RF12B 编程使用说明

1. 概述

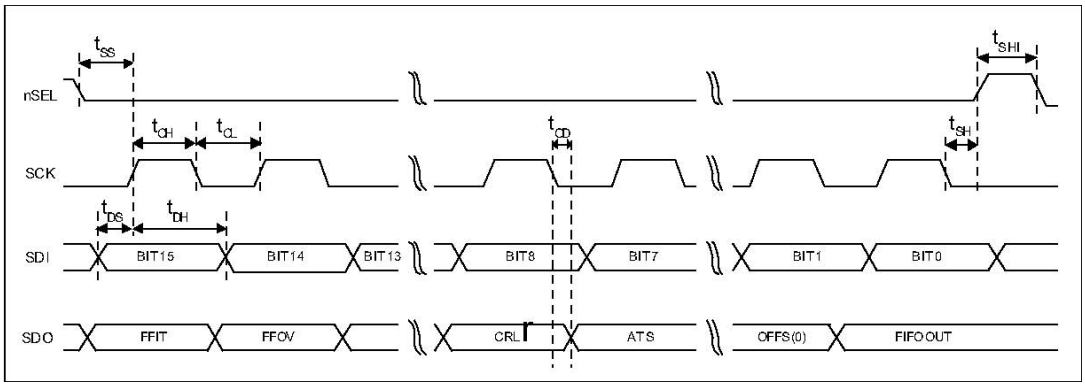
RF12B 是一款低成本高集成的 FSK 收发一体 IC,其内部集成了所有的 RF 功能模块电路,外围只须一个 MCU,一个晶振,一个旁路电容和一个外置天线就可组成一个带有 PLL 技术的高可靠性的收发系统,具有设计简单,生产无需调试的特点.可工作在 433/868/915MHZ 频段.两个 RF12B 即可组成一个完整的双向收发系统.在无需外加功放电路的情况下，距离可达到 150 米以上.

RF12B 还集成了一个数字接口,轻易实现由 MCU 通过软件设置,就可精确调整各种射频参数(如中心频点,接收带宽等).而无需调整硬件电路,可轻易实现跳频功能.

RF12B 可应用于无线防盗和报警系统,无线传感器，无线键盘和鼠标,家居自动化遥控,无线高速数据采集系统,无线玩具等场合

2. 控制命令

1. SPI 接口时序



2. 配置设置命令（Configuration Setting Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	0	0	0	0	0	0	el	ef	b1	b0	x3	x2	x1	x0	8008h

e l: 使能内部发射寄存器

e f: 使能内部 FIFO 寄存器

b1..b0: 波段选择：

b1	b0	工作频段 [MHz]
0	0	保留
0	1	433
1	0	868
1	1	915

x3..x0: 选择晶振负载电容：

x3	x2	x1	x0	晶振负载电容 [pF]
0	0	0	0	8.5
0	0	0	1	9.0
0	0	1	0	9.5
0	0	1	1	10.0
……		……		
1	1	1	0	15.5
1	1	1	1	16.0

3. 电源管理命令（Power Management Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	0	0	0	0	1	0	er	ebb	et	es	ex	eb	ew	dc	8208h

er： 打开接收机

ebb：打开基带电路

et： 打开发射机

es： 打开频率合成器

ex： 打开晶体振荡器

eb： 打开低压检测器

ew： 打开唤醒定时器

dc： 禁止时钟输出

4. 频率设置命令（Frequency Setting Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	1	0	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1	f0	A680h

f11..f0: 用于设置工作频率：

433 频段：Fc=430+F*0.0025 MHz

868 频段：Fc=860+F*0.0050 MHz

915 频段：Fc=900+F*0.0075 MHz

Fc 为发射机中心频率，F 为频率参数，36≤F≤3903

5. 数据速率命令（Data Rate Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	1	1	0	cs	r6	r5	r4	r3	r2	r1	r0	C623h

r7..r0: 用于设置数据速率：

BR=10000000/29/（R+1）/（1+cs*7）

BR 为数据速率，R 为数据速率参数

6. 接收机控制命令（Receiver Control Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	0	1	0	p20	d1	d0	i2	i1	i0	g1	g0	r2	r1	r0	9080h

p20: 选择引脚 20 的功能：

p20	功能
0	外部中断输入
1	VDI 输出

i2..i0:选择接收带宽：

i2	i1	i0	带宽 [kHz]
0	0	0	保留
0	0	1	400
0	1	0	340
0	1	1	270
1	0	0	200
1	0	1	134
1	1	0	67
1	1	1	保留

d1..d0: 选择 VDI 输出响应时间

d1	d0	响应时间
0	0	最快
0	1	中等
1	0	较慢
1	1	常开

g1..g0: 选择 LNA 增益

g1	g0	LNA 增益(dBm)
0	0	0
0	1	-6
1	0	-14
1	1	-20

r2..r0: 选择 DRSSI 门限

r2	r1	r0	RSSI 设置门限 [dBm]
0	0	0	-103
0	0	1	-97
0	1	0	-91
0	1	1	-85
1	0	0	-79
1	0	1	-73
1	1	0	保留
1	0	1	保留

实际 DRSSI 门限与 LNA 增益设置有关，由下式确定：

$RSSI_{th} = RSSI_{setth} + G_{LNA}$

$RSSI_{th}$：实际 DRSSI 门限，$RSSI_{setth}$：所设 DRSSI 门限，G_{LNA}：LNA 增益。

7. 数据滤波命令（Data Filter Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	0	1	0	al	ml	1	s	1	f2	f1	f0	C22Ch

al：启用时钟恢复自动锁定

ml:时钟恢复快速锁定使能

s：选择滤波类型：

s	滤波类型
0	数字滤波
1	模拟 RC 滤波

f1..f0: 设置门限 DQD 参数

8. FIFO 和复位模式命令（FIFO and Reset Mode Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	0	1	0	f3	f2	f1	f0	sp	al	ff	dr	CA80h

f3..f0: 设置 FIFO 中断门限

sp: 选择同步字长度

sp	同步字高字节	同步字低字节 (复位值)	同步字
0	2Dh	D4h	2DD4h
1	未用	D4h	D4h

al: 设置 FIFO 填充条件：

al	填充条件
0	同步字
1	一直填充

ff: FIFO 填充允许

dr: 禁止高灵敏复位模式

9. 同步字命令（Synchron pattern Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	1	1	0	b7	b6	b5	b4	b3	b2	b1	b0	CED4h

同步字可用此命令重新设置

10. 接收机 FIFO 读出命令（Receiver FIFO Read Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	B000h

当产生 FFIT 中断时，使用该命令从接收机读出 FIFO 数据，数据在第 8 个 SCK 开始输出

11. **AFC 命令（AFC Command）**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	1	0	0	a1	a0	rl1	rl0	st	fi	oe	en	C4F7h

a1..a0: 选择 AFC 自动模式：

a1	a0	
0	0	关闭自动模式 (由 MCU 控制
0	1	上电运行一次
1	0	VDI 有效是保留偏移值
1	1	一直保持，与 VDI 无关

rl1..rl0: 选择频率漂移范围：

r1	r0	漂移范围（单位：频率间隔
0	0	无限制
0	1	+15/-16
1	0	+7/-8
1	1	+3-4

频率间隔：

315，433 频段：2.5kHz

868 频段：5kHz

915 频段：7.5kHz

st: 上升沿将最近频率漂移存入 AFC 输出寄存器。

fi: AFC 高精度使能

oe: 使能 AFC 输出寄存器

en: 打开 AFC 功能

12. 发射机配置控制命令（AFC Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	0	1	1	0	0	mp	m3	m2	m1	m0	0	p2	p1	p0	9800h

mp： 调制极性选择

m2..m0:用于选择调制频偏：

m3	m2	m1	m0	调制频偏 [kHz]
0	0	0	0	15
0	0	0	1	30
0	0	1	0	45
0	0	1	1	60
0	1	0	0	75
0	1	0	1	90
0	1	1	0	105
0	1	1	1	120
1	0	0	0	135
1	0	0	1	150
1	0	1	0	165
1	0	1	1	180
1	1	0	0	195
1	1	0	1	210
1	1	1	0	225
1	1	1	1	240

p2..p0: 用于选择发射功率：

p2	p1	p0	发射功率[dBm]
0	0	0	0
0	0	1	-3
0	1	0	-6
0	1	1	-9
1	0	0	-12
1	0	1	-15
1	1	0	-18
1	0	1	-21

13. PLL Setting Command

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	1	0	0	0	ob1	ob0	1	ddy	ddit	1	bw0	CC77h

注意： 使用A0版本的IC时， 默认值为CC67,在程序中必须执行CC77，使PLL不在默认状态使用。

使用A1版本的IC， 默认值为CC77.

ob1-ob0: CLK 时钟输出上升下降时间控制：

ob1	ob0	所选 CLK 频率
0	0	5 or 10 MHz (推荐)
0	1	3.3 MHz
1	X	2.5 MHz or less

ddy: 使能相位检测延迟

ddi: 禁止PLL环路抖动.

bw1-bw0: 选择PLL带宽

bw0	最高速率 [kbps]	离中心频点 1MHz 位置处相位噪声[dBc/Hz]
0	86.2	-107
1	256	-102

14. 发射寄存器写命令（Transmitter Register Write Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	0	1	1	1	0	0	0	t7	t6	t5	t4	t3	t2	t1	t0	B8AAh

该命令用于将数据字节写入发射机以待发射

15. 唤醒定时器命令（Wake-Up Timer Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	1	r4	r3	r2	r1	r0	m7	m6	m5	m4	m3	m2	m1	m0	E196h

唤醒定时器唤醒时间由下面公式确定：

$T_{wake-up} = M * 2^{R}$ [ms]

16. 低占空比命令（Low Duty-Cycle Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	1	0	0	0	d6	d5	d4	d3	d2	d1	d0	en	C8OEh

d6..d0: 设置接收占空比：

D.C.= (D * 2 +1) / M *100%

D.C.：接收占空比，D：占空比参数，M：唤醒定时器参数尾数

en：使能低占空比模式

17. 低压检测及时钟输出分频命令（Low Battery Detector and Microcontroller Clock Divider Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	1	1	0	0	0	0	0	0	d2	d1	d0	0	v3	v2	v1	v0	C000h

d2..d0: 选择时钟引脚（CLK）输出频率：

d2	d1	d0	时钟频率[MHz]
0	0	0	1
0	0	1	1.25
0	1	0	1.66
0	1	1	2
1	0	0	2.5
1	0	1	3.33
1	1	0	5
1	1	1	10

此时钟信号直接由晶体振荡器分频得到，可以作为 MCU 时钟信号输入，从而省掉 MCU 晶体以降低系统成本

如未用此信号，可以通过置位“dc”位（见电源管理命令）禁止输出。

内部集成可变晶体负载电容，一方面可节省成本，另一方面可以通过调整负载电容获得精确的参考频率。

v3..v0:设置低压检测门限：

Vlb=2.2+V*0.1 [V]

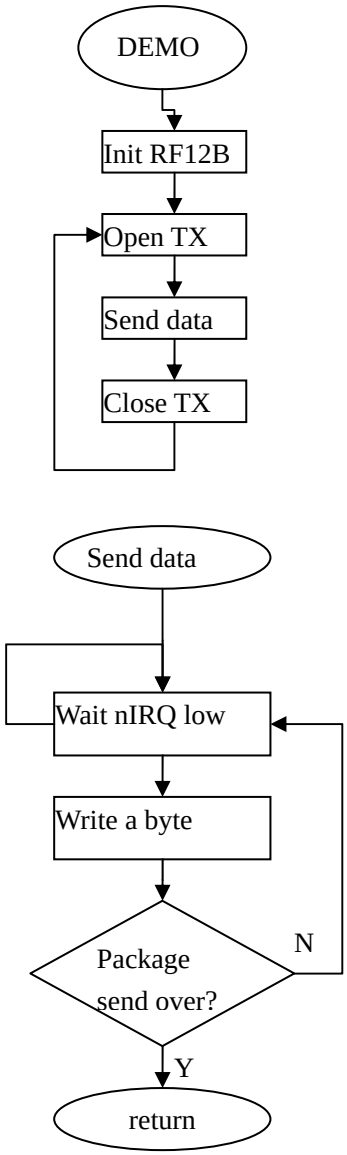
18. 状态读出命令（Status Read Command）

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POR
	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-

该命令以 0 开始，用于读出 RF12B 内部状态寄存器内容。

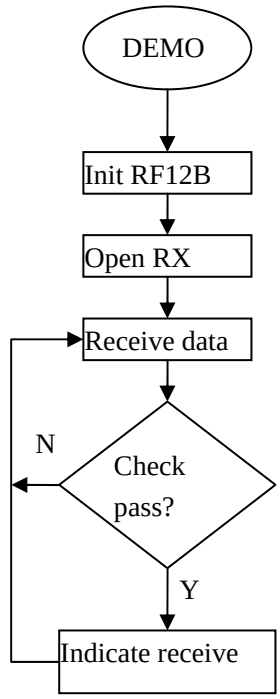
3. 演示流程

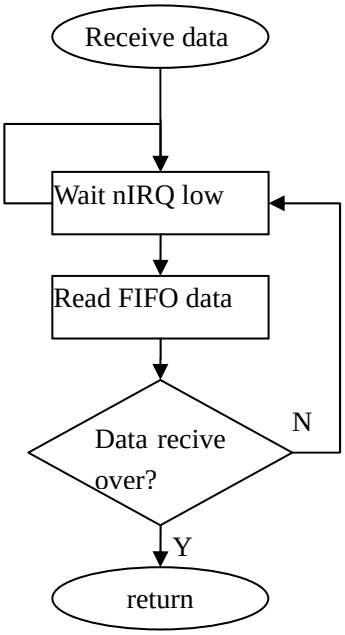
发射：



说明： 初始化 RF12B 并打开发射机，RF12B 发射完一个字节后会将 nIRQ 拉低，通知 MCU 写入后续字节以待发射。

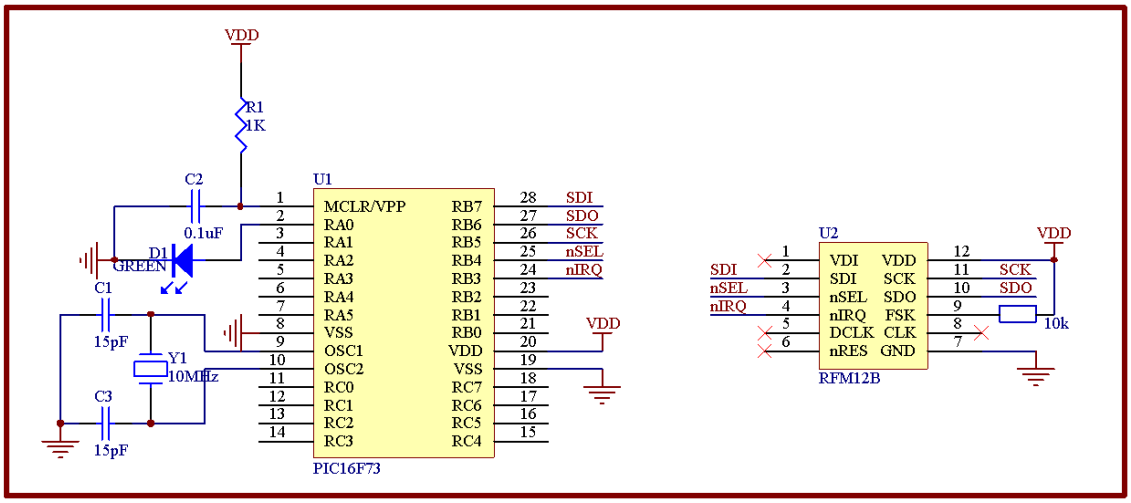
接收：





说明：RF12B 参数配置完毕之后，打开 FIFO 接收模式。接收到数据之后会将 nIRQ 拉低通知 MCU 读取数据。数据包接收完毕之后务必将 FIFO 复位以便后续数据包接收。

4. 实例代码（PIC）



RF12 发射演示

```
/*********************************************************
    copyright (c) 2010
Title:              RFM12B transmitter simple example based on PIC C
Current version:    v1.1
Function:           Package send Demo
Processor           PIC16F73 DIP-28
Clock:              10MHz Crystal
Operate frequency:  434MHz
Data rate:          4.8kbps
Package size:       23byte
Author:             Simon.Yang
Company:            Hope microelectronic Co.,Ltd.
Contact:            +86-0755-82973805
E-MAIL:             faerf@hoperf.com
Date:               2010-06-28
*********************************************************/
#include "pic.h"

typedef unsigned char uchar;
typedef unsigned int  uint;

#define  SDI            RB7
#define  SDO            RB6
#define  SCK            RB5
#define  nSEL           RB4
#define  LED            RA0
```

```
#define  SDI_OUT()     TRISB7=0
#define  SDO_IN()      TRISB6=1
#define  SCK_OUT()     TRISB5=0
#define  nSEL_OUT()    TRISB4=0
#define  LED_OUT()     TRISA0=0

void Init_RF12(void);
void Write0( void );
void Write1( void );
void WriteCMD( uint CMD );
void DelayUs( uint us );
void DelayMs(uint ms);
void WriteFSKbyte( uchar DATA );

__CONFIG(0x3FF2);

void Init_RF12(void)
{
  LED_OUT();
  LED=0;
  nSEL_OUT();
  SDI_OUT();
  SDO_IN();
  SCK_OUT();
  nSEL=1;
  SDI=1;
  SCK=0;
  WriteCMD(0x80D8); //使能发射/FIFO 寄存器,433MHz,12.5pF
  WriteCMD(0x8208); //打开晶体振荡器
  WriteCMD(0xA640); //频率设置 434MHz
  WriteCMD(0xC647); //数据速率 4.8KHz
  WriteCMD(0x94A0); //VDI 最快,择接收带宽:134kHz,LNA 增益:0dBm,DRSSI:-103dBm
  WriteCMD(0xC2AC); //数字滤波,DQD
  WriteCMD(0xCED4); //同步字低位设为 D4
  WriteCMD(0xCA80); //
  WriteCMD(0xCA83); //FIFO 中断门限 8bit,SYNC 填充,FIFO 填充允许, 同步字 2DD4
  WriteCMD(0xC49B); //使能 AFC,调制频漂 +37.5/-40 Khz
  WriteCMD(0x9850); //!mp,调制频偏 90kHz, 发射功率 0dBm
  WriteCMD(0xCC77); // PLL 设置
  WriteCMD(0xE000); //唤醒定时器命令,NOT USED
  WriteCMD(0xC80E); //低占空比命令,NOT USED
  WriteCMD(0xC000); // CLK OUT 1.0MHz,低压 LBD 2.2V
}
```

```
void main()
{
  uint ChkSum;
  Init_RF12();
  while(1)
   {
     ChkSum=0;
     WriteCMD(0x8228);     //打开接收机
     DelayUs( 4 );
     WriteCMD(0x8238);
     NOP();
     NOP();
     WriteFSKbyte( 0xAA ); //引导码
     WriteFSKbyte( 0xAA );
     WriteFSKbyte( 0xAA );
     WriteFSKbyte( 0x2D ); //同步码
     WriteFSKbyte( 0xD4 );

     WriteFSKbyte( 0x30 );//DATA0
     ChkSum+=0x30;
     WriteFSKbyte( 0x31 );//DATA1
     ChkSum+=0x31;
     WriteFSKbyte( 0x32 );
     ChkSum+=0x32;
     WriteFSKbyte( 0x33 );
     ChkSum+=0x33;
     WriteFSKbyte( 0x34 );
     ChkSum+=0x34;
     WriteFSKbyte( 0x35 );
     ChkSum+=0x35;
     WriteFSKbyte( 0x36 );
     ChkSum+=0x36;
     WriteFSKbyte( 0x37 );
     ChkSum+=0x37;
     WriteFSKbyte( 0x38 );
     ChkSum+=0x38;
     WriteFSKbyte( 0x39 );
     ChkSum+=0x39;
     WriteFSKbyte( 0x3A );
     ChkSum+=0x3A;
     WriteFSKbyte( 0x3B );
     ChkSum+=0x3B;
     WriteFSKbyte( 0x3C );
     ChkSum+=0x3C;
```

```
      WriteFSKbyte(0x3D);
      ChkSum+=0x3D;
      WriteFSKbyte( 0x3E );
      ChkSum+=0x3E;
      WriteFSKbyte( 0x3F );//DATA15
      ChkSum+=0x3F;
      ChkSum&=0x0FF;
      WriteFSKbyte( ChkSum );
      WriteFSKbyte( 0xAA );
      WriteCMD( 0x8208 );       //关掉接收机
      WriteCMD( 0x8200 );      //接收完毕,进入 sleep
      LED=1;
      DelayMs(100);
      LED=0;
      DelayMs(1000);
    }
}

void Write0( void )
{
  SDI=0;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void Write1( void )
{
```

```
  SDI=1;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void WriteCMD( uint CMD )
{
  uchar n=16;
  SCK=0;
  nSEL=0;
  while(n--)
   {
     if(CMD&0x8000)
      Write1();
     else
      Write0();
     CMD=CMD<<1;
   }
  SCK=0;
  nSEL=1;
}

void WriteFSKbyte( uchar DATA )
{

  uchar RGIT=0;
  uint temp=0xB800;
```

```
  temp|=DATA;
Loop: SCK=0;
  nSEL=0;
  SDI=0;
  SCK=1;
  if(SDO)                //Polling SDO
   {
    RGIT=1;
   }
  else
   {
    RGIT=0;
   }
  SCK=0;
  SDI=1;
  nSEL=1;
  if(RGIT==0)
   {
    goto Loop;
   }
  else
  {
   RGIT=0;
   WriteCMD(temp);
  }
}


void DelayUs( uint us )
{
  uint i;
  while( us-- )
    {
       i=2;
       while( i-- )
         {
           NOP();
         }
    }
}

void DelayMs(uint ms)
{
  uchar i;
```

```
  while(ms--)
  {
   i=9;
   while(i--)
   {
     DelayUs(1);
   }
  }
}
```

RF12 接收演示

```
/*************************************************************
    copyright (c) 2010
Title:              RFM12B recieve simple example based on PIC C
Current version:    v1.1
Function:           Package send Demo
Processor           PIC16F73 DIP-28
Clock:              10MHz Crystal
Operate frequency:  434MHz
Data rate:          4.8kbps
Package size:       23byte
Author:             Simon.Yang
Company:            Hope microelectronic Co.,Ltd.
Contact:            +86-0755-82973805
E-MAIL:             faerf@hoperf.com
Date:               2010-06-28
*************************************************************/
#include "pic.h"

typedef unsigned char uchar;
typedef unsigned int  uint;

#define  SDI            RB7
#define  SDO            RB6
#define  SCK            RB5
#define  nSEL           RB4
#define  nIRQ           RB3
#define  LED            RA0
#define  LED_OUT()      TRISA0=0
#define  nIRQ_IN()      TRISB3=1
#define  SDI_OUT()      TRISB7=0
#define  SDO_IN()       TRISB6=1
```

```
#define  SCK_OUT()     TRISB5=0
#define  nSEL_OUT()    TRISB4=0

void Init_RF12(void);
void Write0( void );
void Write1( void );
void WriteCMD( uint CMD );
uchar RF12_RDFIFO(void);
void Delayus( uint us );

__CONFIG(0x3FF2);
bank1 uchar RF_RXBUF[19];

void Init_RF12(void)
{
  LED_OUT();
  nSEL_OUT();
  SDI_OUT();
  SDO_IN();
  SCK_OUT();
  nIRQ_IN();
  nSEL=1;
  SDI=1;
  SCK=0;
  SDO=0;
  LED=0;
  WriteCMD(0x80D8);//enable register,433MHz,12.5pF
  WriteCMD(0x82D8);//enable receive,!PA
  WriteCMD(0xA640); //Frequency 434MHz
  WriteCMD(0xC647); //Data Rate 4.8KHz
  WriteCMD(0x94A0);// VDI,FAST,BW:134kHz,LNA:0dBm,DRSSI:-103dBm
  WriteCMD(0xC2AC); //Data Filter,DQD
  WriteCMD(0xCA80);
  WriteCMD(0xCED4);
  WriteCMD(0xCA83);//FIFO8,SYNC,2DD4
  WriteCMD(0xC49B);// enable AFC
  WriteCMD(0x9850);//!mp,90kHz,MAX OUT
  WriteCMD(0xCC77);
  WriteCMD(0xE000);//NOT USE
  WriteCMD(0xC800);//NOT USE
  WriteCMD(0xC000);//CLK 1.0MHz,LBD 2.2V
}
```

```
void main()
{
  uchar i=0,j=0;
  uint  CheckSum;

  Init_RF12();

  while(1)
   {
     while(!nIRQ)
     {
      RF_RXBUF[i++]=RF12_RDFIFO();
      if(i==17)
       {
        i=0;
        WriteCMD(0xCA80);
        WriteCMD(0xCA83);        //reset FIFO and read to receive next Byte
        CheckSum=0;
        for(j=0;j<16;j++)
         CheckSum+=RF_RXBUF[j];  //add 0x30-----0x3F
        CheckSum&=0x0FF;
        if(CheckSum==RF_RXBUF[16])
         {
           LED=1;
         }
        Delayus(1);
        LED=0;
        }
       }
   }
}

void Write0( void )
{
  SDI=0;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
```

```
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void Write1( void )
{
  SDI=1;
  SCK=0;
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  NOP();
  SCK=1;
  NOP();
}

void WriteCMD( uint CMD )
{
  uchar n=16;
  SCK=0;
  nSEL=0;
  while(n--)
   {
```

```
    if(CMD&0x8000)
     Write1();
    else
     Write0();
    CMD=CMD<<1;
   }
  SCK=0;
  nSEL=1;
}



uchar RF12_RDFIFO(void)
{
  uchar i,Result;
  SCK=0;
  SDI=0;
  nSEL=0;
  for(i=0;i<16;i++)
  {                    //skip status bits
    SCK=1;
    NOP();
    NOP();
    SCK=0;
    NOP();
    NOP();
  }
  Result=0;
  for(i=0;i<8;i++)
   {                   //read fifo data byte
    Result=Result<<1;
    if(SDO)
    {
      Result|=1;
    }
    SCK=1;
    NOP();
    NOP();
    SCK=0;
    NOP();
    NOP();
   }
  nSEL=1;
  return(Result);
```

```
}
void Delayus( uint us )
{
  uint i;
  while( us-- )
    {
       i=1000;
       while( i-- )
         {
           NOP();
         }
    }
}
```

HOPE MICROELECTRONICS CO.,LTD
Add:4/F, Block B3, East Industrial Area, Huaqiaocheng, Shenzhen, Guangdong, China
Tel: 86-755-82973805
Fax: 86-755-82973550
Email: sales@hoperf.com
trade@hoperf.com
Website: http://www.hoperf.com
http://www.hoperf.cn
http://hoperf.en.alibaba.com