FN-MK30 Serial MP3 Player Module

Datasheet

V1.0

Contents

1. Overviews

1.1. Brief Introduction

FN-MK30 is a newly designed serial MP3 player module for robot design and industrial automation applications. This module has an onboard compact 2W 8ohm speaker. It is able to play back specified audio files and realize other functions through simple UART serial commands provided. It uses a SPI flash memory as the storage unit, and users can update audio files easily through connecting it to computer as simple as using a USB flash drive.

1.2. Features

- ✧ Uses standard UART serial port.
- ✧ Equipped with PH2.0 female connector.
- ✧ Built-in 4Mbytes (32Mbits) SPI flash memory.
- ✧ Update audio files via USB connection to computer.

  -The internal memory will be detected as a USB flash drive on computer.
- ✧ Equipped with a 2W 8ohm speaker.
- ✧ Able to play back specified audio files in the root directory or folders.
- ✧ Supports combination playback (group playback).
- ✧ Supports random playback and repeat/loop playback.
- ✧ 30-level adjustable volume.

1.3. Size

1.4. 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
USB Port	Standard USB 2.0
UART Port	Standard serial port and 3.3V TTL level
Working Voltage	DC 3.3~5.0V
Standby Current	<10mA
Operating Temperature	-40~+80℃
Humidity	5%~95%

1.5. Naming Rules of Audio Files(Tracks) and Folders

1). Audio files directly stored in the root directory of the SPI flash memory need to be renamed as 0001, 0002, 0003..... and so on.

Here it works according to physical index order when you copy the files from computer to the internal memory. For example, when the module receives a command to play the track 0001.mp3, it will play the 1st track you copied from computer, probably 0001.mp3 or not (maybe it would play 0007.mp3 if it was the first one you copied from computer). In order to avoid this problem, when you make the copy, rename the audio files firstly on computer and put all the renamed files in a folder, then press “Ctrl+A” on the key board to select all, and press “Ctrl+C” to copy, and go back to the memory, and press “Ctrl+V” to past all of the files into the memory. Or users just directly give up this way and just move the audio files to folders and choose to control and play them in a folder as below.

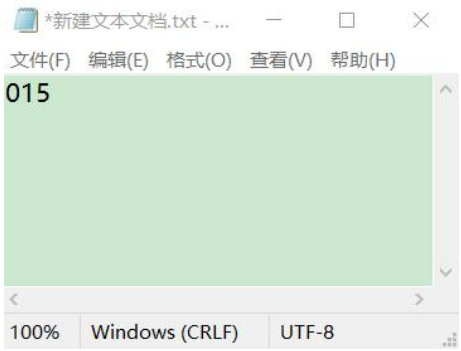
2). Folders in the root directory must be renamed as 01, 02, 03.....99, and the audio files in a folder must be renamed as 001, 002, 003.....and so on. It is also possible to keep the original name when you rename a file. For example, the original name is “Yesterday Once More.mp3”, then you can rename it as “001Yesterday Once More.mp3”.

1.6. About Volume Setting

This module has 31 volume levels from “00” to “30”. “00” represents mute while “30” represents the maximum volume level. By default, the module works with the maximum volume level. Usually users can send the relevant serial command to adjust the volume (convert the volume level number into hex first). Besides, you can create a config file in the root directory of the memory to set the volume in advance. Please refer to the steps as below.

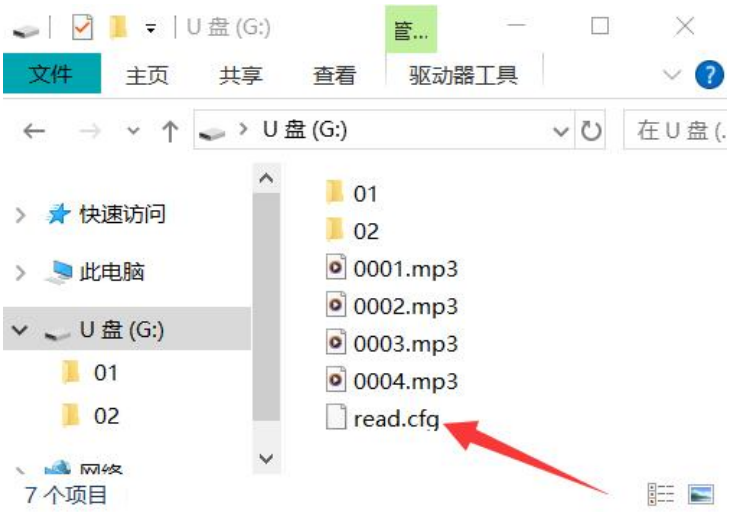
1). Create a text file (.txt) in the root directory of the memory.

2). Open the file and enter three digits like 015, which is a moderate volume level. See as below.



3). Save the file and change the file name to “read”, and change the extension name “.txt” to “.cfg”. See as below.

4). If you need to change the volume level, you still can open the file and edit it with the program of text file.



location of the config file in the memory

Notes:

1). There must be an extra “0” ahead of the actual volume level number, so it must be three digits in the config file.

2). Only numbers “000” to “030” are valid in the config file. “000” represents mute while “030” represents the maximum volume level.

3). When changing the file name, please ensure your computer has displayed the extension names of the files.

4). The volume setting in the config file is always in priority.

2. 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.

2.1. Serial Commands Format

Baud rate: 9600 bps

Data bits: 8

Stop bits: 1

Checkout: none

Flow Control: none

Format: $S Ver. Length CMD Feedback Para_MSB Param_LSB Check_MSB Check_LSB $O	
$S	Start byte 0x7E
Ver.	Version 0xFF by default
Length	Number of byte from version info to Check_LSB, typically 0x06 (checksum not counted)
CMD	Command Code
Feedback	0x01: Need feedback--send confirmation back to MCU; 0x00: No need feedback
Para_MSB	Most significant byte of parameter
Para_LSB	Least significant byte of parameter
Check_MSB	Most significant byte of checksum
Check_LSB	Least significant byte of checksum
$O	End byte 0xEF

2.2. About Checksum

Regarding to calculating checksum, you can use the following formula to count.

Checksum (2 bytes) = 0xFFFF–(CMD + Feedback + Para_MSB + Para_LSB) + 1

Normally it’s okay whether users choose to use checksum or not, our module can receive a frame of serial data with or without checksum, but some of users use a MCU without crystal oscillator, so if in that case we strongly suggest users to add checksum to make sure a stable communication.

2.3. Serial Commands

2.3.1. Control Commands

Command	Function Description	Note
0x01	Play Next	

0x02	Play Previous	
0x03	Specify playback of a track	See 2.6.1 for details
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	See 2.6.2 for details
0x07	N/A(Reserved)	
0x08	Specify single repeat playback in a folder	See 2.6.3 for details
0x09	N/A(Reserved)	
0x0A	Set Sleep	See 2.6.4 for details
0x0B	Awake from sleep	
0x0C	Reset	
0x0D	Play	
0x0E	Pause	
0x0F	Specify playback a track in a folder	See 2.6.5 for details
0x14	N/A(Reserved)	
0x15	Stop playing inserted advertisement and go back to play the music interrupted	See 2.6.6 for details
0x16	Stop all playback activities	
0x17	Specify repeat playback of a folder	See 2.6.7 for details
0x18	Set random playback	See 2.6.8 for details
0x19	Set repeat playback of current track	See 2.6.9 for details
0x1A	Set DAC	See 2.6.10 for details
0x21	Set combination playback(playback of a group)	See 2.6.11 for details
0x25	Insert an advertisement	See 2.6.12 for details

2.3.2. Query Commands

Command	Function Description	Note
0x40	Module returns an error data with this command	
0x41	Module reports a feedback with this command	
0x42	Query current status	See 2.7.1 for details
0x43	Query current volume	
0x49	Query number of tracks	Total number of audio files
0x4D	Query current track	Based on physical order
0x4E	Query number of tracks in a folder	See 2.7.2 for details
0x4F	Query number of folders	See 2.7.3 for details

2.4. Examples of Sending Serial Commands

Command Description	Serial Command [with checksum]	Serial Command [without checksum]	Note
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 in 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 1st track
	7E FF 06 03 00 00 02 FE F6 EF	7E FF 06 03 00 00 02 EF	Specify playback of the 2nd 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
Volume Up	7E FF 06 04 00 00 00 FE F7 EF	7E FF 06 04 00 00 00 EF	
Volume Down	7E FF 06 05 00 00 00 FE F6 EF	7E FF 06 05 00 00 00 EF	
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 single repeat playback in a folder	7E FF 06 08 00 01 01 FE F1 EF	7E FF 06 08 00 01 01 EF	Loop playback of track 001 in folder 01
	7E FF 06 08 00 02 01 FE F1 EF	7E FF 06 08 00 02 01 EF	Loop playback of track 001 in folder 02
Set sleep mode	7E FF 06 0A 00 00 00 FE F1 EF	7E FF 06 0A 00 00 00 EF	
Awake from sleep	7E FF 06 0B 00 00 00 FE F0 EF	7E FF 06 0B 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 track in 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”
Stop	7E FF 06 15 00 00 00 FE E6 EF	7E FF 06 15 00 00 00 EF	Stop playback of the inserted ad
	7E FF 06 16 00 00 00 FE E5 EF	7E FF 06 16 00 00 00 EF	Stop all playback tasks
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”
Set random playback	7E FF 06 18 00 00 00 FE E3 EF	7E FF 06 18 00 00 00 EF	
Set 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
Set DAC	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
Insert an advertisement	7E FF 06 25 00 01 01 FE D4 EF	7E FF 06 25 00 01 01 EF	insert track "001" in the folder “ ADVERT1”
	7E FF 06 25 00 01 02 FE D3 EF	7E FF 06 25 00 01 02 EF	insert track "002" in the folder “ADVERT1”
	7E FF 06 25 00 02 01 FE D3 EF	7E FF 06 25 00 02 01 EF	insert track "001" in the folder “ADVERT2”
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 number of tracks	7E FF 06 49 00 00 00 FE B2 EF	7E FF 06 49 00 00 00 EF	

Query number of tracks in a folder	7E FF 06 4E 00 00 01 FE AC EF	7E FF 06 4E 00 00 01 EF	Query number of tracks in the folder “01”.
	7E FF 06 4E 00 00 0B FE A2 EF	7E FF 06 4E 00 00 0B EF	Query number of tracks in the folder “11”.
Query number of folders	7E FF 06 4F 00 00 00 FE AC EF	7E FF 06 4F 00 00 00 EF	

2.5. Returned Data from Module

2.5.1 Returned data after the module is powered on

1). After the module is powered on, normally it needs about no more than 500ms to 1500ms(depending on the actual track quantities in the storage device) initialization time. Once the initialization is done, the module returns a data to MCU. If it does not return a data after more than the initialization time, it means there is an error for initialization, and please check the hardware connections.

2). The returned data from module after initialization means the current effective storage device/online equipment. For example, the module returns 7E FF 06 3F 00 00 08 xx xx EF. 0x3F is the returned command by module, and 0x08 represents the SPI flash is effective/online.

3). MCU can not send commands to control the module until the initialization of the module is done and a data is returned, otherwise the commands sent by MCU will be ignored and also this will effect initializing of the module.

2.5.2. Returned data after a track is finished playing

Track Played	Returned Data
1st track in folder 01 is finished playing	7E FF 06 3E 00 01 01 xx xx EF
2nd track on folder 02 is finished playing	7E FF 06 3E 00 02 02 xx xx EF

1). There is a returned data after a track is finished playing. For example, the returned data is 7E FF 06 3E 00 01 01 xx xx EF. 0x3E represents SPI flash memory. 0x01 and 0x01 represents the 1st track in folder 01.

2). Because all of the files(tracks) in the root of the flash memory are read in physical sequence, the returned data still follow the physical sequence, which needs to be noted.

2.5.3 Returned data of feedback from module

Module returns ACK	7E FF 06 41 00 00 00 xx xx EF

1). In order to enhance stability between data communication, the function of a feedback from module is added. Once there is a feedback to MCU from the module, it means the module has successfully received the command that MCU sent out. 0x41 is the returned command by module.

2). Users are free to choose this feedback or not. It’s also fine not to choose this function.

2.5.4 Returned data of errors

Returned Data of Errors	Meaning Description
7E FF 06 40 00 00 01 xx xx EF	Module busy(this info is returned when the initialization is not done)

7E FF 06 40 00 00 02 xx xx EF	Currently sleep mode(supports only specified device in sleep mode)
7E FF 06 40 00 00 03 xx xx EF	Serial receiving error(a frame has not been received completely yet)
7E FF 06 40 00 00 04 xx xx EF	Checksum incorrect
7E FF 06 40 00 00 05 xx xx EF	Specified track is out of current track scope
7E FF 06 40 00 00 06 xx xx EF	Specified track is not found
7E FF 06 40 00 00 07 xx xx EF	Inter-cut error(an inter-cut operation only can be done when a track is being played)
7E FF 06 40 00 00 09 xx xx EF	Initialization error SPI flash memory
7E FF 06 40 00 00 0A xx xx EF	Entered into sleep mode

2.6. Detailed Annotation of Control Commands

2.6.1.Specify playback of a track in the root of SPI flash memory

The available selective tracks is from 0001.mp3 to 3000.mp3 in the root of SPI flash memory. 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.

1). For example, select the first song played, and send the command 7E FF 06 03 00 00 01 FF E7 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(MSB of Parameter)

01 --- Least significant byte of the track(LSB of Parameter)

FF --- Most significant byte of checksum(MSB of checksum)

E7 --- Least significant byte of checksum(LSB of checksum)

EF --- End byte

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

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

2.6.2. 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). MSB=0x00; LSB=0x0F, 15 is converted to hexadecimal 0x000F.

2.6.3. Specify single repeat playback in a folder

Start to repeatedly play the track 001 in folder 01	7E FF 06 08 00 01 01 xx xx EF
Start to repeatedly play the track 002 in folder 01	7E FF 06 08 00 01 02 xx xx EF

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+/-, and so on. You can specify single track playback or make it sleep to turn off single repeat playback status.

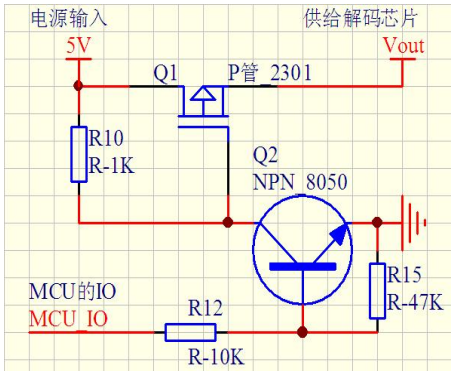
2.6.4. Set sleep mode, awake from sleep and reset

Set sleep mode	7E FF 06 0A 00 00 00 FE F1 EF
Awake from sleep	7E FF 06 0B 00 00 00 FE F0 EF
Reset	7E FF 06 0C 00 00 00 FE EF EF

1). After set the module enter into sleep mode, there is also another way other than sending the command to awake the module that re-power up the module.

2). Regarding the reset, it's a soft reset, and the reset time is 5-8 seconds. It is allowed to send the reset command under any status.

Note: When the module enters into the sleep mode, the standby power consumption is about 10mA. If users are very strict to the power consumption, you can use a MOS and a transistor to control power supply of the module. It is possible to cut off the power supply completely when standby is not necessary. Please refer to the schematic as below.



2.6.5. Specify playback of a track in a folder

Specify playback of track 001 in folder 01	7E FF 06 0F 00 01 01 xx xx EF
Specify playback of track 100 in folder 11	7E FF 06 0F 00 0B 64 xx xx EF
Specify playback of track 255 in folder 99	7E FF 06 0F 00 63 FF xx xx EF

1). The default folders are named as "01", "11", "99" in this way. In order to be with a better system stability, it is

made to support maximum 99 folders and maximum 255 tracks in each folder..

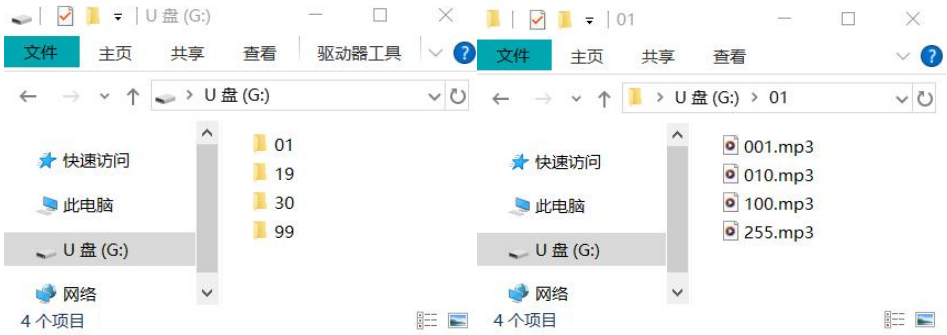
2). For example, if specify to play “100.mp3” in the folder "01", send the command 7E FF 06 0F 00 01 64 xx xx EF

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

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

3). You must specify both the folder and the file name to target a track.

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



2.6.6. Stop

Stop playback of the inter-cut advertisement	7E FF 06 15 00 00 00 FE E6 EF
Stop all playback tasks	7E FF 06 16 00 00 00 FE E5 EF

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 track 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.

2.6.7. Specify repeat playback of a folder

Specify repeat playback of folder “02”	7E FF 06 17 00 00 02 FE E2 EF
Specify repeat playback of folder “01”	7E FF 06 17 00 00 01 FE E3 EF

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

2). After sending the command, it’ll repeatedly play the tracks in the specific folder, and it will not stop until it receives a command to stop.

2.6.8. Set random playback

Random playback of tracks in the whole memory	7E FF 06 18 00 00 00 FE E3 EF

This command is used to randomly play audio files in the SPI flash according to physical sequence and no matter if there is a folder or not. The first audio file that is conducted to be played is the first one copied to the flash memory.

2.6.9. Set repeat playback of current track

Turn on single repeat playback	7E FF 06 19 00 00 00 FE E2 EF

Turn off single repeat playback	7E FF 06 19 00 00 01 FE E1 EF

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.

2.6.10. Set DAC

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

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.

2.6.11. Set combination playback(playback of a group)

1). We added this function to meet some users' special need that when users need to send only one frame data to play multiple tracks one by one without pause. It supports maximum 15 tracks together for combination playback. All of the sound files used for combination playback need to be put in folders(folder 01-folder 99).

2). If MCU sends a frame data as **7E FF 15 21 01 02 01 03 01 04 01 05 01 06 02 01 03 05 04 07 05 09 EF**, see the analysis as below.

Command: 0x21

Number of bytes: 0x15=21 bytes --- **FF 15 21 01 02 01 03 01 04 01 05 01 06 02 01 03 05 04 07 05 09**(two parameters for one track, i.e. the folder number and the track number)

The module will play track 002 in folder 01, track 003 in folder 01, track 004 in folder 01, track 005 in folder 01, track 006 in folder 01, track 001 in folder 02, track 005 in folder 03, track 007 in folder 04, and track 009 in folder 05.

3). During combination playback, it is allowed to Play/Pause and set volume, but not allowed to set Previous and Next. If need to stop, just direct send the stop command. And it is not allowed to play another group of combination during it is working. Users need to send the stop command to stop the current combination playback before start another group of combination playback.

4). If a track specified to be played in combination is not in the folder, it will stop playing at this track position, so please make sure the track specified to play must be available in the folder.

5). If users are very strict to the combination playback, please edit the sound sources with some audio edit software like Adobe Audition or GoldWave to cut off the silence at the beginning and the end of the sound.

6). Due to this frame command data is long, we cut off the byte "Feedback" compared with other commands. Please be noted.

2.6.12. Insert an advertisement

Insert track "001" in the folder “ADVERT1”	7E FF 06 25 00 01 01 FE D4 EF
Insert track "002" in the folder “ADVERT1”	7E FF 06 25 00 01 02 FE D3 EF
Insert track "001" in the folder “ADVERT2”	7E FF 06 25 00 02 01 FE D3 EF

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

2). After sending the command 0x25, the system will save the ID V3 information of the track being played and pause, and then it will play the specified insert track (advertisement). When the insert 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 to build a folder named “ADVERT1” in the storage device and put the tracks (ads) you need in the folder and name the files as “001.mp3”, 002.mp3. It supports maximum 9 folders from “ADVERT1 to ADVERT9”, and each folder can have maximum 255 tracks.

4). If you send an insert command when the module is at Pause status or Stop status, it will not work and there will be returned error information. In the course of an inter-cut, you can continue to insert the other tracks (ads). When the last inserted track goes to the end, the systems till goes back to the ID V3 position saved at the first time.

2.7. Detailed Annotation of Main Query Commands

2.7.1 Query current status

Query current status	7E FF 06 42 00 00 00 FE B9 EF

1). There are 4 status(playing, paused playing, stopped playing, and in sleep) that can be queried during the module is decoding. Users can query the current status via sending the command as above(0x42).

2). Interpretation of returned data

Returned Data	Status
7E FF 06 42 00 08 01 xx xx EF	A track in the SPI flash is being played
7E FF 06 42 00 08 02 xx xx EF	A track in the SPI flash is paused playing
7E FF 06 42 00 08 00 xx xx EF	A track in the SPI flash is stopped playing
7E FF 06 42 00 10 00 xx xx EF	Module in sleep

3). MSB and LSB Representations

MSB Representation		LSB Representation	
0x08	SPI flash	0x00	Stopped
0x10	Module in sleep mode	0x01	Playing
		0x02	Paused

2.7.2 Query number of tracks in a folder

Query number of tracks in folder 01	7E FF 06 4E 00 00 01 FE AC EF
Query number of tracks in folder 11	7E FF 06 4E 00 00 0B FE A2 EF

If the folder queried is empty without any files, the module will report an error, and the data 7E FF 06 40 00 00 06 FE B5 EF will be returned.

2.7.3 Query number of folders

Query number of folders	7E FF 06 4F 00 00 00 FE AC EF

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

3. Connection Examples

3.1. Connection Example of UART Serial Port

Connection to a 5V MCU **Connection to a 3.3V MCU**

The module uses 3.3V TTL level, so if you use a 5V MCU we suggest you attach a 1K resistor. Please refer to the upper-left diagram.

3.2. Connection Example of A Key or Button

When the key I/O port (P1) gets short connected to GND, the module will play back the first audio file stored in the internal memory. So it's possible to add a button between the key I/O port (P1) and GND. When you have got this module on hand, you can use a pair of tweezers to short-circuit these two contacts to test if the module can work well.