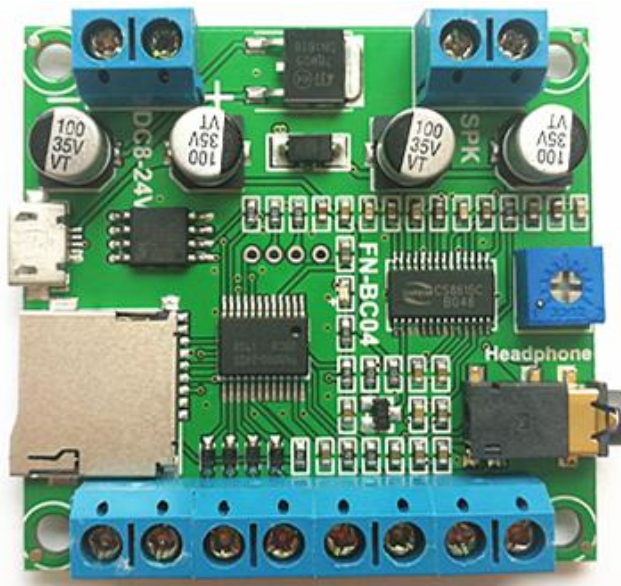




# **FN-BC04 MP3 Sound Module with 10W Amplifier**

## **User's Manual**

**V2.0**



## **Contents**

- 1. Overviews .....2**
  - 1.1. Brief introduction ..... 2
  - 1.2. Features ..... 2
  - 1.3. Technical parameters ..... 2
- 2. Connections .....2**
- 3. Button Control Mode .....3**
  - 3.1. Trigger Mode Selection.....3
  - 3.2. Audio Files Loading.....5
- 4. Serial Control Mode ..... 6**
  - 4.1. Naming Rules o Audio Files and Folders ..... 6
  - 4.2. Command Format.....6
  - 4.3. About Checksum ..... 7
  - 4.4. Serial Communication Commands .....7
  - 4.5. Returned Data from Module .....10
  - 4.6. Detailed Annotation of Control Commands .....12
  - 4.7. Detailed Annotation of Query Commands ..... 16
  - 4.8. About Delay of Serial Programming ..... 17



## 1. Overviews

### 1.1. Brief Introduction

FN-BC04 is a high quality MP3 sound module developed by Flyron Technology Co., Ltd. Equipped with an on-board 10W amplifier, the sound module can be controlled by 4 separate buttons hooked up to the 'one-on-one' inputs terminals and by UART R232 serial port working with a MCU. Great audio output, industry-grade design and strong anti-jamming capability make it possible to be used for many different applications.

### 1.2. Features

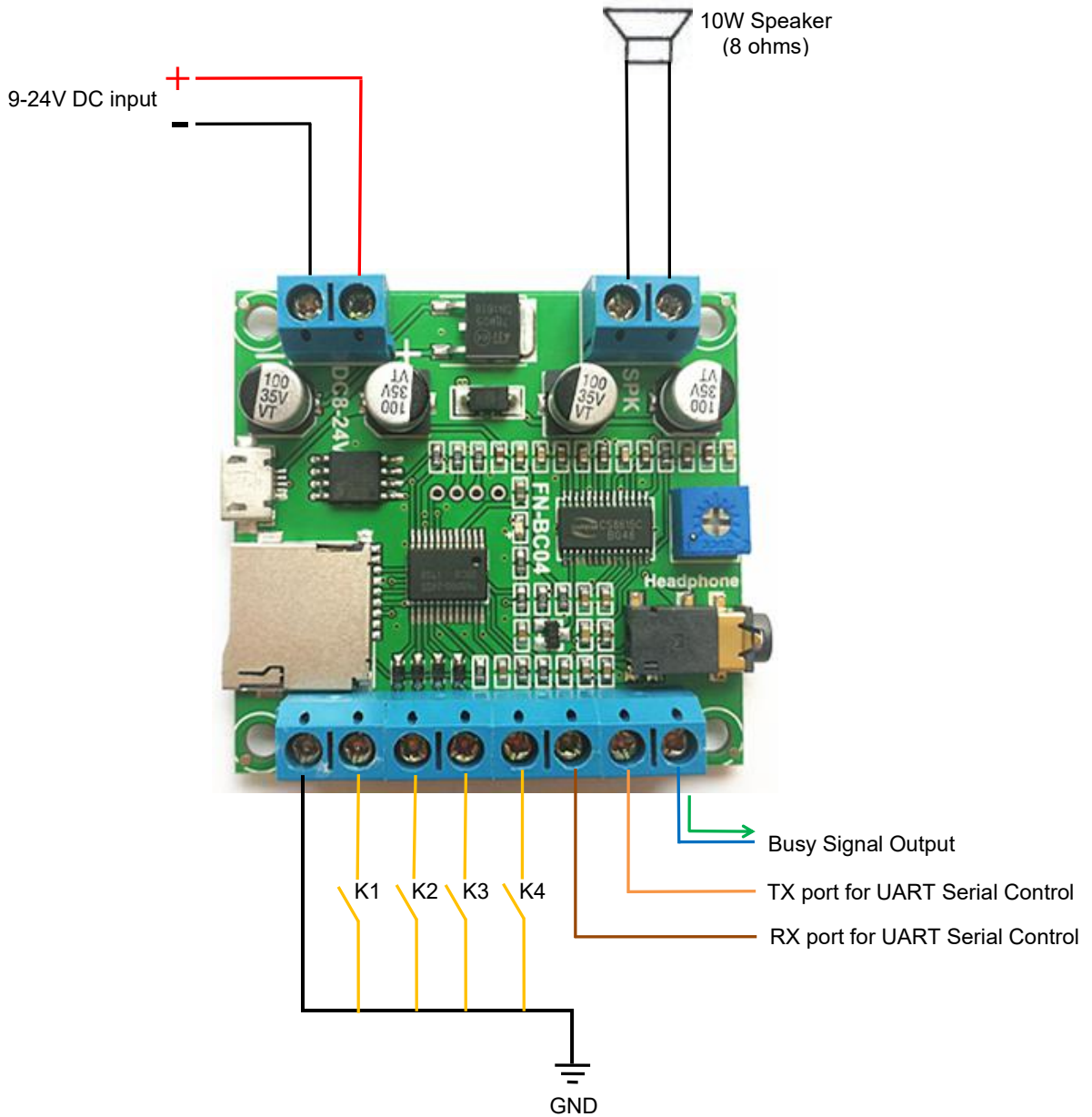
- 1). Equipped with a high quality audio decoder, which supports MP3 and WAV audio formats.
- 2). Sampling rates supported: 8/11.025/12/16/22.05/24/32/44.1/48(KHz).
- 3). 24 bit DAC output and supports dynamic range 90dB and SNR 85dB.
- 4). Supports one-on-one 4 button trigger control mode and RS232 serial port control mode.
- 5). In button control mode, it plays back 4 sounds one-on-one by negative triggering or low pulse signal.
- 6). Uses micro SD card and the built-in SPI flash memory for storage devices.
- 7). Built-in 4MB SPI flash memory and supports max. 32GB micro SD card.
- 8). Possible to load audio files to the flash memory directly via the micro USB connecting with computer like using a USB flash drive.
- 9). Equipped with a class D mono 10 watts amplifier that can drive a 10 watts speaker directly.
- 10). Equipped with a 3.5mm audio jack for stereo output that can drive an earphone directly or connect with an external amplifier.
- 11). Adjustable sound volume through the blue potentiometer.
- 12). A busy signal output available. It outputs a signal in low level when playing and a signal in high level when standby.
- 13). Wide range for power input (DC 9-24V).
- 14). PCB size: 52mmx49mm

### 1.3. Technical Parameters

- 1). Working voltage: DC 9V-24V
- 2). Working current:  $\leq 1000\text{mA}$
- 3). Power Consumption:  $\leq 10\text{W}$
- 4). On-board flash memory size: 4MB
- 5). Audio format: MP3 and WAV

## 2. Connections

K1, K2, K3 and K4 are representing Normally Open (N.O.) manual buttons. TX port and RX port are used to connect with a MCU, through which sending serial commands to control the module. Please refer to the connection example below.



- Notes:**
- 1). The Busy port outputs a signal in low level when the module is playing a sound, and it output a signal in high level when the module is standby.
  - 2). When you use buttons to control the module, please skip the connection for UART serial control, and vice versa.

### 3. Button Control Mode

#### 3.1. Trigger Mode Selection

In button control mode, there are 4 trigger modes available for users to choose according to the actual needs. Any of these 4 trigger modes can be set/acquired through a configuration file named “read.cfg”, which comes from a text file(.txt) originally. Users just need to fill in a digit/parameter that is corresponding to a trigger mode in a new built text file. Save it and rename it “read.cfg”, then put it in the root directory of the micro SD card or the flash memory together with the 4 audio files. Please refer to the sheet below about the digits and the associated trigger modes.

Digit in file "read.cfg"	Corresponding Trigger Mode
0	Pulse interruptible one-on-one playback
1	Level hold loop playback
2	Pulse non-interruptible one-on-one playback
3	Standard MP3 key mode playback

- Pulse interruptible one-on-one playback: In this mode, a single negative pulse will start playback. It is possible to interrupt the playback by pressing the same button used to activate. Once playback is interrupted, it will automatically restart the audio file immediately. It's also possible to interrupt the playback by pressing any of the other 3 buttons. Once playback is interrupted, it will automatically start the sound that is associated with the button pressed.
- Level hold loop playback: In this mode, the negative pulse must be held/maintained to the sound module trigger for audio file to complete. The audio file will only playback while button, or negative pulse, is held/maintained during playback. Once the button being held, or negative pulse, is removed, the playback will be stopped/cancelled. Once the button is kept holding, when the playback of the audio file is finished, it will start to play it repeatedly(loop playback).
- Pulse non-interruptible one-on-one playback: In this mode, a single negative pulse will start playback. It's not possible to interrupt the playback by pressing the same button or the other buttons. Once an audio file is triggered, the audio file will not be able to be interrupted/cancelled during playback. The playback will only end when the audio file has played its entirety.
- Standard MP3 key mode playback: In this mode, the buttons between K1 and K4 will be functioned as Previous, Next, Play/pause, and Stop respectively. In this case not like the other 3 trigger modes above, more than 4 audio files can be placed.

For example, if the trigger mode of level hold loop playback is needed, firstly build a new TXT file on the computer, and simply enter the digit "1" as below, and save the file.



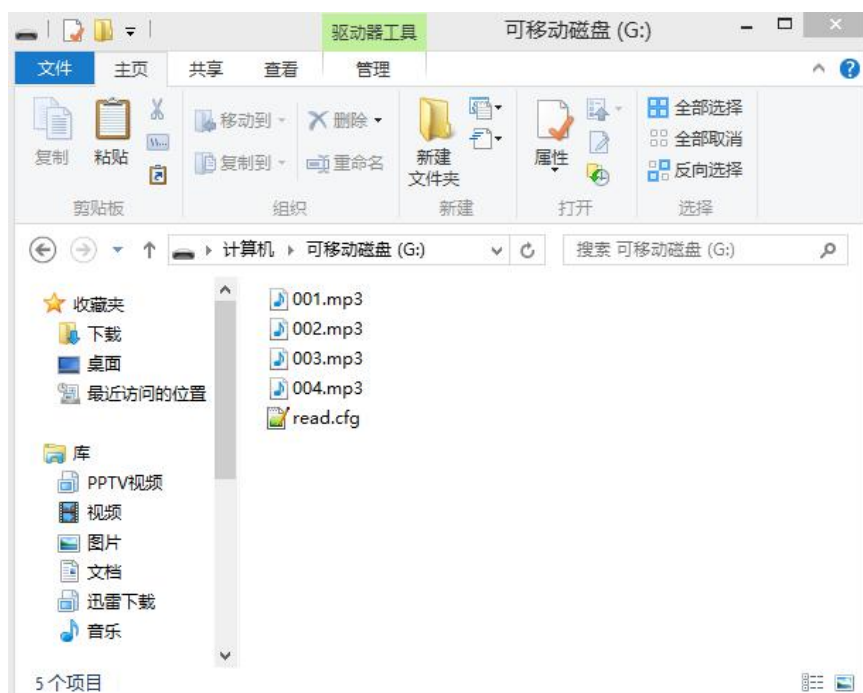
And change the file name “xxx.txt” to “read.cfg” as below, then the configuration file with level hold loop playback mode is made successfully.



### 3.2. Audio Files Loading

4 audio files need to be directly stored in the root directory of the SD card or the on-board flash memory. No folders can be in the 'root directory'. The arrangements of the audio files are managed by a physical indexing sequence. In other words, the file that is to be loaded first in the storage device will be associated with input “K1”. The last file to be loaded in the storage device will be associated with input “K4”. In order to guarantee a correct 'one-on-one' order, please refer to the following steps.

- 1). Build a new folder on the computer and put the 4 audio files in this new folder.
- 2). Rename the audio files from 001.mp3/wav to “004.mp3/wav”, and make sure they are ranked from “001.mp3/wav” to “004.mp3/wav” in order.
- 3). Plug in your SD card on computer or connect the module with computer through the micro USB(make sure the SD card or the on-board flash memory is empty).
- 4). Format the SD card into FAT or FAT32 or format the on-board flash into FAT.
- 5). Select all of the 4 audio files in the folder.
- 6). Right click on the first file (001.mp3/wav) and choose "Send to removable disk".
- 7). This should send the 4 audio files to the SD card or the on-board flash memory in a correct sequence.
- 8). Put the configuration file with the trigger mode needed into the root directory together with audio files as below, then refresh.





- 9). Safely remove the SD card and it onto the module or disconnect the flash memory of the module from computer.
- 10). Apply power to the module and push any of 4 buttons to play back a corresponding sound.

**Notes:** 1). *If the trigger mode of pulse interruptible one-on-one playback is needed, it's also workable if you don't put the configuration file. The module takes this trigger mode as the default if without a configuration file.*  
2). *When the SD card is plugged/ed into the module. Only audio files from the SD card will be played.*

## 4. UART RS232 Serial Control Mode

Serial control mode is provided for users who want to use a MCU to control this module. It's more flexible and is able to realize many more functions to control this module via serial serial commands through the ports TX and RX on the module..

### 4.1. Naming Rules of Audio Files and Folders

- 1). Audio files directly stored in the root directory of the storage device(micro SD card or the flash memory) need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav, .....3000.mp3/3000.wav(If there are more than 10 files, storing them in the micro SD card instead of the flash memory is suggested).

Here it works according to physical sequence when you copy the files from computer to SD card or SPI flash. For example, when the module receives a command to play the track 0001.mp3, it will play the 1<sup>st</sup> 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 one folder, then press "Ctrl+A" on the key board to select all, and press "Ctrl+C" to copy, and go back to the SD card or the SPI flash, and press "Ctrl+V" to past all the files into the SD card or SPI flash. 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). Ordinary folders must be renamed as 01, 02, 03.....99, and the audio files must be renamed as 001.mp3/001.wav, 002.mp3/002.wav, 003.mp3/003.wav, .....255.mp3/255.wav. 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".

**Note:** In addition, there are two special purposed folders "MP3" and "ADVERT" that can be chosen by users to use or not according to the actual needs. Audio files stored in these two folders need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav, .....3000.mp3/3000.wav.

### 4.2. Command Format

- Communication baud rate: 9600 bps
- Data bits: 1
- Checkout: none
- Flow Control: none



Format: \$S / Ver. / Number / Command / Feedback / Param_MSB / Param_LSB / Check_MSB / Check_LSB / \$O	
\$S	Start byte 0x7E
Ver.	Version byte, 0xFF by default
Number	Number of bytes from version info to Check_LSB, typically 0x06 (checksum not counted)
Command	Command byte
Feedback	0x01: Need feedback --send confirmation back to MCU; 0x00: No need feedback
Param_MSB	Most significant byte of parameter
Param_LSB	Least significant byte of parameter
Check_MSB	Most significant byte of checksum
Check_LSB	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 number is 6 bytes, and these 6 bytes are "FF 06 09 00 00 02". Start byte, end byte and checksum are not counted.

### 4.3. About Checksum

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

$$\text{Checksum (2 bytes)} = 0xFFFF - (\text{CMD} + \text{Feedback} + \text{Para\_MSB} + \text{Para\_LSB}) + 1$$

Normally it's okay whether users choose to use checksum or not, our module can receive a 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 the communication stability.

### 4.4. Serial Communication Commands

#### 4.4.1. Control commands

Command	Function	Note
0x01	Play Next	
0x02	Play Previous	
0x03	Specify playback of a track	See 4.6.1 for details
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	See 4.6.2 for details
0x08	Specify single repeat playback	See 4.6.3 for details
0x09	Specify playback of a device	See 4.6.4 for details
0x0A	Set sleep mode	See 4.6.5 for details
0x0B	Awake from sleep	
0x0C	Reset	



0x0D	Play	
0x0E	Pause	
0x0F	Specify playback of a track in a folder	See 4.6.6 for details
0x15	Stop playing the inserted advertisement	See 4.6.7 for details
0x16	Stop	
0x17	Specify repeat playback of a folder	See 4.6.8 for details
0x18	Set random playback	See 4.6.9 for details
0x19	Set repeat playback of current track	See 4.6.10 for details
0x21	Combination playback	See 4.6.11 for details
0x25	Insert an advertisement	See 4.6.12 for details

#### 4.4.2. Examples of sending control commands

Command Description	Serial Commands [with checksum]	Serial Commands [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
Increase volume	7E FF 06 04 00 00 00 FE F7 EF	7E FF 06 04 00 00 00 EF	
Decrease volume	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 the root directory	7E FF 06 08 00 00 01 FE F2 EF	7E FF 06 08 00 00 01 EF	Repeatedly play the 1st 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 of a device	7E FF 06 09 00 00 02 FE F0 EF	7E FF 06 09 00 00 02 EF	Specified device is SD card
	7E FF 06 09 00 00 04 FE EE EF	7E FF 06 09 00 00 04 EF	Specified device is on-board SPI flash
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 0 0 0 0 00 FE F0 EF	7E FF 06 0B 0 0 0 0 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"





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"
Stop playing the inserted advertisement	7E FF 06 15 00 00 00 FE E6 EF	7E FF 06 15 00 00 00 EF	Go back and continue to play the music interrupted
Stop	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 02 00 FE E2 EF	7E FF 06 17 00 02 00 EF	Specify repeat playback of the folder "02"
	7E FF 06 17 00 01 00 FE E3 EF	7E FF 06 17 00 01 00 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	Random playback of the whole device
Set repeat playback of current track	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

#### 4.4.3. Query commands

Command	Function	Note
0x3F	Query current online storage device	See 4.7.1 for details
0x40	Module returns an error data with this command	
0x41	Module reports a feedback with this command	
0x42	Query current status	See 4.7.2 for details
0x43	Query current volume	
0x48	Query number of tracks in the micro SD Card	
0x49	Query number of tracks in the SPI flash	
0x4C	Query current track in the micro SD Card	
0x4D	Query current track in the SPI flash	
0x4E	Query number of tracks in a folder	See 4.7.3 for details
0x4F	Query number of folders in the current storage device	See 4.7.4 for details

#### 4.4.4 Examples of sending query commands

Command Description	Serial Commands [with c heck sum]	Serial Commands [without c heck sum]	Notes
---------------------	-----------------------------------	--------------------------------------	-------



Query current online storage device	7E FF 06 3F 00 00 00 FE BC EF	7E FF 06 3F 00 00 00 EF	
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 in the 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 number of tracks in the SPI flash	7E FF 06 49 00 00 00 FE B2 EF	7E FF 06 49 00 00 00 EF	Total file numbers of current device
Query current track in the 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 current track in the SPI flash	7E FF 06 4D 00 00 00 FE AE EF	7E FF 06 4D 00 00 00 EF	Query the track being played
Query number of tracks in a folder	7E FF 06 4E 00 00 01 FE AC EF	7E FF 06 4E 00 01 00 EF	
Query number of folders in the current storage device	7E FF 06 4F 00 00 00 FE AC EF	7E FF 06 4F 00 00 00 EF	

## 4.5. Returned Data from Module

### 4.5.1. 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 7 E FF 06 3F 00 00 02 xx xx EF. 0x3F is the returned command by module, and 0x02 represents SD card is effective online. See the details as below.

Online Equipment	Returned Data
SD card online	7E FF 06 3F 00 00 02 xx xx EF
PC online	7E FF 06 3F 00 00 04 xx xx EF
SPI flash online	7E FF 06 3F 00 00 08 xx xx EF

- 3) MCU cannot 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.

### 4.5.2. Returned data after a track is finished playing

Track Played	Returned Data
1st track is finished playing in SD card	7E FF 06 3D 00 00 01 xx xx EF
2nd track is finished playing in SD card	7E FF 06 3D 00 00 02 xx xx EF
1st track is finished playing in SPI flash	7E FF 06 3E 00 00 01 xx xx EF
2nd track is finished playing in SPI flash	7E FF 06 3E 00 00 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 3D 00 00 01 xx xx EF. 0x3D represents SD card and 0x3E represents SPI flash. 0x00 and 0x01 represents the 1<sup>st</sup> track. 0x01 and 0xF4 represents the 500<sup>th</sup> track (0x01F4=500).
- 2) Because all of the files (tracks) in the storage device are read in physical sequence, the returned data still follow the physical sequence, which needs to be noted.

#### 4.5.3. 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.

#### 4.5.4. 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	insert error (an insert operation only can be done when a track is being played )
7E FF 06 40 00 00 08 xx xx EF	SD card reading failed (SD card pulled out or damaged)
7E FF 06 40 00 00 0A xx xx EF	Entered in to sleep mode

When the module is powered on, it reads SD card first and if SD card is not available it goes to SPI flash.

#### 4.5.5. Returned data after a storage device is plugged in or pull out

Status	Re turned Data
SD card is plugged in	7E FF 06 3A 00 00 02 xx xx EF
USB cable connected to PC is plugged in	7E FF 06 3A 00 00 04 xx xx EF



SD card is pulled out	7E FF 06 3B 00 00 02 xx xx EF
USB cable connected to PC is pulled out	7E FF 06 3B 00 00 04 xx xx EF

0x3A represents a device is plugged in while 0x3B represents a device is pull out. 0x02, 0x04 represent SD card and USB cable connected to PC respectively.

## 4.6. Detailed Annotation of Control Commands

### 4.6.1. Specify playback of a track in the root directory of the storage device (0x03)

The available selectable tracks are from 1<sup>st</sup> to 3000<sup>th</sup> in 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. Normally if the total track quantity is less than 1,000 in the storage device the response speed of the module is quicker than 50ms. If the total track quantity is more than 1000 the response speed is a little slower than 100ms.

- For example, if set the first track to be played, send the command 7E FF 06 03 00 00 01 FE F7 EF
  - 7E -- - Start byte
  - FF -- - Version Information
  - 06 --- Number of bytes
  - 03 --- Actual command (specify playback of a track)
  - 00 --- 0 x01 : need feedback ; 0 x00 : no need feedback
  - 00 --- Most significant byte of the track (MSB of Parameter)
  - 01 --- Least significant byte of the track (LSB of Parameter) FE --- Most significant byte of checksum (MSB of checksum) F7 -- - Least significant byte of checksum (LSB of checksum ) EF --- End byte 0xEF
- Regarding track selection, if the 100<sup>th</sup> song (track) is selected to be played, firstly convert 100 to hexadecimal. It is double-byte by default, i.e. 0 x006 4. MS B=0 x00 ; LS B=0 x64

### 4.6.2. Specify volume (0x06)

- Our system power-on default volume is level 30, if you want to set the volume, then directly send the corresponding command.
- For example, if specify the volume to level 15, send the command 7E FF 06 06 00 00 0F FF D5 EF.
- MSB=0x00; LSB=0x0F, 15 is converted to hexadecimal 0x000F.

### 4.6.3. Specify single repeat playback in the root directory of the storage device (0x08)

Repeatedly play 1 <sup>st</sup> track	7E FF 06 08 00 00 01 xx xx EF
Repeatedly play 2 <sup>nd</sup> track	7E FF 06 08 00 00 02 xx xx EF
Repeatedly play 3 <sup>rd</sup> track	7E FF 06 08 00 00 03 xx xx EF

During single repeat playback, you can still normally execute the operations Play/Pause, Previous, Next, Volume +/- and repeat playback. Users can specify single track playback or send the stop command to turn off single repeat playback status.

**4.6.4 Specify playback of a device (0x09)**

Specify playback of SD card	7E FF 06 09 00 00 02 FE F0 EF
Specify playback of SPI flash	7E FF 06 09 00 00 04 FE EE EF

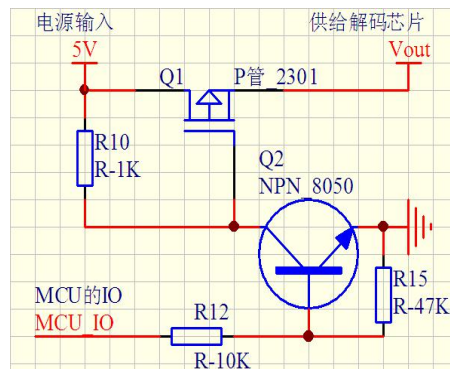
- 1) The module supports two types of playback devices (micro SD card and on-board SPI flash).
- 2) It automatically enters into the standby status after specifying a device, waiting for MCU to specify a track to play. It takes about 200ms from specifying device to the module finishes initialization of file system. Please wait for 200ms and then send the specified command to play a track.

**4.6.5. Set sleep mode, awake from sleep and reset (0x0A) (0X0B) (0X0C)**

Set sleep mode	7E FF 06 0A 00 00 00 FE F1 EF
Awake from sleep	7E FF 06 0B 00 0 0 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 other two ways other than sending the command to awake the module.
  - a. Specify playback of the storage device, either SPI flash or SD card.
  - b. Pull out the storage device and re-plug it in.
- 2) The reset is a soft reset. The reset time is 5-8 seconds. This allows 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.**



**4.6.6. Specify playback of a track in a folder (0X0F)**

Specify playback of track 001 in the folder 01	7E FF 06 0F 00 01 01 xx xx EF
Specify playback of track 100 in the folder 11	7E FF 06 0F 00 0B 64 xx xx EF
Specify playback of track 255 in the 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. This feature supports MP3 and WAV audio formats.
- 4). The following two images illustrates the naming method of folders and files.



**4.6.7. Stop (0x15) (0x16)**

Stop playing the inserted advertisement	7E FF 06 15 00 00 00 FE E6 EF
Stop	7E FF 06 16 00 00 00 FE E5 EF

During playback of the module, there are two modes to stop. One is to stop playing the insert advertisement, and go back and continue to play the music interrupted and the other mode is to stop all playback (stop decoding). For example, suppose the module is playing an insert advertisement and now if send a stop command 0x16, it will stop all playback tasks.

**4.6.8. Specify repeat playback of a folder (0x17)**

Specify repeat playback of the folder "02"	7E FF 06 17 00 00 02 FE E2 E F
Specify re pea t playback of the folder "01"	7E FF 06 17 00 00 01 FE E3 E F

The folder names must be 01- 99, and no more than 99. After sending the command, it repeatedly plays the tracks in the specific folder and it will not stop until it receives a command to stop.

**4.6.9. Set random playback (0x18)**

Random playback of the whole s to rage de vice	7E FF 06 18 00 00 00 FE E3 EF
--	-------------------------------

This command is used to randomly play all of the tracks in the storage device according to physical sequence and no matter if there is a folder or not in the device.

#### 4.6.10. Set repeat playback of current track (0x19)

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

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. If you need to turn off repeat playback, just send the turn-off command.

#### 4.5.11 Combination playback (0x21)

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.

#### 4.6.12. Insert an advertisement (0x25)

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/wav", 002.mp3/wav. 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.

## 4.7. Detailed Annotation of Query Commands

### 4.7.1. Query current online storage device

Query current online storage device	7E FF 06 3F 00 00 00 FE BC EF
-------------------------------------	-------------------------------

When the module is working, users can use the command as above (0x3F) to query the status of the online storage devices. For example, if the module returns the data 7E FF 06 3F 00 00 0A xx xx EF, LSB 0x0A (0000 1010) represents SD card online. If LSB is 0x1F (0000 1111), it represents both of SD card and PC online (PC online means module is connecting with PC via a USB cable).

### 4.7.2. Query current status

Query current online storage device	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 (0 x42)

2). Interpretation of returned data

Returned Data	Status
7E FF 06 42 00 02 01 xx xx EF	SD card is being played
7E FF 06 42 00 02 02 xx xx EF	SD card is paused playing
7E FF 06 42 00 08 00 xx xx EF	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	
0x02	SD card	0x00	Stopped





0x08	SPI flash	0x01	Playing
0x10	Module in sleep mode	0x02	Paused

#### 4.7.3. 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.

#### 4.7.4. Query number of total folders in current storage device

Query number of total folders in current storage device	7E FF 06 4F 00 00 00 FE AC EF
---	-------------------------------

Users can query this 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 do not build any sub-folders in a folder).

### 4.8. About Delay of Serial Programming

- 1). After the module is powered on, it needs about 500ms to 1500ms (depending on the track quantities in the storage device) to initialize. After that, some data related to initialization returns to MCU. Users can choose to ignore these data.
- 2). After specifying playback of a device (SD card or SPI flash), it needs 200ms delay before sending the command to execute the relative operation.
- 3). The module processes a serial data per 10ms, so when MCU continuously sends commands one by one, 20ms delay must be added before sending next command, otherwise the command MCU sends out will not be executed.
- 4). If specifying playback of a track in a folder, the delay must be longer than 40ms, as it needs time to target a track in a folder. And even so song as sending the commands related to query a track or a folder, 40ms delay is required.