

## 1. LAB การทดลองเรื่อง Voltage Sensor (ไม่ได้ใช้ Library)

โค้ดที่นำไปใช้งาน

```
const float alpha = 0.6195;  
  
const float beta = 0.661;  
  
const int pin_v = 34;  
  
void setup() {  
  
    Serial.begin(115200);  
  
}  
  
void loop() {  
  
    float adc_voltage = analogRead(pin_v);  
  
    float voltage = ((adc_voltage * 3.3) / (float)(1 << 9));  
  
    float calibrated_voltage = (alpha * voltage) + beta;  
  
  
    Serial.print("ADC_v value : ");  
  
    Serial.print(adc_voltage, 2);  
  
    Serial.print(" | Voltage : ");  
  
    Serial.print(voltage, 3);  
  
    Serial.print(" | Voltage Calibrated : ");  
  
    Serial.print(calibrated_voltage);  
  
    Serial.println(" V");  
  
    delay(1000);  
  
}
```

## 2.การทดลองเรื่อง Current Sensor: ACS712 (ไม่ได้ใช้ Library)

```
#include <math.h>

#define CURRENT_SENSOR_PIN 35

#define SAMPLE_SIZE 100

const float alpha_poly = -0.7583;

const float beta_poly = 2.2562;

const float gamma_poly = -0.0299;

void setup() {

    Serial.begin(9600);

    pinMode(CURRENT_SENSOR_PIN, INPUT);

}

void loop() {

    long sum_adc = 0;

    for (int i = 0; i < SAMPLE_SIZE; i++) {

        sum_adc += analogRead(CURRENT_SENSOR_PIN);

        delayMicroseconds(50);

    }

    float average_adc = (float)sum_adc / SAMPLE_SIZE;

    Serial.print("Average ADC : ");

    Serial.println(average_adc);

    float voltage = (average_adc / 4095.0) * 3.30;

    float set_zero_voltage = voltage - 1.525;

    float current_sensor = set_zero_voltage / 0.185;

    float polynomial_calibration = (alpha_poly * current_sensor * current_sensor) + (beta_poly * current_sensor) + gamma_poly;

    Serial.print("Current sensor (before cal): ");

    Serial.println(current_sensor, 3);

    Serial.print("Calibrated Polynomial Current Sensor: ");

    Serial.println(polynomial_calibration, 3);
```

```

Serial.println("-----");
delay(700);
}

```

### 3,การทดสอบเรื่อง K-type Thermocouple sensor: MAX6675

การติดตั้ง Library กด Sketch → Include Library → Manage Libraries ค้นหา max6675 และติดตั้ง

```
#include "max6675.h"
```

```
#define SO 19
```

```
#define CLK 14
```

```
#define CS 15
```

```
MAX6675 thermoCouple(CLK, CS, SO);
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  float Celsius = thermoCouple.readCelsius();
```

```
  float Fahrenheit = thermoCouple.readFahrenheit();
```

```
  float Kelvin = Celsius + 273.15;
```

```
  Serial.print("Temperature [Celsius] : ");
```

```
  Serial.println(Celsius);
```

```
  Serial.println("-----");
```

```
  delay(100);
```

```
  Serial.print("Temperature [Fahrenheit] : ");
```

```
  Serial.println(Fahrenheit);
```

```
  Serial.println("-----");
```

```
  delay(100);
```

```
  Serial.print("Temperature [Kelvin] : ");
```

```
  Serial.println(Kelvin);
```

```
  Serial.println("-----");
```

```
delay(700);

}
```

#### 4.การทดลองเรื่อง Full Waterproof Temperature Sensor: DS18T20

การติดตั้ง Library กด Sketch → Include Library → Manage Libraries ค้นหา OneWire , DallasTemperature และติดตั้ง

```
#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE_WIRE_BUS 32

const float alpha = 1.0444;

const float beta = -1.3354;

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

void setup() {

    Serial.begin(9600);

    sensors.begin();

}

void loop() {

    sensors.requestTemperatures();

    float tempC = sensors.getTempCByIndex(0);

    if (tempC != DEVICE_DISCONNECTED_C) {

        Serial.print("Temperature: ");

        Serial.print(tempC);

        Serial.println(" °C");

        float calibrated_tempC = (alpha * tempC) + beta;

        Serial.print("Temperature_Calibrated: ");

        Serial.print(calibrated_tempC);

        Serial.println(" °C");

    } else {

        Serial.println("Error: Could not read temperature data");
    }
}
```

```
}

delay(1000);

}
```

## 5.การทดลองเรื่อง Metal Inductive Sensor: SN04-N (ไม่ได้ใช้ Library)

```
#define LED 34

#define SENSOR 35

int sensor;

bool sensor_detect;

void setup() {

Serial.begin(9600);

pinMode(SENSOR, INPUT);

pinMode(LED, OUTPUT);

}

void loop() {

sensor = digitalRead(SENSOR);

if(sensor == 1){

Serial.println("Found the metal !");

}

if(sensor == 0){

Serial.println("Doesn't the metal !");

}

delay(500);

}
```

## 6.การทดลองเรื่อง Load Cell Sensor: HX711

การติดตั้ง Library กด Sketch → Include Library → Manage Libraries ค้นหา HX711 และติดตั้ง

```
#include "HX711.h"

HX711 myScale;

// adjust pins if needed.

uint8_t dataPin = 23;
uint8_t clockPin = 22;

void setup()
{
    Serial.begin(115200);
    Serial.println();
    Serial.println(__FILE__);
    Serial.print("HX711_LIB_VERSION: ");
    Serial.println(HX711_LIB_VERSION);
    Serial.println();
    myScale.begin(dataPin, clockPin);
}

void loop()
{
    calibrate();
}

void calibrate()
{
    Serial.println("\n\nCALIBRATION\n=====");
    Serial.println("remove all weight from the loadcell");
    // flush Serial input
    while (Serial.available()) Serial.read();
    Serial.println("and press enter\n");
```

```

while (Serial.available() == 0);

Serial.println("Determine zero weight offset");

// average 20 measurements.

myScale.tare(20);

int32_t offset = myScale.get_offset();

Serial.print("OFFSET: ");

Serial.println(offset);

Serial.println();

Serial.println("place a weight on the loadcell");

// flush Serial input

while (Serial.available()) Serial.read();

Serial.println("enter the weight in (whole) grams and press enter");

uint32_t weight = 0;

while (Serial.peek() != '\n')

{

    if (Serial.available())

    {

        char ch = Serial.read();

        if (isdigit(ch))

        {

            weight *= 10;

            weight = weight + (ch - '0');

        }

    }

}

Serial.print("WEIGHT: ");

Serial.println(weight);

myScale.calibrate_scale(weight, 20);

float scale = myScale.get_scale();

```

```

Serial.print("SCALE: ");
Serial.println(scale, 6);
Serial.print("\nuse scale.set_offset()");
Serial.print(offset);
Serial.print("); and scale.set_scale()");
Serial.print(scale, 6);
Serial.print(");\n");
Serial.println("in the setup of your project");
Serial.println("\n\n");
}

```

แล้วนำค่า Scale ที่ได้ใส่เก็บนี้

```

#include "HX711.h"

HX711 scale;

uint8_t dataPin = 23;
uint8_t clockPin = 22;
float known_weight_g = 606.0; // น้ำหนักอ้างอิง X กรัม

void setup()
{
    Serial.begin(115200);

    Serial.println();
    Serial.println(__FILE__);

    Serial.print("HX711_LIB_VERSION: ");

    Serial.println(HX711_LIB_VERSION);

    Serial.println();

    scale.begin(dataPin, clockPin);

    scale.set_scale(-400.374268); // นำค่า Scale ที่ได้ใส่เก็บนี้

    scale.tare(20);

    Serial.println("Scale is ready. Place the X g weight on it.");
}

```

```
}

void loop()
{
    if (scale.is_ready())
    {
        float measured_weight = scale.get_units(1);

        float error_percent = ((measured_weight - known_weight_g) / known_weight_g) * 100.0;
        Serial.print("Reading: ");
        Serial.print(measured_weight, 2);
        Serial.print(" g");
        Serial.print(" | Error: ");

        if (error_percent >= 0)
        {
            Serial.print("+");
        }

        Serial.print(error_percent, 2);
        Serial.println(" %");

        delay(1000);
    }
}
```

## 7. การทดลองเรื่อง Air Sensor: MQ135

การติดตั้ง Library กด Sketch → Include Library → Manage Libraries ค้นหา MQ135 และติดตั้ง

```
#include <MQ135.h>

#define PIN_MQ135 33
#define LED_PIN 23
#define SAMPLE_SIZE 100

MQ135 mq135_sensor(PIN_MQ135);

float temperature = 21.0;
float humidity = 25.0;
float rzero = mq135_sensor.getRZero();
float correctedRZero = mq135_sensor.getCorrectedRZero(temperature, humidity);
float resistance = mq135_sensor.getResistance();
float ppm = mq135_sensor.getPPM();
float correctedPPM = mq135_sensor.getCorrectedPPM(temperature, humidity);

void setup() {
    Serial.begin(9600);
    pinMode(LED_PIN, OUTPUT);
}

void LightOutWhenCOexceed() {
    if (ppm > 5000) {
        digitalWrite(LED_PIN, 1);
    } else if (ppm < 5000) {
        digitalWrite(LED_PIN, 1);
        delay(500);
        digitalWrite(LED_PIN, 0);
        delay(500);
    }
}

void ShowValueMQ135() {
```

```

Serial.print("MQ135 RZero: ");
Serial.print(rzero);
Serial.print("\t Corrected RZero: ");
Serial.print(correctedRZero);
Serial.print("\t Resistance: ");
Serial.print(resistance);
Serial.print("\t PPM: ");
Serial.print(ppm);
Serial.print("\t Corrected PPM: ");
Serial.print(correctedPPM);
Serial.println("ppm");
}

float AVGvalue() {
    float sum_sample = 0;
    float mean_sample;
    for (int i = 0; i < SAMPLE_SIZE; i++) {
        sum_sample += ppm;
    }
    mean_sample = sum_sample / 100.0;
    Serial.print("Mean of PPM : ");
    Serial.println(mean_sample);
}

void loop() {
    ShowValueMQ135();
    LightOutWhenCOexceed();
    AVGvalue();
}

```