

基于 Linux 操作系统的 USB 设备驱动 程序开发及传输实现*

万玛宁,侯志化

(首都师范大学 信息工程学院, 北京 100048)

摘 要: 要在 Linux 操作系统下开发一款支持智能设备与主机之间通过 USB 通信的读写控制程序,为了满足控制和通信的需要,要求快速、便捷地开发一款 USB 设备驱动程序以建立主机和智能设备的连接,同时能够保证数据的正确和稳定传输. 为解决上述问题,本文实现了一种 Linux 操作系统下,基于 CyAPI 库的 USB 驱动程序开发方法. 在智能设备读写控制程序中所开发的 USB 驱动程序进行了验证,通过调用设备驱动程序直接实现了对 USB 设备的控制和数据传输,同时能够及时正确的收到智能设备的信息反馈,设备驱动程序展现了良好的稳定性.

关键词: 设备驱动程序;Linux;USB;CyAPI

中图分类号: TP274. 2

文献标识码: A

文章编号: 1000—7180(2018)11—0124—04

Development and Implementation of Transmission of USB Device Driver Based on Linux Operating System

Wan Ma-ning, Hou Zhi-hua

(College of Information Engineering, Capital Normal University, Beijing 100048, China)

Abstract: To develop a read-write control procedures which can support the USB communication between intelligent device and the host under Linux operating system, in order to meet the needs of the control and communicate, under the requirement of developing a USB device driver to establish the connections quickly and conveniently, meanwhile, the transmission between host and intelligent device must be correct and stable. In order to solve the above problems, this paper realizes the development method of USB driver based on CyAPI library under a Linux operating system. In the the development of read-write control procedures of intelligent devices, the USB driver is verified, by calling device driver directly realized the control of the USB device and data transmission, at the same time the information feedback of intelligent device can be received timely and correct, the device driver shows good stability.

Key words: device driver; linux library; USB; CyAPI

1 引言

USB(通用串行总线)因其操作简单、即插即用、通信速率快和可靠性高等优点,已经成为 PC 机和外设间接口的优先选择,而设备驱动程序在 PC 机

和外设的通信之中起到了沟通桥梁作用^[1]. 随着嵌入式技术的发展以及 IOT(物联网)和 CPS(信息物理系统)的应用,系统内的数据量日益庞大,USB2. 0 已经渐渐不能满足数据传输的需要,相较于 USB 2. 0 的最大传输带宽 480 Mbps,USB3. 0 最大传输

收稿日期: 2018—04—20; 修回日期: 2018—05—22

基金项目: 国家自然科学基金(61741211, 61772350, 61472260); 国家高技术研究发展计划(863 计划)(2008AA000000); 体系结构国家重点实验室开放课题(CARCH201607), 北京市科技新星计划(XX2018081) 深圳市科技计划项目(JCYJ20150529164656096, JCYJ20170302153955969); 装备预先研究项目(公开)(JZX2017-0988/Y300)

带宽可以达到 5.0 Gbps(640 MB/s),能够有效支持实时数据传输,并且向下可以实现对 USB2.0 的兼容. 因为 USB3.0 在数据传输方面高速和稳定的特性,使其在嵌入式领域得到了更加广泛的应用,通用性极佳^[2]. 随着影音技术的进步和拍照设备的升级换代,视频和图像成像技术有了极大的提高,但是与此同时所占用的文件存储空间也越来越大^[3],为了智能视频采集设备和主机之间实时技术传输的需要,设计一款内置于视频设备的基于 VIRTEX-5 芯片的通用数据采集设备,通过使用 USB3.0 接口实现同主机的互联通信,满足了当前嵌入式系统设计者的越来越多样化的需求,工作稳定高效,有效地保证了系统开发周期^[4-5],使用这款芯片可以有效实现数据采集、处理和通信等功能,满足了智能设备和主机之间数据高速和实时传输的需要,数据传输部分系统结构如图 1 所示.

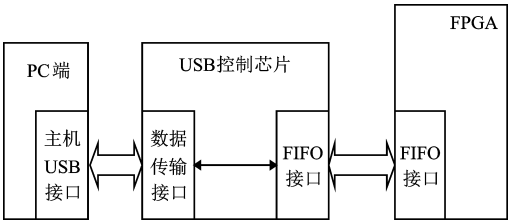


图 1 数据传输结构图

本文在 Cypress 提供的 CyAPI 库的基础上,结合 libusb 库,提出了一种快速高效的驱动程序开发方法并且实现了数据的稳定快速传输,分模块实现的方法,将设备驱动程序划分为连接模块、传输模块,在底层直接调用 libusb 库和 CyAPI 库实现对设备的打开、关闭和数据读写. 传输控制程序实现数据处理,通过调用设备驱动函数,实现了对设备的各种控制和通信. 使用这种方法在极大的提高了开发效率的同时保证了传输速率,满足了 USB 设备工作的需要.

2 总体方案设计

2.1 工具和芯片选择

为了简化驱动程序的开发,我们借助使用 libusb 库和 CyAPI 库简化了开发流程,通过调用这些工具库中的 API 函数和结构体直接完成对 USB 设备的控制和数据采集. USB 控制芯片选择 Cypress 公司出品的 CYUSB3014 芯片^[6-7].

2.2 程序方案设计

2.2.1 驱动程序方案设计

本文设备驱动程序的开发在 Cyress 官方提供

的驱动程序框架基础上,在 Linux 操作系统的环境下利用 QT 编程工具,使用 C++ 编程语言,借助调用的 CyAPI 库和 libusb 库完成实现了对设备的初始化和数据传输功能^[8].

按照分模块实现的思想,将设备驱动程序按照实现的功能将驱动程序封装成划分为连接模块、传输模块,各模块提供若干接口函数供后续读写控制程序,包括未来可能需要的其他相关功能程序,来调用实现具体功能. 其中连接模块是 USB 工作和实现功能的基础,实现的功能包括对 USB 设备的初始化和使用完后对设备的释放. 传输模块是 USB 传输功能实现的主体,实现的功能是在 USB 设备完成初始化之后的数据写入和读出,数据的写入操作根据传输控制程序的需要,按照写入的内容的不同又分为固件程序写入和数据写入. 每个模块内部通过多个函数分别实现各项子功能,为每个功能设计一个接口函数供传

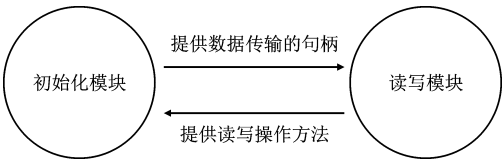


图 2 驱动程序模块关系图

输控制程序调用,不同的模块在底层都直接使用 CyAPI 库内对应的函数来直接对 USB 设备进行操作,各模块之间的相互关系如下图 2 所示. 对设备的直接操作使用 CyAPI 库内提供的函数实现,使用 libusb 内定义的结构体记录和存储设备信息. 传输控制程序通过各功能的接口函数变量将所要传输的命令和数据提供给设备驱动程序,驱动程序对这些信息进行处理,再调用 CyAPI 库完成数据的传输,同时接受通用数据采集返回的信息.

2.2.2 传输控制程序方案设计

仿照设备驱动程序的实现思路,依然按照模块化的设计思想来实现 USB 控制部分. 程序采用图形化界面,使用 QT 内的图形化界面实现,将界面根据所要实现的功能划分为多个区域,使用 QT 工具箱内提供了各种控件实现功能按键和数据显示等功能. 程序根据功能细分为初始化模块、读写模块,其中根据需要,有把读写模块划分为固件程序的写功能和数据文件的读写功能. 其中为了提高读写效率,设备的读写操作使用多线程实现. 对 USB 设备的操作直接通过设备驱动程序完成.

在工作过程中,读写控制程序通过与驱动程序

进行实时信息交互,能够及时获取设备的工作状态和数据采集设备所采集到的数据. 获取数据后,按照约定格式和消息值对数据进行处理并且及时向开发人员以及用户反馈. 驱动程序工作原理框图如图 3 所示.

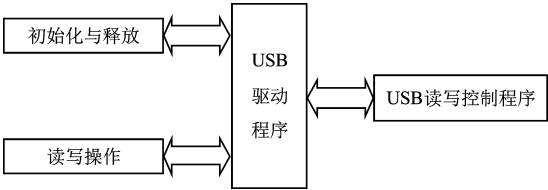


图 3 驱动程序工作原理框图

根据智能设备的工作和传输需要,但是对 USB 设备的操作都需要调用设备驱动程序,当需要使用 USB 设备时,每个程序直接调用动态链接库的接口函数实现对 USB 的各种操作.

3 驱动程序和传输控制程序实现

3.1 驱动程序各模块的实现

连接模块. 设备的初始化功能通过调用接口函数 Init_FX3USB() 实现,根据功能需要,初始化分

为两个部分,第一部分是设备查找阶段,完成 USB 库的打开和设备数量和基本简单信息的获取;第二部分是详细信息获取阶段,根据调用者在第一部分完成后选择的设备获取设备的详细信息. 下面分别介绍每个部分的实现:

第一部分. 提供接口函数 dev_num_get(). 在初始化的过程中,首先调用 QT 的监听函数对 USB 设备进行检测,调用 cyusb_open()函数获取当前设备数量,初始化 libusb 库并且构建 cpress 设备的专用信息结构体 cydev. 根据所检测到的 USB 设备数量,依次调用 cyusb_gethandle()、cyusb_getvendor()和 cyusb_getproduct()获取设备的句柄、VID 和 PID,将所需信息通过接口参数返回给调用程序.

第二部分. 设置接口函数 select_dev_info(). 分别调用 get_device_Detail()和 get_config_detail()获取设备信息和设备配置信息,函数内对设备信息的获取直接调用 cyusb_get_device_descriptor(),直接调用 cyusb_get_active_config_descriptor()获取设备配置信息,将获取的信息存入对应的结构体内. 整体的设备初始化过程时序图如图 4 所示.

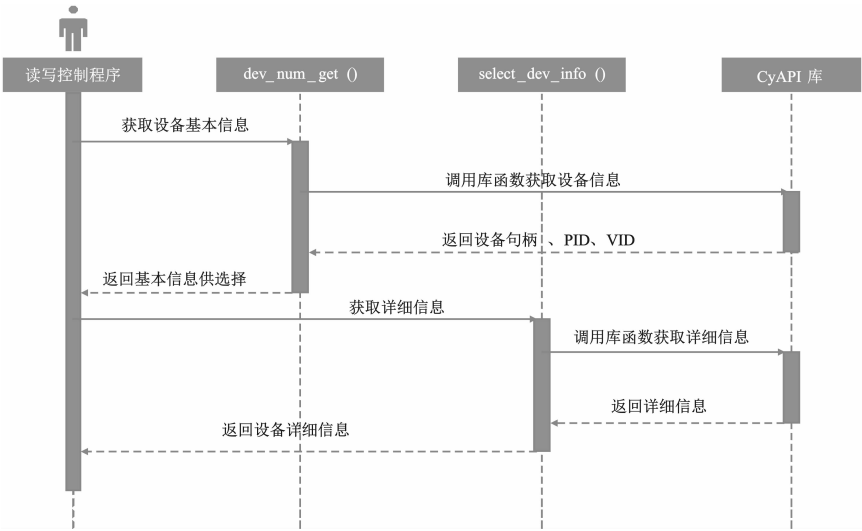


图 4 设备初始化时序图

传输模块. 数据的写入和读出使用块传输的方式,具体实现如下:

数据写入和读出. 设计接口函数 data_write()和 data_read(),可以根据需要从指定地址写入或者读出指定长度的数据. 数据发送和接受过程直接调用 CyAPI 内封装的 cyusb_bulk_transfer()根据句柄和选择的引脚完成对数据的读写操作. 在每进行完一次写操作之后都再进行一次读操作,并将读出

的数据和返回的消息值返回给调用程序.

在每个功能的运行过程中都设置了验证,对程序运行的结果进行检查,保证数据读写的正确性,提高了程序的容错能力. 同时驱动程序具有可扩展性,可以根据后期的开发需要进行重新的定制与修改,可以满足多样化的需求.

3.2 传输控制程序的实现

初始化模块. 在程序初始化的过程中自动执行

dev_num_get()函数获取当前已经接入主机的 USB 设备信息,并将信息在界面上显示,在用户点击界面选中所需要的设备后,根据用户的选择执行 select_dev_info()获取设备的各项信息,将设备信息存入界面信息显示区域的控件所关联的变量中,更新相关空间信息显示在界面上.

读写模块,读写模块分为固件写操作和数据写操作,对固件程序文件和数据文件采用文件操作的方式打开,对打开的过程要进行检查.使用多线程进行数据读写,分别设计发送和接收线程,每进行一次写操作将写入的数据在界面上实时显示,并接收一次返回的数据,根据返回的消息值确认数据是否写入成功.对设备驱动程序返回的错误信息值存入对应控件的字符串,最终在界面上显示.

4 设备驱动程序功能验证

4.1 验证方案设计

为了验证驱动和传输控制程序的效率和稳定性,取已经安装通用数据采集设备的视频设备,测试电脑安装 Linux 系统并且搭建程序运行环境,运行使用本文方法开发的 USB 传输控制程序.测试用例以及测试结果如下:

(1)设备识别功能测试:在 USB 设备插入和拔出的状态下分别对设备进行初始化操作,查看 USB 设备在插入状态下是否可以正常实现初始化操作,在拔出状态下,检查程序容错能力,是否可以正确的返回错误信息值.

(2)读写功能:接入 USB 设备,运行传输控制程序,选择固件程序文件进行写入操作,查看是否写入成功.固件写入成功后选择多个数据文件分别进行写入操作,检查数据传输是否正常并测试传输速率.进行多次试验将最后测试的速率汇总,计算传输速率的平均值.

4.2 验证结果

当设备接入后,初始化过程能够顺利实现.设备断开的状态下,能够准确接收到预先设计好的的错误信息.固件程序和文件数据的传输都能平稳、正常进行,经测试传输速率在 100MB/S 左右,完全满足数据传输的需要.

4.3 验证结论

这种设备驱动程序的开发方法完全可以满足视

频采集设备中的通用数据采集设备和主机之间进行实时通信和数据传输需要,能够保证数据的高速、准确传输;驱动程序的数据传输具有良好的稳定性;对设备的控制准确高效,能够灵活的初始化和释放设备,对驱动程序的调用和使用方便快捷.

5 结束语

本文介绍了基于 Linux 操作系统的 USB 驱动程序开发方法和传输功能的实现,充分的利用了和发挥了 CyAPI 库和 libusb 库给 USB 设备驱动开发带来的便利.使用本文设备驱动程序的 USB 读写控制程序,具有良好的传输效率和稳定性,有效的保证了传输的实时性.结果证明这是一种完全可以满足 USB 设备高速和实时传输的驱动程序开发方法.

参考文献:

- [1] 赵静,马尚昌. 基于 CY7C68013 的 USB2.0 通信接口的设计[J]. 微型机与应用,2015,22(10):61-63.
- [2] 张志伟,靳鸿,杨少博. 高速大容量记录仪的 USB3.0 高速读数接口设计[J]. 电子技术应用,2016. 42(12): 54-57.
- [3] 张颖,易金花,张晓玉,等. 基于嵌入式 Linux 的上肢康复机器人用户系统研究[J]. 电子技术应用,2014. 40(5):14-17.
- [4] 兀颖. 基于 USB3.0 的数据采集系统设计[D]. 西安:中国科学院西安光学精密机械研究所,2013.
- [5] 汤勇,胡志东. 基于 CYUSB3014 USB3.0 总线开发技术[J]. 电子世界,2013(23):12-13.
- [6] 陈松. 基于 USB3.0 的高速数据传输系统的研究与设计[D]. 成都:电子科技大学,2014.
- [7] 曾希强. Linux 下基于 libusb 的 USB 设备驱动程序设计与实现[J]. 信息与电脑,2009(7):101-103.
- [8] 刘治国,陈新华. 基于 Linux 和 Qt 的智能家居系统的设计与实现[J]. 电子技术应用,2012. 38(4):23-26.
- [9] 王继刚,方芳,张华强. 嵌入式操作系统异常处理框架设计与实现[J]. 电子技术应用,2017. 43(5):60-63.

作者简介:

万玛宁 女,(1981-),硕士,讲师.研究方向为通讯总线、高可靠嵌入式系统. E-mail:wanmaning@cnu.edu.cn.

侯志化 男,(1991-),硕士研究生.研究方向为高可靠嵌入式系统.