American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE4103: Microprocessor and /Embedded Systems Laboratory

OEL Lab Report Cover

Title: Radar System with Arduino

Experiment No:		Date of Submission:	10 February 2022
Course Title:	MICROPROCESSOR AND EMBEDDED SYSTEM		
Course Code:		Section:	N
Semester:	Spring 2021-22	Degree Program:	BSc [CSE]
Course Teacher:	Prof. Dr. Muhibul Haque Bhuyan		

Group Name/No.: **05**

No	Name	ID	DEPT
1	Zubair Ahmed	19-39745-1	CSE
2	Saidul Mursalin Khan	19-41766-3	CSE
3	Istiak Mahmud Tamim	19-41675-3	CSE
4	Tasnia Tapti	19-40005-1	CSE
5	Tonmoy Turjya Barman	19-39732-1	CSE
6	Md Abu Sufian	19-41749-3	CSE
7	Nadim Ahmed	19-41770-3	CSE
8	Ratul Bhattacharjee	19-41794-3	CSE

Faculty use only

FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Title: Radar System with Arduino

Objective: An obstacle detection mechanism was built in Tinkercad software for this experiment.

- Create a flowchart in Proteus Software to construct a simple Obstacle Detection System using an Arduino UNO.
- Create a basic Obstacle Detection System by moving the object from various distances.
- Create a basic Obstacle Detection System that uses different LEDs to represent different distances.

Theory and Methodology:

Arduino Uno:

Arduino is an open-source platform that may be used to make interactive electronics projects. Arduino is made up of a programmable microcontroller and IDE (Integrated Development Environment) software that runs on a computer and is used to write and upload computer code to the microcontroller board.

The Arduino Uno R3 is a microcontroller board that uses a detachable ATmega328 AVR microprocessor in a dual-inline-package (DIP) format. There are 20 digital input/output pins on it (of which 6 can be used as PWM outputs and 6 can be used as analog inputs). It may be programmed using the Arduino computer program, which is simple to use.

The Arduino has a large network of supporters, making it a relatively simple method to get started with embedded electronics. The Arduino Uno R3 is the most recent and third iteration.

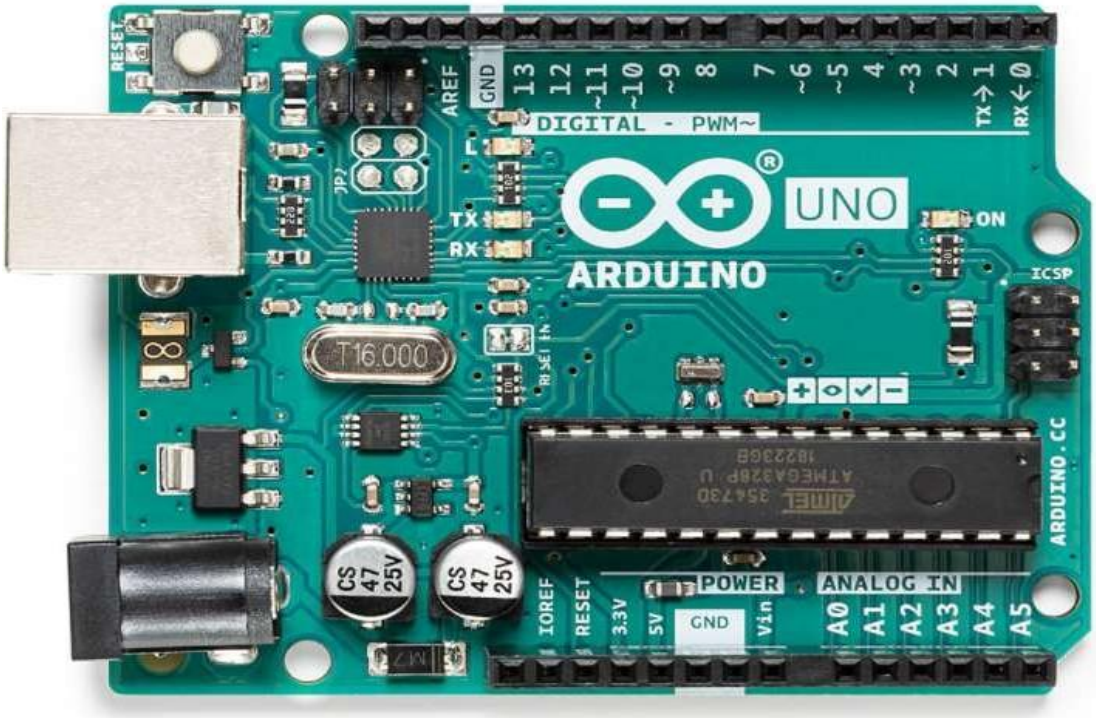


Figure 1: Arduino UNO R3, Top View.

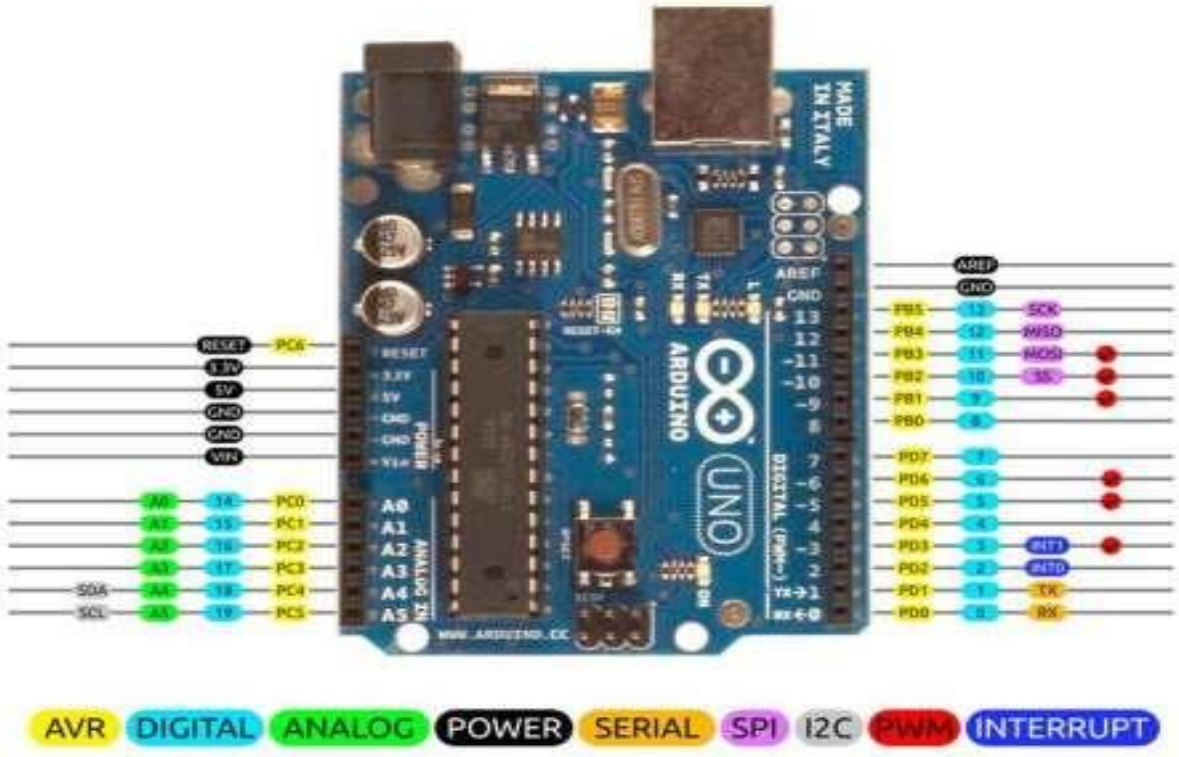


Figure 2: Arduino UNO Pin Diagram.

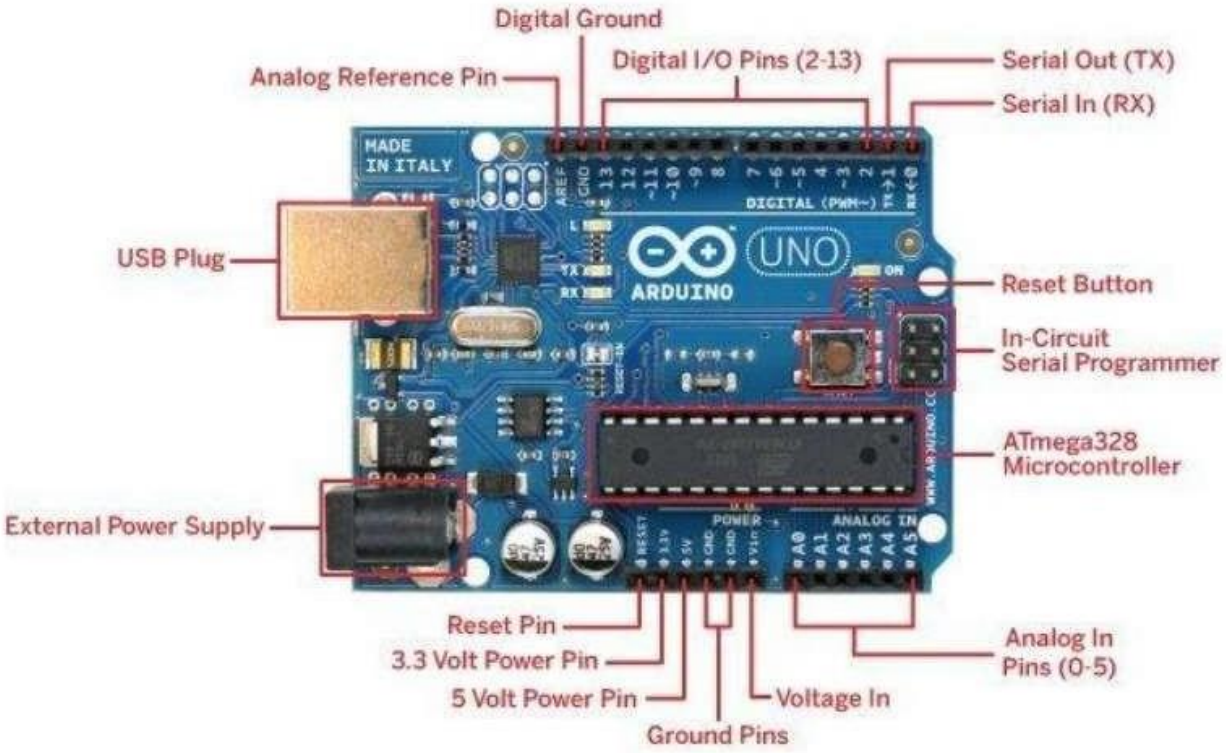


Figure 3: Overview of the Arduino UNO R3 Board.

Sonar Sensor:

The HC-SR04 Ultrasonic Distance Sensor is made up of two ultrasonic transducers at its heart. The one serves as a transmitter, converting electrical signals into ultrasonic sound pulses at a frequency of 40 KHz. The receiver listens for the pulses that have been broadcast. If it receives them, it generates an output pulse whose width can be used to calculate the pulse's travel distance. It's as easy as pie!

The sensor is compact, easy to utilize in any robotics project, and provides excellent non-contact range detection with a 3mm accuracy between 2 cm and 400 cm (about an inch to 13 feet). It may be immediately connected to an Arduino or any other 5V logic microcontroller because it runs on 5 volts.

Figure 4: HC-SR04 Sonar Sensor Pinout.

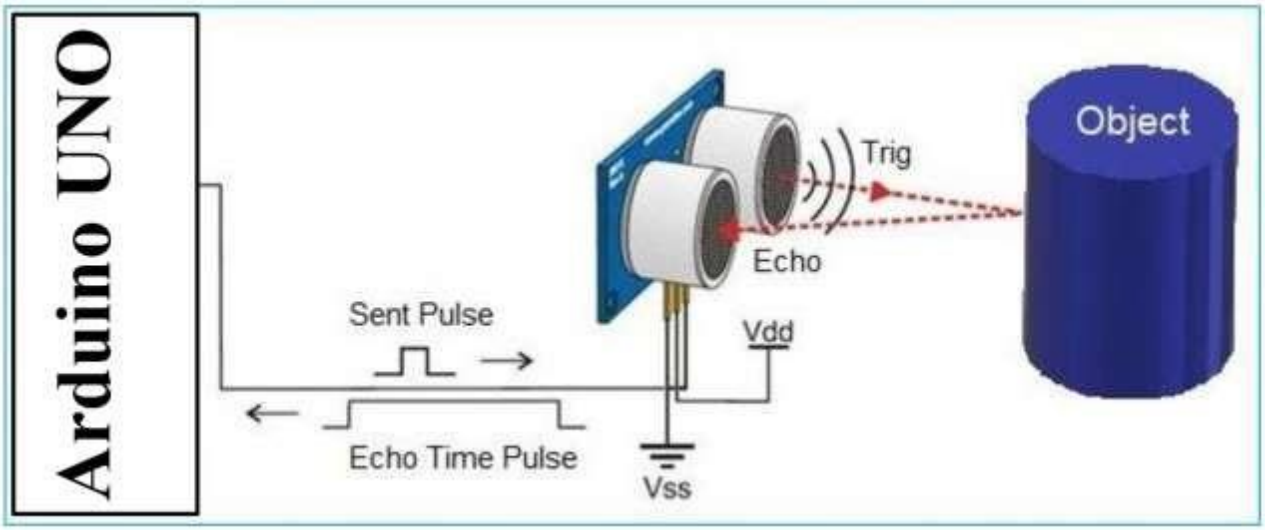


Figure 5: HC-SR04 Sonar Sensor Overview.

LED Breakout Board:

Breakout boards are a typical electrical component that separates each conductor from a bundled cable into a terminal that can readily accept a hook-up wire for distribution to another device. They're a typical component in electronic projects since they allow for quick and clean device installation.

Instead of utilizing regular LEDs, the LED Breakout Board was used for this project since it instantly connects to Arduino. Furthermore, each LED is connected in series with a resistor to guarantee that excessive voltage does not damage the LEDs.

Apparatus:

- Laptop.
- Proteus 8.11 Professional.
- Arduino Red LED Breakout Board.
- Grove Ultrasonic Ranger.
- Grove 128*64 OLED display.

Simulation Setup:

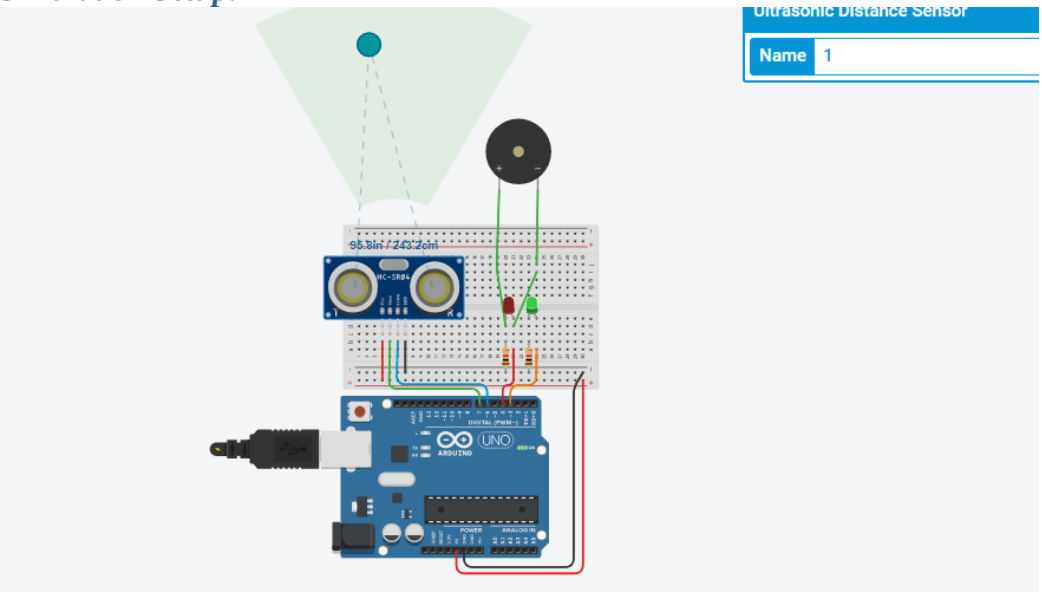
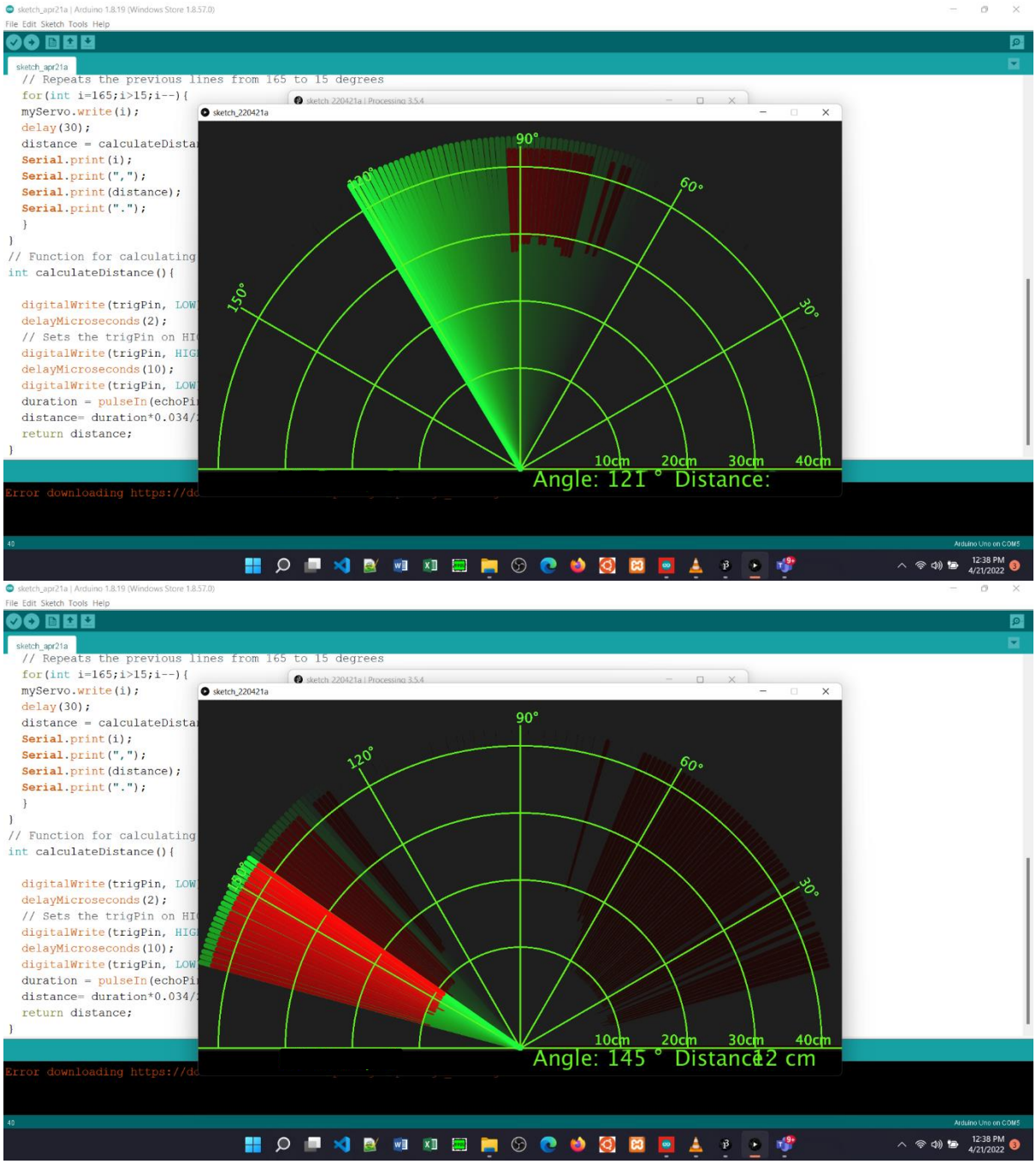
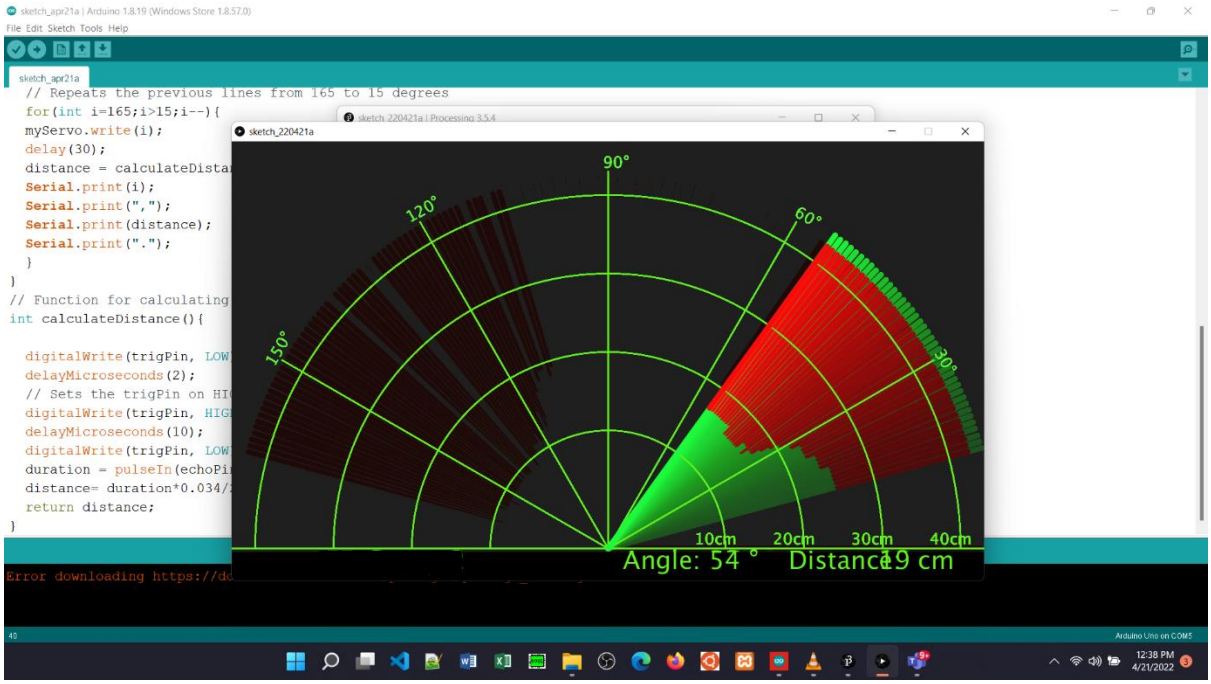
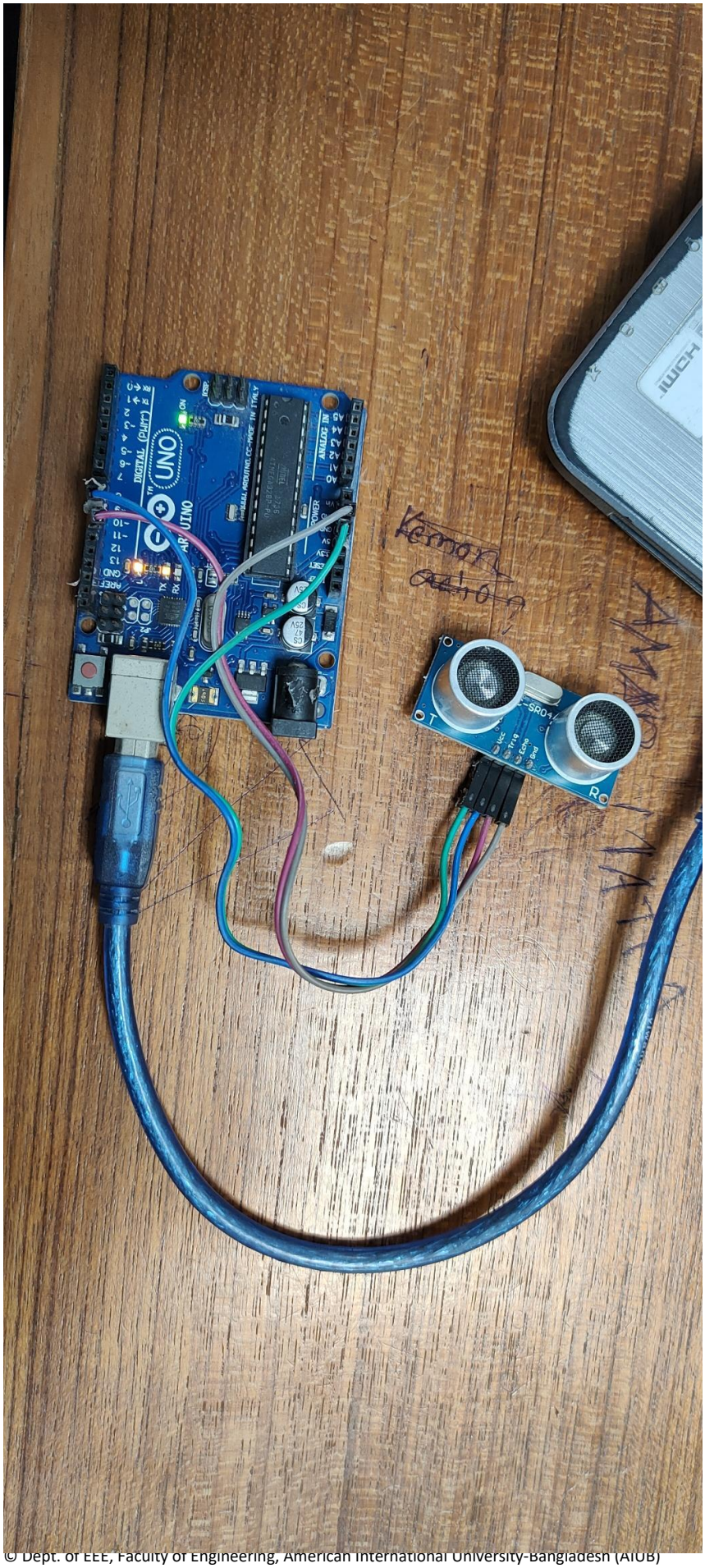


Fig: Simulation of radar system with Adruino


sketch_apr21a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_apr21a
// Repeats the previous lines from 165 to 15 degrees
for(int i=165;i>15;i--){
myServo.write(i);
delay(30);
distance = calculateDistan
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
}
}
// Function for calculating
int calculateDistance(){
digitalWrite(trigPin, LOW
delayMicroseconds(2);
// Sets the trigPin on HI
digitalWrite(trigPin, HIG
delayMicroseconds(10);
digitalWrite(trigPin, LOW
duration = pulseIn(echoPi
distance= duration*0.034/
return distance;
}
Error downloading https://dd
sketch_220421a | Processing 3.5.4
sketch_220421a
90°
120°
60°
150°
30°
10cm
20cm
30cm
40cm
Angle: 121 ° Distance:
Arduino Uno on COM5
12:38 PM
4/21/2022

sketch_apr21a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_apr21a
// Repeats the previous lines from 165 to 15 degrees
for(int i=165;i>15;i--){
myServo.write(i);
delay(30);
distance = calculateDistan
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
}
}
// Function for calculating
int calculateDistance(){
digitalWrite(trigPin, LOW
delayMicroseconds(2);
// Sets the trigPin on HI
digitalWrite(trigPin, HIG
delayMicroseconds(10);
digitalWrite(trigPin, LOW
duration = pulseIn(echoPi
distance= duration*0.034/
return distance;
}
Error downloading https://dd
sketch_220421a | Processing 3.5.4
sketch_220421a
90°
120°
60°
150°
30°
10cm
20cm
30cm
40cm
Angle: 145 ° Distance: 12 cm
Arduino Uno on COM5
12:38 PM
4/21/2022


sketch_apr21a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_apr21a
// Repeats the previous lines from 165 to 15 degrees
for(int i=165;i>15;i--){
myServo.write(i);
delay(30);
distance = calculateDista
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
}
}
// Function for calculating
int calculateDistance(){
digitalWrite(trigPin, LOW
delayMicroseconds(2);
// Sets the trigPin on HI
digitalWrite(trigPin, HIG
delayMicroseconds(10);
digitalWrite(trigPin, LOW
duration = pulseIn(echoPi
distance= duration*0.034/
return distance;
}
Error downloading https://dd
sketch_220421a | Processing 3.5.4
sketch_220421a
90°
120°
60°
150°
30°
10cm
20cm
30cm
40cm
Angle: 54 °
Distance 19 cm
Arduino Uno on COM5
12:38 PM
4/21/2022


UNO
ARDUINO
DIGITAL (PWM~)
ANALOG IN
POWER
SR04
Vcc
Trig
Echo
Gnd

Schematic diagram: Students will be adding schematic diagram from Proteus simulation in this section.

Coding Program:

```
// Includes the Servo library
#include <Servo.h>.
// Defines Tirg and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;
// Variables for the duration and the distance
long duration;
int distance;
Servo myServo; // Creates a servo object for controlling the servo motor
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}
void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for(int i=15;i<=165;i++){
  myServo.write(i);
  delay(30);
  distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree

  Serial.print(i); // Sends the current degree into the Serial Port
  Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
  Serial.print(distance); // Sends the distance value into the Serial Port
  Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
  }
  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165;i>15;i--){
  myServo.write(i);
  delay(30);
  distance = calculateDistance();
  Serial.print(i);
  Serial.print(",");
  Serial.print(distance);
  Serial.print(".");
  }
}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time
in microseconds
  distance= duration*0.034/2;
  return distance;
}
```

Discussions: Arduino-based Radar was designed in this experiment utilizing the Tinkercad program. A mental map was built before beginning the experiment in proteus to help with comprehension. The experiment was then carried out in accordance with the protocol. The Red LED was turned on in when the obstruction was 6cm away. LEDs turned on and off when the distance between them changed. As a result, all of the simulation results matched the expected outcome, and the experiment was successful.