

COCSC20- INTERNET OF THINGS

EXERCISE- 2

JANUARY 19, 2023

AMOGH GARG – 2020UCO1688

PROBLEM:

Design a circuit for interfacing the Ultrasonic sensor in TINKERCAD framework and plot the captured data using Processing tool.

COMPONENTS REQUIRED:

- 1 x Arduino Uno R3
- 1 x HC-SR04 Ultrasonic Range Sensor
- 1 x Servo Motor
- Processing tool

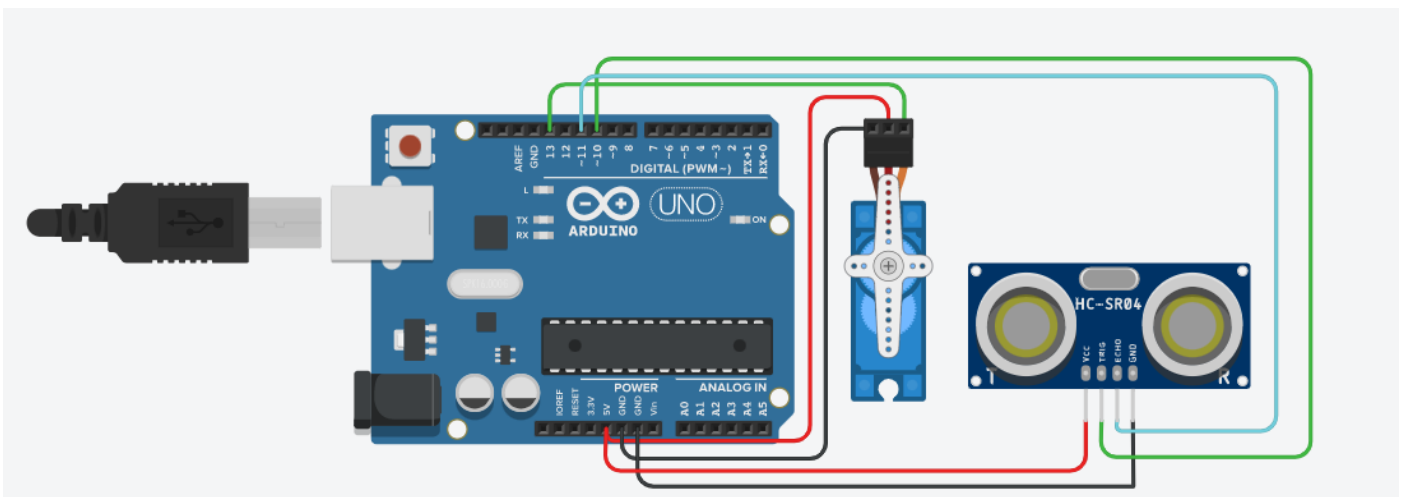
UNDERSTANDING REQUIRED:

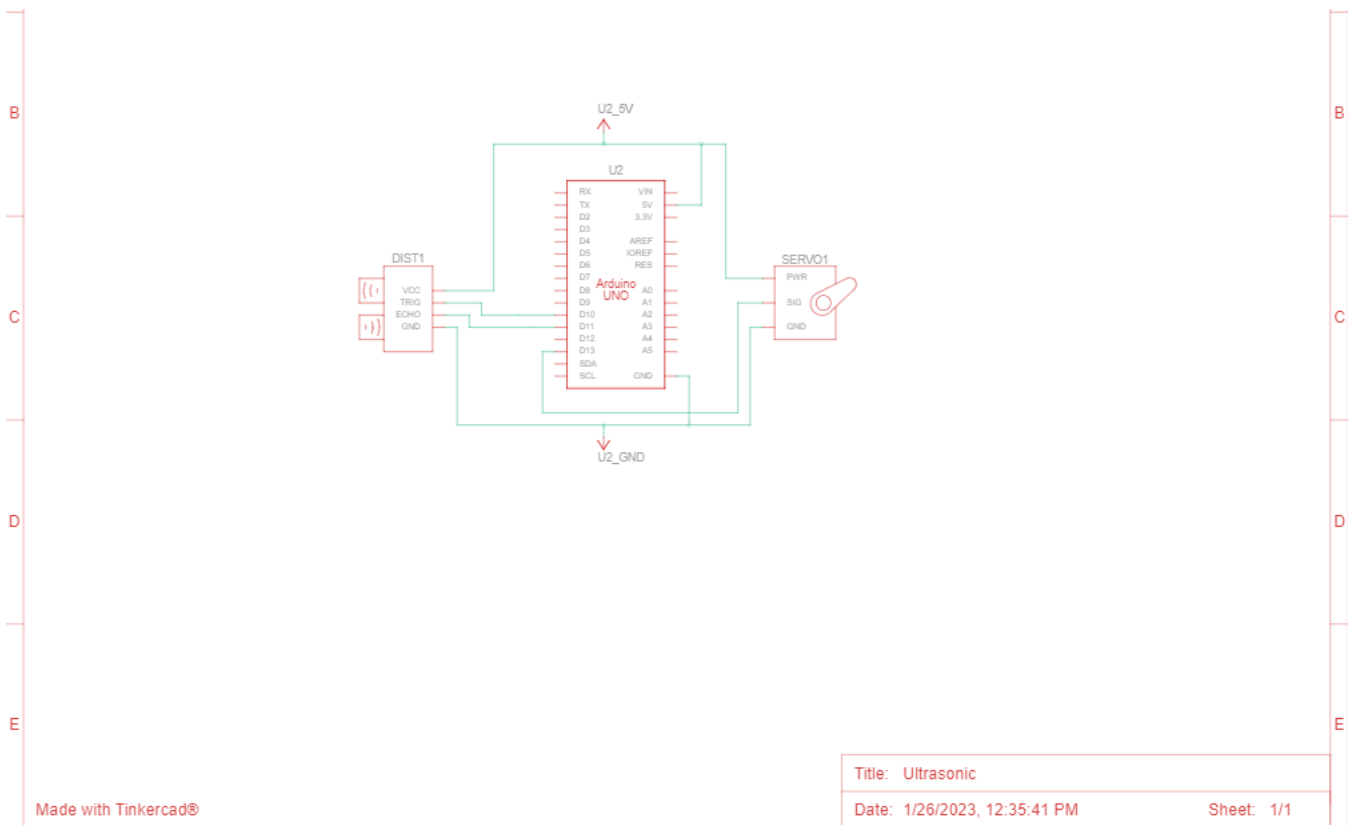
- Knowledge about sensor technology, various types of sensors
- Arduino Backgrounds, Type of Arduino
- Working and principal of Ultrasonic sensor HC-SR04
- Circuit design and sketch of interfacing
- Knowledge about processing tool

SOLUTION:

- Import the Arduino Uno and Servo motor combined component available on the TINKERCAD framework.
- Connect the VCC point of the HC-SR04 to the 5V point of the Arduino Uno and GND to the GND of HC-SR04.
- Trigger is connected to D10 and echo is connected to D11.
- Write the Arduino code and processing code in the respective IDEs and start simulation.

CIRCUIT DIAGRAM:





CODE:

Arduino code:

```
// Includes the Servo library

#include <Servo.h>

// Defines Trirg and Echo pins of the Ultrasonic Sensor

const int trigPin = 10;

const int echoPin = 11;

// Variables for the duration and the distance

long duration;

int distance;

Servo myServo; // Creates a servo object for controlling the servo motor

void setup() {

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  Serial.begin(9600);

  myServo.attach(13); // Defines on which pin is the servo motor attached

}

void loop() {

  // rotates the servo motor from 15 to 165 degrees

  for(int i=15;i<=165;i++){

    myServo.write(i);

    delay(30);

    distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree

    Serial.print(i); // Sends the current degree into the Serial Port
```

```

Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing

Serial.print(distance); // Sends the distance value into the Serial Port

Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
}

// Repeats the previous lines from 165 to 15 degrees
for(int i=165;i>15;i--){

myServo.write(i);

delay(30);

distance = calculateDistance();

Serial.print(i);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

}

// Function for calculating the distance measured by the Ultrasonic sensor

int calculateDistance(){

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds

distance= duration*0.034/2;

return distance;

}

```

Processing Code:

```

// Includes the Servo library

#include <Servo.h>.

// Defines Trig and Echo pins of the Ultrasonic Sensor

const int trigPin = 10;

const int echoPin = 11;

// Variables for the duration and the distance

long duration;

int distance;

Servo myServo; // Creates a servo object for controlling the servo motor

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

```

```

Serial.begin(9600);

myServo.attach(13); // Defines on which pin is the servo motor attached
}

void loop() {

  // rotates the servo motor from 15 to 165 degrees
  for(int i=15;i<=165;i++){
    myServo.write(i);
    delay(30);

    distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree
    Serial.print(i); // Sends the current degree into the Serial Port
    Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
    Serial.print(distance); // Sends the distance value into the Serial Port
    Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
  }

  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165;i>15;i--){
    myServo.write(i);
    delay(30);

    distance = calculateDistance();
    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
  }
}

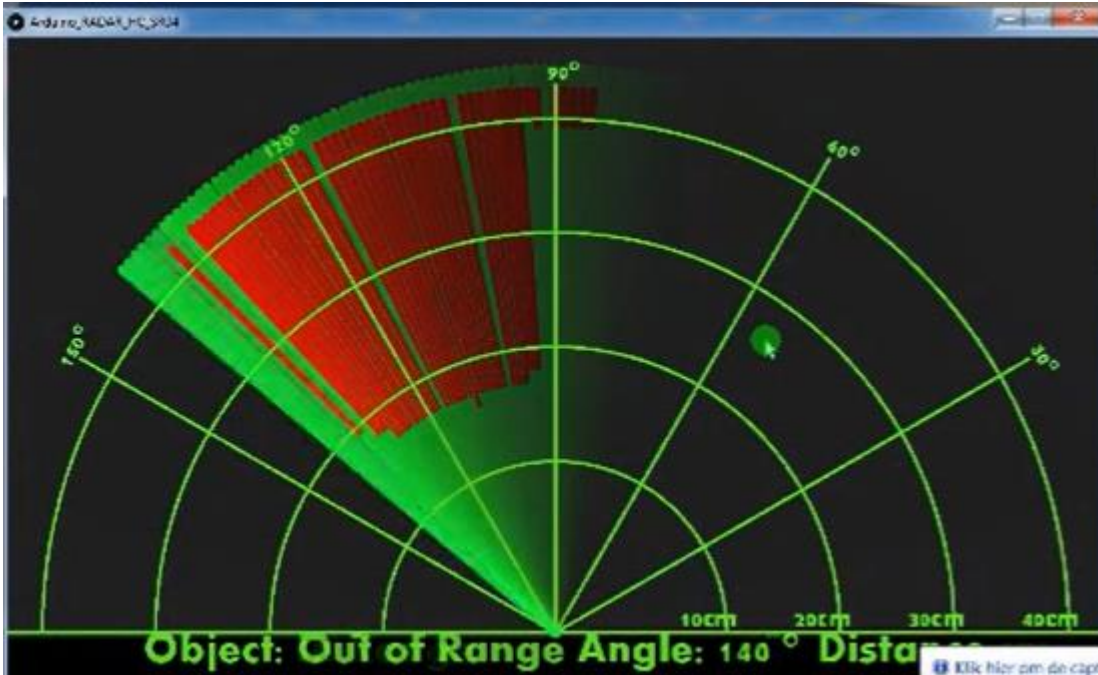
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds
  distance= duration*0.034/2;

  return distance;
}

```

RESULT:

The sensor detects the presence and distance of the objects correctly. Below is the output screen-shot.



SOURCES OF ERROR:

- Poor quality power source
- Poor mounting angle
- Factors like wind current, extreme temperatures, humidity etc.

What is the maximum and minimum distance can be captured with sensor? Explain why?

The range of the HC-SR04 ultrasonic sensor is between 2cm and 400cm because this is the maximum distance at which the sensor can accurately measure the distance to an object. The sensor works by emitting an ultrasonic wave and measuring the time it takes for the wave to bounce back to the sensor after it hits an object. The sensor then uses the speed of sound and the time it took for the wave to bounce back to calculate the distance to the object. At a distance of 2cm, the ultrasonic wave has enough time to bounce back to the sensor before the next wave is emitted. At a distance of 400cm, the wave takes the longest time to bounce back, and the sensor can still accurately measure the distance. Beyond 400cm, the wave takes too long to bounce back and the sensor will not be able to accurately measure the distance.