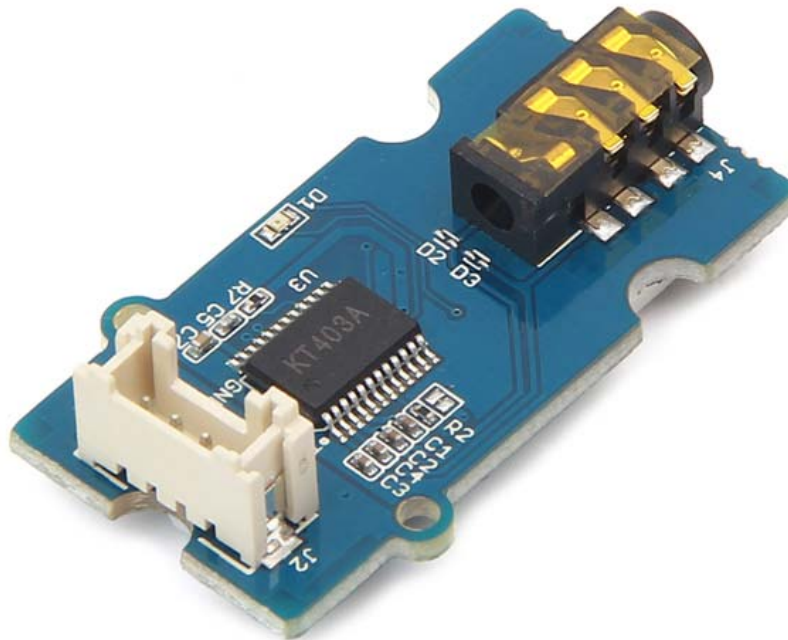# Grove - MP3 v2.0

Grove - MP3 v2.0 is a tiny-sized and compact audio module. It supports various audio file operations for an audio file of MP3, WAV and WMV format, such as random music playing, play music in specific files, and so on. With serial communication, you can use all predefined command or command combinations to do all operations on music files. This module also supports general file systems such as FAT16 and FAT32. It gets a Grove UART interface, a 3.5 mm audio jack and a micro-SD slot. With this module, you can add some noise to your silent applications.

Version

Product Version	Changes	Released Date
Grove - MP3 v1.0	Initial	April 28 2013
Grove - MP3 v2.0	Change to KT403A	Dec 15 2015

Features

- General operations on audio files
- On-board micro-SD slot and 3.5 mm audio jack
- Support sample rate of 8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48(KHz)
- 24-bit DAC output, 90 dB (at Max.) dynamic output range, signal-noise ratio at 85 dB
- MP3, WMV and WAV audio format and FAT16, FAT32 files system supported
- Embed 10 levels of equalization in total

> **Tip**
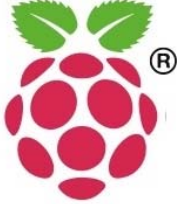> More details about Grove modules please refer to Grove System

Application ideas

- Middle-level audio module for any applications.

Specifications

Parameter	Value
Input	5 V (DC)
Operating current (no signal output state)	less than 15 mA
Operating current	less than 40 mA
Chip	KT403A (datasheet)
Chip LDO output voltage	3.3 V
Chip output current	100mA(at Max.)
File formats supported	MP3, WAV, WMA
Maximum memory supported for SD card	8 GB
Sampling rate	8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48(KHz)

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE

Caution
The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.
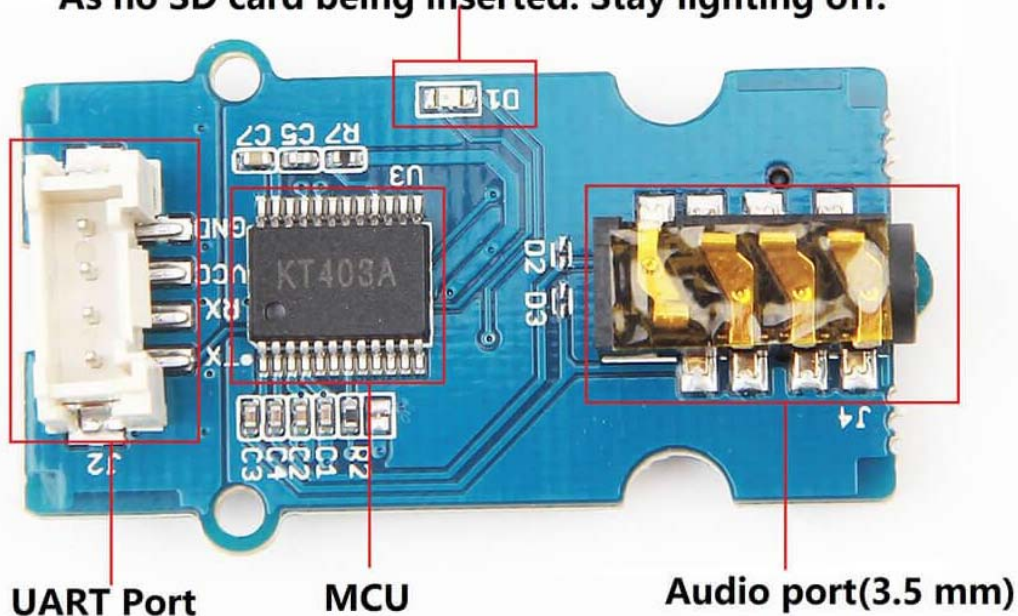
Hardware Overview

Status indicator

As SD card being inserted: Blink for playing music;

Light off at standy status; Stay lighting on for other situations;

As no SD card being inserted: Stay lighting off.
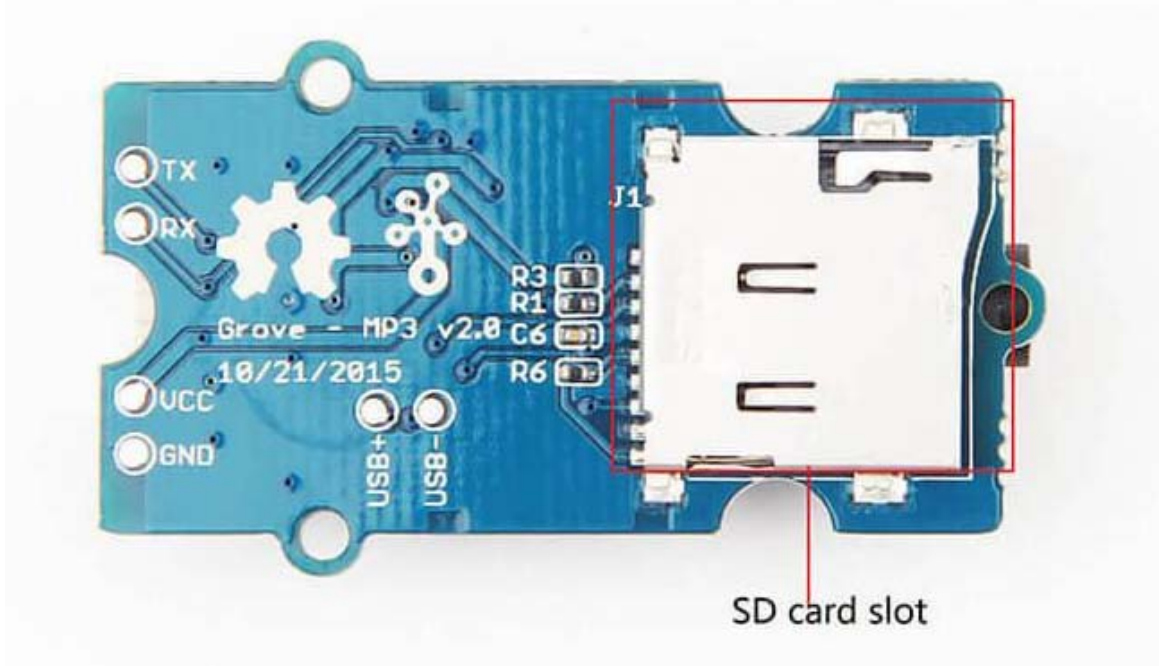
UART Port   MCU   Audio port(3.5 mm)

SD card slot

Getting Started

Play With Arduino

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield	Grove - MP3 v2.0

Note
We need SD card with music inside and Headset/earphone or stereo with 3.5 mm audio jack as well.

- **Step 2.** Connect Grove-MP3 v2.0 to port D2 of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.
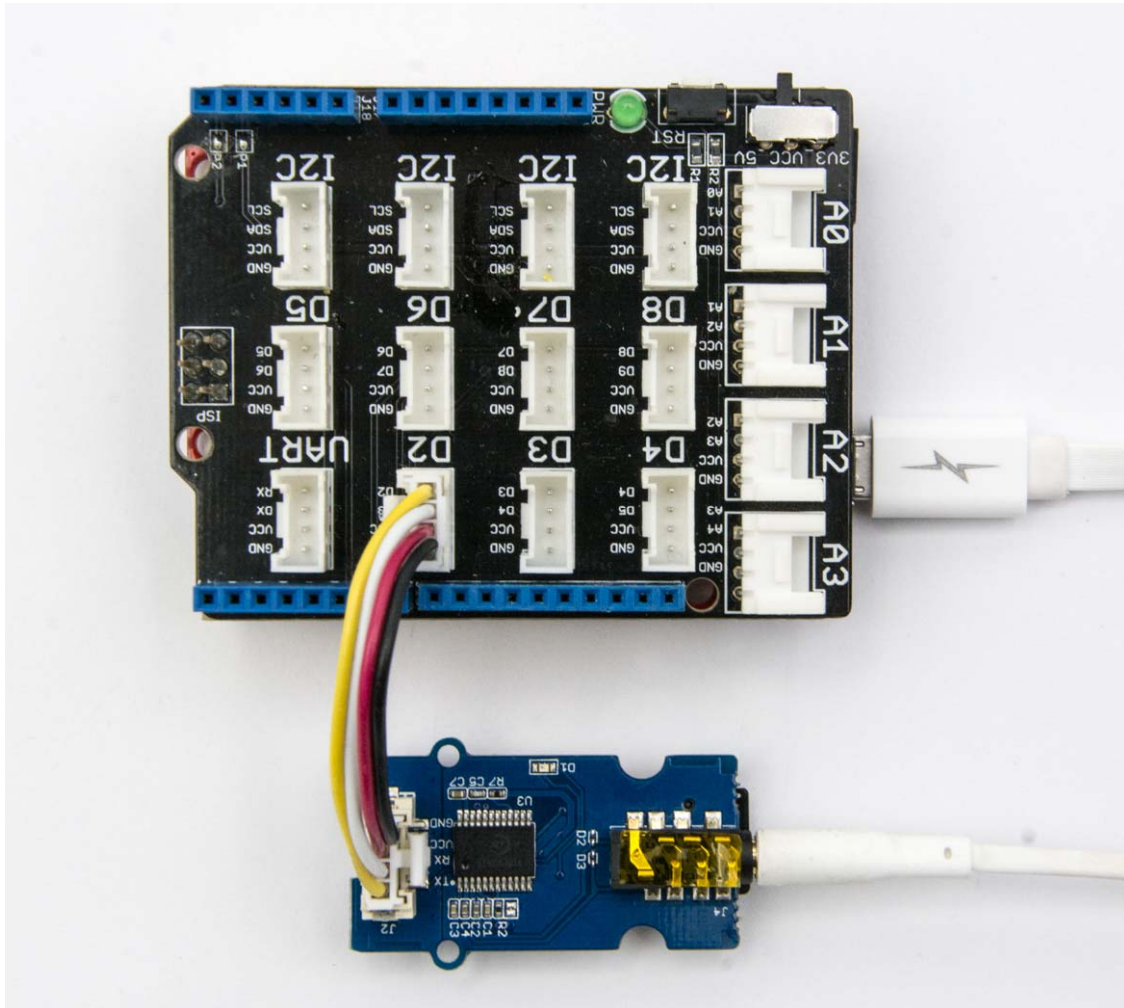
Note
If we don't have Grove Base Shield, We also can directly connect Grove-MP3 v2.0 to Seeeduino as below.

Seeeduino	Grove-MP3 v2.0
5V	Red
GND	Black
D3	White
D2	Yellow

Software

- **Step 1.** Download the Grove-MP3 v2.0 from Github.
- **Step 2.** Refer How to install library to install library for Arduino.
- **Step 3.** Copy the code into Arduino IDE and upload. If you do not know how to upload the code, please check how to upload code.

Here is the code.

```
1/*
2 * MP3_Play_Test.ino
3 * A quick start example for Grove-Serial MP3 Player V2.0
4 * Note: The MP3 chip of Grove-Serial MP3 Player V2.0 is different from Grove-Serial MP3 Player
5V1.0
6 * Description: This demo let you can send instruction 1-8 to control the Grove-Serial MP3 Player, via
7the serial port.
8 *
9 * Copyright (c) 2015 seeed technology inc.
10 * Author     : Wuruibin
11 * Created Time: Dec 2015
12 * Modified Time:
13 *
14 * The MIT License (MIT)
15 *
16 * Permission is hereby granted, free of charge, to any person obtaining a copy
17 * of this software and associated documentation files (the "Software"), to deal
18 * in the Software without restriction, including without limitation the rights
19 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
20 * copies of the Software, and to permit persons to whom the Software is
21 * furnished to do so, subject to the following conditions:
22 *
23 * The above copyright notice and this permission notice shall be included in
24 * all copies or substantial portions of the Software.
25 *
26 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
27 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
28 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
29THE
30 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
31 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
32FROM,
33 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
34 * THE SOFTWARE.
35 */
36
37
38#include <SoftwareSerial.h>
39#include <MP3Player_KT403A.h>
40
41static uint8_t recv_cmd[8] = {};
42
43
44// Note: You must define a SoftwareSerial class object that the name must be mp3,
45//      but you can change the pin number according to the actual situation.
```

```
46  SoftwareSerial mp3(2, 3);
47
48  void setup()
49  {
50      mp3.begin(9600);
51      Serial.begin(9600);
52
53      while(!Serial);
54
55      Serial.println("Grove - Serial MP3 Demo");
56      Serial.println(
57          "Input command:\r\n\r\n"
58          "P[ ] play music by default index\r\n"
59          "Pm[ ] play music in MP3 folder by index\r\n"
60          "Pf[ ][ ] play music by specify folder and index\r\n"
61          "p Pause\r\n"
62          "R Resume\r\n"
63          "N Next\r\n"
64          "L Previous\r\n"
65          "l Loop\r\n"
66          "I Increase volume\r\n"
67          "D Decrease volumern\r\n");
68
69      delay(100);
70
71      SelectPlayerDevice(0x02);      // Select SD card as the player device.
72      SetVolume(0x0E);               // Set the volume, the range is 0x00 to 0x1E.
73  }
74
75  void loop()
76  {
77      uint8_t len = 0;
78      uint8_t i;
79
80      if(Serial.available())
81      {
82          char chr = '\0';
83          while(chr != '\n')  // Blockly read data from serial monitor
84          {
85              chr = Serial.read();
86              // Serial.print(chr);
87              recv_cmd[len++] = chr;
88          }
89      }
90
91      if(len > 0)
92      {
93          // Print reveiced data
94          // Serial.print("Received cmd: ");
95          // for(i = 0; i < len; i++) {
96          //     Serial.print(recv_cmd[i]);
97          //     Serial.print(" ");
98          // }
99          // Serial.println();
100
101         switch (recv_cmd[0])
```

```
102      {
103        case 'P':
104          if(recv_cmd[1] == 'm')
105          {
106            /**
107             * Play music in "MP3" folder by index
108             * example:
109             * "Pm1" -> ./MP3/0001.mp3
110             */
111            PlayMP3folder(recv_cmd[2] - '0');
112            Serial.print("Play ");
113            Serial.write(recv_cmd[2]);
114            Serial.println(".mp3 in MP3 folder");
115          }
116          else if(recv_cmd[1] == 'f')
117          {
118            /**
119             * Play specify folder and music
120             * example:
121             * "Pf11" -> ./01/001***.mp3
122             */
123            SpecifyfolderPlay(recv_cmd[2] - '0',recv_cmd[3] - '0');
124            Serial.print("Play ");
125            Serial.write(recv_cmd[3]);
126            Serial.print("xxx.mp3");
127            Serial.print(" in folder ");
128            Serial.write(recv_cmd[2]);
129            Serial.println();
130
131          }
132          else
133          {
134            /**
135             * Play music by default index
136             * example:
137             * "P1" -> ./***.mp3
138             */
139            SpecifyMusicPlay(recv_cmd[1] - '0');
140            Serial.print("Play xxx.MP3 by index ");
141            Serial.write(recv_cmd[1]);
142            Serial.println();
143          }
144          // Serial.println("Specify the music index to play");
145          break;
146        case 'p':
147          PlayPause();
148          Serial.println("Pause the MP3 player");
149          break;
150        case 'R':
151          PlayResume();
152          Serial.println("Resume the MP3 player");
153          break;
154        case 'N':
155          PlayNext();
156          Serial.println("Play the next song");
157          break;
```

```
158        case 'L':
159            PlayPrevious();
160            Serial.println("Play the previous song");
161            break;
162        case 'l':
163            PlayLoop();
164            Serial.println("Play loop for all the songs");
165            break;
166        case 'I':
167            IncreaseVolume();
168            Serial.println("Increase volume");
169            break;
170        case 'D':
171            DecreaseVolume();
172            Serial.println("Decrease volume");
173            break;
174        default:
175            break;
176      }
177
178      // clean data buffer
179      for(i = 0; i < sizeof(recv_cmd); i++) {
180        recv_cmd[i] = '\0';
181      }
182   }
      delay(100);

   //  printReturnedData();
   }
```

- **Step 4.** We will see the info at COM terminal as below.

Note
For advanced programming, you can download datasheet of chip KT403A.

- **Step 5.** Please key in the related command to play the music.

Resources

- **[Eagle&PDF]** Grove-MP3_v2.0_Schematic files
- **[Librarie]** Grove-Serial MP3 Player V2.0 Libraries
- **[Datasheet]** KT403A Datasheet

Projects

Arduino Based Security Project Using Cayenne: Arduino based Home Security Project. SMS/email alerts when intruder detected. Personalized messages for authorized personnel.

Leaf Piano: We made a piano using a touch sensor, as well as leaves for piano keys.

Tech Support

Please submit any technical issue into our forum or drop mail to techsupport@seeed.cc.