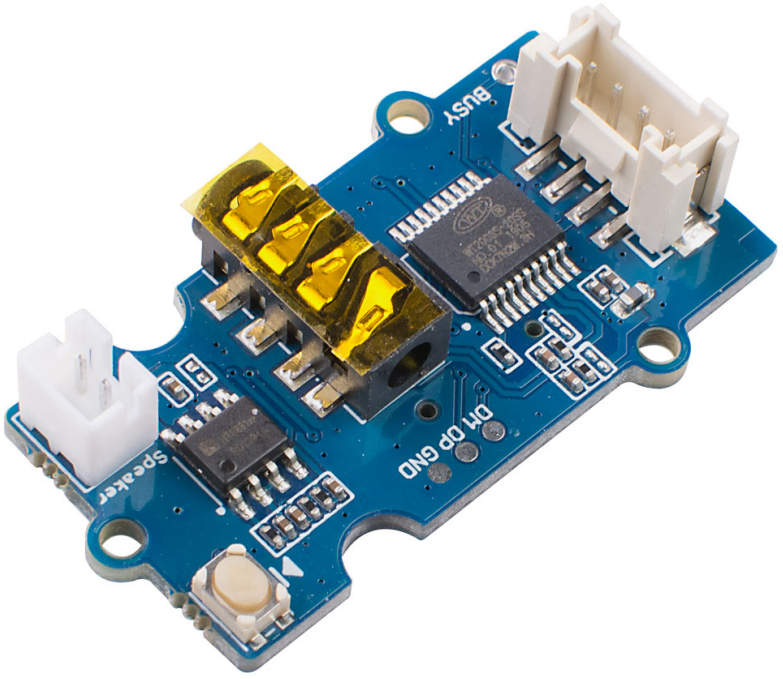
Grove - MP3 v3.0



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.

[https://www.seeedstudio.com/Grove-MP3-V3-p-4297.html]

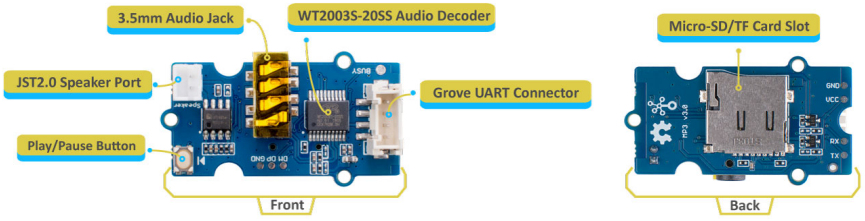
Feature

- 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



[https://files.seeedstudio.com/wiki/Grove-MP3-V3/img/hardware.jpg]

Platforms Supported

Arduino	Raspberry Pi		
	TO DO		

Getting Started

Play With Arduino

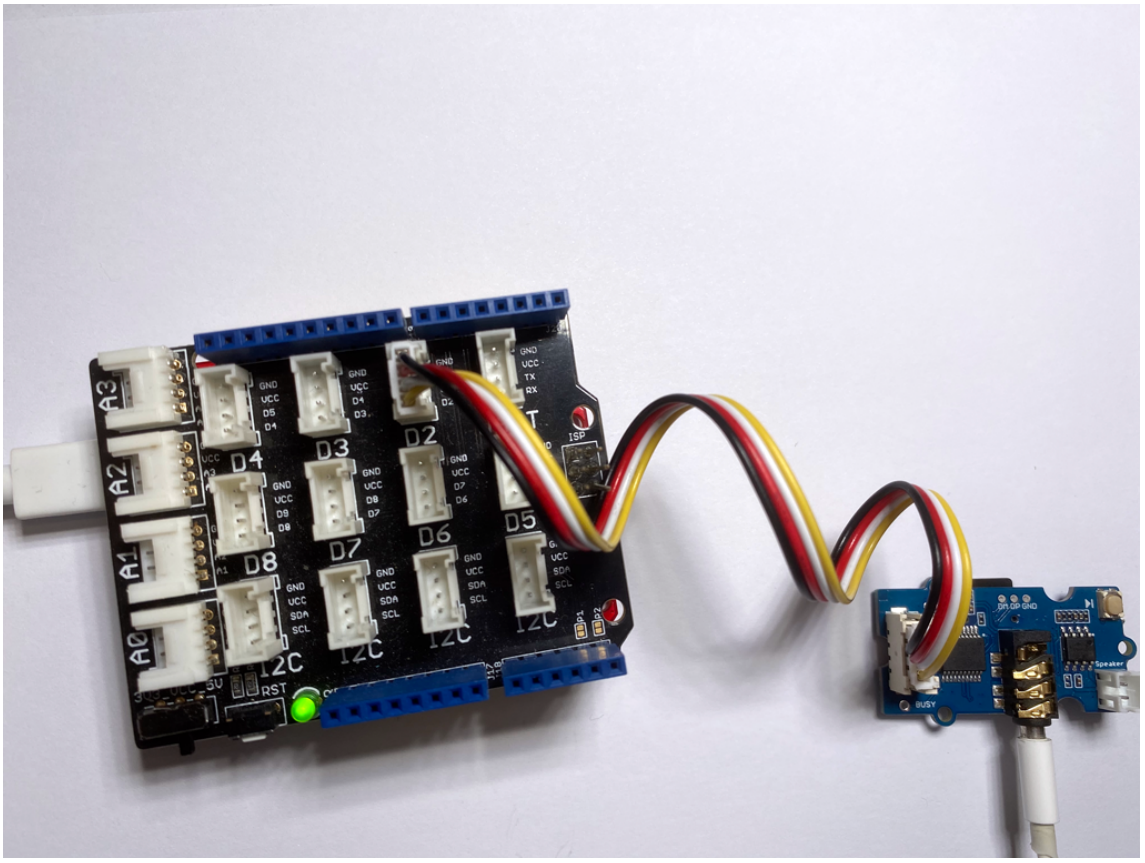
Materials required

Seeeduino V4.2	Base Shield
	
Get ONE Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get ONE Now [https://www.seeedstudio.com/Base Shield-V2-p-1378.html]

> In addition, you can consider our new Seeeduino Lotus M0+ [https://www.seeedstudio.com/Seeeduino-Lotus-Cortex-M0-p-2896.html], 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 [https://wiki.seeedstudio.com/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 Seeed_Serial_MP3 [https://github.com/Seeed-Studio/Seeed_Serial_MP3_Player] Library from Github.

> **Note**
> Refer How to install library to install library [https://wiki.seeedstudio.com/How_to_install_Arduino_Library/] for Arduino.

- **Step 3.** Restart the Arduino IDE. Open **WT2003S_Terminal_Player** example via the path: **File** → **Examples** → **Seeed_Serial_MP3_Player** → **WT2003S_Terminal_Player**. You can play `.mp3` format music file using this moudle, 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:

```
#include "WT2003S_Player.h"

#ifdef __AVR__
    #include <SoftwareSerial.h>
    SoftwareSerial SSerial(2, 3); // RX, TX
    #define COMSerial SSerial
    #define ShowSerial Serial

```

```
    WT2003S<SoftwareSerial> Mp3Player;
#endif

#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
    #define COMSerial Serial1
    #define ShowSerial SerialUSB

    WT2003S<Uart> Mp3Player;
#endif

#ifdef ARDUINO_ARCH_STM32F4
    #define COMSerial Serial
    #define ShowSerial SerialUSB

    WT2003S<HardwareSerial> Mp3Player;
#endif


uint8_t vol = 10;
uint32_t spi_flash_songs = 0;
uint32_t sd_songs = 0;
STROAGE workdisk = SD;
struct Play_history {
    uint8_t disk;
    uint16_t index;
    char name[8];
}* SPISong, *SDSong;

void readSongName(struct Play_history* ph, uint32_t num
    Mp3Player.volume(0);
    delay(100);
    switch (disk) {
        case SPIFLASH:
            Mp3Player.playSPIFlashSong(0x0001);
            break;
        case SD:
            Mp3Player.playSDRootSong(0x0001);
            break;
        case UDISK:
            Mp3Player.playUDiskRootSong(0x0001);
            break;
```

```
    }
    ShowSerial.println("2...");
    for (int i = 0; i < num ; i++) {
        delay(300);
        ph[i].disk = disk;
        ph[i].index = Mp3Player.getTracks();
        Mp3Player.getSongName(ph[i].name);
        Mp3Player.next();
    }
    ShowSerial.println("4...");
    Mp3Player.pause_or_play();
    Mp3Player.volume(14);
    delay(100);
}

void getAllSong() {
    uint8_t diskstatus = Mp3Player.getDiskStatus();
    ShowSerial.println(diskstatus);
    spi_flash_songs = Mp3Player.getSPIFlashMp3FileNumbe
    ShowSerial.print("SPIFlash:");
    ShowSerial.println(spi_flash_songs);
    if (spi_flash_songs > 0) {
        SPISong = (struct Play_history*)malloc((spi_fla
        readSongName(SPISong, spi_flash_songs, SPIFLASH
    }
    if (diskstatus && 0x02) { // have SD
        sd_songs = Mp3Player.getSDMp3FileNumber();
        ShowSerial.print("SD:");
        ShowSerial.println(sd_songs);
        if (sd_songs > 0) {
            SDSong = (struct Play_history*)malloc((sd_s
            ShowSerial.println("1...");
            readSongName(SDSong, sd_songs, SD);
        }
    }
}
void printSongs() {
    ShowSerial.print("--------------------");
    ShowSerial.print("index");
    ShowSerial.print("<-------->");
    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: ");
```

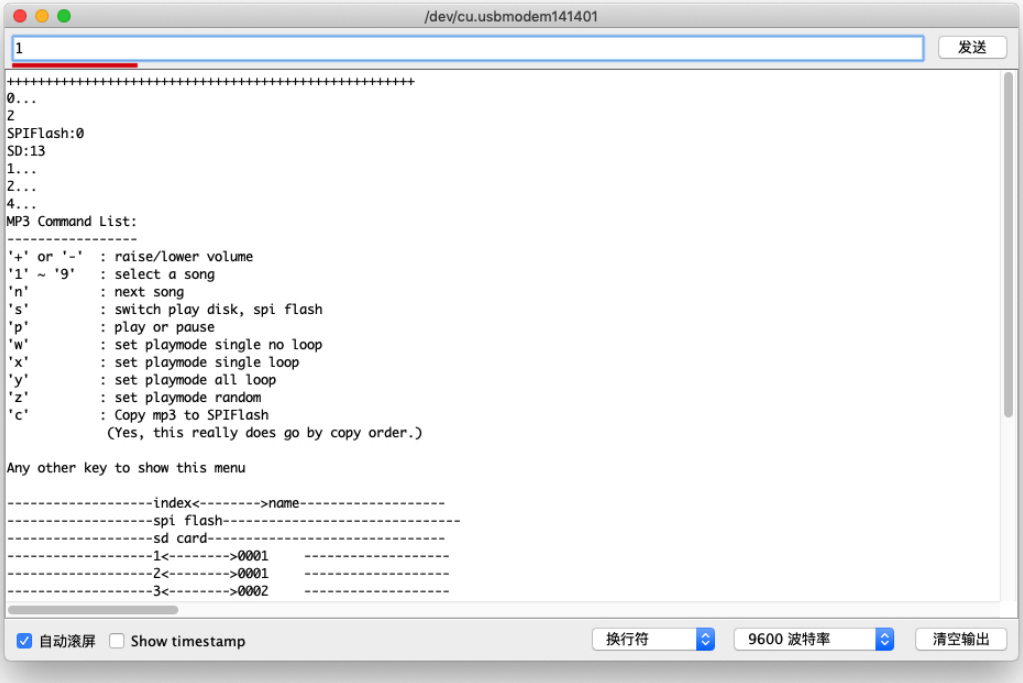
```
                    vol = Mp3Player.getVolume();
                    Mp3Player.volume(++vol);
                    ShowSerial.print(vol);
                    ShowSerial.println();
                    break;
                }
            case '-': {
                    ShowSerial.print("Volume down: ");
                    vol = Mp3Player.getVolume();
                    if (--vol > 31) {
                        vol = 0;
                    }
                    Mp3Player.volume(vol);
                    ShowSerial.print(vol);
                    ShowSerial.println();
                    break;
                }
            case 't': {
                    uint8_t status;
                    ShowSerial.print("status:");
                    status = Mp3Player.getStatus();
                    if (status == 0x01) {
                        ShowSerial.print("playing");
                    }
                    if (status == 0x02) {
                        ShowSerial.print("stop");
                    }
                    if (status == 0x03) {
                        ShowSerial.print("pause");
                    }
                    ShowSerial.println();
                    break;
                }
            case 'n': {
                    Mp3Player.next();
                    break;
                }
            case 'p': {
                    Mp3Player.pause_or_play();
                    break;
                }
```

```
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.copySDto
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'
205                     ShowSerial.print(cmd + ": ");
206                     ShowSerial.print(SDSong[cmd - '0'].
207                 }
208                 if (workdisk == SPIFLASH) {
209                     Mp3Player.playSPIFlashSong(cmd - '0
210                     ShowSerial.print(cmd + ": ");
211                     ShowSerial.print(SPISong[cmd - '0']
212                 }
213                 ShowSerial.println();
```

```
                break;
            default:
                break;
        }
    }
}

void printMenu(void) {
    ShowSerial.println("MP3 Command List:");
    ShowSerial.println("-----------------");
    ShowSerial.println("'+' or '-'  : raise/lower volum
    ShowSerial.println("'1' ~ '9'   : select a song");
    ShowSerial.println("'n'         : next song");
    ShowSerial.println("'s'         : switch play disk,
    ShowSerial.println("'p'         : play or pause");
    ShowSerial.println("'w'         : set playmode sing
    ShowSerial.println("'x'         : set playmode sing
    ShowSerial.println("'y'         : set playmode all
    ShowSerial.println("'z'         : set playmode rand
    ShowSerial.println("'c'         : Copy mp3 to SPIFl
    ShowSerial.println("             (Yes, this really
    ShowSerial.println();
    ShowSerial.println("Any other key to show this menu
    ShowSerial.println();
}
```

- **Step 3.** Upload the demo. If you do not know how to upload the code, please check How to upload code [https://wiki.seeedstudio.com/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.



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 [https://files.seeedstudio.com/wiki/Grove-MP3-V3/res/Grove-MP3.zip]
- **[PDF]** WT2003S Datasheet [https://files.seeedstudio.com/wiki/Grove-MP3-V3/res/Grove-MP3.zip]

Tech Support

Please submit any technical issue into our forum [https://forum.seeedstudio.com/]

[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]