

## Assignment7

### mini clock

- 1). Conceptual Design ทำเป็นนาฬิกาที่สามารถใช้ได้ในชีวิตประจำวัน สามารถตั้งเวลาได้ และเมื่อถึงเวลาที่ตั้งไว้ก็จะปลุกและไฟ LED จะติดและสามารถปรับความสว่างของไฟจากตัวต้านทานปรับค่าได้ ส่วนหน้าจอแสดงเวลาใช้เป็นหน้าจอ OLED
- 2). การใช้โดยย่อ มีปุ่มทั้ง 3 ปุ่ม (ไล่จากขวามาซ้าย) ปุ่มแรก เอาไว้ใช้เพิ่มเวลา หากไม่กดปุ่มอื่นๆก่อน ก็จะไม่ทำอะไรเกิดขึ้น ปุ่มที่ 2 คือปุ่มเปลี่ยนเวลา หากกดแล้วจะเริ่มเลือกเวลาในหน่วยนาที่ก่อน สามารถกดปุ่มที่ 1 เพื่อเพิ่มเวลาได้ หากกดอีกครั้งก็จะเปลี่ยนไปเลือกเวลาในหน่วยชั่วโมงแทนและกดอีกครั้งเป็นการ confirm เปลี่ยนเวลา และปุ่มสุดท้าย ปุ่มที่ 3 ใช้งานเหมือนกับปุ่มที่ 2 แต่เป็นการตั้งเวลาแทนและหากถึงเวลาที่ตั้งไว้จะมีเสียงดังและไฟ LED ติด

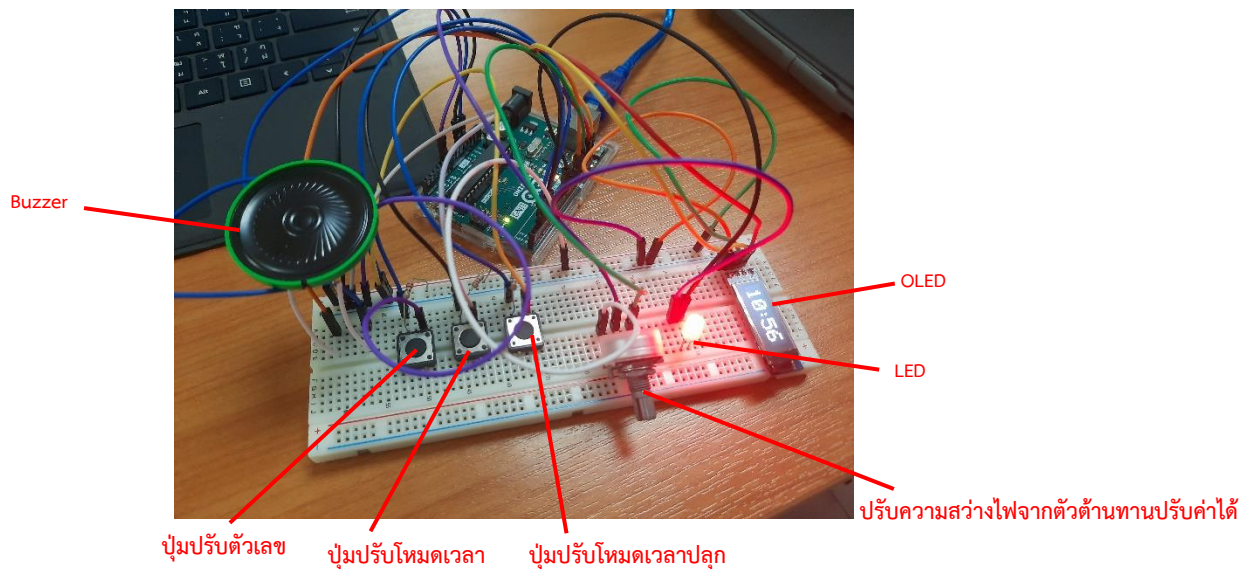
### 3). อธิบายโดยย่อใน

#### function หลัก

- function setup เป็นการกำหนด pin และ ตั้งค่าหน้าจอ display และแสดงเวลาเริ่มต้น
- function loop นับเวลาเก็บในตัวแปร, เช็คโหมด,เช็คการปลุกและเช็คการกดปุ่ม

#### function ย่อย

- function drawTimer แสดงตัวเลขของเวลาในรูปแบบ 00:00 ,
- function checkTime ถ้า seconds ถึง 60 ก็จะ reset second ,เพิ่ม minute ถ้า minute ถึง 60 ก็จะ reset minute, เพิ่ม hours และถ้า hours ครบ 24 ก็จะ reset hours
- function buttonTimerModeHandler เป็น function เปลี่ยน mode ของการเปลี่ยนเวลา
- function buttonSetHandler เป็น function เปลี่ยน mode ของการตั้งเวลา



## กลุ่ม1 หมอกจางๆและควัน

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
```

```
# define OLED_RESET 4
# define SCREEN_W 128
# define SCREEN_H 32
# define Buzzer 12
Adafruit_SSD1306 display(SCREEN_W, SCREEN_H, &Wire, OLED_RESET);

int hours = 0;
int minutes = 0;
int seconds = 50;
int Timer_hours;
int Timer_minutes;
const int Timer_seconds = 0;
bool Time_Stop = 1;

unsigned long currentMillis = 0;
int elapsedTimeUpdateMillis = 0;
unsigned long previousTimeUpdateMillis = 0;

const int PIN_BUTTON_TIMER_MODE = 4;
const int PIN_BUTTON_MODE = 3;
const int PIN_BUTTON_SET = 2;

unsigned long elapsedMusicMillis = 0;
unsigned long previousMusicMillis = 0;
struct music {
    int note;
    int Time;
};
typedef const struct music Test;
```

กำหนดตัวแปรต่างๆ

```
Test Music[56]={
    {NOTE_DS,125},
    {NOTE_CS5,125},
    {NOTE_B4,250},
    {NOTE_FS4,250},
    {0,125},
    {NOTE_FS4,125},
    {NOTE_FS4,125},
    {NOTE_FS4,125},
    {NOTE_FS4,125},
    {NOTE_FS4,125},

    {NOTE_B4,125},
    {NOTE_B4,125},
    {NOTE_B4,125},
    {NOTE_B4,125},
    {NOTE_A4,250},
    {NOTE_B4,125},
    {0,125},

    {NOTE_G4,125},
    {0,125},
    {0,125},
    {NOTE_G4,125},
    {0,125},
    {NOTE_G4,125},
    {NOTE_G4,125},
    {NOTE_G4,125},
    {NOTE_G4,125},
    {NOTE_G4,125},

    {NOTE_B4,125},
    {NOTE_B4,150},
    {NOTE_B4,150},
    {NOTE_B4,150},
    {NOTE_B4,150},
    {NOTE_CS5,250},
    {NOTE_DS,250},
    {0,125},
    {0,125},
    {NOTE_A4,125},
    {0,125},
```

array โน้ตและเวลา

กำหนดความถี่เสียง

```
int BUTTON_MODE_DEBOUNCE_TIME = 250;
int BUTTON_SET_DEBOUNCE_TIME = 175;

unsigned long elapsedButtonModeMillis = 0;
unsigned long previousButtonModeMillis = 0;

unsigned long elapsedButtonSetMillis = 0;
unsigned long previousButtonSetMillis = 0;

const int MODE_SHOW_TIME = 0;
const int MODE_SET_MINUTES = 2;
const int MODE_SET_HOURS = 1;
const int MODE_SET_TIMER_MINUTES = 4;
const int MODE_SET_TIMER_HOURS = 3;
byte currentMode = MODE_SHOW_TIME;
byte previousMode = 0;

int vr_pin = A0;
int led_pin = 9;
bool light = false;
```

กำหนดตัวแปรและโหมด

```
void setup()
{
  pinMode(vr_pin, INPUT);
  pinMode(led_pin, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  Serial.begin(9600);
  hours = EEPROM.read(0);
  minutes = EEPROM.read(1);
  Timer_hours = EEPROM.read(2);
  Timer_minutes = EEPROM.read(3);
  display.clearDisplay();
  drawTime();
  display.display();
}
int i = 1;
```

กำหนดpin ตั้งค่าหน้าจอ

```

void loop()
{
  display.fillScreen(BLACK);
  currentMillis = millis();
  checkTime();
  if (currentMode > 2)
  {
    increaseSeconds();
    checkTimer();
    drawTimer();
    if(currentMode == 3)
    {
      display.drawTriangle(22 , 0, 29, 3, 36, 0, WHITE);
    }
    else
    {
      display.drawTriangle(87 , 0, 94, 3, 101, 0, WHITE);
    }
  }
  else if (currentMode == MODE_SHOW_TIME)
  {
    increaseSeconds();
    drawTime();
    if(hours == Timer_hours && minutes == Timer_minutes && seconds == Timer_seconds)
    {
      Time_Stop = 0;
    }
    elapsedMusicMillis = currentMillis - previousMusicMillis;
    //Serial.println(seconds);
    if(Time_Stop == 0 && elapsedMusicMillis> Music[i-1].Time)
    {
      tone(Buzzer,Music[i].note,Music[i].Time);
      i++;
      previousMusicMillis = currentMillis;
      if(i > 56){
        i = 1;
      }
      light = true;
    }
    else if (Time_Stop == 1){
      noTone(12);
      i=1;
    }
  }
  else
  {
    previousTimeUpdateMillis = currentMillis;
    drawTime();
    if(currentMode == 1)
    {
      display.drawTriangle(22 , 0, 29, 3, 36, 0, WHITE);
    }
    else
    {
      display.drawTriangle(87 , 0, 94, 3, 101, 0, WHITE);
    }
  }

  if (digitalRead(PIN_BUTTON_TIMER_MODE) == LOW) {
    buttonTimerModeHandler();
    Time_Stop = 1;
    light = false;
    analogWrite(led_pin,0);
  }
  if (digitalRead(PIN_BUTTON_MODE) == LOW) {
    buttonModeHandler();
    Time_Stop = 1;
    light = false;
    analogWrite(led_pin,0);
  }
  if (digitalRead(PIN_BUTTON_SET) == LOW) {
    buttonSetHandler();
    Time_Stop = 1;
    light = false;
    analogWrite(led_pin,0);
  }
  if(light == true){
    int vr = analogRead(vr_pin);
    int vr1 = map(vr,0,1023,0,255);
    analogWrite(led_pin,vr1);
    //Serial.println(vr1);
  }
  display.display();
}

```

นับเวลาเก็บในตัวแปร, เช็คโหมด,เช็คการ  
ปลุกและเช็คการกดปุ่ม

```
void drawTimer()
{
  display.setTextSize(4);
  display.setTextColor(WHITE);
  if (Timer_hours<10){
    display.setCursor(9, 5);
    display.println("0");
    display.setCursor(39, 5);
    display.println(Timer_hours);
  }
  else{
    display.setCursor(9, 5);
    display.println(Timer_hours);
  }
  display.setCursor(54, 5);
  display.println(":");
  if (Timer_minutes<10){
    display.setCursor(74, 5);
    display.println("0");
    display.setCursor(104, 5);
    display.println(Timer_minutes);
  }
  else{
    display.setCursor(74, 5);
    display.println(Timer_minutes);
  }
}
```

วาดเวลาในหน้าจอเวลาปลูก

```
void drawTime()
{
  display.setTextSize(4);
  display.setTextColor(WHITE);
  if (hours<10){
    display.setCursor(9, 5);
    display.println("0");
    display.setCursor(39, 5);
    display.println(hours);
  }
  else{
    display.setCursor(9, 5);
    display.println(hours);
  }
  display.setCursor(54, 5);
  display.println(":");
  if (minutes<10){
    display.setCursor(74, 5);
    display.println("0");
    display.setCursor(104, 5);
    display.println(minutes);
  }
  else{
    display.setCursor(74, 5);
    display.println(minutes);
  }
}
```

วาดเวลาในหน้าจอเวลาปกติ

```

void checkTime() {
  if (seconds > 59)
  {
    seconds = 0;
    minutes++;
    //EEPROM.put(1, minutes);
  }
  if (minutes > 59)
  {
    minutes = 0;
    hours++;
    //EEPROM.put(0, hours);
  }
  if (hours > 23)
  {
    hours = 0;
  }
}

void checkTimer() {
  if (Timer_minutes > 59)
  {
    Timer_minutes = 0;
    Timer_hours++;
  }
  if (Timer_hours > 23)
  {
    Timer_hours = 0;
  }
}

void increaseSeconds()
{
  elapsedTimeUpdateMillis = currentMillis - previousTimeUpdateMillis;

  if (elapsedTimeUpdateMillis > 1000)
  {
    seconds++;
    previousTimeUpdateMillis = currentMillis - (elapsedTimeUpdateMillis - 1000);
  }
}

```

เช็คเวลาและเปลี่ยนนาฬิกาหรือชั่วโมงเมื่อครบ

เช็คเวลาในโหมดปลุกและเปลี่ยนนาฬิกาหรือชั่วโมงเมื่อครบ

เก็บค่าการนับวินาที

```

void buttonModeHandler()
{
    previousMode = 0;
    elapsedButtonModeMillis = currentMillis - previousButtonModeMillis;
    if (elapsedButtonModeMillis > BUTTON_MODE_DEBOUNCE_TIME)
    {
        previousButtonModeMillis = currentMillis;
        currentMode++;

        if (currentMode > 2)
        {
            currentMode = 0;
            //EEPROM.update(1, minutes);
            //EEPROM.update(0, hours);
        }
    }
}

void buttonTimerModeHandler()
{
    elapsedButtonModeMillis = currentMillis - previousButtonModeMillis;
    if (elapsedButtonModeMillis > BUTTON_MODE_DEBOUNCE_TIME)
    {
        previousButtonModeMillis = currentMillis;
        currentMode = 3;
        currentMode+=previousMode;
        previousMode++;

        if (currentMode > 4)
        {
            currentMode = 0;
            previousMode = 0;
            //EEPROM.update(3, Timer_minutes);
            //EEPROM.update(2, Timer_hours);
        }
    }
}

```

เลือกเปลี่ยนเวลานาที,ชั่วโมงของเวลาปกติ

เลือกเปลี่ยนเวลานาที,ชั่วโมงของเวลาในโหมดปลุก



```
void buttonSetHandler()
{
    elapsedButtonSetMillis = currentMillis - previousButtonSetMillis;
    if (elapsedButtonSetMillis > BUTTON_SET_DEBOUNCE_TIME)
    {
        previousButtonSetMillis = currentMillis;

        if (currentMode == MODE_SET_MINUTES)
        {
            minutes++;
        }
        if (currentMode == MODE_SET_HOURS)
        {
            hours++;
        }
        if (currentMode == MODE_SET_TIMER_MINUTES)
        {
            Timer_minutes++;
        }
        if (currentMode == MODE_SET_TIMER_HOURS)
        {
            Timer_hours++;
        }
    }
}
```

เปลี่ยนนาฬิกา, ชั่วโมงในโหมดที่เลือก