AMERICAN INTERNATIONAL UNIVERSITY- BANGLADESH
Faculty of Engineering
Laboratory Report
Cover Sheet

Students must complete all details except the faculty use part.

Please submit all reports to your subject supervisor or the office of the concerned faculty.

Experiment Title: Interfacing the Arduino with an external sensor using serial communication protocol for implementing an obstacle detection system.

Experiment Number: **07**	Due Date: **02/05/2024**	Semester: **Spring 23-24**
Subject Code: **COE310**	Subject Name: **Microprocessor and Embedded Systems**	Section :**E**
Course Instructor: **Protik Parvez Sheikh**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report is my/our original work and no part of it has been copied from any other student's work or fromany other source except where due acknowledgement is made.
3. No part of this report has been written for me/us by any other person except where such collaboration hasbeen authorized by the lecturer/teacher concerned and is clearly acknowledged in the report.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the School for review andcomparison, including review by external examiners.

I/we understand that

7. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, includingelectronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. Enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy your work

Group Number (if applicable): 04 ☐ Individual Submission ☑ Group Submission

No.	Student Name	Student Id	Program	Signature
Submitted by:				
1	SAMIA SHARMIN DOLA	22-47126-1	CSE	
Group Members:				
2	SHAYAN ABRAR	22-47156-1	CSE	
3	MD SAMIN YEASAR	22-47139-1	CSE	
4	RIFAH SANZIDA	22-47154-1	CSE	
5	MD. JOBAER HOSSAIN	22-47116-1	CSE	

For faculty use only:	**Total Marks:**	**Marks Obtained:**
Faculty comments		

Theory and Methodology:

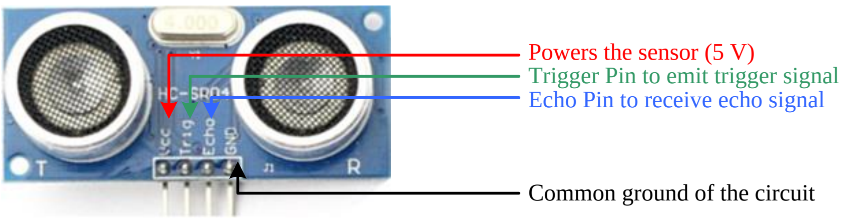
Arduino is an open-source platform used for creating interactive electronics projects. Arduino consists of both a programmable microcontroller and a piece of software, or IDE (Integrated Development Environment) that runs on your computer used to write and upload computer code to the microcontroller board. Arduino Uno also doesn't need a hardware circuit (programmer/ burner) to load a code into the board. We can easily load a code into the board just using a USB cable and the Arduino IDE (which uses an easier version of C++ to write codes).

In this experiment, we will use a sonar sensor (HCS04) to detect the distance of an obstacle. Based on the distance between the sensor and the object being detected, one or more LEDs will glow as soon as it detects the obstacle.

The HCSR04 ultrasonic sensor uses a sonar signal to determine the distance to an object. This sensor reads from 2 cm to 400 cm (0.8 inches to 157 inches) with an accuracy of 0.3 cm (0.1 inches). The HCSR04 module consists of a transmitter, receiver, and control circuit. It has four pins, such as VCC, GND, Trigger, and Echo. A list of some features and specifications of this sensor is given below, but for more information, you should consult the sensor's datasheet:

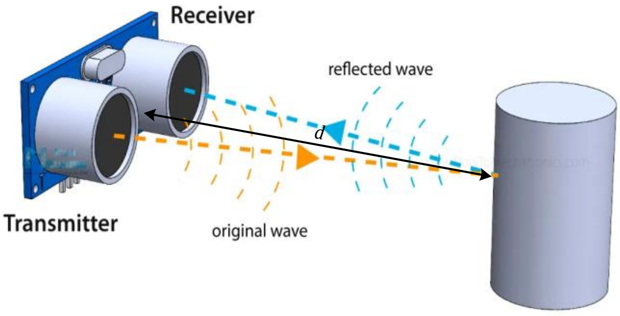
- Power Supply: +5 V DC
- Quiescent Current: < 2 mA
- Working Current: 15 mA
- Effective Angle: < 15°
- Ranging Distance: 2 cm – 400 cm/1″ – 13 ft
- Resolution: 0.3 cm
- Measuring Angle: 30°
- Trigger Input Pulse width: 10 μs TTL pulse
- Echo Output Signal: TTL pulse proportional to the distance range

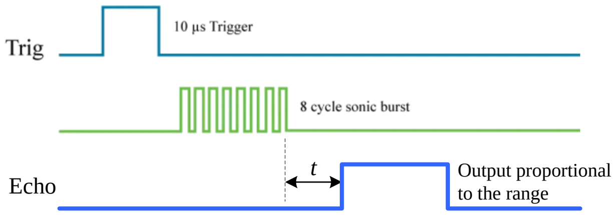
Here is the pin configuration of the sensor:



You can easily interface it with Arduino boards. Using the output Trigger pin, the module automatically sends eight 40 kHz pulse signals and detects whether there is a pulse signal back at the Echo pin. The Trigger pin of the sensor is connected to digital pin 11 and the Echo pin to digital pin 12 of the Arduino Uno R3 board with connecting wires. An LED is connected to pin 2 to show that an obstacle is detected. Here, pins 11 and 2 will act as output pins because the trigger will be generated from Arduino, and the LED state (HIGH/LOW) will also be changed by the Arduino board. If more than one LED is to be connected then we can use more digital pins, for example, 3 and 4. All these pins must be declared as output pins in the setup function.

The ultrasound transmitter (Trigger pin) emits a high-frequency ultrasonic sound wave (40 kHz) that travels through the air. If it finds an object, it bounces back to the module. The ultrasound receiver (Echo pin) receives the reflected sound wave (echo) as shown in the following schematic diagram:



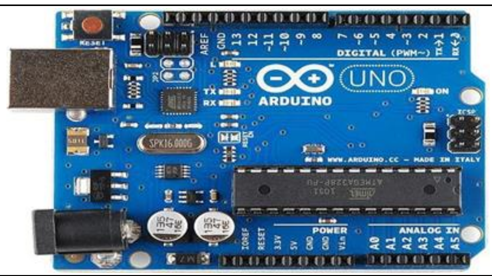


As the trigger signal generated from the Arduino board travels out from the Trigger pin and comes back to the Echo pin, the travel distance of the signal from the microcontroller to the object and back again to the microcontroller is double the distance (d) between the microcontroller and the board. We know that the sound velocity (v) in the air is 340 m/s.

$$2d = vt \text{ or simply, } d = \frac{t}{2}v$$

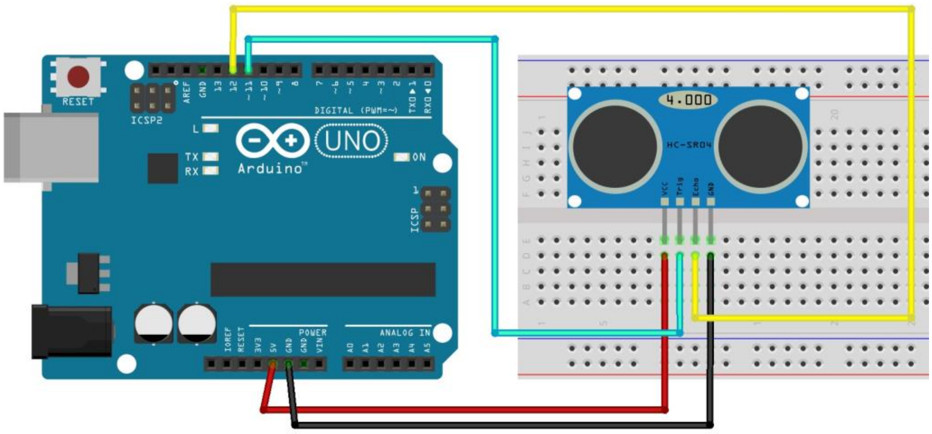
,where the t = travel time.

Apparatus:

1) Arduino IDE (any version)	Software
2) Arduino Uno (R3) board	
3) Sonar Sensor (HCSR04)	
4) LED	

Experimental Procedure:

Connect the circuit diagram as follows, upload the program into the board, and then run the program. Connect three LEDs of red, green, and yellow to pins 2, 3, and 4, respectively through three 100 Ω resistors to limit the current flow through LEDs (though it is not shown in the circuit diagram). Observe the object detection distances in the serial monitor and based on that LEDs turning ON.



Connection diagram

Program:

Write the following codes into the IDE and upload them into the Arduino board after compiling them correctly. **Please set up the Tools with the appropriate COM port before uploading the code.**

```
// define the pin numbers

const int trigPin = 11;

const int echoPin = 12;

// define variables

long duration;

float distance, distanceinches, distanceThreshold;

void setup() {

 Serial.begin(9600); // Starts the serial communication

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

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

pinMode(2, OUTPUT); // Sets pins 2, 3, and 4 as the Output pin

pinMode(3, OUTPUT);
```

```
pinMode(4, OUTPUT);

}

void loop() {

 // Clears the trigPin

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Sets the trigPin on HIGH state for 10 microseconds

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

 // Reads the echoPin, returns the sound wave travel time in microseconds

 duration = pulseIn(echoPin, HIGH);

 // Calculating the distance

 distance = (duration/2)*1e-6*340*100;

 distanceinches = (distance/2.54);

 // Prints the distance on the Serial Monitor

 Serial.print("Distance = ");

 Serial.print(distance);

 Serial.print(" cm; ");

 Serial.print("Distance = ");

 Serial.print(distanceinches);

 Serial.println(" inches");

// set threshold distance to activate LEDs

distanceThreshold = 80;

if (distance > distanceThreshold) {

 digitalWrite(2, LOW);

 digitalWrite(3, LOW);

 digitalWrite(4, LOW);
```

```
}
if (distance < distanceThreshold && distance > distanceThreshold-30) {
 digitalWrite(2, HIGH);
 digitalWrite(3, LOW);
 digitalWrite(4, LOW);
}
if (distance < distanceThreshold-30 && distance > distanceThreshold-50) {
 digitalWrite(2, HIGH);
 digitalWrite(3, HIGH);
 digitalWrite(4, LOW);
}
 if (distance < distanceThreshold-50 && distance > distanceThreshold-70 ) {
 digitalWrite(2, HIGH);
 digitalWrite(3, HIGH);
 digitalWrite(4, HIGH);
}
delay(200); // Wait for 200 millisecond(s)
}
```

Experimental Outcome:

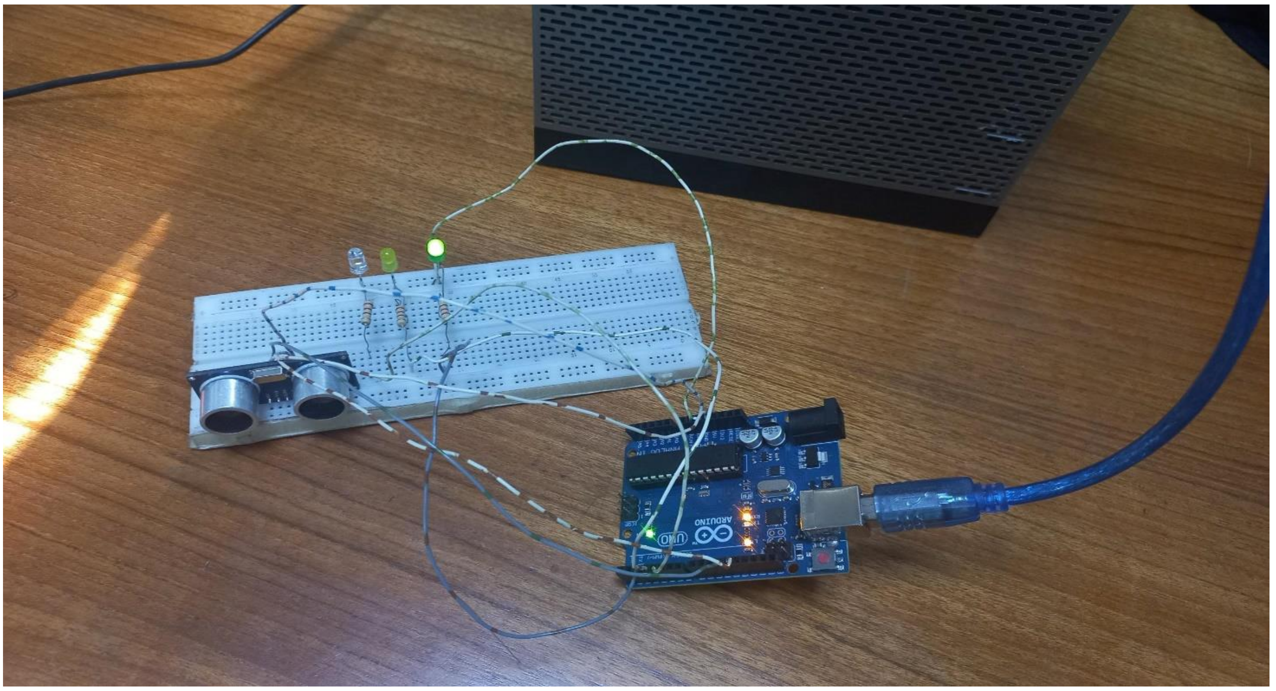


Fig:1. One Led turned ON when a distance at a (51 to 80cm) distance

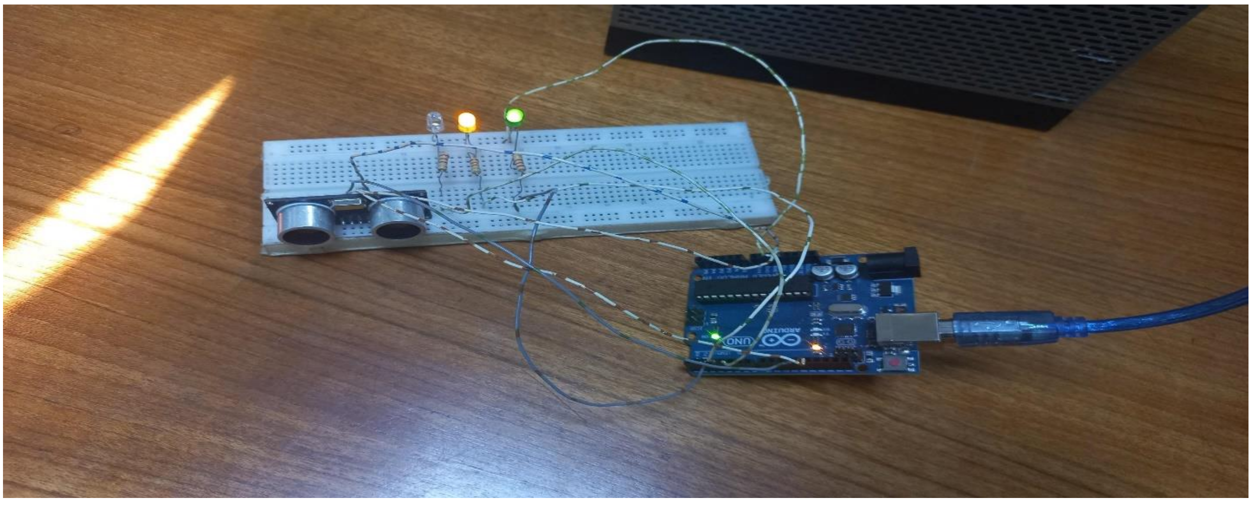


Fig2. Two Leds turned ON when a distance at a (31 to 50cm) distance

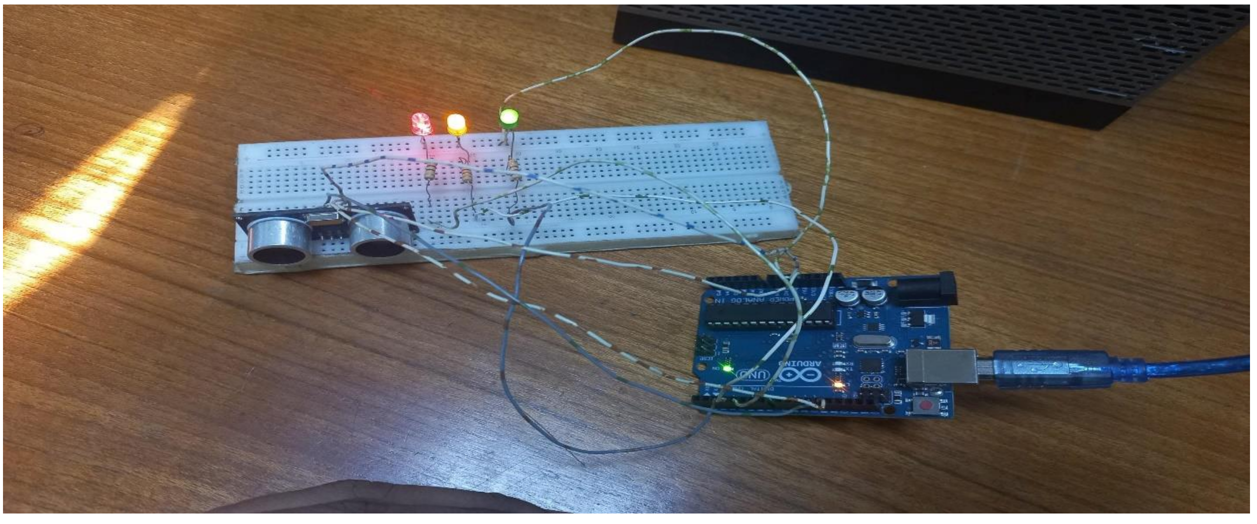


Fig.3. ALL Leds turned ON when a distance at a (0 to 30cm) distance

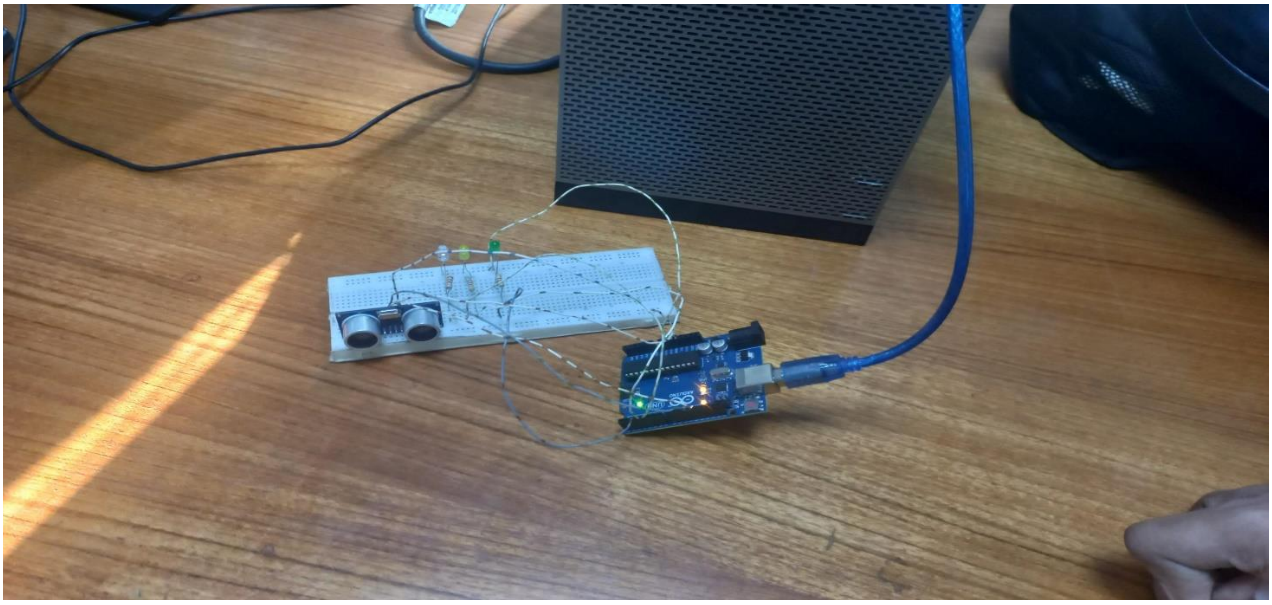


Fig.4. ALL Leds turned OFF when a distance at a (81cm and greater) distance

Simmulation Outcome:

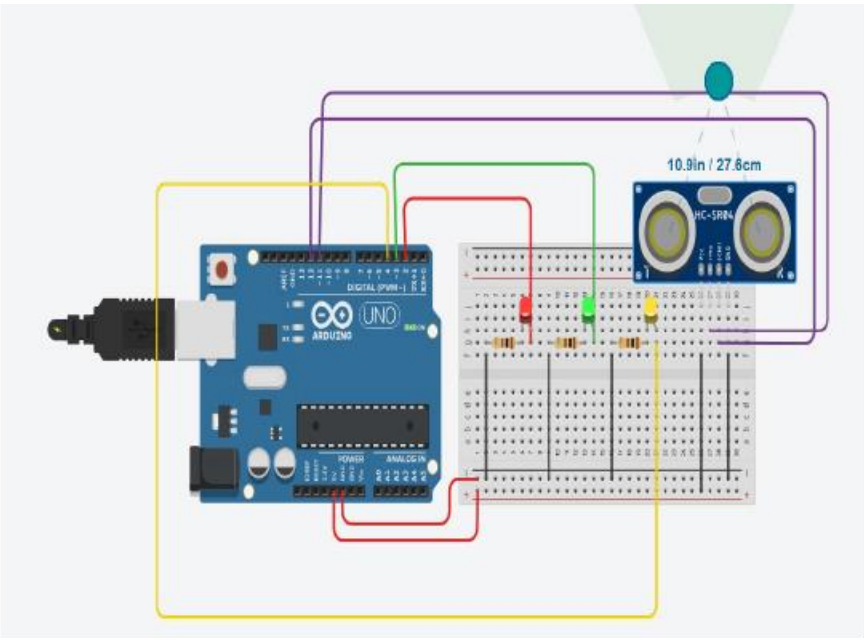


Fig.1. ALL Leds turned ON when a distance at a (0 to 30cm) distance

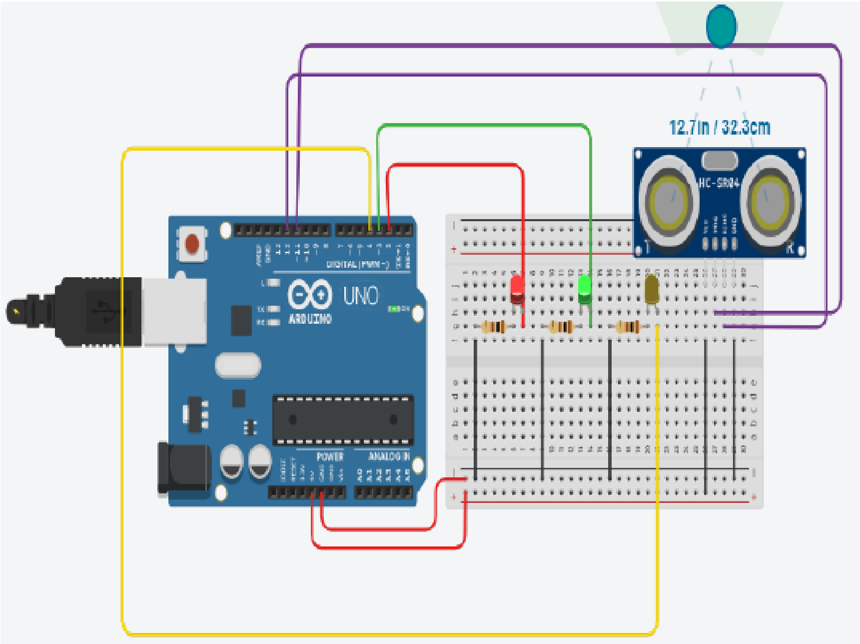


Fig2. Two Leds turned ON when a distance at a (31 to 50cm) distance

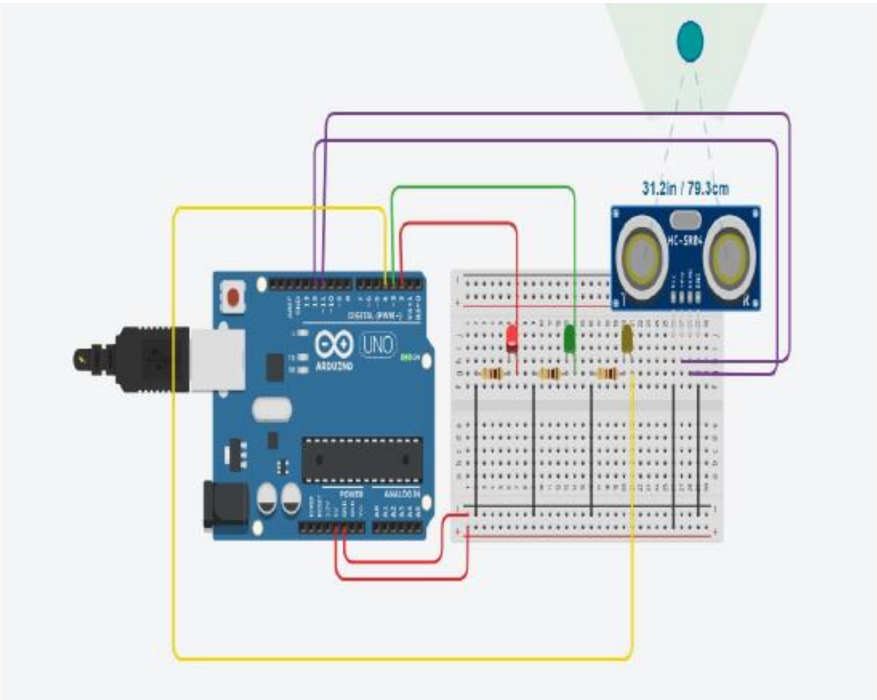


Fig:3. One Led turned ON when a distance at a (51 to 80cm) distance

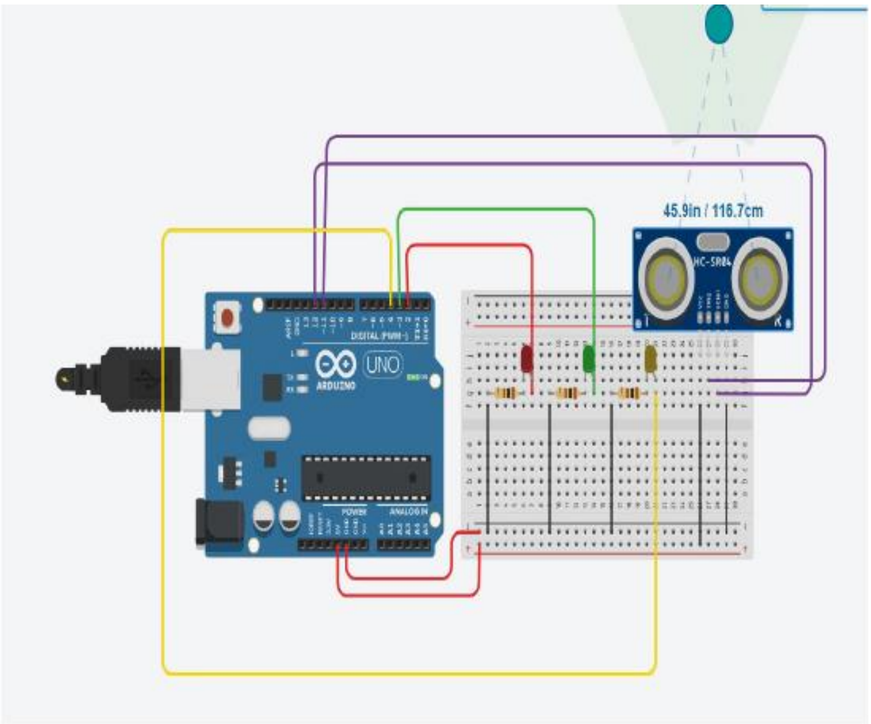


Fig.4. ALL Leds turned OFF when a distance at a (81cm and greater) distance

Discussion:

In this experiment, a sonar sensor was used to detect the distance of an object or obstacle and the distance between the object and sonar was displayed using the LEDs that were connected. The sonar sensor that was used in this experiment were studied carefully before using it. The pin operations and how the sonar works were observed and carefully understood. After that, the sonar sensor was set accordingly with the Arduino Uno board accordingly. The LEDs were set as per necessity. After that, the circuit was operated and the systems operations were observed. The method of how the system was detecting the distance between the object and sonar were observed. The method of how it was performed using Trigger and Echo was observed accordingly as well and co-related with the experiment's working methodology. The LEDs were turning ON accordingly based on the distance and formulas that were set on the code. Changes in the detection mechanism was also observed by changing the detecting parameters accordingly. All the results that were observed were carefully noted down for further evaluation. The similar system was developed on the simulation softwares like TinkerCad. The results that were obtained on the physical operation were evaluated with the simulated outcomes. There were some minor discrepancies that were observed. The distance that were generated on the simulation's serial monitor were a bit different compared to the ones that were observed on the physical testing. This might be caused due to minor system and human errors. This caused the inconsistencies in the values of the serial monitor. Moreover, the detecting rate of the system in the physical environment and simulation virtual environment were a bit different. This was ruled as normal as human error was general in the physical world. From the observation it can be said that after both hardware and software implementation showed the expected outcomes and the experimental objectives was achieved.

Conclusion:

In conclusion, the experiment highlighted the importance of thorough understanding and careful implementation of hardware and software components in sensor-based systems. Additionally, it emphasized the need for validation through both physical testing and simulation to ensure the reliability and accuracy of the results obtained.

Reference(s):

[1] Arduino IDE, https://www.arduino.cc/en/Main/Software accessed on May 3, 2019.
[2] Arduino and Proteus Library, https://etechnophiles.com/add-simulate-ultrasonic-sensorproteus-2018-edition/ accessed on May 3, 2019.
[3] Ultrasonic Distance Sensor in Arduino With TinkerCad https://www.instructables.com/id/Ultrasonic-Distance-Sensor-Arduino-Tinkercad/ accessed on May 3, 2019.

