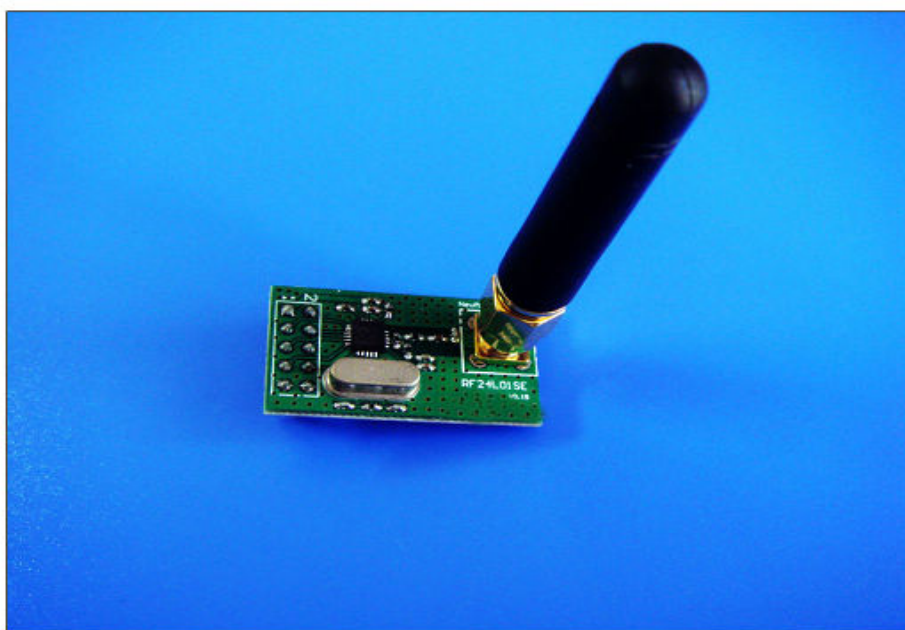
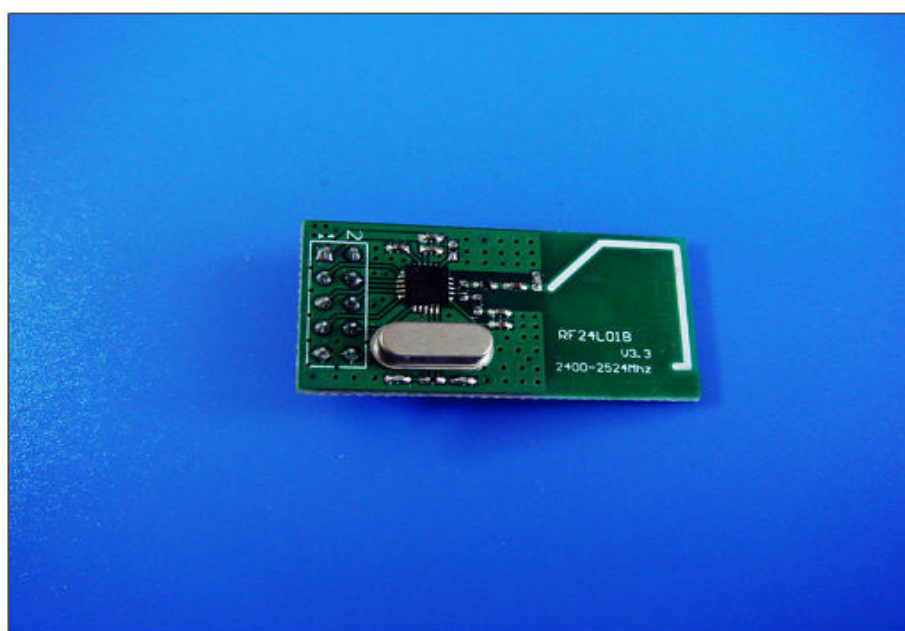


nRF24L01 无线模块 用户手册



目录

产品概述	3
基本特性	3
引脚接口说明	4
模块尺寸	6
nRF2401 工作模式	7
Enhanced ShockBurst™收发模式	7
Enhanced ShockBurst™数据发送流程	8
空闲模式	9
关机模式	9
nRF24L01 模块参数设置	9
主要参数设置	10
程序设计分析	10
nRF24L01 初始化	10
nRF24L01 SPI写操作	11
nRF24L01 SPI读操作	11
nRF24L01 写寄存器函数	12
nRF24L01 连续读多个寄存器函数	12
nRF24L01 连续写多个寄存器函数	12
nRF24L01 接收模式设置	13
nRF24L01 接收数据流程	13
nRF24L01 发送数据流程	13
无线应用注意事项	14
我们的承诺	15

产品概述

nRF24L01 是挪威 NordicVLSI 公司出品的一款新型射频收发器件, 采用 4 mm×4 mm QFN20 封装;nRF24L01 工作在 ISM 频段: 2.4~2.524 GHz。且内置频率合成器、功率放大器、晶体振荡器、调制器等功能, 并融合增强型 ShockBurst 技术, 其中地址、输出功率和通信频道可通过程序进行配置, 适合用于多机通信。nRF24L01 功耗很低, 在以-6 dBm 的功率发射时, 工作电流也只有 9 mA;而对应接收机的工作电流只有 12.3 mA, 多种低功率工作模式(掉电模式和空闲模式)使节能设计更方便。nRF24L01 在业界领先的低功耗特点使其特别适合采用钮扣电池供电的 2.4G 应用, 整个解决方案包括链路层和 MultiCeiver 功能提供了比现有的 nRF24XX 更多的功能和更低的电源消耗, 与目前的蓝牙技术相比在提供更高速率的同时, 而只需花更小的功耗

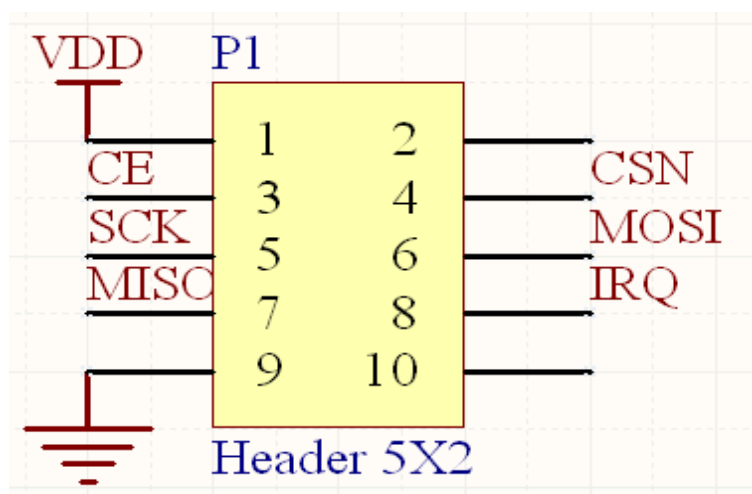
基本特性

- (1) 2.4Ghz 全球开放 ISM 频段免许可证使用
- (2) 最高工作速率 2Mbps, 高效 GFSK 调制, 抗干扰能力强
- (3) 125 频道, 满足多点通信和跳频通信需要
- (4) 内置硬件 CRC 检错和点对多点通信地址控制
- (5) 低功耗 1.9 - 3.6V 工作, 适合电池供电应用
- (6) 待机模式下状态为 22uA; 掉电模式下为 900nA

- (7) 模块可软件设地址，只有收到本机地址时才会输出数据（提供中断指示），可直接接各种单片机使用，软件编程非常方便
- (8) 内置专门稳压电路，即使开关电源也有很好的通信效果
- (9) 标准 DIP 间距接口，便于嵌入式应用
- (10) 具有自动应答机制，和 CRC 校验，数据通讯稳定可靠。

同时，为便于用户开发，我们提供配套评估套件，为产品开发保驾护航，使无线应用开发大大加速，并避免不必要的误区。

引脚接口说明



引脚功能介绍

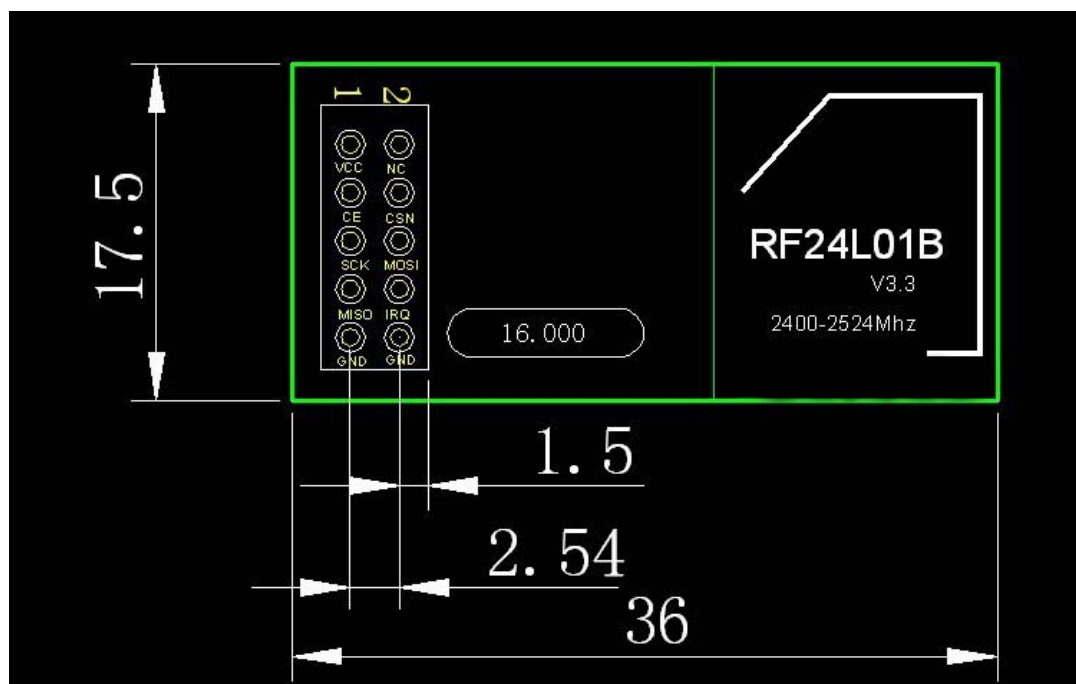
管脚	名称	管脚功能	备注说明
1	VCC	电源	1.9-3.6 V
2	NC	无	悬空
3	CE	数字输入	收发模式切换控制端口

4	CSN	数字输入	SPI片选使能端，低电平有效
5	SCK	数字输入	SPI时钟
6	MOSI	数字输入	SPI从设备数据输入
7	MISO	数字输出	SPI从设备数据输出
8	IRQ	数字输出	工作状态指示
9	GND	电源地	和系统共地
10	NC	无	悬空

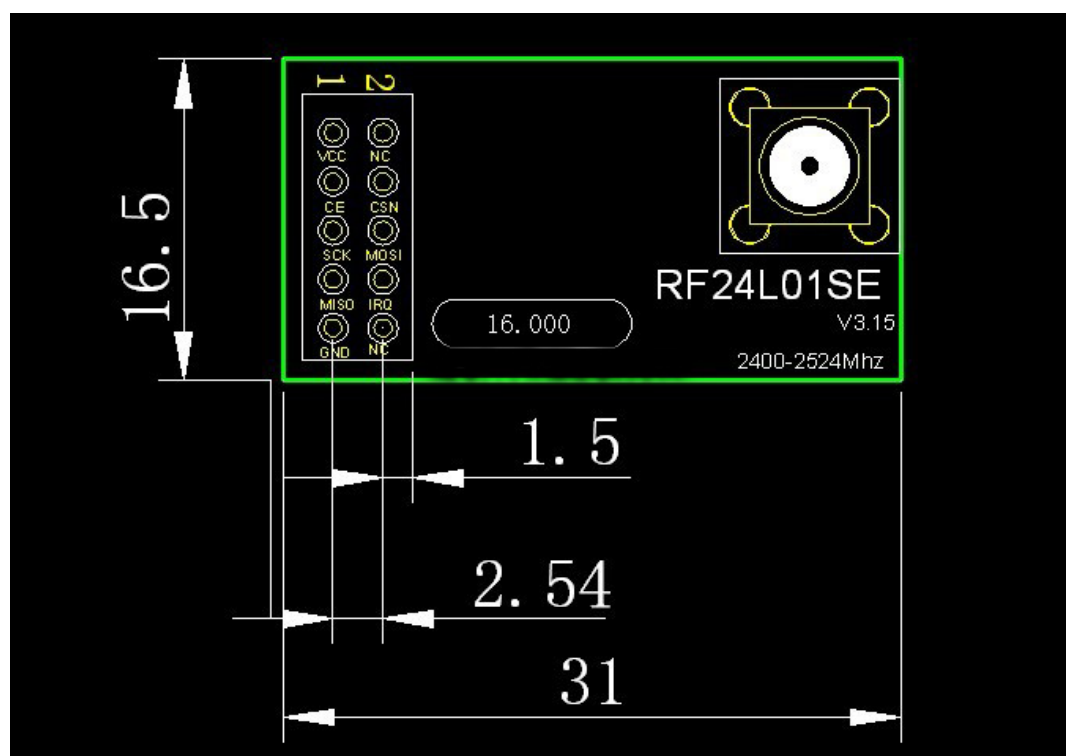
备注

1. VCC 引脚的电压范围为2.3-3.6V 之间，不能在这个区间之外，如超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右；
2. 硬件没有集成SPI功能的单片机也可以控制本模块，用普通单片 I/O口模拟 SPI 时序进行读写操作即可；
3. 模块接口采用标准2.54mmDIP插针，13 脚、14 脚为接地脚，需要和系统电路的逻辑地连接起来；
4. 与 51 系列单片机 P0 口连接时候，需要加 10K 的上拉电阻，与其余口连接不需要。其他系列的5V单片机，如AVR、PIC，请参考该系列单片机 I/O 口输出电流大小，如果超过 10mA，需要串联 2-5K电阻分压，否则容易烧毁模块！如果是 3.3V 的MCU，可以直接和RF24L01模块的I/O口线连接。

模块尺寸（单位：mm）



RF24L01B 模块尺寸示意图（PCB 板载天线）



RF24L01SE 模块尺寸示意图（SMA 天线接口）

nRF2401 工作模式

发送模式

接收模式

待机模式

关机模式

这四种工作模式由 PWR_UP register 、 PRIM_RX register 和 CE 三者组合状态决定，详见下表。

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO
TX mode	1	0	1→0	Stays in TX mode until packet transmission is finished
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

Enhanced ShockBurstTM 收发模式

Enhanced ShockBurstTM 收发模式下，使用片内的先入先出堆栈区，数据低速从微控制器送入，但高速 (2Mbps) 发射，这样可以尽量节能，因此，使用低速的微控制器也能得到很高的射频数据发射速率。与射频协议相关的所有高速信号处理都在片内进行，这种做法有三大好处：尽量节能；低的系统费用 (低速微处理器也能进行高速射频发射)；数据在空中停留时间短，抗干扰性高。Enhanced ShockBurstTM 技术同时也减小了整个系统的平均工作电流。

在 Enhanced ShockBurstTM 收发模式下， nRF24L01 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，在发送模式下，置 CE 为高，至少 10 μ s，将发送过程完成后。

Enhanced ShockBurstTM 数据发送流程

- A. 把接收机的地址和要发送的数据按时序送入 nRF24L01;
- B. 配置CONFIG寄存器，使之进入发送模式。
- C. 微控制器把CE置高(至少 10 μ s)，激发nRF24L01 进行Enhanced ShockBurstTM发射;
- D. nRF24L01 的Enhanced ShockBurstTM发射
 - (1) 给射频前端供电;
 - (2) 射频数据打包(加字头、CRC校验码);
 - (3) 高速发射数据包;
 - (4) 发射完成，nRF24L01 进入空闲状态。

Enhanced ShockBurstTM数据接收流程

- A. 配置本机地址和要接收的数据包大小;
- B. 配置CONFIG寄存器，使之进入接收模式，把CE置高。
- C. 130 μ s 后，nRF24L01 进入监视状态，等待数据包的到来;
- D. 当接收到正确的数据包(正确的地址和 CRC 校验码)，nRF2401 自动把字头、地址和 CRC 校验位移去;
- E. nRF24L01 通过把 STATUS 寄存器的 RX_DR 置位(STATUS 一般引起微控制器中断)通知微控制器;

F. 微控制器把数据从 nRF2401 读出;

G. 所有数据读取完毕后, 可以清除 STATUS 寄存器。nRF2401 可以进入四种主要的模式之一。

空闲模式

nRF24L01 的空闲模式是为了减小平均工作电流而设计, 其最大的优点是, 实现节能的同时, 缩短芯片的起动时间。在空闲模式下, 部分片内晶振仍在工作, 此时的工作电流跟外部晶振的频率有关。

关机模式

在关机模式下, 为了得到最小的工作电流, 一般此时的工作电流为 900nA 左右。关机模式下, 配置字的内容也会被保持在 nRF2401 片内, 这是该模式与断电状态最大的区别。

nRF24L01 模块参数设置

nRF2401 的所有配置工作都是 MCU 通过 SPI 接口读写操作完成, 共有 30 字节的配置字。工作于 Enhanced ShockBurst™ 收发模式, 这种工作模式下, 系统的程序编制会简单, 且稳定性也会更高, 因此, 下文着重介绍如何把 nRF24L01 配置为 Enhanced ShockBurst™ 收发模式, 从而掌握无线数据控制操作方法。

ShockBurst™ 的配置字使 nRF24L01 能够自动处理射频协议, 在

配置完成后，在 NewMsg_RF24L01 工作的过程中，只需改变其最低一个字节中的内容，以实现接收模式和发送模式之间切换。

主要参数设置

1. 数据宽度：声明射频数据包中数据占用的位数。这使 nRF24L01 能够区分接收数据包中的数据和 CRC 校验码；
2. 地址宽度：声明射频数据包中地址占用的位数。这使 nRF24L01 能够区分地址和数据；
3. 接收通道：接收数据的地址，有通道 0 到通道 5 的地址；
4. CRC 校验：使 nRF24L01 能够生成 CRC 校验码和解码。当使用 nRF24L01 片内的 CRC 技术时，要确保在配置字 (CONFIG 的 EN_CRC) 中 CRC 校验被使能，并且发送和接收使用相同的协议。
5. 工作频率：可以设置在 2.4-2.524GHZ 之间，间隔 1MHZ
6. 发射功率：发射功率的大小决定了通信距离和整体功耗
7. 通信速率：1Mbps、2Mbps 通信速率可选择设置

程序设计分析

nRF24L01 初始化

```
void init_NRF24L01(void)
```

联系电话：13704018223 陈工
在线咨询：QQ:35625400 474882985

E-mail: chj_006@sina.com
MSN:1188mm88@hotmail.com

```
{
  inerDelay_us(100);
  CE=0;    //
  CSN=1;   // SPI 禁止
  SCK=0;   //
  SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS, TX_ADR_WIDTH);
    // 写本地地址
  SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, RX_ADDRESS, RX_ADR_WIDTH);
    // 写接收端地址
  SPI_RW_Reg(WRITE_REG + EN_AA, 0x01);
    // 频道 0 自动 ACK 应答允许
  SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01);
    // 允许接收地址只有频道 0, 如果需要多频道可以参考 Page21
  SPI_RW_Reg(WRITE_REG + RF_CH, 0);
    // 设置信道工作为 2.4GHZ, 收发必须一致
  SPI_RW_Reg(WRITE_REG + RX_PW_P0, RX_PLOAD_WIDTH);
    // 设置接收数据长度, 本次设置为 32 字节
  SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x07);
    // 设置发射速率为 1MHZ, 发射功率为最大值 0dB
}
```

nRF24L01 SPI 写操作

```
uint SPI_RW(uint uchar)
{
  uint bit_ctr;
  for(bit_ctr=0;bit_ctr<8;bit_ctr++) // output 8-bit
  {
    MOSI = (uchar & 0x80);           // output 'uchar', MSB to MOSI
    uchar = (uchar << 1);             // shift next bit into MSB..
    SCK = 1;                          // Set SCK high..
    uchar |= MISO;                    // capture current MISO bit
    SCK = 0;                          // ..then set SCK low again
  }
  return(uchar);                     // return read uchar
}
```

nRF24L01 SPI 读操作

```
uchar SPI_Read(uchar reg)
{
  uchar reg_val;
```

```
CSN = 0;                // CSN low, initialize SPI communication...
SPI_RW(reg);            // Select register to read from..
reg_val = SPI_RW(0);    // ..then read register value
CSN = 1;                // CSN high, terminate SPI communication
return(reg_val);        // return register value
}
```

nRF24L01 写寄存器函数

```
uint SPI_RW_Reg(uchar reg, uchar value)
{
    uint status;
    CSN = 0;                // CSN low, init SPI transaction
    status = SPI_RW(reg);   // select register
    SPI_RW(value);          // ..and write value to it..
    CSN = 1;                // CSN high again
    return(status);         // return nRF24L01 status uchar
}
```

nRF24L01 连续读多个寄存器函数

```
uint SPI_Read_Buf(uchar reg, uchar *pBuf, uchar uchars)
{
    uint status, uchar_ctr;
    CSN = 0;                // Set CSN low, init SPI
transaction
    status = SPI_RW(reg);   // Select register to write to
and read status uchar
    for(uchar_ctr=0;uchar_ctr<uchars;uchar_ctr++)
        pBuf[uchar_ctr] = SPI_RW(0);    //
    CSN = 1;
    return(status);         // return nRF24L01 status uchar
}
```

nRF24L01 连续写多个寄存器函数

```
uint SPI_Write_Buf(uchar reg, uchar *pBuf, uchar uchars)
{
    uint status, uchar_ctr;
    CSN = 0;                //SPI 使能
    status = SPI_RW(reg);
    for(uchar_ctr=0; uchar_ctr<uchars; uchar_ctr++) //
```

```
SPI_RW(*pBuf++);  
CSN = 1;           //关闭 SPI  
return(status);  
}
```

nRF24L01 接收模式设置

```
void SetRX_Mode(void)  
{  
    CE=0;  
    SPI_RW_Reg(WRITE_REG + CONFIG, 0x0f);  
    // IRQ 收发完成中断响应, 16 位 CRC , 主接收  
    CE = 1;  
    inerDelay_us(130);  
}
```

nRF24L01 接收数据流程

```
unsigned char nRF24L01_RxPacket(unsigned char* rx_buf)  
{  
    unsigned char revale=0;  
    sta=SPI_Read(STATUS); // 读取状态寄存其来判断数据接收状况  
    if(RX_DR)             // 判断是否接收到数据  
    {  
        CE = 0;           、  
        SPI_Read_Buf(RD_RX_PLOAD, rx_buf, TX_PLOAD_WIDTH);  
        // read receive payload from RX_FIFO buffer  
        revale =1;        //读取数据完成标志  
    }  
    SPI_RW_Reg(WRITE_REG+STATUS, sta);  
    //接收到数据后 RX_DR, TX_DS, MAX_PT 都置高为 1, 通过写 1 来清楚中断标志  
    return revale;  
}
```

nRF24L01 发送数据流程

```
void nRF24L01_TxPacket(unsigned char * tx_buf)  
{  
    CE=0;           //StandBy I 模式  
    SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS, TX_ADR_WIDTH);  
    // 装载接收端地址  
    SPI_Write_Buf(WR_TX_PLOAD, tx_buf, TX_PLOAD_WIDTH);
```

联系电话: 13704018223 陈工

在线咨询: QQ:35625400 474882985

E-mail: chj_006@sina.com

MSN:1188mm88@hotmail.com

```
// 装载数据
SPI_RW_Reg(WRITE_REG + CONFIG, 0x0e);
// IRQ 收发完成中断响应, 16 位 CRC, 主发送
CE=1;          //置高 CE, 激发数据发送
inerDelay_us(10);
}
```

无线应用注意事项

- (1) 无线模块的 VCC 电压范围为 1.8V-3.6V 之间, 不能在这个区间之外, 超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右。
- (2) 除电源 VCC 和接地端, 其余脚都可以直接和普通的 51 单片机 I/O 口直接相连, 无需电平转换。当然对 3V 左右的单片机更加适用了。
- (3) 硬件上面没有 SPI 的单片机也可以控制本模块, 用普通单片机 I/O 口模拟 SPI 不需要单片机真正的串口介入, 只需要普通的单片机 I/O 口就可以了, 当然用串口也可以了。模块按照接口提示和母板的逻辑地连接起来
- (4) 标准 DIP 插针, 如需要其他封装接口, 或其他形式的接口, 可联系我们定做。
- (5) 任何单片机都可实现对无线模块的数据收发控制, 并可根据我们提供的程序, 然后结合自己擅长的单片机型号进行移植;
- (6) 频道的间隔的说明: 实际要想 2 个模块同时发射不相互干扰, **两者频道间隔应该至少相差 1MHZ**, 这在组网时必须注意, 否则同频比干扰。

(7) 实际用户可能会应用其他自己熟悉的单片机做为主控芯片，所以，建议大家在移植时注意以下 4 点：

- A: 确保 IO 是输入输出方式，且必须设置成数字 IO；
- B: 注意与使用的 IO 相关的寄存器设置，尤其是带外部中断、带 AD 功能的 IO，相关寄存器一定要设置好；
- C: 调试时先写配置字，然后控制数据收发
- D: 注意工作模式切换时间

我们的承诺

最后，欢迎您使用我们的产品，在应用中有技术问题请及时向我们联系，我们会予以技术知道，同时运输中出现产品问题我们会全面责任并予以更换。

愿与您一起走向成功