

Contents

<u>Subject</u>	<u>Page</u>
<b>1. Specifications of Board ET-REMOTE MP3.V2</b>	2
<b>2. Specifications &amp; How to manage File for Micro SD Card</b>	3
<b>3. Feature of Board ET-REMOTE MP3 V2</b>	6
<b>4. How to use Board ET-REMOTE MP3 V2</b>	11
<b>4.1 COMMAND MODE</b>	11
4.1.1) Operation	11
4.1.2) Format of Command & Respond for ET-REMOTE MP3 V2	11
4.1.3) Application	21
<b>4.2 LIST_SONG MODE</b>	23
4.2.1) Operation	23
4.2.2) Application	24
<b>4.3 MP3_SW MODE</b>	25
4.3.1) Operation	25
4.3.2) Application	26
<b>4.4 TRIG MODE</b>	27
4.4.1) Operation	27
4.4.2) Application	29
4.4.3) How to Trig by Signal Trig from external Board MP3	30
<b>5. Describe how to run example program in CD</b>	35
<b>5.1 Example Program for COMMAND_MODE</b>	35
<b>5.2 Example Program for TRIG_MODE</b>	36
<b>Circuit Board</b>	37


**ET-REMOTE MP3 V2.0****1. Specifications of Board ET-REMOTE MP3. V2**

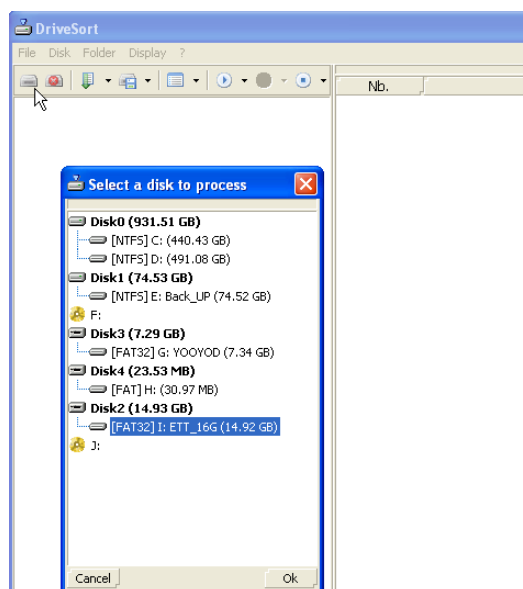
- 1.1 Only support File .MP3 or .WAV, but it cannot record sound.
- 1.2 Use Micro SD Card with capacity of 2GB-16GB to store sound file and support FLASH 16 and 32.
- 1.3 Provide +5VDC Power Supply for board and produce signal sound through Stereo Output.
- 1.4 There are 4 operation modes as follows;
  - **COMMON MODE**: Playing song is controlled by Commands through RS232.
  - **LIST\_SONG MODE**: See a list of song and order of sound file in Micro SD Card.
  - **MP3\_SW MODE**: Playing song is controlled by 3 switches on board.
  - **TRIG MODE**: Playing song is controlled by Trig at Pin that receives Signal Trig; in this case, it can trig 16 orders of song (see more information of Trig in the section of TRIG MODE).
- 1.5 Can play sound file from internal and external folder.
- 1.6 Can see a list of song and order of file in Micro SD Card by using DIP-SW\_MODE or sending Command. It shows result through RS232 by Program Terminal or other programs that support the RS232 Interface.
- 1.7 LED displays status of playing song, triggering (TRIG) and ending file (EOF).
- 1.8 Every time it finishes playing any file, it sends Command End of File through RS232 and it sends 20ms Pulse Width (Logic '0') to Connector EOF of board; so, user can check the end of file.
- 1.9 4 Baud Rates can be set in COMMAND Mode; 4800, 9600, 57600, and 115200 Baud/sec.
- 1.10 This COMMAND Mode can play song according to the number order (not longer than 99 orders), or play by name; stop playing song by either STOP or PAUSE, tune the volume up/down; repeat 1 track, or repeat all, or play 1 track and stop, or play all tracks one time and stop; play forward (FW) or rewind (RV) 1 song in each time; show Play Time, name and order of file in Micro SD through RS232.
- 1.11 **COMMAND Mode**: When it sends Command Play in each time, user can set the format of playing; it plays the next song instantly after sent the Command Play, or it waits for a minute to end the current song first and then starts playing the new song.
- 1.12 **MP3\_SW. MODE**: It can use Commands; Play, STOP, PAUSE, FW, RW, tune the volume up/down by 3 switches on board.
- 1.13 **MP3\_SW. MODE**: It can use DIP-SW\_CONF to set the format of playing song; it plays all songs one time and then stops; or it repeats all songs. Moreover, user can choose the format of playing song either to play the song instantly after Power ON (Auto), or play the song when pressing SW-Play on board.
- 1.14 **TRIG MODE**: It can stop (STOP) and tune the Volume up/down by 3 switches (SW.) on board.
- 1.15 **TRIG MODE**: It can play song from 1 to 16 by using Pin that receives Signal TRIG of the song position that user requires playing; it trigs into ground and then set it free, it starts playing the song instantly.
- 1.16 **TRIG MODE**: It uses DIP-SW\_CONF to set formats of playing song; it plays 1 song and stops, or it repeats 1 song. Moreover, it can be set to start playing the song instantly after triggered, or it waits for a while until the current song finishes first and then starts playing the triggered song later.
- 1.17 Every time board is reset, the Volume is set to be Default value; moreover, it does not remember any pending operation before resetting.
- 1.18 Adjustment of DIP-SW\_MODE and DIP-SW\_CONF only is effective when Board is reset.

## 2. Specifications and How to manage File for Micro SD Card

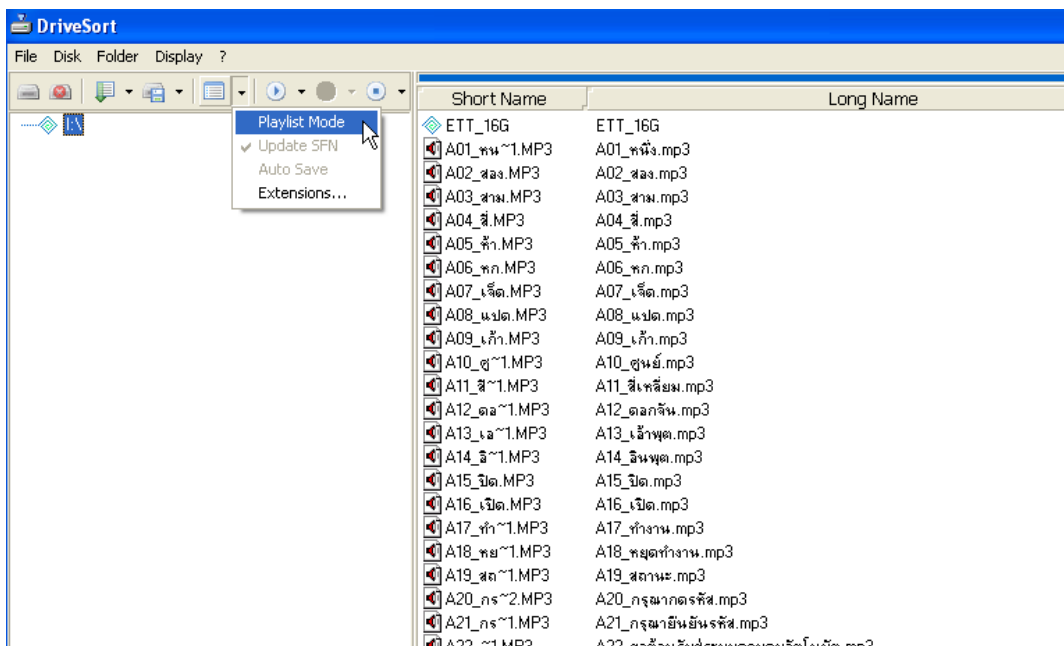
- 2.1 Setup name of file or folder by either Thai or English and it can be set by either capital letter or small letter as preferred. However, it does not allow to use the same letters as the Command such as \$ \* # @ & =
- 2.2 File surname of song must be .MP3 or .WAV only; in this case, user can set it by either capital letters or small letters as well.
- 2.3 When user requires calling any file name and file surname, it always shows in English capital letters. If any file name is setup in Thai letters, it shows strange and illegible letters instead. In this case, it only displays 8 letters at maximum (if any file name is longer than 8 letters according to the format 8.3) through RS232; it shows result by Program Terminal or other programs that support the RS232 Interface.
- 2.4 If using **Command Play by name**, it always uses capital letters for playing file name and file surname, although the file name-file surname is setup by capital letters or small letters.
- 2.5 It has to format Micro SD Card that is used to store track into FAT16 or FAT32 first.
- 2.6 When user copies 1 song file from PC into Micro SD Card in each time, the first song file that is copied and pasted in the Card will be played first. The next song file will be copied and pasted in the order of 2,3,4,..., and so on, this file order is used to send Command Play respectively.
- 2.7 When user copies all song files (Copy All) from PC into Micro SD Card in each time, user does not exactly know which file will be copied and pasted in the Card first, or which file is pasted in the order of 2,3,4,..., and so on. When using Command Play to play song files respectively, user does not know that it is playing the preferable song file or not. In this case, user can use **DIP-SW\_Mode** or Command **List File** to see the file order and file name-file surname of all song files in Card through RS232 first because user can play the song file in the preferable order. Or, it uses Program **Drive Sort** that is provided in CD to re-arrange all song files as required, please follows these instructions;

### How to use Program Drive Sort to rearrange files in Micro SD Card

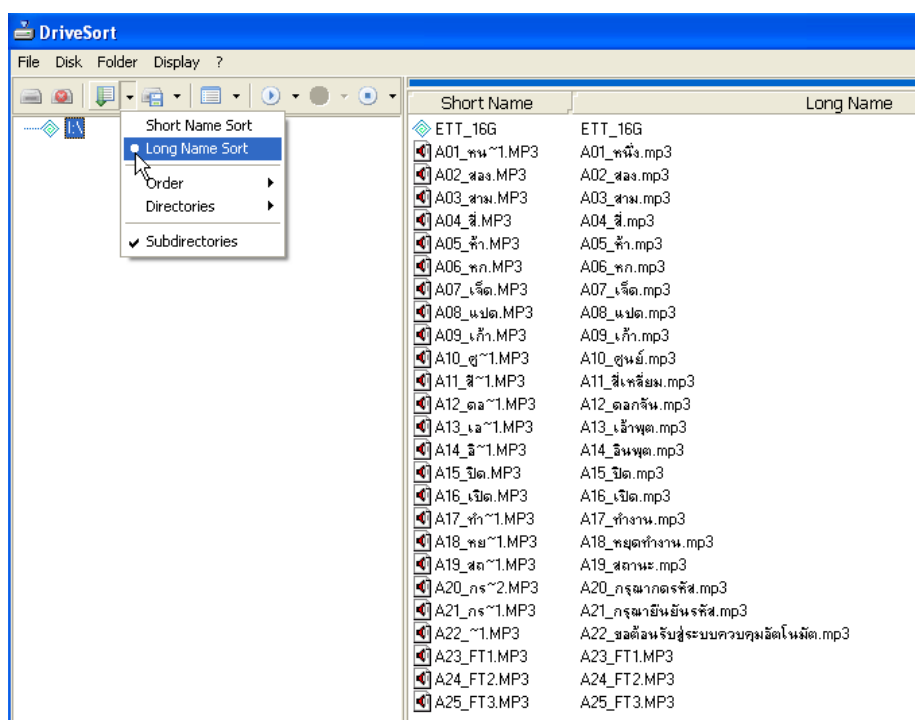
- 1) First of all, copy and paste the preferable song file into Micro SD Card completely.
- 2) Run Program **DriveShort.exe**; click Icon 'Open a disk' []; and the window 'Select a disk To Process' appears as shown in the picture below;




- Double click the Drive that is connected with Micro SD Card and it shows a list of song file on the right window. Click the black arrow on the right side of Icon 'Playlist' [ ] to remove the tick mark in front of 'Playlist Mode', it shows all song files in the Card in the window on the right side as shown in the picture below. It shows the full file names in the blank of Long Name, and it shows the short file name in the blank of Short Name.



- Click the black arrow on the right side of Icon 'Short current folder' [ ]; next, choose 'Long Name Sort' to arrange files according to the name in the blank 'Long Name', it is the actual file name that user can see in the Card. Next, click Icon 'Short current folder' [ ], user can see that the list of file name in the blank 'Long Name' is rearranged in the alphabetic order. The file on top is the first file and it arranges in the order respectively as shown in the picture below.

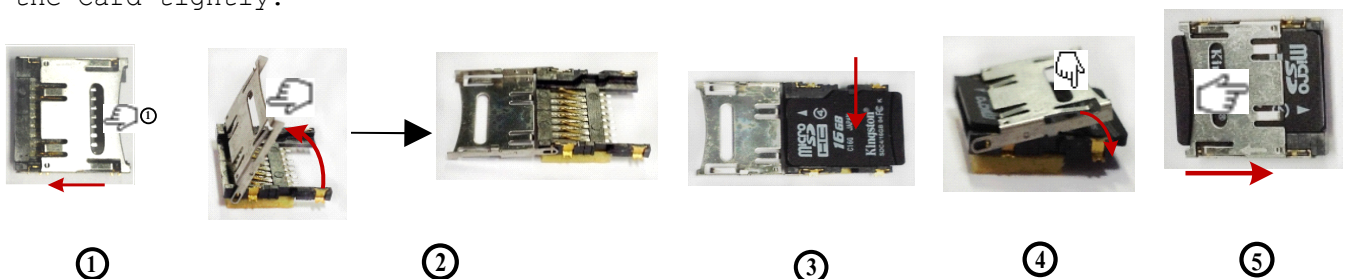


- 5) Save the new file arrangement, click Icon 'Save current folder' [  ]. It ends the process of arranging file; next, it closes the program and then inserts this Micro SD Card into Board ET-REMOTE MP3.
  - 6) The technique that is used to arrange files by Program DriveShort depends on specifying file name. Before copying file into Card, it should edit or change the file name as follows;
    - If it setup the file name by number only, it has to use the equal digit such as 01.mp3, 12.mp3, 99.mp3. When programming, it arranges the file name 01.mp3 to be the first order and then so on. Remember, it does not setup the file name as 1.mp3, it has to add a naught ('0') in front; or, if it is 3 digits, it has to add 2 naught ('00').
    - If it setup the file name by letters, every files should be preceded by English letter that has the equal digit; then, it is followed by number with the equal digit; and finally, it is free to add any letter (if have) after the number such as FM01.mp3, FM02K.MP3, FM03.mp3,...FM45.mp3. When programming, it arranges files from FM01-FM45 and then so on, respectively.
- 2.8 After Power ON or Reset, user can play the song file in MP3\_SW Mode or COMMAND Mode by Command 'Play All'; the first song file in Micro SD Card is always played first and then followed by the next order of 2,3,4,..., respectively.

### 3. Feature of Board ET-REMOTE MP3. V2

1. **LED-PWR:** This LED shows status of +5V Power Supply for board.
2. **LED-TRIG:** This LED shows status of Trig of Connector No.19 that receives Signal Trig. The LED is lit up according to the incoming Signal Trig (Trig Active '0') when using in TRIG MODE.
3. **LED-ACT:** This LED shows status while playing song files; in this case, LED blinks while the song file is playing.
4. **SOCKET-Micro SD Card:** This Socket is used to insert Micro SD Card that stores song files. Remember, before inserting or removing this Card, it always disconnects the Power Supply of Board MP3 first. The method to insert the Micro SD Card is described below;

- 1) Push the Socket cover to the left side.
- 2) Pull the Socket cover.
- 3) Insert Micro-SD Card.
- 4) Close the Socket cover.
- 5) Press the Socket and push the Socket Cover to the right side to lock the Card tightly.



**Warning:** Every time user inserts or removes this Micro SD Card from Board MP3, it has to disconnect Power Supply from Board MP3 first.

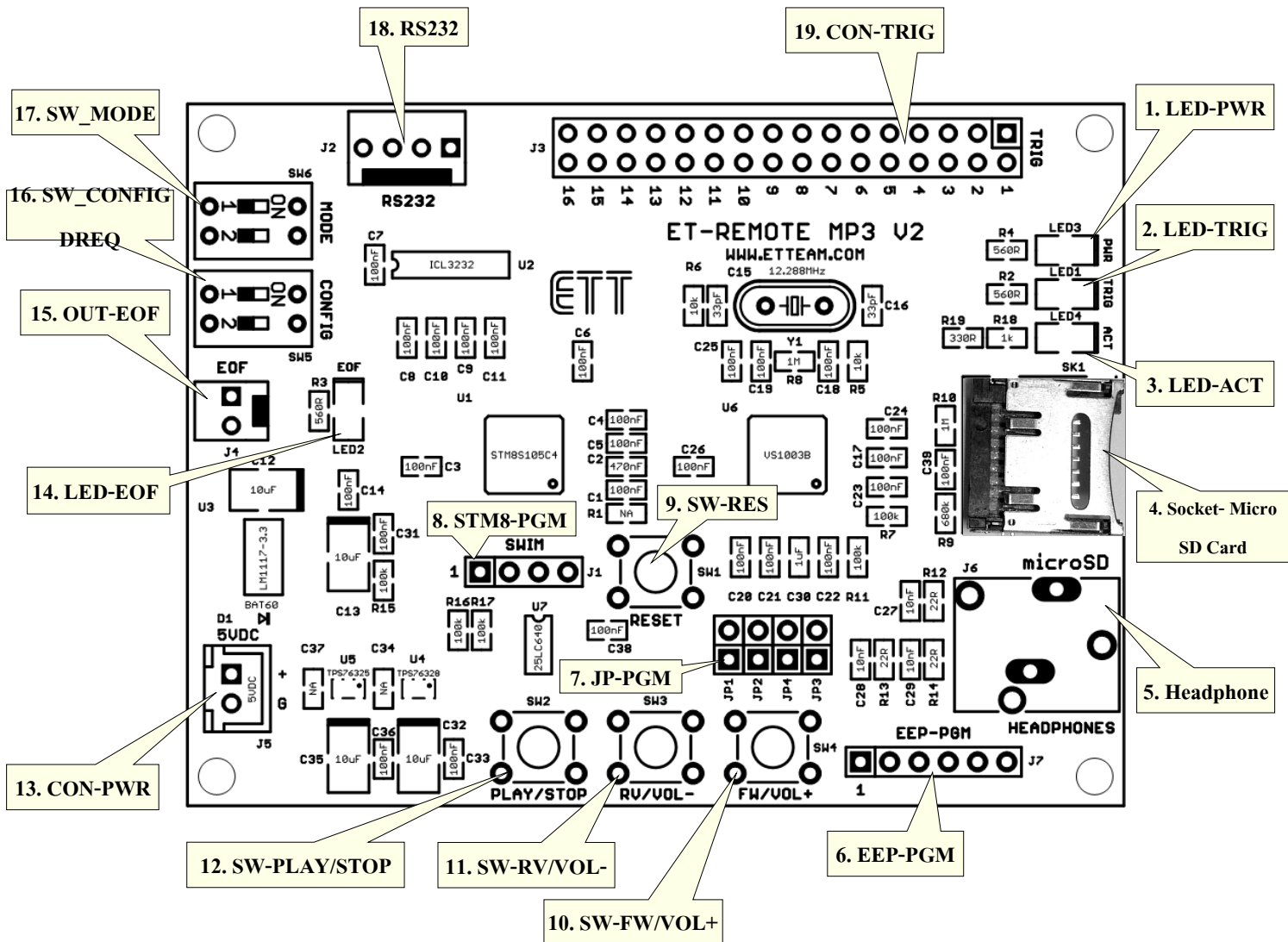


Figure shows the structure of Board ET-REMOTE MP3.V2.

5. **Head-Phone:** It is connected point for Stereo Output. If it is headphones, it can directly interface with board. If it is Speaker, it has to interface through the amplifier; it cannot be connected directly because it makes the Chip MP3 damaged.
6. **EEP-PGM:** This Connector downloads Firmware into E2PROM; it is provided for the company to Upload Firmware only, it is not related to use board of user.

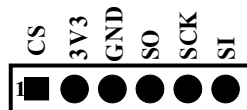


Figure shows feature of Pin for Connector EEP-PGM.

7. **JP-PGM:** When user requires loading Firmware into EEPROM, it has to remove Jumper1-4 from board first. After loaded the Firmware completely, it has to interface (Jump) all Jumper 1-4 again and Board MP3 can run as normal. This Jumper is not related to user directly, it is provided for the use of company. So, user has to ensure and check if this Jumper 1-4 has been interfaced (jump) completely when using this Board MP3, otherwise this Board MP3 does not run.

8. **STM8-PGM:** This Connector downloads Firmware into STM8. It is provided for company to Upload Firmware, it is not related to use board of user.

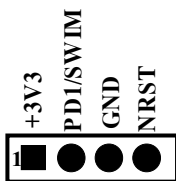


Figure shows feature of Pin for Connector STM8-PGM.

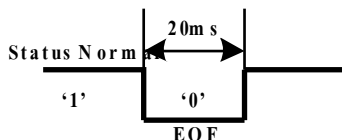
9. **SW-RES:** This SW.Reset is used to restart board when it hang-up, or update the change of DIP-SW\_MODE or DIP-SW\_CONFIG.
10. **SW-FW/VOL+:** There are 2 functions when using this Switch in **MP3\_SW MODE**. Firstly, it is used to move 1 song file forwards in each time by pressing and removing SW as normal. Secondly, it is used to tune the volume up by pressing and holding SW. for 0.5 second and then removing it. When using this Switch in **Trig Mode**, it is only used to tune the volume up; in this case, it only presses and removes the switch as normal.
11. **SW-RV/VOL-:** There are 2 functions when using this Switch in **MP3\_SW MODE**. Firstly, it is used to move 1 song file backwards in each time by pressing and removing SW as normal. Secondly, it is used to tune the volume down by pressing and holding SW. longer than 0.5 second and then removing it. When using this Switch in **Trig Mode**, it is only used to tune the volume down; in this case, it only presses and removes the switch as normal.
12. **SW-PLAY/STOP:** There are 3 functions when using this Switch in **MP3\_SW MODE** as follows;
- 1) **PLAY:** It plays the song file. When user presses and removes the switch, song file that has not been played yet starts playing instantly.
  - 2) **STOP:** It stops playing the song file that is being played; when user presses and removes the switch, it stops playing the song file instantly. When user presses and removes this switch again, it restarts playing the first song file.
  - 3) **PAUSE:** It temporarily stops playing the current song file (Pause). While the current song file is being played, user presses and holds the switch longer than 0.5 seconds and then removes, it temporarily stops playing the current song file. When user presses and removes this switch again, it starts playing the song file that is previously paused again. When using this Switch in **Trig Mode**, it is only used to stop (STOP) playing the current song file; in this case, it only presses and removes the switch as normal.
13. **CON-PWR:** This connector is used for Power Supply DC 5V of Board MP3; in this case, user has to ensure that the point of electric connection is interfaced correctly.



Figure displays Connector Power Supply of Board.

14. **LED-EOF:** This LED shows status of Signal End of File (EOF). Every time it finishes playing the song file or it is commanded by the Command Forward or Reverse, Signal EOF happens for a while and this LED is ON and OFF according to the signal happened.

15. **OUT-EOF:** This Connector Output is used to send Signal EOF to check the end of the current file. This signal happens when it finishes playing the File or it is commanded by the Command Forward or Reverse. When it is in the normal status, the feature of signal at Pin EOF is Logic 1 (3.3V); when Signal EOF happens, signal at Pin EOF falls to Logic 0 for 20 ms and then it returns to Logic 1 (3.3V) again. In summary, every time it finishes playing the song file, Board MP3 sends Signal Pulse EOF Logic 0 with 20ms Width to user through this connector.



Feature of Signal EOF



Feature of Connector EOF

16. **SW\_CONFIG:** This DIP SW. is used to set Baud Rate for receiving-sending command through RS232 when using Board MP3 in COMMAND MODE. This DIP SW. is used to set specifications of playing song file when using Board MP3 in MP3\_SW MODE of TRIG MODE. The change of DIP SW. in each time is only effective when Board is reset. So, every time user sets DIP SW., user has to reset Board MP3 to update the new change. For table of Set DIP-SW CONFIG, user can read more information in the Application of each operation mode.
17. **SW\_MODE:** This DIP SW. is used to set and choose the operation mode. When using requires using this board, it has to set DIP-SW\_MODE to set and choose the operation mode first. There are 4 operation modes as shown in the table below. After Set DIP-SW\_MODE completely, it always resets Board MP3 to update the new change, otherwise this Set DIP-SW\_MODE is ineffective and it has not an effect on the operation of board.

Table: Show Set DIP SW\_MODE

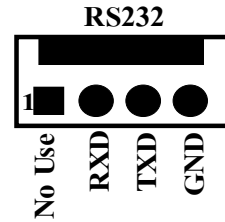
DIP-SW_MODE		Format of SET SW_MODE	Operation MODE
S1	S2		
ON (0)	ON (0)		It runs in <b>COMMAND MODE</b> . It sends command through RS232 and it uses DIP-SW_CONFIG to set Baud Rate for sending-receiving.
OFF (1)	ON (0)		It runs in <b>LIST_SONG MODE</b> . It sees all lists of song file and song order in Micro SD Card through RS232. It uses Program Terminal to display results.
ON (0)	OFF (1)		It runs in <b>MP3_SW MODE</b> . It controls to play file by 3 switches on board and it uses DIP-SW_CONFIG to set formats of playing song file.
OFF (1)	OFF (1)		It runs in <b>TRIG MODE</b> . It plays the song file by receiving Signal Trig from external and it uses DIP-SW_CONFIG to set formats of playing the song file.

18. **RS232:** It is Connector to interface Cable RS232. It provides IC Line Driver 232 on board completely; so, it can interface with PORT 232 of PC directly. If it is interfaced with MCU, it has to provide IC Line Driver 232 for PORT 232 of MCU on the other side and it can communicate without any error. This connector is used when user requires seeing a list of song file and file



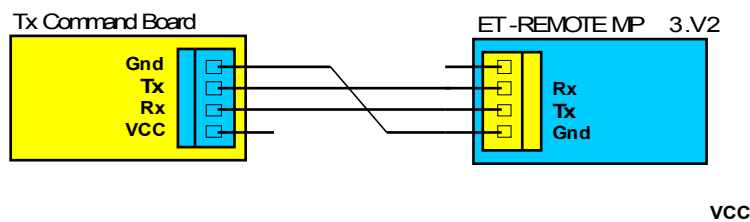
order in Micro SD Card and it uses Program Terminal to display result; or, it is used to send Command Play in Command Mode by PC or MCU. It has to set specification of communication as follows;

BAUD RATE : 4800 , 9600 , 5700 , 115200 by DIF-SW\_CONFIG  
 DATA BIT : 8  
 PARITY : NONE  
 STOP BIT : 1  
 HANDSHAK : NONE



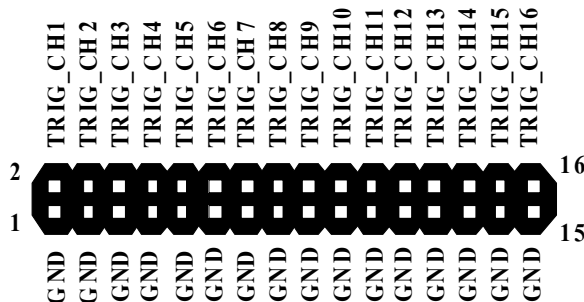
It shows Connector RS232.

It interfaces Cable RS232 as shown in the picture below;



It shows how to interface Cable RS232 from device that sends Command to Board ET-REMOTE MP3.V2.

**19. CON-TRIG:** This 16-Channel connector is used to receive Signal TRIG from external and it is used in TRIG MODE. The inner Pin of board is set Pull Up at 3.3V; it is Function of 16-CH Input to receive Signal Trig from external or on board. The outer Pin of board is all GND that is used to test Signal TRIG. When any couple of pins that are matching are shorted and removed, Signal Trig happens; then, the order of song file that accords with channel of Signal Trig starts playing instantly. This Connector receives Signal Trig Logic '0' (Active '0') with 20ms Width or higher. When the status of Signal Trig is Logic '1', the voltage level of the signal does not exceed 3.3V because this Connector Input only supports the voltage level that is not higher than 3.3V. If Signal Trig that is in the status of Logic '1' is higher than 3.3V such as 5V, it has to interface the circuit that reduces the voltage level to 3.3V at the center first; in this case, it may use the ETT product such as ET-MINI 3To5TTL (8-Channel connectible) or ET-BUSIO-DCIN (1-CH connectible), please see more information in section of TRIG MODE.



It shows Connector to receive Signal TRIG.

**Warning:** Signal that is used for Trig in the status of Logic '1', the voltage level does not exceed 3.3V only.

## 4. How to use Board ET-REMOTE MP3 V2

### 4.1 COMMAND MODE

#### 4.1.1) Operation

This operation mode sends external Command through RS232 to control to play song file. In this case, it can use any series of Controller to send command, or it can send Command through Program Terminals by typing commands from Keyboard.

When board starts running, it types and shows the message '**ET-REMOTE MP3 V2 (Firmware V2.1) = #OK**' through RS232; it notifies user to know that it is ready to receive Command. The board that is used to send Command can read this Respond and use this Respond to check if Board MP3 is ready before sending Command. Or, it may set Time Delay for 2-3 seconds because Board ET-REMOTE MP3 can initial itself completely before receiving Command.

When user sends Command into board and the Command is done completely, the board sends out Echo in the format of **#OK (3Byte)** if the Command that is sent out is correct. If Command Play File is sent out before file is played in each file, board always sends order of file and file name-file surname that will be played in the format of **\$NN.FFFF.MP3 (or .WAV) when NN = order of file (2Byte), FFFF = Filename (not higher than 8 Byte)**. Next, when it finishes playing the current song file; or, the current song file is interrupted by Command STOP, Forward, or Reverse and the current song file also ends up, Board sends Signal EOF in the format of **&E (2 Byte)** to notify user to know that it finishes playing the current song file and it is ready to receive the next command or play the next song file. Signal EOF can be sent out through RS232; moreover, it can send Pulse 0 with 20ms Width to Connector No.15(OUT-EOF); in this case, user can see that Signal EOF happens from LED-OEF on board. If user sends command to show Play Time while playing the song file, the board also sends the Play Time throughout in the format **@mm:ss when mm = minute(2Byte), ss = second (2Byte)**.

*Remember:* When user starts sending Command into Board MP3, it should set both parts of receiving and sending Command synchronously (Syn). It should set Time Delay 200-500ms or higher for board that is used to send Command; next, it starts sending the first Command into board, it sets Board MP3 to be in the ready status to receive the command before Power Up. When it sends Command continuously, it should set Time Delay between Commands if it does not check any Echo Command from Board MP3; in this case, Board MP3 finishes running the first Command completely.

#### 4.1.2 Format of Command and Respond for ET-REMOTE MP3 V2

The format of Command that is used is in all ASCII Code; it can be replaced by ASCII Symbol such as '\*', or HEX Code that is 0x2A, '0'=0x30, 'A'=0x41. If it sends Command in the format of letter A-Z, it only sends the Command by the capital letters only; it does not support any small letter. User can see any file number or filename-file surname that is replaced in the Command Play (Command P1,P2) by DIP-SW\_MODE (please see in the section of LIST\_SONG MODE), or Command Call by file name. This file number, file name and file surname that is called can be used to replace in Command Play (P1,P2). The Command is described as follows;

**- 1. COMMAND 'P0' (Play File All)**

This Command plays all files in Micro SD Card. It starts playing the first song file; in this case, it is the first file that has been copied and pasted into the Card. The format of command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	P	0	=	0x0D (enter)

**Ex.**

```
char enter = 0x0D ;
printf("*P0=%c",enter) ; // ส่งคำสั่งเล่นไฟล์ทั้งหมดใน Micro SD Card
```

**- 2. COMMAND 'P1' (Play File By Number)**

This Command plays sound file by file number from 01-99; this file number must be sent in the format of 2Byte ASCII. When it plays sound file, it has to refer to the actual amount of file in the Card but is only plays 99 file at the maximum, although amount of file in the Card is higher than 99 files. The format of command is shown in the table below;

Start	CMM		Mark1	Number (2Byte)	END
Byte1	Byte2	Byte3	Byte4	Byte5-6	Byte7
*	P	1	=	01-99	0x0D (enter)

**Ex.**

```
char enter = 0x0D ;
printf("*P1=07%c",enter) ; // ส่งคำสั่งเล่นไฟล์ลำดับที่7 ใน Micro SD Card
```

**- 3. COMMAND 'P2' (Play File By Name)**

This command plays sound file by name. File name and file surname must be capital letter only; moreover, the file name must be 8 Byte at the maximum and the file surname must be either .WAV or .MP3 only. If using this Command to play song file, the file name must be set by English only. If the file name is longer than 8 Byte, it can use the Command Call by file name or DIP-SW\_MODE to see list of song file briefly. The format of Command is shown in the table below;

Start	CMM		Mark1	FILE NAME.MP3/WAV (ชื่อไม่เกิน 8Byte)	END
Byte1	Byte2	Byte3	Byte4	Byte5-16	End Byte
*	P	2	=	FFFFFFFF.MP3	0x0D (enter)

FFFFFFFF = File name is set by ASCII Code in capital letter and does not exceed 8 byte.

**Ex.**

```
char enter = 0x0D ;
printf("*P2=LINE8.MP3%c",enter) ; // ส่งคำสั่งเล่นไฟล์ชื่อ LINE8.MP3
```

#### - 4. COMMAND 'PU' (Pause Play File)

This Command is used to temporarily stop playing sound file (PAUSE). If user requires playing the sound file at the stopped position, user only sends this Command again. While using this Command Pause Play File, it still uses other commands such as to tune the volume up/down, setup specifications of playing file, show the current file, and Command PU to return to play file. The format of Command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	P	U	=	0x0D (enter)

#### Ex.

```
char enter = 0x0D ;
printf("*P1=07%c",enter) ; // ส่งคำสั่งเล่นไฟล์ลำดับที่7 ใน Micro SD Card
Echo_cmm() ; // Check Echo Command 'OK' (ดู Function จาก Respond Command) อาจใช้ delay

printf("*PU=%c",enter) ; // ส่งคำสั่งหยุดไฟล์ชั่วคราว
Echo_cmm() ; // Check Echo Command 'OK'
printf("*PU=%c",enter) ; // ส่งคำสั่งหยุดอีกครั้งเพื่อกลับมาเล่นไฟล์ต่อ
```

แทนได้

#### - 5. COMMAND 'ST' (Stop Play File )

This Command stops playing all song files (STOP). It exits from the Command Play that playing the current file instantly; next, it goes to the initial status to receive other command. The format of Command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	S	T	=	0x0D (enter)

#### Ex.

```
char enter = 0x0D ;
printf("*P1=19%c",enter) ; // ส่งคำสั่งเล่นไฟล์ลำดับที่19 ใน Micro SD Card
Echo_cmm() ; // Check Echo Command 'OK' (ดู Function จาก Respond Command) อาจใช้ delay

printf("*ST=%c",enter) ; // ส่งคำสั่ง STOP ไฟล์ พร้อมรับคำสั่งใหม่
```

แทนได้

#### - 6. COMMAND 'FW' (Forward)

This Command is used to move 1 sound file forwards in each time (FW). In this case, when it sends 1 command, it also moves 1 sound file forwards. When user sends this Command FW to move files forwards until it reaches the last file, it returns to the first file again. This moving file is in the format of Close Loop and it has to send this Command FW while playing file. The format of command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	F	W	=	0x0D (enter)

**Ex.**

```
char enter = 0x0D ;
printf("*FW=%c",enter) ; // ส่งคำสั่งเลื่อนไฟล์เสียงไปข้างหน้า 1 ไฟล์
```

**- 7. COMMAND 'RV' (Revert)**

This Command is used to move 1 sound file backwards in each time (RV). In this case, when it sends 1 command, it also moves 1 sound file backwards. When user sends this Command RV to move files backwards until it reaches the first file, it returns to the last file again. This moving file is in the format of Close Loop and it has to send this Command RV while playing file. The format of command is shown in the table below;

Start	CMM1		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	R	V	=	0x0D (enter)

**Ex.**

```
char enter = 0x0D ;
printf("*RV=%c",enter) ; // ส่งคำสั่งเลื่อนไฟล์เสียงถอยหลังมา 1 ไฟล์
```

**- 8. COMMAND 'VL' (Volume)**

This Command is used to tune the volume up/down (Volume). When user requires adjusting volume in the blank of Volume (Byte5-6), it has to add 2 digits of ASCII from 00 (Silent) to 99 (loudest). The format of Command is shown in the table below;

Start	CMM		Mark1	Volume(2byte)	END
Byte1	Byte2	Byte3	Byte4	Byte5-6	Byte7
*	V	L	=	00-99	0x0D (enter)

Byte5...Byte6 = Volume ASCII 2 Byte '00-99'

**Ex.**

```
char enter = 0x0D ;
printf("*VL=68%c",enter) ; // ส่งคำสั่งปรับ Volume ไปที่ระดับความแรง 68
```

**- 9. COMMAND 'CF' (Configuration Play)**

This Command is used to set Configuration of playing sound file. User should set the Configuration of playing sound file completely before using the Command Play because it starts playing the sound file instantly, according to the setting Configuration after using the Command Play. This Configuration value is 3Byte ASCII and it is replaced by '0' and '1'. The Default value is always set at '000'. It always remembers the Configuration value that is set by user until it is reset; or, it sends Command CF to set new value. In this case, Command STOP has not any effect on this Configuration because it does not change any value. If user

requires changing or editing any Configuration value for playing file to be different from the Default value while writing program, user should set this Command above program because Board MP3 always enables (ON) this Configuration value for playing file. The Configuration value of playing file can be set according to Byte as follows;

- 1) **Byte5: '0'** = It starts playing sound file instantly after received any Command Play, without waiting for the end of current sound file.  
**Byte5: '1'** = It waits until it finishes playing the current sound file first and then it starts splaying the incoming Command Play.
- 2) **Byte6: '0'** = It plays all sound files in Micro SD Card one time, or it only plays the specific sound file one time and stops. Next, it is ready to receive the new command.  
**Byte6: '1'** = It plays all sound files in Micro SD Card repeatedly, or it only plays the specific sound files repeatedly and continuously.
- 3) **Byte7: '0'** = It disables Play Time while playing sound file (Play Time Disable).  
**Byte7: '1'** = It enables Play Time while playing sound file (Play Time Enable).

The format of Command is shown in the table below;

Start	CMM		Mark1	Value Configuration. (3byte)			END
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
*	C	F	=	0-1	0-1	0-1	0x0D (enter)

**Byte5...Byte7** = 3Byte Configurations for playing file, each Byte is ASCII "0" or "1" only; please see the further meaning in the table.

**Ex.**

```
char enter = 0x0D
```

```
printf("CF=101%c",enter) ; // ร้องนกกว่าไฟล์ที่เล่นอยู่จบลงแล้วถึงเล่นไฟล์ที่ส่งมา, เล่นไฟล์ที่ส่งมาเพียงรอบเดียว
```

```
หรือ // และแสดงเวลาขณะเล่นด้วย
```

```
printf("CF=011%c",enter) ; // กำหนดให้เล่นไฟล์ทันทีเมื่อมีการสั่ง Play และเล่นไฟล์นั้นซ้ำ แสดงเวลาขณะเล่นด้วย
```

**- 10.COMMAND 'NC' (Show File Name Current)**

This Command is used to call to see the current sound file. The format of Command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	N	C	=	0x0D (enter)

When this Command is sent out while playing any sound file and the command is done completely, it shows Respond with the filename and file surname of the current sound file and it ends with Respond of Command that is '#OK'. The format of command is shown below;

**\$NN.FFFFFFFF.MP3 (หรือ WAV)**

← Respond File Name Current

NN = Order of the current sound file number is ASCII 2Byte ('00-99').

FFFFFFF = The current file name that is ASCII does not exceed 8Byte.

**Ex. How to send Command NC**

```
char enter = 0x0D ;
printf("*NC=%c",enter) ; // ส่งคำสั่งเรียกดูรายชื่อไฟล์เสียงที่กำลังเล่นอยู่
```

**Ex. Respond**

\$03.TEST1.MP3 : ไฟล์ที่กำลังเล่นอยู่เป็นไฟล์ลำดับที่ 3 ชื่อ ไฟล์ TEST1.MP3

**- 11. COMMAND 'NA' (Display All File Name In Card)**

This Command is use to call to see all sound files surname that are .MP3 and .WAV in Micro SD Card. If this Command is sent out while playing sound file, it stops playing file instantly and all List File will be sent out through RS232. When using Program Terminal to receive incoming data, user can see order of file and List file as shown in the example. This order of file and list file can be used in Command Play by Order or Command Play by Name. The Format of Command is shown in the table below;

Start	CMM		Mark1	END
Byte1	Byte2	Byte3	Byte4	Byte5
*	N	A	=	0x0D (enter)

**Ex. Respond all files in Card**

หมายเลขลำดับที่ไฟล์

ชื่อและนามสกุลไฟล์

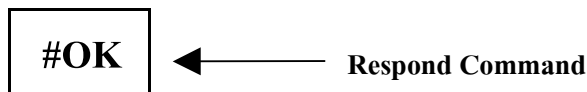
Respond คำสั่ง NA

**Ex. How to send Command NA**

```
char enter = 0x0D ;
printf("*NA=%c",enter) ; // ส่งคำสั่งเรียกดูไฟล์ทั้งหมดใน Micro SD Card
```

## - 12. RESPOND COMMAND

This Respond is from Board MP3, every time the Command that is sent to Board MP3 is correct and it is done completely. The format of Respond is 3Byte ASCII in capital letter as shown below;



**Ex.** How to check Respond (Echo) Command (Code for AVR Mega 128 ;Compiler WinAVR)

```

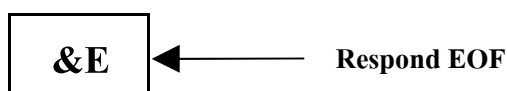
//----- Check Echo Command (#OK)-----

void Echo_Cmm()
{ unsigned char buf_cmm[2] ;
  unsigned char n , ch ;
  do{
    do{
      loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
      ch = UDR0 ;//Read data Start Byte
    }while(ch != '#') ;//Check Byte Start of Echo cmm('#')
    for(n=0;n<2;n++) ;//if ch = # then Keep cmm 2 byte(Keep 'OK')
    {
      loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
      buf_cmm[n] = UDR0 ;//Read Data 'OK'
    }
  }while((buf_cmm[0]!='O') || (buf_cmm[1]!='K')) ;//Check Echo Command = 'OK' Exit Loop
}
  
```

The operation of this program starts reading Byte Start to check if it is '#'. If yes, it continues reading more 2Byte Data to check if it is 'OK'. If yes, it exits from Loop. If no, it returns to read Data again.

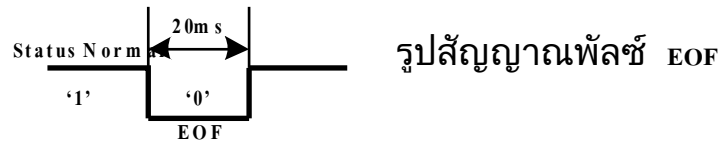
## - 13.RESPOND End Of File

This Respond is from Board MP3. Every time it finishes playing song file; although the song file is ended by normal play, or Command STOP, Forward, or Reverse, Board MP3 always sends this Respond. The format of Respond is 2Byte ASCII in capital letter as shown below;



This Respond Command is sent out in the format of ASCII through RS232; moreover, it is sent out in the format of Pulse Logic '0' with 20ms Width through Connector No.15 (OUT-EOF). While there is no any Signal EOF, Signal EOF is Logic '1' (3.3V).





**Ex.** How to check Respond (Echo) EOF through RS232 (Code for AVR Mega 128; Compiler WinAVR)

```
//----- Check Echo End Of File (&E) -----
void Echo_EOF()
{ unsigned char eof, ch ;
  do{
    do{
      loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
      ch = UDR0 ;//Read Data
    }while(ch != '&') ;//Check Byte Start of Echo EOF('&')
    loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
    eof = UDR0 ;//Read Data
  }while(eof != 'E') ; //Check Echo EOF = 'E' Exit loop
}
```

Refer to the example above; it is the method to check Respond EOF. It starts reading Byte Start to check if it is '&'; If yes, it continues reading more 1 Byte data to check if it is 'E'. If yes, it exits from Loop; on the other hand, if no, it returns to read Data again.

**Ex.** How to check Respond EOF from Signal Pulse (Code for AVR Mega 128 ;Compiler WinAVR)

```
//----- Check Status Signal EOF : End OF File -----
void Chk_Eof()
{ char eof ;
  do { //Loop Check Eof = 0 :end file
    eof = PINB ;//Read PB0 : 0 = end of message , 1 = Ready for Play or Playing
    eof = eof & 0x01 ;//Mark bit 0
  }while(eof == 1) ;//eof = 0 exit loop
  do{ //Loop Check EOF = 1 : Ready
    eof = PINB ;//Read PB0 : 0 = end of message , 1 = Ready for Play or Playing
    eof = eof & 0x01 ;//Mark bit 0
  }while(eof == 0) ;//eof = 1 exit loop
}
```

This example shows 2 steps of checking Signal EOF. Firstly, it checks if Signal EOF is '0'; if yes, it means that it finishes playing the file. Secondly, it checks if Signal EOF returns to be '1'; if yes, it means that it is ready to receive the next command. If user only checks the first step and sends the command instantly but the Signal EOF does not return to '1', the command that is sent out may not be done.

**- 14. RESPOND Play Time**

This Respond is used to send Play Time from Board MP3 through RS232 all the time while playing song file. User can use Program Terminal to show Play Time and can use MCU to read the value of Play Time while playing sound file. If user would like to set Board MP3 to send Respond Play Time, user has to send Command CF; in this case, it has to set the 7<sup>th</sup> Byte of the command to be '1' to Enable Play Time first (Default Disable Play Time). The format of this Respond is 6Byte ASCII in capital letters as follows;

@MM:SS

← Respond Play Time

MM = Minute ASCII 2 Byte

SS = Second ASCII 2 Byte

**Ex.** How to read Respond Play Time (Code for AVR Mega 128 ;Compiler WinAVR)

```
//----- Read Play Time -----//
//   Buf_Time[0] = Ascii Minute Byte High   //
//   Buf_Time[1] = Ascii Minute Byte Low    //
//   Buf_Time[2] = Ascii :                   //
//   Buf_Time[3] = Ascii Secound Byte High  //
//   Buf_Time[4] = Ascii Secound Byte Low   //
//-----//

void RD_Time()
{ unsigned char ch, n ;
  do {
    loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
    ch = UDR0 ;//Read data Start Byte
  }while(ch != '@') ;//Check Byte Start of Echo cmm ('@')

  for(n=0 ; n<5 ; n++) ;//if ch = @ then Keep data 5 byte(MM:SS)
  {
    loop_until_bit_is_set(UCSR0A,RXC) ;//Check data in Rx_buffer
    Buf_Time[n] = UDR0 ;//Read Data 'MM:SS'
  }
}
```

Refer to the example above, it sets the Variable *Buf\_Time[]* for storing the Play Time that is read. When it starts the program, it checks the Byte Start first; if it is '@', it starts storing 5 Byte into the Variable *Buf\_Time[]* respectively.

**NOTE:** Every time after sending the Command, it always checks the **Echo (#OK)**, or it uses Time Delay before sending the next command. It protects the operation from sending and overlap of command, otherwise it has error when sending Command to control Board MP3.

**Table: Summarize COMMAND and RESPOND COMMAND ET-REMOTE MP3 V2.**

No.	Name (CMM)	Start	CMM			Mark1	Byte Data			End Byte
		Byte1	Byte2	Byte3	Byte4					
1	Play File All (P0)	*	P	0	=	0x0D (Enter)				
2	Play File By Number (P1)	*	P	1	=	01-99			0x0D	
3	Play File By Name (P2)	*	P	2	=	FFFFFFFF.MP3			0x0D	
4	Pause Play File (PU)	*	P	U	=	0x0D (Enter)				
5	Stop Play File (ST)	*	S	T	=	0x0D (Enter)				
6	Forward (FW)	*	F	W	=	0x0D (Enter)				
7	Reverse (RV)	*	R	V	=	0x0D (Enter)				
8	Volume (VL)	*	V	L	=	00-99			0x0D	
9	Configuration Play (CF)	*	C	F	=	0-1	0-1	0-1	0x0D	
10	Show File Name Current (NC)	*	N	C	=	0x0D (Enter)				
	Respond Show File	\$NN.FFFFFFFFF.MP3 (หรือ WAV)								
11	Display All File Name In Card (NA)	*	N	A	=	0x0D (Enter)				
12	Respond Command	#OK -----> 3 Byte								
13	Respond End Of File	&E -----> 2 Byte								
14	Respond Play Time	@MM:SS -----> 6 Byte								

0x0D = Hexadecimal number to replace Enter

No.2 = Byte Data '01-99' is the order number of Ascii 2 byte

No.3= Byte Data 'FFFFFFFF.MP3'is 8Byte file name and 3Byte File surname at the maximum

No.8=Byte Data '00-99' is Volume Ascii 2 Byte; when '00' = Silent, '99' = Loudest

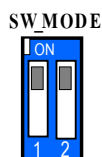
No.9 = Byte data '0-1 , '0-1' , '0-1' is Configuration Ascii 3 byte; each Byte can be set to be either '0' or '1'

No.10 = Respond Show File NN= order of file, FFFFFFFF =8Byte File Name

No.14 = Respond Play Time MM=minute, SS= second

**4.1.3) Application**





- 1) Copy and paste File surname .MP3 or .WAV into Micro SD Card, and then insert the Card into the Socket
- 2) Check if Jumper JP1-JP4 on board is set and the headphone is interfaced completely.
- 3) Interface Cable RS232 for sending Command to PC or MCU completely.
- 4) Set DIP-SW\_MODE by setting it to the position of COMMAND MODE (S1=ON,S2=ON) as shown in the picture below;



Picture shows how to set SW\_MODE to run in COMMAND MODE.

- 5) Set DIP-SW\_CONFIG to choose and set Baud Rate for sending-receiving data as required, please refer to the table below. Every time user changes the value of Set DIP-SW., it always resets the Board to update the new change.

**Table: Show how to Set DIP-SW\_CONFIG (Set Baud Rate for COMMAND MODE)**

DIP-SW_CONFIG		Format of SET SW_CONFIG	BAUD RATE
S1	S2		
ON (0)	ON (0)		115200
OFF (1)	ON (0)		57600
ON (0)	OFF (1)		9600
OFF (1)	OFF (1)		4800

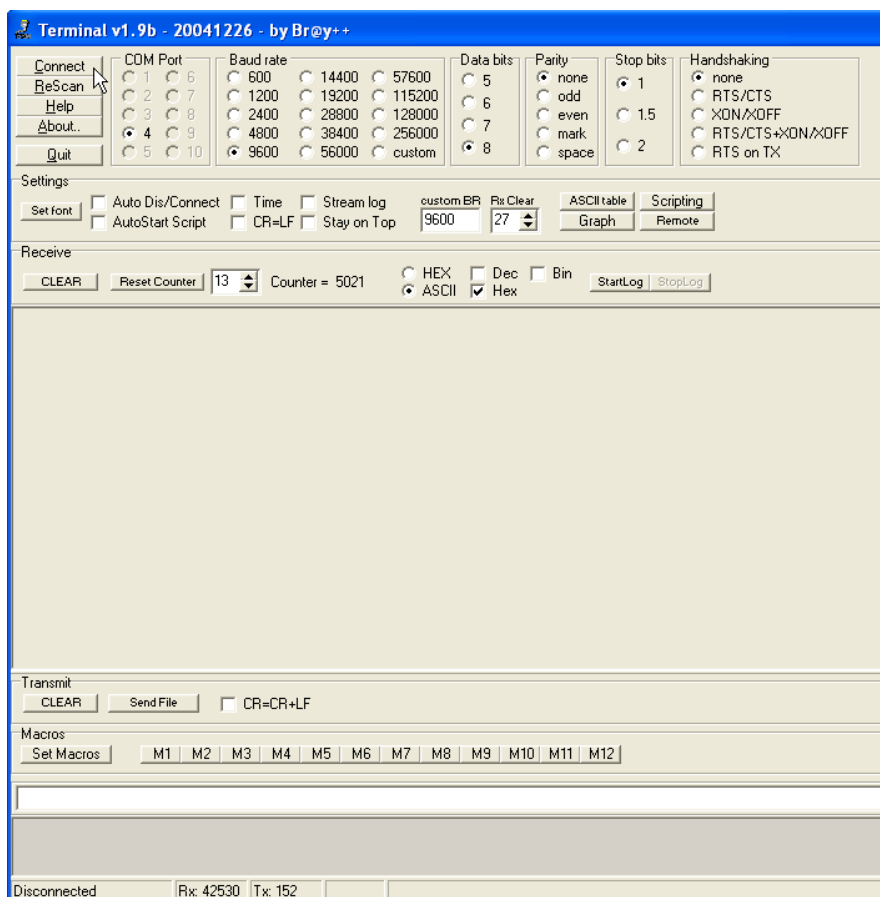
- 6) Provide Power Supply DC 5V for board (be careful, do not alternate pole), user can see that LED-PWR is lit up and it is ready to run.
- 7) Use Program Terminal when interfacing with PC or MCU; please try to send the Command as mentioned above. In this case, user should set Configurations of RS232 as described above.

**Example of sending Command through RS232 by Program Terminal**

**Step 1:** Interface Cable RS232 from PC to Connector RS232 of Board ET-REMOTE MP3.V2 completely.

**Step 2:** Set DIP-SW MODE to run in COMMAND MODE (S1,S2=ON) and Set DIP-SW CONFIG to choose and set Baud Rate = 9600 (S1=ON,S2=OFF).

**Step 3:** Copy Program Terminal from CD into PC and run this Program Terminal; next, set values as shown in the picture below. When user set values completely, click Button **Connect** to OPEN PORT RS232 of PC.



Com Port : ตามที่ผู้ใ้ต่อ

Baud rate : 9600

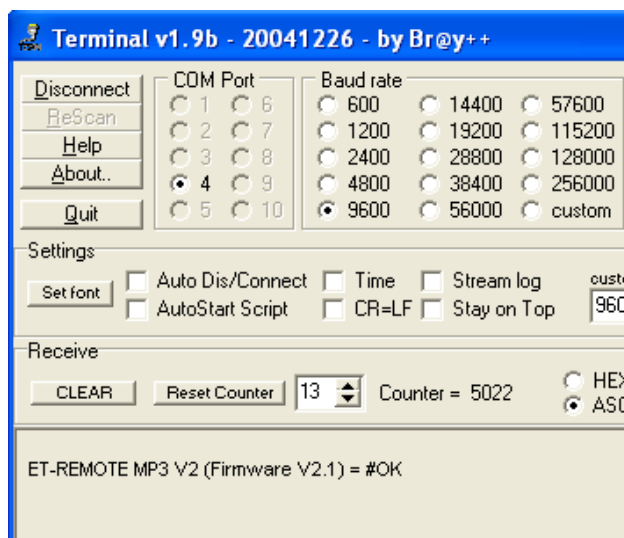
Data bit : 8

Parity : none

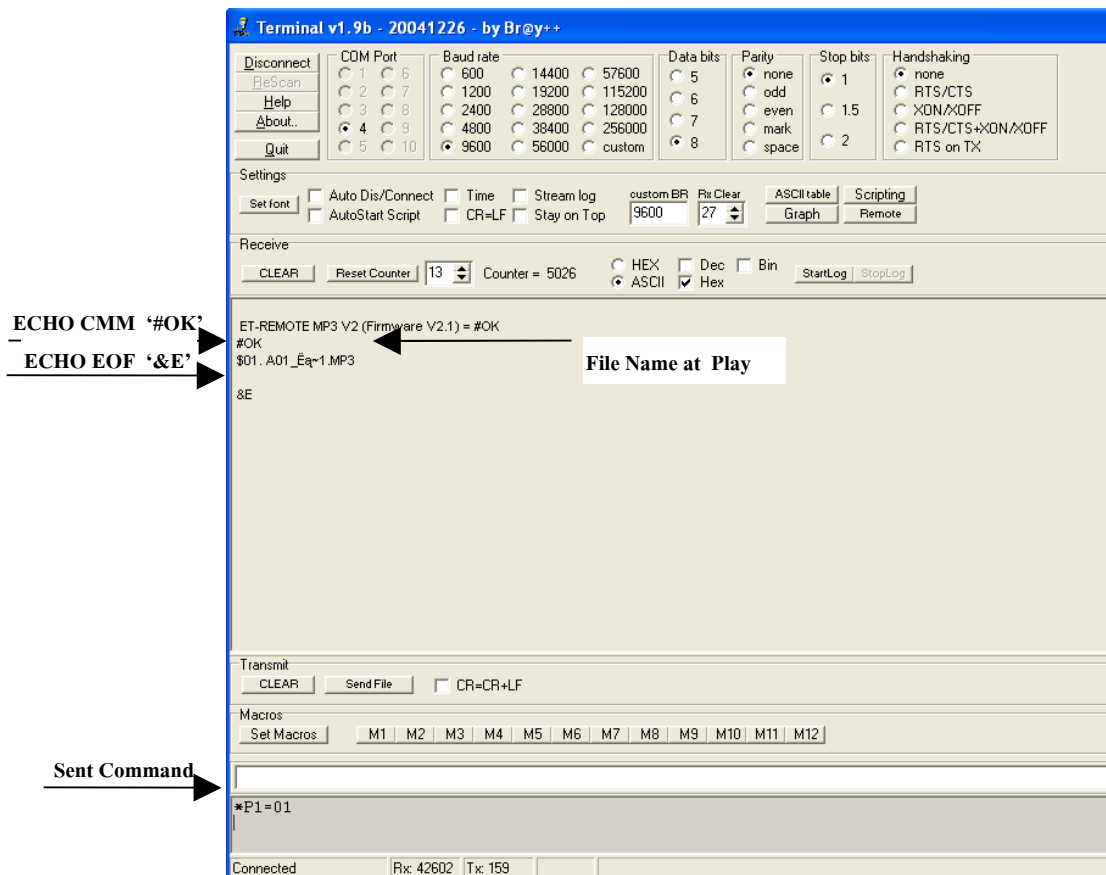
Stop bit : 1

Handshaking : none

**Step 4:** Provide 5V Power Supply for Board ET-REMOTE MP3.V2, and it shows message on Program Terminal as shown in the picture below;



**Step 5:** Test the operation by typing Command Play to play the first sound file into the gray blank at the bottom; in this case, it types \*P1=01 and press *Enter* on board, Board sends Respond as shown in the picture below; and finally, the first sound file will be played instantly. User can type other commands to test the operation.



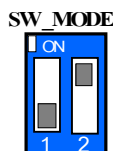
## 4.2 LIST\_SONG MODE

### 4.2.1) Operation

This operation mode is used to call to see list and order of all files surname .MP3 and .WAV in Micro SD Card. Initially, when user sets DIP-SW\_MODE to LIST\_SONG Mode and Set DIP-SW\_CONFIG to set Baud Rate for sending data; next, it provides Power Supply into Board MP3, the program starts reading file name and order of file from Micro SD Card. In this case, it always reads the first file that is copied into the Card and it responds with the order of file and file name that is read through RS232 of board. When user opens the Program Terminal (it has to open the program before providing Power Supply into board), or opens other programs that supports the display through RS232; in this case, it shows list of file name and order of file in Micro SD Card (please see more information from the section of Application). The list of file name and file surname that is shown through Program Terminal is always displayed in capital letters, although it is set by capital letters or small letters; moreover, the file name must not be greater than 8Byte(according to the format of file 8.3). If any file name is set in Thai, it shows strange and illegible instead. When user requires replacing the order of file, file name-file surname in the Command Play in COMMAND MODE, it has to refer to the result that shows in this mode. Every time user changes any data in Micro SD Card, it has to reset Board MP3 to update the new change.

**4.2.2) Application**

- 1) Copy and paste File Surname .MP3 or .WAV into Micro SD Card and then insert this Card into Socket.
- 2) Check if Jumper JP1-JP4 on board is set completely.
- 3) Interface Cable RS232 from Board MP3 to Connector RS232 DB9 of PC.
- 4) Set DIP-SW\_MODE to the position LIST\_SONG MODE (S1=OFF,S2=ON) as shown in the picture below;



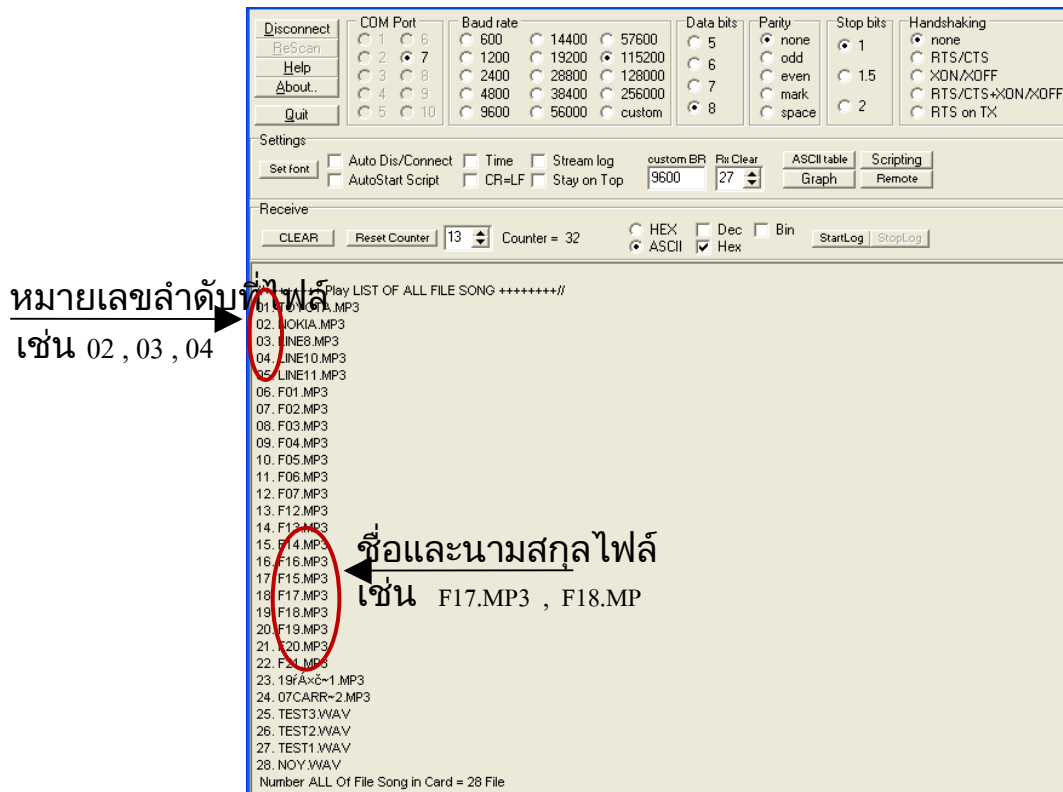
It shows how to set SW\_MODE to run in LIST\_SONG MODE.

- 5) Set DIP-SW\_CONFIG to set the preferable Baud Rate for sending data, please read more information in the table below. Every time user changes this Set DIP-SW., it always resets Board to update the new change.

**Table: Show How to Set DIP-SW\_CONFIG (Set Baud Rate for LIST\_SONG MODE)**

DIP-SW_CONFIG		Format of SET	BAUD RATE
S1	S2	SW_CONFIG	
ON (0)	ON (0)		115200
OFF (1)	ON (0)		57600
ON (0)	OFF (1)		9600
OFF (1)	OFF (1)		4800

- 6) Open Program Terminal or other programs that supports the display through RS232 and then set Configurations as follows;
  - Baud Rate: Set DIP-SW. in step No.4, DATA BIT: 8,
  - PARITY: NONE, STOP BIT: 1, HANDSHAK: NONE
- 7) Provide Power Supply DC 5V for Board (be careful, do not alternate pole); in this case, user can see that LED-PWR is lit up.
- 8) Program Terminal shows order of file and list of file name in Micro SD Card; moreover, it shows details in each part as shown in the picture below;



It shows order of file and list of file name that is called in **MODE LIST\_SONG**.

This describes how to replace order of sound file or filename-file surname in the Command Play in the COMMAND MODE such as Play. In case of Play by Order, it can be replaced by the preferable order of file to play such as 01 or 02 or 15. In case of Play by Name, it can be replaced by the preferable filename-file surname that is shown on Terminal such as F17.MP3 or TEST2.WAV.

- 9) When user changes any data in Micro SD Card or Set DIP-SW, it has to reset Board to update the new change.

### 4.3 MP3\_SW MODE

#### 4.3.1) Operation

This operation mode is used to control the operation of Board MP3 through 3 Switches on board; it is suitable for playing file as same as the general MP3 Player. These 3 Switches on board have different functions such as Play, Pause, Stop, Forward, Reverse, Volume up, and Volume Down; user can read more information of each function from section "Application" below. Moreover, it can use DIP-SW\_CONFIG to set Configurations of playing file; for example, it sets Board MP3 to play all files in Card one time and stop; or, it repeats all files; or, it starts playing sound file instantly after Power ON; or, it starts playing file after pressing SW.Play. Please read more information of setup value from table in the section "Application" below as well.

When this operation mode starts running, the program always checks what Configurations of DIP-SW\_CONFIG is set for playing sound file. For example, it starts playing file instantly after Power ON, or it has to press SW.PLAY/STOP first and then it starts playing the file. It always



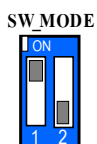
plays the first sound file that is copied into the Micro SD Card. When it finishes playing the first sound file, it continues playing the next order of file, 2,3,4,..., respectively until it plays all files in the Card. Every time it ends playing each sound file or it presses any Forward, Reverse, or Stop, Board always sends Signal EOF Active '0' to Connector No.15 (LED EOF alternates between ON and OFF), it notifies user to know that it finishes playing the sound file completely.

If it resets Board MP3 or Power OFF and then Power ON again while playing any sound file, it always starts playing the first order of sound file again; moreover, the setting Volume always returns to the Default value. This Board MP3 does not remember these values. However, any Configuration value that is set by DIP-SW\_CONFIG is unchanged, if user does not change or edit any value. The change of Set DIP-SW\_CONFIG will be effective or it updates the Configuration when it resets Board MP3 only.

There are 3 Switches on board and each switch can be set to run more than one function; it depends on the feature if pressing Switch. There are 2 features of pressing Switch to separate the operation. Firstly, it presses and holds the Switch less than 0.5 seconds and then remove (press and remove the Switch instantly); and secondly, it presses and holds the Switch longer than 0.5 second and then remove. User can read more information of each function from table "Application" below;

#### **4.3.2) Application**





- 1) Copy and paste File surname .MP3 or .WAV in Micro SD Card and then insert the Card into Socket.
- 2) Check if Jumper JP1-JP4 on board is set completely.
- 3) Set DIP-SW\_MODE by setting to the position of MP3\_SW MODE (S1=ON,S2=OFF) as shown in the picture below;



**It shows show to Set SW\_MODE to run in MP3\_SW MODE.**

- 4) Set DIP-SW\_CONFIG to set Configuration of playing file, please see more information from the table below. Every time user changes any value of Set DIP-SW., it always resets Board MP3 to update the new change.

**Table: Show how to Set DIF-SW CONFIG (Set properties play for MP3 SW MODE)**

DIP-SW_CONFIG		Format of SET	Properties Play
S1	S2	SW_CONFIG	
ON (0)	ON (0)		It plays sound file instantly after Power ON or Reset, without pressing SW.PLAY/STOP. It plays all files in the Card one time and then stops automatically.
OFF(1)	ON (0)		It plays sound file when pressing SW.PLAY/STOP. It plays all files in the Card one time and then stops automatically.
ON (0)	OFF (1)		It plays sound file instantly after Power ON or Reset, without pressing SW.PLAY/STOP. It plays all files in the Card repeatedly (Repeat All).
OFF (1)	OFF (1)		It only plays sound file when pressing SW.PLAY/STOP. It plays all files in the Card repeatedly (Repeat All).

- 5) Provide Power Supply DC 5V for board (be careful, do not alternate pole), user can see that LED-PWR is lit up.
- 6) Test the operation by pressing SW.PLAY/STOP to start playing sound file; next, test function of SW. as shown in the table below.
- 7) Function of 3 Switches on board when pressing is listed below;

**Table: Show functions of all 3 Switches on Broad For MP3 SW MODE**

SW. NAME	Shortly press (less than 0.5 seconds)-Release		Press and Hold (longer than 0.5 seconds)-Release	
	Press while playing file	Press while not playing file	Press while playing file	Press while not playing file
<b>PLAY/STOP</b>	STOP ([]) )	PLAY (< )	PAUSE(    )	-
<b>RV/VOL-</b>	Reverse 1 File (<<)	-	Volume Down (--)	Volume Down (--)
<b>FW/VOL+</b>	Forward 1 File (>>)	-	Volume Up (++)	Volume Up (++)

- 8) Every time it finishes playing sound file, or it presses SW.FW, or RV, Board MP3 always sends Signal EOF (Active LOW) to Connector OUT-EOF; in this case, user can see that LED-EOF is blinking. This signal can be used to check the end of playing file.

#### 4.4 TRIG MODE

##### 4.4.1) Operation

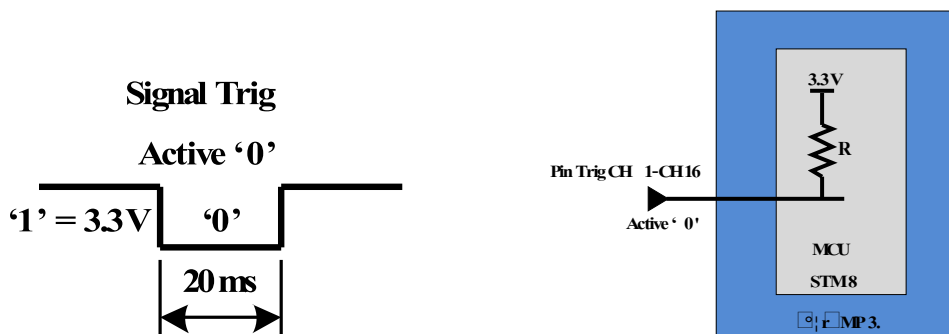
This operation mode is used to control the operation of Board MP3 through Connector Input TRIG (19.CON-TRIG) on board; there is 16-channel Connector to receive Signal TRIG (1-16). When Signal TRIG is coming in any channel, any sound file number that accords with the triggered channel number starts playing file instantly. For example, if there is incoming Signal TRIG at INPUT Channel 12, it start playing the song file number 12 instantly. This operation mode has restriction about amount of playing song file that does not exceed 16 files (file number 1-16 only); although there are more than 16 song files in the Card. If it has less than 16 song files in Card but it trigs unavailable or inexistent song file number in the Card, it does not play any song file.

Moreover, user can setup Configuration for playing sound file by DIP-SW\_CONFIG. For example, it plays the triggered sound file one time and then stops to wait for new Signal TRIG; or, it repeatedly plays the triggered song file. User can set the Configuration to play the triggered song file instantly or, it has to wait until it finishes the current song file first and then it starts playing the new triggered song file. If it trigs many times while playing file, it adheres to a principle of the latest Signal TRIG mainly. When it sets the Configuration of playing file by using DIP-SW\_MODE to set specifications of playing file, user can read more information from table of "Application". Furthermore, this operation mode can use 3 Switches on board to control stop (STOP) and tune the Volume up/down; in this case, each switch has one function as described in the table below when pressing and releasing the switch as normal.

**Table: Show functions of 3 Switches On Board For TRIG MODE**

SW. NAME	Press-Release Switch as normal	
	Press switch while playing file	Press switch while not play file
PLAY/STOP	STOP ([])	-
RV/VOL-	Volume Down (--)	Volume Down (--)
FW/VOL+	Volume Up (++)	Volume Up (++)

This operation mode is suitable for the application that receives Signal TRIG from pressing switch or receives value of Sensor from external device to trig. The feature of Signal Trig is Active Logic '0' with 20ms Width or higher. Moreover, when the voltage level is Logic '1', it does not exceed 3.3V because Input of MCU does not support 5V. The internal structure of Pin that receives Signal Trig is to interface with R-Pull Up with 3.3V as shown in the picture below;



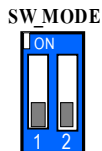
It shows the internal structure of Pin that receives Signal Trig CH1-CH16 and the feature of Signal Trig.

So, it can directly interface Connector that receives Signal Trig of Board MP3 with Pin I/O of device that sends Signal Trig such as MCU or Sensor uses the Power Supply that does not exceed 3.3V only. If voltage level the of device that sends Signal Trig exceeds 3.3V; or, Signal Logic '1' at Pin I/O exceeds 3.3V, it has to use circuit to reduce the voltage level, please read more information from the section of "Application" below;

**Warning:** Signal that is used to trig while it is Logic '1' does not exceed 3.3V only.

**4.4.2) Application**

- 1) Copy and paste the file surname .MP3 or .WAV into Micro SD Card and then insert the Card into Socket.
- 2) Check if Jumper JP1-JP4 on board has been set completely.
- 3) Set DIP-SW\_MODE to the position of TRIG MODE (S1=OFF,S2=OFF) as shown in the picture below;



**It shows how to set SW\_MODE to run in TRIG MODE.**

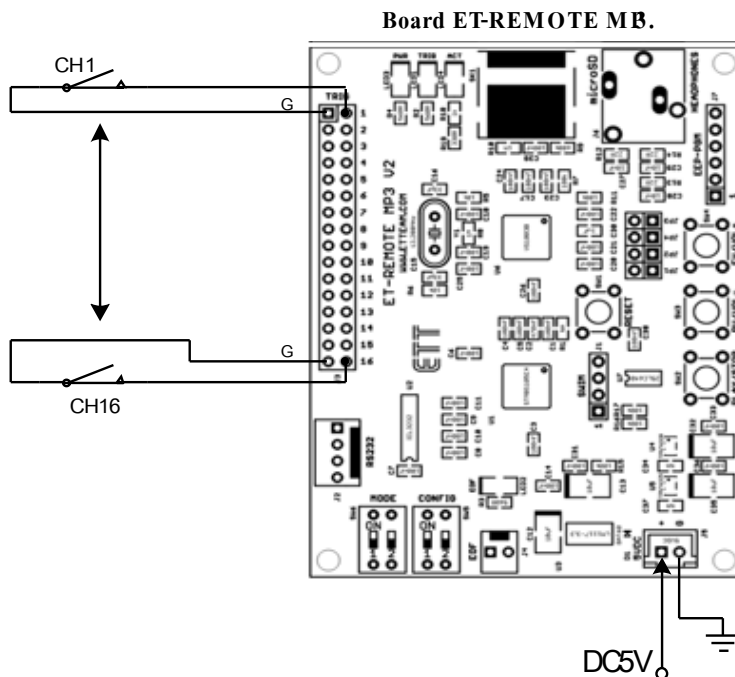
- 4) Set DIP-SW\_CONFIG to set Configuration for playing file, please see more information in the table below. Every time user changes any value of Set DIP-SW., it always resets board to update new value.

**Table: Show how to Set DIP-SW\_CONFIG (Set properties play for TRIG MODE)**

DIP-SW_CONFIG		Format of SET	Properties Play
S1	S2	SW_CONFIG	
ON (0)	ON (0)		It plays the triggered song file instantly. It plays the triggered song file one time and then stops instantly.
OFF (1)	ON (0)		It waits until it finishes playing the current song file first and then it starts playing the song file that is triggered lately one time and then stops.
ON (0)	OFF (1)		It plays the triggered song file instantly; in this case, it repeatedly plays this song file.
OFF (1)	OFF (1)		It waits until it finishes playing the current song file; next, it starts playing the recent song file that is triggered repeatedly.

*-If it sets DIP-SW to play the song file repeatedly, user can trig to play other song file while playing the current song file; moreover, it also plays the new triggered song file repeatedly.*

- 5) Provide Power Supply DC 5V for board (be careful, do not alternate pole), user can see LED-PWR is lit up.
- 6) Please look at Connector Trig; the inner board that is specified by the channel number is the channel to receive Signal Trig Low. On the opposite side of channel that receives Signal Trig (outer board), it is all Pin Ground. User can test the operation by triggering any channel into Ground (trig and release), any order of song file that accords with the triggered channel starts playing instantly (it should have at least 16 files in the Card to test Trig CH1-CH16). Refer to Connector Trig on Board; user can interface Push-Button Switch with each channel directly to trig and play the preferable order of song file. The method to connect SW.Trig is shown in the picture below;



It shows how to interface Push-Button Switch to trig and play sound file.

- 7) Every time after it finishes playing the song file as normal or it uses Trig to play new file, Board MP3 always sends Signal EOF (Active LOW) to Connector OUT-EOF; in this case, user can see that LED-EOF is blinking. User can use this signal to check the end of playing file.

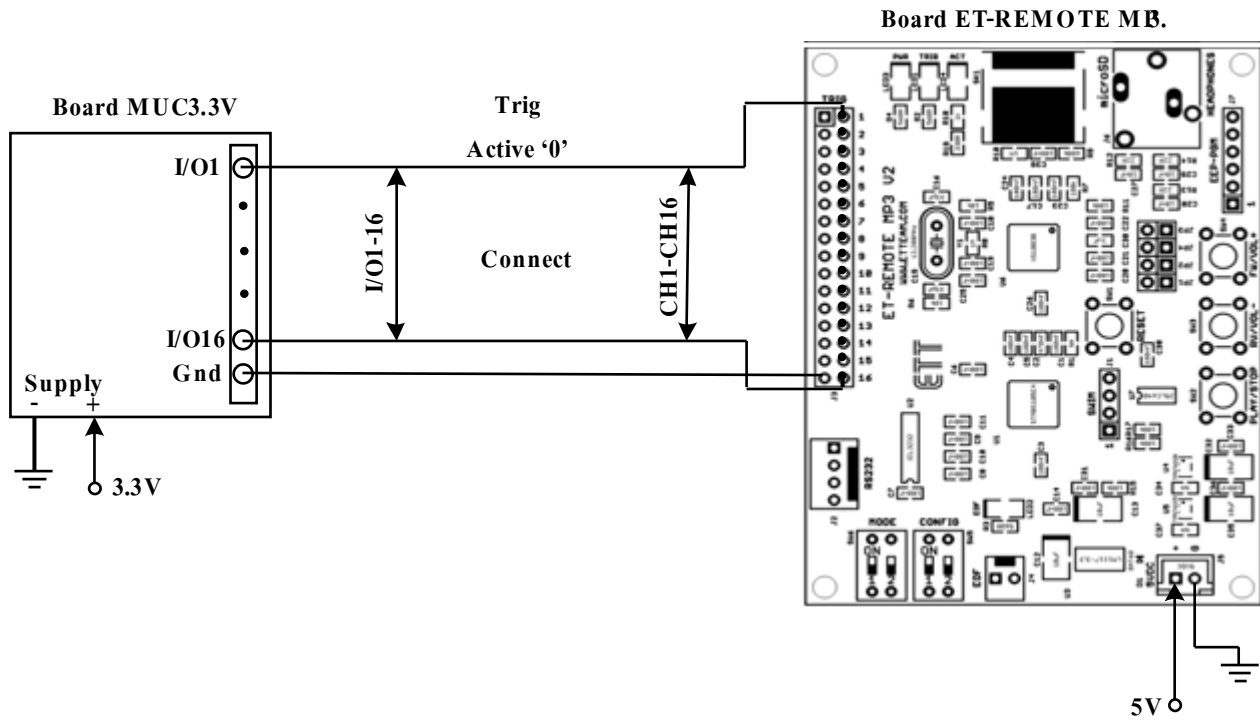
#### **4.4.3) How to trig by Signal TRIG from external Board MP3**

User can use MCU or Sensor to send Signal Trig from external to Connector Trig of Board MP3 to control to play sound file. In this case, user has to consider the Logic Level of Signal Trig while it is Logic '1' because Board ET-REMOTE MP3 can receive the level of Signal Trig that does not exceed 3.3V only. If the level of Signal Trig exceeds 3.3V, it has to do through circuit to reduce the Voltage Level to 3.3V first; then user can interface it to Connector that receives Signal Trig of Board MP3. Following example shows how to interface Signal Trig from external circuit to control to play sound file on Board MP3 (TRIG-MODE), it is good guideline for user to develop program in the future as described below;

##### **1) How to send Signal TRIG by MCU or Sensor 3.3V**

The method to send Signal Trig from external by MCU or Sensor 3.3V is to directly interface I/O of MCU to the Connector of Board MP3 that receives Signal Trig as shown in the picture below.

If it write program while it is in the normal status, it should set I/O 1-16 to be Logic '1'. If user requires playing sound file in any channel, it has to send Logic '0' to Pin I/O and hold the status for 20 ms or higher; next, it has to set this Pin I/O to be Logic '1' again. For the actual use, user can choose and interface any particular channel that user requires playing file. Amount of sound file in Micro SD Card should be equal to amount of select channel.

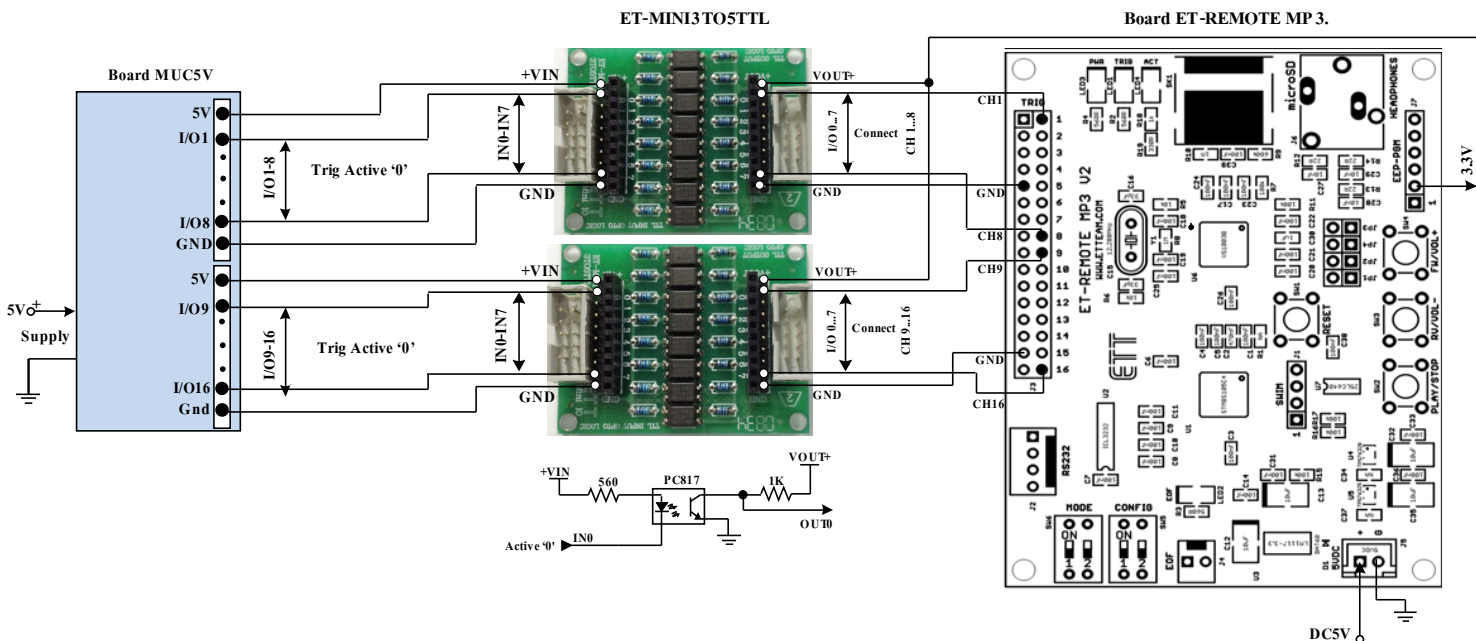


It shows how to interface Pin I/O 1-16 of Board MCU 3.3V with Connector Trig CH 1-16 of Board MP3.

**2) How to send Signal Trig by MCU or Sensor 5V**

If user requires sending Signal Trig from external by MCU or Sensor 5V to control to play sound file of Board MP3, it cannot interface I/O directly; in this case, it has to interface through circuit to reduce the voltage level into 3.3V first. There are many types of circuit to reduce the voltage level, it depends on the connective device; in this case, we would like to use the ETT device to reduce the voltage level of Signal Trig from 5V to 3.3V as follows;

- This is an example shows how to interface circuit when using 'ET-MINI3TO5TTL' to reduce the voltage level (it is suitable for triggering many channels).



It shows how to interface Pin I/O 1-16 of Board MCU 5V through ET-MINI3TO5TTL to reduce the voltage level.

Refer to the picture above, when it sends Signal Trig from external by MCU or Sensor 5V to control Board MP3, it uses 'ET-MINI3TO5TTL' to reduce the voltage level. So, the Signal Trig that is sent from MCU or Sensor must be Active '0' only. In the part of Signal Trig, user can choose the particular channel to interface for the actual use, it is unnecessary to interface all channels as shown in the picture.

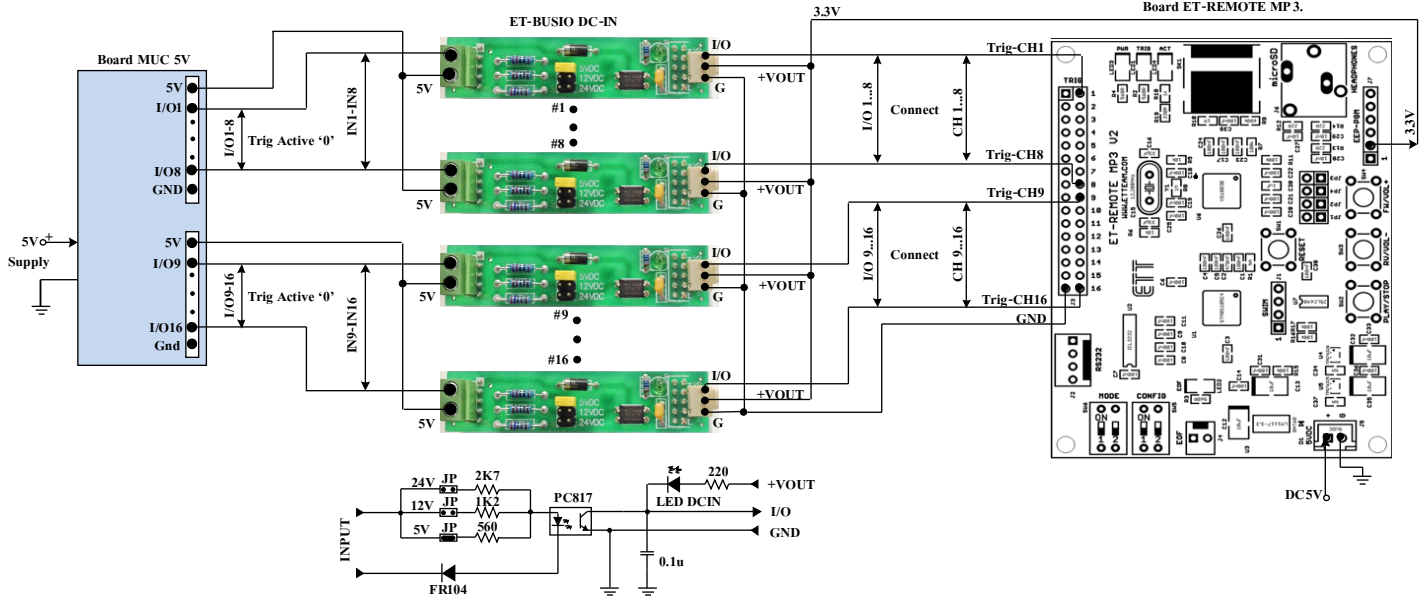
Writing program into this circuit while it is in the normal status that is no any Signal Trig, each Pin I/O of MCU must be set to be '1' first. When user requires playing any sound file, it has to send Signal Trig Logic '0' to Pin I/O in the preferable position to play the sound file and it should hold the status for 20 ms or higher. Next, it should reset Pin I/O to be Logic '1' again, and finally, the sound file in the channel that is triggered will be played instantly.

- **This is an example shows how to interface circuit when using 'ET-BUSIO DC IN' to reduce the voltage level (it is suitable for triggering 1 or more channels).**

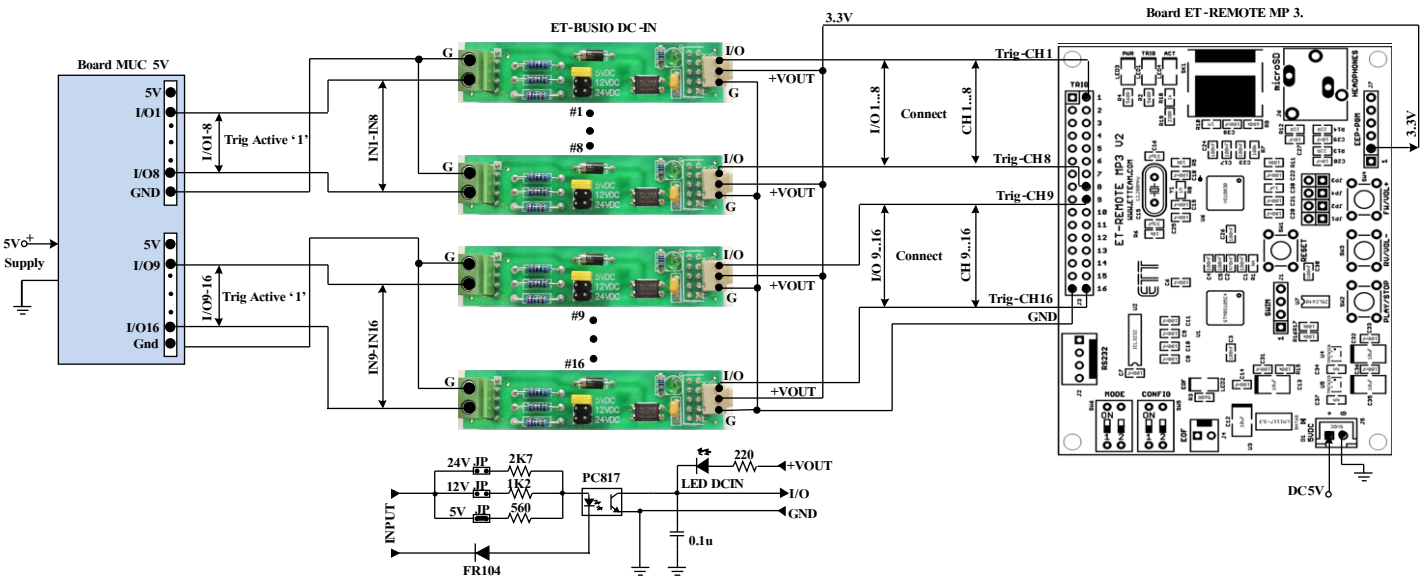
Board ET-BUSIO DC IN that reduces the Voltage level has only one channel on the side of Input. It can choose to receive 3 Voltage levels of Signal Trig from external; 5V, 12V, and 24V; in this case, it can set Jumper on Board ET-BUSIO DC IN to choose the preferable voltage level. Moreover, this board can receive external Signal Trig that is either Active '0' or Active '1'. Please refer to the circuit in picture (A), if using Signal Trig from external that is Active '0'; and please refer to the circuit in picture (B), if using Signal Trig from external that is Active '1'. When user interfaces the actual circuit, user can choose the particular channel to interface, it is unnecessary to interface all channels; however, amount of file in Micro SD Card must be equal to amount of channel. For example, if playing the file No.9, it should have at least 9 files in the Card.

This is description about writing program; please refer to the circuit in both pictures below. Refer to the circuit in the picture (A); it uses Signal Trig from external that is Active '0'. While it is in the normal status, there is no any Signal Trig and MCU sends Signal '1' for each Pin I/O. If user requires playing any sound file, it has to send Signal Trig Logic '0' to Pin I/O in the position of preferable sound file that will be played; in this case, it should hold the status for 20ms or more and then sets this Pin I/O to be Logic '1' again, and finally, it starts playing the chosen sound file instantly. Refer to the circuit in the picture (B); it uses Signal Trig from external that is Active '1'. While it is in the normal status, there is no any Signal Trig and MCU sends Signal '0' for each Pin I/O. If user requires playing any sound file, it has to send Signal Trig Logic '1' to Pin I/O in the position of preferable sound file that will be played; in this case, it should hold the status for 20ms or more and then sets this Pin I/O to be Logic '0' again.





(A) It shows how to interface Pin I/O 1-16 of Board MCU 5V for Trig (Active '0') through Board 'ET-BUSIO DC-IN' to reduce the Voltage level.



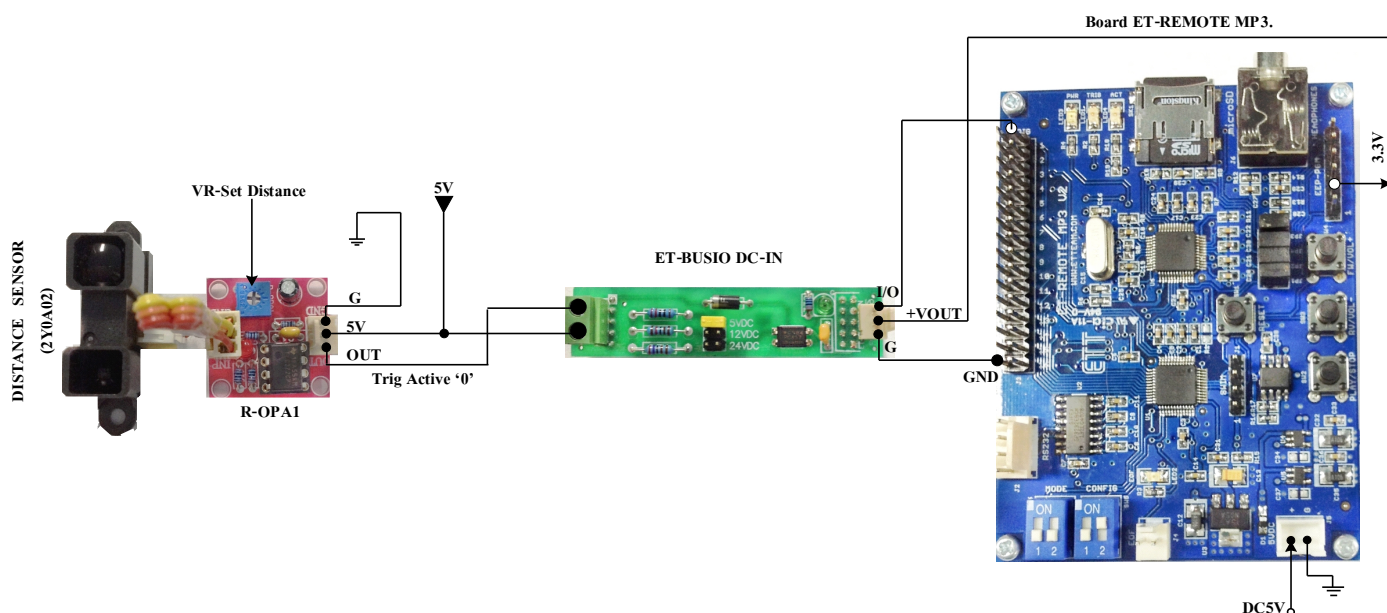
(B) It shows how to interface Pin I/O 1-16 of Board MCU 5V for Trig (Active '1') through Board 'ET-BUSIO DC-IN' to reduce the Voltage level.

### 3) Example of interfacing circuit by Distance Sensor to send Signal Trig

This section shows how to apply this program in Trig Mode by using Distance Sensor that runs by 5V Power Supply (#2Y0A02) to send Signal Trig from external instead of MCU to control to play sound file. It is interfaced with 'Board R-OPA1' of ETT that converts Signal Analog of the Distance Sensor to be either Signal Digital '0' or '1'; next, it uses this Signal to trig. When Output from this R-OPA1 is in the normal status, it is '1'; but when any obstacle blocks or obstructs the Distance Sensor in the specific distance, Output is Logic '0'. In summary, this Signal Trig runs (Active) by Logic '0'. In this case, it uses Board 'ET-BUSIO DCIN' to reduce the Voltage level of Signal Trig from 5V to 3.3V first; next, it



interfaces Signal to Connector Signal Trig of Board MP3 as shown in the circuit below;



**It shows how to interface Distance Sensor to control to play file of Board MP3 in TRIG Mode.**

Refer to the circuit above; the 5V Power Supply that provides for Sensor and Board MP3 is separated into 2 sets to interface into Ground separately as shown in the circuit above. Or, it uses only one set and then interface into Ground together. First of all, user has to interface circuit as shown in the picture above; in this case, it uses Channel No.1 of Signal Trig of Board MP3. For Set DIP-SW\_CONFIG, user can set the format of playing file as required. Next, user has to setup value of Sensor by using obstacle to block Sensor in the specific distance that user require detect; and then it should adjust VR on Board R-OPA1 until LED on board is lit up, it means that it is work. Next, user can test the operation of Sensor by blocking the Sensor and then see the Signal from Sensor is sent to trig Board MP3 in the first channel; and finally, the sound file in the first order will be played instantly.

## 5. Describe how to run example Program in CD

Examples that are provided in CD illustrate how to control Board ET-REMOTE MP3 V2 in COMMAND MODE and TRIG MODE by using MCU to be controller. It divides MCU into 2 groups. Firstly, it is a group of MCU 3.3V such as #ARM7 LPC2138 and #STM8S208. Secondly, it is a group of MCU 5V such as #AVRmegal28, #PIC18F8722, and #89C51RE2. Example program of each MCU is the same, so user should see and read details and circuit from the Comment above each program well before loading program to test the operation. The operation of each example program can be described as follows;

### 5.1 Example Program for COMMAND\_MODE

The method to interface circuit for this example program in COMMAND MODE is to interface Cable RS232 from Board MP3 to Board MCU that is used to send command. Connector RS232 in the part of Board MCU, it has to be interfaced through Line Driver MAX232 first; it is not allowed to interface directly from Pin of MCU.

#### Ex1\_SENT\_CMM

This example program illustrates how to play sound file from No.1 to No.15 and then it returns to play the sound file from No.2 to No.15, and so on. So, it should have at least 15 sound files in Micro SD Card for running this program.

The operation of program starts sending the Command CF to set the format of playing file for Board MP3; follows by Command Set Volume and then follows by Command P1 to play the first file track. When it sends command in each time, it always checks Echo Command after the end of playing the first file; it checks from function Echo\_EOF(), the program sends Command P1 to play the second sound file. Every time it checks and finds this EOF, the program always sends the Command P1 to play the next sound file until it reaches the sound file No.15. Next, it returns to play the sound file No.2 again and so on.

#### Ex2\_Read\_Play\_Time

This example is used to play the sound file name **A23\_FT1.MP3** for 30 seconds; next, it plays the sound file name **A24\_FT2.MP3** for 1 minute; and, it returns to play the file name **A23\_FT1.MP3** again. It repeats to play the sound files in this format continuously. When running program, it should setup the sound file name in Micro SD Card as mentioned above, or it should change the sound file in the program according to the file name in Micro SD Card; in this case, the length of the sound file should be longer than 1 second.

The operation of program starts with sending Command CF to setup Board MP3 to show Play Time. Next, it starts sending the command Play by Name, checks Echo Command and calls Function RD\_Time() to read the Play Time. The Play Time is stored in the variable Buf\_Time(0-4); in this case, Buf\_Time[0-1] stores value in the part of minute and Buf\_Time[3-4] store value in the part of second. It checks if the current sound file is played for 30 seconds completely; if no, it returns to read the Play Time again; if yes, it starts sending the next Command Play by Name to play the second file. Next, it reads the Play Time of the second file. It checks if the second file is played for 1 minute completely; if yes, the Program returns to play the song file that is specified at the beginning again.

**5.2 Example Program for TRIG\_MODE**

The method to interface circuit for the example in TRIG MODE for MCU 5V is to use 'ET-MINI3TO5', it reduces the voltage level from Pin I/O that is 5V into 3.3V before interfacing to Pin Trig of Board MP3. If using MCU 3.3V, it can interface Pin I/O of MCU to Pin Trig of Board MP3 directly. Connector EOF from Board MP3 can be interfaced with Pin I/O of MCU that is either 5V or 3.3V directly, without interfacing through any Circuit. In this case, it has to pull-up Pin G if G of Board MP3 and MCU are not interfaced together.

**Ex1\_Trig\_By\_Cmm**

This example illustrates how to receive Command from user through RS232 of Board MCU. The format of Command is in the table below. When user sends Command into Board MCU, the Board also sends Signal Trig to Pin I/O according to the specific pin position in the Command. Next, this Signal Trig is sent to pin of Board MP3 that receives Signal Trig, the song file in the channel number that is triggered will be played instantly.

ASCII Command For Test

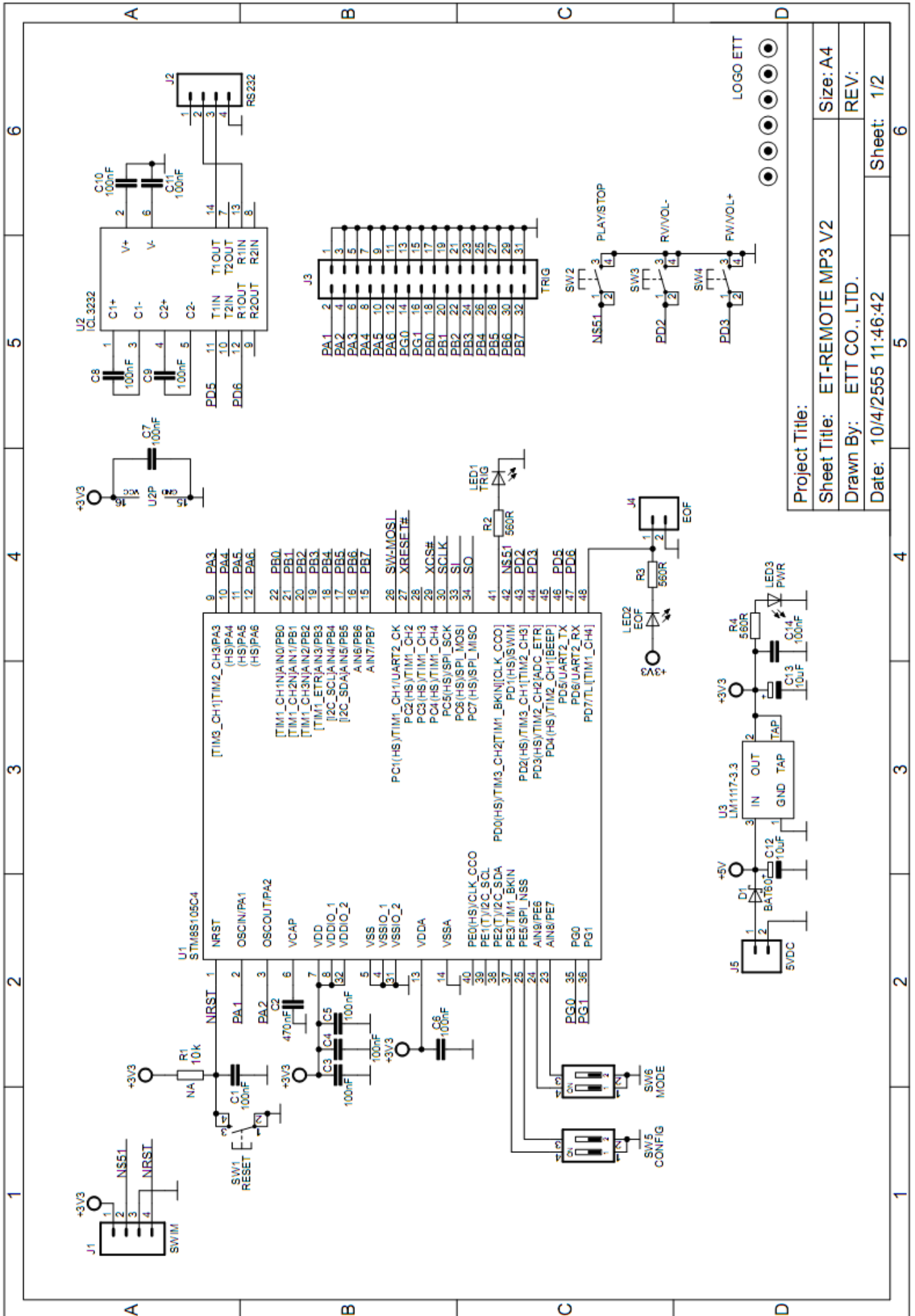
Start Byte	Number Song	End
1Byte	2Byte	1 Byte
*	01-16	0x0D (Enter)

The operation starts with setting Pin I/O that sends Signal Trig to be all Logic '1' first; and then it checks if the command that is sent from user is correct. If yes, it sends Signal Trig Logic '0' of the chosen Channel to the specific Pin I/O for 20ms; next, it sets Pin I/O to be Logic '1' again to wait for receiving the new Command.

**Ex2\_Trig\_By\_Check\_EOF**

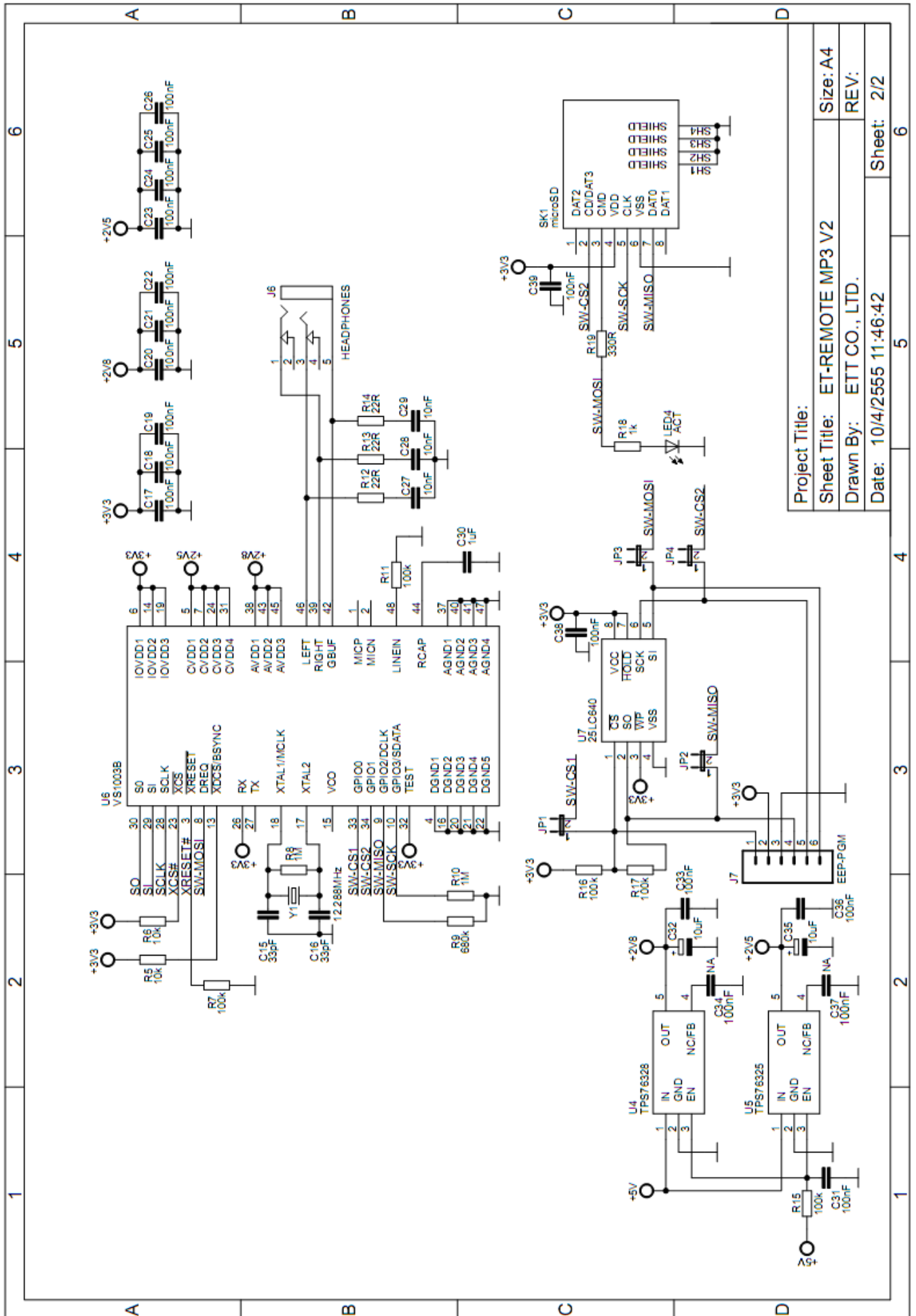
This example illustrates how to command MCU to send Signal Trig to Connector Signal Trig of Board MP3. It starts triggering the Channel from No.1-No.16; so, it starts playing the sound file from No.1-No.16, according to timing of Signal Trig. When it has triggered all 16 channels completely, the program returns to trig the Channel No.1 again. In this case, it should have at least 16 sound files in Micro SD Card for testing the program.

The operation starts with setting Pin I/O that sends Signal Trig to be all Logic '1' first; next, it calls function Play\_File(). This function starts sending Signal Logic '0' to Pin I/O of MCU that is interfaced with the first Channel that receives Signal Trig of Board MP3; next, it sets Time Delay for 20ms, and finally, it sets Pin I/O to be Logic '1' again. Next, it repeats to read Signal EOF from Board MP3 to check if it is Logic '0'; if yes, it means that it finishes playing the triggered song file and the program starts triggering to play the next sound file in the second channel. Every time it ends any sound file, it sends Signal Trig in the next channel continuously until all 16 channels are complete; next, it returns to trig the first channel again and so on.



Project Title: ET-REMOTE MP3 V2		Size: A4
Drawn By: ETT CO., LTD.		REV:
Date: 10/4/2555 11:46:42		Sheet: 1/2

Circuit of Board ET-REMOTE MP3 V2 (Sheet1)



Project Title:		ET-REMOTE MP3 V2	Size: A4
Sheet Title:		ET-REMOTE MP3 V2	REV:
Drawn By:		ETT CO., LTD.	Sheet: 2/2
Date:		10/4/2555 11:46:42	6

Circuit of Board ET-REMOTE MP3 V2 (Sheet2)