



ARM Cortex-M0

微控制器 原理与实践

温子祺 刘志峰 沈文胜 曹快东 潘海燕 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

ARM Cortex-M0 微控制器 原理与实践

温子祺 刘志峰 冼安胜 林秩谦 潘海燕 编著

北京航空航天大学出版社

内 容 简 介

本书以新唐公司 ARM Cortex-M0 内核的 NuMicro M051 系列微控制器为蓝本,由浅入深,软硬结合,全面系统地介绍基于该微控制器的原理与结构、开发环境与工具、各种接口与功能单元应用的软件编写方法。本书以夯实基础,面向应用,理论与实践、方法与实现紧密结合为主线展开,根据 ARM Cortex-M0 的运行速度快、资源丰富、功能强大等显著特点,采用 C 语言作为系统软件的开发平台,由浅入深,以螺旋式上升的方式进行编排。在讲解原理和设计方法的同时,还穿插了作者相关的经验、技巧和注意事项,有很强的实用性和指导性。

本书可以作为高等院校电子、自动化、仪器仪表和计算机等相关专业的辅助教材,也可作为 ARM Cortex-M0 微控制器的培训教材,可供相关技术人员学习和参考。

图书在版编目(CIP)数据

ARM Cortex-M0 微控制器原理与实践 / 温子祺等编

著. — 北京:北京航空航天大学出版社,2013.1

ISBN 978-7-5124-1037-4

I. ①A… II. ①温… III. ①微控制器 IV.

①TP332.3

中国版本图书馆 CIP 数据核字(2012)第 295544 号

版权所有,侵权必究。

ARM Cortex-M0 微控制器原理与实践

温子祺 刘志峰 冼安胜 林秩谦 潘海燕 编著

责任编辑 刘 晨 刘朝霞

*

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@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:34.5 字数:735 千字

2013 年 1 月第 1 版 2013 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-1037-4 定价:79.00 元(含光盘 1 张)

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

嵌入式领域的发展日新月异,读者也许还没有注意到,但是如果停下来想一想 MCU 系统 10 年前的样子并与当今的 MCU 系统比较一下,会发现 PCB 设计、元件封装、集成度、时钟速度和内存大小已经经历了好几代的变化。在这方面最热门的话题之一是仍在使用 8 位 MCU 的用户何时才能摆脱传统架构,并转向使用更先进的 32 位微控制器架构,如基于 ARM Cortex-M 的 MCU 系列。在过去几年里,嵌入式开发者向 32 位 MCU 的迁移一直呈现强劲势头,采取这一行动的最强有力的理由是市场和消费者对嵌入式产品复杂性的需求大大增加。随着嵌入式产品彼此互联越来越多、功能越来越丰富,目前的 8 位和 16 位 MCU 已经无法满足处理要求,即使 8 位或 16 位 MCU 能够满足当前的项目需求,它也存在限制未来产品升级和代码重复使用的严重风险。第二个常见原因是嵌入式开发者开始认识到迁移到 32 位 MCU 带来的好处,且不说 32 位 MCU 能提供超过 10 倍的性能,单说这种迁移本身就能够带来更低的能耗、更小的程序代码、更快的软件开发时间以及更好的软件重用性。

随着近年来制造工艺的不断进步,ARM Cortex 微控制器的成本也不断降低,已经与 8 位和 16 位微控制器处于同等水平;另一个原因是基于 ARM 的器件的选择余地、性能范围和可用性。如今,越来越多的微控制器供应商提供基于 ARM 的微控制器,这些产品能提供选择范围更广的外设、性能、内存大小、封装、成本等。另外,基于 ARM Cortex-M 的微控制器还具有专门针对微控制器应用的一些特性,这些特性使 ARM 微控制器具有日益广泛的应用范围。与此同时,基于 ARM 的微控制器的价格在过去 5 年里已大幅降低,并且面向开发者的低成本甚至免费开发工具也越来越多。

与其他架构相比,选择基于 ARM 的微控制器也是更好的投资。现今,针对 ARM 微控制器开发的软件代码可在未来多年内供为数众多的微控制器供应商重复使用。随着 ARM 架构的应用更加广泛,聘请具有 ARM 架构行业经验的软件工程师也比聘请其他架构工程师更加容易,这也使得嵌入式开发者的产品和资产能够更加面向未来。

本书微控制器的选型以新唐公司 ARM Cortex-M0 内核的 NuMicro M051 系列微控制器为蓝本。

本书共分为 5 大篇。

前言

第1篇为初步认知篇,简略介绍微控制器的发展趋势,详细讲解 ARM 的由来,并初步了解 ARM 微控制器指令集和 C 语言编程基础。

第2篇为基础入门篇,着重讲解 NuMicro M051 系列微控制器的内部资源的基本使用,如 GPIO、定时器、外部中断、串口(含模拟串口)、看门狗、Flash 内存控制器、I²C 总线控制器、SPI 通信、模拟/数字转换等,同时对 74LS164 串行输入并行输出锁存器、数码管、LCD 进行简单介绍。基础入门篇做到原理与实践相结合的过程体系,初学者能够迅速掌握 NuMicro M051 系列微控制器的基本应用。最后阐述了 NuMicro M051 如何进行功耗控制、软件复位等应用和 Keil 内建的 RTX-Kernel 实时系统以及 LIB 的生成、调用,特别是 RTX-Kernel 实时系统的学习将对以后进军嵌入式实时系统提供了厚实的根基。

第3篇为深入篇,对接口编程、微控制器编程优化、微控制器稳定性作深入的研究,以深入接口和深入编程进行讲解,是技术上的重点,同样是技术上的难点。这样我们对微控制器的理解不再浮于表面,而是站在一名项目开发者的角度,思考着众多的技术性问题,譬如深入接口部分是以数据校验为重点,包含奇偶校验、校验和、CRC16 循环冗余检验,加深大家对数据校验的理解。深入编程以编程规范、代码架构、C 语言的高级应用(如宏、指针、强制转换、结构体等复杂应用)、程序防跑飞等要点作深入的研究。深入篇从技术角度来看是整本书内容的精华部分,在研究如何优化微控制器的性能、稳定性搞得焦头烂额的时候指引了明确的方向。深入篇是我们必看的部分,因其涉及的内容是微控制器与 C 编程的精髓,并为解决这多方面的问题,提供了不可多得的参考价值。

第4篇为番外篇,何谓番外篇,因为本篇超出了介绍微控制器的范畴,但是又不得不说,因为在高级实验篇很大部分的篇章已经涉及了界面的应用。说实话,现在的微控制器程序员或多或少与界面接触,甚至要懂得界面的基本编写,也就是说,微控制器程序员同时演绎着界面程序员的角色,这在中小型企业比较常见,编写的往往是一些比较简单的调试界面,常用于调试或演示给老板或参观的人看,当产品竣工时,要提供相应的 DLL 给系统集成部,缔造出不同的应用方案。在番外篇中,界面编程开发工具为 VC++2008,通过 VC++2008 向大家展示界面如何编写,同时如何实现串口通信、USB 通信、网络通信,只要使用笔者编写好的类,实现它们的通信是如此的简单,就像在 C 语言中调用函数一样,只需要掌握 Init()、Send()、Recv()、Close()函数的使用就可以了,相信大家会在这篇中基本掌握界面编程,最后驾轻就熟,编写出属于自己的调试工具。

第5篇为高级通信接口开发篇,阐述了 USB 与网络通信的原理及其应用。在我们进行产品研发的过程当中,不可避免地要接触各种各样的 USB 设备,并要为其编写程序。一旦当前的 USB 设备满足不了项目的要求时,往往使用网络设备取代 USB 设备,这个现象是十分常见的。其实很大一部分人如果是初始接触 USB 或者网络设备开发,他们就感觉到非常痛苦的事情,为什么这样说呢?因为要对 USB 或

者网络设备进行开发,必须要对 USB 或网络协议要熟悉。难能可贵的是本书在有限篇幅里简明扼要地对 USB 和网络的协议描述得一清二楚,并通过实验进行验证,以此消除他们对 USB 和网络编程的恐惧,从此对 USB 与网络设备的开发驾轻就熟。由于篇幅限制,第 5 篇关于 USB 和网络章节的相关内容见随书光盘。

本书以 SmartM-M051 开发板为实验平台(见实物图附录 A. 2),该开发板是为初学者设计的一款实用型的开发板,不仅含有基本的设备单元,同时在开发板的实用性的基础上能够搭载 USB 模块与网络模块,很好地满足了书中所有实验的要求。本人还编写了单片机多功能调试助手(详见附录 B),专为大家排忧解难,该软件不但能够实现串口、USB、网络调试、常用校验值计算、编码转换等功能。

天下大事,必作于细,无论是从微控制器入门与深入的角度出发,还是从实践性与技术性的角度出发,都是本书的亮点,可以说是作者用尽了心血进行编写,多年工作经验的积累,读者通过学习本书相当于继承了作者的思路与经验,找到了快捷径,能够花最少的时间获得最佳的学习效果,节省不必要的摸爬打滚的时间。

参与本书编写工作的主要人员有温子祺、刘志峰、冼安胜、林秩谦、潘海燕小姐负责本书的前期排版,最终方案的确定和本书的定稿全部由温子祺负责;其次还要感谢佛山市安讯智能科技有限公司工作的卢永坚、何超平、王雨杰、程国洪、张铭坤先生和美的公司工作的龙俊贤先生,他们对本书提出了不少建设性的建议;感谢温子龙、温子明、李祖达、陈春柳经理等对本人的支持;感谢北京航空航天大学出版社的胡晓柏主任,在从写书到出版的过程中提出了不少有价值的参考意见,让此书不断完善。

本书主要取材于实际的项目开发经验,对于微控制器编程的程序员说是一个很好的消息,本书例程编程规范良好,代码具有良好的移植性,移植到不同的平台十分方便。最后希望本书能对微控制器应用推广起到一定的作用,由于程序代码较复杂、图表比较多,难免会有纰漏,恳请读者批评指正,并且可以通过 E-mail:wenziqi@hotmail.com 进行反馈,并欢迎大家访问 www.smartmcu.com,我们希望能够得到读者的参与和帮助。

温子祺
2013 年 1 月

目 录

绪 论	1
0.1 什么是微控制器	1
0.2 微控制器历史	1
0.3 微控制器应用领域	2

第 1 篇 初步认知篇

第 1 章 微控制器发展趋势	7
1.1 概 述	7
1.2 ARM Cortex-M 微控制器优势	8
1.2.1 指令集效率	8
1.2.2 8 位应用程序的神话	10
1.2.3 性 能	11
1.2.4 8 位和 16 位微控制器的局限	11
1.2.5 低功耗	12
1.2.6 内存访问效率	12
1.2.7 通过降低操作频率来降低能耗	12
1.2.8 通过缩短活跃周期来降低能耗	13
1.2.9 低功耗的总体优势	13
1.2.10 软件开发	13
1.2.11 从 8 位或 16 位微控制器向 ARM 移植软件	13
1.2.12 调 试	14
1.2.13 选 择	15
1.2.14 软件可移植性	15
1.2.15 迁移成本	15

目 录

1.2.16 结 论	16
1.3 ARM Cortex-M 微控制器程序迁移	17
第 2 章 ARM 概述	28
2.1 ARM	28
2.2 RISC	30
2.2.1 简 介	30
2.2.2 概念分析	30
2.2.3 特 点	31
2.2.4 区 别	32
2.2.5 种 类	32
2.2.6 CPU 发展	33
2.2.7 CPU 的制造过程	34
第 3 章 ARM Cortex-M0	35
3.1 总线架构	35
3.1.1 什么是 AMBA	36
3.1.2 什么是 AHB-Lite	37
3.1.3 什么是 CoreSight	38
3.2 Cortex-M0 的结构特点	40
3.2.1 编程模型	40
3.2.2 存储模型	40
3.2.3 异常处理	41
3.2.4 功耗管理	42
3.2.5 指令集	43
3.3 开发工具	43
第 4 章 ARM 微控制器的指令集	46
4.1 ARM 微控制器的指令的分类与格式	46
4.2 ARM 指令的条件域	47
4.3 ARM 指令的寻址方式	48
4.4 ARM 指令集	50
4.4.1 跳转指令	51
4.4.2 数据处理指令	52
4.4.3 乘法指令与乘加指令	57
4.4.4 程序状态寄存器访问指令	60

4.4.5 加载/存储指令	61
4.4.6 批量数据加载/存储指令	63
4.4.7 数据交换指令	64
4.4.8 移位指令(操作)	65
4.4.9 协微控制器指令	66
4.4.10 异常产生指令	68
4.4.11 Thumb 指令及应用	69
第 5 章 ARM C 语言编程	70
5.1 C 语言简史	70
5.2 C 语言特点	71
5.2.1 优 点	72
5.2.2 缺 点	73
5.3 数据类型	73
5.3.1 基本数据类型	73
5.3.2 数据类型修饰符 signed 和 unsigned	73
5.4 常量和变量	74
5.4.1 常 量	74
5.4.2 变 量	74
5.5 操作符	76
5.5.1 算术操作符	76
5.5.2 关系操作符	76
5.5.3 逻辑操作符	76
5.5.4 位操作符	76
5.6 控制结构	78
5.6.1 选 择	78
5.6.2 循 环	78
5.7 结构体	78
5.8 编译指令	79
5.8.1 #define 和 #undef	79
5.8.2 #if 和 #endif	79
5.8.3 #error	79
5.9 标准 C 库的应用	79
5.9.1 标准 C 库的组成	79
5.9.2 标准 C 库的使用流程	80

第 2 篇 基础入门篇

第 6 章 NuMicro M051 系列微控制器	85
6.1 概 述	85
6.1.1 低门数微控制器特征	85
6.1.2 NVIC 特征	85
6.1.3 调 试	86
6.1.4 总线接口	86
6.2 系统管理器	86
6.2.1 系统复位	86
6.2.2 系统电源架构	87
6.3 系统存储映射	88
6.4 系统管理器控制寄存器映射	89
6.5 嵌套向量中断控制器(NVIC)	95
6.5.1 异常模式和系统中断映射	96
6.5.2 操作描述	100
第 7 章 平台搭建与下载工具	105
7.1 平台搭建	105
7.1.1 启动程序	105
7.1.2 创建工程	105
7.1.3 编译代码	108
7.1.4 安装 Nu-Link for Keil 驱动	109
7.1.5 设置 Nu-Link	115
7.1.6 下载代码	117
7.1.7 硬件仿真	119
7.2 ISP 下载	122
7.2.1 ISP 下载工具概述	122
7.2.2 ISP 下载步骤	122
7.3 ICP 下载	125
7.3.1 ICP 下载工具概述	125
7.3.2 ICP 下载步骤	126
7.4 JTAG 与串行调试(SWD)	129
7.4.1 JTAG 简介	129
7.4.2 SWD 简介	129

第 8 章 通用输入/输出口	133
8.1 通用 I/O 模式的设置	133
8.1.1 输入模式	134
8.1.2 输出模式	134
8.1.3 开漏模式	134
8.1.4 准双向模式	135
8.2 相关寄存器	135
8.3 实 验	136
第 9 章 定时器控制器与系统定时器	142
9.1 定时器控制器	142
9.1.1 概 述	142
9.1.2 特 征	142
9.1.3 定时器操作模式	143
9.1.4 相关寄存器	144
9.1.5 实 验	146
9.2 系统定时器	150
9.2.1 概 述	150
9.2.2 相关寄存器	151
9.2.3 示例代码	152
第 10 章 PWM 发生器和捕捉定时器	153
10.1 概 述	153
10.2 特 征	155
10.2.1 PWM 功能特性	155
10.2.2 PWM 捕捉功能模块特性	155
10.3 功能描述	155
10.3.1 PWM 定时器操作	155
10.3.2 PWM 双缓存、自动重载以及单触发模式	156
10.3.3 调至占空比	157
10.3.4 死区发生器	157
10.3.5 捕捉操作	158
10.3.6 PWM 定时器中断结构	159
10.3.7 PWM-定时器开启步骤	160
10.3.8 PWM-定时器关闭步骤	160

目 录

10.3.9 捕捉开始步骤	160
10.4 相关寄存器	161
10.5 实 验	171
第 11 章 串口控制器	176
11.1 概 述	177
11.2 特 性	179
11.3 相关寄存器	180
11.4 串口发送实验	190
11.5 串口收发实验	194
11.6 模拟串口实验	198
第 12 章 外部中断	204
12.1 外部中断简介	204
12.2 相关寄存器	205
12.3 实 验	206
第 13 章 看门狗	211
13.1 概 述	211
13.2 特 征	213
13.3 相关寄存器	213
13.4 实 验	214
第 14 章 Flash 内存控制器(FMC)	219
14.1 概 述	219
14.2 特 性	219
14.3 FMC 组织结构	219
14.4 在系统编程(ISP)	222
14.5 相关寄存器	223
14.6 ISP 实验	227
第 15 章 I²C 总线控制器	234
15.1 概 述	234
15.2 特 征	235
15.3 功能描述	235
15.4 操作模式	238

15.5	相关寄存器	239
15.6	AT24C02	242
15.7	实 验	244
第 16 章	串行外围设备接口(SPI)控制器	252
16.1	概 述	252
16.2	特 性	252
16.3	功 能	253
16.4	时序波形图	258
16.5	相关寄存器	261
16.6	实 验	265
第 17 章	模拟/数字转换	273
17.1	概 述	273
17.2	特 征	275
17.3	操作步骤	276
17.4	相关寄存器	280
17.5	实 验	287
第 18 章	RTX Kernel 实时系统	291
18.1	实时系统与前后台系统	291
18.2	RTX Kernel 技术参数	294
18.3	RTX Kernel 配置	296
18.4	RTX Kernel 组成部分	297
18.5	实 验	304
第 19 章	杂项补遗	313
19.1	详解启动文件	313
19.2	LIB 的生成与使用	318
19.2.1	LIB 文件的创建	319
19.2.2	LIB 文件的使用	320
19.3	Hex 文件	322
19.3.1	Hex 的结构	322
19.3.2	Hex 的数据记录	323
19.4	功耗控制	325
19.4.1	相关寄存器	326

目 录

19.4.2	空闲模式唤醒实验	328
19.4.3	掉电模式唤醒实验	331
19.5	系统复位	335
19.5.1	相关寄存器	335
19.5.2	实 验	336
19.6	scatter 文件	337
19.6.1	scatter 文件简介	338
19.6.2	实 验	340
19.7	USER 配置	347
19.7.1	相关寄存器	347
19.7.2	实 验	348
19.8	欠压电压值设定(BOD)	355
19.8.1	相关寄存器	356
19.8.2	实 验	357
19.9	CMSIS 编程标准	362
19.9.1	CMSIS 标准的软件架构	363
19.9.2	CMSIS 规范	364
19.9.3	CMSIS 标准的代码实现	366
19.10	外部总线接口(EBI)	367
19.10.1	操作步骤	368
19.10.2	相关寄存器	372
第 20 章	串行输入并行输出	374
20.1	74LS164 简介	374
20.2	74LS164 结构	375
20.3	74LS164 函数	377
第 21 章	数码管	380
21.1	数码管简介	380
21.2	字型码	380
21.3	驱动方式	382
21.3.1	数码管驱动方式	382
21.3.2	动态驱动	382
21.4	实 验	383

第 22 章 LCD	391
------------------	-----

22.1 液晶简介	391
22.2 LCD1602 液晶及显示实验	391
22.3 LCD12864 液晶及显示实验	401

第 3 篇 深入篇

第 23 章 深入接口	417
-------------------	-----

23.1 简介	417
23.2 校验介绍	417
23.2.1 奇偶校验	417
23.2.2 校验和	419
23.2.3 循环冗余码校验	420
23.3 数据校验实战	422
23.3.1 数据帧格式定义	423
23.3.2 实 验	426

第 24 章 深入编程	437
-------------------	-----

24.1 编程规范	437
24.1.1 排 版	438
24.1.2 注 释	439
24.1.3 标识符	441
24.1.4 函 数	443
24.2 代码架构	444
24.2.1 功能模块构建	444
24.2.2 简易前后台系统构建	445
24.2.3 简易定时系统构建	456
24.3 高级应用集锦	466
24.3.1 宏	466
24.3.2 函数指针	468
24.3.3 结构体、共用体	470
24.3.4 程序优化	471
24.3.5 软件抗干扰	490
24.3.6 软件低功耗设计	492

目 录

第 4 篇 番外篇

第 25 章 界面开发	497
25.1 VC++2008	498
25.2 HelloWorld 小程序	498
25.3 实现串口通信	503
25.3.1 创建界面	503
25.3.2 CSerial 类	504
25.3.3 编写程序	512
25.3.4 运行程序	514
25.4 动态链接库	514
25.4.1 动态链接库优点	515
25.4.2 动态链接库创建流程	516
25.4.3 编写串口动态链接库	518
25.4.4 调用串口动态链接库	522
附录 A 开发板原理图	528
A.1 原理图	528
A.2 实物图	532
附录 B 单片机多功能调试助手	534
参考文献	535

绪 论

0.1 什么是微控制器

微控制器是指一个集成在一块芯片上的完整计算机系统。尽管它的大部分功能集成在一块小芯片上,但是它具有一个完整计算机所需要的大部分部件:CPU、内存、内部和外部总线系统,目前大部分还会具有外存。同时集成诸如通信接口、定时器、实时时钟等外围设备。而现在最强大的微控制器系统甚至可以将声音、图像、网络、复杂的输入/输出系统集成在一块芯片上,如图 0.1.1 所示。



图 0.1.1 NuMicro M051 系列微控制器

0.2 微控制器历史

微控制器诞生于 20 世纪 70 年代末,经历了 SCM、MCU、SoC 三大阶段。

(1) SCM 即单片微型计算机(Single Chip Microcomputer)阶段,主要是寻求最佳的单片形态嵌入式系统的最佳体系结构。“创新模式”获得成功,奠定了 SCM 与通用计算机完全不同的发展道路。在开创嵌入式系统独立发展道路上,Intel 公司功不可没。

(2) MCU 即微控制器(Micro Controller Unit)阶段,主要的技术发展方向是:不断扩展满足嵌入式应用时,对象系统要求的各种外围电路与接口电路,突显其对象的智能化控制能力。它所涉及的领域都与对象系统相关,因此,发展 MCU 的重任不可避免地落在电气、电子技术厂家。从这一角度来看,Intel 逐渐淡出 MCU 的发展也有其客观因素。在发展 MCU 方面,最著名的厂家当数 NXP(原 Philips)公司。

NXP 公司以其在嵌入式应用方面的巨大优势,将 MCS-51 从单片微型计算机迅速发展到微控制器。因此,当我们回顾嵌入式系统发展道路时,不要忘记 Intel 和 NXP 的历史功绩。