OMAPL138 PRU 开发例程

Revision History

Draft Date	Revision No.	Description
2016/05/25	V1.1	1.模板更新。
2015/04/28	V1.0	1.初始版本。

目　　录

1 环境搭建

1.1 安装 pru-sdk 包

PRU 的 SDK 源码 ti-pru-sw-1.00.00-r0+svnr33.tar.bz2 位于光盘资料 tools 目录下，将此文件解压到虚拟机任意路径下，解压后如下图所示：

```
tl@tl-desktop:~/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33$ ls
app_loader     kernel_loader  Makefile        readme.txt
example_apps   LICENCE.txt    peripheral_lib  utils
```

图 1

1.2 编译 pru-sdk 包

执行以下命令进行编译：

Host# make CROSS_COMPILE=arm-none-linux-gnueabi- LINUXKERNEL_INSTALL_DIR=/home/tl/omapl138/linux-3.3

编译成功如下图所示：

```
PRU Assembler Version 0.76
Copyright (C) 2005-2010 by Texas Instruments Inc.


Pass 2 : 0 Error(s), 0 Warning(s)

Writing Code Image of 27 word(s)



PRU Assembler Version 0.76
Copyright (C) 2005-2010 by Texas Instruments Inc.


Pass 2 : 0 Error(s), 0 Warning(s)

Writing Code Image of 47 word(s)

make[1]: Leaving directory `/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33/exa
mple_apps'
tl@tl-desktop:~/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33$
```

图 2

参数解析：

CROSS_COMPILE：交叉编译工具链；

LINUXKERNEL_INSTALL_DIR： 内核安装路径；

请务必使用创龙提供的 linux-3.3 内核源码进行编译，编译前确保内核源码已正确编译过。

1.3 安装 pru-sdk 包到文件系统中

执行以下命令进行编译：

```
Host#    sudo make  DESTDIR=/home/tl/omapl138/rootfs install
```

安装完如下图所示：

```
tl@tl-desktop:~/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33$ sudo make  DESTDIR=/hom
e/tl/omapl138/rootfs install
cp -f example_apps/bin/* /home/tl/omapl138/rootfs/usr/share/ti/ti-pru-eg/
cp -f peripheral_lib/edma_driver/module/edmautils.ko /home/tl/omapl138/rootfs/li
b/modules/2.6.37/kernel/drivers/pru/
```

图 3

参数解析：

DESTDIR： 文件系统位置；

2 LED 开发例程

本例程主要通过 PRU 控制开发板 LED 灯的闪烁。

2.1 源码安装

源码路径位于光盘资料"demo/app/tl-pru-example/tl-pru-leds.tar.bz2"，将源码解压到虚拟机任意路径下，如下图所示：

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-leds$ ls
Makefile  Rules.make  tl-pru-leds.c  tl-pru-leds.p
```

图 4

2.2 修改配置文件

修改 Rules.make 配置文件：

```
CCTOOLS?=arm-none-linux-gnueabi-gcc      //交叉编译工具链
```

where the ti_pru_sw path

PRU_SW_DIR=/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33 //PRU_SDK 安装目录

LIBDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/lib

INCDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/include

UTILS_DIR=$(PRU_SW_DIR)/utils

如下图所示：

```
CCTOOLS?=arm-none-linux-gnueabi-gcc

# where the ti_pru_sw path
PRU_SW_DIR=/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33

LIBDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/lib
INCDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/include
UTILS_DIR=$(PRU_SW_DIR)/utils
```

图 5

2.3 编译

执行以下命令进行编译：

Host# make

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-leds$ make
arm-none-linux-gnueabi-gcc -Wall -I/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33
/app_loader/include -D__DEBUG -O2 -mtune=arm926ej-s -march=armv5te -c -o obj/tl-pru
-leds.o tl-pru-leds.c
/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33/utils/pasm -b tl-pru-leds.p


PRU Assembler Version 0.76
Copyright (C) 2005-2010 by Texas Instruments Inc.


Pass 2 : 0 Error(s), 0 Warning(s)

Writing Code Image of 42 word(s)

mv tl-pru-leds.bin bin
arm-none-linux-gnueabi-gcc -Wall -I/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33
/app_loader/include -D__DEBUG -O2 -mtune=arm926ej-s -march=armv5te -o bin/tl-pru-le
ds obj/tl-pru-leds.o -L/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33/app_loader/
lib -lprussdrv -lpthread
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-leds$ ls
bin  Makefile  obj  Rules.make  tl-pru-leds.c  tl-pru-leds.p
```

图 6

编译生成的可执行文件位于 bin 目录下，如下图所示：

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-leds/bin$ ls
tl-pru-leds  tl-pru-leds.bin
```

图 7

2.4 运行程序

将 bin 目录中的 tl-pru-leds 和 tl-pru-leds.bin 目标文件拷贝到开发板上，并放于同一目录下。在此目录下执行以下命令：

Target# ./tl-pru-leds

如下图所示：

```
root@am180x-evm:~/tl-pru-example/tl-pru-leds/bin# ./tl-pru-leds

INFO: Starting tl-pru-leds example.
File ./tl-pru-leds.bin open passed
```

图 8

演示现象：

开发板 LED 会同时闪烁十次。

3 按键开发例程

本例程演示 PRU 通过按键对 LED 灯进行控制。

开发板 LED 编号和 GPIO 对应关系如下：

表 1

开发板型号	GPIO0[0]	GPIO0[5]	GPIO0[1]	GPIO0[2]
TL138/1808-EVM	D7	D6	D9	D10
TL138/1808-EasyEVM	D7	D6	D9	D10
TL138/1808-EthEVM	D7	D6	D9	D10
TL138/1808F-EasyEVM	\	GD1	GD2	GD3
TL138/1808F-EVM	\	D1	D2	D3

3.1 源码安装

源码路径位于光盘资料"demo/app/tl-pru-example/tl-pru-keys.tar.bz2"，将源码解压到虚拟机任意路径下，如下图所示：

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-keys$ ls
Makefile  Rules.make  tl-pru-keys.c  tl-pru-keys.p
```

图 9

3.2 修改配置文件

修改 Rules.make 配置文件：

CCTOOLS?=arm-none-linux-gnueabi-gcc　　//交叉编译工具链

where the ti_pru_sw path

PRU_SW_DIR=/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33　　//PRU_SDK 安装目录

LIBDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/lib

INCDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/include

UTILS_DIR=$(PRU_SW_DIR)/utils

如下图所示：

```
CCTOOLS?=arm-none-linux-gnueabi-gcc

# where the ti_pru_sw path
PRU_SW_DIR=/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33

LIBDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/lib
INCDIR_APP_LOADER?=$(PRU_SW_DIR)/app_loader/include
UTILS_DIR=$(PRU_SW_DIR)/utils
```

图 10

3.3 编译

执行以下命令进行编译：

Host#　make

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-keys$ make
arm-none-linux-gnueabi-gcc -Wall -I/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svn
r33/app_loader/include -D__DEBUG -O2 -mtune=arm926ej-s -march=armv5te -c -o obj/
tl-pru-keys.o tl-pru-keys.c
/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33/utils/pasm -b tl-pru-keys.p


PRU Assembler Version 0.76
Copyright (C) 2005-2010 by Texas Instruments Inc.


Pass 2 : 0 Error(s), 0 Warning(s)

Writing Code Image of 111 word(s)

mv tl-pru-keys.bin bin
arm-none-linux-gnueabi-gcc -Wall -I/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svn
r33/app_loader/include -D__DEBUG -O2 -mtune=arm926ej-s -march=armv5te -o bin/tl-
pru-keys obj/tl-pru-keys.o -L/home/tl/pru/pru-sdk/ti-pru-sw-1.00.00-r0+svnr33/ap
p_loader/lib -lprussdrv -lpthread
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-keys$ ls
bin  Makefile  obj  Rules.make  tl-pru-keys.c  tl-pru-keys.p
```

图 11

编译生成的可执行文件位于当前 bin 目录下，如下图所示：

```
tl@tl-desktop:~/pru/tl-pru-example/tl-pru-keys/bin$ ls
tl-pru-keys  tl-pru-keys.bin
```

图 12

3.4 运行程序

将 bin 目录中的 tl-pru-keys 和 tl-pru-keys.bin 目标文件拷贝到开发板上，并放于同一目录下。在此目录下执行以下命令：

Target# ./tl-pru-keys

如下图所示：

```
root@am180x-evm:~/tl-pru-example/tl-pru-keys/bin# ./tl-pru-keys

INFO: Starting tl-pru-keys example.
File ./tl-pru-keys.bin open passed
```

图 13

演示现象：

以 TL138-EVM 为例，开发板 LED D6 被点亮，按下按键 SW5 时 LED D9 会被点亮，按下按键 SW6 时，LED D10 会被点亮，大约历时 10s，测试程序执行完毕，LED D6 熄灭，程序退出。

4 程序代码分析

以 LED 例程为例：

- **Linux 端代码分析**

程序设计流程如下：

- 初始化 pru 模块
- 打开 pru 中断
- 下载程序到 pru 中
- 等待程序执行完毕，pru 会发送一个中断给 arm
- 关闭 pru 并退出

Linux 端代码主要位于 tl-pru-leds.c 文件内，代码如下图所示：

```
int main (void)
{
    unsigned int ret;
    tpruss_intc_initdata pruss_intc_initdata = PRUSS_INTC_INITDATA;

    printf("\nINFO: Starting %s example.\r\n",  "tl-pru-leds");
    /* Initialize the PRU */
    prussdrv_init ();

    /* Open PRU Interrupt */
    ret = prussdrv_open(PRU_EVTOUT_0);
    if (ret)
    {
        printf("prussdrv_open open failed\n");
        return (ret);
    }

    /* Get the interrupt initialized */
    prussdrv_pruintc_init(&pruss_intc_initdata);
        /* Execute example on PRU */
    prussdrv_exec_program (PRU_NUM, "./tl-pru-leds.bin");
    /* Wait until PRU0 has finished execution */
        prussdrv_pru_wait_event (PRU_EVTOUT_0);
        prussdrv_pru_clear_event (PRU0_ARM_INTERRUPT);

        /* Disable PRU and close memory mapping*/
        prussdrv_pru_disable (PRU_NUM);
        prussdrv_exit ();


    return(0);
}
```

图 14

- **PRU 端代码分析**

  PRU 端代码主要位于 tl-pru-leds.p 文件内，主要以汇编为主。主函数程序工作流程如下：

  - Pinmux 配置
  - Gpio Direct 配置
  - Gpio Data 输出 1，LED ON
  - 延时
  - Gpio Data 输出 0，LED OFF

```
/*程序入口地址，汇编程序需要注明入口地址*/
.origin 0                    //表示 PRU 从程序的 0 开始的位置开始取指
.entrypoint START            //告诉编译器，程序入口地址是从 label：START 开始
/*宏定义*/
#define  LOOP_TIMES              10       // define the LED flash times

// Refer to this mapping in the file - \prussdrv\include\pruss_intc_mapping.h
#define PRU0_ARM_INTERRUPT    34

#define  GPIO_BASE            0x01E26000 // the base address of GPIO registers
#define GPIO01_DIRECT         0x10      // the offset address of DIR01 register
#define  GPIO0_SETDATAOUT 0x18      // the offset address of SET_DATA01    registers
#define  GPIO0_CLEARDATAOUT   0x1C      // the offset address of CLR_DATA01 registers

#define  SYSCFG_BASE              0x01c14000
#define  PINMUX1_OFFSET           0x124

// PINMUX_GPIO0_0    PINMUX_GPIO0_1    PINMUX_GPIO0_2    PINMUX_GPIO0_5
#define PINMUX1_GPIO_LED_MASS    (0xf << 28 | 0xf << 24 | 0xf << 20 | 0xf << 8)
#define PINMUX1_GPIO_LED         (0x8 << 28 | 0x8 << 24 | 0x8 << 20 | 0x8 << 8)
```

```
/*START 标号开始了 PRU 程序，类似 C 程序从 main 函数开始*/
START:

    // setup pinmux gpio led
    MOV         r3, SYSCFG_BASE + PINMUX1_OFFSET
    LBBO    r2, r3, 0, 4

    MOV         r4, PINMUX1_GPIO_LED_MASS
    NOT         r4, r4
    AND         r2, r2, r4

    MOV         r4, PINMUX1_GPIO_LED
    OR      r2, r2, r4
    SBBO    r2, r3, 0, 4

/*函数的定义*/
// Delay 200ms funcation
DELAY_400MS:
    MOV         r0, 45600000            // delay: 400ms
    LOOP_DELAY:                         // PRU CORE CLK: 228MHz, cycle:8.77ns
    SUB     r0, r0, 1               // 1 CPI
    QBNE    LOOP_DELAY, r0, 0       // 1 CPI
    RET

/*函数的调用*/
CALL    DELAY_400MS         // Call to delay some time
    /*发送中断信号给 ARM*/
    // Send notification to Host for program completion
    MOV         r31.b0, #PRU0_ARM_INTERRUPT
    /*关闭 PRU*/
```

```
// Halt the processor
HALT
```

更多帮助

销售邮箱：sales@tronlong.com

技术邮箱：support@tronlong.com

创龙总机：020-8998-6280

技术热线：020-3893-9734

创龙官网：www.tronlong.com

技术论坛：www.51ele.net

线上商城：https://tronlong.taobao.com

TMS320C6748、OMAPL138 交流群：227961486、324023586

TI 中文论坛：http://www.deyisupport.com/

TI 英文论坛：http://e2e.ti.com/

TI 官网：www.ti.com

TI WIKI：http://processors.wiki.ti.com/

附录 1

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 15