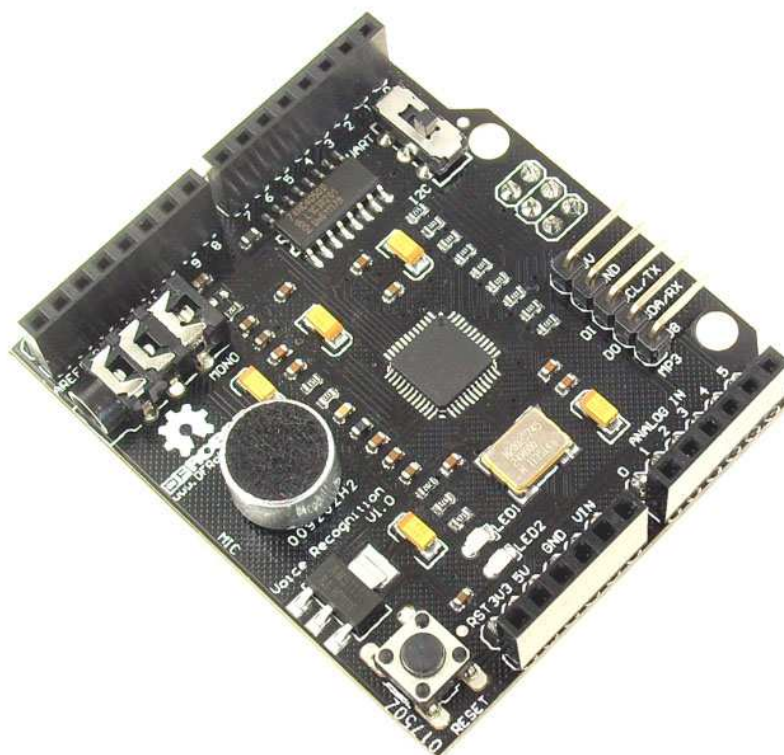




DFRduino Voice Recognition V1.0 Users Manual



Dreamfactory 梦工厂

DFRduino Voice Recognition Users Manual

TEL: (北京公司) 庄先生 010-82355005

(成都公司) 卫先生 15902808530

(上海公司) 桑先生 13774201234

DFRduino Voice Recognition

- A. 注意！在没有认真阅读本说明之前，请勿给模块加电！错误接线将导致模块永久性损坏或烧毁微控制器。
- B. 注意！请认真查看引脚功能说明，正确接线！请勿将电源反接，否则将导致模块永久性损坏。
- C. 注意！请勿使用超出额定电压的电源！保证电源的稳定，如果出现高压脉冲，可能会导致微控制器永久性损坏。
- D. 注意！本产品无防水防潮功能，请在干燥环境下保存或使用！不可将重物堆积在上面。

概 述

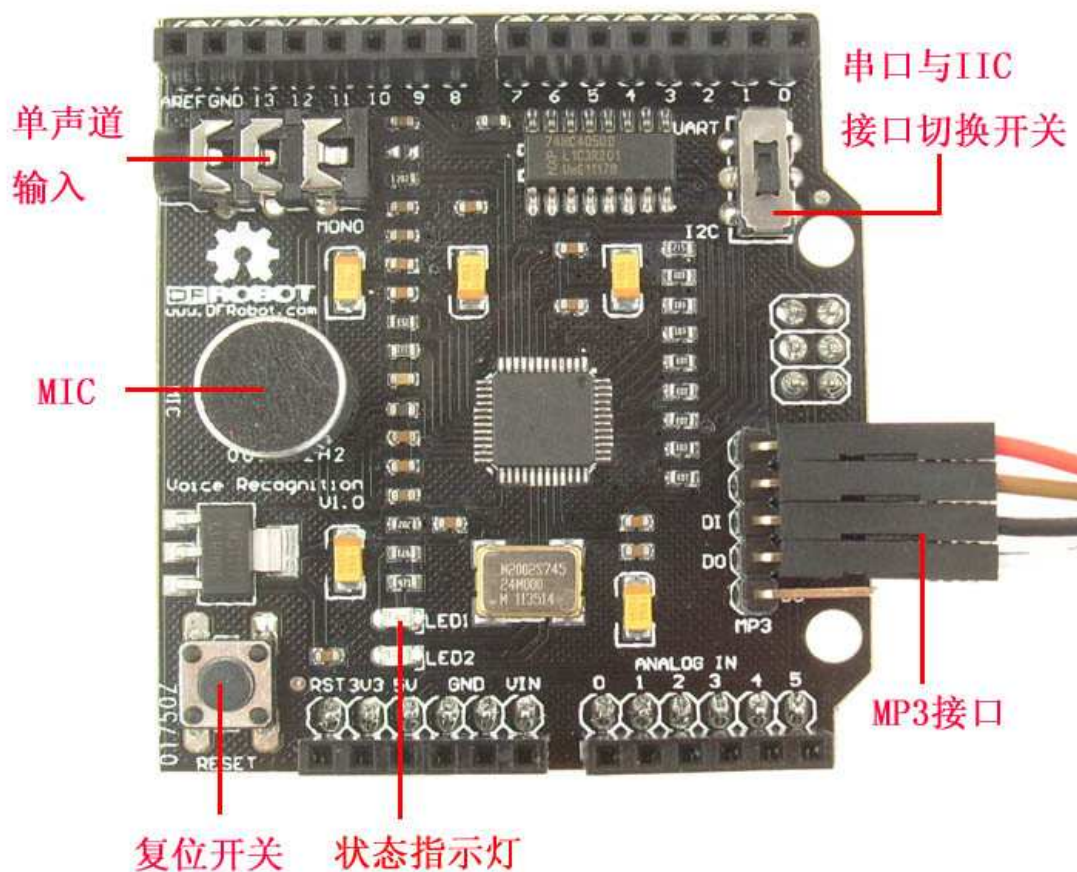
Voice Recognition 语音识别模块是一款只需要在主控 MCU 的程序中设定好要识别的关键词语列表，并动态地把这些关键词语以字符的形式传送到芯片内部，就可以对用户说出的关键词语进行识别，不需要用户事先训练和录音的非特定人语音识别模块。该模块可以设置 50 项候选识别句，每个识别句可以是单字，词组或短句，长度为不超过 10 个汉字或者 79 个字母的拼音串，可由一个系统支持多种场景。

Voice Recognition 语音识别模块采用叠层设计，可以直接插接到 Arduino 控制器上，用户使用 Arduino 便可以快速设计产品原型，例如电磁炉/微波炉/洗衣机/智能家电操作，导航仪，MP3/MP4，数码像框，机顶盒/彩电遥控器，智能玩具/对话玩具，PMP/游戏机，自动售货机，地铁自动售票机，导游机，楼宇电视的广告点播，照明系统的声控等等。

性能描述

1. 工作电压：5V
2. 兼容 Arduino 和 Arduino MEGA 控制板
3. 具有板载 MIC，支持单声道输入
4. 具有 DFRduino Player 模块接口
5. 具有 DFRduino Player 模块 UART/I2C 接口切换开关（MEGA 只能使用 UART 接口）
6. 尺寸：长 54mm× 宽 47mm

Voice Recognition 的使用



Voice Recognition 模块说明

开关按钮说明

标注	说明
RESET	Arduino 复位按钮
UART/I2C	DFRuino Player 模块接口转换（注意：mega 控制板只能使用 UART 接口）

LED 指示灯说明

标注	说明
LED1	模块状态指示，常亮表示模块工作正常
LED2	

接口端说明

标注	说明
MONO	单声道输入接口（通过 Voice.Initialise(MONO) 进行选择）
MIC	麦克风（通过 Voice.Initialise(MIC) 进行选择）
MP3	DFRuino Player 模块接口

占用 Arduino 端口说明

端口	说明
数字口 0	DFRuino Player 模块串口控制口 RXD
数字口 1	DFRuino Player 模块串口控制口 TXD
模拟 4	DFRuino Player 模块 IIC 控制口 SDA
模拟 5	DFRuino Player 模块 IIC 控制口 SCL
数字口 2	Arduino 中断输入口，语音识别模块发出中断信号
数字口 4	语音模块 SPI 通讯接口 SS
数字口 9	语音模块复位
数字口 10	使用语音模块时，该端口需要设置为输出高电平，库文件中已进行此操作
数字口 11	语音模块 SPI 通讯接口 MOSI
数字口 12	语音模块 SPI 通讯接口 MISO
数字口 13	语音模块 SPI 通讯接口 SCK

Voice Recognition 模块语音识别

语音识别，识别的是“语音”。对于非特定人语音识别来说，在描述关键词语时，是用音标标注出要识别的关键词语。

对于 Voice Recognition 模块支持的中文识别来说，就是用拼音来描述出关键词语。

也就是说，只要是拼音可以拼出的发音，都是可以进行识别的。

因此，在某些场合需要识别一些简单的外文或者纯方言发音的时候，可以用拼音标注的方法来实现。

例如，有些场合需要识别一些简单的英文单词，可以用拼音标注：

one → wan

two → tu

three → si rui

例如，有些场合需要识别一些纯方言发音的词汇，也可以用拼音标注：

上海话的“晚”发音是“ya”，那么“晚报”这个词汇，用普通话标注是“wan bao”，如果要标注成上海话发音，就是“ya bao”，这样上海话说的“晚报”也就可以被识别了。

值得注意的是：模块支持的是中文普通话，有些外文或者方言发音无法用拼音描述，所以模块不一定能够完成所有需要的外文或者方言任务。

Voice Recognition 模块识别中的常见问题

- 1) 改变使用环境，或许在某些环境中的噪声或者回声会影响到判断说话结束。
- 2) 说话人自己的音量，如果声音很低，也会导致判断人说话是否结束比较困难。
- 3) 改变命令词语内容，比较好念，开口音响亮等，方便使用者连续清晰念出语音命令。

Voice Recognition 模块语音识别应用场景

1. 在识别精度要求高的场景中，使用“触发识别”模式

在识别精度要求高的场景中，应该采用“触发识别”模式。原因是：

- 1) 用户在每次按热键后，精神处于最集中的状态，此时用户说的语音命令会比较认真，清晰。避免了用户过于随意的发音导致的识别误差。
- 2) 每次按热键后，产品应该给以一个明显的开始信号，比如发出“当”的一声或者其他提示信号，可以给用户一个明确开始的提示，方便用户掌握说语音命令的时间。
- 3) 由于按键触发后，用户就会贴近麦克风并说出语音命令，避免了其他环境声音被录入模块芯片导致的误识别。

另：这种方式还是一种省电的方式，在不识别时，彻底不让芯片工作以省电。

2. 增添“垃圾关键词语”——吸收错误识别

DFRduino 语音识别模块

机器人工厂

TEL:010-82355005

北京龙凡汇众机器人科技有限公司

E_mail: service@dfrobot.com

在设定好要识别的关键词语后，为了进一步降低误识别率，可以再添加一些其他的任意词汇进识别列表，用来吸收错误识别，从而达到降低误识别率的目的。

可以把这些关键词语称之为“垃圾关键词语”。

比如，某个应用场景中，需要识别的关键词语是 4 条，“前进”，“后退”，“开门”，“关门”。在这 4 个关键词语设置进芯片后，可以再另外设置 10~30 个词语进模块芯片，比如“前门”，“后门”，“阿阿阿”，“呜呜”等等。

只有识别结果是 4 个关键词语之内的，才认为识别有效。如果识别结果是“垃圾关键词语”，则说明是其他的声音导致的误识别，产品应该重新开始一次识别过程。

这样，可以非常非常有效地降低误识别率。极大地提高终端用户的主观使用体验。

“垃圾关键词语”的选取，最好可以选择一些字数和关键词语一样的词语，用来吸收可能发生的错误识别。

3. 口令触发模式

在一些应用场合，希望识别精度高，但是又无法要求用户每次都用手按键来“触发识别”。此时，可以采用“口令触发模式”。

产品定义一句短语，作为触发口令。比如，可以定义“芝麻开门”作为触发口令。

产品在等待用户触发时，启动一个“循环识别”模式，把触发口令“芝麻开门”和其他几十个用来吸收错误的词汇设置进芯片。只有当检测到识别出的结果是触发口令时，才认为是终端用户叫了这个口令。此时，给出提示音，

并启动一个“触发识别模式”，并且把相应的识别列表设置进模块芯片，提示用户在提示音后几秒钟内说出要执行的操作。

在等待用户的过程时，如果识别的结果是那些用来吸收错误的词汇，则认为是误识别，或者其他的声音干扰，而不进行任何的处理，直接再次进入“循环识别”模式。

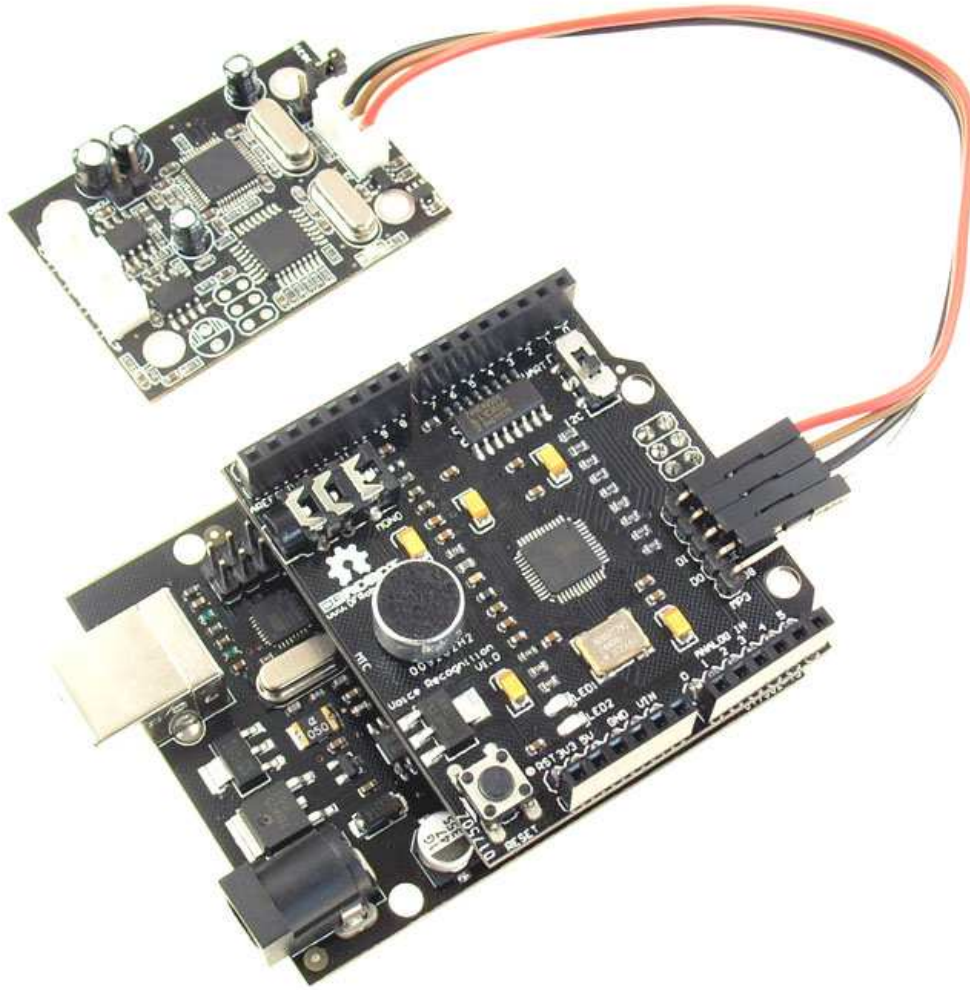
这种口令触发模式，融合了其他两种模式的优点，并且结合提到的“垃圾关键词语”的方法，可以为产品提供更加方便实用的语音操作特性。

4. 应用场景

电磁炉/微波炉/洗衣机/智能家电操作，导航仪，MP3/MP4，数码像框，机顶盒/彩电遥控器，智能玩具/对话玩具，PMP/游戏机，自动售货机，地铁自动售票机，导游机，楼宇电视的广告点播，照明系统的声控等等。

5. 应用实例

用语音控制开灯关灯，并具有语音提示，实现人机互动操作。需要器材有：Arduino 控制器，Arduino 扩展板，语音识别模块，MP3 模块，喇叭，LED 灯，电源适配器，USB 电缆。



Voice Recognition 模块 Arduino 演示代码

```
#include <Wire.h>
#include <Voice.h>
#define ArduinoPlayer_address 0x35 //定义 ArduinoPlayer 语音播放模块 I2C 地址
#define SUM 2 //SUM识别关键词的个数, 最大不超过50个
uint8 nAsrStatus=0;
char sRecog[SUM][80] = {"kai deng", "guan deng"};
//每个关键词拼音字母个数不超过79个, 用户可以自行修改

int led=8; //被控 LED 灯接口
void TwiSend(const char *cmd) //I2C 发送命令函数
{
char len = 0;
```



```
len = strlen(cmd); //计算字符串长度
Wire.beginTransmission(ArduinoPlayer_address); // ArduinoPlayer 语音播放模块 I2C 地址
while(len--)
{
    Wire.send(*(cmd++));
    Wire.endTransmission();
}

void finally (unsigned char n)
{
    switch(n)
    {
        case 0: //开灯命令
            Serial.println( "kai deng");
            Serial.println( " ");
            TwiSend("\\:v 250\r\n");
            TwiSend("\\kaideng\r\n");
            delay(2000);
            digitalWrite(led,HIGH);
            break;
        case 1: //关灯命令
            Serial.println( "guan deng");
            Serial.println( " ");
            TwiSend("\\:v 250\r\n");
            TwiSend("\\guandeng\r\n");
            delay(2000);
            digitalWrite(led,LOW);
            break;
        default:
            Serial.println( "error");
            Serial.println( " ");
            break;
    }
}

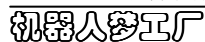
void ExtInt0Handler ()
{
    Voice.ProcessInt0();
    // LD3320 送出中断信号
}

void setup()
{
    Wire.begin();
    Serial.begin(9600);
    Voice.Initialise(MIC);
    attachInterrupt (0, ExtInt0Handler, LOW);
    pinMode(led, OUTPUT);
}
```

```
}  
void loop()  
{  
    uint8 nAsrRes;  
    nAsrStatus = LD_ASR_NONE;  
  
    while(1)  
    {  
        switch(nAsrStatus)  
        {  
            case LD_ASR_RUNNING:  
            case LD_ASR_ERROR:  
                break;  
            case LD_ASR_NONE:  
            {  
                nAsrStatus=LD_ASR_RUNNING;  
                if (Voice.RunASR(SUM, 80, sRecog)==0)  
                {  
                    nAsrStatus= LD_ASR_ERROR;  
                    Serial.println( "ASR_ERROR");  
                }  
  
                Serial.println( "ASR_RUNNING.....");  
  
                break;  
            }  
            case LD_ASR_FOUNDDOK:  
            {  
                nAsrRes =Voice. LD_GetResult(); // 一次 ASR 识别流程结束，去取识别结果  
                finally(nAsrRes);  
                nAsrStatus = LD_ASR_NONE;  
                break;  
            }  
            case LD_ASR_FOUNDDZERO:  
            default:  
            {  
  
                nAsrStatus = LD_ASR_NONE;  
                break;  
            }  
        }  
        delay(500);  
    }  
}
```

本手册中的演示代码均通过验证！

DFRduino 语音识别模块



TEL:010-82355005

北京龙凡汇众机器人科技有限公司

E_mail: service@dfrobot.com

本手册版权归 DFRobot 所有！

发布日期	版本号	备注
2011 年 12 月 21 日	V1.0	建文档

Copyright by DFRobot