

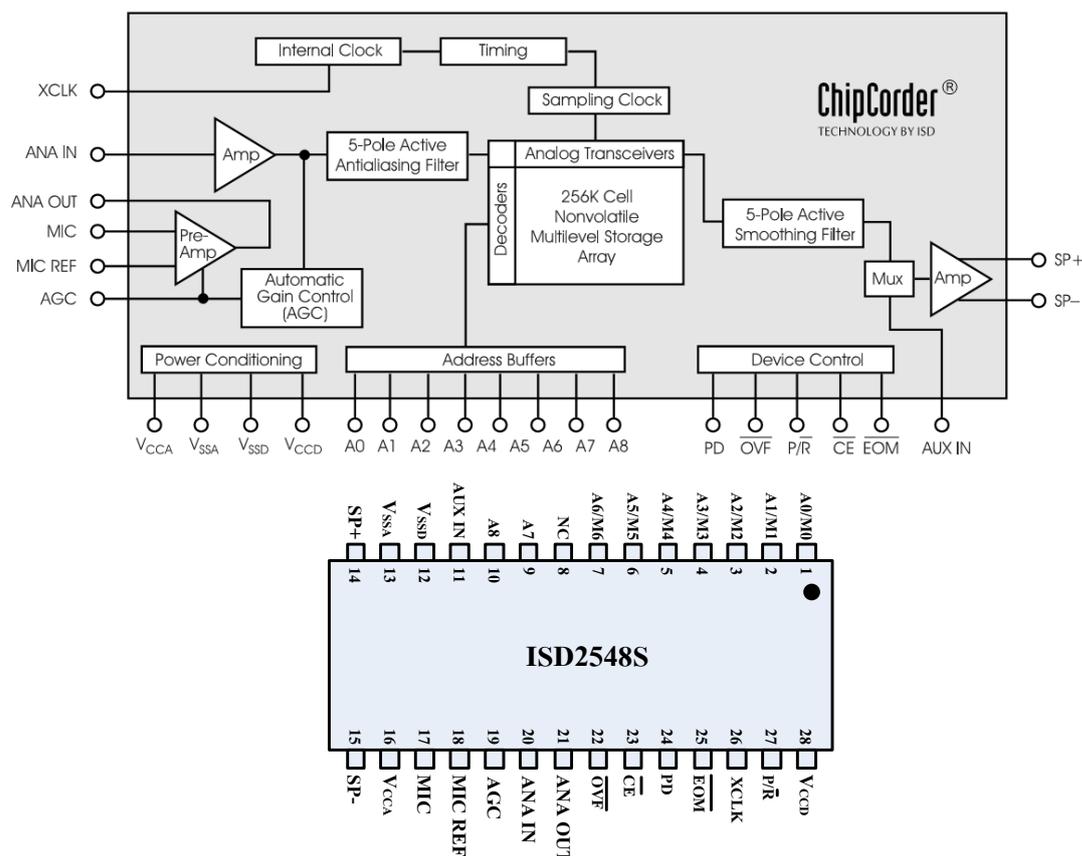
ET- MINI ISD2548

ET-MINI ISD2548 เป็นบอร์ดที่ใช้สำหรับบันทึกเสียง และเล่นกลับเสียงที่บันทึกไว้ เหมาะสำหรับบันทึกเสียง ประเภทข้อความที่มีความยาวไม่เกิน 48 วินาที/บอร์ด ควบคุมการบันทึกและการเล่นกลับโดยใช้ MCU Interface แบบ Serial Data (SPI) ผ่าน IC #74HC595 เพื่อลดจำนวนการใช้งานขา I/O ของ MCU ในการควบคุมการทำงานของ ISD2548 ให้น้อยลง (โดยปกติ ISD2548 จะควบคุมการทำงานแบบขนานซึ่งจะสิ้นเปลืองขา I/O)

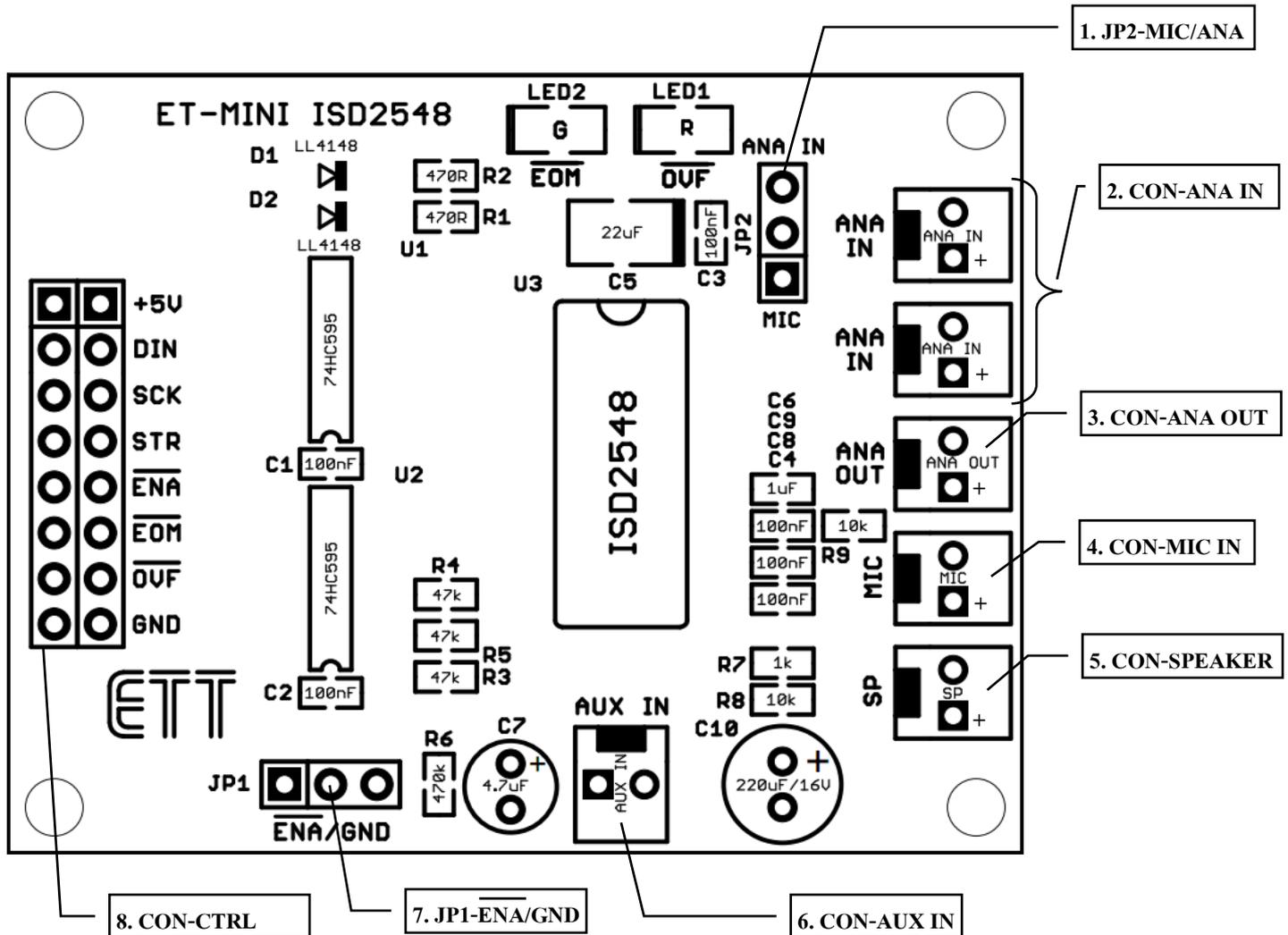
1. คุณสมบัติของบอร์ด ET-MINI ISD2548

- สะดวกต่อการใช้งาน ในการบันทึกเสียง และ เล่นกลับเสียงที่บันทึกไว้
- ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์
- ในแอดเดรสโหมดสามารถกำหนดตำแหน่งการเริ่มบันทึกหรือเริ่มเล่นได้ จากขา Address ของตัว Chip (A0-A8)
- สามารถบันทึกเสียงได้ความยาวสูงสุด 48 วินาทีต่อ Chip หนึ่งตัว
- มีจำนวน Message Address 320 แอดเดรส ดังนั้น 1 Address จะบันทึกเสียงได้ 150 msec หรือ 0.15 วินาที
- Input Sample Rate 5.3 KHz , Filter Pass Band 2.3 KHz (On-Chip Clock Source)
- สัญญาณเสียง Output สามารถต่อหูฟังได้ หรือ ผ่านชุดขยายสัญญาณก็ได้
- เสียงที่บันทึกเก็บไว้จะไม่ถูกลบเมื่อ Power off โดยไม่จำเป็นต้องใช้แบตเตอรี่สำหรับ Backup
- สามารถบันทึกเสียงได้ 100,000 ครั้ง และเก็บเสียงนั้นไว้ได้นานถึง 100 ปี
- ไฟเลี้ยงบอร์ด ET-MINI ISD2548 = 5 VDC
- อุณหภูมิการทำงานอยู่ในช่วง -40 ถึง +85 องศาเซลเซียส

2. โครงสร้างและบล็อกไดอะแกรมของ Chip ISD2548



3. โครงสร้างของบอร์ด ET-MINI ISD2548



1. **JP2-MIC/ANA** : เป็น Jumper ใช้สำหรับเลือกช่องสัญญาณ INPUT ที่ต้องการจะบันทึกเสียง โดยเลือกมาทาง ANA-IN จะเป็นการเลือกบันทึกสัญญาณ Line Out จากภายนอกที่ต่อเข้ามาทางขั้วต่อ ANA IN (หมายเลข2) ช่องใดก็ได้เนื่องจากต่อขนานกันอยู่ ถ้าเลือกมาทางด้าน MIC จะเป็นการเลือกบันทึกสัญญาณจากไมโครโฟนที่ต่อเข้ามาทางขั้วต่อ MIC (หมายเลข4)
2. **CON-ANA IN** : เป็นขั้วต่อ Input รับสัญญาณเสียง Analog จากอุปกรณ์กำเนิดเสียงภายนอกเพื่อจะบันทึกเก็บไว้ใน ISD2548 โดยสามารถเลือกต่อเข้าที่ขั้วใดขั้วหนึ่งก็ได้ เนื่องจากทั้ง 2 ขั้วต่อขนานกันอยู่
3. **CON-ANA OUT** : เป็นขั้วต่อ Output ของ Pre-Amp ที่รับสัญญาณ Input มาจากทางช่อง MIC ซึ่งสัญญาณ Output นี้สามารถนำไปต่อเข้ากับภาค Amplifier เพื่อขยายสัญญาณให้แรงขึ้นได้
4. **CON-MIC IN** : เป็นขั้วต่อไมโครโฟน ใช้สำหรับบันทึกเสียงจากผู้ใช้งานเก็บไว้ที่ ISD2548
5. **CON-SPEAKER** : เป็นขั้วต่อ Output Amplifier ใช้สำหรับต่อหูฟัง หรือต่อไปยังชุดขยายเสียงภายนอกซึ่งสัญญาณ Output นี้จะเป็นสัญญาณเสียงที่เกิดจากการเล่นกลับของเสียงที่ผู้ใช้บันทึกไว้ หรือ เสียงที่ส่งมาจากขั้วต่อ AUX IN

6. **CON-AUX IN** : เป็นขั้วต่อ Input สำหรับรับสัญญาณเสียงจากภายนอกเข้ามาขยายแล้วส่งออกที่ Output Amplifier ก็คือส่งออกที่ขั้ว CON-SPEAKER จะไม่ถูกบันทึกลงในตัว ISD2548
7. **JPI-ENA/GND** : เป็น Jumper สำหรับเลือกการควบคุมขั้ว \overline{ENA} ของ Shift Register 74HC595 ทั้ง 2 ตัวบนบอร์ด โดย ถ้า Set มาทางด้าน GND จะทำให้ Shift Register ทั้ง 2 ตัว ถูก Enable หรือพร้อมทำงาน ตลอดเวลา แต่ถ้า Set มาทางด้าน \overline{ENA} จะให้ผู้ใช้ควบคุมการ Enable ของ Shift Register ทั้ง 2 ตัว ด้วยตัวเอง ผ่านขั้วต่อ CON-CTRL ที่ Pin \overline{ENA}
8. **CON-CTRL** : เป็นขั้วต่อสำหรับต่อสัญญาณจาก MCU มาควบคุมการทำงานของ ISD2548 รวมทั้งอ่านค่าสถานะ การทำงาน ของ ISD2548 ไปใช้งาน โดยรายละเอียดของแต่ละขาเป็นดังนี้
- **+5V และ GND** : ใช้สำหรับต่อไฟเลี้ยงตัวบอร์ด DC 5 V
 - **DIN** : เป็นขารับ Data แบบ Serial (SPI) จาก MCU ขนาด 8 บิต
 - **SCK** : เป็นขารับสัญญาณ Clock (SPI) จาก MCU เพื่อใช้เลื่อน data ใน Shift Register โดยทำงานที่ขอบขาขึ้น
 - **STR** : เป็นขารับสัญญาณ Strobe จาก MCU ทำงานที่ Logic '1' ใช้ในการปล่อยข้อมูลที่รับเข้ามา แบบ Serial จาก Shift Register ทั้ง 2 ตัว ให้ออกมาทาง Output แบบขนานเพื่อส่งไปยัง Pin ต่างๆ ของ ISD2548
 - **\overline{ENA}** : เป็นขารับสัญญาณ \overline{ENA} จาก MCU เพื่อใช้ Enable การทำงานของ Shift Register ทั้ง 2 ตัว โดย Active '0'
 - **\overline{EOM}** : เป็นขา Output บอกสถานะ การเล่นข้อความเสียงจบลง โดยตัว ISD2548 จะส่งสัญญาณ \overline{EOM} ออกมาให้ที่ขานี้เพื่อให้ MCU อ่านค่าไปใช้งาน ในสถานะปกติขานี้จะให้ Logic = '1' แต่ถ้ามีการเล่นข้อความเสียงจบลงขานี้จะให้ Logic = '0' ออกมา 18.75 ms แล้วกลับเป็น '1' อัดโนมัติ (จะต้องต่อ R-Pull Up 10K ไว้ที่ขานี้ด้วยเวลาจะใช้งาน)
 - **\overline{OVF}** : เป็นขา Output บอกสถานะ หน่วยความจำของ ISD2548 เต็ม โดยตัว ISD2548 จะส่งสัญญาณ \overline{OVF} ออกมาให้ที่ขานี้เพื่อให้ MCU อ่านค่าไปใช้งาน ในสถานะปกติขานี้จะให้ Logic = '1' เมื่อหน่วยความจำเต็มขานี้จะให้ Logic = '0' ค้างไว้ เมื่อ Reset ISD2548 ขานี้ก็จะกลับมาเป็น '1' ปกติ (จะต้องต่อ R-Pull Up 10K ไว้ที่ขานี้ด้วยเวลาจะใช้งาน)

4. การทำงานเบื้องต้นของ ISD2548

สำหรับตัว ISD2548 นั้นจะสามารถบันทึกเสียงได้ 48 วินาที และ มีจำนวน Message Address อยู่ที่ 320 แอดเดรส ดังนั้น เราจะได้ 1 Message Address สามารถบันทึก เสียงได้เท่ากับ $48/320 = 0.15$ วินาที ซึ่งค่าที่กล่าวมาข้างต้นนี้จะมี ความสำคัญเมื่อเราใช้ MCU เป็นตัวควบคุมการเล่นและการบันทึกเสียง

ISD2548 นั้นสามารถเลือกโหมดการทำงานได้ 2 Mode ใหญ่ๆ คือ Address Mode และ M-Mode โดยใน M-Mode จะมีโหมดย่อยอยู่หนึ่งโหมดคือ Push-Button Mode ซึ่งสามารถใช้การควบคุมรูปแบบการเล่นข้อความเสียงร่วมกับ M-Mode ได้ แต่ในคู่มือนี้จะขอกล่าวถึงการใช้งาน ISD2548 ใน Address Mode และ Push-Button Mode เท่านั้นซึ่งจะสอดคล้องกับ ตัวอย่างที่ให้มาใน CD ส่วนอุปกรณ์ที่จะใช้ควบคุมการบันทึกและการเล่นกลับข้อความเสียงนั้นจะขอกล่าวถึงการ ใช้ MCU เป็นตัวควบคุมเท่านั้น การทำงานนอกเหนือจากนี้สามารถดูได้จาก Data Sheet ของ ISD2548

4.1) การทำงานใน Address Mode

สำหรับในโหมดนี้ เราจะใช้ขา A0-A8 เป็นตัวกำหนดแอดเดรสในการเล่นกลับและการบันทึกเสียง โดยขา A7 และ A8 จะเป็นตัวเลือกโหมดการทำงานโดยต้องกำหนดให้เป็น 00 หรือ 01 หรือ 10 ตัว ISD2548 ถึงจะทำงานในโหมดนี้ ซึ่งถ้าเราสังเกตว่า Message Address ของ ISD2548 จะมีอยู่ 320 Address ดังนั้นเราจะสามารถอ้างแอดเดรสได้ตั้งแต่ 0-319 หรือ 00000000b - 10011111b (0x000-0x13F) ซึ่งจะเห็นว่าในตำแหน่งแอดเดรสสูงสุด ยังคงทำให้ บิต A7 และ A8 อยู่ใน 3 สถานะคือ 00,01,10 ดังที่กล่าวไปข้างต้น ดังนั้นการใช้งานโหมดนี้เพียงคำหนึ่งคำว่า เวลาอ้างแอดเดรสในการเล่นกลับหรือบันทึก อย่างอื่นให้เกินค่า 0x31F เท่านั้นก็จะถือว่าทำงานอยู่ในโหมดนี้เสมอ ส่วนขาที่จะใช้ควบคุมการบันทึกหรือการเล่นจะมีอยู่ 3 ขา คือ 1) \overline{CE} = จะเป็นขารับสัญญาณ Pulse Active Low (ปกติควรให้เป็น High ไว้) จะทำหน้าที่ควบคุมการ Start การเล่นกลับ หรือ การบันทึก

2) PD = เป็นขารับสัญญาณ Pulse Active High (ปกติควรให้เป็น Low ไว้) จะทำหน้าที่ Reset ตัว ISD2548 ถ้ามีการเล่นกลับหรือ บันทึกอยู่ก็จะถูกสั่งให้หยุดทันที และตำแหน่ง Message-Address ก็จะถูกชี้ให้มาอยู่ในตำแหน่งที่ผู้ใช้ได้กำหนดเอาไว้ที่ ขา A0-A8 ขณะนั้น

3) P/\overline{R} = เป็นขาสำหรับเลือกการบันทึกหรือการเล่นกลับ โดยถ้าขานี้เป็น High จะเป็นการเลือก ‘การเล่นกลับ’ แต่ถ้าเป็น Low จะเป็นการเลือก ‘การบันทึก’

ส่วนขาที่ใช้บอกสถานะการทำงานของ ISD2548 จะมีอยู่ด้วยกัน 2 ขาได้แก่

- 1) \overline{EOM} = ทุกครั้งที่มีการเล่นข้อความจบลงตัว ISD2548 จะส่งสัญญาณ Pulse Low ขนาดความกว้าง 18.75 ms ออกมาที่ขา \overline{EOM} เพื่อบอกให้ทราบว่าข้อความเล่นจบลงแล้ว และสัญญาณ Pulse \overline{EOM} ก็จะกลับไปเป็น High อัตโนมัติ
- 2) \overline{OVF} = สัญญาณนี้จะเป็น Pulse Low เมื่อพื้นที่หน่วยความจำของ ISD 2548 เต็ม หรือมีการบันทึกเสียงจนเต็มหน่วยความจำของ ISD2548 และสัญญาณนี้จะเป็น Low ไปจนกระทั่งมีการ Reset ISD2548 ที่ขา PD สัญญาณนี้ก็จะกลับมาเป็น High เหมือนเดิม

4.1.1) ขั้นตอนการบันทึก (Recording) ใน Address Mode

- 1) กำหนด Address เริ่มต้นที่ต้องการจะใช้บันทึกเสียง ที่ขา A0-A8 โดยสามารถ Set Address ได้ในช่วง 0x000-0x13F เช่น ต้องการเริ่มต้นบันทึกที่ Address 0x005 ก็ให้ขา A0,A2 = ‘1’ ส่วนขา Address ที่เหลือให้เป็น ‘0’
- 2) Set ขา PD ไปเป็น Low เพื่อออกจากสถานะ Reset ISD2548 จะพร้อมเริ่มทำงาน
- 3) Set ขา P/\overline{R} ไปเป็น Low เพื่อเลือกใช้งานการบันทึกเสียง
- 4) Set ขา \overline{CE} ไปเป็น Low ค้างไว้ พร้อมกับป้อนสัญญาณเสียงที่จะบันทึก การบันทึกก็จะเริ่มขึ้น โดยจะเริ่มบันทึกเสียงลงในตำแหน่ง Message Address ที่ผู้ใช้ได้กำหนดไว้ในข้อ 1 ซึ่ง Message Address นี้เมื่อเก็บเสียงเต็มในแต่ละ Address แล้ว มันจะเพิ่ม Address ขึ้นอัตโนมัติถ้ายังมีการบันทึกอยู่ (\overline{CE} ยังคงเป็น Low อยู่) โดยจากที่กล่าวไว้ข้างต้นใน 1 Address จะเก็บเสียงได้ 0.15 วินาที ถ้าเสียงที่อัดเต็มหน่วยความจำของตัว ISD2548 แล้ว มันจะส่งสัญญาณ Overflow เป็น Pulse Low ออกมาที่ขา OVF เพื่อบอกให้ผู้ใช้ทราบว่า หน่วยความจำเต็มแล้ว เสียงที่ส่งมาหลังจากเกิดสถานะนี้จะไม่มีความหมาย
- 5) เมื่อจะหยุดบันทึก ก็ให้ Set ขา \overline{CE} กลับมาเป็น High และสัญญาณ \overline{EOM} จะถูกบันทึกลงใน Address หน่วยความจำต่อจากข้อความที่เราบันทึก เมื่อเราต้องการจะบันทึกข้อความอื่นๆอีกในแอดเดรสต่อไป ก็ให้กลับไปเริ่มจากขั้นตอนที่ 2 ใหม่ โดยเราต้องคำนึงว่า Address ที่จะกำหนดเพื่อบันทึกข้อความต่อไป จะต้องไม่ไปทับช่วง Address ของข้อความที่บันทึกไปแล้ว

มีฉะนั้นจะทำให้ข้อความที่บันทึกก่อนหน้าบางส่วนเสียหายไปได้

ในการบันทึกข้อความสั้นๆหลายๆข้อความลงใน ISD2548 ตัวเดียวกันนั้นผู้ใช้สามารถกำหนด Address เริ่มต้นที่จะบันทึกได้ แต่จะไม่สามารถทราบได้ว่าข้อความที่บันทึกจบที่ Address ใด ดังนั้นจะทำให้ยากต่อการกำหนด Address ของข้อความต่อไป จะขอแนะนำวิธีการบันทึกโดยให้ผู้ใช้กำหนดเวลาการบันทึกในแต่ละข้อความเท่าๆกัน แล้วใช้การจับเวลาที่บันทึกเมื่อถึงเวลาที่กำหนดก็ให้สั่งหยุดบันทึกทันที เพื่อให้ผู้ใช้สามารถกำหนด Address เริ่มต้นของข้อความต่อไปได้โดยแอดเดรสไม่ทับกัน(ดูตัวอย่างได้จาก CD ในตัวอย่าง “Ex1_Play_Record”) เช่น ต้องการบันทึกข้อความยาวไม่เกิน 2 วินาที จากคุณสมบัติ ISD2548 ใน 1 Address สามารถบันทึกได้ 0.15 วินาที ดังนั้นในหนึ่งข้อความจะต้องใช้ Address เท่ากับ $2/0.15 \sim 14$ Address ถ้าเราเริ่มต้นบันทึกข้อความแรกที่ Address 0 ข้อความที่ 2 ก็ควรจะเริ่มต้นที่ Address 15 ข้อความที่ 3 ก็ควรเริ่มต้นที่ Address 30 เป็นต้น สังเกตว่าเมื่อเราเริ่มต้นที่ Address 0 เมื่อเราบวกไปอีก 14 Address ข้อความจะต้องสิ้นสุดไม่เกิน Address 13 ให้เราบวกเพิ่มเข้าไปอีก 1 Address สำหรับสัญญาณ \overline{EOM} ดังนั้นในข้อความแรกจะต้องสิ้นสุดที่ Address 14 และ Address ของข้อความที่ 2 ก็ต้องเริ่มต้นที่ Address 15 เป็นต้น อย่าลืมบวกเพิ่มอีก 1 Address เสมอในแต่ละข้อความสำหรับเก็บสัญญาณ \overline{EOM} มีฉะนั้นถ้า สัญญาณ \overline{EOM} ถูกบันทึกทับลงไปจะทำให้เวลาเล่นกลับจะแยกจุดสิ้นสุดของแต่ละข้อความไม่ได้ สรุปการคำนวณหาจำนวน Address ที่ใช้ ในความยาวของข้อความที่ต้องการ ได้ดังนี้

$$\text{จำนวน Address ที่ใช้} = \left[\frac{\text{ความยาวของข้อความ(วินาที)}}{0.15(\text{วินาที})} \right] + 1 (\text{EOM Address})$$

ในการเขียนโปรแกรมเมื่อผู้ใช้เขียนโปรแกรมมาถึงขั้นตอนที่ 4 คือ $\overline{CE} = \text{Low}$ การบันทึกเริ่มขึ้น ให้ delay เป็นเวลา 2 วินาทีตามความยาวของข้อความที่เรากำหนด เมื่อครบ 2 วินาที ก็ให้ทำในขั้นตอนที่ 6 เพื่อหยุดการบันทึก ($\overline{CE} = \text{High}$) เป็นต้น

4.1.2) ขั้นตอนการเล่นกลับ (Playback) ใน Address Mode

1) กำหนด Address เริ่มต้นที่ต้องการจะเล่นกลับเสียง ที่ขา A0-A8 โดยสามารถ Set Address ได้ในช่วง 0x000-0x13F เช่น ต้องการเริ่มต้นเล่นกลับที่ Address 0x005 ก็ให้ขา A0,A2 = '1' ส่วนขา Address ที่เหลือให้เป็น '0'

2) Set ขา PD ไปเป็น Low เพื่อออกจากสภาวะ Reset ISD2548 จะพร้อมเริ่มทำงาน

3) Set ขา $\overline{P/R}$ ไปเป็น High เพื่อเลือกใช้งานการเล่นกลับ

4) ทำการ Set ขา \overline{CE} ซึ่งการ Set ขา \overline{CE} นี้จะมีผลต่อการเล่นกลับอยู่ 2 กรณี คือ

- Set ขา \overline{CE} เป็น Low ค้างไว้ (ปกติเป็น High) ตำแหน่งข้อความที่ถูกชี้โดยขา Address ที่ผู้ใช้กำหนดในข้อ 2 ก็จะถูกเล่นทันที ในขณะที่เล่นขาสัญญาณ \overline{EOM} จะมีสถานะเป็น High เมื่อข้อความถูกเล่น จบลง ตัว ISD2548 ก็จะส่งสัญญาณ Logic Low ออกมาที่ขา \overline{EOM} เป็นเวลา 18.75 ms และกลับไปเป็น High ใหม่ ซึ่งผู้ใช้สามารถตรวจสอบการจบของข้อความได้จากขา สัญญาณนี้ เมื่อข้อความแรกจบลง ข้อความที่ 2 ที่ 3 ก็จะถูกเล่นต่อเนื่องไปเรื่อยๆ และสัญญาณ \overline{EOM} Logic Low ก็จะถูกส่งออกมาเมื่อเล่นจบในแต่ละข้อความ ข้อความจะถูกเล่นไปจนกระทั่ง ขาสัญญาณ \overline{CE} ถูก Set กลับมาเป็น High หรือเล่นไปจนสิ้นสุด Address ของหน่วยความจำ ISD2548 การเล่นก็จะสิ้นสุดลง ถ้าขา \overline{CE} ถูก Set กลับมาเป็น High ในระหว่างที่มีข้อความเล่นอยู่ ข้อความที่เล่นอยู่จะถูกเล่นจนจบ การเล่นถึงจะหยุดลง เมื่อการเล่นหยุดลง และมีการ Set \overline{CE} ไปเป็น Low ใหม่ ข้อความแรกที่ถูกชี้โดยขา Address ที่ผู้ใช้กำหนดในข้อ 1 ก็จะถูกเริ่มเล่นใหม่

- Set ขา \overline{CE} ในลักษณะของ Pulse คือ Set ขา \overline{CE} จากที่เป็น High อยู่ ให้เป็น Low และ Set กลับมาให้เป็น High ใหม่ในทันที ทำให้ข้อความที่ถูกชี้ตำแหน่งโดยขา Address ที่ผู้ใช้กำหนดในข้อ 1 ถูกเล่นทันที ในขณะที่เสียงถูกเล่นขาสัญญาณ \overline{EOM} จะมีสถานะเป็น High เมื่อข้อความถูกเล่น จบลง ตัว ISD2548 ก็จะส่งสัญญาณ Logic Low ออกมาที่ขา

\overline{EOM} เป็น เวลา 18.75 ms และกลับไปเป็น High ใหม่ และการเล่นก็จะสิ้นสุดลง เมื่อต้องการให้เล่นข้อความต่อไปผู้ใช้จะต้องตรวจสอบสัญญาณ \overline{EOM} ให้กลับมาเป็น High ก่อน แล้วจึงไปเริ่มทำในขั้นตอนที่ 1 ใหม่เพื่อเล่นข้อความต่อไป

5) การสั่งหยุดเล่นข้อความระหว่างเล่นก็ให้ Set ขา PD ไปเป็น High ซึ่งจะทำให้ ISD 2548 อยู่ในสภาวะ Reset

4.2) การทำงานใน Push-Button Mode (Mode ย่อยของ M-Mode)

การใช้งานในโหมดนี้จะต้องกำหนดให้ขา A6,A7,A8 เป็น High ส่วนขา A0-A5 ก็จะเปลี่ยนหน้าที่ไปเป็นขา M0-M5 โดยแต่ละขาจะทำหน้าที่ต่างกันไป ซึ่งปกติแล้วเวลาต่อใช้งานขา M0-M5 เราควร Full Down ไว้ เมื่อเราต้องการให้ขาไหนทำงานตามหน้าที่ของมันนั้นก็ให้ Set เป็น High ซึ่งนอกจากเราจะกำหนดให้ขาที่ต้องการใช้งาน Active เป็น High แล้วเรายังจะต้องมีขาควบคุมอื่นๆทำงานร่วมด้วย ซึ่งก็จะขึ้นอยู่กับการทำงานของแต่ละขาว่ามีรูปแบบการทำงานอย่างไร โดยจะไม่ขอกล่าวถึงในที่นี้ให้ดูรายละเอียดใน Data sheet

การทำงานใน Push-Button Mode นี้ การบันทึก และการเล่นกลับจะใช้ขาควบคุม \overline{CE} , PD, P/\overline{R} , \overline{EOM} ทำหน้าที่คล้ายๆกับใน Address-Mod แต่ละขาสัญญาณจะทำหน้าที่ดังนี้

- 1) ขา \overline{CE} = จะเป็นขาจับสัญญาณ Pulse Active Low (High -->Low) ใช้ควบคุมการ START หรือ PAUSE ในการเล่น และการบันทึก เริ่มต้นเมื่อนี้ได้รับ Plus Low การเล่นกลับหรือการบันทึกก็จะเริ่มขึ้น และในขณะที่มีการเล่นกลับอยู่ ถ้ามีการส่ง Pulse Low มาที่ขานี้ก่อนที่จะเกิดสัญญาณ \overline{EOM} หรือ \overline{OVF} การเล่นก็จะหยุดลงแบบ PAUSE ทันที และถ้าทำการส่ง Pulse Low มาอีก การเล่นเสียงก็จะกลับมาเล่นต่อจากตำแหน่งที่หยุด
- 2) ขา PD = จะเป็นขาจับสัญญาณ Pulse Active High (Low-->High) ใช้ควบคุมการ STOP หรือ RESET ในการเล่น และการบันทึก และเมื่อเกิดสภาวะ STOP/RESET ขึ้น การเล่นหรือการบันทึกก็จะหยุดลง และจะทำให้ Address Pointer ของ Message ถูก Reset ไปเป็น 0x000 ใหม่ ซึ่งถ้ามีการสั่งเล่นหรือบันทึกเสียงใหม่ เสียงก็จะกลับไปเริ่มเล่นหรือบันทึกที่ Address 0x000 อีกครั้ง ทุกครั้งที่ขานี้ถูก Active สภาวะ STOP/RESET จะเกิดขึ้นเสมอ และ Pointer Address ก็จะชี้ไปยังตำแหน่ง Address เริ่มต้นที่ 0x000 เสมอ เช่นกัน
- 3) ขา P/\overline{R} = เป็นขาสำหรับเลือกการบันทึกหรือการเล่นกลับ โดยถ้าขานี้เป็น High จะเป็นการเลือก ‘การเล่นกลับ’ แต่ถ้าเป็น Low จะเป็นการเลือก ‘การบันทึก’
- 4) ขา \overline{EOM} = เป็น Output ทำหน้าที่แสดงสถานะการทำงานของการเล่นกลับ และการบันทึก ซึ่งสามารถนำ LED มาต่อใช้แสดงผลการทำงานได้ โดยจะแสดงสถานะเป็น High เมื่อมีการเล่นกลับหรือการบันทึก และจะแสดงสถานะเป็น Low เมื่อไม่มีการบันทึกหรือการเล่นกลับ ส่วนการจัดการข้อความใน Mode นี้เราไม่สามารถอ้าง Address ได้โดยตรง เหมือนกับใน Address-Mode จะใช้การควบคุมหน้าที่ของขา M0-M5 เป็นตัวจัดการข้อความแทน เมื่อเริ่มต้นทำงานหรือมีการ Reset หรือมีการเปลี่ยนการใช้งานจากการบันทึกไปเป็น เล่นกลับ หรือจาก เล่นกลับมาเป็นบันทึก Address ที่ใช้อ้างอิงภายใน ISD 2548 ของโหมดนี้จะถูกกำหนดให้เริ่มต้นที่ Address 0x000 เสมอซึ่งจะไม่ได้ขึ้นอยู่กับขา A0-A8 แล้ว ต่อไปเราจะมาดูขั้นตอนการบันทึกและการเล่นกลับในโหมดนี้

4.2.1) ขั้นตอนการบันทึก (Recording) ใน Push-Button Mode

- 1) กำหนดขา A6-A8 ไปเป็น High เพื่อเลือกโหมด Push-Button ส่วนขา A0-A5 ควร Full Down หรือกำหนดให้เป็น Low
- 2) Set ขา PD ไปเป็น Low เพื่อออกจากสภาวะ Reset ISD2548 จะพร้อมเริ่มทำงาน
- 3) Set ขา P/\overline{R} ไปเป็น Low เพื่อเลือกใช้งานการบันทึกเสียง

- 4) Set ขา \overline{CE} ในลักษณะ Pulse Low คือ Set ขา \overline{CE} จากที่เป็น High อยู่ ให้เป็น Low และ Set กลับมาเป็น High ใหม่อีกครั้ง การบันทึกก็จะเริ่มต้นขึ้น สถานะของขา \overline{EOM} ปกติจะเป็น Low อยู่ก็จะกลายเป็น High แสดงสถานะการบันทึก โดยเสียงจะเริ่มถูกบันทึกที่ Address 0x000
- 5) เมื่อขา \overline{CE} ได้รับ Pulse Low อีกครั้ง การบันทึกก็จะหยุดลง สัญญาณที่ขา \overline{EOM} จากที่เป็น High อยู่ก็จะกลับมาเป็น Low เพื่อแสดงสถานะหยุดการบันทึก ส่วน Address pointer ของหน่วยความจำ จะยังไม่ถูก Clear หลังจากการบันทึกหยุดลง ในแต่ละครั้งค่า EOM Marker ก็จะถูกเก็บไว้ในหน่วยความจำ Address ที่ต่อจากข้อความที่บันทึกขณะนั้น ซึ่งค่า \overline{EOM} Marker นี้จะเป็นตัวบอกการจบของข้อความในแต่ละข้อความ ในช่วงเวลานี้ถ้าขา P/\overline{R} ถูก Set เป็น High สำหรับเล่นกลับ และ \overline{CE} ได้รับ Pulse Low การเล่นกลับก็ควรจะถูก Start ที่ Address 0x000 และในทางกลับกัน ถ้าเรา Set ขา P/\overline{R} กลับมาเป็น Low เพื่อบันทึกอีกครั้ง โดยที่ขา PD ยังคงเป็น Low อยู่ ตำแหน่ง Address ที่จะเริ่มถูกบันทึกจะเป็นตำแหน่ง Address ที่ต่อจากตำแหน่ง Address ของการเล่นกลับล่าสุด

ในกรณีที่เรบันทึกข้อความจนหน่วยความจำ ของ ISD2548 เต็ม สัญญาณที่ขา \overline{EOM} จากที่เป็น High อยู่ก็จะกลับมาเป็น Low โดยอัตโนมัติแสดงให้เห็นว่า การอัดสิ้นสุดลงและหน่วยความจำเต็มแล้ว โดยที่เราไม่ต้องส่ง Pulse Low ไป ที่ขา \overline{CE} เพื่อหยุดการบันทึกเองเหมือนกับข้อความที่บันทึกตอนแรกๆ

- 6) หลังจากที่มีการบันทึกหยุดลง ถ้าเราจ่าย Pulse Low ไปยังขา \overline{CE} อีก การบันทึกก็จะเริ่มขึ้นใหม่ และ Address สำหรับการบันทึกก็จะถูกชี้ไปที่เริ่มต้นที่ตำแหน่งต่อจาก Address ที่ใช้เก็บค่า \overline{EOM} Marker ของข้อความที่บันทึกก่อนหน้านี้ และสถานะของขา \overline{EOM} ก็จะเปลี่ยนจาก Low เป็น High เพื่อแสดงสถานะ การบันทึก เพื่อให้ข้อความถูกบันทึกในตำแหน่ง Address ต่อกันไปเรื่อยๆ ก็ให้กลับไปทำตามขั้นตอนที่ 5 และกลับมาทำตามขั้นตอนที่ 6 สลับกันไปเรื่อยๆ

ในการบันทึกของแต่ละข้อความหลังจากที่เราทำตามขั้นตอนที่ 1 ถึงขั้นตอนที่ 4 มาแล้ว ให้ทำตามขั้นตอนที่ 5 และ 6 สลับกันไปเรื่อยๆข้อความที่บันทึกจะถูกบันทึกในตำแหน่ง Address ที่ต่อกันไปเรื่อยๆและจะถูกค้นด้วย \overline{EOM} Marker เพื่อเป็นการบอก จุดสิ้นสุดของแต่ละข้อความ ถ้าในระหว่างการบันทึก หรือ หลังจากทำการบันทึกข้อความใดข้อความหนึ่งจบแล้ว ถ้า ISD2548 ถูก Reset ด้วยวิธีใดๆก็ตาม เช่น PD ถูก Set ให้เป็น High แล้วกลับมาเป็น Low ใหม่เป็นต้น หลังจากสถานะ Reset ถ้าเราเริ่มต้นการบันทึกอีกครั้ง ข้อความที่บันทึกจะกลับไปเริ่มต้นการบันทึกที่ Address 0x000 ใหม่เสมอ จะไม่บันทึกต่อจากตำแหน่งข้อความที่บันทึกก่อนหน้านี้ นั่นคือ ถ้า ISD2548 ถูก Reset จะทำให้ Address Pointer ของหน่วยความจำถูก Reset ตามไปด้วย และถ้ามีการบันทึกใหม่ข้อความใหม่ก็จะถูกบันทึกไปทับข้อความเดิมเสมอ และถ้าข้อความใหม่ที่บันทึกสั้นกว่าข้อความเดิม ข้อความเดิมที่ยังไม่ถูกบันทึกทับก็จะยังคงอยู่

4.1.2) ขั้นตอนการเล่นกลับ (Playback) ใน Push-Button Mode

- 1) กำหนดขา A6-A8 ไปเป็น High เพื่อเลือกโหมด Push-Button ส่วนขา A0-A5 ควร Full Down หรือกำหนดให้เป็น Low
- 2) Set ขา PD ไปเป็น Low เพื่อออกจากสถานะ Reset ISD2548 จะพร้อมเริ่มทำงาน
- 3) Set ขา P/\overline{R} ไปเป็น High เพื่อเลือกใช้งานการเล่นกลับ
- 4) Set ขา \overline{CE} ในลักษณะ Pulse Low คือ Set ขา \overline{CE} จากที่เป็น High อยู่ ให้เป็น Low และ Set กลับมาให้เป็น High ใหม่อีกครั้งการเล่นกลับก็จะเริ่มต้นขึ้น สถานะของขา \overline{EOM} ปกติจะเป็น Low อยู่ก็จะกลายเป็น High เพื่อแสดงสถานะการเล่นกลับ โดยเสียงจะเริ่มเล่นกลับที่ Address 0x000 เมื่อเล่นจบข้อความหนึ่งแล้วการเล่นก็จะถูกหยุดอัตโนมัติ และขา \overline{EOM} ก็จะเปลี่ยนสถานะจาก High ไปเป็น Low

- 5) ในขณะที่ข้อความถูกเล่นอยู่ ถ้ามีการส่ง Pulse Low มาที่ขา \overline{CE} อีกการเล่นก็จะถูกหยุดชั่วคราว (หยุดแบบ PAUSE) และ Address Pointer จะยังไม่ถูก Reset ส่วนสัญญาณที่ขา \overline{EOM} จากที่เป็น High ก็จะกลับมาเป็น Low ช่วงเวลาที่ PAUSE อยู่ นี้ ถ้าเรา Set ให้ขา P/\overline{R} ไปเป็น Low เพื่อการบันทึก ส่วนขา PD คงเดิมไว้ เมื่อเราส่ง Pulse Low ไปที่ขา \overline{CE} การบันทึกก็จะเริ่มขึ้น และหยุดลงเมื่อส่ง Pulse Low ไปให้ขา \overline{CE} อีกครั้ง โดยข้อความที่ถูกบันทึกใหม่นี้จะถูกบันทึกต่อจากตำแหน่ง Address Pointer ของการเล่นข้อความที่หยุดลงล่าสุด จะไม่ไปเริ่มต้นที่ Address Pointer 0x000
- 6) จากขั้นตอนที่ 4 หลังจากมีการเล่นข้อความใดข้อความหนึ่งจบแล้ว การเล่นจะหยุดลงเองอัตโนมัติ เมื่อผู้ใช้ต้องการเล่นข้อความต่อไป ก็ให้ ส่ง Pulse Low ไปที่ขาสัญญาณ \overline{CE} อีก ข้อความที่อยู่ต่อจากข้อความแรกก็จะถูกเล่น นั่นคือ ทุกครั้งที่เราส่ง Pulse Low ไปที่ ขา \overline{CE} หลังจากที่ข้อความที่เล่นก่อนหน้าจบลง ข้อความถัดไปก็จะถูกนำมาเล่นเสมอ จนกว่าจะเกิด Overflow หรือ ขา PD ถูก Set เป็น High Address Pointer ก็จะถูก Reset เป็น 0x000 เมื่อต้องการเล่นกลับใหม่ก็ให้กลับไปเริ่มต้นทำในขั้นตอนที่ 1 ใหม่ การเล่นก็จะกลับมาเริ่มต้นเล่นที่ตำแหน่งข้อความ Address 0x000 อีกครั้ง

ใน Push-Button โหมดนี้สามารถกำหนดการเล่นในการเล่นกลับข้อความ เช่น การ Skip ข้อความ การเล่นซ้ำในข้อความแรกเป็นต้น ซึ่งสามารถทำได้ด้วยการส่งงานที่ขา M0-M5 ตัวอย่างเช่น

-การใช้ขา M3 เพื่อกำหนดให้เล่นซ้ำข้อความที่ตำแหน่ง Address Pointer 0x000(เล่นซ้ำเฉพาะข้อความแรกเท่านั้น)

การกำหนดการเล่นซ้ำ สามารถกำหนดได้ทั้งในขณะที่มีการเล่นกลับอยู่ในข้อความใดๆ หรือ ไม่มีการเล่นกลับอยู่ก็ได้ โดยขา PD จะต้องเป็น Low และขา P/\overline{R} จะต้องยังคงเป็น High อยู่ เมื่อต้องการให้มีการเล่นซ้ำในข้อความแรก ก็ให้ Set ขา M3 จากปกติเป็น Low ไปเป็น High จากนั้นจ่าย Pulse Low ให้กับขา \overline{CE} ข้อความแรกก็จะถูกเล่นกลับ และเมื่อข้อความแรกเล่นจบมันก็จะวนกลับมาเริ่มต้นเล่นซ้ำใหม่

การออกจากการเล่นซ้ำ ทำได้โดย Set ขา M3 จากที่เป็น High อยู่ให้กลับมาเป็น Low เหมือนเดิม และจ่าย Pulse Low ให้กับขา \overline{CE} ก็จะเป็นการออกจากการเล่นซ้ำ

ในการ Set การเล่นซ้ำ หรือ Set ออกจากการเล่นซ้ำ ถ้าเราส่งสัญญาณ Pulse Low ไปที่ขา \overline{CE} ในจังหวะที่มีการเล่นกลับอยู่ จะทำให้การเล่นกลับหยุดลงแบบ PAUSE ไปด้วย ดังนั้นเราจะต้องส่ง Pulse Low ไปที่ขา \overline{CE} อีกครั้งหนึ่งเป็นลูกที่ 2 เพื่อให้ข้อความที่เล่นอยู่ถูกเล่นต่อไปจนจบข้อความ จากนั้นการเล่นกลับก็จะกลับมาเริ่มต้นเล่นใหม่ ถ้าเป็นการ Set M3 ให้เล่นซ้ำ หรือหยุดการเล่นเลย ถ้าเป็นการ Set M3 ให้ออกจากการเล่นซ้ำ

-การใช้ขา M0 เพื่อ Skip (ข้าม) ข้อความที่ไม่ต้องการเล่นให้ข้ามไป

การกำหนดการ Skip สามารถกำหนดได้ทั้งในขณะที่มีการเล่นกลับอยู่ในข้อความใดๆ หรือ ไม่มีการเล่นกลับอยู่ก็ได้ โดยขา PD จะต้องเป็น Low และขา P/\overline{R} จะต้องยังคงเป็น High อยู่ เมื่อต้องการ skip ข้อความที่ไม่ต้องการเล่นกลับให้ข้ามไปก็ให้ Set ขา M0 จากปกติเป็น Low ไปเป็น High จากนั้นจ่าย Pulse Low ให้กับขา \overline{CE} โดยต้องการให้ skip ผ่านไปที่ข้อความ ก็ให้จ่าย Pulse Low ไปที่ขา \overline{CE} เท่านั้นลูก หลังจากนั้นก็ให้ Set ขา M0 จาก High กลับไปเป็น Low แล้วจ่าย Pulse Low ให้กับขา \overline{CE} อีกครั้งหนึ่ง ข้อความที่อยู่ต่อจากข้อความที่ถูก skip ข้ามไปครั้งล่าสุด ก็จะถูกเล่นกลับ

5. การใช้งานบอร์ด ET-MINI ISD2548

จากที่กล่าวมาในหัวข้อที่ 4 ทั้งหมดนั้นจะเป็นการควบคุมการทำงานที่ขา I/O ของตัว ISD2548 โดยตรง แต่สำหรับบอร์ด ET-MINI ISD 2548 นั้นจะมี Shift Register # 74HC595 มาต่อเป็นภาคหน้าของ ISD2548 เพื่อลดจำนวนขา I/O ที่จะ

ใช้ควบคุมการทำงานของ ISD2548 ให้น้อยลง ซึ่ง Shift Register นี้จะใช้การรับส่งข้อมูลแบบ Serial-in/Parallel-Out ซึ่งจะใช้การ Interface ในลักษณะของ SPI ดังนั้นในการควบคุมการทำงานของ ISD2548 ผู้ใช้จะต้องใช้ MCU เป็นตัวกลางในการรับส่งข้อมูลสำหรับควบคุมการทำงานของ ISD2548 ซึ่งข้อมูลที่จะใช้ส่งไปควบคุมการทำงานนั้นก็ต้องสอดคล้องกับกระบวนการทำงานของ ISD2548 ที่กล่าวไว้ในหัวข้อที่ 4 ด้วย

5.1) การส่งข้อมูลแบบ SPI

ก่อนอื่นเราจะมาดูการทำงานในส่วนของการส่งข้อมูลแบบ SPI กันก่อน ซึ่ง Shift Register เบอร์นี้จะเป็น Shift Register ขนาด 8-bit หรือ 1 Byte ส่วน Data ที่จะใช้ส่งไปควบคุม ISD2548 จะมีอยู่ด้วยกัน 12 Bit ดังนั้น Data ที่จะส่งไปควบคุม ISD2548 ในแต่ละครั้งก็ต้องส่ง 12 Bit โดยขาควบคุม ISD2548 จะเรียงตามบิต Data ที่จะส่งออกไปดังนี้

Bit-Data	Bit-11	Bit-10	Bit-9	Bit-8	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
PIN-Control ISD2548	CE	PD	P/R	A8	A7	A6	A5	A4	A3	A2	A1	A0
Mark-Data	X	X	X	X	X	X	X	X	X	X	X	X

X = Mark data 0 หรือ 1

โดยเริ่มต้นผู้ใช้จะต้องทำการ Mark-Data ก่อนว่าบิตไหนเป็นอะไรโดยดูตามกระบวนการทำงานในหัวข้อที่ 4 เมื่อ Mark data ได้แล้วผู้ใช้ก็จะได้ Data ที่จะต้องส่งออกไปควบคุม ISD2548 ในการส่ง Data นี้ ผู้ใช้จะต้องส่ง Bit-11 ออกไปเป็นบิตแรก โดยให้ MCU ส่ง data bit แรก ออกไปยัง Pin – Din ก่อน จากนั้นให้ MCU ส่งสัญญาณ Clock ลูกแรกที่ขอบขาขึ้น (Low to High) ไปยัง Pin-SCK เพื่อ Shift data bit แรกไปเก็บไว้ใน Register ของ 74HC595 จากนั้นก็ให้ MCU ส่งบิตที่ 10 ออกมาที่ Pin-Din แล้วตามด้วย Clock ลูกที่ 2 ให้ทำเช่นนี้จนครบ 12 บิต (Clock ครบ 12 ลูก) Data ที่ส่งมา ก็จะถูกระบายอยู่ใน Shift Register ทั้ง 2 ตัว ซึ่งตรงกับตำแหน่ง Pin ต่างๆของ ISD2548 ตามตารางข้างต้น สุดท้ายให้ MCU ส่งสัญญาณ Clock ขอบขาขึ้น (Low to High) มาที่ขา STR เพื่อทำการปล่อยข้อมูลที่รับเข้ามาเก็บไว้ใน Shift Register ออกมาทางด้าน Output แบบขนานพร้อมกัน ข้อมูลที่ออกมาจะไปควบคุมการทำงานขา I/O ของ ISD2548 ที่ต่ออยู่ ISD2548 ก็จะทำงานตามจังหวะของ Data ที่ส่งออกมาในแต่ละครั้งตามที่ผู้ใช้ต้องการ เป็นการเสร็จสิ้นในการส่ง Data 1 ครั้ง

เมื่อทราบหลักการส่งข้อมูลแบบอนุกรม SPI แล้ว ต่อไปเราจะใช้รูปแบบการส่งข้อมูลนี้ไปควบคุมการทำงานของ ISD2548 ตามกระบวนการทำงานที่กล่าวไว้ในหัวข้อที่ 4 ดังนี้

5.2) การใช้งาน ET-MINI ISD2548 ใน Address Mode

5.2.1) ขั้นตอนการบันทึก (Recording) สำหรับการบันทึกในที่นี้จะขอเริ่มบันทึกที่ Address 0x000 โดยบันทึกผ่านช่องสัญญาณ MIC

- 1) ต่อ I/O (OUTPUT-PIN) ของ MCU เข้ากับขา DIN,SCK,STR ของบอร์ด ISD2548
- 2) ต่อ I/O (INPUT-PIN) ของ MCU เข้ากับขา EOM ของบอร์ด ISD2548 (จะใช้ เวลาเล่นกลับ)
- 3) ต่อ ไมโครโฟน สำหรับอัดเสียงเข้าที่ขั้วต่อ MIC และ Set Jumper JP2 มาทางด้าน MIC ในกรณีที่ต้องการอัดเสียงจาก Line Out ของแหล่งกำเนิดเสียงอื่นๆ ให้นำสัญญาณ Line-Out ต่อเข้าที่ขั้ว ANA_IN ขั้วใดขั้วหนึ่งของบอร์ด ISD2548 และ Set JP2 มาทางด้าน ANA_IN แทน
- 4) Set Jumper JP1 มาทางด้าน GND เพื่อ Enable Shift Register 74HC595 เอาไว้ หรือถ้าต้องการควบคุมการ Enable เอง

- ด้วย MCU ก็ให้ Set JP1 มาทางด้าน \overline{ENA} แล้วต่อ PIN \overline{ENA} ไปยังขา Output ของ MCU สำหรับควบคุมการทำงาน
- 5) จ่ายไฟ 5 V เลี้ยงบอร์ด ET-MINI ISD2548
 - 6) ให้ MCU ส่ง Data ชุดแรกค่า 0xE00 ($\overline{CE}, PD, P/\overline{R} = 1$ และ Set Address เริ่มต้น A0-A8 = 0) ไปยังบอร์ด ISD2548 สำหรับ Initial ISD2548
 - 7) ให้ MCU ส่ง Data ชุดที่ 2 ค่า 0x800 ($PD, P/\overline{R} = 0$) เพื่อให้ ISD2548 ออกจากสถานะ Reset และทำงานในโหมดบันทึกเสียง ส่วนบิตอื่นที่เหลือให้คงเดิมไว้
 - 8) ให้ MCU ส่ง Data ชุดที่ 3 ค่า 0x000 ($\overline{CE}, PD, P/\overline{R} = 0$) เพื่อ Set ขา \overline{CE} ให้เป็น Low ซึ่งหลังจากส่ง Data ชุดนี้เรียบร้อยแล้ว ผู้ใช้สามารถเริ่มอัดเสียงผ่านไมโครโฟนได้
 - 9) เมื่อจะหยุดบันทึก ก็ให้ MCU ส่ง Data ชุดที่ 4 ค่า 0xE00 เพื่อทำการ Set ขา $\overline{CE}, PD, P/\overline{R}$ ให้กลับมาเป็น 1 เพื่อกำหนดให้ ISD2548 กลับมาอยู่ในสถานะเริ่มต้นการ Initial ใหม่อีกครั้ง
 - 10) เมื่อต้องการจะบันทึกใหม่ใน Address อื่น ก็ให้กลับไป เริ่มในขั้นตอนที่ 6 ใหม่ โดยในส่วนของ Bit-Data ในตำแหน่ง A0-A8 ให้ผู้ใช้แทนด้วย Address ที่ต้องการจะใช้เริ่มบันทึก(0x000-13F) ส่วนในตำแหน่งบิต $\overline{CE}, PD, P/\overline{R}$ ให้คงเดิมตาม Step ข้างต้น เช่น ถ้าต้องการบันทึกที่ Address 0x020 Data ชุดแรกที่ต้องส่งก็ควรจะเป็น 0xE20 Data ชุดที่ 2 ก็ควรจะเป็น 0x820 ชุดที่ 3 ก็จะเป็น 0x020 เป็นต้น
- เพื่อให้ง่ายในการอัดเสียงผู้ใช้อาจจะใช้ SW. แบบกดคิดปด้อยดับต่อเข้ากับ MCU แล้วให้ MCU ตรวจสอบสถานะการกด SW ถ้ามีการกด ก็ให้ MCU ส่ง Data ใน Step ที่ 6-8 ออกไป ในจังหวะที่กด SW. ลงไปผู้ใช้อก็สามารถอัดเสียงได้ทันที และเมื่อปล่อย SW. ก็ให้ MCU ส่ง Data ใน Step ที่ 9 ออกไปเพื่อหยุดการบันทึกเป็นต้น

5.2.2) ขั้นตอนการเล่นกลับ (Playback) สำหรับการเล่นกลับในที่นี้จะขอเริ่มต้นที่ Address 0x000

- 1) ต่อ I/O (OUTPUT-PIN) ของ MCU เข้ากับขา DIN, SCK, STR ของบอร์ด ISD2548
- 2) ต่อ I/O (INPUT-PIN) ของ MCU เข้ากับขา \overline{EOM} ของบอร์ด ISD2548 สำหรับบิต \overline{EOM} นี้ไม่ได้ต่อผ่านตัว Shift Register ดังนั้นสามารถใช้ MCU อ่าน ค่าตรงๆ ได้เลย
- 3) ต่อหูฟัง หรือ ตัวขยายเสียงเข้าที่ Connector SP
- 4) Set Jumper JP1 มาทางด้าน GND เพื่อ Enable Shift Register 74HC595 รอไว้ หรือถ้าต้องการควบคุมการ Enable เอง ด้วย MCU ก็ให้ Set JP1 มาทางด้าน \overline{ENA} แล้วต่อ PIN \overline{ENA} ไปยังขา Output ของ MCU สำหรับควบคุมการทำงานเอง
- 5) จ่ายไฟ 5 V เลี้ยงบอร์ด ET-MINI ISD2548
- 6) ให้ MCU ส่ง Data ชุดแรกค่า 0xE00 ($\overline{CE}, P/\overline{R}, PD=1$ และ Set Address เริ่มต้น A0-A8 = 0) สำหรับ Initial ISD2548 ให้ทำงานใน Play Mode ส่วน Address เริ่มต้นการเล่นถ้าไม่ใช่ 0x000 เป็นตำแหน่งอื่นๆ ก็ให้แทนค่า Address นั้นในตำแหน่งบิต A0-A8 ใหม่
- 7) ให้ MCU ส่ง Data ชุดที่ 2 ค่า 0x200 ($\overline{CE}, PD=0$) ซึ่งก็คือการส่ง Clock Pulse Low ไปยังขา \overline{CE} ของ ISD2548 และออกจากสถานะ Reset ส่วนบิตอื่นที่เหลือจะยังคงเดิมไม่เปลี่ยนแปลง
- 8) ให้ MCU ส่ง Data ชุดที่ 3 ค่า 0xA00 ($\overline{CE} = 1$) ซึ่งก็คือการส่ง Clock Pulse High ไปยังขา \overline{CE} ของ ISD2548 ส่วนบิตอื่นที่เหลือจะยังคงเดิมไม่เปลี่ยนแปลง หลังจากส่ง Data ชุดนี้ออกไป การเล่นกลับก็จะเริ่มขึ้น โดย Start ที่ Address ตามที่ผู้ใช้ได้กำหนดไว้ในบิต A0-A8

- 9) ให้ MCU วน loop อ่านค่าบิต \overline{EOM} จาก ISD2548 โดยถ้าอ่านค่าได้เป็น '1' แสดงว่ามีการเล่นอยู่ยังไม่จบข้อความ ถ้าอ่านค่าได้เป็น '0' แสดงว่าการเล่นข้อความนั้นจบลงแล้ว และเมื่อตรวจสอบว่า EOM เป็น '0' แล้ว ก่อนที่จะสั่งให้เล่นข้อความต่อไป ก็จะต้องตรวจสอบสัญญาณ EOM ให้กลับเป็น '1' ก่อน เพื่อให้ ISD2548 พร้อมทำงาน จากนั้นเมื่อต้องการให้เล่นข้อความในตำแหน่ง Address อื่นอีก ก็ให้กลับไปเริ่มทำใน Step ที่ 7 ใหม่
- 10) ในกรณีที่ต้องการหยุดการเล่นกลับก่อนที่จะมีสัญญาณ \overline{EOM} ส่งออกมา ก็ให้ MCU ส่ง Data 0xE00 เพื่อหยุดการเล่นทันทีเสมือนการ Initial ISD2548 ใหม่

5.3) การใช้งาน ET-MINI ISD2548 ใน Push-Button Mode

5.3.1) ขั้นตอนการบันทึก (Recording)

- 1) ต่อ I/O (OUTPUT-PIN) ของ MCU เข้ากับขา DIN, SCK, STR ของบอร์ด ISD2548
- 2) ต่อ I/O (INPUT-PIN) ของ MCU เข้ากับขา \overline{EOM} ของบอร์ด ISD2548 (Mode นี้ \overline{EOM} จะใช้บอกสถานะการทำงาน)
- 3) ต่อ ไมโครโฟน สำหรับอัดเสียงเข้าที่ขั้วต่อ MIC และ Set Jumper JP2 มาทางด้าน MIC ในกรณีที่ต้องการอัดเสียงจาก Line Out ของแหล่งกำเนิดเสียงอื่นๆ ให้นำสัญญาณ Line-Out ต่อเข้าที่ขั้ว ANA_IN ขั้วใดขั้วหนึ่งของบอร์ด ISD2548 และ Set JP2 มาทางด้าน ANA_IN แทน
- 4) Set Jumper JP1 มาทางด้าน GND เพื่อ Enable Shift Register 74HC595 รอไว้ หรือถ้าต้องการควบคุมการ Enable เองด้วย MCU ก็ให้ Set JP1 มาทางด้าน \overline{ENA} แล้วต่อ PIN \overline{ENA} ไปยังขา Output ของ MCU สำหรับควบคุมการทำงานเอง
- 5) จ่ายไฟ 5 V เลี้ยงบอร์ด ET-MINI ISD2548
- 6) ให้ MCU ส่ง Data ชุดแรกค่า 0xFC0 สำหรับ Initial ISD2548 โดยอ้างอิง Data ตามตารางบิตที่กล่าวไว้ด้านบน ซึ่งจะได้บิต A5-A0 ปกติให้ '0' ไว้ก่อน ('1' จะเป็นการเลือกรูปแบบการทำงาน), บิต A8-A6 เป็น '1' สำหรับเลือกให้ ISD2548 ทำงานใน Push-Button Mode และบิต \overline{CE} , PD, $\overline{P/R}$ ให้เป็น '1' สำหรับ Initial ISD2548 ให้อยู่ในสถานะ Reset
- 7) ให้ MCU ส่ง Data ชุดที่ 2 ค่า 0xBC0 (PD = 0) เพื่อให้ ISD2548 ออกจากสถานะ Reset ส่วนบิตอื่นให้คงเดิมไว้
- 8) ให้ MCU ส่ง Data ชุดที่ 3 ค่า 0x1C0 (\overline{CE} , PD, $\overline{P/R}$ = 0) ซึ่งก็คือการส่ง Clock Pulse Low ไปยังขา \overline{CE} ของ ISD2548 รวมทั้ง Set ให้ ISD2548 ทำงานในโหมดบันทึกเสียง ส่วนบิตอื่นให้คงเดิมไว้
- 9) ให้ MCU ส่ง Data ชุดที่ 4 ค่า 0x9C0 (\overline{CE} =1) ซึ่งก็คือการส่ง Clock Pulse High ไปยังขา \overline{CE} ของ ISD2548 ส่วนบิตอื่นให้คงเดิมไว้เช่นเดิม หลังจากส่ง Data ชุดนี้ออกไป การบันทึกก็จะเริ่มต้นขึ้น สถานะของขา \overline{EOM} ปกติที่เป็น Low อยู่ก็จะกลายเป็น High แสดงสถานะการอัด โดยเสียงจะ เริ่มถูกบันทึกที่ Address 0x000
- 10) เมื่อต้องการจะหยุดการบันทึก ก็ให้ MCU ส่ง Data ใน Step ที่ 8 และ 9 ออกไปตามลำดับอีกครั้ง(ก็คือการส่ง Clock Pulse ไปอีกหนึ่งลูกเพื่อหยุดการบันทึก) การบันทึกก็จะหยุดลง สัญญาณที่ขา \overline{EOM} จากที่เป็น High อยู่ก็จะกลับมาเป็น Low เพื่อแสดงสถานะหยุดการบันทึกเรียบร้อยแล้ว (ซึ่งผู้ใช้จะต้องตรวจสอบสัญญาณนี้เสมอก่อนที่จะเริ่มบันทึกใหม่) ส่วน Address pointer ของหน่วยความจำ จะยังไม่ถูก Clear หลังจากการบันทึกหยุดลง (ถ้ายังไม่มีการ Reset ISD2548)
- 11) จาก Step 10 ถ้าต้องการจะบันทึกอีก ก็ให้ MCU ทำการส่ง Clock Pulse ตาม Data ใน Step 8 และ 9 อีก การบันทึกก็จะเริ่มขึ้นใหม่โดยข้อความที่บันทึกจะถูกบันทึกต่อจากข้อความแรก และถูกมองเป็นข้อความลำดับที่ 2 ไม่ใช่เป็นข้อความเดียวกับข้อความแรก และเมื่อเราส่งชุด Data ใน Step 8 และ 9 กลับกันไปเช่นนี้ ก็จะเป็นการ บันทึกและหยุด บันทึกกลับกันไป โดยการบันทึกก็จะเป็นการบันทึกข้อความต่อกันไปเรื่อยๆจนจะเต็มหน่วยความจำของ ISD2548 ซึ่งข้อความที่บันทึกต่อกันในแต่ละครั้งนั้นจะถือเป็นข้อความชุด ใหม่เสมอ ไม่ใช่ข้อความต่อเนื่องชุดเดียวกัน

12) เมื่อต้องการให้เริ่มต้นบันทึกที่ตำแหน่ง Address 0x000 ใหม่ ให้ผู้ใช้กลับไปเริ่มทำใน Step 6 ใหม่ เพื่อทำการ Reset เริ่ม Initial ISD2548 ใหม่

เพื่อให้ง่ายในการอัดเสียงในการส่ง Data Step 8-9 ผู้ใช้อาจจะใช้ SW. แบบกดติดปล่อยดับต่อเข้ากับ MCU แล้วให้ MCU ตรวจสอบสถานะ การกด SW ถ้ามีการกด ก็ให้ MCU ส่ง Data ใน Step ที่ 8-9 ออกไป ในจังหวะที่กด SW.ลงไปผู้ใช้ก็สามารถอัดเสียงได้ทันที และเมื่อปล่อย SW. ก็ให้ MCU ส่ง Data ใน Step ที่ 8-9 ออกไปเพื่อหยุดการบันทึก และหลังจากปล่อย SW. ควรตรวจสอบสัญญาณ \overline{EOM} เสมอว่าเป็น 0 หรือยัง เพื่อตรวจสอบว่า ISD2548 หยุดบันทึกเรียบร้อยแล้ว

5.3.2) ขั้นตอนการเล่นกลับ (Playback)

- 1) ต่อ I/O (OUTPUT-PIN) ของ MCU เข้ากับขา DIN, SCK, STR ของบอร์ด ISD2548
- 2) ต่อ I/O (INPUT-PIN) ของ MCU เข้ากับขา \overline{EOM} ของบอร์ด ISD2548 (ใน Mode นี้ \overline{EOM} จะใช้บอกสถานะการทำงาน)
- 3) ต่อหูฟัง หรือ เครื่องขยายเสียงเข้าที่ Connector SP
- 4) Set Jumper JP1 มาทางด้าน GND เพื่อ Enable Shift Register 74HC595 รอไว้ หรือถ้าต้องการควบคุมการ Enable เอง ด้วย MCU ก็ให้ Set JP1 มาทางด้าน \overline{ENA} แล้วต่อ PIN \overline{ENA} ไปยังขา Output ของ MCU สำหรับควบคุมการทำงาน
- 5) จ่ายไฟ 5 V เลี้ยงบอร์ด ET-MINI ISD2548
- 6) ให้ MCU ส่ง Data ชุดแรกค่า 0xFC0 สำหรับ Initial ISD2548 โดยอ้างอิง Data ตามตารางบิตที่กล่าวไว้ด้านบน โดยให้ บิต A5-A0 เป็น '0' ไว้ก่อน ('1' จะเป็นการเลือกรูปแบบการทำงาน), ส่วนบิต A8-A6 เป็น '1' เพื่อเลือกให้ ISD2548 ทำงานใน Push-Button Mode และบิต \overline{CE} , PD, P/R ให้เป็น '1' สำหรับ Initial ISD2548 ให้อยู่ในสถานะ Reset
- 7) ให้ MCU ส่ง Data ชุดที่ 2 ค่า 0xBC0 (PD = 0) เพื่อให้ ISD2548 ออกจากสถานะ Reset ส่วนบิตอื่นให้คงเดิมไว้
- 8) ให้ MCU ส่ง Data ชุดที่ 3 ค่า 0x3C0 (\overline{CE} =0) ซึ่งก็คือการส่ง Clock Pulse Low ไปยังขา \overline{CE} ของ ISD2548 ส่วนบิตอื่นให้คงเดิมไว้
- 9) ให้ MCU ส่ง Data ชุดที่ 4 ค่า 0xBC0 (\overline{CE} =1) ซึ่งก็คือการส่ง Clock Pulse High ไปยังขา \overline{CE} ของ ISD2548 ส่วนบิตอื่นให้คงเดิมไว้ เช่นเดิม หลังจากส่ง Data ชุดนี้ออกไป การเล่นก็จะเริ่มขึ้น สถานะของขา \overline{EOM} ปกติที่เป็น Low อยู่ก็จะกลายเป็น High เพื่อแสดงสถานะการเล่นกลับ โดยเสียงจะเริ่มเล่นกลับที่ Address 0x000 เมื่อเล่นจบข้อความหนึ่งแล้วการเล่นก็จะถูกหยุดอัตโนมัติ และ ขาสัญญาณ \overline{EOM} ก็จะเปลี่ยนสถานะจาก High ไปเป็น Low ตามเดิมเพื่อบอกให้ทราบสถานะหยุดเล่น ซึ่งผู้ใช้จะต้องตรวจสอบสัญญาณนี้เสมอ ก่อนที่จะสั่งเล่นข้อความต่อไป
- 10) จาก Step 9 หลังจากมีการเล่นข้อความใดข้อความหนึ่งจบแล้ว และตรวจสอบสัญญาณ \overline{EOM} เป็น Low แล้ว การเล่นจะหยุดลงเองอัตโนมัติ เมื่อผู้ใช้ต้องการเล่นข้อความต่อไป ก็ให้ MCU ส่ง Data ตาม Step 8 และ 9 ใหม่ ซึ่งก็คือการส่ง Clock Pulse Low to High ไปที่ขาสัญญาณ \overline{CE} ของ ISD2548 ข้อความที่อยู่ต่อจากข้อความแรกก็จะถูกเล่น นั่นคือ ทุกครั้งที่เราส่ง Pulse Low to High ไปที่ ขา \overline{CE} หลังจากที่ข้อความที่เล่นก่อนหน้าจบลง ข้อความถัดไปก็จะถูกนำมาเล่นเสมอ จนกว่าจะเกิด Overflow หรือ ขา PD ถูก Set เป็น High (Reset ISD2548) ก็จะทำให้ Address Pointer ถูก Reset ให้กลับไปเริ่มต้นที่ Address 0x000 อีกครั้ง ดังนั้นเมื่อต้องการเล่นกลับโดยเริ่มต้นข้อความที่ Address 0x000 ใหม่ก็ให้กลับไปเริ่มต้นทำใน Step 6 ใหม่
- 11) ในขณะที่ข้อความยังเล่นไม่จบผู้ใช้สามารถ ให้ MCU ส่ง Data ใน Step 8 และ 9 ซึ่งก็คือการส่ง Clock Pulse Low to High ไปที่ขาสัญญาณ \overline{CE} ของ ISD2548 จะทำให้การเล่นหยุดลงชั่วคราว(หยุดแบบ PAUSE) และ Address Pointer จะไม่ถูก

Reset ส่วนสัญญาณที่ขา EOM จากที่เป็น High ก็จะกลับมาเป็น Low เมื่ออยู่ในสภาวะ PAUSE เมื่อจะออกจากสภาวะ PAUSE ก็ให้ทำการส่ง Data ตาม Step 8 และ 9 อีกครั้ง การเล่นก็จะกลับมาเล่นต่อจากตำแหน่งที่หยุดลงอีกครั้ง

6. ตัวอย่างโปรแกรมใน CD

สำหรับตัวอย่างที่ให้มาใน CD นั้นจะเป็นภาษา C ทำไว้กับ MCU AVR-MEGA128 , PIC18F8722 และ 89C51RE2 โดยสามารถดู Compiler ที่ใช้งาน และวงจรการต่อที่รองรับในแต่ละตัวอย่างได้จาก Comment ที่อยู่บนหัวของโปรแกรม ซึ่งตัวอย่างที่ให้มาในแต่ละ MCU นั้นจะเป็นตัวอย่างเดียวกันหมด ในการทดสอบตัวอย่างแต่ละโปรแกรมนั้นควรมีอุปกรณ์

ดังต่อไปนี้ *1. บอร์ด MCU รุ่น ET-BASE AVR ATmega128 r3 หรือ CP-JR51RE2 V1.0 หรือ ET-BASE PIC8722(ICD2)

*2. บอร์ด ET-MINI ISD2548

3. LCD 16x2 + ET-CONV14LCD (ใช้กับตัวอย่างที่1,2,3)

4. TEST INPUT (SW.กดติดปล่อยดับ ใช้กับตัวอย่างที่1,3,4)

5. ET-MINI DS3232 (ใช้กับตัวอย่างที่2)

*ใช้กับทุกตัวอย่าง

Ex. ADDRESS MODE

Ex1_Play_Record : สำหรับตัวอย่างนี้จะเป็นการบันทึกเสียง และเล่นกลับเสียงที่บันทึก ก่อนอื่นจะต้องต่อ LCD, SW และ บอร์ด ET-MINI ISD2548 (Set JP1=GND,JP2= เลือกช่องสัญญาณที่ต่อใช้งาน MIC หรือ ANA IN) เข้ากับบอร์ด MCU ที่นำมาทดลองเสียก่อน เมื่อต่อวงจรส่วนต่างๆตามใน Comment ที่หัวโปรแกรมเรียบร้อยแล้ว ก็ให้โหลดโปรแกรม ทดสอบได้ โดยตัวอย่างนี้สามารถบันทึกเสียงในแต่ละข้อความได้ยาวไม่เกิน 1.2 วินาที สามารถเลือกช่องในการบันทึกเสียง และเล่นกลับได้ เริ่มต้นทดสอบให้ผู้ใช้กด SW+ หรือ SW- เพื่อเลือกช่องที่จะบันทึกเสียงสังเกตตัวเลขที่หน้าจอ LCD จะเปลี่ยนไปตาม SW ที่กด (Default = Message 01) จากนั้นให้กด SW. Record การบันทึกก็จะเริ่มขึ้นทันที ในขณะที่บันทึก หน้าจอ LCD ก็จะแสดงข้อความ REC [(0)] ตัวเลข 0 ในวงเล็บจะกระพริบ หลังจากบันทึกครบ 1.2 วินาที โปรแกรมก็จะออกจากการบันทึกอัตโนมัติ ที่หน้าจอ LCD ก็จะกลับมาแสดงเหมือนกับตอนที่บันทึก จากนั้นให้ผู้ใช้กด SW. Play Back เพื่อฟังเสียงที่บันทึกไว้ เมื่อจะบันทึกอีกก็ให้กด SW+ , SW- เลือกช่องบันทึกใหม่ถ้าไม่ต้องการบันทึกทับช่องเดิม ส่วน SW. Play All จะเป็นการเล่นกลับข้อความเสียงทั้งหมดที่บันทึกไว้ โดยเริ่มเล่นข้อความจากช่องที่ 1 ก่อน

จำนวนช่องข้อความเก็บเสียงนี้จะมีช่องก็ขึ้นอยู่กับว่าเราแบ่งบันทึกข้อความช่องละกี่วินาที ซึ่งเมื่อรวมเวลาเก็บเสียงในแต่ละช่องแล้วจะต้องไม่เกิน 48 วินาที ตามหน่วยความจำของ ISD2548 ที่มีอยู่ ในการหา Address เริ่มต้นของแต่ละช่องที่ใช้บันทึกเสียง เพื่อไม่ให้เกิดการบันทึกทับ Address กันให้กลับไปดูในหัวข้อที่ 4 ซึ่งสรุปเป็นสมการไว้ให้แล้ว

Ex2_Tell_Time: สำหรับตัวอย่างนี้จะเป็นการบอกเวลาทุกๆ 30 นาที โดยก่อนที่จะทดสอบตัวอย่างนี้ ก่อนอื่นผู้ใช้จะต้องบันทึกเสียงลงใน ISD2548 เสียก่อน ซึ่งทำได้โดยการใช้โปรแกรม และต่อวงจรตามตัวอย่างที่ 1 แล้วทำการบันทึกข้อความลงในช่อง Message 02-19 โดยในแต่ละช่องจะบันทึกข้อความอะไรก็ได้ Comment ที่หัวโปรแกรมของตัวอย่างที่ 2 โดยไฟล์ข้อความจะให้มาใน CD อยู่ใน Folder “ตัวอย่างเสียง_Rec/ข้อความบอกเวลา” ในการบันทึกสามารถต่อสัญญาณในช่องหูฟังของ PC มาเข้าที่ขั้วต่อ ANA IN ของบอร์ด ET-MINI ISD2548 ได้เลย และให้ Set JP2 มาทาง ANA IN ด้วย หลังจากบันทึกเสียงเรียบร้อยแล้ว ก็ให้ต่อวงจรของตัวอย่างที่ 2 ตาม Comment ที่หัวโปรแกรม (เวลาต่อวงจรให้เอาแหล่งจ่าย

ไฟเลี้ยงทั้งหมดออกก่อนมิฉะนั้นอาจทำให้เสียงที่อัดไว้หายไป) แล้วทำการโหลดโปรแกรมตัวอย่างที่ 2 เพื่อทดสอบ ในตัวอย่างจะตั้งเวลาไว้ที่ 11:29:55 เมื่อโปรแกรมเริ่มทำงาน และเวลาบนจอ LCD เดินมาอยู่ที่ 11:30:00 ก็จะมีเสียงบอกเวลาว่า “ขณะนี้เวลา สิบเอ็ดนาฬิกา สามสิบนาที ศูนย์วินาที” เป็นต้น โดยจะบอกเวลาทุกๆ 30 นาที ลักษณะการทำงานของโปรแกรมจะอ่านค่าเวลาจาก RTC แล้วนำค่าที่อ่านได้แต่ละครั้งในส่วนที่เป็น วินาที มาตรวจสอบว่า เท่ากับ 00 หรือไม่ถ้าเท่า ก็จะไปตรวจสอบค่าของนาฬิกาว่ามีค่าเท่ากับ 30 หรือ 00 หรือไม่ ถ้าค่าเวลาวินาทีและนาฬิกา เป็นไปตามเงื่อนไขที่ตั้งไว้ก็จะไปสั่งให้เล่นข้อความเสียงในแต่ละช่องที่สอดคล้องกับเวลาขณะนั้น โดยใช้การนำข้อความแต่ละช่องมาเล่นต่อกัน

Ex3_Ticket สำหรับตัวอย่างนี้จะเป็นการบอกหมายเลขบัตรคิวตามช่องบริการที่มีการกด SW. โดยจะมีช่องบริการ 6 ช่องสามารถบอกหมายเลขบัตรคิวได้ตั้งแต่ 01-99 ซึ่งก่อนที่จะทำการทดสอบตัวอย่างนี้อันดับแรกผู้ใช้จะต้องบันทึกเสียงลงใน ISD2548 เสียก่อน ซึ่งทำได้โดยการใช้โปรแกรม และต้องจรรยาตามตัวอย่างที่ 1 แล้วทำการบันทึกข้อความลงในช่อง Message 02-24 โดยในแต่ละช่องจะบันทึกข้อความอะไรให้ดู Comment ที่หัวโปรแกรมของตัวอย่างที่ 3 โดยไฟล์ข้อความจะเข้ามาใน CD อยู่ใน Folder “ตัวอย่างเสียง_Rec/ข้อความบัตรคิว” ในการบันทึกสามารถต่อสัญญาณในช่องหูฟังของ PC มาเข้าที่ขั้วต่อ ANA IN ของบอร์ด ET-MINI ISD2548 ได้เลย และให้ Set JP2 มาทาง ANA IN ด้วย หลังจากบันทึกเสียงเรียบร้อยแล้ว ก็ให้ต่อวงจรของตัวอย่างที่ 3 ตาม Comment ที่หัวโปรแกรม (เวลาต่อวงจรให้เอาแหล่งจ่ายไฟเลี้ยงทั้งหมดออกก่อนมิฉะนั้นอาจทำให้เสียงที่อัดไว้หายไป) แล้วทำการโหลดโปรแกรมตัวอย่างที่ 3 เพื่อทดสอบ เมื่อโปรแกรมเริ่มทำงาน บนจอ LCD จะแสดงข้อความ “COUNTER = ” และ “NUMBER = ” เมื่อผู้ใช้กด SW. ใดๆ ที่หน้าจอก็จะแสดงหมายเลข Counter ตามตำแหน่ง SW. ที่กด และแสดงลำดับ Number ของบัตรคิว แล้วก็มีข้อความเสียงพูดว่า “เชิญหมายเลข.....ที่ช่องบริการ.....ค่ะ ” ทุกครั้งที่มีการกด SW. ก็จะมีข้อความเสียงพูดเช่นนี้เสมอ ในส่วนหมายเลข Counter ก็จะพูดไปตามหมายเลขช่องที่ถูกกด คือ 1-6 ส่วนหมายเลข Number จะเป็นหมายเลขต่อเนื่องโดยจะพูดต่อจากลำดับหมายเลขก่อนหน้า คือ 01-99

Ex. PUSH BUTTON MODE

Ex4_Play_Record_skip : สำหรับตัวอย่างนี้จะเป็นการบันทึกเสียง และเล่นกลับเสียงที่บันทึก ก่อนอื่นจะต้องต่อ SW และ บอร์ด ET-MINI ISD2548 (Set JP1=GND,JP2= เลือกลงช่องสัญญาณที่ต่อใช้งาน MIC หรือ ANA IN) เข้ากับบอร์ด MCU ที่นำมาทดลองเสียก่อนเมื่อต่อวงจรส่วนต่างๆตามใน Comment ที่หัวโปรแกรมเรียบร้อยแล้ว ก็ให้โหลดโปรแกรมทดสอบได้ โดยการทำงานของโปรแกรมจะเป็นดังนี้

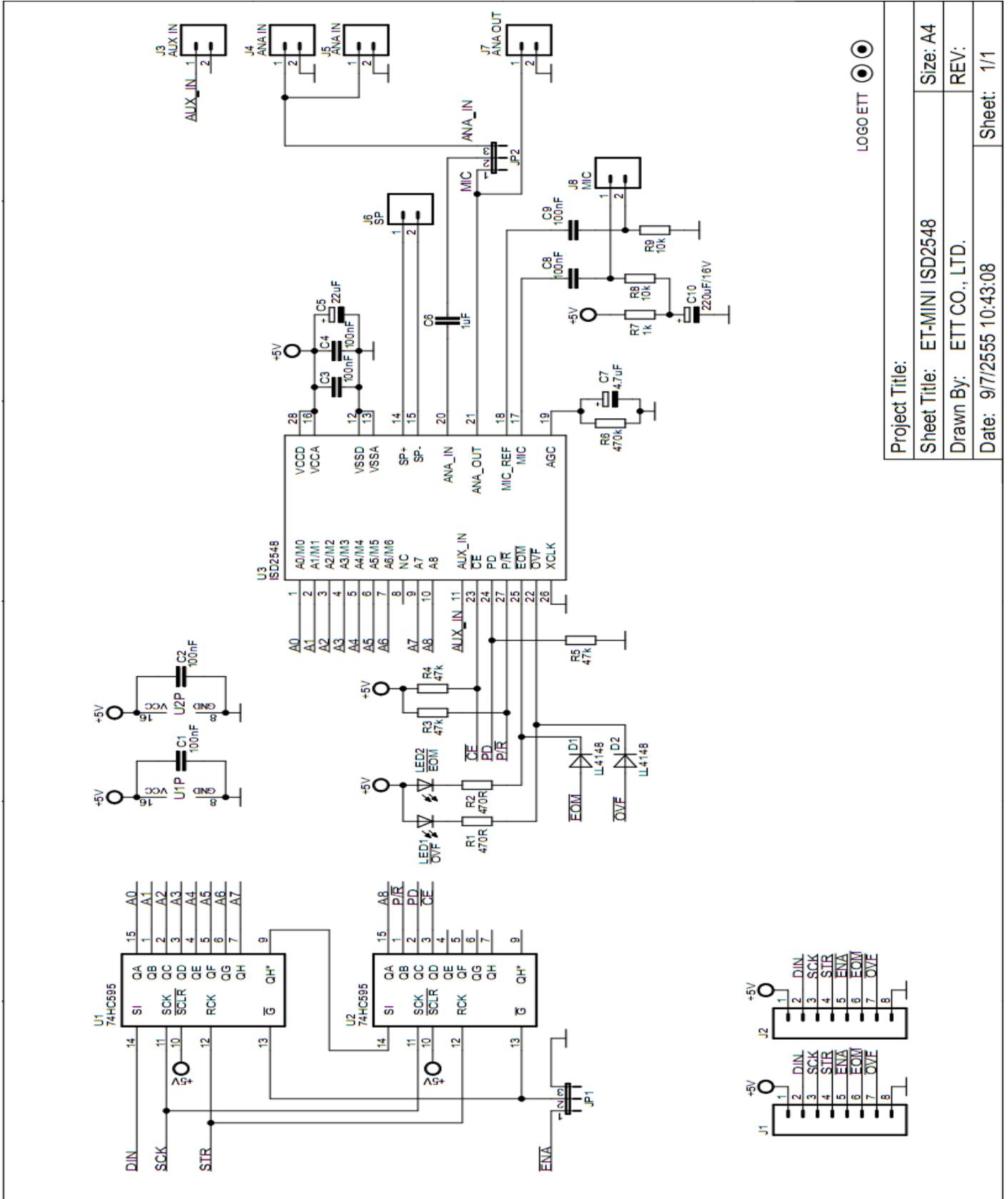
- การบันทึกเสียง ให้ผู้ใช้ทำการกด SW. Record ค้างไว้การบันทึกก็จะเริ่มขึ้น โดยจะเริ่มบันทึกที่ Address 0x000 เมื่อเราปล่อย SW. Record การบันทึกก็จะหยุดลงเป็นอันจบข้อความแรก ถ้าเรายังไม่กด SW. Reset และกด SW. Record ค้างไว้ใหม่การบันทึกก็จะเริ่มขึ้นอีก โดยข้อความที่ 2 ที่ถูกบันทึกก็จะถูกบันทึกต่อจากข้อความแรก ตรงใดที่เรายังไม่กด SW. Reset และหน่วยความจำยังไม่เต็ม ข้อความที่เราบันทึกแต่ละครั้ง จะเรียงต่อกันไปเรื่อยๆ

-การเล่นกลับ ให้ผู้ใช้ทำการกด SW. Play แล้วปล่อยการเล่นกลับก็จะเริ่มขึ้น โดยจะเริ่มเล่นจากข้อความแรก เมื่อเล่นจบการเล่นก็จะหยุดลงอัตโนมัติ เมื่อทำการกด SW. Play อีก ข้อความที่อยู่ต่อจากข้อความแรกจะถูกเล่น ตรงใดที่ยังไม่มีการกด SW. Reset หรือ ข้อความที่เล่นไม่ใช่ข้อความสุดท้ายในหน่วยความจำ ทุกครั้งที่กด SW. Play ข้อความก็จะถูกเล่นตามลำดับที่อยู่ต่อกันไปเรื่อยๆตามการกด SW. Play

-การ Skip เมื่อผู้ใช้ทำการกด SW. Skip แล้วปล่อยในแต่ละครั้ง จะเป็นการข้ามข้อความที่จะเล่นไปข้างหน้าครั้งละหนึ่งข้อความ เช่น ถ้าตำแหน่งข้อความที่จะถูกเล่นเริ่มที่ข้อความแรก เมื่อกด SW. Skip 1 ครั้ง แล้วตามด้วยการกด SW. Play

ก็จะได้อินข้อความที่ถูกลบเป็นข้อความที่ 2 ไม่ใช่ข้อความแรก เป็นต้น

-การ Reset ทุกครั้งที่ผู้ใช้กด SW. Reset จะทำให้ตำแหน่ง Address Pointer ภายใน ISD2548 ถูก Reset ไปเป็น 0 ดังนั้นเวลากด SW. Play หรือ SW. Record หลังจากการกด SW. Reset จะทำให้การเล่นกลับมาเริ่มต้นเล่นที่ข้อความแรกเสมอไม่ว่าเราจะเล่นค้างอยู่ที่ข้อความใดก็ตาม ส่วนการบันทึกก็จะกลับมาบันทึกที่ตำแหน่งข้อความแรกเสมอเช่นกัน



Project Title:	
Sheet Title: ET-MINI ISD2548	Size: A4
Drawn By: ETT CO., LTD.	REV:
Date: 9/7/2555 10:43:08	Sheet: 1/1

รูป วงจร ET-MINI ISD2548