

PARALLEL WORLD

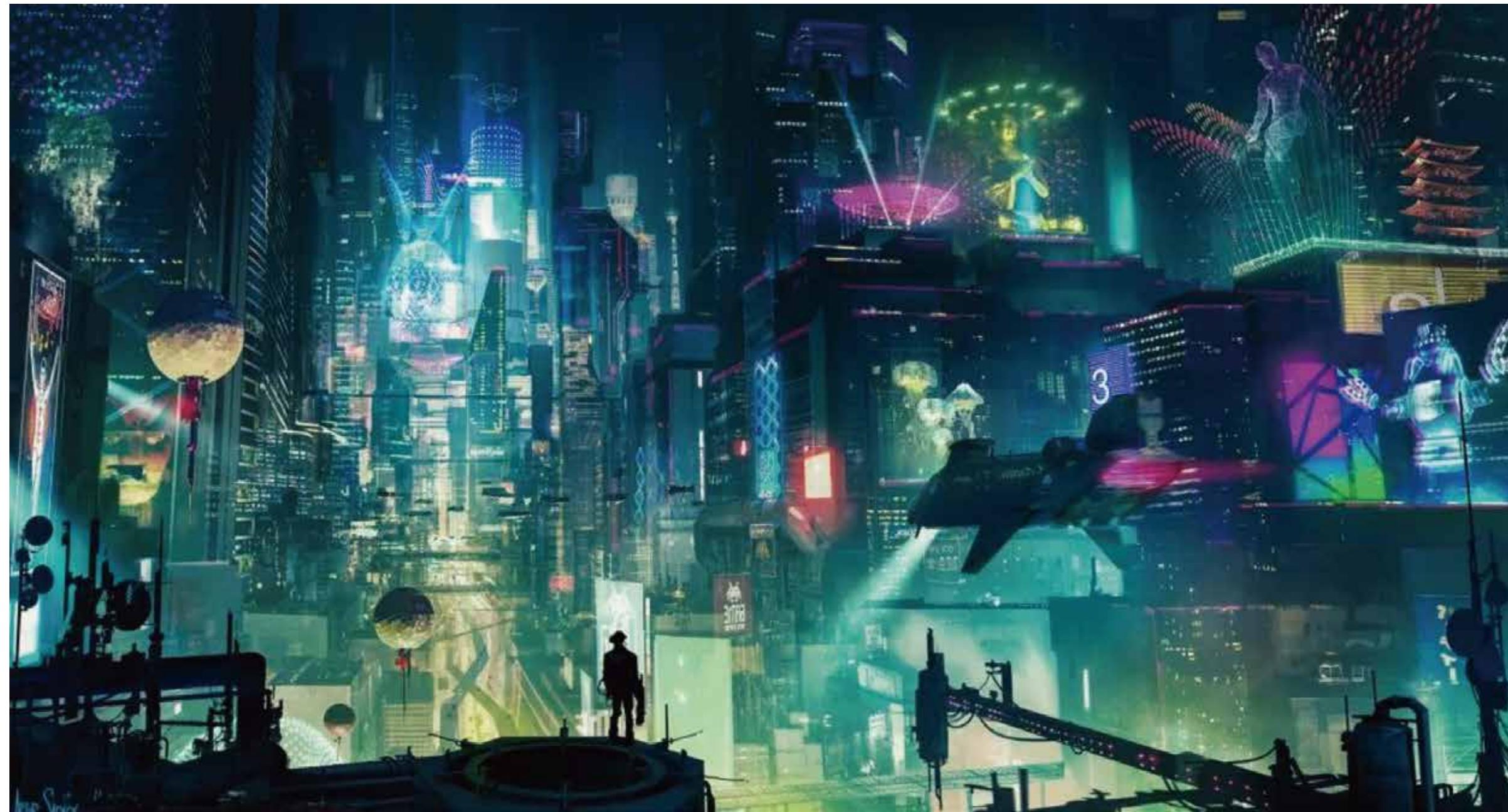
An Art Interactive Device

It simulates the communication between human beings and "themselves" (creatures) in the parallel world.

Video Link: <https://youtu.be/c5zaE4mR4nI>

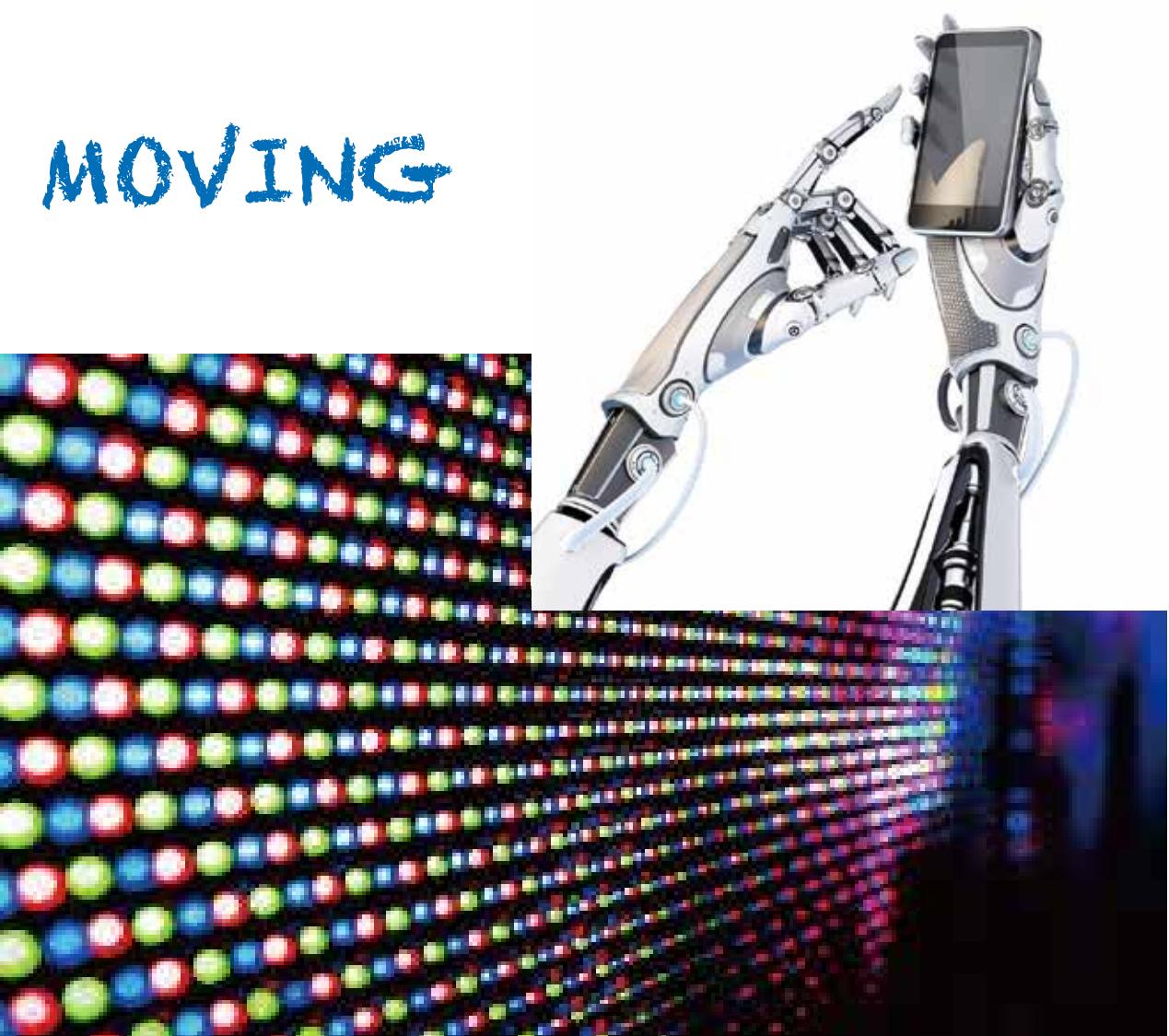
Ray

INSPIRATION



**MY INSPIRATION COMES FROM CYBORG.
I HAVE A LOT OF THOUGHTS AND ASSUMPTIONS ABOUT IT.
I WONDER IF SUCH CREATURES ALREADY EXIST AND LIVE
AS USUAL AS HUMANS. THEY ALSO HAVE THEIR OWN
SOCIAL SYSTEM AND BEHAVIOR, BUT THEY ARE VERY
DIFFERENT FROM HUMANS.**

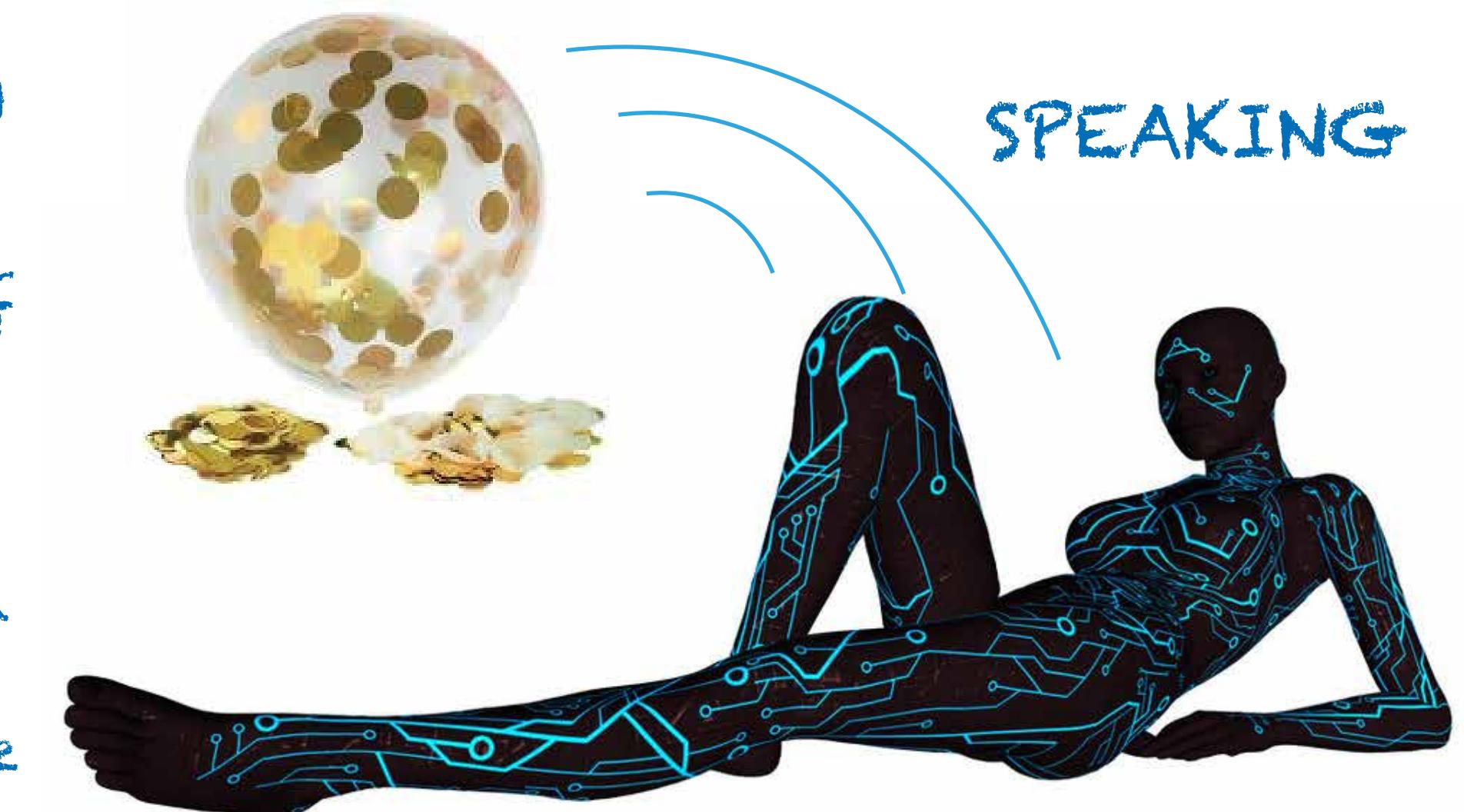
Therefore, I introduced the concept of "parallel world" in this project, which is intended to interact with the creatures from "parallel world".



MOVING

For example, the form of human speaking may be the vibration of a balloon in the parallel world; The form of human hand waving may correspond to the flashing of LED bulbs.

Here, I use the rotation of the balloon to simulate the "communication" signal, and use the LED light to simulate the warning of creatures in the "parallel world" to the safe distance.



SPEAKING

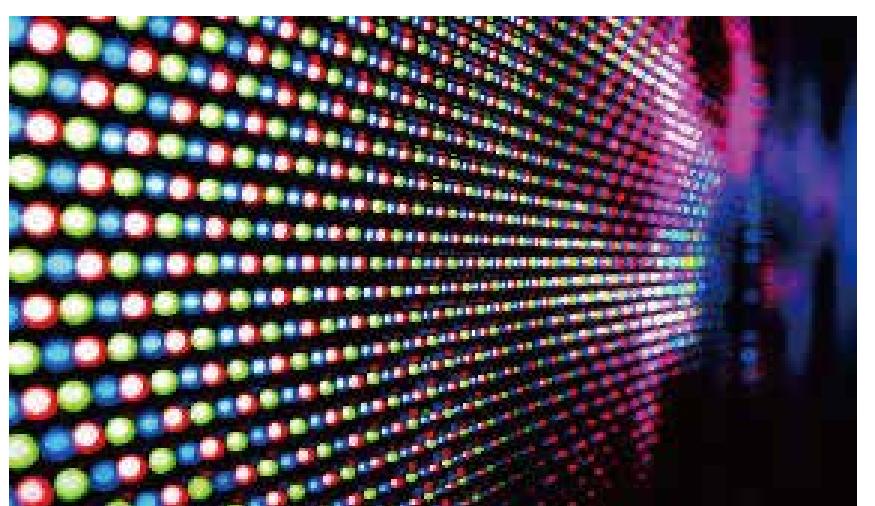
FUNCTIONAL PRINCIPLE



LED BELT
OUTPUT



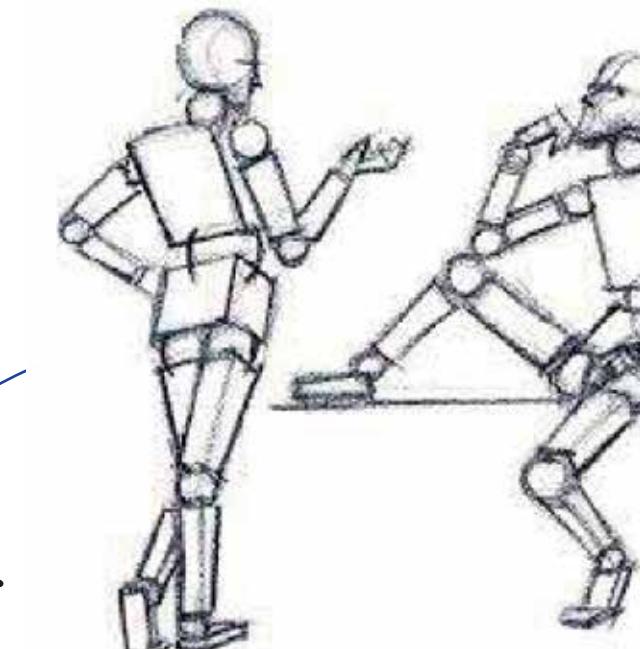
Ultrasonic Sensor



INPUT

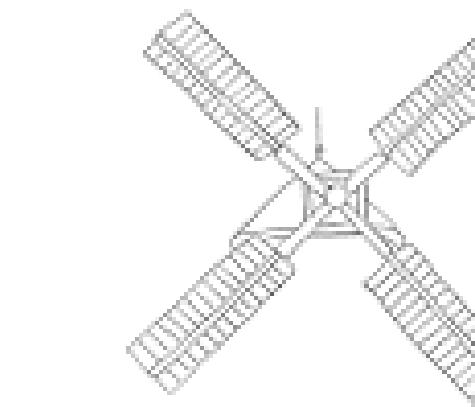
USER

Behavior

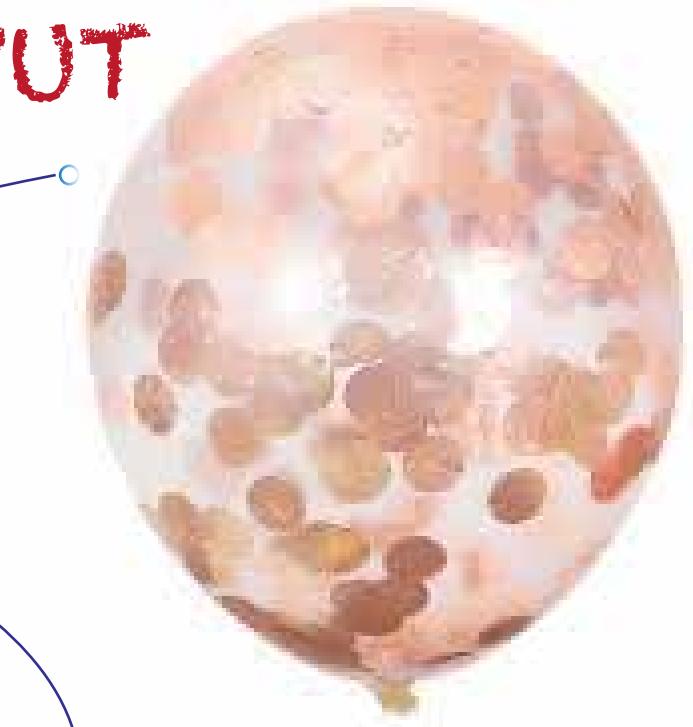


DISTANCE

The link fan blows the sequins in the balloon



OUTPUT



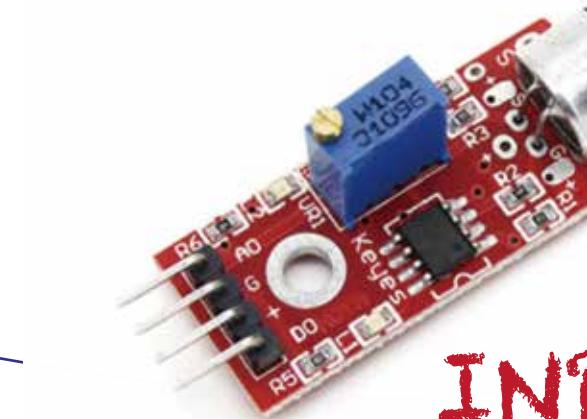
Behavior

Dankie Gracias
Спасибо Merci Takk
Köszönjük Terima kasih
Grazie Dziękujemy Terimakasih
Dakujeme Vielen Dank Paldies
Kiitos Taname teid 謝謝
感謝您 Obrigado Teşekkür Ederiz
Σας Ευχαριστούμ 감사합니다
Bedankt Děkujeme vám ありがとうございます
Tack

Thank You
Talking



Motor



INPUT

Keyes Microphone Sound Detection Senso.

microphone

FUNCTION ONE
PRINCIPLE

Louder the voice
Faster motor turns

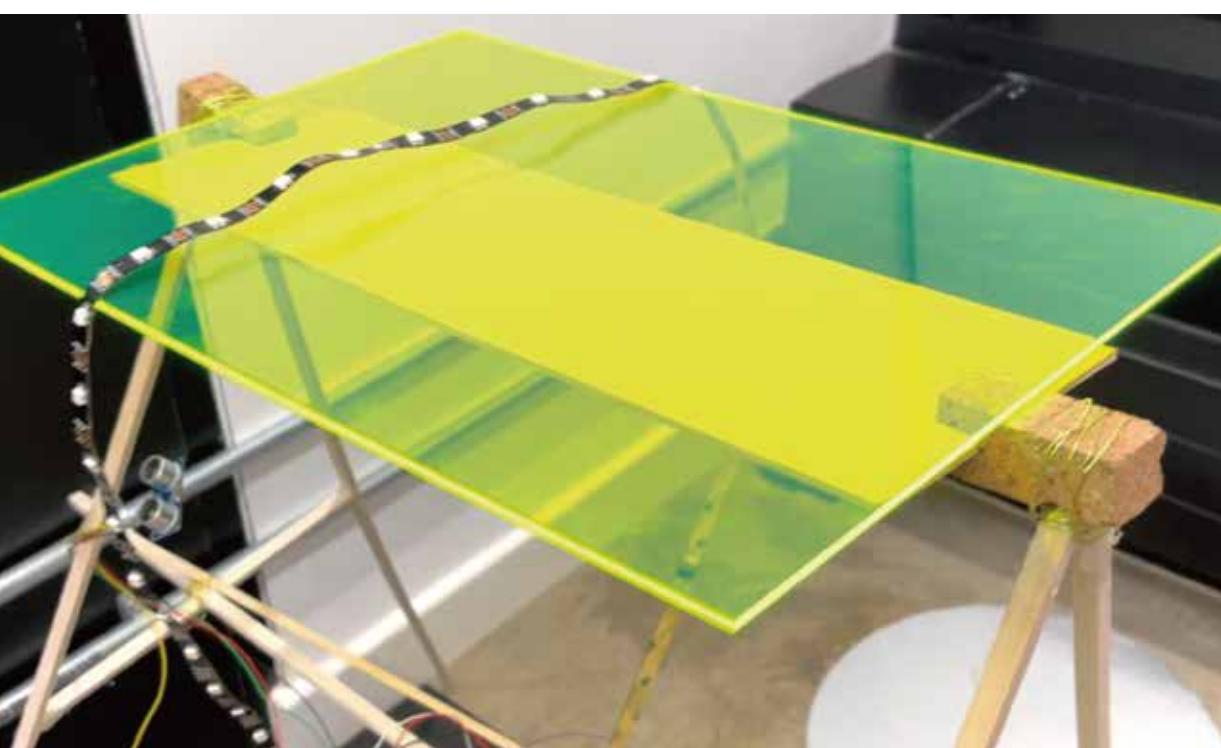
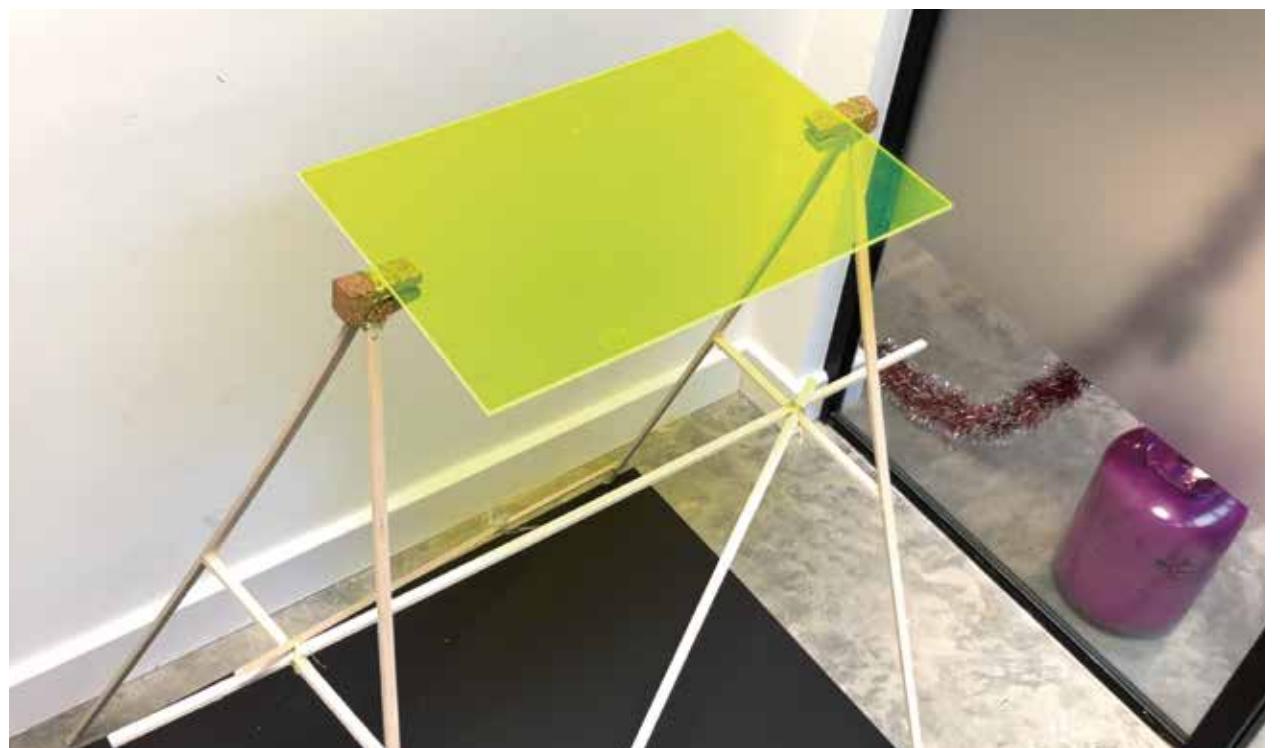
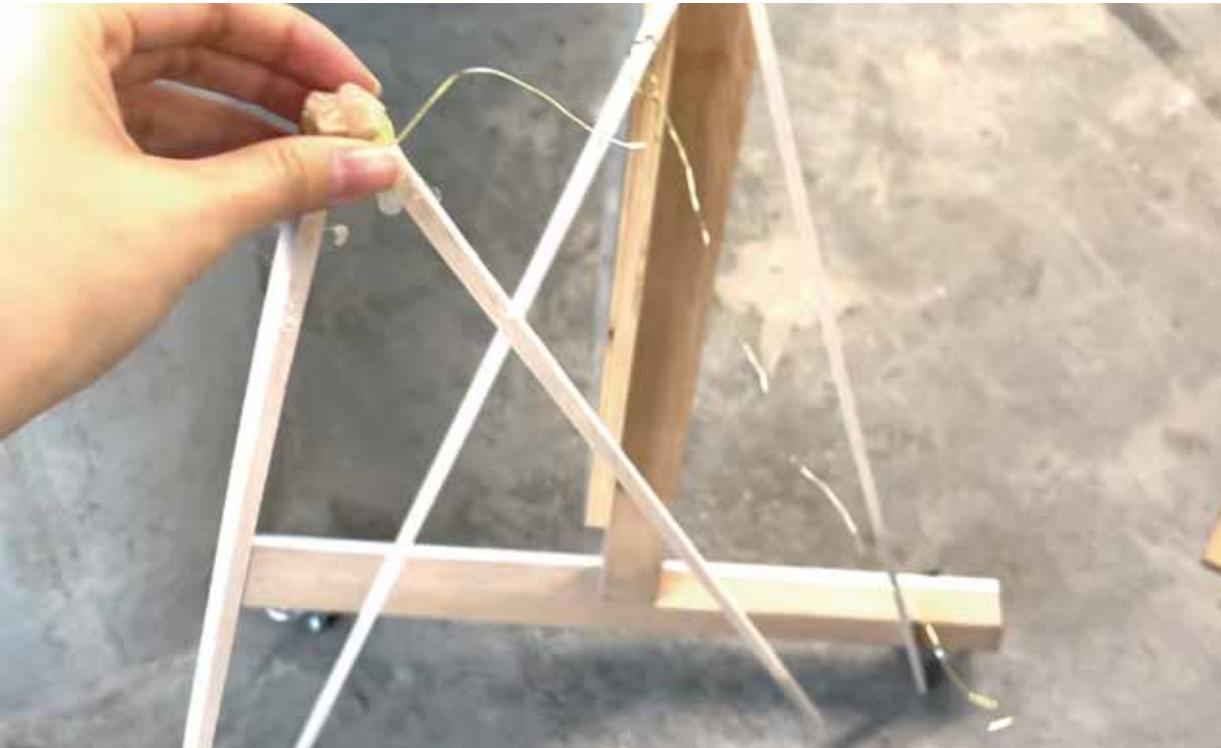


FUNCTION TWO
PRINCIPLE

Distance control



PRODUCTION PROCESS



EXHIBITION BOOTH PRODUCTION

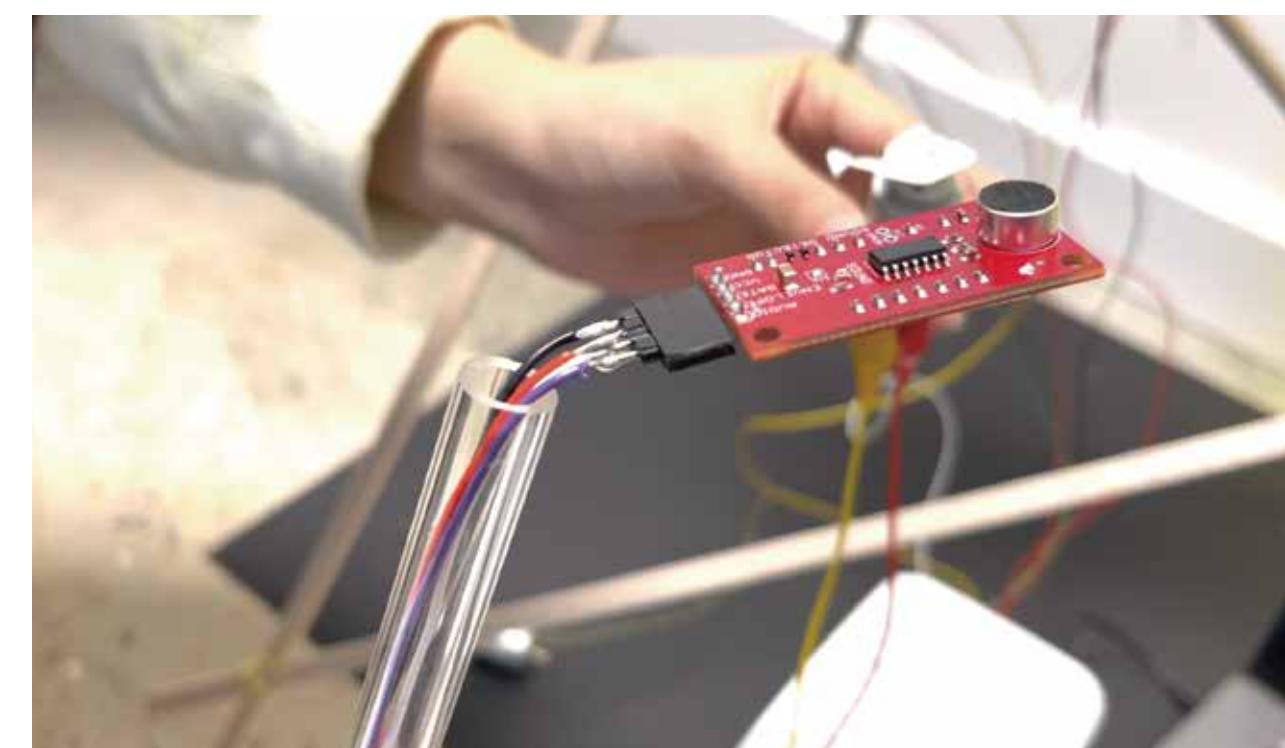
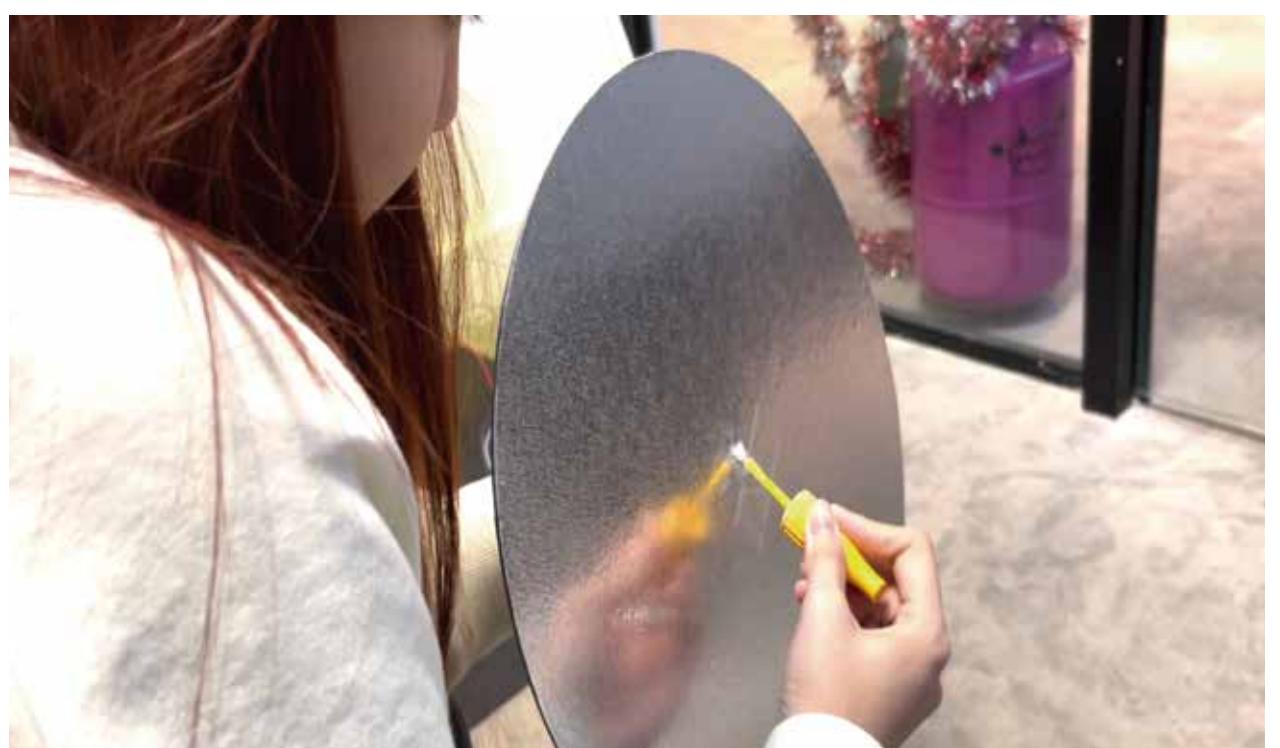
Materials: wooden stick, iron wire, acrylic board, hot melt adhesive, electric saw

TURNTABLE MAKING

Materials: Paper shell, scissors, super glue, motor

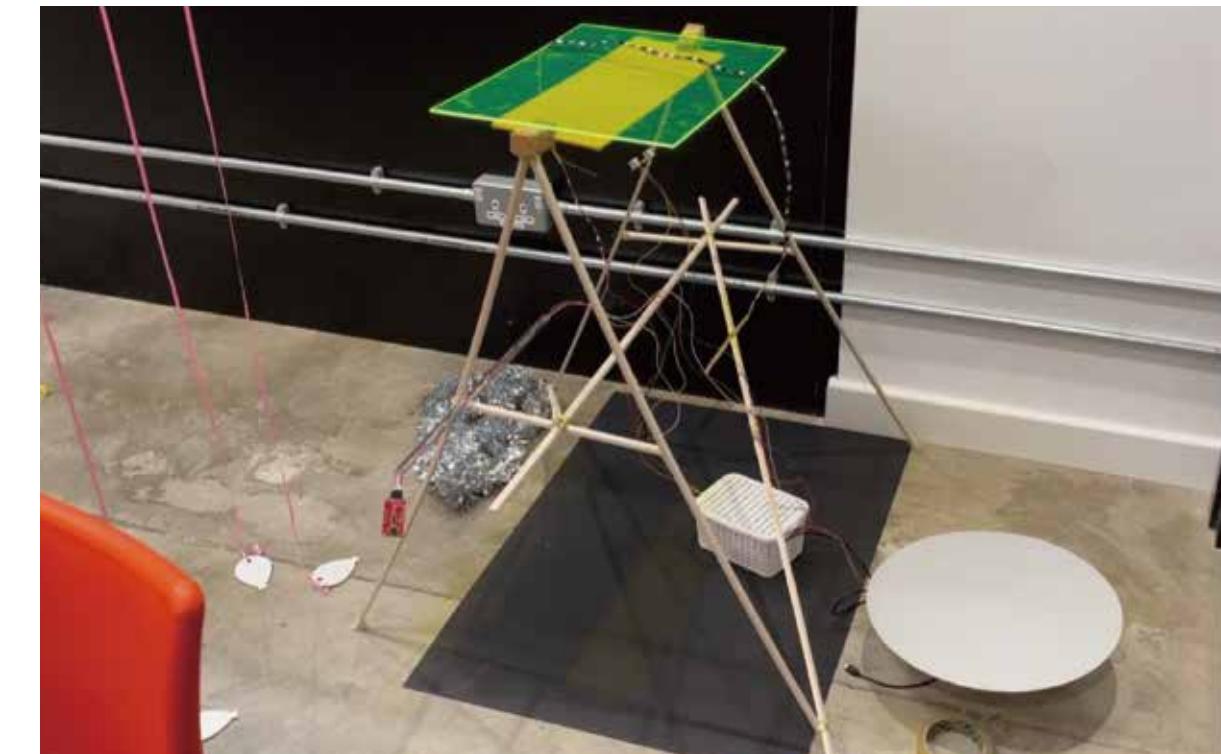
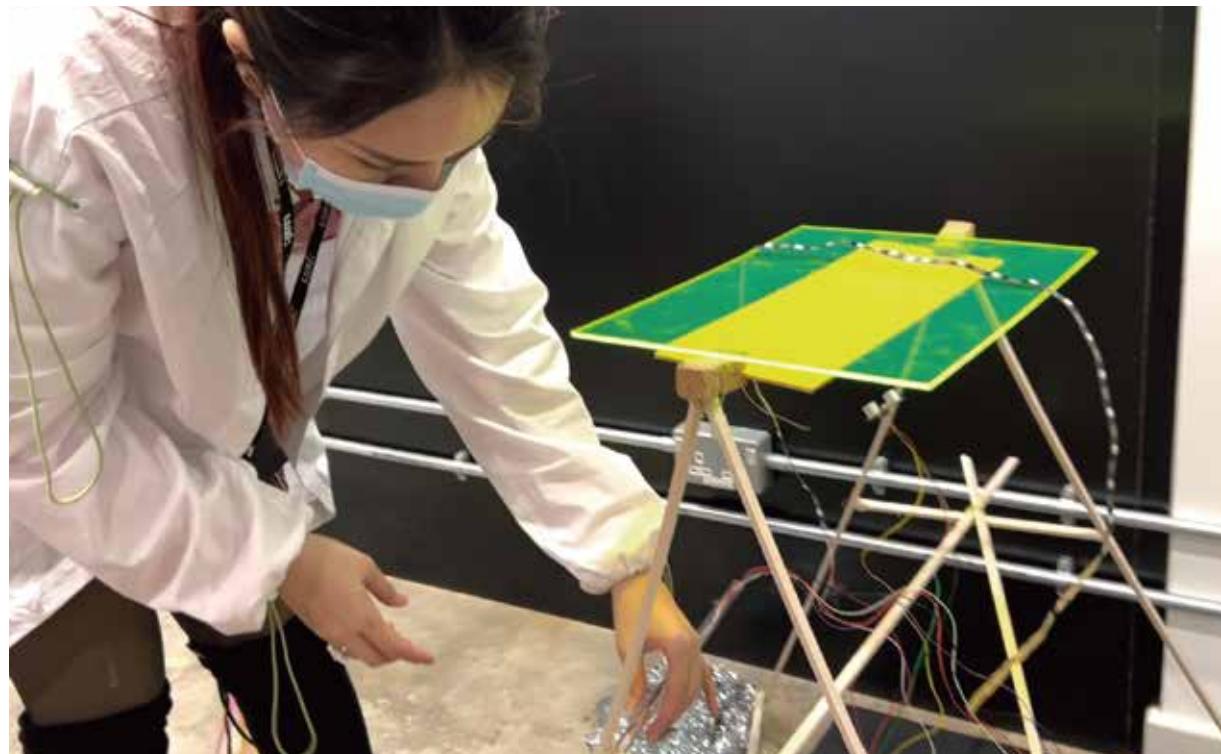
HANDHELD MICROPHONE PRODUCTION

Materials: Transparent hollow plastic tube, microphone sensor



PRODUCTION PROCESS

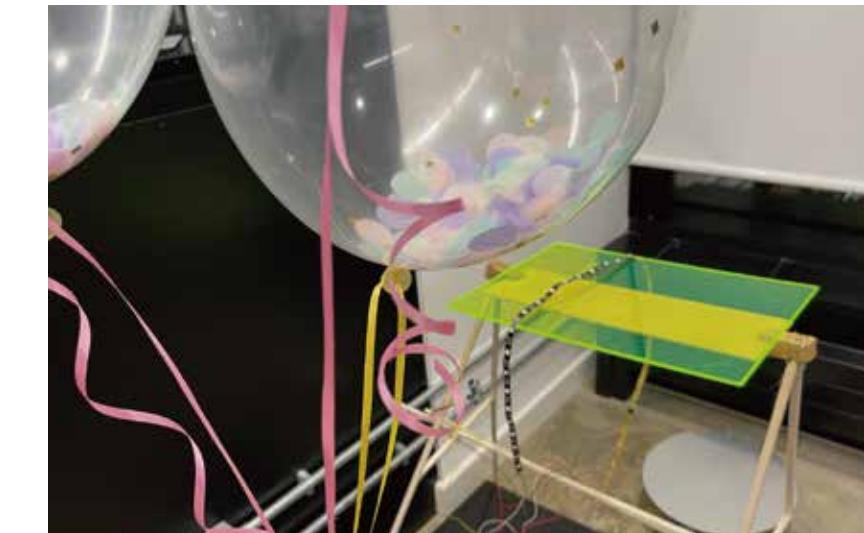
ASSEMBLE PARTS



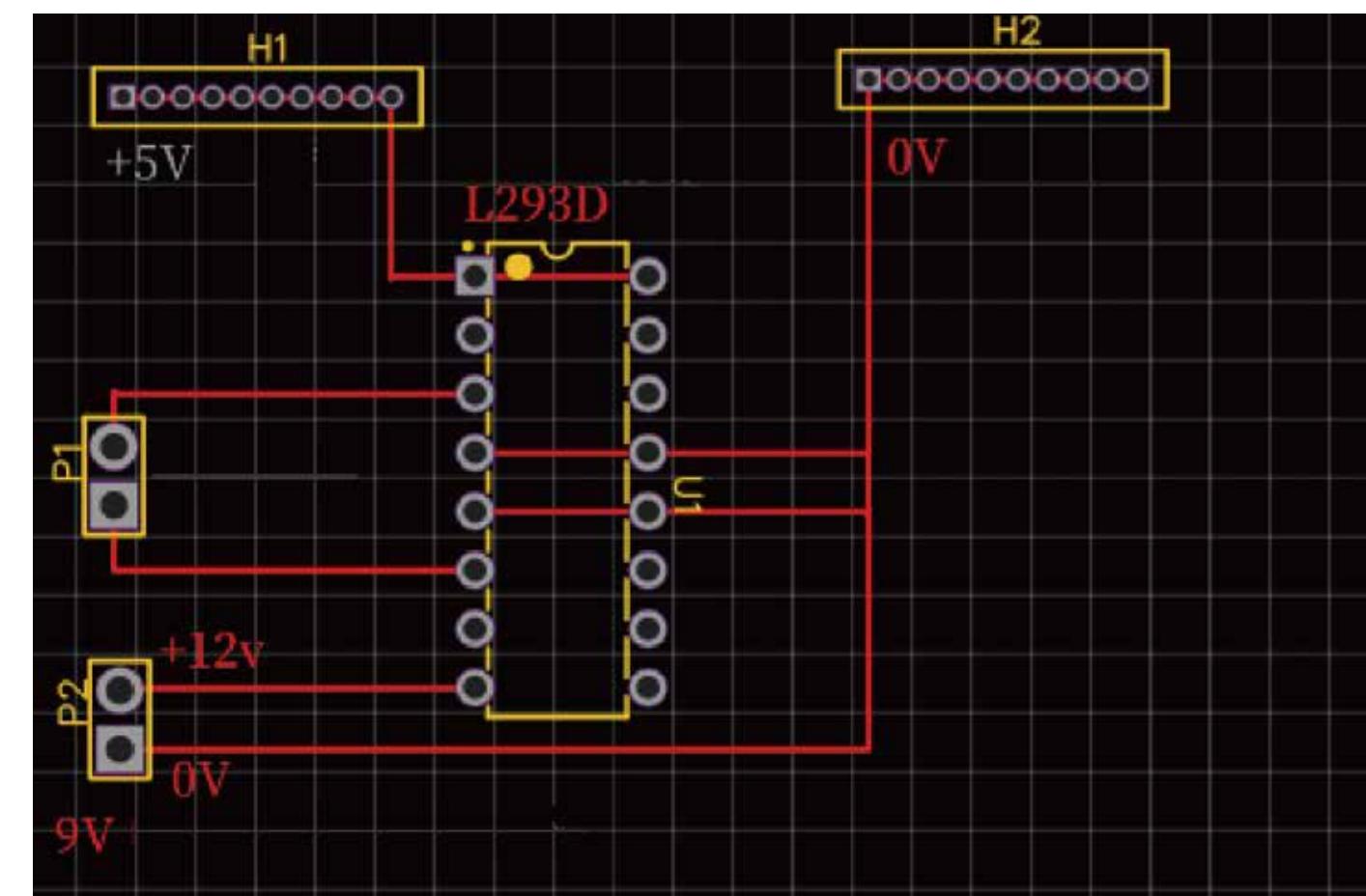
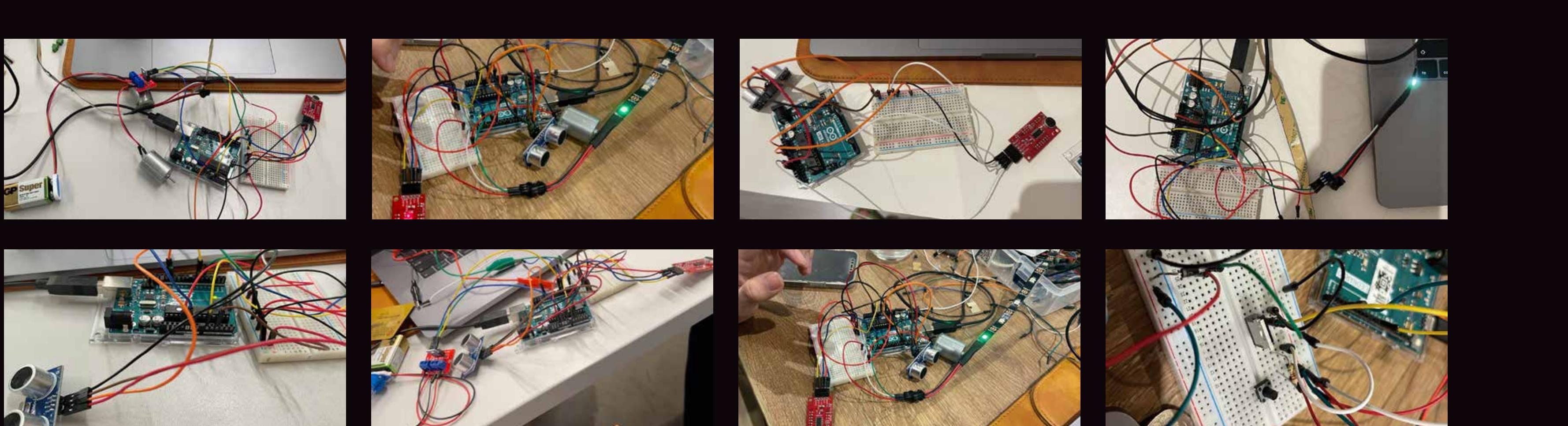
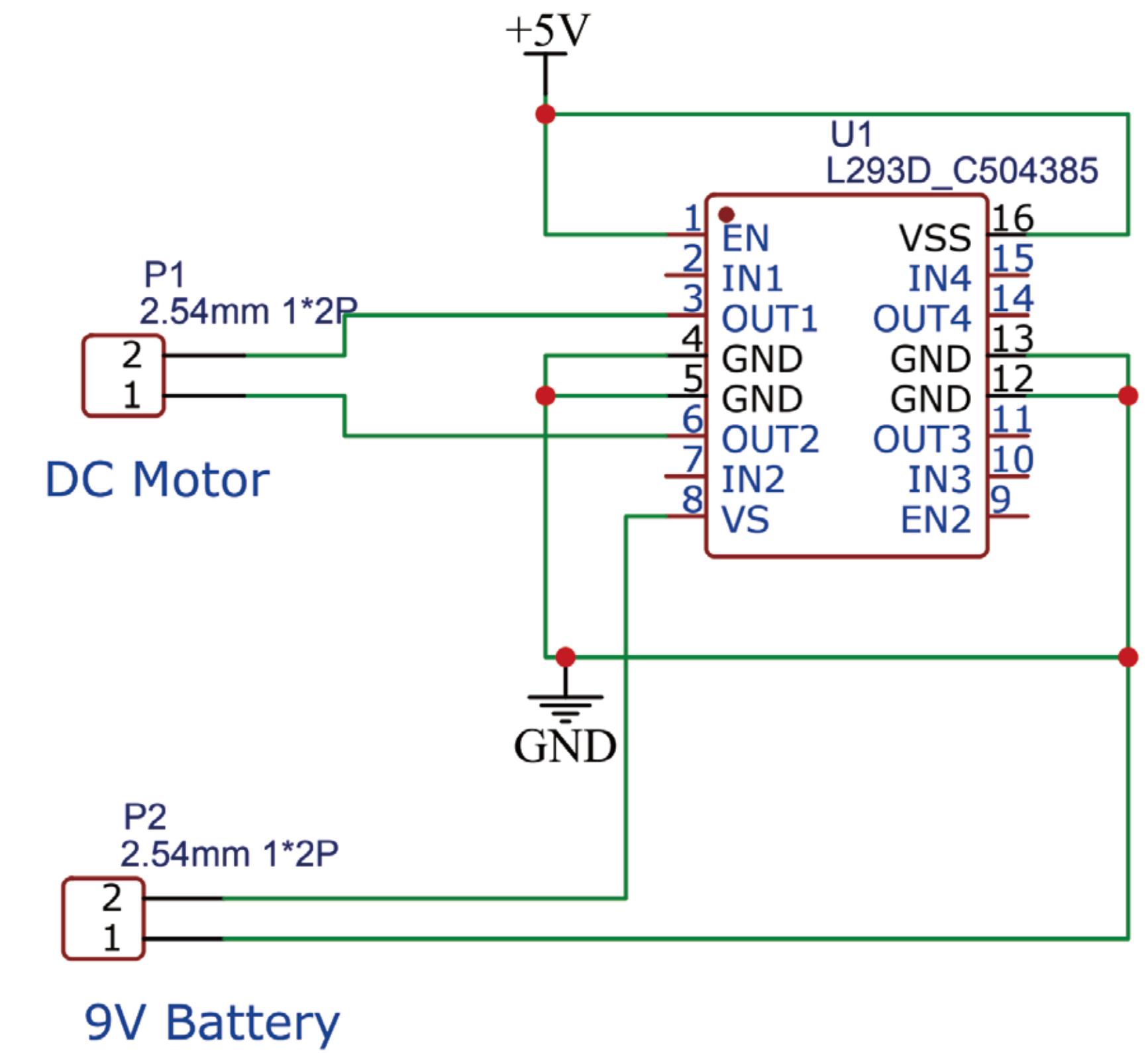
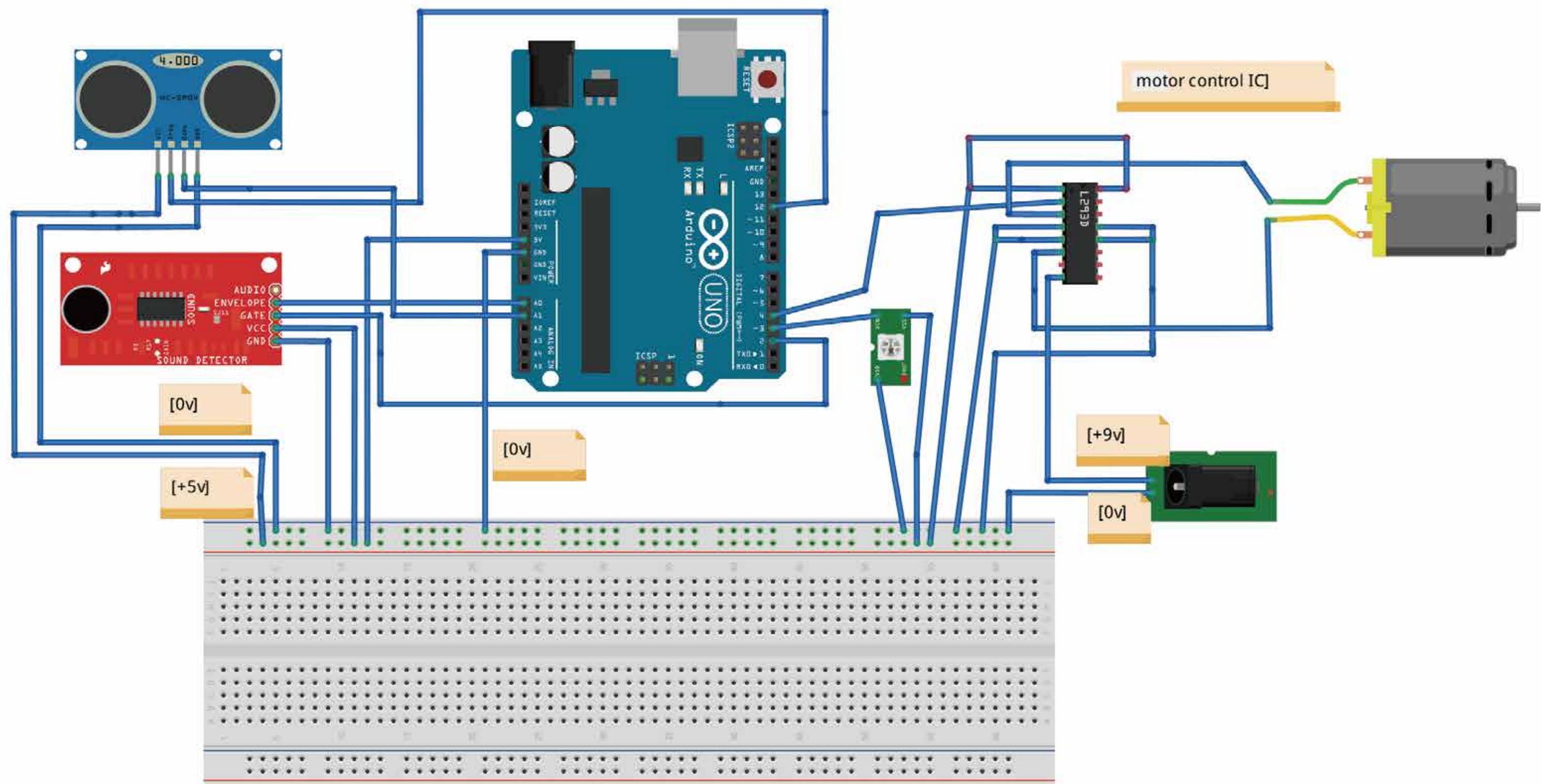
EXHIBITION



**NOW WE CAN FEEL THE
PARALLEL WORLD AND
COMMUNICATE WITH THE
CREATURES IN IT !**



CIRCUIT DIAGRAM



CODING PART

code20211124 | Arduino 1.8.17 Hourly Build 2021/09/06 02:34

```

code20211124 §

*****  

* sound_distance.ino  

* Connections:  

* The Sound Detector is connected to the Adrduino as follows:  

* (Sound Detector -> Arduino pin)  

* GND -> GND  

* VCC -> 5V  

* Gate -> Pin 2  

* Envelope -> A0  

* *****  

#include <Adafruit_NeoPixel.h>  

#ifndef __AVR__  

#include <avr/power.h> // Required for 16 MHz Adafruit Trinket  

#endif  

// Define hardware connections  

#define PIN_GATE_IN 2  

#define IRQ_GATE_IN 0  

#define PIN_LED_OUT 13  

#define PIN_MOTOR_OUT 6 //  

#define PIN_ANALOG_IN A0  

int VoiceVal=0;  

int VoiceMAX=550;  

int NoiseVal=40;  

// 控制 WS2812 灯条的引脚编号  

#define PIN 5  

// 定义控制的 LED 数量  

#define NUMPIXELS 6//16  

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

// 相邻 LED 之间的延迟, 单位毫秒  

#define DELAYVAL 500  

const int trigPin = 12;  

const int echoPin = 15; //A2  

//define sound velocity in cm/uS  

#define SOUND_VELOCITY 0.034  

#define CM_TO_INCH 0.393701  

long duration;  

float distanceCm;

```

code20211124 §

```

if(distanceCm >= 400)
{
  DistanceVal = 0; // >4m
  NUMPIXELS=6;
  for(int i=0; i<NUMPIXELS; i++) { // For each pixel...
    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    pixels.show(); // Send the updated pixel colors to the hardware.
    delay(DELAYVAL); // Pause before next pass through loop
  }
}

else if( (distanceCm > 300) && ( distanceCm <= 400 ) )
{
  NUMPIXELS=5;
  DistanceVal = map(distanceCm, 200,300, 0, 255); // 0~1m; // 3~4m
  pixels.setPixelColor(0, pixels.Color(DistanceVal,0 , 0));
  pixels.show(); // Send the updated pixel colors to the hardware.
  delay(DELAYVAL); // Pause before next pass through loop
}

else if( (distanceCm > 200) && ( distanceCm <=300 ) )
{
  NUMPIXELS=4;
  DistanceVal = map(distanceCm, 200,300, 0, 255); // 0~1m; // 3~4m
  pixels.setPixelColor(1, pixels.Color(0, DistanceVal, 0));
  pixels.show(); // Send the updated pixel colors to the hardware.
  delay(DELAYVAL); // Pause before next pass through loop
}

else if( (distanceCm > 100) && ( distanceCm <=200 ) )
{
  NUMPIXELS=3;
  DistanceVal = map(distanceCm, 100,200, 0, 255); // 0~1m; // 3~4m
  pixels.setPixelColor(2, pixels.Color(0,0,DistanceVal));
  pixels.show(); // Send the updated pixel colors to the hardware.
  delay(DELAYVAL); // Pause before next pass through loop
}

else if( (distanceCm > 20) && ( distanceCm <=100 ) )
{
  NUMPIXELS=2;
  DistanceVal = map(distanceCm, 0,20, 0, 255); // 0~1m; // 3~4m
  for(int i=0; i<NUMPIXELS; i++) { // For each pixel...
    // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
    // Here we're using a moderately bright green color:
    pixels.setPixelColor(i, pixels.Color(DistanceVal,DistanceVal, 0));
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance
    distanceCm = duration * SOUND_VELOCITY/2;

    // Prints the distance on the Serial Monitor
    Serial.print("Distance (cm): ");
    Serial.println(distanceCm);
    /*
    if( distanceCm >20 )
    {
      distanceCm=20;
    }
    else{
      DistanceVal = map(distanceCm, 0,20, 0, 255); // 0~20cm only
      for(int i=0; i<NUMPIXELS; i++) // For each pixel...
        // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
        // Here we're using a moderately bright green color:
        pixels.setPixelColor(i, pixels.Color(DistanceVal,0 , 0));
        pixels.show(); // Send the updated pixel colors to the hardware.
        delay(DELAYVAL); // Pause before next pass through loop
    }
    */
    // Convert to inches
    /*
    distanceInch = distanceCm * CM_TO_INCH;
    Serial.print("Distance (inch): ");
    Serial.println(distanceInch);
    */
    // pixels.clear(); // Set all pixel colors to 'off'
  }
}

```

code20211124 §

```

// Display status
Serial.println("Initialized");
}

void loop()
{
  int VoiceValue;

  // Check the envelope input
  VoiceValue = analogRead(PIN_ANALOG_IN);
  // Convert envelope value into a message
  Serial.println("Voice Value: ");
  Serial.println(String(VoiceValue));
  if(VoiceValue >= VoiceMAX) {
    VoiceValue = VoiceMAX;
    analogWrite(PIN_MOTOR_OUT, 255);
  }else if(VoiceValue <= NoiseVal){
    VoiceValue = 0;
    analogWrite(PIN_MOTOR_OUT, 0);
  }else {
    VoiceVal = map(VoiceValue, 20,550, 0, 255); //
    analogWrite(PIN_MOTOR_OUT, VoiceVal);
  }

  /*
  if(value < 10)
  {
    Serial.println("Quiet.");
  }
  else if( (value > 10) && ( value < 30) )
  {
    Serial.println("Moderate.");
  }
  else if(value > 30)
  {
    Serial.println("Loud.");
  }
  */

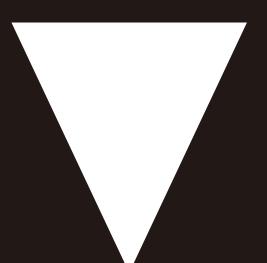
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance
  distanceCm = duration * SOUND_VELOCITY/2;
}

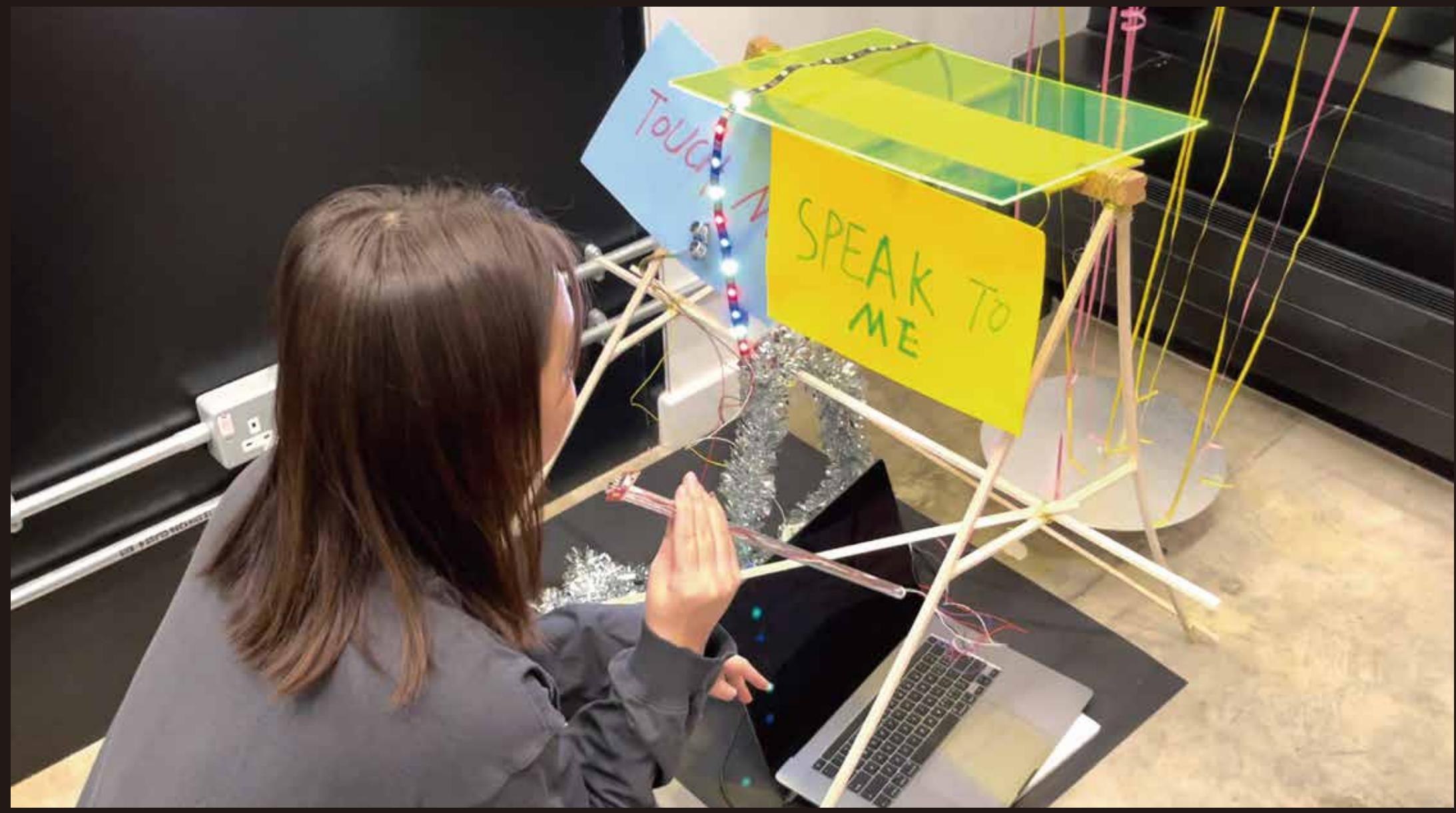
```

LINK:



Final

Video Link : <https://youtu.be/c5zaE4mR4nI>



Tell what challenges you overcame

The speed of the motor is affected by the volume. However, there are many ambient sounds and noises on site. I need to adjust the parameter value according to the decibels on site so that the speed can reach the expected value.

The link between the motor and the rotary table is a big problem, because it is difficult to paste a cardboard on a rotating fine iron. Whenever the motor speed becomes faster, the motor and cardboard will fall off. I think it may be better to change some advanced materials under the appropriate budget, and the problem will be solved.

When the ultrasonic sensor detects the distance, it must ensure that there is nothing within the distance. However, the school site is too small to receive interference. I can only put the sensor up, but I think it affects the visual effect of the device.

What are the future possibilities for this project

I think this device can participate in major art exhibitions. I think it is a very cool thing to communicate and interact with people in parallel worlds. If this concept is extended and replaced with more beautiful and advanced shell decoration with sufficient budget and time, it will be a perfect interactive art device exhibition.

The information I want to convey is the communication between people and the future, and our major is based on the existence of technology to promote social progress. I hope more people can see the possibility of technology and creativity and the future full of sense of science and technology.