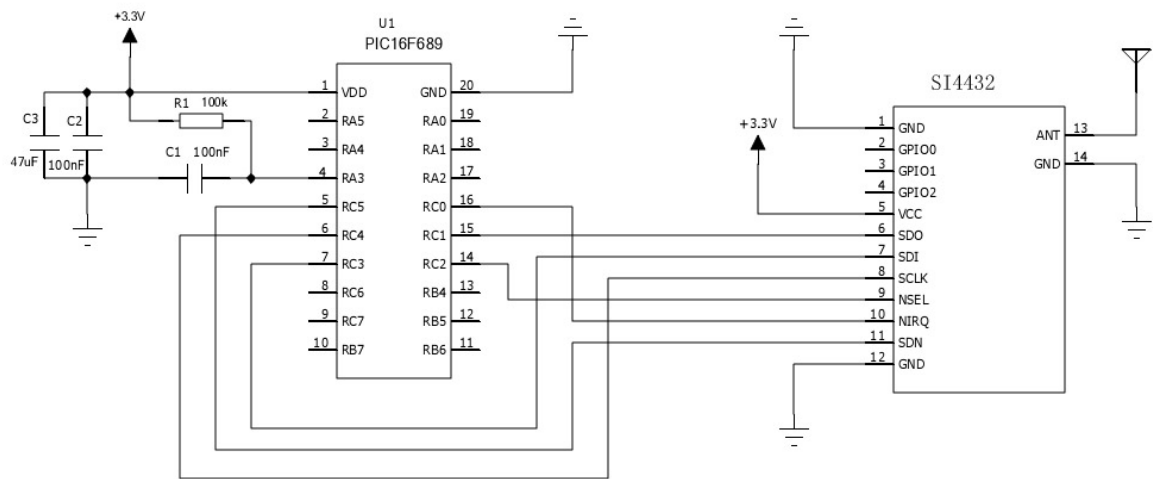


XL4432-D01 模块设计指南



1 SI4432 概述:

Si4432 主要有关闭模式，挂机模式，发射模式和接收模式组成。关闭状态下可以降低功耗，各模式切换必须先进入挂起状态再切换。其中的挂机模式，给 SPI 寄存器地址 07h 赋予不同的值，又分为五种不同的子模式。待机模式，睡眠模式，传感器模式，预备模式，调谐模式。上电复位后，或者芯片由掉电状态退出后将默认进入预备模式。

Si4432 数据传输方式主要有三种，FIFO 模式，直接模式，和 PN9 模式。

在 FIFO Mode 下,使用片内的先入先出堆栈区来发送和接收数据。对 FIFO 的操作是通过 SPI 对 07H 寄存器的连续读或者写进行的。在 FIFO Mode 下,Si4432 自动退出发送或者接收状态,当相关的中断信号产生,并且自动处理字头和 CRC 校验码。在接收数据时，自动把字头和 CRC 校验码移去。在发送数据时,自动加上字头和 CRC 校验码。

在直接收发模式下,Si4432 如传统的射频收发器一样工作

PN9 模式，在这种模式下的 Tx 数据是内部产生使用伪随机(PN9 序列)位发生器。这种模式的目的是用作测试模式不断观察调制频谱,而不必负载/提供数据。

配置 Si4432 模块是通过 SPI 方式进行的,可配置为 FIFO 方式和直接方式，推荐 Si4432 工作于 FIFO 收发模式,这种工作模式下,系统的程序编制会更加简单,并且稳定性也会更高。其中配置 Si4432 主要包括载波频率、调制方式、数据发送速率、CRC 校验、前导码、同步字、数据头、地址等,具体可参考寄存器配置工具进行。

Si4432 的寄存器操作:

Si4432 共有 128 个寄存器(O — 127), 它们控制芯片的工作和记录芯片的状态。可通过 SPI 对它们进行访问。SPI 的 it 顺序是可配置的, 其缺省配置(MSB 在前)与 MCU 的顺序相同。命令格式为 2 字节结构: 读 / 写标志(1 bit, 0—读, 1—写), 寄存器地址(7 bit)+待写数据(对于读操作, 该值也必须有, 只是可为任意值)。每次可以读写 1 / 多(burst)个字节, 它们是由时钟信号决定的, 在读写一个字节后, 如果时钟继续有效, 那么, 地址将会自动加 1, 接下来的操作将是对下一个寄存器的读写。只能在空闲状态下对寄存器进行初始化, 否则, 可能会出现意外的结果。为了提高传输信号的质量, 增大发射距离, 保证数据的传输, 建议用曼彻斯特 Manchester 编码、CRC 校验和采用 GFSK 调制。

状态机:

在完成不同的功能时, 芯片所处的状态是不同的。这些状态在满足一定的条件时可实现相互转移。

关闭(shutdown)和空闲(idle)状态称为低功耗状态, 而 idle 又可细分为 5 个不同的子状态, 它们在低功耗下完成各种与无线数据收发无关的操作。发送(Tx)和接收(RX)状态称为激活状态, 它们完成无线数据的收发。除了关闭状态外(只能通过 MCU 的 I / O 脚来设置), 其余状态都可以通过 SPI 接 EI 进行设置和读取。可通过寄存器 07h 实现状态的切换, 这种切换表现在两个方面: 1)当设置其中的某一位时, 状态立即发生切换; 2)在完成收发任务后, 决定返回到 idle 状态的哪一个子状态(在本系统中为休眠状态, 即设置 enwt=1)。可通过 02h 寄存器获取当前的状态。芯片的常态为 idle, 为了保证不漏收数据, 可利用唤醒定时器来定时唤醒芯片进入融状态(此时要设置 08h 寄存器中的 enldm 位为 1, 并设置定时常数寄存器 14h 和 19h 中的值), 在确认没有数据 / 收完数据后再返回到原来的 idle 子状态。

用普通 51 单片机 IO 口模拟 SPI:

SPI 读操作函数:

```
unsigned char SPI_Read(void)
```

```
{
```

```
    unsigned char i,rxdata;
```

```
    rxdata = 0x00;
```

```
    for (i = 0;i < 8;i++)
```

```
    {
```

```

        rxdata = rxdata<<1;

        SI4432_SCLK=0;

        if (SI4432_SDO==1)    //读取最高位，保存至最末尾，通过左移位完成整个
字节
        {

            rxdata |= 0x01;

        }

        delay_10us(2);

        SI4432_SCLK=1;

        delay_10us(2);

    }

    return rxdata;

}

```

SPI 写操作函数:

```

void SPI_Write(unsigned char txdata)

{

    unsigned char i;

    for (i = 0;i < 8;i++)

    {

        SI4432_SCLK=0;

        if ((txdata&0x80)==0x80)    //总是发送最高位

        {

```

```

        SI4432_SDI=1;

    }

    else

    {

        SI4432_SDI=0;

    }

    txdata = txdata<<1;

    delay_10us(2);

    SI4432_SCLK=1;

    delay_10us(2);

}

}

```

SI4432 寄存器读操作函数:

```

unsigned char  SI4432_ReadReg(unsigned char  addr)

{

    unsigned char value;

    SI4432_SEL=0;

    SPI_Write(addr|RR);

    value = SPI_Read();

    SI4432_SEL=1;

    return value;

}

```

SI4432 寄存器写操作函数:

```
void SI4432_WriteReg(unsigned char addr, unsigned char value)

{

    SI4432_SEL=0;

    SPI_Write(addr|WR);

    SPI_Write(value);

    SI4432_SEL=1;

}
```

程序设计思路

SI4432 软件编程采用模块化设计思想，系统中各主要功能模块均编成独立的函数由主程序调用。功能模块包括：初始化程序(包括初始化单片机、SPI、Si4432)，无线发送程序，无线接收程序等。无线发送程序负责写入数据，并根据通信协议为数据加上前导码、同步字、数据长度及 CRC 校验字节，形成数据包将其发送出去；无线接收程序负责接收并检验数据包中的 CRC 字节，以确保接收到的数据的正确性。

无线收发模块之间的通信是以数据包的形式发送的，本系统定义的数据包格式如下：

Preamble	Sync Word	Packet Length	PAYLOAD	CRC
8n 位	2 字节	0 或 1 字节	用户配置	2 字节

其中，Preamble(前导码)是一连串的 10101010，其数量为 8n 位，n 的大小由用户编程决定。数据包在传输过程中会在每个包的前面加上可设置长度的前导码；接收端为了识别帧的到来，需要前导码进行帧同步，从而确定收发系统之间何时发送和接收数据。SyncWord(同步字)在前导码之后，要用设定好的同步字来作为同步模式的标志码。本系统设定的同步字为 2 个字节，同步字内容为 0x2DD4，接收端在检测到同步字后才开始接收数据。Packet Length 是数据载荷长度。

PAYLOAD(有效数据)是用户所发送的数据。CRC(CRC 校验和)由内置 CRC 校检。Si4432 内部集成有调制 / 解调、编码 / 解码等功能，从而 Preamble、Syncword、Packet Length 和 CRC 都是硬件自动加上去的，用户只需设定数据包的组成结构和部分结构的具体内容(如前导码和同步字)。

(1)初始化程序

初始化程序包括单片机的初始化，SPI 的初始化，以及 Si4432 的关于无线收发频率、工作模式、发射速率等内部寄存器的初始化配置。

系统上电后，单片机处于默认状态，根据系统功能需求重新进行初始化配置。单片机的数字交叉开关允许将内部数字系统资源映射到端口 I / O 引脚，可通过设置交叉开关控制寄存器，将片内资源配置到具体的端口 I / O 引脚上。这一特性允许用户根据自己的特定应用选择通用端口 I / O 和所需数字资源的组合，提高了应用的灵活性。本系统中，主要配置了 SPI 通信的 4 线，液晶 LCD 的数据线接口、控制线接口和 RS232 串口数据输入 / 输出等。

初始化 SPI 时，可以通过对 SPI1CFG 寄存器和 SPI1CN 寄存器的配置来选择具体使用规则。这里，选择主 SPI，4 线模式，时钟极性为低电平，在时钟上升沿时对数据采样；通过配置 SPI1CKR 寄存器，可将同步时钟频率设为晶振频率的 1 / 4。

上电之初，Si4432 也处于默认状态，需要进行配置才能工作。Si4432 有 70 多个寄存器需要配置，它们决定了 Si4432 的工作模式，具体配置可以参考 Si4432 的数据手册。Si4432 的初始化是一个重要的部分，配置的恰当与否对系统最终的通信效果有很大的影响。主控制器单片机通过 SPI 配置 Si4432 的 1ch、1dh 等寄存器，写入相应的初始化 RF 控制字(主要是频率、传输速度、传输方式等)；通过配置 33h、34h 等寄存器来设置包的结构、前导码长度、同步字内容等。本系统采用同步传输模式，以 0x2DD4 作为同步模式的标志码，传输完同步字后才开始传输数据载荷。每次发送数据必须以同步字 0x2DD4 作为发送数据的同步标志，接收端在检测到同步字后才开始接收数据。

(2)无线发送程序

无线发送程序流程如图 3 所示。完成单片机、SPI 和 Si4432 的初始化后，配置寄存器写入相应的初始化 RF 控制字。接下来，通过配置 Si4432 的寄存器 3eh 来设置包的长度，通过 SPI 连续写寄存器 7fh，往 TX FIFO 里写入需要发送的数据。然后打开“发送完中断允许”标志，将其他中断都禁止。当有数据包发送完时，引脚 nIRQ 会被拉低以产生一个低电平从而通知单片机数据包已发送完毕。完成中断使能后，使能发送功能，数据开始发送。等待 nIRQ 引脚因中断产生而使电平拉低，当 nIRQ 引脚变为低时读取中断状态并拉高 nIRQ，否则继续等待。如果数据发送成功，指示灯会变亮。一次数据发送成功后，进入下一次数据循环发送状态。

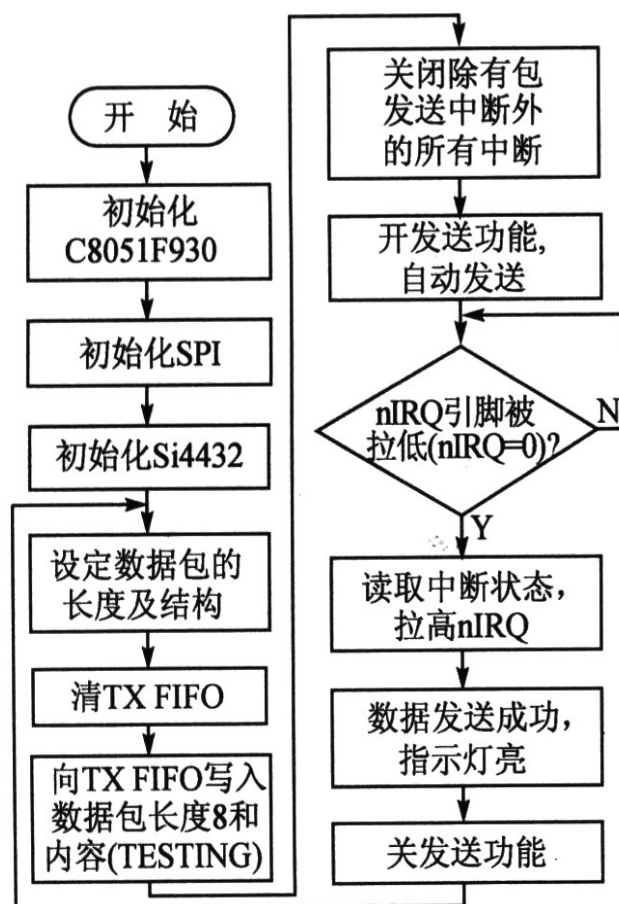


图 3 无线发送程序流程

(3)无线接收程序

无线接收程序流程如图 4 所示。

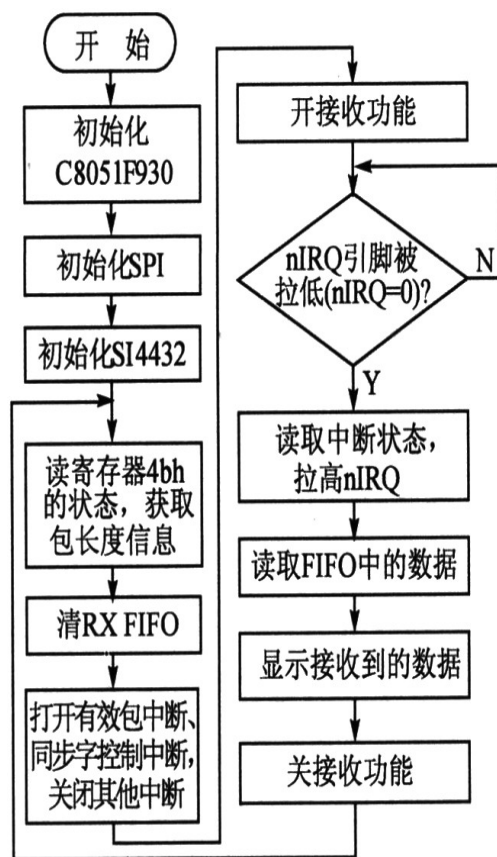


图 4 无线接收程序流程

程序完成单片机、SPI 接口和 Si4432 的初始化后，配置寄存器写入相应的初始化 RF 控制字。通过访问寄存器 7fh 从 RX FIFO 中读取接收到的数据。相应的控制字设置好之后，若引脚 nIRQ 变成低电平，则表示 Si4432 准备好接收数据。完成这些初始化配置后，通过寄存器 4bh 读取包长度信息。

然后，打开“有效包中断”和“同步字检测中断”，将其他中断都禁止。引脚 nIRQ 用来检测是否有有效包被检测到，若引脚 nIRQ 变为低电平，则表示有有效的数据包被检测到。本系统用 0x2DD4 作为同步模式的标志码，接收模块通过检测这个同步字来同步接收数据。最后，使能接收功能，数据开始接收。等待 nIRQ 引脚因中断产生而使电平拉低，读取中断标志位复位 nIRQ 引脚，使 nIRQ 恢复至初始的高电平状态以准备下一次中断触发的检测。通过 SPI 读取 RX FIFO 中的数据，将数据送至液晶显示，之后进入下一次数据接收状态。