

文章编号:1007-2780(2012)01-0103-05

基于 Linux 下的 OLED 显示模块设计

李 想^{1,2}, 郑喜凤^{1*}, 陈 宇¹

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033, E-mail: asine315@163. com;
2. 中国科学院 研究生院, 北京 100039)

摘 要: 分析了目前显示市场上各种主流显示器件的特性, 介绍了 OLED 显示模块以及 S3C2440 的结构。针对 OLED 显示模块中 8 位数据并行接口占用 I/O 管脚较多, 接口设计繁琐, 无法满足系统需求的情况, 采用 SPI 总线接口, 进行相应的软件、硬件设计, 达到了系统节约接口的要求。在占用资源较少的情况下实现了 OLED 屏幕的正常显示。

关 键 词: S3C2440; SPI; OLED 显示; 嵌入式 Linux
中图分类号: TN141. 9 **文献标识码:** A **DOI:** 10. 3788/YJYXS20122701. 0103

Design of OLED Display Module Based on S3C2440

LI Xiang^{1,2}, ZHENG Xi-feng^{1*}, CHEN Yu¹

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences
Changchun 130033, China, E-mail: asine315@163. com;
2. Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: The characteristics of main display devices in the current market is analyzed. And this paper chooses the OLED display module, based on the general requirements of the S3C2440-based-system and the advantages and disadvantages of the display devices above. By analyzing the structure of S3C2440, this work uses SPI bus interface to solve the problem that the 8-bit parallel interface take too many I/O pins. Through the design of the hardware and software, the design meets the demand of the system saving interfaces, and achieving the OLED screen display module application on the condition.

Key words: S3C2440; SPI; OLED display; embedded Linux

1 引 言

目前显示市场上主流的平板显示器件包括 CRT、PDP、LCD、LED、OLED 等。CRT 作为 20 世纪显示器的霸主, 由于其高辐射、高能耗、大体积等缺点, 逐渐淡出了显示市场。PDP 虽然重量更小, 亮度也较均匀, 但其抗压能力较弱, 不能广泛应用于可能产生意外重压的场所。而 LCD 虽具有诸如低压微功耗、节省空间、对视力损害较

小、电磁辐射低等诸多特点, 但 LCD 存在着视角较窄、响应速度较慢、适温范围较窄等缺点。作为一种新型的显示器件, OLED 以有机材料作为发光层, 通过电流激发有机材料产生发光。与 LCD 相比, OLED 的亮度更高、视角更宽、发光元件为固相, 故适温范围较宽; 与 PDP 相比, 其抗震性能更为优良。现今, OLED 已经广泛应用于手持设备、通讯终端、测控仪器等领域^[1]。

本文选择了维信诺 V0004-BA-004 模块作为

收稿日期: 2011-10-11; 修订日期: 2011-10-25
作者简介: 李想(1988—), 男, 吉林辽源人, 硕士研究生, 主要从事嵌入式系统设计工作。
* 通讯联系人, E-mail: xfzheng6598@yahoo. com. cn

OLED 显示模块,针对系统中 I/O 管脚资源紧张且布线繁琐的情况,采用了 SPI 串行接口,用较少的 I/O 口实现了显示功能。

2 OLED 显示模块的 SPI 接口电路设计

本设计的单片机选用三星公司生产的微处理器 S3C2440A。其突出特点为处理器核心是一个由 ARM 公司设计的 16/32 位 ARM920T 的 RISC 处理器,实现了 MMU、AMBA 总线和哈佛结构高速缓冲体系结构。片内含有 1.2 V 内核供电,3.3 V 外部 I/O 供电,2 通道 SPI,可以满足系统要求^[2]。

OLED 显示部分选用了维信诺公司生产的 V0004-BA-004 显示模块。该模块为 128 列×64 行点阵的 OLED 单色(绿色)、字符、图形显示模块;集成了 Solomon SSD1305T7 驱动芯片;模块内含 128×64 bits 显示 RAM,其中每一位数据对应着 OLED 屏幕上一点的明暗状态;该模块具备 SPI 总线接口、I²C 总线接口、8-bit 6800 总线接口和 8-bit 8080 总线接口,在系统实际运行中只需要 5 V 电压即可正常点亮屏幕,其逻辑框图如图 1 所示,其管脚功能如表 1 所示^[3]。

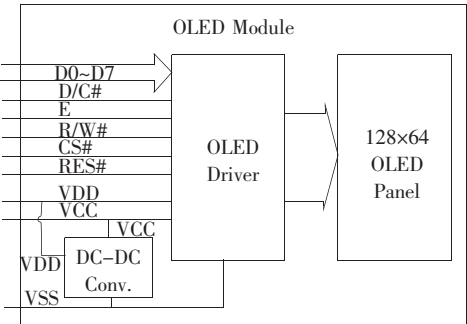


图 1 V0004-BA-004 OLED 显示模块逻辑框图
Fig. 1 Block figure of V004-BA-004 module

6800 总线接口与 8080 总线接口的数据总线均为 8 根,占用接口资源较多;而 SPI 总线仅需占用 2 根,分别作为时钟信号与数据总线,设计方便,结构简明。本设计的主控板 I/O 并未全部引出,系统接口资源较紧张。在这种限制下,采用了 SPI 接口。在 SPI 模式下,R/W#(WR#)和 E(RD#)均接地,D0 为时钟信号 SPICLK,D1 为数据线 SPIMOSI,D2 悬空,其余所有数据线接地。此种接口设计极大地节约了系统接口资源,

表 1 V0004-BA-004 显示模块管脚功能表
Table 1 Interface of V0004-BA-004 module

管脚序号	管脚名称	管脚功能
1	GND	接地
2	VCC	内置驱动电压
3	VDD	芯片逻辑电压
4~11	D7~D0	双向数据总线
12	E(RD#)	使能/读控制
13	R/W#(WR#)	读写/写控制
14	D/C#	数据/命令控制
15	RES#	复位信号
16	CS#	片选信号
17	BS2	总线接口选择引脚
18	BS1	总线接口选择引脚
19	DCEN	内部直流变换使能
20	FG	壳体接地

并且使得布线更为简洁,特别适合控制器与显示器分离的设计结构。

3 Linux 下的驱动设计

驱动程序是一种可以使计算机和设备通信的程序,相当于操作系统和硬件之间的接口,只有通过这个接口,才可以控制硬件设备的工作^[4]。对于 Linux 系统而言,其设备驱动程序主要有以下功能:

- (1)初始化设备;
- (2)实现硬件与操作系统内核之间的数据传输与读取;
- (3)读取应用程序传送给设备文件的数据及发送应用程序所要求的数据;
- (4)检测处理设备出现的错误;
- (5)释放设备。

对于驱动程序来说,没有必要完全实现上面所有的功能。在具体的应用设计中,只需要实现其部分功能,能够完成系统要求即可。本设计中所编写的驱动程序只需要实现接口函数中的 open、release、write、ioctl 等功能^[5-7]。

驱动程序接口结构的程序如下:
static struct file_ops oled_ops =
{

```
.owner = THIS_MODULE,
.write = oled_dev_write,
.ioctl = oled_dev_ioctl,
.open = oled_dev_open,
.release = oled_dev_release
```

```
};
```

各函数功能说明如下:oled_dev_open 操作用来打开 OLED 显示屏,oled_dev_ioctl 操作用来实现系统对 OLED 显示的控制命令,oled_dev_write 操作用来向 OLED 屏幕中写入需要显示的数据,oled_dev_release 操作用来关闭 OLED 屏幕.oled_dev_write 代码如下:

```
static ssize_t oled_dev_write(struct file *
filp, const char __user * buf, size_t count, loff_
t * offp)
{
    unsigned long flags;
    int i;
    if(count>0)
    {
        {
            // 这里先将数据拷贝到临时缓冲中,以防
            止在 irqsave/irqrestore 之间复制用户数据可能
            出现的睡眠。
            count=min(count,BUFFER_SIZE);
            if( copy_from_user ( global_dev-> temp _
buffer,buf,count))return -EFAULT;
        }
        spin_lock_irqsave (&global_dev-> lock,
flags);
        if(filp->f_flags&O_NONBLOCK)
        {
            //处理非阻塞打开的情况
            if(BUFFER_FREE()<1)
            {
                spin_unlock_irqrestore(&global_dev-
>lock, flags);
                return -EAGAIN;
            }
            else{
                //有多少写多少
                count=min(BUFFER_FREE(),count);
                goto write_exe;
            }
        }
        else{
            while(BUFFER_FREE()<count){
                spin_unlock_irqrestore (&global_dev->
lock,flags);
```

```
        if ( wait_event_interruptible ( global _
dev->wait_q,BUFFER_FREE()>=count))re-
turn -ERESTARTSYS;
```

```
        spin_lock_irqsave (&global_dev->
lock,flags);
```

```
    }
```

```
}
```

至于其余设备的相关操作,可以参考 oled_dev_write 来实现^[8]。在实现了各入口函数后,需要在操作系统内核启动时注册字符设备。在注册设备成功后,需要对 OLED 进行初始化设置。OLED 初始化程序片段如下:

```
{
    static void oled_init(void)
    write_com(0xda); //管脚硬件配置
    write_com(0x12); //对比度控制值
    write_com(0xa0); //配置分段映射
    write_com(0xc8); //设置输出扫描
    write_com(0xd8); //设置低功耗
    write_com(0x00); //设定显示模式
    write_com(0xdb); //设置取消级别
    write_com(0xd9); //预充电时间
    write_com(0xd5); //设置时钟分频
}
```

编译完毕之后将驱动程序加载到内核中,将驱动程序加载到内核中有两种方式,即动态加载和静态加载^[9-10]。静态加载时,驱动程序会直接加载到内核之中,在配置内核时可以自由裁剪,但静态加载的缺点是调试驱动程序较繁琐,每次修改都需要重新编译下载内核,不利于节省时间。因此在调试成功前,应采用动态加载方式。利用 Linux 的 module 特性,将驱动程序编译成模块,在系统启动后利用 insmod 命令将驱动程序添加进去,不需要驱动程序时利用 rmmod 命令卸载驱动。这样,在反复的调试驱动程序时,就免去了每次修改驱动程序代码后都需重新编译内核和重新启动系统的繁琐操作,节省了开发时间。而在调试结束后需要将驱动程序静态加载到内核中。应用程序以 fopen 的方式打开驱动程序,在返回句柄后就可以以该形式访问 open、ioctl、write、release 等接口,在以上操作完成后通过 close 函数关闭 OLED 驱动程序的接口。

4 OLED 显示模块的文字显示程序设计

驱动程序的目的是提供计算机系统与底层硬件的通信接口。若想控制硬件工作,仅有驱动程序还不够,必须要在驱动程序的基础上编写应用程序。OLED 显示模块的最主要功能是屏幕显示,主要显示内容为文字显示,因此本文所述的应用程序指的是 OLED 文字显示程序。在 V0004-BA-004 显示模块内置了 128×64 bits 的显存, RAM 容量为 1 kB。显存的任务是存储需要显示在屏幕上的数据,屏幕上各像素点的显示状态与显存的各个位的数据一一对应,显存的数据则直接作为显示信号。本设计中,若显示数据位“1”,则与之相对应的像素点为点亮状态;若数据显示为“0”,则与之相对应的像素点为熄灭状态。显存 RAM 分为 8 页,每页为 8 行 $\times$ 128 列,这样每页中的每列刚好为 8 bit,低位在上,高位在下,每个字符均占两页,其中中文字符占 16 列,英文字符占 8 列。

V0004-BA-004 显示模块中没有内置字符发生器,因此需要提取字模来生成需显示的文字字库,本设计采用字模 3 软件提取字库。字模 3 软件是面向液晶产品开发的专用字模提取软件,操作简单,功能强大,可以自由设定字模的大小及字体,并同时支持 C 语言和汇编语言数据格式。根据本设计的实际需要,采用数据纵向、字节倒序的字模格式。例如,从上到下 8 个像素点的状态为“* * -- * * -”(* 代表点亮状态,-代表熄灭状态),则转换后的字模数据为 B01100011,即 0x63。之后调用写入字符函数,将显示数据送入显存,即可稳定显示字符。

设计中,向 OLED 屏幕写入中文字符的方式为逐字分页写入,即将需要显示的中文或英文字符分成上、下两个部分,分别写入上、下连续的两页。例如,若需要在 OLED 屏幕上显示汉字“李”,则需要将“李”字的字模数组均分为两部分,将字模数组的前半部分显示在第一页,将字模数组的后半部分显示在第一页正下方。向 OLED

屏幕写入中文字符的函数为 write_chs,代码如下,其余函数的实现可参考 write_chs 函数:

```
void write_chs(int fd,int r,int c,int n)
{
    unsigned short arg;
    char p1[16],p2[16];
    unsigned char sc_temp[32];
    int i;
    memcpy(sc_temp, *(chinese+n),32)//将
字库中的中文字模复制到临时数组中
    for(i=0;i<16;i++)
        p1[i]=sc[i];
    for(i=16;i<32;i++)
        p2[i-16]=sc[i];//将字模数据分为上下
两页
    i=0;
    if(ioctl(fd,SET_POS,&arg))
        printf("ioctl error! \n");
    MKPOS(arg,r++,c);//定位屏幕坐标
    write(fd,p1,16);//写中文字符的上半
部分
    MKPOS(arg,r,c);
    if(ioctl(fd,SET_POS,&arg))
        printf("ioctl error! \n");
    write(fd,p2,16);//写中文字符的下半部分
}
```

若有更复杂的要求,则可参考以上程序通过软件编程来实现。

5 结 论

针对 OLED 显示模块 8 位数据并行接口占用系统 I/O 资源较多,布线繁琐的缺点,采用 SPI 总线接口,在嵌入式 Linux 系统下完成了 OLED 显示模块的驱动设计和应用程序设计,节省了系统接口资源,简化了系统布线。编写的驱动程序与显示程序经过实际编译与调试,证明了其能够很好地解决在系统外设模块较多情况下的布线繁琐及接口资源不足的问题,在满足系统要求的前提下,取得了较好的显示效果。

参 考 文 献:

[1] 金珏,郑喜凤,丁铁夫. 基于 DSP 的 OLED 显示器软硬件设计及实现 [J]. 液晶与显示,2008,4(2):209-212.

