# 嵌入式系统设计论文解读（五篇）

来源：网络 作者：红尘浅笑 更新时间：2024-09-08

*第一篇：嵌入式系统设计论文解读嵌入式系统设计论文专业：电子信息工程（信号处理）班级：姓名：指导教师：评分：年月 日【摘要】当今信息时代，嵌入式系统的应用无处不在，而ARM嵌入式系统应用市场份额约占75%。从嵌入式系统的基本概念入手，分别从...*

**第一篇：嵌入式系统设计论文解读**

嵌入式系统设计论文

专

业：

电子信息工程（信号处理）

班

级：

姓

名：

指导教师：

评

分：

年

月 日

【摘要】

当今信息时代，嵌入式系统的应用无处不在，而ARM嵌入式系统应用市场份额约占75%。从嵌入式系统的基本概念入手，分别从ARM的定义、ARM微处理器、ARM开发工具及调试方法来介绍ARM嵌入式系统基础知识。接着，讨论了ARM嵌入式系统的实时性要求，介绍了目前市场上的实时多任务操作系统（RTOS）。最后，概括了ARM技术的应用领域及其产业化发展，并预测ARM技术发展的前景。

关键词：嵌入式系统；ARM；微处理器；RTOS

目录

1.引言......................................................................1 2.嵌入式系统................................................................1 2.1 2.2

...................................................1...................................................1 3.ARM嵌入式系统............................................................2 3.1 什么是ARM.........................................................2 3.2 ARM3.3 ARM3.4 ARM3.5 ARM

.............................................2.................................................3

.............................................3

...................................4 4.ARM嵌入式系统的实时性要求................................................5 4.1 嵌入式系统软件需要RTOS

....................................5 4.2 RTOS...............................................................6 5.嵌入式系统的信息产业化发展.................................................6 5.1 ARM

.................................................6 5.2 嵌入式系统产业化发展.................................................7 6......................................................................7 参考文献.....................................................................8

1.引言

随着计算机技术、网络技术和微电子技术的深入发展，嵌入式系统的应用无处不在。

ARM是目前公认的业界领先的32位嵌入式RISC（精简指令计算机）微处理器。ARM技术日益成熟和不断发展，正在逐步渗入到我们生活的各个方面。

本文从实际出发，首先介绍嵌入式系统的基本概念，随之重点阐述了由ARM微处理器构成的嵌入式系统（简称ARM嵌入式系统）的基础知识，最后分析了ARM技术的产业化发展过程及发展趋势。.嵌入式系统

2.1

嵌入式系统的英文叫做Embedded System，是一种包括硬件和软件的完整的计算机系统，但又跟通用计算机系统不同。嵌入式系统的定义是：“嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可剪裁，适用于应用系统对功能、可靠性、成本、体积和功耗有严格要求的专用计算机系统。”嵌入式系统所用的计算机是嵌入到被控对象中的专用微处理器，但是功能比通用计算机专门化，具有通用计算机所不能具备的针对某个方面特别设计的、合适的运算速度、高可靠性和较低比较成本的专用计算机系统。

2.2

嵌入式系统作为一类特殊的计算机系统，自底向上包含有3个部分，如图1所示。

（1）硬件环境：是整个嵌入式操作系统和应用程序运行的硬件平台，硬件平台包括嵌入式处理器和外围设备。嵌入式处理器是嵌入式系统的核心，是控制、辅助系统运行的硬件单元。

（2）嵌入式操作系统：完成嵌入式应用的任务调度和控制等核心功能。具有内核较精简、可配置、与高层应用紧密关联等特点。嵌入式操作系统具有相对 不变性。

（3）嵌入式应用程序：运行于操作系统之上，利用操作系统提供的机制完成特定功能的嵌入式应用。不同的系统需要设计不同的嵌入式应用程序。.ARM嵌入式系统

3.1 什么是ARM

ARM是Advanced RISC Machines的缩写，是微处理器行业的一家知名企业，该企业设计了大量廉价、高性能、低功耗的RISC

ARM技术有很好的性能和功效，其合作伙伴包括世界许多顶级的半导体公司。目前，共有30家半导体公司与ARM签订了硬件技术使用许可协议，其中包括Intel、IBM、LG半导体、NEC、SONY、PHILIPS和国家半导体这样的大公司。可以说，ARM不仅仅代表一个公司，代表了一类微处理器，代表了一种技术，还代表了一种新型的产业发展模式。

3.2 ARM处理器核系列及应用

ARM公司开发了一系列ARM处理器核。目前最新的系列已经是ARM11了。ARM6及更早的系列已经罕见了，ARM7以后的核也不是都获得广泛应用。目前应用最多的是ARM7系列、ARM9系列、ARM9E系列、ARM10系列、SecurCore系列、Intel的StrongARM、XScale

ARM7系列：包括ARM7TDMI、ARM7TDMI-S、带有高速缓存处理器宏单元的ARM720T和扩充了Iazelle 的ARM7EJ-S。该系列广泛应用于多媒体和嵌入式设备，包括Internet设备、网络和调制解调器设备以及移动电话、PDA等无线设备。

ARM9系列：包括ARM9TDMI、ARM920T和带有高速缓存处理器宏单元的ARM940T。该系列主要应用于引擎管理、仪器仪表、安全系统、机顶盒、高端打印机、PDA、网络电脑以及带有MP3音频和MPEG4视频多媒体格式的智能电话

ARM9E系列：为综合处理器，包括ARM926EJ-S、带有高速缓存处理宏单元的ARM966E-S/ARM946E-S。该系列强化了数字信号处理功能，可应用于需要DSP与微控制器结合使用的情况，将Thumb技术和DSP都扩展到ARM指令集中，并 具有EmbeddedICE-RT

ARM10系列：包括ARM1020E和ARM1020E微处理器核。其核心在于使用向量浮点（VFP）单元VFP10提供高性能的浮点解决方案，从而极大地提高了处理器的整型和浮点运算性能，为用户界面的2D和3D图形引擎应用夯实基础，SecurCore系列：包括SC100、SC110、SC200和SC210处理器核。该系列主要针对新兴的安全市场，以一种全新的安全处理器设计为智能卡和其他安全IC开发提供独特的32位系统设计，并具有特定的反伪造方法，从而有助于防止对

StrongARM系列：StrongARM处理器将Intel处理器技术和ARM体系结构融

Xscale系列：提供全性能、高性价比和低功耗的解决方案，支持16位Thumb指令和DSP指令。

3.3 ARM微处理器的特点

采用RISC体系架构的ARM微处理器一般有如下特点:（1）体积小、低功耗、低成本、高性能；

（2）支持Thumb(16位)/ARM（32位）双指令集，能很好的兼容8位/16位器件；

（3）大量使用寄存器，指令执行速度更快；（4）大多数数据操作都在寄存器中完成；（5）寻址方式灵活简单，执行效率高；（6）指令长度固定。

3.4 ARM微处理器的指令结构

ARM微处理器在较新的体系结构中支持两种指令集：ARM指令集和Thumb指令集。其中，ARM指令为32位的长度，Thumb指令为16位长度。Thumb指令集为ARM指令集的功能子集，但与等价的ARM代码相比较，可节省30％-40％以上的存储空间，同时具备32位代码的所有优点。

3.5 ARM嵌入式系统开发工具及调试方法

ARM处理器产品作为一种高性能、低功耗的处理器产品，现在已经得到广泛的应用。

ARM开发工具也因此得到发展，除ARM公司自己推出ARM集成开发工具外，还有一些公司也研发ARM开发工具。目前市场上有ARM SDT、ARM ADS、MULTI 2024、Hitools for ARM、Embest IDE for ARM五种集成开发环境。这些产品在国内有相对较畅通的销售渠道，用户容易购买。前三种由国外厂商出品，历史比较悠久，在全球范围内应用较为广泛，后两种由国内厂商推出，具有很高的性价比。

ARM应用软件的开发工具根据功能的不同，分别有编译软件、汇编软件、链接软件、调试软件、嵌入式实时操作系统、函数库、评估板、JTAG仿真器、在线仿真器等。

用户选用ARM处理器开发嵌入式系统时，选择合适的开发工具可以加快开发进度，节省开发成本。因此一套含有编辑软件、编译软件、汇编软件、链接软件、调试软件、工程管理及函数库的集成开发环境（IDE）一般来说是必不可少的，至于嵌入式实时操作系统、评估板等其他开发工具则可以根据应用软件规模和开发计划选用。目前常见的调试方法有以下几种。

（1）指令集模拟器。

部分集成开发环境提供了指令集模拟器，可方便用户在PC机上完成一部分简单的调试工作，但是由于指令集模拟器与真实的硬件环境相差很大，因此即使用户使用指令集模拟器调试通过的程序也有可能无法在真实的硬件环境下运行，（2）驻留监控软件。

驻留监控软件（Resident Monitors）是一段运行在目标板上的程序，集成开发环境中的调试软件通过以太网口、并行端口、串行端口等通讯端口与驻留监控软件进行交互，由调试软件发布命令通知驻留监控软件控制程序的执行、读写存储器、读写寄存器、设置断点等。

驻留监控软件是一种比较低廉有效的调试方式，不需要任何其他的硬件调试和仿真设备。ARM公司的Angel就是该类软件，大部分嵌入式实时操作系统也是采用该类软件进行调试，不同的是在嵌入式实时操作系统中，驻留监控软件是作为操作系统的一个任务存在的。

驻留监控软件的不便之处在于它对硬件设备的要求比较高，一般在硬件稳定之后才能进行应用软件的开发，同时它占用目标板上的一部分资源，而且不能对 程序的全速运行进行完全仿真，所以对一些要求严格的情况不是很适合。

（3）JTAG仿真器。

JTAG仿真器也称为JTAG调试器，是通过ARM芯片的JTAG边界扫描口进行调试的设备。JTAG仿真器比较便宜，连接比较方便，通过现有的JTAG边界扫描口与 ARM CPU 核通信，属于完全非插入式(即不使用片上资源)调试，它无需目标存储器，不占用目标系统的任何端口，而这些是驻留监控软件所必需的。另外，由于JTAG调试的目标程序是在目标板上执行，仿真更接近于目标硬件，因此，许多接口问题，如高频操作限制、AC和DC参数不匹配，电线长度的限制等被最小化了。使用集成开发环境配合JTAG仿真器进行开发是目前采用最多的一种调试方式。目前国际市场上较流行的两种JTAG仿真器：EPI公司的JEENI和ARM公司的Multi-ICE。

（4）在线仿真器。

在线仿真器使用仿真头完全取代目标板上的CPU，可以完全仿真ARM芯片的行为，提供更加深入的调试功能。但这类仿真器为了能够全速仿真时钟速度高于100MHz的处理器，通常必须采用极其复杂的设计和工艺，因而其价格比较昂贵。在线仿真器通常用在ARM的硬件开发中，在软件的开发中较少使用，其价格高昂也是在线仿真器难以普及的因素。

另外国际市场上较流行的有两种JTAG仿真器：EPI公司的JEENI和ARM公司的Multi-ICE。

4.ARM嵌入式系统的实时性要求

4.1 嵌入式系统软件需要RTOS开发平台

通用计算机具有完善的操作系统和应用程序接口(API)，是计算机基本组成不可分离的一部分，应用程序的开发以及完成后的软件都在OS平台上面运行，但一般不是实时的。嵌入式系统则不同，应用程序可以没有操作系统直接在芯片上运行；但是为了合理地调度多任务、利用系统资源、系统函数以及和专家库函数接口，用户必须自行选配RTOS开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，保障软件质量。

4.2 RTOS

RTOS是英文Real Time multi-tasking Operation System的缩写，即实时多任务操作系统。它是嵌入式应用软件的基础和开发平台。目前在中国大多数嵌入式软件开发还是基于处理器直接编写，没有采用商品化的RTOS，不能将系统软件和应用软件分开处理。RTOS最关键的部分是实时多任务内核，它的基本功能包括任务管理、定时器管理、存储器管理、资源管理、事件管理、系统管理、消息管理、队列管理、旗语管理等，这些管理功能是通过内核服务函数形式交给用户调用的，也就是RTOS的API。RTOS的引入，对嵌入式软件的标准化和加速知识创新是一个里程碑。

5.嵌入式系统的信息产业化发展

5.1 ARM技术的应用领域

现在，嵌入式技术无处不在，ARM几乎成为嵌入式技术的代名词。作为一种16/32位高性能、低成本、低功耗的嵌入式RISC微处理器，ARM微处理器目前已经成为应用广泛的嵌入式微处理器。ARM微处理器及技术的应用几乎已经深入到各个领域。

(1)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但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应用领域扩展，ARM控制器的低功耗、高性价比，向传统的8位/16位微控制器提出了挑战。

(2)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，ARM以其高性能和低成本，在该领域的地位日益巩固

(3)网络应用：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，并获得广泛支持，也对DSP的应用领域提出了挑战。

(4)消费类电子产品：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应用。

(5)成像和安全产品：现在流行的数码相机和打印机中绝大部分采用ARM技术。手机中的32位SIM智能卡也采用了ARM技术。除此以外，ARM微处理器及技术还应用到许多不同的领域，并会在将来取得更加广泛的应用。

5.2 嵌入式系统产业化发展

市场需求为嵌入式系统产业化发展注入巨大推动力。嵌入式系统的市场是巨大的，市场需求是嵌入式系统产业化发展的巨大推动力。据报告，10%～20%的计算机芯片是为台式或便携式电脑设计的，80%～90%的计算机芯片是为嵌入式设备设计的，这意味着每年有10亿至20亿个CPU是为嵌入式设备设计制造的。2024年，全球嵌入式系统产品的产值已达2024亿美元，估计全世界嵌入式系统产品潜在的市场将超过10 000亿美元。世界范围内嵌入式系统带来的工业年产值已超过了1万亿美元。

6.随着半导体工艺技术的发展及系统设计水平的提高，ARM技术将会不断的变化和进步，ARM技术的应用领域会再次扩大，ARM技术带来的工业产值将会剧增，ARM技术产品的市场前景更加美好。总之，ARM技术的不断创新会给人类社会生活带来奇迹和享受。

参考文献

［1］ARM公司.ARM Architecture Reference Manual［M］.ARM公司，2024.［2］周立功.ARM嵌入式系统基础教程［M］.北京：北京航空航天大学出版社，2024.［3］郑文波，曹金安.嵌入式系统的产业化发展——市场、技术与前景［J］.自动化博览，2024，(1):17.［4］罗蕾.嵌入式实时操作系统及应用开发［M］.北京：北京航空航天出版社，2024.［5］中国计算机学会微机专业委员会.2024中国计算机科学技术年度发展报告——嵌入式系统部分［EB/OL］.读书的好处

1、行万里路，读万卷书。

2、书山有路勤为径，学海无涯苦作舟。

3、读书破万卷，下笔如有神。

4、我所学到的任何有价值的知识都是由自学中得来的。——达尔文

5、少壮不努力，老大徒悲伤。

6、黑发不知勤学早，白首方悔读书迟。——颜真卿

7、宝剑锋从磨砺出，梅花香自苦寒来。

8、读书要三到：心到、眼到、口到

9、玉不琢、不成器，人不学、不知义。

10、一日无书，百事荒废。——陈寿

11、书是人类进步的阶梯。

12、一日不读口生，一日不写手生。

13、我扑在书上，就像饥饿的人扑在面包上。——高尔基

14、书到用时方恨少、事非经过不知难。——陆游

15、读一本好书，就如同和一个高尚的人在交谈——歌德

16、读一切好书，就是和许多高尚的人谈话。——笛卡儿

17、学习永远不晚。——高尔基

18、少而好学，如日出之阳；壮而好学，如日中之光；志而好学，如炳烛之光。——刘向

19、学而不思则惘，思而不学则殆。——孔子

20、读书给人以快乐、给人以光彩、给人以才干。——培根

**第二篇：基于嵌入式Linux的磁场测量系统的设计解读**

基于嵌入式Linux的磁场测量系统的设计

摘要：实现了一种全集成可变带宽中频宽带低通滤波器，讨论分析了跨导放大器-电容(OTA—C)连续时间型滤波器的结构、设计和具体实现，使用外部可编程电路对所设计滤波器带宽进行控制，并利用ADS软件进行电路设计和仿真验证。仿真结果表明，该滤波器带宽的可调范围为1～26 MHz，阻带抑制率大于35 dB，带内波纹小于0．5 dB，采用1．8 V电源，TSMC 0．18μm CMOS工艺库仿真，功耗小于21 mW，频响曲线接近理想状态。关键词：Butte 1 引言

随着科技的发展，嵌入式操作系统在越来越多的领域发挥着重要的作用，目前已成为产品技术水平的标志之一。其中Linux因为其拥有开放性、多用户、多任务、良好的用户界面、丰富的网络功能、可靠的系统安全和良好的可移植等特性被广泛的应用到仪器测量设备中。

传 统的磁场测量设备(持斯拉计、高斯计)普遍存在精度低(典型测量精度为1．5％)、操作不便等缺点。本文提出一种基于嵌入式Linux的中频磁场测量系 统，它不但可以满足当前磁场测量数据采集的需要，还因为其嵌入了操作系统Linux，使具有可靠性好、升级方便的特点，既提高了磁场测量的准确性，又为仪 器的功能升级带来便利。可应用于实验室仪器，医疗仪器，姿态控制，安全检测等需磁场检测的领域。2 磁场测量系统的硬件结构

磁场测量系统在硬件结构上采用ARM9作为控制器，与信号放大、整流滤波、程控放大等硬件构成了整个磁场测量系统。而且，测量系统还搭配了USB、RS232、以太网通信接口，系统通讯能力强，可实现网络连网功能。其硬件结构如图1所示。

ARM9 嵌入式处理器采用三星公司的$3C2410。S3C2410是基于ARM920T内核的16／32位RISC嵌入式微处理器，主要面向手持设备以及高性价 比，低功耗的应用。它采用5级流水线和全性能的MMU，同时该芯片集成了大量的功能扩展单元，例如LCD控制器、I2C总线、触摸屏接口、USB接口等。强大的芯片功能简化了系统设计，不但缩小了系统体积，而且提高了系统的可靠性。

USB、RS232和以太网接口可为系统提供不同的通信方式，适合不同测量环境和条件，以太网接口还可实现系统联网功能。

在磁场测量系统中，可使用触摸屏简便地对系统进行控制，实现不同显示方式切换、参数设置和测量数据保存等功能。

磁场测量电压信号部分，由磁场传感器得到微弱的电压信号，经放大整流等措施后输入控制系统。

报警输出可实现用户自设定报警的上下限值，方便用户测量现场的监控。磁场测量系统前端信号处理模块

磁场测量系统前端传感器采用的测量方法为电磁感应法。电磁感应法是将测量线圈置入交变磁场中，根据法拉第电磁感应定律在线圈的引线间会产生感应电动势，并且感应电动势的大小与穿过线圈的磁通量的大小成正比。感应电动势e为： 测量系统前端信号处理模块的结构如图2所示。

(1)为了实现多路磁场传感器的信号输入，设计中采用片选芯片74HCl5进行通道的选择。通过对74HCl53控制端S0，Sl输入不同的数据组合(00，1O，0l，11)，输出端lY和2Y就可实现不同输入通道的选择。

(2)在整个系统的电路设计中，前置放大电路的主要作用是将传感器输出的、和磁场强度成正比的微弱电压信号放大。根据其要求，设计的前置放大电路采用了差动放大的方式，电路如图3所示。它具有高共模抑制比、输入阻抗高、输出阻抗低、失调小、温漂小、线性好等优点。

**第三篇：《嵌入式系统设计》教案**

嵌入式系统设计教案

课程总学时： 32 讲课学时： 24 实验学时： 8 授 课 人： 杨词慧

南昌航空大学信息工程学院

目录 嵌入式系统概述.......................................................................................................................1 1.1 嵌入式系统的基本概念...............................................................................................1 1.2 嵌入式系统的发展历史...............................................................................................2 1.3 嵌入式系统的体系结构...............................................................................................2 1.4 嵌入式处理器...............................................................................................................3 1.5 嵌入式操作系统...........................................................................................................4 1.6 嵌入式系统的应用及发展趋势...................................................................................7 ARM体系结构.........................................................................................................................8 2.1 ARM设计思想.............................................................................................................8 2.2 ARM体系结构分析.....................................................................................................9 2.3 ARM处理器系列.......................................................................................................11 2.4 ARM处理器模式.......................................................................................................15 2.5 ARM体系的异常处理...............................................................................................16 2.6 ARM内部寄存器.......................................................................................................18 2.7 ARM体系的存储系统...............................................................................................19 ARM指令系统及程序设计基础...........................................................................................21 3.1 ARM寻址方式...........................................................................................................21 3.2 ARM指令集...............................................................................................................24 3.3 Thumb指令集............................................................................................................33 3.4............................................................................................................................................35 3.5............................................................................................................................................35 3.6 嵌入式系统设计与开发过程.....................................................................................36 STM32微控制器...................................................................................................................37 4.1 STM32微控制器的性能指标....................................................................................37 2 3 4

I 嵌入式系统概述

教学目的：使学生对嵌入式系统的基本概念和体系结构、嵌入式处理器、嵌入式操作系统、嵌入式系统的历史、应用及发展趋势有一定的了解。

教学重点：嵌入式系统的基本概念、嵌入式处理器和嵌入式操作系统。教学难点：嵌入式系统的体系结构、嵌入式处理器。教学方法与教学手段：课堂讲授，多媒体教学。教学时间：2课时。教学内容

1.1 嵌入式系统的基本概念

(1)嵌入式系统的定义

先举例说明生活中的各种嵌入式系统设备，如iPhone、小米手机、洗衣机、电压力锅等。IEEE（国际电气和电子工程师协会）的定义：嵌入式系统是用于控制、监视或者辅助操作机器和设备的装置（Devices used to control, monitor, or assist the operation of equipment, machinery or plants）。

微机学会的定义：嵌入式系统是以嵌入式应用为目的的计算机系统，可分为系统级、板级和片级。a)系统级：各种类型的工控机、PC104等模块。b)板级：各种类型的带CPU的主板或OEM产品。c)片级：各种以单片机、DSP、微处理器为核心的产品。

一般定义：嵌入式系统是以应用为中心，以计算机技术为基础，软件硬件可裁剪，对功能、可靠性、成本、体积、功耗要求严格的专用计算机系统。

(2)嵌入式系统的特点 a)专用、软硬件可剪裁配置。b)低功耗、高可靠性、高稳定性。c)软件代码矮小精悍。d)代码可固化。e)实时性。f)弱交互性。

g)软件开发通常需要专门的开发工具、环境和方法。h)要求开发、设计人员具有较高的技能。i)具有较长的生命周期。(3)嵌入式系统的分类

按嵌入式微处理器的位数可分为：4位、8位、16位、32位和64位。按实时性能可分为：非实时系统和实时系统。

按软件结构可分为：嵌入式单线程系统和嵌入式事件驱动系统。

按应用领域可分为：信息家电类、消费电子类、医疗电子类、移动终端类、通信类、汽车电子类、工业控制类、航空电子类、军事电子类等。

1.2 嵌入式系统的发展历史

(1)以单芯片为核心的可编程控制器形成的系统

1971年11月，Intel推出Intel 4004。随后出现Intel 8080/8085、8086、Motorola的6800、68000，Zilog的Z80、Z8000。

以微处理器为核心构成的系统

单板机：Intel的iSBC系列、Zilog的MCB等

将计算机做在一个芯片上，大部分应用于专业性强的工业控制系统中，没有操作系统的支持，系统结构和功能相对单一，处理效率较低，存储容量较小。

(2)以嵌入式CPU为基础、以简单操作系统为核心的嵌入式系统

CPU种类繁多，通用性较弱；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，用户界面不够友好。

(3)以嵌入式操作系统为标志的嵌入式系统

嵌入式操作系统可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；具备文件和目录管理，支持多任务、网络应用，具备图形窗口和用户界面；有大量的应用程序接口API。

(4)以Internet为标志的嵌入式系统 嵌入式设备与Internet的结合。

1.3 嵌入式系统的体系结构

(1)体系结构

(2)硬件层

 嵌入式处理器：ARM、DSP、FPGA  存储器系统：ROM、FLASH、SDRAM  中断控制器、DMAC、定时器/计数器、UART、USB控制器、LCD控制器等  I/O接口：USB、I2C、SPI、CAN等

(3)中间层

 硬件抽象层(Hardware Abstraction Layer, HAL)。位于操作系统内核与硬件电路之间的接口层，隐藏硬件接口细节。

 板级支持包（Board Support Package, BSP）。为上层的驱动程序提供访问硬件设备寄存器的函数包。 设备驱动程序

1.4 嵌入式处理器

(1)嵌入式处理器的分类

中高端的嵌入式微处理器（Embedded Micro-Processor Unit, EMPU）低端的微控制器（Microcontroller Unit, MCU）DSP处理器（Digital Signal Processor, DSP）高度集成的片上系统（System on Chip, SoC）(2)嵌入式微处理器

由通用计算机中的CPU演变而来，只保留与嵌入式应用紧密相关的功能硬件，在工作温度、抗电磁干扰、可靠性、功耗等方面做了各种增强。

主要的嵌入式处理器类型：ARM、MIPS、PowerPC、68000系列等。特点

a)在设计中考虑低功耗。

b)采用可扩展的处理器结构。处理器内部留有很多扩展接口。c)具有很强的存储区保护功能。d)提供丰富的调试功能。

e)对实时任务具有很强的支持能力。(3)微控制器

俗称单片机，将整个计算机系统集成到一块芯片中。

WatchWatchdogdogGPIO/计数GPIOUARTUART定时定时/计数A/DA/DD/A器D/A器2CCANCANII2CSPISPIUSBUSBCPUCPUSRAMSRAM核心部分FlashFlash8位单片机示意图以一种微处理器为核心，芯片内部集成Flash、RAM、总线逻辑、定时器/计数器、I/O口、串行口、PWM、A/D、D/A等。

最早的单片机： 1976年，Intel的8048，Motorola的68HC05、Zilog公司的Z80.(4)DSP处理器

对系统结构和指令进行了特殊设计，使其适合DSP算法 高效乘累加运算、超标量操作、指令流水线 高效数据存取、硬件重复循环 确定性操作（程序执行时间可预测）应用场合：音视频编解码、数字滤波、FFT等(5)片上系统

将整个系统做在一个芯片上 优点

a)通过改变内部工作电压，降低芯片功耗 b)减少芯片对外的引脚数，简化制造过程

c)减少外围驱动接口单元及电路板之间的信号传递，加快微处理器数据处理的速度 d)内嵌线路可避免信号传递时所造成的系统杂讯 联发科推出28nm双核处理器MT6572 a)代号武松，基于Cortex-A7架构 b)主频为1.2GHz c)处理器上整合了Wi-Fi、FM收音机、GPS以及蓝牙四种功能 d)支持500万像素摄像头(6)嵌入式处理器的发展趋势 a)内部结构

SoC设计，与DSP、Flash、FPGA融合；性能更强，集成更多的功能部件；双核或多核结构 b)功耗更低 c)可靠性更高 d)支持ISP、ISD 1.5 嵌入式操作系统

(1)操作系统的概念及功能

操作系统。是一组计算机程序的集合，用来有效地控制和管理计算机的硬件和软件资源，并为用户提供方便的应用接口。

功能：处理器管理；存储器管理；设备管理；文件管理；用户接口(2)操作系统的分类 按程序调度的方法分为

 顺序操作系统。只含一个运行程序，独占CPU时间，顺序执行。如DOS系统。 分时操作系统。系统内同时有多道程序运行。如Unix系统。 实时操作系统。

从应用角度，嵌入式操作系统可分为  面向低端信息家电

 面向高端信息家电  面向个人通信终端  面向通信设备  面向汽车电子  面向工业控制

从实时性的角度，嵌入式操作系统可分为  具有强实时特点的嵌入式操作系统  具有弱实时特点的嵌入式操作系统  没有实时特点的嵌入式操作系统(3)实时操作系统（RTOS）

是具有实时性且能支持实时控制系统工作的操作系统，首要任务是调度一切可利用的资源来完成控制任务。

对现场不停监测，一旦有事件发生能立即处理。与通用OS的区别：实时性，代码尺寸小。一般包括以下几个重要组成部分：

 实时内核：任务管理、定时器管理、存储器管理、任务间通信与同步等。 网络组件  文件系统  图形用户界面

IEEE的Unix委员会规定了实时操作系统须具备以下几个特点：  支持异步事件的响应。 中断和调度任务的优先机制。 支持指令性计划占式调度。 支持同步。

(4)常见的嵌入式操作系统 嵌入式Linux  实时的嵌入式Linux：如RT-Linux、KURT-Linux等。RT-Linux将通常的Linux任务优先级设为最低。

 一般的嵌入式Linux：如μCLinux。 开源，内核小、效率高，可定制 μC/OS和μC/OS-II  μC/OS-II(MicroController Operating System)是由Jean J.Labrosse开发的实时操作系统内核。

 已被移植到Intel、ARM、Motorola等公司的81种不同的处理器上。

 Labrosse用一年时间开发了μC/OS实时操作系统； 1992年在《Embedded System Programming》上 发表介绍文章，并公布源代码；1993年写了《μC/OS, The Real-Time Kernel》；书及源码推动了μC/OS-II的发展。

 μC/OS-II只是一个实时操作系统的内核，全部核心代码只有8.3 KB。 只包含进程调度、时钟管理、内存管理和进程间的通信与同步等基本功能。Windows CE  多线程、完整优先权、多任务的32位嵌入式操作系统。 基本内核大小至少为200KB。VxWorks  美国WindRiver公司于1983年设计

 是目前嵌入式系统领域中使用最广泛、市场占有率最高的系统。 拥有良好的持续发展能力和高性能的内核及友好的用户开发环境。

 支持多种处理器，如ARM、x86、i960、SunSparc、MIPS RX000、PowerPC、StrongARM等。 以良好的可靠性和实时性，广泛应用在通信、军事、航空、航天等领域。

 应用案例：美国F-

16、FA-18战斗机，B-2隐形轰炸机，“爱国者”导弹，1997年4月在火星登陆的火星探测器。Palm OS  32位嵌入式操作系统，由3Com公司的Palm Computing部门开发。

 在掌上电脑和PDA市场上占有很大的市场份额，曾占据90%的PDA市场份额。 2024年04月29日惠普12亿美元收购Palm QNX  一个实时、可扩充的操作系统。

 部分遵循POSIX（可移植操作系统接口）相关标准。

 内核仅提供4种服务：进程调度、进程间通信、底层网络通信和中断处理。 内核非常小巧（QNX4.X约为12KB），运行速度极快。iOS  苹果推出，基于Darwin  最新：iOS8 Android  Google开发的基于Linux平台的开源手机操作系统 Delta OS  是电子科技大学实时系统教研室和北京科银京成技术有限公司联合研制并开发的全中文嵌入式操作系统。

 绝大部分代码由C语言编写。

 已成功应用于通信、网络、信息家电等多个应用领域。pSOS ISI公司研发的产品。

模块化、高性能、完全可扩展。

1.6 嵌入式系统的应用及发展趋势

(1)嵌入式系统的应用领域  消费电子领域  通信网络领域

 工业控制领域、机器人领域  交通管理与汽车电子领域  国防与航空航天领域  医疗仪器领域

(2)嵌入式系统的发展趋势

 小型化、智能化、网络化、可视化  多核技术的应用  低功耗、绿色环保

 云计算、可重构、虚拟化等技术被进一步应用到嵌入式系统中  嵌入式系统软件将逐渐PC化  融合趋势

 微控制器MCU与SoC的结合  微控制器MCU与DSP的结合  ARM与DSP的结合

 微控制器MCU与CPLD/FPGA的结合  安全性 ARM体系结构

教学目的：使学生对ARM设计思想、ARM处理器系列、ARM体系结构、ARM处理器模式、ARM内部寄存器、ARM体系的存储系统有一定的了解。

教学重点：ARM设计思想、ARM体系结构、ARM处理器模式、ARM内部寄存器。教学难点：ARM处理器模式、ARM内部寄存器。教学方法与教学手段：课堂讲授，多媒体教学。教学时间：4课时。教学内容

2.1 ARM设计思想

(1)RISC 传统的CISC(Complex Instruction Set Computing)指令集中，约20%指令占整个程序代码的80%。RISC(Reduced Instruction Set Computing)是一种设计思想，其目标是设计出一套能在高时钟频率下单周期执行、简单而有效的指令集。

RISC设计重点在于降低硬件执行指令的复杂度，而传统的CISC更侧重于硬件执行指令的功能性，使CISC指令变得复杂。

(2)RISC设计思想的实现

 指令集。减少了指令种类，指令只实现简单的功能，指令长度固定。

 流水线。指令的处理过程被拆分成几个更小的、能够被流水线并行执行的单元。

 寄存器。更多通用寄存器，可存数据和地址，可为所有数据操作提供快速的局部存储访问。 load-store结构。处理器只处理寄存器中数据，用load和store指令完成寄存器和外存间的数据传送

(3)ARM的设计思想  较小的核。降低功耗

 高的代码密度。考虑成本和物理尺寸限制

 较小的处理器内核管芯(Die)面积。留给外设电路的空间较大  硬件调试技术(4)ARM微处理器的特点

 体积小，功耗低，成本低，性能高；  支持Thumb/ARM指令集,兼容8/16位器件；

 大量使用寄存器，指令执行速度更快；  大多数数据操作都在寄存器中完成；  寻址方式灵活简单，执行效率高；  指令长度固定。

2.2 ARM体系结构分析

(1)包含典型 的RISC 体系结构特征

 统一寄存器文件加载/存储体系结构，数据处理操作只针对寄存器内容；  简单寻址模式，所有加载/存储地址只通过寄存器内容和指令字段确定。 指令长度固定，简化了指令译码。(2)还提供

 可组合使用转换与算术或逻辑运算指令  自动递增和自动递减寻址模式，可优化程序循环  加载存储多个指令以最大化数据吞吐量  几乎所有指令都采取条件执行的方式(3)普林斯顿结构和哈佛结构

普林斯顿结构：也称冯·诺伊曼结构，它将程序指令存储器和数据存储器合并在一起的存储结构。ARM7系列基于普林斯顿结构。

哈佛结构：将程序指令存储和数据存储分开的存储结构。ARM9系列之后都基于哈佛结构。

(4)流水线

ARM7的三级流水线在执行单元完成了大量的工作，执行单元的工作往往占用多个时钟周期，从而成为系统性能的瓶颈。

ARM9采用哈佛架构，避免了数据访问了取指的总路线冲突，采用五级流水线设计。

五级流水线技术把三级流水线中的执行单元进一步细化，减少了在每个时钟周期内必须完成的工

作量，进而允许使用较高的时钟频率。

无论三级流水线还是五级流水线，在以下情况下都会发生阻塞：  多周期指令、跳转分支指令  中断发生

 相邻指令之间的寄存器冲突：如果当前指令（A）的目的操作数寄存器和下一条指令（B）的源操作数寄存器一致，B指令就需要等A回写之后才能译码。

(5)ARM体系结构的版本

2.3 ARM处理器系列

(1)ARM处理器系列

 经典ARM处理器：ARM7、ARM9、ARM10、ARM11  Cortex-A系列处理器：Cortex-A15、Cortex-A9、Cortex-A8、Cortex-A7、Cortex-A5  Cortex-R系列处理器：Cortex-R7、Cortex-R5、Cortex-R4  Cortex-M系列处理器：Cortex-M4、Cortex-M3、Cortex-M1、Cortex-M0+、Cortex-M0、CMSDK、CMSIS  SecurCore处理器(2)ARM内核版本命名规则

(3)ARM7系列微处理器 a)主要特点

 最高主频：130MIPS(Million Instructions Per Second)；  功耗低；

 代码密度高，兼容16位的微处理器；  可得到广泛的操作系统和实时操作系统支持；  众多的开发工具，优秀的调试机制；  采用3级流水线结构；

 提供0.25μm、0.18μm和0.13μm的生产工艺。b)冯·诺伊曼结构：数据和指令使用同一条总线。

c)包括ARM7TDMI、ARM7TDMI-S、ARM7EJ-S和ARM720T四种类型，适用于不同的市场要求 d)ARM7TDMI  ARM公司最早为业界普遍认可并广泛应用的处理器核。 T：Thumb；D：Debug；M：Multiplier；I：Embedded ICE logic。e)ARM7TDMI-S  是ARM7TDMI的可综合(synthesizable)版本（软核）。

 ARM以“软”核的方式把ARM7TDMI核授权给处理器厂商，处理器厂商可进行修改和综合。这就是ARM7TDMI-S。

 综合出的整个核比“硬”核大50%，电源效率降低50%。f)ARM7EJ-S  是可综合的、带有增强型DSP（E变种）和Java加速功能（J变种）的32位RISC嵌入式处理器。

 主要用于数字音频播放器、带Java功能的无线手持设备、喷墨打印机、数码相机和PDA等方面。

g)ARM720T  专为使用Windows CE、Symbian OS操作系统平台设计。 主要用于数字音频播放器、喷墨打印机和数码相机等。(4)ARM9系列微处理器 a)主要特点

 5级整数流水线；

 单一的32位AMBA(Advanced Microcontroller Bus Architecture)总线接口；  MMU支持Windows CE、Symbian OS、Linux等；  支持实时操作系统，包括VxWorks；  统一的数据Cache和指令Cache；

 提供0.25μm、0.18μm和0.13μm的生产工艺。

b)包括ARM9TDMI、ARM920T、ARM940T和ARM9E四种类型。后三种含有Cache。

c)采用Harvard体系结构

 指令与数据分开存储。 采用指令快存 和数据快存。

d)在相同工艺条件下，ARM9TDMI的处理能力是ARM7TDMI的两倍。e)ARM920T 

Motorola MC9328MX1和Samsung S3C2410X处理器都采用ARM920T 核心。 主要应用于通信终端、3G基带和应用处理器、基于OS的平台设备、数码相机、音频/视频解码和机顶盒等。

f)ARM940T  与ARM920T相比，实现了一个更小的D-Cache、I-Cache和MPU。适于不需运行操作系统的平台。

g)ARM9E  使用单一的处理器核，提供微控制器、DSP、Java应用系统的解决方案；  DSP指令集；

 在0.13μm工艺下，主频可达300MIPS的性能；  集成实时跟踪调试功能；  可选的VFP9浮点处理协处理器；

 高性能的AHB(Advanced High performance Bus)。(5)ARM10系列微处理器 a)主要特点

 6级流水线；

 在典型的0.13μm工艺下，主频可达400MIPS的性能；  单一的32位AMBA 总线接口；

 MMU支持Windows CE、Symbian OS、Linux等；  统一的数据Cache和指令Cache；

 提供0.25μm、0.18μm和0.13μm的生产工艺；  并行读取/写入部件。

b)包括：ARM1020、ARM10200、ARM1020E、ARM1022E、ARM1026EJ-S。c)使用ARM10TDMI处理器核，采用ARMv5T结构。d)ARM10TDMI  在相同工艺条件下，处理能力是ARM9TDMI的两倍；

 采用提高时钟频率、6级流水线、转移预测逻辑、64位存储器和无阻塞的存/取逻辑等措施提升性能。

e)ARM10E。新节能模式，64位Load/Store体系，与ARM10TDMI相比具有的特点

 DSP指令集；

 可选的VFP10浮点处理协处理器；

 在实时控制和三维图像处理时，主频可达650MFLPS(百万次浮点运算每秒)。

(6)ARM11系列微处理器

 ARM1156T2-S内核、ARM1156T2F-S内核、ARM1176JZ-S内核和ARM11JZF-S内核  ARM1156T2-S内核和 ARM1156T2F-S内核  基于ARM v6指令集体系结构；

 是首批含有ARM Thumb-2内核技术的产品。(7)Cortex-A系列微处理器

 适于高计算要求、运行丰富操作系统及提供交互媒体和图形体验的应用领域。 支持传统 ARM、Thumb指令集和新的高性能紧凑型 Thumb-2 指令集。 移动互联网的支持

 低功率设计，支持 Adobe Flash 10.1  高性能 NEON 引擎，广泛支持媒体编解码器  高性能

 Cortex-A15：为新一代移动基础结构应用和无线基础结构应用提供高性能的解决方案。

 Cortex-A9。800 MHz1 GHz 的频率下，提供的性能超过 2024 DMIPS。

 Cortex-A5 低成本实现，在 400-800 MHz 的频率下，提供的性能超过 1200 DMIPS，是尺寸最小、功耗最低的 ARM 多核处理器。

 多核技术：Cortex-A15、Cortex-A5和Cortex-A9 处理器都支持 ARM 第二代多核技术。 高级扩展

 Thumb-2，提供最佳代码大小和性能。 TrustZone，安全扩展，提供可信计算。

 Jazelle 技术，提高执行环境（如 Java、.Net、MSIL、Python 和 Perl）速度。

(8)Cortex-R系列微处理器

 为具有严格的实时响应限制的深层嵌入式系统提供高性能计算解决方案。

 快速。以高时钟频率获得高处理性能。

 确定性。处理在所有场合都必须符合硬实时限制。 安全。系统必须可靠且可信。

 成本效益。在处理器及其内存系统中都具有竞争力的成本和功耗。 应用领域

 智能手机

 企业系统：硬盘驱动器、联网和打印  消费电子：机顶盒、数字电视和播放器  医疗行业、工业和汽车行业的可靠系统  功能集

(9)Cortex-M系列微处理器

 向上兼容的高能效、易于使用的处理器

 针对成本和功耗敏感的 MCU 和终端应用的混合信号设备进行过优化。 更低的功耗，更长的电池寿命。

 高密度指令集，更小的代码，更低的硅成本。

RISC 处理器内核高性能32 位CPU具有确定性的运算低延迟3 阶段管道Thumb-2 技术16/32 位指令的最佳混合小于8 位设备3 倍的代码大小对性能没有负面影响低功耗模式集成的睡眠状态支持多电源域基于架构的软件控制嵌套矢量中断控制器(NVIC)低延迟、低抖动中断响应不需要汇编编程以纯C 语言编写的中断服务例程工具和RTOS 支持广泛的第三方工具支持Cortex 微控制器软件接口标准(CMSIS)最大限度地增加软件成果重用CoreSight调试和跟踪JTAG 或2 针串行线调试(SWD)连接支持多处理器支持实时跟踪(10)SecurCore系列微处理器

 为安全要求较高应用设计。智能卡

 SIM、ID、银行业、付费电视、公共交通、电子政务

2.4 ARM处理器模式

(1)32位ARM处理器工作状态

32位ARM处理器有三种工作状态  ARM状态。对应32位ARM指令集  Thumb状态。对应16位Thumb指令集  Jazelle状态。对应8位的Jazelle指令集

用于在处理器指令层次对JAVA加速

只有进入特定的状态，相应的指令集才有效。CPSR的J(Jazelle)和T(Thumb)位反映程序的状态。Thumb-2  与现有 ARM 和 Thumb 解决方案向后兼容，同时扩展了 Thumb 指令集的可用功能；  使用少于 31% 的内存以降低系统成本；  提供比现有高密度代码高出 38% 的性能。(2)64位ARM处理器工作状态

ARMv8架构

两种主要执行状态：AArch64, AArch32 AArch64：引入了一套新的指令集“A64”专门用于64位处理 AArch32：兼容现有的32位ARM指令集 ARMv8架构支持三个主要指令集

A32(或 ARM)：32 位固定长度指令集

T32(Thumb)：以 16 位固定长度指令集的形式引入，在引入 Thumb-2 技术时增强为 16 位和 32 位混合长度

A64：提供与 ARM 和 Thumb 指令集类似功能的64位固定长度指令集(3)ARM处理器运行模式

 用户模式(usr)：ARM处理器正常的程序执行状态。 快速中断模式(fiq)：用于高速数据传输或通道处理  外部中断模式(irq)：用于通用的中断处理。 管理模式(svc)：操作系统使用的保护模式。

 数据访问终止模式(abt)：当数据或指令预取终止时进入该模式，可用于虚拟存储及存储保护。

 系统模式(sys)：运行具有特权的操作系统任务。

 未定义指令中止模式(und)：当未定义的指令执行时进入该模式，可用于支持硬件协处理器的软件仿真。

运行模式可通过软件改变，也可通过外部中断或异常处理改变。用户模式之外的模式称为非用户模式或特权模式。除用户模式和系统模式之外的5种称为异常模式，常用于处理中断和异常、访问受保护的系统资源等情况。

2.5 ARM体系的异常处理

(1)ARM体系中3种控制程序执行流程的方式：  顺序执行  跳转分支指令

 异常中断：处理器暂时中断当前数据流的现象。(2)对异常的响应。ARM处理器执行完当前指令后：  进入与特定的异常相应的操作模式；

 将引起异常指令的下一条指令的地址保存到新模式的R14中；

 将CSPR原值保存到新模式的SPSR中；

 通过设置CSPR的第7位来禁止IRQ。如果为FIQ中断，则还要设置CSPR的第6位来禁止FIQ；

 给PC强制赋向量地址值。(3)中断向量表

指定了异常中断及其处理程序的对应关系，它通常存放在存储地址的低端。

大小为32字节，其中每个异常中断占据4字节空间，用于存放一个跳转指令或者一个向PC寄存器中赋值的指令。

(4)异常优先级：当几个异常中断同时发生时，就必须按照一定的次序来处理这些异常中断。(5)从异常返回

将连接寄存器LR的值减去相应偏移量后送到PC中； 将SPSR复制回CPSR中；

若在进入异常处理时设置了中断禁止位，则要清除。通过普通指令控制PC返回

软件中断的返回指令 MOVS R15, R14;将链接寄存器内容移入PC并转换模式 IRQ，FIQ和预取异常终止中断的返回指令 SUBS R15, R14, #4 数据终止异常的返回指令 SUBS R15, R14, #8;异常在导致异常的指令的下一条指令后产生

2.6 ARM内部寄存器

(1)寄存器结构

32位ARM有31个32位通用寄存器，6个状态寄存器。通用寄存器可用来保存数据和地址信息，用R为前缀加寄存器序号表示

15个通用寄存器（R0~R14）、一个或两个状态寄存器及程序计数器可在任意时间和处理器模式下被访问，有些处理器模式拥有自身独立的寄存器

(2)通用寄存器

分成三类：

R0~R7：未分组寄存器。每个未分组寄存器在所有的处理器模式下都表示同一个物理寄存器。R8~R14：分组寄存器。每个分组寄存器与一个用户模式的寄存器对应。R15：程序计数器PC。

分组寄存器R8~R14可分为两组：

R8~R12：每个寄存器对应两组不同的物理寄存器，一组是FIQ模式下的，记为R8\_fiq~R12\_fiq, 另一组是除FIQ模式外的：R8\_usr~R12\_usr。

R13~R14：分别对应6个不同的物理寄存器。用户模式和系统模式共用一个寄存器，另外5个对应其余5种。

R13 \_ R14 \_

可以是以下几种模式之一：usr、svc、abt、und、irq和fiq。

R13：被称为堆栈指针SP，但没有任何指令强制性使用R13作为堆栈指针

R14又被称为链接寄存器LR(Link Register)。当调用子程序时，返回地址被自动保存到R14。由于ARM采用了多级流水线技术，所以当正常读取PC值时，该值为当前指令地址值加8，或是加12。

2.7 ARM体系的存储系统

(1)地址空间

将存储器看作是从零地址开始的字节的线性组合 0-3字节：第1个存储的字数据 4-7字节：第1个存储的字数据 依次排列(2)存储器格式 大端格式

小端格式

(3)存储器访问对准

无论取指还是内存访问都以字、半字或字节对准访问 a)非对齐的指令预取操作

ARM状态：将一个非对齐地址写入PC，数据的第0位和第1位被忽略，PC的bit[1:0]为0 Thumb状态：数据的第0位被忽略，PC的bit[0]为0 b)非对齐地址内存的访问操作（LOAD/STORE操作）

执行结果不可预知

忽略字单元地址低两位的值，半字单元最低位的值（分别对应访问字和半字）

在LDR和SWP指令中，对存储器访问忽略造成地址不对齐的低地址位，然后使用这些低地址位控制装载数据的循环 ARM指令系统及程序设计基础

教学目的：使学生对ARM指令系统及程序设计的基础知识有一定的了解。教学重点：ARM指令系统。

教学难点：ARM程序设计的基础知识。教学方法与教学手段：课堂讲授，多媒体教学。教学时间：4课时。教学内容

3.1 ARM寻址方式

(1)寻址方式

处理器根据指令中给出的地址信息寻找物理地址的方式。寻找操作数或操作数地址的方式。

(2)立即寻址

也叫立即数寻址，操作数本身在指令中给出，该操作数被称为立即数。例如： ADD R0, R0, #1;R0R0+1 立即数须以“#”为前缀，对于十六进制表示的立即数，还需在“#”后加上“0x”或“&”。(3)寄存器寻址

操作数存在寄存器中。例如： ADD R0, R1, R2;R0R1+R2(4)寄存器间接寻址

以寄存器的值作为操作数地址。例如： ADD R0, R1, [R2]

;R0  R1+[R2](5)基址变址寻址

a)基址加偏移的寻址方式

将寄存器内容与指令中给出的地址偏移量相加，得到操作数的有效地址。如： LDR R0, [R1, #4];R0  [R1+4]

前变址：基址加变址作为操作数地址。

后变址：基址作为操作数的地址，传送后自动更新基址寄存器的值。

b)基址加索引的寻址方式

将基址寄存器的值与索引寄存器的值相加，形成操作数的有效地址。例如： LDR R0, [R1, R2]

;R0  [R1+R2] c)多寄存器寻址

一条指令可完成多个（最多16个）寄存器值的传送。例如：

LDMIA R0, {R1, R2, R4};R1  [R0] 22

;R2  [R0+4];R4  [R0+8](6)寄存器移位寻址

操作数为寄存器中数做相应的移位而得到 例如：

ARM中的移位或循环移位操作：  LSL：逻辑左移（Logical Shift Left） LSR：逻辑右移（Logical Shift Right）ADD

R0, R1, R2, LSL #3;R0R1+8×R2  ASR：算术右移（Arithmetic Shift Right）。移位过程中保持符号位不变，若源操作数为正数，则字的高端空出的位补0。若源操作数为负数，则字的高端空出的位补1。

 ROR：循环右移（Rotate Right）。从字的最低端移出的位依次填入字的高端空出的位。 RRX：扩展的循环右移（Rotate Right Extended）。操作数向右移一位，左侧空位由状态寄存器C位填充。当移位的类型为RRX时，无需指定移位的位数，其它的则须指定移位的位数。(7)相对寻址

以PC当前值作为基地址，指令中的地址标号作为位移量，两者相加后得到操作数的有效地址。例如：

BL

NEXT

MOV „„

PC, LR

;从子程序返回 NEXT

;跳转至子程序 NEXT „„

(8)堆栈寻址

a)堆栈：按先进后出(FILO)的方式工作，使用堆栈指针(Stack Pointer, SP)指示当前操作位置。b)根据栈指针的指向位置可将堆栈分为

 满堆栈：SP指向最后压入堆栈的数据。 空堆栈：SP指向下个将放入数据空位置

c)根据堆栈的生成方式可将堆栈分为

 递增堆栈(Ascending Stack)：堆栈由低地址向高地址生成。

 递减堆栈(Descending Stack)：由高地址向低地址生成。d)ARM支持四种类型堆栈工作方式

满递增堆栈：SP指向最后压入的数据，且由低地址向高地址生成。满递减堆栈：SP指向最后压入的数据，且由高地址向低地址生产。

空递增堆栈：SP指向下个将放入数据的空位置，且由低地址向高地址生成。空递减堆栈：SP指向下个将要放入数据的空位置，且由高地址向低地址生成。

3.2 ARM指令集

(1)ARM指令集分类 加载/存储指令

数据处理指令 分支指令

状态寄存器访问指令 异常/中断指令 协处理器指令(2)ARM指令的特点 所有指令都是32bit；

大多数指令都在单周期内完成； 所有指令都可以条件执行； load/store体系结构；

指令集可以通过协处理器扩展。(3)ARM指令的格式

(4)ARM指令的条件执行

所有ARM指令都可包含一个可选的条件码，只有当CPSR中条件标志位满足指定条件时，指令才会被执行。否则以NOP指令通过流水线。

(5)加载/存储指令

Load：将内存中数据装载到寄存器 Store：将寄存器中的数据存入内存 a)单寄存器传输指令

{} {B} Rd, addressing1 {} SB | H | SH Rd, addressing2 {} {B} Rd, LABEL LDR/STR指令：寄存器在前，地址在后 LDM/STM指令：地址在前，寄存器在后

LABEL：相对PC的寻址方式。编译器在汇编时，会将标号LABEL汇编成PC的偏移量存入该指令的立即数字段。

B: Byte；S: Sign；H: Half LDR/STR：读/写一个32bit字到/从一个32位寄存器，要求读/写地址字对齐。LDRB：内存8bit字节32bit寄存器；不要求地址对齐，寄存器高24位清零。STRB：寄存器低8位内存的某个地址；不要求地址对齐。

LDRH：16bit半字 32bit寄存器；要求地址半字对齐，寄存器的高16bit清零。STRH：寄存器低16bit内存；要求地址半字对齐。

LDRSH：有符号16bit半字32bit寄存器中；要求地址半字对齐，寄存器高16bit根据符号位扩展。

LDRSB：有符号8bit字节32bit寄存器中；不要求地址对齐，寄存器高24bit根据符号位扩展。

“数据”一列指的是这条指令所访问的存储单元

“基址寄存器”一列的内容是指执行指令之后“基址寄存器”的内容 b)多寄存器传输指令

{} Rn{!}, {^} LDM / STM：从由基址寄存器指示的一片连续存储器到寄存器列表所指示的多个寄存器之间传送数据。

!：表示执行完操作后将变化之后的地址值写入基址寄存器

^：对于LDM操作，如恢复的寄存器中含有PC(R15)寄存器，则指令执行的同时CPU自动将SPSR拷贝到CPSR中，例如：

LDMFD {R0-R12, LR, PC}^ 数据的传送发生在User用户模式下的寄存器，而非当前模式寄存器，例如：

LDMDB SP, {R0-LR}^ 例如：

STMIA R0!, {R1-R5};以R0为地址指针，将R1-R5保存到内存，指针向上移动

c)交换指令

SWP {B} {} Rd, Rm, [Rn] SWP：字交换；SWPB：字节交换 Rd  [Rn], [Rn]  Rm 例如：

SWP R1, R2, [R3];R1[R3] ,[R3] R2 SWP R1, R1, [R2];R1与[R2]内容互换(6)数据处理指令

a)数据传送指令

MOV {cond} {S} Rd, Operand 例如： MOV R1, R0 MOVEQ PC, R14

;将R14值传到PC b)数据取反传送指令 MVN {cond} {S} Rd, Operand 例如：

MVN R1, #2;将立即数2取反送至R1 c)算术运算指令

ADD、ADC ADD|ADC {cond} {S} Rd, Oper1, Oper2 例如： ADDS ADCS ADCS ADC R0, R4, R8 R1, R5, R9;加低端的字;带进位加第二字

R2, R6, R10;带进位加第三字 R3, R7, R11;带进位加第四字

可实现 128位加法 SUB：减法指令

SUB {cond} {S} Rd, Oper1, Oper2 SBC：带借位减法指令

SBC {cond} {S} Rd, Oper1, Oper2 例：SBCS R0, R1, R2;R0=R1–R2–C RSB：反向减法指令

RSB {cond} {S} Rd, Oper1, Oper2 操作数2减去操作数1，例如： RSB R0, R1, R2;R0 = R2 – R1 RSC：带借位的逆向减法指令 RSC {cond} {S} Rd, Oper1, Oper2 d)逻辑运算指令 AND：与

AND {cond} {S} Rd, Oper1, Oper2 ORR：或

ORR {cond} {S} Rd, Oper1, Oper2 EOR：异或

EOR {cond} {S} Rd, Oper1, Oper2 e)比较指令 CMP CMP {cond} Operand1, Operand2 CMN：反值比较 CMN {cond} Operand1, Operand2 将第一个数与第二个数的反值进行比较，相当于完成两个数相加。例如： CMN R1, R0;R1+R0，并设置CPSR f)测试指令 TST TST {cond} Operand1, Operand2

把两个操作数按位进行与运算，根据结果更新CPSR。

TEQ TEQ {cond} Operand1, Operand2 把两个操作数按位进行异或运算，根据结果更新CPSR g)乘法指令 MUL MUL {cond} {S} Rd, Oper1, Oper2 RdOperand1×Operand2。例如： MUL

R0, R1, R2 MLA MLA {cond} {S} Rd,Oper1,Oper2,Oper3 RdOperand1×Operand2+Operand3 例如：

MLA R0, R1, R2, R3;R0 = R1×R2+R3;R0 =R1×R2 UMULL：64位无符号数乘法指令 UMULL {cond} {S} Rd\_L, Rd\_H, Operand1, Operand2 Rd\_H:Rd\_L Operand1×Operand2 例如： UMULL R0, R1, R2, R3;R0 =(R2×R3)的低32位;R1 =(R2×R3)的高32位 UMLAL： 64位无符号数乘加指令 UMLAL {cond} {S} Rd\_L, Rd\_H, Operand1, Operand2 Rd\_H:Rd\_L+= Operand1×Operand2 例如：

UMLAL

R0, R1, R2, R3;R0 =(R2×R3)的低32位+R0;R1 =(R2×R3)的高32位+R1 SMULL：64位有符号数乘法指令 SMULL {cond} {S} Rn\_L, Rn\_H, Operand1, Operand2 Rd\_H:Rd\_L Operand1×Operand2 SMLAL： 64位有符号数乘加指令 SMLAL {cond} {S} Rn\_L, Rn\_H, Operand1, Operand2 Rd\_H:Rd\_L+= Operand1×Operand2(7)分支指令

a)B(分支指令)和 BL(带链接分支指令)B {cond} LABEL BL {cond} LABEL

允许向前或向后跳转最高32MB。

BL：带返回的分支指令，用于调用一个将其返回地址存入链接寄存器的函数。例如：

SUBPRG BL …… SUBPRG „„; MOV PC, LR;返回

b)BX(分支并可选地交换指令集)和BLX(带链接分支并可选地交换指令集)BX {cond} Rm BLX {cond} LABEL | Rm Rm：转移地址，bit[0]为0时，目标地址处为ARM指令，否则为Thumb指令。BX 和 BLX是唯一可使用的切换指令集的方法。

BX 和 BLX指令执行与B和BL指令相同的分支操作，并可从ARM指令集切换到THUMB指令集。

c)长跳转

通过向PC寄存器中写目标地址值，可实现在4GB地址空间中进行任意跳转。例如： MOV MOV LR, PC;保存返回地址

R15, #0x00110000;无条件转向0x110000(8)状态寄存器访问指令 a)MRS MRS {} Rd, 将状态寄存器的内容传送到通用寄存器中。例如： MRS b)MSR MSR {} \_, Rm MSR {} \_,# 表示下列情况之一： c：控制域，即PSR[7:0] PSR[32:24] CPSR|SPSR  通用寄存器。例如： MSR CPSR\_F, #&F00000000

R3, CPSR;CPSRR3

x：扩展域，即PSR[15:8] s：状态域，即PSR[23:16] f：标志域，即

;设置所有标志位

(9)异常/中断指令（续）a)SWI SWI {} 产生SWI异常中断，以实现在用户模式下调用操作系统的监控功能程序。它将处理器置于管理(SVC)模式。例如：

SWI 0x01

;调用编号为01的系统例程

b)BKPT BKPT 产生软件断点中断，可用于程序调试。例如 BKPT 0xF010(10)协处理器指令 a)CDP CDP {} , , CRd, CRn, CRm {,} 编号为CP#的协处理器接受指令并执行。具体操作由Cop1和Cop2定义，CRn和CRm为源操作数，结果CRd。例如：

CDP P6, 2, C5, C10, C3, 16

;激活协处理器P6的操作，操作码1和操作码2值分别为2和16，目标寄存器为C5，源操作数寄存器为C10和C3 b)MRC和MRC MRC | MCR {} , , CRd, CRn, CRm {,} 在ARM与协处理器寄存器间传送数据。MRC：ARM寄存器协处理器寄存器 MCR：协处理器寄存器ARM寄存器 例如： MCR P3, 3, R0, C4, C5, 6;ARM处理器R0协处理器P3的寄存器C4和C5 c)LDC和STC 在协处理器寄存器与存储器间传送数据 LDC：协处理器寄存器存储器

STC：存储器协处理器寄存器 前变址格式

LDC | STC {} {L} , CRd, [Rn, ]{!} 后变址格式

LDC | STC {} {L} , CRd, Rn, 3.3 Thumb指令集

(1)Thumb指令集的特点

 是ARM指令集压缩形式的子集，所有Thumb指令均有对应的ARM指令。 采用16位二进制编码，代码密度小。

 执行Thumb指令时，先动态解压缩，然后作为标准的ARM指令执行。 如何区分指令流取决于CPSR的位T。 大多Thumb数据处理指令采用2地址格式。 移位操作变成单独指令。

 没有协处理器指令、单寄存器交换指令、乘加指令、64位乘法指令及程序寄存器处理指令。 仅分支指令B有条件执行功能。

(2)Thumb状态切换

a)ARM状态进入Thumb状态 执行带状态切换的转移指令BX。例如： BX R0;若R0最低位为1，则转Thumb状态

异常返回。通常用于返回到进入异常前所执行的指令流，而不是特地用于切换到Thumb模式。适用于异常前执行的是Thumb指令。

b)Thumb 状态进入ARM状态 执行BX指令。

利用异常进入ARM指令流。(3)Thumb指令寄存器的使用  对R0~R7具有全部访问权限。

 R8~R12的访问受到限制，只能通过MOV、ADD和CMP访问。

 在每一种特权模式下都有一组SP、LR和SPSR，分别对应ARM状态的R13、R14和R15。(4)Thumb指令与ARM指令的相似点  Load/Store结构。

 支持8位字节、16位半字和32位字数据类型。 半字以2字节边界对准，字以4字节边界对准。(5)Thumb指令与ARM指令差异点

 跳转指令：条件跳转在范围上有更多的限制，转向子程序只具有无条件转移。

 数据处理指令：对通用寄存器进行操作，操作结果需放入其中一个操作数寄存器。因此，许多Thumb指令采用2地址格式，ARM数据处理指令大多采用3地址格式。 单寄存器加载和存储指令：只能访问R0~R7。

 PUSH和POP指令使用堆栈指针R13作为基址实现满递减堆栈，除R0~R7外，PUSH指令还可存储R14，POP指令可加载程序指令PC。

 大多数Thumb指令是无条件执行的，所有ARM指令都是条件执行的。 由于采用高密度编码，Thumb指令格式没有ARM指令格式规则。(6)Thumb分支指令

a)B 指令——目标为Thumb代码 B 偏移量左移1位

B指令是Thumb指令集中唯一可条件执行的指令。PC = PC +(SignExtend(signed\_immed\_8)

3.4 3.5 35

3.6 嵌入式系统设计与开发过程

(1)嵌入式软件开发的特点 a)需要交叉开发环境

 交叉开发环境：实现编译、链接和调试应用程序代码的环境，它分散在有通信连接的宿主机与目标环境之中。

 宿主机(Host)是一台通用计算机，一般是PC机。

 目标机(Target)可以是嵌入式应用软件的实际运行环境，也可以是能替代实际环境的仿真系统。

 交叉软件开发工具包括：交叉编译器、交叉调试器和模拟软件等。b)引入任务设计方法

 嵌入式应用系统以任务为基本执行单元  用多个并发的任务代替通用软件的多个模块 c)需要固化程序 d)软件开发难度大

 实时性。

 稳定性、可靠性、抗干扰性。(2)软硬件平台的选择 选择处理器需考虑的因素  处理性能  技术指标、功耗

 软件支持工具、是否内置调试工具  供应商是否提供评估板 软件平台的选择

 操作系统。开发工具，与硬件接口难度，内存，是否提供硬件驱动程序，可裁剪性、实时性等

 编程语言。通用性、可移植性、执行效率、可维护性等  集成开发环境

(3)嵌入式系统设计与开发流程  需求分析  体系结构设计

 硬件的设计、制作及测试  软件的设计、实现及测试  系统集成

 系统性能测试及可靠性测试

STM32微控制器

教学目的：使学生对STM32的性能指标、硬件结构有一定的了解。教学重点：STM32的硬件结构。教学难点：STM32的硬件结构。

教学方法与教学手段：课堂讲授，多媒体教学。教学时间：4课时。教学内容

4.1 STM32微控制器的性能指标

(1)STM32F103的主要功能

 ARM Cortex-M3核  最高主频72MHz  512kB Flash、64kB RAM  1个USB、1个CAN、5个USART、3个SPI、2个I2C、2个I2S、1个SDIO、112个GPIO

 3个ADC、2个DAC、8个定时器

 FSMC总线（支持NOR，NAND，SRAM） 12通道DMA控制器

 Serial wire debug(SWD)、JTAG 接口  启动：用户Flash、系统存储器、SRAM(2)JC-STM32CB核心板

 核心芯片：ST STM32F103ZET6  8M NOR Flash：SST39VF6401，16bit  128M NAND Flash：K9F1G08U0C，8bit  8MHz CPU晶振，32768Hz RTC晶振  1个10M/100M以太网接口  1路音频输出和1路音频输入  1个10-pin JTAG调试接口  1个4-pin SWD串行调试接口  Reset按钮、自定义LED，电源LED  1个精密可调电阻连接到片内ADC输入

 使用CPLD扩展IO口及地址译码，型号EPM240T100C5N  板载10/100M网卡控制器，型号DM9000AEP，16bit总线接口

STM32

**第四篇：嵌入式系统设计报告**

嵌入式系统设计实验报告

班 级：学 号：姓 名：成 绩：指导教师：

20090612 2009112107 侯金钟 武俊鹏、刘书勇 1.实验一

1.1 实验名称

嵌入式系统硬件开发环境

1.2 实验目的

1．熟悉UP-net3000实验平台。

2.超级终端设置及BIOS 功能使用。

1.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

1.4 实验内容及要求

熟悉UP-net3000实验平台的硬件电路和外设，ARM JTAG的安装和使用，利用超级终端检验外设的工作状态。

1.5 实验设计与实验步骤

１．建立工程

（1）运行ARM SDT 2.5 集成开发环境（ARM Project Manager）.（2）在新建的工程中，如图1A-2 所示，选中工程树的“根部”。

（3）因为开发板上的嵌入式处理器ARM7TDMI 没有浮点处理器，所以，如图1A-3 所

示，在弹出的对话框中设置Floating Point Processor 为none，并保持其他的设置不变。（4）选中工程树的“根部”，通过菜单Project | Tool Configuration for work1.apj | asmlink | Set，对整个工程的连接方式进行设置。（5）在弹出的对话框中，选中Entry and Base 标签，如图1A-4 所示，设置连接的Read-Only（只读）和Read-Write（读写）地址。

（6）选择Linker Configuration 的ImageLayout 标签，（7）选择Project | Edit Project Tamplete 菜单，弹出Project Template Editor 对话框。

（8）选择Project | Edit Variables for work1.apj，弹出Edit Variables for work1.apj 对话框。

2．进行程序的在线仿真、调试

1.6 实验过程与分析

熟悉UP-net3000实验平台的硬件电路和外设，安装了ARM JTAG，利用超级终端检验了外设的工作状态。

1.7 实验结果总结

软件安装成功，结果显示正常。

1.8 心得体会

通过此次试验，我对ARM的环境的功能有一定的了解与完善。对试验台有了基本的认识与使用。

2.实验二

2.1 实验名称

嵌入式系统软件开发环境

2.2 实验目的

1.熟悉ADS1.2 开发环境，学会ARM 仿真器的使用。

2.使用ADS 编译、下载、调试并跟踪一段已有的程序，了解嵌入式开发的基本思想和过程。

2.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

2.4 实验内容及要求

本次实验使用ADS 集成开发环境。新建一个简单的工程文件，并编译这个工程文件。学习ARM 仿真器的使用和开发环境的设置。下载已经编译好的文件到嵌入式控制器中运行。学会在程序中设置断点，观察系统内存和变量。

2.5 实验设计与实验步骤

（1）运行ADS1.2 集成开发环境（CodeWarrior for ARM Developer Suite）。

（2）在新建的工程中，选择Debug 版本，使用Edit | Debug Settings菜单对Debug 版本进行参数设置。（3）在Debug Settings 对话框中选择Target Settings 项。在Post-linker一栏中选择ARM from ELF。

（4）在Debug Settings 对话框中选择ARM Linker 项

（5）在第四步中如果选择简单的地址连接设置，在Debug Settings 对话框中选择ARM Linker 项

（6）回到的工程窗口中，选择Release 版本，使用Edit | Release Settings 菜单对Release 版本进行参数设置。（7）参照第（3）、（4）、（5）、（6）步在Release Settings 对话框中设置Release版本的Post-linker、连接地址范围、入口模块和输出文件。（8）回到如图1B-3 所示的工程窗口中，选择Targets 选项卡，如图1B-11 所示。选中DebugRel 版本，按Del 键将其删除。DebugRel 子树是一个折衷版本，通常用不到，所以在这里删除。

2.6 实验过程与分析

1）回到工程窗口选中Debug 版本，执行菜单Project | Make 对工程进行编译连接。（2）在ADS 中执行菜单Project | Debug 启动ADS1.2 的调试工具AXD。（3）在AXD 中执行菜单Options | Configure Target 对AXD 进行设置。（4）点Select 按钮选择远程连接为ARM ethernet driver，点Configure 按钮输入仿真器的IP 地址。

（5）等待程序装载完毕以后，通过Execute | Go 菜单以及Execute | Stop（或者工具栏中的相应按钮）运行或暂停程序。程序暂停后在窗口中将显示出程序暂停的位置。（6）通过Execute | Step 菜单（或者工具栏中的相应按钮）可以单步运行程序。

（7）程序停止后可以通过Processor Views | Sources 菜单查看源文件，并可在适当位置按F9 设置端点。

（8）使用在Processor View 菜单下的Registers、Variables 和Memory 命令可以查看工作寄存器或者内存变量。读者可以逐一地尝试，为以后调试程序打下基础。

2.7 实验结果总结

超级终端输出“Hello World!”。

2.8 心得体会

基本了解了ADS1.2的配置条件，学会了ARM仿真器的使用方法。

3.实验三

3.1 实验名称

键盘及LED实验

3.2 实验目的

1．学习键盘及LED 驱动原理。

2．掌握ZLG7289芯片串行接口的使用方法，用ZLG7289芯片驱动17键的键盘和8个共阴极LED。

3.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

3.4 实验内容及要求

通过ZLG7289芯片驱动17键的键盘和8个共阴极LED，将按键值在LED上显示出来。要求从右至左循环显示至少四位数字。

基本功能实现之后可考虑实现从左至右显示四位及四位以上数字（最大八位），并可设置清零键等扩展功能。

3.5 实验设计与实验步骤

利用所给的基础代码进行调试，观察输出结果，结合指导书和教材掌握基本原理和源代码的编写方式。根据实验的要求设计函数流程，并反复调试，实现功能。1．新建工程，将“Exp3 键盘及LED 驱动实验”中的文件添加到工程。2．定义ZLG7289 寄存器（ZLG7289.h）

#define ZLG7289\_CS(0x20)//GPB5 #define ZLG7289\_KEY(0x10)//GPG4 #define ZLG7289\_ENABLE()do{ZLG7289SIOBand=rSBRDR;ZLG7289SIOCtrl=rSIOCON;rSIOCON=0x31;rSBRDR=0xff;rPDATB&=(~ZLG7289\_CS);}while(0)#define ZLG7289\_DISABLE()do{rPDATB|=ZLG7289\_CS;rSBRDR=ZLG7289SIOBand;rSIOCON=ZLG7289SIOCtrl;}while(0)3.编写ZLG7289 驱动函数（ZLG7289.c）4．定义键盘映射表：（Keyboard16.c）

unsigned char KeyBoard\_Map[]= {4,8,11,0,0,0,0,0,5,9,12,15,1,0,0,0,6,10,13,16,2,3,0,0,7,0,14,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};//64 键值映射表，通过查找键盘映射表来确定键盘扫描码对应的按键值。

5．定义键值读取函数。（Keyboard16.c）6．编写主函数，将按键值在数码管上显示。

3.6 实验过程与分析

利用键盘驱动函数实现基本数字输入，然后利用循环左移函数实现输入数字做一样功能，并通过改变函数中相应delay的值来消除键盘按键带来的抖动。

3.7 实验结果总结

按键值可以在LED上显示出来。要求从右至左循环显示八位数字，同时可以复位清零。达到实验的效果。

3.8 心得体会

通过本次实验，我了解了LED的显示屏幕的数字的移位功能，这个功能不止可以用一个方法实现，而循环左移是其中比较巧妙且省力的一种，在遇到类似情况的时候，可以优先考虑能不能利用到文件中已经存在的函数，这样可以不必自己编写函数，省时省力。是一种可靠的方法。4.实验四

4.1 实验名称

电机转动控制及中断实验

4.2 实验目的

1．熟悉ARM本身自带的六路即三对PWM，掌握相应寄存器的配置。

2．编程实现ARM系统的PWM输出和I/O输出，前者用于控制直流电机，后者用于控制步进电机。

3.了解直流电机和步进电机的工作原理，学会用软件的方法实现步进电机的脉冲分配，即用软件的方法代替硬件的脉冲分配器。

4.了解44B0处理器上中断的应用。5.学习在44B0处理器上中断的应用。

6.进一步熟悉平台外围硬件及其驱动程序的编写。

4.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

4.4 实验内容及要求

1．学习步进电机和直流电机的工作原理，了解实现两个电机转动对于系统的软件和硬件要求。学习ARM 知识，掌握 PWM 的生成方法，同时也要掌握 I/O的控制方法。

2．编程实现ARM芯片的一对PWM输出用于控制直流电机的转动，通过A/D旋钮控制其转动方式。

3．编程实现ARM的四路I/O通道，实现环形脉冲分配用于控制步进电机的转动，通过A/D旋钮控制步进电机的转角。

4．通过键盘控制直流电机与步进电机的切换。5．设置并启动定时器。

6．设置中断，编写定时器中断服务程序，对中断次数进行计数并在LED上显示结果。

4.5 实验设计与实验步骤

1．添加并打开工程。

2．进行直流电机初始化设置和代码编写。3．进行步进电机初始化设置和代码编写。

4．对Timer3编程，编写定时器中断服务程序，完成对中断次数的计数。5．编写LED计数显示函数，使LED能正确计数并显示0-9999。6．编写中断初始化函数和中断响应函数。7．终端下载测试。

4.6 实验过程与分析 1．对直流电机进行编程和测试，掌握转速和旋转方向的设定方法。

2．对步进电机进行编程和测试，掌握ARM的四路I/O通道，实现环形脉冲分配用于控制步进电机的转动，通过A/D旋钮控制步进电机的转角。

3．对主函数进行编程，用键盘响应直流电机与步进电机的切换控制。

4．掌握中断相关语句的应用，弄清定义的中断向量、中断向量号，编写中断响应函数，并完成中断响应控制。

4.7 实验结果总结

实现了直流电机与步进电机的基本设置和控制，可以通过键盘控制电机之间的切换。完成了中断的响应和定时中断。当对其中一个旋钮转动时，就可以由直流电机转换成步进电机的转换，达到实验的效果。

4.8 心得体会

通过本次实验，我了解了直流电机和步进电机的工作原理，同时也知晓了电机间的转换过程，了解其中的道理内涵，熟悉了ARM自带的A/D转换器的工作原理及编程方法，了解了中断的意义和实现方法，实现了简单了中断处理程序。同时我也收获了很多关于ARM处理机的相关知识。

5.实验五

5.1 实验名称

触摸屏驱动实验

5.2 实验目的

1．了解触摸屏的基本概念与原理。

2．理解触摸屏与LCD的关系。3.编程实现对触摸屏的控制。

4.熟悉用 ARM 内置的 LCD控制器驱动 LCD。

5.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

5.4 实验内容及要求

1．了解触摸屏基本原理，理解对触摸屏进行输出标定、与LCD显示器配合的过程。2．通过编程实现触摸两点自动在两点间划直线。3．通过编程实现在触摸屏上动态画出曲线。5.5 实验设计与实验步骤

1．添加并打开工程。

2．在头文件中定义宏和常量及驱动函数。

#define ADS7843\_CTRL\_START 0x80 #define ADS7843\_GET\_X 0x50 #define ADS7843\_GET\_Y 0x10 #define ADS7843\_CTRL\_12MODE 0x0 #define ADS7843\_CTRL\_8MODE 0x8 #define ADS7843\_CTRL\_SER 0x4 #define ADS7843\_CTRL\_DFR 0x0 #define ADS7843\_CTRL\_DISPWD 0x3 // Disable power down #define ADS7843\_CTRL\_ENPWD 0x0 // enable power down #define ADS7843\_PIN\_CS(1<<6)//GPF6 #define ADS7843\_PIN\_PEN(1<<5)//GPG5 /////////触摸屏动作//////// #define TCHSCR\_ACTION\_NULL 0 #define TCHSCR\_ACTION\_CLICK 1 //触摸屏单击 #define TCHSCR\_ACTION\_DBCLICK 2 //触摸屏双击 #define TCHSCR\_ACTION\_DOWN 3 //触摸屏按下 #define TCHSCR\_ACTION\_UP 4 //触摸屏抬起 #define TCHSCR\_ACTION\_MOVE 5 //触摸屏移动

#define TCHSCR\_IsPenNotDown()(rPDATG&ADS7843\_PIN\_PEN)(ADS7843\_CTRL\_START|ADS7843\_GET\_X|ADS7843\_CTRL\_12MODE |ADS7843\_CTRL\_DFR|ADS7843\_CTRL\_ENPWD)//采样x 轴电压值，数据为12 位，参考电压输入模式为差分模式，允许省电模式

#defineADS7843\_CMD\_Y(ADS7843\_CTRL\_START|ADS7843\_GET\_Y|ADS7843\_CTRL\_12MODE |ADS7843\_CTRL\_DFR|ADS7843\_CTRL\_ENPWD)int TchScr\_Xmax=1840,TchScr\_Xmin=176, TchScr\_Ymax=195,TchScr\_Ymin=1910;//触摸屏返回电压值范围 #defineADS7843\_CMD\_X 3．校准触摸屏坐标，进行坐标转换。

4．实现触屏取点并显示功能。

将触摸动作及触摸点坐标在超级终端上显示出来。5．实现两点间自动划线功能。6．实现触摸屏动态划线功能。

可以使用TchScr\_GetScrXY()函数（第三个参数为0）来获得液晶屏的x、y 方向的电压

范围，分别点触摸屏有效面积的左上角和右下角，得到下列参数：

TchScr\_Xmax=1840;TchScr\_Xmin=176;TchScr\_Ymax=195;TchScr\_Ymin=1910;//此数值仅供参考，请以实际校对为准

5.6 实验过程与分析 1．在定义触屏响应功能的函数中对点击触屏进行响应函数的修改，在其中添加修改点颜色的函数，修改得到的触摸点的颜色，并显示在LCD上。

2．获取第一个点坐标并储存，获取第二个点坐标并储存，由编写的划线函数取得储存的两点间直线上所有点的坐标，并对其改变颜色，显示在LCD上，即完成划直线功能。

3．将划线函数应用到响应触屏移动消息的函数下，即可对连续获得的触摸坐标进行连续的画短直线，连接成曲线，完成动态划线功能。

5.7 实验结果总结

了解了触摸屏响应动作消息的函数的工作原理，通过修改实现了触摸屏响应不同动作进行画点、划线、动态划线的功能。验证触摸屏的灵敏度的实验。

5.8 心得体会

通过这次实验，我基本掌握了通过编程驱动触摸屏以及触摸屏响应时间，实现了触摸屏对不同动作消息的响应。同时也知晓了在触摸屏上的描点画线的实现，但是由于不知名的原因，描点画线的误差较大，位置偏差较大，同时触摸屏有时会不灵敏，出现时好时坏的现象，但是由于我们的辛勤钻研，最终克服了这个困难，实现了最后的触摸屏的实现。

6.实验六

6.1 实验名称

UCOS-Ⅱ在ARM微处理器上的裁剪

6.2 实验目的

1．了解UCOS-Ⅱ内核的主要结构。

2.掌握UCOS-Ⅱ裁剪的基本原理与嵌入式编程实现方法。3.学习如何根据具体情况对UCOS-Ⅱ操作系统进行裁剪。

4．通过对UCOS-Ⅱ配置文件(OS\_CFG.H)中相关的配置常量进行设置，实现对UCOS-Ⅱ的裁剪。

6.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

6.4 实验内容及要求

对UCOS-Ⅱ内核进行裁剪并移植到ARM7微处理器上。

6.5 实验设计与实验步骤 1.按照要求，载入STARTUP目录下文件，完成系统初始化、环境配置。2.载入UCOS-Ⅱ的全部源码，与处理器架构相关的文件位于arch目录下。3.在os\_cpu.h中编写与处理器和编译器相关的代码。

4.编写os\_cpu\_c.c等6个与操作系统相关的函数。5.编写os\_cpu.asm等4个与处理器相关的函数。

6.6 实验过程与分析

按照实验步骤进行，得到了需要的系统。

6.7 实验结果总结

按照要求进行了裁剪，得到了满足需要又紧凑的应用软件系统。

6.8 心得体会

通过本次实验，我了解了UCOS-Ⅱ内核的主要结构，掌握UCOS-Ⅱ裁剪的基本原理与嵌入式编程实现方法，学会了如何根据具体情况对UCOS-Ⅱ操作系统进行裁剪。

7.实验七

7.1 实验名称

UCOS-Ⅱ在ARM微处理器上的移植和编译

7.2 实验目的

1．了解UCOS-Ⅱ内核的主要结构。

2.掌握将UCOS-Ⅱ内核移植到ARM7处理器上的基本方法。

7.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

7.4 实验内容及要求

1.将UCOS-Ⅱ内核进行移植到ARM7微处理器上。

2.编写两个简单任务，在超级终端上观察两个任务的切换。

7.5 实验设计与实验步骤

1．该实验的文件分为两类，其一是 STARTUP 目录下的系统初始化、配置等文件，其二是 UCOS-Ⅱ的全部源码，arch 目录下的 3 个文件是和处理器架构相关的。

2.设置 os\_cpu.h 中与处理器和编译器相关的代码。

3.用 C 语言编写 6 个操作系统相关的函数（OS\_CPU\_C.C）。4.用汇编语言编写 4 个与处理器相关的函数（OS\_CPU.ASM）。5.编写一个简单的多任务程序来测试一下移植是否成功。6.编译并下载移植后的 UCOS-Ⅱ。

7.6 实验过程与分析

1.按照实验步骤进行，将µC/OS-II 内核移植到了ARM7 微处理器上。2编写了两个简单任务，在超级终端上观察两个任务的切换。

7.7 实验结果总结

将µC/OS-II 内核顺利移植到了ARM7 微处理器上。

7.8 心得体会

通过本次实验，使我更加了解了µC/OS-II 内核的主要结构，掌握了ARM的C语言和汇编语言的编程方法，了解了ARM7处理器结构，掌握了将µC/OS-II 内核移植到ARM 7 处理器上的基本原理与嵌入式编程实现方法。

8.实验八

8.1 实验名称

综合实验

8.2 实验目的

对前七次实验进行总结，应用之前所学的知识，将前几次实验内容结合起来，完成键盘，LED，触摸屏，直流电机，步进电机各种功能的组合。实现一个较为全面的功能结构。

8.3 实验环境

硬件：ARM 嵌入式开发平台、用于ARM7TDMI 的JTAG 仿真器、PC 机Pentium100 以 上、串口线。

软件：PC 机操作系统win98、Win2000 或WinXP、ARM SDT 2.51 或ADS1.2 集成开发 环境、仿真器驱动程序、超级终端通讯程序。

8.4 实验内容及要求

对前七次实验进行总结，应用之前所学的知识，完成自拟的嵌入式系统，要求综合前期基础实验的各种功能。

8.5 实验设计与实验步骤

1．添加并打开工程。2．进行LCD设计，在LCD显示屏输出文本。

3．进行LED及键盘设计，完成LED输出显示功能和键盘输入功能。4．进行电机控制设计，完成键盘控制电机转动功能。5．进行中断设计，完成定时中断功能。

6．进行触摸屏设计，完成触摸屏感应和划线功能。7．进行裁剪和移植功能设计和完成。

8.6 实验过程与分析

1．完成LCD显示功能，在LCD显示屏上输出文本：“Hello World！”。2．进入界面触屏控制选择功能，实现触屏选择功能。

3．自定义四种种功能，第一为电机控制，从键盘读取命令，并将功能编号显示在LED上，LED显示的是计数的数据，同时旋转按钮完成电机转速的控制选择。

4．第三种功能为划线，功能编号显示在LED高四位上，同时LCD屏幕清屏，为划线功能做准备，可以实现划线功能。

5．第四种功能为定时中断，当由键盘控制时，LED显示数值清零，实现了中断。6．实现裁剪与移植功能。

8.7 实验结果总结

完成了各种基本功能，并通过自拟的系统将各种功能整合起来，完成了一个小的嵌入式系统，对前七次的功能有了更深入的了解。通过LED的计数，当在计数值在前30秒之内，由键盘控制LED的数值及显示，按键盘上的某一个按键，实现对LED上的数值清零，后30秒由触摸屏控制清零，并且在前30秒之内，旋转按钮，实现对直流电机的运转，并且到步进电机的转换。

8.8 心得体会

通过这次实验，我更加深刻地掌握了前几次实验中的基本功能的实现方法，并且把几种功能联合在一起，实现一些功能，把LED 显示屏，LCD触摸屏，键盘，直流电机，步进电机等等设备联合在一起，对该实验有一定的帮助与提高，而我和我的队友也对嵌入式系统有了更深入的了解，在此期间，也学习了关于ARM处理器的开发与实践，了解了关于手机的嵌入式设备的产生过程，我也深深的对此充满了兴趣，对未来的嵌入式课程设计奠定了深厚的基础，可是令我遗憾的是，我和队员的水平所限，没有完成中断优先级的控制。本来想完成更多的功能，可是最后由于时间紧迫，也有一些其他的事情来分神分心，所以就只能完成这些，但是在未来的几周内，嵌入式课程设计也给了我们很大的空间去做未完成的事情。我相信我们会做的更加完美，功能更强大，用于未来的生活中去实践。

**第五篇：《嵌入式系统设计》教学大纲**

《嵌入式系统设计》课程教学大纲

嵌入式系统设计(Design of Embedded Systems)

一、课程教学目的和基本要求

教学目的：

嵌入式系统技术已被广泛地应用于工业控制系统、信息家电、通信设备、医疗仪器、智能仪器仪表等众多领域。如手机、PDA、MP3、手持设备、智能电话、机顶盒等，可以说嵌入式系统无处不在。本课程讲述当前主流的嵌入式处理器—ARM的微处理器和实时操作系统µC/OS，介绍嵌入式系统的基本原理和应用及设计方法。通过本课程的学习，着重培养学生的实际动手能力，使学生掌握ARM的系统结构、指令系统、程序设计方法、系统扩展方法、应用及开发技术等。使学生在工作中具有利用嵌入式系统开发产品和解决实际问题的基本能力。

基本要求：

1、掌握嵌入式系统技术的基本原理。

2、了解并掌握ARM的系统结构、指令系统、程序设计方法、系统扩展方法、应用及开发技术等。

3、比较熟练地用ADS进行ARM实验箱的编程及调试。

4、了解并掌握实时操作系统µC/OS，并学会用µC/OS开发小型应用系统。

二、先修课程及本课程总学时

先修课程：C程序设计语言、单片机原理 本课程总学时：36

三、课程章节及各章节学时安排

第一章 嵌入式系统概述（2学时）

第二章 嵌入式系统硬件基础（2学时）

第三章 设计方法及ADS集成开发环境（4学时）

第一节 嵌入式系统的设计方法（2学时）第二节 ADS集成开发环境（2学时）第四章 ARM7体系结构（4学时）

第一节ARM7体系结构介绍（2学时）第二节ARM7TDMI(-S)指令集简介。（2学时）第五章 ARM指令集（8学时）

第一节 ARM处理器寻址方式。（2学时）第二节ARM指令集。（2学时）第三节ARM汇编程序设计。（2学时）第四节C与汇编混合编程。（2学时）第六章 硬件结构与功能（12学时）第一节 LPC2106/2105/2104结构（2学时）第二节 GPIO及引脚连接模块。（2学时）

第三节 UART0和UART1、I2C接口、SPI接口。（2学时）第四节 定时器0和定时器1。（2学时）第五节 脉宽调制器（PWM）。（2学时）

第六节 看门狗及Flash存储器系统和编程。（2学时）第七章 µC/0S－II程序设计（4学时）

第一节 µC/0S－II简介及移植（2学时）第二节 移植µC/0S－II。（2学时）

四、课程主要内容

第一章 嵌入式系统概述（2学时）

1、嵌入式系统简介（包括什么是嵌入式系统，嵌入式系统的特点，嵌入式系统的分类）。

2、嵌入式系统的应用领域。

3、嵌入式系统的现状和发展趋势。

4、嵌入式微处理器和嵌入式操作系统。

5、嵌入式系统中的一些重要概念。难点：什么是嵌入式系统。

第二章 嵌入式系统硬件基础（2学时）

1、嵌入式系统的总体结构。

2、与嵌入式系统相关的硬件知识。难点：总体结构。

第三章 设计方法及ADS集成开发环境（4学时）第一节 嵌入式系统的设计方法

包括嵌入式系统的设计流程，嵌入式系统的一般设计方法，嵌入式系统的硬件/软件协同设计技术。第二节 ADS集成开发环境

1、ADS 1.2 集成开发环境的组成

2、工程的编辑

3、工程的调试

4、仿真器的安装与应用 难点：调试、仿真器。

第四章 ARM7体系结构（4学时）第一节ARM7体系结构介绍

1、处理器模式。

2、内部寄存器。

3、程序状态寄存器。

4、异常。

第二节ARM7TDMI(-S)指令集简介。难点：异常。

第五章 ARM指令集（8学时）第一节 ARM处理器寻址方式。第二节ARM指令集。

第三节ARM汇编程序设计。第四节C与汇编混合编程。难点：ARM汇编程序设计。

第六章 硬件结构与功能（12学时）第一节 LPC2106/2105/2104结构

1、存储器寻址。

2、系统控制模块。

3、存储器加速模块。

4、向量中断控制器。

第二节 GPIO及引脚连接模块。

第三节 UART0和UART1、I2C接口、SPI接口。第四节 定时器0和定时器1。第五节 脉宽调制器（PWM）。

第六节 看门狗及Flash存储器系统和编程。

难点：存储器重新映射，编程实现功能部件的初始化和控制。本章实验：LPC2106/2105/2104功能部件基础实验。

第七章 µC/0S－II程序设计（4学时）第一节 µC/0S－II简介及移植

1、移植规则。

2、编写LPC2106/2105/2104的启动代码。第二节 移植µC/0S－II。难点：移植µC/0S－II。本章实验：学习移植µC/0S－II。

五、教材及主要参考书

教材：

《嵌入式系统设计与开发》，马文华，科学出版社，2024年

主要参考书：

[1] 田苗苗．嵌入式控制系统．北京航空航天大学出版社，2024年 [2] 吕 骏 译．嵌入式系统设计．电子工业出版社，2024年

[3] 周立功等．ARM微控制器基础与实战．北京航空航天大学出版社，2024年

[4] 李驹光等．ARM应用系统开发祥解．清华大学出版社，2024年

[5] 劭贝贝译．uCOS-II－源代码公开的实时嵌入式操作系统．中国电力出版社，2024年 [6] 田苗苗．嵌入式系统设计与实例开发．清华大学出版社，2024年 [7] 袁勤勇等译．嵌入式系统构件．机械工业出版社，2024年 [8] 田苗苗．《实用嵌入式系统设计与开发—基于ARM微处理器与μCOS-II实时操作系统》(第二版)，清华大学出版社，2024.9 [9] 邵贝贝译．《源代码公开的实时嵌入式操作系统——μCOS-II》，北京航空航天大学出版社，2024年

六、其他需要说明的情况

1．采用多媒体教学。

2．实验课单列，每周2学时。

3．每周作业量2～3小时，主要针对基本概念、开发环境的操作和编程。

七、成绩评定

1.理论考核

占60% 2.实践考核

占40%（其中包括：实验、作业、课堂提问、学生自行设计制作的作品）。

本文档由站牛网zhann.net收集整理，更多优质范文文档请移步zhann.net站内查找