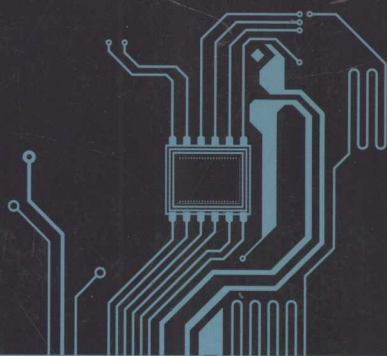




无线传感器 网络应用技术

—— 基于TinyOS及ZigBee PRO的实例设计

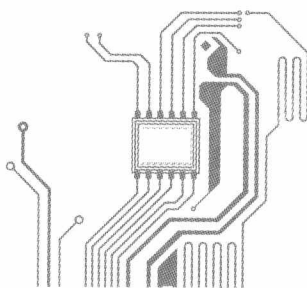
◎ 赵 成 编著



- ◎ 注重应用与实践，以技术为导向安排章节及实例，循序渐进，便于入门
- ◎ 程序讲解重点突出，注重剖析程序的设计与运行原理
- ◎ 力求在公开的官方资料上开发，崇尚应用实践的可重现性

清华大学出版社





无线传感器 网络应用技术

—— 基于TinyOS及ZigBee PRO的实例设计

◎ 赵 成 编著

清华大学出版社
北京

内 容 简 介

作者立足无线传感器网络的应用与实践,本着深入浅出、通俗易懂、承上启下、操作性强等编写原则,突出具体的应用设计,力求使本书成为讲授方法、指导实践性质的教材。本书既符合本科院校的系统化教学需要,又适用于高职高专类院校的实操性要求,也可用于企业技术培训。

本书结合无线传感器网络操作系统 TinyOS 与 ZigBee PRO 协议,分两部分介绍无线传感器网络的应用开发技术。第一部分是 TinyOS 2.x 应用部分,依次按 NesC 语言编程、GPIO 控制、串口通信、定时器、传感器测量、射频数据收发、低功耗控制、节点 RSSI 测量、路由协议介绍。第二部分是 ZigBee PRO 协议部分,与 TinyOS 部分相呼应,从 ZigBee 协议特点、简单模块控制到无线组网等技术依次展开,逐步掌握 ZigBee 开发无线传感器应用工程的方法。TinyOS 与 ZigBee 相得益彰,一个采用开源操作系统、编程简单,适用于应用与算法分析,另一个架构完备,设计工作量小,两者都是当前主要的无线传感器网络开发方式。

全书内容丰富、文字简练、图文并茂、结合实际,较为详尽地阐述了当前无线传感器网络的主流应用技术。本书可作为物联网工程、传感器网络、通信工程等专业高年级本科生及研究生的教材,也可供从事无线传感器网络系统开发的工程技术人员、科研人员阅读参考。

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

无线传感器网络应用技术:基于 TinyOS 及 ZigBee PRO 的实例设计/赵成编著.--北京:清华大学出版社,2016

(电子设计与嵌入式开发实践丛书)

ISBN 978-7-302-43836-6

I. ①无… II. ①赵… III. ①无线电通信—传感器—网络操作系统—程序设计 IV. ①TP212
②TP316.8

中国版本图书馆 CIP 数据核字(2016)第 101931 号

责任编辑:刘 星 薛 阳

封面设计:刘 键

责任校对:时翠兰

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:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.5 字 数:423 千字

版 次:2016 年 9 月第 1 版 印 次:2016 年 9 月第 1 次印刷

印 数:1~2500

定 价:39.00 元

前言

无线传感器网络课程在高校中的开设并非一朝一夕,早在物联网概念兴起之前,它就已经是一门理论丰富、实践新颖的课程。一方面,在科研领域,可以使用 TOSSIM、OMNeT++、OPNET、NS 等仿真软件进行算法仿真;另一方面,在应用领域,可以使用 Smart Dust、Mica、Tmote、Mote2 等系列节点进行软硬件应用系统设计。目前,随着无线传感器网络(以后简称为 WSNs)技术以及物联网产业的发展,在软件方面出现了大量开源的、半开源的以及商业应用的无线传感器网络操作系统(或协议栈),在硬件方面也出现了很多面向无线传感器网络应用的低价、低功耗、高性能的微处理器。毫无疑问,这已经在影响并改变无线传感器网络的研究与应用现状了,未来的趋势必将将是无线传感器网络技术的普及,包括理论算法的创新普及与系统应用创新普及。很明显,众多高校秉承包容、开放的理念,都在以极大的热情迎接无线传感器网络技术的新发展,特别是,高校物联网工程专业的申报得到了热烈响应,相应地,无线传感器网络课程也需要进行重新定位。

现在,物联网技术正在蓬勃发展,各个高校竞相建设物联网专业,青年学生对未来物联网的发展也充满了兴趣。基于多年来与企业、高校同行的交流,作者感觉国内现在缺少面向教育教学方面的物联网专业书籍,而专门针对 WSNs 教学的技术应用入门类书籍更难以寻觅。在教学岗位上,也听到教育设备厂家的呼声,他们忙着生产设备,为教学服务,却没有时间认真地整理一部好的讲义。因此,简短地说,无线传感器网络课程作为物联网专业的重要专业课,相应的课本应该是一本重实践性质的基础教材。

本书将目前主流的两种无线传感器网络的开发方式一并介绍,不仅可以对比两者开发、应用的特点,也可以更完整地掌握 WSNs 技术的开发。更主要的是,本书的每一个实践项目的源码都是经过多年的实验、课程设计与毕业设计不断使用与改进的,都可以在实际的实践项目开发中运行。另外,两种开发方式并列,可以根据课时灵活掌握讲授的内容。再者,本书定位于高校物联网专业偏重实践应用的教材,版式设计方便讲解、利于教学,内容化繁为简,重点突出。

全书共 18 章,分两篇介绍无线传感器网络应用技术。前篇包含第 1~9 章,介绍无线传感器网络操作系统 TinyOS 的程序设计;后篇包含第 10~18 章,介绍 ZigBee PRO 协议栈的应用。前篇基于 TinyOS,依次按 NesC 语言编程、GPIO 控制、串口通信、定时器控制、传感器测量、射频数据收发、低功耗控制、节点 RSSI 测量、路由协议介绍。后篇基于 ZigBee PRO

Foreword

协议栈,依次按 ZigBee 概述、GPIO 控制、串口通信、定时器控制、传感器测量、Z-Stack 协议通信、Z-Stack 点对点无线测温、Z-Stack 组播与广播、Z-Stack 协议组网展开介绍。TinyOS 与 ZigBee 相得益彰,通过具体的应用程序设计与分析,可以逐步掌握使用 TinyOS 与 ZigBee 开发无线传感器应用工程的方法。本书每章均配有习题,便于读者掌握和巩固知识,书后附有附录供读者查阅。使用本教材的教学参考学时为 32~48 学时。

基于教学与研发团队多年来的积累,本书提供众多的配套资料,包括 PPT 课件、实验指导书、程序源代码、应用软件、实验设备及模块等,可以通过自建网站、FTP 服务器、云存储等途径获取配套资料。并且,定期通过 QQ 群及微信平台发布与更新原创的应用方案及开源资料。

- 随书 PPT 课件(1 份),方便教师或培训讲师进行课程讲授;
- 书中涉及的程序代码及相关的应用软件(持续更新),方便开展实验或实训操作;
- 额外提供电子版的实验指导书(多份),方便自学或指导实验(实训)操作;
- 配套的实验设备及模块,依照口袋实验室的理念设计,方便自己动手 DIY 实例项目。

更多的资料请登录 www.dcsmart.cc 网站查看与下载。

本书由赵成编著。第 3 章由郭荣幸编写;第 4 章由华红艳教授编写;第 7、8 章由周鹏教授编写;第 11 章由赵雨斌教授编写;第 18 章由陈宇副教授编写。其余由赵成编写。全书由赵成统稿。

在本书的撰写过程中,我的领导与同事们始终给予大量的支持与鼓励,我的学生张蛟鹏精心完成了前篇全部组件结构与程序模型的电脑制图工作,张经纬同学细心调试了全部程序代码;韩柏科技以及迪卡工作室的工程师们也给予了很多技术支持,在此一并致谢。

该书算是对所有关心与帮助过我的同事、朋友和学生们的回馈,如果能产生一定的帮助与实际效果,这当然是我最大的欣慰。您可通过邮件(ziapc99@126.com)联系作者。

限于作者水平及成书时间的关系,书中难免存在错误和不妥之处,诚望读者批评指正。

赵 成

2016 年 7 月

目 录

上篇 基于 TinyOS 2.x 操作系统的应用设计

第 1 章 TinyOS 程序设计原理	3
1.1 TinyOS 操作系统	3
1.1.1 TinyOS 简介	3
1.1.2 TinyOS 的特点	4
1.1.3 TinyOS 调度机制	5
1.1.4 TinyOS 通信模型	5
1.1.5 TinyOS 能源效率	6
1.2 NesC 简单程序示例	6
1.2.1 NesC 语言简介	6
1.2.2 NesC 语言的特点	7
1.2.3 NesC 简单程序示例	7
1.2.4 NesC 编程结构	8
1.3 NesC 编程语法	9
1.3.1 代码页格式	9
1.3.2 NesC 编程语法	10
本章小结	11
习题	12
第 2 章 TinyOS LED 控制	13
2.1 LED 控制任务描述	13
2.1.1 任务描述	13
2.1.2 LED 控制项目结构	14
2.1.3 应用基础	14
2.2 程序的结构与模型	16
2.2.1 程序结构	16

Contents

2.2.2 程序模型	17
2.3 应用 Task 控制 LED	18
2.3.1 LED 控制程序源码	18
2.3.2 LED 控制原理	19
2.3.3 扩展分析	20
2.4 应用 Event 控制 LED	20
2.4.1 LED 控制程序源码	20
2.4.2 LED 控制原理	21
2.4.3 扩展分析	21
本章小结	22
习题	22
第 3 章 串口通信	24
3.1 串口通信任务描述	24
3.1.1 任务描述	24
3.1.2 串口通信项目结构	25
3.1.3 应用基础	25
3.2 程序的结构与模型	27
3.2.1 程序结构	27
3.2.2 程序模型	28
3.3 串口通信程序设计	28
3.3.1 串口通信程序源码	28
3.3.2 串口通信原理	33
3.3.3 扩展分析	34
本章小结	35
习题	35
第 4 章 定时器	37
4.1 定时器控制任务描述	37
4.1.1 任务描述	37
4.1.2 定时器控制项目结构	37
4.1.3 应用基础	38
4.2 程序的结构与模型	40
4.2.1 程序结构	40
4.2.2 程序模型	40
4.3 定时器控制	41
4.3.1 定时器应用程序源码	41
4.3.2 定时器控制原理	43
4.3.3 扩展分析	43

本章小结	44
习题	44
第 5 章 光敏传感器测量	46
5.1 光敏传感器测量任务描述	46
5.1.1 任务描述	46
5.1.2 光敏传感器测量项目结构	46
5.1.3 应用基础	47
5.2 程序的结构与模型	50
5.2.1 程序结构	50
5.2.2 程序模型	50
5.3 光敏传感器测量程序	51
5.3.1 光敏传感器测量程序源码	51
5.3.2 光敏传感器测量原理	54
5.3.3 扩展分析	55
本章小结	56
习题	57
第 6 章 射频通信	58
6.1 射频通信任务描述	58
6.1.1 任务描述	58
6.1.2 射频通信项目结构	59
6.1.3 应用基础	60
6.2 程序的结构与模型	64
6.2.1 程序结构	64
6.2.2 程序模型	65
6.3 射频数据通信实践	66
6.3.1 射频数据通信实践程序源码	66
6.3.2 射频数据通信原理	69
6.3.3 扩展分析	69
6.4 MAC 协议组网实践	70
6.4.1 MAC 协议组网实践程序源码	70
6.4.2 MAC 协议组网原理	72
6.4.3 扩展分析	73
本章小结	75
习题	75
第 7 章 低功耗 RF 控制	76
7.1 低功耗 RF 控制任务描述	76

7.1.1	任务描述	76
7.1.2	低功耗 RF 控制项目结构	76
7.1.3	应用基础	77
7.2	程序的结构与模型	80
7.2.1	程序结构	80
7.2.2	程序模型	81
7.3	低功耗 MAC 协议实践	82
7.3.1	低功耗 MAC 协议实践程序源码	82
7.3.2	低功耗 RF 控制原理	85
7.3.3	扩展分析	86
	本章小结	86
	习题	87
第 8 章	节点 RSSI 的测量	89
8.1	节点 RSSI 测量任务描述	89
8.1.1	任务描述	89
8.1.2	节点 RSSI 测量项目结构	89
8.1.3	应用基础	90
8.2	程序的结构与模型	92
8.2.1	程序结构	92
8.2.2	程序模型	93
8.3	基于 RSSI 的节点测距	93
8.3.1	基于 RSSI 的节点测距源码	93
8.3.2	基于 RSSI 的节点测距原理	96
8.3.3	扩展分析	97
	本章小结	99
	习题	100
第 9 章	路由协议	102
9.1	路由协议通信任务描述	102
9.1.1	任务描述	102
9.1.2	路由协议通信项目结构	103
9.1.3	应用基础	103
9.2	程序的结构与模型	105
9.2.1	程序结构	105
9.2.2	程序模型	106
9.3	路由协议通信应用程序	107
9.3.1	路由协议通信程序源码	107
9.3.2	路由协议通信程序分析	110

9.3.3 扩展分析	111
9.4 路由协议组件设计	113
9.4.1 路由协议数据结构	113
9.4.2 Flooding 路由协议组件设计	113
9.4.3 Gossiping 路由组件设计	115
9.4.4 扩展分析	117
本章小结	117
习题	117

下篇 基于 ZigBee PRO 协议的应用设计

第 10 章 ZigBee 技术概述	123
10.1 ZigBee 技术简介	123
10.1.1 ZigBee 的概念	123
10.1.2 ZigBee 设备类型	124
10.1.3 ZigBee 射频通道	124
10.1.4 ZigBee 网络拓扑结构	125
10.1.5 ZigBee 网络地址	126
10.1.6 ZigBee 技术特点	126
10.2 ZigBee 软件协议栈	128
10.2.1 Z-Stack 协议简介	128
10.2.2 Z-Stack 的工作原理	128
10.2.3 Z-Stack 的分层	129
10.3 CC2530 应用电路	130
10.3.1 CC2530 芯片的器件特性	130
10.3.2 CC2530 的功能结构	131
10.3.3 CC2530 的应用电路设计	132
本章小结	135
习题	135
第 11 章 CC2530 GPIO 控制	137
11.1 GPIO 接口输出控制	137
11.1.1 GPIO 接口控制基础	137
11.1.2 GPIO 接口输出控制	138
11.1.3 GPIO 输出控制效果	139
11.2 GPIO 接口输入控制	139
11.2.1 GPIO 接口控制基础	139
11.2.2 GPIO 接口输入控制	140
11.2.3 GPIO 输入控制效果	142

11.3	GPIO 接口中断控制	142
11.3.1	GPIO 接口的中断控制	142
11.3.2	GPIO 接口中断控制	144
11.3.3	GPIO 中断控制效果	146
	本章小结	146
	习题	146
第 12 章	CC2530 UART 接口控制	149
12.1	UART 串口输出控制	149
12.1.1	UART 发送控制	149
12.1.2	UART 接口的数据发送	151
12.2	UART 串口收发控制	153
12.2.1	UART 接口收发控制	153
12.2.2	UART 接口收发控制程序	154
12.3	UART 串口时钟显示控制	157
12.3.1	UART 接口时钟显示控制	157
12.3.2	UART 时钟显示控制程序	157
	本章小结	161
	习题	162
第 13 章	CC2530 定时器控制	163
13.1	Timer1 定时控制	163
13.1.1	Timer1 应用基础	163
13.1.2	Timer1 定时应用	164
13.2	Timer3 定时控制	166
13.2.1	Timer3 应用基础	166
13.2.2	Timer3 定时控制程序	167
13.3	Timer4 定时控制	170
13.3.1	Timer4 应用基础	170
13.3.2	Timer4 定时输出 PWM 波	172
	本章小结	174
	习题	174
第 14 章	片内 ADC 测温	176
14.1	片内 ADC 测温寄存器	176
14.1.1	片内 ADC 测温基础	176
14.1.2	片内 ADC 测温寄存器	177
14.2	片内 ADC 测温应用	180
14.2.1	片内 ADC 测温程序	180

14.2.2 数据校准	183
本章小结	185
习题	186
第 15 章 Z-Stack 协议通信	187
15.1 Z-Stack 协议简介	187
15.1.1 Z-Stack 协议的概念	187
15.1.2 Z-Stack 协议的工作流程	189
15.1.3 OSAL 中的任务处理	189
15.2 Z-Stack 无线自组网	190
15.2.1 SampleApp 工程源码	190
15.2.2 SampleApp 工程分析	196
15.2.3 网络管理	199
15.3 Z-Stack 串口通信	201
15.3.1 任务描述	201
15.3.2 程序设计	201
15.3.3 Z-Stack 串口通信分析	203
本章小结	206
习题	206
第 16 章 Z-Stack 点对点无线测温	208
16.1 无线测温基础	208
16.1.1 点对点通信结构	208
16.1.2 射频数据包格式	209
16.1.3 点对点无线测温网络基础	210
16.2 无线测温协调器程序	210
16.2.1 GenericApp 头文件	210
16.2.2 GenericApp 应用程序	211
16.3 无线测温终端设备程序	213
16.3.1 传感器驱动程序源码	213
16.3.2 终端设备测温程序源码	214
本章小结	217
习题	217
第 17 章 Z-Stack 组播与广播	219
17.1 组播与广播通信	219
17.1.1 组播与广播的概念	219
17.1.2 组播与广播的结构设计	220
17.1.3 组播与广播的应用	224

17.2 组播的应用实践	224
17.3 广播的应用实践	227
本章小结	228
习题	229
第 18 章 Z-Stack 协议组网	231
18.1 Z-Stack 协议网络拓扑控制	231
18.1.1 Z-Stack 组网控制	231
18.1.2 PAN ID 与网络地址	234
18.1.3 Z-Stack 组网实践	238
18.2 网络传感器测量原理	243
18.2.1 Z-Stack 模板结构与功能	243
18.2.2 SimpleApp 工程测量原理	243
18.2.3 传感器测量程序分析	244
18.3 Z-Stack 网络管理	253
18.3.1 网络状态指示	254
18.3.2 网络管理器	254
18.3.3 网际通信	256
本章小结	258
习题	258
附录 A CC2530 射频模块的电路原理图	263
附录 B CC2530 模块的底板电路原理图	264
附录 C 无线传感器网络的缩略语	265
参考文献	267

上 篇

基于 TinyOS 2.x 操作
系统的应用设计

TinyOS 程序设计原理

无线传感器网络(WSNs)是目前国内外研究的热点之一,它是由众多具有感知、通信和计算能力的微型无线传感器网络节点自组织形成的网络通信系统。它能够协作实时监测、感知和采集网络分布区域内的各种环境或监测对象数据,并通过网络多跳路由的形式将这些数据传输给观察者(计算机、手持终端等),从而进一步分析、判断以获得所需的重要信息。

由于无线传感器节点是一种典型的资源受限的嵌入式系统,而 WSNs 网络的应用却越来越复杂化,实时处理能力的需求也在提高,传统的前后台系统越来越不能适应系统发展,需要引入操作系统来组织和管理硬件,提供基础软件库,减少软件开发难度,节约开发成本,提高效率。

TinyOS 是当前热门的无线传感器网络操作系统之一,通过高效利用无线传感器网络节点的有限资源,能帮助开发人员快速设计应用服务程序。

学习目标:

- (1) 了解 TinyOS 操作系统的基本概念;
- (2) 了解 NesC 编程语言的特点;
- (3) 了解 NesC 简单应用程序的结构。

1.1 TinyOS 操作系统

1.1.1 TinyOS 简介

TinyOS 的含义即是“微型操作系统”,由美国加州大学伯克利分校(UC Berkeley)设计,是一个无线传感器网络的专用操作系统、一个基于事件驱动的微型操作系统,也是一个开源的嵌入式操作系统。TinyOS 程序编译后占用非常小的容量(大部分在 30KB 以下),考虑到无线传感器节点的一般特点(简化的硬件体系、低存储容量、CPU 低性能及有限能量),TinyOS 操作系统非常适用于无线传感器网络。

TinyOS 最初是用汇编和 C 语言实现的,在面向无线传感器网络开发应用程序时,汇编

和 C 语言不够高效、快捷,目标代码比较长,经过进一步研发,设计出了支持组件化的 NesC 编程语言。

TinyOS 是用 NesC 语言编写的 WSNs 操作系统,在 TinyOS 中仍采用 NesC 语言进行应用程序设计。

1.1.2 TinyOS 的特点

1. 提供(provide)组件

TinyOS 系统提供一些常用组件,如传感器组件、通信组件、执行组件等。传感器组件用于采集环境数据,如湿度、亮度、温度等;通信组件则用于实现节点间互相通信;执行组件用于控制继电器、LED 指示灯、步进电机等硬件模块。

2. 使用(uses)组件

基于 TinyOS 系统开发应用程序,首先会从系统常用组件中选择相关的组件作为服务组件,从下层提供支持;然后,自定义的应用程序作为应用组件,从上层使用下层服务组件中已有的函数(命令型、事件型)资源,构建轻量级开发模式;最后,还可以扩展组件,独立编写组件模块,作为应用程序中的服务组件使用。

3. 作为操作系统的固有特点

TinyOS 是一个开源的轻量级操作系统,安装目标多是针对微型的嵌入式系统,体现出颇具优势的特点,例如低功耗、小体积、轻线程、事件驱动模式、组件化编程模式、主动消息通信技术等。

(1) 在无线传感器节点上的微控制器中,通过定时器设计分时复用系统,可以支持并发操作的执行,但是并发任务调用频繁,执行线程微小,针对这些状况 TinyOS 提出了轻线程技术。

(2) TinyOS 基于事件驱动模式进行工作。当 TinyOS 要完成某个任务时,采用事件触发去唤醒节点工作,当没有事件发生时,微处理器进入睡眠状态,从而为节点节省能量。

(3) TinyOS 系统采用层次型组件结构,如同一个网络协议栈,由硬件抽象组件(Hardware Abstractions)、合成硬件组件(Synthetic Hardware)和高层软件组件(High Level Software)组成。下层组件向上层组件通知事件的发生,上层组件调用下层组件的命令,组件之间通过接口进行连接。

(4) TinyOS 发送消息时采用主动消息(Active Message)通信技术,在发送消息的同时,传送处理这个消息的相应函数和数据;接收方收到消息后,根据相应函数和数据立即进行处理,减少系统通信量。

4. 实时性特点

基于事件驱动机制的 TinyOS 相比线程模式的 $\mu\text{C}/\text{OS-II}$ 或 $\mu\text{T-Kernel}$ 操作系统,实时性不高,但其内核占用空间极小。TinyOS 在某些场合因实时性相对较低,会出现任务过载、任务阻塞、任务队列溢出、通信吞吐量下降等一系列问题,甚至会导致系统崩溃。TinyOS 面临着改善其实时处理特性的问题,可以通过不断完善其整体架构,来提高综合处理性能。