



## Final Project

เทอร์โมมิเตอร์แจ้งเตือนตามเกณฑ์อุณหภูมิโดยไฟ LED และเสียง

### จัดทำโดย

นาย สัณห์รัฐ พรหมจรรย์ 63070501069

นาย สิริวิชญ์ ศาสนกุล 63070501072

### เสนอ

ผศ. สุรพนธ์ ตั้มนาค

ดร. ประพงษ์ ปรีชาประพาหงส์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา

CPE328 Embedded Systems

ภาคเรียนที่ 2 ปีการศึกษา 2565

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

## ที่มาและความสำคัญ

ในช่วง 3-4 ปีที่ผ่านมา ผู้คนเริ่มให้ความสำคัญการควบคุมอุณหภูมิห้องที่ใช้ทำการชุดเหมือง Cryptocurrency มากขึ้น เนื่องจากการใช้งานเครื่องมือเหล่านั้น ต้องใช้พลังงานที่สูง หากไม่อยู่ในอุณหภูมิที่เย็นพอที่จะให้เครื่องจักรทำงานได้ จะทำให้เกิดความเสียหายตามมาได้ พวกเราจึงได้ลองคิดที่จะทำ Project เล็ก ๆ เกี่ยวกับเรื่องนี้ขึ้นมา ไม่ได้เจาะจงเฉพาะห้องชุดเหมือง แต่เป็นการตรวจจับอุณหภูมิของสภาพแวดล้อมเล็ก ๆ หรือห้องส่วนตัวของผู้ใช้งาน เพื่อควบคุมอุณหภูมิของสิ่งต่าง ๆ บริเวณนั้น ไม่ว่าจะเป็นอาหาร สิ่งของ หรือแม้กระทั่งอุณหภูมิของห้องขณะที่เรากำลังอยู่ในห้อง เพื่อที่จะให้อยู่ในอากาศที่เราต้องการ

## การดำเนินงาน

โดย Final Project ของเราได้ทำการวางแผนการดำเนินงาน ไว้ดังนี้

1. การคิดหัวข้อของ Final Project
2. คิดหลักการทำงานของ Project ว่าจะทำให้สามารถทำได้บ้าง ประโยชน์การใช้งาน
3. ศึกษารายละเอียดของการทำ Final Project เช่น Datasheet ของอุปกรณ์ที่ใช้ เป็นต้น
4. ลองทำ Simulation ในการต่อวงจรกับ ATmega328p เพื่อหลีกเลี่ยงความเสียหายจากความผิดพลาดในการต่อของจริง
5. นำอุปกรณ์ที่เข้ามาต่อเข้าในวงจร และเชื่อมเข้ากับ ATmega328p ให้ตรงตาม Proposal
6. เขียนโปรแกรม ให้วงจรสามารถทำงานได้ ตามที่เรา Simulate ไว้
7. แก้ไข ปรับปรุง และอาจปรับเปลี่ยนหลักการทำงานของวงจรตามที่วางแผน หากมีปัญหาด้านโปรแกรม

## รายชื่ออุปกรณ์

ชื่ออุปกรณ์	จำนวน	หมายเหตุ
Arduino Uno R3(ATMega328p)	1 ชิ้น	อ่านอินพุตจากเซ็นเซอร์ต่างๆ และควบคุมอุปกรณ์ภายนอกผ่านขาเอาต์พุต
TMP36	1 ชิ้น	ตัววัดอุณหภูมิ
Piezo	1 ชิ้น	ตัวแสดงสัญญาณเสียงเตือน
สายไฟ	38 เส้น	(จำนวนเส้นที่ใช้ใน Tinkercad)
Breadboard	2 บอร์ด	ตัวสร้างการเชื่อมต่อวงจรไฟฟ้าสำหรับการออกแบบหรือการทดสอบวงจร
LCD 16*2	1 ชิ้น	ตัวแสดงผลบนหน้าจอ
หลอดไฟ LED RGB	1 หลอด	ตัวแสดงสัญญาณไฟ
ตัวต้านทาน 330 โอห์ม	2 ชิ้น	ตัวที่ต้านการไหลผ่านของกระแสไฟฟ้า
Multimeter	1 ชิ้น	ใช้วัดว่ามีค่า ADC เข้าไปไหม เพื่อใช้คำนวณค่า TMP

โดยความรู้ที่เราได้นำมาใช้ใน Final Project นี้คือความรู้ของวิชา Embedded System ในเรื่องของ ATMega328p ในการควบคุมการทำงานร่วมกับอุปกรณ์อื่น ๆ นั่นก็คือ LCD, Temperature Sensor (TMP36), Piezo และ LED RGB สำหรับการแสดง Output เพื่อเป็นตัวกำหนดและตัวบอกอุณหภูมิ เป็นตัวแสดงสัญญาณเสียงเตือนลำโพงและแสดงไฟให้กับผู้ใช้งาน โดยหลักการทำงานที่เราได้คิดไว้คร่าว ๆ มีดังนี้

- การแจ้งเตือนไฟระดับ 0 คือ อุณหภูมิปกติ
- การแจ้งเตือนไฟระดับที่ 1 คือ อุณหภูมิสูงกว่าปกติแต่อยู่ในเกณฑ์ที่รับได้
- การแจ้งเตือนไฟระดับที่ 2 คือ อุณหภูมิสูงผิดปกติ
- การแจ้งเตือนด้วยสัญญาณเสียง คือ สัญญาณเตือนเมื่ออุณหภูมิสูงผิดปกติและจะหยุดเมื่ออุณหภูมิลดลงอยู่ในเกณฑ์ที่รับได้

## หลักการการทำงานของ Final Project

โดยเราจะอ้างอิงจากอุณหภูมิที่ตัว TMP36 วัดได้จากสภาพแวดล้อมนั้น ๆ โดยแบ่งเป็นกรณีดังนี้

- การแจ้งเตือนไฟระดับ 0 คือ อุณหภูมิไม่เกิน 30 องศา จะขึ้นเป็นไฟสีเขียว ซึ่งบ่งบอกได้ว่าอุณหภูมิขณะนี้ เป็นอุณหภูมิปกติ
- การแจ้งเตือนไฟระดับที่ 1 คือ อุณหภูมิสูงกว่า 30 องศาแต่ไม่ถึง 50 องศา จะขึ้นเป็นไฟสีเหลือง บ่งบอกได้ว่าขณะนี้อุณหภูมิสูงกว่าปกติในเกณฑ์ที่ให้ผู้ใช้งานพิจารณาว่าจะทำการเปิดอุปกรณ์เพิ่มความเย็น เช่น พัดลม หรือแอร์ หรือไม่
- การแจ้งเตือนไฟระดับที่ 2 คือ อุณหภูมิสูงตั้งแต่ 50 องศา ขึ้นไป จะขึ้นเป็นไฟสีแดงพร้อมมีเสียงแจ้งเตือน บ่งบอกได้ว่าอุณหภูมิขณะนี้สูงกว่าปกติในเกณฑ์ควรลดอุณหภูมิด่วน เช่น เปิดเครื่องทำความเย็นเพื่อลดอุณหภูมิ
- การแจ้งเตือนด้วยสัญญาณเสียง คือ อุณหภูมิสูงตั้งแต่ 50 องศา ขึ้นไป จะสัญญาณเตือนเป็นคลื่นเสียง ออกมาเรื่อย ๆ และสัญญาณเสียงจะหยุดดังได้ก็ต่อเมื่อ มีอุณหภูมิที่ลดลงจนถึง 40 องศา หรือต่ำกว่า

โดยอุณหภูมิที่ไม่ได้กำหนดไว้ตามที่เราคิดไว้ สามารถตั้งค่าเองได้ขึ้นอยู่กับจุดประสงค์ในการใช้งานอุปกรณ์นี้ของผู้ใช้งาน ซึ่งตัวอย่างที่เราคิดไว้ใช้สำหรับผู้ใช้งานทั่วไปที่ใช้ในห้องส่วนตัว

## การใช้ ATmega328p

ใน Final Project นี้ มีการใช้ module ของ AVR ได้แก่ AVR I/O และ AVR ADC โดยจะมีการใช้ AVR I/O ในการให้สัญญาณ Input Output แก่วงจร เช่น การให้สัญญาณไฟสีเขียว เหลือง แดง กับหลอดไฟ LED และการให้สัญญาณเพื่อให้ buzzer (Piezo) ส่งเสียงแจ้งเตือน ในส่วนของ AVR ADC มีการใช้รับสัญญาณ Analog แปลงเป็นสัญญาณ Digital เพื่อใช้ค่าในการทำงานต่อไป โดยจะมีรับสัญญาณ Analog จาก TMP36 ที่เป็นการวัดค่าอุณหภูมิ

## หลักการทำงานของ Code

ส่วนของโค้ดภายในวงจร สามารถแบ่งอธิบายได้ดังนี้

### 1.) activatePiezo()

```
void activatePiezo(uint8_t on) {  
    // Set the appropriate pin for the Piezo buzzer as an output  
    // Assuming the pin connected to the Piezo buzzer is PORTC0  
    DDRC |= (1 << DDC0);  
  
    // Control the Piezo buzzer based on the 'on' parameter  
    if (on) {  
        // Turn on the Piezo buzzer by setting the pin high  
        PORTC |= (1 << PORTC0);  
    } else {  
        // Turn off the Piezo buzzer by setting the pin low  
        PORTC &= ~(1 << PORTC0);  
    }  
}
```

เป็นการเปิดใช้งานตัว buzzer ของเรา โดยจะตั้งค่า pin ที่เชื่อมกับ buzzer ให้เป็นโหมด output และควบคุมสถานะของ pin นั้นตามค่าพารามิเตอร์ ถ้า on เป็นค่าที่ไม่เท่ากับศูนย์ จะตั้งค่าพินใน PORTC เป็น 1 เพื่อใช้งาน buzzer ถ้า on เป็น 0 จะเป็นการปิดการใช้งาน buzzer

### 2.) ledRGB()

```
void ledRGB(float temp) {  
    static float prevTemp = 0.0; // Variable to store the previous temperature  
  
    if (temp <= 30.0) {  
        PORTD |= (1 << PORTD0);  
        PORTD &= ~(1 << PORTD1) & ~(1 << PORTD2);  
    } else if (temp > 30.0 && temp < 50.0) {  
        PORTD |= (1 << PORTD0) | (1 << PORTD2);  
        PORTD &= ~(1 << PORTD1);  
    } else {  
        PORTD |= (1 << PORTD2);  
        PORTD &= ~(1 << PORTD0) & ~(1 << PORTD1);  
    }  
  
    // Check if the temperature has crossed 50.0 degrees  
    if (prevTemp <= 50.0 && temp > 50.0) {  
        activatePiezo(1); // Activate Piezo for 1 second when temperature crosses 50.0 degrees  
    } else if (prevTemp > 40.0 && temp <= 40.0) {  
        activatePiezo(0); // Deactivate Piezo when temperature goes below 40.0 degrees  
    }  
  
    // Store the current temperature as the previous temperature  
    prevTemp = temp;  
}
```

เป็นการสร้าง condition ให้กับตัว TMP36 ของเรา โดยหลังจากที่รับอุณหภูมิมาแล้วจะทำการส่ง Output ไปที่จอ LCD และเปลี่ยนสีของ LED ตาม condition ที่เรากำหนดไว้ แต่ในกรณีที่อุณหภูมิเกิน 50 องศา จะส่งสัญญาณให้ buzzer แจ้งเตือนจนกว่าอุณหภูมิจะลงมา 40 องศาหรือต่ำกว่า จึงจะหยุดการทำงาน

### 3.) setup()

```
void initLCD() {
    DDRB |= 0x0F; // set pins 8-11 as output
    PORTB &= 0xF0; // set pins 4-7 as low (not used)

    // set pins 0-2 as output for ledRGB1()
    DDRD |= (1 << DDD0) | (1 << DDD1) | (1 << DDD2);

    // set pins 5-7 as output for ledRGB2()
    DDRD |= (1 << DDD5) | (1 << DDD6) | (1 << DDD7);

    DDRD |= (1 << DDD3) | (1 << DDD4); // set pins 3-4 as output
    PORTD &= ~(1 << PORTD3) & ~(1 << PORTD4); // set pins 3-4 as

    sendLCDCommand(0x33);
    sendLCDCommand(0x32);
    sendLCDCommand(0x28);
    sendLCDCommand(0x0E);
    sendLCDCommand(0x01);
    sendLCDCommand(0x80); // Cursor start line 1
}

void initADC() {
    // Set reference voltage to AVcc
    ADMUX |= (1 << REFS0);
    // Enable ADC with prescaler 128
    ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADIFSC);
}

void setup() {
    initLCD();
    initADC();
}
```

ในส่วนนี้จะเป็นการรับค่ามาแสดงผลบนจอ LCD รวมถึงการเคลียร์หน้าจอหากมีการเปลี่ยนข้อมูล เพื่อรอแสดงผลข้อมูลต่อไป และรวมไปถึงการตั้งค่า ADC ซึ่งเป็นการรับค่าอุณหภูมิที่มาจาก TMP36

### 4.) loop()

```
void loop() {
    sendLCDCommand(0x01); // Clear Display
    sendLCDCommand(0x80); // Show results at the beginning
    uint16_t adcValue;

    // Start conversion
    ADMUX |= (1 << MUX0);
    ADCSRA |= (1 << ADSC);

    // Wait for ADIF
    while (ADCSRA & (1 << ADSC)); // or -> while (!(ADCSRA & ADIF))

    adcValue = ADC / 1024.0 * 5000.0; // adcValue/1024.0*5 = Volt
    float Vout = adcValue;
    float temp = (Vout - 500) / 10.0;

    sprintf(buffer, "Temp = ");
    lcdDisplayString(buffer);
    dtostrf(temp, 0, 2, buffer);
    lcdDisplayString(buffer);
    ledRGB(temp);

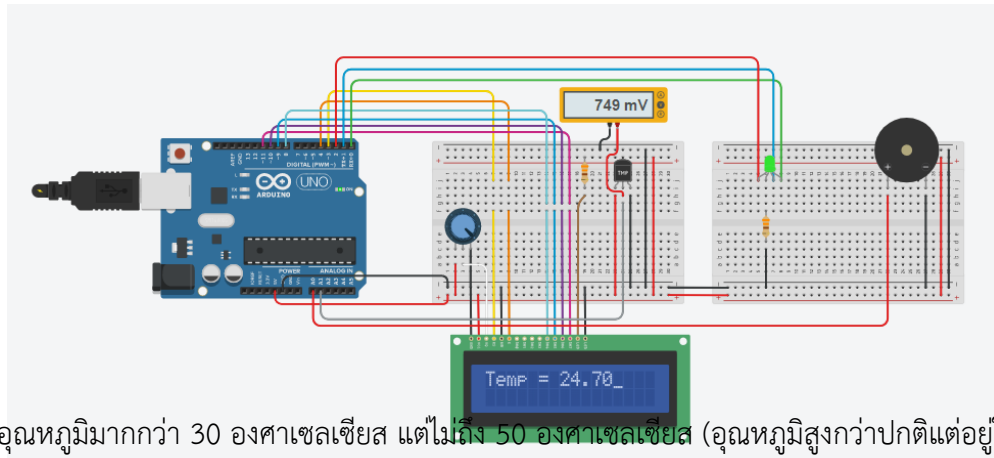
    _delay_ms(1000);
}
```

ในส่วนของฟังก์ชัน loop จะทำงานวนไปเรื่อย ๆ โดยจะเป็นการรับค่าที่มาจากตัว TMP36 แล้วนำมากำหนดพร้อมแสดงผลออกมาผ่านจอ LCD

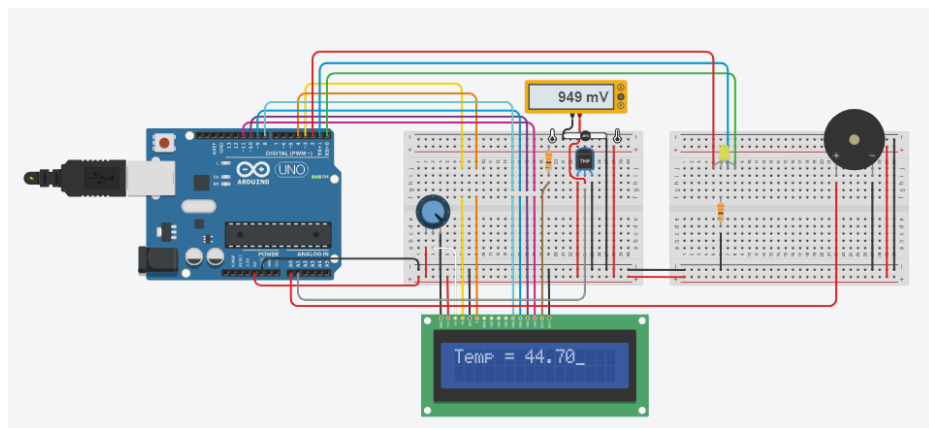
## ผลลัพธ์ของ Final Project

หลังจากที่เราได้วางแผนการทำงานเสร็จเรียบร้อยแล้ว ก็ทำงานเปิดใช้งานตัวเครื่องมือของเรา ซึ่งผลลัพธ์ที่เราได้ออกแบบเอาไว้ทั้งหมด 3 กรณีมีผลลัพธ์ดังนี้

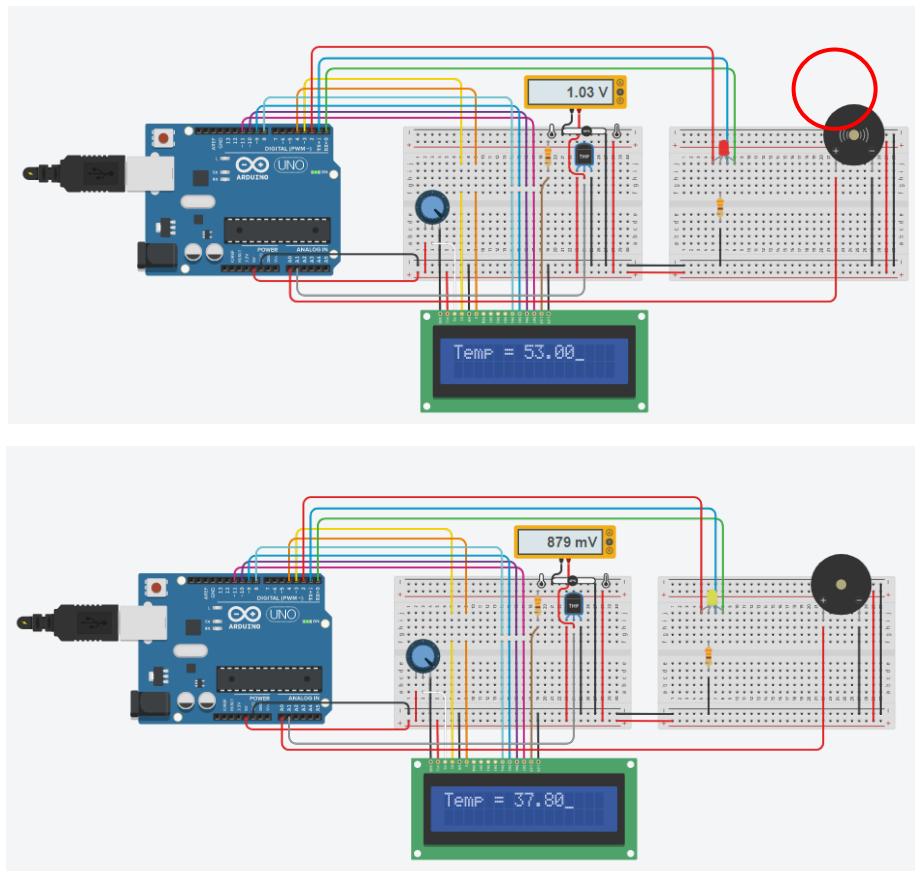
กรณีที่ 1: อุณหภูมิน้อยกว่า 30 องศาเซลเซียส (อุณหภูมิปกติ) หลอดไฟ LED สีเขียวและ buzzer ไม่ทำงาน



กรณีที่ 2: อุณหภูมิมากกว่า 30 องศาเซลเซียส แต่ไม่ถึง 50 องศาเซลเซียส (อุณหภูมิสูงกว่าปกติแต่อยู่ในเกณฑ์ที่รับได้) หลอดไฟ LED สีเหลืองและ buzzer ไม่ทำงาน



กรณีที่ 3: อุณหภูมิมากกว่า 50 องศาเซลเซียส (อุณหภูมิสูงผิดปกติ) หลอดไฟ LED สีแดงและ buzzer ทำงาน และ buzzer หยุดทำงานหลังจากอุณหภูมิลงมาต่ำกว่า 40 องศาเซลเซียส



Project Demo: [Circuit design Final-Project | Tinkercad](#)