



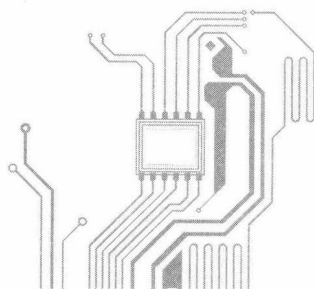
ARM Cortex-M3 嵌入式开发与实践 —— 基于STM32F103

◎ 张勇 编著

- ◎ 按照“认知—应用—提高”的思路，循序渐进带您进入ARM Cortex-M3的开发之路
- ◎ 全面介绍STM32F103微控制器内部架构及其常用典型硬件电路，针对片上外设，基于Keil MDK阐述寄存器和库函数两种类型工程程序设计方法
- ◎ 基于嵌入式实时操作系统 μ C/OS-II和库函数类型工程，探讨面向任务程序设计方法

清华大学出版社





ARM Cortex-M3 嵌入式开发与实践

—— 基于STM32F103

◎ 张勇 编著

清华大学出版社
北京

内 容 简 介

本书基于 ARM Cortex-M3 内核微控制器 STM32F103 和嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$, 详细讲述嵌入式系统的硬件设计与软件开发技术, 主要内容包括嵌入式系统概述、STM32F103 微控制器、STM32F103 学习平台、LED 灯控制与 Keil MDK 工程框架、按键与中断处理、定时器、串口通信、存储器管理、LCD 屏与温/湿度传感器、 $\mu\text{C}/\text{OS-II}$ 系统与移植、 $\mu\text{C}/\text{OS-II}$ 任务管理、信号量与互斥信号量、消息邮箱与消息队列等。本书的特色在于理论与应用紧密结合, 实例丰富, 对于基于 STM32F1 系列微控制器及嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 的教学和工程应用, 都具有一定的指导和参考价值。

本书可作为普通高等院校物联网、电子工程、通信工程、自动化、智能仪器、计算机工程和嵌入式控制等相关专业的高年级本科生教材, 也可作为嵌入式系统爱好者和工程开发技术人员的参考用书。

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 嵌入式开发与实践: 基于 STM32F103/张勇编著. —北京: 清华大学出版社, 2017
(电子设计与嵌入式开发实践丛书)
ISBN 978-7-302-46052-7

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

中国版本图书馆 CIP 数据核字(2017)第 002756 号

责任编辑: 刘 星 梅栾芳

封面设计: 刘 键

责任校对: 时翠兰

责任印制: 李红英

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: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19.25

字 数: 470 千字

版 次: 2017 年 3 月第 1 版

印 次: 2017 年 3 月第 1 次印刷

印 数: 1~2000

定 价: 49.00 元

产品编号: 071637-01

前言

自 1971 年第一块单片机诞生至今,嵌入式系统的发展经历了初期阶段和蓬勃发展期,现已进入了成熟期。在嵌入式系统发展初期,各种 EDA 工具还不完善,芯片的制作工艺和成本颇高,嵌入式程序设计语言以汇编语言为主,该时期只有电子工程专业技术人员才能从事嵌入式系统设计与开发工作。到了 20 世纪 80 年代,随着 MCS-51 系列单片机的出现以及 C51 程序设计语言的成熟,单片机应用系统成为嵌入式系统的代名词,MCS-51 单片机迅速在智能仪表和自动控制等相关领域得到普及。同时期,各种 DSP 芯片、FPGA 芯片和 SoC 芯片也如雨后春笋般涌现出来,应用领域从最初的自动控制应用扩展到各种各样的智能应用系统。随后 1997 年,ARM 公司推出 ARM7 微控制器,之后推出 Cortex 系列微控制器和微处理器,成为嵌入式系统设计的首选芯片,标志着嵌入式系统进入蓬勃发展期。

全球的半导体厂商在芯片制造上“百花齐放,百家争鸣”,是嵌入式系统蓬勃发展阶段的突出写照。这段时期,嵌入式系统工程师同时兼做硬件工程师和软件工程师,需要涉猎各种各样的芯片应用知识,并开发各具特色的应用程序。直到 21 世纪初,开源嵌入式实时操作系统出现,嵌入式系统工程师才真正分为嵌入式系统硬件工程师和嵌入式系统软件工程师,硬件工程师负责硬件电路板设计、芯片外设访问驱动函数开发和嵌入式实时操作系统移植等,软件工程师负责系统资源管理与调度、图形用户交互界面设计和应用程序设计等,这标志着嵌入式系统已经发展到成熟期,从几十个工程师到成百上千的工程师,通过细致分工协力合作进行同一项嵌入式系统研发。

本书内容分为两篇,第 1 篇主要面向硬件工程师和物联网与电子设计类本科生,第 2 篇偏向于硬件工程师,同时也兼顾软件工程师。本书由作者近几年来在江西财经大学软件与通信工程学院的“嵌入式系统应用”和“嵌入式系统原理”课程教学的讲义改编而成的,按章节顺序进行教学活动和课程实验。

第 1 篇包括 9 章,是全书的硬件基础和芯片级别程序设计部分。第 1 章介绍嵌入式系统的发展历程和应用领域;第 2 章介绍 ARM Cortex-M3 内核微控制器芯片 STM32F103 的内部结构、引脚配置、存储器、片内外设、异常与 NVIC 中断等;第 3 章介绍 ALIENTEK 战舰 STM32F1 学习板的硬件原理,重点介绍本书中使用的电路模块,如 STM32F103 核心电路、电源电路与按键电路、LED 与蜂鸣器电路模块、串口模块、Flash 与 EEPROM 电路模块、温/湿度传感器模块、LCD 屏模块和 SRAM 模块等,这部分内容为后面程序设计的硬件电路基础;第 4 章讨论 STM32F103 的 GPIO 访问方法以及 LED 灯控制技术,并完整地介

Foreword

绍基于 Keil MDK 创建工程的方法,后面的工程均基于该工程框架;第 5 章深入分析 NVIC 中断的工作原理,重点介绍 GPIO 口外部输入中断的处理方法,并给出按键响应实例;第 6 章阐述 STM32F103 内部通用定时器、看门狗定时器和系统节拍定时器的应用与实例,其中,系统节拍定时器主要用于为嵌入式实时操作系统提供时钟节拍(一般设为 100Hz);第 7 章介绍串口通信,一般借助中断方式从上位机接收串口数据,通过函数调用方式向上位机发送串口数据;第 8 章介绍 STM32F103 访问 Flash 芯片 W25Q128 和 EEPROM 芯片 24C02 的方法;第 9 章介绍 STM32F103 驱动 TFT LCD 屏的方法,介绍 LCD 屏显示字符和汉字的方法,并阐述温/湿度传感器 DHT11 的应用方法,展示 LCD 屏显示环境温/湿度值的应用实例。

第 2 篇为嵌入式实时操作系统级别的程序设计部分,介绍嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 在微控制器 STM32F103 上的移植和工程设计方法,包括 4 章,依次介绍系统组成与移植文件、任务管理与工程框架、信号量与互斥信号量、消息邮箱与消息队列。这篇内容中没有对嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 的内部工作原理进行剖析,感兴趣的读者可参考文献[6]、[8]。

作为教材,需要体现知识的完整性和可扩展性。通过本书的教学活动,展示给学生和读者一个从事嵌入式系统设计“认知—应用—提高”的全过程。“认知”体现为对嵌入式系统核心芯片的学习和掌握,重点在于学习一款芯片的存储器、中断与片内外设(合称为芯片的三要素),这也是第 2 章关于 STM32F103 芯片的重点内容;“应用”体现在应用芯片进行嵌入式电路板的设计,并掌握各个电路模块的工作原理和访问技术,会应用 C 语言进行驱动函数设计,即第 3~9 章的全部内容;“提高”是指实现该电路板嵌入式实时操作系统的移植,并将底层硬件的访问方法抽象为函数调用,即第 2 篇的内容,使得没有硬件电路基础的软件工程师可在此基础上开发出高性能的用户应用程序,并实现友好的图形用户界面。

本书全部工程的源代码可从清华大学出版社网站本书主页下载。本书中的全部工程都是完整且相互联系的,后续章节的工程建立在前面章节工程的基础上,是添加了新的功能而构建的。本书以有限的篇幅巧妙地将所有工程的源代码都包含进来,强烈建议读者自行录入源程序,以加强学习效果。

本书第 3 章的硬件学习平台借鉴广州市星翼电子科技有限公司(www.alientek.com) ALIENTEK 战舰 STM32F1-V3 开发板的硬件电路原理图,需要指出的是,第 3 章的学习平台是一个完整的硬件平台,也是作者鼓励学生们分组开展设计的硬件实验平台。同时,本书的所有工程均使用 ALIENTEK 战舰 STM32F1-V3 开发板测试通过。需要特别说明的是,星翼电子主持的开源技术论坛(www.openedv.com)收集了学习 32 位 STM32 系列微控制器最优秀的资源。在本书成稿之际,对星翼电子公司的张洋总经理给予的大力支持表示诚挚的谢意,同时,还要感谢学生何维和彭锦在工程验证上所做的大量工作,感谢清华大学出版社工作人员对本书出版付出的辛勤工作。由于作者水平有限,书中难免会有疏漏之处,敬请同行专家和读者批评指正。

作者于江财枫林校区

2016 年 11 月

目 录

第 1 篇 STM32F103 硬件系统与 Keil MDK 工程

第 1 章 嵌入式系统概述	3
1.1 嵌入式系统范例	3
1.2 嵌入式系统概念	4
1.2.1 嵌入式系统与 ARM 的关系	5
1.2.2 嵌入式系统与嵌入式操作系统的关系	5
1.2.3 嵌入式系统研发特点	6
1.3 ARM 发展历程及应用领域	7
1.3.1 ARM 发展史及命名规则	7
1.3.2 ARM 微处理器系列	8
1.3.3 ARM 微处理器应用领域	12
1.4 嵌入式操作系统	13
1.4.1 Windows CE	14
1.4.2 VxWorks	15
1.4.3 嵌入式 Linux	15
1.4.4 Android 系统	16
1.5 μ C/OS-II 与 μ C/OS-III	16
1.5.1 μ C/OS 发展历程	16
1.5.2 μ C/OS-II 特点	18
1.5.3 μ C/OS-III 特点	19
1.5.4 μ C/OS 应用领域	21
1.6 本章小结	21
习题	22
第 2 章 STM32F103 微控制器	23
2.1 STM32F103 概述	23

Contents

2.2	STM32F103ZET6 引脚定义	24
2.3	STM32F103 架构	30
2.4	STM32F103 存储器	33
2.5	STM32F103 片内外设	36
2.6	STM32F103 异常与中断	39
2.7	本章小结	41
	习题	41
第 3 章	STM32F103 学习平台	42
3.1	STM32F103 核心电路	43
3.2	电源电路与按键电路	48
3.3	LED 与蜂鸣器驱动电路	49
3.4	串口通信电路	49
3.5	Flash 与 EEPROM 电路	50
3.6	温/湿度传感器电路	51
3.7	LCD 屏接口电路	51
3.8	JTAG 与复位电路	52
3.9	SRAM 电路	53
3.10	本章小结	54
	习题	54
第 4 章	LED 灯控制与 Keil MDK 工程框架	55
4.1	STM32F103 通用目的输入/输出	55
4.1.1	GPIO 寄存器	56
4.1.2	AFIO 寄存器	59
4.2	STM32F103 库函数用法	61
4.3	Keil MDK 工程框架	64
4.4	LED 灯闪烁实例	72
4.4.1	寄存器类型工程实例	72
4.4.2	库函数类型工程实例	76
4.5	本章小结	80
	习题	80
第 5 章	按键与中断处理	81
5.1	NVIC 中断工作原理	81
5.2	GPIO 外部输入中断	88
5.3	用户按键中断实例	89
5.3.1	寄存器类型工程实例	89
5.3.2	库函数类型工程实例	95

5.4 本章小结	99
习题	99
第6章 定时器	100
6.1 系统节拍定时器	100
6.1.1 系统节拍定时器工作原理	100
6.1.2 系统节拍定时器实例	103
6.2 看门狗定时器	106
6.2.1 窗口看门狗定时器工作原理	106
6.2.2 窗口看门狗定时器寄存器类型实例	107
6.2.3 窗口看门狗定时器库函数类型实例	109
6.3 实时时钟	111
6.3.1 实时时钟工作原理	112
6.3.2 实时时钟寄存器类型实例	113
6.3.3 实时时钟库函数类型实例	116
6.4 通用定时器	118
6.4.1 通用定时器工作原理	118
6.4.2 通用定时器寄存器类型实例	120
6.4.3 通用定时器库函数类型实例	121
6.5 本章小结	124
习题	124
第7章 串口通信	125
7.1 串口通信工作原理	125
7.2 STM32F103 串口	126
7.3 串口通信寄存器类型实例	129
7.4 串口通信库函数类型实例	134
7.5 本章小结	137
习题	137
第8章 存储器管理	138
8.1 SRAM 存储器	138
8.1.1 访问 SRAM 存储器寄存器类型实例	144
8.1.2 访问 SRAM 存储器库函数类型实例	149
8.2 EEPROM 存储器	153
8.2.1 访问 EEPROM 寄存器类型实例	157
8.2.2 访问 EEPROM 库函数类型实例	164
8.3 Flash 存储器	167
8.3.1 STM32F103 同步串行口	167

8.3.2	W25Q128 访问控制	170
8.3.3	访问 Flash 存储器寄存器类型工程实例	172
8.3.4	访问 Flash 存储器库函数类型工程实例	182
8.4	本章小结	187
	习题	187
第 9 章	LCD 屏与温/湿度传感器	188
9.1	LCD 屏显示原理	188
9.2	温/湿度传感器	200
9.3	LCD 显示实例	204
9.3.1	寄存器类型实例	204
9.3.2	库函数类型实例	211
9.4	本章小结	216
	习题	216
 第 2 篇 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$		
第 10 章	$\mu\text{C}/\text{OS-II}$ 系统与移植	221
10.1	$\mu\text{C}/\text{OS-II}$ 系统移植	221
10.2	$\mu\text{C}/\text{OS-II}$ 系统结构与配置	230
10.3	$\mu\text{C}/\text{OS-II}$ 系统任务	236
10.3.1	空闲任务	236
10.3.2	统计任务	237
10.3.3	定时器任务	237
10.4	本章小结	238
	习题	238
第 11 章	$\mu\text{C}/\text{OS-II}$ 任务管理	239
11.1	$\mu\text{C}/\text{OS-II}$ 用户任务	239
11.2	$\mu\text{C}/\text{OS-II}$ 多任务工程实例	243
11.3	统计任务实例	256
11.4	系统定时器	259
11.5	本章小结	263
	习题	263
第 12 章	信号量与互斥信号量	264
12.1	$\mu\text{C}/\text{OS-II}$ 信号量	264
12.2	$\mu\text{C}/\text{OS-II}$ 互斥信号量	266
12.3	信号量与互斥信号量实例	268
12.4	本章小结	282

习题	282
第 13 章 消息邮箱与消息队列	283
13.1 μ C/OS- II 消息邮箱	283
13.2 μ C/OS- II 消息队列	285
13.3 消息邮箱与消息队列实例	287
13.4 本章小结	296
习题	296
参考文献	297

第 1 篇

STM32F103 硬件系统 与 Keil MDK 工程

本篇内容包括第 1~9 章,为全书的硬件基础和芯片级别的程序设计部分,依次介绍嵌入式系统概念、STM32F103 微控制器、STM32F103 学习平台、LED 灯控制与 Keil MDK 工程框架、按键与中断处理、定时器、串口通信、存储器管理以及 LCD 屏与温/湿度传感器等内容。

嵌入式系统概述

STM32F103 微控制器主要应用于各类嵌入式系统中,本章从宏观角度介绍嵌入式系统和各类嵌入式操作系统的概念,重点分析广泛应用于 STM32F103 微控制器的嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 和 $\mu\text{C}/\text{OS-III}$ 的特点。

本章的学习目标:

- 了解嵌入式系统的组成;
- 熟悉 ARM 微控制器的发展历程;
- 熟悉嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 的特点。

1.1 嵌入式系统范例

相对于通用计算机系统而言,嵌入式系统也称为嵌入式计算机系统,随着物联网技术的飞速发展,普遍认可的嵌入式系统的定义是:“以应用为中心,以计算机技术为基础,软硬件可裁剪,满足应用系统对功能、可靠性、成本、体积和功耗等严格要求的专用计算机系统”。例如,生活中随处可见的天网监控系统、智能家居、汽车、大型家电、数字机顶盒、医疗设备、银行 ATM 机、GPS 导航仪和交通控制系统等,都集成了大量嵌入式系统。

下面通过一个实例进一步阐述嵌入式系统的范畴。

一般地,高校教学楼每层都安设了饮水机,方便教师和学生用水;此外,高速公路服务区、列车站和机场中也安设了各种智能饮水机,为旅行者提供开水。饮水机的主要功能是提供 100°C 的开水,其智能化体现在全自动操作上,例如可以自动进水、自动补水、满水时自动停止进水、自动温度控制、防干烧保护和温度显示等。有些高级的饮水机还提供冷水,即水烧开后,将开水分流一部分进入冷却仓中,可直接饮用。

饮水机的整个控制系统是一种典型的嵌入式系统,其核心类似于 STM32F103 的微控制器,这里用 STM32F103 表示,通过各种外部设备和传感器实现饮水机的智能控制,如图 1-1 所示。

图 1-1 中,控制中心 STM32F103 通过周期性地访问水位传感器和温度传感器,实时地记录饮水机的水位和水温,同时,控制 LED 灯实时显示饮水机的工作状态(例如,绿灯亮表

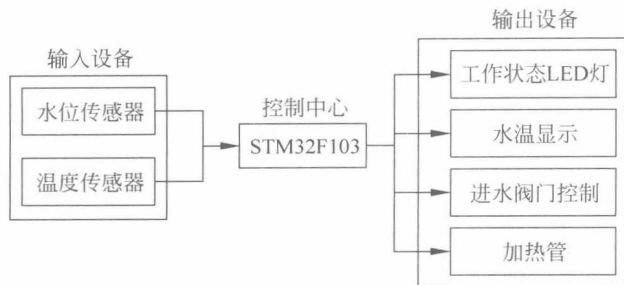


图 1-1 饮水机嵌入式系统结构

示开水,红灯亮表示加热)并实时显示水温。当水位的高度低于设定的门限高度时,STM32F103 打开进水阀门自动进水,当水位涨到设定的最高水位时,STM32F103 关闭进水阀门。当水温低于设定的温度后,STM32F103 将自动启动加热管加热水仓中的水,当温度达到 100°C 时,STM32F103 关闭加热管,停止加热并进入保温状态,在此过程中,通过 LCD 屏或数码管显示水温变化。图 1-1 列出了饮水机的基本功能,饮水机嵌入式系统还应具有自检、报警、恒温处理等功能。

由图 1-1 可见,典型的嵌入式系统的硬件主要包括 3 部分,即控制中心、输入设置和输出设备,有时也称为数据处理中心、数据采集端和数据输出端。不同的应用系统,其嵌入式系统也不尽相同,一般地,控制中心是由 ARM 微控制器、DSP、FPGA 或传统 8051 单片机等可编程器件组成的核心电路,通过软件实现相应的控制或数据处理功能;输入设备和输出设备根据应用场合的不同,选用相应的传感器或显示终端。

如果将图 1-1 所示的饮水机添加 WiFi 或蓝牙设备,实现联网控制,则该饮水机就成为物联网的一分子。假设从北京至广州的高速公路的全部服务区的饮水机都通过 APP(手机应用程序)联网,则游客可实时了解各个服务区饮水机的情况,从而可选择合适的服务区采水。这正是物联网给人们的生产生活带来的方便。

诚然,设计嵌入式系统要按照具体问题,具体分析的原则,根据实际问题的应用需求,选择合适的嵌入式系统。有些专家将物联网称为嵌入式系统的联网,可见嵌入式系统在物联网中具有核心地位,而微控制器又是嵌入式系统的核心。因此,基于微控制器的硬件设计和软件开发技术,是电子、通信、智能控制和物联网等相关专业学生必须具备的专业素养,可以从学习基于 STM32F103 微控制器的硬件和软件设计入手,不断开拓嵌入式系统新的应用领域。

1.2 嵌入式系统概念

数字技术和软件技术是嵌入式系统的核心技术,其中,数字技术包括数字信号处理技术和数字化芯片技术,软件技术包括芯片级的程序设计技术和操作系统级的程序设计技术。电路系统由传统的模拟电子系统演化为以可编程数字化芯片为核心、添加必要外设接口实现相应功能的嵌入式系统,在三个相互关联又相对独立的技术领域表现突出,即以单片机为核心的嵌入式控制领域、以 DSP(数字信号处理器)或 FPGA(现场可编程门阵列)为核心的

嵌入式数字信号处理领域、以 ARM 或 SoC(片内系统芯片)为核心的嵌入式操作系统及其应用领域。一般地,嵌入式系统被理解为一个相对概念,即在硬件上,它是嵌入在更大规模硬件系统中的电路系统,嵌入式系统的本质在于其硬件系统具有灵活的可编程、可再配置软件等特性,即嵌入式系统必须具有自身的软件系统。

1.2.1 嵌入式系统与 ARM 的关系

广义上,凡是嵌入到应用系统中的电子系统都可以统称为嵌入式系统,即使是通用的计算机系统,如果是嵌入在特定的应用系统中,也可被称为嵌入式系统,例如,在虚拟仪表系统中用于数据采集、分析和显示的嵌入式计算机系统。狭义上,嵌入式系统除了具有硬件和软件之外,还要求硬件系统具有体积小、重量轻、功耗低、成本低、可靠性高、可升级等特点,要求软件系统具有体积小、可裁剪性、健壮性、专用性、实时性等特点。因此,从狭义上讲,嵌入式系统硬件往往是以 ARM 芯片为核心的硬件平台,嵌入式系统软件为基于芯片级开发的无操作系统汇编或 C 语言实时性软件,或者是基于嵌入式实时操作系统开发的图形界面应用程序。

而 ARM 是指 ARM 公司设计的基于 RISC 架构的 32 位高性能微处理器,一般采用哈佛总线结构,具有高速指令缓存和数据缓存,指令长度固定且多级流水执行,具有 MMU(存储器管理单元)和 AMBA(高级微处理器总线结构)总线接口等。ARM 芯片除具有 ARM 核心外,通常还具有丰富的外设接口,例如,外扩 RAM(随机访问存储器)和 Flash 控制器、LCD 控制器、串行接口、SD 接口、USB 接口、I²C 和 I²S 总线接口等,此外,ARM 芯片还具有低功耗、体积小等特性。ARM 芯片的高性能、多接口特点决定了其比单片机和 DSP 更适合作为嵌入式系统的核心微处理器,因此,ARM 系统几乎成为嵌入式系统的代名词。

1.2.2 嵌入式系统与嵌入式操作系统的关系

一般地,嵌入式系统是面向特定应用和环境、集成硬件和软件的单板机,嵌入式系统的硬件资源有限,突出地表现在其具有较小容量的 RAM 和 ROM 空间,通过外扩 SD 卡等存储介质扩展存储空间;嵌入式系统的软件,包括嵌入式操作系统软件,都固化在 Flash 芯片中。因此,嵌入式操作系统软件体积较小,一般在 32MB 以下。

嵌入式系统的软件分为两种:其一为直接基于 ARM 芯片开发的汇编或 C 语言实时性程序,这时的程序代码负责管理 ARM 片上的所有资源,包括存储空间和片上外设,程序除根据需要设计特定的功能之外,还要编写 ARM 芯片初始化代码和中断向量表,更重要的是,程序在访问嵌入式系统的硬件时,必须充分考虑硬件接口的时序特点;其二是为嵌入式系统定制多任务、实时的嵌入式操作系统,嵌入式操作系统抽象了嵌入式系统的硬件访问方式,通过提供 API(应用程序接口)函数的方式,在嵌入式操作系统基础上设计用户应用程序,只需调用相应的 API 函数即可,使得嵌入式系统的应用程序设计工作更加简单方便。

由此可见,嵌入式操作系统也具有桌面操作系统的特点,即管理硬件资源、调度软件进程、处理软件中断等,嵌入式操作系统通常包括硬件驱动软件、系统内核、设备驱动接口、文件系统、图形界面等。嵌入式操作系统要求具有实时性、多任务、模块化、可移植性、可定制等特点,流行的嵌入式操作系统有 Windows CE、嵌入式 Linux、 μ C/OS-II、VxWorks 等。

因此,嵌入式系统可以表示为

嵌入式系统 = ARM 硬件系统 + 嵌入式操作系统 + 操作系统级应用软件系统
或者

嵌入式系统 = ARM 硬件系统 + 芯片级应用软件系统

1.2.3 嵌入式系统研发特点

嵌入式系统研发需要具备电子类和软件类两方面的专门知识,是一门交叉组合型学科。嵌入式系统研发可分为 4 类。

首先,嵌入式系统的硬件平台设计,需要根据应用环境选择合适的 ARM 芯片,满足处理速度和存储深度的要求,同时,需要兼顾性价比和芯片特点与生存周期等因素。ARM 芯片选型后,根据要实现的功能,添加相应的外设接口处理芯片和电源与时钟芯片等,借助 Altium Designer 等 EDA 软件完成硬件平台的原理性设计和 PCB 设计。目前,嵌入式系统硬件平台的设计基本上实现了模型化设计,即 ARM 芯片与外设芯片的接口电路都形成了规范,只需要按模型将 ARM 芯片与所需外设芯片连接起来就可以得到特定的嵌入式系统硬件平台。尽管如此,读懂和分析这个模型仍然需要一定的电路基础。

其次,基于 ARM 芯片的芯片级汇编或 C 语言程序设计,要求设计者对 ARM 芯片工作原理和内部结构有较好的认识和理解,这类程序包括系统初始化程序和特定功能的算法程序,需要对汇编语言和指令以及 C 语言编程有一定的基础。目前,芯片级软件设计达到了框架化的水平,即在现有的框架程序的基础上,添加特定的软件功能以达到程序设计的目的。因此,程序员须对框架程序有深入全面的了解,程序员的主要工作集中在使用 C 语言开发算法上。

然后,嵌入式操作系统的定制和驱动程序的开发,这类研发已经完全商业化。设计者可以根据自己选用的 ARM 芯片,直接购买特定的相兼容的嵌入式操作系统软件,只需要操作鼠标就可以定制出功能强大的专用嵌入式操作系统。并且嵌入式操作系统供应商也会提供几乎所有常见外设的驱动程序,例如触摸屏、LCD 屏、网口、串口、USB 口等。如果设计者想自行研发具有独立知识产权的嵌入式操作系统,那么认真学好开源的 $\mu\text{C}/\text{OS-II}$ 嵌入式操作系统是一个建设性的忠告。

最后,基于嵌入式操作系统开发用户应用程序,特别是开发具有良好图形界面的用户应用程序,是对设计者的一个挑战。基于不同的嵌入式操作系统,开发应用程序的方式有很大的不同。嵌入式 Linux 和 Windows CE 都提供了良好的界面设计支持,分别可以借助 QT 和 Visual Studio 进行应用软件开发。用户可能需要对基于事件消息驱动的编程有进一步的了解,同时,如果是基于 Windows CE,那么掌握 .Net Framework 编程是一条简捷的路径。嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 没有图形用户界面(GUI),但 Micrium 公司提供了商业性的用户界面系统 $\mu\text{C}/\text{GUI}$ (或 SEGGER 公司的 emWin 系统),与 $\mu\text{C}/\text{OS-II}$ 无缝连接。笔者于西安电子科技大学出版社出版的《Windows CE 应用程序设计》在嵌入式应用程序开发方面能起到较好的引导作用。

本书的中心任务是讨论 STM32F103 微控制器芯片级和 $\mu\text{C}/\text{OS-II}$ 操作系统级的程序设计方法,程序设计语言为 C 语言,并涵盖了嵌入式系统开发的硬件设计、软件设计和操作

系统级别的应用程序设计等内容。

1.3 ARM 发展历程及应用领域

ARM(Advanced RISC Machine,高级精简指令集机器)是 ARM 公司设计的 32 位总线的高性能微处理器。ARM 公司(www.arm.com)本身不生产芯片,通过转让或出售 ARM 技术给 OEM(原始设备生产商)专业生产商生产和销售 ARM 芯片给第三方用户。全球内有约 200 家大型半导体生产厂商购买了 ARM 知识产权,生产具有 ARM 核的芯片,每秒就有约 90 片 ARM 芯片被使用。

自 1985 年第一个 ARM1 原型诞生至今,ARM 公司设计的成熟 ARM 体系结构(或称指令集体系结构 ISA)有 ARMv4、ARMv4T、ARMv5TE、ARMv5TEJ、ARMv6 和 ARMv7 等,并且版本号还在不断升级,对应的处理器家族有 ARM7、ARM9、ARM9E、ARM10E、ARM11、Cortex、SecurCore 和 XScale 处理器系列等。应用领域涉及商业、军事、航天航空、网络与无线通信、消费电子、医疗电子、仪器仪表和汽车电子等各行各业。

1.3.1 ARM 发展史及命名规则

每个 ARM 处理器都对应于一个特定的 ARM 指令集体系结构版本,例如,ARM920T 微处理器支持指令集体系结构 ARMv4T。ARM 体系结构的发展史如表 1-1 所示。

表 1-1 ARM 体系结构发展史

版 本	典型微处理器类型	特 点
ARMv1~ARMv4	已退市	早期的版本中只有 ARMv4,目前在某些 ARM7 和 StrongARM 处理器中可见,可以被视为 32 位寻址的 32 位指令集体系结构
ARMv4T	ARM7TDMI, ARM7TDMI-S, ARM920T, ARM922T	支持 16 位的 Thumb 指令集,比 32 位的 ARM 指令集节省约 35% 的存储空间
ARMv5TE	ARM946E-S, ARM966E-S, ARM968E-S, ARM996HS	增加了 ARM 与 Thumb 状态切换的指令,增强了 DSP 类型指令,尤其是在语音数字信号处理方面提高了 70% 以上的性能
ARMv5TEJ	ARM7EJ-S, ARM926EJ-S, ARM1026EJ-S	添加了 Java 加速技术
ARMv6	ARM1176JZ(F)-S	改进了异常处理,更好地支持多处理器指令,增加了支持 SIMD(单指令多数据)的多媒体指令,对视频和音频解码性能提高近 4 倍
ARMv6T2	ARM1156T2(F)-S	支持 Thumb-2 技术
ARMv7	Cortex-A8, Cortex-A9, Cortex-R4(F)	支持 NEON 技术,使得 DSP 和多媒体处理性能提高 4 倍,支持向量浮点运算,为下一代 3D 图像和游戏硬件服务
ARMv7-M	Cortex-M3	优化了微控制器,低功耗