ใบงานที่ 14

การแสดงผลบนอุปกรณ์ 7 Segment

จุดประสงค์การเรียนรู้

1.ศึกษาการนับตัวเลข 0-F โดยใช้ไอซี 74HC595

2.ศึกษาการแสดงตัวเลข 4 หลักโดยใช้โมดูลไอซี Tm1637

เครื่องมือและอุปกรณ์การทดลอง

1.เครื่องไมโครคอมพิวเตอร์

2.บอรด์ NodeMCU ESP32

3.โปรแกรมการทดลอง

4.อุปกรณ์อิเล็กทอรนิกส์สำหรับทดลอง

การทดลองที่ 1 การนับตัวเลข 0-F โดยใช้ไอซี 74HC595

อุปกรณ์ในการทดลอง

1.บอรด์ NodeMCU ESP32

2.Segment ชนิด Common Cathode

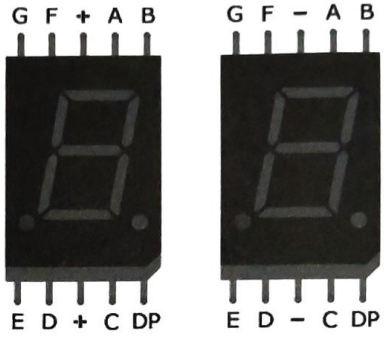
3.ตัวต้านทาน 220 โอห์ม จำนวน 8 ตัว

4.ไอซีเบอร์ 74HC595

5.แผงต่อวงจร

6.สายไฟต่อวงจร

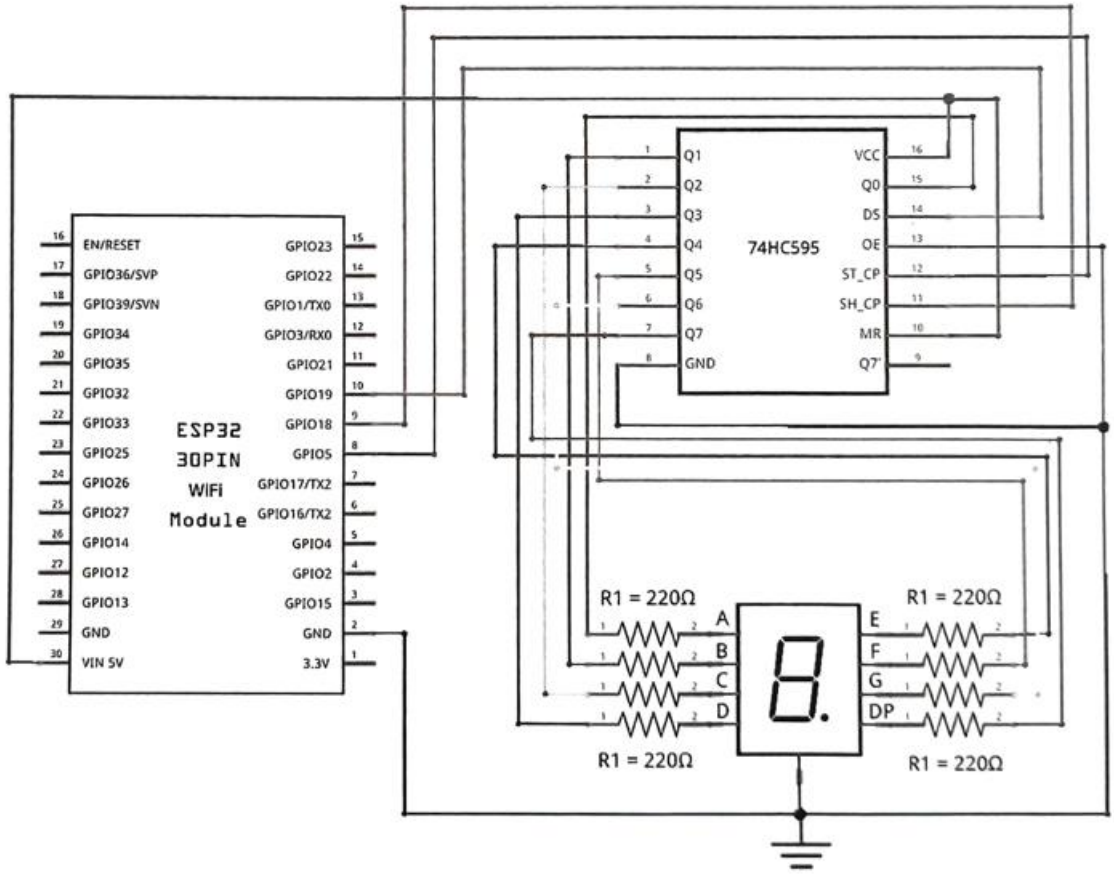
การนับตัวเลข 0-F โดยใช้ไอซี 74HC595

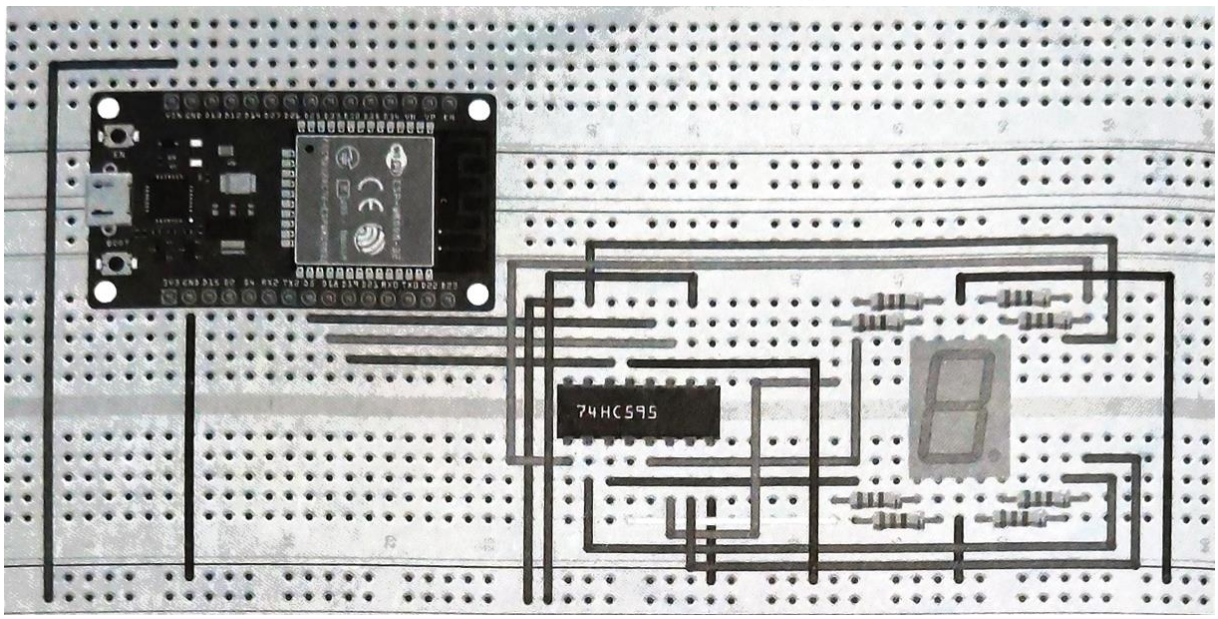


ขาบวกร่วมหรือ Common Anode *ขาลบร่วมหรือ Common Cathode*

ในตัวอย่างนี้จะเป็นการทดลองใช้อุปกรณ์ 7 Segment แบบหลักเดียวมาแสดงผลเป็นตัวเลขและตัวอักษร ด้วยการนับตัวเลขและตัวอักษรตั้งแต่ 0-F วนซ้ำไปเรื่อยๆ โดยเว้นระยะให้มีการเปลี่ยนตัวเลขทุกๆ 1 วินาที สมมติว่าในกรณีนี้บอร์ดที่เราใช้มีจำนวนขา Digital Output ที่จำกัด ซึ่งอาจทำให้ไม่เหลือขาเพียงพอที่จะใช้งานร่วมกับอุปกรณ์อื่นๆ เพิ่มเติมได้ ดังนั้นเราจึงได้นำเอาไอซีเบอร์ 74HC595 ที่เป็นไอซีสำหรับเลื่อนบิต (Shift Register) เข้ามาใช้ เพื่อช่วยลดจำนวนขา ที่ต้องใช้เชื่อมต่อกับบอร์ด NodeMCU โดยรับข้อมูลเป็นแบบอนุกรมเข้า มาทีละบิตจนครบ 8 บิต ก่อนจะส่งเป็นข้อมูลแบบขนานออกไปยังอุปกรณ์ 7 Segment เพื่อแสดงผลเป็นตัวเลขและตัวอักษรต่างๆ ต่อไป

ประกอบวงจรตามรูป



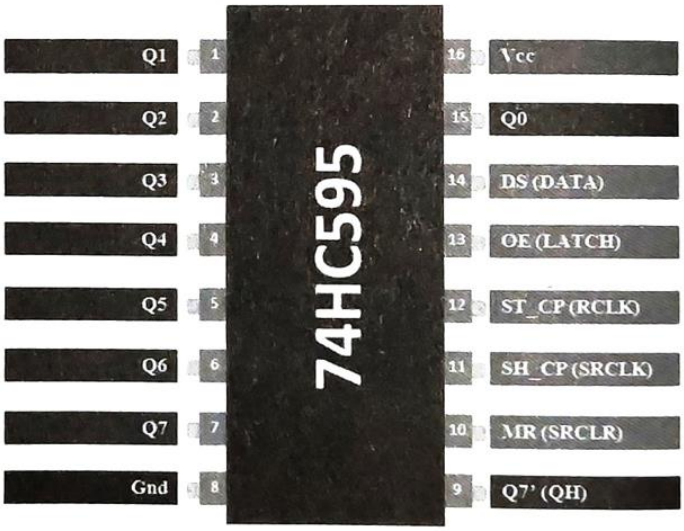


ลงมือต่อวงจรตามรูป โดยรายละเอียดขาทั้ง 16 ของไอซีเบอร์ 74HC595 มีดังนี้

• ขา Q0 - Q7 มีจำนวนทั้งสิ้น 8 ขา เป็นขาเอาต์พุตที่เอาไว้ควบคุมการปิด/เปิดหลอดไฟ LED ในแต่ละตำแหน่ง

• ขา Q7 (QH) เป็นขาที่ใช้เชื่อมต่อกับขา DS ของไอซี 74HC595 ตัวอื่น เพื่อส่งข้อมูล แบบ Serial ให้ทีละบิต

• ขา MR (SRCLR) เป็นขาที่ใช้รีเซ็ตข้อมูล ใน Shift Register ด้วยการต่อขานี้ลง GND แต่ถ้าเป็นการใช้งานปกติทั่วไป เรามักจะต่อ ขานี้ไว้กับ VCC เสมอ



• ขา SH_CP (SRCLK) เป็นขาที่รับเอาสัญญาณพัลส์หรือสัญญาณนาฬิกาเข้ามา เพื่อใช้ควบคุมจังหวะการส่งข้อมูลในแต่ละบิต

• ขา ST_CP (RCLK) เป็นขาที่ใช้ควบคุมการส่งข้อมูลว่าจะส่งไปตอนไหนเมื่อใด โดยจะทำการส่งข้อมูลเมื่อขานี้ถูกกำหนดให้มีสถานะเป็น LOW และจะหยุดส่งข้อมูลเมื่อขานี้ถูกกำหนดให้มีสถานะเป็น HIGH • OE (LATCH) เป็นขาที่ใช้ควบคุมการทำงานให้กับขาเอาต์พุตทั้ง 8 ขา (Q0 - Q7) โดยถ้าหากขานี้ ถูกต่อลง GND จะทำให้มีสัญญาณเอาต์พุตออกไปที่ขา 20 - Q7 แต่ถ้าหากต่อกับ VCC จะทำให้ไม่มี สัญญาณเอาต์พุตออกไปที่ขา 20 - Q7

• ขา DS (DATA) เป็นขาอินพุตที่เอาไว้ใช้สำหรับป้อนข้อมูลเข้าไปทีละบิต

สรุปการเชื่อมต่อสายสัญญาณจากไอซี 74HC595 ไปยังขาต่างๆ ของ 7 Segment และจากไอซี 74HC595 ไปยังบอร์ด NodeMCU แบบคร่าวๆ ได้ดังนี้

ขาของ 74HC595	**ขาของ 7 Segment หรือ NodeMCU**
ขา 1 (Q1) - ขา 7 (Q7)	ขา B, C, D, E, F, G และ DP ของ 7 Segment (ผ่านตัวต้านทาน)
ขา 8 (GND)	ขา GND ของ NodeMCU (GND OV)
ขา 9 (Q7')	
ขา 10 (MR)	ขา VIN ของ NodeMCU (Vcc +5V)
ขา 11 (SH_CP)	ขา D18 หรือ GPI018 ของ NodeMCU
ขา 12 (ST_CP)	ขา D5 หรือ GPI05 ของ NodeMCU
ขา 13 (08)	ขา GND ของ NodeMCU (GND OV)
ขา 14 (DS)	ขา D19 หรือ GPI019 ของ NodeMCU
ขา 15 (Q0)	ขา A ของ 7 Segment (ผ่านตัวต้านทาน)
ขา 16 (VCC)	ขา VIN ของ NodeMCU (Vcc +5V)
-	ต่อขาร่วมของ 7 Segment กับ GND ถ้าเป็นชนิด Common

Cathode หรือต่อกับ Vcc ถ้าเป็นชนิด Common Anode

หลังจากลงมือเชื่อมต่อสายสัญญาณต่างๆ เรียบร้อยแล้ว ขั้นตอนต่อไปก็คือ การเขียนโค้ดและอัพโหลดโปรแกรม ซึ่งมีรายละเอียดดังนี้

```
int latchPin = 5;          //ประกาศตัวแปร latchPin กำหนดให้ขา D5/GPIO5 ของ
                           NodeMCU เป็นขาที่ใช้ควบคุมการส่งข้อมูล
int clockPin = 18;         //ประกาศตัวแปร clockPin กำหนดให้ขา D18/GPIO18 ของ NodeMCU
                           เป็นขาที่ให้สัญญาณนาฬิกามาควบคุมจังหวะการส่งข้อมูล
int dataPin = 19;          //ประกาศตัวแปร dataPin กำหนดให้ขา D19/GPIO19 ของ
                           NodeMCU เป็นขาที่ใช้สำหรับส่งข้อมูล
int num[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F,
0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71};   //ประกาศตัวแปรแบบอาร์เรย์ เพื่อเก็บข้อมูลเป็นไบต์
                                       ก่อนที่จะส่งไปยังไอซี 74HC595
void setup() {
  pinMode(latchPin, OUTPUT);     //กำหนดให้ขา D5/GPIO5 ของ NodeMCU เป็น OUTPUT
  pinMode(clockPin, OUTPUT);     //กำหนดให้ขา D18/GPIO18 ของ NodeMCU เป็น OUTPUT
  pinMode(dataPin, OUTPUT);      //กำหนดให้ขา D19/GPIO19 ของ NodeMCU เป็น OUTPUT
  digitalWrite(latchPin, HIGH);  //กำหนดให้ขา latchPin (ขา D5/GPIO5 ของ NodeMCU) มีสถานะ
                                 เป็น HIGH เพื่อหยุดไม่ให้มีการส่งข้อมูลไปยังไอซี 74HC595
}
void loop() {
  for (int i=0; i<16; i++) {     //ใช้คำสั่ง for วนรอบการทำงานทั้ง 16 รอบ ไปเรื่อยๆ
                                 โดยเริ่มจาก 0 ไปจนถึง 15 ตามค่า i
    DataOut(num[i]);             //เรียกฟังก์ชั่น DataOut() เพื่อแสดงข้อมูลตัวเลขที่อยู่ใน
                                 ตัวแปรอาร์เรย์ ซึ่งจะเปลี่ยนไปตามค่าของ i
    delay(1000);                 //หน่วงรอเป็นเวลา 1 วินาที
  }
}
void DataOut(byte data) {        //ประกาศฟังก์ชั่น DataOut ผ่านตัวแปร data ชนิด byte
  digitalWrite(latchPin, LOW);   //กำหนดให้ขา latchPin (ขา D5/GPIO5 ของ NodeMCU) มีสถานะ
                                 มีสถานะเป็น LOW เพื่อยอมให้มีการส่งข้อมูลไปยังไอซี 74HC595
  shiftOut(dataPin, clockPin, MSBFIRST, data);  //ใช้ฟังก์ชั่น shiftOut() โดยกำหนดค่า
                                 พารามิเตอร์ต่างๆ เช่น ทยอยส่งข้อมูลออกไปทีละบิตผ่านทางขา dataPin,
                                 ใช้สัญญาณนาฬิกาจากขา clockPin เป็นตัวให้จังหวะ, MSBFIRST ส่งข้อมูล
                                 บิตลำดับสูงไปก่อน และข้อมูลที่ส่งไปเป็นข้อมูลจากตัวแปร data ชนิด byte
  digitalWrite(latchPin, HIGH);  //กำหนดให้ขา latchPin มีสถานะเป็น HIGH เพื่อหยุดหรือ
                                 ยกเลิกการส่งข้อมูลไปยังไอซี 74HC595
}
```

หลังจากอัพโหลดโค้ด โปรแกรมลงบนบอร์ด ถ้าทุกอย่าง ถูกต้อง บนอุปกรณ์ 7 Segment จะแสดงผลเป็นตัวเลขและตัวอักษร โดยเริ่มนับตั้งแต่ 0 ไปจนถึง F วน ไปเรื่อยๆ

บันทึกผลการทดลอง

.....

.....

.....

.....

การทดลองที่ 2 การแสดงตัวเลข 4 หลักโดยใช้โมดูลไอซี Tm1637

อุปกรณ์ในการทดลอง

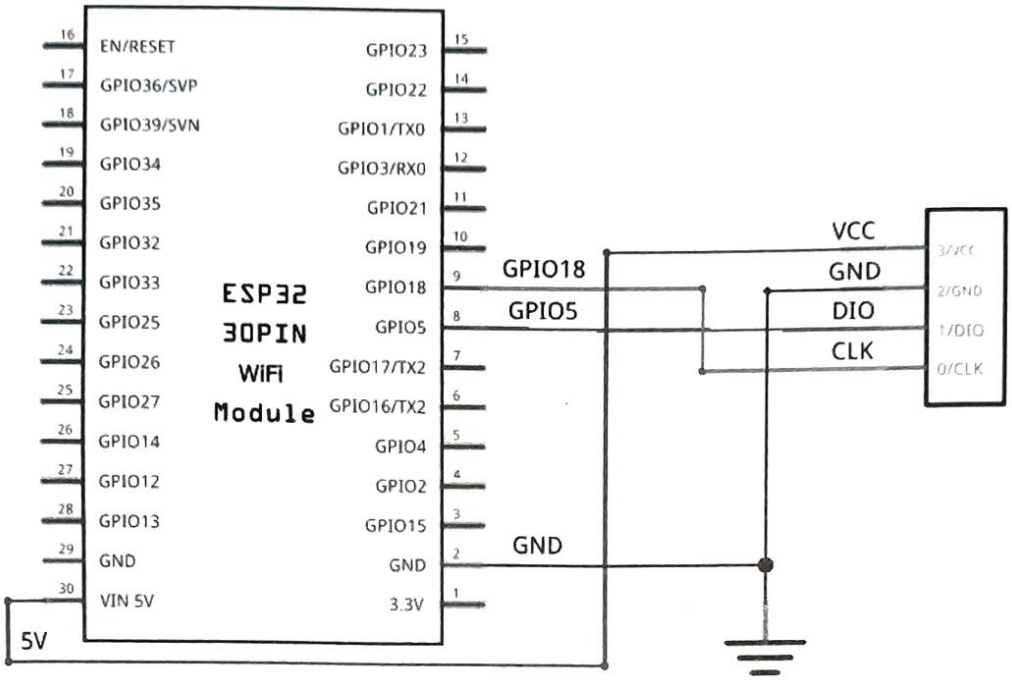
1.บอรด์ NodeMCU ESP32

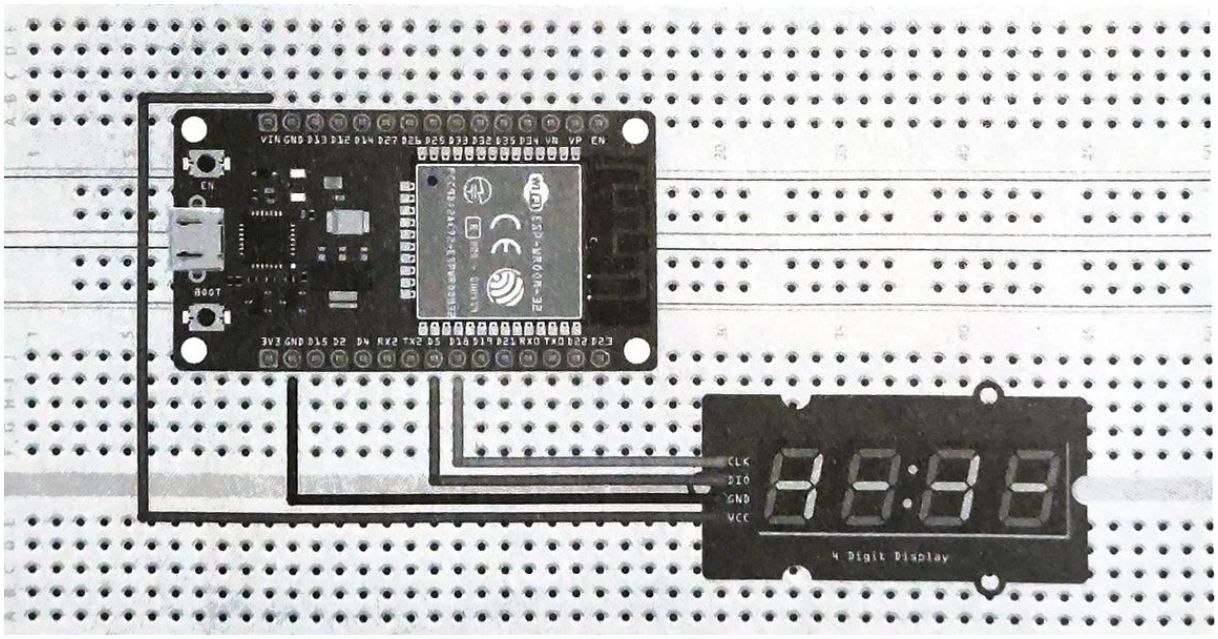
2.อุปกรณ์โมดูล 7 Segment แบบ 4 หลัก ที่ใช้ไอซีเบอร์ TM1637

3.แผงต่อวงจร

4.สายไฟต่อวงจร

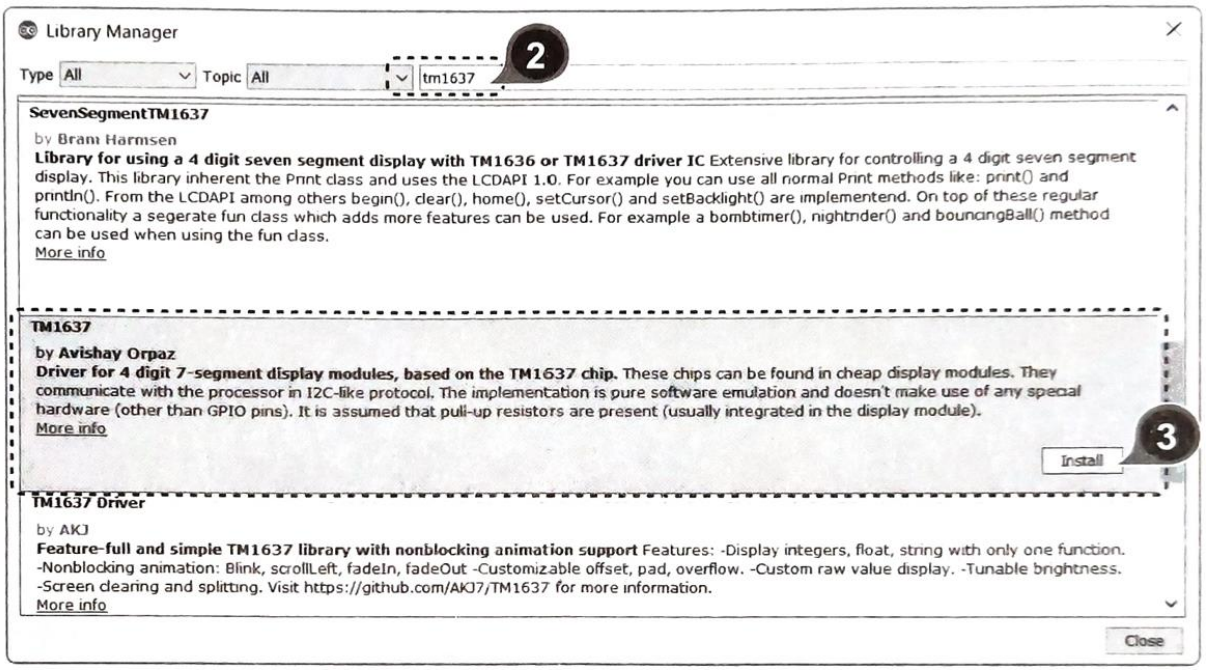
ประกอบวงจรตามรูป





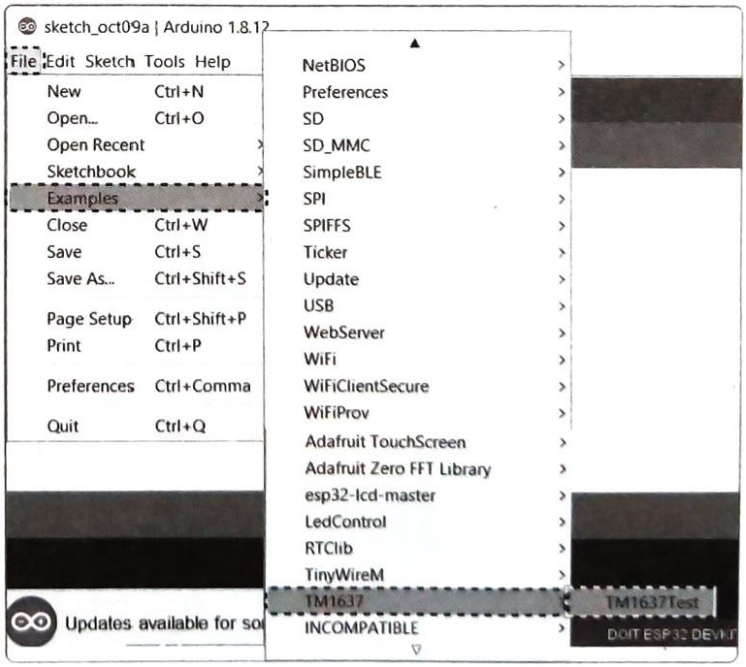
หลังจากลงมือเชื่อมต่อสายสัญญาณเรียบร้อยแล้ว ขั้นตอนต่อไปก็คือ การเขียนโค้ดและอัพโหลดโปรแกรม แต่ในที่นี้เนื่องจากโมดูลจอ 7 Segment จะมีชิป IC Controller ที่ใช้ควบคุมการแสดงผลมาให้ในตัว เพราะฉะนั้น ก่อนลงมือเขียนโค้ด เราจะต้องติดตั้งไลบรารี่ (Library) ที่จำเป็น ซึ่งในที่นี้คือไลบรารี่ TM1637Display.h ให้ กับ Arduino IDE เสียก่อน โดย

1. Arduino IDE คลิกเมนู Tools Manage Libraries... หรือกดคีย์ Ctrl + Shift +I
2. ในช่อง Search พิมพ์คำว่า tm1637 แล้วกดคีย์ Enter
3. ที่ไลบรารี่ TM1637 by Avishay Orpaz ดังรูป คลิกปุ่ม Install เพื่อติดตั้ง หรือหากใครไม่สะดวก ที่จะติดตั้งผ่าน Library Manager ก็สามารถเข้าไปดาวน์โหลดไฟล์ไลบรารี่ดังกล่าวนี้มาติดตั้งที่เครื่องได้ โดยไปที่ https://github.com/avishorp/TM1637

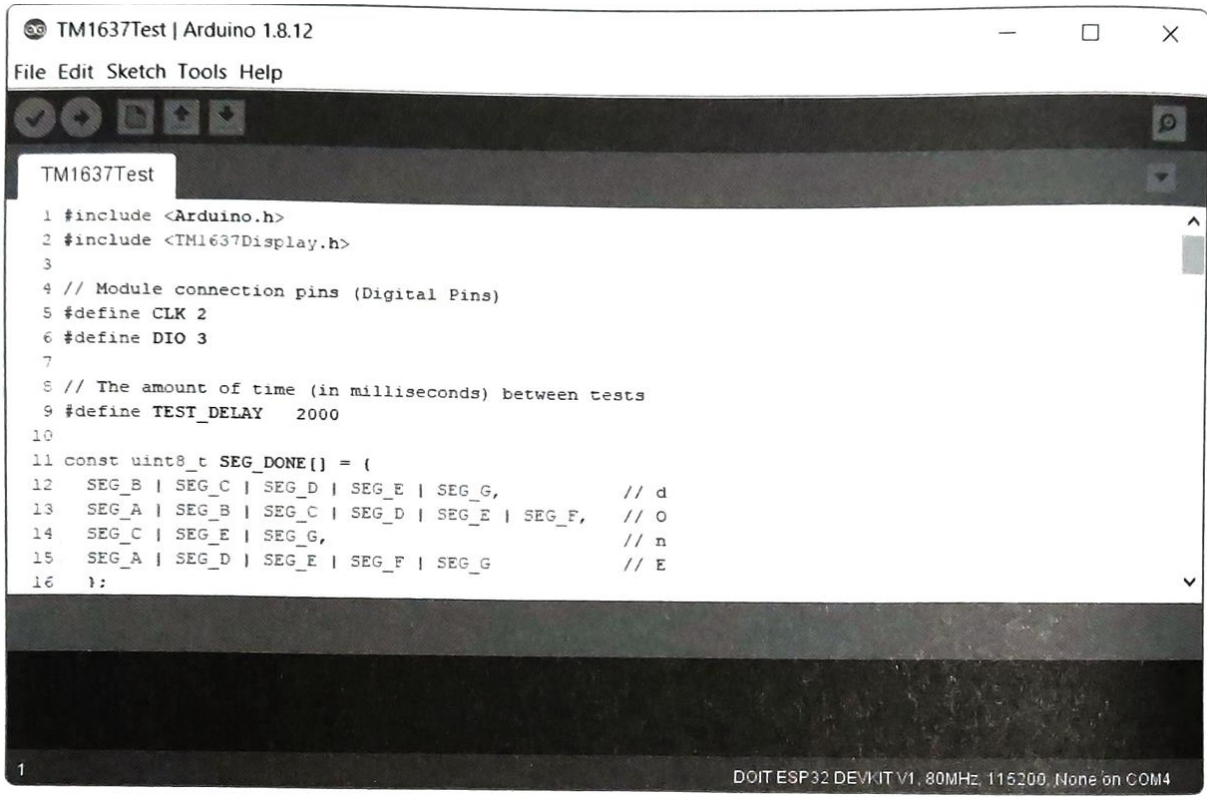


หลังจากติดตั้งไลบรารี่ (Library)เสร็จแล้ว ก็ลงมือเขียนโค้ดและอัพโหลด โปรแกรม โดยในที่นี้จะเลือกใช้ตัวอย่าง โค้ด (Example) ที่ไลบรารี่ให้มา ซึ่งเป็น โค้ดโปรแกรมที่ใช้ทดสอบการทำงาน ของโมดูล 7 Segment แบบ 4 หลัก ที่ใช้ ไอซีเบอร์ TM1637 เป็นตัวควบคุม ซึ่งมี รายละเอียดดังนี้

เริ่มต้นให้เราไปที่เมนู File >Examples TM1637 คลิกเลือก TM1637Test



ตัวอย่างโค้ดโปรแกรมจะถูกเปิดขึ้นมา ดังรูป



```
#include <Arduino.h>                    //เรียกใช้งานไลบรารี่หลักสำหรับบอร์ด Arduino
#include <TM1637Display.h>              //เรียกใช้ไลบรารี่สำหรับไอซี TM1637 เพื่อควบคุมการแสดงผล
                                        บนโมดูล 7 Segment

#define CLK 18                          //กำหนดให้ขา GPIO18/D18 เป็นขา CLK
#define DIO 5                           //กำหนดให้ขา GPIO5/D5 เป็นขา DIO
#define TEST_DELAY 2000                 //กำหนดให้เก็บค่าตัวเลข 2000 ไว้ในตัวแปร TEST_DELAY
                                        เพื่อเอาไว้ใช้เป็นค่าหน่วงเวลาในแต่ละตัวอย่างการทดสอบ
//ประกาศตัวแปร SEG_DONE เป็นค่าคงที่จำนวนเต็มขนาด 8 บิต ที่ใช้เก็บข้อมูลในแต่ละเซ็กเมนต์ของการ
แสดงผลในแต่ละหลักของโมดูล 7 Segment
const uint8_t SEG_DONE[] = {
  SEG_B | SEG_C | SEG_D | SEG_E | SEG_G,           // d
  SEG_A | SEG_B | SEG_C | SEG_D | SEG_E | SEG_F,   // O
  SEG_C | SEG_E | SEG_G,                           // n
  SEG_A | SEG_D | SEG_E | SEG_F | SEG_G            // E
  };
TM1637Display display(CLK, DIO);        //เริ่มต้นใช้งานไลบรารี่ TM1637Display

void setup() {}

void loop() {
  int k;                                //ประกาศตัวแปร k ชนิด int ใช้เก็บข้อมูลตัวเลข
                                        จำนวนเต็มที่ใช้เป็นตัวอย่างในการทดสอบ
  uint8_t data[] = {0xff, 0xff, 0xff, 0xff}; //ประกาศตัวแปร data ที่ใช้เก็บข้อมูลที่ทำให้
                                                หลอดไฟทุกดวงของทั้ง 4 หลัก "ติดสว่าง"
  uint8_t blank[] = {0x00, 0x00, 0x00, 0x00};  //ประกาศตัวแปร blank ที่ใช้เก็บข้อมูลที่
                                                ทำให้หลอดไฟทุกดวงของทั้ง 4 หลัก "ดับ"
  display.setBrightness(0x0f);          //กำหนดค่าความสว่างสูงสุดที่ 15 ให้กับจอแสดงผล (ต่ำสุด 0)
 //All segments on
 display.setSegments(data);             //แสดงข้อมูลออกทาง 7 Segment ผลลัพธ์คือ 88:88
 delay(TEST_DELAY);                     //หน่วงรอเป็นเวลา 2 วินาที

 //Selectively set different digits
 data[0] = display.encodeDigit(2);      //เก็บข้อมูลตัวเลข 2 ไว้ที่หลัก 0 ของตัวแปร data
data[1] = display.encodeDigit(9);       //เก็บข้อมูลตัวเลข 9 ไว้ที่หลัก 1 ของตัวแปร data
 data[2] = display.encodeDigit(5);      //เก็บข้อมูลตัวเลข 5 ไว้ที่หลัก 2 ของตัวแปร data
 data[3] = display.encodeDigit(3);      //เก็บข้อมูลตัวเลข 3 ไว้ที่หลัก 3 ของตัวแปร data
 display.setSegments(data);             //แสดงข้อมูลออกทาง 7 Segment ผลลัพธ์คือ 2953
 delay(TEST_DELAY);                     //หน่วงรอเป็นเวลา 2  วินาที
```

```
display.clear();                    //เคลียร์หน้าจอ
display.setSegments(data+2, 2, 2);  //แสดงข้อมูลในตัวแปร data ตั้งแต่หลักที่ 2 ขึ้นไป
                                    เป็นตัวเลข 2 หลัก โดยวางตำแหน่งเริ่มต้นไว้ที่หลักที่ 2
                                    (นับจากซ้ายสุดเป็นหลักที่ 0 ขวาสุดเป็นหลักที่ 3) ผลลัพธ์
                                    คือ __53
delay(TEST_DELAY);                  //หน่วงรอเป็นเวลา 2 วินาที

display.clear();                    //เคลียร์หน้าจอ
display.setSegments(data+2, 2, 1);  //แสดงข้อมูลในตัวแปร data ตั้งแต่หลักที่ 2 ขึ้นไป
                                    เป็นตัวเลข 2 หลัก โดยวางตำแหน่งเริ่มต้นไว้ที่หลักที่ 1
                                    ผลลัพธ์คือ _53_
delay(TEST_DELAY);                  //หน่วงรอเป็นเวลา 2 วินาที

display.clear();                    //เคลียร์หน้าจอ
display.setSegments(data+1, 3, 1);  //แสดงข้อมูลในตัวแปร data ตั้งแต่หลักที่ 1 ขึ้นไป
                                    เป็นตัวเลข 3 หลัก โดยวางตำแหน่งเริ่มต้นไว้ที่หลักที่ 1
                                    ผลลัพธ์คือ _953
delay(TEST_DELAY);                  //หน่วงรอเป็นเวลา 2 วินาที
//Show decimal numbers with/without leading zeros
display.showNumberDec(0, false);    //แสดงเลขจำนวนเต็ม 0 โดยไม่ต้องให้มี 0 นำหน้า (___0)
delay(TEST_DELAY);                  //หน่วงรอเป็นเวลา 2 วินาที
display.showNumberDec(0, true);     //แสดงเลขจำนวนเต็ม 0 โดยให้มี 0 นำหน้า (0000)
delay(TEST_DELAY);
display.showNumberDec(1, false);    //แสดงเลขจำนวนเต็ม 1 โดยไม่ต้องให้มี 0 นำหน้า (___1)
delay(TEST_DELAY);
display.showNumberDec(1, true);     //แสดงเลขจำนวนเต็ม 1 โดยให้มี 0 นำหน้า (0001)
delay(TEST_DELAY);
display.showNumberDec(301, false);  //แสดงเลขจำนวนเต็ม 301 โดยไม่ต้องให้มี 0 นำหน้า (_301)
delay(TEST_DELAY);
display.showNumberDec(301, true);       //แสดงเลขจำนวนเต็ม 301 โดยให้มี 0 นำหน้า (0301)
delay(TEST_DELAY);
display.clear();                        //เคลียร์หน้าจอ
display.showNumberDec(14, false, 2, 1); //แสดงตัวเลข 14 โดยไม่ต้องให้มี 0 นำหน้า
                                        เป็นตัวเลข 2 หลัก และวางตำแหน่งเริ่มต้นไว้ที่
                                        หลักที่ 1 ผลลัพธ์คือ _14_
delay(TEST_DELAY);
display.clear();                        //เคลียร์หน้าจอ
display.showNumberDec(4, true, 2, 2);   //แสดงตัวเลข 4 โดยให้มี 0 นำหน้า เป็นตัวเลข
                                        2 หลัก และวางตำแหน่งเริ่มต้นไว้ที่หลักที่ 2
                                        ผลลัพธ์คือ __04
```

```
delay(TEST_DELAY);
display.showNumberDec(-1, false);          //แสดงตัวเลข -1 โดยไม่ต้องให้มี 0 นำหน้า
                                           ผลลัพธ์คือ __-1
delay(TEST_DELAY);
display.showNumberDec(-12);                //แสดงตัวเลข -12 ผลลัพธ์คือ _-12
delay(TEST_DELAY);
display.showNumberDec(-999);               //แสดงตัวเลข -999 ผลลัพธ์คือ -999
delay(TEST_DELAY);
display.clear();
display.showNumberDec(-5, false, 3, 0);    //แสดงตัวเลข -5 โดยไม่ต้องให้มี 0 นำหน้า
                                           เป็นตัวเลข 3 หลัก และวางตำแหน่งเริ่มต้นไว้ที่
                                           หลักที่ 0 ผลลัพธ์คือ _-5_
delay(TEST_DELAY);
display.showNumberHexEx(0xf1af);           //แสดงตัวเลขฐาน 16 ผลลัพธ์คือ F1AF
delay(TEST_DELAY);
display.showNumberHexEx(0x2c);             //แสดงตัวเลขฐาน 16 ผลลัพธ์คือ __2C
delay(TEST_DELAY);
display.showNumberHexEx(0xd1, 0, true);    //แสดงตัวเลขฐาน 16 คือ d1 โดยให้มี 0 นำหน้า
                                           วางตำแหน่งเริ่มต้นไว้ที่หลักที่ 0 ผลลัพธ์คือ 00d1
delay(TEST_DELAY);
display.clear();
display.showNumberHexEx(0xd1, 0, true, 2); //แสดงตัวเลขฐาน 16 คือ d1 โดยให้มี 0 นำหน้า
                                           เป็นตัวเลข 2 หลัก และวางตำแหน่งเริ่มต้นไว้ที่
                                           หลักที่ 0 ผลลัพธ์คือ d1__
delay(TEST_DELAY);                         //หน่วงรอเป็นเวลา 2 วินาที
 //Run through all the dots
 for(k=0; k <= 4; k++) {                   //ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 4
   display.showNumberDecEx(0, (0x80 >> k), true);  //แสดงตัวเลข 0 โดยให้มี 0 นำหน้า
                                           เมื่อตัวแปร k เป็น 0 หลอดไฟที่ตำแหน่ง (:) จะติด
                                           (0x80) และพอเลื่อนบิตไปทางขวา หลอดไฟก็จะดับ
                                           ไปจนถึง 4 ผลลัพธ์คือ 00:00
   delay(TEST_DELAY);                      //หน่วงรอเป็นเวลา 2 วินาที
 }
 //Brightness Test
 for(k = 0; k < 4; k++)                    //ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 3
 data[k] = 0xff;                           //นำข้อมูลที่ทำให้หลอดไฟทุกดวงติดสว่างไปเก็บไว้ใน
                                           ตัวแปร data ของแต่ละหลัก
```

```
for(k = 0; k < 7; k++) {                //ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 6
  display.setBrightness(k);             //ปรับระดับความสว่างตามค่าของตัวเลขในตัวแปร k
  display.setSegments(data);            //แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลัก
                                        ผลลัพธ์คือ 88:88
  delay(TEST_DELAY);                    //หน่วงรอเป็นเวลา 2 วินาที
}
  //On/Off test
  for(k = 0; k < 4; k++) {              //ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 3
    display.setBrightness(7, false);    //หลอดไฟทุกดวงดับ หรือปิดหน้าจอ
    display.setSegments(data);          //แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลักผลลัพธ์คือ 88:88
    delay(TEST_DELAY);
    display.setBrightness(7, true);     //หลอดไฟทุกดวงติดสว่าง หรือเปิดหน้าจอ
    display.setSegments(data);          //แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลักผลลัพธ์คือ 88:88
    delay(TEST_DELAY);
  }

  //Done!
  display.setSegments(SEG_DONE);        //แสดงข้อมูลที่เก็บไว้ในตัวแปรออกทาง 7 Segment
                                        ผลลัพธ์คือ dOnE
  while(1);                             //วนลูปทำซ้ำไปเรื่อยๆ ตราบเท่าที่ผลลัพธ์ยังเป็นจริง
}
```

หลังจากอัพโหลดโค้ดโปรแกรมลงบนบอร์ด ถ้าทุกอย่างถูกต้อง บนโมดูลอุปกรณ์ 7 Segment จะแสดงผล เป็นตัวเลขและตัวอักษรต่างๆ โดยเริ่มจากหลอดไฟทุกตำแหน่งติดแสดงให้เห็นเป็นตัวเลข 8888 ก่อนจะแสดง ตัวอย่างอื่นตามลำดับวนไปเรื่อยๆ เช่น การแสดงชุดตัวเลขในตำแหน่งต่างๆ, การปรับเพิ่ม/ลดความสว่าง, การเปิด /ปิดเครื่องหมาย (;) และการเปิด/ปิดหน้าจอ เป็นต้น โดยแต่ละตัวอย่างจะใช้เวลาในการแสดงผลนาน 2 วินาที ก่อนจะสิ้นสุดด้วยการแสดงข้อความ done เป็นลำดับสุดท้าย แล้ววนกลับไปเริ่มต้นใหม่ ทั้งหมดนี้ก็เพื่อเป็นการ ทดสอบการทำงานให้กับโมดูลอุปกรณ์ 7 Segment นั่นเอง

บันทึกผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....