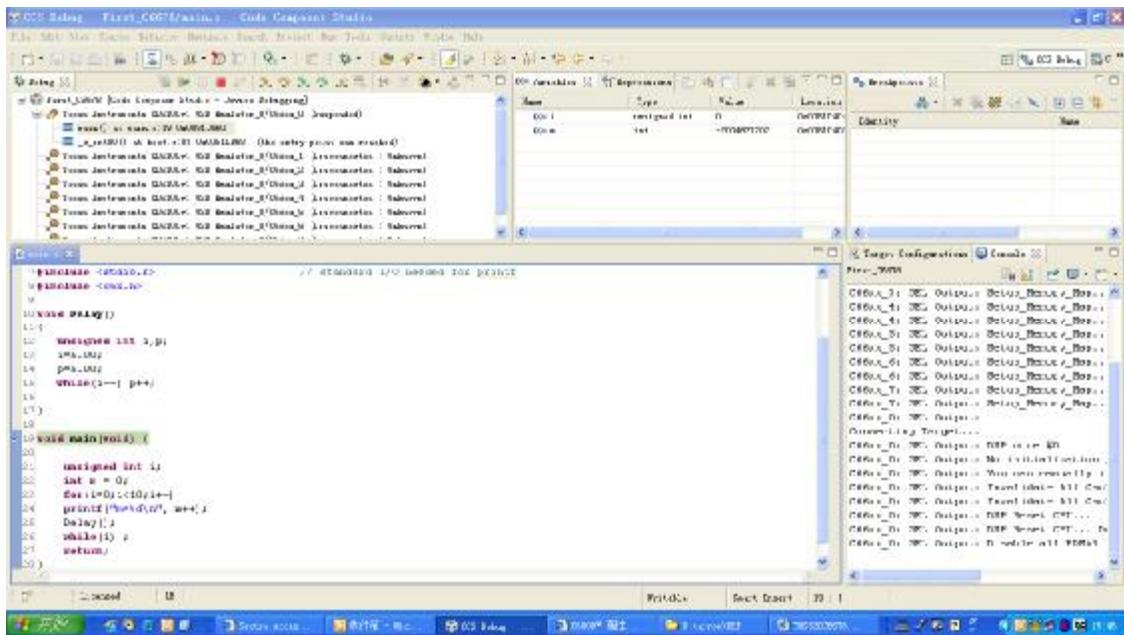
C:\program files\Texas  
Instruments\ccsv5\ccs\_base\_5.0.3.00023\emulation\boards\evmc6678l\gel\evmc6678l.gel  
配置完成后保存。



再次右击配置文件

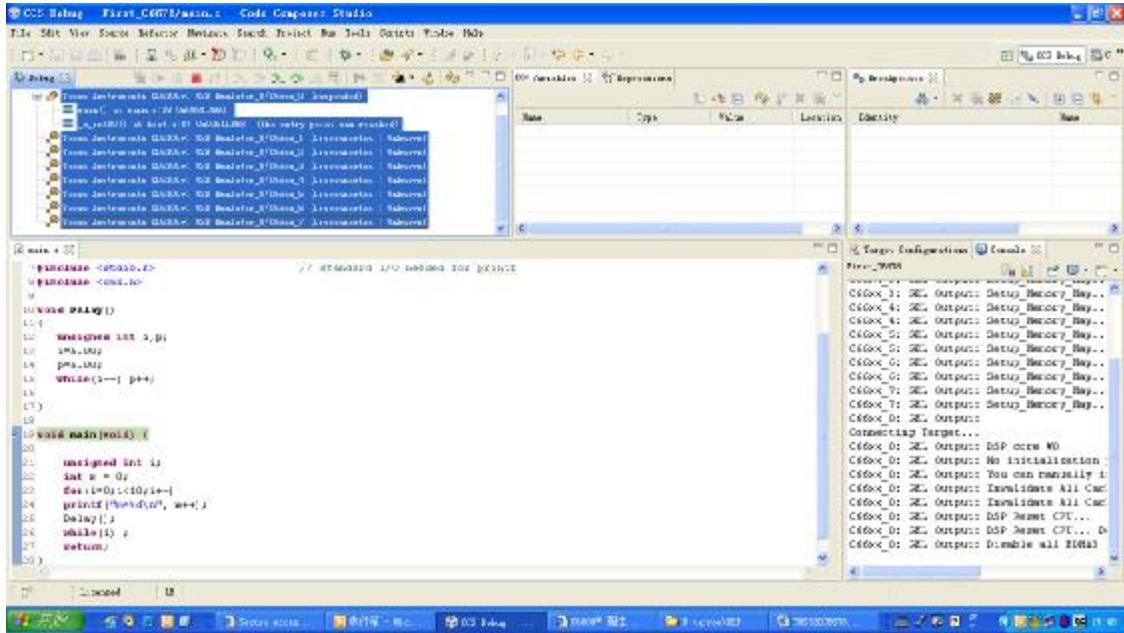


进入调试界面。

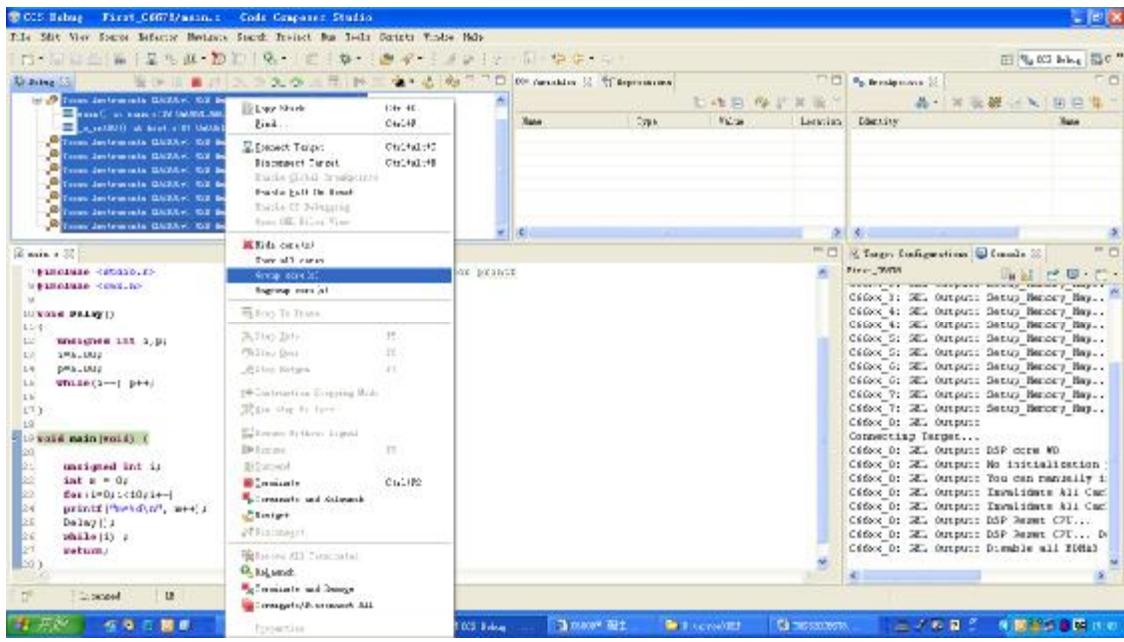


注意：使用 XDS100 仿真器的网友请稍有耐心，进入此调试界面的时间稍长

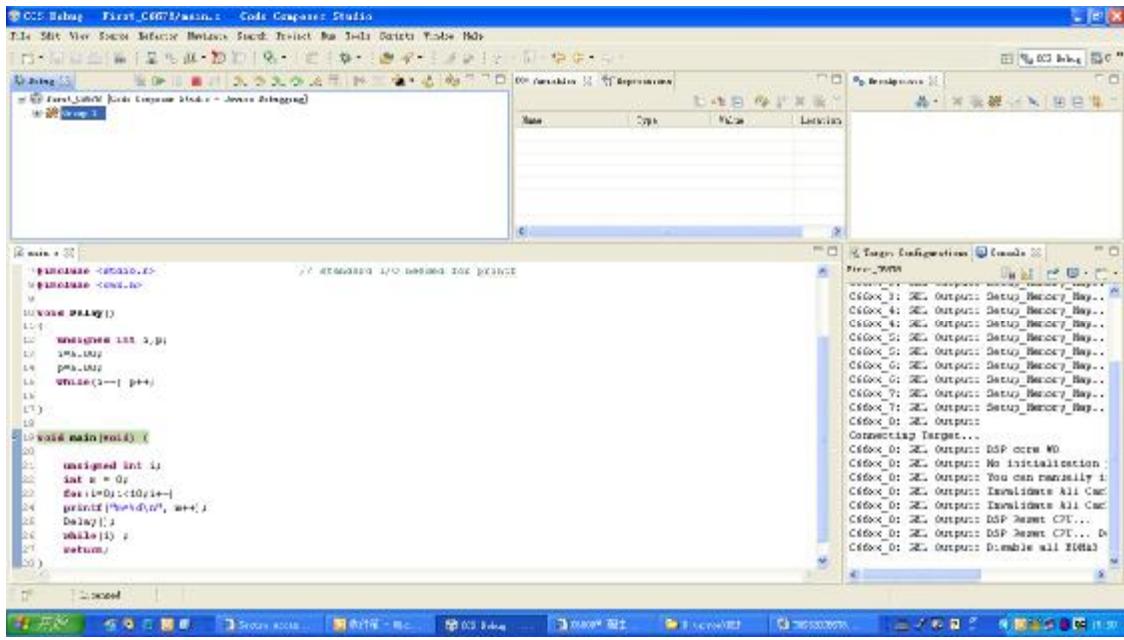
选中所有的核，选中方法为，按住 shift 键，单击第一个核，再单击最后一个核



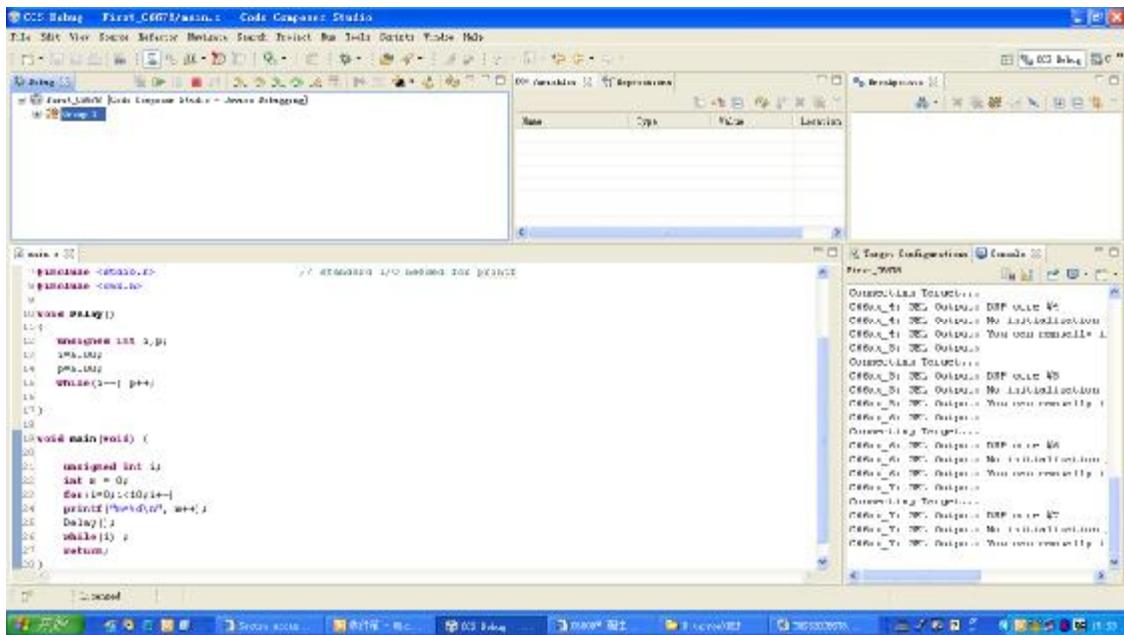
右键，选 group core(s)



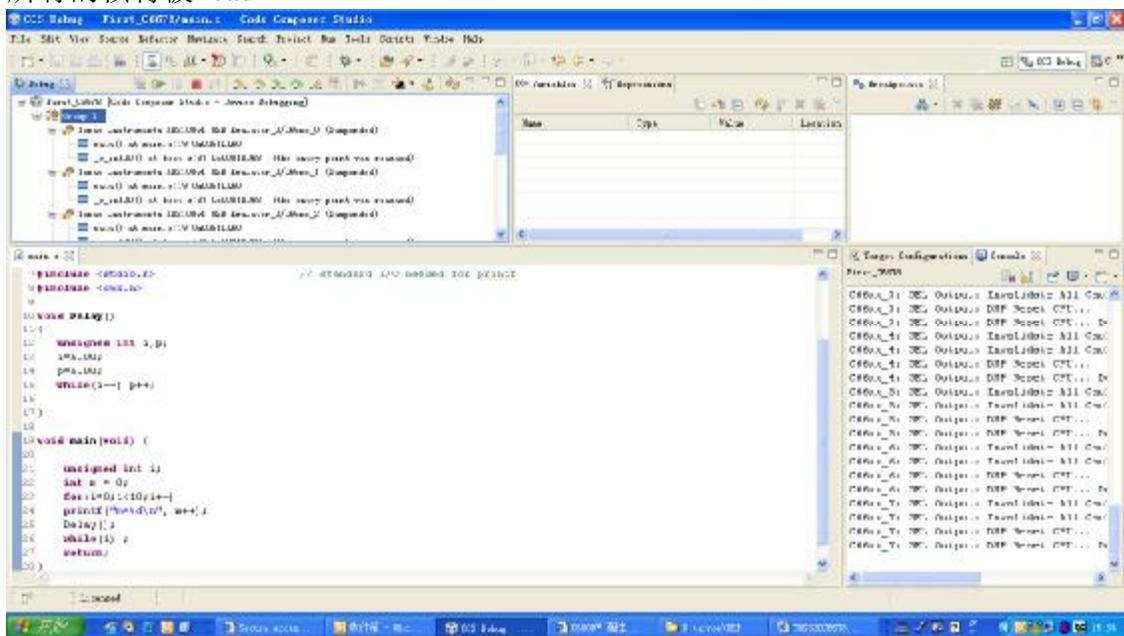
所有的核，成为一个组



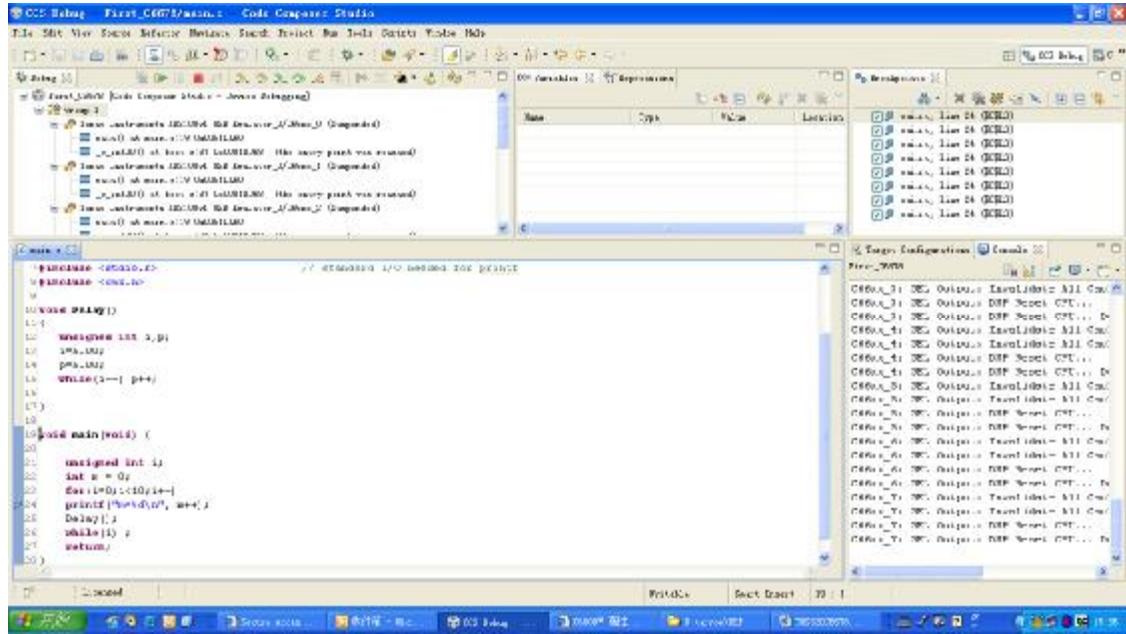
在 group1 上右键，点击 connect target, 仿真器将会连接所有的核



再选中 group1,点击 load 按钮, 选中编译好的点 out 文件, 确认所有的核将被 load



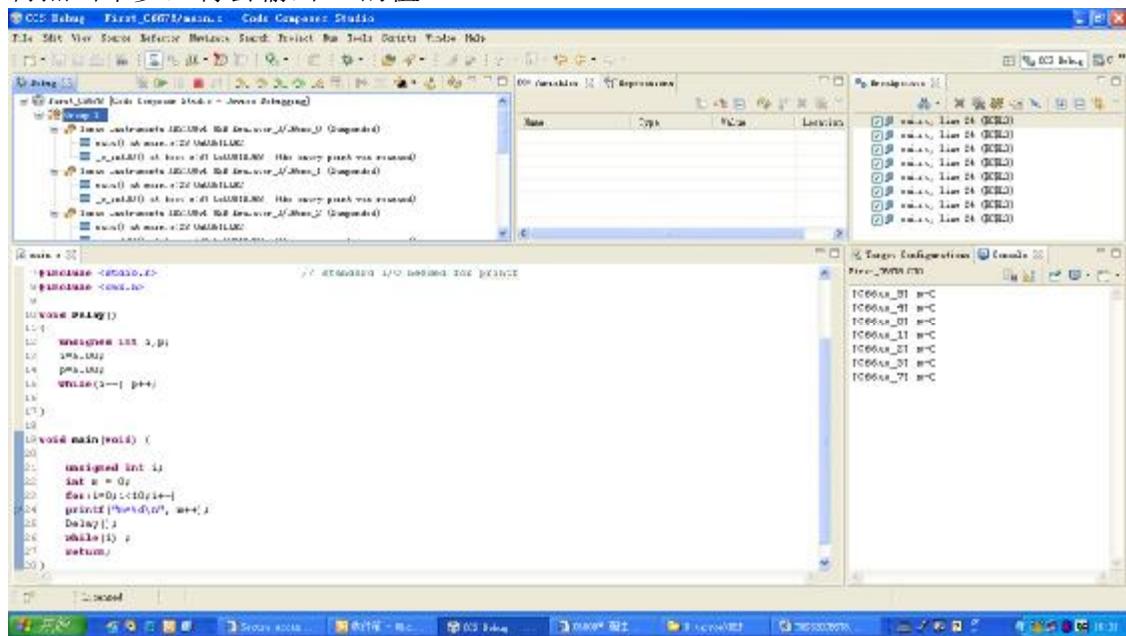
在程序第 24 行处, 打了一个断点, 此断点在右上角的断点窗口中, 会出现 8 个断点, 每个核一个



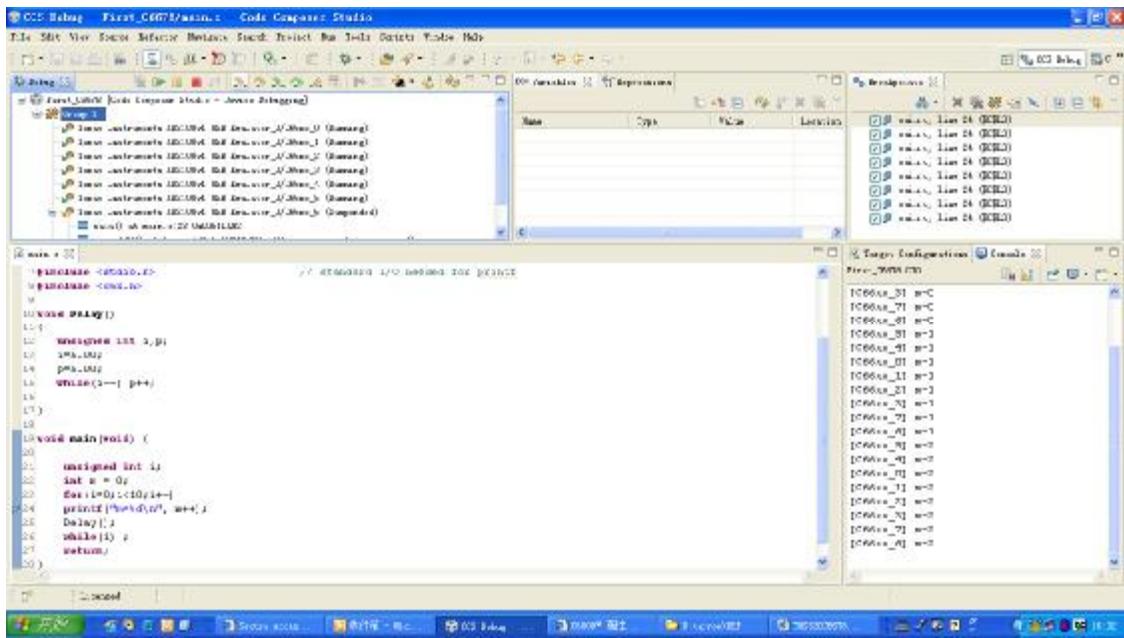
点 resume 按钮，程序将运行到断点处

注意，我调试到此处时，console 窗口出现了红色输出，意识好像是不能同时进行并行调试，一闪而过，我没有记的太完全，请网友注意

再点击单步，将会输出 m 的值



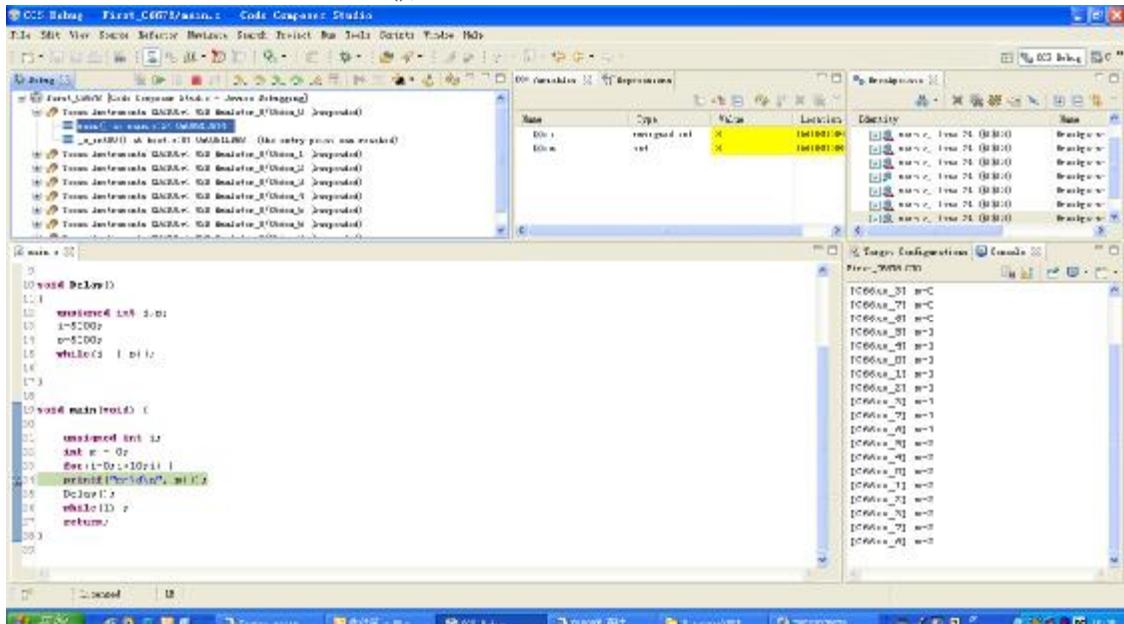
继续单步.....



注意，进行核的组合后，局部变量的值在局部变量窗口里是没有显示的

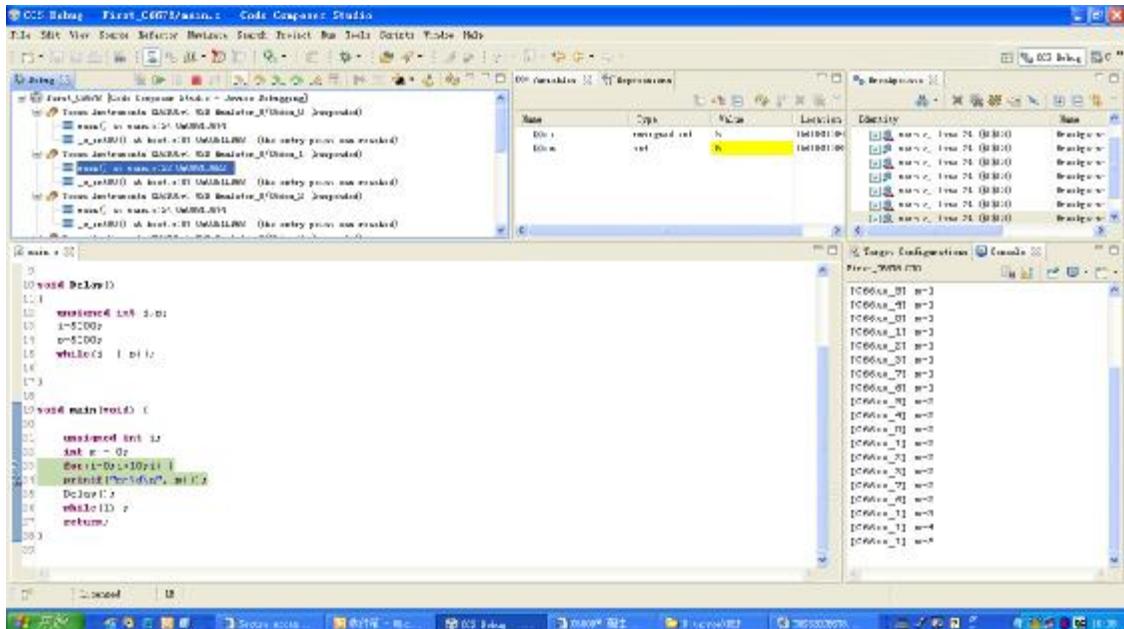
下面，我们可以对每个核进行单步，也可以看到每个核的局部变量的值在 group1 上右击，选 ungroup core(s)

分开后，点击每核下的 main(),该核的局部变量的值会显示在局部变量窗口中

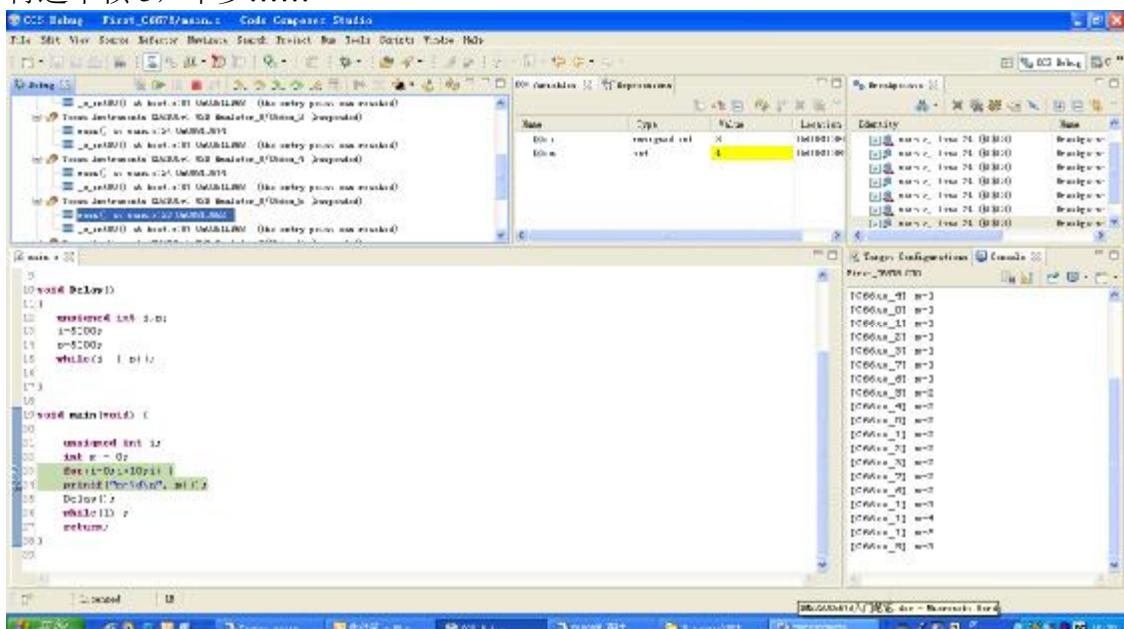


下面，我们对核 1 进行单步，选中核 1 的 main(),点击单步按钮.....

在 console 窗口中，我们看到了核 1 的连续输出，其他核局部变量的值没有发生变化



再选中核 5，单步.....



Console 窗口有核 5 的输出信息

至此，多核调试同一个程序笔记完毕。不足之处，请见谅！