

音频播放MP3模块用户手册

V1.2



发布说明:

日期	版本	内容
20210710	V1.0	初版
20211228	V1.1	更新优化Arduino程序函数
20221016	V1.2	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 -
4.1. 协议时序说明.....	- 3 -
4.1.1. 注意事项.....	- 3 -
4.2. 指令说明.....	- 3 -
4.2.1. 注意事项说明.....	- 4 -
5. 应用示例.....	- 6 -
5.1. Arduino IDE示例代码.....	- 6 -
5.1.1. 音频播放-简单测试.....	- 6 -
5.1.2. 音频播放-自制MP3播放器.....	- 7 -
5.2. 音频模块功能演示视频参考.....	- 9 -
6. 附录.....	- 10 -
6.1. 附录1-UNO接口说明.....	- 10 -
6.2. 附录1-Arduino如何导入库?	- 10 -
6.3. 附录2-Mind+如何导入库?	- 10 -
6.4. 附录3-Mixly如何导入库?	- 10 -
6.5. 附录4-MakeCode如何导入扩展?	- 11 -
7. 联系我们.....	- 12 -

1. 简介

音频播放MP3模块（乐高外壳、黑板），采用DY-SV17F智能语音模块制作的一款具有自定义音频播放功能的模块。模块采用一线串口控制方式，只需要一个IO口即可实现音频播放功能。可通过数据线连接电脑更新音频（MP3格式/WAV格式-采样率不高于44k）文件，支持4MByte-flash存储音频文件，集成0.5W喇叭，支持播放、循环、暂停、停止、睡眠、选曲、音量加减等功能。

注意：模块连接USB到电脑导入音频时，请断开电路接线，请断开电路接线，请断开电路接线（尤其当使用3.3V主控板时-包含：Micro:Bit，掌控板等）！

注意：模块不支持热插拔，请先连接好电路再接电源！刷入新音频后，需断电重启！

音频播放MP3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：DC3.3V-5V（**3.3V使用时请注意**，连接USB时务必断开电路连接，如有不明白请勿连接，请先联系[技术人员](#)）

通讯类型：一线串口通信

存储空间：4MByte flash

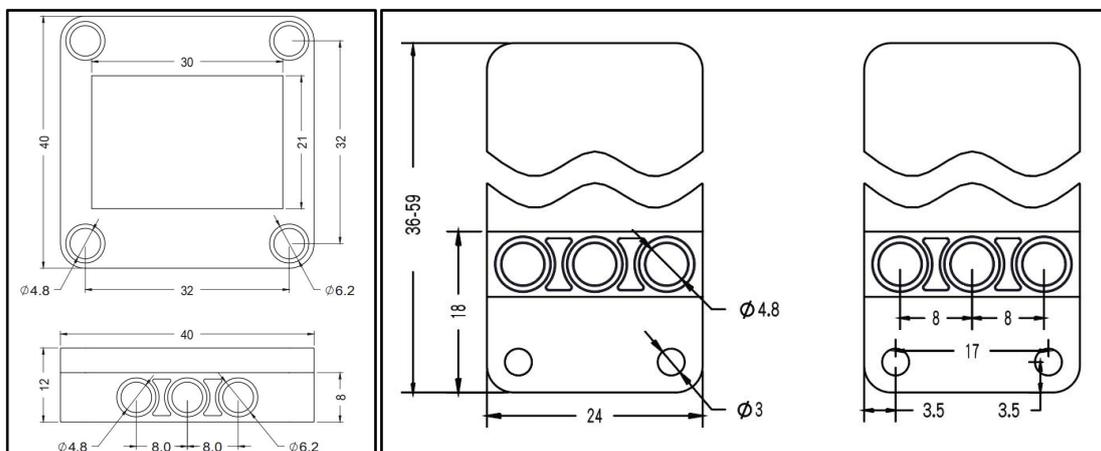
内置语音：自定义（可通过电脑更新，格式MP3/WAV-采样率不高于44K）

音频采样率：8/11.025/12/16/22.05/24/32/44.1/48

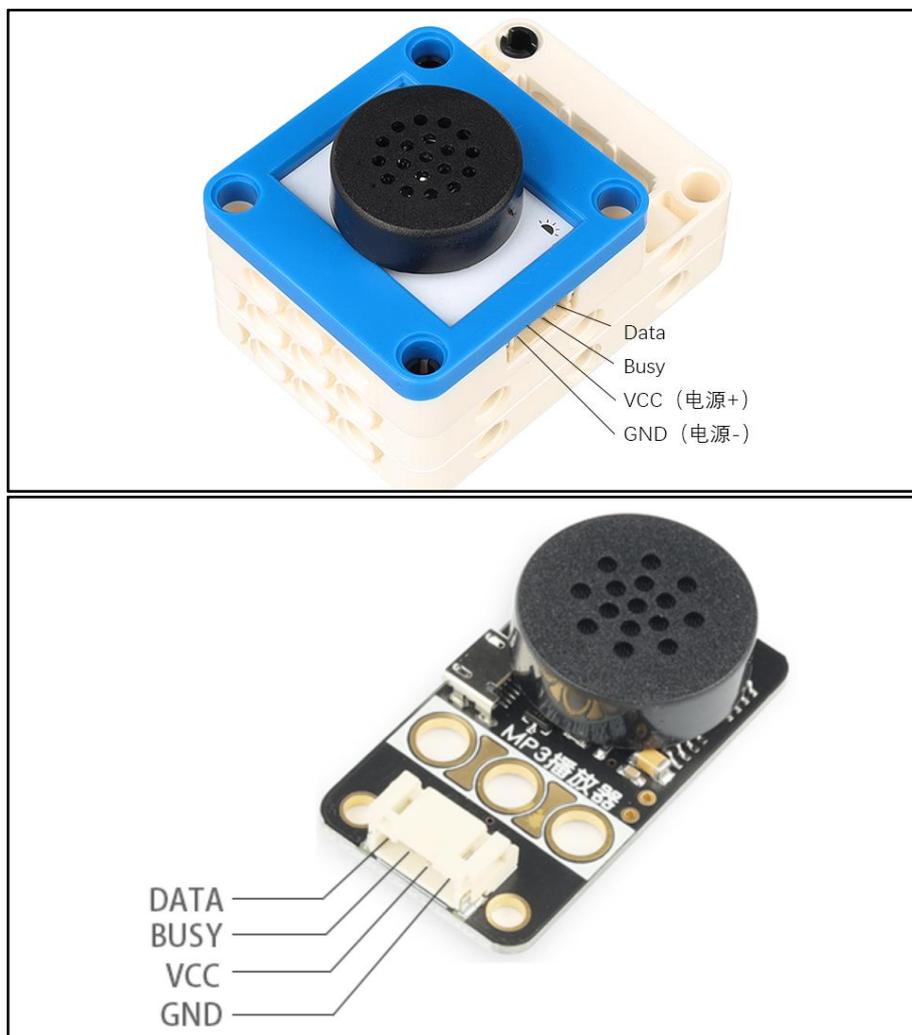
音量调节：31级音量（0-30，默认20）

喇叭功率：1W

外壳尺寸：单位MM



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（此端口可不用）：上电30ms内做为模式配置引脚3（忽略，了解即可）；30ms后作为Busy信号输出引脚，高电平表示正在播放，低电平表示无播放。

Micro USB端口：连接电脑存储更新音频文件（mp3/wav格式），推荐使用mp3格式，wav格式文件采样率不得高于44k。

注意：刷入音频时，请勿连接电路以免烧坏电路；重新刷入新音频文件后，一定要断电重启！！！！

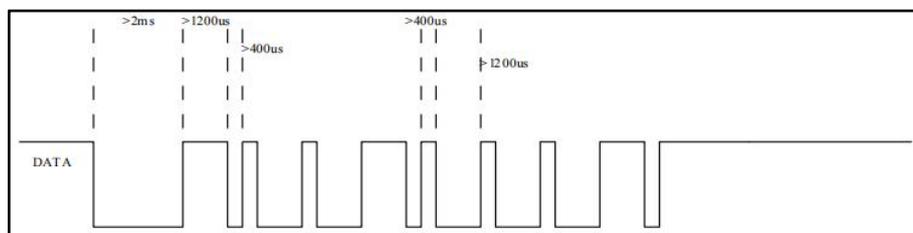
4. 控制说明

音频播放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或其他功能。

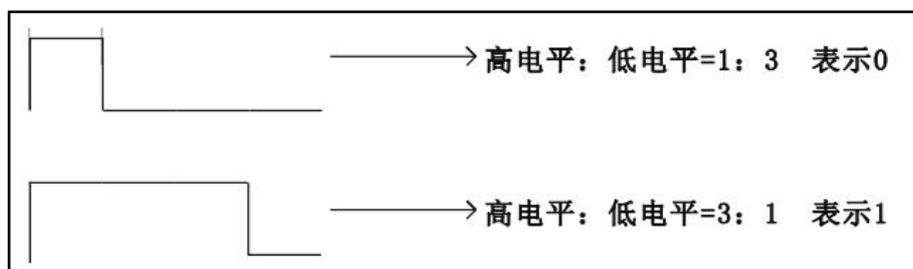
4.1. 协议时序说明

Data端口发送语音地址，正在播放时可以被新的命令打断。下图示例发送的是89H，时序：



4.1.1. 注意事项

- 指令发送，需要一个2-3ms（推荐3ms）的起始位，即拉低信号；指令之间间隔5ms即可；
- 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（推荐400us : 1200us），如下图；



- Data端口发送数据，先发送低位，再发送高位；
- BUSY信号为HIGH（高电平）时，表示正在播放音频；Busy信号响应时间由于模块运行机制问题，无法确定，大概50-500ms。

4.2. 指令说明

模块控制命令：

指令(HEX)	功能	说明
00	数字0	数字0—9可以用需要数字的功能，比如选曲、设置音量、设置EQ、设置播放模式、设置通道、设置插播曲目，先发数字后发功能指令
01	数字1	
02	数字2	
03	数字3	
04	数字4	
05	数字5	

06	数字6	
07	数字7	
08	数字8	
09	数字9	
0A	清零数字	清除发送的数字
0B	选曲确认	配合数字实现 (选曲、插播根据曲目名播放)
0C	设置音量(总31级, 0-30级, 默认为20级)	
0D	设置EQ [®]	
0E	设置播放模式 [®]	
0F	设置通道(不支持)	
10	设置插播曲目	
11	播放	
12	暂停	
13	停止	停止所有模式下正在播放的音乐
14	上一曲	曲目序号由存储顺序决定, 执行上一曲、下一曲
15	下一曲	
16	上一目录	播放上一目录中最后一首曲目
17	下一目录	播放下一目录中第一首曲目
18	选择SD卡	不支持, 默认识别配置FLASH
19	选择U盘	
1A	选择FLASH	
1B	系统睡眠	
1C	结束播放	结束当前播放, 在插播曲目时使用, 结束插播曲目, 回到原来播放位置继续播放。

4.2.1. 注意事项说明

- 曲目名命名形式: 五位数字+格式, 如00001.mp3、00255.mp3;
- “曲目插播”只有1级插播, 连续插播会覆盖前面的插播曲目(插播立即播放), 曲目播放结束回到第一次插播断点继续播放;
- “选曲”和“插播”是根据曲目名字播放, 例如曲目名为“00123.mp3”, 则选曲输入的数据依次为“0x01”“0x02”“0x03”“0x0B”, 完成选曲;

- ① EQ: [均衡器 \(Equalizer\)](#)，EQ模式有00-Normal（普通）、01-POP（流行）、02-ROCK（摇滚）、03-JAZZ（爵士）、04-Classical（古典），不同的音乐配置不同的模式，效果更好。
- ② 播放模式，上电默认02-单曲停止；模式有：
 1. 00-全盘循环：顺序播放全盘曲目，播放完后循环播放；
 2. 01-单曲循环：循环播放当前曲目；
 3. 02-单曲停止：播放完当前曲目一次，停止播放；
 4. 03-全盘随机：随机播放盘符内曲目；
 5. 04-目录循环：顺序播放当前目录文件夹内曲目，播放完后循环播放，目录不包含子目录；
 6. 05-目录随机：在当前目录文件夹内随机播放，目录不包含子目录；
 7. 06-目录顺序：顺序播放当前目录文件夹内曲目，播放完后停止播放，目录不包含子目录；
 8. 07-全盘顺序：顺序播放全盘曲目，播放完后停止。

5. 应用示例

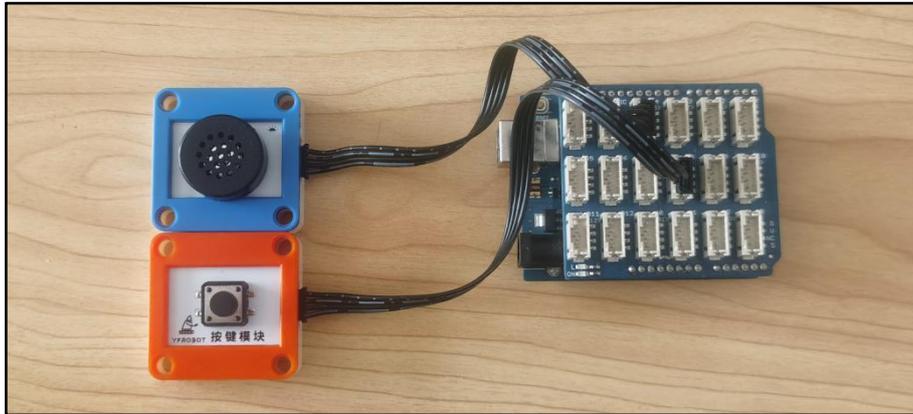
⚠ 注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！

5.1. Arduino IDE示例代码

5.1.1. 音频播放-简单测试

电路连接

音频播放MP3模块的引脚G、V、Busy、Data分别连接 Arduino UNO的GND、VCC、D5、D4引脚；按键模块的G、V、S分别连接 Arduino UNO的GND、VCC、D8引脚。



打开程序：“..\Arduino IDE例程\Example01_AudioPlayerSimpleTest”，编译上传并观察结果。

⚠ 注意：此时你程序需要添加库文件“*OneButton.zip*”（路径“..\Arduino Library\OneButton.zip”），否则无法正常编译，添加方式见[附录2](#)

```
Example01_AudioPlayerSimpleTest | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example01_AudioPlayerSimpleTest
#define NEXT_CATALOGUE 0x17 //下一目录
#define SYS_HIBERNATION 0x1B //系统休眠
#define CLOSE_DOWN 0x1C //结束播放

#define dataPin 4 // 音频模块 data引脚 D4
#define busyPin 5 // 音频模块 Busy引脚 D5
#define btnPin 8 // 按键引脚 D8
#define LEDPin 13 // 指示灯引脚 D13

#define MusicNumber 11 // 现存音频数量，请根据实际情况修改，否则模块执行会出现错误

#include "OneButton.h"
OneButton button(btnPin, true);
boolean flagBT = true; // 记录busy 响应时间标志位

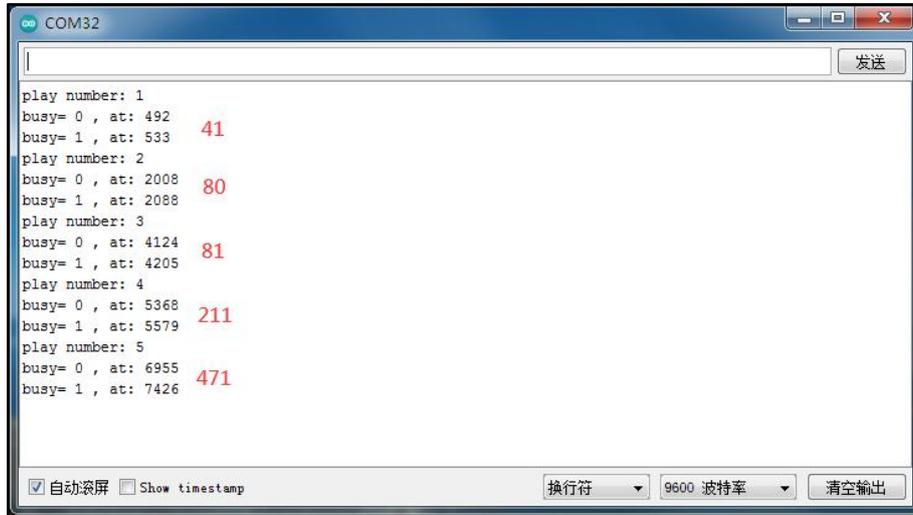
void setup() {
  // put your setup code here, to run once:
  // ...
}
```

程序运行结果

按下按键，音频模块播放对应音频（根据存储的音频）。

串口监视器打印busy信号响应时间，从打印信息中我们可以看出响应时间不

固定（这是由于模块运行机制导致的）。



```
COM32
play number: 1
busy= 0 , at: 492
busy= 1 , at: 533 41
play number: 2
busy= 0 , at: 2008
busy= 1 , at: 2088 80
play number: 3
busy= 0 , at: 4124
busy= 1 , at: 4205 81
play number: 4
busy= 0 , at: 5368
busy= 1 , at: 5579 211
play number: 5
busy= 0 , at: 6955
busy= 1 , at: 7426 471
自动滚屏 Show timestamp 换行符 9600 波特率 清空输出
```

代码解析

sendData()

函数：发送指令，例如：`sendData(3); // 发送3指令`

CPlay()

函数：发送多条指令，

例如：`CPlay(2, m, MUSIC_SELECT); // 选择 m 曲目播放`

2 - 表示3条指令

m - 表示第一条指令，发送的选择曲目

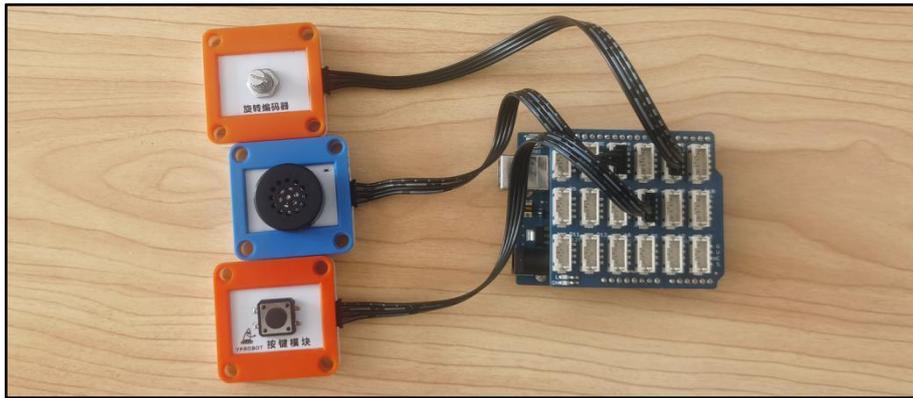
MUSIC_SELECT - 表示第二条指令，确认选择曲目并播放

查看程序：“..\Arduino IDE例程\Example01_AudioPlayerSimpleTest2”，另一种编程方法，比较那种更好使用。

5.1.2. 音频播放-自制MP3播放器

电路连接

音频播放MP3模块的引脚G、V、Busy、Data分别连接 Arduino UNO的GND、VCC、D5、D4引脚；按键模块的G、V、S分别连接 Arduino UNO的GND、VCC、D8引脚；旋转编码器的引脚G、V、A、B分别连接 Arduino UNO的GND、VCC、D6、D2引脚。



打开程序：“..\Arduino IDE例程\Example03_LoopStopPlay”，编译上传并观察结果。

⚠ 注意：此时你程序需要添加库文件“[OneButton.zip](#)”（路径“..\Arduino Library\OneButton.zip”），否则无法正常编译，添加方式见[附录2](#)

```
Example02_DIYMP3Player | Arduino 1.8.13
文件 编辑 项目 工具 帮助

Example02_DIYMP3Player
#define encoderPinA 6 // 编码器A相 D6
#define encoderPinB 2 // 编码器B相 D2
#define dataPin 4 // 音频模块 data引脚 D4
#define busyPin 5 // 音频模块 Busy引脚 D5
#define btnPin 8 // 按键引脚 D8
#define LEDPin 13 // 指示灯引脚 D13

#define MusicNumber 11 // 现存音频数量，请根据实际修改，否则模块执行会出现错误

#define MODE01 1 // 选曲模式
#define MODE02 2 // 设置音量模式
#define MODE03 3 // 设置EQ模式
#define MODE04 4 // 设置播放模式

#include "OneButton.h"
OneButton button(btnPin, true);

保存完成
项目使用了 5288 字节，占用了 (16%) 程序存储空间，最大为 32256 字节。
全局变量使用了 581 字节，(28%) 的动态内存，余留 1467 字节局部变量，最大为 2048 字节。

47 Arduino Uno 在 COM32
```

程序运行结果

单击按键，播放当前选择曲目；

双击按键，选择控制模式：1-选曲模式，2-设置音量，3-设置EQ，4-设置播放模式；

长按按键，停止播放。

串口监视器打印操作信息。

```
COM32
DIY MP3
Select Music: 1
Select Music: 2
play: 2
pause: 2
Select Music: 3
Select Music: 4
Current mode : Mode02 - Set Volume
Set Volume: 20
Current mode : Mode03 - Set EQ Mode
Set EQ Mode: 1
Set EQ Mode: 2
Set EQ Mode: 3
Set EQ Mode: 4
Set EQ Mode: 0
Current mode : Mode04 - Set play Mode
Set play Mode: 2 - Single Play Stop Mode.
Set play Mode: 3 - Random Play Mode.
Set play Mode: 4 - Catalog Loop Play Mode.
Set play Mode: 3 - Random Play Mode.
Set play Mode: 2 - Single Play Stop Mode.
Set play Mode: 1 - Single Lopp Play Mode.
play: 4
Set play Mode: 2 - Single Play Stop Mode.
play: 4
pause: 4
Set play Mode: 1 - Single Lopp Play Mode.
play: 4
Set play Mode: 2 - Single Play Stop Mode.
Set play Mode: 1 - Single Lopp Play Mode.
Set play Mode: 0 - Loop Play Mode.
play: 4
Set play Mode: 1 - Single Lopp Play Mode.
```

代码较为长，初学者演示理解功能即可。

代码解析

参考5.2.1例程。

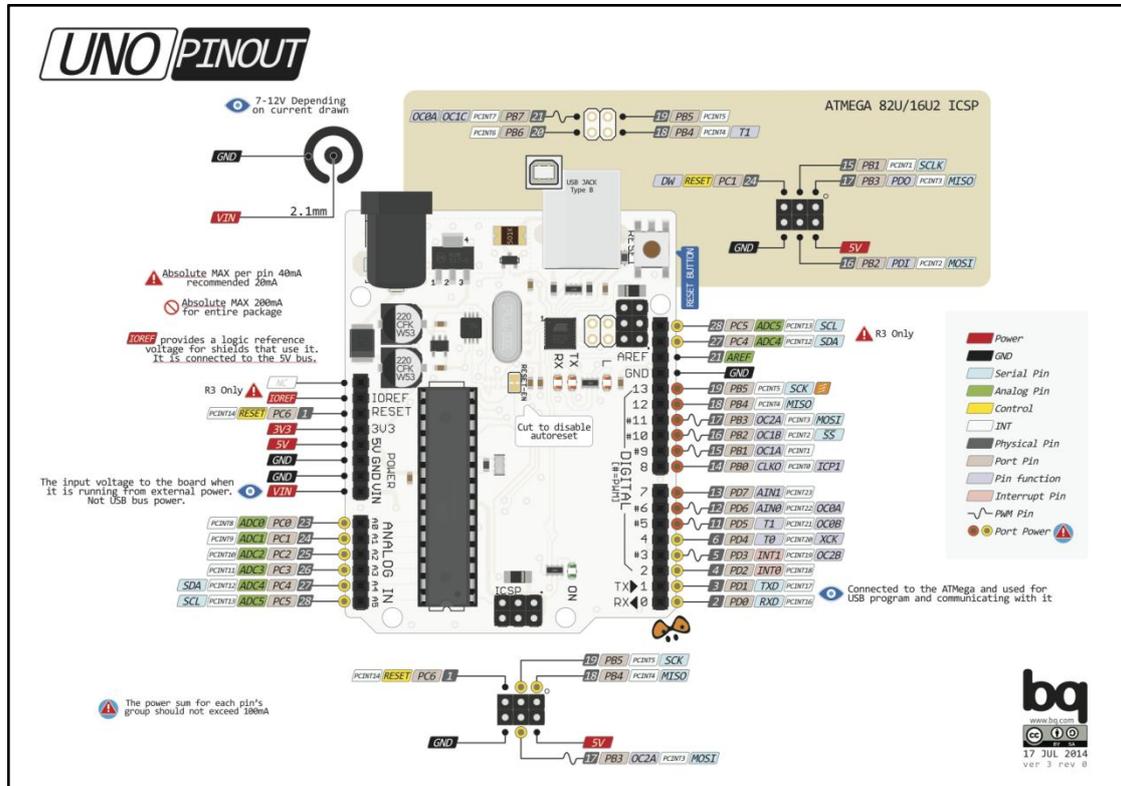
5.2. 音频模块功能演示视频参考

点击链接查看视频：

<http://yfrobot.com.cn/wiki/index.php?title=LEGO-%E9%9F%B3%E9%A2%91%E6%92%AD%E6%94%BE%E6%A8%A1%E5%9D%97>.

6. 附录

6.1. 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

6.2. 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](#)。

6.3. 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](#)。

6.4. 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](#)。

6.5. 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](#)。

7. 联系我们

YFROBOT网站: www.yfrobot.com / 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

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。