

☁️ Final Project - Cloud Lamp ☁️

https://github.com/msc-creative-computing/p-comp-leaho/tree/main/FinalProject_CloudLamp



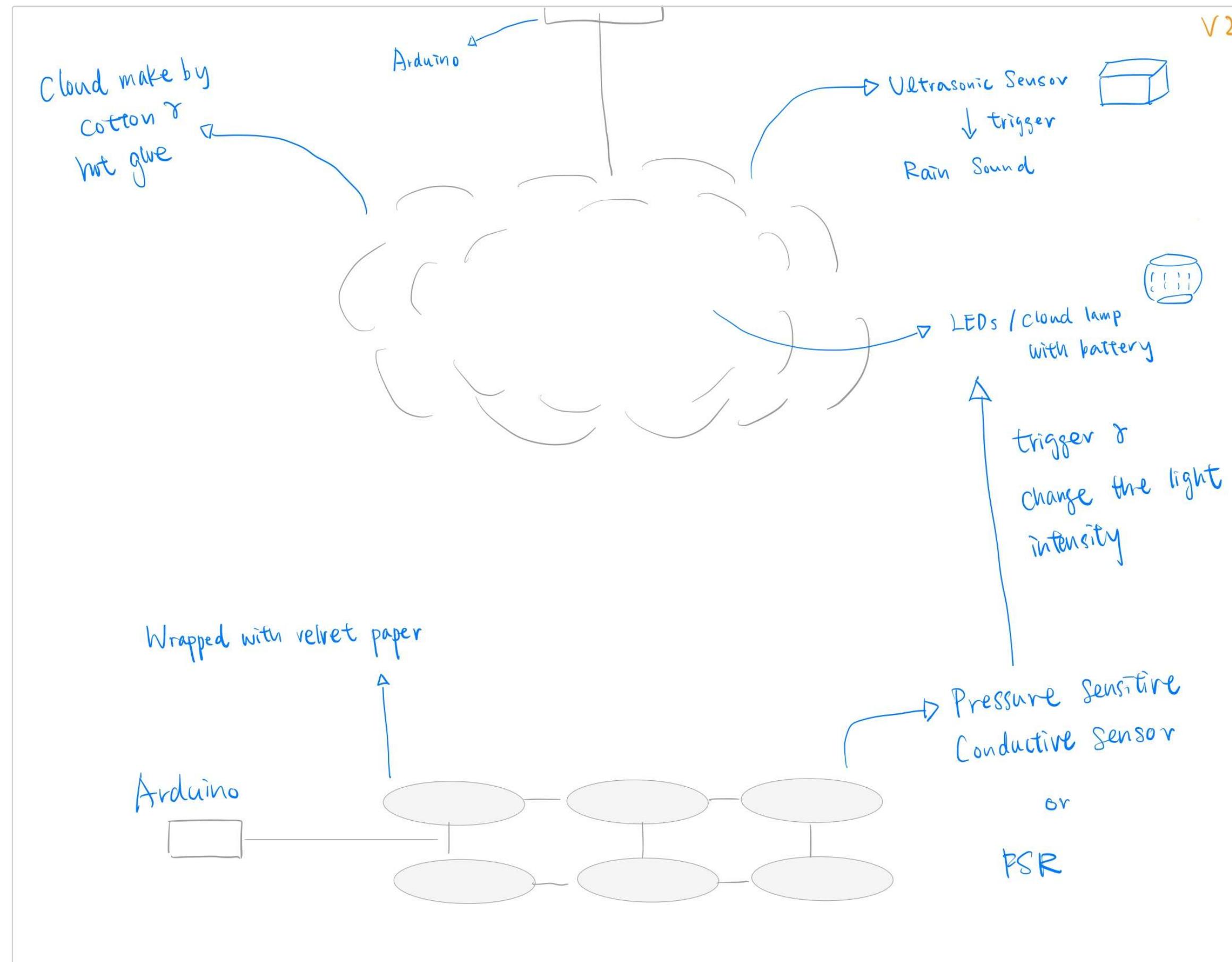
“Let the rain wash away all the pain of yesterday”

This project uses Arduino and c++ language to design an interactive cloud lamp. Due to the epidemic in the past two years, everyone has reduced going out. Therefore, I hope this cloud lamp can add to the atmosphere at home and simulate the feeling of the weather outside. For me, a rainy day is the most representative of the weather in the UK, and it has a romantic atmosphere, so I chose clouds as the primary appearance. This device can trigger sounds through distance, and lights will come on in a dark environment. Neo pixels are used as the primary light source, and the flashing light is triggered through the light dependent resistor. In addition, the music is read from the SD CARD and played when the audience approaches the cloud light. Hoped that through this device, the audience can simulate the atmosphere of a rainy day at home, get a relaxed mood and relief during stressful days.

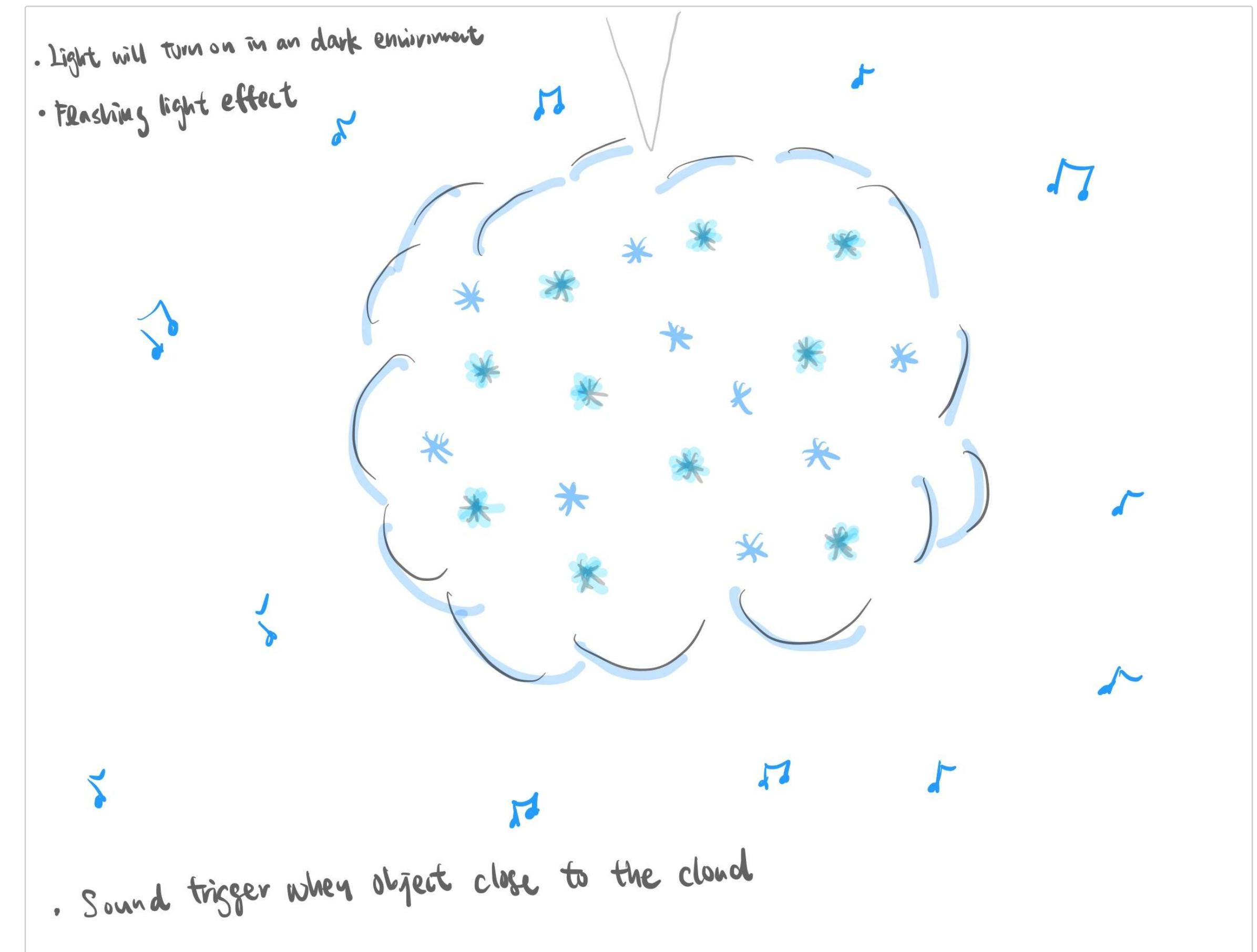


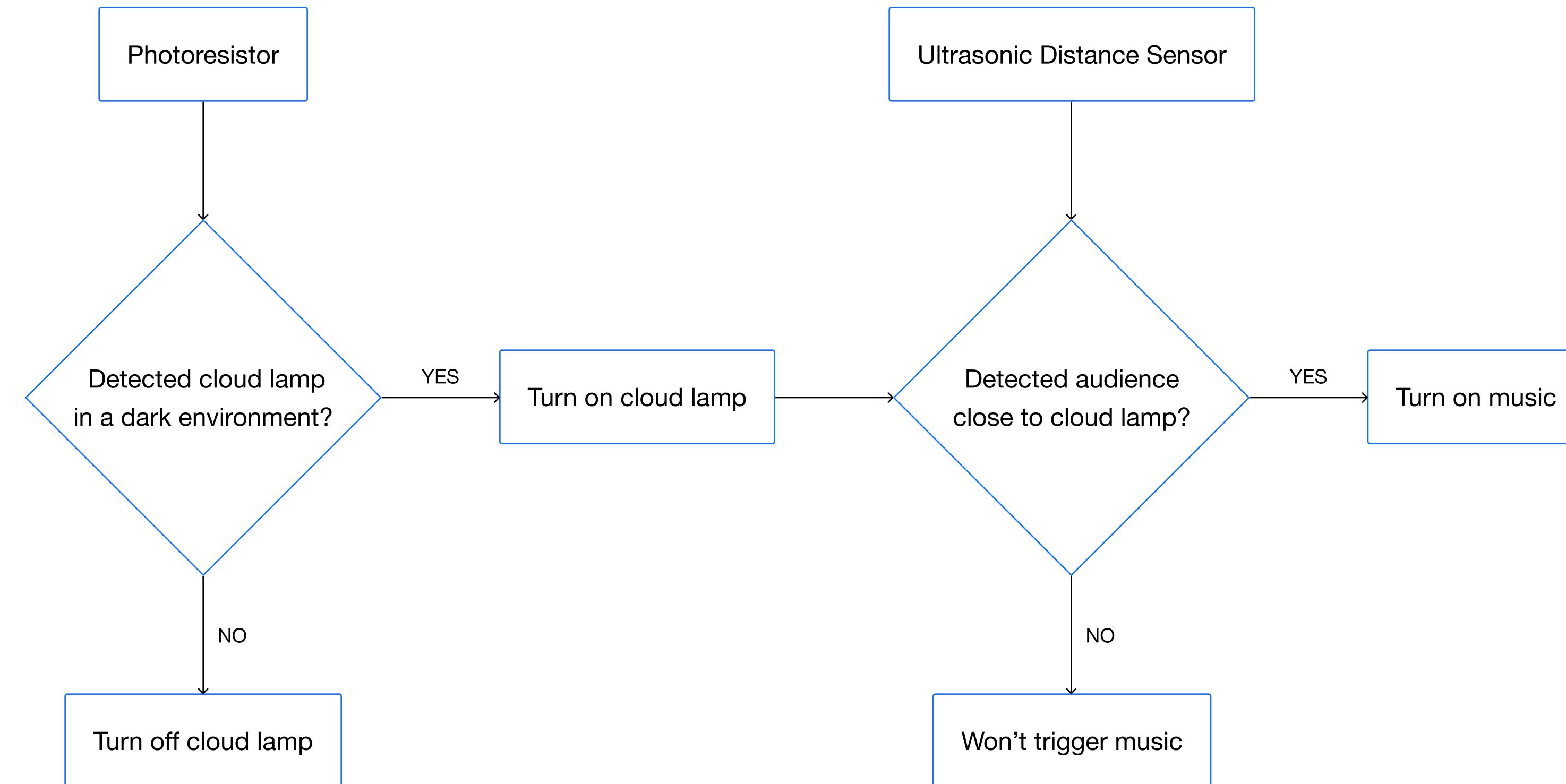


Initial Idea



Final Idea





▶ Youtube

<https://youtu.be/AOFLFPCR2Jg>

/github

https://github.com/msc-creative-computing/p-comp-leaho/tree/main/FinalProject_CloudLamp/03_FinalOutcome



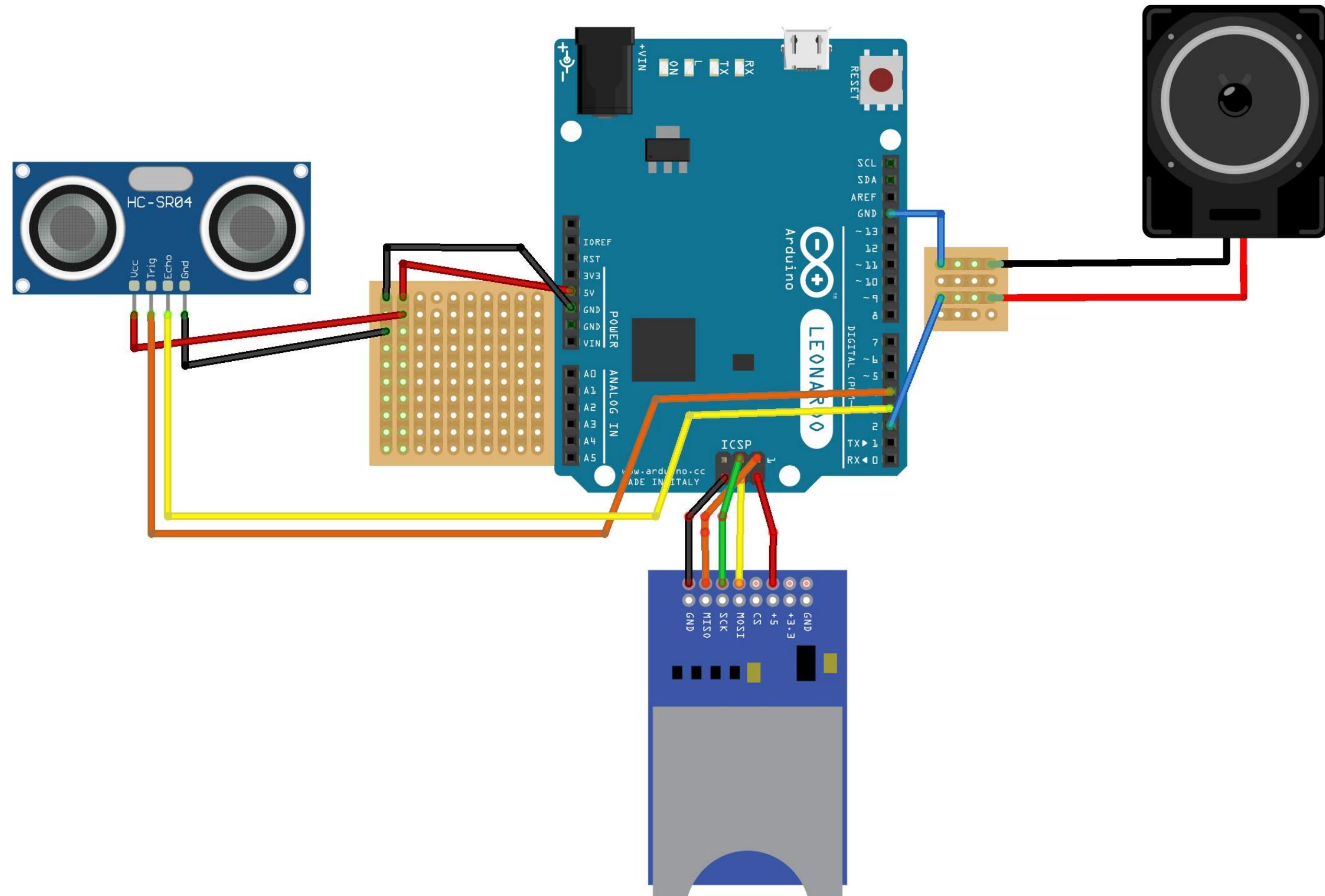
Watch video on Youtube



Watch video on Youtube

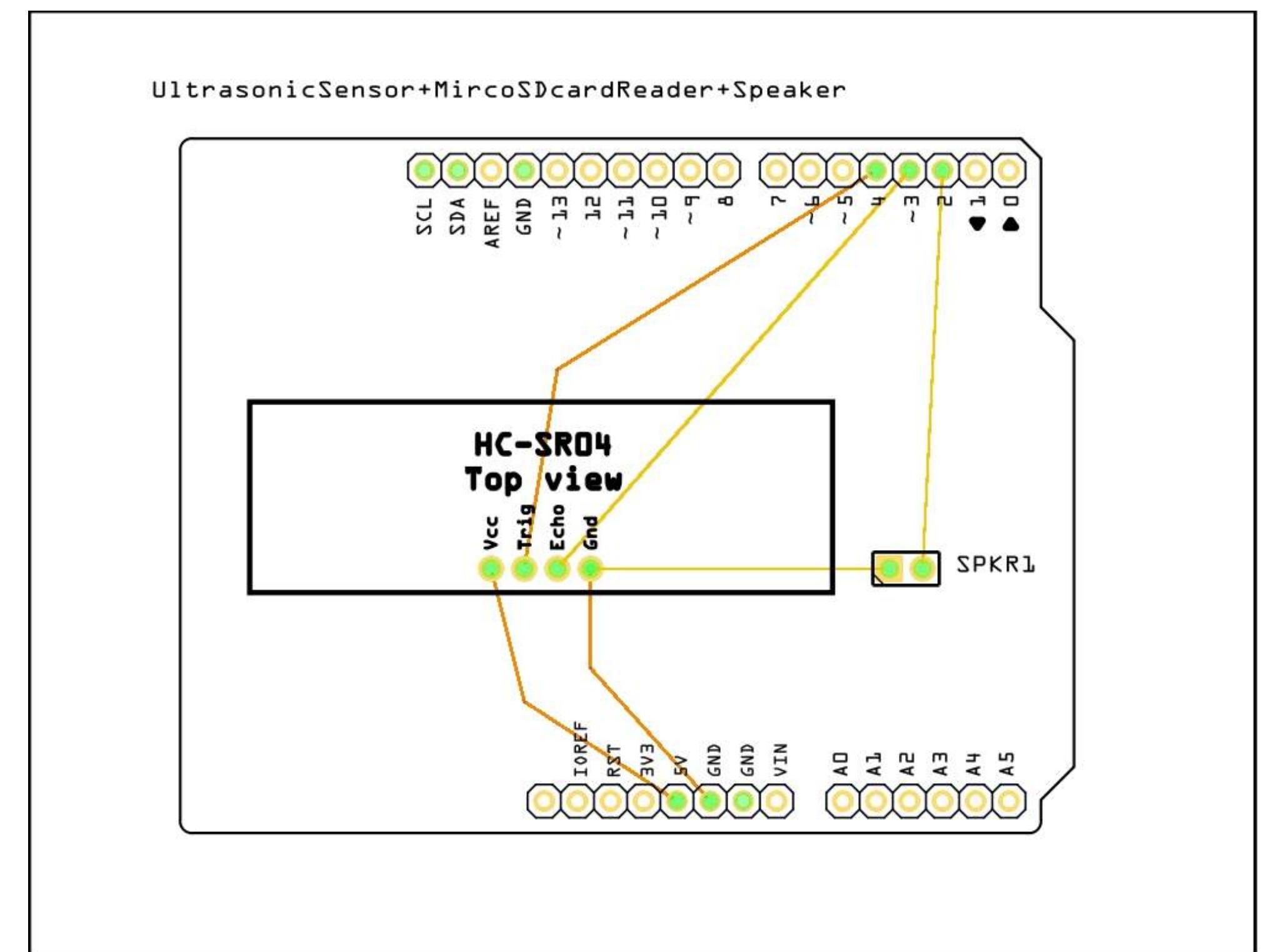
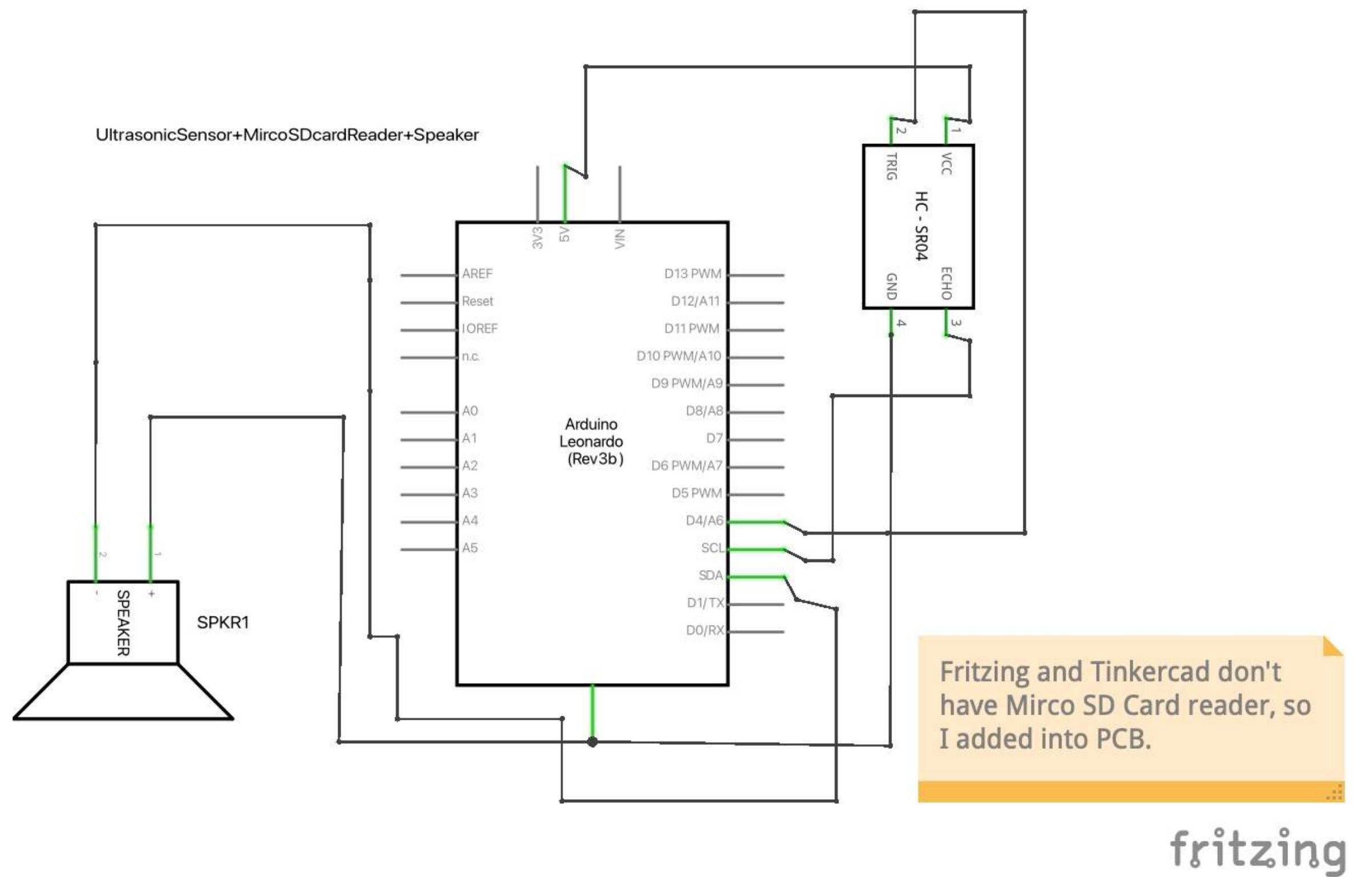


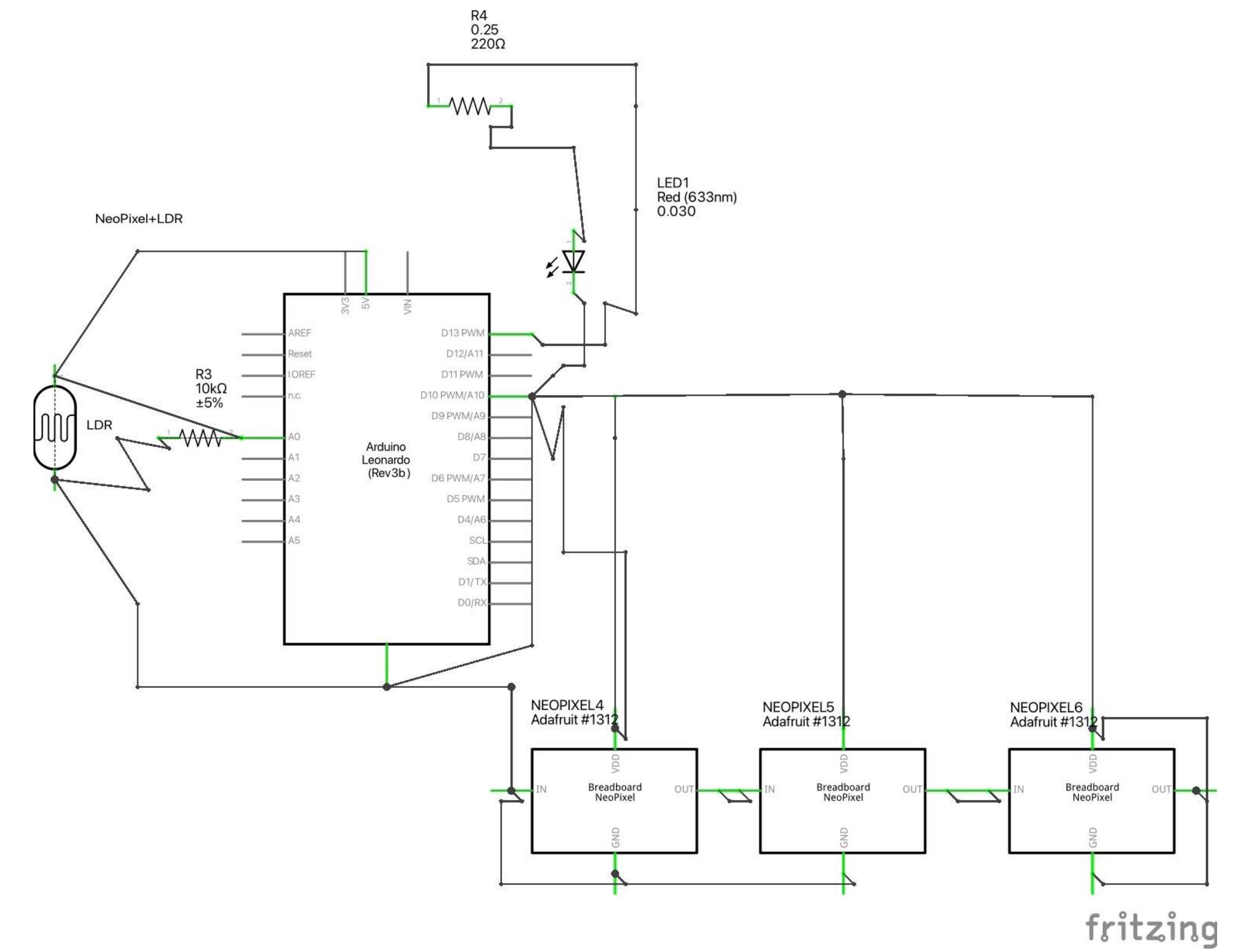
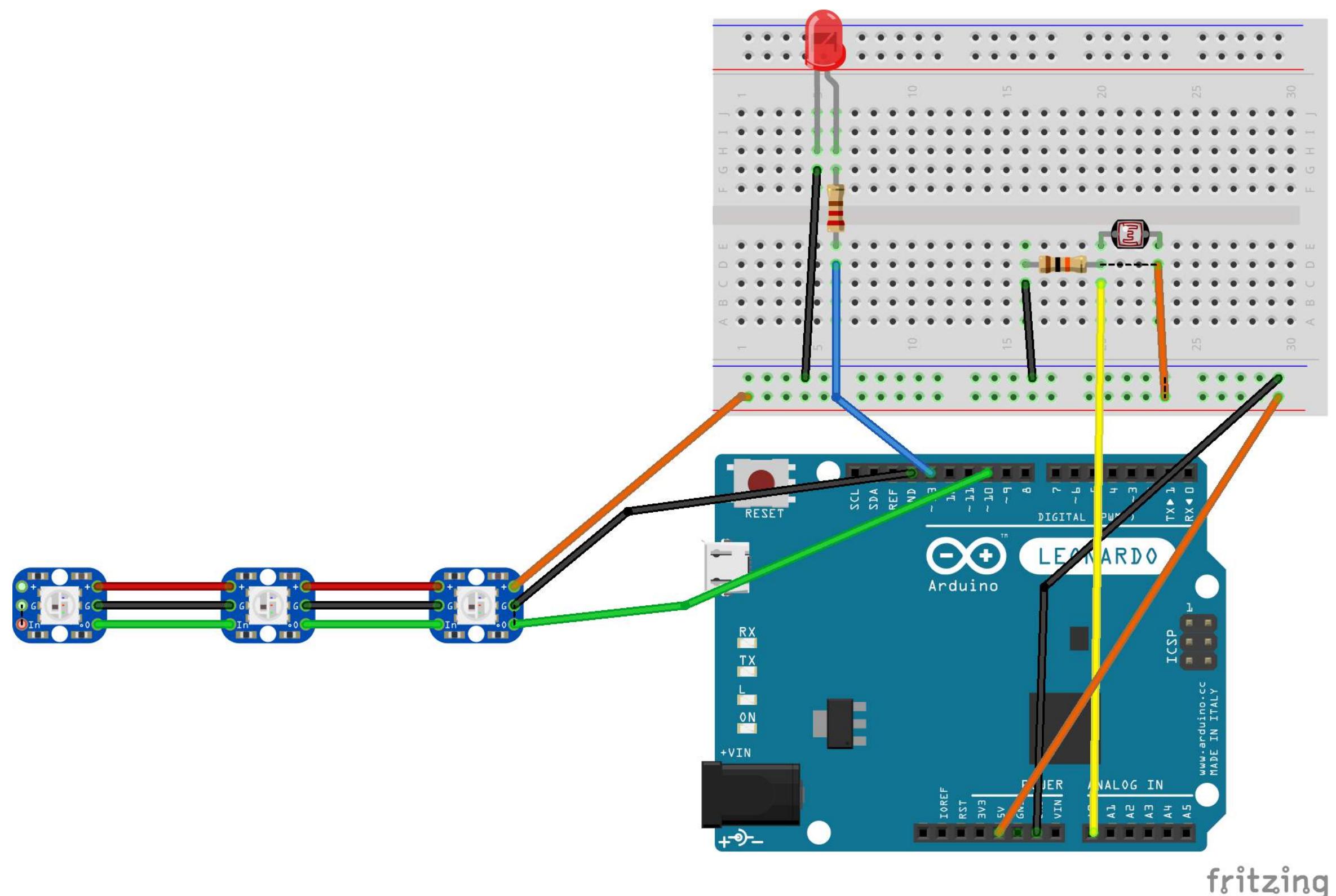
LEA HO (20035099)



fritzing







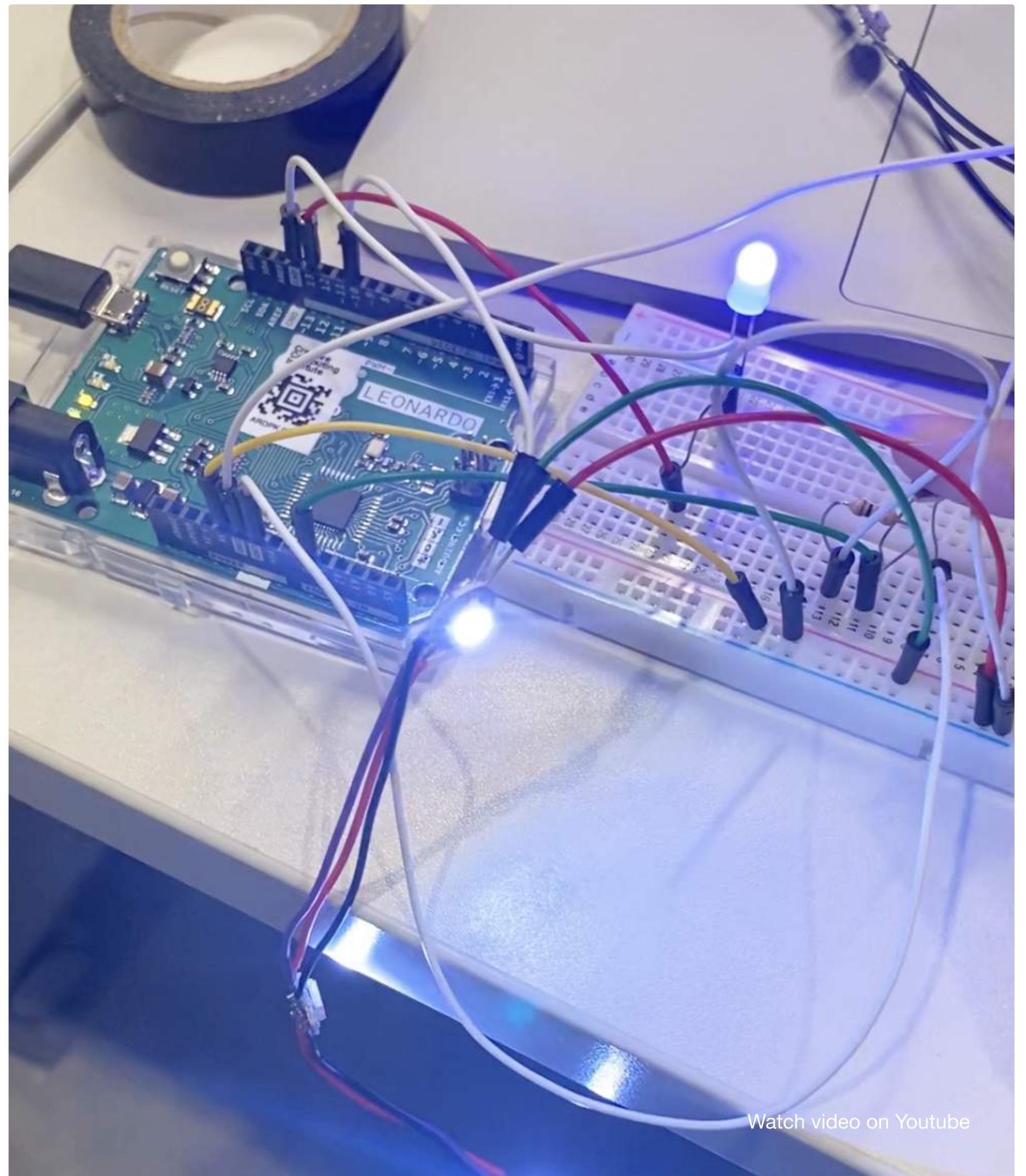
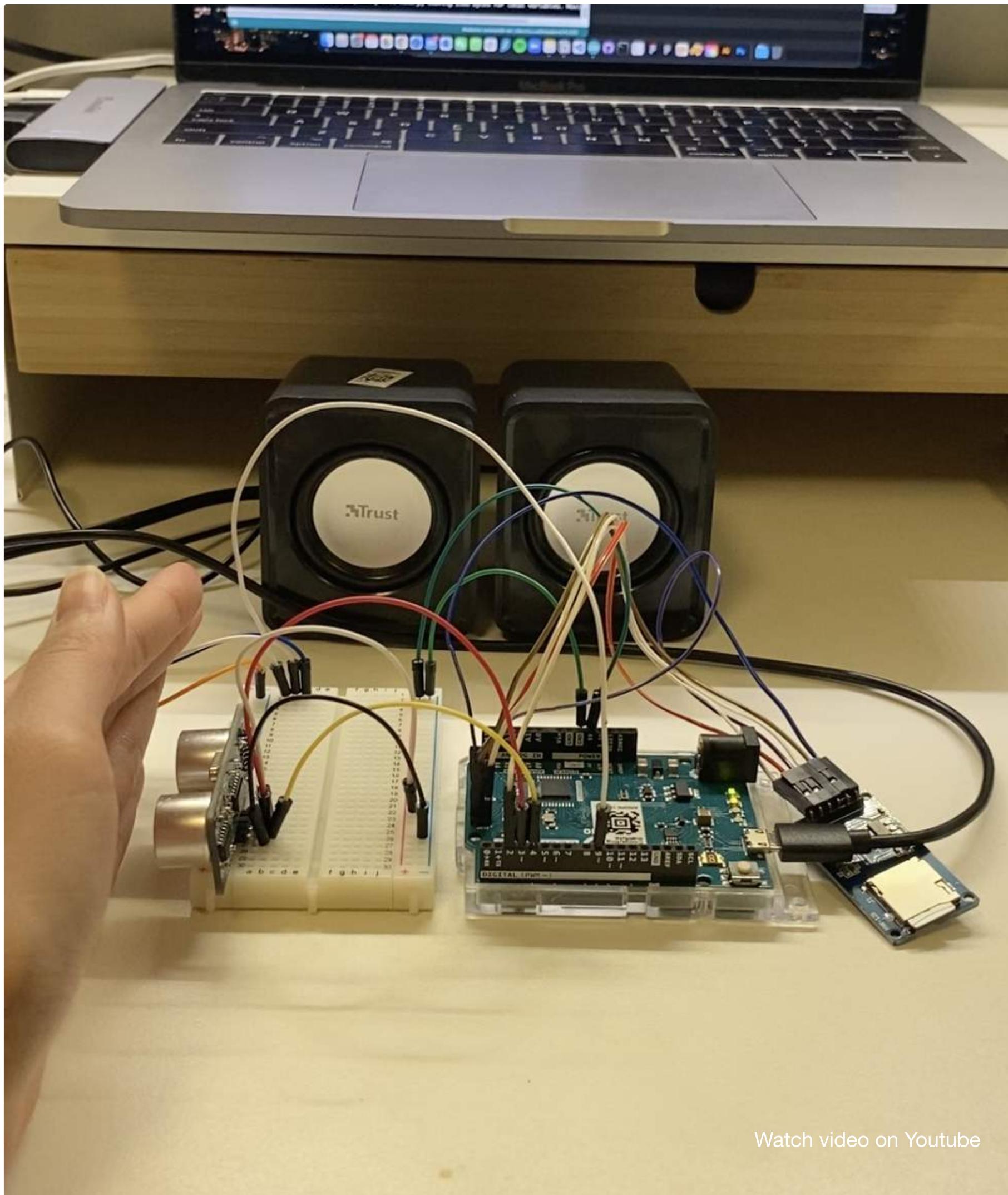
▶ Youtube

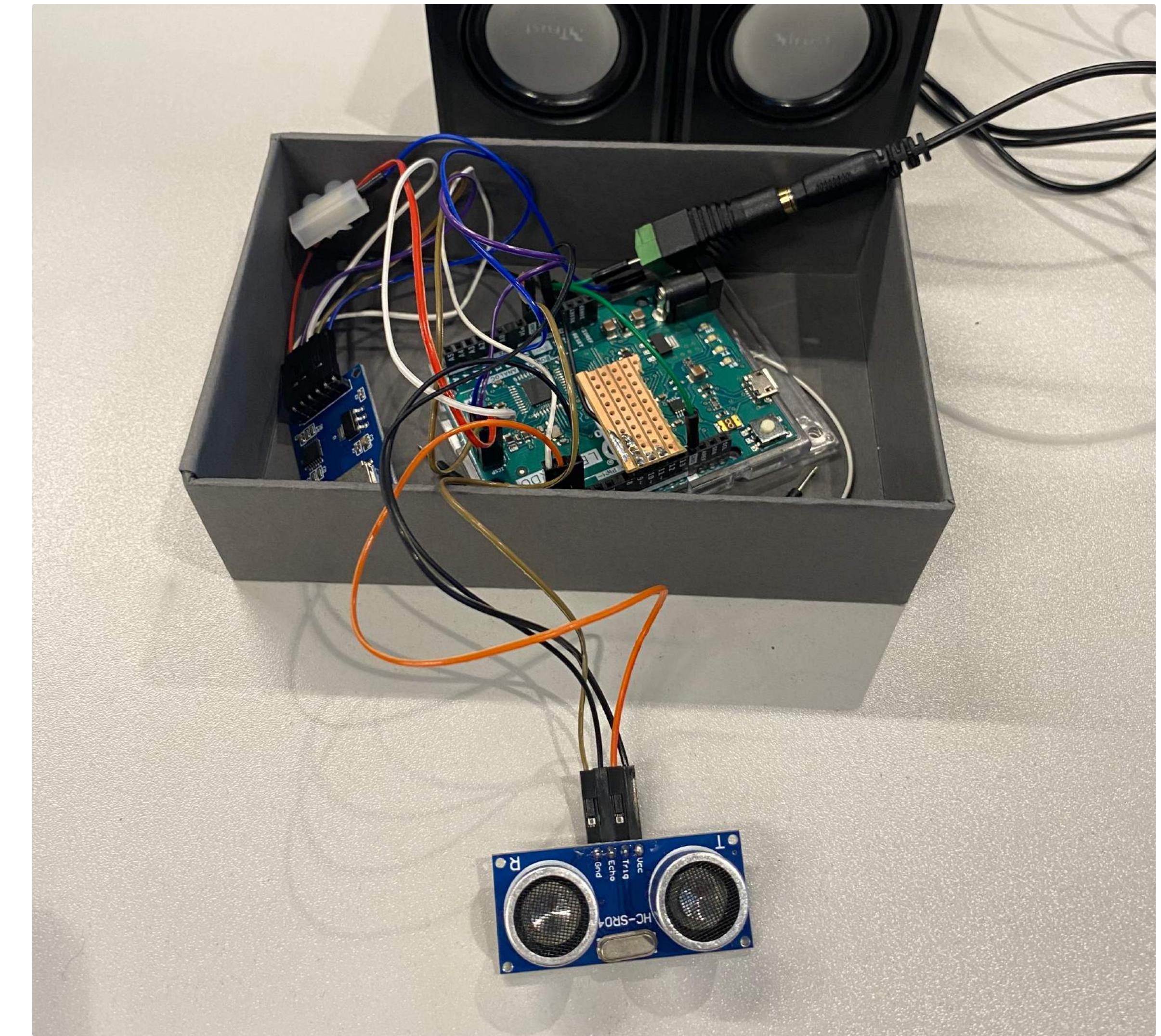
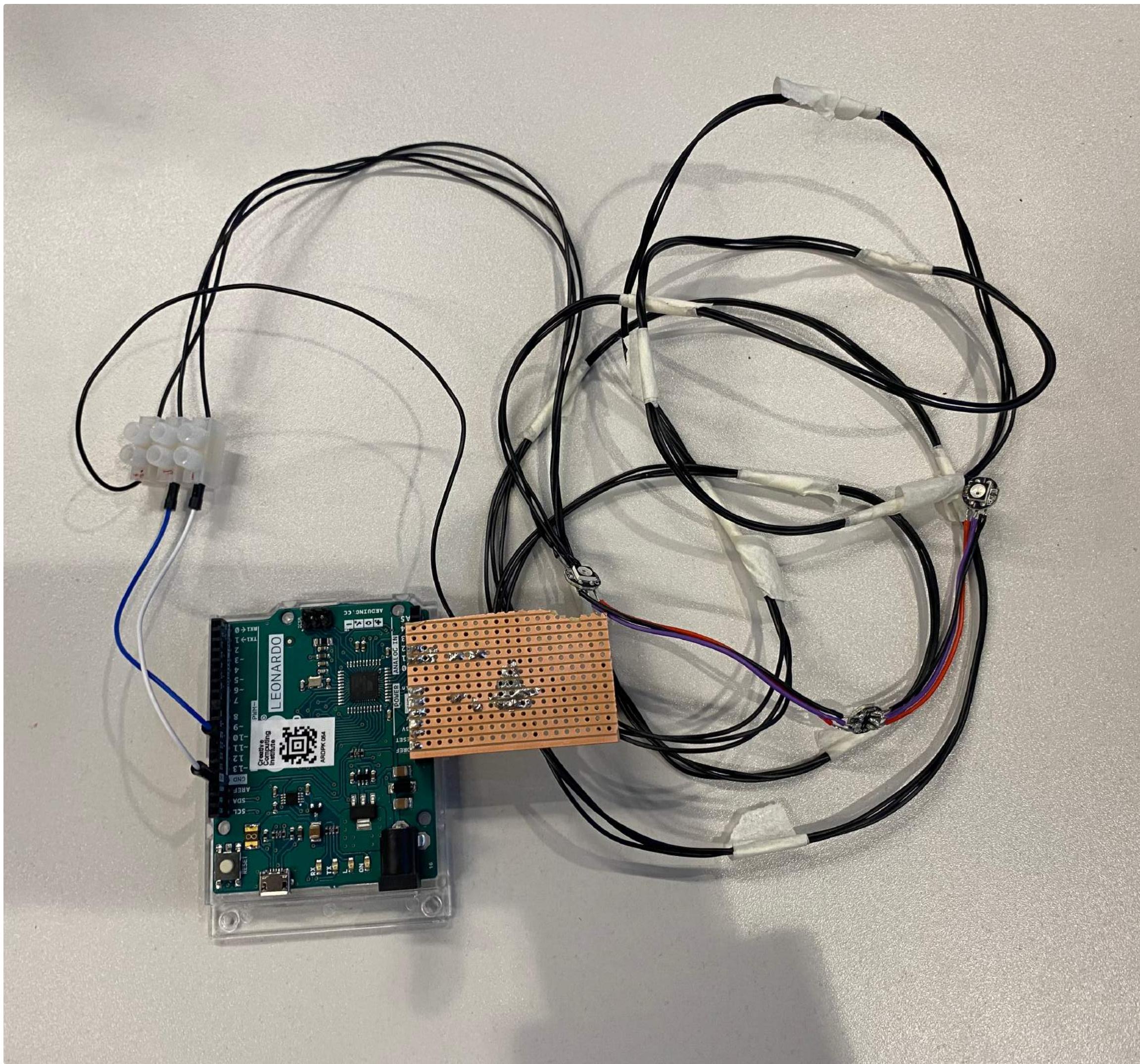
Ultrasonic + Sound Effect

<https://youtu.be/P1T1vtxPjmY>

Neo Pixels + Photoresistor

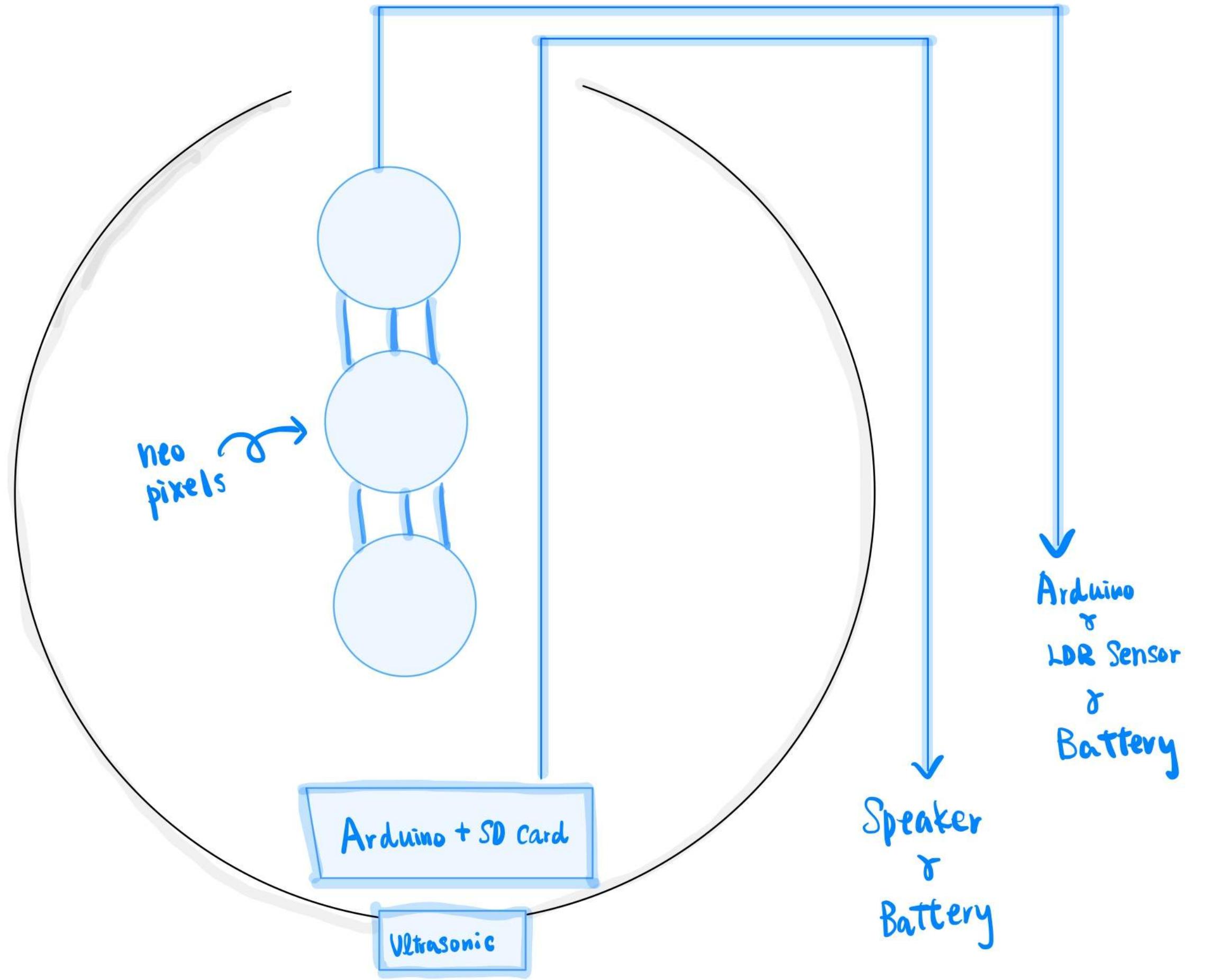
<https://youtu.be/vOGsxIwwf0g>





 Youtube<https://youtu.be/FZVTUhaGy0k>

First, place the neo pixels, ultrasonic distance sensor with Arduino board in the centre of the paper lamp and rerun the function. After all the parts are implemented successfully, use hot glue to place the cotton ball on the outer layer of the paper lamp to design the shape of the cloud.





Github

https://github.com/msc-creative-computing/p-comp-leaho/blob/main/FinalProject_CloudLamp/02_UltrasonicSensor_MircoSDcard_Speaker/01_UltrasonicSensor_MircoSDcard_Speaker/01_UltrasonicSensor_MircoSDcard_Speaker.ino

```

01_UltrasonicSensor_MircoSDcard_Speaker | Arduino 1.8.16
01_UltrasonicSensor_MircoSDcard_Speaker

#include "SD.h"
#define SD_ChipSelectPin 2
#include "TMRpcm.h"
#include "SPI.h"

int trigger_pin = 3;
int echo_pin = 4;

bool startPlaying = false;
bool.isPlaying = false;

TMRpcm tmrpcm;

void setup()
{
pinMode (trigger_pin, OUTPUT);
pinMode (echo_pin, INPUT);

tmrpcm.speakerPin=9;

Serial.begin(9600);

if(!SD.begin(SD_ChipSelectPin))
{
    Serial.println("SD fail");
    return;
}

Serial.println("File Playing");

}

void loop()
{
Serial.println("\n");

int duration;
int distance;

```



```

01_UltrasonicSensor_MircoSDcard_Speaker | Arduino 1.8.16
01_UltrasonicSensor_MircoSDcard_Speaker
Serial.println("File Playing");

}

void loop()
{
Serial.println("\n");

int duration;
int distance;

digitalWrite (trigger_pin, HIGH);
delayMicroseconds (10);
digitalWrite (trigger_pin, LOW);
duration = pulseIn (echo_pin, HIGH);
distance = abs(duration/2)/29.1;

Serial.print("1st Sensor: ");
Serial.print(distance);
Serial.print("cm ");

if (distance <= 10){
    if (!isPlaying){
        startPlaying = true;
    }
}
else if (distance >= 10){
    if(isPlaying){
        tmrpcm.stopPlayback();
        isPlaying = false;
    }
}

if (startPlaying && !isPlaying){
    tmrpcm.play("1.wav");
    startPlaying = false;
    isPlaying = true;

    delay(1000);
}
}

```





https://github.com/msc-creative-computing/p-comp-leaho/blob/main/FinalProject_CloudLamp/01_neoPixel_lightSensor/01_neoPixel_lightSensor.ino

```
01_neoPixel_lightSensor | Arduino 1.8.16
01_neoPixel_lightSensor §

#include <Adafruit_NeoPixel.h>
#include <avr/power.h>

#define PIN          10
#define NUMPIXELS    3

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

long randomNumber;
long randomBrightness;
long randomDelay;

//const int ledPin = 13;
const int ldrPin = A0;

void setup() {
  Serial.begin(9600);
  // pinMode(ledPin, OUTPUT);
  pinMode(ldrPin, INPUT);
  pixels.begin();
}

void loop() {
  randomNumber = random(1,9);
  randomBrightness = random(0,256);
  randomDelay = random(100,400);

  int ldrStatus = analogRead(ldrPin);

  if (ldrStatus <= 300 && randomNumber==1)
  {
    pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  }

  else if (ldrStatus <= 300 && randomNumber==2)
  {
    pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  }
}

/Users/jl/Desktop/FinalProject_CloudLamp/02_UltrasonicSensor_MircoSDcard_Speaker/01_UltrasonicSensor_M
32
Arduino Leonardo on /dev/cu.usbmodem11101
```

```
01_neoPixel_lightSensor | Arduino 1.8.16
01_neoPixel_lightSensor §

else if (ldrStatus <= 300 && randomNumber==2)
{
  pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==3)
{
  pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==4)
{
  pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==5)
{
  pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==6)
{
  pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==7)
{
  pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
  pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else
{
  pixels.setPixelColor(0, pixels.Color(0,0,0));
  pixels.setPixelColor(1, pixels.Color(0,0,0));
  pixels.setPixelColor(2, pixels.Color(0,0,0));
}

/Users/jl/Desktop/FinalProject_CloudLamp/02_UltrasonicSensor_MircoSDcard_Speaker/01_UltrasonicSensor_M
32
Arduino Leonardo on /dev/cu.usbmodem11101
```





Github

https://github.com/msc-creative-computing/p-comp-leaho/blob/main/FinalProject_CloudLamp/01_neoPixel_lightSensor/01_neoPixel_lightSensor.ino

The screenshot shows the Arduino IDE interface with the sketch titled "01_neoPixel_lightSensor". The code uses the NeoPixel library to control three NeoPixel LEDs connected to pins 0, 1, and 2. It also includes a photoresistor sensor connected to pin A0. The code reads the light sensor value, generates a random number between 1 and 7, and then sets the colors of the three NeoPixels based on the sensor reading and the random number. The code ends with a delay and a serial print statement.

```
01_neoPixel_lightSensor | Arduino 1.8.16

01_neoPixel_lightSensor §

}

else if (ldrStatus <= 300 && randomNumber==5)
{
    pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
    pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==6)
{
    pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
    pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else if (ldrStatus <= 300 && randomNumber==7)
{
    pixels.setPixelColor(0, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
    pixels.setPixelColor(1, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
    pixels.setPixelColor(2, pixels.Color(randomBrightness,randomBrightness,randomBrightness));
}

else
{
    pixels.setPixelColor(0, pixels.Color(0,0,0));
    pixels.setPixelColor(1, pixels.Color(0,0,0));
    pixels.setPixelColor(2, pixels.Color(0,0,0));
}

pixels.show();

delay(randomDelay);

pixels.setPixelColor(0, pixels.Color(0,0,0));
pixels.setPixelColor(1, pixels.Color(0,0,0));
pixels.setPixelColor(2, pixels.Color(0,0,0));
pixels.show();

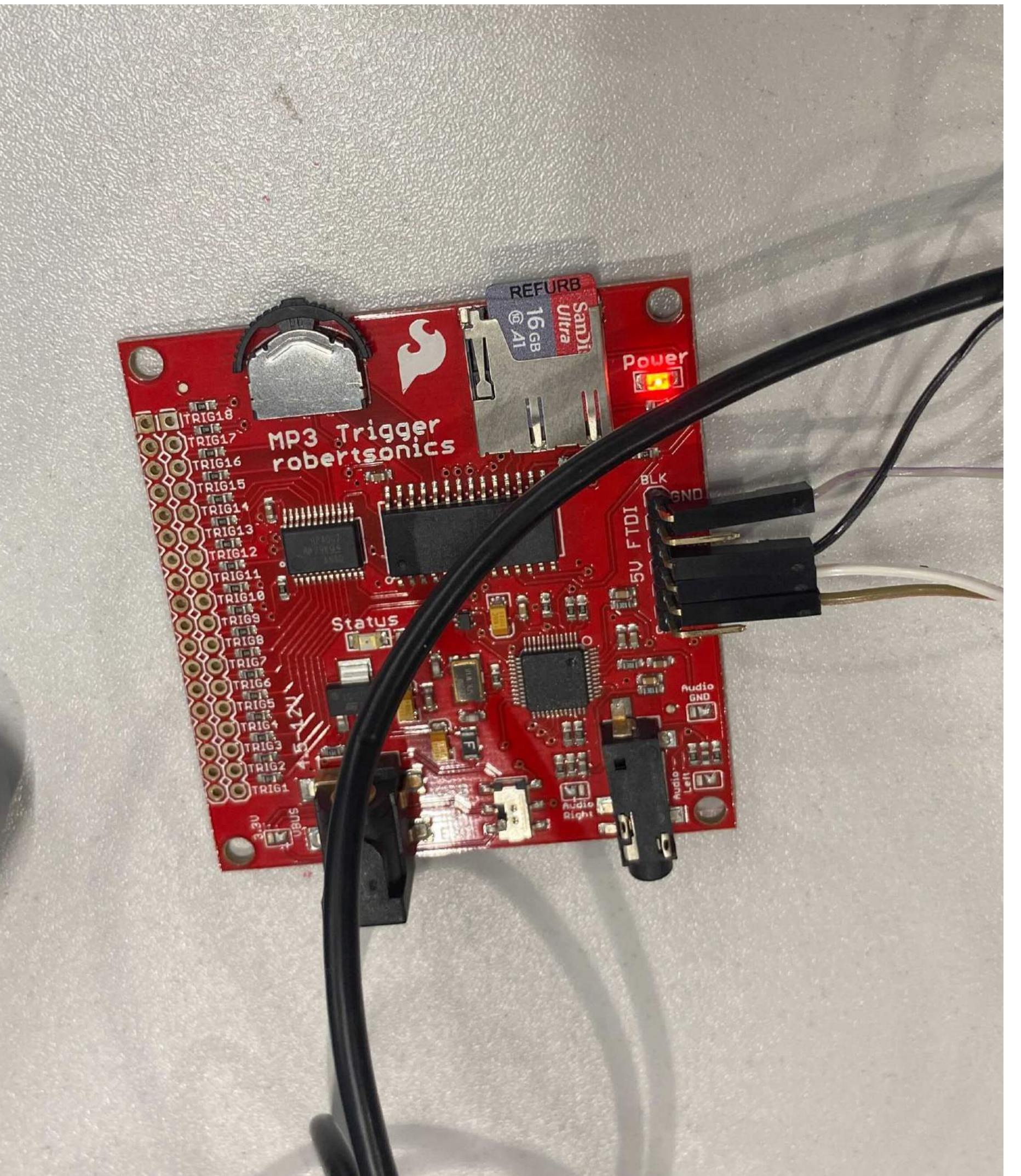
Serial.println("-----");
}

/Users/jl/Desktop/FinalProject_CloudLamp/02_UltrasonicSensor_MircoSDcard_Speaker/01_UltrasonicSensor_M

32
Arduino Leonardo on /dev/cu.usbmodem11101
```



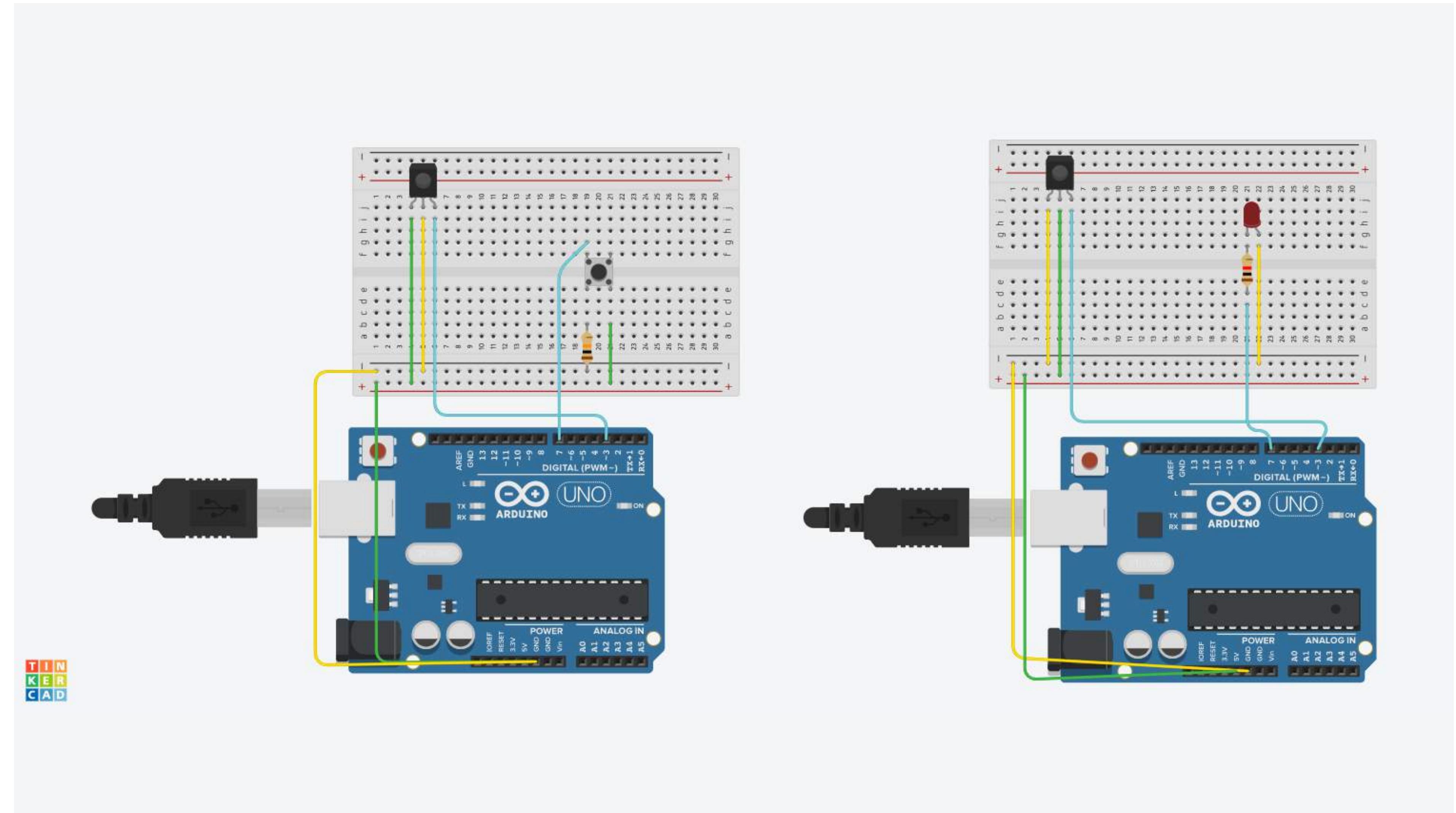
One of the most challenging aspects of the process is that the Arduino reads the music files from the SD card and plays the sound. After many attempts, I still failed to connect Sparkfun's mp3 trigger to Arduino. After consulting Matt, I found that the board might cause the problem. Then I switched to the micro SD card reader. However, the content in the SD card still cannot be read successfully after connecting with Arduino. After research, it turns out that the SPI library I used for the SD card reader has a slightly different pinout connection between the Uno board and the Leonardo board. As a result, Leonardo needs to connect the sd card reader to the ICSP connector, while the Uno need to be connected to pins 10-13.



The original idea was to store all the wires in the cloud light and control the switch remotely. So I tried to use an infrared(IR) receiver and transmitter to communicate with Arduino boards on Tinkercad. LED will light up when one of them receives a signal. However, another Arduino can't receive any message and seems to have some problems related to the voltage. But due to the short time, I gave up this part and converted the remote control into a light dependent resistor as a switch. Nevertheless, I still want to solve this part and control the switch remotely and change the light intensity.

TINKERCAD

<https://www.tinkercad.com/things/72ipFVNStPA-ir/editel?sharecode=hgNiRONDqdePnTcX5vH4LrXv1nuiaqWr9hSs8KAO6YI>



The future direction of this project will be to solve the existing problem then add new functions to the cloud lamp. First, I will store all the wires in the cloud lamp and add IR to control the lamp. In addition to having a better appearance, it will also make it easier for users to control the light switch. And be able to change the light brightness and colours to customize their favourite atmosphere. In addition, it will also fix the noisy sound problem. At present, the sound of rain cannot be clearly heard, and there is a lot of noise from the speaker.

Moreover, I want to add new functions to the cloud light. For example, install the projector in the cloud lamp to reflect the raindrops and to feel the rainy atmosphere even more. Also, add a display screen next to the light to show the outside temperature and weather conditions. I hope that the improvement of the design can provide a new experience and atmosphere to the user.



☁ Thank You ☁

🔗 Github link:

https://github.com/msc-creative-computing/p-comp-leaho/tree/main/FinalProject_CloudLamp

🔗 Youtube:

<https://youtu.be/AOFLFPCR2Jg>

