Experiment No. - 5

Student Name: Vivek Kumar **UID: 21BCS8129**
Branch: BE-CSE(LEET) **Section/Group:20BCS-ST-801/B**
Semester: 6th **Date of Performance: 10/03/2023**
Subject Name: IOT Lab **Subject Code: 20CSP-358**

1. **Aim:**
   To measure the distance of an object using an ultrasonic sensor.
2. **Objective:**
   a. Learn about ultrasonic sensor in detail.
   b. Learn about IoT programming.
   c. Measure the distance of an object using an ultrasonic sensor.
3. **Requirements:**
   a. Arduino Uno R3 board
   b. Ultrasonic sensor (HC-SR04)
   c. 16×2 LCD I2C Display
   d. Jumper Wires
4. **Procedure:**

Arduino:

It is an open-source electronics platform. It consists ATmega328 8-bit Micro controller. It can be able to read inputs from different sensors & we can send instructions to the micro controller in the Arduino. It provides Arduino IDE to write code & connect the hardware devices like Arduino boards & sensors.

Ultrasonic Sensor:

An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion.

Working Principle of Ultrasonic Sensor:

Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

We know that the speed of sound in air is nearly 344 m/s,

So, the known parameters are time and speed (constant). Using these parameters, we can calculate the distance traveled by the sound wave.

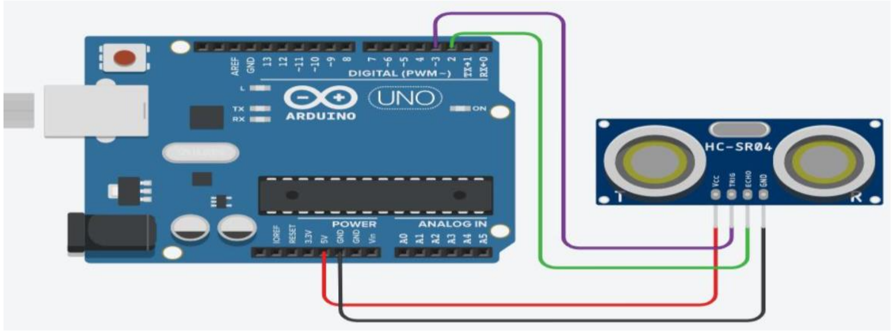
Formula: Distance = Speed * Time

In the code, the “duration” variable stores the time taken by the sound wave traveling from the emitter to the receiver. That is double the time to reach the object, whereas the sensor returns the total time including sender to object and object to receiver. Then, the time taken to reach the object is half of the time taken to reach the receiver.

Distance = Speed of Sound in Air * (Time Taken / 2)

5. Setup:

a. Connect the Echo pin of the sensor to the D2 pin of the Arduino.
b. Connect the Trig pin of the sensor to the D3 pin of the Arduino.
c. Navigate to Tools and select board and port.
d. Verify and compile the code, then upload the code to the Arduino Uno R3 board.
e. Monitor the output in the Serial monitor (Set the baud rate as 9600). To open Serial monitor **Tools>Serial Monitor** or (**Ctrl+Shift+M**).



Steps to Interface LCD display:

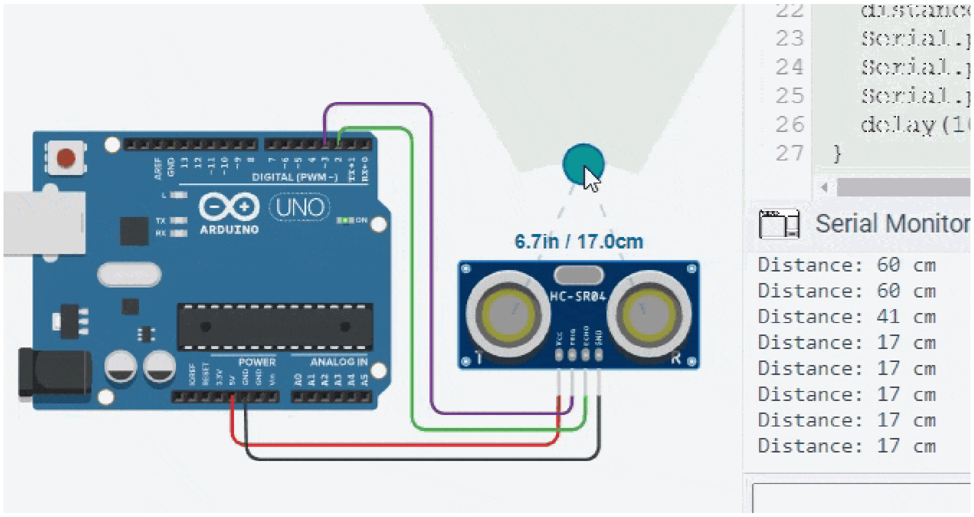
a. Install driver library for Liquid Crystal Display.
b. Import the header file “LiquidCrystal_I2C.h” in the code.
c. Connect the SDA pin of an LCD display to the SDA pin of the Arduino Board and the SCL pin of an LCD display to the SCL of the Arduino Board.
d. Connect VCC to 5V pin and GND to GND pin.
e. Include the below code to define the display device.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x20, 16, 2); // Format => (ADDRESS,Width,Height )

void setup()
{
   lcd.init(); // Initialize the lcd
   lcd.backlight(); // Turn on the Backlight
}
```

6. Steps/Program:

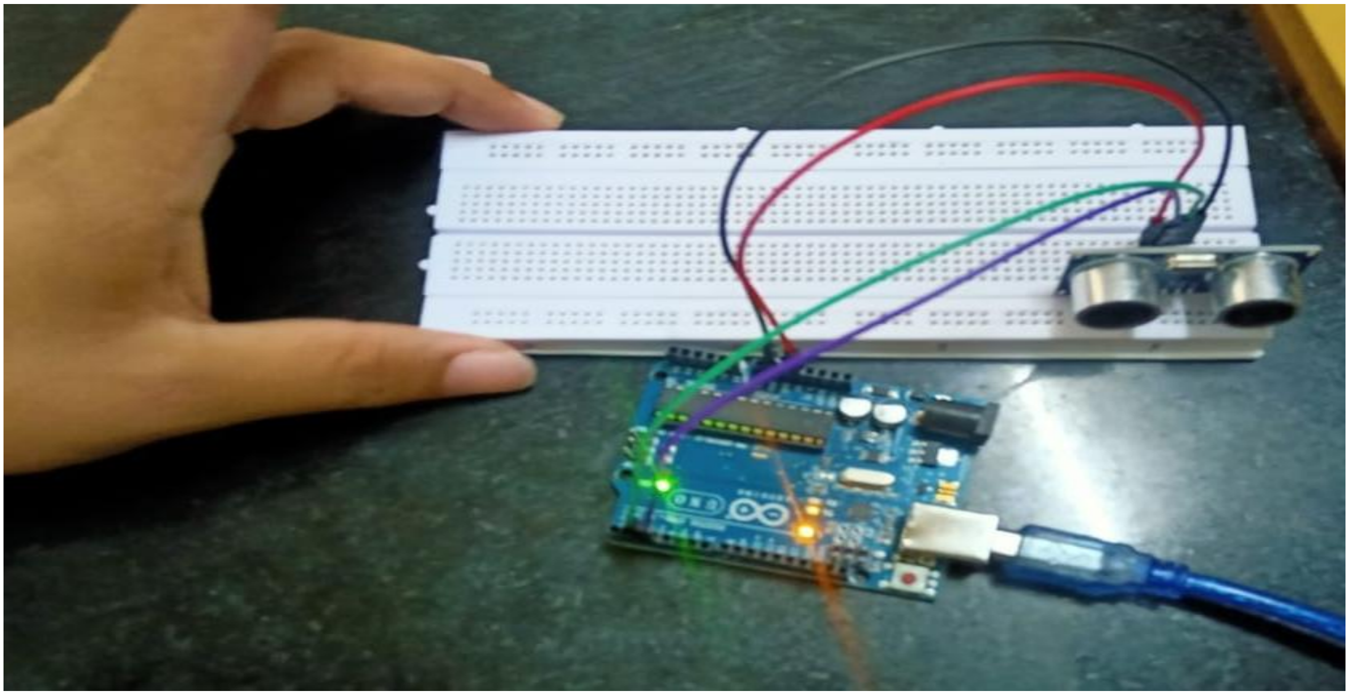
```
#define echoPin  2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3// attach pin D3 Arduino to pin Trig of HC-SR04                    \

long duration; // Variable to store time taken to the pulse to reach receiver
int distance; // Variable to store distance calculated using formula
void setup()
{
  pinMode(trigPin,OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  // Serial Communication is starting with 9600 of baudrate speed
  Serial.begin(9600);
  // The text to be printed in serial monitor
  Serial.println("Distance measurement using Arduino Uno.");
  delay(500);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2); // wait for 2 ms to avoid collision in serial monitor
  digitalWrite(trigPin, HIGH); // turn on the Trigger to generate pulse
  delayMicroseconds(10); // keep the trigger "ON" for 10 ms to generate pulse for 10 ms.
  digitalWrite(trigPin,LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0344 / 2;
  Serial.print("Distance: ");
  Serial.print(distance); // Print the output in serial monitor
  Serial.println(" cm");
  delay(100);
}
```

7. Output:



```
  distanc
  Serial.
  Serial.
  Serial.
  delay(1
}
```

```
Distance: 60 cm
Distance: 60 cm
Distance: 41 cm
Distance: 17 cm
Distance: 17 cm
Distance: 17 cm
Distance: 17 cm
Distance: 17 cm
```



Learning outcomes (What I have learnt):

- Learnt about ultrasonic sensor in detail.
- Learnt about IoT programming.
- Measured the distance of an object using an ultrasonic sensor.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	