分 类 号_____ 学号_____M201672905_____

学校代码__10487__ 密级_____

華中科技大學

硕 士 学 位 论 文

基于 Vxworks 的调试通道的设计与实现

学位申请人：郑松

学 科 专 业：计算机应用技术

指 导 教 师：张杰 讲师

答 辩 日 期：2018 年 3 月 26 日

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master

A design and implementation of debug channel based on Vxworks.

Student : Joe

Major : Computer Applications Technology

Supervisor : Instructor JieZhang

Huazhong University of Science & Technology

Wuhan 430074, P. R. China

March 26, 2018

独创性声明

本人声明所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除文中已标明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于　保密 □，在 ____ 年解密后适用本授权书。

　　不保密 □。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

摘 要

数据传输是现代通讯过程中的一个重要环节,在数据的传输过程中,不仅仅要求数据传输的准确率要高,而且要求速度快,连接方便。传统的 RS232 串口通讯和并口通讯都存在传输速度低,扩展性性差、安装麻烦等缺点,而基于 USB 接口的数据传输系统能够较好的解决这些问题。目前 USB 接口以其传输速率高、即插即用、支持热插拔等优点,逐步成为 PC 机的标准接口。

本文中的数据传输系统采用了 USB 接口进行上位机与下位机之间的数据通讯。下位机采用的是 VxWorks 实时操作系统。

关键词： 实时操作系统,设备驱动,USB 口转串口,VxWorks,CP2103

Abstract

This is a LaTeX template example file. This template is used in written thesis for Huazhong Univ. of Sci. & Tech.

This template is published under LPPL v1.3 License.

Key words: LaTeX, Huazhong Univ. of Sci. & Tech., Thesis, Template

目 录

一 绪论

VxWorks 操作系统是美国 WindRiver 公司于 1983 年推出的一种嵌入式实时操作系统 (RTOS), 是一个运行在目标机上的高性能、可裁剪的实时操作系统, 是一个专门为嵌入式实时系统设计开发的操作系统, 其具有良好的持续发展能力、高性能的内核以及友好的用户开发环境, 为开发人员提供了高效的实时多任务调度、中断管理、实时的系统资源以及实时的任务间通信。在嵌入式实时操作系统领域当中占有一席之地。VxWorks 支持 X86、PowerPC、ARM 等众多主流的处理器, 且在各种 CPU 平台上提供了统一的编程接口和一致的运行环境。在军事、航空航天、工业控制、通信等高精尖以及实时性要求极高的领域当中, 有着更加广泛而深入的应用。应用实例包括 1997 年火星表面登入的火星探测器、爱国者导弹、飞机导航、F-16、FA-18 战斗机等。自从对我国的销售解禁之后, VxWorks 也大量的应用于我国的军事、国防工业当中, 通常在进行 VxWorks 应用程序的开发或者是将 Linux 下的应用程序移植到

1.1 课题背景以及意义

1.2 国内外概况

1.3 论文的主要内容和组织结构

二 实时操作系统 VxWorks

2.1 概述

2.1.1 实时操作系统

实时操作系统 (Real Time Operation System, 简称 RTOS) 是整个实时系统的核心。POSIX1003.1 标准为 RTOS 下了一个简单的定义: RTOS 是能够在有限的响应时间内为应用提供所要求级别服务的操作系统[?]。当外界事件或者是数据产生时, RTOS 需要快速的进行处理, 并且处理的结果又能够在规定的时间之内来控制生产过程或者对处理系统做出快速的响应, 调度一切可利用的资源来完成实时任务。实时系统按照实时的效果可以分为软实时和硬实时, 硬实时要求在规定的时间内必须完成操作, 这是通过操作的在设计的时候就得到保证的; 软实时只需要按照任务的优先级, 尽可能快的完成任务即可。

一个实时操作系统的特征通常包括以下几点:

- **高精度计时系统**
  计时精度是影响实时性的一个重要因素, 在实时系统当中, 经常需要精确确定实时地操作某个设备或执行某个任务, 或精确的计算一个时间函数。这些不仅仅依赖于一些硬件提供的时钟精度, 也依赖于实时操作系统的高精度计时功能。
- **多级中断机制**
  中断是实时操作系统当中的一个关键设施, 是用于通知系统发生外部事件的常用机制。一个实时操作系统通常需要处理多种外部信息或事件, 但处理的紧迫程度有轻重缓急之分。有的必须立即作出反应, 有的则可以延后处理。因此, 需要建立多级中断嵌套处理机制, 以确保对紧迫程度较高的实时事件进行及时响应和处理。
- **实时调度机制**
  实时操作系统不仅要及时响应实时事件中断, 同时也要及时调度运行实时任务。但是, 处理机调度并不能随心所欲的进行, 因为涉及到两个进程之间的切换, 只能在确保“安全切换”的时间点上进行, 实时调度机制包括两个方面, 一是在调度策略和算法上保证优先调度实时任务; 二是建立更多“安全切换”时间点, 保证及时调度实时任务。

内核作为操作系统的核心, 负责控制这计算机上的所有硬件和软件资源, 在必要的时候给应用程序分配硬件资源, 并执行相应的操作命令。内核的主要功能为以下四个:

- 系统内存管理

- 软件程序管理
- 硬件设备管理
- 文件和网络系统管理

本次所需完成的调试通道正是基于一个目前业界有名的实时操作系统 VxWorks。

2.1.2 VxWorks 简介

VxWorks 操作系统是美国 Wind River System 公司于 1983 年推出的一个运行在目标机上的高性能、可裁剪的实时操作系统 (RTOS), 该系统专门为嵌入式实时系统领域而设计开发, 其具有良好的持续发展能力、高性能的内核以及友好的用户开发环境, 为开发人员提供了高效的实时多任务调度、中断管理、实时的系统资源以及实时的任务间通信。并且拥有多达 1800 个功能强大的应用程序接口[?]。

VxWorks 采用微内核设计, 支持多种硬件环境, 包括 X86、PowerPC、ARM 等众多主流的处理器, 同时还支持 RISC、DSP 技术, 且在各种 CPU 平台上提供了统一的编程接口和一致的运行环境。VxWorks 凭借着其优异的性能在军事、航空航天、工业控制、通信等高精尖以及实时性要求极高的领域当中, 有着更加广泛而深入的应用。应用实例包括火星探测器、爱国者导弹、飞机导航、F-16、FA-18 战斗机等[?]。自从对我国的销售解禁之后, VxWorks 也大量的应用于我国的军事、国防工业当中。

VxWorks 操作系统由 400 多个相对独立的短小精炼的目标模块组成, 用户可以根据自己的实际需求选择适当的模板来裁剪和配置系统, 这样可以有效的保证系统的安全性和可靠性。系统的连接器可以按照应用的需要自动链接一些目标模块。这样, 通过目标模块之间的按需组合, 可以得到许多满足功能需求的应用。此外, VxWorks 支持广泛的工业标准, 如 POSIX1003.1b 实时扩展、ANSIC(浮点支持) 和 TCP/IP 网络协议等。这种广泛的协议支持在主机和目标机之间提供了无缝的工作环境, 任务可以通过网络箱其他系统的主机存取文件, 即远程文件存取, 也支持远程的过程调用。这些标准也促进了多种不同产品之间的互通性, 提升了可移植性。由于其高度的灵活性, 用户能够很容易的对该操作系统进行重新定制或作适当的开发, 以满足自己的实际应用。

2.2 微内核 wind

2.2.1 wind 的多任务机制

现代实时系统基于多任务和任务间通信的互补概念。多任务环境允许将实时应用程序构建为一组独立任务, 每个任务都有自己的执行线程和一组系统资源。任务间通信设施允许这些任务同步并进行通信以协调其活动。在 VxWorks 中, 任务间的通信工具包括快速信号量、消息队列、管道、套接字。

2.2.2 wind 的任务调度

2.2.3 wind 线程同步

2.3 集成开发环境 Tornado

2.3.1 tadj

2.4 VxWorsks 上的 USB 协议栈

USB(Universial Serial Bus, 通用串行总线) 是这十几年来应用在 PC 领域的最新型的接口技术, 出现的契机是为了为了解决日益增加的 PC 外设与有限的主板插槽之间的矛盾, 其实现的原理是由一些 PC 大厂商 (Microsoft、Intel 等) 定制出来的, 自从 1995 年在 Comdex 上展出以来至今已广泛地被各个 PC 厂家所支持。目前已经在各类外部设备中都广泛的采用 USB 接口。USB 接口标准目前有三种: USB1.1, USB2.0 和 USB3.0。USB 接口应用如此的广泛是由其独特和实用的特性决定的。USB 的体系结构如图**??**所示

图 2-1 USB-structure

2.4.1 USB 数据传输的优势

USB 的主要有优点有:

1. 使用方便, 支持热插拔: 连接外设不必再打开主机箱, 而且方便携带, 允许在任何时候 USB 外设的热插拔, 而且不必关闭主机的电源; USB 外设没有需要用户选择的设置; 例如端口地址和中断请求线; 自动配置外设, 当用户连接 USB 外设到一个正在运行的系统时, 系统会自动的检测外设, 载入合适的驱动软件 (如果有的话), 并将其初始化; 以上的特点使得 USB 外设符合便携易用的潮流。
2. 速度快: 目前设备上使用大多数是 USB 2.0 的接口, 最新生产的设备都配备了 USB3.0 的接口。USB2.0 接口的理论速度可以达到 480Mbps(即 60MB/s), 并且可以向下兼容 USB1.1。USB3.0 接口的理论速度可以达到 5.0Gbps(即 640MB/s), 并提供 USB2.0 的兼容。
3. 连接灵活: 一个 USB 主控制器理论上可以连接 127 个设备

2.4.2 VxWorks 上的 USB 驱动程序结构

2.5 CP2012 模块

三　驱动程序的设计和实现

3.1　设备驱动概述

通常为了实现与应用程序的平台无关性，操作系统都会为应用程序提供一套标准的接口，VxWorks 也不例外，这样就可以通过调整底层驱动或者是接近驱动那部分的操作系统中间层来提高应用层开发的效率，避免重复编码。在我们的通用操作系统(如 Mac OS、Linux、Windows) 当中，通常会将这套应用层的接口标准从操作系统中独立出来，专门以标准库的形式存在，这样可以屏蔽操作系统之间的差异，增强应用程序的平台无关性。

VxWorks 中也对应用层提供了一套标准的文件操作接口，实际上与 GPOS 提供接口类似，我们将其称作为标准 I/O 库，VxWorks 下由 ioLib.c 文件提供。ioLib.c 文件提供如下标准接口函数：creat()、open()、unlink()、remove()、close()、rename()、read()、write()、ioctl()、lseek()、readv()、writev() 等[?]，VxWorks 与通用操作系统有很大的一个不同点是：VxWorks 不区分用户态和内核态，用户层可以直接对内核函数进行调用，而无需使用陷阱指令之类的机制，以及存在使用权限上的限制。因此 VxWorks 提供给应用层的接口无需通过外围库的方式，而是直接以内核文件的形式提供。

设备驱动时直接控制设备操作的那部分程序，也是设备上层的一个软件接口。设备驱动程序的功能完成软件层对硬件的访问，实际上从软件工程的角度来说就是介于软件和硬件之间实现软件层标准接口的程序。软件层访问硬件必须要通过调用驱动程序。所以驱动程序不能自动执行，只能被系统或者程序调用。

应用程序必须要通过驱动程序才能够与硬件设备进行通信，而驱动程序的编写与操作系统的关系密不可分。设备驱动程序在操作系统中如何存在、如何与操作系统的其它部分相联系、如何与操作系统的其他部分相联系、如何为用户提供服务都是操作系统的设计人员在设计操作系统时制定的，系统已经为驱动程序制定好了一个框架，无论驱动程序的开发人员以何种方式控制设备，他们所开发的驱动程序都是以预先设计好的方式存在、与操作系统其他部分相联系和为用户提供服务的。将这种由操作系统的设计人员指定的设备驱动程序结构定义为驱动程序的外部结构，而由于驱动开发人员在开发设备驱动时采用的具体策略不同导致的不同的驱动程序结构称为驱动程序的内部结构。驱动程序的外部结构决定了操作系统的 I/O 体系结构，驱动程序的内部结构决定了不同的设备驱动方式。

在 VxWorks 系统中，在控制器权转到设备驱动程序之前，用户的请求进行尽可能少的处理。VxWorks I/O 系统的角色更像是一个转接开关，负责将用户请求转接到合适的驱动例程上。每一个驱动都能够处理原始的用户请求，到最合适它的设备上。另外，驱动程序开发者也可以利用高级别的库例程来实现基于字符设备或者块设备的标准协议。因此，VxWorks 的 I/O 系统具有两方面的优点：一方面使用尽可能少的使用

驱动相关代码就可以为绝大多数设备写成标准的驱动程序,另一方面驱动程序开发者可以在合适的地方使用非标准的程序自主的处理用户请求。

3.1.1 设备驱动的功能以及组成部分

3.1.1.1 设备驱动程序结构

VxWorks 的 I/O 框架由 ioLib.c 文件提供,但 ioLib.c 文件提供的函数仅仅是一个最上层的接口,并不能完成具体的用户请求,而是将请求进一步向其他内核模块进行传递,位于 ioLib.c 模块之下的模块就是 iosLib.c。我们将 ioLib.c 文件称为上层接口子系统,将 iosLib.c 文件称为 I/O 子系统,注意二者的区别。上层接口子系统直接对用户层可见,而 I/O 子系统则一般不可见(当然用户也可以直接调用 iosLib.c 中定义的函数,但一般需要做更多的封装,且违背了内核提供的服务层次),其作为上层接口子系统与下层驱动系统的中间层而存在。

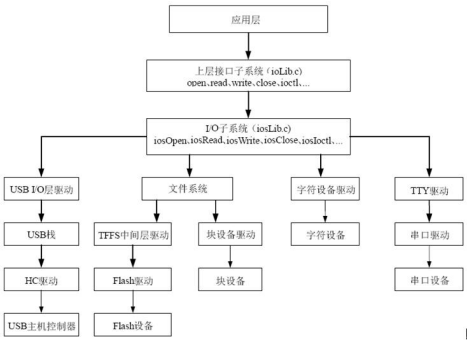


图 3-1 VxWorks 驱动内核层次结构

I/O 子系统在整个驱动层次中起着十分重要的作用,其对下管理着各种类型的设备驱动。换句话说,各种类型(包括网络设备)的设备都必须向 I/O 子系统进行注册方可被内核访问。所以在 I/O 子系统这一层次,内核维护着三个十分关键的数组用以对设备所属驱动、设备本身以及当前系统文件句柄进行管理。

需要指出的是,VxWorks 文件系统在内核驱动层次中实际上是作为块设备驱动层次中的一个中间层而存在的,其向 I/O 子系统进行注册,而将底层块设备驱动置于自身的管理之下以提高数据访问的效率。在这些文件系统中,dosFs 和 rawFs 是最常用的两种文件系统类型,在 VxWorks 早期版本就包含对这两种文件系统的支持。

3.1.1.2 驱动程序管理表

3.1.1.3 驱动程序管理步骤

3.2 USB 转串口设备驱动程序的实现

3.2.1 特定需求的单设备支持

3.2.2 通用的多设备支持

3.3 字体

普通**粗体**斜体

黑体楷体仿宋

3.4 公式

单个公式,公式引用:**??**。

$$c^2 = a^2 + b^2 \tag{3.1}$$

多个公式,公式引用:**??**,**??**。

$$F = ma \tag{3.2a}$$

$$E = mc^2 \tag{3.2b}$$

3.5 罗列环境

1. 第一层
2. 第一层

   2.1 第二层

   2.2 第二层

   a) 第三层

   b) 第三层

解释环境 解释内容

3.6 代码环境

```
import os

def main():
    '''
    doc here
    '''
    print 'hello, world' # Abc
    print 'hello, 中文' # 中文
```

3.7 定律证明环境

定义 3.1. 这是一个定义。

命题 3.1. 这是一个命题。

公理 3.1. 这是一个公理。

引理 3.1. 这是一个引理。

定理 3.1. 这是一个定理。

证明. 这是一个证明。 □

3.8 算法环境

算法 3.1: How to write algorithms

Data: this text

Result: how to write algorithm with LaTeX2e

```
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

3.9 表格

表格见??。

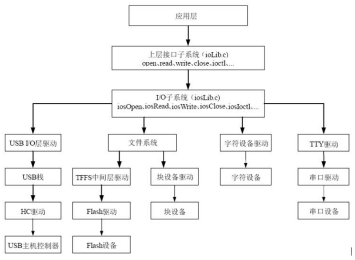
表 3.1　一个表格

a	b
c	d

3.10　图片

图片见**??**。图片格式支持 eps, png, pdf 等。多个图片见**??**, 分开引用: **??**, **??**。

图 3-2　hust-title



(a) VxWorks-driver-structure　　(b) 图片 2

图 3-3　多个图片

3.11　参考文献示例

这是一篇中文参考文献[?]; 这是一篇英文参考文献[?]; 同时引用[?,?]。

3.12　\autoref 测试

公式 ??
脚注 ??
项 ??,??,??
图 ??
表 ??
附录 ??
章 ??
小节 ??,??,??
算法 ??,??

证明环境 ??,??,??,??,??,??

致 谢

致谢正文。

参考文献

[1] Renard K G, Nichols M H, Woolhiser D A, et al. 1003.1-2008 - IEEE Standard for Information Technology- Portable Operating System Interface (POSIX) Base Specifications, Issue 7. 2008, c1-3826.

[2] 孔祥营. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado. 中国电力出版社, 2002.

[3]〔美〕WindRiver. VxWorks BSP 开发人员指南(VxWorks 开发人员指南丛书). 清华大学出版社, 2004.

[4] 徐媛媛. 嵌入式实时操作系统的设备驱动: [PhD Dissertation]. 华中科技大学, 2003.

附录 A　攻读学位期间发表的学术论文

[1] 论文 1
[2] 论文 2

附录B　这是一个附录

附录正文。