



周立功单片机公司策划

ARM[®]微控制器 基础与实战

(第2版)

周立功 等编著

 北京航空航天大学出版社

<http://www.buaapress.com.cn>

ARM 微控制器基础与实战

(第 2 版)

周立功 等编著

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内 容 简 介

以目前流行的、单片化的 LPC2100 系列 ARM7 微控制器为核心,由浅入深地介绍 ARM 微控制器的体系结构及应用编程,逐步引导读者掌握使用 ARM 的知识。分 3 部分:第 1 部分为基础篇,包括 1、2 章,主要从使用 ARM 芯片的开发者角度,论述 ARM7TDMI 的体系结构和指令系统。第 2 部分为实验篇,包括第 3~6 章,详细介绍 LPC2100 系列 ARM7 微控制器 LPC2104、LPC2114 和 LPC2119(带 CAN 控制器)的内部结构;然后分别给出 LPC2100 系列 ARM7 微控制器的各个功能部件的编程方法,包括汇编代码和 C 语言代码。第 3 部分为提高篇,包括第 7~9 章,介绍嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 在 LPC2100 系列 ARM7 微控制器上的移植方法,嵌入式系统开发平台的概念、建立方法,以及一些组成软件开发平台的软件模块的使用方法。

本书可作为高等院校嵌入式系统课程的参考用书,也可作为 ARM 应用技术开发人员的参考资料。

图书在版编目(CIP)数据

ARM 微控制器基础与实战(第 2 版)/周立功等编著. —2 版.

—北京:北京航空航天大学出版社,2005.8

ISBN 7-81077-710-6

I. A… II. 周… III. 微控制器 IV. TP332.3

中国版本图书馆 CIP 数据核字(2005)第 048561 号

ARM 微控制器基础与实战(第 2 版)

周立功 等编著

责任编辑 王慕冰

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpres@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:29.25 字数:749 千字

2005 年 8 月第 2 版 2005 年 8 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-710-6 定价:43.00 元

ARM 微控制器快速入门的“敲门砖”

学习与开发的困境

回顾以往,我国单片机的普及教育已经搞了许多年,但为什么还是有很多大学生毕业之后不能立即投入到实际的应用开发工作之中呢?通过严峻的人才需求形势可以看到:很多电子类专业的大学生毕业之后的出路只有改行,而很多企业却在感叹人才难求。年复一年如此惊人地相似,不得不引起我们对传统的教材和教学方法进行深刻的反思。单片机与嵌入式系统应用技术是一门实战性很强的学科,离开了实践就如无源之水。其实,几乎所有成才的开发工程师都是一边学习、一边实践干出来的,很少有真正的专家,包括我们业界知名的很多专家在内,是通过课堂教出来的。今天,8279、8255 还买得到吗?已经过去这么多年了,学生还在学习那些已经很早就淘汰了的器件。一个大学本科生读了四年大学,却连一个可靠的复位电路都设计不好,要从可靠性设计出发来设计产品就更无从谈起。由此可以看出,我们的教材与教学方法违背了这门学科成才的规律,我们的大学一定要彻底改变观念,尽快地从“教书型”的阴影中走出来,迅速转向“研究型”培养人才的正确轨道。

其实,我一直认为企业不是培养人才的地方,这应该是大学的责任。但我们却不得不花很多的精力用于人才的培养,而我们得到的直接好处就是从 2000 年到现在公司几乎没有出现过优秀人才跳槽的现象,即使一般的人才跳槽也很少。究其原因,在于我们一直在尝试如何培养一流人才。通过这些年的努力,我们掌握了快速、恰当地学习的方法和培养卓越人才的“秘诀”。关键是我们帮助那些年轻人找到了自己的位置和方向,而且我们也实现了自己的愿望,我和他们一样有共同的成就感。基于此,我愿意将我的经验和感想奉献出来,与大家共同分享。

谈到学习,就不得不谈人们的思想与观念,这是人才成长与成功的源泉。超前的思想意识、正确的观念、良好的心态加上正确的学习方法就可以使人达到“不用扬鞭自奋蹄”的崭新境界。所谓教书育人这些根本的东西常常容易被人所忽视,这看起来似乎是政治老师的责任,其实不然。我们应该将思想和方法教育放在首位,充分调动学习者的积极性,从而化被动为主动。效果最好的教学方法就是在实验室里上课,老师一边讲解理论,学生一边在计算机上操作和验证老师讲解的内容,以加深理解。每位学生人手一套开发实验板,一个学期下来至少可以做到考试这门课程不用复习,毕业之后对开发流程和集成开发环境可以达到烂熟于心的程度。

回顾我国的单片机推广历程,可以设想,如果当初 Intel 公司首先推出的不是 80C31 而是一颗 20 引脚的 Flash 单片机,那么中国的单片机发展之路肯定要顺利得多。回到现实中来看,我国的 ARM 推广和应用技术发展之路与单片机的启蒙是何等惊人地相似。多年来,单片机化电子产品的开发在很大程度上完全取决于开发工程师个人的经验、知识水平和智慧。大多数开发工程师在产品开发过程中的随意性、离散性都比较大,普遍缺乏系统的总结。从根本上来说,谈不上自觉地将离散的经验上升到知识的程度,更谈不上有比较清晰的项目管理和软硬件平台。无论是企业的决策者还是开发工程师,元器件的成本成为其头等大事,还未做出产

品来第一件事情就想到“价格战”，缺乏做精品的意识，缺乏保持企业可持续发展的战略眼光和思想，其实这些都是能够看得见的成本。事实上，危害更大的、看不见的“无形成本”却很少有人关注。比如，投入的人力资源是否合理？要投入多少开发费用？员工跳槽以后如何对产品进行升级和继续开发？产品如何快速上市？如何“先知先觉”地设计出高可靠性的产品等等因素，这些都是制约企业可持续发展的根本所在。

成千上万的企业家和开发工程师无不感到困惑，又欲罢不能。当我们今天面临由 8 位上升到 32 位微控制器市场时，到底如何克服过去学习和开发中所遇到的困难而少走弯路呢？

选择一个功能恰当的 ARM 微控制器

俗话说得好：万丈高楼平地起。也就是说，无论做什么事，打好基础是根本，因此要想彻底掌握 32 位 ARM 单片机和嵌入式操作系统应用技术，一定要从基础做起。那些内嵌的功能模块如 TCP/IP、CAN-bus、A/D 转换器、802.11、调制解调器、USB…有如美丽的外衣，仅仅是虚有其表，其实真正的根本还是 ARM 内核和一个基本的嵌入式操作系统内核，只有把握了“根”，所有其它的问题都将迎刃而解。

通过网站可以看到，关于 ARM 的开发套件不下 20 种，到底选择哪一种最合适？我认为，作为一个初学者，挑选之前应该目标明确，应以最低的代价得到最大的收获。因为学好之后，将来做哪方面的产品开发到现在还没有完全定性。还有用多长的时间可以快速学会，这也是很重要的，因为起步不顺利会严重打击学习者的积极性。

事实上，有很多从事单片机应用开发的工程师可能一辈子都不会用到串行通信技术，甚至做了多年的开发工作之后，还是没有深入地掌握串行接口应用技术，但却丝毫不影响他成为一个实实在在的单片机应用工程师。即使有一天在工作中要用到串行通信技术，我们通过长期的学习和开发所积累的“经验与知识”也足以帮助我们快速掌握串行通信技术。所以学习基于 ARM 内核的单片机应用技术同样如此，不应该过早地、过多地将精力用在 TCP/IP 等复杂功能模块和 μ Clinux 等复杂的嵌入式操作系统内核的研究和学习上，这些诱人的“热点”很容易使人迷失方向，进而忽视对基础知识的深入学习，到头来说得头头是道，可真的干起来却离实际的需求差得太远。我们一定要知道，知识是永远也学不完的——学海无涯、人生苦短！我们学习的目的是为了满足不同应用的需求。那么到底如何在有限的人生历程之中花最低的代价“创造”最高的价值呢？常言道：伤其十指不如断其一指。即是说：我们做任何事情首先一定要“集中优势兵力”击中要害，然后再根据实际需要“各个击破”。我们常常所说的“不战而胜”，“取法其上，得乎其中”其实讲的就是方法。只要方法得当，掌握 ARM 和嵌入式操作系统应用技术其实十分简单。因此，以我们的学习体会和开发经验为基础，郑重向读者推荐 PHILIPS 公司的 LPC2100 系列 ARM7 微控制器，作为读者入门首选的 ARM 微控制器。

选择一个简单易懂的嵌入式操作系统

从过去到现在，学习操作系统的人越来越多，而真正掌握和搞通操作系统的人却越来越少。很多人从本科到博士都一直在学习操作系统，讲起来也头头是道，考试也不乏满分，可真正要他写一个简单的操作系统却又很困难。为什么？因为他在学习过程中根本就没有得到过很好的训练，大多数人也从来没有自己动手去写过一个哪怕错误百出但真正能够跑起来的操作系统，尽管大家天天在那里呼吁：我们中国人要有自己知识产权的操作系统，到头来还不是

一句空话。学习 ARM 微控制器也如此。如果不引起我们整个产业界的重视,如果没有了一个良性循环,最终的结局还是落后,那是早已写好的程序——“死循环”。

当然,有人会说,没有必要去研究嵌入式操作系统的源码,同样也能够做出产品。这一点也不假。而事实上,只有真正地掌握嵌入式操作系统,才能达到随心所欲的境界。无论嵌入式操作系统多么复杂,无论代码多大,其机理都是完全一样的,因此选择哪一个嵌入式操作系统作为快速入门的“敲门砖”,就很有讲究了。我认为,首先要简单,有源码且稳定;其次,学了之后要能够实用。从目前来看,可以选择的嵌入式操作系统有 μ Clinux 和 μ C/OS-II。对于这两个嵌入式操作系统,不同的人有不同的看法,可以说在各种网站的讨论社区形成了两大阵营,各持己见,对于初学者来说莫衷一是、无所适从。

其实,我并不反对选择使用 μ Clinux,相反我还提倡将 μ Clinux 用于产品,这是未来的发展方向,但现实的情况如何呢?并非每个人和每个单位都有这个能力。首先,由于 μ Clinux 代码太大,对于大多数人来说,要完全或者基本上读懂不是一件容易的事情。其二,网上确实也有很多公开的中间件源码,但并不保证拿来就可以稳定、可靠地使用,还是需要进行二次开发,对于大多数人来说,这是一个不小的困难。其三,尽管讨论社区也很活跃,但却很难找到真正的技术支持,这是一个不可避免的事实。因此如果没有实力和优秀的开发团队,切不可贸然选择 μ Clinux 作为企业的开发平台。是不是开发产品就不能选择 μ Clinux? 也不绝对,在一般的条件下,企业最好选择成熟的 μ Clinux 开发平台和软硬件模块。商场如战场,一切从头开发不可取,对于企业和个人来说,回避风险从来就是第一原则。如果产品开发失败,严重的后果就是企业倒闭;对于个人来说,就失去了一次本来可以成功表现自己的机会。

对于初学者,我推荐源码公开的 μ C/OS-II 嵌入式操作系统。 μ C/OS-II 已经有很多产品成功使用的案例且得到了美国军方的认证,北京航空航天大学出版社也出版了配套的专著。国内熟悉这个嵌入式操作系统的开发人员特别多,十分容易通过网站上的讨论社区得到帮助。更关键的是,我们很容易通过阅读源码达到彻底掌握嵌入式操作系统的目的。最好的办法是通过阅读 μ C/OS-II 之后,自己写一个简单的能够跑起来的 OS,这是我们公司培训相关人才的一个成功的经验。我们在北京航空航天大学出版社出版的《嵌入式实时操作系统 Small RTOS51 原理及应用》一书就是通过学习积累的成果,我们开发 EasyARM2104、EasyARM2100 开发套件和编写本书的工程师都是这样走过来的。事实证明:通过深入地学习和解剖 μ C/OS-II 嵌入式操作系统是迅速培养嵌入式系统应用人才的成功之路。

要有深入浅出的配套教材

如果仅仅是购买了一个开发套件而没有相应的配套技术资料,学习的效果可想而知是很差的。因为只有配套的资料才能体现出设计者的原创思想,更为重要的是,若在学习过程中遇到了困难,配套的技术资料可以帮助加深理解,解决问题,我们可以看看设计者是怎样解释这些问题的。本书是 EasyARM2104、EasyARM2100 开发套件设计者一边做,一边积累,并再加工的原创技术资料,非常实用。其精髓就是使用了简单易懂的语言和例题来解释复杂的技术难题,这也是本书最主要的特色。

良好的技术支持也是必不可少的

当然,购买产品不仅得到了硬件,而且也得到了软件,但一定要明白,之所以选择某个产品,更重要的是购买了“服务”。如果得不到良好的技术支持,还不如不买,因为遇到了技术难题,却找不到专家解答,这是非常痛苦的事情,得到的效果往往是事倍功半。这样,浪费的不仅仅是时间,更重要的是失去了稍纵即逝且可以展翅飞翔的机遇。

周立功

2003年9月3日

修订版前言

本书从 2003 年 11 月出版以来,得到了广大读者的好评和支持,先后印刷了 3 次,总共印数达 12 000 册。PHILIPS 公司推出了几款新的 LPC2100 系列 ARM7 微控制器,其功能特点比 LPC2106/2105/2104 更强,应用更为广泛,为了使读者可结合实际应用来学习 ARM,特修订本书。

修订版全书共有 9 章,删除了第 1 版中第 4 章“EasyARM2104 开发实验板”的全部内容,因为修订版介绍的实验程序与开发板无关。由于可使用专用的工程模板来建立工程,第 1 版中的第 7 章“移植代码的使用”已没有多大意义,所以修订时删除了这一章的内容。删除了第 1 版中第 8 章“移植代码新增的函数手册和配置手册”的全部内容,因为新的 $\mu\text{C}/\text{OS} - \text{II}$ 移植不再使用操作系统与应用代码分别编译的方法,相关的接口函数不需要介绍。修订版将第 1 版的第 9~12 章进行压缩,仅保留与用户应用相关的中间件接口函数和应用范围,在修订版的第 8 章“嵌入式系统开发平台”中介绍。

第 4 章的内容改为可加密的、带 A/D 转换器的 LPC2114/2124 芯片硬件结构。由于 LPC2114/2124 具有 LPC2106/2105/2104 的所有片内外设,而且控制寄存器基本相同,所以本章着重介绍的是 LPC2114/2124 新增的功能。

第 5 章的内容改为可加密的、内置 CAN 控制器的 LPC2119/2129 芯片硬件结构。本章主要介绍的是 CAN 控制器功能部件及其应用电路设计、编程实例。

第 6 章介绍 LPC2100 系列 ARM7 微控制器各个功能部件的编程方法,与第 1 版的第 5 章相比,这些程序不针对某一个开发板,而是直接给出了相关部分的实验电路,另外还增加了 A/D 实验程序。

第 7 章介绍如何把 $\mu\text{C}/\text{OS} - \text{II}$ 移植到 ARM7 体系结构上以及如何在 LPC2000 系列 ARM7 微控制器上使用移植代码,修正了第 1 版第 6 章介绍的移植不合理的地方。

第 8 章为新增的“嵌入式系统开发平台”,介绍嵌入式开发平台的概念,然后从开发平台的角度介绍用于 LPC2000 系列 ARM7 微控制器的软件模块(中间件)接口函数和使用方法。

参与本书写作、策划及其 EasyARM2104 与 EasyARMARM2100 开发套件设计的工作主要人员有陈明计、戚军、黄绍斌、岳宪臣、朱旻、李仕兵等人,全书由周立功负责策划、审定和统稿。

在此感谢美国 PHILIPS 半导体公司潘志强先生,上海 PHILIPS 半导体公司高级经理金宇杰先生,PHILIPS 公司亚太区 MMS 部门销售总监梅润平先生、MCU 事业部经理华果先生,香港 PHILIPS 公司的刘俊杰、郭志锐、陈华程等先生多年来的大力支持和帮助。如果没有他们长此以来对我们的支持,我们肯定不能取得今天这样的成绩。

由于作者水平有限,本书在各个方面难免有疏忽、不恰当甚至错误的地方,恳请各位同行指正。

周立功

2005 年 3 月 29 日

如果读者在阅读本书时有什么问题,或需要技术支持,可与我公司联系。具体联系方式如下:

广州周立功单片机发展有限公司(邮编: 510630)

地址: 广州市天河区天河北路 689 号光大银行大厦 16 楼 D2

电话: (020)38730916 38730917 38730976 38730977

传真: (020)38730925

电子邮箱: info@zlgmcu.com

网址: <http://www.zlgmcu.com>

技术支持

电话: (020)22644358 22644359 22644361

电子邮箱: ARM@zlgmcu.com

技术论坛: <http://www.zlgmcu.com.cn> <http://www.21ICbbs.com>

广州专卖店

地址: 广州市天河区新赛格电子城 203~204 室

电话: (020)87578634 传真: (020)87575542

深圳周立功

地址: 深圳市深南中路 2070 号电子科技大厦 A 座 22 楼 2201 室

电话: (0755)83781768 83781788 83782922 传真: (0755)83781798

成都周立功

地址: 成都市一环路南一段 57 号金城大厦 612 室

电话: (028)85499320 85437446 传真: (028)85439505

重庆周立功

地址: 重庆市石桥铺科园一路二号大西洋国际大厦 1116 室

电话: (023)68796438 68796439 传真: (023)68796439

北京周立功

地址: 北京市海淀区黄庄知春路 113 号银网中心 715 室

电话: (010)62536178 62536179 82628073 82614433

传真: (010)82614433

上海周立功

地址: 上海市北京东路 668 号科技京城东座 7E 室

电话: (021)53083452 53083453 53083496 53083497

传真: (021)53083491

杭州周立功

地址: 杭州市登云路 428 号浙江时代电子商城 275 号

邮编: 310000

电话: (0571)88009275 88009932 88009933 88271834

传真: (0571)88009274

南京周立功

地址: 南京市珠江路 280 号珠江大厦 2006 室

电话: (025)83613221 83613271 83603500 83603005

传真: (025)83613271

武汉周立功

地址: 武汉市洪山区广埠电路珞瑜路 158 号 12128 室

电话: (027)87168497 87168297 87168397 传真: (027)87163755

西安办事处

地址: 西安市长安北路 54 号太平洋大厦 1201 室

电话: (029)87881296 87881295 83063000 传真: (029)87880865

目 录

第 1 章 从程序员角度看 ARM7TDMI(-S)

1.1	简 介	1
1.1.1	ARM	1
1.1.2	ARM 的体系结构	1
1.1.3	ARM7TDMI(-S)	2
1.2	ARM7TDMI(-S)的模块和内核框图	3
1.3	体系结构直接支持的数据类型	3
1.4	处理器状态	5
1.5	处理器模式	5
1.6	内部寄存器	6
1.6.1	简 介	6
1.6.2	ARM 状态寄存器集	6
1.6.3	Thumb 状态寄存器集	9
1.7	程序状态寄存器	11
1.7.1	简 介	11
1.7.2	条件代码标志	12
1.7.3	控制位	12
1.7.4	保留位	13
1.8	异 常	13
1.8.1	简 介	13
1.8.2	异常入口/出口汇总	14
1.8.3	进入异常	14
1.8.4	退出异常	15
1.8.5	快速中断请求	15
1.8.6	中断请求	15
1.8.7	中 止	16
1.8.8	软件中断指令	16
1.8.9	未定义的指令	17
1.8.10	异常向量	17
1.8.11	异常优先级	17
1.9	中断延迟	18
1.9.1	最大中断延迟	18
1.9.2	最小中断延迟	18

1.10	复位	18
1.11	存储器及存储器映射 I/O	19
1.11.1	简介	19
1.11.2	地址空间	19
1.11.3	存储器格式	20
1.11.4	未对齐的存储器访问	21
1.11.5	指令的预取和自修改代码	22
1.11.6	存储器映射的 I/O	25
1.12	寻址方式简介	27
1.13	ARM7TDMI(-S)指令集	27
1.13.1	简介	27
1.13.2	ARM 指令集	28
1.13.3	Thumb 指令集	31
1.14	协处理器接口	32
1.14.1	简介	32
1.14.2	可用的协处理器	33
1.15	调试接口简介	33

第2章 ARM7TDMI(-S)指令集及汇编

2.1	ARM 处理器寻址方式	35
2.2	指令集介绍	38
2.2.1	ARM 指令集	38
2.2.2	Thumb 指令集	62

第3章 LPC2106/2105/2104 硬件结构与功能

3.1	简介	79
3.1.1	特性	80
3.1.2	引脚信息	81
3.2	LPC2106/2105/2104 存储器寻址	85
3.2.1	片内存储器	85
3.2.2	存储器映射	86
3.2.3	LPC2106/2105/2104 存储器重新映射和 Boot Block	88
3.2.4	预取指中止和数据中止异常	91
3.3	系统控制模块	91
3.3.1	系统控制模块功能汇总	91
3.3.2	引脚描述	91
3.3.3	晶体振荡器	92
3.3.4	寄存器描述	92
3.3.5	外部中断输入	93

3.3.6	存储器映射控制	96
3.3.7	PLL(锁相环)	97
3.3.8	功率控制	102
3.3.9	复位	104
3.3.10	VPB 分频器	104
3.3.11	唤醒定时器	105
3.4	存储器加速模块	106
3.4.1	简介	106
3.4.2	存储器加速器模块的操作模式	108
3.4.3	MAM 配置	109
3.4.4	寄存器描述	109
3.5	向量中断控制器	111
3.5.1	特性	111
3.5.2	描述	111
3.5.3	寄存器描述	111
3.5.4	VIC 寄存器	113
3.5.5	中断源	116
3.5.6	VIC 使用注意事项	118
3.6	GPIO	121
3.6.1	特性	121
3.6.2	应用	121
3.6.3	引脚描述	121
3.6.4	寄存器描述	122
3.6.5	GPIO 使用注意事项	123
3.7	引脚连接模块	124
3.7.1	简介	124
3.7.2	应用	124
3.7.3	寄存器描述	124
3.8	UART0	129
3.8.1	特性	129
3.8.2	引脚描述	129
3.8.3	寄存器描述	130
3.8.4	结 构	136
3.9	UART1	138
3.9.1	特性	138
3.9.2	引脚描述	139
3.9.3	寄存器描述	139
3.9.4	结 构	147
3.10	I ² C 接口	150

3.10.1	特 性	150
3.10.2	应 用	150
3.10.3	描 述	150
3.10.4	引脚描述	153
3.10.5	寄存器描述	153
3.10.6	结 构	158
3.11	SPI 接口	161
3.11.1	特 性	161
3.11.2	描 述	161
3.11.3	引脚描述	164
3.11.4	寄存器描述	164
3.11.5	结 构	166
3.12	定时器 0 和定时器 1	168
3.12.1	特 性	168
3.12.2	应 用	169
3.12.3	引脚描述	169
3.12.4	寄存器描述	169
3.12.5	定时器举例操作	174
3.12.6	结 构	175
3.13	脉宽调制器 PWM	177
3.13.1	特 性	177
3.13.2	描 述	178
3.13.3	引脚描述	181
3.13.4	寄存器描述	181
3.14	实时时钟	188
3.14.1	特 性	188
3.14.2	描 述	188
3.14.3	结 构	188
3.14.4	寄存器描述	189
3.14.5	RTC 中断	190
3.14.6	混合寄存器组	190
3.14.7	完整时间寄存器	192
3.14.8	时间计数器组	193
3.14.9	报警寄存器组	194
3.14.10	基准时钟分频器(预分频器)	194
3.15	看门狗	198
3.15.1	特 性	198
3.15.2	应 用	199
3.15.3	描 述	199

3.15.4 寄存器描述	199
3.15.5 方框图	201

第4章 LPC2114/2124 硬件结构

4.1 概 述	203
4.2 特 性	203
4.3 应 用	204
4.4 LPC2114/2124 功能与引脚配置	205
4.5 LPC2114/2124 最小系统原理图	209
4.6 引脚连接模块	211
4.6.1 简 介	211
4.6.2 寄存器描述	211
4.7 GPIO	213
4.8 外部中断	213
4.9 SPI 接口	217
4.10 A/D 转换器	217
4.10.1 特 性	217
4.10.2 描 述	217
4.10.3 引脚描述	218
4.10.4 寄存器描述	218
4.10.5 操 作	220
4.10.6 使用示例	220
4.11 芯片加密	221

第5章 LPC2119/2129/2194 内置 CAN 功能的 ARM

5.1 概 述	223
5.2 特 性	223
5.3 功能与引脚配置	223
5.4 带 CAN 收发器最小系统原理图	226
5.5 引脚连接模块	226
5.6 CAN 模块介绍	226
5.6.1 CAN 模块存储器映射	226
5.6.2 CAN 控制器寄存器一览表	228
5.6.3 各控制寄存器介绍	229
5.6.4 组合 CAN 寄存器	239
5.6.5 全局验收过滤器	240
5.6.6 FullCAN 模式	245
5.7 CAN 网络的硬件连接	247
5.7.1 CAN 接口板	247

5.7.2	CAN 通信电缆	247
5.7.3	CAN 接口卡和调试软件	248
5.8	CAN 的基础实验	249
5.8.1	CAN 控制器的初始化	249
5.8.2	CAN 控制器初始化示例程序	250
5.8.3	CAN 控制器发送示例程序	253
5.8.4	CAN 中断接收示例程序	254
5.9	CANstarter-II 快速开发套件	255
5.9.1	简介	255
5.9.2	特点	257
5.9.3	实例简介	257

第6章 LPC2100 系列 ARM7 微控制器基础实验

6.1	LPC2100 系列 ARM7 微控制器系统基础	259
6.1.1	系统时钟介绍	259
6.1.2	REMAP 操作及调试	261
6.1.3	启动代码说明	263
6.2	LPC2100 系列 ARM7 微控制器功能部件实战	269
6.2.1	GPIO	270
6.2.2	中 断	282
6.2.3	定时器	290
6.2.4	UART	295
6.2.5	I ² C 接口	307
6.2.6	SPI 接口	322
6.2.7	PWM	328
6.2.8	实时时钟	335
6.2.9	模/数转换器	342
6.2.10	WDT	347
6.2.11	低功耗	350
6.2.12	IAP 应用	357
6.2.13	除法运算	361
6.3	PC 机人机界面	367
6.3.1	EasyARM 软件窗口介绍	367
6.3.2	EasyARM 软件通信协议	369
6.3.3	EasyARM 应用例程	370

第7章 移植 μ C/OS-II 到 ARM7

7.1	μ C/OS-II 简介	380
7.1.1	概 述	380

7.1.2	μ C/OS-II 的特点	380
7.2	移植规划	381
7.2.1	编译器的选择	381
7.2.2	任务模式的取舍	381
7.2.3	支持的指令集	381
7.3	移植 μ C/OS-II	381
7.3.1	概 述	381
7.3.2	关于头文件 includes.h 和 config.h	383
7.3.3	编写 OS_CPU.H	383
7.3.4	编写 OS_CPU_C.C 文件	384
7.3.5	编写 OS_CPU_A.S	389
7.3.6	关于中断及时钟节拍中断	392
7.4	移植代码应用到 LPC2000	394
7.4.1	编写或获取启动代码	395
7.4.2	挂接 SWI 软件中断	395
7.4.3	中断及时钟节拍中断	395
7.4.4	编写应用程序	396

第 8 章 嵌入式系统开发平台

8.1	如何建立嵌入式系统开发平台	398
8.1.1	使用平台开发是大势所趋	398
8.1.2	建立开发平台的方法	401
8.1.3	编写自己的软件模块	401
8.2	数据队列	402
8.2.1	简 介	402
8.2.2	API 函数集	403
8.2.3	使用范例	406
8.3	串口驱动	408
8.3.1	简 介	408
8.3.2	API 函数集	408
8.3.3	使用范例	409
8.4	Modem 接口模块	412
8.4.1	简 介	412
8.4.2	Modem 的状态	412
8.4.3	API 函数集	413
8.4.4	使用范例	414
8.5	I ² C 总线模块	415
8.5.1	简 介	415
8.5.2	API 函数集	416

8.5.3 使用范例	417
8.6 SPI 总线模块	420
8.6.1 简介	420
8.6.2 API 函数集	420
8.6.3 使用范例	422

第9章 在 $\mu\text{C}/\text{OS}-\text{II}$ 中使用其它片内外设

9.1 通用 I/O 口的使用	425
9.2 定时器的使用	427
9.3 PWM 的使用	428
9.4 实时时钟的使用	429

附录 A JTAG 接口设计指南

附录 B ARM 指令集相关列表汇总

B.1 ARM 指令列表	435
B.2 Thumb 指令列表	437
B.3 汇编预定义变量及伪指令	439
B.4 其它	443

附录 C PHILIPS LPC 系列 32 位 ARM 微控制器

C.1 LPC2100 系列 32 位 ARM 微控制器	445
C.2 LPC2200 系列 32 位 ARM 微控制器	446

附录 D EasyARM2100 开发套件

D.1 简介	447
D.2 功能特点	447

附录 E EasyARM2200 开发套件

E.1 简介	449
E.2 功能特点	449
E.3 结构框图	450

参考文献