#include <iocc2530.h>

#include <stdio.h>

//#define RX

#define LED1 P1\_0

#define LED2 P1\_1

static char buf[128];

static int len=0;

unsigned char i;

void Delay(unsigned int n)

{

unsigned int i,tt;

for(i=0;i<5;i++)

for(tt=0;tt<n;tt++)

;

}

void rf\_init()

{

//硬件CRC以及AUTO\_ACK使能

FRMCTRL0 |= (0x20 | 0x40);

//设置TX抗混叠过滤器以获得合适的带宽

TXFILTCFG = 0x09;

//调整AGC目标值

AGCCTRL1 = 0x15;

//获得最佳的EVM

FSCAL1 = 0x00;

// RXPKTDONE 中断位使能

RFIRQM0 |= (1<<6);

// RF 中断使能

IEN2 |= (1<<0);

//开中断

EA = 1;

//信道选择，选择11信道

FREQCTRL = 0x0b;

//目标地址过滤期间使用的短地址

SHORT\_ADDR0 = 0x05;

SHORT\_ADDR1 = 0x00;

//目标地址过滤期间使用的PANID

PAN\_ID0 = 0x22;

PAN\_ID1 = 0x00;

//清除RXFIFO缓冲区并复位解调器

RFST = 0xed;

//为RX使能并校准频率合成器

RFST = 0xe3;

//禁止帧过滤

FRMFILT0 &= ~(1<<0);

}

void tx()

{

unsigned char i;

unsigned char tx[13]={'a','b','c','d','e','f','g','h','i','j','k','\r','\n'};

//为RX使能并校准频率合成器

RFST = 0xe3;

// TX\_ACTIVE | SFD

while (FSMSTAT1 & ((1<<1) | (1<<5)));

//禁止RXPKTDONE中断

RFIRQM0 &= ~(1<<6);

//禁止RF中断

IEN2 &= ~(1<<0);

// 清除TXFIFO缓存

RFST = 0xee;

// 清除 TXDONE 中断

RFIRQF1 = ~(1<<1);

// 发送的第一个字节是传输的帧长度

RFD = 13+2;//tx中包含的13个字符+2;填充缓冲区填充过程需要增加2字节，CRC校验自动填充

//将mac的内容写到RFD中

for(i=0;i<13;i++)

{

RFD = tx[i];

}

// 打开RX中断

RFIRQM0 |= (1<<6);

//打开RF中断

IEN2 |= (1<<0);

//校准后使能TX

RFST = 0xe9;

//等待传输结束

while (!(RFIRQF1 & (1<<1)));

//清除 TXDONE状态

RFIRQF1 = ~(1<<1);

//LED1灯状态改变

LED1=~LED1;

//延时

Delay(20000);

}

void initLed(void)

{

//P1为普通 I/O 口

P1SEL &= ~(1<<0);

//P1.0 P1.1设置为输出

P1DIR |= 0x03;

//关闭LED1

LED1=1;

//关闭LED2

LED2=1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

串口初始化函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void initUARTSEND(void)

{

PERCFG = 0x00; //位置 1 P0 口

P0SEL = 0x3c; //P0\_2,P0\_3,P0\_4,P0\_5用作串口

P2DIR &= ~0xc0; //P0 优先作为UART0

U0CSR |= 0x80; //选择 UART 模式

U0GCR |= 11; // 查表获得 U0GCR 和 U0BAUD

U0BAUD = 216; // 115200

UTX0IF = 0; //UART0 TX 中断标志初始置位 0

}

void initClock(void)

{

/\*相关寄存器可以在 cc2530数据手册中查找\*/

CLKCONCMD &= ~0x40; //设置系统时钟源为 32MHZ晶振

while(CLKCONSTA & 0x40); //等待晶振稳定

CLKCONCMD &= ~0x47; //设置系统主时钟频率为 32MHZ X0XXX000

SLEEPCMD |= 0x04;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

串口发送字符串函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void UartTX\_Send\_String(char \*Data,int len)

{

int j;

for(j=0; j<len; j++)

{

U0DBUF = \*Data++;

while(UTX0IF == 0);

UTX0IF = 0;

}

}

void main(void)

{

//关闭总中断

EA = 0;

initLed();

initClock();

initUARTSEND();

//初始化RF

rf\_init();

//中断使能

EA = 1;

//发送或等待接收中断

while(1)

{

//宏定义RX

#ifndef RX

//如果没有定义RX，开始发送

tx();

//延时

Delay(20000);

Delay(20000);

//如果定义RX，等待接收中断

# else

#endif

}

}

//接收中断处理

#pragma vector=RF\_VECTOR

\_\_interrupt void rf\_isr(void)

{

unsigned char i;

int rssi=0;

char crc\_ok=0;

EA=0;

//关中断

IEN2 &= ~0X01;

//接收帧结束

if (RFIRQF0 & (1<<6))

{

//接收帧长度

len = RFD ;

//printf("\nlen = %d\n\*\*\*\*\*\*\*\*\*\*\*\n",len);

len &= 0x7f;

//将接收的数据写入buf中

for (i = 0; i < len - 2; i++)

{

buf[i] = RFD;

Delay(200);

//向串口发送接收到的数据

//UartTX\_Send\_String(&buf[i],1);

}

rssi = RFD - 73; //读取RSSI结果

crc\_ok = RFD;

printf("[%d],crcResult=%d\n",rssi,(crc\_ok&0x80));////crc\_ok&0x80读取CRC校验结果 BIT7

//向串口发送接收到的数据

UartTX\_Send\_String(buf,len-2);

RFST = 0xED;

// 清RF中断

S1CON = 0;

//清 RXPKTDONE中断

RFIRQF0 &= ~(1<<6);

//LED1等状态改变

LED1 = ~LED1;

}

IEN2 |= (1<<0);

EA=1;

}