

无线网络传感器及其微型操作系统的研究^{*}

王万里, 郑扣根, 姚 翔, 吴朝晖
(浙江大学 计算机学院, 浙江 杭州 310027)

摘 要: 首先分析了无线网络传感器的基本特性, 设计了无线网络传感器硬件平台 ZUSensor; 然后提出了架构在 ZUSensor 上的微型操作系统 ZUOS 的基本设计要求; 最后总结了微型操作系统 ZUOS 的基本实现方法, 并重点分析了 ZUOS 的组件模型和通信协议。
关键词: 普适计算; 无线网络传感器; 微型操作系统
中图法分类号: TP316 文献标识码: A 文章编号: 1001-3695(2005)09-0039-04

Study on Wireless Networked Sensors and Micro Operating System

WANG Wan-li, ZHENG Kou-gen, YAO Xiang, WU Zhao-hui
(College of Computer Science, Zhejiang University, Hangzhou Zhejiang 310027, China)

Abstract: The basic characteristics of wireless networked sensors are anylized and a hardware platform ZUSensor of such sensors is designed; the fundamental requirements for the micro operating system ZUOS running on ZUSensor are discussed, especially its component model and the communication protocol.
Key words: Pervasive Computing; Wireless Networked Sensors; Micro Operating System

计算范例(Computing Paradigm) 对计算技术的发展有着至关重要的作用。综观计算机进入人类社会的历史, 可以清楚地发现计算范例已经历了主机计算(Mainframe Computing) 和桌面计算(Desktop Computing) 两个时代, 并开始跨入普适计算^[1]时代。普适计算是一种可以包含各种计算设备的计算模式。在普适计算时代, 计算机主要不是以单独的计算机设备的形式出现, 而是将嵌入式处理器、存储器、通信模块和传感器集成在一起, 以信息设备(Information Appliances) 的形式出现。

普适计算的硬件设备范围广泛, 包括传统的计算机设备以及目前应用越来越多的各种无线移动设备和智能设备。其中, 集传感器、控制器、计算能力、通信能力于一身的资源受限的嵌入式设备, 即无线网络传感器^[2] 将大量应用于普适计算领域。它们跟外界物理环境交互, 将收集到的信息通过网络传给传统的计算机。随着传感器技术、嵌入式计算技术、通信技术和半导体与微机电系统制造技术的飞速发展, 制造微型、弹性、低功耗的无线网络传感器已逐渐成为现实。

1 研究现状

目前, 国际上一些著名大学和研究机构都在积极开展各自带有研究性质的无线网络传感器项目, 如 Rockwell WINS^[3], MIT 的 μ AMPS^[4], UCLA 的 Medusa MK-2^[5] 以及 UC Berkeley 的 TinyOS^[6] 等。这些无线网络传感器硬件平台可以分为两类: 其中, UCLA 的 Medusa MK-2 和 UC Berkeley 的 TinyOS 构建在 AVR 处理器架构上; 而 Rockwell WINS, MIT 的 μ AMPS 构建在 StrongARM(现为 XScale 架构) 处理器架构上。

这两类平台分别代表了两种不同的研究方向: 基于 AVR 架构的研究倾向于普适计算的小型化方向, 重点是支持无线网络传感器的具有多跳网络通信功能的微型操作系统及其整体解决方案的研究; 基于 StrongARM 架构的研究则倾向于普适计算的中型实用化方向, 该平台更接近于现有的 PC 架构, 能够得到当前主要操作系统(如 Linux, Windows CE) 和应用软件的支持, 因而研究重点是软件功能的完善。

2 无线网络传感器 ZUSensor 的设计

2.1 无线网络传感器的特性

无线网络传感器一般集成一个低功耗的微控制器(MCU) 以及若干存储器、无线电/光通信装置、传感器等组件, 通过传感器、动臂机构, 以及通信装置和它们所处的外界物理环境交互。单个传感器的功能是有限的, 但是当它们被大量地分布到物理环境中, 并组织成传感器网络, 就可以完成强大的实时跟踪、状态监测、信息传递等功能。

无线网络传感器本身具有一些独特的特性, 使得它们和传统计算范例中的台式机或服务器有着天壤之别, 这些特性也使得应用于其上的系统软件的设计颇具挑战:

(1) 小尺寸和低功耗。设备物理尺寸和功耗直接决定了其处理能力、存储能力和互连的能力。从系统软件的角度看, 单个无线网络传感器的物理尺寸和功耗都是非常有限的, 因此必须尽可能高效地使用它们。

(2) 密集的并发操作。无线网络传感器的一个重要任务就是相互之间能快速地传递信息。一方面, 信息必须能即时地从传感器上捕获, 经过处理后向其他节点传送; 另一方面, 信息也可以从其他节点接收到, 然后通过网络转发。这些操作都具

有实时性的要求,而且并发的可能性非常大,这就迫使系统软件能正确即时地处理它们。

(3) 有限的控制层次。传统的计算系统是把与一组设备相关的并发处理分布到总线结构连接的多层次控制器上。而在无线网络传感器中,控制器的数量远比传统的系统少,并且其处理能力和外接能力也更弱,因此传感器和其他执行器件(如 RF),通常直接和主微控制器连接,且完全由主微控制器负责控制。

(4) 面向具体应用的设计。无线网络传感器主要倾向于针对具体的应用。因此它们往往只集成某个具体应用所需的硬件,不同应用也要求不同的硬件组合,这就要求系统软件的设计高度模块化,能够非常容易地将这些软硬件组合在一起。

(5) 健壮的运行要求。实际应用中的无线网络传感器可能数量巨大,且多是在无人职守的场合,因此健壮的运行能力是非常重要的,一方面要尽量增强单个无线网络传感器的可靠性;另一方面也要增强整个网络的可靠性。

2.2 ZUSensor 的结构

针对上文所描述的关于此类网络传感器的基本特性,设计了一个小型、实用的网络传感器硬件平台 ZUSensor,并以此作为研究工作的目标平台。

ZUSensor 的结构如图 1 所示,它包含一个具有内部 Flash 程序存储器、内部 EEPROM 数据存储器的微控制器,外接若干 LED,一个短距离射频器件(RF)和一个数字温度传感器。

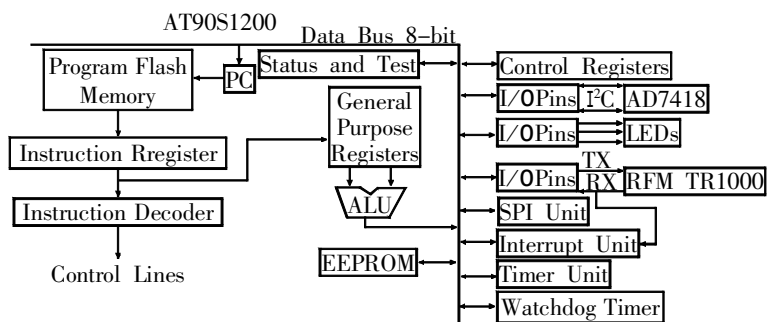


图 1 无线网络传感器 ZUSensor 的结构

ZUSensor 采用的主微控制器(MCU)是 8 位哈佛体系结构(Harvard Architecture)的 ATMEL AT90S1200^[7],16 位寻址能力,共有 32 个 8 位通用寄存器,工作频率 1MHz,工作电压 5V。AT90S1200 自带的系统内部存储空间非常有限,只有 1KB Flash 程序存储器和 64 字节的 EEPROM 数据存储器。此外,AT90S1200 还集成了通用定时器和看门狗定时器(Watchdog Timer)以及两个通用 I/O 接口(共 15 条 I/O 引脚)。

数字温度传感器(AD7418)代表了一大类数字传感器,这些数字传感器一般都有内部 A/D 转换器,以及标准通信协议接口。ZUSensor 使用了同步双线的 I²C 协议,同时主 MCU 的系统软件利用其 I/O 引脚模拟 I²C 协议与 AD7418 通信。一般情况下,最多可有 8 个不同的 I²C 设备挂接在这个串行总线上,当然多个设备之间的仲裁也必须由主 MCU 上的系统软件来完成,不过目前 ZUSensor 并没有使用这么多的 I²C 设备。

短距离射频器件(RF)是 ZUSensor 中非常重要的器件,采用 RFM 公司的 TR1000。TR1000 外围电路简单,具有睡眠模式以降低功耗,可达到 115.2 kbps 的数据传输速率,理想传输环境下有效通信距离为 1 000m。射频器件并没有缓冲,因此系统软件必须即时地处理接收或发射事件。

发光二极管(LED)用于在通用 I/O 引脚上模拟 I/O 数据的输出情况。ZUSensor 采用电池供电,电池采用 Panasonic 公司的 CR2054。CR2054 储存了 560mAh 的电,在连续收发数据的工作状态下可以供电 35 个小时,而处于休眠等节能状态下能够供电至少一年。由于电池无法支持 ZUSensor 长时间处于工作状态,因此大部分时间里 ZUSensor 的多数组件(包括主 MCU)都处于 idle 状态,而仅保留通信组件活动,以便及时接收外界信息,并唤醒主 MCU 作进一步处理。

随着研究工作的不断深入,ZUSensor 还可以进一步改进,如采用处理能力、存储能力更为强大的微控制器,增加更多的传感器件(如电池容量监控器件、无线信号强度传感器等),以提高 ZUSensor 的实际工作能力。

3 微型操作系统 ZUOS 的设计

3.1 ZUOS 的设计要求

无线网络传感器研究面临的主要问题是如何在资源受限的条件下完成感知、通信、控制和计算的工作。这些限制包括:有限的能量供应、有限的计算能力、有限的存储空间和有限的通信能力等。目前最缺少的关键技术就是系统软件对管理和操作这类设备的支持,因此支持无线网络传感器的微型操作系统^[8]的研究是无线网络传感器技术研究的核心。而且,此操作系统还必须能够充分有效地利用有限的能量、计算能力、存储空间和通信能力以完成特定的应用需求。

ZUOS 是架构于无线网络传感器(以 ZUSensor 为例)上的微型操作系统。如上所述,ZUSensor 充分体现了无线网络传感器的典型特性,而这些特性对设计其上的微型操作系统提出了挑战。我们在仔细研究 ZUSensor 硬件特点的基础上,提出了 ZUOS 的基本设计要求如下:

- (1) 精巧的核心调度模块。要求系统能在低于几百字节的指令空间和低于几十字节的数据空间内运行。
- (2) 支持事件(event)和任务(task)两级调度算法。事件优先级高,用于处理中断,可抢占任务执行;任务不具有抢占性,以先进先出的方式执行,用于计算相对集中的操作。
- (3) 支持组件化的系统功能设计。
- (4) 支持精简高效的通信协议,支持应用通信安全策略。
- (5) 支持传感器节点间的对等通信和广播通信模式。
- (6) 支持基于事件的异步通信处理模式。

3.2 ZUOS 的结构

ZUOS 采用组件化的设计思想,整个系统由一个精巧的调度模块(包括一组操作系统原语)和若干组件构成。组件分为两种,一种以主 MCU 外接的硬件设备为单位,即一个硬件设备对应一个组件;一种以逻辑应用为单位,即一组相关的逻辑操作对应一个组件。每个组件都由以下四个部分中的一个或几个组成: 一组事件处理函数(event handlers); 若干可调度的任务(tasks); 一组命令(commands); 组件的状态(states)。

其中,事件处理函数用于表示对发生的底层硬件事件的处理,如 MCU 外部中断、定时器中断等,它可以向调度模块提交任务,但并不等待任务的执行;任务用于表示组件中计算相对集中的操作,任务可以调用自己或其他组件的命令,任务不具有

抢占性,且可以被事件处理函数抢占,但任务与任务之间是原子化的,即一个任务必须执行完之后才能执行下一个任务;命令用于执行对底层硬件的操作,是非阻塞的,且必须向调用者返回命令执行的结果(成功或失败);组件的状态用于表示组件当前的工作状态,可以被自己或其他组件所参考。ZUOS 中的组件结构如图 2 所示。

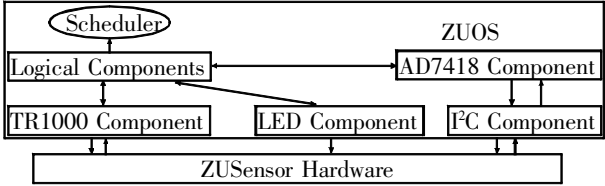


图 2 ZUOS 的结构

下面以 TR1000 组件(硬件组件)和 Message 组件(逻辑组件)的伪代码为例进行说明:

TR1000 组件有两个命令,分别是 tr1000_init 和 tx_data, tr1000_init 初始化射频器件 TR1000, tx_data 利用射频把字节数据按 bit 发送。TR1000 组件有一个事件处理函数 rx_data,它利用射频器件接收 bit 信号,进行奇偶校验并组合成字节数据,然后向调度模块提交 recv_msg 任务进行详细的消息处理,但 rx_data 并不等待此任务的执行。TR1000 组件本身并不包含任务,这是为了保持硬件组件尽可能简单,与具体的应用逻辑分开,而把其上的逻辑操作再抽象成一个新的组件,因此事件处理函数提交的 recv_msg 任务是属于 Message 组件的。需要注意的是,由于任务可以调用命令,因此为了防止被任务调用的命令再次调用事件处理函数而出现 event handler-->task-->command-->event handler 循环,在设计中不允许出现命令调用事件处理函数的情况。TR1000 组件的状态量 tr1000_state 用于记录当前射频器件的状态:发送、接收或是空闲。

```
COMPONENT TR1000{
    COMMAND: tr1000_init;
    COMMAND: tr_data;
    EVENT_HANDLER: rx_data;
    STATE: tr1000_state;
};
COMPONENT Message{
    TASK: recv_msg;
    TASK: send_msg;
};
```

逻辑组件 Message 主要负责对消息的详细处理,它只包含两个可提交的任务:recv_msg 任务负责处理系统消息接收队列中的消息,根据消息的具体含义再向调度模块提交其他任务或直接调用其他命令;send_msg 任务负责处理系统消息发送队列中的消息,它调用 TR1000 组件的 tx_data 命令把消息按 bit 发送。TR1000 组件和 Message 组件充分体现了 ZUOS 所采用的基于事件的异步通信处理模式,这样处理模式是基于如下的考虑:消息接收是非常重要的,接收动作要尽量简单、迅速;而对消息的处理属于计算相对集中的工作,可在空闲时进行。

采用组件化设计思想的好处是使得 ZUOS 的层次结构非常清晰,各组件之间只通过有限的方式联系(task 和 command)。由于无线网络传感器多面向具体应用,而不同的应用需要集合不同的组件,因此一个完善的组件库对 ZUOS 的开发是非常重要的。目前,ZUOS 包含如表 1 所示的一些组件。

表 1 ZUOS 中的组件

组 件	描 述
AD7418 组件	提供配置、读取数字温度传感器的命令:7418_ctrl 和 7418_read 以及记录当前传感器温度信息的状态量 7418_temp
LED 组件	提供控制 LED 的命令 led_on,led_off 以及记录当前 LED 开关情况的状态量 led_state
TR1000 组件	提供初始化 TR1000 和发送数据的命令 tr1000_init 和 tx_data,处理数据接收事件的处理函数 rx_data 以及一个记录 TR1000 状态信息的状态量 tr1000_state
I ² C 组件	提供利用主 MCU 的 I/O 引脚模拟 I ² C 通信协议的命令,包括 iic_start, iic_stop, iic_ack, iic_noack, iic_rbyte, iic_wbyte, ii_waddr
Message 组件	提供可提交的任务 recv_msg 和 send_msg
Temperature 组件	提供可提交的任务 read_temp,把从 AD7418 读出的 2 字节温度数据加工成 1 字节(只用低 7bit)的数据,此任务将提交 send_msg 任务发送温度数据消息

ZUOS 的调度模块非常简单,仅采用一个 FIFO 的任务执行队列,即所有的任务优先级都是相同的。各组件通过操作系统原语 post_task 向执行队列添加任务,然后由调度模块从中顺序执行。随着硬件的发展和具体应用的需要,也可以设计更为复杂的基于优先级的任务调度模块,当然这也对主 MCU 的处理能力、存储器的容量等提出了更高的要求。ZUOS 的操作系统原语主要用来维护 ZUOS 自身所需的数据。目前,ZUOS 有五条操作系统原语,如表 2 所示。

表 2 ZUOS 中的操作系统原语

原 语	描 述
post_task	向 ZUOS 的任务执行队列添加任务,如果队列已满则不允许提交新的任务。由于 ZUOS 中的任务都是以静态链接好的函数形式存在,因此系统维护 8 个 entry 的函数入口地址数组(每个地址为 16 bits)来记录提交的任务,此队列共需 16 个字节数据空间
add_rx_msg	向 ZUOS 的消息接收队列添加一条消息,如果消息队列已满,则最后一条消息被覆盖。队列长度为 4 个 entry
move_rx_msg	从 ZUOS 的消息接收队列中移出一条消息到寄存器,后面的消息前移,如果消息队列为空,则获得 4 字节全 0 的消息
add_tx_msg	和 add_rx_msg 类似,操作的是 ZUOS 的消息发送队列,队列长度为 2 个 entry
move_tx_msg	与 move_tx_msg 类似,操作的是 ZUOS 的消息发送队列

3.3 ZUOS 的通信协议

无线网络传感器 ZUSensor 的一个重要任务就是接收、发送无线信息。由于硬件条件的制约,传统计算范例中使用的很多通信协议在此并不实用。因此,为 ZUOS 设计一个简单、高效的通信协议^[9]显得十分必要。

ZUOS 的通信协议采用长度 4 字节的消息格式,固定消息长度使 rx_data 事件处理函数的设计更加方便。由于无线传输中干扰的情况经常存在,因此需对接收到的数据进行校验。鉴于 ZUSensor 的硬件能力,采用的是计算量较小的奇偶校验。奇偶校验分字节进行,每个字节的最高位是校验位,所以一个字节中只有 7 位才是真正的数据。消息格式如图 3 所示。

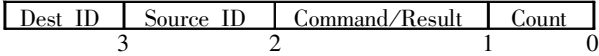


图 3 ZUOS 的消息格式

Dest ID 和 Source ID 分别是目的节点和发送节点的 ID 号,处于传感器网络中的每个 ZUSensor 节点都有一个唯一的 ID 号,ID 号保存在主 MCU 的数据存储器中,可在生产硬件时烧录进去。Command/Result 字段是消息的主要内容,消息接收者根据 Command 字段执行相应的处理,消息发送者把某些操作的结果放入 Result 字段后再发送消息。Count 字段表示消息在传感器网络的传输过程中可以经历的节点数目,消息每到

一个节点后 Count 值减 1, 直到消息到达目的节点或 Count 值减为 0 时, 消息就不再继续转发。由于需要保留一个特殊 ID 以支持广播通信模式, 因此所有可用的 ID 号数目是 125 个(即低 7 位不是全 0 或全 1), 而广播 ID 号的值是 127(即低 7 位全 1 的情况)。当传感器网络中的某一节点发送消息时, 所有处在射频覆盖范围内的节点都会收到此消息, 并调用各自的事件处理函数 rx_data 接收并组合消息, 然后提交 recv_msg 任务对消息进行处理, 若允许提交新的 recv_msg 任务, 则再通过原语 add_rx_msg 向系统消息接收队列添加消息。而 recv_msg 任务则通过原语 move_rx_msg 从消息队列中移出一个消息进行处理, 当 recv_msg 任务在处理一个消息时, 会出现以下情况:

(1) 如果是全 0 消息, 则不作任何处理。

(2) Dest ID 表明这是一个广播消息, 则 recv_msg 任务提交 send_msg 任务转发这一消息, 为避免广播风暴, Count 值也需要减 1, 即广播消息也是有范围限制的。

(3) Dest ID 等于自己的 ID, 则说明这个消息是发送给自己的, 则不再提交 send_msg 任务进行消息转发, 而根据其中的 Command 字段进行具体处理。

(4) Dest ID 不等于自己的 ID, 说明这个消息不是发送给自己的, 递减消息中的 Count 字段, 若不为 0, 则提交 send_msg 任务转发消息, 否则不再进行消息转发。

ZUOS 的消息发送则相对简单: send_msg 任务只需判断原语 move_tx_msg 获得的是否是全 0 消息, 对于正常的消息, send_msg 调用 TR1000 组件中的 tx_data 命令把消息按 bit 发送出去。由于 ZUSensor 硬件条件的制约, ZUOS 的通信协议是十分简单的, 如 ZUOS 只能维护分别为四个 entry 的消息接收队列和两个 entry 的消息发送队列, 因此会出现消息覆盖而导致消息无故丢失的情况, 另外由于主 MCU 计算能力的不足以及没有足够的存储空间维持一张路由表, ZUOS 的通信协议没有路由的能力, 这也会导致传感器网络之间大量的转发操作。

3.4 ZUOS 评价

对照 2.1 节中关于无线网络传感器的基本特性以及 3.1 节中对 ZUOS 提出的基本设计要求, 对 ZUOS 作一评价:

(1) 整个 ZUOS 仅占用 898 字节的程序空间和 60 字节的数据空间。

(2) 采用 FIFO 的任务调度队列, 调度模块非常精简, 且支持事件和任务两级调度, 事件的优先级比任务高, 可以抢断任务的执行。

(3) 采用组件化的设计方法, 使得 ZUOS 的层次结构非常清晰, 并有助于在长期开发的过程中形成一套组件库, 这样在针对不同应用时只需组合适当的组件即可生成适用于不同应用的 ZUOS。

(4) 采用了简单的通信协议, 支持对等通信和广播通信模式。利用事件和任务两级调度完成基于事件的异步通信处理模式, 即把消息的发送、接收和消息处理分开进行。

(5) 在无任务可以调度的情况下, 执行 sleep 指令进入主 MCU 的 power-down 模式, 以节省电能, 在这种模式下主 MCU 只在外部中断(即有消息接收到)发生时才会被唤醒。

(6) 利用主 MCU 自带的 Watchdog Timer 避免 ZUOS 因为意外情况进入死循环, 从而在一定程度上增强了单个 ZUSensor

的健壮性。

尽管 ZUOS 已经可以成功地运行在 ZUSensor 上, 但目前还是相当不成熟的。一方面, ZUOS 网络通信协议非常简单, 缺乏路由能力, 尤其是安全策略方面还需要开展进一步的研究工作, 即如何防止被窃听或冒充网络中的某个节点^[10]; 另一方面, 则是通用开发平台的不足, ZUOS 是用汇编语言编写的, 只支持静态链接的系统配置框架, 国外的一些研究机构在这一方面的工作已初有成效^[11]。

4 总结

本文以普适计算的迅速发展为背景, 在分析作为普适计算硬件平台重要组成部分的无线网络传感器的基本特性的基础上, 研究了这些特性对设计无线网络传感器系统软件的影响, 并设计了小型、实用的无线网络传感器硬件平台 ZUSensor; 总结了架构在 ZUSensor 上的微型操作系统 ZUOS 的基本设计要求, 并重点研究了 ZUOS 的整体结构, 以及组件模型和通信协议的实现方法; 概括了 ZUOS 的特点及今后研究工作的方向。

此外, 特别感谢惠普移动计算 2003 大学合作项目的资助。

参考文献:

- [1] ark Weiser. Hot Topics: Ubiquitous Computing [J]. IEEE Computer, 1993, 26(10): 71-72.
- [2] B Warneke, *et al.* Smart Dust: Communicating with a Cubic-Millimeter Computer [J]. IEEE Computer Magazine, 2001, 34(1): 44-51.
- [3] Rockwell Automation. Wireless Sensing Networks[EB/OL]. <http://wins.rockwellscientific.com/>, 2003-10.
- [4] MIT. MIT μ AMPS Project [EB/OL]. <http://www-mtl.mit.edu/research/icsystems/uamps/>, 2003-10.
- [5] UCLA. Medusa MK-2 Programming Resource [EB/OL]. <http://nesl.ee.ucla.edu/projects/ahlos/mk2/>, 2003-10.
- [6] UC Berkeley. TinyOS: An Open-source Operating System Designed for Wireless Embedded Sensor Networks [EB/OL]. <http://www.tinyos.net/>, 2003-10.
- [7] ATMEL. 8-bit AVR Microcontroller with 1K Byte of In-System Programmable Flash[EB/OL]. http://www.atmel.com/dyn/resources/prod_documents/DOC0838.PDF, 2003-11.
- [8] Jason Hill, Robert Szewczyk, Alec Woo, *et al.* System Architecture Directions for Networked Sensors [C]. Cambridge, Massachusetts, USA: Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, ACM Press, 2000. 93-104.
- [9] T Von Eicken, *et al.* Active Message: A Mechanism for Integrated Communication and Computation [C]. Gold Coast, Australia: The 19th International Symposium on Computer Architecture, ACM Press, 1992. 256-266.
- [10] C Karlof, D Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures [C]. Anchorage, AK, USA: The 1st IEEE International Workshop on Sensor Network Protocols and Applications, IEEE, 2003. 113-127.
- [11] David Gay, Phil Levis, Robert Von Behren, *et al.* The NesC Language: A Holistic Approach to Net-worked Embedded Systems [C]. San Diego, California, USA: ACM SIGPLAN Conference on Programming Language Design and Implementation, ACM, 2003. 1-11.

作者简介:

王万里(1980-), 男, 安徽芜湖人, 硕士研究生, 主要研究方向为嵌入式系统、Linux; 郑扣根(1964-), 男, 江苏镇江人, 教授, 博士, 主要研究方向为操作系统、人工智能、地理信息系统(GIS)、并行算法等; 姚翔(1983-), 男, 浙江嘉兴人, 本科生; 吴朝晖(1966-), 男, 浙江温州人, 教授, 博士生导师, 博士, 主要研究方向为分布式人工智能、网络计算、生物认证与嵌入式系统。