AUDRUINO IDE SMART HOME SYSTEM

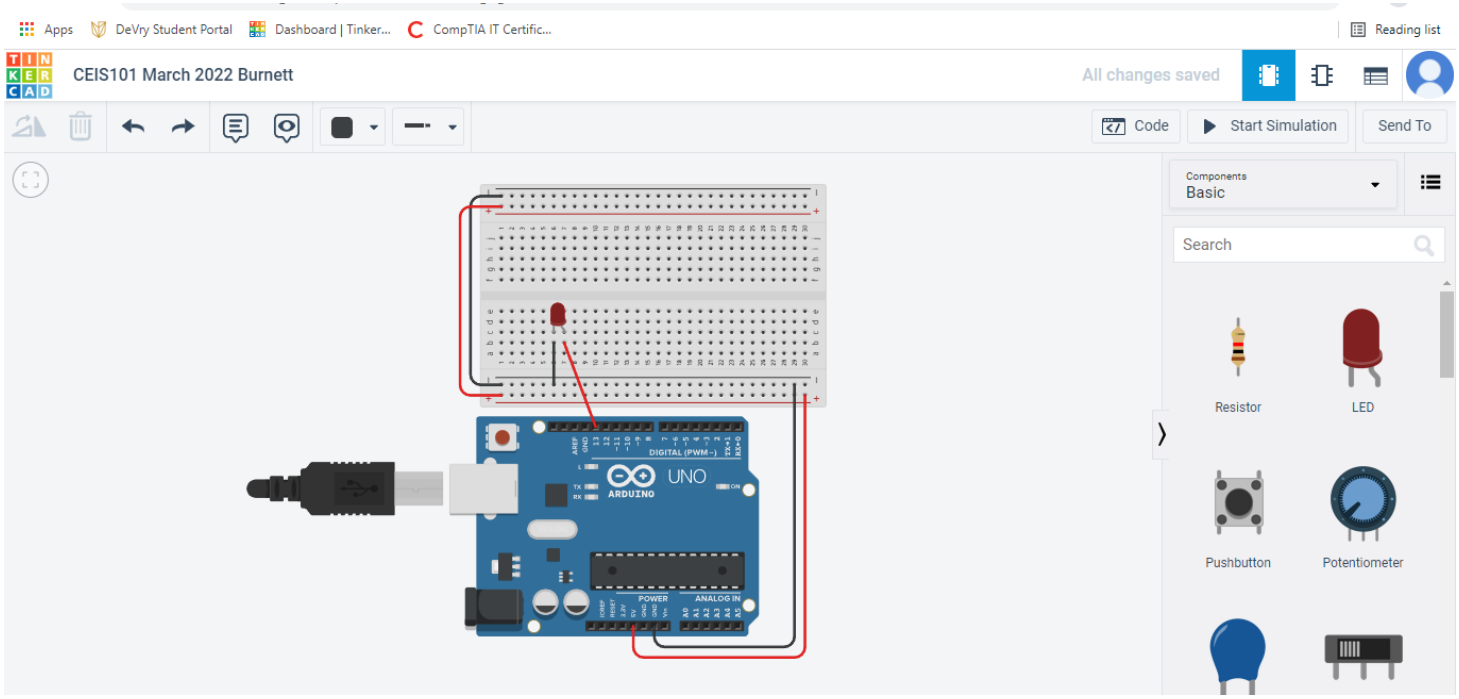
CEIS101 | Faith Burnett | DeVry University

CIRCUIT SIMULATION

- In the first part of the project, I will be using Tinkercad, a 3-D design and modeling program
- This is to gain an understanding of how a simple circuit for a blinking LED works

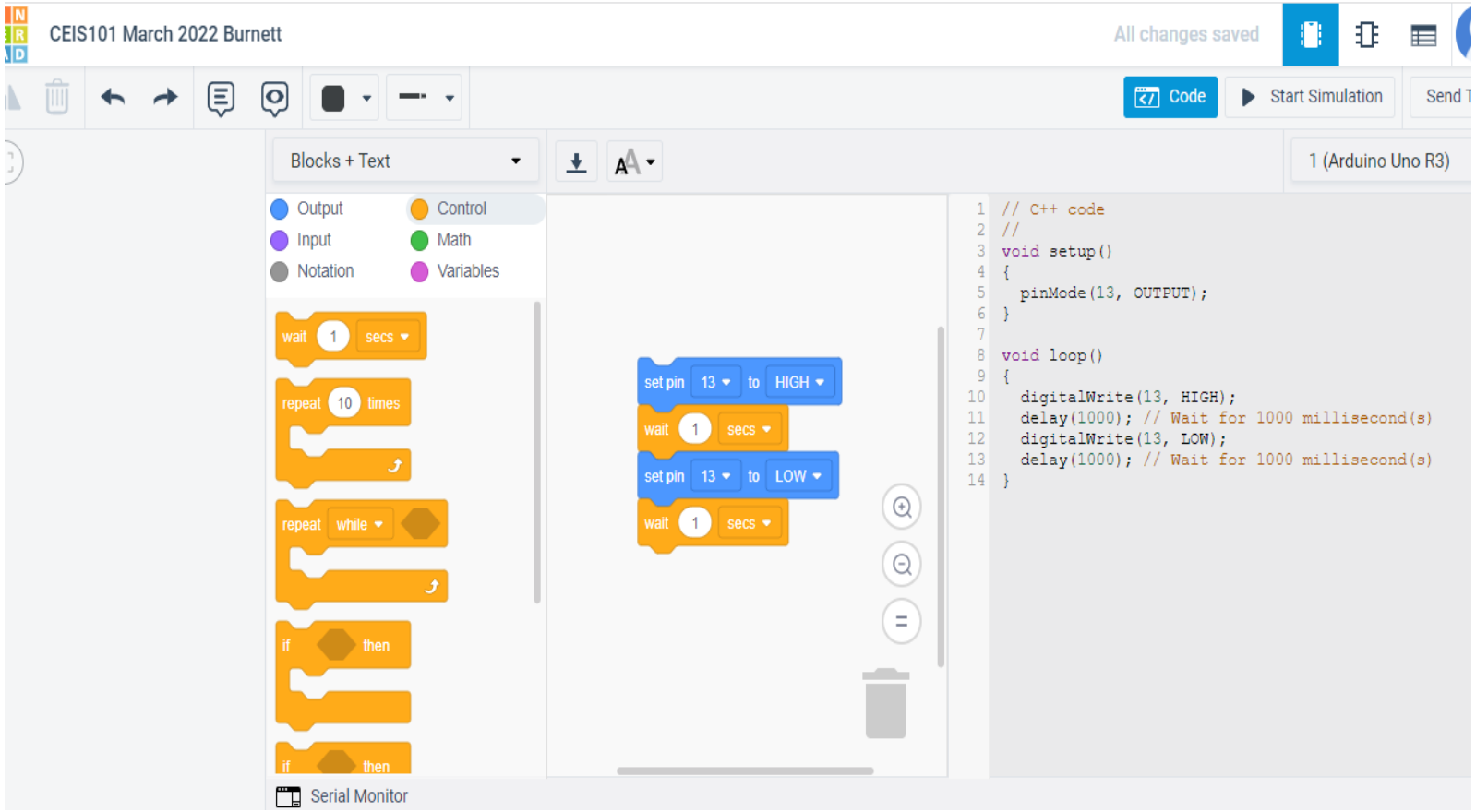
CONSTRUCTION OF CIRCUIT SIMULATION

- To the right you will see the Adruino breadboard starter, two jumper wires connect each set of rails ground to ground and power to power
- To connect the Arduino to the breadboard we also use two jumper wires ground connects to the ground pin and power connects to the 5v pin
- Next, we connect the LED, the shorter leg(Cathode) will be connected to the ground rail, The longer leg(Anode) will be connected to Pin13



CODING FOR LED CIRCUIT

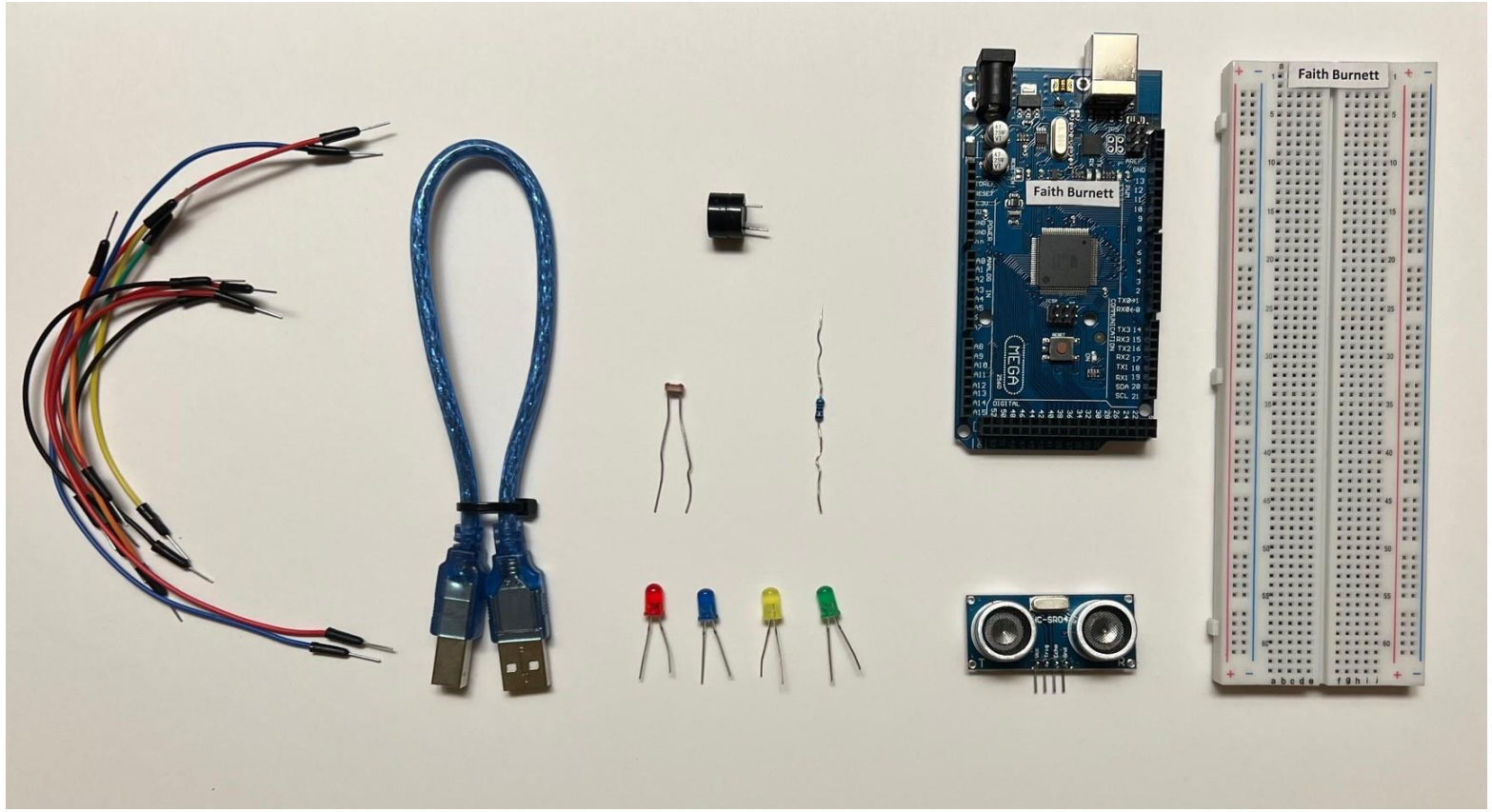
- In the setup function pin13 is being initialized as an output
- In the loop function pin13 will be set to go on(High) and off(Low) with a small delay in between depending on desired blink speed



IOT KIT INVENTORY

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

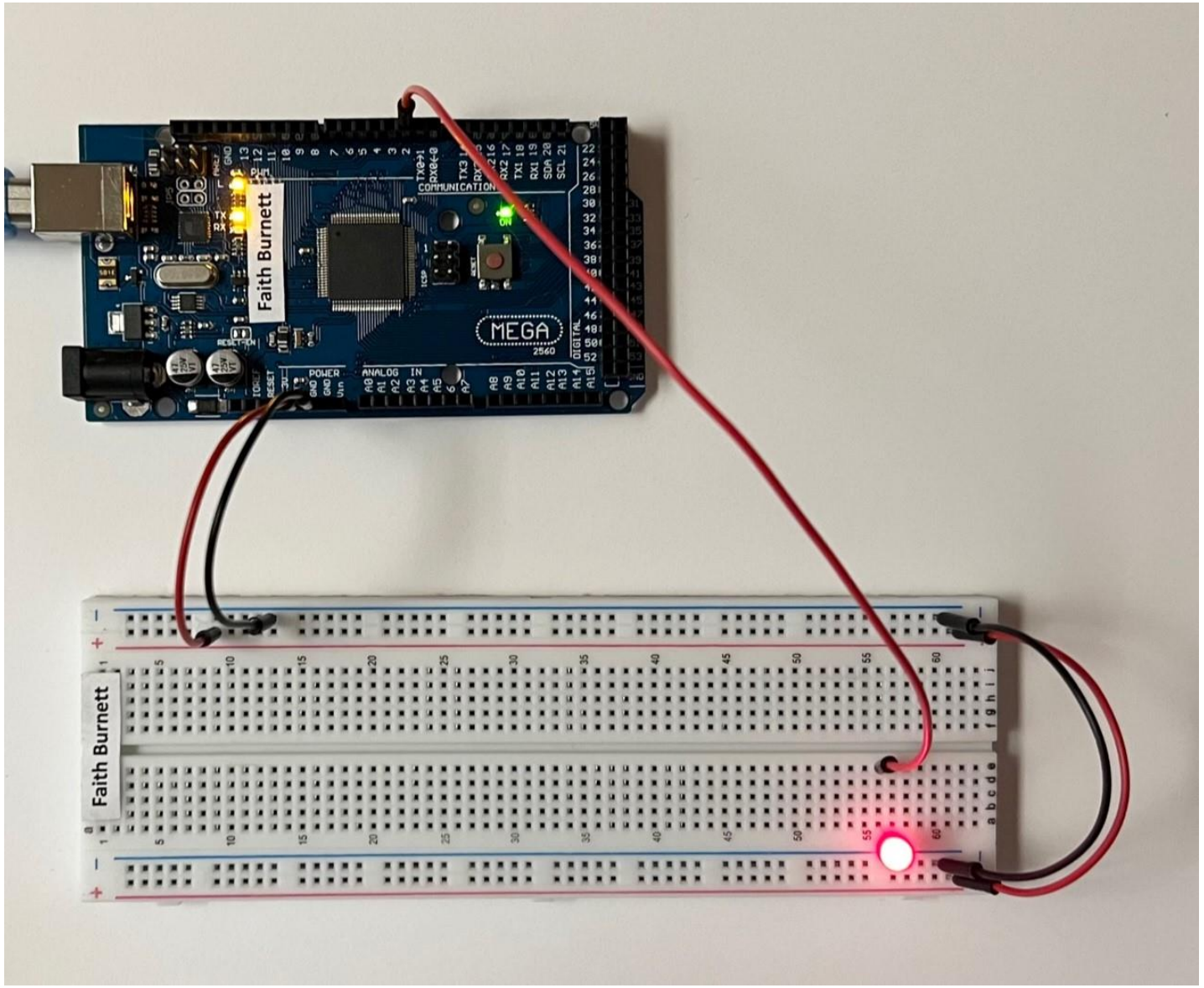
PROJECT COMPONENTS



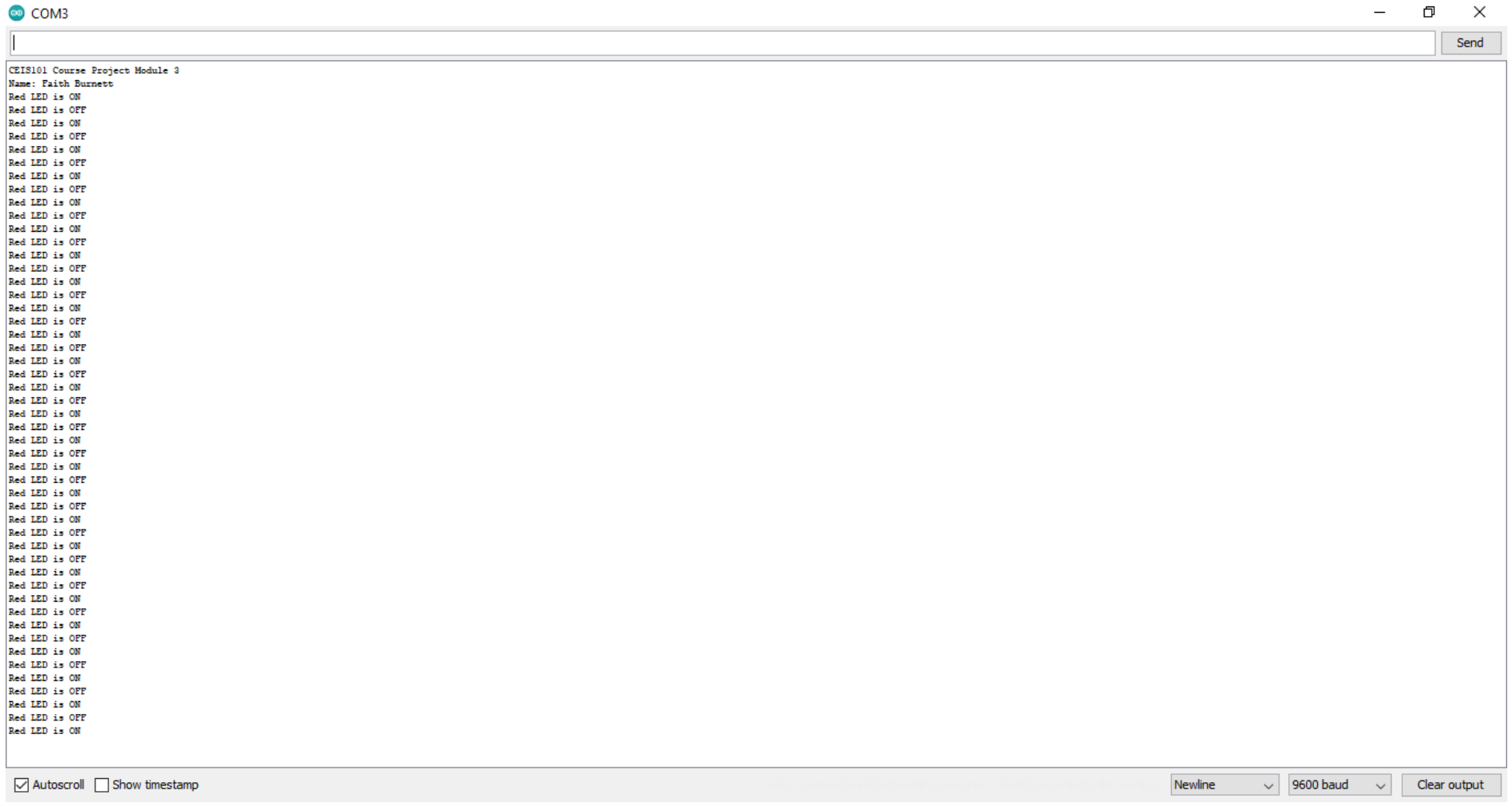
- Adruino Mega 2560
- Breadboard
- Resistor 10kΩ
- LEDs
- Ultrasonic Sensor
- Active Buzzer
- Photoresistor
- Wires
- USB Type B cable

THE PHYSICAL CIRCUIT

- As demonstrated in the simulated circuit the Arduino and rails are connected using the jumper wires
- The LED is connected to the ground rail and then to pin2(any of the digital pins could be used in place of pin2)



SERIAL MONITOR



```
CEIS101 Course Project Module 3
Name: Faith Burnett
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
Red LED is OFF
Red LED is ON
```

☑ Autoscroll ☐ Show timestamp

Newline

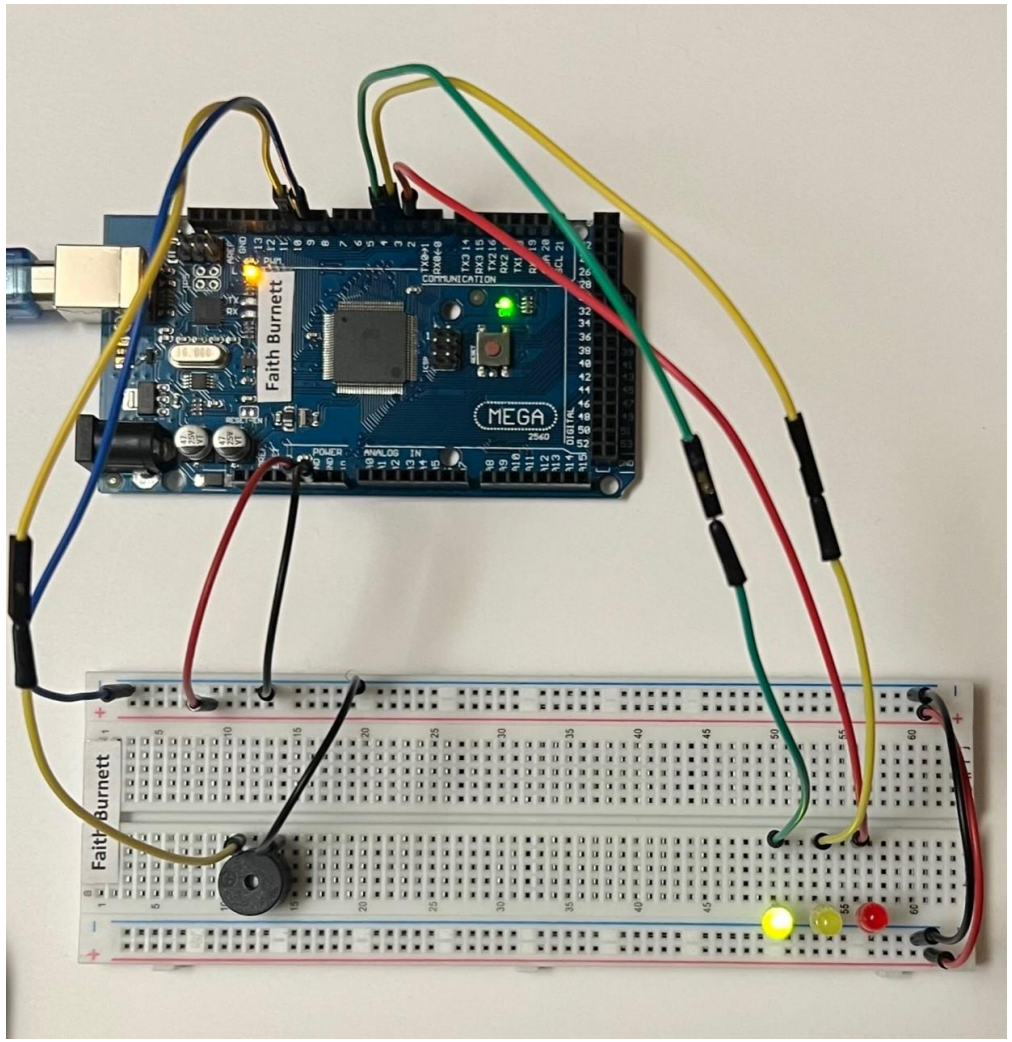
9600 baud

DOOR SENSOR

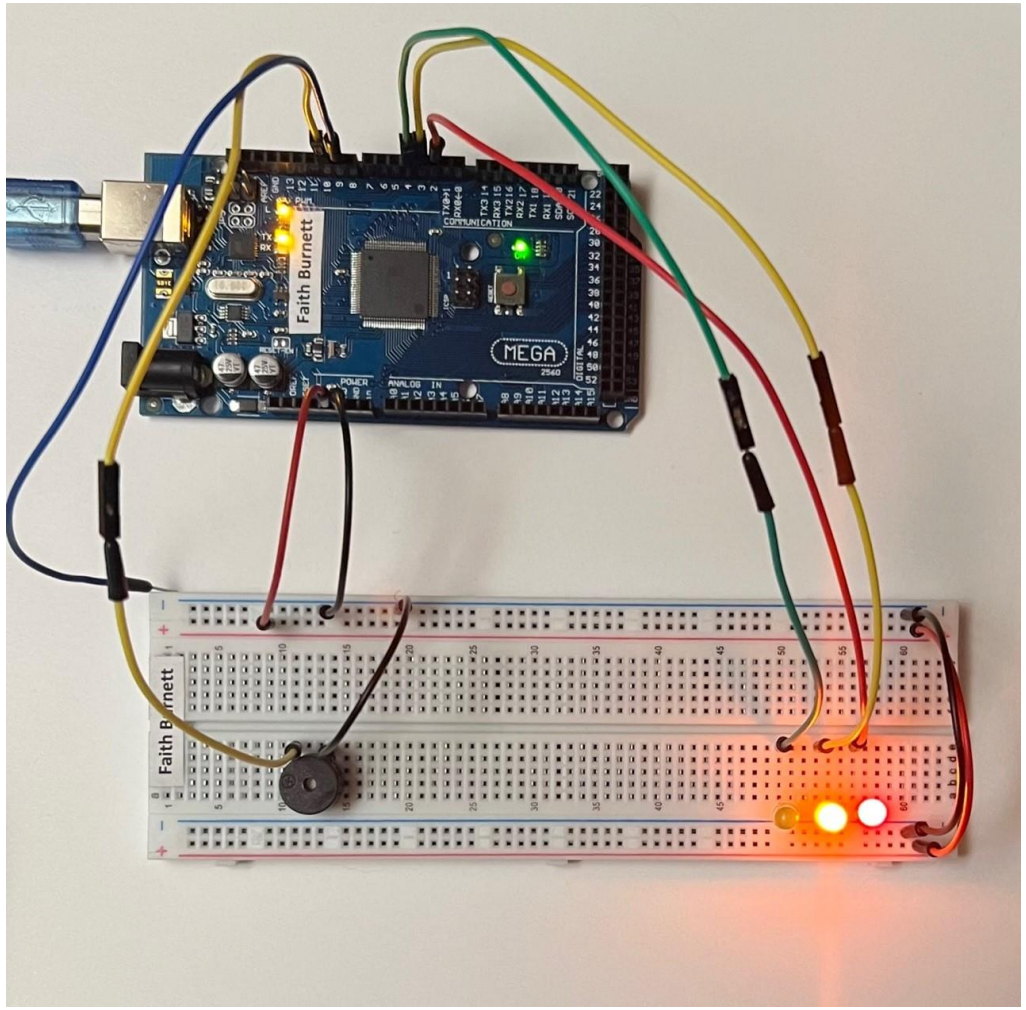
- In this section of the project, I will be adding two additional LEDs, representing levels of security risk, and an active buzzer representing an alarm
- A blue jumper wire will act as the door

GREEN LED ON/DOOR CLOSED

- Green and yellow LEDs are now connected to pins 3 and 4
- The active buzzer is connected, black wire to ground on breadboard and yellow to pin 10 for power
- The blue wire(“the door”) is connected to pin9 allowing the green LED to be lit and showing us that the door is closed

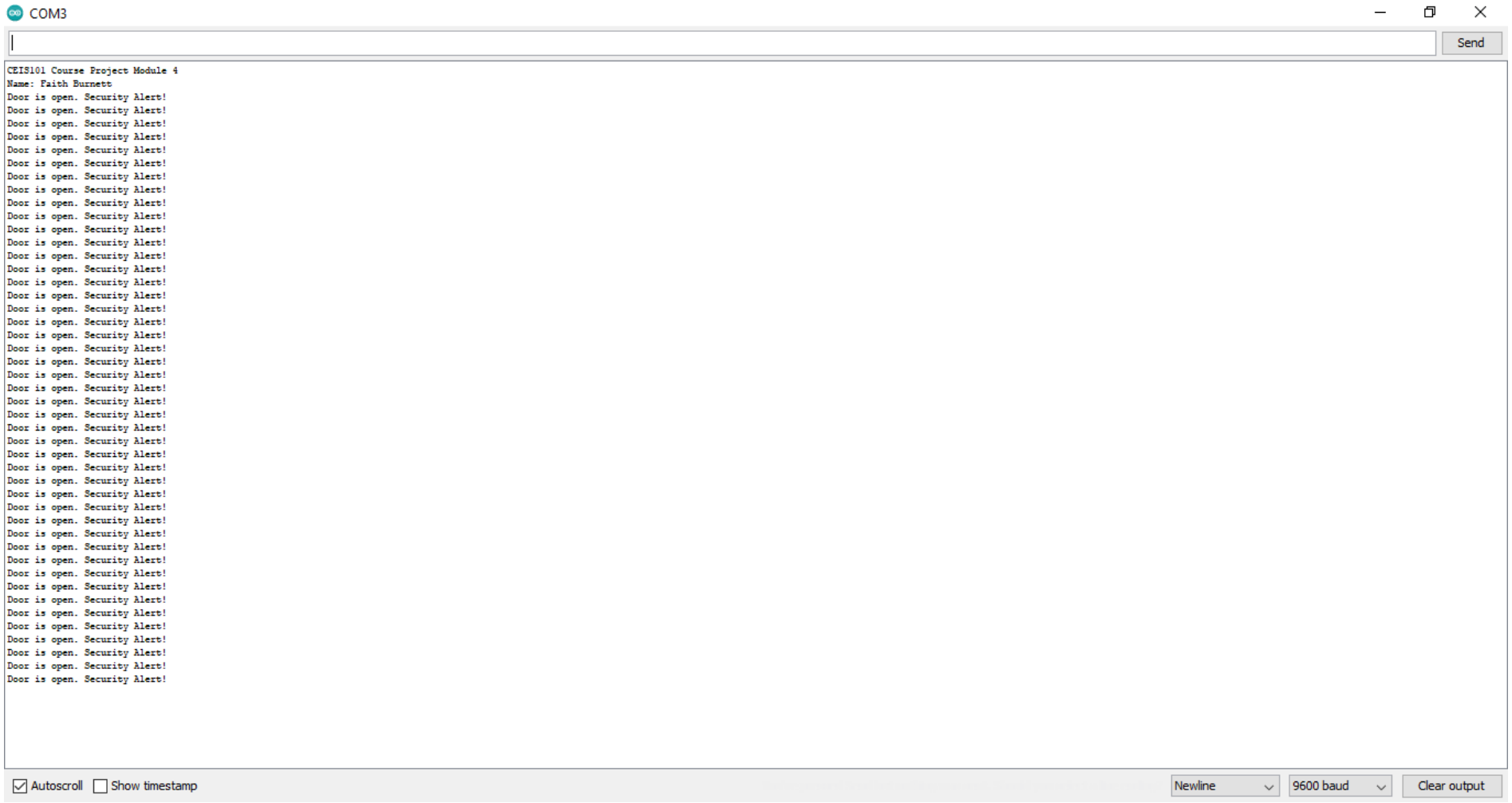


YELLOW AND RED LED ON/DOOR OPEN

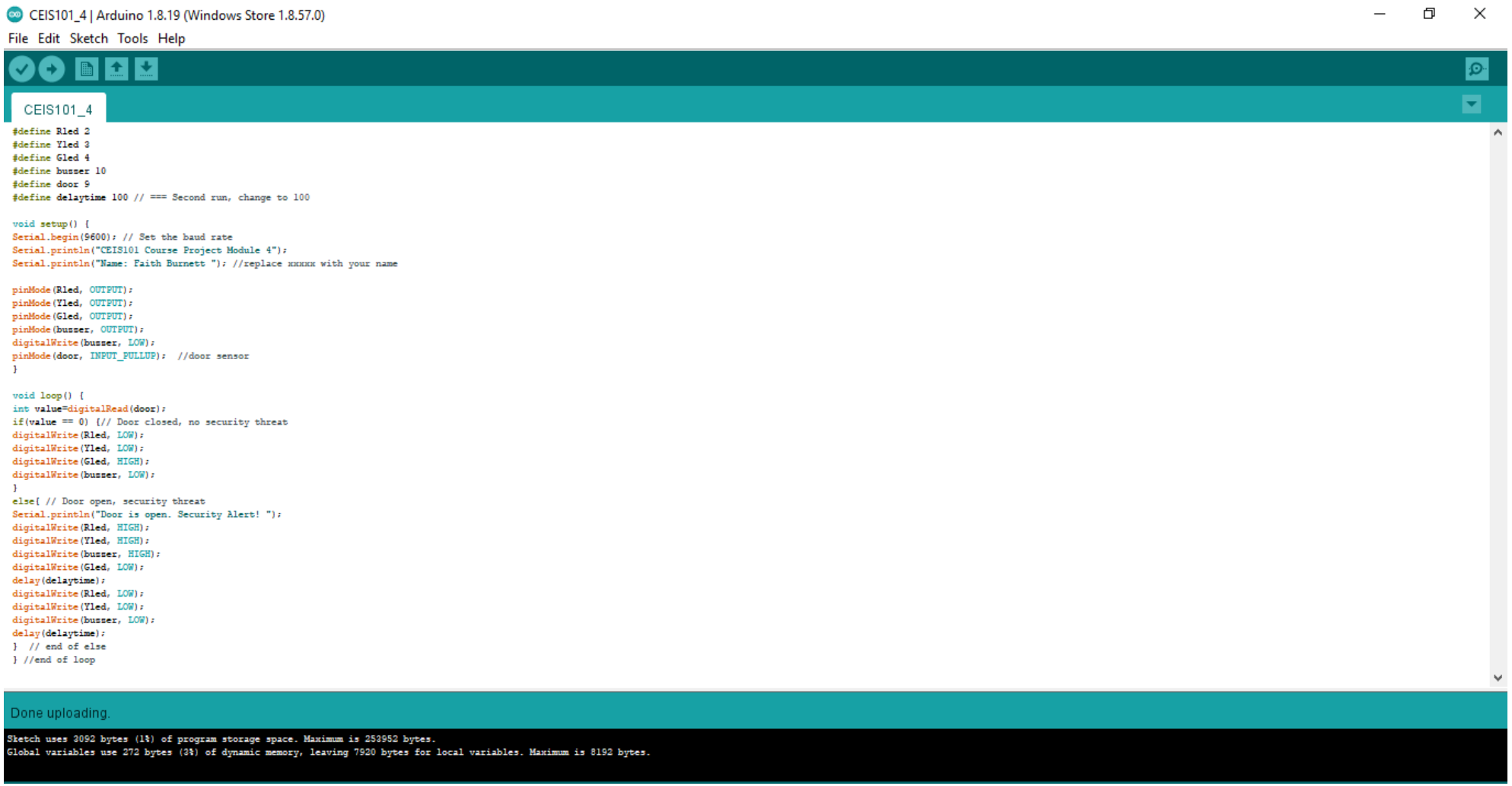


- The blue wire has been disconnected
- The active buzzer is going off and yellow and red LEDs are on
- The alarm tells me that the door is open, the lights are possibly a deterrent for culprit

SERIAL MONITOR



CODE FOR DOOR SENSOR

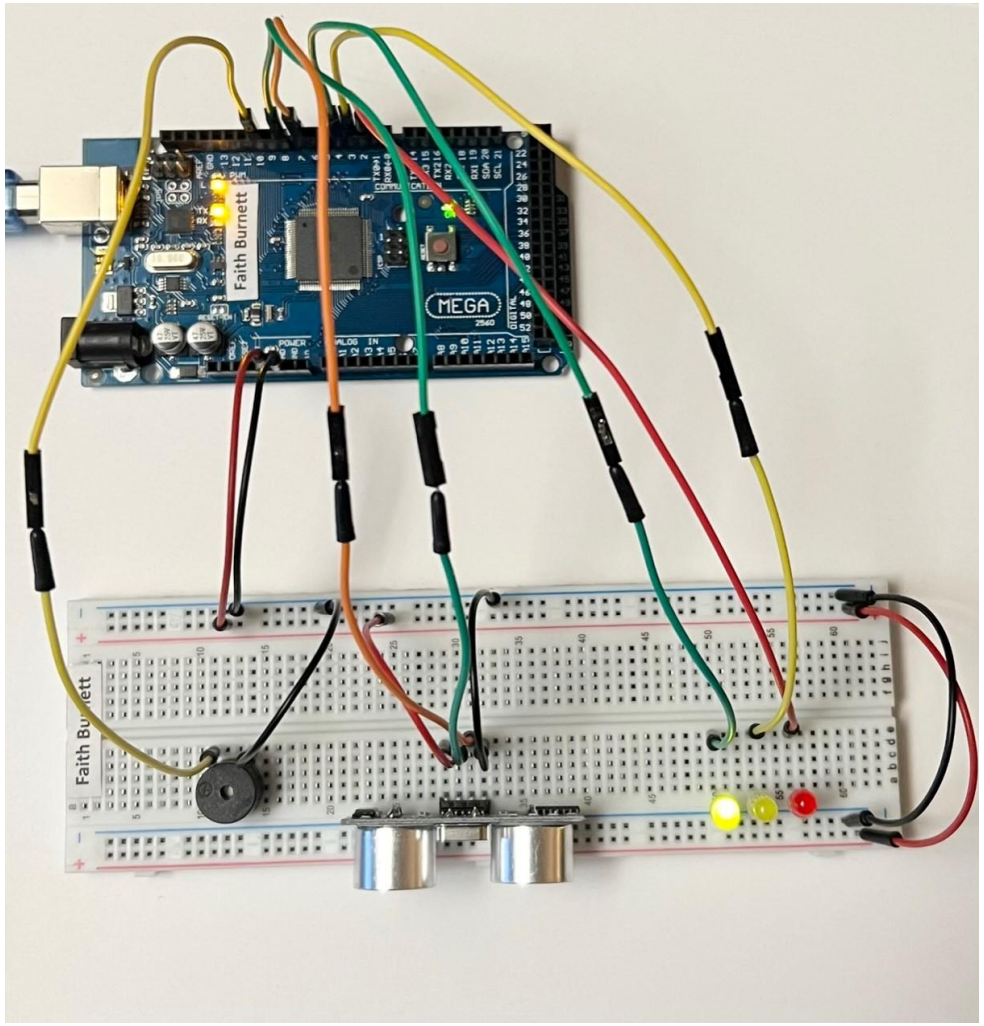


DISTANCE SENSOR

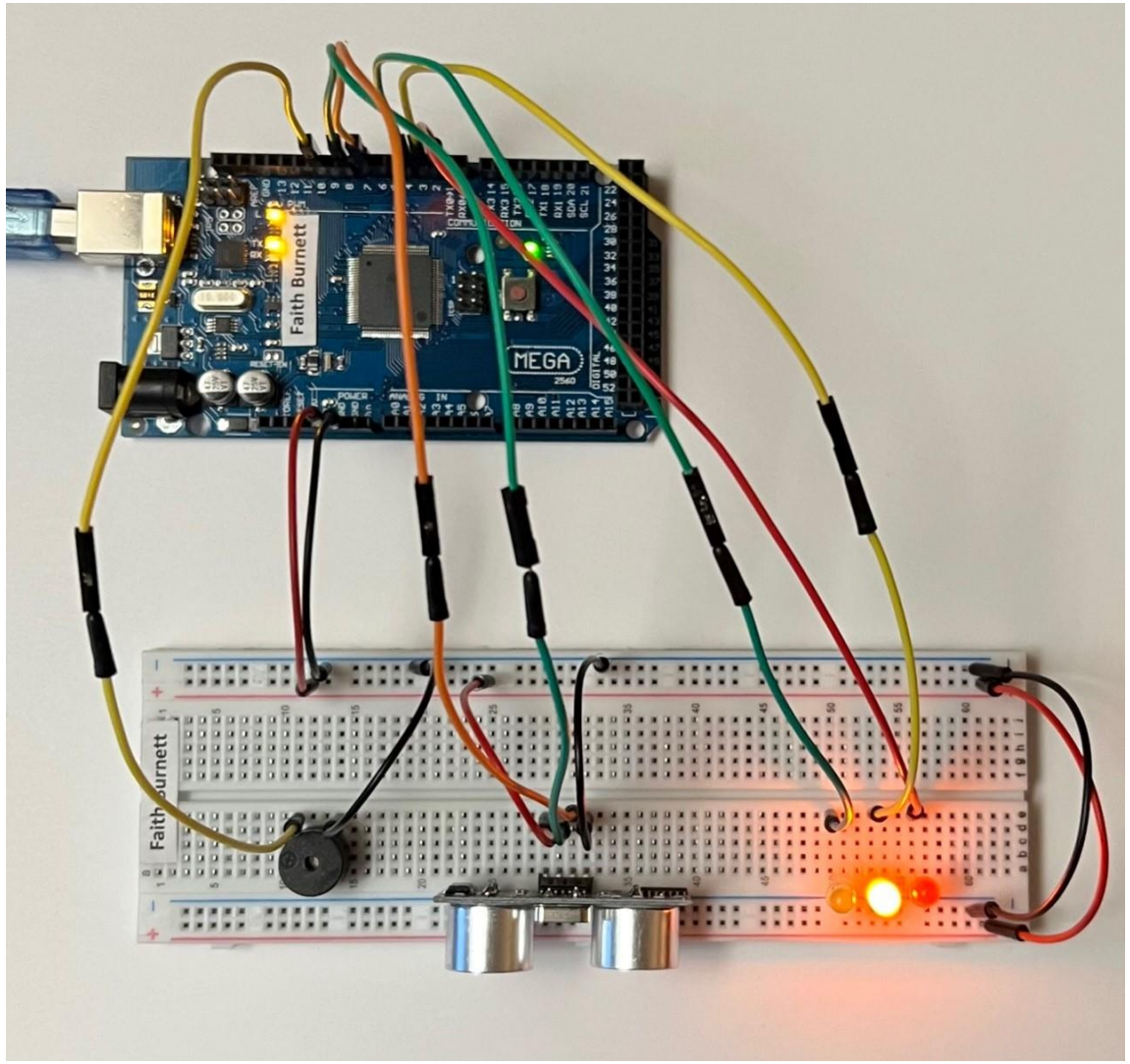
- Adding an ultrasonic sensor allows for a better method of detecting risks
- In this stage the active buzzer will fluctuate in tones based on what level of risk is detected
- The LEDs will individually light up when the corresponding level of risk arises

LOW RISK

Green LED is on meaning we are at low to no risk



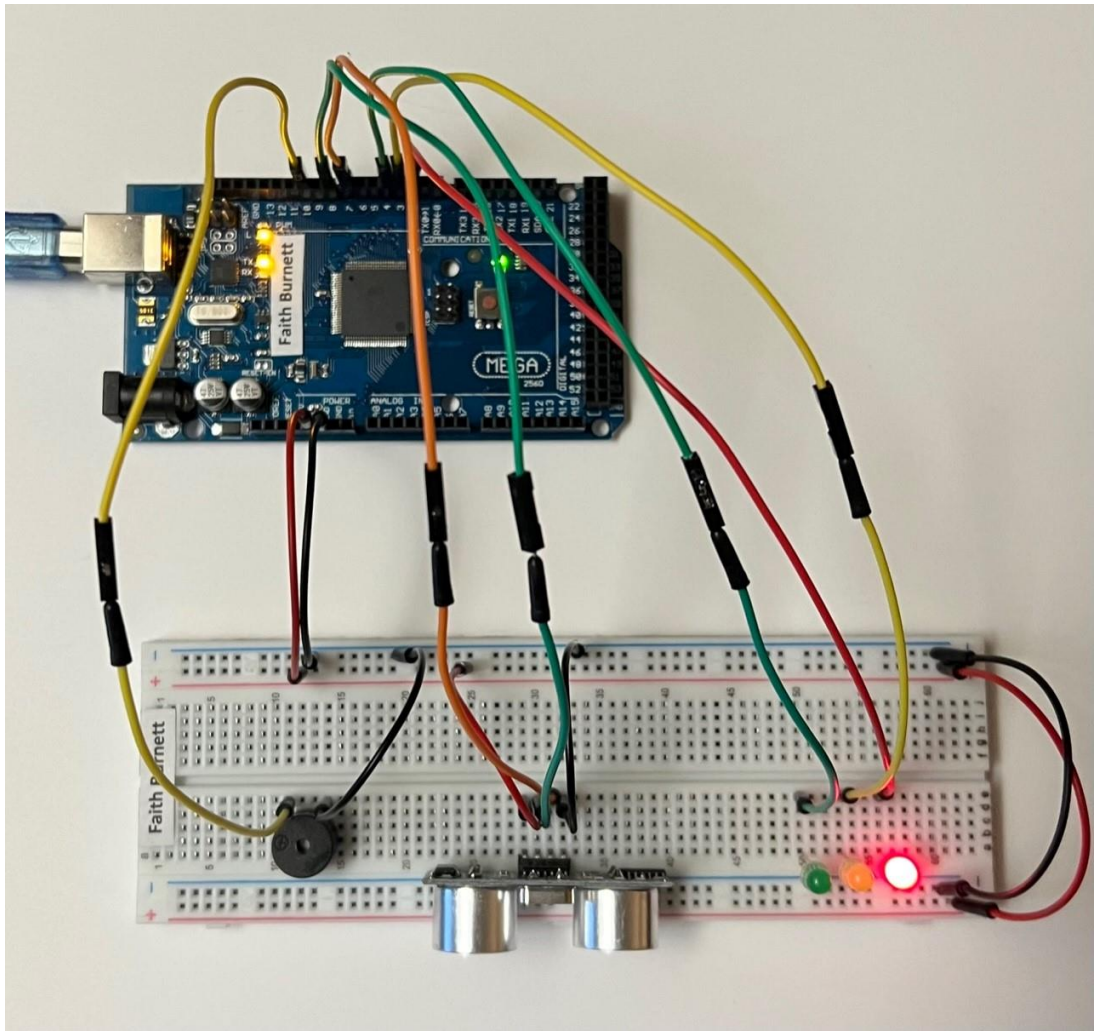
MEDIUM RISK



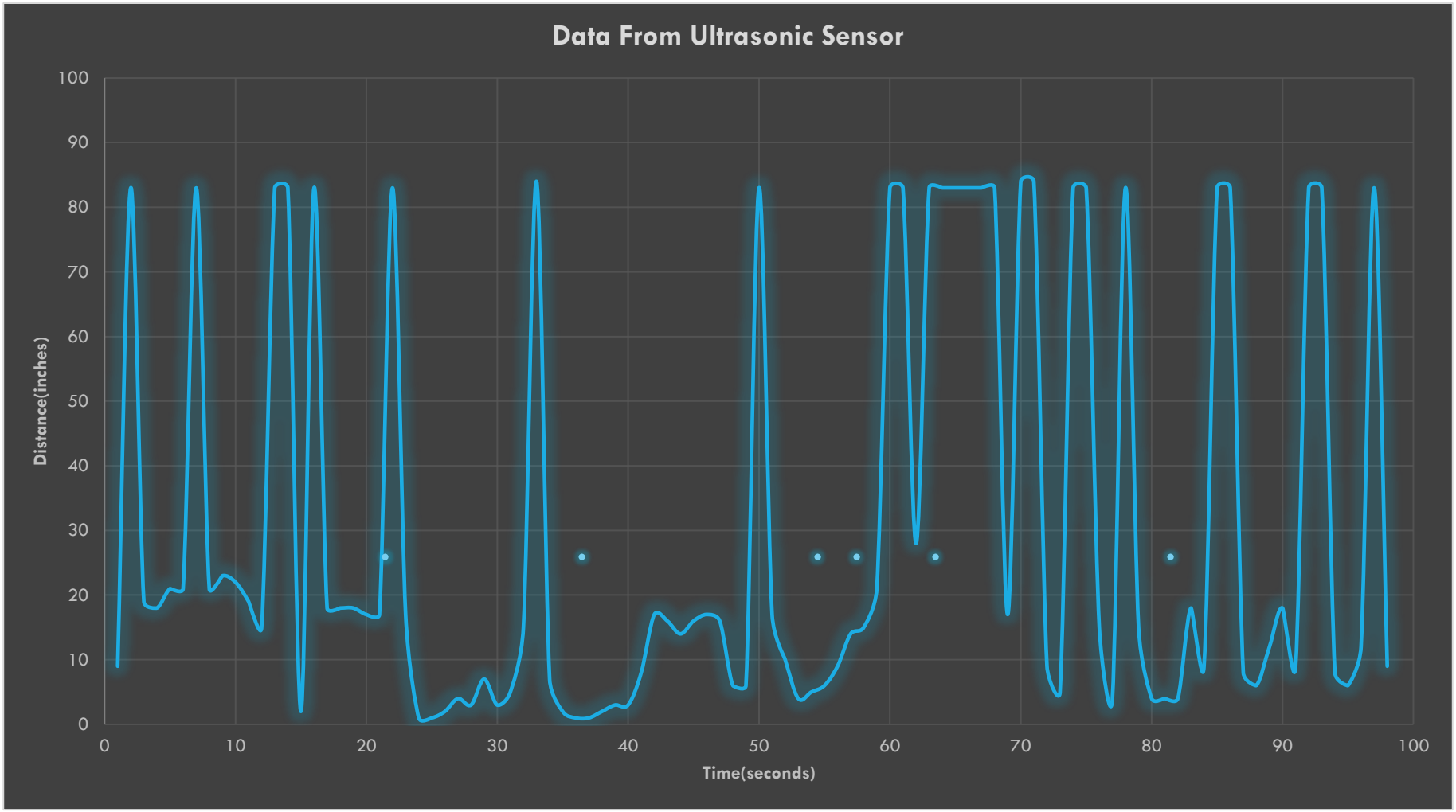
- There is a medium level of risk and the yellow LED is on
- A slightly high pitched alarm is sounding from the active buzzer

HIGH RISK

- There is a high risk detected the red LED is now on
- A high pitched alarm is sounding from the active buzzer



DATA COLLECTION



CODE FOR DISTANCE SENSOR

CEIS101_5 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_5

```
#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: Faith Burnett "); //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
  if (distance <=6){   // Alert range condition
    digitalWrite(Rled, HIGH); // Alert green LED on
    digitalWrite(Yled, LOW);
    digitalWrite(Gled, LOW);
    }
  if (distance <12 and distance > 6){  // Warning range condition
  digitalWrite(Rled, LOW);
  digitalWrite(Yled, HIGH);  // Warning yellow LED on
```

Done Saving.

```
Sketch uses 4190 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 240 bytes (2%) of dynamic memory, leaving 7952 bytes for local variables. Maximum is 8192 bytes.
```

CEIS101_5 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_5

```
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
  if (distance <=6){   // Alert range condition
    digitalWrite(Rled, HIGH); // Alert green LED on
    digitalWrite(Yled, LOW);
    digitalWrite(Gled, LOW);
    }
  if (distance <12 and distance > 6){  // Warning range condition
  digitalWrite(Rled, LOW);
  digitalWrite(Yled, HIGH);  // Warning yellow LED on
  digitalWrite(Gled, LOW);
  }

//==================== Beeping Rate Code Start ======
digitalWrite(buzzer,HIGH);
for (int i= distance; i>0; i--)
delay(10);

digitalWrite(buzzer,LOW);
for (int i= distance; i>0; i--)
delay(10);
//==================== Beeping Rate Code End =======
}
 else{ //Safe range condition
  digitalWrite(Rled, LOW);
  digitalWrite(Yled, LOW);
  digitalWrite(Gled, HIGH);  // Safe distance green LED on
  digitalWrite(buzzer, LOW);
}// end of outer IF statement

if (distance < 156) // Filter noise to show readings only less than the sensor range of 13 ft = 156 inches
  Serial.println(distance); // print distance to show in Serial Monitor

delay(100); //pause program to stabilize ultrasonic sensor readings

} //end of loop
```

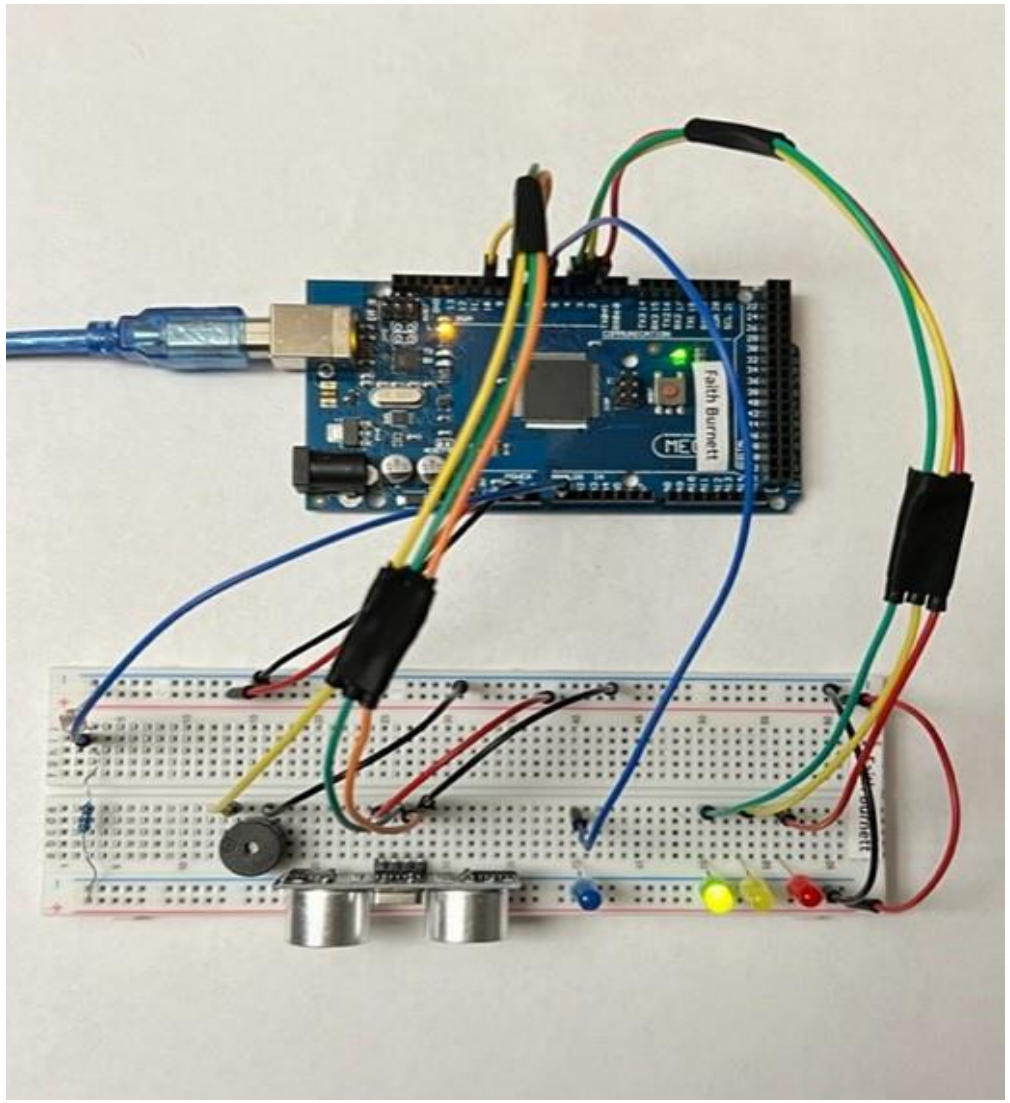
Done Saving.

Sketch uses 4190 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 240 bytes (2%) of dynamic memory, leaving 7952 bytes for local variables. Maximum is 8192 bytes.

AUTOMATED LIGHT

- The photoresistor will act as a light sensor for the home system
- A blue LED will be our indicator for when light has been detected

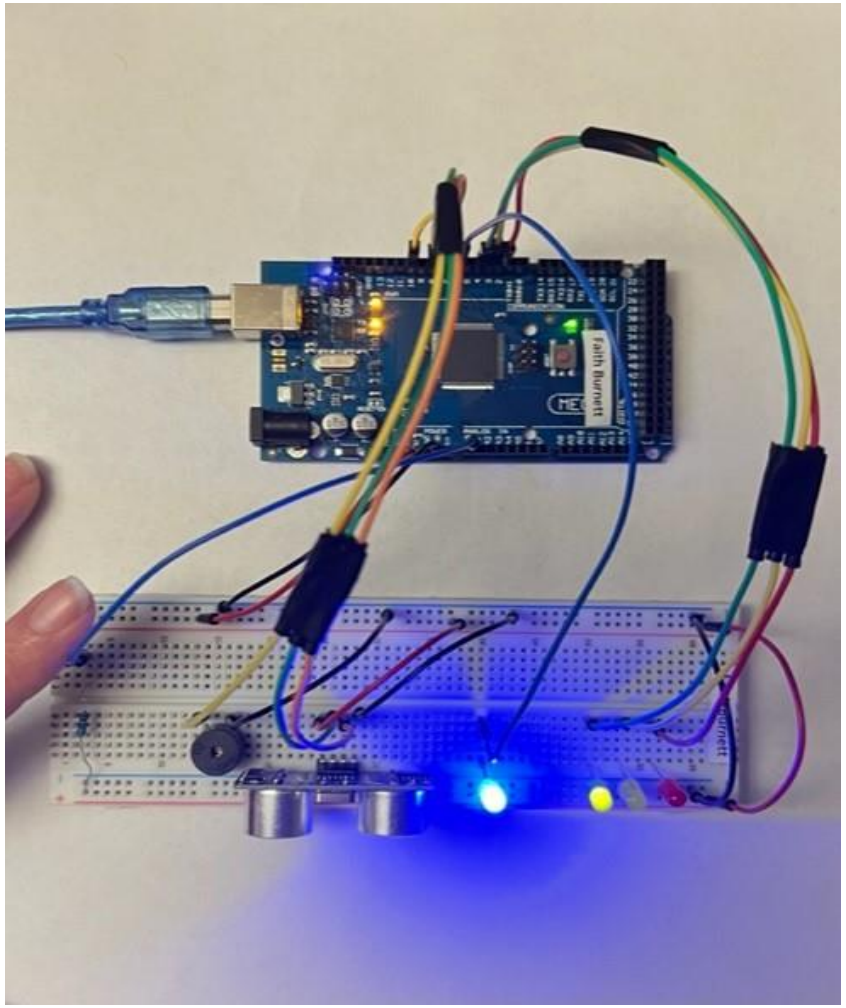
AUTOMATED LED OFF



The blue LED is off meaning the photoresistor has detected light

AUTOMATED LED ON

- By covering the photoresistor the blue LED will turn on
- This emulates an automatic light coming on when it gets dark



CODE FOR AUTOMATED LIGHT

CEIS101_6a | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_6a

```
#define trigPin 8
#define echoPin 7
#define Rled 2
#define Yled 3
#define Gled 4
#define busser 10
#define photocell A0
#define autoLight 6

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

pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(Rled, OUTPUT);
pinMode(Yled, OUTPUT);
pinMode(Gled, OUTPUT);
pinMode(busser, 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 intensity on the sensor

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

//==== Distance Sensor ===
long duration, distance, inches;

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

```
Sketch uses 4126 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 292 bytes (3%) of dynamic memory, leaving 7900 bytes for local variables. Maximum is 8192 bytes.
```

CEIS101_6a | Arduino 1.8.19 (Windows Store 1.8.57.0)

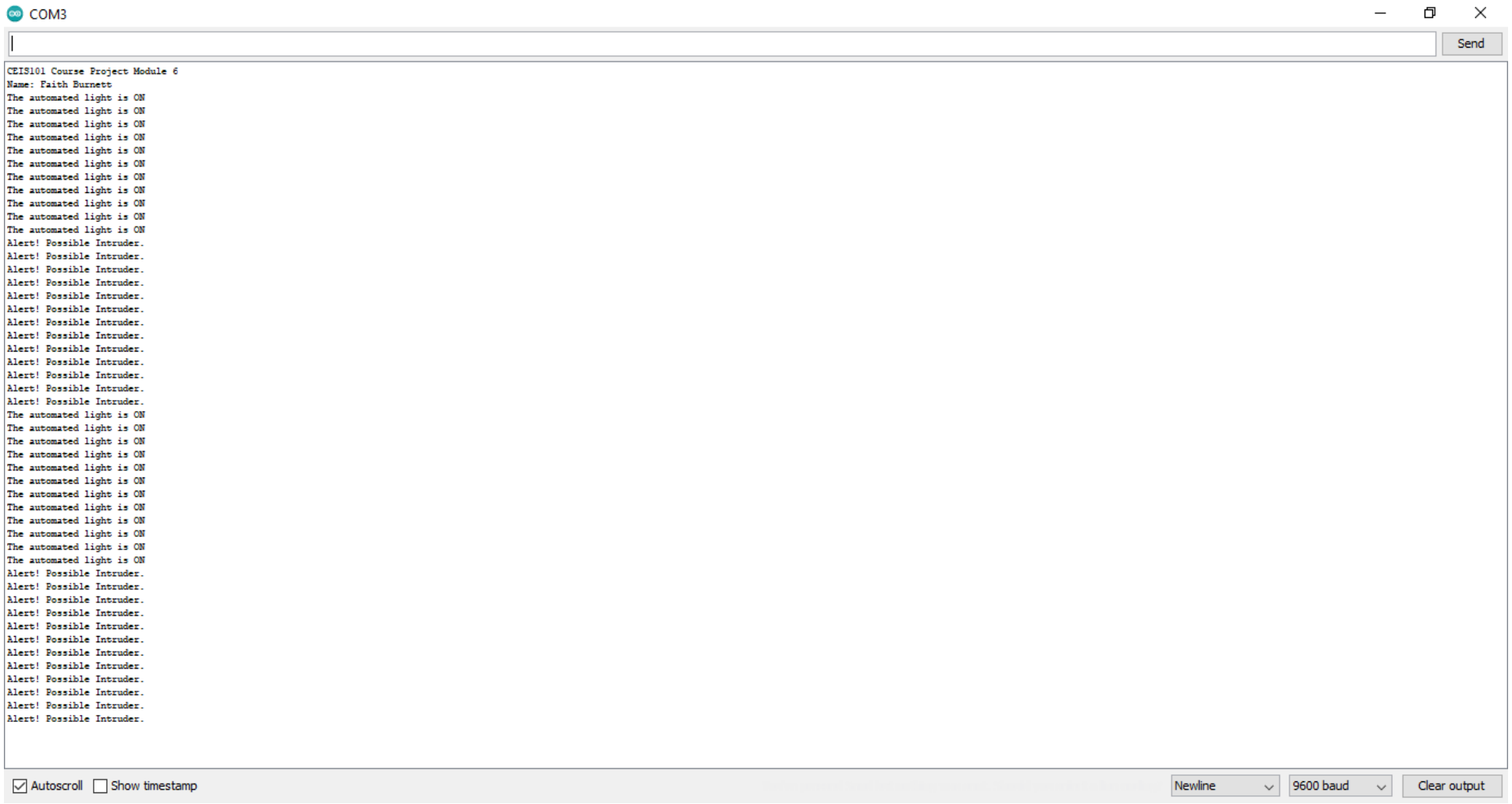
File Edit Sketch Tools Help

CEIS101_6a

```
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
  if (distance <=6){   // Alert range condition
    Serial.println("Alert! Possible Intruder.");
    digitalWrite(Rled, HIGH); // Alert green LED on
    digitalWrite(Yled, LOW);
    digitalWrite(Gled, LOW);
    }
  if (distance <12 and distance > 6){  // Warning range condition
  digitalWrite(Rled, LOW);
  digitalWrite(Yled, HIGH);  // Warning yellow LED on
  digitalWrite(Gled, LOW);
  }

//=================== Beeping Rate Code Start ======
digitalWrite(busser,HIGH);
for (int i= distance; i>0; i--)
delay(10);

digitalWrite(busser,LOW);
for (int i= distance; i>0; i--)
delay(10);
//=================== Beeping Rate Code End =======
}
 else{ //Safe range condition
  digitalWrite(Rled, LOW);
  digitalWrite(Yled, LOW);
  digitalWrite(Gled, HIGH);  // Safe distance green LED on
  digitalWrite(busser, LOW);
}// end of outer IF statement

delay(100); //pause program to stabilise ultrasonic sensor readings

} //end of loop
```

Sketch uses 4126 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 292 bytes (3%) of dynamic memory, leaving 7900 bytes for local variables. Maximum is 8192 bytes.

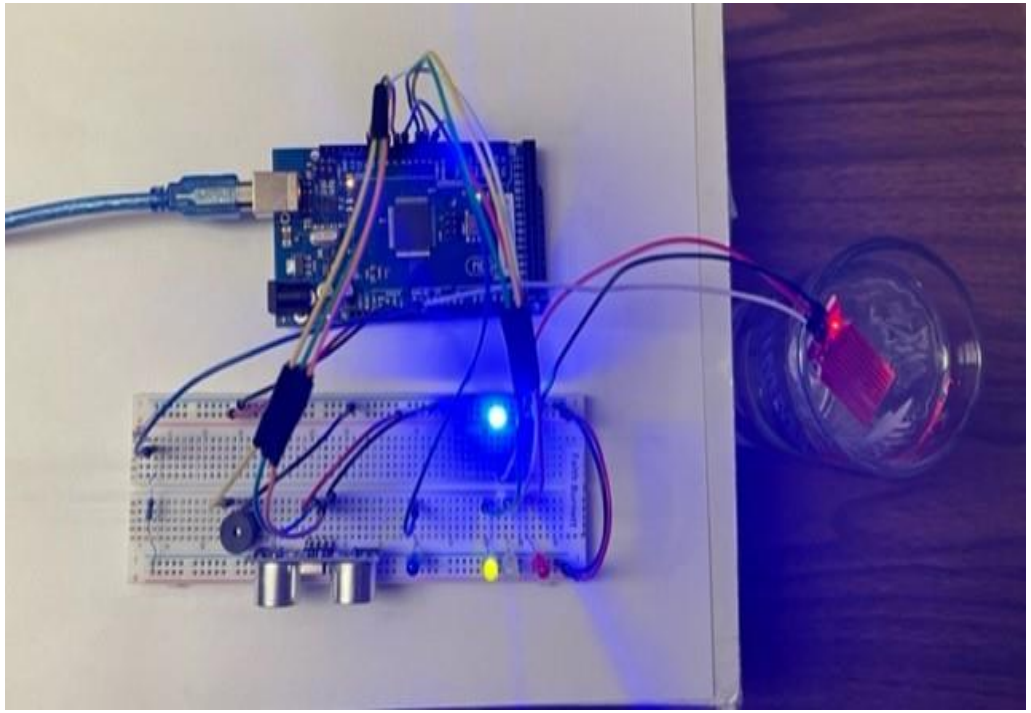
SERIAL MONITOR



WATER SENSOR & DHT-11

- Adding a water sensor will help us detect a possible flood
- The DHT-11 will give the humidity and temperature of the home

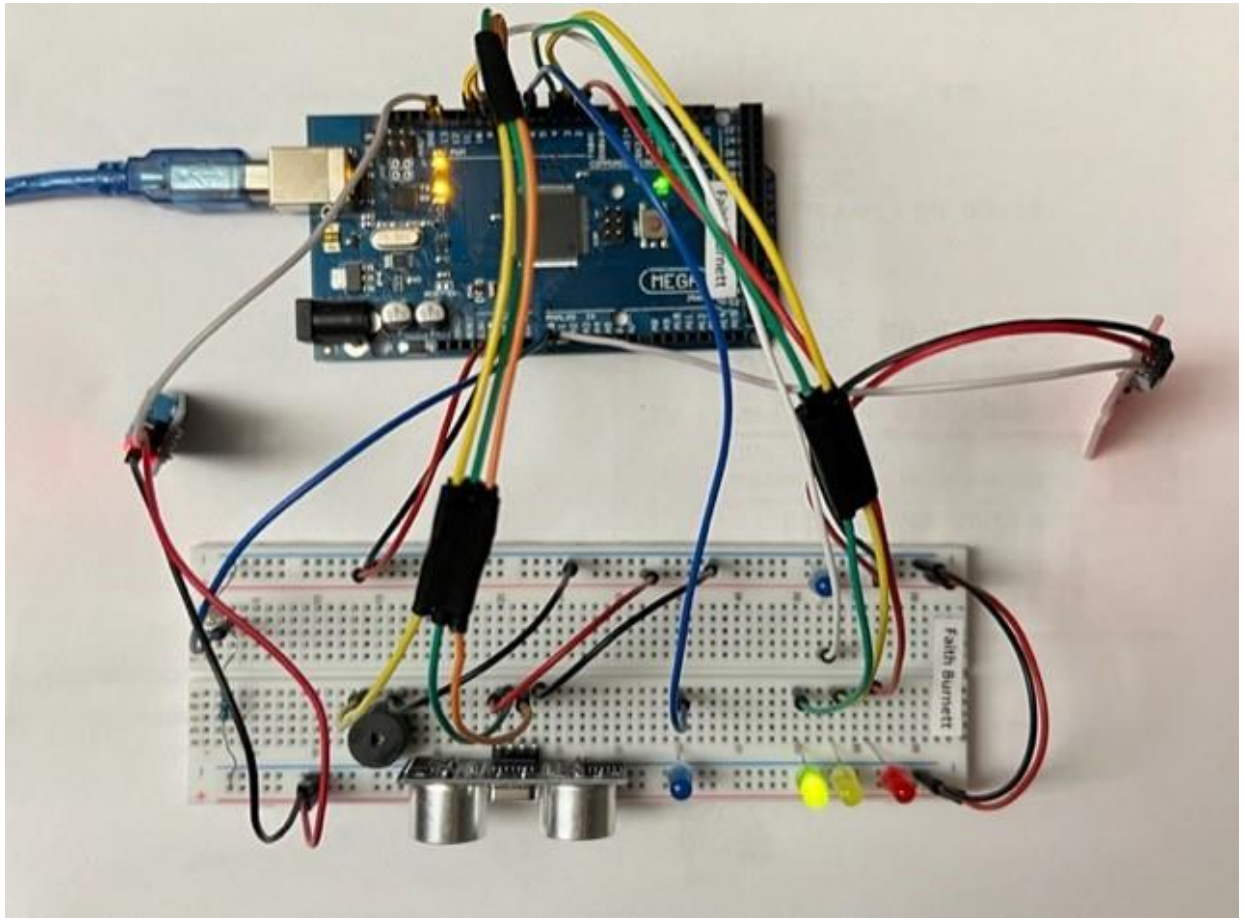
WATER SENSOR



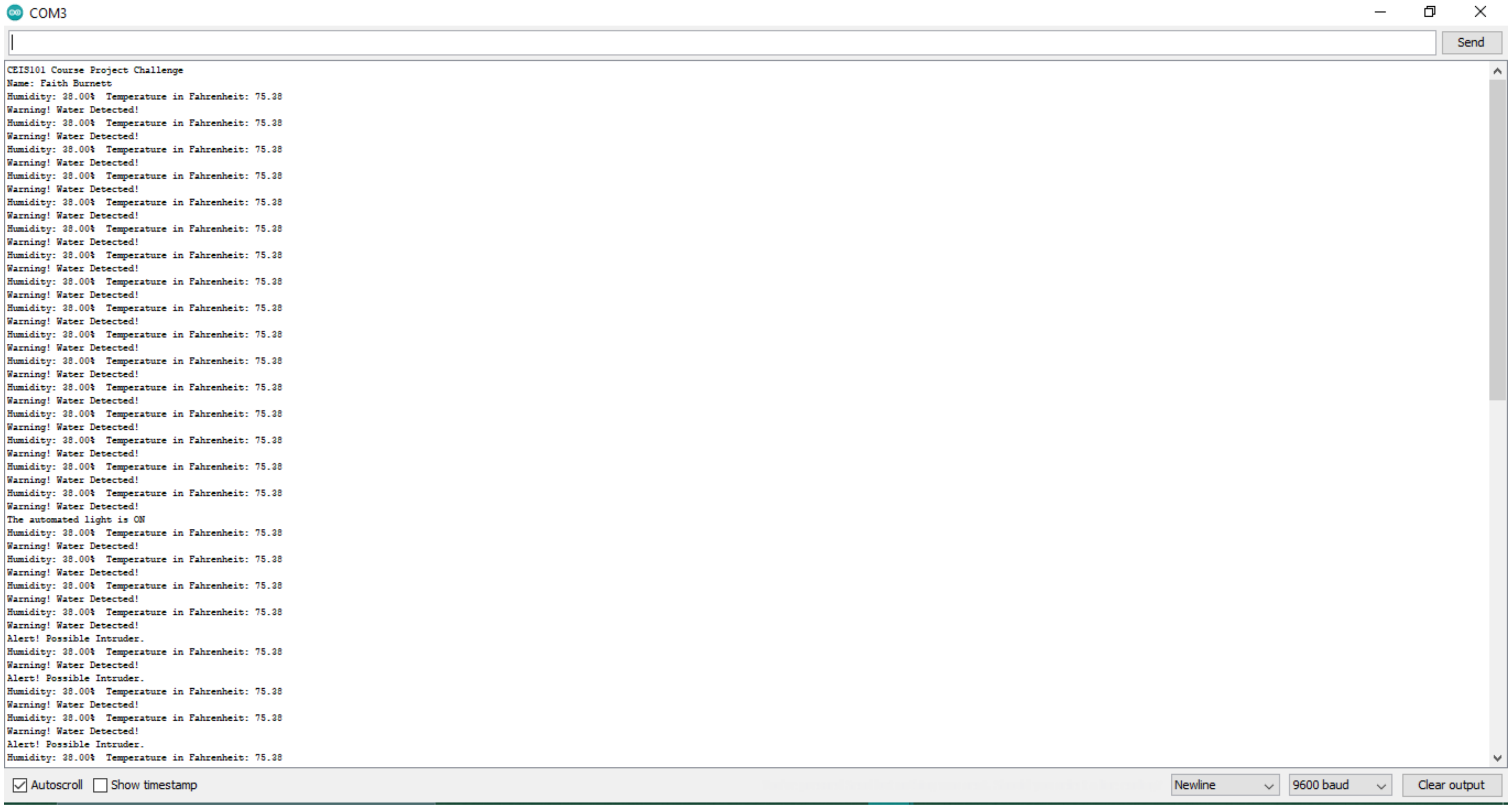
The water sensor is connected to another blue LED that will fluctuate the lights intensity based on the level of water

DHT-11

Connecting the DHT-11will light a red LED on the component itself letting me know that it is powered and ready to be used



SERIAL MONITOR



CODE FOR WATER SENSOR & DHT-11

CEIS101_6c | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_6c

```
#include "DHT.h" //include DHT 11 library
#define DHTPIN 13 // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11 // Define type of DHT sensor
DHT dht(DHTPIN, DHTTYPE); // Initialise DHT sensor

#define trigPin 8
#define echoPin 7
#define Rled 2
#define Yled 3
#define Gled 4
#define buzzer 10
#define photocell A0
#define autoLight 6
#define waterSignal A1
#define waterLight 9

void setup() {
Serial.begin(9600); // Sets the baud rate for logging
Serial.println("CEIS101 Course Project Challenge");
Serial.println("Name: Faith Burnett "); //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);
pinMode(waterLight, OUTPUT);
dht.begin();
}

void loop() {
//=== Temperature Humidity Sensor ===
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature(true);
  Serial.print(F("Humidity: "));
  Serial.print(humidity);
  Serial.print(F("%  Temperature in Fahrenheit: "));
  Serial.println(temperature);

// === Water Sensor ===
```

Done uploading.

```
Sketch uses 7250 bytes (2%) of program storage space. Maximum is 253952 bytes.
Global variables use 373 bytes (4%) of dynamic memory, leaving 7819 bytes for local variables. Maximum is 8192 bytes.
```


CEIS101_6c | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_6c

```
  Serial.println(temperature);

// === Water Sensor ===
int waterValue=analogRead(waterSignal); // Read the value of the photocell

if (waterValue > 500){ // Alert condition
analogWrite(waterLight, 255); // 100% duty cycle
Serial.println("Alert! Water Detected!");
}
else if(waterValue <= 500 and waterValue > 300){ // Warning condition
analogWrite(waterLight, 64);// 25% duty cycle
Serial.println("Warning! Water Detected!");
}
else{ // No water condition
analogWrite(waterLight, 0);// 0% duty cycle
}

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

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

//==== Distance Sensor ===
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
```

Done uploading.

```
Sketch uses 7250 bytes (2%) of program storage space. Maximum is 253952 bytes.
Global variables use 373 bytes (4%) of dynamic memory, leaving 7819 bytes for local variables. Maximum is 8192 bytes.
```


CEIS101_6c | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

CEIS101_6c

```
  Serial.println(temperature);

// === Water Sensor ===
int waterValue=analogRead(waterSignal); // Read the value of the photocell

if (waterValue > 500){ // Alert condition
analogWrite(waterLight, 255); // 100% duty cycle
Serial.println("Alert! Water Detected!");
}
else if(waterValue <= 500 and waterValue > 300){ // Warning condition
analogWrite(waterLight, 64);// 25% duty cycle
Serial.println("Warning! Water Detected!");
}
else{ // No water condition
analogWrite(waterLight, 0);// 0% duty cycle
}

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

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

//==== Distance Sensor ===
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
```

Done uploading.

```
Sketch uses 7250 bytes (2%) of program storage space. Maximum is 253952 bytes.
Global variables use 373 bytes (4%) of dynamic memory, leaving 7819 bytes for local variables. Maximum is 8192 bytes.
```