Music Shield V2.0

From Wiki 来自痴汉的爱

[中文 (http://www.seeedstudio.com/wiki/index.php?title=Music_Shield_拓展板_V2.0&uselang=zh)]

Contents

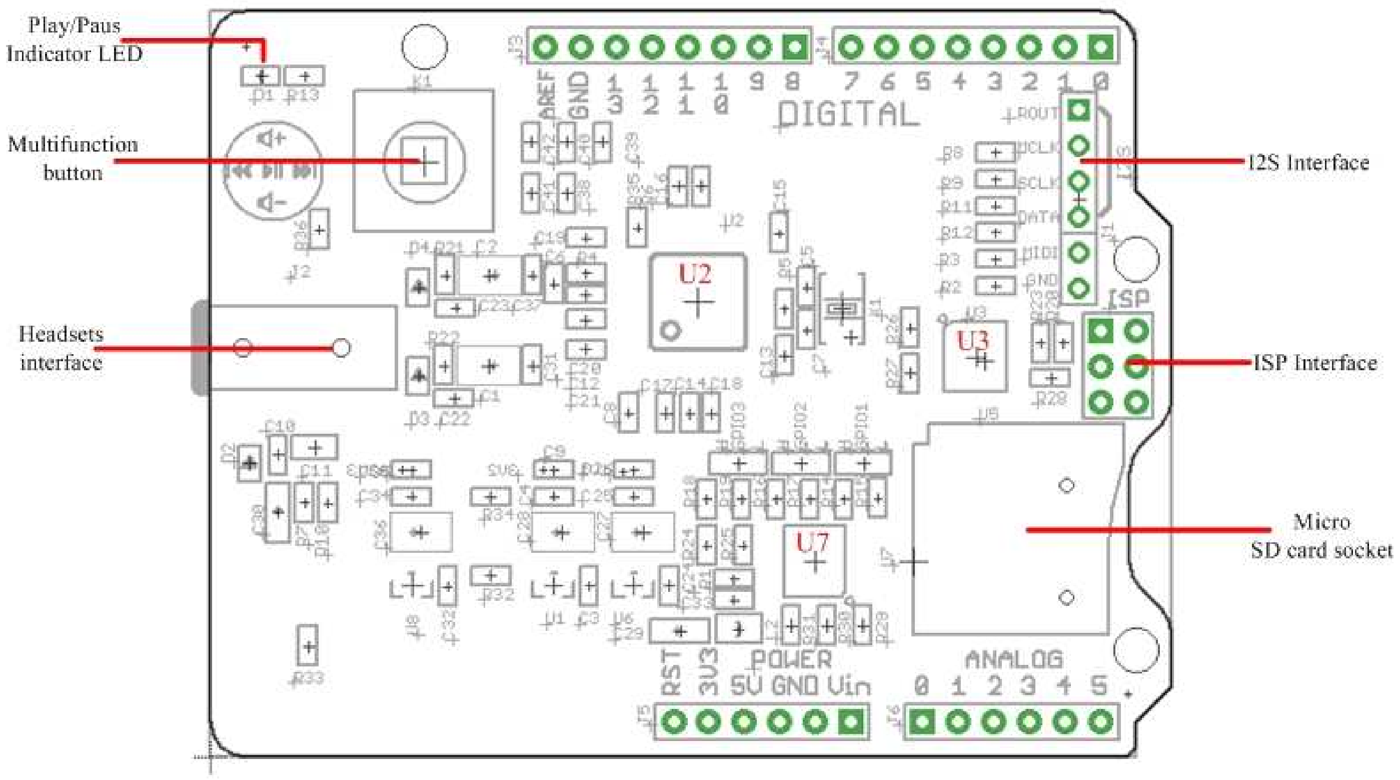
Introduction

Time to build your real-time MIDI instrument/music player! It can play many format including MP3,WMA,WAV,AAC,MIDI,Ogg VorbisThe. Music Shield is an audio encoder/decoder compatible with Arduino, Seeeduino, Seeeduino Mega and Arduino Mega. It is based on the VC1053B chip, which enabled it to play sound files from SD card and do short-time recording as well. You can also use it to play MIDI notes by slightly changing its hardware installations. Due to the SPI communication mode, it keeps a minimum number of IO port that facilitates users' own developments of this device. Additionally, the new multifunction button provides greater convenience for users to control.

Notice: The recording function works with Seeeduino Mega and Arduino Mega only. And the maximum size SD card you can use is 2GB.

Specification

↑TOP



Multifunction button: Change volume and select songs
Play/Pause indicator LED (GREEN)：Blinks while playing.
Headsets interface: It can drive 16 ohm or 32 ohm earphone and could serve as a external audio input port.
Micro SD card: can be FAT16 or FAT32, The maximum size SD card you can use is 2GB.
U2: VS1053B IC,Ogg Vorbis/MP3/AAC/WMA/FLAC/MIDI audio codec.
U3,U7: 74VHC125 IC, Quad Buffer.
I2S: for digital audio input/output.
ISP interface: for bringing SPI port when using with Mega series products.

Pins usage on Arduino

Pins Used for Play Control:
D3 - receiving signal from button for Volume Up;
D4 - receiving signal from switch for Next Song function;
D5 - receiving signal from switch for Play&Stop and Record function;
D6 - receiving signal from switch for Previous Song function;
D7 - receiving signal from button for Volume Down.
D8 - Green Led instructions;

Pins Used for SPI Interface:
D10 - SPI Chip Select;
D11 - SPI MOSI;
D12 - SPI MISO;
D13 - SPI SCK;

Pins Used for VS1053 Interface:
A0 - Reset of VS1053;
A1- Data Require of VS1053;
A2 - Data Select of VS1053;
A3 - Chip Select of VS1053;

↑TOP

Demonstration

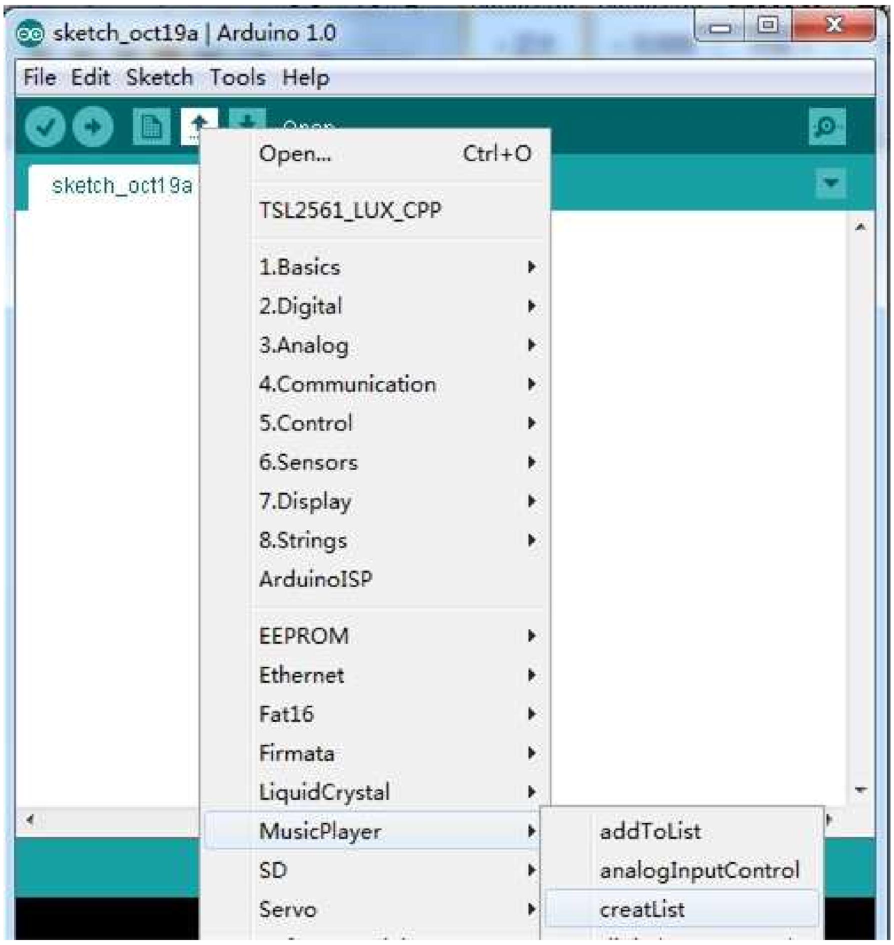
Note: 1. If you want to use MIDI function, you need to change the hardware installation.
2. If you changed the hardware installation in order to use MIDI function, you are not able to use playback & recording functions until you restore it to the original condition.

Play music

1. Make sure there are songs in the micro SD card;
2. Download Music shield V2.0 library (https://github.com/Seeed-Studio/Music_Shield)
3. Unzip and copy the folder to Arduino’ s library path: ..\arduino-1.0\libraries;

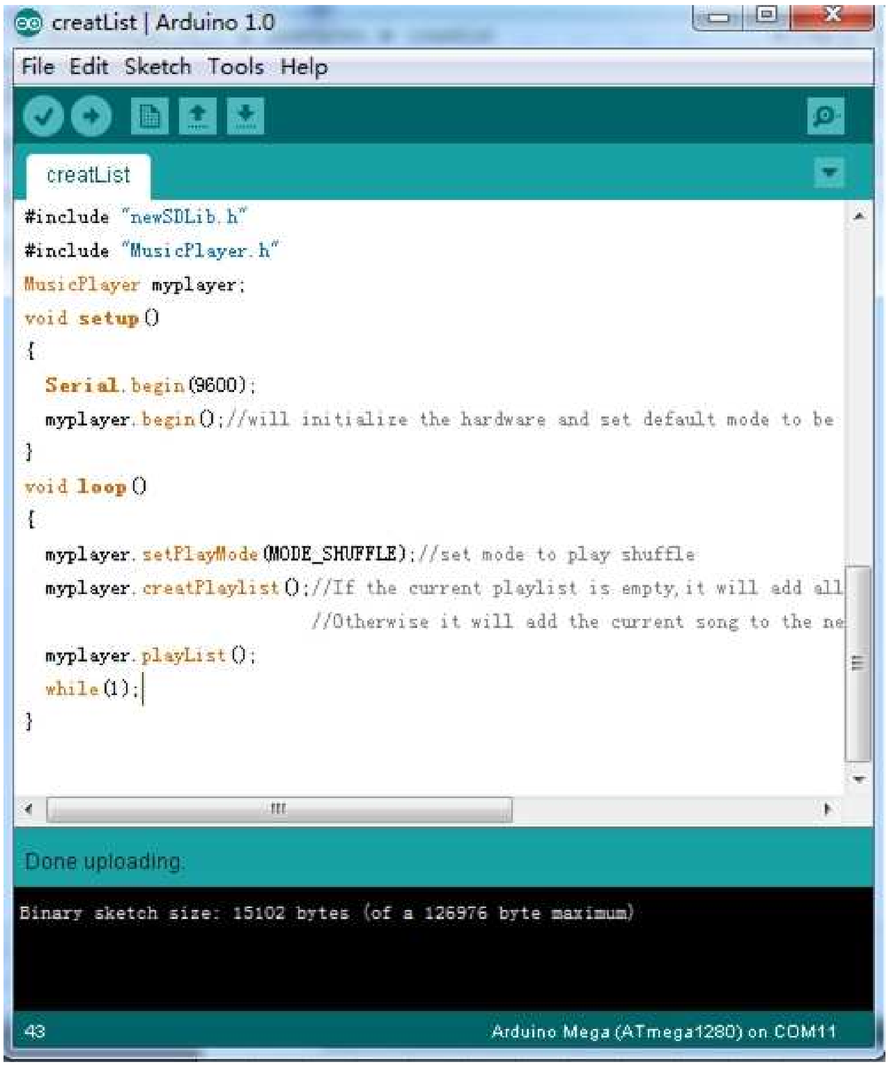
Demo 1: Play songs (e.g. in shuffle mode)
In order to use the playback function, you need to create a playlist first.
1. Restart the Arduino IDE. Open “creatList” example via the path: File --> Examples --> MusicPlayer --> creatList as below.

↑TOP



2. Set the play mode. In “creatList”, the function we use is described as follow.
Name: setPlayMode(unsigned char playmode);
Function: Set the play mode. There are four modes you can set: MODE_NORMAL、MODE_SHUFFLE、 MODE_REPEAT_LIST、 MODE_REPEAT_ONE. Each mode stands for different playing orders.

↑TOP



```
#include "newSDLib.h"
#include "MusicPlayer.h"
MusicPlayer myplayer;
void setup()
{
  Serial.begin(9600);
  myplayer.begin();//will initialize the hardware and set default mode to be
}
void loop()
{
  myplayer.setPlayMode(MODE_SHUFFLE);//set mode to play shuffle
  myplayer.creatPlaylist();//If the current playlist is empty,it will add all
                           //Otherwise it will add the current song to the ne
  myplayer.playList();
  while(1);
}
```

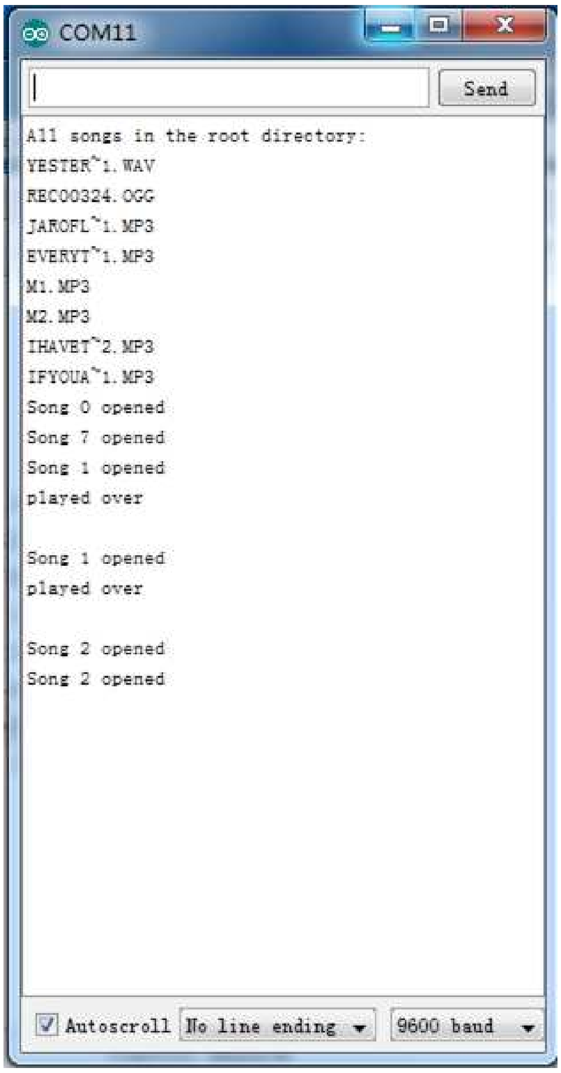
Done uploading.

```
Binary sketch size: 15102 bytes (of a 126976 byte maximum)
```

43 Arduino Mega (ATmega1280) on COM11

3. Select the type of Arduino board that you are using by the path: Tools --> Board --> for example Arduino UNO.
4. Select the correct serial port you are using by the path: Tools --> Serial Port --> for example COM3.
5. Upload the code. Click to Serial Monitor when “Done uploading” appears, you will find the order of songs is randomized on the list.

↑TOP



```
All songs in the root directory:
YESTER~1.WAV
REC00324.OGG
JAROFL~1.MP3
EVERYT~1.MP3
M1.MP3
M2.MP3
IHAVET~2.MP3
IFYOUA~1.MP3
Song 0 opened
Song 7 opened
Song 1 opened
played over

Song 1 opened
played over

Song 2 opened
Song 2 opened
```

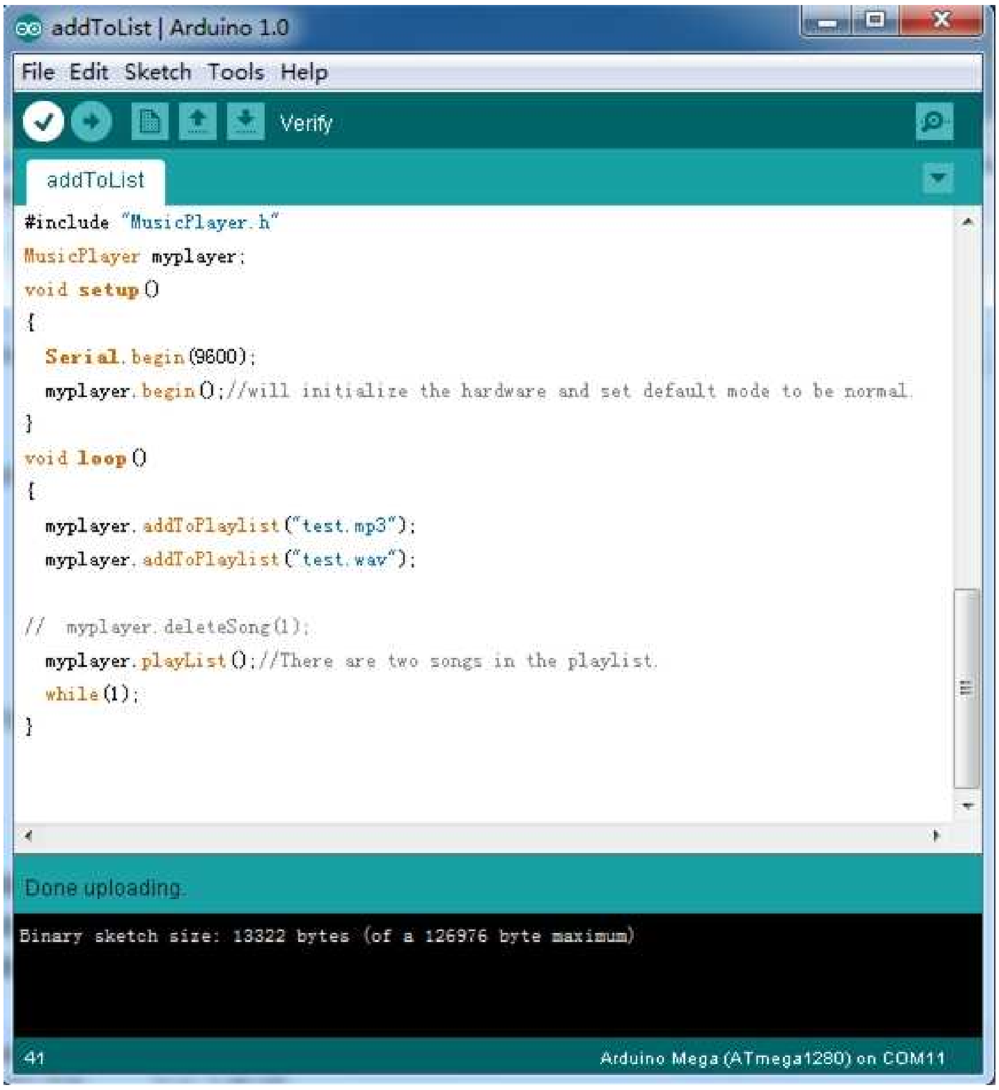
When press multifunction button to up or down, the volume will change. Of course, you can try others play modes.

.

Demo 2: Play selected songs

1. This demo will show you how to play part of the songs from all songs in the SD card. Open the “addToList” example via the path: File --> Examples --> MusicPlayer --> addToList.

↑TOP



```
#include "MusicPlayer.h"
MusicPlayer myplayer;
void setup()
{
  Serial.begin(9600);
  myplayer.begin();//will initialize the hardware and set default mode to be normal.
}
void loop()
{
  myplayer.addToPlaylist("test.mp3");
  myplayer.addToPlaylist("test.wav");

//  myplayer.deleteSong(1);
  myplayer.playList();//There are two songs in the playlist.
  while(1);
}
```

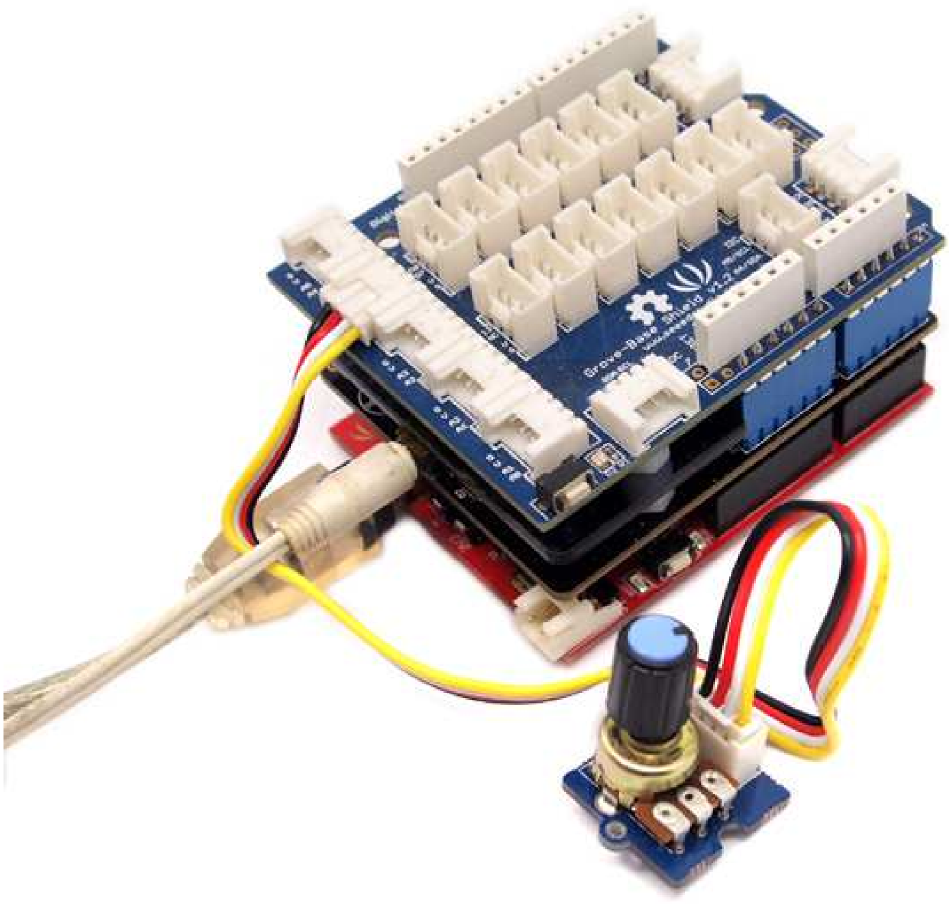
```
Done uploading.
Binary sketch size: 13322 bytes (of a 126976 byte maximum)
```

2.Select songs from the playlist.You just need to list songs you want to play by name correctly in the function addToPlayList(char *songName) .
But you must ensure that the song has been stored in the SD card and the format of those songs must be one of MP3,WMA,WAV,AAC,MIDI,Ogg Vorbis.
3.Upload code. When you complete the upload,new add songs will be played.

Demo 3: Control Volume by analog port

1．Plug the Grove-Base Shield onto the Music shield, Connect the Grove socket of the Rotary and anolog port 4 of the Base Shield with a Grove cable. You can change to
the digital port as well. But don't forget to change the port number in the definition of the demo code at the same time.

↑TOP



2. Open the "analogInputControl" example and upload it onto your Arduino Board.
3. Rotate the knob to change music volume.

Demo 4: Record music:(Only support ATmega1280 and ATmega2560 based board)

1. Upload any sketch in Music Shield library, for example the sketch "creatList". Open the Serial Monitor and it will play audio files on SD card.
2. Press down the multifunction button for 5 seconds, then the indicator LED will light off.
3. Press down the multifunction button for 5 seconds again, then the music shield will begin to record, the green indicator LED will blink.
4. Quickly press down the multifunction button again, it will stop recording.
5. Record will be played in the last place.

Using MIDI,no need to modify the hardware

The VS1058X's real time MIDI mode:

The "real time MIDI mode",in which it will instantly execute MIDI commands send to it through either SPI or UART,can be enabled with the method below:

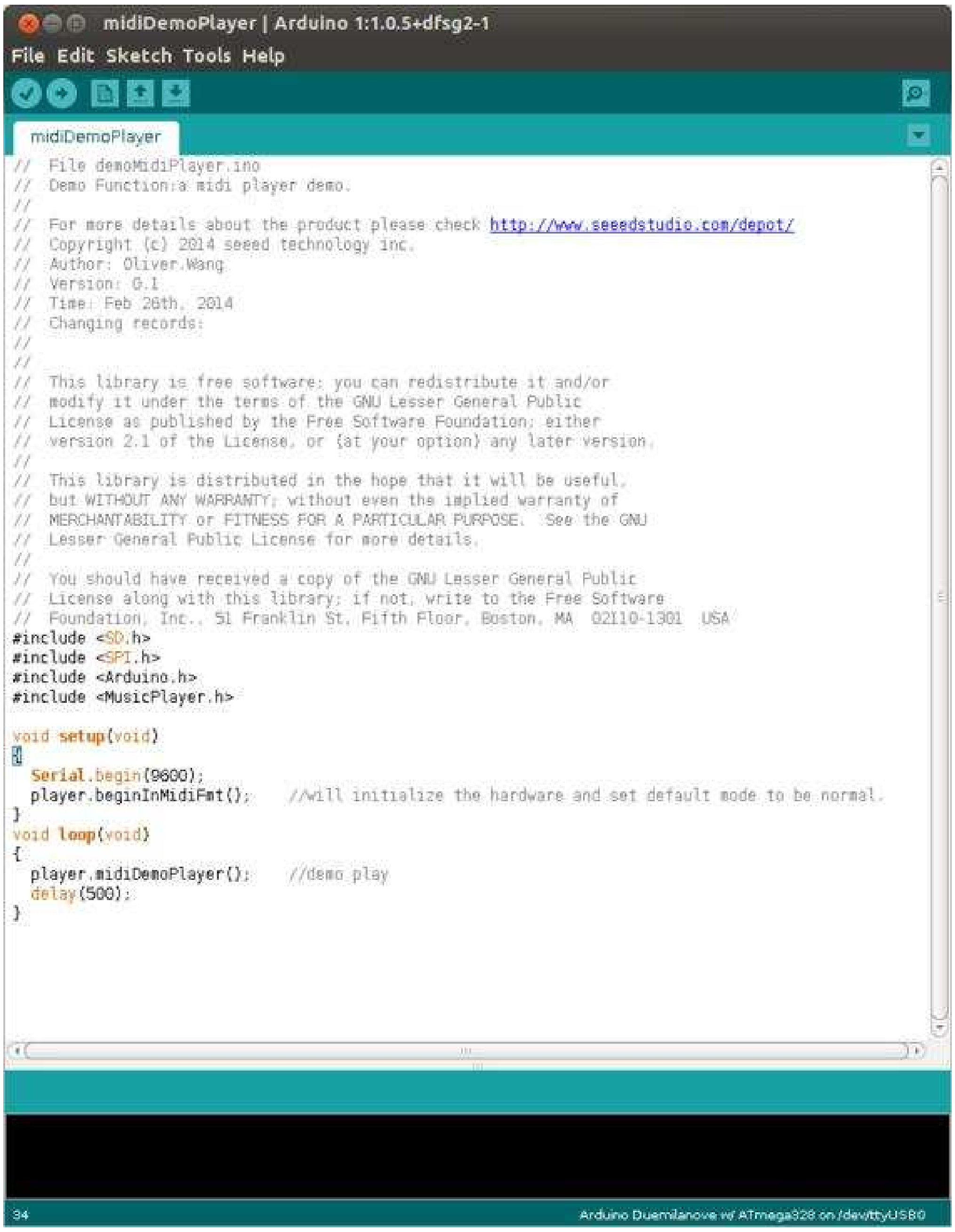
Method:

```
    At the beginning,send a small software patch through SPI port.

/*software patch for MIDI Play*/
const unsigned short gVS1053_MIDI_Patch[28]={
/*if you don't let GPIO1 = H,please send this patch by spi*/
0x0007, 0x0001, 0x8050, 0x0006, 0x0014, 0x0030, 0x0715, 0xb080, /* 0 */
0x3400, 0x0007, 0x9255, 0x3d00, 0x0024, 0x0030, 0x0295, 0x6890, /* 8 */
0x3400, 0x0030, 0x0495, 0x3d00, 0x0024, 0x2908, 0x4d40, 0x0030, /* 10 */
0x0200, 0x000a, 0x0001, 0x0050,
};
```

↑TOP

```
using that function to load:
/*
**@ function name: loadMidiPatch
**@ usage:load a software patch for vs10xx
**@ input:none
**@ retval:none
*/
void VS10XX::loadMidiPlugin(void)
{
  int i=0;
  Serial.print("load MIDI Plugin...\r\n");
  while(i < sizeof(gVS1053_MIDI_Patch)/sizeof(gVS1053_MIDI_Patch[0]))
  {
    unsigned short addr, n, val;
    addr = gVS1053_MIDI_Patch[i++];
    n = gVS1053_MIDI_Patch[i++];
    while(n--)
    {
      val = gVS1053_MIDI_Patch[i++];
      writeRegister(addr, val >> 8, val&0xff);
    }
  }
  Serial.print("done\r\n");
}
I would like to tell you that there is an open source library called jdksmidi,by which you can make your own MIDI decod
jdksmidi git-hub page:https://github.com/jdkoftinoff/jdksmidi
we offer you some real time mode MIDI APIs(MusicPlayer.cpp):
midiNoteOn()
midiNoteOff()
midiWriteData()

Now,it's time to build your real-time MIDI instrument/music player in any format(single-channel or multi-channel).
Your contribution is appreciated.
A demo MIDI player was add to the latest library.
MIDI Demo(upload the code. When completed, you will hear Fancy MIDI music):
```

↑TOP


midiDemoPlayer | Arduino 1:1.0.5+dfsg2-1
File Edit Sketch Tools Help
midiDemoPlayer
34
Arduino Duemilanove w/ ATmega328 on /dev/ttyUSB0

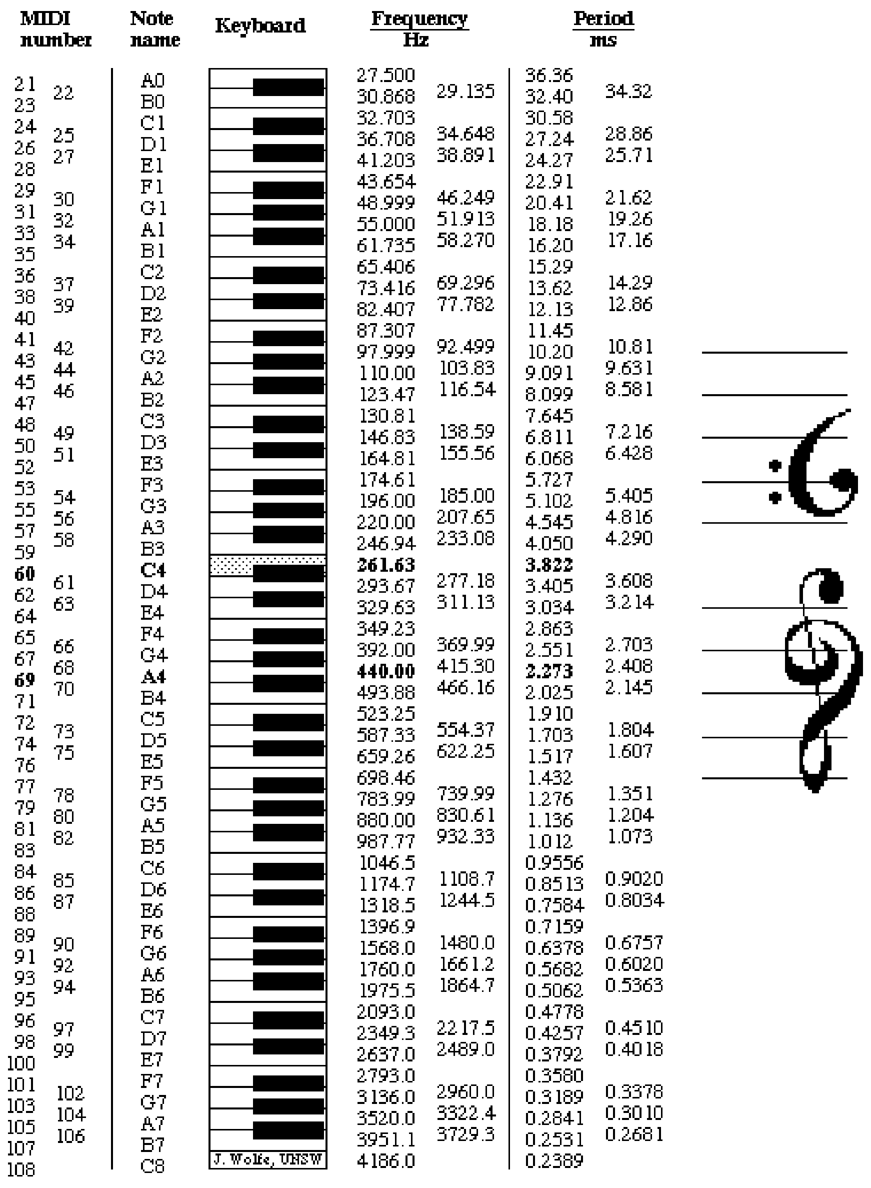
```
//  File demoMidiPlayer.ino
//  Demo Function:a midi player demo.
//
//  For more details about the product please check http://www.seeedstudio.com/depot/
//  Copyright (c) 2014 seeed technology inc.
//  Author: Oliver.Wang
//  Version: 0.1
//  Time: Feb 20th, 2014
//  Changing records:
//
//
//  This library is free software; you can redistribute it and/or
//  modify it under the terms of the GNU Lesser General Public
//  License as published by the Free Software Foundation; either
//  version 2.1 of the License, or (at your option) any later version.
//
//  This library is distributed in the hope that it will be useful,
//  but WITHOUT ANY WARRANTY; without even the implied warranty of
//  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
//  Lesser General Public License for more details.
//
//  You should have received a copy of the GNU Lesser General Public
//  License along with this library; if not, write to the Free Software
//  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
#include <SD.h>
#include <SPI.h>
#include <Arduino.h>
#include <MusicPlayer.h>

void setup(void)
{
  Serial.begin(9600);
  player.beginInMidiFmt();    //will initialize the hardware and set default mode to be normal.
}
void loop(void)
{
  player.midiDemoPlayer();    //demo play
  delay(500);
}
```

Resources

Music Shield Eagle Files (http://www.seeedstudio.com/wiki/File:Music_shield_eagle_files.zip)
Music Shield Schematic.pdf (http://www.seeedstudio.com/wiki/File:Music_Shield_Schematic.pdf)
VS1053 IC.pdf (http://seeedstudio.com/wiki/File:VS1053.pdf)
Music Shield V2.0 libraries (https://github.com/Seeed-Studio/Music_Shield)

MIDI number to note reference list:

↑TOP


Keyboard
J. Wolfe, UNSW

MIDI number		Note name	Frequency Hz		Period ms	
21	22	A0	27.500	29.135	36.36	34.32
23		B0	30.868		32.40	
24	25	C1	32.703	34.648	30.58	28.86
26	27	D1	36.708	38.891	27.24	25.71
28		E1	41.203		24.27	
29	30	F1	43.654	46.249	22.91	21.62
31	32	G1	48.999	51.913	20.41	19.26
33	34	A1	55.000	58.270	18.18	17.16
35		B1	61.735		16.20	
36	37	C2	65.406	69.296	15.29	14.29
38	39	D2	73.416	77.782	13.62	12.86
40		E2	82.407		12.13	
41	42	F2	87.307	92.499	11.45	10.81
43	44	G2	97.999	103.83	10.20	9.631
45	46	A2	110.00	116.54	9.091	8.581
47		B2	123.47		8.099	
48	49	C3	130.81	138.59	7.645	7.216
50	51	D3	146.83	155.56	6.811	6.428
52		E3	164.81		6.068	
53	54	F3	174.61	185.00	5.727	5.405
55	56	G3	196.00	207.65	5.102	4.816
57	58	A3	220.00	233.08	4.545	4.290
59		B3	246.94		4.050	
60	61	**C4**	**261.63**	277.18	**3.822**	3.608
62	63	D4	293.67	311.13	3.405	3.214
64		E4	329.63		3.034	
65	66	F4	349.23	369.99	2.863	2.703
67	68	G4	392.00	415.30	2.551	2.408
69	70	**A4**	**440.00**	466.16	**2.273**	2.145
71		B4	493.88		2.025	
72	73	C5	523.25	554.37	1.910	1.804
74	75	D5	587.33	622.25	1.703	1.607
76		E5	659.26		1.517	
77	78	F5	698.46	739.99	1.432	1.351
79	80	G5	783.99	830.61	1.276	1.204
81	82	A5	880.00	932.33	1.136	1.073
83		B5	987.77		1.012	
84	85	C6	1046.5	1108.7	0.9556	0.9020
86	87	D6	1174.7	1244.5	0.8513	0.8034
88		E6	1318.5		0.7584	
89	90	F6	1396.9	1480.0	0.7159	0.6757
91	92	G6	1568.0	1661.2	0.6378	0.6020
93	94	A6	1760.0	1864.7	0.5682	0.5363
95		B6	1975.5		0.5062	
96	97	C7	2093.0	2217.5	0.4778	0.4510
98	99	D7	2349.3	2489.0	0.4257	0.4018
100		E7	2637.0		0.3792	
101	102	F7	2793.0	2960.0	0.3580	0.3378
103	104	G7	3136.0	3322.4	0.3189	0.3010
105	106	A7	3520.0	3729.3	0.2841	0.2681
107		B7	3951.1		0.2531	
108		C8	4186.0		0.2389	

Retrieved from "http://www.seeedstudio.com/wiki/index.php?title=Music_Shield_V2.0&oldid=100983"

- This page was last modified on 2 December 2014, at 06:09.
- This page has been accessed 44,201 times.

↑TOP