

เอกสารประกอบการทดลอง
วิชา 01236260 Cyber-Physical System and Sensor

ผู้สอน: ผศ.ดร.วันวิสา ชชวงศ์
ดร.นัชนัยน์ รุ่งเหมือนฟ้า
ผศ.สรพงษ์ วชิรรัตนพรกุล

นักศึกษาที่ทำการทดลอง

- | | | |
|-------------------------------|--------------------------|------------------------------|
| 1. ชื่อ <u>นางสาวณัฐร์นภา</u> | นามสกุล <u>ยุทธนา</u> | รหัสนักศึกษา <u>67010272</u> |
| 2. ชื่อ <u>นางสาวกุณร์พญ</u> | นามสกุล <u>ตั้งสันณ์</u> | รหัสนักศึกษา <u>67010960</u> |
| 3. ชื่อ <u>นายธีรักษ์</u> | นามสกุล <u>ฟันฟั่ง</u> | รหัสนักศึกษา <u>67011018</u> |
| 4. ชื่อ | นามสกุล | รหัสนักศึกษา |

--

เอกสารประกอบการทดลองนี้ เป็นส่วนหนึ่งของวิชา 01236254 Circuits and Electronics

ภาคเรียนที่ 1 ประจำปีการศึกษา 2566

สาขาวิชาวิศวกรรมระบบไอโอทีและสารสนเทศ

และสาขาวิชาวิศวกรรมระบบไอโอทีและสารสนเทศ และพิสิกส์อุตสาหกรรม

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การทดลองเรื่อง Voltage Sensor

- ทำการ Download Driver CP210x จาก [CP210x USB to UART Bridge VCP Drivers - Silicon Labs \(silabs.com\)](http://www.silabs.com/products/interface/uart-bridge-drivers/cp210x-uart-drivers) โดยเลือก download ตัว **CP210x Windows Drivers** เมื่อ download เสร็จให้ทำการแตกไฟล์บีบอัดนี้ แล้วเลือกติดตั้ง **CP210xVCPIInstaller_x64**
- ทำการเปิดโปรแกรม Arduino IDE เข้าไปที่เมนู File>Preferences จะมีหน้าต่าง Preferences แสดงขึ้นมา ให้ทำการ copy link ต่อไปนี้

https://dl.espressif.com/dl/package_esp32_index.json

 แล้วนำไปวางในช่อง Additional boards managers URLs และกด OK
- เข้าไปที่เมนู Tools>Board>Boards Manager จะได้หน้าต่าง Boards Manager ในช่องค้นหาให้ใส่คำว่า ESP32 แล้วทำการติดตั้ง Boards ESP32
- ทำการเชื่อมต่อบอร์ด ESP32 เข้ากับคอมพิวเตอร์ด้วยสาย USB แล้วที่ Arduino IDE ให้เข้าไปที่ Board>ESP32 แล้วเลือกบอร์ดเป็น ESP32 Dev Module หรือ Node32s และ Com Port ที่เชื่อมต่อ
- ให้อ่านค่าตัวถ่วงด้านทันของ Voltage Sensor

$$R1 = 3012 = \underline{29.89} \text{ k } \Omega$$

$$R2 = 7502 = \underline{7.494} \text{ k } \Omega$$

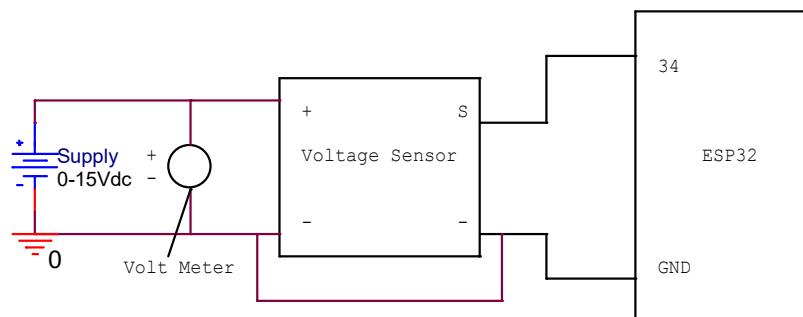
- จงหาค่าแรงดันสูงสุดที่ Voltage Sensor จะวัดได้ เมื่อมีค่า MaxADCin V เป็น 5.0V กับ 3.0V จากสูตรต่อไปนี้

$$V_{inMax} = \left(\frac{R1 + R2}{R2} \right) V_{MaxADCin} \quad V$$

ตั้งนั้นที่ 5.0V จะได้ $V_{inMax} = \underline{24.94} \text{ k } V$

ตั้งนั้นที่ 3.0V จะได้ $V_{inMax} = \underline{14.97} \text{ k } V$

- จงต่อแหล่งจ่ายแรงดันไฟฟ้า, โมดูล Voltage Sensor และ ESP32 ตามรูป และป้อนแรงดันไฟฟ้าที่ 5V



8. จงเขียน Code เพื่อทำการอ่านค่าระดับที่ได้จากการวัดแรงดันไฟฟ้าผ่านทาง Voltage Sensor และบันทึกค่าระดับที่ได้

ค่าระดับที่ได้จากการวัดแรงดันไฟฟ้า 5V มีค่า = 4095

9. จงแก้ไข Code ที่เขียนได้ในข้อ 8 เพื่อทำให้สามารถอ่านค่าระดับ และแรงดันไฟฟ้าที่วัดได้จาก Voltage Sensor

```
sketch_oct8a.ino
23 void loop() {
24
25     // ADC value from voltage sensor
26     int analog_voltage_value = analogRead(ADCpin);
27     Serial.print("voltage sensor : [");
28     Serial.print(analog_voltage_value);
29     Serial.println("] Volts");
30     delay(500);
31
32     // Calculate a Voltage from ADC
33     float vout = (analog_voltage_value * 3.3) / (4095);
34     Serial.print("[sensor] : [");
35     Serial.print(vout);
36     Serial.println("] Volts");
37     delay(500);
38 }
```

10. ทำการปรับเหล่งจ่ายแรงดันไฟฟ้า ตามตารางที่ 1 และทำการบันทึกค่าระดับและแรงดันไฟฟ้าที่วัดจาก Voltage Sensor เปรียบเทียบกับมัลติมิเตอร์พร้อมทั้งหาค่า Error จากสูตรต่อไปนี้

$$\text{Error} = \frac{\text{แรงดันไฟฟ้าที่วัดด้วยมิเตอร์} - \text{แรงดันไฟฟ้าที่วัดsensor}}{\text{แรงดันไฟฟ้าที่วัดด้วยมิเตอร์}} \times 100 \%$$

ตารางที่ 1

V จากแหล่งจ่าย	ค่าระดับ Analog	V จาก Sensor	V จากมิเตอร์	Error
0V	0	0 V.	0 V.	0 %
1.58V	1810	1.55 V.	1.5749 V.	1.58 %
10V	2240	9.02 V.	9.995 V.	9.75 %
15V	3712	14.94 V.	14.999 V.	0.39 %

คำถามท้ายการทดลอง

1. จงอธิบายว่าทำไม ค่าระดับ และแรงดันไฟฟ้าที่วัดได้จาก Sensor มีค่าไม่นิ่ง
 รายงานไฟฟ้าความต้านทาน ส่วนใหญ่แล้วจะคงที่ แต่แรงดันไฟฟ้า
 จังหวะท่าไปนี้ ค่าระดับเกิดการผันผวนเพียง

การทดลองเรื่อง Current Sensor: ACS712

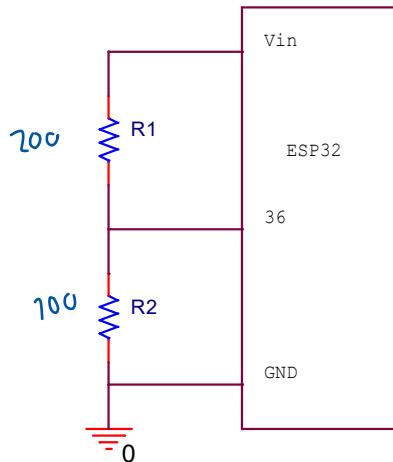
- เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา ACS712 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
- ต่อ ESP32 เข้ากับคอมพิวเตอร์ และทำการวัดและบันทึกแรงดันไฟฟ้าที่ขา Vin ของบอร์ด ESP32
แรงดันไฟฟ้าที่ขา Vin ของ ESP32 = 4.82 V
- ต่อวงจรตามรูป

$$V_{out} = \left(\frac{R_1}{R_1 + R_2} \right) V_{in}$$

$$3.3 \pm 0.05V = \left(\frac{220 \Omega}{220 + R_2} \right) 4.82$$

$$R_2 \approx 16.54 \Omega$$

$$R_2 = 100 \Omega$$



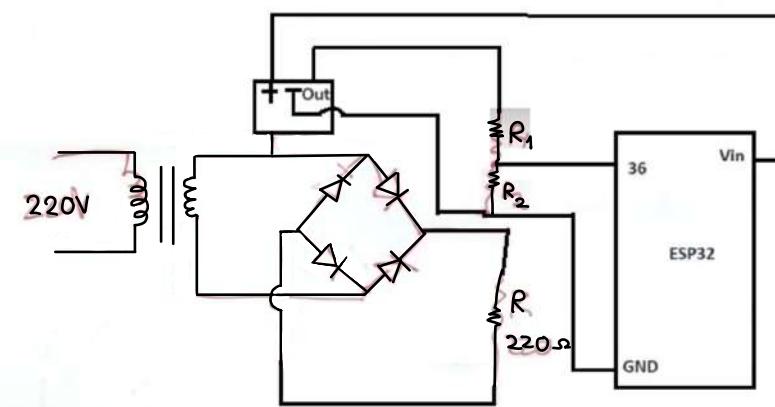
- จงหาค่า R1, R2 โดยให้ R1 อยู่ในช่วง 220-680Ω และแรงดันไฟฟ้าที่เข้าขา ESP32 จะต้องมีค่า ในช่วง 3.29-3.3V (เลือกค่าแรงดันไฟฟ้าค่าเดียวเท่านั้น)

$$R_2 = \underline{100} \Omega$$

- จงต่อวงจรตามรูป และใช้มิเตอร์วัดแรงดันไฟฟ้าตรงขา ADC (36 หรือ 34 หรือ 32) ของ ESP32 ว่า ตรงกับที่คำนวณในข้อ 4 หรือไม่ บันทึกแรงดันไฟฟ้าที่ได้

$$\text{แรงดันไฟฟ้าตรงขา ADC } (\underline{16} \text{ ใส่หมายเลข}) = \underline{3.28} V$$

- จงต่อวงจร โมดูล ACS712 และ ESP32 ตามรูป $\text{error} = 0.60\%$



7. จงเขียน Code เพื่อหาค่าต่อไปนี้

Sensor	Power Supply	ERROR
0.52	0.5	4%
0.92	1	8%
1.46	1.5	2.67%
1.88	2	6%
2.43	2.5	2.8%
2.93	3	2.33%
3.44	3.5	1.71%
3.86	4	3.5%
4.22	4.5	6.22%
4.56	5	8.8%

```
void loop() {
    float ADC = analogRead(CURRENT_SENSOR_PIN);
    Serial.print("ADC : ");
    Serial.println(ADC);

    // change ADC to voltages compare with 3.3V
    float voltage = (ADC/4095)*3.3;
    float set_zero_voltage = voltage - 2.16;
    float current_sensor = set_zero_voltage/0.189;
    float linear_calibration = (alpha*current_sensor)+beta;
    float polynomial_calibration = (alpha_poly*current_sensor*current_sensor)+(beta_poly*current_sensor)+gamma;

    Serial.print("Voltage sensor : ");
    Serial.println(voltage);
    Serial.print("Current sensor : ");
    Serial.println(current_sensor);
    Serial.print("Calibrated Linear Current Sensor : ");
    Serial.println(linear_calibration);
    Serial.print("Calibrated Polynomial Current Sensor : ");
    Serial.println(polynomial_calibration);
    Serial.println(".....");
    delay(700);
}
```

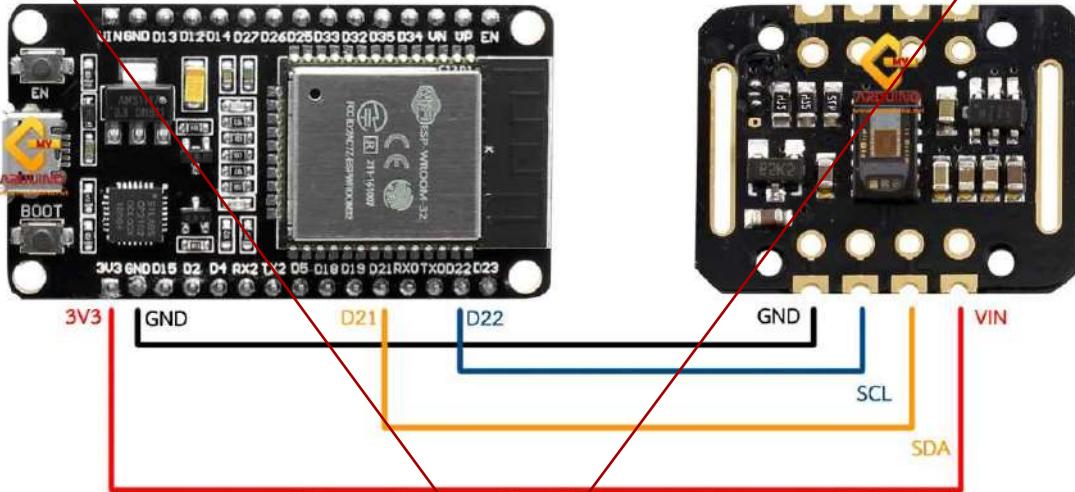
คำถามท้ายการทดลอง

1. จงอธิบายว่าจะทำอย่างไรจึงจะทำให้ ACS712 สามารถตรวจน้ำและไฟฟ้าต่ำของไฟฟ้า AC ได้อย่างถูกต้อง

ต้อง calibrated ใน order ที่สูงขึ้นเพื่อให้ในค่าความชันมีมากขึ้น

การทดลองเรื่อง Heart Rate sensor: MAX30102

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา MAX30102 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ทำการต่อโมดูล Heart Rate MAX30102 เข้ากับ ESP32 ดังรูป



Ref: [สอนใช้งาน ESP32 MAX30102 เซ็นเซอร์วัดอัตราการเต้นของหัวใจ สำหรับ Arduino อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ฟรี](http://cybertice.com)
(cybertice.com)

3. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถอ่านค่า Heart Rate จากนิวมีโอได้

Code

4. ทำการทดสอบโมดูล Heart Rate โดยยังไม่ต้องวางนิ้วไปที่ โมดูล บันทึกรูป Serial Monitor

รูปผล Serial Monitor

5. ทำการทดสอบโมดูล Heart Rate โดยวางนิ้วไปที่ โมดูล บันทึกรูป Serial Monitor

รูปผล Serial Monitor

6. จงเขียนและบันทึก Code ที่สามารถแจ้งเตือนค่า Heart Rate ที่มีค่ามากกว่า 90 ครั้ง/วินาที ผ่านทาง Internet (Netpie หรือ NodeRed หรือ Blank หรือ Line) พร้อมทั้งบันทึกภาพผลการทดลอง ที่ได้

Code

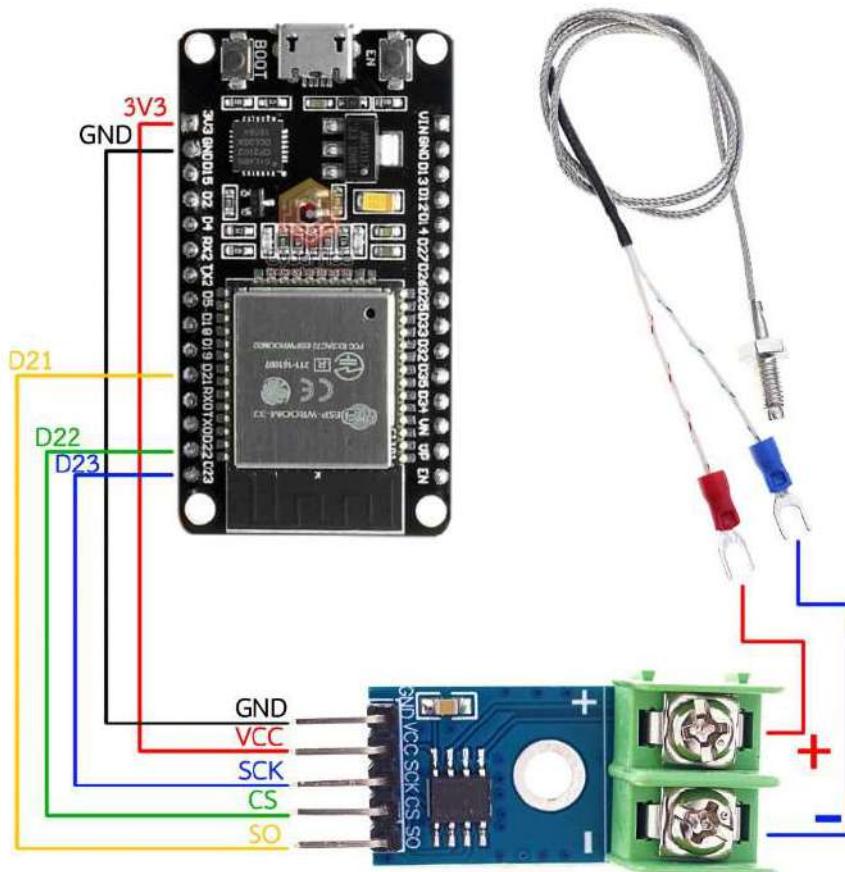
รูปผลการทดลอง

คำถาวรท้ายการทดลอง

1. จงอธิบายวิธีการวัด Heart Rate พร้อมกันอย่างน้อย 3 เครื่องแล้วส่งค่าไปประมวลผลผ่านทาง Raspberry Pi ก่อนแสดงผลทาง Internet เพื่อนำไปใช้ในการวัดค่า Heart Rate ของนักกีฬาที่วิ่งบนลู่วิ่งพร้อมกัน

การทดลองเรื่อง K-type Thermocouple sensor: MAX6675

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา MAX6675 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ทำการต่อโมดูล K-type thermocouple MAX6675 เข้ากับ ESP32 ดังรูป



Ref: สอนใช้งาน ESP32 วัดอุณหภูมิ ความร้อนสูง Temperature Sensor Probe with MAX6675 Module - ชาช Arduino
อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ส่งฟรี (cybertice.com)

3. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถอ่านค่าอุณหภูมิในรูปขององศา °C, °F และ °K

```

#define CS 15 // Chip Select Pin សម្រាប់បន្ទូលទឹកបែប
MAX6675 Thermocouple(CLK, CS, SO);

void setup() {
  Serial.begin(9600);
}

void loop() {
  // declare variables that collect value from sensor ធម៌តុដែលត្រូវបានរកសារពីការបែបបន្ទូលទឹក
  float Celsius = Thermocouple.readCelsius();
  float Fahrenheit = Thermocouple.readFahrenheit();
  float Kelvin = Celsius + 273.15;

  // show Celsius temperature នៃការបែបបន្ទូលទឹក
  Serial.print("Temperature [Celsius] : ");
  Serial.println(Celsius);
  Serial.println("-----");
  delay(100);

  // Show Fahrenheit temperature នៃការបែបបន្ទូលទឹក
  Serial.print("Temperature [Fahrenheit] : ");
  Serial.println(Fahrenheit);
  Serial.println("-----");
  delay(100);

  // Show Kelvin temperature នៃការបែបបន្ទូលទឹក តែគាំទេ
  Serial.print("Temperature [kelvin] : ");
  Serial.println(Kelvin);
  Serial.println("-----");
  delay(100);
  delay(700);
}

```

4. ធានាអនុញ្ញាតការងារបន្ទូលទឹក MAX6675 ប័ណ្ណក្នុង Serial Monitor

```

Temperature [Kelvin] : 296.40
-----
Temperature [Celsius] : 23.25
-----
Temperature [Fahrenheit] : 73.85
-----
Temperature [Kelvin] : 296.40
-----
Temperature [Celsius] : 23.00
-----
Temperature [Fahrenheit] : 73.40
-----
Temperature [Kelvin] : 296.15
-----

```

5. ធានាអនុញ្ញាតការងារបន្ទូលទឹក MAX6675 ដោយការនាំព្រូបខែងមិនឲ្យចូលរួមនៅក្នុងដំឡើងដើម្បីក្លាសការងារបន្ទូលទឹកពីការងារបន្ទូលទឹកដែលត្រូវបានរកសារពីការបែបបន្ទូលទឹក

```

Temperature [Celsius] : 91.50
-----
Temperature [Fahrenheit] : 196.70
-----
Temperature [Kelvin] : 364.65
-----
Temperature [Celsius] : 92.00
-----
Temperature [Fahrenheit] : 197.60
-----
Temperature [Kelvin] : 365.15
-----
Temperature [Celsius] : 92.25
-----

```

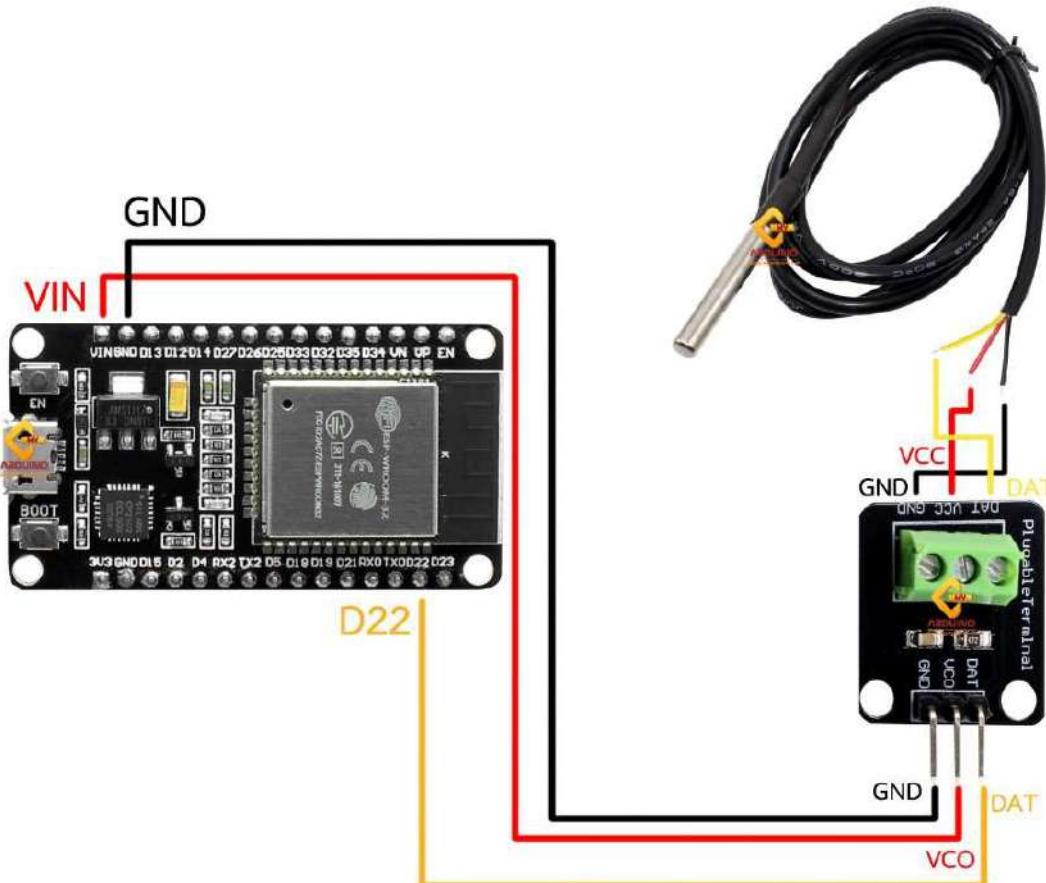


6. ทำการทดสอบโมดูล MAX6675 โดยการนำไฟรบของโมดูลไปแขวนในน้ำแข็งครึ่งแก้ว และบันทึกรูป การทดลองกับรูปจาก Serial Monitor

```
Temperature [Kelvin] : 284.40
-----
Temperature [Celsius] : 10.50
-----
Temperature [Fahrenheit] : 50.90
-----
Temperature [Kelvin] : 283.65
-----
Temperature [Celsius] : 10.00
-----
Temperature [Fahrenheit] : 50.00
-----
Temperature [Kelvin] : 283.15
```

การทดลองเรื่อง Full Waterproof Temperature Sensor: DS18T20

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา DS18T20 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ทำการต่อโมดูล Full Waterproof Temperature Sensor DS18T20 เข้ากับ ESP32 ดังรูป



Ref: [สอนใช้งาน ESP32 DS18B20 Full Waterproof Temperature Sensor เชื่อมต่อวัดอุณหภูมิในน้ำ - ขา Arduino อุปกรณ์ Arduino ถูกภาพดี ราคาถูก ส่งไว ส่งฟรี \(cybertice.com\)](#)

3. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถอ่านค่าอุณหภูมิในรูปขององศา °C

```

    open();
    case 0x20:
        type_s = 0;
        break;
    case 0x22:
        type_s = 0;
        break;
    default:
        return;
    }
    ds.reset();
    ds.select(addr);
    ds.write(0x40); // start conversion, with parasite power on at the end
    delay(1000); // maybe 750ms is enough, maybe not.

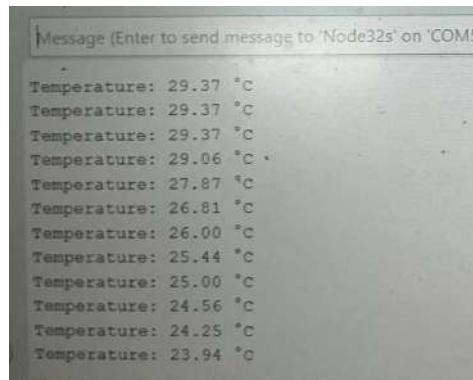
    present = ds.reset();
    ds.select(addr);
    ds.write(0xE0); // Read Scratchpad

    for (i = 0; i < 9; i++) { // we need 9 bytes
        data[i] = ds.read();
    }

    int16_t raw = (data[1] << 8) | data[0];
    if (type_s) {
        raw = raw << 3; // 9 bit resolution default
        if (data[7] == 0x00) {
            // "count result" gives full 12 bit resolution
            raw = (raw & 0x0FFF) + 12 - data[8];
        }
    } else {
        byte cfg = (data[4] & 0x0F);
        // at lower res, the low bits are undefined, so let's zero them
        if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
        else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 107.5 ms
        else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
        // default is 12 bit resolution, 750 ms conversion time
    }
}

```

4. ทำการทดสอบโมดูล DS18T20 โดยการวัดอุณหภูมิในอากาศ บันทึกรูป Serial Monitor



5. ทำการทดสอบโมดูล DS18T20 โดยการนำไฟรบของโมดูลไปแช่ในน้ำร้อนที่ต้มจนเดือด แล้วบันทึกรูปการทดลองกับรูปจาก Serial Monitor

```

Temperature = 26.57 Celsius, 194.60 Fahrenheit
Temperature = 90.50 Celsius, 194.90 Fahrenheit
Temperature = 90.69 Celsius, 195.24 Fahrenheit
Temperature = 90.87 Celsius, 195.57 Fahrenheit
Temperature = 91.06 Celsius, 195.91 Fahrenheit
Temperature = 91.12 Celsius, 196.02 Fahrenheit
Temperature = 91.25 Celsius, 196.25 Fahrenheit
Temperature = 91.31 Celsius, 196.36 Fahrenheit
Temperature = 91.37 Celsius, 196.48 Fahrenheit
Temperature = 91.44 Celsius, 196.59 Fahrenheit
Temperature = 91.44 Celsius, 196.59 Fahrenheit
Temperature = 91.50 Celsius, 196.70 Fahrenheit
Temperature = 91.56 Celsius, 196.81 Fahrenheit
Temperature = 91.50 Celsius, 196.70 Fahrenheit
Temperature = 91.37 Celsius, 196.48 Fahrenheit
Temperature = 91.44 Celsius, 196.59 Fahrenheit
Temperature = 91.75 Celsius, 197.15 Fahrenheit
Temperature = 91.87 Celsius, 197.37 Fahrenheit
Temperature = 92.00 Celsius, 197.60 Fahrenheit
Temperature = 92.12 Celsius, 197.82 Fahrenheit

```



6. ทำการทดสอบโมดูล DS18T20 โดยการนำโปรแกรมของโมดูลไปเชื่อมต่อเครื่องแก้ว แล้วบันทึกรูป การทดลองกับรูปจาก Serial Monitor

```
Temperature = 19.44 Celsius, 66.99 Fahrenheit
Temperature = 18.87 Celsius, 65.97 Fahrenheit
Temperature = 18.37 Celsius, 65.07 Fahrenheit
Temperature = 17.94 Celsius, 64.29 Fahrenheit
Temperature = 17.56 Celsius, 63.61 Fahrenheit
Temperature = 17.06 Celsius, 62.71 Fahrenheit
Temperature = 15.38 Celsius, 59.67 Fahrenheit
Temperature = 13.81 Celsius, 56.86 Fahrenheit
Temperature = 12.50 Celsius, 54.50 Fahrenheit
Temperature = 11.50 Celsius, 52.70 Fahrenheit
Temperature = 10.75 Celsius, 51.35 Fahrenheit
Temperature = 10.13 Celsius, 50.22 Fahrenheit
```

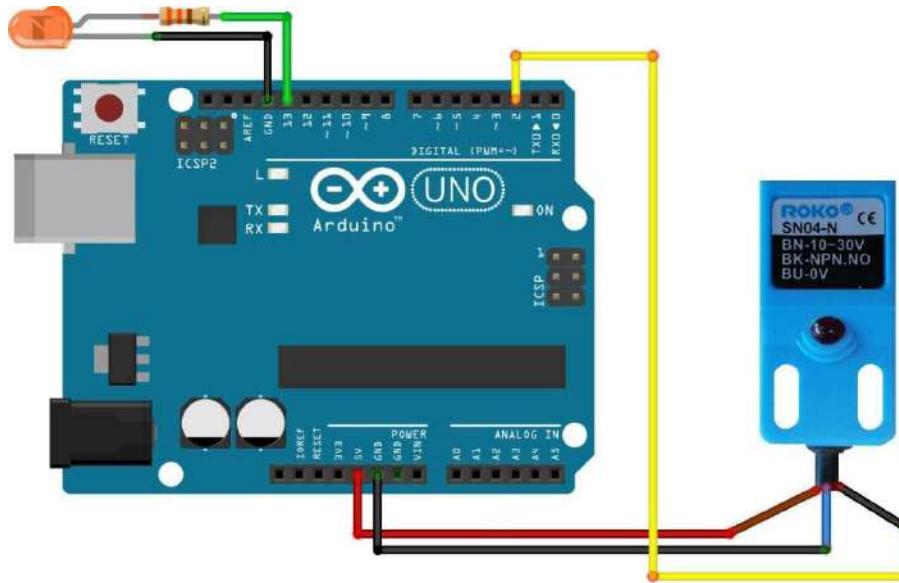
คำถามที่ยังไม่ได้ตอบ

1. โมดูล MAX6675 และโมดูล DS18T20 มีความละเอียดในการวัดอุณหภูมิเป็นอย่างไร และแต่ละโมดูล เหมาะกับการนำไปใช้งานด้านไหน จงอธิบาย

Module MAX6675 รับสัญญาณ Analog จากอินพุต sensor มีความไวต่อ noise
แต่ DS18B20 วัดแบบ Digital sensor

การทดลองเรื่อง Metal Inductive Sensor: SN04-N

- ทำการต่อโมดูล Metal Inductive Sensor SN04-N เข้ากับ Arduino UNO ดังรูป



- ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถตรวจจับโลหะได้

```
#define LED 34
#define SENSOR 35
float sensor;
bool sensor_detect;

void setup() {
  Serial.begin(9600);
  pinMode(SENSOR, INPUT);
  pinMode(LED, OUTPUT);
}

void loop() {
  sensor = digitalRead(SENSOR);
  if(sensor==0){
    Serial.println("Found the metal!");
  }
  else{
    Serial.println("Doesn't found the metal!");
  }
}
```

3. ทำการทดสอบโมดูล SN04-N โดยยังไม่ต้องนำโลหะมาวางใกล้โมดูล บันทึกรูป Serial Monitor

4. ทำการทดสอบป้องกัน SN04-N โดยการนำโลหะ เช่นคิลป์หนีบกระดาษ หรือเข็มหมุด มาวางใกล้โมดูลตามระยะในตาราง แล้วบันทึกผลลงในตารางและ บันทึกรูป Serial Monitor

	โลหะ	ระยะ 1 ซม.	ระยะ 8 มม.	ระยะ 4 มม.	ระยะ 1 มม.
1.	คลิปหนีบกระดาษ	X	X	X	✓
2.	เข็มหมุด	X	X	X	✓
3.	ตัวต้านทานปรับค่าได้	✓	✓	✓	✓
4.	เหรี่ยญูบาท	X	✓	✓	✓
5.	เหรี่ยญูสิบบาท	✓	✓	✓	✓
6.	หัวเข็มขัดโลหะ	✓	✓	✓	✓

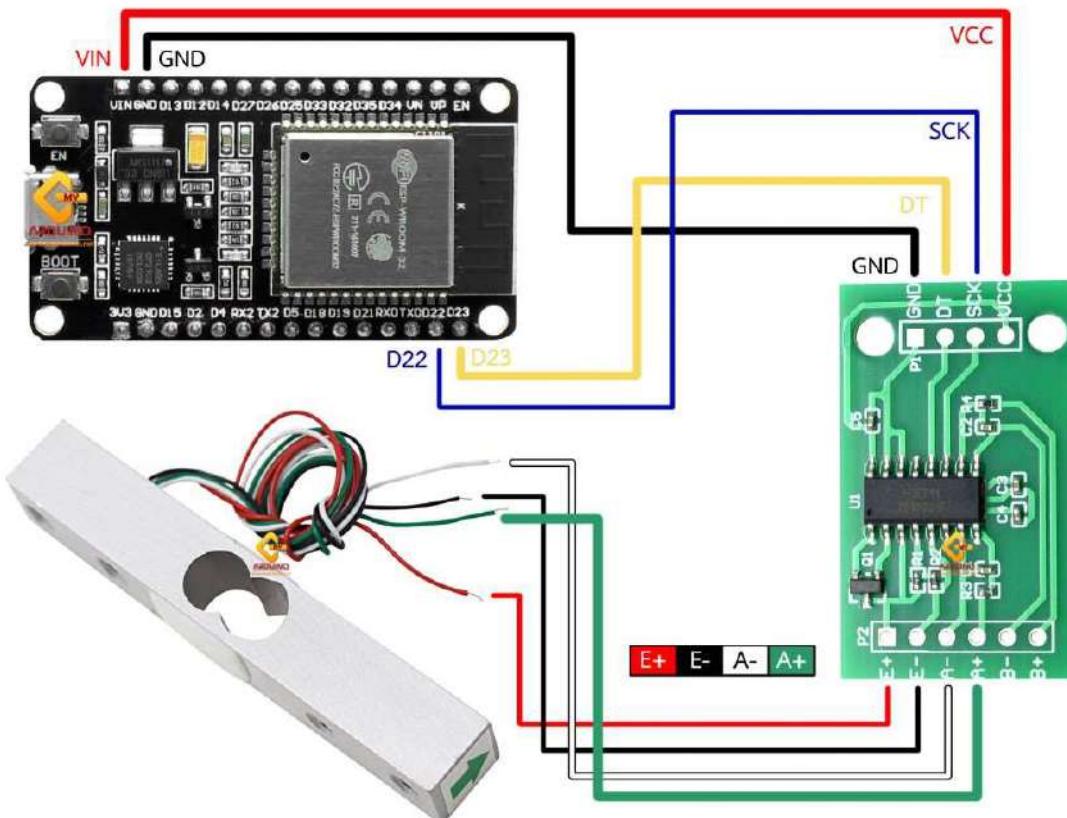
คำจำกัดความทั่วไปการทดลอง

- โมดูล SN04-N สามารถตรวจจับโลหะได้ที่ระยะได ขนาดของโลหะมีผลต่อการตรวจจับหรือไม่ และมีปัจจัยอื่นหรือไม่ที่ทำให้สามารถตรวจจับโลหะในระยะที่ใกล้ขึ้น พร้อมทั้งบอกการนำไปประยุกต์ใช้งานอย่างน้อย 3 อย่าง

SNO 4 - N ភាគាសកទម្រង់បំពុំ: តីកំរើ: ប. ពេះកប់ ១-៣ ផ្លូវលេខ២៤ ឱ្យនាន់ដែលតែងការអរគុណបំផុំ
អំពីគ្រប់រំលែក តាមកម្រិត: នាក់សារធម៌, និងស្ថាប័ន ពេរ: ៩:៨០ការពាណិជ្ជកម្ម ការអនុវត្តន៍
នៅក្នុងការបង្កើតរំលែក

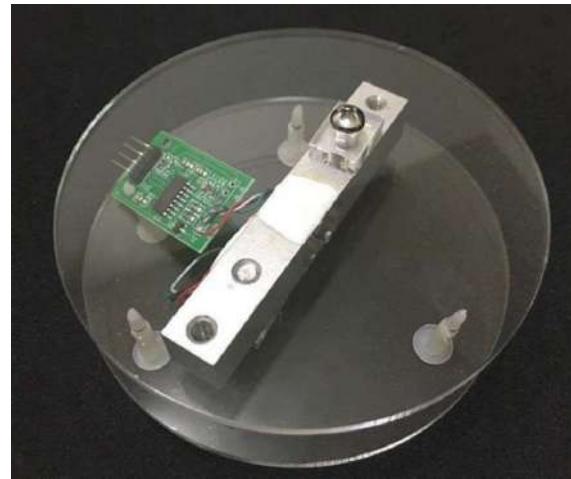
การทดลองเรื่อง Load Cell Sensor: HX711

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา HX711 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ทำการต่อโมดูล Load Cell Sensor HX711 เข้ากับ ESP32 ดังรูป



Ref: [สอนใช้งาน ESP32 ทำเครื่องชั่งน้ำหนัก เชื่อมอุปกรณ์ ESP32 กับ Load Cell HX711 ผ่าน Arduino IDE](http://cybertice.com)

3. ให้นักศึกษาทำการประกอบฐานการซึ่งน้ำหนัก โดยให้มีการวัด Loadcell ไว้ที่ตู้ฐานซึ่งน้ำหนักตามรูปตัวอย่าง โดยสามารถออกแบบและใช้วัสดุอะไรก็ได้ตามสะดวก



4. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถวัดค่าน้ำหนักที่ซึ่งได้ในหน่วยของ Kg กับ mg โดยสามารถตัดสินใจได้ว่าว่าของที่ซึ่งมีน้ำหนักเป็น kg หรือ mg

```

2 // HX711 CIRCUIT
3
4 // ปรับขนาดการเข้ามต่องคุณ
5 uint8_t dataPin = 23;
6 uint8_t clockPin = 22;
7
8 // 1. กำหนดค่าเริ่มต้น (kg/mg weight) ที่ใช้ในการปรับเปลี่ยน
9 float known_weight_g = 162.0; // ปานน้ำหนักถ่วง x กิโล
10
11 void setup()
12 {
13     Serial.begin(115200);
14     Serial.println();
15     Serial.println(_FILE_);
16     Serial.print("HX711_LIB_VERSION: ");
17     Serial.println(HX711_LIB_VERSION);
18     Serial.println();
19     Serial.println();
20
21     scale.begin(dataPin, clockPin);
22
23     // ตั้งค่า calibration Factor ที่คุณทราบได้มาเพื่อน้ำหนัก x กรัม
24     scale.set_scale(-441.191345);
25
26     // ตั้งค่าบันทึกเวลา
27     scale.tare(20);
28     Serial.println("scale is ready. Place the x g weight on it.");
29 }

```

5. ทำการทดสอบโมดูล HX711 โดยยังไม่ต้องมีสิ่งของวางบนฐานซึ่งน้ำหนัก แล้วบันทึกรูป Serial

```

13:09:56.322 -> Reading: -0.17 g
13:09:57.317 -> Reading: -0.10 g
13:09:58.279 -> Reading: 0.00 g
13:09:59.280 -> Reading: -0.13 g
13:10:00.279 -> Reading: -0.17 g
13:10:01.281 -> Reading: -0.21 g
13:10:02.325 -> Reading: -0.34 g
13:10:03.310 -> Reading: -0.36 g
13:10:04.323 -> Reading: -0.43 g
13:10:05.279 -> Reading: -0.34 g
13:10:06.317 -> Reading: -0.40 g
13:10:07.315 -> Reading: -0.09 g
13:10:08.311 -> Reading: -0.14 g
13:10:09.317 -> Reading: -0.16 g
13:10:10.320 -> Reading: -0.01 g
13:10:11.279 -> Reading: -0.39 g
13:10:12.279 -> Reading: -0.23 g

```

6. ทำการทดสอบโมดูล HX711 โดยทำการวางสิ่งของต่างๆ ตามตาราง แล้วบันทึกการทดลองกับรูปจาก Serial Monitor พร้อมทั้งบันทึกค่าน้ำหนักที่วัดได้ลงในตาราง



```

13:09:22.214 > Reading: 161.80 g | Error: -0.12 %
13:09:24.298 > Reading: 161.77 g | Error: -0.14 %
13:09:25.281 > Reading: 161.69 g | Error: -0.19 %
13:09:26.322 > Reading: 161.52 g | Error: -0.30 %
13:09:27.306 > Reading: 161.77 g | Error: -0.14 %
13:09:28.325 > Reading: 162.22 g | Error: +0.13 %
13:09:29.326 > Reading: 161.73 g | Error: -0.17 %
13:09:30.315 > Reading: 161.84 g | Error: -0.10 %
13:09:31.315 > Reading: 161.69 g | Error: -0.19 %
13:09:32.281 > Reading: 161.74 g | Error: -0.16 %
13:09:33.281 > Reading: 161.84 g | Error: -0.10 %
13:09:34.281 > Reading: 161.61 g | Error: -0.24 %
13:09:35.319 > Reading: 161.56 g | Error: -0.27 %
13:09:36.281 > Reading: 161.57 g | Error: -0.26 %
13:09:37.322 > Reading: 161.85 g | Error: -0.09 %
13:09:38.281 > Reading: 161.75 g | Error: -0.15 %
13:09:39.280 > Reading: 161.61 g | Error: -0.24 %
13:09:40.316 > Reading: 161.83 g | Error: -0.10 %
13:09:41.313 > Reading: 161.59 g | Error: -0.25 %
13:09:42.284 > Reading: 161.53 g | Error: -0.29 %
13:09:43.281 > Reading: 161.79 g | Error: -0.13 %
13:09:44.324 > Reading: 161.85 g | Error: -0.09 %

```

	สิ่งของ	น้ำหนักที่วัดได้ (Kg หรือ mg)	น้ำหนักจริงตาม Spec
1.	เหรียญสิบบาท	8.49×10^{-3}	8.5×10^{-3}
2.	แท็บเล็ต - <u>Tablet</u>	0.37 kg	0.38 kg
3.	มือถือรุ่น <u>Iphone 12</u>	0.16185 kg	0.162 kg
4.	เครื่อง Notebook ยี่ห้อ <u>ACER</u> รุ่น <u>Predator Helios NE01b</u>	2.79 kg	2.80 kg

คำถามท้ายการทดลอง

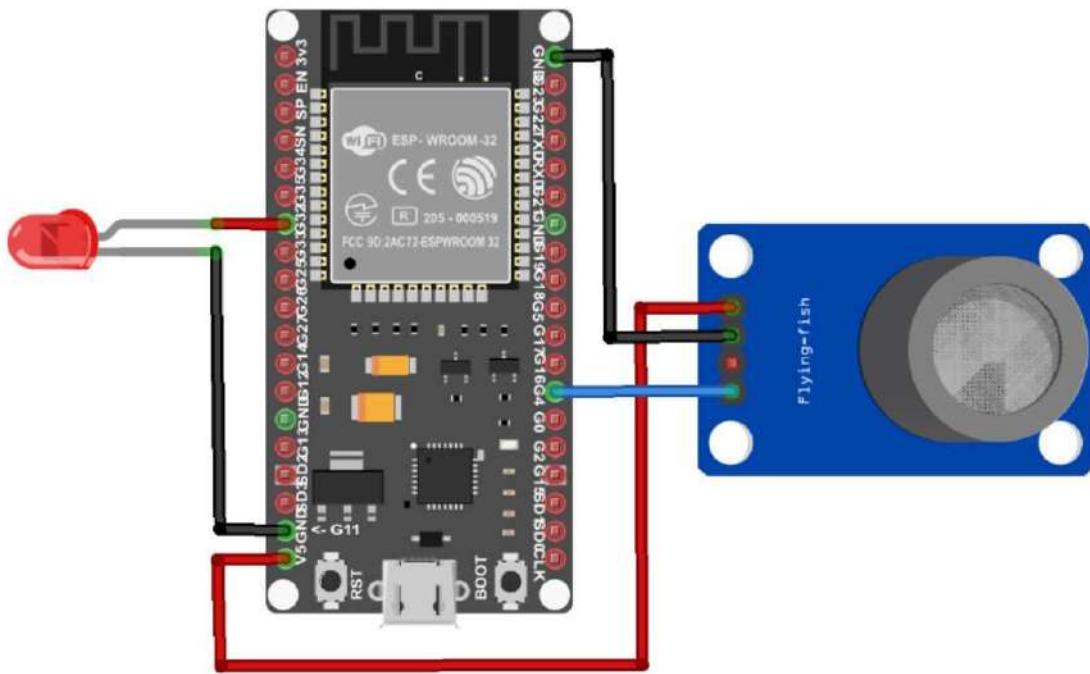
- โมดูล HX711 มีข้อจำกัดในการใช้งานอย่างไร และถ้าต้องการนำมาประยุกต์ใช้เป็นเครื่องชั่งน้ำหนักที่อยู่ในช่วง 30kg-120kg จะต้องมีการต่อเชื่อม Loadcell, HX711 กับ Esp32 อย่างไร จงวาดรูปประกอบ

ขั้นตอนการต่อ HX711

- สัญญาณบวก
- ซองสัญญาณ : HX711 #2 ช่อง คือ Channal A,b
ทำให้สามารถใช้ Loadcell มากสุด 2 ตัว

การทดลองเรื่อง Air Sensor: MQ135

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library และค้นหา MQ135 และติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ทำการต่อโมดูล Air Sensor MQ135 เข้ากับ ESP32 ดังรูป



โดยมีการเชื่อมต่อดังนี้

ESP32	MQ135
Vin	VCC
GPIO ขา 4	Aout
GND	GND
ESP32	LED
GPIO ขา 32	A
GND	K

3. ทำการเขียนและบันทึก Code เพื่อทำให้มีคุณสมารถสแกปอากาศในเรื่องของ CO₂ หรือฝุ่นได้ ที่แสดงบน Serial Monitor พร้อมหน่วยที่ทำการวัด (ทำอย่างได้อย่างหนึ่งเป็นขั้นต่อไป หรือทำทั้งสองจะได้ค่าเพิ่ม) โดยถ้ามีค่าเกินมาตรฐานที่กำหนด LED จะต้องติดค้าง และค่าต่ำกว่ามาตรฐาน LED จะกระพริบ

```

39 // If ppm more than 5000 ppm that means, it's danger ==> Turn ON LED
40 if (ppm > 5000) {
41   digitalWrite(LED_PIN, 1);
42 }
43 // if ppm less than 5000 ppm that means, it's medium or normal ==> blinking LED
44 else if (ppm < 5000) {
45   digitalWrite(LED_PIN, 1);
46   delay(500);
47   digitalWrite(LED_PIN, 0);
48   delay(500);
49 }
50 }
51
52 void showValueQ135() {
53   Serial.print("Q135 RZero: ");
54   Serial.print(rzero);
55   Serial.print("t Corrected RZero: ");
56   Serial.print(correctedRzero);
57   Serial.print("t Resistance: ");
58   Serial.print(resistance);
59   Serial.print("\t ppm ");
60   Serial.print(ppm);
61   Serial.print("\t Corrected PPM: ");
62   Serial.print(correctedPPM);
63   Serial.println("ppm");
64 }
65
66 float AvgValue(){
67   float sum_sample = 0;
68   float mean_sample;
69   for(int i=0;i<SAMPLE_SIZE;i++){
70     sum_sample += ppm;
71   }
72   mean_sample = sum_sample/SAMPLE_SIZE;
73   Serial.print("Mean of PPM : ");
74   Serial.println(mean_sample);
75 }
```

4. ทำการทดสอบโมดูล MQ135 โดยวัดอากาศในสภาพปกติ แล้วบันทึกรูปทดลองและรูปจาก Serial Monitor

ঝঁঝঁ burn sensory

5. ทำการทดสอบโมดูล MQ135 โดยทำการนำแพ็ม่าไปรย์ใกล้ๆ โมดูล หรือนำควันจากธูปมาวางใกล้ๆ โมดูล แล้วบันทึกการทดลองกับรูปจาก Serial Monitor

```

MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61      Corrected RZero: 34.35  Resistance: 21.77      PPM: 3708.23      Corrected PPM: 3598.4lppm
Mean of PPM : 3708.23

```

คำถามท้ายการทดลอง

1. ในการวัดด้วยโมดูล MQ135 จะมีวิธีการอย่างไรที่จะทำให้สามารถวัดในพื้นที่โล่งกว้างได้ จงอธิบาย

ตอบ

ส 2 กอง

1. สามารถใช้ sensor ติดเชือก 2 ตัว
กันหมุน ทำงานในช่องๆ ๆ กันแล้วก็จะไม่เคลื่อนที่กลับไปกลับมาได้
2. สามารถใช้ sensor ติดลงบน
กันหมุน ใบช่องๆ ๆ กันแล้วก็จะไม่เคลื่อนที่

