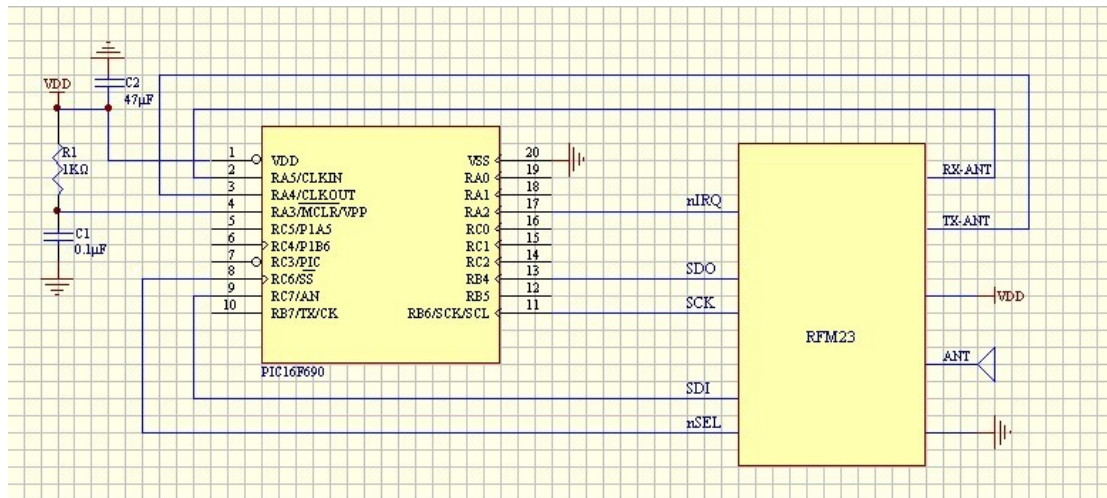


Title: RFM23 transmit demo program



/*

Title: RFM23 transmit demo program

Current version: v1.0

Function: Package send Demo

Processor PIC16F690 DIP-20

Clock: internal 8M.

Operate frequency: 434MHz

Data rate: 4.8kbps

modulation: FSK

deviation: 45K

frame mode: PH + FIFO

payload 0x30, 0x31...0x3f, 0x78(chksum)

Company: HopeRF microelectroni

Contact: +86-0755-82973806

Date: 2000-05-31

*/

```
#include<pic.h>
```

```
#include<math.h>
```

```
const          unsigned          char          tx_buf[17]          =
{0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3a,0x3b,0x3c,0x3d,0x3e,0x3f,0x78};
// ch8  is 140
```

```
unsigned char temp0;
```

```
unsigned char send_command(void);
```

```
void to_tx_mode(void);
void to_ready_mode(void);
```

```
void send_8bit_command(unsigned char i);
void send_read_address(unsigned char i);
void send_write_address(unsigned char i);
void spi_write(unsigned char i, unsigned char l);
```

```
void rf23_init_parameter(void);
void delay_50ms(void);
```

```
void delay_5ms(void);
void delay_1ms(void);
```

```
void port_init(void);
void power_on_delay(void);
```

```
unsigned char spi_read(unsigned char i);
```

```
void Write0( void );
void Write1( void );
```

```
void Write8bitcommand(unsigned char command);
```

```
void main()
{
    OSCCON = 0X70;

    WDTCON = 0X00;

    power_on_delay();
    port_init();

    INTCON = 0x00;

    rf23_init_parameter();

    to_tx_mode();
    while(1);
}
```

```
void Write0( void )
```

```

{
    RB6=0;
    NOP();

    RB4=0;
    NOP();

    RB6=1;
    NOP();
}

```

```

void Write1( void )

```

```

{
    RB6=0;
    NOP();

    RB4=1;
    NOP();

    RB6=1;
    NOP();
}

```

```

void Write8bitcommand(unsigned char command)

```

```

{
    unsigned char n=8;
    RC6 = 1;
    RB6=0;
    RC6=0;
    while(n--)
    {
        if(command&0x80)
            Write1();
        else
            Write0();
        command = command << 1;
    }
}

```

```

void delay_50ms(void)

```

```

{
    unsigned char j;

```

```
    for(j = 0; j<10; j++)
    {
        delay_5ms();
    }
}
```

```
void delay_5ms(void)
{
    int i;
    for(i = 0; i<650; i++)
    {
        ;
    }
}
```

```
void delay_1ms(void)
{
    unsigned char i;
    for(i = 0; i<130; i++)
    {
        ;
    }
}
```

```
void port_init(void)
{
    ANSEL = 0;
    ANSELH = 0;
    WPUA = 0;
    IOCA = 0;
    TRISA = 0x0f;
    TRISB = 0x80;
    WPUB = 0x00;
    TRISC = 0b10101111;
    RA4 = RA5 = 0;
}
```

```
void power_on_delay(void)
{
    unsigned int i;
    for(i = 0; i<1000; i++)
    {
        delay_1ms();
    }
}
```

```
unsigned char spi_read(unsigned char i)
```

```
{  
    unsigned char result;  
    send_read_address(i);  
    result = send_command();  
    RC6 = 1;  
    return(result);  
}
```

```
void rf23_init_parameter(void)
```

```
{  
  
    spi_write(0x06, 0x00);  
  
    spi_write(0x07, 01);  
  
    spi_write(0x09, 0x7f);  
  
    spi_write(0x0a, 0x05);  
    spi_write(0x0b, 0xf4);  
    spi_write(0x0c, 0xef);  
    spi_write(0x0d, 0x00);  
    spi_write(0x0e, 0x00);  
  
    spi_write(0x0f, 0x70);  
  
    spi_write(0x10, 0x00);  
    spi_write(0x12, 0x00);  
    spi_write(0x13, 0x00);  
  
    spi_write(0x70, 0x20);  
  
    spi_write(0x1c, 0x04);  
    spi_write(0x1d, 0x40);  
    spi_write(0x1e, 0x08);  
  
    spi_write(0x20, 0x41);  
    spi_write(0x21, 0x60);  
    spi_write(0x22, 0x27);  
    spi_write(0x23, 0x52);  
    spi_write(0x24, 0x00);
```

spi_write(0x25, 0x06);

spi_write(0x6e, 0x27);

spi_write(0x6f, 0x52);

spi_write(0x30, 0x8c);

spi_write(0x32, 0xff);

spi_write(0x33, 0x42);

spi_write(0x34, 64);

spi_write(0x35, 0x20);

spi_write(0x36, 0x2d);

spi_write(0x37, 0xd4);

spi_write(0x38, 0x00);

spi_write(0x39, 0x00);

spi_write(0x3a, 'h');

spi_write(0x3b, 'o');

spi_write(0x3c, 'p');

spi_write(0x3d, 'e');

spi_write(0x3e, 17);

spi_write(0x3f, 'h');

spi_write(0x40, 'o');

spi_write(0x41, 'p');

spi_write(0x42, 'e');

spi_write(0x43, 0xff);

spi_write(0x44, 0xff);

spi_write(0x45, 0xff);

spi_write(0x46, 0xff);

spi_write(0x6a, 0x0b);

spi_write(0x6d, 0x03);

spi_write(0x79, 0x0);

spi_write(0x7a, 0x0);

spi_write(0x71, 0x22);

spi_write(0x72, 0x48);

spi_write(0x73, 0x0);

```

spi_write(0x74, 0x0);

spi_write(0x75, 0x53);
spi_write(0x76, 0x64);
spi_write(0x77, 0x00);

}

void send_write_address(unsigned char i)
{
    i |= 0x80;
    Write8bitcommand(i);
}

void spi_write(unsigned char m, unsigned char n)
{
    address |= 0x80;
    Write8bitcommand(m);
    send_8bit_command(n);
}

void send_read_address(unsigned char i)
{
    i &= 0x7f;
    Write8bitcommand(i);
}

void send_8bit_command(unsigned char i)
{
    unsigned char n = 8;
    RB6=0;
    while(n--)
    {
        if(i&0x80)
            Write1();
        else
            Write0();
        i = i << 1;
    }
    RB6=0;
}

```

```

}

unsigned char send_command(void)
{
    unsigned char Result, i;

    RB6=0;
    Result=0;
    for(i=0;i<8;i++)
    {
        Result=Result<<1;
        RB6=1;
        NOP();
        if(RC7)
        {
            Result|=1;
        }

        RB6=0;
        NOP();
    }
    return(Result);
}

```

```

void to_tx_mode(void)
{
    unsigned char i;

    Flag.is_txing = 1;

    to_ready_mode();
    RA5 = 0;
    RA4 = 1;
    delay_50ms();

    spi_write(0x08, 0x03);
    spi_write(0x08, 0x00);

    spi_write(0x34, 64);
    spi_write(0x3e, 17);
    for (i = 0; i<17; i++)

```



```
{
    spi_write(0x7f, tx_buf[i]);
}
spi_write(0x05, 04);

spi_write(0x07, 09);

while(RA2);

to_ready_mode();

}

void to_ready_mode(void)
{

    spi_write(0x07, 01);

}
```