

# 无线传输模组

---

## 使用说明书

V1.0 – 2006.11.15

凌阳科技大学计划教育推广中心  
北京海淀上地信息产业基地中黎科技园 1 号楼 5 层

TEL: 86-10-62981668    FAX: 86-10-62962425    E-mail: [unsp@sunplus.com.cn](mailto:unsp@sunplus.com.cn)    <http://www.unsp.com.cn>

## 版权声明

凌阳科技股份有限公司保留对此文件修改之权利且不另行通知。凌阳科技股份有限公司所提供之信息相信为正确且可靠之信息，但并不保证本文件中绝无错误。请于向凌阳科技股份有限公司提出订单前，自行确定所使用之相关技术文件及规格为最新之版本。若因贵公司使用本公司之文件或产品，而涉及第三人之专利或著作权等智能财产权之应用及配合时，则应由贵公司负责取得同意及授权，本公司仅单纯贩售产品，上述关于同意及授权，非属本公司应为保证之责任。又未经凌阳科技股份有限公司之正式书面许可，本公司之所有产品不得使用于医疗器材，维持生命系统及飞航等相关设备。

# 目 录

1	系统简介 .....	1
1.1	基本特性与参数指标 .....	1
1.2	功能简介 .....	1
1.3	结构示意图 .....	1
1.4	实物图 .....	2
1.5	注意事项 .....	2
2	硬件模块说明 .....	1
2.1	电路原理图简介 .....	1
2.1.1	nRF2401A 及其外围电路.....	1
2.1.2	指示电路 .....	2
2.1.3	接口电路 .....	2
3	软件说明 .....	3
3.1	初始化程序 .....	3
3.2	数据发送程序 .....	6
3.3	数据接收程序 .....	8
3.4	其他程序 .....	9
4	编程举例 .....	11
4.1	单向数据传输 .....	11
4.1.1	要求 .....	11
4.1.2	分析 .....	11
4.2	双向数据传输 .....	14
4.2.1	要求 .....	15
4.2.2	分析 .....	15
5	常见问题解答 .....	19
6	附录 .....	20
6.1	配件清单及配件说明 .....	20
6.2	相关资料清单 .....	20
6.3	电路原理图 .....	21
6.4	公司联系方式 .....	22

## 1 系统简介

无线传输模组以 nRF2401A 芯片为核心，提供了一路的无线发射和两路的无线接收功能，该模组通过一个 10Pin 接口直接插接到 61 板上，方便使用。

### 1.1 基本特性与参数指标

- ◆ 工作电压：3.3V
- ◆ 模块尺寸：27mm\*31mm

### 1.2 功能简介

- ◆ 可以进行无线数据发送和接收
- ◆ 可以开发无线的电子产品：如无线遥控器、无线麦克、无线耳机、无线的文件传输系统等

### 1.3 结构示意图

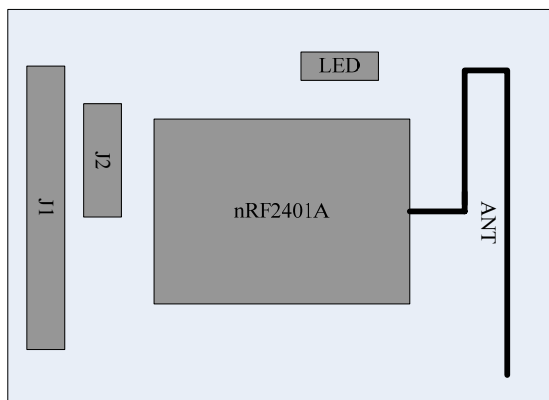


图 1.1 无线传输模组结构示意图

应用时，只需要通过 10Pin 接口 J1 直接和 61 板的 I/O 端口相连接即可。如果需要两个通道接收，可以将预留端口 J2 引出，接到单片机对应的 I/O 上即可。

## 1.4 实物图



图 1.2 无线传输模组实物图

## 1.5 注意事项

- 1、nRF2401A 为低电压低功耗器件，其工作电压不高于 3.6V，使用时请选择 61 板的 I/O 端口电平跳线为 3V（将 61 板 J5 跳线跳至 3V 端）。
- 2、芯片供电电压 VDD 和信号高电平电压应一致。

## 2 硬件模块说明

### 2.1 电路原理图简介

#### 2.1.1 nRF2401A 及其外围电路

nRF2401A 及其外围电路如图 2.1，包括 nRF2401A 芯片部分、稳压部分、晶振部分、天线部分。电压 VDD 经电容 C1、C2、C3 处理后为芯片提供工作电压；晶振部分包括 Y1、C9、C10，晶振 Y1 允许值为：4MHz、8MHz、12 MHz、16 MHz，如果需要 1Mbps 的通信速率，则必须选择 16MHz 晶振。天线部分包括电感 L1、L2，用来将 nRF2401A 芯片 ANT1、ANT2 管脚产生的 2.4G 电平信号转换为电磁波信号，或者将电磁波信号转换为电平信号输入芯片的 ANT1、ANT2 管脚。

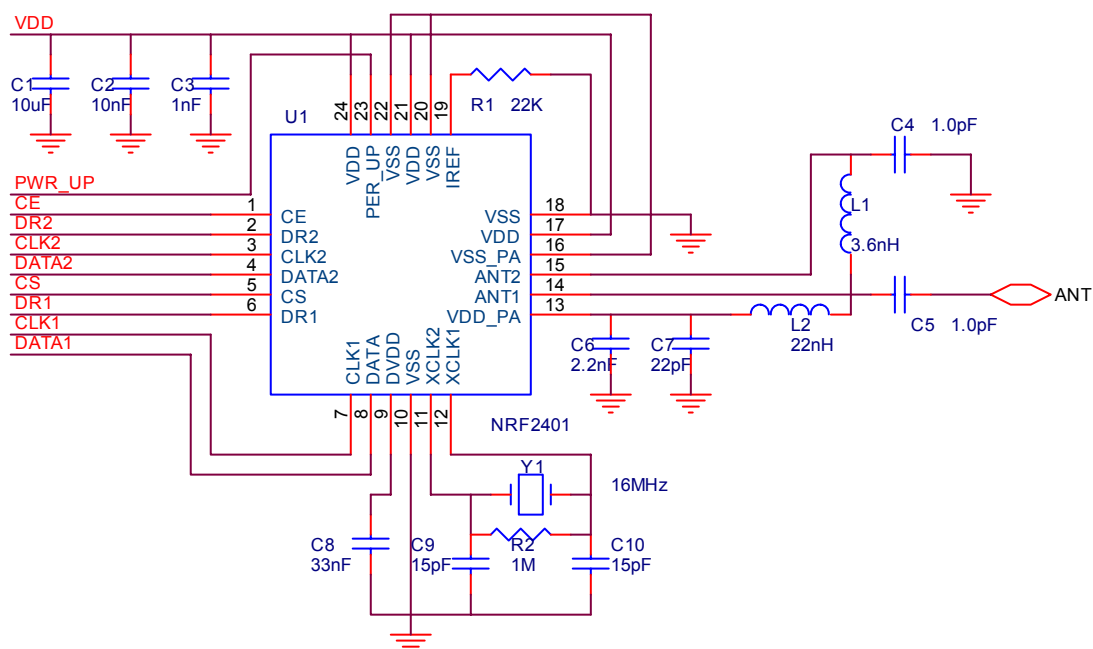


图 2.1 nRF2401A 及其外围电路

### 2.1.2 指示电路

电路板上提供了一个电源指示灯，其电路如图 2.2。

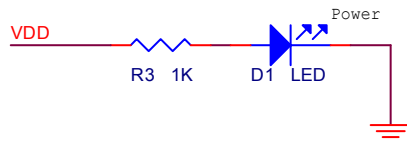


图 2.2 电源指示电路

### 2.1.3 接口电路

为方便与 61 板的连接，模组提供了两个接口 J1、J2，其中 J1 为 nRF2401A 的控制端口和通道 1 的收发通道，J2 为预留端口，是通道 2 的接收通道。J1 接口为 10Pin 的插孔，其布局 and 61 板的 I/O 端口布局一致，可以直接插接到 61 板的 I/O 端口使用。J2 预留，如果需要采用双通道接收时，可以将 J2 对应的 3 根信号线引出，接到单片机的 I/O 上即可。

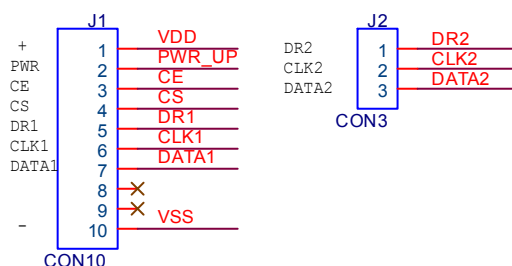


图 2.3 接口电路

对应的 I/O 连接和功能描述如表 2.1 所示。

表 2.1 芯片管脚连接及功能描述

2401A 管脚	作用描述
PWR_UP	nRF2401A 芯片上电
CE	nRF2401A 使能
CS	nRF2401A 片选
DR1	通道 1 数据请求
CLK1	通道 1 时钟
DATA1	通道 1 数据
DR2	通道 2 数据请求
CLK2	通道 2 时钟
DATA2	通道 2 数据

### 3 软件说明

本模组提供了系统的驱动程序文件 nRF2401.C，包括芯片的初始化、数据发送、数据接收。下面逐个介绍各个程序的流程和应用：

#### 3.1 初始化程序

初始化函数 void nRF2401\_Initial(void)

程序包括端口初始化和向芯片写控制字操作，初始化流程如图 3.1 所示：

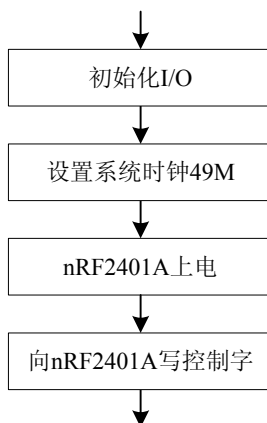


图 3.1 初始化程序流程图

##### SPCE061A 单片机的初始化：

初始化 I/O 将会把 nRF2401A 模组用到的所有接口初始化为对应的输入输出属性；设置系统时钟为 49MHz，以提高通信的速度，及延时的准确性。

##### nRF2401A 初始化：

对 nRF2401A 的初始化包括 nRF2401A 上电和向 nRF2401A 写控制字。nRF2401A 上电是将芯片的 PWR\_UP 管脚设置为高电平，上电以后才可以对 nRF2401A 进行控制和读写操作。

nRF2401A 共有 18 Byte（144bit）的命令字，下面逐位介绍其作用。

##### bit143~bit120：测试保留区

测试保留区，用户不需设置。

##### bit119~bit112：通道 2 接收数据长度（bit）

通道 2 接收数据的长度 Data2\_W，用于接收。如果通道 2 接收到一帧数据，在去掉地址和校验位后，其数据长度和 bit119 ~ bit112 值一致，则认为通道 2 接收到一帧数据，置位 DR2，否则丢弃。

##### bit111~bit104：通道 1 接收数据长度（bit）

通道 1 接收数据的长度 Data1\_W，用于接收。如果通道 1 接收到一帧数据，去掉地址和校验位后，



其数据长度和 **bit111 ~ bit104** 值一致，则认为通道 1 接收到一帧数据，置位 DR1，否则丢弃。

通道 1 和通道 2 的数据长度受 nRF2401A 帧数据总长度限制，帧数据总长度为 256 bit，每一帧的数据长度  $\text{Data\_W} \leq 256 - \text{Addr\_W} - \text{CRC}$ 。

#### **bit103~bit64: 通道 2 地址**

通道 2 的地址 **Addr\_2**，用于接收。设置本机地址，通信过程中，发射端通过配置该地址发送数据才能与之通信。

#### **bit63~bit24: 通道 1 地址**

通道 1 的地址 **Addr\_1**，用于接收。设置本机地址，通信过程中，发射端通过配置该地址发送数据才能与之通信。

#### **bit23~bit18: 通道地址有效位数**

通道 1 和通道 2 的地址有效位数，用于接收。结合 **bit63~bit24** 和 **bit103~bit64** 选定通道的接收地址。其值不超过 40。

如果 **bit63~bit24** 的值为 00 00 00 01 05，**bit23~bit18** 的值为 8，则通道 1 的地址为 05；

如果 **bit63~bit24** 的值为 00 00 00 01 05，**bit23~bit18** 的值为 32，则通道 1 的地址为 00 00 01 05

#### **bit17: CRC 校验位选择**

选择 CRC 校验位数，用于收发。如果该位为“0”，则 CRC 校验为 8 位；如果该位为“1”，则 CRC 校验为 16 位；

#### **bit16: CRC 校验使能**

使能 CRC 校验，用于收发。如果该位为“0”，则不进行 CRC 校验；如果该位为“1”，则进行 CRC 校验。

#### **bit15: 通道 2 接收使能**

通道 2 的接收允许，用于接收。如果该位为“0”，则仅有通道 1 可以接收；如果该位为“1”，则通道 1 和通道 2 均可接收。注意通道 1 通道 2 同时接收时，通道 2 的接收频率比通道 1 的收发频率高 8MHz。

#### **bit14: 收发模式选择**

该位决定 nRF2401A 的收发方式，用于收发。如果该位为“0”，nRF2401A 工作在 Direct 模式（直接收发模式）；如果该位为“1”，nRF2401A 工作在 Shock Burst 模式。Direct 模式在此不作详细介绍，用户如果使用，可以参考芯片的数据手册；Shock Burst 收发操作会在后面重点介绍。

#### **bit13: RF 波特率选择**

nRF2401A 通信的波特率选择位，用于收发。如果该位为“0”，则选择 250Kbps 波特率；如果该位为“1”，则选择 1Mbps 波特率。前者的通信误码率较低，如果速度没有严格要求，通常选择 250 Kbps。

#### **bit12~bit10: 芯片外晶振选择**

nRF2401A 的外晶振选择，nRF2401A 支持 5 种不同频率的晶振，随着外晶振频率的不同，**bit12~bit10** 值也不同，其对应关系如下：

表 3.1 晶振设置对照表

bit12~bit10	晶振频率 (MHz)
000	4
001	8
010	12
011	16
100	20

模组选用 16M 晶振，所以 **bit12~bit10** 值为 011。

#### **bit9~bit8: RF 发射功率选择**

nRF2401A 发射功率选择，用于发射。nRF2401A 有 4 种不同的发射功率，与 **bit9~bit8** 的对应关系如下：

表 3.2 发射功率对照表

bit9~bit8	发射功率 (dBm)
00	-20
01	-10
10	-5
11	0

通常我们选择最大发射功率 0dBm，取 **bit9~bit8** 值为 11。

#### **bit7~bit1: RF 频率选择**

nRF2401A 的收发频率选择，用于收发。如果 **bit7~bit1** 的值为  $n$ ，通道 1 的收发频率为  $(2400+n)$  MHz；通道 2 的接收频率为  $(2400 + n + 8)$  MHz。在通信时，只有发射端和接收端的频率一致，发射端的信号才可能被接收。

#### **bit0: 芯片收发切换**

nRF2401A 收发选择位。如果该位为“0”，则芯片工作于发送状态；如果该位为“1”，则芯片工作于接收状态。

向 nRF2401A 写控制字操作须严格按照图 3.1 的时序进行：

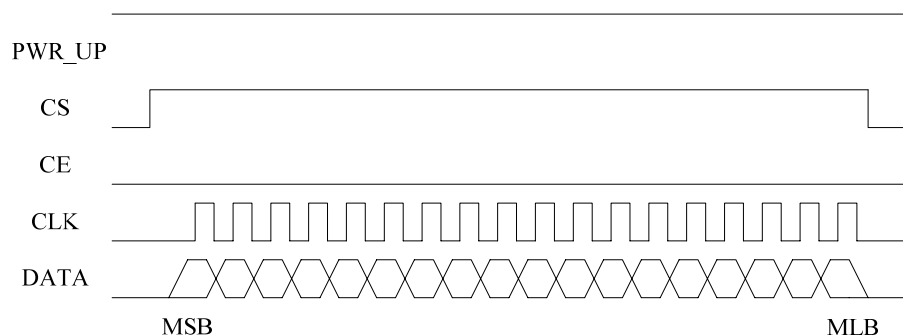


图 3.2 向 nRF2401A 写命令字时序图

PWR\_UP 为高、CE 为低时，置位 CS，芯片处于命令字写入状态，通过通道 1 向芯片的控制字缓冲区写入命令字，按照由高位到低位的顺序，命令字全部写入后，将 CS 置低，nRF2401A 芯片将会根据命令字配置相应的内部模块。在第一次配置操作结束后，只有最后两个字节命令字可以被更改，前 16 个字节的修改无效，如果需要修改前 16 个字节的命令字（如通道接收地址，接收数据长度等），则需要掉电（PWR\_UP 置低）后重新上电（PWR\_UP 置高），才能对芯片进行彻底初始化操作。

需要注意的地方：

- 1、并不是 18 个 Byte 的命令字全部需要写入，通常我们只需要写后 15 个 Byte 即可；
- 2、15 个 Byte 的命令字在 nRF2401.C 文件中通过 InitData[15]定义，用户需根据自己的通信格式对其进行修改；
- 3、bit15~bit0 可以根据需要反复修改，但是 bit144~bit16 只有上电后的第一次初始化有效。

## 3.2 数据发送程序

nRF2401A 采用 Shock Burst 方式发送数据。

单片机向 nRF2401A 传送数据的函数有以下 4 个：

void nRF2401\_SendByte(unsigned int TxData)，用于发送一个字节数据；

void nRF2401\_SendWord(unsigned int TxData)，用于发送一个字数据；

void nRF2401\_SendBuffer\_Byte(unsigned int TxData[],unsigned int DataByte)，用于发送一个字节数组；

void nRF2401\_SendBuffer\_Word(unsigned int TxData[],unsigned int DataWord)，用于发送一个字数组。

以上几个函数的流程是一样的，只是发送数据的长度不同。

单片机向 nRF2401A 发送数据流程如图 3.1 所示：

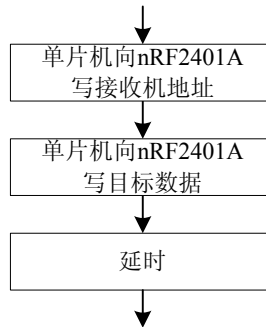


图 3.3 数据发送程序流程图

单片机向 nRF2401A 发送数据的时序如下：

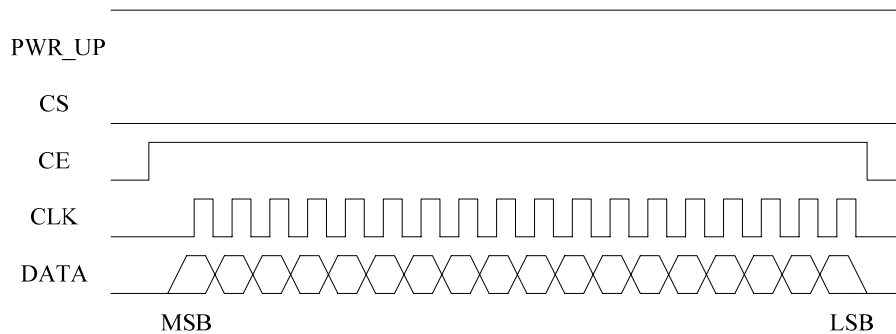


图 3.4 单片机向 nRF2401A 发送数据时序图

MSB→LSB 数据格式为如下：

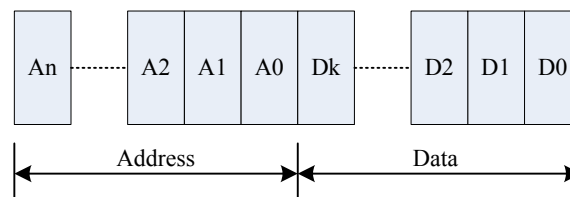


图 3.5 单片机向 nRF2401A 发送数据格式

$A_n \sim A_0$  为接收机地址，不超过 40 位，通过更换地址，可以向多个 nRF2401A 模块发送数据； $D_k \sim D_0$  为待发送的数据。

以上数据由单片机发送到 nRF2401A 之后，nRF2401A 将会进行打包并发射，打包后的数据格式为：



图 3.6 nRF 对外发送数据的打包格式

其中 Pre 为 8 位的校验头，CRC 为 8 位或 16 位的校验尾，在 Shock Burst 模式下由 nRF2401A 自动添加。

需要注意的地方：

- 1、Address、Data、CRC 的位数之和不超过 256;
- 2、Address 长度必须和目标接收通道的地址一致（数据和长度）;
- 3、Data 长度必须和目标接收通道的接收数据宽度一致。
- 4、发射端和接收端的 RF 频率须一致，如两端均采用通道 1，则二者控制字的 bit7~bit1 一致，如接收端采用通道 2，则发射端 bit7~bit1 值比接收端 bit7~bit1 值大 8。

函数的使用:

**void nRF2401\_SetAddress(unsigned int Address[], unsigned int AddrByte):**

设置目标 nRF2401A 的地址，发送端在发送数据前调用，只需要一次即可。其中 Address[] 数组为目标 nRF2401A 地址，AddrByte 为有效地址的字节数。

**void nRF2401\_SendByte(unsigned int TxData):**

用于向 nRF2401A 发送一帧数据（目标地址 + 一个字节数据 TxData）;

**void nRF2401\_SendWord(unsigned int TxData):**

用于向 nRF2401A 发送一帧数据（目标地址 + 一个字数据 TxData）;

**void nRF2401\_SendBuffer\_Byte(unsigned int TxData[], unsigned int DataByte):**

用于向 nRF2401A 发送一帧数据（目标地址 + 一个字节数组 TxData[]），其中 DataByte 为数组元素个数;

**void nRF2401\_SendBuffer\_Word(unsigned int TxData[], unsigned int DataWord):**

用于向 nRF2401A 发送一帧数据（目标地址 + 一个字数组 TxData），其中 DataWord 为数组元素个数;

### 3.3 数据接收程序

当接收端成功接收到数据后，将会置位对应的数据请求管脚 DR1/DR2，单片机通过查询该管脚状态，或者通过中断方式接收数据，nRF2401.C 文件提供的从 nRF2401A 读取数据的函数有两个:

**void nRF2401\_ReceiveByte(unsigned int RxData[]):** 以 Byte 格式从 nRF2401A 读取数据;

**void nRF2401\_ReceiveWord(unsigned int RxData[]):** 以 Word 格式从 nRF2401A 读取数据。

以上两个函数的操作流程是一样的，只是存储的数据格式不同。数据接收流程见图 3.7:

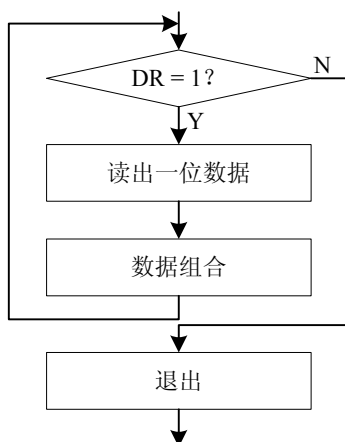


图 3.7 单片机读取数据流程图

单片机从 nRF2401A 读取数据的时序如下：

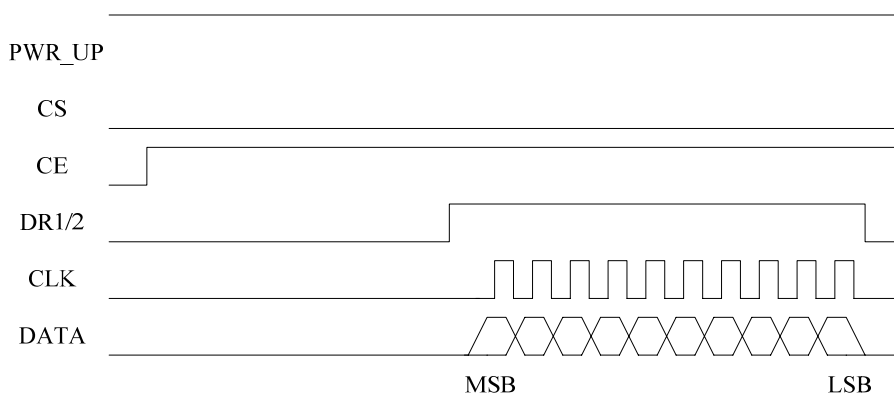


图 3.8 单片机从 nRF2401A 读取数据时序图

函数的使用：

**void nRF2401\_ReceiveByte(unsigned int RxData[]):**

以 Byte 格式从 nRF2401A 读取数据，接收数据存储在数组 RxData[]中。

**void nRF2401\_ReceiveWord(unsigned int RxData[]):**

以 Word 格式从 nRF2401A 读取数据，接收数据存储在数组 RxData[]中。

### 3.4 其他程序

其他程序如下：

**void nRF2401\_ON():**

nRF2401A 上电操作，在初始化 nRF2401A 之前执行该操作；

**void nRF2401\_OFF():**

nRF2401A 掉电操作，在需要重新初始化 nRF2401A 时执行该操作；

**void nRF2401\_Mode(unsigned int uiMode):**

nRF2401A 收/发模式切换，uiMode = 0，设置 nRF2401A 为接收；uiMode = 1，设置 nRF2401A 为发送；

**unsigned int nRF2401\_RxStatus():**

判断 nRF2401A 是否有数据接收，如果已经接收到数据，返回值为“1”，如果没有接收到数据，返回值为“0”；

## 4 编程举例

### 4.1 单向数据传输

以两套无线模组为例，其中一套（A 套）发送数据，另外一套（B 套）接收数据。（）

#### 4.1.1 要求

地址：32 bit 地址（设为：0x00 00 01 FF）

码率：250Kbps

数据：每帧 5 Byte（设为：0x01, 0x02, 0x03, 0x04, 0x05）

频率：2.45GHz

通道：发射端和接收端均采用通道 1

CRC：16 位

#### 4.1.2 分析

对照 3.1 节控制字描述，发送端控制字应为：

bit17~bit16: 11, 16 位 CRC

bit14: 1, Shock Burst 模式

bit13: 0, 波特率 250Kbps

bit12~bit10: 011, 16M 晶振

bit9~bit8: 11, 发射功率 0dBm

bit7~bit1: 0110010 (50)，收发频率 2.45GHz

bit0: 0, 发送模式

所以发送端的初始化命令字 InitData[15] =

```
{  
    0x00,           // 通道二接收数据宽度，发送无效  
    0x00,           // 通道一接收数据宽度，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道二地址，发送无效  
    0x00,           // 通道一地址，发送无效
```



```
0x00,          // 通道一地址, 发送无效
0x00,          // 通道一地址, 发送无效
0x00,          // 通道一地址, 发送无效
0x00,          // 通道一地址, 发送无效
0x03,          // 接收地址宽度, 发送无效
                // b17~b16:CRC 模式选择(11-16bit)
0x4F,          // b15:接收通道选择, 发送无效
                // b14:收发模式(1-ShockBurst 模式)
                // b13:波特率选择(0-250Kbps)
                // b12~b10:晶振选择(011-16M)
                // b9~b8:发射功率(11-0db)
0x64           // b7~b1: 发射频率 (0110010, 2.45GHz)
                // b0:收发控制(0-发送)

}
```

发送端程序为:

```
#include "nRF2401.h"
#include "SPCE061A.h"

unsigned int Address[4] = {0x00,0x00,0x01,0xFF};          // 接收端地址:00 00 01 FF
unsigned int SendBuf[5] = {0x01,0x02,0x03,0x04,0x05};    // 待发送数据:01 02 03 04 05

int main()
{
    nRF2401_Initial();          // 初始化 nRF2401A
    nRF2401_SetAddress(Address,4); // 设置接收端地址 00 00 01 FF
    nRF2401_SendBuffer_Byte(SendBuf,5); // 发送数组

    while(1)
    {
        *P_Watchdog_Clear = 0x1;
    }
}
```

```

    }
}

```

对照 3.1 节控制字描述，接收端控制字应为：

bit111~bit104: 00101000 (0x28)，接收数据宽度 5 Byte

bit63~bit24: 00000000 00000000 00000000 00000001 11111111 (0x00000001FF)，通道 1 地址

bit23~bit18: 100000 (0x20)，通道 1 地址有效位 32 位，故通道 1 地址为 (0x000001FF)

bit17~bit16: 11，16 位 CRC

bit15: 0，单通道（通道 1 接收，通道 2 不接收）

bit14: 1，Shock Burst 模式

bit13: 0，波特率 250Kbps

bit12~bit10: 011，16M 晶振

bit7~bit1: 0110010 (0x32)，收发频率 2.45GHz

bit0: 1，接收模式

所以发送端的初始化命令字 InitData[15] =

```

{
    0x00,          // 通道二接收数据宽度
    0x28,          // 通道一接收数据宽度
    0x00,          // 通道二地址
    0x00,          // 通道二地址
    0x00,          // 通道二地址
    0x00,          // 通道二地址
    0x00,          // 通道二地址
    0x00,          // 通道一地址
    0x00,          // 通道一地址
    0x00,          // 通道一地址
    0x01,          // 通道一地址
    0xFF,          // 通道一地址
    0x83,          // 接收地址宽度（32 位）
                  // b17~b16: CRC 模式选择(11-16bit)
    0x4F,          // b15: 接收通道选择，通道 1 接收
                  // b14: 收发模式(1-ShockBurst 模式)
}

```

```
                                // b13: 波特率选择(0-250Kbps)
                                // b12~b10: 晶振选择(011-16M)
0x65                            // b7~b1: 收发频率 (0110010, 2.45GHz)
                                // b0: 收发控制(1-接收)

}
```

接收端程序为:

```
#include "SPCE061A.h"
#include "nRF2401A.h"

unsigned int RxBuf[5];           // 接收缓冲,保存接收到的数据

int main()
{
    nRF2401_Init();              // nRF2401A 初始化
    while((nRF2401_RxStatus()) != 1) // nRF2401A 没有数据请求
    {
        *P_Watchdog_Clear = 0x0001;
    }
    nRF2401_ReceiveByte(RxBuf);   // 接收数据

    while(1)
    {
        *P_Watchdog_Clear = 0x0001;
    }
}
```

以上程序位于\\nRF2401 无线传输模块\Demo Code\Exa1\Send 和 Receive 中。在 While()循环中设置断点,并全速运行,然后复位发送端,接收端接收到数据后将会在断点处停下。

## 4.2 双向数据传输

两套无线模组为例, A 套发送数据时, B 套接收; B 套发送数据时, A 套接收; A 向 B 发送一个数

据之后进入接收状态，B 接收到数据后返回 A，A 接收到数据后进入 Sleep。

#### 4.2.1 要求

地址：32 bit 地址（A 套和 B 套接收端地址一样：0x00 00 00 01）

码率：250Kbps

数据：每帧 1Byte

频率：2.40GHz

通道：通道 1

CRC：16 位

#### 4.2.2 分析

A 套作为发送的控制字：

bit17~bit16: 11, 16 位 CRC

bit15: 0, 单通道（通道 1 接收，通道 2 不接收）

bit14: 1, Shock Burst 模式

bit13: 0, 波特率 250Kbps

bit12~bit10: 011, 16M 晶振

bit9~bit8: 11, 发射功率 0dBm

bit7~bit1: 0000000 (0x0), 收发频率 2.4GHz

bit0: 0, 发送模式。

A 套作为接收的控制字：

bit111~bit104: 00001000 (0x08), 接收数据宽度 1 Byte

bit63~bit24: 00000000 00000000 00000000 00000000 00000001 (0x0000000001), 通道 1 地址

bit23~bit18: 100000 (0x20), 通道 1 地址有效位 32 位，故通道 1 地址为 (0x00000001)

bit17~bit16: 11, 16 位 CRC

bit15: 0, 单通道（通道 1 接收，通道 2 不接收）

bit14: 1, Shock Burst 模式

bit13: 0, 波特率 250Kbps

bit12~bit10: 011, 16M 晶振

bit7~bit1: 0000000 (0x0), 收发频率 2.4GHz

bit0: 1, 接收模式。

组合可得到既适合 A 套发送又适合 A 套接收的初始化命令字 InitData[15] =

```
{  
    0x00,          // 通道二接收数据宽度  
    0x08,          // 通道一接收数据宽度  
    0x00,          // 通道二地址  
    0x00,          // 通道二地址  
    0x00,          // 通道二地址  
    0x00,          // 通道二地址  
    0x00,          // 通道二地址  
    0x00,          // 通道一地址  
    0x00,          // 通道一地址  
    0x00,          // 通道一地址  
    0x00,          // 通道一地址  
    0x01,          // 通道一地址  
    0x83,          // 接收地址宽度（32 位）  
                    // b17~b16: CRC 模式选择(16bit)  
    0x4F,          // b15: 接收通道选择，通道 1 接收  
                    // b14: 收发模式(ShockBurst 模式)  
                    // b13: 波特率选择(250Kbps)  
                    // b12~b10: 晶振选择(16M)  
    0x64           // b7~b1: 收发频率（2.45GHz）  
                    // b0: 收发控制(接收/发送)  
}
```

只需要在程序中切换控制字 bit0 的值，即可切换 nRF2401A 无线模块的收发，该切换操作由函数 nRF2401A\_Mode()实现。

B 套采用和 A 套一样的初始化命令字。

A 套程序如下：

```
#include "nRF2401A.h"  
  
#include "SPCE061A.h"
```

```
unsigned int Address[4] = {0x00,0x00,0x00,0x01};    // 接收端地址:00 00 00 01
unsigned int RxData[1];                             // 接收

int main()
{
    nRF2401_Initial();                               // 初始化 nRF2401A
    nRF2401_SetAddress(Address,4);                   // 设置接收端地址 00 00 00 01
    nRF2401_Mode(1);                                 // 发送

    nRF2401_SendByte(1);                             // 发送数据
    Delay_Us(1000);

    nRF2401_Mode(0);                                 // 接收
    while(nRF2401_RxStatus() == 0)
    {
        *P_Watchdog_Clear = 0x1;
    }
    nRF2401_ReceiveByte(RxData);                     // 接收到数据

    *P_SystemClock = 0x0007;                         // 睡眠
    while(1)
    {
        *P_Watchdog_Clear = 0x1;
    }
}
```

B 套程序如下:

```
#include "nRF2401A.h"
#include "SPCE061A.h"
```

```
unsigned int Address[4] = {0x00,0x00,0x00,0x01};    // 接收端地址:00 00 00 01
unsigned int RxData[1];                             // 接收

int main()
{
    nRF2401_Initial();                              // 初始化 nRF2401A
    nRF2401_SetAddress(Address,4);                  // 设置接收端地址 00 00 00 01
    nRF2401_Mode(0);                                // 接收
    while(nRF2401_RxStatus() == 0)
    {
        *P_Watchdog_Clear = 0x1;
    }
    nRF2401_ReceiveByte(RxData);

    Delay_Us(10000);

    nRF2401_Mode(1);                                // 发送
    nRF2401_SendByte(RxData[0]);
    Delay_Us(1000);
    *P_SystemClock = 0x0007;                        // 睡眠
    while(1)
    {
        *P_Watchdog_Clear = 0x1;
    }
}
```

以上程序位于\\nRF2401 无线传输模块\Demo Code\Exa2\Send 和 Receive 中。下载程序后首先复位接收端单片机，然后再复位发送端单片机，发送端和接收端单片机都进入睡眠。按照接收端→发送端顺序复位，重复试验现象。

## 5 常见问题解答

Q: 为什么通信不成功?

A: 通信不成功的原因可能在发射端也可能在接收端, 重点检查以下几个参数:

- 1、发送端发送的地址与接收端的通道地址(包括地址值和有效位)是否一致;
- 2、发送端发送的数据宽度和接收端的设置是否一致;
- 3、发送端的发射频率与接收端的接收频率是否一致;

Q: 4.1 的例子如何观察现象?

A: 可以在接收程序中接收函数后面设置一个断点, 并运行接收端程序; 复位发送端, 接收端会收到一帧数据。

Q: 4.2 的例子如何观察现象?

A: 先复位接收端单片机, 运行其程序; 然后复位发送端单片机, 则接收端和发送端单片机都将进入睡眠。



## 6 附录

### 6.1 配件清单及配件说明

无线传输模组	1 个
产品说明书	1 本

### 6.2 相关资料清单

无线传输模组相关的资料如下：

无线传输模组使用说明书；

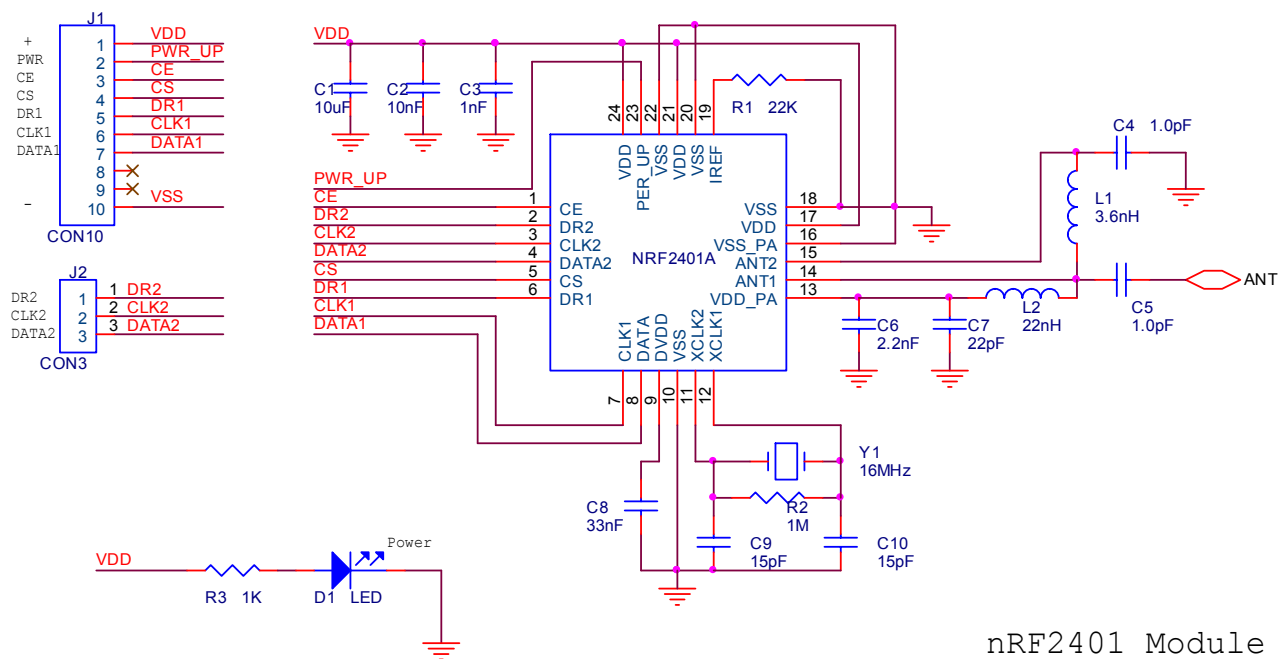
无线传输模组产品说明书；

无线传输模组配套程序；

电路原理图（orCAD 格式以及 PDF 文档）、PCB 图。

以上资料可在凌阳大学计划网站：[www.unsp.com.cn](http://www.unsp.com.cn) 的下载专区中下载。

## 6.3 电路原理图



nRF2401 Module

## 6.4 公司联系方式

- ✦ 客服专线：010-62981668-2911
- ✦ 技术支持：010-62981668-2916/2917/2965
- ✦ 传真号码：010-62962425
- ✦ 咨询信箱：unsp@sunplus.com.cn
- ✦ 欢迎登陆：<http://www.unsp.com.cn>
- ✦ 技术论坛：<http://www.unsp.com.cn/dvbbs/>
- ✦ 邮政编码：100085
- ✦ 联系地址：北京市海淀区上地信息产业基地中黎科技园 1 号楼 5 层