Arduino MP3 Shield User Guide
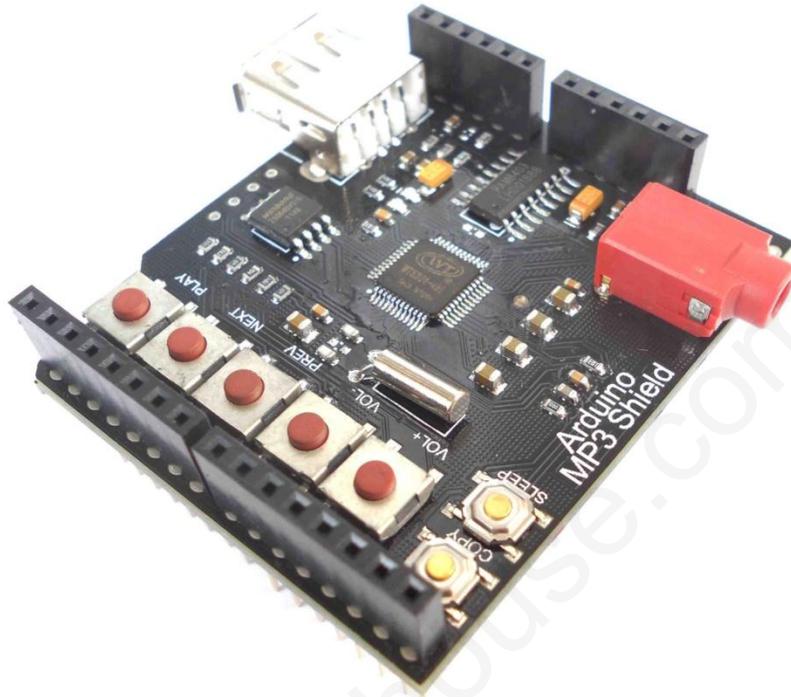
Instruction
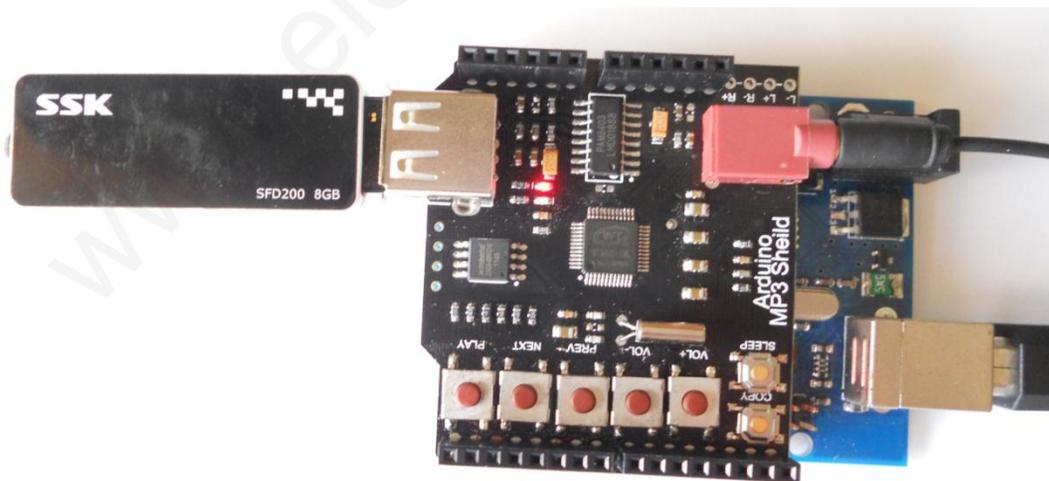
Arduino MP3 Shield is a high quality MP3 module, supporting standard MP3 and WAV format audio files. It also supports three kinds of storage devices, including an onboard SPI-Flash, external SD Card and U-Disk.
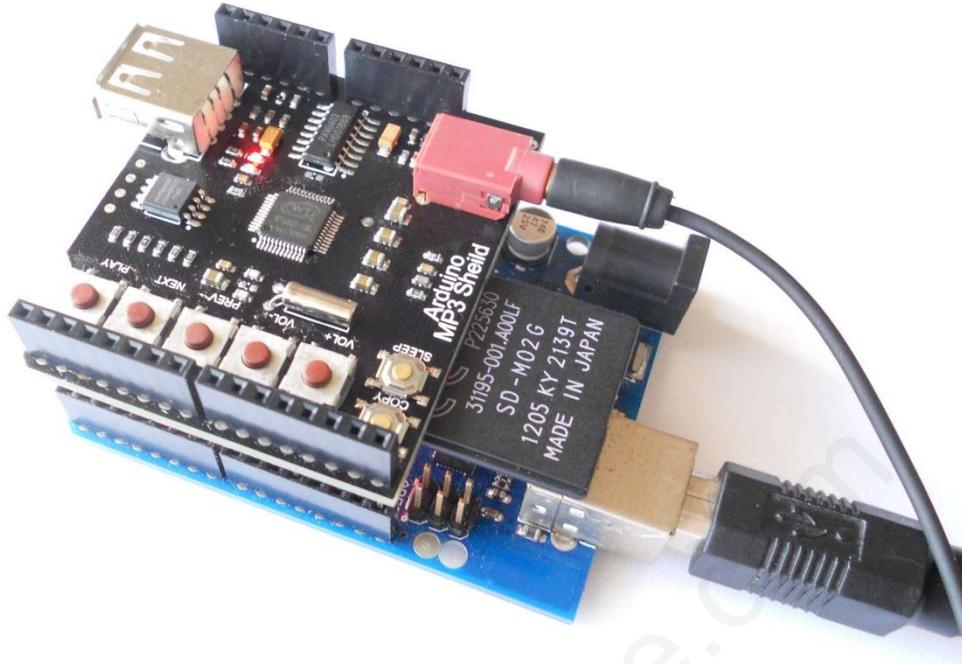
The Arduino MP3 Shield has an onboard 3W Class-D stereo audio amplifier which could drive 4ohm load such as loudspeaker.

The Arduino MP3 Shield has seven buttons and a UART interface, that is to say user has two ways to control the module flexibly. The UART interface is connected to the Arduino hardware UART by default. Since most Arduino boards have only one UART interface, this module's UART interface can be redefined to Arduino D7 (TX) and D8 (RX) through the jumpers on board. With SoftSerial lib, Arduino could also send data to this module.

We tested with 8G SDHC card and 16G U-disk. It works in perfect compatibility. We haven't tested larger SD and U-disk yet.

Feature

- Plug in and play for Arduino
- Control by Serial UART interface or keys onboard.
- Directly drive 32ohm headset.
- On-board Audio Amplifier to drive 3W/4Ω external loudspeaker.
- 32 levels adjustable volume.
- Support FAT16/FAT32 file system.
- Support up to SD/SDHC card and U-Disk.
- On board 64MBit SPI-FLASH.
- Support copying audio files from SD card or U-Disk to SPI-FLASH.
- Supporting Sleep Mode.
- Support 16K~320Kbps MP3 files and 8K~44.1KHz sampling rate WAV files (SPI-FLASH only supports MP3 format)

Parameter

- Power Supply: 4.5V~5.5V, typically 5V.
- Interface: UART (5V TTL) and 7 buttons.
- Size: 47mm x 53mm.

LED

There are 2 LEDs on the board.

- *Power indicating: the LED is always on when power is supplied.*
- *State indicating: LED is on when playing music, blinking when copying files to onboard SPI-FLASH.*

Function Description

On board Audio Amplifier could drive up to 3W/4Ω external loudspeaker. Infact, we test it with 5W/8Ω loudspeker, and it works well.
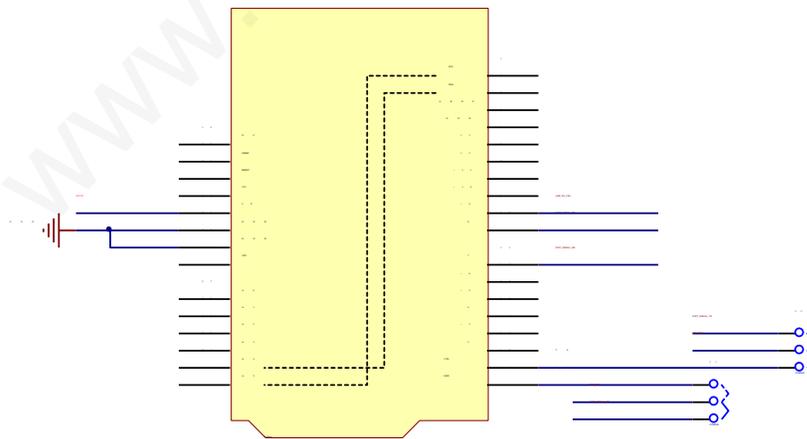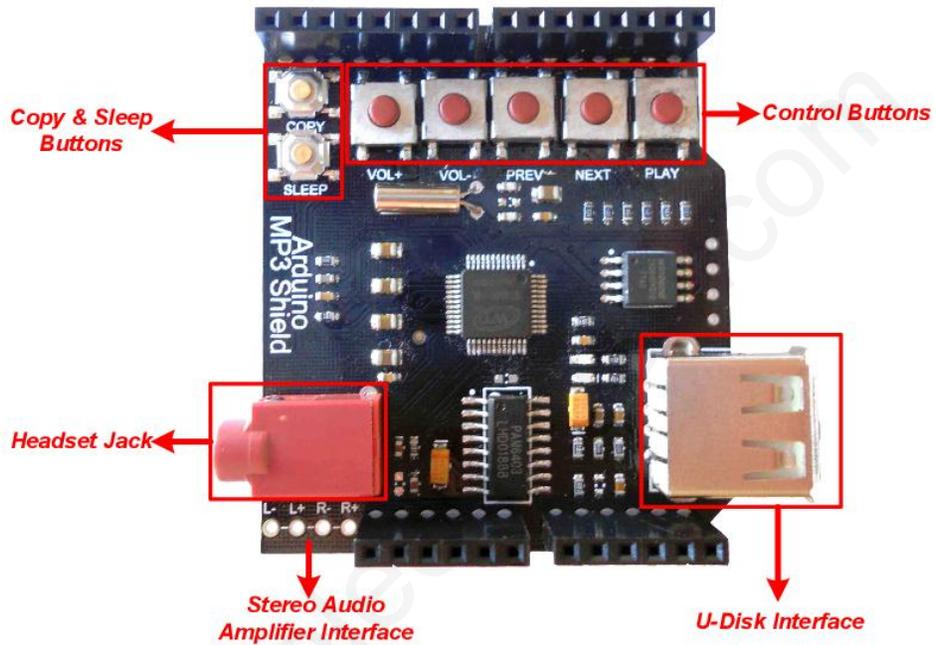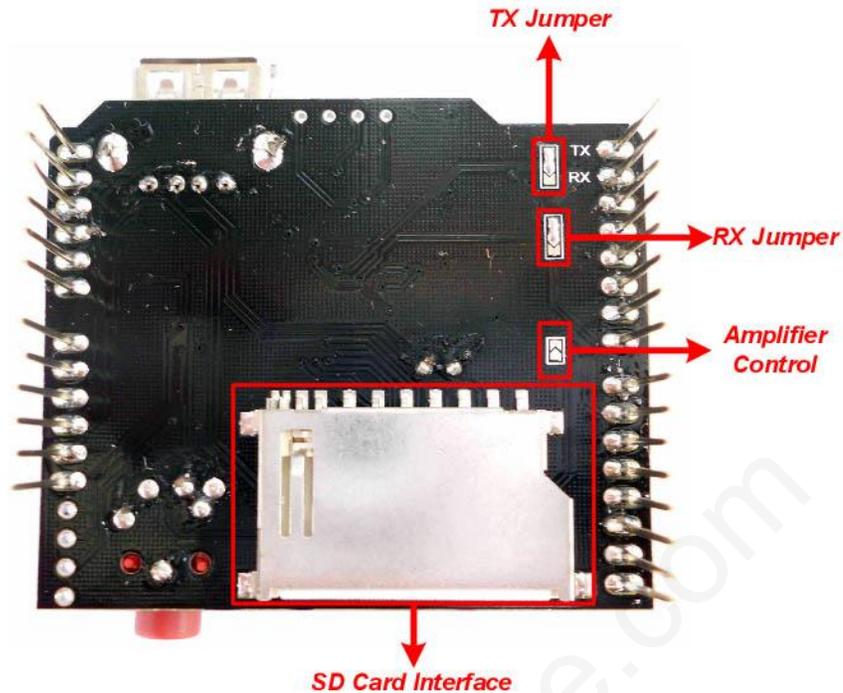
By default, the RX Jumper and TX Jumper are connected to hardware UART pins on Arduino. You could change it with soldering tool. If you bridge the lower two pads, Arduino D7 and D8 need to serve as RX and TX. You could find more information in latter description.

The amplifier could be controlled by Arduino. By default, it is enabled and not controlled by Arduino. If you need, please bright the Amplifier Control pads. More information will be explained later.

CONFIGURATION DESCRIPTION

Arduino MP3 Shield configuration is stored in a file named "iSound.mp3" (Note that in fact it is not an audio file but a txt file. You could open it with software such as Notepad). The format of the "iSound.mp3" is as bellows:

spxx//
clxx//
cfxx xx xx xx//

Details:

spxx// **Play settings when powered on**. *sp must be lower case letters,* **xx** *must be 00 or 01*. **00** *means* **"Do not play automatically when powered on"**. **01** *means* **"Play automatically when power on".**

clxx// **Play Mode Settings**. *cl must be lower case letters,* **xx** *must be 00, 01, 02 or 03.* **00** *means* **Play single music,** **01** *means* **Repeat track,** **02** *means* **Repeat all,** *and* **03** *means* **Play randomly.**

cfxx xx xx xx// **User-defined data.** *User can read these four bytes data through UART interface. Refer to* [Get user-defined data](#)*.*

"iSound.mp3" is a file where you could configure the Arduino MP3 Shield. Follow the steps below to configure it:

Create a text file named "iSound.mp3" on your PC. Usually open the Notepad in Windows (on Mac or Linux, see the note), create a new file, enter the data in the rule above, and save as mp3 file.

Copy the "iSound.mp3" to U-Disk or SD card. Make sure that iSound.mp3 is the last file copied to U-Disk or SD card. This could ensure that at next step, iSound.mp3 file is the last file which is written to the flash.

Use COPY function to download music files and the "iSound.mp3"configuration file to the flash. All configuration information is saved in the SPI-FLASH. See COPY MUSIC FROM SD-CARD/ U-DISK TO SPI-FLASH

"iSound.mp3" Example:

We need configure our Arduino MP3 Shield to let it "Play automatically when powered on" and "Repeat all music" and save our user-defined 0x12 0x34 0x56 0x78. We need edit iSound.mp3 like this:

sp01//

cl00//

cf12 34 56 78//

Note: for MAC or Linux users, there will be a problem. EOL (End of line) symbol on Windows OS is "\r\n". But on MAC or Linux, it is "\n". The following is an example showing the difference. We make the newline EOL visible in Notepad++

On Windows OS:
```
1   sp01//CRLF
2   cl03//CRLF
3   cf12 34 56 78//
```

On MAC/Linux:
```
1   sp01//LF
2   cl03//LF
3   cf12 34 56 78//
```

The iSound.mp3 should end with a Windows EOL symbol. That is"\r\n". We recommend Notepad++. You could change the EOL symbol.
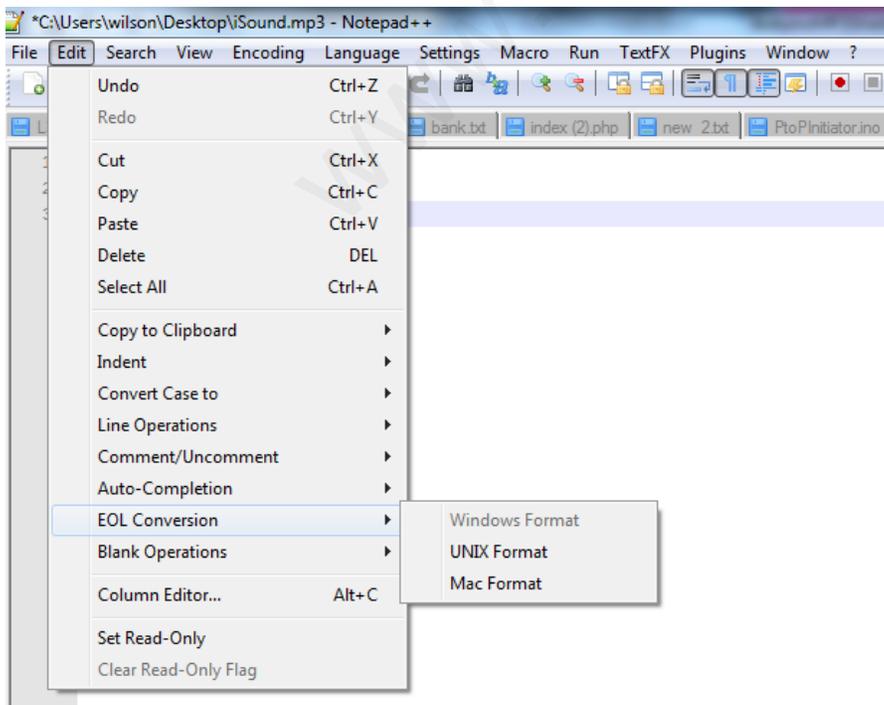
COPY MUSIC FROM SD-CARD/ U-DISK TO SPI-FLASH

The Arduino MP3 Shield support 3 kinds of storage media, SD card, U-Disk and SPI FLASH. The three kinds of memory have the following priority (High first): SPI-FLASH, SD card and U-Disk.

There are two methods to copy music from SD-CARD/U-DISK: through COPY button, or UART interface.

Copy Music through COPY button

Follow the following 5 steps:

1. *Create the "iSound.mp3" configuration file, copy music files to SD card/ U-DISK, then copy " iSound.mp3" to SD card/ U-DISK(this can make sure the "iSound.mp3" is the last file in SD card/U-DISK).*
2. *Power off Arduino MP3 Shield.*
3. *Press COPY button and hold.*
4. *Power on Arduino MP3 Shield, and hold COPY button for about five seconds until the state indicating LED starts blinking.*
5. *Release the Copy key and wait state indicating LED turns off.*

In fact, the configuration file could be stored in SPI-FLASH until you download another "iSound.mp3" file to it. That means, you only have to download the configuration file once. However, each time you download music files to SPI-FLASH, old music files in the SPI will be erased.

Copy Music through UART command

You could also copy files to SPI-FLASH by sending command via UART Interface

See [Copy File from SD card to SPI FLASH](#) and [Copy File from U-Disk to SPI FLASH](#).

NOTE

1. *The music files copied to the SPI-FLASH has the same order with those in the SD card/U-Disk.*
2. *The SPI-FLASH only has 8MB space. Each time you copy files to SPI-FLASH, the first file will be copied anyway, even it is larger than 8MB, of course resulting in bad file in the SPI-FLASH. However, if there is not enough space for the second file, it will not be copied into SPI-FLASH. For example, we have many music files in the SD card, more than 8MB, the first file size is 3MB, and the second is 4MB, and the third one is 2MB. When we do copy operation, only the first two file can be copied to the flash. You can send command to check the file numbers in the SPI-Flash. See [Check Total Number of SPI FLASH Files](#) .*

Music Order

All the music files must be stored in the root directory of SD or U-disk. Any file name is not restricted as long as it obeys FAT file system.

*FAT32 file system has a rule. File order is decided by the file address. File which is created or copied into file system earlier has a higher priority. The Arduino MP3 Shield plays music in that order. To control the music order in which Arduino MP3 Shield plays, we need to control the order in which the music files are saved in SD or U-disk. For example, you could save music one by one to SD card. You save **B.mp3** first and then save **A.mp3**. By default, it will play **B.mp3** fore **A.mp3** file.*

There is a simple way to save music as desired in bulk:

*Create a folder on PC and copy all music files to the folder. Add prefix 0000~9999 to the file names like **0000-God is a girl.mp3**, **0001-Baby.mp3**, **0002-Set Fire To the Rain.mp3** and so on. Then reorder the music by name. Select all files (CTRL+A may help), then copy (CTRL+C may help), and paste these file to the SD card/U-disk. In this way, the music files will be stored in the desired order.*

KEY

Standard Key

KEY	OPERATION	FUNCTION
PLAY	SHORT	Play/Pause music
	LONG	Stop playing
PREV	SHORT	Previous music
	LONG	Fast forward (when playing music)
NEXT	SHORT	Next music
	LONG	Fast backward(when plying music)
VOL-	SHORT	Volume -
	LONG	Volume- fast
VOL+	SHORT	Volume +
	LONG	Volume + fast

COPY and SLEEP

The two bronze head button on the upper left corner are COPY and SLEEP button.

COPY Button

The basis function of COPY button is copy music files to SPI-FLASH. For more information, please refer to Copy Music through COPY button.

Another function of this COPY button is to play next. It functions while this module is the stage of playing music.

SLEEP

Arduino MP3 Shield can sleep or wake up by SLEEP button.

SLEEP: Press SLEEP button and hold 2 more seconds.

WAKE UP: Press SLEEP button to generate a high level pulse to wakeup Arduino MP3 Shield.

JUMPERS

On the bottom of the Arduino MP3 Shield, there are three jumpers, **TX Jumper**, **RX Jumper**, and **Amplifier Control Jumper**. They are used to select alternate function of Arduino MP3 Shield.
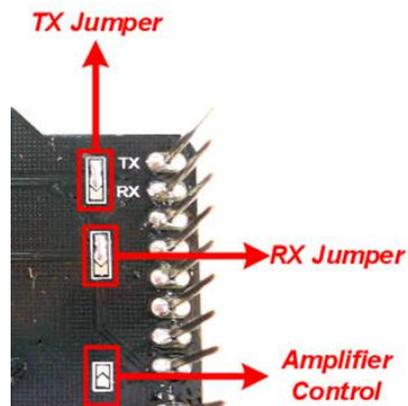
TX and RX Jumper

By default, the Arduino MP3 Shield connects Arduino board with **hardware UART** through the jumper. To change the connection, dis-bridge upper two pads and bridge the two below. Then on this module, D7 becomes TX, D8 becomes RX. **While programming with SoftSerial library on Arduino, Arduino D7 should serve as RX and Arduino D8 as TX.**

Amplifier Control

Amplifier Control pin is used to enable or disable the on board amplifier. By default, the amplifier is enabled, and the control pin is disconnected. To control the amplifier, bridge the two pads of the Amplifier Control Jumper in the picture above. Then the control pin is connected to **Arduino D9**. Set D9 **high** to **enable** the on board amplifier, **low** to **shutdown** the amplifier.

Command Description

Through UART interface, user can send command to select music, select storage, set play mode, inter-cut music, music combination play and so on.

UART Configuration

UART setting:

- Baud Rate: 9600bps
- Parity bit: None
- Start bit: 1
- Data bit: 8
- Stop bit: 1

Command Format

START	LENGTH	COMMAND	PARAMETER	END
0x7E				0x7E

NOTE: LENGTH value is the total count of areas including LENGTH, COMMAND and PARAMETER.

Writing Command

Writing command means commands which tell Arduino MP3 Shield to do something. After each command executed correctly, the command code will be returned.

Play SD Card File

Run this command to play selected SD card file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA0	XX	XX	0x7E

TRACKS HIGH BYTE and **TRACKS LOW BYTE** are HEX numbers. They refer to the music file which you want to reply. This is not the file name, but the file's order number. We have explained the file order above. For example, if you want to play the second music, **TRACKS HIGH BYTE** is **00,** and **TRACKS LOW BYTE** is **02**.

Play SPI FLASH File

Run this command to play selected SPI FLASH file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA1	XX	XX	0x7E

TRACKS HIGH BYTE and **TRACKS LOW BYTE** are the same with **0xA0** command.

Play U-Disk File

Run this command to play selected U-Disk file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA2	XX	XX	0x7E

TRACKS HIGH BYTE and **TRACKS LOW BYTE** are the same with **0xA0** command.

Pause OR Play

Run this command to switch between pause and play.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA3	0x7E

Stop Playing

Run this command to stop playing.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA4	0x7E

Next Music

Run this command to play next music.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA5	0x7E

Previous Music

Run this command to play previous music.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA6	0x7E

Set Volume

Run this command to set volume.

START	LENGTH	COMMAND	VOLUME LEVEL	END
0x7E	0x02	0xA7	0x00~0x1F	0x7E

The volume value is between 0x00~0x1f.

Set Play Mode

Run this command to set play mode.

START	LENGTH	COMMAND	PARAMETER	END
0x7E	0x02	0xA9	0x00 Single Play	0x7E

	0x01 Repeat single music *0x02 Repeat all* *0x03 Play random*

Parameter:

- *0x00: Play one music and stop*
- *0x01: Repeat playing one music*
- *0x02: Repeat playing all music*
- *0x03: Play music randomly*

*We have introduced the iSound.mp3 file to configure the module. What is the difference? Well, configuration in iSound.mp3 file will be loaded each time while this module is powered on. It means that configuration in iSound.mp3 will be saved after powered off. But the setting by **0xA9** command will not be kept while powered off.*

Copy File from SD card to SPI FLASH

Run this command to copy file from SD card to SPI Flash.

START	LENGTH	COMMAND	PARAMETER	END
0x7E	0x03	0xAA	0x00	0x7E

This command will try to copy all the files in SD card to SPI FLASH.

*The Arduino MP3 Shield will return **0xAA** immediately after receiving this command correctly. At the same time the BUSY LED blinks. If copy successfully, it returns **0xAA 0x00**. If failed then return **0xAA 0x01**. After copy finished, BUSY LED turn off. That is to say, in a successful copy operation, you should receive **0xAA 0xAA 0x00**.*

Copy File from U-Disk to SPI FLASH

Run this command to copy file from SD card to SPI Flash.

START	LENGTH	COMMAND	PARAMETER	END
0x7E	0x03	0xAB	0x00	0x7E

This command will try to copy all the files in U-disk to SPI FLASH.

*The Arduino MP3 Shield will return **0xAB** immediately after receiving this command correctly. At the same time the BUSY LED blinks. If copy successfully, it returns **0xAB 0x00**. If failed, it returns **0xAB 0x01**. After copy finished, BUSY LED turn off. That is to say, in a successful copy operation, you should receive **0xAB 0xAB 0x00**.*

Reading Command

Reading command means commands which check the state of Arduino MP3 Shield.

Check Volume

START	LENGTH	COMMAND	END
0x7E	0x02	0xC1	0x7E

Return:

COMMAND	VOLUME
0xC1	*0x00~0x1F*

Check Work State

START	LENGTH	COMMAND	END
0x7E	0x02	0xC2	0x7E

Return value:

COMMAND	RETURN VALUE
0xC2	0x01 Playing 0x02 Stop 0x03 Pause

Check Total Number of SPI FLASH Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC3	0x7E

Return value:

COMMAND	RETURN VALUE
0xC3	File numbers(2Bytes)

Check Total Number of SD Card Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC4	0x7E

Return value:

COMMAND	RETURN VALUE
0xC4	File numbers(2Bytes)

Check Total Number of U-Disk Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC5	0x7E

Return value:

COMMAND	RETURN VALUE
0xC5	File numbers(2Bytes)

Check Current Playing File

START	LENGTH	COMMAND	END
0x7E	0x02	0xC6	0x7E

Return value:

COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE
0xC3	XX	XX

TRACKS HIGH BYTE and **TRACKS LOW BYTE** are the same with **0xA0** command. Please refer to Command **0xA0**

Get user-defined data

START	LENGTH	COMMAND	END
0x7E	0x02	0xC7	0x7E

Return value:

COMMAND	RETURN VALUE
0xC7	XX XX XX XX (4 bytes)

NOTE: Return value is always 5 bytes, one byte COMMAND, and 4 bytes data. If the iSound.mp3 "cf " string don't specify 4 bytes data, then the return value will be 0xFF.

Reference CONFIGURATION DESCRIPTION.

Physical Dimension
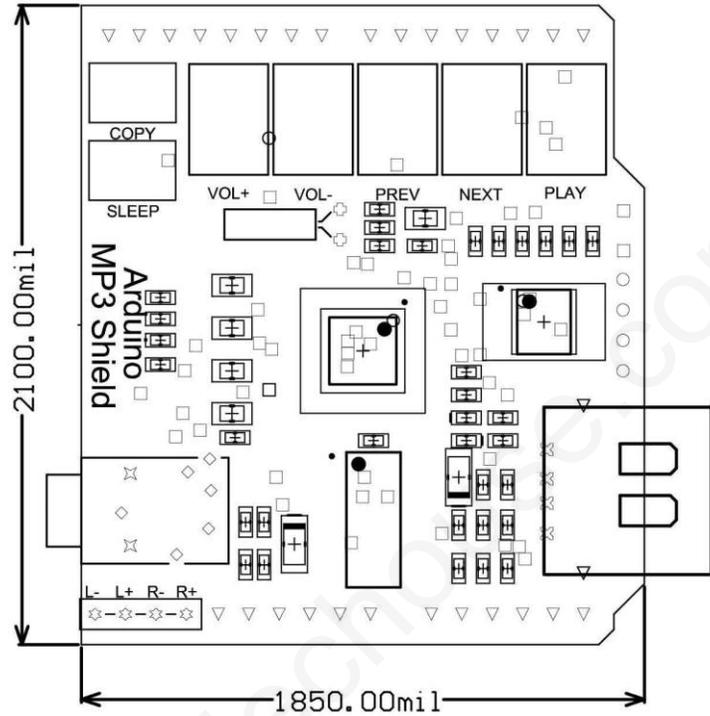
Code

Download the library: MP3_shield_library.zip

Disclaimer and Revisions

The information in this document may change without notice. Please visit www.elechouse.com for new information.

Revision History

Rev.	Date	Author	Description
A	*July 3rd, 2012*	*Lich*	*Initial version*
B	*August 23, 2012*	*Wilson*	*Add the mapping pin*
C	*Oct. 28th, 2012*	*Wilson*	*Add note for the iSound file*