

基于ZIGBEE技术的农业信息无线数据传输

刘 原^{1,2}, 宋良图¹

(1 中科院合肥智能机械研究所 合肥, 230031)

(2 中国科学技术大学自动化系 合肥, 230031)

摘要: 介绍了以 AVR 单片机 Atmega8L 和无线传输芯片 CC2420 为核心构成的无线嵌入式传输智能控制器的设计方法, 给出了系统硬件设计和软件的实现方法。

关键词: ZIGBEE; CC2420; 全双工无线通讯; 数字化农业

Abstract: This paper introduces a design of method of intelligent controller of wireless embedded transmit which is composed of AVR Atmega8L single-chip microcomputer, it has hardware system design and software design.

Key words: ZIGBEE; CC2420; Full-duplex wireless data transmit; Digital agriculture

中图分类号: TP368.1 文献标识码: B 文章编号: 1001-9227(2006)04-0021-05

0 引言

农业信息数据传输具有需要数据采集或监控的网点多、要求传输的数据量不大且要求设备成本低、数据传输可靠性高、安全性高、设备体积很小、不便放置较大的充电电池或者电源模块、用电池供电。地形复杂、监测点多、需要较大的网络覆盖等特点。另外, 现有移动网络的覆盖盲区, 使用现存移动网络进行低数据量传输的遥测遥控系统, 效果差或成本太高。因此, 要选择一种合适的传输标准, 实现并得到最好性价比的无线传输系统。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。其工作频带范围是 2.400~2.4835GHz, 采用

IEEE802.15.4 规范要求的直接序列扩频方式, 数据速率 250kbps。并且具有超低电流消耗 (RX: 19.7mA, TX: 17.4mA), 高接受灵敏度, 抗邻频道干扰能力强的特点。同时, 它是利用 4 总线 SPI 接口的方式与微处理器通讯, 接口配置容易。Zigbee 技术的目标就是针对工业、家庭自动化、遥测遥控、汽车自动化、农业自动化和医疗护理等。因此, 本系统中我们选用 Chipcom 公司的 Zigbee 芯片 CC2420 和 AVR 单片机 Atmega8L, 组成低成本的无线农业传感器信息传输模块。

1 CC2420 芯片简介

1.1 内部结构 (如图 1 所示)

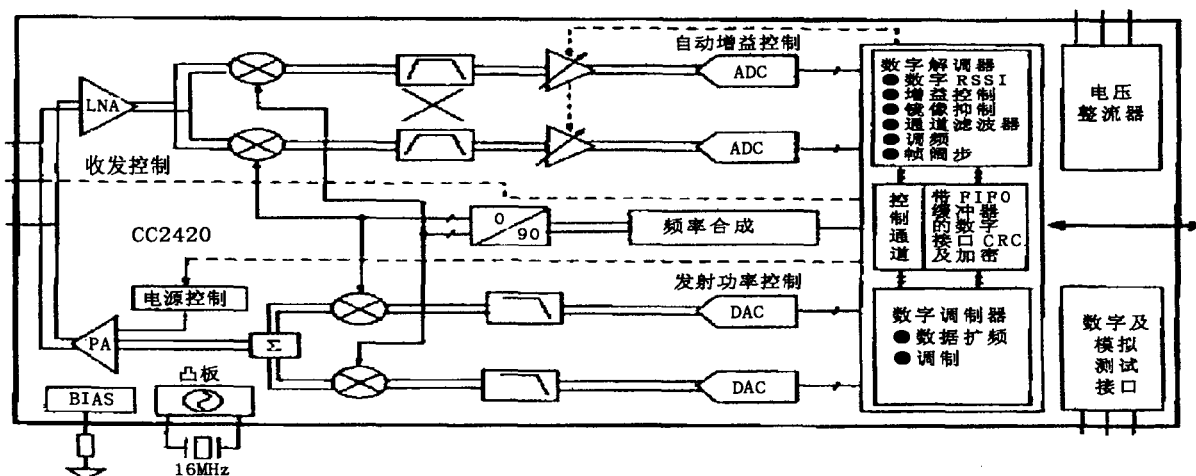


图 1 CC2420 内部结构

CC2420 芯片的内部结构如图 1 所示。天线接收的射频信号经过低噪声放大器和 I/Q 下变频处理后,

中频信号只有 2MHz, 此混合 I/Q 信号经过滤波、放大、AD 变换、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。发射机部分基于直接上变频。要发送的数据先被送入 128 字节的发送缓存器

中,头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 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的射频信号最终被调制到 2.4GHz,并经放大后送到天线发射出去。

IEEE802.15.4 规范中规定使用 DSSS 调制方式,CC2420 中的调制和扩频功能框图如图 2 所示。

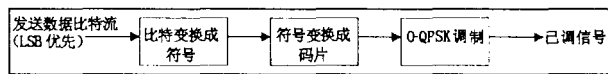


图 2 CC2420 调制和扩频功能框图

CC2420 先将要传输的数据流进行变换,每个字节被分组为两个符号,每个符号包括 4 个比特 LSB 优先传输。每个被分组的符号用 32 码片的伪随机序列表示,共有 16 个不同的 32 码片伪随机序列。经过

DSSS 扩频变换后,码片速率达到 2M/s,此码片序列再经过 O-QSK 调制,每个码片被调制为半个周期的正弦波。码片流通过 I/Q 通道交替传输,两通道延时为半个码片周期。CC2420 为 IEEE 802.15.4 的数据帧格式提供硬件支持其 MAC 层的帧格式为:头帧+数据帧+校验帧;PHY 层的帧格式为,同步帧+PHY 头帧+MAC 帧,帧头序列的长度可以通过寄存器的设置来改变。可以采用 16 位 CRC 校验来提高数据传输的可靠性。发送或接收的数据帧被送入 RAM 中的 128 字节的缓存区进行相应的帧打包和拆包操作。

2 系统外围电路的设计

系统的总体无线应用方案如图 3 所示:

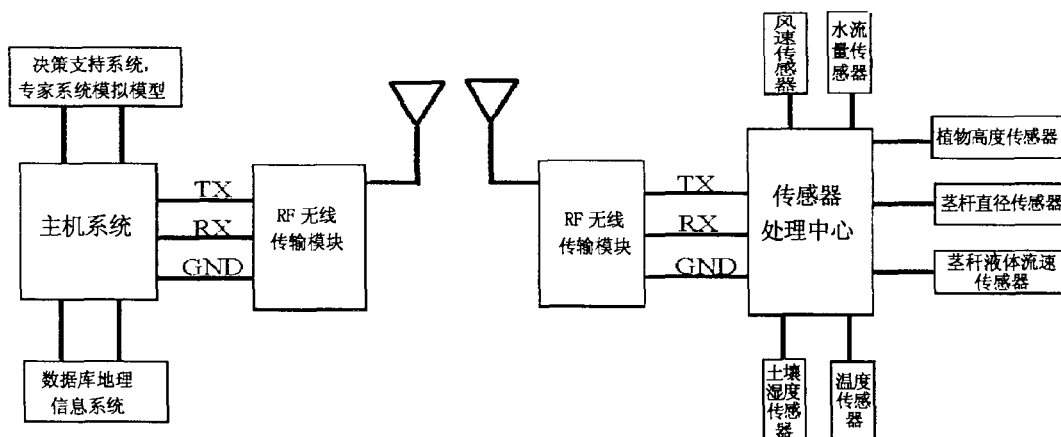


图 3 系统总体应用框图

在本系统的无线传输模块设计中,对于主机系统和下位机传感器处理中心的无线模块我们采用的是相同的硬件和软件设计。但是,通过在硬件中采用跳线的方式来选择无线模块工作在主机模式还是从机模式。这样,可以达到增强无线模块的通用性、灵活性和节约成本的目的。当系统上电复位后,单片机首先判断无线模块的主从模式,然后分别调用不同的程序块。在主机模式下,通过中断的嵌套的方式,可以让主机系统实时的来控制下位机的数据发送和接受,而在从机模式下,无线模块首先接受上位机无线模块发送过来的数据,然后对数据进行枚举后,判断出主机系统需要的实时控制的信息,再把需要应答的数据通过从机模式的无线模块发送给主机。这样,就保证了主从无线传输模块的全双工通讯。主机系统可以实时的控制数据的传输和监测下位机的工作状态,RF 模块无线传输模块的结构如图 4 所示。

图 4 所示电路是无线模块的一个结构框图。图中的单片机是 Atmegal8 系列的 AVR 单片机。RS-232 串口控制收发芯片是 MAX3243,无线射频芯片是

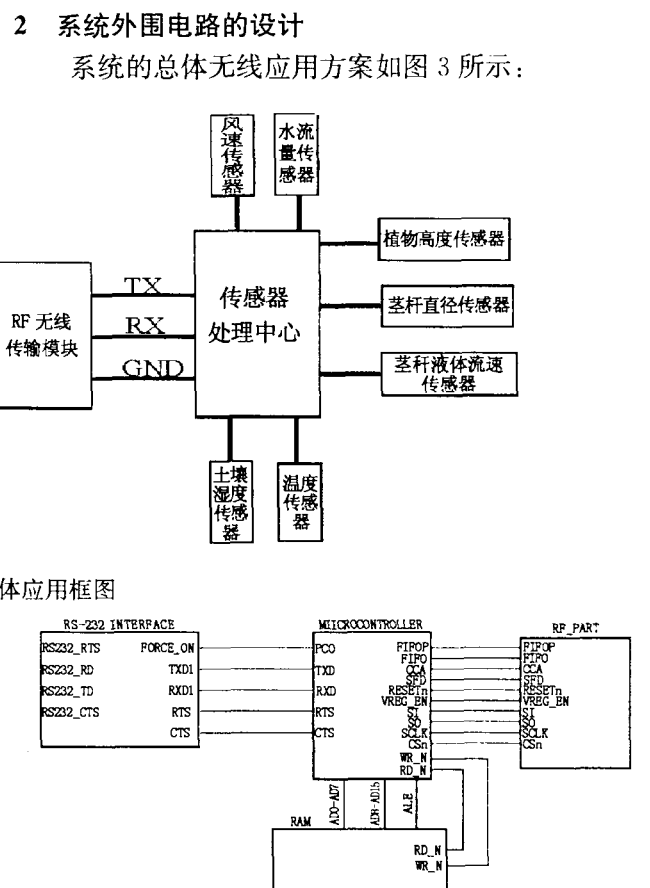


图 4 无线模块的结构图

CHONIC 公司的 CC2420 芯片。另外,还有 32kBytes 的外部数据存储器,用来存放临时数据,可以随时读出或者上传给上位机。单片机内部有 4Kb 的 Falsh 内存和 4kb 的 SDRAM。单片机通过 SPI 接口和 CC2420 进行通信。RS232 部分与计算机的串口或者是 GRPS 的串口相接,把 RF 模块需要发送和接受的数据通过单片机作出相应的处理。单片机 RF 部分分别是由 8M 和 16M 的晶振提供振荡源。由于 RF 模块是低电压工作模块,所以外部电源提供的电压不能超过 3.6V。

CC2420 只需要极少的外围元器件,它的外围电路包括晶振时钟电路、射频输入/输出匹配电路和微

控制器接口电路三个部分，芯片本振信号既可由外部有源晶体提供，也可由内部电路提供。由内部电路提供时需外加晶体振荡器和两个负载电容，电容的大小取决于晶体的频率及输入容抗等参数。例如当采用 16MHz 晶振时其电容值约为 22pF。射频输入/输出匹配电路主要用来匹配芯片的输入输出阻抗，使其输入输出阻抗为 50Ω，同时为芯片内部的 PA 及 LNA 提供直流偏置。

3 系统的软件设计

本系统用 C 语言编程，软件采用模块化结构程序设计方式，软件模块包括系统初始化、RF 芯片寄存器配置和模式设定、RAM 的读写、RF 接受和发送等子模块。根据本系统的特点，每一对无线模块在电路上通过跳线设置好主从机设置后，程序内部自动实现模式的配置，总框图如图 5 所示：

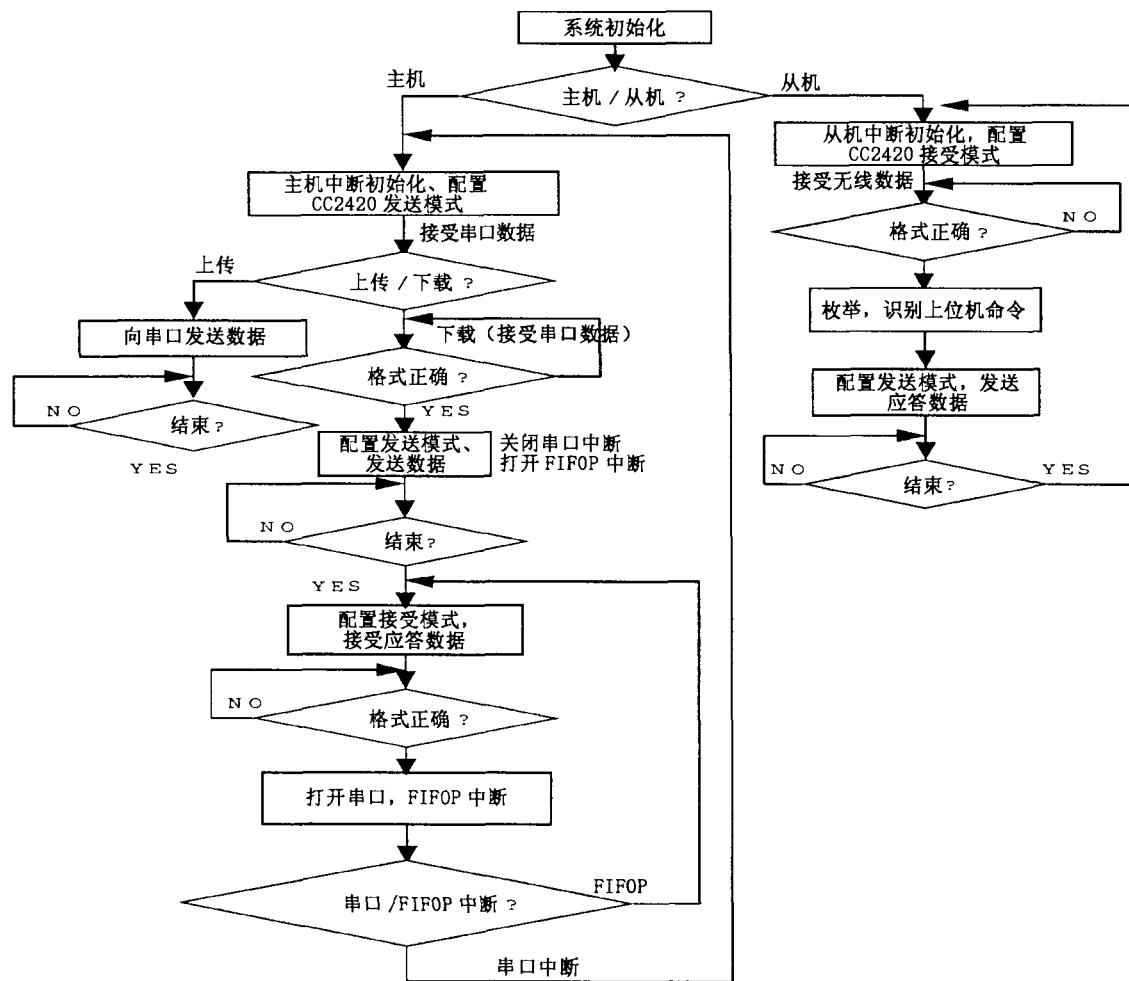


图 5 系统软件设计

3.1 系统初始化

```

void SystemInit(BASIC_RF_RXINFO * pRRI,
UNIT8 channel ,WORD panID, WORD myAddr){
    assign_addr();//分配寄存器地址,定义命令枚举表
    Com_init();//串口波特率,帧结构设定
    SET_VREG_ACTIVE();//CC2420 电源管理器
    打开
    Halwait(1000);//等待 1s 钟
    SET_RESET_ACTIVE();//软件复位
    Halwait(1);
    SET_RESET_INACTIVE();
    Halwait(5);
    SPI_SETREG();//设置 SPI 寄存器
    
```

```

RfSettings();//设置通讯协议如地址,序列号,接受状态和信息
    halRfWaitForCrystalOscillator();//等待晶振稳定
    DISABLE_GLOBAL_INT();//将短地址和 PAN_ID 写入 CC2420 的 RAM 中去
    FASTSPI_WRITE_RAM_LE (&myAddr,
    CC2420RAM_SHORTADDR, 2, n);
    FASTSPI_WRITE_RAM_LE (&panId,
    CC2420RAM_PANID, 2, n);
}
    
```

3.2 中断处理

本系统的中断主要是串口中断和 SPI 接口的 RF

芯片 FIFOP 发生的外部中断,对于主机无线模块中的中断系统,当系统复位的时候,首先主机无线模块应该响应的是串口中断,根据串口接受的数据来判断上存数据还是下载数据以及如何配置 RF 无线芯片的工作模式。在 AVR 中没有专门的中断优先级控制器来区分中断的优先级,所以本系统的具体做法是:在初始化部分置位全局中断 I,打开串口中断,关闭其他所有中断。当 AVR 单片机相应串口中断的时候,硬件会禁止全局中断 I,从而禁止系统响应其他中断,当从中断服务程序中退出时,硬件重新使能全局中断 I。当无线模块处于配置模式或者发送/接受状态的时候,关闭串口中断,打开 FIFOP 中断。这样,一方面可以保证数据的正确传输;另一方面,可以使主机系统中的无线模块串口的优先级最大,实现实时控制和监测。对于从机无线模块中的中断系统,当复位后首先打开的是 FIFOP 中断,关闭串口中断,保证 FIFOP 中断的优先级高于串口中断,这样就可以保证主机对下位机的监控。

3.3 RF 无线芯片工作模式配置

CC2420 的 RF 状态控制如 6 图所示。

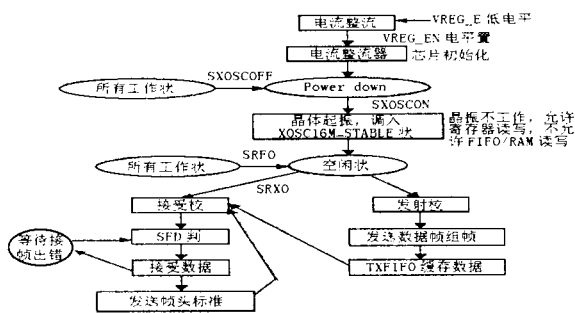


图 6 RF 状态控制图

CC2420 可以通过 4 线 SPI 总线 (SI、SO、SCLK、CSn) 设置芯片的工作模式,并实现读/写缓存数据、读/写状态寄存器等。通过控制 FIFO 和 FIFOP 管脚接口的状态可设置发射/接收缓存器。注意:在 SPI 总线接口上进行的地址和数据传输大多是 MSB 优先的。CC2420 片内有 33 个 16 比特状态设置寄存器,在每个寄存器的读/写周期中,SI 总线上共有 24 比特数据,分别为:1 比特 RAM/寄存器选择位 (0: 寄存器, 1: RAM), 1 比特读/写控制位 (0: 写, 1: 读), 6 比特地址选择位、16 比特数据位。在数据传输过程中 CSn 必须始终保持低电平。

3.4 接受模式

接受模式下的部分源程序如下:

```
void RfReceive (void) { //配置
CC2420 的接受模式
rfSettings.receiveOn = TRUE;
FASTSPI_STROBE(CC2420_SRXON); //通
```

过 SPI 写 CC2420 寄存器状态

```
FASTSPI_STROBE(CC2420_SFLUSHRX);
ENABLE_FIFOP_INT(); //使能 FIFOP 中断,关闭串口中断
DISABLE_MSCOM_INT();}
FIFOP_Receive();{ //CC2420 中断服务子程序
WORD frameControlField; INT8 length; BYTE
pFooter[2];
if((FIFOP_IS_1)) && (! FIFO_IS_1))
{ //当 FIFO 溢出的时候
FASTSPI_STROBE(CC2420_SFLUSHRX) //
清空寄存器
Return; }
FASTSPI_READ_FIFO_BYTE(length);
//写入有效数据的长度
if (length < BASIC_RF_ACK_PACKET_SIZE)
{ //如果数据包太短
FASTSPI_READ_FIFO_GARBAGE(length);
//将此次数据丢失
} else { //写入寄存器有效数据长度
```

```
rfSettings.pRxInfo->length=length-BASIC
_RF_PACKET_OVERHEAD_SIZE;
//判断是否是有效数据包
if((length==BASIC_RF_ACK_PACKET
SIZE)&&(frameControlField==BASIC_RF_ACK
FCF)&&(rfSettings.pRxInfo->seqNumber==
rfSettings.txSeqNumber)){
```

```
FASTSPI_READ_FIFO_NO_WAIT((BYTE*)
pFooter, 2); //读页脚,CRC 检查
```

```
} else { //接受剩下的数据包,
读源地址
```

```
FASTSPI_READ_FIFO_NO_WAIT((BYTE*)
&rfSettings.pRxInfo->srcAddr, 2);
```

```
FASTSPI_READ_FIFO_NO_WAIT(rfSettings.
pRxInfo->pPayload, rfSettings.pRxInfo->
length); //读数据包的有效长度。
```

```
FASTSPI_READ_FIFO_NO_WAIT((BYTE*)
pFooter, 2); //读页脚得到 RSSI 的值
```

```
RfSettings.pRxInfo->rsi=pFooter[0];
ENABLE_GLOBAL_INT();} //开启全
局中断
```

3.5 发送模式

发送模式下的源程序如下:

```

BOOL RFSendPacket(RF_TX_IBOOLINFO *
PRTI) {
    WORD frameControlField; UINT8 packetLength;
    BOOL success; BYTE spiStatusByte;
    while (FIFOP_IS_1 || SFD_IS_1) //当发送/接受处于 IDLE 状态时
        FASTSPI_STROBE(CC2420_SFLUSHTX);
    //刷新 TX FIFO 寄存器
    if (! rfSettings.receiveOn)
        FASTSPI_STROBE(CC2420_SRXON);
    do { //等待 RSSI 有效
        FASTSPI_UPD_STATUS(spiStatusByte);
    } while (! (spiStatusByte & BM(CC2420_RSSI_VALID)))
    do { //DANG RSSI 值有效的时候,开始发送数据
        FASTSPI_STROBE(CC2420_STXONCCA);
        FASTSPI_UPD_STATUS(spiStatusByte);
        HalWait(100);
    } while (! (spiStatusByte & BM(cc2420_TX_active)))
    //将数据包写入 TX_FIFO 寄存器
    packetLength = pRTI->length + BASIC_RF_PACKET_OVERHEAD_SIZE;
    FASTSPI_WRITE_FIFO((BYTE *) &packetLength, 1); //数据包长度
    FASTSPI_WRITE_FIFO((BYTE *) &frameControlField, 2); //数据帧范围
    FASTSPI_WRITE_FIFO((BYTE *) &rfSettings.txSeqNumber, 1); //序列号
    FASTSPI_WRITE_FIFO((BYTE *) &rfSettings.panId, 2); //目的地 PAN ID
    FASTSPI_WRITE_FIFO((BYTE *) &pRTI->destAddr, 2); //目的地地址
    FASTSPI_WRITE_FIFO((BYTE *) &rfSettings.myAddr, 2); //源数据地址
    FASTSPI_WRITE_FIFO((BYTE *) pRTI->pPayload, pRTI->length); //有效长度
    } while (! SFD_IS_1); //判断是否传送 OK
    success = TRUE;
    ENABLE_GLOBAL_INT(); //开启全局中断
}

```

另外,通过 CCA 管脚状态的设置可以控制清除通道估计,通过 SFD 管脚状态的设置可以控制时钟/

定时信息的输入。这些接口必须与微处理器的相应管脚相连来实现系统射频功能的控制与管理。

3.4 SPI 口的配置和读写操作

本系统中把单片机作为主机,CC2420 作为从机。通过将需要的从机的 SS 引脚拉低,主机启动一次通讯过程。主机和从机将需要发送的数据放入相应的移位寄存器。配置主机的时候,对 SPI 数据寄存器写入数据,即启动 SPI 时钟。将 8 比特的数据移入从机,传输结束后 SPI 时钟停止。传输结束状态标志 SPIF 置位。如果此时 SPCR 寄存器的 SPI 中断使能位 SPIE 置位,中断就会发生。主机可以继续往 SPDR 写入数据以移位到从机中去,或者是将从机的 SS 拉高以说明数据包发送完成。最后进来的数据将一直保存于缓冲寄存器里。配置从机的时候,只要 SS 为高, SPI 接口将一直保持睡眠状态,并保持 MISO 为三态。可以更新 SPI 数据寄存器 SPDR 的内容。即使此时 SCK 引脚有输入时钟, SPDR 的数据也不会移出,直至 SS 被拉低。一个字节完全移出之后,传输结束标志, SPIF 置位。如果此时 SPCR 寄存器的 SPI 中断使能位 SPIE 置位,就会产生中断请求。在读取移入的数据之前,从机可以继续往 SPDR 写入数据。最后进来的数据将一直保存于缓冲寄存器里。

4 结论

CC2420 是首款符合 ZigBee 技术的高集成度工业用射频收发器件,其 MAC 层和 PHY 层协议符合 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,因此本系统中采用的应用方案特别适合农业信息传递所遇到地形复杂,降低成本和数据传输速率不高的要求。并且,此无线传输模块也可以为家庭自动控制、工业监控、传感网络、消费电子、智能玩具等提供了理想的解决方案。

参考文献

- 1 CC2420 DATASHEET. PDF CHIPCON 公司
- 2 沈文, Eagle lee 等. AVR 单片机 C 语言开发入门[M]. 北京:清华大学出版社
- 3 马潮,詹卫前,耿德根. Atmega8 原理及应用手册[M]. 北京:清华大学出版社
- 4 周怡颖,凌志浩,吴勤勤. ZIGBEE 无线通讯技术及其应用探讨[J]. 自动化仪表, 2005 年 06 期
- 5 王权平,王莉. ZIGBEE 技术及其应用[J]. 现代电信科技, 2004 年 01 期
- 6 李群芳,黄建. 单片微型计算机与接口技术[M]. 北京:电子工业出版社