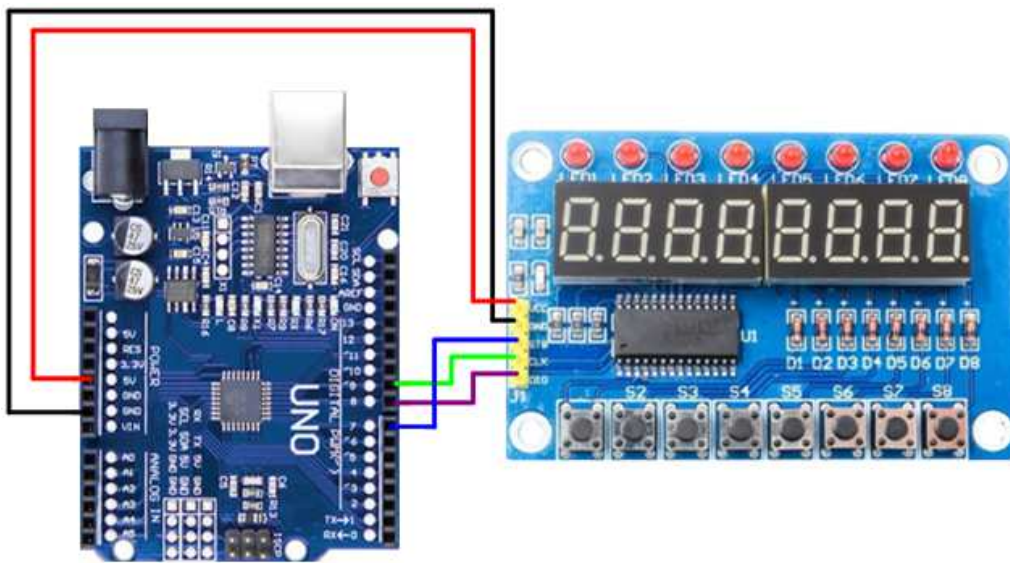


Lab 11

TM1638 Module

การทดลองที่ 11.1

ต่อวงจรตามรูปที่ 11.1 จากนั้นเชื่อมต่อบอร์ด Arduino UNO เข้ากับเครื่องคอมพิวเตอร์ เปิดไฟล์ TM1638_button_w_Arduino ทำการคอมไพล์และอัปโหลดโปรแกรมลงบอร์ด จากนั้นทดลองกดแต่ละปุ่มบนสวิตช์ และ สังเกตการเปลี่ยนแปลงของแอลอีดีบนบอร์ด TM1638



รูปที่ 11.1 วงจรเชื่อมต่อระหว่างบอร์ด UNO กับ TM1638

```
const int strobe = 7;
const int clock = 9;
const int data = 8;
void setup()
{
  pinMode(strobe, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(data, OUTPUT);
  sendCommand(0x8f); // activate
  reset();
}
void loop()
{
  uint8_t buttons = readTM1638_Buttons();
  for(uint8_t position = 0; position < 8; position++)
  {
    uint8_t mask = 0x1 << position;
    setTM1638LED(buttons & mask ? 1 : 0, position);
  }
}
```

รูปที่ 11.2 ตัวอย่างบางส่วนของโค้ดของการทดลองที่ 11.1 ในไฟล์ TM1638_button_w_Arduino

การทดลองที่ 11.2

โหลดไฟล์ TM1638_demo_sw_led_and_7seg จาก LMS จากนั้นนำมาเปิดด้วย Arduino IDE ทำการคอมไพล์และอัปโหลดโปรแกรมลงบนบอร์ด Arduino UNO สังเกตการทำงานของวงจร จากนั้นทดลอง กดแต่ละปุ่มบนสวิตช์ และ สังเกตการเปลี่ยนแปลงของแอลอีดีบนบอร์ด TM1638

```
const int strobe = 7;
const int clock = 9;
const int data = 8;

#define COUNTING_MODE 0
#define SCROLL_MODE 1
#define BUTTON_MODE 2

void setup()
{
    pinMode(strobe, OUTPUT);
    pinMode(clock, OUTPUT);
    pinMode(data, OUTPUT);

    sendCommand(0x8f); // activate
    reset();
}

void loop()
{
    static uint8_t mode = COUNTING_MODE;

    switch(mode)
    {
        case COUNTING_MODE:
            mode += counting();
            break;
        case SCROLL_MODE:
            mode += scroll();
            break;
        case BUTTON_MODE:
            buttons();
            break;
    }

    delay(200);
}
```

รูปที่ 11.3 ตัวอย่างบางส่วนของโค้ดของการทดลองที่ 11.2

การทดลองที่ 11.3

โหลดไฟล์ TM1638_display_LED จาก LMS จากนั้นนำมาเปิดด้วย Arduino IDE ทำการคอมไพล์และอัปโหลดโปรแกรมลงบนบอร์ด Arduino UNO สังเกตการทำงานของวงจรและการเปลี่ยนแปลงของแอลอีดีบนบอร์ด TM1638

```
void setup()
{
  pinMode(strobe, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(data, OUTPUT);

  sendCommand(0x8f); // activate 7segments and set maximum brightness
  reset();
  display_Hello();
}

void display_Hello()
{
  sendCommand(0x44); // set single address

  digitalWrite(strobe, LOW);
  shiftOut(data, clock, LSBFIRST, 0xc0); // leftmost 7-segment digit
  shiftOut(data, clock, LSBFIRST, 0x76); // character to be displayed ='H'
  digitalWrite(strobe, HIGH);

  digitalWrite(strobe, LOW);
  shiftOut(data, clock, LSBFIRST, 0xc2); // 2rd 7-segment
  shiftOut(data, clock, LSBFIRST, 0x79); // character to be displayed ='E'
  digitalWrite(strobe, HIGH);

  digitalWrite(strobe, LOW);
  shiftOut(data, clock, LSBFIRST, 0xc4); // 3rd 7-segment
  shiftOut(data, clock, LSBFIRST, 0x38); // character to be displayed ='L'
  digitalWrite(strobe, HIGH);

  digitalWrite(strobe, LOW);
  shiftOut(data, clock, LSBFIRST, 0xc6); // 4th 7-segment
  shiftOut(data, clock, LSBFIRST, 0x38); // character to be displayed ='L'
  digitalWrite(strobe, HIGH);

  digitalWrite(strobe, LOW);
  shiftOut(data, clock, LSBFIRST, 0xc8); // 5th 7-segment
  shiftOut(data, clock, LSBFIRST, 0x3f); // character to be displayed ='O'
  digitalWrite(strobe, HIGH);
}
```

รูปที่ 11.4 ตัวอย่างบางส่วนของโค้ดของการทดลองที่ 11.3

การคิดคะแนน

ให้ทำทุก Checkpoint คือ Checkpoint 11.1-Checkpoint 11.4 ส่งภายในเวลาที่ลงปฏิบัติการเท่านั้น ไม่มีแล็บชด โดยให้เรียก TA หรือเจ้าหน้าที่ห้องฮาร์ดแวร์ หรือ อาจารย์ผู้สอน ในส่วนของ Checkpoint 11.5 หากใครทำไม่ทันก็ไม่ใช่ไร แต่ถ้าใครทำทันและส่งตรวจก็จะมีคะแนนเสริมให้

Checkpoint 11.1

ให้อ่วงจรตามรูป 11.1 จากนั้นเขียนโปรแกรมบน Arduino สร้างวงจรรนาฬิกา 24 ชม. อย่างง่าย โดยใช้ฟังก์ชันที่มีให้ในการทดลองที่ 11.1-11.3 มาประยุกต์ใช้งาน โดยกำหนดให้มีรูปแบบการแสดงผล คือ 00:00:00 – 23:59:59 (ชม:นาที่:วินาที) โดยให้นาฬิกาแสดงอัปเดตการแสดงผลทุก ๆ วินาที กำหนดให้ใช้ฟังก์ชัน delay ในการหน่วงเวลา และแสดงผลโดยมีเครื่องหมาย - คั่นระหว่างตัวเลขหลักชั่วโมงกับหลักนาที และมีการคั่นด้วย - ระหว่างตัวเลขหลักนาทีกับหลักวินาที ดังนี้

12-23-45 (สิบสองนาฬิกา ยี่สิบสามนาที สิบห้าวินาที)

Checkpoint 11.2

ให้เขียนโปรแกรมเพื่อสร้างนาฬิกาดิจิทัล 24 ชั่วโมงเช่นเดียวกันกับใน Checkpoint 11.1 เพียงแต่เปลี่ยนรูปแบบของการจับเวลามาใช้การขัดจังหวะโดยใช้วงจร Timer/Counter แทน ห้ามใช้ฟังก์ชัน delay ในการหน่วงเวลา

Checkpoint 11.3

ให้เขียนโปรแกรมเพื่อรับการกดสวิตช์กดติดปล่อยดับ S1-S8 ซึ่งอยู่บนบอร์ด TM1638 กำหนดให้มีการตอบสนองการกดสวิตช์ดังนี้

- กด S1 แอลอีดีแสดงผลไฟวิ่งไปทางขวา
- กด S8 แอลอีดีแสดงผลไฟวิ่งไปทางซ้าย
- กด S2-S7 เพื่อสั่งให้ LED2-LED7 ติดค้าง

นักศึกษาสามารถดูตัวอย่างการทำงานของวงจรที่ต้องส่งได้จากลิงก์

<https://youtu.be/TTWh1T8uQKk>

Checkpoint 11.4

ให้นำนาฬิกาซึ่งสร้างโดยใช้วิธีการรับสัญญาณอินเทอร์รัพต์จากวงจร Timer/Counter ใน Checkpoint 11.2 มาเขียนโค้ดเพิ่มเติมเพื่อให้สามารถรับการกดปุ่มจาก S1-S8 โดยแต่ละปุ่มมีหน้าที่ดังนี้

- S1/S2 ทำหน้าที่ควบคุมการ เพิ่ม/ลด ค่าหลักชั่วโมง
- S4/S5 ทำหน้าที่ควบคุมการ เพิ่ม/ลด ค่าหลักนาที
- S7 ตั้งให้ค่าวินาที = 59
- S8 ตั้งให้วินาที = 0

นักศึกษาสามารถดูตัวอย่างการทำงานได้ที่คลิป <https://youtu.be/LfueJEb8KYO>

Checkpoint 11.5

ให้นำนาฬิกาที่สร้างใน Checkpoint 11.4 โดยเพิ่มฟังก์ชันอีกอย่างเข้ามา คือ กำหนดให้มีการกระพริบติดดับของเครื่องหมาย - ทุก ๆ 1 วินาที (เครื่องหมาย - ติด นาน 1 วินาที และ ดับนาน 1 วินาที สลับกัน) และเมื่อมีการกดสวิทช์เลื่อนตัวเลขหลักใด ให้ตัวเลขหลักนั้นมีการติด 0.5 วินาที และดับ 0.5 วินาที สลับกันไป และเมื่อไม่มีการกดสวิทช์นานกว่า 3 วินาที ก็ให้ตัวเลขหลักนั้นกลับมาแสดงผลตามปกติ