

## CHARACTERISTICS

- support MP3, and WAV decoding
- support FAT file system
- Sampling rate (KHz) : 8/11.025/12/16/22.05/24/32/44.1/48
- Hardware decoding using DSP to 24-bit DAC; dynamic range: 90dB, signal-to-noise ratio: 85dB
- Control methods: single-line control, standard UART
- Support SPIFLASH as USB drive; direct control as USB drive
- Support operation: last audio, next audio, play, pause, stop, named audio playback, ...
- Support access to files and directories information
- Support access to status and various information
- Support replaying specified audio, pausing current audio, inserting another audio and returning to the point of pause for continuation
- Support both Chinese and English path descriptions for named audio playback
- Support intelligent assembly of audio playback from different directories
- Support 30 levels of volume control
- Support 5 types of equalizer control
- Support 7 modes of control
- Support multiple times of replay
- Support sleeping mode with power consumption as low as 600  $\mu$  A, and awakening via one-line control command
- Support speed forward and backward at specific time
- Support replay at specific time
- Support specific starting time
- Support BUSY signal output
- Enable stable application via comprehensive command set
- Tailor-made application for command generation

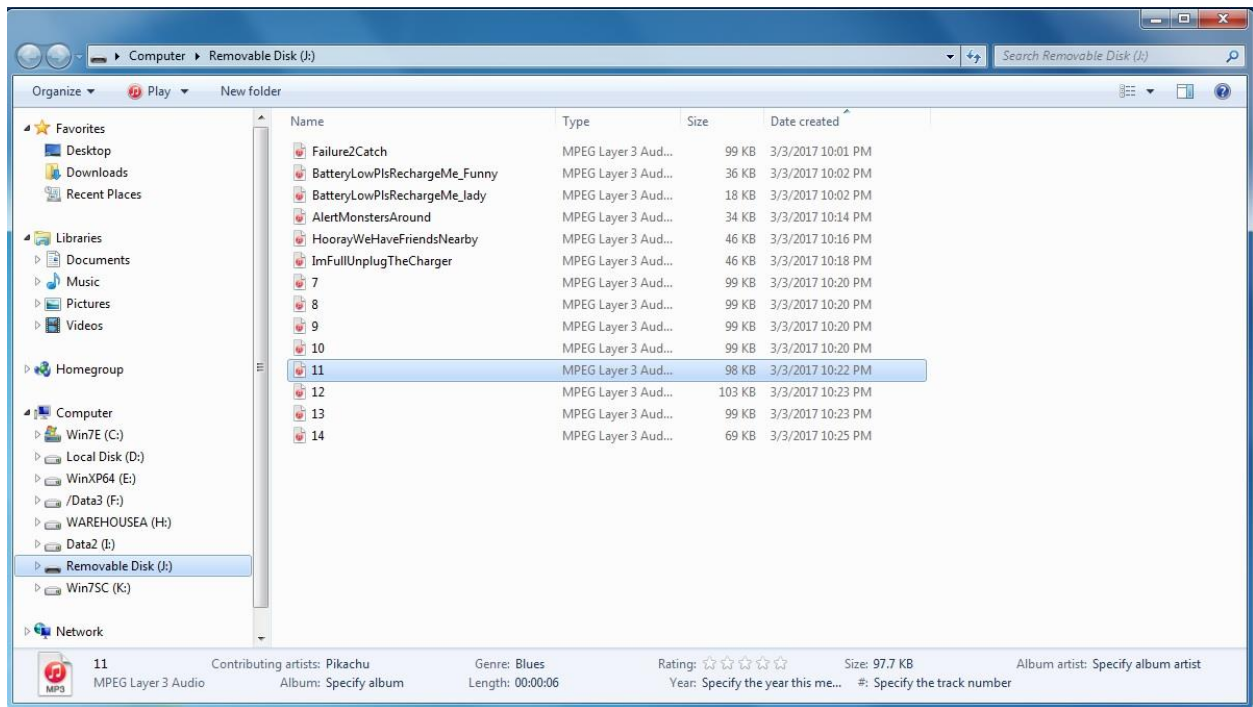
## GENERAL DESCRIPTIONS

JQ8400 module is a SOC solution that combines a 16bit MCU with an ADSP specialised in audio decoding. By using hardware decoding method, it ensures system stability, quality of sound, and small footprint matching requirements of embedded applications.

Changing contents in SPI-flash

Saving the hassle of dedicated programming MCU, SPI-flash enables direct plug-in with a computer via USB port for direct file operations as an USB drive. It largely simplified R&D and production. For the flexibility of R&D, control commands can be sent via one-line mode or RS232 serial communication.

By connecting the built-in USB port to a computer, the module will be recognized as an USB drive as shown below in Windows environment.



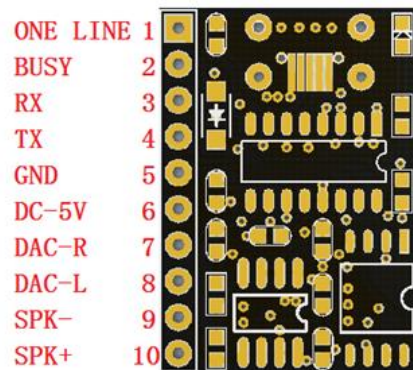
In order to use composite audio playback, a new directory named 'ZH' has to be created, and all audio files for composite playback have to be stored in it.

## HARDWARE PARAMETERS

Name	Parameters
MP3 file format	<ol style="list-style-type: none"> <li>1. Supports all 11172-3 bit rates and ISO13813-3 layer 3 audio decoding</li> <li>2. Sampling rates (KHz): 8/ 11.025/ 12/ 16/ 22.05/ 24/ 32/ 44.1/ 48</li> <li>3. Supports Normal, Jazz, Classic, Pop, Rock sound effects</li> </ol>
USB port	USB 2.0 standard
UART port	One-line interface, standard UART interface, TTL
Power input	DC 3.3 – 5.5V; best at 4.2V, IO port level at 3.3V
Default current	Sleep mode: 500uA; normal mode: 10mA
PCB size	18 x 25mm
Working temperature	-40 to 85 degree Celcius
Humidity	5% to 95%

## PINOUTS

### 五、模块管脚说明



Pins	Labels	Descriptions
1	ONE LINE	One-line interface port
2	BUSY	HIGH when playing, LOW in other conditions
3	RX	Data receive port; to be connected with TX of MCU
4	TX	Data transmit port; to be connected with RX of MCU
5	GND	Ground
6	DC-5V	Power supply; 3.3V – 5.0V
7	DAC-R	DAC right channel audio out
8	DAC-L	DAC left channel audio out
9	SPK -	To Speaker
10	SPK +	To Speaker

### One-Line Communication Protocols

[SKIPPED: due to the need of synchronization requirement, precise adjustment via the use of oscilloscope is needed. It is not tested at all.]

### UART Communication Protocols

It adopts full duplex serial communication protocol:

Baud rate 9600, 8-bit data, no parity, 1 stop bit.

### Message Format

Starting Code + Command Code + Data Length + Data 1 – Data n + Checksum

Starting Code : fixed as "AA" (in hex)

Command Code : 01 – 26 (in hex)

Data Length : respective data bytes in commands; length=1 stands for command only, no data.

Checksum : The low byte of the sum of all the bytes before the checksum byte.

Data format: high 8-bit first, low 8-bit second.

### Communication Mechanism

1. Default power-on state is idle and listening for commands from MCU as a guest
2. The module will not initiate communication and a MCU must take initiative.
3. The TTL level of the serial port is 3.3V. Use a 1K ohm resistor if connecting with a 5V MCU.
4. Unless otherwise specified, all data in this protocol is in hexadecimal.

### Protocol Parameters

Return values of the module:-

Status of playback: default idle when power-on

00: stop, 01: playing, 02: pause

Storage definition: default state after switching storage is idle

USB: 00, SD: 01, FLASH:02, NO DEVICE: FF

Volume: There are totally 31 level from 0 to 30, default level when power-on is level 20.

Playing mode definition: default state when power-on is stopping at single audio playback

Loopback All (00): play all audio in sequence, loops back to the first audio at the end

Single audio loop (01): continuously repeat playback of the current audio

Single audio stop (02): playback the current audio once and stop

Play all at random (03): randomly play all audio

Directory loop (04): Sequentially play all audios in the current directory, not including any sub-directory

Random play in directory (05): randomly play all audio in the current directory

Sequential play in directory (06): Sequentially play all audios in the current directory, excluding all sub-directory

Sequential play (07): Sequentially playback all audio on the storage, then stop

Definition of EQ: NORMAL (00), POP (01), ROCK (02), JAZZ (03), CLASSIC (04)

Default EQ when power-on: MP3 channel (00)

MP3 channel (00): Playing via MP3 channel, DAC output MP3 music

AUX channel (01): DAC output audio from P26 and P27

MP3+AUX (02): AUX and MP3 channels open at the same time, DAC output combines MP3 audio with input of P26 and P27

Definition of combined playback: combines according to file names stored in the "ZH" directory. It is advisable to use two digits as file names; for instance, 01.mp3, 02.mp3, two alphabets or one Chinese character may be used though.

## **COMMANDS**

### **Status Inquiry (01)**

Command: AA 01 00 AB

Return: AA 01 01 status SM

Comments: This command can be used any time.

### **Play (02)**

Command: AA 02 00 AC

Return: none

Comments: Upon receipt of this command, the module will play at the beginning of the current audio.

### **Pause (03)**

Command: AA 03 00 AD

Return: none

### **Stop (04)**

Command: AA 04 00 AE

Return: none

### **Last song (05)**

Command: AA 05 00 AF

Return: none

### Next song (06)

Command: AA 06 00 B0

Return: none

### Specify song (07)

Command: AA 07 02 filename(Hi) filename(Lw) SM

Return: none

Example: AA 07 02 00 08 BB play the 8<sup>th</sup> audio of the current drive, available number of audio from 1 to 65535

### Specify drive and directory (08)

Command: AA 08 length drive directory SM

Comments: length = length of drive + length of directory = 1 + length of directory

Example: /AA/BB\*???

Command : 0xAA, 0x08, 0x0B , 0x01, 0x2F, 0x41, 0x41, 0x2F, 0x42, 0x42, 0x2A, 0x3F, 0x3F, 0x3F, 0x09

*[This example is modified from an example of Chinese directory and file name. Please testify it yourself. / = 0x2F, \* = 0x2A, ? = 0x3F, A = 0x41, BB = 0x42, m = 0x6d, p = 0x70, 3 = 0x33]*

### Inquiry of current on-line storage (drive number) (09)

Command: AA 09 00 B3

Return: AA 09 01 drive SM

Comments: Drive code is differentiated by the position of bit: USB: bit(0) SD: bit(1) Flash: bit(2)

This command informs the current drive code. It is advisable to inquire the current drive code before switching drive.

### **Inquiry of the current playing storage (drive number) (0A)**

Command: AA 0A 00 B4

Return: AA 0A 01 drive code SM

### **Switch to specified drive (0B)**

Command: AA 0B 01 drive code SM

Return: none

Comments: If the drive is on-line, this command can switch to the specified drive and wait for playing of the 1<sup>st</sup> audio there.

Examples:

AA 0B 01 00 B6 : switch to a USB drive

AA 0B 01 01 B7 : switch to a TF card drive

AA 0B 01 02 B8 : switch to a flash card

### **Inquiry of total number of audio files (0C)**

Command: AA 0C 00 B6

Return: AA 0C 02 Number(hi) Number(Lw) SM

### **Inquiry of current file number (0D)**

Command: AA 0D 00 B7

Return: AA 0D 02 Filenumber(Hi) Filenumber(Lw) SM

### **Last Directory (0E)**

Command: AA 0E 00 B8

Return: none

Comments: this command will play the 1<sup>st</sup> audio of the directory.

*[I am not sure if this command means to go up one level in the directory tree. Please test it yourself.]*

### **Next directory (0F)**

Command: AA 0F 00 B9

Return: none

Comments: this command will play the 1<sup>st</sup> audio of the directory.

*[I am not sure if this command means to go down one level in the directory tree. Please test it yourself.]*

### **Stop playing (10)**

Command: AA 10 00 BA

Comments: This command terminates the current operation. If the current operation is insertion play then it will return to the previous status.

### **Inquiry of the first audio (11)**

Command: AA 11 00 BB

Return: AA 11 02 Filenumber(Hi) Filenumber(Lw) SM

Comments: This command returns the ordinal number of the first audio in the current directory.

*[I am not sure why we need this command??]*

### **Inquiry of the total number of audio in the directory (12)**

Command: AA 12 00 BB

Return: AA 12 02 Filenumber(Hi) Filenumber(Lw) SM

Comments: The return number does not include the number of files in the sub-directory

### **Volume control (13)**

Command: AA 13 01 VOL SM

Return: none

Example: AA 13 01 14 D2 : set the volume to level 20



### **Volume increase by one level (14)**

Command: AA 14 00 BE

Return : none

### **Volume decrease by one level (15)**

Command: AA 15 00 BF

Return: none

### **Insertion play (16)**

Command: AA 16 03 drive Filenumber(Hi) Filenumber(Lw) SM

Return: none

Example: AA 16 03 00 00 09 CC : play the 9<sup>th</sup> audio in the USB drive

Comments: After the specified playback finishes, it will return to the previous break point and play

### **Specify playing directory (17)**

Command: AA 17 length drive directory SM

Return: none

Comments: length = drive length + directory length = 1+ directory length

### **Stop insertion play**

Command: use Stop playing (10)

### **Set up loop back mode (18)**

Command: AA 18 01 loop mode SM

Return: none

Example: AA 18 01 03 C6 : set to single audio playback and stop

### **Set up number of repeated loops (19)**

Command: AA 19 02 number(Hi) number(Lw) SM

Return: none

Comments: This command only works when the module is set for Loop back all, Single Audio loop and Directory loop.

Example: AA 19 02 00 06 CB : repeat loop back 6 times.

### **EQ setting (1A)**

Command: AA 1A 01 EQ SM

Return: none

Example: AA 1A 01 02 C7 : set to ROCK mode

### **Combined playback (1B)**

Command: AA 1B length Audio1(Hi) Audio1(Lw) ... AudioN(Hi) AudioN(Lw) SM

Return: none

Example: AA 1B 04 30 31 30 32 8C : combined playback of both file "01" and "02"

Comments: Combined playback using file names is more convenient and accurate, and free from the constraint of copying sequence.

### **Stop combined playback (1C)**

Command: AA 1C 00 C6

Return: none

Comments: Stop combined playback and return to the previous state.

### **Set up channel (1D)**

Command: AA 1D 01 channel SM

Return: none

### **Inquiry of the short file name (1E)**

Command: AA 1E 00 C8

Return: AA 1E length filename SM

### **Select audio but not playback (1F)**

Command: AA 1F 02 Filenumber(Hi) Filenumber(Lw) SM

Return : none

### **Repeat reading control (20)**

Command: AA 20 04 Start\_minute start\_second stop\_minute stop\_second SM

Return: none

### **Stop repeat reading (21)**

Command: AA 21 00 CB

Return: none

### **Specify duration for fast backward (22)**

Command: AA 22 02 duration(Hi) duration(Lw) SM

Return: none

Comments: duration in seconds

### **Specify duration for fast forward (23)**

Command: AA 23 02 duration(Hi) duration(Lw) SM

Return: none

Command: duration in seconds

### **Inquiry of duration of the current audio (24)**

Command: AA 24 00 CE

Return: AA 24 03 hour minute second SM

### **Turn on playback duration (25)**

Command: AA 25 00 CF

Return: AA 25 03 hour minute second SM

Comments: The module will return when playback finishes at specified time.

### **Stop playback duration (26)**

Command: AA 26 00 D0

Return: none

Comments: Turn off playback duration