

TMS320C6748 开发例程使用手册

Revision History

Draft Date	Revision No.	Description
2018/10/29	V3.3	<ul style="list-style-type: none">1. 完善 StarterWare 例程 ECAP_APWM，底板 TL138/6748-EasyEVM 的 A3 和 A4 版的测试引脚不一样。2. UART_INT 例程不适用 TL138/6748-EasyEVM 的 A4 版。
2018/10/11	V3.2	<ul style="list-style-type: none">3. 完善 SYS/BIOS 例程 Board 的操作步骤。
2018/08/03	V3.1	<ul style="list-style-type: none">1.添加算法例程——车牌识别。
2018/07/24	V3.0	<ul style="list-style-type: none">1.修改 StarterWare 的 DEMO 综合例程——重新换了新的 LOGO 图片。4. 修改 SYS/BIOS 工程新建。5. 修改 SYS/BIOS 例程 Queue。
2018/01/18	V2.8	<ul style="list-style-type: none">1.修改 4.62 NandFlash——NAND FLASH 读写测试，增加了高配版例程说明。
2017/03/20	V2.7	<ul style="list-style-type: none">1.添加其他例程 GPIO_LED_C_Mix。2.添加其他例程 GPIO_LED_C++_Mix。3.添加 SYS/BIOS 例程 AD 模块采集测试（使用 uPP）。4.添加 SYS/BIOS 例程伺服电机测试。
2017/01/20	V2.6	<ul style="list-style-type: none">1. 更新 NSP 组件为 1.10.3.15 版本，程序可兼容旧版本 NSP，更新以下例程： MJPEG_Streamer、NDK_Runtime、RawSocket、TCP、TCP_Benchmark、TCP_Client、Telnet、UDP、WebServer、WebServer_Audio_Video、WebServer_RMII。2. 添加 SYS/BIOS 例程 Gate 门保护。3. 添加 SYS/BIOS 例程 Queue 队列。4. 添加 SYS/BIOS 例程 Event 事件模块。5. 添加 SYS/BIOS 例程 Mailbox 邮箱通信。6. 添加 SYS/BIOS 例程 CPU 低功耗模式。7. 添加 SYS/BIOS 例程 VFSCALE 动态调频调压测试。8. 添加 SYS/BIOS 例程综合 demo。9. 添加 SYS/BIOS 例程 uPP_B_TO_A 例程 uPP 回环测试。10. 添加 SYS/BIOS 例程 NDK_UIA 系统分析基于网络传输。11. 添加 StarterWare 例程 SPI_EDMA_FLASH。12. 添加 StarterWare 例程 EMIF_AD8568_100K。13. 添加 StarterWare 例程 EMIF_AD7606_200K_EDMA。14. 添加 StarterWare 例程 EMIF_AD7606_10K。15. 添加 StarterWare 例程 TL5147_LCD_2CH_SYNC。16. 添加 StarterWare 例程 EDMA3_LED。17. 添加 StarterWare 例程 EDMA3_LINK。18. 添加 PRU 例程 PRU_AD8568_SAVE。

		19. 添加 PRU 例程 PRU_TIMER2_POLL。 20. 添加 PRU 例程 PRUtoDSP_Interrupt。 21. 优化 VPIF_OV2640 例程, 速率提高至 30 帧, 并通过 LCD 显示帧率。
2016/12/09	V2.5	1. 基于算法例程的 ImageProcessQR 二维码扫描测试例程。 2. 基于算法例程的 ImageProcessChar 文字识别测试例程。 3. 基于算法例程的 ImageProcessAlph 字母识别测试例程。
2016/08/26	V2.4	1. 添加基于 StarterWare 的 SD 卡读写测试例程。 2. 添加基于 StarterWare 的 SPIO 测试例程。 3. 添加基于 StarterWare 的 UART2 FIFO 中断模式测试例程。 4. 添加基于 StarterWare 的 TL5147 摄像头实现 3 路图片采集 5. 添加基于 PRU 的 PRU 控制 AD5724 输出连续波形例程。
2016/07/25	V2.3	1. 添加 C++ 的 GPIO 输出例程。 2. 添加基于 SYS/BIOS 的 UART2 中断接收例程。
2016/04/25	V2.2	1. 添加基于 StarterWare 的 5.6、10.4、12、15 寸触摸屏驱动例程。 2. 添加基于 StarterWare 的复合视频双通道输入 LCD 显示例程。 3. 添加标准汇编的 GPIO 输出例程。 4. 添加线性汇编的 GPIO 输出例程。 5. 添加 GPIO_KEY 直接操作寄存器中断实现流水灯例程。 6. 更新 SD 卡烧写程序到 NAND FLASH 内容。 7. 更新 Web_IAP 网络烧写工具版本。
2015/01/05	V2.1	1. 添加基于 SYS/BIOS 的二进制信号量例程。 2. 添加基于 SYS/BIOS 的计数型信号量例程。 3. 添加基于 StarterWare 的 uPP 板间通信例程。 4. 添加基于 StarterWare 的双 CAN 通信例程。 5. 添加基于 PRU 的 LED 输出例程。 6. 添加基于 PRU 的 KEY 输入例程。 7. 添加基于 PRU 的 DAC 输出例程。 8. 添加基于 PRU 的 7606 ADC 采集例程。 9. 添加基于 PRU 的 8568 ADC 采集例程。 10. 添加基于 PRU 的 uPP 通信例程。 11. 添加网络烧写程序的操作方法。 12. 优化 FFT_Benchmark 例程。
2015/09/24	V2.0	1. 更新 FaceDetect 人脸识别测试案例。 2. 删除基于 SYS/BIOS 的 GPIO_LED_CLOCK 例程。 3. 删除基于 SYS/BIOS 的 AUDIO_LINE_IN 音频输入例程。 4. 删除基于 SYS/BIOS 的 AUDIO_LINE_OUT 音频输出例程。

5. 更新基于 Starerware 的 Line Out 音频输入 (EMDA 方式) 例程。
6. 添加基于 Starerware 的 Line Out 音频输出 (中断方式) 例程。
7. 添加基于 Starerware 的 Line Out 音频输出 (查询方式) 例程。
8. 添加基于 Starerware 的 Line Out 音频输出 (导入文件方式) 例程。
9. 更新基于 Starerware 的 Line In 音频输入 (EMDA 方式) 例程。
10. 添加基于 Starerware 的 Line In 音频输入 (中断方式) 例程。
11. 添加基于 Starerware 的 Line In 音频输入 (查询方式) 例程。
12. 添加基于 Starerware 的 LineIn 音频输入 (保存文件方式) 例程。
13. 添加基于 Starerware 的 LineIn 音频输入 (波形输入方式) 例程。
14. 更新基于 Starerware 的 Mic In 音频输入 (EMDA 方式) 例程。
15. 添加基于 Starerware 的 Mic In 音频输入 (中断方式) 例程。
16. 添加基于 Starerware 的 Mic In 音频输入 (查询方式) 例程。
17. 添加基于 Starerware 的 Mic In 音频输入 (保存文件方式) 例程。
18. 添加基于 Starerware 的 Mic In 音频输入 (波形输入方式) 例程。
19. 添加基于 Starerware 的 EMIFA 总线 8 通道并口 AD 数据采集 (EMIF_AD8568)。
20. 添加基于 Starerware 的 EMIFA 总线 8 通道并口 AD 数据采集 (EMIF_AD8568v2)。
21. 添加 TL6748-EasyEVM 与 TL6748F-EVM 开发板使用 EMIF 接口模块说明。
22. 添加基于 Starerware 的 ClockOut 时钟输出测试例程。
23. 添加基于 Starerware 的 CPU 时钟测试例程。
24. 添加基于 SYS/BIOS 的时间戳 (通用) 例程。
25. 添加基于 SYS/BIOS 的时间戳 (专用) 例程。
26. 添加基于 SYS/BIOS 的 Line In 音频输入与保存例程。
27. 添加基于 SYS/BIOS 的 Mic In 音频输入与保存例程。
28. 添加基于 SYS/BIOS 的 MP3 音频播放 (SD 卡) 例程。
29. 添加基于 SYS/BIOS 的 WAV 音频播放 (SD 卡) 例程。
30. 添加基于 SYS/BIOS 的网络 Web 服务器 (支持串口输入 IP) 例程。

2015/05/12

V1.9

1. 添加基于 Starerware 的 NRF24L01 无线模块测试。
2. 添加基于 Starerware 的 EMIFA 总线 6 通道并口 AD 数据采集 (EMIF_AD7656)。
3. 添加基于 Starerware 的 EMIFA 总线 6 通道并口 AD 数据采集 (EMIF_AD7656_v2)。
4. 添加基于 Starerware 的 HC-SR04 超声波测距测试。
5. 添加基于 Starerware 的 DHT11 温湿度传感器测试。
6. 添加基于 Starerware 的串口 WIFI 模块测试。
7. 添加基于 Starerware 的 RFID 射频识别测试。
8. 添加基于 Starerware 的红外遥控测试。
9. 添加基于 Starerware 的 SATA 枚举测试。
10. 添加基于 Starerware 的串口蓝牙模块测试。
11. 添加基于 Starerware 的三轴加速陀螺仪测试。
12. 添加基于 Starerware 的串口转 Zigbee 无线测试。
13. 添加 H264 编码算法。

2015/04/30

V1.8

1. 添加基于 StarterWare 的网络 Web 服务器 (使用 RMII 接口) 例程。
2. 添加基于 SYS/BIOS 的网络 Web 服务器 (使用 RMII 接口) 例程。
3. 添加基于 SYS/BIOS 的 EDMA3 一维数据传输例程。
4. 添加基于 SYS/BIOS 的 LineOut 音频输出例程。
5. 添加基于 SYS/BIOS 的 McBSP_LoopBackMcBSP 内部回环测试。
6. 添加基于 SYS/BIOS 的 McBSP 外部回环测试。
7. 添加基于 Starerware 的 TL5147 复合视频输入 VGA 显示测试。
8. 添加基于 Starerware 的 TL5147 复合视频输入 LCD 显示测试。

2015/03/18

V1.7

1. 调整文档目录排版, 修改例程分类为 StarterWare、SYS/BIOS 和算法三部分。
2. 添加基于 C6748_NandWritev2.out 程序烧写方法, 烧写速度约为基于 C6748_NandWrite .out 的 10 倍。
3. 添加基于 StarterWare 经过 EDMA 优化的 EMIFA 总线 FPGA 读写测试例程。
4. 添加基于 StarterWare 的图形库控件例程。
5. 添加 AAC 解码 AACLCDecode 算法例程。
6. 添加 AAC 解码 AACHEv2Decode 算法例程。
7. 添加 AAC 编码 AACLCEncode 算法例程。
8. 添加直方图均衡化算法例程。
9. 添加图像反色算法例程。
10. 添加边缘检测算法例程。
11. 添加灰度图像二值化算法例程。
12. 添加灰度图像线性变换算法例程。
13. 添加图像缩放算法例程。
14. 添加图像旋转算法例程。

2015/02/10

V1.6

-
- 15. 添加 G711A 率语音编码算法例程。
 - 16. 添加 G711A 率语音解码算法例程。
 - 1. 添加基于 SD 卡烧写 NANDFLSH 程序方法。
 - 2. 添加 CCS 开启多线程编译方法。
 - 3. 添加自动生成 NANDFLASH 烧写.ais 文件方法。
 - 4. 添加基于 SYS/BIOS 的 IP Camera 例程。
 - 5. 添加基于 StarterWare 的多串口模块（查询方式）例程。
 - 6. 添加基于 StarterWare 的多串口模块（中断方式）例程。
 - 7. 添加基于 SYS/BIOS 的定时器（通用）例程。
 - 8. 添加基于 SYS/BIOS 的定时器（专用）例程。
 - 9. 添加基于 SYS/BIOS 的定时器（动态创建）例程。
 - 10. 添加基于 SYS/BIOS 的定时器（动态创建、更改定时周期）例程。
 - 11. 更改基于 SYS/BIOS 的 GPIO_KEY_HWI 例程名为 HWI_Runtime。
 - 12. 添加基于 SYS/BIOS 的硬件中断（HWI 设备专用组件）例程。
 - 13. 添加基于 SYS/BIOS 的硬件中断（HWI 挂钩函数）例程。
 - 14. 添加基于 SYS/BIOS 的硬件中断（HWI 发布软件中断）例程。
 - 15. 添加基于 SYS/BIOS 的硬件中断（HWI 触发任务）例程。
 - 16. 更改基于 SYS/BIOS 的 GPIO_KEY_SWI 例程为 SWI_Runtime。
 - 17. 添加基于 SYS/BIOS 的软件中断（静态配置）例程。
 - 18. 添加基于 SYS/BIOS 的软件中断（有条件触发 ANDN）例程。
 - 19. 添加基于 SYS/BIOS 的软件中断（有条件触发 DEC）例程。
 - 20. 添加基于 SYS/BIOS 的软件中断（无条件触发 OR）例程。
 - 21. 添加基于 SYS/BIOS 的 MP3 解码（使能缓存及通过 SD 存取）例程。
-
- 1. 添加基于 StarterWare 的关联 32-bit 模式定时器/计数器例程。
 - 2. 添加基于 StarterWare 的独立 32-bit 模式定时器/计数器例程。
 - 3. 添加基于 StarterWare 的独立 32-bit 模式/额外 4-bit 分频定时器/计数器例程。
 - 4. 添加基于 SYS/BIOS 的 UART2 串口查询收发例程。
 - 5. 添加基于 SYS/BIOS 的以太网数据链路层通信例程。
 - 6. 添加基于 SYS/BIOS 的 MP3 解码例程。

2015/01/13

V1.5

		7. 添加基于 SYS/BIOS 的 CodecEngine 数据复制算法例程。
2014/12/10	V1.4	<ol style="list-style-type: none">1. 添加支持 4.3 寸屏例程。2. 添加基于 StarterWare 的 4 通道 DAC 例程（模拟 SPI 总线）。3. 添加基于 StarterWare 的 4 通道 DAC 例程（SPI 总线）。
2014/11/03	V1.3	<ol style="list-style-type: none">1. 添加基于 TL-EasyBox 实验板 Demo 例程。2. 添加基于 StarterWare 的综合例程。3. 添加基于 StarterWare 的人脸识别跟踪例程。4. 添加基于 StarterWare 的按键触发 EDMA 事件例程。5. 添加基于 StarterWare 的按键及定时器中断例程。6. 添加基于 StarterWare 的 EDMA 串口收发例程。7. 添加基于 SYS/BIOS 的触摸屏例程。
2014/10/13	V1.2	<ol style="list-style-type: none">1. 增加基于 StarterWare 和 FPGA 通信的 uPP 测试例程。2. 增加基于 StarterWare 和 FPGA 通信的 EMIFA 测试例程。3. 增加基于 SYS/BIOS 的 SD 卡 RAW 模式读写测试例程。4. 增加基于 SYS/BIOS 的 SD 卡文件系统读写测试例程。5. 增加基于 SYS/BIOS 的 WebServer 测试例程。
2014/10/01	V1.1	<ol style="list-style-type: none">1. 优化 GEL 文件。2. 增加基于 StarterWare 的内存读写测试例程。3. 增加基于 StarterWare 的 NANDFLASH 读写测试例程。4. 增加基于 StarterWare 的 VGA 显示测试例程。5. 增加基于 StarterWare 的 RGB24 转灰度测试例程。6. 增加基于 StarterWare 的灰度图像直方图测试例程。7. 增加基于 StarterWare 的数学函数库测试例程。8. 增加基于 StarterWare 的数字识别测试例程。9. 增加基于 SYS/BIOS 的工程创建说明。10. 增加基于 SYS/BIOS 的例程。11. 调整目录结构。
2014/05/05	V1.0	<ol style="list-style-type: none">1. 初始版本。

目 录

1 开发板准备	16
1.1 查看仿真器驱动是否正常安装	17
2 程序加载和烧写	19
2.1 基于仿真器的程序加载和烧写	19
2.1.1 设置工程配置文件信息	19
2.1.2 测试仿真器是否正常连接	20
2.1.3 加载 GEL 文件	21
2.1.4 CCS 连接开发板 CPU	22
2.1.5 查看 CPU 版本号	23
2.1.6 加载文件烧写程序	24
2.1.7 基于仿真器烧写程序到 NAND FLASH	24
2.1.8 NAND FLASH 烧写格式镜像转换方法	29
2.2 基于串口的程序加载和烧写	31
2.2.1 串口加载和烧写格式镜像转换方法	31
2.2.2 程序加载和运行	33
2.2.3 基于串口烧写程序到 NAND FLASH	39
2.2.4 基于串口烧写程序到 SPI FLASH	42
2.3 基于 SD 卡运行程序	44
2.3.1 SD 启动格式镜像转换方法	44
2.3.2 将程序烧写到 SD 卡	46
2.3.3 SD 卡模式启动开发板	49
2.4 基于 SD 卡烧写程序到 NAND FLASH	49
2.4.1 格式化 SD 卡	49
2.4.2 准备烧写镜像	53
2.5 基于网络的程序烧写和启动	55
3 CCS 工程新建、编译和导入	59
3.1 裸机工程创建	61
3.1.1 新建工程	61
3.1.2 编写程序	64

3.1.3 编译和运行程序	64
3.2 SYS/BIOS 工程创建	66
3.2.1 创建 SYS/BIOS 平台配置文件	68
3.2.2 新建 SYS/BIOS 工程	74
3.2.3 编译与运行程序	79
3.3 CCS 工程导入和编译步骤	79
3.4 开启多线程编译	86
3.5 添加自动生成 NAND FLASH 烧写.ais 文件	88
4 基于 StarterWare 的 Demo 例程演示	92
4.1 DEMO——综合例程	94
4.2 GPIO_LED——GPIO 输出 (LED 灯)	98
4.3 GPIO_KEY——GPIO 输入 (按键中断)	98
4.4 GPIO_KEY_EDMA——按键触发 EDMA 事件	99
4.5 GPIO_KEY_TIMER_EventCombine——按键及定时器中断	101
4.6 TIMER——定时器	101
4.7 TIMER_Dual_32-bit_Chained——关联 32-bit 模式定时器/计数器	102
4.8 TIMER_Dual_32-bit_UnChained——独立 32-bit 模式定时器/计数器	104
4.9 TIMER_Dual_32-bit_UnChained_4-bit_Prescaler——独立 32-bit 模式/额外 4-bit 分频定时器/计数器	106
4.10 UART0_INT——UART0 串口中断收发	108
4.11 UART1_POLL——UART1 串口查询收发	110
4.12 UART2_INT——UART2 串口中断收发	111
4.13 UART2_EDMA——EDMA 串口收发	112
4.14 RS485——RS485 串口查询收发	113
4.15 TL_MULTUART_INT——多串口模块 (查询方式)	114
4.16 TL_MULTUART_POLL——多串口模块 (中断方式)	116
4.17 IIC EEPROM——IIC EEPROM 读写	117
4.18 SPI_FLASH——SPI FLASH 读写	118
4.19 SPI_EDMA_FLASH——SPI FLASH 读写测试	119
4.20 SPI_DAC_AD5724——4 通道 DAC 模块 (模拟 SPI 总线)	120

4.21	SPI_DAC_AD5724v2——4 通道 DAC 模块 (SPI 总线)	124
4.22	WatchDog——看门狗.....	124
4.23	NMI——不可屏蔽中断.....	125
4.24	PWM——高精度脉冲宽度调制器 PWM 输出.....	125
4.25	ECAP_APWM——增强型捕获模块 ECAP 辅助输出.....	127
4.26	PWM_ECAP——增强型捕获模块 ECAP 捕获.....	130
4.27	RTC——RTC 时钟	132
4.28	LCD——LCD 显示	133
4.29	VGA——VGA 显示	135
4.30	LCD_TOUCH_4INCH3——4.3 寸触摸屏	136
4.31	LCD_TOUCH——7 寸触摸屏.....	138
4.32	LCD_TOUCH_5INCH6/10INCH4/12INCH/15INCH——多种尺寸触摸屏.....	138
4.33	GRLIB_DEMO——StarterWare 图形库控件	140
4.34	MMCSD——SD 卡读写	140
4.35	SATA——SATA 枚举测试.....	141
4.36	USB_DEV_BULK——USB OTG 从方式 (USB BULK 管道通信)	141
4.37	USB_DEV_MSC——USB OTG 从方式 (虚拟存储设备)	150
4.38	USB_DEV_SERIAL——USB OTG 从方式 (USB 虚拟串口)	160
4.39	USB_HOST_KEYBOARD——USB OTG 主方式 (USB 键盘)	163
4.40	USB_HOST_MOUSE——USB OTG 主方式 (USB 鼠标)	163
4.41	USB_HOST_MSC——USB OTG 主方式 (U 盘内容查看)	165
4.42	ENET_HTTPD——网络 Web 服务器	166
4.43	ENET_HTTPD_RMII——网络 Web 服务器 (使用 RMII 接口)	169
4.44	ENET_ECHO——网络 Socket 通信	171
4.45	AUDIO_LINE_OUT——Line Out 音频输出 (EMDA 方式)	177
4.46	AUDIO_LINE_OUT_INTR——Line Out 音频输出 (中断方式)	182
4.47	AUDIO_LINE_OUT_POLL——Line Out 音频输出 (查询方式)	183
4.48	AUDIO_LINE_OUT_WAV——Line Out 音频输出 (导入文件方式)	184
4.49	AUDIO_MIC_IN——Mic In 音频输入 (EMDA 方式)	187
4.50	AUDIO_MIC_IN_INTR——Mic In 音频输入 (中断方式)	188

4.51	AUDIO_MIC_IN_POLL——Mic In 音频输入（查询方式）	189
4.52	AUDIO_MIC_IN_SAVEMEM——Mic In 音频输入（保存文件方式）	189
4.53	AUDIO_MIC_IN_WAVE——Mic In 音频输入（波形输入方式）	195
4.54	AUDIO_LINE_IN——Line In 音频输入（EMDA 方式）	198
4.55	AUDIO_LINE_INTR——Line In 音频输入（中断方式）	199
4.56	AUDIO_LINE_POLL——Line In 音频输入（查询方式）	200
4.57	AUDIO_LINE_IN_SAVEMEM——Line In 音频输入（保存文件方式）	201
4.58	AUDIO_LINE_IN_WAVE——Line In 音频输入（波形输入方式）	206
4.59	McBSP——McBSP 总线数据收发	209
4.60	VPIF_OV2640——VPIF 总线 CMOS 摄像头数据采集	211
4.61	Memory_Benchmark——内存读写速度测试	213
4.62	NandFlash——NAND FLASH 读写测试	213
4.63	EMIF_AD7606——EMIFA 总线 8 通道并口 AD 数据采集	215
4.64	EMIF_AD7606v2——EMIFA 总线 8 通道并口 AD 数据采集	219
4.65	EMIF_AD7606_10K——使用 CPU 采集方式按 10k 速度采样	222
4.66	EMIF_AD7606_200K_EDMA——使用 EDMA 方式按 200k 采样率进行采样	223
4.67	EMIF_AD7656——EMIFA 总线 6 通道并口 AD 数据采集	224
4.68	EMIF_AD7656v2——EMIFA 总线 6 通道并口 AD 数据采集	227
4.69	EMIF_AD8568——EMIFA 总线 8 通道并口 AD 数据采集	229
4.70	EMIF_AD8568v2——EMIFA 总线 8 通道并口 AD 数据采集	231
4.71	EMIF_AD8568_100K——以 100 KHz 采样 AD8568 8 通道 AD 数据	235
4.72	EMIF_FPGA——EMIFA 总线 FPGA 读写测试	238
4.73	EMIF_FPGA_DMA——EMIFA 总线 FPGA 读写测试（经过 EDMA 优化）	240
4.74	EDMA3——EDMA3 一维数据传输	242
4.75	EDMA3_TRANSPOSE——EDMA3 二维数据传输	243
4.76	uPP_B_TO_A——uPP 总线 FPGA 读写测试	244
4.77	uPP_2CH——uPP 板间双通信测试	248
4.78	TL2515_CAN——双 CAN 通信测试	250
4.79	TL5147_VGA——复合视频输入 VGA 显示测试	252
4.80	TL5147_LCD——复合视频输入 LCD 显示测试	254

4.81	TL5147_LCD_2CH——复合视频双通道输入 LCD 显示测试	257
4.82	H264Encoder——编码例程测试	258
4.83	NRF24L01——2.4G 无线模块测试	262
4.84	HC-SR04——超声波测距测试	265
4.85	DHT11——温湿度传感器测试	267
4.86	WIFI_UART——串口 WIFI 模块测试	269
4.87	RFID——RFID 射频识别测试	273
4.88	ECAP_REMOTE——红外遥控测试	275
4.89	BTUART——串口蓝牙模块测试	277
4.90	MPU6050——三轴加速陀螺仪测试	279
4.91	ZIGBEE——串口转 Zigbee 无线测试	282
4.92	ClockOut——时钟频率测试	284
4.93	DSPClockSpeed——CPU 时钟测试	287
4.94	Fsfat——SD 卡读写测试	288
4.95	SPI0——SPI0 测试例程	289
4.96	UART2_INT_FIFO——UART2 FIFO 中断模式	291
4.97	TL5147_LCD_3CH_Multi——TL5147 摄像头实现 3 路图片采集	292
4.98	TL5147_LCD_2CH_SYNC——双路 5147 摄像头同时采集显示	295
4.99	EDMA3_LINK——EDMA 链接工作方式	296
4.100	EDMA3_LED——EDMA 方式让 LED 闪烁	297
4.101	EMIF_AD7606_PinBoard——EMIF 拓展板测试	297
5	基于 SYS/BIOS 的 Demo 例程演示	300
5.1	Board——综合例程	300
5.2	GPIO_LED——任务	306
5.3	GPIO_LED_MUTEX——抢占式多任务	306
5.4	GPIO_LED_STATIC——静态创建任务	307
5.5	Semaphore_Binary——二进制信号量	311
5.6	Semaphore_Counting——计数型信号量	312
5.7	Clock——时钟	313
5.8	Timestamp——SYS/BIOS 时间戳（通用）	313

5.9	Timestamp_C674x——SYS/BIOS 时间戳（专用）	314
5.10	Timer——定时器（通用）	314
5.11	Timer_C674x——定时器（专用）	316
5.12	Timer_C674x_Runtime——定时器（动态创建）	316
5.13	Timer_C674x_Runtime_Reload——定时器（动态创建、更改定时周期） ...	317
5.14	HWI_C674x——硬件中断（HWI 设备专用组件）	317
5.15	HWI_C674x_Hook——硬件中断（HWI 挂钩函数）	318
5.16	HWI_C674x_Nest——硬件中断（HWI 中断嵌套）	319
5.17	HWI_Runtime——硬件中断（HWI）	319
5.18	HWI_Runtime_Post_SWI——硬件中断（HWI 发布软件中断）	319
5.19	HWI_Runtime_Post_Task——硬件中断（HWI 触发任务）	320
5.20	SWI——软件中断（静态配置）	320
5.21	SWI_Runtime——软件中断（SWI）	322
5.22	SWI_Runtime_Post_Conditionally_andn——软件中断（有条件触发 ANDN）	
	322	
5.23	SWI_Runtime_Post_Conditionally_dec——软件中断（有条件触发 DEC） ...	323
5.24	SWI_Runtime_Post_Unconditionally_or——软件中断（无条件触发 OR） ...	323
5.25	MEMORY——内存分配	324
5.26	MMCSD——SD 卡 RAW 模式	325
5.27	MMCSD_FatFs——SD 卡 FAT 文件系统	328
5.28	UART1——UART1 串口查询收发	330
5.29	UART2——UART2 串口查询收发	332
5.30	UART2_INT——UART2 串口中断接收	333
5.31	AudioLineInSave——Line In 音频输入与保存	334
5.32	AudioMicInSave——MicIn 音频输入与保存	334
5.33	AudioPlayMP3——MP3 音频播放（SD 卡）	335
5.34	AudioPlayWAV——WAV 音频播放（SD 卡）	335
5.35	LCD_TOUCH——触摸屏	336
5.36	TCP_Client——TCP 客户端	337
5.37	TCP——TCP 服务器	344

5.38	UDP——UDP 通信	348
5.39	TCP_Benchmark——TCP 发送/接收速度测试	351
5.40	Telnet——Telnet 协议	355
5.41	Telnet——TFTP 协议	362
5.42	WebServer——网络 Web 服务器	368
5.43	NDK_UIA——基于网络传输的系统分析	376
5.44	NDK_Runtime——网络 Web 服务器（支持串口输入 IP）	384
5.45	WebServer_RMII——网络 Web 服务器（使用 RMII 接口）	387
5.46	WebServer_Audio_Video——Web 服务器音视频实验	395
5.47	MJPEG_Streamer——IP Camera 网络摄像头	399
5.48	RawSocket——以太网数据链路层通信	402
5.49	EDMA3——EDMA3 一维数据传输	413
5.50	McBSP_LoopBack——McBSP 内部回环测试	414
5.51	McBSP——McBSP 外部回环测试	415
5.52	Gate——SYS/BIOS 门保护	416
5.53	Queue——SYS/BIOS 队列	416
5.54	Event——SYS/BIOS 事件模块	417
5.55	Mailbox——SYS/BIOS 邮箱通信	418
5.56	SLEEP——CPU 低功耗模式	418
5.57	VFSCALE——动态调频调压测试	419
5.58	uPP_B_TO_A——uPP 回环测试	420
5.59	ADS1278_uPP——AD 模块采集测试（使用 uPP）	423
5.60	伺服电机测试——通过 DSP 与 FPGA 控制伺服电机运行	424
6	算法 Demo 例程演示	426
6.1	FIR——有限长单位冲激响应滤波器	426
6.2	IIR——无限脉冲响应数字滤波器	432
6.3	Matrix——矩阵运算	437
6.4	FFT——快速傅里叶变换/逆变换	439
6.5	FFT_Benchmark——快速傅里叶变换/逆变换（打开/关闭缓存速度对比）	442
6.6	FFT_DIT2——基 2 时间抽取快速傅里叶变换/逆变换（原址计算）	465

6.7	DCT——图像离散余弦变换	468
6.8	RGB2Gray——RGB24 图像转灰度	471
6.9	HIST——灰度图像直方图.....	475
6.10	IntEqualize——直方图均衡化.....	484
6.11	ImageReverse——图像反色.....	493
6.12	Canny——边缘检测.....	499
6.13	Threshold——灰度图像二值化	505
6.14	LinerTrans——灰度图像线性变换.....	512
6.15	Zoom——图像缩放	519
6.16	Rotate——图像旋转	525
6.17	MATH——数学函数库.....	532
6.18	UniversalCopy——基于 Codec Engine 的数据复制算法.....	533
6.19	MP3Decode——MP3 解码	534
6.20	MP3Decode_SD——MP3 解码（使能缓存及通过 SD 存取）	535
6.21	AACLCDecode——AACLCDecode AAC 解码	538
6.22	AACHEv2Decode——AACHEv2Decode AAC 解码	539
6.23	AACLCEncode——AACLCEncode AAC 编码	541
6.24	G711ADecode——G711A 率语音编码	542
6.25	G711ADecode——G711A 率语音解码	543
6.26	ImageProcess——数字识别	545
6.27	FaceDetect——人脸识别跟踪	548
6.28	ImageProcessQR——二维码扫描	549
6.29	ImageProcessChar——文字识别.....	553
6.30	ImageProcessAlph——字母识别.....	556
6.31	Plate Recognition Demo——车牌识别	559
7	基于 TL-EasyBox 实验板 Demo 例程演示	569
7.1	BUZZER——蜂鸣器	570
7.2	MATRIX_KEY——4x4 键盘	570
7.3	DAC_TLC5615——DAC 输出	571
7.4	EMIF_AD7606——EMIFA 总线 8 通道并口 AD 数据采集	573

7.5	VPIF_OV2640——摄像头	574
7.6	DCMOTOR——直流电机	575
7.7	STEPPERMOTOR——步进电机	576
7.8	EASYBOX_DEMO——实验板综合测试	577
8	基于 PRU 的 Demo 例程演示	579
8.1	PRU_GPIO_LED——PRU 控制 GPIO 输出	583
8.2	PRU_GPIO_KEY——PRU 控制 GPIO 输入	583
8.3	PRU_TL5724_DAC——PRU 驱动 DAC 输出测试	583
8.4	PRU_TL5724_DACv2——PRU 控制 AD5724 输出连续波形	585
8.5	PRU_TL7606_ADC——PRU 触发 ADC 采集模拟量	588
8.6	PRU_TL8568_ADC——PRU 触发 ADC 采集模拟量	590
8.7	PRU_uPP_B_TO_A——PRU 控制 uPP 传输数据	591
8.8	PRU_AD8568_SAVE——PRU 控制 AD8568 采集数据存储 SD 卡	595
8.9	PRU_TIMER2_POLL——PRU 控制定时器	596
8.10	PRU to DSP_Interrupt——PRU 向 DSP 发送中断	596
9	其它 Demo 例程演示	597
9.1	GPIO_LED_Assembly——GPIO 输出（标准汇编）	597
9.2	GPIO_LED_LinearAssembly——GPIO 输出（线性汇编）	597
9.3	GPIO_LED_C++——GPIO 输出（C++）	597
9.4	GPIO_KEYv2——不使用 StarterWare 函数库中断例程	597
9.5	GPIO_LED_C_Mix——C 语言调用线性汇编、汇编语句及函数	598
9.6	GPIO_LED_C_C_Mix——C++语言调用 C 语言、线性汇编、汇编语句及函数	598
	附录 A	599

1 TMS320C6748 开发板准备

测试开发板硬件

将开发板的 UART2 连接到 PC 机。

备注:

- 如实验无特别说明, 默认都是使用 UART2 作为调试串口。
- 如实验无特别说明, 表示广州创龙 OMAPL138/C6748 系列开发板均支持对应实验。
- 广州创龙 OMAPL138/C6748 系列开发板共用此用户手册, 由于各个开发板之间的硬件资源存在差异, 因此有部分实验需要在特定的开发板上完成。

如使用的开发板的 UART2 是 DB9 接口, 请将 RS232 串口线通过 USB 转串口转接线连接到 PC 机的 USB 接口。RS232 串口线的另一端 DB9 母头 (带孔) 接开发板 UART2 接口。

如使用的开发板的 UART2 是 MicroUSB 接口, 请 MicroUSB 数据线插到 PC 机的 USB 接口, 另一头接开发板的 UART2 接口。

确保 USB 转串口驱动已正常安装后, 驱动可以在 tools 目录下找到, 也可以使用鲁大师扫描安装驱动。

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, OMAPL138/TMS320C6748 光盘的 tools 目录下有 ZOC 的安装包 zoc602.zip。安装方法请查阅使用手册 TMS320C6748 相关软件安装里 ZOC 软件安装步骤。

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。

接着点击 Save 完成串口终端的设置。将开发板断电, 然后将开发板的拨码开关编号 1~5 对应拨到 00101, 1 为 ON, 0 为 OFF, 这个档位是 UART2 启动模式, Boot 引脚详细配置说明请查看附录 1。然后将开发板上电, 在串口调试终端看到 BOOTME 信息即说明开发板硬件正常和 USB 转串驱动已正确安装, 如下图所示:

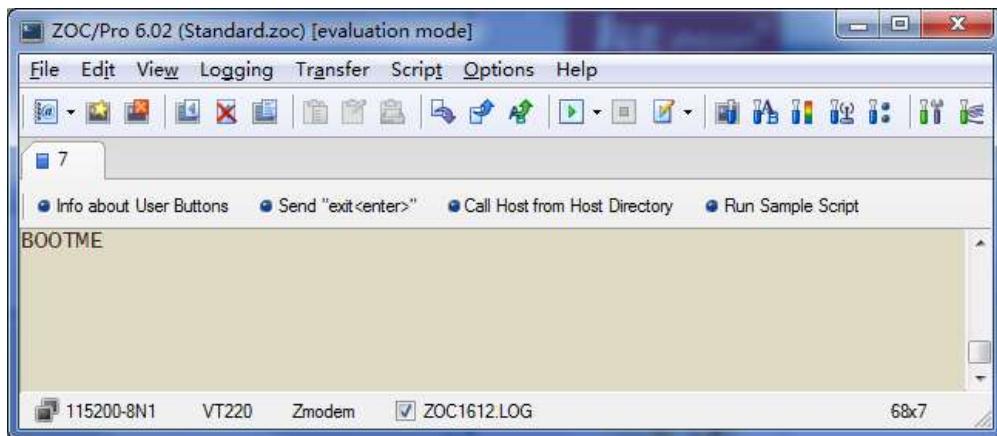


图 1

对于开发板常见启动方式，拨码开关 1~5 档位为如下：

NAND FLASH: 01110

MMC/SD0: 00111

UART1: 11101

UART2: 00101

SPI1 FLASH: 00110

DEBUG: 01111

备注：1 为 ON，0 为 OFF。

1.1 查看仿真器驱动是否正常安装

开发板断电，把拨码开关调到 DEBUG 模式：01111，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”或者“设备->端口（COM 和 LPT）”，查看是否有对应的仿真器的选项出现，如有说明仿真器驱动已经正常安装，否则请先正确安装 CCS。安装方法请查阅使用手册 TMS320C6748 相关软件安装里 CCS 软件安装步骤。

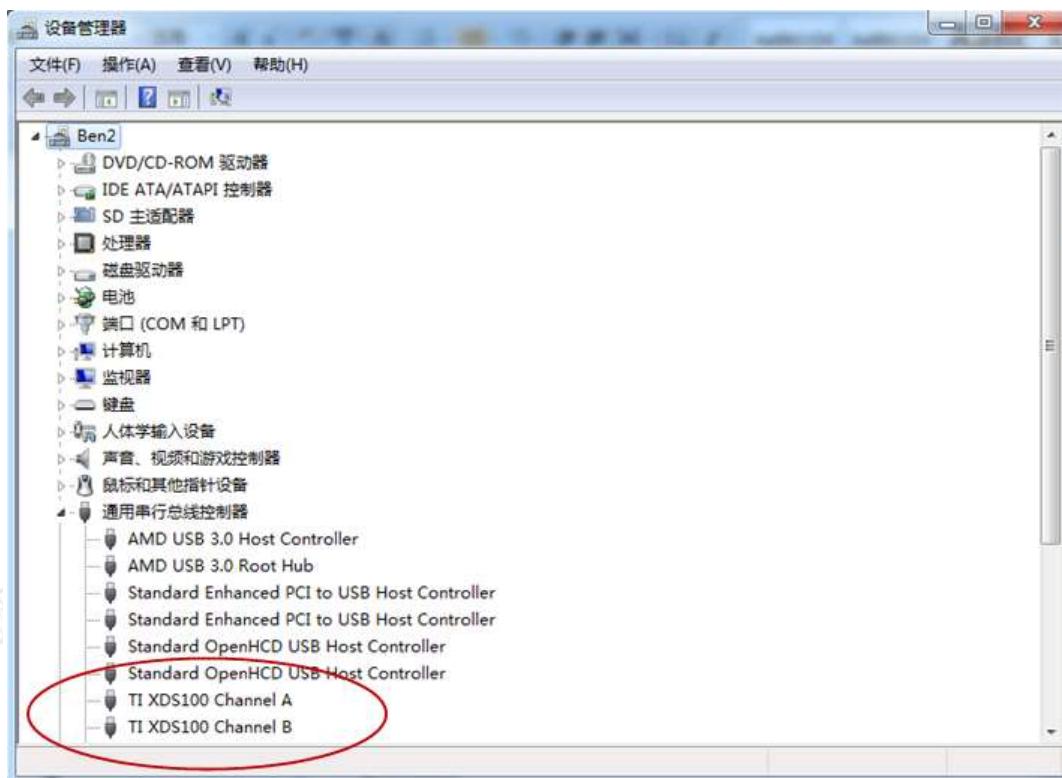


图 2



图 3

2 TMS320C6748 程序加载和烧写

2.1 基于仿真器的程序加载和烧写

2.1.1 设置工程配置文件信息

打开 CCS，点击菜单"View->Target Configurations"，右边弹出如下对话框：

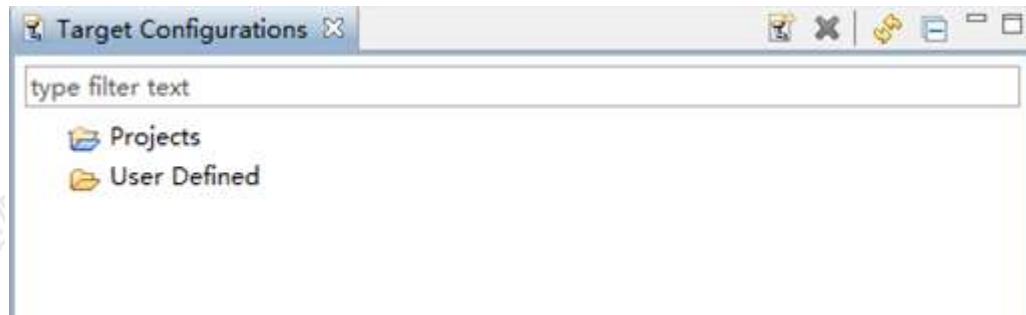


图 4

点击左边第一项，新建配置文件。



图 5

输入工程配置文件名字，点击 Finish。

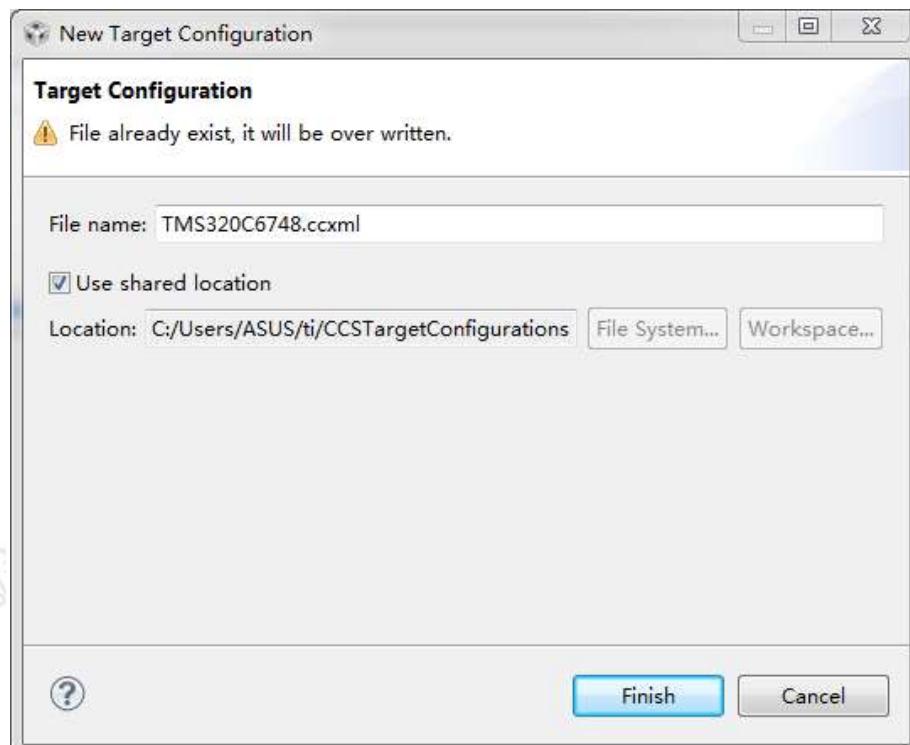


图 6

在弹出的对话框的"Connection"下拉框中选择对应的仿真器类型，在"Board or Device"下拉框中选择 TMS320C6748，点击右边的 Save 按钮。

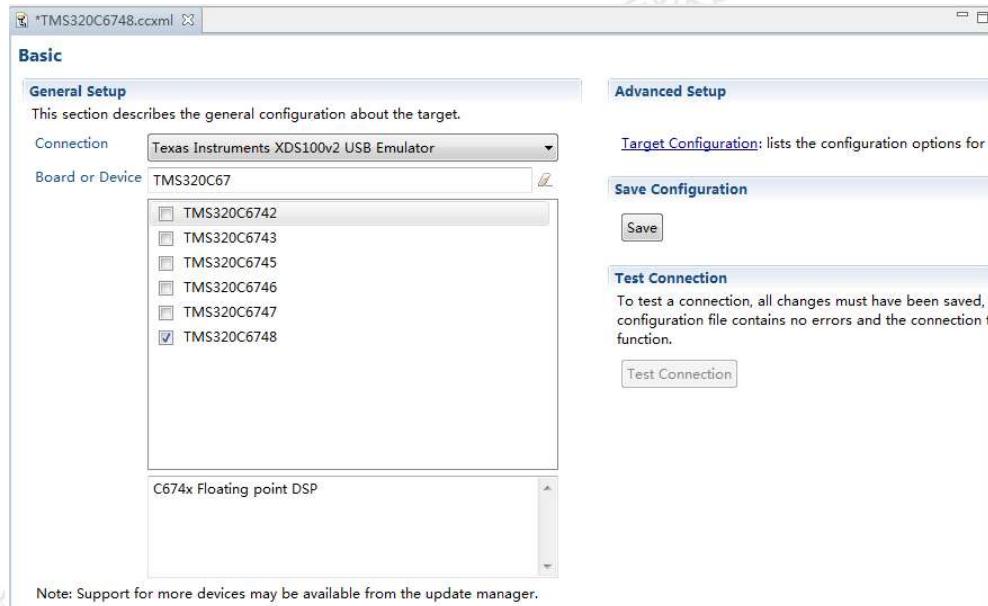


图 7

2.1.2 测试仿真器是否正常连接

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。

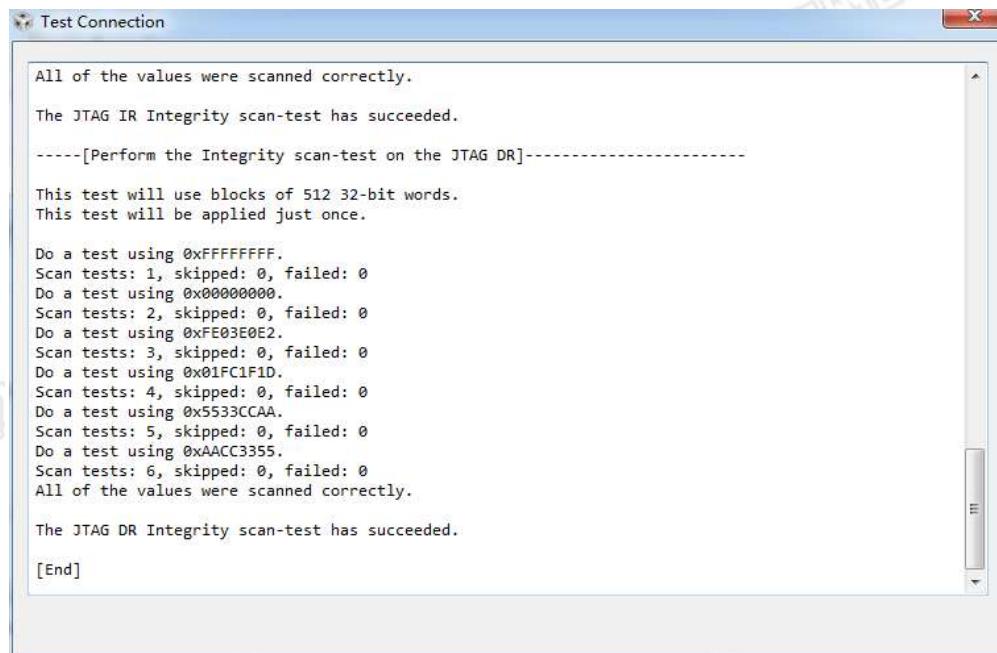


图 8

2.1.3 加载 GEL 文件

例程中使用"Tronlong_C6748.gel"文件初始化时钟配置如下：

CPU: 456MHz,

DDR2-624, DDR_CLK: 156MHz,

EMIFA: 114MHz。

加载步骤：

点击 CCS 菜单"Run->Debug"，弹出以下界面。可以看到 C674X_0。



图 9

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。

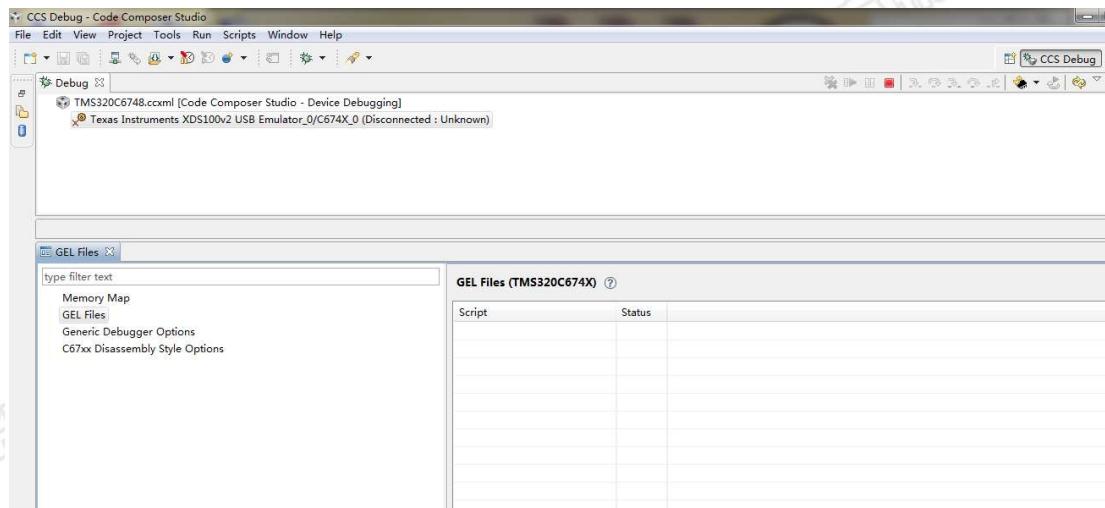


图 10

在对话框内点击右键，在弹出的界面中选择"Load GEL"。选择 C6748 光盘 images 下的 GEL 文件"Tronlong_C6748.gel"，再点击确定，接着右下角的"GEL Files (TMS320C674X)"对话框会出现 Success 提示语句，如下图：



图 11

2.1.4 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 已经和开发板 CPU 正常连接起来了。

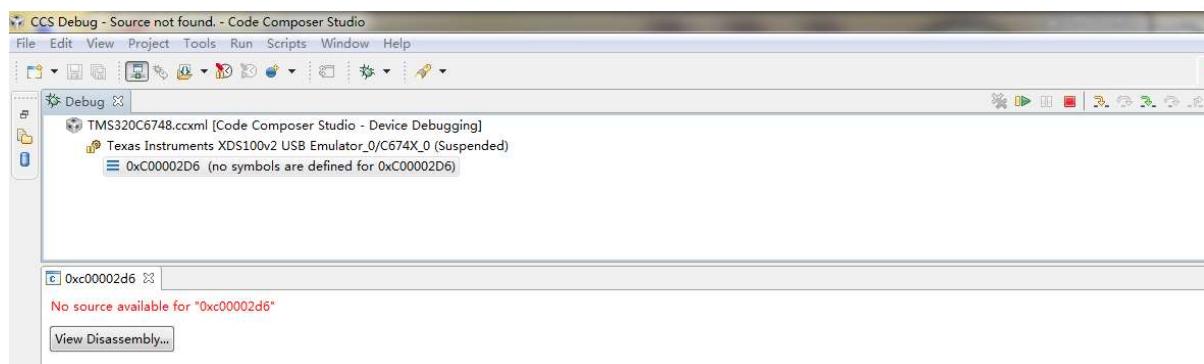


图 12

备注：此处提示"No source****"的信息，不是错误信息，可以将其忽视，仅提示当前运行的 DSP 程序找不到对应的源文件，如果加载的是有效的 DSP 程序而且当前 DSP 程序源文件位于磁盘中，CCS 会自动定位到相应文件并指向在相应的语句所在行。

"Tronlong_C6748.gel"文件加载后将自动初始化开发板，如下图所示：

```
TMS320C6748.ccxml
C674X_0: Output: Target Connected.
C674X_0: Output:
C674X_0: Output: Memory Map Cleared.
C674X_0: Output:
C674X_0: Output: Memory Map Setup Complete.
C674X_0: Output:
C674X_0: Output: PSC Enable Complete.
C674X_0: Output:
C674X_0: Output: PLL0 init done for Core:456MHz, EMIFA:114MHz
C674X_0: Output: DDR initialization is in progress....
C674X_0: Output: PLL1 init done for DDR:156MHz
C674X_0: Output: Using DDR2 settings
C674X_0: Output: DDR2 init for 156 MHz is done
C674X_0: Output:
C674X_0: Output: DSP Wake Complete.(Only For OMAPL138)
C674X_0: Output:
C674X_0: Output: PRU Wake Complete.
```

图 13

2.1.5 查看 CPU 版本号

点击菜单"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 out 镜像文件格式转换时将会用到。此处查询到的 DSP 核版本号为：d800k008，如下图：

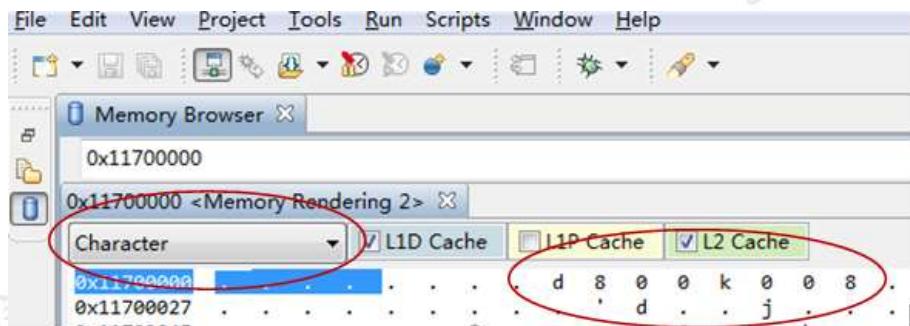


图 14

2.1.6 加载文件烧写程序

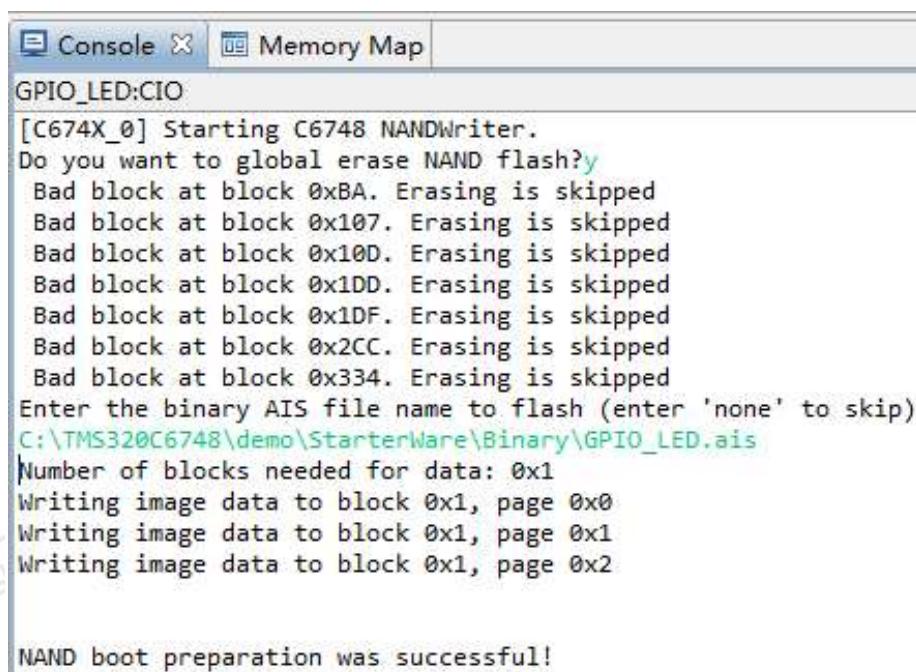
点击"Run->Load->Load Program"，选择光盘中 images 目录下的 C6748_NandWrite.out 或 C6748_NandWritev2.out 文件，点击 OK。接着点击绿色三角启动按键 。

C6748_NandWritev2.out 使用了 Load Memory 方式，大大提高了烧写速度，其烧写速度为使用 C6748_NandWrite.out 文件的 10 倍左右。

2.1.7 基于仿真器烧写程序到 NAND FLASH

(1) 使用 C6748_NandWrite.out 文件

以 LED 流水灯程序为例，将光盘"demo\StarterWare\Binary\GPIO_LED.ais"拷贝到一个非中文路径，在 Console 窗口"Do you want to global erase NAND FLASH?"提示行后面输入 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需要烧写的 DSP 程序镜像 GPIO_LED.ais 的完整路径，并按回车键。等待一会会出现"NAND boot preparation was successful!"即烧写成功，如下图所示：



```
Console Memory Map
GPIO_LED:CIO
[C674X_0] Starting C6748 NANDWriter.
Do you want to global erase NAND flash?y
Bad block at block 0xBA. Erasing is skipped
Bad block at block 0x107. Erasing is skipped
Bad block at block 0x10D. Erasing is skipped
Bad block at block 0x1DD. Erasing is skipped
Bad block at block 0x1DF. Erasing is skipped
Bad block at block 0x2CC. Erasing is skipped
Bad block at block 0x334. Erasing is skipped
Enter the binary AIS file name to flash (enter 'none' to skip)
C:\TMS320C6748\demo\StarterWare\Binary\GPIO_LED.ais
Number of blocks needed for data: 0x1
Writing image data to block 0x1, page 0x0
Writing image data to block 0x1, page 0x1
Writing image data to block 0x1, page 0x2

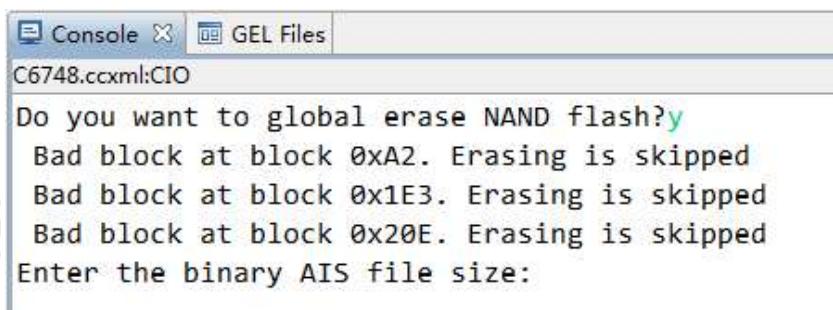
NAND boot preparation was successful!
```

图 15

烧写完 GPIO_LED.ais 镜像后，拔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，这是 NAND FLASH 启动档位，接着上电就可以看到开发板 LED 流水灯效果。

(2) 使用 C6748_NandWritev2.out 文件

以 LED 流水灯程序为例，将光盘"demo\StarterWare\Binary\GPIO_LED.ais"拷贝到一个非中文路径，在 Console 窗口"Do you want to global erase NAND FLASH?"提示行后面输入 y，按回车键进行擦除 NAND FLASH。擦除完后，会提示"Enter the binary AIS file size:"，如下图所示：



```
Console GEL Files
C6748.ccxml:CIO
Do you want to global erase NAND flash?y
Bad block at block 0xA2. Erasing is skipped
Bad block at block 0x1E3. Erasing is skipped
Bad block at block 0x20E. Erasing is skipped
Enter the binary AIS file size:
```

图 16