

清华

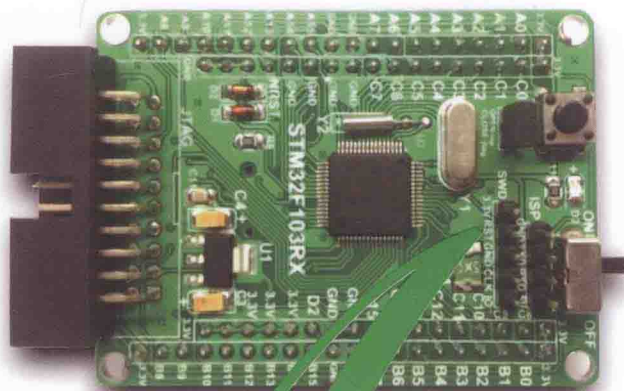
开发者书库

著名的嵌入式系统专家执笔，多年教学与项目开发经验的系统总结！

全面论述Cortex-M3的体系架构、程序设计及项目实战的经典著作！

书中完全公开5个实际项目，并提供全部源代码、设计图纸及演示视频

手把手教你学会ARM开发，QQ在线答疑



The Architecture and Programming of ARM Cortex-M3

ARM Cortex-M3 体系结构与编程

冯新宇◎编著

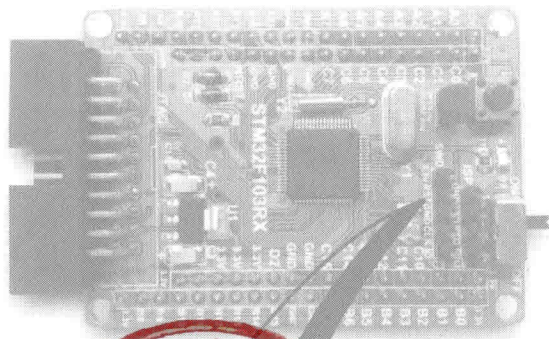
Feng Xinyu

清华大学出版社



清华

开发者书库



The Architecture and Programming of ARM Cortex-M3

ARM Cortex-M3 体系结构与编程

冯新宇◎编著

Feng Xinyu

清华大学出版社
北京

内 容 简 介

本书从 Cortex-M3 处理器入手,主要讲解其家族成员中较为常用的微处理器 STM32。全书从 STM 基本 I/O 端口配置讲起,深入浅出地介绍了该处理器的重要内容,包括基本 I/O 端口、中断、ADC、定时器等。最后 5 章给出了 5 个实际案例,有的案例来源于学生的电子设计大赛作品,有的案例来源于科研课题,例如平衡车设计、井下通信分站设计等。它们都较好地诠释了 STM32 的典型应用,可以快速地帮助读者入门并动手操作。

本书可作为电子、通信及控制等相关专业的参考书,也可以作为相关技术人员的参考资料。

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

ARM Cortex-M3 体系结构与编程/冯新宇编著.--北京:清华大学出版社,2016

(清华开发者书库)

ISBN 978-7-302-41720-0

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

中国版本图书馆 CIP 数据核字(2015)第 239623 号

责任编辑:盛东亮 赵晓宁

封面设计:李召霞

责任校对:胡伟民

责任印制:沈 露

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 刷 者:北京鑫丰华彩印有限公司

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:186mm×240mm 印 张:18.25

字 数:455 千字

版 次:2016 年 2 月第 1 版

印 次:2016 年 2 月第 1 次印刷

印 数:1~2500

定 价:45.00 元

产品编号:059548-01

前言

PREFACE

本书的名称为《ARM Cortex-M3 体系结构与编程》，更多讲解的是编程及应用开发，结构方面涉及较少。作为 Cortex-M3 重要的一员，STM32 是现在应用较多的一款芯片，从应用的角度出发，这本书实际是在讲 STM32 的应用，这个先和读者交代一下。

从 51 单片机的简单应用，演变到嵌入式、物联网、云计算、框计算等，越来越多的“概念”呈现在我们眼前，电子技术的日新月异，推动着相关行业的发展，改变着我们的生活。现在已经习惯把单片机相关的开发，统称为嵌入式开发。高校的授课主要以 51 单片机为主，有个别专业开始开设 M3 的选修课，STM32 作为其重要家族成员，正慢慢地被越来越多的学生学习。

2012 年，开始研发井下小型的通信基站，想选一款合适的芯片，后来选中了一款基于 STM32 的工业核心开发板作为主控制器，该项目已经实际运行使用，读者可以作为蓝本，在此基础上进一步开发、学习。该项目作为一个案例，出现在本书的最后一章。当真正开始着手写本书时，发现无从下手，一拖再拖，因为 STM32 的内容真的很多，资料短时间内整理起来又很困难，从一本书的角度很难说得透彻、清晰，所以有些概念的理解还是希望读者有一点 51 单片机和 C 语言的基础。本书所列出的章节都是学生日常参加电子设计大赛、毕业设计用的一些内容，STM32 本身很多重要的应用并未列入其中。同时，互联网上有很多优秀的电子资源，比较适合作为初学者学习的素材，例如“野火”、“战舰”、“原子”等，本书很多想法和内容也来自它们。很多学生大二开始学习 STM32，从流水灯开始，做普通的巡线小车、小平衡车到最后做出能载 100 多公斤并自由行进的大平衡车，整个学习过程不到一年，但他们收获了很多知识。STM32 入门相对 51 单片机复杂一些，但是 STM32 的使用要比 51 单片机更容易和便捷，书中的很多例子来源于笔者指导的毕业设计和电子设计大赛，后面章节特别涉及最小系统设计、电源设计、电机驱动设计，从全书看是有一些重复的，但是作为独立的设计，这种重复还是必要的，希望读者理解。书中涉及最小系统、数码管显示电路、键盘模块、巡线模块、各种驱动电源模块，都已经做成了标准的 PCB，在实际教学实验中使用，读者可以方便修改制作。这本书完成匆忙，很多东西加工得不是很细致，留个 QQ 群号 185156135，方便读者交流学习。

本书的很多素材资料来源于其他老师和学生，在此一并感谢！参与本书编写的还有范

红刚老师、宋一兵、管殿柱、王献红、李文秋、赵景波、张忠林、曹立文、张凯、李民杰、张成照、梁亮、宋熠林和张学飞等。

感谢我的家人，假期陪我一起工作！

冯新宇

2016年1月

学习说明

STUDY SHOWS

本书工程文件下载地址

(1) 清华大学出版社网站本书页面,网址:

<http://www.tup.tsinghua.edu.cn>

(2) baidu 网盘,网址:

<http://pan.baidu.com/s/1bnUlQ5T>

注意: 本书配套资源仅限购买本书的读者个人学习使用,不得以任何方式传播!

本书学习交流联络方式

(1) 微信公众号:嵌入式系统设计(emsyde)

(2) QQ 群号:185156135

(3) 作者电子邮件:88574099@163.com

(4) 嵌入式开发硬件资源:<https://shop58461739.taobao.com/?spm=alzl0.l-c.0.0.N63Uxr>



目录

CONTENTS

第 1 章 ARM Cortex-M3 核介绍	1
1.1 Cortex-M3 主要特性	1
1.2 典型 M3 核处理器特性	3
1.2.1 命令规则	4
1.2.2 产品功能和外设配置	4
1.3 习题	5
第 2 章 开发环境搭建	6
2.1 MDK 安装	6
2.2 新建工程	9
2.3 采用 ST-Link 调试仿真代码	15
2.4 习题	19
第 3 章 基本 I/O 端口控制	20
3.1 MDK 新建工程	20
3.2 MDK 工程配置	27
3.3 库函数操作代码分析	30
3.4 时钟配置	30
3.4.1 时钟树	31
3.4.2 时钟源	32
3.4.3 APB2 外设时钟使能寄存器(RCC_APB2ENR)	33
3.5 I/O 端口配置	34
3.5.1 I/O 基本情况	34
3.5.2 GPIO 配置寄存器描述	35
3.5.3 端口输出数据寄存器(GPIOx_ODR) (x= A~E)	37
3.6 用库函数操作流水灯	37
3.6.1 GPIO_Init 函数	39

3.6.2	RCC_APB2PeriphClockCmd	41
3.6.3	控制 I/O 输出电平	42
3.6.4	LED.h 文件	43
3.6.5	软件调试易现问题	44
3.7	使用库函数法控制数码管	46
3.7.1	数码管基础知识	46
3.7.2	硬件电路设计	47
3.7.3	软件说明	47
3.8	简单按键输入	50
3.9	习题	52
第 4 章	中断	53
4.1	STM32 中断和异常	53
4.2	STM32 中断相关的基本概念	55
4.2.1	优先级	55
4.2.2	中断控制器 NVIC	56
4.2.3	NVIC 的优先级组	58
4.3	外部中断	58
4.3.1	外部中断基本情况	59
4.3.2	使用外部中断的基本步骤	60
4.4	习题	64
第 5 章	串口通信	65
5.1	串口通信基础	65
5.1.1	基本概念	65
5.1.2	常用的串行通信接口	66
5.1.3	应用串行通信的数据采集结构	70
5.2	STM32 串口操作	71
5.2.1	寄存器方式操作串口	72
5.2.2	库函数方式操作串口	77
5.3	习题	82
第 6 章	直接寄存器访问(DMA)	83
6.1	DMA 基础知识	83
6.2	STM32 的 DMA 操作	85
6.2.1	寄存器方式操作 DMA	85

6.2.2	库函数方式操作 DMA	90
6.2.3	DMA 操作实例	93
6.3	习题	96
第 7 章	模拟数字转换(ADC)	97
7.1	ADC 基础知识	97
7.1.1	ADC 主要特性	97
7.1.2	ADC 框图及引脚分布	98
7.1.3	通道选择	99
7.1.4	ADC 的转换模式	99
7.1.5	ADC 寄存器和固件库函数列表	100
7.2	STM32ADC 操作	102
7.2.1	寄存器方式操作 ADC	102
7.2.2	库函数方式操作 ADC	111
7.2.3	ADC 操作实例	115
7.3	习题	118
第 8 章	定时器	119
8.1	定时器基础知识	119
8.1.1	高级定时器	119
8.1.2	基本定时器	120
8.1.3	通用定时器	120
8.2	STM32 定时器操作	121
8.2.1	寄存器方式操作定时器	121
8.2.2	库函数方式操作定时器	127
8.2.3	定时器操作实例	130
8.3	习题	131
第 9 章	CAN 总线设计	132
9.1	CAN 总线基本工作原理	132
9.2	CAN 协议的特点	133
9.3	CAN 协议通信过程	134
9.4	CAN 的报文格式	135
9.4.1	数据帧	136
9.4.2	遥控帧	137
9.4.3	错误帧	138

9.4.4	过载帧	139
9.4.5	帧间隔	140
9.4.6	优先级的决定	141
9.5	CAN 总线错误处理机制	141
9.5.1	错误状态	142
9.5.2	错误检测	143
9.6	同步	144
9.6.1	同步类型	144
9.6.2	同步原则	144
9.7	CAN 总线拓扑结构	145
9.7.1	STM32 的 CAN 通信模块	145
9.7.2	CAN 控制器 MCP2515 介绍	146
9.8	CAN 通信的软件设计	148
9.8.1	系统程序流程	148
9.8.2	系统接收发送中断处理	149
9.8.3	CAN 总线初始化配置	149
9.8.4	报文的发送	151
9.8.5	报文的接收	152
9.9	CAN 通信示例	153
9.10	习题	157
第 10 章	倒立摆设计	158
10.1	设计内容与实现指标	158
10.1.1	倒立摆的选择	158
10.1.2	系统设计指标	159
10.2	系统方案确定	160
10.2.1	系统结构组成	160
10.2.2	系统模型分析	160
10.2.3	系统控制方案确定	164
10.3	系统硬件设计	165
10.4	电机的选择及驱动电路的设计	166
10.4.1	电机的选择	166
10.4.2	电机驱动电路的设计	166
10.5	测量电路设计	169
10.5.1	摆杆角度测量电路的设计	169
10.5.2	旋臂位置测量电路的设计	170

10.6	通信电路的设计	172
10.6.1	上位机通信电路的设计	172
10.6.2	无线传输电路的设计	174
10.7	辅助电路设计	175
10.7.1	语音提示电路的设计	175
10.7.2	电源电路的设计	177
10.8	系统软件设计	181
10.8.1	系统控制程序设计	181
10.8.2	起摆程序设计	182
10.8.3	PID 控制程序设计	183
10.8.4	电机驱动程序设计	185
10.8.5	上位机通信程序设计	186
10.8.6	无线通信程序设计	188
10.9	作品的制作与调试	189
10.9.1	倒立摆机械结构的制作问题	189
10.9.2	PCB 设计应注意的问题	189
10.9.3	电路板的制作问题	190
10.10	PID 参数的整定	190
10.10.1	比例参数整定	191
10.10.2	积分参数整定	191
10.10.3	微分参数整定	191
10.11	习题	192
第 11 章	智能小车设计	193
11.1	硬件电路设计	193
11.1.1	硬件系统方案设计	193
11.1.2	最小系统电路设计	194
11.1.3	电源电路设计	195
11.1.4	电机驱动电路设计	197
11.1.5	环境检测传感器电路设计	199
11.2	人机交互电路设计	201
11.2.1	OLED 显示电路设计	201
11.2.2	红外遥控电路设计	201
11.2.3	蜂鸣器提示电路设计	202
11.3	总体软件设计	202
11.3.1	道路基准采集模式软件	202

11.3.2	PID 寻迹模式软件	203
11.3.3	迷宫模式软件	203
11.3.4	OLED 显示软件设计	204
11.4	PID 控制软件设计	205
11.4.1	PID 介绍	205
11.4.2	比例(P)控制器	206
11.4.3	比例积分(PI)控制器	206
11.4.4	比例微分(PD)控制器	207
11.4.5	比例积分微分(PID)控制器	207
11.4.6	PID 寻迹	208
11.5	迷宫算法设计	210
11.5.1	左手法	210
11.5.2	迷宫搜索	211
11.5.3	迷宫最短路径算法	211
11.6	设计测量方法与数据处理	212
11.6.1	传感器分布	212
11.6.2	五路模拟传感器数据测量	212
11.7	传感器软件滤波	214
11.7.1	软件滤波处理介绍	214
11.7.2	软件滤波的方法	214
11.8	调试方法	215
11.8.1	PID 参数调试	215
11.8.2	迷宫模式调试	216
11.9	习题	217
第 12 章	平衡车设计	218
12.1	硬件电路设计	218
12.1.1	硬件系统方案设计	218
12.1.2	环境检测传感器电路设计	219
12.2	人机交互电路设计	220
12.3	MPU-6050 使用方法	222
12.3.1	引脚说明	222
12.3.2	SMPRT_DIV 寄存器	222
12.3.3	CONFIG 寄存器	223
12.3.4	GYRO_CONFIG 寄存器	224
12.3.5	ACCEL_CONFIG 寄存器	224

12.3.6	加速度计测量寄存器	225
12.3.7	TEMP_OUT_H 和 TEMP_OUT_L 寄存器	226
12.3.8	陀螺仪测量寄存器	226
12.3.9	PWR_MGMT_1 寄存器	227
12.3.10	WHO_AM_I 寄存器	228
12.4	总体软件设计	228
12.4.1	车身状态采集模式软件	228
12.4.2	PID 车身保持模式软件	231
12.4.3	人机交互模式软件设计	234
12.4.4	卡尔曼滤波算法	237
12.5	习题	239
第 13 章	电子秤设计	240
13.1	设计指标	240
13.2	设计方案	240
13.3	硬件电路设计说明	241
13.3.1	主控制器相关电路	241
13.3.2	TFT 液晶屏相关电路设计	242
13.3.3	AD 芯片 HX711 相关电路设计	243
13.3.4	WT588D 语音模块相关电路设计	244
13.3.5	称重传感器相关电路设计	246
13.4	软件设计思路及代码分析	247
13.4.1	TFT 触控液晶模块部分	247
13.4.2	WT588D 语音模块部分	248
13.4.3	HX711 芯片部分	249
13.4.4	DS18B20 芯片部分	251
13.4.5	数据计算部分	254
13.5	习题	255
第 14 章	井下通信分站设计	256
14.1	硬件电路设计	256
14.1.1	监控分站主要设计目标及参数	256
14.1.2	硬件电路设计方案	257
14.2	软件方案设计	260
14.2.1	软件总体程序的思路	260
14.2.2	RS485 接口的使用及程序流程	264

14.2.3	CAN 数据传输	267
14.2.4	OLED 显示	270
14.2.5	键盘输入	273
14.3	习题	275
参考文献		276

Cortex-M3 采用 ARM V7 构架,不仅支持 Thumb-2 指令集,而且拥有很多新特性。较之 ARM7 TDMI,Cortex-M3 拥有更优的性能、更高的代码密度、可嵌套中断、低成本、低功耗等众多优势。

国内 Cortex-M3 市场,ST(意法半导体)公司的 STM32 无疑是最大赢家,ST 无论市场占有率,还是技术支持方面,都远超其他对手。Cortex-M3 芯片的选择上,STM32 无疑是大家的首选,而且可以比较方便地购买。目前,在以下领域有较广泛的应用:

- (1) 医疗和手持设备;
- (2) PC 游戏机外设和 GPS 平台;
- (3) 工业应用中可编程控制器(PLC)、变频器、打印机和扫描仪等;
- (4) 警报系统、视频对讲和暖气通风空调系统等。

1.1 Cortex-M3 主要特性

Cortex-M3 是 ARM 公司基于 ARM V7 架构的基础上设计出来的一款新型的芯片内核。相对于其他 ARM 系列的微控制器,Cortex-M3 内核拥有以下优势和特点:

1. 三级流水线和分支预测

现代处理器中,大多数都采用了指令预存及流水线技术,来提高处理器的指令运行速度。执行指令的过程中,如果遇到了分支指令,由于执行的顺序也许会发生改变,指令预取队列和流水线中的一些指令就可能作废,需要重新取相应的地址,这样会使得流水线出现“断流现象”,处理器的性能会受到影响。尤其在 C 语言程序中,分支指令的比例能达到 10%~20%,这对于处理器来说无疑是一件很恐怖的事情。因此,现代高性能的流水线处理器都会就一些分支预测的部件,在处理器从存储器预取指令的过程中,当遇到分支指令时,处理器就能自动预测跳转是否会发生,然后才从预测的方向进行相应地取值,从而让流水线能连续地执行指令,保证它的性能。

2. 哈佛结构

哈佛结构式的处理器采用独立的数据总线和指令总线,处理器可以同时进行对指令和

数据的读写操作,使得处理器的运行速度得以提高。

3. 内置嵌套向量中断控制器

Cortex-M3 首次在内核部分采用了嵌套向量中断控制器,即 NVIC。也正是采用了中断嵌套的方式,使得 Cortex-M3 能将中断延迟减小到 12 个时钟周期(一般,ARM7 需要 24~42 个时钟周期)。Cortex-M3 不仅采用了 NVIC 技术,还采用了尾链技术,从而使中断响应时间减小到了 6 个时钟周期。

4. 支持位绑定操作

在 Cortex-M3 内核出现之前,ARM 内核是不支持位操作的,而是要用逻辑与、或的操作方式来进行屏蔽对其他位的影响。这样的结果带来的是指令的增加和处理时间的增加。Cortex-M3 采用了位绑定的方式让位操作成为可能。

5. 支持串行调试(SWD)

一般的 ARM 处理器采用的都是 JTAG 调试接口,但是 JTAG 接口占用的芯片 I/O 端口过多,这对于一些引脚少的处理器来说很浪费资源。Cortex-M3 在原来的 JTAG 接口的基础上增加了 SWD 模式,只需要两个 I/O 端口即可完成仿真,节约了调试占用的引脚。

6. 支持低功耗模式

Cortex-M3 内核在原来的只有运行/停止的模式上增加了休眠模式,使得 Cortex-M3 的运行功耗也很低。

7. 拥有高效的 Thumb2 16/32 位混合指令集

原有的 ARM7、ARM9 等内核使用的都是不同的指令,例如 32 位的 ARM 指令和 16 位的 Thumb 指令。Cortex-M3 使用了更高效的 Thumb2 指令来实现接近 Thumb 指令的代码尺寸,达到 ARM 编码的运行性能。Thumb2 是一种高效的,紧凑的新一代指令集。

8. 32 位硬件除法和单周期乘法

Cortex-M3 内核加入了 32 位的除法指令,弥补了一些除法密集型运用中性能不好的问题。

同时,Cortex-M3 内核也改进了乘法运算的部件,使得结果的 32 位乘 32 位的乘法在运行时间上减少到了一个时钟周期。

9. 支持存储器非对齐模式访问

Cortex-M3 内核的 MCU 一般用的内部寄存器都是 32 位编址。如果处理器只能采用对齐的访问模式,那么有些数据就必须被分配,占用一个 32 位的存储单元,这是一种浪费的现象。为了解决这个问题,Cortex-M3 内核采用了支持非对齐模式的访问方式,从而提高了存储器的利用率。

10. 内部定义了统一的存储器映射

在 ARM7、ARM9 等内核中没有定义存储器的映射,不同的芯片厂商需要自己定义存储器的映射,这使得芯片厂商之间存在不统一的现象,给程序的移植带来了麻烦。Cortex-M3 则采用了统一的存储器映射的分配,使得存储器映射得到了统一。

11. 极高的性价比

Cortex-M3 内核的 MCU 相对于其他的 ARM 系列的 MCU 性价比高许多。

1.2 典型 M3 核处理器特性

以 STM32F103xxx 为例,介绍其主要特性。中等容量增强型主要特性有:

1. 内核: ARM 32 位的 Cortex-M3 CPU

- (1) 最高 72MHz 工作频率,在存储器的 0 等待周期访问时,可达到 1.25Dmips/MHz。
- (2) 单周期乘法和硬件除法。

2. 存储器

- (1) 64KB~128KB 的闪存程序存储器。
- (2) 高达 20KB 的 SRAM。

3. 时钟、复位和电源管理

- (1) 2.0~3.6V 电压和 I/O 引脚。
- (2) 上电/断电复位(POR/PDR)、可编程电压监测器(PVD)。
- (3) 4~16MHz 晶体振荡器。
- (4) 内嵌经出厂调校的 8MHz 的 RC 振荡器。
- (5) 内嵌带校准的 40KHz 的 RC 振荡器。
- (6) 产生 CPU 时钟的 PLL。
- (7) 带校准功能的 32KHz RTC 振荡器。

4. 低功耗

- (1) 睡眠、停机和待机模式。
- (2) VBAT 为 RTC 和后备寄存器供电。

5. 两个 12 位模数转换器,1 μ s 转换时间(多达 16 个输入通道)

- (1) 转换范围为 0~3.6V。
- (2) 双采样和保持功能。
- (3) 温度传感器。

6. DMA

- (1) 7 通道 DMA 控制器。
- (2) 支持的外设,包括定时器、ADC、SPI、I2C 和 USART。

7. 多达 80 个快速 I/O 端口

- (1) 26/37/51/80 个 I/O 端口,所有 I/O 端口可以映射到。
- (2) 16 个外部中断,几乎所有端口均可容忍 5V 电压信号。

8. 调试模式

- (1) 串行单线调试(SWD)和 JTAG 接口。