



OMAPL138 基于 CCS 的 DSP 程序开发入门教程

1. 工程建立前准备工作.....	4
1.1 检查开发板硬件	4
1.2 查看仿真器驱动是否正常安装	4
1.3 设置工程配置文件信息	5
1.4 测试仿真器是否正常连接	7
1.5 加载 GEL 文件	7
1.6 初始化 OMAPL138 和唤醒 DSP	8
2. CCS 工程建立和编译.....	10
2.1 新建工程	10
2.2 编写程序	11
2.3 编译和运行 DSP 程序	12
2.4 更改程序并再此加载运行	14
3. StarterWare 软件包介绍	16
3.1 StarterWare 简介	16

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

1 / 55



3.2	StarterWare 软件获取和安装	17
3.3	StarterWare 目录说明	18
4.	led 裸机程序开发	19
4.1	工程建立	19
4.2	添加头文件和库文件	21
4.3	设置编译参数	22
4.4	源代码编写和解析	25
4.5	编译和运行 led 控制程序	34
4.6	使用现有 CCS 工程编译和运行 led 程序	35
5.	按键中断裸机程序演示和解析	39
5.1	导入工程	39
5.2	源代码解析	42
6.	基于 StarterWare 工程的 uart 串口通信程序演示	48
6.1	导入 StarterWare 工程	48
6.2	修改 CPU 频率参数	50

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



6.3	重新编译和运行程序	52
-----	-----------------	----

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

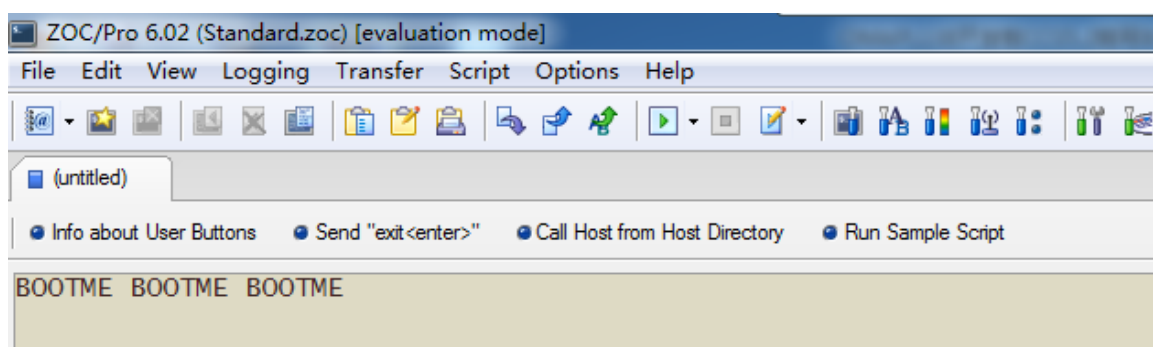
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



1. 工程建立前准备工作

1.1 检查开发板硬件

将开发板的拨码开关编号 5~1 (BOOT[4: 0]) 对应拨到 10100, 具体说明请查看底板拨码开关旁边的丝印或者附录 BOOTME 说明, ‘1’ 代表 ON, ‘0’ 代表 OFF, ‘x’ 代表任意置位, 这是 UART2 档位。RS232 串口线的 DB9 母头 (带孔) 插到 PC 机的 RS232 串口 DB9 公头, PC 机没有 RS232 串口的可以通过 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 口。设置 PC 机调试终端 COM 口, 波特率为 115200, 8N1, 无检验位。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。



1.2 查看仿真器驱动是否正常安装

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, 否则请正常安装仿真器的驱动程序。

联系人 : 朱先生

联系电话 : 13318712959

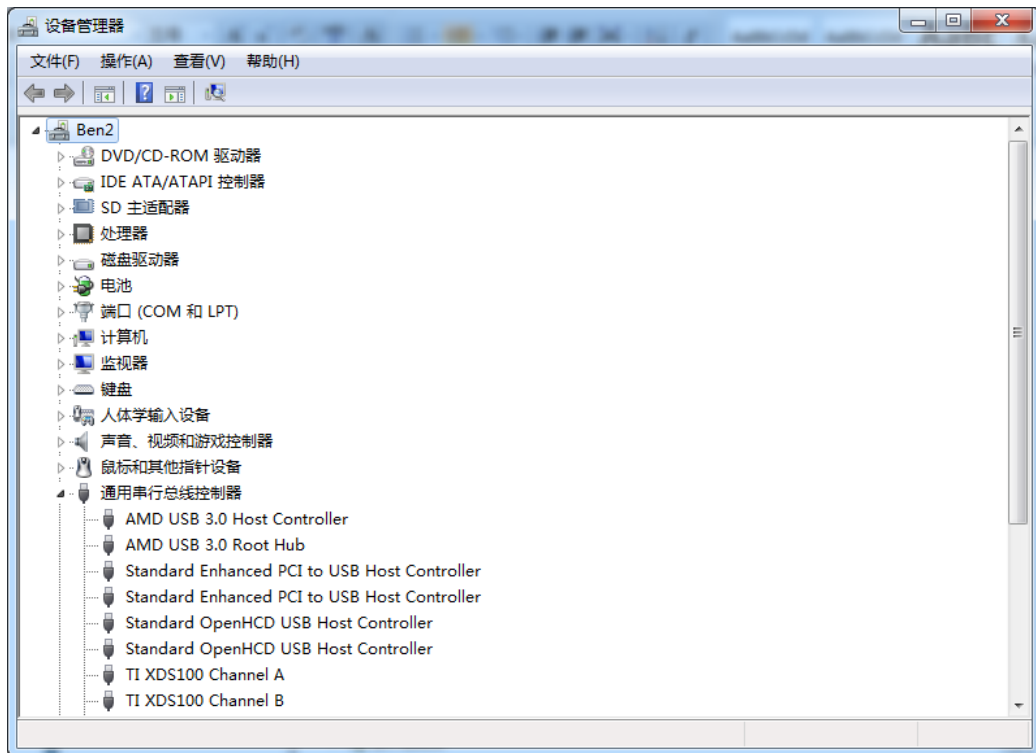
QQ : 2532609929

销售邮箱 : sales@tronlong.com

公司总机 : 020-89986280

公司网站 : www.tronlong.com

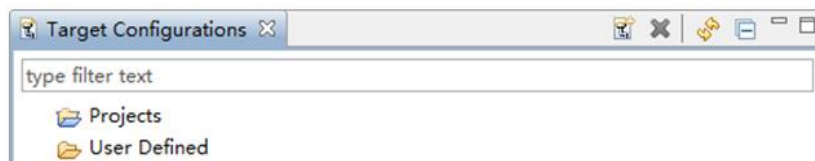
公司总部 : 广州市天河区五山路华南农业大学真维斯活动中心2楼



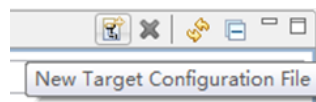
1.3 设置工程配置文件信息

备注：本文档的 CCS 版本是 CCS5.2.1。

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



点击右边第一项，新建配置文件。



输入工程名字：omap1138.ccxml，点击 finish。

联系人：朱先生

联系电话：13318712959

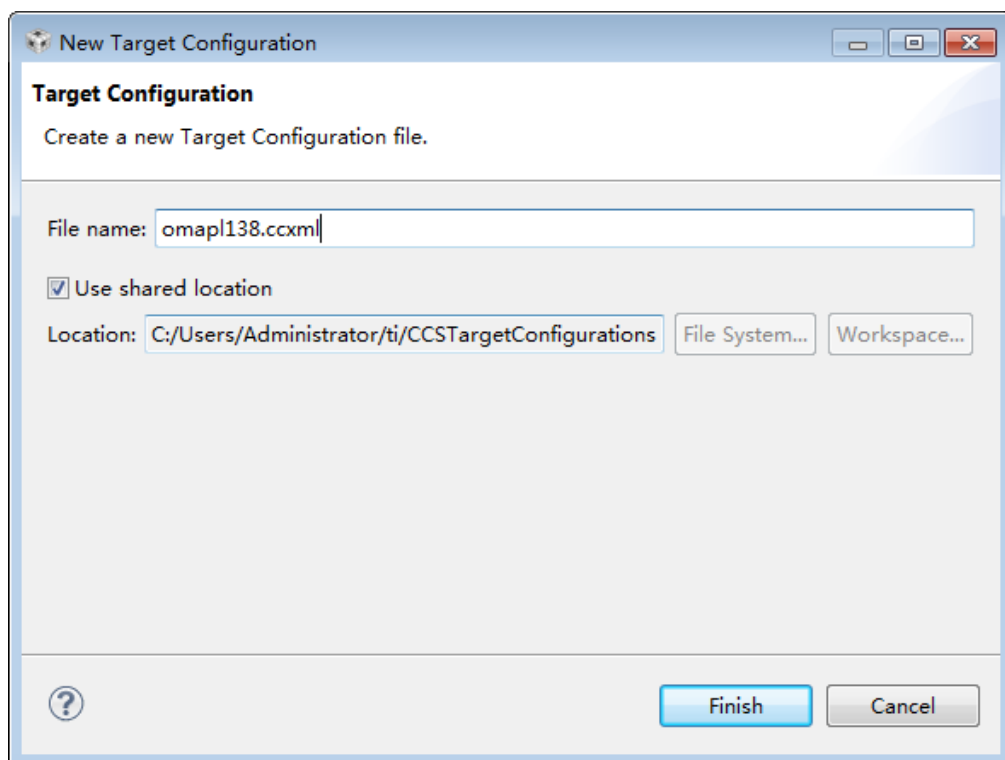
QQ：2532609929

销售邮箱：sales@tronlong.com

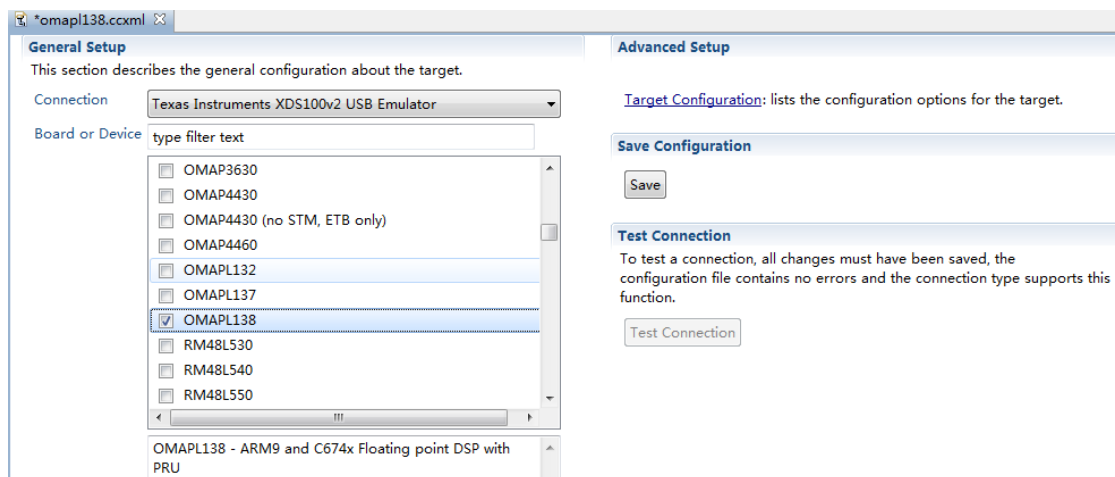
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



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。



若使用 XDS100V3 仿真器，设置方法请参考前面烧写 uboot 到 NAND FLASH 章节。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

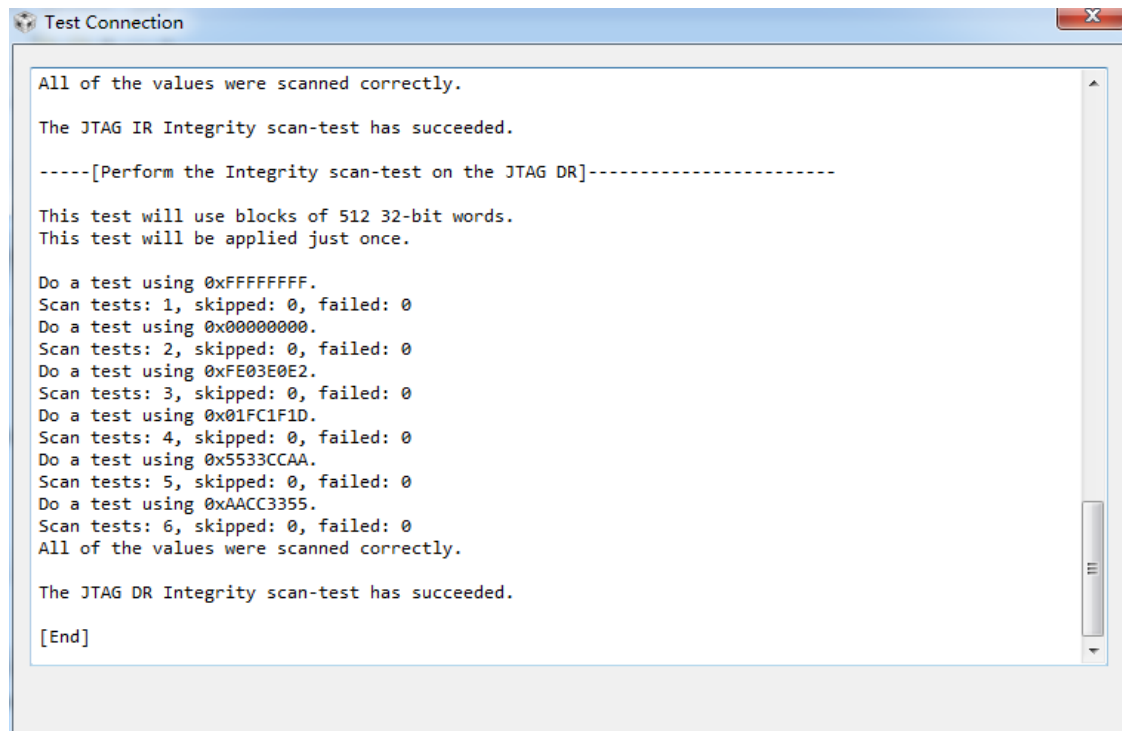
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

6 / 55

1.4 测试仿真器是否正常连接

开发板上电，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。



1.5 加载 GEL 文件

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



右击第四项“ARM9”，选择“Open GEL Files View”选项，右下角会弹出“GEL Files (ARM9)”对话框。在对话框内点击右键，选择“Load GEL...”。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

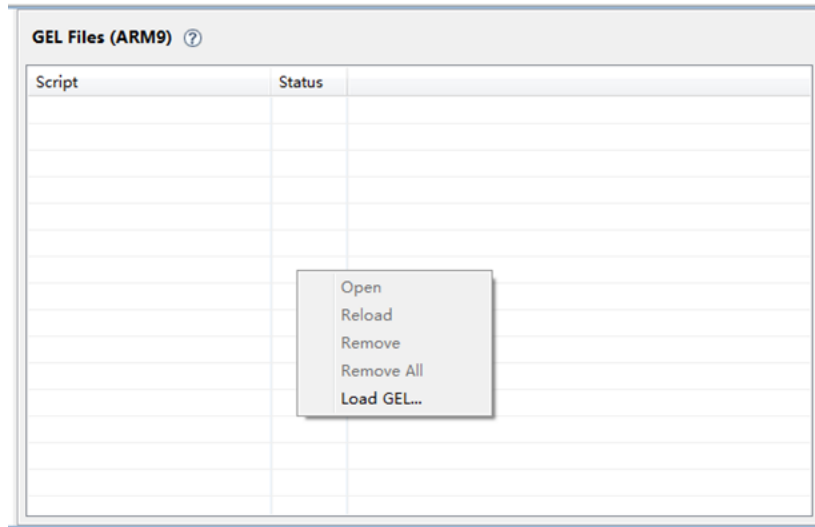
销售邮箱：sales@tronlong.com

公司总机：020-89986280

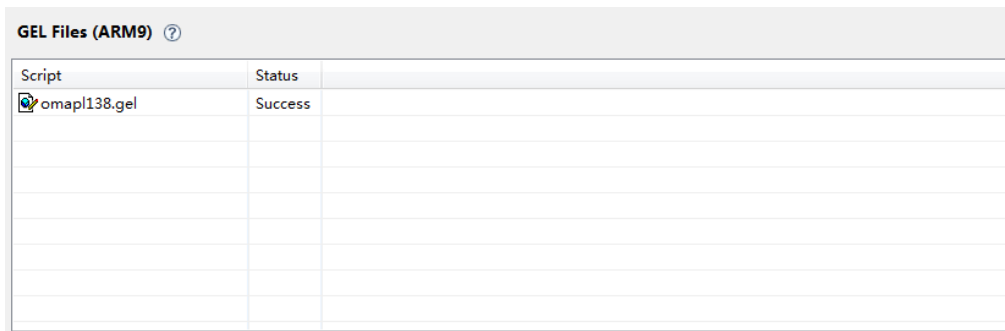
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

7 / 55

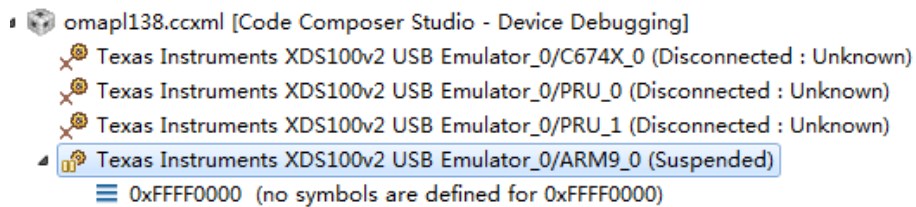


选择光盘中 images\omap138.gel 文件，点击确定，右下角的“GEL Files(ARM9)”对话框会出现“Success”提示语句，如下图：



1.6 初始化 OMAPL138 和唤醒 DSP

右击第四项“ARM9”，选择“Connect Target”选项，会显示“Suspend”状态。这说明 CCS 已经和 OMAPL138 的 ARM 核正常连接起来了。



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

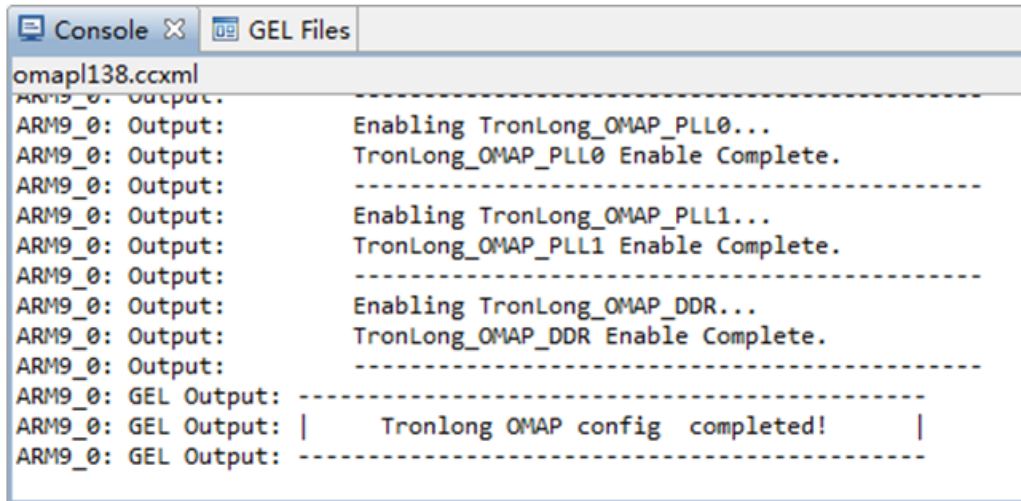
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

8 / 55

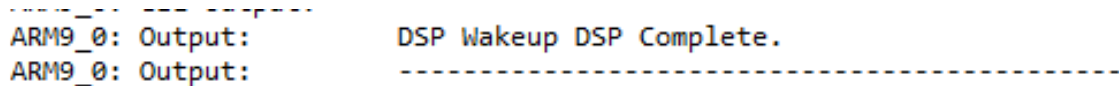
点击“Scripts->TronLong_OMAP-> TronLong_OMAP_ALL”，运行后会提示“Config Complete”，OMAPL138 初始化完成。



```

omap138.ccxml
ARM9_0: Output:
ARM9_0: Output: Enabling TronLong_OMAP_PLL0...
ARM9_0: Output: TronLong_OMAP_PLL0 Enable Complete.
ARM9_0: Output:
ARM9_0: Output: Enabling TronLong_OMAP_PLL1...
ARM9_0: Output: TronLong_OMAP_PLL1 Enable Complete.
ARM9_0: Output:
ARM9_0: Output: Enabling TronLong_OMAP_DDR...
ARM9_0: Output: TronLong_OMAP_DDR Enable Complete.
ARM9_0: Output:
ARM9_0: GEL Output: -----
ARM9_0: GEL Output: |      Tronlong OMAP config completed!      |
ARM9_0: GEL Output: -----
  
```

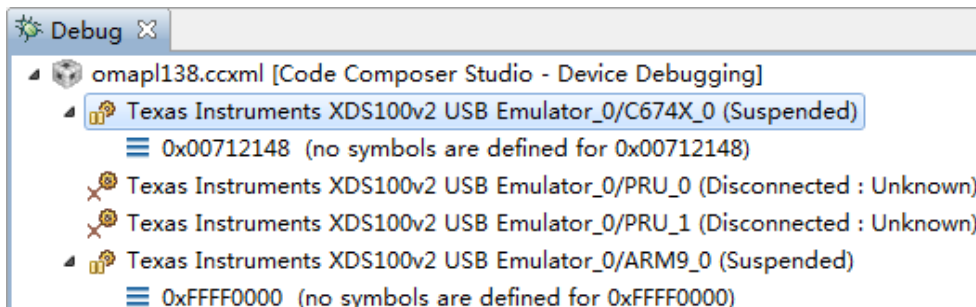
点击“Scripts->Wakeup DSP Core->Wakeup_DSP”，运行后会提示“DSP Wakeup DSP Complete.”，这说明 CCS 已经将 OMAPL138 的 DSP 核唤醒了，如下图所示：



```

ARM9_0: Output: DSP Wakeup DSP Complete.
ARM9_0: Output: -----
  
```

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。



```

Debug
omap138.ccxml [Code Composer Studio - Device Debugging]
  Texas Instruments XDS100v2 USB Emulator_0/C674X_0 (Suspended)
    0x00712148 (no symbols are defined for 0x00712148)
  Texas Instruments XDS100v2 USB Emulator_0/PRU_0 (Disconnected : Unknown)
  Texas Instruments XDS100v2 USB Emulator_0/PRU_1 (Disconnected : Unknown)
  Texas Instruments XDS100v2 USB Emulator_0/ARM9_0 (Suspended)
    0xFFFF0000 (no symbols are defined for 0xFFFF0000)
  
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

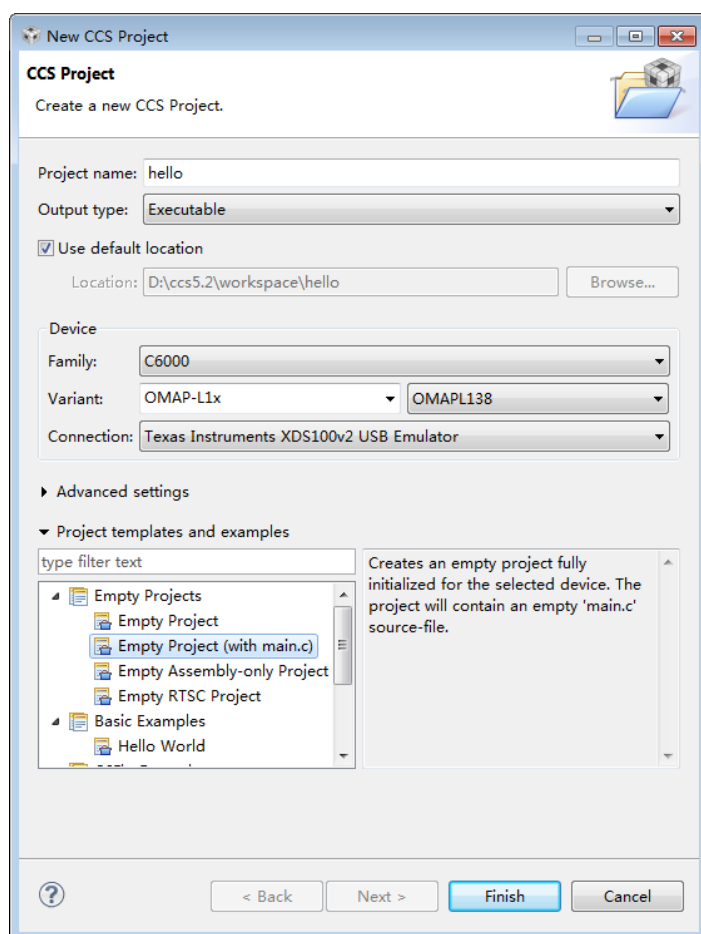
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

2. CCS 工程建立和编译

2.1 新建工程

打开 CCS5，点击“file->NEW->Project->Code Composer Studio->CCS Project”，然后点击“Next”，弹出对话框，并修改为如下：



Project name: hello，CCS 工程名字；

Family: C6000；

Variant: OMAP-L1x->OMAPL138；

Connection:XDS100V2，仿真器类型；

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

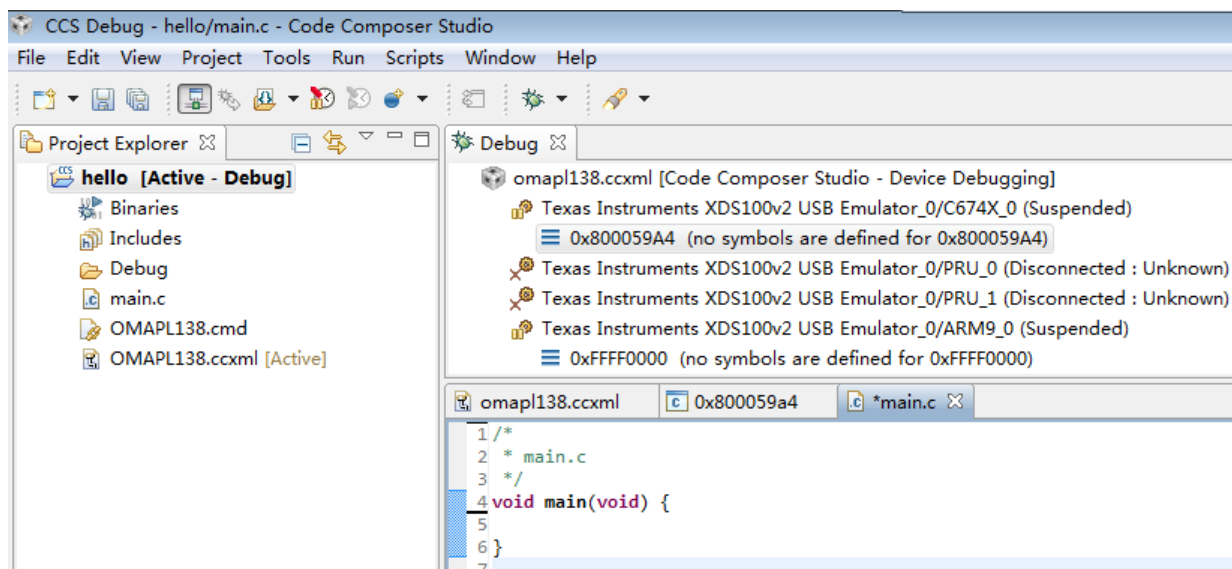
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

10 / 55



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：



2.2 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>
/*
 * main.c
 */
void main(void) {
    printf("*****Hello OMAPL138!.....www.tronlong.com*****");
}
}
```

编辑完成后如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

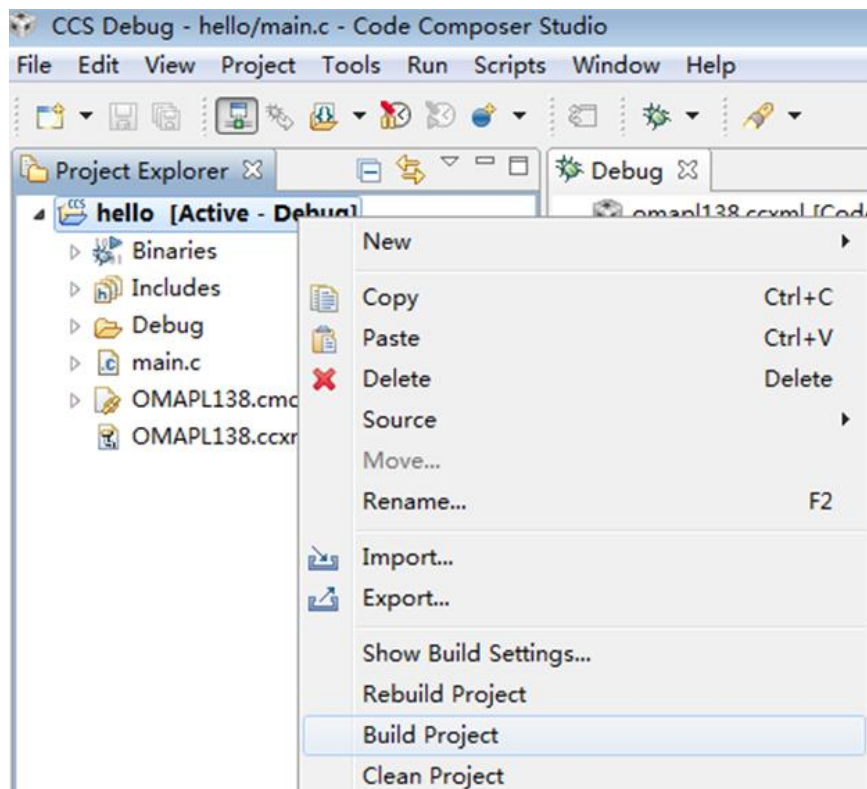
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

```
1#include <stdio.h>
2/*
3 * main.c
4 */
5void main(void) {
6    printf("*****Hello OMAPL138!.....www.tronlong.com*****");
7
8}
```

2.3 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：



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：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

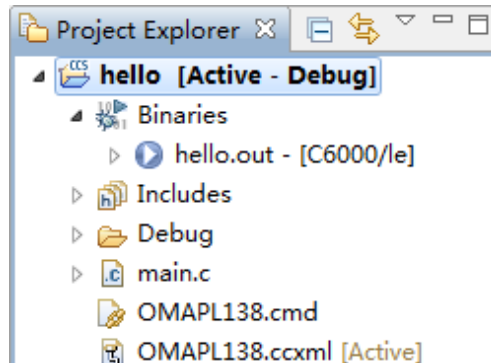
销售邮箱：sales@tronlong.com


公司总机：020-89986280

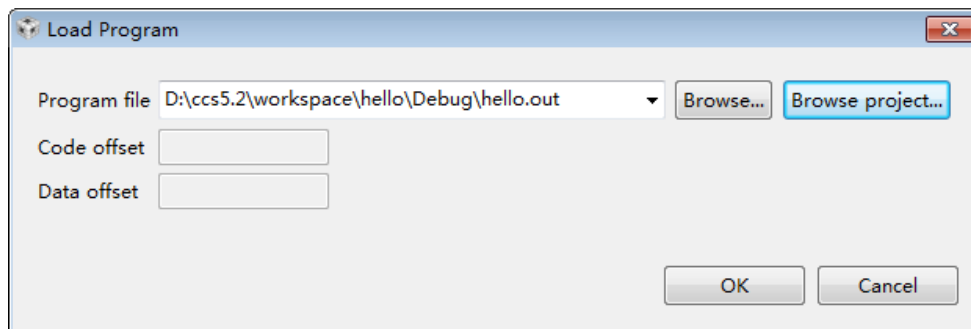
公司网站：www.tronlong.com


公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

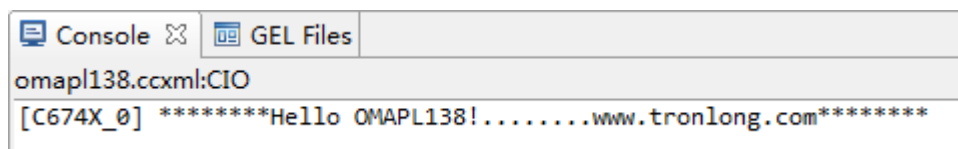
12 / 55



点击左上角的  load 键，选择要加载的 hello.out 文件



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。



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

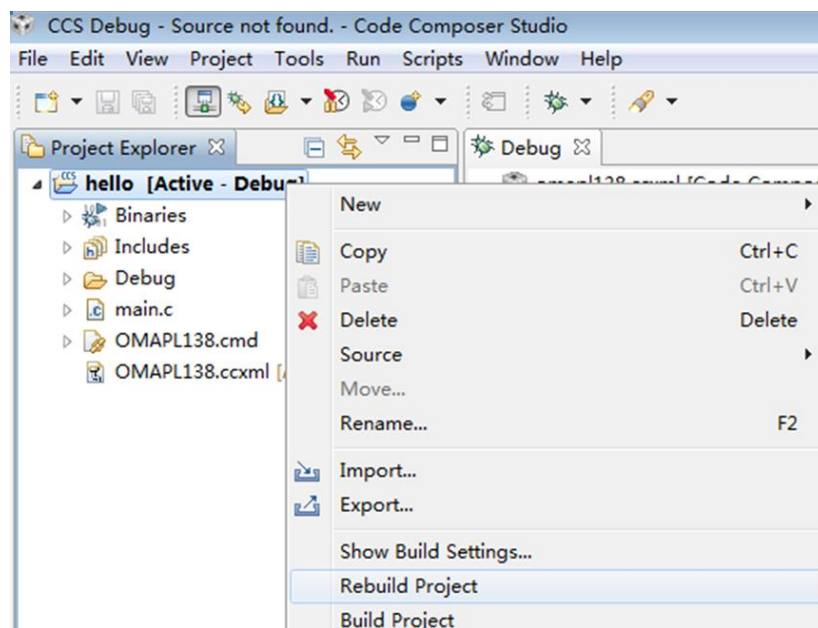
13 / 55


2.4 更改程序并再次加载运行

修改 DSP 程序，修改后如下：

```
1#include <stdio.h>
2/*
3 * main.c
4 */
5void main(void) {
6    printf("*****Hello OMAPL138!.....www.tronlong.com*****\n");
7    printf("*****Test Again!.....www.tronlong.com*****\n");
8
9}
```

按“Ctrl+S”保存修改，点击 CCS 菜单“View->Project Explorer”，弹出工程界面。右击工程，在弹出的菜单中点击“Rebuild Project”，这样会重新编译整个工程，产生新的.out 镜像文件。



点击左上角的  load 键，选择要加载的 hello.out 文件：

联系人：朱先生

联系电话：13318712959

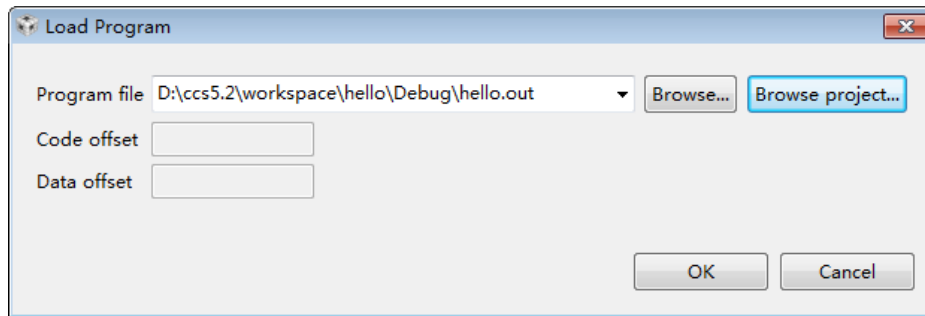
QQ：2532609929


销售邮箱：sales@tronlong.com

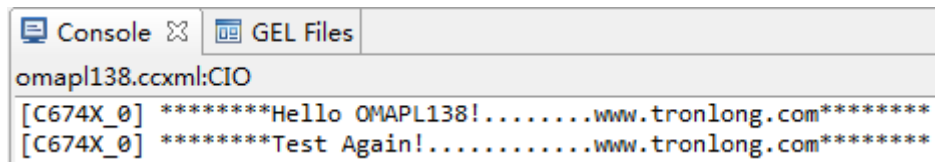
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



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 控制台有更改后的信息打印出来，如下图所示。



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

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



3. StarterWare 软件包介绍

3.1 StarterWare 简介

2011 年, 德州仪器(TI)宣布推出免费 StarterWare 软件包, 为 Sitara™ 32 位 ARM®微处理器(MPU)、C6000™数字信号处理器(DSP)以及 DSP+ARM 处理器开发人员提供可随时投入生产的用户友好型软件, 从而可大幅缩短学习时间, 加速嵌入式设计的开发进程。嵌入式软件编程占用产品开发大部分时间现已成为过去。StarterWare 可为应用开发人员提供无需使用操作系统的高灵活起点。此外, StarterWare 还可实现针对其它 TI 嵌入式器件的便捷移植。

TI 最新 StarterWare 软件包可简化 ARM MPU 与 DSP 的开发流程, 为配置和连接外设提供应用编程接口(API)。此外, StarterWare 还包含轻量级软件协议栈, 支持 USB、网络以及图形功能。StarterWare 配套提供的范例代码与应用手册可将开发时间缩短达 10 倍。

软件包不需要使用或学习复杂的操作系统, 从而可帮助开发人员加速产品上市进程。开发人员无需通过操作系统来使用硬件, 他们可对系统进行微调, 为 CPU、外设以及存储器实现最佳资源管理, 与 Linux®操作系统下的运行情况相比, 可将系统时延降低达 500 倍。在需要操作系统管理多个数据流的更复杂系统中, StarterWare 还可提供一个优化基础。许多实时操作系统(RTOS)厂商都计划在其 TI 嵌入式处理器产品中使用 StarterWare。

StarterWare 除了为 TI Sitara 32 位 ARM MPU 与 C6000 DSP 提供这些优势之外, TI DSP+ARM 处理器也可获得简化型低时延处理器间通信软件层的支持, 无需操作系统, 便可为优化系统性能在两个内核之间实现处理与输入 / 输出(I/O)任务的高效共享。

TI StarterWare 软件包可帮助客户在 TI 嵌入式器件之间实现便捷移植。软件设计人员可像 MCU 一样对 MPU 进行编程, 使用 IAR 与 Keil™等该熟悉的 MCU 工具链从 Stellaris® ARM Cortex™-M 微处理器(MCU)移植到 Sitara 32 位 ARM MPU。此外, 开发人员还可从 ARM MPU 扩展到 C6000 DSP 或 DSP + ARM 处理器, 因为该软件提供与 DSP 及 ARM MPU 相同的 API 调用。

联系人 : 朱先生

联系电话 : 13318712959

QQ : 2532609929

销售邮箱 : sales@tronlong.com

公司总机 : 020-89986280

公司网站 : www.tronlong.com

公司总部 : 广州市天河区五山路华南农业大学真维斯活动中心2楼

16 / 55



开发人员可立即采用针对每款 TI 器件定制的 StarterWare 软件包开始创新。适用于 TI AM18x ARM9™ MPU、OMAP-L138 DSP+ARM9 处理器以及 TMS320C6748 DSP 的 StarterWare 现已可供免费下载，预计未来几个月还将提供针对其它器件的软件包。

这些 StarterWare 软件包建立在 StellarisWare®软件套件的成功基础之上，可降低应用门槛，加速产品上市进程，并进一步推动 MCU 开发人员的创新。

3.2 StarterWare 软件获取和安装

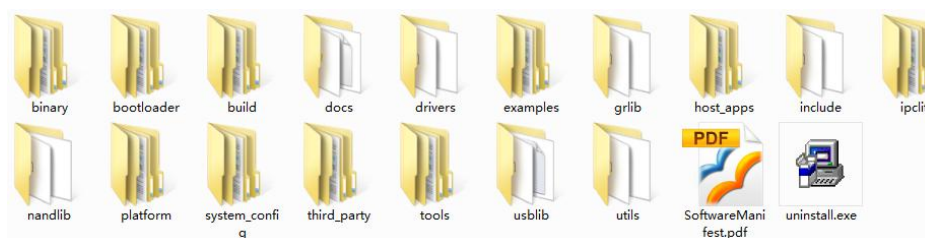
OMAPL138 StarterWare 软件包含有一系列例程和辅助工具等，对于裸机应用程序的快速开发有很大的帮助。特别是 examples 目录下有很多现成的 demo 程序，只要稍作改动即可作为自己产品的程序。此 Starterware 对 OMAPL138 的 DSP 和 ARM 端均适用。其下载地址为：

http://software-dl.ti.com/dsps/dsps_public_sw/c6000/starterware/01_10_04_01/index_FDS.html

光盘中 tools 目录下有 StarterWare 在 Windos 下的软件安装包：

OMAPL138_StarterWare_1_10_04_01-Setup.exe

在 Windows 下双击即可安装，安装后如下图所示。（**注意：**StarterWare 务必安装在默认路径“C:\ti\OMAPL138_StarterWare_1_10_04_01”目录，否则在 CCS 加载 StarterWare 工程时会出错。）



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



3.3 StarterWare 目录说明

以下是 StarterWare 各个目录的说明：

- (1) binary: 演示程序的可执行镜像。
- (2) bootloader: 启动代码。
- (3) build: 演示程序工程文件。
- (4) docs: StarterWare 源码说明文档。
- (5) drivers: 驱动源码
- (6) examples: 演示程序源码。
- (7) grlib: 图形库文件的源码和头文件。
- (8) host_apps: PC 机（开发主机）和 OMAPL138 平台演示程序的交互程序。
- (9) include: StarterWare 演示程序用到的头文件。
- (10) ipclite: OMAP-L138 DSP+ARM 双核通信演示程序。
- (11) nandlib: NAND FLASH 库文件源码。
- (12) platform: 平台相关的库文件，例如 evmOMAPL138 和 lcdkOMAPL138 开发板。
- (13) system_config: 系统配置的库文件，例如中断和缓存。
- (14) third_party: 第三方提供的程序。
- (15) tools: 各种开发工具，例如 GEL 文件、SFH、AISgen 工具。
- (16) usbllib: usb 库文件源码。
- (17) utils: StarterWare 通用库文件源码。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



4. led 裸机程序开发

本小结将讲解如何利用 TI 给的 StarterWare 软件包开发一个基于 DSP C6748 的 led 跑马灯的程序。

4.1 工程建立

光盘中有 DSP led 程序镜像和源码，路径为：

led.out: demo\dsp\led\Debug\led.out//DSP 镜像

main.c: demo\dsp\led\main.c//DSP 程序源码

在工程建立前，请先使用 gel 文件唤醒 DSP 核，具体步骤参考《1. 工程建立前准备工作》。然后点击“file->NEW->Project->Code Composer Studio->CCS Project”，然后点击“Next”，弹出对话框，并修改为如下：

联系人：朱先生

联系电话：13318712959

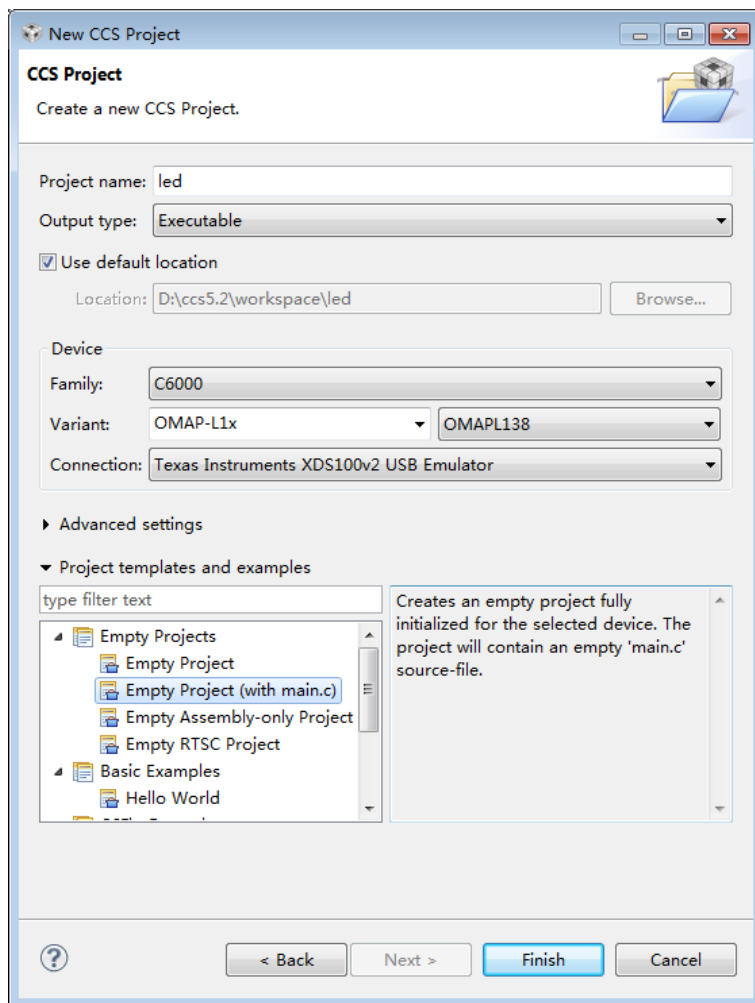
QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



Project name: led，CCS 工程名字；

Use default location:打勾，使用默认的路径。请记住此路径，将会在添加库文件时用到。

Family: C6000；

Variant: OMAP-L1x->OMAPL138；

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：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

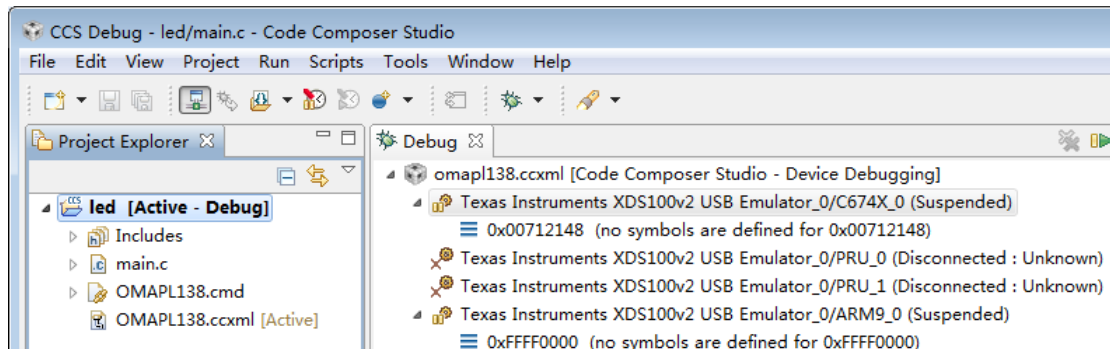
销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

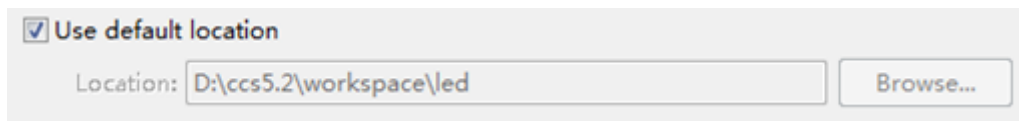
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

20 / 55



4.2 添加头文件和库文件

将 StarterWare 安装路径下的“include”目录复制到 led 工程的工作目录下。“include”目录的内容是 OMAPL138 开发所需的头文件。led 的工程目录在新建工程时已经设置好工作目录了，例如本例的默认路径，如下图所示：



在 led 工程的工作目录中新建库文件的存放目录“library”，然后将 StarterWare 安装路径下的以下文件复制到新建的“library”目录：

binary\c674x\cgt_ccs\omapl138\drivers\Release\drivers.lib

drivers.lib 是 OMAPL138 一般外设开发所需的库文件，其源文件在 StarterWare 的 drivers 目录下。比如 StarterWare 的“include”目录下的头文件“gpio.h”，它里面有很多相关函数声明，具体实现都是在 StarterWare 的“drivers”目录下的“gpio.c”文件里面。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

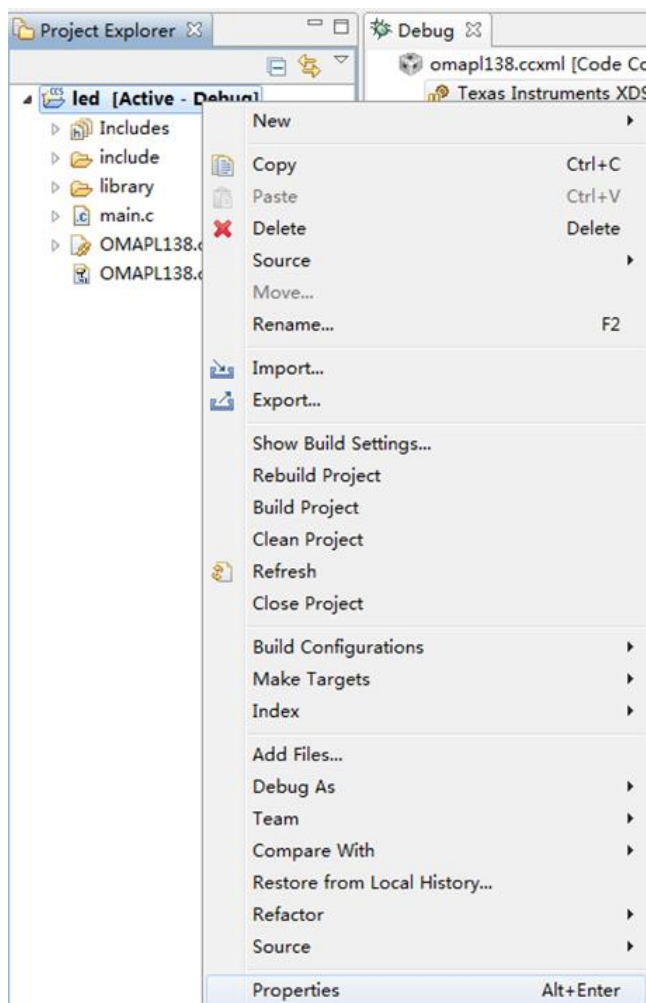
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

4.3 设置编译参数

右击工程，在弹出的菜单中点击 Properties（属性），如下图所示：



在弹出的对话框中点击“General”，在“Output format”处选择“eabi(ELF)”，表明 DSP 程序的输出镜像的格式是 ELF（StarterWare 的库默认编译的是 ELF 文件），如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

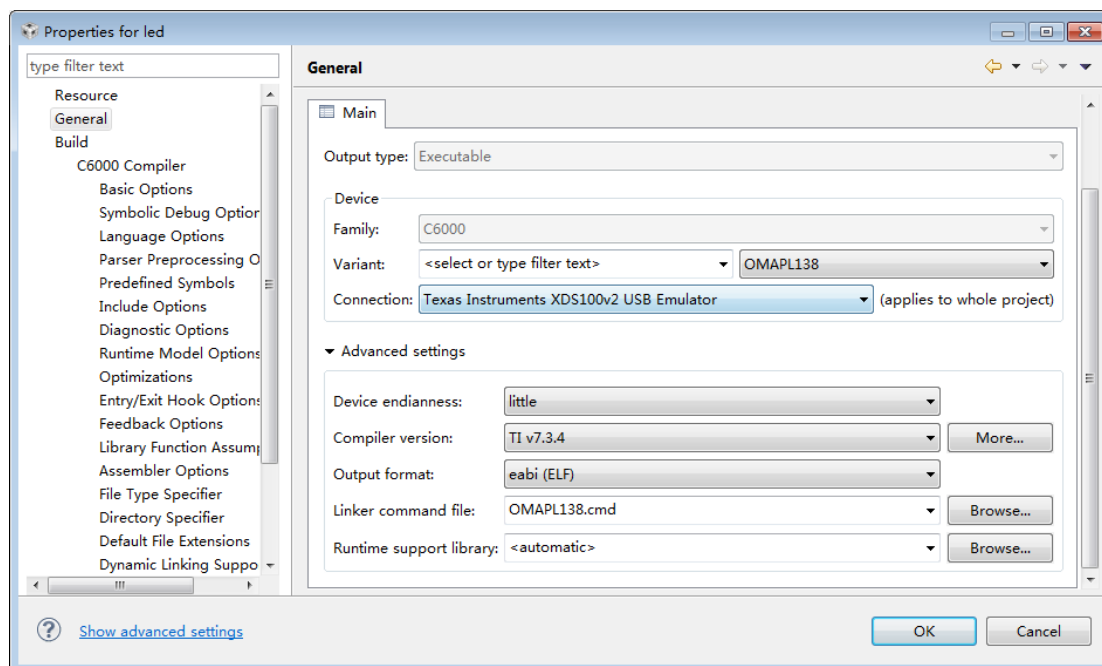
销售邮箱：sales@tronlong.com

公司总机：020-89986280

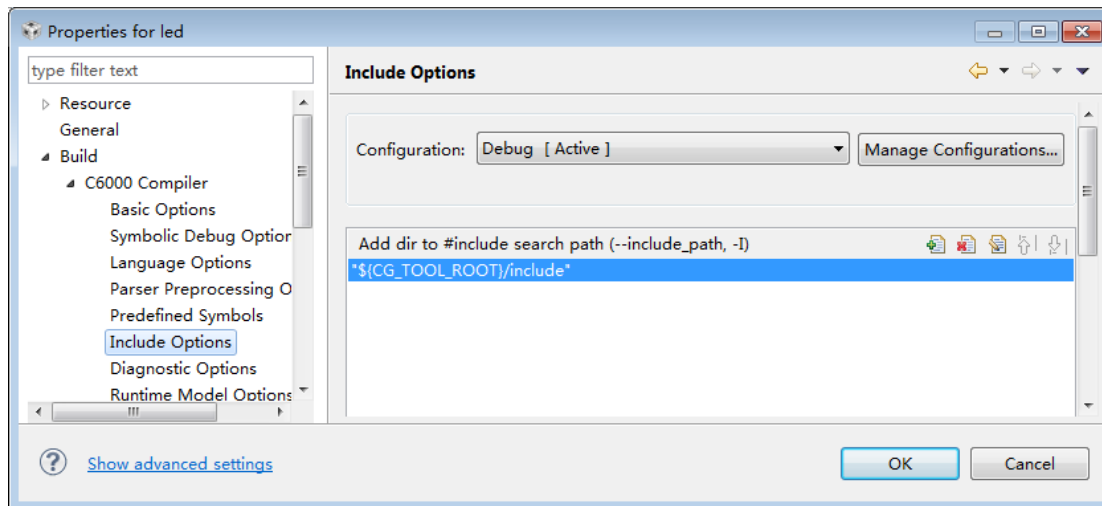
公司网站：www.tronlong.com


公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

22 / 55



点击“Build->C6000 Compiler->Include Options”，弹出如下界面：



点击右边的  符号添加要包含的头文件路径，弹出如下对话框：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

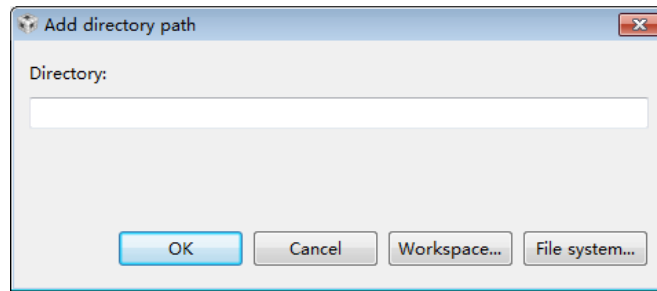
销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

23 / 55

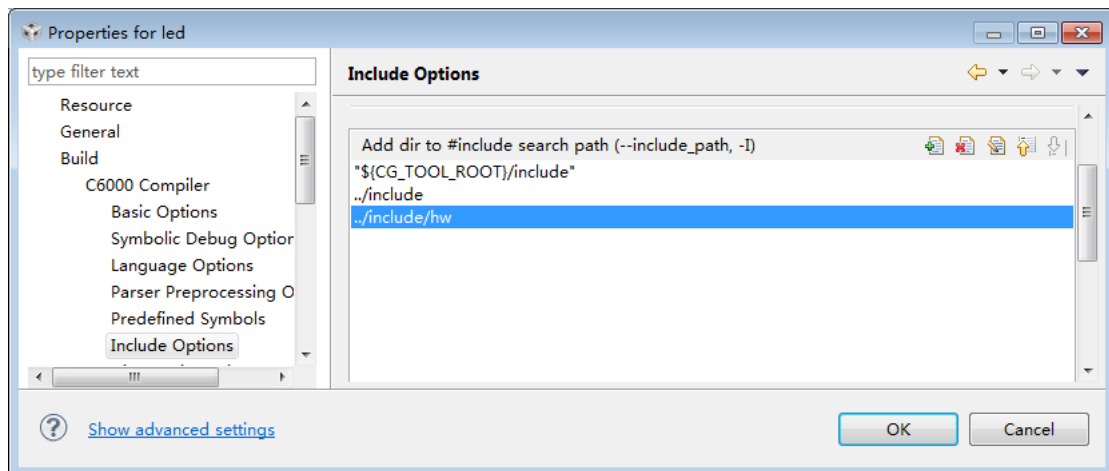


在空白处先后两次输入“../include”和“../include/hw”，以下为解释：

../include: gpio.h 头文件所在路径，工程目录下的“include”目录，led 程序程序将用到。

../include/hw: hw_types.h 和其他头文件所在路径，工程目录下的“include/hw”目录，led 程序程序将用到。

增加后如下图所示：



点击 OK，完成编译参数设置。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



4.4 源代码编写和解析

将光盘 demo\dsp\led\下的 led 程序源代码 main.c 的内容复制到工程的 main.c 文件中，直接用文件覆盖也可行。CCS 中设置为 12 号字体时，中文注释会比较美观。以下为此函数的解析：

```
/* 头文件 */
#include "hw_types.h"           // HWREG (x)
#include "hw_syscfg0_OMAPL138.h" // SYSCFG0_PINMUX(x)
#include "soc_OMAPL138.h"       // SOC_GPIO_0_REGS
#include "gpio.h"               // GPIO_DIR_OUTPUT

/* 函数声明 */
void GPIOBankPinInit(void); //GPIO初始化函数声明

/* 主函数 */
void main(void)
{
    unsigned int i;

    GPIOBankPinInit(); //GPIO初始化

    /* 跑马灯控制程序 */
    for(;;)
    {
        for(i=0x00FFFFFF;i>0;i--); //延时
        GPIOPinWrite(SOC_GPIO_0_REGS, 3, GPIO_PIN_LOW); //D10灭,GPIO0[2]
        GPIOPinWrite(SOC_GPIO_0_REGS, 1, GPIO_PIN_HIGH); //D7亮,GPIO0[0]

        for(i=0x00FFFFFF;i>0;i--); //延时
        GPIOPinWrite(SOC_GPIO_0_REGS, 1, GPIO_PIN_LOW); //D7灭,GPIO0[0]
        GPIOPinWrite(SOC_GPIO_0_REGS, 6, GPIO_PIN_HIGH); //D6亮,GPIO0[5]

        for(i=0x00FFFFFF;i>0;i--); //延时
        GPIOPinWrite(SOC_GPIO_0_REGS, 6, GPIO_PIN_LOW); //D6灭,GPIO0[5]
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



```
GPIOPinWrite(SOC_GPIO_0_REGS, 2, GPIO_PIN_HIGH); //D9亮,GPIO0[1]

for(i=0x00FFFFFF;i>0;i--); //延时
GPIOPinWrite(SOC_GPIO_0_REGS, 2, GPIO_PIN_LOW); //D9灭,GPIO0[1]
GPIOPinWrite(SOC_GPIO_0_REGS, 3, GPIO_PIN_HIGH); //D10亮,GPIO0[2]
}
}

/* GPIO初始化 */
void GPIOBankPinInit(void)
{
    unsigned int PINMUX_1_VAL; //局部变量

    /* 读取PINMUX1寄存器的值 */
    PINMUX_1_VAL = HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(1));

    /* 将底板4个用户led的对应管脚复用为GPIO, 即GPIO0[0]、GPIO0[1]、GPIO0[2]、GPIO0[5] */
    /*
    PINMUX_1_VAL = (PINMUX_1_VAL & 0x000ff0ff) | 0x88800800;

    /* 将设置后的值写回PINMUX1寄存器 */
    HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(1)) = PINMUX_1_VAL;

    /*配置底板4个用户led的对应管脚为输出, 以下为各组 GPIO第一个管脚 的对应值(范围1~144)
    * GPIO0[0] 1 , GPIO1[0] 17
    * GPIO2[0] 33 , GPIO3[0] 49
    * GPIO4[0] 65 , GPIO5[0] 81
    * GPIO6[0] 97 , GPIO7[0] 113
    * GPIO8[0] 129 */
    GPIODirModeSet(SOC_GPIO_0_REGS, 1, GPIO_DIR_OUTPUT); //GPIO0[0]
    GPIODirModeSet(SOC_GPIO_0_REGS, 2, GPIO_DIR_OUTPUT); //GPIO0[1]
    GPIODirModeSet(SOC_GPIO_0_REGS, 3, GPIO_DIR_OUTPUT); //GPIO0[2]
    GPIODirModeSet(SOC_GPIO_0_REGS, 6, GPIO_DIR_OUTPUT); //GPIO0[5]
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

```

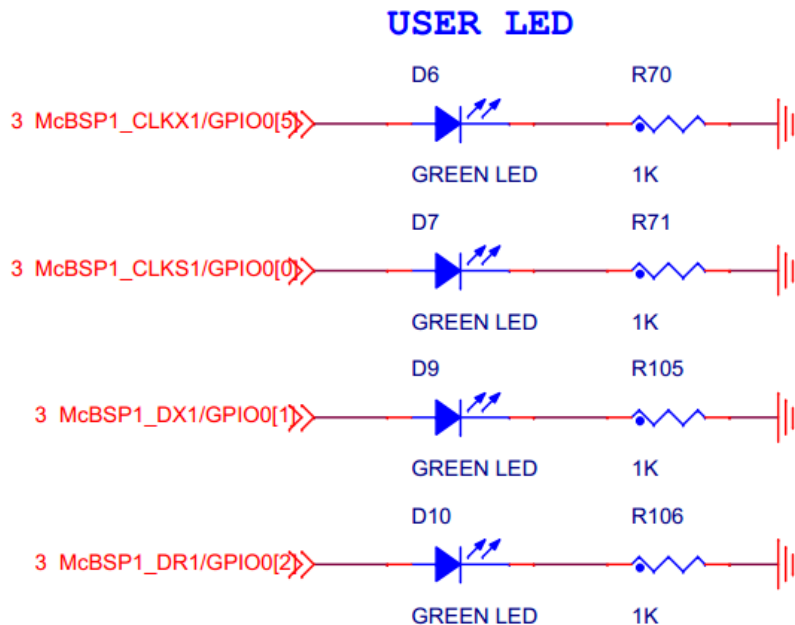
/* 初始化底板4个用户led状态为灭 */
GPIOPinWrite(SOC_GPIO_0_REGS, 1, GPIO_PIN_LOW);//D7灭,GPIO0[0]
GPIOPinWrite(SOC_GPIO_0_REGS, 2, GPIO_PIN_LOW);//D9灭,GPIO0[1]
GPIOPinWrite(SOC_GPIO_0_REGS, 3, GPIO_PIN_LOW);//D10灭,GPIO0[2]
GPIOPinWrite(SOC_GPIO_0_REGS, 6, GPIO_PIN_LOW);//D6灭,GPIO0[5]
}

```

以上是 led 跑马灯控制程序，对于 GPIO 口的操作，有以下几点步骤：

(1) 查看开发板的原理图，找到与 led 连接的 GPIO 编号。

广州创龙电子的 OMAP-L138 开发板与 led 连接的 GPIO 分别是 GPIO0[0]、GPIO0[1]、GPIO0[2]、GPIO0[5]，如下图所示：



(2) 查看 OMAPL138 的数据手册，查找 led 对应的管脚的 PINMUX 寄存器的地址，将对应的管脚的寄存器中相应位设置为 GPIO 的工作模式。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

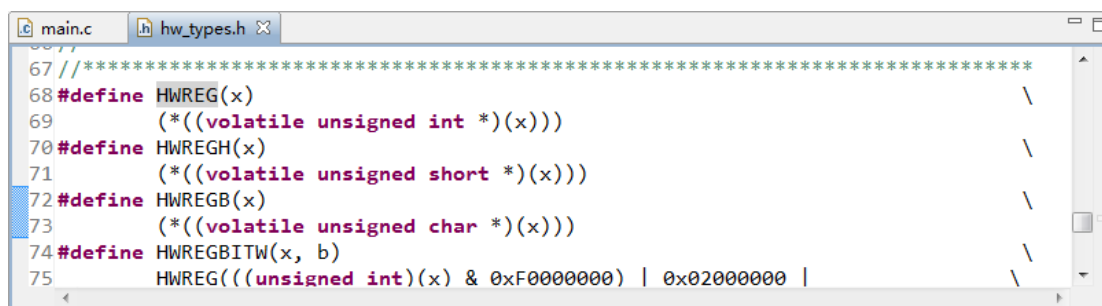
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

27 / 55

本例中使用 StarterWare 的库函数 HWREG(x)来设置 PINMUX 的值，例如：

HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(1))

HWREG(x)在工作目录的 include/hw/hw_types.h 的第 69 行定义了，如下图所示：

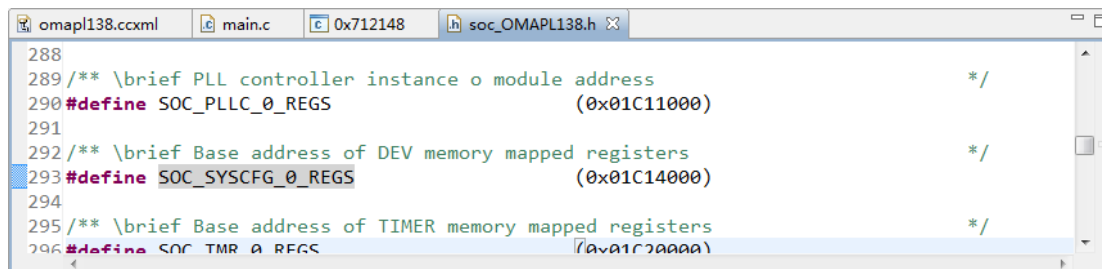


```

67 //*****
68 #define HWREG(x) \
69     (*((volatile unsigned int *) (x))) \
70 #define HWREGH(x) \
71     (*((volatile unsigned short *) (x))) \
72 #define HWREGB(x) \
73     (*((volatile unsigned char *) (x))) \
74 #define HWREGBITW(x, b) \
75     HWREG(((unsigned int)(x) & 0xF0000000) | 0x02000000 |

```

SOC_SYSCFG_0_REGS 是 PINMUX 的基地址，在工作目录的 include/hw/soc_OMAPL138.h 的 293 行定义了，如下图所示：



```

288
289 /** \brief PLL controller instance o module address */
290 #define SOC_PLLC_0_REGS (0x01C11000)
291
292 /** \brief Base address of DEV memory mapped registers */
293 #define SOC_SYSCFG_0_REGS (0x01C14000)
294
295 /** \brief Base address of TIMER memory mapped registers */
296 #define SOC_TMR_0_REGS (0x01C20000)

```

可以看出 SOC_SYSCFG_0_REGS 的值为 0x01C14000，类似的地址都可以在 datasheet 中找到，下图为 Rev.D 的 datasheet 中 SYSCFG0 的值：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

28 / 55



OMAP-L138 C6-Integra™ DSP+ARM® Processor

Check for Samples: [OMAP-L138](#)

0x01C1 4000	0x01C1 4FFF	4K	SYSCFG0	
0x01C1 5000	0x01C1 FFFF			
0x01C2 0000	0x01C2 0FFF	4K	Timer0	
0x01C2 1000	0x01C2 1FFF	4K	Timer1	
0x01C2 2000	0x01C2 2FFF	4K	I2C 0	
0x01C2 3000	0x01C2 3FFF	4K	RTC	
0x01C2 4000	0x01C3 FFFF			
0x01C4 0000	0x01C4 0FFF	4K	MMC/SD 0	
0x01C4 1000	0x01C4 1FFF	4K	SPI 0	
0x01C4 2000	0x01C4 2FFF	4K	UART 0	

(1) The DSP L2 ROM is used for boot purposes and cannot be programmed with application code

Copyright © 2009–2011, Texas Instruments Incorporated

Device Overview

23

[Submit Documentation Feedback](#)
Product Folder Link(s): [OMAP-L138](#)

SYSCFG0_PINMUX(1) 是 PINMUX1 的偏移地址，在工作目录的 include/hw 的 hw_syscfg0_OMAPL138.h 的第 70 行定义了，如下图所示：

```

68 #define SYSCFG0_MSTPRI1    (0x114)
69 #define SYSCFG0_MSTPRI2    (0x118)
70 #define SYSCFG0_PINMUX(n)  ((0x120 + (n * 4)))
71 #define SYSCFG0_SUSPSRC    (0x170)
72 #define SYSCFG0_CHIPSIG    (0x174)
73 #define SYSCFG0_CHIPSIG_CLR (0x178)
74 #define SYSCFG0_CFGCHIP0    (0x17C)
75 #define SYSCFG0_CFGCHIP1    (0x180)
76 #define SYSCFG0_CFGCHIP2    (0x184)

```

可以算出 PINMUX1 的偏移地址为 0x124，那么可以算出 PINMUX1 的值为：

$$\text{PINMUX1} = (0x01C14000 + 0x124) = 0x01C14124$$

类似的地址都可以在 datasheet 中找到，下图为技术参考手册 datasheet 中 PINMUX1 的值：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



OMAP-L138 DSP+ARM Processor

Technical Reference Manual



01C1 4120h	PINMUX0	Pin Multiplexing Control 0 Register	Privileged mode	Section 11.5.10.1
01C1 4124h	PINMUX1	Pin Multiplexing Control 1 Register	Privileged mode	Section 11.5.10.2
01C1 4128h	PINMUX2	Pin Multiplexing Control 2 Register	Privileged mode	Section 11.5.10.3
01C1 412Ch	PINMUX3	Pin Multiplexing Control 3 Register	Privileged mode	Section 11.5.10.4
01C1 4130h	PINMUX4	Pin Multiplexing Control 4 Register	Privileged mode	Section 11.5.10.5

⁽¹⁾ This register is for internal-use only.

SPRUH77A–December 2011
[Submit Documentation Feedback](#)

System Configuration (SYSCFG) Module 229

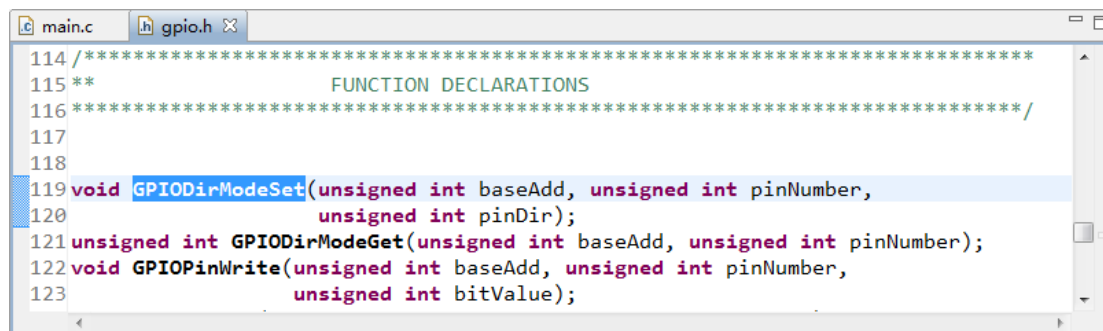
Copyright © 2011, Texas Instruments Incorporated

(3) 设置 GPIO 的方向寄存器。

本例程中使用 StarterWare 的库函数 `GPIONDirModeSet()` 将 GPIO 口配置为输出，例如：

```
GPIONDirModeSet(SOC_GPIO_0_REGS, 1, GPIO_DIR_OUTPUT)
```

`GPIONDirModeSet()` 在工作目录的 `include/gpio.h` 的第 119 行声明了，如下图所示：



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

30 / 55



SOC_GPIO_0_REGS 是 GPIO 的基地址，在工作目录的 include/hw/soc_OMAPL138.h 的第 384 行定义了，如下图所示：

```
main.c soc_OMAPL138.h
381 #define SOC_USB_0_PHY_REGS (0x01C14184)
382
383 /** \brief Base address of GPIO memory mapped registers */
384 #define SOC_GPIO_0_REGS (0x01E26000)
385
386 /** \brief Base address of PSC memory mapped registers */
387 #define SOC_PSC_1_REGS (0x01E27000)
388
389 /** \brief Base address of I2C memory mapped registers */
390 #define SOC_I2C_1_REGS (0x01E28000)
```

可以看出 SOC_GPIO_0_REGS 的值为 0x01E26000，类似的地址都可以在 datasheet 中找到，下图为 Rev.D 的 datasheet 中 GPIO 基地址的值：



OMAP-L138 C6-Integra™ DSP+ARM® Processor

Check for Samples: [OMAP-L138](#)

0x01E2 6000	0x01E2 6FFF	4K		GPIO	
0x01E2 7000	0x01E2 7FFF	4K		PSC 1	
0x01E2 8000	0x01E2 8FFF	4K		I2C 1	
0x01E2 9000	0x01E2 BFFF				
0x01E2 C000	0x01E2 CFFF	4K		SYSCFG1	
0x01E2 D000	0x01E2 FFFF				
0x01E3 0000	0x01E3 7FFF	32K		EDMA3 CC1	
0x01E3 8000	0x01E3 83FF	1K		EDMA3 TC2	
0x01E3 8400	0x01EF FFFF				
0x01F0 0000	0x01F0 0FFF	4K		eHRPWM 0	
0x01F0 1000	0x01F0 1FFF	4K		HRPWM 0	
0x01F0 2000	0x01F0 2FFF	4K		eHRPWM 1	
0x01F0 3000	0x01F0 3FFF	4K		HRPWM 1	
0x01F0 4000	0x01F0 5FFF				
0x01F0 6000	0x01F0 6FFF	4K		ECAP 0	
0x01F0 7000	0x01F0 7FFF	4K		ECAP 1	

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

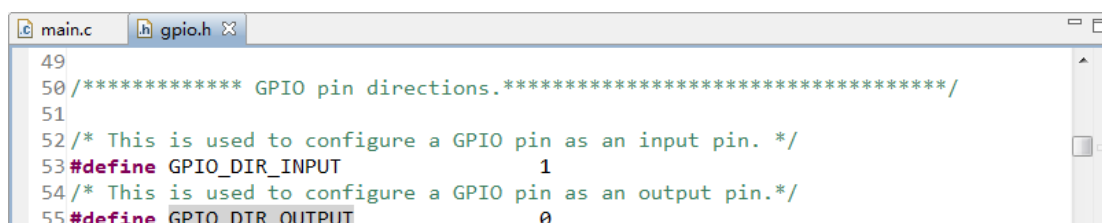
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

第二个参数“1”是 GPIO0[0]的编号, 以下为各组 GPIO 第一个管脚的对应值(范围 1~144):

GPIO0[0]=1, GPIO1[0]=17, GPIO2[0]=33, GPIO3[0]=49, GPIO4[0]=65, GPIO5[0]=81,

GPIO6[0]=97, GPIO7[0]=113, GPIO8[0]=129

GPIO_DIR_OUTPUT 定义 GPIO 的方向为输出, 在工作目录的 include/gpio.h 的第 53 和 55 行定义输出输入的标志, 如下图所示:



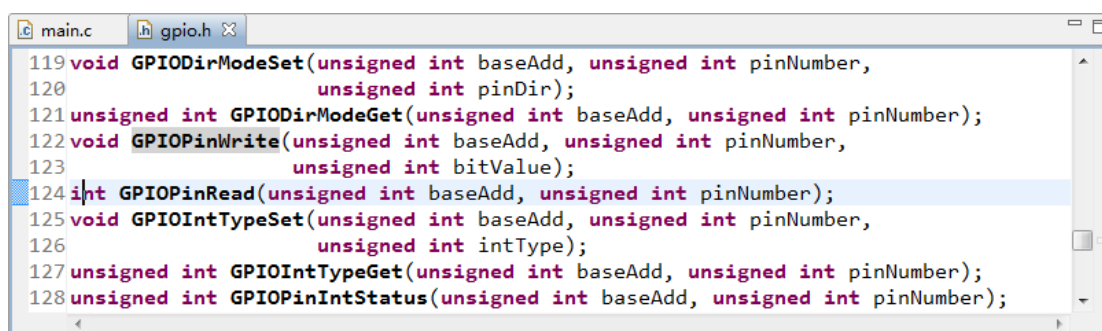
```
49
50 /***** GPIO pin directions. *****/
51
52 /* This is used to configure a GPIO pin as an input pin. */
53 #define GPIO_DIR_INPUT      1
54 /* This is used to configure a GPIO pin as an output pin. */
55 #define GPIO_DIR_OUTPUT     0
```

(4) 配置 GPIO 的数据寄存器。

本例中使用 StarterWare 的库函数 GPIOPinWrite()来设置 GPIO 的值, 例如:

GPIOPinWrite(SOC_GPIO_0_REGS, 1, GPIO_PIN_HIGH)

GPIOPinWrite()在工作目录的 include/gpio.h 的第 122 行声明了, 如下图所示:



```
119 void GPIODirModeSet(unsigned int baseAdd, unsigned int pinNumber,
120                    unsigned int pinDir);
121 unsigned int GPIODirModeGet(unsigned int baseAdd, unsigned int pinNumber);
122 void GPIOPinWrite(unsigned int baseAdd, unsigned int pinNumber,
123                  unsigned int bitValue);
124 int GPIOPinRead(unsigned int baseAdd, unsigned int pinNumber);
125 void GPIOIntTypeSet(unsigned int baseAdd, unsigned int pinNumber,
126                   unsigned int intType);
127 unsigned int GPIOIntTypeGet(unsigned int baseAdd, unsigned int pinNumber);
128 unsigned int GPIOPinIntStatus(unsigned int baseAdd, unsigned int pinNumber);
```

联系人 : 朱先生

联系电话 : 13318712959

QQ : 2532609929

销售邮箱 : sales@tronlong.com

公司总机 : 020-89986280

公司网站 : www.tronlong.com

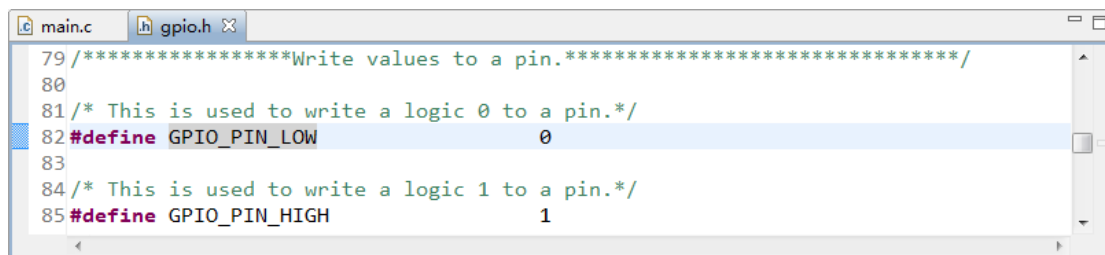
公司总部 : 广州市天河区五山路华南农业大学真维斯活动中心2楼

32 / 55



SOC_GPIO_0_REGS 是 GPIO 的基地址，第二个参数“1”是 GPIO0[0]的编号。

GPIO 的输出值通过 GPIO_PIN_HIGH（高电平）和 GPIO_PIN_LOW（低电平）来指定，在工作目录的 include/gpio.h 的 82 和 85 行定义了，如下图所示：



```
79/*****Write values to a pin.*****/
80
81/* This is used to write a logic 0 to a pin.*/
82#define GPIO_PIN_LOW      0
83
84/* This is used to write a logic 1 to a pin.*/
85#define GPIO_PIN_HIGH     1
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

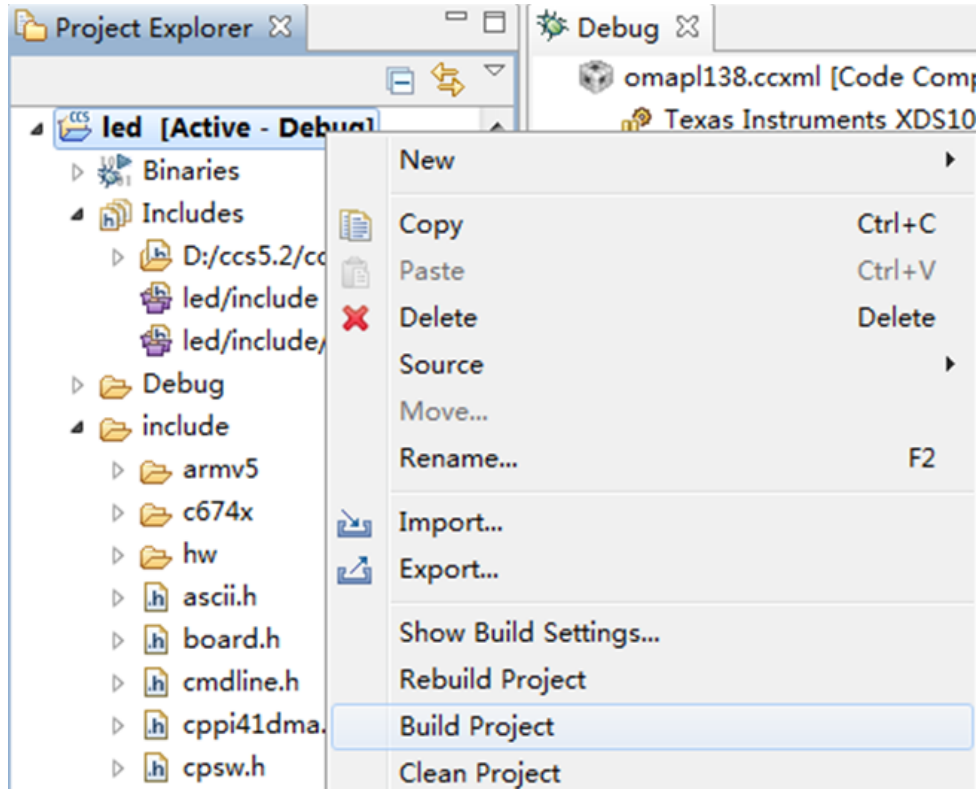
公司总机：020-89986280

公司网站：www.tronlong.com

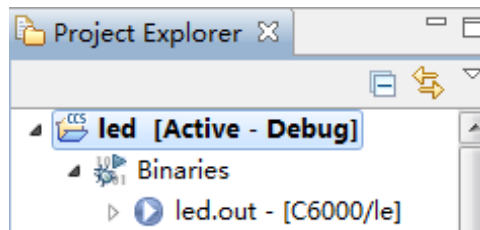
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼


4.5 编译和运行 led 控制程序

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：



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



点击左上角的  load 键，选择要加载的 led.out 文件，如下图所示：

联系人：朱先生

联系电话：13318712959

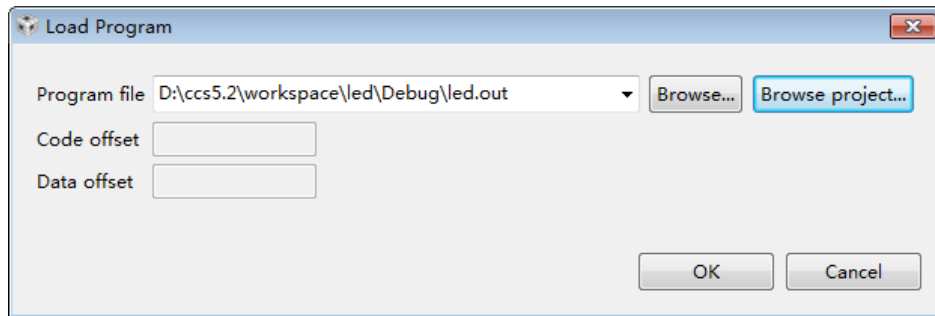
QQ：2532609929


销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

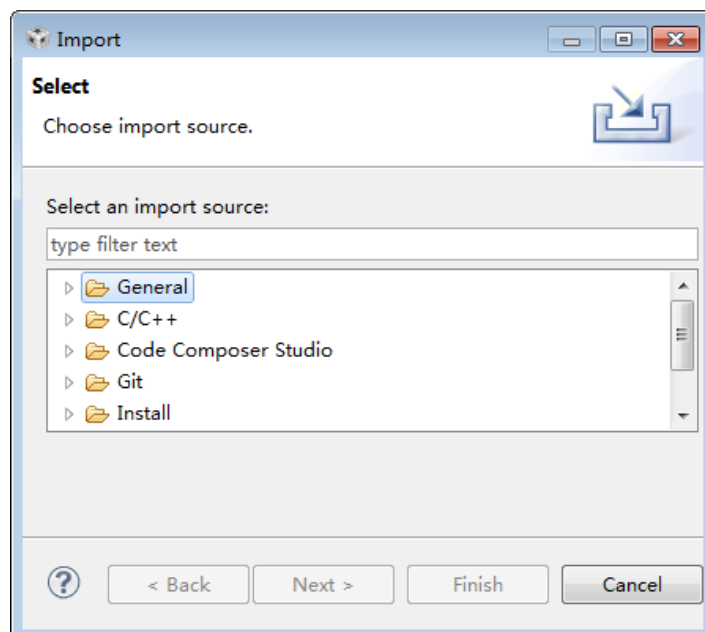
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



等待程序加载完成，然后点击程序运行按钮，可以发现底板的 led 流水灯执行了。按 F5 会进去函数里面，按 F6 就单步走。

4.6 使用现有 CCS 工程编译和运行 led 程序

光盘中有 led 程序的 CCS 工程，在光盘的路径为“demo\dsp\led”。用户可以基于现有工程直接编译和运行 led 程序。在工程建立前，请先使用 gel 文件唤醒 DSP 核，具体步骤参考《1. 工程建立前准备工作》。然后点击 CCS 菜单“File->Import”，弹出以下对话框：



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

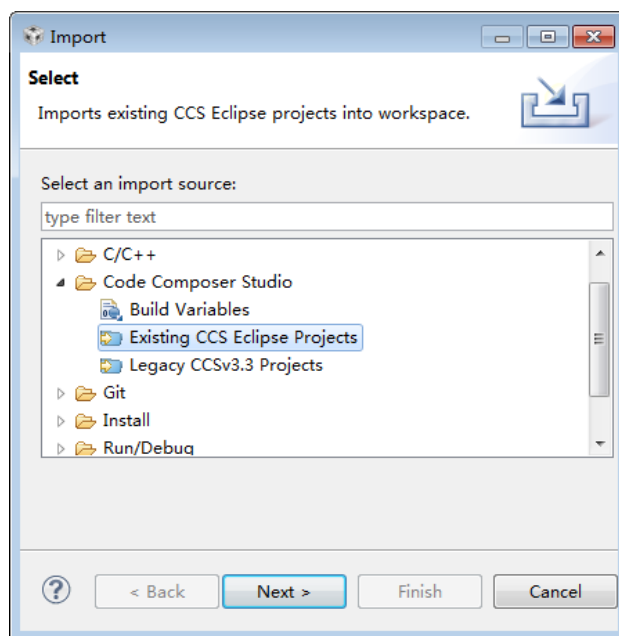
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

35 / 55

点击“Code Composer Studio->Existing CCS Eclipse Projects”后再点击“Next”，如下图所示：



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，选择后如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

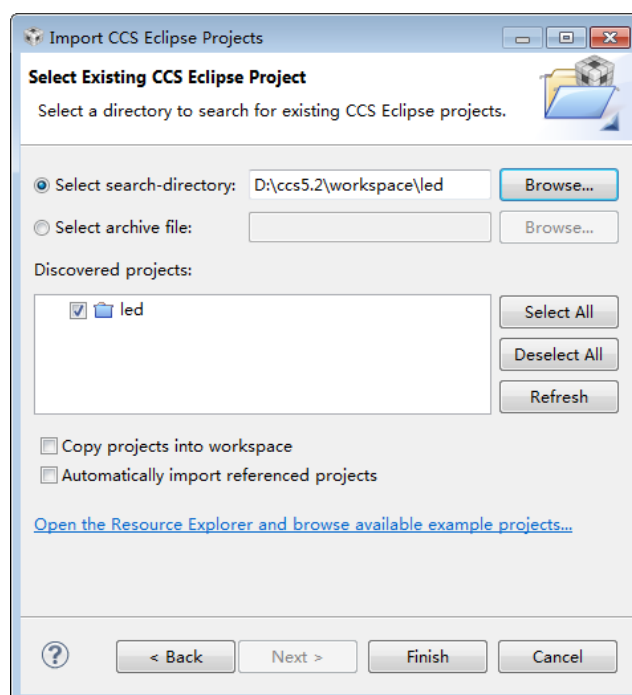
销售邮箱：sales@tronlong.com

公司总机：020-89986280

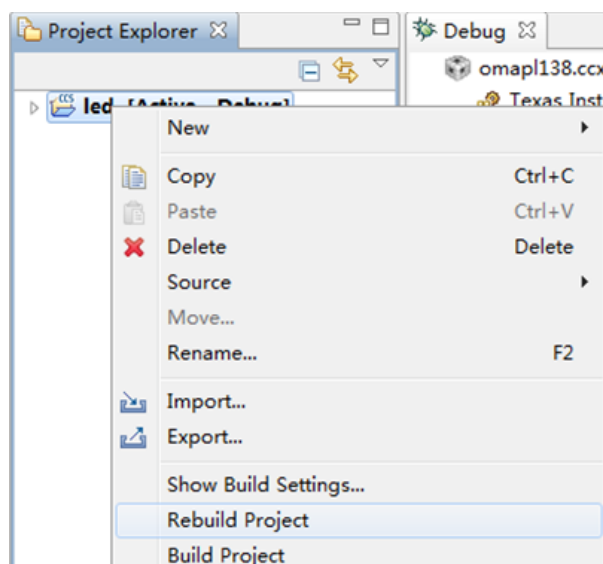
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

36 / 55



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：



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

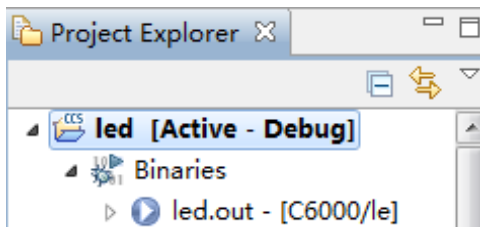
公司总机：020-89986280


公司网站：www.tronlong.com

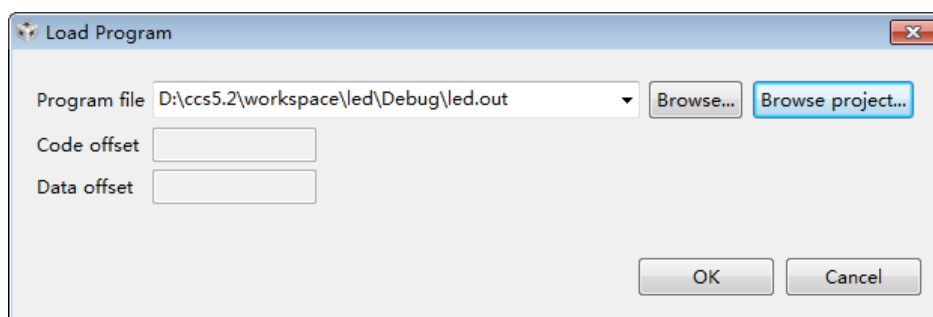
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼


37 / 55

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



点击左上角的  load 键，选择要加载的 led.out 文件，如下图所示：



等待程序加载完成，然后点击  程序运行按钮，可以发现底板的 led 流水灯执行了。
按 F5 会进去函数里面，按 F6 就单步走。

5. 按键中断裸机程序演示和解析

本小结将讲解如何利用现成的裸机程序，使用中断方式测试底板的按键，并详细解析此按键中断程序。

光盘中有按键中断程序镜像和源码，路径为：

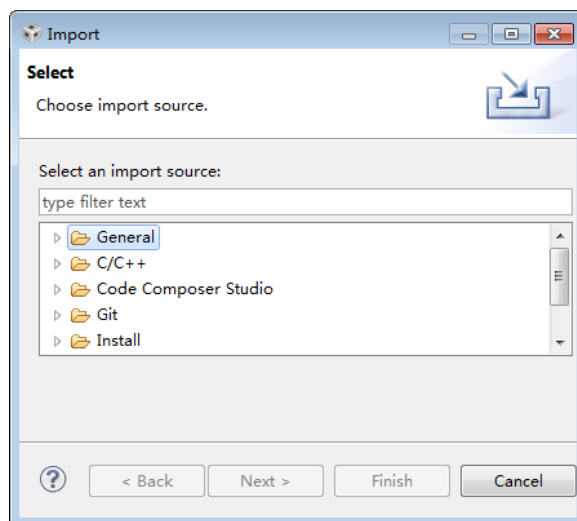
button.out: demo\dsp\ button\Debug\ button.out //DSP 镜像

main.c: demo\dsp\button\main.c //DSP 程序源码

5.1 导入工程

在工程导入前，请先使用 gel 文件唤醒 DSP 核，具体步骤参考《1. 工程建立前准备工作》。

然后点击 CCS 菜单 “File->Import”，弹出以下对话框：



点击 “Code Composer Studio->Existing CCS Eclipse Projects” 后再点击 “Next”，如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

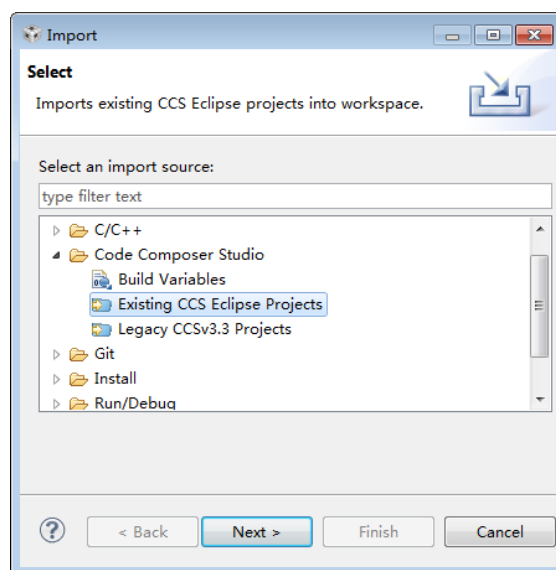
销售邮箱：sales@tronlong.com

公司总机：020-89986280

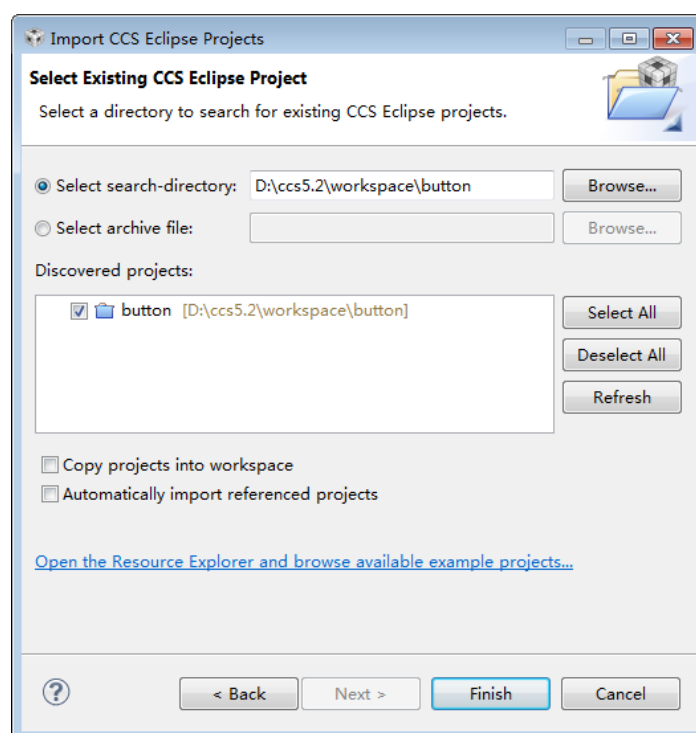
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

39 / 55



在弹出的对话框的“Select search-directory”后面点击“Browse”选择 button 的 CCS 工程的存放路径，选择后如下图所示：



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

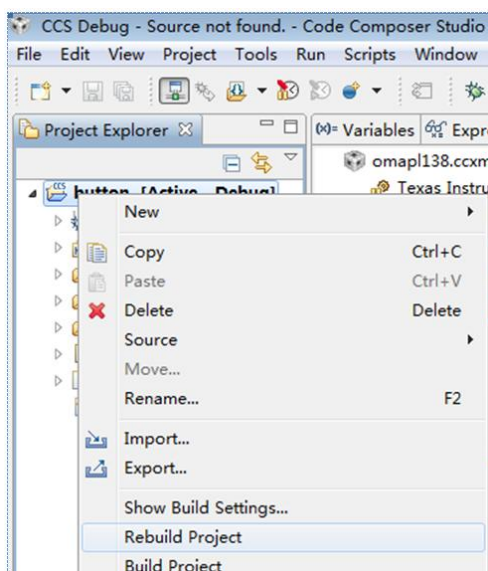
公司总机：020-89986280

公司网站：www.tronlong.com

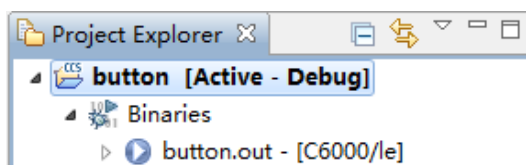
公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼


40 / 55

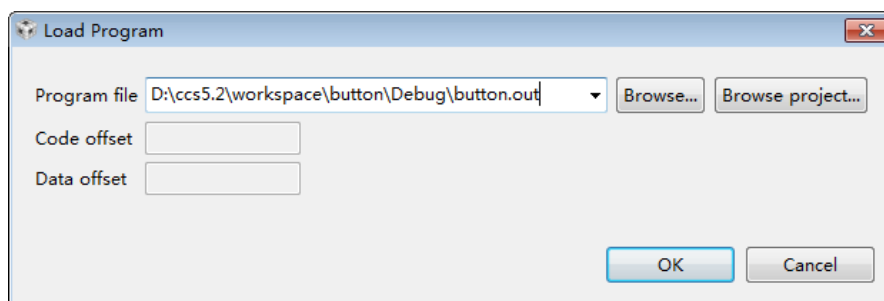
点击 Finish 就可以导入现有 CCS 工程。然后右击工程，在弹出来的对话框中点击“Rebuild Project”，即可对当前工程编译，如下图：



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



点击左上角的  load 键，选择要加载的 button.out 文件，如下图所示：



联系人：朱先生

联系电话：13318712959

QQ：2532609929


销售邮箱：sales@tronlong.com

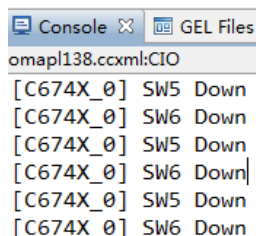
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

41 / 55

等待程序加载完成，然后点击 程序运行按钮，再按下 SW5 和 SW6 按键，可以发现 CCS 的 console 有按键被按下的信息提示，如下图所示：



```
omap138.ccxml:CIO
[C674X_0] SW5 Down
[C674X_0] SW6 Down
[C674X_0] SW5 Down
[C674X_0] SW6 Down
[C674X_0] SW5 Down
[C674X_0] SW6 Down
```

5.2 源代码解析

以下为按键中断函数的解析：

```
/* 头文件 */
#include <stdio.h>
#include "hw_types.h"           // HWREG (x)
#include "hw_syscfg0_OMAPL138.h" // SYSCFG0_PINMUX(x)
#include "soc_OMAPL138.h"      // SOC_GPIO_0_REGS
#include "gpio.h"              // GPIO_DIR_INTPUT
#include "interrupt.h"

/* 函数声明 */
void GPIOBankPinInit(void);      //GPIO初始化
void SetupInt(void);            //设置中断控制器
void ConfigureIntGPIO(void);    //配置按键中断
void GPIO0_6_Isr(void);         //SW5按键中断服务
void GPIO6_1_Isr(void);         //SW6按键中断服务

/* 主函数 */
void main(void) {

    GPIOBankPinInit();
    SetupInt();
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

42 / 55



```
ConfigureIntGPIO();
while(1);          //等待按键中断执行

}

/* GPIO初始化 */
void GPIOBankPinInit(void){

    unsigned int PINMUX_1_VAL;
    unsigned int PINMUX_19_VAL;

    PINMUX_1_VAL=HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(1));//读取PINMUX1寄存器的值
    PINMUX_19_VAL=HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(19));//读取PINMUX19寄存器的值

    PINMUX_1_VAL=(PINMUX_1_VAL&0xfffff0f)|0x00000080;//将底板按键SW5的对应管脚复用为GPIO，即GPIO0[6]
    PINMUX_19_VAL=(PINMUX_19_VAL&0xff0ffff)|0x00800000;//将底板按键SW6的对应管脚复用为GPIO，即GPIO6[1]

    /* 0x01C14124,PINMUX1[7-4]=0x8h */
    HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(1))=PINMUX_1_VAL;
    /* 0x01C1416C,PINMUX19[23-20]=0x8h */
    HWREG(SOC_SYSCFG_0_REGS + SYSCFG0_PINMUX(19))=PINMUX_19_VAL;
    /*配置底板2个用户按键的对应管脚为输入，以下为各组 GPIO第一个管脚 的对应值(范围1~144)
    * GPIO0[0] 1 , GPIO1[0] 17
    * GPIO2[0] 33 , GPIO3[0] 49
    * GPIO4[0] 65 , GPIO5[0] 81
    * GPIO6[0] 97 , GPIO7[0] 113
    * GPIO8[0] 129 */
    GPIODirModeSet(SOC_GPIO_0_REGS, 7, GPIO_DIR_INPUT);//设置GPIO0[6]为输入模式，即DIR01[6]=1,DIR01地址为0x01E26010
    GPIODirModeSet(SOC_GPIO_0_REGS, 98, GPIO_DIR_INPUT);//设置GPIO6[1]为输入模式，即DIR67[1]=1, DIR67地址为0x01E26088
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

43 / 55



GPIOIntTypeSet(SOC_GPIO_0_REGS, 7, GPIO_INT_TYPE_FALLEDGE); //设置GPIO0[6]为下降沿触发中断模式

GPIOIntTypeSet(SOC_GPIO_0_REGS, 98, GPIO_INT_TYPE_FALLEDGE); //设置GPIO6[1]为下降沿触发中断模式

GPIOBankIntEnable(SOC_GPIO_0_REGS, 0); //设置允许GPIO0[15: 0]产生中断，即BINTEN[0]=1，BINTEN地址为0x01E26008

GPIOBankIntEnable(SOC_GPIO_0_REGS, 6); //设置允许GPIO6[15: 0]产生中断，即BINTEN[6]=1，BINTEN地址为0x01E26008

}

/* 设置DSP中断控制器 */

void SetupInt(void){

// DSP中断控制器初始化

IntDSPINTCInit();

// DSP全局中断使能

IntGlobalEnable();

}

/* 配置按键中断 */

void ConfigureIntGPIO(void){

// 注册按键中断服务函数

IntRegister(C674X_MASK_INT4, GPIO0_6_isr); //注册SW5按键中断服务到INT4中断

IntRegister(C674X_MASK_INT5, GPIO6_1_isr); //注册SW6按键中断服务到INT5中断

// 映射按键中断对应CPU中断

IntEventMap(C674X_MASK_INT4, SYS_INT_GPIO_B0INT); //映射SW5按键对应INT4中断

IntEventMap(C674X_MASK_INT5, SYS_INT_GPIO_B6INT); //映射SW6按键对应INT5中断

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



```
// 使能按键对应的CPU中断
IntEnable(C674X_MASK_INT4);//使能INT4中断
IntEnable(C674X_MASK_INT5);//使能INT5中断

}

/* SW5按键中断服务 */
void GPIO0_6_Isr(void){

    printf("SW5 Down\n");

}

/* SW6按键中断服务 */
void GPIO6_1_Isr(void){

    printf("SW6 Down\n");

}
```

以上是按键中断程序注释，对于按键中断，有以下几点步骤：

- (1) 查看开发板的原理图，找到与用户按键连接的 GPIO 编号。

广州创龙电子的 OMAP-L138 开发板与用户按键连接的 GPIO 分别是 GPIO0[6]、GPIO6[1]，如下图所示：

联系人：朱先生

联系电话：13318712959

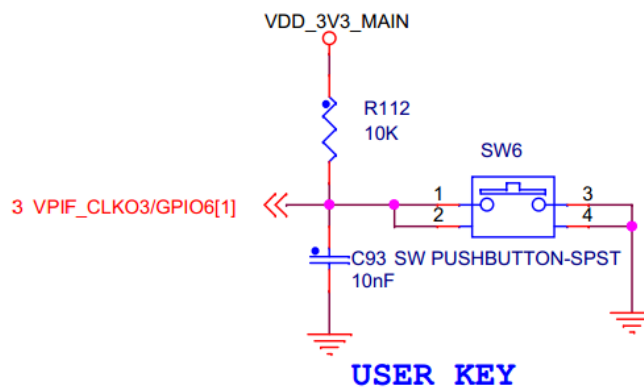
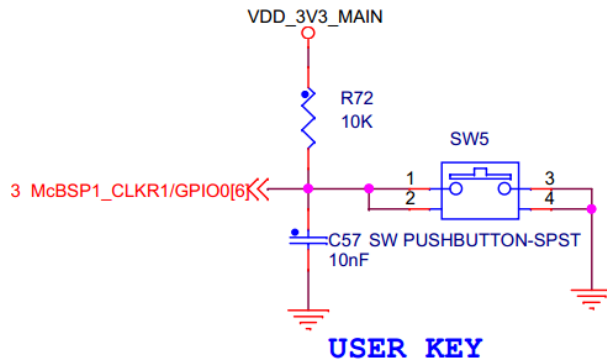
QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



- (2) 查看 OMAPL138 的数据手册，查找用户按键对应的管脚的 PINMUX 寄存器的地址，
将对应的管脚的寄存器中相应位设置为 GPIO 的工作模式。
- (3) 设置 GPIO 的方向为输入。
- (4) 设置 DSP 中断控制器。

设置 DSP 中断控制器主要作用是初始化 DSP 中断控制器和使能全局中断，分别使用以下函数：

初始化 DSP 中断控制器：IntDSPINTCInit()

使能全局中断 IntGlobalEnable();

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

46 / 55



(5) 配置按键中断。

配置按键中断函数中有以下几个作用：

- ✧ 注册按键中断服务函数，使用 IntRegister()函数；
- ✧ 映射按键中断对应 CPU 中断，使用 IntEventMap()；
- ✧ 使能按键对应的 CPU 中断，使用 IntEnable()；

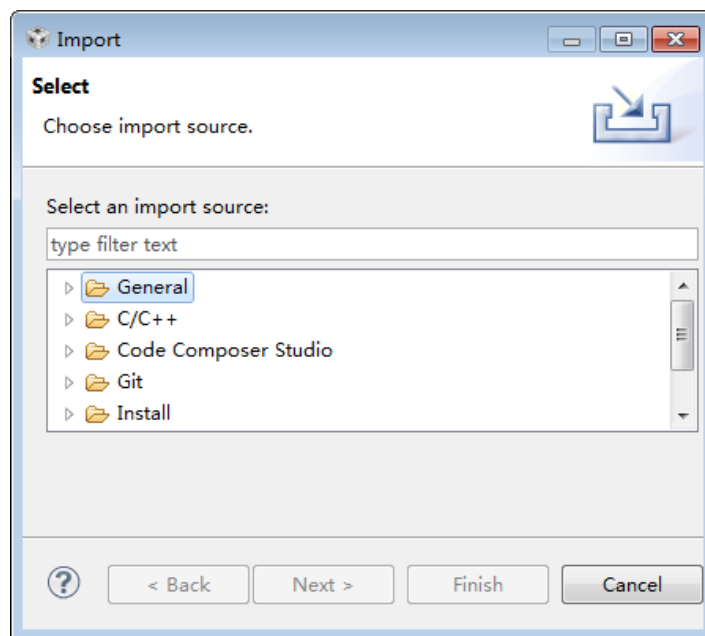
6. 基于 StarterWare 工程的 uart 串口通信程序演示

此 uart 程序完成以下功能：

- ✓ 往串口调试终端输出信息：StarterWare UART echo application；
- ✓ 等待接收串口调试终端发送过来的信息；
- ✓ 将接收到的串口调试终端信息发送到串口调试终端上显示出来；

6.1 导入 StarterWare 工程

StarterWare 里有很多可以直接使用的演示程序，可以大大的减少开发的时间。本小节将讲解如何基于 StarterWare 的 uart 工程演示串口通信功能。在演示前，请先使用 gel 文件唤醒 DSP 核，具体步骤参考《1. 工程建立前准备工作》。然后点击 CCS 菜单“File->Import”，弹出以下对话框：



联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

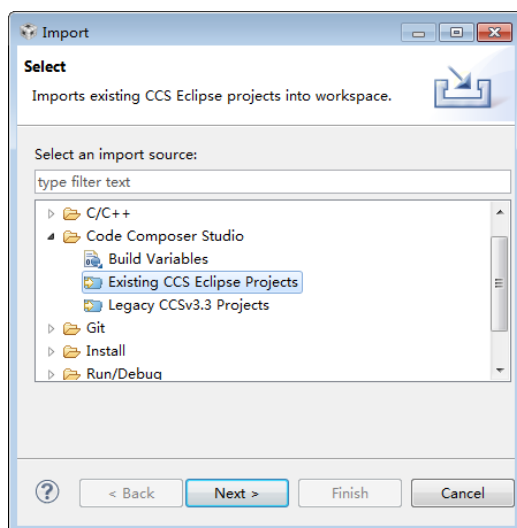
公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

48 / 55

点击“Code Composer Studio->Existing CCS Eclipse Projects”后再点击“Next”，如下图所示：



在弹出的对话框的“Select search-directory”后面点击“Browse”选择 StarterWare 的安装路径“C:\ti\OMAPL138_StarterWare_1_10_04_01”，然后点击“Select All”，就选择了 StarterWare 中所有的工程。工程文件来自 StartWare 的 build 目录。选择后如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

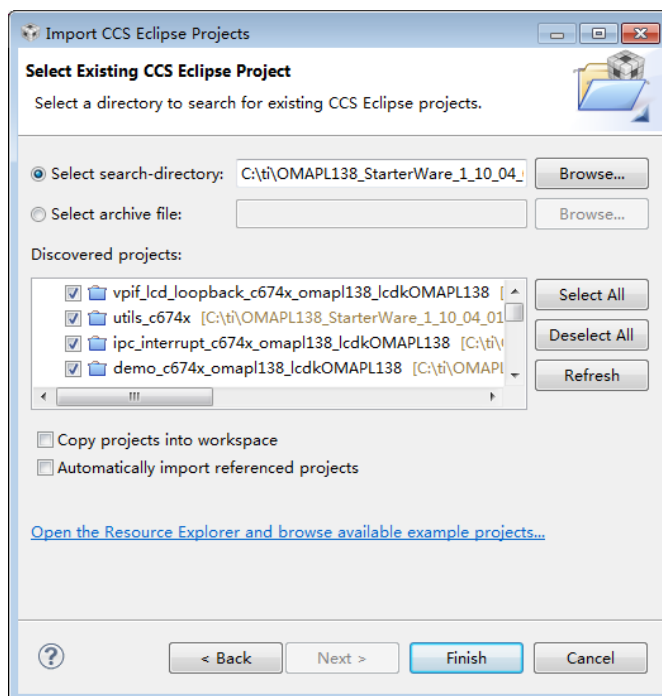
销售邮箱：sales@tronlong.com

公司总机：020-89986280

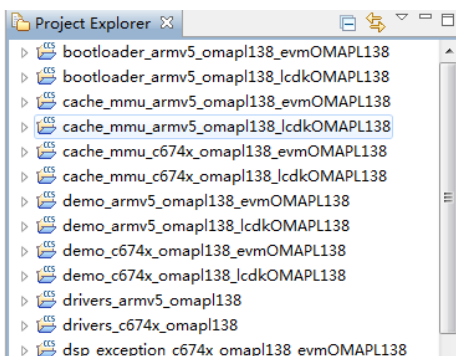
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

49 / 55



点击 Finish 就可以导入 StartWare 现有的全部 CCS 工程。导入后如下图所示：



6.2 修改 CPU 频率参数

找到“uart_c674x_omap138_evmOMAPL138”工程，打开 uartEcho.c 文件。找到第 84 行，如下图所示。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

50 / 55



```
0x712148 uartEcho.c
67 unsigned int intFlags = 0;
68 unsigned int config = 0;
69
70 /* Enabling the PSC for UART2.*/
71 PSCModuleControl(SOC_PSC_1_REGS, HW_PSC_UART2, PSC_POWERDOMAIN_ALWAYS_ON,
72 PSC_MDCTL_NEXT_ENABLE);
73
74 /* Setup PINMUX */
75 UARTPinMuxSetup(2, FALSE);
76
77 /* Enabling the transmitter and receiver*/
78 UARTEnable(SOC_UART_2_REGS);
79
80 /* 1 stopbit, 8-bit character, no parity */
81 config = UART_WORDL_8BITS;
82
83 /* Configuring the UART parameters*/
84 UARTConfigSetExpClk(SOC_UART_2_REGS, SOC_UART_2_MODULE_FREQ,
85 BAUD_115200, config,
86 UART_OVER_SAMP_RATE_16);
87
```

按住 Ctrl，然后点击串口频率变量 SOC_UART_2_MODULE_FREQ，CCS 会自动打开此变量的定义位置。可以看出，串口频率变量 SOC_UART_2_MODULE_FREQ 的定义位于 StarterWare 的 include/socOMAPL138.h 的第 592 行，如下图所示：

```
0x712148 uartEcho.c soc_OMAPL138.h
566
567 #define SOC_SYSCLK_1_FREQ (300000000)
568 #define SOC_SYSCLK_2_FREQ (SOC_SYSCLK_1_FREQ/2)
569 #define SOC_SYSCLK_3_FREQ (SOC_SYSCLK_1_FREQ/3)
570 #define SOC_SYSCLK_4_FREQ (SOC_SYSCLK_1_FREQ/4)
571
572 #define SOC_ASYNC_2_FREQ (24000000)
573
574 /** I2C */
575 #define SOC_I2C_0_MODULE_FREQ (SOC_ASYNC_2_FREQ)
576 #define SOC_I2C_1_MODULE_FREQ (SOC_SYSCLK_4_FREQ)
577
578 /** MCBSP */
579 #define SOC_MCBSP_0_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
580 #define SOC_MCBSP_1_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
581
582 /** LCDC */
583 #define SOC_LCDC_0_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
584
585 /** SPI */
586 #define SOC_SPI_0_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
587 #define SOC_SPI_1_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
588
589 /** UART */
590 #define SOC_UART_0_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
591 #define SOC_UART_1_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
592 #define SOC_UART_2_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
593
594 /** EHRPWM */
595 #define SOC_EHRPWM_0_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
596 #define SOC_EHRPWM_1_MODULE_FREQ (SOC_SYSCLK_2_FREQ)
```

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



浏览第 567、568、592 行可以发现如下规律：

$SOC_UART_2_MODULE_FREQ = SOC_SYSCLK_1_FREQ / 2$

$SOC_SYSCLK_1_FREQ$ 是指 CPU 的频率，这个参数必须和 GEL 文件参数对应。从提供的 GEL 文件可以看出已经设置 CPU 运行频率为 456MHz 了，GEL 文件中的定义如下图所示：

```
338 hotmenu TronLong_OMAP_PLL0 ()
339 {
340     GEL_TextOut("\tEnabling TronLong_OMAP_PLL0...\n", "Output", 1, 1, 1);
341     device PLL0(0, 18, 0, 0, 1, 4, 9); //456M
```

因此需要将 $SOC_SYSCLK_1_FREQ$ 改为 456000000，修改完成后如下图所示：

```
566
567 #define SOC_SYSCLK_1_FREQ (456000000)
568 #define SOC_SYSCLK_2_FREQ (SOC_SYSCLK_1_FREQ/2)
569 #define SOC_SYSCLK_3_FREQ (SOC_SYSCLK_1_FREQ/3)
570 #define SOC_SYSCLK_4_FREQ (SOC_SYSCLK_1_FREQ/4)
```

6.3 重新编译和运行程序

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，如下图所示：

联系人：朱先生

联系电话：13318712959

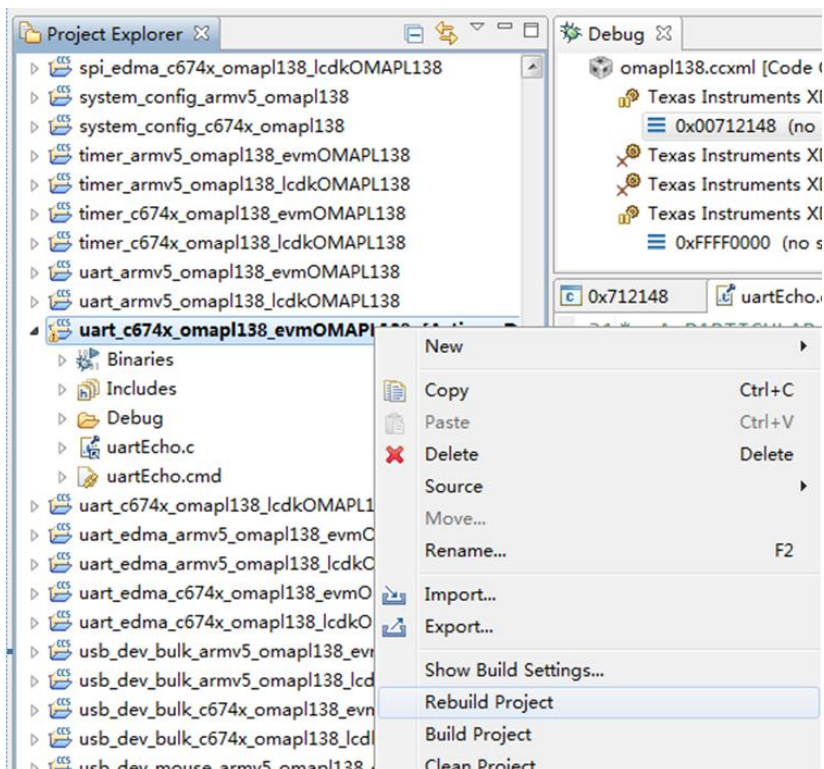
QQ：2532609929

销售邮箱：sales@tronlong.com

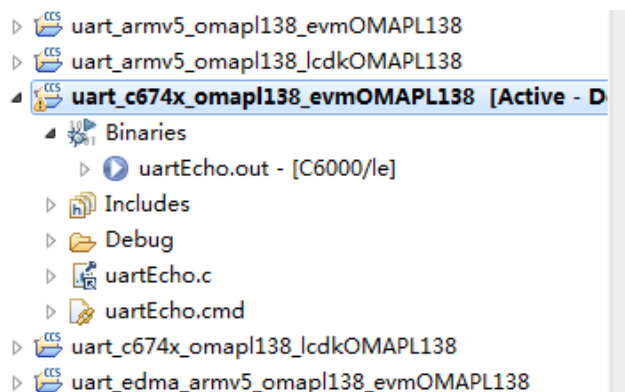
公司总机：020-89986280


公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



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



点击左上角的  load 键，选择要加载的 uartEcho.out 文件，如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

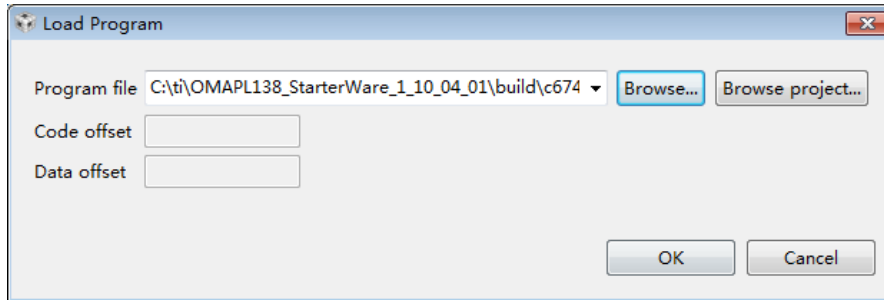
销售邮箱：sales@tronlong.com


公司总机：020-89986280

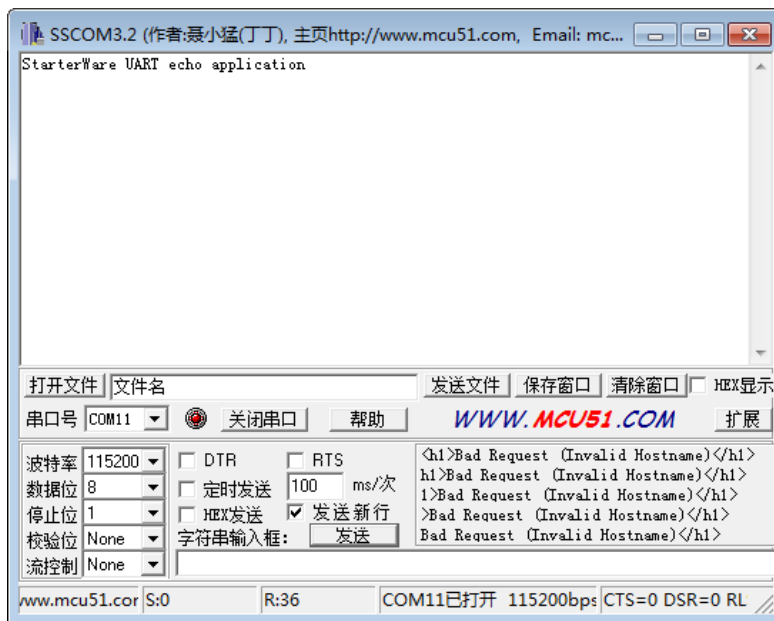
公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼

53 / 55



等待程序加载完成，关闭 ZOC，打开串口调试助手（光盘 tools 目录下有此软件），设置正确 COM 口，波特率为 115200，8N1，无检验位。然后点击  程序运行按钮，可以发现串口调试助手正确输出调试信息了，如下图所示：



在输入行输入任意信息，例如“www.tronlong.com”，然后点击发送，就可以看到发送到 OMAPL138 DSP 端的信息被反发送到串口调试助手上面显示出来了，如下图所示：

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼



可以看出使用 StarterWare 可以快速出一个应用程序，大大的节省了开发的时间。StarterWare 中还有其他的 demo 程序可以直接使用，例如 i2c、spi、usb 等外设程序，都对客户的开发十分有帮助。

联系人：朱先生

联系电话：13318712959

QQ：2532609929

销售邮箱：sales@tronlong.com

公司总机：020-89986280

公司网站：www.tronlong.com

公司总部：广州市天河区五山路华南农业大学真维斯活动中心2楼