Security System | 5

Electrical and Computer Engineering 5

Enhanced Security System

Developed by Shruthi Aiyar, Zoom Chow, and Kaylana Nickels

Overview

This lab discusses the basics of security monitors, implementing Arduino C and MATLAB to create a functioning security system.

You will utilize the HC-SR04 ultrasonic sensor in order to detect motion and distance of objects, combining that with MATLAB’s webcam feature to implement both motion-detection and camera elements to the security system.

Challenge 1 will go over the basics of ultrasonic sensors, and throughout the lab you will progress to more advanced applications of these sensors using multiple software systems. By the end of the lab, you will have created a fully functioning security system with a motion detector and camera feature.

What You Will Need

Materials:

- Breadboard
- Jumper wires
- LED (generic, any color) - 3
- 330 ohm resistor
- HC-SR04 ultrasonic sensor
- Arduino MEGA 2560

Machinery:

- Laptop/computer

Software:

- Arduino IDE
- Onshape CAD
- Eagle CAD
  - Ultrasonic sensor library
- MATLAB
  - Webcam Add-on
  - Arduino support package

Challenge #1: Intro to Ultrasonic Sensors

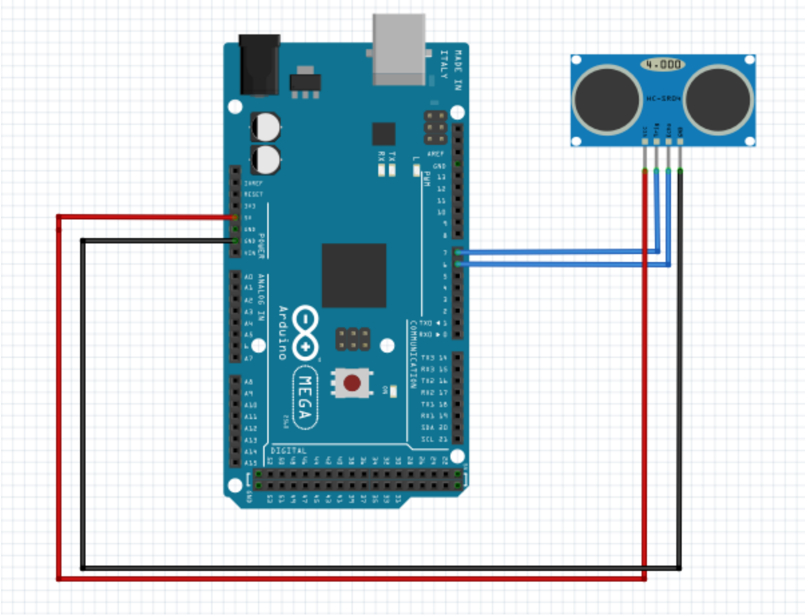
Objective

In challenge 1, we will be introducing ultrasonic sensors, using our Arduino to see how they work. Ultrasonic sensors utilize SONAR to detect the distance of objects. We will build an introductory-level sensor circuit in order to understand how sensors operate and how we can implement them into a larger-scale security system.

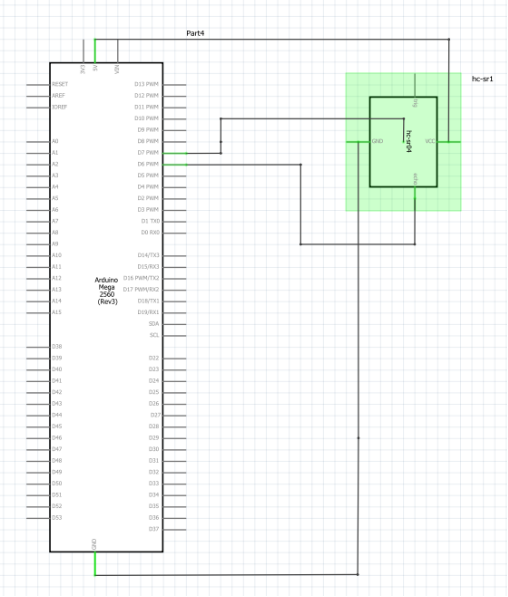
Components

- Arduino MEGA 2560
- Breadboard
- Jumper wires
    - o Red for 5V, black for GND, any color for the rest
- HC-SR04 ultrasonic sensor

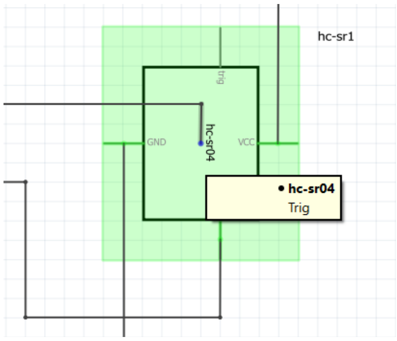
Fritzing Diagram



Schematic Diagram



- Although the connection to pin 7 may look weird, when clicking towards it connection the name trig will appear, showing that its placement is correct



Wiring

Using the fritzing and schematic diagrams above, wire your circuit.

Note: ***KEEP THIS CIRCUIT***, as we will add to it in later challenges. This is a basic ultrasonic sensor circuit with no other components, so you can understand the key elements of how this sensor works. By understanding these basics, you will be able to implement them to higher-level concepts later in this lab.

Code

```
const int trigPin = ____; // Number of ultrasonic sensor trigger pin
const int echoPin = ____; // number of ultrasonic sensor echo pin

void setup() {
   Serial.begin(9600); // Starting Serial Terminal
}

void loop() {
   long duration, inches, cm;
```

```
    pinMode(______, OUTPUT); // Output from Arduino
    digitalWrite(______, LOW);
    delayMicroseconds(2);
    digitalWrite(______, HIGH);
    delayMicroseconds(10);
    digitalWrite(______, LOW);
    pinMode(______, INPUT); // Input to Arduino
    duration = pulseIn(echoPin, HIGH);
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);
    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(100);
}

long microsecondsToInches(long microseconds) {
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
    return microseconds / 29 / 2;
}
```

Note: The trigger (TRIG) pin on the ultrasonic sensor acts by sending out a "ping." It triggers the ultrasonic sound pulses, beginning the process. The echo pin produces a pulse when the signal is received. Therefore, in the sensor, both pins are connected by how long it took for the signal to be received, which is proportional to the length of the pulse. We can use this proportional relationship to measure the distance from the sensor, in inches and centimeters.

Note: Open the serial monitor after running the code to see the distance measured.

☑ Task

- Build the circuit. This is a simple circuit with only the ultrasonic sensor connected to the Arduino using jumper wires.
- Fill in the code. Make sure you understand what the inputs and outputs are in the ultrasonic sensor.
- Upload the code onto Arduino and open the serial monitor. You should see the distance measured by the ultrasonic sensor, in inches and cm.

Introduction to Piezo Buzzer & CAD

Objective

Here we are introducing the piezo buzzer, learning what it does, and what it's purpose is within this Lab. In short terms the piezo buzzer is an electronic device that's used to produce a tone, alarm or sound. It's purpose within our program is to act as an alarm, making noise whenever motion is detected by the Ultrasonic Sensor.

Now having knowledge of the piezo buzzer, you should already have previous basic knowledge of Onshape through Pre-Lab 13.1, log into your account and let's create a CAD model for our buzzer!

Components

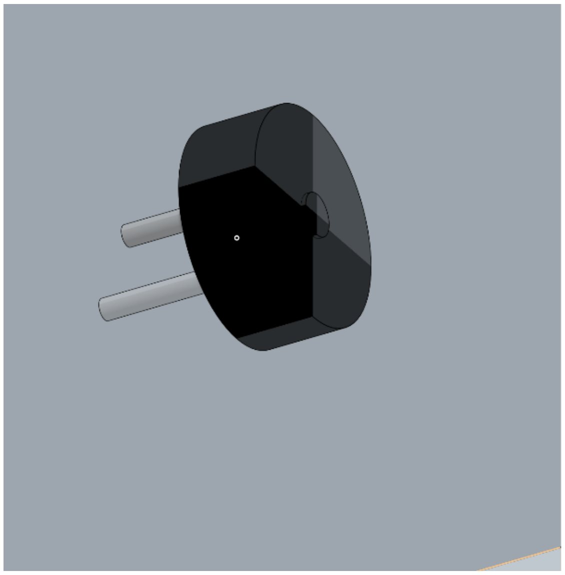
- Onshape CAD
- Piezo Buzzer (for reference)

Steps to CAD a Piezo Buzzer:

1. Open a new document and name it "Piezo Buzzer: Security Lab"
2. Create a sketch on the surface labeled as the front
3. Sketch a circle and constrain it to the center, add the diameter dimensions of the circle to be 12mm
4. Once you save the sketch, extrude it to 9.6mm and save the extrusion
5. Next you will make another sketch on the same surface as your last one and create a smaller circle in the middle, it should be centered with the center and have the diameter of 2.4 mm roughly
6. This circle will be extruded using through all in order to create the hole that you see at the top of your buzzer
7. Considering the hole doesn't truly go through your buzzer completely in real life, we will create another sketch on top of the previous having the same dimensions and placement and have it extrude a little bit just so that the hole is not completely through and open at the top
8. Now having the main body done, we will create the legs/pins of your buzzer
9. Create a new sketch, on the same surface, for your first pin, sketch a circle and dimension it to be 2.4mm in diameter
10. Then constrain your circle to be 3.6mm away from the center towards the right
11. When extruding this sketch we will extrude it to be 6.92mm in length away from the front of the piezo buzzer
12. The next leg will be sketched as another circle, being 2.4mm in diameter as well and constrained 3.6mm away from the center towards the left. This way the legs are equally away from the center, and away from each other by 7.2mm
13. Extrude this sketch to be 4.28mm in length away from the front of the piezo buzzer, this leg being the shorter one of the two as you will witness the difference in length on your buzzer in real life.
14. Our final step is to right click and apply a black color to the buzzer's body, and silver/grey to the legs/pins

Image:

Your final product should look similar to this:



☑ Task

- Complete Onshape CAD of Piezo Buzzer and Save Image

Challenge #2: Security System

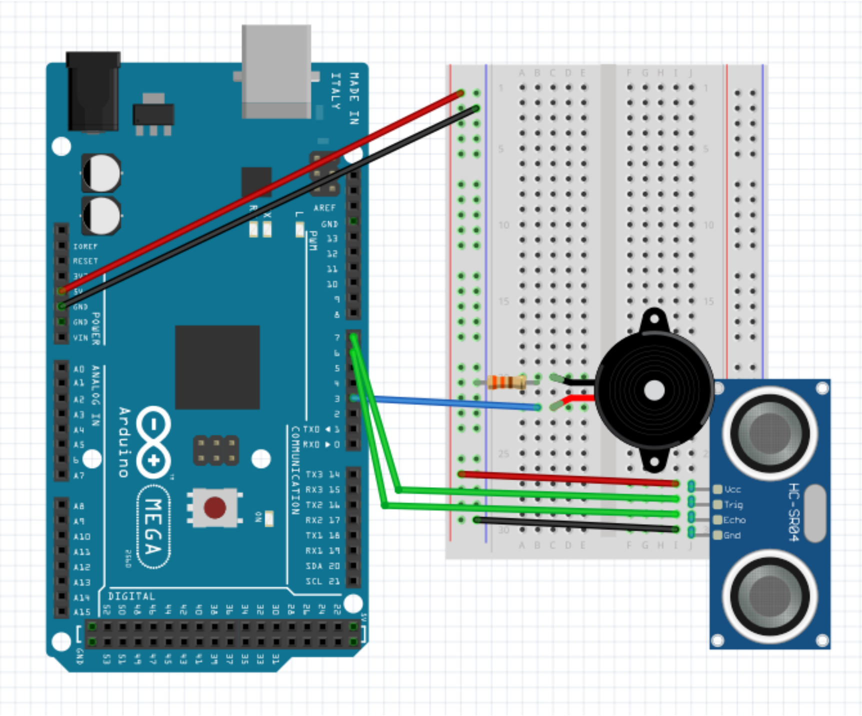
Objective

Students will use schematic and fritzing diagrams provided to build the security system circuit and run fill-in code that will read whether motion is detected. When motion is detected, the serial monitor will print "Motion detected." This will all be done in Arduino C.

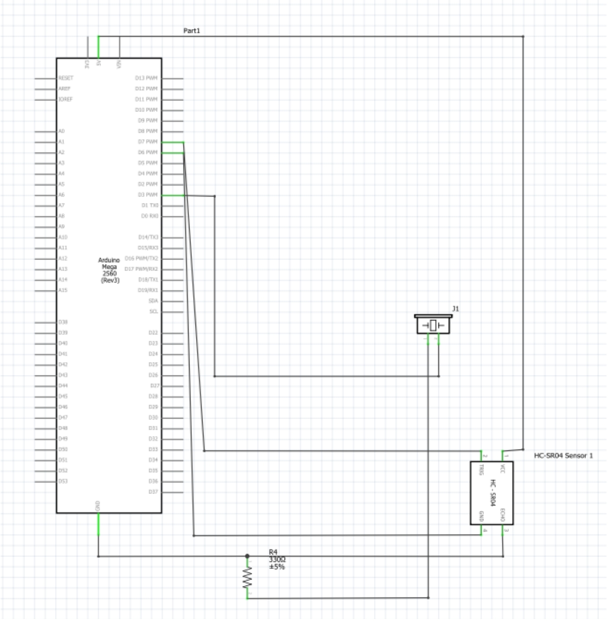
Components

- Arduino MEGA 2560
- Breadboard
- Jumper wires
    - o Red for 5V, black for GND, any color for the rest
- HC-SR04 ultrasonic sensor
- Piezo buzzer

Fritzing Diagram



Schematic Diagram



Wiring

Based on the fritzing and schematic diagrams above, wire your circuit. Essentially, you will be adding a piezo buzzer to the ultrasonic sensor circuit from Challenge 1.

Note: ***KEEP THIS CIRCUIT***, as we will add to it in later challenges. This is the security system circuit consisting of a piezo buzzer and ultrasonic sensor, so you can understand how the piezo and ultrasonic sensor work in conjunction. By understanding this, you will be able to implement them to higher-level concepts later in this lab.

Code

```
#define trigPin ____
#define echoPin ____
#define soundbuzzer ____

int sound = 500;

void setup() {
   Serial.begin (9600);
   pinMode(________, OUTPUT);
   pinMode(________, INPUT);
   pinMode(________, OUTPUT);
}

void loop() {
   long durationindigit, distanceincm;
   digitalWrite(trigPin________, LOW);
   delayMicroseconds(2);

   digitalWrite(________, HIGH);
   delayMicroseconds(10);
   digitalWrite(_______, LOW);
   durationindigit = pulseIn(echoPin, HIGH);

   distanceincm = (durationindigit/5) / 29.1;

   if (distanceincm < 20) {
   sound = 1000;
   }

   else {
      noTone(soundbuzzer);
   }

   if (distanceincm > 20 || distanceincm <= 0) {
      Serial.println("Outside the permissible range of distances");
      noTone(soundbuzzer);
   }

   else {
      Serial.print("Motion detected: ");
      Serial.print(distanceincm);
      Serial.println(" cm");
      tone(soundbuzzer, sound);
   }

   delay(300);
}
```

Task

- Add a piezo buzzer to the circuit built in Challenge 1 to introduce new elements to your security system.
- Fill in the code. Make sure you understand what the inputs and outputs are.
- Run the code and play around with the circuit! When the sensor detects objects at a certain distance specified in the code, the buzzer should sound.

Challenge #3: LED + Ultrasonic Sensor

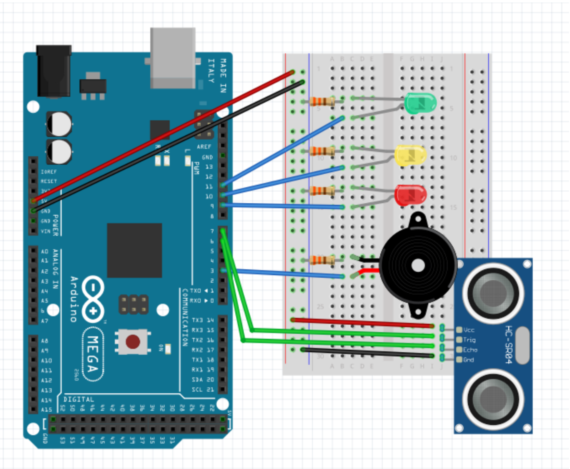
Objective

Students will incorporate LEDs that will blink when motion is detected and the serial monitor will read “Motion detected.” We will provide updated fritzing and schematic diagrams that include the LEDs for the student to build, as well as fill-in code to simulate the motion sensor with the LED component.

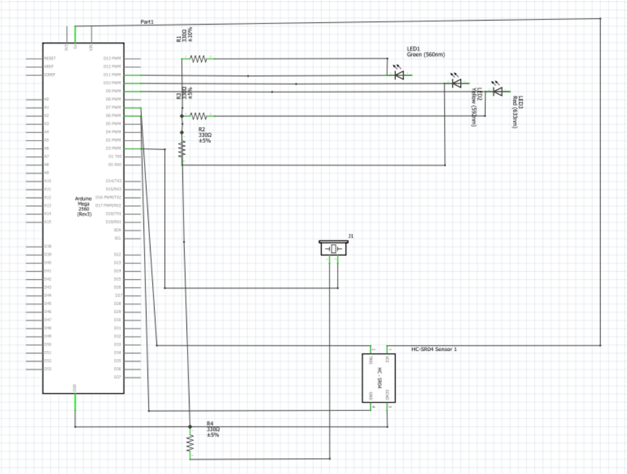
Components

- LED (Do not use RGB or IR LED)
- Arduino MEGA 2560
- Breadboard
- Jumper wires
    - Red for 5V, black for GND, any color for the rest
- HC-SR04 ultrasonic sensor
- Piezo buzzer

Fritzing Diagram



Schematic Diagram



Wiring

Based on the fritzing and schematic diagrams above, wire your circuit. Essentially, you will be adding an LED to the security system circuit (piezo buzzer + ultrasonic sensor) from Challenge 2.

Code

```
#define trigPin ___
#define echoPin ___
#define LEDlampRed ___
#define LEDlampYellow ___
#define LEDlampGreen ___
#define soundbuzzer ___

int sound = 500;

void setup() {
   Serial.begin(9600);
   pinMode(______, OUTPUT);
   pinMode(______, INPUT);
   pinMode(______, OUTPUT);
   pinMode(______, OUTPUT);
   pinMode(______, OUTPUT);
   pinMode(______, OUTPUT);
}

void loop() {
   long durationindigit, distanceincm;
   digitalWrite(______, LOW);
   delayMicroseconds(2);

   digitalWrite(______, HIGH);
   delayMicroseconds(10);
   digitalWrite(______, LOW);
   durationindigit = pulseIn(echoPin, HIGH);

   distanceincm = (durationindigit/5) / 29.1;

   if (distanceincm < 50) {
      digitalWrite(LEDlampGreen, HIGH);
   }

   else {
      digitalWrite(LEDlampGreen, LOW);
   }

   if (distanceincm < 20) {
      digitalWrite(LEDlampYellow, HIGH);
   }

   else {
      digitalWrite(LEDlampYellow, LOW);
   }
   if (distanceincm < 5) {
      digitalWrite(LEDlampRed, HIGH);
      sound = 1000;
   }

   else {
      digitalWrite(LEDlampRed, LOW);
   }

   if (distanceincm > 5 || distanceincm <= 0) {
      Serial.println("Outside the permissible range of
      distances");
      noTone(soundbuzzer);
   }

   else {
      Serial.print("Motion detected: ");
      Serial.print(distanceincm);
      Serial.println(" cm");
      tone(soundbuzzer, sound);
   }
   delay(300);
}
```

Task

- Add the LEDs to the circuit built in Challenge 2 to introduce new elements to your security system.
- Fill in the code. Make sure you understand what the inputs and outputs are.
- Run the code and play around with the circuit! When the sensor detects objects at a certain distance specified in the code, the buzzer should sound and the LEDs should turn on depending on the distance.

Challenge #4A: MATLAB Webcam

Objective

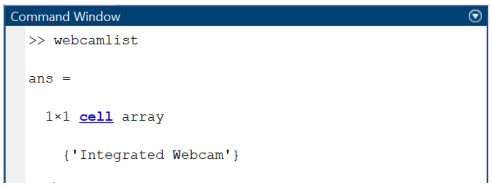
The student will use the steps provided below to set up/start the webcam feature on MATLAB using the webcam add-on. They will take a picture of themselves using their webcam camera.

Components

- MATLAB
- Webcam

Steps

1. Download the MATLAB USB Webcams Support Package if not installed already
   a. Home →Add-Ons →Get Hardware Support Packages →Install from Internet →Search for "USB Webcams" →Next →Accept terms/conditions and install
2. In the command window type, "webcamlist". It will reply back with the name(s) of the webcam installed or attached.



```
>> webcamlist

ans =

  1×1 cell array

    {'Integrated Webcam'}
```

3. Set the webcam to be an object

```
>> cam = webcam(1)
```

4. Test to see that your webcam is on and working by using the *preview()* function

```
>> preview(cam)
```

Up top on the preview window you can see the camera name. On the bottom part of the window you can see the camera name, resolution, frame rate, and the timestamp (in seconds).

You can close the preview with the function *closePreview()*.

5. Take a Picture!
   To take an image with your webcam use the function *snapshot()*. Use the function *imshow()* to display your image.

```
>> img = snapshot(cam);
>> imshow(img)
```

6. To save your image use the function *imwrite()*.

```
>> imwrite(img, 'myfirstimage.jpg');
```

7. Close the webcam using the function *delete()*.

Task

- Download the MATLAB Webcam add-on and test to see that it works.

Challenge #4B: Webcam Taking Pictures

Objective

Students will use the webcam feature to take pictures at a set interval. Fill-in code is provided below to accomplish this.

Components

- MATLAB
- Webcam

Code

```
clear all; close all; clc;
cam = webcam(______);

while ______                    % endless loop
    img = snapshot(______); % takes an image
    imshow(______)              % shows the image
    pause(10);                  % waits 10 seconds before next iteration
end
```

Task

- Create a while loop with the MATLAB webcam feature.
- Fill in the code so that a picture will be taken every ten seconds.

Challenge #5A: Enhanced Security System

Objective

Connecting sensor system to MATLAB webcam - This creates a cohesive security system with both an LED component and security camera component. The student will utilize loops in MATLAB for the webcam to take a picture when motion is detected by the circuit, and will combine this with the code from Challenge 3 to turn on the LED and have the piezo buzzer sound when motion is detected. The final result will produce both a blinking LED, piezo buzzer sound, and an image when motion is detected by the sensor. We will provide fill-in code for the webcam component, but the student will have to combine this with the previous Challenge 3 code.

Components

- MATLAB
- Webcam
- LED (Do not use RGB or IR LED)
- Arduino MEGA 2560
- Breadboard
- Jumper wires
    - Red for 5V, black for GND, any color for the rest
- HC-SR04 ultrasonic sensor

Wiring

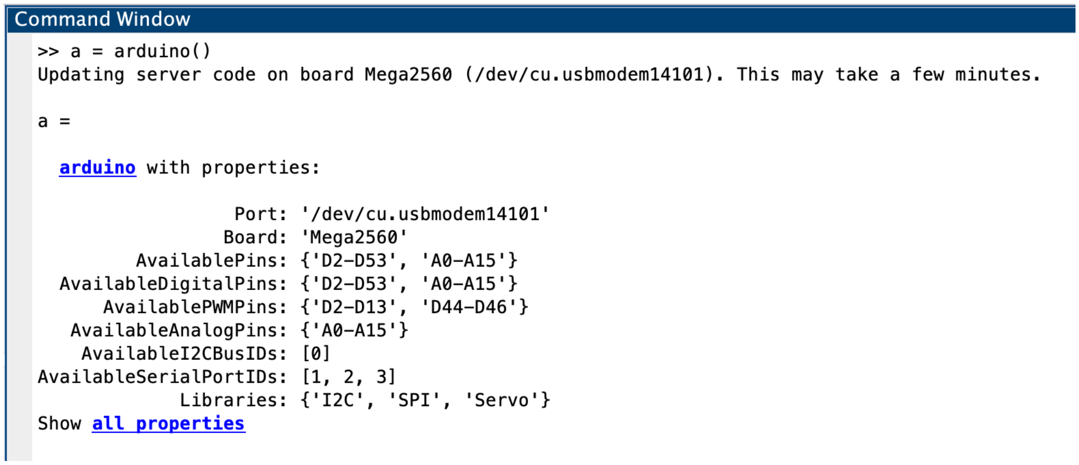
Keep the same wiring as challenge 3.

Downloading the support package

1. Download and install the MATLAB Support Package for Arduino Hardware: On MATLAB, go to Add-ons ⟶ Get add-ons ⟶ Search “Arduino” and it is the first result.
    a. When installing, make sure to check the box to include the Ultrasonic library.
2. Test the connection between MATLAB and Arduino:
    a. Type the following code into your MATLAB Command window:

    ```
    >> a = arduino()
    ```

    After a couple of minutes, the properties of your Arduino board should show up on your MATLAB command window. It should look like this:



```
>> a = arduino()
Updating server code on board Mega2560 (/dev/cu.usbmodem14101). This may take a few minutes.

a =

  arduino with properties:

                  Port: '/dev/cu.usbmodem14101'
                 Board: 'Mega2560'
         AvailablePins: {'D2-D53', 'A0-A15'}
  AvailableDigitalPins: {'D2-D53', 'A0-A15'}
      AvailablePWMPins: {'D2-D13', 'D44-D46'}
   AvailableAnalogPins: {'A0-A15'}
    AvailableI2CBusIDs: [0]
AvailableSerialPortIDs: [1, 2, 3]
             Libraries: {'I2C', 'SPI', 'Servo'}
Show all properties
```

Code

```
clear all; close all; fclose all; clc;

arduinoSerial = serial('______', 'baudrate', 9600, 'InputBufferSize',
5120); % insert Arduino serial port
fopen(arduinoSerial);
while(1)
      while(arduinoSerial.BytesAvailable)
            cam = webcam(1);
            img = snapshot(______);
            imshow(______);
            fgetl(arduinoSerial);
            return;
      end
end
```

Task

- Have Arduino work with MATLAB to take a picture when motion detected by ultrasonic sensor

Challenge #5B: PCB with Eagle CAD

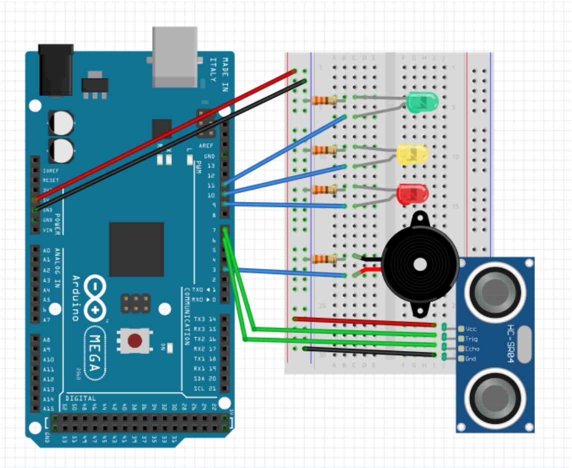
Objective

Follow the diagrams below to recreate the PCB through Eagle CAD.

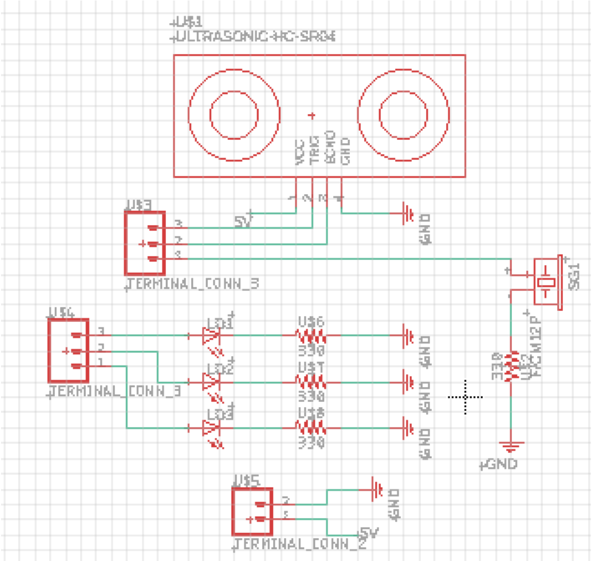
Components

- Eagle CAD
- Download zip file from this link and add files to library: https://github.com/sparkfun/SparkFun-Eagle-Libraries

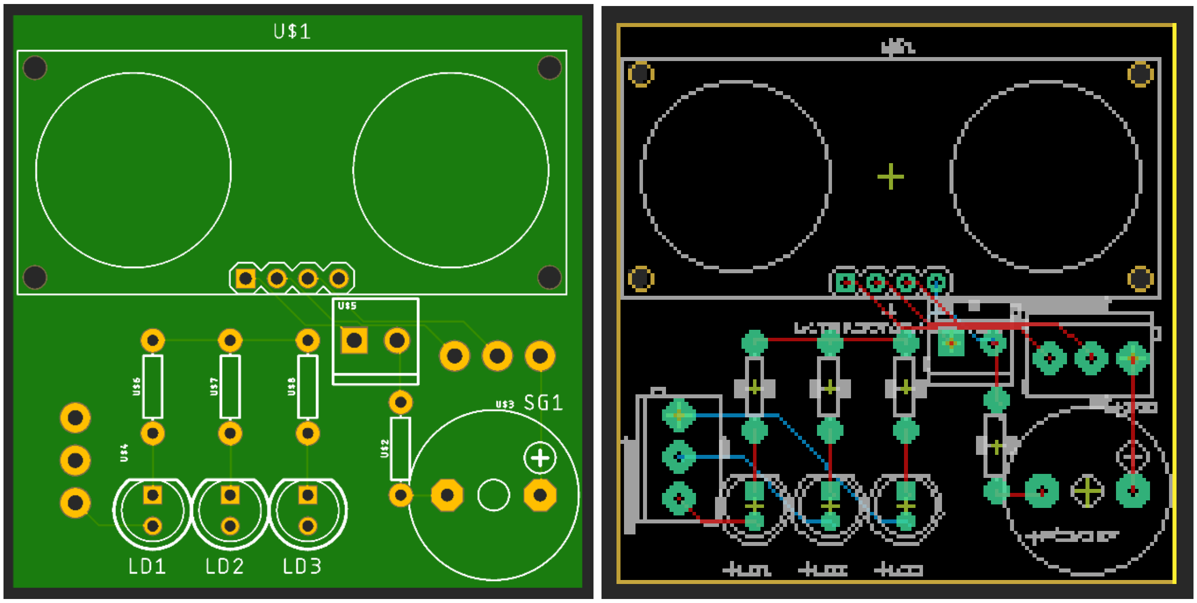
Fritzing Diagram



Schematic Diagram



PCB Diagram



Task

- Create the schematic diagram
- Check for errors and create the PCB with no errors

Appendix: References

1. https://www.mathworks.com/help/supportpkg/arduinoio/ug/getting-started-with-matlab-support-package-for-arduino-hardware.html → Getting Started with MATLAB Support Package for Arduino Hardware
2. https://create.arduino.cc/projecthub/Krepak/ultrasonic-security-system-3afe13?ref=tag&ref_id=security&offset=10 → Ultrasonic Security System
3. https://www.mathworks.com/help/supportpkg/arduinoio/ug/configure-setup-for-arduino-hardware.html → Setup and Configure Arduino Hardware
4. https://www.mathworks.com/help/supportpkg/arduinoio/ref/arduinoio.ultrasonic.html → Connection to ultrasonic sensor on Arduino hardware
5. https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm → Setup sensor
6. https://www.diymodules.org/eagle-show-object?type=dm&file=diy-modules.lbr&device=ULTRASONIC-HC-SR04 → Ultrasonic sensor library for Eagle CAD
7. ECE_5_Lab_2_Part_2 → Steps to set up MATLAB webcam
8. https://www.instructables.com/Basic-PCB-Design-in-EAGLE-Part-2-of-Proximity-Sens/ → LED used in Eagle CAD PCB