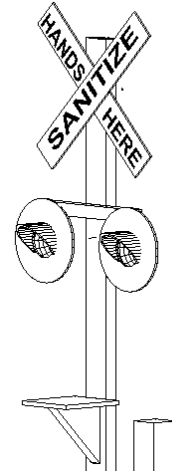
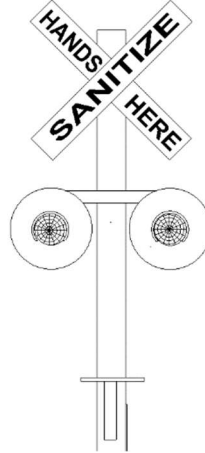




The Sanitize Stations



System description 04/14/2021

There were 10 Sanitize stations built for the following locations: G-Scale, Ticket Office, Caboose 308, RPO, Gift Shop, Bathroom, Platform 1, Platform 2, Platform 3 and, Platform 4.

. **DO NOT UPLUG** the small connectors to turn the units off!



These connectors were installed to ease construction and were not meant to be used often.

To turn the signals off, unplug the power supplies in the **AMMO Boxes** from the 110v outlet. This will turn off the Bell Sounds module near car 100 on Platform 2.

The Station by the bathrooms will automatically turn off when the bathrooms are powered down.

The Station by the ticket booth can be turned off using the christmas lights switch on the western wall inside of the Ticket Office.

The 12 volt power supply and electronic flasher circuit are housed in an ammo box.

The RPO and gift shop share one flasher system in one ammo box located on the north end of the gift shop car.

Platforms 3 and 4 share one flasher system in one ammo box located under platform 3.



The other stations have their own ammo boxes except for the Ticket Office and the Bathrooms. For the Ticket Office, the parts are mounted on the post above the Station. For the Bathrooms, the parts are mounted inside the storage compartment.

To turn the flashing lights on and off, the power plugs must be plugged in/and removed. The Ticket office Sanitize Station is plugged into the Xmas lights circuit and can be turn on and off by the wall switch in the office.

The timer circuit uses either a PICAXE 08M2 or ATtiny13A microprocessor chip. The chips are programmed to output pulses for .95 to 1.0 seconds on two of its outputs. These outputs drive an H-Bridge chip to supply the current needed to drive the red auto LED lamps.

The LED lamps are wired in parallel, anode to cathode. Only two wires are needed to drive both lamps.

The L298N H-Bridge module provides a 5 Volt output to supply power to the microprocessor chip as well as switching the 12 volts to the red 1156 LED lamps. See Appendix 1 for the logic diagram.

Adding audio

An audio sound track was added to one of the stations. The audio track is an 8 hr sound track of railroad crossing sounds. The sound track comes from a DF Player. This is an mp3 player that will drive a speaker directly. In order to supply the 5v power to the audio module, the square wave that drives the LED blinking lights was fed to a full wave bridge rectifier AC inputs. The DC output supplies power to the 78M05 three terminal regulator. A SPDT momentary toggle switch is used to adjust the volume. Grounding the ADKEY_1, pin 12 starts the audio track on power up.

One unit was made and it is mounted on the Sanitize Station on Platform 2.

The audio track is an 8 hours of railroad crossing bells and horn sounds.

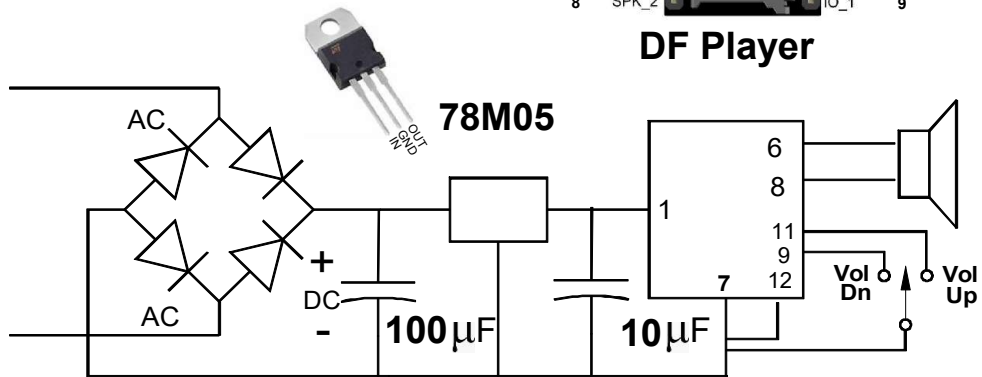


Schematic



1	VCC	BUSY	16
2	RX	USB -	15
3	TX	USB +	14
4	DAC_R	ADKEY_2	13
5	DAC_L	ADKEY_1	12
6	SPK_1	IO_2	11
7	GND	GND	10
8	SPK_2	IO_1	9

DF Player



The square wave signal which drives the flashing lights (LEDs) is used to supply DC power to the audio player. A full wave bridge rectifier converts the AC signal to DC. The two capacitors and the three terminal regulator supply the 5 VDC required by the DF player.

A splitter cable is used to get power to the audio module.





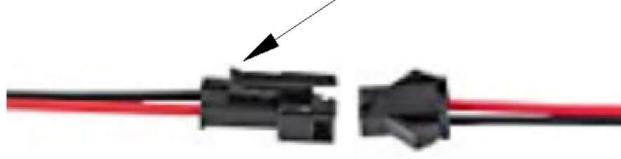
Construction Notes:

The cable from the Sanitize Stations and the ammo boxes have a connector on each end for ease of storage.



To disconnect the cable at the Station, press the release clip as shown below.

Press to release



The connectors on the ammo boxes just pull out. There is no locking devices,



The cable from the ammo box, at Platform 3 to the Station on Platform 4 is in a conduit to protect it from the weed-wacker.

There are no ammo boxes for the Stations at the Bathrooms or the Ticket Office. They have only the one latching connector.



Appendix 1 Locations



G-Scale - Plug in AC cable at east side of layout.



Platform 3 – Plug in AC cable
Platform 4 – Works with platform 3



Platform 1 – Plug in AC cable



Gift Shop – Plug in AC cable on North end of Gift Shop car.
RPO – Works with gift shop

Picture unavailable at this time

Caboose 308 – Plug in AC cable to power cord behind the fence

Ticket office - Turn on xmas light switch on west wall.

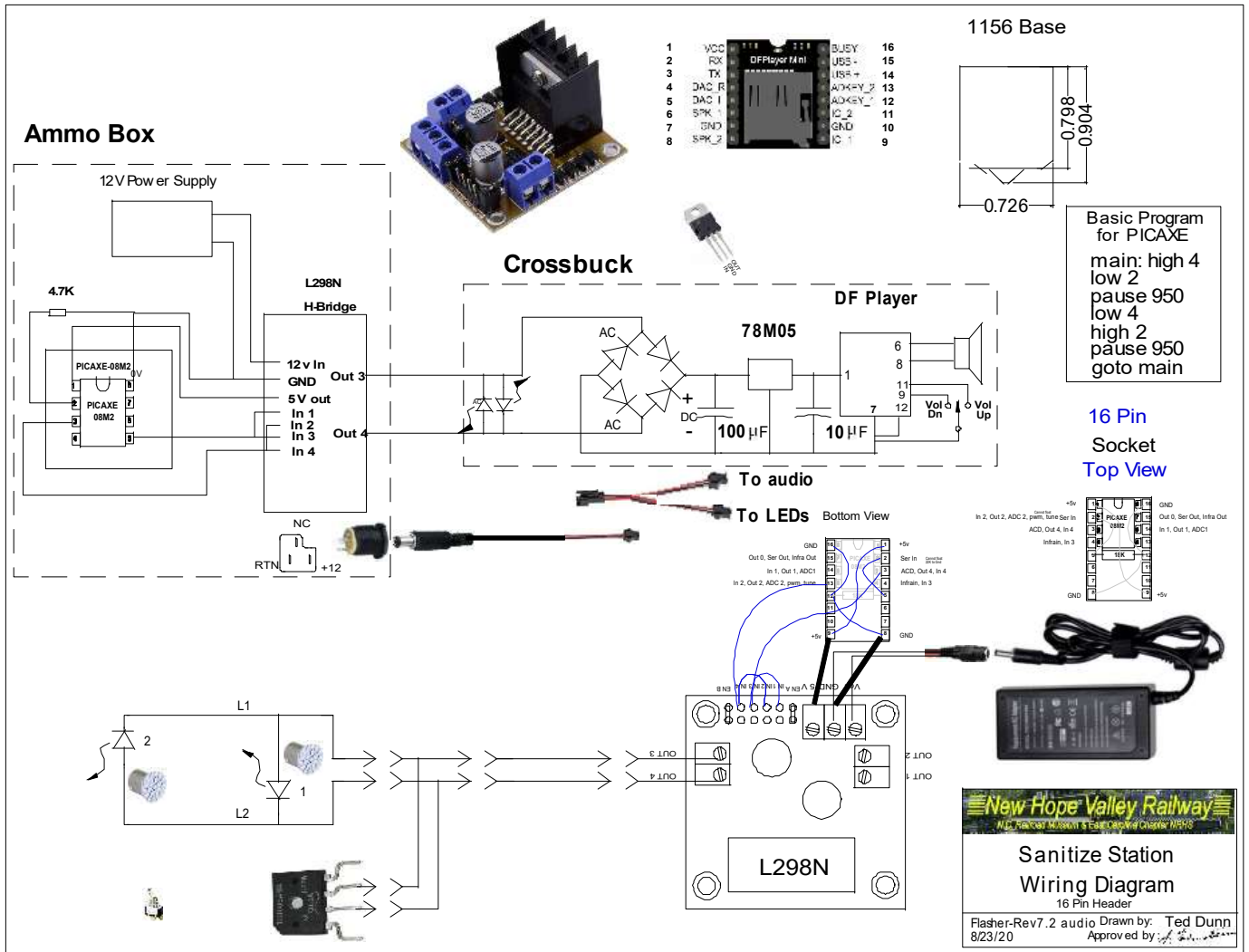
Bathroom – Will turn on automatically when bathrooms are in use.



Platform 2 – Plug in AC cable



Appendix 2 Schematic





Appendix 3 L298N Motor Drive Bridge logic diagram

The logic table and diagrams show how only two wires are needed to control the LED lamps. Fig. 4 lights the lower LED and Fig. 5 lights the upper LED. In Figs. 1,2 and 3 the LEDs are off.

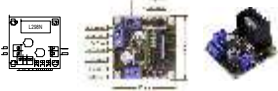
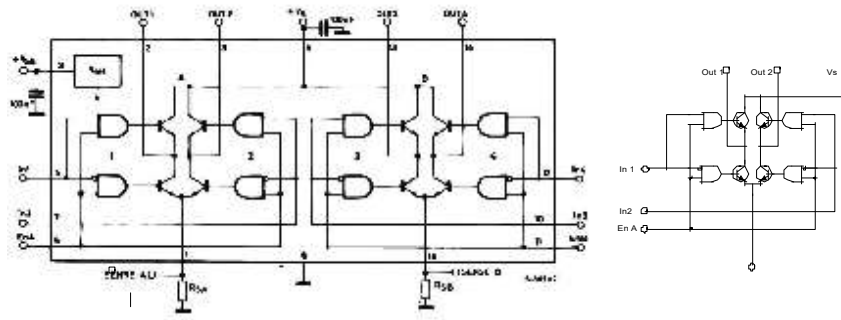
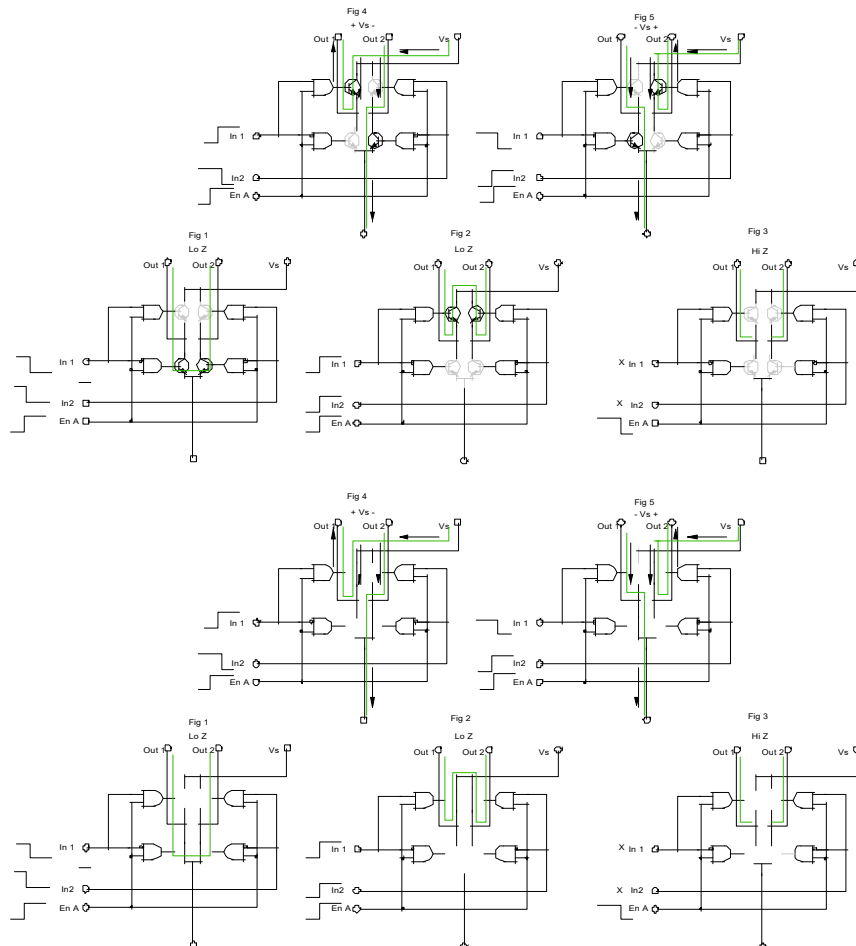


	Fig 1	Fig 2	Fig 3	Fig 4	Fig 5
In1	0	1	X	1	0
In2	0	1	X	0	1
En A	1	1	0	1	1
Out 1 to Out 2	Lo Z	Lo Z	Hi Z	+ to -	- to +





Appendix 4 DF Player



Introduction

The [DFPlayer Mini MP3 Player For Arduino](#) is a small and low price MP3 module with an simplified output directly to the speaker. The module can be used as a stand alone module with attached battery, speaker and push buttons or used in combination with an [Arduino UNO](#) or any other with RX/TX capabilities.

Specification

- supported sampling rates (kHz): 8/11.025/12/16/22.05/24/32/44.1/48
- 24 -bit DAC output, support for dynamic range 90dB , SNR support 85dB
- fully supports FAT16 , FAT32 file system, maximum support 32G of the TF card, support 32G of U disk, 64M bytes NORFLASH
- a variety of control modes, I/O control mode, serial mode, AD button control mode
- advertising sound waiting function, the music can be suspended. when advertising is over in the music continue to play
- audio data sorted by folder, supports up to 100 folders, every folder can hold up to 255 songs
- 30 level adjustable volume, 6 -level EQ adjustable

Application

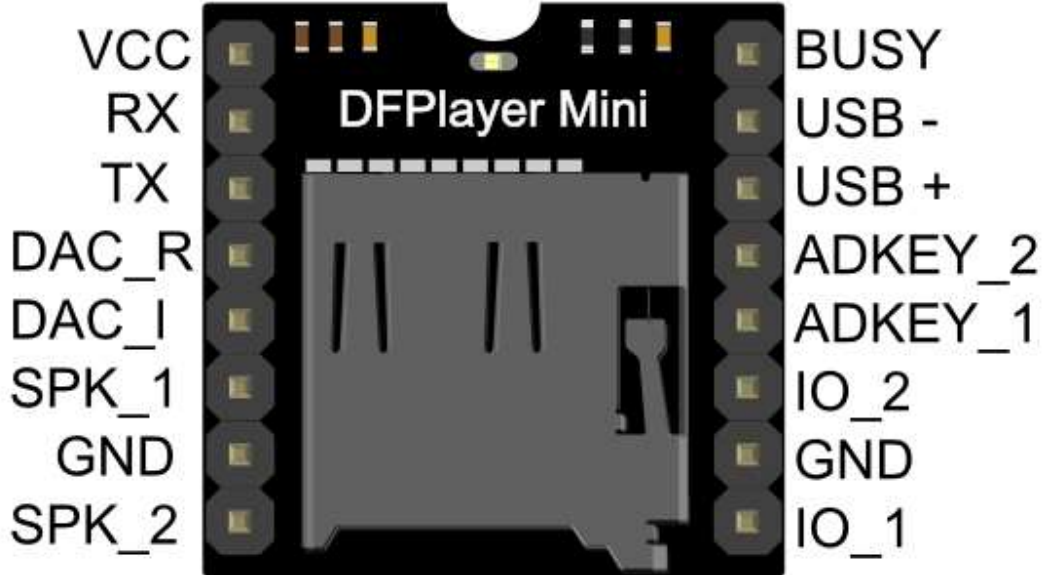
- Car navigation voice broadcast;

≡New Hope Valley Railway≡



- Road transport inspectors, toll stations voice prompts;
- Railway station, bus safety inspection voice prompts;
- Electricity, communications, financial business hall voice prompts;
- Vehicle into and out of the channel verify that the voice prompts;
- The public security border control channel voice prompts;
- Multi-channel voice alarm or equipment operating guide voice;
- The electric tourist car safe driving voice notices;
- Electromechanical equipment failure alarm;
- Fire alarm voice prompts;
- The automatic broadcast equipment, regular broadcast.

Pin Map





Pin	Description	Note
VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
RX	UART serial input	
TX	UART serial output	
DAC_R	Audio output right channel	Drive earphone and amplifier
DAC_L	Audio output left channel	Drive earphone and amplifier
SPK2	Speaker-	Drive speaker less than 3W
GND	Ground	Power GND
SPK1	Speaker+	Drive speaker less than 3W
IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
GND	Ground	Power GND
IO2	Trigger port 2	Short press to play next (long press to increase volume)
ADKEY1	AD Port 1	Trigger play first segment
ADKEY2	AD Port 2	Trigger play fifth segment
USB+	USB+ DP	USB Port
USB-	USB- DM	USB Port
BUSY	Playing Status	Low means playing \High means no

Work Mode

1. Serial Mode

Support for asynchronous serial communication mode via PC serial sending commands Communication Standard:9600 bps Data bits :1 Checkout :none Flow Control :none

- Instruction Description

New Hope Valley Railway



Format:	SS	VER	Len	CMD	Feedback	para1	para2	checksum	SO
SS	Start bit 0x7E		Each command feedback begin with \$, that is 0x7E						
VER	Version		Version Information						
Len	the number of bytes after “Len”		Checksums are not counted						
CMD	Commands		Indicate the specific operations, such as play / pause, etc.						
Feedback	Command feedback		If need for feedback, 1: feedback, 0: no feedback						
para1	Parameter 1		Query high data byte						
para2	Parameter 2		Query low data byte						
checksum	Checksum		Accumulation and verification [not include start bit \$]						
SO	End bit		End bit 0xEF						

For example, if we specify play NORFLASH, you need to send: 7E FF 06 09 00 00 04 FF DD EF
Data length is 6, which are 6 bytes [FF 06 09 00 00 04]. Not counting the start, end, and verification.

- Serial Control Cmd

CMD	Function Description	Parameters(16 bit)
0x01	Next	
0x02	Previous	
0x03	Specify tracking(NUM)	0-2999
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	0-30
0x07	Specify EQ(0/1/2/3/4/5)	Normal/Pop/Rock/Jazz/Classic/Base
0x08	Specify playback mode (0/1/2/3)	Repeat/folder repeat/single repeat/ random
0x09	Specify playback source(0/1/2/3/4)	U/TF/AUX/SLEEP/FLASH
0x0A	Enter into standby – low power loss	
0x0B	Normal working	
0x0C	Reset module	
0x0D	Playback	
0x0E	Pause	
0x0F	Specify folder to playback	1~10(need to set by user)
0x10	Volume adjust set	{DH= 1:Open volume adjust } {DL: set volume gain 0~31}
0x11	Repeat play	{1:start repeat play} {0:stop play}



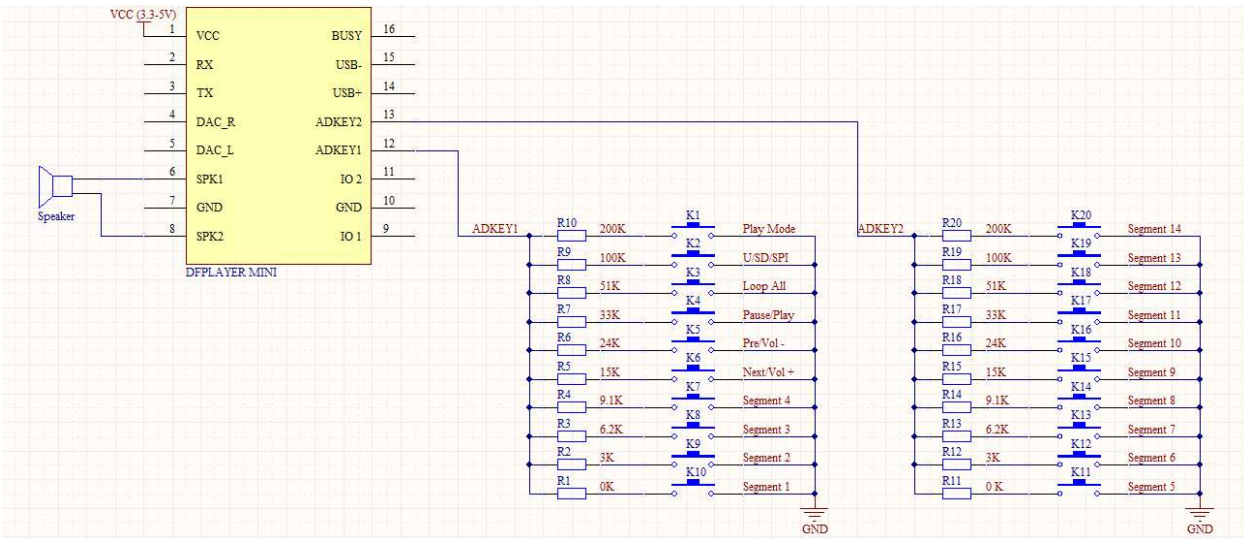
- Serial Query Cmd

Commands	Function Description	Parameters(16 bit)
0x3C	STAY	
0x3D	STAY	
0x3E	STAY	
0x3F	Send initialization parameters	0 - 0x0F(each bit represent one device of the low-four bits)
0x40	Returns an error, request retransmission	
0x41	Reply	
0x42	Query the current status	
0x43	Query the current volume	
0x44	Query the current EQ	
0x45	Query the current playback mode	
0x46	Query the current software version	
0x47	Query the total number of TF card files	
0x48	Query the total number of U-disk files	
0x49	Query the total number of flash files	
0x4A	Keep on	
0x4B	Queries the current track of TF card	
0x4C	Queries the current track of U-Disk	
0x4D	Queries the current track of Flash	

2. AD KEY Mode

We use the AD module keys, instead of the traditional method of matrix keyboard connection, it is to take advantage of increasingly powerful MCU AD functionality, Our module default configuration 2 AD port, 20 key resistance distribution.

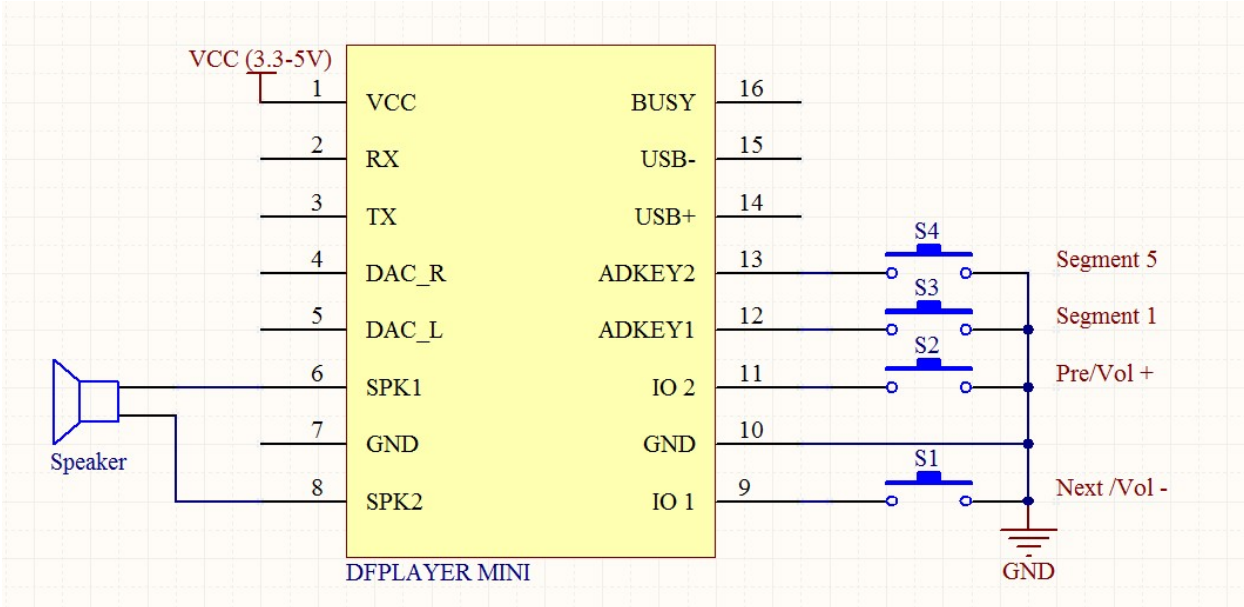
- Refer diagram



3. I/O Mode

Here comes the most simple way to use this module.

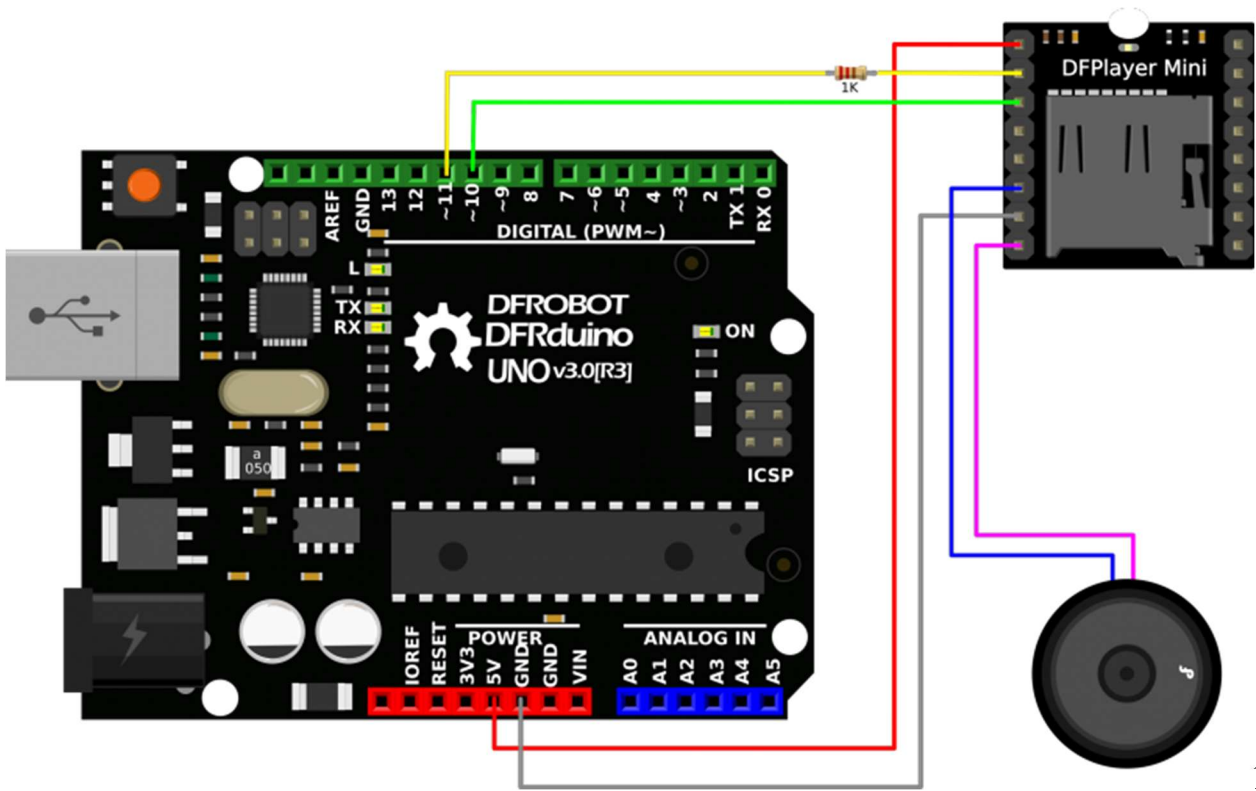
- Refer diagram



note: short

time press means pre/next, and long time press means vol-, vol +

Connection Diagram



Note: For simple use, the upper diagram is ready, but if you find the noise is quite loud, then you could attach an 1K resistor to the TX pin.

Copy your mp3 into your micro SD card

NOTE: The order you copy the mp3 into micro SD card will affect the order mp3 played, which means play(1) function will play the first mp3 copied into micro SD card.

For Mac User

NOTE: If you are using Mac OS X to copy the mp3, the file system will automatically add hidden files like: "._0001.mp3" for index, which this module will handle as valid mp3 files. It is really annoying. So you can run following command in terminal to eliminate those files.

```
dot_clean /Volumes/<SDVolumeName>
```

Please replace the to the volume name of your SD card.

Sample Code

We've created an Arduino library for DFPlayer Mini to simplify the method for you to make it work. Connect the hardware as the picture above shown and play with the sample code. You can download the latest library here: [DFRobotDFPlayerMini](#).

Sample code "GetStarted", switching to next song every 3 seconds



```
/*
DFPlayer - A Mini MP3 Player For Arduino
<https://www.dfrobot.com/index.php?route=product/product&product_id=1121>

*****
This example shows the basic function of library for DFPlayer.

Created 2016-12-07
By [Angelo qiao](Angelo.qiao@dfrobot.com)

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/******Notice and Trouble shooting*****
1.Connection and Diagram can be found here
<https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
*****/

#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }
  Serial.println(F("DFPlayer Mini online.));

  myDFPlayer.volume(10); //Set volume value. From 0 to 30
  myDFPlayer.play(1); //Play the first mp3
}

void loop()
{
  static unsigned long timer = millis();

  if (millis() - timer > 3000) {
    timer = millis();
    myDFPlayer.next(); //Play next mp3 every 3 second.
  }
}
```


≡≡≡New Hope Valley Railway≡≡≡



```
}

if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message
    from DFPlayer to handle different errors and states.
}
}

void printDetail(uint8_t type, int value){
    switch (type) {
        case TimeOut:
            Serial.println(F("Time Out!"));
            break;
        case WrongStack:
            Serial.println(F("Stack Wrong!"));
            break;
        case DFPlayerCardInserted:
            Serial.println(F("Card Inserted!"));
            break;
        case DFPlayerCardRemoved:
            Serial.println(F("Card Removed!"));
            break;
        case DFPlayerCardOnline:
            Serial.println(F("Card Online!"));
            break;
        case DFPlayerPlayFinished:
            Serial.print(F("Number:"));
            Serial.print(value);
            Serial.println(F(" Play Finished!"));
            break;
        case DFPlayerError:
            Serial.print(F("DFPlayerError:"));
            switch (value) {
                case Busy:
                    Serial.println(F("Card not found"));
                    break;
                case Sleeping:
                    Serial.println(F("Sleeping"));
                    break;
                case SerialWrongStack:
                    Serial.println(F("Get Wrong Stack"));
                    break;
                case CheckSumNotMatch:
                    Serial.println(F("Check Sum Not Match"));
                    break;
                case FileIndexOut:
                    Serial.println(F("File Index Out of Bound"));
                    break;
                case FileMismatch:
                    Serial.println(F("Cannot Find File"));
                    break;
                case Advertise:
                    Serial.println(F("In Advertise"));
                    break;
                default:
                    break;
            }
    }
    break;
}
```



```

    default:
      break;
  }
}

```

Sample code "FullFunction", including all the functions. Please read the comments and documents in detail

```

/*****
DFPlayer - A Mini MP3 Player For Arduino
<https://www.dfrobot.com/index.php?route=product/product&product_id=1121>

*****
This example shows the all the function of library for DFPlayer.

Created 2016-12-07
By [Angelo qiao](Angelo.qiao@dfrobot.com)

GNU Lesser General Public License.
See <http://www.gnu.org/licenses/> for details.
All above must be included in any redistribution
*****/

/*****Notice and Trouble shooting*****/
1.Connection and Diagram can be found here
<https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
2.This code is tested on Arduino Uno, Leonardo, Mega boards.
*****/

#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with
mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }
  Serial.println(F("DFPlayer Mini online.));

  myDFPlayer.setTimeout(500); //Set serial communicaiton time out 500ms

```

≡New Hope Valley Railway≡



```
//-----Set volume-----
myDFPlayer.volume(10); //Set volume value (0~30).
myDFPlayer.volumeUp(); //Volume Up
myDFPlayer.volumeDown(); //Volume Down

//-----Set different EQ-----
myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
// myDFPlayer.EQ(DFPLAYER_EQ_POP);
// myDFPlayer.EQ(DFPLAYER_EQ_ROCK);
// myDFPlayer.EQ(DFPLAYER_EQ_JAZZ);
// myDFPlayer.EQ(DFPLAYER_EQ_CLASSIC);
// myDFPlayer.EQ(DFPLAYER_EQ_BASS);

//-----Set device we use SD as default-----
// myDFPlayer.outputDevice(DFPLAYER_DEVICE_U_DISK);
myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);
// myDFPlayer.outputDevice(DFPLAYER_DEVICE_AUX);
// myDFPlayer.outputDevice(DFPLAYER_DEVICE_SLEEP);
// myDFPlayer.outputDevice(DFPLAYER_DEVICE_FLASH);

//-----Mp3 control-----
// myDFPlayer.sleep(); //sleep
// myDFPlayer.reset(); //Reset the module
// myDFPlayer.enabledDAC(); //Enable On-chip DAC
// myDFPlayer.disableDAC(); //Disable On-chip DAC
// myDFPlayer.outputSetting(true, 15); //output setting, enable the output and set the
gain to 15

//-----Mp3 play-----
myDFPlayer.next(); //Play next mp3
delay(1000);
myDFPlayer.previous(); //Play previous mp3
delay(1000);
myDFPlayer.play(1); //Play the first mp3
delay(1000);
myDFPlayer.loop(1); //Loop the first mp3
delay(1000);
myDFPlayer.pause(); //pause the mp3
delay(1000);
myDFPlayer.start(); //start the mp3 from the pause
delay(1000);
myDFPlayer.playFolder(15, 4); //play specific mp3 in SD:/15/004.mp3; Folder
Name(1~99); File Name(1~255)
delay(1000);
myDFPlayer.enableLoopAll(); //loop all mp3 files.
delay(1000);
myDFPlayer.disableLoopAll(); //stop loop all mp3 files.
delay(1000);
myDFPlayer.playMp3Folder(4); //play specific mp3 in SD:/MP3/0004.mp3; File
Name(0~65535)
delay(1000);
myDFPlayer.advertise(3); //advertise specific mp3 in SD:/ADVERT/0003.mp3; File
Name(0~65535)
delay(1000);
myDFPlayer.stopAdvertise(); //stop advertise
delay(1000);
myDFPlayer.playLargeFolder(2, 999); //play specific mp3 in SD:/02/004.mp3; Folder
Name(1~10); File Name(1~1000)
```

≡ New Hope Valley Railway ≡



```
delay(1000);
myDFPlayer.loopFolder(5); //loop all mp3 files in folder SD:/05.
delay(1000);
myDFPlayer.randomAll(); //Random play all the mp3.
delay(1000);
myDFPlayer.enableLoop(); //enable loop.
delay(1000);
myDFPlayer.disableLoop(); //disable loop.
delay(1000);

//----Read information----
Serial.println(myDFPlayer.readState()); //read mp3 state
Serial.println(myDFPlayer.readVolume()); //read current volume
Serial.println(myDFPlayer.readEQ()); //read EQ setting
Serial.println(myDFPlayer.readFileCounts()); //read all file counts in SD card
Serial.println(myDFPlayer.readCurrentFileNumber()); //read current play file number
Serial.println(myDFPlayer.readFileCountsInFolder(3)); //read fill counts in folder
SD:/03
}

void loop()
{
  static unsigned long timer = millis();

  if (millis() - timer > 3000) {
    timer = millis();
    myDFPlayer.next(); //Play next mp3 every 3 second.
  }

  if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message
    from DFPlayer to handle different errors and states.
  }
}

void printDetail(uint8_t type, int value){
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
      break;
    case WrongStack:
      Serial.println(F("Stack Wrong!"));
      break;
    case DFPlayerCardInserted:
      Serial.println(F("Card Inserted!"));
      break;
    case DFPlayerCardRemoved:
      Serial.println(F("Card Removed!"));
      break;
    case DFPlayerCardOnline:
      Serial.println(F("Card Online!"));
      break;
    case DFPlayerPlayFinished:
      Serial.print(F("Number:"));
      Serial.print(value);
      Serial.println(F(" Play Finished!"));
      break;
    case DFPlayerError:
```



```
Serial.print(F("DFPlayerError:"));
switch (value) {
  case Busy:
    Serial.println(F("Card not found"));
    break;
  case Sleeping:
    Serial.println(F("Sleeping"));
    break;
  case SerialWrongStack:
    Serial.println(F("Get Wrong Stack"));
    break;
  case CheckSumNotMatch:
    Serial.println(F("Check Sum Not Match"));
    break;
  case FileIndexOut:
    Serial.println(F("File Index Out of Bound"));
    break;
  case FileMismatch:
    Serial.println(F("Cannot Find File"));
    break;
  case Advertise:
    Serial.println(F("In Advertise"));
    break;
  default:
    break;
}
break;
default:
  break;
}
}
```

NOTE: The folder name needs to be mp3, placed under the SD card root directory, and the mp3 file name needs to be 4 digits, for example, "0001.mp3", placed under the mp3 folder. If you want to name it in Both English and Chinese, you can add it after the number, for example, "0001hello.mp3" or "0001 后来.mp3".

➔ Go Shopping [DFPlayer - A Mini MP3 Player](#)

Category: [DFRobot](#) > [Sensors & Modules](#) > [Modules category: Product Manual category: DFR Series category: Module category: Diagram category: DFRobot](#)