

Broadview®  
www.broadview.com.cn

# ARM

# Cortex-M3

## 应用开发 实例详解

刘波文 编著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# ARM Cortex-M3

## 应用开发实例详解

刘博文 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书针对市场上新型热门的 ARM 芯片 SAM3U Cortex-M3 系列, 通过大量工程案例, 系统深入地介绍了 ARM Cortex-M3 基础应用与综合开发的流程、方法和技巧。全书共分 17 章, 第 1、2 章简要介绍了 Cortex-M3 架构及指令系统、ARM RealView MDK 开发环境; 第 3~13 章结合实例详细介绍了 I/O 端口、A/D、D/A、LCD、数码管显示、SPI、I<sup>2</sup>C、USB、以太网控制器、NOR Flash、SD 的开发原理和实际应用; 第 14~17 章介绍了数据传输存储显示、永磁电机驱动控制、车载智能终端导航及 RFID+GPRS 无线物联网技术 4 个综合开发项目。全书层次清晰、技术先进; 实例丰富、典型, 从基础应用到综合实例, 覆盖范围广, 实践指导性强, 深入展示了 ARM Cortex-M3 处理器的应用开发技术与经验。同时书中实例均系作者原创, 通用性和可移植性强, 方便读者举一反三, 并能快速地学以致用。

本书适合高校计算机、电子、通信、自动控制等相关专业的大学生, 以及从事 ARM 嵌入式开发的科研工程师使用, 是读者学习 ARM Cortex-M3 应用的理想参考书。

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 Cortex-M3 应用开发实例详解/刘博文编著.--北京: 电子工业出版社, 2011.2

ISBN 978-7-121-12610-9

I. ①A... II. ①刘... III. ①微控制器 IV. ①TP332.3

中国版本图书馆 CIP 数据核字 (2010) 第 249785 号

策划编辑: 张春雨

责任编辑: 许 艳

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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: 31.25 字数: 800 千字

印 次: 2011 年 2 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 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

ARM Cortex-M3 是一种基于 ARM7v 架构的最新 ARM 嵌入式内核，ARM 公司对 Cortex-M3 的定位是向专业嵌入式市场提供低成本、低功耗的芯片。因此在成本和功耗方面，Cortex-M3 具有相当好的性能；与传统的 ARM 7 相比，它具有更小的基础内核，价格更低，速度更快。此外，Cortex-M3 还集成了睡眠模式和可选的完整的八区域存储器保护单元，采用 THUMB-2 指令集，最大限度地降低了汇编器使用率。随着 ARM Cortex-M3 高校普及计划的展开，ARM Cortex-M3 时代会逐渐到来。

目前，市场上的相关图书比较少，仅有的几本只是基础入门教程，根本无法满足读者的需求。为了解决这类问题，本书将重点结合大量实际项目案例来讲解，介绍处理器芯片功能的同时，又深化了编程设计的应用，让读者迅速入门和提高。

全书共计 17 章，具体内容安排如下：

第 1、2 章简要介绍 Cortex-M3 架构及指令系统、ARM RealView MDK 开发环境。读者通过学习，将对 ARM Cortex-M3 有个入门性的认识，为后续的学习打下基础。

第 3~13 章详细介绍了 I/O 端口、A/D、D/A、LCD、数码管显示、SPI、I<sup>2</sup>C、USB、以太网控制器、NOR Flash、SD 卡的开发原理和实际应用。为了便于读者加深理解，巩固学习效果，每章都结合具体的应用示例来讲解。

第 14~17 章详细介绍了数据传输存储显示、永磁电机驱动控制、车载智能终端导航及 RFID+GPRS 无线物联网技术 4 个综合开发项目，对前面的知识进行了综合性运用。读者通过学习，编程水平可以得到快速提升，实现从入门到精通的目标。

本书适合高校计算机、电子、通信、自动控制等相关专业的学生，以及从事 ARM 嵌入式开发的工程师使用，是读者自学 ARM Cortex-M3 的必备参考书。

与同类书比较，本书特色和优势主要如下：

(1) 实例丰富、技术新潮：精选了 15 个典型工程实例，包括基础应用及综合应用实例，从入门到精通，深入介绍了 ARM Cortex-M3 芯片基础应用与综合项目开发的流程、方法和技巧，实践指导性强。

(2) 全书工程实例以“硬件电路设计+软件流程分析+软件设计与程序注释+程序运行实时追踪”形式讲授，深入展示了 ARM Cortex-M3 处理器的应用开发技术与技巧。书中实例均系作者原创，可移植性强，读者稍加修改便可应用于自己的工作或者课题（毕业设计）中，物超所值。

(3) 提供有限赠送图书配套开发板活动，加强互动。为促进读者更好地学习 Cortex-M3 处理器，作者还设计并制作了配套开发板和实验课件资源，有需要的读者通过发邮件（[powenliu@yeah.net](mailto:powenliu@yeah.net)）进行问题验证后即可得到。

## 鸣谢

感谢深圳英蓓特信息技术有限公司提供 RealView MDK 开发套件，特别感谢英蓓特公司的资深工程师景朝斌，对本书提供开发板、程序代码完善及技术支持；感谢 ST 公司梁平、吴建明先

生在资料收集及电机套件案例方面提供的大力协助。同时感谢武汉理工大学崔虎威、公安一所的徐晓丽，分别为本书资料收集及原理图制作方面所做的无私奉献。

本书由刘博文编著，另外编写编写的人还有：黄国灿、黄红光、崔虎威、赵汶、唐清善、邱宝良、李宁宇、黄小宽、付军鹏、张广安、金平、徐春林、谢正义、郑贞平等，在此一并表示感谢！

由于水平有限，再加之时间仓促，书中不足和错误之处在所难免，恳请专家和读者批评指正。

编 著 者

# 目 录

第 1 章 Cortex-M3 及 ST 微处理器概述	1
1.1 Cortex-M3 处理器特点	1
1.1.1 高性能	1
1.1.2 采用 Thumb-2 指令集具有更高的效率和性能	1
1.1.3 先进的中断处理功能	2
1.1.4 总线矩阵和接口的应用	2
1.1.5 调试支持	3
1.1.6 低成本低功耗、易于使用	3
1.2 Cortex-M3 内核结构	4
1.2.1 Cortex-M3 的基本结构	4
1.2.2 寄存器组	4
1.2.3 处理器工作模式和状态	8
1.2.4 异常和中断	9
1.2.5 向量表	9
1.2.6 堆栈实现与双堆栈机制	10
1.3 Cortex-M3 的指令集	10
1.4 STM32F 系列微控制器简介	16
1.4.1 STM32F 微控制器的主要特点	16
1.4.2 系统架构	16
1.4.3 存储器组织	18
1.4.4 STM32F 处理器的启动配置	20
1.4.5 STM32F 微控制器的应用	21
第 2 章 RealView MDK 开发工具	22
2.1 RealView MDK 概述	22
2.2 $\mu$ Vision4 集成开发环境简介	26
2.3 $\mu$ Vision4 软件开发流程	27
2.3.1 使用项目	28
2.3.2 源文件	31
2.4 $\mu$ Vision4 编译器选项	32
2.5 项目编译和生成	34
2.5.1 编译项目	34
2.5.2 项目生成	35
2.6 对目标设备编程 (Flash 编程器)	35
2.6.1 Flash 编程器功能简介	35
2.6.2 Flash 编程器工具配置	36

2.7	调试程序 (Debug)	37
2.7.1	μVision4 调试器简介	37
2.7.2	μVision4 调试器的使用	38
2.8	简易入门实例	40
2.8.1	编辑 Hello.C	41
2.8.2	编译、链接 Hello 例程	41
2.8.3	测试程序	41
2.9	工具集的使用技巧和窍门	42
2.9.1	启动代码的图形化配置	42
2.9.2	仿真 I/O 端口	44
2.9.3	仿真中断和时钟中断	44
2.9.4	为 PC 的 COM 端口分配串行 I/O	45
2.9.5	向文件中写入调试输出信息	45
第 3 章	I/O 端口应用	46
3.1	通用 I/O 端口 (GPIO) 功能描述	46
3.1.1	单独的位设置或位清除	47
3.1.2	外部中断/唤醒线	47
3.1.3	复用功能 (AF)	47
3.1.4	软件重新映射 I/O 复用功能	48
3.1.5	GPIO 锁定机制	48
3.1.6	输入配置	48
3.1.7	输出配置	49
3.1.8	复用功能配置	49
3.1.9	模拟输入配置	50
3.1.10	外设的 GPIO 配置	50
3.2	GPIO 相关寄存器描述	50
3.2.1	端口配置低寄存器 (GPIOx_CRL) (x=A..G)	50
3.2.2	端口配置高寄存器 (GPIOx_CRH) (x=A..G)	51
3.2.3	端口输入数据寄存器 (GPIOx_IDR) (x=A..G)	52
3.2.4	端口输出数据寄存器 (GPIOx_ODR) (x=A..G)	52
3.2.5	端口位设置/清除寄存器 (GPIOx_BSRR) (x=A..G)	53
3.2.6	端口位清除寄存器 (GPIOx_BRR) (x=A..G)	53
3.2.7	端口配置锁定寄存器 (GPIOx_LCKR) (x=A..G)	54
3.3	应用实例——按键控制驱动 LED	54
3.3.1	设计要求	55
3.3.2	硬件电路设计	55
3.3.3	应用程序流程图设计	56
3.3.4	应用程序代码设计	56

3.3.5	程序运行追踪	61
3.3.6	实例注意事项总结	61
第 4 章	A/D 转换	62
4.1	A/D 转换器 (ADC) 介绍	62
4.1.1	A/D 转换器功能描述	62
4.1.2	校准	67
4.1.3	数据对齐	67
4.1.4	可编程的通道采样时间	68
4.1.5	外部触发转换	68
4.1.6	DMA 请求	69
4.1.7	双 ADC 模式	69
4.1.8	ADC 中断	70
4.2	ADC 相关寄存器设置及功能说明	70
4.2.1	ADC 状态寄存器 (ADC_SR)	70
4.2.2	ADC 控制寄存器 1 (ADC_CR1)	71
4.2.3	ADC 控制寄存器 2 (ADC_CR2)	72
4.2.4	ADC 采样时间寄存器 1 (ADC_SMPR1)	74
4.2.5	ADC 采样时间寄存器 2 (ADC_SMPR2)	75
4.2.6	ADC 注入通道数据偏移寄存器 x (ADC_JOFR <sub>x</sub> ) (x=1..4)	75
4.2.7	ADC 看门狗高阈值寄存器 (ADC_HTR)	76
4.2.8	ADC 看门狗低阈值寄存器 (ADC_LRT)	76
4.2.9	ADC 规则序列寄存器 1 (ADC_SQR1)	77
4.2.10	ADC 规则序列寄存器 2 (ADC_SQR2)	77
4.2.11	ADC 规则序列寄存器 3 (ADC_SQR3)	78
4.2.12	ADC 注入序列寄存器 (ADC_JSQR)	78
4.2.13	ADC 注入数据寄存器 x (ADC_JDR <sub>x</sub> ) (x=1..4)	79
4.2.14	ADC 规则数据寄存器 (ADC_DR)	79
4.3	应用实例——压力传感器 A/D 转换	80
4.3.1	设计要求	80
4.3.2	硬件电路设计	80
4.3.3	硬件原理图及说明	81
4.3.4	软件程序设计流程	83
4.3.5	程序代码与注释	83
4.3.6	程序运行追踪	90
4.3.7	实例注意事项总结	91
第 5 章	D/A 转换	92
5.1	DAC 转换器介绍	92



5.1.1	DAC 转换器功能描述	92
5.1.2	双 DAC 通道转换	97
5.2	DAC 相关寄存器功能说明	100
5.2.1	DAC 控制寄存器 (DAC_CR)	100
5.2.2	DAC 软件触发寄存器 (DAC_SWTRIGR)	103
5.2.3	DAC 通道 1 的 12 位右对齐数据保持寄存器 (DAC_DHR12R1)	103
5.2.4	DAC 通道 1 的 12 位左对齐数据保持寄存器 (DAC_DHR12L1)	103
5.2.5	DAC 通道 1 的 8 位右对齐数据保持寄存器 (DAC_DHR8R1)	104
5.2.6	DAC 通道 2 的 12 位右对齐数据保持寄存器 (DAC_DHR12R2)	104
5.2.7	DAC 通道 2 的 12 位左对齐数据保持寄存器 (DAC_DHR12L2)	105
5.2.8	DAC 通道 2 的 8 位右对齐数据保持寄存器 (DAC_DHR8R2)	105
5.2.9	双 DAC 的 12 位右对齐数据保持寄存器 (DAC_DHR12RD)	105
5.2.10	双 DAC 的 12 位左对齐数据保持寄存器 (DAC_DHR12LD)	106
5.2.11	双 DAC 的 8 位右对齐数据保持寄存器 (DAC_DHR8RD)	107
5.2.12	DAC 通道 1 数据输出寄存器 (DAC_DOR1)	107
5.2.13	DAC 通道 2 数据输出寄存器 (DAC_DOR2)	107
5.3	DAC 应用实例——产生三角波电压并输出	108
5.3.1	硬件电路设计	108
5.3.2	软件程序设计流程	108
5.3.3	程序代码设计	109
5.3.4	程序运行监测	112
5.3.5	实例注意事项总结	112
<b>第 6 章</b>	<b>LCD/触摸屏接口</b>	<b>113</b>
6.1	LCD 概述	113
6.1.1	LCD 工作原理	113
6.1.2	LCD 分类	113
6.1.3	如何显示	114
6.1.4	液晶显示模块的控制	115
6.2	触摸屏概述	115
6.2.1	电阻触摸屏工作原理	115
6.2.2	触摸屏电路控制	116
6.3	应用实例——LCD/触摸屏显示	116
6.3.1	设计要求	116
6.3.2	硬件电路设计	117
6.3.3	软件程序设计流程	124
6.3.4	程序代码与注释	124
6.3.5	程序运行追踪	132
6.3.6	实例注意事项总结	132

第 7 章 数码管显示接口应用	133
7.1 数码管概述	133
7.1.1 数码管工作原理	133
7.1.2 如何驱动数码管	134
7.2 应用实例——7 段数码管驱动	135
7.2.1 设计要求	135
7.2.2 硬件电路设计	135
7.2.3 数码管驱动程序流程图分析	137
7.2.4 数码管驱动应用程序源代码设计	137
7.2.5 程序运行实时监测	143
7.2.6 实例注意事项总结	143
第 8 章 SPI 接口	144
8.1 串行外设 (SPI) 接口介绍	144
8.1.1 串行外设接口功能描述	144
8.1.2 配置 SPI 为从模式	147
8.1.3 配置 SPI 为主模式	148
8.1.4 配置 SPI 为单工通信	148
8.1.5 数据发送与接收过程	149
8.1.6 CRC 计算	152
8.1.7 状态标志	153
8.1.8 关闭 SPI	153
8.1.9 利用 DMA 的 SPI 通信	154
8.1.10 错误标志	154
8.1.11 SPI 中断	155
8.2 微处理器 SPI 接口相关寄存器功能说明	155
8.2.1 SPI 控制寄存器 1 (SPI_CR1)	155
8.2.2 SPI 控制寄存器 2 (SPI_CR2)	157
8.2.3 SPI 状态寄存器 (SPI_SR)	157
8.2.4 SPI 数据寄存器 (SPI_DR)	158
8.2.5 SPI CRC 多项式寄存器 (SPI_CRCPR)	158
8.2.6 SPI Rx CRC 寄存器 (SPI_RXCR)	159
8.2.7 SPI Tx CRC 寄存器 (SPI_TXCR)	159
8.3 应用实例——利用 SPI 接口访问串行 Flash	160
8.3.1 设计要求	160
8.3.2 硬件电路设计	160
8.3.3 硬件电路原理图说明	167
8.3.4 应用程序流程图及时序分析	167
8.3.5 应用程序代码设计	168

8.3.6	程序运行实时监测	171
8.3.7	实例注意事项总结	171
<b>第 9 章</b>	<b>I<sup>2</sup>C 接口</b>	<b>172</b>
9.1	I <sup>2</sup> C 接口简介	172
9.1.1	I <sup>2</sup> C 接口功能描述	173
9.1.2	I <sup>2</sup> C 中断请求	181
9.1.3	SMBus	182
9.1.4	I <sup>2</sup> C 调试模式	182
9.2	微处理器 I <sup>2</sup> C 接口相关寄存器功能	183
9.2.1	I <sup>2</sup> C 控制寄存器 1 (I2C_CR1)	183
9.2.2	I <sup>2</sup> C 控制寄存器 2 (I2C_CR2)	184
9.2.3	I <sup>2</sup> C 自身地址寄存器 1 (I2C_OAR1)	185
9.2.4	I <sup>2</sup> C 自身地址寄存器 2 (I2C_OAR2)	186
9.2.5	I <sup>2</sup> C 数据寄存器 (I2C_DR)	186
9.2.6	I <sup>2</sup> C 状态寄存器 1 (I2C_SR1)	187
9.2.7	I <sup>2</sup> C 状态寄存器 2 (I2C_SR2)	189
9.2.8	I <sup>2</sup> C 时钟控制寄存器 (I2C_CCR)	190
9.2.9	I <sup>2</sup> C TRISE 寄存器 (I2C_TRISE)	190
9.3	应用实例——实现温度传感器读写	191
9.3.1	设计要求	191
9.3.2	硬件电路设计	191
9.3.3	温度传感器硬件原理图说明	195
9.3.4	应用程序流程图分析	196
9.3.5	应用程序代码设计	196
9.3.6	程序运行实时追踪	199
9.3.7	实例注意事项总结	199
<b>第 10 章</b>	<b>全速 USB 接口</b>	<b>200</b>
10.1	USB 接口简介	200
10.1.1	USB 功能描述	200
10.1.2	USB 功能模块的组成	201
10.1.3	USB 接口编程中需要考虑的问题	202
10.2	USB 接口相关寄存器描述符说明	209
10.2.1	通用寄存器	209
10.2.2	端点寄存器	213
10.2.3	缓冲区描述表	216
10.3	应用实例——USB 音频设备设计	218
10.3.1	设计要求	218

10.3.2	硬件电路设计	218
10.3.3	硬件电路原理图说明	218
10.3.4	USB 音频设备编程介绍	220
10.3.5	应用程序流程图分析	223
10.3.6	应用程序代码设计	223
10.3.7	程序运行实时追踪	233
10.3.8	实例注意事项总结	233
<b>第 11 章</b>	<b>以太网控制器模块设计</b>	<b>234</b>
11.1	以太网接口介绍	234
11.1.1	以太网模块主要功能	234
11.1.2	以太网模块引脚和内部信号	236
11.1.3	以太网模块功能描述: SMI、MII 和 RMII	237
11.2	以太网寄存器描述	242
11.2.1	MAC 寄存器描述	242
11.2.2	MMC 寄存器描述	254
11.2.3	IEEE 1588 时间戳寄存器描述	259
11.2.4	DMA 寄存器描述	263
11.3	应用实例——实现与收发器 DP83848C 的以太网通信	274
11.3.1	设计要求	274
11.3.2	硬件电路设计	274
11.3.3	硬件电路原理设计	280
11.3.4	应用程序流程图分析	282
11.3.5	应用程序代码设计	283
11.3.7	程序运行实时追踪	290
11.3.8	实例注意事项总结	291
<b>第 12 章</b>	<b>NOR Flash 应用</b>	<b>292</b>
12.1	FSMC 接口功能描述	292
12.1.1	外部设备地址映像	293
12.1.2	NOR 和 PSRAM 地址映像	294
12.1.3	NOR 闪存和 PSRAM 的非对齐访问支持	294
12.1.4	NAND 和 PC 卡地址映像	294
12.2	AHB 总线接口简介	295
12.2.1	支持的存储器和操作	295
12.2.2	配置寄存器	296
12.3	NOR 闪存和 PSRAM 控制器	296
12.3.1	外部存储器接口信号	296
12.3.2	NOR 闪存和 PSRAM 控制器支持的存储器类型及其操作方式	297

12.3.3	NOR 闪存和 PSRAM 控制器时序图	298
12.4	NOR 闪存和 PSRAM 控制器寄存器功能	305
12.4.1	SRAM/NOR 闪存片选控制寄存器 1~4 (FSMC_BCR1~4)	305
12.4.2	SRAM/NOR 闪存片选时序寄存器 1~4 (FSMC_BTR1~4)	307
12.4.3	SRAM/NOR 闪存写时序寄存器 1~4 (FSMC_BWTR1~4)	308
12.5	应用实例——实现存储器 M29W128FL 读/写	310
12.5.1	硬件设计	310
12.5.2	硬件电路原理图说明	321
12.5.3	应用程序流程图	321
12.5.4	NOR 闪存应用程序代码设计	322
12.5.5	调试程序及程序运行实时追踪	326
12.5.6	实例注意事项总结	326
<b>第 13 章</b>	<b>SD/MMC 应用</b>	<b>327</b>
13.1	SDIO 接口概述	327
13.1.1	SDIO 的主要功能	327
13.1.2	SDIO 总线拓扑	327
13.2	SDIO 接口功能描述	328
13.2.1	SDIO 适配器	329
13.2.2	SDIO AHB 接口	335
13.3	SDIO 接口卡介绍	337
13.3.1	卡状态寄存器	337
13.3.2	SD 状态寄存器	339
13.3.3	SD 的 I/O 功能模式	341
13.3.4	命令集简介	342
13.3.5	命令响应格式简介	344
13.4	SDIO 接口寄存器功能描述	347
13.4.1	SDIO 电源控制寄存器 (SDIO_POWER)	347
13.4.2	SDIO 时钟控制寄存器 (SDIO_CLKCR)	348
13.4.3	SDIO 参数寄存器 (SDIO_ARG)	348
13.4.4	SDIO 命令寄存器 (SDIO_CMD)	349
13.4.5	SDIO 命令响应寄存器 (SDIO_RESPCMD)	350
13.4.6	SDIO 响应 1..4 寄存器 (SDIO_RESPx)	350
13.4.7	SDIO 数据定时器寄存器 (SDIO_DTIMER)	351
13.4.8	SDIO 数据长度寄存器 (SDIO_DLEN)	351
13.4.9	SDIO 数据控制寄存器 (SDIO_DCTRL)	352
13.4.10	SDIO 数据计数器寄存器 (SDIO_DCOUNT)	353
13.4.11	SDIO 状态寄存器 (SDIO_STA)	353
13.4.12	SDIO 清除中断寄存器 (SDIO_ICR)	354

13.4.13	SDIO 中断屏蔽寄存器 (SDIO_MASK)	356
13.4.14	SDIO FIFO 计数器寄存器 (SDIO_FIFOCNT)	357
13.4.15	SDIO FIFO 数据寄存器 (SDIO_FIFO)	358
13.5	应用实例——实现 SD/MMC 存储卡的读/写	358
13.5.1	设计要求	358
13.5.2	硬件电路设计	359
13.5.3	硬件电路原理图说明	360
13.5.4	应用程序流程图分析	360
13.5.5	应用程序代码设计	361
13.5.6	程序运行实时追踪	367
13.5.7	实例注意事项总结	367
<b>第 14 章</b>	<b>无线数据传输综合应用项目</b>	<b>368</b>
14.1	嵌入式图形用户界面 $\mu$ C/GUI 移植	368
14.2	无线数据传输应用实例	370
14.2.1	设计要求	370
14.2.2	主要器件介绍	371
14.2.3	芯片 nRF24L01 增强型 ShockBurst 模式配置与编程	379
14.2.4	硬件电路设计	379
14.2.5	应用程序设计流程图	380
14.2.6	程序代码设计	381
14.3	实例注意事项总结	390
<b>第 15 章</b>	<b>永磁同步电机控制综合应用项目</b>	<b>392</b>
15.1	永磁同步电机概述	392
15.1.1	永磁同步电机的工作原理	392
15.1.2	电机伺服系统的基本结构	393
15.1.3	功率驱动控制单元	394
15.1.4	磁场定向控制 (FOC)	394
15.2	永磁同步电机控制应用实例设计	397
15.2.1	设计要求	398
15.2.2	硬件电路设计	399
15.2.3	应用程序设计流程图	403
15.2.4	程序代码设计	404
15.3	程序运行实时追踪	418
15.4	实例注意事项总结	418
<b>第 16 章</b>	<b>车载 GPS 移动导航应用项目</b>	<b>419</b>
16.1	GPS 系统概述	419
16.1.1	GPS 系统工作原理	420

16.1.2	GPS 数据格式 .....	420
16.2	车载 GPS 移动导航应用实例 .....	424
16.2.1	设计要求 .....	424
16.2.2	硬件电路设计 .....	425
16.2.3	应用程序设计流程图 .....	425
16.2.4	软件代码设计 .....	425
16.3	实例注意事项总结 .....	446
<b>第 17 章</b>	<b>RFID+GPRS 基于物联网技术的综合应用 .....</b>	<b>447</b>
17.1	$\mu$ C/OS-II 概述 .....	447
17.1.1	$\mu$ C/OS-II 简介 .....	447
17.1.2	$\mu$ C/OS-II 的特点 .....	448
17.2	$\mu$ C/OS-II 内核介绍 .....	448
17.2.1	$\mu$ C/OS-II 内核结构 .....	449
17.2.2	任务管理 .....	456
17.2.3	时间管理 .....	457
17.2.4	任务之间的通信与同步 .....	458
17.2.5	内存管理 .....	459
17.3	$\mu$ C/OS-II 移植 .....	460
17.3.1	$\mu$ C/OS-II 文件体系结构 .....	460
17.3.2	移植 $\mu$ C/OS-II 满足的条件 .....	461
17.3.3	移植要点 .....	461
17.4	综合实例——RFID+GPRS 基于物联网技术的应用 .....	464
17.4.1	硬件电路设计 .....	464
17.4.2	$\mu$ C/OS-II 软件设计 .....	467
17.4.3	程序运行实例说明 .....	481
17.5	实例注意事项总结 .....	483

# 第 1 章 Cortex-M3 及 ST 微处理器概述

Cortex-M3 是 ARM 公司于 2004 年底推出的基于 ARMv7-M 架构的处理器，它是一个低功耗的处理器，具有门数少、中断延迟小、调试容易等特点。本章对 ARM Cortex-M3 处理器及 STM32F 系列微控制器进行简要介绍。

## 1.1 Cortex-M3 处理器特点

Cortex-M3 主要是为了在微控制器、汽车车身系统、工业控制系统和无线网络等对功耗和成本敏感的嵌入式应用领域实现高系统性能而设计的。归纳起来，Cortex-M3 处理器具有如下特点。

### 1.1.1 高性能

Cortex-M3 处理器使用了 ARMv7-M 体系结构，是一个可综合、高度可配置的处理器。Cortex-M3 处理器的核心是基于哈佛架构的 3 级流水线内核，该内核集成了分支预测、单周期乘法、硬件除法等众多功能强大的特性，使其在 Dhrystone benchmark 上具有出色的表现（1.25 DMIPS/MHz）。可提供 1.25DMIPS/MHz 的性能，在一个具有 32 个物理中断的标准处理器实现上（0.13 $\mu$ m Metro 50MHz），达到了突出的 0.06mW/MHz 能效比。

处理器可通过两种途径来提高它的性能：一种是“work hard”，也就是直接通过提高时钟频率来提高性能，这种情况以高功耗作为代价，并增加了设计的复杂性；另一种是“work smart”，在低时钟频率的情况下提高运算效率，使处理器可以凭借简单的低功耗设计来完成与第一种情况相同的功能。

### 1.1.2 采用 Thumb-2 指令集具有更高的效率和性能

Cortex-M3 内核实现了 Thumb-2 指令集——传统 Thumb 指令集的一个超集，既获得了传统 32 位代码的性能，又具有 16 位的高代码密度。

Thumb-2 指令集的到来告别了状态切换的旧时代，再也不需要花时间来切换 32 位 ARM 状态和 16 位的 Thumb 状态，从而大大提高了软件开发和代码维护的效率。

Thumb-2 指令集为编程带来了更多的灵活性，许多数据操作使用更短的代码，提高了代码密度。许多指令都是单周期的，包括乘法相关指令。从整体性能上看，Cortex-M3 比得过绝大多数其他架构的处理器。

取指操作都是按 32 位处理的，同一周期最多可以取出两条指令，留下更多的带宽给数据传输。

Thumb-2 指令集还引入了一个新的 If-Then 结构，意味着可以有多达 4 个后续指令进行条件执行。在位字段处理、硬件除法和 If-Then 指令的协助下，Thumb-2 指令集架构底层的



关键特性使 C 代码的执行变得更加自然。

Cortex-M3 的设计允许处理器高频运行，即使在相同频率下，Cortex-M3 的每指令周期数（CPI）更低，可以做更多工作。

在开发方面，Thumb-2 指令集自动优化了性能和代码密度，无须交互使用 ARM 代码和 Thumb 代码，简化了编译目标的长期维护和支持工作。因此，用户不但可以继续使用 C 代码，而且免去了建立预编译目标代码库的麻烦，使代码在更大程度上获得了重复利用。

### 1.1.3 先进的中断处理功能

Cortex-M3 处理器紧密结合一个可配置的嵌套向量中断控制器（NVIC），提供了工业领先的中断处理性能。嵌套向量中断控制器（NVIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。NVIC 是 Cortex-M3 处理器中一个完整组成的部分，它可以进行高度配置，为处理器提供出色的中断处理能力。在 NVIC 的标准执行中，它提供了一个非屏蔽中断（NMI）和 32 个通用物理中断，这些中断带有 8 级的抢占优先权。NVIC 可以通过综合选择配置为 1~240 个物理中断中的任何一个，并带有多达 256 个优先级，向量化的中断功能剧烈地缩短了中断延迟，不再需要软件去判断中断源，中断的嵌套也是在硬件水平上实现的，不需要软件代码来实现。并可通过“尾链中断机制”和“晚到中断机制”来优化中断响应。

NVIC 支持中断嵌套（压栈），允许通过提高中断的优先级对中断进行提前处理。它还支持中断的动态优先权重置。优先权级别可以在运行期间通过软件进行修改。会防止正在处理的中断被进一步激活，直到中断服务程序完成，所以在改变它们的优先级的同时，也避免了意外重新进入中断的风险。

典型情况下，从中断发生到进入服务可减少 70% 的周期数，这是通过寄存器硬件堆栈，以及退出和重启多寄存器 Load-Store 操作完成的，既加速了中断的响应，也不再需要任何汇编代码来完成寄存器数据传送，大大简化了代码。

### 1.1.4 总线矩阵和接口的应用

Cortex-M3 处理器集成了一个 AMBA AHB-Lite 总线来连接系统外设，并降低系统集成的复杂性。总线矩阵支持不对齐的数据访问，使不同的数据类型可以在存储器中紧密衔接，显著降低 SRAM 的需求和系统成本。

Cortex-M3 处理器总线矩阵把处理器和调试接口连接到外部总线；也就是把基于 32 位 AMBA AHB-Lite 的 ICode、DCode 和系统接口连接到基于 32 位 AMBA APB 的专用外设总线（Private Peripheral Bus，PPB）。它们分别是：

- 指令存储区总线（两条）。两条代码存储区总线负责对代码存储区的访问，分别是 ICode 总线和 DCode 总线，前者用于取指和取向量，后者用于查表、数据读写及调试访问等操作。它们按照最佳速度进行优化。
- 系统总线。系统接口总线分别获取和访问系统存储空间中的指令和数据，覆盖区域包括 SRAM、片上外设、片外 RAM、片外扩展设备，以及系统级存储区的部分空间。