/\*

识别关键词的个数为SUM；

n为数组中对应关键词的序列号，例如数组sRecog中的第一个关键词为“kai deng”则对应的序列号为0；

Voice.Initialise();初始化设置输入方式MIC/MONO。

\*/

#include <TimerOne.h>

#include <avr/wdt.h>

#include <Voice.h>

#include <Syn6288.h>

Syn6288 syn;

uint8 nAsrStatus=0;

#define SUM 38

#define DHT11\_PIN 6 //DHT11传感器插数字口6

#define light\_sensor\_pin A2 //光线传感器插模拟口2

#define light\_threshold 200 //光线传感器阈值

#define relay 8 //继电器插数字口8

#define D3 3

#define state\_led 7 //状态指示灯插数字口7

int val;

int val1=0;

int in1=1;

char sRecog[SUM][30] = {"xiao jian guo","kai deng ","liang yi dian","an yi dian","guan deng","xu yao kai deng ma","zai liang yi dian","zai an yi dian","zui liang "," wo hao ma","hu xi deng mo shi","guan","liang","liang yi","an","an yi","yi dian"};

// 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16小坚果，开灯，关灯，需要开灯吗

uint8\_t text2[]={0xCE,0xD2,0xD4,0xDA};// 我在

uint8\_t text5[]={0xD5,0xFD,0xD4,0xDA,0xD7,0xAA,0xBB,0xBB};//正在转换

uint8\_t text6[]={0xCB,0xA7,0xBC,0xAB,0xC1,0xCB,0xD6,0xF7,0xC8,0xCB};//帅极了，主人

uint8\_t text7[]={0xBA,0xC3,0xB5,0xC4};//好的

uint8\_t text8[]={0xD5,0xFD,0xD4,0xDA,0xBF,0xAA,0xB5,0xC6};//正在开灯

uint8\_t text9[]={0xD5,0xFD,0xD4,0xDA,0xB9,0xD8,0xB5,0xC6};//正在关灯

uint8\_t text10[]={0xB4,0xF8,0xD7,0xDF,0xCE,0xD2,0xB0,0xC9};//请问你在说什么？

uint8\_t text11[]={0xB9,0xE2,0xCF,0xDF,0xB2,0xBB,0xD7,0xE3,0x20,0xA3,0xAC,0xD5,0xFD,0xD4,0xDA,0xBF,0xAA,0xB5,0xC6};//光线不足，正在开灯

uint8\_t text12[]={0xCF,0xD6,0xD4,0xDA,0xB9,0xE2,0xCF,0xDF,0xC3,0xF7,0xC1,0xC1,0xA3,0xAC,0xB2,0xBB,0xD0,0xE8,0xD2,0xAA,0xBF,0xAA,0xB5,0xC6};//光线明亮，不需要开灯

uint8\_t flag; //标志位，收到小坚果口令后将置为1，动作执行完后清零

void finally(uint8\_t n) //n为sRecog数组中对应关键词的序列号

{

switch(n)

{

case 0: syn.play(text2,sizeof(text2),0);flag=1;break; // 我在

case 1: //开灯

if(flag==1)

{

//播放正在开灯

syn.play(text8,sizeof(text8),0);

//变量循环+1

for(val=0;val<10;val++)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(50);

}

flag=0;

}

break;

//亮一点？ 增加亮度

case 2:

if(flag==1)

{

//播放好的

syn.play(text7,sizeof(text7),0);

//变量循环+1

for(val=10;val<50;val++)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(50);

}

flag=0;

}

break;

//暗一点？

case 3:

if(flag==1)

{

//播放好的

syn.play(text7,sizeof(text7),0);

//变量循环+1

for(val=255;val>50;val--)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(10);

}

delay(50);

flag=0;

}

break;

case 4: //关灯

if(flag==1)

{

//播放正在关灯

syn.play(text9,sizeof(text9),0);

analogWrite(D3, 0);

flag=0;

}

break;

case 5: //需要开灯吗？

if(flag==1)

{

//读取光照度模拟量

int shine=analogRead(light\_sensor\_pin);

//小于光照度阀值时

if(shine < light\_threshold)

{

//光线不足，正在开灯

syn.play(text11,sizeof(text11),0);

//变量循环+1

for(val=0;val<20;val++)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(50);

}

}

else

{

//光线明亮，不需要开灯

syn.play(text12,sizeof(text12),0);

analogWrite(D3, 0);

}

flag=0;

}

break;

//再暗一点？

case 7:

if(flag==1)

{

//播放好的

syn.play(text7,sizeof(text7),0);

for(val=50;val>10;val--)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(50);

}

delay(50);

flag=0;

}

break;

//再亮一点？ 增加亮度

case 6:

//最亮

case 8:

if(flag==1)

{

//播放好的

syn.play(text7,sizeof(text7),0);

//变量循环+1

for(val=50;val<255;val++)

{

//PWM输出

analogWrite(D3, val);

//设定延时时间

delay(10);

}

flag=0;

}

break;

case 9: //我好吗?

if(flag==1)

{

//播放好的

syn.play(text6,sizeof(text6),0);

delay(50);

flag=0;

}

break;

//呼吸灯模式

case 10:

if(flag==1)

{

flag=0;

//播放正在转换

syn.play(text5,sizeof(text5),0);

while(!flag){

//变量循环+1

for(;val1>-1;val1=val1+in1)

{

if(val1==255){

in1=-1;

}

if(val1==0){

in1=1;

}

//PWM输出

analogWrite(D3, val1);

//设定延时时间

delay(50);

}

}

delay(50);

flag=0;

}

break;

//非正常命令，请问你在说什么

default:syn.play(text10,sizeof(text10),0);break;

}

}

void ExtInt0Handler ()

{

//芯片送出中断信号

Voice.ProcessInt0();

}

void timerIsr()

{

wdt\_reset();

}

void setup()

{

Serial.begin(9600);

//Initialise mode MIC or MONO,default is MIC

//VoiceRecognitionV1 is VoiceRecognitionV1.0 shield

//VoiceRecognitionV2 is VoiceRecognitionV2.1 module

Voice.Initialise(MIC,VoiceRecognitionV1);

attachInterrupt(0,ExtInt0Handler,LOW);

pinMode(relay,OUTPUT);

digitalWrite(relay,LOW);

pinMode(state\_led,OUTPUT);

digitalWrite(state\_led,LOW);

Timer1.initialize(90000);

// attach the service routine here

Timer1.attachInterrupt(timerIsr);

wdt\_enable(WDTO\_120MS);

}

void loop()

{

static uint8\_t nAsrRes=0;

static unsigned long time = millis();

if(millis()-time>100)

{

switch(nAsrStatus)

{

case LD\_ASR\_RUNING:

case LD\_ASR\_ERROR:break;

case LD\_ASR\_NONE:

{

nAsrStatus=LD\_ASR\_RUNING;

//识别不正确

if (Voice.RunASR(SUM,30,sRecog)==0)

{

nAsrStatus= LD\_ASR\_ERROR;

Serial.println( "ASR\_ERROR");

}

Serial.println( "ASR\_RUNING.....");

digitalWrite(state\_led,HIGH);

break;

}

case LD\_ASR\_FOUNDOK:

{

digitalWrite(state\_led,LOW);

// 一次ASR识别流程结束，去取ASR识别结果

nAsrRes =Voice. LD\_GetResult();

finally(nAsrRes);

nAsrStatus = LD\_ASR\_NONE;

break;

}

case LD\_ASR\_FOUNDZERO:

default:

{

digitalWrite(state\_led,LOW);

nAsrStatus = LD\_ASR\_NONE;

break;

}

}// switch

}//if

}