



- 首页
- 社区专区
- 新闻动态
- 最近更新
- 随机页面
- 帮助
- 工具箱
 - 链入页面
 - 相关更改
 - 特殊页面
 - 打印版本
 - 永久链接

页面 讨论

阅读

查看源代码

查看历史

(SKU:DFR0299)DFPlayer Mini模块

目录 [\[隐藏\]](#)

- 1 简介
- 2 产品参数
- 3 引脚说明
- 4 使用教程
 - 4.1 连线图
 - 4.2 样例代码
 - 4.2.1 "GetStarted" 库样例, 每3秒切换下一首
 - 4.2.2 "FullFunction" 库样例, 包含了所有的功能, 详见注释以及文档
- 5 资料说明
 - 5.1 串口模式
 - 5.2 AD按键模式
 - 5.3 IO模式
- 6 疑难解答
- 7 更多



DFPlayer Mini

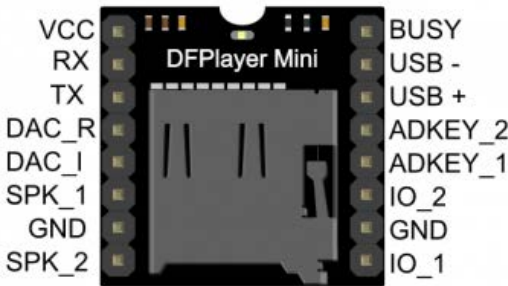
简介

DFPlayer Mini是一款小巧且价格低廉的MP3模块，可以直接接驳扬声器。模块配合供电电池、扬声器、按键可以单独使用，也可以通过串口控制，作为Arduino UNO或者是任何有串口的单片机的一个模块。模块本身完美的集成了MP3、WAV、WMA的硬解码。同时软件支持TF卡驱动，支持FAT16、FAT32文件系统。通过简单的串口指令即可完成播放指定的音乐，以及如何播放音乐等功能，无需繁琐的底层操作，使用方便，稳定可靠。

产品参数

- 支持采样率(KHz):8/11.025/12/16/22.05/24/32/44.1/48
- 24位DAC输出，动态范围支持:90dB，信噪比支持:85dB
- 完全支持FAT16、FAT32文件系统，最大支持32G的TF卡，支持32G的U盘、64M字节的NORFLASH
- 多种控制模式可选。IO控制模式、串口模式、AD按键控制模式
- 广播语插播功能，可以暂停正在播放的背景音乐。广告播放完毕回到背景音继续播放
- 音频数据按文件夹排序，最多支持100个文件夹，每隔文件夹可以分配255首曲目
- 30级音量可调，6级EQ可调

引脚说明

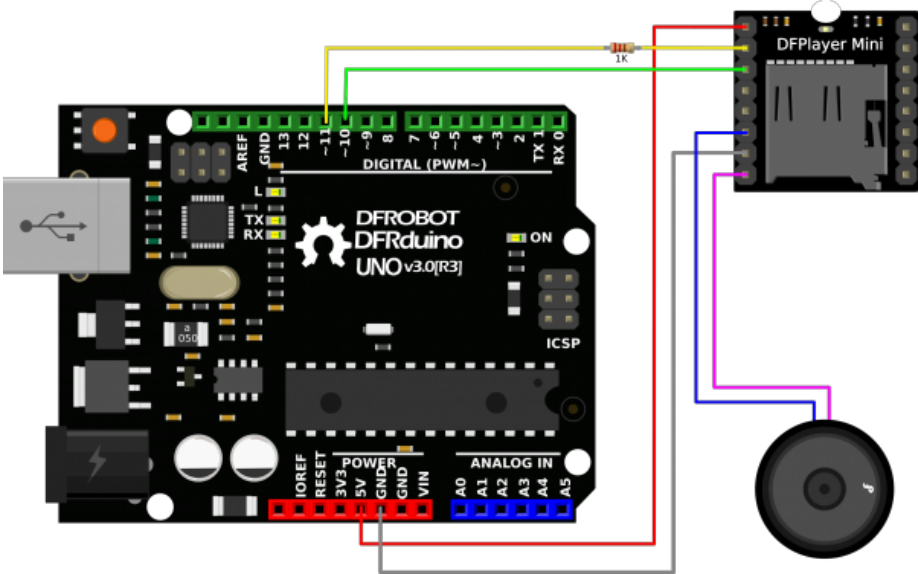


引脚号	引脚名称	功能描述	备注
1	VCC	模块电源输入	3.3V—5V，建议5V，不要超过5.2V
2	RX	UART串行数据输入	
3	TX	UART串行数据输出	
4	DAC_R	音频输出右声道	驱动耳机、功放
5	DAC_L	音频输出左声道	驱动耳机、功放
6	SPK2	接小喇叭	驱动小于3W喇叭
7	GND	地	电源地
8	SPK1	接小喇叭	驱动小于3W喇叭
9	IO1	触发口	默认上一曲（长按音量减）
10	GND	地	电源地
11	IO2	触发口	默认下一曲（长按音量加）
12	ADKEY1	AD口1	当触发时是第一首（长按循环第一首）
13	ADKEY2	AD口2	当触发时是第五首（长按循环第五首）

14	USB+	USB+ DP	接U盘或插电脑的USB口
15	USB-	USB- DM	接U盘或插电脑的USB口
16	Busy	播放状态	有音频，输出低；无音频，输出高

使用教程

连线图



样例代码

库文件下载链接：[DFRobotDFPlayerMini](#)

[库安装](#)

"GetStarted" 库样例, 每3秒切换下一首

```

/*****
DFPlayer - A Mini MP3 Player For Arduino
<https://www.dfrobot.com/index.php?route=product/product&product_id=1121>

*****
This example shows the basic function of library for DFPlayer.

Created 2016-12-07
By [Angelo qiao](Angelo.qiao@dfrobot.com)

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.Connection and Diagram can be found here
<https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
*****/

#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3.
    Serial.println(F("Unable to begin:"));
  }
}
```

```
        Serial.println(F("1.Please recheck the connection!"));
        Serial.println(F("2.Please insert the SD card!"));
        while(true);
    }
    Serial.println(F("DFPlayer Mini online.));

    myDFPlayer.volume(10); //Set volume value. From 0 to 30
    myDFPlayer.play(1); //Play the first mp3
}

void loop()
{
    static unsigned long timer = millis();

    if (millis() - timer > 3000) {
        timer = millis();
        myDFPlayer.next(); //Play next mp3 every 3 second.
    }

    if (myDFPlayer.available()) {
        printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message from DFPlayer
        to handle different errors and states.
    }
}

void printDetail(uint8_t type, int value){
    switch (type) {
        case TimeOut:
            Serial.println(F("Time Out!"));
            break;
        case WrongStack:
            Serial.println(F("Stack Wrong!"));
            break;
        case DFPlayerCardInserted:
            Serial.println(F("Card Inserted!"));
            break;
        case DFPlayerCardRemoved:
            Serial.println(F("Card Removed!"));
            break;
        case DFPlayerCardOnline:
            Serial.println(F("Card Online!"));
            break;
        case DFPlayerPlayFinished:
            Serial.print(F("Number:"));
            Serial.print(value);
            Serial.println(F(" Play Finished!"));
            break;
        case DFPlayerError:
            Serial.print(F("DFPlayerError:"));
            switch (value) {
                case Busy:
                    Serial.println(F("Card not found"));
                    break;
                case Sleeping:
                    Serial.println(F("Sleeping"));
                    break;
                case SerialWrongStack:
                    Serial.println(F("Get Wrong Stack"));
                    break;
                case CheckSumNotMatch:
                    Serial.println(F("Check Sum Not Match"));
                    break;
                case FileIndexOut:
                    Serial.println(F("File Index Out of Bound"));
                    break;
                case FileMismatch:
                    Serial.println(F("Cannot Find File"));
                    break;
                case Advertise:
                    Serial.println(F("In Advertise"));
                    break;
                default:
                    break;
            }
            break;
        default:
            break;
    }
}
```

"FullFunction" 库样例, 包含了所有的功能, 详见注释以及文档

```

/*****
DFPlayer - A Mini MP3 Player For Arduino
<https://www.dfrobot.com/index.php?route=product/product&product_id=1121>

****

This example shows the all the function of library for DFPlayer.

Created 2016-12-07
By [Angelo qiao](Angelo.qiao@dfrobot.com)

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.Connection and Diagram can be found here
<https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
*****/

#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
    mySoftwareSerial.begin(9600);
    Serial.begin(115200);

    Serial.println();
    Serial.println(F("DFRobot DFPlayer Mini Demo"));
    Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

    if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3.
        Serial.println(F("Unable to begin:"));
        Serial.println(F("1.Please recheck the connection!"));
        Serial.println(F("2.Please insert the SD card!"));
        while(true);
    }
    Serial.println(F("DFPlayer Mini online.));

    myDFPlayer.setTimeout(500); //Set serial communicaiton time out 500ms

    //----Set volume----
    myDFPlayer.volume(10); //Set volume value (0~30).
    myDFPlayer.volumeUp(); //Volume Up
    myDFPlayer.volumeDown(); //Volume Down

    //----Set different EQ----
    myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
    // myDFPlayer.EQ(DFPLAYER_EQ_POP);
    // myDFPlayer.EQ(DFPLAYER_EQ_ROCK);
    // myDFPlayer.EQ(DFPLAYER_EQ_JAZZ);
    // myDFPlayer.EQ(DFPLAYER_EQ_CLASSIC);
    // myDFPlayer.EQ(DFPLAYER_EQ_BASS);

    //----Set device we use SD as default----
    // myDFPlayer.outputDevice(DFPLAYER_DEVICE_U_DISK);
    myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);
    // myDFPlayer.outputDevice(DFPLAYER_DEVICE_AUX);
    // myDFPlayer.outputDevice(DFPLAYER_DEVICE_SLEEP);
    // myDFPlayer.outputDevice(DFPLAYER_DEVICE_FLASH);

    //----Mp3 control----
    // myDFPlayer.sleep(); //sleep
    // myDFPlayer.reset(); //Reset the module
    // myDFPlayer.enableDAC(); //Enable On-chip DAC
    // myDFPlayer.disableDAC(); //Disable On-chip DAC
    // myDFPlayer.outputSetting(true, 15); //output setting, enable the output and set the gain to 15

    //----Mp3 play----
    myDFPlayer.next(); //Play next mp3
    delay(1000);
    myDFPlayer.previous(); //Play previous mp3
    delay(1000);
    myDFPlayer.play(1); //Play the first mp3
    delay(1000);
    myDFPlayer.loop(1); //Loop the first mp3
    delay(1000);
    myDFPlayer.pause(); //pause the mp3

```

```

    delay(1000);
    myDFPlayer.start(); //start the mp3 from the pause
    delay(1000);
    myDFPlayer.playFolder(15, 4); //play specific mp3 in SD:/15/004.mp3; Folder Name(1~99); File
Name(1~255)
    delay(1000);
    myDFPlayer.enableLoopAll(); //loop all mp3 files.
    delay(1000);
    myDFPlayer.disableLoopAll(); //stop loop all mp3 files.
    delay(1000);
    myDFPlayer.playMp3Folder(4); //play specific mp3 in SD:/MP3/0004.mp3; File Name(0~65535)
    delay(1000);
    myDFPlayer.advertise(3); //advertise specific mp3 in SD:/ADVERT/0003.mp3; File Name(0~65535)
    delay(1000);
    myDFPlayer.stopAdvertise(); //stop advertise
    delay(1000);
    myDFPlayer.playLargeFolder(2, 999); //play specific mp3 in SD:/02/004.mp3; Folder Name(1~10);
File Name(1~1000)
    delay(1000);
    myDFPlayer.loopFolder(5); //loop all mp3 files in folder SD:/05.
    delay(1000);
    myDFPlayer.randomAll(); //Random play all the mp3.
    delay(1000);
    myDFPlayer.enableLoop(); //enable loop.
    delay(1000);
    myDFPlayer.disableLoop(); //disable loop.
    delay(1000);

    //----Read information----
    Serial.println(myDFPlayer.readState()); //read mp3 state
    Serial.println(myDFPlayer.readVolume()); //read current volume
    Serial.println(myDFPlayer.readEQ()); //read EQ setting
    Serial.println(myDFPlayer.readFileCounts()); //read all file counts in SD card
    Serial.println(myDFPlayer.readCurrentFileNumber()); //read current play file number
    Serial.println(myDFPlayer.readFileCountsInFolder(3)); //read fill counts in folder SD:/03
}

void loop()
{
    static unsigned long timer = millis();

    if (millis() - timer > 3000) {
        timer = millis();
        myDFPlayer.next(); //Play next mp3 every 3 second.
    }

    if (myDFPlayer.available()) {
        printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message from DFPlayer
to handle different errors and states.
    }
}

void printDetail(uint8_t type, int value){
    switch (type) {
        case TimeOut:
            Serial.println(F("Time Out!"));
            break;
        case WrongStack:
            Serial.println(F("Stack Wrong!"));
            break;
        case DFPlayerCardInserted:
            Serial.println(F("Card Inserted!"));
            break;
        case DFPlayerCardRemoved:
            Serial.println(F("Card Removed!"));
            break;
        case DFPlayerCardOnline:
            Serial.println(F("Card Online!"));
            break;
        case DFPlayerPlayFinished:
            Serial.print(F("Number:"));
            Serial.print(value);
            Serial.println(F(" Play Finished!"));
            break;
        case DFPlayerError:
            Serial.print(F("DFPlayerError:"));
            switch (value) {
                case Busy:
                    Serial.println(F("Card not found"));
                    break;
                case Sleeping:
                    Serial.println(F("Sleeping"));
                    break;
                case SerialWrongStack:
                    Serial.println(F("Get Wrong Stack"));

```

```
        break;
    case CheckSumNotMatch:
        Serial.println(F("Check Sum Not Match"));
        break;
    case FileIndexOut:
        Serial.println(F("File Index Out of Bound"));
        break;
    case FileMismatch:
        Serial.println(F("Cannot Find File"));
        break;
    case Advertise:
        Serial.println(F("In Advertise"));
        break;
    default:
        break;
    }
    break;
default:
    break;
}
```

注意：文件夹命名需要是mp3，放置在SD卡根目录下，而mp3文件命名需要是4位数字，例如“0001.mp3”，放置在mp3文件夹下。如需中英文命名，可以添加在数字后，例如“0001hello.mp3”或者“0001后来.mp3”。

资料说明

串口模式

模块支持异步串口通讯模式.通过串口接受控制命令

- 串口指令格式

指令名称	对应功能	功能描述
\$S	起始位0x7E	每条命令反馈均以\$开头，即0x7E
VER	版本	版本信息[目前默认为0xFF]
Len	len后字节个数	校验和不计算在内
CMD	命令字	表示具体的操作，比如播放/暂停等等
Feedback	命令反馈	是否需要反馈信息，1反馈，0不反馈
para1	参数1	查询的数据高字节（比如歌曲序号）
para2	参数2	查询的数据低字节
checksum	校验和[占两个字节]	累加和校验[不计起始位\$]
\$0	结束位	结束位0xEF

举个例子，如果我们制定播放NOR FLASH，就需要发送:7E FF 06 09 00 00 04 FF DD EF,数据长度为6，这6个字节分别是[FF 06 09 00 00 04]。不计算起始、结束、和校验。再然后对得到的结果进行取反。

- 串口控制指令

CMD命令(指令)	对应的功能	参数（16位）
0x01	下一曲	
0x02	上一曲	
0x03	指定曲目（NUM）	1-2999
0x04	音量+	
0x05	音量-	
0x06	指定音量	0-30
0x07	指定EQ 0/1/2/3/4/5	Normal/Pop/Rock/Jazz/Classic/Bass
0x08	单曲循环指定曲目播放	1-2999
0x09	指定播放设备 1/2/3/4/5	U盘/SD/AUX/SLEEP/FLASH
0x0A	进入休眠——低功耗	
0x0B	保留	
0x0C	模块复位	
0x0D	播放	
0x0E	暂停	
0x0F	指定文件夹播放	1-10（需要自己设定）
0x10	扩音设置（无）	[DH=1:开扩音][DL:设置增益0-31]
0x11	全部循环播放	[1:循环播放][0:停止循环播放]
0x12	指定MP3文件夹曲目	1-9999
0x13	插播广告	1-9999
0x14	支持15个文件夹	见下面的详细说明

0x15	停止播放，播放背景	
0x16	停止播放	

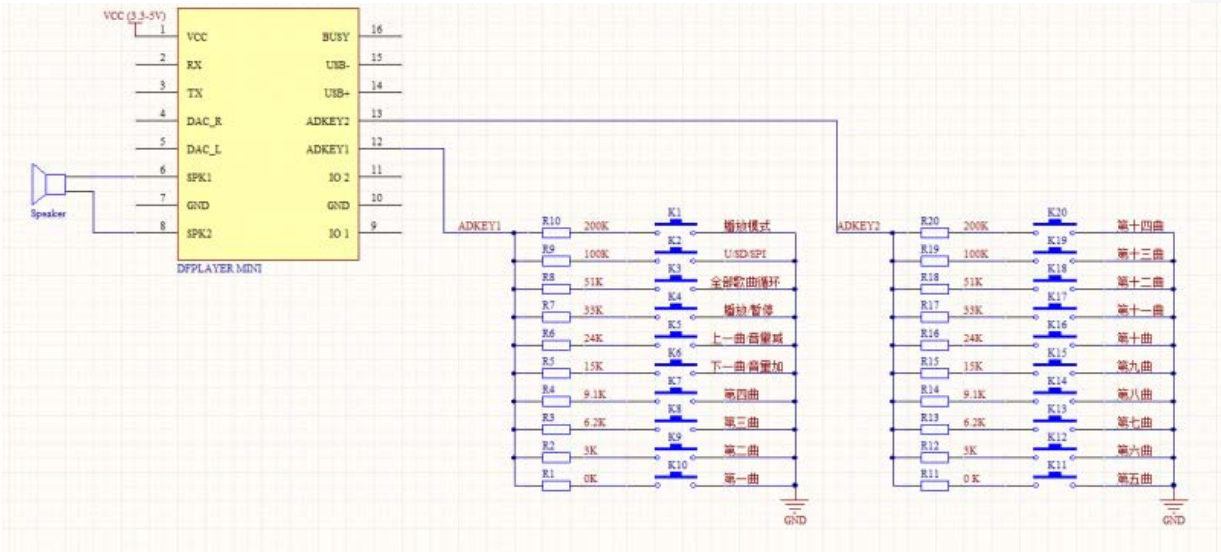
- 串口查询指令

CMD命令详解(指令)	对应的功能	参数（16位）
0x3C	STAY	
0x3D	STAY	
0x3E	STAY	
0x3F	发送初始化参数	0-0x0F（低四位每位代表一种设备）
0x40	返回错误，请求重发	
0x41	应答	
0x42	查询当前状态	
0x43	查询当前音量	
0x44	查询当前EQ	
0x45	查询当前播放模式	该版本保留此功能
0x46	查询当前软件版本	该版本保留此功能
0x47	查询UDISK的总文件数	
0x48	查询TF卡的总文件数	
0x49	查询FLASH卡的总文件数	
0x4A	保留	
0x4B	查询UDISK的当前曲目	
0x4C	查询TF的当前曲目	
0x4D	查询FLASH的当前曲目	

AD按键模式

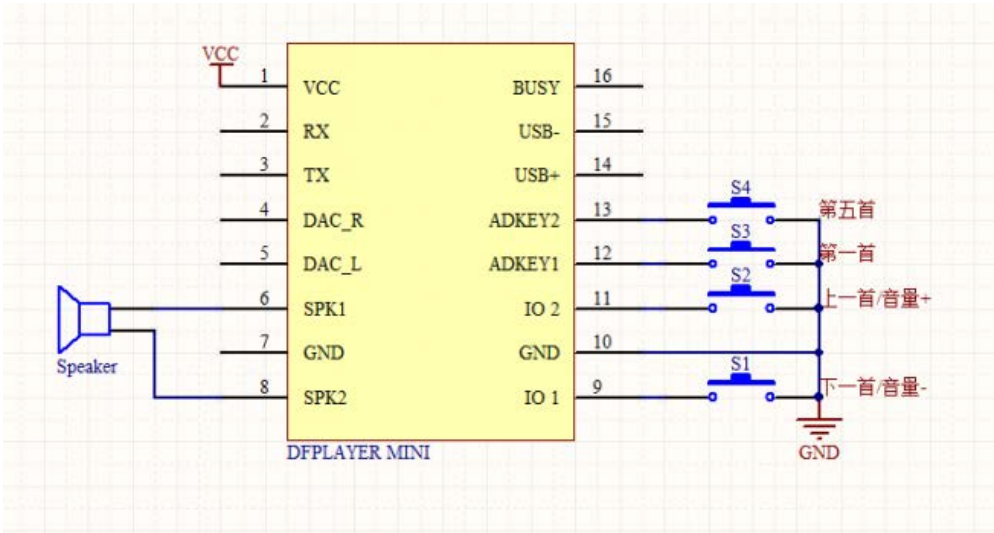
模块还支持AD按键控制方式，取代了传统了矩阵键盘的接法，这样做的好处是充分利用了MCU越来越强大的AD功能。设计简约而不简单，我们模块默认配置2个AD口，20个按键的阻值分配。

- 按键使用示意原理图



IO模式

本模块也支持简单的IO模式



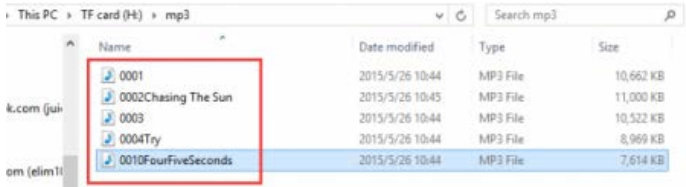
PS:短按：低电平时间约为200ms

疑难解答

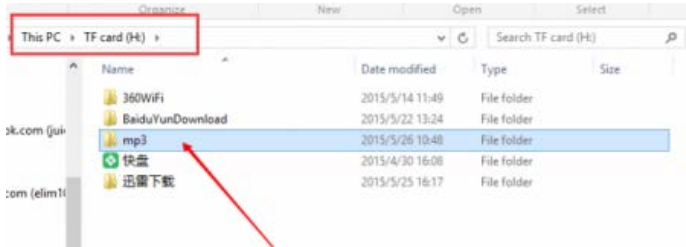
- 问：Arduino控制DFPlayer mini时，喇叭有杂音，如何处理？
- 答：在TX，RX与MP3模块的RX，TX之间连一个1k电阻。因为DFPlayer Mini模块工作电压应该是3.3V，而主控板传入电压为5V，因此需要1K左右电阻分压。
- 问：为什么 play(1) 播放的不是 0001.mp3, 而是其他的音乐? 为什么播放的顺序不是按照文件名播放?
- 答：play(1)是播放TF卡中的第一首歌, 播放顺序是由**mp3**放入**TF**卡的顺序决定的, 而不是按照文件名排序播放的, 如果需要按照文件名播放, 可以参考下面的playMp3Folder(1)
- 问：是否可以按照文件名播放?
- 答：可以使用函数**playMp3Folder (1)**; 播放文件的格式为"0001***.mp3(或支持其他格式)". 您可能需要注意这些:

(1). 该音频文件的名称应该命名一个四位数字开始，如：

- 0001.mp3
- 0002Chasing The Sun.mp3
- 0003.mp3
- 0004Try.mp3
- 0010FourFiveSeconds.mp3



(2).该音频文件应该放在“MP3”它坐落在TF卡的根目录文件夹;



- 问：想用下插播广告的功能，请问有样例吗？
- 答：推荐阅读【英文】——[如何实现插播广告的功能？](#)

更多问题及有趣的应用，可以 [访问论坛](#) 进行查阅或发帖！

更多

- [开发文档](#)
- [推荐阅读](#): [Arduino - 测试 DFPlayer Mini MP3 播放模组](#)

 [DFRobot商城购买链接](#)

3个分类: [Product Manual](#) | [DFR Series](#) | [Module](#)

本页面最后修改于2018年8月15日 (星期三) 13:54。

此页面已被浏览过53,333次。

[隐私政策](#) [关于DFRobot Product Wiki](#) [免责声明](#)

