ARM公司微控制器系统级设计专家Joseph Yiu享誉业界的代表作品!
ARM MCU工具总监Reinhard Keil作序!
系统论述Cortex-M0与Cortex-M0+的内核、体系结构、指令集、编译器、程序设计及软件移植的经典作品!

清華 开发者书库

The Definitive Guide to ARM Cortex-M0 and Cortex-M0+ Processors
(2nd Edition)

ARM Cortex-M0 与 Cortex-M0+ 权威指南

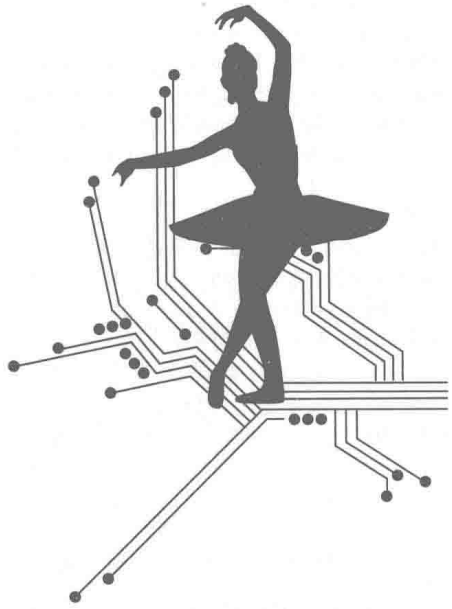
(第2版)

Joseph Yiu◎著
吴常玉 张淑 吴卫东◎译

清華大學出版社

ELSEVIER

开发者书库



The Definitive Guide to ARM Cortex-M0 and Cortex-M0+ Processors

(2nd Edition)

ARM Cortex-M0 与 Cortex-M0+ 权威指南

（第2版）

Joseph Yiu◎著

吴堂玉 张淘 吴卫东◎译

清华大学出版社

北京

The Definitive Guide to the ARM Cortex-M0 and Cortex-M0+ Processors, 2nd Edition
Joseph Yiu
ISBN: 9780128032770

图书在版编目(CIP)数据

ARM Cortex-M0 与 Cortex-M0+权威指南(第 2 版)/(英)姚文祥著;吴常玉,张淑,吴卫东译. —北京:清华大学出版社,2018
(清华开发者书库)
书名原文:The Definitive Guide to ARM Cortex-M0 and Cortex-M0+Processors,2nd Edition
ISBN 978-7-302-47331-2

Ⅰ. ①A… Ⅱ. ①姚… ②吴… ③张… ④吴… Ⅲ. ①微处理器—指南 Ⅳ. ①TP332-62

中国版本图书馆 CIP 数据核字(2017)第 124546 号

责任编辑:盛东亮 赵晓宁
封面设计:李召霞
责任校对:梁 毅
责任印制:王静怡

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
质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载:http://www.tup.com.cn,010-62795954
印 装 者:三河市铭诚印务有限公司
经 销:全国新华书店
开 本:186mm×240mm **印 张**:40.25 **字 数**:923 千字
版 次:2018 年 1 月第 1 版 **印 次**:2018 年 1 月第1次印刷
印 数:1~2500
定 价:128.00 元

产品编号:069089-01

译者序

FOREWORD

从2008年开始，基于Cortex-M3的单片机以其高性能、低成本及易于使用等诸多优势，已经取代ARM7，成长为32位微控制器的主流。而同属于Cortex系列的Cortex-M0和Cortex-M0+处理器则以低功耗、低成本为突出特点，其性能相对于传统的8位/16位处理器具有巨大优势，因此在市场上也占有一席之地。

从最初的8051、ARM7到今天的Cortex-M0和Cortex-M3等处理器，我使用单片机已经超过10年，在项目开发过程中，个人体会最深的就是芯片的高性能和易用性。

以前编写8位机的代码时，为了保证任务的正确执行，我们可能还需要考虑代码的执行速度能否满足任务的需求，而对于Cortex-M0，由于芯片本身的性能及编译器的效率，一般情况下，我们无须为执行时间而优化代码，编写单片机的代码就如同计算机程序一样方便。

另一方面，因为这些单片机具有诸多外设以及众多控制和状态寄存器，了解它们需要花费一定的时间，但厂家一般都提供了丰富的底层驱动库，结合开发工具提供的工程示例，我们无须在底层代码上花费太多时间，从而可以专注于应用功能的实现。由于这些单片机都具有相同的内核架构，因此熟悉了其中一个，其他产品也能很快上手。

在所有介绍Cortex-M0/M0+的书籍中，本书无疑是最经典的一本，一方面，作者本人就是ARM公司的专家，了解Cortex-M0/M0+架构的设计；另一方面，作者选取的角度非常合适，既有架构设计的细节，也有程序代码实现示例，而且对容易出现问题的地方进行了说明。另外，由于Cortex-M0/M0+处理器版本的更新，本书作为第二版，也对上一版的内容进行了修改和补充。

本书内容丰富，相信无论是新手还是熟练开发人员，都可以从中找到有用的信息。限于译者水平，疏漏之处敬请批评指正，最后希望这本书能给读者带来帮助。

译　者

2017年11月

推荐序
FOREWORD

我的职业生涯始于1982年，在微处理器软件部门工作，20世纪90年代主要使用的微控制器架构是8051，因为那时所有的嵌入式应用都是基于8051的。多年以来，嵌入式取得了巨大进展，且出现了许多种处理器架构。目前来说，由于芯片厂家和技术极多，微控制器市场已经非常分散。数年以前，每个嵌入式应用都是从头开始设计的，根本没有软件重用，而且工程师要应对项目的挑战，也需要从头培训。

但近年来，微控制器系统变得越来越复杂，且为了满足更多特性及方便操作的要求，所需的性能也越来越高。这些系统一般对价格敏感，因此许多微控制器系统都是基于高性能32位处理器的单芯片设计。同时，成本压力使得软件开发工作也需要标准化，而复杂些的I/O连接则会涉及多种设备。

为了解决这些挑战，业界已经将ARM Cortex-M处理器系列当作了标准微控制器架构。超过200家公司获得了这些处理器的授权，涉及标准微控制器、各种传感器以及完整的物联网无线通信系统等产品。

为了支持多种应用，ARM在Cortex-M架构的基础上开发了多款处理器，在低端领域，Cortex-M0和Cortex-M0+处理器可用于之前8位微控制器占主流的应用。毫无疑问，这些处理器目前已经广泛应用于低成本设备。

有了这些特性更加丰富的微控制器，基于这些设备的软件开发工作也变得更加复杂。实时操作系统和预编译中间件的使用，已经被无数人所接受，且在软件工程中的作用越发重要。对于开发人员而言，将这些部件组合起来可能会有一些问题，但遵循工业标准，可以降低系统开发成本，并加快推向市场的时间。Cortex-M处理器架构和CMSIS软件编程标准就是这种硬件和软件标准化的基础。

Joseph所著的本书介绍了基于ARM Cortex-M0或Cortex-M0+处理器的应用设计方法。由于这本书可以让你加深对自己日常工作的理解，因此我将这本书推荐给每位嵌入式工程师。

Reinhard Keil

ARM MCU工具总监

前言
PREFACE

从2011年开始，嵌入式系统技术有了很大的变化，当时本书的第一版刚好出版。2012年，ARM发布了Cortex-M0＋处理器，并在2014年发布了Cortex-M7。今天，Cortex-M处理器应用广泛，其中包括多种微控制器、混合信号以及无线通信芯片。

除了处理器设计，嵌入式软件开发技术也有了一定的进步，随着ARM Cortex-M微控制器的广泛应用，微控制器软件开发人员在开发方面也越发成熟。同时，随着开发组件的优化，人们也在继续改进电池寿命以及能耗效率。

有了这些变化，微控制器用户需要快速适应新的技术，本书的这一版也有了许多新信息和内容的提升。除了和Cortex-M0＋处理器有关的信息外，还介绍了使用多种常见开发组件的例子。例如，本书详细描述了微控制器低功耗特性的使用，并在一个简单的应用中使用了RTOS。

由于物联网(IoT)受到了越来越多的关注，且正在成为主流，越来越多的人都开始学习嵌入式编程。另外，许多高校在教学方面，也从老式的8位和16位微控制器转向了ARM Cortex-M等32位处理器。因此，本书中的许多部分都进行了重新编写，并且还加入了许多基本的例子，以满足初学者、学生或业余爱好者的需求。

当然，专业嵌入式软件开发人员、研究人员或半导体产品设计人员等许多读者都希望看到一些更加深入的信息，为了满足这些人的需求，本书还增加了不少技术细节以及高级应用示例。

希望你能从本书中学到东西，并在下一个项目中使用Cortex-M处理器时找到乐趣。

学习资源下载地址：http://booksite.elsevier.com/9780128032770。

Joseph

致谢

ACKNOWLEDGEMENTS

在本书以及本书第一版的调查和书写过程中，我得到了许多人的帮助。

首先要感谢给我提供第一版反馈的各位读者，有了你们的帮助，第二版才能有进步。

包括 Colin Jones 和 Edmund Player 在内的 ARM 公司不少同事，他们都对本书的内容进行了检查，许多公司也给予了我很大的帮助，比如 ST Microelectronics、Freescale 和 IAR Systems。

当然，没有第一版的成功，第二版也就无从谈起，因此我对下面这些在第一版时就给了我很大帮助的人员表示感谢：Amit Bhojraj、Bob（Robert）Boys、David Donley、Derek Morris、Dominic Pajak、Drew Barbier、Jamie Brettle、Jeffrey S. Mueller、Jim Kemerling、Joe Yu、John Davies、Jon Marsh、Kenneth Dwyer、Milorad Cvjetkovic、Nick Sampays、Reinhard Keil、Simon Craske 以及 William Farlow。

我还要感谢 Elsevier 的工作人员，本书的出版有赖于他们专业的工作。

最后，非常感谢我工作和生活中的朋友，有了他们不遗余力的鼓励，本书才得以开始并完成。

Joseph

术语和缩写

缩写	含义
AAPCS	ARM 架构过程调用标准
AHB	高级高性能总线
ALU	算术逻辑单元
AMBA	高级微控制器总线架构
APB	高级外设总线
API	应用编程接口
BE8	字节不变大端模式
BPU	断点单元
CMOS	互补金属氧化物半导体
CMSIS	Cortex 微控制器软件接口标准
CPU	中央处理单元
DAP	调试访问端口
DDR	双倍速率(存储器)
DS-5	开发平台 5
DWT	数据监视点和跟踪单元
EABI/ABI	嵌入式应用二进制接口
EWARM	IAR ARM 嵌入式开发平台
EXC_RETURN	异常返回
FPGA	现场可编程门阵列
GCC	GNU C 编译器
GPIO	通用输入输出
GPU	图形处理单元

HAL	硬件抽象层
ICE	在线仿真器
IDE	集成开发环境
ISA	指令集架构
ISR	中断服务程序
JTAG	联合测试行动小组(一种标准测试和调试接口)
LR	链接寄存器
LSB	最低位
MCU	微控制器单元
MDK/MDK-ARM	ARM Keil 微控制器开发套件
MSB	最高位
MTB	微跟踪缓冲
MSP	主栈指针
NMI	不可屏蔽中断
NVIC	嵌套向量中断控制器
OS	操作系统
PC	程序计数器
PCB	印刷线路板
PSP	进程栈指针
PSR/xPSR	程序状态寄存器
RTC	实时时钟
RVDS	ARM RealView 开发组件
RTOS	实时操作系统
RTX	Keil 实时执行内核
SCB	系统控制块
SCS	系统控制空间
SoC	片上系统

SP	栈指针
SPI	串行外设接口
SWD	串行线调试
TAP	测试访问端口
TRM	技术参考手册
UART	通用异步收发器
ULP	超低功耗
USB	通用串行总线
WIC	唤醒中断控制器

本书约定

本书中使用了多种约定，例如，

(1) 普通汇编程序代码：

```
MOV R0, R1   ;将数据从寄存器 R1 送至寄存器 R0 中
```

(2) 汇编程序语法("<>"中的内容需要被实际的寄存器名代替)：

```
MRS <reg>, <special_reg>
```

(3) 程序代码：

```
for (i = 0; i < 3; i++) {func1(); }
```

(4) 伪代码

```
if(a > b) {...
```

(5) 数值：

- 4'hC、0x123 都是十六进制数值；
- #3 表示 3 号条目(例如，IRQ#3 表示编号为 3 的 IRQ)；
- #immed_12 表示 12 位立即数；
- 寄存器位主要用于表示基于位的一部分数值，例如，bit[15:12]表示 15 到 12 位。

(6) 寄存器访问类型：

- R 为只读；
- W 为只写；
- R/W 为可读可写；
- R/Wc 表示可读，可被写操作清除。

目录

CONTENTS