

基于 MC13213 的单芯片 ZigBee 平台的物理层协议研究与实现

吴瑾¹, 潘启勇², 王宜怀¹,

(1. 苏州大学 计算机科学与技术学院, 江苏 苏州 215004;

2. 常熟理工学院 物理与电子工程学院, 江苏 常熟 215500)

摘要: 采用 Freescale 的 MC13213 芯片构建了单芯片的 ZigBee 硬件平台, 阐述了物理层的基本内容, 分析了物理层的 SPI 事务协议、Modem 的工作模式等编程结构, 实现了构件化的底层硬件驱动程序和物理层数据包收发程序, 并基于构件对物理层协议进行了详细的测试, 验证了物理层功能的可靠性和稳定性。结果表明, 基于单芯片设计的 ZigBee 物理层协议稳定可靠, 易于应用到实际项目中。

关键词: ZigBee 协议; IEEE 802.15.4; 无线传感网络; MC13213; 物理层

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2010)23-0061-05

Research and implementation of physical layer protocol based on single-chip ZigBee platform using MC13213

WU Jin¹, PAN Qi Yong², WANG Yi Huai¹

(1. School of Computer Science and Technology, Soochow University, Suzhou 215004, China;

2. College of Physics & Electronic Engineering, Changshu Institute of Technology, Changshu 215500, China)

Abstract: This paper builds a single-chip ZigBee hardware platform using Freescale MC13213, describes the basic content of physical layer protocol. By analyzing the programming structure of SPI services protocols and the Modem working mode in physical layer, the design of the underlying hardware drivers and the sending/receiving programs for the physical layer data packet with the component mode are realized. Based on this component mode, the paper also do detailed test for the physical layer protocol to verify the reliability and stability of the physical layer functions. The results show that the single-chip ZigBee physical layer protocol is reliable and stable, and it can be applied in the actual projects easily.

Key words: ZigBee protocol; IEEE 802.15.4; wireless sensor networks; MC13213; physical layer

ZigBee 技术是一种近距离、低复杂度、低功耗、低数据速率、低成本的双向无线通信技术, 该技术基于 IEEE 802.15.4 标准, 由成立于 2001 年 8 月的 ZigBee 联盟提出。2004 年 12 月, ZigBee 联盟制定了 ZigBee Specification V1.0。至今 ZigBee 技术已经得到了广泛的发展和运用。目前, 我国绝大部分的 ZigBee 硬件都是由国外厂商设计和生产的。早期的 ZigBee 硬件都是微控制器(MCU)和 IEEE 802.15.4 射频芯片分离的。随着片上系统(SoC)的出现, ZigBee 硬件也发展到了在一个芯片内部集成了 MCU 和射频芯片, 如 Freescale 公司的 MC1321x, TI 公司的 CC243x, Ember 公司的 EM250 以及 Jennic 公司的 JN5121

和 JN5139 等^[1-2]。其中, Freescale 公司的 MC1321x 降低了 ZigBee 开发者对硬件射频电路的要求, 加速了 ZigBee 系统的开发, 同时具有较高的稳定性和可靠性。

为了更好地推广应用 ZigBee, 本文对物理层协议及编程方法进行了深入研究。物理层是 ZigBee 的关键技术, 完整的 ZigBee 协议包括应用层 (APL)、网络层 (NWK)、媒介接入控制层 (MAC) 和物理层 (PHY) 等。物理层通过操作底层硬件为上层提供服务接口, 因此物理层的稳定可靠关系到整个协议栈的健壮性, 是其他层设计的基础。

采用嵌入式构件化的设计方法, 可提高物理层设计

《微型机与应用》2010 年第 29 卷第 23 期

的可移植性和可重用性,目前很少有人做这方面的工作。本文采用射频片上系统(SoC)——MC13213设计了一个较通用的单芯片 ZigBee 硬件平台,分析和实现了 ZigBee 协议物理层,按照构件化的方法进行设计,并对构件进行了详细的测试,这不仅为基于物理层的简单应用提供了方法,而且为后续的 MAC 层的应用打下了基础。

1 ZigBee 物理层功能概述

ZigBee 工作在免申请的工业科学医疗频段。IEEE 802.15.4 标准中定义了两个可用的物理层:基于 2.4 GHz 频段的“短距离”实现和基于 868/915 MHz 频段的“长距离”实现,两者都使用直接序列扩频(DSSS)技术。中国目前的 ZigBee 工作频段为 2.4 GHz。

ZigBee 物理层通过射频固件和射频硬件为 MAC 层和物理无线信道之间提供了服务接入点 SAP(Service Access Point)。

IEEE 802.15.4 定义的物理层参考模型如图 1 所示。其中 PD-SAP(PHY Data Service Access Point)是物理层提供给 MAC 层的数据服务接口,PLME-SAP(Physical Layer Management Entity-Service Access Point)是物理层提供给 MAC 层的管理服务接口,RF-SAP 是由底层无线射频驱动程序提供给物理层的接口。

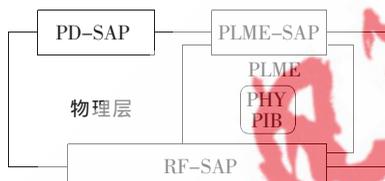


图 1 物理层参考模型

物理层主要完成以下工作:激活和禁用射频收发器,对信道进行能量检测 ED(Energy Detect),提供所接收数据包的链路质量指示 LQI(Link Quality Indication),空闲信道评估 CCA(Clear Channel Assessment),信道频率选择,数据发送和接收等。

2 MC13213 单芯片 ZigBee 物理层编程结构

Freescale 公司推出的单芯片 ZigBee 解决方案——MC13213 采用 SoC 技术,在 9 mm×9 mm 的 LGA 封装内集成了 HCS08 MCU 和遵循 IEEE 802.15.4 标准的第二代无线射频收发器 MC1320x^[3-4](后文中将用 MCU 和 Modem 分别代表 MCU 模块和射频收发器模块)。具有 4 KB 的 RAM、60 KB 的 Flash,1 个串行外设接口(SPI),2 个异步串行通信接口(SCI),1 个键盘中断模块(KBI),2 个定时器/脉宽调制 TPM(Timer/PWM)模块,1 个 8 通道 10 位的模数转换器(ADC),以及多达 32 个的 GPIO 口等^[5]。Modem 内部已经集成了功率放大器 PA(Power Amplifier)、低噪声放大器 LNA(Low Noise Amplifier)和收发开关(T/R switch),这在很大程度上降低了系统成本和射频电路的设计难度。

《微型机与应用》2010 年第 29 卷第 23 期

2.1 Modem 与 MCU 的交互方式

Modem 可以通过 SPI 接口、IRQ 中断请求以及几个状态和控制信号与主控 MCU 实现交互,如图 2 所示。

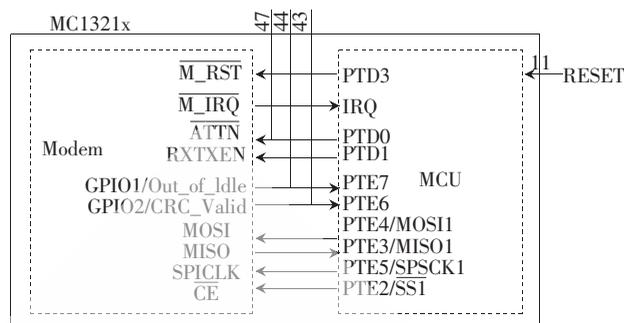


图 2 MC13213 的内部功能性互连

SPI 命令通道是 Modem 与 MCU 之间的主要交互方式,使用标准的 4 线 SPI 进行通信。MCU 通过 SPI 命令结构可以读/写 Modem 的寄存器内容、设置 Modem 的初始化参数、读取 Modem 的状态和控制信息。IRQ 中断为 Modem 提供了一种通知 MCU 有关 Modem 内部所发生事件的方法,这样就免除了 MCU 一直轮询 Modem,降低了 MCU 的运行开销。ATTN 用来把 Modem 从低功耗模式唤醒,RXTXEN 用来允许 Modem 的发送、接收和 CCA 等操作。GPIO1 引脚反映了 Modem 收发机是否忙,GPIO2 引脚可以反映所接收数据包的循环冗余校验 CRC(Cyclical Redundancy Check)是否有效或者反映 CCA 的结果^[6]。

2.2 Modem 的 SPI 事务操作

SPI 事务是在标准 SPI 协议基础上实现的一个扩展 SPI 协议。由于 Modem 中的寄存器和 RAM 大小都配置为 16 bit 即一个字(word)的宽度,所以它规定了每次 SPI 事务过程必须由 1 B 的头(header)和 2×N B 的载荷(payload)组成,每个字节对应一个 SPI 脉冲(SPI burst),其中 1≤N≤64,且为整数,代表每个 SPI 事务中所包含的字(word)数,当 N=1 时,称为 SPI 单次事务(SPI singular transaction);其他情况称为 SPI 循环事务(SPI recursive transaction)。header 的最高位为 R/W 位,表示操作类型是读还是写;header 的低 6 位是寄存器地址,表示了 SPI 操作的 64 个可能的寄存器地址(注意,有一部分寄存器没有实现)。

2.3 Modem 的数据传输模式

Modem 定义了两种数据传输模式:Stream 模式和 Packet 模式。在 Stream 模式中,数据的发送和接收是逐字(word-by-word)处理的。而在 Packet 模式中,发送时,发送方先将待发送数据缓存在 Modem 的发送缓冲区(TX RAM)中,然后再发送;接收时,接收方先在接收缓冲区(RX RAM)中缓存收到的整个数据包,然后再通知 MCU 来读取。虽然 Packet 模式下数据的接收有稍许延迟,但其降低了对 MCU 的资源要求^[7],在本协议栈实现过程中使用这种数据传输模式。

欢迎网上投稿 www.pcachina.com 69

3 物理层构件设计

基于 MC13213 单芯片的 ZigBee 平台实现物理层协议构件程序的设计, 首先必须编写底层硬件驱动程序, 然后设置 Modem 的运行方式, 再进行数据包收发程序的设计等。

3.1 底层硬件驱动程序的实现

硬件驱动程序介于底层硬件和 ZigBee 协议栈之间, 可以使得运行于硬件之上的 ZigBee 协议栈更易于维护和移植。其中芯片初始化程序对 MCU 的一些硬件模块进行正确的配置, 以保证 MCU 可以正常工作。这里所做的主要配置包括: 关闭看门狗, 设置内部时钟模块的校准(trim)值, 配置 MCU 的时钟模块等。

初始化完成后会涉及到 SPI 循环事务的实现, 下面以 Packet 模式下发送和接收 3 B 数据的完整过程来描述对 Modem RX/TX RAM 的 SPI 循环读写操作, 如图 3 所示。其中, RX/TX RAM 的长度为 128 B。图中假定 MCU 均是以字节数组的形式来保存待发送或接收到的数据。

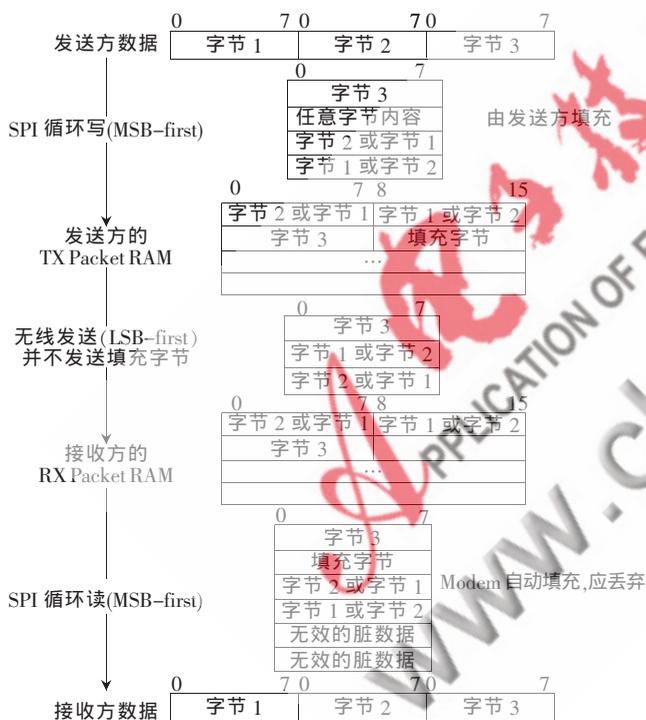


图3 读写 Modem RX/TX RAM 的详细示意图

从图3可以看出, 读/写 RAM 时的 SPI 通信是最高有效位优先(MSB-first)的, 而在无线发送/接收过程中是最低有效位优先 LSB-first(Least Significant Bit first)的, 但在编写 SPI 循环读写操作时并不需要考虑上述两种顺序, 也并不会导致接收方在接收发送方的数据时产生比特位顺序的改变。

需要特别注意的是, 由于 SPI 事务要求所有的数据传输都是按 16 bit 宽度进行的, 当发送数据是奇数个字节时, 其最后一字节数据要进行特别处理, 即需填充一

个任意字节以凑满 16 bit 宽度, 但是这个拼凑的字节和最后那个有效字节的发送顺序必须按照图 3 中的顺序进行, 即先发填充字节, 以保证在 TX RAM 中, 最后一字节紧跟在前面的偶数个字节之后。而在最后一字节数据之前的偶数个字节数据由于是 16 bit 宽度的倍数, 所以在发送每个字时对字节发送顺序没有特别要求, 只要接收方和发送方按照同一种顺序收发各字节即可。

3.1.1 使用 SPI 循环写事务向 TX RAM 中写入待发送数据

执行这个操作之前, 待发送数据长度应已经写入 TX_Pkt_Control 寄存器的 tx_pkt_length[6:0] 字段。

MCU 向 TX RAM 中写入待发送数据的一般流程如下:

(1) 根据需要配置 TX_Pkt_Control 寄存器的 tx_ram2_select 位, 以选择使用两块 TX RAM 中的一块。

(2) 计算写入待发送数据所需要的 SPI 脉冲个数, 注意:

① CRC 字节不需写入到 TX RAM 中, 它是由硬件自动产生的;

② 待发送数据的最大长度为 125 B(去掉 2 B 的 CRC);

③ 必须为偶数个字节, 若数据长度为奇数个字节, 应加 1 使其变为偶数。

(3) 做一个 SPI 循环写事务来写入数据:

① MCU 拉低 SPI 模块的片选信号 \overline{CE} , 选中 Modem;

② MCU 向 Modem 发送第一个 SPI 脉冲, 其中 R/\overline{W} 位应为 0, 表示写操作;

③ 按照(2)中计算的 SPI 脉冲个数, 写入待发送数据;

④ MCU 拉高 \overline{CE} , 使片选失效;

(4) 整个写操作结束。

3.1.2 使用 SPI 循环读事务读取 RX RAM 中的已接收数据

MCU 读取 RX RAM 中的已接收数据的一般流程如下:

(1) MCU 读 Modem 的 RX_Status 寄存器 rx_pkt_latch [6:0] 字段以获取数据长度。

(2) 计算读取 RX RAM 中的已接收数据所需要的 SPI 脉冲个数:

① 通常不读取 2 B 的 CRC, 所以数据长度应减去 2;

② 若数据长度为奇数个字节, 应加 1 使其变为偶数;

③ 按照 Modem SPI 事务协议的规定, 应丢弃读到的第一个字(word), 因为在第一次读取时, 内部 RAM 的地址还没有准备好, 这样又导致了数据长度加 2。

(3) 做一个 SPI 循环读事务来读取数据:

① MCU 拉低 SPI 模块的片选信号 \overline{CE} , 选中 Modem;

② MCU 向 Modem 发送第一个 SPI 脉冲, 其中 R/\overline{W} 位应为 1, 表示读操作;

③ 按照(2)中计算的 SPI 脉冲个数读取所有数据。注

网络与通信 Network and Communication

意,协议规定应丢弃读到的第一个字(word)。当数据为奇数个字节时,应丢弃图3中的那个填充字节;

④MCU拉高 \overline{CE} ,使片选失效。

(4)整个读操作结束。

3.2 设置 Modem 运行模式

Modem有多种运行模式,主要可分成两类:活动模式和低功耗模式。其中活动模式包括Idle模式、Receive(RX)模式、Transmit(TX)模式和CCA/ED模式;低功耗模式包括Off模式、Hibernate模式、Doze模式^[8]。

Idle模式是Modem退出任何其他模式后的默认模式,也是进入任何其他模式的初始模式;RX、TX模式分别为Modem接收、发送数据时所处的工作模式;CCA/ED模式为空闲信道评估/能量检测时所处的工作模式,用来评估信道是否空闲或测量信道的当前能量值^[8]。

收发机状态设置是通过调用设置收发机状态函数实现的,其函数头如下:

```
//-----*
//功能: 设置收发机状态函数,把收发机设置成用户期望的状态
//参数: nDesiredStatus - 用户期望状态
//返回: SUCCESS - 成功设置成指定模式;
//      等于用户期望模式-收发机之前就处于用户期望状态;
//      其他 - 表示执行失败;
//说明: 无
//-----*
INT8U PLMSetTRXState(INT8U nDesiredStatus);
```

3.3 物理层数据包的收发

物理层数据称为物理层数据单元PPDU(PHY Protocol Data Unit),包括同步包头、物理层包头和物理层载荷三部分^[3],如图4所示。其中同步包头可以使接收设备锁定在比特流上,并与比特流保持同步;物理层包头包含了数据包的长度信息,在0~127之间;物理层服务数据单元PSDU(PHY Service Data Unit)也称物理层载荷,携带MAC层的帧信息,即MAC层协议数据单元MPDU(MAC Protocol Data Unit)。注意,CRC检验字节包含在PSDU中的最后两字节中。

4 B	1 B	7 bit	1 bit	变量
前同步码	帧定界符	帧长度	预留位	PSDU
同步包头		物理层包头		物理层载荷

图4 PPDU格式

物理层数据包的发送和接收比较简单,对于MC13213的Modem来说,用户所要做的只是调用SPI循环事务写入或读取物理层包头和载荷,然后拉高RXTXEN引脚使能Modem的发送和接收即可。PPDU的同步包头是由硬件自动添加的。

(1)物理层数据包的发送

Packet模式下发送数据包的过程是在物理层数据请

求函数中实现的,在函数实现过程中,应根据要求,向上层通知数据发送结果的状态信息。其函数头如下:

```
//-----*
//功能: 数据请求函数,生成物理层协议数据单元(PPDU)并无线发送出去
//参数: nPSDULength-物理层PSDU(即MAC层的MPDU)中的字节数,
//      长度要<=aMaxPHYPacketSize(物理层最大数据包容量)
//      pPSDU - 指向物理层PSDU数据的指针
//返回: SUCCESS - 发送成功;其他值 - 发送失败
//说明: 由MAC层调用
//-----*
INT8U PDDDataRequest(INT8U nPSDULength, INT8U *pPSDU);
(2)物理层数据包的接收
一般情况下,Modem的接收机是关闭的。当Modem接收机处于打开状态时有其他Modem在同一信道上发送数据,则Modem会接收到这些数据。
本协议栈实现Packet模式下接收数据包时,为物理层数据包实现了一个环形的接收缓冲区,以保证数据的及时可靠接收,其结构定义如下:
//PSDU最大数据包长度(不包括CRC)
#define PSDUMaxLen 125
//PSDU数据包定义
typedef struct PSDURxPacket_tag
{
    INT8U m_nLen;
    INT8U m_nData[PSDUMaxLen];
    INT8U m_nStatus;
    INT8U m_nLQI;
} PSDURxPacket_t;
//PSDU数据接收缓冲区个数
#define PSDURxBufferNum 8
//PSDU数据接收缓冲区定义
typedef struct PSDURxBuffer_tag
{
    INT8U m_nPSDUCount;
    INT8U m_nHead;
    INT8U m_nTrail;
    PSDURxPacket_t m_sPSDU[PSDURxBufferNum];
} PSDURxBuffer_t;
```

//定义PSDU环形接收缓冲区

```
static PSDURxBuffer_t s_sPSDURxBuffer;
```

3.4 其他编程相关说明

空闲信道评估(CCA)用来判断信道是否空闲。能量检测(ED)用来测量目标信道中接收信号的功率强度,由于这个检测本身不进行解码操作,所以检测结果是有效信号功率和噪声信号功率之和。另外,链路质量指示

(LQI) 提供了接收数据包时无线信号的强度和信道质量信息。与能量检测不同的是, LQI 要对信号进行解码, 生成的是一个信噪比指标。这个信噪比指标和物理层数据单元一起提交给上层处理。Modem 中 RX_Status 寄存器的 cca_final[7:0] 字段保存了以上操作的结果值。

当 Modem 完成 MCU 指定的某个功能(如发送完成、接收完成、CCA/ED 完成等), 就会产生 IRQ 中断, 然后 MCU 会读取 Modem 的 IRQ 状态寄存器, 针对不同的 IRQ 中断类型分别进行处理。

物理层管理实体(PLME)维护了物理层正常工作所必须的一些属性参数, 包括物理层支持的信道列表、当前用于发送和接收的信道、物理层的发射功率以及 CCA 模式 4 个属性。每个属性都有一个唯一的属性标识符, 并且某些属性还有一些特定的取值范围。属性的读和写分别由属性设置和读取函数来实现, 由于物理层的属性较少, 直接通过 switch/case 语句实现即可。

4 物理层构件测试

按照前面分析的 ZigBee 物理层编程结构编写测试程序, 可用于对物理层的构件功能进行测试。测试可先进行 SPI 单次读写事务测试, 然后再进行数据包收发测试。

4.1 SPI 单次读写事务的正确性测试

在对 Modem 的内部寄存器初始化之前, 可利用 SPI 单次读事务获取 Modem 寄存器的内容, 并通过串口输出显示与 Modem 寄存器的复位值对比是否一致。接着利用 SPI 单次写事务, 对 Modem 进行初始化, 初始化后, 再把修改后的 Modem 寄存器的内容通过串口输出显示, 与修改值比较, 即可得出 SPI 写事务是否正确。

4.2 物理层数据包的收发测试

物理层数据包的收发测试需要一个发送节点和一个接收节点相互配合。对于能否正确收发需要测试两种情况, 发送节点分别发送奇数个和偶数个字节的数据, 看接收节点能否正确收到。这部分的测试是借助于串口调试工具来完成的, 接收节点把收到的数据发往 PC 机串口显示。

物理层数据包收发的可靠性测试条件如下: 一个发送节点和一个接收节点, 二者相距 5 m 左右, 其中发送节点每次发送长度为 20 B 的数据, 并且数据中的最后 2 B 作为一个 16 bit 的整数, 用来记录发送次数, 每发送一次其值加 1。发送节点何时开始发送数据由 PC 方测试软件控制, 接收节点负责接收数据并发给 PC 端测试软件显示, 通过比较发送字节数与接收字节数以及数据中的发送次数字段, 可以得出数据丢失情况。所有的测试数据会写入后台的 ACCESS 数据库中, 以供将来进行数据的统计分析。试验中测试的一组数据如表 1 所示, 丢包率不超过 0.1%。

本文研究了 ZigBee 协议物理层的 SPI 事务协议、Modem 的工作模式等关键技术和编程方法, 实现了基于

表 1 物理层数据包收发测试结果

发送功率 /dBm	发送间隔 /ms	发送数据 包个数	接收数据 包个数
-0.66	500	100	100
-0.66	500	1 000	1 000
-0.66	200	100	100
-0.66	200	1 000	999
3.4	500	100	100
3.4	500	1 000	1 000
3.4	200	100	100
3.4	200	1 000	1 000

单芯片 ZigBee 平台的构件化的底层硬件驱动程序和物理层数据包收发程序。测试表明, 此硬件平台稳定可靠、实现容易, 不但方便了其他研究人员学习和实践 ZigBee 技术, 降低了研究 ZigBee 技术的门槛, 还由于采用了单芯片设计和构件化设计方法, 具有较高的可移植性和可重用性, 使其很容易应用到实际项目中。同时为 ZigBee 协议栈后续其他层的研究和实现打下了基础, 并为其他协议的应用研究提供了参考。

参考文献

- [1] 刘新, 吴秋峰. 无线个域网技术及相关协议[J]. 计算机工程, 2006, 32(22): 102-103.
- [2] 吴光荣, 章剑雄. ZigBee 网络系统节点硬件设计与实现[J]. 杭州电子科技大学学报, 2008, 28(4): 49-52.
- [3] IEEE. Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate wireless personal area networks[S]. IEEE Std 802.15.4-2006, September 2006.
- [4] IEEE. Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks[S]. IEEE Std 802.15.4-2003, October 2003.
- [5] ZigBee Alliance. ZigBee specification [EB/OL]. <http://www.zigbee.org/en/spec-download>, 2007.
- [6] Freescale. MC13213: 2.4GHz RF transceiver and 8-bit MCU with 60K of flash for ZigBee applications [EB/OL]. http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13213&nodeId=0106B9869925657103, 2009.
- [7] Freescale. MC13224V: MC1322x platform in a package[EB/OL]. http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13224V&nodeId=0106B9869925657103, 2009.
- [8] Freescale. MC13213RM. pdf [DB/OL]. <http://www.freescale.com/>, 2009.

(收稿日期: 2010-07-06)

作者简介:

吴瑾, 女, 1975 年生, 硕士, 讲师, 主要研究方向: 嵌入式系统与智能控制。

潘启勇, 男, 1971 年生, 硕士, 讲师, 主要研究方向: 嵌入式系统与智能控制。

王宜怀, 男, 1962 年生, 博士, 教授, 主要研究方向: 嵌入式系统与智能控制等。

《微型机与应用》2010 年第 29 卷第 23 期