

สารบัญ

เรื่อง	หน้า
1. คุณสมบัติของบอร์ด ET-REMOTE MP3. V2	2
2. คุณสมบัติและการจัดการ File สำหรับ Micro SD Card	3
3. ลักษณะของบอร์ด ET-REMOTE MP3. V2	5
4. การใช้งานบอร์ด ET-REMOTE MP3 V2.	10
4.1 COMMAND MODE	10
4.1.1) การทำงาน	10
4.1.2) รูปแบบคำสั่งและ Respond ที่ใช้กับ ET-REMOTE MP3 V2.	11
4.1.3) การใช้งาน	20
4.2 LIST_SONG MODE	22
4.2.1) การทำงาน	22
4.2.2) การใช้งาน	23
4.3 MP3_SW MODE	24
4.3.1) การทำงาน	24
4.3.2) การใช้งาน	25
4.4 TRIG MODE	26
4.4.1) การทำงาน	26
4.4.2) การใช้งาน	27
4.4.3) การ Trig โดยใช้สัญญาณ Trig จากภายนอกบอร์ด MP3	29
5. อธิบายการทำงานของโปรแกรมตัวอย่างใน CD	33
5.1 ตัวอย่างโปรแกรมสำหรับ COMMAND_MODE	33
5.2 ตัวอย่างโปรแกรมสำหรับ TRIG_MODE	34
Circuit Board	35

ET- REMOTE MP3. V2.0


1. คุณสมบัติของบอร์ด ET-REMOTE MP3. V2

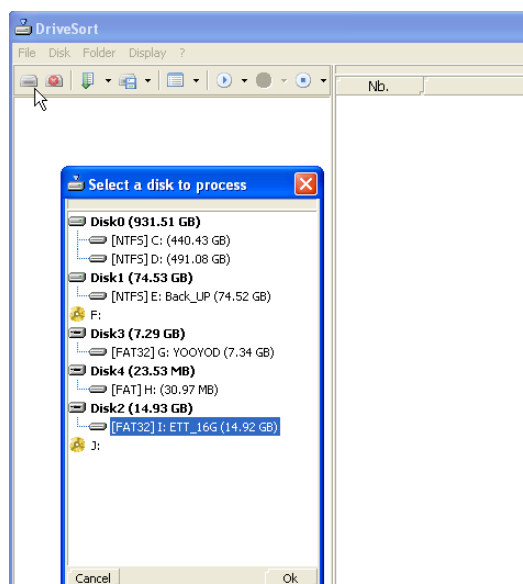
- 1.1 เป็นบอร์ดสำหรับเล่นไฟล์ MP3 หรือ WAV เท่านั้น ไม่สามารถ Record เสียงได้
- 1.2 ใช้ Micro SD Card เป็นตัวเก็บไฟล์เสียง ขนาดความจุที่รองรับ 2 GB-16 GB รองรับระบบ Flash 16 และ 32
- 1.3 ใช้ไฟเลี้ยงบอร์ด DC +5 V และ ให้สัญญาณเสียงทาง Output แบบ Stereo
- 1.4 เลือกโหมดการทำงานได้ 4 โหมด คือ 1.) COMMAND MODE : ควบคุมการเล่นด้วย Command ผ่านทาง RS232
2.) LIST_SONG MODE : ใช้เรียกดูรายชื่อและลำดับไฟล์เสียงใน Micro SD Card 3.) MP3_SW MODE : ควบคุมการเล่นด้วยสวิทช์ 3 ตัว บนบอร์ด 4.) TRIG MODE : ควบคุมการเล่นด้วยการ Trig ที่ขารับสัญญาณ Trig โดยเลือก Trig ได้ 16 ลำดับเพลง (ดูวิธีการ Trig ได้ในหัวข้อ TRIG MODE)
- 1.5 สามารถเล่นไฟล์เสียงได้ทั้งที่อยู่ภายในและภายนอก Folder
- 1.6 สามารถเรียกดูรายชื่อ และลำดับของไฟล์ ใน Micro SD Card ได้ด้วย DIP-SW_MODE หรือใช้การส่ง Command โดยจะแสดงผลผ่านทาง RS232 ด้วยโปรแกรม Terminal หรือโปรแกรมอื่นๆที่รองรับการเชื่อมต่อแบบ RS232
- 1.7 มี LED แสดงสถานะ ขณะเล่น , การ Trig และ การสิ้นสุดไฟล์(EOF)
- 1.8 ทุกครั้งที่การเล่นไฟล์จบลงจะมีการส่ง Command End of File ออกมาทาง RS232 และส่งเป็นพัลส์ (Logic'0') ความกว้าง 20 ms ออกมาที่ขาต่อ EOF ของบอร์ด เพื่อให้ผู้ใช้ตรวจสอบสถานะการจบของ ไฟล์ได้
- 1.9 ในโหมด COMMAND สามารถกำหนด Baud Rate ได้ 4 ค่า คือ 4800 , 9600 , 57600 และ 115200 Baud/sec
- 1.10 ในโหมด COMMAND สามารถสั่งเล่นเพลงได้โดยลำดับหมายเลข (ไม่เกิน 99 ลำดับ) หรือ โดยชื่อ ,สั่งหยุดเล่นแบบ STOP หรือ PAUSE ได้ , สั่งปรับ Volume ได้ , สั่งให้เล่นซ้ำ 1 เพลง หรือ เล่นซ้ำทั้งหมด หรือ เล่นเพลงเดียวจบ หรือ เล่นเพลงทั้งหมด 1 รอบแล้วจบได้ , สั่งให้เดินหน้า(FW) หรือถอยหลัง(RV) ครั้งละ 1 เพลงได้ , สั่งให้แสดงเวลาขณะเล่น แสดง ชื่อ และลำดับไฟล์เสียงทั้งหมดใน Micro SD Card ออกทาง RS232 ได้
- 1.11 ในโหมด COMMAND การส่งคำสั่ง Play ออกไปแต่ละครั้งเราสามารถกำหนดได้ว่า จะให้คำสั่ง Play มีผลทันที ในการเล่นเพลงต่อไปได้เลย หรือ รอให้เพลงที่เล่นอยู่จบลงก่อนแล้วเพลงที่ส่งมาใหม่ถึงจะเริ่มเล่น
- 1.12 ในโหมด MP3_SW. สามารถสั่ง Play , STOP , PAUSE , FW , RW , ปรับ Volume ได้ด้วย SW. 3 ตัว บนบอร์ด
- 1.13 ในโหมด MP3_SW. สามารถใช้ DIP-SW_CONF กำหนดรูปแบบการเล่นได้คือ เล่นเพลงทั้งหมด 1 รอบแล้วจบ หรือ เล่นซ้ำทั้งหมด และเลือกได้ว่าจะให้เล่นทันทีเมื่อ Power ON (Auto) หรือ ให้เล่นเมื่อมีการกด SW-Play ที่บอร์ด
- 1.14 ในโหมด TRIG สามารถสั่ง STOP และปรับ Volume จาก SW. 3 ตัวบนบอร์ดได้
- 1.15 ในโหมด TRIG สามารถสั่งเล่นเพลงได้ตั้งแต่เพลงลำดับที่ 1-16 ด้วยการให้ขารับสัญญาณ Trig ของตำแหน่งเพลงที่ต้องการจะเล่น Trig ลงกราวด์แล้วปล่อย เพลงนั้นก็จะถูกเล่น
- 1.16 ในโหมด TRIG สามารถใช้ DIP-SW_CONF กำหนดรูปแบบการเล่นได้คือให้เล่น 1 เพลงแล้วจบ หรือ เล่นซ้ำ 1 เพลง และเลือกได้ว่าจะให้เพลงที่ถูก Trig นั้นเล่นทันทีหรือรอให้เพลงที่เล่นอยู่จบก่อนแล้วเพลงที่ถูก Trig ถึงจะถูกเล่น
- 1.17 ทุกครั้งที่บอร์ดถูก Reset ค่า Volume จะถูก Set ไปเป็น Default และจะไม่จำการทำงานที่ค้างอยู่ก่อนจะถูก Reset
- 1.18 การปรับ DIP-SW_MODE และ DIP-SW_CONF จะมีผลเมื่อบอร์ดถูก Reset


2. คุณสมบัติและการจัดการ File สำหรับ Micro SD Card

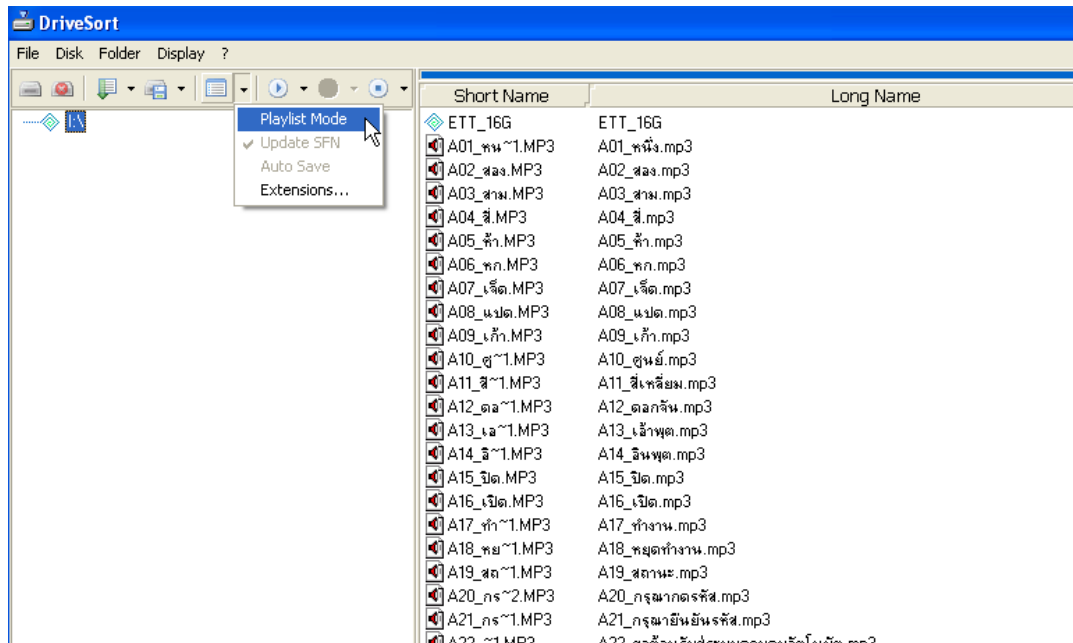
- 2.1 การตั้งชื่อไฟล์ หรือ Folder สามารถตั้งได้ทั้งภาษาไทย และอังกฤษ และสามารถใช้ได้ทั้งตัวพิมพ์ใหญ่ และตัวพิมพ์เล็ก แต่ห้ามใช้ตัวอักษรที่เหมือนกับ Command ได้แก่ \$ * # @ & =
- 2.2 ไฟล์เพลงจะต้องเป็นนามสกุล .MP3 หรือ .WAV เท่านั้นซึ่งใช้ได้ทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็กเช่นกัน
- 2.2 เวลาผู้ใช้เรียกดู ชื่อ-นามสกุล ไฟล์ จะแสดงเป็นตัวพิมพ์ใหญ่เสมอสำหรับชื่อไฟล์ที่เป็นภาษาอังกฤษ ส่วนชื่อไฟล์ที่เป็นภาษาไทยก็จะแสดงตัวอักษรในลักษณะที่อ่านไม่รู้เรื่อง และจะแสดงให้เห็นได้ไม่เกิน 8 ตัวอักษร(ในกรณีชื่อไฟล์ยาวเกิน8ตัวอักษร ตามรูปแบบ 8.3) ผ่านช่องทาง RS232 และแสดงผลด้วยโปรแกรม Terminal หรือโปรแกรมอื่นๆที่รองรับการ Interface RS232
- 2.3 ในกรณีที่ผู้ใช้ Command **สั่งเล่นไฟล์โดยชื่อ ให้ใช้ ชื่อ-นามสกุล ไฟล์ที่จะสั่งให้เล่น เป็นตัวพิมพ์ใหญ่เสมอ** ไม่ว่าชื่อหรือนามสกุล ไฟล์ที่เราตั้งไว้จะเป็นตัวพิมพ์ใหญ่ หรือ ตัวพิมพ์เล็กก็ตาม
- 2.4 Micro SD Card ที่จะนำมาใช้เก็บไฟล์ จะต้อง format เป็น FAT16 หรือ FAT32 เสียก่อน
- 2.5 การ Copy ไฟล์เพลงจาก PC ลงใน Micro SD Card ถ้าผู้ใช้ Copy ครั้งละไฟล์ ไฟล์เพลงที่ถูก Copy มาวางใน Card เป็นไฟล์แรกก็จะมีลำดับไฟล์เป็นลำดับที่1 ไฟล์ต่อไปที่ถูก Copy ก็จะมีลำดับเป็นลำดับ ที่ 2 , 3 , 4 ไปเรื่อยๆ โดยลำดับไฟล์นี้จะนำไปใช้ในการส่งคำสั่ง เล่นไฟล์โดยลำดับ
- 2.6 การ Copy ไฟล์เสียงจาก PC ลงใน Micro SD Card ถ้าผู้ใช้ Copy แบบ Copy All คือ Copy ไฟล์ทั้งหมด ลงใน Card ครั้งเดียวเลย ผู้ใช้จะไม่รู้ว่าไฟล์ไหนถูกวางลงใน Card เป็นไฟล์แรก หรือ ไฟล์ที่ 2 , 3 , 4... ดังนั้นเวลาสั่งเล่นไฟล์ด้วยคำสั่งเล่นไฟล์โดยลำดับ ผู้ใช้จะไม่รู้ว่าไฟล์ที่สั่งให้เล่นนั้นใช้ไฟล์เพลงที่ต้องการจะให้เล่นหรือไม่ ดังนั้นในกรณีนี้ผู้ใช้สามารถใช้ DIP-SW_Mode หรือ ใช้คำสั่งดู List File เพื่อดูลำดับ และ รายชื่อ-นามสกุล ของไฟล์เพลงทั้งหมดใน Card ผ่านทาง RS232 ก่อนได้ เพื่อจะสั่งให้เล่นไฟล์โดยลำดับได้ถูกต้องตามที่ผู้ใช้ต้องการ หรือจะใช้โปรแกรม Drive Sort ที่ให้มาใน CD จัดลำดับไฟล์ใหม่เพื่อให้ได้ไฟล์เรียงลำดับตามที่เราต้องการ โดยมีวิธีใช้งานดังนี้



การใช้โปรแกรม Drive Short เรียงลำดับไฟล์ใน Micro SD Card

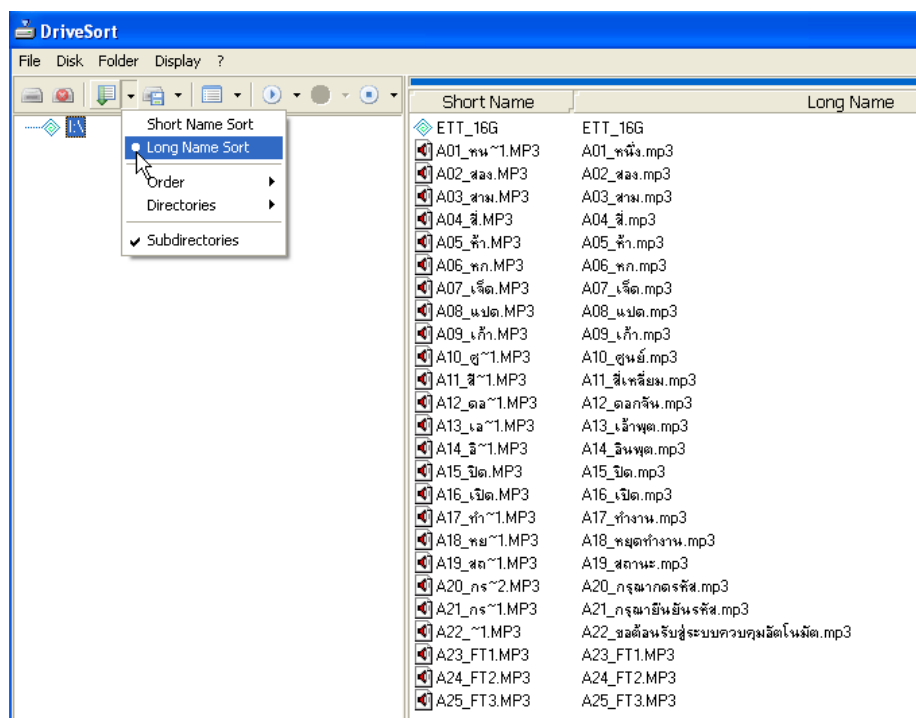
- 1) Copy File เสียงที่ต้องการลง Micro SD Card ให้เรียบร้อยแล้ว
- 2) Run โปรแกรม DriveShort .exe ขึ้นมา แล้วคลิกที่ Icon 'Open a disk' [] จะปรากฏหน้าต่าง 'Select a disk To process' ขึ้นมาดังรูปด้านล่าง




- 3) Double Click ไดรฟ์ ที่ต่อ Micro SD Card ไว้ หน้าต่างด้านขวามือก็จะแสดงรายชื่อไฟล์เสียงขึ้นมา แล้ว Click ที่ ลูกศรสีดำด้านขวาของ Icon 'Playlist' [] แล้วเอาเครื่องหมายถูกหน้า 'Playlist Mode' ออกเพื่อให้หน้าต่างด้านขวามือแสดงไฟล์ทั้งหมดใน Card ดังรูปด้านล่าง โดยใน ช่อง Long Name จะแสดงชื่อไฟล์แบบเต็ม ส่วน ในช่อง Shot Name จะแสดงชื่อไฟล์แบบสั้น



- 4) Click ลูกศรสีดำด้านขวาของ Icon 'Short current folder' [] แล้วเลือกที่ Long Name Shot เพื่อจัดเรียงไฟล์ตามชื่อในช่อง 'Long Name' ซึ่งเป็นชื่อที่เรามองเห็นจริงตามใน Card เสร็จแล้วให้คลิกที่ Icon 'Short current folder' [] จะเห็นว่าในช่อง 'Long Name' รายชื่อไฟล์จะถูกเรียงลำดับตามตัวอักษรใหม่ โดยไฟล์แรกที่อยู่บนสุดก็จะ เป็นไฟล์ลำดับที่ 1 ไหลลงมาเรื่อยๆ ดังรูปด้านล่าง



5) ทำการ Save การจัดเรียงไฟล์โดย Click ที่ Icon ‘Save current folder’ [] เป็นอันเสร็จสิ้นการจัดเรียงไฟล์ปิดโปรแกรมแล้วนำ Micro SD Card มาใส่ลงใน บอร์ด ET-REMOTE MP3.

6) เทคนิคการจัดเรียงไฟล์ด้วยโปรแกรม DriveShort อยู่กับการตั้งชื่อไฟล์ โดยก่อนที่จะ Copy ไฟล์ลง Card ควรแก้ไขชื่อไฟล์ให้มีลักษณะดังนี้

- ถ้าตั้งชื่อไฟล์เป็นตัวเลขอย่างเดียวให้ใช้จำนวนหลักเท่ากัน เช่น 01.mp3,12.mp,99.mp3 เป็นต้น เวลา

โปรแกรมเรียงก็จะเรียงไฟล์ชื่อ 01.mp3 ไว้ในลำดับที่1ไล่ไปเรื่อยๆตามชื่อ อย่าใส่ 1.mp3 อย่างเดียวให้เติม 0 ไปข้างหน้าด้วย ถ้าเป็นตัวเลข 3 หลักก็เพิ่ม 0 เข้าไป 2 ตัว เป็นต้น

- ถ้าตั้งชื่อไฟล์เป็นตัวอักษรควรนำด้วยตัวอักษรภาษาอังกฤษที่มีจำนวนตัวอักษรเท่ากันให้เหมือนกันทุกไฟล์แล้วตามด้วยตัวเลขที่มีจำนวนหลักเท่ากันส่วนตัวอักษรที่ต่อจากตัวเลขถ้ามีจะเป็นอะไรก็ได้ เช่น FM01.mp3,

FM02K.MP3, FM03.mp3 , ...FM45.mp3 เป็นต้น เวลาโปรแกรมเรียงก็จะเรียงจาก FM01-FM45 ให้เราตามลำดับ

2.7 หลังจาก Power ON หรือ Reset การสั่งเล่นไฟล์เพลงใน Mode MP3_SW หรือ ใน Mode COMMAND ด้วยคำสั่ง Play All ไฟล์เพลงใน Micro SD Card ลำดับที่ 1 จะถูกเล่นเป็นไฟล์แรกเสมอ และตามด้วยลำดับที่ 2 , 3 , 4.. ไปเรื่อยๆ

3. ลักษณะของบอร์ด ET-REMOTE MP3. V2

1. LED-PWR : LED แสดงสถานะ Power Supply 5V ที่จ่ายเลี้ยงบอร์ด

2. LED-TRIG : LED แสดงสถานะการ Trig ของขั้วรับสัญญาณ Trig หมายเลข 19 โดย LED จะติดตามจังหวะของสัญญาณที่เข้ามา Trig (Trig Active ‘0’) เมื่อมีการใช้งานใน TRIG MODE

3. LED-ACT : LED แสดงสถานะการเล่นไฟล์เพลง โดย LED จะกระพริบในขณะที่ไฟล์เพลงกำลังถูกเล่นอยู่

4. SOCKET-Micro SD Card : เป็น Socket สำหรับใส่ ตัว Micro SD Card ที่ใช้เก็บไฟล์เพลง โดยก่อนที่จะถอดหรือใส่ Card จะต้องถอดไฟเลี้ยงออกจากบอร์ด MP3ก่อนเสมอ มีวิธีการใส่แสดงดังรูปด้านล่าง

1) ดันฝา Socket มาทางซ้ายมือดังรูป

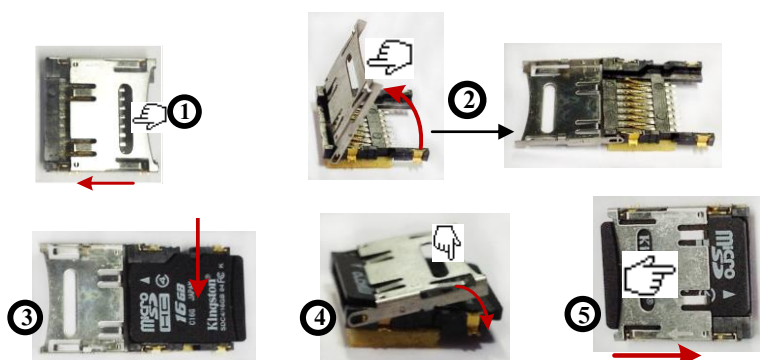
2) จัดฝา Socket ขึ้นมา ดังรูป

3) ใส่ Micro-SD Card ดังรูป

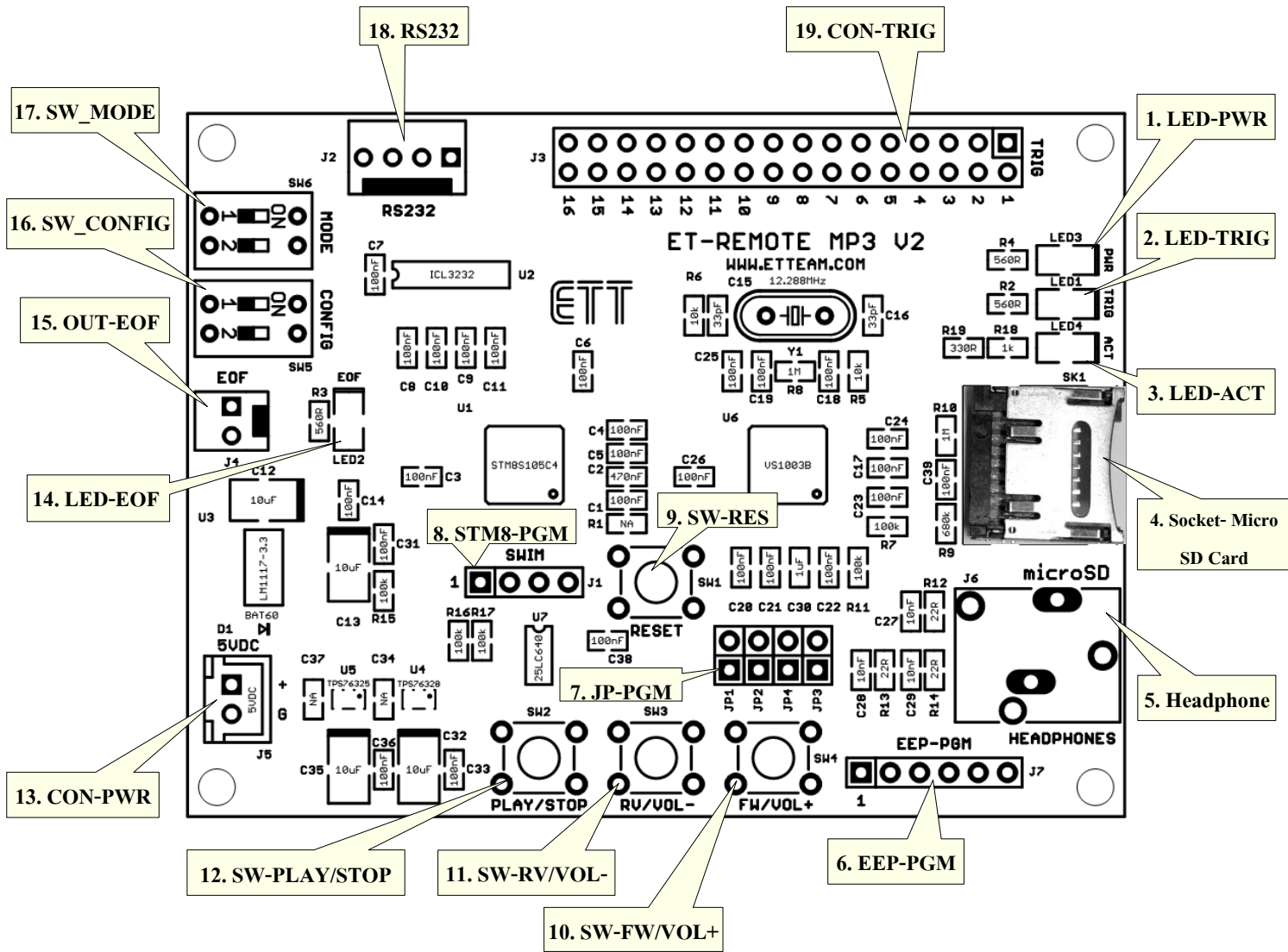
4) ปิดฝา Socket ดังรูป

5) กดที่ฝา Socket พอประมาณแล้วดันฝา

Socket มาทางขวาเพื่อ Lock Card ดังรูป

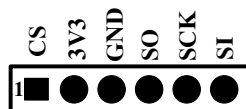


คำเตือน! ทุกครั้งที่ใส่หรือถอด Micro SD Card จากบอร์ด MP3 จะต้องถอดแหล่งจ่ายไฟเลี้ยงออกจากบอร์ด MP3 ก่อนเสมอ



รูปแสดงโครงสร้างของบอร์ด ET-REMOTE MP3. V2

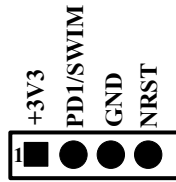
- 5. Head-Phone :** เป็นจุดต่อสัญญาณเสียง Output (Stereo) ออกไปใช้งาน โดยถ้าเป็นชุดหูฟังสามารถนำมาต่อฟังได้ทันที แต่ถ้าเป็นลำโพงจะต้องผ่านชุดขยายเสียง อย่างนำมาต่อโดยตรง จะทำให้ Chip MP3 เสียหาย
- 6. EEP-PGM :** เป็นขั้วต่อสำหรับ Download Firmware ให้กับ E2PROM ซึ่งมีไว้ใช้งานสำหรับทางบริษัทในการ Upload Firmware จะไม่เกี่ยวข้องกับการใช้งานบอร์ดของผู้ใช้



รูปแสดงขาสัญญาณขั้วต่อ EEP-PGM

- 7. JP-PGM :** เมื่อจะ Load Firmware ลง EEPROM จะต้องถอดตัว Jumper1-4 ออกเสมอ หลังจาก Load Firmware เสร็จเรียบร้อยให้ Jump ตัว Jumper1-4 กลับเข้าที่ เพื่อให้บอร์ด MP3 ใช้งานได้ตามปกติ ในส่วน Jumper นี้จะไม่เกี่ยวข้องกับผู้ใช้งานโดยตรงจะเป็นส่วนใช้งานของทางบริษัท ดังนั้นเวลาใช้งานบอร์ด MP3 ผู้ใช้จะต้องตรวจสอบว่ามีการ jump ตัว Jumper1-4 ไว้เรียบร้อยแล้วเสมอ มิฉะนั้นบอร์ด MP3 จะไม่ทำงาน

8. **STM8-PGM** : เป็นขั้วต่อสำหรับ Download Firmware ให้กับตัว STM8 ซึ่งมีไว้ใช้งานสำหรับทางบริษัทในการ Upload Firmware จะไม่เกี่ยวข้องกับ การใช้งานบอร์ดของผู้ใช้เช่นกัน



รูปแสดงขาสัญญาณขั้วต่อ STM8-PGM

9. **SW-RES** : เป็น SW. Reset บอร์ดให้เริ่มทำงานใหม่ เมื่อเกิดกรณีบอร์ดแฉก หรือ ใช้ Update สถานะการเปลี่ยนแปลงของ DIP-SW_MODE หรือ DIP-SW_CONFIG
10. **SW-FW/VOL+** : เมื่อใช้งานใน MP3_SW Mode สวิตช์นี้จะมี 2 หน้าที คือ ใช้สำหรับเลื่อนเพลงไปข้างหน้าครั้งละ 1 เพลง โดยการกดแล้วปล่อย SW. ปกติ ส่วนอีกหน้าที่หนึ่งจะใช้เป็น SW. เพิ่มความดังเสียงโดยการกด SW. ค้างไว้ให้เกิน 0.5 วินาที แล้วปล่อย ก็จะเป็นการเพิ่มความดังเสียง เมื่อใช้งานใน Trig Mode สวิตช์นี้จะทำหน้าที่เพิ่มความดังเสียงเท่านั้น โดยกดแล้วปล่อยปกติ
11. **SW-RV/VOL-** : เมื่อใช้งานใน MP3_SW Mode สวิตช์นี้จะมี 2 หน้าที คือ ใช้สำหรับเลื่อนเพลงถอยหลังครั้งละ 1 เพลง โดยการกดแล้วปล่อย SW. ปกติ ส่วนอีกหน้าที่หนึ่งจะใช้เป็น SW. ลดความดัง เสียงโดยการกด SW. ค้างไว้ให้เกิน 0.5 วินาที แล้วปล่อย ก็จะเป็นการลดความดังเสียง เมื่อใช้งานใน Trig Mode สวิตช์นี้จะทำหน้าที่ลดความดังเสียงเท่านั้น โดยกดแล้วปล่อยปกติ
12. **SW-PLAY/STOP** : เมื่อใช้งานใน MP3_SW Mode สวิตช์นี้จะมี 3 หน้าที คือ 1) ทำหน้าที่สั่ง Play ไฟล์เสียง โดยในขณะที่ไฟล์เสียงยังไม่ถูกเล่นเมื่อเรากดและปล่อย SW. ไฟล์เสียงก็จะถูกเล่น 2) ทำหน้าที่สั่ง STOP ไฟล์เสียงที่กำลังเล่นอยู่ โดยในขณะที่ไฟล์เสียงกำลังเล่นอยู่ เมื่อเรากดและปล่อย SW. การเล่นก็จะหยุดลง และเมื่อเรากดและปล่อย SW นี้อีกครั้ง ก็จะเป็นการสั่ง Play ไฟล์เสียงก็จะถูกเริ่มเล่นใหม่ตั้งแต่ไฟล์แรกอีกครั้ง 3) ทำหน้าที่สั่ง PAUSE ไฟล์เสียงที่กำลังเล่นอยู่ โดยในขณะที่ไฟล์เสียงกำลังเล่นอยู่ เมื่อเรากด SW. ค้างไว้ให้เกิน 0.5 วินาที แล้วปล่อยการเล่นก็จะหยุดลงชั่วคราวและเมื่อเรากดและปล่อย SW นี้อีกครั้ง ก็จะเป็นการสั่ง Play ไฟล์เสียงก็จะถูกเล่นต่อจากตำแหน่งที่หยุดอีกครั้ง เมื่อใช้งานใน Trig Mode สวิตช์นี้จะทำหน้าที่ STOP เพลงที่เล่นอยู่ โดยกดแล้วปล่อยปกติ

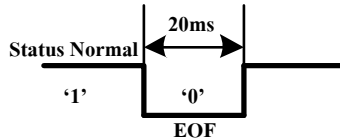
13. **CON-PWR** : เป็นขั้วต่อรับไฟเลี้ยง DC 5 V เพื่อเลี้ยงบอร์ด MP3 โดยจะต้องคำนึงถึงการต่อขั้วไฟให้ถูกต้องด้วย



รูปแสดงขั้วต่อไฟเลี้ยงบอร์ด

14. **LED-EOF** : LED แสดงสถานะของสัญญาณ End Of File (EOF) โดยทุกครั้งที่ File ถูกเล่นจบลง หรือ ถูกสั่งงานด้วยคำสั่ง Forward , Reverse สัญญาณ EOF ก็จะเกิดขึ้นช่วงเวลาหนึ่ง ทำให้ LED ติดและดับลงตามจังหวะสัญญาณที่เกิดขึ้น

15. OUT-EOF : เป็นขั้วต่อ Output สำหรับส่งสัญญาณ EOF ออกมาให้ผู้ใช้ไว้ใช้สำหรับตรวจสอบการจบของไฟล์ที่เล่นอยู่ โดยสัญญาณนี้จะเกิดขึ้นเมื่อ File ถูกเล่นจบลง หรือ ถูกสั่งงานด้วยคำสั่ง Forward , Reverse ลักษณะของสัญญาณในสถานะปกติที่ขา EOF จะเป็น logic 1 (3.3V) และเมื่อมีสัญญาณ EOF เกิดขึ้น สัญญาณที่ขา EOF ก็จะตกลงเป็น Logic 0 เป็นเวลา 20 ms และกลับมาเป็น Logic 1 ตามเดิม สรุปก็คือทุกครั้งที่มีไฟล์เสียงถูกเล่นจบลง ตัวบอร์ด MP3 ก็จะส่งสัญญาณพัลส์ EOF Logic 0 ที่มีความกว้าง 20 ms ออกมาให้ผู้ใช้ที่ขั้วต่อนี้



รูปแสดงลักษณะสัญญาณ EOF



รูปแสดงขั้วต่อสัญญาณ EOF

16. SW_CONFIG : เป็น DIP SW. ใช้กำหนด Baud Rate ในการรับ-ส่งคำสั่งผ่าน RS232 เมื่อเลือกใช้งานบอร์ด MP3. ใน COMMAND MODE , และเป็น DIP SW. ใช้กำหนดคุณสมบัติในการเล่นไฟล์เสียงในลักษณะต่างๆเมื่อเลือกใช้งานบอร์ด MP3. ใน MP3_SW Mode หรือ TRIG Mode โดยการเปลี่ยนแปลง DIP SW. ในแต่ละครั้งจะมีผลเมื่อบอร์ดถูก Reset แล้วเท่านั้น ดังนั้นทุกครั้งที่มีการ Set DIP SW. ให้ทำการ Reset บอร์ด MP3 เพื่อ Up Date ผลของการเปลี่ยนแปลง ส่วนตารางการ Set DIP-SW CONFIG ให้ดูที่หัวข้อการใช้งานของแต่ละโหมดการทำงาน

17. SW_MODE : เป็น DIP SW. ใช้เลือกโหมดการทำงาน โดยเวลาจะใช้งานบอร์ดผู้ใช้จะต้อง Set DIP-SW_MODE เพื่อเลือกโหมดการทำงานเป็นอันดับแรก โดยจะมีด้วยกัน 4 โหมดการทำงาน ให้เลือกตามตารางด้านล่าง หลังจากที่ Set DIP-SW_MOD เรียบร้อยแล้วให้ทำการ Reset บอร์ด MP3 ใหม่เสมอ เพื่อ Up Date ผลการ Set มิฉะนั้น การ Set DIP SW. จะยังไม่มีผลใดๆเกิดขึ้นกับการทำงานของบอร์ด

ตารางการ Set DIP SW_MODE

DIP-SW_MODE		รูปการ SET SW_MODE	Operation MODE
S1	S2		
ON (0)	ON (0)		เลือกทำงานใน COMMAND MODE เป็นการส่งคำสั่งผ่าน RS232 และกำหนด Baud Rate ในการรับ-ส่งได้จาก DIP-SW_CONFIG
OFF (1)	ON (0)		เลือกทำงานใน LIST_SONG MODE เป็นการเรียกดูรายชื่อและลำดับไฟล์เสียงทั้งหมดที่มีใน Micro SD Card ผ่านทาง RS232 แสดงผลด้วยโปรแกรม Terminal
ON (0)	OFF (1)		เลือกทำงานใน MP3_SW MODE เป็นการควบคุมการเล่นผ่าน SW. 3 ตัวที่อยู่บนบอร์ด และกำหนดรูปแบบการเล่นได้จาก DIP-SW_CONFIG
OFF (1)	OFF (1)		เลือกทำงานใน TRIG MODE เป็นการสั่งเล่นไฟล์เสียงด้วยการรับสัญญาณ Trig จากภายนอกเข้ามาสั่ง Play และกำหนดรูปแบบการเล่นได้จาก DIP-SW_CONFIG

18. RS232 : เป็น Connector สำหรับต่อสายสัญญาณ RS232 โดยบนบอร์ดจะต่อ IC Line Driver 232 ไว้ให้แล้ว ดังนั้นสามารถนำไปต่อเข้ากับ Port 232 ของ PC ได้เลย แต่ถ้านำไปต่อกับ MCU Port 232 ของ MCU อีกฝั่งหนึ่งจะต้องมี IC Line Driver 232 อยู่ด้วยถึงจะสื่อสารกันได้ ซึ่งข้อนี้จะใช้เมื่อต้องการเรียกดูรายชื่อและลำดับไฟล์เสียงใน Micro SD Card แสดงผลด้วยโปรแกรม Terminal ต่างๆ หรือใช้ในการส่งคำสั่งควบคุมการเล่นไฟล์เสียงใน Command Mode ด้วย PC หรือ MCU โดยให้ Set คุณสมบัติการสื่อสารดังนี้

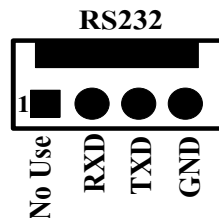
BAUD RATE : 4800 , 9600 , 5700 , 115200 เลือกรายการจาก DIF-SW_CONFIG

DATA BIT : 8

PARITY : NONE

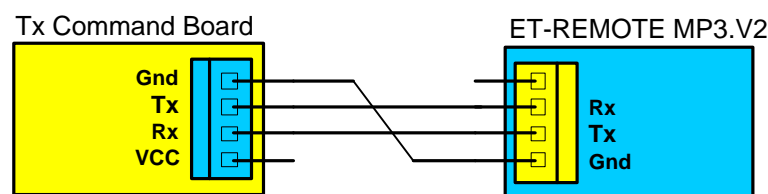
STOP BIT : 1

HANDSHAK : NONE



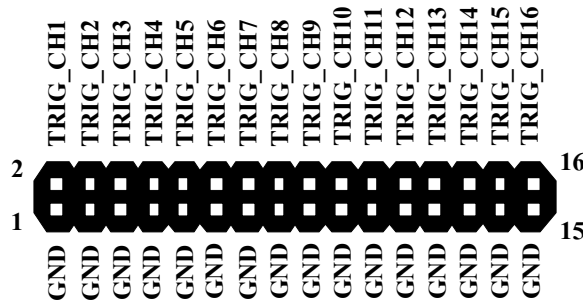
รูปแสดงขั้วต่อสัญญาณ RS232

การต่อสาย RS232 แสดงดังรูปด้านล่าง



รูปแสดงการต่อสาย RS232 จากอุปกรณ์ส่ง Command ไปยัง ET-REMOTE MP3.V2 Board

19. CON-TRIG : เป็นขั้วต่อรับสัญญาณ TRIG จากภายนอกขนาด 16 ช่องสัญญาณ ซึ่งจะใช้งานใน TRIG MODE โดย Pin ซีกทางด้านในของบอร์ดจะถูก Pull Up ไว้ที่แรงดัน 3.3V ทำหน้าที่เป็น Input 16 ช่องเพื่อรับสัญญาณ Trig จากภายนอก หรือบนบอร์ด ส่วน Pin ซีกด้านนอกของบอร์ดจะเป็น GND ทั้งหมดเพื่อเอาไว้ทดสอบการ Trig โดยการนำ Pin ที่เป็นคู่ตรงกัน ช็อตกันแล้วปล่อย จะทำให้เกิดสัญญาณ Trig ขึ้น ไฟล์เสียงในลำดับที่ตรงกับช่องที่มีสัญญาณ Trig เข้ามาก็จะถูกเล่น โดยข้อนี้จ้ะรับสัญญาณ Trig Logic '0' (Active '0') ที่มีความกว้าง 20 ms ขึ้นไป และในสภาวะที่สัญญาณ Trig เป็น Logic '1' ระดับแรงดันของสัญญาณจะต้องไม่เกิน 3.3 V เท่านั้น เนื่องจากขั้วต่อ Input นี้จะรองรับระดับแรงดันได้ไม่เกิน 3.3V ในกรณีที่สัญญาณ Trig ในสภาวะที่เป็น Logic '1' เกิน 3.3V เช่น 5V จะต้องใช้ วงจรลดระดับแรงดันให้เหลือ 3.3 V มาต่อคั่นกลางเสียก่อน ซึ่งอาจจะใช้โปรดัคส์ของอิทีที เช่น ET- MINI 3To5TTL (ต่อได้ 8 แชนแนล) หรือ ET-BUSIO-DCIN (ต่อได้ 1 แชนแนล) สามารถดูรายละเอียดการใช้งานเพิ่มเติมได้ในหัวข้อ TRIG MODE



รูปแสดงขั้วรับสัญญาณ TRIG

คำเตือน! สัญญาณที่ใช้ Trig ในสถานะที่เป็น Logic '1' จะต้องมียกระดับแรงดันไม่เกิน 3.3 V เท่านั้น

4. การใช้งานบอร์ด ET-REMOTE MP3 V2.

4.1 COMMAND MODE

4.1.1) การทำงาน

การทำงานในโหมดนี้จะใช้วิธีการส่ง Command จากภายนอกเข้ามาผ่านทาง RS232 เพื่อควบคุมการเล่นของไฟล์เสียง ซึ่งการส่ง Command นั้นจะใช้คอนโทรลเลอร์ตระกูลใดๆเป็นตัวส่งก็ได้ หรือจะส่งผ่านทางโปรแกรม Terminal ต่างๆด้วยวิธีการพิมพ์จาก Keyboard ก็ได้

เมื่อบอร์ดเริ่มทำงานตัวบอร์ดจะพิมพ์ข้อความ **ET-REMOTE MP3 V2 (Firmware V2.1) = #OK** ออกมาทาง RS232 เพื่อบอกให้ผู้ใช้ทราบว่าพร้อมรับคำสั่งแล้ว โดยบอร์ดที่เป็นตัวส่งคำสั่งนั้นสามารถอ่าน Respond นี้เพื่อนำไปใช้ตรวจสอบความพร้อมของ Board MP3. ก่อนที่จะทำการส่งคำสั่ง หรืออาจจะ delay ไว้ประมาณ 2 – 3 วินาทีก็ได้ เพื่อให้บอร์ด ET-REMOTE MP3 ได้ Initial ตัวเองให้เรียบร้อยพร้อมรับคำสั่งเสียก่อน

เมื่อผู้ใช้ทำการส่ง Command ให้กับบอร์ดและคำสั่งถูกกระทำแล้วตัวบอร์ดก็จะยิง Echo ซึ่งมีรูปแบบคือ **#OK (3Byte)** กลับออกมาให้ เมื่อ Command ที่ส่งนั้นถูกต้อง ถ้าคำสั่งที่ส่งไปเป็นคำสั่ง Play File ก่อนที่ File จะถูกเล่นในแต่ละไฟล์ ตัวบอร์ดจะส่งลำดับและชื่อนามสกุลไฟล์ที่จะถูกเล่นออกมาให้เสมอ โดยมีรูปแบบคือ **\$NN.FFFF.MP3 (หรือ .WAV)** เมื่อ NN = ลำดับไฟล์(2Byte) , FFFF = ชื่อไฟล์ (ไม่เกิน 8 Byte) ต่อไปเมื่อไฟล์เสียงที่เล่นอยู่จบลง หรือมีการส่งคำสั่งเช่น STOP, Forward, Revers เป็นต้นเข้ามาขัดจังหวะทำให้การเล่นไฟล์นั้นต้องจบลง ตัวบอร์ดก็จะส่งสัญญาณ EOF ซึ่งมีรูปแบบคือ **&E (2 Byte)** กลับออกมาให้ผู้ใช้เพื่อบอกให้รับทราบว่า ไฟล์ที่เล่นอยู่จบแล้ว พร้อมทั้งจะรับคำสั่งใหม่หรือเล่นไฟล์ต่อไป โดยสัญญาณ EOF นอกจากจะส่งออกมาทาง RS232 แล้ว ก็จะส่งเป็นพัลส์ 0 ความกว้าง 20 ms ออกไปยังขั้วต่อหมายเลข15(OUT -EOF) ด้วย สามารถสังเกตการเกิดสัญญาณ EOF ได้จาก LED -EOF ที่อยู่บนบอร์ด ถ้าผู้ใช้มีการส่งคำสั่งให้แสดงเวลาการเล่นด้วย เมื่อไฟล์ถูกเล่นตัวบอร์ดก็จะมีการส่งค่าเวลาการเล่นไฟล์ออกมาตลอดระยะเวลาของการเล่นไฟล์ โดยรูปแบบของเวลาที่ส่งออกมาก็จะเป็นดังนี้ **@mm:ss** เมื่อ mm = นาที(2Byte) , ss = วินาที(2Byte)

ข้อควรจำ ในการเริ่มต้นส่ง Command ให้กับบอร์ด MP3 นั้นเพื่อให้การรับ Command และ การส่ง Command สอดคล้องกัน(Sync) ในบอร์ดที่เป็นตัวส่ง Command ควรจะ delay ไว้ประมาณ 200-500 ms ขึ้นไปแล้วจึงทำการส่ง Command แรกให้กับบอร์ดได้ เพื่อให้บอร์ด MP3. อยู่ในสถานะพร้อมรับข้อมูลก่อนเมื่อเริ่มต้น Power up และในกรณีที่ต้องการส่ง Command ติดกัน ควรจะ delay คั่นระหว่าง Command ด้วย ถ้าในระหว่าง Command ไม่ได้มีการตรวจสอบ Echo Command จากบอร์ด MP3 ที่ส่งกลับมาให้เพื่อที่จะให้บอร์ด MP3 กระทำ Command แรกเรียบร้อยแล้ว

4.1.2) รูปแบบคำสั่งและ Respond ที่ใช้กับ ET-REMOTE MP3 V2.

รูปแบบของ Command ที่ใช้นั้นจะอยู่ในรูปของ ASCII Code ทั้งหมด ซึ่งสามารถแทนด้วยสัญลักษณ์ ASCII เช่น '*' หรือแทนด้วย Hex Code ซึ่งก็จะเท่ากับ 0x2A , '0'=0x30 , 'A' = 0x41 เป็นต้น ในกรณีที่ส่งเป็นตัวอักษร A-Z จะต้องส่งเป็นตัวพิมพ์ใหญ่เท่านั้น ไม่รองรับตัวพิมพ์เล็ก หมายเลขลำดับไฟล์ หรือ ชื่อ-นามสกุลไฟล์ที่จะใช้แทนลงใน Command การเล่นนั้น (คำสั่ง P1,P2) สามารถเรียกดูได้จากการใช้ DIP-SW_MODE (ดูการใช้งานในหัวข้อ LIST_SONG MODE) หรือใช้คำสั่งเรียกดูชื่อไฟล์ ซึ่งหมายเลขลำดับ,ชื่อและนามสกุลไฟล์ที่แสดงให้เห็นจากการเรียกดูนั้น จะใช้สำหรับนำไปแทนลงใน Command การเล่น(P1,P2) โดย Command จะมีดังต่อไปนี้

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

เป็นคำสั่งเล่นไฟล์เสียงที่มีอยู่ใน Micro SD Card ทั้งหมดโดยจะเริ่มเล่นเพลงในลำดับที่1 ก่อน ซึ่งก็คือไฟล์ที่ Copy ลงใน Card เป็นไฟล์แรก รูปแบบคำสั่งแสดงในตาราง

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)

เป็นคำสั่งเล่นไฟล์เสียงด้วยหมายเลขลำดับของไฟล์ที่ผู้ใช้งานต้องการ โดยสามารถเลือกหมายเลขลำดับไฟล์ได้ตั้งแต่ 01-99 ซึ่งลำดับไฟล์นี้ต้องส่งเป็น ASCII ขนาด 2 Byte ลำดับไฟล์ที่ส่งจะอ้างอิงกับจำนวนไฟล์ที่มีอยู่จริงใน Card แต่จะสั่งเล่นได้ไม่เกินลำดับ99 ถึงแม้ใน Card จะมีจำนวนไฟล์เสียงเกินก็ตาม รูปแบบคำสั่งแสดงในตาราง

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)

เป็นคำสั่งเล่นไฟล์เสียงตามชื่อไฟล์ โดยชื่อไฟล์และนามสกุลจะต้องใช้ตัวพิมพ์ใหญ่เท่านั้น และชื่อไฟล์จะต้องไม่เกิน 8 Byte ส่วนนามสกุลจะต้องเป็น WAV หรือ MP3 เท่านั้น ถ้าสั่งเล่นด้วยคำสั่งนี้ชื่อไฟล์ควรเป็นภาษาอังกฤษเท่านั้น และในกรณีชื่อไฟล์เกิน 8 Byte ก็ให้ใช้คำสั่งเรียกดูชื่อไฟล์ หรือใช้ DIP-SW_MODE เรียกดูชื่อไฟล์ซึ่งจะแสดงรายชื่อไฟล์ที่เกิน 8 Byte แบบย่อให้เห็น รูปแบบคำสั่งแสดงในตาราง

Start	CMM		Mark1	FILE NAME.MP3/WAV (ชื่อไม่เกิน8Byte)	END
Byte1	Byte2	Byte3	Byte4	Byte5-16	End Byte
*	P	2	=	FFFFFFFF.MP3	0x0D (enter)

FFFFFFFF = ชื่อ ไฟล์ ASCII ตัวพิมพ์ใหญ่ ไม่เกิน 8 byte

Ex.

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

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

เป็นคำสั่งหยุดการเล่นไฟล์เสียงชั่วคราว(PAUSE) เมื่อต้องการให้กลับมาเล่นไฟล์เสียงในตำแหน่งที่หยุดไว้
อีกครั้งหนึ่ง ก็ให้ส่งคำสั่งนี้ซ้ำอีกครั้ง ในขณะที่การเล่นไฟล์ถูกหยุดด้วยคำสั่งนี้ คำสั่งที่ยังสามารถใช้งานได้ก็คือ
คำสั่งปรับ Volume ,คำสั่งกำหนดคุณสมบัติการเล่น ,คำสั่งแสดงชื่อไฟล์ที่กำลังเล่น และ คำสั่ง PU เพื่อกลับไปเล่น
ไฟล์เสียงต่อ รูปแบบคำสั่งแสดงในตาราง

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

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

เป็นคำสั่ง STOP การเล่น เพื่อหยุดเล่นไฟล์เสียงทั้งหมด ซึ่งจะออกจากคำสั่งที่สั่งเล่นไฟล์เสียงนั้นๆทันที
เพื่อมาอยู่ในสถานะเริ่มต้น พร้อมรับคำสั่งอื่นๆต่อไป รูปแบบคำสั่งแสดงในตาราง

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

- 6. COMMAND 'FW' (Forward)

เป็นคำสั่งสำหรับเลื่อนไฟล์เสียงไปข้างหน้าครั้งละ 1 ไฟล์ (FW) คือส่งคำสั่ง 1 ครั้งไฟล์ก็จะถูกเลื่อนไปข้างหน้า 1 ไฟล์ เมื่อส่งคำสั่ง FW เลื่อนไฟล์ไปจนถึงไฟล์สุดท้าย ไฟล์ก็จะเลื่อนวนกลับมาเริ่มต้นที่ไฟล์แรกอีก การเลื่อนนี้จะเป็นในลักษณะ Close loop การส่งคำสั่งให้ส่งขณะที่ไฟล์กำลังเล่นอยู่ รูปแบบคำสั่งแสดงในตาราง

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)

เป็นคำสั่งสำหรับเลื่อนไฟล์เสียงถอยหลังครั้งละ 1 ไฟล์ (RV) คือส่งคำสั่ง 1 ครั้งไฟล์ก็จะถูกเลื่อนถอยหลังมา 1 ไฟล์ เมื่อส่งคำสั่ง RV เลื่อนไฟล์ถอยหลังลงมาถึงไฟล์แรก ไฟล์ก็จะเลื่อนวนกลับมายังไฟล์สุดท้ายอีก การเลื่อนนี้จะเป็นในลักษณะ Close loop การส่งคำสั่งให้ส่งขณะที่ไฟล์กำลังเล่นอยู่ รูปแบบคำสั่งแสดงในตาราง

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)

Command นี้จะใช้สำหรับปรับความดังของเสียง (Volume) เมื่อต้องการปรับความดังเสียงในช่อง Volume (Byte5-6) ให้ใส่ ASCII (ตัวเลข) 2หลัก ตั้งแต่ 00 คือไม่มีเสียง ถึงค่า 99 คือเสียงดังสุด รูปแบบคำสั่งแสดงในตาราง

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

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

Ex.

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

- 9. COMMAND 'CF' (Configuration Play)

เป็นคำสั่งใช้กำหนดคุณสมบัติการเล่นไฟล์เสียง โดยควรกำหนดคุณสมบัติการเล่นไฟล์เสียงไว้ก่อนที่จะมีการสั่ง Play ไฟล์เสียง เพื่อเวลาสั่ง Play จะได้เห็นผลตามคุณสมบัติการเล่นที่กำหนดไว้ทันที โดยค่า Configuration จะมีขนาด 3 Byte แทนด้วย ASCII '0' และ '1' ซึ่งค่า Default จะถูกกำหนดไว้ที่ '000' โดยค่า Configuration ที่ผู้ใช้กำหนดเองนั้น จะจำอยู่จนกว่าบอร์ดจะถูก Reset หรือมีการส่งคำสั่ง CF ไป Set ค่าใหม่ คำสั่ง STOP จะไม่มีผลทำให้ค่า Configuration มีการเปลี่ยนแปลงใดๆ ดังนั้นเวลาเขียนโปรแกรมถ้า ต้องการเปลี่ยนแปลงคุณสมบัติการเล่นไฟล์ไปจากค่า Default ผู้ใช้ควรกำหนดคำสั่งนี้ไว้ที่ส่วนหัวของโปรแกรม เพื่อว่าทุกครั้งที่ บอร์ด MP3-ON คุณสมบัติการเล่นไฟล์จะได้เป็นตามที่ผู้ใช้กำหนดไว้เสมอ โดยคุณสมบัติการเล่นสามารถกำหนดได้ตาม Byte มีดังนี้

1) Byte5 : '0' = กำหนดให้คำสั่ง Play ใดๆ ที่ส่งมาเล่นไฟล์เสียงทันทีโดยไม่ต้องรอไฟล์เสียงที่เล่นอยู่จบก่อน

Byte5 : '1' = กำหนดให้คำสั่ง Play ใดๆ ที่ส่งมารอจนกว่าไฟล์เสียงที่เล่นอยู่เล่นจบก่อน
ถึงจะเริ่มเล่นไฟล์เสียงที่ส่งมา

2) Byte6 : '0' = กำหนดให้เล่นไฟล์เสียงใน Micro SD Card ทั้งหมด 1 รอบ หรือ เล่นเฉพาะไฟล์เสียงที่กำหนด 1 รอบ แล้วจบ พร้อมรับคำสั่งใหม่

Byte6 : '1' = กำหนดให้เล่นไฟล์เสียงใน Micro SD Card ซ้ำทั้งหมด หรือ เล่นซ้ำไฟล์เสียงที่กำหนด ไปเรื่อยๆ

3) Byte7 : '0' = กำหนดให้หยุดหรือไม่ให้แสดงค่าเวลาการเล่นไฟล์เสียง (Play Time Disable) ในขณะที่เล่นไฟล์เสียง

Byte7 : '1' = กำหนดให้แสดงค่าเวลาการเล่นไฟล์เสียง (Play Time Enable) ในขณะที่ไฟล์เสียงกำลังเล่นอยู่

รูปแบบคำสั่งแสดงในตาราง

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 = คุณสมบัติการเล่น 3 Byte แต่ละ Byte เป็น ASCII '0' หรือ '1' เท่านั้น ดูความหมายด้านบนตาราง

Ex.

```
char enter = 0x0D
```

```
printf("*CF=101%c",enter) ; // รอจนกว่าไฟล์ที่เล่นอยู่จบลงแล้วถึงเล่นไฟล์ที่ส่งมา, เล่นไฟล์ที่ส่งมาเพียงรอบเดียว
หรือ // และแสดงเวลาขณะเล่นด้วย
```

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

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

เป็นคำสั่งใช้สำหรับเรียกดูไฟล์เสียงที่กำลังเล่นอยู่ในขณะนั้น รูปแบบคำสั่งแสดงในตาราง

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

เมื่อส่งคำสั่งนี้ออกไปขณะที่ไฟล์เสียงกำลังเล่นอยู่และคำสั่งถูกกระทำแล้ว ก็จะมี Respond ตอบกลับมาเป็นชื่อ-นามสกุล ของไฟล์เสียงที่เล่นอยู่ แล้วถึงจบด้วยด้วย Respond ของคำสั่งคือ '#OK' โดยมีรูปแบบคือ

\$NN.FFFFFFFF.MP3 (หรือ WAV)

← Respond File Name Current

NN = หมายเลขลำดับไฟล์เสียงที่กำลังเล่น ASCII 2 Byte ('00-99')

FFFFFFF = ชื่อไฟล์เสียงที่กำลังเล่น ASCII ไม่เกิน 8 Byte

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

\$03.TEST1.MP3 : ไฟล์ที่กำลังเล่นอยู่เป็นไฟล์ลำดับที่3 ชื่อ ไฟล์ TEST1.MP3

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

เป็นคำสั่งใช้สำหรับเรียกดูไฟล์เสียงที่มีนามสกุล .MP3 และ .WAV ทั้งหมดที่มีอยู่ใน Micro SD Card การใช้คำสั่งนี้ถ้าส่งไปขณะที่ไฟล์เสียงกำลังเล่นอยู่ จะเป็นการ STOP การเล่นทันที และ List File ทั้งหมดก็จะถูกส่งออกมาทาง RS232 เมื่อใช้โปรแกรม Terminal รับข้อมูลที่ส่งมาก็จะเห็นลำดับที่และรายชื่อไฟล์ดังแสดงในตัวอย่าง โดยลำดับที่และรายชื่อไฟล์ที่แสดงนี้ผู้ใช้สามารถนำไปใช้ในการคำสั่งเล่นไฟล์เสียงโดยลำดับที่ หรือโดยชื่อไฟล์ได้ รูปแบบคำสั่งแสดงในตาราง

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

Ex. Respond ไฟล์ทั้งหมดใน Card

หมายเลขลำดับที่ไฟล์

ชื่อและนามสกุลไฟล์

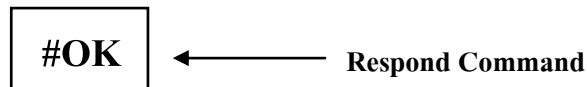
Respond คำสั่ง NA

Ex. การส่งคำสั่ง NA

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

- 12. RESPOND COMMAND

สำหรับ Respond นี้จะเป็นการตอบกลับจากบอร์ด MP3 ทุกครั้งเมื่อคำสั่งที่ส่งมายังบอร์ด MP3 นั้นถูกต้อง และคำสั่งถูกกระทำเรียบร้อยแล้ว โดยรูปแบบของ Respond จะเป็น ASCII ตัวพิมพ์ใหญ่ ขนาด 3 Byte ซึ่งมีรูปแบบดังนี้



Ex. การตรวจสอบ Respond(Echo) Command (Code for AVR Mega 128 ;Compiler WinAVR)

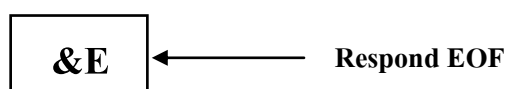
```
//----- Check Echo Command (#OK)-----

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
}
```

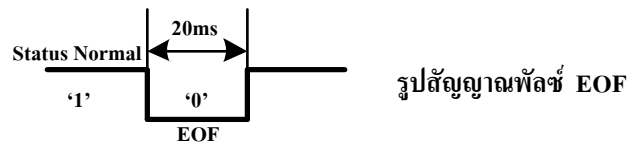
การทำงานของโปรแกรมนี้นี้เริ่มต้นจะอ่านค่า Byte Start ว่าใช่ '#' หรือไม่ ถ้าใช่ก็จะทำการอ่านค่า Data อีก 2 byte แล้วนำมาตรวจสอบว่าใช่ 'OK' หรือไม่ ถ้าใช่ ก็ออกจาก Loop ถ้าไม่ใช่ก็วนกลับไปอ่าน Data ใหม่เป็นต้น

- 13.RESPOND End Of File

สำหรับ Respond นี้จะเป็นการตอบกลับจากบอร์ด MP3 ทุกครั้งเมื่อการเล่นไฟล์เพลงนั้นๆจบลง ไม่ว่าจะจบลงจากการเล่นปกติ หรือจบลงจากการใช้คำสั่ง STOP , Forward หรือ Reverse ก็ตามบอร์ด MP3 ก็จะส่ง Respond นี้ ออกมาให้เสมอ โดยรูปแบบของ Respond จะเป็น ASCII ตัวพิมพ์ใหญ่ขนาด 2 Byte ซึ่งมีรูปแบบดังนี้



สำหรับ Respond Command นี้ นอกจากจะส่งเป็น ASCII ออกมาทาง RS232 แล้ว ก็ยังส่งเป็นพัลส์ Logic '0' ความกว้าง 20 ms ออกไปยังขาต่อหมายเลข 15 (OUT -EOF) ด้วย โดยในขณะที่ไม่มีสัญญาณ EOF เกิดขึ้น สัญญาณ EOF ก็จะมีสถานะเป็น Logic '1' (3.3V)



Ex. การตรวจสอบ Respond (Echo) EOF ทาง 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
}
```

ในตัวอย่างข้างบนเป็นการตรวจสอบ Respond EOF เริ่มต้นจะอ่านค่า Byte Start ว่าใช่ '&' หรือไม่ ถ้าใช่ก็จะทำการอ่านค่า Data อีก 1 byte แล้วนำมาตรวจสอบว่าใช่ตัว 'E'หรือไม่ ถ้าใช่ ก็ออกจาก Loop ถ้าไม่ใช่ก็วนกลับไปอ่าน Data ใหม่เป็นต้น

Ex. การตรวจสอบ Respond EOF จาก 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
}
```

ในตัวอย่างนี้เราจะตรวจสอบสัญญาณ EOF 2 Step คือ Step แรกจะตรวจสอบสัญญาณ EOF ว่าเป็น 0 หรือยัง ถ้าเป็น 0 แสดงว่าเล่นไฟล์จบแล้ว จากนั้นก็จะตรวจสอบว่าสัญญาณ EOF กลับมาเป็น 1 หรือยัง ถ้าเป็น 1 แล้ว แสดงว่าพร้อมรับคำสั่งต่อไป ถ้าผู้ใช้ตรวจสอบแค่ Step แรก Step เดียว แล้วส่งคำสั่งเลขโดยที่สัญญาณ EOF ยังไม่กลับมาเป็น 1 จะทำให้คำสั่งที่ส่งมาอาจไม่ถูกกระทำได้

- 14. RESPOND Play Time

สำหรับ Respond นี้จะเป็นการส่งค่าเวลาขณะทำการเล่นไฟล์เพลงจากบอร์ด MP3.ออกมาให้ตลอดเวลาการเล่น ทาง RS232 ผู้ใช้สามารถใช้โปรแกรม Terminal แสดงเวลาการเล่นได้ และสามารถ ใช้ MCU อ่านค่าเวลาขณะเล่นไปใช้งานได้ ซึ่งการจะให้บอร์ด MP3. ส่ง Respond ค่าเวลาการเล่นออกมานั้นผู้ใช้จะต้องทำการส่งคำสั่ง CF โดยกำหนดให้ Byte ที่ 7 ของคำสั่งเป็น '1' เพื่อ Enable Play Time เสียก่อน(Default Disable Play Time) โดยรูปแบบของ Respond นี้จะเป็น ASCII ตัวพิมพ์ใหญ่ขนาด 6 Byte ซึ่งมีรูปแบบดังนี้

@MM:SS
←
Respond Play Time

MM = นาที ASCII 2 Byte

SS = วินาที ASCII 2 Byte

Ex. การอ่าน 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'
    }
}
```

ในตัวอย่างนี้เราจะต้องตั้งตัวแปร Buf_Time[] สำหรับเก็บค่าเวลาที่อ่านมาได้เพื่อนำไปใช้งาน เริ่มต้นโปรแกรมจะตรวจสอบ Byte Start ก่อนถ้าเป็น '@' ก็เริ่มทำการเก็บค่าเวลา 5 Byte ลงในตัวแปร Buf_Time [] ตามลำดับ

ข้อควรจำ หลังจากการส่งคำสั่งทุกครั้งควรมีการตรวจสอบ Echo (#OK) หรือใช้ delay ก่อนที่จะมีการส่งคำสั่งต่อไปเสมอเพื่อคำสั่งที่ส่งไปจะไม่ทับซ้อนกัน มิฉะนั้นอาจเกิดความผิดพลาดในการส่งคำสั่งควบคุมบอร์ด MP3 ได้

สรุปตาราง COMMAND และ RESPOND COMMAND ET-REMOTE MP3 V2.

		Start	CMM		Mark1	Byte Data			End Byte
No.	Name (CMM)	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 = เลขฐาน16 สำหรับใช้แทน Enter

No.2 = Byte Data '01-99' คือหมายเลขลำดับเพลง Ascii 2 byte

No.3 = Byte Data 'FFFFFFFF.MP3' คือ ชื่อไฟล์ ไม่เกิน 8 byte และนามสกุล 3 Byte

No.8 = Byte Data '00-99' คือค่า Volume Ascii 2 Byte เมื่อ '00' = ไม่มีเสียง , '99' = เสียงดังสุด

No.9 = Byte data '0-1' , '0-1' , '0-1' คือค่า Configuration Ascii 3 byte แต่ละ Byte กำหนดได้ '0' หรือ '1'

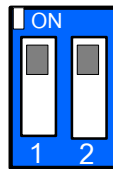
No.10 = Respond Show File NN=หมายเลขลำดับไฟล์ , FFFFFFFF = ชื่อไฟล์ไม่เกิน 8 byte

No.14 = Respond Play Time MM= นาที , SS= วินาที

4.1.3) การใช้งาน

- 1) Copy File นามสกุล MP3 หรือ WAV ลงใน Micro SD Card แล้วใส่การ์ดลงใน Socket
- 2) ตรวจสอบ Jumper JP1-JP4 บนบอร์ดว่ามีการใส่ไว้เรียบร้อยแล้ว และ ต่อหูฟังสำหรับฟังเสียง
- 3) ต่อสาย RS232 ที่จะใช้ส่ง Command ไปยัง PC หรือ MCU ให้เรียบร้อยแล้ว
- 4) Set DIP-SW_MODE โดยเลือกไปที่ตำแหน่ง COMMAND MODE (S1=ON,S2=ON) ดังรูปด้านล่าง

SW_MODE



รูปแสดงการ Set SW_MODE ให้ทำงานใน COMMAND MODE

- 5) Set DIP-SW_CONFIG เพื่อเลือก Baud Rate ในการรับ-ส่งข้อมูลตามที่ใช้ต้องการ ดูในตารางด้านล่าง ทุกครั้งที่มีการเปลี่ยนแปลงการ Set DIP-SW. ให้ Reset บอร์ดเสมอเพื่อ Update ผลการ Set

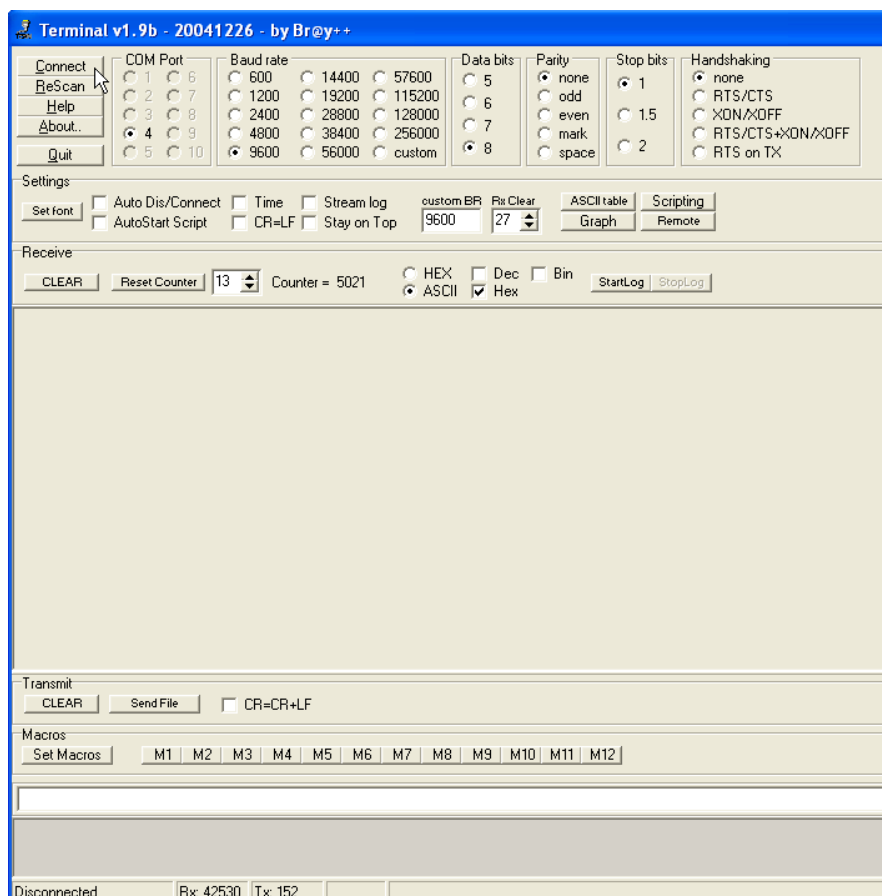
ตารางการ Set DIP-SW_CONFIG (Set Baud Rate for COMMAND MODE)

DIP-SW_CONFIG		รูปการ 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) จ่าย Power Supply DC 5V ให้กับบอร์ด (ระวังอย่าต่อสลับขั้ว) จะเห็น LED-PWR ติด บอร์ดพร้อมใช้งาน
- 7) ให้ใช้โปรแกรม Terminal (เมื่อต่อกับ PC) หรือ MCU ลองส่งคำสั่งตามที่กล่าวไปข้างต้นดู โดยกำหนดคุณสมบัติของ RS232 ตามที่ได้ อธิบายไว้ข้างต้น

ตัวอย่างการทดสอบ ส่ง Command ผ่าน RS232 โดยใช้ โปรแกรม Terminal

- Step1.** ต่อสาย RS 232 จาก PC ไปยังขั้วต่อ RS232 ของบอร์ด ET-REMOTE MP3 .V2 ให้เรียบร้อยแล้ว
- Step2.** Set DIP-SW MODE ให้ทำงานใน COMMAND MODE (S1,S2=ON) และ Set DIP-SW CONFIG เพื่อเลือก Baud Rate = 9600(S1= ON,S2= OFF)
- Step3.** Copy โปรแกรม Terminal จาก CD ลง PC แล้ว Run โปรแกรม Terminal ขึ้นมา ให้ Set ค่าต่างๆ ดังแสดงในรูปด้านล่าง เมื่อกำหนดค่าต่างๆ เรียบร้อยแล้ว ให้คลิกที่ปุ่ม Connect เพื่อ OPEN PORT RS232 ของ PC



Com Port : ตามที่ใช้ต่อ

Baud rate : 9600

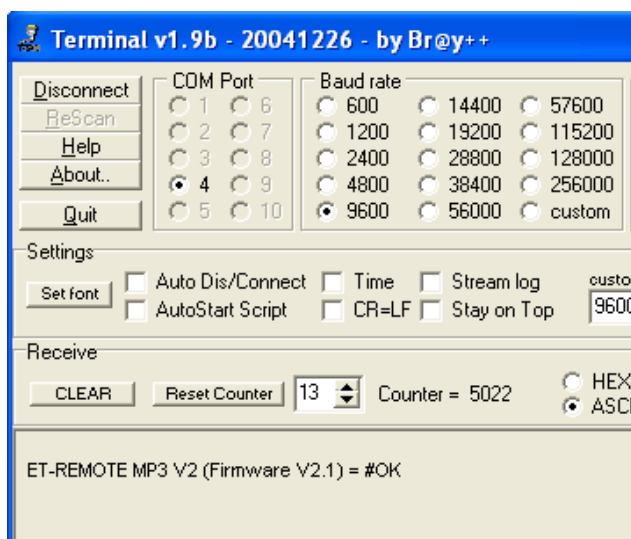
Data bit : 8

Parity : none

Stop bit : 1

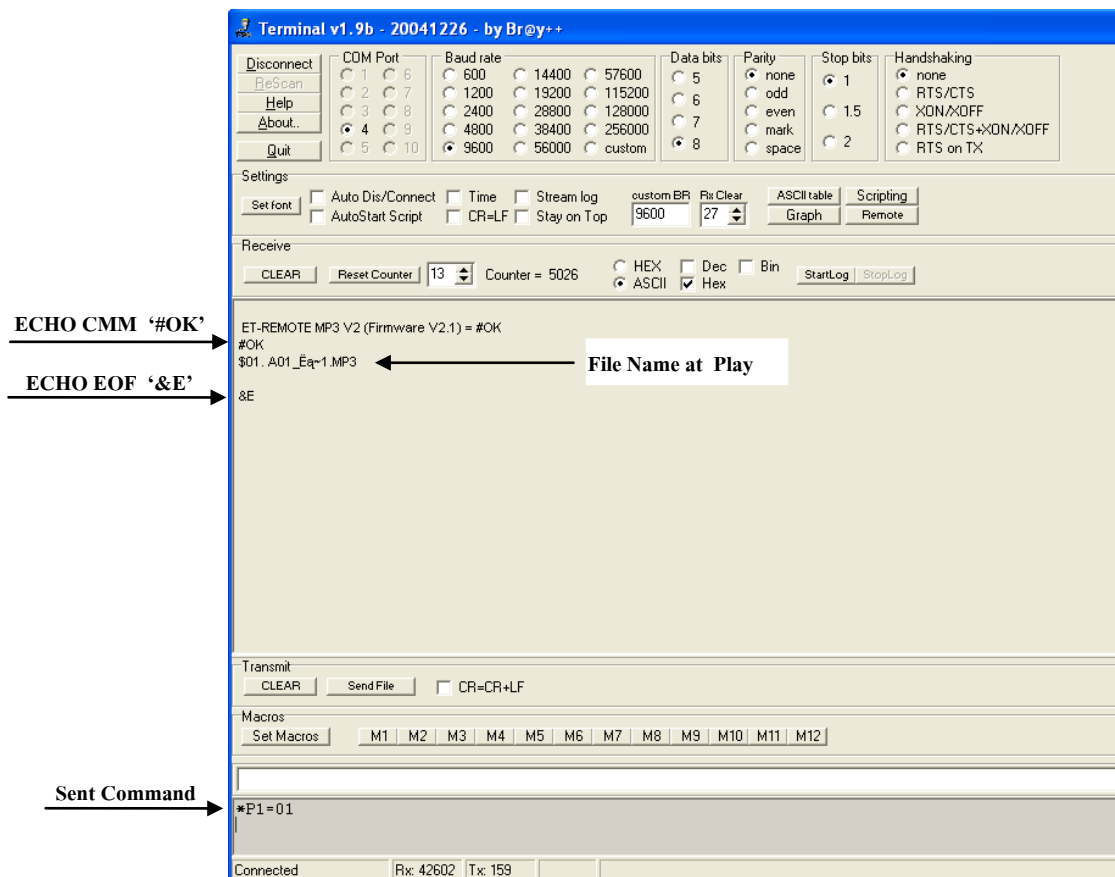
Handshaking : none

Step4. จ่ายไฟเลี้ยง 5V ให้กับบอร์ด ET-REMOTE MP3. V2 ที่โปรแกรม Terminal ก็จะปรากฏข้อความดังรูปด้านล่าง



Step5. ทดสอบพิมพ์คำสั่งให้เล่นไฟล์เสียงในลำดับที่ 1 ลงในช่องสี่เหลี่ยมด้านล่างสุด โดยพิมพ์ดังนี้ *P1=01 แล้วกด

Enter บอร์ดก็จะส่ง Respond ออกมาดังรูปด้านล่าง และไฟล์เสียงลำดับที่ 1 ก็จะถูกเล่น ให้ผู้ใช้ลองพิมพ์คำสั่งอื่นๆเพื่อทดสอบการทำงาน



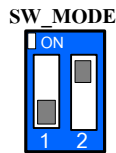
4.2 LIST_SONG MODE

4.2.1) การทำงาน

ในโหมดนี้จะใช้สำหรับเรียกดูรายชื่อ และลำดับที่ ของไฟล์เสียงที่มีนามสกุล .MP3 และ .WAV ทั้งหมดที่ถูกเก็บไว้ใน Micro SD Card โดยเริ่มต้นเมื่อเราทำการ Set DIF-SW_MODE ไปที่ LIST_SONG Mode และ Set DIF-SW_CONFIG เพื่อกำหนด Baud Rate ในการส่ง จากนั้นเมื่อจ่าย Power Supply ให้กับบอร์ด MP3 ตัวโปรแกรมก็จะทำการอ่านชื่อไฟล์และลำดับที่ของไฟล์จาก Micro SD Card โดยไฟล์ที่ถูก Copy ลงใน Card ก่อนจะถูกอ่านขึ้นมาเป็น File แรกเสมอ และส่งลำดับที่กับชื่อไฟล์ที่อ่านมาได้ออกไปทาง RS232 ของบอร์ด เมื่อผู้ใช้เปิดโปรแกรม Terminal (ต้องเปิดรอไว้ก่อนจ่ายไฟเลี้ยงบอร์ด MP3) หรือโปรแกรมที่รองรับการแสดงผลทาง RS232 ก็จะเห็นรายชื่อไฟล์และลำดับที่ของไฟล์เสียงที่อยู่ใน Micro SD Card ขึ้นมา (ดูรูปได้จากหัวข้อการใช้งาน) โดยรายชื่อและนามสกุลไฟล์ที่แสดงให้เห็นทางโปรแกรม Terminal จะเป็นตัวพิมพ์ใหญ่เสมอ ไม่ว่าชื่อ-นามสกุลไฟล์ที่เรากำลังจะดูนั้นจะเป็นตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็กก็ตาม และจะแสดงในส่วนแรกของชื่อไฟล์ไม่เกิน 8 Byte (ตามรูปแบบไฟล์ 8.3) และกรณีที่ชื่อไฟล์เป็นภาษาไทยก็จะแสดงเป็นสัญลักษณ์แบบอ่านไม่รู้เรื่องแทน เวลาจะนำลำดับที่ , ชื่อ-นามสกุลไฟล์ ไปใช้แทนในคำสั่ง Play ที่อยู่ใน Command Mode ให้อ้างอิงตามผลที่แสดงในโหมดนี้ ทุกครั้งที่มีการเปลี่ยนแปลง ข้อมูลใน Micro SD Card ใหม่ และผู้ใช้งานต้องการ Update การแสดงผล ผู้ใช้จะต้องทำการ Reset บอร์ด MP3 ใหม่เสมอ

4.2.2) การใช้งาน

- 1) Copy File นามสกุล MP3 หรือ WAV ลงใน Micro SD Card แล้วใส่การ์ดลงใน Socket
- 2) ตรวจสอบ Jumper JP1-JP4 บนบอร์ดว่ามีการใส่ไว้เรียบร้อยแล้ว
- 3) ต่อสาย RS232 จากบอร์ด MP3 ไปยังขั้ว RS232 DB9 ของ PC
- 4) Set DIP-SW_MODE โดยเลือกไปที่ตำแหน่ง LIST_SONG MODE (S1=OFF,S2=ON) ดังรูปด้านล่าง



รูปแสดงการ Set SW_MODE ให้ทำงานใน LIST_SONG MODE

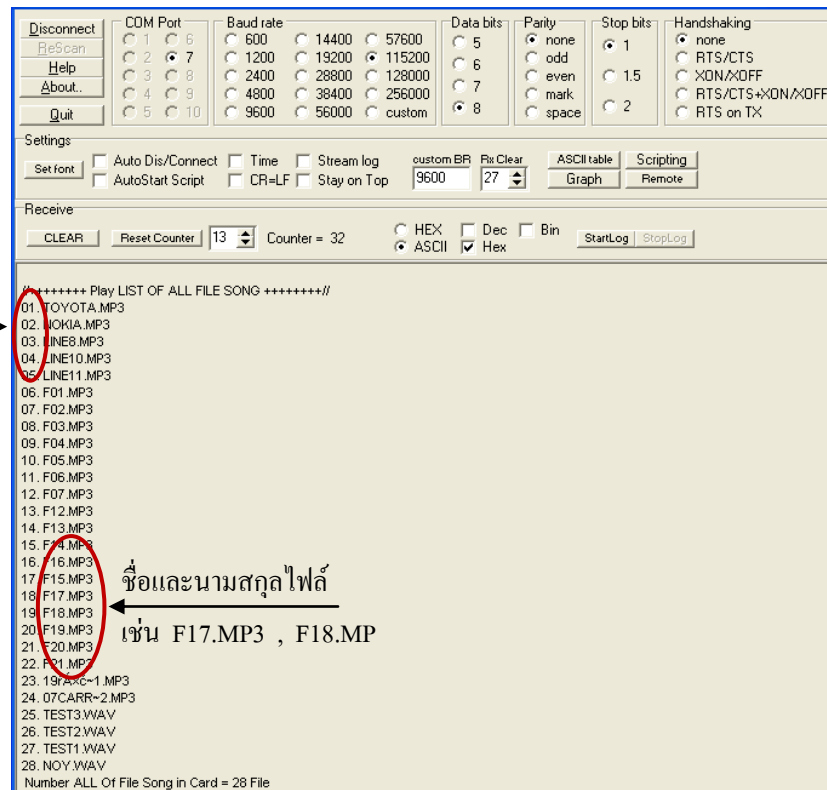
- 5) Set DIP-SW_CONFIG เพื่อเลือก Baud Rate ในการส่งข้อมูลตามที่ใช้ต้องการ ดูในตารางด้านล่าง และทุกครั้งที่มีการเปลี่ยนแปลงการ Set DIP- SW. ให้ Reset บอร์ดเสมอเพื่อ Update ผลการ Set

ตารางการ Set DIP-SW_CONFIG (Set Baud Rate for LIST_SONG MODE)

DIP-SW_CONFIG		รูปการ 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) เปิดโปรแกรม Terminal หรือโปรแกรมที่รองรับการแสดงผลทาง RS232 ขึ้นมารอไว้ โดย Set คุณสมบัติดังนี้
 Baud Rate : ตามที่กำหนด DIP-SW ในข้อ 4 , DATA BIT : 8 , PARITY : NONE
 STOP BIT : 1 , HANDSHAK : NONE
- 7) จ่าย Power Supply DC 5V ให้กับบอร์ด (ระวังอย่าต่อสลับขั้ว) จะเห็น LED-PWR ติด
- 8) โปรแกรม Terminal ก็จะแสดง ลำดับที่และ รายชื่อ File ที่อยู่ใน Micro SD Card ให้เห็นโดยมีรายละเอียดแต่ละส่วนดังรูปด้านล่าง

หมายเลขลำดับที่ไฟล์
เช่น 02 , 03 , 04



รูปแสดงลำดับที่และรายชื่อไฟล์เสียง ที่เรียกดูใน MODE LIST_SONG

เวลาจะนำค่าลำดับที่ หรือ ชื่อ-นามสกุลไฟล์เสียง ไปแทนค่าในคำสั่ง Play ใน Command Mode เช่น Play โดย ลำดับที่ ก็ให้นำลำดับที่ของไฟล์ที่ต้องการจะสั่ง Play ไปแทนค่า เช่น 01 หรือ 02 หรือ 15 เป็นต้น ถ้าต้องการให้ Play โดยชื่อก็ให้นำรายชื่อ-นามสกุลของไฟล์ ที่ต้องการจะสั่ง Play ที่แสดงให้เห็นบน Terminal ไปแทนค่าลงในคำสั่ง เช่น F17.MP3 หรือ TEST2.WAV เป็นต้น

- 9) เมื่อมีการเปลี่ยนแปลงข้อมูลใน Micro SD Card หรือ Set DIP-SW ใดๆใหม่ ให้ผู้ใช้ ทำการ Reset บอร์ดใหม่ทุกครั้งเพื่อ Update ผลการเปลี่ยนแปลงที่เกิดขึ้น

4.3 MP3_SW MODE

4.3.1) การทำงาน

สำหรับในโหมดนี้จะเป็นการควบคุมการทำงานของบอร์ด MP3 ผ่าน Switch 3 ตัวที่อยู่บนบอร์ด ซึ่งเหมาะสำหรับการใช้เปิดฟังเหมือนเครื่องเล่น MP3 ทั่วไป โดยในโหมดนี้ Switch 3 ตัวบนบอร์ดจะมีหน้าที่ต่างๆดังนี้ Play , Pause, Stop , Forward , Reverse , Volume up ,Volume Down โดยรายละเอียดการใช้งาน Switch ของแต่ละหน้าที่ได้จากตารางในหัวข้อ “การใช้งาน” ด้านล่าง และเรายังสามารถกำหนดคุณสมบัติการเล่นได้โดยใช้ DIP-SW_CONFIG กำหนดคุณสมบัติการเล่น เช่น เล่นไฟล์เสียงทั้งหมดใน Card 1 รอบ แล้ว STOP หรือ เล่นซ้ำทั้งหมด หรือ เมื่อ Power ON ให้ Play ทันที หรือให้รอกด SW. Play ก่อน เป็นต้น ดูรายละเอียดการ Set ได้จากตารางในหัวข้อ “การใช้งาน” ด้านล่างเช่นกัน

เริ่มต้นการทำงานในโหมดนี้ ตัวโปรแกรมก็จะตรวจสอบ DIP-SW_CONFIG ว่ากำหนดคุณสมบัติการเล่นไว้เช่นไร ไม่ว่าจะเป็นกำหนดให้เล่นทันที เมื่อ Power ON หรือ รอให้กด SW.PLAY/STOP ก่อนแล้วถึงเริ่มเล่น การเล่นไฟล์เสียงจะถูกเริ่มเล่นจากไฟล์ลำดับที่ 1 ก่อนเสมอ ซึ่งก็คือไฟล์ที่ถูก Copy ลงไปไว้ใน Micro SD Card เป็นไฟล์

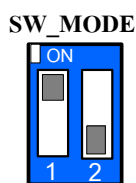
แรกนั่นเอง เมื่อเล่นไฟล์ลำดับที่ 1 จบ ก็จะเล่นไฟล์ลำดับที่ 2,3,4...ต่อไปเรื่อยๆจนครบไฟล์ที่มีอยู่ใน Card ทั้งหมด และทุกครั้งที่ไฟล์เสียงแต่ละไฟล์เล่นจบลง หรือมีการกด Forward , Reverse หรือ Stop ตัวบอร์ดจะส่งสัญญาณ EOF Active '0' ออกมาที่ขั้วต่อหมายเลข 15 (LED EOF จะติดและดับลง) เพื่อบอกให้รู้ว่าไฟล์ที่เล่นจบลงแล้ว

ในขณะที่เล่นไฟล์เสียงอยู่ถ้ามีการ Reset บอร์ด MP3 หรือ Power OFF และ Power On ขึ้นมาใหม่ การเล่นจะกลับไปเริ่มเล่นที่ไฟล์เสียงลำดับที่ 1 ใหม่เสมอรวมทั้งค่า Volume ที่ Set ไว้ ก็จะกลับไปอยู่ที่ค่า Default เสมอ บอร์ด MP3 จะไม่จำค่าสถานะเหล่านี้ แต่ค่าคุณสมบัติการเล่นที่กำหนดจาก DIP-SW_CONFIG จะยังคงเหมือนเดิม ถ้าผู้ใช้ ยังไม่มีการปรับเปลี่ยนใดๆ โดยการปรับเปลี่ยนการ Set ค่า DIP-SW_CONFIG จะมีผล หรือถูก Update คุณสมบัติการเล่นก็ต่อเมื่อ มีการ Reset บอร์ด MP3 ใหม่

สำหรับการทำงานของ Switch ทั้ง 3 ตัวบนบอร์ด จะทำงานมากกว่าหนึ่งหน้าที่ในปุ่มเดียว โดยจะอาศัยการ กด Switch ใน 2 ลักษณะเพื่อแยกแยะการทำงาน คือ กดแช่น้อยกว่า 0.5 วินาที แล้วปล่อย (กดแล้วปล่อยทันที) และอีกกรณีหนึ่งคือ กดแช่ตั้งแต่ 0.5 วินาทีขึ้นไปแล้วปล่อย โดยการกดในลักษณะไหนจะทำหน้าที่อะไรให้ดูได้จากตาราง ในหัวข้อ “การใช้งาน” ด้านล่าง

4.3.2) การใช้งาน

- 1) Copy File นามสกุล MP3 หรือ WAV ลงใน Micro SD Card แล้วใส่การ์ดลงใน Socket
- 2) ตรวจสอบ Jumper JP1-JP4 บนบอร์ดว่ามีการใส่ไว้เรียบร้อยแล้ว
- 3) Set DIP-SW_MODE โดยเลือกไปที่ตำแหน่ง MP3_SW MODE (S1=ON,S2=OFF) ดังรูปด้านล่าง



รูปแสดงการ Set SW_MODE ให้ทำงานใน MP3_SW MODE

- 4) Set DIP-SW_CONFIG เพื่อกำหนดคุณสมบัติการเล่น โดยดูได้จากตารางด้านล่าง ทุกครั้งที่มีการเปลี่ยนแปลงการ Set DIP-SW. ให้ Reset บอร์ด เสมอเพื่อ Update ผลการ Set

ตารางการ Set DIP-SW_CONFIG (Set properties play for MP3_SW MODE)

DIP-SW_CONFIG		รูปการ SET SW_CONFIG	Properties Play (คุณสมบัติการเล่น)
S1	S2		
ON (0)	ON (0)		- เล่นไฟล์เสียงทันที เมื่อ Power ON หรือ Reset โดยไม่ต้องกด SW.PLAY/STOP โดยเล่นไฟล์เสียงทั้งหมดใน Card 1 รอบ แล้วหยุดการเล่นอัตโนมัติ
OFF (1)	ON (0)		- เล่นไฟล์เสียงเมื่อมีการกด SW.PLAY/STOP โดยเล่นไฟล์เสียงทั้งหมดใน Card 1 รอบ แล้วหยุดการเล่นอัตโนมัติ
ON (0)	OFF (1)		- เล่นไฟล์เสียงทันที เมื่อ Power ON หรือ Reset โดยไม่ต้องกด SW.PLAY/STOP โดยให้เล่นไฟล์เสียงใน Card ซ้ำทั้งหมด (Repeat All)
OFF (1)	OFF (1)		- เล่นไฟล์เสียงเมื่อมีการกด SW.PLAY/STOP โดยให้เล่นไฟล์เสียงใน Card ซ้ำทั้งหมด (Repeat All)

- 5) จ่าย Power Supply DC 5V ให้กับบอร์ด (ระวังอย่าต่อสลับขั้ว) จะเห็น LED-PWR ติด
- 6) ให้ทดสอบการกด SW. PLAY/STOP เพื่อเริ่มต้นการเล่นไฟล์เสียง จากนั้นทดสอบหน้าที่ SW.ตามตารางด้านล่าง
- 7) สำหรับหน้าที่ของ SW. ทั้ง 3 ตัวบนบอร์ด มีหน้าที่ตามรูปแบบการกดดังนี้

ตารางแสดงหน้าที่การทำงานของ SWITCH ทั้ง 3 ตัว On Board For MP3_SW MODE

SW. NAME	กดสั้น(ต่ำกว่า 0.5วินาที)-ปล่อย		กดแช่ (เกิน0.5วินาที) -ปล่อย	
	กดขณะเล่นไฟล์	กดขณะไม่เล่นไฟล์	กดขณะเล่นไฟล์	กดขณะไม่เล่นไฟล์
PLAY/STOP	STOP ([]) หยุด	PLAY (<) เล่น	PAUSE() หยุดชั่วคราว	-
RV/VOL-	Reverse 1 File (<<)	-	Volume Down (--)	Volume Down (--)
FW/VOL+	Forward 1 File (>>)	-	Volume Up (++)	Volume Up (++)

- 8) ทุกครั้งที่ไฟล์เสียงเล่นจบลง หรือ กด SW. FW,RV บอร์ด MP3 จะส่งสัญญาณ EOF (Active LOW) ออกมาที่ขั้วต่อ OUT-EOF สังเกต LED -EOF จะกระพริบ ซึ่งสามารถนำสัญญาณนี้ไปตรวจสอบการจบของไฟล์ที่เล่นได้

4.4 TRIG MODE

4.4.1) การทำงาน

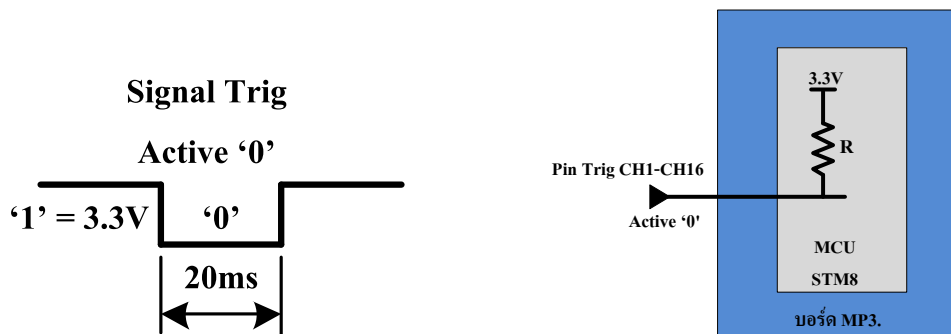
สำหรับในโหมดนี้จะเป็นการควบคุมการทำงานของบอร์ด MP3 ผ่านขั้วต่อ Input TRIG (19.CON-TRIG) ที่อยู่บนบอร์ด โดยจะมีขั้วต่อช่องรับสัญญาณ Trig อยู่ 16 ช่อง(1-16) โดยเมื่อมีสัญญาณ Trig เข้ามาที่ช่องใดไฟล์เสียงที่มีหมายเลขลำดับที่ตรงกับช่องที่มีสัญญาณ Trig เข้ามาก็จะถูกเล่น เช่น มีสัญญาณ Trig เข้ามาที่ Input ช่อง 12 ไฟล์เสียงที่มีหมายเลขลำดับที่12 ก็จะถูกเล่นเป็นต้น ดังนั้นในโหมดนี้จะมีข้อจำกัดคือจะสามารถเล่นไฟล์เสียงที่มีหมายเลขลำดับไฟล์ได้ไม่เกินลำดับที่ 16 นั่นคือเล่นไฟล์ได้ไม่เกิน 16 ไฟล์ (ลำดับไฟล์ที่1-16) ถึงแม้ใน Card จะมีไฟล์เสียงอยู่เกิน 16 ไฟล์ก็ตาม ในกรณีที่มีไฟล์เสียงอยู่ใน Card ไม่ถึง 16 ไฟล์ แต่มีการ Trig ในลำดับที่ไม่มีไฟล์เสียงอยู่ใน Card ก็จะไม่มีการเล่นไฟล์ใดๆเกิดขึ้น

นอกจากนี้เรายังสามารถกำหนดคุณสมบัติการเล่นไฟล์เสียงได้จาก DIP-SW_CONFIG เช่น เล่นไฟล์เสียงที่ถูก Trig 1รอบแล้วจบเพื่อรอสัญญาณ Trig ใหม่ หรือ เล่นไฟล์เสียงที่ถูก Trig เข้าไปเรื่อยๆ หรือ เราสามารถกำหนดได้ว่าให้เล่นไฟล์เสียงที่ถูก Trig ทันที หรือ รอให้ไฟล์เสียงที่เล่นอยู่จบก่อนแล้วถึงเริ่มเล่นไฟล์ที่มีการ Trig สัญญาณเข้ามาใหม่เป็นต้น ในกรณีที่มีการ Trig หลายๆครั้งในขณะที่มีการเล่นไฟล์อยู่ จะยึดเอาสัญญาณ Trig ครั้งล่าสุดเป็นหลัก การ Set คุณสมบัติการเล่นไฟล์ด้วย DIP-SW_CONFIG สามารถดูได้จากตารางในหัวข้อ “ การใช้งาน” นอกเหนือจากนั้นในโหมดนี้เรายังสามารถใช้ SW. 3 ตัวบนบอร์ดควบคุม การ STOP และปรับ Volume ได้อีกด้วย โดย SW.แต่ละตัวจะทำหน้าที่ดังตารางด้านล่าง เวลากด SW. ก็ กด-ปล่อย แบบปกติ

ตารางแสดงหน้าที่การทำงานของ SWITCH ทั้ง 3 ตัว On Broad For TRIG MODE

SW. NAME	กด-ปล่อย ปกติ	
	กดขณะเล่นไฟล์	กดขณะไม่เล่นไฟล์
PLAY/STOP	STOP (□) หยุด	-
RV/VOL-	Volume Down (--)	Volume Down (--)
FW/VOL+	Volume Up (++)	Volume Up (++)

ดังนั้นในโหมดนี้จะเหมาะสำหรับการใช้งานในลักษณะรับค่าการกด SW.หรือรับค่า Sensor จากอุปกรณ์ภายนอกเข้ามา Trig โดยลักษณะของสัญญาณ Trig นั้นจะ Active ที่ Logic '0' มีความกว้างตั้งแต่ 20 ms ขึ้นไป และระดับแรงดันในสถานะที่เป็น Logic '1' จะต้องไม่เกิน 3.3V เนื่องจาก Input ของ MCU ไม่รองรับ 5V โดยโครงสร้างภายในของขารับสัญญาณ Trig นั้นได้มีการต่อ R-Pull Up แรงดัน 3.3V ไว้แล้ว ดังรูป



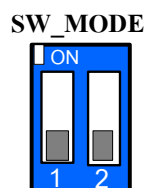
รูปแสดงโครงสร้างภายในของขารับสัญญาณ Trig CH1-CH16 และ ลักษณะสัญญาณ Trig

ฉะนั้นที่ขารับสัญญาณ Trig ของบอร์ด MP3 จะสามารถต่อได้โดยตรงกับขา I/O ของอุปกรณ์ ที่ใช้ส่งสัญญาณ Trig เช่น MUC หรือ Sensor ที่ใช้ไฟเลี้ยงไม่เกิน 3.3 V เท่านั้น ในกรณีที่อุปกรณ์ใช้ส่งสัญญาณ Trig ใช้ไฟเลี้ยงเกิน 3.3 V หรือให้สัญญาณ Logic '1' ออกมาที่ ขา I/O เกิน 3.3V จะต้อง ใช้วงจรลดระดับแรงดันเข้ามาช่วย โดยดูรายละเอียดได้จากหัวข้อ 'การใช้งาน ' ด้านล่าง

คำเตือน ! สัญญาณที่ใช้ Trig ในสถานะที่เป็น Logic '1' จะต้องมียกระดับแรงดันไม่เกิน 3.3 V เท่านั้น

4.4.2) การใช้งาน

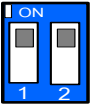
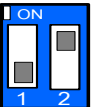
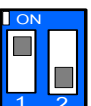
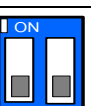
- 1) Copy File นามสกุล MP3 หรือ WAV ลงใน Micro SD Card แล้วใส่การ์ดลงใน Socket
- 2) ตรวจสอบ Jumper JP1-JP4 บนบอร์ดว่ามีการใส่ไว้เรียบร้อยแล้ว
- 3) Set DIP-SW_MODE โดยเลือกไปที่ตำแหน่ง TRIG MODE (S1=OFF,S2=OFF) ดังรูปด้านล่าง



รูปแสดงการ Set SW_MODE ให้ทำงานใน TRIG MODE

- 4) Set DIP-SW_CONFIG เพื่อกำหนดคุณสมบัติการเล่น โดยดูได้จากตารางด้านล่าง ทุกครั้งที่มีการเปลี่ยนแปลงการ Set DIP-SW. ให้ Reset บอร์ดเสมอเพื่อ Update ผลการ Set

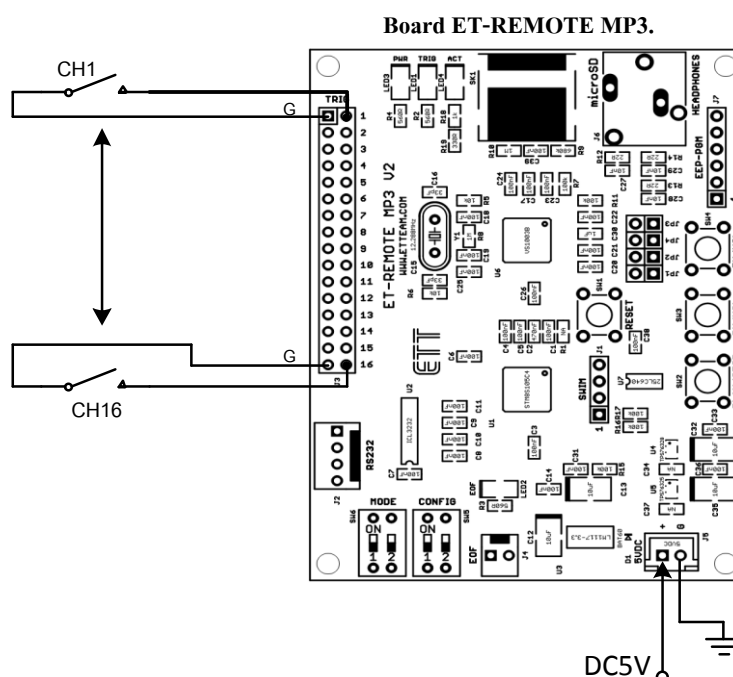
ตารางการ Set DIP-SW_CONFIG (Set properties play for TRIG MODE)

DIP-SW_CONFIG		รูปการ SET	Properties Play
S1	S2	SW_CONFIG	(คุณสมบัติการเล่น)
ON (0)	ON (0)		- เล่นไฟล์เสียงลำดับที่ถูก Trig ทันที โดยเล่นไฟล์เสียงลำดับที่ถูก Trig 1 รอบ แล้วหยุดการเล่น
OFF (1)	ON (0)		- รอไฟล์เสียงที่เล่นอยู่จบก่อน ถึงจะเริ่มเล่นไฟล์เสียงลำดับที่ถูก Trig ลำสุด โดยเล่นไฟล์เสียงลำดับที่ถูก Trig 1 รอบ แล้วหยุดการเล่น
ON (0)	OFF (1)		- เล่นไฟล์เสียงลำดับที่ถูก Trig ทันที โดยเล่นไฟล์เสียงลำดับที่ถูก Trig เข้าไปเรื่อยๆ
OFF (1)	OFF (1)		- รอไฟล์เสียงที่เล่นอยู่จบก่อน ถึงจะเริ่มเล่นไฟล์เสียงลำดับที่ถูก Trig ลำสุด โดยเล่นไฟล์เสียงลำดับที่ถูก Trig เข้าไปเรื่อยๆ

- ในกรณีที่กำหนด DIP-SW แบบเล่นไฟล์เสียงซ้ำ ในระหว่างการเล่นผู้ใช้สามารถ Trig เพื่อเปลี่ยนการเล่นไฟล์เสียงอื่นได้ และไฟล์เสียงที่ถูก Trig ใหม่ก็จะถูกกำหนดให้เล่นซ้ำเช่นกัน

5) จ่าย Power Supply DC 5V ให้กับบอร์ด (ระวังอย่าต่อสลับขั้ว) จะเห็น LED-PWR ติด

6) ให้สังเกต ที่ Connector Trig โดยด้านในบอร์ดที่มีเลขแชนแนลกำกับอยู่จะเป็น แชนแนลสำหรับรับสัญญาณ Trig Low ส่วนด้านตรงข้ามกับแชนแนลรับสัญญาณ Trig (ซีกด้านนอกบอร์ด) จะเป็นขา Ground ทั้งหมด ให้ผู้ใช้ลอง Trig แชนแนลใดแชนแนลหนึ่งลง Ground (Trig แล้วปล่อย) ไฟล์เสียงที่มีลำดับที่ตรงกับช่องที่มีสัญญาณ Trig เข้ามาจะถูกเล่น (ใน Card ควรมีไฟล์เสียงอย่างน้อย 16 ไฟล์ สำหรับทดสอบการ Trig CH1-CH16) จาก Connector Trig บนบอร์ด ผู้ใช้สามารถนำ SW.แบบ กดติดปล่อยดับ มาต่อในแต่ละแชนแนลได้โดยตรงเพื่อใช้ Trig เล่นไฟล์เสียงในลำดับที่ต้องการได้ ซึ่งการต่อ SW. Trig แสดงดังรูปด้านล่าง



รูปแสดงการนำ SW.แบบกดติดปล่อยดับมาต่อเพื่อใช้ Trig การเล่นไฟล์เสียง

7) ทุกครั้งที่ไฟล์เสียงเล่นจบลงแบบปกติ หรือ มีการ Trig เพื่อเล่นไฟล์ใหม่ จะทำให้บอร์ด MP3 ส่งสัญญาณ EOF (Active LOW) ออกมาที่ขั้วต่อ OUT – EOF สังเกต LED – EOF จะกระพริบ ซึ่งสามารถนำสัญญาณนี้ไปตรวจสอบการจบของไฟล์ที่เล่นได้

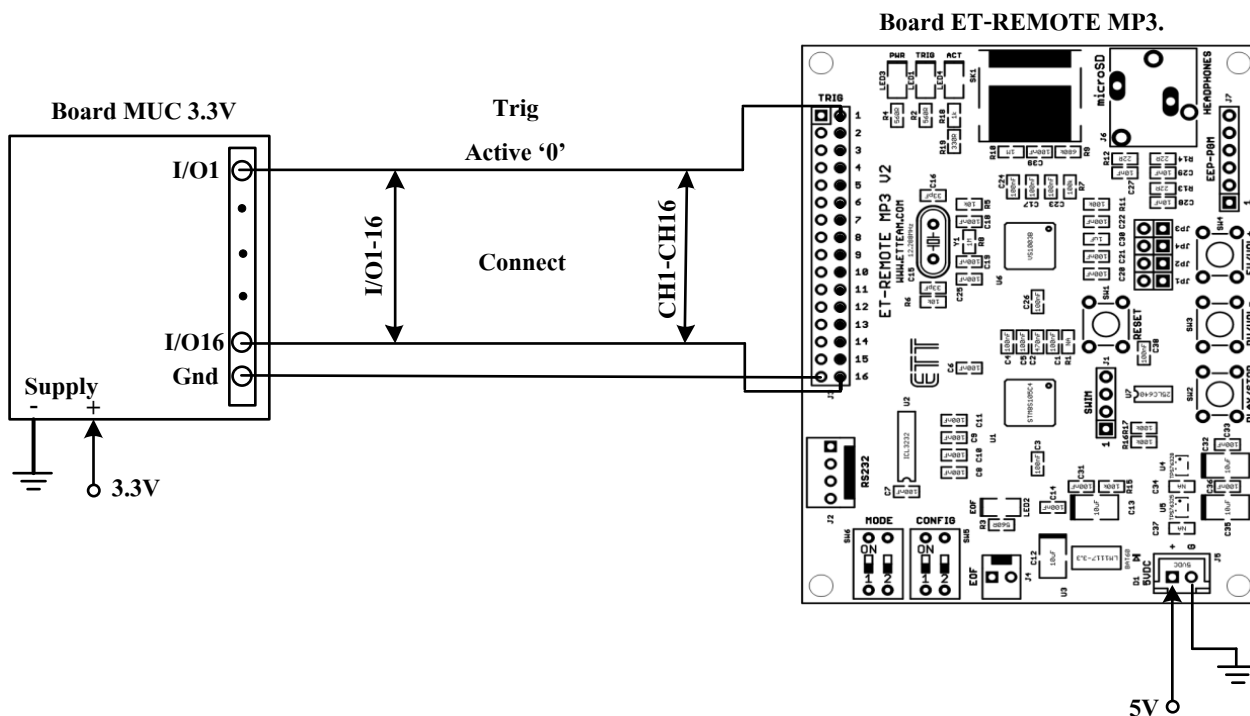
4.4.3) การ Trig โดยใช้สัญญาณ Trig จากภายนอกบอร์ด MP3

ผู้ใช้งานสามารถส่งสัญญาณ Trig จากภายนอก ด้วย MCU หรือ Sensor เข้ามาที่ขั้วต่อสัญญาณ Trig ของบอร์ด MP3 เพื่อควบคุมการเล่นไฟล์เสียงได้ โดยสิ่งที่ต้องคำนึงถึงคือระดับ Logic ของสัญญาณ Trig ในสถานะที่เป็น Logic '1' โดยตัวบอร์ด ET-Remote MP3 สามารถรับระดับของสัญญาณ Trig ที่เข้ามาได้ไม่เกิน 3.3V เท่านั้น กรณีที่สัญญาณ Trig เกิน 3.3V ก็ต้องผ่านวงจรลดระดับแรงดันลงมาให้เหลือ 3.3V ก่อนที่จะต่อไปยังขั้วรับสัญญาณ Trig ของบอร์ด MP3 ดังนั้นต่อไปนี้จะเป็นอย่างวงจรการต่อสัญญาณ Trig จากภายนอกไปควบคุมการเล่นไฟล์เสียงบนบอร์ด MP3 (TRIG- MODE) เพื่อให้ผู้ใช้งานไปเป็นแนวทาง การพัฒนาต่อไป ดังนี้

1) การส่งสัญญาณ Trig โดยใช้ MCU หรือ Sensor ที่ใช้ไฟเลี้ยง 3.3V

สำหรับการส่งสัญญาณ Trig ควบคุมการเล่นของบอร์ด MP3 จากภายนอกโดย MCU หรือ Sensor ที่ใช้ไฟเลี้ยง 3.3V นั้น ผู้ใช้สามารถต่อ I/O ของ MCU ไปยังขั้วรับสัญญาณ Trig ของบอร์ด MP3 ได้โดยตรงดังรูปด้านล่าง

ส่วนการเขียนโปรแกรม ในสถานะปกติให้ I/O 1-16 มีสถานะเป็น Logic '1' ไว้ เมื่อต้องการให้ไฟล์เสียงในแชนแนลใดถูกเล่น ก็ให้ส่ง Logic '0' ออกมาที่ขา I/O นั้นค้างไว้ประมาณ 20 ms ขึ้นไปแล้วจึง Set ขา I/O นั้นกลับมาเป็น Logic '1' เหมือนเดิม การต่อใช้งานจริงนั้นสามารถเลือกต่อเฉพาะแชนแนลที่ต้องการจะใช้งานเท่านั้นก็ได้ โดยใน Micro SD Card จะต้องมีจำนวนไฟล์เสียงอย่างน้อยเท่ากับแชนแนลที่เลือกใช้งาน

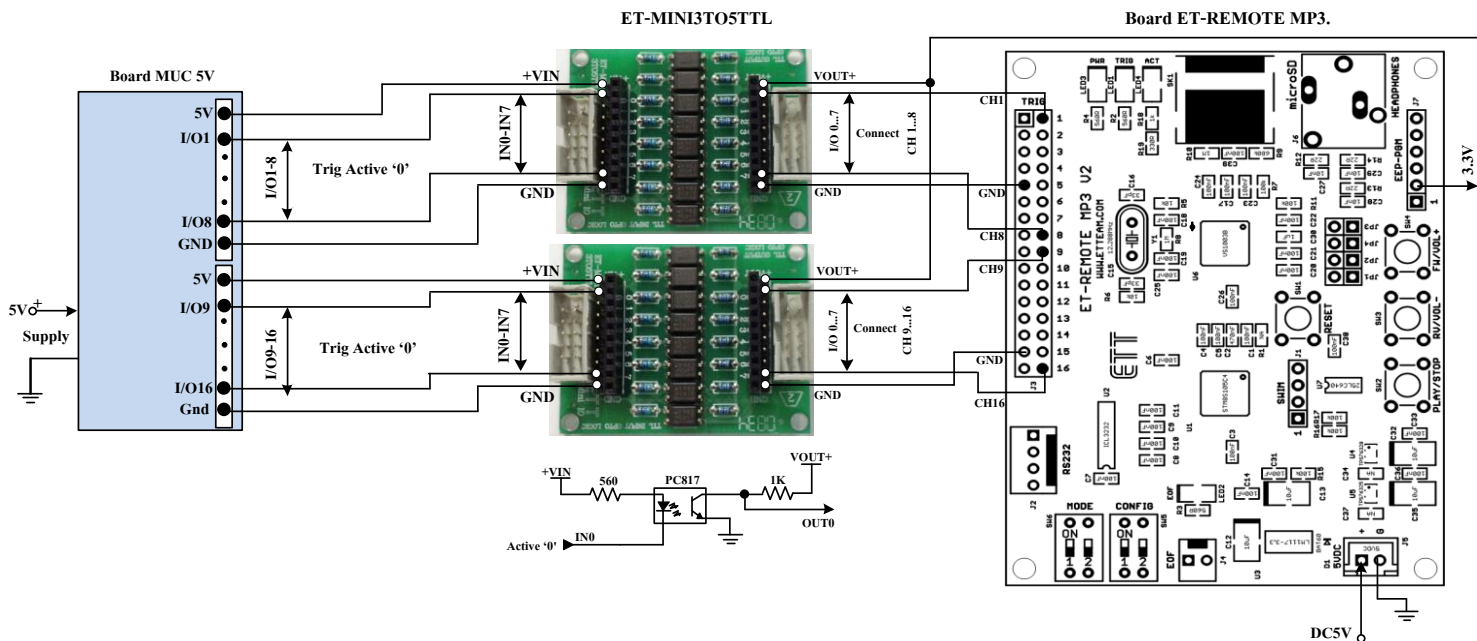


รูปการต่อขา I/O 1-16 ของบอร์ด MCU 3.3V เข้ากับขั้วรับสัญญาณ Trig CH 1-16 ของบอร์ด MP3

2) การส่งสัญญาณ Trig โดยใช้ MCU หรือ Sensor ที่ใช้ไฟเลี้ยง 5V

สำหรับการส่งสัญญาณ Trig ควบคุมการเล่นของบอร์ด MP3 จากภายนอกโดย MCU หรือ Sensor ที่ใช้ไฟเลี้ยง 5V นั้นจะไม่สามารถต่อ I/O ได้โดยตรง จะต้องผ่านวงจรระดับแรงดันไม่ให้เกิน 3.3V เสียก่อน ซึ่งวงจรระดับแรงดันนั้นจะมีหลายลักษณะซึ่งขึ้นอยู่กับอุปกรณ์ที่จะนำมาใช้ต่อวงจร โดยในที่นี้เราจะขอใช้บอร์ดลดระดับแรงดัน ที่ทางอีทีที่มีจำหน่ายมาต่อใช้งานสำหรับลดระดับแรงดันของสัญญาณ Trig ที่มีระดับแรงดัน 5V ให้เหลือ 3.3 V ดังนี้

- การต่อวงจรโดยใช้ **ET-MINI3TO5TTL** เป็นตัวลดระดับแรงดัน (เหมาะสำหรับ เลือก Trig หลายๆ แชนแนล)



รูปการต่อขา I/O 1-16 ของบอร์ด MCU 5V โดยผ่านบอร์ดลดระดับแรงดัน ET-MINI3TO5TTL

สำหรับการส่งสัญญาณ Trig ควบคุมการเล่นของบอร์ด MP3 จากภายนอกด้วย MCU หรือ Sensor ที่ใช้ไฟเลี้ยง 5V ดังรูปด้านบนนั้น จะใช้ ET-MINI3TO5TTL เป็นตัวลดระดับแรงดัน ดังนั้นสัญญาณ Trig ที่ส่งมาจาก MCU หรือ Sensor จะต้อง Active '0' เท่านั้น โดยในส่วนที่เป็นสัญญาณ Trig ผู้ใช้สามารถเลือกต่อเฉพาะแชนแนลที่ต้องการจะใช้งานจริงเท่านั้นก็ได้ ไม่จำเป็นต้องต่อครบทุกแชนแนลเหมือนในรูป

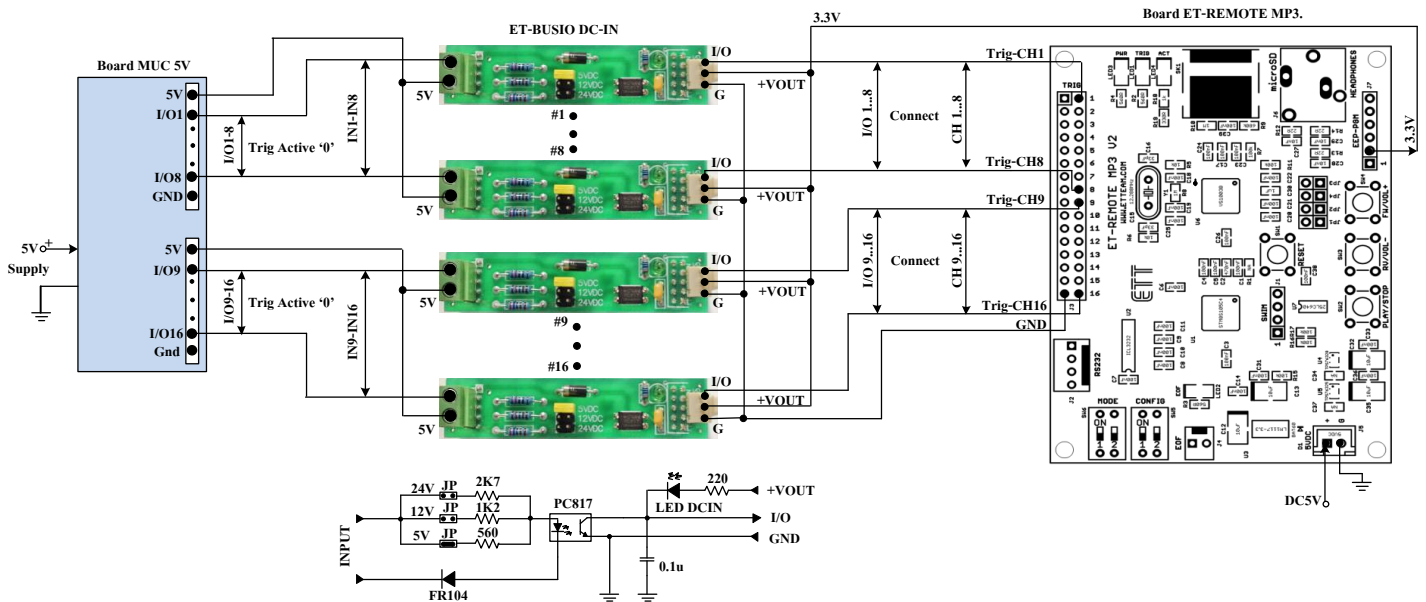
ส่วนการเขียนโปรแกรมในวงจรนี้อันดับแรกในสถานะปกติไม่มีการ Trig ขาสัญญาณ I/O แต่ละขาของ MCU จะต้องกำหนดให้เป็น 1 ไว้ เมื่อต้องการสั่งให้เล่นไฟล์เสียง ก็ให้ส่งสัญญาณ Trig Logic '0' ออกมาที่ขา I/O ในตำแหน่งของไฟล์เสียงที่ต้องการจะเล่น โดยค้างไว้ประมาณ 20 ms ขึ้นไป แล้วจึง Set ขา I/O นั้นกลับไปเป็น Logic '1' เหมือนเดิม ไฟล์เสียงในแชนแนลที่ถูก Trig ก็จะถูกเล่น

- การต่อวงจรโดยใช้ **ET-BUSIO DC IN** เป็นตัวลดระดับแรงดัน (เหมาะสำหรับใช้ Trig 1 แชนแนล หรือมากกว่า)

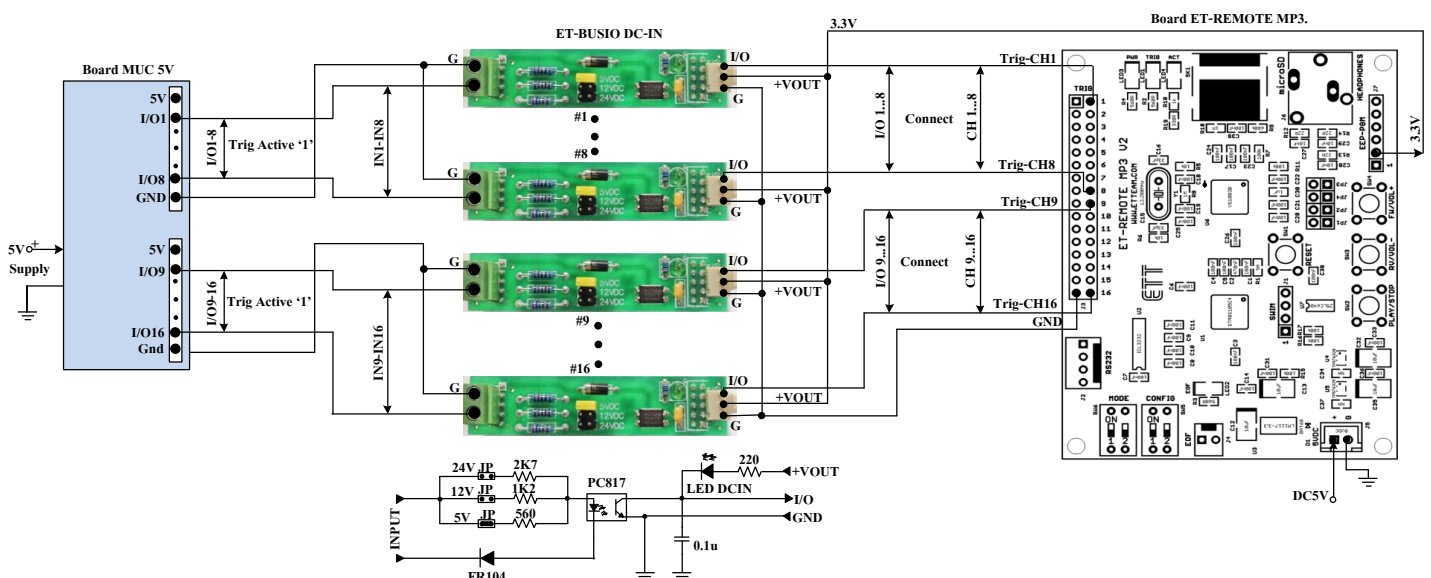
สำหรับบอร์ดลดระดับแรงดัน ET-BUSIO DC IN นี้จะมีลักษณะเป็นแชนแนลเดียว ทางด้าน Input สามารถเลือกรับสัญญาณ Trig จากภายนอกที่ระดับแรงดันต่างๆ ได้ 3 ระดับ คือ 5V, 12V และ 24V ซึ่งสามารถเลือกได้จาก Jumper บนตัวบอร์ด ET-BUSIO DC IN เอง นอกจากนั้นตัวบอร์ดยังสามารถรับ สัญญาณ Trig จากภายนอกได้ทั้งแบบ Active '0' หรือ Active '1' อีกด้วย โดยวงจรการต่อจะเป็นดังรูป(A) ถ้าใช้สัญญาณ Trig จากภายนอกที่ Active '0' และ วงจรการต่อจะเป็นดังรูป (B) ถ้าใช้สัญญาณ Trig จากภายนอกที่ Active '1' การต่อวงจรสามารถเลือกต่อ

แขนแนลที่จะใช้งานเท่านั้นก็ได้ ไม่จำเป็นต้องต่อครบทุกแขนแนล เพียงแต่ใน Micro SD Card จะต้องมีย่านไฟล์อย่างน้อยเท่ากับหมายเลขแขนแนลที่จะเลือกใช้งาน เช่น ต้องการใช้งานในแขนแนลที่ 9 ก็ควรมีไฟล์เสียงใน Card อย่างน้อย 9 ไฟล์ เป็นต้น

สำหรับการเขียนโปรแกรมใช้งาน จากรูปถ้าเราต่อวงจรตามรูป(A) ก็คือ จะเป็นการใช้สัญญาณ Trig จากภายนอกที่ Active '0' โดยในสภาวะปกติไม่มีการ Trig , MCU จะต้องส่งสัญญาณให้ I/O แต่ละขาเป็น 1 เมื่อต้องการสั่งให้เล่นไฟล์เสียง ก็ให้ส่งสัญญาณ Trig Logic '0' ออกมาที่ขา I/O ในตำแหน่งของไฟล์เสียงที่ต้องการจะเล่น โดยค้างไว้ประมาณ 20 ms ขึ้นไป แล้วจึง Set ขา I/O นั้นกลับมาเป็น Logic '1' เหมือนเดิม ไฟล์เสียงก็จะถูกเล่น แต่ถ้าเราต่อวงจรตามรูป (B) ก็คือ จะเป็นการใช้สัญญาณ Trig จากภายนอกที่ Active '1' โดยในสภาวะปกติไม่มีการ Trig MCU จะต้องส่งสัญญาณให้ I/O แต่ละขาเป็น 0 เมื่อต้องการสั่งให้เล่นไฟล์เสียงใด ก็ให้ส่งสัญญาณ Trig Logic '1' ออกมาที่ขา I/O ในตำแหน่งของไฟล์เสียงที่ต้องการจะเล่น โดยค้างไว้ประมาณ 20 ms ขึ้นไป แล้วจึง Set ขา I/O นั้นกลับมาเป็น Logic '0' เหมือนเดิม



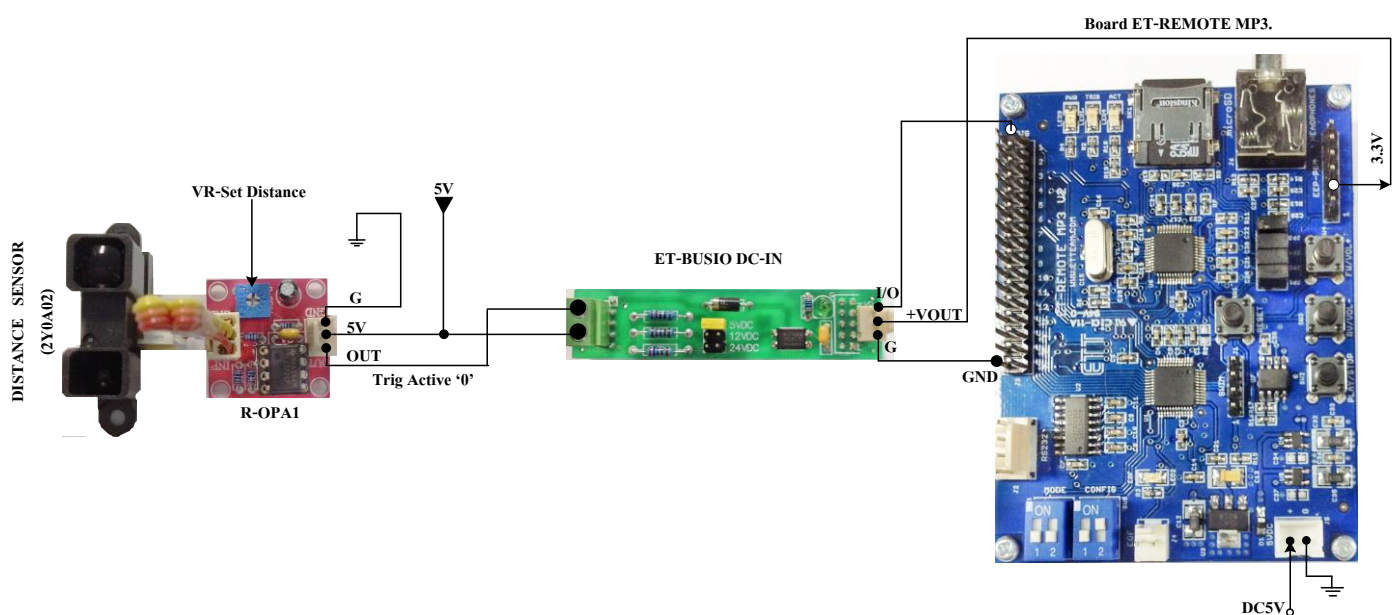
(A) รูปการต่อขา I/O 1-16 ของบอร์ด MCU 5V สำหรับ Trig (Active '0') ผ่านบอร์ดระดับแรงดัน ET-BUSIO DC-IN



(B) รูปการต่อขา I/O 1-16 ของบอร์ด MCU 5V สำหรับ Trig (Active '1') ผ่านบอร์ดระดับแรงดัน ET-BUSIO DC-IN

3) ตัวอย่างการต่อใช้งานโดยใช้ Sensor วัดระยะทางเป็นตัวส่งสัญญาณ Trig

สำหรับในหัวข้อนี้จะเป็นการประยุกต์การใช้งานใน Trig Mode โดยนำตัว Sensor วัดระยะทาง ซึ่งทำงานที่ไฟเลี้ยง 5 V (#2Y0A02) มาเป็นตัวส่งสัญญาณ Trig จากภายนอกแทน MCU เพื่อควบคุมการเล่นไฟล์เสียง โดยจะต่อพ่วงกับบอร์ด R-OPA1 ของอีทีที ซึ่งทำหน้าที่แปลงสัญญาณ Analog ที่ได้จากตัว Sensor วัดระยะทาง ให้เป็นสัญญาณ ดิจิตอล '0' หรือ '1' เพื่อนำมาใช้เป็นสัญญาณ Trig โดย Output จาก R-OPA1 นี้ในสภาวะปกติจะเป็น '1' และเมื่อมีวัตถุมาบังตัว Sensor ในระยะทางที่ตั้งค่าไว้ จะได้ Output ออกมาเป็น Logic '0' ดังนั้นสรุปได้ว่า สัญญาณ Trig Active ที่ Logic '0' ในส่วนของบอร์ดลดระดับแรงดันจะใช้ ET-BUSIO DCIN เป็นตัวลดระดับแรงดันของสัญญาณ Trig จาก 5V ลงมาเหลือ 3.3V ก่อนที่จะต่อสัญญาณไปยังขารับสัญญาณ Trig ของบอร์ด MP3 วงจรแสดงดังรูปด้านล่าง



รูปแสดงการต่อ Distance Sensor ควบคุมการเล่นไฟล์ของบอร์ด MP3 ใน Trig Mode

จากรูปวงจรข้างต้นในส่วนของการจ่าย ไฟ 5V สำหรับเลี้ยงชุด Sensor และบอร์ด MP3 จะใช้แยกเป็น 2 ชุด เพื่อแยกกราวด์ดังรูปข้างบน หรือจะใช้ชุดเดียวแล้วต่อกราวด์ร่วมกันก็ได้ เริ่มต้นการใช้งานให้ต้องจตามรูปด้านบน โดยเราจะใช้ช่องสัญญาณ Trig แชนแนลที่ 1 ของบอร์ด MP3 ส่วนการ Set DIP-SW_CONFIG เพื่อเลือกรูปแบบการเล่นให้ผู้ใช้ทำการกำหนดได้ตามต้องการ จากนั้นให้ทำการตั้งค่า Sensor ก่อน โดยหาอะไรมาบัง Sensor ในระยะที่เราต้องการตรวจจจากนั้นปรับ VR ที่บอร์ด R-OPA1 จน LED บนบอร์ดติดถือว่าใช้ได้ จากนั้นก็ลองหาวัตถุมาตัดหน้า Sensor ดู สัญญาณจาก Sensor ก็จะถูกส่งไป Trig บอร์ด MP3 ในช่องแชนแนลที่ 1 ไฟล์เสียงในลำดับที่หนึ่งก็จะถูกเล่น เป็นต้น

5. อธิบายการทำงานของโปรแกรมตัวอย่างใน CD

สำหรับตัวอย่างใน CD นั้น จะเป็นตัวอย่างการควบคุมบอร์ด ET-REMOTE MP3 V2 ในโหมด COMMAND-MODE และ TRIG-MODE โดยใช้ MCU เป็นตัวควบคุมการทำงาน โดยแบ่งเป็น MCU 3.3V ได้แก่ เบอร์ #ARM7 LPC2138 และ #STM8S208 ส่วน MCU 5V ได้แก่ เบอร์ #AVRmega128 ,#PIC18F8722 และ #89C51RE2 โดยตัวอย่างโปรแกรมของ MCU แต่ละตัวนั้นจะเป็นโปรแกรมตัวอย่างเดียวกันหมด เวลาจะโหลดโปรแกรมทดสอบให้ดูรายละเอียดและวงจรการต่อใช้งานได้ที่ Comment บนหัวโปรแกรมของแต่ละโปรแกรมก่อน โดยการทำงานของโปรแกรมตัวอย่างแต่ละโปรแกรมอธิบายได้ดังนี้

5.1 ตัวอย่างโปรแกรมสำหรับ COMMAND_MODE

การต่อวงจรสำหรับ ตัวอย่างใน Command Mode ให้ต่อสาย RS232 จากบอร์ด MP3 ไปยังบอร์ด MCU ที่ใช้ส่งคำสั่ง โดยบอร์ด MCU ในส่วนขั้วต่อ RS232 จะต้องผ่าน Line Driver Max232 ก่อน ห้ามต่อตรงจากขา MCU

Ex1_SENT_CMM

สำหรับตัวอย่างนี้ จะเป็นการสั่งให้เล่นไฟล์เสียง ตั้งแต่ไฟล์ลำดับที่ 1 จนถึงลำดับที่15 แล้วกลับมาเริ่มต้นเล่นไฟล์ลำดับที่ 2 จนถึงลำดับที่15 ซ้ำเช่นนี้ไปเรื่อยๆ ดังนั้นในการ Run โปรแกรมผู้ใช้งานจะต้อง มีไฟล์เสียงใน Micro SD Card อย่างน้อย 15 ไฟล์

การทำงานของโปรแกรม จะเริ่มต้นด้วยการส่งคำสั่ง CF เพื่อกำหนด รูปแบบการเล่นให้กับบอร์ด MP3 แล้วตามด้วยคำสั่ง Set Volume จากนั้นจะส่งคำสั่ง P1 เพื่อเล่นไฟล์ในลำดับที่ 1 ในการส่งคำสั่งแต่ละครั้งจะมีการตรวจสอบ Echo Command เสมอ หลังจากไฟล์ลำดับที่ 1 เล่นจบลง ซึ่งตรวจสอบจากฟังก์ชัน Echo_EOF() โปรแกรมก็จะส่งคำสั่ง P1 เพื่อเล่นไฟล์ในลำดับที่2 ต่อไป และเมื่อทุกครั้งที่ตรวจสอบพบ EOF โปรแกรมก็จะส่งคำสั่ง P1 เล่นไฟล์ในลำดับต่อไปเรื่อยๆจนถึงลำดับที่15 ก็จะกลับมาเริ่มเล่นไฟล์ในลำดับที่2 ใหม่ เป็นเช่นนี้ซ้ำไปเรื่อยๆ

Ex2_Read_Play_Time

สำหรับตัวอย่างนี้จะเป็นการสั่งให้เล่นไฟล์เสียงชื่อ A23_FT1.MP3 เป็นเวลา 30 วินาที แล้วจึงสั่งให้เล่นไฟล์เสียงชื่อ A24_FT2.MP3 เป็นเวลา 1 นาที แล้วจึงวนกลับไปสั่งให้เล่นไฟล์เสียงชื่อ A23_FT1.MP3 อีกครั้งหนึ่งซ้ำเช่นนี้ไปเรื่อยๆ เมื่อจะ Run โปรแกรม ใน Micro SD Card ควรจะตั้งชื่อไฟล์เสียงตามที่กล่าวไปข้างต้น หรือ อาจจะแก้ไขชื่อไฟล์เสียงในโปรแกรมให้ตรงกับชื่อไฟล์ที่มีอยู่ใน Micro SD Card ก็ได้ โดยไฟล์เสียงควรมีความยาวมากกว่า 1 นาที

การทำงานของโปรแกรมจะเริ่มต้นด้วยการส่งคำสั่ง CF เพื่อกำหนดให้บอร์ด MP3 แสดงค่าเวลาการเล่นด้วย จากนั้นก็เริ่มส่งคำสั่งให้เล่นไฟล์โดยชื่อที่กำหนด และทำการตรวจสอบ Echo Command จากนั้นก็ทำการเรียกฟังก์ชัน RD_Time() เพื่ออ่านค่าเวลาการเล่น โดยค่าเวลาจะถูกเก็บไว้ที่ตัวแปร Buf_Time(0-4) โดย Buf_Time[0-1] จะเก็บค่านาที ส่วน Buf_Time[3-4] จะเก็บค่าวินาที โดยจะตรวจสอบว่าไฟล์เสียงที่สั่งเล่น เล่นครบ 30 วินาทีหรือยัง ถ้ายังก็จะวนกลับไปอ่านค่าเวลาใหม่ ถ้าครบแล้วก็จะทำการสั่งให้เล่นไฟล์เสียงตามชื่อที่ระบุเป็นไฟล์ที่2 ต่อไป จากนั้นก็จะทำการอ่านค่าเวลาการเล่นของไฟล์ที่2 จนพบว่าไฟล์ที่ 2 เล่นครบ 1 นาที โปรแกรมก็จะวนกลับไปสั่งให้เล่นไฟล์เสียงตามชื่อที่ระบุในตอนแรกใหม่ อีกครั้ง

5.2 ตัวอย่างโปรแกรมสำหรับ TRIG_MODE

การต่อวงจรสำหรับ ตัวอย่างใน Trig Mode สำหรับ MCU 5V จะต้องใช้ ET-MINI3TO5 เป็นตัวลดระดับแรงดัน จากขา I/O ของ MCU ที่เป็น 5 V ให้เหลือ 3.3V ก่อนที่จะต่อไปยังขารับสัญญาณ Trig ของบอร์ด MP3 ถ้าใช้ MCU 3.3V สามารถต่อขา I/O ของ MCU ไปยังขารับสัญญาณ Trig ของบอร์ด MP3 ได้โดยตรง ส่วนการต่อขั้วสัญญาณ EOF จาก บอร์ด MP3 ไปใช้งาน สามารถนำไปต่อเข้ากับขา I/O ของ MCU ทั้ง 5 V และ 3.3 V ได้โดยตรงไม่ต้องผ่านวงจรใดๆ โดยจะต้องดึงขา G ไปด้วย ถ้า G ของบอร์ด MP3 และ MCU ไม่ได้ต่อกัน

Ex1_Trig_By_Cmm

สำหรับตัวอย่างนี้จะเป็นการรับ Command จากผู้ใช้ผ่าน RS232 ของบอร์ด MCU โดยคำสั่งจะมีรูปแบบดังตารางด้านล่าง เมื่อผู้ใช้ส่งคำสั่งมายังบอร์ด MCU ตัวบอร์ดก็จะส่งสัญญาณ Trig ออกไปที่ขา I/O ตามตำแหน่งขาที่ได้ระบุมาในคำสั่ง สัญญาณ Trig ก็จะถูกส่งไปยังขารับสัญญาณ Trig ของบอร์ด MP3 ไฟล์เสียงในลำดับแขนแนล ที่ถูก Trig ก็จะถูกลั่น

ASCII Command For Test

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

การทำงานจะเริ่มต้นด้วยการกำหนดขา I/O ที่ใช้ส่งสัญญาณ Trig ให้เป็น Logic 1 ทั้งหมดก่อน จากนั้นก็จะทำการตรวจสอบคำสั่งที่ส่งมาจากผู้ใช้ เมื่อคำสั่งถูกต้องก็จะส่งสัญญาณ Trig Logic 0 ของแขนแนลที่เลือกออกไปที่ขา I/O นั้น เป็นเวลา 20 ms และทำการ Set ขา I/O นั้นกลับมาเป็น Logic 1 เช่นเดิมและกลับมารับคำสั่งใหม่

Ex2_Trig_By_Check_EOF

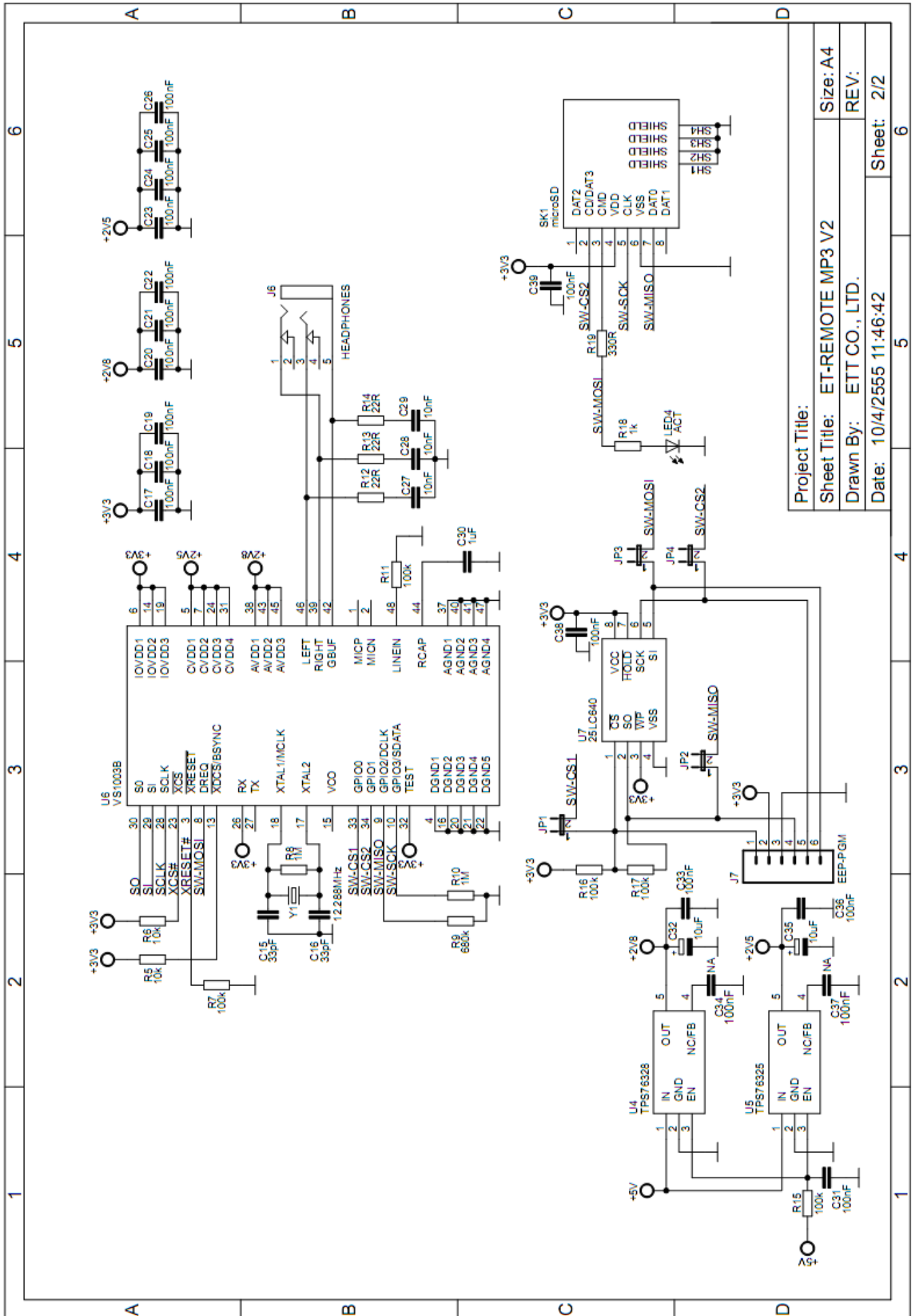
สำหรับตัวอย่างนี้จะเป็นการสั่งให้ MCU ส่งสัญญาณ Trig ไปยังขั้วต่อรับสัญญาณ Trig ของบอร์ด MP3 โดยให้เริ่ม Trig ตั้งแต่แขนแนลที่ 1-16 ไฟล์เสียงก็จะถูกเริ่มเล่นตั้งแต่ไฟล์หมายเลขลำดับที่ 1 จนถึงลำดับที่ 16 ตามจังหวะของสัญญาณ Trig เมื่อ Trig ครบ 16 แขนแนล โปรแกรมก็จะวนกลับมาเริ่ม Trig ในแขนแนลที่ 1 ใหม่ ดังนั้นในการทดสอบโปรแกรมใน Micro SD Card ควรจะมีไฟล์เสียง อย่างน้อย 16 ไฟล์

การทำงานจะเริ่มต้นด้วยการกำหนดขา I/O ที่ใช้ส่งสัญญาณ Trig ให้เป็น Logic 1 ทั้งหมดก่อน จากนั้นก็จะเรียกฟังก์ชัน Play_File() โดยฟังก์ชันนี้จะเริ่มทำการส่งสัญญาณ Trig Logic 0 ออกไปที่ขา I/O ของ MCU ที่ต่ออยู่กับช่องรับสัญญาณ Trig แขนแนล 1 ของบอร์ด MP3 จากนั้นก็จะ Delay ไว้ 20 ms แล้ว Set ขา I/O นั้นกลับมาเป็น Logic 1 เหมือนเดิม แล้วจึงทำการอ่านสัญญาณ EOF จากบอร์ด MP3 ว่าเป็น Logic 0 หรือยัง ถ้าเป็น Logic 0 แสดงว่าไฟล์เสียงที่ถูก Trig เล่นจบแล้ว โปรแกรมก็จะทำการ Trig เล่น ไฟล์เสียงในแขนแนลที่ 2 ต่อไป ทุกครั้งที่ไฟล์เสียงเล่นจบลงก็จะส่งสัญญาณ Trig ในแขนแนลต่อไปเรื่อยๆจนครบ 16 แขนแนล แล้วจึงวนกลับมาเริ่มต้น Trig ในแขนแนลที่ 1 ใหม่เป็นเช่นนี้ไปเรื่อยๆ

***** MANUAL ET-REMOTE MP3 V2.0 *****

By Mr.Sittiphol Yooyod.

www.ett.co.th



Project Title:

Sheet Title: ET-REMOTE MP3 V2

Drawn By: ETT CO., LTD.

Date: 10/4/2555 11:46:42

Size: A4

REV:

Sheet: 2/2