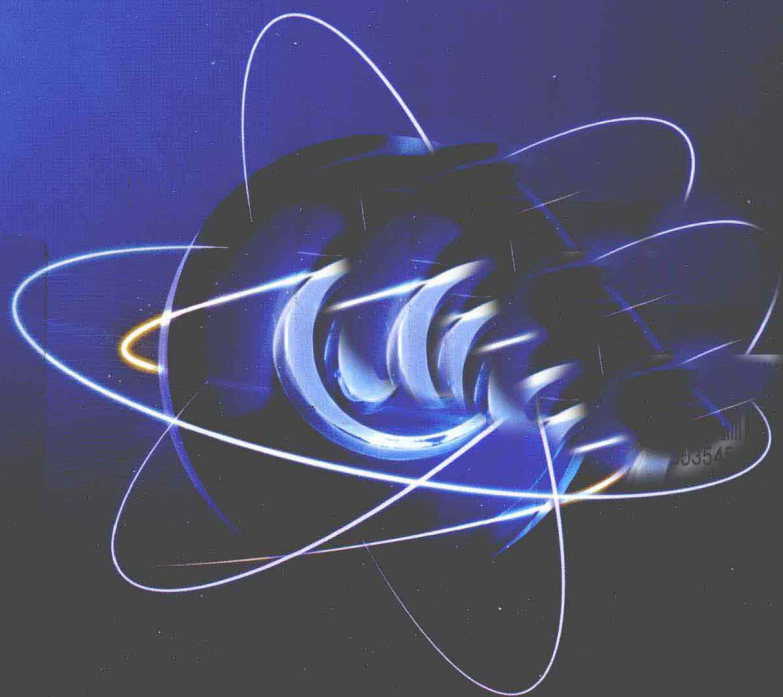


无线传感器网络实验 教程

赵东风 丁洪伟 李远壮 赵一帆 编著

WUXIANCHUANGANQI WANGLUOSHIYAN JIAOCHENG



云南大学出版社
YUNNAN UNIVERSITY PRESS

无线传感器网络实验 教程

赵东风 丁洪伟 李远壮 赵一帆 编著

 云南大学出版社

图书在版编目 (CIP) 数据

无线传感器网络实验教程/赵东风等编著. —昆明:
云南大学出版社, 2010

ISBN 978 - 7 - 5482 - 0278 - 3

I. ①无… II. ①赵… III. ①无线电通信—传感器—
实验—教材 IV. ①TP212 - 33

中国版本图书馆 CIP 数据核字 (2010) 第 219304 号

无线传感器网络实验教程

赵东风 丁洪伟 李远壮 赵一帆 编著

策划编辑: 徐 曼

责任编辑: 徐 曼 朱光辉

封面设计: 周 旻

出版发行: 云南大学出版社

印 装: 昆明市五华区教育委员会印刷厂

开 本: 787mm × 1092mm 1/16

印 张: 14.25

字 数: 347 千

版 次: 2011 年 1 月第 1 版

印 次: 2011 年 1 月第 1 次印刷

书 号: ISBN 978 - 7 - 5482 - 0278 - 3

定 价: 32.00 元

地 址: 云南省昆明市翠湖北路 2 号云南大学英华园

邮 编: 650091

发行电话: 0871 - 5033244 5031071

网 址: <http://www.ynup.com>

E-mail: market@ynup.com

前 言

无线传感器网络 (wireless sensor network, WSN) 是由部署在监测区域内数量庞大的廉价的微型传感器节点组成, 在无线通信系统协议控制下形成一个多跳的自组织网络系统, 实现区域内协同感知、采集与处理目标信息, 最终将这些信息汇聚到数据中心。

无线传感器网络起源于 20 世纪 90 年代末期。进入新世纪后, 无线传感器网络在理论研究和应用实践方面都有了很大的发展, 因而也引起了学术界、军界和工业界的广泛关注。科学研究与企业的倾情投入, 使得 WSN 这一融合多种技术并以应用为特征的网络系统, 快速转化为应用产品, 成功范例举不胜数。无线传感器网络涉及传感器技术、网络通信技术、无线传输技术、嵌入式计算技术、分布式处理技术、微电子制造技术、软件编程技术等多学科交叉的研究领域, 具有鲜明的跨学科研究特点。无线传感器网络极大地扩展了现有网络的功能和人们认识世界的能力, 它被美国商业周刊和 MIT 技术评论评为 21 世纪最有影响的 21 项技术和改变世界的 10 大技术之一。

无线传感器网络的研究具有很强的实验科学的特点。传统的互联网研究已经具有很长的历史, 利用已有的基础化设施, 为新的研究和教学搭建实验环境相对比较容易。但无线传感器网络是一种全新的网络技术, 与传统的互联网差别较大, 因此可继承的资源十分有限。因此, 在教学实践中开设 WSN 的实验教学课程就具有很强的挑战性, 需要在传统互联网的教学基础上, 结合多学科的优势特点, 构建新的实验环境, 设计新的实验内容。

本书采用宁波中科集成电路设计中心的 GAINZ 和 GAINSJ 节点为实验平台, 以 WinAVR 软件为开发环境, 使用 AVR Studio 软件作为试验的程序下载软件, 后台软件为自主研发的无线传感器网络分析平台和串口调试助手, 组成了一套无线传感器网络的实验系统。在这些硬件和软件的基础上, 设计出了一系列试验内容, 使读者跟随本书的阅读章节不断加深对无线传感器网络的认识和了解。

无线传感器网络的应用性突出,要深入学习了解 WSN,首先,必须掌握实验环境建立、软件开发、程序调试等基础操作。所以,在本书的第一章中,对 WinAVR 软件开发环境和 AVR Studio 下载软件进行了比较详细的介绍,并依据这两款软件的特点,分别设计了有关的试验内容,让读者能熟悉使用该专用软件。无线传感器网络还包含了嵌入式技术,GAINZ 节点所使用的处理器就是 ATmega128L 芯片。在第二章和第三章中,对 ATmega128L 芯片的结构和原理进行了介绍,并就此内容设计了 I/O 端口、中断处理、定时/计数器、模数转换器、串口通信等实验。通过这些实验,使读者对 ATmega128L 单片机的操作有一个初步的认识,熟悉嵌入式编程的方法,为后续的开发奠定基础。GAINZ 节点通信模块使用了 CC2420 芯片,CC2420 是 TI-Chipcon 公司推出的首款符合 2.4GHz IEEE 802.15.4 标准的射频收发器。在第四章中,对 CC2420 芯片进行了介绍,并在此基础上,设计了芯片启动、寄存器初始化、数据包发送、地址解析等实验,使读者对 CC2420 芯片的工作原理有一定的认识。通过这些章节的学习,读者便能比较熟练地操作 GAINZ 节点。在第五章,以 GAINZ 节点为实验平台,讨论了无线传感器网络的应用实现。并通过后台控制软件,介绍了实验数据分析方法。第六章还以支持 ZigBee 协议栈的 GAINZ 为平台介绍基于 ZigBee 协议的路由实验。

本书的章节安排旨在让读者熟悉 WSN 的开发环境,掌握嵌入式编程方法,熟悉 CC2420 芯片的功能。通过各章节方法学习和实验实践,使读者由简单到熟练,由基础到应用,全面学习 WSN 技术,最终学有所成。想必这是读者阅读本书的最大愿望,也是编著者们衷心的希望。

需要指出的是,无线传感器网络的应用还处于快速发展阶段,许多相关技术还不是十分成熟,软硬件开发平台还需要不断完善。因此,要全面介绍无线传感器网络最新技术,编写出与时俱进的实验教程,的确还存在诸多的困难,编著者也在不断研究进取。由于 WSN 发展之快,书中难免会出现一些纰漏甚至错误,希望广大读者提出宝贵的意见,我们将不断修改补充新的内容。

最后还要提到,参加本书编著的人员还有保利勇博士、杨志军博士、柳虔林博士、周冬明博士、何敏博士,还有李怀义、黄炜皓、鲁进、谭明川等,他们都为本书的完成付出了辛勤的汗水,成为本书的精彩段落。

本书编著者

目 录

第一章 无线传感器网络实验环境	(1)
1.1 GAINZ 开发平台	(1)
1.1.1 GAINZ 硬件平台	(2)
1.1.2 GAINZ 节点软件结构	(3)
1.2 处理器 ATmega128 介绍	(4)
1.3 CC2420 射频芯片介绍	(5)
1.4 WinAVR 开发环境	(7)
1.4.1 WinAVR 开发环境介绍	(7)
1.4.2 WinAVR 开发环境的安装和使用实验	(8)
1.5 AVR Studio 程序下载平台	(13)
1.5.1 AVR Studio 平台介绍	(13)
1.5.2 AVR Studio 平台的操作实验	(14)
第二章 ATmega128 单片机	(17)
2.1 ATmega128 单片机介绍	(17)
2.2 I/O 端口的结构和操作	(20)
2.2.1 I/O 端口的结构和原理	(20)
2.2.2 I/O 端口的操作实验	(24)
2.3 中断处理的原理和应用	(26)
2.3.1 中断处理的原理	(26)
2.3.2 中断处理实验	(30)
2.4 定时/计数器的原理和操作	(33)
2.4.1 定时/计数器的原理	(33)
2.4.2 定时/计数器 0 和定时/计数器 2 的操作实验	(50)
2.4.3 定时/计数器 1 和定时/计数器 3 的操作实验	(54)
2.4.4 定时/计数器的综合实验	(59)

第三章 串口通信和环境数据采集	(62)
3.1 串口通信	(62)
3.1.1 ATmega128 单片机的串口通信原理	(62)
3.1.2 基于 USART 的串口同步通信实验	(69)
3.1.3 基于 USART 的串口异步通信实验	(72)
3.2 环境数据采集	(76)
3.2.1 模数转化器 (ADC) 的结构和原理	(77)
3.2.2 基于 ADC 的环境数据采集实验 1	(81)
3.2.3 基于 ADC 的环境数据采集实验 2	(85)
第四章 CC2420 射频收发器原理和应用	(90)
4.1 CC2420 射频收发器的原理	(90)
4.1.1 CC2420 射频收发器的内部结构	(90)
4.1.2 CC2420 射频收发器的引脚结构	(91)
4.1.3 CC2420 的 IEEE 802.15.4 工作模式	(92)
4.1.4 CC2420 和处理器之间的接口	(93)
4.1.5 地址解析	(94)
4.1.6 无线电控制状态机	(94)
4.1.7 RSSI/能量检测	(95)
4.1.8 CCA (空闲信道评估)	(96)
4.1.9 频率和信道编程实现原理	(96)
4.2 CC2420 射频收发器的启动和 SPI 初始化	(96)
4.2.1 对 CC2420 射频收发器的寄存器的访问	(96)
4.2.2 SPI 原理	(97)
4.2.3 电池检测	(99)
4.2.4 CC2420 射频收发器的启动和 SPI 初始化的应用实验	(100)
4.3 CC2420 射频收发器的寄存器和 RAM 操作应用	(109)
4.3.1 RAM 访问	(109)
4.3.2 CC2420 寄存器的介绍和说明	(110)
4.3.3 CC2420 内部寄存器的初始化实验	(112)

4.4	CC2420 射频收发器发送数据包	(118)
4.4.1	IEEE 802.15.4 帧格式	(118)
4.4.2	CC2420 的标准数据包发送实验	(121)
4.5	CC2420 射频收发器的综合实验	(123)
第五章	光传感数据采集网络	(129)
5.1	无线传感器网络	(129)
5.2	IEEE 802.15.4 和 ZigBee	(130)
5.3	WSN 网络分析与管理平台 (SNAMP)	(131)
5.3.1	总体架构	(131)
5.3.2	功能简介	(132)
5.4	光传感数据采集网络的实现实验	(135)
第六章	基于 ZigBee 协议的路由实验	(147)
6.1	实验开发环境介绍	(147)
6.1.1	GAINSJ 节点开发板简介	(147)
6.1.2	集成开发环境 Jennic CodeBlocks	(149)
6.1.3	Jennic Flash Programmer	(154)
6.1.4	iSnamp-J	(155)
6.1.5	ZigBee 项目程序模版说明	(162)
6.2	ZigBee 协议栈使用指南	(163)
6.2.1	发起网络和加入网络	(163)
6.2.2	建立两个节点间的通信	(165)
6.2.3	发送数据	(166)
6.2.4	ZigBee 终端设备休眠	(166)
6.2.5	邻居表、路由表和绑定表	(167)
6.2.6	地址分配	(169)
6.3	智能灯光控制实验	(170)
6.4	大型路由环境监测实验	(193)
参考文献		(217)

第一章 无线传感器网络实验环境

1.1 GAINZ 开发平台

无线传感器网络 GAINZ 节点是基于低功耗微处理器芯片 ATmage128L, 射频部分采用 Chipcon CC2420 芯片, 扩展存储采用容量达 512KB 的低功耗 Flash 存储器。整个系统采用了通用的接口插槽, 将传感、处理和通信模块进行分离, 可以实现按照不同的应用需求进行不同的扩展。

GAINZ 节点由传感器模块、处理模块、无线通信模块和能量供应模块四个部分组成。传感器模块负责监测区域内信息的采集和数据转换, 节点提供了可扩展连接不同传感器的接口; 处理模块负责控制传感器节点的操作、存储和处理采集的数据和其他节点发来的数据, 实现数据安全、通信协议、同步定位、功耗管理、任务管理等功能; 无线通信模块负责与其他传感器节点进行无线通信, 交换控制消息和收发采集数据; 能量供应模块为传感器节点提供运行所需的电源。GAINZ 节点的总体架构如图 1-1 所示。

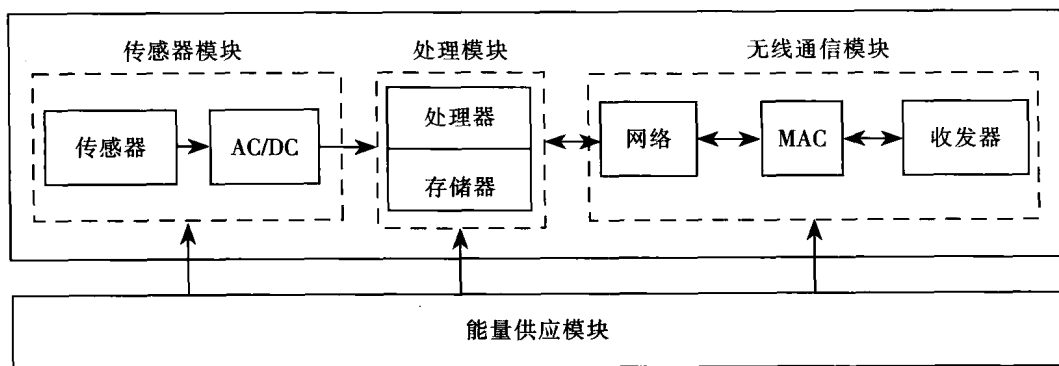


图 1-1 GAINZ 节点总体架构图

GAINZ 节点支持三种拓扑结构, 如图 1-2 中所示。星状网拓扑结构中所有节点都与基站和网关进行双向通信 (图 1-2 中 A 图)。基站可以是一台 PC、PDA、专用控制设备、嵌入式网络服务器, 或其他与高数据传输率设备通信的网关, 网络中各节点基本相同。除了向各节点传输数据和命令外, 基站还与互联网等更高层系统进行数据传输。各节点将基站作为一个中间网络点, 各节之间不传输数据或命令。

网状拓扑结构中的各节点性能相同, 相互之间可以通信, 与基站进行数据传输和命令传输 (图 1-2 中 B 图), 网状网络中的每个节点都有多条路径与网关或其他节点连接,

因此它的容错能力较强。

混合网络兼有星状网络的简洁和低功耗，以及网状网络的容错能力和自愈特点（图 1-2 中 C 图）。在混合网络中，路由节点和中继节点组成网状结构，而传感器节点则在它的周围呈星状分布，中继器扩展了网络传输距离，同时提高了系统的容错能力。由于传感器节点可与多个路由器或中继器通信，当某个中继器发生故障或某条无线链路出现干扰时，网络可在其他路由器周围进行自组织构成新的通信连接。

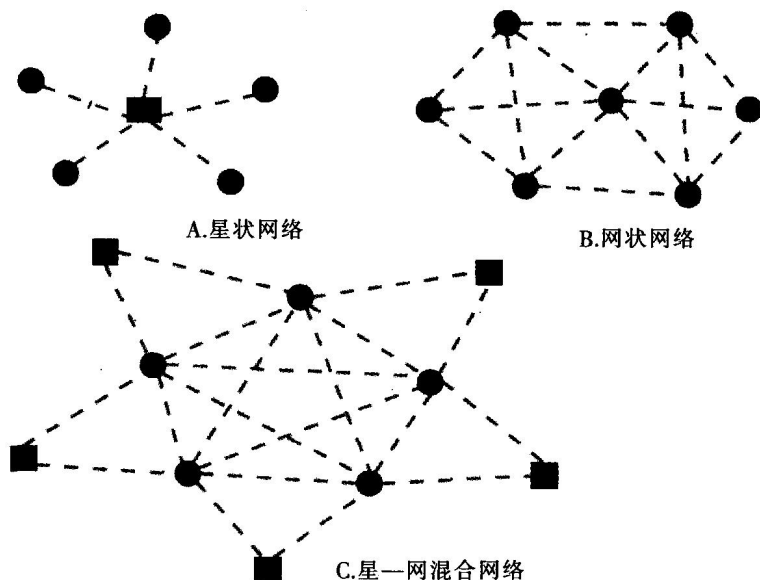


图 1-2 拓扑结构图

1.1.1 GAINZ 硬件平台

GAINZ 实验开发平台采用 ATmega128L 单片机，它是基于 AVR RISC 结构的 8 位低功耗 CMOS 微处理器。由于其采用了先进的指令集及单周期指令，ATmega128 的数据吞吐率高达 1MIPS/MHz，从而减缓了系统在功耗和处理速度之间的矛盾（如图 1-3 所示）。

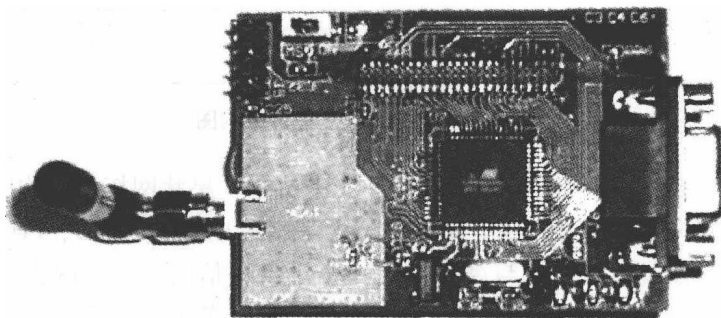


图 1-3 GAINZ 节点

GAINZ 兼容了同类节点的所有功能，并在传输距离和节点的可靠性上都有很大的提

高, 并提供了包括光传感节点、温度传感节点、压力传感节点、人体感应节点、加速度传感节点、磁场感应节点、液位感应节点等多种特定传感节点, 由于增加了传感器扩展板, 使传感部分与处理器、射频通信在物理上相分离, 节点可以更换不同的传感器, 而无需更换主要硬件电路。该节点使用了2节AA电池供电, 适合进行大范围的科学研究和需要重复利用的工作环境。

GAINZ 节点中有可以用于编程和调试的 JTAG 接口和用于动态检测数据的直接接到计算机上的串口, 调试程序和获取实验数据都十分方便。

1.1.2 GAINZ 节点软件结构

GAINZ 节点中的软件由面向 WSN 的嵌入式操作系统, 兼容 802.15.4/zigbee 的协议栈以及相应的应用程序组成。总体架构如图 1-4 所示, 分为嵌入式 OS 内核和 API 层。嵌入式内核提供了简单高效的调度、中断处理单元和时间队列管理, 还提供了能量监控单元, 以及兼容 802.15.4/zigbee 的网络协议栈, 内核底层提供了对节点上所有硬件设备的驱动。节点软件具有以下特点:

- 利用内核提供的 API, 可以快速进行上层应用开发, 搭建应用于特定场合的无线传感器网络, 既适合工程应用, 也可以用于学术研究;
- 利用串口通信 API, 可以方便地实现节点和 PC 机之间的通信, 便于用户调试;
- 利用传感采集 API, 可以实现节点对周围环境的监控, 得到传感数据;
- 利用射频通信 API, 可以轻松组成无线传感器网络;
- 利用内核提供的 API, 可以得到射频的 RSSI 值, 方便管理节点上的各种时钟资源, 可进行定位算法和时间同步等上层应用研究开发;
- 利用内核的其他单元模块, 可以方便地在节点上实现自有的通信协议方案;
- 节点软件代码采用 C 实现, 易于维护和升级。

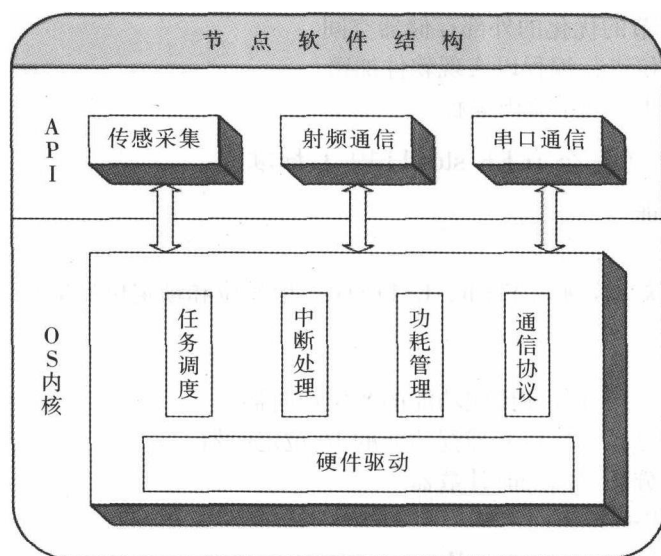


图 1-4 GAINZ 节点软件结构

1.2 处理器 ATmega128 介绍

ATmega128L 单片机是基于 AVR RISC 结构的 8 位低功耗 CMOS 微处理器。由于其先进的指令集及单周期指令系统, ATmega128 的数据吞吐率高达 1MIPS/MHz, 可以缓减系统在功耗和处理速度之间的矛盾。ATmega128 的特点如下:

1. 高性能, 低功耗的 AVR8 位微控制器

AVR 内核具有丰富的指令集和 32 个通用工作寄存器。所有寄存器都直接与运算逻辑单元 (ALU) 相连, 使得一条指令可以在一个时钟周期内同时访问两个独立的寄存器。这种结构大大提高了代码效率, 并且具有比普通的指令集微处理器高 10 倍的数据吞吐率。

2. 先进的 RISC 结构

- 133 条指令大多数在一个时钟周期内完成
- 32 × 8 通用工作寄存器 + 外设控制器
- 全静态工作
- 工作于 16 MHz 时性能高达 16MIPS
- 只需两个时钟周期的硬件乘法器

3. 非易失性程序和数据存储器

- 128KB 的系统内可编程 FLASH (具有在写的过程中还可以读的能力, 即 RWW), 可擦写 10 000 次

- 具有独立锁定位、可选择的启动代码区
- 4K 字节的 EEPROM, 可擦写 10 000 次
- 4K 字节的内部 SRAM
- 多达 64K 字节的优化的外部存储器空间
- 可以对锁定位进行编程以实现软件加密
- 可以通过 SPI 实现系统内编程

4. JTAG 接口 (符合 IEEE std. 1149.1 标准)

- 边界扫描功能
- 支持扩展的片内调试
- 通过 JTAG 接口实现对 Flash、EEPROM、熔丝位和锁定位的编程

5. 外设特点

- 两个具有独立预分频和比较功能的 8 位定时器
- 两个具有预分频、比较和捕捉功能的 16 位定时器
- 具有独立预分频器的实时计数器
- 2 通道 8 位 PWM
- 6 通道 2 到 16 位精度的 PWM
- 输出比较调节器
- 8 通道 10 位 ADC

- 8 个单端通道
- 7 个微分通道
- 2 个增益为 1x、10x 或 200x 的差分通道
- 面向字节的两线接口
- 两个可编程的串行 USART
- 可工作于主机/从机模式的 SPI 串行接口
- 具有独立片内振荡器的可编程看门狗定时器
- 片内模拟比较器

6. 特殊的 MCU 特点

- 上电复位和可编程的低电压检测
- 内部可校准的 RC 振荡器
- 外部和内部中断源
- 六种睡眠模式：空闲模式、ADC 噪声抑制模式、省电模式、掉电模式、待命模式及可扩展的待命模式
- 可以通过软件选择时钟频率
- 可以通过熔丝位选择 ATmega103 兼容模式
- 全局上拉禁止功能

7. I/O 和封装

- 53 个可编程 I/O 口线
- 64 引脚 TQFP 与 64 引脚 MLF 封装

1.3 CC2420 射频芯片介绍

CC2420 是 TI - Chipcon 公司推出的首款符合 2.4GHz IEEE 802.15.4 标准的射频收发器。该器件功能齐备完善，是一款适用于 ZigBee 产品的 RF 器件。它基于 Texas Instruments 公司的 SmartRF 03 技术，以 0.18 μ m CMOS 工艺制成，只需要极少外部器件，性能稳定且功耗极低。CC2420 的选择性和敏感性指数超过了 IEEE 802.15.4 标准的要求，可确保短距离通信的有效性和可靠性。利用芯片开发的无线通信设备支持数据传输率高达 250Kb/s，可以实现多点对多点的快速组网。其 MAC 层和物理层协议都符合 802.15.4 规范，工作于免授权的 2.4GHz 频段。利用此芯片开发的短距离射频传输系统成本低、功耗小，适于电池长期供电。此外，该芯片还具有硬件加密、安全可靠、组网灵活、抗毁性强等特点，为家庭自动控制、工业监控、传感器网络、消费电子、智能玩具等提供了理想的解决方案。

芯片的内部结构如图 1-5 所示，天线接收的射频信号经过低噪声放大器和 I/O 下变频处理后，中频信号只有 2MHz，此混合 I/O 信号经过滤波、放大、AD 转换、自动增益控制、数字解调和扩频，最终恢复出传输的正确数据。

发射机部分直接上变频。要发送的数据先被送入 128B 的发送缓冲器中，先导帧和起始帧是通过硬件自动产生的。根据 IEEE 802.15.4 标准，所要发送的数据流的每 4 个比特被 32 位码片的扩频序列扩频后送到 DA 变换器。然后，经过低通滤波和上变频的混频后，

射频信号最终被调制到 2.4GHz，并经过放大后送到天线发射出去。

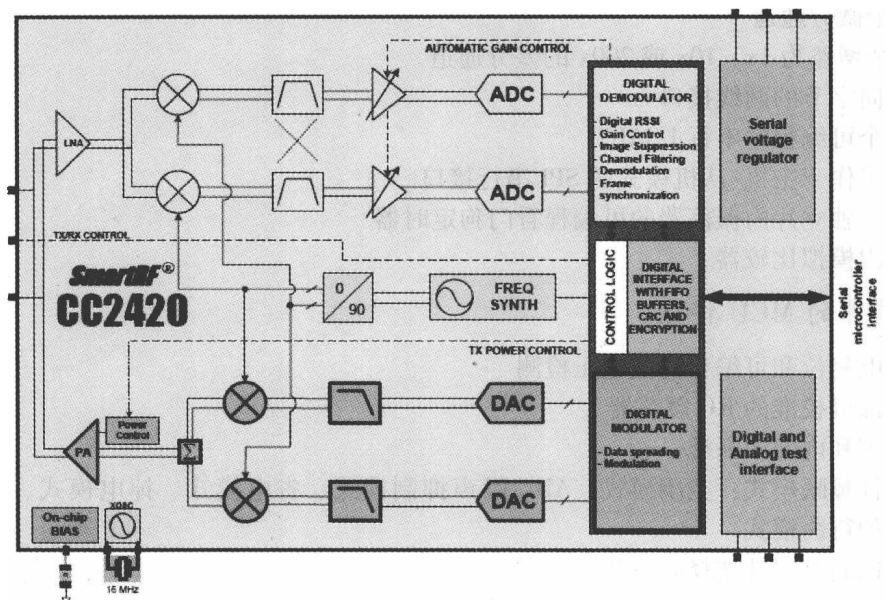


图 1-5 CC2420 内部结构图

芯片的引脚配置如图 1-6 所示：

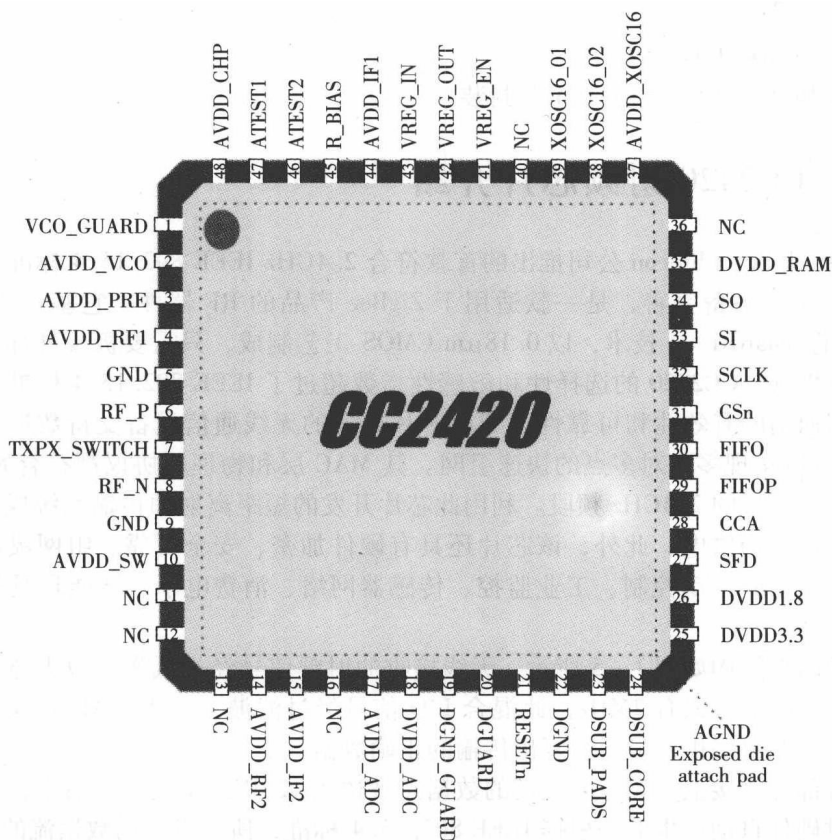


图 1-6 CC2420 引脚配置图

1.4 WinAVR 开发环境

1.4.1 WinAVR 开发环境介绍

WinAVR 是一组开放源代码的程序集，它以 GCC 为 C/C++ 编译器，用于 ATMEL 公司 AVR 系列单片机的开发。WinAVR 包含了基于 AVR 开发需用到的各种软件工具，包括：avr-gcc (compiler)，avrdude (programmer)，avr-gdb (debugger) 等。WinAVR 的开发界面如图 1-7 所示。

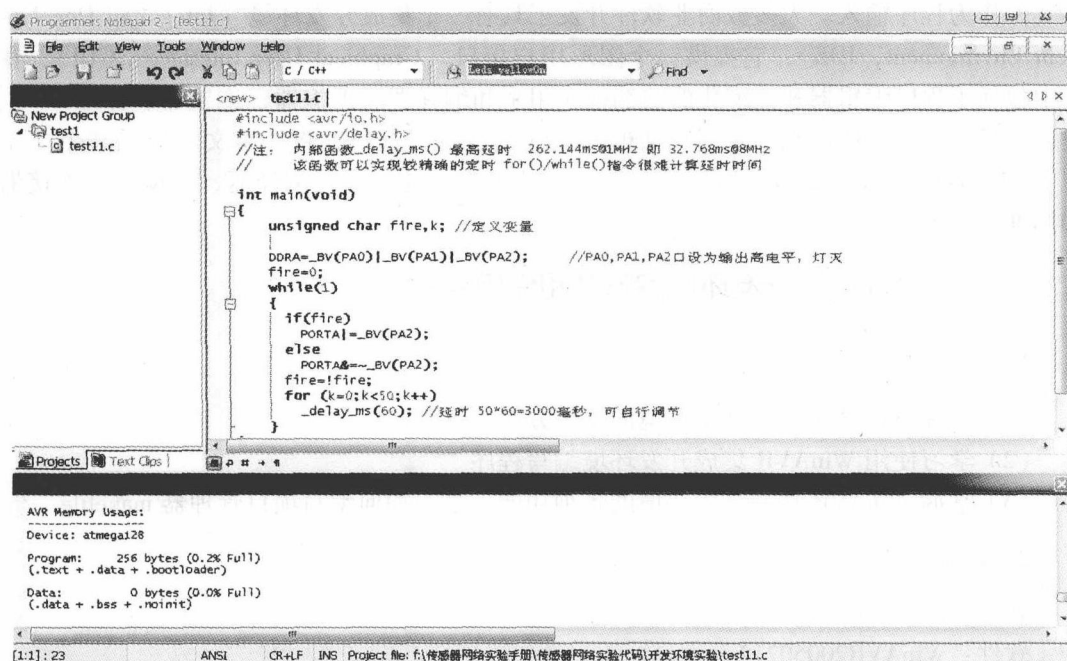


图 1-7 WinAVR 开发界面

其中，GCC 是一个编译器，它可以多种语言编译成特定处理器的汇编代码。在 WinAVR 中，GCC 可以将 C/C++ 编译成 AVR 汇编。GNU as (通常称为 gas) 是汇编器，它用来将特定处理器的汇编语言转化成目标代码。在 WinAVR 中，gas 可以将 AVR 汇编转化成目标代码。GNU ld 是连接器。在 WinAVR 中，ld 可以将编译出来的目标代码和库文件连接成最终的二进制文件。这里的二进制文件使用 ELF 文件格式。

GCC 同时还是一个“驱动”程序，它可以自动调用 gas (汇编器) 和 ld (连接器)。如果没有 gas 和 ld，GCC 不能独自生成最终的软件产品。GCC 是开源软件中的一个软件包，而 gas 和 ld 则来自开源软件的另一个独立的却又相关的 GNU Binutils (二进制工具) 软件包。在 Binutils 中还有另外一些对创建软件十分重要的程序，比如 objcopy 可以将 ELF 文件转化为 Intel HEX 或者 Motorola HEX 等其他有用的文件格式，ar 则可以管理目标代码库 (archives)，size 程序可以依据 ELF 文件或者 HEX 文件报告变量大小的信息。因为 WinAVR 二进制工具被配置成用在 AVR 上，所有这些程序的名称都被加上了“avr-”的

前缀,也就是说 WinAVR 二进制工具包括如下的一些程序: `avr - as`、`avr - ld`、`avr - ar`、`avr - size`、`avr - objcopy` 等等。

AVRLibC 是另一个开源软件,它用来给 AVR 的 GCC 编译器提供标准 C 库。在这个工程中带了许多文档,说明了如何使用 GCC 工具和 AVRLibC 库来为 AVR 开发软件。GNU make 负责把软件开发过程中用到的文件连接在一起,make 执行一个 makefile 脚本,它是你自己编写的一个文本文件,在其中列出了控制各种选择如何生成的命令行。make 经常被用来管理软件开发。

所有这些程序都是基于命令行 (Command Line Interface, CLI) 的工具,它们使用命令行中附加的参数或者开关选项来控制程序的行为,或者像 make 那样,要用户编一个文本文件作为程序输入。大多数商业软件开发工具有一个集成开发环境 (Integrated Development Environment, IDE),它提供一个图形用户接口 (Graphical User Interface, GUI),其中包括了程序编辑器和图形化的前端程序用来和编译器、汇编器、连接器、标准 C 库、建库工具接口。这些前端工具用对话框来让你设定各种选项并将一组文件加入某个“工程”(Project)。这些图形化的前端隐藏和封装了真正的命令行编译器、汇编器、连接器和标准库。

1.4.2 WinAVR 开发环境的安装和使用实验

1. 实验目的

- (1) 掌握 WinAVR 集成开发环境的安装方法
- (2) 学习使用 WinAVR 集成开发环境编写程序
- (3) 掌握 WinAVR 集成开发环境的各种功能,学习如何编写项目管理器 makefile

2. 实验设备

硬件: PC 机 (一台)

软件: WinAVR20050214

3. 实验内容

- (1) 安装 WinAVR 开发平台
- (2) 建立一个 C 程序并添加到工程中,设置 IDE (集成开发环境) 编译环境
- (3) 学习使用项目管理器 makefile,成功编译程序

4. 实验原理

WinAVR 主要包括 GNU 程序包、GNU GCC 和 AVR - LIBC 三部分。其中,GNU 程序包非常庞大,WinAVR 软件只包含 AVR 相关的部分,有 AVR 汇编器、连接器以及与机器指令相关的一些工具。GNU GCC 是该软件的编译器,GCC 需要一个叫 makefile 的参数文件才能进行编译。AVR - LIBC 是 AVR 单片机 C 运行时库,为应用程序提供标准 C 函数的连接时定位库。

WinAVR 软件没有自己的集成项目管理器,需要自己编写 makefile 文件来管理程序的编译连接。Makefile 是一个脚本文件,一个标准的可执行文件 makefile.exe 负责解析并根

据脚本内容来调用编译器、连接器和其他工具，生成可执行文件。WinAVR 开发环境有一个简单的 makefile 文件生成工具 Mfile，使用它可以方便地生成合适的 makefile。Mfile 是一个工具，可以让用户方便地修改 makefile 上的参数。

I/O 专用寄存器（SFR）被编址到与内部 SRAM 同一个地址空间，为此对它的操作和 SRAM 变量操作相似。Avr-libc 对 SFR 定义了符号，使用这些符号需要包含头文件 io.h。由于 AVR 各器件间存在同名寄存器地址不同的问题，io.h 不直接定义 SFR 寄存器宏，它根据在命令行给出的一mmcu 选项再包含合适的 ioxxxx.h 文件。在器件对应的 ioxxxx.h 文件中定义了器件 SFR 的预处理宏，在程序中采用直接对它赋值或引用的方式读写 SFR。

PORTA 为表示端口的 A 的变量，变量 **PA0...PA7** 分别对应端口 A 的 0 到 7 号引脚，各引脚的值依次为 0、1、2、3、4、5、6、7。在 GAINZ 开发板上，PA0 口与黄灯相连，PA1 口与绿灯相连，PA2 口与红灯相连。

DDRA 为端口 A 的数据方向寄存器，共 8bit，每一 bit 分别对应一个引脚的输入输出方向，若该 bit 为 1，表示该引脚为输出，反之为输入。

_BV(bit) 表示 $1 < bit$ 。

如代码：**DDRA = _BV(PA0)**；表示设置 PA0 口为输出。

PORTA = _BV(PA0)；表示设置 PA0 口为高电平（意味着黄灯灭）。

5. 实验步骤

(1) 安装 WinAVR 开发平台。

点击安装软件，选择中文简体语言，如图 1-8 所示。



图 1-8

按照默认选择，点击下一步（如图 1-9）。选择“我同意”，然后安装在指定的位置，最后选择所需要安装的组件，建议选择全部安装。