

Grove - MP3 v3.0

Table of contents

Feature

Specification

Hardware Overview

Platforms Supported

Getting Started

 Play With Arduino

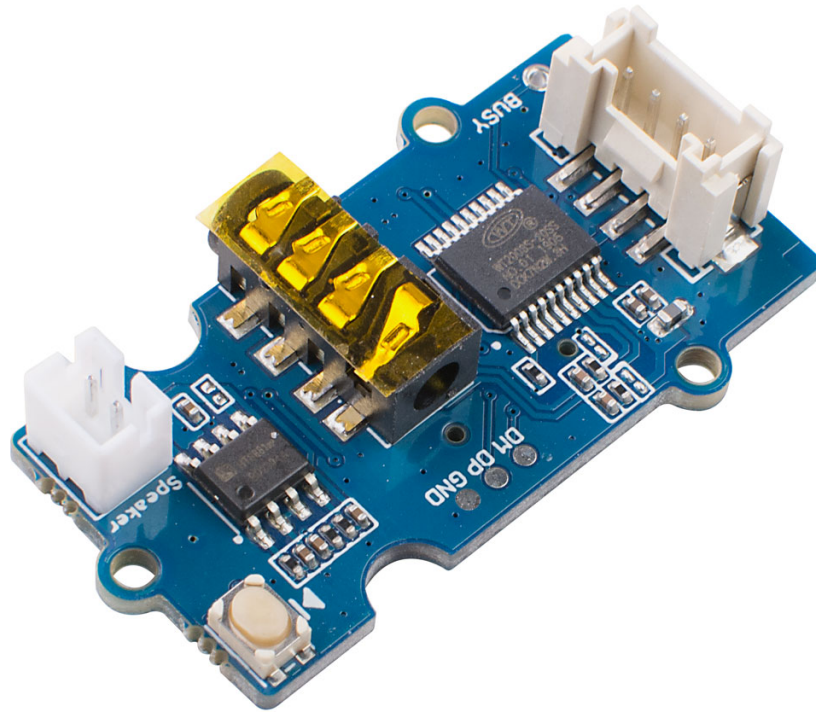
 Hardware Connection

 Software

FAQ

Resources

Tech Support



The Grove - MP3 is a 20x40mm super mini Music module based on WT2003S-20SS audio decoder. It supports high-quality MP3 format audio files with a sampling rate of 8~48KHz and a bit rate of 8~320Kbps. In order to expand the storage capacity, we added a TF card slot on the back of the module. TF card adopts DIO interface mode, supports up to 32GB, supports FAT16, FAT32 file system. Now with this little music module, you can carry hundreds and thousands of music in your pocket.

As the name indicates, the Grove - MP3 V3 is the upgraded version of Grove - MP3 V2.

Compared with Grove MP3 V2, the V3 added a JST2.0 speaker port, so that you can output the audio via speaker and 3.5mm earphone at the same time.

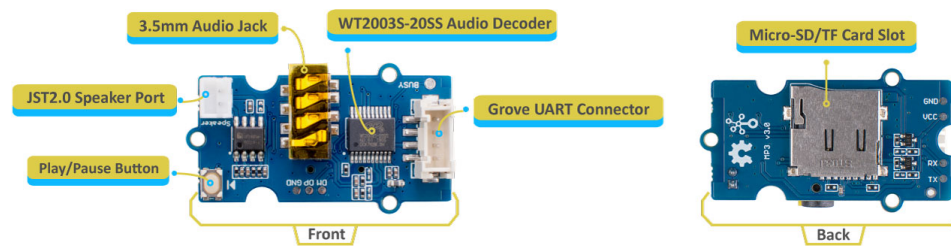
Feature

- Supports MP3 format audio files
- Sampling rate: 8~48KHz / bit rate: 8~320Kbps
- Support up to 32GB TF card
- Support speaker and earphone output audio at the same time
- Compatible with 3.3V and 5V platform.
- Support 32-level volume adjustment

Specification

Parameter	Value
Supply voltage	3.3V / 5V
Sampling rate	8~48KHz / bit rate: 8~320Kbps
Interface	I2C(Default I2C Address: 0x36) & Non-Changeable
Output	Speaker/3.5mm Audio Jack
Resolution	Support 32-level volume adjustment

Hardware Overview




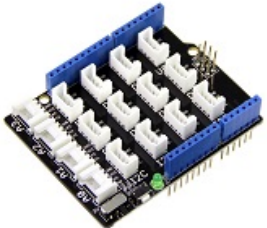

Platforms Supported

Arduino	Raspberry Pi	
		

Getting Started

Play With Arduino

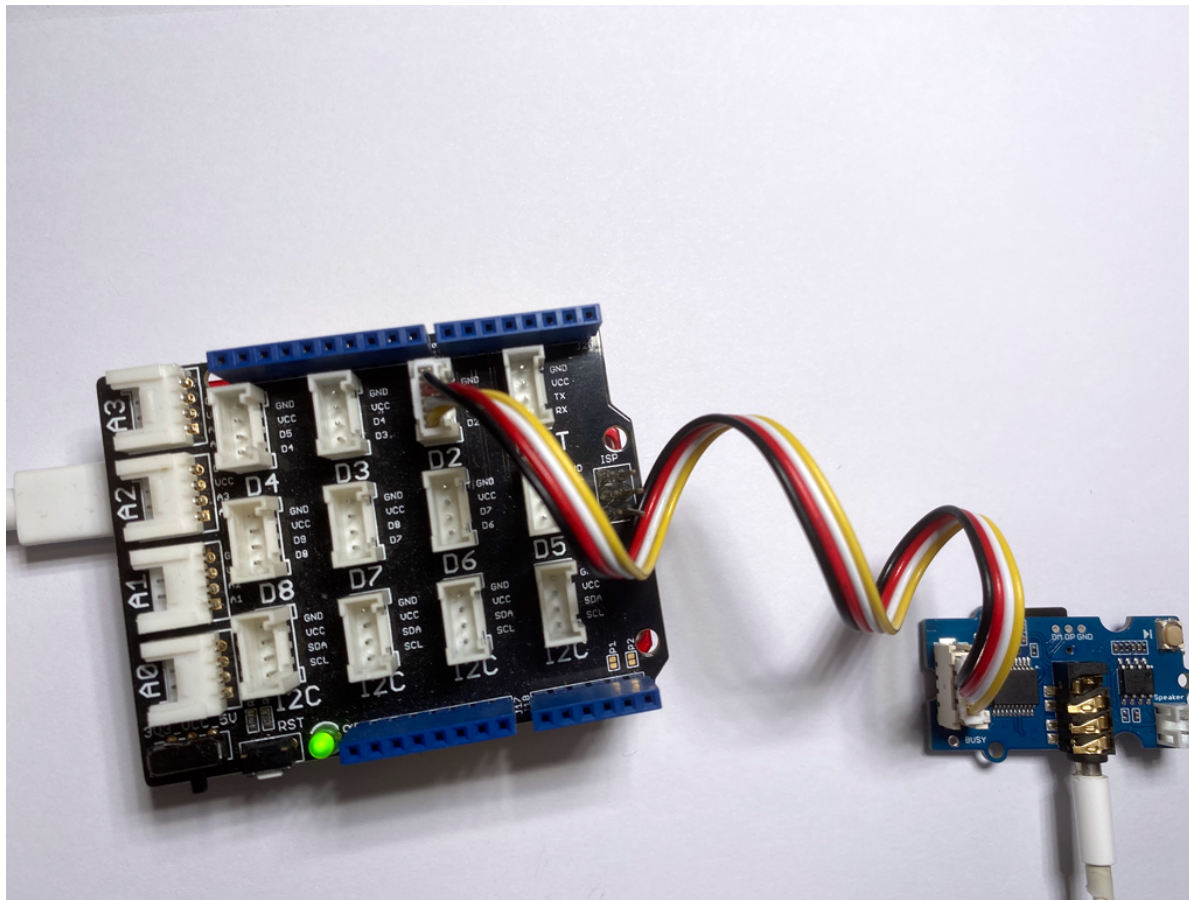
Materials required

Seeeduino V4.2	Base Shield	Grove - MP3 V3 -Music Player
		
Get ONE Now	Get ONE Now	Get ONE Now

In addition, you can consider our new [Seeeduino Lotus M0+](#), which is equivalent to the combination of Seeeduino V4.2 and Baseshield.

Hardware Connection

- **Step 1.** Connect the Grove - MP3 V3 Music Player to the **D2** port of the Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3** Connect the Seeeduino to PC via a USB cable.



Software



Attention

If this is the first time you work with Arduino, we strongly recommend you to see [Getting Started with Arduino](#) before the start.

- **Step 1.** Copy your `.mp3` music file to the tf card and save them in the root location in the tf card.
- **Step 2.** Download the [Sseed_Serial_MP3](#) Library from Github.



Note

Refer How to install library to [install library](#) for Arduino.

- **Step 3.** Restart the Arduino IDE. Open **WT2003S_Terminal_Player** example via the path: **File** → **Examples** → **Sseed_Serial_MP3_Player** → **WT2003S_Terminal_Player**. You can play `.mp3` format music file using this module, and use 3.5mm Audio Jack, Speaker via JST2.0 speaker port or even output both in the same time.

The WT2003S_Terminal_Player Example code is as follow:

```
1  #include "WT2003S_Player.h"
2
3  #ifdef __AVR__
4      #include <SoftwareSerial.h>
5      SoftwareSerial SSerial(2, 3); // RX, TX
6      #define COMSerial SSerial
7      #define ShowSerial Serial
8
9      WT2003S<SoftwareSerial> Mp3Player;
10 #endif
11
12 #ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
13     #define COMSerial Serial1
14     #define ShowSerial SerialUSB
15
16     WT2003S<Uart> Mp3Player;
```




```
17 #endif
18
19 #ifdef ARDUINO_ARCH_STM32F4
20     #define COMSerial Serial
21     #define ShowSerial SerialUSB
22
23     WT2003S<HardwareSerial> Mp3Player;
24 #endif
25
26
27 uint8_t vol = 10;
28 uint32_t spi_flash_songs = 0;
29 uint32_t sd_songs = 0;
30 STROAGE workdisk = SD;
31 struct Play_history {
32     uint8_t disk;
33     uint16_t index;
34     char name[8];
35 }* SPISong, *SDSong;
36
37 void readSongName(struct Play_history* ph, uint32_t num, STROAGE disk)
38     Mp3Player.volume(0);
39     delay(100);
40     switch (disk) {
41         case SPIFLASH:
42             Mp3Player.playSPIFlashSong(0x0001);
43             break;
44         case SD:
45             Mp3Player.playSDRootSong(0x0001);
46             break;
47         case UDISK:
48             Mp3Player.playUDiskRootSong(0x0001);
49             break;
50     }
51     ShowSerial.println("2...");
```

```

52     for (int i = 0; i < num ; i++) {
53         delay(300);
54         ph[i].disk = disk;
55         ph[i].index = Mp3Player.getTracks();
56         Mp3Player.getSongName(ph[i].name);
57         Mp3Player.next();
58     }
59     ShowSerial.println("4...");
60     Mp3Player.pause_or_play();
61     Mp3Player.volume(14);
62     delay(100);
63 }
64
65 void getAllSong() {
66     uint8_t diskstatus = Mp3Player.getDiskStatus();
67     ShowSerial.println(diskstatus);
68     spi_flash_songs = Mp3Player.getSPIFlashMp3FileNumber();
69     ShowSerial.print("SPIFlash:");
70     ShowSerial.println(spi_flash_songs);
71     if (spi_flash_songs > 0) {
72         SPISong = (struct Play_history*)malloc((spi_flash_songs + 1) *
73         readSongName(SPISong, spi_flash_songs, SPIFLASH);
74     }
75     if (diskstatus && 0x02) { // have SD
76         sd_songs = Mp3Player.getSDMp3FileNumber();
77         ShowSerial.print("SD:");
78         ShowSerial.println(sd_songs);
79         if (sd_songs > 0) {
80             SDSong = (struct Play_history*)malloc((sd_songs + 1) * size
81             ShowSerial.println("1...");
82             readSongName(SDSong, sd_songs, SD);
83         }
84     }
85 }
86 void printSongs() {

```

```

87     ShowSerial.print("-----");
88     ShowSerial.print("index");
89     ShowSerial.print("<----->");
90     ShowSerial.print("name");
91     ShowSerial.print("-----");
92     ShowSerial.println();
93     ShowSerial.println("-----spi flash-----");
94     for (int i = 0 ; i < spi_flash_songs; i++) {
95         ShowSerial.print("-----");
96         ShowSerial.print(SPISong[i].index);
97         ShowSerial.print("<----->");
98         ShowSerial.print(SPISong[i].name);
99         ShowSerial.print("-----");
100        ShowSerial.println();
101    }
102    ShowSerial.println("-----sd card-----");
103    for (int i = 0 ; i < sd_songs; i++) {
104        ShowSerial.print("-----");
105        ShowSerial.print(SDSong[i].index);
106        ShowSerial.print("<----->");
107        ShowSerial.print(SDSong[i].name);
108        ShowSerial.print("-----");
109        ShowSerial.println();
110    }
111 }
112
113 void setup() {
114     while (!ShowSerial);
115     ShowSerial.begin(9600);
116     COMSerial.begin(9600);
117     ShowSerial.println("+++++");
118     Mp3Player.init(COMSerial);
119
120     ShowSerial.println("0...");
121     getAllSong();

```

```
122     printMenu();
123     printSongs();
124 }
125
126 void loop() {
127     if (ShowSerial.available()) {
128         char cmd = ShowSerial.read();
129         switch (cmd) {
130             case '+': {
131                 ShowSerial.print("Volume up: ");
132                 vol = Mp3Player.getVolume();
133                 Mp3Player.volume(++vol);
134                 ShowSerial.print(vol);
135                 ShowSerial.println();
136                 break;
137             }
138             case '-': {
139                 ShowSerial.print("Volume down: ");
140                 vol = Mp3Player.getVolume();
141                 if (--vol > 31) {
142                     vol = 0;
143                 }
144                 Mp3Player.volume(vol);
145                 ShowSerial.print(vol);
146                 ShowSerial.println();
147                 break;
148             }
149             case 't': {
150                 uint8_t status;
151                 ShowSerial.print("status:");
152                 status = Mp3Player.getStatus();
153                 if (status == 0x01) {
154                     ShowSerial.print("playing");
155                 }
156                 if (status == 0x02) {
```

```
157         ShowSerial.print("stop");
158     }
159     if (status == 0x03) {
160         ShowSerial.print("pause");
161     }
162     ShowSerial.println();
163     break;
164 }
165 case 'n': {
166     Mp3Player.next();
167     break;
168 }
169 case 'p': {
170     Mp3Player.pause_or_play();
171     break;
172 }
173 case 'w': {
174     Mp3Player.playMode(SINGLE_SHOT);
175     break;
176 }
177 case 'x': {
178     Mp3Player.playMode(SINGLE_CYCLE);
179     break;
180 }
181 case 'y': {
182     Mp3Player.playMode(CYCLE);
183     break;
184 }
185 case 'z': {
186     Mp3Player.playMode(RANDOM);
187     break;
188 }
189 case 'c': {
190     ShowSerial.print(Mp3Player.copySDtoSPIFlash());
191     break;
```

```

192     }
193     case '1':
194     case '2':
195     case '3':
196     case '4':
197     case '5':
198     case '6':
199     case '7':
200     case '8':
201     case '9':
202         ShowSerial.print("play:");
203         if (workdisk == SD) {
204             Mp3Player.playSDRootSong(cmd - '0' - 1);
205             ShowSerial.print(cmd + ": ");
206             ShowSerial.print(SDSong[cmd - '0'].name);
207         }
208         if (workdisk == SPIFLASH) {
209             Mp3Player.playSPIFlashSong(cmd - '0' - 1);
210             ShowSerial.print(cmd + ": ");
211             ShowSerial.print(SPISong[cmd - '0'].name);
212         }
213         ShowSerial.println();
214         break;
215     default:
216         break;
217     }
218 }
219 }
220
221 void printMenu(void) {
222     ShowSerial.println("MP3 Command List:");
223     ShowSerial.println("-----");
224     ShowSerial.println("'+' or '-' : raise/lower volume");
225     ShowSerial.println("'1' ~ '9' : select a song");
226     ShowSerial.println("'n' : next song");

```

```
227 ShowSerial.println("'s'      : switch play disk, spi flash");
228 ShowSerial.println("'p'      : play or pause");
229 ShowSerial.println("'w'      : set playmode single no loop");
230 ShowSerial.println("'x'      : set playmode single loop");
231 ShowSerial.println("'y'      : set playmode all loop");
232 ShowSerial.println("'z'      : set playmode random");
233 ShowSerial.println("'c'      : Copy mp3 to SPIFlash");
234 ShowSerial.println("          (Yes, this really does go by copy
235 ShowSerial.println();
236 ShowSerial.println("Any other key to show this menu");
237 ShowSerial.println();
238 }
```

- **Step 3.** Upload the demo. If you do not know how to upload the code, please check [How to upload code](#).
- **Step 4.** Open the **Serial Monitor** of Arduino IDE by click **Tool-> Serial Monitor**. Or tap the `Ctrl` + `Shift` + `M` key at the same time. Set the baud rate to **9600**.
- **Step 5.** The result should look like below. Follow the command list to play music or use other useful features.

```
1
+++++
0...
2
SPIFlash:0
SD:13
1...
2...
4...
MP3 Command List:
-----
'+' or '-' : raise/lower volume
'1' ~ '9' : select a song
'n'       : next song
's'       : switch play disk, spi flash
'p'       : play or pause
'w'       : set playmode single no loop
'x'       : set playmode single loop
'y'       : set playmode all loop
'z'       : set playmode random
'c'       : Copy mp3 to SPIFlash
           (Yes, this really does go by copy order.)

Any other key to show this menu

-----index<----->name-----
-----spi flash-----
-----sd card-----
-----1<----->0001-----
-----2<----->0001-----
-----3<----->0002-----
```

☒ 自动滚屏 ☐ Show timestamp 换行符 9600 波特率 清空输出

FAQ

Q1# TF card cannot be recognized.

A1: Check the file system of the TF card, make sure it is FAT16 or FAT32 file system.

Resources

- **[ZIP]** [Grove - MP3 V3 Schematic](#)
- **[PDF]** [WT2003S Datasheet](#)

Tech Support

Please submit any technical issue into our [forum](#)
