//发送温度数据

#include <ioCC2530.h>

#define uint unsigned int

//定义控制LED灯的端口

#define LED1 P1\_0 //定义LED1为P10口控制

#define LED2 P1\_1 //定义LED2为P11口控制

float avgtemp;

char tempvalue[10];

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

//延时函数

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

void Delay(uint n)

{

uint i,t;

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

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

}

//初始化时钟

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;

}

//初始化Led

void InitLed(void)

{

P1DIR |= 0x03; //P1\_0、P1\_1定义为输出

LED1 = 1; //LED1灯熄灭

LED2 = 1; //LED2灯熄灭

}

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

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等于1（发生中断），则一直循环

;

UTX0IF = 0;

}

}

void InitUart()

{

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

PERCFG = 0x00; //位置1 P0口

P0SEL = 0x3c; //P0用作串口

P2DIR &= ~0XC0;//P0用作串口

U0CSR |= 0x80; //UART方式

U0GCR |= 11; //baud\_e = 11;

U0BAUD |= 216; //波特率设为115200

UTX0IF = 1;

IEN0 |= 0x80;//开总中断

IEN2 |= 0x04;//发送中断使能

IEN0 |= 0x04;//开接收中断使能

URX0IE = 1;

U0CSR |= 0X40; //允许接收

}

//测量片内温度传感器温度

float getTemperature(void)

{

uint value;

ADCCON3 = 0x3e; //选择1.25V为参考电压；14位分辨率；对片内温度传感器采样

ADCCON1 |= 0x30;//ADC启动方式选择为ADCCON1.ST=1事件

ADCCON1 |= 0x40;//ADC启动转换

while(!(ADCCON1 & 0x80))//等待AD转换完成

;

value = ADCL >>4;/\*ADCL寄存器低4位无效，ADC最高只能达到12位有效位。网络上很多代码这里都是右移两位，那是不对的\*/

value |= (((uint)ADCH)<<4);//连接ADCH和ADCL，并赋值给value

return (value - 1367.5)/4.5;

//return ((value) >> 4) - 315; //根据AD值，计算出实际的温度

//return value\*0.06229-311.43;

//return (value-1367.5)/4.5-4; //上面三个方法不知道用那个 高人指点一下

//return value\*0.06229-303.3-4;/\*温度的计算公式为：温度=（（测量电压-某一电压）/温度系数）-温度的误差值\*/

}

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(char \*tx,int len)

{

unsigned char i;

//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 = len+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<len;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

//延时

Delay(20000);

}

void main(void)

{

initClock();

InitUart();

InitLed();

ATEST = 0x01;//开启温度传感器

TR0 = 0X01; //将温度传感器与ADC连接起来

while(1)

{

avgtemp = getTemperature();//取得温度数

tempvalue[0] = (unsigned char)(avgtemp)/10 + 48; //十位

tempvalue[1] = (unsigned char)(avgtemp)%10 + 48; //个位

UartTX\_Send\_String(tempvalue,2);

tx(tempvalue,2);

Delay(50000); //延时

if((tempvalue[0]\*10+tempvalue[1])>24)

{

LED1=0; //标志发送状态

Delay(50000);

Delay(50000);

}

else

LED1=1;

}

}