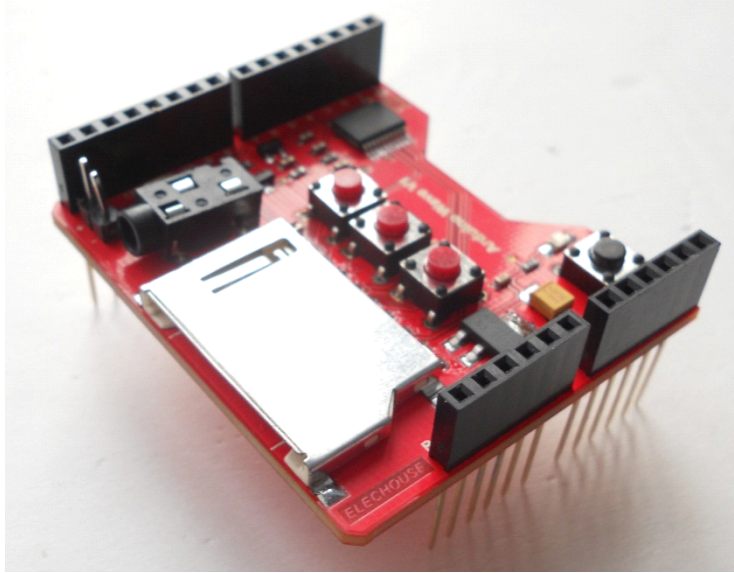


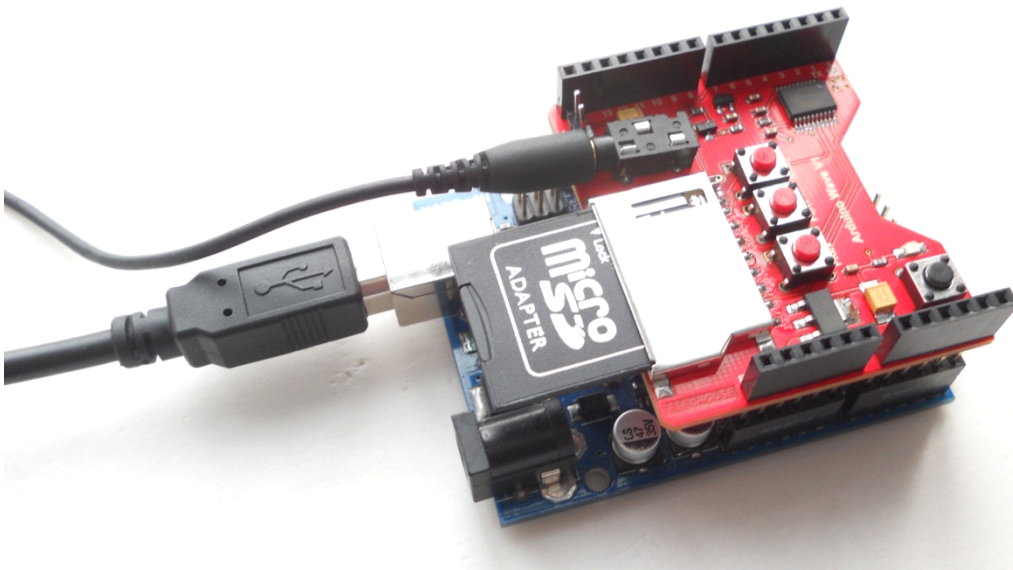
Arduino Wave Module



Introduction

We know that Ladyada designed an Arduino Wave Shield. It is a wonderful Arduino module. Arduino reads the .wave file in the SD card then send to DAC chip, while the sound is generated. But the problem is, it is a little complicated to use this shield for beginners, pretty much code and, it involves file system.

So we are thinking about designing a module, with a chip to do all the file system work. Arduino just need little code to control it. Also, it should be cheap. Finally here comes this shield.



Feature

- Support SD card (Max 1GB)
- Support AD4 (6KHZ ~ 36KHZ) and WAV (6KHz~14KHz) format.

- *Support 4 Bit ADCPM format files.*
- *Interfaces for earphone and speaker.*
- *16 Bit DAC / PWM audio output.*
- *Two line serial interface, control it only via 2 pins.*
- *Sleep current: 3uA*
- *Mono output*

This module can directly drive 0.5W 8 ohm speaker via PWM output. We also add socket for earphone via DAC output. No matter what your MCU is, you only need two wires to control this module. Code size is very small and very easy to be embedded into your project code.

How to use it

In this document we will show you how simple to use this module.

SD card

SD card has to be less than 1G, which is one of defects of this module. Considering it might be hard to buy SD card less than 1G in the market, we will include a SD card along with this module if you choose the option.

Format the card to FAT system. Please refer to this link if you don't know how to format it.

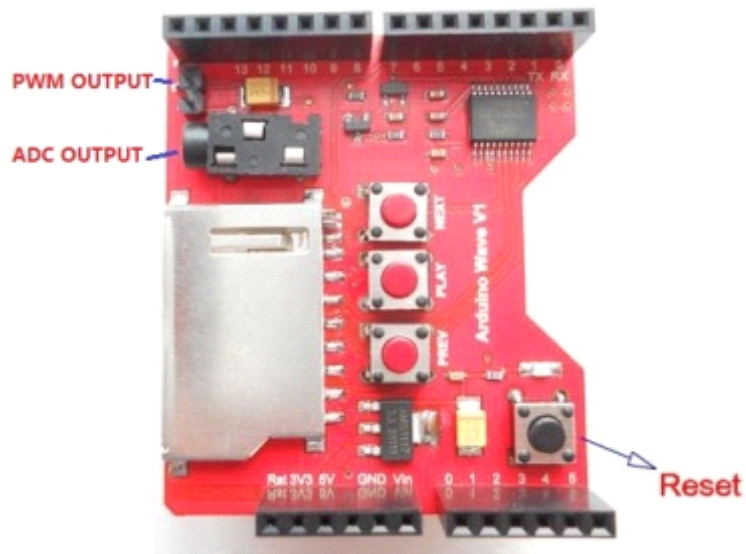
File

This module could play AD4 and WAV file. You can convert MP3 file to AD4 or WAV. To create AD4 files, [download this software](#). In the package contains instruction showing how to use it. To create WAV file, you can refer to this document. We also supply audio sample.

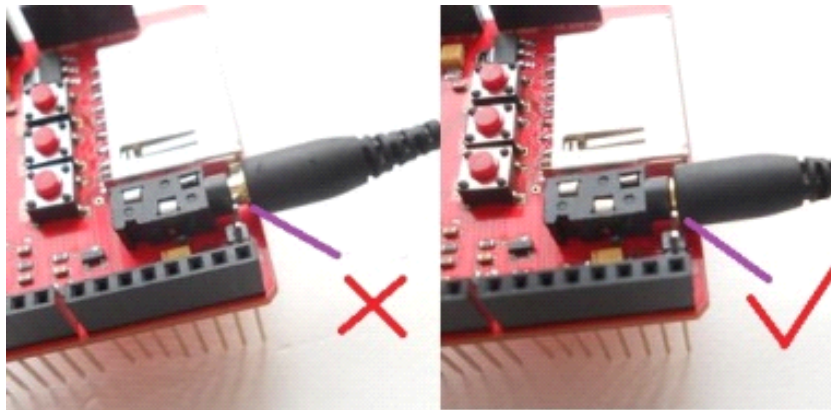
Load voice files to root directory of SD card, and rename the files in this way: for WAV format voice, name in hex 0000.wav, 0001.wav, 0002.wav.....; for ad4 format voice, name in hex 0000.ad4, 0001.ad4, 0002.ad4..... This module can play max 512 files.

Connection

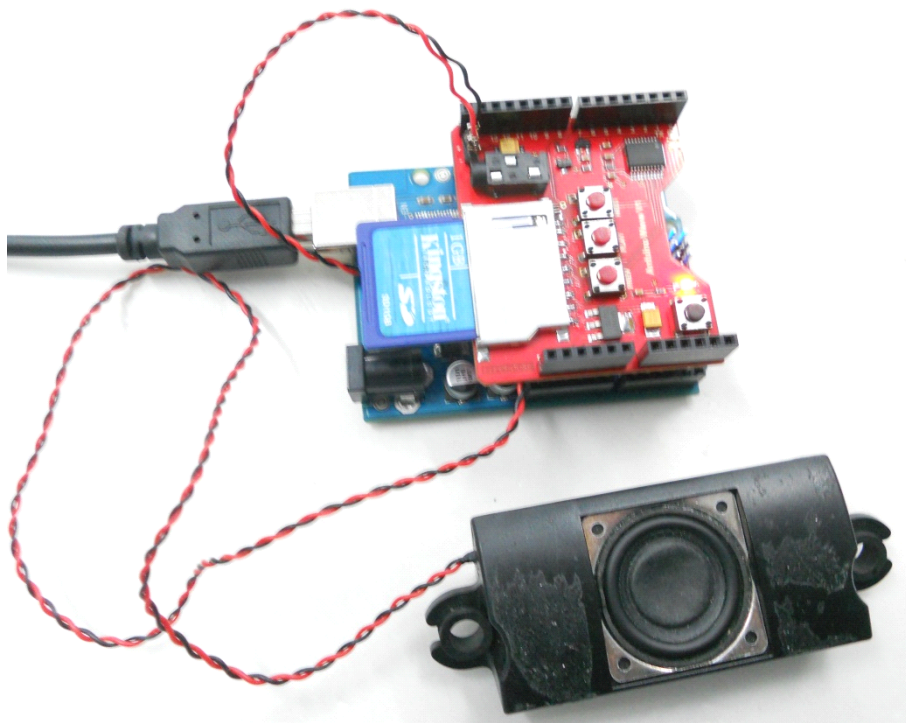
Just plug in this module in Arduino. It is compatible with all Arduino series. There are two audio output interfaces on this module: PWM OUTPUT and ADC OUTPUT.



Just note that while you connect your earphone, your phone head might not match the socket very well.

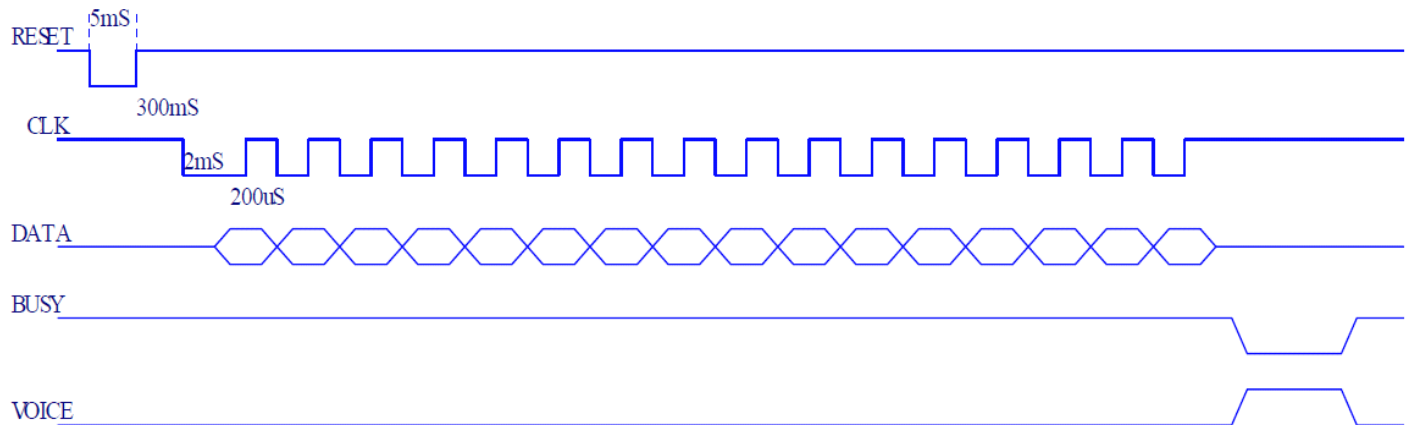


Connect speaker via PWM output interface:



Protocol and Command

Arduino sends command via two pins: CLK (Arduino A5) and DAT (Arduino A4). The communication protocol is like I2C, but not standard I2C.



Send the file name to play the audio. If you need to play file 0005.AD4, send data 0x0005.

To control voice volumn and others, send the following command:

CODE	FUNCTION	DESCRIPTION
FFF0H~FFF7H	VOICE VOLUME ADJUSTMENT	THE VOLUME CAN BE ADJUSTED DURING PLAY OR STAND BY STATUS. FFF0H IS MIN, FFF7H IS MAX, TOTAL 8 LEVEL .
FFFEH	PALY/PAUSE	PLAY/PAUSE THE VOICE IN THE ADDRESS
FFFFH	STOP	STOP TO PLAY THE VOICE

The default volume is the maximum. FFF0H is mute. Volume can be adjusted in play or stop status.

Code

[Click here to download the sample code:](#)

```
int RST = 3; // reset pin
int CLK = 9; // clock pin
int DAT = 8; // data pin

void setup() {

  pinMode(RST, OUTPUT);
  pinMode(CLK, OUTPUT);
  pinMode(DAT, OUTPUT);

  digitalWrite(RST, HIGH);
  digitalWrite(CLK, HIGH);
  digitalWrite(DAT, HIGH);

  digitalWrite(RST, LOW);
  delay(5);
  digitalWrite(RST, HIGH);
  delay(300);
}

void loop() {
```

```
send(0x0000); //play file 0000
delay(10000); //delay 10 seconds
```

```
send(0x0001); //play file 0001
delay(10000);
```

```
send(0x0002); //play file 0002
delay(10000);
```

```
send(0xffff0); //set voice volumn to 0 (turn off)
delay(3000);
```

```
send(0xffff4); //set voice volumn to 4
delay(3000);
```

```
send(0xffff7); //set voice volumn to 7
delay(3000);
```

```
send(0xffffe); // pause
delay(5000);
send(0xffffe); //play
```

```
while(1);
}
```

void send(int data) // This function is to send data to this module.

```
{
  digitalWrite(CLK, LOW);
  delay(2);
  for (int i=15; i>=0; i--)
  {
    delayMicroseconds(50);
    if((data>>i)&0x0001 >0)
    {
      digitalWrite(DAT, HIGH);
    }
    else
    {
      digitalWrite(DAT, LOW);
    }
    delayMicroseconds(50);
    digitalWrite(CLK, HIGH);
    delayMicroseconds(50);

    if(i>0)
      digitalWrite(DAT, LOW);
    else
      digitalWrite(DAT, HIGH);
    delayMicroseconds(50);

    if(i>0)
      digitalWrite(CLK, LOW);
    else
      digitalWrite(CLK, HIGH);
  }

  delay(20);
}
```

Tips: the code is not optimized. Any improving work on the code is encouraged.

Upload the code to Arduino and then press RESTE button (the black on) on the module. You can also try the red buttons. They are NEXT, Play/Pause and PREVIOUS.

Now you could enjoy your music.

Known Problems and Solution

Earphone socket is not on the best place. For big earphone head, it is very hard to plug in. We have risen the socket so most of the head we see can be plugged in. We will improve the design in next generation.

Hardware I2C pins are occupied. It is not a big problem. [Software I2C](#) can be used if I2C is need for your project.

Dimensions (Assemble)

L x W x H: 44mm x 51mm x 24mm

