**Ultrasonic Sensor Eyes**

Joshua Gray, Briana Murdock, Michael O’Neal and Harrison Smith

CSIS 2810: Computer Architecture

Professor Hau Moy Kwan

December 2, 2020

TABLE OF CONTENTS

[ABSTRACT](#_x63tjzw1svnh) 3

[INTRODUCTION](#_u5ee66u9ul9) 4

[THE PROBLEM](#_z5w2c0gckrlb) 4

[THE DESIGN](#_gnaxbg8q1twn) 4

[THE ASSEMBLY AND PROPOSED SOLUTION](#_4jbmluna3qbk) 5

[CONCLUSION](#_mlybwdwbarjj) 6

[APPENDIX](#_bafd17vx5no1) 7

[Appendix - A: The Schematic](#_es5iuqqw31h9) 7

[Appendix - B: The Final Project Photos](#_i1qgtdn8r4re) 7

[Appendix - C: The Components](#_77cz8l1d7dcp) 9

[Appendix - D: The Code](#_6sy71yk0r6p7) 10

[References](#_r0s1gb8dr93p) 15

# ABSTRACT

The idea of living in the fictional world of Star Trek brings excitement to fans across the world. With the development of relatively new technology such as cell phones, it almost feels that our world is colliding with theirs, but there is still a ways to go. The idea of this project is to take the first step in creating androids or robots. We are creating a hardware for a computer that will be able to detect where an object is relative to it. Our project attempts to accomplish this by using ultrasonic sound waves from two HC-SR04 sensors to determine the relative location of the object. To show that it knows where the object is, eyes are drawn to two MAX7219 Dot Matrix Displays.

# INTRODUCTION

With any interactive technology, there must be some system that allows detection of surrounding objects, individuals, or obstacles. The example we gave previously is of robots and androids which need this technology to interact with the surrounding world. Similar technologies that are more applicable to our world today might be that of interactive video games. Some examples of this might be the X-box Kinect or PlayStation VR where the individual becomes the controller.

# THE PROBLEM

The problem we attempt to overcome here is that an input needs to be read in from the outside world or surrounding area. This input must then be used to calculate and accomplish some task. In our case, the task is simply to focus on a particular object in the surrounding area. In addition, there must be some type of output that indicates that the hardware ‘acknowledges’ the object in the room and recognizes where it is.

# THE DESIGN

To accomplish this task, we use a system of hardware based on Seon, or the Unexpected Maker (Seon). We use two HC-SR04 ultrasonic sensors to gather an input. This uses echolocation by emitting an ultrasonic sound wave from one side, catching it in the other, and using the time it takes to come back to determine the distance of the object. We use the combination of the two inputs to calculate which sensor detects an object closest to it.

As a visual output to indicate that the hardware knows where the object is located, we use two MAX7219 Dot Matrix Displays. If the object is calculated to be farther to the right, a pair of eyes are drawn to look to the right. The same happens if the object is calculated to be farther to the left. If no object is detected, or if there is no object within the predetermined range, the eyes will look forward.

A couple of additional things that are incorporated into the design are, 1) The eyes, based on a randomly calculated time, will blink, or do a double blink. This adds a bit of life to the machine. 2) A photoresistor is used to determine how bright it is in the room. Based upon this, the eyes will show brighter if the room is brighter, or darker in a darker room. This is so that the light from the room does not drown out the light from the LED displays.

Materials required to build the Ultrasonic Sensor Eyes include a solderless breadboard, 13 jumper wires, 3 female-to-male dupont wire, Arduino Nano R3 microcontroller, Photo resistors, 2 HC-SR04 ultrasonic sensors, 2 Maxim Integrated 8x8 LED Matrix MAX7219, and a cardboard box enclosure. The enclosure does not need to be a cardboard box. (For a complete schematic of the hardware and the code used, see **Appendix A** and **Appendix D** respectively. For an assembled complete project, see **Appendix B** *Figure 2.4*. For images of the components, see **Appendix C.**)

# THE ASSEMBLY AND PROPOSED SOLUTION

We were able to assemble a device that was able to draw eyes in the direction of the detected object. One issue we came across was that the sensors were not picking up objects as well as we were hoping. After doing some research, we learned that it has to do with the nature of the ultrasonic sensors and how they work.

There are two problems with this sensor. It needs a pretty solid object to bounce sound off, thus walls are good but cloth absorbs the sound wave and the echo never bounces back. The second is you can encounter ghost echoes when the sound wave strikes a corner and creates an echo for an object that is not really there (Namphibian).

After understanding more about how these sensors work and sound itself, two conclusions were devised that would create an improvement for a future project. One is simply to find a sensor that works on more types of surfaces. The other idea comes from Janet Raloff and Bethany Brookshire in their article How the ears work. “The outer ear’s shape helps to collect sound and direct it inside the head toward the middle and inner ears” (Brookshire and Raloff). This leads to the idea of building a sort of cone or cup based on the human ear for the sensors to pick up sound from a larger area as opposed to only directly in front of the sensors. (For photos of the final project, see **Appendix B** figures 2.1, 2.2 and 2.3**.**)

# CONCLUSION

Our initial plan was enjoyable to create and was effective in accomplishing the intent of the project. Our hardware is able to detect whether an object rests to the right of the device, to the left of the device, or if it’s not detected at all. Upon conducting additional research after the completion of the project, we see that it may be more beneficial to use a different type of sensor to detect more types of objects, as well as to detect an object at an angle to the sensors as opposed to only directly in front of them. By all means, though this project was a success and though it’s a very modest contribution to the technological world, it provided us with a glimpse of what may be in store.

# APPENDIX

## Appendix - A: The Schematic

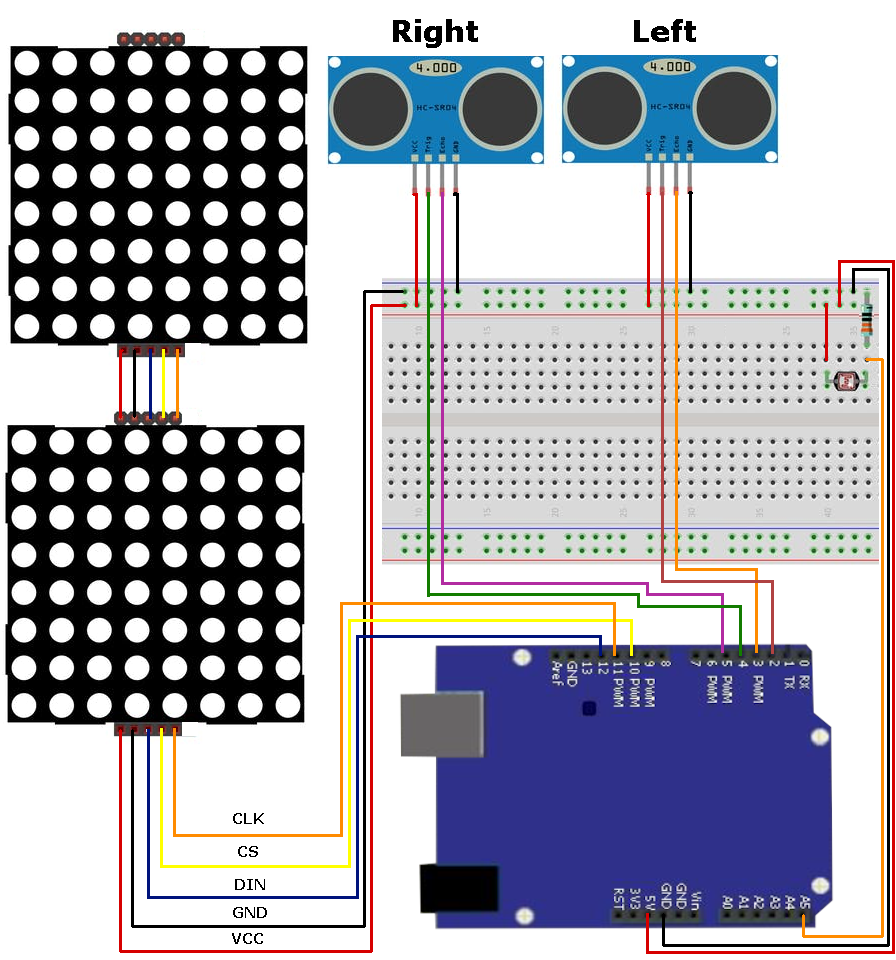


Figure 1.1 - The Schematic

## Appendix - B: The Final Project Photos

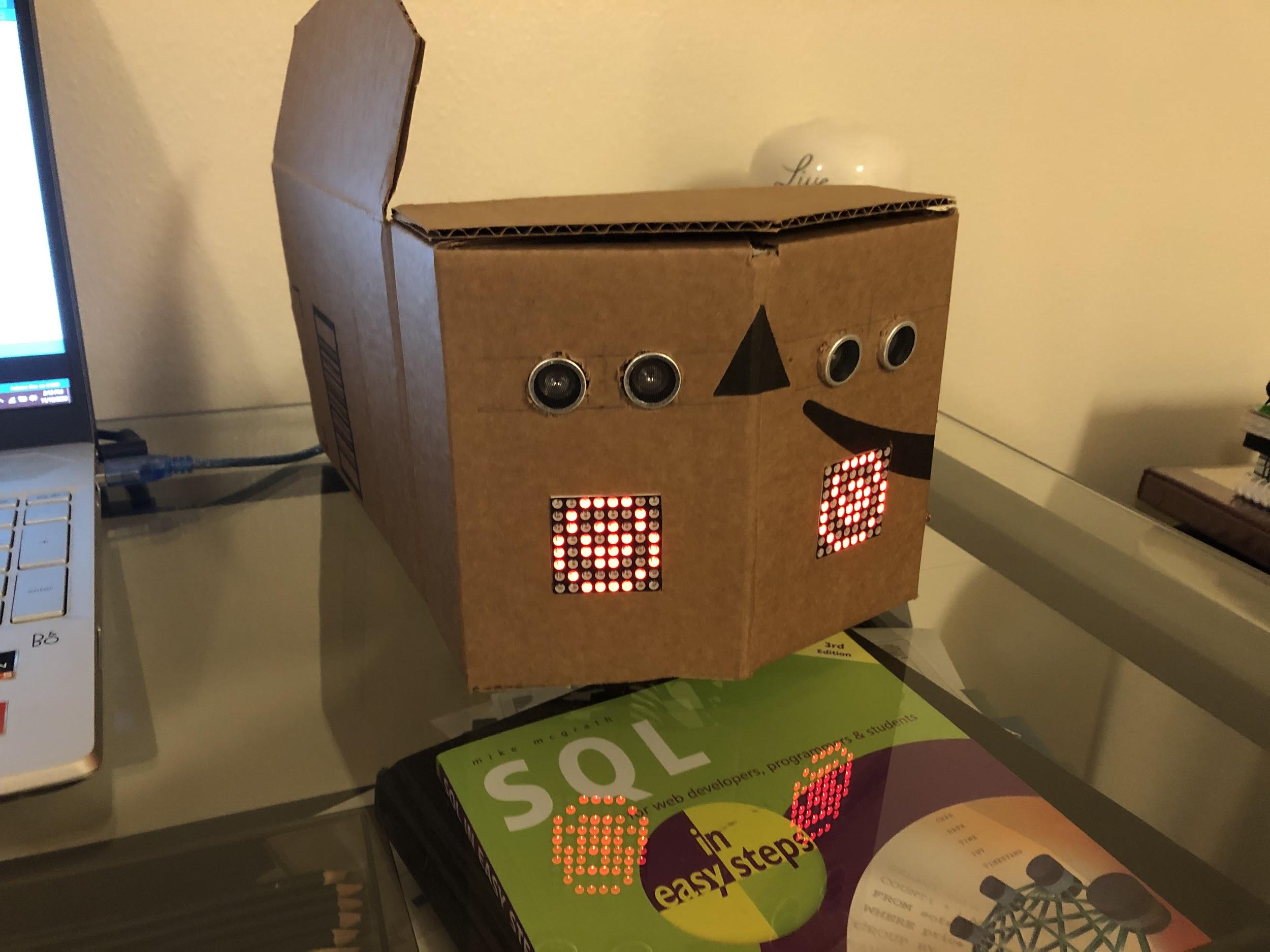


Figure 2.1 - the eyes in this figure are looking straight ahead when no object is detected.



Figure 2.2 - the eyes look to the left when an object is detected to the left.

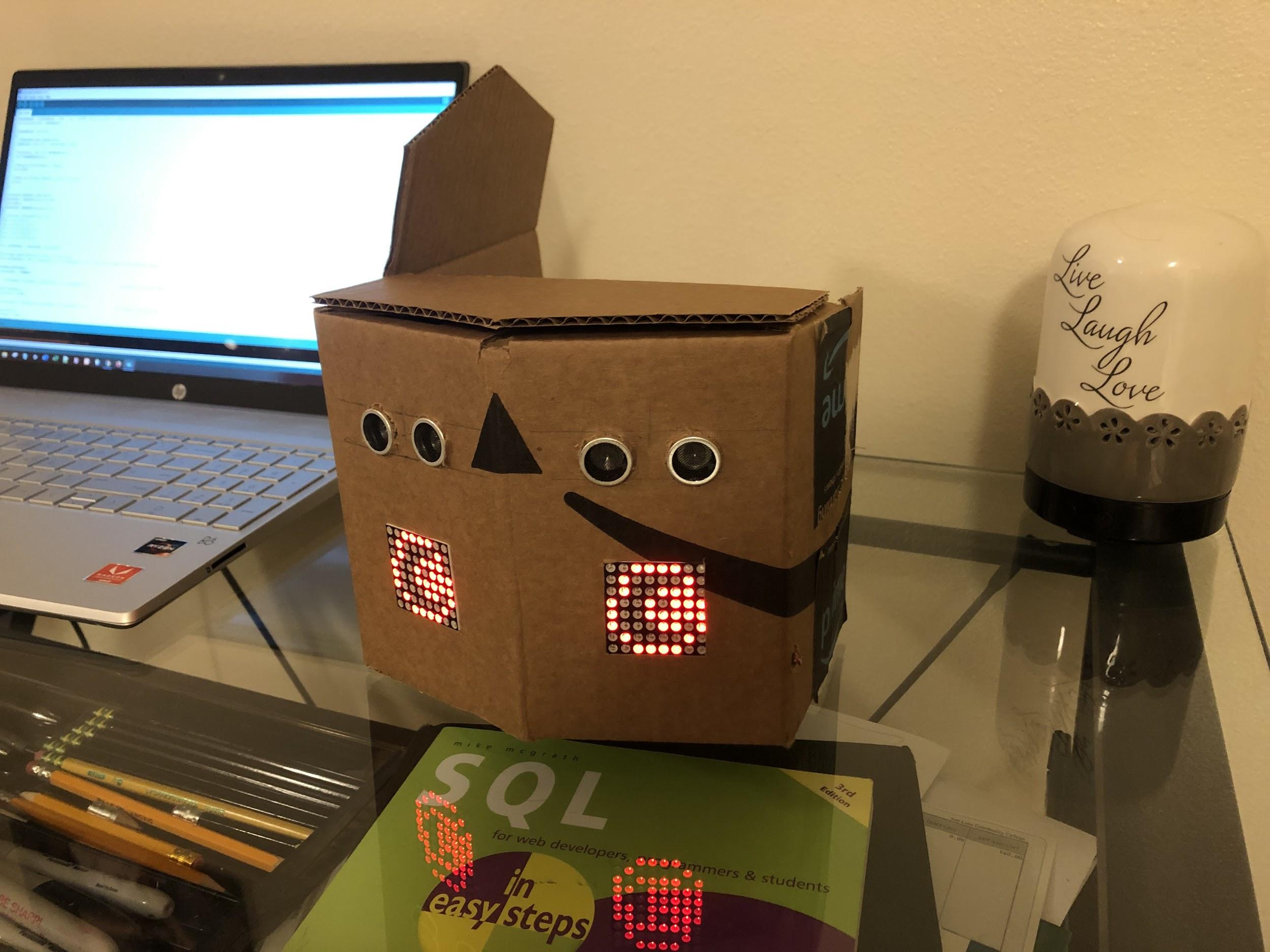


Figure 2.3 - the eyes look to the right when an object is detected to the right.



Figure 2.4 - the internal hardware and components.

## Appendix - C: The Components



Figure 3.1 - HC-SR04 ultrasonic sensors



Figure 3.2 - Photoresistor

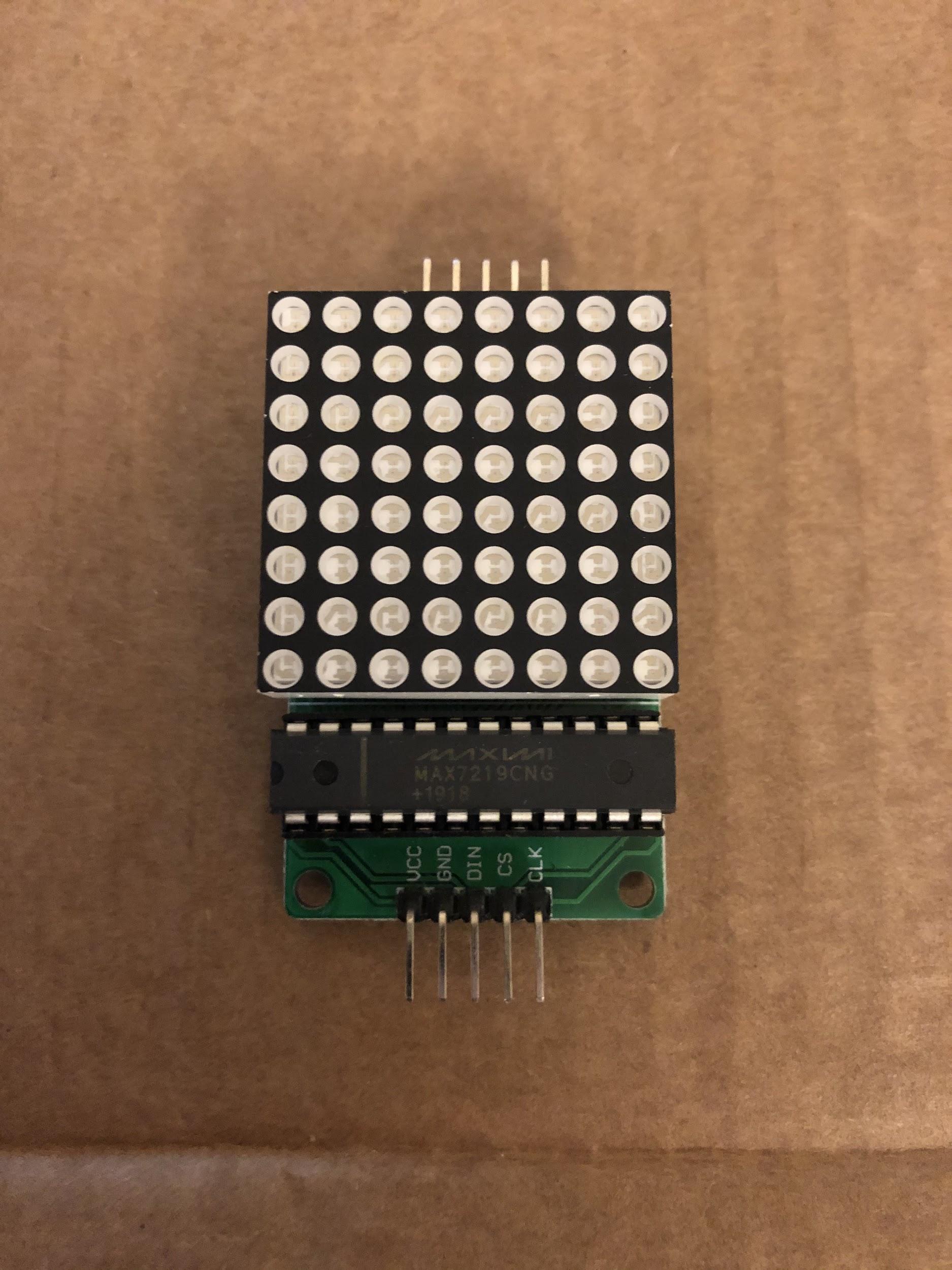


Figure 3.3 - Maxim Integrated 8x8 LED Matrix MAX7219

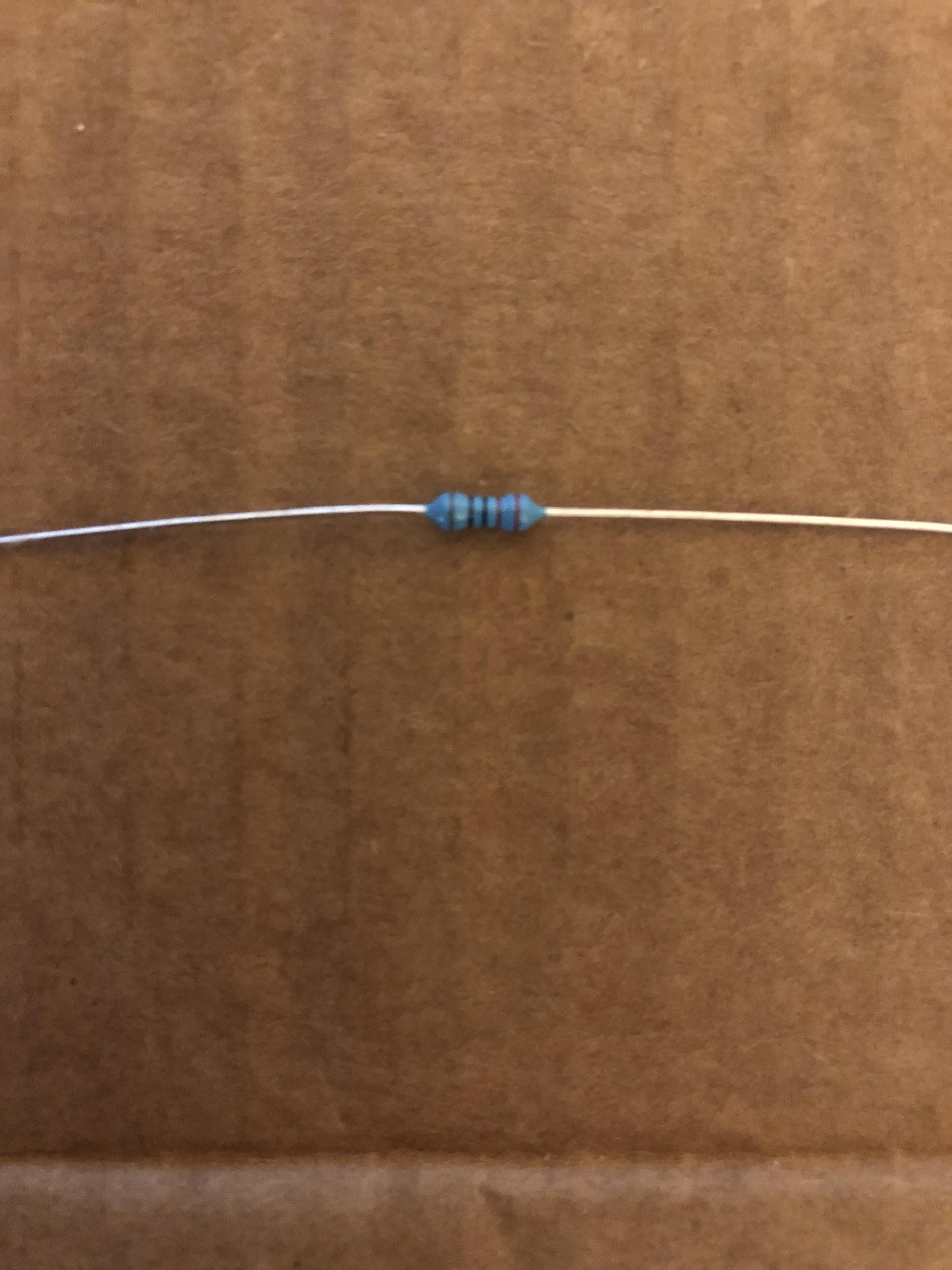


FIgure 3.4 - 330 Ohm resistor

## Appendix - D: The Code

//Libraries to include

#include "LedControl.h" //Used to control LED displays

#include "NewPing.h" //Used to control Ultrasonic Sensors

//Variables

//Constants

#define LIGHT A5 //Sets light reading pin

#define tL 2 //Trigger pin on Ultrasonic Sensor 1

#define eL 3 //Echo pin on Ultrasonic Sensor 1

#define tR 4 //Trigger pin on Ultrasonic Sensor 2

#define eR 5 //Echo pin on Ultrasonic Sensor 2

#define maxDist 300 //The max distance for the ultrasonic pulse

#define FORWARD 0 //Constant value to pass to set display to forward

#define RIGHT 1 //Constant value to pass to set display to right

#define LEFT 2 //Constant value to pass to set display to left

//Classes

LedControl lc = LedControl(12, 11, 10, 1); //Set up LED display and pins LedControl(dataPin,clockPin,csPin,numDevices)

NewPing sensorR(tR, eR, maxDist); // Initialise Right Sensor

NewPing sensorL(tL, eL, maxDist); // Initialise Left Sensor

//Primitive

unsigned long delaytime = 1000; //Delay time between displays

float lightAmount = 1; //Light amount read from photoresistor

float nextBlink = millis() + 1000; //Randomly sets when the next blink will be

long durationL, durationR; //Sets variables for durations

int distanceL, distanceR; //Sets variables for distances

int currentState = -1; //Defines what direction the eyes are looking currently

//Arrays: Eye states

byte eyeForward[8]=

{

B00111000,

B01000110,

B10011001,

B10110101,

B10111101,

B10011001,

B01000110,

B00111000

}; //eye\_forward

byte eyeLeft[8]=

{

B00111000,

B01011110,

B10110101,

B10111101,

B10011001,

B10000001,

B01000110,

B00111000

}; //eye right

byte eyeRight[8]=

{

B00111000,

B01000110,

B10000001,

B10011001,

B10110101,

B10111101,

B01011110,

B00111000

}; //eye left

byte eyeBlink[8]=

{

B00111000,

B01101100,

B11011110,

B11011110,

B11011110,

B11011110,

B01101100,

B00111000

}; //eye blink

//Main setup

void setup() {

//Starts LED Display

lc.shutdown(0,false);

//Set the pin modes for Left Sensor

pinMode(tL, OUTPUT);

pinMode(eL, INPUT);

//Set the pin modes for Right Sensor

pinMode(tR, OUTPUT);

pinMode(eR, INPUT);

//Set each trigger pin on the Ultrasonic sensors to start at LOW

digitalWrite(tL, LOW);

digitalWrite(tR, LOW);

//Set brightness on displays

lc.setIntensity(0,0);

pinMode(LIGHT, INPUT);

//Clear the display

lc.clearDisplay(0);

//Sets state of displays to forward

currentState = FORWARD;

}

//Main program

void loop(){

//Reads Light amount

lightAmount = analogRead(LIGHT);

lightAmount = (lightAmount / 255) \* 12;

lc.setIntensity(0, lightAmount);

if(nextBlink < millis())

{

//Generates next random blink

nextBlink = millis() + random(2000, 10000);

if(random(1, 13) <= 6) showEyeBlink();

else showEyeDblBlink();

//Delay to allow display to appear

delay(250);

//Break out of this iteration of the loop early

return;

}

//Captures distance from sensors

distanceL = sensorL.ping\_median(5);

delay(250);

distanceR = sensorR.ping\_median(5);

float difference = (distanceR - distanceL);

//Calculates which direction to draw to the display

//If too far away, or not detected: Forward

if(abs(difference) < 250 || (distanceL == 0 && distanceR == 0))

{

drawEye(eyeForward);

currentState = FORWARD;

}

else if(distanceR < distanceL && distanceL > 0 && distanceR != 0) //If closer to the right: Right

{

drawEye(eyeRight);

currentState = RIGHT;

}

else if(distanceL < distanceR && distanceR > 0 && distanceL != 0) //If closer to the left: Left

{

drawEye(eyeLeft);

currentState = LEFT;

}

delay(250);

}

void showEyeBlink()

{

drawEye(eyeBlink);

delay(125);

if(currentState == FORWARD) drawEye(eyeForward);

else if(currentState == LEFT) drawEye(eyeLeft);

else if(currentState == RIGHT) drawEye(eyeRight);

}

void showEyeDblBlink()

{

drawEye(eyeBlink);

delay(75);

if(currentState == FORWARD) drawEye(eyeForward);

else if(currentState == LEFT) drawEye(eyeLeft);

else if(currentState == RIGHT) drawEye(eyeRight);

delay(75);

showEyeBlink();

}

//Draws Eye based on which array is passed

void drawEye(byte state[]){

lc.setRow(0,0,state[0]);

lc.setRow(0,1,state[1]);

lc.setRow(0,2,state[2]);

lc.setRow(0,3,state[3]);

lc.setRow(0,4,state[4]);

lc.setRow(0,5,state[5]);

lc.setRow(0,6,state[6]);

lc.setRow(0,7,state[7]);

}

# References

Brookshire, Bethany & Raloff, Janet. “Explainer: How the Ears Work.” *Science News for Students,* <https://www.sciencenewsforstudents.org/article/explainer-how-ears-work#:~:text=The%20outer%20ear's%20shape%20helps,tube%20called%20the%20ear%20canal>. Accessed 24 November 2020.

Seon. Ultrasonic Eyes. *Project hub.* <https://create.arduino.cc/projecthub/unexpectedmaker/ultrasoniceyes-b9fd38?ref=tag&ref_id=microcontroller&offset=3> Accessed 24 November 2020.

Namphibian. Distance Sensor - Not Working on Fabric Surface. *Stack Exchange.* <https://arduino.stackexchange.com/questions/15943/distance-sensor-not-working-on-fabric-surface> Accessed 24 November 2020.