ET-DISPLAY 8x32 RED

ET-DISPLAY 8X32 RED เป็นบอร์ด DISPLAY แบบ Dot Matrix LED ขนาด 8x32 (256 DOT) ใช้ IC MAX7219 จำนวน 4 ตัว เป็นตัว Drive และ Scan รวมทั้งสามารถส่งคำสั่งควบคุมความสว่างของ Display ได้ด้วย โดย IC หนึ่งตัวจะใช้ควบคุม Dot Matrix LED จำนวน 8x8 DOT การนำไปใช้งานผู้ใช้สามารถนำบอร์ด MCU ต่างๆจากภายนอกเขียนควบคุมการทำงานได้ ซึ่งตัวบอร์ดจะใช้การเชื่อมต่อแบบ Serial SPI(CS,CLK, DIN) ทำให้ประหยัด I/O ในการควบคุมการทำงานของบอร์ดและง่ายต่อการเขียนโปรแกรม เนื่องจากตัว MAX7219 จะ SCAN LED ให้ในตัว ทำให้ LED ติดสว่างสม่ำเสมอ ดังนั้นผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมให้ MCU แสกน Display อยู่ตลอดเวลา ทำให้ MCU มีเวลาไปทำอย่างอื่นตามที่ผู้ใช้เขียนโปรแกรมได้มากขึ้น

สำหรับ Display รุ่นนี้ จะมี Option เสริมพิเศษซึ่งสามารถซื้อเพิ่มเติมได้สำหรับผู้ใช้ที่ต้องการความสะดวกในการนำ Display ไปพัฒนาเขียนโปรแกรมและใช้งานจริง ได้แก่

1) BOX DISPLAY 8x32 BIG RED เป็นกล่องสำหรับใส่ Display+ Filter สีแดง
2) ET-CPU-DISPLAY 8x32 เป็นบอร์ดสำหรับใช้เขียนโปรแกรม Control DISPLAY โดยใช้ MCU Mega 328 ซึ่งสามารถต่อยึดเข้ากับตัว Display ด้านหลังได้เลย รวมทั้งสามารถประกอบเข้ากับตัวกล่อง DISPLAY ที่จำหน่ายได้เลย การพัฒนาโปรแกรมสามารถพัฒนาโดยใช้ Arduino หรือ AVR STUDIO ก็ได้ (บอร์ด CPU นี้ จะ Download โปรแกรมโดยใช้ ET-AVR ISP MK II ซึ่งต้องซื้อเพิ่มถ้าผู้ใช้ยังไม่มี)
3) ET-IR REMOTE KEY เป็น Remote ส่งสัญญาณ Key ต่างๆ โดยต้องเขียนรับสัญญาณ Remote มาถอดรหัส Key เอง
4) TSOP4838+ CABLE 15 CM เป็น ตัวรับสัญญาณ IR จาก IR REMOT KEY พร้อมสายต่อตัวรับสัญญาณ IR ระหว่างบอร์ด Control กับ Display ซึ่งตัวรับสัญญาณ IR จะต้องบัดกรีลงบนบอร์ด Display เองซึ่งได้ออกแบบ PCB รองรับไว้แล้ว
5) ET-SWITCHING ADAPTER 12 V 1A Type J เป็น แหล่งจ่ายไฟเลี้ยงใช้ต่อกับบอร์ด ET-CPU-DISPLAY 8x32 ได้โดยตรง ซึ่งตัวบอร์ด Control จะมีวงจรแปลงไฟจาก 12 V เป็น 5 V เพื่อส่งไปเลี้ยงอุปกรณ์บนบอร์ดและส่งไปเลี้ยงบอร์ด Display ด้วยทางขั้วต่อ Control SPI

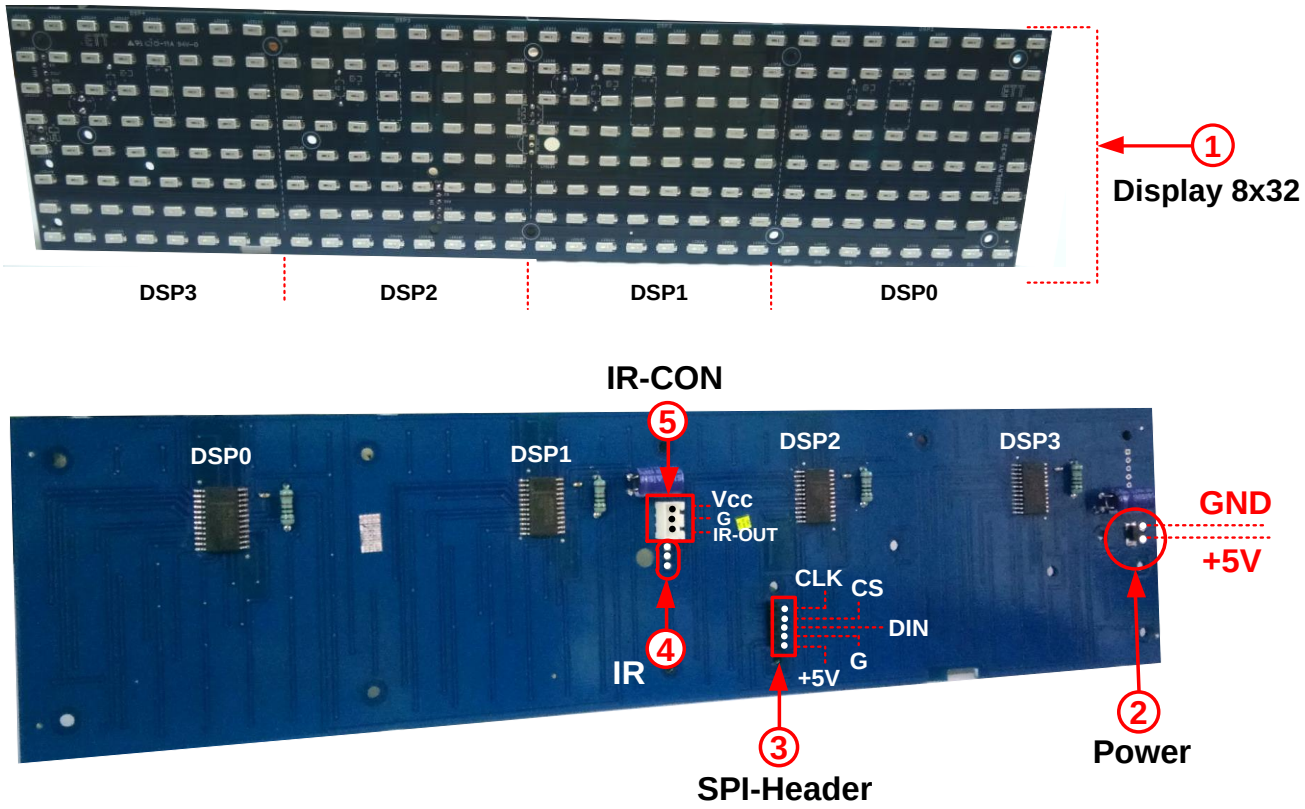
Option ในหัวข้อ (3) , (4), (5) นี้จะต้องใช้งานกับบอร์ด ET-CPU-DISPLAY 8x32 เท่านั้น ในกรณีที่ใช้บอร์ด Control อื่นๆ ผู้ใช้จะต้องจัดวงจรในส่วนของ ตัวรับสัญญาณ IR (TSOP4838) เองบนบอร์ด Control เนื่องจากวงจรในส่วนภาครับ IR ไม่ได้ออกแบบไว้ที่บอร์ด Display และในส่วนของแหล่งจ่ายไฟเลี้ยงแนะนำให้ใช้แหล่งจ่าย 12 V และหาวงจรแปลงจาก 12V เป็น 5 V , 1A จ่ายให้กับบอร์ด Display และ บอร์ด Control ของผู้ใช้เอง ไม่แนะนำให้ใช้แหล่งจ่ายแบบ Switching ที่แปลงจาก 220V เป็น 5 V โดยตรง เนื่องจากพบปัญหาเกิดสัญญาณรบกวนบอร์ด Display ทำให้ IC MAX7219 บนบอร์ด Display ค้างทำงานไม่ได้

ในส่วนของรายละเอียดเกี่ยวกับบอร์ด Control ET-CPU-DISPLAY 8x32 ในการต่อใช้งานกับบอร์ด ET-DISPLAY 8x32 RED ให้ดูใน "คู่มือ ET-CPU-DISPLAY 8x32"

1. คุณสมบัติ ET-DISPLAY 8x32 RED

- ไฟเลี้ยงบอร์ด Display DC 5 V กระแสใช้งานทั่วไปประมาณ 1 A ที่ Display สว่างสูงสุด
- Display เป็นแบบ Dot Matrix สีแดง ขนาด 8x32 Dot
- ใช้ IC Max7219 จำนวน 4 ตัว เป็นตัว Drive และ Scan ที่ความถี่ 800 Hz และสามารถควบคุมความสว่างของ Display ได้ 16 ระดับ
- ตัวบอร์ด Display มีขนาด กxย = 8x32 Cm.
- บนบอร์ดมีจุดใส่ IR TSOP4838 (Option) และขั้วต่อสาย IR สำหรับต่อไปยังบอร์ด Control ET-CPU-DISPLAY8x32(Option)
- ตัวบอร์ด Display จะใช้การเชื่อมต่อสำหรับ Control แบบ Serial SPI โดยมี 5 Pin ประกอบด้วย Control 3 Pin (CLK, CS, DIN) และ VCC , GND ซึ่ง PIN CLK รองรับความถี่ได้สูงสุด 10MHz

2. โครงสร้างและขั้วต่อใช้งาน ET-DISPLAY 8x32 RED



(1) Display8x32 : เป็น LED Display สีแดง แบบ Dot matrix ขนาด 8x32 Dot ใช้แสดงผลตามการเขียนโปรแกรมของผู้ใช้

(2) Power : เป็นขั้วต่อไฟเลี้ยง Display โดยรับแรงดันที่ +5V ,กระแส 1 A โดยจัดเรียงขั้วตามรูป ถ้ามีการจ่ายไฟเลี้ยงมาที่ขั้ว SPI-Header แล้ว ก็ไม่จำเป็นต้องจ่ายไฟเลี้ยงที่จุดนี้

(3) SPI-Header : เป็นขั้วต่อ Header 5 Pin ตัวผู้ ใช้ต่อสัญญาณ Control และ ไฟเลี้ยง +5V ให้กับ Display ซึ่งมีการจัดเรียงขาดังรูป

CLK : Seaial-Clock Input รองรับความถี่สูงสุด 10MHz เมื่อ Clock ON ที่ขอบขาขึ้น data จะถูก Shift เข้าไปยัง Register ของ Max7219 เมื่อ Clock ON ที่ขอบขาลง Data ภายใน Register จะถูก Shift ออกมาที่ Pin DOUT โดย Clock input นี้ จะทำงานเมื่อ Pin CS เป็น 0

CS : Load Data Input ทำหน้าที่ Load Serial Data 16 bit ที่เข้ามาออกไป Control LED โดยทำงานที่ขอบขาขึ้น

DIN : Serial Data Input ทำหน้าที่รับ Serial Data 16 bit เข้าไปยัง Shift Register ภายในของ Max7219

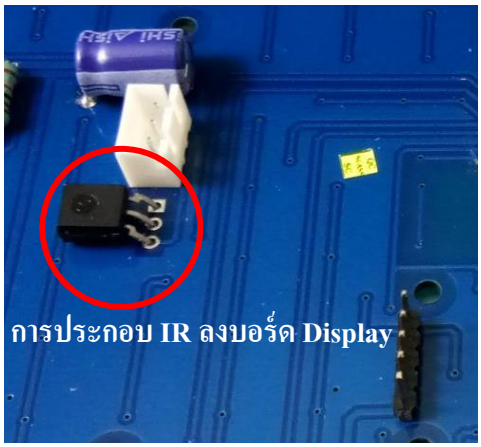
G, +5V : จุดต่อไฟเลี้ยงบอร์ด Display +5V จุดที่2

ในการเชื่อมต่อสัญญาณ Control จะเป็นแบบ SPI ระดับสัญญาณเป็น TTL(5V) ผู้ใช้สามารถนำสัญญาณ Control และไฟเลี้ยงจากบอร์ด MCU ใดๆมาต่อเข้าที่จุดนี้ก็สามารถเขียนโปรแกรมควบคุม Display ได้เลย โดยให้คำนึงถึงไฟเลี้ยงที่จะจ่ายให้ Display จะต้องเป็น 5V จ่ายกระแสได้อย่างน้อย 1 A ในกรณี

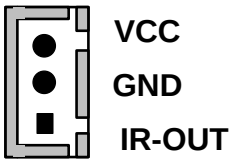
ไฟเลี้ยงจากบอร์ด MCU ที่นำมาใช้ จ่ายกระแสไม่พอ ให้ผู้ใช้จ่ายไฟเลี้ยงจากภายนอก 5 V ช่วยได้ที่จุดต่อ Power(2) บนบอร์ด Display (ควรใช้ 5V ที่ผ่านวงจร Regulator จะดีกว่า 5V จาก Switching Adapter โดยตรง เนื่องจากจะช่วยลดสัญญาณรบกวนที่ไปกวน IC Max7219 ลง มิฉะนั้นอาจทำให้ IC ทำงานผิดพลาดได้)

ในกรณีที่ใช้บอร์ด MCU รุ่น ET-CPU-DISPLAY 8x32 สามารถเสียบเข้าที่ขั้ว SPI-Header ได้ทันทีไม่ต้องต่อสายใดๆ เนื่องจากตัวบอร์ด MCU รุ่นนี้ได้ออกแบบมาสำหรับ Display รุ่นนี้โดยเฉพาะ (ดูหัวข้อที่ 4. การต่อใช้งาน) ส่วนไฟเลี้ยงบอร์ด MCU จะใช้ 12 V และถูกแปลงเป็น 5V เพื่อไปเลี้ยงอุปกรณ์บนบอร์ด MCU และบอร์ด Display

(4) IR : เป็นจุดสำหรับต่อตัว IR TSOP4838 เพื่อใช้รับสัญญาณ Remote ซึ่งเป็น Option เพิ่ม โดยผู้ใช้ต้องบัดกรีอุปกรณ์ลงบนบอร์ดเอง ดังรูปด้านล่าง



(5) IR-CON : เป็นขั้วต่อแบบ Block 3 Pin โดยมีการจัดเรียง Pin ตามรูป ซึ่งขั้วนี้จะต่อขนานอยู่กับตัว IR ในหมายเลข (4) ซึ่งจะใช้เป็นจุดดึง PIN IR บนบอด Display ไปยังบอร์ด MCU ที่นำมาใช้ Control อีกที ซึ่งจะใช้เมื่อผู้ใช้ต้องการเขียนโปรแกรมให้สามารถใช้ Remote ควบคุม Display ได้ ซึ่งขั้วทั้ง 3 Pin นี้ ไม่ได้เชื่อมต่อสัญญาณใดๆบนบอร์ด Display ดังนั้น Pin VCC และ G จะต้องดึงไปยังบอร์ด Control ด้วย



VCC,GND : ไฟเลี้ยง IR โดย ขั้ว VCC จะต้องมี R = 1 K ต่ออนุกรมไปยังแหล่งจ่าย 5 V (ดูหัวข้อที่ 4. การต่อใช้งาน)

IR : เป็นสัญญาณความถี่ Output ที่รับมาจาก Remote ตัวส่ง เพื่อส่งต่อไปยัง MCU เพื่อให้ MCU ทำการอ่านความถี่ที่ส่งมาและนำไปถอดรหัสเพื่อหาค่า Key ที่ถูกกด

3. การทำงานของ ET-Display 8x32 RED

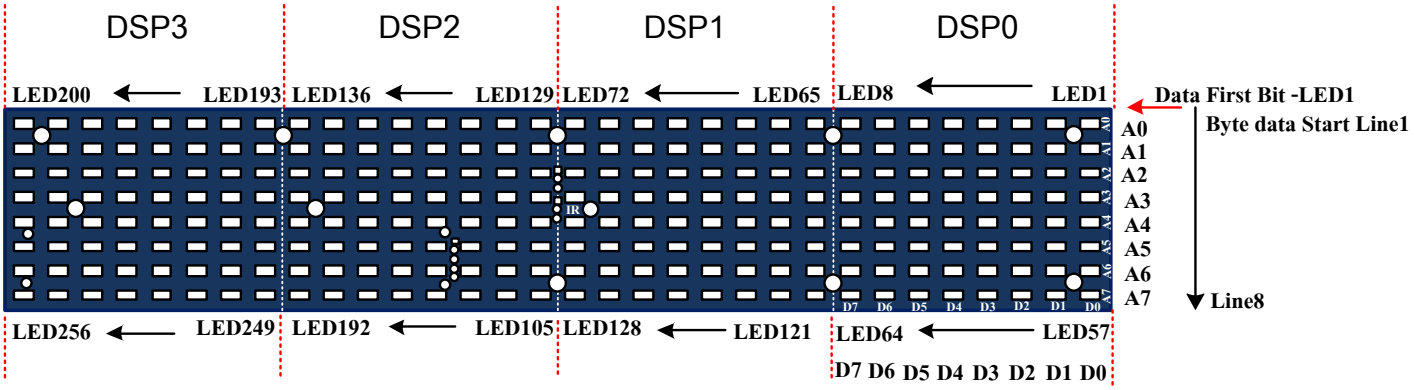


Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

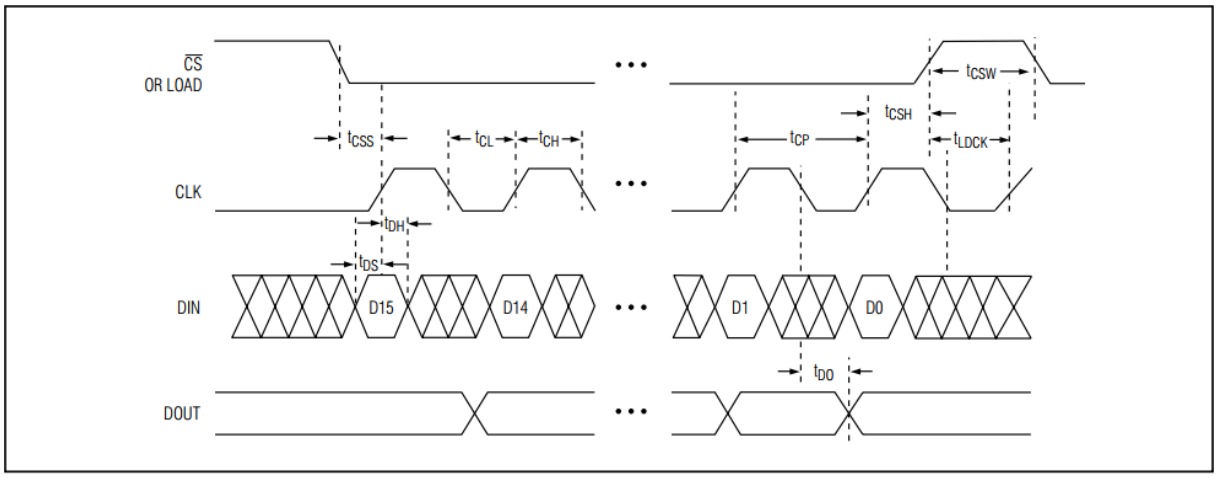


Figure 1. Timing Diagram

สำหรับ Display นี้ จะใช้ IC MAX7219 เป็นตัว Drive และรับคำสั่ง Control การทำงานจากผู้ใช้ แบบ Serial Data (SPI) โดยมีรูปแบบการส่งคำสั่งดังตารางด้านบน ขนาดคำสั่งจะเป็นแบบ 16 Bit แบ่งเป็น 8 บิตบน(D15-D8) จะเป็นในส่วนของ Address Register ของคำสั่ง หรือตำแหน่ง Address (A0-A7)หรือ Line ที่ผู้ใช้ต้องการเลือกให้ทำงานตามรูปด้านบน ส่วน 8 บิตล่าง (D7-D0) จะเป็น Data ของคำสั่ง หรือ Data ที่ต้องการ ให้ LED ดวงนั้นๆติด โดยบิตที่เป็น 1 LED จะติด ส่วน บิตที่เป็น 0 จะทำให้ LED ดับ

โดยค่า Address 0x01-0x08 (เลือกA0-A7 ตามลำดับ) จะใช้เลือก Line ที่ต้องการให้ทำงานกับชุด Data ที่ส่งออกไป ส่วน Address 0x09-0x0F จะใช้ Set ค่าพารามิเตอร์ต่างๆให้กับตัว Max7219 เช่นปรับความสว่างเป็นต้น ซึ่งรายละเอียดในส่วนนี้สามารถดูได้ใน Data Sheet ของ Max7219 สำหรับตัวอย่างเช่น ต้องการส่ง Data ออกไปควบคุมให้ LED1(D0),LED2(D1),LED3(D2) ติด ใน Line 1 (A0) Data ที่จะต้องส่งออกไปขนาด 16 บิต ก็คือ 0x0107 เป็นต้น โดยในการส่ง Data 1 บิต จะต้องใช้ Clock 1 ลูกในการ Shift Data เข้าไปใน Register (บิต 15 จะถูก Shift เข้าไปบิตแรก) ดังนั้นในการส่งข้อมูล 1 ชุด ผู้ใช้จะต้องใช้ Clock ทั้งหมด 16 ลูก โดย Data จะถูก Shift เข้าไปใน Register ที่ขอบขาขึ้น

ของ Clock และที่ขอบขาลงของ Clock จะเป็นการ Shift Data เก่าก่อนหน้า ออกมาที่ Pin DOUT ของ MAX7219 (บิต 15 จะถูก Shift ออกมาก่อน)

จากที่กล่าวไปข้างต้นจะเป็นการทำงานเบื้องต้นของ Max7219 หนึ่งตัว สำหรับ Display ของเราจะใช้ MAX7219 จำนวน 4 ตัวต่อ Pin DIN และ DOUT อนุกรมกันอยู่ (ดูวงจรประกอบท้ายคู่มือ) ส่วนขา Control อื่นๆ จะขนานกัน ดังนั้นเวลา Shift Data เข้าแต่ละครั้ง และ OUT Data ออกแสดงผลบน Display MAX7219 จะทำงานพร้อมๆกันทั้งหมด

จากรูป Display ด้านบนจะเห็นว่าเราจะแบ่งชุด Display ออกเป็น 4 ชุดย่อย คือ DSP0-DSP3 โดย MAX7219 แต่ละตัวจะควบคุม Matrix LED 64 ดวง(8x8) ตามรูปเช่น DSP0 ควบคุม LED1-LED64 ,DSP1 ควบคุม LED72-128 เป็นต้น เพื่อให้เข้าใจเวลาเขียนโปรแกรมส่งข้อมูลมาแสดงบน Display เริ่มต้นให้พิจารณาที่ DSP0 ก่อน โดยให้พิจาณา Data จาก Table1 ด้านบน Clock 8 ลูกแรกจะเป็นการ Shift Data D15-D8 เพื่อเลือกบรรทัดที่ต้องการแสดงผลให้กับข้อมูลที่ส่งตามมา ส่วน Clock 8 ลูกหลัง จะเป็นการ Shift Data D7-D0 ซึ่งจะเป็น Data ที่ใช้กำหนดให้ LED ตำแหน่งที่ต้องการติดในบรรทัดที่เลือกไว้ โดย LED8 จะถูกกำหนดโดย Bit D7, LED1 ถูกกำหนดโดย Bit D0 เป็นต้น โดยการไล่ Bit Data แสดงดังรูปด้านบน

สรุปก็คือ Data D7-D0 ที่ถูกส่งมาทุกครั้งจะเริ่มที่ DSP0 ก่อน สำหรับ Clock 16 ลูกแรก โดย Data จะShift จากขวา มาซ้ายของ Display ส่วน Line จะถูกกำหนดโดย Data Address(D15-D8) ตามที่ผู้ใช้เลือก โดย Address แรก 0x01 จะเริ่มจากบรรทัดบนสุด(A0 : Line1)

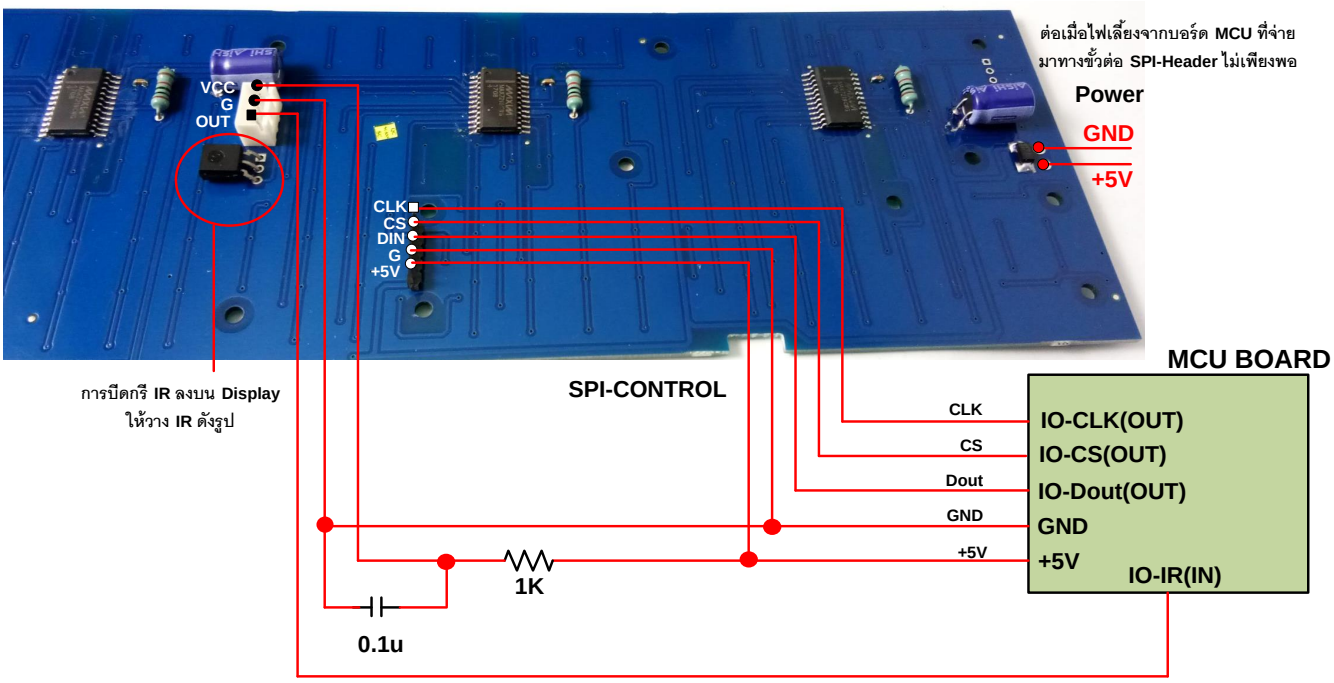
ตัวอย่างเช่น ต้องการแสดงผลใน Line 1(A0) โดยส่ง Data 4 ชุด คือ 0x01,0x02,0x03,0x04 ซึ่งเมื่อส่ง Data ครบ 4 ชุดที่ DSP3 : LED 193 จะติด(Data 0x01) , DSP2 : LED 130 จะติด(Data 0x02) , DSP3 : LED 66,65 จะติด(Data 0x03) , DSP20: LED 3 จะติด(Data 0x04) การทำงานจะเริ่มจาก เมื่อส่ง Clock ลูกที่ 1-16 : Data 0x01 จะมาอยู่ที่ DSP0(LED1-8) ต่อมาส่ง Clock ลูกที่ 17-32 :Data 0x01 จะมาอยู่ที่ DSP1 และ Data 0x02 จะถูก Shift เข้ามาอยู่ที่ DSP1 เมื่อผู้ใช้ส่ง Clock ครบ 64 ลูก Data ก็จะถูก Shift เข้ามายัง Display จนครบก็จะเห็น LED ติด ตามที่กล่าวข้างต้น สังเกตว่า ในขณะที่ Clock Shift ข้อมูลเข้ามาด้วยความไวดังนั้นผู้ใช้จะยังไม่เห็น LED ติด เมื่อ Clock หยุด Shift Data และ PIN CS (LOAD) ของ Max7219 เป็น Logic 1 เราก็จะเห็น LED ติดตามที่เราต้องการ โดย LED จะยังติดค้างอยู่ตลอดเวลาเนื่องจาก Max7219 แต่ละตัวจะทำการ Scan ข้อมูลอยู่ตลอดเวลา ดังนั้นเมื่อมีการ Shift Data ชุดใหม่เข้ามาก็จะเห็น LED แสดงผลเปลี่ยนไปตามข้อมูลชุดใหม่ที่เข้ามาทันที

ในตัวอย่างการทำงานข้างต้นจะเป็นการแสดงผล Data 1 บรรทัดของ Display เท่านั้น เมื่อต้องการให้แสดงผลเต็มหน้าจอก็จะต้องไล่ส่งข้อมูลให้ครบทุกบรรทัด ซึ่งจะใช้ Clock ทั้งสิ้น 512 ลูก ดังนั้นในการเขียนโปรแกรมก็จะต้องเขียนให้มีการ Update ข้อมูลทั้งหน้าจอ ถ้า Update เป็นบางส่วนจะทำให้ข้อมูลในตำแหน่งอื่นๆแสดงผิดพลาดได้ เนื่องจาก Data IN และ Data OUT ของ IC ต่ออนุกรมกันอยู่ การ Shift Data แต่ละครั้งจะมีผลต่อการเปลี่ยนแปลงของข้อมูลในตำแหน่งต่อไปด้วย

เพื่อให้ง่ายในการเขียนโปรแกรมแนะนำให้สร้างตัวแปร Array ขึ้นมาขนาด 8(Row:Line) x 4(Colum:DSP0-DSP3) แล้วให้จัดการข้อมูลลงในตัวแปร Array ให้เรียบร้อยก่อน แล้วทำการนำข้อมูลในตัวแปร Array ทั้งหมดส่งไปแสดงผลบน Display ทีเดียว ตามรูปแบบการทำงานที่กล่าวมาแล้วข้างต้น สามารถดูตัวอย่างโปรแกรมที่ให้ไปประกอบความเข้าใจได้

4. การต่อใช้งาน ET-DISPLAY 8x32 RED

- การต่อบอร์ด DISPLAY กับบอร์ด MCU อื่นๆ และ IR TSOP4838(Option)



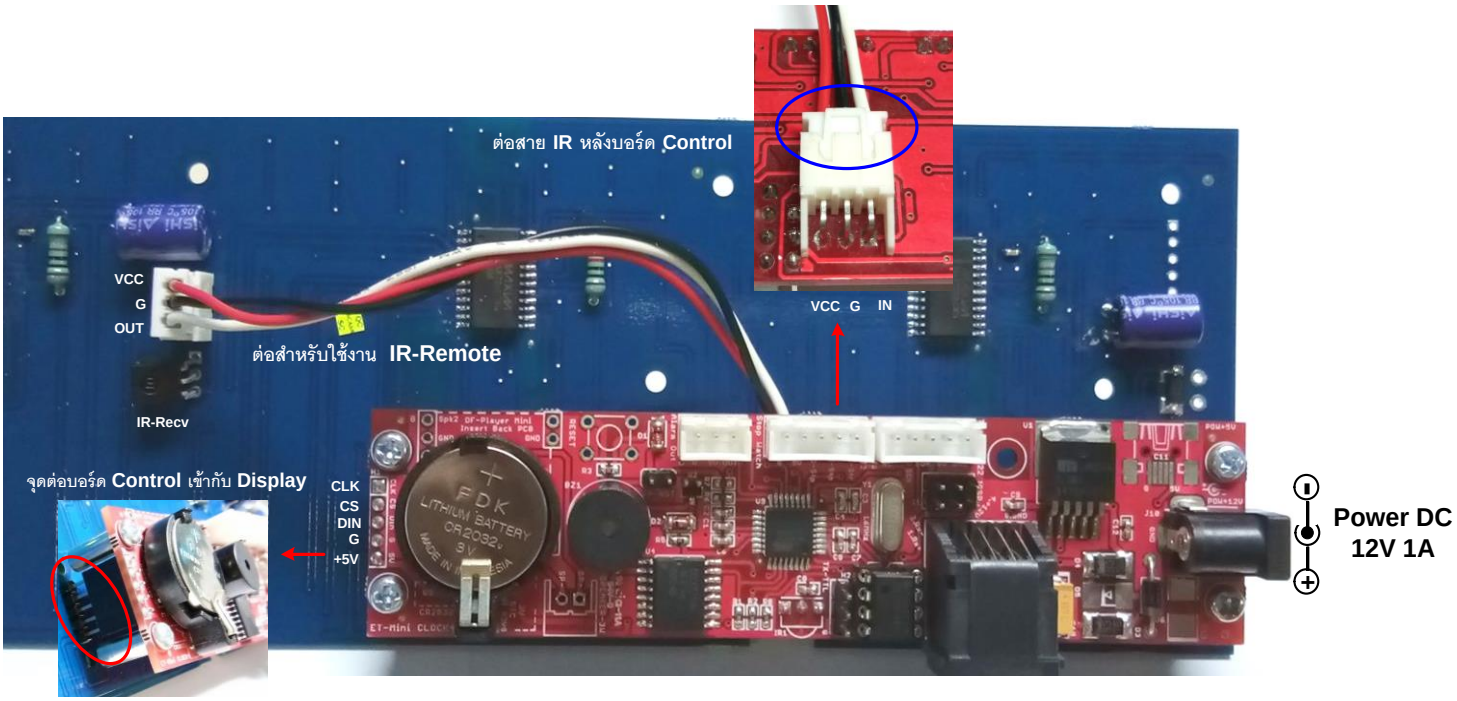
จากวงจรการต่อในส่วนของขั้วต่อ SPI-Control จะใช้ไฟเลี้ยงจากบอร์ด MCU มาเลี้ยงบอร์ด Display ด้วย ถ้าไฟเลี้ยงจากบอร์ด MCU ไม่พอเลี้ยงบอร์ด Display(~1Amp) ให้ต่อไฟเลี้ยง 5 V ที่ขั้วต่อ Power ของบอร์ด Display เสริมเข้าไปอีก 1 ชุด สำหรับสัญญาณ Control CLK จะทำหน้าที่ Shift Data ที่รออยู่ที่ Pin Dout ของMCU เข้าไปยัง Pin DIN ของ Max 7219 ตัวแรก ทีละบิต จน Data บิตแรกถูกเลื่อนเข้าไปอยู่ใน Max7219 ตัวที่4 จึงปล่อยสัญญาณ CS ที่ขอบขาขึ้น เพื่อทำการโหลด data ของ Max7219 ทั้ง 4 ตัวออกมาควบคุม Display

ในกรณีที่ต้องการใช้งาน IR Remote ผู้ใช้จะต้องบัดกรีตัวรับสัญญาณ IR เข้าไปในบอร์ด Display ก่อน โดยวาง IR ตามรูป จากนั้นก็ต่อสายสัญญาณ 3เส้นที่ขั้วต่อ Block 3 Pin ออกมาเชื่อมต่อกับวงจรเพิ่มเติมภายนอกตามรูปซึ่งผู้ใช้ต้องต่อเพิ่มเอง ก่อนจะต่อขา Out ไปยัง Pin INPUT ของ MCU เนื่องจาก ที่บอร์ด Display ได้ต่อตัว IR และ ขั้วต่อบล็อก 3 Pin ขนานกันไว้เฉยๆ ไม่มีการจัดวงจรในส่วนของการใช้งาน IR ไว้ จึงต้องต่อวงจรในส่วนนี้เพิ่มเข้าไป สำหรับสัญญาณ IR ที่ตัว IR รับเข้ามาจาก Remote ตัวส่ง จะเป็นสัญญาณความถี่ที่เข้ารหัสมาตามมาตรฐานของ Remote ตัวส่งที่ใช้ ดังนั้น เมื่อ MCU รับสัญญาณความถี่เข้ามาแล้วผู้ใช้จะต้องเขียนโปรแกรมจับความถี่ที่เข้ามาทำการถอดรหัสเพื่อจะได้แยกว่ามีการกด Key ใดเข้ามาซึ่งผู้ใช้จะต้องศึกษาในส่วนนี้เพิ่มเติมเอง

-การต่อบอร์ด DISPLAY กับบอร์ด ET-CPU-DISPLAY 8x32(Option) และ IR TSOP4838(Option)

จากรูปด้านล่าง เป็นการต่อใช้งานบอร์ด Control ET-CPU-Display 8x32 เข้ากับบอร์ด Display ในกรณีที่ผู้ใช้ไม่ได้ใช้งานในส่วนของ IR-Remote ก็ไม่จำเป็นต้องต่อสาย IR-Remote ให้ประกบเฉพาะในส่วนบอร์ด Control และบอร์ด Display เข้าด้วยกันตามรูปด้านล่างเท่านั้น หลังจากนั้นจ่ายไฟ DC 12V ประมาณ 1A ขึ้นไป เข้าที่ Jack Power ของบอร์ด Control ซึ่งแรงดันนี้จะถูกบอร์ด Control แปลงลงเหลือ 5V เพื่อใช้เลี้ยงบอร์ด Control และบอร์ด Display ต่อไป ผู้ใช้ก็จะสามารถเขียนโปรแกรมควบคุม Display และ Download โปรแกรมลงบนบอร์ด Control ได้แล้ว

สำหรับบอร์ด Control ET-CPU-Display 8x32 จะใช้ CPU AVR Mega328 ผู้ใช้สามารถเขียนโปรแกรมควบคุม Display ด้วยโปรแกรม Arduino ได้ จากนั้น Compile เป็น Hex File ออกมา แล้วใช้ ตัว ET-AVR ISP mkII (รองรับ Window7 สูงสุด) เป็นตัว Download ต่อเข้าทาง ขั้วต่อ 6 Pin บนบอร์ด Control และต้องใช้โปรแกรม AVR Studio ในการ Download โดยวิธีการ Download โปรแกรม การ Compile File และ Pin I/O ที่ต่อไปควบคุม บอร์ด Display และวงจรของบอร์ด ET-CPU-DISPLAY 8x32 สามารถดูได้ในคู่มือของบอร์ด Control ET-CPU-Display 8x32



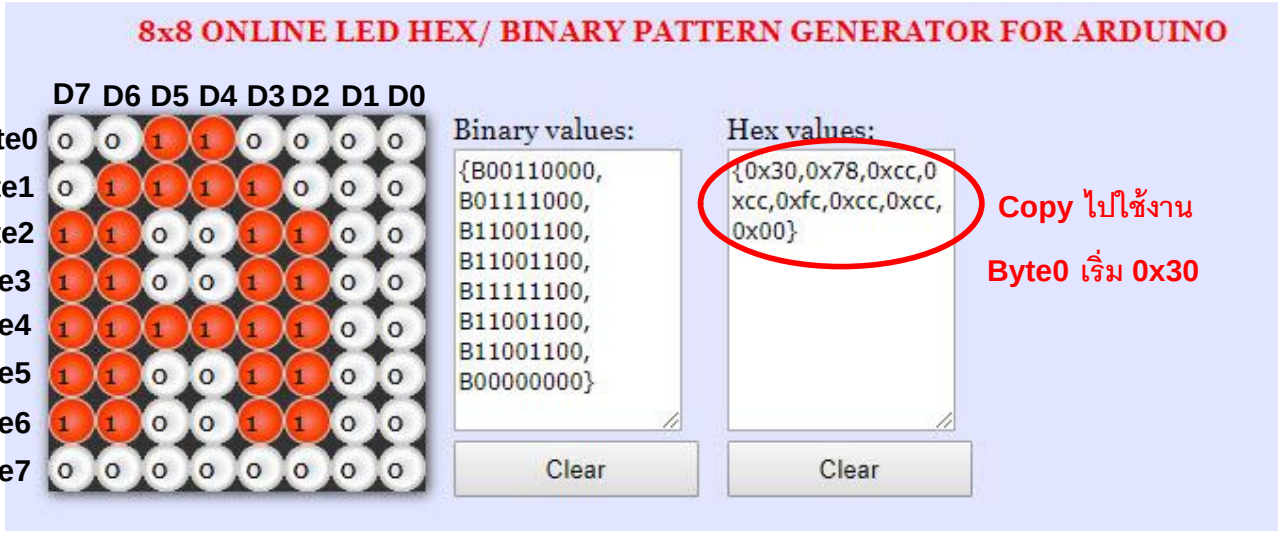
รูปการต่อบอร์ด Control ET-CPU DISPLAY8x32 เข้ากับบอร์ด ET-DISPLAY8x32RED

5. ตัวอย่างโปรแกรม ET-DISPLAY 8x32 RED

สำหรับตัวอย่างที่ให้ไปจะพัฒนาด้วย โปรแกรม Arduino โดยตัวอย่างจะอ้างอิงกับบอร์ด ET-CPU-Display 8x32 ซึ่งเป็นบอร์ดที่ทำรองรับบอร์ด Display นี้โดยตรงใช้ CPU AVR Mega328 ในการ Download โปรแกรมจะต้องใช้ file.hex Download ผ่านกล่อง ET-AVR ISP mkII โดยใช้โปรแกรม AVR Studio ในการ Download โดยวิธีการ Download โปรแกรม การ Compile File และ Pin I/O ที่ต่อไปควบคุม บอร์ด Display และวงจรของบอร์ด ET-CPU-DISPLAY 8x32 สามารถดูได้ในคู่มือของบอร์ด ET-CPU-Display 8x32

ในตัวอย่างโปรแกรมที่ 4, 5 , 6 ผู้ใช้จะต้องทำการ Add Library ใน Folder Lib_Use เข้ามาในโปรแกรมด้วยมิฉะนั้นจะทำให้ Compile error ได้โดยดูวิธีการ Add Library ได้ในคู่มือ บอร์ด ET-CPU-Display 8x32

สำหรับ Font ที่ใช้ในตัวอย่างต่างๆผู้ใช้สามารถเข้าไปสร้างเองได้ที่ Link “ https://www.riyas.org/2013/12/online-led-matrix-font-generator-with.html ” ซึ่งมีหน้าตาดังรูปด้านล่าง โดยพื้นที่สร้าง ตัวอักษรจะมีขนาด 8x8 dot เมื่อสร้างเสร็จให้ผู้ใช้ Copy ค่าในช่อง Hex value ไปแปะไว้ในโปรแกรมที่ประกาศตัวแปรเป็น const unsigned char font[ ] PROGMEM ={ data } ; ได้เลย สำหรับการจัดเรียงบิต Data และ Byte data ของโปรแกรมสร้าง Font แสดงดังรูปด้านล่าง



ตัวอย่างโปรแกรม Ex1 : จะแสดง LED bit 0 วิ่งจาก DSP0 ไป DSP3 จากนั้น ก็จะมีเสียง Beep แล้วแสดง LED ทั้งหมดของ 1 DSP (8x8) วิ่งจาก DSP0 ไป DSP3 วนสลับเช่นนี้ไปเรื่อยๆ

ตัวอย่างโปรแกรม Ex2.1 : จะแสดงตัวเลข 0-3 วิ่งจาก DSP0 ไปยัง DSP3 วนซ้ำไปเรื่อยๆ

*ตัวอย่างโปรแกรม Ex2.2 :*จะแสดงตัวเลข 0-3 พร้อมกันใน DSP3-DSP0 ตามลำดับ แล้วเปลี่ยนเป็น 4-7 และ 8-9 วนซ้ำไปเรื่อยๆ

ตัวอย่างโปรแกรม Ex3.1 : จะแสดงการนับจาก 0-9 ใน DSP0 วนซ้ำไปเรื่อยๆ

ตัวอย่างโปรแกรม Ex3.2 : จะแสดงการนับแบบ 4 หลักจาก 0000-9999 เมื่อนับถึงค่าสูงสุดก็จะกลับมาเริ่มนับที่ 0000 ใหม่

ในตัวอย่างที่ 4.1 และ 4.2 จะมีการ Include File DS3231_CLOCK.h มาใช้งาน ดังนั้นก่อน Compile โปรแกรม ผู้ใช้จะต้องทำการ Add Library Folder Lib_DS3231_CLOCK_Ext ที่อยู่ใน Folder Lib_Use เข้ามาในโปรแกรม Arduino ก่อน โดยดูวิธีการ Add ได้ในคู่มือ ET-CPU-DISPLAY 8x32

ตัวอย่างโปรแกรม Ex4.1 : จะเป็นการอ่านค่าเวลา นาที และ วินาที จาก RTC DS3231 มาแสดงผลบน DSP3-DSP0

ตัวอย่างโปรแกรม Ex4.2 : จะเป็นการอ่านค่าอุณหภูมิ จากตัว RTCDS3231 มาแสดงผล หน่วยเป็นองศาเซลเซียส

ในตัวอย่างที่ 5.1 และ 5.2 จะมีการ Include File DHT.h มาใช้งาน ดังนั้นก่อน Compile โปรแกรม ผู้ใช้จะต้องทำการ Add Library Folder Lib_DHT_AM2302_Ext ที่อยู่ใน Folder Lib_Use เข้ามาในโปรแกรม Arduino ก่อน โดยดูวิธีการ Add ได้ในคู่มือ ET-CPU-DISPLAY 8x32

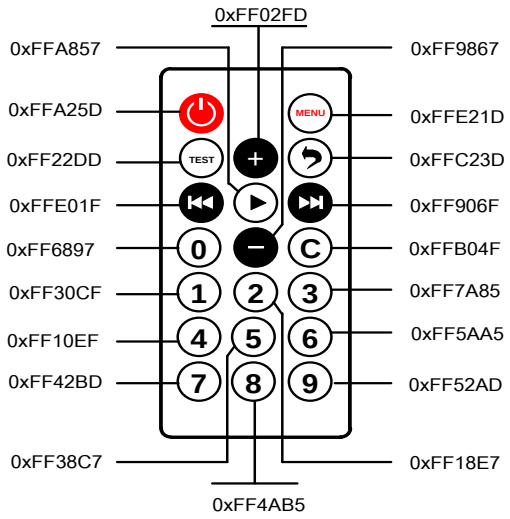
ตัวอย่างโปรแกรม Ex5.1 : เป็นการอ่านค่าอุณหภูมิจาก Temp Sensor AM2302 มาแสดงผลที่ DSP3-DSP0

ตัวอย่างโปรแกรม Ex5.2 : เป็นการอ่านค่า %ความชื้นจาก Temp Sensor AM2302 มาแสดงผลที่ DSP3-DSP0

ในตัวอย่างที่ 6 จะมีการ Include File IR_CLOCK .h มาใช้งาน ดังนั้นก่อน Compile โปรแกรม ผู้ใช้จะต้องทำการ Add Library Folder Lib_IR_CLOCK_Ext ที่อยู่ใน Folder Lib_Use เข้ามาในโปรแกรม Arduino ก่อน โดยดูวิธีการ Add ได้ในคู่มือ ET-CPU-DISPLAY 8x32

ตัวอย่างโปรแกรม Ex6 : การทดสอบตัวอย่างนี้ผู้ใช้จะต้องต่อ IR ที่บอร์ด Display และต่อสาย IR มายังบอร์ด Control และใช้ Remote ที่ทางอีทีทีจำหน่ายมาใช้งาน เมื่อ Run โปรแกรม Display จะดับหมด เมื่อผู้ใช้กด Remote Key0-9 ที่ DSP0 ก็จะแสดงหมายเลขตาม Key ที่ถูกกด

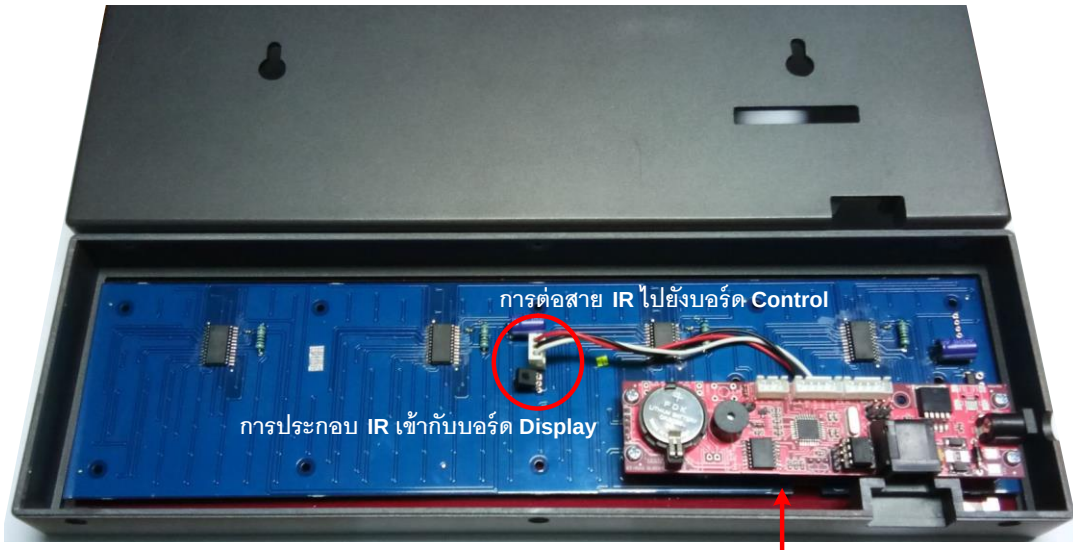
Key Code ET-IR Remote Key (3Byte)



ตัวอย่างโปรแกรม Ex7.1 : ที่ DSP0 จะแสดง ตัวเลข 0-9 และตัวอักษร A-Z เปลี่ยนไปเรื่อยๆ

*ตัวอย่างโปรแกรม Ex7.2 :*แสดงตัวอักษร abcd ที่ DSP3-DSP0 พร้อมกัน แล้วเปลี่ยนเป็น 1234 และ _567 วนไปเรื่อยๆ

6. การประกอบ ET-DISPLAY 8x32 RED เข้ากับ Option ต่างๆ



การประกอบบอร์ด **Control** และบอร์ด **Display** เข้าด้วยกัน

1) ประกอบ IR-Rec เข้ากับบอร์ด Display ถ้าต้องการใช้งาน (ถ้าไม่ใช้งานข้ามไปข้อ3)

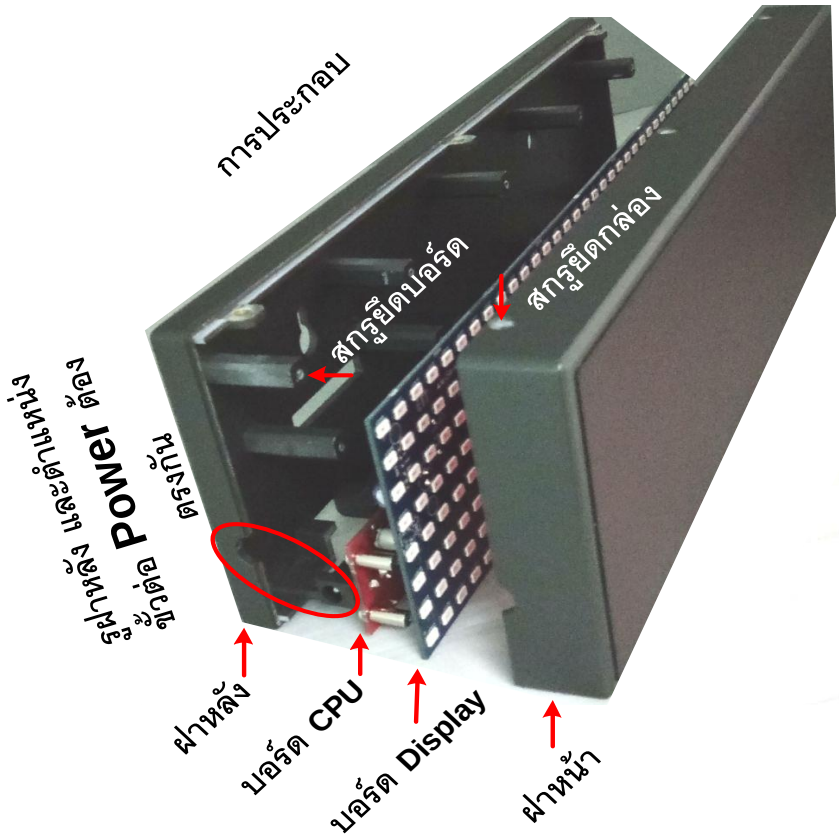
2) ต่อสาย IR ไปยังบอร์ด Control ถ้ามีการใช้งาน IR

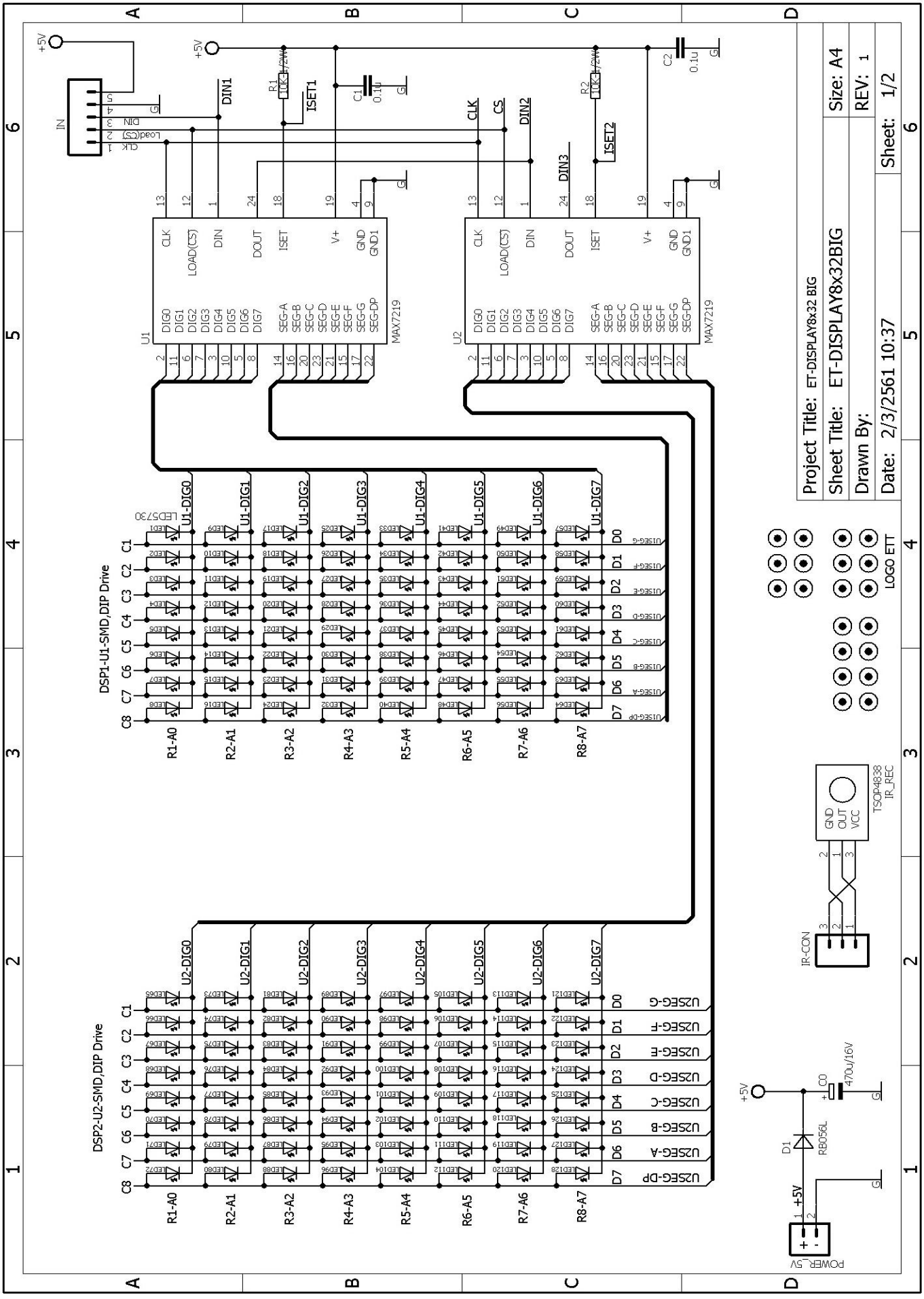
3) ประกอบบอร์ด Display และบอร์ด Control เข้าด้วยกันขันน็อตยึดทั้ง 4 มุมให้เรียบร้อย

4) วางบอร์ด Display ไว้บนฝาหลังด้านในของกล่องโดยหันด้าน LED ขึ้นสังเกต ขั้วต่อ Power ของบอร์ด Control จะต้อง ตรงกับรูด้านข้างของฝาหลัง

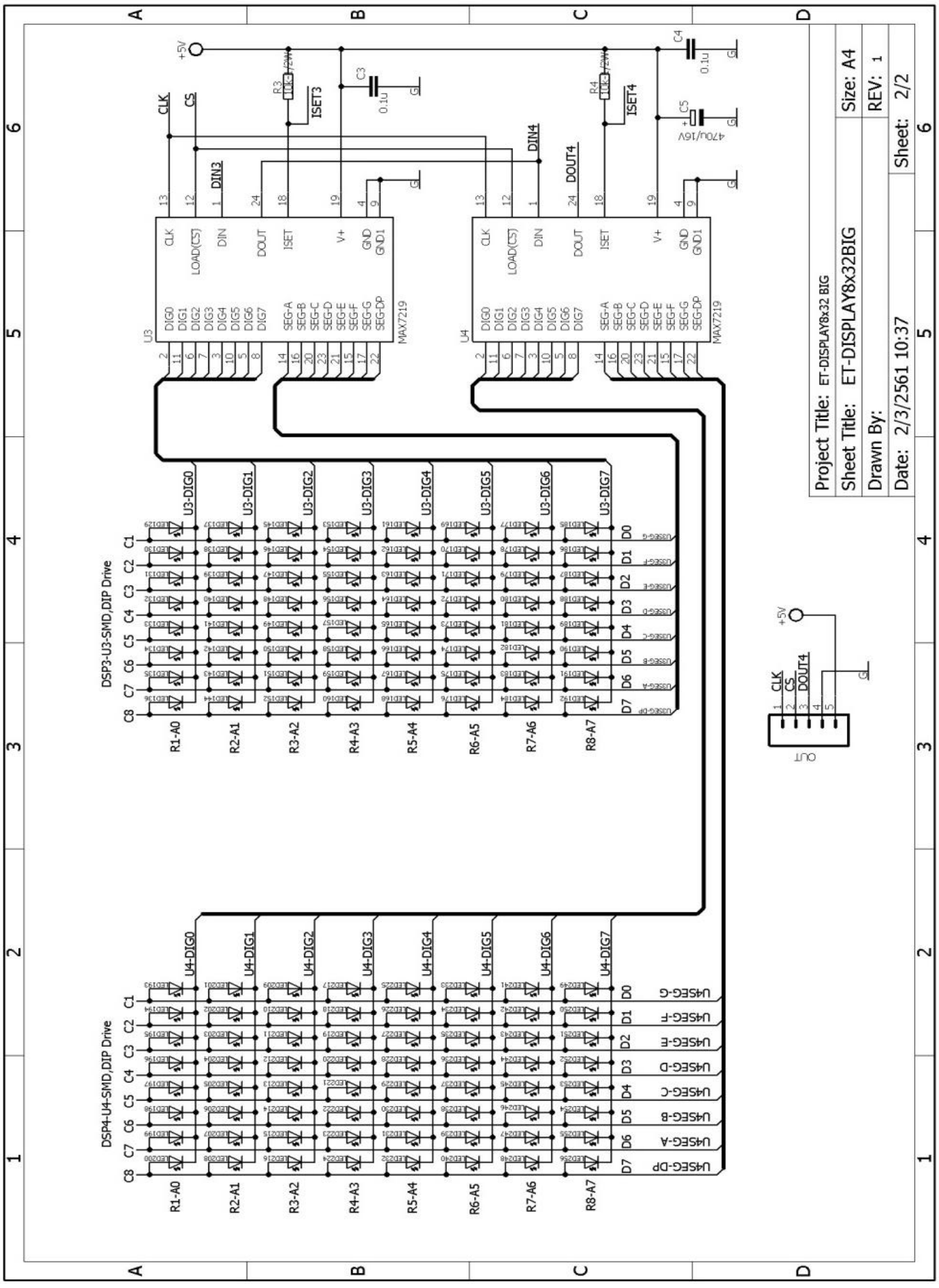
5)ใช้สกรูยึดบอร์ด Display เข้ากับฝาหลังของกล่อง

6) ประกอบฝาหน้าเข้ากับฝาหลัง แล้วขันสกรูให้เรียบร้อย





วงจร ET-DISPLAY 8x32 RED Sheet1



วงจร ET-DISPLAY 8x32 RED Sheet2