GDB3200B MP3Player Module

1. Summary

1.1. Introduction

GD3200B MP3PLayer Module is a serial MP3 module which provides integrated MP3 and WMV hardware decoding. The software supports a TF card with a FAT16 or FAT32 file system. Simple serial commands are used to play and manage tracks without the need to access the underlying device operations. Ease of use, stability and reliability are the most important features of this module.

1.2. Features

GD3200B. Note that the module based on the MH2024-24SS chip appears to be exactly the same.

Support for MP3 and WMV decoding

Support for sampling rates of 8KHz, 11.025KHz, 12KHz, 16KHz, 22.05KHz, 24KHz, 32KHz, 44.1KHz and 48KHz

24-bit DAC output, Stereo. Dynamic range: 90dB, SNR: 85dB

Supports FAT16 or FAT32 file system, maximum TF card size is 32Gb

A variety of control modes: serial mode (UART. USB), A/D key control mode

Broadcast spots feature: Playback can be paused for a track to be inserted

Built-in 3W Mono amplifier (device 8002).

Two modes of file organization: Named folders ‘01’ to ‘99’ with up to 255 tracks each, or a single folder (‘MP3’) with up to ~3000 tracks.

30 volume levels, 6 equalization settings.

1.2.1. Application

Car navigation voice broadcast

Road transport inspectors, toll station voice prompts

Railway station, bus safety inspection voice prompts

Electricity, communications, financial/business hall voice prompts

Vehicle operation confirmation and prompting

The public security border control broadcast voice prompts

Multi-channel voice alarm or equipment operating guide voice

Vehicle safe driving voice notices

Electromechanical equipment failure alarm

Fire alarm voice prompts

Automatic broadcast equipment, regular broadcast

Personal MP3 player

1.3. Disclaimer

This document is a heavily edited version of the original. As such, no part of this document can be taken as an accurate description of the product concerned and no action of any sort should be undertaken based on information in this document without first checking the accuracy of the statement against an actual example of the module. Note that there are many different manufacturers of the module and therefore any particular module could differ from any description provided in this document.

The description in this document relates to operation with a TF card and MP3 files. Operation with other sources of files or other file formats has not been examined.

This document is based on examples using the GD3200B and M2024K-24SS.

1.4. TF Card file organisation

The module operates with two variations of file layout within the TF card. These variations are mostly incompatible and implementing both within a single application is likely to be difficult.

1.4.1 Single folder.

The TF card contains a single folder named 'MP3'. Tracks are contained within this folder. Up to 65535 tracks – the maximum for a 16-bit integer - could be included but the recommended maximum is about 3,000: with more than this processing times may become unacceptable.

In this mode the track numbering strictly follows the order in which the files are laid out in the index of the file system of the TF card. Note that this sequence may not be obvious – for instance, the Windows GUI will not display the files in this sequence, although a Windows console DIR command will if no sort sequence is specified. There are some Windows applications that display a file list in the index sequence, but most will use Windows functions, which will sort the list. There is no simple way to directly control the sequence of files in the index, other than by copying them into the folder in the required order.

All commands that use (or imply) a track number use the sequence from the folder index.

The file name is ignored. If it is important to match file names to track numbers then the files must be copied to the TF-card in the sequence of their filenames. Standard Windows copy functions (cut/copy and paste, or select and drag) do not guarantee any particular order.

The 0x48 query (TF Folder File Count) command is available for ensuring only valid track numbers are selected. The returned value will refer to the MP3 folder.

1.4.2 Multiple Folder.

This is also called 'Named Files'.

The TF card has between 1 and 99 folders named '01', '02'…'99'. Each folder contains up to 255 files named '001?.mp3', 002?.mp3'….'255?.mp3'. The ? indicates any legal filename characters.

In this mode the folder is selected based on a match of the folder name to the folder number. The track is selected based on a match of the first three characters of the file name to the track number.

This naming selection applies only to the 0x0F (Play Track from Folder) command.

All other commands that use (or imply) a track number use the track number identified from the TF card file system index. The effect is that in Multiple Folder mode commands such as Next, Previous, Repeat Sequential, Repeat Random etc should be implemented in the application code: the equivalent module commands will give results that may be inconsistent with the folder/file numbering scheme.

The 0x4F (TF Folder Count) and 0x4E (TF Folder File Count) queries are available for ensuring only valid folder and track numbers are selected. The returned value for the 0x4E query will refer to the folder most recently selected with the 0x0F (Play Track From Folder) command. Note that this command is undocumented in the original reference.

2. Module Application

2.1. Specification Description

Item	Description
MP3Format	1、Support 11172-3 and ISO13813-3 layer3 audio decoding
	2、Support sampling rate (KHZ):8/11.025/12/16/22.05/24/32/44.1/48
	3、Support Normal、Jazz、Classic、Pop、Rock，Bass
UART Port	Standard Serial (TTL Level); default baud rate is 9600*
Working Voltage	DC3.2~5.0V; Typ. DC4.2V
Standby Current	20mA
Operating Temperature	-40~+70
Humidity	5% ~95%

Table 2.1 Specification Description

* Some references indicate this is adjustable, but there is no other information available

2.2. Pin Description

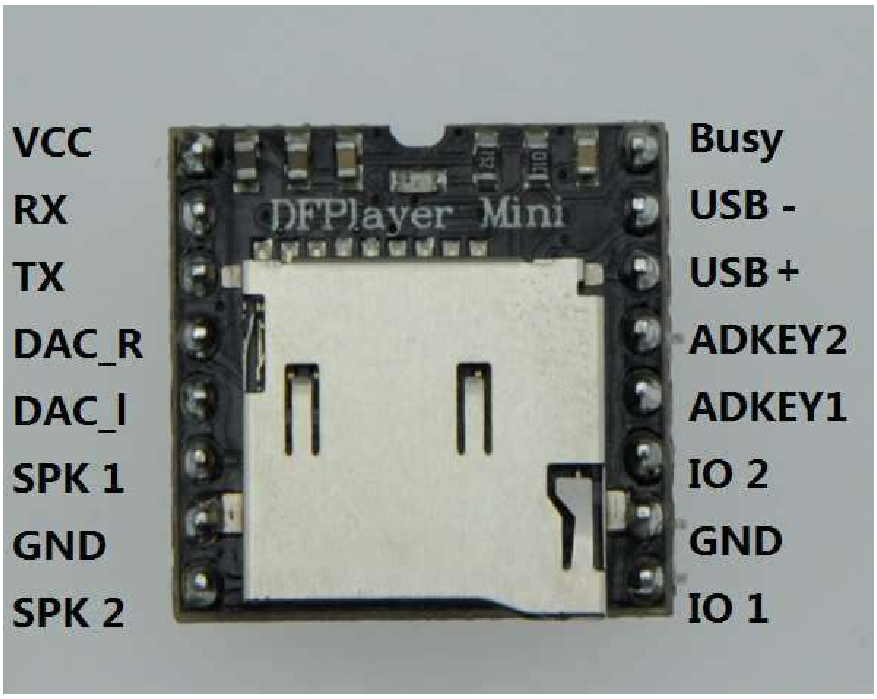


Figure 2.1 Module pinout

No	Pin	Description	Note
1	VCC	Input Voltage	DC3.2~5.0V;Typ: DC4.2V
2	RX	UART serial input	TTL (3.3v)
3	TX	UART serial output	TTL (3.3v)

4	DAC_R	Audio output right channel	Earphone and amplifier output
5	DAC_L	Audio output left channel	Earphone and amplifier output
6	SPK2	Speaker-	Speaker (3W maximum)
7	GND	Ground	Power GND
8	SPK1	Speaker+	Speaker (3W maximum)
9	IO1	Trigger port 1	Digital input for key-press mode
10	GND	Ground	Power GND
11	IO2	Trigger port 2	Digital input for key-press mode
12	ADKEY1	AD Port 1	Analog/Digital input for key-press mode.
13	ADKEY2	AD Port 2	Analog/Digital input for key-press mode.
14	USB+	USB+ DP	USB Port
15	USB-	USB- DM	USB Port
16	BUSY	Playing Status	Low means playing, High means idle

Table 2.2 Pin Description

3. Serial Communication Protocol

Serial port is a common communication standard in the industrial control field. The module implements an industrial level of optimization, with frame checksum, retransmission, error handling, and other measures to significantly strengthen the stability and reliability of communication. User can implement RS485 for networking for longer distance transmission. Serial communication baud rate is 9600. (Note: the original documentation states that the baud rate is adjustable but provides no information about how to adjust it).

Note that all communication with the module is through the commands and queries listed in this section – there is no facility for accessing any internal components (for example DAC control or TF card).

3.1. Serial Communication Format

Serial communication with the player is through a set of commands as listed below. For some commands the player will issue a response. The format for the command and response is set out in table 3.1.

Support for asynchronous serial communication mode (UART)
Baud Rate (default) :9600 bps
Data bits: 8
Stop Bits: 1
Flow Control: None. Command acknowledgement is optional.

Format : $S VER Len CMD Feedback para1 para2 checksum $O

#	Field	Size	Description	Usage
0	$S	Byte	Start byte (0x7E)	Command string identifier
1	VER	Byte	Version (0xFF)	Version Information (not used)
2	Len	Byte	Number of bytes after $S (6)	Command length is fixed (checksum and end byte are not included)
3	CMD	Byte	Command	Indicates the specific operation
4	Feedback	Byte	Feedback Request	1: feedback expected 0: no feedback
5	Para1	Byte	Parameter 1	High data byte (DH)
6	Para2	Byte	Parameter 2	Low data byte (DL)
7,8	Checksum	Word	Checksum	16-bit accumulation bytes 1~6, 2s complement
9	$O	Byte	End byte (0xEF)	Message terminator

Table 3.1 Message Format

For example, if we specify play NORFLASH, you need to send: 7E FF 06 09 00 00 04 FF DD EF. Data length is 6, which are 6 bytes [FF 06 09 00 00 04] not counting the start, end, and checksum.

Note that there is no additional command terminator, such as CR or CR/LF.

3.1.1 Checksum calculation.

The checksum is a 16-bit unsigned integer value calculated from bytes VER to para2 (bytes 1 to 6) of the Command or Query. If an incorrect checksum is sent error 3 (Validation) will be returned and the command or query is not processed

The checksum is calculated by adding the byte values into a 16-bit variable (ignoring overflow) and subtracting from 0x10000.

It is recommended that applications validate the checksum for module responses.

3.2 Serial Communication Commands

Note: The meaning of the commands and queries listed below depends on the folder/file layout of the card. The below description must be interpreted in terms of the folder/file layout that is implied in the description. See 1.4 above.

In each case the commands will likely default to something useful if they are used with an inappropriate folder/file format, but the results may not match the description. For example, using the 0x03 command when a TF card is organised with multiple folders will select the track based on the file sequence in the card directory, which might refer to any folder.

Playback mode (0x08) is designed for one large MP3 folder and will not work correctly with multiple folders. When using multiple folders the playback mode should be implemented in user code.

3.2.1. Commands.

Commands are used to configure and control the module. Commands do not generate a response, however if the acknowledge byte is set in the command string then the acknowledge response will be sent. Some commands take a value in the parameter field as indicated in the table.

Command Code	Function Description	Parameters(16 bit)
0x01	Next Track, or first track if already at last track.	0
0x02	Previous Track, or last track if already at track 0.	0
0x03	Specify track.	Track Number (0-2999)
0x04	Increase volume. Max is 30.	0
0x05	Decrease volume. Min is 0.	0
0x06	Specify volume	Volume Setting (0-30)
0x07	Specify EQ	0: Normal 1: Pop 2: Rock 3: Jazz 4: Classic 5: Bass
0x08	Specify playback mode Note (1)	0: Repeat current track 1: Folder Repeat 2: Single Repeat 3: Random *4:* *5: Single*
0x09	Specify playback source Note (1)	1: U_Disk 2: TF Card 3: AUX 4: SLEEP 5: Flash
0x0A	Enter into standby	0
0x0B	Normal working (Resume from standby)	0

0x0C	Reset module	0
0x0D	Playback (Resume from Pause) Note (2)	0
0x0E	Pause	0
0x0F	Specify folder and file to playback	Folder Number (DH 1-99) File Number (DL 1-255) (See 3.4.4)
0x10	Volume adjust set	DH: 1 = Enable volume adjust DL: Gain (0-30)
0x11	Start / Stop Repeat play	1 = Start repeat play 0 = Stop repeat play

Table 3.2.1 Command Codes

Note (1). Folder and track are re-set.
Note (2). Play resumes at the start of the next track.

3.2.2. Queries.

Queries are used to obtain information from the module. The query commands use the same command code for the query and for the response. The parameter field for the query is ignored. The response contains detail in the parameter field for the responses as indicated in the table. If the acknowledge byte in the query is set then the query is acknowledged before the response is sent. Note that several of the responses generated by the module are created by an event rather than a query (initialization, media insert/remove, and track finish).

Query Code	Function Description	Response Parameters(16 bit)
0x3A	* U-Disk track finished playing	Track number (See 3.3.2)
0x3B	* TF track finished playing	Track number (See 3.3.2)
0x3C	* Flash track finished playing	Track number (See 3.3.2)
0x3D	* Send initialization parameters	0 - 0x0F Each bit represents one device. (See 3.3.1)
0x40	Error, request retransmission	Error code (See 3.3.4)
0x41	# Acknowledge	None
0x42	Query the current status	0 = Paused 1 = Playing
0x43	Query the current volume	Volume (0-30)
0x44	Query the current equalization	EQ (0-5)
0x45	Query the current playback mode	Mode (0-3)
0x46	Query the current software version	Version (0xFF)
0x47	Query the total number of U-Disk files	Number of files (zero if no device)
0x48	Query the total number of TF Card files	Number of files (zero if no device)
0x49	Query the total number of flash files	Number of files (zero if no device)

0x4A	Keep on	(?)
0x4B	^Query the current track of TF card	Track number (Code 0x4C!!)
0x4C	^Query the current track of U-Disk	Track number
0x4D	^Query the current track of Flash	Track number
0x4E	Get Folder Files	Number of Files in folder
0x4F	Query number of TF folders	Number of folders

Table 3.2.2 Query Codes

* The response is initiated by the relevant event, not by a query from the MCU.
# Generated if the Ack bit is set in the command or query
^ Do not query for a device that doesn't exist!

Note 1: Response to query 0x4E (for multi-folder operation) does not include the folder number in the response. The returned figure applies to the folder most recently used for a 0x0F (Play Folder and Track) command. This query may be unreliable.

Note 2: File and folder number queries may terminate the current track and reset the song number.

3.3. Returned Data from Module

3.3.1. Returned Data from Module at Power-on

The module requires a certain amount of time for initialization, dependant on data source (U-disk, TF card, flash, etc.), the media size, number of files, format etc.. This is generally in the range of 1.5 – 3 sec but can be much longer. If module initialization data (0x3F) has not been received within a reasonable time following initialization this indicates a module initialization error, possibly because no media source was detected. Correct the problem and reset the module power supply so the module can re-detect the connected hardware.

The module initialization data indicates the online devices using a 4-bit bitmap in DL. If multiple devices are online the values are OR'ed.

Bit	Value	Media Source
0	0x1	U-Disk
1	0x2	TF-Card
2	0x4	USB
3	0x8	Flash

Table 3.3.1.1 Initialization data device bit values

For instance 7E FF 06 3F 00 00 01 xx xx EF (DL = 0x01) indicates that only the U-disk was online during power-on. Other bit values are listed in the below table.

Device	Notification
U-Disk on-line	7E FF 06 3F 00 00 01 xx xx EF
TF Card on-line	7E FF 06 3F 00 00 02 xx xx EF
PC (USB) on-line	7E FF 06 3F 00 00 04 xx xx EF
FLASH on-line	7E FF 06 3F 00 00 08 xx xx EF

U-disk & TF Card on-line	7E FF 06 3F 00 00 03 xx xx EF

Table 3.3.1.2 Initialization Response Examples

3).Module will not process or respond to commands sent by MCU until module initialization is complete and the initialization response is sent. Sending commands to the module before initialization is complete will not affect the initialization. A delay is also recommended after the initialization message is returned.

4) Module initialization status is listed below:

Setting	Status
Media Source	As reported (see above)
Equalization	0
Volume	30
Playback Mode	5 (??)
Current folder	First available from media index
Current file	First available from media index

Table 3.3.1.4 Initialization Status

3.3.2. Returned Data for Track Finished Playing

The module returns information when a track finishes playing. The command code in the response indicates the source device and the parameter value indicates the track number. The module will send the following responses for the corresponding events:

Playback Event	Notification
U-Disk finish playback 1st track	7E FF 06 3C 00 00 01 xx xx EF
U-Disk finish playback 2nd track (etc)	7E FF 06 3C 00 00 02 xx xx EF
TF card finish playback 1st track	7E FF 06 3D 00 00 01 xx xx EF
TF card finish playback 2nd track (etc)	7E FF 06 3D 00 00 02 xx xx EF
Flash finish playback 1st track	7E FF 06 3E 00 00 01 xx xx EF
Flash finish playback 2nd track (etc)	7E FF 06 3E 00 00 02 xx xx EF

Table 3.3.2 Track Finish Responses

Default mode of operation is for the module to enter into pause status automatically after the specified track is played.

The module includes a dedicated I/O pin for pause status indication: Pin 16 (Busy). Output is high for playback and low for pause or sleep.

Continuous playback applications can be achieved as below. If the module finishes the first track of the TF card, it will return a track-finished response. For instance:

7E FF 06 3D 00 00 01 xx xx EF

3D ---- U-disk command (??)
00 01 ---- finished playing track 1

If the external MCU receives this command it should wait 100ms and then send the Playback command [7E FF 06 0D 00 00 00 FF EE EF]. At the end of the track the module will initialize the next track, so sending the Playback command in response to the track-finished response causes the module to play tracks in sequence. Note that this numbering corresponds to the file sequence as stored in the card file system index, which may not be the numerical sequence of file names.

When the module finishes playing a song, the track pointer will be incremented automatically to point to the next song. If the MCU sends a Next command the module will skip that song and play the following one.

If the module finishes playing the last track, the player will automatically jump to the first track, and pause.

After specifying the device, the module play pointer will point to the first track of the device root directory, and enters the pause state, and wait for the MCU to send a track playing command.

3.3.3. Returned Data of Module - Acknowledge

Command acknowledge	7E FF 6 41 0 0 0 xx xx EF

Table 3.3.3 Acknowledge Command

In order to strengthen the stability of the data communication, acknowledge processing has been implemented. The Feedback byte (byte 5 of the command or query sequence) can be set to indicate that the module should respond to each command or query with an acknowledge response. This can be used to ensure each communication gets an acknowledgement signal which will indicate the module has successfully received data sent by the MCU and is processing the command.

Acknowledgement messages will be sent for each command or query where the Ack bit is set. While commands are generally acted on immediately, the acknowledgement may take some time (a few seconds) to be returned. For a query, the acknowledgement will always precede the response. It is not necessary to wait for the acknowledgement before sending the next command or query, however the size of the module receive buffer is unknown. Recommended operation is to use the acknowledge message and send one command or query at a time.

Note that the parameter values in the acknowledgment message are meaningless.

For example, the communication sequence to set the volume and then query the volume with acknowledgements is:

Sender	Command / Query	Message
MCU (Command)	7E FF 06 06 01 00 0A FE EA EF	Volume 10
Module	7E FF 06 41 00 00 00 FE BA EF	Acknowledge
MCU (Query)	7E FF 06 43 01 00 00 FE B7 EF	Current Volume
Module	7E FF 06 41 00 00 00 FE BA EF	Acknowledge
Module	7E FF 06 43 00 00 0A FE AE EF	Volume is 10

Table 3.3.3.1 Command/Query/Acknowledge Example

3.3.4. Returned Data of Module - Error

In order to strengthen the stability of the data communication, there is an error response feature. Module will respond with error information after receiving invalid data. The response code for an error is 0x40.

Error Condition	Notification

Module is busy (0)	7E FF 06 40 00 00 00 xx xx EF
Frame Error – not all data received (1)	7E FF 06 40 00 00 01 xx xx EF
Verification error (2)	7E FF 06 40 00 00 02 xx xx EF
Invalid parameter (3)	7E FF 06 40 00 00 03 FE B8 EF
Other (4)	*7E FF 06 40 00 00 04 FE B7 EF*

Table 3.3.4 Error Response

In the case of a poor communication environment, it is strongly recommended that customers process this command. If the application environment is adequate there is no need to handle it.

The module returns busy during initialization because the module needs to initialize the file system. This can take several seconds.

3.3.5. Returned Data of Module – Media Insert and Remove

To improve the flexibility of the module, responses are provided which are generated by the insertion or removal of the source device media.

Event	Notification
Insert U-disk	7E FF 06 3A 00 00 01 xx xx EF
Insert TF card	7E FF 06 3A 00 00 02 xx xx EF
Remove U-disk	7E FF 06 3B 00 00 01 xx xx EF
Remove TF card	7E FF 06 3B 00 00 02 xx xx EF

Table 3.3.5 Media Insert/Remove Responses

When a device media is inserted playback will default to the first track of device root directory as an ‘audition’. If users do not need this feature, you can wait 100ms after receiving the message of media inserted and then send a pause command.

3.4. Play Commands

3.4.1. Specify Track to Play – 0x03 Command

The song selection commands can be used for a number of songs that range from 0 to 2999. In fact, the module can support more tracks than this, but if a large number of tracks are available the system may operate very slowly. It is not usually necessary to support a large number of files. If the customer has unconventional applications, please communicate with us in advance.

1).For example, select the first song played, serial transmission section: 7E FF 06 03 00 00 01 FF E6 EF
7E --- START Byte
FF --- Version Information
06 --- Data length (not including checksum)
03 --- Command code (Specify Track).
00 --- Acknowledgment not required.

00 --- Track number high byte [DH]
01 --- Track number low byte [DL]
FF --- Checksum high byte
E6 --- Checksum low byte
EF --- End Byte

(The command descriptions and the message examples use hexadecimal notation for clarity. The commands as sent use single-byte (8-bit) or double-byte (16-bit) binary values.)

3.4.2. Commands to Specify Volume

The system power-on default volume is 30. If you want to set the volume, then send the corresponding commands promptly after initialization is complete... For example, to specify the volume as 15, use the serial port to send the command 7E FF 06 06 00 00 0F FF D5 EF (DH = 0x00; DL = 0x0F). 15 is hexadecimal 0x000F – refer to the instructions of playing a track by track number.

3.4.3. Specify Device Play

The module supports four types of playback devices. The device must be on line in order to be specified as the playback device. The software will automatically detect the on-line devices without user attention. The table below lists the source device selection commands.

The module will automatically enter the Suspend state after the specified device is selected, waiting for the user to specify a track for playing. It will take about 200ms from specifying device to the module initialize file information. Please wait for 200ms and then send the specified track command.

Device for playback	Command	Comment
Specify playback device –U-disk	7E FF 06 09 00 00 01 xx xx EF	xx xx : Checksum
Specify playback device –TF Card	7E FF 06 09 00 00 02 xx xx EF	
Specify playback device – Auxilliary input	*7E FF 06 09 00 00 03 xx xx EF*	*(??Device numbers do not correspond to command table)*
Specify playback device –Flash Memory	*7E FF 06 09 00 00 03 xx xx EF*	
Specify playback device -SLEEP	7E FF 06 09 00 00 05 xx xx EF	IE, No Device

Table 3.4.3 Playback Device Commands

3.4.4. Specify Track to Play – 0x0F command

Example	Command
Specify folder ‘01’, file ‘001.mp3’	7E FF 06 0F 00 01 01 FE EA EF
Specify folder ‘11’, file ‘100.mp3’	7E FF 06 0F 00 0B 64 FE 7D EF
Specify folder ‘99’, file ‘255.mp3’	7E FF 06 0F 00 63 FF FD 8A EF

Table 3.4.4 Folder/Track Selection Examples

Specifying the folder and file (Command 0x0F) is a functional extension developed to support use of the module in ‘Multiple Folder’ (Named File) mode. See 1.4 above.

If Folders are named as ‘01’, ‘02’ … ‘99’ and if files are named ‘001.mp3’, ‘002.mp3’ … ‘255.mp3’ the folder and file to be played can be nominated in a single command.

This folder/file organization supports folders named from ‘01’ to ‘99’ and files named from ‘001.mp3’ to ‘255.mp3’.

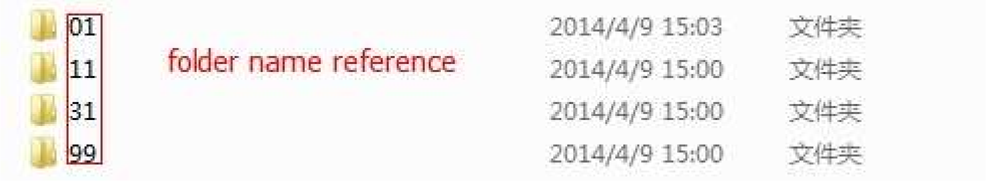
The folder number is Param1 and the file number is Param2. For example, the command to specify folder ‘01’ and file ‘100.MP3’ is 7E FF 06 0F 00 01 64 FE 87 EF. Param1 (DH) represents the name of the folder as a number (1) and Param2 (DL) represents the number of the file in the folder (100, shown as 0x64 in the example).

The 0x0F command requires both the folder and the file name to exactly identify the file. The limit of 255 files per folder exists because the file number is included in the command as a single 8-bit byte.

If the command parameters cannot be matched to a folder or a file then the command is ignored (or Error 3 may be returned). Files and folders will only be identified correctly by name if the exact format described above is followed, and folders and files are named using the standard ASCII character set.

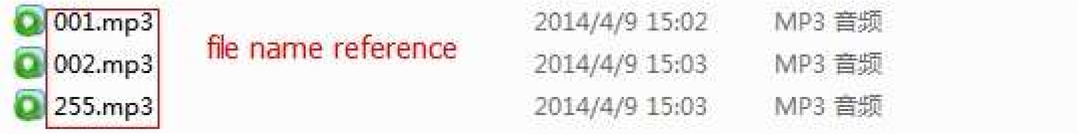
The alternative to using the 0x0F command in ‘Multiple Folder’ mode is to use the 0x03 command in ‘Single Folder’ (MP3) mode, which allows for up to a theoretical maximum of 65535, or a practical maximum of about 3,000 files in the ‘MP3’ folder. But note that in this case the file number is not converted into a file name but instead refers to the position of the file in the index of the directory file manager (FAT or FAT32). The file name used for the 0x03 command will only correspond to the file name if the files have been copied into the folder in the exact sequence of their names.

The following diagram illustrates how both the folder and file names are specified when the naming option (0x0F) is used.



01	2014/4/9 15:03	文件夹
11	2014/4/9 15:00	文件夹
31	2014/4/9 15:00	文件夹
99	2014/4/9 15:00	文件夹

Figure 3.1folder name



001.mp3	2014/4/9 15:02	MP3 音频
002.mp3	2014/4/9 15:03	MP3 音频
255.mp3	2014/4/9 15:03	MP3 音频

Figure 3.2 file name

Note: Folder and file naming starts at 0.

The 0x0F command is the only command that refers to tracks by folder number and filename. All other commands and queries use track numbers derived from the physical sequence of files as listed in the index of the file system.

‘Next’ and ‘Previous’ will always select the track to play based on the sequence in the file system index, but in ‘Multi Folder’ mode the index of the current track is generally not known, so the track selection may not be as expected.

It is not recommended to use the 0x0F command if the folder and file format naming is not exactly as described above. The command will still find and play a file, if possible, but it may not be the file that was expected.

Note that there is no command to return the number of files in a specified folder. The only available command for the number of files in a folder (0x4E) returns the value for the folder of the current track. This means that a file must be played in order to select a specific folder for the purpose of finding the number of files in that folder. A suitable procedure for managing files and folders is:

- Create a file of a few seconds silence in every folder.
- Name that file consistently (eg, 000.MP3).
- Play that file using the 0x0F command in order to set the default folder to a specific number.
- While the silent file is playing, use the 0x4E command to get the number of files in the folder.

3.5. Key Ports

The AD module inputs can be used to enable control of the module playback by keys, but it does not follow the usual matrix keyboard arrangement. In order to take advantage of increasingly powerful MCU AD functionality, the module can use a voltage from a resistor ladder controlled by key switches to generate multiple function commands from just two AD inputs.

Note: If the module is operated in the vicinity of strong electromagnetic interference or strong inductive or capacitive load additional precautions may be needed. See Note (2) below.

Resistance Ladder Diagram

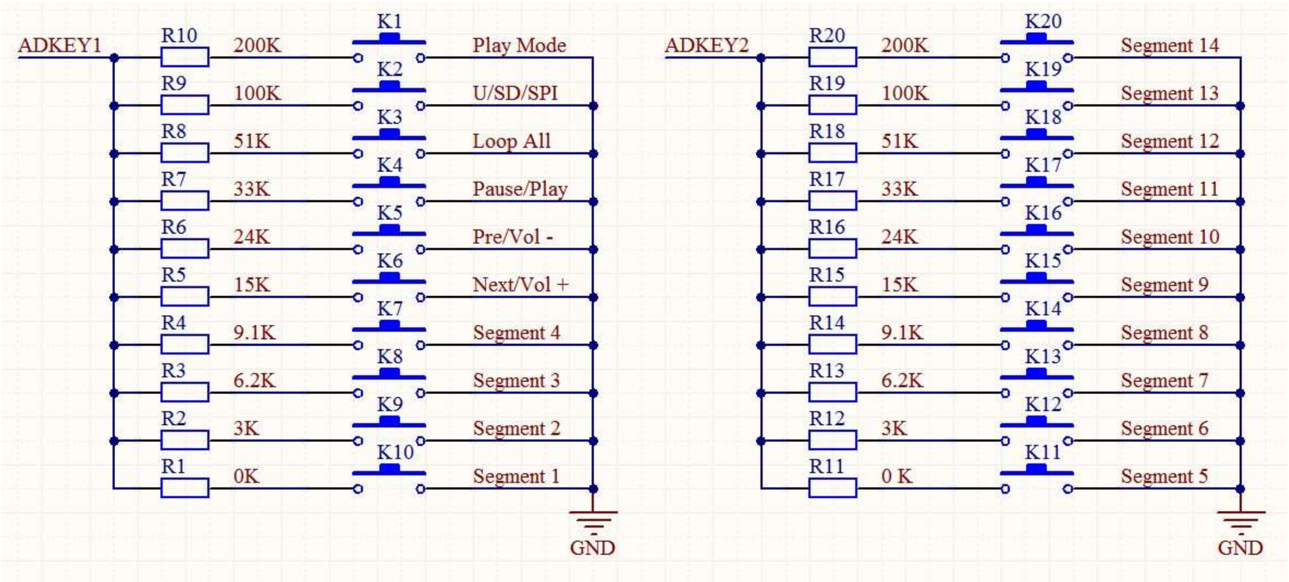


Figure 3.5 AD Key Command Reference

20 function keys allocation table

Key	Short Push	Long Push	Description
K1	Play Mode		Switch to interrupt / non interrupted
K2	Playback device switches		U/TF/SPI/Sleep
K3	Operating Mode		All cycle
K4	Play/Pause		
K5	Previous	Vol+	
K6	Next	Vol-	
K7	4	Repeat play tracking 4	Long push always to repeat play
K8	3	Repeat play tracking 3	Long push always to repeat play
K9	2	Repeat play tracking 2	Long push always to repeat play
K10	1	Repeat play tracking 1	Long push always to repeat play
K11	5	Repeat play tracking 5	Long push always to repeat play
K12	6	Repeat play tracking 6	Long push always to repeat play
K13	7	Repeat play tracking 7	Long push always to repeat play
K14	8	Repeat play tracking 8	Long push always to repeat play

K15	9	Repeat play tracking 9	Long push always to repeat play
K16	10	Repeat play tracking 10	Long push always to repeat play
K17	11	Repeat play tracking 11	Long push always to repeat play
K18	12	Repeat play tracking 12	Long push always to repeat play
K19	13	Repeat play tracking 13	Long push always to repeat play
K20	14	Repeat play tracking 14	Long push always to repeat play

Table 3.5.1 Function Keys Allocation

4. Application Circuit

4.1. Serial Communication Connect

Module's serial port is 3.3V TTL level, so the default interface level is 3.3V. If the MCU system is 5V it is recommended to connect a 1K resistor in series with the MCU Tx (or a 332k/22k voltage divider, or a level converter). See Note (3) below. The MCU receive should be able to correctly interpret the 3.3v levels provided by the module.

Note: A Logic level converter designed for I2C may not work correctly with the module. These converters are designed for a floating GND, while the module requires a common ground. A voltage divider is the preferred method of level conversion.

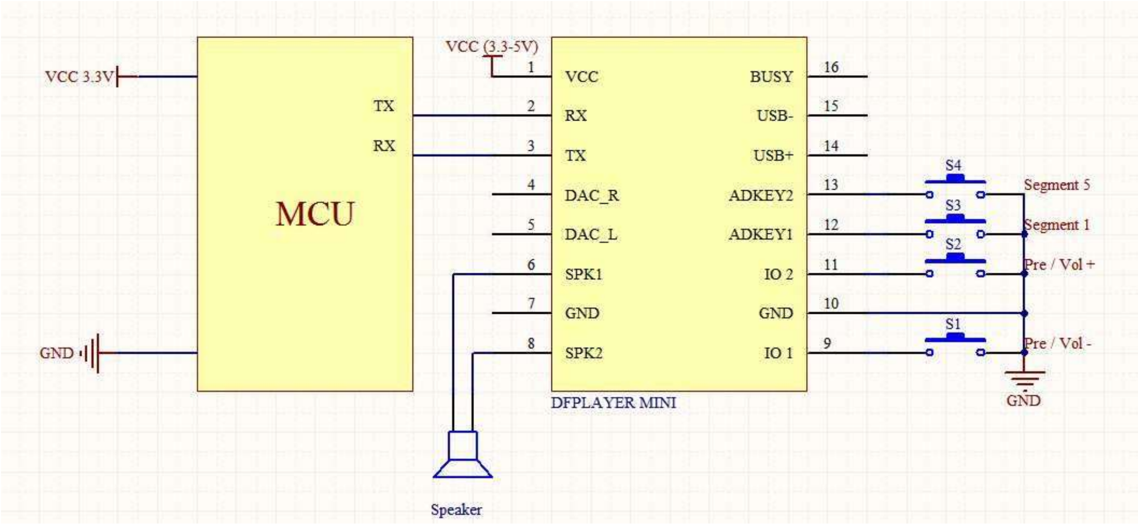


Figure 4.1 Serial Connect (3.3V)

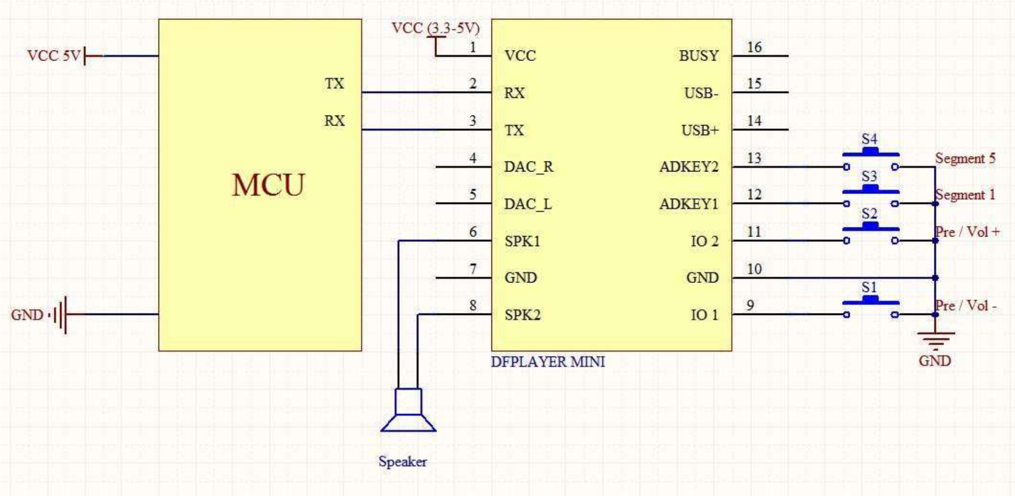


Figure 4.2 Serial Connect (5v)

Note: Power supply arrangement is not shown. If power is supplied from a regulator on the MCU or similar low-power device then high volume operation with a speaker may overload the supply.

4.2. Other Reference Diagrams

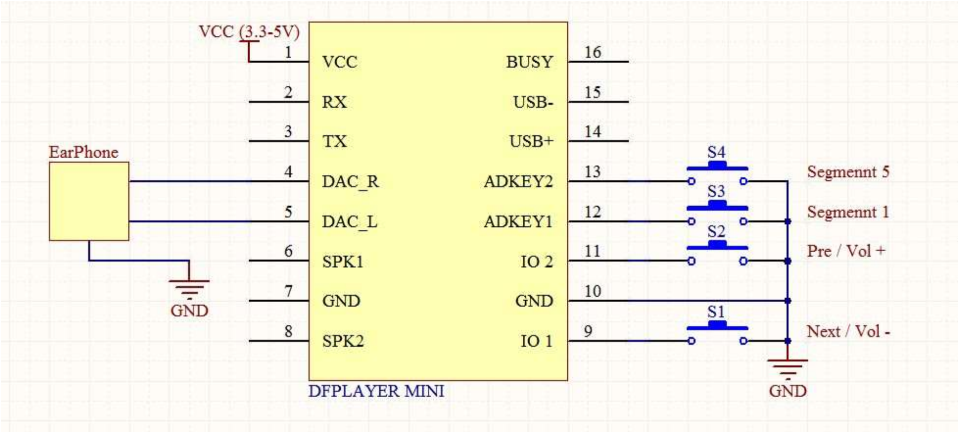


Figure 4.3 headset connect module

A high impedance headset is recommended in order to minimize noise. A 100R resistor can be used in series with the headphone or earpiece to increase the impedance and allow playback at a higher volume, which also reduces noise.

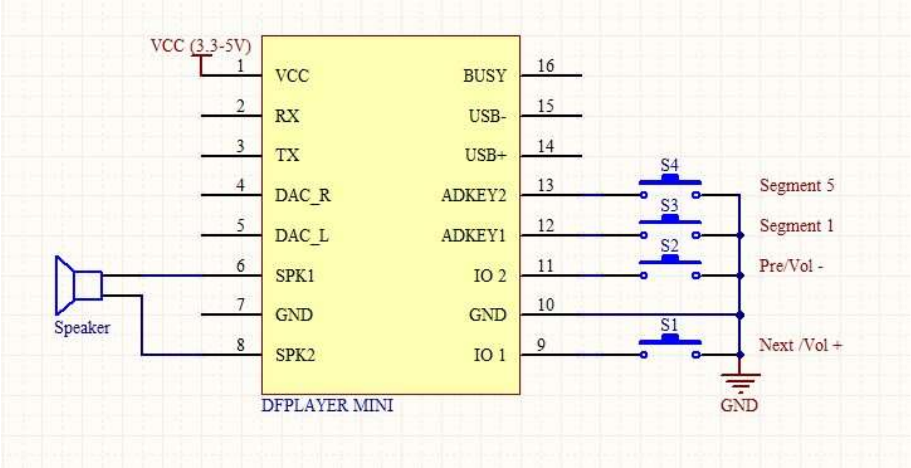


Figure 4.4 speaker connect module

Note: Speaker output is mono only. Speaker connection is two-wire balanced pair. Maximum load is 3W. Speaker should not be connected to Gnd.

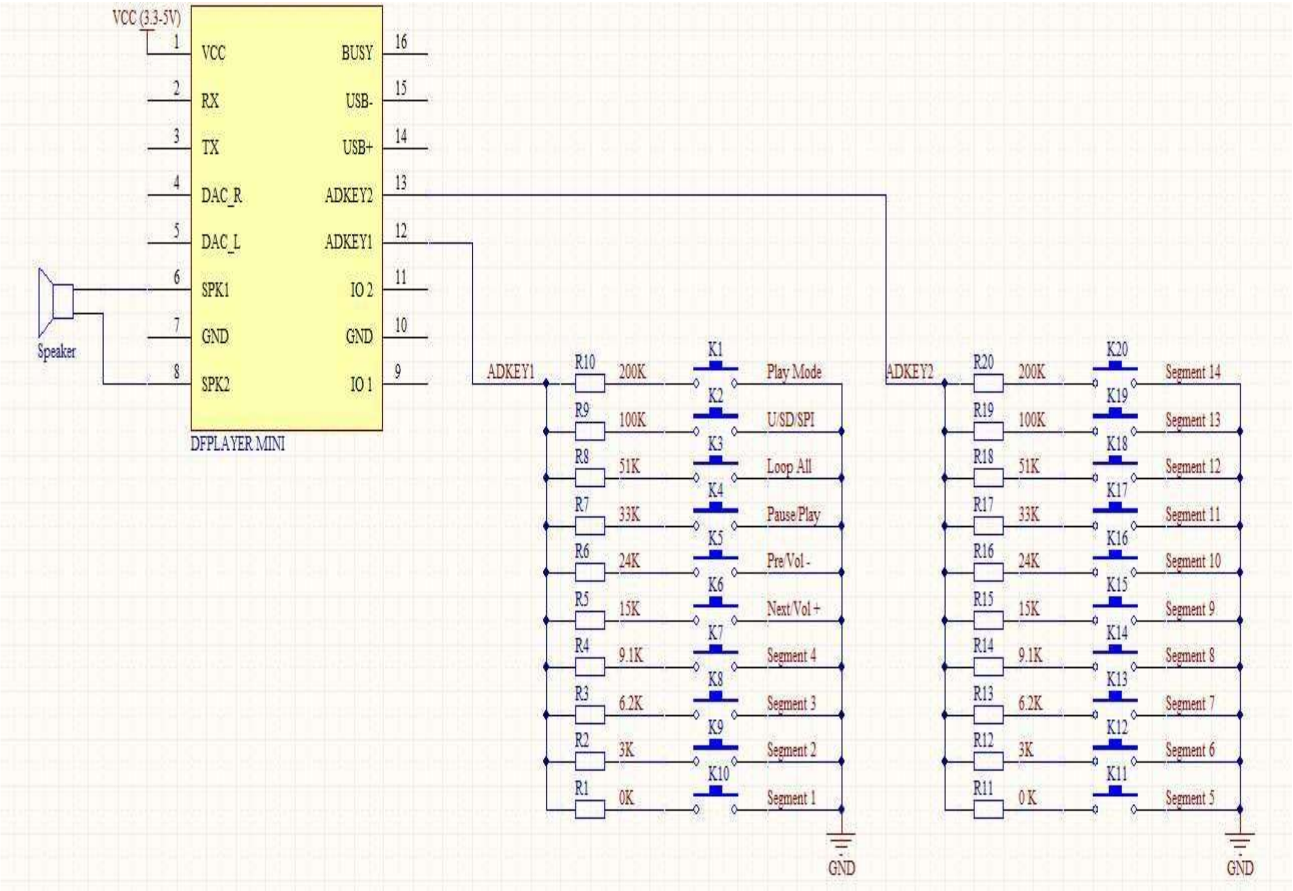


Figure 4.5 Ad key connect refer

5.MP3-TF-16P Size (unit: mm)

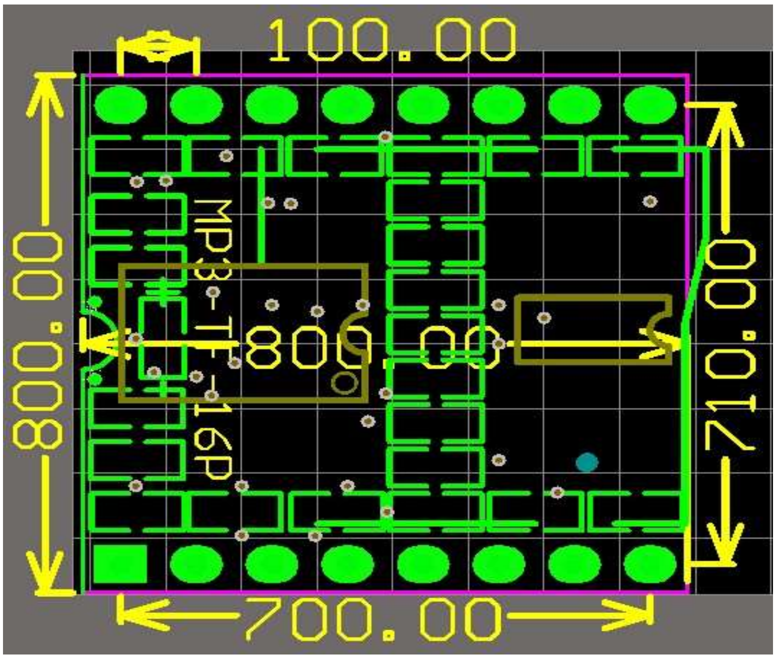


Figure 5.1 pcb size

6. Specifications

I/O Input Specification						
Item	Description	Min	Type	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 Specification						
Item	Description	Min	Type	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

Table 6 Input/Output Specification.

1. The module's external interfaces are 3.3V TTL level, so please note the level conversion during the hardware circuit design. For use in an environment in which there is strong electromagnetic interference protective measures such as GPIO opto-isolation may be required.

2. ADKEY key values are in accordance with usage in a typical environment. If there are strong inductive or capacitive loads then a separate isolated power supply should be used together with inductors for power filtering. Input power should be as stable and clean as possible. If a clean and stable supply cannot be provided operation can be improved by increasing the voltage range assigned to each key (and reducing the number of keys available). If this option is required please contact your device supplier for details.

3. For general Serial communication, please pay attention to level conversion. If operating in an electrically noisy environment or over a long communication link such as RS485 then please note that signal isolation, in strict accordance with industry standard design for communication circuits, may be required.

General Notes for constructors.

The module is very sensitive to poor connections. Wiring from the power supply and the MCU should be kept as short as possible. Any prototyping arrangement must use very secure connections.

There does not appear to be any way to adjust the device UART baud rate from the default of 9600 baud.

Timing delays are required for some commands. Command and queries should not be overlapped: the acknowledge option should be used, and the acknowledgement confirmed before issuing an additional command or query.

The most reliable mode of operation is a single folder of up to about 3,000 MP3 files. The reason this is most reliable is that the command to get the number if files in a folder is erratic – sometimes it does not return until the current track is completed. This information is required when using named mode in order to ensure that an invalid track is not selected. If there is some other way to ensure this (for instance, by creating exactly 255 tracks in every folder) then the code that finds the number of tracks per folder can be disabled and this mode is completely reliable.

Despite this supposed unreliability, the prototype has been run continuously for more than 48 hours in named mode with random selection form all folders without any errors.

GDB3200B MP3Player Module

There can be considerable background noise between tracks. The cause is not clear but might be due to UART activity.

The investigation on which much of the information in this document is based was done with the following setup.

- Generic MF2024 and GD3200B modules, Chinese clones, unknown manufacturer. As far as could be determined all modules from a sample of 6 behaved the same.

- Arduino ProMini 3.3v 8MHz and 5v 16MHz, Chinese clones, unknown manufacturer.

- Generic 32Gb TF card. 19 folder, ~4500 files (Named mode), ~3000 files (MP3-Folder mode).

- Arduino power: Regulated 5v from a USB supply to the Raw input, OR
  Regulated 5v applied to Vcc and Gnd
- Module power; regulated 3.3v/5v from the Pro Mini.

- High-impedance amplifier fed from the headphone terminals.

With the 5v MCU, a 2.2k/3.3k voltage divider was used between the Pro Mini serial out and the MP3 module serial in. This provided a measured signal level of 3.0v for high and 300mv for low at the module pin. This arrangement considerably increased the reliability of the communication with the module, compared with the recommended arrangement of a 1K series resistor.