EXPERIMENT NO.5

STUDENT NAME: Harshit Raj **UID: 20BCS9266**
BRANCH: CSE **SECTION: 608/A**
SEMESTER: 6th **SUBJECT: IOT**

AIM: To measure the distance of an object using an ultrasonic sensor.

COMPONENTS REQUIRED:

You will need the following components − 1. Arduino Uno R3 board
2. Ultrasonic sensor (HC-SR04)
3. 16×2 LCD I2C Display
4. Jumper Wires

THEORY:

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.

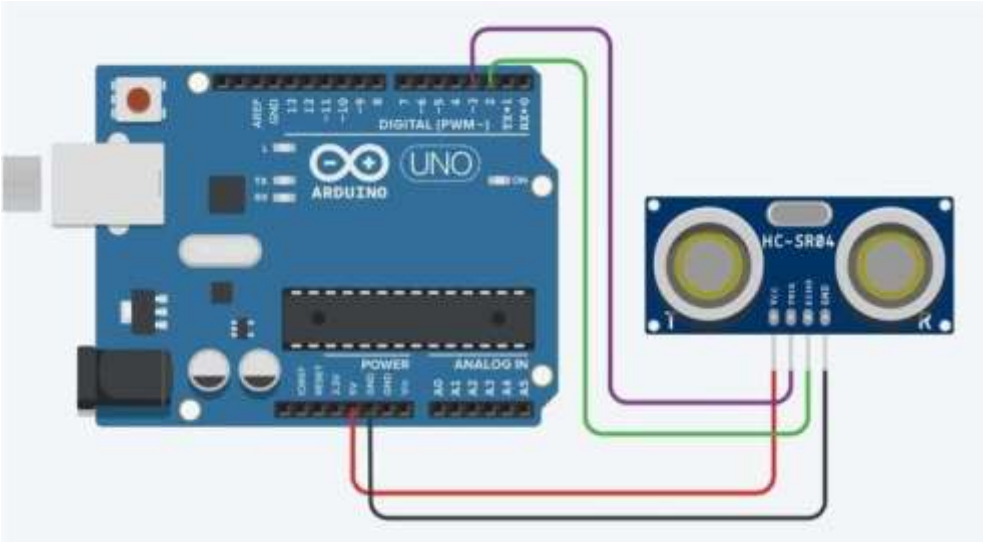
Working Principle of Ultrasonic Sensor:

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.
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. so we can write the expression as,
Distance = Speed of Sound in Air * (Time Taken / 2) Note: Speed of sound in air = 344 m/s.

PROCEDURE:

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



Follow the circuit diagram as shown in the image given above.
To interface with the LCD display, we need to install the supporting library to the Arduino IDE.

Steps to Interface LCD display:

1. Install driver library for Liquid Crystal Display.
   - Navigate Tools>Library Manager (or) Enter (Ctrl+Shift+I) to open library manager.
   - Search for “LiquidCrystal I2C” and install the “LiquidCrystal I2C” library. 2. Import the header file “LiquidCrystal_I2C.h” in the code.
3. 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.
4. Connect VCC to 5V pin and GND to GND pin.

SKETCH:

4theep | Arduino 1.8.19

File Edit Sketch Tools Help

4theep

```
#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
```

```
Sketch uses 3252 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 246 bytes (11%) of dynamic memory, leaving 1802 bytes for local variables. Maximum is 2048 bytes.
Invalid library found in C:\Users\student\Documents\Arduino\libraries\led: no headers files (.h) found in C:\Users\student\Documents\Arduino\libraries\led
```

4theep | Arduino 1.8.19

File Edit Sketch Tools Help

4theep

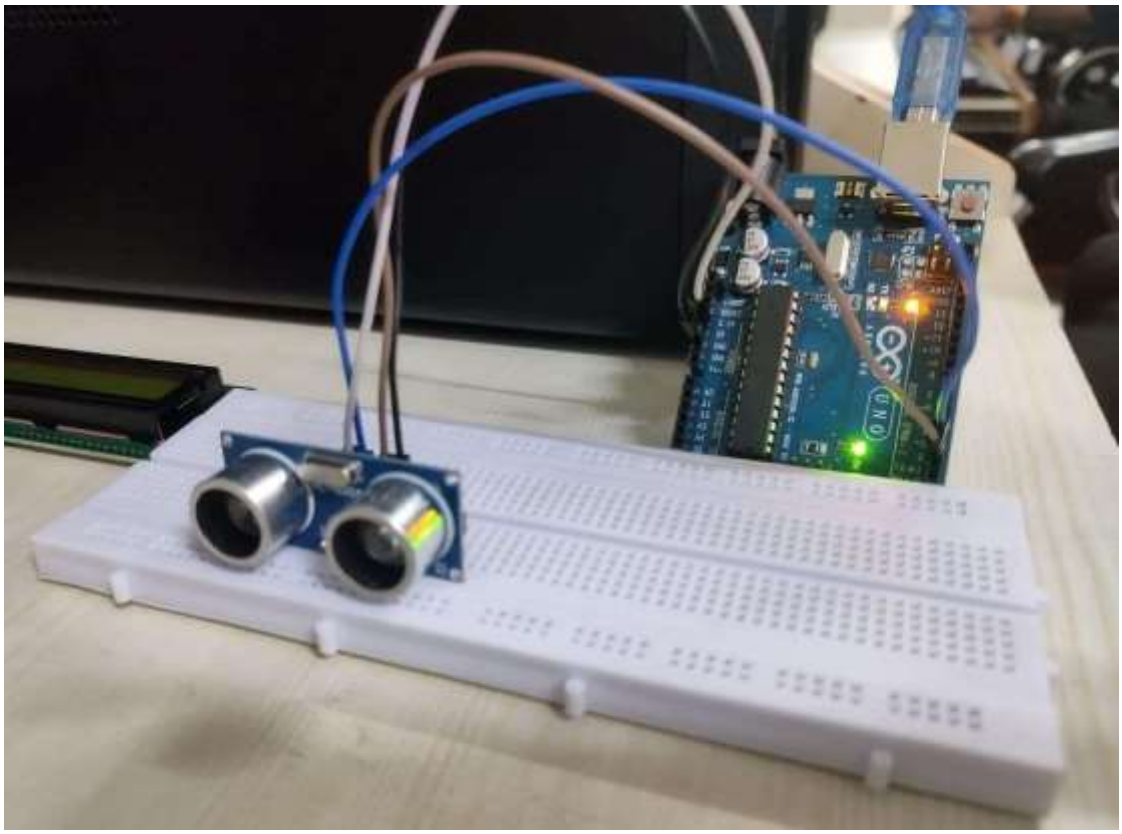
```
void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // wait for 2 ms to avoid
                          // collision in serial monitor

    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); // Turn off the pulse trigger to stop
                       // pulse generation

    // If pulse reached the receiver echoPin
    // become high Then pulseIn() returns the
    // time taken by the pulse to reach the
    // receiver

    duration = pulseIn(echoPin, HIGH);
    distance
        = duration * 0.0344 / 2; // Expression to calculate
                                 // distance using time

    Serial.print("Distance: ");
    Serial.print(
        distance); // Print the output in serial monitor
    Serial.println(" cm");
    delay(100);
}
```

```
Sketch uses 3252 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 246 bytes (11%) of dynamic memory, leaving 1802 bytes for local variables. Maximum is 2048 bytes.
Invalid library found in C:\Users\student\Documents\Arduino\libraries\led: no headers files (.h) found in C:\Users\student\Documents\Arduino\libraries\led
```

RESULT


COM8
Send
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Distance: 12 cm
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


COM8
Send
Distance: 3 cm
Distance: 3 cm
Distance: 4 cm
Distance: 5 cm
Distance: 4 cm
Distance: 4 cm
Distance: 4 cm
Distance: 4 cm
Distance: 4 cm
Distance: 4 cm
Distance: 4 cm
Distance: 5 cm
Distance: 4 cm
Distance: 4 cm
Autoscroll
Show timestamp
Newline
9600 baud
Clear output



LEARNING OUTCOMES:

- Learnt about the proximity sensor and its applications.
- Learnt how to connect proximity sensor on Arduino.
- Learnt how to connect LCD display. ⌜ Learnt how to code in Arduino.