

# **CEIS 101 Final Course Project Presentation**

**CEIS101 Final Course Project**

**March 2025**

**Robert Pitter**

**Prof. Genevieve Sapijaszko**

# Introduction

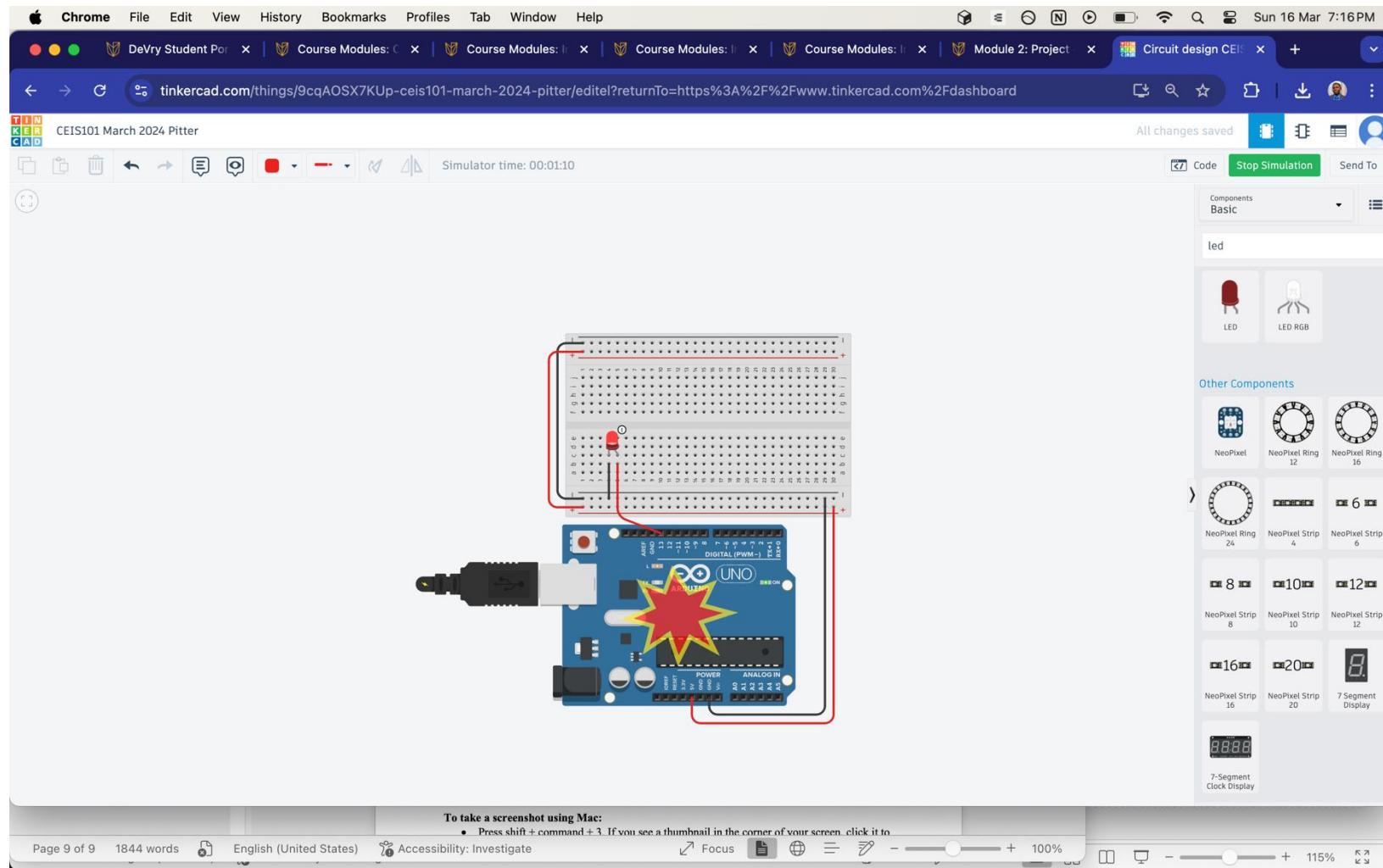
- This project is a home automation and security system circuit that includes distance sensors, door sensors and light sensors.
- The circuit was designed using TinkerCad and specifically, the tools that are involved in the final design of the circuit are:
  - Arduino Uno R3
  - Wires (19 Total)
  - LEDs (4 Total)
  - Piezo
  - Resistor (10 kΩ)
  - PhotoResistor
  - UltraSonic Distance Sensor (4-pin)

# CEIS101 Module 2

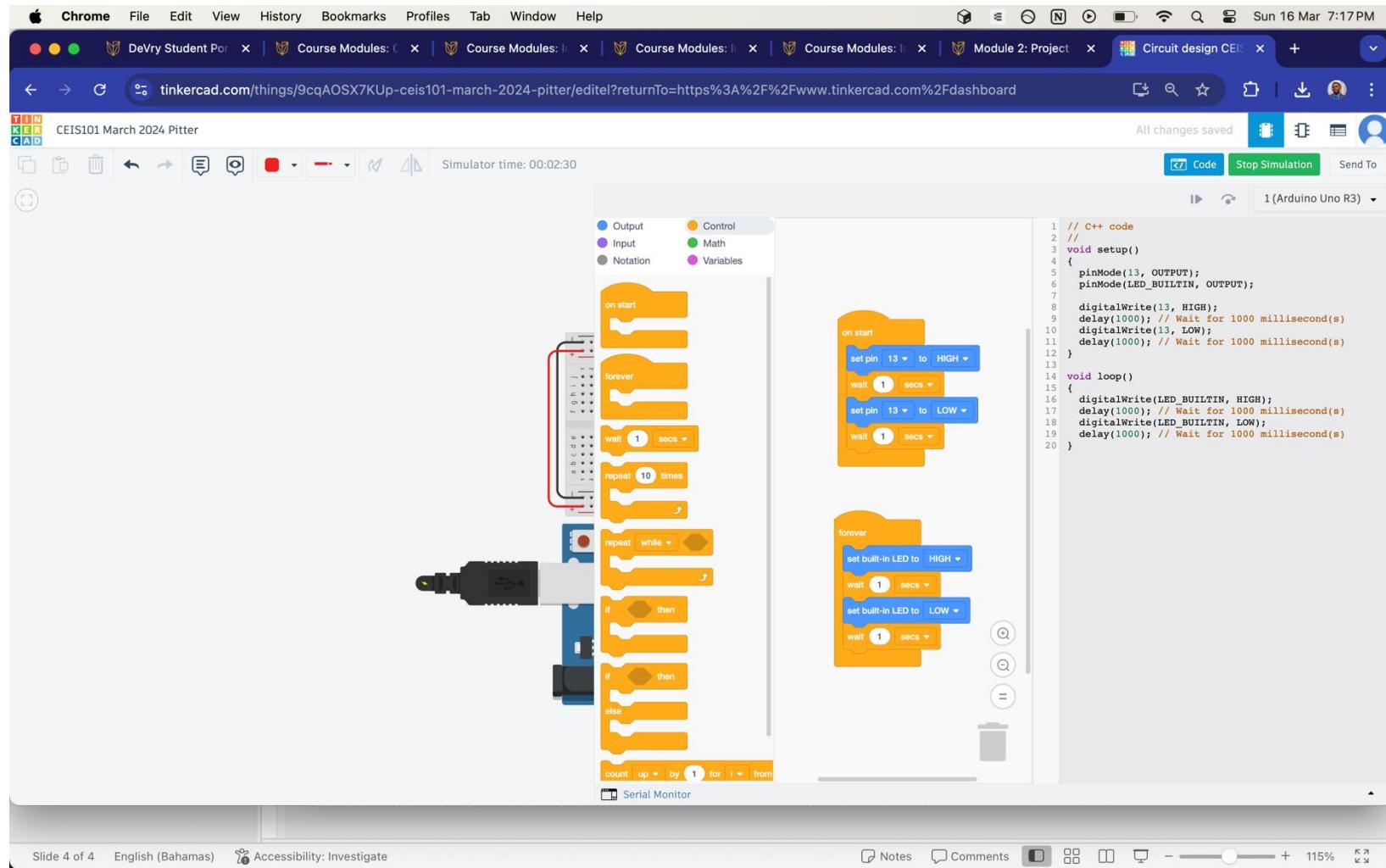
## Circuit Simulation

The aim for this project was to create a simple circuit with a blinking LED.

# Circuit (screenshot)



# Code (screenshot)



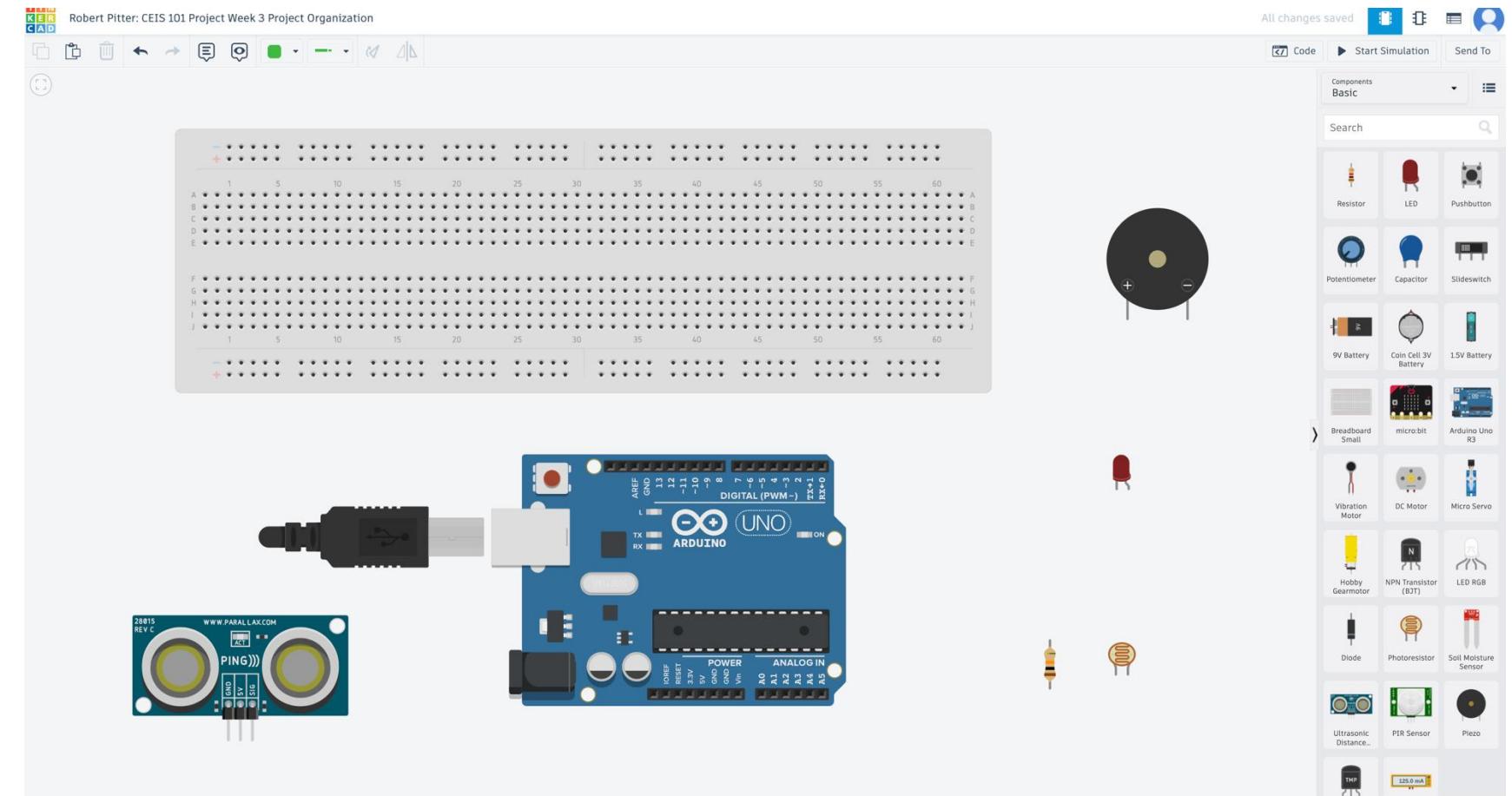
# CEIS101 Module 3

## **Adding Door Sensor to Smart Home System**

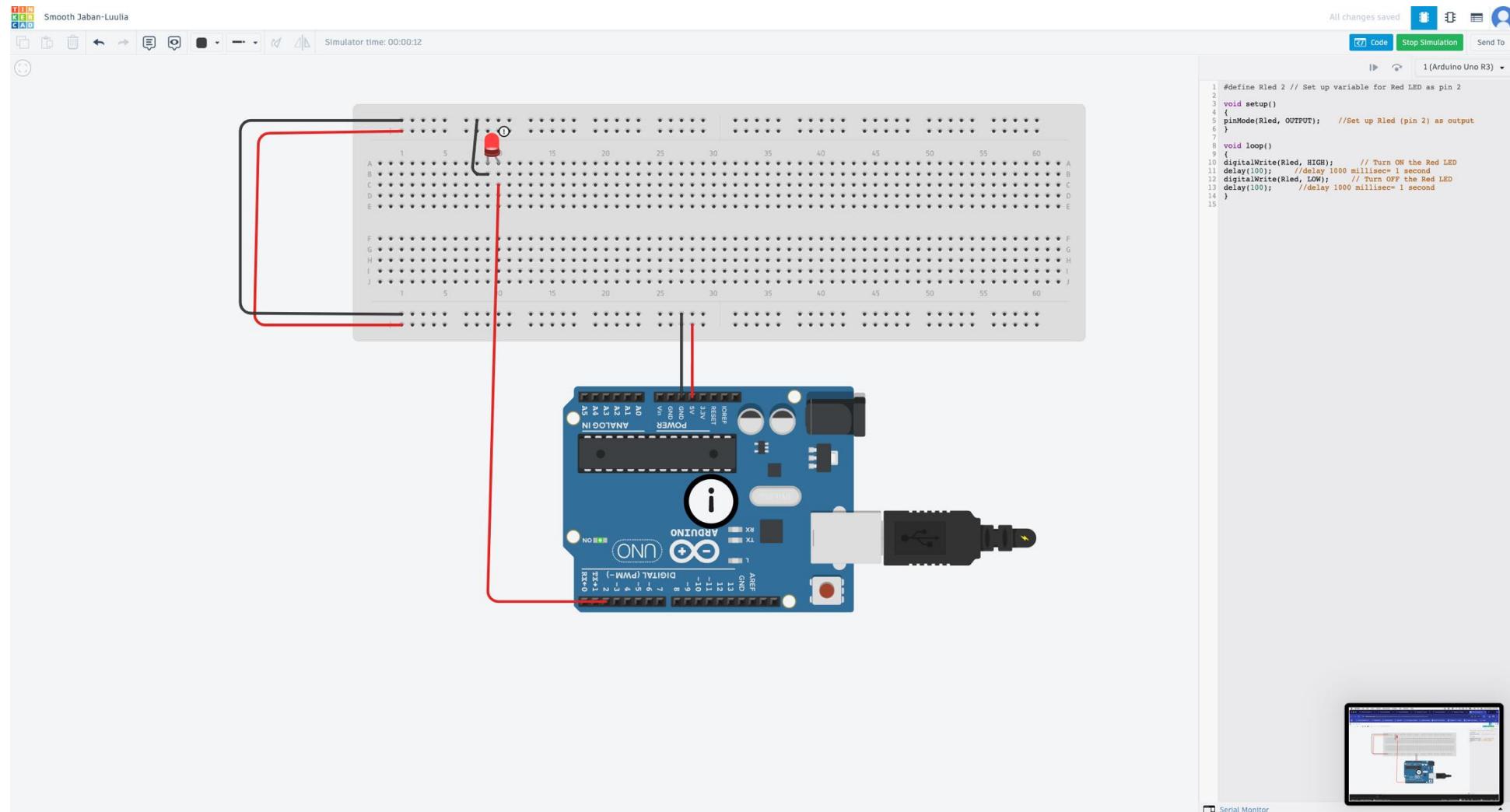
The aim for this project was to showcase all the components used in the project

# Organization of Project Components (picture)

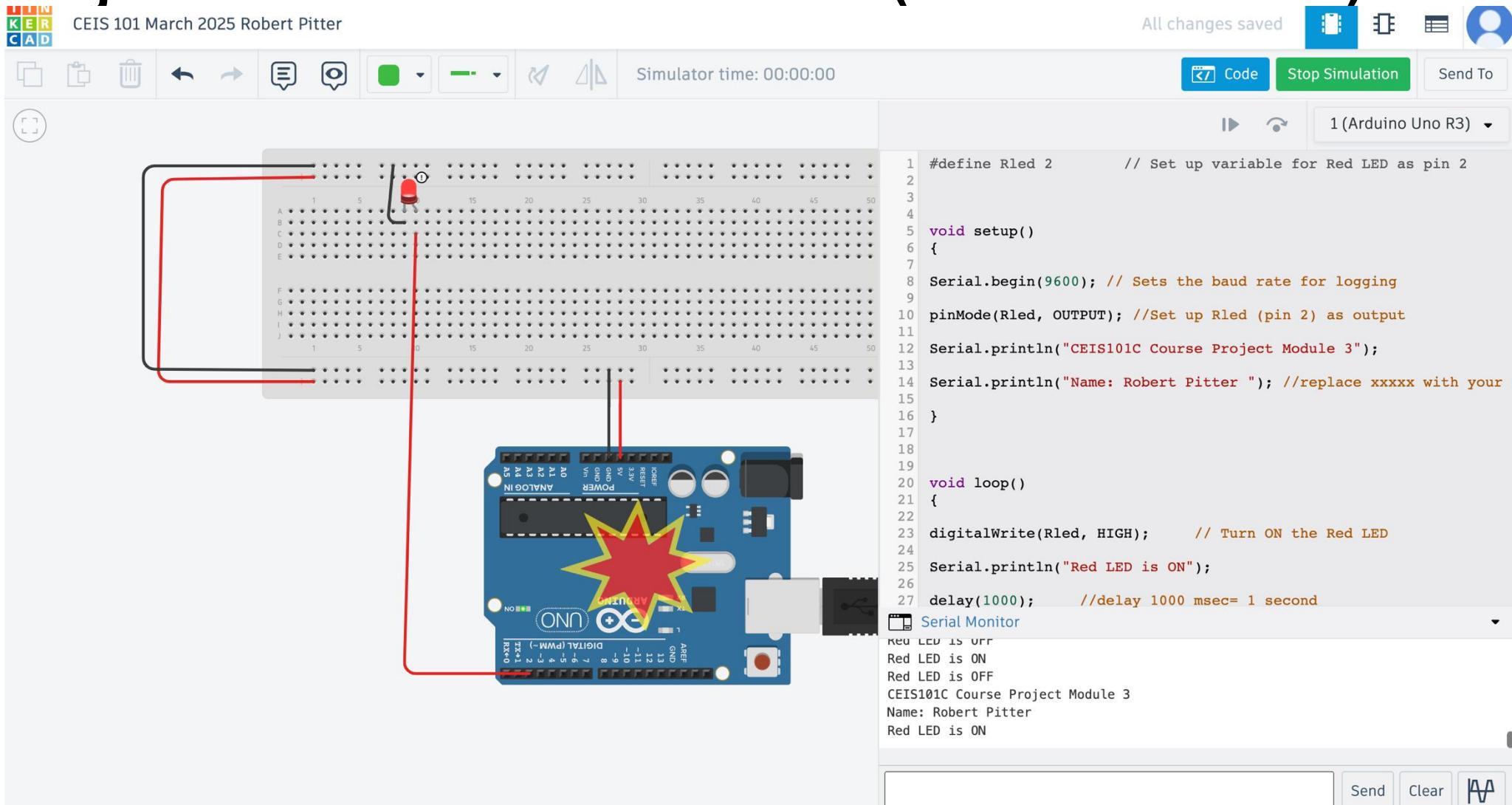
- Arduino Uno
- Breadboard
- Resistor 10kΩ
- LEDs
- Ultrasonic Sensor
- Piezo
- Photoresistor



# Project 3 Circuit with red LED on (picture)



# Project 3 Serial Monitor (screenshot)



# CEIS101 Module 4

## **Adding Door Sensor to Smart Home System**

The aim for this project was to build and simulate a door sensor in a Smart Home System

# Circuit of door closed with Green LED ON (picture)

The screenshot shows a breadboard simulation interface with the following components and connections:

- Slideswitch:** A digital input component connected to pin 9 of the Arduino Uno.
- Green LED:** Connected to pin 3 of the Arduino Uno.
- Buzzer:** Connected to pin 10 of the Arduino Uno.
- Power:** A 9V battery symbol connected to the Arduino Uno's power pins.
- Arduino Uno:** The central microcontroller board.

The breadboard grid is labeled with letters A through J and numbers 1 through 55. The Arduino Uno has its pins labeled: GND, 5V, AREF, 3.3V, 2.3V, 1.1V, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55.

**Code:**

```
1 #define Rled 2
2 #define Yled 3
3 #define Gled 4
4 #define buzzer 10
5 #define door 9
6 #define delaytime 100 // === Second run, change to 100
7
8 void setup() {
9     Serial.begin(9600); // Set the baud rate
10    Serial.println("CEIS101C Course Project Module 4");
11    Serial.println("Name: Robert Pitter "); //replace xxxxx with your name
12
13    pinMode(Rled, OUTPUT);
14    pinMode(Yled, OUTPUT);
15}
```

**Serial Monitor:**

```
CEIS101C Course Project Module 4
Name: Robert Pitter
Door is open. Security Alert!
Door is open. Security Alert!
CEIS101C Course Project Module 4
Name: Robert Pitter
```

# Circuit of door open with Green LED OFF (picture)

TINKERCAD Copy of CEIS 101 March 2025 Robert Pitter All changes saved

Simulator time: 00:00:01 Code Stop Simulation Send To

Slideswitch Name 1

```
#define Rled 2
#define Yled 3
#define Gled 4
#define buzzer 10
#define door 9
#define delaytime 1000 // === Second run, change to 100

void setup() {
  Serial.begin(9600); // Set the baud rate
  Serial.println("CEIS101C Course Project Module 4");
  Serial.println("Name: Robert Pitter "); //replace xxxxx with your name

  pinMode(Rled, OUTPUT);
  pinMode(Yled, OUTPUT);
  pinMode(Gled, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(door, INPUT);
}
```

Serial Monitor

Door is open. Security Alert!  
Door is open. Security Alert!  
Door is open. Security Alert!  
CEIS101C Course Project Module 4  
Name: Robert Pitter  
Door is open. Security Alert!

Send Clear

# Arduino Code (screenshot)

The screenshot shows the Arduino IDE interface. The top bar indicates "All changes saved". Below the menu bar, there are buttons for "Code", "Start Simulation", and "Send To". The central area is a text editor with the following code:

```
1 #define Rled 2
2 #define Yled 3
3 #define Gled 4
4 #define buzzer 10
5 #define door 9
6 #define delaytime 1000 // === Second run, change to 100
7
8 void setup() {
9   Serial.begin(9600); // Set the baud rate
10  Serial.println("CEIS101C Course Project Module 4");
11  Serial.println("Name: Robert Fitter "); //replace xxxxx with your
12
13  pinMode(Rled, OUTPUT);
14  pinMode(Yled, OUTPUT);
15  pinMode(Gled, OUTPUT);
16  pinMode(buzzer, OUTPUT);
17  digitalWrite(buzzer, LOW);
18  pinMode(door, INPUT_PULLUP); //door sensor
19 }
20
21 void loop() {
22   int value=digitalRead(door);
23   if(value == 0) { //Door closed, no security threat
24     digitalWrite(Rled, LOW);
25     digitalWrite(Yled, LOW);
26     digitalWrite(Gled, HIGH);
27     noTone(buzzer);
28   }
29   else{ //Door open, security threat
30     Serial.println("Door is open. Security Alert! ");
31     digitalWrite(Rled, HIGH);
32     digitalWrite(Yled, HIGH);
33     tone(buzzer, 500);
34     digitalWrite(Gled, LOW);
35     delay(delaytime);
36     digitalWrite(Rled, LOW);
37     digitalWrite(Yled, LOW);
38     digitalWrite(buzzer, LOW);
39     noTone(buzzer);
40     delay(delaytime);
41   } //end of else
42 } //end of loop
43
```

The bottom part of the IDE shows the "Serial Monitor" window, which displays the repeated message "Door is open. Security Alert!"

# Serial Monitor (screenshot)

Serial Monitor

CEIS101C Course Project Module 4  
Name: Robert Pitter  
Door is open. Security Alert!

CEIS101C Course Project Module 4  
Name: Robert Pitter  
Door is open. Security Alert!

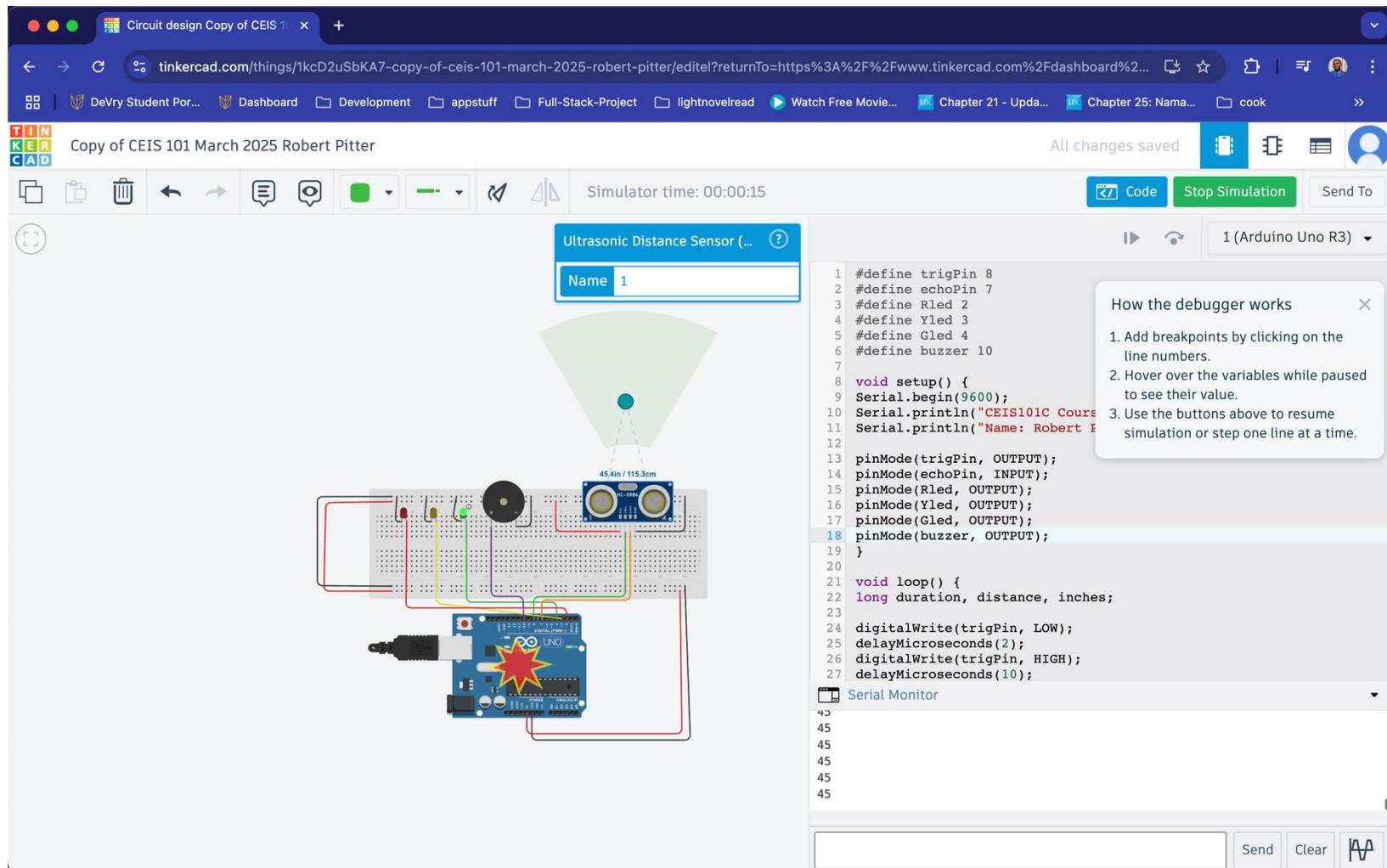
Send Clear

# CEIS101 Module 5

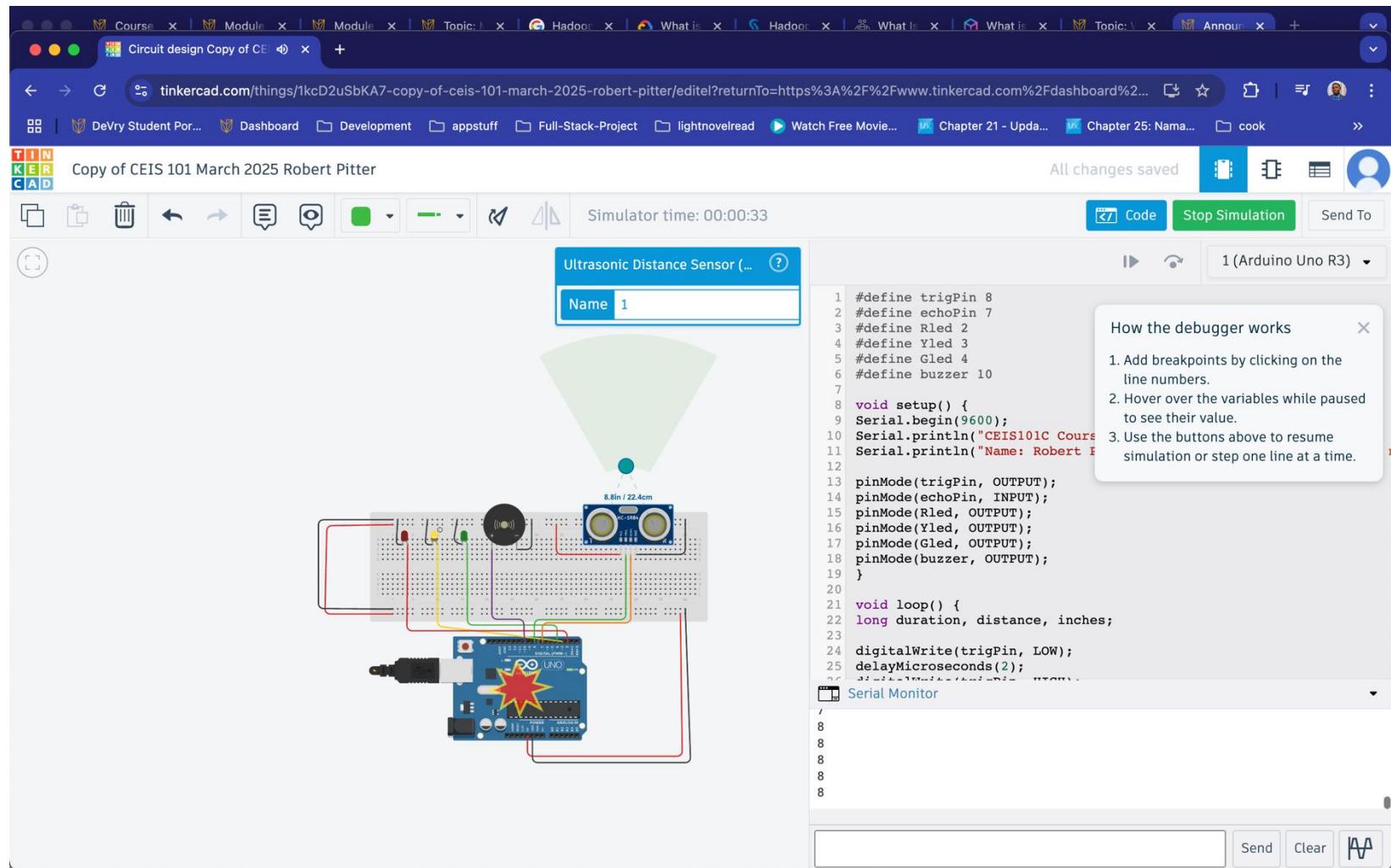
## **Adding Distance Sensor to Smart Home System and Conducting Data Analysis**

The goal of this project was to not only add the distance sensor but also record the data from the readings and plot the results in a graph. An additional chart was added which shows the distribution of results.

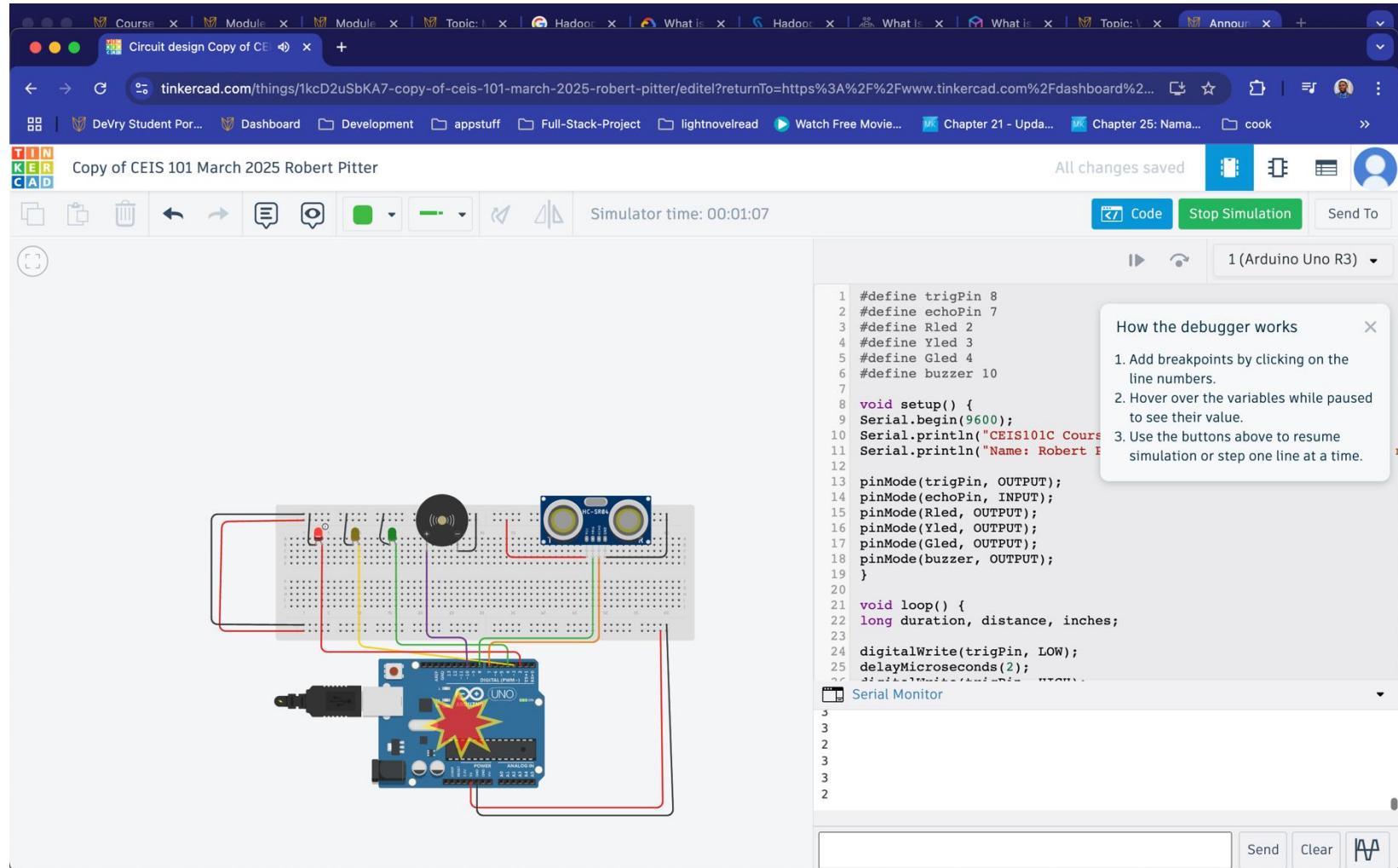
# Circuit with green LED on (picture)



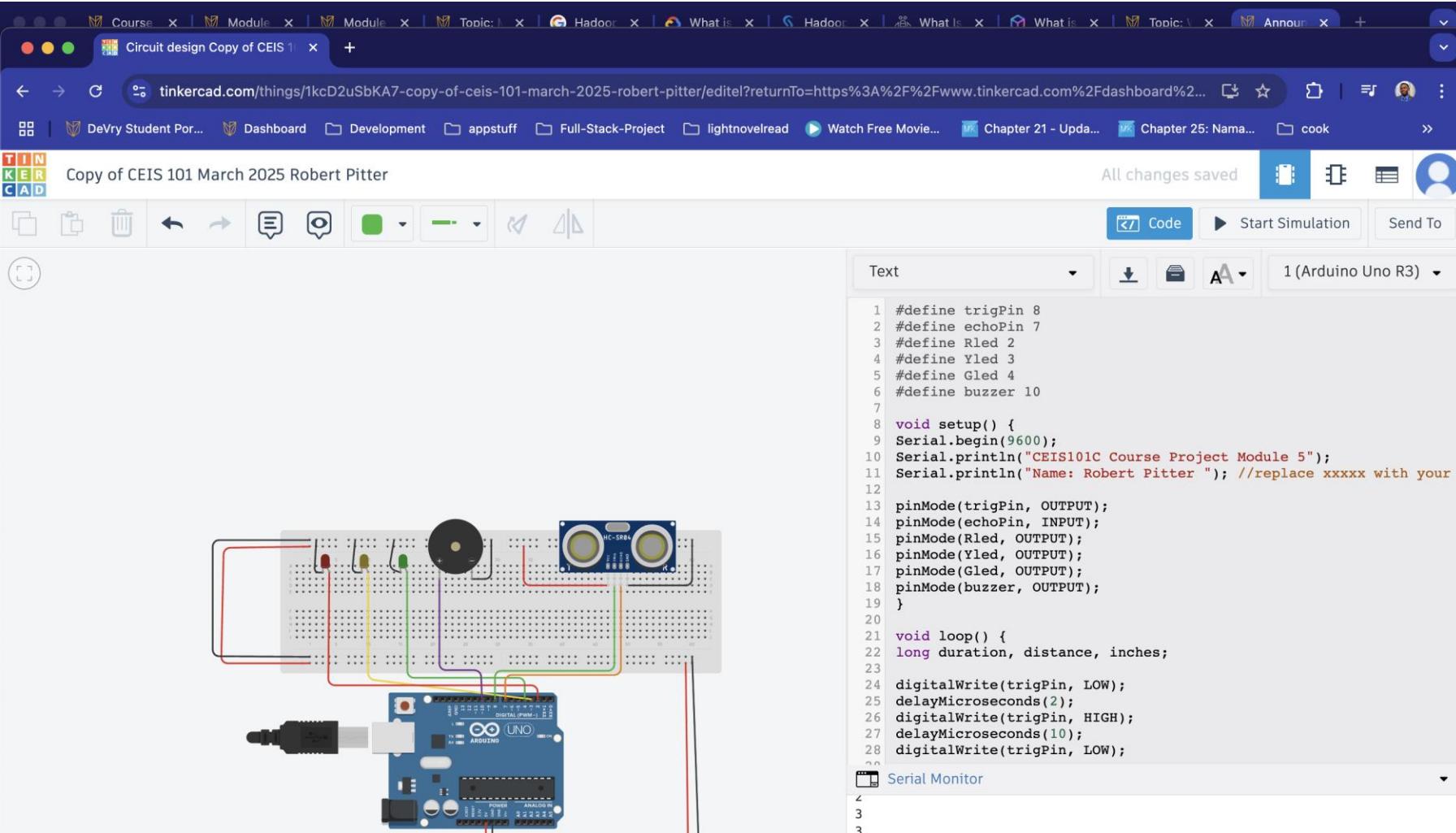
# Circuit with yellow LED on (picture)



# Circuit with red LED on (picture)



# Arduino Code (screenshot)



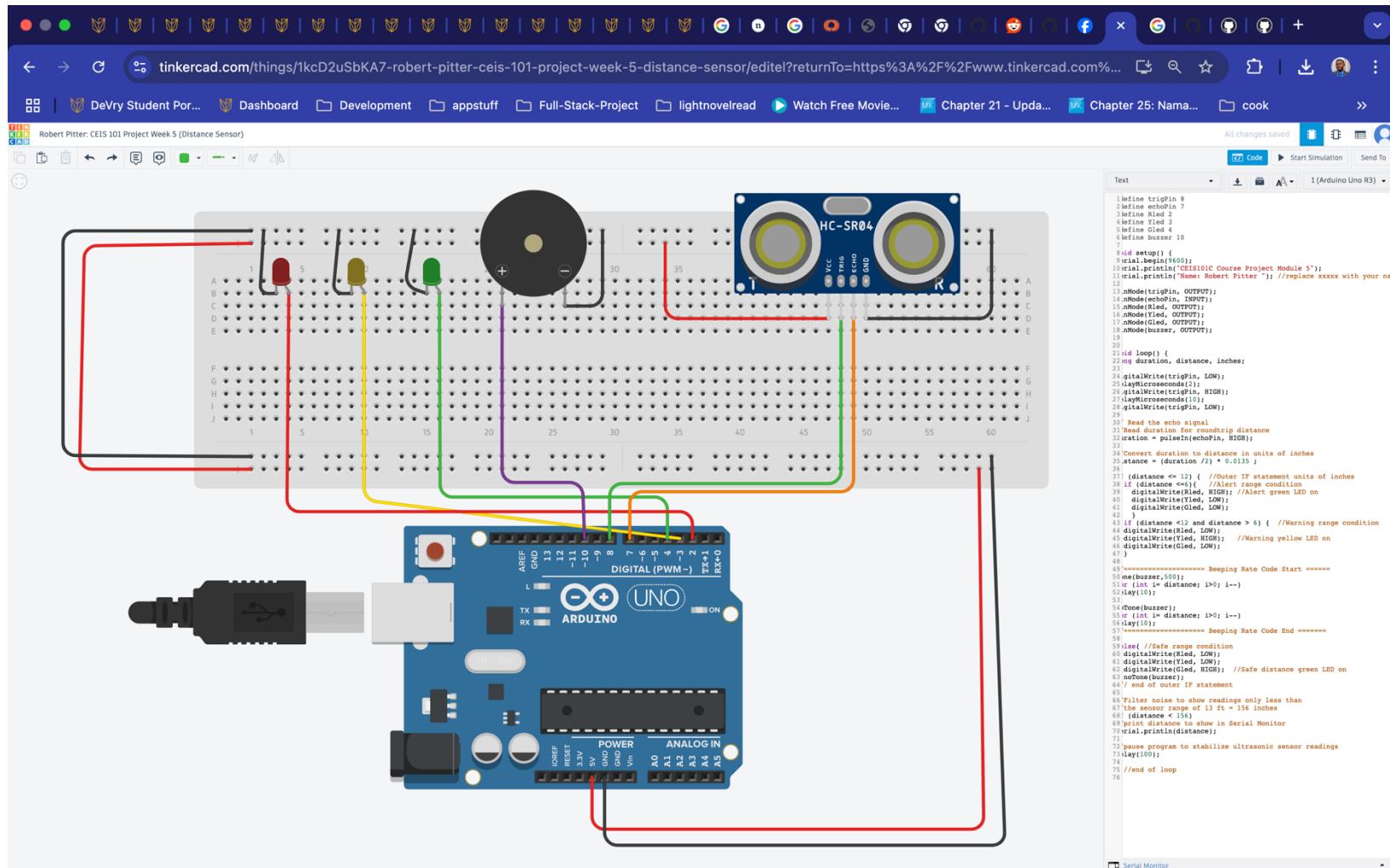
The screenshot shows a Tinkercad workspace titled "Copy of CEIS 101 March 2025 Robert Pitter". The workspace includes a circuit diagram and the corresponding Arduino code.

**Circuit Diagram:** An Arduino Uno R3 is connected to a breadboard. A red wire connects pin 8 (trigPin) to the breadboard. A green wire connects pin 7 (echoPin) to the breadboard. A blue wire connects the breadboard to an HC-SR04 ultrasonic sensor. The breadboard also has a yellow wire connected to ground and a red wire connected to power. A black cable is connected from the breadboard to a computer.

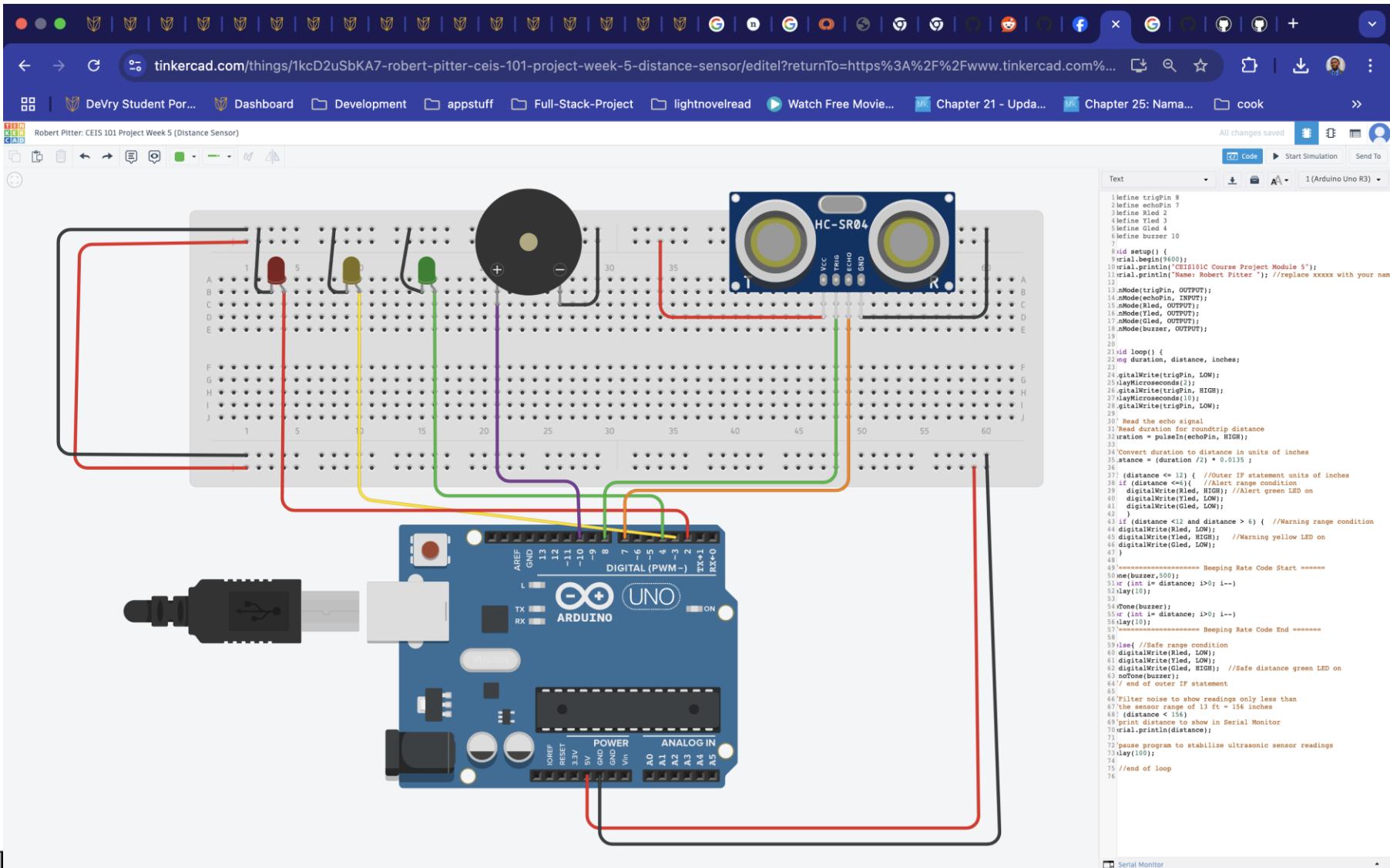
**Code:**

```
1 #define trigPin 8
2 #define echoPin 7
3 #define Rled 2
4 #define Yled 3
5 #define Gled 4
6 #define buzzer 10
7
8 void setup() {
9   Serial.begin(9600);
10  Serial.println("CEIS101C Course Project Module 5");
11  Serial.println("Name: Robert Pitter "); //replace xxxx with your name
12
13 pinMode(trigPin, OUTPUT);
14 pinMode(echoPin, INPUT);
15 pinMode(Rled, OUTPUT);
16 pinMode(Yled, OUTPUT);
17 pinMode(Gled, OUTPUT);
18 pinMode(buzzer, OUTPUT);
19 }
20
21 void loop() {
22   long duration, distance, inches;
23
24   digitalWrite(trigPin, LOW);
25   delayMicroseconds(2);
26   digitalWrite(trigPin, HIGH);
27   delayMicroseconds(10);
28   digitalWrite(trigPin, LOW);
29 }
```

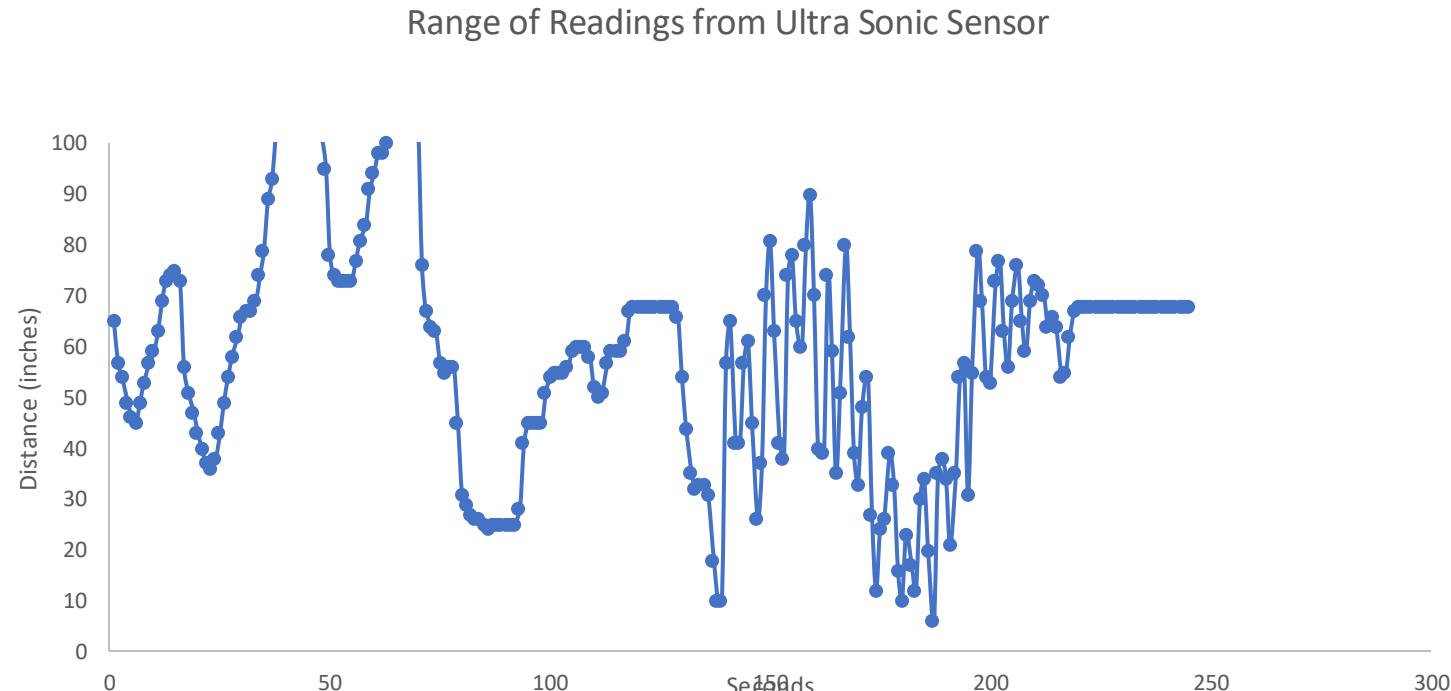
**Serial Monitor:** The serial monitor shows the output of the code: "3" and "3".



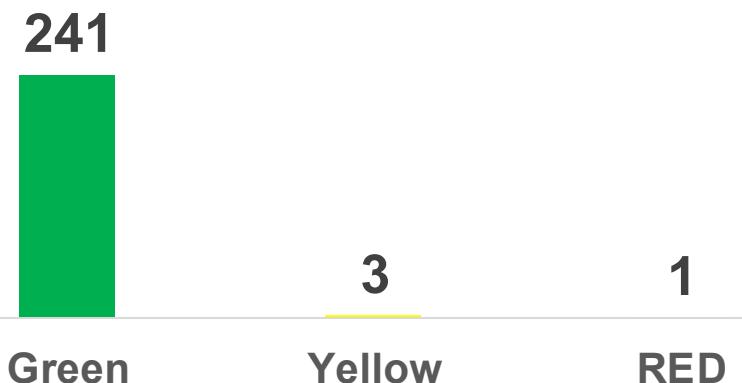
# Arduino Code (screenshot)



# Plot of data (graph from Excel)



Distribution of Distances Recorded with  
UltraSonic Sensor

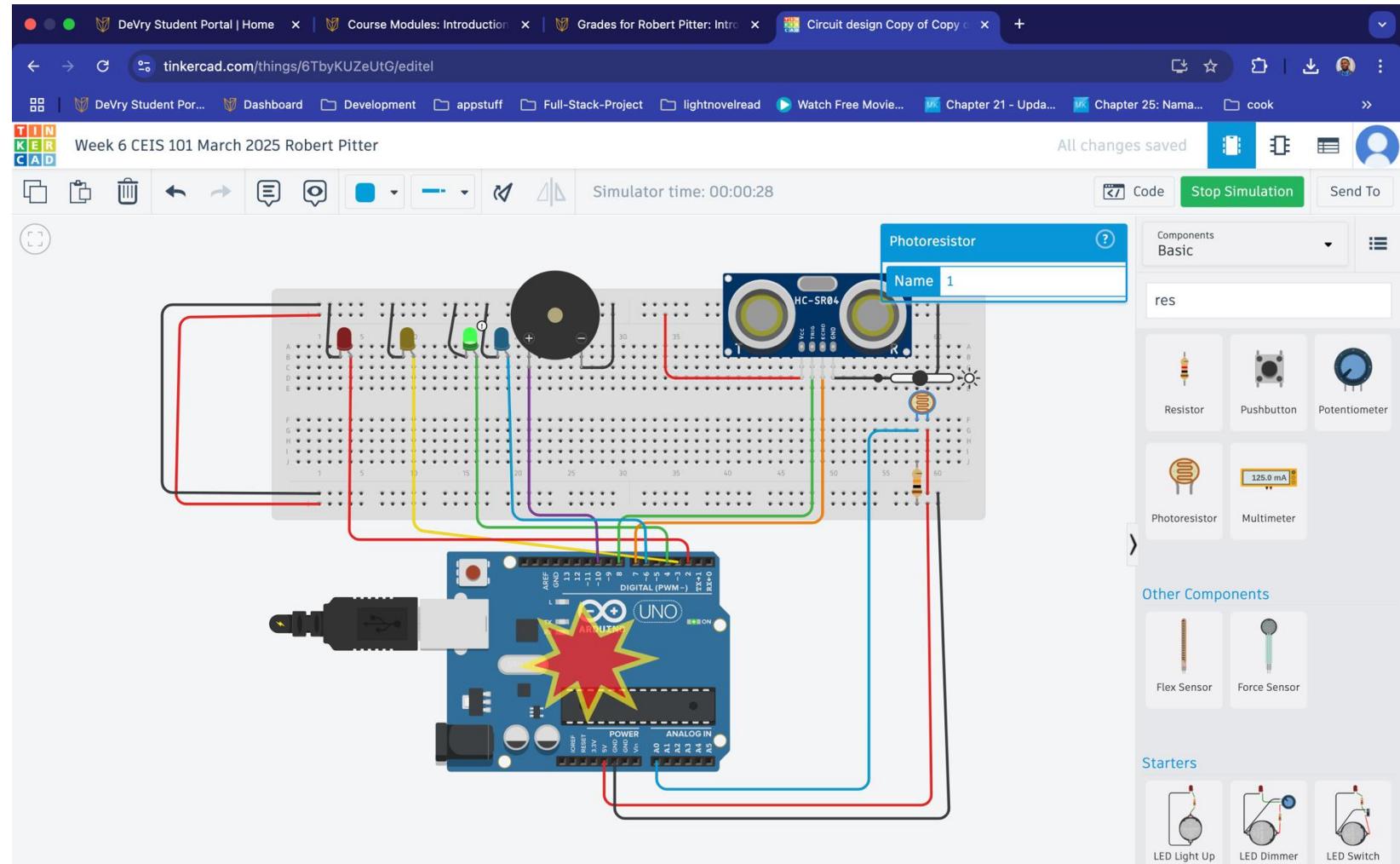


# CEIS101 Module 6

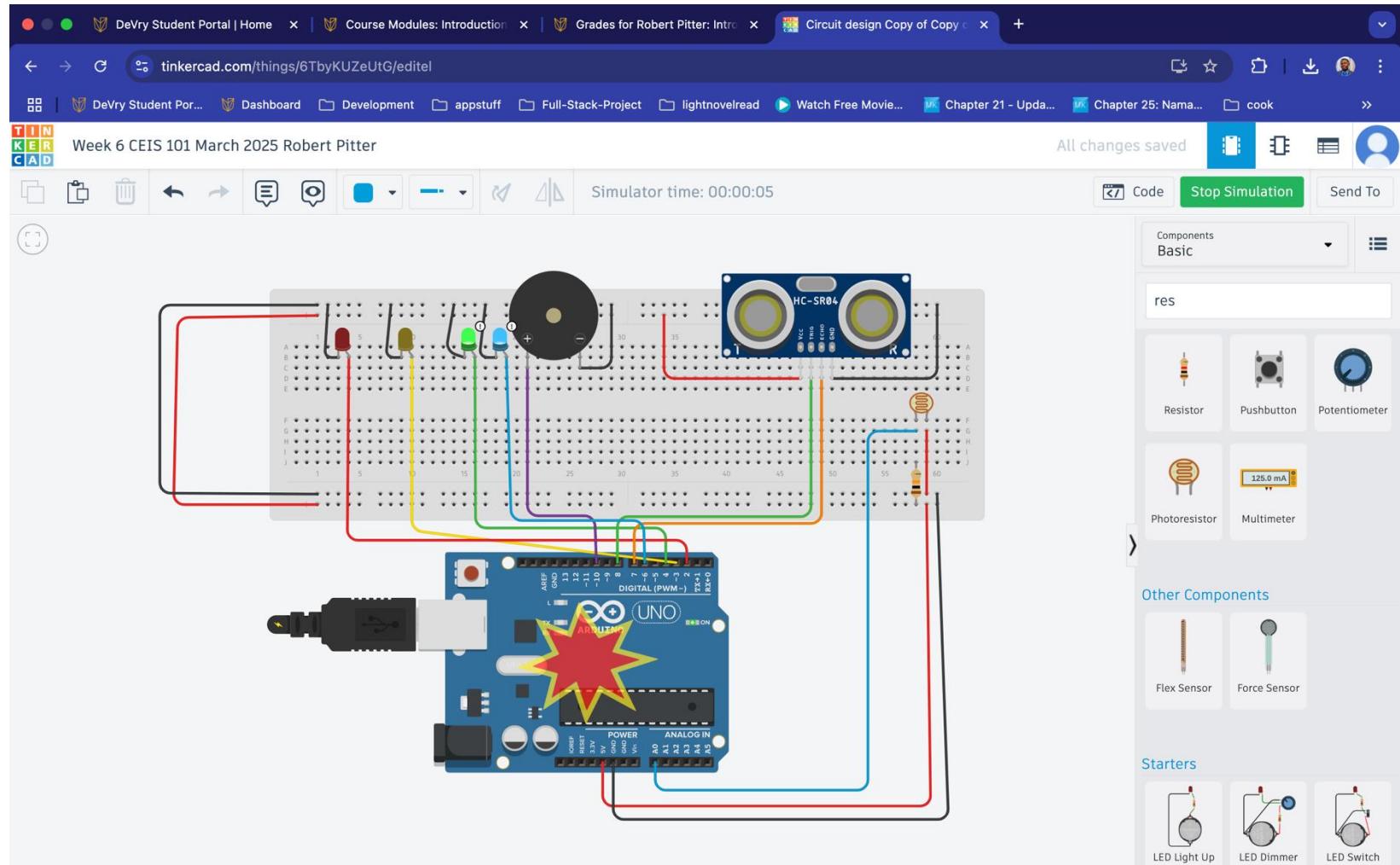
## **Adding Automated Light to Smart Home System**

The goal of the project was to add a light sensor as an add-on to the circuit built in module 5. With this, the home automation system can now also detect if there is light (indicating daytime) or not. Combining those results along with the distance detection can help determine if there is a threat present or not.

# Circuit with automated LED off (picture)



# Circuit with automated LED on (picture)



# Arduino Code (screenshot)

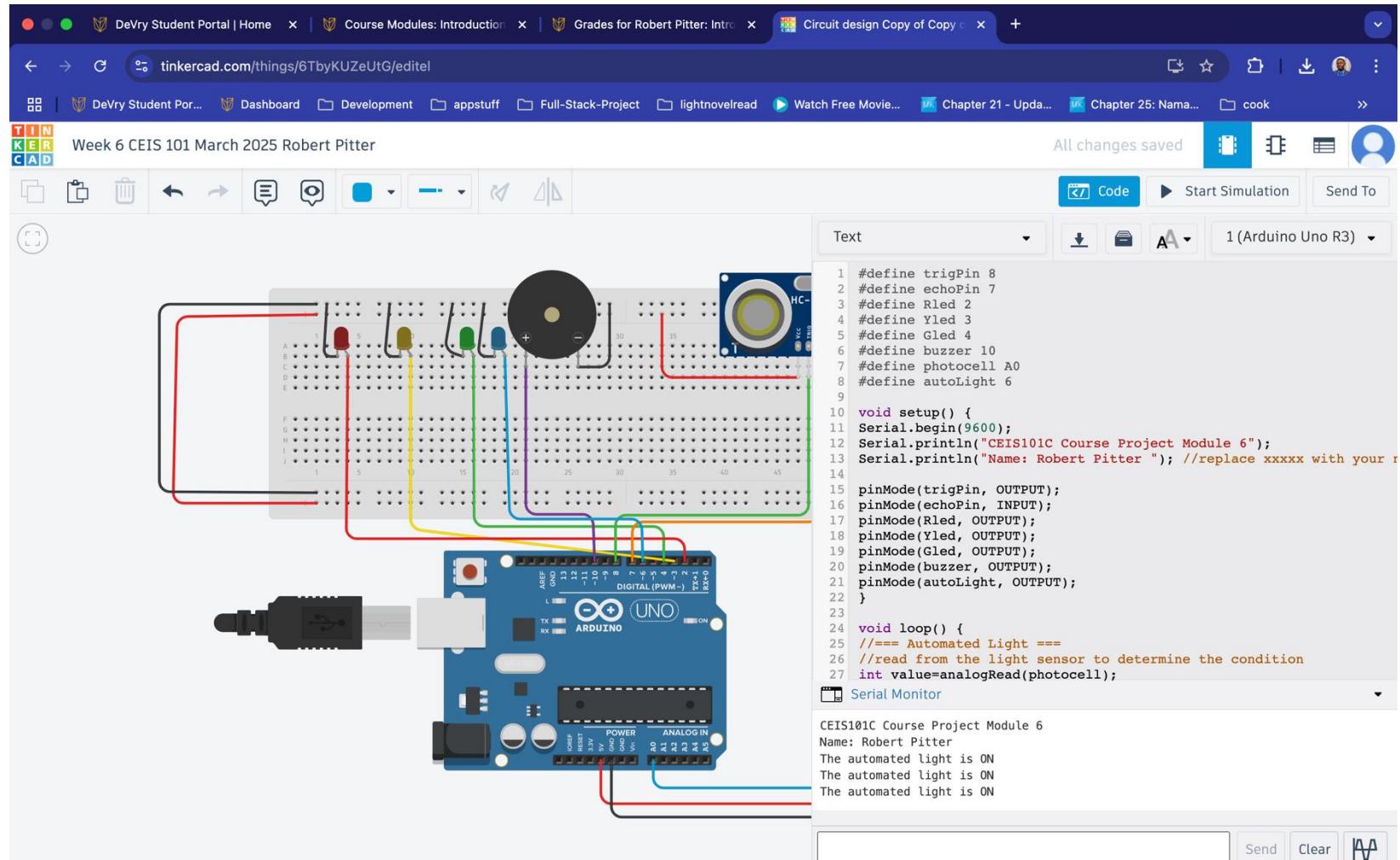
The screenshot shows a Tinkercad workspace for an Arduino Uno R3. On the left, a breadboard circuit is connected with various components: two HC-SR04 ultrasonic distance sensors, a 555 timer module, a 74HC14 optoisolator, and several LEDs and resistors. On the right, the Arduino code is displayed:

```
1 #define trigPin 8
2 #define echoPin 7
3 #define Rled 2
4 #define Gled 3
5 #define Bled 4
6 #define buzzer 10
7 #define photocell A0
8 #define autoLight 6
9
10 void setup() {
11   Serial.begin(9600);
12   Serial.println("CEIS101C Course Project Module 6");
13   Serial.println("Name: Robert Pitter"); //replace xxxx with your name
14
15   pinMode(trigPin, OUTPUT);
16   pinMode(echoPin, INPUT);
17   pinMode(Rled, OUTPUT);
18   pinMode(Gled, OUTPUT);
19   pinMode(Bled, OUTPUT);
20   pinMode(buzzer, OUTPUT);
21   pinMode(photocell, INPUT);
22 }
23
24 void loop() {
25   //*** Automated Light ***
26   //read from the light sensor to determine the condition
27   int value=analogRead(photocell);
28
29   //uncomment the line below, open the serial plotter to see the effect on the light
30   //Serial.println(value);
31
32   if (value > 450) {
33     digitalWrite(autoLight, HIGH);
34     Serial.println("The automated light is ON");
35   }
36   else {
37     digitalWrite(autoLight, LOW);
38   }
39
40   //*** Distance Sensor ***
41   long duration, distance, inches;
42
43   digitalWrite(trigPin, LOW);
44   delayMicroseconds(2);
45   digitalWrite(trigPin, HIGH);
46   delayMicroseconds(10);
47   digitalWrite(trigPin, LOW);
48
49   // Read the echo signal
50   //Read duration for roundtrip distance
51   duration = pulseIn(echoPin, HIGH);
52 }
```

The Serial Monitor at the bottom shows the output of the code:

```
CEIS101C Course Project Module 6
Name: Robert Pitter
The automated light is ON
```

# Serial Monitor (screenshot)



# Challenges/Lessons Learned

- The biggest challenges I faced were really just making sure I read and understood the project requirements and instructions. In one case, the circuit wasn't behaving as expected and I started added some logging to see what the issue was. It turns out I had a wire in a different spot than it should be so the component I was interested in was never live. I discovered after I went back and reviewed the instructions again.
- Outside of that, throughout this project I learned that it isn't so difficult to experiment with IOT projects without necessarily buying the components. Thus, I will do more experimentation in the future.

# Career Skills

- During the course of this project I developed circuit design and development skills, wiring skills and IOT programming skills.
- Overall, I believe the IOT programming skills will help me to advance my career since my field (data analysis) benefits from such skills.

# Conclusion

Overall, I enjoyed the course and will definitely revisit not only TinkerCad but also do some more IOT projects in the future.