**实验二：流水灯程序**

1. 实验目的
2. 熟悉linux系统，学会简单linux指令
3. 熟悉OK6410-A开发板的烧入步骤
4. 熟悉ARM寄存器，地址等。
5. 系统性的了解UBOOT和linux内核，yaffs2系统映像等知识
6. 实验环境

开发机环境

操作系统：ubuntu 12.04

交叉编译环境：arm-linux-gcc 4.3.2

6410板子内核源码：linux-3.0.1

目标板环境：OK6410-A linux-3.0.1

1. 实验原理

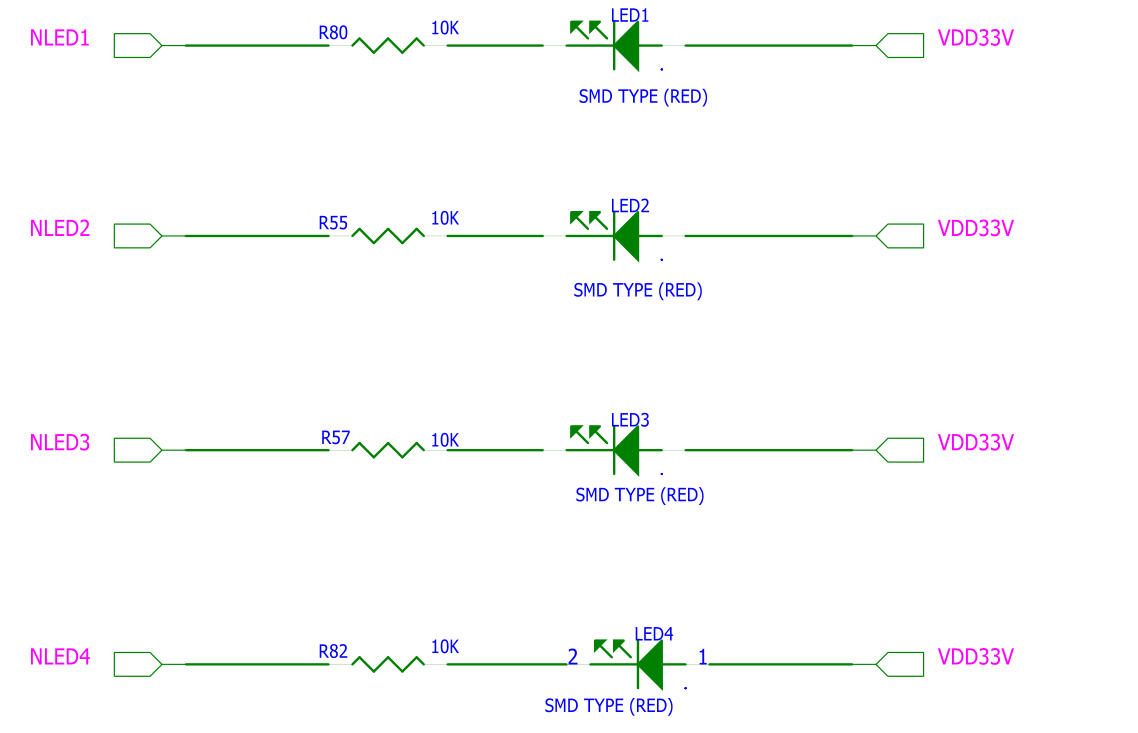


图1-OK6410LED原理图

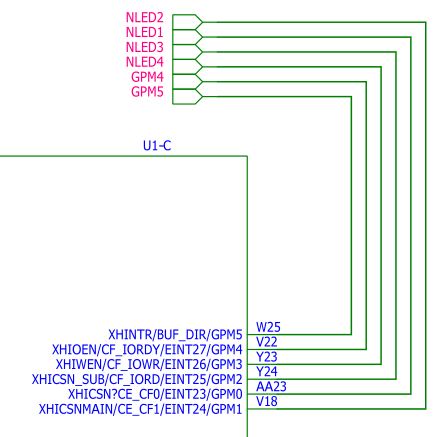


图2-LED原理图

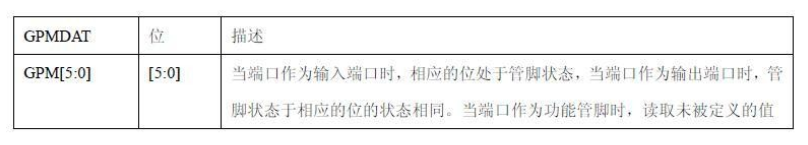
从上面的原理图可以得知，LED与CPU引脚的连接方法如下，低电平点亮。

LED1 -GPM0

LED2 -GPM1

LED3 -GPM2

LED4 -GPM3 



通过上面可以得知，需要先将GPM0设置为输出方式。将相应的寄存器进行配置。然后将GPMDAT寄存器的第0位置0灯亮，置1灯灭。

1. 实验代码
2. 编写驱动程序：led.c

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/fs.h>

#include <linux/init.h>

#include <linux/miscdevice.h>

#include <linux/delay.h>

#include <asm/uaccess.h>

#include <linux/device.h>

#include <linux/cdev.h>

#include <asm/irq.h>

#include <mach/gpio.h>

#include <mach/gpio-bank-m.h>

#include <mach/regs-gpio.h>

#include <mach/map.h>

//#include <plat/regs-gpio.h>

#include <plat/gpio-cfg.h>

#include <mach/hardware.h>

#include <linux/io.h>

MODULE\_LICENSE**(**"GPL"**);**

#define LED\_MAJOR 240

int led\_open**(**struct inode **\***inode**,** struct file **\***filp**)**

**{**

unsigned int tmp**;**

tmp **=** readl**(**S3C64XX\_GPMCON**);**

tmp **=** **(**tmp **&** **~(**0xffff**)** **|** **(**0x1111**));** //set the GPIO output mode

writel**(**tmp**,** S3C64XX\_GPMCON**);**

printk**(**KERN\_INFO "$$$$$$$$$$$led\_open$$$$$$$$$\n"**);**

**return** 0**;**

**}**

ssize\_t led\_read**(**struct file **\***filp**,** char \_\_user **\***buf**,** size\_t count**,** loff\_t **\***f\_pos**)**

**{**

printk**(**KERN\_INFO "$$$$$$$$$$led\_read$$$$$$$$$\n"**);**

**return** count**;**

**}**

ssize\_t led\_write**(**struct file **\***filp**,** char \_\_user **\***buf**,** size\_t count**,** loff\_t **\***f\_pos**)**

**{**

char mbuf**[**10**];**

unsigned int tmp**;**

copy\_from\_user**(**mbuf**,** buf**,** count**);**

**switch** **(**mbuf**[**0**])**

**{**

**case** 0**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **|=** **(**0x1**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 1**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **&=** **~(**0x1**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 2**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **|=** **(**0x2**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 3**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **&=** **~(**0x2**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 4**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **|=** **(**0x4**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 5**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **&=** **~(**0x4**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 6**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **|=** **(**0x8**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**case** 7**:**

tmp **=** readl**(**S3C64XX\_GPMDAT**);**

tmp **&=** **~(**0x8**);**

writel**(**tmp**,** S3C64XX\_GPMDAT**);**

**break;**

**default:**

**break;**

**}**

printk**(**KERN\_INFO "$$$$$$$$$$led\_write$$$$$$$$$\n"**);**

**return** count**;**

**}**

int led\_release**(**struct inode **\***inode**,** struct file **\***filp**)**

**{**

printk**(**KERN\_INFO "$$$$$$$$$$led\_release$$$$$$$$$\n"**);**

**return** 0**;**

**}**

struct file\_operations my\_fops **=** **{**

**.**owner **=** THIS\_MODULE**,**

**.**open **=** led\_open**,**

**.**read **=** led\_read**,**

**.**write **=** led\_write**,**

**.**release **=** led\_release**,**

**};**

static int led\_init**(**void**)**

**{**

int rc**;**

printk**(**KERN\_INFO "Test led dev\n"**);**

rc **=** register\_chrdev**(**LED\_MAJOR**,** "led"**,** **&**my\_fops**);**

**if** **(**rc **<** 0**)**

**{**

printk**(**KERN\_INFO "register %s mychar dev error\n"**,** "led"**);**

**return** **-**1**;**

**}**

printk**(**KERN\_INFO "$$$$$$$$$ register led dev OK\n"**);**

**return** 0**;**

**}**

static void led\_exit**(**void**)**

**{**

unregister\_chrdev**(**LED\_MAJOR**,** "led"**);**

printk**(**KERN\_INFO "Good Bye!\n"**);**

**}**

module\_init**(**led\_init**);**

module\_exit**(**led\_exit**);**

1. 编写Makefile文件

obj**-**m **:=** led**.**o

KERNELDIR **=** /work/linux-3.0.1

all**:**

make **-**C $**(**KERNELDIR**)** M**=**`pwd` modules

clean**:**

make **-**C $**(**KERNELDIR**)** M**=**`pwd` clean

1. 编写测试文件：test.c

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

int main**(**void**)**

**{**

int fd**;**

char buf**[**10**]** **=** **{**0**,** 1**,** 2**,** 3**,** 4**,** 5**,** 6**,** 7**};**

fd **=** open**(**"/dev/my\_led"**,** O\_RDWR**);**

**if** **(**fd **<** 0**)**

**{**

printf**(**"Open /dev/my\_led file error\n"**);**

**return** **-**1**;**

**}**

**while** **(**1**)**

**{**

write**(**fd**,** **&**buf**[**1**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**0**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**3**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**2**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**5**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**4**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**7**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**6**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**5**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**4**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**3**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**2**],** 1**);**

usleep**(**100**);**

write**(**fd**,** **&**buf**[**1**],** 1**);**

sleep**(**1**);**

write**(**fd**,** **&**buf**[**0**],** 1**);**

usleep**(**100**);**

**}**

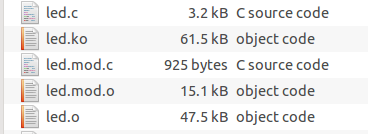
close**(**fd**);**

**return** 0**;**

**}**

1. 实验步骤
2. 编译驱动程序和测试程序

在终端中运行：#make命令，编译成功生生下列文件



在终端中运行：#arm-linux-gcc test.c -o test，编译成功生成文件



1. 将文件拷贝到SD卡
2. 将SD卡插入到OK6410开发板中
3. 在OK6410终端中运行程序

加载驱动：#insmod sdcard/led.ko

创建设备文件：# mknod /dev/my\_led c 240 0

运行测试文件：#./sdcard/test

卸载驱动程序;#rmmod sdcard/led.ko

1. 运行结果：此时可以看到OK6410开发板的LED灯逐个点亮后熄灭，实现流水灯功能。
2. linux批处理代码：文件命名为t

#!/bin/sh

insmod **/**sdcard**/**led**.**ko

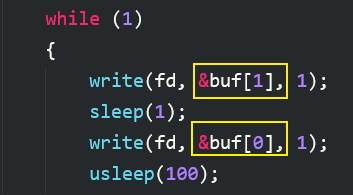
mknod **/**dev**/**my\_led c 240 0

**./**sdcard**/**test

直接在OK6410中运行:#./sdcard/t

可以观察到同样的测试结果

1. 更改测试文件中的代码可实现不同的功能



1. 实验总结

本次实验实在第一次点亮LED的基础上对LED灯进行更为复杂的操作,加深了对代码的理解，在此基础上实现二次开发。