

西安科技大学

硕士学位论文

基于ARM与 μ C/OS-II 的高端仪表平台的研究

姓名：安洁

申请学位级别：硕士

专业：控制理论与控制工程

指导教师：黄向东

20090430

论文题目：基于 ARM 与 μ C/OS-II 的高端仪表平台的研究

专 业：控制理论与控制工程

硕 士 生：安 洁

(签名) 安洁

指导教师：黄向东

(签名) 黄向东

摘 要

针对仪器仪表向高端产品的发展趋势，课题提出并设计实现了一种基于嵌入式 μ C/OS-II 操作系统和 ARM7 微处理器为核心的控制平台，使仪表的使用更加方便、智能。系统融合了嵌入式系统、USB 通信、LAN 通信、显示等多项快速发展的技术，通过 USB 模块和 LAN 网络的数据传输，实现了高端仪表与外部设备的通信，整个平台具有高速、实时传输数据等特性，能够广泛地应用于多种行业的现场测量中。

硬件方面，课题采用具有 ARM7TDMI 核的 LPC2220 微处理器作为系统的控制平台，并结合应用设计出了显示模块、USB 通信模块、LAN 通信模块。控制平台通过 USB 通信模块和 LAN 通信模块，建立与外部设备的数据处理通道，将与 SPI 接口连接的仪表数据进行传输处理。USB 接口电路采用了 Cypress 公司的 CY7C68001 芯片，LAN 通信模块则采用了 CIRRUSLOGIC 的以太网控制器 CS8900 实现底层驱动。

软件方面，首先将 μ C/OS-II 操作系统移植到 ARM7 上，并在嵌入式 μ C/OS-II 环境下编写了各硬件模块的驱动程序。在驱动程序的基础上设计了 VFD 显示程序、USB 通信和网络通信等应用模块，验证了数据处理平台具有的各项功能。网络通信模块中，WEB SERVER 在控制平台实现，在上位 PC 上输入服务器的固定 IP 地址，实现控制命令的发送、数据包的接收等功能。

经测试，系统运行正常，较好的实现了各项设计目标，从而证明了本文的方法是可行的。本系统为高端仪表的数据处理提供了一个有效的解决方案，具有良好的应用前景。

关 键 词：数据处理；嵌入式系统；LPC2220；嵌入式 μ C/OS-II；通信

研究类型：应用研究

**Subject : Research in Advanced Instrumentation Platform Based on
ARM and μ C/OS-II**

Specialty : Control Theory and Control Engineering

Name : An Jie

(Signature) An Jie

Instructor : Huang Xiangdong

(Signature) Huang Xiangdong

ABSTRACT

For advanced products development trend of instrumentation, the paper presents and realizes a kind of control platform based on μ C/OS-II embedded operating system and ARM7 microprocessor. And make the using of instrumentation more convenient and intelligent. The system integrates a variety of technologies which are developing very soon, such as embedded system, USB communications protocol, LAN communications protocol and show. The system achieves to transmit data through USB and LAN communication module. The system has the advantage of high-speed and real-time. So it can be widely applied to various fields.

As to the hardware, the embedded control platform is established based on LPC2220. And design the show module, USB communications module, LAN communications module with application. The control platform connects external equipment with USB module and LAN module for transmitting the data of instrumentation which connects the control platform with SPI interface. USB interface circuit adopts CY7C68001 of Cypress. LAN communications module use Ethernet control unit CS8900 of CIRRUSLOGIC for realizing bottom driver.

As to the software, μ C/OS-II operating system is transplanted to the processor, and the driver of every module has been included. The function of displaying instrumentation data with SPI 、USB communication and LAN communication has been achieved too. WEB SERVER achieved by microprocessor in network communication module. It realizes the function of sending the control command and receiving data packet.

The test result of this supervisory system indicates that it works normally and reaches the design goals. This proves that the method is feasible, and the system can be used broadly for advanced instrumentation.

Keywords : Data Processing Embedded System LPC2220 Embedded μ C/OS-II
Communication

Thesis : Application Research

西安科技大学

学位论文独创性说明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中加以标注和致谢的地方外，论文中不包含其他人或集体已经公开发表或撰写过的研究成果，也不包含为获得西安科技大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

学位论文作者签名：安洁 日期：2009年5月5日

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西安科技大学。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。本人允许论文被查阅和借阅。学校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。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西安科技大学。

保密论文待解密后适用本声明。

学位论文作者签名：安洁

指导教师签名：黄向东

2009年5月5日

1 绪论

1.1 课题研究背景及意义

1.1.1 课题研究背景

测量技术是信息技术的关键和基础，信息社会起带动作用的科学技术首推信息技术。信息技术包括测量技术、计算机技术和通讯技术。测量技术与信息技术的关系见图 1.1。仪器仪表则是认识、探索和改造物质世界必须的测控工具。钱学森同志在论述信息技术的内在关系时，特别强调测量技术的基础性(信息之源)及关键性(科技攻关的第一位关键)。

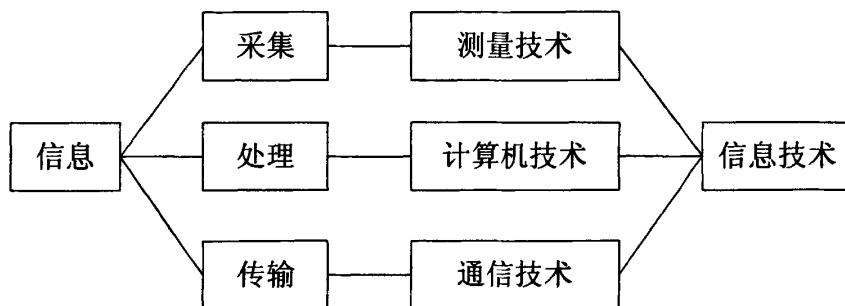


图 1.1 测量技术与信息技术关系图

随着计算机、网络和通信技术的发展，21世纪人类进入了后PC 时代。在此阶段，人们开始考虑如何将各类仪器设备变得更加智能化、数字化，从而使其轻巧便利、易于控制或具有某些特定的功能。^[1]为了实现这些新要求，嵌入式技术(embedded technology)提供了一种灵活、高效和高性价比的解决方案。后PC时代的智能仪器设备已逐渐模糊了硬件与软件的界限，SoC (system on chip) 系统的发展就是这种软硬件无缝结合趋势的证明。

嵌入式系统是一种用于控制、监测或协助特定装置和设备正常运转的计算机。嵌入式系统是计算机技术、通信技术、半导体技术、微电子技术、语音图像数据传输技术、甚至传感器等先进技术和具体应用对象相结合后的更新换代产品。它以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

嵌入式系统是现代科学的多学科互相融合的以应用技术产品为核心、以计算机技术为基础、以通信技术为载体、以产品为对象、引入各类传感器、进入Internet 网络技术

的连接而适应应用环境的产品。

随着计算机应用的不断发展,各种新外设不断涌现,适配器的种类和数量也不断增加,对PC机有限的扩展能力提出了严峻的挑战。而对于采用并行口或串行口等不需要适配器接入的外设,也存在不支持热插拔、传输速率较低的问题。由此,在USB协议提出之后,USB总线接口正逐步取代原有的各种接口。事实上几乎所有的PC外设都可以移植带USB接口上来。

1.1.2 课题的提出

近年来,现代仪器仪表产品向着计算机化、网络化、智能化、多功能化的方向发展,跨学科的综合设计、高精尖的制造技术使其能更高速、更灵敏、更可靠、更简捷地获取被分析、检测、控制对象的全方位信息。嵌入式系统是未来真正实现光、机、电、算(计算机)一体化、自动化的结构,走向更名副其实的智能系统(带有自诊断、自控、自调、自行判断决策等高智能功能)的基本保证。

针对仪器仪表向高端产品的发展趋势,本系统进行了模块化的设计,与高精度测量模块组成了一个具有先进控制、采集、传输的高端仪表。高端仪表组织结构如图1.2所示。

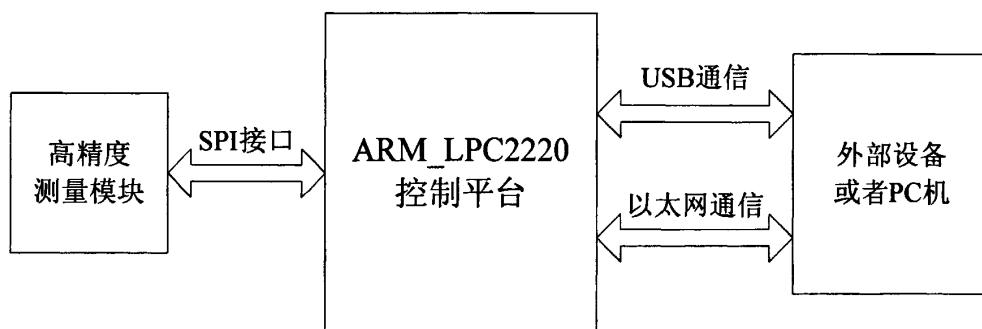


图1.2 平台组织结构图

本系统将高精度测量模块采集的数据进行实时显示和实时传输到外部设备或者PC机,可实现外部设备或者PC机对仪表功能的扩展,为各种现场条件下提供一种智能、高效的操作系统。

1.1.3 课题的研究意义

本课提出并初步设计了基于嵌入式 $\mu\text{C}/\text{OS-II}$ 操作系统的ARM7处理器控制平台。本系统可以应用在高端仪表的数据处理方面,随着仪表的发展,会拥有一个开阔的应用前景。控制平台具有小型化、低成本、低功耗、可扩展性强等特点,具有一定的工程应用和参考价值。

1.2 国内外现状及发展趋势

1.2.1 发展历史

西方国家将仪器仪表纳入信息技术产业并采取优先发展的政策,它把测量用工业仪表、电测仪器、实验室分析仪器,以及相关软件归属于信息技术生产行业。并把发展信息技术和信息产业摆在优先地位,同时加大了这方面的战略性投资。美国90年代末期对信息技术的投资占其设备投资的45%,使10年前还很薄弱的美国信息产业现在已处于领先地位,成为美国国内销售和出口最大的产业部门,增长速度是国民经济增速的两倍,还大大提高了劳动生产率。对于仪器仪表产业美国商务部分析说:“仪器仪表总产值仅占工业总产值的4%,但其对国民经济生产总值(GNP)的影响却达到66%”,起到了四两拨千斤的作用。

我国仪器仪表科学技术与国外的差距很大,并不适应国内相关技术的要求,而且还有继续增大的趋势。它已成为国民经济制造业的主要瓶颈之一。我国仪器仪表的地位愈来愈重要,社会许多深层次的问题和为可持续发展服务领域无不要求仪器仪表提供准确的信息。例如配合打假活动的产品鉴别仪器、系统;自然灾害的监测、预警、调度系统;公安侦破测试仪器;环境自动监测系统;交通指挥调度系统等。各行各业,随着对仪器仪表的要求也趋向于要求快速、自动、准确地测控,投资费用在增加,并占设备投入的相当比例。例如运载火箭的试制费用一半用于购置仪器仪表;高科技的奥运工程也要求10%投入自动测试监控的装备。企业的技术改造采用自动测控技术装备,可以增加新的生产能力、降低成本。其对科技、经济社会和国防带动作用是显而易见的。因此在发展战略上:

(1) 应将仪器仪表正式纳入信息技术产业,采取同样的发展政策;

(2) 把实现快速、自动、准确测量的“自动测控系统”及“虚拟仪器”作为主攻目标,以系统带产品、元器件、应用软件、新技术、新材料、新工艺的发展。以实现“精确军事打击”、“精确工业”、“精确农业”、“精确侦破”、“精确灾害预警”,提高科技攻关的能力等目标,为的是最终提高社会主义强国的综合国力。

(3) 尽量拓宽仪器仪表的应用领域,在各行各业发挥其“四两拨千斤”的潜能,也是仪器仪表本身科技发展方向与市场需求的源泉。

(4) 以信息化带动工业化,测控仪表是发挥后发优势的有力和必备的手段,可加速实现社会生产力的跨越式发展,提高国际竞争力。

(5) 为各个领域做好服务,把系统应用设计放在首要位置,提高实施系统集成“交钥匙”的测控自动化项目的能力。

(6) 组建国家仪器仪表科技创新体系,以指导和引导全社会的仪器仪表科技创新工

作。发挥社会主义优越性,实现组织联合,以少胜多,减少重复投入。同时贯彻双赢原则,成果共享。

嵌入式系统出现于20世纪70年代末,尤其近年来,计算机技术、微电子技术,通信网络技术一体化趋势日益显著,嵌入式系统的研究已成为当今研究的一个热点。

第一阶段是以单片微型计算机技术为核心,具有与监测、伺服、指示设备相配合的功能。这类系统大部分应用于一些专业性强的工业控制系统中,一般没有操作系统的支持,通过C语言与汇编语言编程对系统进行直接控制。这一阶段系统的主要特点是:系统结构和功能相对单一,处理效率较低,存储容量有限,几乎没有用户接口。

第二阶段是以嵌入式CPU、以简单操作系统为核心的嵌入式系统。主要特点是:CPU种类繁多,通用性比较弱;系统开销小,效率高;操作系统达到一定的兼容性和扩展性;应用软件较专业化。

第三阶段是以嵌入式操作系统为标志的嵌入式系统。主要特点是:嵌入式操作系统能运行于各种不同类型的微处理器上,兼容性好;操作系统内核小、效率高,并且具有高度的模块化和扩展性;具备文件和目录管理、多任务、网络支持、图形窗口以及用户界面等功能;具有大量的应用程序接口API,开发应用程序较简单;嵌入式应用软件丰富。

第四阶段是以Internet为标志的嵌入式系统。目前大多数嵌入式系统还孤立于Internet之外,但随着Internet的发展以及Internet技术与信息家电、工业控制技术结合日益密切,嵌入式设备与Internet的结合将代表嵌入式系统的未来。嵌入式系统应用将日益广泛。

1.2.2 嵌入式系统研究现状

目前根据其发展现状,嵌入式计算机可以分成下面几类^[2]:

(1) 嵌入式微处理器 (Embedded Microprocessor Unit, EMPU)

嵌入式微处理器的基础是通用计算机中的CPU。在应用中,将微处理器装配在专门设计的电路板上,只保留和嵌入式应用有关的母板功能,这样可以大幅度减小系统体积和功耗。为了满足嵌入式应用的特殊要求,嵌入式微处理器虽然在功能上和标准微处理器基本是一样的,但在工作温度、抗电磁干扰、可靠性等方面一般都做了各种增强。

(2) 嵌入式微控制器 (Microcontroller Unit, MCU)

嵌入式微控制器又称单片机。嵌入式微控制器一般以某一种微处理器内核为核心,芯片内部集成ROM/EPROM、RAM、总线、总线逻辑、定时P计数器、WatchDog、I/O、串行口、脉宽调制输出、APD、DPA、Flash RAM、E2PROM 等各种必要功能和外设。为适应不同的应用需求,一般一个系列的单片机具有多种衍生产品,每种衍生产品的处理器内核都是一样的,不同的是存储器和外设的配置及封装。这样可以使单片机最大限度地和应用需求相匹配,功能不多不少,从而减少功耗和成本。和嵌入式微处理器相比,微控制器的最大特点是单片化,体积大大减小,从而使功耗和成本下降、可靠性提高。

(3) 嵌入式DSP处理器 (Embedded Digital Signal Processor, EDSP)

DSP 处理器对系统结构和指令进行了特殊设计,使其适合于执行DSP 算法,编译效率较高,指令执行速度也较高。在数字滤波、FFT、谱分析等方面DSP 算法正在大量进入嵌入式领域,DSP 应用正从在通用单片机中以普通指令实现DSP 功能,过渡到采用嵌入式DSP 处理器。

(4) 嵌入式片上系统 (System On Chip)

随着EDI 的推广和VLSI 设计的普及化,及半导体工艺的迅速发展,在一个硅片上实现一个更为复杂的系统的时代已来临,这就是System On Chip(SOC)。各种通用处理器内核将作为SOC 设计公司的标准库,和许多其它嵌入式系统外设一样,成为VLSI 设计中一种标准的器件,用标准的VHDL 等语言描述,存储在器件库中。用户只需定义出其整个应用系统,仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的器件以外,整个嵌入式系统大部分均可集成到一块或几块芯片中去,应用系统电路板将变得很简洁,对于减小体积和功耗、提高可靠性非常有利。

1.2.3 嵌入式系统发展趋势

早在1990年之前,是系统通常是很简单的且具有很长的产品生命周期的自主设备。近些年来,嵌入式工业经历了巨大的变革:产品市场窗口现在预计翻番的周期狂热到6-9个月;全球重新定义市场的机会和膨胀的应用空间;互联现在是一个需求而不是辅助性的,包括用有线和刚刚显露头角的无限技术;基于电子的产品更复杂化;互联嵌入式系统产生新的依赖网络基础设施的应用;微处理器的处理能力按摩尔定律预计的速度在增加,该定律认为集成电路和晶体管个数每18个月翻一番。

嵌入式控制器的应用几乎无处不在:移动电话、家用电器、汽车—无不有它的踪影。嵌入控制器因其体积小、可靠性高、功能强、灵活方便等许多优点,其应用已深入到工业、农业、教育、国防、科研以及日常生活等各个领域,对各行各业的技术改造、产品更新换代、加速自动化化进程、提高生产率等方面起到了极其重要的推动作用。

嵌入式计算机在应用数量上远远超过了各种通用计算机,一台通用计算机的外部设备中就包含了5-10个嵌入式微处理器。在制造工业、过程控制、网络、通讯、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等方面均是嵌入式计算机的应用领域。

1.3 论文的主要内容

本课题所做的主要工作如下:

(1) 嵌入式控制平台的设计:以嵌入式 μ C/OS-II操作系统的ARM7处理器为核心,设计实现系统控制的硬件平台和软件平台。

(2)显示单元的设计:实现平台数据显示。

(3) USB通信模块设计：熟悉USB协议，完成所需的软硬件设计，实现上位PC与控制平台的通信。

(4) LAN通信模块设计：掌握TCP/IP协议，通过网线互联的方式实现上位PC与控制平台的通信。

(5) 软件系统的设计：以嵌入式 μ C/OS-II软件平台为基础，编写所需的驱动程序和应用程序，建立控制平台的软件系统，实现应用系统的各种功能。

(6) 系统的整合、调试和完善。

1.4 论文组织及章节安排

本文共分为 6 章：

第一章：简述了课题的研究背景及论文的主要内容。

第二章：介绍了嵌入式平台的硬件设计及 USB 协议。

第三章：简单介绍了嵌入式系统的软件平台。

第四章：阐述了 FLASH 型存储器的磨损算法。

第五章：详细介绍了系统在嵌入式平台上的应用开发设计。

第六章：总结了本文所作的工作，并指出了课题的不足之处以及后续研究的方向。

1.5 本章小结

本章首先介绍了嵌入式系统的发展现状，然后对课题的研究对象、应用领域及研究意义进行了说明，最后简要说明了本课题所实现的嵌入式平台的基本结构及论文的主要工作内容。

2 高端仪表平台的硬件设计

嵌入式系统开发包括硬件与软件的开发。从技术角度定义：嵌入式系统是以应用为中心、以计算机技术为基础、软硬件可剪裁、适应应用系统，对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。由于嵌入式系统具有微内核、系统精简、高实时性、专用性强等特点，因此非常适合应用于通信领域。有别于传统的控制系统，嵌入式系统由嵌入式微处理器/微控制器、存储器、输入输出和软件组成。这种操作系统和应用软件紧密结合在一起，正式嵌入式系统和基于 Windows 应用系统的主要差别所在，使其可靠性和实时性得到了重要保证^[3]。

2.1 系统硬件平台设计

本课题实现的是一个嵌入式开发平台，采用 ARM 核心板与接口板分离的结构，ARM 核心板包括 ARM7TDMI-S 处理器、2M NOR Flash 程序存储器、2 路 USB-Host 控制器，接口板主要包括 USB 接口、显示控制模块、网络控制模块、SD 卡接口、调试串口、UART、电源管理等 7 部分。采用这种接口易于对处理器进行升级，因为 ARM 核心板采用了 6 层板，而接口板采用 4 层板就可以，升级时只需要更换核心板就可以了。总体设计框图如图 2.1。

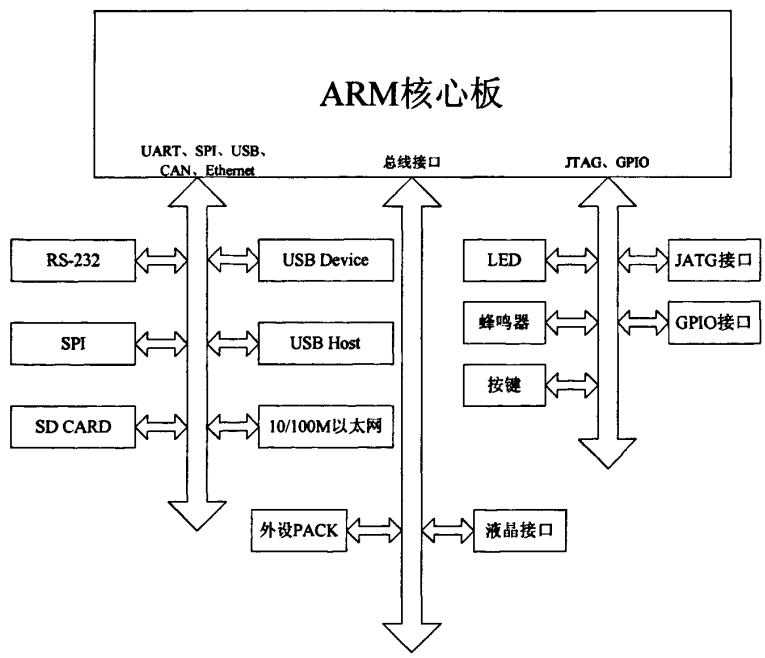


图 2.1 系统总体设计框图

2.2 ARM 概述

ARM 是 Advanced RISC Machines 的缩写，它是一家微处理器行业的知名企业，该企业设计了大量高性能、廉价、耗能低的 RISC 处理器。ARM 公司只提供 IP 授权，并不自行制造处理器。由于它的 ARM RISC 结构微处理器具有低功耗、高运算性能、以及高集成度等特性，使得许多芯片设计与芯片制造公司竞相向 ARM 公司购买 ARM 处理器的核心，作为自己公司微处理器的运算核心。因此，ARM 公司也成为全球最大的 IP 供应商^[4]。

目前，采用 ARM 技术知识产权为核心的微处理器，已经遍及汽车、工业控制、消费电子、通信系统、网络系统、无线系统等类产品市场，基于 ARM 技术的微处理器应用已占据 32 位 RISC 芯片 75% 以上的市场份额。

ARM 技术在下列领域中广泛应用：

(1)网络领域：随着宽带技术的推广，采用 ARM 技术的 ADSL 芯片正逐步获得竞争优势。此外，ARM 在语音及视频处理上进行了优化，并获得了广泛支持。

(2)工业控制领域：作为 32 位的 RISC 架构，基于 ARM 核的微控制器芯片不仅占据了高端微控制器市场的大部分市场份额，同时也逐渐向低端微控制器领域渗透。

(3)消费电子产品：ARM 技术在目前流行的数字音频播放器、数字机顶盒和游戏机中得到广泛采用。

(4)无线通讯领域：目前已有超过 85% 的无线通讯设备采用了 ARM 技术。

目前嵌入式处理器主要有 Am186/88、386EX、SC-400、PowerPC、MIPS、ARM 系列等。最受市场欢迎的嵌入式处理器就是由 ARM 公司出品的 ARM 系列处理器。目前主流的 ARM 处理器系列有：ARM7 系列、ARM9 系列、ARM10 系列、X Scale 系列、Strong ARM。下面介绍 ARM 体系结构的一些重要概念。

2.2.1 处理器状态

ARM 处理器有两种操作状态：ARM 状态下，处理器执行 32 位，字方式的 ARM 指令。Thumb 指令下，处理器执行 16 位，半字方式的 Thumb 指令。两个状态之间的切换不影响处理器模式或寄存器内容^[6]。

2.2.2 处理器模式

ARM 体系结构支持 7 种处理器模式，处理器在每种状态下拥有独立的堆栈寄存器 R13、链接寄存器 R14 和保护程序状态寄存器 SPSR（用户态与系统态共用），这种结构有利于系统状态切换和程序调用。它的 7 种模式如下：

(1)用户模式（user）正常程序工作模式，不能直接切换到其它模式。

(2)系统模式 (sys) 用于支持操作系统的特权任务等, 与用户模式类似, 但具有可以直接切换到其它模式等特权。

(3)快中断模式 (fiq) 支持告诉数据传输及通道处理, FIQ 异常响应时进入此模式。

(4)中断模式 (irq) 用于通用中断处理, IRQ 异常响应时进入此模式。

(5)管理模式 (svc) 操作系统保护代码, 系统复位和软件中断响应时进入此模式。

(6)数据访问中止模式 (abt) 处理器越权访问、取指令失败、执行 bkpt 指令, 或者无法完成一条指令的执行时, 进入数据访问中止模式。数据访问中止模式属于特权态, 完成异常处理后, 处理器通过“中断返回”回到原来的模式。

(7)未定义模式 (und) 支持硬件协处理器的软件仿真, 未定义指令异常响应时进入此模式。

2.2.3 流水线

一条指令的执行可分为取指、译码、取操作数、执行运算、存储器访问、写寄存器 6 个步骤。ARM 处理器使用流水线来增加处理器指令流的速度。这样可使几个操作同时进行, 并使处理和存储器系统连续操作。ARM7 架构采用 3 级流水线: 取指、译码、执行。ARM9 架构采用 5 级流水线。Xscale 架构采用 7 级流水线结构。

2.2.4 异常

只要正常的程序流程被暂时中止, 处理器就进入异常模式。例如响应一个来自外设的中断。ARM 处理器有 7 种异常。这 7 种异常类型如表 2-1 所示。

表 2-1 ARM 处理器的异常表

地址	异常类型	进入时的模式	优先级
0x0000 0000	复位	管理	1 (最高优先级)
0x0000 0004	未定义指令	未定义	6
0x0000 0008	软件中断	管理	7 (最低优先级)
0x0000 000C	中止 (预取)	中止	5
0x0000 0010	中止 (数据)	中止	2
0x0000 0018	中断 IRQ	中断	4
0x0000 001C	快中断 FIQ	快中断	3

2.2.5 存储器和存储器映射 I/O

ARM 处理器采用冯·诺依曼结构, 指令和数据共用一条 32 位数据总线。采用统一编址方式管理存储器和存储器映射 I/O。对不用设备的总线宽度、访问延迟可以通过专门的寄存器来设置。

2.3 ARM 芯片选择和功能

本课题采用 PHILIPS 的 LPC2220 微控制器^[5],它是基于一个支持实时仿真和嵌入式跟踪的 32/16 位 ARM7TDMI-S CPU。对代码规模有严格控制的应用可使用 16 位 Thumb 模式将代码规模降低超过 30%,而性能的损失却很小。LPC2220 的主要特性有:

- 16/32 位 ARM7TDMI-S 微控制器, LQFP144 和 TFBGA144 封装;
- 16/64kB 片内静态 RAM;
- 串行 boot 装载程序通过 UART0 来实现在系统下载和变成;
- EmbeddedICE-RT 和嵌入式跟踪接口使用片内 RealMonitor 软件对任务进行实时调试并支持对执行代码进行无干扰的高速实时跟踪。
- 8 路 10 位 A/D 转换器, 转换时间低至 2.44 μ s;
- 2 个 32 位定时器带 4 路捕获和 4 路比较通道、PWM 单元 (6 路输出)、实时时钟 RTC 和看门狗;
- 多个串行接口, 包括 2 个 16C550 工业标准 UART 和 2 个 SPI 接口;
- 向量中断控制器, 可配置优先级和向量低至;
- 通过外部存储器接口可将存储器配置成 4 组, 每组的容量 高达 16Mb, 数据宽度位 8/16/32 位;
- 多达 76 个通用 I/O 口 (可承受 5V 电压), 可使用 9 个边沿或电平触发的外部中断管脚;
- 通过可变成的片内锁相环 PLL 可实现最大位 60/75MHz 的 CPU 操作频率, 设置时间为 100 μ s;
- 带外部晶体的片内振荡器频率范围: 1~30MHz, 外部振荡器的频率高达 50MHz;
- 2 个低功耗模式: 空闲和掉电;
- 通过外部中断将处理器从掉电模式中唤醒;
- 可通过个别使能/禁止外部功能来优化功耗;
- 双电源, CPU 操作电压范围: 1.65V~1.95V (1.8V \pm 0.15V); I/O 操作电压范围: 3.0~3.6V (3.3V \pm 10%), I/O 口可承受 5V 电压。

2.4 系统总体设计方案及分析

2.4.1 USB 模块

近年发展起来的USB2.0技术以其高速易扩展等特点得到越来越广泛的应用。在本设计中主要针对USB移动类存储设备(U盘),使用USB2.0芯片CY7C68001和AMR7核心板连接实现ARM7与PC机的USB主控功能。

本设计将CY7C68001配置在ARM开发板的外设PACK总线上，采用CY7C68001的并行异步读写方式完成二者之间数据和命令的交换。其原理框图如图2.2所示。

使用开发板提供的对外部用户总线接口BANK2的访问控制API函数，实现ARM处理器与CY7C68001芯片的数据交换。CY7C68001芯片有存储器接口，4个状态信号（FLAGA,FLAGB,FLAGC,READY），一个中段信号USBINT。状态信号FLAGA,FLAGB,FLAGC和READY配置在系统状态寄存器0中，可由ARM7查询。

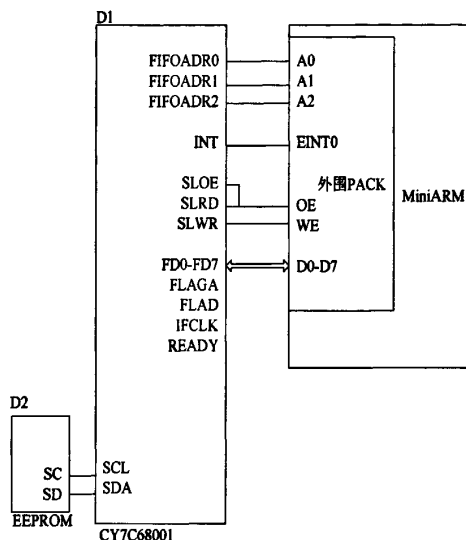


图2.2 CY7C68001与M22A的硬件接口

2.4.2 显示控制模块

本课题要外接一个真空荧光显示屏（VFD），显示控制器采用的是HOLTEK的HT16511^[10]。

HT16511是一个1/8到1/16占空因数驱动的VFD显示（真空荧光显示）控制器，对微控制器来说它是一种理想的外围设备。HT16511的作用是透过微控制器的指令控制栅极，而对真空荧光显示器所对应显示点进行控制。主要应用在具有仪表控制功能的消费产品以及工业测量设备。

其主要特点如下：

驱动器工作电压为：5V

驱动电压最大可达VDD-35V

12×4 键盘扫描功能

5 位 LED 输出接口

多种显示模式，从 8-栅极/16-区段到 12-栅极/20-区段

8 段/栅输出驱动线

SEG 1	SEG4	SEG8	SEG12	SEG16	SEG20	
00HL	00HU	01HL	01HU	02HL		DIG1
03HL	03HU	04HL	04HU	05HL		DIG2
06HL	06HU	07HL	07HU	08HL		DIG3
09HL	09HU	0AHL	0AHU	0BHL		DIG4
0CHL	0CHU	0DHL	0DHU	0EHL		DIG5
0FHL	0FHU	10HL	10HU	11HL		DIG6
12HL	12HU	13HL	13HU	14HL		DIG7
15HL	15HU	16HL	16HU	17HL		DIG8
18HL	18HU	19HL	19HU	1AHL		DIG9
1BHL	1BHU	1CHL	1CHU	1DHL		DIG10
1EHL	1EHU	1FHL	1FHU	20HL		DIG11
21HL	21HU	22HL	22HU	23HL		DIG12
24HL	24HU	25HL	25HU	26HL		DIG13
27HL	27HU	28HL	28HU	29HL		DIG14
2AHL	2AHU	2BHL	2BHU	2CHL		DIG15
2DHL	2DHU	2EHL	2EHU	2FHL		DIG16

b0 b3 b4 b7

xxHL	xxHU
------	------

低四位 高四位

图 2.3 HT16511 显存空间的地址配置

开发板提供了高速同步串行口SPI接口，可以方便地同VFD驱动芯片HT16511的串行数据接口连接。其中，保证开发板上JP6跳线组的P0.7跳线断开，使SSEL引脚接上拉电阻。这样ARM做主机，HT16511做从机。

2.4.3 网络控制模块

评估板提供了 1 路 10/100M 以太网接口（控制电路在模块上），电路原理如图 2.4 所示。评估板接口电路主要包括：指示灯、网络变压器、RJ45 接口。网络变压器的型号为 HR601680，该变压器发送端和接收端的变比都是 1:1。

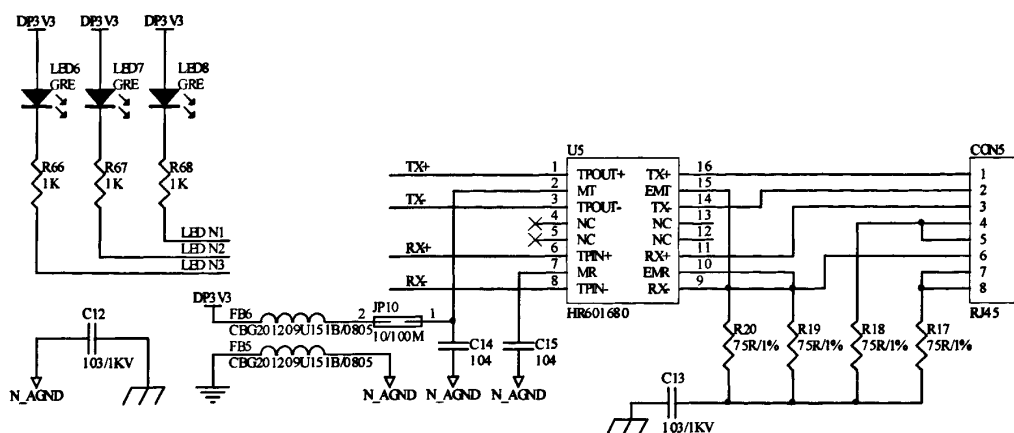


图 2.4 以太网接口电路

2.4.4 串口模块

本系统用了通用串口(UART0)。采用 SP3232E 进行 TTL 电平与 RS232 电平转换。由于 RS232 电平与 TTL 电平是相反的, -3~-15V 代表逻辑电平“1”, 3~-15V 代表逻辑“0”, 所以必须进行电平转换才能实现与计算机串口的通信。

SP3232E 系列是一个 2 驱动器/2 接收器的低功耗器件, SP3232E 系列有一个高效的电荷泵, 工作电压为 3.3V 时只需 0.1μF 电容就可进行操作。电荷泵允许 SP3232E 系列在 3.3V 到 5.0V 内的某个电压下发送符合 RS232 的信号。SP3243E 是一个 3 驱动器/5 接收器器件, 除以上特性外还包含一个辅助接收器, 当器件处于关断模式时, 接收器仍能通过外部器件的铃响指示器向监控器报警。

本系统是 3.3V 系统, 因此使用了 SP3232E 进行 RS232 电平转换。LPC2200 系列 MCU 的 UART1 带有完整的调制解调器 (MODEM) 接口, 所以要使用 8 路 RS232 转换芯片 SP3243E。如图 2.5 所示, JP3、JP2 分别为 UART0、UART1 口线的连接跳线, 当把它们断开时, 这些引脚保留给用户作为其它功能使用。

注意: LPC2200 系列 MCU 的 ISP 使能引脚 (P0.14 口) 与 DCD1 功能复用, 在系统复位时若 P0.14 口为低电平, 则进入 ISP 状态; 同样, 在程序仿真调试过程中, 若把 JP1 短接, 则 DCD1 保持为低电平, 影响 MODEM 接口正确使用。

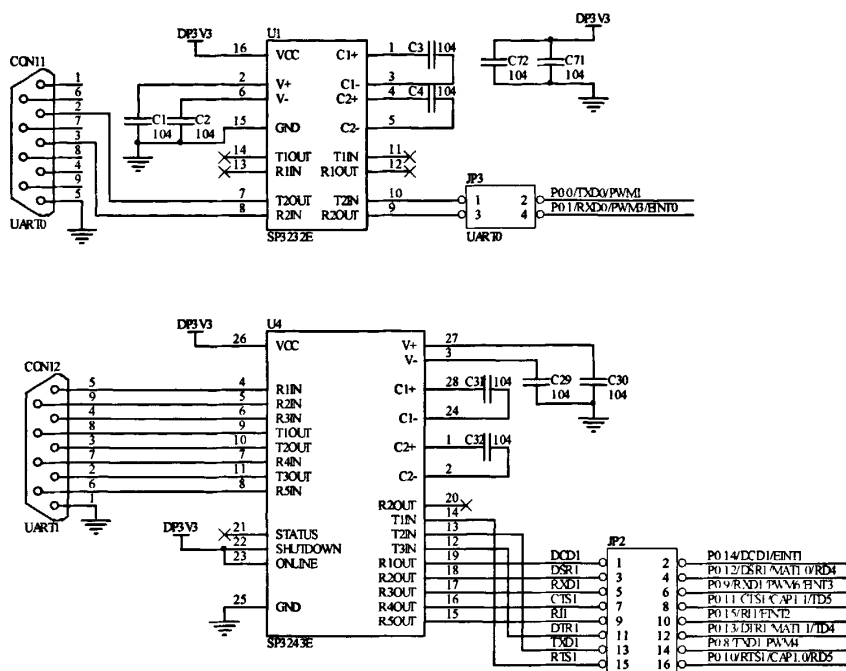


图 2.5 串口模块电路图

2.4.5 SD 卡接口模块

ARM 开发板具有 SD 卡接口，支持 SD/MMC 卡的读写，电路原理如图 2.6 所示。SD/MMC 卡可以采用 SD 总线访问，也可以采用 SPI 总线访问，考虑到 LPC22xx 微控制器有 SPI 接口而没有 SD 总线接口，所以直接使用 SPI0 (P0.4_SCK0、P0.5_MISO0 和 P0.6_MOSI0 引脚) 与其连接。如果采用 I/O 口模拟 SD 总线，不但增加了软件的开销，而且大大降低了总线数据传输速度。JP13 可以断开 SD 卡接口电路与 ARM 核心板的连接。

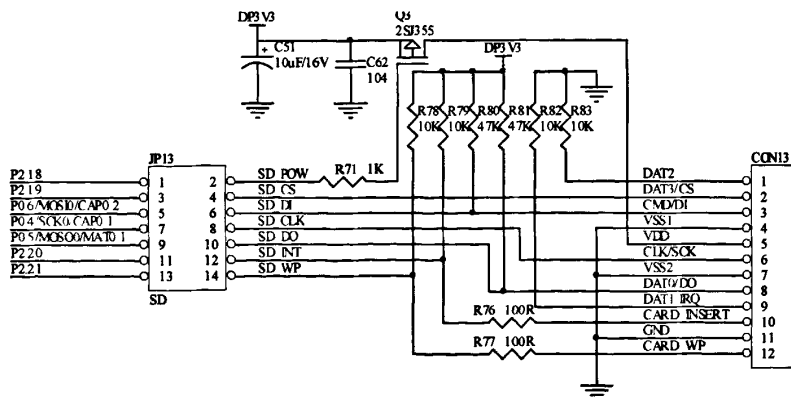


图 2.6 SD 接口电路

可见, SD/MMC 卡接口电路包含以下几个部分:

·SPI 总线

如图 2.7 所示, LPC2220 中 SPI 接口的 P2.19、P0.4_SCK、P0.6_MOSI、P0.5_MISO 直接连接到卡座的相应接口, 其中 SPI 的两个数据线 P0.6_MOSI、P0.5_MISO 还分别接上上拉电阻, 这是为了使本电路可以与 MMC 卡的接口兼容。SPI 模式下无需用到的信号线 DAT2 和 DATA1 分别接下拉电阻。

·卡供电控制

卡的供电采用可控方式, 这是为了防止 SD/MMC 卡进入不确定状态时, 可以通过对卡重新上电使卡复位而无需拔出卡。

可控电路采用 P 型 MOS 管 2SJ355, 由 LPC22xx 的 GPIO 口 P2.18 进行控制, 当 P2.18 输出高电平时, 2SJ355 关断, 不给卡供电; 当 P2.18 输出低电平时, 2SJ355 开通, DP3V3 电源(电压为 3.3V) 给卡供电。

采用 2SJ355 的目的是当它开通时, 管子上的压降比较小。用户也可以采用其它 P 型的 MOS 管, 但是要考虑管子开通时, 漏极与源极之间的压降要足够小(保证 SD/MMC 卡的工作电压在允许范围内), 管子允许通过的电流也要满足卡的要求, 一般一张 SD/MMC 卡工作时的最大电流通常为 45mA 左右, 所以选用的 MOS 管要求允许通过 100mA 左右的电流。

·卡检测电路

卡检测电路包括两部分: 卡是否完全插入到卡座中和卡是否写保护。

检测信号由卡座的两个引脚以电平的方式输出。当卡插入到卡座并插入到位时, CARD_INSERT(第 10 脚) 由于卡座内部触电连接到 GND, 输出低电平; 当卡拔出时, 该引脚由于上拉电阻 R79 的存在而输出高电平, 该输出由 LPC22xx 的输入引脚 GPIO(P2.20)来检测。

卡是否写保护的检测与卡是否完全插入到卡座中的检测原理是一样的。

2.4.6 外围 PACK 接口

MiniARM 工控板是总线开放型的模块, 为片外存储器提供了一个接口, 支持 SRAM、ROM、FLASH 和外部 I/O 器件。开发板上设计了一个外设 PACK, 电路如图 2.7 所示, 具有 24 根地址总线 A0~A23, 16 根数据总线 D0~D15, 读使能信号 OE、写使能信号 WE、字节定位选择信号 BLS0 和 BLS1, 片选信号为 nCS2_Y0, 所以外设 PACK 的地址为 0x82000000~0x821FFFFFFF。外设 PACK 上还有 6 个 I/O 口, 其中, 有 2 个 I/O 为外部中断引脚, 这样就极大的方便了与外部 I/O 器件进行连接。

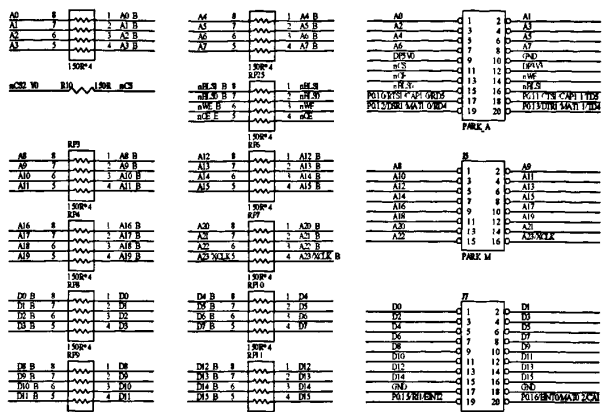


图 2.7 外围 PACK 接口电路

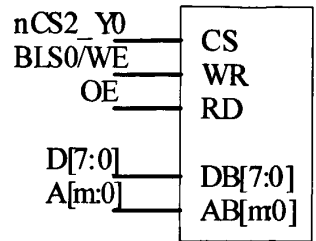


图 2.8 8 位总线器件连接示意图

在外设 PACK 处也可以扩展 8 位总线器件，8 位总线器件的连接示意图如图 2.8 所示。对于 8 位总线器件来说，就没有字节定位信号了，此时，总线器件的写使能信号可以使用 WE 也可以使用 BLS0。

2.4.7 电源管理模块

ARM 开发板电源来自外置 9V 直流电源，LM2575 具有输入电压范围宽，输出电流大(1A)，耗散功率小的优点。因此使用 LM2575 简易开关电源获得+5.0V 的电压，数字地(GND)和模拟地(AGND)使用磁珠相隔离。电源跳线 JP7 可以辅助测试评估板的电流功耗。

SPX1117 有很低的静态电流，在满负载时其低压差仅为 1.1V。当输出电流减少时，静态电流随负载变化，并提高效率，因此使用 SPX1117-3.3 获得+3.3V 电压。

表 2.4 M22A 评估板电源标号

网络标号	用途
DP3V3	+3.3V 数字电源，专供核心板模块经过 Power Safeguard 保护
DP5V0	+5.0V 数字电源，专供核心板模块
GND	数字地（系统地）
AGND	模拟地

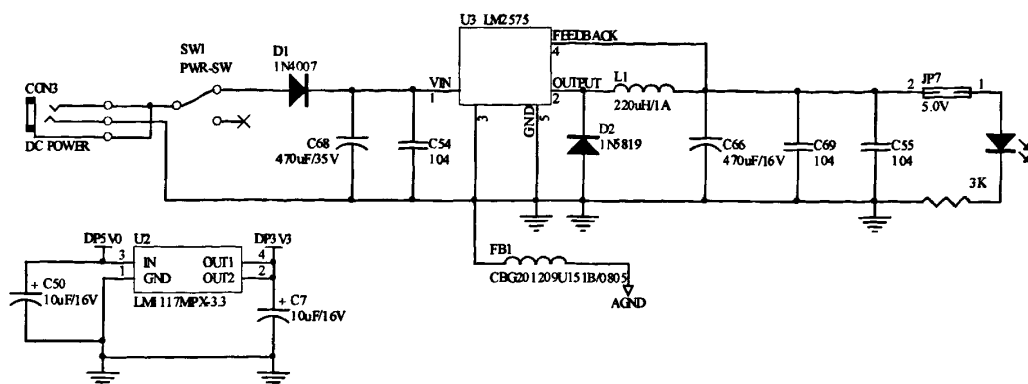


图 2.9 电源管理模块电路图

2.4.8 仿真和调试

LPC2210/2220 支持通过 JTAG 串行端口进行仿真和调试。跟踪端口允许跟踪程序的执行。调试和跟踪功能只在 GPIO 的 P1 口复用。这意味着当应用在嵌入式系统内运行时，位于 P0 口的所有通信、定时器和接口外设的开发和调试阶段都可用。

开发板采用 ARM 公司提出的标准 20 脚 JTAG 仿真调试接口，JTAG 信号的定义与 ARM 核心板的连接如图所示。

ARM 核心板在复位期间，如果引脚 P1.26/RTCK 为低，使能 JTAG 调试引脚 (P1.31~P1.26)，此时这些引脚不能够作为 GPIO 使用。由于引脚 P1.26/RTCK 内部含有上拉电阻，所以需要外接一下拉电阻，推荐的电阻阻值为 4.7K。

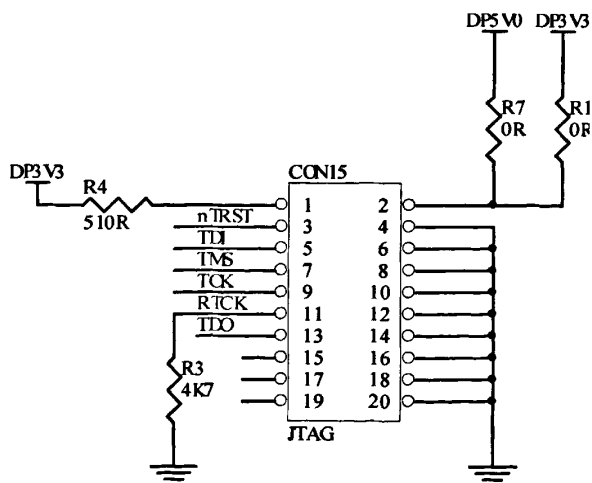


图 2.10 JTAG 接口电路

在实际应用过程中，可以使用 5V 电源也可以使用 3V 电源，可以根据实际情况来选择。评估板上选择 3V 电源，即，电阻 R7 取下。

2.5 USB 总线协议分析

USB 是用于将适用 USB 的外围设备连接到主机的外部总线结构,其主要是用在高速和低速的外设。USB 是通过 PCI 总线和 PC 的内部系统数据线连接,实现数据的传送。USB 同时又是一种通信协议,它支持主系统(host)和 USB 的外围设备(device)之间数据传送。USB 总线的主要优势体现在以下几个方面^[7]:

(1)USB 为所有的 USB 外设提供了单一的、易于操作的标准的连接类型。这样就简化了 USB 外设的设计,同时也简化了用户在判断哪个插头对应哪个插槽时的任务,实现了单一的数据通用接口。

(2)USB 排除了各个设备如鼠标、调制解调器、键盘和打印机设备对去系统资源的需求,因而减少了硬件的复杂性和对端口的占用,整个的 USB 的系统只有一个端口和一个中断,节省了系统资源。

(3)USB 支持热插拔(hot plug),也就是说在不关 PC 的情况下可以安全的插上和断开 USB 设备,动态的加载驱动程序。其他普通的外围连接标准,如 PCI 设备等必须在关掉主机的情况下才能增加或移走外围设备。

(4)USB 支持 PNP。当插入 USB 设备的时候,计算机系统检测该外设并且通过自动的加载相关的驱动程序来对该设备进行配置,并使其正常工作。

(5)USB 在设备供电方面提供了灵活性。USB 直接连接到 Hub 或者是连接到 Host 的设备可以通过 USB 电缆供电,也可以通过电池或者其它的电力设备来供电,或使用两种供电方式的组合.并且支持节约能源的挂机和唤醒模式。

(6)USB 提供高速 480Mbps 的速率、全速 12Mbps 的速率和低速 1.5Mbps 的速率这三种数据传输率来适应各种不同类型的外设。

(7)针对不能处理突发的非连续传送的设备,如音频和视频设备,USB 可以保证其固定带宽。

(8)为了适应各种不同类型外围设备的要求,USB 提供了四种不同的数据传送类型。

2.5.1 USB 系统描述

USB 系统主要被定义为三个部分:USB 的设备、USB 的主机、USB 的互连。

USB 设备需要提供自检和属性设置的信息,USB 设备必须在任何时刻执行与所定义的 USB 设备的状态相一致的动作。USB 设备主要分为两种类型:集线器和功能部件。只有集线器可以提供更多的 USB 的连接点,功能部件为主机提供了具体的功能。每个功能设备都包含设置信息,来描述该设备的性能和所需资源,主机要在功能部件使用前对其进行设置。当设备被连接、编号后,该设备就拥有一个唯一的 USB 地址。所有 USB 设备必须在零号端口上有一指定的通道,每个 USB 设备的 USB 控制通道将与之相连。

通过此控制通道,所有的 USB 设备都列入一个共同的准入机制,以获得控制操作的信息。在零号端口上,控制通道中的信息应完整的描述 USB 设备。

USB 的主机通过主机控制器与 USB 设备进行交互。主机功能有:检测 USB 设备的安装和拆卸、管理在主机和 USB 设备之间的控制流和数据流、收集状态和动作信息、提供能量给连接的 USB 设备。在任何 USB 系统中,只有一个主机。USB 和主机系统的接口称作主机控制器,主机控制器可由硬件、固件和软件综合实现^[8]。

USB 设备与主机之间进行连接和通信的操作,主要包括以下方面:

- 总线的拓扑结构:USB 设备与主机之间的各种连接方式;
- 内部层次关系:根据性能叠置,USB 任务被分配到系统每一个层次;
- 数据流模式:描述了数据在系统中通过 USB 从产生方到使用方的流动方式;
- USB 的调度:USB 提供了一个共享的连接,对可以使用的连接进行了调度以支持同步数据传输,并且避免了优先级判别的开销。

2.5.2 USB 总线数据传输

USB 总线属一种轮询方式的总线,即每一次的通讯均由主机端发起,主机控制端口初始化所有数据传输。USB 数据的传输是通过管道进行的。USB 系统软件通过缺省管道(与端点 0 相对应)管理设备,设备驱动程序通过其它的管道来管理设备的功能接口。USB 的结构包含四种基本的数据传输类型^[9]。

(1)控制传送

控制传输属于突发式、非周期性的,由主机软件发起的请求或响应的通信,通常用于命令事务和状态事务。控制传输适用于传输少量的、且对传输时间和传输速率均无要求、但必须保证传输正确的数据。其主要用于发送和接收与 USB 设备的配置信息有关的数据,如设备地址、配置描述符等,也可用于传输其他用途的数据。控制传输可用于低速、全速和高速设备,且所有的 USB 设备都必须支持控制传输。

(2)批量传送

批量传输是非周期性的、大量的突发性传送。典型的用于传送那些可以利用任何带宽的数据,而且当没有可用带宽时,可以延时传输。批量传输适合使用在时间不重要的传输。批量传输可以传输大量的数据而不会阻塞总线,因为它会让其他传输类型先执行,等待有可以传输的时间。使用批量传输的包括从主机传送数据给打印机,从扫描仪传送数据给主机,以及读/写磁盘等。在一个闲置的总线中,批量传输是最快的传输类型。只有全速与高速设备,才可以使用批量传输。

(3)中断传输

中断数据是少量的,且其数据延迟时间也是有限范围的。这种数据可由设备在任何时刻发送,并以不慢于设备指定的速度在 USB 上传送。中断数据一般由事件通告,特

征及座标号组成, 只有一个或几个字节。虽然不必要精确指定传输率, 但 USB 必须对交互数据提供一个反应时间的最低界限。

(4)同步传输

同步传输适用于传输大量的、速率恒定的、且对服务周期有要求的数据。同步数据的建立、传送和使用是连续且实时的, 同步数据要使接收者与发送者保持相同的时间安排, 保证大量数据可以尽快地通过总线。只有全速和高速设备可以做同步传输。

2.5.3 USB 设备请求

在 USB 通信协议中, 由于主机是取得绝对的主动权, 主机和设备之间就必须遵循某种已经沟通的特定的命令格式, 以达到通信的目的。这个命令格式就是 USB 规范中所定义的“设备请求(device request)”。而设备请求的设置、清除与取得都必须通过控制传输的数据交换来达到。

USB 规范中定义了一些标准的设备请求。USB 标准设备请求是用来完成 USB 设备枚举的过程, USB 设备必须对标准设备请求作出响应, 不管该设备是否已经被分配了一个默认的地址或该设备目前正在配置。所有的标准设备请求都是使用默认管道来传输的。这里列出常用的 n 种 USB 标准设备请求获取状态、清除特性、设置特性、设置地址、获取描述符、设置描述符、获取配置信息、设置配置、获取接口信息、设置接口、同步帧。

2.5.4 USB 通信流

USB 数据交换是由 host 发起, 由 Token 包开始, 接下来是双方向上的数据包, 在数据包传送完之后, 就会由设备返回一个握手包。USB 系统通过 IN, OUT 和 SETUP 的包来指定 USB 地址和端点(最多是 128 个, 0 通常被用来用做缺省的传送配置信息的), 并且这些被指定的设备必须通过上面形式的包来回应这种形式的指定。每个 SETUP 的包含 8 个 byte 的数据, 数据用来指示传送的数据类型。对于 DATA 数据包来说, 设置两种类型的数据包是为了能够在传送数据的时候做到更加的精确。ACK handshake 包用来指不数据传送的正确性, 而 STALL handshake 则表示数据包在传送的过程中出了故障, 并且请示 host 重新发数据或者清除这次传送。PRE 格式的包主要是用在在一个 USB 的系统中如果存在不同速率的设备的时候, 对于不同于总线速度的设备中, 会回应一个 PRE 的包从而会忽略该设备。

一个 USB 逻辑设备对 USB 系统来说就是一个端点的集合。USB 系统软件通过缺省控制管道来管理设备, 而客户端软件通过管道束来管理接口。管道束一端为端点, 另一端为缓冲区。客户端软件要求通信数据在主机的个缓冲区和设备的一个端点之间进行。主机控制器或 USB 设备将数据打包后在 USB 总线上传输, 由主机控制器协调何时通过

总线访问传输数据。关于端点和管道的分析如下。

(1)设备端点

端点是 USB 设备唯一的可识别部分，是主机和设备间通信流的结束点。一系列相互独立的端点一起构成了 USB 逻辑设备。每个逻辑设备具有唯一的地址，在设备连接到 1 机时由主机分配。设备中的每个端点在设备内部有唯一的端点号，在设备设计时由开发者指定。每个端点都是个单向的连接点，支持数据流入或者流出设备。端点的特性决定了它与客户端软件进行传输的类型，应在设计时指定，包括端点的总线访问频率要求、总线延迟要求、带宽要求、端点号、数据包的最大长度、端点的传输类型和方向等信息。

USB 设备需要一种缺省的控制方法，该方法将端点 0 既作为输入又作为输出端点，通过缺省控制管道初始化并配置 USB 设备。设备一旦连接到 USB 总线并上电，在收到一个总线复位命令后，就可以对端点 0 进行访问。而对于非 0 端点，只有在成功配置之后才能使用。

(2)管道

USB 管道是建立在 USB 设备的端点与主机软件之间的某种联系，管道代表了借助缓冲区在主机软件和设备的某个端点之间传输数据的能力。在 USB 系统中，有两种类型的管道，它们是流管道和消息管道。流管道从总线事务的数据包中获得数据，并且该数据不具有 USB 所定义的结构。数据流从一个流管道的一端流入，从另一端流出。在整个通信流中，流管道始终是单向的。消息管道和流管道有很大的不同，消息管道允许数据流在同一个管道的两个方向上发生，并且传输的数据具有特定的结构。缺省控制管道(包含端点 0)就是一个消息管道。

2.6 USB Mass Storage 类协议

2.6.1 USB 的端点

端点是 USB 中一个独特的概念，它是一个可以与 USB Host 交换数据的硬件单元。USB Host 与 USB 设备之间都是通过端点来传输数据的，端点是桥梁和纽带，不同的端点其传输数据的能力不同，适于不同的应用场合。端点可通过 Set Mode 命令配置为 4 种不同的模式，分别为：模式 0 模式、非同步传输模式 1 模式、同步输出传输模式 2 模式、同步输入传输模式 3 模式、同步输入输出传输，端点 2 叫做主端点，它在有些方面是比较特别的，它是进行吞吐大数据的主要端点，同样地它执行主机的特性以减轻传输大数据的任务：

(1)双缓冲，允许 USB 与本地 CPU 之间的并行读写操作，这样就增加了数据的吞吐量。缓冲区切换是自动处理的。这导致了透明的缓冲区操作；

(2)支持 DMA 直接存储器访问操作,可以和对其它端点的正常输入/输出操作交叉进行;

(3)缓冲区的最大信息包长度较其它端点大;

(4)可配置为同步传输或非同步批量和中断传输。

2.6.2 Mass Storage 协议

USB 协议能够在启动或是当设备插入系统时对设备进行配置,这就是 USB 设备为什么可以直插拔的原因。USB 设备被分成以下几类:显示器、通讯设备、音频设备、人机输入、海量存储。特定类的设备又可划分成子类,划分子类后的软件就可以搜索总线并选择所有它可以支持的设备。每个设备可以有一个或多个配置,配置用于定义设备的功能。如果某个设备有几种不同的功能,则每个功能都需要一个配置。配置是接口的集合。接口指定设备中的哪些硬件与 USB 交换数据。每一个与 USB 交换数据的硬件就叫做一个端点。因此,接口是端点的集合。USB 的设备类别定义了特定类或子类中的设备需要提供的缺省配置、接口和端点。描述符描述设备、配置、接口或端点的一般信息,图 2.11 为 USB 描述符的层次结构。

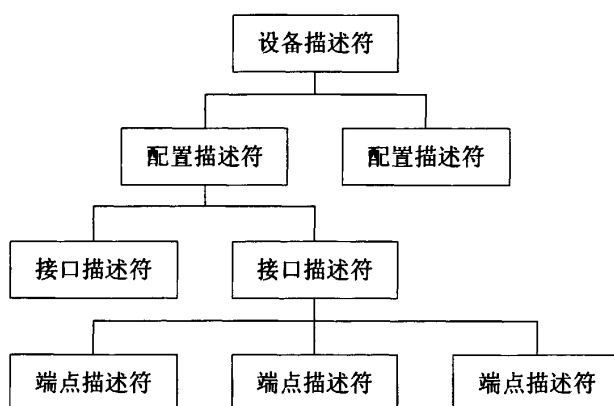


图 2.11 描述符的层次结构

主机向设备提出的这些问题实际上就是 USB 协议中规定的各种标准请求,设备必须对这些问题进行回答,而回答的方式就是向主机传送相应的描述符:设备描述符、配置描述符、接口描述符、端点描述符。

(1)设备描述符

设备描述表给出了 USB 设备的一般信息。这包括对设备及所有设备配置起全程作用的信息。一个 USB 设备只能有一个设备描述表。所有的 USB 设备都有缺省控制通道。缺省控制通道的最大包长在设备描述表中得到了说明。

(2)配置描述符

配置描述表给出了设备配置的信息,描述表包括一个 b Configuration Value 域,在

Set Configuration()请时被用作参数来设置所需配置。这个描述表给出了此配置下的接口数, 每个接口可能独立操作。比如, 一个 ISDN 设备可能配置有两个接口, 每个都提供 64KB/S 的有独立数据源与数据接收者的双向通道在另一个配置下 ISDN 可能表现为单个接口, 将两个通道合成一个 128KB/S 的双向通道。当主机发出请求要得配置描述表时, 所有相关接口与端节点的描述表都被返回。一个 USB 设备有一个或多个配置。每个配置只有一个或多个接口。而每个接口又有 0 个或多个端节点。在一个配置下, 一个端节点不会在接口之间共享, 除非端节点被同一个接口的不同设置使用。在不同配置端节点, 可无此限制。一个配置好后, 设备可支持对配置的有限调整, 如果一个接口有备选设置, 在配置好后可选择不同设置。

(3)接口描述符

此描述表在一个配置内给出一个接口的信息。如果一个配置支持不止一个接口, 端节点的描述表会跟在接口描述表后被返回, 接口描述表总是作为配置描述表的一部分被返回。接口描述不可直接用 Set Description()和 Get Descriptor()存取。一个接口可能包含备选设置, 以使得端节点或他们的特性在设备配置好以后能改变。一个接口的缺省设置总是可选设置。Set Interface()与 Get Interface()用来选择与返回选择了的接口设置。可选的接口设置使得部分的设备配置能在其它接口进行操作的情况下改变。如果一个配置对于它的一个或多个接口有备选设置, 每一设置包括一个独立接口描述表和相关结点。如果一个设备配置支持单个接口, 并且接口有两个可选设置, 配置描述表返回以后会紧跟着返回 b Interface Number 与 b Alternate Setting 域皆为 0 的第一个设置的接口描述表及相关的结点描述表, 而随之后是另一个设置接口描述表与结点描述表。第二个接口描述表的 b Interface Number 域也应为 0, 但 b Alternate Setting 域应为 1。如果一个接口仅使用节点 0, 则接口描述表以后就不再返回节点描述表, 并且此接口表示的是一个请求接口, 它使用连在节点 0 上的缺省通道。在这种情况下 b Number Endpoints 域被设置成 0。一个接口描述表的节点个数不把结点 0 计在内。

(4)端点描述符

每个接口使用的结点都有自己的描述表, 此描述表被主机用来决定每个节点的带宽需求。

2.7 本章小结

本章在分析 ARM 处理器的结构和特点的基础上, 设计了以 PHILIPS 的 LPC2220 微控制器为核心的核心板, 并设计了 USB 接口、显示模块、网络控制模块、串口模块、电源模块、仿真和调试环境, 形成了嵌入式平台的硬件环境, 为以后的开发打下了基础。

3 高端仪表平台的软件设计

3.1 ADS 集成开发环境

ARM ADS 全称为 ARM Developer Suite。是 ARM 公司推出的新一代 ARM 集成开发工具。现在 ADS 的最新版本是 1.2, 它取代了早期的 ADS1.1 和 ADS1.0。它除了可以安装在 Windows NT4, Windows 2000, Windows 98 和 Windows 95 操作系统下, 还支持 Windows XP 和 Windows Me 操作系统。

ADS 由命令行开发工具, ARM 实时库, GUI 开发环境(CodeWarrior 和 AXD), 实用程序和支持软件组成。本系统的应用程序就通过这些部件开发调试。

3.2 嵌入式操作系统 $\mu\text{C}/\text{OS-II}$

$\mu\text{C}/\text{OS-II}$ 意为“微控制器操作系统版本 2”, $\mu\text{C}/\text{OS-II}$ 是源码公开的著名实时内核, 可用于各类 8 位、16 位和 32 位单片机或 DSP。从 $\mu\text{C}/\text{OS}$ 算起, 该内核已有 10 多年应用史, 在诸多领域得到广泛应用。

$\mu\text{C}/\text{OS-II}$ 是一个完整的、可移植、可固化、可剪裁的抢先式实时多任务内核。 $\mu\text{C}/\text{OS-II}$ 使用 ANSI C 语言编写, 包括一小部分汇编代码, 使之可以供不同架构的微处理器使用。至今, 从 8 位到 64 位, $\mu\text{C}/\text{OS-II}$ 已在超过 40 种不同架构的微处理器上运行。

嵌入式 $\mu\text{C}/\text{OS-II}$ 操作系统特点如下^[13]:

提供源代码: 购买作者撰写的《嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 》一书即可获得 $\mu\text{C}/\text{OS-II}$ V2.52 版本的所有源代码, 购买此书的其它版本可以获得相应版本的全部源代码。

可移植性: $\mu\text{C}/\text{OS-II}$ 的源代码绝大部分是使用移植性很强的 ANSI C 写的, 将与微处理器硬件相关的汇编语言使用量压缩到最低的限度, 以使 $\mu\text{C}/\text{OS-II}$ 便于移植到其它微处理器上。目前, $\mu\text{C}/\text{OS-II}$ 已经被移植到多种不同架构的微处理器上。

可固化: 只要具备合适的软硬件工具, 就可以将 $\mu\text{C}/\text{OS-II}$ 嵌入到产品中成为产品的一部分。

可剪裁: $\mu\text{C}/\text{OS-II}$ 使用条件编译实现可剪裁, 用户程序可以只编译自己需要的功能, 而不编译不需要的功能, 以减少 $\mu\text{C}/\text{OS-II}$ 对代码空间和数据空间的占用。

可剥夺: $\mu\text{C}/\text{OS-II}$ 是完全可剥夺性的实时内核, $\mu\text{C}/\text{OS-II}$ 总是运行就绪条件下优先级最高的任务。

多任务: $\mu\text{C}/\text{OS-II}$ 可以管理 64 个任务, 然而, $\mu\text{C}/\text{OS-II}$ 的作者建议用户保留 8 个给 $\mu\text{C}/\text{OS-II}$ 。这样, 留给用户的应用程序最多可有 56 个任务。

可确定性:绝大多数 $\mu\text{C}/\text{OS-II}$ 的函数调用和服务的执行时间具有确定性,也就是说,用户总是能知道 $\mu\text{C}/\text{OS-II}$ 的函数调用与服务执行了多长时间。

任务栈: $\mu\text{C}/\text{OS-II}$ 的每个任务都有自己单独的栈,使用 $\mu\text{C}/\text{OS-II}$ 的占空间校验函数,可确定每个任务到底需要多少栈空间。

系统服务: $\mu\text{C}/\text{OS-II}$ 提供很多系统服务,例如信号量、互斥信号量、时间标志、消息邮箱、消息队列、块大小固定的内存的申请与释放及时间管理函数等。

中断管理:中断可以使正在执行的任务暂时挂起,如果优先级更高的任务被中断唤醒,则高优先级的任务在中断嵌套全部退出后立即执行,中断嵌套层数可达 255 层。

稳定性与可靠性: $\mu\text{C}/\text{OS-II}$ 是 $\mu\text{C}/\text{OS}$ 的升级版, $\mu\text{C}/\text{OS}$ 自 1992 年以来已经有数百个商业应用。2000 年 7 月, $\mu\text{C}/\text{OS-II}$ 在一个航空项目中得到了美国联邦航空管理局对商用飞机的、符合 RTCA DO178B 标准的认证。这表明,该操作系统的质量得到了认证,可以在任何应用中使用。

3.3 嵌入式 $\mu\text{C}/\text{OS-II}$ 的移植规划

移植,就是使一个实时内核能在其它的微处理器或微控制器上运行。尽管大部分 $\mu\text{C}/\text{OS-II}$ 的代码是用 C 语言编写的,但是在编写与处理器硬件相关的代码时还是不得不使用汇编语言。移植的主要工作就是编写这些与处理器硬件相关的代码。在移植前针对所使用的微处理器进行规划,主要有以下几个方面的考虑:编译器的选择、任务模式的选择和支持的指令集^{[14][16]}。

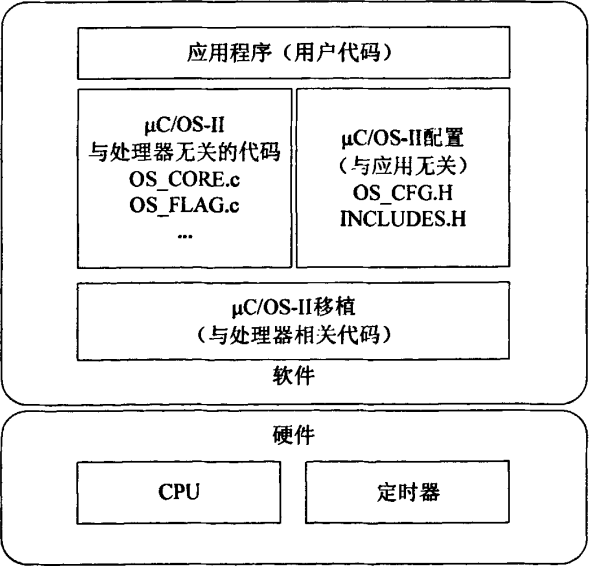


图 3.1 $\mu\text{C}/\text{OS-II}$ 硬件软件体系结构

3.3.1 编译器的选择

目前,针对 ARM 处理器核的 C 语言编译器有很多,如 SDT、ADS、IAR、TASKING 和 GCC。SDT 和 ADS 均为 ARM 公司自己开发,ADS 为 SDT 的升级版。GCC 虽然支持广泛,很多开发套件使用它作为编译器,但是与 ADS 比较其编译效率较低,这对充分发挥芯片性能不利,所以最终选择使用 ADS 编译程序和调试。

3.3.2 任务模式的取舍

ARM7 处理器核具有用户、系统、管理、中止、未定义、中断和快中断七种模式,其中除了用户模式外其它均为特权模式。其中管理、中止、未定义、中断和快中断模式与相应异常相联系,任务使用这些模式不太适合,而系统模式除了是特权模式外,其它与用户模式一样,因而可选为任务使用的模式只有用户模式和系统模式。为了尽量减少任务代码错误对整个程序的影响,缺省的任务模式定为用户模式,可选为系统模式,同时提供接口使任务可以在这两种模式间切换。

3.3.3 支持的指令集

带 T 变量的 ARM7 处理器核具有两个指令集:标准 32 位 ARM 指令集和 16 位 Thumb 指令集,两种指令集有不同的应用范围。为了最大限度地支持芯片的特性,任务应当可以使用任意一个指令集并可以自由切换,而且不同的任务应当可以使用不同的指令集,这次移植的代码已经实现了这一点。

3.4 移植 $\mu\text{C}/\text{OS-II}$

要移植一个操作系统到一个特定的 CPU 体系结构上并不是一件很容易的事情,它对移植者有以下要求^[13]:

- 对目标体系结构有很深了解;
- 对 OS 原理要有较深入的了解;
- 对所使用的编译器要有较深入的了解;
- 对需要移植的操作系统要有相当的了解;
- 对具体使用的芯片也要一定的了解。

3.4.1 移植需要编写的文件

根据 $\mu\text{C}/\text{OS-II}$ 的要求,移植 $\mu\text{C}/\text{OS-II}$ 到一个新的体系结构上需要提供 3 个文件:

OS_CPU.H (C 语言头文件)

OS_CPU_C.C (C 程序源文件)

OS_CPU_A.ASM (汇编程序源文件)

移植代码包括的内容见表 3.1 表:

表 3.1 移植代码内容

移植内容	类型	所属文件	描述
BOOLEAN、INT8U、INT8S、 ...	数据类型	OS_CPU.H	与编译器无关的数据类型
OS_STK	数据类型	OS_CPU.H	堆栈的数据类型
OS_ENTER_CRITICAL() 和 OS_EXIT_CRITICAL()	宏	OS_CPU.H	开关中断的代码
OS_STK_GROWTH	常量	OS_CPU.H	定义堆栈的增长方向
OS_TASK_SW	函数	OS_CPU.H	任务切换时执行的代码
OSTaskStkInit()	函数	OS_CPU_C.C	任务堆栈初始化函数
OSInitHookBegin()、 OSInitHookEnd()、...	函数	OS_CPU_C.C	μC/OS-II 在执行某些操作时调用的用户函数，一般为空
OSStartHighRdy()	函数	OS_CPU_A.ASM	进入多任务环境时运行优先级最高的任务
OSIntCtxSw()	函数	OS_CPU_A.ASM	中断退出时的任务切换函数
OSTickISR()	中断服务程序	OS_CPU_A.ASM	时钟节拍中断服务程序

3.4.2 头文件 includes.h 和 config.h

μC/OS-II 要求所有.C 文件都要包含头文件 includes.h, 这样使得用户项目中的每个.C 文件不用分别去考虑它实际上需要哪些头文件。使用 includes.h 的缺点是它可能会包含一些实际不相关的头文件, 这意味着每个文件的编译时间可能会增加, 但却增强了代码的可移植性。

在本移植中另外增加了一个头文件 config.h, 要求所有用户程序必须包 config.h, 在 config.h 中包含 includes.h 和特定的头文件和配置项。而 μC/OS-II 的系统文件依然只是包含 includes.h, 即 μC/OS-II 的系统文件完全不必改动。所有的配置改变包括头文件的

增减均在 config.h 中进行, 而 includes.h 定下来后不必改动 $\mu\text{C}/\text{OS-II}$ 的系统文件需要编译的次数大大减少, 编译时间随之减少。

3.4.3 编写 OS_CPU.H

$\mu\text{C}/\text{OS-II}$ 不使用 C 语言中的 short、int、long 等数据类型的定义, 因为它们与处理器类型有关, 隐含着不可移植性。代之以移植性强的整数数据类型, 这样, 既直观又可移植, 不过这就成了必须移植的代码。根据 ADS 编译器的特性, 这些代码如程序清单所示。

```
typedef    unsigned char    BOOLEAN;
typedef    unsigned char    INT8U;
typedef    signed char      INT8S;
typedef    unsigned short   INT16U;
typedef    signed short     INT16S;
typedef    unsigned int     INT32U;
typedef    signed int       INT32S;
typedef    float            FP32;
typedef    double           FP64;
typedef    INT32U           OS_STK
```

3.5 移植代码应用到 LPC2220

将 $\mu\text{C}/\text{OS-II}$ 移植到 LPC2220 时, 需要完成的工作主要有以下几个部分:

- 编写或获取启动代码;
- 挂接 SWI 软件中断;
- 中断及时钟节拍中断;
- 编写应用程序。

3.5.1 编写驱动代码

一般在 32 位 ARM 应用系统中, 大多数采用 C 语言进行软件编程。但是在运行应用代码前需要进行系统初始化。常用一个汇编文件作启动代码, 它可以实现异常向量表定义、堆栈初始化、系统变量初始化、中断系统初始化、I/O 初始化、外围部件初始化、地址重映射等操作。

3.5.2 挂接 SWI 软件中断

为使底层接口函数与处理器状态无关, 同时在任务调用相应的函数不需要知道函数

位置, 在移植中使用软中断指令 SWI 作为底层接口,使用不同的功能号区分不同的函数。
在移植中分配四个功能号给要移植的四个功能函数。参考如下:

```
底层功能函数声明(os_cup.h):
__swi(0x00)void OS_TASK_SW(void);           /*任务级任务切换函数*/
__swi(0x01)void OSStartHighRdy(void);       /*运行优先级最高的任务*/
__swi(0x02)void OS_ENTER_CRITICAL(void); /*关中断*/
__swi(0x03)void OS_EXIT_CRITICAL(void);  /*开中断*/
```

将软中断异常处理程序挂接到内核是通过修改启动代码中的异常向量表实现, 代码如下所示:

```
Reset

        LDR    PC,ResetAddr
        LDR    PC,UndefinedAddr
        LDR    PC,SWI_Addr
        LDR    PC,PrefetchAddr
        LDR    PC,DataAbortAddr
        LDR    0xb9205f80
        LDR    PC,[PC,#-0xff0]
        LDR    PC,FIQ_Addr

ResetAddr      DCD    ResetInit
UnderfinedAddr DCD    Underfined
SWI_Addr       DCD    SoftwareInterrupt
PrefetchAddr   DCD    PrefetchAbort
DataAbortAddr  DCD    DataAbort
Nouse          DCD    0
IRQ_Addr       DCD    IRQ_Handler
FIQ_Addr       DCD    FIQ_Handler
```

3.5.3 中断及时钟节拍中断

编写中断服务程序代码比较简单, 编写 C 语言处理函数即可。关键在于把程序与芯片的相关中断源挂接, 使芯片在产生相应的中断后会调用相应的处理程序。这需要以下两个方面的工作:

- 增加汇编接口的支持
- 方法是在文件中 IRQ.S 适当位置添加如下所示的代码, 其中 xxx 替换为自己需要的

字符串。这样，汇编接口就完成了。

```
xxx_Handler    HANDLER    xxx_Exception
```

·初始化向量中断控制器

//中断外设初始化;

```
VICVectAddrX=(uint32) xxx_Handler;
```

```
VICVectCntlX=(0x20 | Y);
```

```
VICIntEnable=1<< Y;
```

3.5.4 编写应用程序

移植 $\mu\text{C}/\text{OS-II}$ 是为了在自己的系统使用 $\mu\text{C}/\text{OS-II}$ 。要在自己的系统中使用 $\mu\text{C}/\text{OS-II}$ 编写自己的应用程序就必须遵守 $\mu\text{C}/\text{OS-II}$ 的编程规范。主要包括主函数和用户任务，它们的关系如图 3.2。

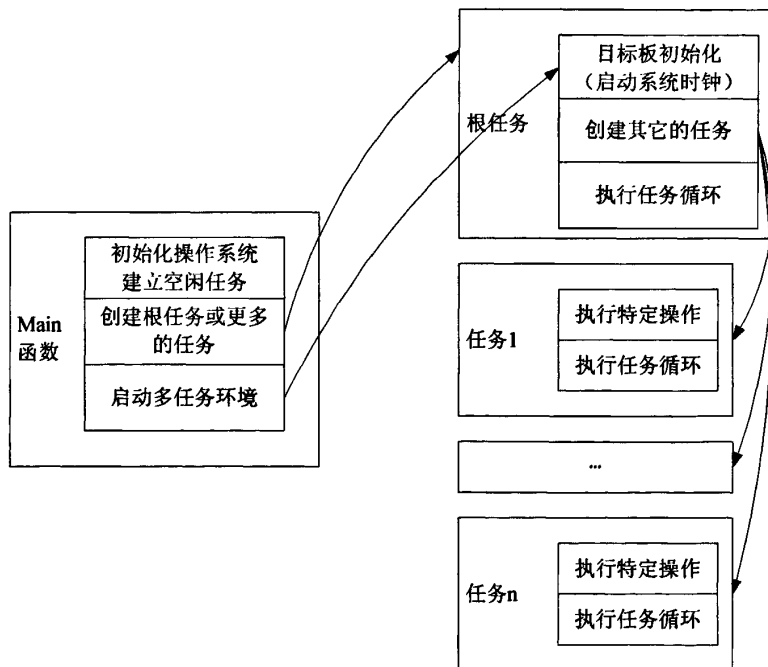


图 3.2 函数分配图

```
Void Task(void *pdata)
{
    pdata=pdata;    /* 避免编译警告 */
    Target Init();   /* 目标板初始化 */
    for(; ;)
    {
```

```

    OSTimeDly(OS_TICKS_PER_SEC / 50);
    if (Get Key() !=KEY1)
    {
        coutinue;
    }
    OSTimeDly(OS_TICKS_PER_SEC / 50);
    if (Get Key() !=KEY1)
    {
        coutinue;
    }
    OSTaskCreate(Task2, (void *)0,&TaskStk[TASK_STK_SIZE - 1],10);
    while(Get KET() !=0)
    {
        OSTimeDly(OS_TICKS_PER_SEC / 50);
    }
}
}

```

//任务功能：鸣叫两声，然后删除自己

```

void Task2(void *pdata)
{
    pdata=pdata;    /* 避免编译警告 */

    BeeMoo();        /* 使蜂鸣器鸣叫 */
    /* 延时 */
    OSTimeDly(OS_TICKS_PER_SEC / 8);
    BeeNoMoo();      /* 使蜂鸣器停止鸣叫 */

    OSTimeDly(OS_TICKS_PER_SEC / 4);

    BeeMoo();
    /* 延时 */
    OSTimeDly(OS_TICKS_PER_SEC / 8);
}

```

```
BeeNoMoo();    /* 使蜂鸣器停止鸣叫 */  
  
OSTaskDel(OS_PRIO_SELF);    /* 删除自己 */  
}
```

3.6 本章小结

本章首先介绍嵌入式 $\mu\text{C}/\text{OS-II}$ 操作系统的特点, 然后讲述了移植操作系统到硬件平台的规划和具体操作方法。

4 FLASH 存储器的磨损算法研究

存储器的管理是嵌入式操作系统关键的内容之一,在本系统中,使用了外部的 Flash 和 SDRAM 两种存储器。Flash 存储器是一种可在系统(In-System)进行电擦写,掉电后信息不丢失的存储器。它具有低功耗、大容量、擦写速度快、可整片或分扇区在系统编程(烧、写)、擦除等特点,并可由内部嵌入的算法完成对芯片的操作,因而在各种嵌入式系统中得到了广泛的应用。

4.1 Flash 存储器概述

随着嵌入式技术发展,NAND 型 Flash 的应用越来越广泛。由于 Flash 具有擦除次数有限、先擦后写的特点,会带来使用寿命有限的缺陷。因此,在不影响正常使用情况下如何延长 Flash 使用寿命,成为目前的一个研究热点,就已有的研究成果来看,主要是在 Flash 读写操作中引入存储单元擦写控制算法,平均各单元的擦写频率,可有效避免 Flash 存储单元因擦写频率分布不均造成的物理损坏。本章将就 Flash 在擦写数据频率高的场合中使用时,对擦写管理机制进行研究,并对目前磨损均衡算法进行改良,进一步提升其效果。

4.1.1 普通Flash 擦写管理方式

普通Flash 擦写管理方式中,首先建立了一张逻辑扇区到映射到物理扇区的文件分区表,并将文件分区表写入固定的物理扇区,这样能够快速读写数据。但是由于每次擦写数据都将修改文件分区表所在的扇区,导致文件分区表所在的扇区不断的被擦写,会在较短的时间内造成该物理扇区损坏,并丢失全部存储数据;此外,普通Flash 文件的擦写按照文件分区表来分配,对扇区的使用频率未作评估,文件分区表中靠前的物理扇区使用频率将大大超过靠后的部分,造成磨损不平衡,降低了Flash 的使用寿命^[21]。

4.1.2 磨损均衡管理方式

磨损均衡算法(Wear Leveling Algorithm) 是在工业中最常用的算法之一。将其运用于Flash管理中,将同一逻辑地址映射到不同的物理地址中,这样能够保证写入/擦除操作遍历整个Flash 扇区。该算法核心思想在于将新数据写到擦除次数最少的空闲区中^{[23][24]}。

损耗均衡算法要使各个物理块的擦除次数近似均衡,但如果过高地追求擦除次数均衡,会带来额外的擦除操作,降低读写速度。所以,损耗均衡的目标是在满足均衡条件的前提下,尽量减少总的擦除次数。给定擦除次数差值的上限,定义其为阈值,记为 TH_0 。若第 i 块的擦除次数记为 E_i ,则损耗均衡问题的数学描述为:

$$\min \sum E_i \text{ 或 } \min \sum \left[E_i - \frac{\sum E_i}{N} \right]^2 \quad \text{s.t. } \max \{E_i\} - \min \{E_i\} \leq TH$$

通过设定阈值可以控制损耗均衡的效果,在保证擦除次数差值不超过阈值的前提下,使总擦除次数尽量少,即分配到各个块的擦除次数尽量少,从而减小系统消耗。擦除操作主要在垃圾回收过程中进行。通过改进垃圾回收策略,可以减少总的擦除次数。

4.1.3 静态损耗均衡和动态损耗均衡

损耗均衡分为动态损耗均衡和静态损耗均衡。动态损耗均衡的工作范围只限于动态数据(经常被更新的数据)和未使用空间之间,在写入数据时实现。每次把要写入的数据写入擦除次数最少的块内,这样就达到了让Flash 各个块擦除次数近似均衡。静态损耗均衡主要针对很少被更新的数据和只读数据,它会强制搬移这些数据到擦除次数更多的块中去,这样可以在更大程度上增加使用寿命和可靠性。不过,代价就是影响数据写入的速度,并且可能造成不必要的额外擦除操作。所以,阈值控制主要针对静态损耗均衡,把最大擦除次数和最小擦除次数的差值控制在阈值之内,从而取得满意的损耗均衡效果。

4.2 磨损均衡算法的设计

4.2.1 磨损均衡算法介绍

对于Flash 中每个物理单元块操作后,都有一个状态值,如表4.1所示。

表4.1 单元块操作对应标记

状态	定义	描述
空闲块	Free	该块已被擦除,块中所有页面为空闲状态,可写入数据
干净块	Clean	该块已进行写入数据操作,并且写入数据完全有效
脏块	Dirty	对块进行写入数据操作失败,可对其进行擦除,成为空闲块
坏块	Bad	该块只能读取数据,无法写入数据,应当丢弃。

对应四种操作状态标记,分别设置四个状态块列,各列均采用链表方式对各物理块进行管理。这四种状态块列分别为空闲块列(Free_List)、干净块列(Clean_List)、脏块列(Dirty_List)和坏块列(Bad_List)。各块列中,按照擦除次数升次方式对各物理单元块进行连接,如图一所示。当有空闲单元块进入空闲块列时,通过其擦除次数将该块插入到链表中相应位置。出列时,按照链表顺序,从链表头取出空闲单元块。采用这种管理方式,能够保证每次所取的空闲单元块都是使用频率最小的单元块,这样能够动态磨损均衡管理。对其它块列采用同样方式进行管理。

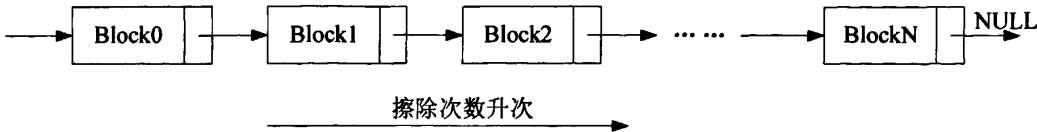


图4.1 块链内部结构

Flash 初始化时, 扫描所有物理块, 找出坏块, 将坏块逻辑号加入坏块管理表中, 然后对其余正常物理块进行初始化擦除操作, 更改其擦除次数。该初始化操作完成后, 所有单元块标记为Free 空闲块, 加至空闲块列中。写入新数据时, 从空闲块列中选取一个块, 将新数据按页顺序依次写入该块, 若写入操作完全有效, 则将该块标记更改为Clean, 放入干净块列中。若在写入过程中发生错误导致写入失败, 则复制已写入的有效信息至新空闲块后, 将该块标记为Dirty, 放入脏块列中。

4.2.2 垃圾回收策略

现有垃圾回收方法采用设置阈值法, 控制垃圾回收。该方法在空闲块列中空闲块数量小于阈值时, 启动脏块回收操作。这种方法每次只需从脏块列中回收一个脏块, 便能满足阈值需求。由于脏块采用按擦除次数方式排列, 每次取出脏块列中链表头的单元块, 按擦除次数插入到空闲块列中, 便完成整个回收操作。回收过程如图4.2所示。

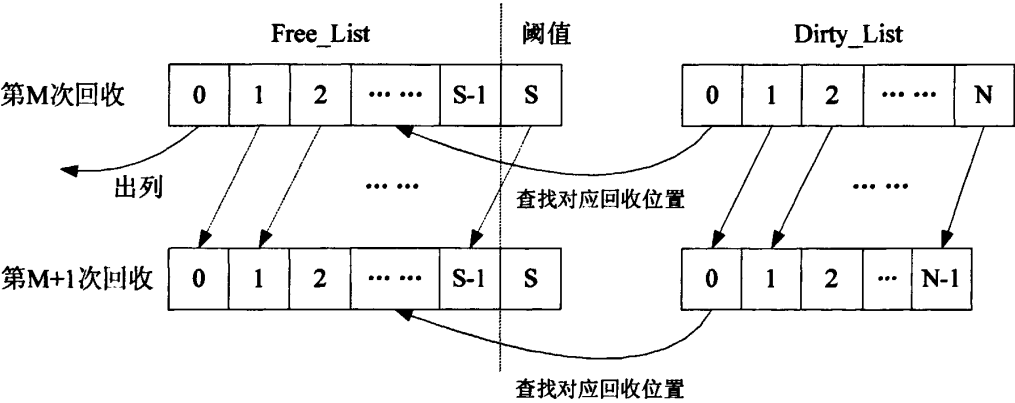


图4.2 空闲块数量小于阈值方式回收过程

垃圾回收可以在满足一定条件时进行,也可以作为后台线程在CPU 空闲时进行。垃圾回收的消耗主要有两方面: (1) 拷贝回收块内有效页的消耗; (2) 擦除回收块的消耗。由于擦除操作的消耗远大于拷贝有效页的消耗,所以从垃圾回收的角度看,垃圾回收应该释放尽可能多的可用空间,以减少擦除次数。垃圾回收的直接目的是释放尽可能多的存储空间。所以,垃圾回收应首先选择无效页最多的块,但这样会影响损耗均衡;从提高损耗均衡性能的角度看,垃圾回收应首先选择擦除次数最小的块,但这样可能会增加擦除次数,降

低写入数据的速度。所以说,垃圾回收和损耗均衡在一定程度上是矛盾的,尤其是对于静态数据块,从垃圾回收的角度来说,该块不应该被回收,因为这样做完全是增加系统消耗;但是,从损耗均衡的角度看,该块数据应该转移到擦除次数较大的块,以均衡各块的擦除次数。所以,在垃圾回收过程中应该综合考虑回收率和损耗均衡,在两者之间做一个折衷。

4.2.3 磨损均衡算法与简单Flash 算法比较

假设Flash 存储器中包含M个物理单元块,擦除总次数为N 次,每天擦除X 次。采用两种算法分别计算其使用寿命。

$$\cdot \text{简单Flash算法 预期使用寿命}(D1) = \frac{\text{擦除总次数}}{\text{每天擦除次数}} = \frac{M}{X}(\text{天})$$

$$\cdot \text{使用磨损均衡算法 预期使用寿命}(D2) = \frac{\text{单元块数量} \times \text{擦除总次数}}{\text{每天擦除次数}} = N \frac{M}{X}(\text{天})$$

由于在Flash中磨损均衡算法存在阈值,它是非绝对的控制算法,所以对D2取平均值近似逼近实际使用寿命,即 $(D2) = \frac{N}{2} \times \frac{M}{X}(\text{天})$

对比两种方式,很明显的看出使用磨损均衡算法后,Flash预期使用寿命增加N/2 倍。

4.3 对磨损均衡算法改进

4.3.1 基于全部脏块列的垃圾回收方式

原垃圾回收方法实质在于达到启动垃圾回收的临界条件后,回收一个脏块进行写入操作。它的优点在于能充分利用Flash 存储空间,存储更多数据。但它的缺点是当第一次启动垃圾回收后,每次写入数据操作都采先擦除一块再写入一块的方式,由于Flash 擦除时间远大于写入时间,在输入数据速率很高的情况下,会产生数据接收速率远不及数据输入速率,输入数据堆积的情况,这会对写数据带来严重后果。为避免这种情况发生,需保证有尽量多的空闲单元块能够接收数据,为此,需要对该垃圾回收方式进行改进。改进后的回收方式需预先设置一个空闲块列中单元块数量下限值F,该值应尽力满足为下次更新操作提供所需的空闲块。但如果该值选取过大,即空闲块数过多,会减少Flash 中可利用内存。若改值过小,则不能解决现有问题。改进后垃圾回收方式如下:当空闲块列为空后,完全释放脏块列,其步骤如下:擦除脏块,更改擦除次数值,更改块状态标记为FREE,修改其链表,将该块放入空闲块列。若脏块列块数值小于回收块值F,则还需从干净块列中回收剩余数目块。对干净块列回收过程与脏块回收类似,这种方式能保证每次进行回收的块是整个块列中数据使用频率最低的单元块。这样能将擦除有效数据对整个存储系统带来的影响降到最低。整个回收过程如图4.3所示。

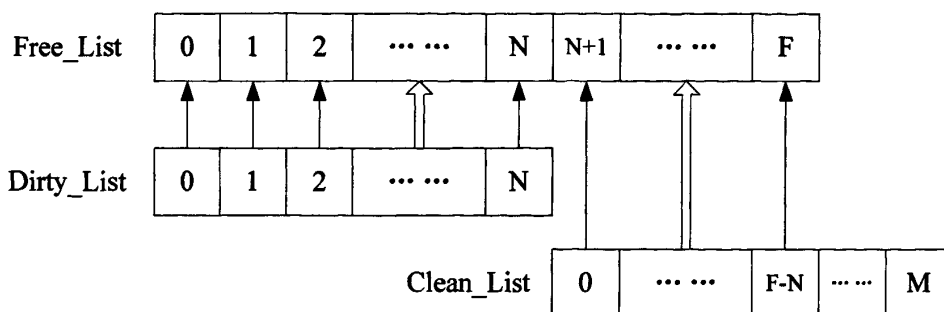


图4.3 空闲块列为空后回收过程

4.3.2 基于磨损均衡算法对静态文件管理方式进行改进

当存储静态文件时，该存储块很长一段时间不会被擦除，这样会影响整个Flash 各单元块磨损均衡效果。这种情形可通过控制其擦除次数的方式进行改善。当垃圾回收被触发时，读取干净块列中所有单元块擦除次数NumErase，找出整个Flash中单元块最大擦除次数，记为MaxErase。当各块擦除次数与最大擦除次数差值大于阈值时，即

$|\text{NumErase} - \text{MaxErase}| > \text{阈值}$ ，便意味着该单元块中存放着静态文件，使用频率低，这时便启动静态文件单元块回收操作：将符合回收条件的所有块中有效信息以及链表复制到空闲块中，释放该块链表，将该块按擦除次数插入脏块队列中^[22]。这种回收方式本质是在静态文件块与擦除次数最多的块之间进行文件搬移，将静态文件存放在擦除频率最高的块中，以此减少对擦除频率最高块调用次数。通过这种回收方式，能保证flash 中所有物理块擦除次数在数量级上达到平衡。该回收条件中阈值选择非常重要，若阈值选取过大，可能导致需释放的静态文件存储块数量大于空闲块数量，有效数据无法全部接收；若选取阈值过小，会导致静态文件存储块不能及时释放，其擦除次数与动态文件存储块擦除次数不能在数量级上达到平衡。回收过程如图4.4、4.5所示。

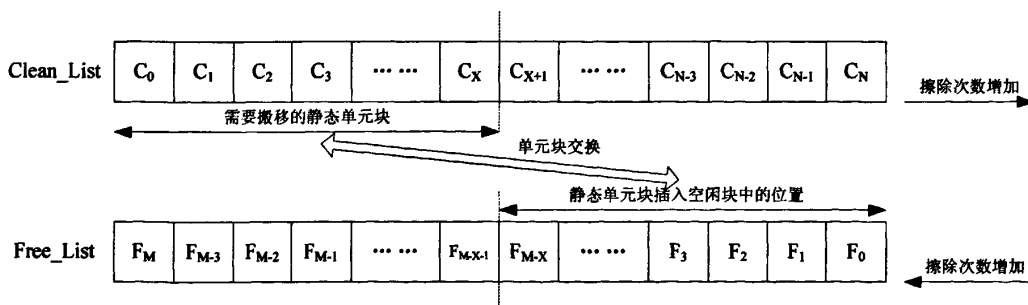


图4.4 静态文件回收前各列存储结构

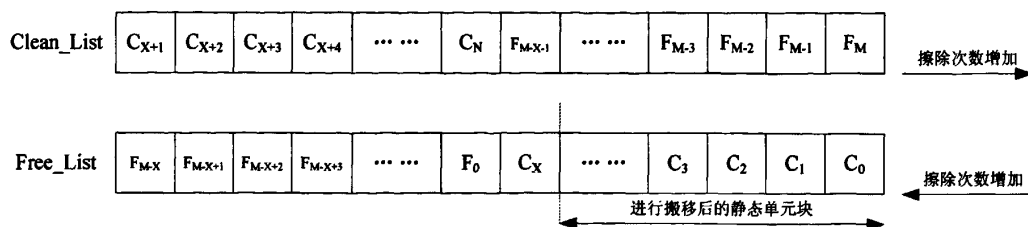


图4.5 静态文件回收后各列存储结构

4.4 本章小结

本章针对Flash型的存储器特性,对现有的磨损均衡算法中垃圾回收和静态文件单元管理方式进行改进,提高Flash存储数据效率,同时针对不同的文件类型,分别进行对应的磨损均衡管理,从而有效地延长整个Flash 的预期使用寿命。通过对该算法的改进,相信Flash在嵌入式系统中的应用会更加有效。

5 系统应用开发设计

5.1 显示及键盘电路设计

真空荧光显示屏VFD (Vacuum Fluorescent Display) 是从真空电子管发展而来的显示器件,它的基础特性与电子管的工作特点基本相同。由发射电子的阴极(直热式,统称灯丝)、加速控制电子流的栅极、玻璃基板上印上电极和荧光粉的阳极及栅网和玻盖构成。它利用电子撞击荧光粉,使荧光粉发光,是一种自身发光显示器件。VFD根据结构一般可分为二极管和三极管两种;根据显示内容可分为:数字显示、字符显示、图案显示、点阵显示;根据驱动方式可分为:静态驱动(直流)和动态驱动(脉冲)。显示发光形式有点阵式和固定图形、文字式等。由于它可以做多色彩显示,亮度高,又可以用低电压来驱动,易与集成电路配套,所以被广泛应用在家用电器、办公自动化设备、工业仪器仪表及汽车等各种领域中。在音箱面板上主要用来显示调节音量的高低状态,显示声音信号的强弱高低。

显示控制器采用的是HOLTEK的HT16511。HT16511是一个1/8到1/16占空比驱动的VFD显示(真空荧光显示)控制器。包含12根段输出线、8根栅输出线、8根段/栅输出线、显示存贮器、控制电路、键扫描电路。串行数据通过一个三线串行接口输入到HT16511。该电路是单片机的理想外围器件。多重显示方式:12段×16位~20段×8位。HT16511的作用是透过微控制器的指令控制栅极,而对真空荧光显示器所对应显示点进行控制。主要应用在具有仪表控制功能的消费产品以及工业测量设备。

图5.1是系统硬件电路框图,主要包括LPC2220 与驱动芯片HT16511 的连接。HT16511 与VFD 显示器的连接,使用HT16511 外扩的12×4 的按键扫描电路以及指示灯和开关电路。

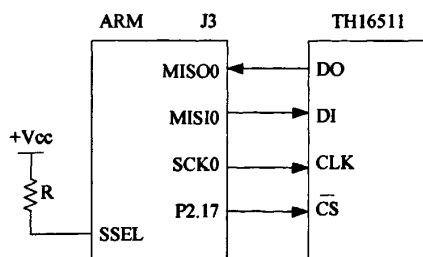


图5.1 HT16511与开发板连接结构图

从图5.1中可以看出,读/写数据时必须置CS为低;写数据(DI)过程中,在时钟(CLK)的下降沿往数据线上送数,送完数后需要将CLK置高;在读数据时,需要先将CLK置高一段时间,然后再拉低一段时间,才开始从数据线上读取数据。

驱动程序主要包括显示驱动和按键驱动。显示驱动程序主要包括向HT16511 写命令和写控制显示的数据;而按键驱动程序则主要是从HT16511 读取键值,再拼接为有效键值。由系统流程设计有,应先设置显示模式,然后设置读/写数据模式和读/写数据的起始地址,再发送数据,最后设置显示亮度。每发送一个命令或数据之后都需要一段延时。

显示驱动程序为:

```
extern void update_disp (unsigned char 3 disp_data) {
    unsigned char k ;
    gpio_set ( HTCS );
    delay(50);          // 延时50μs
    gpio_clr ( HTCS );
    write_char (0x0f); // 发送0x0f,设定显示模式16 字12 段
    gpio_set ( HTCS );
    delay(100);
    gpio_clr ( HTCS );
    write_char (0x40); // 发送0x40,数据模式:一般模式、地址自增、向显存写数据
    gpio_set ( HTCS );
    delay(100);
    gpio_clr ( HTCS );
    write_char (0xc0); // 发送0xc0,设置地址为00 H
    for (k = 0 ;k < 36 ;k ++ ) {
        write_char ( 3 (k + disp_data) ); // 向显示RAM发送数据
        delay(50);
    }
    gpio_set ( HTCS );
    delay(100);
    gpio_clr ( HTCS );
    write_char (0x8f); // 发送0x8f,打开显示
    return ;
}
```

按键驱动程序为:

```
extern void read_key(void) {
    gpio_set ( HTCS );
    delay(50);
    gpio_clr ( HTCS );
```

```

write_char (0x46); // 发送0x46,读键模式
datum[ 0 ] = read_char (); // 由于键值在HT16511 中是以6 个字节的
                             形式存放的,
                             每次读取时都必须把6 个字节的数据全部读出,
                             否则会
                             造成下次读数出错

delay(10);
datum[ 1 ] = read_char ();
delay(10);
datum[ 2 ] = read_char ();
delay(10);
datum[ 3 ] = read_char ();
delay(10);
datum[ 4 ] = read_char ();
delay(10);
datum[ 5 ] = read_char ();
delay(10);
gpio_set ( HTCS );
for (i = 5 ;i >= 0 ;i22) {
data|= datum[ i ];
if (i != 0)
data = data << 8 ;
}
return data ;
}

```

HT16511 的驱动模块适用于各种类型的微处理器。具有实现容易、性价比高、显示精度高、显示对比度高、反应速度快等优点。因此,该设计适用于很多要求较高的工业应用场合。在应用中,只需要将HTCS、HTCLK、HTDO、HTDI 定义为微处理器的一般双向I/O口。根据使用的VFD 真值表确定显示数据或字母的字形码。用一个数组disp[]来存储这些字形码,并使用上述的update_disp() 函数将数组发送到HT16511 驱动芯片,即可完成显示。对于按键扫描,使用read_key() 函数已经读到按键值,并放在data寄存器中。

5.2 USB通信接口设计

CY7C68001 是用来连接微处理器或DSP 的DMA 从装置,其内部不含微处理器;集成有USB 2.0 收发器(物理层)、USB 2.0 串行接口引擎SIE(链路层,实现底层通信协

议)、4 kB的FIFO和电压调节器、锁相环;支持高速(480 Mb/s)或全速(12 Mb/s)传输;3.3V操作电压,24 MHz外部振荡频率;可以选择8位或16位总线方式;具有同步与异步的FIFO接口;他可以向用户提供足够的端口、缓冲区和传输速度;提供USB 2.0协议要求的全部4种传输方式(控制传输、中断传输、批量传输和同步传输),可以满足用户对各种类型数据传输的需求。片上的串行接口处理机(SIE)能完成大部分的USB协议操作,使用户可以摆脱复杂的协议细节,简化了用户配置代码,加快了程序开发过程^{[26][27]}。

5.2.1 开发板对CY7C68001的操作

CY7C68001有2个外部接口:

- (1)命令接口:用来访问CY7C68001寄存器、Endpoint0缓冲器及描述表;
- (2)FIFO数据接口:用来访问4个1kB的FIFO中的数据。

这两个外部接口均可以通过同步或是异步的方式进行访问。在本设计中均采用异步的方式进行访问。

系统中CY7C68001作为开发板的外设,占用开发板的外设PACK的I/O空间,地址为0x82000007~0x82000000。CY7C68001的地址线DIDOADR[2:0]为100B时,选中CY7C68001的命令口。通过CY7C68001的命令口,可以访问37个寄存器、Endpoint0缓冲器(64个字节FIFO)和描述表(500个字节FIFO)等。写入命令口的内容称为命令字,命令字包括要寻址的寄存器的子地址,或是要写入寄存器数据的高4位或低4位。读命令口必须在给命令口写入命令字之后,所读出的为相应寄存器的8位数据。在这里,对寄存器的操作均以8位数据宽度进行。

开发板对CY7C68001寄存器的写操作过程:

- 将开发板对I/O空间访问的等待周期设为7;
- 通过检查系统状态寄存器0位2(USBRDY)等待CY7C68001的Ready线变高;
- 寻址CY7C68001命令口(最低3位地址必须为100B),将10xx, xxxxB写命令字写入命令寄存器(xx, xxxxB代表需寻址的寄存器的子地址);
- 通过检查系统状态寄存器0位2等待CY7C68001的Ready线变高;
- 寻址CY7C68001命令口(最低3位地址必须为100B),将0xxx, xxHUB数据高4位命令字写入命令寄存器(xxHUB代表要写入数据的高4位);
- 通过检查系统状态寄存器2位2等待CY7C68001的Ready线变高;
- 寻址CY7C68001命令口(最低3位必须位100B),将0xxx, xxHLB数据低4位命令字写入命令寄存器(xxHLB代表要写入数据的低4位)。

开发板对CY7C68001寄存器的读操作过程:

- 开发板对I/O空间访问的等待周期设为7;
- 通过检查系统状态寄存器0位2(USBRDY)等待CY7C68001的Ready线变高;

·寻址CY7C68001命令口（最低3位地址必须为100B），将11xx, xxxxB读命令字写入命令寄存器；

·等CY7C68001申请PACK的外部中段EINT0；

·寻址CY7C68001命令口（最低3位地址必须为100B），读CY7C68001命令口。

FIFO数据接口的操作对数据接口的写操作作用到了SX2_FifoWriteSingle()函数，负责向USB的数据缓冲区中写数据，对数据接口读操作作用到了函数SX2_FifoReadSingle()；另外，程序中用到了中断函数int_isr()，在读周期，主机申请读寄存器的值，此时产生中断，从SX2读进数据。

5.2.2 USB编程

CY7C68001有EEPROM自举和通过DSP自举2种自举方式，在本系统中两种方式均可以，一般采用EEPROM来进行USB初始化，在EEPROM中初始化的顺序如下：

0xC4通知EZ_USB SX2有一个有效的EEPROM存在；

IFCONFIG设置IFCONFIG寄存器，其函数为Write_SX2reg()；

POLAR设置各个信号的极性，其中SLOW,ALRD,SLWR只能由EEPROM来配置；

Descriptor是否从EEPROM中装入初始化表。当值为0xc4时装入，反之等待Master装入；

Descriptor Length这是两个字节的空，指示EEPROM中Descriptor的长度；

Descriptor Data描述表的内容，其函数为Load_descriptor()。

USB的编程主要分为2个部分，一是HOST方面驱动程序的编写，一是有关ARM与USB数据和命令的交换。

HOST端需要编写USB设备驱动程序和应用程序。

USB设备驱动程序主要是通过调用微软的USB.D.SYS来实现PC机与USB总线的数据交换。因而USB设备驱动程序主要完成的功能如下：

- (1)对相应的USB设备建立设备驱动对象并完成对USB设备的初始化；
- (2)完成USB设备的即插即用功能；
- (3)完成USB设备电源的管理；
- (4)实现对USB.D.SYS的调用，完成对USB设备的控制与数据的交换；
- (5)实现对USB通讯错误的处理。

USB设备应用程序主要用到以下几个函数：

OpenDriver() 打开驱动程序；

CloseDriver() 关闭驱动程序；

Sx2 GetDeviceDesc() 获取描述表；

Sx2 GetconfigDesc()获取配置说明；

Sx2 GetPipeInfo() 获取接口说明;

Sx2 SendVendorReq() 向USB发送请求;

Sx2 BulkdataTrans() 向PC和USB间传递数据。

开发板是通过其外设PACK上的I/O口来完成CY7C68001的数据交换的。这部分程序中主要完成以下功能:

- (1)完成对命令接口的读与写;
- (2)完成对FIFO接口的读与写;
- (3)完成开发板对USB的自举;
- (4)完成开发板与USB的控制命令的传输;
- (5)完成开发板与USB的各种方式的数据传输。

5.3 LAN通信设计

本课题中以太网设计采用网线交叉连接方法,将开发板与上位PC进行连接,实现嵌入式Web服务器。运用10/100M的网口来下载程序、ARM板与上位机通信。TCP/IP协议和以太网协议是使用最广泛的通讯协议。以太网底层协议的实现由以太网控制器来负责,CS8900是一款比较常用的10Mbps嵌入式以太网控制芯片。

5.3.1 TCP/IP协议的五层参考模型

网络协议是系统与系统之间进行通讯的接口。网络是一个信息交换的场所,所有接入网络的计算机都可以通过彼此之间的物理连设备进行信息交换,这种物理设备包括最常见的电缆、光缆、无线WAP和微波等,但是单纯拥有这些物理设备并不能实现信息的交换,这就好像人类的身体不能缺少大脑的支配一样,信息交换还要具备软件环境,这种“软件环境”是人类实现规定好的一些规则,被称作“协议”,有了协议,不同的计算机可以遵照相同的协议使用物理设备,并且不会造成相互之间的“不理解”。协议是连入网络的计算机都要遵循的一定的技术规范,关于硬件、软件和端口等的技术规范。TCP/IP(Transmission Control/Internet Protocol)是最常见的一种网络协议族,是Internet最基本的协议,简单地说,就是由底层的IP协议和TCP协议组成的。确切地说,TCP/IP协议是一组包括TCP协议和IP协议,UDP(User Datagram Protocol)协议、ICMP(Internet Control Message Protocol)协议和其他一些协议的协议组^[30]。

TCP/IP协议并不完全符合OSI的七层参考模型^[31]。传统的开放式系统互连参考模型,是一种通信协议的7层抽象的参考模型,其中每一层执行某一特定任务。该模型的目的是使各种硬件在相同的层次上相互通信。这7层是:物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。而TCP/IP软件分成五个概念层次,它主

要是建立在第五层(即硬件层)上的四个软件层。图5.2给出了这些概念性层次之间传输的数据形式。本系统的TCP/IP协议栈实现了四个基本协议:ARP地址解析协议,IP网际互联协议,ICMP网络控制报文协议,TCP传输控制协议。每一层都呼叫它的下一层所提供的网络来完成自己的需求。如表5.1所示。

表5.1 TCP/IP层次结构

分层	常见协议
应用层	FTP、HTTP等
传输层	TCP和UDP
网络层	IP、ICMP、ARP等
链路层	Ethernet MAC

(1)应用层(Application Layer):在这个最高层中,用户调用应用程序通过TCP/IP互联网来访问可用的服务,主要协议有Http ,Telnet, FTP和e-mail等.在本系统中,应用层由用户实现HTTP协议以实现Web服务器。

(2)传输层(Transport Layer): 在此层中，它提供了节点间的数据传送服务，如传输控制协议（TCP）、用户数据报协议（UDP）等，TCP和UDP给数据包加入传输数据并把它传输到下一层中，这一层负责传送数据，并且确定数据已被送达并接收。

(3)网络层(Network Layer): 网络层的主要协议是网际协议IP、地址解析协议ARP和网络控制报文协议ICMP。IP协议是整个TCP/IP协议族的核心，提供传输层数据分片、封装和路由功能以及控制报文的生成与传送服务，提供了一种不可靠的、尽最大努力传送的、无连接分解的传送服务。ARP协议主要是实现IP地址到对应的物理地址的解析，ICMP协议用来与其它主机或路由器交换错误报文和一些控制信息。

(4)链路层(Data Link Layer): 链路层是TCP/IP协议族的最底层，它负责接收IP数据报并把数据报通过选定的网络接口发送出去，或者从网络上接收物理帧，抽出IP数据报，交给IP层。在本设计中这部分功能由控制卡芯片CS8900来实现。

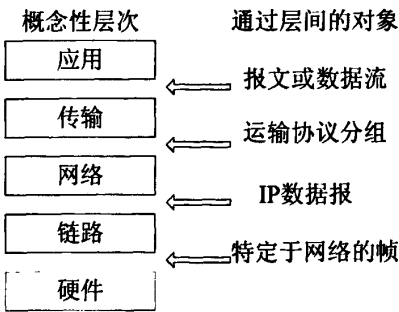


图5.2 TCP/IP软件概念性层次和各层之间传输对象的形式

5.3.2 以太网控制器CS8900工作原理

TCP/IP协议栈嵌入在 μ C/OS-II系统中,但它并不提供底层网络驱动。在本系统中, μ C/OS-II首先要初始化网络设备,它主要处理TCP/IP协议的网络层和传输层,其应用层交付用户处理,链路层部分主要有CS8900完成。因此,为了实现WEB SERVER,还需要添加三个模块:网卡芯片CS8900的驱动^[32]。

CS8900是CIRRUSLOGIC公司生产的低功耗、性能优越的16 位以太网控制器,功能强大。该芯片的突出特点是使用灵活,其物理层接口、数据传输模式和工作模式等都能根据需要而动态调整,通过内部寄存器的设置来适应不同的应用环境。

CS8900A 内部功能模块主要是802.3 介质访问控制块(MAC)。802.3 介质访问控制块支持全双工操作,完全依照IEEE 802.3 以太网标准(ISO/IEC880223, 1993),它负责处理有关以太网数据帧的发送和接收,包括:冲突检测、帧头的产生和检测、CRC校验码的生成和验证。通过对发送控制寄存器(TxCMD) 的初始化配置,MAC 能自动完成帧的冲突后重传。如果帧的数据部分少于46个字节,它能生成填充字段使数据帧达到802.3 所要求的最短长度。

它的主要特点如下:

- 符合IEEE 802.3 以太网标准,并带有ISA 接口;
- 片内4K 字节RAM;
- 适用于I/O 操作模式,存储器操作模式和DMA 操作模式;
- 带有传送、接收低通滤波的10Base-T连接端口;
- 支持10Base2,10Base5 和10Base-F的AUI接口;
- 自动生成报头,自动进行CRC 检验,冲突后自动重发;
- 最大电流消耗为55mA (5V 电源);
- 全双工操作;
- 支持外部 EEPROM。

另外,要实现CS8900与主机之间的数据通讯,在电路设计时可根据具体情况灵活选择合适的数据传输模式。CS8900支持的传输模式有I/O 模式和Memory 模式,另外还有DMA 模式。其中,I/O 模式访问CS8900存储区的缺省模式,比较简单易用。

CS8900基本工作原理是:在收到由主机发来的数据报(从目的地址域到数据域,如图5.3所示)后,侦听网络线路。如果线路忙,它就等到线路空闲为止,否则,立即发送该数据帧。发送过程中,首先,它添加以太网帧头(包括先导字段和帧开始标志),然后,生成CRC 校验码,最后,将此数据帧发送到以太网上。接收时,它将从以太网收到的数据帧在经过解码、去掉帧头和地址检验等步骤后缓存在片内。在CRC 校验通过后,它会根据初始化配置情况,通知主机CS8900收到了数据帧^[33]。

7	1	2或6	2或6	2	0-1500	0-46	4
先导字段	帧开始标志	目的地址	源地址	数据字段长度	数据	填充字段	检验和
Preamble Delimiter	Star-of-Frame Delimiter	Destination Address	Source Address	Length Field	LLC Data	Pad Bits	CRC CheckSum

图5.3 802.3帧格式

(1)CS8900 的驱动

CS8900 主要是用来接受和发送网络数据,它的驱动程序主要有三个部分:1 网卡芯片的上电初始化,包括网卡上电复位,设定网卡物理地址,设定收发缓冲区位置和大小等;2 完成数据发送的函数的编写;3 完成网络数据接收的函数的编写。这些底层网络设备驱动程序与 TCP/IP 协议栈是通过收发缓冲区的首地址和收发的数据长度进行接口。

CS8900 以太网控制器是以存储器为核心,本地和远端控制器并发操作.这种体系结构满足了数据带宽的需要。

(2)应用层基于 HTTP 协议的 WEB SERVER 的实现

超文本传输协议 HTTP 是目前通过 Internet 进行信息交换最主要的方式.它是建立在请求/响应模型上的,首先客户进程建立一条同服务器进程的 TCP 连接,然后发出请求并读取服务器进程的响应,服务器进程关闭连接表示本次响应结束,它返回的文件通常含有指向其他服务器上文件的指针(超文本链接).所以 Web 服务器可以通过超文本链接“指向”另一 Web 服务器。每一次 HTTP 的请求/响应都需要建立一条新的 TCP 链接,因此一个包含 HTML 内容和图片的页面将需要建立多次 TCP 链接。

5.3.3 μ C/OS-II下的Socket编程

套接字(Socket)是TCP/IP 网络通讯的基本构建模块,它实际是一个可以关联名字的通信端点。不论是客户机工作方式还是服务器工作方式,在进行通讯前必须创建各自的套接字并建立连接。Socket套接字有两种类型:流套接字(SOCK_STREAM)和数据报套接字(SOCK_DGRAM)。

(1)流套接字(SOCK_STREAM)

流套接字可以提供可靠的、面向连接的通讯流。如果通过流套接字发送了顺序的数据:“a”、“b”。那么数据到达远程的顺序也是“a”、“b”。它使用了TCP(Transmission Control Protocol)协议,保证了数据传输是正确的,并且是顺序的。TCP是经常出现的TCP/IP的前半部分。IP代表Internet Protocol, IP只处理网络路由。其工作流程如图5.4。

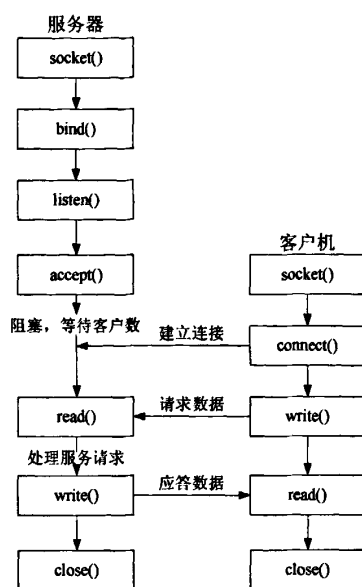


图5.4 面向连接的Socket的工作流程

(2)数据报套接字(SOCK_DGRAM)

数据报套接字定义了一种无连接的服务，数据通过相互独立的报文进行传输，是无序的，并且不保证可靠，无差错。原始套接字允许对低层协议如IP或ICMP直接访问，主要用于新的网络协议实现的测试等。

数据报套接字也使用IP，但是它不使用TCP，它使用使用者数据报协议UDP(User Datagram Protocol)。UDP不像流套接字那样维护一个打开的连接，只需要把数据打成一个包，把远程的IP贴上去，然后把这个包发送出去。这个过程不需要建立连接。其工作流程如图5.5。

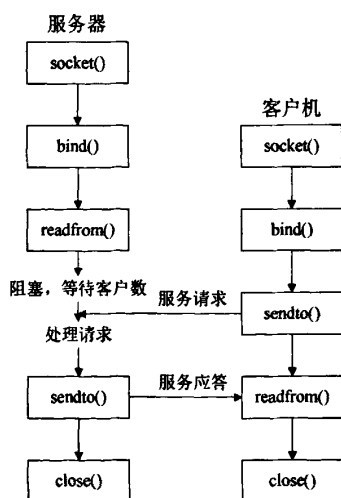


图5.5 无连接的socket连接

5.4 系统软件设计

系统的工作状态分为0态和1态两种：0态主要完成器件初始化、按键处理和显示等；1态为正常的工作状态，完成按键、显示、存储、通信等各项功能。

系统开机工作后，首先进入0态工作，可以按键控制由0态进入1态。

5.4.1 0态任务

(1)任务1，系统启动以后，首先进入任务1，完成：

- 开发板上资源模块初始化，包括：LAN、定时器、RTC等
- HT16511初始化
- CY7C68001初始化
- 完成0态显示：全屏显示2s，VICTOR显示2s，进入0态基本显示(A区显示初值+00，B区显示键值初值0)
- 建立任务2(0态的按键处理)、任务3(0态的显示处理)

(2)任务2，完成0态工作时的按键读取及处理：

- 定时查询的方法，1s查询10次
- 读取键值并做去抖处理
- 判断按键类型
- 启动任务3进行显示刷新，或进入1态工作

(3)任务3，完成0态工作时的VFD显示。

总之，0态工作时，完成上电初始化(任务1)以后，系统在任务2、任务3之间进行调度与切换。编程时，主要处理好这两个任务间的同步与通信，确保：①按键每秒钟查询10次；②根据任务2最新的判断结果，及时刷新VFD显示。

0态人机交互界面如图5.6所示。按键操作及显示如表5.2。

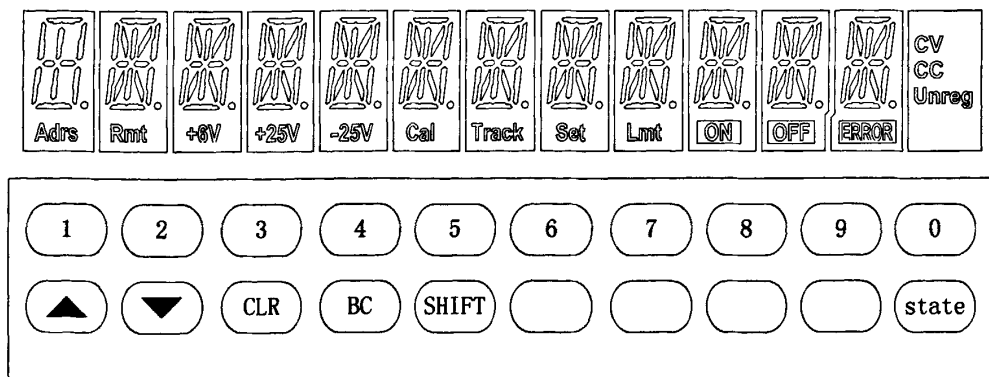


图5.6 0态界面

表5.2 0态按键操作及显示处理要求

按键	键位	键类型	操作及显示
0#~9#	KEY0~KEY9	单击键	B 区键值显示 0~9, 初值 0
UP/DN	KEY10/KEY11	连击键	A 区增减值显示-99~+99, 初值+00
CLR	KEY12	延迟键	A 区、B 区值清零
BC	KEY13	复合键	单击时 B 区键值显示 B 延迟时 B 区键值显示 C ·按此键, 设置键的 SHIFT 态, 'Unrog'标志显示或灭显
SHIFT	KEY14	单击键	·当 SHIFT 态, 'Unrog'标志显示时 ·再按 0#~9#键, B 区显示键值 S0~S9, 'Unrog'标志灭显 ·再按其它键, 无操作显示, 'Unrog'标志灭显
STATE	K19	单击键	系统工作由 0 状态进入 1 状态

单击键: 按键单次有效, 按键则只执行一次相关操作, 若键不释放则无任何操作, 直至键释放。

连击键: 按键连续有效, 若键释放不超过1S, 则只执行一次相关操作; 若键长时按, 则连续执行相同操作, 先是慢速的执行500mS/次, 3S后再快速的执行200mS/次, 键释放后重新开始。

延迟键: 按键延迟单次有效, 键按下2S后, 则只执行一次相关操作, 直至键释放。

复合键: 单击键与延迟键结合, 若键释放不超过1S为单击, 则只执行一次相关操作; 键长按2S后为延迟, 则只执行一次不同的相关操作。

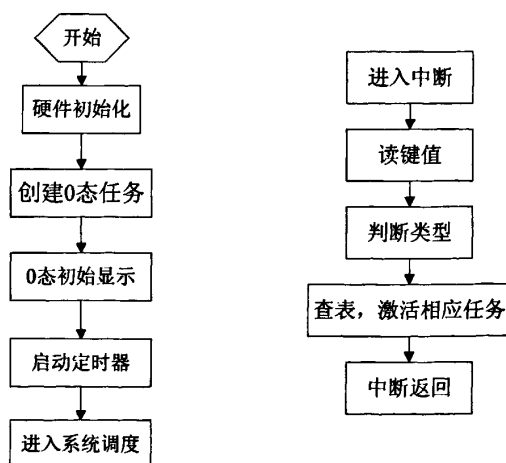


图5.7 0态系统流程图

5.4.2 1态任务

这里，利用LPC2220内部的定时器进行按秒计时，定时时间到，便读取测量值并再启动电压转换。

(1)任务1，1态工作时的按键处理。

·定时查询的方法，1s查询10

·读取键值并做去抖处理

(2)任务2，1态工作时的VFD显示。

正常显示时，A区显示时间，并根据RTC计时进行按秒刷新，格式为：

XX(时):XX(分):XX(秒)；

B区显示电压测量值，根据定时器的定时按秒刷新。

(3)任务3，记录模式时，将时间和电压测量值存入电子硬盘，按RTC计时每秒记录一个数据点。此时，VFD还是正常显示测量值。

(4)任务4，将电子硬盘的记录文件保存到SD卡，电压测量等工作不停止。

(5)任务5，通过USB与PC通信。

(6)任务6，通过LAN与PC通信。

1态是系统的正常工作状态，要完成的任务较多，编程时要注意理清头绪。以上只是初步的方案，在开发过程中，会根据遇到的实际情况对任务进行更为合理的安排。

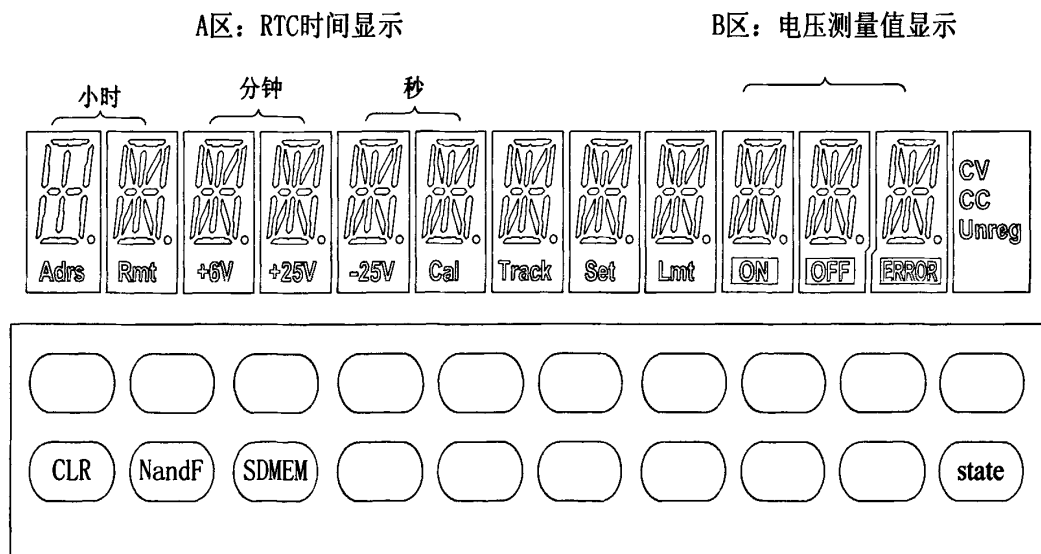


图5.8 1态人机交互界面

表5.3 1态按键操作及功能要求

按键	键位	键类型	操作及显示
CLR	K10	单击键	在非记录模式，A 区时间和 B 区电压值清零 ·系统进入/退出记录模式，‘Unreg’标志显示或灭显 ·A 区 RTC 时间显示、B 区电压值显示
NandF	K11	单击键	·在记录模式，将相应 RTC 时间和电压测量值，以 FAT 文件格式存入电子硬盘（NandF LASH 256M），按 RTC 计时每秒记录 1 个数据点 ·在非记录模式下，将电子硬盘的记录文件保存到 SD 卡
SDMEM	K12	单击键	·在保存过程中显示‘SDMEM’，至少维持显示 1S ·保存结束，返回到 A 区时间和 B 区电压值显示 ·如果记录文件不存在，则显示‘MEM NO’，维持显示 1S
STATE	K19	单击键	系统工作由 1 状态进入 0 状态

5.5 系统功能测试及试验结果

5.5.1 平台整体功能测试

通过两个程序对本系统进行测试：State1和Web_Server。State1是对硬件平台进行初始化，显示电压在VFD上并将数据记录到电子硬盘上。Web_Server是将采集到的结果通过网线互联的方式以网页形式显示在上位PC上。平台整体实物如图5.9。

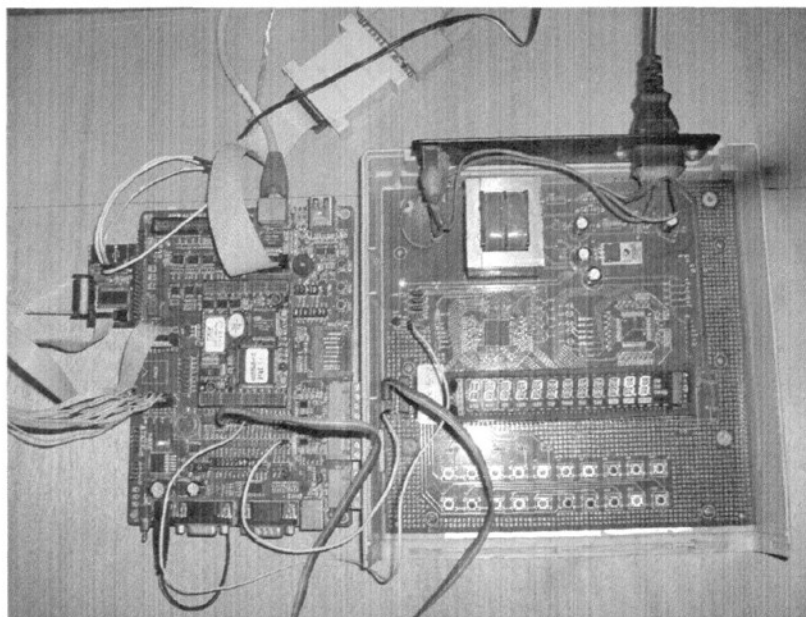


图5.9 平台整体实物

5.5.2 显示模块测试

在本系统中，高精度测量模块采用某仪表公司生产的一款高精度产品，它具有直流电压、电流，交流电压、电流，电阻等测量功能。显示模块通过控制平台显示经过处理的高精度测量模块采集的电压。

显示控制器HT16511的RAM分2个区域进行读写操作，一部分向嵌入式微处理器发送读操作命令，处理器得到指令后向HT16511的RAM发送写指令，显示高精度模块的测量数据和时间。显示模块实物如图5.12。

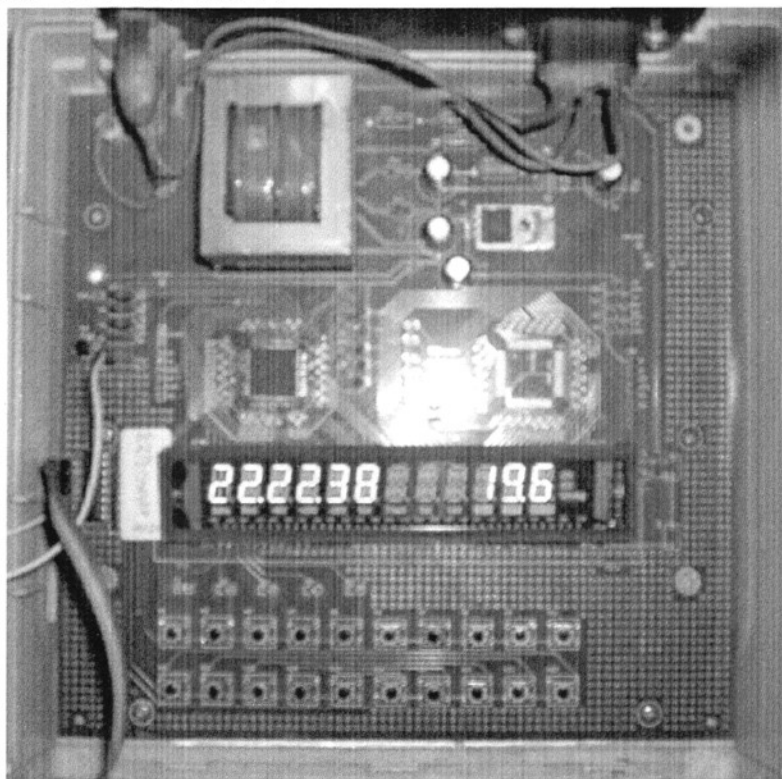


图5.12 硬件显示控制电路实物

5.5.3 LAN通信

以太网设计采用网线交叉连接方法，将开发板与上位PC进行连接，实现嵌入式Web服务器。图5.13是上位PC的LAN通信界面。



图5.13 LAN通信上位PC界面

5.6 本章小结

本章介绍各个模块硬件电路的功能，VFD连接到开发板的GPIO口模拟SPI总线，将结果显示到VFD屏上；开发板通过USB模块和LAN模块，将数据实时传输到外部设备或者PC机上。通过软件调试验证了平台具有的各项功能，给出实物加以论证，并且为后续开发提供了一个基础平台。

6结论

6.1 总结

本课题的研究重点是基于ARM7和 μ C/OS-II操作系统的嵌入式应用开发，经过一年多的课题研究，主要取得了以下成果：

(1)硬件上，设计了以LPC2220为核心的嵌入式硬件开发平台和部分外部功能模块，包括VFD模块、USB模块、LAN模块、显示模块、SPI接口、SD卡接口，系统通过SPI接口读取高精度测量模块的数据，并将数据显示到VFD上和实时传输到外部设备或者PC机上。并设计了PCB板，已调试成功。

(2)软件上，成功移植嵌入式 μ C/OS-II操作系统到ARM7微处理器上，根据开发的硬件平台，对 μ C/OS-II操作系统进行了剪裁和配置。认真分析USB协议和TCP/IP协议，根据应用设计开发了显示的程序、USB通信和LAN通信的驱动程序。

(3)完成了各功能模块的测试，通过电压的显示和传输验证了控制系统的显示、通信等功能。

6.2 完善及展望

通过本文的工作，为后续在此平台上的二次开发打下了良好的基础，可以将本平台用于高速数据采集处理、高速数据通信、大容量数据存储等方面。由于时间有限，本课题尚有许多工作需要逐步完善：

(1)本系统预留了串口模块，可通过RS232与外部设备进行数据交换；

(2)预留了SD卡接口，可用做外部存储容量扩展；

(3)将本系统配合高端仪表，可实现多种功能，如在外部设备实现对高精度测量模块测量的数据进行图像识别功能、并通过控制平台向高精度测量模块发送命令等。

致 谢

在论文的撰写接近尾声的时候，回顾三年来的学习和生活，我深感自己收获颇多，这不仅包括知识面的加宽、工作能力的提高，更重要的是我对生活和未来的发展有了一个全新的认识。我的每一个进步和提高，都离不开我的导师—黄向东副教授的谆谆教导。

在我攻读硕士学位期间，黄老师在学习、生活、思想上都给了我悉心的指导与帮助。黄老师以其渊博的知识、丰富的实践经验使我学到了许多专业知识和科学研究方法。他严谨缜密的治学作风和一丝不苟的工作态度无处不感染着我、熏陶着我，使我受益终生。感谢黄老师给我创造了一个良好的学习环境和论文创作平台，使我在学习期间能高效地提高理论知识，并将理论知识充分地应用于实际工作中。在此，我要向黄老师致以最诚挚的谢意！

感谢朝夕相处的实验室的同学们，他们是李猛、王贤彬、景宁波、王宗锐、康学玺等同学，感谢他们在我课题研究和论文撰写期间为我提供的帮助和支持。

感谢我的母校西安科技大学对我的培养，在这里学到的知识将是我一生宝贵的财富！

最后，我要深深地感谢我的父母、兄妹和我的朋友，是他们无微不至的关心和帮助支持着我不断前进，感谢他们为我付出的一切！

参考文献

- [1]陆益. 工业自动化仪器仪表数字化系统技术及其发展[J].可编程控制器与工厂自动化,2006.6
- [2]徐刚, 阳富民等. 嵌入式系统总数. 计算机应用与研究. 2000. 4-9
- [3]赵伟国, 王文海等. 嵌入式系统在智能仪器仪表中的应用. 计算机应用. 2003: 55-57
- [4]杜春雷.ARM 体系结构与编程[M].北京:清华大学出版社,2003: 35-60
- [5]周立功,张华等. 深入浅出 ARM7-LPC213x/214x [M]. 北京:北京航空航天大学出版社,2005.
- [6]David Seal. ARM Architecture Reference Manual[M] .2001
- [7]张弘.USB 接口设计 第 1 版.西安:西安电子科技大学出版社, 2002.4
- [8]张念准, 江浩.USB 总线接口开发指南.第 1 版.北京:国防工业出版社, 2001.3
- [9]徐成玉. 基于 ARM 的 USB 主机系统设计研究 [D] 沈阳: 哈尔滨工业大学 2007.1
- [10]Holtek. HT16511 1/ 8 to 1/16 Duty VFD Controller. 2004.12.21.
- [11]马伟.计算机 USB 系统原理及其主/从机设计[M].北京航空航天大学出版社,2004: 20-26
- [12]王成儒,李英伟.USB2.0 原理与工程开发[M].北京,国防工业出版社,2004: 14-60
- [13]王铁勇, 侯明善等. 嵌入式系统 $\mu\text{C}/\text{OS II}$ 的特点与应用. 控制工程. 2000, 10: 74-76
- [14]Jean J Labrosse.嵌入式实时操作系统 $\mu\text{C}/\text{OS II}$ [M] .第 2 版.北京:北京航空航天大学出版社, 2004.: 209-224
- [15]王田苗.嵌入式系统设计与实例开发基于 ARM 微处理器与 $\mu\text{C}/\text{OS II}$ 实时操作系统[M] 北京:清华大学出版社, 2003.
- [16]杨洪亮,胡金演. $\mu\text{C}/\text{OS II}$ 在 ARM 处理器上的移植[J].微计算机信息,2005,2:101-102.
- [17]Rodger H. Techniques to shrink embedded system design cycles[J] Embedded Computing Design AUGUST 2006: 105-107
- [18]Jae-Ho Lee. Implementing priorit inheritance semaphore on $\mu\text{C}/\text{OS}$ real-time kernel Software Technologies for Future Embedded Systems, 2003 .IEEE Workshop on May. 2003:15-16 .
- [19][加]Kirk Zurell 著,孙玉芳译.嵌入式系统的 C 程序设计[M].机械工业出版社,2002: 30-70
- [20]陈延文. 面向智能仪表的嵌入式操作系统设计[J] 天水师范学院学报 2004.10: 23-24
- [21]Esther Spanjer,Flash management:How it extends the lifetime of solid-state Flash disks in VMEbus system,VMEbus System,2005.8

- [22]王道新, 刘玉.基于 NandFlash 文件系统的优化设计[J].微计算机信息, 2006.03: 12-14
- [23]GaL E,Toledos. Algorithms and Data Structures for Flash Memories[J]. ACM Computing Surveys, 2005, 37 (2) :138-163
- [24]阎航.Flash 文件系统研究综述[J].现代计算机: 下半月版, 2006 (9): 63-67
- [25]Dallas Corporation. Programmable Resolution One-wire Digital Thermometer[M].2000
- [26]吴迪,张军平,张文军.嵌入式计算机中 USB 主机控制器的研究与设计 计算机工程, 2004.12:174-176
- [27]万逸珠, 戚文芽.嵌入式实时操作系统下 USB2.0 接口驱动的实现.现代电子技术, 2004.13:49-51
- [28]Chih-YuanHuang, Li-PinChang, Tei-WeiKuo, A Cyclic-Executive-Based QoS Guarantee over USB, NationalTaiwanUniversity, The 9th IEEE Real-Time and Embedded Technology and Applications Symposium, May 27-30,2003, p.88
- [29]Brian Demuth, Dan Eisenreich. Embedded Internet Applicaion Handbook: A Practical Guide to Hardware and Software Design .Academic Press,2004.
- [30] Jon C Snader. Effective TCP/IP Programming .Addision-Wesley Professional, 2004:27-103 .
- [31]ThomasF.An Introduction to TCP/IP for Embedded Engineers .Embedded System Conference, SanFrancisco. 2002.
- [32]陈鹏,张爱民. 基于以太网控制器 CS8900A 实现嵌入式系统的网络互连[J].电子工程师. 2002,28(12):4-6.
- [33]黄国刚. 基于 CS8900A 的工业级嵌入式网络接口实现[J].计测技术, 200525(1):16-17.
- [34]许群丽,宋仲康,秦强明.嵌入式系统与以太网的连接[J].仪表技术 2006.5
- [35]Dae-EunHyun, Seung-HyunYoon, Myung-SOOKim, Bert Juttler, Modeling and Deformation of Arms and Legs Based on Ellipsoidal Sweeping, 11thPacific Conference on Computer Graphics and Applications(PG03), Oetober08-10, 2003 Canmore, Canada, p.204
- [36]万冬云,朱民耀.嵌入式网络中 WEB SERVER 的实现[J].微计算机信息, 2004.6: 14-15
- [37]杨素英,李丹,仲崇权,张立勇.基于 Web Server 下的嵌入式系统应用研究.[J] 仪器仪表标准化与计量, 2002: 18-19
- [38]Brian Demuth, Dan Eisenreich. Embedded Internet Applicaion Handbook: A Practical Guide to Hardware and Software Design .Academic Press,2004.
- [39]姚晓乐, 王宇坤.Web 开发技术[M]. 北京: 人民邮电出版社, 2000: 56-65

附 录

附录 1 作者在攻读学位期间发表的论文

- 1、安洁，黄向东，王振华.CY7C68001与ARM7的接口设计. 电脑知识与技术, 已录用.

附录 2 作者在攻读学位期间的科研情况

1、高端仪表产品嵌入式开发平台的研制

以 LPC2220 处理器为平台的硬件核心，开发了以 $\mu\text{C}/\text{OS-II}$ 为基础的软件系统，实现了 VFD 显示、USB 通信和嵌入式 Web 服务器等功能，为新一代仪表产品的开发提供了功能丰富的嵌入式平台。

作者: [安洁](#)

学位授予单位: [西安科技大学](#)

相似文献(10条)

1. 学位论文 [陈卫丽](#) [基于ARM和uClinux的信息家电控制平台及其应用](#) 2008

嵌入式技术越来越普及,必将带来控制器领域的升级换代。目前常见的手机、PDA(Personal Digital Assistant)、电子字典、游戏机、智能玩具、交换机、路由器、数控设备或仪表、汽车电子、家电控制系统、医疗仪器、航天航空设备等都是典型的嵌入式系统。在船舶焊接自动对弧驱动监控系统,已经用到了嵌入式系统。在将来的机器人自动焊接系统中,也必将用到嵌入式系统来达到实时控制的目的。

本课题的研究任务是将嵌入式技术应用到智能家居中,搭建一个基于ARM和uClinux的信息家电控制平台,并将此控制平台应用到家用洗碗机控制器中,开发一个具有以太网接口的功能强大的智能终端。使洗碗机除了具有传统的操作功能,还具有远程控制等功能。

本课题在深入研究嵌入式技术的基础上,构建了信息家电控制平台的硬件系统和软件系统。为了提高系统的数据处理能力、实现TCP/IP协议栈和图形界面系统,通过对各种处理器进行比较,选定含有32位ARM内核的S3C44B0X作为核心微处理器。并设计了包括存储系统、系统电源、复位电路、以太网模块、红外模块、UART、LCD显示屏、五键键盘等接口电路,构建了信息控制平台的硬件系统。

本着利用有限的资源、最大程度降低成本的原则,应用成熟而源代码免费开放的嵌入式软件,设计了基于u-boot+uClinux+MiniGUI+Boa+Romfs的解决方案,实现了一种新型智能化信息家电设备软件系统。通过进行网络服务器boa的移植实验,实现了控制平台对网络信号的处理和远程控制的功能。

最后,本文详细论述了洗碗机应用程序的实现方式。从目前的运行情况表明,通过Internet Explorer访问洗碗机的控制网页,可以成功控制洗碗机运行,实现了洗碗机控制器的网络智能控制功能。

信息家电控制平台的设计为新崛起的信息家电系统的开发提供了一种有效的方法,对其他如材料加工工程领域的嵌入式系统的开发也提供了一种技术参考。

2. 学位论文 [姜京梅](#) [基于ARM微处理器和 \$\mu\$ C/OS-II 实时操作系统的嵌入式数字温度计的设计](#) 2004

传统的8位单片机已经越来越不能适应日渐复杂的应用需求。友好的交互界面、网络互联功能、智能化的软件、高效的数据处理几乎成了智能化系统的共同需求。随着嵌入式系统的迅猛发展,这种应用系统正逐步取代传统的以PC为中心的应用,成为未来智能化仪表中的主力军。本文以基于32位ARM微处理器AT91M55800A和嵌入式实时操作系统 μ C/OS-II的嵌入式数字温度计的研制过程为主要内容,阐述了其具体技术及相关实现方法。将高性能、低功耗的32位RISC结构的ARM内核处理器与开放源码的嵌入式多任务操作系统 μ C/OS-II相结合,构建了以微处理器AT91M55800A、闪存FLASH、存储器SRAM、A/D、键盘、显示器为一体的嵌入式数字温度计的硬件平台。在此硬件平台上嵌入 μ C/OS-II嵌入式实时操作系统,结合软件平台EmbestforARM,最终完成了嵌入式数字温度计的设计开发。

3. 学位论文 [张建伟](#) [基于ARM的工程地表平整度测量系统设计](#) 2006

所谓工程地表是指施工后的公路、广场地面、大厅等建筑地面,在工程质量管理中对这种表面的光滑平整程度,即表面平整度有一定要求,以确保工程质量。

为更好实现工程地表平整度的智能测量,本论文提出了一种基于ARM微处理器的、适于工程应用的工程地表平整度测量系统设计方案,并对系统关键技术进行了详细描述。该系统采用了光学传感器和数字化处理技术,应用嵌入式ARM微处理器、串行接口模块以及液晶显示模块,可以实时处理测量数据,并将测量结果显示出来。移植了uClinux操作系统,为日后系统升级、担负更加复杂的任务奠定了基础。

近几年,ARM嵌入式技术得到了迅猛发展,ARM系统以其功耗低、体积小、功能多等特点,成为工业信息化领域的研究热点。将其应用到测量仪器和智能控制领域,可以达到降低成本、简化结构等效果。采用ARM技术的本系统设计,为灵活方便地测量工程地表、评定工程地表的级别提供了一种比较可靠的方案,为改进我国的工程验收设备进行了有益的探索。

本文综述了目前国内外工程地表平整度测量技术的发展状况,讨论了基于ARM7内核的平整度测量仪的设计与实现方案,分析了平整度测量仪的传感器、数据处理、液晶显示等功能模块,并介绍了嵌入式操作系统uClinux的移植过程。

4. 期刊论文 [汤建斌](#). [蒋庆](#). [蔡晋辉](#). [张焱](#). [张庆峰](#). [TANG Jian-bin](#). [JIANG Qing](#). [CAI Jin-hui](#). [ZHANG Yan](#). [ZHANG Qing-feng](#) [基于ARM微处理器的自动称重选别系统-轻工机械](#)2009, 27 (1)

介绍了一种新型的高精度自动称重分选系统,其特点是能快速准确地对产品进行称重分选,并实时计算、存储、显示数据.针对动态称重数据的特点,提出斜率滤波的软件算法,设计了对应的软件滤波程序,改善了数据处理精度和效率.实验结果表明,该系统运行精确可靠.

5. 会议论文 [杨柳](#). [陈英杰](#). [王书茂](#) [基于ARM的车载防滚翻防撞碰撞监控与预警系统的研究](#) 2008

以叉车和装载机为研究对象,基于嵌入式软硬件技术,设计了一套车载防滚翻、防撞碰撞监控与预警系统.该系统包括车辆状态参数采集、处理和显示,并具有碰撞和倾翻的预报警等功能.系统以ARM7内核的微处理器LPC2290为核心,扩展了串口通信、液晶显示、报警等硬件模块.经过数据处理,系统能实时监控和显示各项参数并与稳定性的临界值比较后进行预报警.

6. 学位论文 [房印](#) [基于ARM微处理器的输电线路距离保护装置的研究](#) 2006

在现代电网中,随着超高压、大容量、远距离输电线路的不断增多,对电力系统的安全稳定运行提出了更高、更严格的要求.距离保护作为线路保护的基本组成部分,其工作特性对电力系统的安全稳定运行有着直接和重要的影响.为了适应现代超高压电网稳定运行的要求,微机距离保护装置在硬件和软件上都提出了越来越高的要求.

本文在总结了输电线路微机距离保护研究现状的基础上,分析了国内现有微机保护硬件的现状,对比了8位、16位单片机(SCM)以及32位DSP的保护方案,并总结了其优缺点,提出了一种基于ARM微处理器的保护方案.本装置以SamsungElectronics公司生产的S3C4510B芯片作为保护CPU,主要负责控制数据采集、采样数据处理、实现保护功能、人机界面以及外围串口和以太网通信等,再配合CPLD器件实现采样保持、多路选择开关和开关量输入输出的控制.本文实现了微机保护方案的硬件设计,完成了处理器外围电路和采样电路环节的设计.

本文在分析了国内外现有的软件设计中保护算法和功能控制的基础上,同时结合组态软件可以跳过硬件直接进行软件设计的优点,提出了基于组态设计思想的微机距离保护软件设计方案.本文将保护程序分成一个个独立的模块,然后将各模块的参数分成两类:与程序的执行有关的执行参数和程序本身所传递的数据参数,根据面向对象的思想对各模块编程,使它们成为距离保护的各组件,并对这些组件进行了验证.

7. 学位论文 [尹江会](#) [嵌入式实时操作系统 \$\mu\$ C/OS-II 在ARM中的应用研究](#) 2005

随着计算机技术和通信技术的迅速发展以及Internet的不断扩展,嵌入式系统得到了越来越广泛的应用.实时的数据处理、网络互连功能、智能化的软件以及良好的功能扩展能力几乎成了现代实时测控系统的共同需求.由于嵌入式系统的复杂性不断增加,嵌入式操作系统已经成为嵌入式系统中最重要组成部分.在各种嵌入式操作系统中, μ C/OS-II凭借其结构清晰、源代码开放性等方面的优势,成为了基于监控系统、手持设备等嵌入式系统领域应用中的技术热点.将嵌入式操作系统 μ C/OS-II与模块化硬件设计相结合,共同构成一个可以重复利用的软硬件数字系统平台,除了可以最大限度地提高开发的效率、减少资源的浪费外,还可以通过长期对于该平台的研究,逐步优化平台软硬件资源,满足日益复杂的应用需求.

嵌入式操作系统在嵌入式系统设计中处于核心地位.本文以ARM7TDMI为嵌入式实时系统微处理器,以 μ C/OS-II为嵌入式实时操作系统内核,通过硬件的扩展和软件的功能扩充,实现了一个基本完整的嵌入式实时操作系统.

首先,作者对嵌入式操作系统的现状、发展和一些重要的概念作出表述,并介绍了 μ C/OS-II的产生、发展.接着,详细分析了嵌入式实时操作系统 μ C/OS-II的任务与相关数据结构、任务的状态与调度、系统的通信机制.本文指出了 μ C/OS-II的缺点:尽管它可用于各类单片机,在诸多领域得到广

泛的应用，但是它只能支持固定优先级的64个就绪任务，对于不断出现的新的更加复杂的实时系统，往往满足不了需求。本文的创新点在于：在对μC/OS-II任务管理进行系统分析的基础上，提出了采用优先队列扩展其任务管理的方案，并给出了实现这种方案的具体方法。改进后的μC/OS-II的就绪任务管理数目大大增加，任务调度更加灵活。

然后，详细介绍了ARM核芯片的应用领域、ARM指令集的优点、ARM体系的变种和ARM7TDMI、ARM的寄存器组织和μC/OS-II在ARM7上的移植过程。

其次，由于μC/OS-II只是一个操作系统内核，只提供了资源管理的基本功能，要实现一个相对完整、实用的嵌入式实时多任务操作系统，需要对内核进行功能扩充。在对其成功移植的基础上，对实时操作系统μC/OS-II进行了软件功能的扩充和硬件的扩展。在整个设计中完成了文件系统的建立、串口通信的实现、系统的消息队列、网络相关组件。

最后，设计了简单的应用程序对所搭建平台的功能进行测试，多任务嵌入式实时系统稳定可靠运行。结合实际开发过程，本文总结了嵌入式实时操作系统的特点、程序的编制规则以及应用程序的开发方法。

8. 学位论文 [肖楠 ARM及在Linux在高精度测温系统中的应用](#) 2009

温度的测量在工业领域最为常见，随着电子技术、计算机技术的飞速发展，对现场温度的测量也由过去的模拟刻度温度计、指针温度计向数字显示的智能温度计发展，而且，对测量的精度要求也越来越高。目前，尽管市场上也有高精度的温度测量仪，但一般价格都很昂贵。传统的8位单片机已经越来越不能适应日渐复杂的应用需求。友好的交互界面、网络互联功能、智能化的软件、高效的数据处理几乎成了智能化系统的共同需求。随着嵌入式系统的迅猛发展，这种应用系统正逐步取代传统的以PC为中心的应用，成为未来智能化仪表中的主力军。本文立足于设计一种通用性强的测温系统，可以在软硬件两方面适应多种测温元件，为系统日后升级带来方便。

本文以对通用Linux操作系统在32位ARM微处理器上进行移植并对其进行实时性进行了改造。研制了铂热电阻高精度温度监测系统，阐述了其具体技术指标及相关实现方法。系统以S3C2410为硬件核心，开发了主板及数据采集调理电路。构建了以微处理器S3C2410、闪存FLASH、存储器SRAM、A/D、键盘、显示器为一体的温度监测的硬件平台。在此硬件平台上嵌入RT-Linux嵌入式实时操作系统，构建系统的多任务管理，最终完成了本课题的设计开发。

9. 学位论文 [李安宗 网络化随钻测井数据采集与处理技术的研究与设计](#) 2006

本文研究了基于ARM微处理器和VxWorks实时操作系统的网络化嵌入式数据采集系统体系结构及实现方式，并在网络化随钻测井数据采集与处理系统中设计实现，系统已成功进行了现场试验。网络化随钻测井数据采集系统的思路，是把随钻测井数据采集系统与计算机网络相结合，构成信息采集、传输、处理和应用的综合网络。系统结构符合信息化发展的要求，是具有信息时代特点的新思路。系统设计采用分布式结构，由现场数据采集部分负责实时数据采集与预处理，由现场操作控制计算机负责数据的处理、显示、绘图、记录、远程传输、解释与评价等任务。软件设计采用面向对象的设计思想和软件工程的方法，充分发挥面向对象设计的封装性、继承性和多态性的特点。研究内容及取得的主要成果如下：1) 随钻测井数据采集系统的需求分析及体系结构设计。通过详细分析对随钻测井地面系统功能和性能的需求，提出了网络化的嵌入式数据采集系统的体系结构。系统硬件采用ARM微处理器，软件采用VxWorks实时操作系统。针对随钻测井应用的需要，把该体系结构用于随钻测井地面系统的开发中，分析了系统的组成和实现方式，并设计了系统的数据流程图。2) 设计实现了基于ARM微处理器的32位嵌入式数据采集系统，并成功应用于随钻测井的数据采集中。根据传感器采集的信号形式及泥浆脉冲传输格式，设计了信号的预处理电路、ARM微处理器及其外围电路。设计开发了由CPLD控制的数据采集系统，实现了由硬件控制的连续实时数据采集和数据处理，提高了系统的信号采集和处理能力。3) 基于VxWorks实时操作系统的嵌入式实时多任务采集软件的研究与设计。设计了S3C44B0XARM微处理器的BSP软件包，从而完成了VxWorks操作系统到该微处理器的移植。在此基础上研究设计了随钻测井嵌入式实时多任务采集软件，实现了基于客户/服务器模式的实时采集系统软件框架。4) 研究设计了基于以太网的网络化数据采集与处理系统，实现了本地和远程的网络化数据采集与传输。通过网络技术的集成，系统可以随时随地与网络进行连接，实现资源共享。5) 地面系统软件的面向对象分析、面向对象设计与面向对象编程实现。

10. 期刊论文 [龚仁喜. 刘丰. 黄阳. 孟小碧. 谢玲玲. 龚文英. GONG Ren-xi. LIU Feng. HUANG Yang. MENG Xiao-bi. XIE Ling-ling. GONG Wen-ying 一种基于FPGA+ARM的高速电力谐波检测仪硬件的设计与实现 -广西大学学报\(自然科学版\)](#) 2009, 34(1)

介绍了一种结合FPGA硬逻辑的高速数据处理能力和ARM的高效数字功能扩展能力,实现实时高速电力谐波检测的“FPGA+ARM”硬件新构架.这种新架构采用复用逻辑及流水线技术在FleA上实现了A/D采样控制、加窗、FFT及模平方等运算.采用uClinux为操作平台在ARM处理器完成对FPGA的现场配置、数据通信处理及人机交互接口等功能.实际应用表明,这种架构可较好地解决电力谐波检测中的“实时性与精确度的矛盾”.

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y1545454.aspx

下载时间: 2010年5月25日