Mid Term Project

CISC 071

By

Jeremy Reuwer

Date: 4/10/2019

Purpose

* To submit your Mid Term Project (40 Points)
* Use the skills you learnt to build a project

Rubric

* Correctness: 10 Points. Project should work as you have defined it
* Complexity: 10 Points. Use at least :: 2 sensors, 1 resistor, 1 capacitor, 1 actuator (Speaker, Motor), and 2 LEDs
* Creativity: 10 Points. Project has some new idea (10) or is it a copy (0)
* Code: 10 Points. The program has to be cleanly written with clear comments and descriptions of various steps. The code should have a for loop; if statement and print statement.

Project Description

Arduino ‘Cat Sensor’

This is a little project that was inspired by a late night in my room and wanting an ability to know when my cat would enter and exit to feed her. The project combines an Ultrasonic Sensor for fine detection and a regular microphone as a general alert indicator. Its basic operation consists of placing the device with the ultrasonic sensor facing the desired entryway or hallway – think of it as trip mine based on sound reading alone.

When you first power the device, the buzzer will make A diminished triad 5 times to notify of initialization. The Blue LED operates in correlation to the microphone as a “general presence indicator” meaning that the louder the environment is, the brighter the LED becomes, letting the user know that something might possibly be afoot! I used it at night when I would find myself playing PC games in darkness, with the door open. Generally would keep a level based on the monitor of less than ‘50’ and I would know my cat had walked in because suddenly it would jump to ‘80’ and above, causing the indicator to be noticeably brighter.

The only drawback I experienced in the design phase of the project was that the only form of mobile power I had was not enough to hand the nice and proper 9v connector we received with the kit, and I could not refine the design enough with the components either. I did jury rig 4 AA batteries with the jumper wires provide (not ideal) and it did work sufficiently.

The second half of the project consists of using the Ultrasonic sensor in conjunction with an actuator, the analog buzzer, and 3 different colored LEDs to indicate distance visually when no volume is used on the buzzer. Green indicating 25 cm, and yellow and red indicating values closer to 0 cm. In conjunction the buzzer works on the same concept, the closer the object the higher the pitch becomes, written in the actual frequency desired. You might notice that I did not connect the capacitor to a pin on the Arduino board and write code to control volume based on the potentiometer alone, but rather I took an old trick I learned from my guitar playing days, were a capacitor is used with a pot in the signal chain to either control tone or volume of the guitars electrical signal. When the potentiometer is turned the signal is being absorbed or filtered by the rather large capacitor and the buzzer goes mute.

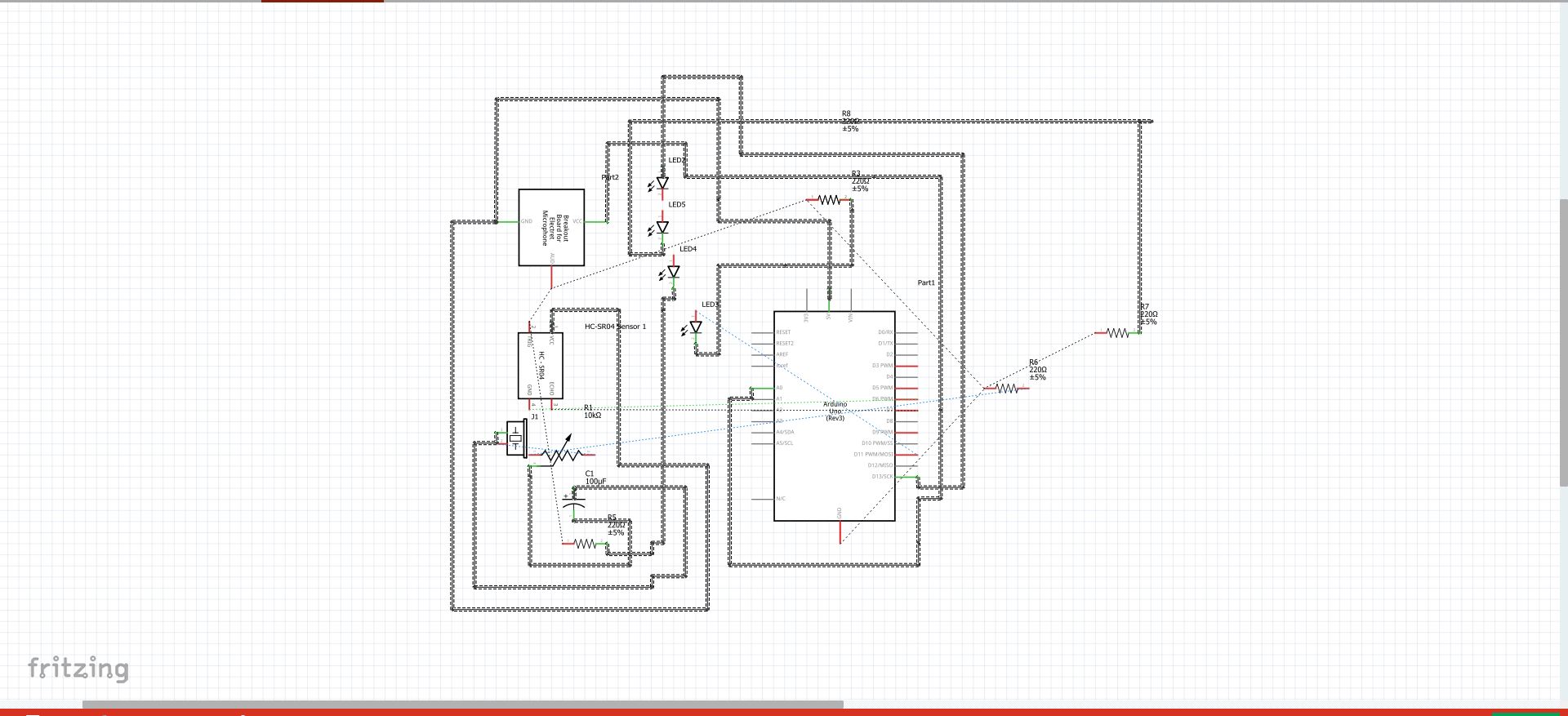
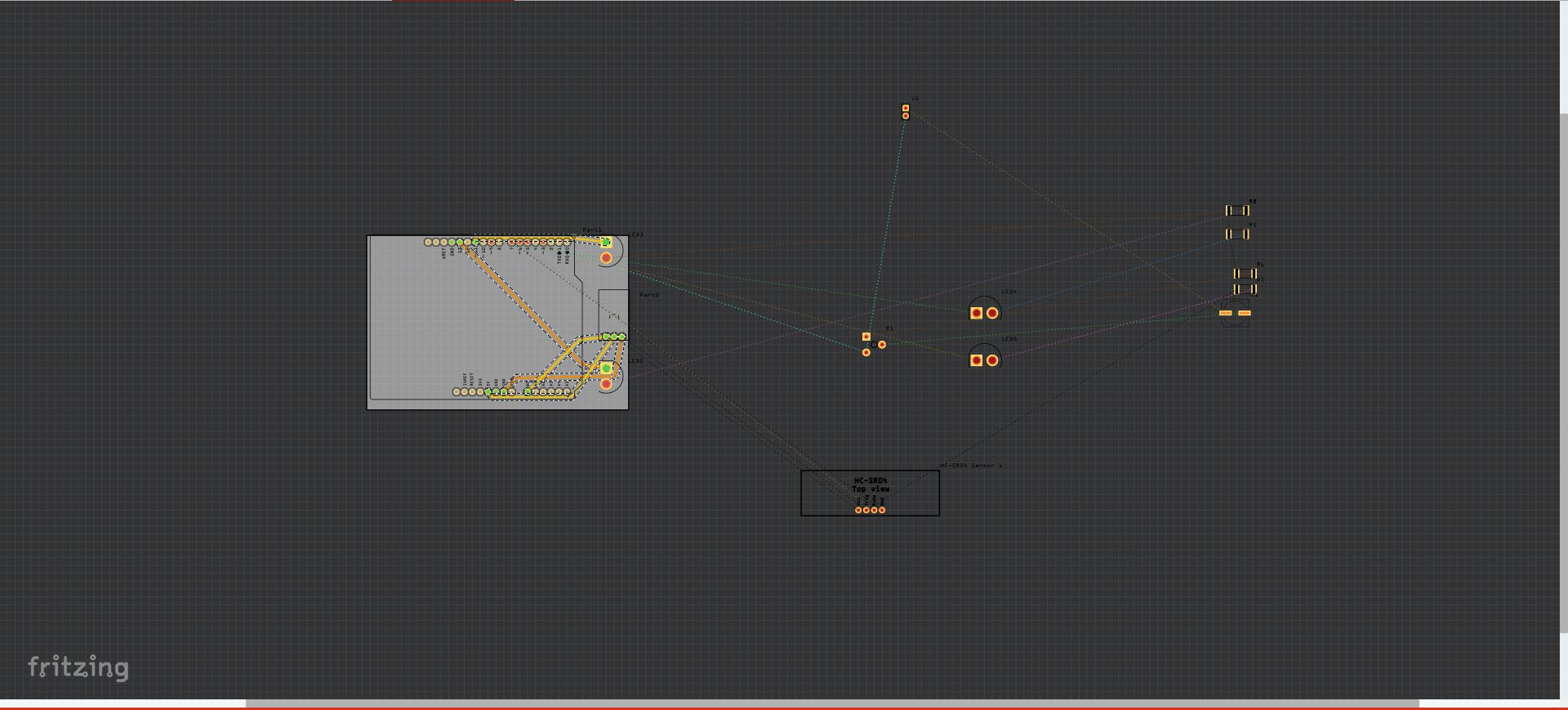
Needless to say, my cat gets a treat every time the Arduino makes lights and noise, she actually eventually recognized the reward and formed a light treat habit.

Components

Put a list of components you use in the table below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index # | Description | Part Number | Value | Count | Comments |
| 1 | UltraSonic Sensor | OSEPP HC-SR04 | Working Current: 15 mA | 1 | To use ability to measure distance as a sort of ‘one way’ sonar. |
| 2 | Resistor | N/A | 220 Ohm | 5 |  |
| 3 | Microphone sensor | OSEPP SOUND-1 | 4mA current drain | 1 | Frequency range: 10 ~ 55 Hz |
| 4 | Capacitor |  | 100 pF | 1 | For use as an analog tone/volume filter for the buzzer, as dictated by the potentiometer |
| 5 | Potentiometer | ASP Spain | 10 KΩ | 1 | Use to signal filter into capacitor based on position. |
| 6 | Piezo Capsule | Adafruit Accessories PS1240 |  | 1 | Audio actuator, the higher the frequency the closer the object. |
| 7 | Green LED |  |  | 1 | Object Detected - Far |
| 8 | Yellow LED |  |  | 1 | Closer |
| 9 | Red LED |  |  | 1 | Close |
| 10 | Blue LED |  |  | 1 | Brightness based indicator |

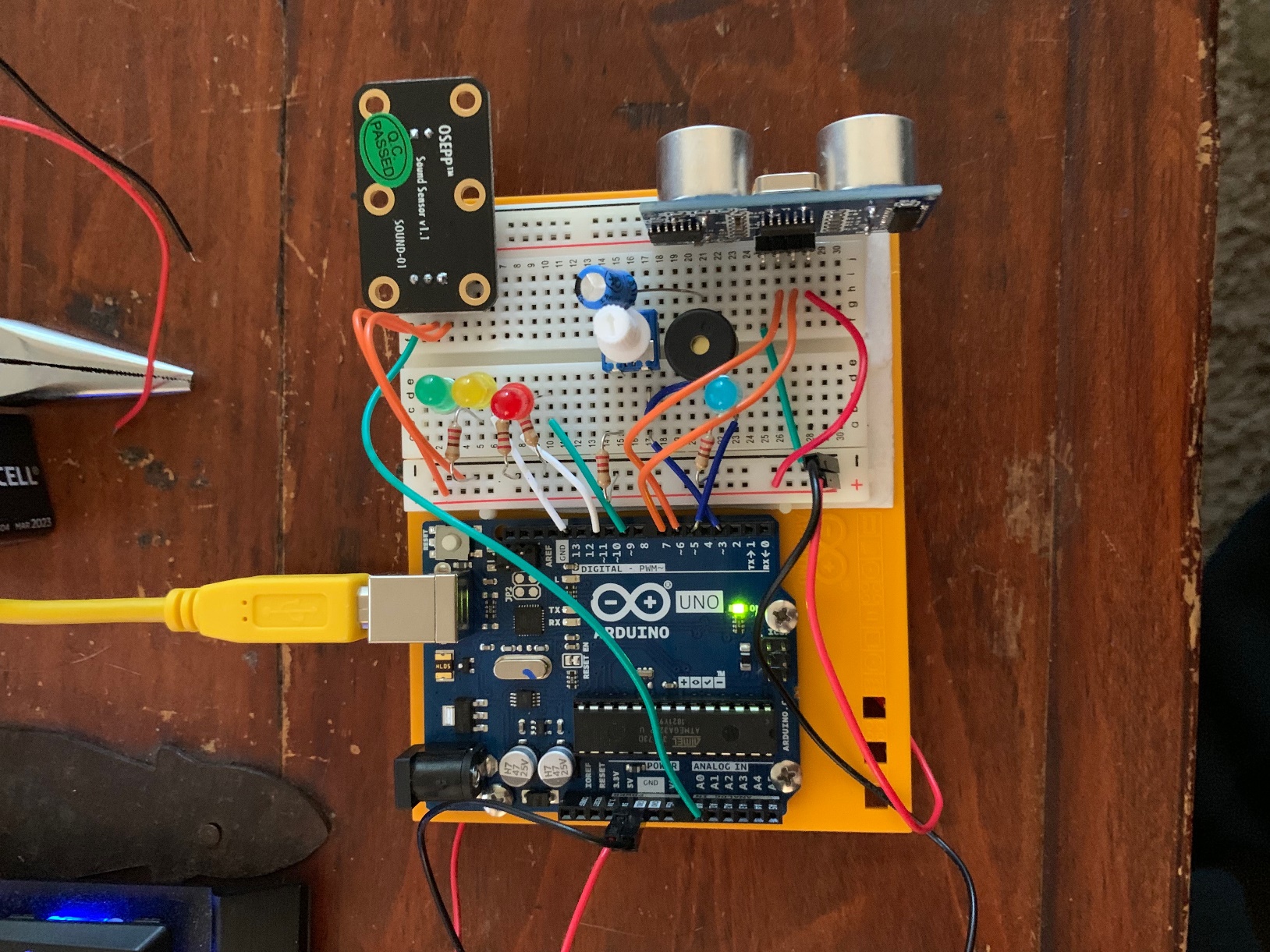
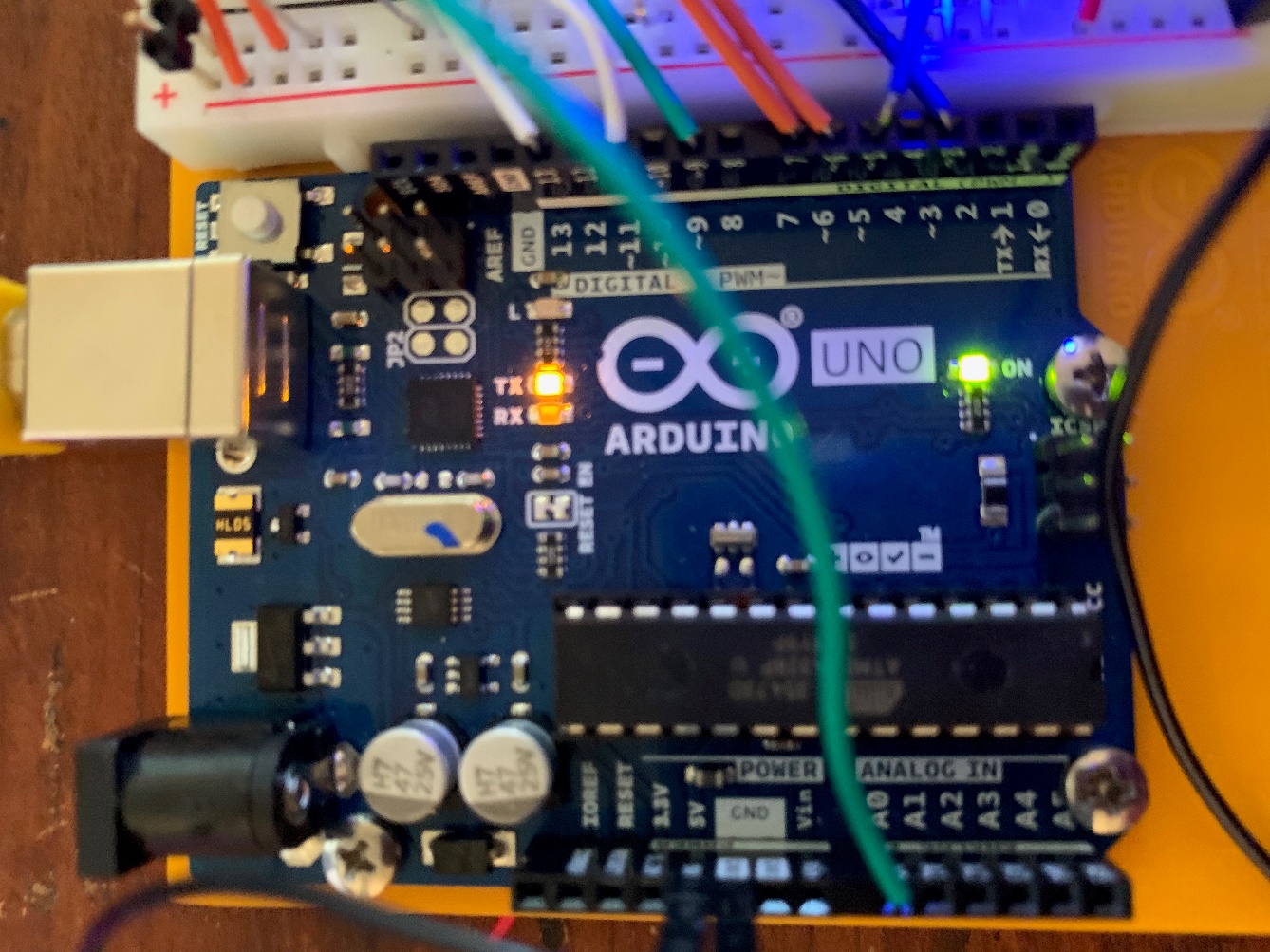
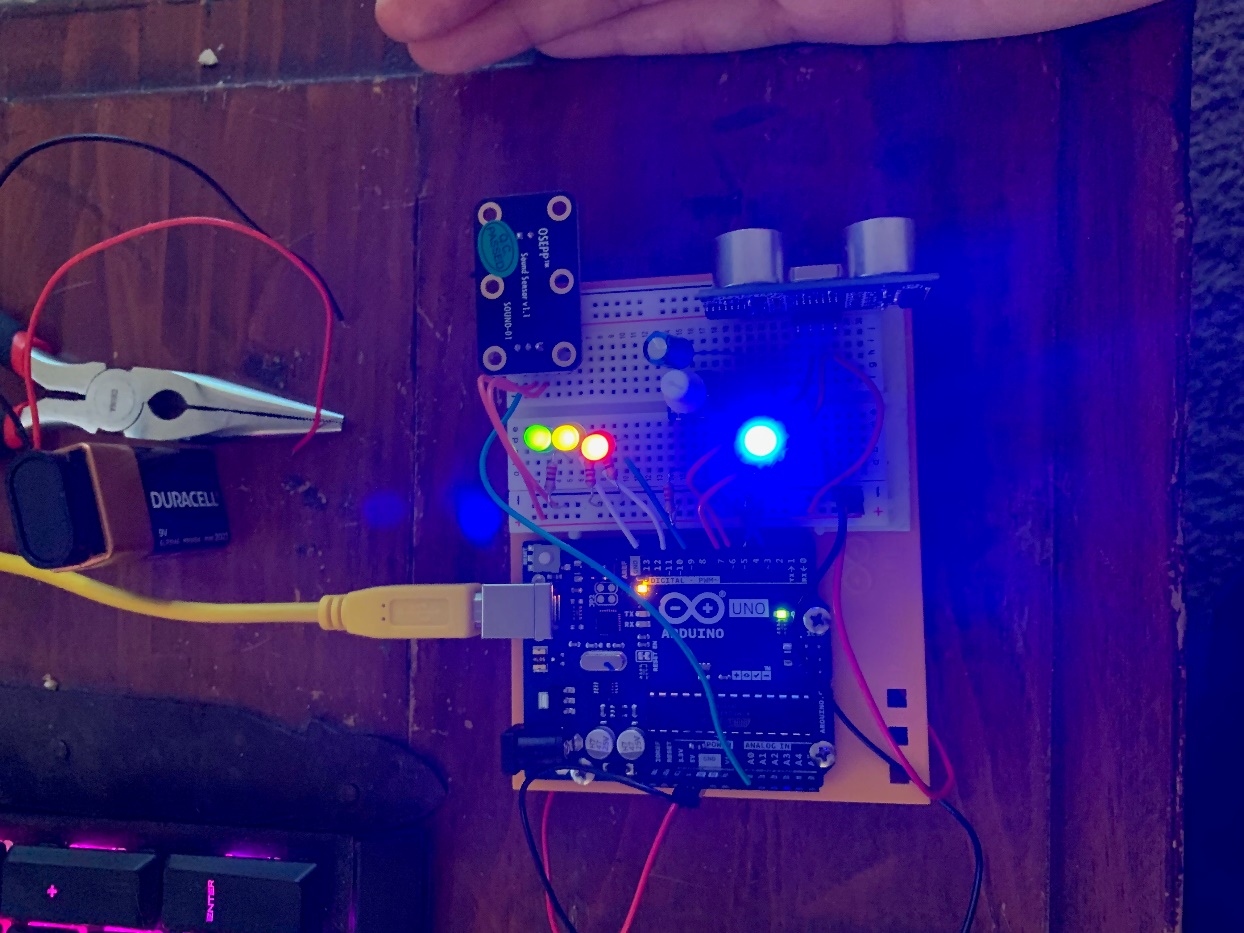
Circuit



Project

This link will take you to a quick, private YouTube video made using iMovie and my IPhone with a total production value of 0$.

<https://youtu.be/p2IqZHUSM3w>



Program Code

///////////////////////////////////////////

// Program Name: Trip proximity sensor with 3 tone buzzer

// Program Version: 1.0.11.0

// Author: Jeremy Reuwer 5580983

// Date: 4/11/2019

// Inputs to Program: measurements from sound sensor and ultrasonic

// Outputs from Program: sensor will trigger light and sound indicators to indicate presence

//////////////////////////////////////////

// initialize pins and their respective components

// I originally planned to have 6 LED indicators, however the breadboard became cluttered and the code still works so I’ve left spaces future 'upgrade-ability'

#define trigPin 7

#define echoPin 6

#define led 13

#define led3 11

#define led5 9

#define buzzer 3

int sound = 250; // set analog buzzer initial frequency

int soundSensor=4; //sound sensor pin number

int LED = 2; //Blue LED input pin number

boolean LEDStatus=false; //create boolean value to have LED status to OFF

int micPin = A0; // microphone sensor input

int ledPin = 5; // select the pin for the LED

int micValue = 0; // variable to store the value coming from the mic sensor

void setup() {

Serial.begin (9600); //initialze arduino

pinMode(soundSensor,INPUT); // get pin state for US sensor

pinMode(LED,OUTPUT); // pin state for blue indicator

pinMode(trigPin, OUTPUT); // state for sending sonic signal

pinMode(echoPin, INPUT); //state for receiving sonic signal

pinMode(led, OUTPUT); // elemnts for first led

pinMode(led3, OUTPUT);

pinMode(led5, OUTPUT);

pinMode(buzzer, OUTPUT); //initilize buzzer

for(int hello = 0; hello < 4; hello++){ //wake up the buzzer with initialization A diminished Triad

tone(buzzer, 160);

delay(300);

tone(buzzer, 192);

delay(300);

tone(buzzer, 231);

delay(300);

}

}

void loop() {

long duration, distance; //establish two varaibles we will be recorded by the sensor

digitalWrite(trigPin, LOW); // take trigger pin and send low status for first pulse

delayMicroseconds(2);

digitalWrite(trigPin, HIGH); // trigger pin on high pulse for contrast

delayMicroseconds(10);

digitalWrite(trigPin, LOW); //repeat

duration = pulseIn(echoPin, HIGH); // have echo pin record the duration of the pulse from on high duty cycle to high duty cycle

distance = (duration/2) / 29.1; // calc distance dividing the duration of the 'echo' by the

//================================================

// Microphone - LED response code

//================================================

micValue = analogRead(micPin); //get analog value of microphone sensor reading

micValue = constrain(micValue, 0, 10); //set sound detect clamp 0-10 .... was 100 but I damaged the microphone!

// turn the ledPin on

int ledbrightness = map(micValue, 0, 10, 0, 255); //establish varible for ledbrightness control

Serial.print("incoming value from microphone ="); //print value

Serial.println( micValue);

analogWrite(ledPin, ledbrightness); // and output desired LED brightness in correlation to microphone levels

delay(100); //delay for adjustable sensativity

//---------------------------------------------------

// Code for US sensor and led light up with 3 -tone "theremin"

//---------------------------------------------------

//created nested 'if' based on the distance recorded by the US sensor

if (distance <= 30) { //distance is thirty meters or less

digitalWrite(led, HIGH); // turn on our first LED

sound = 250; //play 250 hz pitch

}

else {

digitalWrite(led,LOW);

}

if (distance < 20) {

digitalWrite(led3, HIGH);

sound = 270;

}

else {

digitalWrite(led3, LOW);

}

if (distance < 10) {

digitalWrite(led5, HIGH);

sound = 290;

}

else {

digitalWrite(led5,LOW);

}

if (distance > 30 || distance <= 0){

Serial.println("Out of range");

noTone(buzzer); //mute that buzzer if nothings in range

}

else {

Serial.print(distance);

Serial.println(" cm");

tone(buzzer, sound);

}

delay(500); //delay 500 miliseconds for less sensitivty to changes

}

Program Input/Output From Serial Monitor

Hello! A triad of sounds will occur to indicate initliazation

Begin reading

incoming value from microphone =0

Out of range

incoming value from microphone =4

Out of range

incoming value from microphone =0

Out of range

incoming value from microphone =1

Out of range

incoming value from microphone =1

Out of range

incoming value from microphone =8

Out of range

incoming value from microphone =0

Out of range

incoming value from microphone =0

Out of range

incoming value from microphone =10

Out of range

incoming value from microphone =5

28 cm

incoming value from microphone =10

24 cm

incoming value from microphone =4

20 cm

incoming value from microphone =0

27 cm

incoming value from microphone =8

13 cm

incoming value from microphone =3

11 cm

incoming value from microphone =0

9 cm

incoming value from microphone =5

7 cm

incoming value from microphone =0

4 cm

incoming value from microphone =10

3 cm