AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 07

Experiment Title:

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

Date of Perform:	7th May 2025	**Date of Submission:**	14th May 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EEE4103	**Section:**	**R**
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment is made.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized. by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that 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 offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 05

Sl No	Name	ID	PROGRAM	SIGNATURE
1	Shahriar Hossain	22-48990-3	BSc in CSE	
2	Al Mubtasim	22-49002-3	BSc in CSE	
3	Adiba Tanzila	22-49012-3	BSc in CSE	
4	MD.Rakib Hasan	22-49029-3	BSc in CSE	
5	Md.Imdadul Hasan(Ayon)	22-49959-3	BSc in CSE	

Faculty use only

FACULTY COMMENTS		
	Marks Obtained	
	Total Marks	

Table of Contents

Marking Rubrics (to be filled by Faculty):

Level Category	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]
Title and Objectives	Able to clarify the understanding of the lab, no issues are missing and formatting is good.	Able to clarify the understanding of the lab experiment, no issues are missing but its formatting is not good.	Able to clarify the understanding of the lab experiment, but a few issues are wrong, and its formatting is bad.	Able to clarify the understanding of the lab experiment, but it lacks a few important issues of the experiment without maintaining the format.	Unable to clarify the understanding of the lab experiment.	No Response/ copied from others/ identical submissions with gross errors/image file printed
Codes and Methods	Able to explain the experimental codes and simulation methods using Proteus very well.	Able to explain the experimental codes and simulation methods using Proteus but is not formatted well.	Able to explain the experimental codes but simulation method using Proteus is not explained well.	Presents the experimental codes but didn't explain simulation methods using Proteus clearly.	Presents the experimental codes but didn't explain simulation methods using Proteus.	
Results	Key results and images are there. Figures/Tables have all identifications and refer to them properly in the texts.	Key results and images are there. Figures/Tables have all identifications, such as the axis labels, numbers, and captions with a few minor errors; the texts refer them.	Key results and images are there. Figures/Tables lack a few identifications, such as the axis labels, numbers, and captions; the texts refer them.	Misses several key results and images. Figures/Tables lack identification, such as the axis labels, numbers, and captions; the texts don't refer them.	Major results, such as experimental and simulation results' images are not included. Figures and tables are poorly constructed or not presented.	
Discussion and Conclusion	Proper interpretation of results and summarizes the results to draw a conclusion, discusses its applications in real-life situations to connect with the report's conclusion.	Proper interpretation of results and summarizes the results to draw a conclusion but didn't discuss its applications in real-life situations to connect with the conclusion of the report.	Interpretation of results is presented. However, there is a disconnect between the results and discussion.	Misses the interpretation of key results. There is little connection between the results and discussion.	Very poor interpretation of the results. No connection between results and discussions.	
Question and Answer	Able to produce all questions' answers correctly maintaining the lab report format.	Able to produce all questions' answers but didn't maintain the lab report format.	Able to produce all questions' answers but wrong answers to a few questions.	Able to produce all questions' answers but wrong/missing answers to multiple questions.	Unable to produce all questions' answers and completely wrong answers.	
Comments						**Total Marks (25)**

Objectives:

The objectives of this experiment are to-

- Write code for a simple obstacle detection system in Arduino IDE.
- Implement a simple obstacle detection system using an Arduino microcontroller.

Equipment List:

1) Arduino IDE (any version)
2) Arduino Uno (R3) board
3) Sonar Sensor (HCSR04)
4) Breadboard
5) LEDs (red, green, and yellow)
6) Resistors of 100 ohm
7) Jumper wires

Circuit Diagram:

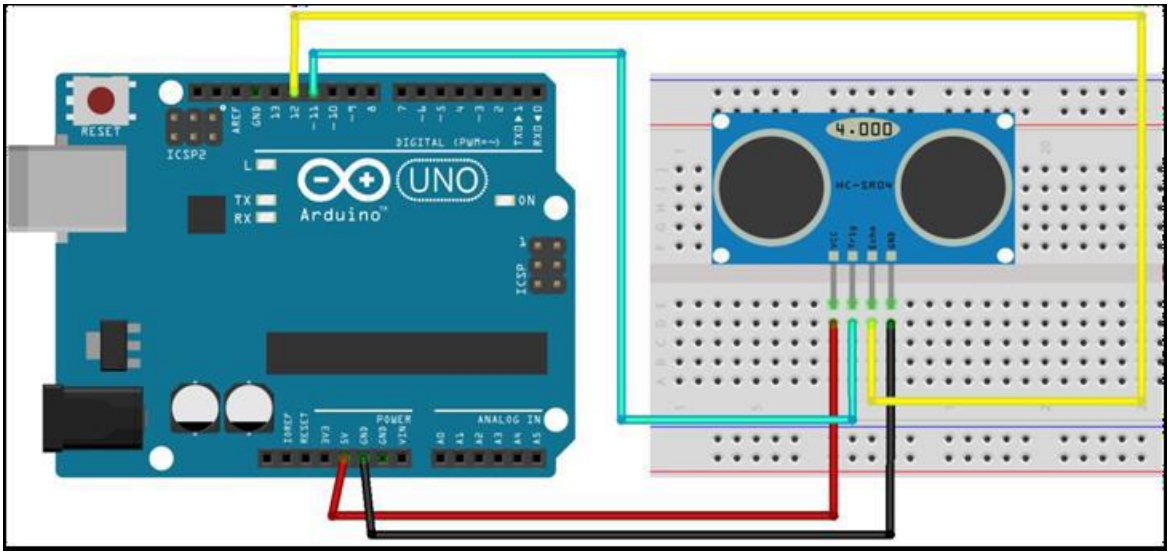
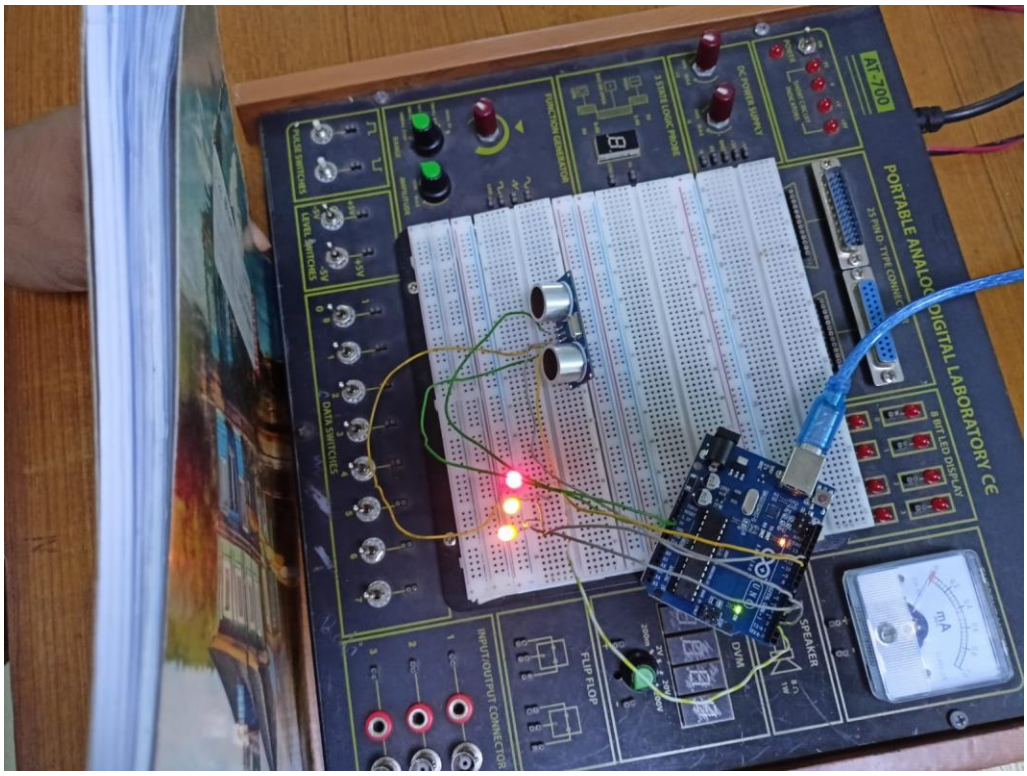


Fig : Arduino board's pin connections with the Sonar Sensors (schematic diagram)

Experimental Output Results:



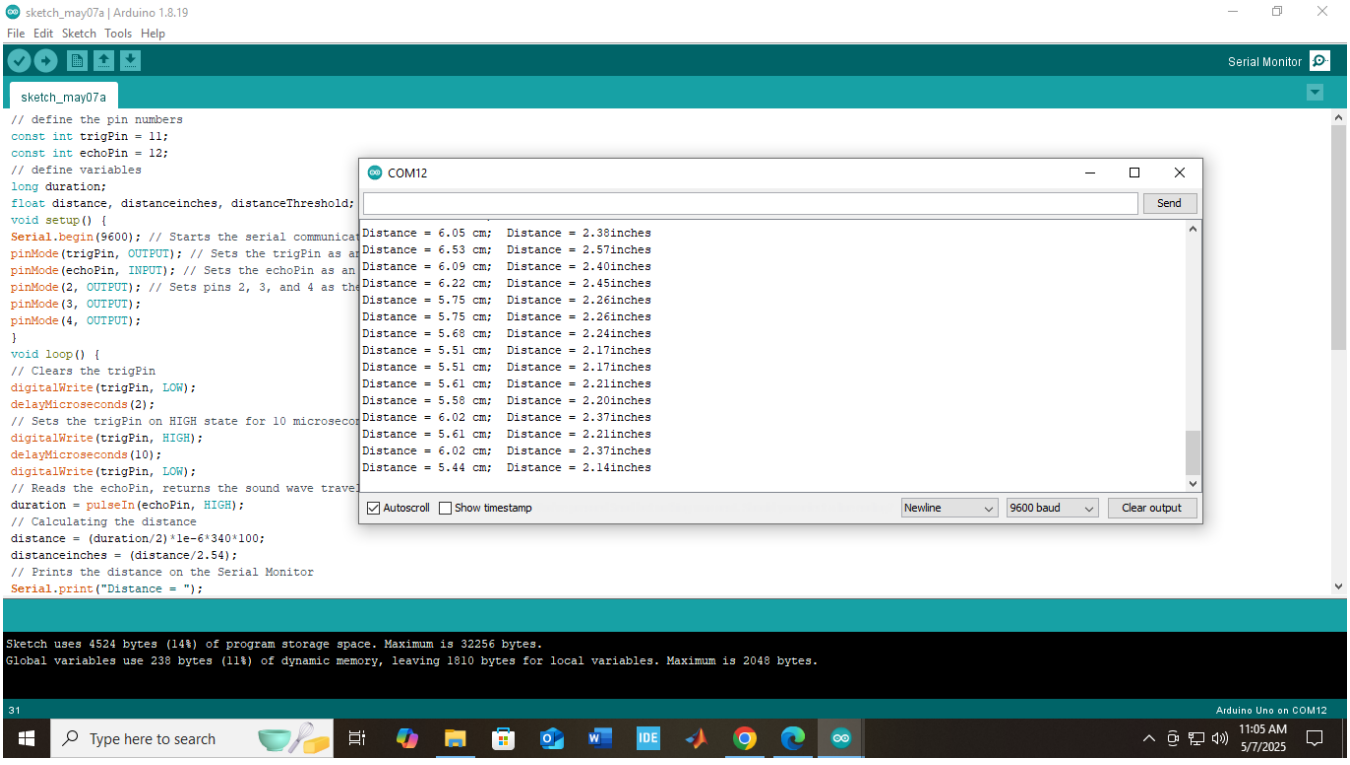


Fig. All LED ON at shown distance

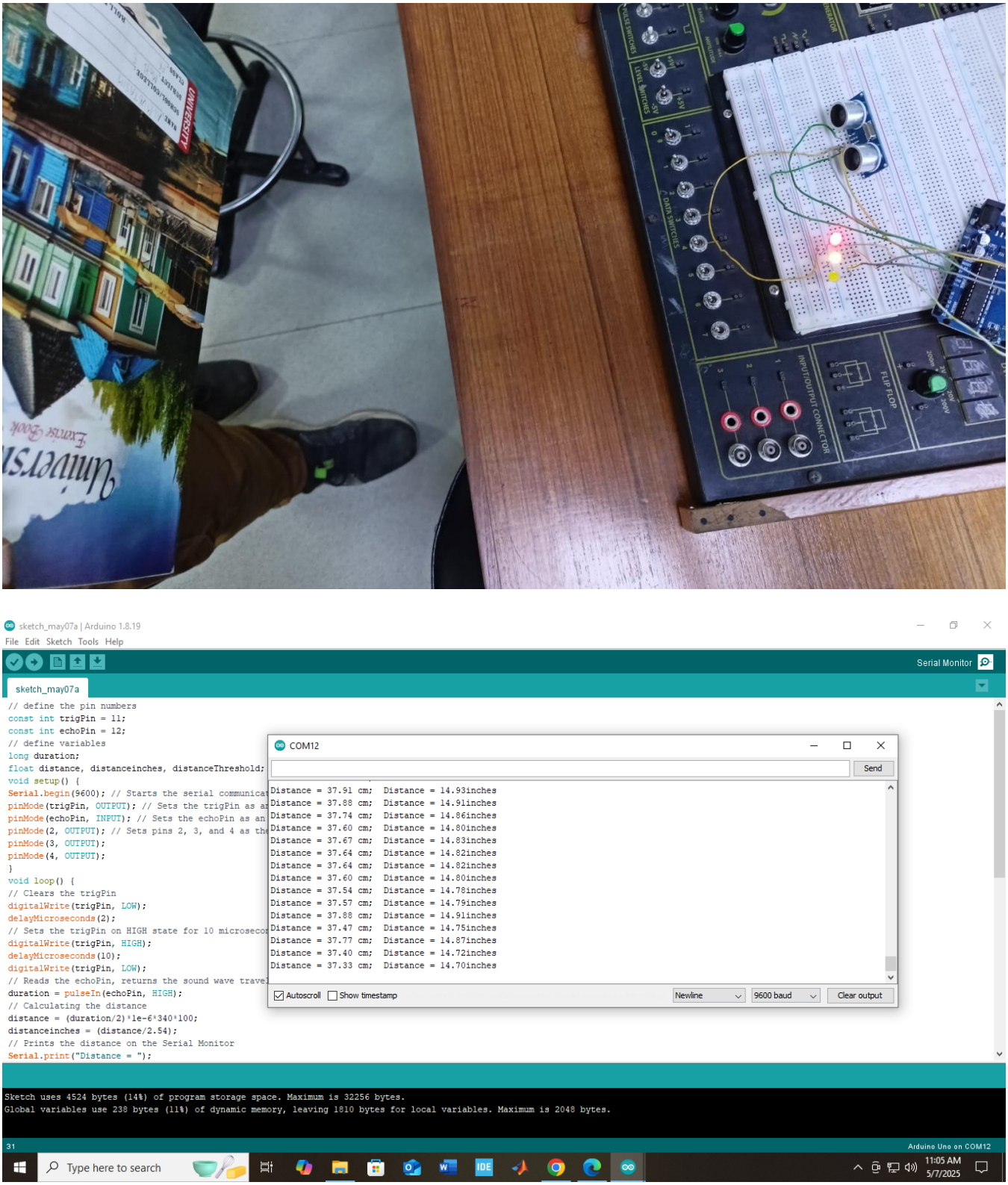


Fig. Two LED ON at shown distance

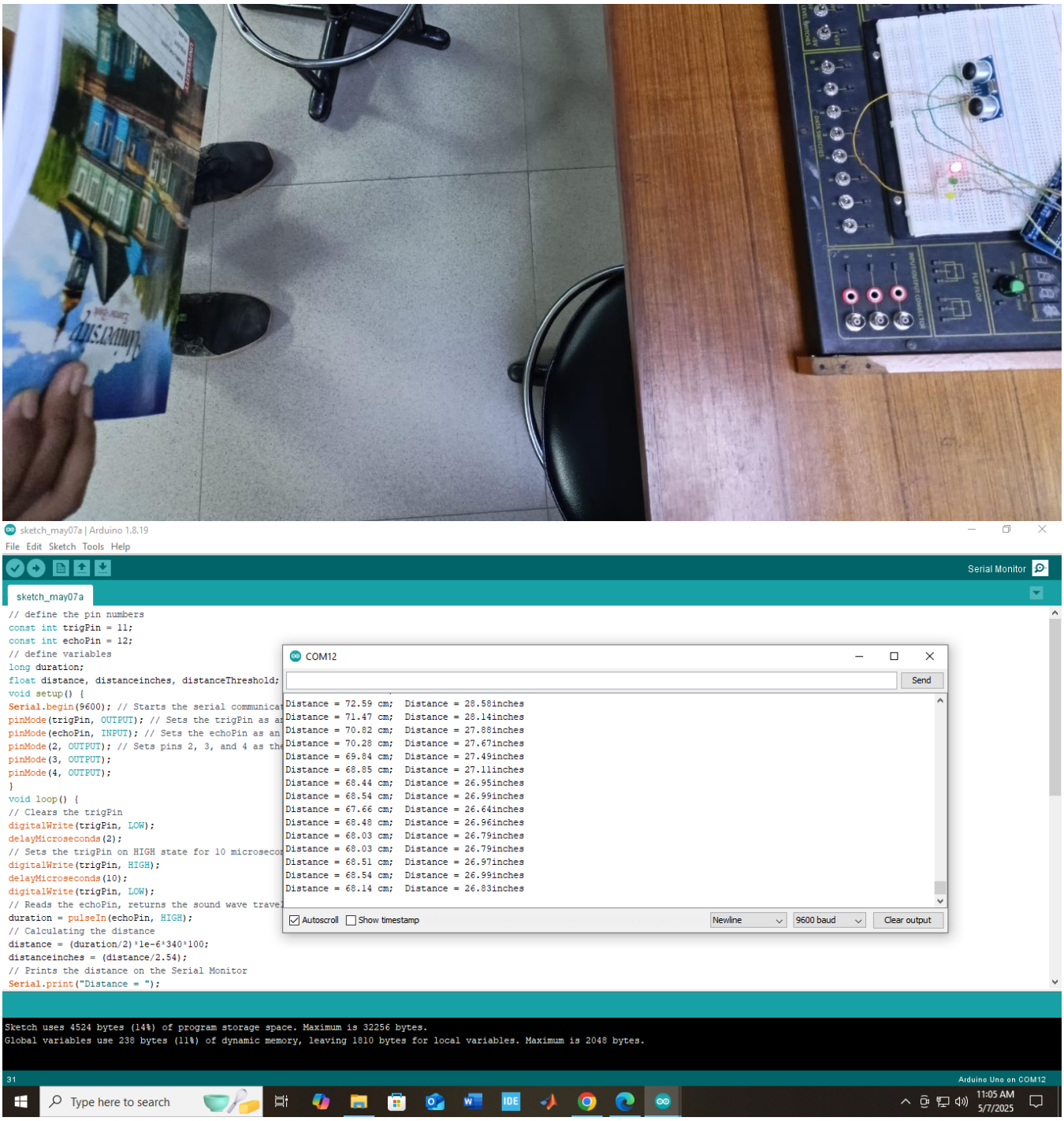


Fig. One LED ON at shown distance

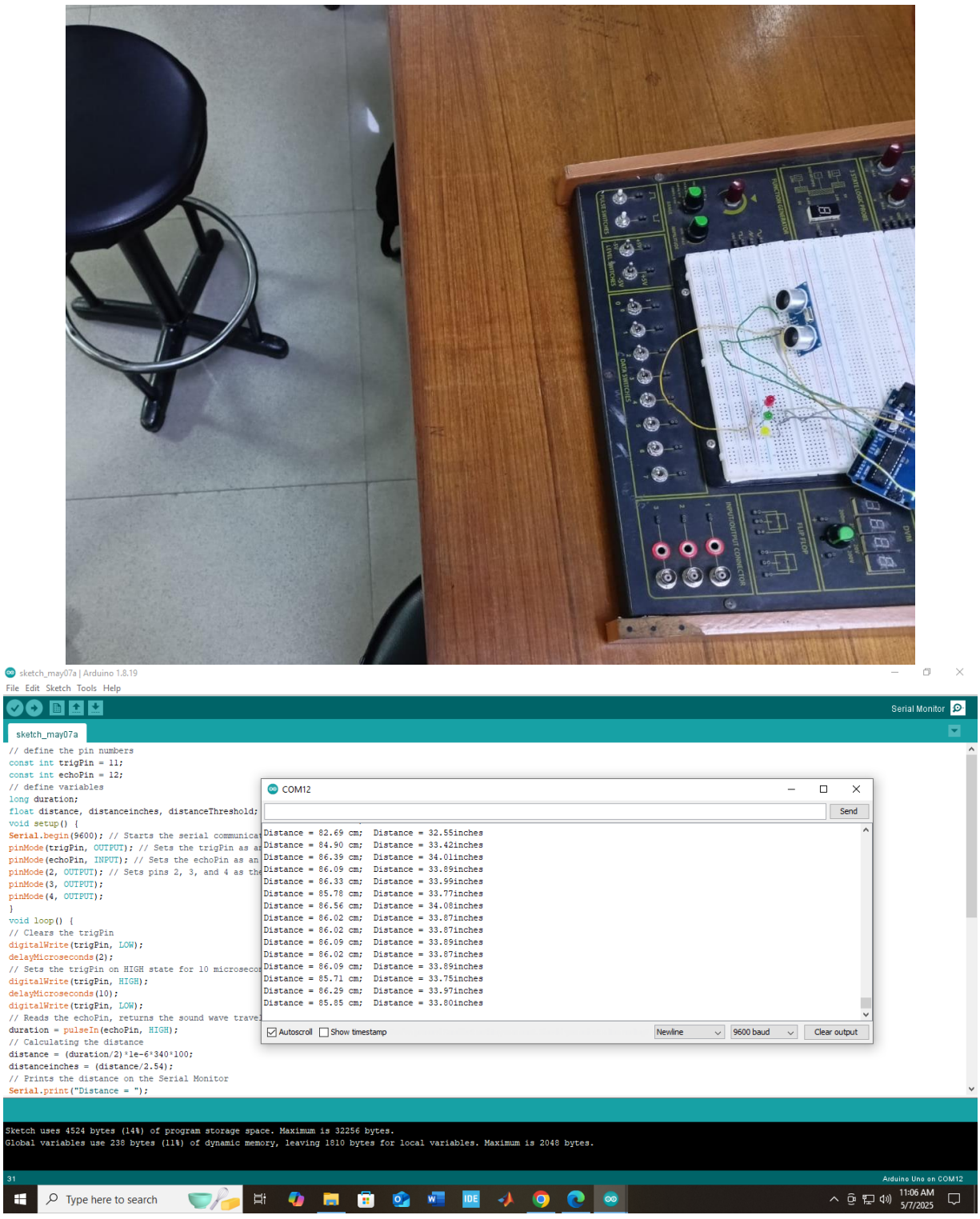


Fig. No LED ON at shown distance

Simulation Output Results:

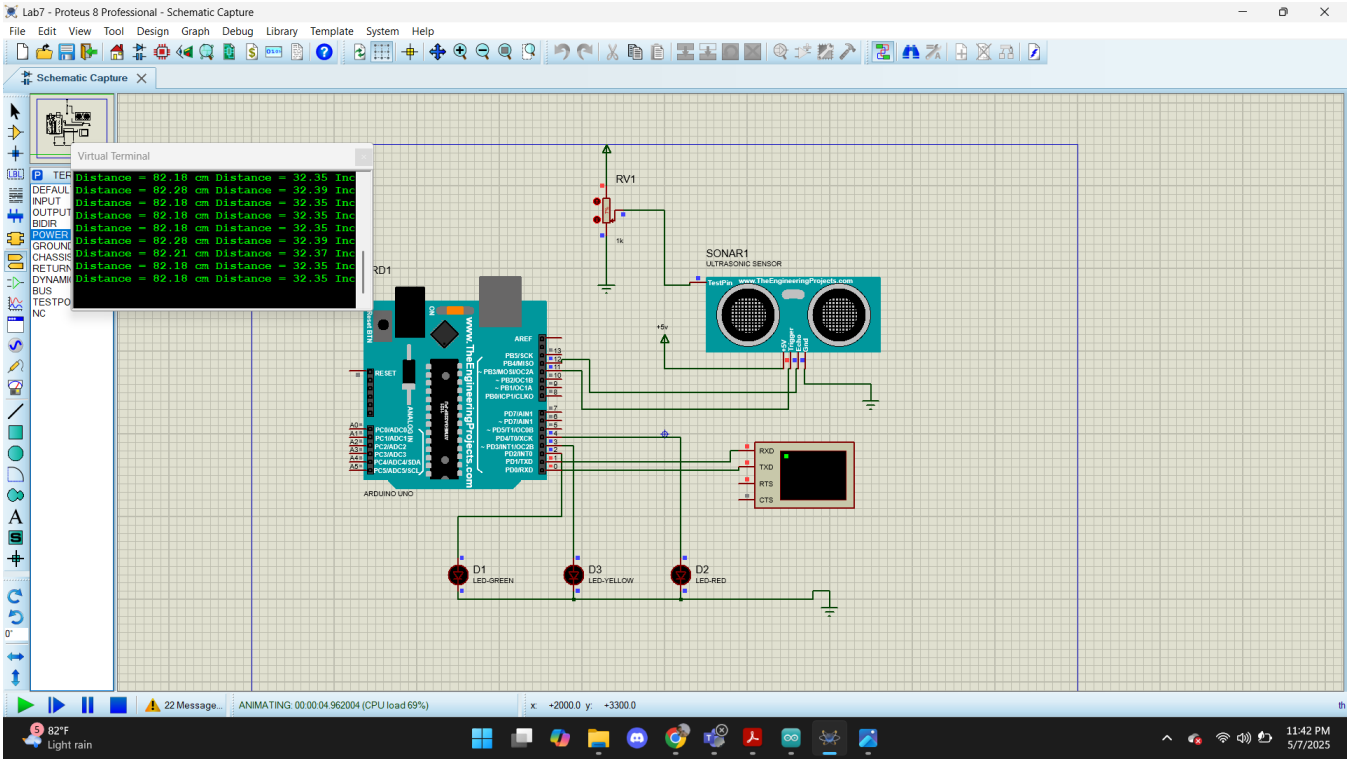


Fig: Simulation showing distance where No LED ON

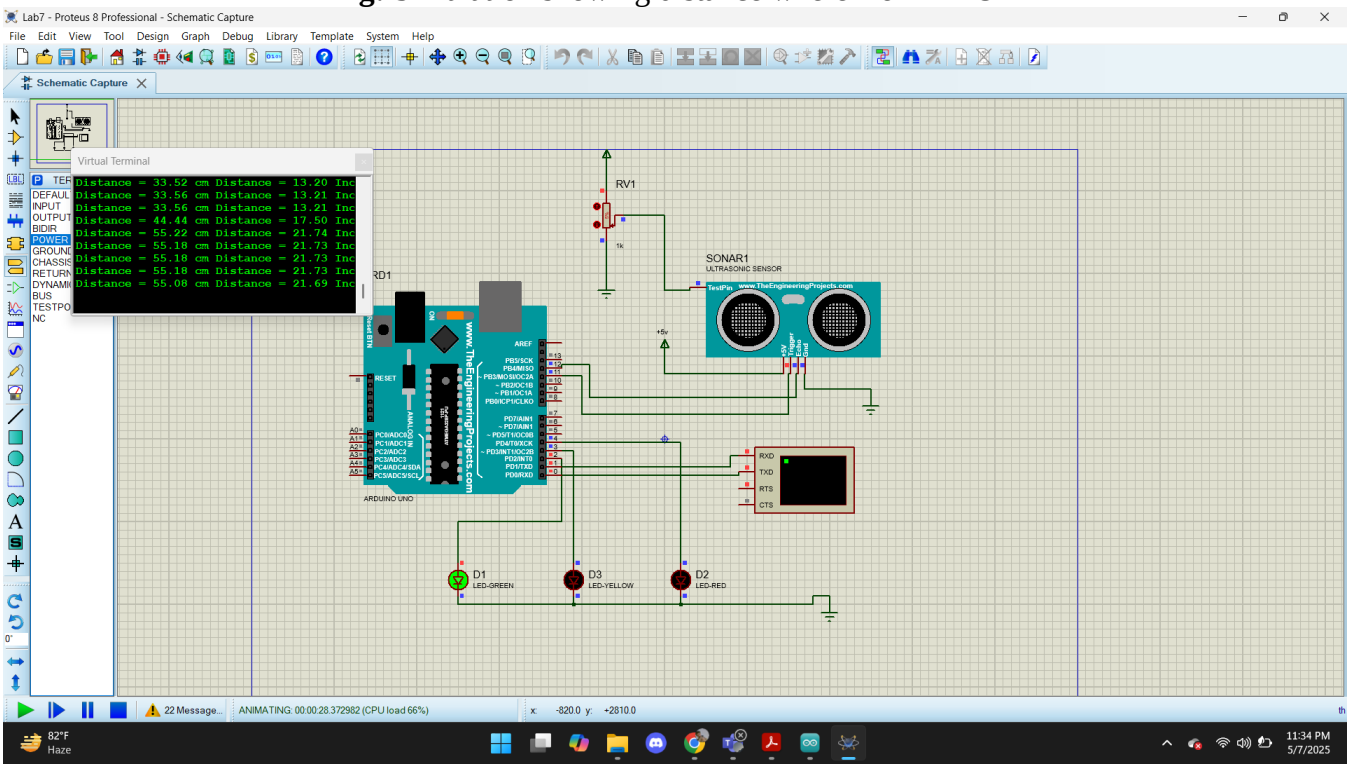


Fig: Simulation showing distance where One LED ON

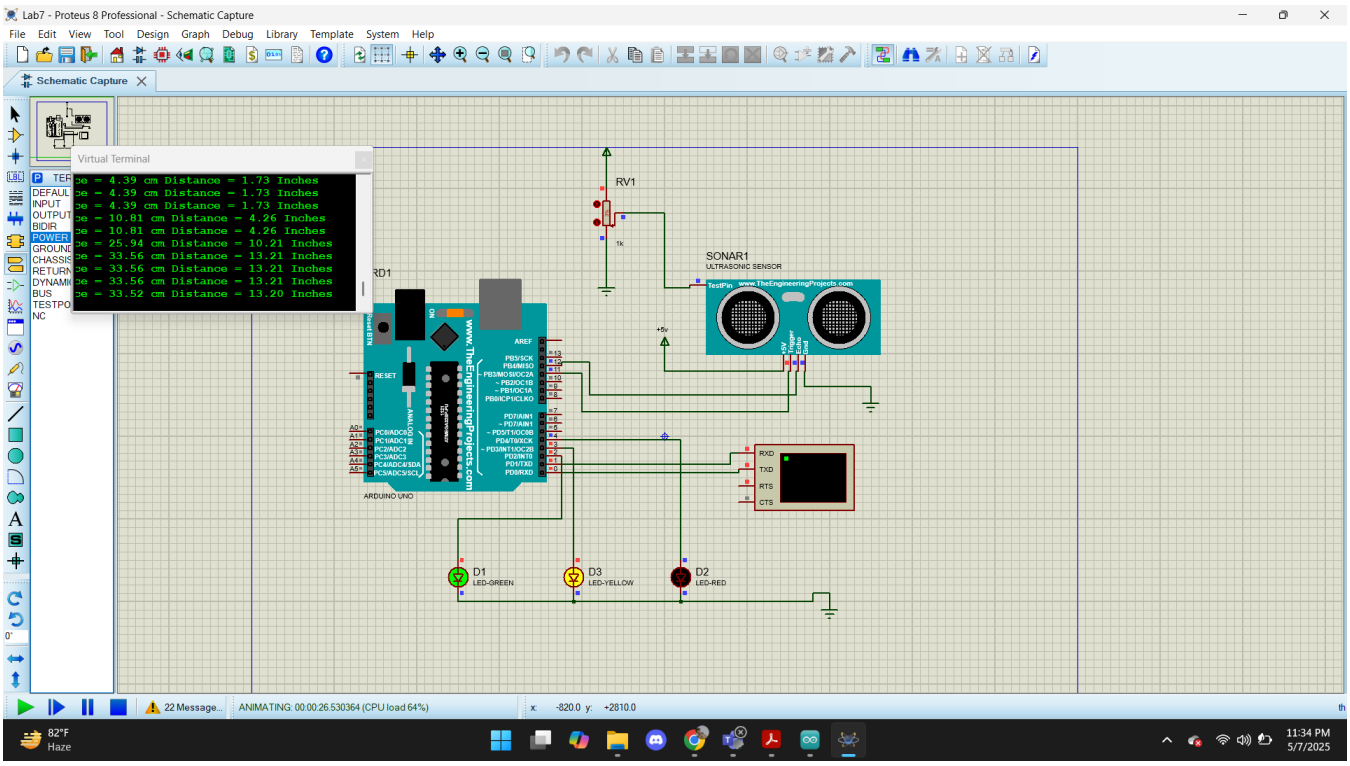


Fig: Simulation showing distance where 2 LEDs ON

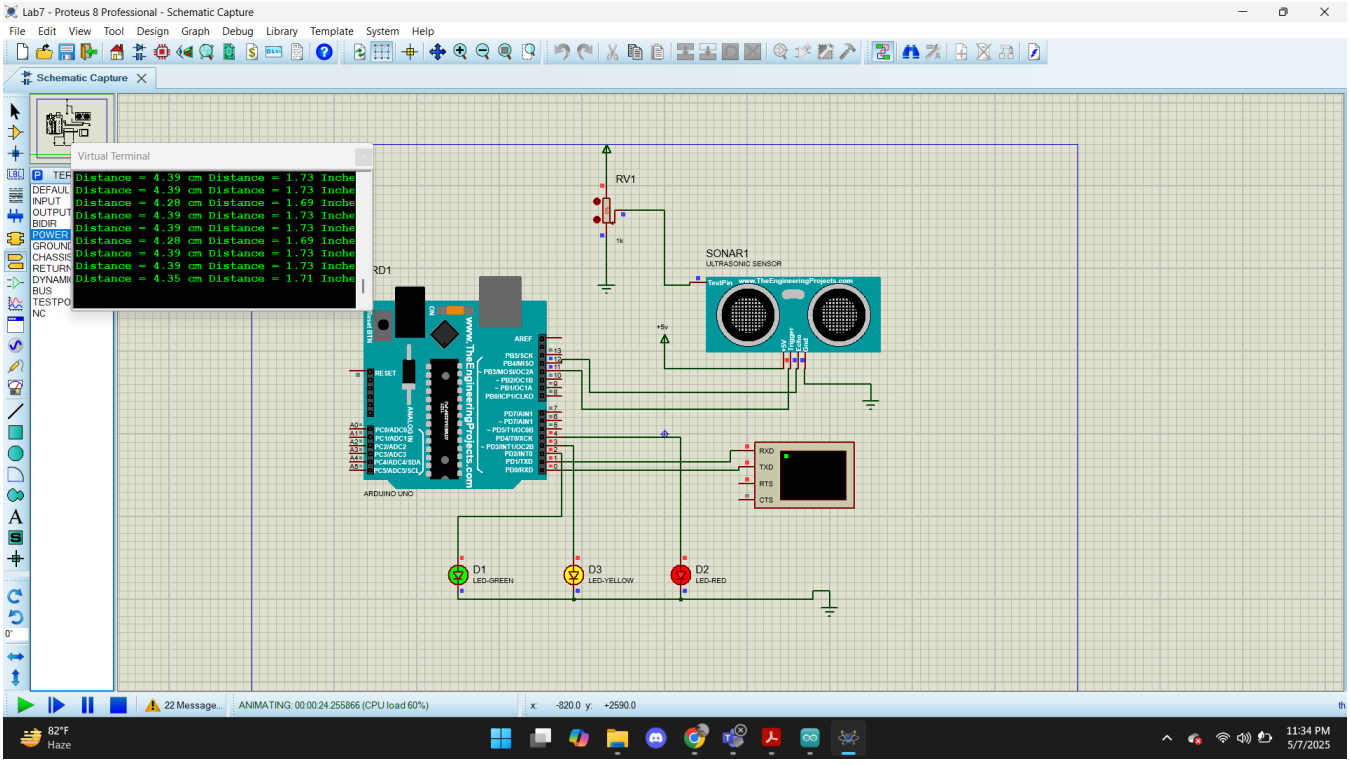


Fig: Simulation showing distance where 3 LEDs ON

Explaination:

- Proteus simulation software was used for circuit simulation.
- Arduino IDE was used to write and compile the code.
- The circuit was first designed in Proteus.
- A HEX file was generated from the Arduino IDE.
- The HEX file was then imported into Proteus.
- The simulation was run to observe the results.
- Sonar sensor behavior was tested by using the Interactive Potentiometer as a distance measurement.
- The LEDs' responses were observed to verify the system's operation.

Answers to the Questions in the Lab Manual:

```
// 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)
}
```

Explaination:

- const int trigPin = 11; and const int echoPin = 12; define the ultrasonic sensor pins.
- long duration; stores the time the ultrasonic signal takes to return.
- float distance, distanceinches, distanceThreshold; are used for storing the distance values and the threshold for LED control.

- **void setup()**
  - Serial.begin(9600); starts serial communication at 9600 baud to display distance data on the Serial Monitor.
  - pinMode(trigPin, OUTPUT); sets the trigPin as an output to send the pulse.
  - pinMode(echoPin, INPUT); sets the echoPin as input to receive the reflected pulse.
  - pinMode(2, OUTPUT);, pinMode(3, OUTPUT);, and pinMode(4, OUTPUT); set digital pins 2, 3, and 4 as output pins for controlling LEDs.
- **void loop()**
  - digitalWrite(trigPin, LOW); delayMicroseconds(2); ensures a clean LOW signal before sending the trigger pulse.
  - digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW); sends a 10µs HIGH pulse to start the ultrasonic measurement.
  - duration = pulseIn(echoPin, HIGH); reads the time taken by the ultrasonic pulse to return.

- distance = (duration/2)*1e-6*340*100; calculates the distance in centimeters using speed of sound.
- distanceinches = (distance/2.54); converts the distance from centimeters to inches.
- Serial.print(...) statements display the measured distance in both cm and inches on the Serial Monitor.
- distanceThreshold = 80; sets 80 cm as the maximum distance for LED activation logic.
- If the distance is more than 80 cm, all LEDs (pin 2, 3, 4) are turned OFF.
- If the distance is 50–80 cm, only LED on pin 2 is ON.
- If the distance is between 30–50 cm, LEDs on pin 2 and 3 are ON.
- If the distance is between 10–30 cm, all three LEDs (2, 3, 4) are ON.
- delay(200); adds a 200 ms pause before the next reading to prevent flickering.

Using ID values(22-49002-3, here assuming the mid values as: 49235(as 0 is used in monitor))

Code:

```
// define the pin numbers
const int trigPin = 9;
const int echoPin = 4;
// 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(5, 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(5, LOW);
}
if (distance < distanceThreshold && distance > distanceThreshold-30) {
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(5, LOW);
}
if (distance < distanceThreshold-30 && distance > distanceThreshold-50) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, LOW);
}
if (distance < distanceThreshold-50 && distance > distanceThreshold-70 ) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
}
delay(200); // Wait for 200 millisecond(s)
}
```

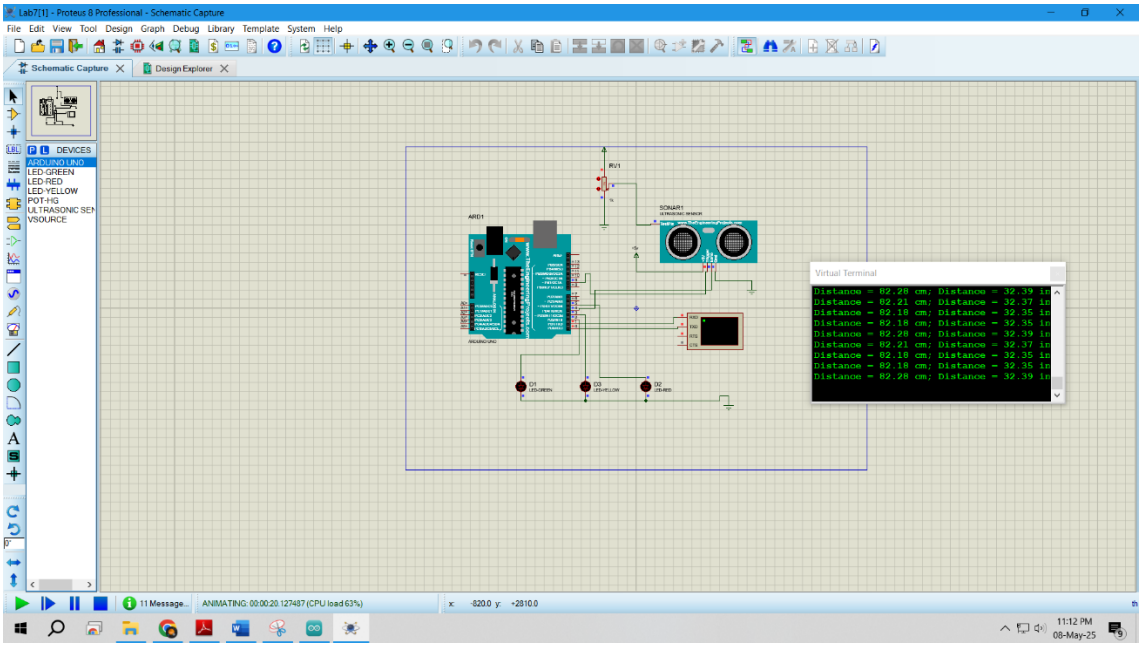


Fig: No LED on in shown distance(Using ID values)

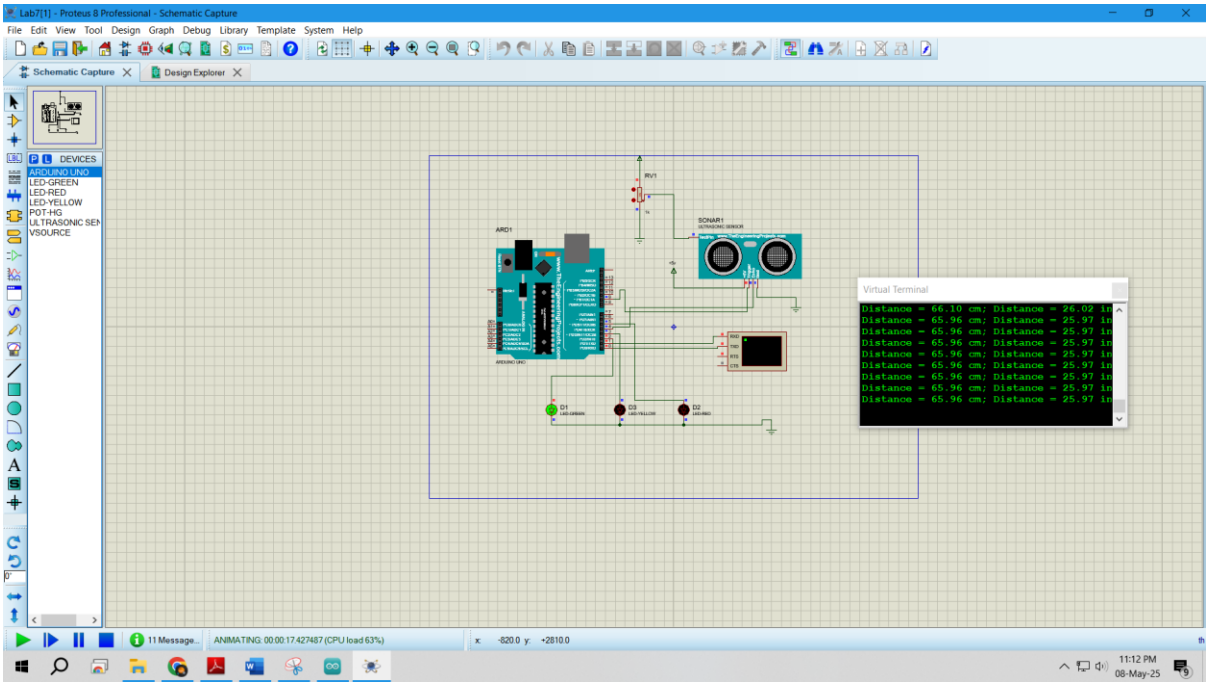


Fig: 1 LED on in shown distance(Using ID values)

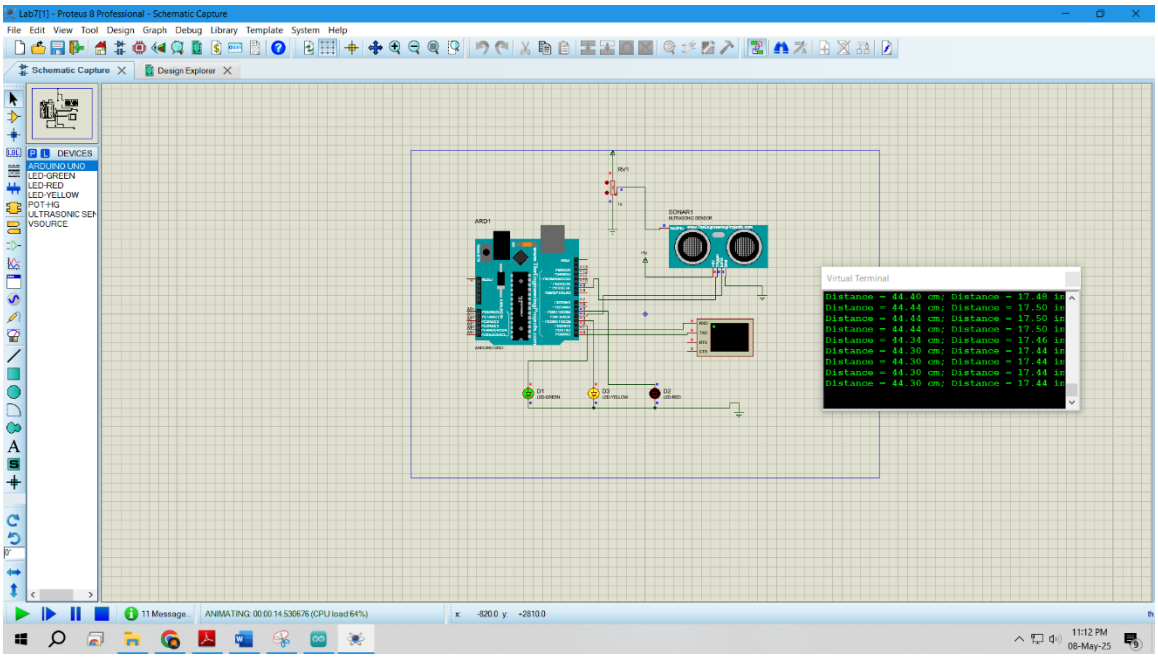


Fig: 2 LED on in shown distance(Using ID values)

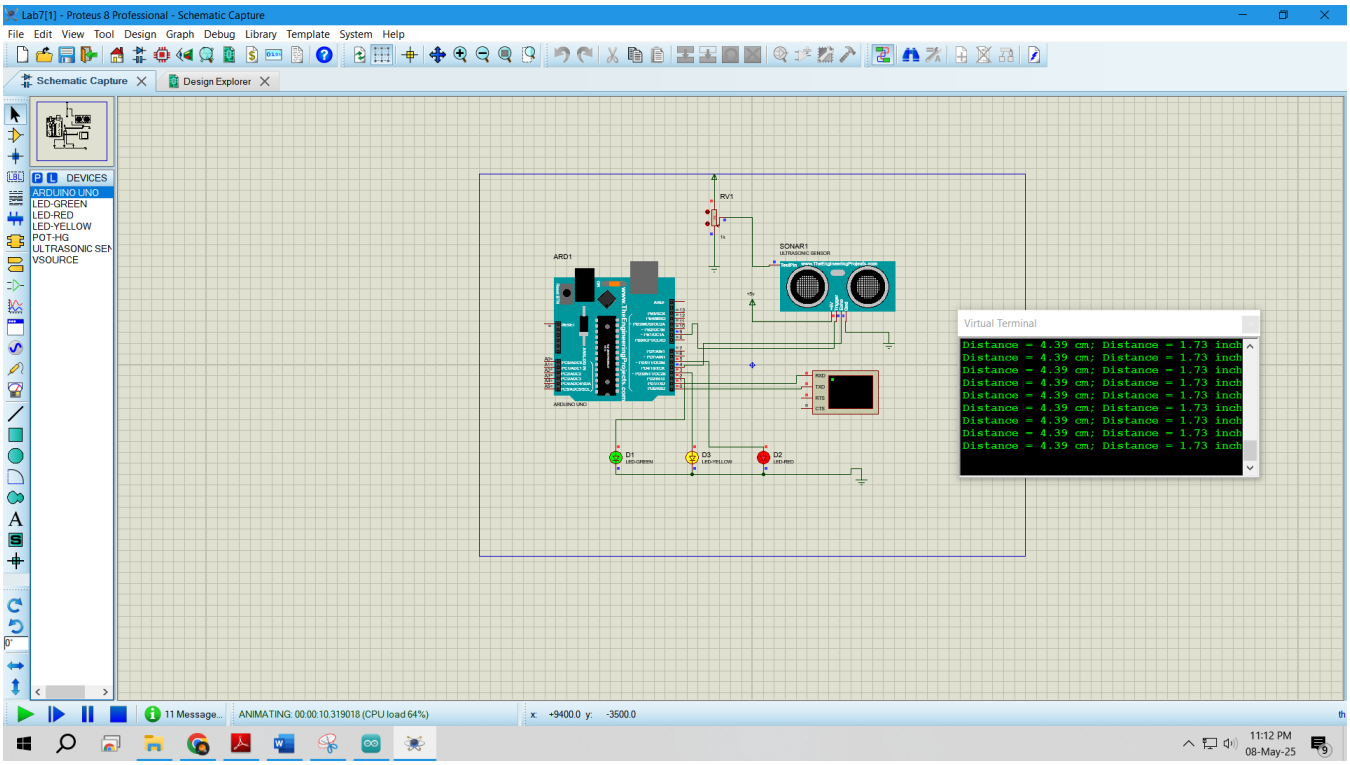


Fig: 3 LED on in shown distance(Using ID values)

Discussions: In this experiment, a sonar sensor was used with an Arduino Uno to detect the distance of an object. LEDs were connected to visually represent the distance, turning ON based on thresholds set in the code. The working principle of the sonar sensor, including Trigger and Echo pin operations, was carefully studied and implemented. The circuit was simulated using only Proteus software to verify the setup. The system accurately displayed distances through LED patterns, and any changes in detection parameters were reflected in the output. Although slight discrepancies were observed between simulation and physical results, the overall behavior matched expectations, and the objectives of the experiment were successfully met.

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 the TinkerCad https://www.instructables.com/id/Ultrasonic-Distance-Sensor-Arduino-Tinkercad/ accessed on May 3, 2019.