

เอกสารประกอบการทดลอง
วิชา 01236260 Cyber-Physical System and Sensor

ผู้สอน: ผศ.ดร.วันวิสา ชัชวงษ์
ดร.นันทน์ รุ่งเหมือนฟ้า
ผศ.สรพงษ์ วชิรรัตนพรกุล

นักศึกษาที่ทำการทดลอง

- | | | | |
|----|-------------------------------|------------------------|------------------------------|
| 1. | ชื่อ <u>นางสาวณัฐนันท์</u> | นามสกุล <u>บุญโต</u> | รหัสนักศึกษา <u>67010272</u> |
| 2. | ชื่อ <u>นางสาวศุคนธ์ทิพย์</u> | นามสกุล <u>ด้วงงัน</u> | รหัสนักศึกษา <u>67010960</u> |
| 3. | ชื่อ <u>นายอภิรักษ์</u> | นามสกุล <u>สินธุ์</u> | รหัสนักศึกษา <u>6701018</u> |
| 4. | ชื่อ | นามสกุล | รหัสนักศึกษา |

..

เอกสารประกอบการทดลองนี้ เป็นส่วนหนึ่งของวิชา 01236254 Circuits and Electronics

ภาคเรียนที่ 1 ประจำปีการศึกษา 2566

สาขาวิชาวิศวกรรมระบบไอโอทีและสารสนเทศ

และสาขาวิชาวิศวกรรมระบบไอโอทีและสารสนเทศ และฟิสิกส์อุตสาหกรรม

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การทดลองเรื่อง Voltage Sensor

1. ทำการ Download Driver CP210x จาก [CP210x USB to UART Bridge VCP Drivers - Silicon Labs \(silabs.com\)](https://www.silabs.com/CP210x-USB-to-UART-Bridge-VCP-Drivers) โดยเลือก download ตัว **CP210x Windows Drivers** เมื่อ download เสร็จให้ทำการแตกไฟล์บีบอัดนี้ แล้วเลือกติดตั้ง **CP210xVCPInstaller_x64**
2. 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 แล้วกด OK

3. เข้าไปที่เมนู Tools>Board>Boards Manager จะได้หน้าต่าง Boards Manager ในช่องค้นหาให้ใส่คำว่า ESP32 แล้วทำการติดตั้ง Boards ESP32
4. ทำการเชื่อมต่อบอร์ด ESP32 เข้ากับคอมพิวเตอร์ด้วยสาย USB แล้วที่ Arduino IDE ให้เข้าไปที่ Board>ESP32 แล้วเลือกบอร์ดเป็น ESP32 Dev Module หรือ Node32s และ Com Port ที่เชื่อมต่อ
5. ให้อ่านค่าตัวต้านทานของ Voltage Sensor

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

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

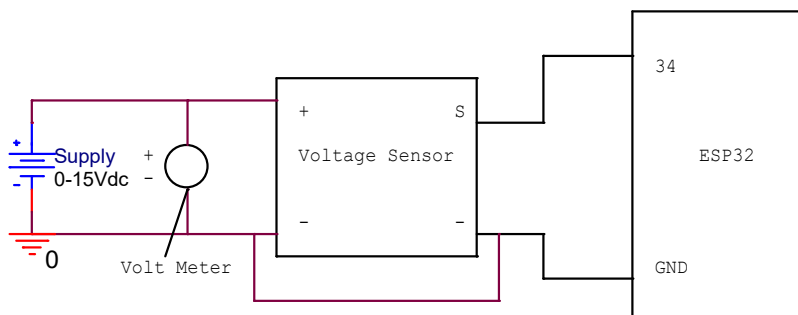
6. จงหาค่าแรงดันสูงสุดที่ Voltage Sensor จะวัดได้ เมื่อมีค่า MaxADCin V เป็น 5.0V กับ 3.0V จากสูตรต่อไปนี้

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

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

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

7. จงต่อแหล่งจ่ายแรงดันไฟฟ้า, โมดูล Voltage Sensor และ ESP32 ตามรูป และป้อนแรงดันไฟฟ้าที่ 5V



8. จงเขียน Code เพื่อทำการอ่านค่าระดับที่ได้จากการวัดแรงดันไฟฟ้าผ่านทาง Voltage Sensor แล้วบันทึกค่าระดับที่ได้

ค่าระดับที่ได้จากการวัดแรงดันไฟฟ้า 5v มีค่า = 4095

9. จงแก้ไข Code ที่เขียนได้ในข้อ 8 เพื่อให้สามารถอ่านค่าระดับ และแรงดันไฟฟ้าที่วัดได้จาก Voltage Sensor

```

sketch_oct8a.ino
24 void loop() {
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. ทำการปรับแหล่งจ่ายแรงดันไฟฟ้า ตามตารางที่ 1 แล้วทำการบันทึกค่าระดับและแรงดันไฟฟ้าที่วัดจาก Voltage Sensor เปรียบเทียบกับมัลติมิเตอร์พร้อมทั้งหาค่า Error จากสูตรต่อไปนี้

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

ตารางที่ 1

Vจากแหล่งจ่าย	ค่าระดับ Analog	Vจาก Sensor	Vจากมิเตอร์	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

สายไฟมีความต้านทาน ส่งผลให้มีการลดลงของแรงดันไฟฟ้า

จึงจะทำให้ค่าระดับเกิดแกว่งขึ้น

การทดลองเรื่อง Current Sensor: ACS712

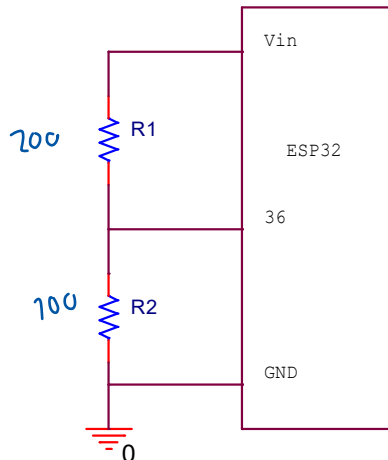
1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library แล้วค้นหา ACS712 แล้วติดตั้ง Lib หรือเข้าเมนู Sketch>Include Library>Add Zip Library
2. ต่อ ESP32 เข้ากับคอมพิวเตอร์ แล้วทำการวัดและบันทึกแรงดันไฟฟ้าที่ขา Vin ของบอร์ด ESP32
แรงดันไฟฟ้าที่ขา Vin ของ ESP32 = 4.82 V
3. ต่อดังตามรูป

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

$$3.3 \pm 0.05 \text{ V} = \left(\frac{270 \Omega}{270 + R_2} \right) 4.82$$

$$R_2 \approx 16.54 \Omega$$

$$R_2 = 100 \Omega$$



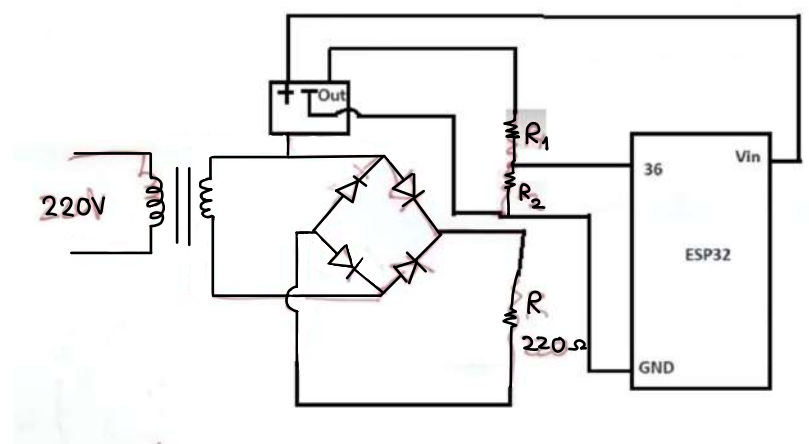
4. จงหาค่า R1, R2 โดยให้ R1 อยู่ในช่วง 220-680Ω และแรงดันไฟฟ้าที่เข้าขา ESP32 จะต้องมีย่านในช่วง 3.29-3.3V (เลือกค่าแรงดันไฟฟ้าค่าเดียวเท่านั้น)

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

5. จงต่อดังตามรูป แล้วใช้มิเตอร์วัดแรงดันไฟฟ้าตรงขา ADC (36 หรือ 34 หรือ 32) ของ ESP32 ว่าตรงกับที่คำนวณในข้อ 4 หรือไม่ บันทึกแรงดันไฟฟ้าที่ได้

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

6. จงต่อดังจร 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/0.185;
  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);
}
```

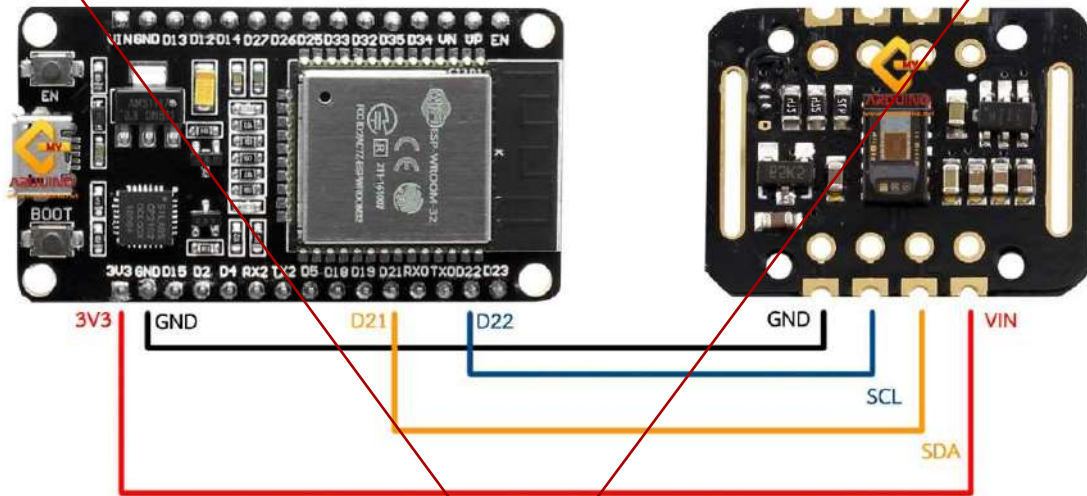
คำถามท้ายการทดลอง

1. จงอธิบายว่าจะทำอย่างไรจึงจะทำให้ ACS712 สามารถวัดกระแสไฟฟ้าต่ำของไฟฟ้า AC ได้อย่างถูกต้อง

ตอบ calibrated ใน order ที่สูงขึ้นเพื่อเพิ่มในค่าของความแม่นยำมากขึ้น

การทดลองเรื่อง Heart Rate sensor: MAX30102

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



Ref: สอนใช้งาน [ESP32 MAX30102](#) เซ็นเซอร์วัดอัตราการเต้นของหัวใจ ชิฟเฟอร์ - นาย [Arduino](#) อุปกรณ์ [Arduino](#) คุณภาพดี ราคาถูก ส่งไว ส่งฟรี ([cybertice.com](#))

3. ทำการเขียนและบันทึก Code เพื่อให้โมดูลสามารถอ่านค่า Heart Rate จากนิ้วมือได้

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) พร้อมทั้งบันทึกภาพผลการทดลองที่ได้

Code

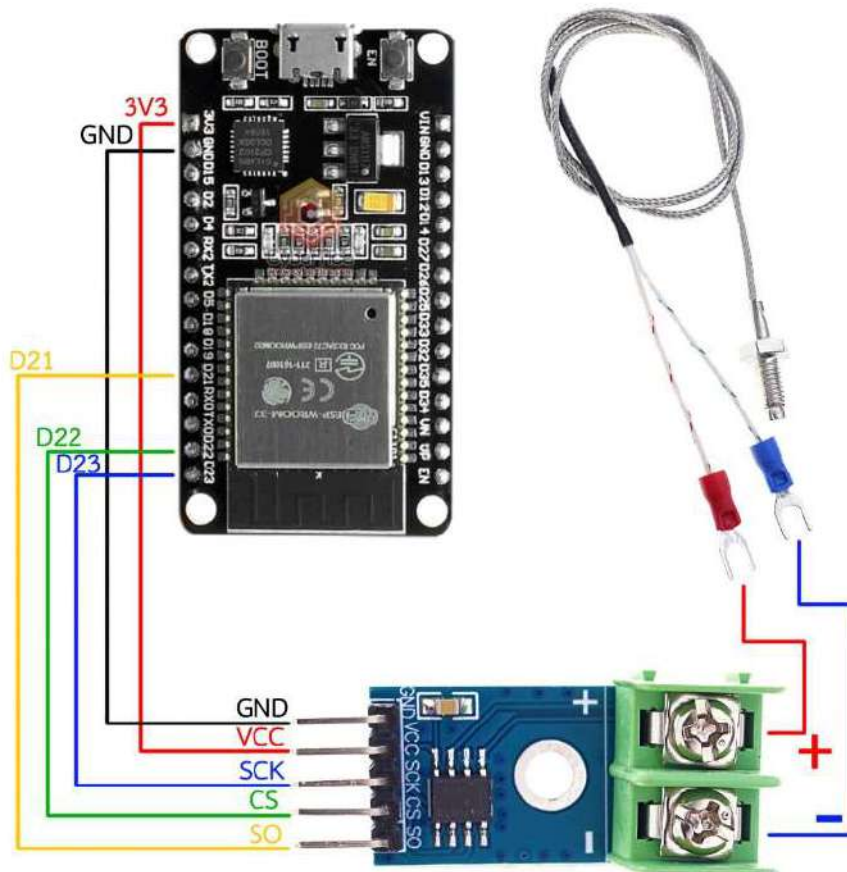
รูปผลการทดลอง

คำถามท้ายการทดลอง

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 แล้วค้นหา MAX6675 แล้วติดตั้ง 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 - จาก Arduino
อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ส่งฟรี (cybertice.com)

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

```

#define CS 15 // Chip Select Pin สำหรับการเลือก Chip

MAX6675 thermocouple(CLK, CS, 50);

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

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

  // Show Celsius temperature แสดงค่าอุณหภูมิในหน่วย องศา
  Serial.print("Temperature [Celsius] : ");
  Serial.println(Celsius);
  Serial.println("-----");
  delay(100);

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

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

```

4. ทำการทดสอบโมดูล MAX6675 บันทึกข้อมูล 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. ทำการทดสอบโมดูล MAX6675 โดยการนำโพรบของโมดูลไปแช่ในน้ำร้อนที่ต้มจนเดือด แล้วบันทึกผลการทดลองกับรูปจาก Serial Monitor

```

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
-----

```

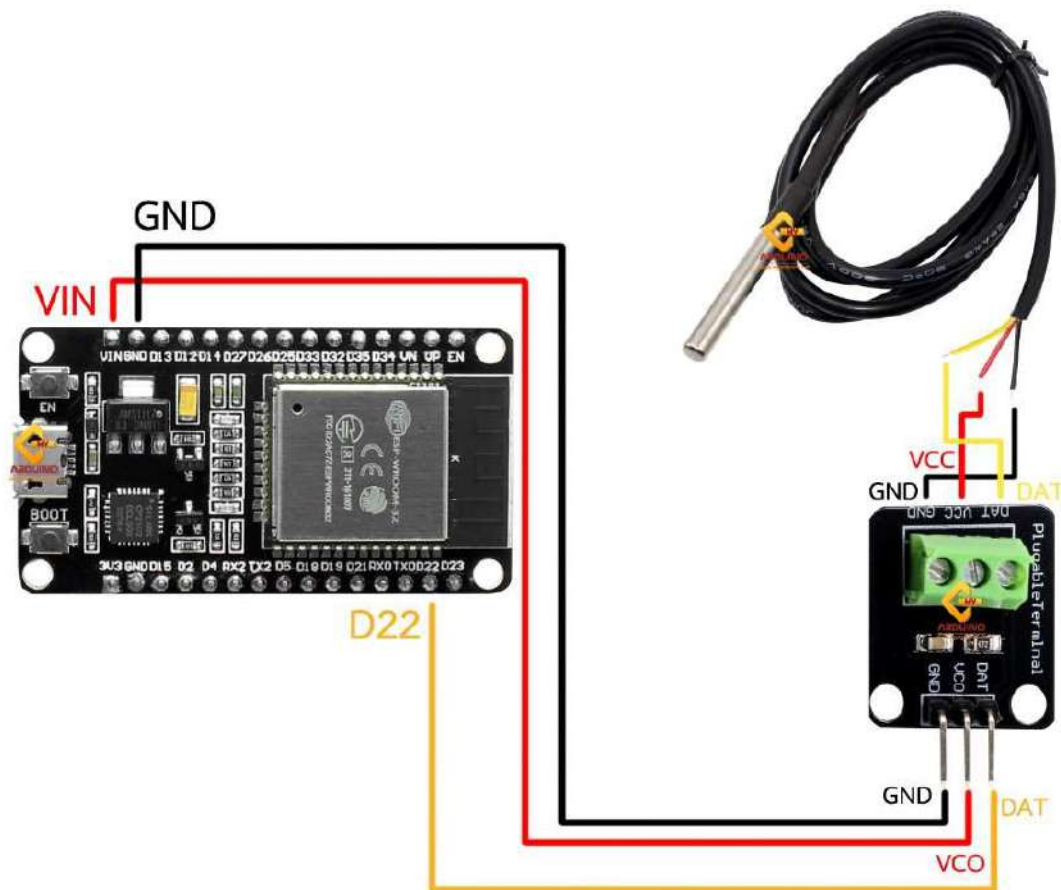


6. ทำการทดสอบโมดูล MAX6675 โดยการนำโพรบของโมดูลไปแช่ในน้ำแข็งครึ่งแก้ว แล้วบันทึกรูปการทดลองกับรูปจาก Serial Monitor

```
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 แล้วค้นหา DS18T20 แล้วติดตั้ง 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 เช่นเซอร์วัดอุณหภูมิในน้ำ - ขา Arduino อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ส่งฟรี (cybertice.com)

3. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถอ่านค่าอุณหภูมิในรูปแบบขององศา °C

```

        break;
    case 0x38:
        type_s = 0;
        break;
    case 0x22:
        type_s = 0;
        break;
    default:
        return;
}
ds.reset();
ds.select(addr);
ds.write(addr, 1); // 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(0x8E); // Read Scratchpad

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

// 16.1 row = (data[1] << 8) | data[0];
if (type_s) {
    row = row << 1; // 9 bit resolution default
    if (data[7] == 0x10) {
        // "count remain" gives full 12 bit resolution
        row = (row & 0xFFF0) + 12 - data[6];
    }
} else {
    byte cfg = (data[4] & 0x0F);
    // at lower res, the low bits are undefined, so let's zero them
    if (cfg == 0x00) row = row & ~7; // 9 bit resolution, 91.75 ms
    else if (cfg == 0x01) row = row & ~3; // 10 bit res, 187.5 ms
    else if (cfg == 0x02) row = row & ~1; // 11 bit res, 375 ms
    // default is 12 bit resolution, 750 ms conversion time
}

```

4. ทำการทดสอบโมดูล DS18T20 โดยการวัดอุณหภูมิในอากาศ บันทึกรูป Serial Monitor

```

Message (Enter to send message to 'Node32s' on 'COM5

Temperature: 29.37 °C
Temperature: 29.37 °C
Temperature: 29.37 °C
Temperature: 29.06 °C
Temperature: 27.87 °C
Temperature: 26.81 °C
Temperature: 26.00 °C
Temperature: 25.44 °C
Temperature: 25.00 °C
Temperature: 24.56 °C
Temperature: 24.25 °C
Temperature: 23.94 °C

```

5. ทำการทดสอบโมดูล DS18T20 โดยการนำโพรบของโมดูลไปแช่ในน้ำร้อนที่ต้มจนเดือด แล้วบันทึกผลการทดลองกับรูปจาก Serial Monitor

```

Temperature = 90.57 Celsius, 195.03 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 = 91.62 Celsius, 196.93 Fahrenheit
Temperature = 91.62 Celsius, 196.93 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 โดยการนำโพรบของโมดูลไปแช่ในน้ำแข็งครึ่งแก้ว 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
```

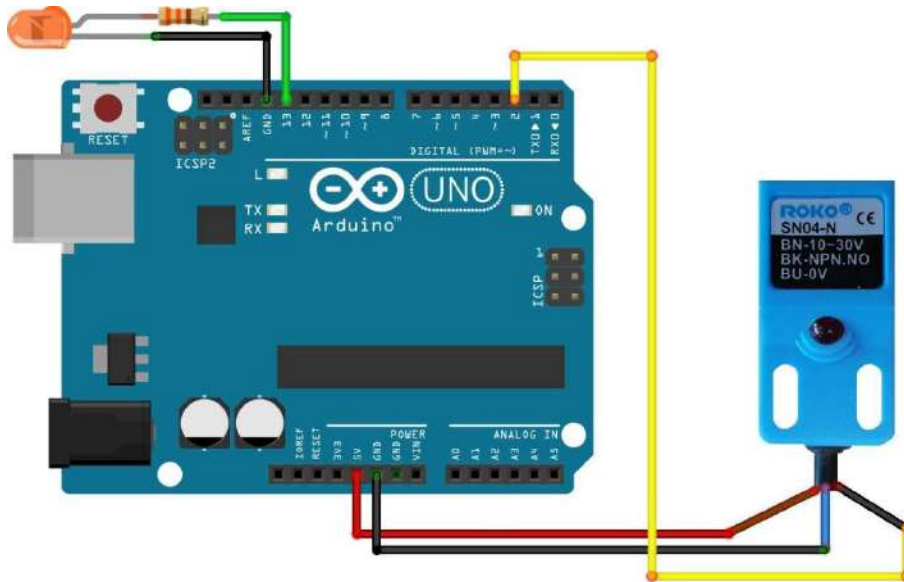
คำถามท้ายการทดลอง

1. โมดูล MAX6675 และโมดูล DS18T20 มีความละเอียดในการวัดอุณหภูมิเป็นอย่างไร และแต่ละโมดูล เหมาะกับการนำไปใช้งานด้านไหน จงอธิบาย

Module MAX6675 รับสัญญาณ Analog ส่วนโมดูล sensor ให้ความไวต่อ noise แต่ DS18B20 วัดแบบ Digital sensor

การทดลองเรื่อง Metal Inductive Sensor: SN04-N

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



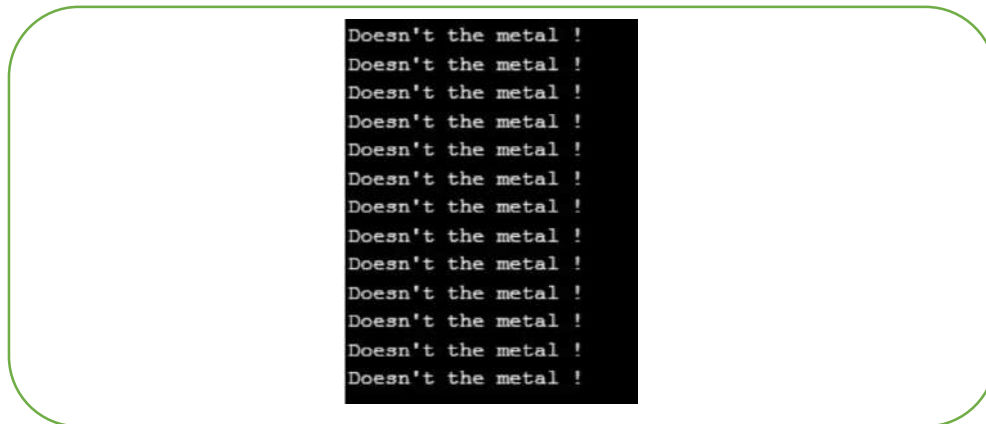
2. ทำการเขียนและบันทึก Code เพื่อทำให้โมดูลสามารถตรวจจับโลหะได้

```
#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. ทำการทดสอบโมดูล SN04-N โดยการนำโลหะ เช่นคลิปหนีบกระดาษ หรือเข็มหมุด มาวางใกล้โมดูล ตามระยะในตาราง แล้วบันทึกผลลงในตารางและ บันทึกรูป Serial Monitor

	โลหะ	ระยะ 1 ซม.	ระยะ 8 มม.	ระยะ 4 มม.	ระยะ 1 มม.
1.	คลิปหนีบกระดาษ	✗	✗	✗	✓
2.	เข็มหมุด	✗	✗	✗	✓
3.	ตัวต้านทานปรับค่าได้	✓	✓	✓	✓
4.	เหรียญบาท	✗	✓	✓	✓
5.	เหรียญสิบบาท	✓	✓	✓	✓
6.	หัวเข็มขัดโลหะ	✓	✓	✓	✓

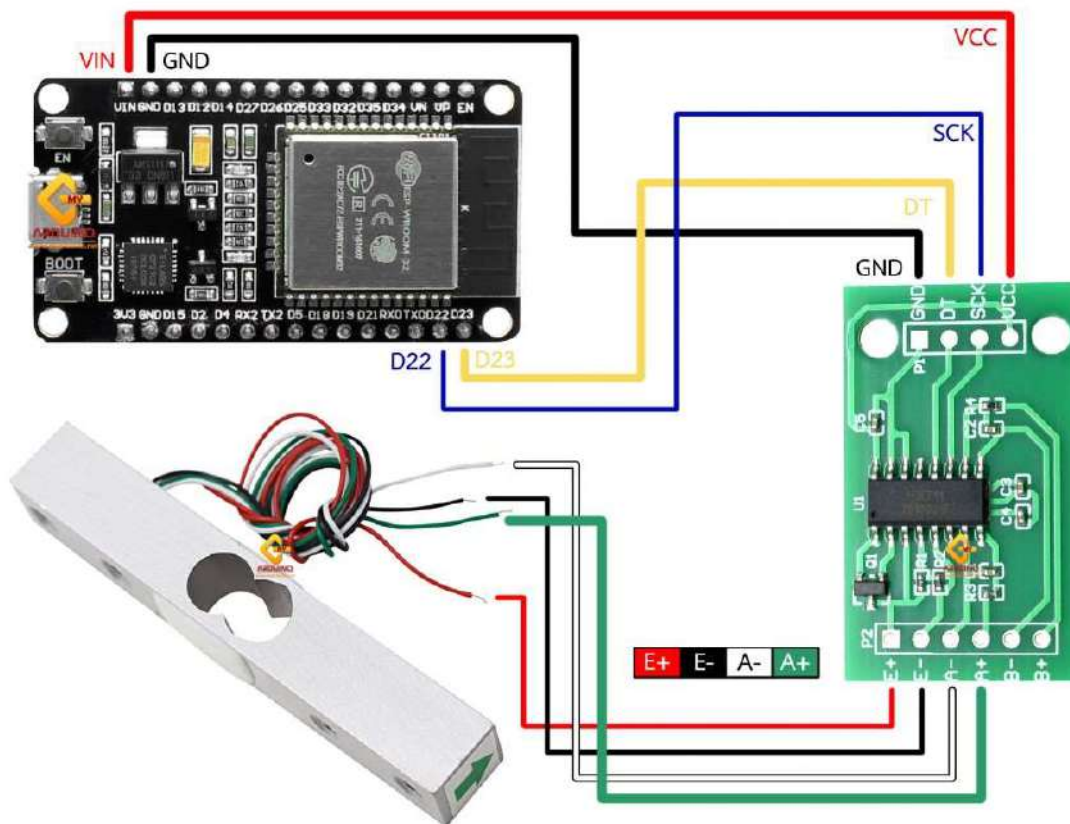
คำถามท้ายการทดลอง

1. โมดูล SN04-N สามารถตรวจจับโลหะได้ที่ระยะใด ขนาดของโลหะมีผลต่อการตรวจจับหรือไม่ และมีปัจจัยอื่นหรือไม่ที่ทำให้สามารถตรวจจับโลหะในระยะที่ไกลขึ้น พร้อมทั้งบอกการนำไปประยุกต์ใช้งานมาอย่างน้อย 3 อย่าง

SN04-N สามารถตรวจจับโลหะได้ระยะที่: บ. เท่ากับ 1-3 ฟุต ผลกระทบ ขนาดมีผลต่อการตรวจจับ
ไม่ควรวางไว้ ใกล้กับ: แสงไฟฟ้าแรงสูง เพราะ จะเกิดผลกระทบจากคลื่นวิทยุแม่เหล็ก
ในชั้นนี้

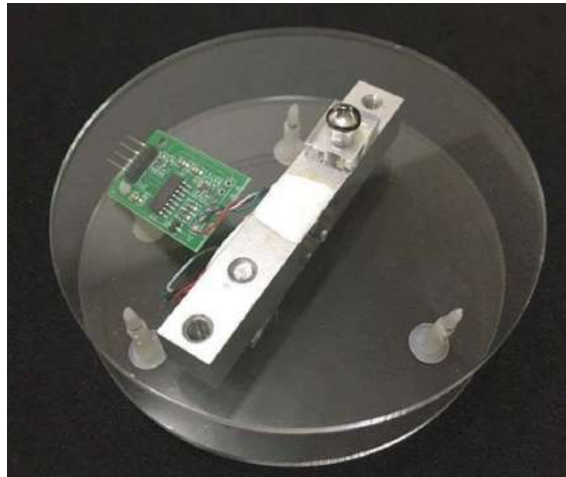
การทดลองเรื่อง Load Cell Sensor: HX711

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



Ref: สอนใช้งาน ESP32 ทำเครื่องชั่งน้ำหนัก เซ็นเซอร์รับแรงกด วัดน้ำหนัก load cell HX711 - จาก Arduino อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ส่งฟรี (cybertice.com)

3. ให้นักศึกษาทำการประกอบฐานการชั่งน้ำหนัก โดยให้มีการวาง Loadcell ไว้ที่ใต้ฐานชั่งน้ำหนักตามรูปตัวอย่าง โดยสามารถออกแบบและใช้วัสดุอะไรก็ได้ตามสะดวก



4. ทำการเขียนและบันทึก Code เพื่อให้โมดูลสามารถวัดค่าน้ำหนักที่ชั่งได้ในหน่วยของ Kg กับ mg โดยสามารถตัดสินใจได้เองว่าของที่ชั่งมีน้ำหนักเป็น kg หรือ mg

```

1 // HX711 library
2
3
4 // ปรับเปลี่ยนการเชื่อมต่อขั้วต่อ
5 uint8_t dataPin = 23;
6 uint8_t clockPin = 22;
7
8
9 // 1. กำหนดค่าน้ำหนักจริง (known weight) ที่ใช้ในการเปรียบเทียบ
10 float known_weight_g = 162.0; // น้ำหนักอ้างอิง X กรัม
11
12 void setup()
13 {
14   Serial.begin(115200);
15   Serial.println();
16   Serial.println(_FILE_);
17   Serial.print("HX711_LIB_VERSION: ");
18   Serial.println(HX711_LIB_VERSION);
19   Serial.println();
20
21   scale.begin(dataPin, clockPin);
22
23   // ใส่ค่า Calibration Factor ที่กำหนดมาสำหรับน้ำหนัก X กรัม
24   scale.set_scale(-441.191345);
25
26   // ตั้งค่าศูนย์ของเครื่องชั่ง
27   scale.tare(20);
28   Serial.println("Scale is ready. Place the X g weight on it.");
29 }

```

5. ทำการทดสอบโมดูล HX711 โดยยังไม่ต้องมีสิ่งของวางบนฐานชั่งน้ำหนัก แล้วบันทึกรูปแบบ 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



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

คำถามท้ายการทดลอง

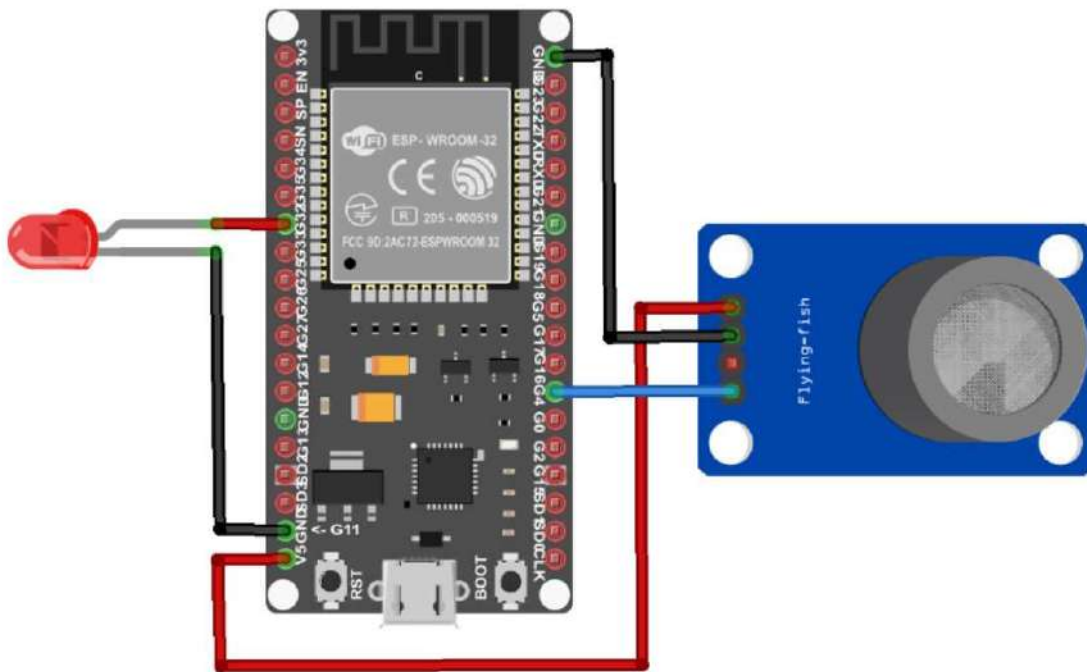
- โมดูล HX711 มีข้อจำกัดในการใช้งานอย่างไร และถ้าต้องการนำมาประยุกต์ใช้เป็นเครื่องชั่งน้ำหนักที่อยู่ในช่วง 30kg-120kg จะต้องมีการต่อเชื่อม Loadcell, HX711 กับ Esp32 อย่างไร จงวาดรูปประกอบ

ข้อจำกัดของโมดูล HX711

- สัญญาณรบกวน
- ช่วงสัญญาณ : HX711 มี 2 ช่อง คือ Channel A, b
ทำในลักษณะต่อ Loadcell มากสุด 2 ตัว

การทดลองเรื่อง Air Sensor: MQ135

1. เปิดโปรแกรม Arduino IDE ทำการ Add library ด้วยการเข้าเมนู Sketch>Include Library>Manager Library แล้วค้นหา MQ135 แล้วติดตั้ง 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 เพื่อให้โมดูลสามารถสภาพอากาศในเรื่องของ CO₂ หรือฝุ่นได้ ที่แสดงบน Serial Monitor พร้อมหน่วยที่ทำการวัด (ทำอย่างใดอย่างหนึ่งเป็นขั้นต่ำ หรือทำทั้งสองจะได้คะแนนเพิ่ม) โดยถ้ามีค่าเกินมาตรฐานที่กำหนด LED จะต้องติดค้าง และค่าต่ำกว่ามาตรฐาน LED จะกระพริบ

```

38
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 ShowValueMQ135() {
53   Serial.print("MQ135 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

MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm
MQ135 RZero: 147.69	Corrected RZero: 149.27	Resistance: 94.39	PPM: 61.47	Corrected PPM: 65.59ppm

ยังไม่ได้ burn sensor

5. ทำการทดสอบโมดูล MQ135 โดยทำการนำแป้งมาโปรยใกล้ๆ โมดูล หรือนำควันจากธูปมาวางใกล้ๆ โมดูล แล้วบันทึกรูปการทดลองกับรูปจาก Serial Monitor

```

MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm
Mean of PPM : 3708.23
MQ135 RZero: 34.61    Corrected RZero: 34.35 Resistance: 21.77    PPM: 3708.23    Corrected PPM: 3598.41ppm

```

คำถามท้ายการทดลอง

1. ในการวัดด้วยโมดูล MQ135 จะมีวิธีการอย่างไรที่จะทำให้สามารถวัดในพื้นที่โล่งกว้างได้ จงอธิบาย

ตอบ

ส 2 กงพี

1. สามารถใช้ sensor ๒ตัวประกอบ ๑ตัว

กำหนดพท. ใช้งานในจัดงาน ดูแหล่งกำเนิด Gas ตัวบริเวณใกล้แสงกำหนด

2. สามารถใช้ sensor ได้นาสระตัว

กำหนดพท. ใช้งาน และตัดหัวลักษณะ grid

