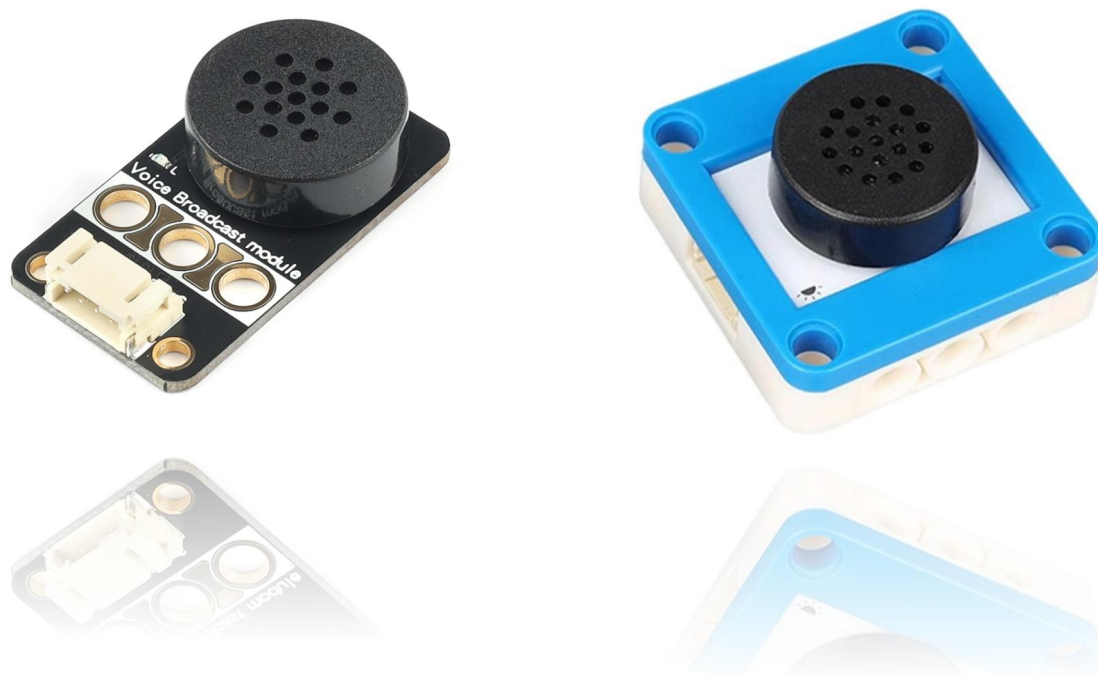


# 语音播报OTP模块用户手册

V1.1



## 发布说明:

日期	版本	内容
20210706	V1.0	初版
20221016	V1.1	更新黑板模块

# YFROBOT

# 目录

1. 简介.....	- 1 -
2. 规格参数.....	- 1 -
3. 引脚说明.....	- 2 -
4. 控制说明.....	- 2 -
单码发送模式说明.....	- 3 -
发送注意事项: .....	- 3 -
发送示例.....	- 4 -
连码发送模式说明.....	- 4 -
发送注意事项: .....	- 4 -
5. 模块语音列表.....	- 5 -
6. 应用示例.....	- 6 -
6.1. 电路连接.....	- 6 -
6.2. Arduino IDE示例代码.....	- 7 -
6.2.1. 语音播报-简单测试.....	- 7 -
6.2.2. 语音播报-音量控制.....	- 7 -
6.2.3. 语音播报-循环播放和停止播放.....	- 8 -
6.2.4. 语音播报-组合语音播放.....	- 8 -
6.2.5. 语音播报-连码播放语音.....	- 9 -
7. 附录.....	- 10 -
7.1. 附录1-语音列表.....	- 10 -
7.2. 附录1-UNO接口说明.....	- 11 -
7.3. 附录1-Arduino如何导入库? .....	- 11 -
7.4. 附录2-Mind+如何导入库? .....	- 11 -
7.5. 附录3-Mixly如何导入库? .....	- 11 -
7.6. 附录4-MakeCode如何导入扩展? .....	- 12 -
8. 联系我们.....	- 13 -

## 1. 简介

语音播报OTP模块(乐高外壳、黑板), 采用NVD系列工业级OTP语音芯片制作的一款具有**固定**语音播放功能的模块。模块采用一线串口控制方式, 只需要一个IO口即可实现语音播放功能。模块集成0.5W喇叭, 支持8级音量调节, 支持循环播放功能。

**OTP (One Time Programmable)** 是单片机的一种存储器类型, 意思是一次性可编程: 程序烧入单片机后, 将不可再次更改和清除。

语音播报OTP模块具有统一的兼容乐高积木的安装孔, 可轻松完成乐高积木的拼接, 实现创意设计。

## 2. 规格参数

工作电压: DC 3.3-5V

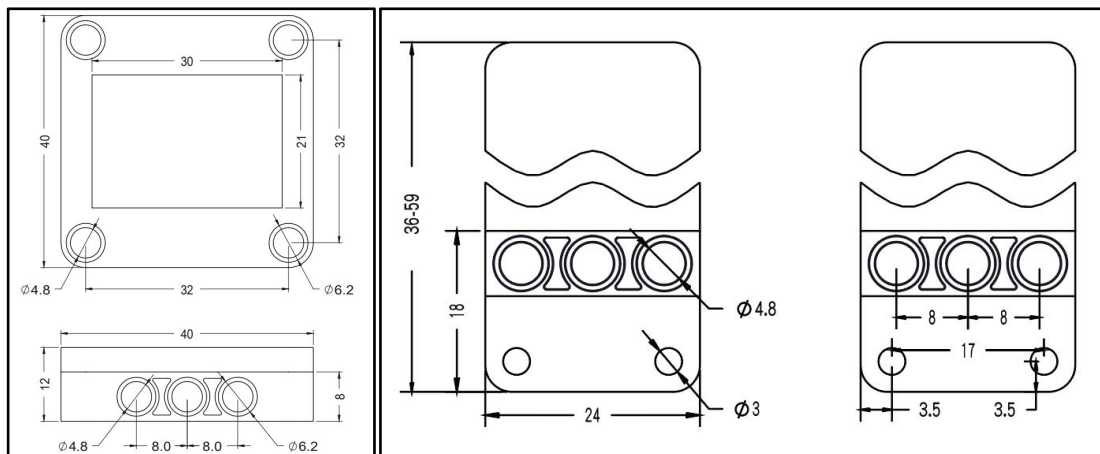
通讯类型: 一线通信

内置语音数: 127组语音

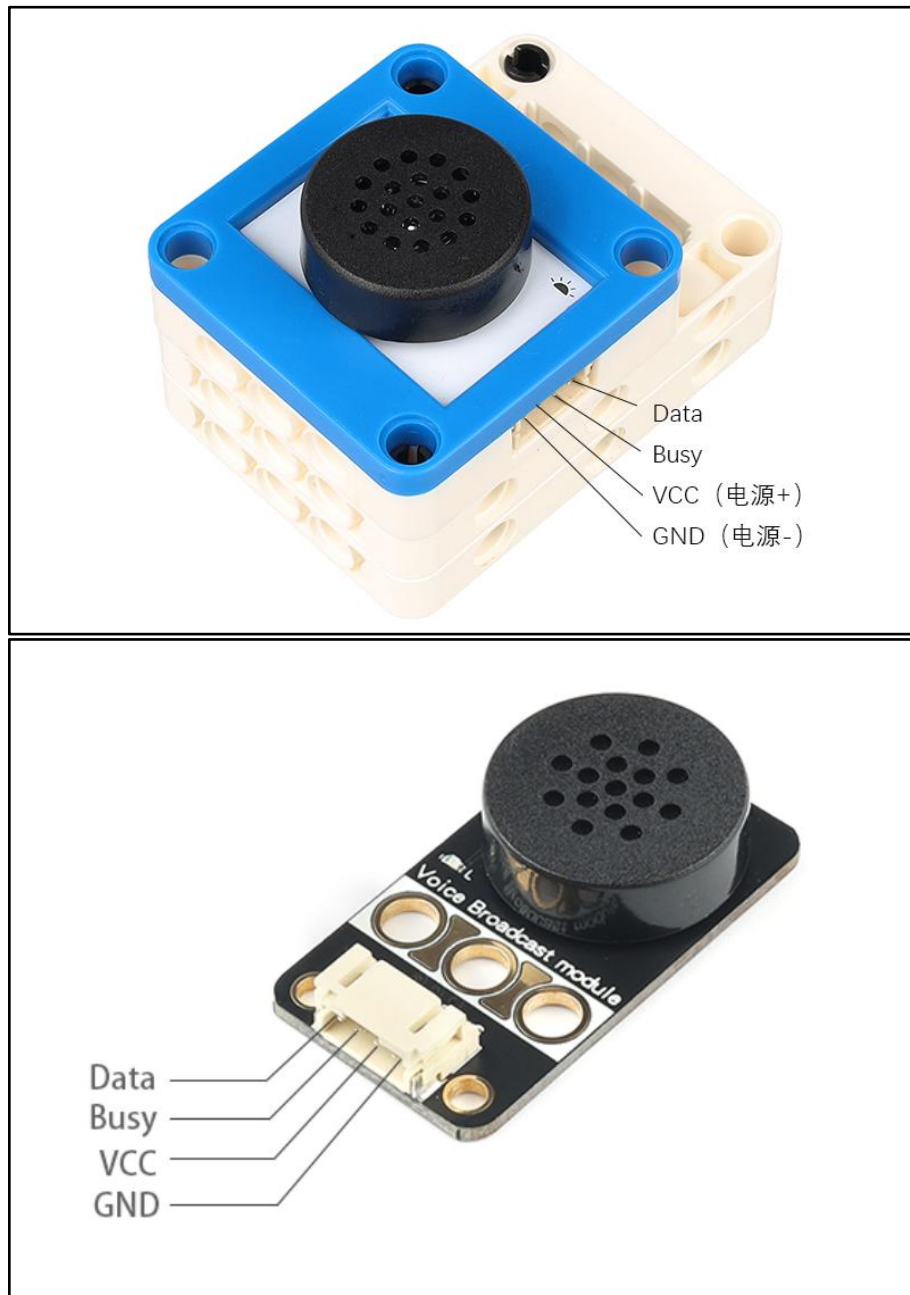
音量调节: 8级音量

喇叭功率: 0.5W

外壳尺寸: 单位MM, 整体高度约20MM



### 3. 引脚说明



Data端口：一线控制端口，控制播报语音，设置音量等功能；

Busy端口：Busy引脚；高电平表示无输出，低电平表示正在播报语音。（**可以不用此接口**）

### 4. 控制说明

语音播报模块采用一线串口控制模式，一线串口控制指通过Data数据端口来

控制任意一段语音的触发播放及停止。

模块控制命令：

串口命令	功能
00H	播放第1段语音
01H	播放第2段语音
· XXH ·	· 播放第N段语音 ·
7EH	播放第127段语音
E0H~E7H	控制8级音量，E0音量最小，E7音量最大，默认最大
F1H	连码头码命令
F3H	连码尾码命令
F4H	连码静音命令，F1后面跟一个字节，代表静音时间，10ms为单位。
F2H	循环指令，播放时发此指令循环该段语音。
FEH	停止指令，停止播放语音

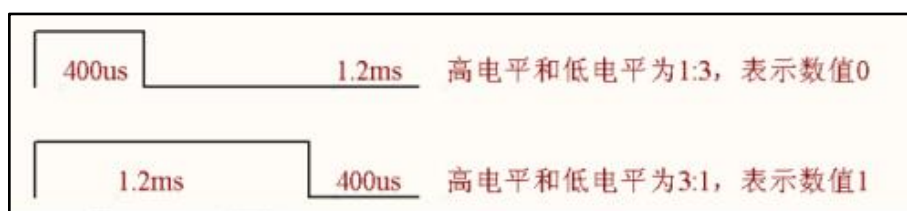
备注：音量命令，循环命令，停止命令都是单个字节发送。静音命令在连码才有效。

## 单码发送模式说明

Data端口发送00-7E命令，则播报对应语音地址下的语音；发一个地址则播放一次，正在播放时可以被新的命令打断，播放新地址的语音；播放期间接收到循环命令则循环当前语音；接收到停止命令则停止播放并清除循环标志；音量命令E1-E7可在播放的时或者待机的时接收。

发送注意事项：

- 语音命令发送，需要一个2-5ms（推荐5ms）的起始位，即拉低信号；
- 语音命令发送，需间隔50-100ms（推荐100ms）；
- Data端口发送数据，采用电平占空比不同来表示不同数据；高电平在前，低电平在后；1:3 表示0，3:1表示1，如下图400us/1200us（推荐使用800us/2400us）；

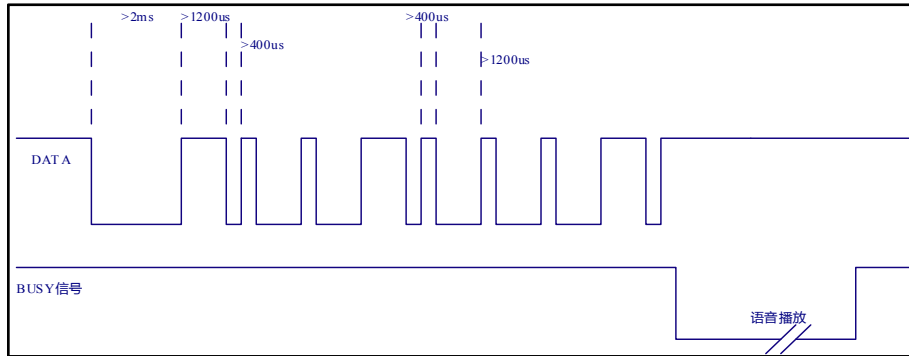


- Data端口发送数据，先发送低位，再发送高位；

- 循环命令生效时，如有停止命令或播放新一段语音命令，循环命令即失效；
- 语音组合功能只需在MCU检测到BUSY信号为HIGH（高电平-空闲）时，连续发码即可！

### 发送示例

- 下图示例发送的是89H，时序为400us：1200us（推荐使用800us：2400us）



### 连码发送模式说明

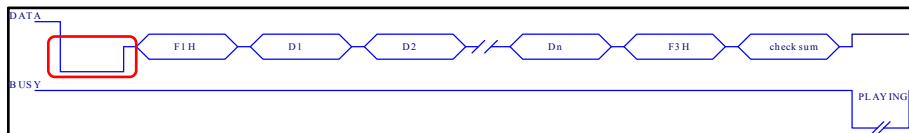
连码就是单片机连续发出数据串组合，语音芯片接收到数据串后，组合语音播放出来。单片机可以组合播放不同地址语音，中间也可以插入静音。数据结构如下。

$$[F1H] + [D1] + [D2] + \dots + [Dn] + [F3H] + [checksum]$$

解析说明：头码F1H+第一个语音地址+第二个语音地址+ ……，接收到尾码F3H代表语音地址数据接收完成，再接收一个校验和（checksum）核对，数据成功接收后将所有地址按顺序播放语音。

### 发送注意事项：

- 数据校验和（checksum）是所有字节的数据（包含F1，F3）和取低8位。收到正确校验码才确认本轮数据接收成功，否则失败；
- 连码数据与数据之间的最大间隔不得超过50ms，超过则丢弃本轮接收；
- 仅第一个字节需要起始位，后面的字节不需要起始位；图示如下：



- 如需插入连码静音，F4+xx；如果xx为02H的话，相当于延时20ms；
- 连码播放期间可以被新语音播报指令（包含单码，连码）打断。

## 5. 模块语音列表

模块具有127段语音，可通过程序控制单独播放或组合播放。前68段语音兼容电子学会机器人等级考试三四级指定教材第五章语音（教材语音内容见附录1），语音内容列表：

地址	语音内容	地址	语音内容	地址	语音内容
0	老师	23	祝	46	转
1	爸爸	24	慢走	47	左
2	妈妈	25	欢迎光临	48	右
3	爷爷	26	亲爱的	49	请
4	奶奶	27	同学们	50	已
5	姥姥	28	工作辛苦了	51	现在
6	姥爷	29	点	52	是
7	哥哥	30	打开	53	红灯
8	姐姐	31	关闭	54	绿灯
9	叔叔	32	千	55	黄灯
10	阿姨	33	百	56	温度
11	上午	34	十/时	57	湿度
12	下午	35	1	58	欢迎常来
13	晚上	36	2	59	还有
14	前方	37	3	60	秒
15	厘米	38	4	61	分
16	新年快乐	39	5	62	变
17	身体健康	40	6	63	等
18	工作顺利	41	7	64	下一次
19	学习进步	42	8	65	功能
20	您好	43	9	66	障碍物
21	谢谢	44	0	67	世界那么大， 我想去看看
22	的	45	当前		

后59段语音为拓展设计，扩展了语音模块的使用范围，可以播报时间、天气、风向、空气质量等。语音内容列表：

地址	语音内容	地址	语音内容	地址	语音内容
----	------	----	------	----	------

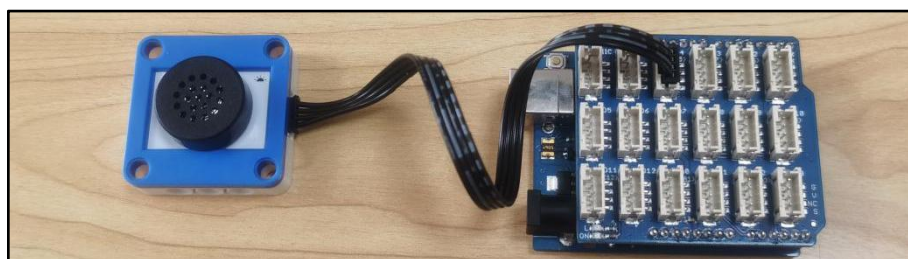
68	今天	88	雨	108	良
69	年	89	雷阵	109	轻度污染
70	月	90	小	110	中度污染
71	日	91	中	111	重度污染
72	星期	92	大	112	上
73	农历	93	夹	113	下
74	现在时刻	94	雪	114	接近
75	北京时间	95	雾	115	远离
76	整	96	霾	116	灯
77	度	97	风	117	风扇
78	百分之	98	东	118	红色
79	距离	99	南	119	绿色
80	厘米	100	西	120	蓝色
81	明天	101	北	121	黄色
82	天气	102	到	122	白色
83	白天	103	级	123	叮-音效
84	夜间	104	偏	124	滴滴滴-闹铃音效
85	晴	105	方向	125	叮叮叮叮升-音效
86	多云	106	空气质量	126	叮叮叮叮降-音效
87	阴	107	优		

## 6. 应用示例

**⚠ 注意：**硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！

### 6.1. 电路连接

语音播报OTP模块的控制端 G、V、Busy、Data分别连接 Arduino UNO的GND、VCC、D5、D4引脚。

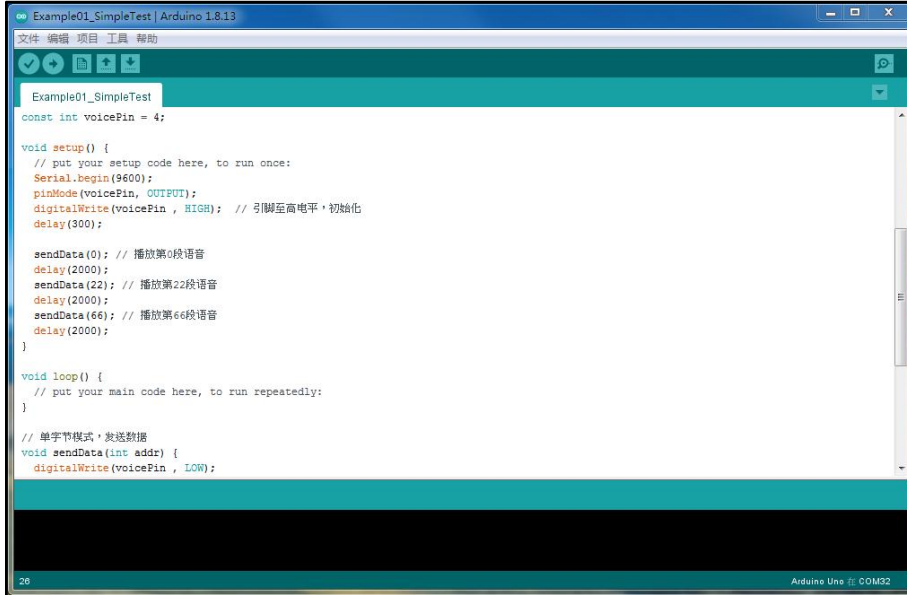




## 6.2. Arduino IDE示例代码

### 6.2.1. 语音播报-简单测试

打开程序：“..\Arduino IDE例程\Example01\_SimpleTest”，编译上传。



```
Example01_SimpleTest | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example01_SimpleTest
const int voicePin = 4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(voicePin, OUTPUT);
  digitalWrite(voicePin, HIGH); // 引脚至高电平，初始化
  delay(300);

  sendData(0); // 播放第0段语音
  delay(2000);
  sendData(22); // 播放第22段语音
  delay(2000);
  sendData(66); // 播放第66段语音
  delay(2000);
}

void loop() {
  // put your main code here, to run repeatedly:
}

// 单字节模式，发送数据
void sendData(int addr) {
  digitalWrite(voicePin, LOW);
}
```

#### 程序运行结果

模块播放第0，22，66段语音。

### 6.2.2. 语音播报-音量控制

打开程序：“..\Arduino IDE例程\Example02\_VolumeControl”，编译上传并观察结果。



```
Example02_VolumeControl | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example02_VolumeControl
const int voicePin = 4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(voicePin, OUTPUT);
  digitalWrite(voicePin, HIGH); // 引脚至高电平，初始化
  delay(300);
  sendData(0xE0); // 设置音量最小等级0
  delay(50);
  sendData(0); // 播放第0段语音
  delay(2000);
  sendData(0xE3); // 设置音量等级3
  delay(50);
  sendData(22); // 播放第22段语音
  delay(2000);
  sendData(0xE7); // 设置音量最大等级7
  delay(50);
  sendData(66); // 播放第66段语音
  delay(2000);
}

void loop() {
}
```

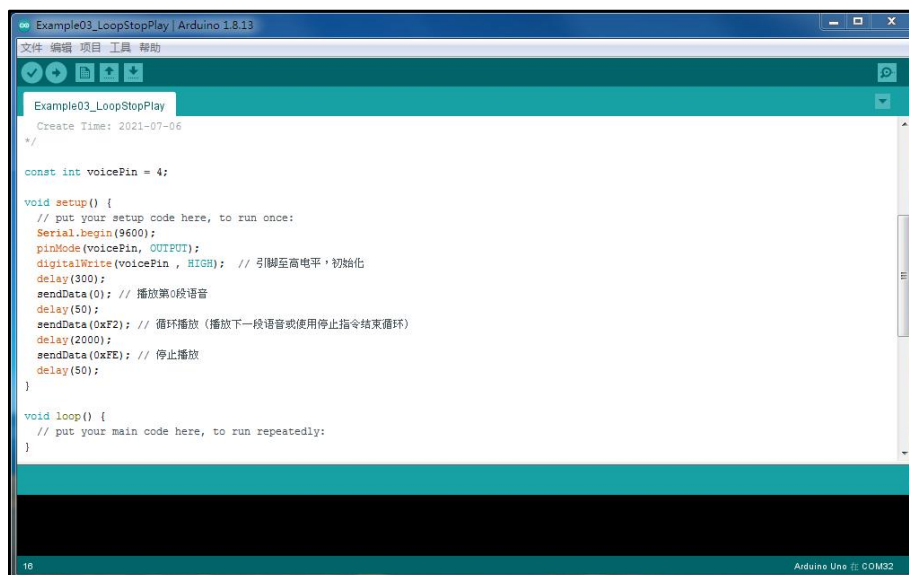
#### 程序运行结果

模块最小音量等级0播放第0段语音，音量等级3播放22段语音，音量等级最

高7播放66段语音。

### 6.2.3. 语音播报-循环播放和停止播放

打开程序：“..\Arduino IDE例程\Example03\_LoopStopPlay”，编译上传并观察结果。



```
Example03_LoopStopPlay | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example03_LoopStopPlay
Create Time: 2021-07-06
*/

const int voicePin = 4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(voicePin, OUTPUT);
  digitalWrite(voicePin, HIGH); // 引脚至高电平，初始化
  delay(300);
  sendData(0); // 播放第0段语音
  delay(50);
  sendData(0xF2); // 循环播放（播放下一段语音或使用停止指令结束循环）
  delay(2000);
  sendData(0xFE); // 停止播放
  delay(50);
}

void loop() {
  // put your main code here, to run repeatedly:
}

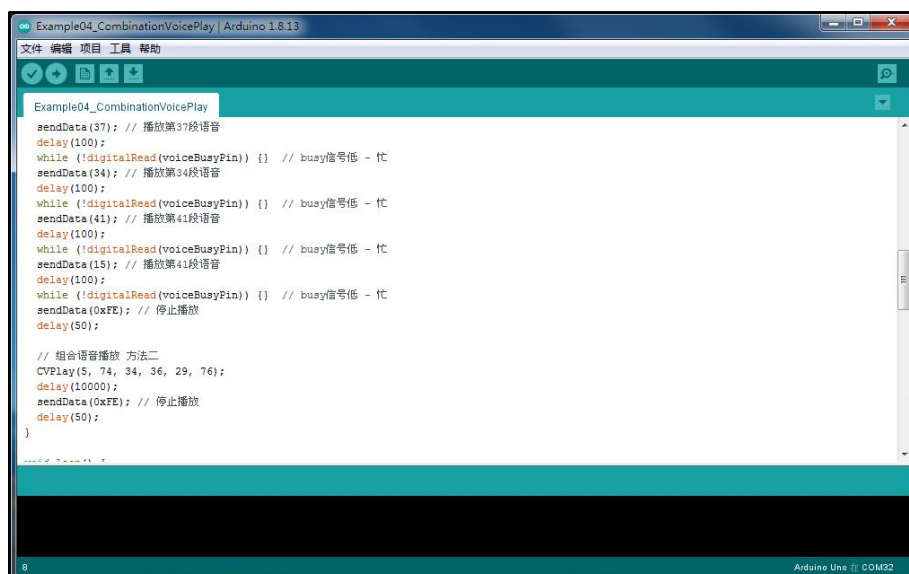
16 Arduino Uno 在 COM32
```

#### 程序运行结果

模块播放第0段语音，设置循环播放，等待2s后，设置停止播放。

### 6.2.4. 语音播报-组合语音播放

打开程序：“..\Arduino IDE例程\Example04\_CombinationVoicePlay”，编译上传并观察结果。



```
Example04_CombinationVoicePlay | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example04_CombinationVoicePlay
sendData(37); // 播放第37段语音
delay(100);
while (!digitalRead(voiceBusyPin)) {} // busy信号低 - 忙
sendData(34); // 播放第34段语音
delay(100);
while (!digitalRead(voiceBusyPin)) {} // busy信号低 - 忙
sendData(41); // 播放第41段语音
delay(100);
while (!digitalRead(voiceBusyPin)) {} // busy信号低 - 忙
sendData(15); // 播放第15段语音
delay(100);
while (!digitalRead(voiceBusyPin)) {} // busy信号低 - 忙
sendData(0xFE); // 停止播放
delay(50);

// 组合语音播放 方法二
CVPlay(5, 74, 34, 36, 29, 76);
delay(10000);
sendData(0xFE); // 停止播放
delay(50);
}

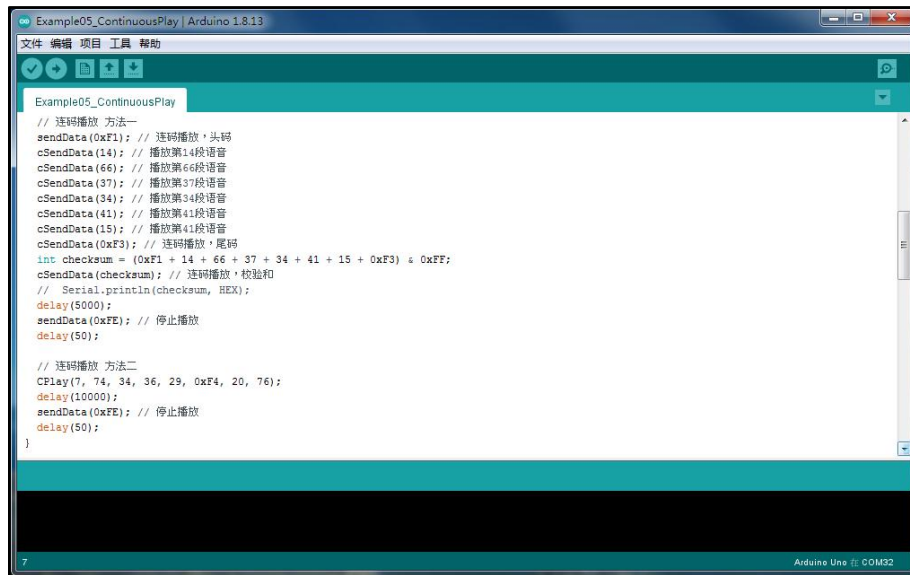
8 Arduino Uno 在 COM32
```

## 程序运行结果

两种方式组合各段语音播放，第一组语音：“前方障碍物37厘米”；第二组语音：“现在时刻12点整”。

### 6.2.5. 语音播报-连码播放语音

打开程序：“..\Arduino IDE\例程\Example05\_ContinuousPlay”，编译上传并观察结果。



```
Example05_ContinuousPlay
// 连码播放 方法一
sendData(0xF1); // 连码播放，头码
cSendData(14); // 播放第14段语音
cSendData(66); // 播放第66段语音
cSendData(37); // 播放第37段语音
cSendData(34); // 播放第34段语音
cSendData(41); // 播放第41段语音
cSendData(15); // 播放第41段语音
cSendData(0xF3); // 连码播放，尾码
int checksum = (0xF1 + 14 + 66 + 37 + 34 + 41 + 15 + 0xF3) & 0xFF;
cSendData(checksum); // 连码播放，校验和
// Serial.println(checksum, HEX);
delay(5000);
sendData(0xFE); // 停止播放
delay(50);

// 连码播放 方法二
CPPlay(7, 74, 34, 36, 29, 0xF4, 20, 76);
delay(10000);
sendData(0xFE); // 停止播放
delay(50);
}
```

## 程序运行结果

两种方式连码播放语音，第一组语音：“前方障碍物37厘米”；第二组语音：“现在时刻12点整”。

至此语音播报OTP模块功能全部展示完毕，实际应用中，用户可以参考上面的程序调用不同语音，组成需要的语句。

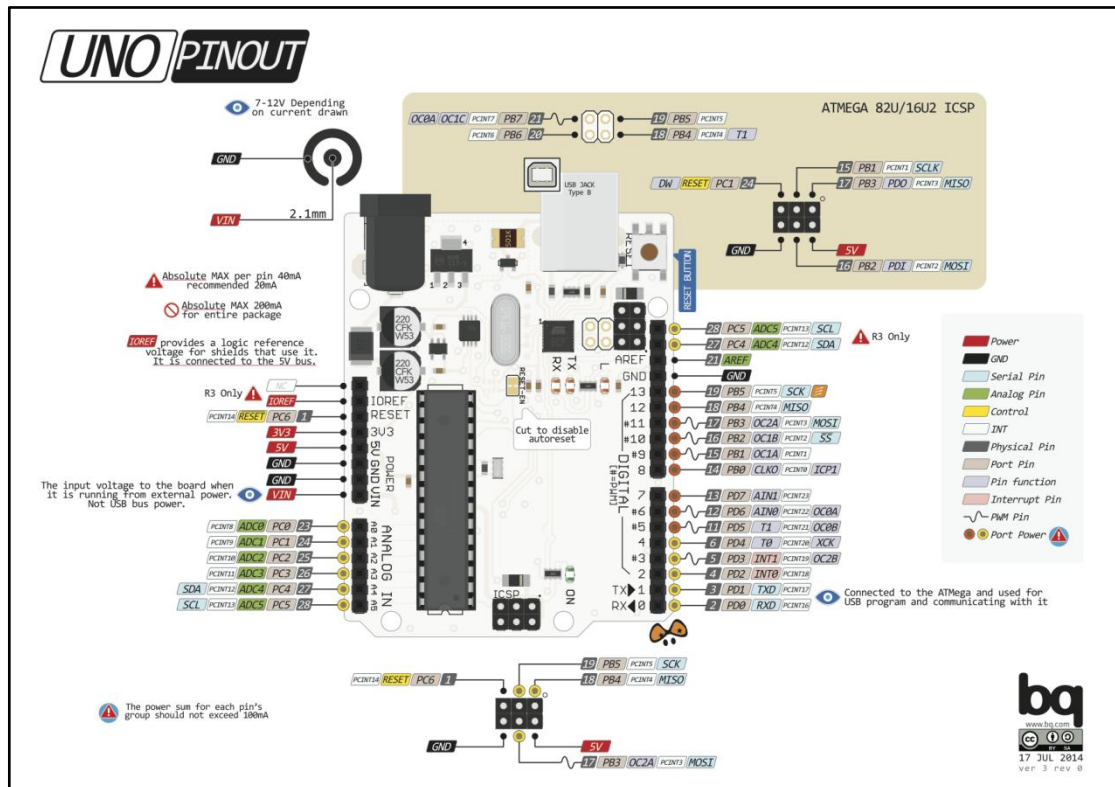
## 7. 附录

### 7.1. 附录1-语音列表

电子学会机器人等级考试三四级指定教材第五章，语音模块内容：

附录 B		语音模块内容对照表			
地 址	内 容	地 址	内 容	地 址	内 容
1	老师	24	祝	47	转
2	爸爸	25	慢走	48	左
3	妈妈	26	欢迎光临	49	右
4	爷爷	27	亲爱的	50	请
5	奶奶	28	同学们	51	已
6	姥姥	29	工作辛苦了	52	现在
7	姥爷	30	点	53	是
8	哥哥	31	打开	54	红灯
9	姐姐	32	关闭	55	绿灯
10	叔叔	33	千	56	黄灯
11	阿姨	34	百	57	温度
12	上午	35	十	58	湿度
13	下午	36	1	59	欢迎常来
14	晚上	37	2	60	还有
15	前方	38	3	61	秒
16	厘米	39	4	62	分
17	新年快乐	40	5	63	变
18	身体健康	41	6	64	等
19	工作顺利	42	7	65	下一次
20	学习进步	43	8	66	功能
21	您好	44	9	67	障碍物
22	谢谢	45	0	68	世界那么大， 我想去看看
23	的	46	当前		

## 7.2. 附录1-UNO接口说明



注：UNO官方版本和兼容版本大部分功能都相同

## 7.3. 附录1-Arduino如何导入库？

教程中有些需要使用库；如何将其导入到自己的Arduino IDE编译器中？

跳转网页查看视频教程：[点击跳转](#)。

## 7.4. 附录2-Mind+如何导入库？

教程中有些需要使用库；如何将其导入到的编译器中？

跳转网页查看教程：[点击跳转](#)。

## 7.5. 附录3-Mixly如何导入库？

教程中有些需要使用库；如何将其导入到的编译器中？

跳转网页查看教程：[点击跳转](#)。

## 7.6. 附录4-MakeCode如何导入扩展?

教程中有些需要使用扩展；如何将其导入到的编译器中？

跳转网页查看教程：[点击跳转](#)。

## 8. 联系我们

YFROBOT网站: [www.yfrobot.com](http://www.yfrobot.com) / [www.yfrobot.com.cn](http://www.yfrobot.com.cn)

手机: 17696701116 (微信/QQ同号)

微信公众号: YFRobotStudio

QQ群: [243067479](https://jq.qq.com/?_w=1027&q=243067479)

邮件: [yfrobot@qq.com](mailto:yfrobot@qq.com)

技术微信



微信公众号



# YFROBOT

## 免责声明和版权公告

本文中的信息, 包括供参考的 URL 地址, 如有变更, 恕不另行通知。

文档“按现状”提供, 不负任何担保责任, 包括对适销性、适用于特定用途或非侵权性的任何担保, 和任何提案、规格或样品在他处提到的任何担保。本文档不 负任何责任, 包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可, 不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各所有者财产, 特此声明。

版权归 © 2021 YFROBOT所有。保留所有权利。