BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

INTERFACING TECHNIQUES

ENCS4380

Task #1

Arduino basics and ultrasonic sensors

Prepared by:

Mohammad Fareed **ID: 1212387**

Ahmad Hamdan **ID: 1210241**

Qossay Rida **ID: 1211553**

Mohammed Abed Alkareem **ID: 1210708**

Instructor: **Dr. Wasel Ghanem**

Teaching assistant: **Mr. Loai Shehadeh**

Date: **Thursday, May 2, 2024**

Abstract

This project involves building a distance measurement system with Arduino, ultrasonic sensors, and LEDs. The ultrasonic sensor will measure the distance between the system and an object, and three LEDs will visually represent this distance. The closer the object, the more LEDs will light up.

Table of Contents

Table of Figures

1. Theory

The HC-SR04 ultrasonic module operates based on the principles of SONAR and RADAR systems. It comprises an ultrasonic transmitter, receiver, and control circuit integrated onto a single board, with only four pins: Vcc, Gnd, Trig, and Echo.

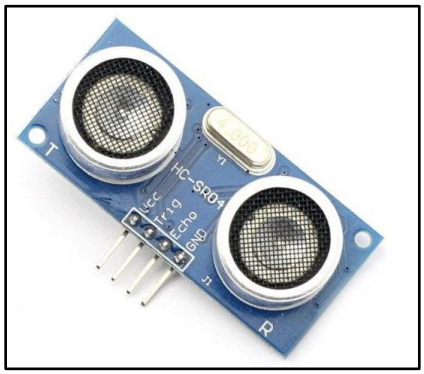

Figure 1-1 Ultrasonic HC-SR04 Module

When a pulse of 10 microseconds or longer is applied to the Trig pin, the module generates eight pulses of 40 kHz. Subsequently, the control circuit in the module sets the Echo pin high.

The Echo pin remains high until it receives the echo signal of the transmitted pulses back. The duration for which the Echo pin stays high, representing the width of the Echo pulse, indicates the time taken for the ultrasonic sound to travel towards the object and return.

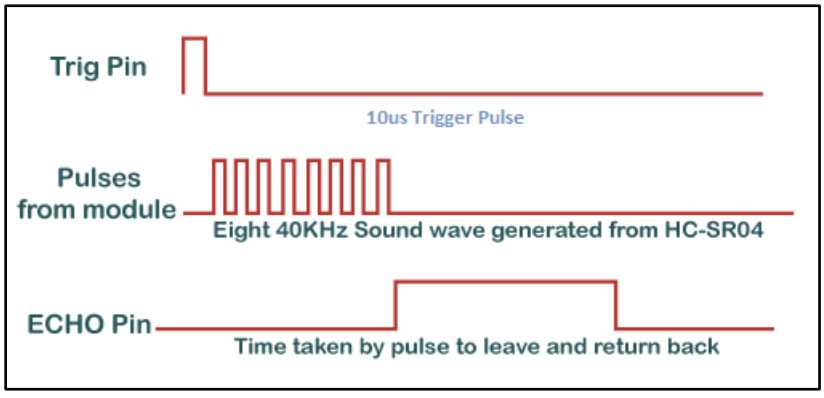

Figure 1-2 ultrasonic Signals

By utilizing this time along with the speed of sound in air, it's possible to calculate the distance to the object using a straightforward formula relating speed, time, and distance.

```
distance = time x 0.034/2
```

2. Procedure

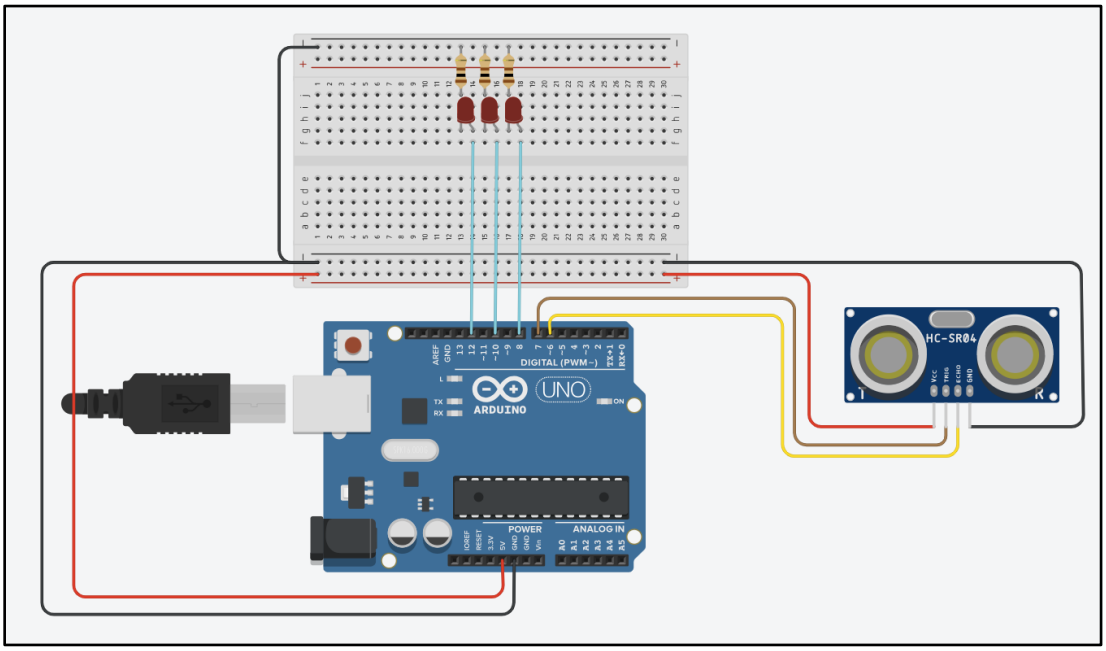

Figure 2-1 ultrasonic circuit

The circuit was connected as in fig(2-1)

The microcontroller was programmed on Arduino IDE

Code Clarification

The code can be found in the appendix

Initialization:

the pins for the ultrasonic sensor (trigPin and echoPin) and the three LEDs (led1Pin, led2Pin, and led3Pin) were defined.

a variable previousMillis was initialized to store the last time the LEDs were toggled.

the interval for LED blinking was set to 500 milliseconds.

Setup:

serial communication for debugging purposes was initialized.

the pins were set as either INPUT or OUTPUT as required.

Loop:

a short pulse to the ultrasonic sensor was sent to trigger a measurement.

the duration of the echo pulse was measured using pulseIn() function and the distance based on the time it took for the pulse to return was calculated.

the measured distance was printed to the Serial Monitor.

Based on the measured distance, control the illumination of the LEDs:

- If the distance is less than 10 cm, all LEDs are turned on.

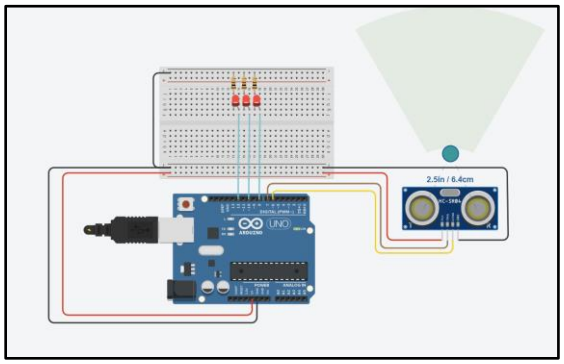

Figure 2-2 when distance is less than 10 cm

- If the distance is between 10 and 20 cm, only two LEDs are turned on.

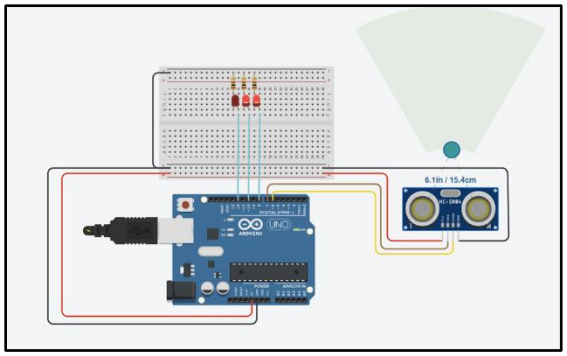

Figure 2-3 when distance is between 10 and 20 cm

- If the distance is between 20 and 30 cm, only one LED is turned on.

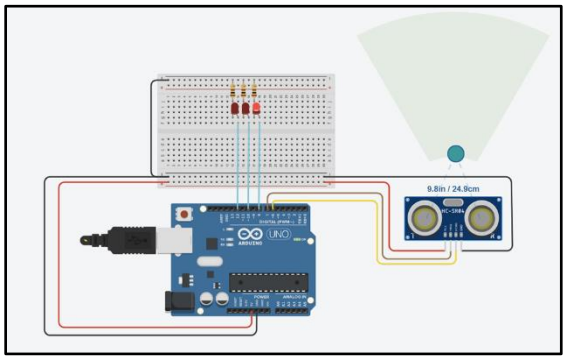

Figure 2-4 when distance is between 20 and 30 cm

- If the distance is greater than or equal to 30 units, the LEDs blink at a 500ms interval.

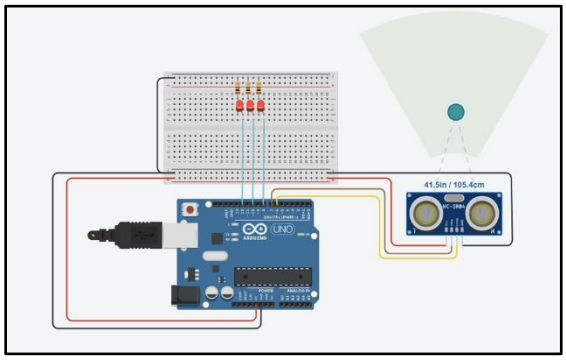

Figure 2-5 when distance is greater than or equal to 30 cm

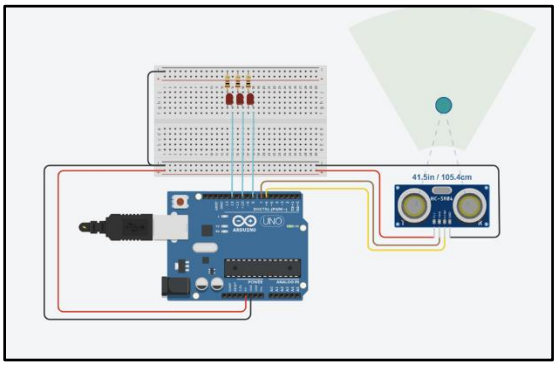

Figure 2-6 when distance is greater than or equal to 30 cm after 0.5sec

Blinking LEDs:

If the distance is greater than or equal to 30 units, the LEDs alternate between ON and OFF states every 500 milliseconds.

Conclusion

To sum up, this project uses Arduino, ultrasonic sensors, and LEDs to measure distance. The closer an object is to the sensor, the more LEDs light up. It's a simple way to gauge proximity visually.

References

[1] Avr Atmega Ultrasonic Module Hc Sr04 Interfacing With Atmega1632 ... (electronicwings.com)

[2] Arduino Ultrasonic distance sensor - JavaTpoint

Appendix

```
const int trigPin = 7;
const int echoPin = 6;
const int led1Pin = 8;
const int led2Pin = 10;
const int led3Pin = 12;

unsigned long previousMillis = 0;
const long interval = 500;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance: ");
  Serial.println(distance);

  if (distance < 10) {
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
    digitalWrite(led3Pin, HIGH);
  } else if (distance >= 10 && distance < 20) {
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
    digitalWrite(led3Pin, LOW);
  } else if (distance >= 20 && distance < 30) {
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
    digitalWrite(led3Pin, LOW);
  } else {
```

```
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;
      if (digitalRead(led1Pin) == HIGH) {
      digitalWrite(led1Pin, LOW);
      digitalWrite(led2Pin, LOW);
        digitalWrite(led3Pin, LOW);
      }
      else{
        digitalWrite(led1Pin, HIGH);
      digitalWrite(led2Pin, HIGH);
      digitalWrite(led3Pin, HIGH);
      }
    }

  }

}
```