如何使用 CCS v5 调试 DM81XX 的 M3 代码

在调试 M3 的代码时，用户需要实时查看参数值。如果在 IPNC RDK 串口终端上直接打印输出参数值，每增加一个参数就需要添加打印代码，还需要重新编译下载，麻烦而且效率低。CCS 提供的调试功能可以方便的查看参数值，随时可以增加参数，还可以在系统运行时修改参数值，方便高效。本文以 DM8127 VPSS M3 为例介绍如何用 CCS 对 M3 进行调试。

1. 调试环境

1.1 软件

- CCS v5.5

  目前最新的版本是基于 Eclipse IDE 架构的 CCS v5.5。
  下载地址 http://processors.wiki.ti.com/index.php/Download_CCS。

- IPNC RDK Software

  较新的版本是 V3.8。

- 串口软件

  例如：Tera Term 终端。

1.2 硬件

- 仿真器（emulator）
  DM8127 可以配合使用的仿真器有 XDS560v2 、 XDS510USB 等，这里使用的是 Spectrum digital XDS560v2。

- DM8127 IPNC 参考设计 硬件平台

1.3 其他参考资料

- IPNC_RDK_InstallGuide.pdf
- HDVPSS DRIVER USER GUIDE
- http://processors.wiki.ti.com/index.php/Category:CCS_Training

2. 调试方法

2.1 调试思路

在调试 VPSS M3 时，用户不能简单的通过 CCS 下载 VPSS M3 的代码进行调试，而需要将整个 DM8127 软件运行起来，通过 A8 上运行的 Linux 下载 VPSS M3 代码。这样用户不能在程序跑起来之前在 CCS 下设置断点，但程序跑起来连上 CCS 后设置断点，这时 M3 上需要调试的代码可能已经运行结束了，也就是程序可能已经运行到断点之后了。

因此要调试 M3 代码，可以在要调试的代码之前添加一个循环，循环条件由一个全局变量控制，在未连上 CCS 之前使程序在此处做循环，连接上 CCS 设置好断点后，再通过 CCS 修改全局变量的值使程序跳出循环，往下运行到断点。

2.2 添加循环代码

在 VPSS M3 的代码中，添加一个全局变量例如：int label_alice，并在需要调试的代码前添加循环。添加位置根据自己调试的代码决定，例如可以加在 IPNC_RDK_INSTALL_DIR/Source/ipnc_rdk/ipnc_mcfw/mcfw/src_bios6/links_m3vpss/sclr/ sclrLink_tsk.c 文件的 SclrLink_init()函数中。

```
label_alice=0;
while(label_alice==0){
                Vps_printf("label_alice\n");//也可以将这打印语句去掉；
                        }
```

这样可以使 VPSS M3 在需要调试的代码之前做循环打印，连上 CCS 后，通过 View->Expressions 中修改 label_alice 的值为 1，使其跳出循环，继续往下运行，从而实现对所需代码的调试。

注意，你需要将上面的代码添加到 src_bios6/下面的某个文件中，而不是 src_linux/下的文件中。因为我们需要调试的是运行 BIOS 的 VPSS M3 上的代码，需要将代码添加到 VPSS M3 对应的代码中。

2.3 编译运行：

1) 修改代码后保存。

2) 打开 Rules.make 文件（在 IPNC_RDK_INSTALL_DIR/Source/ipnc_rdk/目录中）。选择 DM8127、9J003 sensor（可根据用户选用的硬件平台和 sensor 更改）和 DEBUG 模式 。具体选择如下：
   在# Build configuration 中选 APP_BUILD_CFG := debug。
   在# Choose the target platform 中选 IPNC_DEVICE := DM8127。
   在 IMGS_ID 中选 IMGS_ID :=IMGS_MICRON_MT9J003。

3) 编译后下载到板子上使其成功跑起来，这时用户可从串口终端看到程序运行在循环语句中。

3. 调试步骤

3.1 连接 CCS 下载 symbol

3.1.1 硬件连接

连接仿真器和 IPNC 参考设计硬件：

注意：JTAG pin 脚的顺序，不要把 JTAG 接口插反了。Spectrum digital XDS560v2 仿真器需要提供 5V 电源。

3.1.2 打开 CCS

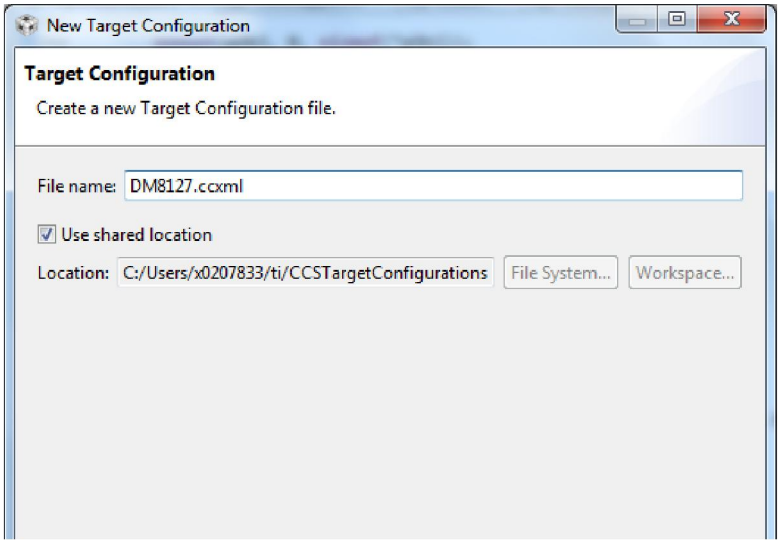
1) 点击 Start -> All Programs ->Texas Instruments-> Code Composer Studio v5.0.1-> Code Composer Studio v5。

2) 选择一个工作目录“Select a Workspace”，选择好目录后点击 OK 即可。

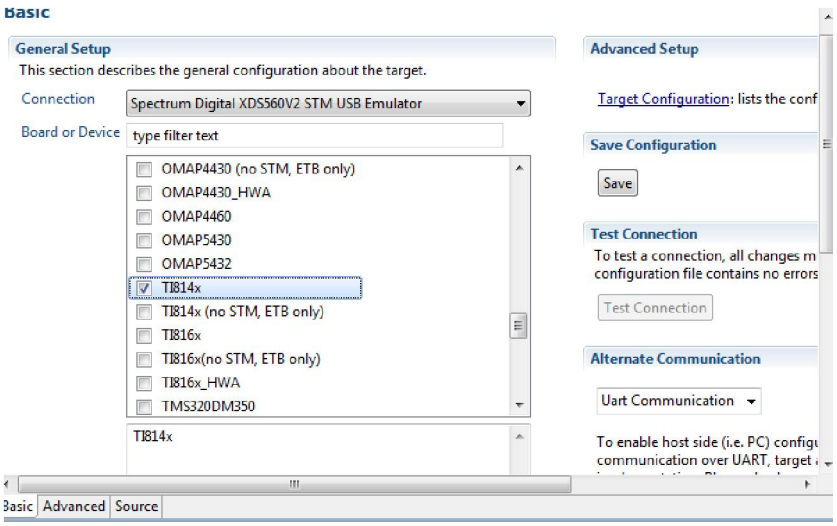
3.1.3 配置 target configuration

1) 新建一个 Target Configurations。
   方法 1：点击 View->Target Configurations，在出现的 Target Configurations 窗口处点击鼠标右键，在出现的菜单中选择 New Target Configuration。
   方法 2：点击 File->New->Target Configuration File 来新建一个 Configuration target

2) 新建一个名为 DM8127 的 target。



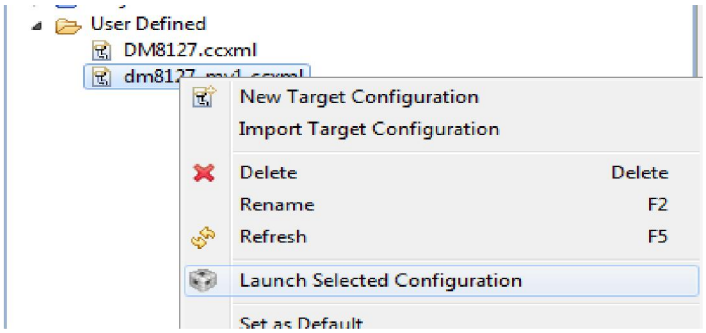
3) Connection 中选择 Spectrum Digital XDS560V2 STM USB Emulator。在 Board or Device 中选择 TI814x，点击 Save 保存。（可点击 Test Connection 看是否能够正常连接。）



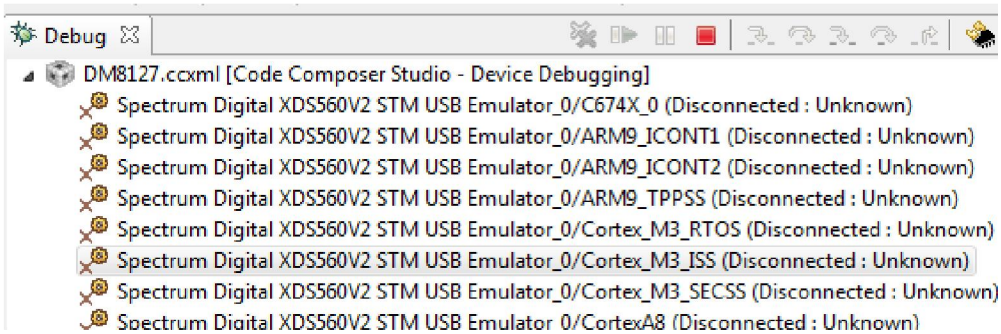
3.1.4 连接 VPSS M3

1) **Launch target configuration**

选中 Target Configurations 窗口中已配置好的 target（DM8127.ccxml）， 在右键菜单中选择 Launch Selected Configuration。

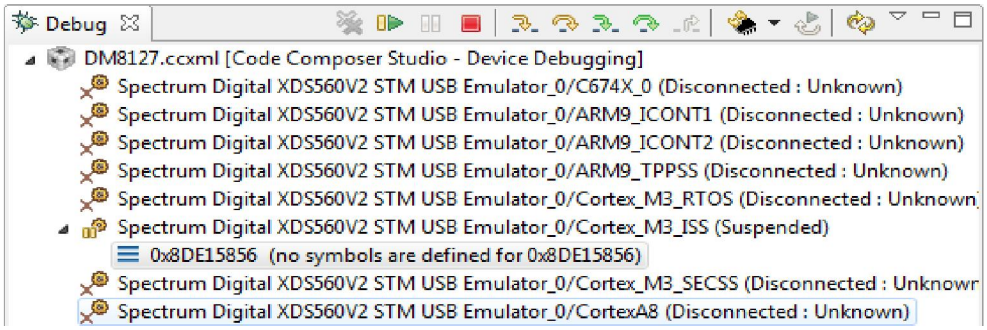


launch 成功后，可以看到下图。此时，PC 和仿真器通信连接成功，但是 VPSS M3 还没有连上。



2) **连接 VPSS M3**

在 debug 窗口中，右键点击 M3_ISS，选择 Connect Target。连接成功后如图所示。



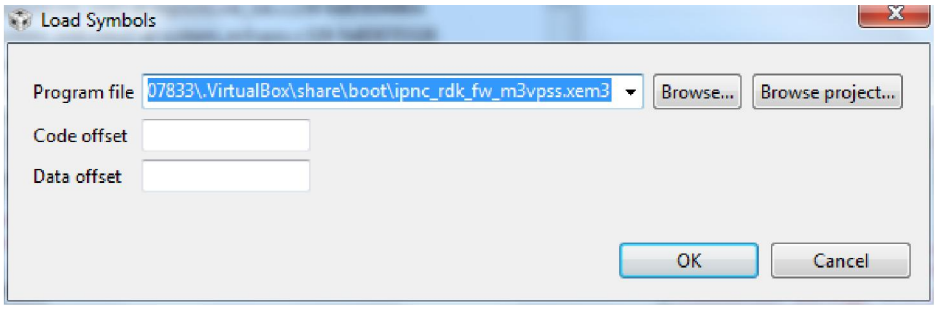
注：1.CCS 里面的 M3_ISS 就是 VPSS M3。

2.连上仿真器后，M3 会自动停止运行。

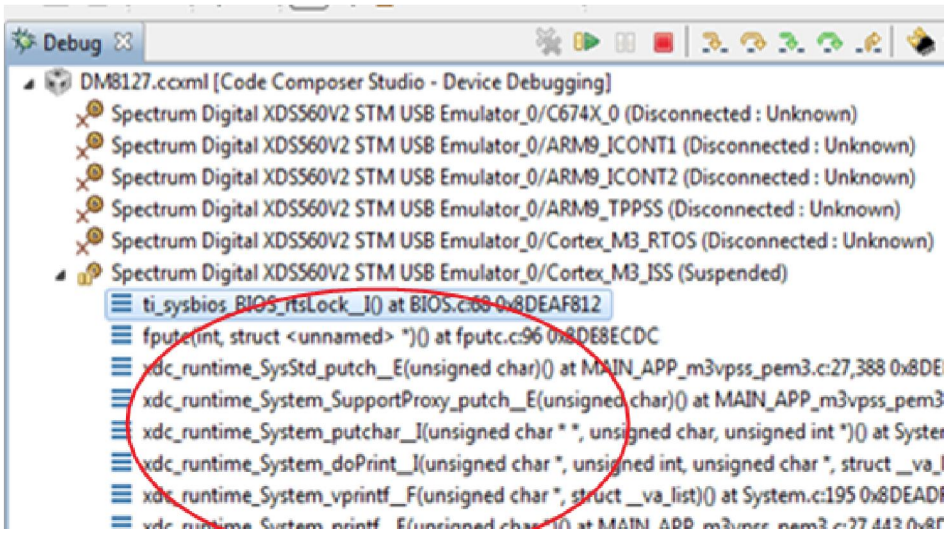
3.1.5 下载 VPSS M3 symbol

在菜单中选择 Run -> Load -> Load Symbols 加载 symbol 信息，选择 ipnc_rdk_fw_m3vpss.xem3 文件。点击 OK 即可。

注：编译完 IPNC RDK 后，VPSS M3 的可执行文件 ipnc_rdk_fw_m3vpss.xem3 文件在$ IPNC_RDK_ INSTALL_DIR /Source/ipnc_rdk/target/filesys/opt/ipnc/firmware 中。



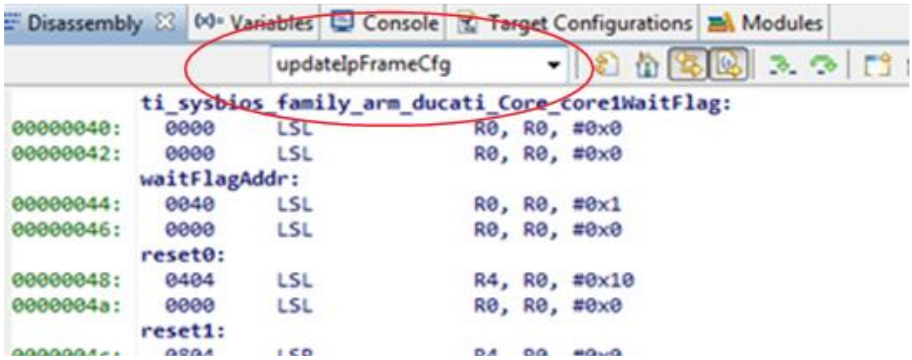
下载 symbol 成功后如下图：



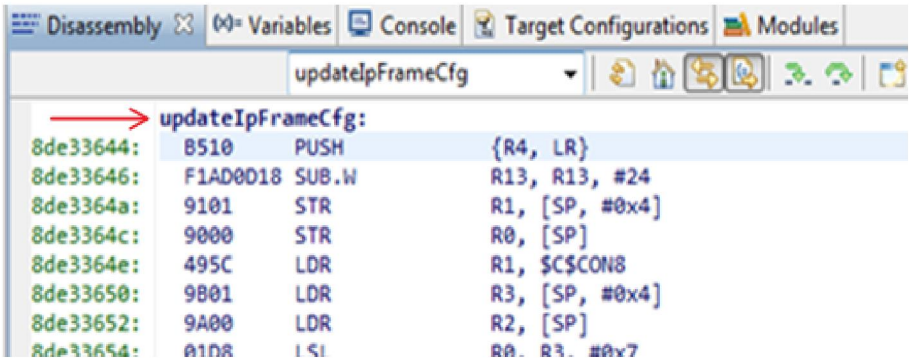
注：上图标记的地方显示的是堆栈。列出的函数中，下面的函数调用了上面的函数，例如 xdc_runtime_System_vprintf_F()调用了函数 xdc_runtime_System_doPrint_I(),而 xdc_runtime_System_doPrint_I()调用函数 xdc_runtime_System_ putchar_I()，依次类推。通过此处堆栈，可以方便的查看程序运行过程中函数调用情况。

3.2 设置断点

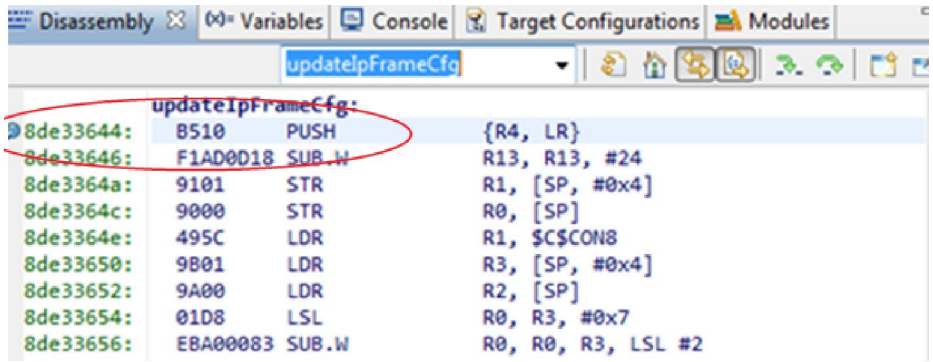
1) 在 disassembly 窗口添加断点所在函数。例如：断点所在函数是 updateIpFrameCfg，在如下图圈出的窗口上键入函数名。



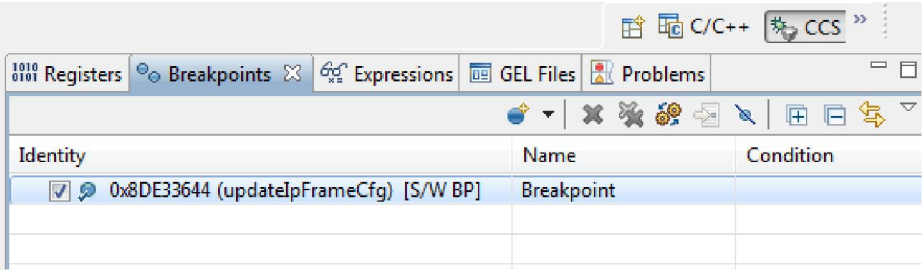
回车后找到函数所在处：



2) 在需要设断点的地址左侧双击，即可添加断点。



3) 添加的断点会在 View->Breakpoint 中显示：

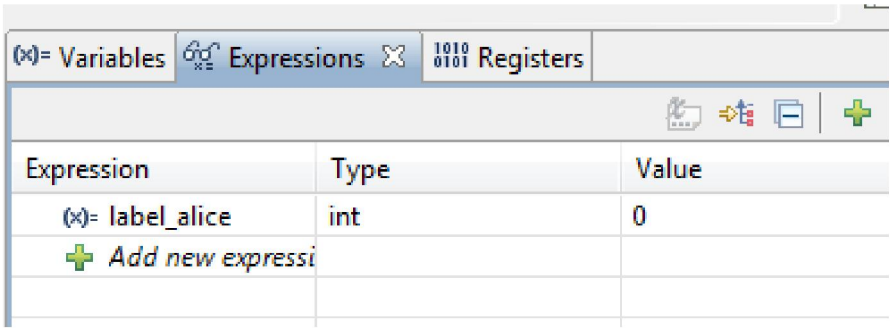


3.3 使程序停在断点处并开始单步调试

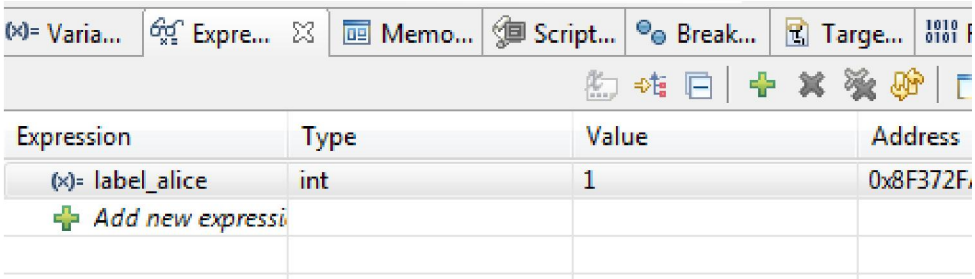
3.3.1 修改全局变量的值，使程序跳出循环

1) 要使程序跳出之前设置的循环，继续往下运行，需要修改全局变量 label_alice 的值，使其为非零。如下所示。

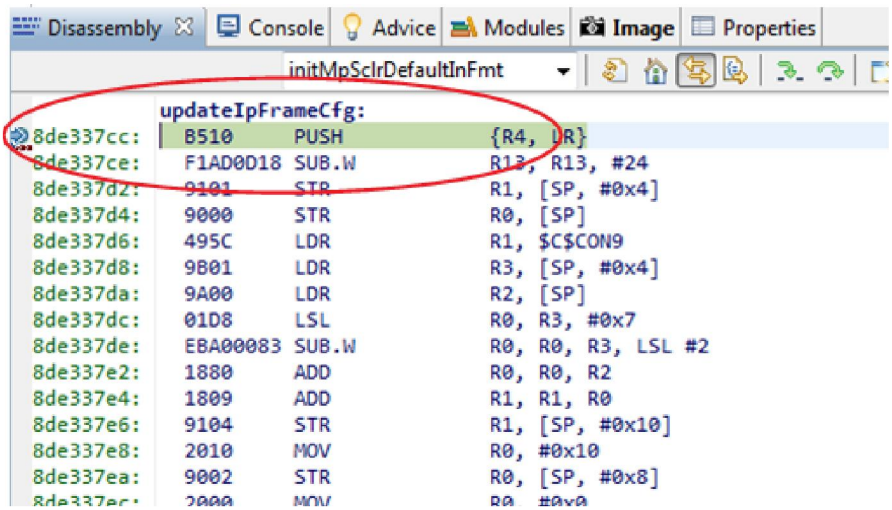
在 View->Expressions 处添加 label_alice。显示目前 label_alice 的值是 0 。



将 label_alice 的值改为 1。



2) 这时点击运行按钮（或者按 F8），程序就会运行到断点。

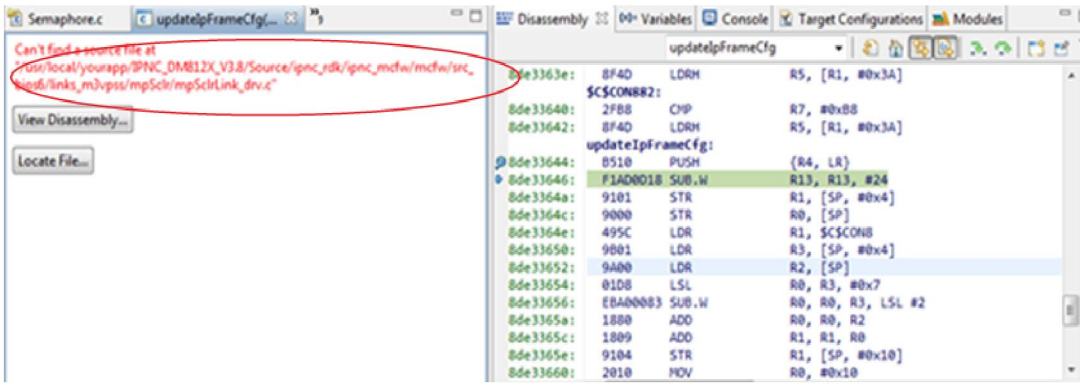


3.3.2 查看源码

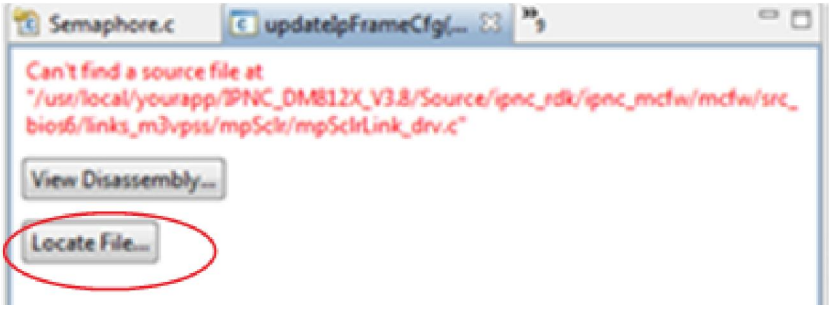
有两种查看源码的方法：

1) 在源码窗口显示

在程序运行到断点时，左边窗口出现错误提示：找不到文件，如下图。

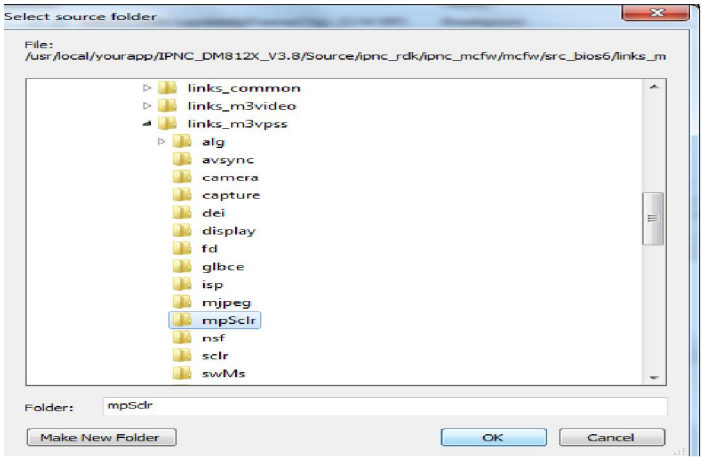


点击 Locate File。

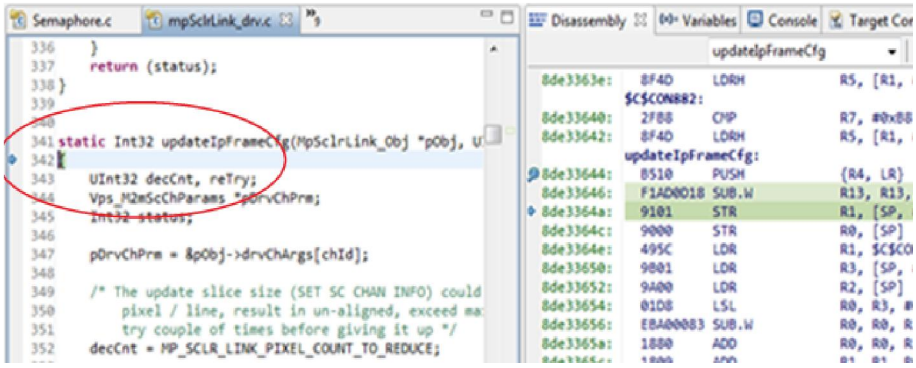


找到提示中显示的需要添加的 C 文件源代码所在位置，添加上去（可以提前

将 M3 VPSS 的源代码从 Linux 拷到 Window 下），CCS 会很据这个文件的目录找到其他的 C 文件，这样在同一个目录下的源码文件都被添加到 CCS 中。

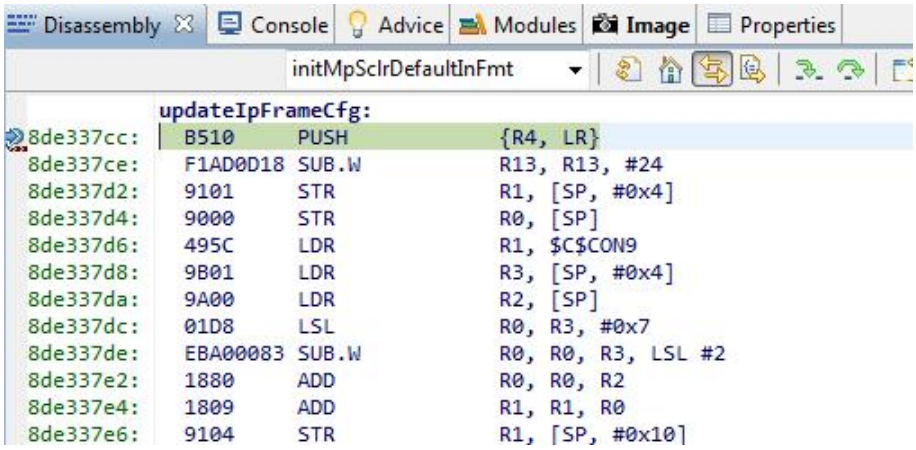


这样便可以显示源码，如下图所示：

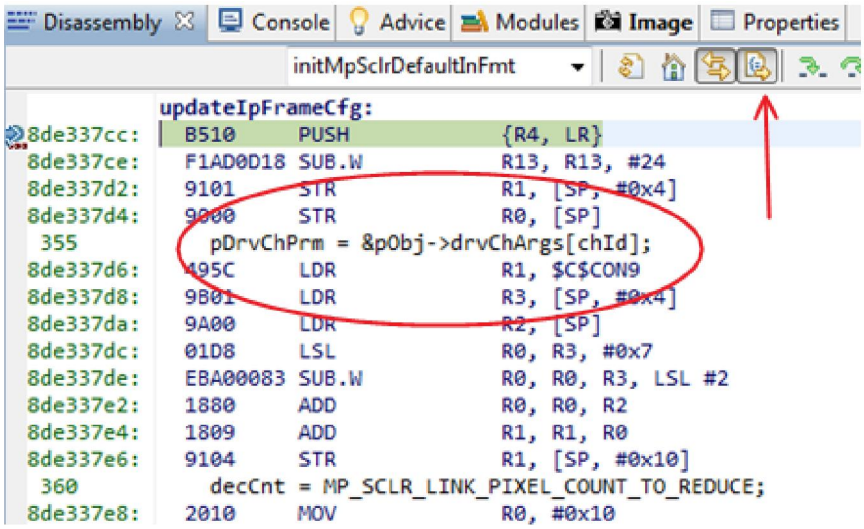


2) 汇编窗口中显示源码

汇编中没有显示源码之前：

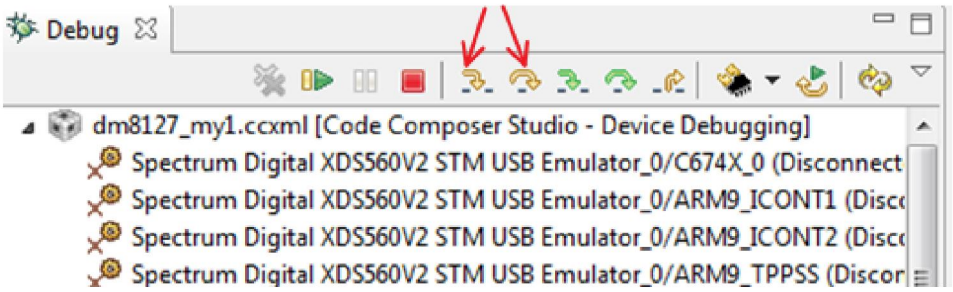


点击左起第四个按钮 show source ，显示源码之后：



3.3.3 单步调试

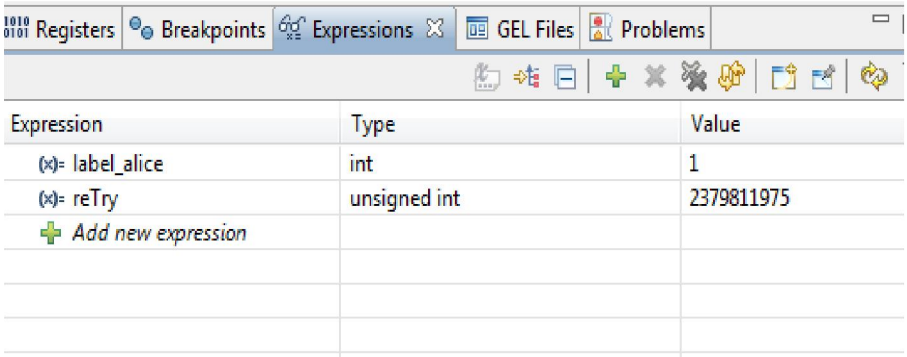
点击下一步（箭头所示，或者使用快捷键 F5 或 F6），即可实现单步调试。



4. 查看参数的方法

1) 在 Expressions 窗口中添加你需要查看的变量。

   点击 Add new expression 添加变量，回车即显示变量值。



2) 将鼠标放在汇编对应的源码变量处，CCS 会自动显示该变量的值。

```
345    Int32 status;
346
347    pDrvChPrm = &pObj->drvChArgs[chId];
348
349    /* The update slice size (SET SC CHAN INFO) could fail, if the no of
350        pixel / line, result in un-aligned, exceed max, etc...
351        try couple of times before giving it up */
352    decCnt = MP_SCLR_LINK_PIXEL_COUNT_TO_REDUCE;
353
354    for (reTry = 0; reTry < MP_SCLR_LINK_SLICE_SIZE_MAX_RE_TRY; reTry++)
355    {
356        /* Configure
357        status = FVI
358
359
360        if (status =
361        {
```

updateIpFra
8de3363e:
8de33640:
8de33642:
8de33644:
8de33646:
8de3364a:

Expression	Type	Value
(x)= reTry	unsigned int	2379811975

3) 查看 View->Variables 窗口，该窗口将所有可用的变量值都显示出来。

Disassembly | (x)= Variables | Memory Browser

Name	Type	Value
slcData	struct <u...	0x8E52BE00
*(slcData)	struct <u...	{...}
(x)= subFrameMdEnable	unsigned ...	1
(x)= slcSz	unsigned ...	1060
(x)= noOfSlcs	unsigned ...	3
(x)= isEnable	unsigned ...	1
(x)= srcFrmH	unsigned ...	2736
(x)= tarFrmH	unsigned ...	1080
(x)= srcFrmW	unsigned ...	3648
(x)= tarFrmW	unsigned ...	1920
(x)= subFrameNum	unsigned ...	0
(x)= slcLnsAvailable	unsigned ...	1060

5. 总结

对于 M3 的调试，关键的一步是在需调试程序之前添加由全局变量控制的循环代码，使连接上 CCS 后程序停在需调试代码之前，然后通过修改全局变量的值使程序继续往下运行到需要调试的代码处。

本文以 VPSS M3 为例调试，本文的调试思想也适用于其他的 M3 模块，以及 DSP 的调试。