

# Linux 下基于 python 的 USB 通信程序开发

蒙卓

(上海工程技术大学 机械工程学院 上海市松江区 201620)

**摘要:** Linux 操作系统通过 libusb 库完成各种和 USB 设备的通信,但其开发过程过于繁杂,以 python 语法简洁的特性调用 libusb,可以极大减少对 USB 设备通信开发难度且无须编写特殊驱动。以 CY7C68013 芯片为例,在 Linux 下通过 python 调用 libusb 对 Cypress Fx2LP 系列芯片通信方法,并介绍了上传数据到芯片内 RAM 具体应用的方法。

**关键词:** Linux USB 设备通信 libusb python CY7C68013 芯片

**The Development of communication program for USB device base Python under Linux**  
Meng Zhuo

(Mechanical Engineering College, Shanghai University of Engineering Science, Shanghai 201620)

**Abstract:** Communications between USB device and Linux Operation System base on library named libusb. Developed on python reduce cycle time, less complexity on USB device driver by python's concise and simple understandable structure. The article focus on how libusb dynastic library callbacked by usb module in Python, specially how to upload data to on-chip-RAM of CY7C68013 produced by Cypress Semiconductor Company.

**Keywords:** Linux communication USB libusb python CY7C68013

## 0 引言

Python 语言是一种面向对象、内置高级数据结构、语法简洁严谨、可读性高、易于扩展的解释型脚本语言。最突出的特点是开发周期短和以模块的形式调用各种其他语言编写的库文件,因此 Python 受到越来越多科技人员的青睐。

libusb 是 Linux 操作系统的硬件支持库,主要负责操作系统与 USB 设备进行通信,使得开发者简化系统底层的依赖和减少重复编程,开发的强度下降、大大缩短开发周期。

Fx2LP 系列芯片是 Cypress 公司推出的 USB2.0 集成微处理芯片,USB 收发器、增强型 8051 单片机及一个 16KB 片上 RAM、一个 4KB FIFO 存储器和一个通用可编程接口 (GPIF) 外部接口[2]封装于单一芯片中且使用 USB 接口进行数据交换,外围固件的开发简化。

Python 中的 usb 模块用 C 语言格式封装了 libusb 函数接口,鉴于 Python 开发上的高效性和 Fx2LP 系列芯片外围固件简单的特性,利用 Python 开发基于 libusb 的 Fx2LP 系列芯片的通信程序,可以缩短开发周期,而且同样能达到其他语言开发的性能和效果。

## 1 实现原理和背景

### 1.1 Linux 下的 USB 系统体系结构简介

USB 是一种分层总线结构,USB 设备和主机之间的信号传输通过 USB 控制器实现。

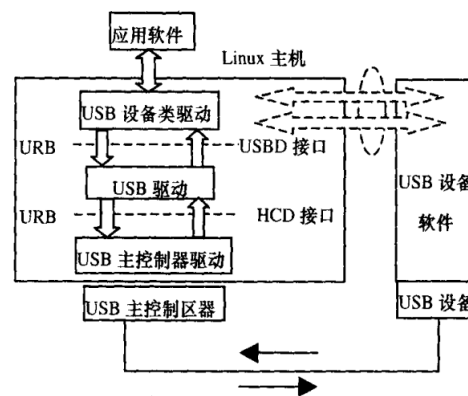


图 1.Linux USB 驱动体系结构<sup>[3]</sup>

Linux 操作系统是通过 libusb 来完成各种与 USB 设备之间的信号传输,libusb 向用户提供应用程序接口 (API) 控制各种 USB 信号传输,包括四种数据传输类型:控制传输,等时传输,中断传输,数据块传输四大类接口;两种传输模式:同步传输模式,异步传输模式。其中控制传输采用特定的方式使用控制数据来对设备进行设置,数据传输是无损性的<sup>[4]</sup>。

### 1.2 Fx2LP 系列 USB 接口简介

通过 Cypress 公司提供特定的 USB 控制指令,表(1)<sup>[4]</sup>,Fx2LP 内核能够直接编辑内 16KB RAM 和

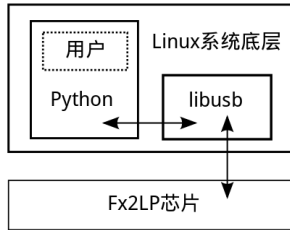
内部 512 字节便笺式 RAM 的数据内容，使用此类指令进行程序的数据上传与下载。

数据方向	bmRequest	bRequest
上传至芯片	0x40	0xa0
从芯片下载	0xc0	0xa0

表(1)Fx2LP 控制指令位

### 1.3 基于 Python 的 USB 通信

使用 Linux-python 组件中的 usb 模块（以 Ubuntu 10.4 发行版为例，包名称为 pyusb），该模块以 ctype 格式调用 libusb 库中的动态接口链接函数<sup>[5]</sup>，并提供了 python 化的接口。通过 import 指令调用此模块完成与 libusb 接口的通信从而到达与 USB 设备通信的目的。usb 模块文件位于 /usr/lib/pymodules/python2.6/usb.so。



图(2) python libusb 和 linux 信息结构图

## 2 程序实现

### 2.1 搜索芯片并加载

加载 usb 模块后，python 已可访问所有 USB 设备，所以需要程序通信目标设备加载至 python 的函数对象中。具体程序如下：

```

#初始化函数及模块
import usb
#加载 CY7C68013 VID、PID
vendor_id=0x04b4
product_id=0x8613
#搜索并加载
busses = usb.busses()
dev = "" #强制指定 dev 的类型为字符
for bus in busses: #历遍所有 USB 组
    devices = bus.devices
    for d in devices: #历遍所有此组上的设备
        if d.idVendor==vendor_id and
d.idProduct==product_id:
            dev = d
#若 VID、PID 均符合将此设备交 dev 供以后调用
            break
handle = dev.open() #打开设备

```

### 2.2 控制传输模式

usb 模块中的 DeviceHandle 类中包含 controlMsg 这一函数，为偏移量、数据长度赋值或者填充缓冲数据后直接调用即可。

#### 2.2.1.导出 RAM 函数

因返回的 data 值类型为元组，如需将数据传输至 Linux 终端(tty)，则需要调用 print 函数。

```

def dump_ram(offset,lenth):
    data=handle.controlMsg(requestType=0xc0,request=0xa0,value=offset,index=0x00,buffer=lenth,timeout=100)
    print data

```

#### 2.2.2 写入 RAM 函数

填充缓冲的程序原始数据后，调用函数就可直接向此偏移量下的 RAM 上传数据。具体函数如下：

```

def write_ram(offset,buf):
    handle.controlMsg(requestType=0x40,request=0xa0,value=offset,index=0x00,buffer=buf,timeout=100)

```

### 2.3 批量传输模式

usb 模块中的 DeviceHandle 类已经提供了 bulkRead、bulkWrite 函数，使用时调用即可，由于设备重枚举后，大部分 PID 或 VID 发生了变化，需要通过程序重新寻找并设定的新设备的 PID 和 VID，然后使用 configurations 类获得此设备的配置信息和端点信息，进行进一步的设定。导出和写入批量传输程序如下：

```

def bulk(dendp,lenthl,wendp,buf): #*endp 为端点号实数
    #初始化 USB 配置
    handle.detachKernelDriver(0) #获得 Linux 底层的设备控制权
    if dendp != " " : #为导出模式时
        data=handle.bulkRead(dendp,lenthl,100)
    if wendp != " " : #为写入模式时
        handle.bulkWrite(wendp,buf,1000)

```

### 2.4 结束会话

对设备的各种操作完成后，应释放资源和结束会话，具体程序如下：

```

handle.releaseInterface() #释放接口
dev.close() #关闭设备

```

## 3 结束语

python 对动态链接库的调用简洁高效，usb 模块省去了重复开发 USB 驱动的时间，而且 python 的跨平台特性使得程序能适应各种平台环境。

参考文献

- 
- [1] Libusb Group. Device handling and enumeration [OL].[http://libusb.sourceforge.net/api-1.0/group\\_\\_dev.html](http://libusb.sourceforge.net/api-1.0/group__dev.html) , 2010-08-30
- [2] 赛普拉斯半导体公司. EZ-USB FX2LP (TM) USB 微控制器 高速 USB 外设控制器 [OL] . <http://www.cypress.com/?docID=14023> , 2010-08-30
- [3] 杨伟, 刘强, 顾新. Linux 下 USB 设备驱动研究与开发 [J]. 计算机工程 2006, 32
- [4] 钱峰. EZ-USB FX2 单片机原理、编程及应用 [M]. 北京: 北京航空航天大学出版社, 2006
- [5] Pyusb Group. PyUSB [OL]. [http://sourceforge.net/apps/mediawiki/pyusb/index.php?title=Main\\_Page#About](http://sourceforge.net/apps/mediawiki/pyusb/index.php?title=Main_Page#About) , 2010-08-30

作者简介: 蒙卓 (1988-), 男, 汉族, 在读学士  
通讯地址: 上海市松江区龙腾路 333 号, 机械工程学院 邮编: 210620 电话: 021-67791492 、 15900642346 Email : [i@mengzhuo.org](mailto:i@mengzhuo.org)。