

```

/*****/
/* Program      : Tx / Rx frequency 2.4 GHz      */
/* MPU          : AT89C4051                      */
/* Compiler     : Keil V7.50                    */
/* Start Date   : 25/05/2007                    */
/* Programmer    : Jongruk Samart                */
/*****/

```

```

#include <reg51.h>
#include <stdio.h>
#include <intrins.h>

```

```

/*****/
/* ประกาศฟังก์ชัน                                */
/*****/

```

```

void init_sys(void);
void CONFIG_SEND (void);
void CONFIG_RECV (void);
void WR_TRW (void);
void RD_TRW (void);
void Send_Data_to_TRW(unsigned char buff_send);
unsigned char read_Data_from_TRW(void);
void Serial_out_byte (unsigned char msg_out);
void delay_10usec (void);
void delay_200usec(void);
void delay_msec (int x);

```

```

/*****/
/* กำหนด I/O Port                                */
/*****/

```

```

sbit TRW_data1 = P1^0;
sbit TRW_dr1   = P1^1;

```

```

sbit TRW_data2  = P1^2;
sbit TRW_dr2    = P1^3;
sbit TRW_ce     = P1^4;
sbit TRW_clk2   = P1^5;
sbit TRW_cs     = P1^6;
sbit TRW_clk1   = P1^7;
sbit LED_TX     = P3^7;
sbit LED_RX     = P3^5;

```

```

/*****
/* กำหนดโค้ดสำหรับตั้งค่าการทำงานของมอดูล TRW 2.4G หรือ TRF 2.4G */
/* ที่ภายในมีชิปเบอร์ NRF2401 ทำหน้าที่ควบคุมการทำงาน */
/* (ดูรายละเอียดจาก datasheet นะครับ) */
*****/

```

```

#define on 0
#define off 1

```

/* การกำหนดในส่วนนี้ต้องดูจาก datasheet เพราะฉะนั้นเริ่มแรกกำหนดตามผมเลยครับ */

```

code unsigned char reserved_test[]    = {0x8E,0x08,0x1C};
code unsigned char data2_w            = 0x08;
code unsigned char data1_w            = 0x08;
code unsigned char addr2[]            = {0x00,0x00,0x00,0x00,0x22};
code unsigned char addr1[]            = {0x00,0x00,0x00,0x00,0x00};
code unsigned char addr_w_crc         = 0x0A3;
code unsigned char rf_pgm              = 0x4F;

```

```

#define ID_TX  0x00  // กำหนดหมายเลขของตัวส่ง โดยกำหนดได้จาก 0 – 255 หมายเลข
#define ID_RX  0x01  // กำหนดหมายเลขของตัวส่ง โดยกำหนดได้จาก 0 – 255 หมายเลข
#define Fq_CH  0      // กำหนดหมายเลขช่องความถี่ที่จะส่ง โดยกำหนดได้จาก 0 – 124 ช่อง

```

/* การกำหนดหมายเลขการส่งการรับและช่องความถี่ของคู่รับส่งต้องสัมพันธ์กันนะครับ สมมุติว่า เครื่อง A ส่งที่ความถี่ช่องที่ 1 เครื่อง B ก็ต้องช่อง 1 ด้วยการกำหนดช่องเพื่อไม่ให้ชนกับช่องของ wireless lan ต่อมา

มาตุการกำหนดหมายเลขครับ ถ้า A ส่งหมายเลข 0 รับหมายเลข 1 เครื่อง B จะต้องส่งหมายเลข 1 และรับเลข 0
นะครับ ที่แตกต่างจากของบริษัท ETT ตรงที่ผมไม่ได้ใช้ EEPROM เก็บค่ากำหนดเหล่านี้ ดังนั้นการจะ
เปลี่ยนแปลงอะไรก็ต้องคอมไพล์และโปรแกรมใหม่ทุกครั้งนะครับ */

```
unsigned char buff_out,buff_in,F_TI,cnt_rx;  
#define max_buff 100  
unsigned char COM_2TRW[max_buff];
```

/* ตั้งแต่ส่วนนี้เป็นต้นไปเขียนตามผมทั้งหมดเลยครับ */

```
/* **** */  
/* Subroutine for serial interrupt */  
/* **** */
```

```
void Serial_in (void) interrupt 4  
{  
if (RI==1) /* receiv data */  
{  
if((cnt_rx+1)>=max_buff) cnt_rx=max_buff-1;  
COM_2TRW[cnt_rx]=SBUF;  
cnt_rx++;  
RI=0;  
} else if (TI==1) /* after send data complete */  
{  
TI=0;  
F_TI=0;  
}  
}
```

```

void Serial_out_byte (unsigned char TRW_2COM)
{
    SBUF=TRW_2COM;
    while(F_TI==1); /* wait until F_TI==0 */
    F_TI=1;
}

/*****
/* Subrutine for Delay time  x-tal= 18.432 MHZ          */
/* nop = 0.65 us @ 1 machine cycle                    */
*****/
void delay_10usec (void)
{
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
}

void delay_200usec(void)
{
    int i;
    for(i=0;i<20;i++)
        delay_10usec();
}

```

```

void delay_msec (int x)
{
    int i,j;
    for(i=0;i<=x;i++)
        for(j=0;j<1540; j++) _nop_();
}

```

```

/*****
/* Subrutine for write data 8 bit to TRW 2.4G */
*****/

```

```

void WR_TRW (void)
{
    int i;
    for (i=0;i<8;i++)
    {
        TRW_clk1 = 0;
        if((buff_out&0x80)==0x80) TRW_data1 =1;
        else TRW_data1 = 0;
        _nop_();
        TRW_clk1 = 1;
        _nop_();
        buff_out<<=1;
    }
}

```

```

/*****
/* Subrutine read data form TRW 2.4G */
*****/

```

```

void RD_TRW (void)
{
    int j;
    buff_in=0;
    for (j=0;j<8;j++)
    {

```

```

    TRW_clk1 =0;
    _nop_();
    buff_in<<=1;
    TRW_clk1 =1;
    _nop_();
    if(TRW_data1==1) buff_in|=0x01;
    else buff_in|=0x00;
}
}

```

```

/*****
/* Subrutine CONFIG_SEND to TRW 2.4G      */
/*****

```

```

void CONFIG_SEND (void)

```

```

{
    unsigned char F_TX_CH;
    TRW_clk1=0;
    TRW_ce=0;
    TRW_cs=1;      // mode config.
    delay_200usec(); // wakeup from stand by must be delay 202 us

```

```

    F_TX_CH = Fq_CH;
    F_TX_CH<<=1;
    F_TX_CH&=0xFE;    // xxxx xxx0 (xxxx xxx)=frequency ch 0 - 124 [0]
    =TX

```

```

    buff_out=reserved_test[0];
    WR_TRW();
    buff_out=reserved_test[1];
    WR_TRW();
    buff_out=reserved_test[2];
    WR_TRW();
    buff_out=data2_w;
    WR_TRW();
    buff_out=data1_w;
    WR_TRW();

```

```

buff_out=addr2[0];
WR_TRW();
buff_out=addr2[1];
WR_TRW();
buff_out=addr2[2];
WR_TRW();
buff_out=addr2[3];
WR_TRW();
buff_out=addr2[4];
WR_TRW();
buff_out=addr1[0];
WR_TRW();
buff_out=addr1[1];
WR_TRW();
buff_out=addr1[2];
WR_TRW();
buff_out=addr1[3];
WR_TRW();
buff_out=ID_TX;    // ID for send whic reciever must be same number
for reciev
WR_TRW();
buff_out=addr_w_crc;
WR_TRW();
buff_out=rf_pgm;
WR_TRW();
buff_out=F_TX_CH;
WR_TRW();
_nop_();
TRW_ce=1;
TRW_cs=0;    // active by mode
TRW_clk1=0;
}

```

```

/*****
/* Subroutine CONFIG_RECV to TRW 2.4G      */
*****/
void CONFIG_RECV (void)
{
    unsigned char F_RX_CH;
    TRW_clk1=0;
    TRW_ce=0;
    TRW_cs=1;      // mode config.
    delay_200usec(); // wakeup from stand by must be delay 202 us

    F_RX_CH = Fq_CH;
    F_RX_CH<=1;
    F_RX_CH|=0x01;    // xxxx xxx0 (xxxx xxx)=frequency ch 0 - 124 [1]
    =RX

    buff_out=reserved_test[0];
    WR_TRW();
    buff_out=reserved_test[1];
    WR_TRW();
    buff_out=reserved_test[2];
    WR_TRW();
    buff_out=data2_w;
    WR_TRW();
    buff_out=data1_w;
    WR_TRW();
    buff_out=addr2[0];
    WR_TRW();
    buff_out=addr2[1];
    WR_TRW();
    buff_out=addr2[2];
    WR_TRW();
    buff_out=addr2[3];
    WR_TRW();
    buff_out=addr2[4];
    WR_TRW();
    buff_out=addr1[0];

```



```

WR_TRW();
buff_out=addr1[1];
WR_TRW();
buff_out=addr1[2];
WR_TRW();
buff_out=addr1[3];
WR_TRW();
buff_out=ID_RX;    // ID for reciev whic sender must be same number
for send come in
WR_TRW();
buff_out=addr_w_crc;
WR_TRW();
buff_out=rf_pgm;
WR_TRW();
buff_out=F_RX_CH;
WR_TRW();
_nop_();
TRW_ce=1;
TRW_cs=0;    // active mode
TRW_clk1=0;
}

```

```

/*****/
/* Subrutine SEND DATA to TRW 24G */
/*****/

```

```

void Send_Data_to_TRW(unsigned char buff_send)
{
    TRW_ce=1;
    TRW_cs=0;    // active mode
    _nop_();
    buff_out=addr1[0];
    WR_TRW();
    buff_out=addr1[1];
    WR_TRW();
    buff_out=addr1[2];
    WR_TRW();

```

```

buff_out=addr1[3];
WR_TRW();
buff_out=ID_TX;    // send Address
WR_TRW();
buff_out=buff_send; // send Data
WR_TRW();
_nop_();
TRW_ce=0;
TRW_cs=0;          //Set to standby mode to Send the data.
TRW_clk1=0;
}

```

```

/*****
/* Subrutine READ DATA from TRW 24G */
*****/

```

```

unsigned char read_Data_from_TRW(void)
{
    unsigned char tmp;
    tmp=0xff;
    TRW_data1=1; // set pin for input
    TRW_dr1=1;   // set pin for input
    TRW_ce=1;
    TRW_cs=0;    // active mode
    _nop_();
    if(TRW_dr1==1)
    {
        RD_TRW();
        tmp=buff_in;
    }

    TRW_cs=0;
    TRW_ce=1;
    TRW_cs=0;    // active mode
    TRW_clk1=0;
    _nop_();
}

```

```

return(tmp);
}

/*****
/* Subrutine Initialize systems */
*****/

void init_sys(void)
{

    TMOD = 0x21;
    TH1 = 0x0FB;
    TL1 = 0x0FB; // 9600 bps
    SCON = 0x52;
    EA = 1;
    ES = 1;
    TR1 = 1;
    F_TI=1;
    TI=1;
    TRW_ce=0;
    TRW_cs=0;
    TRW_clk1=0; // set to stanby mode
    TRW_data1=1; // set pin to input port
    TRW_dr1=1; // set pin to input port
    delay_msec(15);
    CONFIG_RECV();
}

```

```

/*****/
/*      Subroutine Main      */
/*****/
void main (void)
{
    int ptr1;
    init_sys();
    while(1)
    {
        if((cnt_rx!=0)&&(TRW_dr1!=1)) // When message input from RS232 then
send to TRW 2.4G
        {
            LED_TX=on;
            LED_RX=off;
            CONFIG_SEND();
            for(ptr1=0;ptr1<cnt_rx; ptr1++)
            {
                Send_Data_to_TRW(COM_2TRW[ptr1]);
                delay_200usec();
                delay_200usec();
                delay_200usec();
            }
            cnt_rx=0;
            CONFIG_RECV();
            LED_TX=off;
            LED_RX=off;
        }
        else
        {
            if(TRW_dr1==1)
            {
                LED_RX=on;
                LED_TX=off;
                Serial_out_byte(read_Data_from_TRW());
                LED_RX=off;
                LED_TX=off;
            }
        }
    }
}

```