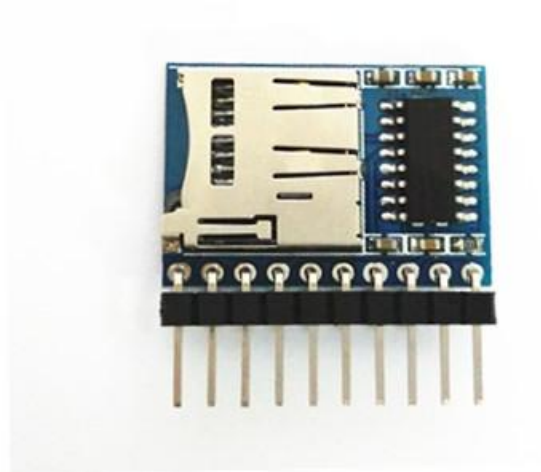


FN-M10P Embedded MP3 Audio Module

Datasheet

V1.0



Contents

| | |
|---|-----------|
| 1. Overviews..... | 2 |
| 1.1. Brief Introduction..... | 2 |
| 1.2. Product Features..... | 2 |
| 1.3. Technical Parameters..... | 2 |
| 1.4. Naming Rules of Audio Files(Tracks) and Folders..... | 2 |
| 2. Pin Configuration and Summary..... | 3 |
| 3. Serial Communication Protocol..... | 4 |
| 3.1. Serial Commands Format..... | 4 |
| 3.2. About Checksum..... | 4 |
| 3.3. Serial Commands..... | 4 |
| 3.4. Examples of Sending Serial Commands..... | 6 |
| 3.5. Returned Data from Module..... | 8 |
| 3.6. Detailed Annotation of Serial Commands..... | 9 |
| 3.7. Detailed Annotation of Main Query Commands..... | 14 |
| 3.8. Examples of Serial Program..... | 15 |
| 4. ADkey Control..... | 17 |
| 5. Connection of Circuit References..... | 18 |
| 6. Features of GPIO..... | 19 |
| 7. PCB Size Drawing..... | 19 |

1.Overviews

1.1. Brief Introduction

FN-M10P is a simplified serial MP3 module with a perfect integrated MP3 and WMV decoder chip. It provides micro SD card driver, and supports FAT16 and FAT32 file systems. It is able to play back specified sound files and realize other functions through simple serial commands. In the mean time, this module supports AD key control mode that facilitates users to develop their jobs in some simple applications. Without the cumbersome underlying operating, easy to use, stable and reliable are the most important features of this module.

1.2. Product Features

- Supports MP3 and WAV decoding.
- Supports FAT16 and FAT32 file system.
- Supports AD key control mode and UART RS232 serial control mode.
- Supports maximum 32GB micro SD card and 32GB USB flash drive.
- Supports maximum 3000 audio files in the root directory of the storage device.
- Supports maximum 99 folders, and each folder can stores 255 audio files.
- Supports inter-cut advertisements.
- Supports random playback and repeat/loop playback.
- 30 levels adjustable volume, and 6 levels adjustable EQ.
- PCB size: 25.80mmx18.00mm

1.3. Technical Parameters

| Item | Description |
|-----------------------|---|
| MP3 Audio Format | Supports 11172-3 and ISO13813-3 layer3 audio decoding |
| | Supports sampling rate (KHZ):8/11.025/12/16/22.05/24/32/44.1/48 |
| | Supports Normal, Jazz, Classic, Pop, Rock, etc. |
| USB Port | Standard USB 2.0 |
| UART Port | Standard serial port and TTL level; |
| Working Voltage | DC3.3~5.0V; Typical: DC4.2V |
| Rated Current | <15mA(without USB flash drive) |
| Operating Temperature | -40~+80℃ |
| Humidity | 5% ~95% |

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

1). Audio files directly stored in the root directory of the storage device(SD card or USB flash drive) need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav,3000.mp3/3000.wav.

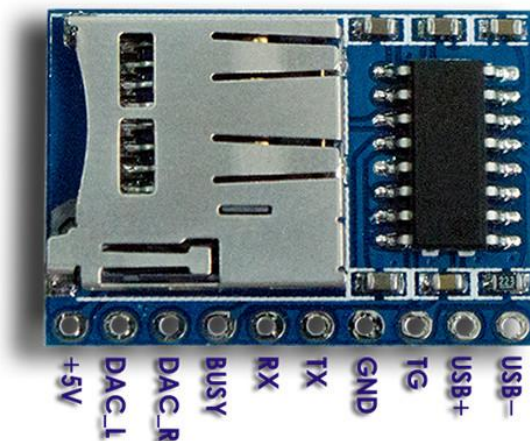
Here it works according to physical sequence when you copy the files from computer to SD card or USB flash drive. For example, when the module receives a command to play the track 0001.mp3, it will play the 1st track you copied from computer, probably 0001.mp3 or not (maybe it would play 0007.mp3 if it was the first one you copied from computer). In order to avoid this problem, when you make the copy, rename the audio files firstly on computer and

put all the renamed files in a 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). 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.

2. Pin Configuration and Summary



| No | Pin | Description | Note |
|----|-----------------|----------------------------|---|
| 1 | VDD power input | DC 5V | No more than 5.2V |
| 2 | DAC_L | Audio output left channel | Drive an earphone or connect to an external amplifier |
| 3 | DAC_R | Audio output right channel | |
| 4 | Busy | Busy output | Low level when working, and high level when standby |
| 5 | RX | UART serial input | 3.3V TTL level |
| 6 | TX | UART serial output | 3.3V TTL level |
| 7 | GND | Ground | Power/audio GND |
| 8 | ADKEY | Triggering output port | |
| 9 | USB+ | USB+ DP | USB Port(connected with a USB flash drive or connected to PC) |
| 10 | USB- | USB- DM | |

3. Serial Communication Protocol

Serial port control mode is a common communication in the control field, based on which we conducted an industrial level of optimization by adding frame checksum, retransmission, error handling, and other measures to significantly strengthen the stability and reliability of communication. The default baud rate is 9600.

3.1. Serial Commands Format

Supports asynchronous serial communication mode, via which accept serial commands sent by upper PC.

Communication Standard: 9600 bps

Data bits: 1

Checkout: none

Flow Control: none

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

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.

3.2. 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.

3.3. Serial Commands

3.3.1. Control Commands

| Command | Function Description | Note |
|---------|-----------------------------|-----------------------|
| 0x01 | Play Next | |
| 0x02 | Play Previous | |
| 0x03 | Specify playback of a track | See 3.6.1 for details |
| 0x04 | Increase volume | |

| | | |
|------|--|---|
| 0x05 | Decrease volume | |
| 0x06 | Specify volume | See 3.6.2 for details |
| 0x07 | Specify EQ(0/1/2/3/4/5) | 0:Normal/1:Pop/2:Rock/3:Jazz/4:Classic/5: Bass |
| 0x08 | Specify single repeat playback | See 3.6.3 for details |
| 0x09 | Specify playback of a device | See 3.6.4 for details |
| 0x0A | Set Sleep | |
| 0x0B | N/A(Reserved) | |
| 0x0C | Reset | |
| 0x0D | Play | |
| 0x0E | Pause | |
| 0x0F | Specify playback a track in a folder | See 3.6.5 for details |
| 0x10 | Audio amplification setting | MSB=1:amplifying on, LSB:set gain 0-31 |
| 0x11 | Set all repeat playback | See 3.6.6 for details |
| 0x12 | Specify playback of folder named "MP3" | See 3.6.7 for details |
| 0x13 | Inter cut an advertisement | See 3.6.8 for details |
| 0x14 | Specify playback a track in a folder that supports 3000 tracks | See 3.6.9 for details |
| 0x15 | Stop playing inter-cut advertisement and go back to play the music interrupted | See 3.6.10 for details |
| 0x16 | Stop | |
| 0x17 | Specify repeat playback of a folder | See 3.6.11 for details |
| 0x18 | Set random playback | See 3.6.12 for details |
| 0x19 | Set repeat playback of current track | See 3.6.13 for details |
| 0x1A | Set DAC | See 3.6.14 for details |

3.3.2.Query Commands

| Command | Function Description | Note |
|---------|---|-----------------------|
| 0x3C | N/A(Reserved) | |
| 0x3D | N/A(Reserved) | |
| 0x3E | N/A(Reserved) | |
| 0x3F | Query current online storage device | See 3.7.1 for details |
| 0x40 | Module returns an error data with this command | |
| 0x41 | Module reports a feedback with this command | |
| 0x42 | Query current status | See 3.7.2 for details |
| 0x43 | Query current volume | |
| 0x44 | Query current EQ | |
| 0x45 | N/A(Reserved) | |
| 0x46 | N/A(Reserved) | |
| 0x47 | Query number of tracks in the root of USB flash drive | |
| 0x48 | Query number of tracks in the root of micro SD card | |

| | | |
|------|---|-----------------------|
| 0x49 | N/A(Reserved) | |
| 0x4A | N/A(Reserved) | |
| 0x4B | Query current track in the USB flash drive | |
| 0x4C | Query current track in the micro SD Card | |
| 0x4D | N/A(Reserved) | |
| 0x4E | Query number of tracks in a folder | See 3.7.3 for details |
| 0x4F | Query number of folders in the current storage device | See 3.7.4 for details |

3.4. Examples of Sending Serial Commands

| Commands 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 1 st track |
| | 7E FF 06 03 00 00 02 FE F6 EF | 7E FF 06 03 00 00 02 EF | Specify playback of the 2 nd 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 |
| 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 EQ | 7E FF 06 07 00 00 01 FE F3 EF | 7E FF 06 07 00 00 01 EF | Specified EQ mode is POP |
| Specify single repeat playback | 7E FF 06 08 00 00 01 FE F2 EF | 7E FF 06 08 00 00 01 EF | Repeatedly play the 1 st 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 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 micro SD |
| Set sleep mode | 7E FF 06 0A 00 00 00 FE F1 EF | 7E FF 06 0A 00 00 00 EF | |
| Reset | 7E FF 06 0C 00 00 00 FE EF EF | 7E FF 06 0C 00 00 00 EF | |
| Play | 7E FF 06 0D 00 00 00 FE EE EF | 7E FF 06 0D 00 00 00 EF | |
| Pause | 7E FF 06 0E 00 00 00 FE ED EF | 7E FF 06 0E 00 00 00 EF | |
| Specify playback of a 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 all repeat playback | 7E FF 06 11 00 00 01 FE E9 EF | 7E FF 06 11 00 00 01 EF | |
| Specify playback of folder named "MP3" | 7E FF 06 12 00 00 01 FE E8 EF | 7E FF 06 12 00 00 01 EF | Play track "0001" in the folder "MP3" |
| | 7E FF 06 12 00 00 02 FE E7 EF | 7E FF 06 12 00 00 02 EF | Play track "0002" in the folder "MP3" |
| | 7E FF 06 12 00 00 FF FD EA EF | 7E FF 06 12 00 00 FF EF | Play track "0255" in the folder "MP3" |
| | 7E FF 06 12 00 07 CF FE 13 EF | 7E FF 06 12 00 07 CF EF | Play track "1999" in the folder "MP3" |
| | 7E FF 06 12 00 0B B8 FE 26 EF | 7E FF 06 12 00 0B B8 EF | Play track "3000" in the folder "MP3" |
| Inter cut an advertisement | 7E FF 06 13 00 00 01 FE E7 EF | 7E FF 06 13 00 00 01 EF | Inter cut track "0001" in the folder "ADVERT" |

| | | | |
|---|-------------------------------|-------------------------|--|
| | 7E FF 06 13 00 00 02 FE E6 EF | 7E FF 06 13 00 00 02 EF | Inter cut track "0002" in the folder "ADVERT" |
| | 7E FF 06 13 00 00 FF FD E9 EF | 7E FF 06 13 00 00 FF EF | Inter cut track "0255" in the folder "ADVERT" |
| Specify playback of a track in a folder that supports 3000 tracks | 7E FF 06 14 00 10 FF FD D8 EF | 7E FF 06 14 00 10 FF EF | Play track "0255" in the specified folder "01" |
| | 7E FF 06 14 00 17 CF FE 01 EF | 7E FF 06 14 00 17 CF EF | Play track "1999" in the specified folder "01" |
| | 7E FF 06 14 00 C0 01 FE 26 EF | 7E FF 06 14 00 C0 01 EF | Play track "0001" in the specified folder "12" |
| | 7E FF 06 14 00 C0 FF FD 28 EF | 7E FF 06 14 00 C0 FF EF | Play track "0255" in the specified folder "12" |
| | 7E FF 06 14 00 C7 CF FD 51 EF | 7E FF 06 14 00 C7 CF EF | Play track "1999" in the specified folder "12" |
| Stop playing inter-cut ad | 7E FF 06 15 00 00 00 FE E6 EF | | Go back and continue to play the music interrupted |
| Stop playback | 7E FF 06 16 00 00 00 FE E5 EF | | Stop software decoding |
| Specify repeat playback of a folder | 7E FF 06 17 00 00 02 FE E2 EF | 7E FF 06 17 00 00 02 EF | Specify repeat playback of the folder "02" |
| | 7E FF 06 17 00 00 01 FE E3 EF | 7E FF 06 17 00 00 01 EF | Specify repeat playback of the folder "01" |
| Set random playback | 7E FF 06 18 00 00 00 FE E3 EF | 7E FF 06 18 00 00 00 EF | |
| Set single repeat playback | 7E FF 06 19 00 00 00 FE E2 EF | 7E FF 06 19 00 00 00 EF | Turn on single repeat playback |
| | 7E FF 06 19 00 00 01 FE E1 EF | 7E FF 06 19 00 00 01 EF | Turn off single repeat playback |
| Set DAC | 7E FF 06 1A 00 00 00 FE E1 EF | 7E FF 06 1A 00 00 00 EF | Turn on DAC |
| | 7E FF 06 1A 00 00 01 FE E0 EF | 7E FF 06 1A 00 00 01 EF | Turn off DAC |
| 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 current EQ | 7E FF 06 44 00 00 00 FE B7 EF | 7E FF 06 44 00 00 00 EF | |
| Query number of tracks in the root of USB flash drive | 7E FF 06 47 00 00 00 FE B4 EF | 7E FF 06 47 00 00 00 EF | |
| Query number of tracks in the root of micro SD card | 7E FF 06 48 00 00 00 FE B3 EF | 7E FF 06 48 00 00 00 EF | |
| Query current track in the 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 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 number of tracks in a folder | 7E FF 06 4E 00 00 01 FE AC EF | 7E FF 06 4E 00 00 01 EF | Query number of tracks in the folder "01". |
| | 7E FF 06 4E 00 00 0B FE A2 EF | 7E FF 06 4E 00 00 0B EF | Query number of tracks in the folder "11". |
| Query number of folders in the current storage device | 7E FF 06 4F 00 00 00 FE AC EF | 7E FF 06 4F 00 00 00 EF | |

3.5. Returned Data from Module

3.5.1 Returned data after the module is powered on

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

2). The returned data from module after initialization means the current effective storage device/online equipment. For example, the module returns 7E FF 06 3F 00 00 03 xx xx EF. 0x3F is the returned command by module, and 0x03 represents USB flash drive and SD card are effective/online at the same time. See the details as below.

| Online Equipment | Returned Data |
|------------------------------------|-------------------------------|
| USB flash drive online | 7E FF 06 3F 00 00 01 xx xx EF |
| 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 |
| USB flash drive and SD card online | 7E FF 06 3F 00 00 03 xx xx EF |

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

3.5.2 Returned data after a track is finished playing

| Track Played | Returned Data |
|--|-------------------------------|
| 1 st track is finished playing in USB flash drive | 7E FF 06 3C 00 00 01 xx xx EF |
| 2 nd track is finished playing in USB flash drive | 7E FF 06 3C 00 00 02 xx xx EF |
| 1 st track is finished playing in SD card | 7E FF 06 3D 00 00 01 xx xx EF |
| 2 nd track is finished playing in SD card | 7E FF 06 3D 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 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 sequence, the returned data still follow the physical sequence, which needs to be noted.

3.5.3 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.

3.5.4 Returned data of errors

| Returned Data of Errors | Meaning Description |
|-------------------------------|--|
| 7E FF 06 40 00 00 01 xx xx EF | Module busy(this info is returned when the initialization is not done) |
| 7E FF 06 40 00 00 02 xx xx EF | Currently sleep mode(supports only specified device in sleep mode) |
| 7E FF 06 40 00 00 03 xx xx EF | Serial receiving error(a frame has not been received completely yet) |
| 7E FF 06 40 00 00 04 xx xx EF | Checksum incorrect |
| 7E FF 06 40 00 00 05 xx xx EF | Specified track is out of current track scope |
| 7E FF 06 40 00 00 06 xx xx EF | Specified track is not found |
| 7E FF 06 40 00 00 07 xx xx EF | Inter-cut error(a inter-cut 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 into sleep mode |

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

3.5.5 Returned data after a storage device is plugged in or pull 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 cable connected to PC is plugged in | 7E FF 06 3A 00 00 04 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 |
| 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. 0x01, 0x02, 0x04 represent USB flash drive, SD card and USB cable connected to PC respectively.

3.6. Detailed Annotation of Control Commands

3.6.1.Specify playback of a track(in the root directory of a storage device)

The available selective tracks is from 0001.mp3/wav to 3000.mp3/wav 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.

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

7E --- Start byte

FF --- Version Information

06 --- Data length (checksum not included)

03 --- Actual command(specify playback of a track)

00 --- 0x01: need feedback, 0x00:no need feedback

00 --- Most significant byte of the track(MSB of Parameter)

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

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

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

EF --- End byte

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

3).If you choose to play the 1000th song(track), firstly convert 1000 to hexadecimal. It is double-byte, i.e. 0x03E8. MSB=0x03; LSB=0xE8

4).And so on in the same way to the other operations, as in the embedded area hexadecimal is the most convenient operation method.

3.6.2.Specify volume

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

2).For example, if specify the volume to level 15, send the command 7E FF 06 06 00 00 0F FF D5 EF.

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

3.6.3. Specify single repeat playback

| | |
|--|-------------------------------|
| Start repeatedly play the first track | 7E FF 06 08 00 00 01 xx xx EF |
| Start repeatedly play the second track | 7E FF 06 08 00 00 02 xx xx EF |

1).We added this control command 0x08, to meet the needs that some users need single repeat playback.

2).During single repeat playback, you can still normally execute the operations Play/Pause, Previous, Next, Volume+/-, EQ and so on. You can specify single track playback or make it sleep to turn off single repeat playback status.

3.6.4. Specify playback of a device

| | |
|-------------------------------------|-------------------------------|
| Specify playback of USB flash drive | 7E FF 06 09 00 00 01 FE F1 EF |
| Specify playback of SD card | 7E FF 06 09 00 00 02 FE F0 EF |

1).The module supports two types of playback devices(USB flash drive and SD card). The device must be on-line, so it can be specified playback. It automatically detects if a device is on-line or not. No need users' attention.

2).It automatically enters into the standby status after specifying a device, waiting 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.

3.6.5. Specify playback of a track in a folder

| | |
|--|-------------------------------|
| 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.



3.6.6. Set all repeat playback

| | |
|---------------------------|-------------------------------|
| Start all repeat playback | 7E FF 06 11 00 00 01 xx xx EF |
| Stop all repeat playback | 7E FF 06 11 00 00 00 xx xx EF |

- 1).We added such a control command 0x11, to meet the needs that some users need repeated playback of all tracks in the root directory of a storage device.
- 2).During all repeat playback, you can still normally execute the operations Play/Pause, Previous, Next, Volume+/-, EQ and so on.
- 3).After starting all repeat playback, the module will play all the tracks in the device ceaselessly again and over again until it receives a command for stop or pause.

3.6.7. Specify playback of folder named "MP3"

| | |
|---------------------------------------|-------------------------------|
| Play track "0001" in the folder "MP3" | 7E FF 06 12 00 00 01 FE E8 EF |
| Play track "0002" in the folder "MP3" | 7E FF 06 12 00 00 02 FE E7 EF |
| Play track "0255" in the folder "MP3" | 7E FF 06 12 00 00 FF FD EA EF |
| Play track "1999" in the folder "MP3" | 7E FF 06 12 00 07 CF FE 13 EF |
| Play track "3000" in the folder "MP3" | 7E FF 06 12 00 0B B8 FE 26 EF |

- 1).We extended such single folder function on the basis of specifying playback of folders. The folder must be named "MP3".
- 2).This special folder supports max. 65536 songs(tracks), but the operation speed will get slow along with the folder

become large, so we suggest users put no more than 3000 songs(tracks) in this folder. Audio files need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav,3000.mp3/3000.wav in this folder as shown below.



3.6.8. Inter cut an advertisement in folder “ADVERT”

| | |
|---|-------------------------------|
| Inter cut track "0001" in the folder “ADVERT” | 7E FF 06 13 00 00 01 FE E7 EF |
| Inter cut track "0002" in the folder “ADVERT” | 7E FF 06 13 00 00 02 FE E6 EF |
| Inter cut track "0255" in the folder “ADVERT” | 7E FF 06 13 00 00 FF FD E9 EF |
| Inter cut track "1999" in the folder “ADVERT” | 7E FF 06 13 00 07 CF FE 12 EF |
| Inter cut track "3000" in the folder “ADVERT” | 7E FF 06 13 00 0B B8 FE 25 EF |

- 1).This module supports inter-cut advertisements during playback of a track, so that it can meet some special needs for some applications.
- 2).After sending the command 0x13, the system will save the IDV3 information of the track being played and pause, then it will play the specified inter-cut track(advertisement). When the inter-cut 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 “ADVERT” in the storage device, and put the tracks(ads) you need in the folder, and rename the files as “0001.mp3/wav”, 0002.mp3/wav.
- 4).If you send an inter-cut command when the module is at Pause status or Stop status, it will not work and there will be an returned error information. In the course of an inter-cut, you can continue to inter cut the other tracks(ads). When the last inter-cut track goes to the end, the system still goes back to the IDV3 position saved at the first time.
- 5).Audio files stored in this special folder need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav,3000.mp3/3000.wav as shown below.



3.6.9. Specify playback of a track in a folder that supports 3000 tracks

| | |
|--|-------------------------------|
| Play track "0255" in the specified folder "01" | 7E FF 06 14 00 10 FF FD D8 EF |
| Play track "1999" in the specified folder "01" | 7E FF 06 14 00 17 CF FE 01 EF |
| Play track "0001" in the specified folder "12" | 7E FF 06 14 00 C0 01 FE 26 EF |
| Play track "0255" in the specified folder "12" | 7E FF 06 14 00 C0 FF FD 28 EF |
| Play track "1999" in the specified folder "12" | 7E FF 06 14 00 C7 CF FD 51 EF |

In order to meet some users' needs that each folder is able to manage more tracks, we specially added this command for users to use. It supports max. 15 folders only(from folder 01 to folder 15), and each folder can store 3000 tracks.

1).The command byte is 0x14.

2).For example, if we specify playback of track "1999" in the folder "12", we need to send the serial command

7E FF 06 14 00 C7 CF FD 51 EF.

0xC7 and 0xCF are parameter, and the combined data is 0xC7CF, and totally 16 bytes.

The high 4 bytes represent the folder name; C here means the folder "12".

The low 12 bytes represent the file name; 7CF here means the file "1999".

3).Audio files stored in these folders need to be renamed as 0001.mp3/0001.wav, 0002.mp3/0002.wav, 0003.mp3/0003.wav,3000.mp3/3000.wav as shown below.



3.6.10. Stop

| | |
|----------------------------|-------------------------------|
| Stop playing advertisement | 7E FF 06 15 00 00 00 FE E6 EF |
| Stop all playback tasks | 7E FF 06 16 00 00 00 FE E5 EF |

1).During playback of the module, there is two modes to stop. One is to stop playing the inter-cut advertisement, and go back and continue to play the music interrupted, and the other mode is to stop all playback(stop decoding).

2).For example, suppose the module is playing an inter-cut advertisement, and now if send a stop command 0x16, it will stop all playback tasks.

3.6.11. Specify repeat playback of a folder

| | |
|--|-------------------------------|
| Specify repeat playback of the folder "02" | 7E FF 06 17 00 00 02 FE E2 EF |
| Specify repeat playback of the folder "01" | 7E FF 06 17 00 00 01 FE E3 EF |

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

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

3.6.12. Set random playback

| | |
|---|-------------------------------|
| Random playback of the whole storage device | 7E FF 06 18 00 00 00 FE E3 EF |
|---|-------------------------------|

1).This command is used to randomly play sound files in the storage device according to physical sequence and no matter if there is a folder or not in the device. The first sound file that is conducted to be played is the first one in the device.

3.6.13. Set repeat playback of current track

| | |
|---------------------------------|-------------------------------|
| Turn on single repeat playback | 7E FF 06 19 00 00 00 FE E2 EF |
| Turn off single repeat playback | 7E FF 06 19 00 00 01 FE E1 EF |

1).During playback, send the turn-on command, and it will repeatedly play the current track. If the module is at Pause or Stop status, it will not respond to this command.

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

3.6.14. Set DAC

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

When the module is powered on, DAC is turned on by default. It is not turned off until it is set up by sending the command.

3.7. Detailed Annotation of Main Query Commands

3.7.1 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 all of USB flash drive, SD card, and PC online(PC online means module is connecting with PC via a USB cable).

3.7.2 Query current status

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

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

2). Interpretation of returned data

| Returned Data | Status |
|-------------------------------|---|
| 7E FF 06 42 00 01 01 xx xx EF | A track in USB flash drive is being played |
| 7E FF 06 42 00 02 02 xx xx EF | A track in SD card is paused playing |
| 7E FF 06 42 00 01 00 xx xx EF | A track in USB flash drive 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 | |
|--------------------|----------------------|--------------------|---------|
| 0x01 | USB flash drive | 0x00 | Stopped |
| 0x02 | SD card | 0x01 | Playing |
| 0x10 | Module in sleep mode | 0x02 | Paused |

3.7.3 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.

3.7.4 Query number of folders in the current storage device

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

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

3.8. Example of Serial Program

Code example: specify playback of a track

```

/*****
- 实现功能：实现芯片上电分别指定播放第一曲和第二曲，基本的程序供用户测试
- 运行环境：STC 晶振：11.0592M 波特率:9600
- 备注：在普中科技的51开发板上调试 OK --- STC89C516RD+
1、该测试程序必须是模块或者芯片方案中有设备在线，譬如U盘、TF卡
*****/
#include "REG52.h"

#define COMM_BAUD_RATE 9600 //串口波特率
#define OSC_FREQ 11059200 //运行晶振：11.05926MHZ
static INT8U Send_buf[10] = {0};

void Delay_Ms(INT32U z)
{
    INT32U x=0, y=0;

```

```
for(x=110 ; x>0 ;x--)
for(y=z; y>0;y-- );
}
```

```
/******
- 功能描述: 串口 1 初始化
- 注: 设置为 9600 波特率
*****/
```

```
void Serial_init(void)
{
    TMOD = 0x20;           // 设置 T1 为波特率发生器
    SCON = 0x50;           // 0101,0000 8 位数据位, 无奇偶校验
    PCON = 0x00;           //PCON=0;
    TH1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);//设置为 9600 波特率
    TL1=256-(OSC_FREQ/COMM_BAUD_RATE/32/12);
    TR1 = 1;               //定时器 1 打开
    REN = 1;               //串口 1 接收使能
    ES = 1;                //串口 1 中断使能
}
```

```
void Uart_PutByte(INT8U ch)
{
    SBUF = ch;
    while(!TI){;}
    TI = 0;
}
```

```
/******
- 功能描述: 串口向外发送命令[包括控制和查询]
- 参数说明: CMD:表示控制指令, 请查阅指令表, 还包括查询的相关指令
            feedback:是否需要应答[0:不需要应答, 1:需要应答]
            data:传送的参数
*****/
```

```
void SendCmd(INT8U len)
{
    INT8U i = 0 ;
    Uart_PutByte(0x7E); //起始
    for(i=0; i<len; i++)//数据
    {
        Uart_PutByte(Send_buf[i]) ;
    }
    Uart_PutByte(0xEF) ;//结束
}
```

```
/******
- 功能描述: 求和校验
- 和校验的思路如下:
    发送的指令, 去掉起始和结束。将中间的 6 个字节进行累加, 最后取反码。接收端就将接收到的一帧数据, 去掉起始和结束。
    将中间的数据累加, 再加上接收到的校验字节。刚好为 0.这样就代表接收到的数据完全正确。
*****/
```

```
void DoSum( INT8U *Str, INT8U len)
{
    INT16U xorsum = 0;
    INT8U i;
    for(i=0; i<len; i++)
    {
        xorsum = xorsum + Str[i];
    }
    xorsum = 0 -xorsum;
    *(Str+i) = (INT8U)(xorsum >>8);
}
```

```

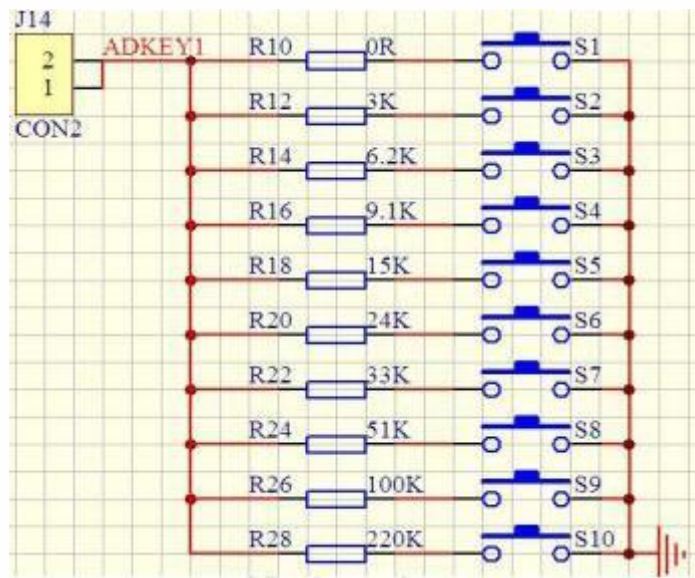
*(Str+i+1) = (INT8U)(xorsum & 0x00ff);
}

void Uart_SendCMD(INT8U CMD ,INT8U feedback , INT16U dat)
{
    Send_buf[0] = 0xff;    //保留字节
    Send_buf[1] = 0x06;    //长度
    Send_buf[2] = CMD;     //控制指令
    Send_buf[3] = feedback;//是否需要反馈
    Send_buf[4] = (INT8U)(dat >> 8);//datah
    Send_buf[5] = (INT8U)(dat);    //datal
    DoSum(&Send_buf[0],6);    //校验
    SendCmd(8);    //发送此帧数据
}

void main()
{
    Serial_init() ;//串口寄存器的初始化设置
    Uart_SendCMD(0x03 , 0 , 0x01) ;//播放第一首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x02) ;//播放第二首
    Delay_Ms(1000) ;//延时大概 6S
    Uart_SendCMD(0x03 , 0 , 0x04) ;//播放第四首
    while(1) ;
}

```

4. ADkey Control

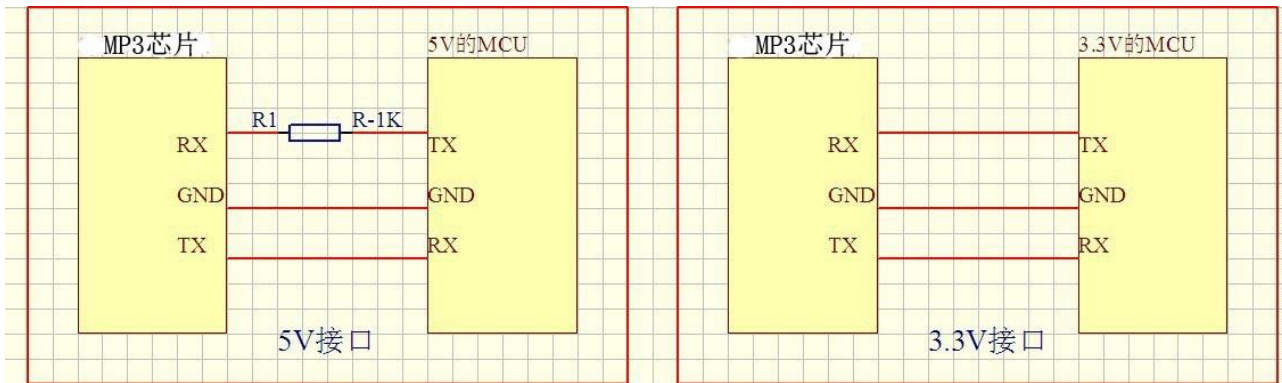


| Button | Resistor Value | Short Press | Long Press |
|--------|----------------|--------------------------------------|------------|
| S1 | 0R—0V | Play back 1 st audio file | N/A |
| S2 | 3K—0.7V | Play back 2 nd audio file | N/A |
| S3 | 6K2—0.7V | Play back 3 rd audio file | N/A |
| S4 | 9K1—1V | Play back 4 th audio file | N/A |
| S5 | 15K—1.3V | Play back 5 th audio file | N/A |
| S6 | 24K—1.7V | Play back 6 th audio file | N/A |

| | | | |
|-----|-----------|--------------------------------------|-----|
| S7 | 33K—2V | Play back 7 th audio file | N/A |
| S8 | 51K—2.3V | Play back 8 th audio file | N/A |
| S9 | 100K—2.7V | Play back 9 th audio file | N/A |
| S10 | 220K—3V | Stop | N/A |

5. Connection of Circuit References

5.1. Connection of UART Serial Port

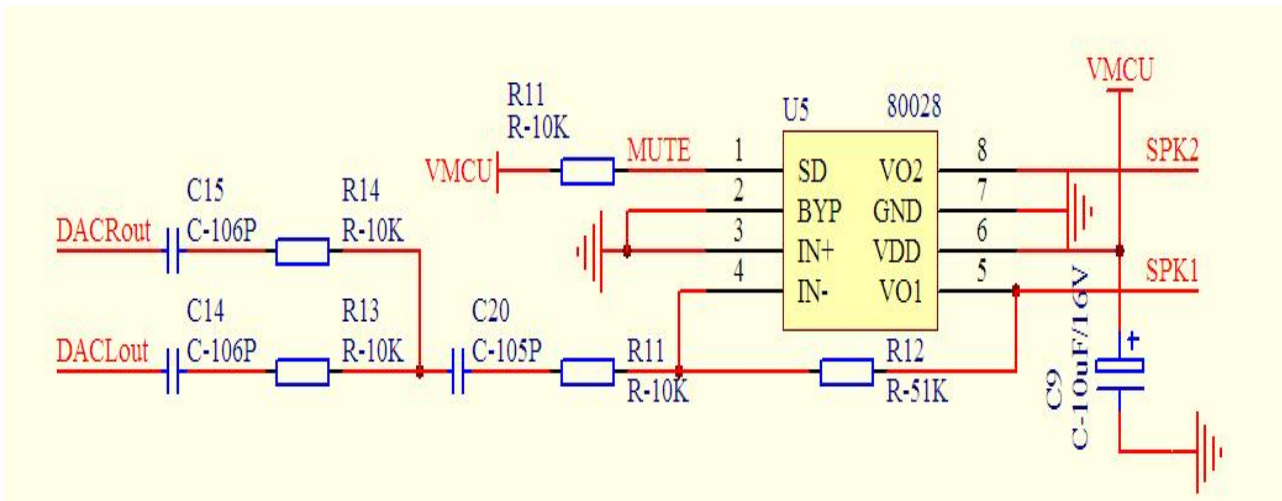


Connection to a 5V MCU

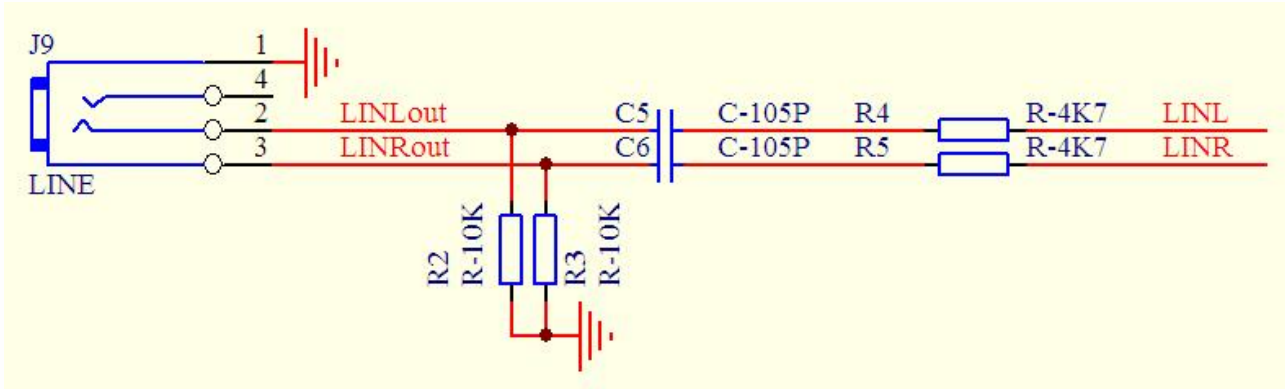
Connection to a 3.3V MCU

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

5.2. Connection of an external mono amplifier for audio output



5.3. Connection of an earphone/2-channel stereo amplifier for audio output



R4 and R5 are the limiting resistors that prevent the audio amplitude is too large, which would cause a bad stability. C5 and C6 are blocking capacitor that prevent the DC level of the external audio source affects offset of the chip; R2 and R3 are reserved resistors that are used for large power amplifier.

6. Features of GPIO

| I/O Input Features | | | | | | |
|---------------------|---------------------------|--------|---------|---------|------|----------------|
| Item | Description | Min | Typical | Max | Unit | Test Condition |
| VIL | Low-Level Input Voltage | -0.3 | - | 0.3*VDD | V | VDD=3.3V |
| VIH | High-Level Input Voltage | 0.7VDD | - | VDD+0.3 | V | VDD=3.3V |
| I/O Output Features | | | | | | |
| Item | Description | Min | Typical | Max | Unit | Test Condition |
| VOL | Low-Level Output Voltage | - | - | 0.33 | V | VDD=3.3V |
| VOH | High-Level Output Voltage | 2.7 | - | - | V | VDD=3.3V |

7. PCB Size Drawing (unit: mm)

