



广州创龙电子科技有限公司

Guangzhou Tronlong Electronic Technology Co.,Ltd

DSP+ARM+FPGA

三核主板方案领导者

OMAPL138 双核通信之 DSP 程序独立仿真教程

Revision History

Revision No.	Description	Draft Date	Remark
V1. 0	1. 初始版本	2014/4/7	

销售邮箱: sales@tronlong.com

技术邮箱: support@tronlong.com

公司总机: 020-8998-6280

公司官网: www.tronlong.com 1/19



广州创龙电子科技有限公司

Guangzhou Tronlong Electronic Technology Co.,Ltd

DSP+ARM+FPGA

三核主板方案领导者

阅前须知

版权声明

广州创龙电子科技有限公司保留随时对其产品进行修改和完善的权利，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。请用户在购买前向我司获取相关产品的最新信息，本文档一切解释权归广州创龙所有。

©2014-2018 Guangzhou Tronlong Electronic Technology Co.,Ltd. All rights reserved.

公司简介

广州创龙电子科技有限公司（简称“广州创龙”，英文简称“Tronlong”），是杰出的嵌入式方案商，专业提供嵌入式开发平台工具及嵌入式软硬件定制设计及技术支持等服务，专注于 DSP+ARM+FPGA 三核系统方案开发，和国内诸多著名企业、研究所和高校有密切的技术合作，如富士康、三一重工、中国科学院、清华大学等国内龙头企业和院校。

TI 嵌入式处理业务拓展经理 Zheng Xiaolong 指出：“Tronlong 是国内研究 OMAP-L138 最深入的企业之一，Tronlong 推出 OMAP-L138+Spartan-6 三核数据采集处理显示解决方案，我们深感振奋，它将加速客户新产品的上市进程，带来更高的投资回报率，使得新老客户大大受益。”

经过近几年的发展，创龙产品已占据相关市场主导地位，特别是在电力、通信、工控、音视频处理等数据采集处理行业广泛应用。创龙致力于让客户的产品快速上市、缩短开发周期、降低研发成本。选择创龙，您将得到强大的技术支持和完美的服务体验。

产品保修

广州创龙所有产品保修期为一年，保修期内由于产品质量原因引起的，经鉴定系非人为因素造成的产品损坏问题，由广州创龙免费维修或者更换。

更多帮助

销售邮箱: sales@tronlong.com

技术邮箱: support@tronlong.com

公司总机: 020-8998-6280

公司官网: www.tronlong.com

销售邮箱: sales@tronlong.com

技术邮箱: support@tronlong.com

公司总机: 020-8998-6280

公司官网: www.tronlong.com 2/19



目 录

1 前言	4
2 提取 Linux MCSDK 工具软件包	4
3 在 Windows 下新建和配置 CCS 工程	5
4 在开发板加载并运行 DSP 程序镜像	13
5.1 设置 U-Boot 参数	13
5.2 安装 syslink 驱动	13
5.3 加载并运行 DSP 程序镜像	14
5 CCS 环境下双核通信 DSP 程序仿真	14
5.1 设置 U-Boot 参数	15
5.2 安装 syslink 驱动	15
5.3 测试仿真器连接状态	15
5.4 连接 DSP	18
5.5 DSP 仿真	19



1 前言

从 TI 官网提供的资料来看，可以看出 TI 建议的 OMAP-L138 的双核开发流程如下：

- (1) ARM 端收集 AD、音视频等数据，并将数据通过双核通信机制（dsplink、syslink 或者其他第三方双核通信软件）送到 DSP 端。ARM 端的数据可以来自 FPGA、CPLD、网络等外接设备，也可以是 ARM 端直接采集的 AD 或者音视频等数据。
- (2) DSP 端将 ARM 端传输过来的数据进行算法处理，接着通过双核通信机制将处理后的数据送往 ARM 端。
- (3) ARM 端根据 DSP 端传输过来的数据，实现逻辑控制、应用界面开发、数据存储、数据网络转发等一系列功能。

从以上几点分析，DSP C6748 核心充当了 ARM 端的一个协处理器，相当于一个普通的外设。而 DSP 端和 ARM 之间的主要以数据传输为主，DSP 比较少涉及外设的控制。

综合以上信息，在进行 DSP+ARM 双核通信开发时，使用 CCS 仿真 DSP 程序就不是一个必要选项了。但是，为了配合部分客户提出在进行双核开发时仿真 DSP 程序的要求，Tronlong 提供了如下仿真步骤。

本教程实现了将 OMAP-L138 的 syslink 双核通信 DSP 端代码在 CCS 环境下独立编译和仿真。

请先按照文档《OMAPL138 的多核软件开发组件 MCSDK 开发入门.pdf》编译 syslink 并获得 syslink.ko 驱动和 slaveloader 程序。

备注：推荐使用我司验证过的 CCS5.5 软件进行以下操作。

2 提取 Linux MCSDK 工具软件包

提取 Linux 虚拟机中 MCSDK 的工具软件包到 Windows 下，包括以下两部分代码软件包目录：

销售邮箱：sales@tronlong.com

技术邮箱：support@tronlong.com

公司总机：020-8998-6280

公司官网：www.tronlong.com 4/19



Host# /home/tl/ti/ipc_1_25_03_15

Host# /home/tl/ti/syslink_2_21_01_05

将以上两个目录拷贝到 Windows 的 CCS5 安装路径下，例如：D:\ti 下。

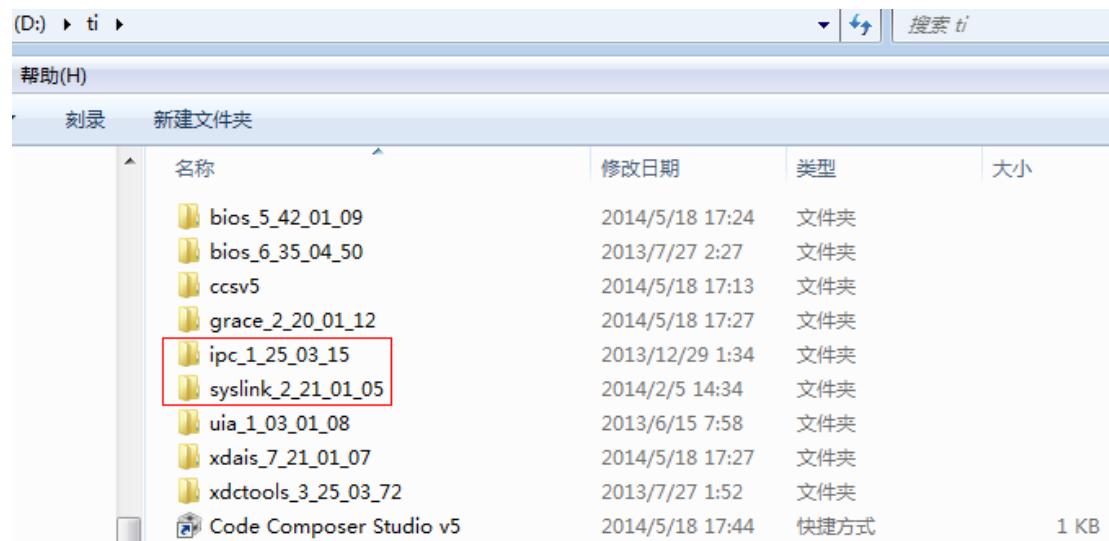


图 1

3 在 Windows 下新建和配置 CCS 工程

以提取 ex01_helloworld 的 DSP 端代码为例。

(1) 先在虚拟机的 /home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/ 目录下执行以下命令清除已经编译的代码：

Host# sudo make clean



```
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$ pwd
/home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$ sudo make clean
rm -rf install
make -C host clean
make[1]: Entering directory `/home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/host'
rm -rf bin
make[1]: Leaving directory `/home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/host'
make -C dsp clean
make[1]: Entering directory `/home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/dsp'
rm -rf configuro bin
make[1]: Leaving directory `/home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/dsp'
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$
```

图 2

- (2) 再将虚拟机中 /home/tl/ti/syslink_2_21_01_05/examples/ex01_helloworld/ 目录下的 shared 和 dsp 目录拷贝到共享目录。

```
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$ ls
dsp host makefile products.mak readme.txt run.sh shared
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$ cp -a shared/ /mnt/hgfs/shareVM/
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$ cp -a dsp/ /mnt/hgfs/shareVM/
tl@tl-desktop:~/ti/syslink_2_21_01_05/examples/ex01_helloworld$
```

图 3

- (3) 打开 Windows 下的 CCS 软件，输入工作路径：

D:\CCS5.5\workspace\ex01_helloworld





广州创龙电子科技有限公司

Guangzhou Tronlong Electronic Technology Co., Ltd

DSP+ARM+FPGA

三核主板方案领导者

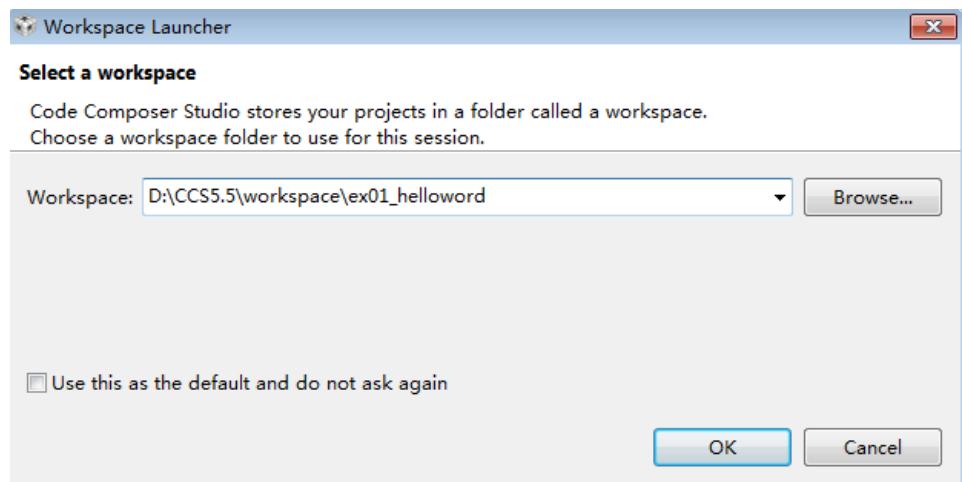


图 4

(4) 点击 CCS 菜单 File->New->CCS Project, 选择如下配置:

销售邮箱: sales@tronlong.com

公司总机: 020-8998-6280

技术邮箱: support@tronlong.com

公司官网: www.tronlong.com 7/19

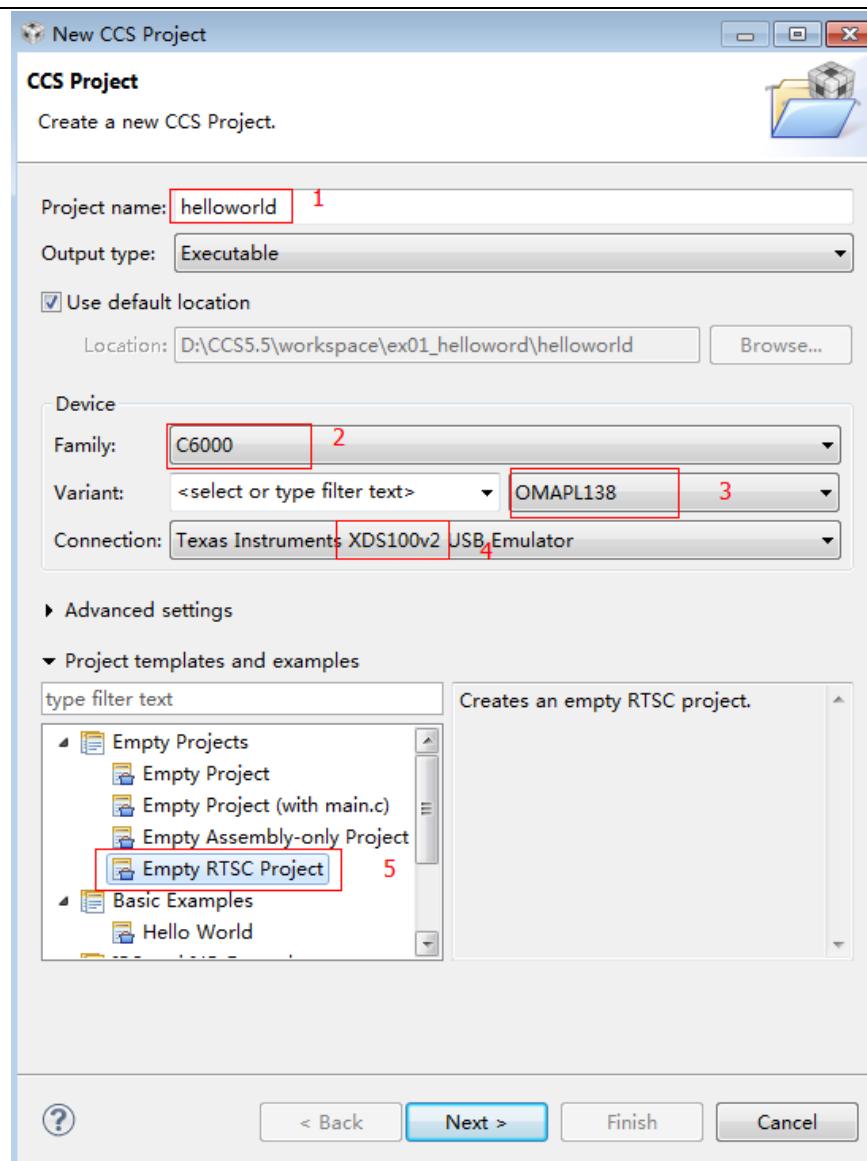


图 5

helloworld: 工程名字

XDS100v2: 仿真器类型, 请对应使用的仿真器型号。

点击 Next, 弹出如下界面:

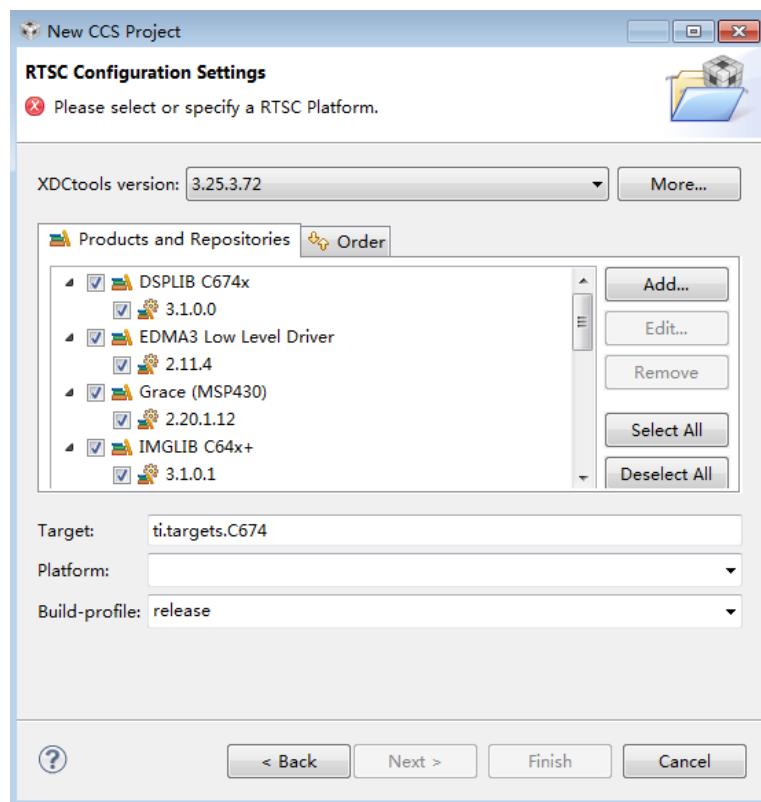


图 6

(5) 点击“Deselect All”去掉默认的程序包选项，点击“Add”，弹出如下界面：

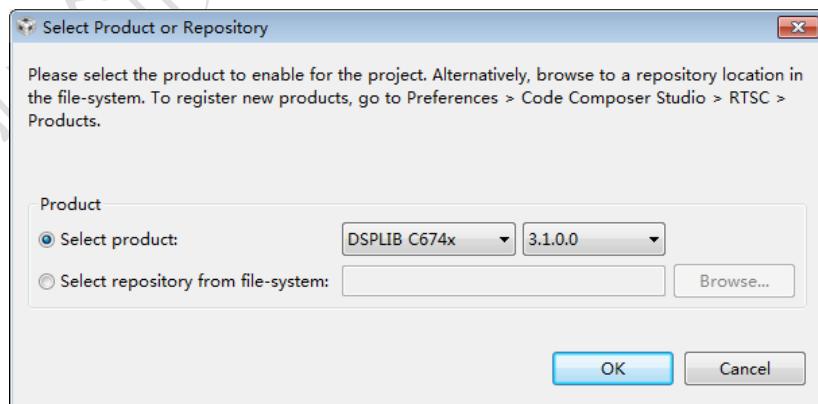


图 7

(6) 点击“Select repository from file-system”，重复两次，将以下两个路径添
销售邮箱：sales@tronlong.com 技术邮箱：support@tronlong.com
公司总机：020-8998-6280 公司官网：www.tronlong.com 9/19



加进来：

D:\ti\ipc_1_25_03_15\packages

D:\ti\syslink_2_21_01_05\packages

(7) 点击 OK, 将弹出界面中 Platform 改为“ti.platforms.evmOMAPL138:dsp”，

并选上 SYS/BIOS 6.35.4.50, 最后得到如下设置：

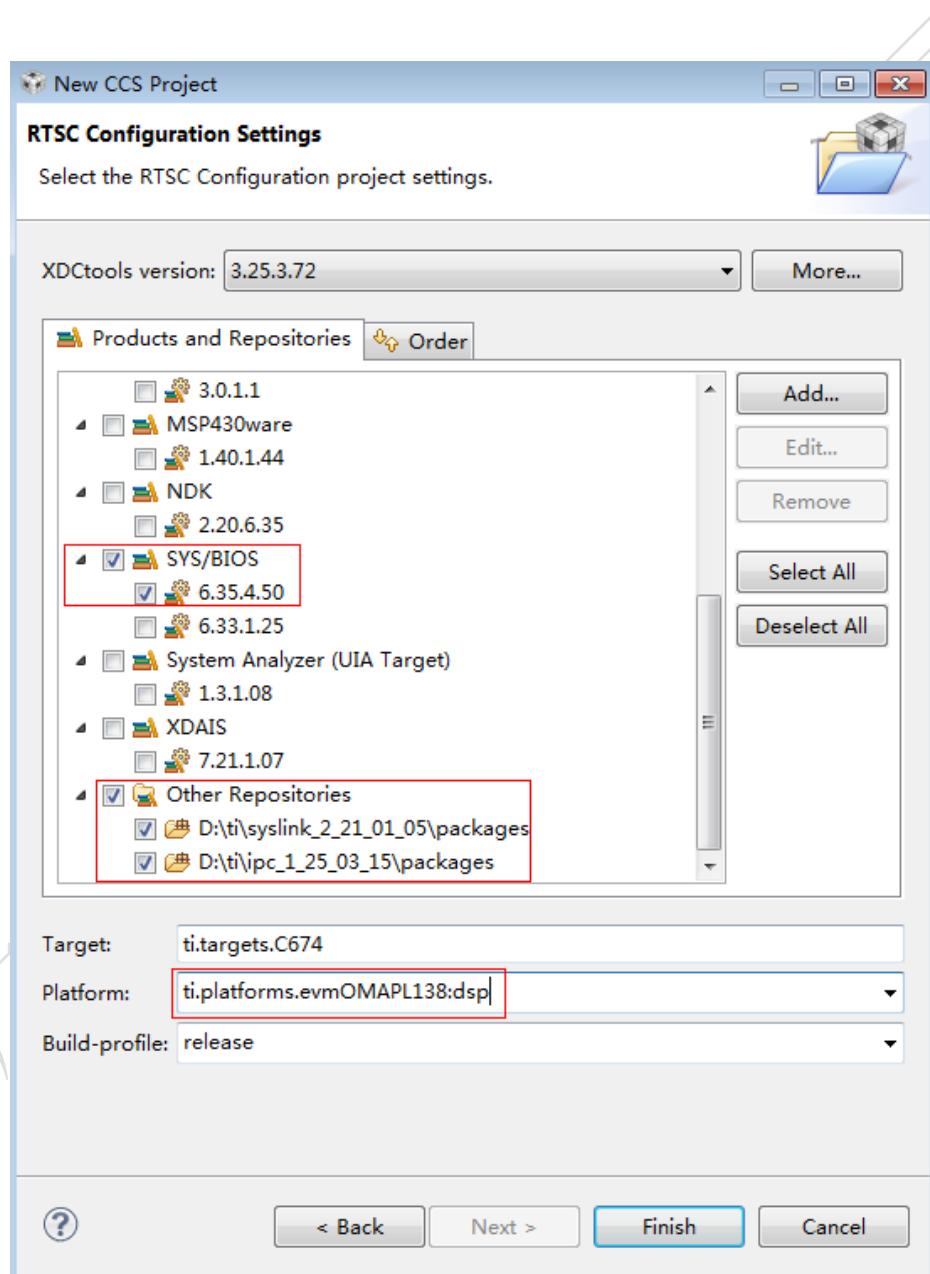


图 8



(8) 点击 Finish, 此时工程设置已经完成, 可以看到工具软件包 packages 文件夹已经被包含了, 如下图所示:

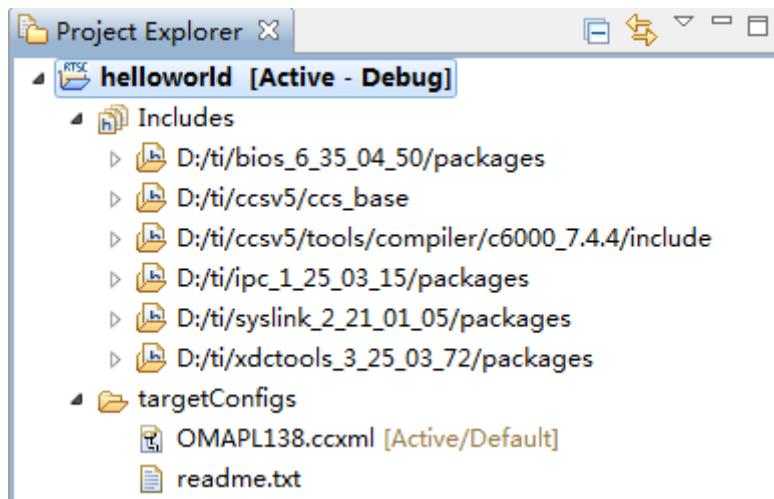


图 9

(9) 将共享目录中的 shared 和 dsp 目录拷贝到工程目录下, 例如本示例工程中, 路径如下:

D:\CCS5.5\workspace\ex01_helloworld\helloworld

拷贝后 CCS 界面如下图所示:

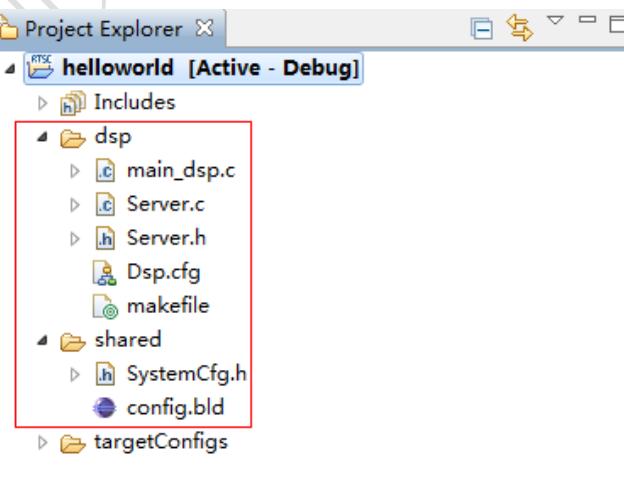


图 10



在 dsp 目录下有 Dsp.cfg 文件，在 shared 目录下有 config.bld 文件。这两个文件的主要作用是配置共享内存、调用 xdc 工具、生成 linker.cmd 的链接文件，双核通信主要围绕这两个文件来修改。

- (10) 删除 dsp/makefile 文件，这个是 Linux 下用的，在 CCS 中无作用。
- (11) 右击工程，在弹出的对话框中选择“Show Build Settings”，弹出如下界面：

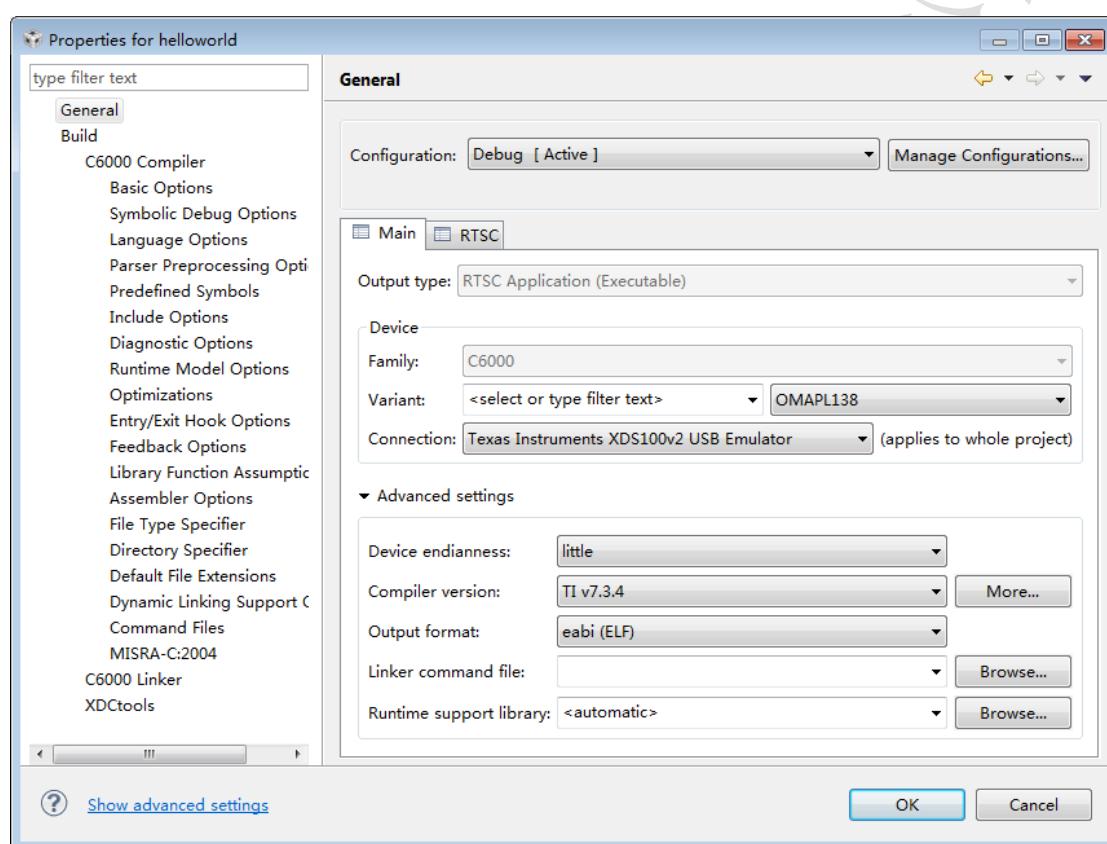


图 11

将“Output Format”修改为 ELF。

- (12) 右击工程，在弹出的界面中点击“Clean Project”清理工程。接着右击工程，在弹出的界面中点击“Build Project”编译工程，就会生成 helloworld.out 文件，如下图所示：



图 12

如果需要有底板 LED 闪烁效果,请使用光盘 demo\syslink\syslink_dsp\Server.c 文件替换工程中 dsp\Server.c, 执行“Clean Project”清理工程后重新编译即可。

4 在开发板加载并运行 DSP 程序镜像

可以通过 SD 卡或者 NFS 的方式启动文件系统。下面以 SD 卡启动方式为例讲解 syslink 示例程序的运行方法。

5.1 设置 U-Boot 参数

上电启动后在 U-Boot 命令行执行以下命令:

```
Target# setenv bootargs console=ttyS2,115200n8 root=/dev/mmcblk0p2 rw  
rootfstype=ext3 mem=32M@0xc0000000 mem=64M@0xc4000000  
Target# sveenv  
Target# boot
```

备注: 通过 nfs 方式只需要将原来的网络参数“mem=128M”修改为“mem=32M@0xc0000000 mem=64M@0xc4000000”即可。

5.2 安装 syslink 驱动

进入开发板后, 执行以下命令安装 syslink 驱动:

```
Target# insmod /lib/modules/3.3.0/kernel/drivers/dsp/syslink.ko TRACE=1  
TRACEFAILURE=1
```



```
root@tl:~# insmod /lib/modules/3.3.0/kernel/drivers/dsp/syslink.ko TRACE=1 TRACE
FAILURE=1
[ 57.238858] SysLink version : 2.21.01.05
[ 57.238888] SysLink module created on Date:Feb  8 2014 Time:21:47:30
[ 57.259879] Trace enabled
[ 57.262592] Trace SetFailureReason enabled
root@tl:~#
```

图 13

5.3 加载并运行 DSP 程序镜像

将 helloworld.out 拷贝到开发板/ex01_helloworld/debug 目录下。没有此目录请先按照 MCSDK 开发入门文档操作。将 debug 目录下的 run.sh 复制为 run2.sh，并修改 run2.sh 内容为如下：

```
set -x

./slaveloader startup DSP helloworld.out

./app_host DSP

./slaveloader shutdown DSP
```

图 14

进入执行示例程序所在目录/ex01_helloworld/debug，分别执行 run.sh 和 run2.sh 脚本文件，发现效果是一样的。如果替换过 Server.c 文件，执行 run2.sh 可观察到底板 LED 闪烁两次。

在开发板/ex01_helloworld/debug 目录下运行 run.sh 是加载 Linux 端编译出来的 server_dsp.xe674 DSP 镜像。而运行 run2.sh 是加载 Windows 的 CCS 端编译出来的 helloworld.out DSP 镜像。

5 CCS 环境下双核通信 DSP 程序仿真

连接好仿真器，打开 CCS。通过 SD 卡或者 NFS 的方式启动文件系统。下面销售邮箱：sales@tronlong.com

技术邮箱：support@tronlong.com

公司总机：020-8998-6280

公司官网：www.tronlong.com 14/19



以 sd 卡启动方式为例讲解 syslink 示例程序的运行方法。

5.1 设置 U-Boot 参数

上电启动后在 U-Boot 命令行执行以下命令：

```
Target# setenv bootargs console=ttyS2,115200n8 root=/dev/mmcblk0p2 rw  
rootfstype=ext3 mem=32M@0xc0000000 mem=64M@0xc4000000
```

```
Target# sveenv
```

```
Target# boot
```

备注：通过 nfs 方式只需要将原来的网络参数 “mem=128M” 修改为 “mem=32M@0xc0000000 mem=64M@0xc4000000” 即可。

5.2 安装 syslink 驱动

进入开发板后，执行以下命令安装 syslink 驱动：

```
Target# insmod /lib/modules/3.3.0/kernel/drivers/dsp/syslink.ko TRACE=1  
TRACEFAILURE=1
```

```
root@tl:~# insmod /lib/modules/3.3.0/kernel/drivers/dsp/syslink.ko TRACE=1 TRACE  
FAILURE=1  
[ 57.238858] SysLink version : 2.21.01.05  
[ 57.238888] SysLink module created on Date:Feb 8 2014 Time:21:47:30  
[ 57.259879] Trace enabled  
[ 57.262592] Trace SetFailureReason enabled  
root@tl:~#
```

图 15

5.3 测试仿真器连接状态

点击 CCS 菜单“View->Target Configurations”，右边弹出如下对话框：



图 16

点击右边第一项，新建配置文件。



图 17

输入工程名字：omapl138.ccxml，点击 finish。

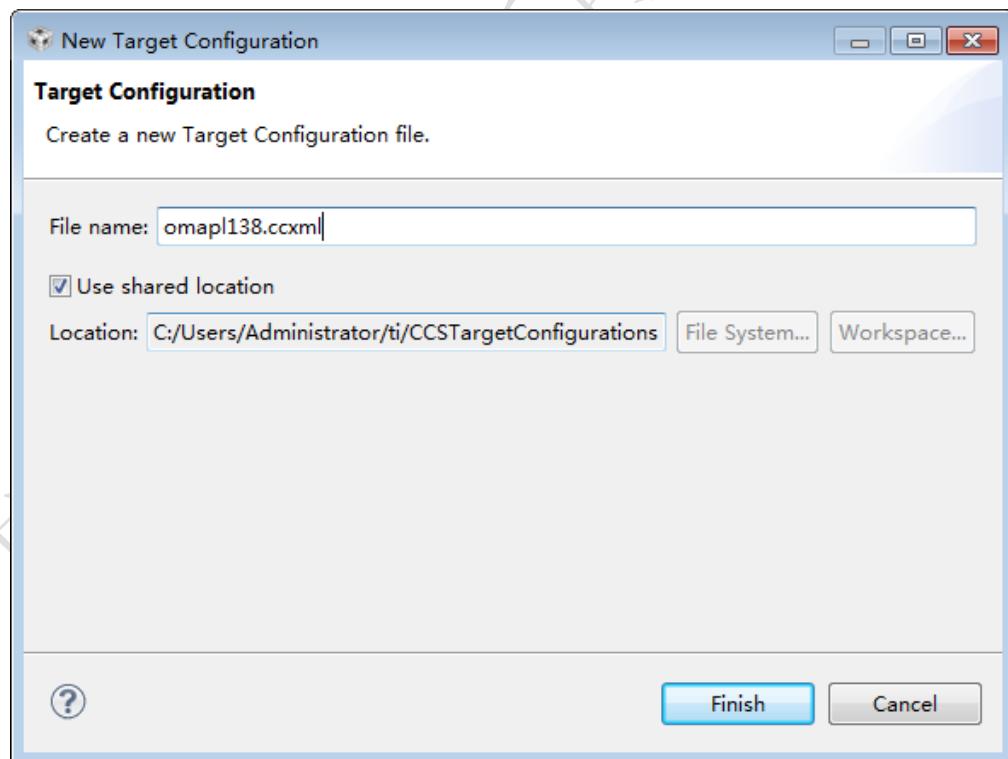


图 18



在弹出的对话框中，“Connection”选择 XDS100V2，“Board or Device”选择 OMAPL138，点击右边的 Save 按钮。

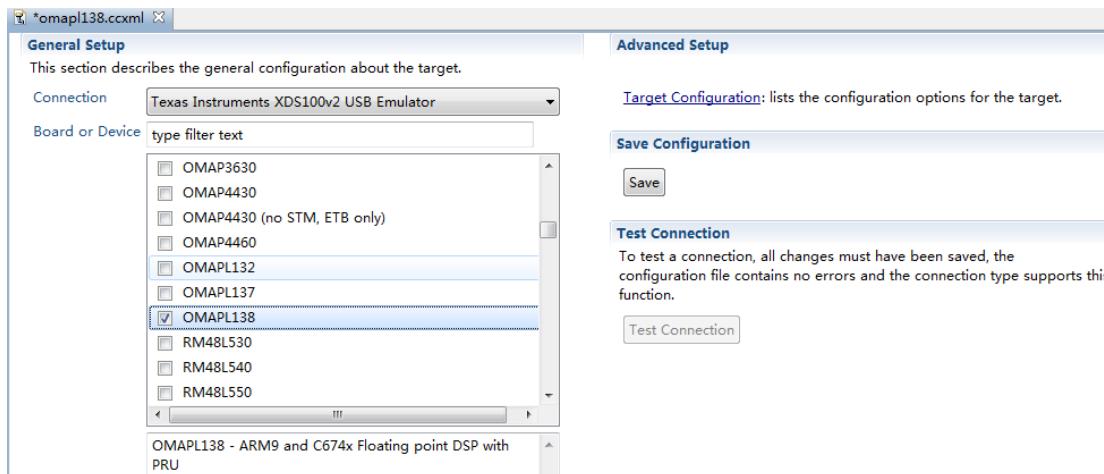


图 19

若使用 XDS100V3 仿真器，设置方法请参考前面烧写 U-Boot 到 NAND FLASH 章节。点击“Test connection”，看是否提示成功连接。如下图。如提示错误，请检查开发板是否上电、接线是否正常。

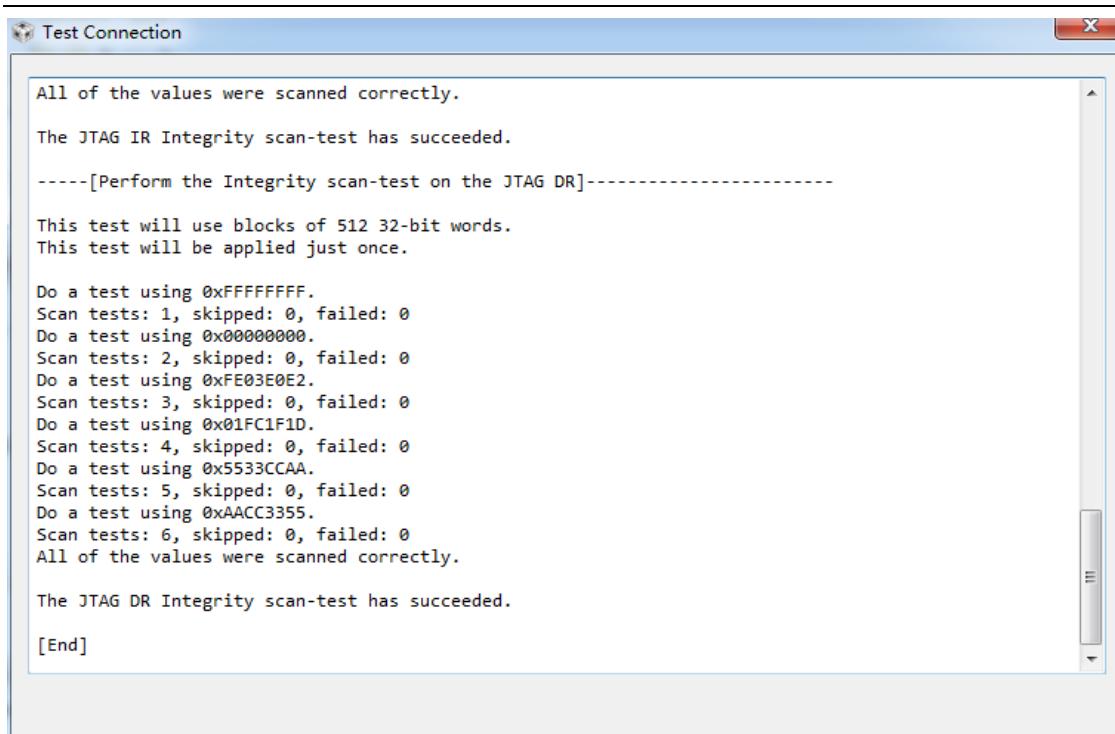


图 20

5.4 连接 DSP

点击“Run->Debug”，弹出以下界面。可以看到第一项“C674X”和第四项“ARM9”。



图 21

右击第一项“DSP”，选择“Connect Target”选项，会显示“Suspend”状态。这说明 CCS 已经和 OMAPL138 的 DSP 核正常连接起来了。

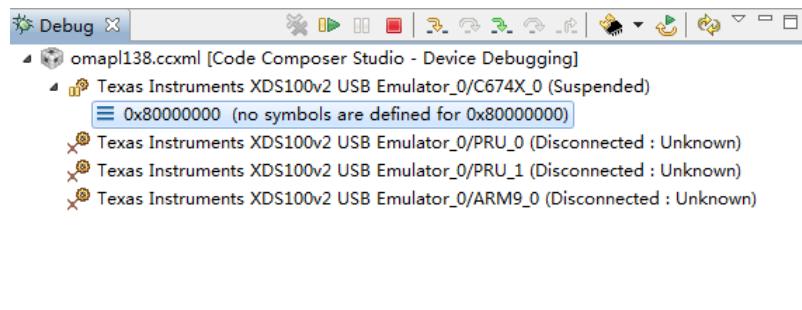


图 22

5.5 DSP 仿真

按照前面章节中 DSP 开发入门文档中的操作去加载上面工程中的 helloworld.out 镜像后，然后点击 按钮程序就可以运行了。若是替换了 Server.c 可以发现底板的 LED 会闪烁两次。按 F5 会进去函数里面，按 F6 就单步走。同时看到核心板的心脏灯在闪烁，说明了在 DSP 仿真时，ARM 端正常工作。

双核通信中的 DSP 仿真，已经不需要 gel 文件和唤醒 DSP 核的操作，因为 ARM 端启动 Linux 系统和加载 syslink.ko 实现了同等功能。