Industrial MP3 Player / 7-Trigger Audio Player for Museums and Talking Displays

User Manual

Model No.: FN-T718

Version: V2.1

1. Overviews

FN-T718 is an industrial audio player/7-trigger MP3 player designed to provide stereo audio output for museum exhibits, talking displays and other industrial/commercial applications. It has not only a built-in 8MB flash memory but also a slot for a micro SD card as well as a USB connector for a USB pen drive that hold the sound file(s), and connectors for up to 7 push buttons for sound activation. Upon activation the unit will immediately play the sound file once (by default) and become ready for the next activation. And it supports 10-type trigger modes to meet different needs in different applications. A trigger mode can be set with a config file easily. Besides, it's possible to switch on an external LED light during playback of a sound file. Please note that a computer is required for copying sound files onto a storage device (built-in flash memory, micro SD card or USB pen drive).

1.1. Features

- ✧ Negative trigger version, positive trigger version and 5 triggers+RS485 version are available.
- ✧ Built-in high quality audio decoder and 8MB SPI flash memory.
- ✧ Supports inserting a USB flash drive (into the USB-A port) or a micro SD card as the extended storage device.
  -When there is an inserted USB flash drive or micro SD card, the audio player will always take the inserted USB flash drive or micro SD card in priority.

- ✧ Supports max. 32GB USB flash drive and max. 32GB micro SD card.
- ✧ 7 trigger inputs available and they can be connected with as many as 7 push buttons/switches/relays, etc.

  -5 triggers+RS485 version is also available for those who need to control the player via RS485 communication.
- ✧ Supports 10 types of trigger modes and each trigger mode can be set easily through a config file (text document).
- ✧ Built-in 50W*2 class D amplifier and the audio output is powerful.
- ✧ Upload audio files easily on computer via USB connection.

  -Connect the USB-C port to computer using a USB data cable.

  -The internal memory and the micro SD card (if it is inserted) will be detected as USB flash drives on computer.
- ✧ Sound volume is adjustable through turning the volume knob.
- ✧ Able to drive an external equipment like a warning light or a motor simultaneously when it is playing a sound.
- ✧ Equipped with a 3.5mm audio output jack that can be connected to an external amplifier or an active speaker.
- ✧ Adopts solid and durable aluminium alloy case.
- ✧ Industrial grade design and strong anti-jamming capability.

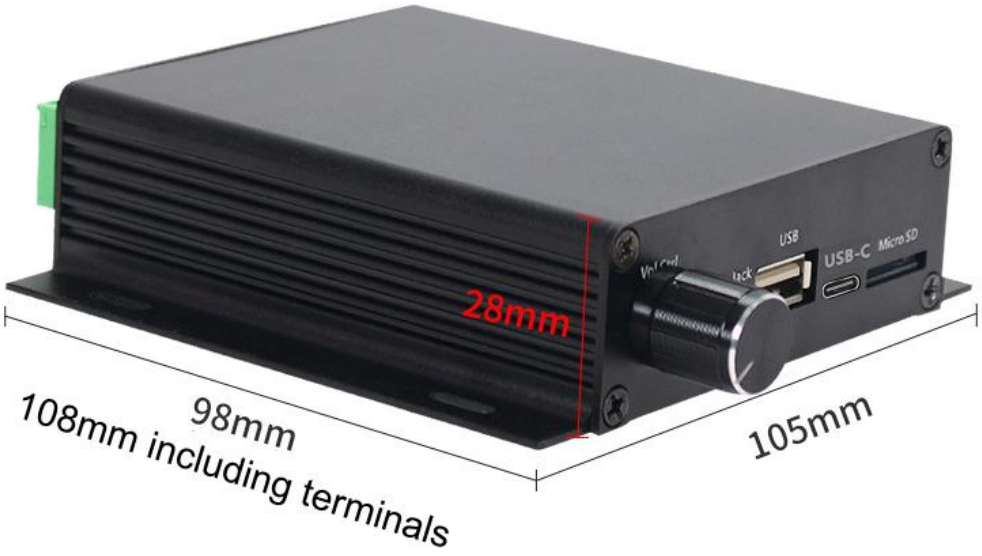
1.2. Technical Parameters

- ✧ Working voltage: 12-24V DC
- ✧ Working current: ≥8500mA @12V (when it has two 50W 8 ohm speakers)

  ≥4250mA @24V (when it has two 50W 8 ohm speakers)
- ✧ Power consumption: ≤100W
- ✧ Flash memory size: 8MB
- ✧ Audio format: MP3 or WAV

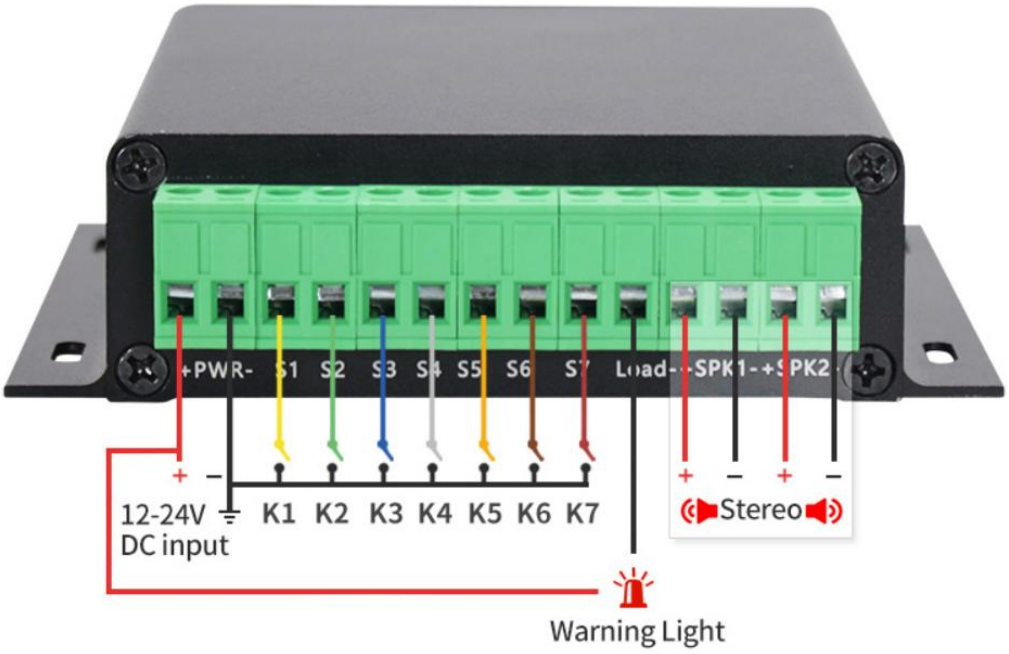
1.3. Interfaces

1.4. Dimensions



2. Examples of Wiring Connections

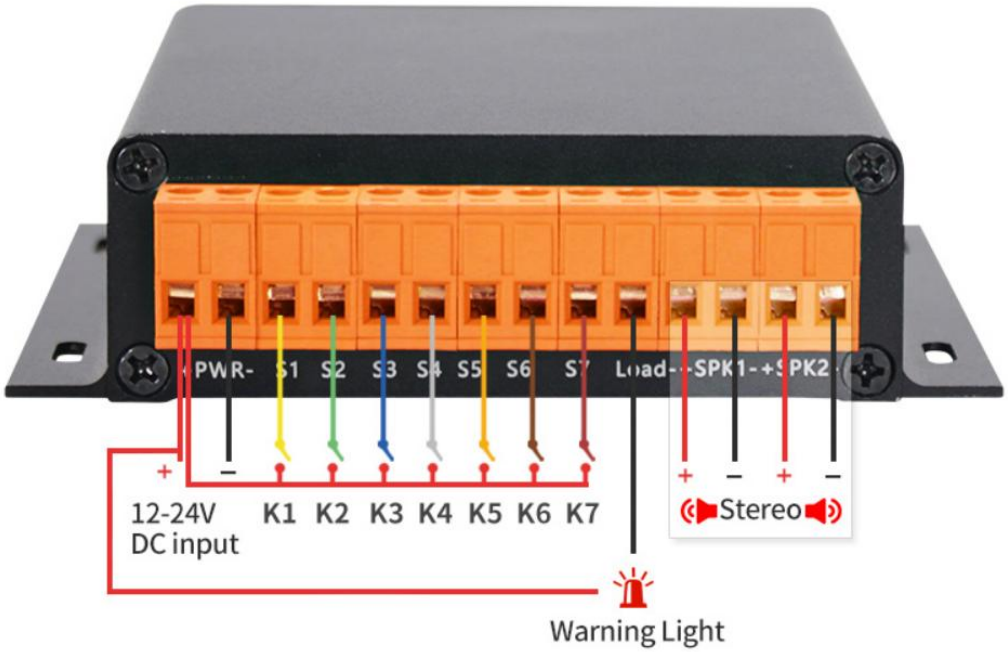
2.1. Negative Trigger Version



Note: The negative trigger version also can be activated with a low level signal (0V) from a PLC or controller.

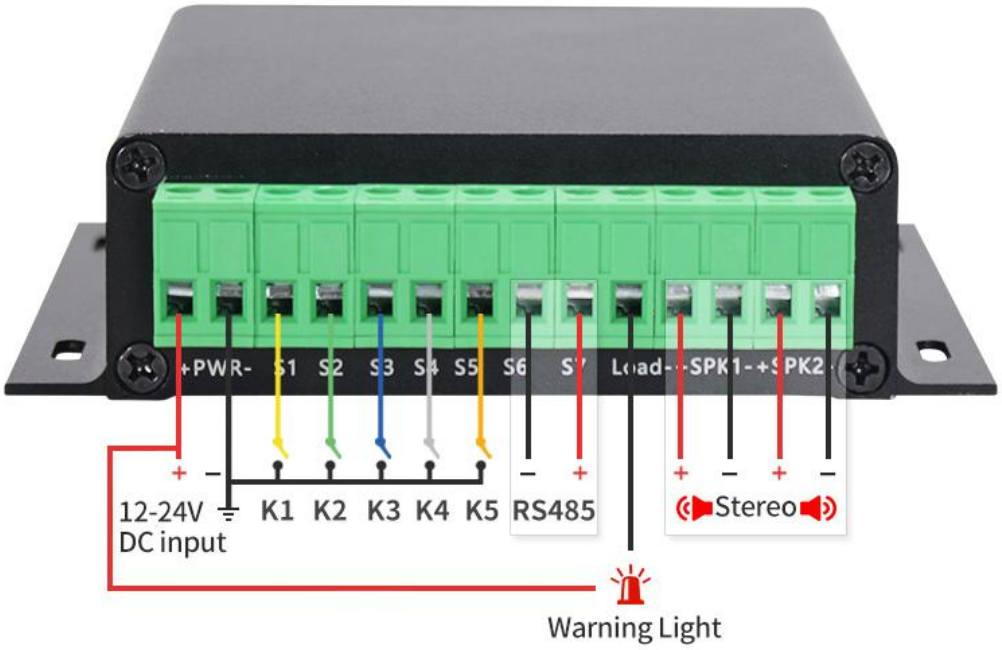
The PLC or controller needs to share ground with the player in advance.

2.2. Positive Trigger Version



Note: The positive trigger version also can be activated with a high level signal (12-24V) from a PLC or controller. The PLC or controller needs to share ground with the player in advance.

2.3. 5 Triggers + RS485 Version



3. Operation Guide for Using Triggers

3.1. Select a Trigger Mode

There are 10 trigger modes available for users to choose according to the actual needs. Any of these 10 trigger

modes can be set/acquired through a config file, which is a text file (.txt). Users just need to fill in a number that is corresponding to a trigger mode in a new built text file. Save it and rename the file “Config” or any name you like, then put it in the root directory of the built-in flash memory/micro SD card/USB flash drive together with the audio files or folders. Please refer to the below sheet about the number and the associated trigger modes.

Number in Config File	Corresponding Trigger Mode
0	Short press the button to start playing, and during playing, if you press the button again, the playback will be interruptible and it will play from the beginning.
1	Press the button and hold to start playing repeatedly, and when the button is released the device stops playing.
2	Short press the button to start playing, and during playing, if you press the button again, the playback will be NOT interruptible.
3	In this mode, K1 works as Next, K2 Previous, K3 Play/Pause, K4 Stop, K5 Volume Up, K6 Volume Down, K7 Random playback (it plays a sound randomly after each triggering). Special function: When K1 is short-circuited to GND first and use this mode in the config file and once power is applied, the device is able to play a sound in a loop or play multiple sounds one by one in a loop until power is off.
4	In this mode, audio file 000.mp3 will play in a loop when power is applied. Any button from K1 to K7 (associated 001.mp3 to 007.mp3) can interrupt it and switch to play the associated sound file. Once the associated sound file finishes the playback, audio file 000.mp3 will continue to play in a loop.
5	Each button from K1 to K7 have their associated folders from 01 to 07 respectively. Each folder can store multiple files (for example from 001.mp3 to 030.mp3). Short press the button to play audio file 001.mp3, and short press again to play the next file 002.mp3, and so on. When the last audio file finish playing, short press again to go back to playing the file 001.mp3. During playing, if you press the button again, the playback will be interruptible and it will play next file.
6	The same as mode “5”, but during playing, the playback will be uninterruptible.
7	Each button from K1 to K7 have their associated folders from 01 to 07 respectively. Each folder can store multiple files (for example from 001.mp3 to 030.mp3). Short press the button to play all of the sound files in order one by one. During playing, if you press the button again, the playback will be interruptible and it will play next file and continue to play the rest of the files. When the last audio file finishes playback, it’ll stop.
8	Combination of the first four trigger ports (S1-S4) for playback of maximum 15 audio files (pulse interruptible)
9	Combination of the first four trigger ports (S1-S4) for playback of maximum 15 audio files (hold and play in a loop)

Note: If there is no config file on the built-in flash memory/micro SD card/USB flash drive, the device will always works with the mode “0” by default.

- **Reference Table for Combination of the First Four Trigger Ports for Trigger Mode 8 and 9**

Corresponding Sound	Trigger Ports			
	S1	S2	S3	S4
Play 1st sound	0	1	1	1
Play 2nd sound	1	0	1	1
Play 3rd sound	0	0	1	1
Play 4th sound	1	1	0	1
Play 5th sound	0	1	0	1
Play 6th sound	1	0	0	1
Play 7th sound	0	0	0	1
Play 8th sound	1	1	1	0
Play 9th sound	0	1	1	0
Play 10th sound	1	0	1	0
Play 11th sound	0	0	1	0
Play 12th sound	1	1	0	0
Play 13th sound	0	1	0	0
Play 14th sound	1	0	0	0
Play 15th sound	0	0	0	0

Note: "0" represents a low level (negative) signal while "1" represents a high level (positive) signal or idle.

3.2. How to create a config file

Users can create a config file without or with volume setting. The volume knob on the device controls the speaker outputs only, so if some users want to adjust volume from the 3.5mm audio jack output, it's necessary to create a config file with volume setting, otherwise a config file without volume setting is enough. By the way, in the same time the speaker outputs is also subject to the volume setting in the config file.

3.2.1. Create a config file without volume setting

1). Firstly create a new text file on computer (desktop or somewhere else).

2). Open it and enter a number (mode) you need. Suppose you need mode "1", just enter "1". See below.

3). Save it and close it.

4). Rename the file “Config” or any other name you like.

3.2.2. Create a config file with volume setting

There are thirty-one volume levels from “00” to “30”. “00” means mute while “30” means the max. volume level.

1). Firstly create a new text file on computer (desktop or somewhere else).

2). Open it and enter a number (mode) you need, and enter a volume level right after the mode number. See below.

3). Save it and close it.

4). Rename the file “Config” or any other name you like.

3.3. Audio Files Loading/Updating

Users can use a micro USB data cable to connect the device to computer. The built-in flash memory will be detected as a USB flash drive on computer. If the built-in flash memory is not large enough to store your audio file, you can use a micro SD card or USB flash drive instead. When there is an inserted micro SD card, there will be two simulated USB flash drives on computer (one is the built-in flash memory and the other one is the micro SD card), so please note to recognize.

3.3.1. For Trigger Mode 0-3, 8-9

To these four modes, the audio files need to be placed on the root directory of the storage device (built-in flash memory/micro SD card/USB flash drive). The arrangements of the audio files are managed by the physical index

order. In other words, the file that is to be loaded first in the storage device will be associated with trigger input “K1”. The last file to be loaded in the storage device will be associated with trigger input “K7”. In order to guarantee a correct ‘one-on-one’ order, please refer to the following steps.

1). Build a new folder somewhere on computer and put the 7 audio files in this new folder.

2). Rename the audio files from 001.mp3 to “007.mp3”, and make sure they are ranked from “001.mp3 to “007.mp3” in order.

3). Connect the device to computer through a USB data cable.

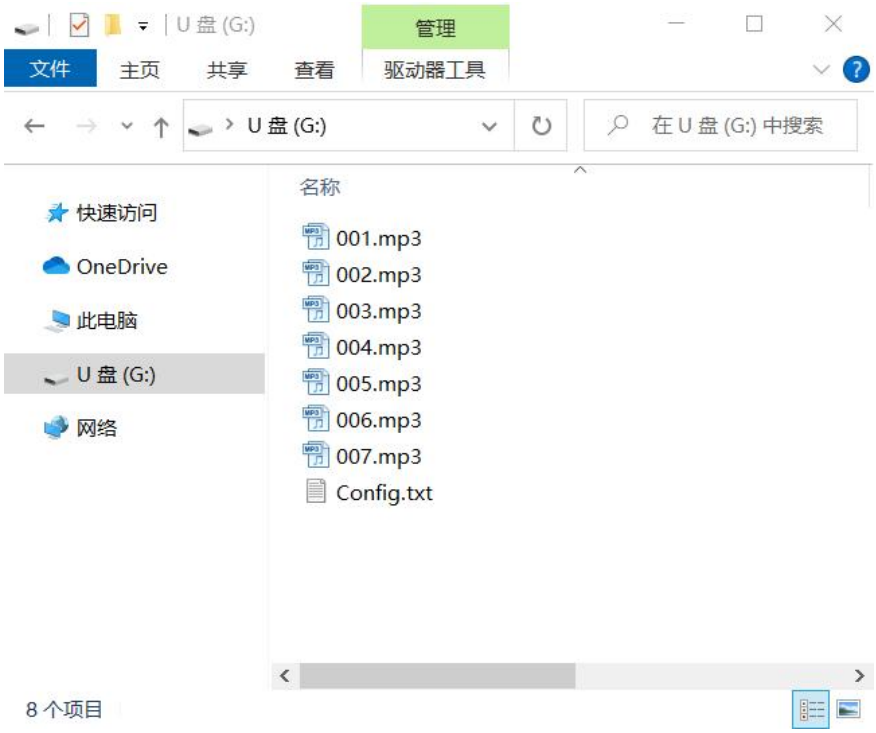
4). Delete the pre-loaded audio files for testing purpose at factory, or empty the micro SD card if it has other files inside.

5). Go back to the folder and select all of the 7 audio files in the folder.

6). Right click on the first file (001.mp3) and choose "Send to removable disk" or “Send to USB flash drive”.

7). This will send the 7 audio files to the internal flash memory or micro SD card one by one in a correct order.

8). Move the prepared config file with the trigger mode needed onto the root directory together with the audio files as below then refresh. Sure, you can also directly create a config file in the memory.



9). Safely remove the USB connection from computer.

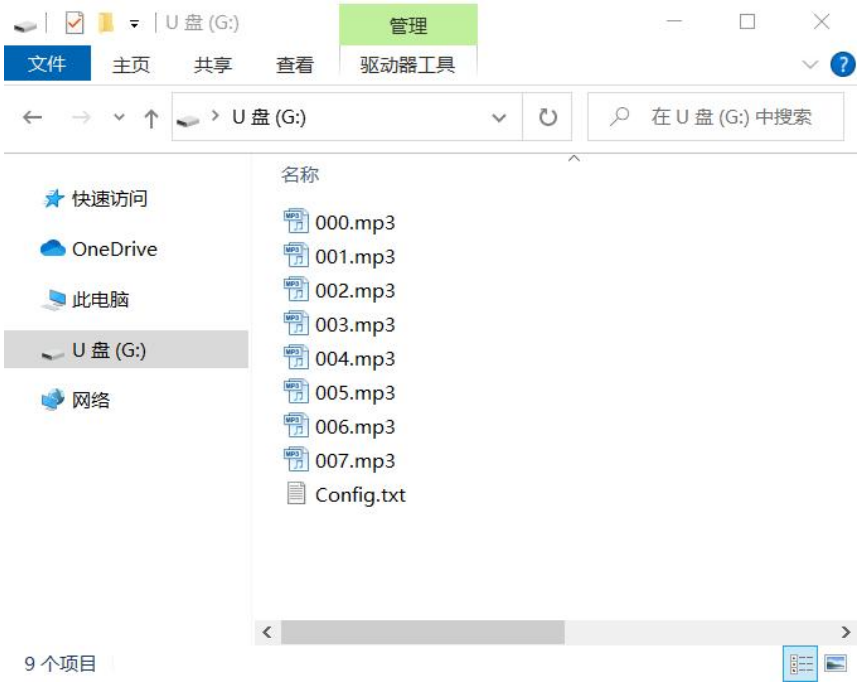
Notes:

1). When you use the trigger mode “3”, you can definitely place more than 7 audio files.

2). When you use the trigger mode “8” or “9”, you can place as many as 15 audio files.

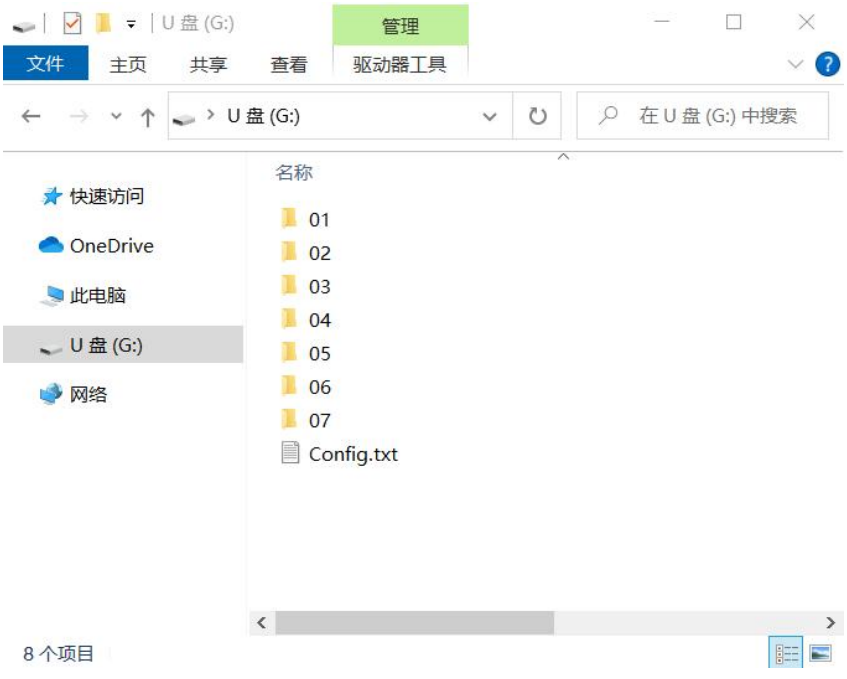
3.3.2. For Trigger Mode 4

In the same way, the audio files and the config file need to be placed on the root directory of the storage device (built-in flash memory/micro SD card/USB flash drive). In this mode, the device recognizes the files by name instead of physical index order, so it'll be alright to directly copy your audio files onto the storage device and rename the files 000.mp3-007.mp3 according to your actual needs. Please refer to the screenshot below.

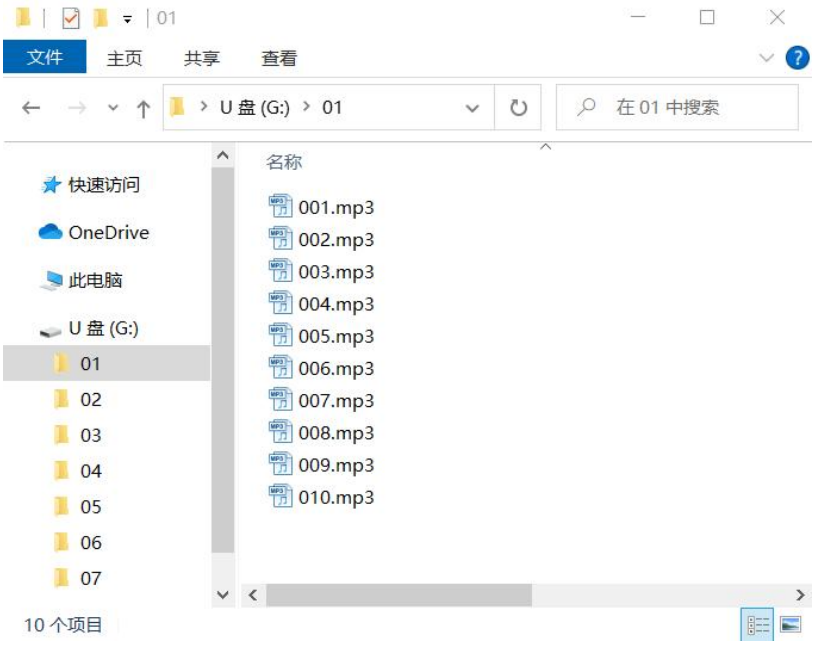


3.3.3. For Trigger Mode 5-7

At first, you need to create seven folders on the storage device (built-in flash memory/micro SD card/USB flash drive), and rename them 01, 02, 03, till 07, then put the config file together with the folders on the root directory. Of course, if you don't have to use so many buttons, then you don't need to create as many as seven folders. See below.



After that, copy your audio files to each associated folder, and rename them 001.mp3, 002.mp3, and so on. In these three modes, the device recognize the audio files by physical index order, so when you copy audio files to each folder, please note the copy order. See the screenshot below.



Note: When you rename a file, you can still keep the original file name and you can rename it like 001-Never Say Goodbye.mp3, 002-Season in the Sun.mp3, 003-Angel.mp3, and so on. This rule applies to all of the trigger modes.

4. Operation Guide for Using RS485 Communication (available only with the 5 triggers+RS485 version)

4.1. Command Format

Baud rate: 9600 (factory default setting)

Data bits: 8

Stop 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 Param_LSB, typically 0x06 (checksum not counted)
Command	Command byte

Feedback	0x00: No feedback when a command is received, but there is a feedback when a sound has finished playing 0x01: There is a feedback when a command is received, and there is another feedback when a sound has finished playing 0x02: No feedback anytime
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.

4.2. About Checksum

Start byte and end byte don’t need to be counted. Accumulate the 6 bytes in the middle, and then invert the accumulated sum and add 1 to get the checksum. Or subtract the accumulated sum of the 6 bytes from 0 to get the same result.

For example, if we play the next track, we need to send the command data “7E FF 06 01 00 00 00 FE FA EF”.

The number of bytes is 6, which are [FF 06 01 00 00 00] respectively. The resulting checksum is “FE FA”.

Process of getting the checksum at the sender:

1). Process of adding: FF+06+01+00+00+00=0x0106

2). Process of negating and adding 1: 0x0106 negated=FEF9+1=0xFEFA

Process of checking at the receiver:

Process of subtracting: 0-0x0106=0xFEFA

Normally it’s okay whether users choose to use the checksum or not, our audio player can receive a serial data with or without checksum, but some of users use a MCU without crystal oscillator, so if so we suggest users to add checksum to make sure the communication stability. If you don’t understand this part here, you can ignore the checksum directly.

4.3. Naming Rules for Audio Files (Tracks) and Folders

1). Audio files (tracks) directly stored in the root directory of the storage device(SD card or USB flash drive) need to be renamed to 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav, ...3000.mp3/3000.wav. Here it works according to physical index order when you copy the files from computer to SD card or USB flash drive. For example,

when the audio player 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 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 USB flash drive, and press “Ctrl+V” to past all the files into the SD card or USB flash drive. 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 must be renamed to 01, 02, 03.....99, and the audio files in the folder must be renamed to 001.mp3/001.wav, 002.mp3/002.wav, 003.mp3/003.wav, ...255.mp3/255.wav. It is also possible to keep the original audio file name when you rename a file. For example, the original audio file name is “Yesterday Once More.mp3”, you can rename it to “001Yesterday Once More.mp3”.

4.4. RS485 Communication Commands

4.4.1. Control commands

Command	Function	Note
0x01	Next	
0x02	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 playback of a track in a loop	See 4.6.3 for details
0x09	Specify a storage device	See 4.6.4 for details
0x0B	Set baud rate	See 4.6.5 for details
0x0C	Reset	
0x0D	Play	
0x0E	Pause	
0x0F	Specify playback of a track in a folder	See 4.6.6 for details
0x10	Specify loop playback of a track in a folder	See 4.6.7 for details
0x11	Set loop playback of all tracks	
0x16	Stop	
0x17	Specify loop playback of all tracks in a folder	See 4.6.8 for details
0x18	Set random playback	See 4.6.9 for details
0x19	Set loop playback for currently playing track	See 4.6.10 for details

0x21	Set combination playback (there is returned data after a track is played)	See 4.6.11 for details
0x22	Set combination playback (there is no returned data after a track is played)	
0x25	Insert an advertisement	See 4.6.12 for details
0xC0	Set a device address	See 4.6.13 for details

4.4.2. Examples of sending the control commands

Command Description	Serial Commands [with checksum]	Serial Commands [without checksum]	Note
Next	7E FF 06 01 00 00 00 FE FA EF	7E FF 06 01 00 00 00 EF	
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 playback of a track in a loop	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 a storage device	7E FF 06 09 00 00 01 FE F1 EF	7E FF 06 09 00 00 01 EF	Specified device is USB flash drive
	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 ED EF	7E FF 06 09 00 00 04 EF	Specified device is built-in SPI flash
Set baud rate	7E 01 06 0B 00 00 02 FF EC EF	7E 01 06 0B 00 00 02 EF	Set baud rate to 19200
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”
Set loop playback of all tracks	7E FF 06 11 00 00 01 FE E9 EF	7E FF 06 11 00 00 01 EF	Start loop playback
	7E FF 06 11 00 00 00 FE EA EF	7E FF 06 11 00 00 00 EF	End loop playback
Stop	7E FF 06 16 00 00 00 FE E5 EF	7E FF 06 16 00 00 00 EF	Stop all playback tasks
Specify loop playback of all tracks in a folder	7E FF 06 17 00 00 01 FE E3 EF	7E FF 06 17 00 00 01 EF	Specify loop playback of all the tracks in the folder “01”
	7E FF 06 17 00 00 02 FE E2 EF	7E FF 06 17 00 00 02 EF	Specify loop playback of all the tracks in the folder “02”

Set random playback	7E FF 06 18 00 00 00 FE E3 EF	7E FF 06 18 00 00 00 EF	All tracks in the storage device are covered
Set loop playback for currently playing track	7E FF 06 19 00 00 00 FE E2 EF	7E FF 06 19 00 00 00 EF	Start loop playback
	7E FF 06 19 00 00 01 FE E1 EF	7E FF 06 19 00 00 01 EF	End loop playback
Set a device address	7E FF 06 C0 00 00 01 FE 3A EF	7E FF 06 C0 00 00 01 EF	Set the device address to 01
	7E FF 06 C0 00 00 63 FD D8 EF	7E FF 06 C0 00 00 63 EF	Set the device address to 99

4.4.3. Query commands

Command	Function	Note
0x3F	Query current online storage device	See 4.7.1 for details
0x40	Audio player returns an error data with this command	
0x41	Audio player reports a feedback with this command	
0x42	Query current status	See 4.7.2 for details
0x43	Query current volume	See 4.7.3 for details
0x47	Query total number of tracks in the USB flash drive	See 4.7.4 for details
0x48	Query total number of tracks in the SD Card	
0x49	Query total number of tracks in the SPI flash	
0x4B	Query currently playing track in the USB flash drive	See 4.7.5 for details
0x4C	Query currently playing track in the SD Card	
0x4D	Query currently playing track in the SPI flash	

4.4.4. Examples of sending the query commands

Command Description	Serial Commands [with checksum]	Serial Commands [without checksum]	Note
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 total number of tracks in USB flash drive	7E FF 06 47 00 00 00 FE B4 EF	7E FF 06 47 00 00 00 EF	
Query total number of tracks in SD card	7E FF 06 48 00 00 00 FE B3 EF	7E FF 06 48 00 00 00 EF	
Query currently playing track in USB flash drive	7E FF 06 4B 00 00 00 FE B0 EF	7E FF 06 4B 00 00 00 EF	Query the track being played
Query currently playing track in 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 currently playing track in 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 total 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 total number of folders in current storage device	7E FF 06 4F 00 00 00 FE AC EF	7E FF 06 4F 00 00 00 EF	USB flash drive or SD card

4.5. Data Returned by the Audio Player

4.5.1. Data returned when the audio player is powered on

1). After the audio player is powered on, normally it needs about no more than 500ms initialization time (depending on the actual track quantities in the storage device). Once the initialization is finished, the audio player returns a data to the controller. 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 data returned by the audio player is related to the current valid storage device/online device. For example, the audio player returns 7E FF 06 3F 00 00 01 xx xx EF. 0x3F is the command returned by the audio player, and 0x01 represents USB flash drive is online. See the details as below.

Status of Storage Device	Data Returned
USB flash drive is online	7E FF 06 3F 00 00 01 xx xx EF
SD card is online	7E FF 06 3F 00 00 02 xx xx EF
SPI flash is online	7E FF 06 3F 00 00 08 xx xx EF
USB flash drive and SPI flash are online simultaneously	7E FF 06 3F 00 00 09 xx xx EF
SD card and SPI flash are online simultaneously	7E FF 06 3F 00 00 0A xx xx EF

3). The controller must wait until the device initialization is completed and the return command is issued before sending the corresponding control command, otherwise the device will not process the sent command; it will also affect the normal initialization of the device.

4.5.2. Data returned after a track has finished playing

Track Played	Data Returned
1st track in USB flash drive has finished playing	7E FF 06 3C 00 00 01 xx xx EF
2nd track in USB flash drive has finished playing	7E FF 06 3C 00 00 02 xx xx EF
1st track in SD card has finished playing	7E FF 06 3D 00 00 01 xx xx EF
2nd track in SD card has finished playing	7E FF 06 3D 00 00 02 xx xx EF
1st track in SPI flash has finished playing	7E FF 06 3E 00 00 01 xx xx EF
2nd track in SPI flash has finished playing	7E FF 06 3E 00 00 02 xx xx EF

1). For example, the returned data is 7E FF 06 3C 00 00 01 xx xx EF. 0x3C represents USB flash drive and 0x3D represents SD card. 0x00 and 0x01 represents the 1st track. 0x01 and 0xF4 represents the 500th track(0x01F4=500).

2). Because all of the files(tracks) in the storage device are read in physical index order, the returned data still follow this rule, which needs to be noted.

4.5.3. Feedback from the audio player

Audio player 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 the system is added. Once there is a feedback to the controller from the audio player, it means the audio player has successfully received the command that the controller sent out. 0x41 is the returned command by the system.

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

4.5.4. Errors returned

Returned Data of Errors	Explanatory Note
7E FF 06 40 00 00 01 xx xx EF	System busy(this info is returned when the initialization is still ongoing)
7E FF 06 40 00 00 02 xx xx EF	Currently in sleep mode(supports specifying a storage device only 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 range
7E FF 06 40 00 00 06 xx xx EF	Specified track is not found
7E FF 06 40 00 00 08 xx xx EF	Data does not comply with the rules

4.5.5. Data returned after a storage device is plugged in or pulled out

Status	Returned Data
USB flash drive is plugged in	7E FF 06 3A 00 00 01 xx xx EF
SD card is plugged in	7E FF 06 3A 00 00 02 xx xx EF
USB flash drive is pulled out	7E FF 06 3B 00 00 01 xx xx EF
SD card is pulled out	7E FF 06 3B 00 00 02 xx xx EF

0x3A represents a device is plugged in while 0x3B represents a device is pulled out. 0x01 and 0x02 represent USB flash drive and SD card respectively.

4.6. Detailed Annotations for Some of the Control Commands

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

The available selective tracks is from 1st to 3000th 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. With this command, a track is specified in physical index order.

1). For example, if set the first track to be played, send the command data 7E FF 06 **03** 00 00 01 FF E7 EF

7E --- Start byte

FF --- Version information

06 --- Number of bytes

03 --- Function code/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 0xEF

2). Regarding track selection, if the 100th track (audio file) is selected to be played, firstly convert 100 to hexadecimal. It is double-byte by default, i.e. 0x0064. MSB/DH=0x00; LSB/DL=0x64

4.6.2. Specify volume (0x06)

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

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/DH=0x00; LSB/DL=0x0F, 15 is converted to hexadecimal 0x000F.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **06** 00 00 00 xx xx EF	Set the volume to level 0 (mute)
7E FF 06 **06** 00 00 19 xx xx EF	Set the volume to level 25

4.6.3. Specify playback of a track in a loop (0x08)

1). This command is only valid to the tracks in the root directory of the storage device.

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

Command data reference:

Command Data (xx xx represents CRC)	Note
7E FF 06 **08** 00 00 01 xx xx EF	Specify playback of 1st track
7E FF 06 **08** 00 00 02 xx xx EF	Specify playback of 2nd track
7E FF 06 **08** 00 00 03 xx xx EF	Specify playback of 3rd track

4.6.4. Specify a storage device (0x09)

The system automatically enters into the standby status after specifying a storage device, awaiting the controller to specify a track to play. It'll take about 200ms from specifying a storage device to the player finishes initialization of file system. Please wait for 200ms and then send the specified command to play a track.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **09** 00 00 01 FE F1 EF	Specify playback of USB flash drive
7E FF 06 **09** 00 00 02 FE F0 EF	Specify playback of SD card
7E FF 06 **09** 00 00 04 FE ED EF	Specify playback of built-in SPI flash

Note: If you just use the built-in SPI flash only, you don't need to specify a storage device. When you use a USB flash drive or a SD card at the same time, you need to send such a command to specify which storage device to be work with the system.

4.6.5. Set baud rate (0x0B)

1). The factory default baud rate is 9600. If you need to change the baud rate, you can use the 0x0B command to modify it.

2). After setting the baud rate, please wait for 1 second and then send the reset command (0x0C), or re-power on the audio player to take it effect.

3). LSB of Parameter is used for setting the baud rate. Please refer to the sheet below for details.

LSB of Parameter	Corresponding baud rate	LSB of Parameter	Corresponding baud rate
0x01	9600	0x05	115200
0x02	19200	0x06	256000
0x03	38400	0x09	4800
0x04	57600	/	/

For example, if you need to set the baud rate to 19200, just send the command data 7E 01 06 **0B** 00 00 02 FF EC EF.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **0B** 00 00 01 xx xx EF	Set the baud rate to 9600

7E FF 06 **0B** 00 00 05 xx xx EF	Set the baud rate to 115200

4.6.6. Specify playback of a track in a folder (0x0F)

1). The folders need to be renamed to "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 data 7E FF 06 0F 00 01 64 xx xx EF.

MSB: represents the folder number (supports max. 99 folders from 01-99)

LSB: represents the track number (supports max. 255 tracks from 0x01 to 0xFF)

3). With this command, the audio player will play the specified track once and then stop.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **0F** 00 02 01 xx xx EF	Specify playback of track 001 in the folder 02
7E FF 06 **0F** 00 01 64 xx xx EF	Specify playback of track 100 in the folder 01
7E FF 06 **0F** 00 63 FF xx xx EF	Specify playback of track 255 in the folder 99

4.6.7. Specify loop playback of a track in a folder (0x10)

1). The folders need to be renamed to "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 “010.mp3” in the folder "02", send the command data 7E FF 06 **10** 00 02 0A xx xx EF.

MSB: represents the folder number (supports max. 99 folders from 01-99)

LSB: represents the track number (supports max. 255 tracks from 0x01 to 0xFF)

3). During loop playback of a track in a folder, the functions of Play, Pause, Previous and Next still can be performed. These operation commands will not interrupt the current loop playback state, so if the command for next track is sent, the next track will be played in a loop.

4). Users can send a stop command to end the loop playback. After the track is played each time, there will be a return data (can be set not to return through the feedback byte).

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **10** 00 01 01 xx xx EF	Specify loop playback of the track 001.mp3 in the folder 01
7E FF 06 **10** 00 01 02 xx xx EF	Specify loop playback of the track 002.mp3 in the folder 01
7E FF 06 **10** 00 02 0A xx xx EF	Specify loop playback of the track 010.mp3 in the folder 02
7E FF 06 **10** 00 02 63 xx xx EF	Specify loop playback of the track 099.mp3 in the folder 02

4.6.8. Specify loop playback of all tracks in a folder (0x17)

1). The folder must be renamed to 01, 01, 03.....99, and no more than 99. After sending the command, it will play all the tracks in the specific folder in a loop, and it will not stop until it receives a command to stop.

2). Once the tracks in the specified folder are looped, the functions of Play, Pause, Previous and Next still can be performed. These operation commands will not interrupt the current loop playback state. In other words, after sending the command for Next, the tracks in this folder will still be looped.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **17** 00 00 01 xx xx EF	Specify loop playback of all tracks in the folder “01”
7E FF 06 **17** 00 00 02 xx xx EF	Specify loop playback of all tracks in the folder “02”
7E FF 06 **17** 00 00 0A xx xx EF	Specify loop playback of all tracks in the folder “10”
7E FF 06 **17** 00 00 63 xx xx EF	Specify loop playback of all tracks in the folder “99”

4.6.9. Set random playback (0x18)

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

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **18** 00 00 00 FE E3 EF	Start random playback

4.6.10. Set loop playback for currently playing track (0x19)

During playback of a track, send the “start loop playback” command and it will repeatedly play the current track. If the audio player is at Pause or Stop status, it will not respond to this command. If you need to end the loop playback, just send the “end loop playback” command and It will stop when the playback of the track has finished.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **19** 00 00 00 FE E2 EF	Start loop playback
7E FF 06 **19** 00 00 01 FE E1 EF	End loop playback

4.6.11. Set combination playback/playback of a group (0x21 or 0x22)

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. It supports max. 99 folders, and they need to be renamed to 01, 02, 03,.....99 in the root directory of the storage device.

2). There are two commands 0x21 and 0x22 for this function. When the command 0x21 is used, there will be a return data to MCU after each track has finished playing; when the command 0x22 is used, there will be no return data to MCU after each track has finished playing.

3). For example, 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 audio player 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.

4). During combination playback, it is allowed to Play/Pause and set volume, but not allowed to execute Previous and Next. If you 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.

5). 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.

6). 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.

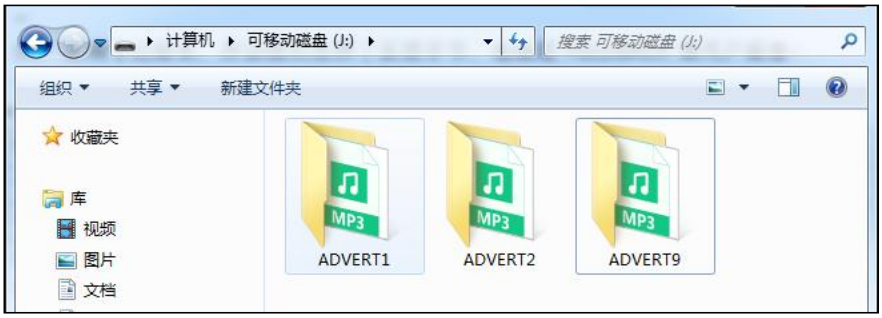
7). With this command, the data will be very long, so we canceled the feedback byte for this function. Please be noted.

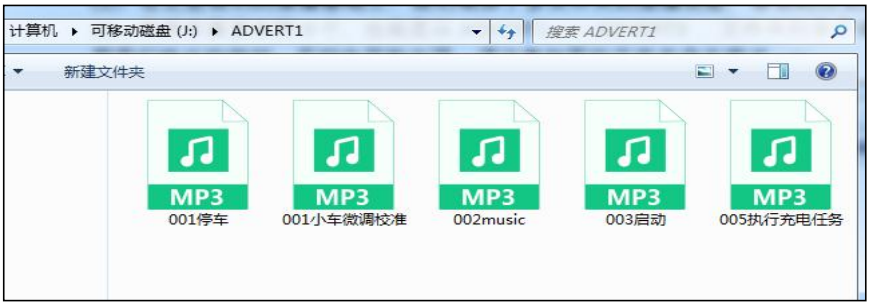
4.6.12. Insert an advertisement (0x25)

1). This audio player supports inserting an advertisement/another sound file 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 IDV3 information of the track that is currently playing and pause, then it will play the inserted track (advertisement). When the inserted track has finished playing, the system will go back and continue to play the track that was interrupted until to the end.

3).The setting method is to create folders and rename them to “ADVERT1”, “ADVERT2”, “ADVERT3”.....“ADVERT9” in the storage device. It supports max. 9 folders. Put the tracks (ads) that need to be inserted into such a folder, and rename the tracks to “001.mp3”, “002.mp3”, “003.mp3”.....“255.mp3”. Refer to the two images below.





4). If you send such a command when the audio player is at Pause status or Stop status, it will not work and there will be an error returned. In the course of an inter-cut, you can continue to insert the other tracks (ads). When the last inserted track has finished playing, the system will still go back to the IDV3 position saved at the first time.

5). Param_MSB represents the folder name and Param_LSB represents the file name. For example, if we send the command data 7E FF 06 **25** 00 02 01 FE D3 EF, the audio player will insert the track “001.mp3” in the folder “ADVERT2” to and play it back.

Command data reference:

Command Data (xx xx represents checksum)	Note
7E FF 06 **25** 00 01 01 xx xx EF	Insert the track “001.mp3” in the folder "ADVERT1"
7E FF 06 **25** 00 02 0A xx xx EF	Insert the track “010.mp3” in the folder "ADVERT2"
7E FF 06 **25** 00 03 0F xx xx EF	Insert the track “015.mp3” in the folder "ADVERT3"

4.6.13. Set a device address (0xC0)

With this function, it is equivalent to giving each audio player a unique name, so multiple audio players can be controlled individually via a RS485 bus. It supports two ways to to setting a device address as below.

1). Set a device address through a serial command (0xC0)

Command data reference:

Command (xx xx represent CRC)	Function
7E FF 06 C0 00 00 01 EF	Set the address to 01
7E FF 06 C0 00 00 63 EF	Set the address to 99

2). Set a device address through a config file

As you can see there are 5 digits in the config file (it's a text document). The 1st digit "0" represents the trigger mode of the buttons (it's purposed for the input triggers), the 2nd and 3rd digits "10" represent the volume level, and the 4th and 5th digits "02" represent a device address.

Notes:

a). When you create a config file like this, you don't need to care the first digit, because it's purposed for the input triggers, but it must exist.

b). The 2nd and 3rd digits, which present a volume level, must exist as well. There are 31 volume levels from 00 to 30. 00 means mute while 30 means the maximum output. So for the 2nd and 3rd digits we suggest you use 30, the maximum output, because volume setting in this config file will always take precedence.

c). The config file must be put in the root directory of the memory.

If the address was set to "01", when you send the serial command from RS485 bus to control the player, "0xFF" will be changed to "0x01", like "7E 01 06 01 00 00 00 EF". If the address was set to "99", when you send the serial command from RS485 bus to control the player, "0xFF" will be changed to "0x63", like "7E 63 06 01 00 00 00 EF".

Please note that the address range set through the RS485 controller is 01-254 (0x01-0xFE in hexadecimal), and the address range set through the config file is 01-99 (0x01-0x63 in hexadecimal).

"0xFF" is a super address. No matter what address is set to through RS485 controller or the config file, this command is always valid.

3). After the address is set by the RS485 controller, it will take effect immediately, and it is with power-off memory.

4). If the address is already set in the config file, the address set by the RS485 controller will not be remembered after

the power is turned off. The address set by the config file will take priority.

4.7. Detailed Annotations for Some of the Query Commands

4.7.1. Query current online storage device (0x3F)

Command data: 7E FF 06 3F 00 00 00 FE BC EF

When the audio player is working, users can use the command as above (0x3F) to query the status of the online storage devices. For example, if the audio player returns the data 7E FF 06 3F 00 00 09 xx xx EF, LSB 0x09 represents USB flash drive online. If LSB is 0x0A, it represents SD card is online. If LSB is 0x0B, it represents both of USB flash drive and SD card is online.

4.7.2. Query current status (0x42)

Command data: 7E FF 06 42 00 00 00 FE B9 EF

1). There are 3 status (playing, paused playing, stopped playing) that can be queried during the system is decoding. Users can query the current status via sending the command as above(0x42).

2). Interpretation of returned data

Returned Data (xx xx represents checksum)	**Corresponding Status**
7E FF 06 **42** 00 01 01 xx xx EF	Audio player is playing a track from USB flash drive
7E FF 06 **42** 00 02 02 xx xx EF	Audio player has paused playing a track from SD card
7E FF 06 **42** 00 01 00 xx xx EF	Audio player has finished playing a track from USB flash drive (now it's at standby)
7E FF 06 **42** 00 02 00 xx xx EF	Audio player has stopped playing a track from SD card

3). MSB and LSB Representations

MSB Representation		**LSB Representation**	
0x01	USB flash drive	0x00	Audio player has finished playing a track (now it's at standby)
0x02	SD card	0x01	Audio player is playing a track
0x04	SPI flash	0x02	Audio player has paused playing a track

4.7.3. Query volume (0x43)

Command data: 7E FF 06 43 00 00 00 FE B8 EF

For example, if the audio player returns the data 7E FF 06 **43** 00 00 1E FE 9A EF, LSB=0x1E represents the current volume level is 30 (max. volume level).

The LSB in the returned data is the current volume value. Converting the hexadecimal number to decimal is the current volume. For example, 0x1E=30, which means the current volume level is 30.

4.7.4. Query total number of tracks in the storage device (0x47, 0x48, 0x49)

The command 0x47 is used to query the total number of tracks in the USB flash drive, the command 0x48 is used to query the total number of tracks in the SD card, and the command 0x49 is used to query the total number of tracks in the SPI flash.

Command data: 7E FF 06 47 00 00 00 FE B4 EF (query the total number of tracks in the USB flash drive)

For example, if the audio player returns the data 7E FF 06 **47** 00 00 08 FE AC EF, MSB=0x00 and LSB=0x08 represent the total number of tracks is 8 in the USB flash drive.

Command data: 7E FF 06 48 00 00 00 FE B3 EF (query the total number of tracks in the SD card)

For example, if the audio player returns the data 7E FF 06 **48** 00 00 08 FE AB EF, MSB=0x00 and LSB=0x08 represent the total number of tracks is 8 in the SD card.

Command data: 7E FF 06 49 00 00 00 FE B2 EF (query the total number of tracks in the SPI flash)

For example, if the audio player returns the data 7E FF 06 **49** 00 00 07 FE AB EF, MSB=0x00 and LSB=0x07 represent the total number of tracks is 7 in the SPI flash.

The MSB and LSB in the returned data indicate the total number of tracks on the current storage device. Converting the hexadecimal number to decimal is the current total number of tracks. For example, MSB=0x00 and LSB=0x0A, converted to decimal it is 10, indicating that the current total number of tracks is 10. Generally, if you don't store many files, you can just look at the LSB value.

4.7.5. Query the currently playing track (0x4B, 0x4C, 0x4D)

The command 0x4B is used to query the currently playing track in the USB flash drive, the command 0x4C is used to query the currently playing track in the SD card, and the command 0x4D is used to query the currently playing track in the SPI flash.

Command data: 7E FF 06 4B 00 00 00 FE B0 EF (query the currently playing track in the USB flash drive)

For example, if the audio player returns the data 7E FF 06 **4B** 00 00 01 FE AF EF, MSB=0x00 and LSB=0x01 represent the currently playing track is the 1st one (physical index order) in the USB flash drive.

Command data: 7E FF 06 4C 00 00 00 FE AF EF (query the currently playing track in the SD card)

For example, if the audio player returns the data 7E FF 06 **4C** 00 00 01 FE AE EF, MSB=0x00 and LSB=0x01 represent the currently playing track is the 1st one (physical index order) in the SD card.

Command data: 7E FF 06 4D 00 00 00 FE AE EF (query the currently playing track in the SPI flash)

For example, if the audio player returns the data 7E FF 06 **4D** 00 00 02 FE AC EF, MSB=0x00 and LSB=0x02 represent the currently playing track is the 2nd one (physical index order) in the SPI flash.

The MSB and LSB in the returned data represent the track currently being played or finished. Converting the hexadecimal number to decimal is the track currently being played. For example, MSB=0x00 and LSB=0x02, converted to decimal is 02, indicating that the current track is the 2nd track in physical index order. Generally, if you don't store many files, you can just look at the LSB value.