

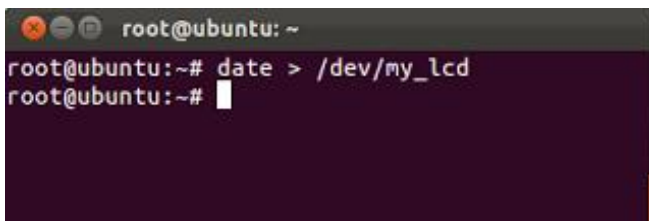
LINUX DEVICE DRIVER FOR A 16x2 LCD MODULE CONNECTED AT PARALLEL PORT

This is a small parallel port char driver for printing text on a 16x2 lcd module connected at parallel port of a PC. I did this as a part of learning linux kernel-module programming. May be this could be considered as a hello world device driver.

INTRODUCTION:

As we know, everything in Linux is treated as a file, even hardware devices like serial ports, hard disks, and scanners. In order to access these devices, a special file called a device node has to be present. All device nodes are stored in the /dev directory. Here, my 16x2 lcd connected to parallel port is also treated as a file and is accessed via a device node....

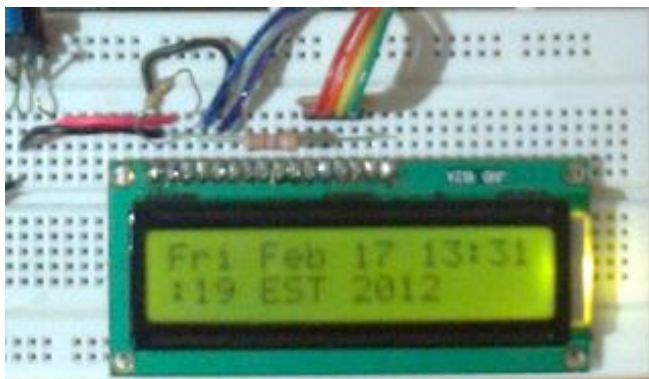
Now, after connecting the 16x2 lcd to parallel port (as in my circuit diagram), and then inserting the driver module, a message "DRIVER INSERTED" is displayed on the 16x2 LCD. Later, if we make a character special file (node) with major number as 61, and if we write any string to it (echo HELLO > file), then it will be displayed on the 16x2 LCD. Now, if we write a long string to it, then it is displayed on the lcd by scrolling it from bottom to top until the string is displayed completely ...



```
root@ubuntu: ~  
root@ubuntu:~# date > /dev/my_lcd  
root@ubuntu:~#
```

[[http://1.bp.blogspot.com/-](http://1.bp.blogspot.com/-w-b7vXYcnaA/Tz6pMLrhpeI/AAAAAAAAAKI/NyRMXyJ7I98/s1600/date.png)

[w-b7vXYcnaA/Tz6pMLrhpeI/AAAAAAAAAKI/NyRMXyJ7I98/s1600/date.png](http://1.bp.blogspot.com/-w-b7vXYcnaA/Tz6pMLrhpeI/AAAAAAAAAKI/NyRMXyJ7I98/s1600/date.png)]

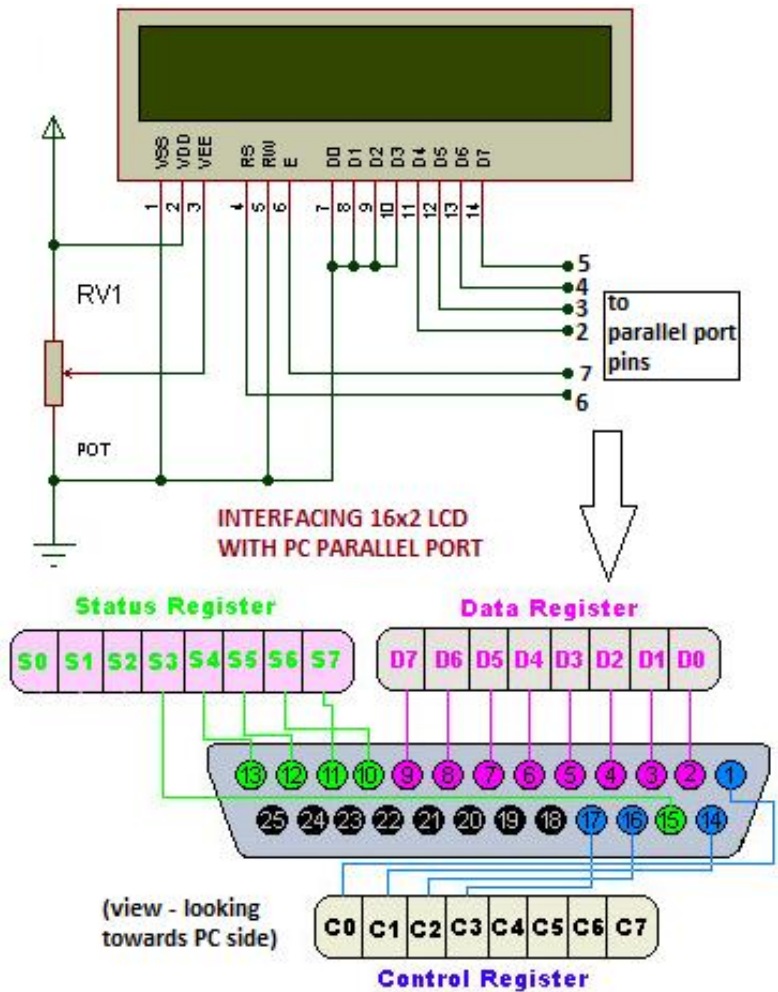


[[http://3.bp.blogspot.com/-6FA6W4su_qY/Tz6pTY3Pj9I/AAAAAAAAAKQ/ykRWRoENOTw/s1600/](http://3.bp.blogspot.com/-6FA6W4su_qY/Tz6pTY3Pj9I/AAAAAAAAAKQ/ykRWRoENOTw/s1600/date_lcd.png)
[date_lcd.png](http://3.bp.blogspot.com/-6FA6W4su_qY/Tz6pTY3Pj9I/AAAAAAAAAKQ/ykRWRoENOTw/s1600/date_lcd.png)]



[http://2.bp.blogspot.com/-rm9tE1uX3XO/Tz64j2_xiyI/AAAAAAAAAKw/4ADwGK9FY_w/s1600/lcd_to_parallelport.jpg]

CIRCUIT DIAGRAM:



<http://blog.vinu.co.in>
vinodstanur@gmail.com

[[http://4.bp.blogspot.com/-Ilp-](http://4.bp.blogspot.com/-Ilp-ZV2uCuU/Tz6ly6FY5MI/AAAAAAAAAKg/ouL3USgn_to/s1600/16x2+LCD.png)

[ZV2uCuU/Tz6ly6FY5MI/AAAAAAAAAKg/ouL3USgn_to/s1600/16x2+LCD.png](http://4.bp.blogspot.com/-Ilp-ZV2uCuU/Tz6ly6FY5MI/AAAAAAAAAKg/ouL3USgn_to/s1600/16x2+LCD.png)]

(i forgot to draw a connection between the ground of lcd and parallel port, don't miss it if any one is trying to do this. pin 25 to 18 is the ground, can select any one or all together)

DRIVER CODE - lcd.c : ('tested ok' in linux kernel 3.0):

```
#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/slab.h>
#include <linux/fs.h>
#include <linux/errno.h>
#include <linux/types.h>
#include <linux/proc_fs.h>
#include <linux/fcntl.h>
#include <linux/ioport.h>
```

```
#include <asm/system.h>
#include <asm/uaccess.h>
#include <asm/io.h>
#include <linux/delay.h>

MODULE_LICENSE("Dual BSD/GPL");

#define LCD(x) outb((c = ((0x0f & x) | (c & 0xf0)), 0x378)
#define RS(x) outb((c = (((~(1 << 4)) & c) | (x << 4))), 0x378)
#define EN(x) outb((c = (((~(1 << 5)) & c) | (x << 5))), 0x378)
#define en 0b00010000
#define rs 0b00100000
#define OUT(x) outb(x, 0x378)
#define MAXSIZE 17

static char lcd_buffer1[17];
static unsigned char c;
static char d_buf[MAXSIZE];
static int port;
static char lcd_space = ' ';
static int major = 61;

static int pport_open(struct inode *inode, struct file *filp)
;
static int pport_close(struct inode *inode, struct file *filp)
);
static ssize_t pport_read(struct file *filp, char *buf, size_t count, loff_t *f_pos);
static ssize_t pport_write(struct file *filp, char *buf, size_t count, loff_t *f_pos);
static void lcd_strobe(void);
static void data(unsigned char);
static void cmd(unsigned char);
static void clear(void);
static void lcd_init(void);
static void printlcd(char *);

static struct file_operations fops = {
    open: pport_open,
    read: pport_read,
    write: pport_write,
    release: pport_close
};

static int pport_open(struct inode *inode, struct file *filp)
{
```

```
        return 0;
    }

static int pport_close(struct inode *inode, struct file *filp
)
{
    return 0;
}

static ssize_t pport_write(struct file *filp, char *buf, size
_t count, loff_t *f_pos)
{
    if(count < MAXSIZE) {
        copy_from_user(d_buf, buf, count);
        d_buf[count] = 0;
        printlcd(d_buf);
        *f_pos += count;
        return count;
    } else {
        copy_from_user(d_buf, buf, MAXSIZE - 1);
        d_buf[MAXSIZE - 1] = 0;
        printlcd(d_buf);
        *f_pos += MAXSIZE - 1;
        return MAXSIZE - 1;
    }
}

static ssize_t pport_read(struct file *filp, char *buf, size_
t count, loff_t *f_pos)
{
    return 0;
}

int init_module(void)
{
    int a;
    a = register_chrdev(major, "registered_pport_61", &fops);
    if(a < 0) {
        printk(KERN_ALERT "error: can't register major number
%d\n", major);
        return a;
    }
    port = check_region(0x378, 1);
    if(port) printk(KERN_ALERT "pport cannot reserve 0x378\n"
);
    request_region(0x378, 1, "registered_pport_61");
    //lcd setup//
```

```
    outb(0, 0x378);
    udelay(10000);
    lcd_init();
    udelay(10000);
    printlcd("DRIVER INSERTED ");
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_ALERT "pport module is going to terminate\n");
;
    printlcd("DRIVER REMOVED ");

    unregister_chrdev(major, "registered_pport_61");
    clear();
}

static void lcd_strobe(void)
{
    EN(1);
    udelay(1);
    EN(0);
    udelay(1);
}

static void data(unsigned char data)
{
    RS(1);
    udelay(40);
    LCD(data >> 4);
    lcd_strobe();
    LCD(data);
    lcd_strobe();
    udelay(10);
    RS(0);
    udelay(10);
}

static void cmd(unsigned char command)
{
    RS(0);
    udelay(40);
    LCD(command >> 4);
    lcd_strobe();
    LCD(command);
    lcd_strobe();
}
```

```
static void clear(void)
{
    cmd(1);
    udelay(2000);
}

static void lcd_init(void)
{
    cmd(0x30);
    cmd(0x30);
    cmd(0x28);
    cmd(0x0c);
    clear();
    cmd(0x6);
}

static void printlcd(char *p)
{
    static int count = 0;
    count = 0;
    clear();
    cmd(0x80);
    while(lcd_buffer1[count])
        data(lcd_buffer1[count++]);
    count = 0;
    cmd(0xc0);
    while(*p) {
        if((*p != '\n') && (*p != '\t')) {
            lcd_buffer1[count++] = *p;
            data(*p);
        } else {
            data(lcd_space);
            lcd_buffer1[count++] = lcd_space;
        }
        p++;
    }
    lcd_buffer1[16] = 0;
    msleep(2000);
}
```

Makefile:

```
obj-m += lcd.o
```

```
all:
```

```
make -C /lib/modules/$(shell uname -r)/build M=$(PWD)
modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD)
clean
```

PROCEDURE:

- 1> save **lcd.c** to a new folder
- 2> create **Makefile** in the same folder
- 3> type "**make**" to build the driver module **led.ko**
- 4> Setup the 16x2 lcd as in circuit diagram and power on the 5v supply to lcd.
- 5> Insert the module using "**sudo insmod lcd.ko**"
- 5> To see the kernel message, type **dmesg** or use **tty3** or **4** to see it directly on screen
- 6> Now if every thing is all right, then we could see a text "**DRIVER INSERTED**" on the second row of 16x2 lcd.
- 7> Now create a node any where, with major number 61.
eg: "**sudo mknod /dev/my_lcd c 61 0**"

TESTING:

- Type "**echo HELLO WORLD > /dev/my_lcd**". This should print **HELLO WORLD** on the lcd.
- type "**date > /dev/my_lcd**" and observe the date on lcd.
- Type "**echo THIS IS A LARGE STRING CONTAINING MORE THAN 32 CHARACTERS. I WANT TO SEE HOW IT IS DISPLAYED ON THE LCD > /dev/my_lcd**" and see the long string on the lcd.
- Type "**ls > /dev/my_lcd**" as root and observe contents on the folder on the lcd.
- Type "**story.txt > /dev/my_lcd**" as root and read the story on LCD. :-)
- Now, we can also try to display some thing using a user program...

Hope this much test is enough....:-)

Screenshot of test and result:

```
root@ubuntu: ~  
root@ubuntu:~# echo Hello world :-\ ) It is wor  
king! > /dev/my_lcd  
root@ubuntu:~#
```

[http://4.bp.blogspot.com/-eJ4oJYl_IMs/Tz6o8z4y_PI/AAAAAAAAAJ4/055lbd4TXN8/s1600/hello+world+its+not+working.png]



[http://1.bp.blogspot.com/-qFLV6iFk1ZE/Tz6p0ZMhrII/AAAAAAAAAKA/eaBPtmYy4T8/s1600/its_working_lcd.png]

```
root@ubuntu: ~  
root@ubuntu:~# date > /dev/my_lcd  
root@ubuntu:~#
```

[<http://1.bp.blogspot.com/-w-b7vXYcnaA/Tz6pMLrhpeI/AAAAAAAAAKI/NyRMXYj7I98/s1600/date.png>]



[http://3.bp.blogspot.com/-6FA6W4su_qY/Tz6pTY3Pj9I/AAAAAAAAAKQ/ykRWRoENOTw/s1600/date_lcd.png]

A good tutorial about kernel module programming is available at
<http://tldp.org/LDP/lkmpg/2.6/html/index.html>
[<http://tldp.org/LDP/lkmpg/2.6/html/index.html>]

Posted 18th February 2012 by [Vinod.S](#)

Labels: [Linux](#), [MY HOBBY PROJECTS](#)

Download Free Forms



Find Essential US Government Forms. Install Free App & Download Forms!



[View comments](#)



Gilmar February 21, 2012 at 6:25 PM

Good tutorial, very informative.

How about USB driver? It would be more fun.

thanks

Ilms

[Reply](#)



gSalom February 26, 2012 at 3:36 AM

Good job. I have learnt a lot with this tutorial. Well done !!

[Reply](#)**Naveen Karuthedath** July 4, 2012 at 12:56 PM

Nice tutorial.. :)
thanks

[Reply](#)**omar ezzat** January 3, 2013 at 1:54 AM

can u make me a video of the test please

[Reply](#)**Osegueda** February 22, 2013 at 2:16 PM

No pude lograr la compilación con make.
que puede ser?

[Reply](#)**Winda** April 17, 2013 at 7:26 PM

hi, thanks for the tutorial.
I try it with pic16f877a, and i used usb serial port in my laptop.
But i failed in step 3 (3>type "make" to build the driver module led.ko)
could you please show me the details / step by step to do the third step? *i'm
beginner with this
Thanks

[Reply](#)**Sushant Bhangale** April 15, 2014 at 6:23 AM

Thanx for Help nice blogg

[Reply](#)**athul pk** September 22, 2014 at 6:02 AM

Thanks,
Am abgnr in lnx
ths s ma fst pgm.....thnx alot

[Reply](#)**Tintu Thomas** October 4, 2014 at 12:23 AM

well done...
keep going ...
very much helpful to beginners like me..

Thank u.. :)

[Reply](#)



Полиграф Шариков [January 8, 2015 at 12:34 AM](#)

asm/system.h no such file or directory compilation terminated

[Reply](#)

Enter your comment...

Comment as: Google Account ▼

Publish

Preview