Modbus RTU Audio Player with 25W*2 Amplifier
Datasheet
Model: FN-MD02
Version: V1.0

1. Overviews

1.1. Brief Introduction

FN-MD02 is a high quality Modbus RTU audio player launched by Flyron Technology Co., Ltd. This product supports RS485 serial communication and adopts Modbus RTU standard protocol, which can be used for different sorts of industrial control systems or industrial automation equipment. It supports for changing audio files via USB connection, and users can do it easily on computer, which is convenient and fast. It has an solid metal enclosure, with the characteristics of durability, stable and reliable performance.

1.2. Features

1). Adopts RS485 Modbus RTU communication protocol.

2). Possible to control multiple audio players by connection in parallel.

3). Built-in 16MBytes flash memory and supports max. 32GB micro SD card.

4). Equipped with a type-C USB connector. Possible to load audio files to the flash memory/micro SD card directly via USB connection to computer.

5). Update audio files easily via USB connection to computer.

   -It works as a USB flash drive on computer. No need any program.

6). Equipped with a class D 25watt x 2 amplifier that can drive max. two 25watts (4-8 ohms) speakers directly for left and right channels.

7). Adjustable sound volume through the volume knob for the speaker output.

8). Equipped with a 3.5mm audio jack for stereo output that can drive an earphone directly or connect to an external amplifier.

9). Able to drive an external equipment like a warning light simultaneously when it is playing a sound.

10). Able to set different communication baud rates.

11). Wide range for power input.

1.3. Technical Parameters

1). Working voltage: DC 12V-40V (12V or 24V is preferable)

2). Working current: ≥4500mA @12V (when it has two 25W speakers, and external load/warning light is not connected)

   ≥2250mA @24V (when it has two 25W speakers, and external load/warning light is not connected)

3). Output current to an external load/warning light: ≤1000mA

4). Power consumption: ≤50W ((external load/warning light not included))

5). Internal flash memory size: 16MB

6). Audio format: MP3

1.4. Dimensions
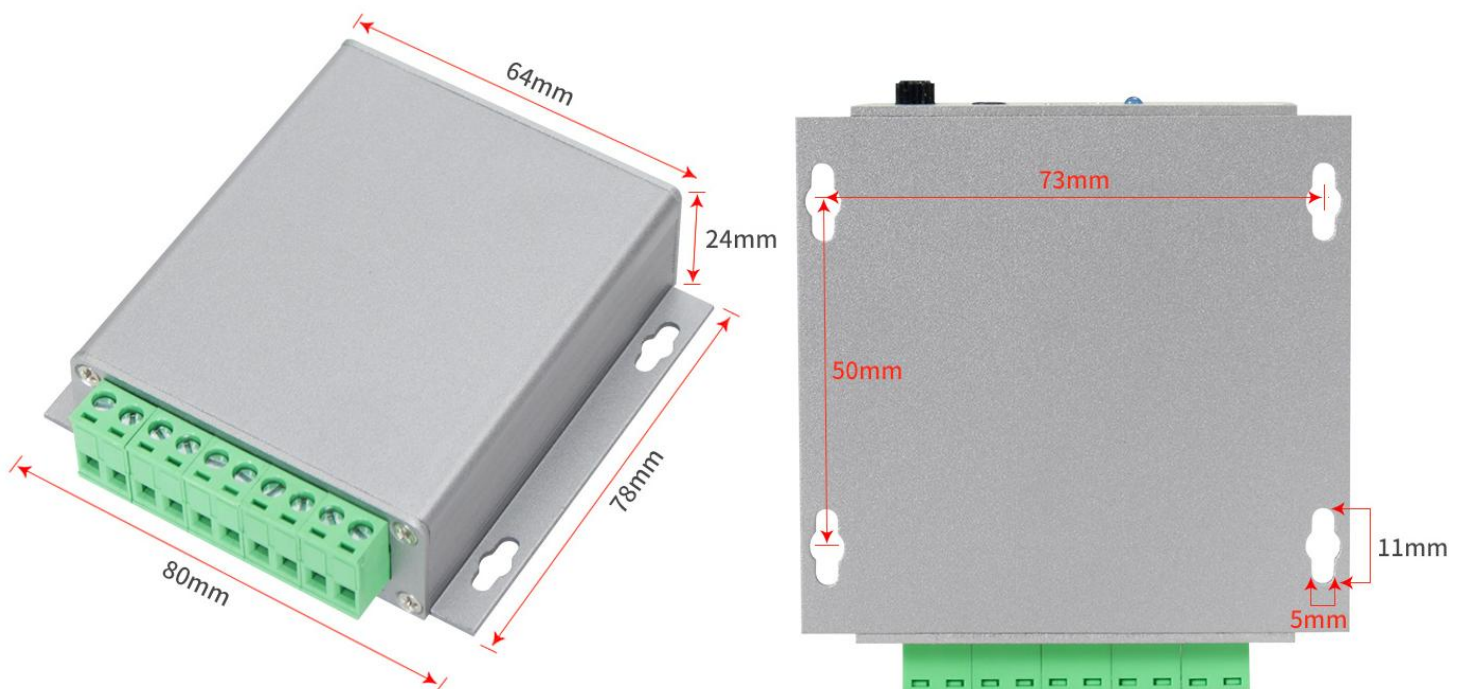
1.5. Basic Connections
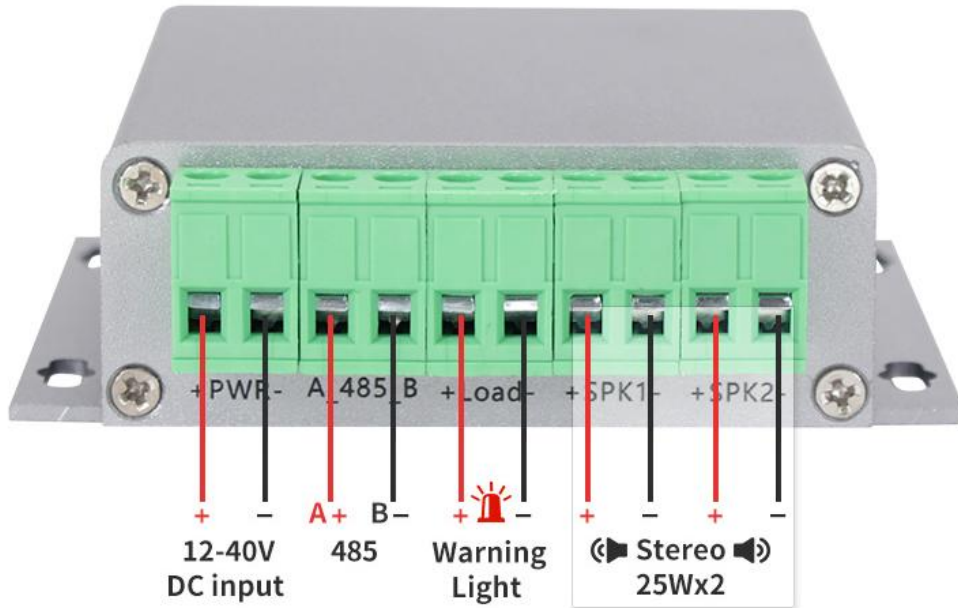
2. Modbus RTU Control Mode

This product supports Modbus RTU serial communication based on RS485, and the default baud rate is 9600. Because of its binary representation and compact data structure, Modbus RTU has high communication efficiency and it is widely used in industrial occasions.

Factory default settings:

Default slave address: 01

Default volume level: 30 (maximum)

RS485 communication settings:

Baud rate: 9600 (default)

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

2.1. Communication Format

Data bits: 8; Stop bits: 1; Error detection: CRC (Cyclic Redundancy Check)							
Addr	Fun	Data start reg hi	Data start reg lo	Data #of regs hi	Data #of regs lo	CRC16_L	CRC16_H
01	06	00	03	00	01	B8	0A
Address code (1 byte), function code (1 byte), start address (2 bytes), data (2 bytes), check code (2 bytes)							

Addr: slave address (device address)

Fun: function code

Data start reg hi: data start address - register high byte

Data start reg lo: data start address - register low byte

Data #of reg hi: number of data read - register high byte

Data #of reg lo: number of data read - register low byte

CRC16_H: Cyclic redundancy check high byte

CRC16_L: Cyclic redundancy check low byte

Format: address code + function code + register address (upper 8 bits) + register address (lower 8 bits) + data (upper 8bits) + data (lower 8bits) + check code (lower 8bits) + check code (upper 8bits)

2.1.1. Address code and function code

Address code:

The address field is at the beginning of the frame consisting of one byte, and the hexadecimal number is 0x00-0xFF. The decimal number is 0-255, of which 255 (0xFF) is our super address. These bits identify the user-specified address of the end device that will receive data from the host connected to it. The address of each terminal device must be unique, and only the addressed terminal will respond to queries containing this address. When the terminal sends back a response, the slave address data in the response tells the host which terminal is communicating with.

Function code:

The function field code tells the addressed terminal what function to perform. The following table lists the function codes that we commonly use, as well as their meanings and functions.

Function Code	Name	Function
03	Read Holding Register	To get the current binary value in one or more holding registers
06	Preset Single Register	To load a specific binary value into a holding register

03H function code: Read the data of the specified register. Our product uses this function to read the system status, the total number of files and other data, that is, the query function of our product.

06H function code: Write data into the register, that is, write the received data sent by the host into the register set by itself. The application of this function in our product is that after our chip receives the command sent by the host, the chip will store this command into the established register, and then perform corresponding actions on this instruction, that is, the control function of our product.

2.1.2. Error check field

This field allows the host and terminals to check for errors during transmission. Sometimes due to electrical noise and other interference, a set of data may change through the wire when it is transmitted from one device to another. Error checking can ensure that the host or terminal does not respond to the data that has changed during transmission. This improves the security and efficiency of the system, and the 16-bit cyclic redundancy method (CRC16) is used for error checking.

In the CRC operation, first a 16-bit register is preset to 0FFFFH (each bit is preset to 1), and then the 8 bits in each byte in the data frame are continuously operated with the current value of the register. Only the 8 data bits of each byte are involved in generating the CRC. The start bits, stop bits and the possible use of parity bits do not affect the CRC. When generating the CRC, the 8 bits of each byte are XORed with the contents of the register, and then the result is shifted to the lower bit. the upper bit is supplemented with "0", and the lowest bit (LSB) is shifted out and detected. If it is 1, the register performs an XOR operation with a preset fixed value (0A001H). If the lowest bit is 0, no processing is performed.

The above processing is repeated until the 8 shift operations are performed. When the last bit (the 8th bit) is shifted, the next 8-bit byte is XORed with the current value of the register. Above another 8 shift XOR operations are also performed in the same way. When all the bytes in the data frame are processed, the final value generated is the CRC value.

The above only explains the calculation method and function of the CRC16 cyclic redundancy check. If you don't understand it, you can ignore this part. Generally, we can hand this check code to the software for calculation, and you don't need to calculate it yourself. It doesn't matter if you don't understand it. , please refer to the relevant information for details.

2.2. Serial Commands

Command	Function	Command Sent (hexadecimal characters)	Note
Control Commands (06H)			
0x01	Next	FF 06 00 **01** 00 00 xx xx	Play next track
0x02	Previous	FF 06 00 **02** 00 00 xx xx	Play previous track
0x03	Specify playback of a track by physical index order	FF 06 00 **03** 00 01 xx xx	Specify playback of the 1st track
		FF 06 00 **03** 00 02 xx xx	Specify playback of the 2nd track
		FF 06 00 **03** 00 63 xx xx	Specify playback of the 99th track
0x04	Volume up	FF 06 00 **04** 00 00 xx xx	Increase volume
0x05	Volume down	FF 06 00 **05** 00 00 xx xx	Decrease volume
0x06	Specify volume	FF 06 00 **06** 00 1E xx xx	Set volume level to 30
0x08	Specify playback of a track in a loop	FF 06 00 **08** 00 01 xx xx	Play the first track in a loop
0x09	Specify a storage device	FF 06 00 **09** 00 02 xx xx	Specify the storage device - micro SD card
		FF 06 00 **09** 00 04 xx xx	Specify the storage device - SPI flash
0x0B	Set baud rate	FF 06 00 **0B** 00 01 xx xx	Set the baud rate to 9600
		FF 06 00 **0B** 00 02 xx xx	Set the baud rate to 19200
		FF 06 00 **0B** 00 03 xx xx	Set the baud rate to 38400
		FF 06 00 **0B** 00 04 xx xx	Set the baud rate to 57600
		FF 06 00 **0B** 00 05 xx xx	Set the baud rate to 115200
		FF 06 00 **0B** 00 06 xx xx	Set the baud rate to 256000
		FF 06 00 **0B** 00 07 xx xx	Set the baud rate to 35250
		FF 06 00 **0B** 00 08 xx xx	Set the baud rate to 2400
		FF 06 00 **0B** 00 09 xx xx	Set the baud rate to 4800
0x0C	Reset	FF 06 00 **0C** 00 00 xx xx	Reset the decoder chip
0x0D	Play	FF 06 00 **0D** 00 00 xx xx	/
0x0E	Pause	FF 06 00 **0E** 00 00 xx xx	/
0x0F	Specify playback of a track in a folder	FF 06 00 **0F** 01 01 xx xx	Specify playback of track 001 in folder 01
		FF 06 00 **0F** 01 02 xx xx	Specify playback of track 002 in folder 01
		FF 06 00 **0F** 02 01 xx xx	Specify playback of track 001 in folder 02
		FF 06 00 **0F** 63 FF xx xx	Specify playback of track 255 in folder 99

0x11	Play all tracks in a loop	FF 06 00 **11** 00 01 xx xx	Turn it on
		FF 06 00 **11** 00 00 xx xx	Turn it off
0x16	Stop	FF 06 00 **16** 00 01 xx xx	/
0x17	Specify loop playback of tracks in a folder	FF 06 00 **17** 00 01 xx xx	Specify loop playback of tracks in folder 01
0x18	Play tracks in random order	FF 06 00 **18** 00 00 xx xx	All tracks in the storage device are covered
0x19	Set currently playing track to loop playback	FF 06 00 **19** 00 00 xx xx	/
	Stop loop playback of current track	FF 06 00 **19** 00 01 xx xx	/
0xC0	Set a device address	FF 06 00 **C0** 00 01 xx xx	Set the device address to 01
		FF 06 00 **C0** 00 63 xx xx	Set the device address to 99
Query Commands (03H)			
0X42	Query the current playback status	FF 03 00 **42** 00 00 xx xx	/
0X43	Query the current volume	FF 03 00 **43** 00 00 xx xx	/
0X49	Query the number of files in the storage device	FF 03 00 **49** 00 00 xx xx	/
0x4D	Query the currently playing track	FF 03 00 **4D** 00 00 xx xx	/
xx xx is the CRC check, and FF is the super address, which can be changed to the corresponding device address.			

3. Detailed Explanation of Serial Commands

Below we describe some commonly used commands in detail.

The command format is

address code + function code + start address high byte+ start address low byte + data high byte + data low byte + CRC check

FF   06   00   CMD   DH   DL   CRC_L CRC_H

FF is the super address, corresponding to the device address, and the factory default address is 01. CMD is the operation code, CRC_L and CRC_H are the cyclic redundancy check. All values are in hex.

Note: When the device receives a command, the LED indicator will flash once. And there will be same data returned to the RS485 host device at the same time.

3.1. Descriptions of Common Control Commands

3.1.1. Specify playback of a track in physical index order (0x03)

This command is used to specify playback of a track in the root directory in physical index order. The selection range of tracks is 1-3000. In fact, more tracks can be supported. Because of file management reasons, supporting too many tracks will cause the system to operate slowly. Generally, most of applications do not need to support so many files.

1). If you choose the 100th track, firstly convert 100 to hexadecimal, the default is double byte, which is 0x0064 (DH=0x00, DL=0x64).

2). Other operations can be deduced and so on.

Command: FF 06 00 03 DH DL CRC_L CRC_H

Example: Send the command "FF 06 00 03 00 01 AD D4" to specify playback of the first track in the root directory (DH = 0x00, DL= 0x01)

Returned data: "FF 06 00 03 00 01 AD D4"

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **03** 00 01 xx xx	Specify playback of the 1st track
FF 06 00 **03** 00 63 xx xx	Specify playback of the 99th track
FF 06 00 **03** 00 FF xx xx	Specify playback of the 255th track
FF 06 00 **03** 0B B8 xx xx	Specify playback of the 3000th track

3.1.2. Specify volume (0x06)

1). There are 31 volume levels (0-30). If you want to set the volume, just send the corresponding command directly.

2). Sending a command to set the volume will be reset when the device restarts. If you need the power-off memory, you can set the volume through a configuration file.

Command: FF 06 00 06 00 DL CRC_L CRC_H (DL is the parameter of the volume to be set).

Example: Send the command "FF 06 00 06 00 1E FC 1D" to set the volume level to 30. 30 is 0x1E in hex, so DL=0x1E.

Returned data: "FF 06 00 06 00 1E FC 1D"

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **06** 00 00 xx xx	Set the volume level to 0 (mute)
FF 06 00 **06** 00 19 xx xx	Set the volume level to 25

3.1.3. Set baud rate (0x0B)

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

2). After setting the baud rate, please wait for 1 second, and then send the reset command 0x0C, or power off and restart to take effect.

3). After the baud rate is set, the device will memorize it, and the baud rate will be changed when it is powered on next time.

Command: FF 06 00 0B 00 DL CRC_L CRC_H (DL is the parameter of the baud rate to be set. See the following table for details)

Example: Send the command "FF 06 00 0B 00 02 6C 17" to set the baud rate to 19200 (DL=0x02)

Returned data: "FF 06 00 0B 00 02 6C 17"

Parameter for DL	Corresponding Baud Rate	Parameter for DL	Corresponding Baud Rate
0x01	9600	0x06	256000
0x02	19200	0x07	35250

0x03	38400	0x08	2400
0x04	57600	0x09	4800
0x05	115200		

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **0B** 00 01 xx xx	Set the baud rate to 9600
FF 06 00 **0B** 00 05 xx xx	Set the baud rate to 115200

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

Command: FF 06 00 0F DH DL CRC_L CRC_H

DH represents the name of the folder, and supports up to 99 folders that is 01-99 (DH=0x01-0x63)

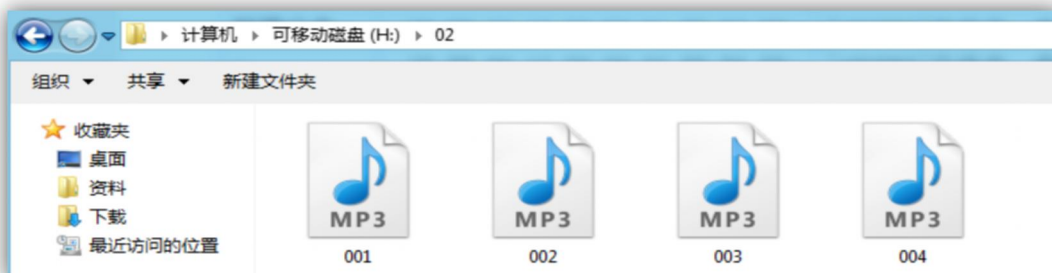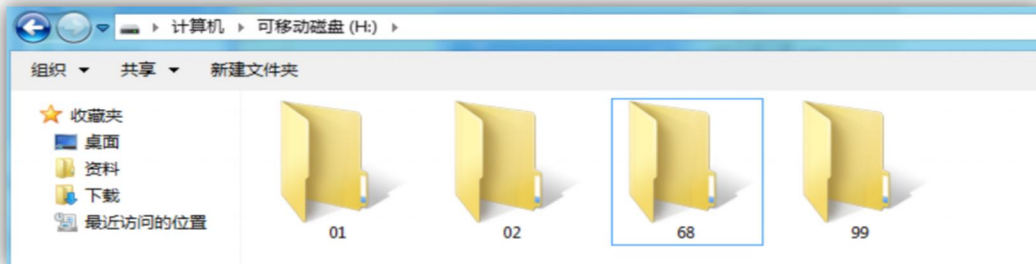DL represents the track, and supports up to 255 tracks that is 001-255 (DL=0x01-0xFF)

Example: Send the command "FF 06 00 0F 01 01 6C 47" to play track "001xxx.mp3" in folder "01" (DH=0x01, DL=0x01)

Returned data: "FF 06 00 0F 01 01 6C 47"

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **0F** 01 01 xx xx	Specify playback of track 001.mp3 in folder 01
FF 06 00 **0F** 01 63 xx xx	Specify playback of track 099.mp3 in folder 01
FF 06 00 **0F** 63 FF xx xx	Specify playback of track 255.mp3 in folder 99

The following two screenshots illustrate the designation of folder names and file names. Please name the folders and files in strict accordance with our instructions.

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

Command: FF 06 00 17 00 DL CRC_L CRC_H (DL represents the name of the folder and supports up to 99 folders that is 01-99, DL=0x01-0x63)

Example: send the command "FF 06 00 17 00 01 ED D0" to play tracks in a loop in the folder 01 (DL=0x01)

Returned data: "FF 06 00 17 00 01 ED D0"

Notes:

1). The naming method of the folder must be "01" to "99", and cannot exceed 99.

2). Once the loop playback starts, you can use play/pause/previous/next. None of these operation commands will interrupt the current loop playback state. That is to say, after sending the "next" command, the current folder will still be in loop playback mode.

3). Users can send a stop command to end the loop playback and return to the trigger playback state.

4). Each time a track is played, there will be a frame of returned data, corresponding to the track that has just been played.

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **17** 00 01 xx xx	Specify loop playback of tracks in folder 01
FF 06 00 **17** 00 63 xx xx	Specify loop playback of tracks in folder 99

3.1.6. Set currently playing track to loop playback (0x19)

Command: FF 06 00 19 00 DL CRC_L CRC_H

DL=0x00 (start loop playback), DL=0x01 (end loop playback)

Example:

Send the command "FF 06 00 19 00 00 4D D3" to start loop playback (DL=0x00)

Send the command "FF 06 00 19 00 01 8C 13" to end loop playback (DL=0x01)

1). Sending this command during playback will loop the current track. If it is currently in a paused or stopped state, the device will not respond to this command.

2). If you want to stop the loop playback, you can send the end command, which will stop the current track after playing.

3.1.7. Set device address (0xC0)

The device supports sending the serial command and using a configuration file to set the address.

1). Set device address by sending the serial command

Command: FF 06 00 C0 00 DL CRC_L CRC_H

DL represents the address to be modified, which can be set to 0-255 (to be converted to hexadecimal number) and

DL=0x00-0xFF

Example: Send the command "FF 06 00 C0 00 FE 1D A8" to set the address to 254 (DL=0xFE)

Returned data: "FF 06 00 17 00 01 ED D0"

Reference Commands:

Command (xx xx represent CRC)	Function
FF 06 00 **C0** 00 01 xx xx	Set the device address to 01
FF 06 00 **C0** 00 63 xx xx	Set the device address to 99
FF 06 00 **C0** 00 F7 xx xx	Set the device address to 247

2). Set device address by a configuration file (.txt)

1). As you can see there are 5 digits in the configuration file (it's a text document). The first digit "0" represents the trigger mode of the button (it's purposed for other products), the second and third digits "10" represent the volume setting, and the fourth and fifth digits "02" are the device address.

Notes:
a). When you create a configuration file like this, you don't need to care the first digit, because it's purposed for other products but it must exist.
b). The configuration file must be placed on the root directory of the memory.

2). There are two types of addresses here. One type is is like this to be set by configuration file, and it can be set to 99. This represents decimal, and the device return address is 0x63. Please note that the address range set by the serial command is 0-255, and the address range set by the configuration file is 0-99. The other one is the super address "0xFF". That is, you send a command through the serial port, and the address is set to "FF", no matter what address is set to in the configuration file, this command is valid.

3). After the device address is set by the serial command, the address will take effect immediately with power-off memory.

4). The addresses set by the two methods can be remembered after power off, but the address set by the configuration file takes precedence after restarting.

5). The function of setting the address is that multiple devices can be connected to the RS485 bus, which is equivalent to giving each device a unique name, so that each device can be controlled individually. For details, you can search for the principle of RS485.

3.2. Descriptions of Query Commands

3.2.1. Query the current playback status (0x42)

Send the command "FF 03 00 42 00 00 F0 00" to query the current playback status of the device (CMD=0x42)

Returned data: "01 03 02 04 01 7B 44" (DH=0x04, DL=0x01, indicating that SPI flash is playing)

"01 03 02 04 02 3B 45" (DH=0x04, DL=0x02, indicating that the SPI flash playback is paused)

"01 03 02 04 00 BA 84" (DH=0x04, DL=0x00, indicating that SPI flash stops playing)

"01 03 02 02 00 B9 24" (DH=0x02, DL=0x00, it means the SD card stops playing)

DH Value from Returned Data	Corresponding Storage Device	DL Value from Returned Data	Corresponding Status
0x02	SD card	0x00	Playback is stopped
0x04	SPI Flash	0x01	Playing
		0x02	Playback is paused

3.2.2. Query volume (0x43)

Send the command "FF 03 00 43 00 00 A1 C0" to query current volume (CMD=0x43)
Returned data: "01 03 02 A1 1E 41 DC" (the current volume level is 30, DL=0x1E)

The DL value in the returned data represents the volume value. Converting its hexadecimal number to a decimal number is the current volume. For example, 0x1E converted to decimal is 30, which means the current volume level is 30. The address bit 0x01 indicates that the current device address is 01.

3.2.3. Query the number of files in the storage device (0x48, 0x49)

Send the command "FF 03 00 48 00 06 50" to query the total number of files on the SD card.
Returned data: "01 03 02 00 08 B9 82" DH=0x00, DL=0x08, the total number of files on the SD card is 8.

Send the command "FF 03 00 49 00 00 81 C2" to query the total number of files on the SPI flash memory.
Returned data: "01 03 02 00 08 B9 82" The total number of files on the SPI flash memory is 8.

The DH and DL bits in the returned data represent the total number of files on the device. For example, converting DH=0x00, DL=0x08 to decimal is 08, which means that the total number of files is 8. Generally, we do not save so many files, so just look at the DL value.

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

Send the command "FF 03 00 4C 00 00 91 C3" to query the current track on the SD card
Returned data: "01 03 02 00 01 79 84" (the first track in physical order is currently being played or finished)

Send the command "FF 03 00 4D 00 00 C0 03" to query the current track on the SPI flash memory
Return data: "01 03 02 00 02 39 85" (the second track in physical order is currently being played or finished)

The DH and DL bits in the returned data represent the track currently being played or finished. For example, converting DH=0x00, DL=0x02 to decimal is 02, which indicates that the current track is the second one in the physical order. Generally, we don't save so many files, so just look at the DL value.

4. How to Update Audio Files

Use an type-C USB data cable to connect it to computer (Windows preferable) to update audio files. The connected device will be detected as a USB flash drive on computer. Delete the sample audio files preloaded at factory for testing purpose and copy your own ones onto the memory. Please refer to the images as below.


A USB drive is detected

Folder name: 01…99; File name: 001xx.mp3, 002xx.mp3, 003xx.mp3…
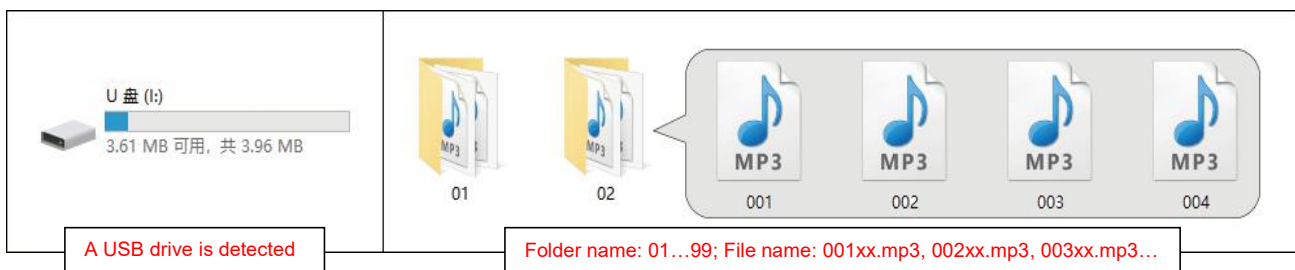
Notes:

1). When there are multiple audio files on the root directory of the memory, they will be played according to the physical index order, which is the file that is to be loaded first in the memory will be played first, instead of by file name. When replacing audio files, please copy them to the memory one by one according to the playback sequence you need. If you need to read files by file name, you can create a folder named 01, 02...99, and the audio files in the folder are named 001.mp3...255.mp3. Use the command 0x0F to specify playback of a track in a specific folder. For example: send "7E FF 06 0F 00 01 01 EF" to specify playback of the file 001.mp3 in the folder 01. For details, please refer to the description of "3.1.1. Specify playback a track in a folder".

2). Before replacing your the audio files, it is best to reformat the memory, and then copy the files onto it to avoid playback errors due to the existence of hidden files.

3). Audio files must be in MP3 format, and must be in the sampling rate supported by the product. Supported sampling rate (KHz): 8/11.025/12/16/22.05/24/32/44.1/48.

4). Before connecting the device to computer, please disconnect the power to the device. After replacing the audio files, you need to disconnect the the device from the computer and then power on the device.

5. Wiring Example for Multiple Devices

If you need to control multiple slave devices, you can refer to the parallel wiring example as below.
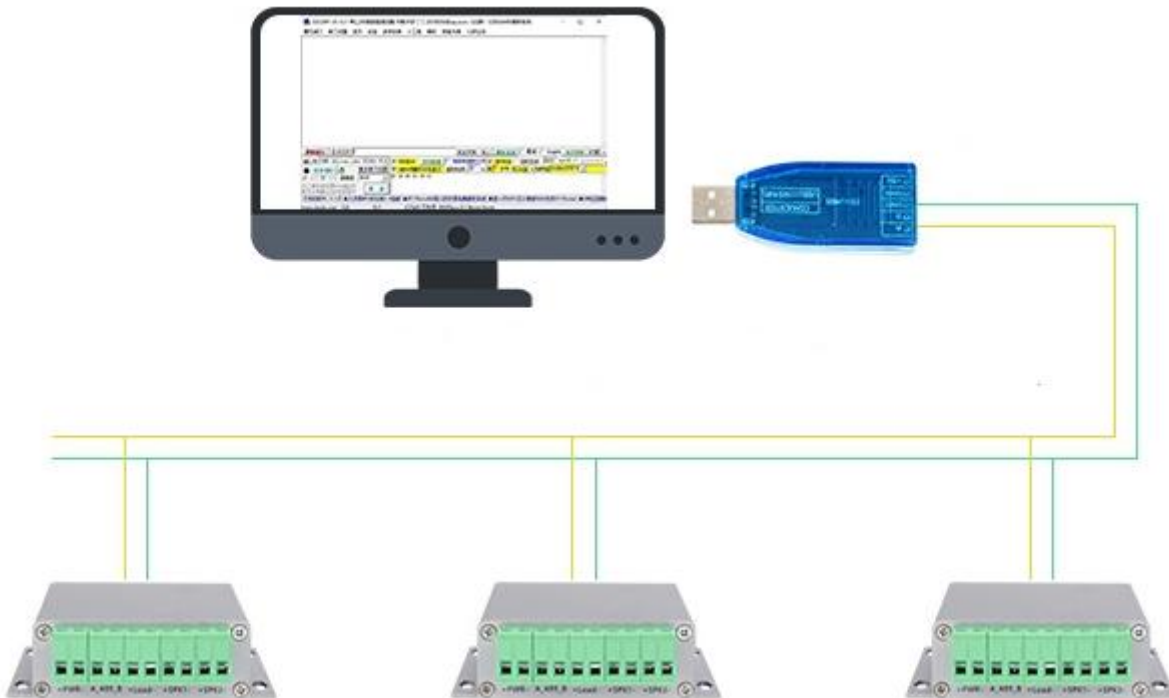
This wiring example shows that the devices are connected to a computer via a USB to RS485 converter. The yellow wire is for connecting the RS485 communication port A (+) from computer to the devices, while the green wire is for connecting the RS485 communication port B (-) from computer to the devices. Running a serial debugging software on computer, users can debug and test the device easily before applying it.