

ใบงานทดสอบการใช้งาน Ultrasonic sensor

1. Ultrasonic sensor (อ่านว่า: อัลตราโซนิกเซ็นเซอร์) คือ อุปกรณ์สำหรับวัดระดับหรือระยะทางชนิดหนึ่งโดยใช้คลื่น Ultrasonic ซึ่งอาศัยหลักการสะท้อนของคลื่นความถี่สูง Ultrasonic โดยอุปกรณ์จะปล่อยคลื่น Ultrasonic ให้กระทบกับวัตถุ จากนั้นรอคลื่น Ultrasonic สะท้อนกลับมาที่เซ็นเซอร์เพื่อคำนวณหาระยะทางที่วัดได้ นอกเหนือจาก Ultrasonic sensor แล้ว ยังมีเซ็นเซอร์ชนิดอื่นๆอีกที่ใช้ในการวัดระยะได้แก่ Radar sensor, Hydrostatic sensor เป็นต้น



Ultrasonic sensor

#Checkpoint1 ระบบตรวจจับระยะทางและแสดงค่าผ่าน Serial monitor

จงเขียนโค้ด Arduino เพื่อควบคุมการทำงานของ Ultrasonic Sensor กับ หลอดไฟ LED โดยมีเงื่อนไขว่า :

- เมื่อมีวัตถุอยู่ห่างจากเซ็นเซอร์ น้อยกว่า 50 เซนติเมตร ให้ หลอดไฟ LED ติด
- เมื่อวัตถุอยู่ห่างออกไป มากกว่า หรือเท่ากับ 50 เซนติเมตร ให้ หลอดไฟ LED ดับ

และต่อวงจรการเชื่อมต่ออุปกรณ์บนกระดานหรือกระดาดช โดยใช้อุปกรณ์ดังนี้ :

- Arduino Uno
- Ultrasonic Sensor
- สาย Jumper

คำใบ้ Checkpoint1

ตัวอย่างCode

```
// ประกาศหมายเลขขาของเซนเซอร์และ LED
int trigPin = 11;    // ขาที่ใช้ส่งเสียงออก
int echoPin = 10;    // ขาที่รับเสียงสะท้อนกลับ

void setup() {
    pinMode(trigPin, OUTPUT); // ตั้งค่าให้ขา trig เป็นขาออก
    pinMode(echoPin, INPUT);  // ตั้งค่าให้ขา echo เป็นขารับข้อมูล
    Serial.begin(9600);       // เริ่มต้นการส่งข้อมูลไปยังหน้าจอ Serial Monitor
}

void loop() {
    // เริ่มการวัดระยะ
    long duration;
    int distance;

    digitalWrite(trigPin, LOW);    // เริ่มจากส่งสัญญาณต่ำก่อน
    delayMicroseconds(2);          // รอ 2 ไมโครวินาที

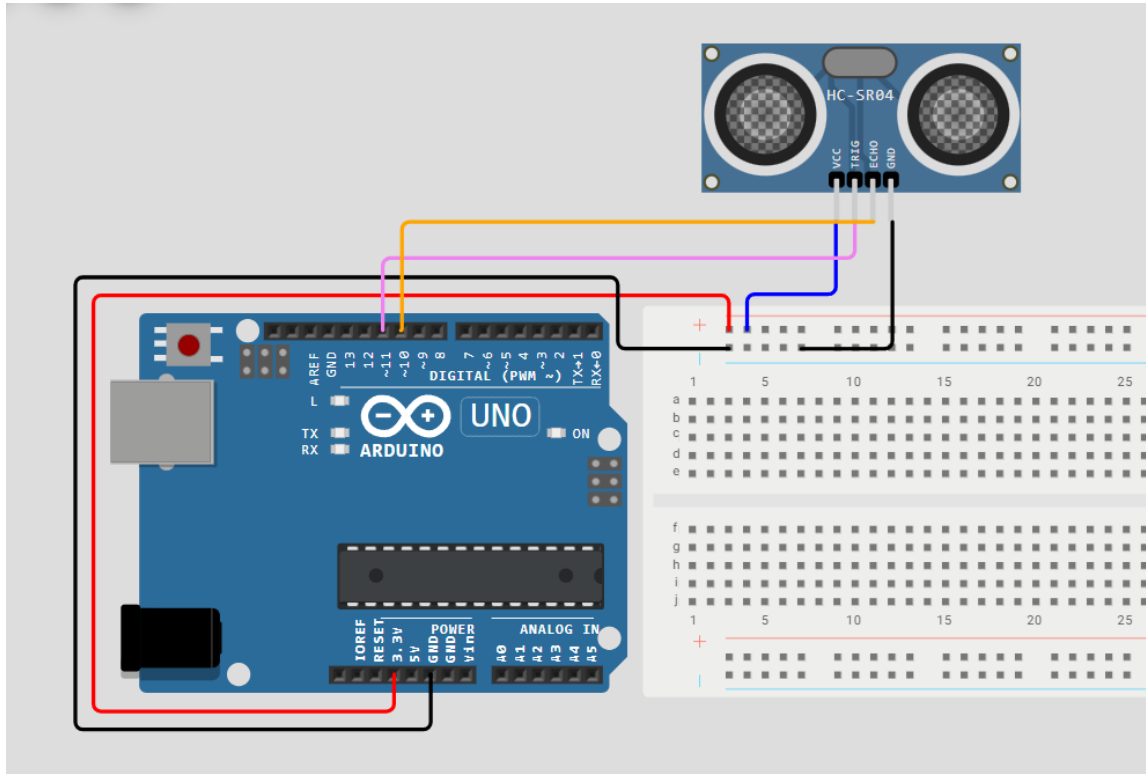
    digitalWrite(trigPin, HIGH);   // ส่งสัญญาณสูง (คลื่นเสียง)
    delayMicroseconds(10);         // รอ 10 ไมโครวินาที
    digitalWrite(trigPin, LOW);    // หยุดส่งสัญญาณ

    duration = pulseIn(echoPin, HIGH); // วัดระยะเวลาที่เสียงสะท้อนกลับมา
    distance = duration / 58;         // แปลงระยะเวลาเป็นระยะทาง (หน่วยเซนติเมตร)

    // แสดงผลระยะทางบน Serial Monitor
    Serial.print("ระยะทาง:");
    Serial.print(distance);
    Serial.println(" ซม.");

    delay(100); // รอ 0.1 วินาที แล้วเริ่มวัดใหม่
}
```

ตัวอย่างการต่อวงจร



น้องๆสามารถเปลี่ยนPinเป็นตัวอื่นได้เลยหากสายไฟยาวไม่เพียงพอ

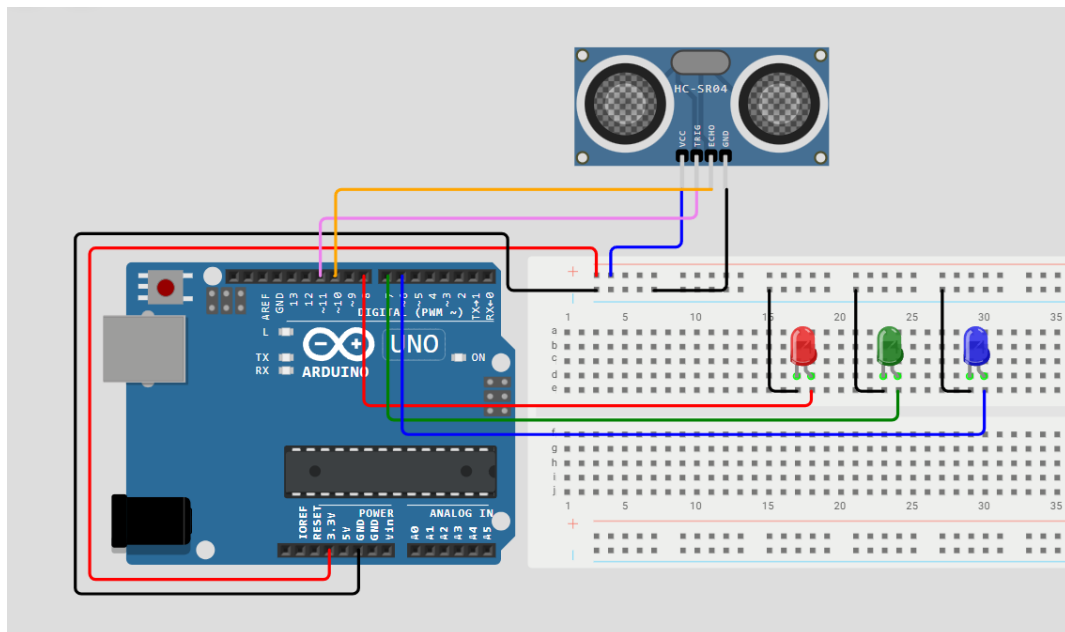
(อย่าลืมเปลี่ยนCodeตรงกำหนดค่าของขาด้วย)

#Checkpoint2 ระบบเตือนระยะ 3 ระดับ

จงเขียนโปรแกรม Arduino โดยใช้เซนเซอร์ HC-SR04 และหลอดไฟ LED 3 ดวงเพื่อแสดงระดับความใกล้ของวัตถุตามเงื่อนไขต่อไปนี้:

ระยะทาง (เซนติเมตร)	การทำงานของ LED
< 10 ซม.	LED ดวงที่ 1 สว่าง
10-24 ซม.	LED ดวงที่ 2 สว่าง
25-49 ซม.	LED ดวงที่ 3 สว่าง
≥ 50 ซม.	ทุกดวงดับ

ตัวอย่างการต่อวงจร



น้องๆสามารถเปลี่ยนPinเป็นตัวอื่นได้เลยหากสายไฟยาวไม่เพียงพอ

(อย่าลืมเปลี่ยนCodeตรงกำหนดค่าของขาด้วย)

คำใบ้ Checkpoint2

ตัวอย่าง Code

```
// ประกาศหมายเลขขาของเซนเซอร์และ LED
int trigPin = 11; // ขาที่ใช้ส่งเสียงออก
int echoPin = 10; // ขาที่รับเสียงสะท้อนกลับ
int ledPin1 = 8; // ขาที่ต่อกับหลอดไฟ LED
int ledPin2 = 7; // ขาที่ต่อกับหลอดไฟ LED
int ledPin3 = 6; // ขาที่ต่อกับหลอดไฟ LED

void setup() {
  pinMode(trigPin, OUTPUT); // ตั้งค่าให้ขา trig เป็นขาออก
  pinMode(echoPin, INPUT); // ตั้งค่าให้ขา echo เป็นขารับข้อมูล
  pinMode(ledPin1, .....); // ตั้งค่าให้ขา LED เป็นขาออก
  pinMode(ledPin2, .....); // ตั้งค่าให้ขา LED เป็นขาออก
  pinMode(ledPin3, .....); // ตั้งค่าให้ขา LED เป็นขาออก
  Serial.begin(9600); // เริ่มต้นการส่งข้อมูลไปยังหน้าจอ Serial Monitor
}

void loop() {
  // เริ่มการวัดระยะ
  long duration;
  int distance;
  digitalWrite(trigPin, LOW); // เริ่มจากส่งสัญญาณต่ำก่อน
  delayMicroseconds(2); // รอ 2 ไมโครวินาที
  digitalWrite(trigPin, HIGH); // ส่งสัญญาณสูง (คลื่นเสียง)
  delayMicroseconds(10); // รอ 10 ไมโครวินาที
  digitalWrite(trigPin, LOW); // หยุดส่งสัญญาณ

  duration = pulseIn(echoPin, HIGH); // วัดระยะเวลาที่เสียงสะท้อนกลับมา
  distance = duration / 58; // แปลงระยะเวลาเป็นระยะทาง (หน่วยเซนติเมตร)
  // แสดงผลระยะทางบน Serial Monitor
  Serial.print("ระยะทาง:");
  Serial.print(distance);
  Serial.println(" ซม.");

  if (distance < 10) {
    // วัดอยู่ใกล้มาก → เปิด led1 สีแดง
    digitalWrite(ledPin1, .....);
    digitalWrite(ledPin2, .....);
    digitalWrite(ledPin3, .....);
  }
}
```

คำใบ้ Checkpoint2

ตัวอย่าง Code (ต่อ)

```
} else if (distance < 25) {  
    // วัตถุอยู่ใกล้ปานกลาง → เปิด led2 สีเหลือง  
    digitalWrite(ledPin1, .....);  
    digitalWrite(ledPin2, .....);  
    digitalWrite(ledPin3, .....);  
  
} else if (distance < 50) {  
    // วัตถุเริ่มเข้าใกล้ → เปิด led3 สีเขียว  
    digitalWrite(ledPin1, .....);  
    digitalWrite(ledPin2, .....);  
    digitalWrite(ledPin3, .....);  
} else {  
    // วัตถุอยู่ไกล → ปิดทุกดวง  
    digitalWrite(ledPin1, .....);  
    digitalWrite(ledPin2, .....);  
    digitalWrite(ledPin3, .....);  
}  
delay(100); // รอ 0.1 วินาที แล้วเริ่มวัดใหม่  
}
```

ทดสอบการใช้งาน Servo Motor

2. SG90 คือชื่อของเซอร์โวมอเตอร์ (Servo Motor) ขนาดเล็กที่มีความนิยมในหมู่นักพัฒนาและผู้สนใจในโปรเจกต์ DIY (Do It Yourself). SG90 ทำงานในแรงดัน 4.8-6V และมีความสามารถในการหมุนได้ในช่วง 0-90 องศาด้วยความแม่นยำสูง และมีขนาดเล็กพอสำหรับโปรเจกต์ที่มีพื้นที่จำกัด หรือต้องการให้น้ำหนักเบา



Servo Motor SG90

#Checkpoint3 ควบคุมการทำงานของ Motor

จะเขียนโปรแกรม Arduino ให้ควบคุม Servo Motor หมุนที่:

0 องศา 1วินาที (ครั้งแรกจะหมุนอยู่แล้ว) → 90 องศา 1วินาที → 180 องศา 1วินาที

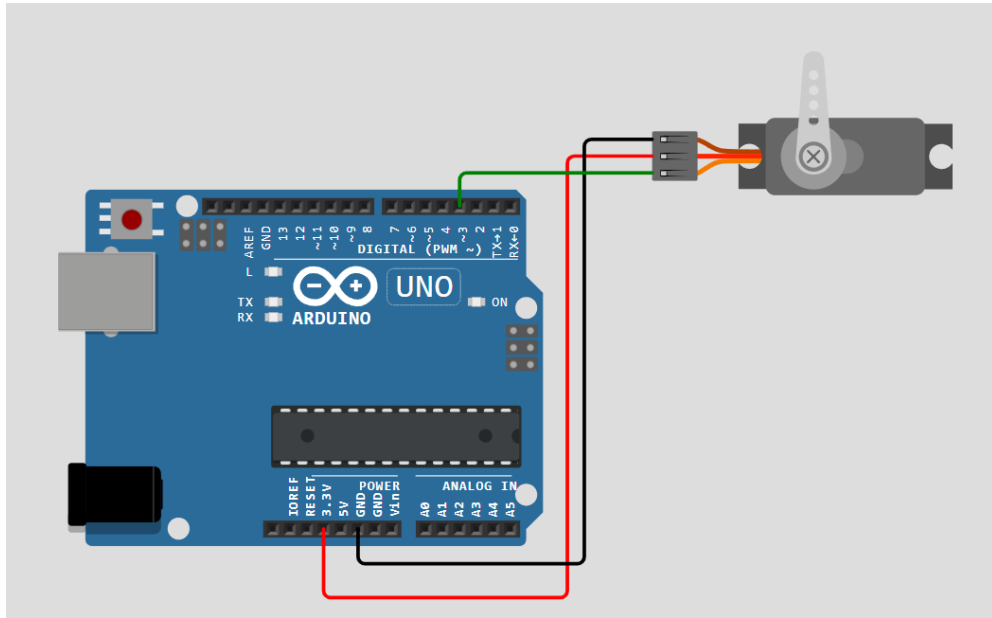
คำใบ้ Checkpoint3

ตัวอย่างCode

```
#include <Servo.h>
Servo myservo; //ประกาศตัวแปรแทน Servo

void setup()
{
  myservo.attach(3); // กำหนดขา 3 ควบคุม Servo
}
void loop()
{
  myservo.write(0); // สั่งให้ Servo หมุนไปองศาที่ 0
  delay(1000); // หน่วงเวลา 1000ms
  myservo.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
  myservo.write(180); // สั่งให้ Servo หมุนไปองศาที่ 180
  delay(1000); // หน่วงเวลา 1000ms
}
```

ตัวอย่างการต่อวงจร



น้องๆสามารถเปลี่ยนPinเป็นตัวอื่นได้เลยหากสายไฟยาวไม่เพียงพอ

(อย่าลืมเปลี่ยนCodeตรงกำหนดค่าของขาด้วย)

เฉพาะServo Moterต้องใช้ขาPWM(~)

#Checkpoint4 Tesla Pilot

จงศึกษารหัสโปรแกรมต่อไปนี้ ซึ่งใช้ Ultrasonic Sensor วัดระยะทางของวัตถุ และเตรียมควบคุมไฟ LED 2 ดวง และ Servo Motor สำหรับการเปิดประตูอัตโนมัติ

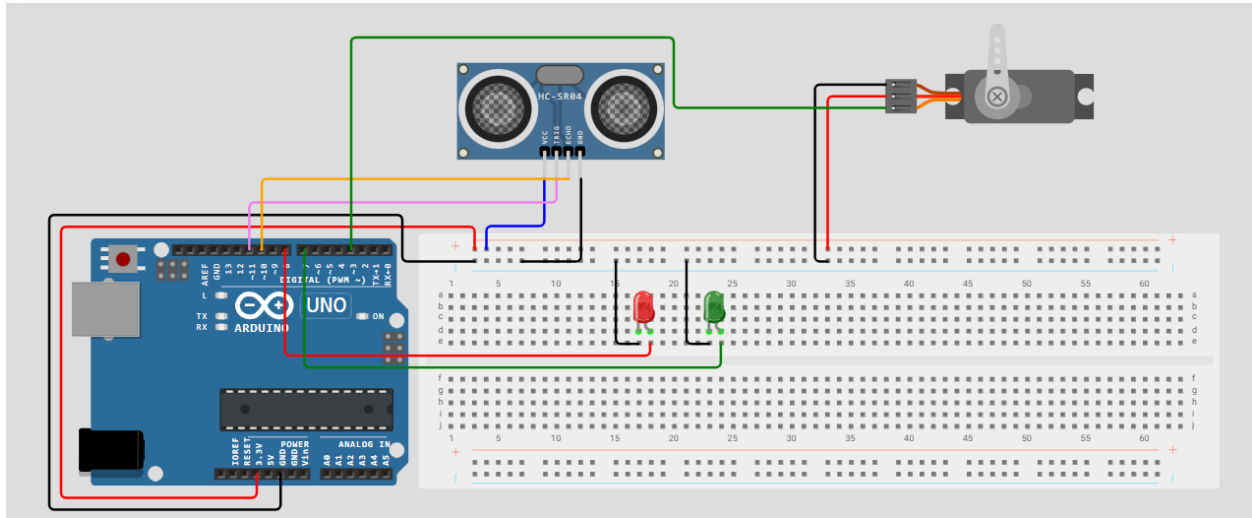
โค้ดมีการคำนวณระยะทางเก็บไว้ในตัวแปร distance แล้ว
ให้นักเรียนเขียนคำสั่งต่อ (ใน loop() หลังจาก distance = duration / 58;) เพื่อควบคุมอุปกรณ์ตามเงื่อนไขดังนี้:

เงื่อนไขการทำงาน

1. ถ้าระยะทาง อยู่ระหว่าง 0 ถึง 25 เซนติเมตร
 - เปิด LED ดวงที่ 1 (ledPin1) **หลอดดวงสีแดง**
 - หมุน Servo ไปที่ 180 องศา
2. ถ้าระยะทาง ตั้งแต่ 50 เซนติเมตรขึ้นไป
 - เปิด LED ดวงที่ 2 (ledPin2) **หลอดดวงสีเขียว**
 - หมุน Servo กลับไปที่ 0 องศา

คำใบ้ Checkpoint4

ตัวอย่างการต่อวงจร



น้องๆสามารถเปลี่ยนPinเป็นตัวอื่นได้เลยหากสายไฟยาวไม่เพียงพอ

(อย่าลืมเปลี่ยนCodeตรงกำหนดค่าของขาด้วย)

เฉพาะServo Moterต้องใช้ขาPWM(~)

ตัวอย่างCode

```
#include <Servo.h>          // เรียกใช้ไลบรารีควบคุม Servo

Servo myservo;              // สร้างตัวแปรสำหรับควบคุม Servo
int trigPin = 11;           // ขาที่ใช้ส่งเสียงออก
int echoPin = 10;           // ขาที่รับเสียงสะท้อนกลับ
int ledPin1 = 8;            // ขาที่ต่อกับหลอดไฟ LED
int ledPin2 = 7;            // ขาที่ต่อกับหลอดไฟ LED

void setup() {
  pinMode(trigPin, OUTPUT); // ตั้งค่าให้ขา trig เป็นขาออก
  pinMode(echoPin, INPUT);  // ตั้งค่าให้ขา echo เป็นขารับข้อมูล
  pinMode(ledPin1, OUTPUT); // ตั้งค่าให้ขา LED เป็นขาออก
  pinMode(ledPin2, OUTPUT); // ตั้งค่าให้ขา LED เป็นขาออก
  myservo.attach(3);        // กำหนดให้ Servo ใช้ขา 3
  myservo.write(0);         // เริ่มต้นหมุน Servo ที่ 0 องศา
  Serial.begin(9600);       // เริ่มต้นการส่งข้อมูล Serial
}

void loop() {
  long duration;
  int distance;

  // ส่งสัญญาณ Ultrasonic
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // รับสัญญาณกลับ และคำนวณระยะ
  duration = pulseIn(echoPin, HIGH);
  distance = duration / 58;
```

ตัวอย่างCode(ต่อ)

```
// แสดงผลระยะทาง
Serial.print("ระยะทาง:");
Serial.print(distance);
Serial.println(" ซม.");

if (distance < 25) {
  // วัตถุใกล้มาก → เปิดทุกดวง
  myservo.write(180);
  digitalWrite(ledPin1, .....);
  digitalWrite(ledPin2, .....);
} else {
  // วัตถุอยู่ไกล → ปิดทุกดวง
  myservo.write(0);
  digitalWrite(ledPin1, .....);
  digitalWrite(ledPin2, .....);
}

delay(100); // รอ 1 วินาที ก่อนเริ่มวัดใหม่
}
```