



Jose Rizal University  
College of Computer Studies Engineering  
Computer Engineering Department

# **Lab Exam Simulation Activity**

**CPE C312 – EMBEDDED SYSTEMS**

**Submitted by:**

Exiquiel John A. Pines

**Submitted to:**

Engr. Nastaran Rezanazarzadeh

**Date Submitted:**

September 22, 2023

## Output:

```
// Lab Exam Simulation Activity
// Exiquiel John A. Pines || BSCPE-401G

// Define pins
const int redLED = 8; const int blueLED = 9; const int greenLED = 10;
const int piezoPin = 11;
const int trigPin = 3; const int echoPin = 2;

// Variables long duration; int distanceCm; int distanceIn;

void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(piezoPin, OUTPUT);
  pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT);

  // Initialize serial communication Serial.begin(9600); Serial.println("Serial Started...");
}

void loop()
{
  noTone(piezoPin);
  digitalWrite(trigPin, LOW);
  digitalWrite(trigPin, HIGH);
  digitalWrite(trigPin, LOW);

  int duration = pulseIn(echoPin, HIGH);
  distanceCm = duration*0.034/2;
  distanceIn = duration*0.0133/2;

  // Print the distance to the serial monitor

  Serial.print("Distance: ");
  Serial.print(distanceCm);
  Serial.println(" cm");

  // Red LED on, danger sound on piezo
  if (distanceCm<50) {
    digitalWrite(redLED, HIGH);
    digitalWrite(blueLED, LOW);
    digitalWrite(greenLED, LOW);

    tone(piezoPin, 2000);
  }

  // Blue LED on, hazard sound on piezo
  else if (distanceCm>=50 && distanceCm<=70) {
    digitalWrite(redLED, LOW);
    digitalWrite(blueLED, HIGH);
    digitalWrite(greenLED, LOW);

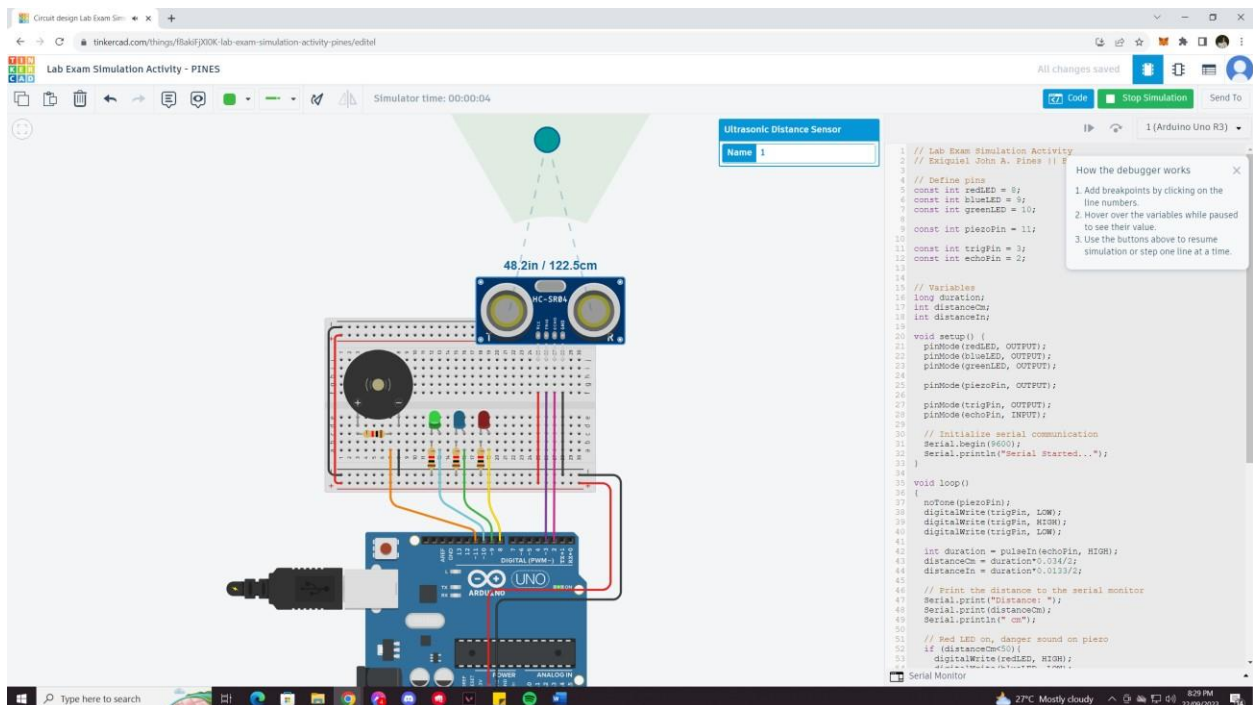
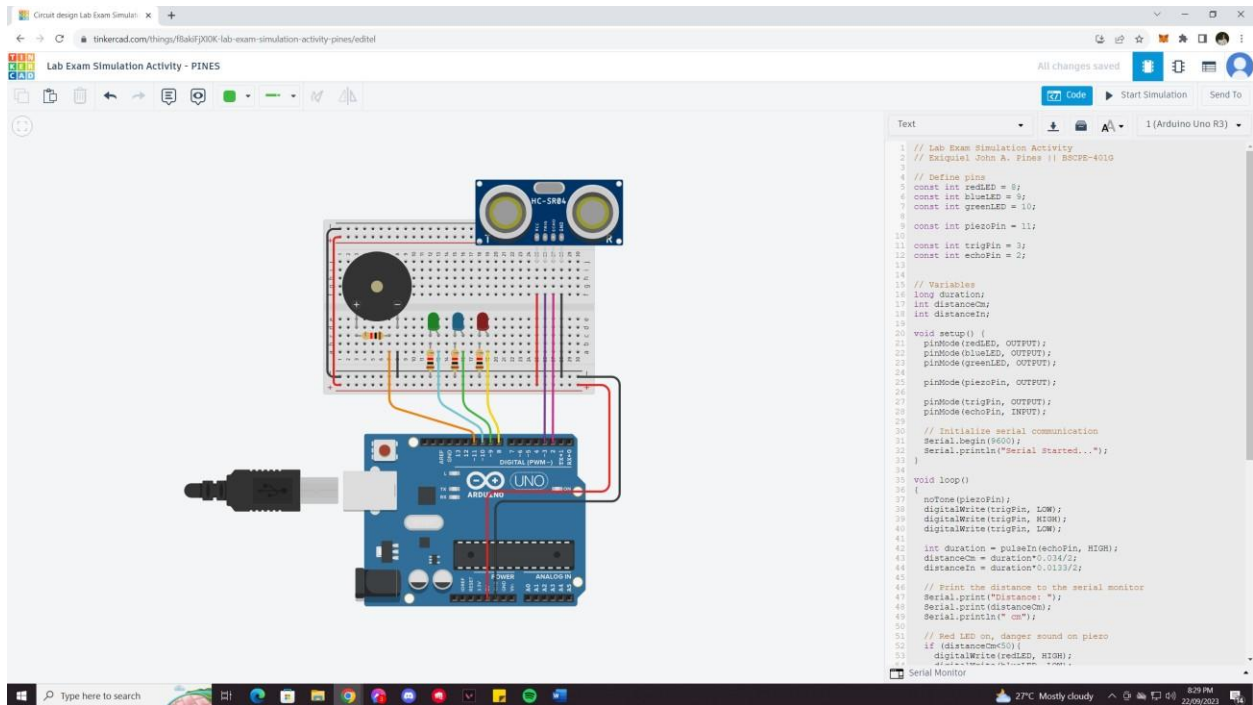
    tone(piezoPin, 1000);
  }

  // green LED on, safe sound on piezo
  if (distanceCm>=100){
    digitalWrite(redLED, LOW);
    digitalWrite(blueLED, LOW);
    digitalWrite(greenLED, HIGH);

    tone(piezoPin, 500);
  }
}
```

```
}  
  
else {  
    digitalWrite(redLED, LOW);  
    digitalWrite(blueLED, LOW);  
    digitalWrite(greenLED, LOW);  
  
    noTone(piezoPin);  
}  
  
delay(100);  
}
```

## Output:



Circuit design Lab Exam Simul

Lab Exam Simulation Activity - PINES

Simulator time: 00:00:12

1 (Arduino Uno R3)

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

34.7in / 88.0cm

HC-SR04

Arduino Uno

```
1 // Lab Exam Simulation Activity
2 // Exiquiel John A. Pines || E
3
4 // Define pins
5 const int redLED = 8;
6 const int blueLED = 9;
7 const int greenLED = 10;
8
9 const int piezoPin = 11;
10
11 const int trigPin = 3;
12 const int echoPin = 2;
13
14
15 // Variables
16 long duration;
17 int distanceCm;
18 int distanceIn;
19
20 void setup() {
21   pinMode(redLED, OUTPUT);
22   pinMode(blueLED, OUTPUT);
23   pinMode(greenLED, OUTPUT);
24   pinMode(piezoPin, OUTPUT);
25   pinMode(trigPin, OUTPUT);
26   pinMode(echoPin, INPUT);
27
28   // Initialize serial communication
29   Serial.begin(9600);
30   Serial.println("Serial Started...");
31 }
32
33 void loop() {
34   noTone(piezoPin);
35   digitalWrite(trigPin, LOW);
36   digitalWrite(trigPin, HIGH);
37   digitalWrite(trigPin, LOW);
38   int duration = pulseIn(echoPin, HIGH);
39   distanceCm = duration*0.034/2;
40   distanceIn = duration*0.0133/2;
41
42   // Print the distance to the serial monitor
43   Serial.print("Distance: ");
44   Serial.print(distanceCm);
45   Serial.println(" cm");
46
47   // Red LED on, danger sound on piezo
48   if (distanceCm<50){
49     digitalWrite(redLED, HIGH);
50     tone(piezoPin, 1000);
51   }
52 }
```

Circuit design Lab Exam Sim

Lab Exam Simulation Activity - PINES

Simulator time: 00:00:19

1 (Arduino Uno R3)

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

22.6in / 57.4cm

HC-SR04

Arduino Uno

```
1 // Lab Exam Simulation Activity
2 // Exiquiel John A. Pines || E
3
4 // Define pins
5 const int redLED = 8;
6 const int blueLED = 9;
7 const int greenLED = 10;
8
9 const int piezoPin = 11;
10
11 const int trigPin = 3;
12 const int echoPin = 2;
13
14
15 // Variables
16 long duration;
17 int distanceCm;
18 int distanceIn;
19
20 void setup() {
21   pinMode(redLED, OUTPUT);
22   pinMode(blueLED, OUTPUT);
23   pinMode(greenLED, OUTPUT);
24   pinMode(piezoPin, OUTPUT);
25   pinMode(trigPin, OUTPUT);
26   pinMode(echoPin, INPUT);
27
28   // Initialize serial communication
29   Serial.begin(9600);
30   Serial.println("Serial Started...");
31 }
32
33 void loop() {
34   noTone(piezoPin);
35   digitalWrite(trigPin, LOW);
36   digitalWrite(trigPin, HIGH);
37   digitalWrite(trigPin, LOW);
38   int duration = pulseIn(echoPin, HIGH);
39   distanceCm = duration*0.034/2;
40   distanceIn = duration*0.0133/2;
41
42   // Print the distance to the serial monitor
43   Serial.print("Distance: ");
44   Serial.print(distanceCm);
45   Serial.println(" cm");
46
47   // Red LED on, danger sound on piezo
48   if (distanceCm<50){
49     digitalWrite(redLED, HIGH);
50     tone(piezoPin, 1000);
51   }
52 }
```

Circuit design Lab Exam Sim

tinkercad.com/things/f8akJX00K-lab-exam-simulation-activity-pines/ed1ef

Lab Exam Simulation Activity - PINES

Simulator time: 00:00:23

1 (Arduino Uno R3)

Ultrasonic Distance Sensor

Name 1

13.8in / 35.0cm

HC-SR04

Arduino Uno

1 // Lab Exam Simulation Activity  
2 // Eniquiel John A. Pines ||  
3  
4 // Define pins  
5 const int redLED = 8;  
6 const int blueLED = 9;  
7 const int greenLED = 10;  
8 const int piezoPin = 11;  
9  
10 const int trigPin = 3;  
11 const int echoPin = 2;  
12  
13  
14  
15 // Variables  
16 long duration;  
17 int distanceCm;  
18 int distanceIn;  
19  
20 void setup() {  
21 pinMode(redLED, OUTPUT);  
22 pinMode(blueLED, OUTPUT);  
23 pinMode(greenLED, OUTPUT);  
24  
25 pinMode(piezoPin, OUTPUT);  
26  
27 pinMode(trigPin, OUTPUT);  
28 pinMode(echoPin, INPUT);  
29  
30 // Initialize serial communication  
31 Serial.begin(9600);  
32 Serial.println("Serial Started...");  
33 }  
34  
35 void loop() {  
36 {  
37 noTone(piezoPin);  
38 digitalWrite(trigPin, LOW);  
39 digitalWrite(trigPin, HIGH);  
40 digitalWrite(trigPin, LOW);  
41  
42 int duration = pulseIn(echoPin, HIGH);  
43 distanceCm = duration\*0.034/2;  
44 distanceIn = duration\*0.013/2;  
45  
46 // Print the distance to the serial monitor  
47 Serial.print("Distance: ");  
48 Serial.print(distanceCm);  
49 Serial.println(" cm");  
50  
51 // Red LED on, danger sound on piezo  
52 if (distanceCm<50) {  
53 digitalWrite(redLED, HIGH);  
54 tone(piezoPin, 1000, 100);  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }

How the debugger works

1. Add breakpoints by clicking on the line numbers.
2. Hover over the variables while paused to see their value.
3. Use the buttons above to resume simulation or step one line at a time.

Serial Monitor

27°C Mostly cloudy 8:29 PM 22/09/2023