

嵌入式应用技术丛书

ARM 嵌入式开发实例

—— 基于STM32的系统设计

肖广兵◎主 编

万茂松 羊 玢◎副主编



- ◆ 全程讲解：ARM学习从零起步
- ◆ 实例教学：工程实例全面剖析
- ◆ 免费资料：代码、教案、工具链

013033581

TP332
296

嵌入式应用技术丛书

ARM 嵌入式开发实例 ——基于 STM32 的系统设计

肖广兵 主 编
万茂松 羊 玢 副主编



北航

C1640339

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

TP332
296

013033281

内 容 简 介

本书以 STM32F103XX 芯片为例,对车载 ARM 嵌入式系统进行了详细的介绍。全书共 8 章,按内容讲解的难度不同划分为 3 篇:基础篇主要介绍有关车载 ARM 嵌入式系统的基础知识,包括 STM32F103XX 芯片的系统资源、工作平台、基本语法指令等内容。提高篇主要介绍 ARM 嵌入式系统在实际工程项目中的初步应用,从 GPIO 接口模块、ADC 模数转换模块、TIMER 定时器等方面进行详细介绍,并着重分析了 ARM 嵌入式系统硬件资源的使用。综合篇主要是对前两篇所有基础知识的总结和应用,着重介绍 ARM 嵌入式系统在车辆中的应用,并给出了相应的程序设计代码。

本书通俗易懂,内容由浅入深,通过示例引导,尤其是结合大量实例进行分析和设计,帮助读者理解和掌握车载 ARM 嵌入式系统的设计方法和编程技巧。在介绍知识点的同时侧重于工程实例的讲解和分析,方便读者自学。既适合 ARM 嵌入式系统方向的本科生、研究生,以及教师作为教学用书,也可以作为广大科研工作者、工程技术人员自学用书和解决工程实际问题的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ARM 嵌入式开发实例:基于 STM32 的系统设计 / 肖广兵主编. —北京:电子工业出版社,2013.4
(嵌入式应用技术丛书)

ISBN 978-7-121-20053-3

I. ①A… II. ①肖… III. ①微处理器—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2013)第 061663 号

策划编辑:陈韦凯

责任编辑:康霞

印刷:三河市鑫金马印装有限公司

装订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本:787×1092 1/16 印张:27.5 字数:704 千字

印次:2013 年 4 月第 1 次印刷

印数:4 000 册 定价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

一、行业背景

ARM 芯片具有体积小、功能强、价格低的特点，在汽车工业、智能仪表、机电一体化、消费多媒体等领域有着广泛的应用，并可以提高生产、消费的自动化和智能化水平。近年来，随着片上处理器的广泛应用，嵌入式系统的开发也变得更加灵活和高效，车载 ARM 系统的开发和应用已经成为嵌入式应用领域的一个重大课题。

二、关于本书

本书以 STM32F103XX 芯片硬件资源环境和 Keil μ Vision for ARM 为依托，介绍了车载 ARM 嵌入式系统的设计方法，包括 ARM 芯片的体系结构、ARM 指令系统、ARM 内部资源、程序设计方法，以及各个功能模块的使用和综合应用系统的开发方法和实例。

本书各章主要内容说明如下：

第 1 章是 ARM 嵌入式基础知识介绍，包括 ARM 的发展、性能特性，开发环境，以及硬件构成等。

第 2 章是 ARM 指令系统的基础使用方法介绍，包括其指令系统简介、指令基本结构、指令类型等。

第 3 章主要是 ARM 嵌入式的内部资源，包括引脚信息、存储器映射、系统控制模块和向量中断模块等。

第 4 章主要介绍 ARM 嵌入式的编程语言，包括汇编语言、数据指令操作、数据运算和流程控制语句等。

第 5 章主要介绍 ARM 嵌入式的功能模块（1），包括 GPIO 接口模块、ADC 模数转换模块、EXTI 中断、TIMER 定时器等。

第 6 章主要介绍 ARM 嵌入式的功能模块（2），包括 RTC 实时时钟、WatchDog 看门狗、USART 串口通信和 CAN 通信模块等。

第 7 章主要介绍 ARM 嵌入式内部资源的 C 编程实例，并以车载温湿度检测仪为例，详细介绍具体的工程设计方法。

第 8 章主要介绍 ARM 嵌入式外部接口的 C 编程实例，并以 TFT 触摸屏汽车故障在线检测诊断为例，详细介绍具体的工程设计方法。

三、本书特色

（1）适合于具有初步 ARM 基础的嵌入式工程师进阶学习，以及高等院校电子类专业的学生和 ARM 嵌入式爱好者阅读。

（2）涵盖了 STM32F103XX 系列单片机从内部资源到用户输入通道、A/D 信号采集、温度/湿度传感芯片、有线通信模块等常用资源或者扩展器件。

（3）基于 Proteus 硬件开发环境提供了相应的仿真运行实例及输出结果。

（4）对于相应的资源或者器件的介绍，都是按照基础知识、硬件电路设计、工程实例分析和应用代码的方式进行的。

（5）提供了大量的工程实例电路和 Keil μ Vision 的工程文件，读者可以直接运行实验。

本书 80% 以上的内容由肖广兵负责编写，万茂松、羊玢参与其他内容的编写并负责全书的审阅、校对工作。同时，参与编写工作的还有吕立亚、孙宁、徐晓美、左付山、余伟、李成龙、韩培、杜康、朱亚飞、廖杰等。在此，对以上人员致以诚挚的谢意。由于时间仓促、程序和图表较多，受学识水平所限，错误之处在所难免，请广大读者给予批评指正。

编者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为，歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

基础篇

第 1 章 初识 ARM 嵌入式系统	(1)
1.1 ARM 嵌入式系统简介	(2)
1.1.1 ARM 嵌入式的发展历程及其应用	(2)
1.1.2 ARM 嵌入式的性能特性	(4)
1.1.3 ARM 嵌入式系统的开发要点	(5)
1.1.4 常用车载 ARM 嵌入式芯片 STM32F103XX	(7)
1.2 ARM 嵌入式系统的开发环境	(8)
1.2.1 Keil MDK 简介	(9)
1.2.2 Keil MDK 的开发步骤	(11)
1.3 ARM 嵌入式系统的硬件构成	(16)
1.3.1 ARM 嵌入式处理器结构	(17)
1.3.2 ARM 嵌入式处理器中的指令	(18)
1.3.3 ARM 硬件配置	(18)
1.4 ARM 嵌入式系统开发实例——车载嵌入式芯片 STM32F103XX	(19)
1.4.1 系统资源与性能参数	(20)
1.4.2 系统硬件设计	(22)
1.4.3 系统外围接口	(23)
第 2 章 ARM 指令系统	(25)
2.1 ARM 指令系统简介	(26)
2.1.1 ARM 汇编语言的基本语法	(27)
2.1.2 ARM 汇编指令中的后缀	(29)
2.1.3 ARM 汇编指令的书写格式	(30)
2.2 ARM 指令集的基本概念	(31)
2.2.1 ARM 指令的基本格式	(31)
2.2.2 ARM 指令中的条件执行	(33)
2.2.3 ARM 指令中的 S 标志	(35)
2.3 ARM 指令集的类型	(36)
2.3.1 跳转指令	(36)
2.3.2 算术运算指令	(42)
2.3.3 逻辑运算指令	(54)
2.3.4 存储器访问指令	(59)
2.3.5 数据传送指令	(61)
2.3.6 协处理器指令	(64)
2.3.7 异常中断产生指令	(68)

2.3.8	ARM 指令小结	(69)
2.4	Thumb 指令集	(69)
2.4.1	Thumb 指令的基本概念	(70)
2.4.2	Thumb 指令集的结构	(70)
2.4.3	Thumb 状态下的寄存器	(71)
2.4.4	Thumb 指令集的类型	(72)
2.4.5	Thumb 指令小结	(72)
第 3 章	STM32 技术基础	(73)
3.1	STM32F103XX 系统简介	(74)
3.1.1	STM32F103XX 系列处理器芯片	(74)
3.1.2	STM32F103XX 器件信息	(76)
3.2	STM32F103XX 引脚信息	(77)
3.3	STM32F103XX 的内部结构	(77)
3.3.1	STM32F103XX 芯片总体结构	(83)
3.3.2	STM32F103XX 片上 Flash 程序存储器	(83)
3.3.3	STM32F103XX 片内静态 RAM	(84)
3.4	STM32F103XX 存储器映射	(84)
3.4.1	存储系统中的大/小端配置	(84)
3.4.2	系统存储器的映射	(85)
3.4.3	系统存储器的访问属性	(86)
3.4.4	系统存储器的地址重映射	(88)
3.4.5	系统存储中止的异常	(89)
3.5	STM32F103XX 的系统控制模块	(89)
3.5.1	晶体振荡器	(90)
3.5.2	外部中断输入	(92)
3.5.3	系统的启动模式	(93)
3.5.4	系统锁相环 PLL	(93)
3.5.5	系统休眠与低功耗	(94)
3.5.6	系统复位	(98)
3.5.7	系统时钟分频	(99)
3.5.8	系统掉电检测与控制	(100)
3.6	STM32F103XX 向量中断控制器	(100)
3.6.1	中断的概念与类型	(101)
3.6.2	外部中断/事件控制器的特点与结构	(102)
3.6.3	EXTI 的寄存器	(104)
3.6.4	中断的处理过程	(107)

提高篇

第 4 章	STM32F103XX 程序设计	(108)
4.1	STM32F103XX 处理器的汇编程序开发框架	(109)

4.1.1	ARM 汇编语言中的代码程序段	(110)
4.1.2	ARM 汇编语言中的数据程序段	(111)
4.1.3	ARM 汇编语言中的宏定义	(111)
4.1.4	ARM 汇编语言中的符号数据	(112)
4.2	STM32F103XX 处理器的数据操作	(114)
4.2.1	C 语言中的数据类型	(114)
4.2.2	C 语言中数据的输入/输出格式	(115)
4.2.3	C 语言中变量和常量	(118)
4.2.4	C 语言中的整型数据	(120)
4.2.5	C 语言中的实型数据	(123)
4.2.6	C 语言中的字符型数据	(126)
4.2.7	C 语言中的字符串常量	(132)
4.2.8	不同类型数据之间的混合运算	(133)
4.3	STM32F103XX 处理器的数据运算	(134)
4.3.1	ARM 程序代码中的算术运算符	(135)
4.3.2	算术运算符的优先级和结合性	(136)
4.3.3	数据类型的强制转换	(137)
4.3.4	自增与自减操作	(138)
4.3.5	赋值运算符和赋值表达式	(140)
4.3.6	逗号运算符	(142)
4.4	STM32F103XX 的流程控制语句	(143)
4.4.1	ARM 中的 if 条件判断语句	(143)
4.4.2	ARM 中的 switch 分支选择语句	(146)
4.4.3	ARM 中的循环控制语句	(150)
第 5 章	STM32F103XX 功能模块 (1)	(155)
5.1	ARM 处理器的选型与功能模块	(156)
5.1.1	ARM 处理器的性能参数	(156)
5.1.2	ARM 处理器的外部接口	(157)
5.1.3	ARM 处理器的芯片封装	(158)
5.1.4	STM32F103XX 系列 ARM 处理器	(160)
5.2	GPIO 接口模块	(162)
5.2.1	GPIO 引脚特性	(162)
5.2.2	GPIO 引脚描述	(162)
5.2.3	GPIO 引脚函数库	(164)
5.2.4	GPIO 的硬件电路	(172)
5.2.5	基础实验一: 汽车安全带报警指示灯与蜂鸣器实验	(174)
5.3	ADC (模数转换) 模块	(179)
5.3.1	ADC 特性	(179)
5.3.2	ADC 引脚的描述	(180)
5.3.3	ADC 库函数	(182)

5.3.4	ADC 硬件电路	(192)
5.3.5	基础实验二：汽车发动机水温测量系统实验	(194)
5.4	EXTI 中断模块	(208)
5.4.1	外部中断/事件的分类	(208)
5.4.2	外部中断/事件的结构	(210)
5.4.3	外部中断/事件的执行顺序与嵌套	(211)
5.4.4	外部中断/事件的选择	(213)
5.4.5	外部中断/事件库函数	(214)
5.4.6	外部中断/事件硬件电路	(221)
5.4.7	基础实验三：汽车紧急制动实验	(223)
5.5	高级控制定时器 TIM1	(230)
5.5.1	TIM1 的结构特性	(231)
5.5.2	TIM1 的功能	(232)
5.5.3	TIM1 的控制寄存器	(239)
5.5.4	TIM1 库函数的功能说明	(244)
5.5.5	基础实验四：PWM 驱动汽车灯光照明实验	(251)
5.6	通用定时器 TIMx	(257)
5.6.1	TIMx 的结构特性	(258)
5.6.2	TIMx 的功能	(259)
5.6.3	TIMx 的控制寄存器	(263)
5.6.4	基础实验五：汽车轮胎压力检测实验	(267)
第 6 章	STM32F103XX 功能模块 (2)	(275)
6.1	实时时钟模块	(276)
6.1.1	RTC 实时时钟的功能特性	(276)
6.1.2	RTC 实时时钟的结构	(277)
6.1.3	RTC 寄存器的操作方式	(278)
6.1.4	RTC 实时时钟的寄存器	(280)
6.1.5	RTC 实时时钟的库函数	(281)
6.1.6	基础实验一：车载时钟与电子日历	(290)
6.2	看门狗 WatchDog 模块	(299)
6.2.1	STM32 系列处理器中的看门狗	(300)
6.2.2	独立看门狗的功能特性	(300)
6.2.3	独立看门狗的寄存器	(301)
6.2.4	独立看门狗的库函数	(302)
6.2.5	窗口看门狗的功能特性	(306)
6.2.6	窗口看门狗的寄存器	(308)
6.2.7	窗口看门狗的库函数	(309)
6.2.8	基础实验二：基于秒中断的新能源汽车电池 SOC 值的检测实验	(313)
6.3	USART 串口通信模块	(318)
6.3.1	USART 的功能特性	(320)

6.3.2	USART 的字符描述	(321)
6.3.3	USART 的发送器	(322)
6.3.4	USART 的接收器	(324)
6.3.5	USART 的中断请求	(327)
6.3.6	USART 的寄存器	(328)
6.3.7	USART 的库函数	(329)
6.3.8	基础实验三：汽车轮速检测实验	(343)
6.4	CAN 通信模块	(349)
6.4.1	CAN 的功能结构	(350)
6.4.2	CAN 的运行模式	(351)
6.4.3	CAN 的功能描述	(353)
6.4.4	CAN 的寄存器	(360)
6.4.5	CAN 的库函数	(362)
6.4.6	基础实验四：基于 CAN 通信的车载电动机冷控实验	(380)

综合篇

第 7 章	STM32F103XX 内部资源的 C 编程实例	(385)
7.1	基于 STM32F103XX 的车载温/湿度检测仪的设计	(386)
7.1.1	项目内容的概述	(386)
7.1.2	项目需求分析	(390)
7.2	系统硬件电路设计	(391)
7.2.1	系统电源设计	(391)
7.2.2	传感电路及运放电路的设计	(395)
7.2.3	DS18B20 数字式温度传感器的电路设计	(397)
7.2.4	SHTXX 数字式湿度传感器的电路设计	(398)
7.2.5	串口通信与 BOOT 启动电路的设计	(400)
7.2.6	人机交互界面	(401)
7.3	系统软件工程的设计	(403)
第 8 章	STM32F103XX 外部接口的 C 编程实例	(406)
8.1	基于 TFT 触摸屏的汽车故障在线检测诊断教学平台设计	(407)
8.1.1	项目内容概述	(407)
8.1.2	项目需求分析	(408)
8.2	系统硬件电路设计	(411)
8.2.1	系统电源设计	(411)
8.2.2	系统逻辑控制电路设计	(414)
8.2.3	TFT LCD 触摸屏电路设计	(416)
8.2.4	串口通信电路与 JTAG 电路	(419)
8.3	系统软件工程设计	(420)

第1章

初识 ARM 嵌入式系统

ARM 嵌入式系统在日常的消费电子领域随处可见，以其优越的性能和完善的开发环境得到了广大电子工程师的青睐。

ARM 微处理器由 ARM 公司提供 IP 授权，交付多个芯片设计厂商进行整合生产。随着 ARM 的发展，其内核版本和支持的生产厂商也越来越多，因此市场上所能够找到的 ARM 芯片也是种类繁多的。

本章主要介绍 ARM 嵌入式系统的基本结构和常见的 ARM 微处理器。

本章重点



- ARM 嵌入式系统的发展历史与特点；
- ARM 嵌入式系统的开发环境；
- ARM 嵌入式系统的硬件构成。

本章难点



- ARM 嵌入式系统的开发流程；
- ARM 嵌入式系统的硬件构成。

1.1 ARM 嵌入式系统简介

近年来, ARM 嵌入式系列产品, 如智能手机、车载 MP4、车载导航仪和平板电脑等, 已经成为大众广泛使用的产品。新型的嵌入式产品不断问世, 各个 IT 公司纷纷投入巨资加紧研发流程, 嵌入式技术人才需求猛增。

随着网络通信技术、计算机技术和微电子技术的迅速发展, 嵌入式系统已经成为当前 IT 行业的焦点。同时在数字信息技术和网络技术高速发展的后 PC 时代, 嵌入式系统因其体积小、可靠性高、功能强和灵活方便等许多优势, 逐步渗透到工业、军事、医疗、汽车及日常生活的各个领域, 在嵌入式相关行业技术改造、产品更新换代、加速自动化进程, 以及提高产品效率等方面起到了极其重要的推动作用。

在汽车行业, 各个零部件由纯机械产品向机电一体化、汽车电子智能化逐步转变。这些车载电子技术的应用改革, 都离不开对 ARM 嵌入式系统的依赖, 如汽车电子油门、电控发动机、车载 IMMO 系统、车载 GPS 导航、全景倒车雷达等。为了使车载 ARM 控制器具有资源管理、快速中断、实时响应等能力, 提供多任务处理, 以及更好地分配系统资源的功能, 用户甚至还需要针对特定的 ARM 嵌入式硬件平台和实际的工程应用进行操作系统的移植。

1.1.1 ARM 嵌入式的发展历程及其应用

ARM 的英文全称是 Advanced RISC Machine, 用户既可以认为 ARM 是一个 IT 行业公司的名称, 也可以认为是一种“嵌入式微处理器核”技术的名称, 甚至还可以认为是具有某种“嵌入式微处理器核”技术的一类芯片, 以及嵌入式系统的总称。对于从事系统开发的嵌入式工程师而言, ARM 通常是指带有 ARM 处理器的嵌入式系统。在本书所介绍的内容中, ARM 是指搭载 ARM 处理器的嵌入式系统。

ARM 公司是微处理器行业的一家知名企业, 1990 年 11 月成立于英国, 是苹果电脑、Acorn 电脑集团和 VLSI Technology 的合资企业。1991 年, ARM 公司推出了 ARM6 处理器家族, VLSI 公司则是第一个将其成功应用在内部研发产品上的生产厂家。后来, 陆续有其他 IT 巨头, 包括 TI 公司、NEC 公司、SHARP 公司, 以及 ST 公司等, 都陆续获得了 ARM 公司的正式授权, 将 ARM 处理器大面积地进行推广, 使得 ARM 处理器在汽车电子、新能源汽车、车载网络、智能手机, 以及其他消费电子中都得到广泛应用。一般情况下, 目前有大约 80% 的电子消费终端都采用了 ARM 处理器, 具体如图 1.1 所示。

当前 ARM 嵌入式芯片的出货量每年都较上一年增长 20 亿片以上。与其他半导体(微处理器)公司不同的是, ARM 公司从来不制造或销售某一个具体型号的 ARM 嵌入式处理器芯片, 而是将 ARM 嵌入式处理器的具体设计授权给相关的商务合作伙伴, 如 Intel 公司、Philips 公司、TI 公司等。

各个授权合作公司根据自身对嵌入式芯片的设计能力、产品领域特点等因素对嵌入式芯片进行功能、资源性的裁剪, 以满足嵌入式市场的需求。基于 ARM 嵌入式系统的低成本和高性能的解决方案, 各个授权合作公司设计出多种多样的处理器芯片、微控制器, 以及嵌入式片上系统(SOC), 即所谓的“ARM 知识产权授权”结构, 如图 1.2 所示。

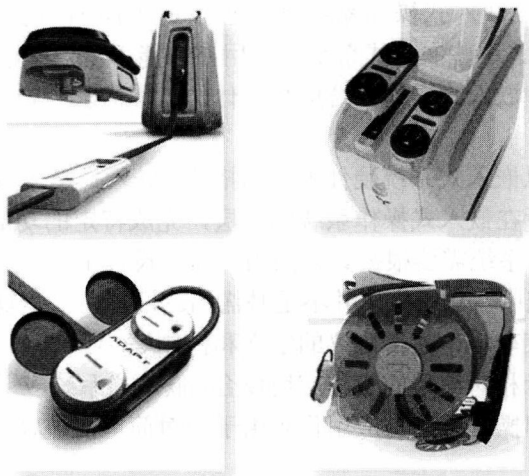


图 1.1 ARM 嵌入式在电子行业中的应用

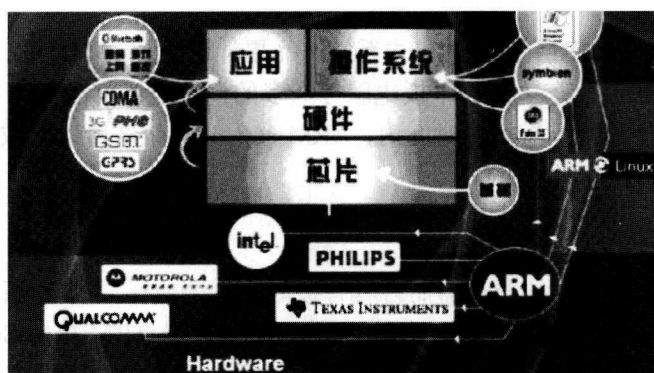


图 1.2 “ARM 知识产权授权”结构图

一般而言，ARM 嵌入式系统由 3 部分组成，即嵌入式系统硬件平台、嵌入式操作系统（代码）和嵌入式系统应用，如图 1.3 所示。

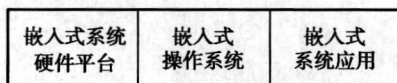


图 1.3 嵌入式系统的组成

1) 嵌入式系统硬件平台

嵌入式系统硬件平台主要是指各种嵌入式处理器和外围设备等，如基于 ARM 核的 STM32XX 处理器、51 系列单片机等。

2) 嵌入式操作系统（代码）

嵌入式操作系统是指在嵌入式硬件平台上运行的代码和操作系统。目前主流的嵌入式操作系统是基于 C 语言（或汇编语言，较少）的嵌入式 Linux、 μ CLinux 和 μ C/OS-II 等。

一般而言，Linux 操作系统具有比较完善的网络接口支持； μ CLinux 操作系统常被用在一些不需要“内存管理单元”（MMU, Memory Management Unit）的嵌入式平台中；而 μ C/OS-II 系统是典型的实时操作系统，可以满足对实时性要求比较高的场合，如汽车电子油门、车载高速 CAN 网络等，具有非常快的响应时间。

除了上述介绍的几种嵌入式操作系统外，在当前主流的消费电子产品中，还广泛使用 Android、Meego 等系统。这些嵌入式系统主要用在智能手机及平板电脑上。在实际的工程应用中，用户具体使用何种嵌入式操作系统应视具体的工程需求而定。

3) 嵌入式系统应用

嵌入式系统应用是以嵌入式系统硬件平台的搭建、嵌入式操作系统的成功移植和运行为前提的。这一部分内容运行在嵌入式操作系统的上层，完成特定的功能目标。

通常情况下，不同的系统需要根据具体的项目需求设计不同的嵌入式应用程序。但是值得注意的是，在嵌入式系统中，系统应用并不是必需的。只有在一些复杂的工程设计中才会需要嵌入式系统应用，如对汽车整车通信协议的定义和控制等。

在一些要求不高或者相对简单的工程应用场合，嵌入式系统应用经常被合并到操作系统及其代码的实现过程中，即操作系统与应用代码由于相对简单而被糅合在一起。

1.1.2 ARM 嵌入式的性能特性

截止 2010 年，内嵌 ARM 核的嵌入式系统产品累计已经超过 200 亿个。在日常生活中用户所接触到的智能手机、汽车电子、平板电脑等众多的消费电子产品中无不存在 ARM 嵌入式处理器的身影。从诞生到现今短短的时间内，ARM 嵌入式系统之所以能取得如此巨大的成就，跟它自身优越的性能是密不可分的。

1. 极低的系统功耗

ARM 嵌入式系统相比其他处理器而言具有极低的系统功耗，这就使得它能广泛地被应用于手持式电子设备的设计场合。目前，ARM 微处理器和内嵌 ARM 核的 SOC 芯片已经在手持终端，如智能手机、车载 MP4、电动汽车的电池管理系统等设备中得到广泛应用。图 1.4 为具有自主研发产权的新能源电动汽车动力锂电池管理系统。



图 1.4 新能源电动汽车动力锂电池管理系统

2. 较短的开发周期

ARM 嵌入式系统的开发周期完全是由 ARM 的商业模式决定的。ARM 公司将成熟的 ARM 技术直接授权给其他合作芯片设计厂商，在很大程度上缩短了 ARM 嵌入式产品的开发周期。而这对于芯片设计厂商而言也是一个巨大的优势。

3. 支持双指令集

绝大部分 ARM 处理器都可以支持 ARM 和 Thumb 两种不同的工作模式,分别用以支持 32 位的 ARM 指令集和 16 位的 Thumb 指令集。对于普通用户而言,这两种指令集都各有所长。

32 位的 ARM 指令集在命令功能上相对更为丰富,性能也更好。在运行代码的过程中,实现同样的功能所需要的指令数(代码量)更少;而 16 位的 Thumb 指令集是 ARM 指令集的一个子集,因此,在实现相同的功能操作时需要较多的指令数(相比 ARM 指令集),但是使用 Thumb 指令集实现的程序代码所占据的程序空间相对较少,具有较高的代码执行效率。除此之外,由于 16 位的 Thumb 指令代码在译码的过程中相对比较简单,因此 Thumb 指令具有更低的系统功耗。

4. 高效的系统总线

在 ARM 嵌入式系统中,处理器没有采用 DSP(数字信号处理器)架构中的多级流水线机制,而是采用了一组专门针对 ARM 内核的片上系统 SOC 开发的总线规范,即 AMBA 总线(Advanced Microcontroller Bus Architecture)。

该总线规范由 ARM 公司设计,独立于 ARM 微处理器的制造工艺技术。在该总线规范中,定义了以下 3 种可供用户组合使用的不同类型的总线。

1) AHB (Advanced High-Performance Bus)

该类型的总线支持多种数据传输方式,以及多个总线主设备之间的数据传输。适用于高性能和高时钟频率的系统模块,如 CPU 处理器、片上存储器、DMA 设备、DSP,以及其他协同处理器等。

2) ASB (Advanced System Bus)

该类型的总线同样也适用于高性能的系统模块。在不需要使用 AHB 的场合,用户也可以选择 ASB 作为系统总线。

3) APB (Advanced Peripheral Bus)

该类型的总线的主要特点是结构简单,低速,极低的功耗。该总线主要适用于低功耗,对实时性要求不高的外部设备,如对汽车门窗锁的控制等。

1.1.3 ARM 嵌入式系统的开发要点

与传统的单片机相比,ARM 嵌入式系统的整体系统性能和数据处理能力有了大幅提升。与之相应的,ARM 嵌入式系统设计的复杂度和难度也有所提升,与传统的单片机设计方法也有着很大不同。

总体而言,ARM 嵌入式系统的开发可以分为“基于 ARM 内核的芯片设计”和“基于 ARM SOC 的开发应用”。本书主要与读者讨论有关“ARM 芯片的开发应用”,不涉及 ARM 芯片的设计。

对于用户而言,在实现对 ARM 嵌入式系统进行开发之前,首先应该对 ARM 嵌入式系统的概念和基本结构做一些了解,然后还需要熟悉 ARM 嵌入式指令集。虽然现在绝大部分嵌入式系统都使用 C 语言开发程序,但是绝大部分芯片的初始化启动程序仍然是使用汇编语言写的,以得到较高的代码执行效率和开机速度。因此,用户在熟练掌握 C 语言的基础上,了解一

定的汇编语言知识也是必要的。除此之外，用户还需要结合所使用的 ARM 处理器芯片，掌握某一个集成开发环境的使用方法，务必做到熟练使用。

1. 明确 ARM 嵌入式系统开发的过程

不同于通用计算机平台上应用软件的开发，在 ARM 嵌入式系统程序的开发过程中具有很多特点和不确定性，其中最重要的一点就是嵌入式软件代码和系统硬件的独立性。

由于嵌入式系统的层次结构和自身的灵活性、多样性，各个层次之间缺乏统一的接口标准，甚至每个嵌入式系统都各不一样。这样就给上层的嵌入式软件设计人员在嵌入式软件代码设计的过程中带来比较大的困难。软件设计人员必须建立在对底层硬件设计充分了解的基础上，才能设计出符合 ARM 嵌入式系统要求的应用层代码。

为了简化开发流程，提高开发效率，用户可以在应用与驱动（API）接口上设计一些相对统一的接口函数，就可以在一定程度上规范应用层嵌入式软件设计标准，同时方便应用程序在跨平台之间的复用和移植。

2. 熟悉开发工具环境里的库函数

对于 ARM 开发工具环境里所提供的库函数，用户需要对其功能、参数、结构、调用函数等有比较清楚的了解，其中最重要的 3 方面如下。

- (1) 考虑硬件对库函数的支持。
- (2) 符合目标系统上的存储器资源分布。
- (3) 应用程序运行环境的初始化。

举例来说，在嵌入式代码设计过程中，有一类动态内存管理函数，如 `malloc()` 等，其本身是独立于目标系统而运行的，但是它所使用的存储器空间需要根据具体的目标来确定，所以 `malloc()` 函数本身并不需要移植或裁剪，这也是一个特例。但那些设置动态内存区（地址和空间）的函数则是和目标系统的存储器分布直接相关的，需要进行移植。

需要补充说明的是，在 ARM 嵌入式系统开发过程中，库函数的使用并不是必需的。用户同样可以自行组织编写程序代码，用于实现与库函数一样的功能。相比自行编写代码而言，系统提供的库函数具有更好的稳定性和可移植性。

在本书后续章节所编写的工程代码中，几乎都采用了 STM32XX 系列 ARM 处理器开发平台（Keil for ARM）所提供的库函数文件。

3. 熟悉 ARM 嵌入式系统的调试操作

嵌入式系统不可避免地会涉及对输入/输出设备的操作，例如，文件操作函数需要访问磁盘 IO，打印机函数需要访问字符输出设备等。在嵌入式调试环境下，所有的标准 C 库函数都是有效且有其默认行为的。一般情况下，部分目标系统硬件所不能支持的操作用户可以通过相应的调试工具来完成。

但是最终嵌入式系统的运行是需要完全脱离调试工具独立运行的，所以在程序移植的过程中，用户需要对这些库函数的运行机制有比较清楚的了解。特别是在系统出现故障甚至逻辑错误的时候，需要用户能够以最短的时间来排查、解决问题。