

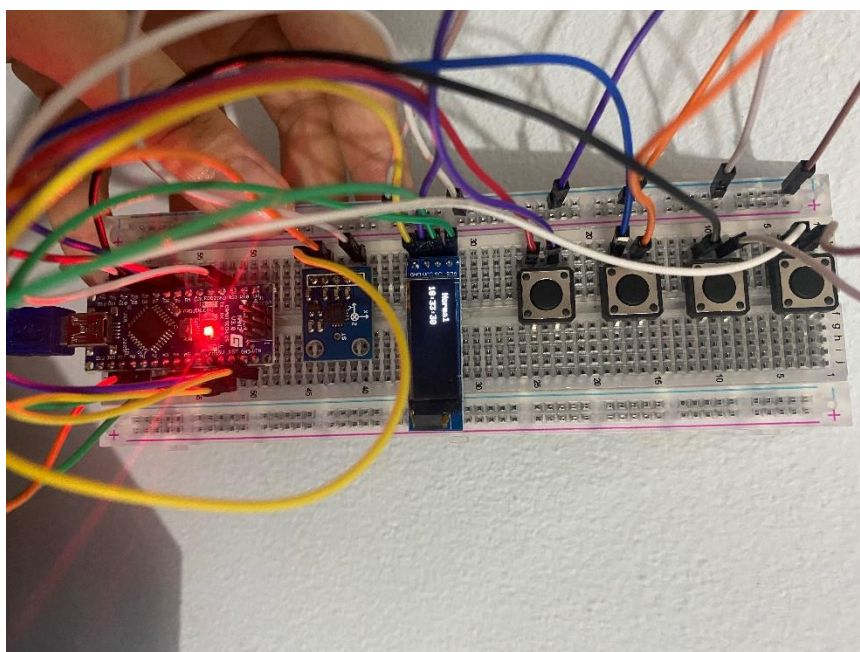
## Assignment #7

ชื่อกลุ่ม : ฟ้ายรักพ่อ

สมาชิกในกลุ่ม : 64010270 นายคณพัฒน์ โพธิ์น้อย

64010315 นายธนธร แดงอ่อน

### รูปถ่ายชิ้นงาน



แนวคิด : นาฬิกาของกลุ่มผมนั้นจะเป็นนาฬิกาที่มีการแจ้งเตือนและสามารถแสดงผลเวลาในโหมดปกติ (Normal) สามารถปรับเวลาของชั่วโมงและนาฬิกาในโหมดตั้งเวลา (Set Time) และตั้งปลุกของชั่วโมงและนาฬิกาในจากโหมดตั้งปลุก (Set Alarm) และการกลับด้านของหน้าจอโดยใช้อุปกรณ์ ADXL335

อุปกรณ์ที่ใช้ทำโครงงาน : 1. Arduino Nano

2. จอ OLED

3. ADXL335 Module

4. Tact Switch 12mm 4 ปุ่ม

5. สาย Jump M-M

การใช้งานโดยย่อ : Switch 4 ปุ่ม

- ปุ่ม 1 = Save ข้อมูลเวลาลง EEPROM
- ปุ่ม 2 = ปุ่มเปลี่ยนนาฬิกา ( กดเมื่ออยู่โหมดปกติ (Normal) )
- ปุ่ม 3 = ปุ่มเปลี่ยนชั่วโมง ( กดเมื่ออยู่โหมดปกติ (Normal) )
- ปุ่ม 4 = ปุ่มเปลี่ยนการทำงาน มี 3 โหมด คือ โหมดปกติ (Normal), โหมดตั้งเวลา (Set Time) และ โหมดตั้งปลุก (Set Alarm)

## โปรแกรมและอธิบายโดยย่อ (ระดับฟังก์ชัน)

```
#include <Adafruit_SSD1306.h>

#include <Adafruit_GFX.h>

#include <TimerOne.h>

#include <EEPROM.h>

#define OLED_RESET -1

Adafruit_SSD1306 OLED(OLED_RESET);

#define button_1 6 //Save data of time in EEPROM

#define button_2 7 //Set Minute

#define button_3 8 //Set Hour

#define button_4 9 //Change Mode

// โหมดปกติ (Normal)

#define clockTime 0 // Normal Mode

// โหมดตั้งเวลา (Set Time)

#define setClockTime 1 //Set Time Mode

// โหมดตั้งปลุก (Set Alarm)

#define setClockAlarm 2 //Set Alarm Mode

int8_t secondTime;

int8_t minuteTime;

int8_t hourTime;

int8_t minuteAlarm;
```

```
int8_t hourAlarm;
```

```
int mode = 0;
```

```
void updateTime()
```

```
{
```

```
// ฟังก์ชันของนาฬิกา เวลาจะเพิ่มทีละ 1 วินาที ถึง 60 วินาที เพิ่ม 1 นาทีและวินาที พร้อมกัน จะ  
เปลี่ยนเป็น 0 ถ้านาทีครบ 60 นาที เพิ่ม 1 ชั่วโมงแล้วครบ 24 ชั่วโมงก็จะกลับไปเป็น 0 เหมือนเดิม
```

```
secondTime++;
```

```
if (secondTime == 60)
```

```
{
```

```
minuteTime++;
```

```
secondTime = 0;
```

```
}
```

```
if (minuteTime == 60)
```

```
{
```

```
hourTime++;
```

```
minuteTime = 0;
```

```
}
```

```
if (hourTime == 24)
```

```
{
```

```
hourTime = 0;
```

```
}
```

```
}
```

```

void up_or_down() //Functions for ADXL335
{
// ฟังก์ชัน กลับบนลงล่าง กลับล่างขึ้นบนของจอ OLED โดยใช้ ADXL335

int directX = analogRead(A0);
if (directX > 400)
{
// Serial.println("RO");
OLED.setRotation(0);
}
if (directX < 300)
{
OLED.setRotation(2);
// Serial.println("RO2");
}
// Serial.println(directX);
}

int bounce; // Functions of bounce
bool Button(int pin)
// ฟังก์ชันbounce ของปุ่ม ถ้ากดแล้ว Return ค่า = true ถ้าไม่กด Return ค่า = false
{
if (!digitalRead(pin))
{
if (millis() - bounce > 500)
{
bounce = millis();

```

```
    return true;
}

    bounce = millis();
}

    return false;
}
```

```
void setup()
{
    Serial.begin(9600);
    OLED.begin(SSD1306_SWITCHCAPVCC, 0x3c);

    // Use Timer of every 1 sec

    Timer1.initialize();

    // Request updateTime every 1 sec follow Timer1.initialize();

    Timer1.attachInterrupt(updateTime);

    //Call time from Rom

    EEPROM.get(0, secondTime);
    EEPROM.get(1, minuteTime);
    EEPROM.get(2, hourTime);
    EEPROM.get(3, minuteAlarm);
    EEPROM.get(4, hourAlarm);


    pinMode(button_1, INPUT_PULLUP);
    pinMode(button_2, INPUT_PULLUP);
    pinMode(button_3, INPUT_PULLUP);
    pinMode(button_4, INPUT_PULLUP);
}
```

```
}
```

```
void loop()
```

```
{
```

```
  up_or_down();
```

```
  if (Button(button_1))
```

```
  {
```

```
    EEPROM.update(0, secondTime); //Write in EEPROM
```

```
    EEPROM.update(1, minuteTime);
```

```
    EEPROM.update(2, hourTime);
```

```
    EEPROM.update(3, minuteAlarm);
```

```
    EEPROM.update(4, hourAlarm);
```

```
  }
```

```
  if (Button(button_2))
```

```
  {
```

```
    if (mode == setClockTime)
```

```
    {
```

```
      minuteTime++; //Add time minute in set time mode
```

```
      minuteTime %= 60;
```

```
    }
```

```
    else if (mode == setClockAlarm)
```

```
    {
```

```
      minuteAlarm++; // Add time minute in set alarm mode
```

```
      minuteAlarm %= 60;
```

```
}  
}
```

```
if (Button(button_3))  
{  
  if (mode == setClockTime)  
  {  
    hourTime++; // Set Hour in Normal and Alarm  
    hourTime %= 24;  
  }  
  else if (mode == setClockAlarm)  
  {  
    hourAlarm++;  
    hourAlarm %= 24;  
  }  
}
```

```
if (Button(button_4))  
{  
  mode++; //Change mode 0 1 2 are normal, set time and set alarm  
  mode %= 3;  
}
```

```
OLED.clearDisplay();  
OLED.setCursor(10, 0);  
OLED.setTextColor(WHITE);
```



```
OLED.setTextSize(1.7);

if (mode == clockTime)
{
    OLED.println("Normal"); // In normal mode to print on this
    OLED.println("");
    OLED.print(hourTime);
    OLED.print(":");
    OLED.print(minuteTime);
    OLED.print(":");
    OLED.print(secondTime);
    // Serial.print("1");
}
else if (mode == setClockTime)
{
    OLED.println("Set Time"); // In Set time mode to print on this
    OLED.println("");
    OLED.print(hourTime);
    OLED.print(":");
    OLED.print(minuteTime);
    OLED.print(":");
    OLED.print(secondTime);
    // Serial.print("2");
}
else if (mode == setClockAlarm)
{
```

```
OLED.println("Set Alarm"); // In Set alarm mode to print on this
OLED.println("");
OLED.print(hourAlarm);
OLED.print(":");
OLED.print(minuteAlarm);
OLED.print(":");
OLED.print(secondTime);
// Serial.print("3");
}

if (minuteTime == minuteAlarm && hourTime == hourAlarm) // Set Alarm
{
    Serial.println("ALARM"); // Notification "ALARM!"
    OLED.clearDisplay();
    OLED.println("ALARM!");
}
OLED.display();
}
```