Experiment - 5

Student Name: Sandeep Kumar **UID: 20BCS4885**
Branch: BE - CSE **Section/Group: 603/A**
Semester: 6th semester **Subject: Internet of Things**

Aim:

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

Objective:

1. Learn about ultrasonic Sensors.
2. Learn about IoT programming.

Components Required:

You will need the following components −

- Arduino Uno R3 board
- Ultrasonic sensor (HC-SR04)
- 16×2 LCD I2C Display
- Jumper Wires

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

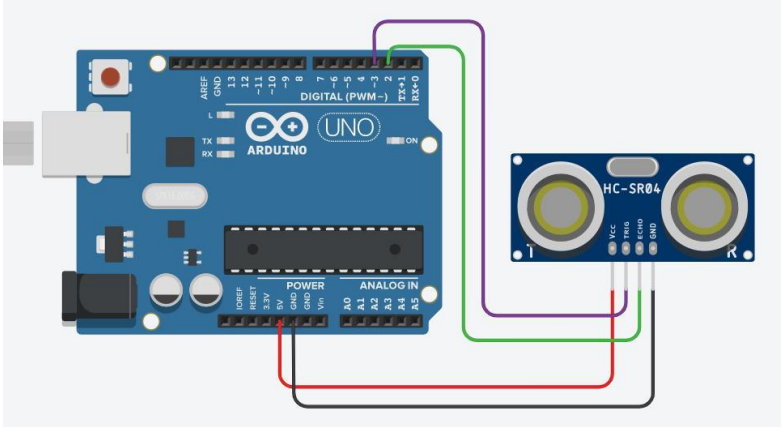
total time including sender to object and object to receiver. Then, the time taken to reach the object is half of the time taken to reach the receiver.

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.

Circuit Diagram:



Setup:

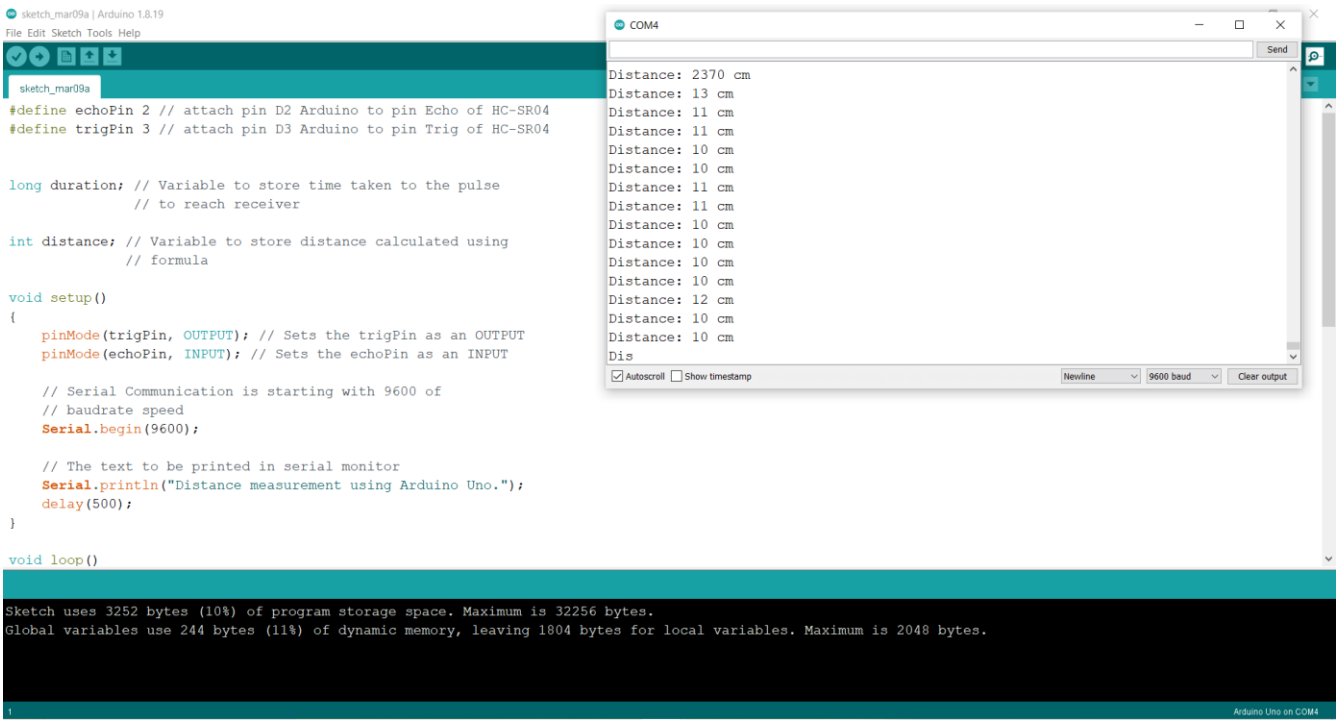
1. Connect the Echo pin of the sensor to the D2 pin of the Arduino.
2. Connect the Trig pin of the sensor to the D3 pin of the Arduino.
3. Navigate to Tools and select board and port.
4. Verify and compile the code, then upload the code to the Arduino Uno R3 board.
5. Monitor the output in the Serial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

Arduino Code (Output in Serial monitor):

```
#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); // 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);
}
```

Output (in Serial Monitor):



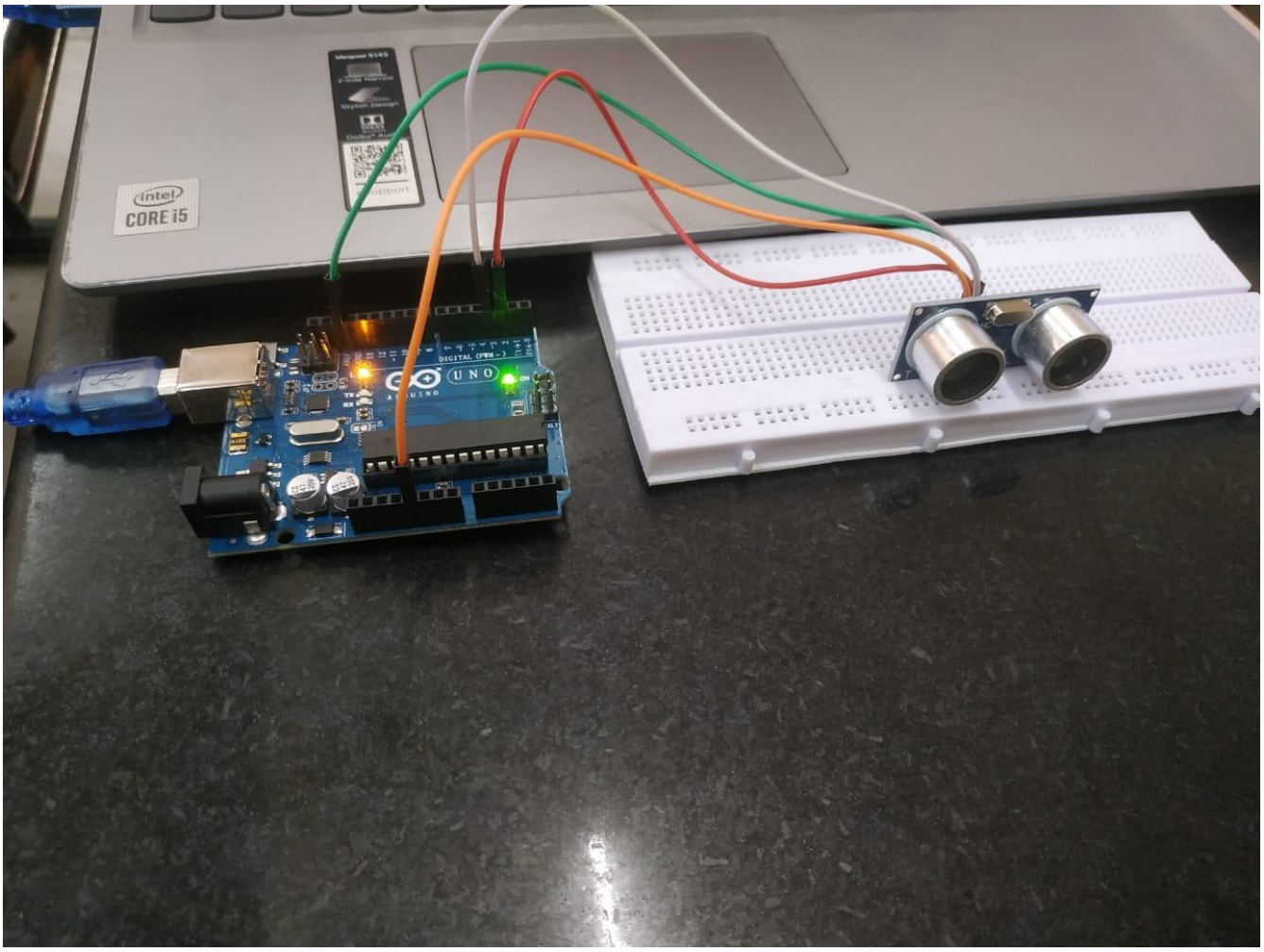
```
File Edit Sketch Tools Help
sketch_mar09a

#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()
```

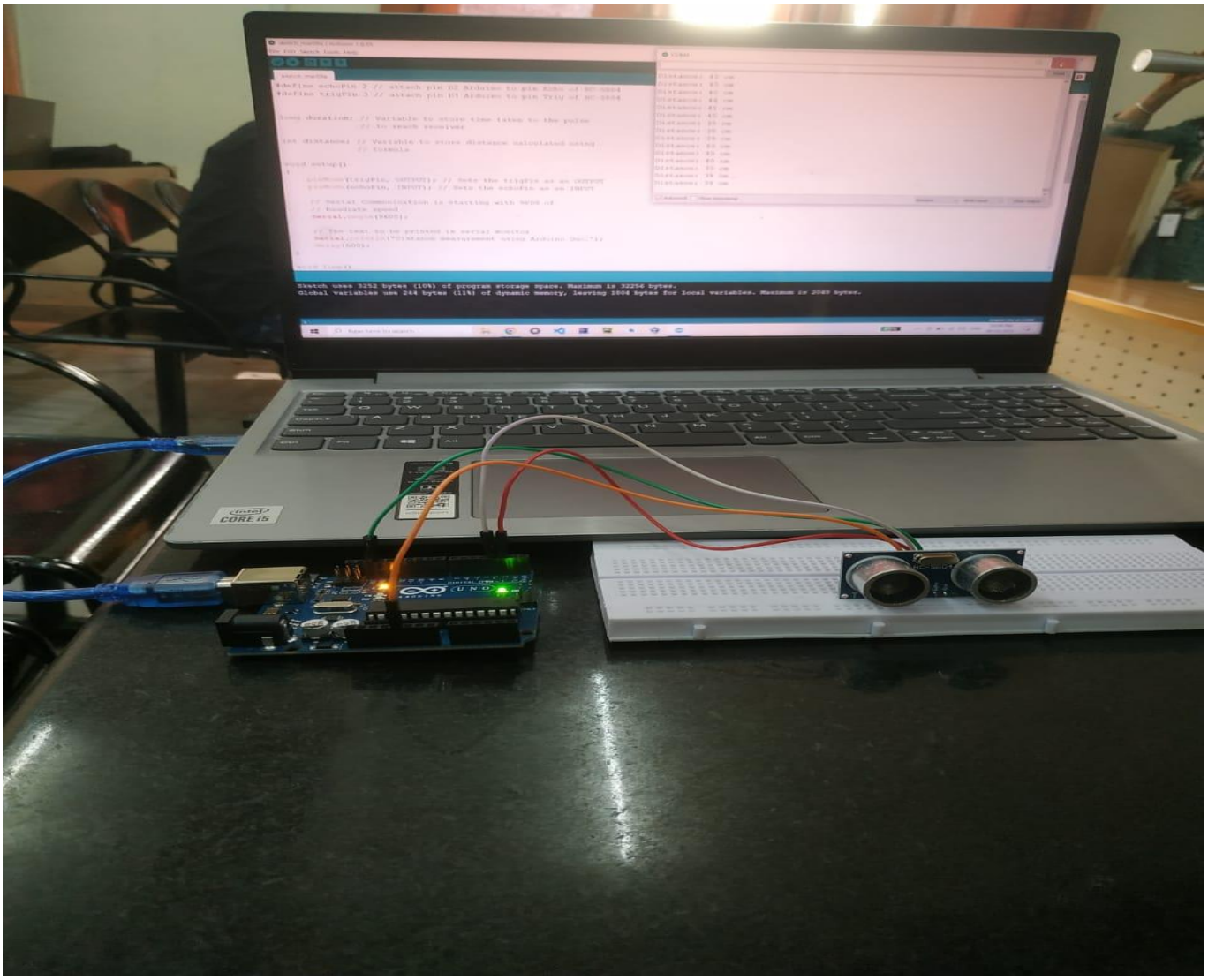
```
COM4
Send
Distance: 14 cm
Distance: 15 cm
Distance: 14 cm
Distance: 14 cm
Distance: 14 cm
Distance: 14 cm
Distance: 14 cm
Distance: 87 cm
Distance: 87 cm
Distance: 88 cm
Distance: 88 cm
Distance: 87 cm
Distance: 88 cm
Distance: 87 cm
Distance: 88 cm
Autoscroll  Show timestamp   Newline   9600 baud   Clear output
```

```
Sketch uses 3252 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 244 bytes (11%) of dynamic memory, leaving 1804 bytes for local variables. Maximum is 2048 bytes.
```

```
1                                                    Arduino Uno on COM4
```

Output:





Learning Outcomes:

- Learn about IoT based simulations.
- Learn about ultrasonic Sensors.
- Understanding the basic application and usage of the IOT devices.