AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 06

Experiment Title: Interfacing the Arduino with an external sensor to establish communication using the RS-232 protocol with implementing an obstacle detection system.

Date of Perform:	14 MARCH, 2023	**Date of Submission:**	21 MARCH, 2023
Course Title:	MICROPROCESSOR AND EMBEDDED SYSTEMS LAB		
Course Code:	COE3104	**Section:**	A
Semester:	SPRING 2022-23	**Degree Program:**	BSc in EEE
Course Teacher:	MD ALI NOOR		

GROUP-6

Sl No	Name	ID	PROGRAM	SIGNATURE
1.	Syed Aftab Uddin	19-41522-3	BSc in EEE	AFTAB
2.	Maimona Rahman Farjana	20-42954-1	BSc in CSE	MAIMONA
3.	Md Sajidul Haque Sohan	20-42022-1	BSc in CSE	SOHAN
4.	Md Sumon	20-42556-1	BSc in CSE	SUMON
5.	Md Rashedul Islam	20-43301-1	BSc in CSE	RASFEDUL
6.	Medha Chowdhury	20-41930-1	BSc in CSE	MEDHA
7.	Syeda Aynul Karim	19-41829-3	BSc in CSE	KARIM

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table Of Contents Page No.

Abstract

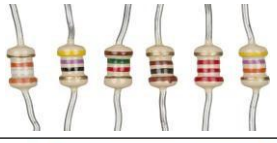
The main goal of this experiment is to connect the Arduino to an external sensor and use the RS-232 protocol to communicate while implementing an obstacle detection system. The Arduino IDE and Tinkercad software were used in this experiment. Initially, this experiment was carried out using the Arduino IDE software and hardware installed in the lab session. Tinkercad was used to carry out this experiment at home in order to better understand the simulation software.

Objectives

The objectives of this experiment are to-

1. Code a simple Obstacle Detection System in Arduino IDE.
2. Implement a simple Obstacle Detection System in Hardware.

Equipment List

No	Equipment Name	Figure
1	Arduino IDE (any version) Software	
2	Arduino Uno (R3) board	
3	Resistors	
4	Sonar Sensor (HCSR04)	

5	LED (Red, Green, Yellow)	
6	Tinkercad	

Circuit Diagram

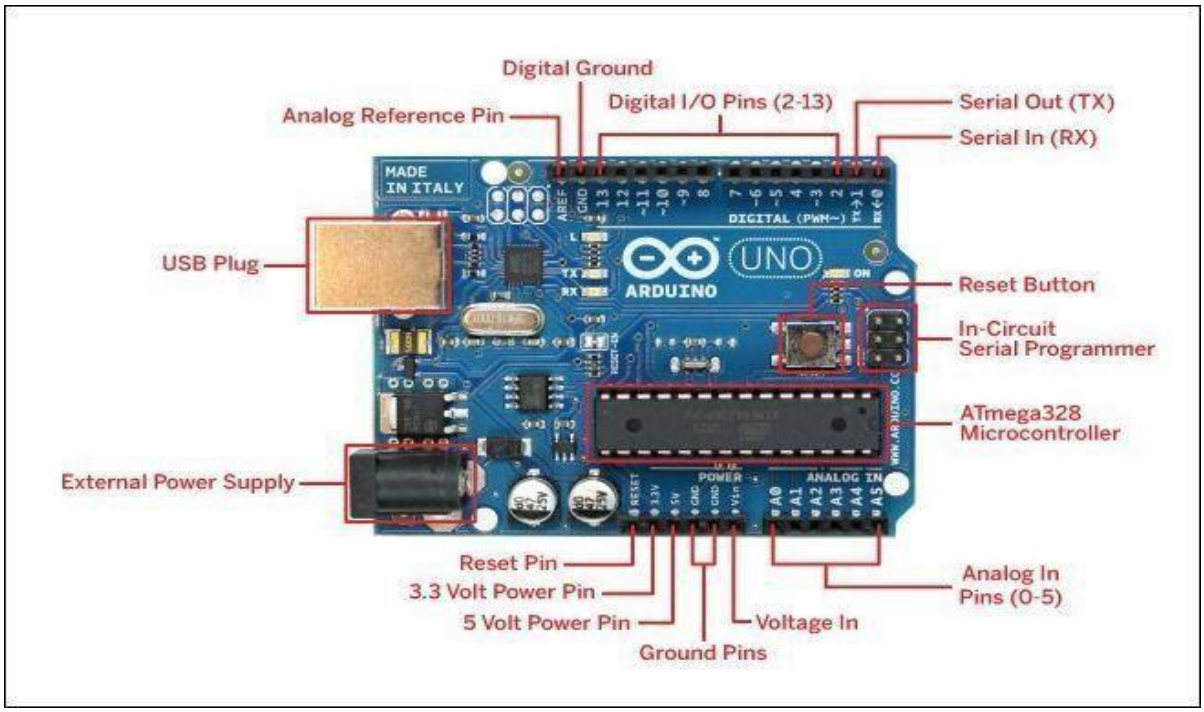


Figure 1: Arduino Board

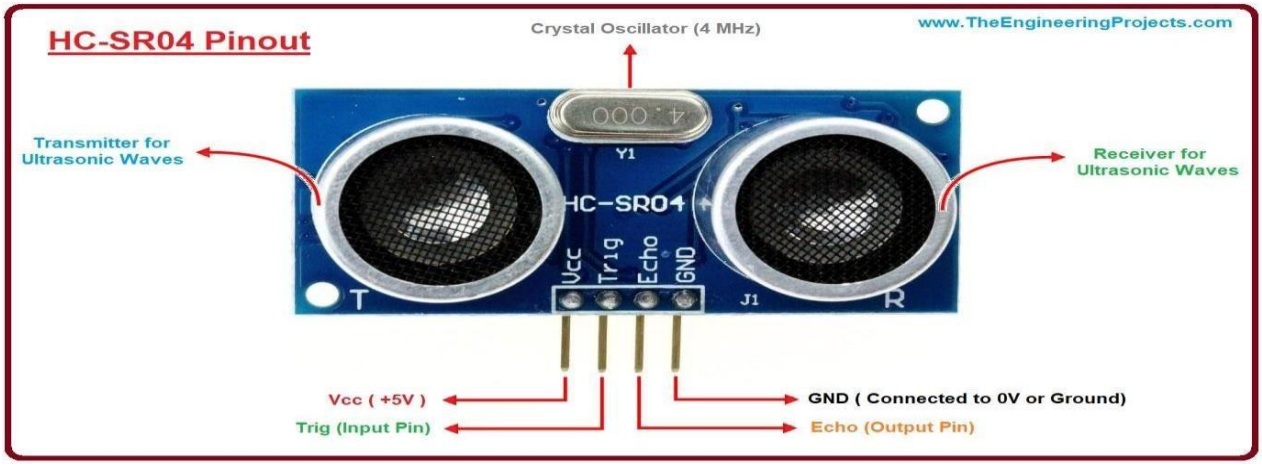


Figure 2: Sonar Sensor (HCSR04)

Hardware Set-Up

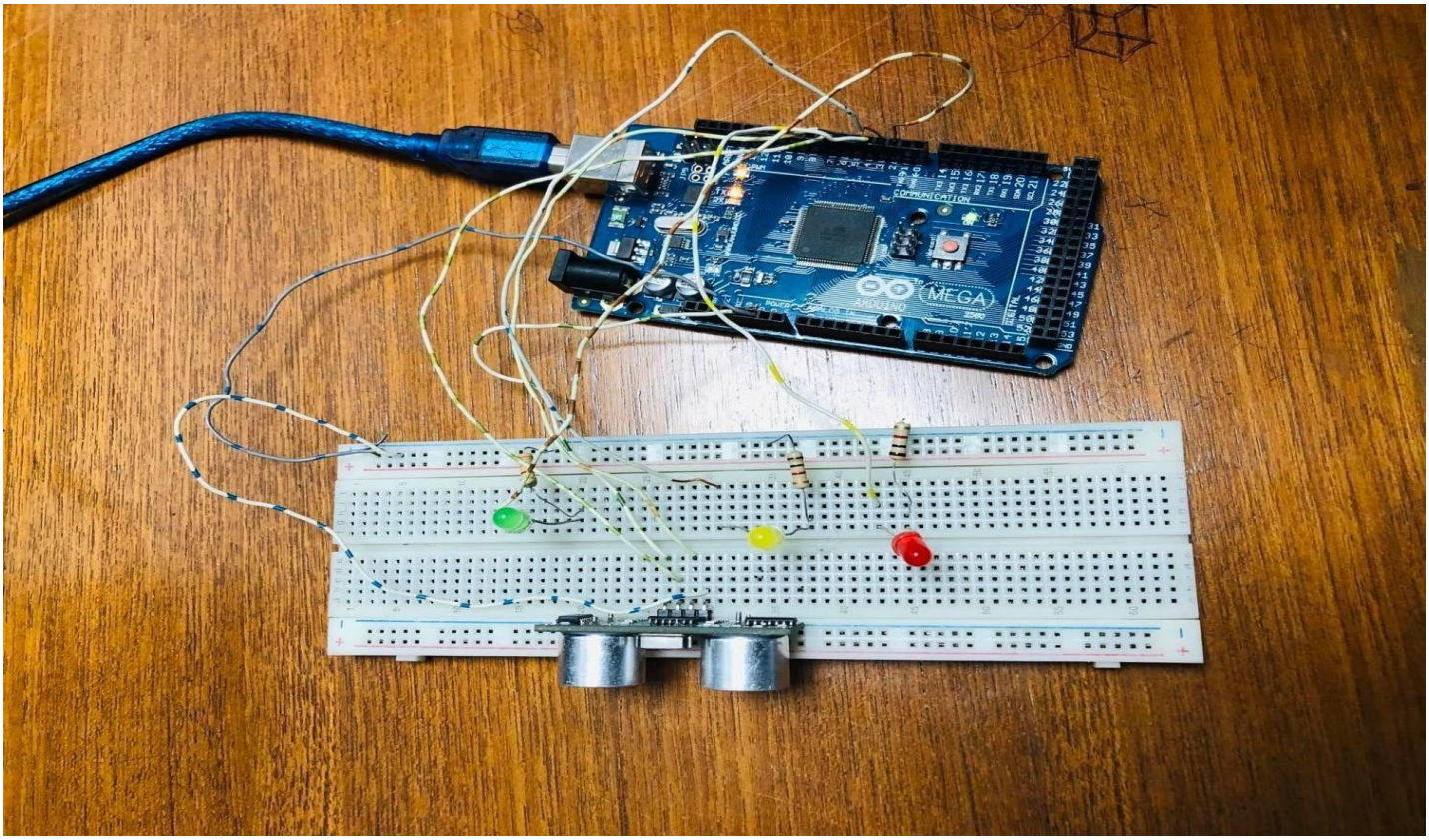


Figure 3: Hardware Set-Up

```
int distanceThreshold = 0;
int cm = 0;
int inches = 0;
long readUltrasonicDistance (int triggerPin, int echoPin)
{
pinMode (triggerPin, OUTPUT); // Clear the trigger
digitalWrite (triggerPin, LOW);
delayMicroseconds(2);
// Sets the trigger pin to HIGH state for 10 microseconds
digitalWrite (triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite (triggerPin, LOW);
pinMode (echoPin, INPUT);
// Reads the echo pin, and returns the sound wave travel time in microseconds
return pulseIn (echoPin, HIGH);
}
void setup()
{
Serial.begin(9600);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
}
void loop()
{
// set a threshold distance to activate LEDs
//considering the features of ultrasonic sensor
distanceThreshold = 50; // set distance at 50cm
// measure the ping time in cm, 340m/s=0.034cm/µs, therefore 0.034/2=0.017 mainly as the signal is working as echo
cm = 0.017 * readUltrasonicDistance(7, 6);
 // convert to inches by dividing by 2.54
inches = (cm / 2.54);
Serial.print(cm);
Serial.print("cm, ");
```

Done compiling.

Figure 4: Arduino IDE Code

Hardware Results

Figure 5: When Distance 35cm to 50cm(Green Light is on)

Figure 6: When Distance 20cm to 35cm(Yellow Light is on)

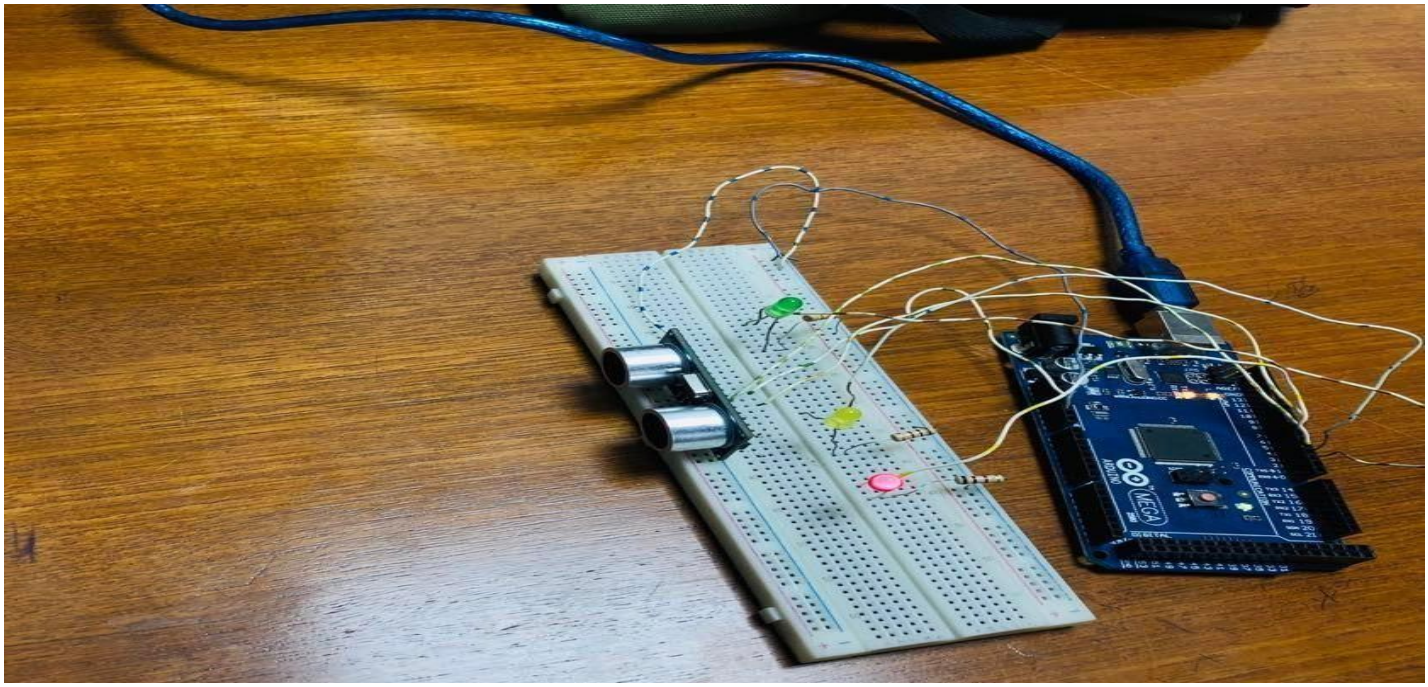


Figure 7: When Distance 0cm to 20cm(Red Light is on

Explanation Of Code

```
int distanceThreshold = 0; int cm = 0; int inches = 0;
long readUltrasonicDistance (int triggerPin, int echoPin)
{ pinMode (triggerPin, OUTPUT); // Clear the
trigger digitalWrite (triggerPin, LOW);
delayMicroseconds(2);
// Sets the trigger pin to HIGH state for 10 microseconds
digitalWrite (triggerPin, HIGH); delayMicroseconds(10);
digitalWrite (triggerPin, LOW); pinMode (echoPin,
INPUT);
// Reads the echo pin, and returns the sound wave travel time in microseconds return
pulseIn (echoPin, HIGH);
} void
setup()
{
Serial.begin(9600); pinMode(2,
OUTPUT); pinMode(3,
OUTPUT); pinMode(4,
OUTPUT);
} void
loop()
{
// set a threshold distance to activate LEDs //considering
the features of ultrasonic sensor distanceThreshold = 50; //
set distance at 50cm
// measure the ping time in cm, 340m/s=0.034cm/µs, therefore 0.034/2=0.017 mainly as the signal is working as echo
cm = 0.017 * readUltrasonicDistance(7, 6);
 // convert to inches by dividing by 2.54 inches
= (cm / 2.54)


Serial.print(cm);
Serial.print("cm, ");
Serial.print(inches);
Serial.println("in");
```

```
if (cm > distanceThreshold)
  { digitalWrite(2,
 LOW); digitalWrite(3,
 LOW); digitalWrite(4,
 LOW);
 }

 if (cm < distanceThreshold && cm > distanceThreshold-15 ) //for green light HIGH, distance 35cm to 50cm
 {          digitalWrite(2,
 HIGH); digitalWrite(3,
 LOW);  digitalWrite(4,
 LOW);
 }

 if (cm < distanceThreshold-15 && cm > distanceThreshold-30) //for yellow light HIGH, distance20cm to
 35cm
 {          digitalWrite(2,
 LOW);  digitalWrite(3,
 HIGH); digitalWrite(4,
 LOW);

 }
 if (cm < distanceThreshold-30 && cm > distanceThreshold-50) // for red light HIGH, distance 0cm to 20cm
 {
 digitalWrite(2,  LOW);
 digitalWrite(3,  LOW);
 digitalWrite(4, HIGH);
 }
 delay(100); // Wait for 100 millisecond(s)
 }
```

Simulation Set-Up

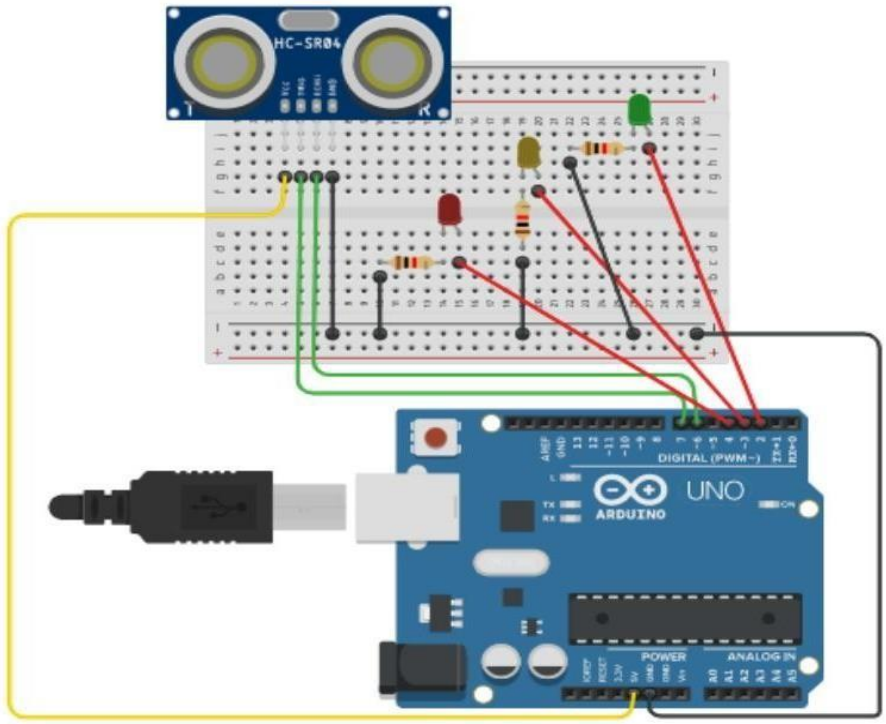


Figure 8: Implementing an obstacle detection system circuit design using Tinkercad

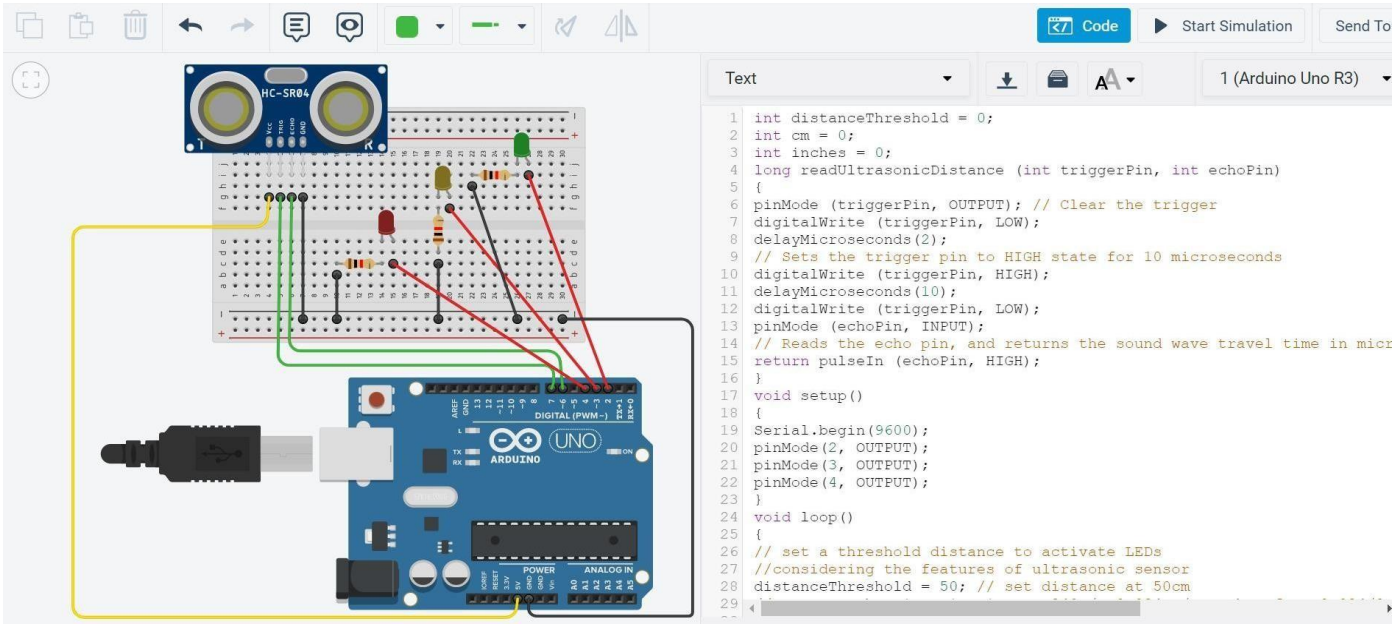


Figure 9: Obstacle detection system code implementation using Tinkercad

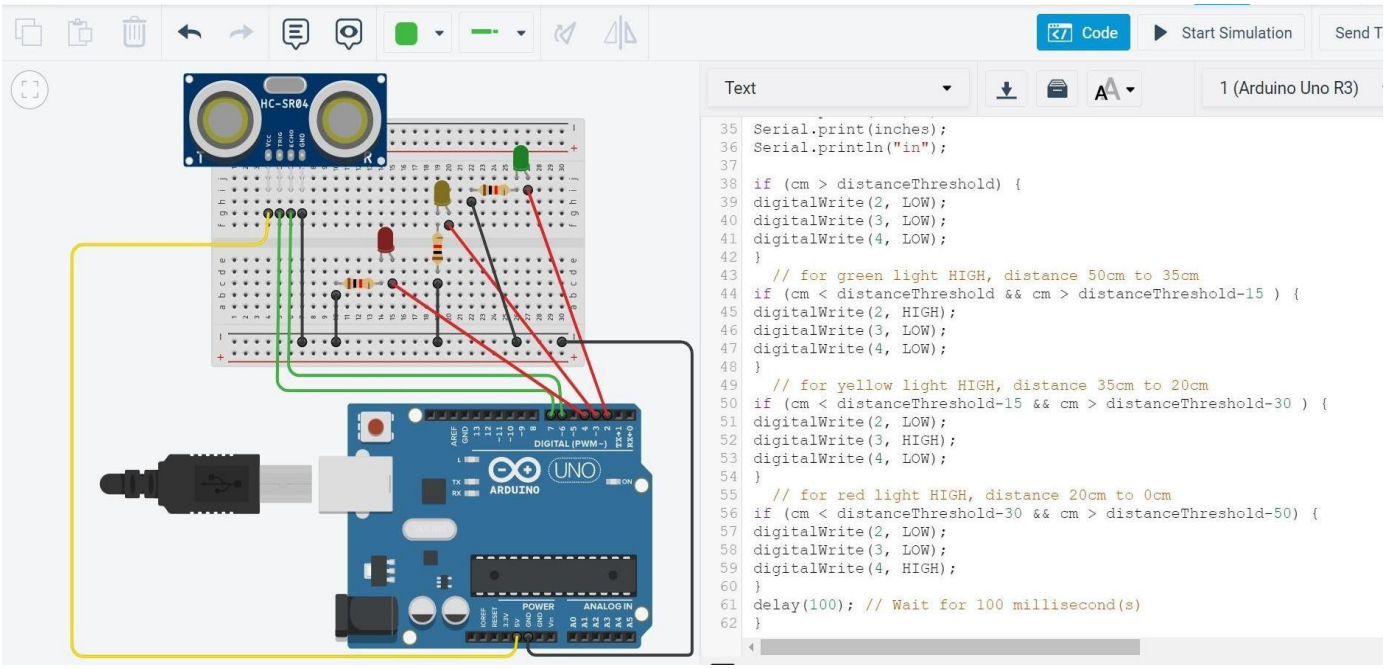


Figure 10: Obstacle detection system code implementation using Tinkercad

Simulation Results

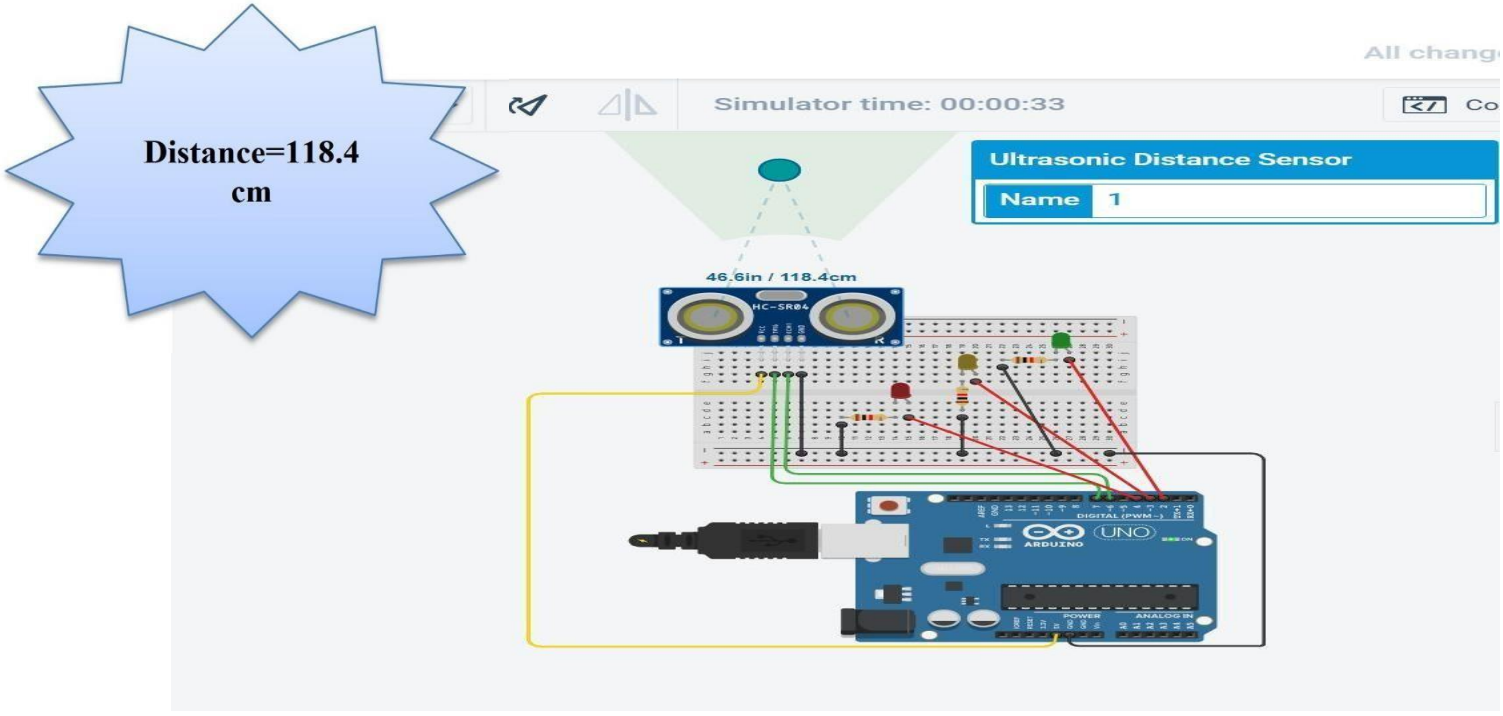


Figure 11: When Distance>50cm(No light is on)

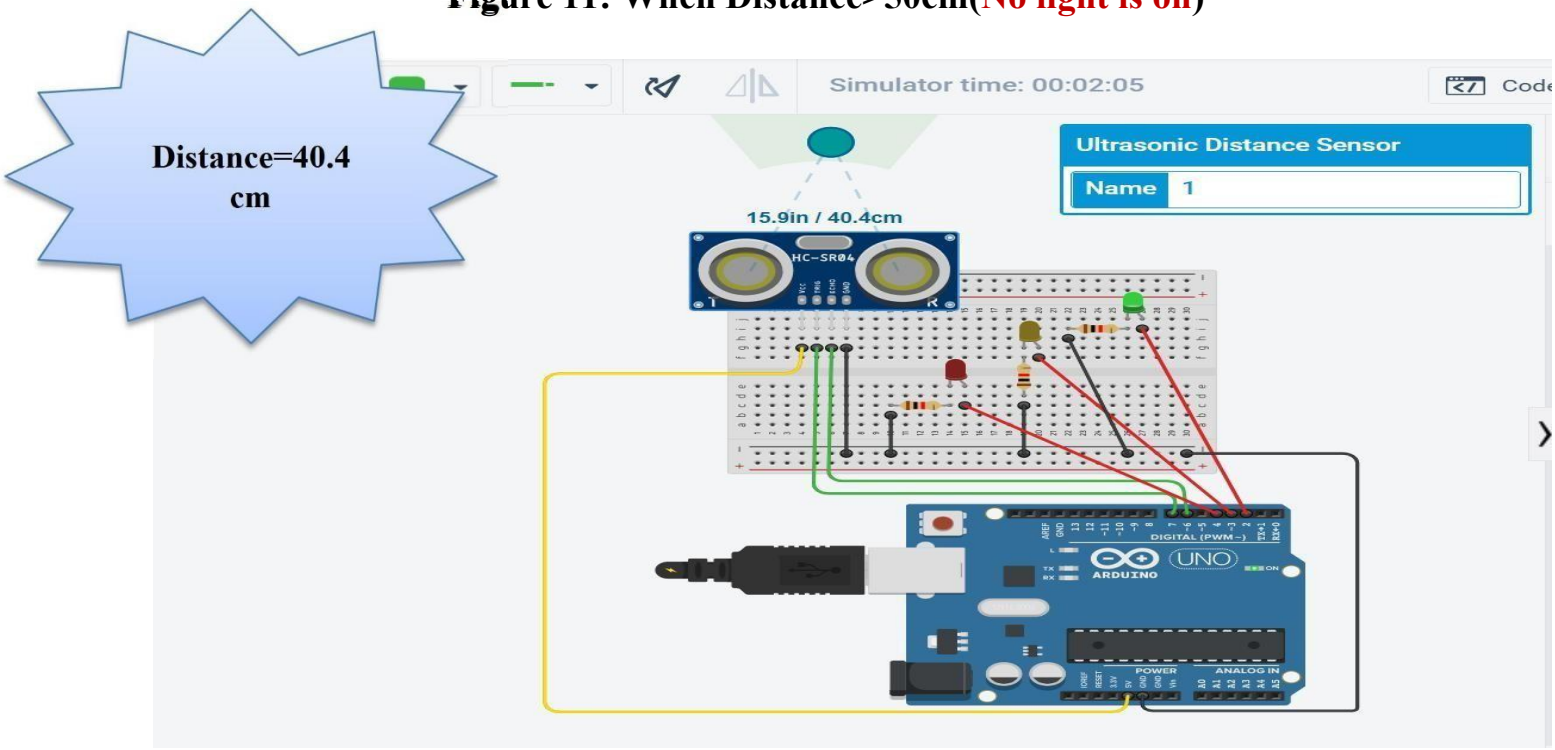


Figure 12: When Distance between 35 cm to 50cm(Green light is on)

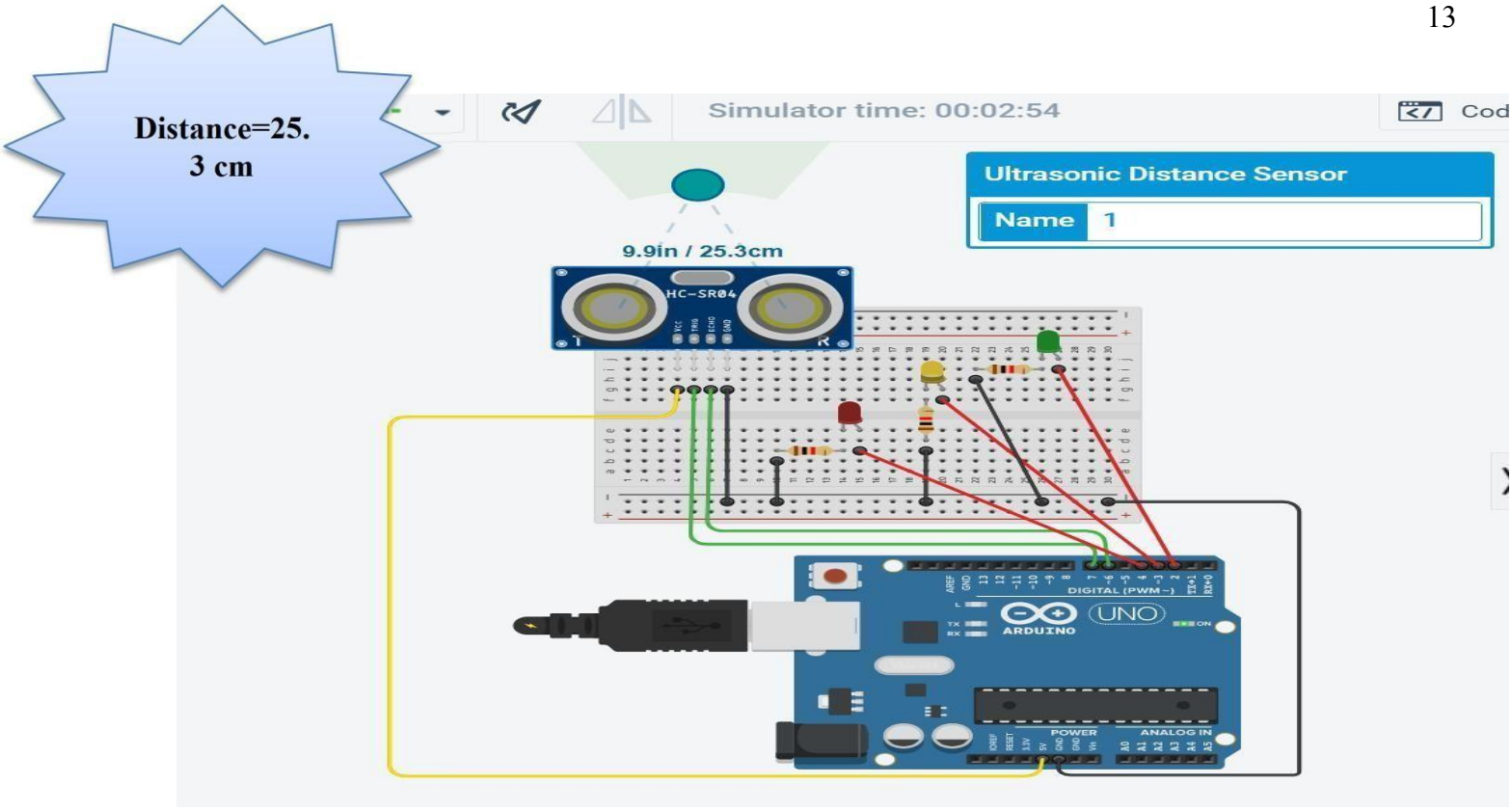


Figure 13: When Distance between 20 cm to 35cm(Yellow light is on)

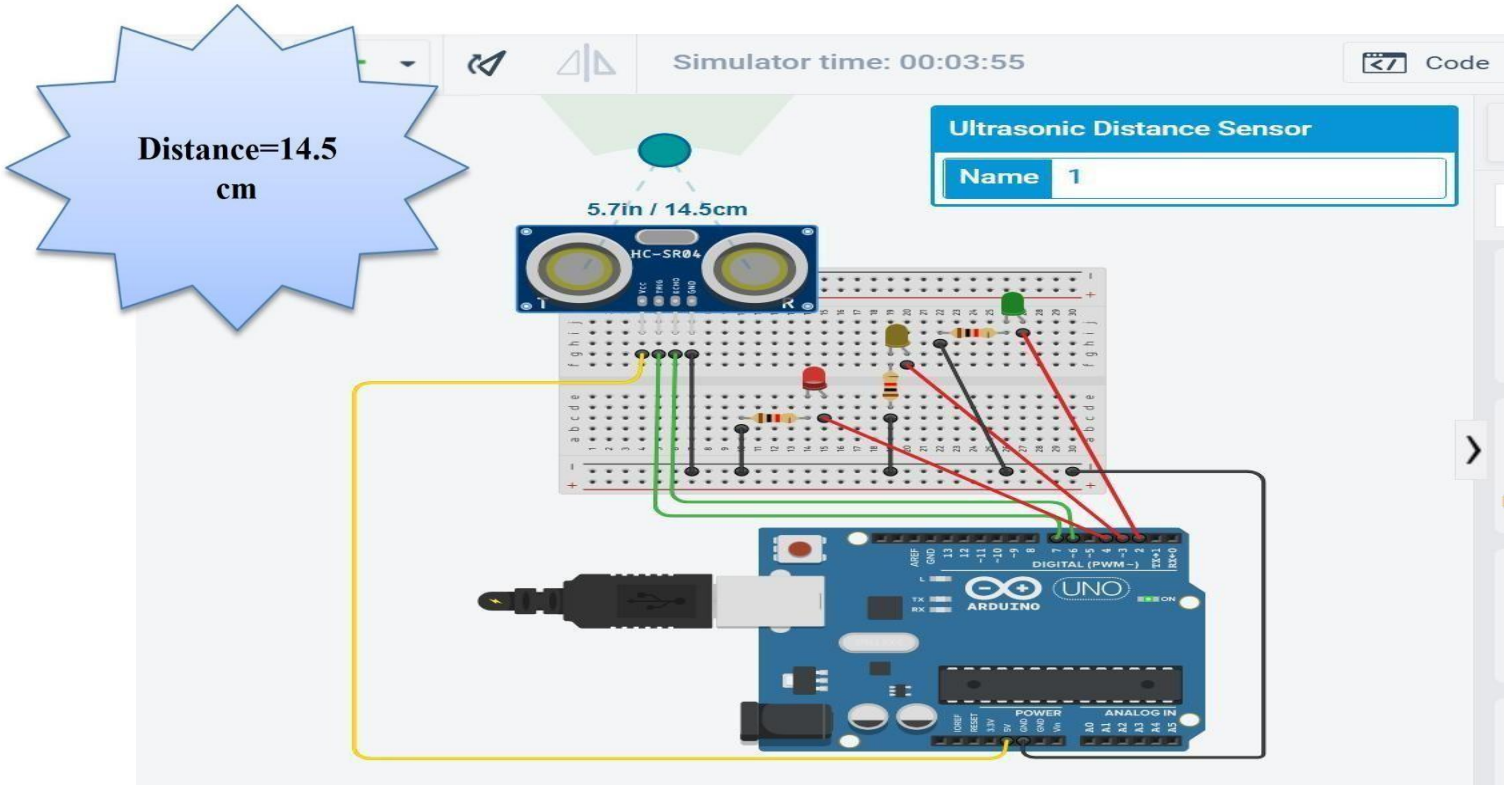


Figure 14: When Distance between 0 cm to 20cm(Red light is on)

Discussion

This experiment was conducted in two ways. First, the obstacle detection design was created using an Arduino board, three colored animated LED lights (red, yellow, green), three resistors, an HC-SR04 sensor breadboard, and connecting wires. Three LED lights were connected to ports 2, 3 and 4. The obstacle detection system code was then written using the Arduino IDE software. Then I connected the Arduino board to my computer and ran the code to achieve the desired result. In this experiment, we used the mouse as an obstacle, as shown in the figure in the Hardware Settings section. When the LED light is on / off, the cord distance is set between 0cm and 50cm. When the distance exceeded 50 cm, the LED light did not turn on. The green light came on when the mouse was somewhere between 35 cm and 50 cm. The yellow light came on when the distance between the mouse and the wall was between 20 cm and 35 cm. A red light came on when the distance between the mice was 0 cm to 20 cm. This experiment was performed using Tinkercad software for a better understanding. At the time of doing this experiment, I had some problems. There were some errors in the pin configuration and some errors in the code. However, these problems were solved with the help of experimental manuals and the internet. The results of runway lighting were successfully achieved in both ways.

Conclusion

The obstacle detection system has been successfully implemented and the desired results have been achieved. Therefore, it can be said that this experiment was carried out successfully.

References

[1] American International University Bangladesh-(AIUB) Lab Manual.

[2] Hardware Implementation of Obstacle Detection for Autonomous Vehicle
https://www.researchgate.net/publication/351993147

[3] Arduino IDE, https://www.arduino.cc/en/Main/Software

[4] Ultrasonic Distance Sensor in Arduino With Tinkercad,
https://www.tinkercad.com/things/kKLhDC67ui1-exp-06ftgroup-06/editel

[4] Hardware Implementation of Obstacle Detection for Autonomous Vehicle
https://www.ijariit.com/manuscripts/v7i3/V7I3-1473.pdf

[5] Object Detection System using Arduino and Android Application for Visually Impaired People
https://www.ijcaonline.org/archives/volume181/number15/karmakar-2018-ijca-917797.pdf