



TMS320C6748 开发板使用手册

Revision History

Revision No.	Description	Draft Date	Remark
V1.0	1.初始版本	2014/5/5	



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 ZhengXiaolong 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



目录

前言	5
1 开发板准备	6
1.1 测试开发板硬件	6
1.2 查看仿真器驱动是否正常安装	6
2 程序烧写	7
2.1 设置工程配置文件信息	7
2.2 测试仿真器是否正常连接	9
2.3 加载 GEL 文件	10
2.4 CCS 连接开发板 CPU	11
2.5 运行 GEL 文件来初始化开发板	12
2.6 查看 CPU 版本号	13
2.7 加载和运行程序	13
2.8 擦除和烧写 Nand Flash	14
2.9 镜像格式转换方法	15
3 CCS 工程新建和编译	17
3.1 新建工程	19
3.2 编写程序	22
3.3 编译和运行 DSP 程序	23
4 CCS 工程导入和编译步骤	25
5 Demo 例程演示	30
5.1 GPIO_LED	30
5.2 GPIO_KEY	30
5.3 TIMER	30
5.4 UART1_POLL	31
5.5 UART2_INT	32
5.6 IIC_EEPROM	33



5.7	SPI_FLASH.....	34
5.8	WatchDog.....	34
5.9	NMI.....	35
5.10	PWM.....	36
5.11	ECAP.....	36
5.12	RTC.....	36
5.13	LCDC_GrLIB.....	37
5.14	MMCSD.....	39
5.15	SATA.....	39
5.16	USB_DEV_BULK.....	39
5.17	ENET_HTTPD.....	46
5.18	ENET_ECHO.....	48
5.19	AUDIO_PLAY.....	55
5.20	MCBSP.....	55
5.21	FFT.....	55
6	附录.....	60
6.1	Boot Mode Selection.....	60



前言

在进行以下操作之前，请先安装 CCS，推荐使用我司验证过的 CCS5.5.0 版本。
安装包 CCS5.5.0.00077_win32.zip 可以在 TMS320C6748 产品资料光盘的 tools 目录下找到，也可以在 TI 官网下载，可以下载各种版本的 CCS。

TI 官网下载链接：

http://processors.wiki.ti.com/index.php/Download_CCS#Code_Composer_Studio_Version_5_Downloads

1 开发板准备

1.1 测试开发板硬件

将开发板的拨码开关 1~5 号对应拨到 00101（说明见附录，1 代表 ON，0 代表 OFF，X 代表任意，即可以是 0 也可以是 1），这是 UART2 档位。RS232 串口线通过 USB 转串口接到 PC 机的 USB 口，DB9 母头（带孔）接底板 UART2 口，设置 PC 机调试终端 COM 口，波特率为 115200，8N1，无检验位。推荐使用 ZOC 串口调试终端软件，TMS320C6748 产品资料光盘的 tools 目录下有 ZOC 的安装包 zoc602.zip。

假如是使用 ZOC 的用户，依次点击菜 ZOC 单栏的 Options->Jump to->Devices，然后选择 Serial/Modem。在 Serial/Modem 中扫描然后选择 COM 口，波特率为 115200，8N1，无检验位，RTS signal off，DTR signal off，其它选项按默认设置。

开发板上电，即可在调试终端看到系统自动打印的 BOOTME 信息，说明开发板硬件正常工作了。

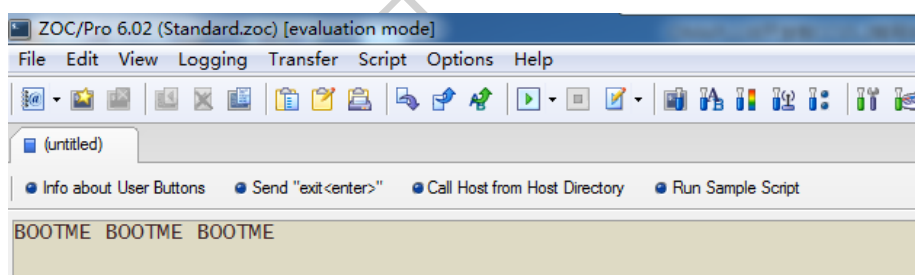


图 1

1.2 查看仿真器驱动是否正常安装

以 XDS100V2 仿真器为例，开发板断电，连接好仿真器和开发板，并将仿真器的 USB 口插进电脑 USB 插槽，开发板上电。右击计算机图标，点击设备->通用串行总线控制器，查看是否有 TI XDS100 Channel A 和 TI XDS100 Channel B 选项出现，如有说明仿真器驱动已经正常安装，否则请正常安装 XDS100V2 的驱动程序。

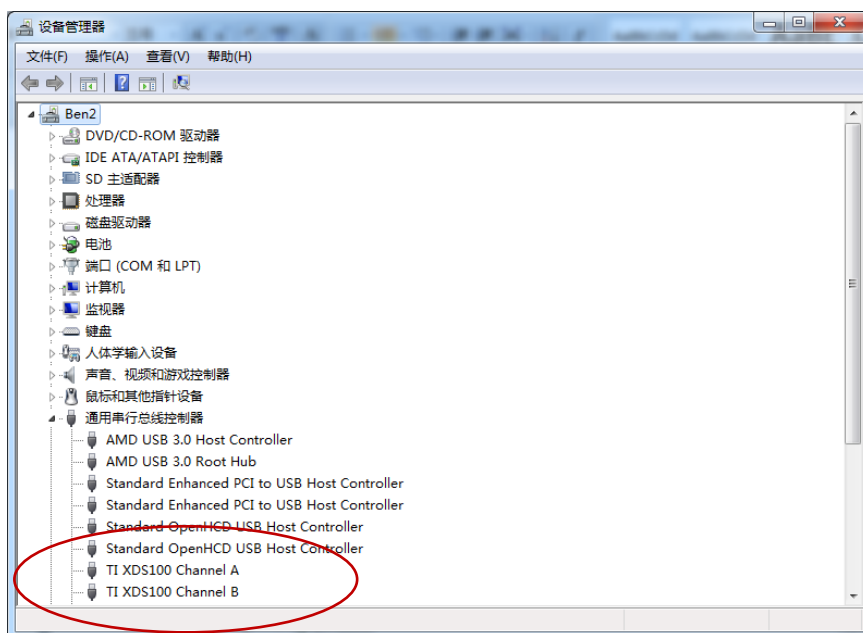


图 2

2 程序烧写

2.1 设置工程配置文件信息

新建"C:\TMS320C6748"目录,然后将光盘中 images 目录下的 C6478_NandWrite.out、C6748AISgen_456M_config.cfg、TronLong_C6748.gel 文件复制到"C:\TMS320C6748"文件夹。打开 CCS5.5, 点击 View->Target Configurations, 右边弹出如下对话框:

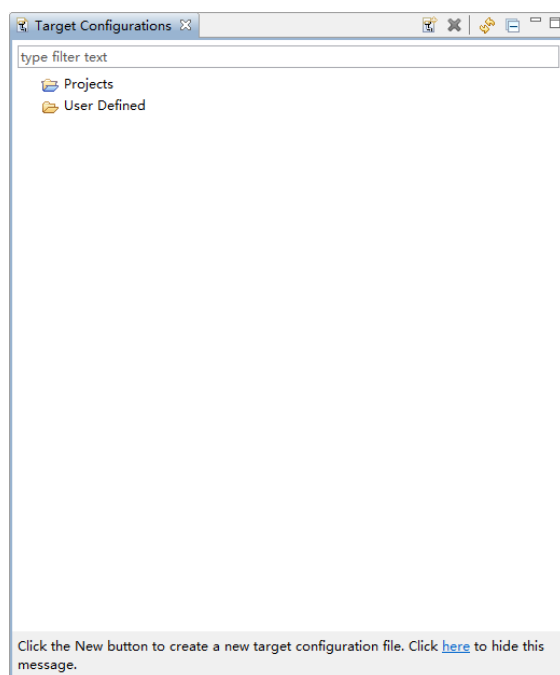


图 3

点击左边第一项，新建配置文件。

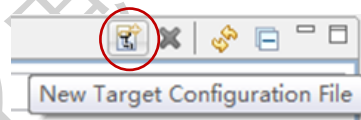


图 4

输入工程名字：TMS320C6748.ccxml，点击 Finish。

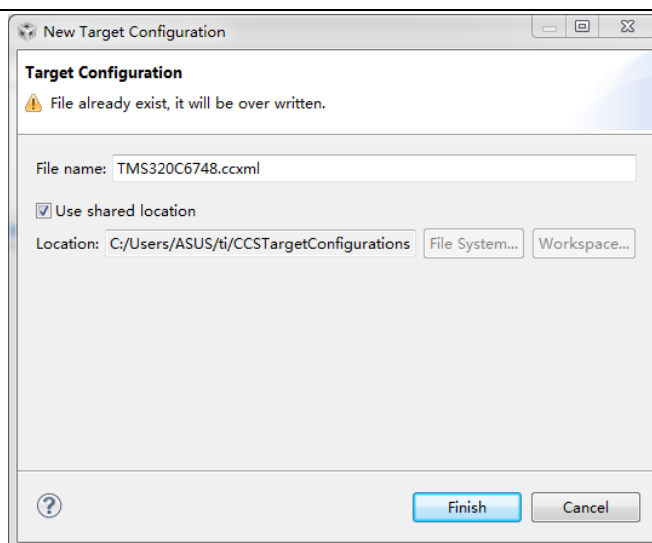


图 5

在弹出的对话框中, Connection 选择 XDS100V2, Board or Device 选择 TMS320C6748, 点击右边的 Save 按钮。

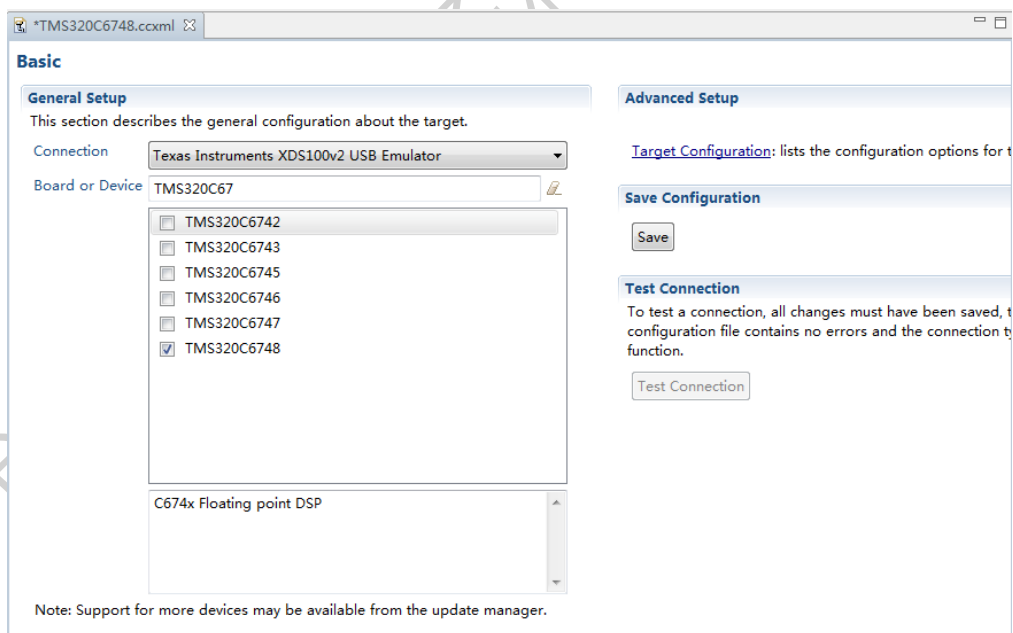


图 6

2.2 测试仿真器是否正常连接

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

9/60

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。

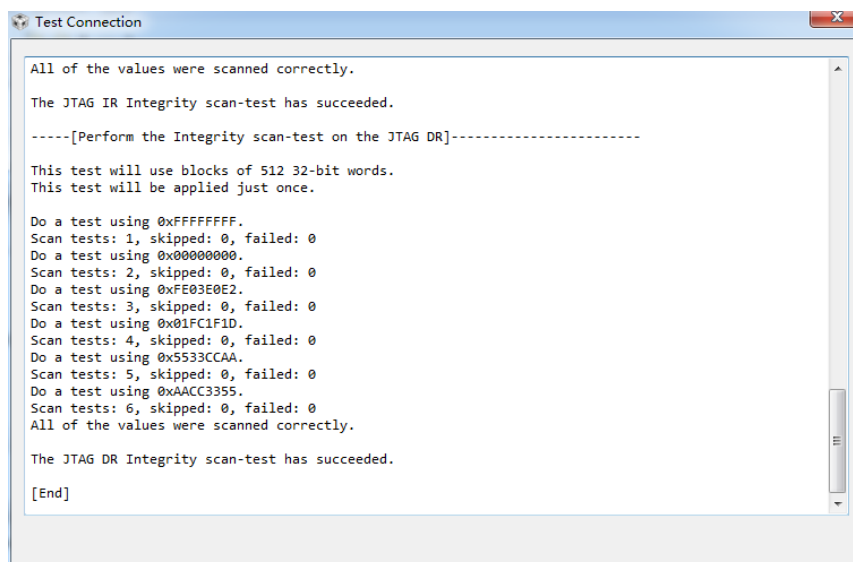


图 7

2.3 加载 GEL 文件

点击 Run->Debug, 弹出以下界面。可以看到 C674X_0。

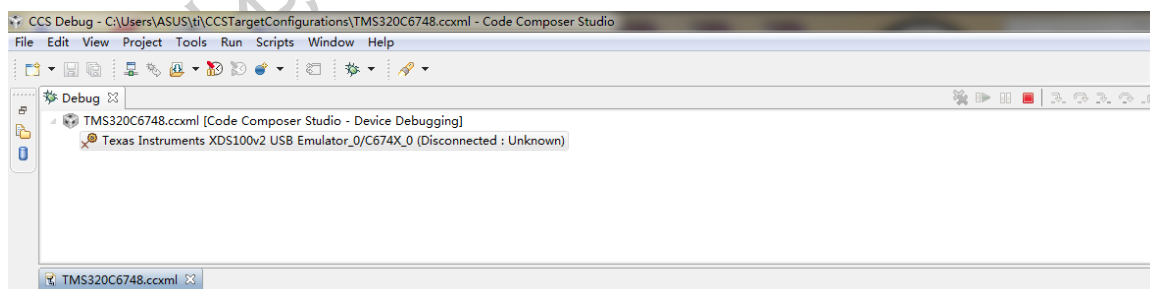


图 8

右击 C674X_0 核, 在弹出的界面中选择"Open GEL Files View"选项, 右下角会弹出 GEL Files(TMS320C674X)对话框。

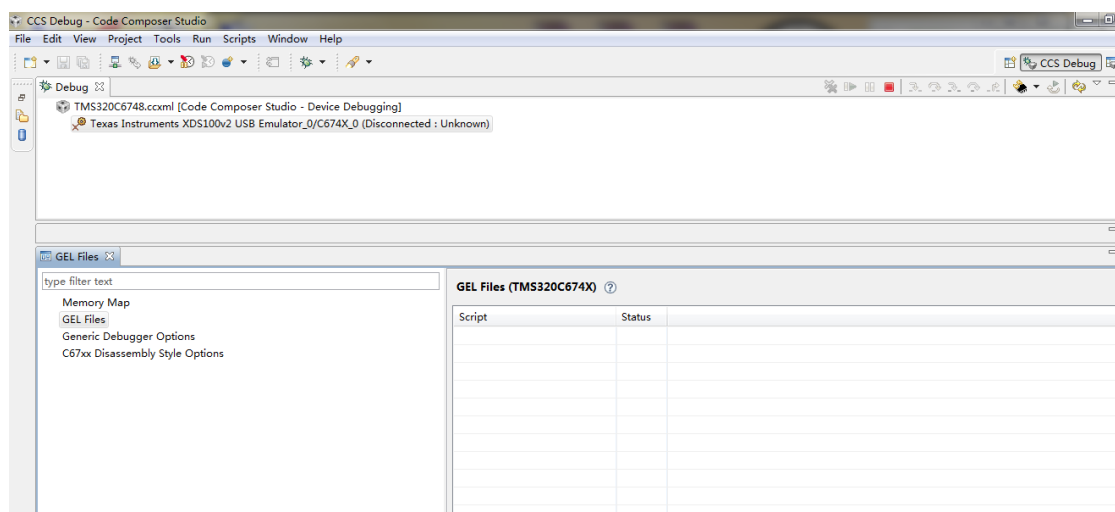


图 9

在对话框内点击右键，在弹出的界面中选择"Load GEL"。选择"C:\TMS320C6748\TronLong_C6748.gel"文件后，点击确定，右下角的 GEL Files(TMS320C674X)对话框会出现“Success”提示语句，如下图：

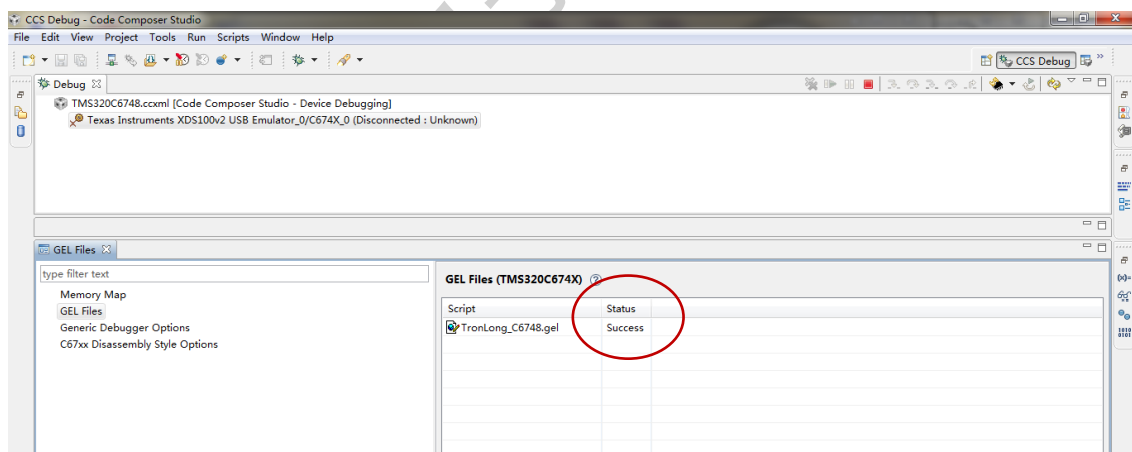


图 10

2.4 CCS 连接开发板 CPU

右击 C674X_0 核，选择 Connect Target 选项，会显示 Suspended 状态。这说明 CCS

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

11/60

已经和开发板 CPU 正常连接起来了。

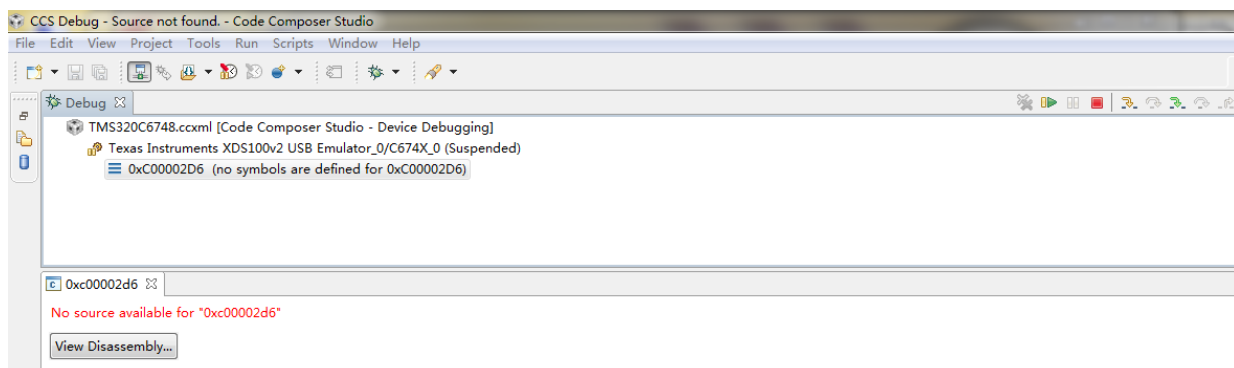


图 11

2.5 运行 GEL 文件来初始化开发板

根据芯片主频，点击"Scripts->TronLong_DSP_C6748-> TronLong_C6748_456MHz", 运行后会提示 Config Complete，开发板初始化完成。

GEL 文件的作用是在调试模式初始化硬件环境（DDR2、NAND FLASH 等），只有加载了 GEL 文件后才会出现此菜单，只要开发板不断电 初始化操作只需要执行一次就可以。

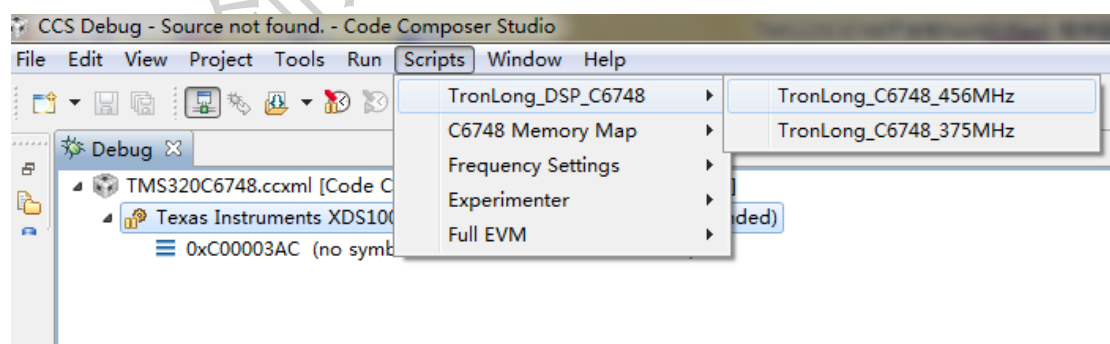


图 12

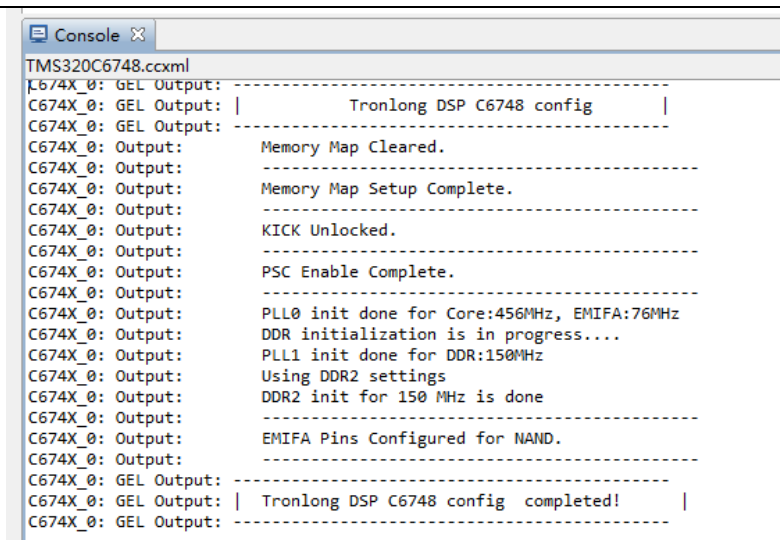


图 13

2.6 查看 CPU 版本号

点击 View->Memory Browser,会出现 Memory Browser 窗口,在输入框中输入 0x11700000 并回车,选择“Character”可查询 TMS320C6748 核的版本号,CPU 版本号在文件格式转换需用到;例如这里的 DSP 核版本号为: d800k008, 如下图:

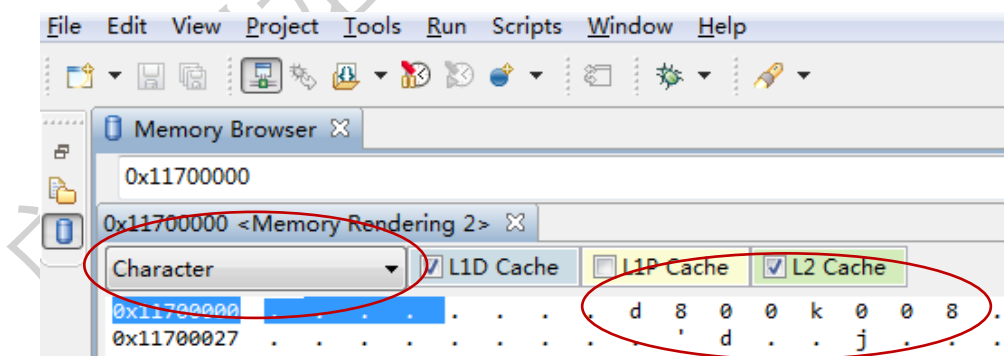


图 14

2.7 加载和运行程序

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

13/60

点击"Run->Load->Load Program", 选择"C:\TMS320C6748\C6478_NandWrite.out"文件, 点击 OK。接着点击绿色三角启动按钮。在 Console 窗口会有打印信息出现, 提示是否擦除 NAND flash。

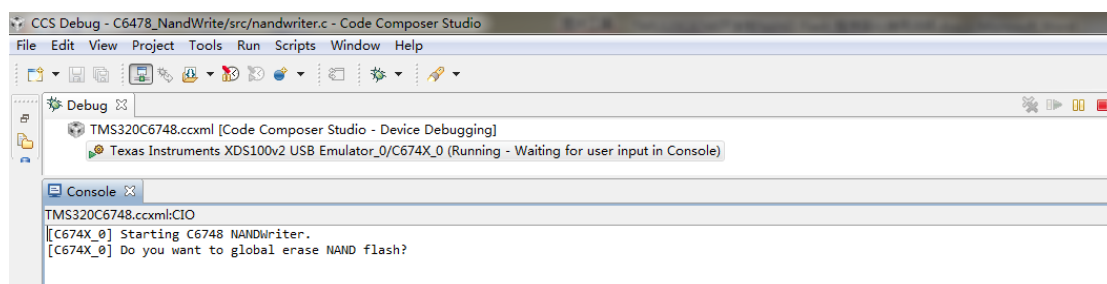


图 15

备注：如果第一次操作没有出现 Console 窗口，请按黄色键暂停运行，然后按重启键复位 CPU，接着点击绿色三角启动按钮，即可看到 Console 窗口会有打印信息出现。

2.8 擦除和烧写 Nand Flash

以 LED 为例，把光盘中 demo 整个目录文件拷贝到 C 盘 TMS320C6748 文件夹下（若不存在请先建立此文件夹）。

在 Console 窗"[C674X_0] Do you want to global erase NAND flash?"行后面输入"y", 按回车键进行擦除 Nand Flash。擦除完后，会提示"Enter the binary AIS file name to flash (enter 'none' to skip) :", 在下一行空白处输入要烧写的 DSP 程序镜像 GPIO_LED.bin 路径 "C:\TMS320C6748\demo\Application\GPIO_LED\Debug\GPIO_LED.bin", 并按回车键。等待出现"NAND boot preparation was successful!"即烧写成功。烧写时间与文件大小有关，一般可在 15 秒内烧写完毕。

烧写完 GPIO_LED.bin 镜像后，拨下开发板仿真器接口，将开发板的拨码开关 1~5 号对应拨到 01110 档位，设置从 Nand flash 启动，可以看到开发板 LED 流水灯效果。

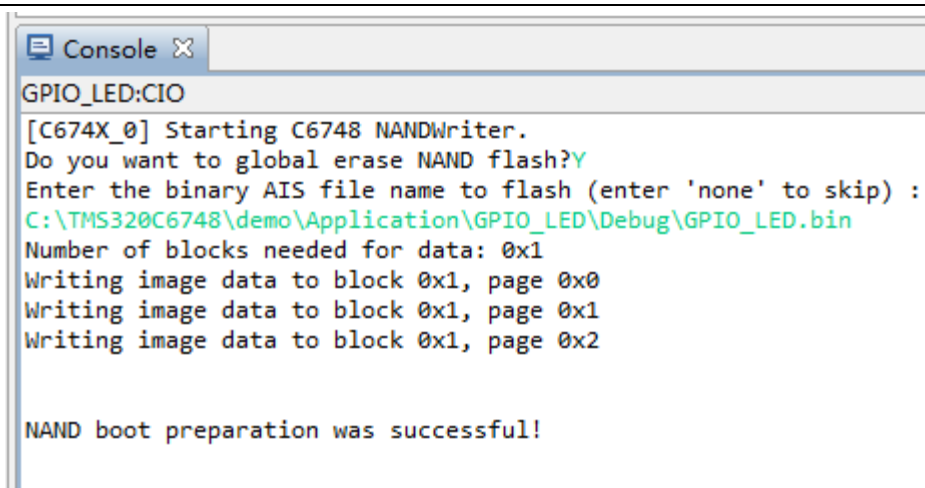


图 16

2.9 镜像格式转换方法

C6478_NandWrite.out 只支持.bin 格式镜像烧写，而 CCS 编译程序生成的镜像为.out 格式文件，需把.out 格式镜像转换为.bin 格式镜像才能进行烧写，有关格式转换步骤如下：

- (1) 如果 AIS 是安装在默认路径，请双击"C:\AISgen_d800k008_Install_v1.13\AISgen for D800K008\AISgen_d800k008.exe"，弹出如下界面：

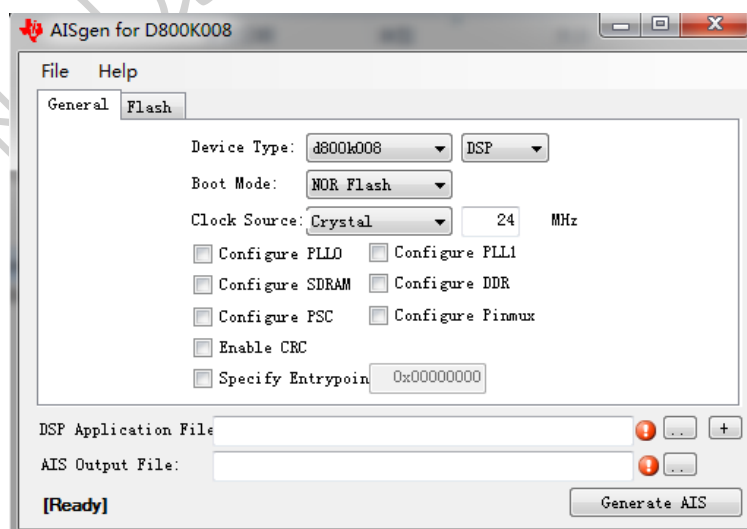


图 17

(2) 点击"File->Load Configuration"，弹出如下界面：

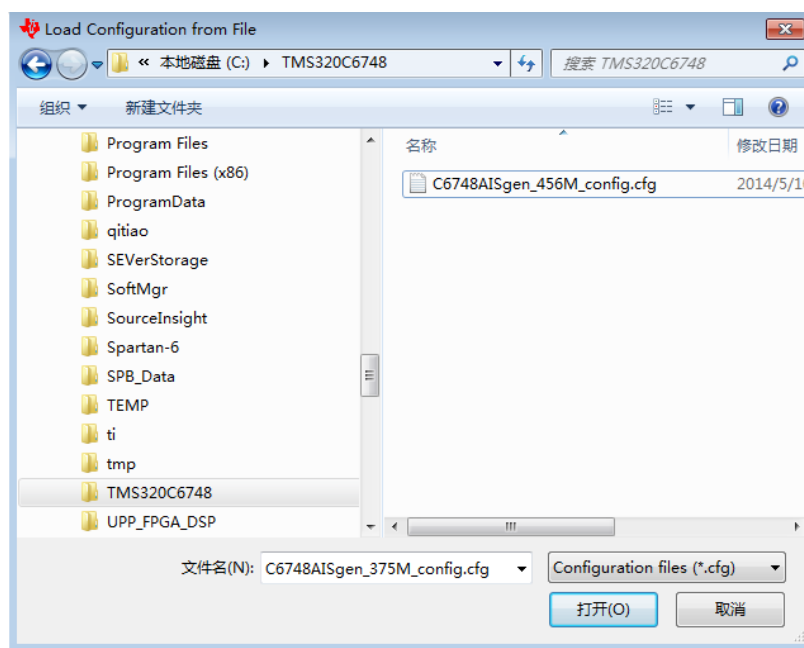


图 18

(3) 根据芯片主频选择"C:\TMS320C6748\C6748AISgen_456M_config.cfg"。在"Device Type"选框中选择 CPU 对应的 CPU 版本号 d800k008，此版本号可根据前面版本号查询操作得到。在"DSP Application File"处选择 CCS 编译后的.out 格式文件路径在。在"AIS Output File"处输入需输出的.bin 格式文件的路径和名字。

最后点击"Generate AIS"按键，启动格式转换，可生成可烧写的.bin 格式镜像，如下图所示：

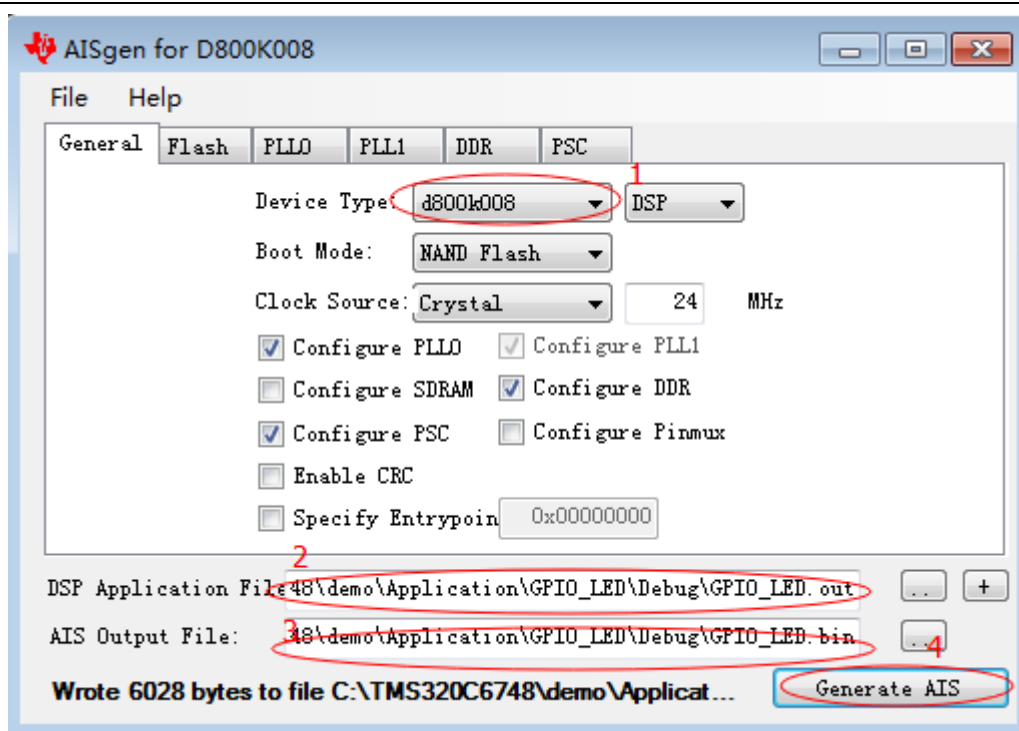


图 19

3 CCS 工程新建和编译

为了更好的显示中文，请修改 Preferences 中 C++编译器字体大小及控制台字体，点击 CCS 菜单"Windows->Preferences"，如下图所示：

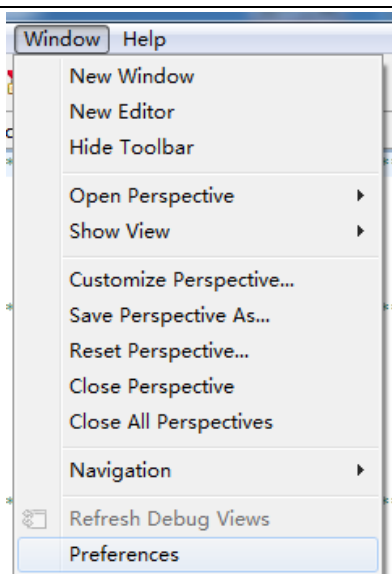


图 20

在弹出的界面中修改 C/C++ Editor Text Font 字体 为 12 号以上，如下图所示：

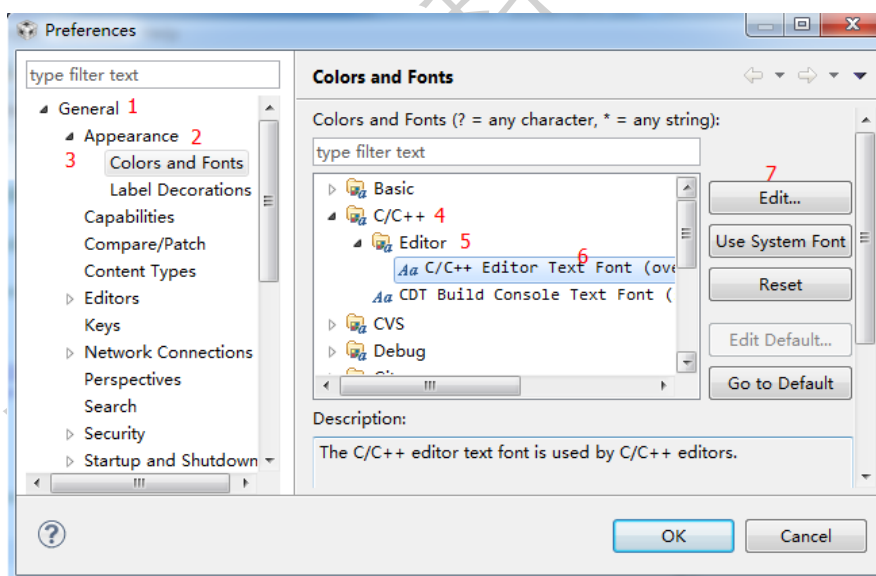


图 21

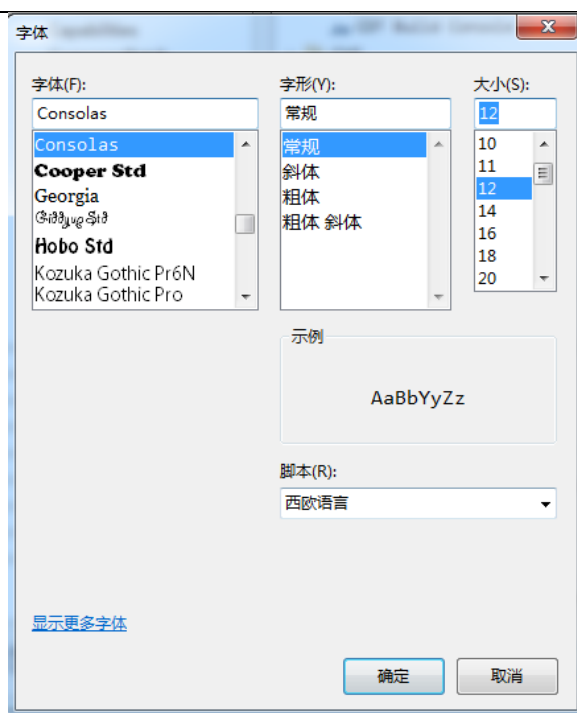


图 22

3.1 新建工程

点击 CCS5 菜单的"File->NEW->Project->Code Composer Studio->CCS Project", 弹出如下界面:

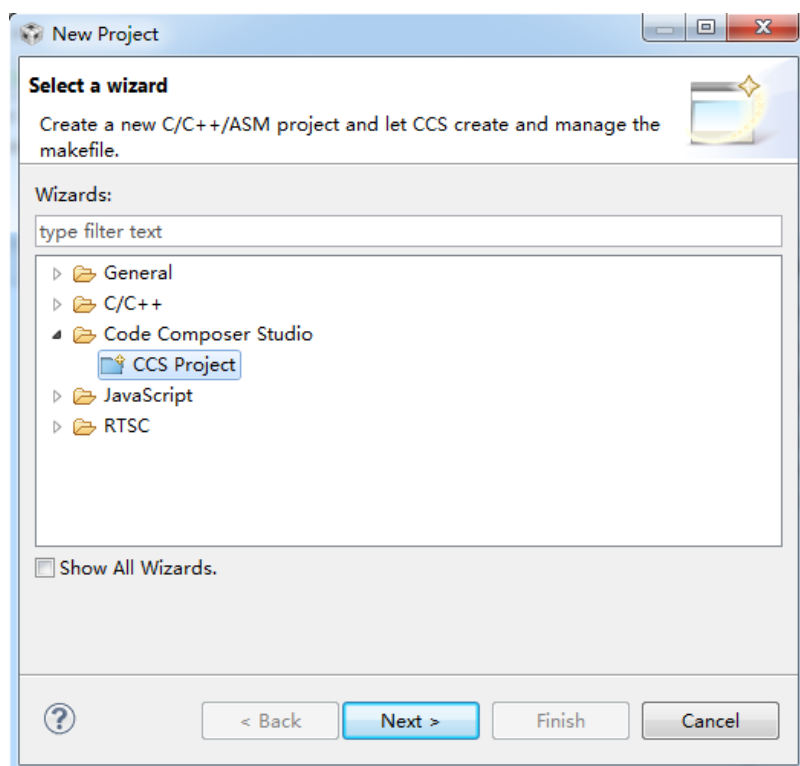


图 23

然后点击 Next，弹出对话框，并修改为如下：

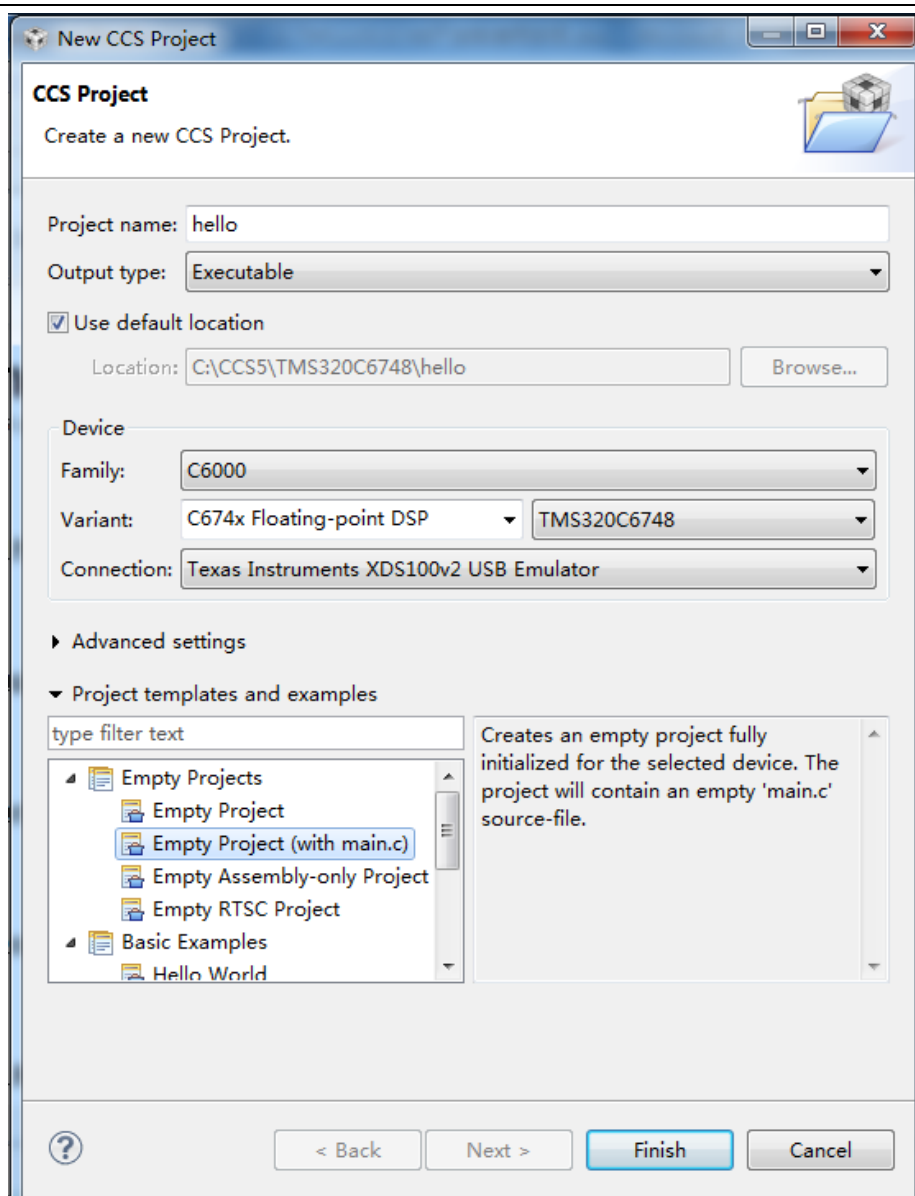


图 24

Project name: hello, CCS 工程名字;

Family: C6000;

Variant: C674x Floating-point DSP->TMS320C6748;

Connection:XDS100v2, 仿真器类型;

在"Empty Projects"下方选择"Empty Project(with main.c)", 点击 Finish 就可以完成工程新建了, 弹出如下界面:

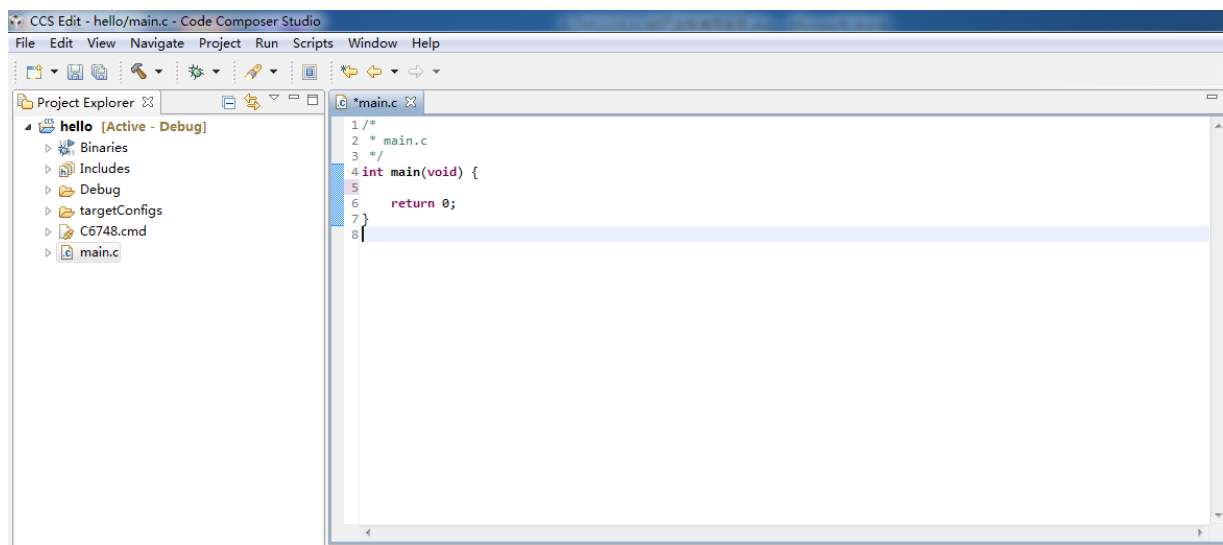


图 25

3.2 编写程序

编辑 main.c 文件，增加用户要实现的功能代码，例如增加打印信息：

```
#include<stdio.h>
```

```
int main(void) {
```

```
    printf("*****Enjoy your TL6748-EVM!.....www.tronlong.com*****\n");
```

```
    return 0;
```

```
}
```

编辑完成后如下图所示：



图 26

3.3 编译和运行 DSP 程序

右击工程，在弹出来的对话框中点击"Build Project"，即可对当前工程编译，如下图：

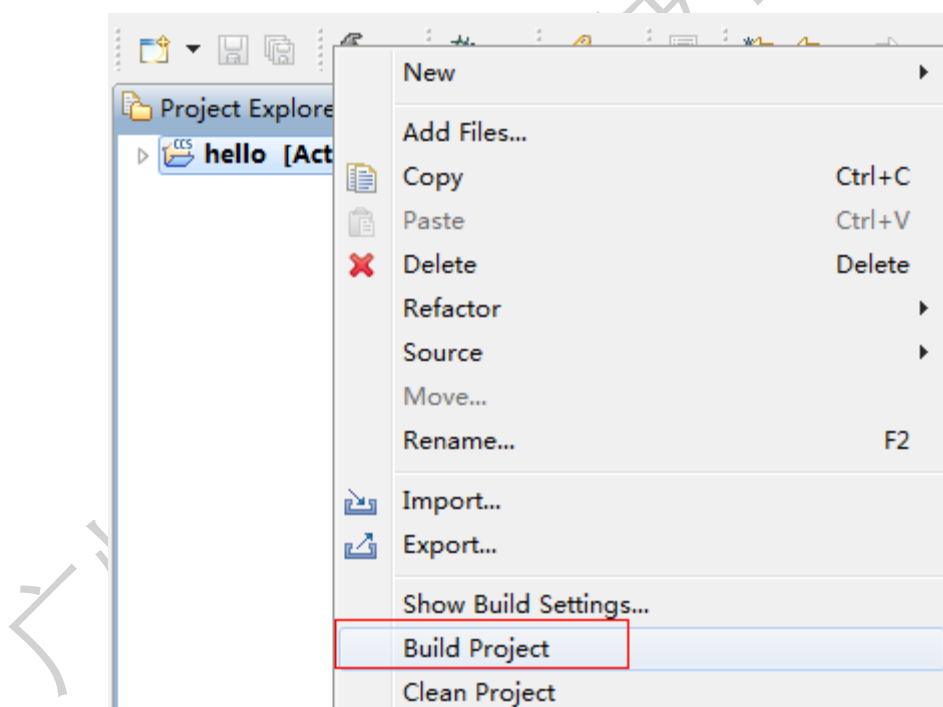


图 27

编译完毕后，可在左侧工程的 Binaries 下看到产生了可执行 hello.out 文件，如下图：

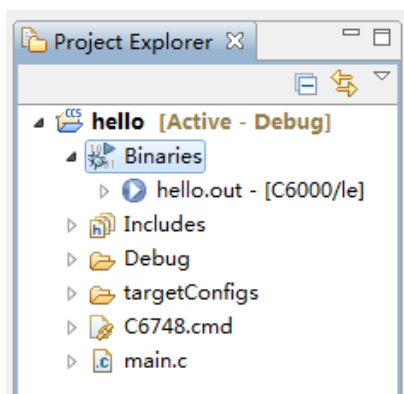


图 28

在程序加载前，先按照前面章节步骤连接开发板并用 GEL 文件初始化 DSP 核，再点击左上角的  load 键，选择要加载的 hello.out 文件。

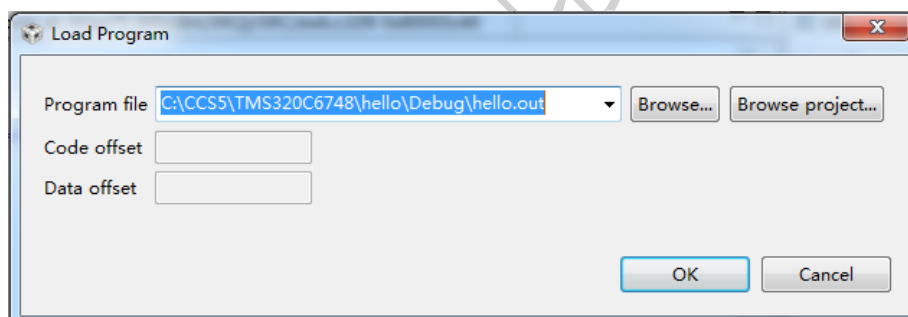


图 29

等待程序加载完成，然后点击  程序运行按钮，可以发现 CCS 的 Console 控制台有程序中指定的信息打印出来，至此最简单的 DSP 裸机程序开发成功。

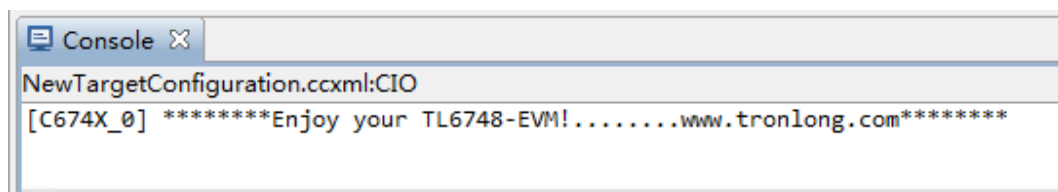


图 30

通过以上步骤，用户可以学习如何新建工程和在 CCS 下运行 DSP 程序了。

4 CCS 工程导入和编译步骤

- (1) 将光盘中 demo 目录拷贝到"C:\TMS320C6748"，用户可以基于 demo 目录下的现有工程直接编译和运行程序。点击 CCS 菜单"File->Import"，弹出以下对话框：

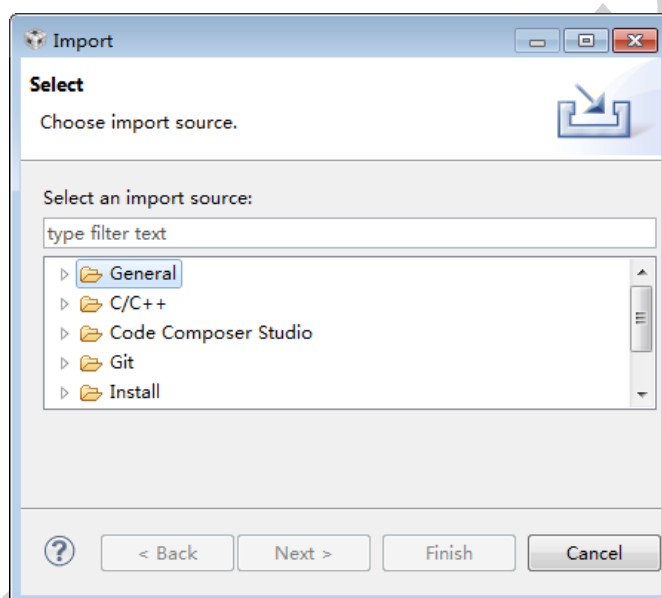


图 31

- (2) 点击"Code Composer Studio->Existing CCS Eclipse Projects"后再点击 Next，如下图所示：

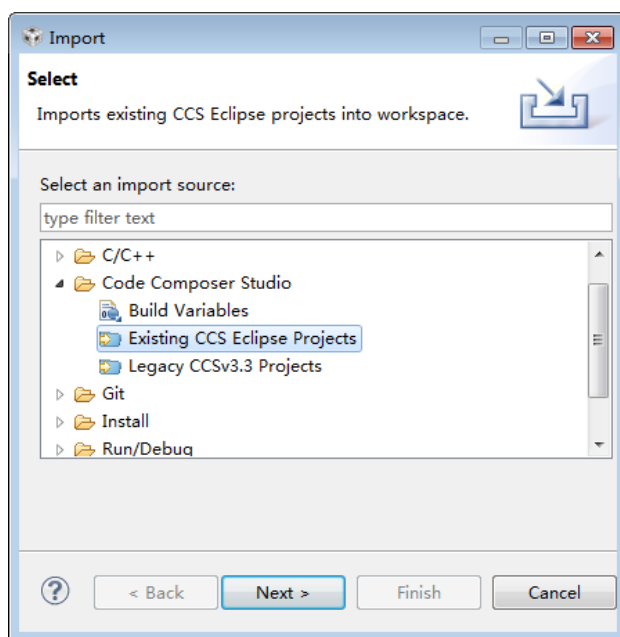


图 32

- (3) 在弹出的对话框的"Select search-directory"后面点击 Browse, 选择 LED 的 CCS 工程的存放路径" C:\TMS320C6748\demo\Application\GPIO_LED", 选择后如下图所示:

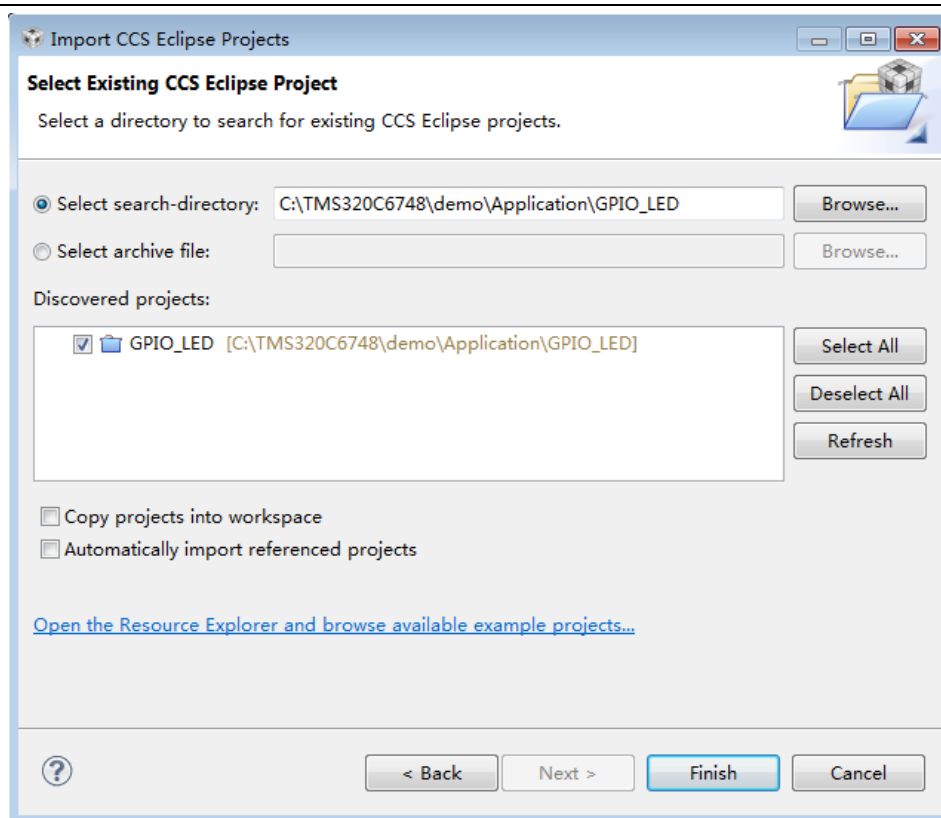


图 33

- (4) 点击 Finish 就可以导入现有 CCS 工程。然后右击工程，在弹出来的对话框中点击"Rebuild Project"，即可对当前工程编译，如下图：

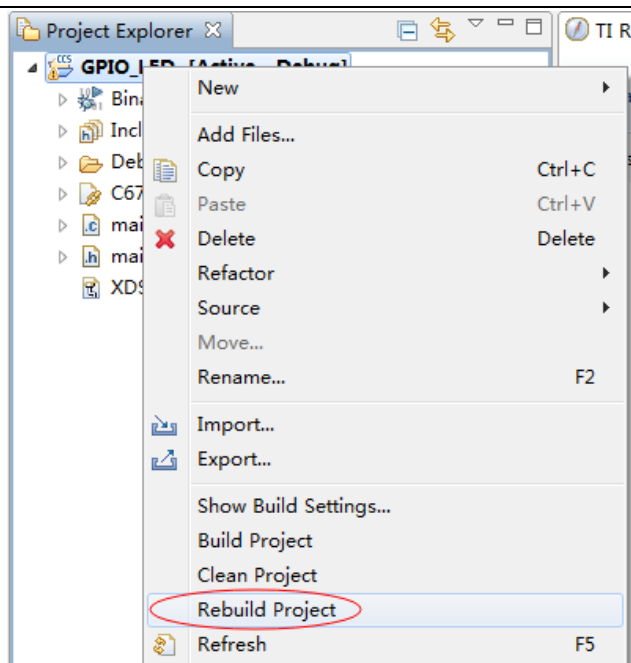


图 34

- (5) 编译完毕后,可在左侧工程的 Binaries 下看到产生了可执行 GPIO_LED.out 文件, 如下图:

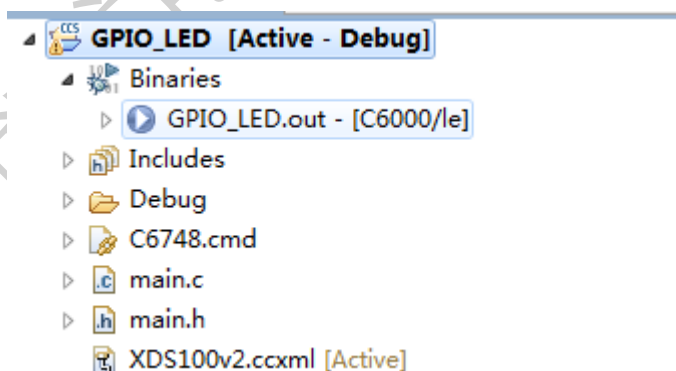


图 35

- (6) 开发板上电, 点击"Run->Debug", 出现 Debug 界面, 并发现 CPU 自动 Connect 了。

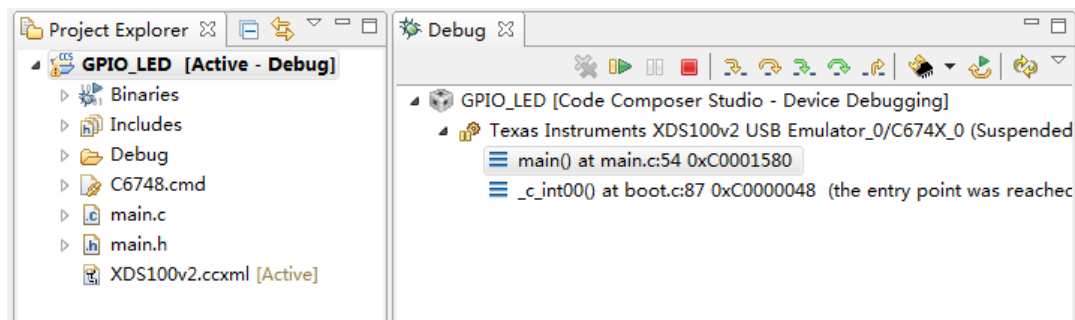


图 36

此处使用的是工程中的.ccxml 文件，在 CPU 自动连接后，点击脚本初始化 CPU，如下图所示：

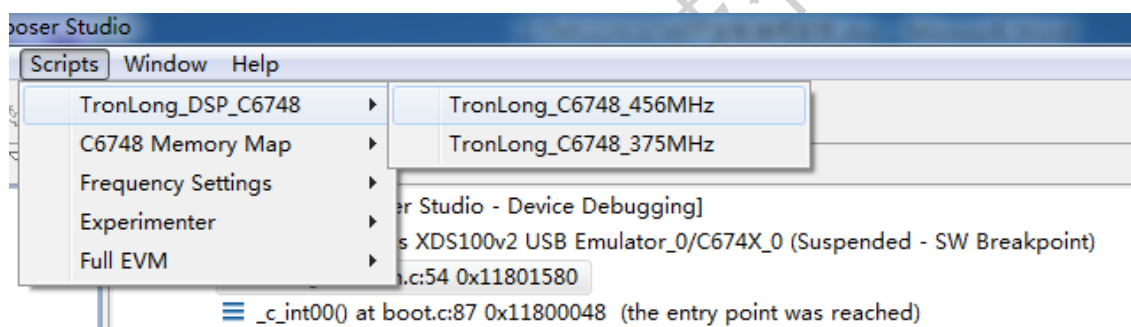


图 37

(7) 再点击左上角的  load 键，选择要加载的 GPIO_LED.out 文件，如下图所示：

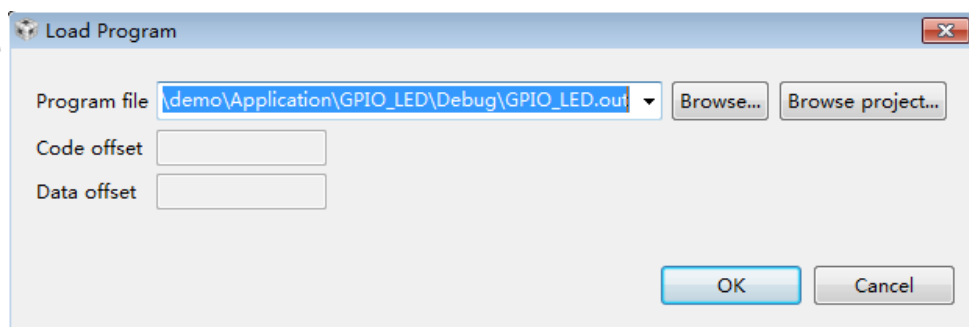


图 38

等待程序加载完成,然后单击程序运行按钮,可以发现底板的 LED 流水灯执行了。

单击黄色的暂停键,再按 F5 会进去函数里面,按 F6 就单步走。

5 Demo 例程演示

5.1 GPIO_LED

此程序的作用是实现 GPIO 输出功能,工程位于光盘 demo\Application\GPIO_LED 目录,按照工程导入步骤加载 GPIO_LED.out 文件,然后单击程序运行按钮。

演示现象

- (1) 核心板及底板 LED 循环点亮。

5.2 GPIO_KEY

此程序的作用是实现 GPIO 输入功能,工程位于光盘 demo\Application\GPIO_KEY 目录,按照工程导入步骤加载 GPIO_KEY.out 文件,然后单击程序运行按钮。

演示现象

- (1) 按下 SW6 按键将标志 Flag 置 1,核心板 LED 开始循环点亮。
- (2) 按下 SW5 按键将标志 Flag 置 0,核心板 LED 停止循环点亮。

5.3 TIMER

此程序的作用是实现定时器功能,此处使用的是定时器 2。DSP C6748 有 4 个定时器/计数器,均可配置为 64 位计数器、两个独立 32 位计数器及自动重装 32 位计数器,可以产生周期中断 DMA 事件及外部事件。定时器/计数器还可以用于捕获外部输入信号边缘并计数。此外,定时器 1 还可以用作 64 位看门狗计数器。

工程位于光盘 demo\Application\TIMER 目录,按照工程导入步骤加载 TIMER.out 文件,然后单击程序运行按钮。

演示现象

- (1) 核心板 LED 定时循环点亮。

时钟为 PLL1_SYSCLK2=300MHz，所以时间间隔为：

$$0x0FFFFFFF/(300000000/2)=1.7895697\text{ S}$$

5.4 UART1_POLL

此程序的作用是实现串口 1 轮询方式数据收发功能，工程位于光盘 demo\Application\UART1_POLL 目录，按照工程导入步骤加载 UART1_POLL.out 文件，将开发板的 UART1 和 PC 机连接，打开 ZOC 串口调试终端，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行按钮。

演示现象

- (1) ZOC 串口调试终端会打印提示信息，此信息是由 DSP 发送出来的，如下图所示：

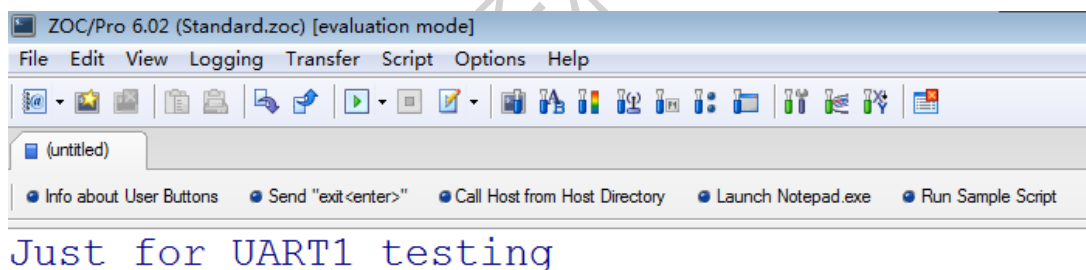


图 39

- (2) 使用键盘输入任意字符，DSP 会将接收到的字符回显到 ZOC 串口调试终端，如下图所示：

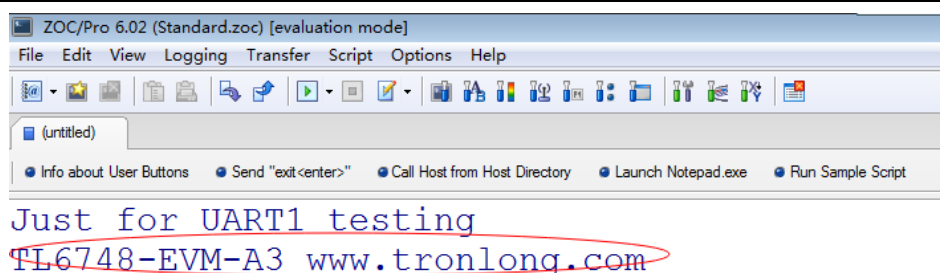


图 40

5.5 UART2_INT

此程序的作用是实现串口 2 中断方式数据收发功能，工程位于光盘 demo\Application\UART2_INT 目录，按照工程导入步骤加载 UART2_INT.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试终端，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) ZOC 串口调试终端会打印提示信息，此信息是由 DSP 发送出来的，如下图所示：

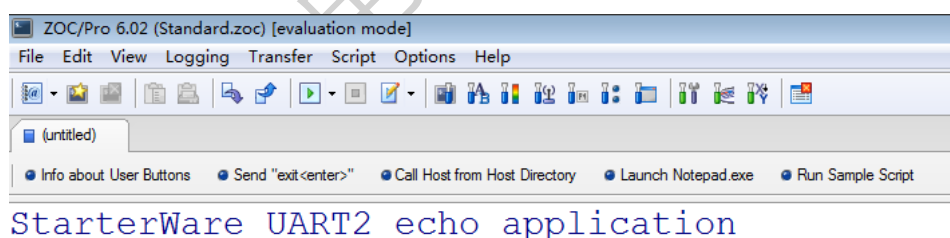


图 41

- (2) 使用键盘输入任意字符，DSP 会将接收到的字符回显到 ZOC 串口调试终端，如下图所示：

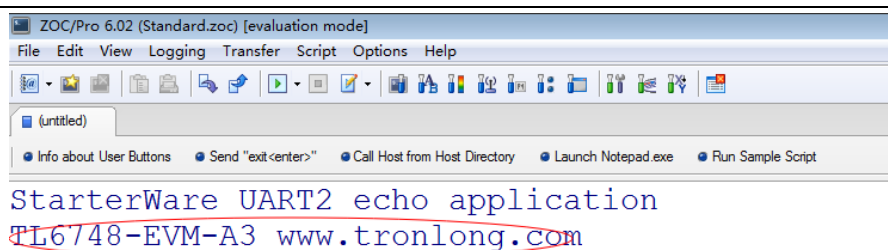


图 42

5.6 IIC_EEPROM

此程序的作用是实现 IIC EEPROM 设备的数据读写功能，工程位于光盘 demo\Application\IIC_EEPROM 目录，按照工程导入步骤加载 IIC_EEPROM.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) ZOC 串口调试终端会打印如下信息：

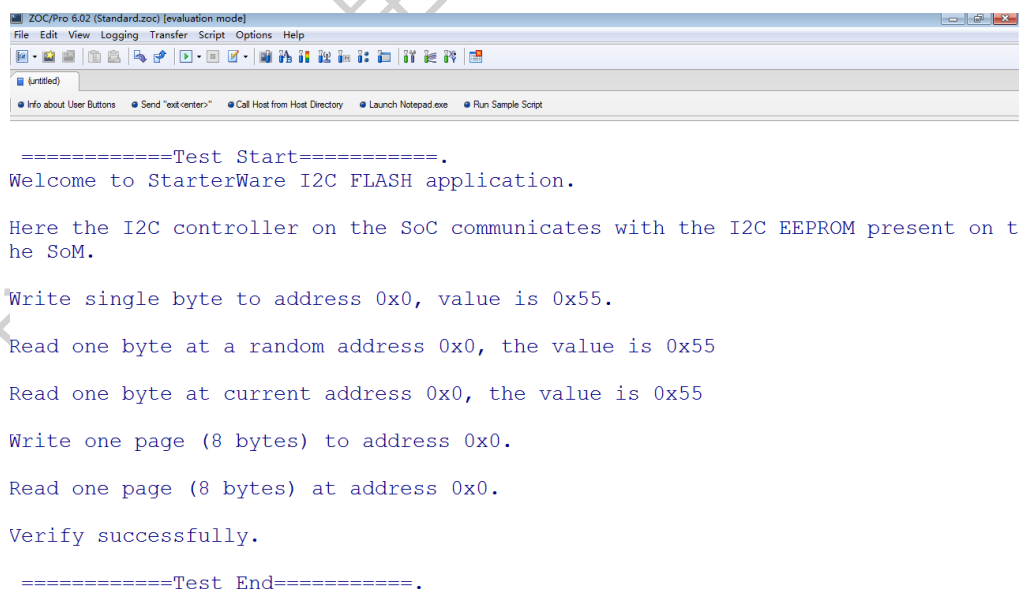


图 43

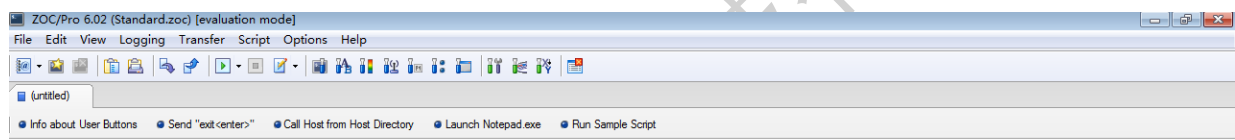
此程序先写入并读出一个字节数据，然后写入并读出一页数据，接着对比写入和读出的数据，根据结果判断 IIC EEPROM 设备读写是否成功。

5.7 SPI_FLASH

此程序的作用是实现 SPI FLASH 设备的数据读写功能，此设备挂载在 SPI1 总线，工程位于光盘 demo\Application\SPI_FLASH 目录，按照工程导入步骤加载 SPI_FLASH.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) ZOC 串口调试终端会打印如下信息：



```

=====Test Start=====.
Welcome to StarterWare SPI FLASH application.

Here the SPI controller on the SoC communicates with the SPI Flash present on the SoM.

Do you want to erase a sector of the flash before writing to it ?.
Input y(Y)/n(N) to proceed.
y
The data in the Flash and the one written to it are equal.
Verify successfully.

=====Test End=====.
```

图 44

此程序先提示是否擦除 SPI FLASH，输入 y 擦除 SPI FLASH 上的数据，然后对比写入和读出的数据，根据结果判断 SPI FLASH 设备读写是否成功。

5.8 WatchDog

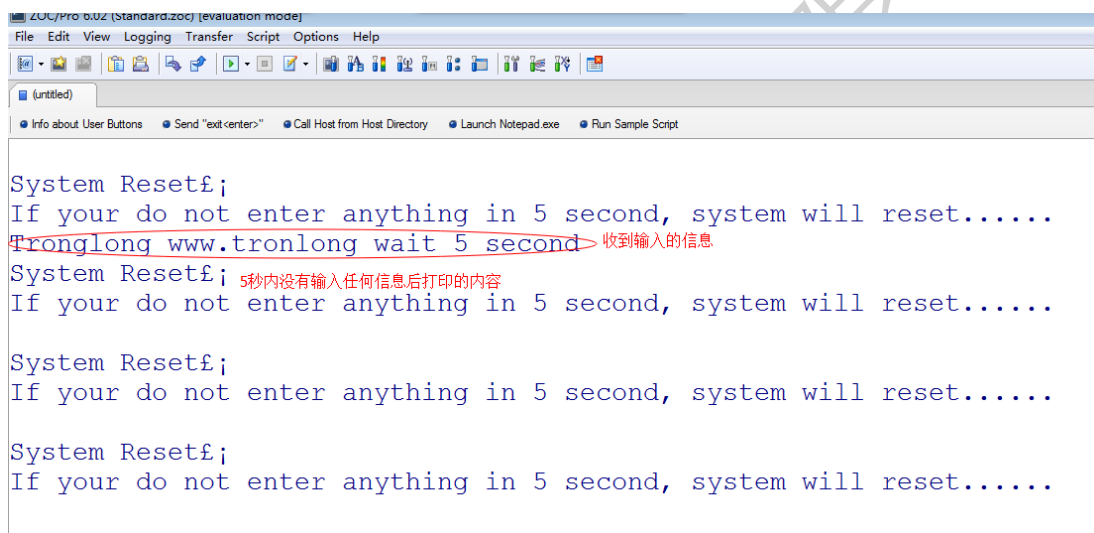
此程序的作用是实现看门狗功能，看门狗定时器 1 用于在程序运行过程中出现错误或陷入死循环等异常情况下复位程序。此程序设定看门狗超时时间为 5s。程序运行后在 5s

内输入任意字符，程序会持续运行，否则系统将复位。

工程位于光盘 demo\Application\WatchDog 目录，按照工程导入步骤加载 WatchDog.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) 串口打印提示信息，在 5s 内无任何信息输入复位将系统，重复打印提示信息。



```

System Resetf;
If your do not enter anything in 5 second, system will reset.....
Tronlong www.tronlong wait 5 second
System Resetf;
If your do not enter anything in 5 second, system will reset.....
System Resetf;
If your do not enter anything in 5 second, system will reset.....
System Resetf;
If your do not enter anything in 5 second, system will reset.....

```

图 45

5.9 NMI

此程序的作用是实现不可屏蔽中断功能。NMI (Non Maskable Interrupt)——不可屏蔽中断(即 CPU 不能屏蔽)，无论状态寄存器中 IF 位的状态如何，CPU 收到有效的 NMI 必须进行响应。工程位于光盘 demo\Application\NMI 目录，按照工程导入步骤加载 NMI.out 文件，然后点击程序运行  按钮。

演示现象

- (1) 每按一下不可屏蔽中断按键 (SW4)，CPU 就响应 NMI 去将核心板的 LED 轮流

点亮。

5.10 PWM

Coming soon.

5.11 ECAP

Coming soon.

5.12 RTC

此程序的作用是实现 RTC 计时功能，工程位于光盘 demo\Application\RTC 目录，按照工程导入步骤加载 RTC.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) ZOC 串口调试终端会打印如下信息，请依次设定小时、分钟、秒、日、月、年，程序就按照设定时间开始计时，并将实时时间打印出来，以此时间为例：17 点 23 分 00 秒，2014 年 6 月 7 日，星期六。

```
RTC Test Application.  
  
Enter the time in 24 hour format.  
Example (hh:mm:ss) 20:15:09  
  
Enter Hours:  
17  
Enter Minutes:  
23  
Enter Seconds:  
00
```

图 46

```
Enter the calendar information.
Example (DD:MM:YY) 31:03:73

Enter the day of the month:
07
Enter the month (Jan=01, Dec=12):
06
Enter the year (Ex: 2010=10, 1987=87:):
14
Enter the Day of the Week (Ex:Sun=00, Sat=06):
06

17:23:57 07-06-14 Sat
```

图 47

5.13 LCDC_GrLIB

此程序的作用是实现 LCD 显示功能，程序会调用 grLib 图形库显示各种图形元素，工程位于光盘 demo\Application\LCDC_GrLIB 目录，按照工程导入步骤加载 LCDC_GrLIB.out 文件，将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无，然后点击程序运行  按钮。

演示现象

- (1) ZOC 串口调试终端会打印如下信息：

```
=====Test Start=====
Welcome to StarterWare LCD Demo application.
Please notice the LCD display right now.

=====Test End=====
```

图 48

- (2) LCD 会显示图片。

更改显示图像方法

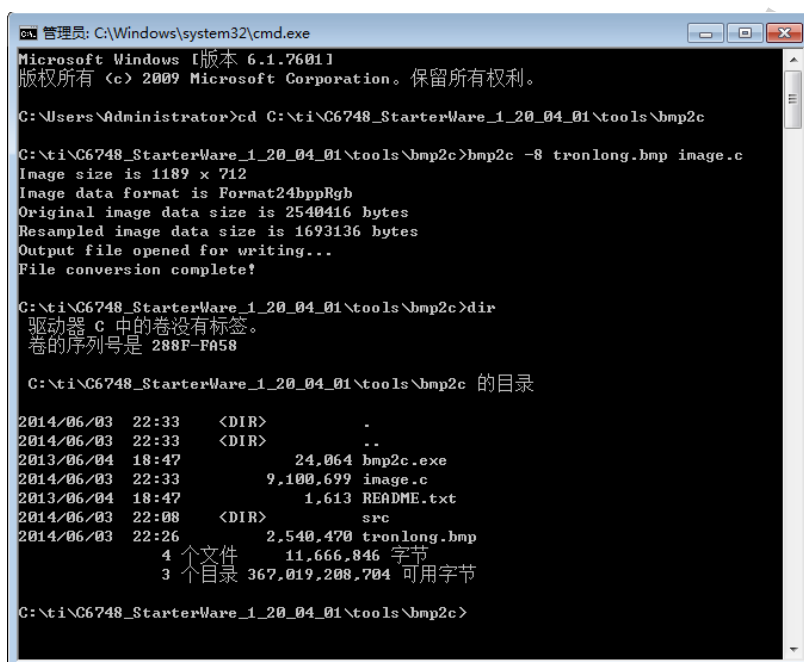
准备好 bmp 格式图形文件（例如 tronlong.bmp），将其拷贝到 StarterWare 安装路径

中的 tools\bmp2c 目录下，若 StarterWare 为默认的安装路径，此目录的完整路径为 C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c。

在 Windows 开始菜单底部输入 cmd 进入控制台窗口，进入此 bmp2c 目录并执行图片转化命令，会生成图形文件的字符数组 C 文件 image.c，命令如下：

DOS# cd C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c

DOS# bmp2c -8 tronlong.bmp image.c



```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>cd C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c

C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c>bmp2c -8 tronlong.bmp image.c
Image size is 1189 x 712
Image data format is Format24bppRgb
Original image data size is 2540416 bytes
Resampled image data size is 1693136 bytes
Output file opened for writing...
File conversion complete!

C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c>dir
驱动器 C 中的卷没有标签。
卷的序列号是 288F-F458

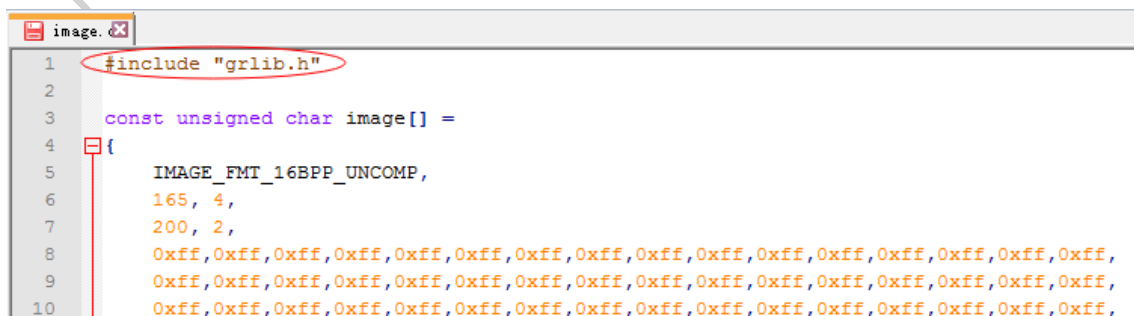
C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c 的目录

2014/06/03  22:33    <DIR>          .
2014/06/03  22:33    <DIR>          ..
2013/06/04  18:47             24,064  bmp2c.exe
2014/06/03  22:33             9,100,699  image.c
2013/06/04  18:47             1,613  README.txt
2014/06/03  22:08    <DIR>          src
2014/06/03  22:26             2,540,470  tronlong.bmp
               4 个文件      11,666,846 字节
               3 个目录    367,019,208,704 可用字节

C:\ti\C6748_StarterWare_1_20_04_01\tools\bmp2c>
  
```

图 49

打开 image.c，在该文件头加入头文件申明“#include "grlib.h"”，如下图所示：



```

1  #include "grlib.h"
2
3  const unsigned char image[] =
4  {
5      IMAGE_FMT_16BPP_UNCOMP,
6      165, 4,
7      200, 2,
8      0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
9      0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
10     0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
  
```

图 50



最后把 image.c 文件拷贝至 LCDC_GrLIB 工程根目录下覆盖原有图像文件 image.c, 重新编译并运行程序即可。建议图片不能太大, 否则会影响编译、加载、烧写的时间。

5.14 MMCSD

Coming soon.

5.15 SATA

Coming soon.

5.16 USB_DEV_BULK

此程序的作用是演示 USB 从方式, 通过 USB BULK 管道实现主机和 TMS320C6748 开发板的数据交换。

首先连接开发板的 USB OTG 和 PC 机 USB 接口, 接着在 PC 机中安装 USB 驱动程序, 具体驱动程序是光盘 demo\host_app\USB_DEV_BULK\driver 下的 installer_x64.exe 和 installer_x86.exe 两个文件。前者是 64 位 Windows 系统驱动, 后者是 32 位 Windows 系统驱动, 双击即可安装。(注意: 在 64 位 Win7 系统下验证过, 其他操作系统请用户自行验证。)

工程位于光盘 demo\Application\USB_DEV_BULK 目录, 按照工程导入步骤加载 USB_DEV_BULK.out 文件, 将开发板的 UART2 和 PC 机连接, 打开 ZOC 串口调试软件, 设置好波特率为 115200, 数据位 8 位, 停止位 1 位, 检验位无, 流控制无, 然后点击程序运行  按钮。

检查 PC 设备管理器中是否枚举到 USB 设备 General Bulk Device, 如下图所示。如果没有该设备请等待安装完成或者重新正确安装 USB 驱动程序。

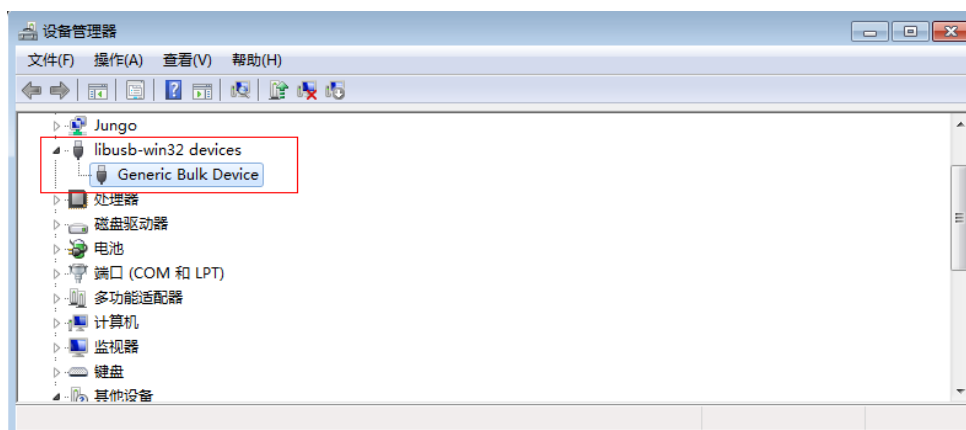


图 51

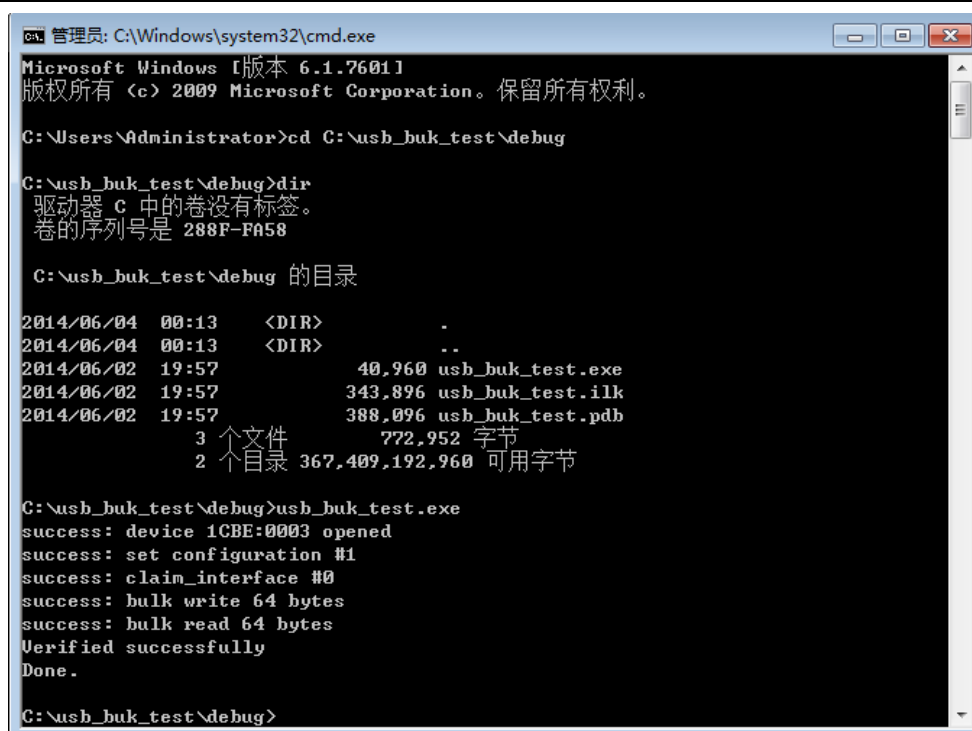
将 demo\host_app\USB_DEV_BULK 下的测试程序源码 usb_bulk_test 整个目录复制到工程 C 盘根目录下，在 Windows 开始菜单底部输入 cmd 进入控制台窗口，进入此 usb_bulk_test 下的 debug 目录并执行测试程序，命令如下：

```
DOS# cd C:\usb_buk_test\debug
```

```
DOS# usb_buk_test.exe
```

演示现象

- (1) PC 机控制台窗口显示读写数据成功信息。



```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>cd C:\usb_buk_test\debug

C:\usb_buk_test\debug>dir
驱动器 C 中的卷没有标签。
卷的序列号是 288F-FA58

C:\usb_buk_test\debug 的目录
2014/06/04 00:13 <DIR>      .
2014/06/04 00:13 <DIR>      ..
2014/06/02 19:57             40,960 usb_buk_test.exe
2014/06/02 19:57           343,896 usb_buk_test.ilc
2014/06/02 19:57           388,096 usb_buk_test.pdb
                3 个文件             772,952 字节
                2 个目录 367,409,192,960 可用字节

C:\usb_buk_test\debug>usb_buk_test.exe
success: device 1CBE:0003 opened
success: set configuration #1
success: claim_interface #0
success: bulk write 64 bytes
success: bulk read 64 bytes
Verified successfully
Done.

C:\usb_buk_test\debug>
  
```

图 52

(2) ZOC 窗口输出提示信息。

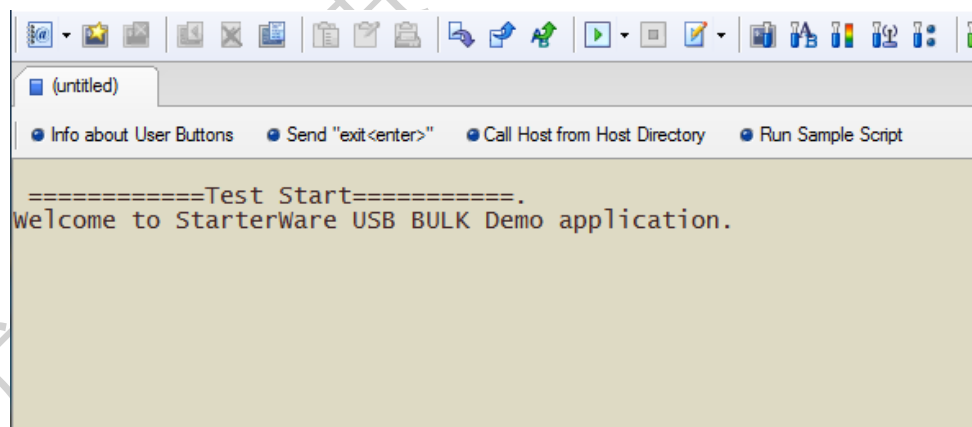


图 53

(3) 屏幕提示开发板和 PC 通信成功，并显示已成功传输字节数量。

UI 程序演示

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

41/60

为了方便使用，TMS320C6748 开发板的光盘中提供了一个 UI 程序。

源码和可执行程序分别位于如下路径：

demo\host_app\USB_DEV_BULK\bulk_test_ui

demo\host_app\USB_DEV_BULK\bulk_test_ui\debug\bulk_test_ui.exe

双击可执行程序 bulk_test_ui.exe，弹出如下界面：



图 54

点击“打开设备”，然后点击“读写设备”，界面显示如下：



图 55

UI 程序将十六进制数据 00~3f 共 64 个数据通过 USB 送往开发板, 然后再读取回来, 显示在 UI 界面上。

USB Tracer 捕获工具的使用

为了更清楚的是看到 USB 数据的传输, 可以使用 USBTrace 工具。USBTrace 是一款功能强大的 USB 监控软件, 它能够分析 USB 总线和与之相连的 USB 设备的连接状况, 监控包括 USB 控制器, USB 集线器和 USB 设备的连接, 请求和运行情况。具备 USB 包抓取, USB 设备类别分析等功能, 适合检测主机方的 USB 控制协议。

该工具在 demo\host_app\tools\USBTrace_Installer 目录下, 先安装原程序, 然后将 Crack 目录下的所有内容覆盖到安装目录中。

USBTrace_x64_Build_77_220512_Installer: 64 位

USBTrace_x86_Build_77_220512_Installer: 32 位

确保开发板正常启动、USB 驱动已正常安装、UI 正常启动, 然后点击桌面的 USBTrace 图标, 弹出如下界面:

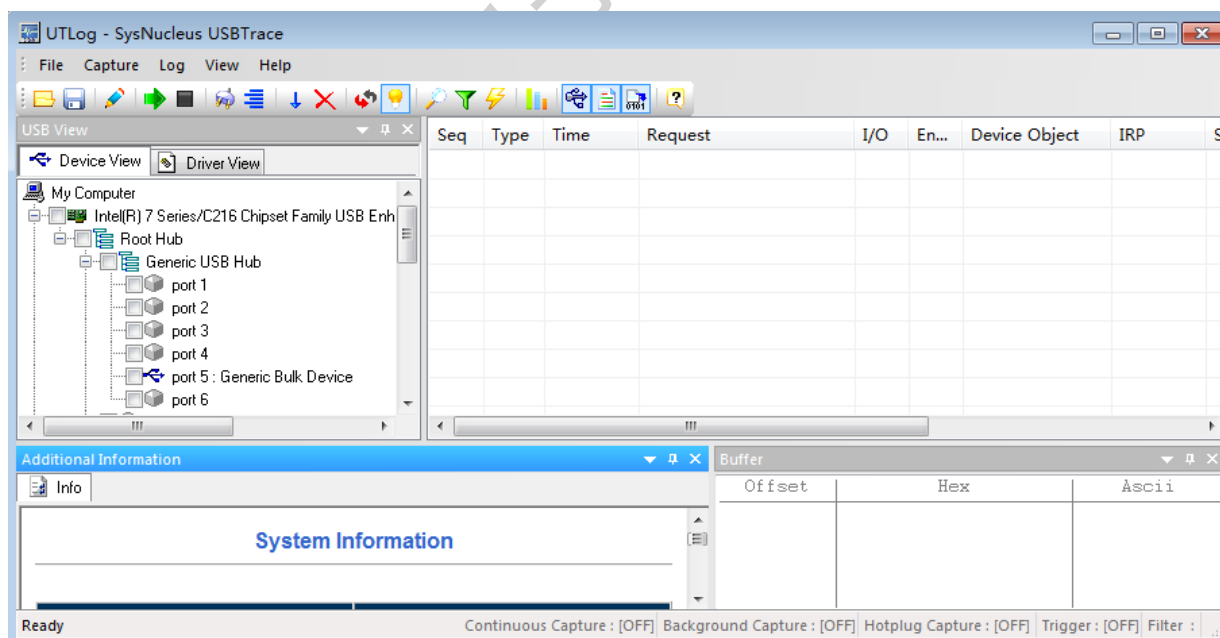


图 56

选择需要捕获的 USB 设备 “Generic Bulk Device”，如下图所示：

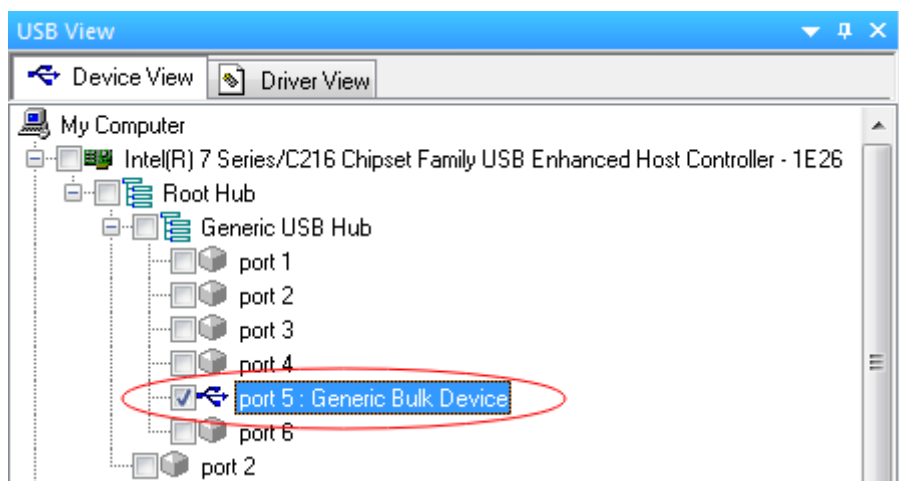


图 57

设置过滤器并开始捕获，点击步骤按照 1->2->3->4->5，如下图所示：

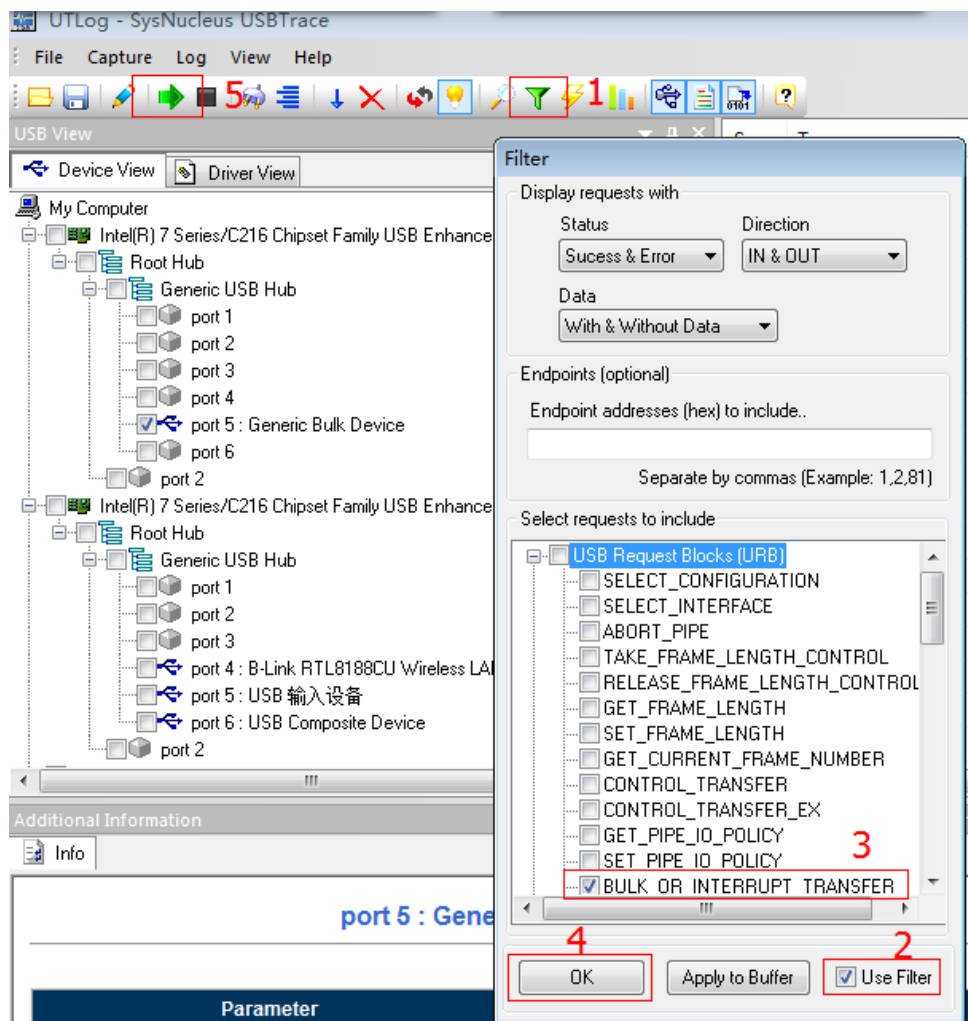


图 58

然后点击 bulk_test_ui 界面的“读写设备”按钮，可发现 USBTrace 右边显示如下图：

Seq	Type	Time	Request	I/O	End...	Device Object	IRP	Status	Buffer Snippet	Buffer Size
0	START	0.000000	START OF LOG							
1	URB	3.841728	BULK_OR_INTERRUPT_T...	OUT	1	\Device\USBPD...	0xBA96...	STATUS_SUCCESS	00 01 02 03 04 05 ...	64
2	URB	3.841729	BULK_OR_INTERRUPT_T...	OUT	1	\Device\libusb...	0xBA96...	STATUS_SUCCESS	00 01 02 03 04 05 ...	64
3	URB	3.841745	BULK_OR_INTERRUPT_T...	IN	1	\Device\libusb...	0xBA96...	STATUS_PENDING	PC往USB输出的信息	
4	URB	3.841810	BULK_OR_INTERRUPT_T...	IN	1	\Device\libusb...	0xBA96...	STATUS_SUCCESS		0
5	URB	3.842208	BULK_OR_INTERRUPT_T...	OUT	81	\Device\USBPD...	0xBA96...	STATUS_SUCCESS		0
6	URB	3.842210	BULK_OR_INTERRUPT_T...	OUT	81	\Device\libusb...	0xBA96...	STATUS_SUCCESS		0
7	URB	3.842226	BULK_OR_INTERRUPT_T...	IN	81	\Device\libusb...	0xBA96...	STATUS_PENDING	从USB输入到PC的信息	
8	URB	3.842306	BULK_OR_INTERRUPT_T...	IN	81	\Device\libusb...	0xBA96...	STATUS_SUCCESS	00 01 02 03 04 05 ...	64

图 59

点击 OUT 或者 IN 行即可在下方看到发送或者接收到的数据，如下图所示：

Offset	Hex	Ascii
00000000	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E	
0000000F	0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D	
0000001E	1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C	..!"#\$%&'()*+,-./0123456789:;<=>?
0000002D	2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B	
0000003C	3C 3D 3E 3F	

图 60

5.17 ENET_HTTPD

此程序的作用是实现网络通信功能,在 TMS320C6748 开发板上搭建一个轻量级 WEB 服务器。连接接好网线（**注意**: 接路由器用直连网线,直连 PC 则用交叉网线），将开发板的 UART2 和 PC 机连接,打开 ZOC 串口调试软件,设置好波特率为 115200,数据位 8 位,停止位 1 位,检验位无,流控制无。工程位于光盘 demo\Application\ENET_HTTPD 目录,按照工程导入步骤加载 ENET_HTTPD.out 文件,然后点击程序运行  按钮。

演示现象

- (1) 串口显示开发板的 IP 地址 (192.168.1.200)。

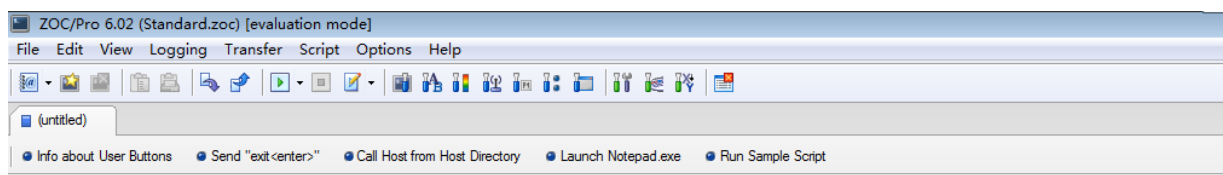


图 61

- (2) 在 Windows 开始菜单底部输入 cmd 进入控制台窗口，执行 ping 命令，确认开发板网络部分工作正常。

DOS# ping 192.168.1.200

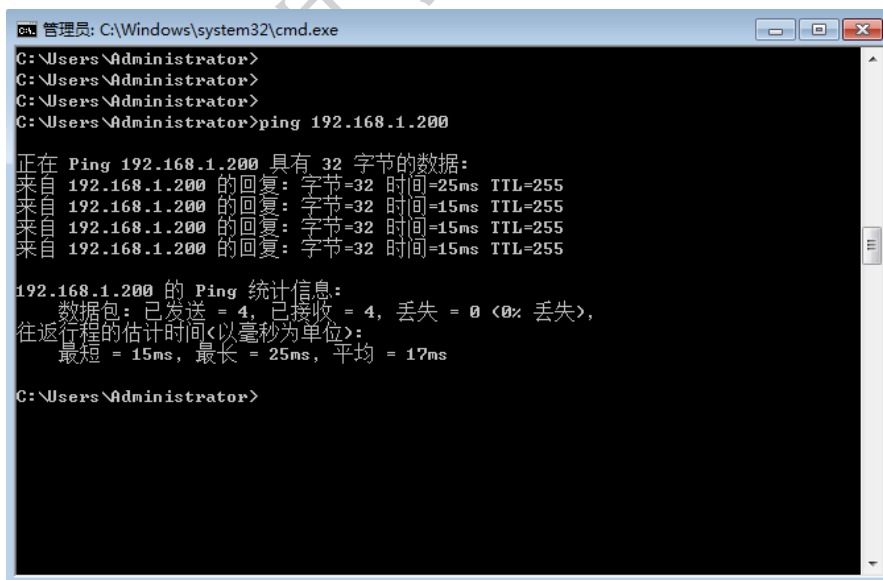


图 62

- (3) 打开浏览器输入网址 <http://192.168.1.200>，就可以看到网址内容。

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

47/60

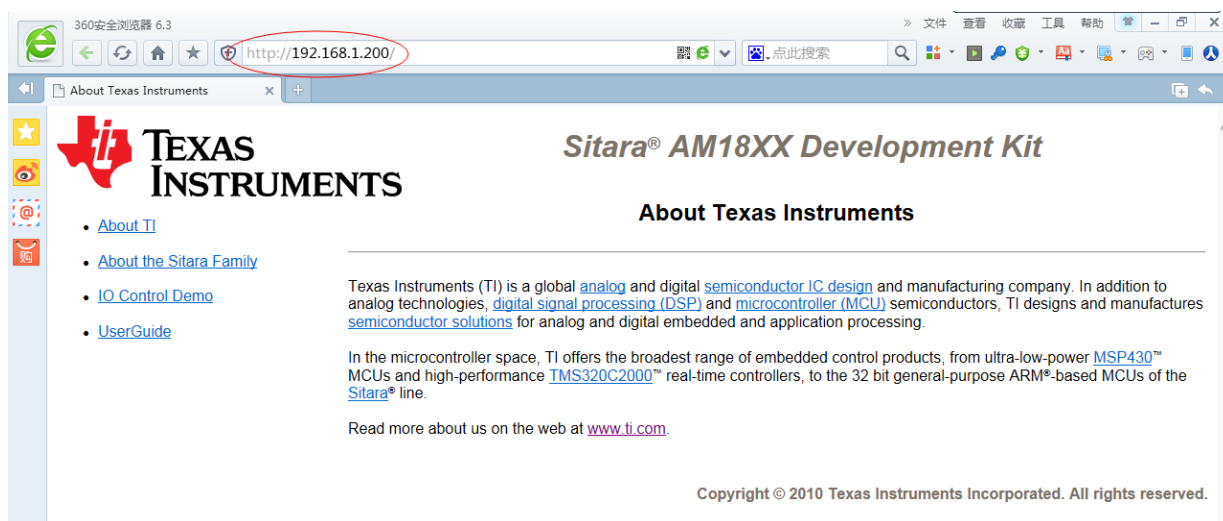


图 63

5.18 ENET_ECHO

本程序主要演示 Socket 通讯，开发板为服务器，PC 机为客户端，两者实现数据交换。

首先连接好网线，连接接好网线（**注意：**接路由器用直连网线，直连 PC 则用交叉网线），将开发板的 UART2 和 PC 机连接，打开 ZOC 串口调试软件，设置好波特率为 115200，数据位 8 位，停止位 1 位，检验位无，流控制无。

工程位于光盘 demo\Application\ENET_ECHO 目录，按照工程导入步骤加载 ENET_ECHO.out 文件，然后点击程序运行  按钮。

演示现象

- (1) 串口显示开发板的 IP 地址（192.168.1.200）和端口（2000），如下图所示：

```
=====Test Start=====.  
Welcome to StarterWare ENET_ECHO application.  
  
TL6748-EVM IP Address Assigned :192.168.1.200  
TCP Port is 2000.  
  
Please run the client app to verify the 'send' and 'recv' action.  
  
=====Test END=====.
```

图 64

UI 程序演示

首先在 Windows 开始菜单底部输入 cmd 进入控制台窗口，执行 ping 命令，确认开发板网络部分工作正常。

DOS# ping 192.168.1.200



图 65

接着双击 demo\host_app\ENET_ECHO\enet_client\debug\enet_client.exe 即可运行 UI 程序，弹出如下界面：



图 66

UI 程序可以选择单次发收，也可以选择多次发收，发送的数据内容用户可以在程序中自行修改，程序会对发送和接收的内容进行比对校验。点击“建立连接”后再点击“读写测试”，弹出如下界面：

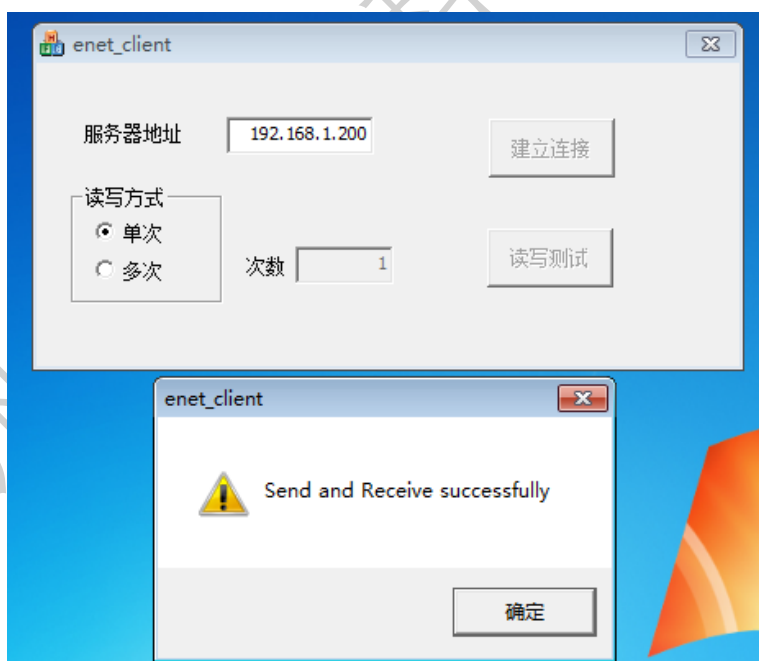


图 67

Wireshark 捕获工具的使用

Wireshark 是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。

Wireshark 软件的路径位于 demo\host_app\tools\Wireshark-win32-1.10.3.exe，请双击安装此软件。正确安装后打开 Wireshark 软件，如下图所示：

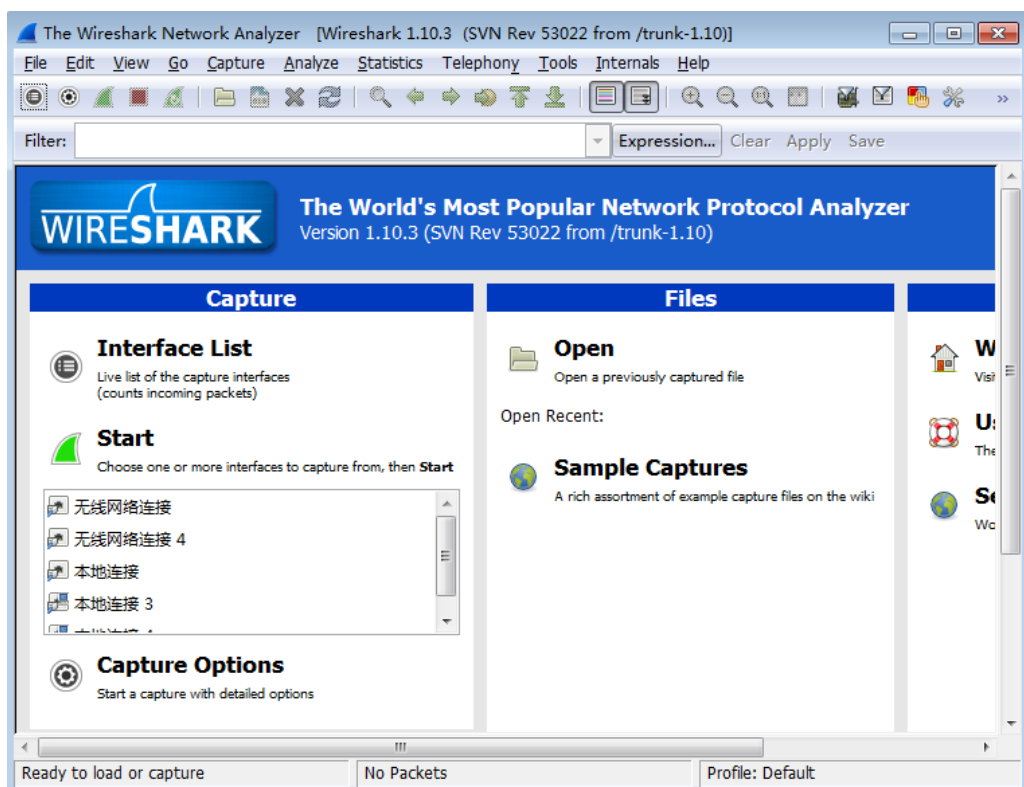


图 68

接着按下图步骤 1->2->3 点击，如下图所示：

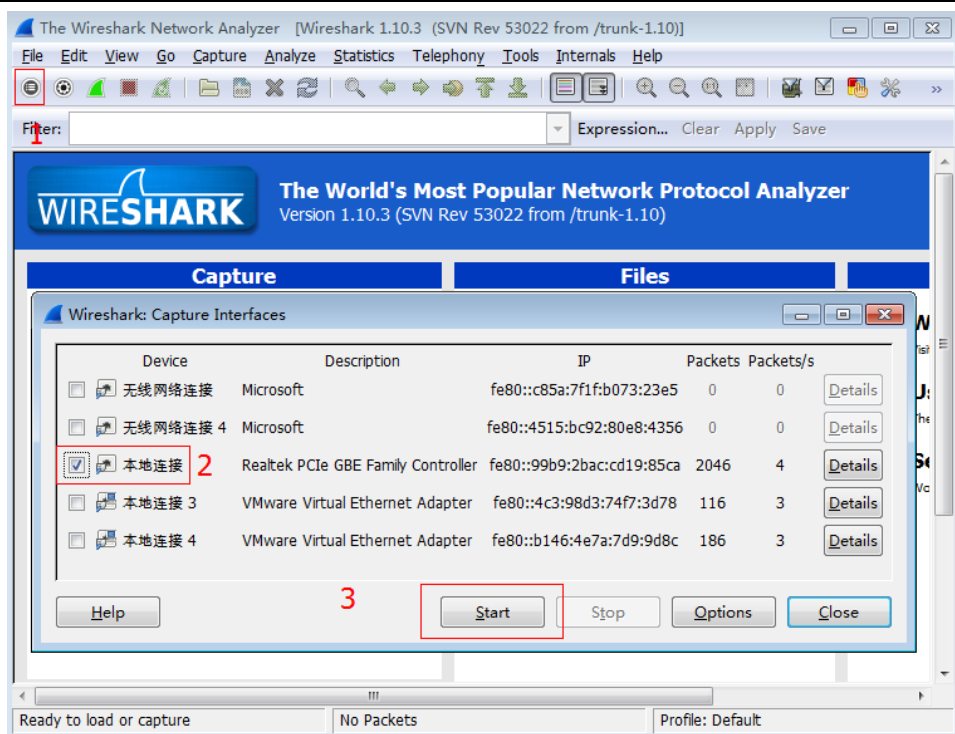


图 69

点击 Start 后，弹出如下界面：

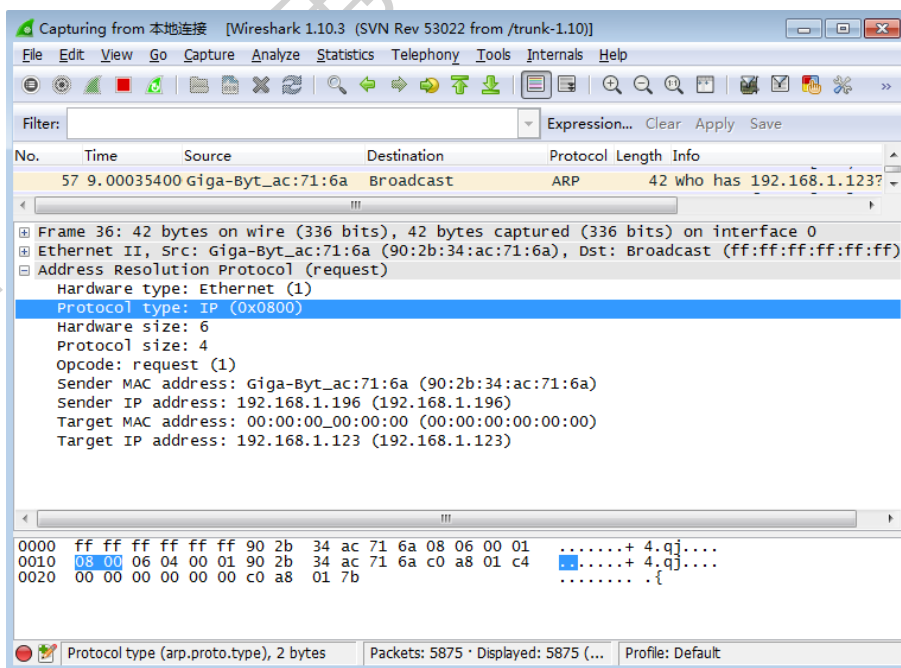


图 70

然后按下图步骤 1->2->3 点击并在步骤 2 中设置需要监控的 IP 地址为 192.168.1.200, 如下图所示:

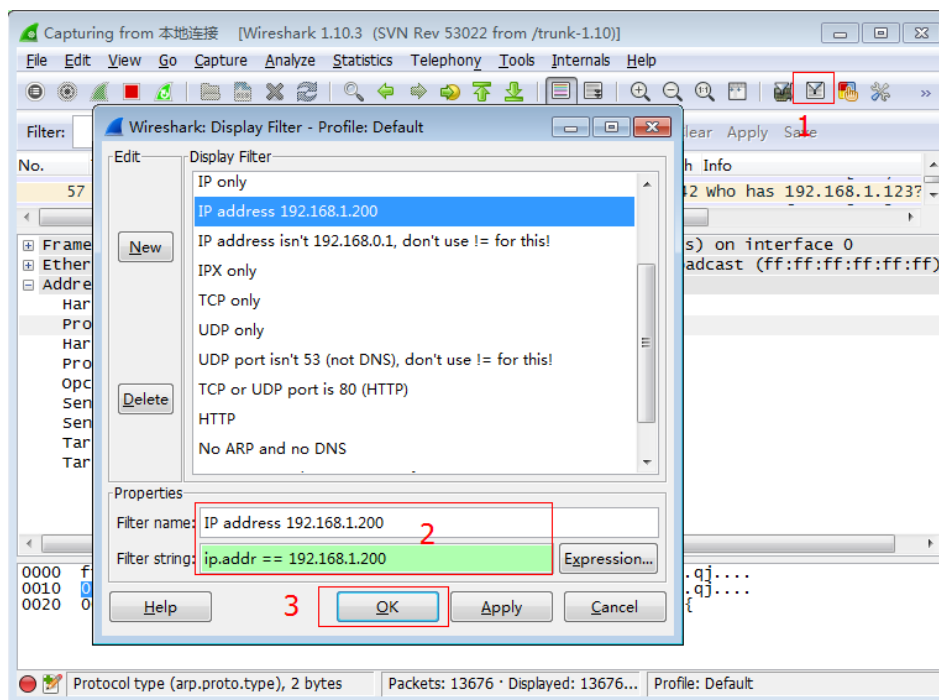


图 71

点击 OK 后弹出如下界面:

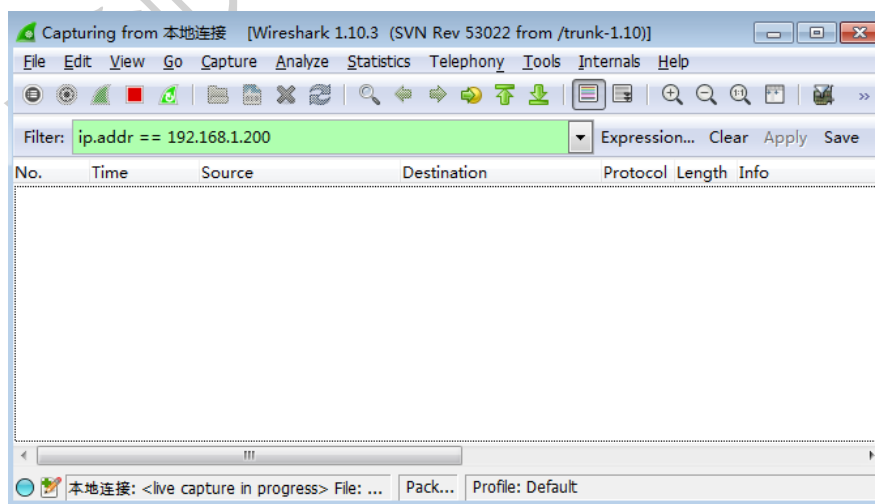


图 72

然后点击 UI 程序中的“读写测试”按钮或者在 PC 中执行“ping 192.168.1.200”命令，Wireshark 会打印监听到的信息，如下图所示：

No.	Time	Source	Destination	Protocol	Length	Info
883	167.219035	192.168.1.196	192.168.1.200	TCP	1514	58667 > cisco-sccp [ACK] Seq=1461 Ack=1 Win=63064 Len=1460
884	167.219055	192.168.1.196	192.168.1.200	TCP	1514	58667 > cisco-sccp [ACK] Seq=1461 Ack=1 Win=63064 Len=1460
885	167.219065	192.168.1.196	192.168.1.200	TCP	1230	58667 > cisco-sccp [ACK] Seq=2921 Ack=1 Win=63064 Len=1176
886	167.225560	192.168.1.200	192.168.1.196	TCP	1514	cisco-sccp > 58667 [ACK] Seq=1 Ack=1461 Win=6000 Len=1460
887	167.231593	192.168.1.200	192.168.1.196	TCP	1514	cisco-sccp > 58667 [PSH, ACK] Seq=1461 Ack=2921 Win=64240 Len=1460
888	167.231621	192.168.1.196	192.168.1.200	TCP	54	58667 > cisco-sccp [ACK] Seq=4097 Ack=2921 Win=64240 Len=0
889	167.236852	192.168.1.200	192.168.1.196	TCP	1230	cisco-sccp > 58667 [PSH, ACK] Seq=2921 Ack=4097 Win=1176 Len=1176
890	167.435065	192.168.1.196	192.168.1.200	TCP	54	58667 > cisco-sccp [ACK] Seq=4097 Ack=4097 Win=63064 Len=0
949	182.005789	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5740/27670, ttl=64
950	182.006580	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5740/27670, ttl=255 (request in 949)
955	183.006812	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5742/28182, ttl=64 (reply in 956)
956	183.007605	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5742/28182, ttl=255 (request in 955)
975	184.008844	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5743/28438, ttl=64 (reply in 976)
976	184.009651	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5743/28438, ttl=255 (request in 975)
978	185.009863	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5745/28950, ttl=64 (reply in 979)
979	185.010668	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5745/28950, ttl=255 (request in 978)

图 73

No.	Time	Source	Destination	Protocol	Length	Info
883	167.219035	192.168.1.196	192.168.1.200	TCP	1514	58667 > cisco-sccp [ACK] Seq=1461 Ack=1 Win=63064 Len=1460
884	167.219055	192.168.1.196	192.168.1.200	TCP	1514	58667 > cisco-sccp [ACK] Seq=1461 Ack=1 Win=63064 Len=1460
885	167.219065	192.168.1.196	192.168.1.200	TCP	1230	58667 > cisco-sccp [ACK] Seq=2921 Ack=1 Win=63064 Len=1176
886	167.225560	192.168.1.200	192.168.1.196	TCP	1514	cisco-sccp > 58667 [ACK] Seq=1 Ack=1461 Win=6000 Len=1460
887	167.231593	192.168.1.200	192.168.1.196	TCP	1514	cisco-sccp > 58667 [PSH, ACK] Seq=1461 Ack=2921 Win=64240 Len=1460
888	167.231621	192.168.1.196	192.168.1.200	TCP	54	58667 > cisco-sccp [ACK] Seq=4097 Ack=2921 Win=64240 Len=0
889	167.236852	192.168.1.200	192.168.1.196	TCP	1230	cisco-sccp > 58667 [PSH, ACK] Seq=2921 Ack=4097 Win=1176 Len=1176
890	167.435065	192.168.1.196	192.168.1.200	TCP	54	58667 > cisco-sccp [ACK] Seq=4097 Ack=4097 Win=63064 Len=0
949	182.005789	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5740/27670, ttl=64
950	182.006580	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5740/27670, ttl=255 (request in 949)
955	183.006812	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5742/28182, ttl=64 (reply in 956)
956	183.007605	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5742/28182, ttl=255 (request in 955)
975	184.008844	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5743/28438, ttl=64 (reply in 976)
976	184.009651	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5743/28438, ttl=255 (request in 975)
978	185.009863	192.168.1.196	192.168.1.200	ICMP	74	Echo (ping) request id=0x0100, seq=5745/28950, ttl=64 (reply in 979)
979	185.010668	192.168.1.200	192.168.1.196	ICMP	74	Echo (ping) reply id=0x0100, seq=5745/28950, ttl=255 (request in 978)

0000	00 08 ee 03 6a c6 90 2b 34 ac 71 6a 08 00 45 00	j..+ 4.qj..E.
0010	05 dc 28 09 40 00 40 06 88 36 c0 a8 01 c4 c0 a8	..(.@.@..6.....
0020	01 c8 e5 2b 07 d0 ac 5e 31 f8 00 00 29 f6 50 10	...+...^ 1....oP.
0030	f6 58 49 5b 00 00 00 01 02 03 04 05 06 07 08 09	..XI[.....
0040	0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19	
0050	1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29	

图 74



5.19 AUDIO_PLAY

此程序的作用是实现 Line out 音频播放功能,主要演示如何设置 MCASP 和 AIC 芯片来播放一段原始音频数据。工程位于光盘 demo\Application\AUDIO_PLAY 目录,按照工程导入步骤加载 AUDIO_PLAY.out 文件,将开发板的 UART2 和 PC 机连接,打开 ZOC 串口调试软件,设置好波特率为 115200,数据位 8 位,停止位 1 位,检验位无,流控制无,然后点击程序运行  按钮,在 Line out 接口插上耳机。

演示现象

- (1) ZOC 串口调试终端会打印如下信息:

```
=====Test Start=====.  
Welcome to StarterWare Audio_Play Demo application.  
  
Please insert microphone to the 'LINE OUT' port of TL6748 EVM.  
  
=====Test End=====.
```

图 75

- (2) 耳机重复播放一段声音。

5.20 MCBSP

Coming soon.

5.21 FFT

此程序的作用是实现 FFT 处理功能,工程位于光盘 demo\Application 目录,按照工程导入步骤加载 FFT.out 文件,然后点击程序运行  按钮。

CCS 支持绘制多种类型的图表 时域图、频域图、瀑布图等等。待测试信号为原始信号做 1K 采样频率进行 1024 个点抽样得到。

信号源公式: $5 * \sin(2 * \pi * 150 * t) + 15 * \sin(2 * \pi * 350 * t)$

该信号含有 150Hz 和 350Hz 两种频率分量。

演示步骤现象

点击"Tools->Graph->Single Time"选择单时域信号图，如下图所示：

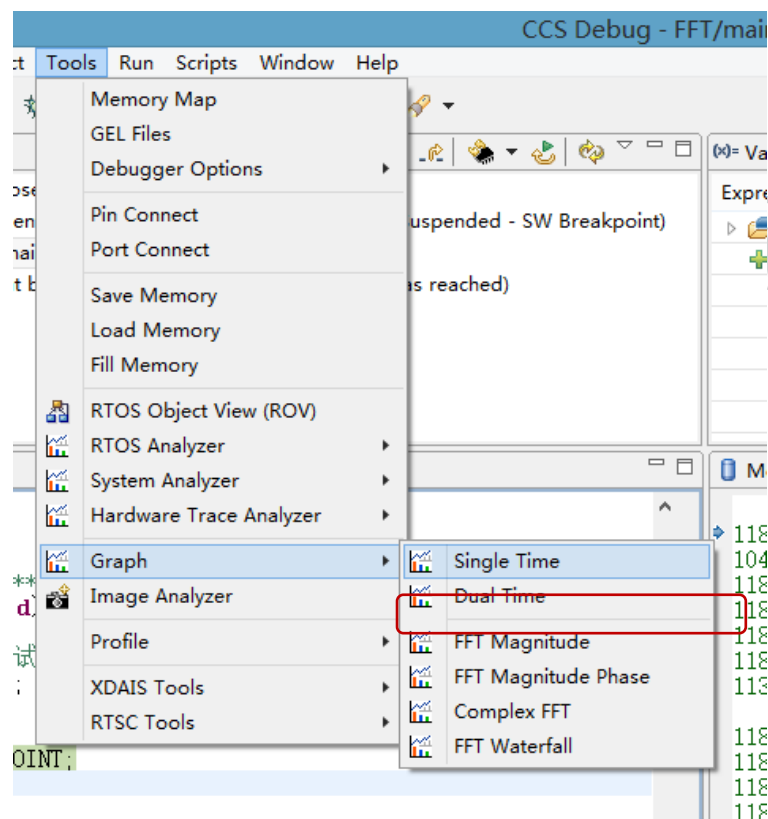


图 76

按照下图进行设置。

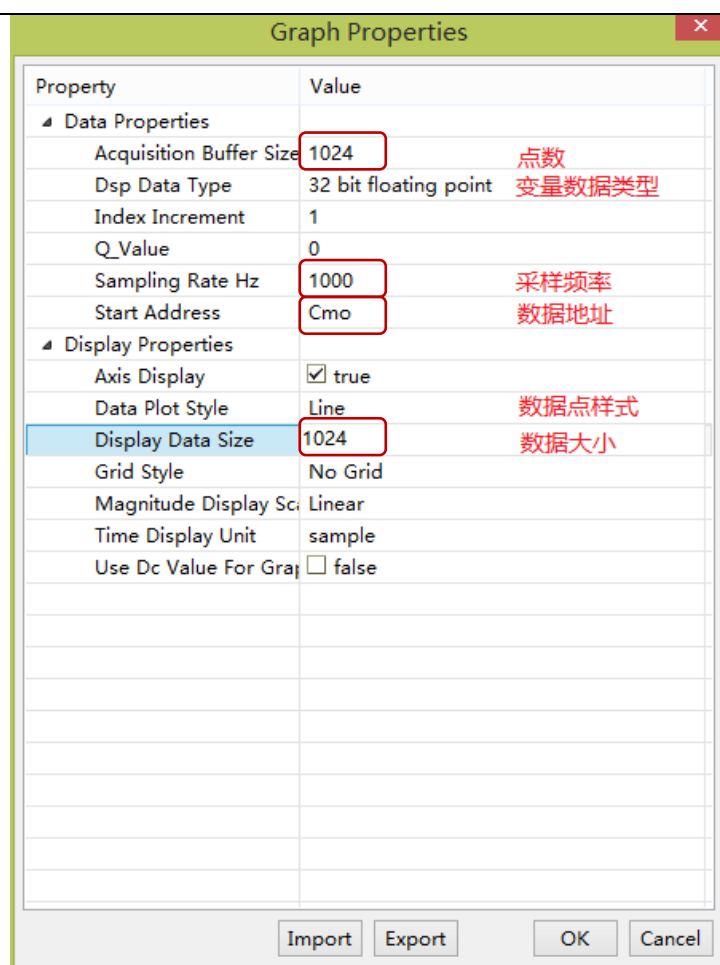


图 77

点击 OK，CCS 界面底部弹出如下界面：

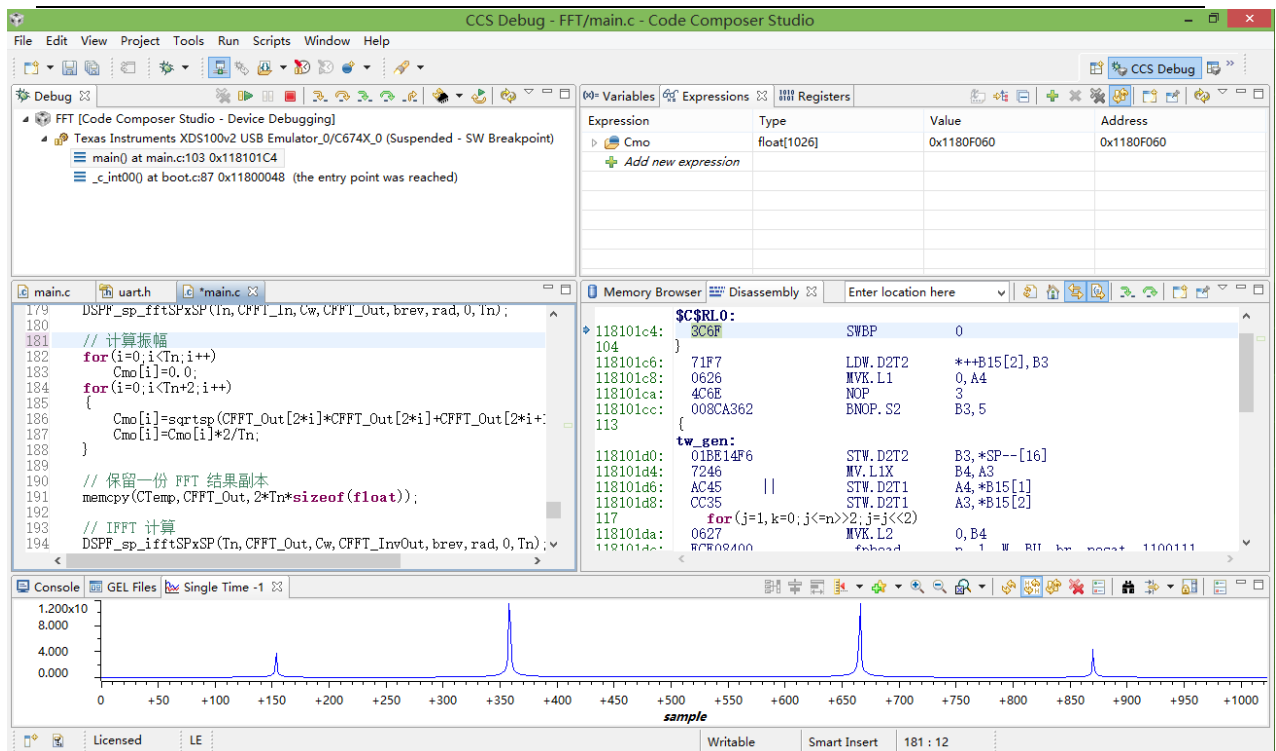


图 78

MATLAB 代码 FFT.m 位于光盘 demo\Application\FFT，程序如下图：

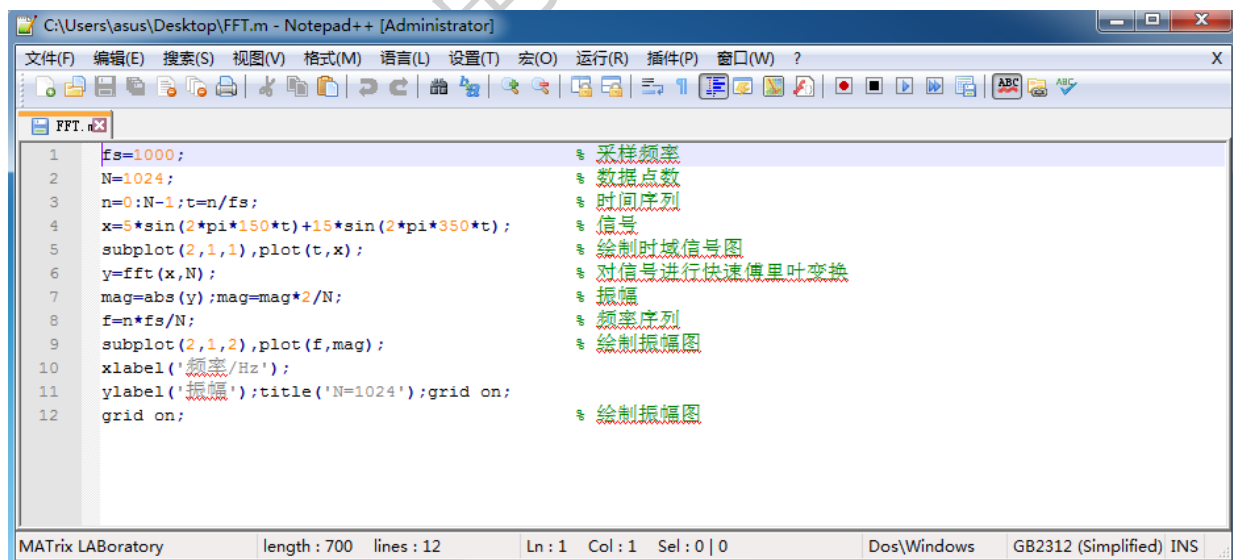


图 79



MATLAB 图表如下图:

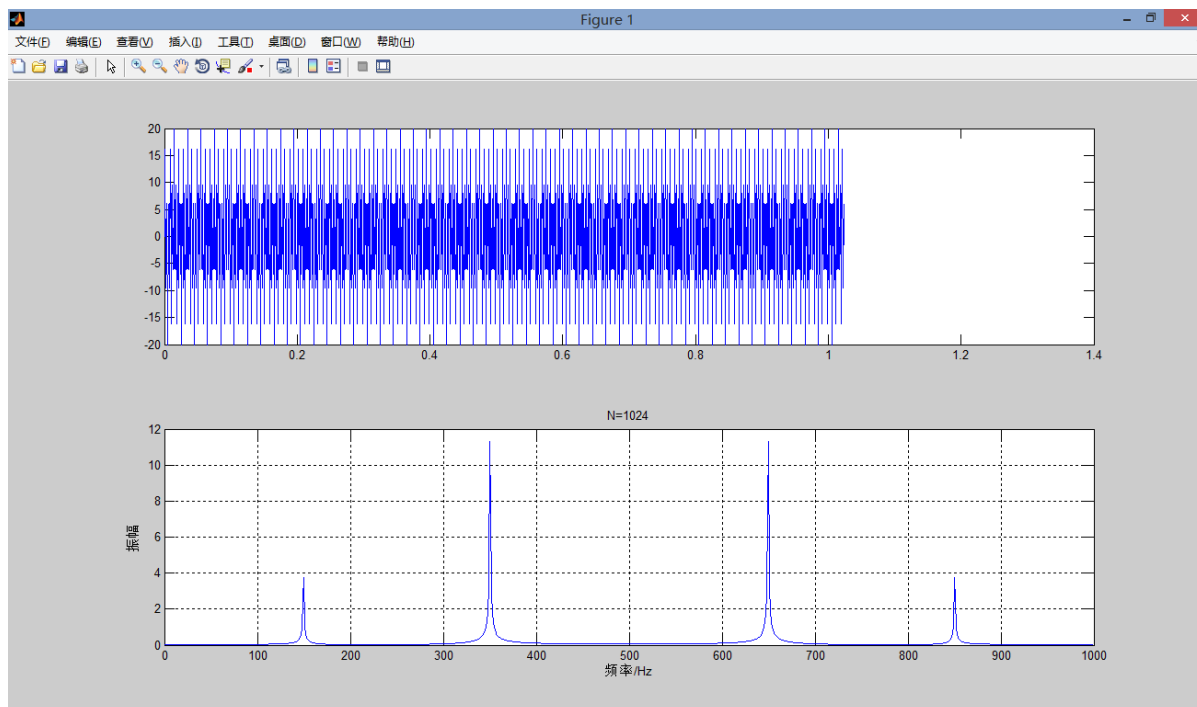


图 80

经对比可以看出 CCS ， 经过 DSP C6748 计算结果与 MATLAB 计算结果一致。



6 附录

6.1 Boot Mode Selection

Table 11. Boot Mode Selection		
BOOT[7:0] ⁽¹⁾	Boot Mode	AIS
0000 0010	NOR	Yes ⁽²⁾
0xx0 1110 ⁽³⁾	NAND 8	Yes
0xx1 0000 ⁽³⁾⁽⁴⁾	NAND 16	Yes
00x1 1100 ⁽⁵⁾⁽⁶⁾	MMC/SD0	Yes
0000 0000	I2C0 EEPROM	Yes
0000 0110	I2C1 EEPROM	Yes
0000 0001	I2C0 Slave	Yes
0000 0111	I2C1 Slave	Yes
0000 1000	SPI0 EEPROM	Yes
0000 1001	SPI1 EEPROM	Yes
0000 1010	SPI0 Flash	Yes
0000 1100	SPI1 Flash	Yes
0001 0010	SPI0 Slave	Yes
0001 0011	SPI1 Slave	Yes
xxx1 0110 ⁽⁷⁾	UART0	Yes
xxx1 0111 ⁽⁷⁾	UART1	Yes
xxx1 0100 ⁽⁷⁾	UART2	Yes
0000 0100	HPI	No
0001 1110	Emulation Debug	No

图 81