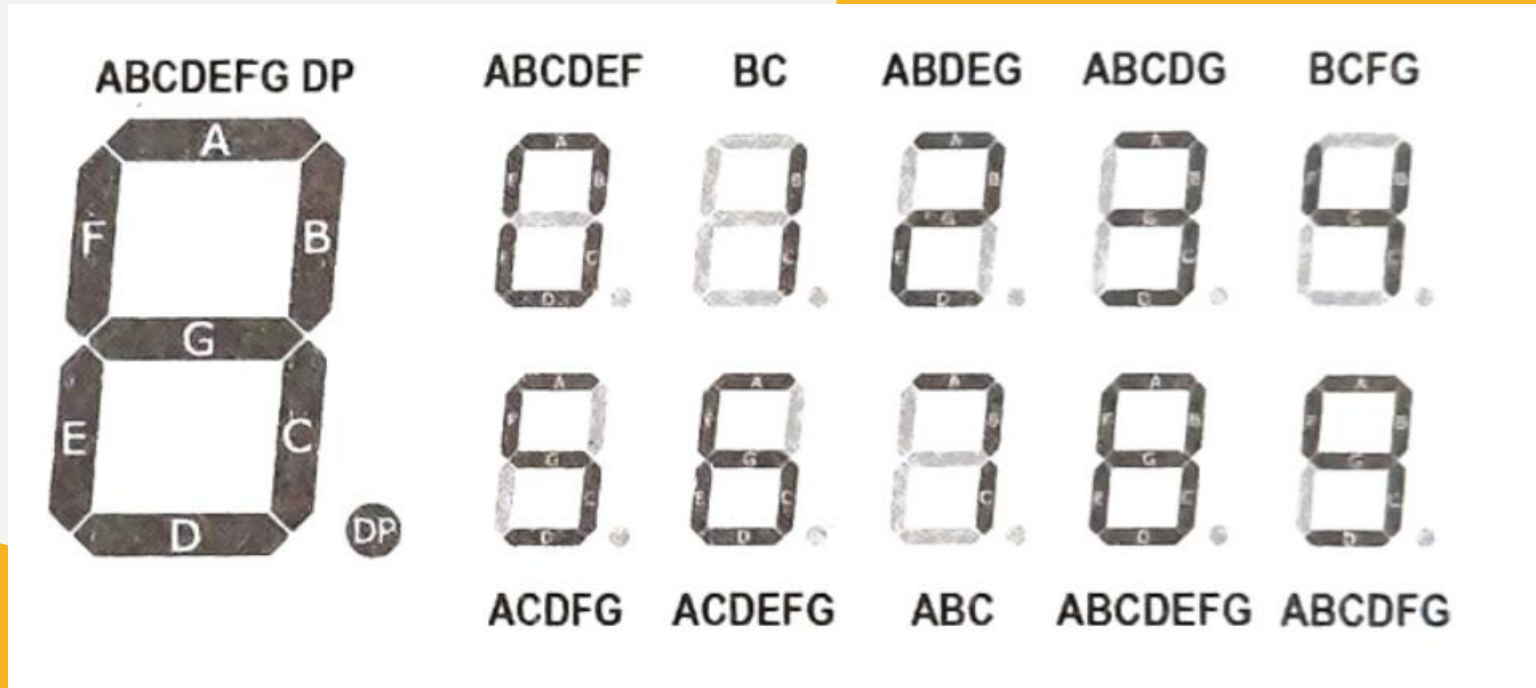
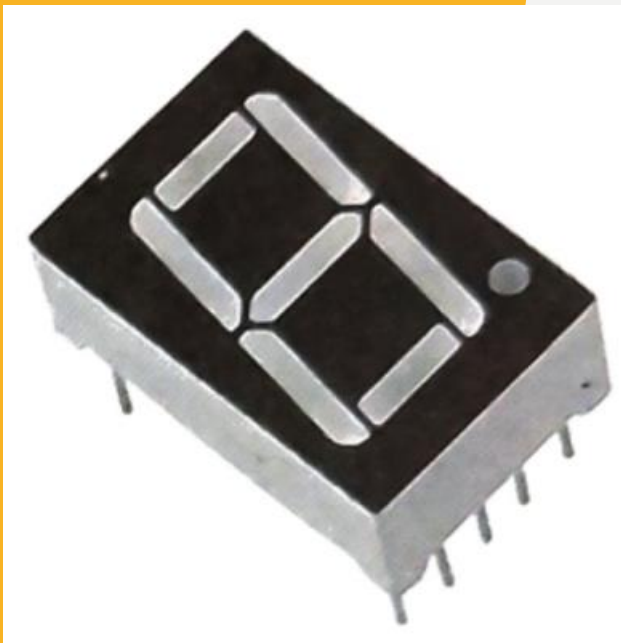


บทที่ 14

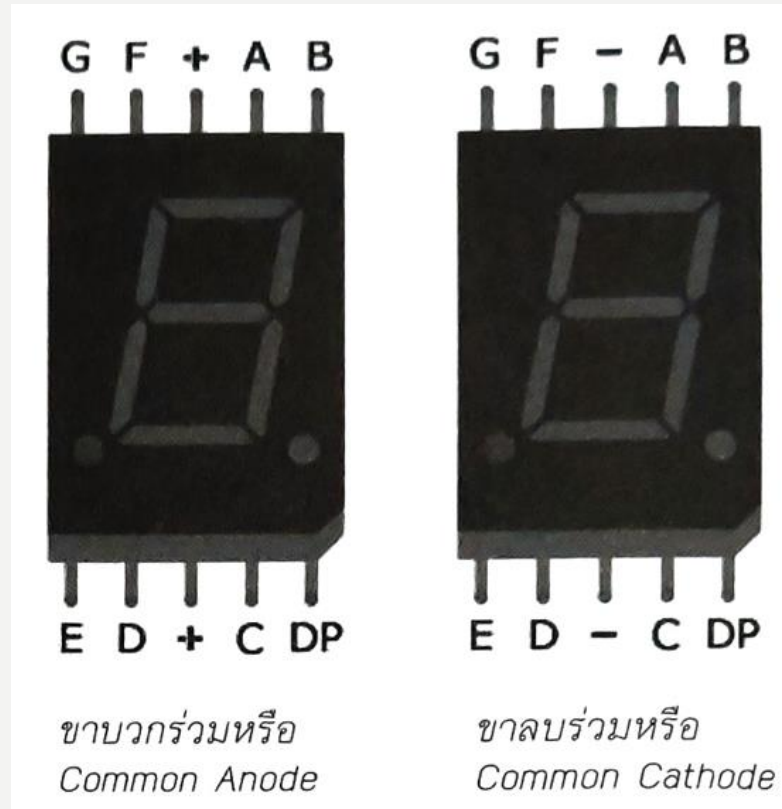
การแสดงผลบนอุปกรณ์ 7 Segment

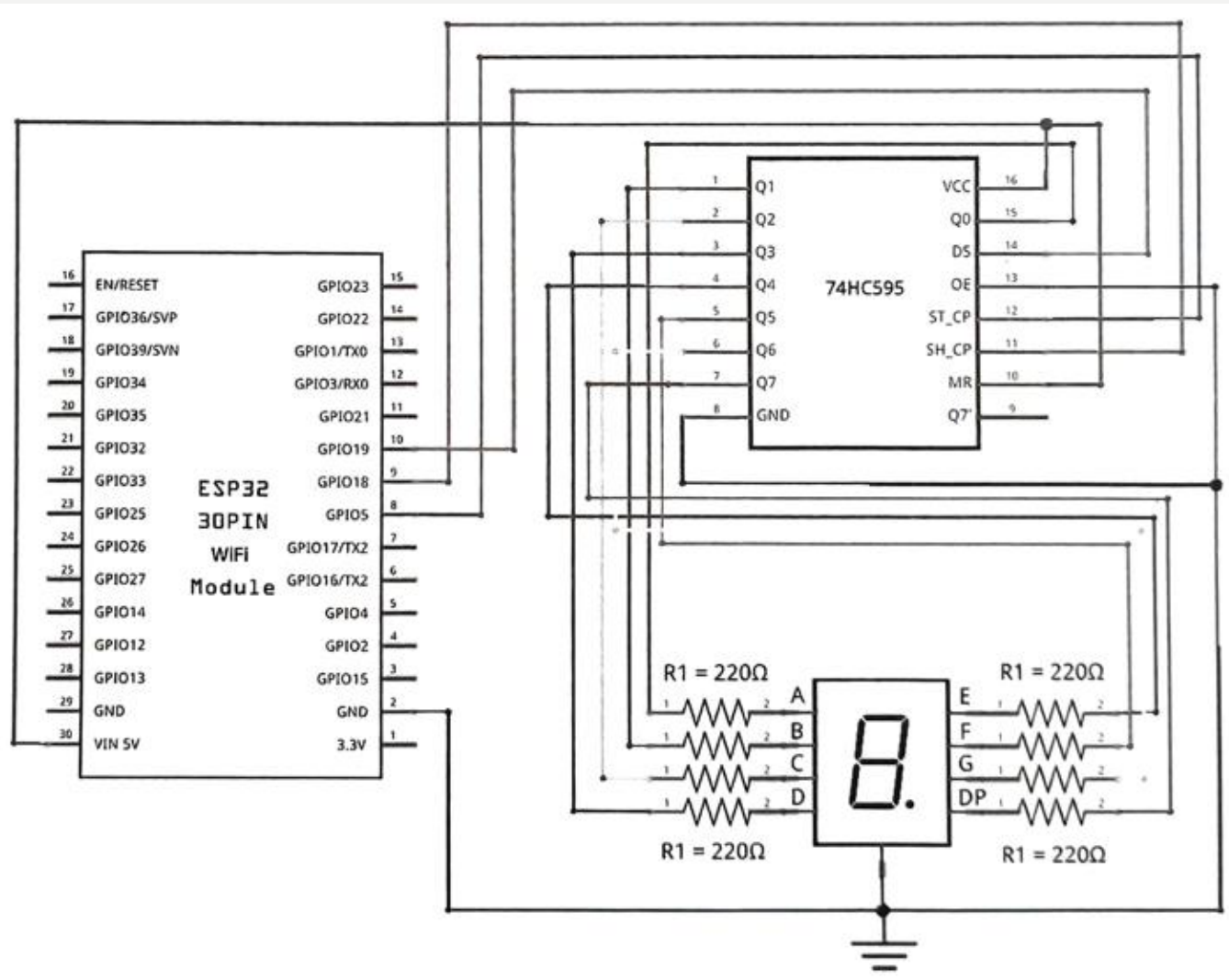


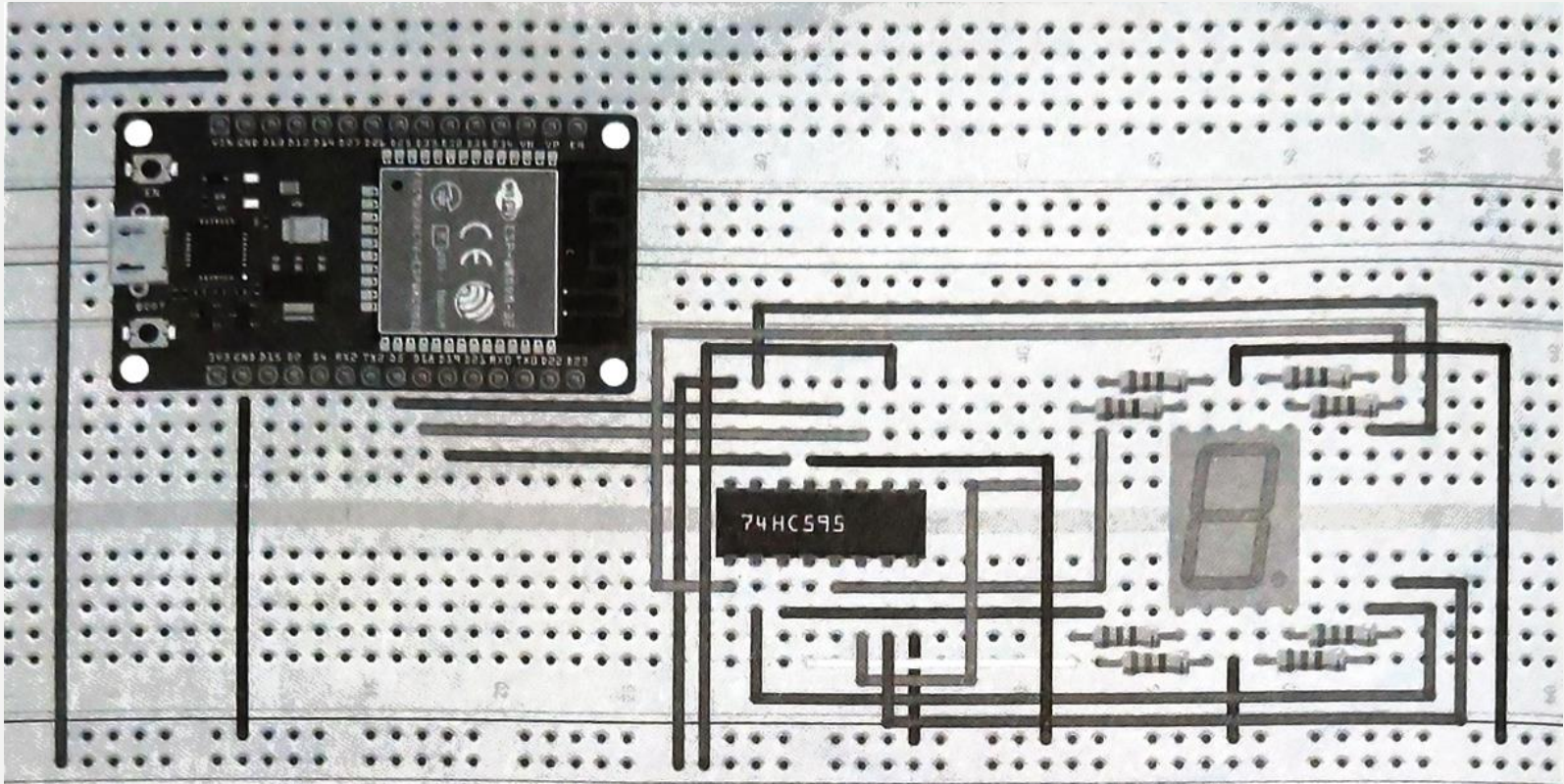
จุดประสงค์การเรียนรู้

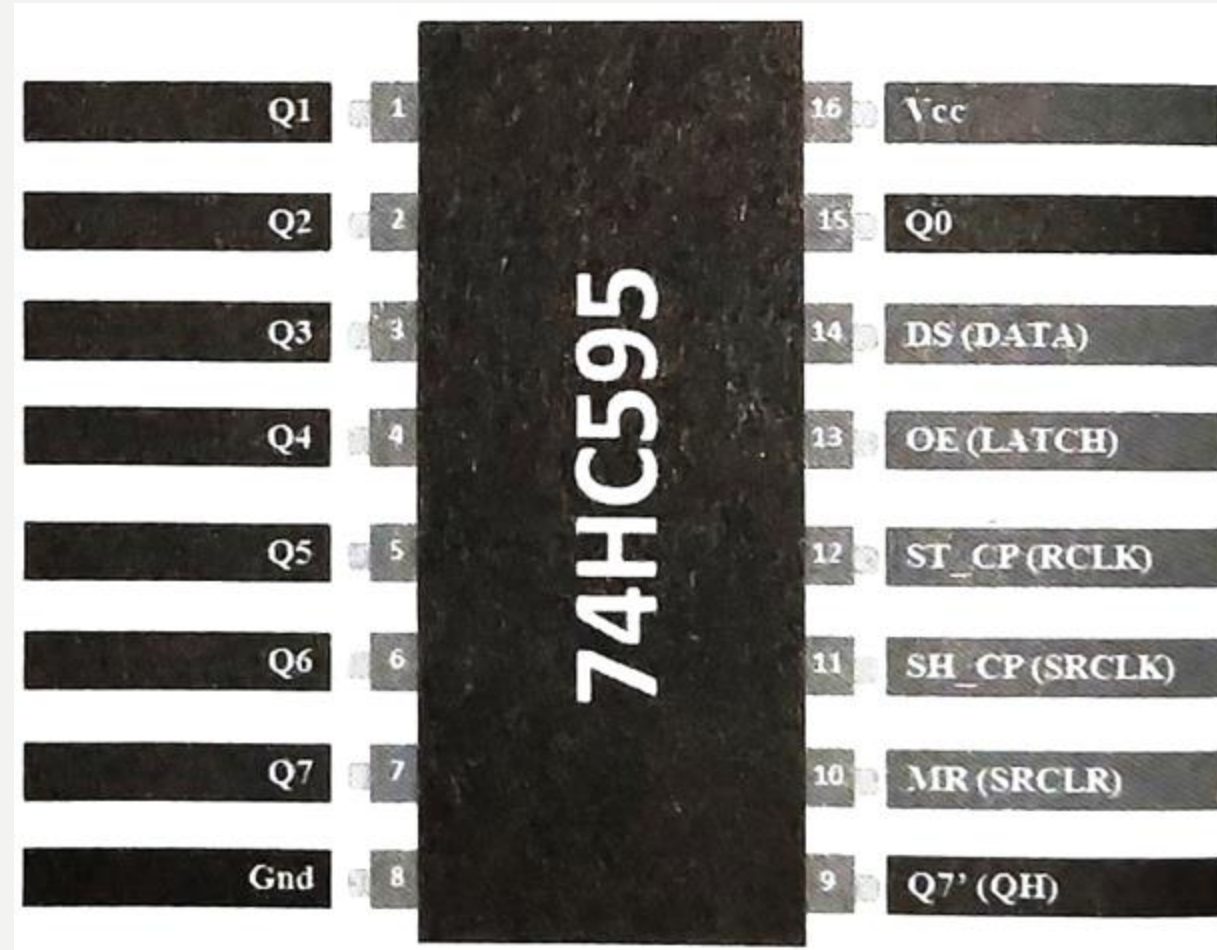
- 1.ศึกษาการนับตัวเลข 0-F โดยใช้ไอซี 74HC595
- 2.ศึกษาการแสดงตัวเลข 4 หลักโดยใช้โมดูลไอซี Tm1637

การทดลองที่ 1 การนับตัวเลข 0-F โดยใช้ไอซี 74HC595

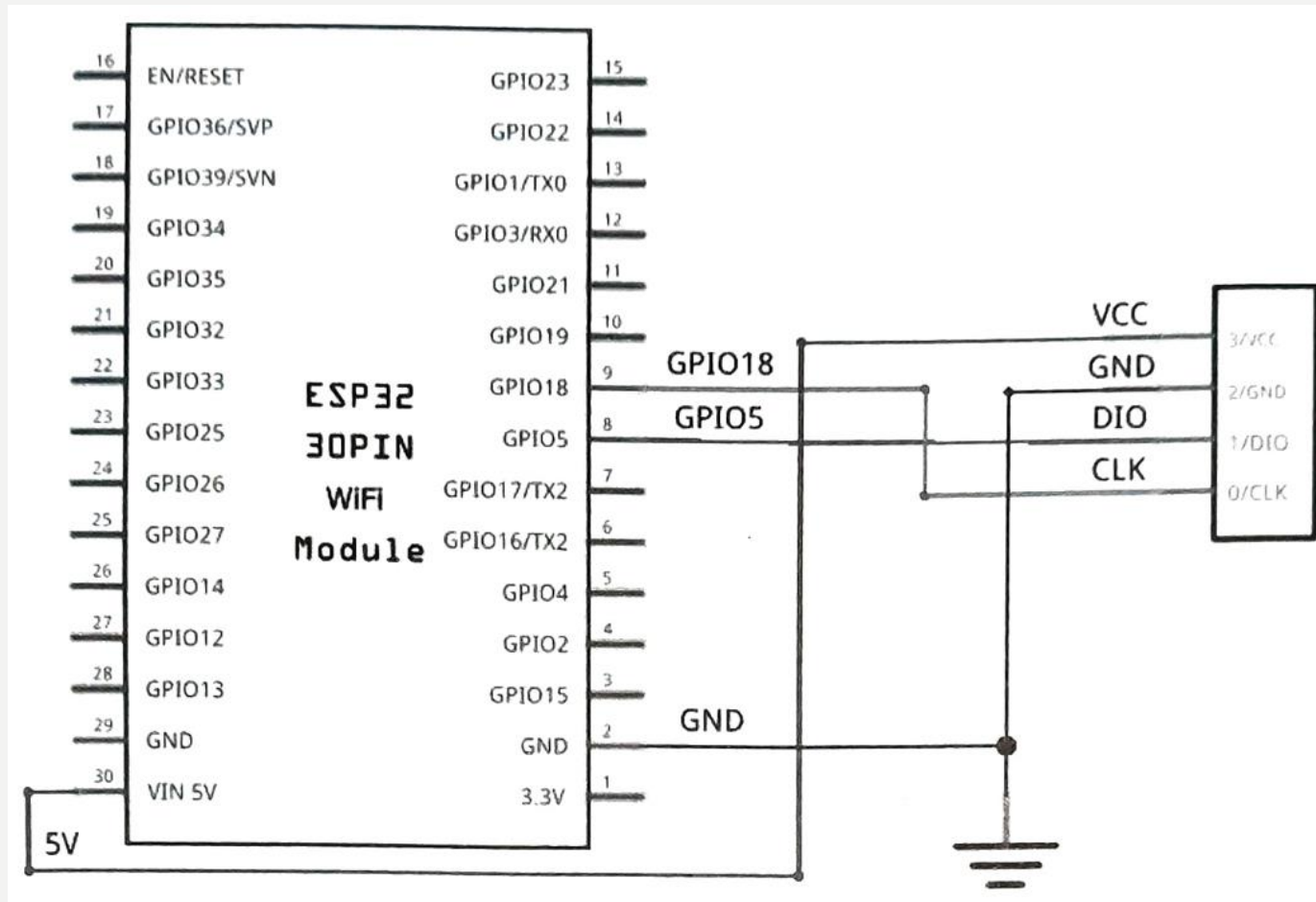


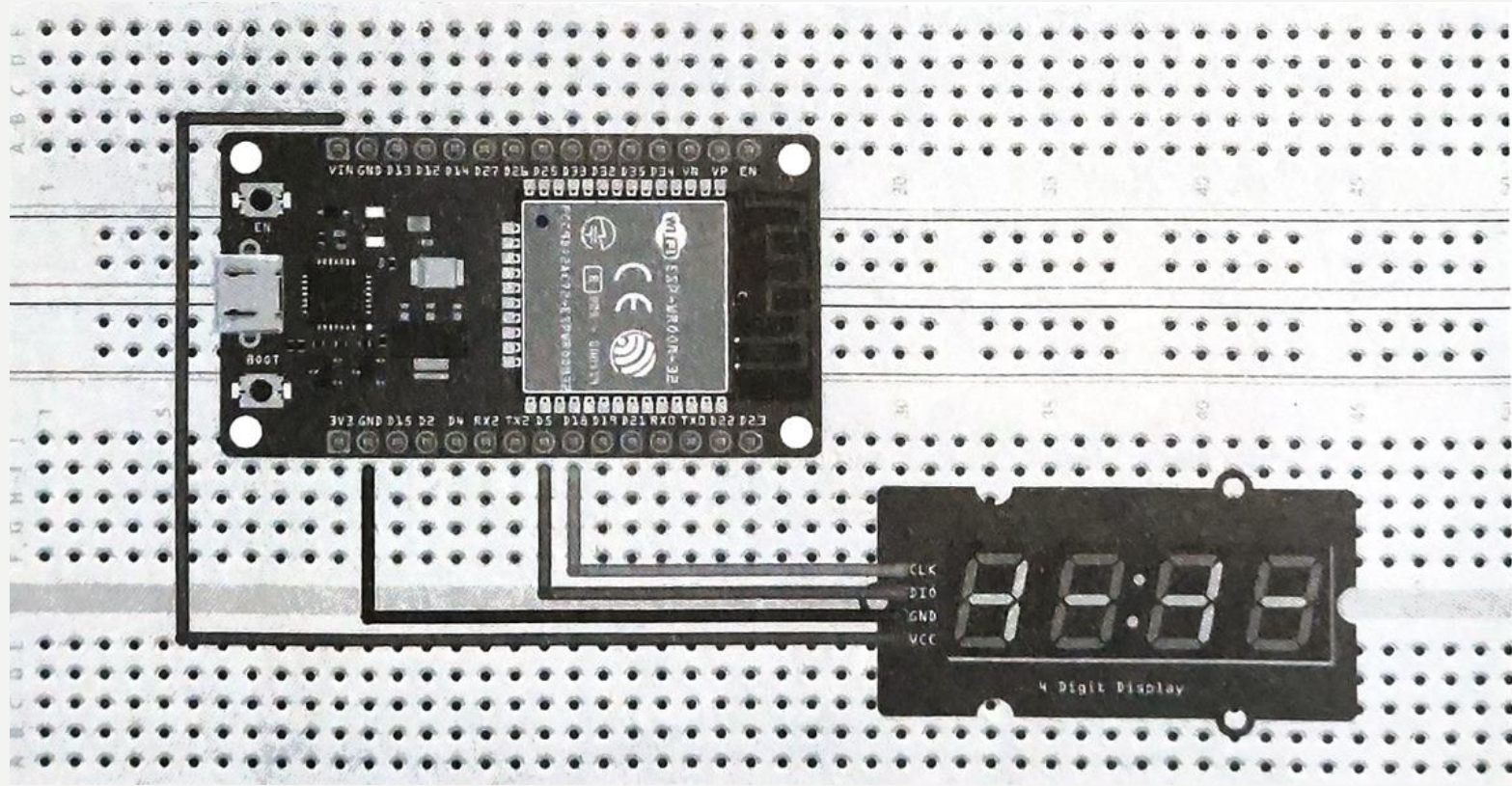






การทดลองที่ 2 การแสดงตัวเลข 4 หลักโดยใช้โมดูลไอซี Tm1637





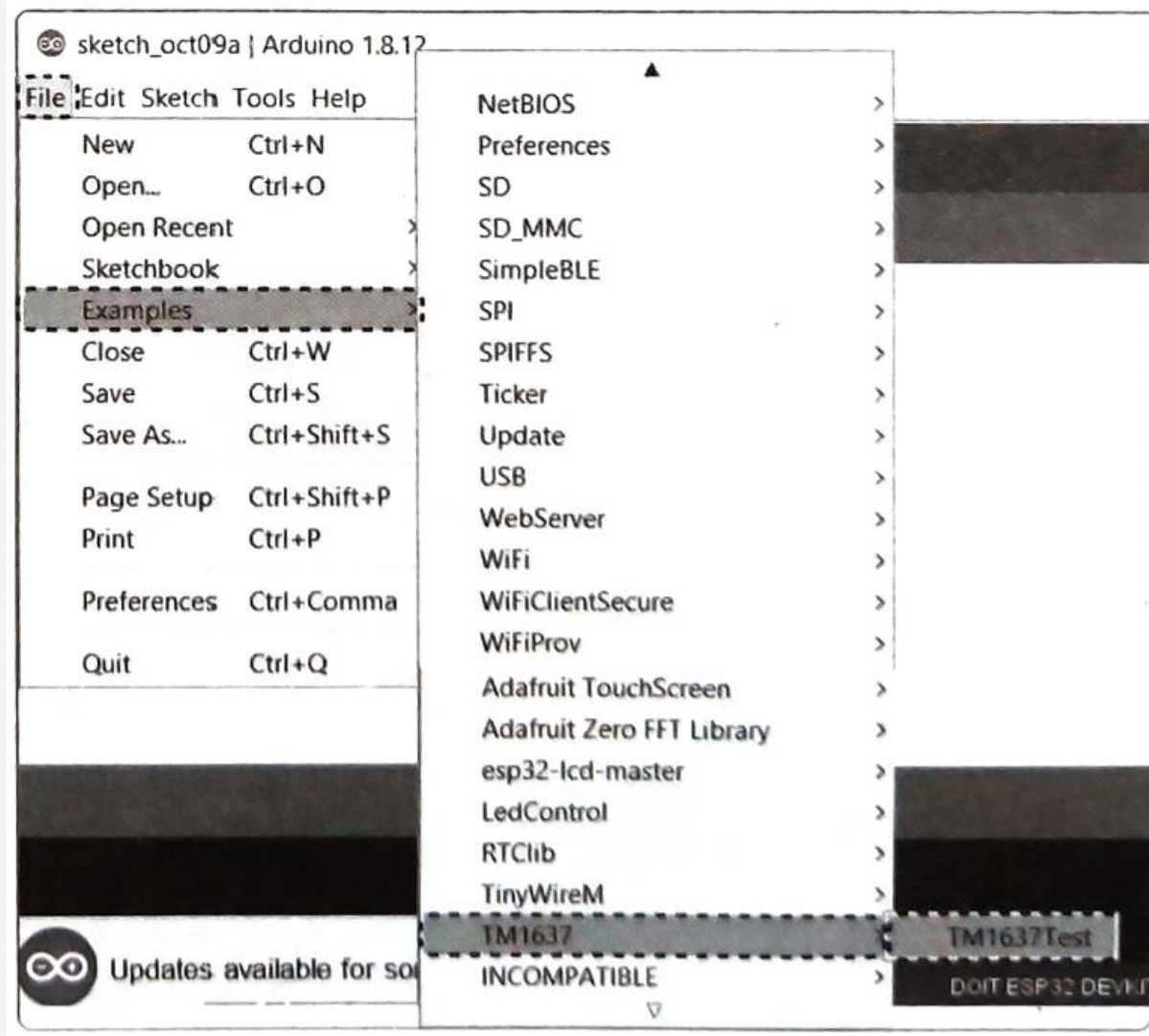
Library Manager

Type Topic **2**

SevenSegmentTM1637
by Bram Harmsen
Library for using a 4 digit seven segment display with TM1636 or TM1637 driver IC Extensive library for controlling a 4 digit seven segment display. This library inherits the Print class and uses the LCDAPI 1.0. For example you can use all normal Print methods like: print() and println(). From the LCDAPI among others begin(), clear(), home(), setCursor() and setBacklight() are implemented. On top of these regular functionality a separate fun class which adds more features can be used. For example a bombtimer(), nightrider() and bouncingBall() method can be used when using the fun class.
[More info](#)

TM1637
by Avishay Orpaz
Driver for 4 digit 7-segment display modules, based on the TM1637 chip. These chips can be found in cheap display modules. They communicate with the processor in I2C-like protocol. The implementation is pure software emulation and doesn't make use of any special hardware (other than GPIO pins). It is assumed that pull-up resistors are present (usually integrated in the display module).
[More info](#) **3**

TM1637 Driver
by AKJ
Feature-full and simple TM1637 library with nonblocking animation support Features: -Display integers, float, string with only one function. -Nonblocking animation: Blink, scrollLeft, fadeIn, fadeOut -Customizable offset, pad, overflow. -Custom raw value display. -Tunable brightness. -Screen clearing and splitting. Visit <https://github.com/AKJ7/TM1637> for more information.
[More info](#)



TM1637Test

```
1 #include <Arduino.h>
2 #include <TM1637Display.h>
3
4 // Module connection pins (Digital Pins)
5 #define CLK 2
6 #define DIO 3
7
8 // The amount of time (in milliseconds) between tests
9 #define TEST_DELAY 2000
10
11 const uint8_t SEG_DONE[] = (
12   SEG_B | SEG_C | SEG_D | SEG_E | SEG_G,      // d
13   SEG_A | SEG_B | SEG_C | SEG_D | SEG_E | SEG_F, // o
14   SEG_C | SEG_E | SEG_G,                      // n
15   SEG_A | SEG_D | SEG_E | SEG_F | SEG_G      // E
16 );
```

```

#include <Arduino.h> //เรียกใช้งานไลบรารีหลักสำหรับบอร์ด Arduino
#include <TM1637Display.h> //เรียกใช้ไลบรารีสำหรับไอซี TM1637 เพื่อควบคุมการแสดงผลบนโมดูล 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 เพื่อเอาไว้ใช้เป็นค่าหน่วยเวลาในแต่ละตัวอย่างการทดสอบ

//ประกาศตัวแปร SEG_DONE เป็นค่าคงที่จำนวนเต็มขนาด 8 บิต ที่ใช้เก็บข้อมูลในแต่ละเซ็กเมนต์ของการแสดงผลในแต่ละหลักของโมดูล 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); //เริ่มต้นใช้งานไลบรารี 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 ที่ใช้เก็บข้อมูลที่ทำให้หลอดไฟทุกดวงของทั้ง 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();
display.showNumberDec(4, true, 2, 2);

delay(TEST_DELAY);
display.showNumberDec(-1, false);

delay(TEST_DELAY);
display.showNumberDec(-12);
delay(TEST_DELAY);
display.showNumberDec(-999);
```

//เคลียร์หน้าจอ
//แสดงตัวเลข 4 โดยให้มี 0 นำหน้า เป็นตัวเลข
2 หลัก และวางตำแหน่งเริ่มต้นไว้ที่หลักที่ 2
ผลลัพธ์คือ __04

//แสดงตัวเลข -1 โดยไม่ต้องให้มี 0 นำหน้า
ผลลัพธ์คือ __-1

//แสดงตัวเลข -12 ผลลัพธ์คือ _-12

//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++) {
```

```
    display.showNumberDecEx(0, (0x80 >> k), true);
```

```
    delay(TEST_DELAY);
```

```
}
```

//ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 4

//แสดงตัวเลข 0 โดยให้มี 0 นำหน้า
เมื่อตัวแปร k เป็น 0 หลอดไฟที่ตำแหน่ง (:) จะติด
(0x80) และพอเลื่อนบิตไปทางขวา หลอดไฟก็จะดับ
ไปจนถึง 4 ผลลัพธ์คือ 00:00

//หน่วงรอเป็นเวลา 2 วินาที

```
//Brightness Test
```

```
for(k = 0; k < 4; k++)
```

```
data[k] = 0xff;
```

//ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 3

//นำข้อมูลที่ทำให้หลอดไฟทุกดวงติดสว่างไปเก็บไว้ใน
ตัวแปร data ของแต่ละหลัก

```
for(k = 0; k < 7; k++) {  
    display.setBrightness(k);  
    display.setSegments(data);  
  
    delay(TEST_DELAY);  
}
```

```
//ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 6  
//ปรับระดับความสว่างตามค่าของตัวเลขในตัวแปร k  
//แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลัก  
ผลลัพธ์คือ 88:88  
  
//หน่วงรอเป็นเวลา 2 วินาที
```

```

//On/Off test
for(k = 0; k < 4; k++) {
    display.setBrightness(7, false);
    display.setSegments(data);
    delay(TEST_DELAY);
    display.setBrightness(7, true);
    display.setSegments(data);
    delay(TEST_DELAY);
}

//Done!
display.setSegments(SEG_DONE);

while(1);
}

```

//ใช้คำสั่งวนลูปไปเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 3
//หลอดไฟทุกดวงดับ หรือปิดหน้าจอ
//แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลักผลลัพธ์คือ 88:88

//หลอดไฟทุกดวงติดสว่าง หรือเปิดหน้าจอ
//แสดงค่าตัวเลขจากตัวแปร data ทั้ง 4 หลักผลลัพธ์คือ 88:88

//แสดงข้อมูลที่เก็บไว้ในตัวแปรออกทาง 7 Segment
ผลลัพธ์คือ dOnE

//วนลูปทำซ้ำไปเรื่อยๆ トラバเท่าที่ผลลัพธ์ยังเป็นจริง

