



李外云 编著



MC2530与无线传感器网络 操作系统TinyOS应用实践



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



013067033

TP212
305

内容简介

CC2530 与无线传感器网络 操作系统 TinyOS 应用实践

李外云 编著



北京航空航天大学出版社



北航

C1674738

TP212
305

P

013087033

内 容 简 介

本书以 TI 公司 2.4 GHz ZigBee 的第二代片上系统 CC2530 为基础,由浅入深、软硬结合,全面系统地介绍了 CC2530 的各种接口和外设功能单元,以及在 TinyOS 操作系统中的应用开发。本书语言简练、浅显易懂、图文并茂、脉络清晰,以实验操作为主导,围绕 CC2530 芯片资源、外设接口和在 TinyOS 操作系统中的实际应用进行取材谋篇,着重于工程实践和技术精华的剖析以及应用设计技巧的点拨。

本书适用于高等院校信息类专业本科生、研究生和教师,也可供国内相关研究机构和公司的研发人员参考使用。

图书在版编目(CIP)数据

CC2530 与无线传感器网络操作系统 TinyOS 应用实践 /
李外云编著. -- 北京:北京航空航天大学出版社,
2013.8

ISBN 978-7-5124-1170-8

I. ①C… II. ①李… III. ①无线电通信—传感器—
网络操作系统 IV. ①TP212②TP316.8

中国版本图书馆 CIP 数据核字(2013)第 139922 号

版权所有,侵权必究。

CC2530 与无线传感器网络操作系统 TinyOS 应用实践

李外云 编著

责任编辑 张 辉 董云凤 张金伟

*

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:30.5 字数:650 千字

2013 年 8 月第 1 版 2013 年 8 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-1170-8 定价:69.00 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

随着信息采集与智能计算技术的迅速发展和互联网与移动通信网的广泛应用,大规模发展物联网及其相关产业的时机日趋成熟,欧美等发达国家已经将物联网作为未来发展的重要领域。美国将物联网技术列为在经济繁荣和国防安全两方面至关重要的技术,以物联网应用为核心的“智慧地球”计划得到了美国政府的积极回应和支持;欧盟 2009 年 6 月制定并公布了涵盖标准化、研究项目、试点工程、管理机制和国际对话在内的物联网领域十四点行动计划。

2009 年 8 月 7 日,国务院总理温家宝视察中科院无锡高新微纳传感网工程技术研发中心时发表重要讲话,提出了“在激烈的国际竞争中,迅速建立中国的‘传感信息中心’或‘感知中国’中心”的重要指示;2009 年 11 月 3 日,在《让科技引领中国可持续发展》的讲话中,温家宝总理再次提出“要着力突破传感网、物联网关键技术,及早部署后 IP 时代相关技术研发,使信息网络产业成为推动产业升级、迈向信息社会的‘发动机’”;2010 年两会期间,物联网再次成为热议话题。随着“感知中国”战略的启动及逐步展开,中国物联网产业的发展面临巨大机遇。

工信部发布的《“十二五”物联网发展规划》明确提出,到 2015 年,中国初步完成物联网产业体系构建,形成较为完善的物联网产业链,并大力发展“智能化工业、农业、物流、交通、电网、环保、安防、医疗、家居”9 大重点领域应用示范工程。根据工信部的预测,中国物联网市场到 2015 年时将逾 5000 亿元,到 2020 年成熟时将启动万亿元级别的市场规模,未来五年年均复合增长率将达到 30%。

可以预见,在不久的将来,无线传感器网络将给我们的生活带来革命性的变化。现阶段国内市场应用正在整体评估和试探阶段,但是相应的研究工作已经开展:近期主要用户为高校、研究机构(包括为交通、环境、农业、能源和国防等政府部门服务的科研院所)和企业研发中心;中长期目标用户为海量使用的政府和企业(包括交通、环境、农业、能源、国防等政府部门),将为研究工作提供平台和应用案例。到目前为止,国内外已经推出了几十种成型的无线传感器网络节点设备。

华东师范大学信息学院通信系电子信息实验教学示范中心在学校设备处 2010 年设备研制基金项目——“无线传感网实验开发平台的研制”的支持下(项目编号:521Z5017),依据目前物联网实验室建设的最新要求,吸收国内外同类产品的优点,充分考虑高校物联网实验教学的特点,精心研制了一款基于 CC2530 的支持 Z-Stack 和最新版本的 TinyOS 操作系统的 enmote 物联网开发平台。

TinyOS 操作系统是加州大学伯克利分校的 David Culler 领导的研究小组为无线传感器网络量身定制的开源的嵌入式操作系统,目前已经成为无线传感器网络中最为流行的操作系统。虽然 TinyOS 支持多种不同的硬件平台,但目前已有的硬件平台(如 mica2、micaz、telosa 和 mica2dot 等)的射频芯片大部分基于 Chipcon 公司(现已被 TI 公司收购)的 CC2420 或 CC1000,各种硬件平台的区别仅仅在于所采用的微处理器不同。如 mica2 平台采用了 Atmel 公司的 ATmega128 作为微处理器,而 telosa 平台采用了 TI 公司的 MSP430 作为微处理器。

Copenhagen 大学计算机系的 DIKU 项目组成立了 TinyOS 8051 工作组(TinyOS 8051 working group),并于 2007 年公布了 TinyOS-S 2.0 操作系统 CC2430 的移植过程,该工作组 2008 年 10 月发布了支持 CC2430 的第四版 TinyOS 平台程序“TinyOS8051wg-0.1pre4.tgz”,具体可参考 <http://www.tinyos8051wg.net/>。TinyOS 8051 目前支持 Keil、IAR 和 SDCC 编译器,实际上该项目组只对 Keil 交叉编译器进行了测试。国内许多有关 CC2430 移植的文献都源于该项目组。

作者在 TinyOS 8051 工作组移植的 CC2430 的基础上,断断续续历时 2 年,在 TinyOS 操作系统的 2.1.1-3 版本上移植了支持 CC2530 的 enmote 开发平台,移植后的平台支持 Keil 和 IAR 两种交叉编译器。

本书以移植后的 enmote 物联网开发平台为硬件基础,由浅入深、软硬结合,全面系统地介绍了 CC2530 的各种接口和外设功能单元,以及在 TinyOS 操作系统中的应用开发。

本书第 1 章简要地介绍了物联网特点、体系结构以及 802.15.4 网络通信协议标准。第 2、3 章分别介绍了 TinyOS 的安装方法和基于 Windows 操作系统的 TinyOS 集成开发环境的配置、交叉编译开发工具的使用方法。第 4 章介绍了本书所有应用程序开发的硬件平台的组成、软件编程和调试方法,为后续的开发提供了硬件基础。第 5 章简要地介绍了 TinyOS 操作系统架构、基于 TinyOS 操作系统平台的搭建以及 CC2530 移植的过程和方法,同时介绍了 nesC 语言应用程序架构、运行模型以及关键编程技术的应用。第 6、7 章详细地介绍了 CC2530 芯片的内部资源和外设接口等硬件功能模块,以及各功能模块在 TinyOS 操作系统下的驱动组件的编程方法和应用测试程序。第 8、9 章详细地剖析了 CC2530 的无线通信功能、基于 TinyOS 的主动无线通信消息机制组件的构建,并对 CC2530 无线通信的发送功率、信道选择、RSSI 以及点对点 and 点对多点无线通信组件的测试过程进行了介绍。第 10 章以光敏传感器、DS18B20 温度传感器、SHTxx 温湿度传感器和超声波传感器为例,详细地介绍了在基于 TinyOS 操作系统的物联网系统中不同类型传感器的驱动编程方法以及测试过程。第 11 章介绍了 TinyOS 操作系统的小数据分发协议和汇聚协议的基本原理、组件构成以及多跳路由协议的应用开发。

本书语言简练、浅显易懂、图文并茂、脉络清晰,以实验操作为主导,围绕 CC2530 芯片资源、外设接口和在 TinyOS 操作系统中的实际应用进行取材谋篇,着重于工程

实践和技术精华的剖析以及应用设计技巧的点拨。

本书在编写过程中参考了大量的网上资料和文献,特别是 TinyOS 和 CC2530 的官方资料,在此衷心地感谢这些资料的作者;深圳亿道信息技术有限公司上海分公司总经理石庆先生和销售总监何章龙先生在本书所使用的硬件平台和上层软件资源方面提供了无私和鼎力的支持,华东师范大学“因仑工作室”的全体师生在硬件平台设计和软件测试等方面提供了大力支持,在此一并表示真诚的感谢。

最后,尤其要感谢我的夫人和女儿,夫人在繁忙的工作外承担了更多的家务和照顾女儿的任务,才使我有更充裕的时间和精力投入工作。

由于作者水平有限,书中错误和不足在所难免,恳请广大朋友批评指正。如果您有任何意见和建议,可发邮件到 liwyyly@sohu.com 与我联系;我将在吸取大家意见和建议的基础上,不断修正和完善书中的相关内容。

作者

2013 年 8 月



第 1 章 无线传感器网络及通信标准简介	1
1.1 无线传感器网络的特点	1
1.2 无线传感器网络的网络结构	3
1.2.1 传感器节点	4
1.2.2 网络协议	4
1.2.3 网络拓扑结构	6
1.3 无线传感器网络的关键技术	7
1.4 无线传感器网络的应用	10
1.5 无线传感器网络通信标准 IEEE 802.15.4	14
1.5.1 IEEE 802.15.4 的主要特点	14
1.5.2 物理层(PHY)规范	17
1.5.3 媒介访问层(MAC)规范	19
1.6 本章小结	25
第 2 章 TinyOS 开发环境的安装与配置	26
2.1 TinyOS - 2.1. x 在 Winows 中的手动安装	26
2.1.1 安装 Cygwin 平台	26
2.1.2 TinyOS 源码和工具包的安装	30
2.1.3 TinyOS 补丁安装	32
2.1.4 TinyOS 环境参数设置	33
2.1.5 TinyOS 其他工具包的安装	34
2.2 TinyOS 自动集成安装过程	35
2.3 安装 IAR EW8051 编译器	38
2.4 开发环境的测试	42
2.5 本章小结	43
第 3 章 TinyOS 在 Windows 环境下的集成开发工具	44
3.1 Source Insight	44
3.1.1 Source Insight 软件介绍	44
3.1.2 nesC 编程语言与 Source Insight	45
3.1.3 Source Insight 的自定义菜单	47
3.1.4 建立 Source Insight 工程	55

3.1.5 Source Insight 自定义菜单的使用	58
3.2 NotePad++	63
3.2.1 NotePad++ 介绍	63
3.2.2 nesC 编程语言与 NotePad++	63
3.2.3 NotePad++ 的自定义编译菜单	64
3.2.4 NotePad++ 自定义菜单的使用	67
3.3 Crimson Editor	71
3.3.1 Crimson Editor 介绍	71
3.3.2 nesC 编程语言与 Crimson Editor	72
3.3.3 Crimson Editor 的自定义编译菜单	73
3.3.4 Crimson Editor 编辑器中的 TinyOS 程序的编译方法	75
3.4 Eclipse 的 TinyOS 插件	76
3.4.1 TinyOS 插件介绍	76
3.4.2 TinyOS 插件安装	76
3.4.3 TinyOS 插件的环境配置	77
3.4.4 Eclipse 的使用	80
3.4.5 TinyOS 程序的模块关联图	84
3.5 本章小结	87
第 4 章 enmote 物联网开发平台介绍	88
4.1 enmote 物联网硬件介绍	88
4.1.1 网关板	88
4.1.2 传感器电池节点板	93
4.1.3 射频模块	94
4.1.4 传感器模块	95
4.1.5 SmartRF04EB 仿真器	96
4.1.6 CC Debugger 仿真器	99
4.2 enmote 物联网开发平台测试	102
4.2.1 enmote 开发平台的硬件连接	102
4.2.2 创建应用程序	103
4.2.3 编译应用程序	106
4.2.4 下载、烧录应用程序	107
4.2.5 调试应用程序	110
4.3 本章小结	115
第 5 章 TinyOS 操作系统与 nesC 语言编程	116
5.1 TinyOS 操作系统	116
5.1.1 TinyOS 操作系统简介	116

5.1.2	TinyOS 技术特点	117
5.1.3	TinyOS 的体系结构	118
5.2	nesC 编程语言	119
5.2.1	nesC 简介	119
5.2.2	接口	120
5.2.3	组件	123
5.2.4	接口连接	124
5.2.5	as 关键字的使用	126
5.2.6	通用接口(Generic Interface)	128
5.2.7	通用组件(Generic Component)	129
5.3	nesC 应用程序	130
5.3.1	nesC 程序架构	130
5.3.2	nesC 程序开发步骤	133
5.3.3	nesC 程序编译过程	134
5.4	nesC 程序的运行模型	136
5.4.1	任务	136
5.4.2	同步和异步	137
5.4.3	原子与原子操作	138
5.5	TinyOS 平台的搭建	140
5.5.1	TinyOS 平台架构	140
5.5.2	TinyOS 平台搭建过程	144
5.5.3	新建平台的测试	148
5.6	CC2530 平台的移植	149
5.6.1	CC2530 平台结构分析	150
5.6.2	源码转换的 perl 脚本	152
5.6.3	编译选项文件	158
5.6.4	编译规则文件的处理过程	161
5.6.5	CC2530 底层驱动	164
5.6.6	CC2530 驱动测试程序	164
5.7	TinyOS-2.x 的启动过程	165
5.7.1	TinyOS-2.x 的启动接口	165
5.7.2	TinyOS-2.x 的启动顺序	166
5.8	本章小结	171
第 6 章	CC2530 基本接口组件设计与应用	172
6.1	CC2530 的通用 GPIO 组件	172
6.1.1	CC2530 的 GPIO 概述	172

6.1.2	GPIO 相关寄存器	174
6.1.3	TinyOS 的 GPIO 接口组件 GeneralIO	175
6.1.4	GeneralIO 接口组件的测试	178
6.2	CC2530 GPIO 中断组件	182
6.2.1	CC2530 GPIO 中断	182
6.2.2	GPIO 中断相关寄存器	183
6.2.3	TinyOS 的 GPIO 中断接口组件 GpioInterrupt	185
6.2.4	GpioInterrupt 中断组件的测试程序	189
6.3	CC2530 随机数组件	193
6.3.1	CC2530 随机数发生器	193
6.3.2	随机数发生器相关寄存器	193
6.3.3	TinyOS 的随机数组件接口	194
6.3.4	TinyOS 随机数组件的软件实现	194
6.3.5	TinyOS 随机数组件的硬件实现	195
6.3.6	TinyOS 随机数组件的测试	197
6.4	CC2530 Flash 组件	200
6.4.1	CC2530 存储器介绍	200
6.4.2	CC2530 存储器空间	200
6.4.3	CC2530 Flash 控制器	202
6.4.4	CC2530 Flash 操作的相关寄存器	203
6.4.5	CC2530 Flash 组件接口与实现	204
6.4.6	CC2530 Flash 组件的测试程序	208
6.5	CC2530 高级加密标准 AES 组件	211
6.5.1	CC2530 AES 协处理器介绍	212
6.5.2	CC2530 AES 相关寄存器	214
6.5.3	CC2530 的 AES 组件接口与组件实现	214
6.5.4	CC2530 AES 组件的测试程序	218
6.6	CC2530 DMA 组件	221
6.6.1	CC2530 DMA 介绍	221
6.6.2	CC2530 DMA 控制器	222
6.6.3	CC2530 DMA 配置结构	224
6.6.4	CC2530 DMA 中断触发源	226
6.6.5	CC2530 DMA 相关寄存器	227
6.6.6	CC2530 的 DMA 组件接口与组件实现	228
6.6.7	CC2530 DMA 组件的测试程序	233
6.7	CC2530 WatchDog 组件	236

6.7.1	CC2530 WDT 定时器介绍	236
6.7.2	CC2530 WDT 相关寄存器	237
6.7.3	CC2530 WDT 组件接口与组件实现	238
6.7.4	CC2530 WDT 组件的测试程序	240
6.8	CC2530 定时器组件	243
6.8.1	CC2530 定时器 1 介绍	243
6.8.2	CC2530 定时器 1 相关寄存器	246
6.8.3	TinyOS 的定时器接口	247
6.8.4	CC2530 的 TinyOS 定时器底层驱动	249
6.8.5	CC2530 定时器组件的测试	255
6.9	本章小结	260
第 7 章	CC2530 外设组件接口开发	261
7.1	CC2530 ADC 组件	261
7.1.1	CC2530 的 ADC 组件介绍	261
7.1.2	CC2530 的 ADC 操作	262
7.1.3	CC2530 的 ADC 相关寄存器	263
7.1.4	TinyOS 的 ADC 组件	266
7.1.5	ADC 组件的测试程序	270
7.2	CC2530 串口通信组件	274
7.2.1	CC2530 串口介绍	274
7.2.2	CC2530 串口相关寄存器	276
7.2.3	CC2530 串口与引脚关系	277
7.2.4	TinyOS 的串口通信接口	278
7.2.5	CC2530 串口通信组件的实现	279
7.2.6	CC2530 串口通信组件的测试程序	283
7.3	SPI 通信协议组件	288
7.3.1	SPI 通信接口介绍	288
7.3.2	SPI 总线组件的 TinyOS 底层驱动	290
7.3.3	LCD 驱动接口与组件	293
7.3.4	SPI/LCD 组件的测试程序	297
7.4	I2C 通信协议组件	299
7.4.1	I2C 协议标准介绍	299
7.4.2	I2C 总线组件的底层驱动	301
7.4.3	I2C 总线中间层驱动组件	306
7.4.4	I2C 总线组件的测试程序	308
7.5	本章小结	312

第 8 章 CC2530 射频通信组件设计	313
8.1 CC2530 射频模块	313
8.1.1 CC2530 射频模块介绍	313
8.1.2 IEEE802.15.4 帧格式	314
8.1.3 CC2530 射频发送模式	316
8.1.4 CC2530 射频接收模式	317
8.1.5 CC2530 射频中断	321
8.1.6 CC2530 射频频率和通道	324
8.1.7 CC2530 射频调制格式	324
8.2 TinyOS 通信接口和组件	326
8.2.1 message_t 消息结构体	326
8.2.2 基本通信接口	327
8.2.3 主动消息接口	328
8.2.4 ActiveMessageC 通信组件	329
8.3 CC2530 射频驱动控制接口和组件	330
8.3.1 CC2530 Packet 接口与实现组件	330
8.3.2 CC2530RFControl 接口与实现组件	333
8.3.3 CC2530 射频中断接口和组件	337
8.4 CC2530 射频数据接收和发送	341
8.4.1 CC2530 的 CSP 协处理器	341
8.4.2 CC2530 的立即执行选通命令	341
8.4.3 CC2530 的射频数据发送操作	342
8.4.4 CC2530 的射频数据接收操作	346
8.5 本章小结	349
第 9 章 CC2530 射频通信组件应用	350
9.1 点对点通信	350
9.1.1 主动消息组件 ActiveMessageC	350
9.1.2 点对点通信实例	351
9.1.3 点对点通信下载测试	356
9.1.4 点对点消息包的捕获	357
9.2 点对多点通信	361
9.2.1 点对多点通信概念	361
9.2.2 点对多点通信实例	362
9.2.3 点对多点通信下载测试	366
9.2.4 点对多点消息包的捕获	367
9.3 CC2530 通信信道设置	368

9.3.1	CC2530 的通信信道	368
9.3.2	CC2530 的通信信道定义	369
9.3.3	CC2530 的通信信道静态设置	370
9.3.4	CC2530 的通信信道动态设置	373
9.3.5	CC2530 信道测试程序	378
9.4	CC2530 RSSI 采集	380
9.4.1	CC2530 的 RSSI	380
9.4.2	CC2530 的 RSSI 获取接口函数	382
9.4.3	CC2530 的 RSSI 采集程序	383
9.5	CC2530 发送功率的设置	388
9.5.1	CC2530 的发送功率	388
9.5.2	CC2530 发送功率的设置方法	389
9.5.3	CC2530 发送功率的静态设置	389
9.5.4	CC2530 发送功率的动态设置	391
9.5.5	CC2530 发送功率测试程序	394
9.6	本章小结	396
第 10 章	TinyOS 传感器节点驱动与应用	397
10.1	SHTxx 温湿度传感器	397
10.1.1	SHTxx 介绍	397
10.1.2	SHTxx 接口说明	398
10.1.3	测量值的转换	400
10.1.4	温湿度传感器节点	401
10.1.5	SHTxx 传感器的 TinyOS 驱动	401
10.1.6	SHTxx 传感器驱动测试	406
10.2	DS18B20 温度传感器	410
10.2.1	DS18B20 介绍	410
10.2.2	DS18B20 操作命令	411
10.2.3	DS18B20 应用电路	412
10.2.4	DS18B20 传感器节点电路	413
10.2.5	DS18B20 的 TinyOS 驱动程序	413
10.2.6	DS18B20 传感器驱动测试	418
10.3	光敏传感器	421
10.3.1	光敏电阻介绍	421
10.3.2	光敏传感器节点	422
10.3.3	光敏传感器驱动程序	423
10.3.4	光敏传感器驱动测试	425

10.4 超声波测距传感器	428
10.4.1 超声波测距原理	428
10.4.2 HC-SR04 超声波测距模块	429
10.4.3 超声波传感器节点	429
10.4.4 超声波传感器的 TinyOS 驱动	430
10.4.5 超声波传感器驱动测试	433
10.5 本章小结	436
第 11 章 TinyOS-2.x 网络协议与应用	437
11.1 分发协议	437
11.1.1 分发协议介绍	437
11.1.2 分发协议接口与组件	438
11.1.3 分发协议实例测试	441
11.2 汇聚协议	445
11.2.1 汇聚协议介绍	445
11.2.2 汇聚协议接口与组件	446
11.2.3 CTP 协议	448
11.2.4 CTP 实现	451
11.2.5 CTP 协议实例测试	453
11.3 多跳路由协议应用	461
11.3.1 多跳路由的根节点程序	461
11.3.2 多跳路由的传感器节点程序	465
11.3.3 多跳路由的数据采集程序	470
11.4 本章小结	472
参考文献	473

第1章

无线传感器网络及通信标准简介

无线传感器网络(Wireless Sensor Networks, 简称 WSN)是当前国际上备受关注的多学科高度交叉、知识高度集成的前沿热点研究领域。它综合了传感器、嵌入式计算、现代网络、无线通信和分布式信息处理等技术,能够通过各类集成化的微型传感器协同完成对各种环境或监测对象的信息的实时监控、感知和采集,这些信息通过无线方式被发送,并以自组织多跳的网络方式传送到用户终端,从而实现物理世界、计算世界以及人类社会这三元世界的连通。

美国商业周刊和 MIT 技术评论在预测未来技术发展的报告中,分别将无线传感器网络列为 21 世纪最有影响的技术和改变世界的技术之一。与传统无线通信网络 Ad Hoc 网络相比,WSN 的自组织性、动态性、可靠性和以数据为中心等特点,使其可以应用到人员无法到达的地方,比如战场、沙漠等。

1.1 无线传感器网络的特点

无线传感器网络由大量体积小,成本低,具有无线通信、传感和数据处理能力的传感器节点组成。所有传感器节点被布置在整个观测区域中,各个节点将所探测到的有用信息经过初步的数据处理和信息融合后通过相邻节点以接力方式传回基站,然后再通过基站以诸如有线网络方式传送给最终用户。与其他传统网络相比,无线传感器网络具有以下特点。

1. 大规模网络

传感器网络的大规模性包括两方面的含义:一方面是传感器节点分布在很大的地理区域内;另一方面是传感器节点部署很密集,在一个面积不大的空间内,密集部署了大量的传感器节点。WSN 中传感器节点的大规模部署,使 WSN 具有较高的节点冗余、网络链路冗余以及采集数据冗余,从而为整个系统提供了很强的容错能力。

此外,借助传感器节点中个别具有移动能力的节点对网络拓扑结构进行调整,可以有效地消除探测区域内的阴影和盲点,降低环境噪声,进一步提高探测的准确性。

2. 生存能力强

WSN 中密集分布在监测区域的大量传感器节点地位平等,没有严格的控制中心。任何一个传感器节点都可以随时加入或离开网络,而不会影响整个网络的正常运行。当某些传感器节点由于环境干扰或人为破坏而不能正常工作时,随机分布的其他大量传感器节点之间可以协调互补,动态连接成新的网络系统,从而保证部分传感器节点的损坏不会影响到全局任务。因此,WSN 在恶劣的战场环境下有着很强的生存能力。

3. 精确性和可靠性高

由于 WSN 可以在监测区域大量布设低成本的传感器节点,使得传感器节点能够与探测目标近距离接触,极大地消除了环境噪声对系统性能的影响。通过多种传感器的混合应用,可以在提高探测性能指标的同时从不同空间视角对监测对象进行监测,而多节点的联合和多方位信息的综合能够有效地提高信噪比,形成覆盖面积较大的实时探测区域,从而提高监测准确性。

4. 自组织能力强

WSN 是一个节点对等网络,每个节点都具有路由功能,网络中不存在严格的中心控制节点。其工作的展开不依赖于任何预设的网络基础设施,节点开机后就可以自我协调、自动布置,快速、自动地组成一个独立的网络。

5. 可扩展性强

WSN 是一个动态的网络,节点随时可能因为种种原因退出或加入网络。此时,原有的 WSN 可以有效地剔除或容纳变化的节点,快速形成新的网络并继续原来的工作,无需外界帮助。

6. 以数据为中心的网络

传感器网络是一个任务型的网络,脱离传感器网络谈论传感器节点没有任何意义。传感器网络中的节点采用编号标识,但是否需要节点编号取决于网络通信协议的设计。由于传感器节点随机部署,导致传感器与节点编号之间的关系是完全动态的,表现为节点编号与节点位置没有必然联系。用户使用传感器网络查询事件时,直接将所关心的事件通告给网络,而不是通告给某个确定的节点,网络在获得指定事件的信息后汇报给用户。这种以数据本身作为查询或者传输线索的思想更接近于自然语言交流的习惯。因此,通常说传感器是一个以数据为中心的网络。

由低成本、低功耗的微型传感器节点构成的以自组织模式工作的 WSN 具有如下的局限性。

7. 能量的有限性

WSN 中的传感器节点体积微小,一般依赖能量有限的电池供电。其特殊的应用领域和大规模的应用数量,决定了在使用过程中无法对其进行能量更新,一旦电池耗尽该节点便随即“死亡”。因此,在 WSN 设计过程中如何提高能量使用效率、延长节点生命周期,是需要考虑的首要因素。

8. 硬件资源的有限性

WSN 对传感器节点这种需要大规模部署的微型嵌入式系统的要求非常高,包括体积小、价格低、功耗小等。受这些要求的限制,传感器节点的计算能力、程序空间和内存空间非常有限,同时面向 WSN 的算法设计也要尽可能简单,易于在传感器节点上实现。

9. 通信能力的有限性

由于无线通信所需能量与通信距离的 n 次方成正比,所以随着通信距离的增加,WSN 中传感器节点的能耗将急剧增加。受能量因素的约束,节点必须在满足通信畅通和生命周期正常的条件下考虑提高通信距离。同时,受外界地理条件、自然环境等的影响,无线通信性能可能经常变化,会频繁出现通信中断等现象,这也使 WSN 的通信能力受到很大的限制。

1.2 无线传感器网络的网络结构

无线传感器网络是一种大规模自组织网络。在无线传感器网络中,传感器节点大量部署在感知对象区域内部或附近,这些节点通过自组织方式构成无线网络,以协作的方式感知、采集和处理网络覆盖区域中特定的信息,可以实现在任意时间对任意地点的信息进行采集、处理和分析。这种以自组织形式构成的网络,通过多跳中继方式将数据传回基站节点,最后借助汇聚节点(又称基站节点)链路将整个区域内的数据传送到远程控制中心进行集中处理。广义的无线传感器网络主要包括分布式传感器节点、汇聚节点、互联网和用户界面等。如图 1-1 所示为广义的无线传感器网络构架。

无线传感器网络中内部节点只能在很小的范围进行无线通信,汇聚节点的处理能力、存储能力和通信能力相对较强,它连接传感器网络与 Internet 等外部网络,实现两种协议栈之间的通信协议转换,同时发布管理节点的监测任务,并把收集的数据转发到外部网络上。

传感器节点通常是一个微型的嵌入式系统,通过携带能量有限的电池供电,因此,节点的数据处理、存储和通信等能力相对较弱。从网络功能上看,每个传感器节点具有数据采集、处理和收发等功能,部分节点除了进行本地信息收集和数据处理