A person wearing large black headphones is looking intently at a laptop screen. The person's hands are clasped near their chin, suggesting deep concentration or thought. The background is blurred, showing what appears to be a desk or office environment. The entire scene is overlaid with a semi-transparent white rectangular frame.

4/23/2022

SMART AUTOMATED HOME SYSTEM WITH SECURITY

CEIS 101

Jeanine Carhart

A young man with dark hair, wearing large black headphones, is looking intently at a laptop screen. He has his hands clasped under his chin, suggesting a thoughtful or focused state. The background is blurred, showing what appears to be a computer lab or office setting with other desks and equipment.

Hi,

4/23/2022

My name is Jeanine Carhart

I'm a student in the Cyber Security program at DeVry University.

I plan to get some hands-on work experience after receiving my certificate in the Fall of 2023 while continuing to earn my Bachelor's Degree in IT

Project: Smart Home Automation and Security System

Week 2.

- * Tinkercad Simulation (2)
- * Tinkercad Code

Week 3.

- * IoT Parts Inventory
- * Project Parts Inventory

Week 4.

- * Green LED – Door Closed
- * Red LED – Door Open
- * Yellow LED – Door Open
- * Arduino Code
- * Serial Monitor

Week 5.

- * Circuit with Green LED
- * Circuit with Yellow LED
- * Circuit with Red LED
- * Arduino Code
- * Data Plot

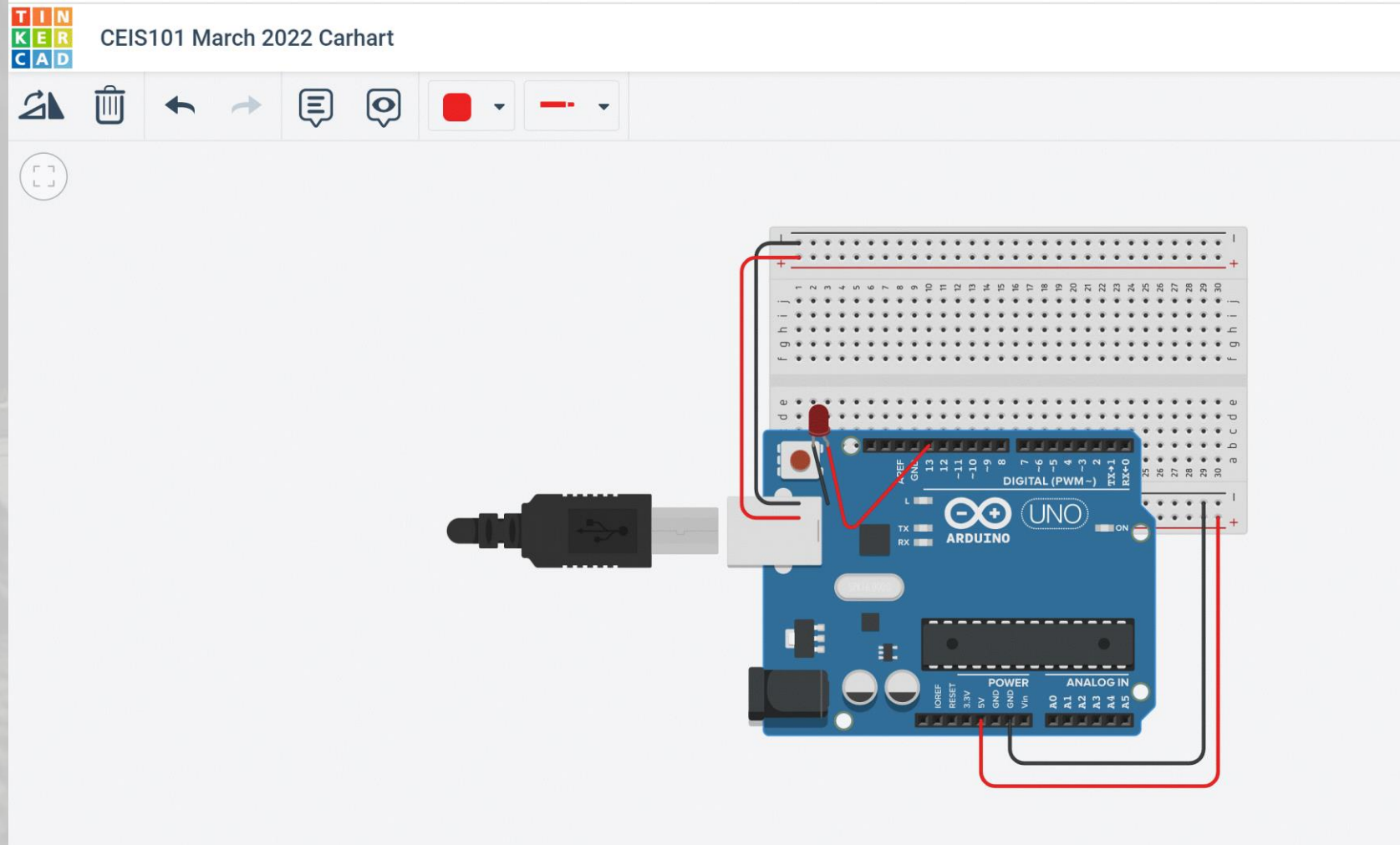
Week 6.

- * Automated Light On
- * Automated Light Off
- * Arduino Code
- * Serial Monitor

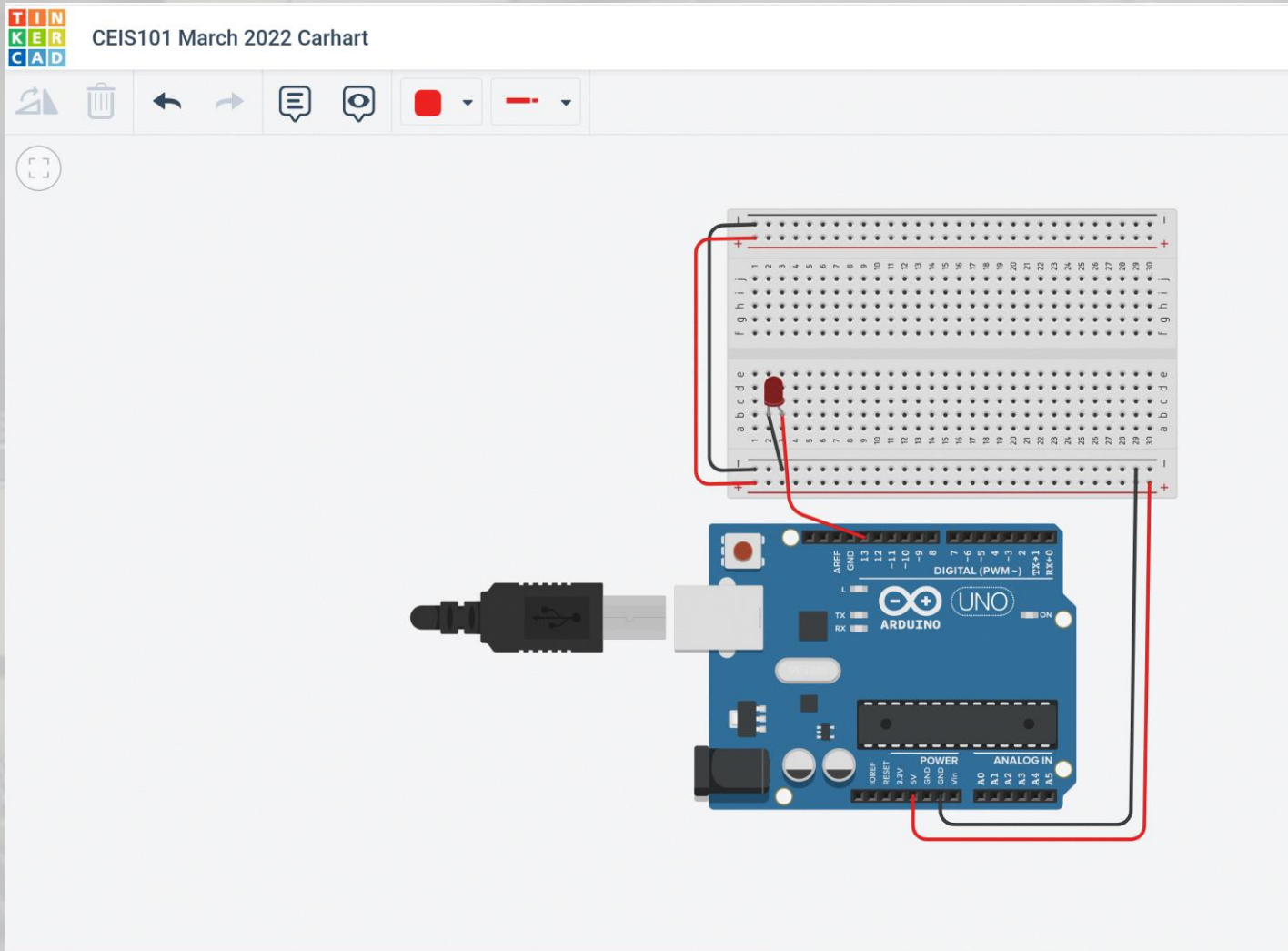
Week 2: Objectives

- To introduce simulation using Tinkercad Circuits
- To familiarize with hardware components required to build physical system
- To familiarize with programming logic and design of hardware

Project 1: Tinkercad Simulation 1



Project 1: Tinkercad Simulation 2



Project 1: Tinkercad Code

TINKERCAD CEIS101 March 2022 Carhart All changes saved

Code Start Simulation Send To

1 (Arduino Uno R3)

Blocks + Text

- Output
- Input
- Notation
- Control
- Math
- Variables

set built-in LED to HIGH

set pin 0 to HIGH

set pin 3 to 0

rotate servo on pin 0 to 0 degree

play speaker on pin 0 with tone 60

turn off speaker on pin 0

print to serial monitor hello world with

set RGB LED in pins 3 3 3

Serial Monitor

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

4/23/2022 7



Week 3: Objectives

Objectives

- To familiarize with hardware components for course project
- To learn how to build circuits with LEDs
- To learn how to program LEDs
- To learn how to initialize the Serial Monitor
- To learn how to send messages to the Serial Monitor

Week 3: Inventory



IoT Parts Inventory



Project Parts Inventory

Project 1: Parts Inventory

IoT Kit

UCTRONICS Kit
ESP32 (2)
Mini Router
Breadboards (3))
Digital Multi Meter
Patch Cable
USB to Micro USB (2)
LCD Modules (2)

Parts Components

Arduino Mega 2560
Resistor 10k Ω
Breadboard
LEDs
Ultrasonic Sensor
Photoresistor
Wires
Active Buzzer
USB Type B Cable

Project 1: IoT Parts Inventory



Project 1: Project Parts Inventory





Week 4: Objectives

- To learn how to connect multiple LEDs to the Arduino Mega Board
- To learn how to connect a buzzer to the Arduino Mega Board
- To learn how to use a wire to emulate an open or closed door
- To integrate an alarm into a security system
- To familiarize with conditional programming instructions

Week 4: Adding the Door Sensor



Green LED – Door Closed



Red LED – Door Open



Yellow LED – Door Open

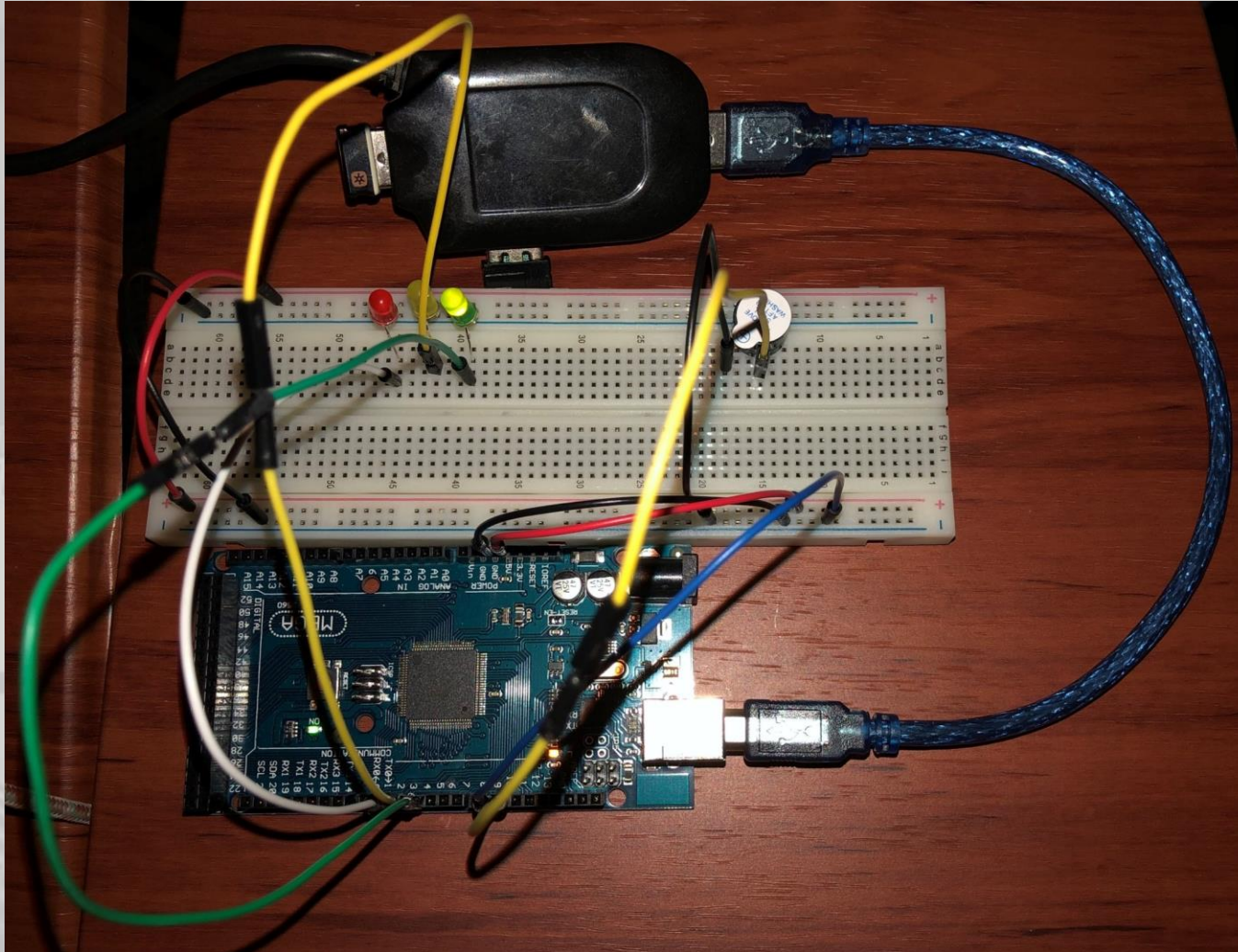


Arduino Code

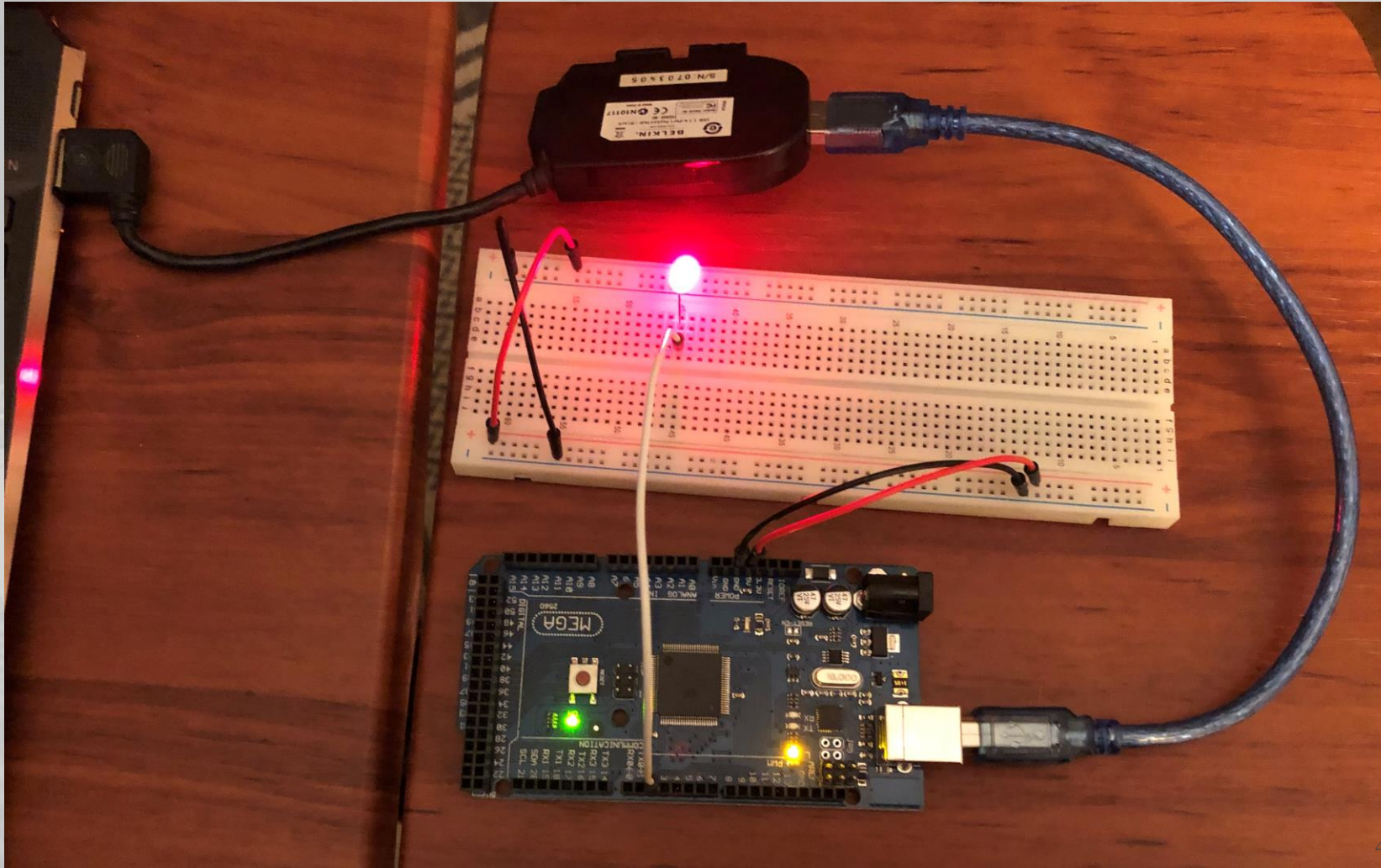


Serial Monitor

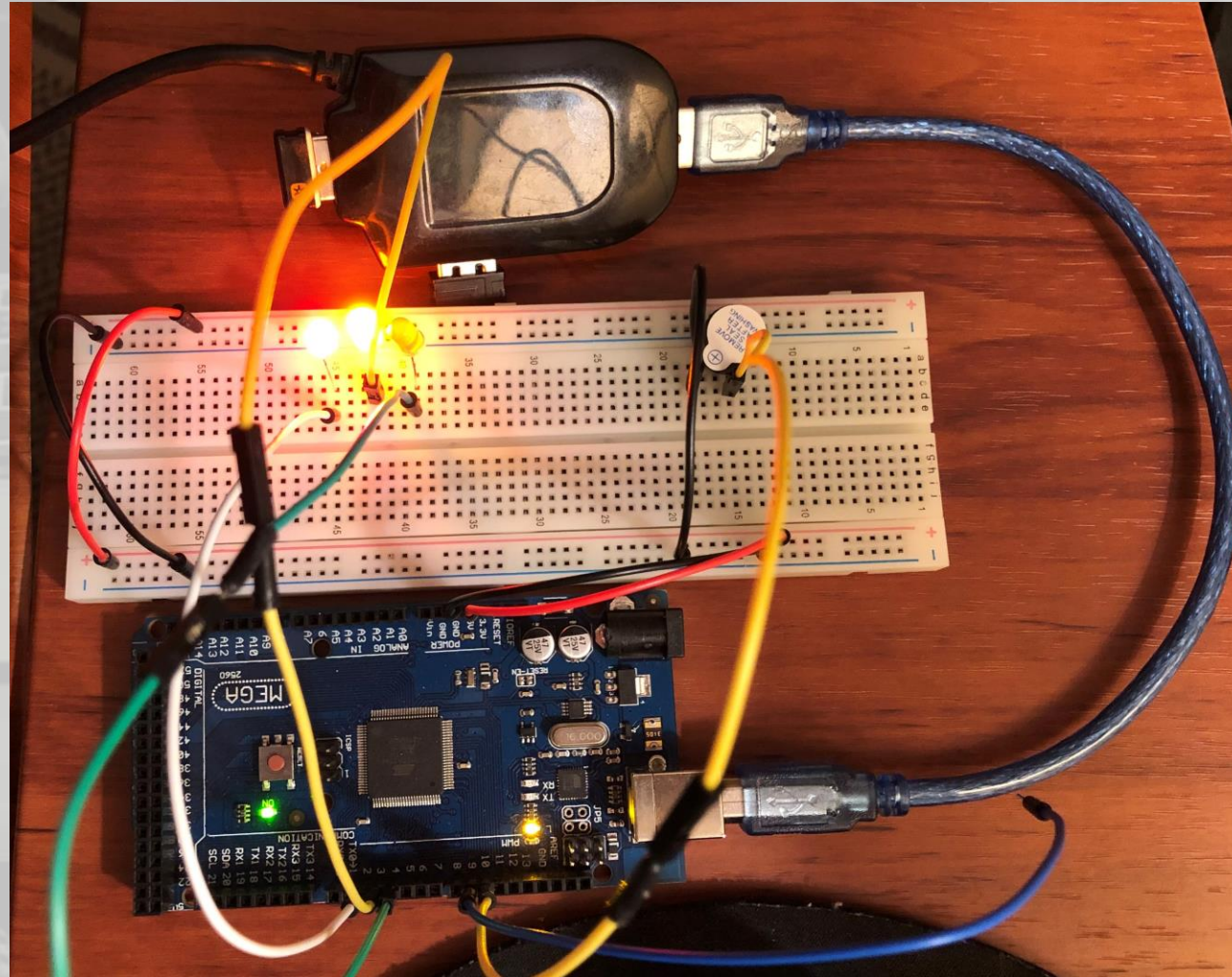
Project 1: Green LED On Door Closed



Project 1: Red LED on Door Open



Project 1: Yellow (& Red) LED on Door Open



Project 1: Arduino Code

```
Module_4_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Module_4_Project
#define Rled 2
#define Yled 3
#define Gled 4
#define buzzer 10
#define door 9
#define delaytime 1000 // === Second run, change to 100

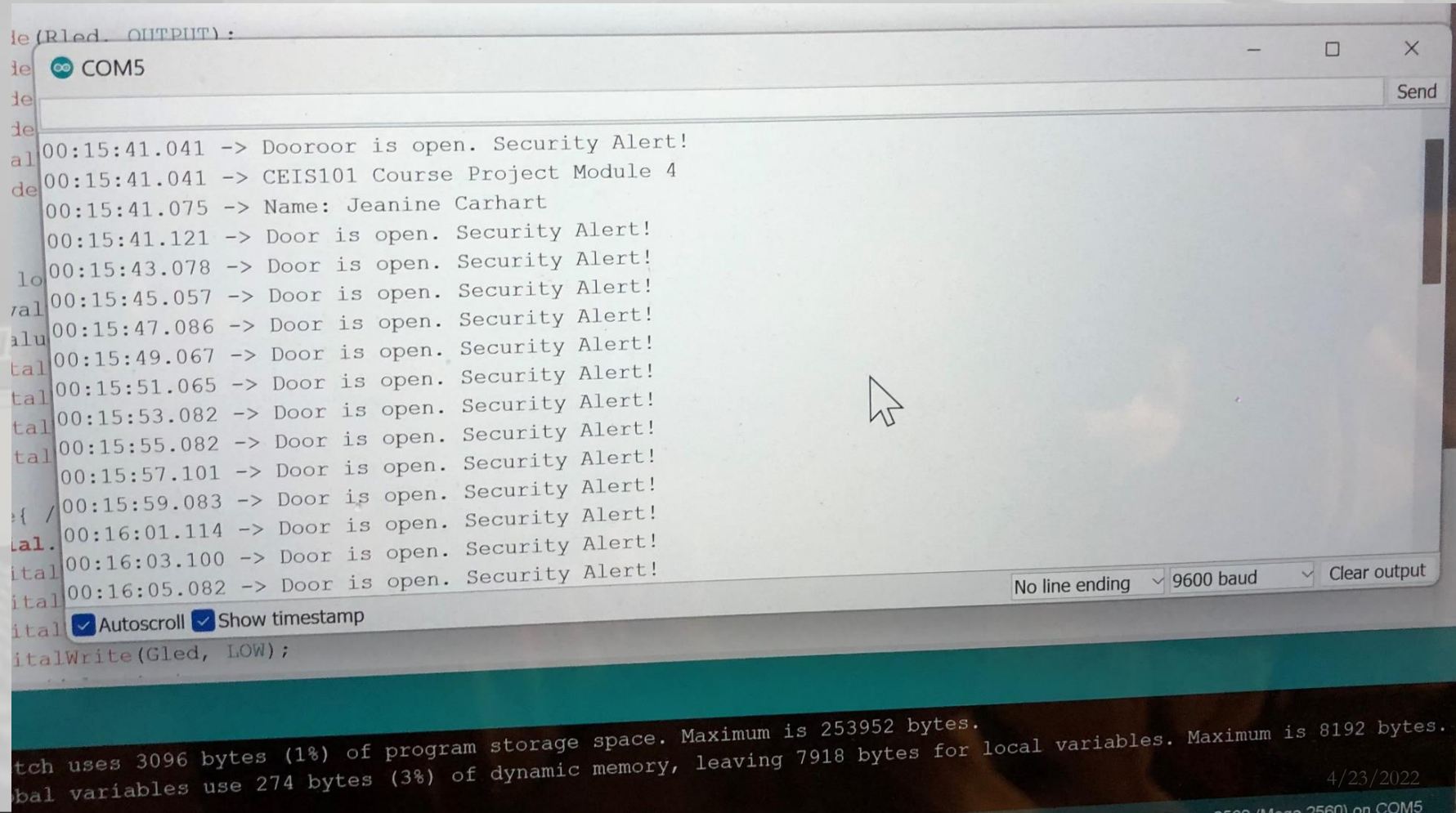
void setup() {
  Serial.begin(9600); // Set the baud rate
  Serial.println("CEIS101 Course Project Module 4");
  Serial.println("Name: Jeanine Carhart "); //replace xxxxx with your name

  pinMode(Rled, OUTPUT);
  pinMode(Yled, OUTPUT);
  pinMode(Gled, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
  pinMode(door, INPUT_PULLUP); //door sensor
}

void loop() {
  int value=digitalRead(door);
  if(value == 0) { // Door closed, no security threat
    digitalWrite(Rled, LOW);
    digitalWrite(Yled, LOW);
    digitalWrite(Gled, HIGH);
    digitalWrite(buzzer, LOW);
  }
  else{ // Door open, security threat
    Serial.println("Door is open. Security Alert! ");
    digitalWrite(Rled, HIGH);
    digitalWrite(Yled, HIGH);
    digitalWrite(buzzer, HIGH);
    digitalWrite(Gled, LOW);
  }
}

Sketch uses 3096 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 274 bytes (3%) of dynamic memory, leaving 7918 bytes for local variables. Maximum is 8192 bytes.
```

Project 1: Serial Monitor



The screenshot shows a serial monitor window titled "COM5" with a "Send" button. The window displays a series of messages received from a device, each starting with a timestamp and followed by a status report. The messages are:

```
00:15:41.041 -> Dooroor is open. Security Alert!  
00:15:41.041 -> CEIS101 Course Project Module 4  
00:15:41.075 -> Name: Jeanine Carhart  
00:15:41.121 -> Door is open. Security Alert!  
00:15:43.078 -> Door is open. Security Alert!  
00:15:45.057 -> Door is open. Security Alert!  
00:15:47.086 -> Door is open. Security Alert!  
00:15:49.067 -> Door is open. Security Alert!  
00:15:51.065 -> Door is open. Security Alert!  
00:15:53.082 -> Door is open. Security Alert!  
00:15:55.082 -> Door is open. Security Alert!  
00:15:57.101 -> Door is open. Security Alert!  
00:15:59.083 -> Door is open. Security Alert!  
00:16:01.114 -> Door is open. Security Alert!  
00:16:03.100 -> Door is open. Security Alert!  
00:16:05.082 -> Door is open. Security Alert!
```

At the bottom of the window, there are settings for "No line ending", "9600 baud", and "Clear output". Below the window, there are two checked checkboxes: "Autoscroll" and "Show timestamp".

Below the serial monitor window, there is a status bar with the following text:

```
itch uses 3096 bytes (1%) of program storage space. Maximum is 253952 bytes.  
bal variables use 274 bytes (3%) of dynamic memory, leaving 7918 bytes for local variables. Maximum is 8192 bytes.
```

The date "4/23/2022" is displayed in the bottom right corner of the status bar.

Week 5: Objectives

- To learn how to connect an ultrasonic range finder to the Mega 2560 board
- To learn how to write code that uses an ultrasonic range finder
- To learn how to write code that makes the buzzer produce sounds
- To learn how to display data to the serial plotter
- To practice analyzing data generated by sensors

Week 5: Adding the Distance Sensor



Circuit with Green LED



Circuit with Yellow LED



Circuit with Red LED



Arduino Code



Data Chart

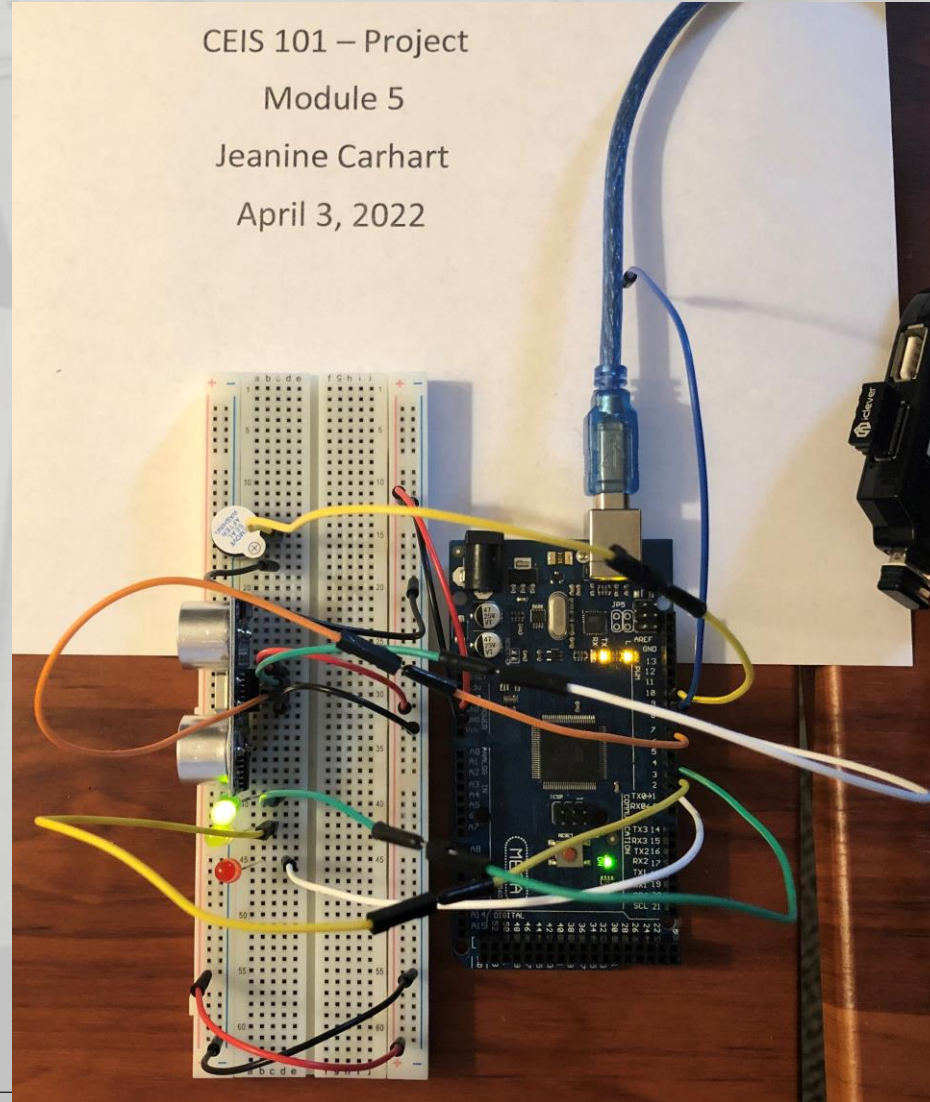
Project 1: Circuit with Green LED

CEIS 101 – Project

Module 5

Jeanine Carhart

April 3, 2022



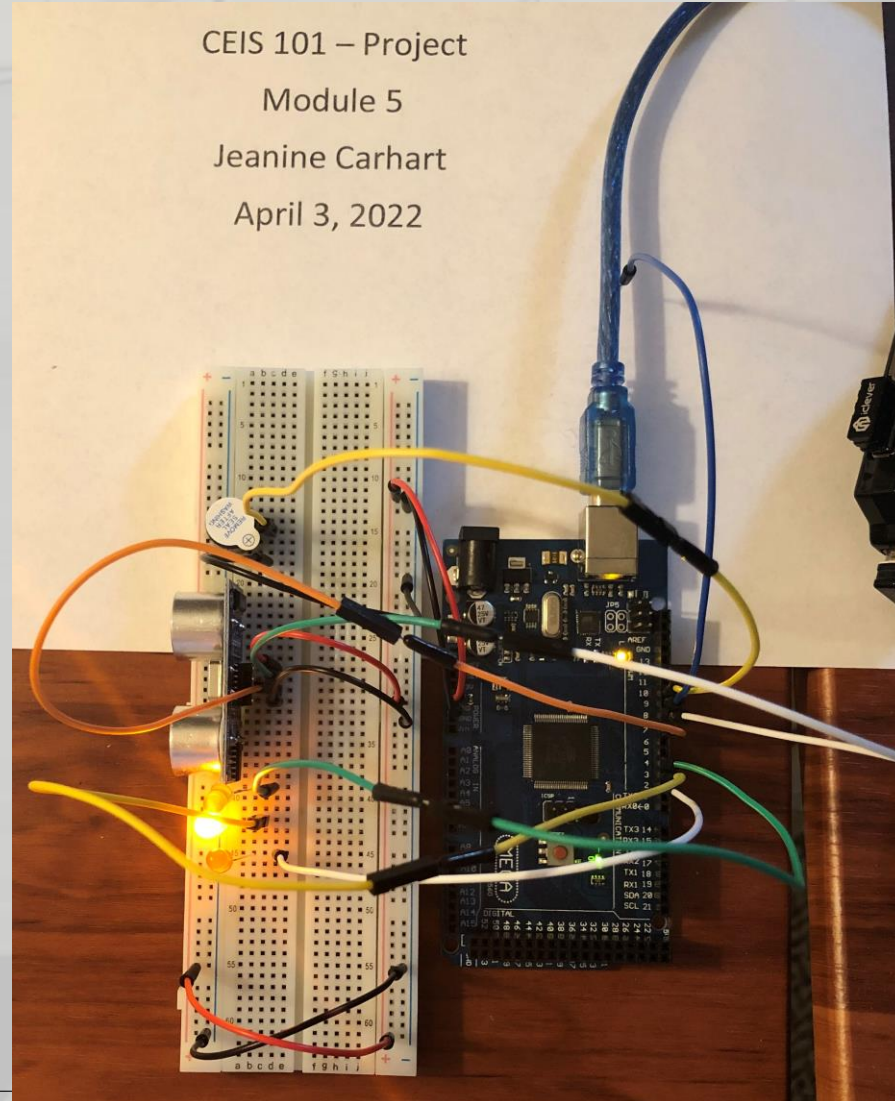
Project 1: Circuit with Yellow LED

CEIS 101 – Project

Module 5

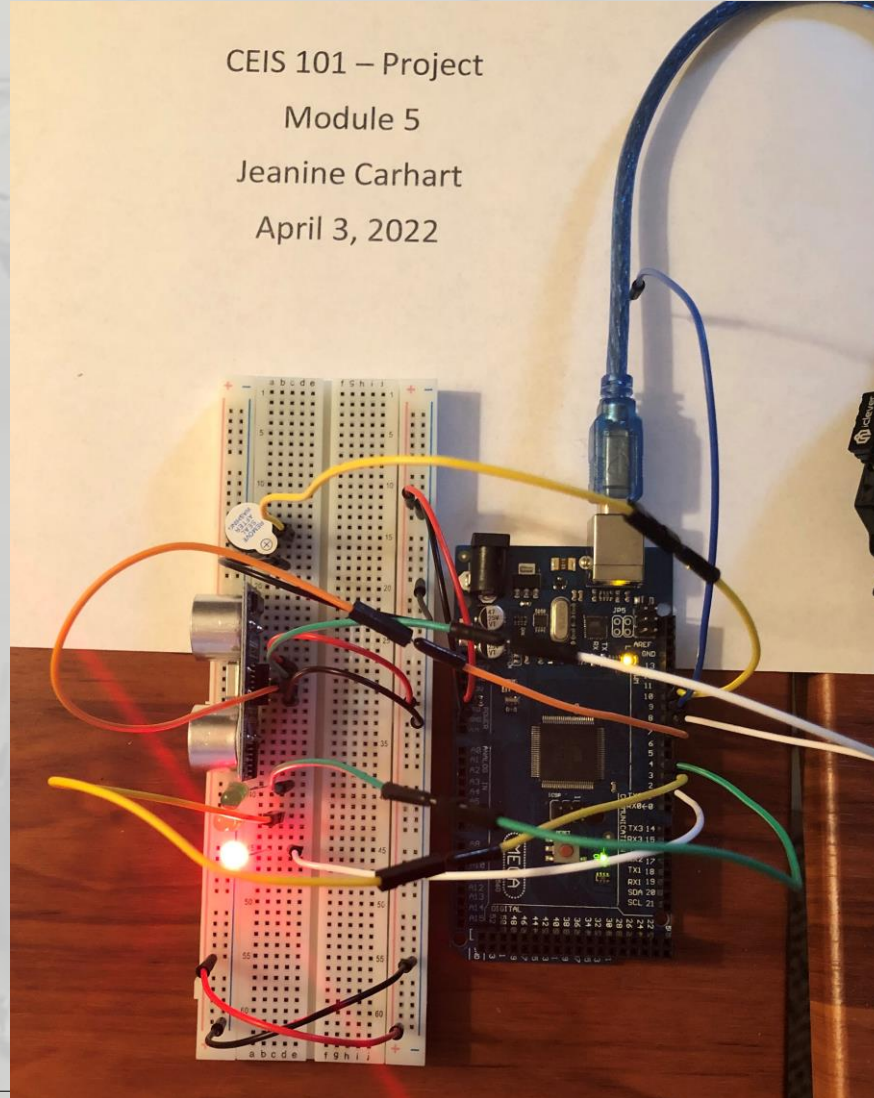
Jeanine Carhart

April 3, 2022



Project 1: Circuit with Red LED

CEIS 101 – Project
Module 5
Jeanine Carhart
April 3, 2022



Project 1: Circuit with Arduino Code

```
Module_5_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Module_5_Project
#define trigPin 8
#define echoPin 7
#define Rled 2
#define Yled 3
#define Gled 4
#define buzzer 10

void setup() {
  Serial.begin(9600);
  Serial.println("CEIS101 Course Project Module 5");
  Serial.println("Name: Jeanine Carhart "); //replace xxxxx with your name

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Rled, OUTPUT);
  pinMode(Yled, OUTPUT);
  pinMode(Gled, OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  long duration, distance, inches;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echo signal
  duration = pulseIn(echoPin, HIGH); // Read duration for roundtrip distance
  distance = (duration / 2) * 0.0135 ; // Convert duration to one way distance in units of inches

  if (distance <= 12) { // Outer IF statement units of inches

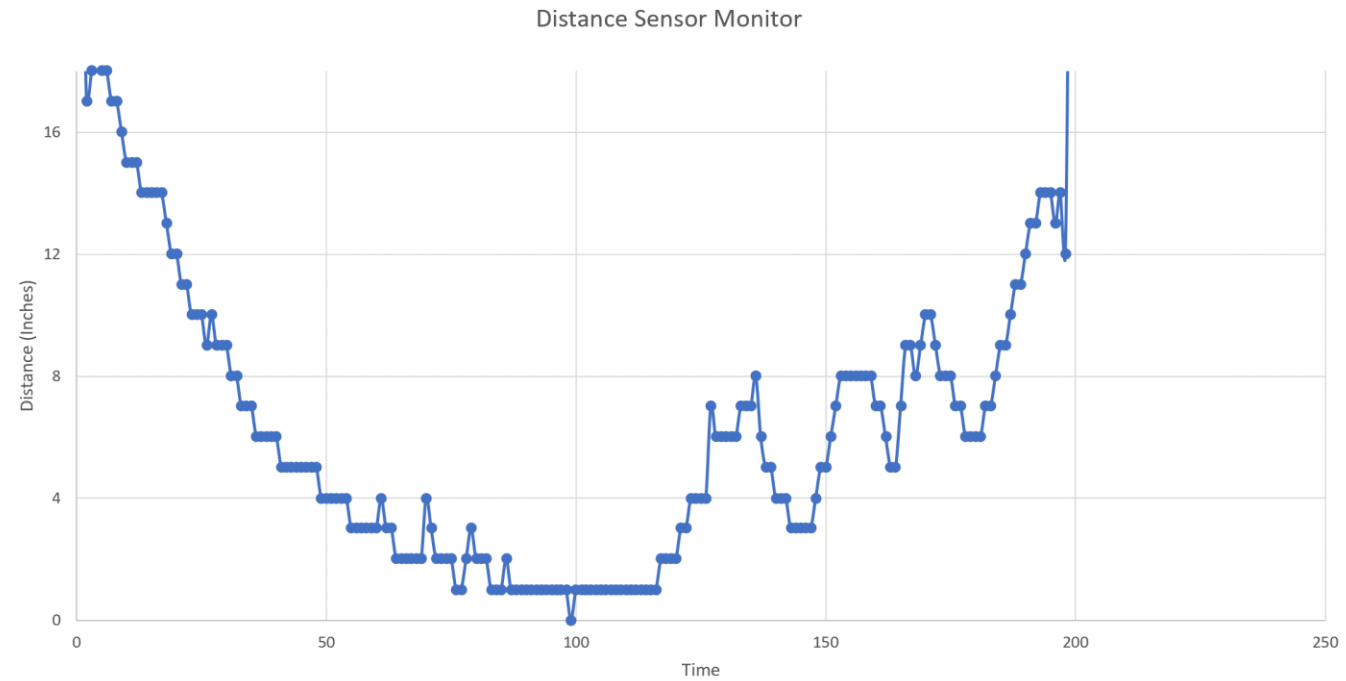
Code formatted for the Arduino forum has been copied to the clipboard.

Sketch uses 4192 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 242 bytes (2%) of dynamic memory, leaving 7950 bytes for local variables. Maximum is 8192 bytes.

68 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM5
```

Project 1: Data Chart

Data Plot: Distance Sensor Monitor





Week 6: Objectives

- 🕒 To learn how to connect a light sensor to the Mega 2560 board
- 🕒 To learn how to write code that uses a light sensor
- 🕒 To learn how to write code that turns on light based on the reading from the light sensor

Week 6: Adding Automated Light



Automated Light On



Automated Light Off



Arduino Code



Serial Monitor

Project 1: Automated Light On

CEIS 101 – Project

Module 6

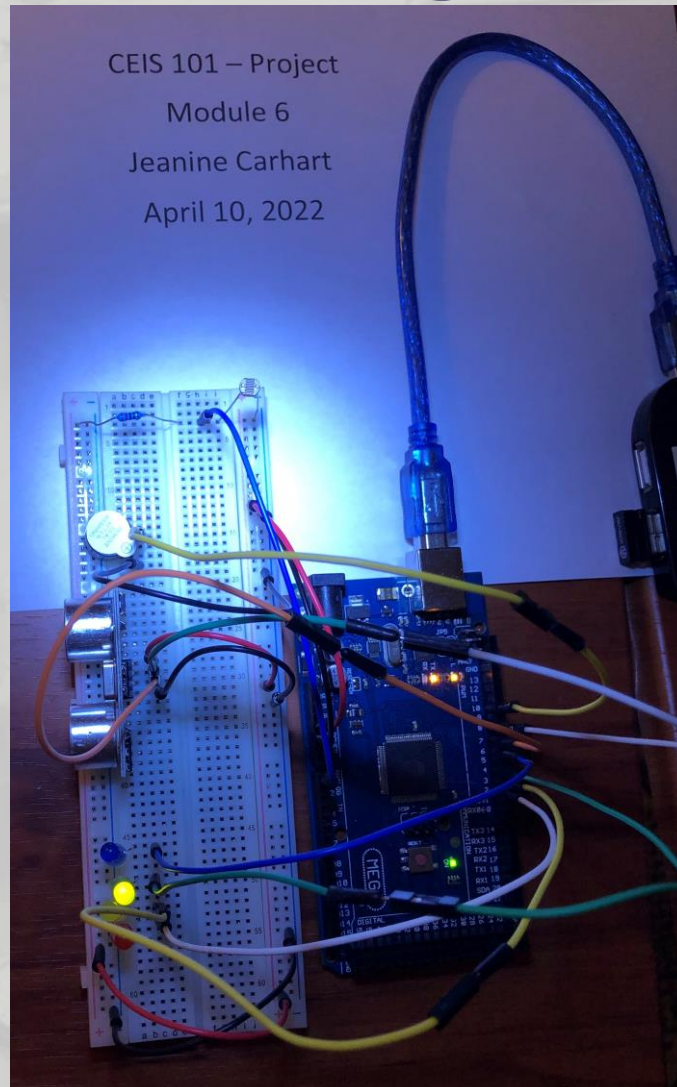
Jeanine Carhart

April 10, 2022



Project 1: Automated Light Off

CEIS 101 – Project
Module 6
Jeanine Carhart
April 10, 2022



Project 1: Arduino Code

```
Module_6_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Module_6_Project
#define trigPin 8
#define echoPin 7
#define Rled 2
#define Yled 3
#define Gled 4
#define buzzer 10
#define photocell A0
#define autoLight 6

void setup() {
  Serial.begin(9600);
  Serial.println("CEIS101 Course Project Module 6");
  Serial.println("Name: Jeanine Carhart "); //replace xxxxx with your name

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Rled, OUTPUT);
  pinMode(Yled, OUTPUT);
  pinMode(Gled, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(autoLight, OUTPUT);
}

void loop() {
  //== Automated Light ==
  int value=analogRead(photocell); // Read the value from the light sensor to determine condition

  //Serial.println(value); uncomment this line and open serial plotter to see the effect of light

  if (value > 450) {
    digitalWrite(autoLight, HIGH);
    Serial.println("The automated light is ON");
  }
  else {
    digitalWrite(autoLight, LOW);
  }
}

Sketch uses 4122 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 294 bytes (3%) of dynamic memory, leaving 7898 bytes for local variables. M
28
```

Project 1: Serial Monitor

```
le (Rled, OUTPUT);
le COM5
le
de
de
al 00:15:41.041 -> Dooroor is open. Security Alert!
de 00:15:41.041 -> CEIS101 Course Project Module 4
00:15:41.075 -> Name: Jeanine Carhart
lo 00:15:41.121 -> Door is open. Security Alert!
val 00:15:43.078 -> Door is open. Security Alert!
alu 00:15:45.057 -> Door is open. Security Alert!
tal 00:15:47.086 -> Door is open. Security Alert!
tal 00:15:49.067 -> Door is open. Security Alert!
tal 00:15:51.065 -> Door is open. Security Alert!
tal 00:15:53.082 -> Door is open. Security Alert!
tal 00:15:55.082 -> Door is open. Security Alert!
e / 00:15:57.101 -> Door is open. Security Alert!
tal. 00:15:59.083 -> Door is open. Security Alert!
ital 00:16:01.114 -> Door is open. Security Alert!
ital 00:16:03.100 -> Door is open. Security Alert!
ital 00:16:05.082 -> Door is open. Security Alert!
italWrite(Gled, LOW);

tch uses 3096 bytes (1%) of program storage space. Maximum is 253952 bytes.
bal variables use 274 bytes (3%) of dynamic memory, leaving 7918 bytes for local variables. Maximum is 8192 bytes.
```




Project Challenges

- Module 3 – Arduino IDE and Chrome didn't want to work together. Uninstalling and re-installing version 1.8.19 finally got it running.
- The yellow light blinked continuously when it wasn't supposed to and the red light wouldn't light up at all for several hours.
- No resolution from the Huddle or from consultation with a couple professors. Listening to the class recording brought up a point that I decided to try, so I shut down Arduino and restarted, loaded the code and it finally came on!
- No other real issues that I couldn't handle.

New Skills Acquired

- Learning to work with components to make electronic devices
- Introduction to Tinkercad and Arduino
- Learned a lot about the IoT, which is helpful.
- Reacquainted me with Automation
- Learned a bit about Distance Sensors



Project Summary

- This class was so enjoyable! I'd never heard of Tinkercad or Arduino, so working with it was a treat. I'll be using both to create something soon.
- It was nice to discover that I actually liked working with circuit boards and wires to see if I could make a device work.
- Doing a bit of coding was a good way to prep us for our next class, Python.



THANK YOU!

Jeanine Carhart
jcarhart3@my.devry.edu