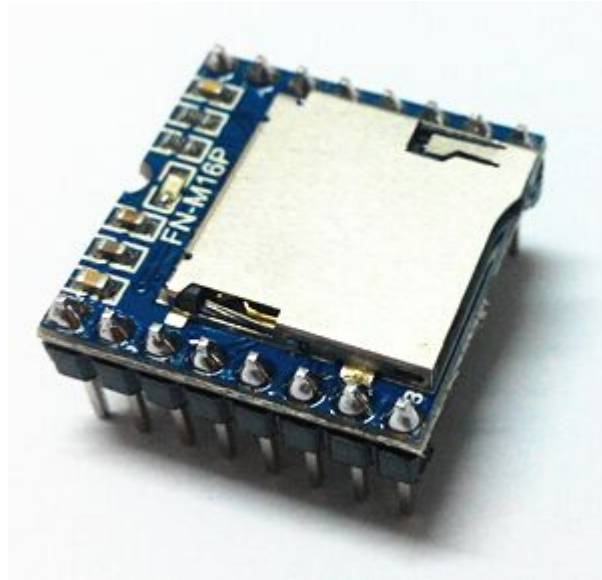


FN-M16P Embedded MP3 Audio Module

Datasheet



Contents

1. Overview.....	2
1.1. Brief Introduction.....	2
1.2. Product Features.....	2
1.3. Technical Parameters.....	2
2. Pin Configuration and Summary.....	2
3. Serial Communication Protocol.....	3
3.1. Serial Commands Format.....	3
3.2. Serial Commands.....	4
3.3. Detailed Explanation of Serial Commands.....	5
3.4. Examples of Sending Serial Commands.....	11
3.5. Examples of Serial Program.....	13
4. ADkey Control.....	15
5. Application Circuits.....	16
6. Features of GPIO.....	18
7. PCB Size.....	18

1. Overview

1.1. Brief Introduction

FN-M16P module is a serial MP3 module that is with a perfect integrated MP3 and WMV decoder chip. It provides micro SD card driver, and supports FAT16 and FAT32 file systems. It is able to play back specified sound files and realize other functions through simple serial commands. In the mean time, this module supports AD key control mode that facilitates users to develop their jobs in some simple applications. Without the cumbersome underlying operating, easy to use, stable and reliable are the most important features of this module.

1.2. Product Features

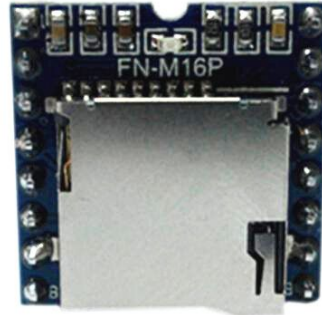
- Supports MP3 and WAV decoding.
- Supports FAT16 and FAT32 file system.
- 24-bit DAC output and supports dynamic range 90dB and SNR 85dB.
- Supports AD key control mode and UART RS232 serial control mode.
- Supports maximum 32GB micro SD card and 32GB USB flash drive.
- Audio files are sorted by folders; supports up to 99 folders, and each folder can be assigned to 255 sound files.
- Supports inter-cut advertisements.
- Supports playback of specifying folders.
- Support random playback.
- Built-in 3W amplifier that can direct drive a 3W/8Ohm speaker.
- 30 levels adjustable volume, and 6 levels adjustable EQ.

1.3. Technical Parameters

Item	Description
MP3 Audio Format	Supports 11172-3 and ISO13813-3 layer3 audio decoding
	Supports sampling rate (KHZ):8/11.025/12/16/22.05/24/32/44.1/48
	Supports Normal, Jazz, Classic, Pop, Rock, etc.
USB Port	Standard USB 2.0
UART Port	Standard Serial; TTL Level; Baud rate adjustable(default baud rate is 9600)
Working Voltage	DC3.3~5.0V; Typical:DC4.2V
Rated Current	<15mA(without USB flash drive)
Operating Temperature	-40~+80℃
Humidity	5% ~95%

2. Pin Configuration and Summary

1	VCC	BUSY	16
2	RX	USB-	15
3	TX	USB+	14
4	DAC_R	ADKEY2	13
5	DAC_L	ADKEY1	12
6	SPK+	I/O2	11
7	GND	GND	10
8	SPK-	I/O1	9



No	Pin	Description	Note
1	VCC	DC3.2~5.0V	
2	RX	UART serial input	
3	TX	UART serial output	
4	DAC_R	Audio output right channel	Drive earphone and amplifier
5	DAC_L	Audio output left channel	Drive earphone and amplifier
6	SPK2	Speaker-	Drive speaker less than 3W
7	GND	Ground	Power GND
8	SPK1	Speaker+	Drive speaker less than 3W
9	IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
10	GND	Ground	Power GND
11	IO2	Trigger port 2	Short press to play next (long press to increase volume)
12	ADKEY1	AD Port 1	Trigger to play first segment
13	ADKEY2	AD Port 2	Trigger to play fifth segment
14	USB+	USB+ DP	USB Port(connected with a USB flash or connected to PC)
15	USB-	USB- DM	USB Port(connected with a USB flash or connected to PC)
16	Busy	Indicator Pin	Low level when working, and high level when standby

3. Serial Communication Protocol

Serial port control mode is a common communication in the control field, based on which We conducted an industrial level of optimization by adding frame checksum, retransmission, error handling, and other measures to significantly strengthen the stability and reliability of communication. The default baud rate is 9600.

3.1. Serial Commands Format

Supports asynchronous serial communication mode, via which accept serial commands sent by upper PC.



Communication Standard:9600 bps

Data bits :1

Checkout :none

Flow Control :none

Format 6 \$S VER Len CMD Feedback para1 para2 checksum \$O		
\$S	Start byte 0x7E	Each command feedback begins with \$, which is 0x7E
VER	Version	Version Information(default 0xFF)
Length	Number of bytes from COMMAND through to Check_LSB (typically 0x06)	Checksum not counted
CMD	Command byte	Means the specific operations, such as play / pause, etc.
Feedback	Command feedback	0x01: Feedback-send confirmation back to MCU; 0x00: No feedback
Param_MSB	Parameter	Most significant byte of parameter
Param_LSB	Parameter	Least significant byte of parameter
Check_MSB	Checksum	Most significant byte of checksum
Check_LSB	Checksum	Least significant byte of checksum
\$O	End byte	0xEF

For example, if we specify playback of SD card, we need to send the command “7E FF 06 09 00 00 02 FF F0 EF”. The data length is 6, and the 6 bytes are “FF 06 09 00 00 02”. Start byte, end byte and checksum are not counted.

3.2. Serial Commands

3.2.1. Commands Direct Sent(No need returned data)

CMD	Function Description	Parameters
0x01	Play Next	
0x02	Play Previous	
0x03	Specify playback of a track	1-2999
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	Volume level:0-30
0x07	Specify EQ(0/1/2/3/4/5)	0:Normal/1:Pop/2:Rock/3:Jazz/4:Classic/5:Bass
0x08	Specify single repeat playback	1-2999
0x09	Specify playback of a device(0/1)	0:USB/1:SD
0x0A	Enter into standby – low power consumption	
0x0B	N/A(Reserved)	
0x0C	Reset	
0x0D	Playback	
0x0E	Pause	
0x0F	Specify playback of a folder	01-99(need to set by user)
0x10	Audio amplification setting	DH=1:amplifying on, DL:set gain 0-31
0x11	All repeat playback	1:start all repeat playback; 0:stop playback
0x12	Specify playback of folder named “MP3”	
0x13	Inter cut an advertisement	

0x14	Specify playback 1 of 3000 tracks in a folder	Supports 15 folders only(01-15)
0x15	Stop playing inter-cut advertisement and go back to play the music interrupted	
0x16	Stop playback	
0x17	Specify repeat playback of a folder	See 3.3.12
0x18	Random playback	See 3.3.13
0x19	Specify playback of current track	See 3.3.14
0x1A	Turn on and turn off DAC	See 3.3.15

3.2.2.Query Parameters of System

CMD	Function Description	Parameters
0x3C	N/A(Reserved)	
0x3D	N/A(Reserved)	
0x3E	N/A(Reserved)	
0x3F	Send initialization parameters	0-0x0F
0x40	Returns an error, request retransmission	
0x41	Feedback from module	
0x42	Query current status	
0x43	Query current volume	
0x44	Query current EQ	
0x45	N/A(Reserved)	
0x46	N/A(Reserved)	
0x47	Query total file numbers of USB flash disk	
0x48	Query total file numbers of micro SD Card	
0x49	N/A(Reserved)	
0x4A	N/A(Reserved)	
0x4B	Query current track of USB flash disk	
0x4C	Query current track of micro SD Card	
0x4D	N/A(Reserved)	
0x4E	Query total file numbers of a folder	
0x4F	Query total folder numbers of the storage device	

3.3. Detailed Explanation of Serial Commands

3.3.1.Specify Playback of a Track(under the root directory of a storage device)

The available selective tracks is from 1st to 2999th under the root directory of the storage device. Actually it can support more, but if we make it support more, the operation speed will become slow. Usually most of applications do not need to support much more files.

Here the module can not specify the exact file name like 001.mp3/0001.mp3 or playback, but it works according to

the indexed sequence when you copy the sound files from PC to micro SD or USB flash disk. So when it receives a command to play the track(sound file)"07", it will play the 7th track according to the indexed sequence instead of the sound file named "007.mp3"(maybe it plays "007.mp3" if it is just ranked the 7th by the indexed sequence).

1).For example, select the first song played, and send the command 7E FF 06 03 00 00 01 FE F7 EF

7E --- Start byte

FF --- Version Information

06 --- Data length (checksum not included)

03 --- Actual command(specify playback of a track)

00 --- 0x01: need feedback, 0x00:no need feedback

00 --- Most significant byte of the track [DH]

01 --- Least significant byte of the track [DL]

FE --- Most significant byte of checksum

F7 --- Least significant byte of checksum

EF --- End byte 0xEF

2).Regarding selection, if choose the 100th song(track), firstly convert 100 to hexadecimal. It is double-byte by default, i.e. 0x0064. DH = 0x00; DL = 0x64

3).If you choose to play the 1000th song(track), firstly convert 1000 to hexadecimal. It is double-byte, i.e. 0x03E8. DH = 0x03; DL = 0xE8

4).And so on in the same way to the other operations, as in the embedded area hexadecimal is the most convenient operation method.

3.3.2.Specify Volume

1).Our system power-on default volume is level 30, if you want to set the volume, then directly send the corresponding commands.

2).For example, if specify the volume to level 15, send the command 7E FF 06 06 00 00 0F FF D5 EF.

3).DH = 0x00; DL = 0x0F, 15 is converted to hexadecimal 0x000F.

3.3.3. Specify Playback of a Device

1).This module supports two types of playback devices by default. The device must be on-line, so it can be specified playback. The software will automatically detect if a device is on-line. No need user's attention.

2).Refer to the table as below to select the appropriate command to send.

3).It will automatically enter the Suspend status after specifying a device, waiting for the user to specify a track to play. It will take about 200ms from specifying device to the module initialize file information. Please wait for 200ms and then send the specified command to play a track.

Specify playback device –USB flash disk	7E FF 06 09 00 00 01 xx xx EF	xx xx represents checksum
Specify playback device –micro SD card	7E FF 06 09 00 00 02 xx xx EF	

3.3.4. Specify Playback of a Folder(Specify Playback of a Track in a Folder)

Specify 001.mp3 in the folder 01	7E FF 06 0F 00 01 01 xx xx EF
Specify 100.mp3 in the folder 11	7E FF 06 0F 00 0B 64 xx xx EF
Specify 255.mp3 in the folder 99	7E FF 06 0F 00 63 FF xx xx EF

1).Specifying playback of a folder is an extended function. The default folders are named as "01", "11" in this way. In order to be with a better system stability, it is made to support maximum 99 folders and each folder maximum 255 songs supported.

2).For example, if specify to play "100.mp3" in the folder "01", send the command 7E FF 06 0F 00 01 64 xx xx EF

DH: represents the name of the folder, maximum supports 99 folders from 01 - 99.

DL: represents the track, maximum supports 255 tracks from 0x01 to 0xFF.

Please refer to the above set rules for setting tracks.

3).You must specify both the folder and the file name to target a track. Individually specify a folder or specify a file name alone is also possible, but the document management will be worse.

4).The following two images illustrates the naming method of folders and files.



3.3.5. Specify Playback of Folder Named "MP3"

Specify playback of folder named "MP3"	7E FF 06 12 00 00 01 FE E8 EF	Play track "0001" in the folder "MP3"
	7E FF 06 12 00 00 02 FE E7 EF	Play track "0002" in the folder "MP3"
	7E FF 06 12 00 00 FF FD EA EF	Play track "0255" in the folder "MP3"
	7E FF 06 12 00 07 CF FE 13 EF	Play track "1999" in the folder "MP3"
	7E FF 06 12 00 0B B8 FE 26 EF	Play track "3000" in the folder "MP3"

1).We extended such single folder function on the basis of specifying playback of folders. The folder must be named "MP3".

2).Supports maximum 65536 songs(tracks), but the operation speed will get slow along with the folder become large, so we suggest users put no more than 3000 songs(tracks) in this folder.

3).Files need to named as below.



3.3.6. Inter Cut an Advertisement under Folder “ADVERT”

Inter cut ads	7E FF 06 13 00 00 01 FE E7 EF	Inter cut track "0001" in the folder “ADVERT”
	7E FF 06 13 00 00 02 FE E6 EF	Inter cut track "0002" in the folder “ADVERT”
	7E FF 06 13 00 00 FF FD E9 EF	Inter cut track "0255" in the folder “ADVERT”
	7E FF 06 13 00 07 CF FE 12 EF	Inter cut track "1999" in the folder “ADVERT”
	7E FF 06 13 00 0B B8 FE 25 EF	Inter cut track "3000" in the folder “ADVERT”

1). This module supports inter-cut advertisements during playback of a track, so that it can meet some special needs for some applications.

2). After sending the command 0x13, the system will save the IDV3 information of the track being played and pause, then it will play the specified inter-cut track(advertisement). When the inter-cut track is finished, the system will go back and continue to play the track that was interrupted until to the end.

3). The setting method is build a folder named “ADVERT” in the storage device, and put the tracks(ads) you need in the folder. And name the files like “0001.mp3/wav”, 0002.mp3/wav.

4). If you send an inter-cut command when the module is at Pause status or Stop status, it will not work and there will be an returned error information. In the course of an inter-cut, you can continue to inter cut the other tracks(ads). When the last inter-cut track goes to the end, the system still goes back to the IDV3 position saved at the first time.

5). Refer to below on the folder name and files names.



3.3.7. Specify Playback of a Folder That Supports 3000 Tracks

Specify playback 1 of 3000 tracks in a folder	7E FF 06 14 00 10 FF FD D8 EF	Play track "0255" in the specified folder "01"
	7E FF 06 14 00 17 CF FE 01 EF	Play track "1999" in the specified folder "01"
	7E FF 06 14 00 C0 01 FE 26 EF	Play track "0001" in the specified folder "12"
	7E FF 06 14 00 C0 FF FD 28 EF	Play track "0255" in the specified folder "12"
	7E FF 06 14 00 C7 CF FD 51 EF	Play track "1999" in the specified folder "12"

In order to meet some users' needs that each folder is able to manage 3000 tracks, we specially added this command for users to use if in need. It supports 15 folders only.

1). The command byte is 0x14.

2). For example, if we specify playback of track "1999" in the folder "12", we need to send the serial command

7E FF 06 14 00 C7 CF FD 51 EF.

0xC7 and 0xCF are parameter, and the combined data is 0xC7CF, and totally 16 bytes.

The high 4 bytes represent the folder name; C here means the folder "12".

The low 12 bytes represent the file name; 7CF here means the file "1999".

3). Refer to below on folder names and file names.



3.3.8. Command for All Repeat(Loop) Playback

1). We added such a control command 0x11, to meet the needs that some users need repeated playback of all tracks under the root directory of a storage device.

Start all repeat playback	7E FF 06 11 00 00 01 xx xx EF	Repeatedly play all the tracks
Stop all repeat playback	7E FF 06 11 00 00 00 xx xx EF	Stop playing all the tracks

2). During all repeat playback, you can still normally execute the operations Play/Pause, Previous, Next, Volume+/-, EQ and so on.

3). After starting all repeat playback, the module will play all the tracks in the device ceaselessly again and over again until it receives a command for stop or pause.

3.3.9. Command for Single Repeat(Loop) Playback

Start single repeat playback	7E FF 06 08 00 00 01 xx xx EF	Repeatedly play the first track
	7E FF 06 08 00 00 02 xx xx EF	Repeatedly play the second track

1). We added this control command 0x08, to meet the needs that some users need single repeat playback.

2). During single repeat playback, you can still normally execute the operations Play/Pause, Previous, Next, Volume+/-, EQ and so on. You can specify single track playback or make it sleep to turn off single repeat playback

status.

3.3.10. Feedback for Query Playback Status

At playing status	7E FF 06 42 00 00 01 xx xx EF	Playing
At pause status	7E FF 06 42 00 00 02 xx xx EF	Paused during playback
At stop status	7E FF 06 42 00 00 00 xx xx EF	Playback finished
At sleep status	7E FF 06 42 00 00 08 xx xx EF	No device online or sleeping

1).There is 4 status that can be queried as above. Users can send the query command to get to know the current status.

3.3.11. Commands for Stop

Stop playing advertisement	7E FF 06 15 00 00 00 FE E6 EF	Stop playing current ad and go back to play the music interrupted
Stop playback	7E FF 06 16 00 00 00 FE E5 EF	Stop software decoding

1).During playback of the module, there is two modes to stop. One is to stop playing the inter-cut advertisement, and go back and continue to play the music interrupted, and the other mode is to stop all playback(stop decoding).

2).For example, suppose the module is playing an inter-cut advertisement, and now if send a stop command 0x16, it will stop all playback tasks.

3.3.12. Specify Repeat(Loop) Playback of a Folder

Specify repeat playback of a folder	7E FF 06 17 00 00 02 FE E2 EF	Specify repeat playback of the folder "02"
	7E FF 06 17 00 00 01 FE E3 EF	Specify repeat playback of the folder "01"

1).The names of folders must be 01-99, and no more than 99.

2).After send the command, it will repeatedly play the tracks in the specific folder, and it will not stop until it received a command to stop.

3.3.13. Command for Random Playback

Random Playback	7E FF 06 18 00 00 02 FE E3 EF	Random playback of the whole storage device
-----------------	-------------------------------	---

1).This command is used to randomly play sound files in the storage device according to physical sequence and no matter if there is a folder or not in the device. The first sound file that is conducted to be played is the first one in the device.

3.3.14. Set Repeat(Loop) Playback of Current Track

Set repeat playback of current track	7E FF 06 19 00 00 00 FE E2 EF	Turn on single repeat playback
	7E FF 06 19 00 00 01 FE E1 EF	Turn off single repeat playback



- 1). During playback, send the turn-on command, and it will repeatedly play the current track. If the module is at Pause or Stop status, it will not respond to this command.
- 2). If you need to turn off repeat playback, just send the turn-off command.

3.3.15. Turn On and Turn OFF DAC

Set up DAC	7E FF 06 1A 00 00 00 FE E1 EF	Turn on DAC
	7E FF 06 1A 00 00 01 FE E0 EF	Turn off DAC(high resistance)

- 1). When the module is powered on, DAC is turned on by default. It is not turned off until it is set up by sending the command.

3.3.16. Query Total File Numbers of A Folder

Query total file numbers of a folder	7E FF 06 4E 00 00 01 FE AC EF	Query the total file numbers of the folder "01".
	7E FF 06 4E 00 00 0B FE A2 EF	Query the total file numbers of the folder "11".

- 1). The valid files that can be queried are MP3 format and WAV format only.

3.3.17. Query Total Folder Numbers of Current Storage Device

Query total folders	7E FF 06 4F 00 00 00 FE AC EF	Query the total folder numbers of current storage device
---------------------	-------------------------------	--

- 1). Users can query the total folder numbers of the current storage device through sending the command above. This just supports to query the folder numbers under the root directory of the device. Not possible to query the sub-folder numbers (Please don't build any sub-folders under a folder).

3.4. Examples of Sending Serial Commands

Commands Description	Serial Commands [with checksum]	Serial Commands [without checksum]	Notes
Play Next	7E FF 06 01 00 00 00 FE FA EF	7E FF 06 01 00 00 00 EF	
Play Previous	7E FF 06 02 00 00 00 FE F9 EF	7E FF 06 02 00 00 00 EF	
Specify playback of a track under the root directory	7E FF 06 03 00 00 01 FE F7 EF	7E FF 06 03 00 00 01 EF	Specify playback of the 1 st track
	7E FF 06 03 00 00 02 FE F6 EF	7E FF 06 03 00 00 02 EF	Specify playback of the 2 nd track
	7E FF 06 03 00 00 0A FE EE EF	7E FF 06 03 00 00 0A EF	Specify playback of the 10th track
Specify volume	7E FF 06 06 00 00 1E FE D7 EF	7E FF 06 06 00 00 1E EF	Specified volume is level 30
Specify EQ	7E FF 06 07 00 00 01 FE F3 EF	7E FF 06 07 00 00 01 EF	Specified EQ mode is POP
Specify single repeat playback	7E FF 06 08 00 00 01 FE F2 EF	7E FF 06 08 00 00 01 EF	Repeatedly play the 1 st track
	7E FF 06 08 00 00 02 FE F1 EF	7E FF 06 08 00 00 02 EF	Repeatedly play the 2nd track
	7E FF 06 08 00 00 0A FE E9 EF	7E FF 06 08 00 00 0A EF	Repeatedly play the 10th track



Specify playback device	7E FF 06 09 00 00 01 FE F1 EF	7E FF 06 09 00 00 01 EF	Specified device is USB flash disk
	7E FF 06 09 00 00 02 FE F0 EF	7E FF 06 09 00 00 02 EF	Specified device is micro SD
Enter into sleep mode	7E FF 06 0A 00 00 00 FE F1 EF	7E FF 06 0A 00 00 00 EF	
Reset	7E FF 06 0C 00 00 00 FE EF EF	7E FF 06 0C 00 00 00 EF	
Play	7E FF 06 0D 00 00 00 FE EE EF	7E FF 06 0D 00 00 00 EF	
Pause	7E FF 06 0E 00 00 00 FE ED EF	7E FF 06 0E 00 00 00 EF	
Specify playback of a folder	7E FF 06 0F 00 01 01 FE EA EF	7E FF 06 0F 00 01 01 EF	Specify track "001" in the folder "01"
	7E FF 06 0F 00 01 02 FE E9 EF	7E FF 06 0F 00 01 02 EF	Specify track "002" in the folder "01"
All repeat playback	7E FF 06 11 00 00 01 FE E9 EF	7E FF 06 11 00 00 01 EF	
Specify playback of folder named "MP3"	7E FF 06 12 00 00 01 FE E8 EF	7E FF 06 12 00 00 01 EF	Play track "0001" in the folder "MP3"
	7E FF 06 12 00 00 02 FE E7 EF	7E FF 06 12 00 00 02 EF	Play track "0002" in the folder "MP3"
	7E FF 06 12 00 00 FF FD EA EF	7E FF 06 12 00 00 FF EF	Play track "0255" in the folder "MP3"
	7E FF 06 12 00 07 CF FE 13 EF	7E FF 06 12 00 07 CF EF	Play track "1999" in the folder "MP3"
	7E FF 06 12 00 0B B8 FE 26 EF	7E FF 06 12 00 0B B8 EF	Play track "3000" in the folder "MP3"
Inter cut an advertisement	7E FF 06 13 00 00 01 FE E7 EF	7E FF 06 13 00 00 01 EF	Inter cut track "0001" in the folder "ADVERT"
	7E FF 06 13 00 00 02 FE E6 EF	7E FF 06 13 00 00 02 EF	Inter cut track "0002" in the folder "ADVERT"
	7E FF 06 13 00 00 FF FD E9 EF	7E FF 06 13 00 00 FF EF	Inter cut track "0255" in the folder "ADVERT"
Specify playback of a folder that supports 3000 tracks	7E FF 06 14 00 10 FF FD D8 EF	7E FF 06 14 00 10 FF EF	Play track "0255" in the specified folder "01"
	7E FF 06 14 00 17 CF FE 01 EF	7E FF 06 14 00 17 CF EF	Play track "1999" in the specified folder "01"
	7E FF 06 14 00 C0 01 FE 26 EF	7E FF 06 14 00 C0 01 EF	Play track "0001" in the specified folder "12"
	7E FF 06 14 00 C0 FF FD 28 EF	7E FF 06 14 00 C0 FF EF	Play track "0255" in the specified folder "12"
	7E FF 06 14 00 C7 CF FD 51 EF	7E FF 06 14 00 C7 CF EF	Play track "1999" in the specified folder "12"
Stop playing inter-cut ad	7E FF 06 15 00 00 00 FE E6 EF		Go back and continue to play the music interrupted
Stop playback	7E FF 06 16 00 00 00 FE E5 EF		Stop software decoding
Specify repeat playback of a folder	7E FF 06 17 00 00 02 FE E2 EF	7E FF 06 17 00 00 02 EF	Specify repeat playback of the folder "02"
	7E FF 06 17 00 00 01 FE E3 EF	7E FF 06 17 00 00 01 EF	Specify repeat playback of the folder "01"



Random playback	7E FF 06 18 00 00 00 FE E3 EF	7E FF 06 18 00 00 00 EF	
Single repeat playback	7E FF 06 19 00 00 00 FE E2 EF	7E FF 06 19 00 00 00 EF	Turn on single repeat playback
	7E FF 06 19 00 00 01 FE E1 EF	7E FF 06 19 00 00 01 EF	Turn off single repeat playback
DAC setup	7E FF 06 1A 00 00 00 FE E1 EF	7E FF 06 1A 00 00 00 EF	Turn on DAC
	7E FF 06 1A 00 00 01 FE E0 EF	7E FF 06 1A 00 00 01 EF	Turn off DAC
Query current status	7E FF 06 42 00 00 00 FE B9 EF	7E FF 06 42 00 00 00 EF	
Query current volume	7E FF 06 43 00 00 00 FE B8 EF	7E FF 06 43 00 00 00 EF	
Query current EQ	7E FF 06 44 00 00 00 FE B7 EF	7E FF 06 44 00 00 00 EF	
Query total file numbers of USB flash disk	7E FF 06 47 00 00 00 FE B4 EF	7E FF 06 47 00 00 00 EF	Total file numbers of current device
Query total file numbers of micro SD card	7E FF 06 48 00 00 00 FE B3 EF	7E FF 06 48 00 00 00 EF	Total file numbers of current device
Query current track of USB flash disk	7E FF 06 4B 00 00 00 FE B0 EF	7E FF 06 4B 00 00 00 EF	Query the track being played
Query current track of micro SD card	7E FF 06 4C 00 00 00 FE AF EF	7E FF 06 4C 00 00 00 EF	Query the track being played
Query total file numbers of a folder	7E FF 06 4E 00 00 01 FE AC EF	7E FF 06 4E 00 00 01 EF	Query the total file numbers of the folder "01".
	7E FF 06 4E 00 00 0B FE A2 EF	7E FF 06 4E 00 00 0B EF	Query the total file numbers of the folder "11".

3.5. Example of Serial Program

Code example: specify playback of a track

```

/*****
- 实现功能：实现芯片上电分别指定播放第一曲和第二曲，基本的程序供用户测试
- 运行环境：STC 晶振：11.0592M 波特率:9600
- 备注：在普中科技的 51 开发板上调试 OK --- STC89C516RD+
1、该测试程序必须是模块或者芯片方案中有设备在线，譬如 U 盘、TF 卡、FLASH
*****/

#include "REG52.h"

#define COMM_BAUD_RATE 9600 //串口波特率
#define OSC_FREQ 11059200 //运行晶振：11.05926MHZ
static INT8U Send_buf[10] = {0};

void Delay_Ms(INT32U z)
{
    INT32U x=0, y=0;
    for(x=110; x>0; x--)
        for(y=z; y>0; y--);
}

/*****
- 功能描述：串口 1 初始化
- 注：设置为 9600 波特率
*****/
void Serial_init(void)

```

```
{
    TMOD = 0x20;           // 设置 T1 为波特率发生器
    SCON = 0x50;           // 0101,0000 8 位数据位, 无奇偶校验
    PCON = 0x00;           //PCON=0;
    TH1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12); //设置为 9600 波特率
    TL1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);
    TR1    = 1;            //定时器 1 打开
    REN    = 1;            //串口 1 接收使能
    ES     = 1;            //串口 1 中断使能
}
```

```
void Uart_PutByte(INT8U ch)
```

```
{
    SBUF = ch;
    while(!TI){;}
    TI = 0;
}
```

```

/*****
- 功能描述: 串口向外发送命令[包括控制和查询]
- 参数说明: CMD:表示控制指令, 请查阅指令表, 还包括查询的相关指令
            feedback:是否需要应答[0:不需要应答, 1:需要应答]
            data:传送的参数
*****/

```

```
void SendCmd(INT8U len)
```

```
{
    INT8U i = 0 ;
    Uart_PutByte(0x7E); //起始
    for(i=0; i<len; i++)//数据
    {
        Uart_PutByte(Send_buf[i]) ;
    }
    Uart_PutByte(0xEF) ;//结束
}
```

```

/*****
- 功能描述: 求和校验
- 和校验的思路如下:

```

发送的指令, 去掉起始和结束。将中间的 6 个字节进行累加, 最后取反码。接收端就将接收到的一帧数据, 去掉起始和结束。将中间的数据累加, 再加上接收到的校验字节。刚好为 0.这样就代表接收到的数据完全正确。

```
void DoSum( INT8U *Str, INT8U len)
```

```
{
    INT16U xorsum = 0;
    INT8U i;
    for(i=0; i<len; i++)
    {
        xorsum = xorsum + Str[i];
    }
    xorsum = 0 -xorsum;
    *(Str+i) = (INT8U)(xorsum >>8);
    *(Str+i+1) = (INT8U)(xorsum & 0x00ff);
}
```

```
void Uart_SendCMD(INT8U CMD ,INT8U feedback , INT16U dat)
```

```
{
    Send_buf[0] = 0xff;    //保留字节

```

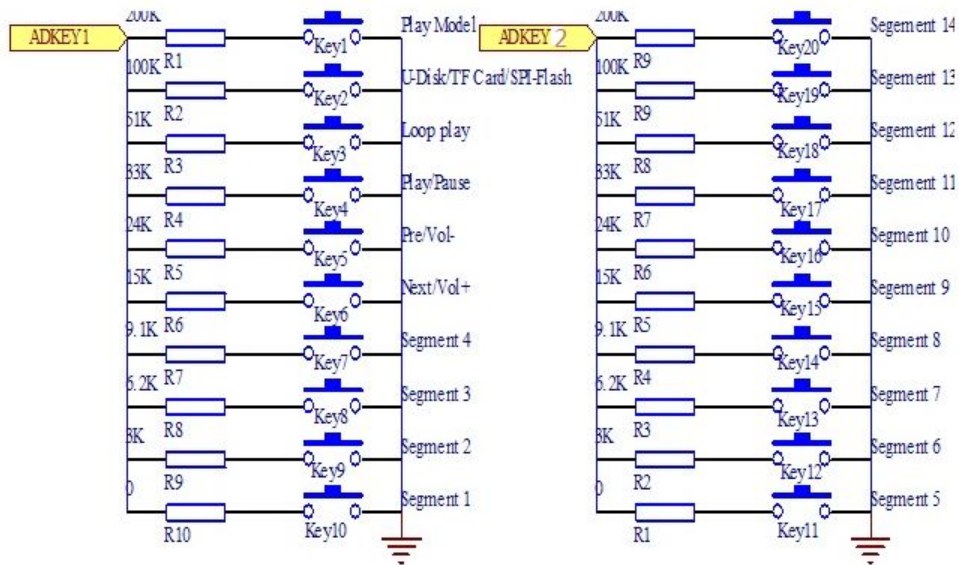
```

Send_buf[1] = 0x06;    //长度
Send_buf[2] = CMD;     //控制指令
Send_buf[3] = feedback; //是否需要反馈
Send_buf[4] = (INT8U)(dat >> 8); //datah
Send_buf[5] = (INT8U)(dat);    //datal
DoSum(&Send_buf[0],6);        //校验
SendCmd(8);                    //发送此帧数据
}

void main()
{
    Serial_init(); //串口寄存器的初始化设置
    Uart_SendCMD(0x03, 0, 0x01); //播放第一首
    Delay_Ms(1000); //延时大概 6S
    Uart_SendCMD(0x03, 0, 0x02); //播放第二首
    Delay_Ms(1000); //延时大概 6S
    Uart_SendCMD(0x03, 0, 0x04); //播放第四首
    while(1);
}

```

4. ADkey Control



Key	Short Push	Long Push	Description
K1	Play Mode		Switch to interrupted / non-interrupted
K2	Switching of playback devices		USB/micro SD
K3	Operating Mode		All repeat playback
K4	Play/Pause		
K5	Previous	Vol+	
K6	Next	Vol-	
K7	4	Repeat playback track 4	Long push always to repeat playback
K8	3	Repeat playback track 3	Long push always to repeat playback

K9	2	Repeat playback track 2	Long push always to repeat playback
K10	1	Repeat playback track 1	Long push always to repeat playback
K11	5	Repeat playback track 5	Long push always to repeat playback
K12	6	Repeat playback track 6	Long push always to repeat playback
K13	7	Repeat playback track 7	Long push always to repeat playback
K14	8	Repeat playback track 8	Long push always to repeat playback
K15	9	Repeat playback track 9	Long push always to repeat playback
K16	10	Repeat playback track 10	Long push always to repeat playback
K17	11	Repeat playback track 11	Long push always to repeat playback
K18	12	Repeat playback track 12	Long push always to repeat playback
K19	13	Repeat playback track 13	Long push always to repeat playback
K20	14	Repeat playback track 14	Long push always to repeat playback

5. Application Circuits

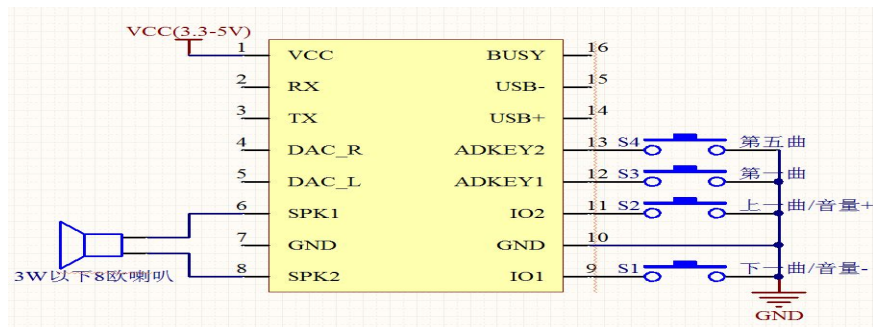


Figure 5-1: Connect speaker and key control

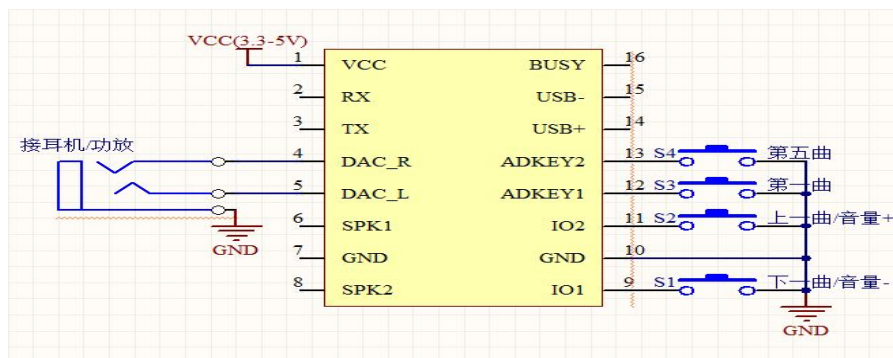


Figure 5-2: Connect earphone/amplifier and key control

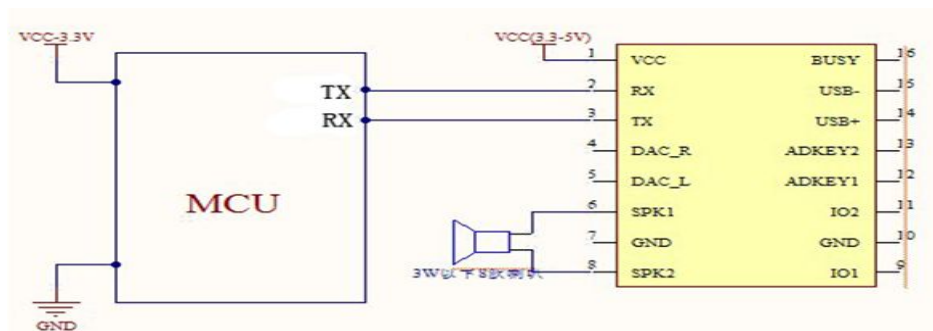


Figure 5-3: Connect 3.3V MCU and speaker

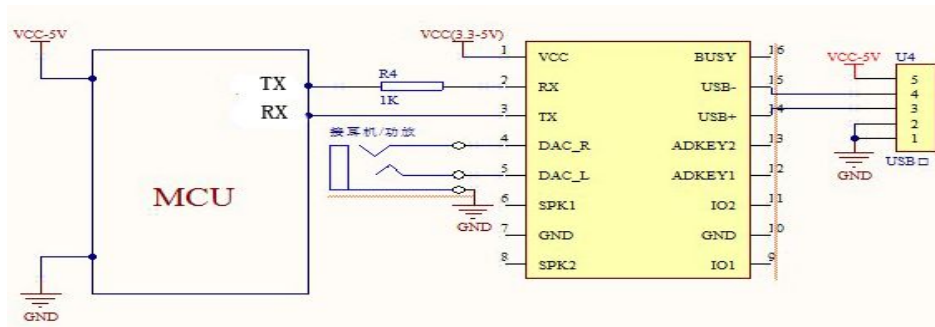


Figure 5-4: Connect 5V MCU and earphone/amplifier

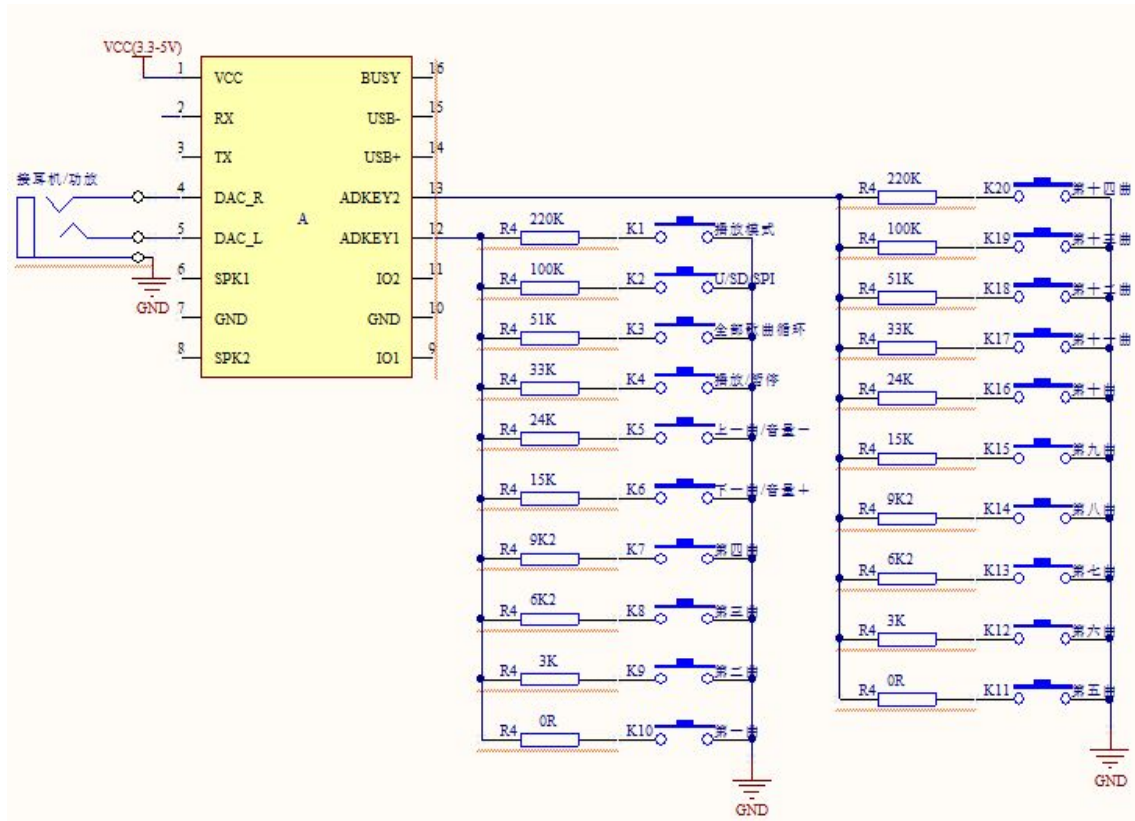


Figure 5-5: Connect AD key control, earphone/amplifier and micro SD card

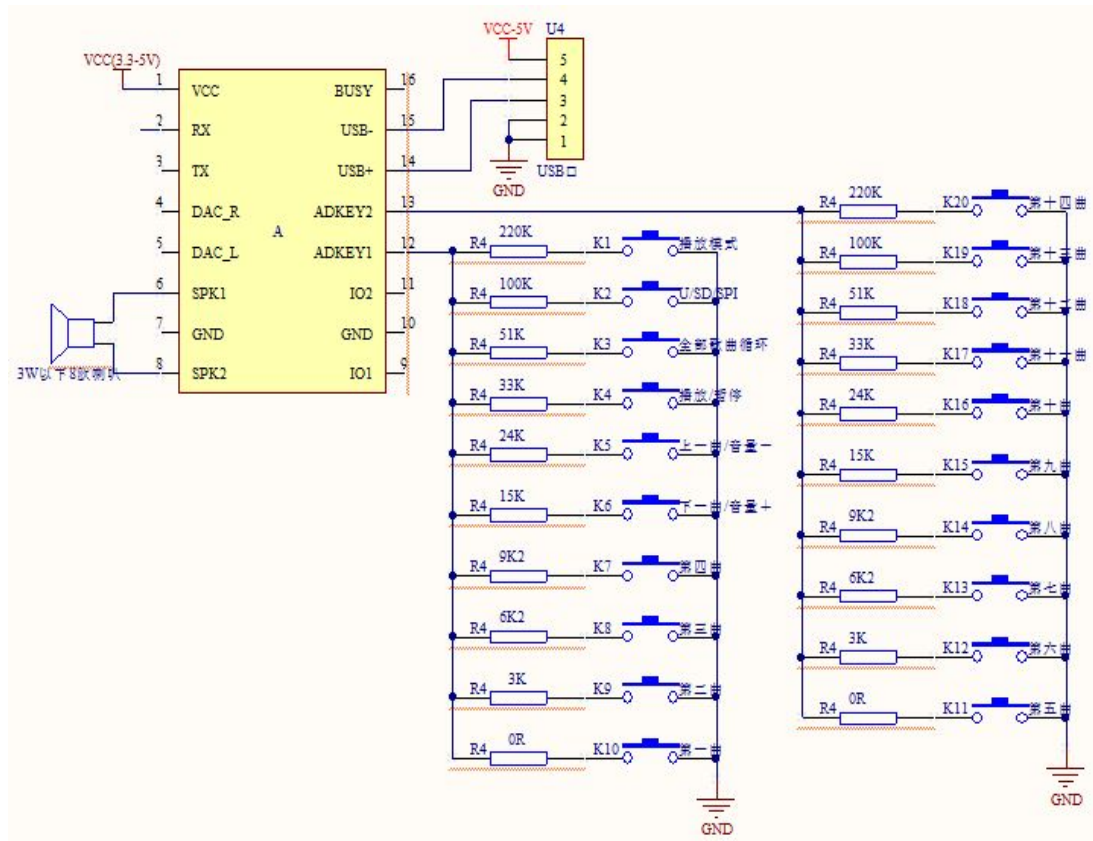


Figure 5-6: Connect AD key control, speaker and USB port

6. Features of GPIO

I/O Input Features						
Item	Description	Min	Typical	Max	Unit	Test Condition
VIL	Low-Level Input Voltage	-0.3	-	0.3*VDD	V	VDD=3.3V
VIH	High-Level Input Voltage	0.7VDD	-	VDD+0.3	V	VDD=3.3V
I/O Output Features						
Item	Description	Min	Typical	Max	Unit	Test Condition
VOL	Low-Level Output Voltage	-	-	0.33	V	VDD=3.3V
VOH	High-Level Output Voltage	2.7	-	-	V	VDD=3.3V

7. PCB Size (unit: mm)

