

How to use DFPlayer mini to play MP3

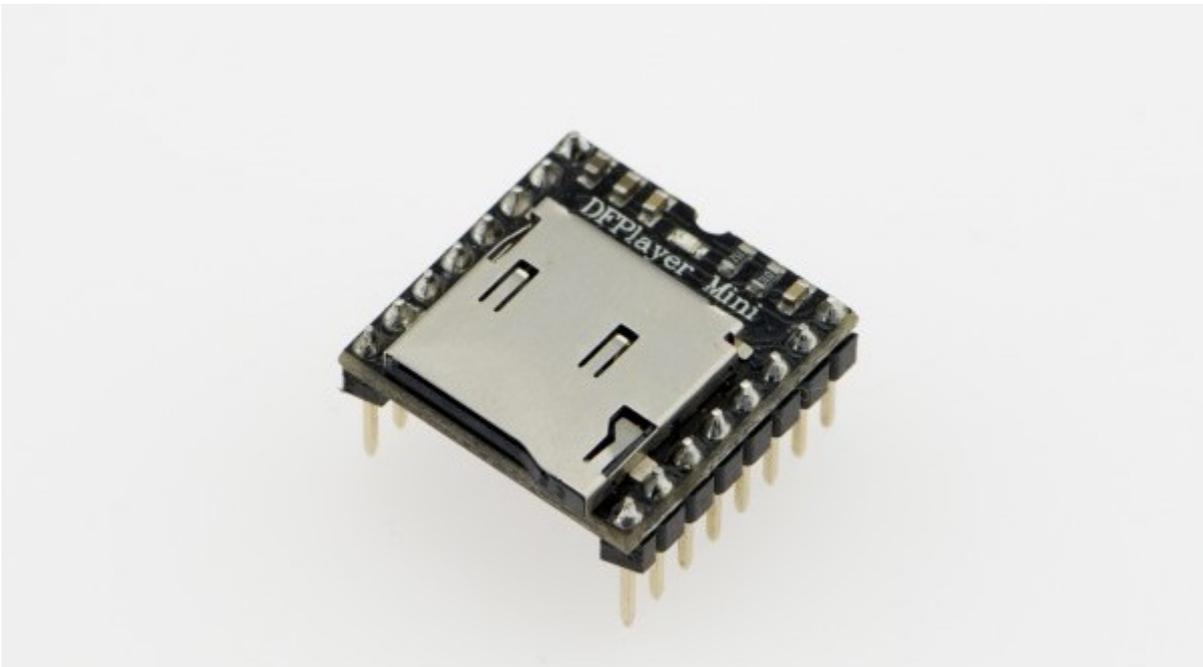
What is a DFPlayer?

11/2024 Update

A new library added below.

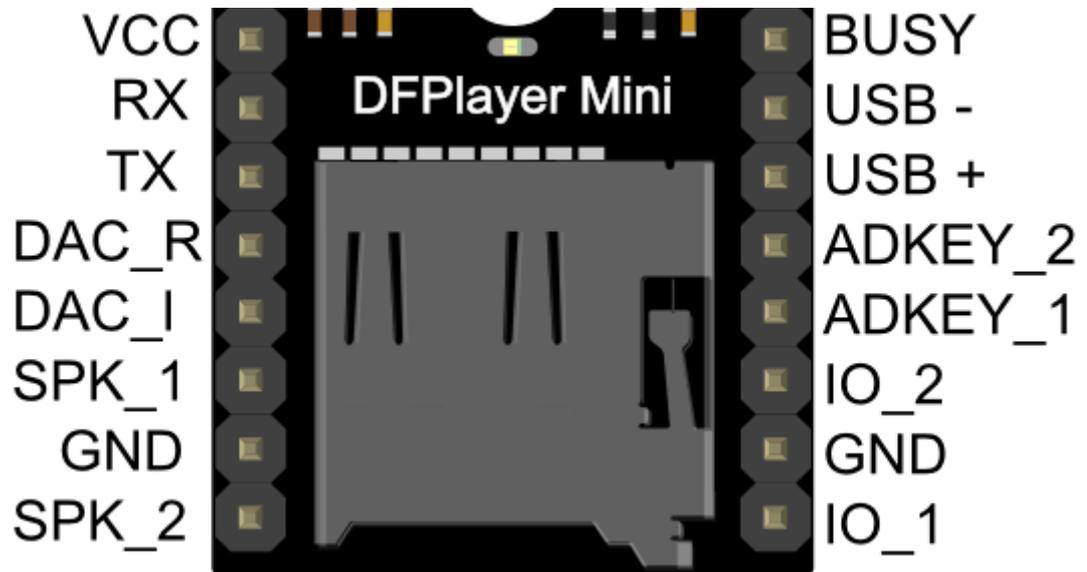
For people trying to avoid `delay()`, please use the `DFPlayerMini_Fast` library instead.

The DFPlayer Mini MP3 Player For Arduino is a small and low-priced MP3 module with a simplified output directly to the speaker. The module can be used as a stand-alone module with an attached battery, speaker and push buttons or used in combination with an Arduino UNO or any other with RX/TX capabilities. [Know More](#)



Wiring

Wiring t



pinout.

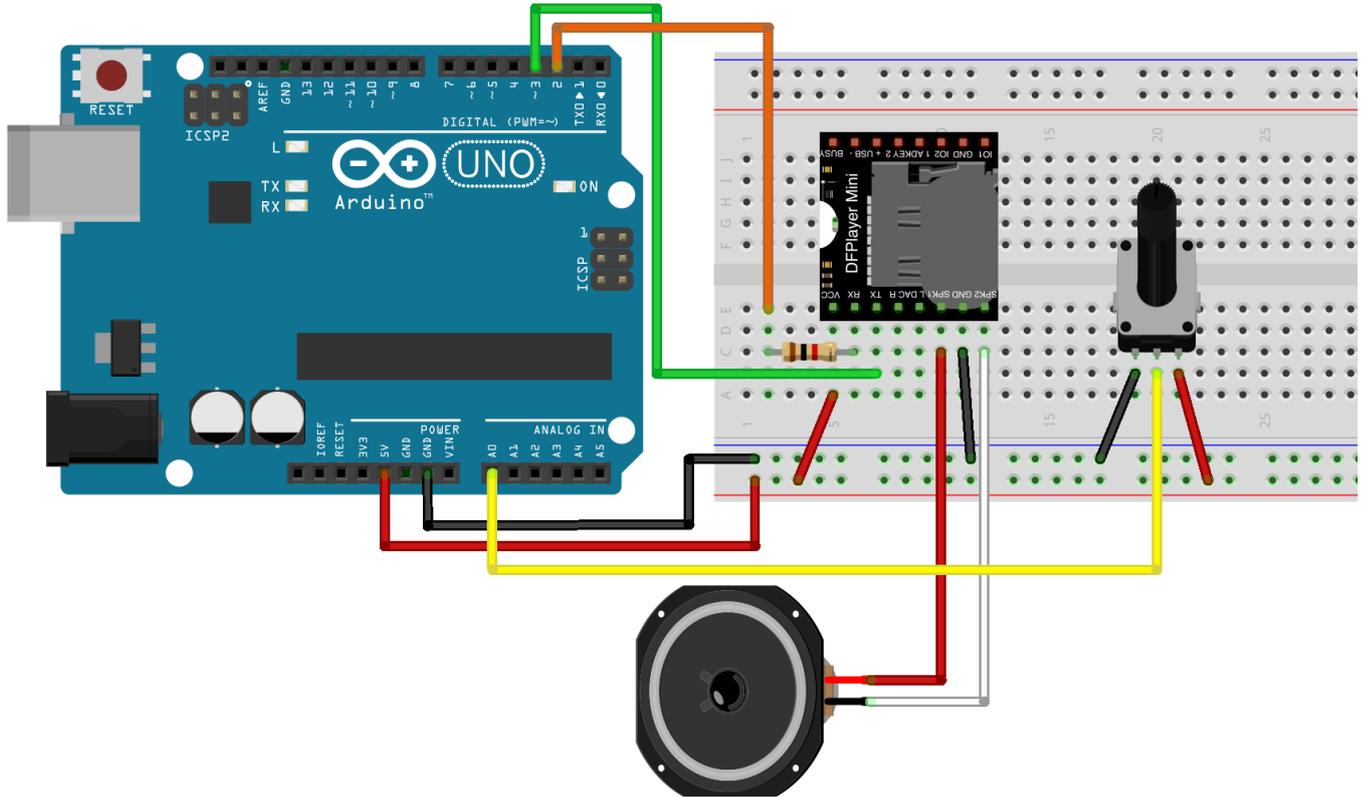
DFplayer Mini Wiring

1. VCC to 5V (Power)
2. RX to D2 via 1K resistor
3. TX to D3
4. SPK_1 to Speaker(+) *red wire*
5. GND to GND (Ground)
6. SPK_2 to Speaker(-) *black wire*

potentiometer Wiring

1. right pin to 5V (Power)
2. middle pin to A0 (Signal)

3. left pin to GND (Ground)



fritz

File handling

The order you copy the mp3 onto the micro SD card will affect the order mp3 played, which means the `play(1)` function will play the first mp3 copied into the micro SD card.

MAC User Attention!

If you are using Mac OS X to copy the mp3, the file system will automatically add hidden files like: `._0001.mp3` for index, which this module will handle as valid mp3 files.

It is really annoying. To remove them, follow the below steps:

1. Finder - Go to your USB drive
2. Press `Shift + Command + .` to reveal all hidden files
3. Select all `._xxxxxx` files and directories and delete
4. Empty Bin
5. Eject your USB drive

Library

`DFRobotDFPlayerMini` library will be used for this module. We have a tutorial on [how to install a library](#) here.

Get Started

In this example, we are using the potentiometer to control two audios. It will play the first audio when the potentiometer turns to the right and play the second when it turns to the left.

DF layer will not initiate!

If you didn't put in the SD card, or have no MP3 files in the SD card, the module will not work. Make sure you are using .mp3, not .wav or any other audio formats.

```
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

// Use pins 2 and 3 to communicate with DFPlayer Mini
static const uint8_t PIN_MP3_TX = 2; // Connects to module's RX
static const uint8_t PIN_MP3_RX = 3; // Connects to module's TX
SoftwareSerial softwareSerial(PIN_MP3_RX, PIN_MP3_TX);

const int pot = A0;
int potValue = 0;

// Create the Player object
DFRobotDFPlayerMini player;

void setup() {

  pinMode(pot, INPUT);

  // Init USB serial port for debugging
  Serial.begin(9600);
  // Init serial port for DFPlayer Mini
  softwareSerial.begin(9600);

  // Start communication with DFPlayer Mini
  if (player.begin(softwareSerial)) {
    Serial.println("OK");

    // Set volume to maximum (0 to 30).
    player.volume(30);
  } else {
```

```

Serial.println("Connecting to DFPlayer Mini failed!");
}
}

void loop() {

  potValue = analogRead(pot);

  if(potValue > 500 ){

    static unsigned long timer = millis();
    if (millis() - timer > 2000) { //2000 is the duration of the audio(1)
      timer = millis();

      //(2) is the 2rd file in the sd card, the order = the order you copied the file to it
      player.play(2);
    }

  }else {
    static unsigned long timer = millis();

    if (millis() - timer > 3000) { //3000 is the duration of the audio(2)
      timer = millis();
      player.play(1);
    }
  }
}
}

```

Better Library

Since the official library uses `delay()` in the code, it can be problematic when the code is used with other components or sensors.

DFRobotDFPlayerMini_Fast library will be used for this module. You will need to do a manual install for this library. `FireTimer` library is also needed.

We have a tutorial on [how to install a library](#) here.

Example code

The wiring will be the same as above. For further details of this library API, please visit their github page.

```
#include <DFPlayerMini_Fast.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(3, 2); // RX, TX

DFPlayerMini_Fast myMP3;

void setup()
{
  Serial.begin(115200);
  mySerial.begin(9600);
  myMP3.begin(mySerial, true);

  delay(1000);

  Serial.println("Setting volume to max");
  myMP3.volume(30);

  Serial.println("Looping track 1");
  myMP3.loop(1);
}

void loop()
{
  //do nothing
}
```

Revision #7

Created 24 November 2022 16:28:08 by Joanne Leung

Updated 16 December 2024 10:30:07 by Joanne Leung