#include 《reg52.h》 //包含头文件

　　#define uint unsigned int

　　#define uchar unsigned char

　　#define RELOAD\_COUNT 0xFA //宏定义波特率发生器的载入值

　　//define RELOAD\_COUNT (256-(((11520000/16)/12)/9600)) 也可以或波特率9600 //256-晶振频率/波特率x16=BRT

　　/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LED灯对应P0口的1个端口\*\*\*\*\*\*\*\*\*\*\*\*\*/

　　sbit LED0=P3^2;

　　/\*\*\*\*\*\*\*\*\*\*\*\*波特率发生器相关功能寄存器的定义\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

　　sfr AUXR=0x8E;

　　sfr BRT=0x9C;

　　sfr AUXR1=0xA2;

　　/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*相关变量\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

　　uchar Receive，i;

　　uint n;

　　uchar Recive\_table［15］; //用于接收wifi模块反馈到MCU上的数据

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

　　名称：延时函数

　　作用：毫秒级延时，微妙级延时函数，为数据收发完成作等待。。.。。.。

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

　　void ms\_delay(uint t)

　　{

　　uint i，j;

　　for(i=t;i》0;i--)

　　for(j=110;j》0;j--);

　　}

　　void us\_delay(uchar t)

　　{

　　while(t--);

　　}

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

　　名称：波特率发生器函数

　　作用：波特率发生器可以是T1定时器实现，也可以是MCU内部独立的波特率发生器，

　　各自不同的载入值计算式，具体根据寄存器相关设置来参考计算，以实现

　　异步串行通讯。(经测试，两种设置方式均可用，可任选一种。)

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

　　void Uart\_Init()//使用定时器1作为波特率发生器(STC89C52、STC89C51、AT89C51或者STC12C560S2等均可)

　　{

　　SCON=0x50; //设置为串行口以方式1工作，8位异步通讯，允许接收中断。

　　//一帧信息为10位，1位起始位，8位数据位(低位在先)，1位停止位。

　　PCON=0x80; //SMOD波特率选择位为1，SMOD=1.

　　TMOD=0x21; //设置定时器1为波特率发生器，工作在模式2，8位自动装载

　　TH1=RELOAD\_COUNT;//波特率9600 ，TH1=256-FOSC/16/12/波特率

　　TL1=TH1;

　　EA=1; //总中断打开

　　ES=0; //关闭串口中断

　　TR1=1; //启动定时器1

　　}

　　/\*

　　void Uart\_Init()//使用独立的波特率发生器(STC12C560S2或带有独立波特率发生器//、//的单片机均可)

　　{

　　SCON=0x50; //设置为串行口以方式1工作，8位异步通讯，允许接收中断。

　　PCON=0x80; //SMOD波特率选择位为1，SMOD=1.

　　BRT=RELOAD\_COUNT;//波特率9600 256-晶振频率/波特率x16=BRT

　　AUXR=0xD5; //T0、T1不12分频，速度是89C51的12倍，启动独立波特率//发生器，每个时钟计数一次

　　//设置串口1的波特率发生器为独立波特率发生器，

　　AUXR1=0x80; //切换到P1口

　　ES=1; //开启串口中断

　　EA=1; //开启总中断

　　}

　　\*/

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

　　名称：串口发送函数

　　功能：MCU向其他与其连接的设备发送数据(此处是无线WIFI模块ESP8266)

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

　　void Send\_Uart(uchar value)

　　{

　　ES=0; //关闭串口中断

　　TI=0; //清发送完毕中断请求标志位

　　SBUF=value; //发送

　　while(TI==0); //等待发送完毕

　　TI=0; //清发送完毕中断请求标志位

　　ES=1; //允许串口中断

　　}

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

　　名称：WIFI模块设置函数

　　作用： 启动模块，以便可以实现无线接入和控制

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

　　void ESP8266\_Set(uchar \*puf) // 数组指针\*puf指向字符串数组

　　{

　　while(\*puf！=‘\0’) //遇到空格跳出循环

　　{

　　Send\_Uart(\*puf); //向WIFI模块发送控制指令。

　　us\_delay(5);

　　puf++;

　　}

　　us\_delay(5);

　　Send\_Uart(‘\r’);//回车

　　us\_delay(5);

　　Send\_Uart(‘\n’); //换行

　　ms\_delay(1000);

　　}

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

　　名称：ESP8266发送数据函数

　　功能：用于与wifi模块相连的终端发送数据

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

　　void ESP8266\_Sent(uchar \*puf) // 数组指针\*puf指向字符串数组

　　{

　　ESP8266\_Set(“AT+CIPSEND=0，4”);

　　while(\*puf！=‘\0’) //遇到空格跳出循环

　　{

　　Send\_Uart(\*puf); //向WIFI模块发送控制指令。

　　us\_delay(5);

　　puf++;

　　}

　　us\_delay(5);

　　Send\_Uart(‘\n’); //换行

　　ms\_delay(10);

　　}

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

　　名称：主函数

　　作用：程序的执行入口

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

　　void main()

　　{

　　LED0=0;//关闭LED灯

　　Uart\_Init();//使用独立的波特率发生器

　　ESP8266\_Set(“AT+CWMODE=2”); //设置路由器模式 1 station模式 2 AP

　　//点 路由器模式 3 station+AP混合模式

　　ESP8266\_Set(“AT+RST”); //重新启动wifi模块

　　ESP8266\_Set(“AT+CWSAP=\”WIFI\”，\”1234567890\”，11，4”);

　　//设置模块SSID:WIFI， PWD：密码 及安全类型加密模式(WPA2-PSK)

　　ESP8266\_Set(“AT+CIPMUX=1”);//开启多连接模式，允许多个各客户端接入

　　ESP8266\_Set(“AT+CIPSERVER=1，5000”); //启动TCP/IP 端口为8080 实现基于网络//控制

　　ES=1; //允许串口中断

　　while(1)

　　{

　　if((Recive\_table［0］==‘+’)&&(Recive\_table［1］==‘I’)&&(Recive\_table［2］==‘P’))//MCU接收到的数据为+IPD时进入判断控制0\1来使小灯亮与灭

　　{

　　if((Recive\_table［3］==‘D’)&&(Recive\_table［6］==‘，’))

　　{

　　if(Recive\_table［9］==‘0’)

　　{

　　LED0=0;//0 灯灭

　　ESP8266\_Sent(“灯灭”);

　　//wifi模块向pc端或手机端 发送”灯灭

　　}

　　else if (Recive\_table［9］==‘1’)

　　{

　　LED0=1; //1 灯亮

　　ESP8266\_Sent(“灯亮”);

　　//wifi模块向pc端或手机端 发送”灯亮”

　　}

　　}

　　}

　　}

　　}

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

　　名称：串行通讯中断

　　作用：发送或接收结束后进入该函数，对相应的标志位软件清0，实现模块对数

　　据正常的收发。

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

　　void Uart\_Interrupt() interrupt 4

　　{

　　static uchari=0;

　　if(RI==1)

　　{

　　RI=0;

　　Receive=SBUF; //MCU接收wifi模块反馈回来的数据

　　Recive\_table=Receive;

　　i++;

　　if((Recive\_table［i-1］==‘\n’))i=0; //遇到换行 重新装值

　　}

　　else TI=0;

　　}