# Locate Sample Code

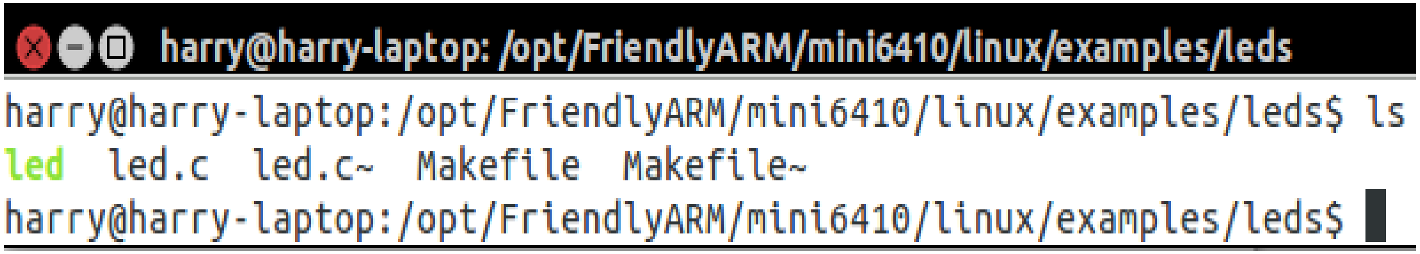
## User Space Program

1. locate the folder holds the examples of user application program, per default location, it is at

/opt/FriendlyARM/mini6410/linux/examples$

2. identify the leds folder to find the user application program shown as follows

/opt/FriendlyARM/mini6410/linux/examples/leds$



harry@harry-laptop: /opt/FriendlyARM/mini6410/linux/examples/leds

```
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds$ ls
led  led.c  led.c~  Makefile  Makefile~
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds$ 
```

# User Space Program

Note: you can compile the code by simply doing make.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int main(int argc, char **argv)
{
        int on;
        int led_no;
        int fd;
        printf("hello\n"); //Feb. 16, 2015
        if (argc != 3 || sscanf(argv[1], "%d", &led_no) != 1
                    || sscanf(argv[2],"%d", &on) != 1
                    || on < 0
                    || on > 1
                    || led_no < 0
                    || led_no > 3) {
                fprintf(stderr, "Usage: leds led_no 0|1\n");
                exit(1);
        }
```

```
        fd = open("/dev/leds0", 0);
        if (fd < 0) {
                fd = open("/dev/leds", 0);
        }
        if (fd < 0) {
                perror("open device leds");
                exit(1);
        }

        ioctl(fd, on, led_no)
        close(fd);

        return 0;
   }
```

Table 1. access to the device driver

| |
|---|
| fd = open("/dev/leds0", 0); |
| ioctl(fd, on, led_no); |

# Locate Device Driver Program

1. go to ~/linux-2.6.38/drivers/char$ ls mini* to find all prebuilt 6410 device drivers, shown below:

```
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers/char$ ls mini*
mini6410_adc.c      mini6410_hello_module.c      mini6410_leds.o      mini6410_pwmHarry.c
mini6410_adc.mod.c  mini6410_hello_module.mod.c  mini6410_pwm2.c      mini6410_pwm.mod.c
mini6410_adc.o      mini6410_hello_module.o      mini6410_pwm2.mod.c  mini6410_pwm.o
mini6410_buttons.c  mini6410_leds.c              mini6410_pwm2.o
mini6410_buttons.o  mini6410_leds.mod.c          mini6410_pwm.c
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers/char$
```

Note the above driver with 'harry" in their file names are those drivers of my modification.

# GPIO Device Driver

## mini6410_leds.c

```
#include <linux/miscdevice.h>
#include <linux/delay.h>
#include <asm/irq.h>
//#include <mach/regs-gpio.h>
#include <mach/hardware.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/init.h>
#include <linux/mm.h>
#include <linux/fs.h>
#include <linux/types.h>
#include <linux/delay.h>
#include <linux/moduleparam.h>
#include <linux/slab.h>
#include <linux/errno.h>
#include <linux/ioctl.h>
#include <linux/cdev.h>
#include <linux/string.h>
#include <linux/list.h>
#include <linux/pci.h>
#include <asm/uaccess.h>
```

```
#include <asm/atomic.h>
#include <asm/unistd.h>
#include <mach/map.h>
#include <mach/regs-clock.h>
#include <mach/regs-gpio.h>
#include <plat/gpio-cfg.h>
#include <mach/gpio-bank-e.h>
#include <mach/gpio-bank-k.h>

#define DEVICE_NAME "leds0"
```

Table 1. There 3 modules for this device driver

| |
|---|
| static long sbc2440_leds_ioctl<br>(struct file *filp, unsigned int cmd, unsigned long arg) |
| static int __init dev_init(void) |
| static void __exit dev_exit(void) |

# sbc2440_leds_ioctl

static long sbc2440_leds_ioctl(struct file *filp, unsigned int cmd, unsigned long arg)

```
static long sbc2440_leds_ioctl(struct file *filp, unsigned int cmd, unsigned long arg)
{
        switch(cmd) {
                unsigned tmp;
        case 0:
        case 1:
                if (arg > 4) {
                        return -EINVAL;
                }
                tmp = readl(S3C64XX_GPKDAT);
                tmp &= ~(1 << (4 + arg));
                tmp |= ( (!cmd) << (4 + arg) );
                writel(tmp, S3C64XX_GPKDAT);
                //printk (DEVICE_NAME": %d %d\n", arg, cmd);
                return 0;
        default:
                return -EINVAL;
        }
}
```

# static int __init dev_init(void)

```
static int __init dev_init(void)
{
        int ret;
        {
                unsigned tmp;
                tmp = readl(S3C64XX_GPECON);
                tmp = (tmp & ~(0xffffU<<16))|
(0x1111U<<16);
                writel(tmp, S3C64XX_GPECON);
                tmp = readl(S3C64XX_GPEDAT);
                tmp |= (0xF << 4);
                writel(tmp, S3C64XX_GPEDAT);
        }
        ret = misc_register(&misc);
        printk (DEVICE_NAME"\Harry: PGE
initialized\n");
        return ret;
}
```

# static void __exit dev_exit(void)

```
static void __exit dev_exit(void)
{
        misc_deregister(&misc);
}

module_init(dev_init);
module_exit(dev_exit);
MODULE_LICENSE("GPL");
MODULE_AUTHOR("FriendlyARM Inc.");
```