

· 电路与控制 ·

基于FT245的Linux USB驱动的设计

曾 强,赵 娟,魏 林,王文廉

(中北大学信息与通信工程学院,山西 太原 030051)

摘 要:Linux操作系统具有开源、稳定且便于移植到嵌入式设备中的优点,因而为嵌有Linux系统的设备设计USB驱动,就成为实际工程需求。为解决支持arm的Linux USB驱动,对USB驱动的数据结构进行了分析,修改了USB驱动的框架程序,用编译模块方法实现了对FT245 USB芯片的驱动加载,完成了基于arm+Linux平台的USB驱动设计,并成功应用到专用测试系统中,通过实验,验证了设计的正确性。

关键词:Linux;便携式设备;USB;arm

中图分类号:TP311.11

文献标识码:A

文章编号:1673-1255(2011)01-0070-04

Design of Linux USB Driver Based on FT245

ZENG Qiang,ZHAO Juan,WEI Lin,WANG Wen-lian

(School of Information and Communication Engineering, North University of China, Taiyuan 030051, China)

Abstract: The advantages of Linux operating system are open source, stabilization and easy to port in the embedded devices, so the actual project needs to design USB driver for devices with the embedded Linux system. In order to solve Linux USB driver supporting arm, the data structure of USB driver is analyzed, and the frame programs of USB driver are modified. The driver loading to FT245 USB chip is realized by using compile module method, USB driver design based on arm + Linux platform is completed, and the design successfully applies in the specific test system. Finally, the correctness of design is verified by experiments.

Key words: Linux; portable device; USB; arm

Linux操作系统是在嵌入式领域新兴起来的开源操作系统,相比于windows操作系统, Linux具有公开源代码、系统稳定、运行效率高、网络功能完善、便于移植到嵌入式设备中等优点^[1]。

USB(universal serial bus)全称通用串行总线,最新标准USB2.0全速高达480 Mbps且具有数据传输可靠,热插即用等优点,在便携式移动设备中得到广泛的应用^[2-4]。

FT245是FTDI公司开发的USB芯片,具有抗干扰、传输稳定的优点,能单芯片实现并行FIFO缓存区

的双向数据传输,且支持VCP(虚拟串口)功能,广泛应用在测试系统中。如表1所示,FTDI公司为该芯片

表1 FTDI提供的USB驱动所支持的系统和平台

SYSTEM	X86	X64	PPC	ARM
Windows*	◆	◆	-	-
Linux	◆	◆	-	-
Mac OS X	◆	◆	◆	-
WindowsCE4.0	◆	-	-	◆
WindowsCE6.0	◆	-	-	◆

制作了支持各大操作系统及应用平台的驱动程序,但

收稿日期:2010-12-20

基金项目:山西省自然科学基金(2009011023)

作者简介:曾强(1985-),男,山西省榆次人,硕士研究生,主要研究方向为智能通信与嵌入式系统;赵娟(1985-),女,山西省太原人,硕士研究生,主要研究方向为图像三维重构系统;魏林(1985-),男,山西省长治人,硕士研究生,主要研究方向为光电测试系统;王文廉(1978-),男,博士,四川成都人,主要从事动态测试技术、智能仪器研究。

却没有支持arm的Linux USB驱动。该设计以实际应用出发,针对FT245芯片详细介绍了arm9平台下Linux USB驱动的开发过程及其移植。在专用测试系统中,该驱动已经应用到手持式数据读取设备中,可取代笔记本进行野外测试设备的数据读取。

1 USB及Linux USB驱动的基本原理

1.1 USB的基本原理

USB是一种快速、双向的串行接口,它采用总线拓扑结构,其主控制器通过USB集线器可扩展主机的sub端口^[2,5]。USB具有4种传输方式:控制传输、中断传输、批量传输、等时传输^[2,6],其数据通信是双向的,数据从主机传向USB设备称下行通信,数据从USB设备传向主机称上行通信。控制权是在host方,不支持设备间的直接通信。

USB设备是由接口和端点构成的。端点是USB通信最基本的形式,它只能是往一个方向传送数据^[6]。接口是用来绑定到设备上的,一个接口只处理一种USB逻辑连接,一个USB设备可以有多个接口,不同的接口可以代表不同的功能,USB驱动程序是绑定到接口上的。此外,一个USB设备通常又有多个配置的,而且可以在配置之间切换以改变设备的状态;但是一个时刻只能激活一个配置,一个配置通常具有一个或者更多的接口,如图1所示^[7]。

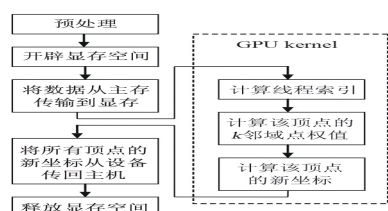


图1 USB系统结构框图

1.2 基于Linux的USB驱动的工作原理

Linux操作系统是把设备看成是一种文件来处理的,这一点有别于其他的操作系统。这类文件称为设备文件,设备文件分为字符设备和块设备,Linux是通过对这些设备文件进行类似于普通文件的操作来控制设备的。

USB是一种字符设备,它的驱动程序是系统内核和USB芯片之间的接口,为应用程序屏蔽了硬件细节,应用程序通过调用驱动中定义的函数来实现

对USB设备的控制与数据的读写。

2 基于FT245的USB驱动的设计流程及其移植

在Linux的内核源码中有一个USB驱动的框架程序,文件名为usb-skeleton.c,目录在/drivers/usb下,大多数的USB驱动都是在这个框架的基础上修改的。以下针对FT245芯片修改USB驱动的框架程序。

2.1 USB驱动程序中的一些重要的数据结构

USB驱动程序用到两个宏,如下:

```
#define FT245_VENDOR_ID 0x0403 //VID
#define FT245_PRODUCT_ID 0x6001 //PID
```

这两个宏的值是USB芯片制造商所提供的,VID是芯片制造商ID,PID是芯片的识别码,不同厂商不同型号的USB芯片的VID和PID是不同的,FT245芯片的VID=0x0403,PID=0x6001。

驱动程序中要有一个支持的设备列表,通过VID和PID的值来识别设备。程序如下:

```
static struct usb_device_id FT245_table[] = {
    { USB_DEVICE(FT245_VENDOR_ID, FT245_ID) },
    {}
};
```

MODULE_DEVICE_TABLE(usb, FT245_table);

File_operation是驱动程序中一个很重要的类,它定义了一组常见文件I/O函数,而这些函数的使用类似于普通文件的I/O操作^[8,9]。程序如下:

```
static struct file_operations ft245_fops = {
    .owner = THIS_MODULE,
    .read = ft245_read,
    .write = ft245_write,
    .open = ft245_open,
    .release = ft245_release;};
```

这里定义的函数是USB驱动程序的入口点,当要打开USB设备时,使用系统函数Open(),系统就会自动调用驱动中的ft245_open()函数,其他函数也类似,如write()对应ft245_write(),read()对应ft245_read(),close()对应ft245_release()。

2.2 USB驱动的具体实现

当一个USB设备被安装而内核认为USB驱动程序应该处理它时,ft245_probe()函数被调用,这是一个探测函数,检查传递给它的设备信息,确定驱动程序是否真的适合该设备。

在ft245_probe()函数里,用一个循环来访问该

接口中存在的每一个端点,来寻找批量IN端点和批量OUT端点,下面是寻找批量OUT端点的程序。

```
for (i = 0; i < iface_desc->desc.bNumEndpoints; ++i) {
    endpoint = &iface_desc->endpoint[i].desc;
    if (!ft245dev->bulk_in_endpointAddr &&
        (endpoint->bEndpointAddress & USB_DIR_IN) && ((endpoint->
        bmAttributes & USB_ENDPOINT_XFERTYPE_MASK) =
        USB_ENDPOINT_XFER_BULK)) {
        /*找到一个批量OUT端点*/
        ft245dev->bulk_out_size =
            le16_to_cpu(endpoint->wMaxPacketSize);
        GFP_KERNEL);
        ft245dev->bulk_out_endpointAddr =
            endpoint->bEndpointAddress;
    }
}
```

wMaxPacketSize是端点一次可以处理的最大字节数,它的大小必须小于或等于FT245中收发缓存区的长度。FT245发送和接收缓存区的长度分别为384 bytes和128 bytes。在实际传输中,数据量如果大于wMaxPacketSize值会被分割。

bEndpointAddress是特定的USB地址,结合USB_DIR_IN和USB_DIR_OUT就可确定该端点的数据是传向设备还是主机。

USB数据的发送与接收是通过读写设备来实现的,当要读写设备时,需要调用ft245_write()和ft245_read()这两个函数。

```
static ssize_t ft245_read(struct file *file, char *buf, size_t
len, loff_t *ppos)
static ssize_t ft245_write(struct file *file, const char
*user_buffer, size_t count, loff_t *ppos)
```

参数file是文件指针,buf、user_buffer指向用户缓存区的指针,len、count是传输数据的长度,ppos是当前文件中的位置指针。

在usb-skeleton.c的程序代码中,skel_read()和skel_write()所使用的方法是不同的,在skel_read()中使用两个简单的接口函数:usb_bulk_msg()和usb_control_msg()来控制 and 传输数据,一般应用于只发送或接收一些简单的数据。而在skel_write()函数中是通过urb(USB request block)的传输方式,Urb可以看成是一个USB的驱动和USB端点通信的桥梁,可用于发送或接收一些复杂的数据。

由于FT245芯片一般用于双向的数据通信,发送和接收都有可能存在一些复杂的数据,所以ft245_read

()和ft245_write()都采用urb的传输方式。

当驱动程序有数据要发送或接收时,要分配一个urb来把数据传输给设备。

```
urb = usb_alloc_urb(0, GFP_KERNEL);
```

然后通过copy_from_user(tbuffer, user_buffer, count)或copy_to_user(buf, &ft245dev->read_buffer[ft245dev->rbufhead], len)函数完成用户空间与内核空间的数据交换。使用usb_fill_bulk_urb()初始化urb属性,再用usb_submit_urb()函数把数据发出去,如果urb数据被成功传输出去,urb回调函数将会被内核调用。这里要注意的是,由于FT245在把数据发往主机时会在每包数据的头加两个字节的标志位,所以在ft245_read()函数中应该忽略掉这两个字节的数,程序如下:

```
ft245_read(struct file *file, char *buf, size_t len, loff_t *ppos)
{...
down_interruptible(&ft245dev->sem);
...
effect_len = urb->actual_length - 2;
memcpy(&ft245dev->read_buffer[ft245dev->rbufhead],
&ft245dev->bulk_out_buffer[2], effect_len);
...
copy_to_user(buf, &ft245dev->read_buffer[ft245dev->
rbufhead], len)
up(&ft245dev->sem);
...}
```

为了避免在多个进程访问时出现冲突,程序中使用信号量编程,down_interruptible()与up()函数分别实现信号量的获取与释放。

2.3 USB驱动的加载

驱动的加载分为2种,一种是直接加载到内核代码中的,只要在Linux系统启动的时候,驱动就会被加载,但是这样内核会变得十分庞大,影响执行效率,这是一种静态的加载方式。通常采用的是另一种方法,把驱动编译成模块,使用insmod和rmmod命令加载或卸载驱动程序,这是一种动态的加载方式。

3 实验结果与分析

图2是一个arm开发板和一个专用测试记录仪,它们之间用一根USB数据线连接,数据记录仪的USB接口采用FTDI公司的FT245 USB芯片,arm开发板使用Linux作为操作系统,在该系统上利用

QT开发软件制作USB读数应用程序,完成了数据的传输任务。该设计实现了基于arm的便携式USB读数设备,可取代笔记本完成野外读数任务。

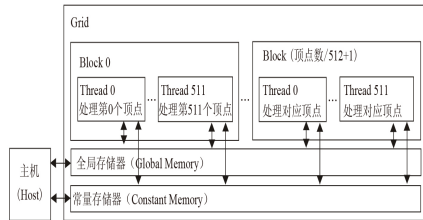


图2 arm开发板与数据记录仪

图3是利用LabVIEW做的数据显示界面,经过多次试验,所显示的数据波形与使用FIDI公司官方提供的FT245驱动的PC机读取的数据波形完全相同。所用试验表明驱动的设计是成功的。

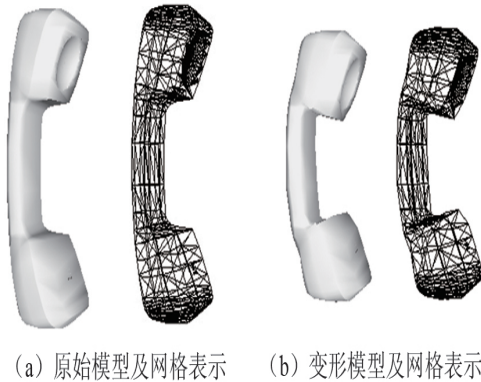


图3 基于LabVIEW的数据显示界面

4 结束语

针对FT245芯片设计制作Linux操作系统下的USB驱动程序,并移植到arm开发板上,经实验验证驱动设计是成功的。该设计在专用测试系统工程的项目中已得到应用,成功解决了FT245 USB芯片在linux+arm平台下没有驱动的难题,具有实际意义。

参考文献

- [1] 魏鹭,张焕强,方贵明. 基于Linux的USB驱动程序实现[J]. 计算机应用,2002,22(8):17-19.
- [2] 蔡震,蒋辉柏,周利华. Linux系统下USB设备驱动程序的开发[J]. 计算机测量与控制,2003,11(2):135-137.
- [3] 冯光磊,郭忠文. 基于ARM和Linux的USB OHCI驱动的设计与实现[J]. 计算机应用,2009,29:53-55.
- [4] 刘永祥,贺金平,吕杨. 基于Linux的USB设备驱动程序实现[J]. 无锡职业技术学院学报,2006,5(3):23-25.
- [5] 艾妮,曾成斌. Linux系统下USB2.0设备驱动程序的开发[J]. 现代机械,2005(2):16-17.
- [6] 闫婷婷,龙卓群. 基于Linux的USB从设备驱动研究[J]. 电子元器件应用,2009:11(10):55-57.
- [7] 杨建华,黄宇东,陈安. 基于嵌入式Linux的USB驱动设计[J]. 福建电脑,2009(4):11-14.
- [8] 赵昌明 穆煜. 嵌入式Linux应用开发教程[M]. 北京:人民邮电出版社,2009:253-264.
- [9] 梁孔科,杨林楠,张丽莲. 基于S3C2410与Linux的USB驱动的设计[J]. 福建电脑,2008(4):2-4.
- [10] Alessandro Rubini. Linux Device Drivers[M]. 2nd. USA: O'Reilly,2001:101-160.
- [11] Linux USB developer and support information[EB/OL], (2010.8.5) [2010.10.20], <http://www.linux-usb.org>.
- [12] 赵凯生,刘爽. 基于ARM的线阵CCD测量系统应用分析[J]. 光电技术应用,2006,21(1):17-19.

(上接第69页)

参考文献

- [1] 王昌明. 实用弹道学[M]. 北京:国防工业出版社,1994.
- [2] 周承仙. 连发射弹弹丸速度及弹着点坐标测量系统的研究[D]. 太原:中北大学,2008.
- [3] 赵豫姝,赵冬娥,赵辉. 一种新型激光测速系统的设计与应用[J]. 弹箭与制导学报2008(4):285-287
- [4] 白兴满. 炮用激光测速系统的设计[D]. 太原:中北大学,2003.

- [5] 王连海,杨慧武. 组合平行激光光幕靶. 中国200420007418.7 [P].
- [6] 范茂彦,张丽芳. 实弹射击自动精确报靶系统研制[J]. 传感器世界,2003(11):3-5.
- [7] 秦向军,周汉昌. 激光光幕自动报靶系统的设计与实现[J]. 计量与测试技术,2009(3):14-17.
- [8] 黄智伟. FPGA系统设计与实践[M]. 北京:电子工业出版社,2005.
- [9] 马潮. AVR单片机嵌入式系统原理与应用实践[M]. 北京:航空航天大学出版社,2007.