



手把手教你学系列丛书

周兴华 刘海渊 编著
周兴华单片机培训中心 策划

手把手

教你学ARM Cortex-M0

——基于STM32F0x2系列



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



手把手教你学系列丛书

手把手教你学 ARM Cortex-M0

——基于 STM32F0x2 系列

周兴华 刘海渊 编著

周兴华单片机培训中心 策划

北京航空航天大学出版社

内 容 简 介

本书以 ST 公司的 STM32F0x2 系列 ARM 处理器为例,从零开始,手把手地教初学者学习 ARM 设计知识。在介绍 STM32F0x2 系列各单元基本特性的同时,使用入门难度低、程序较短且能立竿见影的初级实例,循序渐进地帮助初学者掌握 ARM 的设计知识,以实践为主,辅以理论。本书的实例均经作者实际测试并能在实验板上正常运行,实用性非常强,读者既可以直接用于产品,也可以进一步改良升级。同时本书贯彻“手把手教你学系列丛书”的教学方式。

本书可用作大学本科或专科、中高等职业技术学校、电视大学等的教学用书,也可作为 ARM 爱好者的入门自学用书。

图书在版编目(CIP)数据

手把手教你学 ARM Cortex-M0:基于 STM32F0x2 系列 /
周兴华, 刘海渊编著. -- 北京:北京航空航天大学出版社, 2017.1

ISBN 978-7-5124-2185-1

I. ①手… II. ①周… ②刘… III. ①微处理器—系
统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2016)第 150503 号

版权所有,侵权必究。

手把手教你学 ARM Cortex-M0——基于 STM32F0x2 系列

周兴华 刘海渊 编著

周兴华单片机培训中心 策划

责任编辑 孙兴芳

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

北京市同江印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:54 字数:1 151 千字

2017 年 1 月第 1 版 2017 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2185-1 定价:128.00 元

前言

借助于手机及其他掌上电子产品的普及与推广,32 位的微处理器 ARM 也迅速发展壮大。在世界范围内,ARM 处理器正在成为工程师设计移动产品的首选。

传统的复杂指令集计算机(Complex Instruction Set Computer,CISC)体系由于指令集庞大,指令长度不固定,指令执行周期有长有短,使指令译码和流水线的实现在硬件上非常复杂,给芯片的设计开发和成本的降低带来了很大困难。

ARM 处理器采用了当今处理器设计的主流技术精简指令集计算机(Reduced Instruction Set Computer,RISC)。RISC 技术把设计重点放在了如何使计算机的结构更加简单、合理以及提高运算速度上。RISC 结构通过优先选取使用频率最高的简单指令,避免复杂指令,将指令长度固定,减少指令格式和寻址方式种类,以控制逻辑为主,不用或少用微码控制等来达到上述目的。因此,RISC 结构处理器的指令较少、运行速度快、抗干扰能力强。

ARM 处理器目前以 Cortex 为前缀进行命名,而且每一个大的系列里又分为若干小的系列。

ARM Cortex 处理器采用全新的 ARMv7 架构,根据使用对象的不同,划分为以下 3 大系列:

- ① Cortex-A 系列:开放式操作系统的高性能处理器;
- ② Cortex-R 系列:实时应用的卓越性能;
- ③ Cortex-M 系列:成本敏感的微处理器应用。

在 Cortex-M 系列里,ARM 面向微处理器产业分别推出了 Cortex-M0/M3/M4 三款嵌入式处理器,大部分半导体厂商集中于 M3 内核的生产。Cortex-M4 是 M3 的升级版,相较于 M3 具备更高的信号处理能力;而 M0 则是 M3 的精简版,它以低价格(与 8 位单片机相当)进入市场,但是其超高的性能(运行速率近 5 000 万次/秒)是 8 位单片机无法企及的,因此,ARM 终结 8 位单片机成为就在眼前的事情。

ARM 处理器的应用领域很广,由于其运行快、性能强、功耗低,以前 8051 单片机不能胜任的许多领域它均可一展身手,包括工业控制领域、无线通信领域、网络应用、消费类电子产品、成像和安全产品、移动互联网、3G 领域、科研及军事等。目前,ARM 微处理器占有 32 位 RISC 微处理器约 75% 以上的市场份额。

本书以 ST 公司的 STM32F0x2 系列 ARM 处理器为例,从零开始,手把手地教初学者学习 ARM 设计知识。在介绍 STM32F0x2 各单元基本特性的同时,使用入

门难度低、程序较短且能立竿见影的初级实例,循序渐进地帮助初学者掌握 ARM 的设计知识,以实践为主,辅以理论。本书的实例均经作者实际测试并能在实验板上正常运行,实用性非常强,读者既可以直接用于产品,也可以进一步改良升级。需要说明的是,本书的所有例程都可以不做修改或略做修改而直接用于 ST 公司的其他 ARM 芯片上,因此其通用性非常强。

另外,目前 ARM 芯片及开发套件的价格已经降到非常低的水平,并且开发软件的界面也非常友好,因此,学习 ARM 的时代已经到来。

本书的编写工作得到了北京航空航天大学出版社相关领导的大力支持,在此表示衷心感谢!

本书在编写过程中参考了相关书籍及部分网络流通资料,在此一并致谢!

限于作者水平,书中必定存在不少缺点或漏洞,诚挚欢迎广大读者提出意见并不吝赐教。

如果读者需要书中介绍的学习器材或想参加 ARM 的设计培训班,可与作者联系,联系方式如下:

地址:上海市闵行区莲花路 2151 弄 57 号 201 室

邮编:201103

联系人:周兴华

电话(传真):(021)64654216,13774280345

技术支持 E-mail:zxh2151@sohu.com,zxh2151@163.com

培训中心主页:<http://www.hlelectron.com>

周兴华

2016 年 2 月



基础篇

第 1 章 概 述	3
1.1 采用 C 语言编程	3
1.2 C 语言突出的优点	4
1.3 寄存器操作与库函数操作	6
1.4 ARM 嵌入式处理器的开发环境	7
第 2 章 ARM 发展简介	9
2.1 ARM 是什么	9
2.2 嵌入式处理器 RISC 技术简介	9
2.3 ARM 处理器的发展	10
2.4 ARM 处理器的应用	14
2.5 ARM 处理器的优点	15
2.6 ARM 的优势	15
2.7 ARM 未来发展展望	16
第 3 章 ARM Cortex-M0 处理器内核架构体系	18
3.1 STM32F072 的特性和结构	19
3.2 STM32F072 存储器和外设寄存器边界映射	20
3.3 STM32F072 系统配置	25
3.4 STM32F072 中断控制	30
3.5 STM32F072 引脚封装	32
第 4 章 开发/实验工具介绍及第一个 STM32F072 入门程序	37
4.1 RealView MDK 5.15 开发环境及厂商软件包安装	37
4.2 CMSIS 简介	38
4.3 STM32F0x2 实验工具	41
4.4 STM32F0x2 系列开发过程的文件管理及项目设置	45
4.5 STM32F0x2 开发流程	52
4.6 第一个 STM32F072 入门程序	52

第 5 章 C 语言基础知识	59
5.1 C 语言的标识符与关键字	59
5.2 数据类型	60
5.3 常量、变量及存储方式	61
5.4 数 组	62
5.5 C 语言的运算	65
5.6 流程控制	72
5.7 函 数	77
5.8 指 针	80
5.9 结构体	84
5.10 共用体	90
5.11 枚 举	92
5.12 Keil RealView MDK 在 ARM C 语言开发中的常用方法	93
5.13 中断函数	94

入门篇

第 6 章 STM32F0x2 复位和系统时钟	99
6.1 复 位	99
6.2 时 钟	100
6.3 低功耗模式	106
6.4 RCC 库函数	107
6.5 配置系统时钟频率	117
第 7 章 STM32F0x2 通用 I/O 的特性及应用	120
7.1 通用 I/O 的特点	120
7.2 GPIO 库函数	127
7.3 STM32F072 的 GPIO 输出实验——控制发光二极管闪烁	129
7.4 软件延时较准确的 GPIO 输出实验——控制发光二极管闪烁	131
7.5 STM32F072 的 GPIO 输入/输出实验——按键控制发光二极管闪烁	133
第 8 章 中断/事件及应用设计	139
8.1 嵌套向量中断控制器的特点	139
8.2 外部中断/事件控制器	141
8.3 外部和内部中断/事件线路映像	143
8.4 MISC 库函数及 EXTI 库函数	144

8.5	STM32F072 的外中断实验——控制发光二极管亮/灭	145
8.6	STM32F072 的系统节拍定时器中断实验——控制发光二极管精确亮/灭	150
第 9 章	TFT-LCD 彩色液晶显示器的驱动显示	154
9.1	TFT-LCD 彩色液晶显示器	154
9.2	TFT-LCD 彩色液晶显示器模块的引脚功能	155
9.3	ILI9325/ILI9328 几个重要的控制寄存器及控制命令	156
9.4	TFT-LCD 彩色液晶显示器显示的相关设置步骤	160
9.5	STM32F072 的 TFT-LCD 驱动实验——显示多种颜色及图形	161
第 10 章	SPI 总线特性及 W25Q16 SPI Flash 存储器驱动	181
10.1	SPI 的主要特点	181
10.2	SPI 功能描述	182
10.3	SPI 中断	189
10.4	SPI 库函数	190
10.5	W25Q16 SPI Flash 存储器	198
10.6	W25Q 系列存储器的特点	199
10.7	W25Q 系列存储器的引脚封装及配置	201
10.8	W25Q 系列存储器的引脚功能	202
10.9	W25Q 系列存储器的控制/状态寄存器	203
10.10	W25Q 系列存储器的状态寄存器存储保护模块	204
10.11	W25Q 系列存储器的操作指令	206
10.12	中英文显示的原理	217
10.13	编写生成 CHNGBK_MAKE.hex 应用程序的源代码	218
10.14	中文字库的下载	220
10.15	STM32F072 的 TFT-LCD 驱动实验——显示多种颜色、图形及中英 文字符	223
第 11 章	通用同步异步串行收发器的特性及应用	239
11.1	USART 简介	239
11.2	USART 中断	246
11.3	USART 库函数	247
11.4	STM32F072 的串口通信实验——与 PC 实现通信	249
第 12 章	RTC 实时时钟的特性及应用	254
12.1	RTC 模块的主要特性	254
12.2	RTC 初始化及配置	256
12.3	RTC 中断	256
12.4	RTC 库函数	257

12.5	STM32F072 的实时时钟实验——获取当前时间	261
第 13 章	定时器与计数器的特性及应用	272
13.1	高级控制定时器 TIM1	272
13.2	通用定时器 TIM2/TIM3	274
13.3	通用定时器 TIM14	275
13.4	通用定时器 TIM15/TIM16/TIM17	276
13.5	基本定时器 TIM6/TIM7	279
13.6	TIM 库函数	279
13.7	STM32F072 定时器的定时中断实验——LED1 每 500 ms 闪烁一次	285
13.8	STM32F072 定时器 1 的输入捕获实验	288
13.9	STM32F072 定时器 3 的比较匹配中断实验	291
13.10	STM32F072 定时器 1 的 PWM 输出实验	294
13.11	红外遥控信号接收解调实验	298
第 14 章	数/模转换器的特性及应用	306
14.1	DAC 的特点	306
14.2	DAC 功能设置	307
14.3	DAC 库函数	310
14.4	STM32F072 的 DAC 输出实验	311
第 15 章	模/数转换器的特性及应用	318
15.1	ADC 的主要特性	318
15.2	ADC 的功能及设置	320
15.3	转换的外部触发和触发极性	324
15.4	数据对齐	324
15.5	温度传感器	325
15.6	电池电压监测	326
15.7	ADC 中断	326
15.8	ADC 库函数	326
15.9	STM32F072 的 ADC 转换实验	329
第 16 章	DMA 控制器的特性及应用	333
16.1	DMA 的主要特性	333
16.2	DMA 的功能	334
16.3	DMA 库函数	340
16.4	STM32F072 的 ADC 转换 DMA 数据传送实验	342
第 17 章	I²C 总线接口的特性及应用	348
17.1	I ² C 的主要特性	348

17.2	I ² C 功能描述	349
17.3	I ² C 库函数	359
17.4	STM32F072 的 I ² C 通信实验——读/写 AT24C02	362
第 18 章	比较器的特性及应用	380
18.1	比较器的主要特性	380
18.2	比较中断	381
18.3	COMP 库函数	381
18.4	STM32F072 的模拟比较器实验	383
第 19 章	bxCAN 的特性及应用	391
19.1	bxCAN 的主要特性	391
19.2	bxCAN 工作模式及网络拓扑	392
19.3	bxCAN 功能描述	395
19.4	bxCAN 中断	403
19.5	bxCAN 库函数	404
19.6	STM32F072 的 CAN 通信实验	406
第 20 章	看门狗定时器的特性及应用	412
20.1	独立看门狗	412
20.2	窗口看门狗	414
20.3	IWDG 库函数	417
20.4	STM32F072 的独立看门狗实验	418

提高篇

第 21 章	电阻式触摸屏的原理及设计	425
21.1	低电压输入/输出触摸屏控制器 ADS7846 简介	425
21.2	ADS7846 的工作原理	427
21.3	ADS7846 的控制字	429
21.4	笔中断接触输出	431
21.5	STM32F072 的触摸屏测试实验	431
第 22 章	2.4G 无线收发模块 NRF24L01 的特性及应用	444
22.1	NRF24L01 的主要特性	444
22.2	NRF24L01 的结构及引脚功能	444
22.3	NRF24L01 的工作模式	446
22.4	NRF24L01 的工作原理	446
22.5	配置字	447
22.6	STM32F072 的 NRF24L01 通信实验	447

第 23 章	FatFS 文件系统及电子书实验	463
23.1	FatFS 文件系统的特性	464
23.2	FatFS 文件系统分析	464
23.3	FatFS 文件系统移植	466
23.4	SD 卡的初始化及文件系统实验	470
23.5	电子书实验	496
第 24 章	数码相框设计显示及 GUI 实验	514
24.1	简易数码相框的构成和图像文件的处理	514
24.2	数码相框设计显示实验	515
24.3	GUI	518
24.4	GUI 设计实验	519
第 25 章	RTX Kernel 实时操作系统	542
25.1	RTX Kernel 实时操作系统概述	542
25.2	RTX Kernel 实时操作系统的特性	543
25.3	RTX Kernel 实时操作系统的基本功能及进程间的通信	544
25.4	RTX Kernel 实时操作系统的任务管理	545
25.5	RTX Kernel 实时操作系统的库函数	548
25.6	RealView MDK 开发环境自带的 RTX Kernel 例程分析	565
第 26 章	RTX Kernel 的延时及事件设计实验	579
26.1	时间间隔延迟实验	579
26.2	信号标志的发送/接收实验 1	582
26.3	信号标志的发送/接收实验 2	588
26.4	外部中断的信号标志发送/接收实验	592
第 27 章	RTX Kernel 内存池及邮箱的设计实验	599
27.1	内存池及邮箱的实验 1	599
27.2	内存池及邮箱的实验 2	604
第 28 章	RTX Kernel 的互斥设计实验	609
第 29 章	RTX Kernel 信号量的传送与接收设计实验	613
第 30 章	RTX Kernel 综合设计实验	617
30.1	文件系统实验	617
30.2	手写画板实验	621
30.3	数码相框实验	625
30.4	用户定时器实验	628
30.5	循环定时器实验	633
30.6	综合设计实验	636

第 31 章	μCOS-II 实时操作系统	642
31.1	μ COS-II 实时操作系统概述	642
31.2	μ COS-II 实时操作系统的特点	643
31.3	μ COS-II 实时操作系统的组成	644
31.4	μ COS-II 实时操作系统的时间管理	645
31.5	μ COS-II 实时操作系统的内存管理	645
31.6	μ COS-II 实时操作系统通信同步	645
31.7	μ COS-II 实时操作系统的任务管理及调度	646
31.8	μ COS-II 内核介绍	647
31.9	μ COS-II 实时操作系统的 API 函数	650
第 32 章	μCOS-II 实时操作系统入门及移植	668
32.1	下载 μ COS-II 源代码	668
32.2	文件管理及工程管理	668
32.3	配置 μ COS-II	670
32.4	创建任务	673
32.5	创建 main 函数	674
32.6	编译及应用	676
第 33 章	μCOS-II 事件标志组设计实验	677
33.1	事件标志组	677
33.2	手动测试仪设计实验	678
33.3	自动测试仪设计实验	686
33.4	中断发送事件标志实验	692
第 34 章	μCOS-II 消息邮箱设计实验	697
34.1	消息邮箱	697
34.2	消息邮箱设计实验	698
第 35 章	μCOS-II 动态内存分配设计实验	703
第 36 章	μCOS-II 消息队列设计实验	708
36.1	消息队列	708
36.2	消息队列设计实验	709
第 37 章	μCOS-II 互斥量设计实验	714
37.1	互斥信号量	714
37.2	互斥量设计实验	715
第 38 章	μCOS-II 信号量设计实验	720
38.1	信号量	720
38.2	信号量设计实验	721

第 39 章 μ COS-II 应用设计实验	726
39.1 手写画板实验	726
39.2 数码相框实验	731
39.3 用户定时器实验	735
39.4 循环定时器实验	741
39.5 综合设计实验	745

应用篇

第 40 章 使用 DS18B20 测量温度及使用 DHT11 测量温湿度	755
40.1 单线数字温度传感器 DS18B20	755
40.2 DS18B20 测温实验	762
40.3 DHT11 数字温湿度传感器	769
40.4 DHT11 湿度温度测试实验	772
第 41 章 RS-485 通信组网设计	778
41.1 RS-485 通信的特点	778
41.2 RS-485 通信使用的电缆及布网	779
41.3 RS-485 分布式数据采集和控制网络原理	779
41.4 RS-485 通信网简单实验	780
第 42 章 NRF24L01 无线通信组网设计	797
42.1 NRF24L01 的主要特性及应用领域	797
42.2 NRF24L01 的结构及引脚功能	798
42.3 NRF24L01 工作模式	800
42.4 NRF24L01 工作原理	800
42.5 NRF24L01 配置字	801
42.6 NRF24L01 的寄存器操作命令	802
42.7 NRF24L01 的 C51 驱动程序介绍	803
42.8 NRF24L01 无线通信组网实验	807
第 43 章 CAN 通信组网设计	820
43.1 CAN 通信简介	821
43.2 CAN 通信的特点	823
43.3 CAN 技术简介	824
43.4 CAN 的可靠性	835
43.5 应用举例	836
43.6 CAN 通信组网实验	836
参考文献	851

基础篇



第 1 章

概 述

自从许多读者跟着“手把手教你学系列丛书”学习单片机设计应用技术后,均已取得丰硕的成果。读者利用单片机研制了各种各样的智能控制装置,在生产实践中发挥了重要的作用,有的还做成产品投放市场,创造了很好的经济效益。

有很多读者给作者来信来电,表示“手把手教你学系列丛书”的教学方式很适合他们学习,跟着“手把手教你学系列丛书”学习实践后,渐渐地从不理解到了解、从不懂到学会单片机的设计,因此,他们是非常喜欢“手把手教你学系列丛书”的。

时光如梭、岁月如流,目前科学技术正以一日千里的速度发展着,转眼之间,嵌入式 CPU 就转向了以 ARM 系列为主流市场的应用,但 ARM 芯片的架构比起以前我们熟悉的 51、PIC、AVR 等 8 位单片机要复杂得多,其寄存器、中断功能等比起 8 位机要多好几倍,这就有了寄存器操作与库函数操作等不同的设计方法。对初学者而言,本来基础就差,现在更是陷入云里雾里,更难学好 ARM 设计了。

针对初学者的困惑,作者采用与“手把手教你学系列丛书”相同的教学风格,以实践(实验)为主线,以由浅入深的实例为灵魂,循序渐进地引导读者进行学习、实验,这样初学者就可以学得进、记得牢,不会产生畏难情绪,无形之中就会一步一步地掌握 ARM 芯片的设计。作者编写此书的目的就是希望读者能够轻松、快速、容易地学会 ARM 的设计。

1.1 采用 C 语言编程

为了提高编制计算机系统和应用程序的效率,改善程序的可读性和可移植性,最好的办法就是采用高级语言编程。目前,C 语言逐渐成为国内外开发单片机的主流语言。

C 语言是一种通用的编译型结构化计算机程序设计语言,在国际上十分流行。它兼顾了多种高级语言的特点,并具备汇编语言的功能;它支持当前程序设计中广泛

采用的由顶向下的结构化程序设计技术。一般的高级语言难以实现汇编语言对于计算机硬件直接进行操作(如对内存地址的操作、移位操作等)的功能,而 C 语言既具有一般高级语言的特点,又能直接对计算机的硬件进行操作。C 语言有功能丰富的库函数,而且运算速度快、编译效率高,再者采用 C 语言编写的程序能够很容易地在不同类型的计算机之间进行移植。因此,C 语言的应用范围越来越广泛。

用 C 语言来编写目标系统软件,会大大缩短开发周期,且明显地增强软件的可读性,便于改进和扩充,进而研制出规模更大、性能更完善的系统。

因此,用 C 语言进行 ARM 程序设计是嵌入式处理器开发与应用的必然趋势。对汇编语言掌握到只要可以读懂程序,在时间要求比较严格的模块中能够进行程序的优化即可。采用 C 语言进行设计也不必对 ARM 芯片和硬件接口的结构有很深入的了解,编译器可以自动完成变量存储单元的分配,所以编程者可以专注于应用软件部分的设计,大大加快了软件的开发速度。采用 C 语言可以很容易地进行处理器的程序移植工作,有利于产品中的处理器重新选型。

C 语言模块化程序结构的特点是,可以使程序模块共享,且不断丰富。C 语言可读性强的特点,更容易使大家借鉴前人的开发经验来提高自己的软件设计水平。采用 C 语言,可针对处理器常用的接口芯片编制通用的驱动函数,可针对常用的功能模块、算法等编制相应的函数,这些函数经过归纳整理可形成专家库函数,供广大的工程技术人员和单片机爱好者使用并完善,这样可以大大提高国内单片机软件的设计水平。

过去长时间困扰人们的“高级语言产生代码太长,运行速度太慢,不适合嵌入式处理器使用”的致命缺点已被大幅度克服。目前,用于嵌入式处理器的 C 语言编译代码长度已超过中等程序员的水平。而且,一些先进的新型 ARM 处理器芯片(例如 STM32F4 系列)的片上 SRAM、Flash 空间都很大,运行速度很快(主频最高达 168 MHz,相当于低端的 DSP 处理能力),代码效率所差的 10%~20%已经不是什么问题。关于速度优化的问题,只要有好的仿真器的帮助,用人工优化关键代码就是很简单的事了。至于开发速度、软件质量、结构严谨和程序坚固等方面,C 语言的优势绝非是汇编语言所能比拟的。

1.2 C 语言突出的优点

1. 语言简洁,使用方便灵活

C 语言是现有程序设计语言中规模最小的语言之一,而小的语言体系往往能设计出较好的程序。C 语言的关键字很少,ANSI C 标准一共只有 32 个关键字、9 种控制语句,压缩了一切不必要的成分。C 语言的书写形式比较自由,表达方法简洁,使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。