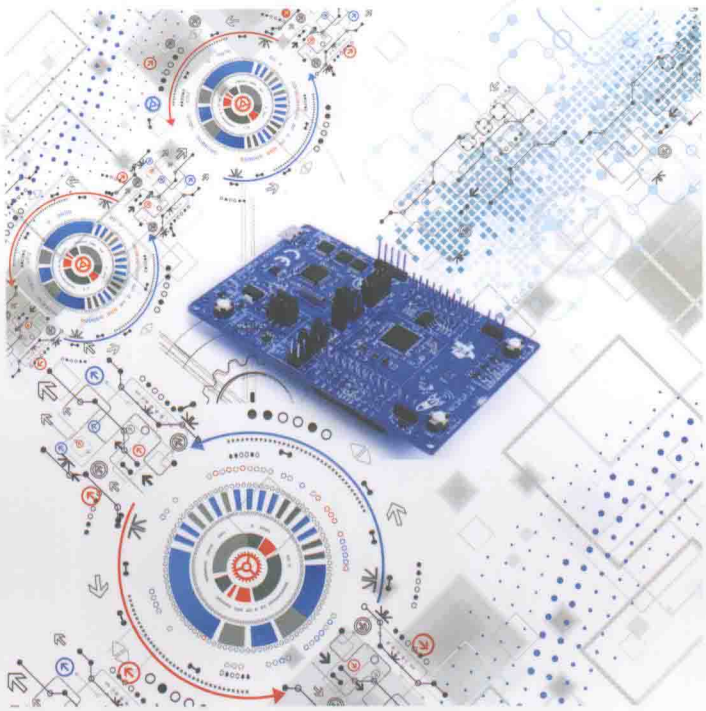
嵌入式技术与应用丛书

STM32F0系列

Cortex-M0原理与实践

张燕妮 主编　丁维才 副主编



中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

嵌入式技术与应用丛书

STM32F0 系列 Cortex-M0 原理与实践

张燕妮　主　编

丁维才　副主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以实际应用开发所需要的知识为主线并重点介绍如何解决开发过程中遇到的问题。全书共分为 17 章。首先分析了 STM32F0x 的性能及价格优势所在，即为何选择 STM32F0x，然后书中结合大量实例详细讲解了系统定时器、GPIO、NVIC、UART、I^2C、SPI、ADC、DAC、PWM、定时器、CAN 等外设，使用 STM32F0x 的固件库写各外设的例程，接着讲解 RTX 实时操作系统以及嵌入式程序结构的 4 种模式优缺点对比，最后分析了 USB 电流/电压监测需求，以及如何根据需求设计相应的解决方案。

本书可作为单片机用户的自学用书、嵌入式工程技术人员的学习和培训用书，也可作为大学生学习单片机的教材。

图书在版编目（CIP）数据

STM32F0 系列 Cortex-M0 原理与实践 / 张燕妮主编. —北京：电子工业出版社，2016.2
（嵌入式技术与应用丛书）
ISBN 978-7-121-28086-3

Ⅰ. ①S… Ⅱ. ①张… Ⅲ. ①微处理器 Ⅳ. ①TP332.3

中国版本图书馆 CIP 数据核字（2016）第 012111 号

责任编辑：刘海艳
印　　刷：北京京科印刷有限公司
装　　订：三河市华成印务有限公司
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：17　字数：435 千字
版　　次：2016 年 2 月第 1 版
印　　次：2016 年 2 月第 1 次印刷
印　　数：3 000 册　　定价：48.00 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

如果问一个设计人员，产品设计时，选择器件最关心的因素是什么，那么跟得上形势发展可能是常见的、很重要的一个因素。如果问老板选择器件最关心的因素是什么，可能只有“便宜”两个字。ST 公司的 STM32F0x 芯片是一款基于 Cortex-M0，兼顾性能与价格优势于一体的 32 位处理器。ST 公司的 Cortex-M3 系列产品已经在国内拥有大量客户群体，STM32F0x 芯片的用户群也在日益扩大中。

Cortex-M0 芯片具有小型、低功耗、低闸数、精简程序代码的特点，内建各种模拟与混合信号组件及多种高速通信能力器件，开发人员可以直接跳过 16 位系统，以接近 8 位系统的成本开销获取 32 位系统的性能。Cortex-M0 芯片是学习 ARM 处理器的最佳入门选择。

STM32F0x 在 Cortex-M0 芯片基础上继承了 ST 公司 Cortex-M3 系列产品优点，并对一些不足之处进行了修正（例如 I^2C 外设），还增加了一些优秀功能（例如 USART 的超时检测），适用于工业控制器、家庭自动化、打印机和白色家电、游戏机、DVD/蓝光播放机和音频/视频接收机等。

STM32F0x 具有如下特点。

- 百货迎百客，按需选择：STM32F0x 根据外设情况，将产品划分成多种系列，并不是一种大而全的芯片，用户可根据产品需求与成本选用不同芯片。例如，对于是否需要 CAN 通信就限制选择范围与成本。
- 一招鲜吃遍天下：STM32F0x 家族成员比较多，但不代表各自为政。STM32F0x 家族的外设与引脚位置不是一对一。在相同封装下 STM32F0x 芯片引脚定义相同，通过外设复用选用不同外设。而且借助 ST 公司提供 STM32F0x 固件库开发的应用程序可在 STM32F0x 家族（只要包含该外设）随意运行。书中的代码除了 CAN 通信一章，程序可以在各种 STM32F0x 上运行。

全书共 17 章。其中第 1～4 章是基础；第 5～15 章是外设功能讲解；第 16、17 章是综合实例与高级功能。

第 1 章首先讲解相比 8 位机，Cortex-M0 的优势，以及如何从 8 位机过渡到 Cortex-M0，并说明 Cortex-M0 的基础与特征，主要有寄存器、存储器映射、系统总线、存储器保护单元、嵌套中断控制器；最后一部分会说明 STM32F0x 在同类产品的优势以及特点，为何选用 STM32F0x 进行产品的开发。

第 2 章主要讲解了进行 STM32F0 软件开发需要准备的条件，分别介绍了 MDK 以及 J-ink、U-ink、Stlink 仿真器。MDK 是原 Keil 公司的产品，对于熟悉 Keil C51 单片机开发的用户会感受到 ARM 公司对 C8051 用户的重视，也非常欢迎该用户群体过渡到 Cortex-M0 的开发中。

第 3 章讲解有关硬件设计的基础，主要包括 STM32F0x 的电源、时钟，以及本书所使用的电路图。

第 4 章讲解进行 STM32F0x 软件开发的固件库内容。固件库内容涉及了 ARM 公司的 CMSIS 标准及 ST 公司的固件库标准。其中 CMSIS 是理解目前 ARM 对软件组织结构以及系统启动文件的规范，也是目前所有 Cortex-M 内核 CPU 软件设计的要求与基础。最后讲解了使用 ST 公司固件库建立工程文件的过程。

第 5 章～第 14 章讲解了 STM32F0x 的系统定时器、GPIO、NVIC、UART、I^2C、SPI、ADC、DAC、PWM、定时器、CAN。每 1 章均提供了 1～2 个实际项目中的使用实例，涉及实际开发中使用各种芯片的注意事项以及技巧（或者说作者的经验教训）。

第 16 章讲解 RTX 操作系统，介绍 Keil 自带的实时操作系统的原理与使用过程。结合第 4 章的超级循环和中断模式，读者可根据自己产品特点选用相应的软件程序框架。

第 17 章是关于如何使用 STM32F0x 设计一个 USB 的电流、电压检测器，从设计需求、硬件设计、软件设计等方面进行详细分析，引导读者如何做产品设计、开发。

本书的第 1～8 章、第 10～11 章和第 13～17 章由张燕妮编写，第 9 章由谢玲和王洪玲编写，第 12 章由贾芳和丁维才编写，同时丁维才对全书代码进行了验证。

书中讲解的源代码只摘取了与相应章节配套的部分进行了说明，需要完整代码，可从电子工业出版社电子信息出版分社（http://xxdz.phei.com.cn）下载。

感谢电子工业出版社的王敬栋、刘海艳两位编辑对本书的大力支持。感谢家人与朋友的理解和支持。

由于 STM32F0x 家族的新成员仍在不断增加中，并且作者水平有限以及时间仓促，难免有差错和不足之处恳请读者批评指正。

张燕妮

目　录

第 1 章

低成本单片机世界的入侵者——Cortex-M0

Cortex-M0 处理器是为迎合现在超低功耗和混合信号设备的需求而设计的。连接方式的多样化（例如以太网、USB、低功耗的无线）以及模拟传感器的使用（例如触摸传感器和加速度计）都带来了对处理器的新需求，通常这些应用均需高集成的模拟和数字功能来处理传输数据，而现有的 8 位和 16 位机需要付出增大代码量和提高时钟频率的代价才能满足应用要求。

Cortex-M0 可以在保持低功耗、低成本的前提下满足这些需求，因此 Cortex-M0 处理器的用户群体在快速增长。

Cortex-M0 处理器在 2009 年由 ARM 公司推出，向上兼容 ARM 的 Cortex-M3，并且损耗很低。Cortex-M0 在最小配置时仅有 12 000 逻辑门，同 8 位或者 16 位机一样的少，但却是完全的 32 位处理器。

本章将主要介绍 Cortex-M0 的基本原理，主要包含编程模型、存储器模型、异常模型、电源管理、指令集；并重点分析 Cortex-M0 相比 8 位机优势在哪里，以及如何从 8 位机过渡到 Cortex-M0；最后给出了 STM32F0 的特点。

1.1 相比 8 位（16 位）机为何要选择 Cortex-M0

Cortex-M0 处理器基于一个高集成、低功耗的 32 位处理器内核，采用 3 级流水线冯·诺伊曼结构（Von Neumann architecture）。通过简单、功能强大的指令集以及全面优化的设计（提供包括一个单周期乘法器在内的高端处理硬件），Cortex-M0 处理器可实现极高的能效。Cortex-M0 处理器采用 ARMv6-M 结构，使用 16 位的 Thumb 指令集，并包含 Thumb-2 技术，代码密度比 8 位和 16 位机都要高，这意味对于同样的程序可选用较少的 Flash 处理器，从而可节约成本和功耗。Cortex-M0 的执行周期是 0.896DMIPS/MHz，可使用较少的指令周期执行一个任务（即使 32 位乘法也可在一个周期内完成）。即使在处理不同优先级的嵌套中断情况，嵌套中断处理控制器（NVIC）也会使中断花销很少。

图 1-1 是 Cortex-M0 的具体实现框图。

Cortex-M0 处理器紧密集成了一个可配置的嵌套向量中断处理器（NVIC），提供业界领先的中断性能。NVIC 具有以下功能：

- 包含一个不可屏蔽的中断（NMI）；

- 提供零抖动中断选项；
- 提供 4 个中断优先级。

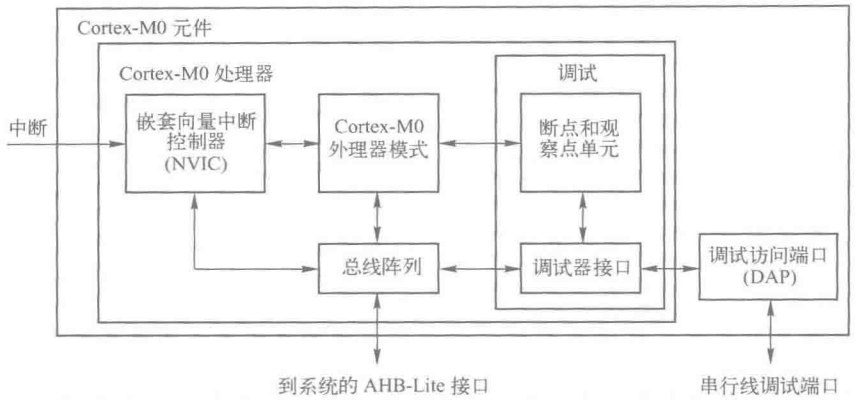


图 1-1　Cortex-M0 的具体实现框图

处理器内核和 NVIC 的紧密结合使得中断服务程序（ISR）快速执行，通过寄存器的硬件堆栈以及加载-乘和存储-乘操作的停止和重启等方式缩短了中断延迟时间。中断处理程序不需要任何汇编封装代码，不用消耗任何 ISR 代码。末尾连锁的优化大大地降低了从一个 ISR 切换到另一个 ISR 时的开销。为了优化低功耗设计，NVIC 还与睡眠模式相结合，提供一个深度睡眠功能，使整个器件能够迅速掉电。

Cortex-M0 处理器提供一个简单的系统级接口，使用 AMBA 技术来提供高速、低延迟的存储器访问。

Cortex-M0 处理器执行一个完整的硬件调试方案，带有大量的硬件断点和观察点选项。通过一个非常适合微控制器和其他小型封装器件的 2 脚串行线调试（SWD）端口，Cortex-M0 提供了一个高度透明的处理器、存储器和外设执行方式。

Cortex-M0 被大量的现代编译器支持。中断服务程序可直接使用 C 语言编码而无须汇编语言。而且指令集仅仅有 56 条指令，便于学习。尽管 Cortex-M0 是一个高性能 pipelined 处理器，指令和中断的执行时序是完全确定的（零抖动），从而使得设计人员可以预测和分析精确时序。支持多种调试方式，除了 ARM 自己公司的 MDK 软件，另有大量第三方软件支持，以及基于 ARM 构建的大量操作系统支持。

1.1.1　性能对比

8 位机或者 16 位机的用户为何要转向 Cortex-M0 呢？Cortex-M0 处理器与典型 16 位机差不多大小，但它具有比 16 位机更好的性能。表 1-1 是基于 DMIPS 的测试对比情况。

表 1-1　几种构架的 DMIPS 对比

体 系 结 构	基于Dhrystone 2.1的预估DMIPS/MHz
最初的80C51	0.0094
PIC18	0.01966

续表

体 系 结 构	基于Dhrystone 2.1的预估DMIPS/MHz
快速8051	0.113
H8S/300H	0.16
HCS12	0.19
MSP430	0.288
H8S/2600	0.303
S12X	0.34
PIC24	0.445
Cortex-M0	0.896

从表 1-1 可以看出，Cortex-M0 的处理速度比所有流行的 16 位处理都快，是最快的 8051 的 8 倍。

注：DMIPS:Dhrystone Million Instructions executed Per Second 主要用于测整数计算能力。Dhrystone: 1976 年发布，主要用于测整数计算能力，Dhrystone 的主体部分是一个由固定顺序指令构成的循环体。Dhrystone 的结果表现形式是微处理器执行固定次数的该循环体所用的时间，用 DMIPS 来表示。DMIPS 标准的优点在于可以比较衡量不同处理器的相对性能，并且被广泛应用于不同处理器间的性能比较。

1.1.2　8 位和 16 位体系结构的缺点

相比 8 位或 16 位机，Cortex-M0 无体系结构的缺陷。下面是 8 位或 16 位机的一些缺陷。

8 位和 16 位机结构的一个明显缺点是内存太小。程序和数据 RAM 太小限制了嵌入式产品的能力。其他的类似堆栈内存太小（例如 8051 堆栈位于内部 RAM，被限制成 256 字节，包含 register bank space）也影响软件设计。ARM 的体系结构的处理器，其内存空间是非常大的，堆栈位于系统内存中，软件设计非常灵活。

许多 8 位和 16 位微控器通过将内存空间划分成内存页方式访问较大内存。软件开发反而因此变得困难，因为在不同的内存页中存取地址不直截了当。因为在不同内存页过渡切换，从而增大了代码尺寸，还降低了性能。然而，ARM 微控器使用 32 位线性地址并不需要内存分页从而更容易使用，并提供更好的开发效率。

8 位微控器体系结构的另外一个缺点是指令集。例如，8051 过度依赖累加器寄存器来处理数据和内存传递，这样增加了代码尺长度。

地址模式也是影响 8 位机和 16 位机性能的一个因素。在 Cortex-M0 中有多种地址模式可用，从而保证代码长度和方便性。

1.1.3　Cortex 的软件移植性

相比 8 位机或者 16 位机，ARM 的 Cortex 处理器的编程是非常方便的。ARM 的 Cortex 微控器可全部使用 C 语言进行软件编程，从而缩短软件开发时间并提高软件的可移植性。即使软件人员计划使用汇编语言，指令集也是非常容易理解的。而且，因为编程模型类似 ARM7TDMI，熟悉 ARM 处理器的人员会很快熟悉 Cortex 微控器。

Cortex-M0 的构架可高效地实现嵌入式操作系统。在复杂应用中，使用嵌入式操作系统可轻松地处理并行任务。并且由于 ARM 是 IP（intellectual property）的供应商，ARM 处理器被广大微控器厂商采用，方便用户选型。

除了硬件，还有大量的嵌入式操作系统、代码库、开发工具以及其他资源供选择。良好的软件生态体系可将精力集中到开发中，提高产品开发速度。

1.2 如何从 8 位机过渡到 Cortex-M0

ARM 公司的 Joseph Yiu 和 Andrew Frame 从堆栈内存、数据类型、非对齐数据等方面分析 8 位机与 Cortex-M 的差异，以及如何从 8 位机过渡到 Cortex-M（包含 Cortex-M0、Cortex-M3、Cortex-M4）。

1. 内存空间

ARM 处理器采用 32 位寻址，可实现一个 4GB 的线性内存空间。该内存空间在结构上分成多个区。每个区都有各自的推荐用法（虽然并不是固定的）。统一内存架构不仅增加了内存使用的灵活性，而且降低了不同内存空间使用不同数据类型的复杂性。

相反地，8051 微控制器具有多个内存空间。内存空间的分割增加有效地利用全部内存空间的困难，而且需要借助 C 语言扩展来处理不同的内存类型。

8051 在外部 RAM 内存空间上最高支持 64KB 的程序内存和 64KB 的数据内存。理论上，可以利用内存分页来扩展程序内存大小。不过，内存分页解决方案并未标准化，换句话说，不同 8051 供应商的内存分页的实现并不相同。这不仅会增加软件开发的复杂性，而且由于处理页面切换增加软件开销，显著降低软件性能。

在 ARM Cortex-M3 或 Cortex-M4 上，SRAM 区和外设区都提供了一个 1MB 的位段区（bit band region）。此位段区允许通过别名地址访问其内部的每个位。由于位段别名地址只需通过普通的内存存取指令即可访问，因此被 C 语言完全支持，不需要任何特殊指令。而 8051 提供了少量的位寻址内存（内部 RAM 上 16 字节和 SFR 空间上 16 字节）。处理这些位数据需要特殊指令支持此功能，并且 C 编译器需要扩展。

图 1-2 是 Cortex-M 内存映射方式。图 1-3 是 8051 的内存映射方式。

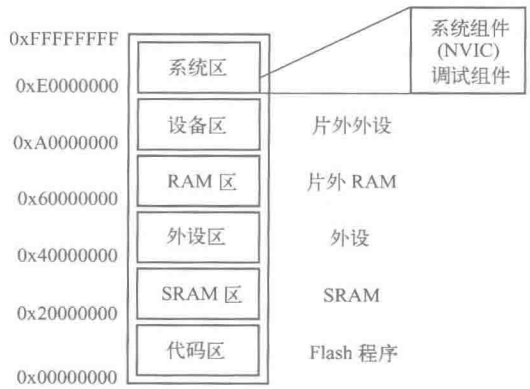


图 1-2 Cortex-M 的内存映射方式