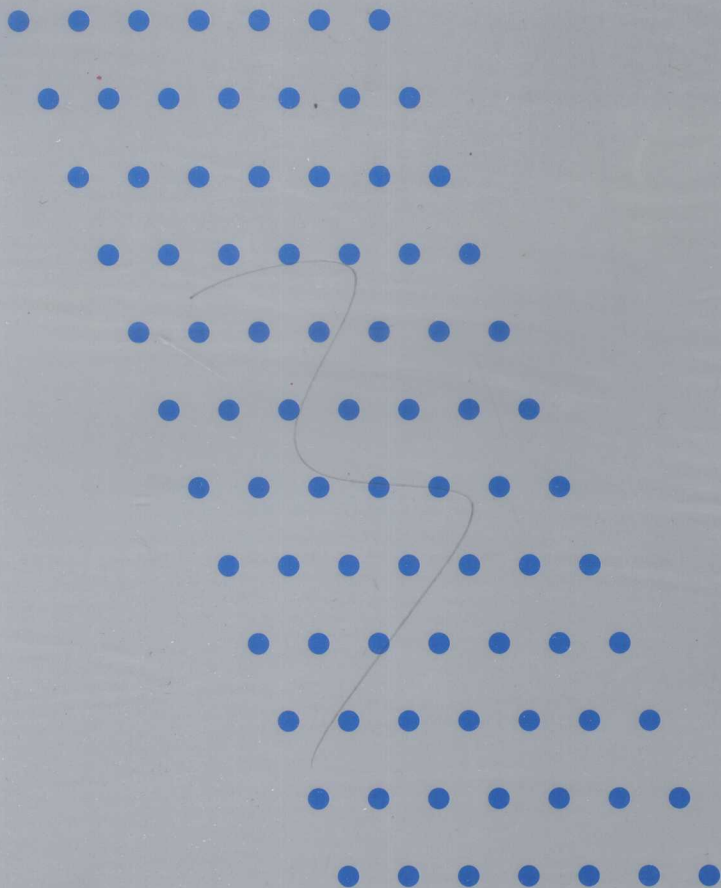
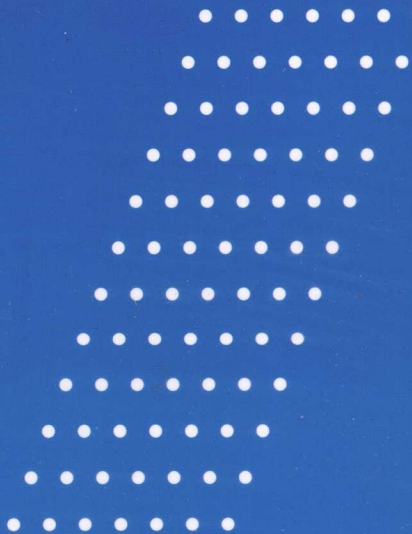


21世纪高等学校嵌入式系统专业规划教材

王黎明 刘小虎 闫晓玲 编著

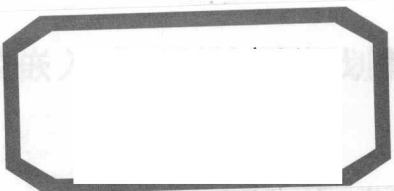
# 嵌入式系统 开发与应用



清华大学出版社

21世纪高等学校

教材



王黎明 刘小虎 闫晓玲 编著

# 嵌入式系统 开发与应用

清华大学出版社  
北京

责任编辑：闫晓玲  
封面设计：李金梅  
责任校对：闫晓玲  
责任印制：闫晓玲

清华大学出版社 发行  
地址：http://www.tup.com.cn, http://www.wqbook.com  
社址：北京清华大学学研大厦A座  
社总机：010-62770175  
邮购部电话：010-62795954, c-service@tup.tsinghua.edu.cn  
质量反馈电话：010-62776969, zhiliang@tup.tsinghua.edu.cn  
网址：http://www.tup.com.cn, http://www.wqbook.com

印 刷：北京  
发 行：北京  
全 国：各  
本 地：各  
版 次：2018年5月第1版  
印 数：1-2000  
定 价：18.00元

057038-01

## 内 容 简 介

本书的立足点是基础化、实用化、可操作性,首先介绍嵌入式系统的概念、引导读者对嵌入式系统有一个整体的印象,从而为读者打开嵌入式系统开发的大门,其次介绍嵌入式开发的集成开发环境;然后以 Stellaris 处理器 LM3S8962 为对象,介绍了系统控制、GPIO、串行通信、定时器模块、ADC 模块、PWM 及模拟比较器;CAN 模块;最后以全国大学生飞思卡尔杯智能汽车比赛为应用案例,介绍其系统设计及实现方法。

本书可作为大专、本科院校自动化、机电、仪器仪表和自动控制等专业的工业控制网络等相关课程的教材或教学参考书,也可供从事工业控制网络系统设计和产品研发的工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

嵌入式系统开发与应用/王黎明等编著. —北京:清华大学出版社,2016

21 世纪高等学校嵌入式系统专业规划教材

ISBN 978-7-302-42913-5

I. ①嵌… II. ①王… III. ①微型计算机—系统开发—高等学校—教材 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2016)第 030756 号

责任编辑:刘向威 薛 阳

封面设计:常雪影

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:24.25

字 数:591 千字

版 次:2016 年 5 月第 1 版

印 次:2016 年 5 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:067038-01

## 出版说明

嵌入式计算机技术是 21 世纪计算机技术两个重要发展方向之一,其应用领域相当广泛,包括工业控制、消费电子、网络通信、科学研究、军事国防、医疗卫生、航空航天等方方面面。我们今天所熟悉的电子产品几乎都可以找到嵌入式系统的影子,它从各个方面影响着我们的生活。

技术的发展和生产力的提高,离不开人才的培养。目前国内外各高等院校、职业学校和培训机构都涉足了嵌入式技术人才的培养工作,高校及其软件学院和专业的培训机构更是嵌入式领域高端人才培养的前沿阵地。国家有关部门针对专业人才需求大增的现状,也着手开发“国家级”嵌入式技术培训项目。2006 年 6 月底,国家信息技术紧缺人才培养工程(NITE)在北京正式启动,首批设定的 10 个紧缺专业中,嵌入式系统设计与软件开发、软件测试等 IT 课程一同名列其中。嵌入式开发因其广泛的应用领域和巨大的人才缺口,其培训也被列入国家商务部门实施服务外包人才培养“千百十工程”,并对符合条件的人才培训项目予以支持。

为了进一步提高国内嵌入式系统课程的教学水平和质量,培养适应社会经济发展需要的、兼具研究能力和工程能力的高质量专业技术人次。在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社与国内多所重点大学共同对我国嵌入式系统软硬件开发人才培养的课程框架和知识体系,以及实践教学内容进行了深入的研究,并在该基础上形成了“嵌入式系统教学现状分析及核心课程体系研究”、“微型计算机原理与应用技术课程群的研究”、“嵌入式 Linux 课程群建设报告”等多项课程体系的研究报告。

本系列教材是在课程体系的研究基础上总结、完善而成,力求充分体现科学性、先进性、工程性,突出专业核心课程的教材,兼顾具有专业教学特点的相关基础课程教材,探索具有发展潜力的选修课程教材,满足高校多层次教学的需要。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

(1) 反映嵌入式系统学科的发展和专业教育的改革,适应社会对嵌入式人才的培养需求,教材内容坚持基本理论的扎实和清晰,反映基本理论和原理的综合应用,在其基础上强调工程实践环节,并及时反映教学体系的调整和教学内容的更新。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点。规划教材建设把重点放在专业核心(基础)课程的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型的专业教学内容和课程体系改革成果的教材。

(4) 支持一纲多本,合理配套。专业核心课和相关基础课的教材要配套,同一门课程可以有多个具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教

学参考书,文字教材与软件教材的关系,实现教材系列资源的配套。

(5) 依靠专家,择优落实。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的、以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

### 21 世纪高等学校嵌入式系统专业规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

目前微控制器的性能越来越好,集成的模块也越来越多,内部自带的寄存器越来越多,整体的框架也越来越复杂,因此库开发成为微控制器开发的主流。通过库开发,可以在不了解微控制器底层寄存器配置的情况下,快速掌握单片机的各个模块资源。本书的目的就是带领读者深入了解开发 Stellaris 处理器 LM3S8962 的过程,最终达到灵活快速地上手其他库,直至开发其他 ARM 芯片的水平。

## 本书特点

- (1) 起点低,案例分析透彻。
- (2) 既注重基础理论,更面向应用。
- (3) 紧跟技术发展,选取典型技术开发实例。
- (4) 突破常规,用于创新。

## 本书的组织结构

第 1 章为基础部分。首先介绍嵌入式系统的概念和组成,然后介绍嵌入式 C 语言,最后介绍典型 Cortex-M3 系列微控制器 Stellaris 处理器,以 LM3S8962 处理器为例列举了基本电路设计,包括滤波电路、复位电路、晶振时钟、以太网接口、RS232、RS485、CAN 总线、电源模块、ADC 采样、LED 及按键、蜂鸣器、I<sup>2</sup>C 扩展开关量输入、I<sup>2</sup>C 扩展开关量输出和继电器输出等电路设计。

第 2 章介绍嵌入式开发的集成开发环境。首先了解嵌入式系统开发一般过程,包括目标平台创建、编译、链接和定址过程;然后介绍了 Stellaris 微控制器的 RVMDK 集成开发环境的使用、测试示例、调试器和指令模拟器;最后介绍了本书主要使用的开发工具链 IAR,主要包括 IAR EWARM 的安装、驱动库的安装、WARM 中新建项目和编译运行程序。本章为后面章节程序的编译和测试提供了必要的基础、环境及工具的支持。

第 3 章主要介绍系统控制模块,包括电源结构与 LDO 控制、时钟控制、复位控制、外设控制、睡眠与深度睡眠,杂项功能和中断操作等。通过本章的学习,可以了解 LM3S8962 处理器的电源结构及特点,如何设置 LDO 的输出,另外还可以学习 LM3S8962 处理器的时钟设置的基本要求及方法。本章的学习是接下来几个章节的基本,也是 LM3S8962 处理器系统初始化关键设置的主要内容。

第 4 章主要介绍 LM3S8962 通用输入输出模块 GPIO,包括 GPIO 的两种应用电路、特殊引脚及寄存器和 GPIO 库函数,然后给出了两只 LED 交替闪烁控制实例和 KEY 控制 LED 实例。通过本章的学习,读者能够完成 GPIO 的基本设置和基本输入输出操作。

第 5 章主要介绍 LM3S8962 串行通信模块,包括 UART、I<sup>2</sup>C 和 SSI。首先介绍 UART 异步串行接口的功能、主要的应用库函数以及 I<sup>2</sup>C 协议的基础和功能。其次分析了 I<sup>2</sup>C 总线的主要驱动库函数,介绍常用的同步串行通信的相关内容,包括 TI 的同步串行帧格式、Freescale SPI 帧格式和 MICROWIRE 帧格式。然后介绍了 SSI 通信的位速率帧格式、FIFO 操作和 SSI 中断以及 SSI 通信的主要库函数,最后给出了应用案例。

第6章主要介绍 LM3S8962 的时钟模块,包括通用定时器模块、系统节拍定时器和看门狗定时器。通用定时器模块主要讲解通用定时器总体特性、功能概述和通用定时器库函数,然后按照通用定时器的不同功能给出了不同的案例。系统节拍模块分析了系统节拍的功能简介、系统节拍的基本操作,然后进行了实例分析。

第7章主要介绍 LM3S8962 模拟量的采集变换。首先介绍 Stellaris 微控制器内核集成的 ADC 的特性、功能。其次介绍了 ADC 使用过程中注意的事项,接下来列出了 ADC 的常用库函数。然后实现了几个 ADC 采样的例子,包括 ADC 单通道触发采样、多通道采样、内部温度传感器采样、定时器触发采样和外部触发采样等。最后重点分析了 ADC 的过采样及实现技术。

第8章主要介绍 LM3S8962 的 PWM 模块和 COMP 模块,包括 PWM 总体特性、PWM 功能概述和 PWM 库函数,并且给出了产生两路 PWM 信号的实例。COMP 模块包括电压比较器概述、电压比较器功能及 COMP 库函数,最后给出了内部参考源输出驱动 LED 灯实例及外部参考源输出触发中断实例。

第9章主要介绍 CAN 总线接口模块的应用,包括 CAN 总线的分层结构及通信协议、CAN 总线接口电路设计、CAN 总线收发器设计以及外设驱动函数的分析。给出了 CAN 模块的应用流程及封装函数。最后介绍在 PC 上 CAN 通信工具使用方法。

第10章是一个综合应用案例分析,以第十届飞思卡尔杯全国大学生智能汽车竞赛为背景,分别介绍了智能汽车的设计概况、智能汽车竞赛的比赛规则和智能汽车的比赛概况。然后分硬件和软件两个方面介绍智能汽车的设计要点。最后介绍电子设计调试基本知识、智能车设计调试及注意事项。

### 说明

本书花费了部分篇幅分析一些案例程序代码,分析代码是一个很令人头痛的问题,要想把程序的来龙去脉讲清楚,就不得不贴上一些源码上来。怎么贴?贴多少?太长的源码印在纸上,即使有详细的注释,也实在不容易前后照顾地看清楚。因此本书很多地方仅仅列出了一些关键性的代码,帮助读者分析,次要的部分用省略号带过。同时书中的全部代码都可以在配套资料中找到。

书中经常会省略很多对基础知识和原理的讨论,而请读者参考相关的文档。因为嵌入式技术涉及的范围较广,读者的确需要相当多的参考资料,而不能仅仅指望一本书。本书的目的在于结合作者实际工作中的一些经验,给读者一个思路和解决问题的办法,让读者能够举一反三。

在科技术语方面,书中尽量采用中英文结合的方式。但是有些术语实在找不到对应的确切中文,则直接使用英文。

### 读者对象

本书是一本介绍 Cortex-M3 内核 Stellaris 处理器开发与应用的书籍,目标读者包括一线程序员、嵌入式产品设计师、片上系统(SoC)工程师、嵌入式系统发烧友、学院研究员以及所有涉猎过单片机和微处理器领域并慧眼识珍看中了 Cortex-M3 的人们。同时本书还适合下列人员阅读:

- 想学习或刚刚进入 Cortex-M3 内核 Stellaris 处理器的开发人员;
- 想学习嵌入式技术的开发人员;

- 对嵌入式技术开发感兴趣的人员；
- 使用 Cortex-M3 进行快速开发产品的开发人员。

尽管本书面向初级 Cortex-M3 内核 Stellaris 处理器的开发人员,但读者需要熟悉相关的硬件知识以及 C 语言,保证至少能读懂书中提到的代码。

### 致谢

在本书的编写过程中,得到了很多人的支持和热心关注,首先在这里对他们表示衷心的感谢。

其次感谢周立功单片机发展有限公司,公司为本书的完成提供了非常完美的测试环境和相应的硬件测试平台,同时本书部分内容及代码由该公司提供。

因为本人水平和编写书稿时间的限制,书中难免有遗漏、错误和不妥之处,恳请广大读者批评指正。联系方式为 icesoar@163.com。

王黎明

2015 年 12 月于武汉



# 目 录

<b>第 1 章 嵌入式系统基础</b> .....	1
1.1 嵌入式系统的概念及组成 .....	1
1.1.1 嵌入式平台的硬件构架.....	2
1.1.2 板级支持包和嵌入式系统.....	3
1.1.3 嵌入式系统上的应用程序.....	3
1.2 嵌入式 C 语言 .....	4
1.2.1 变量与常量.....	4
1.2.2 运算符.....	5
1.2.3 预处理.....	7
1.2.4 位处理.....	9
1.2.5 C 编程基本规则 .....	9
1.3 ARM Cortex-M3 概述 .....	10
1.3.1 ARM—Advanced RISC Machines .....	10
1.3.2 Cortex-M3 简介 .....	11
1.4 典型 Cortex-M3 微控制器简介 .....	18
1.4.1 Cortex-M3 的微处理器 Stellaris 系列 .....	18
1.4.2 Stellaris 系列处理器的分类 .....	19
1.4.3 LM3S8962 引脚功能和硬件电路 .....	20
小结 .....	35
思考题 .....	35
<b>第 2 章 集成开发环境</b> .....	37
2.1 嵌入式软件开发过程.....	37
2.1.1 创建过程 .....	37
2.1.2 编译 .....	37
2.1.3 链接 .....	38
2.1.4 定址 .....	39
2.1.5 调试过程 .....	39
2.2 IAR EWARM 安装 .....	40
2.3 安装驱动库.....	42
2.3.1 下载最新库文件 .....	42
2.3.2 拷贝连接器命令文件 .....	43
2.3.3 拷贝驱动库头文件 .....	43

2.3.4	拷贝底层驱动函数库 .....	45
2.4	EWARM 新建项目 .....	46
2.4.1	建立一个项目文件目录 .....	46
2.4.2	新建工作区 .....	46
2.4.3	生成新项目 .....	46
2.4.4	添加/新建文件 .....	48
2.4.5	项目选项设置 .....	52
2.4.6	通用选项设置 .....	52
2.4.7	C/C++ 编译器选项设置 .....	52
2.4.8	Assembler 选项设置 .....	53
2.4.9	Linker 选项设置 .....	53
2.4.10	Debugger 选项设置 .....	56
2.5	编译和运行应用程序 .....	58
2.5.1	编译连接处理 .....	58
2.5.2	查看 MAP 文件 .....	59
2.5.3	加载应用程序 .....	59
2.5.4	应用程序的相关调试 .....	60
2.5.5	生成 hex 文件 .....	63
	小结 .....	64
	思考题 .....	64
<b>第 3 章</b>	<b>系统控制 .....</b>	<b>65</b>
3.1	电源结构与 LDO 控制 .....	65
3.2	时钟控制 .....	68
3.3	复位控制 .....	76
3.4	外设控制 .....	80
3.5	睡眠与深度睡眠 .....	82
3.6	杂项功能 .....	90
3.7	中断控制 .....	92
3.7.1	中断基本概念 .....	92
3.7.2	Stellaris 中断基本编程方法 .....	93
3.7.3	中断库函数 .....	95
3.7.4	GPIO 中断控制例程 .....	99
	小结 .....	100
	思考题 .....	100
<b>第 4 章</b>	<b>通用输入输出 .....</b>	<b>102</b>
4.1	通用 I/O 口两种应用电路 .....	102
4.2	GPIO 概述 .....	103

4.3	特殊引脚及寄存器 .....	106
4.4	GPIO 库函数 .....	108
4.5	两只 LED 交替闪烁控制实例 .....	116
4.6	KEY 控制 LED 实例 .....	116
	小结 .....	119
	思考题 .....	119
<b>第 5 章</b>	<b>串行通信</b> .....	<b>120</b>
5.1	UART 串口通信 .....	120
5.1.1	UART 异步串口概述 .....	120
5.1.2	UART 总体特性 .....	121
5.1.3	UART 功能概述 .....	125
5.1.4	UART 库函数 .....	128
5.1.5	UART 例程分析 .....	137
5.2	I <sup>2</sup> C 串行通信 .....	146
5.2.1	I <sup>2</sup> C 协议基础 .....	147
5.2.2	I <sup>2</sup> C 功能概述 .....	151
5.2.3	I <sup>2</sup> C 库函数 .....	152
5.2.4	I <sup>2</sup> C 例程分析 .....	160
5.3	SSI 串行通信 .....	166
5.3.1	SSI 总体特性 .....	166
5.3.2	SSI 通信协议 .....	167
5.3.3	SSI 功能概述 .....	173
5.3.4	SSI 库函数参考 .....	174
5.3.5	SSI 驱动例程分析 .....	179
	小结 .....	184
	思考题 .....	184
<b>第 6 章</b>	<b>时钟模块</b> .....	<b>186</b>
6.1	系统节拍定时 .....	186
6.1.1	SysTick 功能简介 .....	186
6.1.2	SysTick 基本操作 .....	187
6.1.3	SysTick 中断控制 .....	188
6.1.4	模拟 PC 按键重复特性实例 .....	190
6.2	通用定时器 .....	192
6.2.1	Timer 总体特性 .....	192
6.2.2	Timer 功能概述 .....	193
6.2.3	Timer 库函数 .....	195
6.2.4	定时器 32 位单次触发定时实例 .....	203

6.2.5	定时器 32 位周期定时实例	205
6.2.6	32 位 RTC 定时实例	206
6.2.7	定时器 16 位单次触发定时实例	209
6.2.8	定时器 16 位周期定时实例	210
6.2.9	定时器 16 位输入边沿计数捕获实例	211
6.2.10	定时器 16 位输入边沿定时捕获实例	213
6.2.11	定时器 16 位 PWM 实例	216
6.2.12	Timer PWM 应用蜂鸣器发声实例	217
6.3	看门狗	218
6.3.1	看门狗功能简述	219
6.3.2	看门狗的工作原理	219
6.3.3	外部看门狗	220
6.3.4	WatchDog 功能概述	221
6.3.5	如何正确使用看门狗	222
6.3.6	WatchDog 库函数	222
6.3.7	WatchDog 复位例程	226
6.3.8	WatchDog 作为普通定时器实例	227
	小结	229
	思考题	229
<b>第 7 章</b>	<b>模数转换 ADC</b>	<b>230</b>
7.1	ADC 总体特性	230
7.2	ADC 功能描述	231
7.3	ADC 应用注意事项	235
7.4	ADC 库函数	237
7.4.1	ADC 采样序列操作	237
7.4.2	ADC 处理器触发	241
7.4.3	ADC 过采样	241
7.4.4	ADC 中断控制	243
7.5	ADC 模块的应用	245
7.5.1	ADC 模块初始化	245
7.5.2	ADC 开始采样	246
7.6	ADC 例程分析	246
7.6.1	处理器触发 ADC 采样实例	246
7.6.2	ADC 内置的温度传感器实例	248
7.6.3	处理器触发多通道 ADC 采样实例	251
7.6.4	外部事件触发 ADC 采样实例	253
7.6.5	定时器溢出触发 ADC 采样实例	256
7.6.6	模拟比较器触发 ADC 采样实例	258

7.6.7	差分输入 ADC 采样实例 .....	260
	小结 .....	263
	思考题 .....	263
<b>第 8 章</b>	<b>脉冲宽度调制及模拟比较器 .....</b>	<b>265</b>
8.1	脉冲宽度调制 .....	265
8.1.1	PWM 总体特性 .....	265
8.1.2	PWM 功能概述 .....	266
8.1.3	PWM 库函数 .....	270
8.1.4	产生两路 PWM 信号实例 .....	285
8.1.5	产生两路带死区的 PWM 实例 .....	287
8.1.6	PWM 发生器中断实例 .....	288
8.2	模拟比较器 .....	290
8.2.1	电压比较器 .....	290
8.2.2	COMP 功能 .....	290
8.2.3	COMP 库函数 .....	292
8.2.4	内部参考源输出驱动 LED 例程 .....	297
8.2.5	外部参考源输出触发中断实例 .....	298
	小结 .....	301
	思考题 .....	301
<b>第 9 章</b>	<b>CAN 接口应用 .....</b>	<b>302</b>
9.1	CAN 总线简介 .....	302
9.2	CAN 的分层结构及通信协议 .....	302
9.2.1	CAN 的分层结构 .....	302
9.2.2	CAN 的通信协议 .....	303
9.3	CAN 总线接口应用电路 .....	307
9.3.1	嵌入式处理器上扩展 CAN 总线接口 .....	307
9.3.2	CAN 总线接口应用电路 .....	307
9.3.3	收发器隔离电路设计 .....	308
9.4	CAN 模块特性及驱动库函数 .....	311
9.4.1	数据结构 .....	312
9.4.2	枚举类型 .....	313
9.4.3	接口函数 .....	315
9.5	CAN 模块应用流程 .....	322
9.6	CAN 总线常用函数及例程 .....	324
9.6.1	CAN 总线常用的函数 .....	324
9.6.2	收发数据 .....	325
9.6.3	不同节点通信案例 .....	326

9.6.4	CAN 总线数据简单发送数据实例 .....	328
9.6.5	CAN 发送 ADC 采样等多组数据实例 .....	330
9.6.6	CAN 总线接收数据实例 .....	333
	小结 .....	337
	思考题 .....	337
<b>第 10 章</b>	<b>智能汽车设计实例分析 .....</b>	<b>338</b>
10.1	智能汽车设计概述 .....	338
10.1.1	智能汽车设计的意义及研究内容 .....	338
10.1.2	智能汽车设计的技术关键 .....	339
10.1.3	中国大学生智能汽车设计竞赛简介 .....	340
10.1.4	中国大学生智能汽车设计竞赛的基本规则 .....	342
10.2	智能汽车硬件设计 .....	347
10.2.1	传感器系统 .....	347
10.2.2	电源系统 .....	353
10.3	智能车软件设计 .....	356
10.3.1	软件系统整体框架 .....	356
10.3.2	角度及角速度测量 .....	357
10.3.3	道路信息提取 .....	361
10.3.4	路径识别算法 .....	362
10.3.5	控制策略及控制算法 .....	367
10.4	智能车系统调试 .....	369
10.4.1	电子设计调试基础知识 .....	369
10.4.2	智能车设计的调试及注意事项 .....	370
	小结 .....	371
	思考题 .....	372
<b>参考文献</b>	<b>.....</b>	<b>373</b>

# 第 1 章 嵌入式系统基础

嵌入式系统(Embedded System)是执行专用功能并被内部计算机控制的设备或者系统。一般地,我们日常见到的手机、MP3、PDA、数码相机、机顶盒、电视机、空调器、汽车以及几乎所有智能控制设备等,都是典型的嵌入式系统,或者说其内部具有嵌入式系统的应用。通常,这些设备当中的嵌入式系统在整个应用系统中起智能控制与信息处理的作用。

嵌入式应用的例子远不止上面提到的那些,目前嵌入式系统已经广泛地应用于国防军事、消费电子、信息家电、网络通信和工控测量等各个领域。可以说,嵌入式系统及其应用无处不在。本章主要介绍嵌入式系统的概念、嵌入式系统的组成、嵌入式 C 语言、ARM 微处理器和典型 Cortex-M3 系列微控制器。

## 1.1 嵌入式系统的概念及组成

关于嵌入式系统的定义,本书不想去太深入讨论。因为它的定义实在是太广泛了,从字面意义理解起来甚至容易让人糊涂,现在嵌入式系统的概念也有被滥用的嫌疑。在一般的文献中嵌入式系统是这样定义的:嵌入式系统是以应用为中心,以计算机技术为基础,并且软硬件可裁减,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计算机系统。但是这种定义较为古板,大家公认的比较有前途的嵌入式系统应该是:硬件以一个高性能的处理器(通常是 32 位处理器)为基础,软件以一个多任务操作系统为基础的综合平台。这个平台的处理能力是以往单片机所无法比拟的,它涵盖了软件和硬件两个方面,因此可称之为嵌入式系统。

注意,这里的重点是“系统”而不是“嵌入式”。在明确了嵌入式系统基本定义的基础上,可从以下几方面来理解嵌入式系统。

(1) 嵌入式系统是面向用户、面向产品、面向应用的。嵌入式系统是与应用紧密结合的,它具有很强的专用性,必须结合实际系统需求进行合理的裁减利用。嵌入式系统和具体应用有机地结合在一起,它的升级换代也是和具体产品同步进行,因此嵌入式系统产品进入市场后也具有较长的生命周期。

(2) 嵌入式系统是将先进的计算机技术、半导体技术、电子技术和各个行业技术的具体应用相结合的产物。这一点就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。

(3) 嵌入式系统必须根据应用需求对软硬件进行裁减,满足应用系统的功能、可靠性、成本和体积等要求。为了提高执行速度和系统可靠性,嵌入式系统中的软件一般都固化在存储器芯片或微处理器机本身中,而不是存储于磁盘等载体中。

(4) 嵌入式系统本身不具备自主开发能力,即使设计完成以后用户通常也是不能对其

中的程序功能进行修改的,必须有一套相应的开发工具和环境才能进行开发。

实际上,凡是与产品结合在一起的具有嵌入式特点的系统都可以叫嵌入式系统。现在人们讲嵌入式系统时,某种程度上指的是近些年比较热的具有操作系统的嵌入式系统。

一个典型的嵌入式系统应由硬件平台、板级支持包(Board Support Package, BSP)、操作系统及应用程序这几部分组成,如图 1-1 所示,下面分别简要介绍。

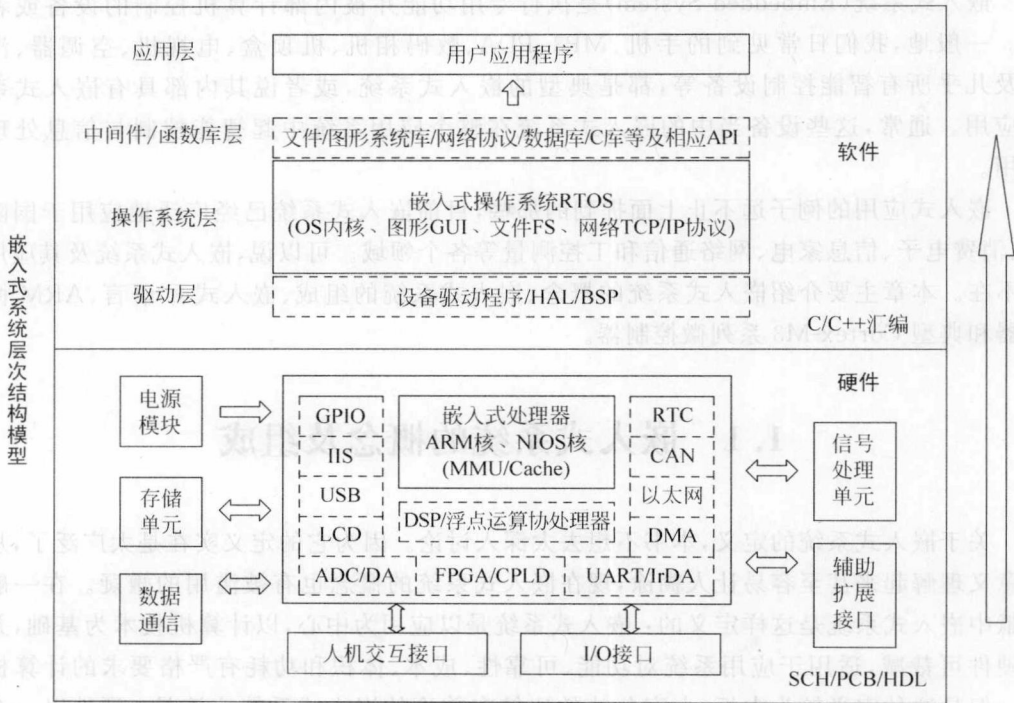


图 1-1 嵌入式系统层次结构模型

### 1.1.1 嵌入式平台的硬件构架

硬件平台主要就是嵌入式处理器及其控制所需要的相关外设。就目前来看,32位处理器是嵌入式处理器的主流,主频常在40MHz以上;当然也有超过400MHz的高端处理器(例如原Intel公司的xscale),甚至超过1GHz的Cortex A8内核处理器。多处理器组成的平台和多内核处理器平台组建也出现在嵌入式领域,如达芬奇技术。不过,现在大量使用的仍然是32位单处理器组成的平台。一个典型的平台的基本组成如图1-2所示,即除了处理器内核以外,一般有时钟、复位、中断控制器、SDRAM控制器和外围总线控制器。

随着片上系统(System on Chip, SoC)技术的发展,越来越多的外设可以集成到片上(处理器芯片内部),其中最重要的是SDRAM控制器。与SRAM控制器相比,SDRAM非常廉价并且存储容量大(通常在8MB以上)。所以有了SDRAM控制器,就可以容易地在系统上扩展大容量的内存空间(8MB甚至64MB以上)。这使得很多复杂的操作系统可以运行在嵌入式处理器上,而且成本低廉。例如有8MB的内存,就足以运行Linux操作系统。



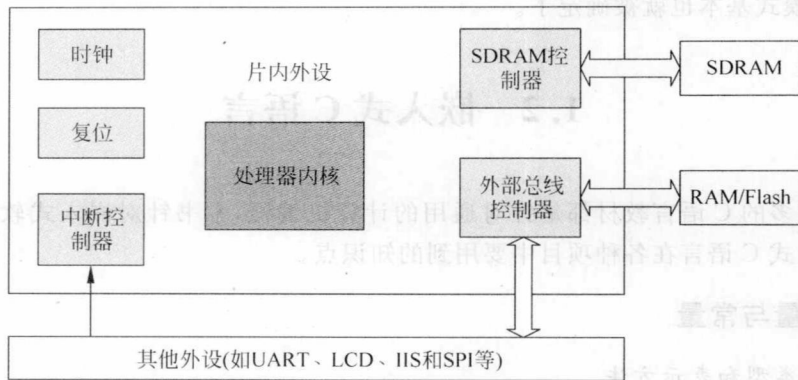


图 1-2 嵌入式平台的基本组成结构

### 1.1.2 板级支持包和嵌入式系统

作为一个成熟的操作系统,要能运行在以各种处理器为核心的硬件平台上,板级支持包就是连接操作系统和硬件平台的桥梁。板级支持包对各种板子的硬件功能提供了统一的软件接口,它包括硬件初始化、中断的产生和处理、硬件时钟和计时器管理、局部和总线内存地址映射以及内存分配等。

虽然板级支持包是介于主板硬件和操作系统之间的一层,但是它属于操作系统的一部分。在开放源码的操作系统中,板级支持包和操作系统之间的层次并不容易划分清楚。可以说,在同一体系结构下移植一个操作系统就是编写板级支持包的过程。

### 1.1.3 嵌入式系统上的应用程序

作为传统意义上的操作系统,运行在系统上的应用程序和操作系统本身是分开的。尤其是存储器管理单元(Memory Management Unit, MMU)出现在处理器上,它可从硬件上把应用程序和操作系统分离开。Linux 和 WinCE 就是这种分离结构的系统,操作系统内核和上层的应用程序可分开编译和管理。这样的系统安全性更高,可维护性更强,更有利于系统各功能模块的划分。通过 MMU 也可在硬件层次上实现动态链接,共享内存或程序的代码空间。但是同时可以看到使用 MMU 把操作系统和应用程序分开是需要代价的,即使 MMU 的转换过程是由硬件专门完成的,这也是要消耗一定的总线资源和系统时钟周期的,结果就是导致处理器的效率降低,成本提高。

很多情况下,尤其是在没有 MMU 的处理器上(例如 ARM7TDMI、ARM9TDMI 和 Cortex-M3),应用程序和操作系统是编译在一起运行的。对于开发者来说,操作系统更像一个函数库,调用此函数库来编写自己的固件(Firmware),与传统意义上的单片机编程没有区别。

因为嵌入式系统应用的一大特点就是面向专一应用,成本低廉。也就是说,不管硬件上运行什么,有无操作系统或应用程序,只要能完成特定的功能即可。对于成型的产品,应用程序很少脱离操作系统而大幅改动。

使用何种方式加载应用程序是与操作系统相关的。选择了一个嵌入式操作系统,应用