工业和信息化“十三五”
人才培养规划教材

基于STM32的嵌入式系统应用

STM32 Embedded System Application

孙光 ◎ 主编
肖迎春 曾启明 ◎ 副主编
王静霞 ◎ 主审

以**应用**为中心，强化**编程规范**的学习，注重学生良好**编程习惯**和**编程风格**的培养
实训项目以全国大学生**电子设计竞赛**的赛题“**帆板角度测量与控制装置**”为蓝本
将**STM32微控制器**的编程落实到直观具体的控制对象上，提高学生的**学习兴趣**

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化“十三五”
人才培养规划教材

基于STM32的嵌入式系统应用

STM32 Embedded System Application

孙光 ◎ 主编
肖迎春 曾启明 ◎ 副主编
王静霞 ◎ 主审

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

基于STM32的嵌入式系统应用 / 孙光主编. -- 北京 : 人民邮电出版社, 2019.10
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-51799-9

Ⅰ. ①基… Ⅱ. ①孙… Ⅲ. ①微控制器－高等学校－教材 Ⅳ. ①TP332.3

中国版本图书馆CIP数据核字(2019)第172973号

内 容 提 要

本书介绍了意法半导体公司出品的基于 Arm Cortex-M3 内核的 STM32F103 微控制器在工程实践中的应用。

全书分为基础篇和应用篇。基础篇介绍了嵌入式系统的基本概念、实训使用的软硬件平台、STM32 的标准外设库、嵌入式 C 语言编程的特点、STM32 系统时钟、彩色 LCD 显示基础、字符编码和显示字库等内容；应用篇依托 STM32 微控制器的主要外设、常用外围器件以及典型应用等设计了 11 个实训项目。

本书适合作为高职院校电子与控制类专业“嵌入式系统应用”等相关课程的教材，也可以作为工程技术人员学习 STM32 微控制器编程的快速入门参考书。

◆ 主　　编　孙　光
副 主 编　肖迎春　曾启明
主　　审　王静霞
责任编辑　祝智敏
责任印制　马振武

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市君旺印务有限公司印刷

◆ 开本：787×1092　1/16
印张：12.5　　2019 年 10 月第 1 版
字数：309 千字　　2019 年 10 月河北第 1 次印刷

定价：42.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前言 FOREWORD

意法半导体公司（ST）作为第一家与 Arm 公司合作正式出品 Cortex-M3 内核微控制器的半导体器件公司，于 2007 年推出 STM32F1 系列微控制器芯片。此后十余年，以 STM32 为代表的 Cortex-M 内核微控制器逐渐在全球通用 32 位微控制器市场占据了主导地位，并不断向下侵蚀了 8 位单片机的市场。

为了适应微控制器市场的这一发展趋势，深圳职业技术学院电子信息工程技术专业在 2012 年开设了以 STM32 微控制器为学习对象的“嵌入式系统应用”课程。毫无疑问，微控制器相关课程是高职电子类专业教学计划的重要内容，目前深圳职业技术学院电子信息工程技术专业设置的微控制器相关课程体系如下表所示。

	课程内容（非课程名称）	课程目的
1	基于 51 单片机的嵌入式 C 语言编程	使学生熟悉微控制器的基本结构并具备基本的 C 语言编程能力
2	基于 STM32 的嵌入式系统应用	使学生具备面对复杂控制对象的 C 语言编程能力
3	基于μCOS 的嵌入式实时操作系统应用	使学生具备在实时操作系统环境下的编程能力

在学校实际教学以及与兄弟院校的交流过程中，对于以上课程体系的设置存在一定争议。有观点认为，既然目前 51 单片机的市场空间已经大大萎缩，不如直接将 STM32 作为微控制器教学的入门课程。

针对此观点，我校通过社团选修课的形式进行了积极尝试，发现即使是选拔成绩较好、具有较高学习热情的学生，也很难直接从 STM32 入手进行微控制器内容的学习。

经过调查和分析，我们发现造成学习困难的原因主要有两个：第一，STM32 微控制器本身内部结构比较复杂，学生感到难以理解；第二，基于 STM32 的嵌入式编程涉及的 C 语言知识点过多，学生在有限的课时内难以全面掌握。

另外，经过市场调研，我们发现虽然 51 单片机作为通用处理器的市场空间已大为压缩，但是在很多追求性价比的专用控制器上，51 单片机仍旧占据着非常大的市场份额。

基于以上事实，我们认为 51 单片机的教学在高职电子类专业整个微控制器课程体系中仍是不可或缺的。在一个简单结构上先进行微控制器的学习，并以此来掌握 C 语言编程的基础知识是比较容易让学生接受的。

由此我们明确了 STM32 课程在整个微控制器课程教学体系中的定位，就是在“基于 51 单片机 C 语言编程”课程的基础上，使学生具备面对复杂对象的嵌入式 C 语言编程能力。

结合高职电子类专业学生的实际情况，我们针对本书的内容设计确定了以下原则。

1. 以应用为中心，有所为有所不为。不纠结于 STM32 微控制器的原理细节，不在书中简单复制和堆砌 STM32 微控制器的知识点，而是注重学生的快速入门和实际编程能力的培养。

2. 在具备基本 C 语言编程能力的前提下，着重加强在基于 STM32 的嵌入式 C 语言编程中常用的宏指令、结构体、指针等内容的学习。

3. 强化编程规范的学习，注重学生良好编程习惯和编程风格的养成。

4. 教学项目的内容编排落实在一个具体的“帆板角度测量与控制装置”上，此装置以全国大学生电子设计竞赛的赛题为蓝本，将STM32目标板作为整个装置的控制核心，尽可能将STM32微控制器的编程落实到直观具体的控制对象上，以提高学生的学习兴趣，明确课程的学习目的。

5. 考虑到一部分不方便使用“帆板角度测量与控制装置”进行实践的读者，本书所有教学项目都设计为可以独立在 STM32 目标板上完成。

6. 教学项目的内容编排除了关注 STM32 微控制器外设的编程，还根据实际应用的需求，加入了彩色 LCD 显示、Wi-Fi 串口模块、物联网云平台的使用等内容。

7. 在大部分教学项目结束时，都给出了相应拓展项目的要求和提示。设置拓展项目的目的一方面是为了巩固学习效果，另一方面是在实施过程中找出问题，为后续项目的学习做铺垫。

本书分为基础篇和应用篇两大部分，教材内容按照基础篇的专题介绍和应用篇的实训项目展开，但并不意味着实际教学需要严格按照教材目录的编排顺序。在课时有限的情况下，可以直接依托实训项目展开教学，每个项目的内容已经足以支撑教学的进行（项目中的一些知识点是对前面专题内容的延伸，如实训项目 1 中关于 STM32 标准外设库的精简结构介绍）。而基础篇的专题可以有选择地穿插在实训项目的教学中，下表列出了推荐的教学顺序（共计 64 课时）。

教材编排顺序		推荐学习顺序	
章节	内容	章节	内容
第 1 章	专题 1——嵌入式系统概述	第 1 章	专题 1——嵌入式系统概述
第 2 章	专题 2——实训项目使用的软硬件平台	第 2 章	专题 2——实训项目使用的软硬件平台
第 3 章	专题 3——CMSIS 与 STM32 标准外设库	第 8 章	实训项目 1——LED 闪烁
第 4 章	专题 4——STM32 嵌入式 C 语言编程的特点	第 9 章	实训项目 2——按键控制 LED 亮灭
第 5 章	专题 5——STM32F10x 微控制器的系统时钟	第 3 章	专题 3——CMSIS 与 STM32 标准外设库
第 6 章	专题 6——彩色 LCD 显示	第 4 章	专题 4——STM32 嵌入式 C 语言编程的特点
第 7 章	专题 7——字符编码与显示字库	第 10 章	实训项目 3——按键控制 LED 闪烁频率（外部中断）
第 8 章	实训项目 1——LED 闪烁	第 6 章	专题 6——彩色 LCD 显示
第 9 章	实训项目 2——按键控制 LED 亮灭	第 7 章	专题 7——字符编码与显示字库
第 10 章	实训项目 3——按键控制 LED 闪烁频率（外部中断）	第 11 章	实训项目 4——彩色 LCD 显示图片与文字
第 11 章	实训项目 4——彩色 LCD 显示图片与文字	第 5 章	专题 5——STM32F10x 微控制器的系统时钟
第 12 章	实训项目 5——按键控制 LED 闪烁频率（定时器中断）	第 12 章	实训项目 5——按键控制 LED 闪烁频率（定时器中断）
第 13 章	实训项目 6——风扇转速的 PWM 控制	第 13 章	实训项目 6——风扇转速的 PWM 控制
第 14 章	实训项目 7——帆板角度与芯片温度检测	第 14 章	实训项目 7——帆板角度与芯片温度检测
第 15 章	实训项目 8——帆板角度与芯片温度检测（DMA 方式）	第 15 章	实训项目 8——帆板角度与芯片温度检测（DMA 方式）
第 16 章	实训项目 9——串行通信控制风扇转速并获取帆板角度	第 16 章	实训项目 9——串行通信控制风扇转速并获取帆板角度
第 17 章	实训项目 10——Wi-Fi 控制风扇转速并获取帆板角度	第 17 章	实训项目 10——Wi-Fi 控制风扇转速并获取帆板角度
第 18 章	实训项目 11——基于 STM32 的物联网云平台温度检测	第 18 章	实训项目 11——基于 STM32 的物联网云平台温度检测

关于本书教学对象的选择有两个问题需要做出说明。

1. 为什么选择 STM32F103 系列芯片?

自从 2007 年 ST 公司推出第一款 Cortex-M3 内核的 STM32F1xx 微控制器至今已逾十年，虽然其后 ST 公司也陆续推出了基于 Cortex-M4 内核的 STM32F4xx 微控制器和基于 Cortex-M7 内核的 STM32F7xx 微控制器，但是从性价比的角度来看，STM32F1xx 微控制器仍然是市场的绝对主流，目前仍不断有新型号推出。在可预见的相当长一段时间内，只要对终端设备的运算能力需求没有本质性的提高，其主流地位仍将持续。

2. 为什么是标准外设库而非 HAL 库?

使用外设驱动函数库进行编程是 STM32 微控制器编程非常重要的特色，外设驱动函数库也从标准外设库发展到了更为抽象化并融入了面向对象思维的 HAL 库。但是从企业产品研发的惯性来看，占据市场主流地位的 STM32F1xx 和 STM32F4xx 微控制器仍然普遍使用标准外设库进行编程，这也是本书选用标准外设库而非 HAL 库进行教学的主要原因。

本书的适用对象为具备基本单片机和 C 语言编程知识的高职院校电子与控制类专业学生。

本书由孙光担任主编，肖迎春、曾启明担任副主编，王静霞担任主审，曾日扬、李坚等为本书实训项目的完善做出了重要贡献，在此一并感谢。

目 录 CONTENTS

Part 01 基础篇

Part 02 应用篇

Part 01

基础篇

Chapter 1

第1章 专题1——嵌入式系统概述

1. 了解嵌入式系统的基本概念
2. 了解 Arm Cortex-M 内核处理器的特点
3. 了解 STM32 处理器的特点

1.1 从单片机到嵌入式系统

单片机是在一颗芯片中集成了中央处理单元（CPU）、数据存储器（RAM）、程序存储器（ROM）、输入/输出端口（I/O）、定时器/计数器（Timer/Counter）等外设的微型电子计算机。

相对功能比较强大的个人计算机（PC），单片机的运算能力是有限的，但是单片机凭借体积小、功耗低、价格便宜等特点，已经渗透到日常生产生活的方方面面，在工业控制、国防技术、家用电器、消费电子产品等领域得到了广泛应用。

随着电子产品人机交互界面彩屏化、触摸化以及通信网络化的趋势，运算能力有限的 8 位乃至 16 位单片机已经越来越不能满足需求，具有较强运算能力的 32 位嵌入式系统的优势日益突显。

嵌入式系统（Embedded System）是一个宽泛的概念，很难下一个确切的定义。图 1-1 描述了嵌入式系统、单片机系统和通用计算机系统之间的关系。

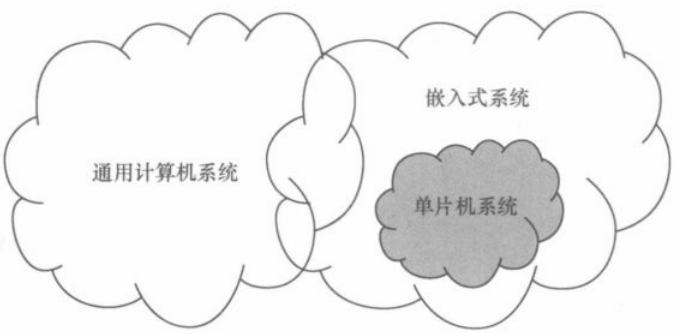

图1-1 嵌入式系统、单片机系统与通用计算机系统

首先，嵌入式系统包括单片机系统，但是其运算能力远远超过传统的单片机。

其次，嵌入式系统不等于通用计算机系统，它是以应用为中心，以计算机技术为基础，软硬

件可裁剪，适用于对功能、可靠性、成本、体积、功耗有严格要求的专用计算机应用系统。

不过随着嵌入式技术的发展，嵌入式系统与通用计算机系统之间也出现了一定程度融合的趋势。

1.2 精简指令集计算机与复杂指令集计算机

学习嵌入式系统，必须要提到两组与电子计算机架构和体系结构相关的概念，一组是精简指令集计算机（Reduced Instruction Set Computer，RISC）和复杂指令集计算机（Complex Instruction Set Computer，CISC），另一组是普林斯顿结构和哈佛结构。我们首先来了解一下 RISC 和 CISC。

RISC 和 CISC 是当前微处理器使用的两种基本架构，它们的区别在于使用了不同的 CPU 设计理念和方法。

顾名思义，采用复杂指令集的处理器结构比较复杂，指令系统比较丰富，有专用指令来完成特定的运算，因此处理复杂运算的效率较高。但由于 CISC 处理器的结构复杂，带来的副作用是价格和功耗较高。典型的 CISC 处理器包括英特尔的 X86 架构处理器以及 8051 单片机。

在 CISC 处理器中，各种指令的使用率相差悬殊，一个典型程序运算所使用的 80%的指令往往只占到整个处理器指令系统的 20%，在实际运行中使用最频繁的是存取指令和加法指令等简单指令。CISC 处理器的复杂结构只是为了少数使用率不高的复杂运算指令服务，这显然是不划算的，于是 RISC 处理器应运而生。

同样从字面理解，采用精简指令集的处理器结构比较简单，对应的指令较少，容易实现单周期指令，适合处理简单的数学和逻辑运算。RISC 把主要精力放在那些经常使用的指令上，尽量使它们简单高效。对不常用的复杂运算，通常采用指令组合来完成，因此在 RISC 处理器中进行复杂运算时效率可能较低，但可以利用流水线技术加以弥补。

从硬件角度来看，CISC 处理的是不等长指令集，它必须对不等长指令进行分割，在执行单一指令的时候需要进行较多的处理工作；而 RISC 执行的是等长指令集，CPU 在执行指令的时候速度较快且性能稳定。在并行处理方面，RISC 明显优于 CISC，RISC 可同时执行多条指令，将一条指令分割成若干个进程或线程，交由多个处理器同时执行。

由于 RISC 执行的是精简指令集，对应的处理器硬件结构相对而言复杂度不高，所以它的制造工艺简单且成本相对低廉。

目前 CISC 与 RISC 正在逐步走向融合，Pentium Pro 芯片就是一个最典型的例子，它的内核基于 RISC 架构，处理器在运行 CISC 指令时将其分解成 RISC 指令，以便在同一时间内能够执行多条指令。

1.3 普林斯顿结构和哈佛结构

普林斯顿结构和哈佛结构是根据计算机的运算内核与程序存储器和数据存储器的不同连接方式而加以区分的两种计算机体系结构。

普林斯顿结构也称冯·诺伊曼结构，如图 1-2 所示，它是一种将程序存储器和数据存储器合并在一起的计算机结构，也就是说，程序存储器和数据存储器共用一条地址总线和数据总线。由于程序指令存储地址和数据存储地址指向同一个存储空间的不同物理位置，因此程序指令和数据的宽度相同，例如英特尔 8086 处理器的程序指令和数据都是 16 位。

这种程序指令和数据共享同一总线的结构，使得信息流的传输成为限制计算机性能提升的瓶颈，也影响了数据处理速度的提高。

使用普林斯顿结构的处理器包括英特尔公司的 8086、Arm 公司的 Arm7、MIPS 公司的 MIPS 处理器等。

哈佛结构是一种将程序指令存储和数据存储分开的计算机结构，如图 1-3 所示，也就是说，程序存储器和数据存储器使用各自独立的地址总线和数据总线。运算内核首先到程序存储器中读取程序指令内容，解码后得到数据地址，再到相应的数据存储器中读取数据，并进行下一步的操作（通常是执行）。程序指令存储和数据存储分开，使程序指令和数据的数据宽度不同，如 Microchip 公司的 PIC16 芯片的程序指令是 14 位，而数据是 8 位。

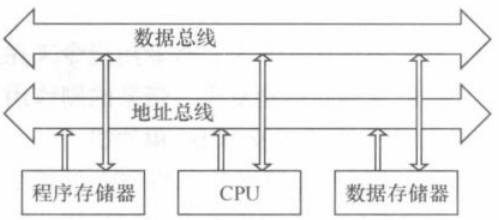

图1-2 普林斯顿结构简化示意图

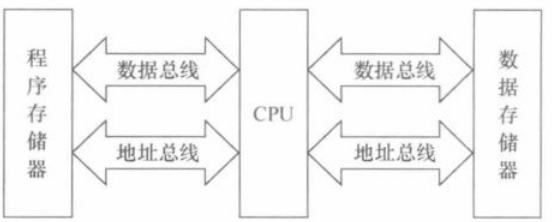

图1-3 哈佛结构示意图

哈佛结构的处理器通常具有较高的执行效率，其独立的程序总线和数据总线使处理器很容易实现流水线操作，也就是在执行当前指令时可以预先读取下一条指令。使用哈佛结构的处理器有很多，例如 Microchip 公司的 PIC 单片机、ATMEL 公司的 AVR 单片机和 Arm 公司的 Arm9 内核之后的处理器等，8051 单片机也属于哈佛结构。

目前的高性能处理器的发展趋势是在芯片内部使用结构复杂、效率较高的哈佛结构，在芯片外部使用结构简单的普林斯顿结构。

本书使用的 STM32 微控制器使用了 Arm 公司的 Cortex-M3 内核，是一款采用哈佛结构的 RISC 处理器。

1.4 Arm 公司及其微处理器

目前，采用 Arm 公司内核的嵌入式系统占据了全球市场的绝对主导地位，我们先介绍一下 Arm 公司的发展历程。

1978 年 12 月，剑桥处理器公司（Cambridge Processing Unit）在英国剑桥创办，主要业务是为当地市场供应电子设备，1979 年合并其他公司后改名为 Acorn。

当时个人计算机刚刚兴起，英国广播公司（BBC）的一部名为《强大的微处理》（The Mighty Micro）的纪录片向电视观众介绍了计算机时代的到来。纪录片播出后引起巨大反响，BBC 决定向受到鼓舞的爱好者出售一款价格合适的国产计算机“BBC Micro”，Acorn 公司赢得了生产“BBC Micro”的合同。产品推向市场后取得了意想不到的成功，预计销售 1.2 万台，结果却卖出了 150 万台，Acorn 初战告捷。

1982 年，Acorn 公司打算使用摩托罗拉公司的 16 位处理器芯片，但是发现这种芯片太慢也太贵，转而向 Intel 公司索要 80286 处理器芯片的设计资料却遭到拒绝，于是被迫开始自行研发。

1985 年，Acorn 设计了自己的第一代 32 位、主频为 6MHz 的 RISC 处理器，简称 ARM（Acorn

RISC Machine)。

1990 年 11 月，Acorn 公司正式改组为 Arm 计算机公司，苹果公司出资 150 万英镑、芯片厂商 VLSI 出资 25 万英镑、Acorn 则以 150 万英镑的知识产权和 12 名工程师入股。苹果公司使用 Arm610 芯片作为 Apple Newton PDA 的 CPU。

随着市场环境的变化，Arm 公司由一家微处理器芯片供应商转型为一家微处理器知识产权供应商。图 1-4 所示为 Arm 公司目前的商业运行模式，它并不出产具体的微处理器芯片，而是将微处理器运算内核的知识产权（Intellectual Property，IP）授权给合作伙伴（Partner），由合作伙伴加上各自的外设后制造成具体的芯片提供给下游客户。

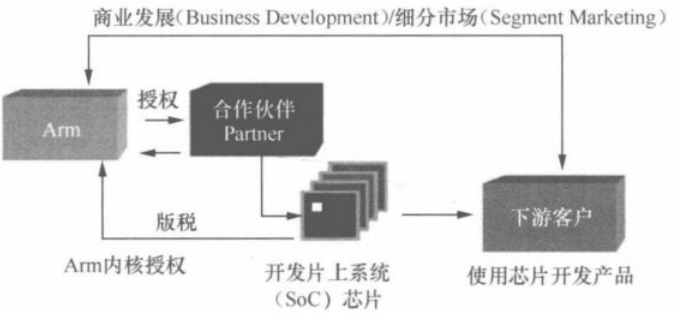

图1-4 Arm公司的商业运行模式

受公司自身财务状况影响，苹果公司后来逐步出售了其持有的 Arm 公司股份。

2010 年 6 月，苹果公司向 Arm 董事会表示有意以 85 亿美元的价格收购 Arm 公司，但遭到 Arm 董事会的拒绝。

2016 年 7 月，曾经投资阿里巴巴的韩裔日本商人孙正义创立的软银集团以 243 亿英镑收购了 Arm 公司。

1.5 Arm Cortex 系列处理器

Arm 处理器发展到现在，其内核架构已从 Arm v1 发展到 Arm v8。图 1-5 所示为具有代表性的 Arm v4 到 Arm v7 架构的进化图，其对应的内核命名也从 Arm7、Arm9 到了 Arm11。Arm 处理器内核进化到 Arm v7 架构后，已经不再沿用过去的数字命名方式，而是冠以 Cortex 的代号。

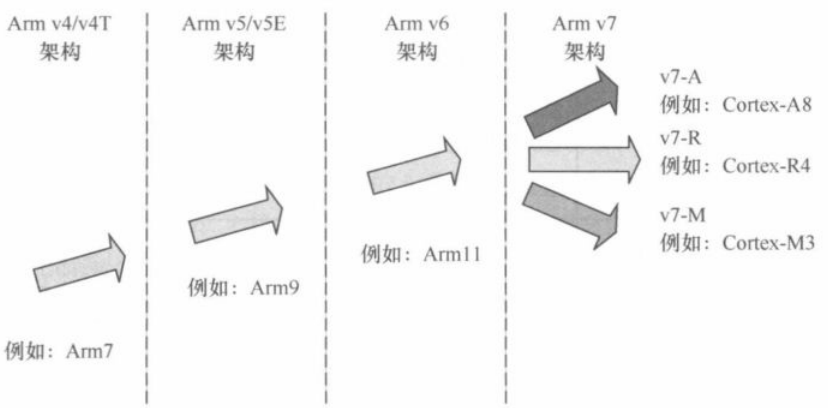

图1-5 Arm内核架构的进化

虽然 Arm7 仍然是一款普林斯顿结构的处理器，但是从 Arm9 开始，Arm 处理器已经成为典型的采用哈佛结构的 RISC 处理器。

Arm Cortex 系列处理器是基于 Arm v7 架构的，分为 Cortex-A、Cortex-R 和 Cortex-M 三个系列。基于 v7A 架构的处理器称为 Cortex-A 系列，基于 v7R 架构的处理器称为 Cortex-R 系列，基于 v7M 架构的处理器称为 Cortex-M 系列。

Cortex-A 系列处理器主要用于高性能场合，是针对日益增长的运行 Linux、Windows、Android 操作系统的移动互联设备和消费电子产品需求设计的。

Cortex-R 系列处理器主要用于实时性（Real Time）要求比较高的场合，针对的是需要运行实时操作系统来控制应用的系统，包括汽车电子、网络和影像系统。

Cortex-M 系列处理器则主要用于微控制器（Microcontroller，MCU）领域，是为那些对功耗和成本非常敏感，同时对性能要求不断增加的嵌入式应用（如汽车电子、家用电器、工业控制、医疗器械、玩具和无线网络等）设计的。

表 1-1 所示为 Cortex 内核推出的时间，其中 Cortex-M 系列已由 2004 年推出的 M3 内核发展到 2016 年推出的 M33 内核；Cortex-R 系列由 2011 年推出的 R4 内核发展到 2016 年推出的 R52 内核；Cortex-A 系列又分成了 32 位和 64 位两个子系列，由 2005 年推出的 A8 内核发展到 2017 年推出的 A55/A75 内核。

表 1-1 Cortex 内核推出的时间

时间/年	Cortex-M 系列	Cortex-R 系列	Cortex-A 系列	
			32 位	64 位
2004	Cortex-M3			
2005			Cortex-A8	
2006				
2007	Cortex-M1		Cortex-A9	
2008				
2009	Cortex-M0		Cortex-A5	
2010	Cortex-M4		Cortex-A15	
2011		Cortex-R4 Cortex-R5 Cortex-R7	Cortex-A7	
2012	Cortex-M0+			Cortex-A53 Cortex-A57
2013			Cortex-A12	
2014	Cortex-M7		Cortex-A17	

续表

时间/年	Cortex-M 系列	Cortex-R 系列	Cortex-A 系列	
			32 位	64 位
2015				Cortex-A35 Cortex-A72
2016	Cortex-M23 Cortex-M33	Cortex-R8 Cortex-R52	Cortex-A32	Cortex-A73
2017				Cortex-A55 Cortex-A75

本书主要关注用于微控制器领域的 Cortex-M 内核，图 1-6 所示是按照运算能力强弱排列的 Cortex-M 系列内核发展进化图，注意内核的序号并不代表内核推出的时间顺序。

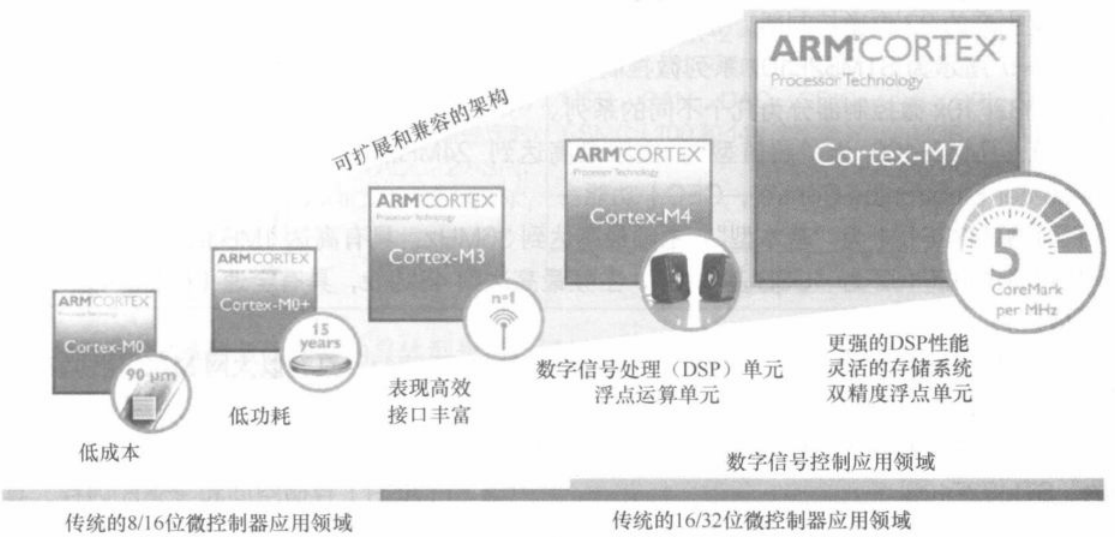

图1-6 Cortex-M内核的进化

Cortex-M3 内核于 2004 年 10 月发布，是面向普通嵌入式市场的高性能、低成本的 Arm 处理器；

Cortex-M1 内核于 2007 年 3 月发布，是专门面向现场可编程门阵列（Field Programmable Gate Array，FPGA）中应用设计实现的 Arm 处理器；

Cortex-M0 内核于 2009 年 2 月发布，目标是打造极低成本、极低功耗的 Arm 处理器，M0 内核并非 M3 内核的简化版，与采用哈佛结构的 M3 内核不同，M0 内核采用的是普林斯顿结构；

Cortex-M4 内核于 2010 年 2 月发布，在 M3 内核的基础上增加了浮点运算、数字信号处理（Digital Signal Processor，DSP）等功能，以满足数字信号控制市场的需求；

Cortex-M0+内核于 2012 年 3 月发布，在 M0 内核的基础上进一步降低了功耗并提高了性能；

Cortex-M7 内核于 2014 年 9 月发布，在 M4 内核的基础上进一步提升了计算性能和数字信号处理能力，主要面向高端嵌入式市场。

基于性能价格比考虑，目前市场上中端控制应用领域仍然以采用 Cortex-M3 系列处理器为