

A light painting photograph in a dark environment. A bright, multi-colored beam of light, primarily red, orange, yellow, green, and blue, is cast upwards from a low-angle perspective. The beam is slightly curved and appears to originate from a small, glowing rectangular opening on the floor. The surrounding area is completely black, making the vibrant colors stand out.

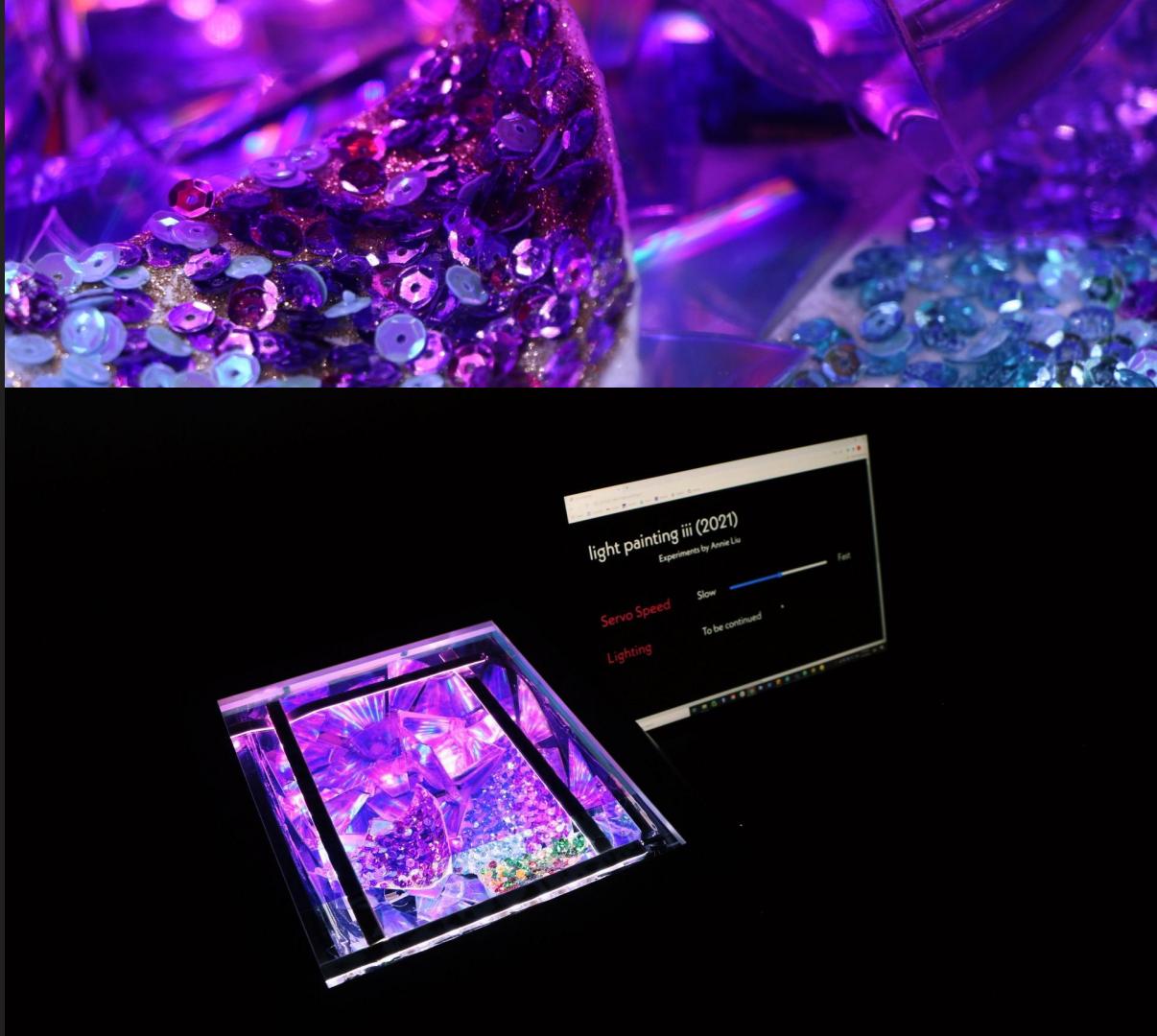
light painting iii

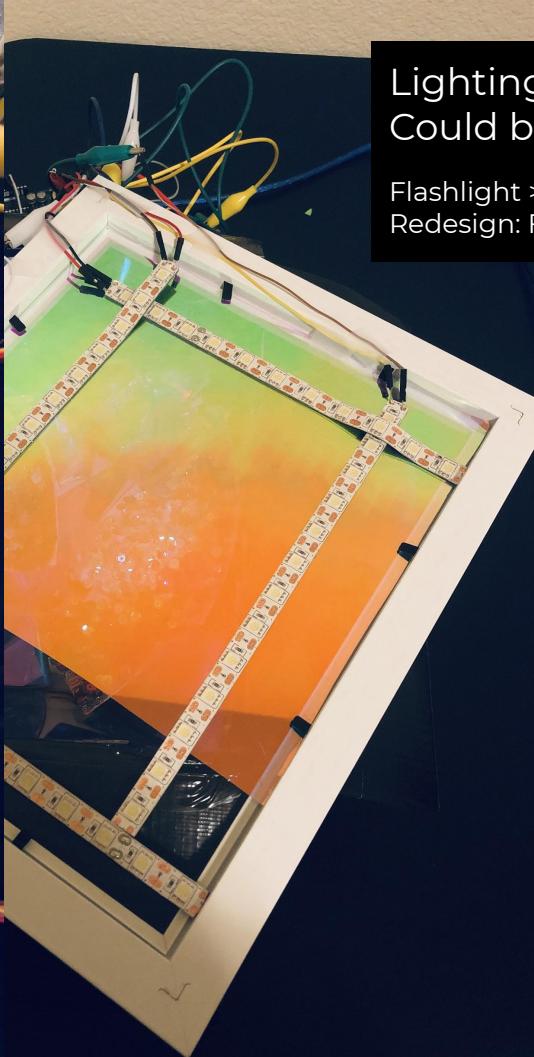
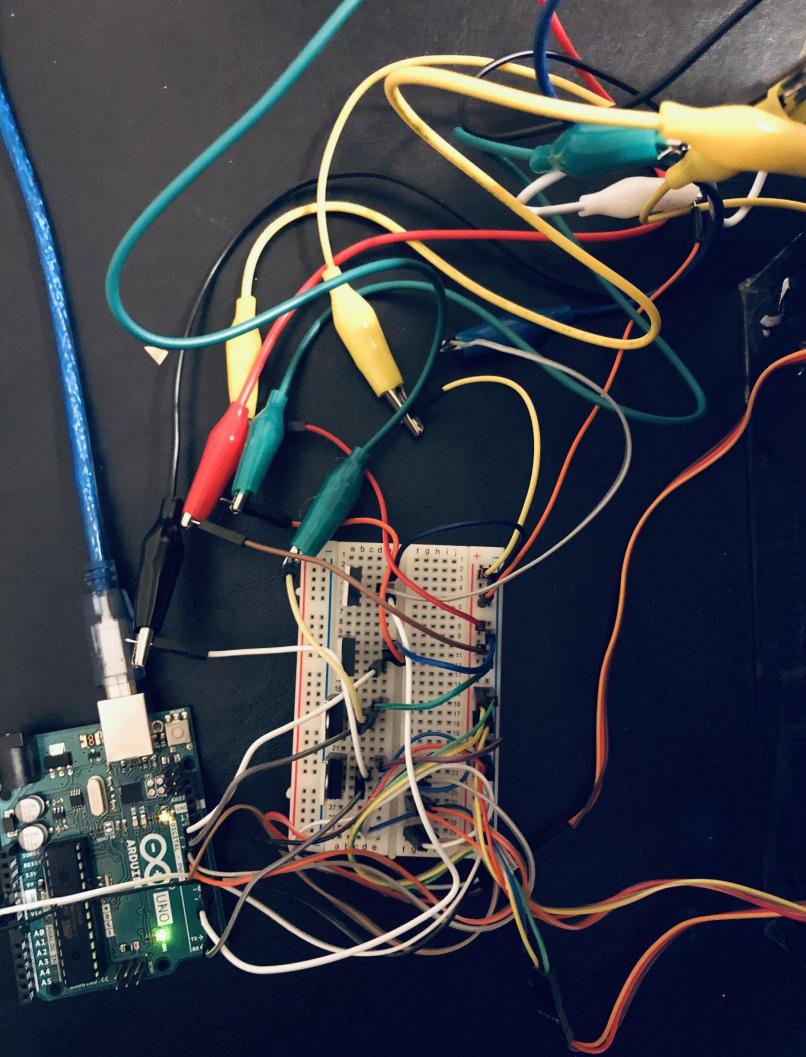
# Concept

Making an interactive website to control the speed of the servos in my light painting instrument.

Original Idea:

Making an interactive website to control servo speed and lighting.

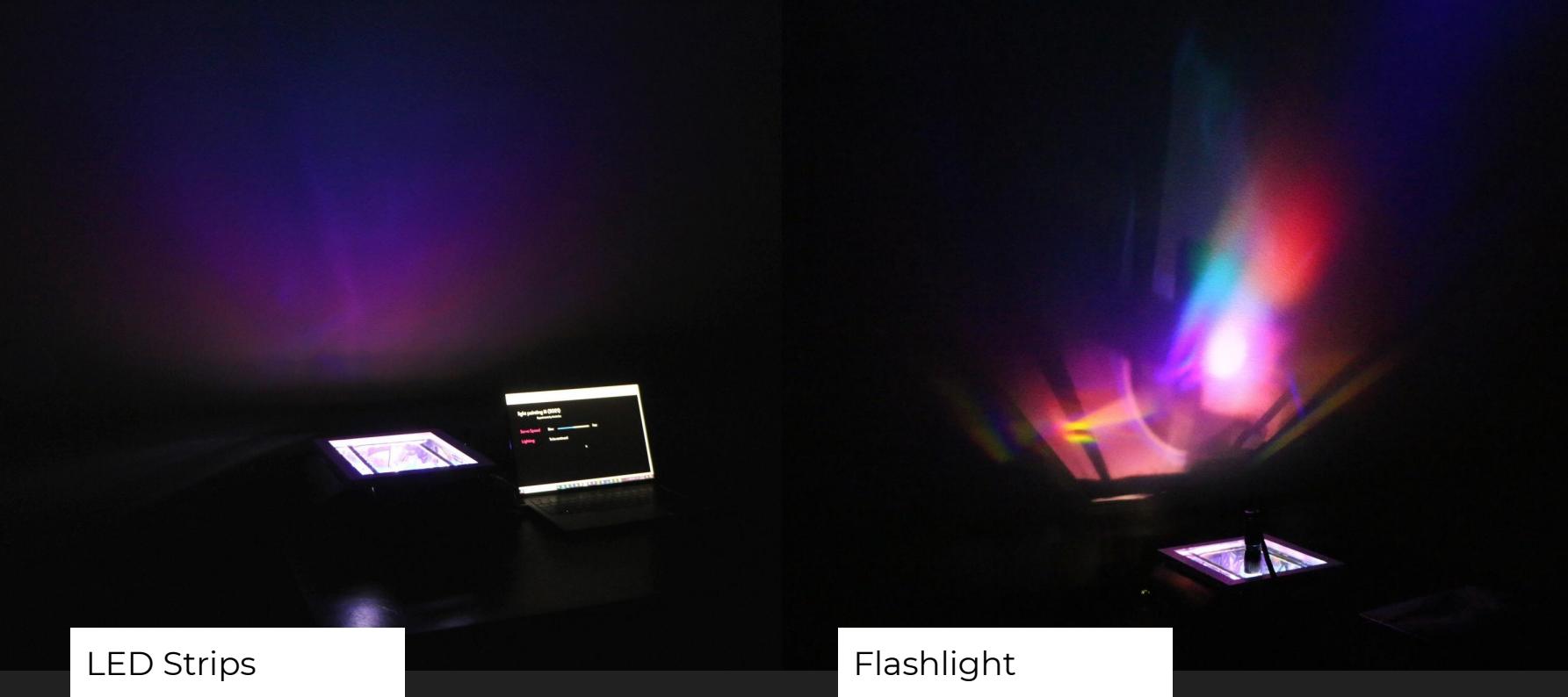




Lighting ---> Tried it, Was Cool, But Could be Cooler

Flashlight > LED Strip  
Redesign: Flashlight Machine



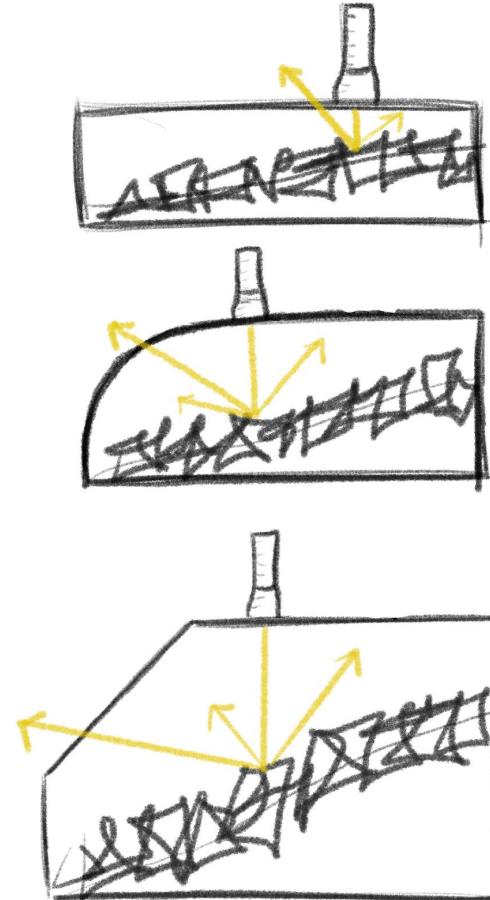
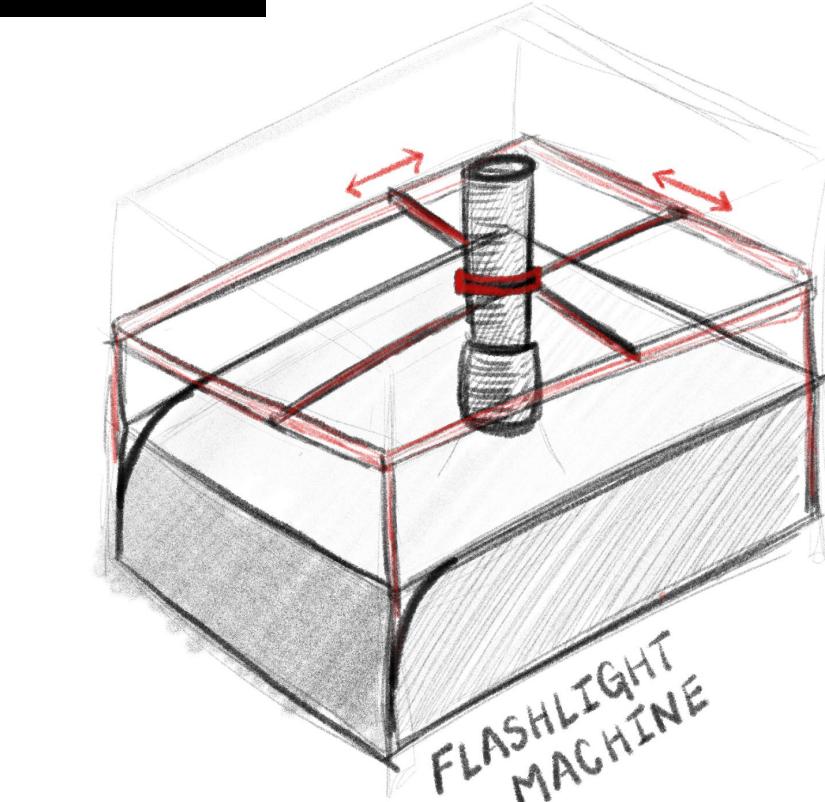


LED Strips

Flashlight

look at that difference !!

## Redesign Sketch





the LED strips made the instrument  
itself look cool though !

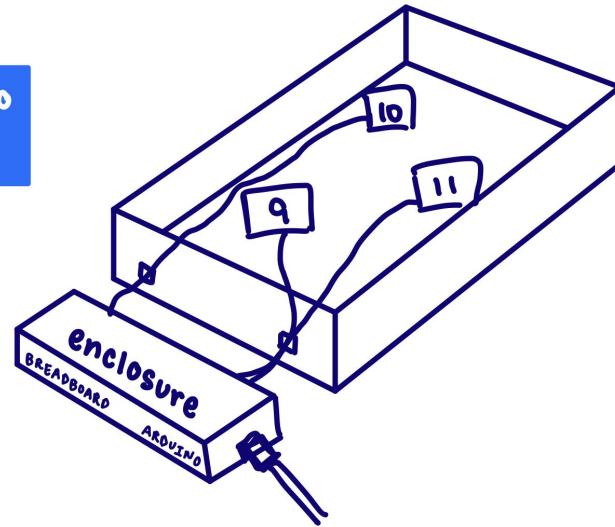
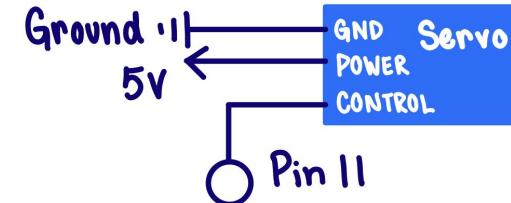
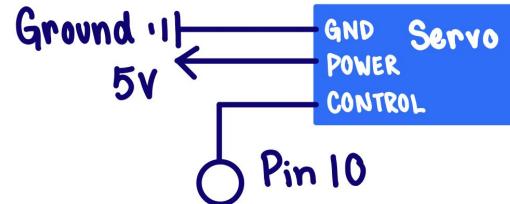
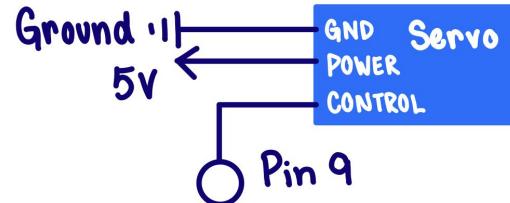


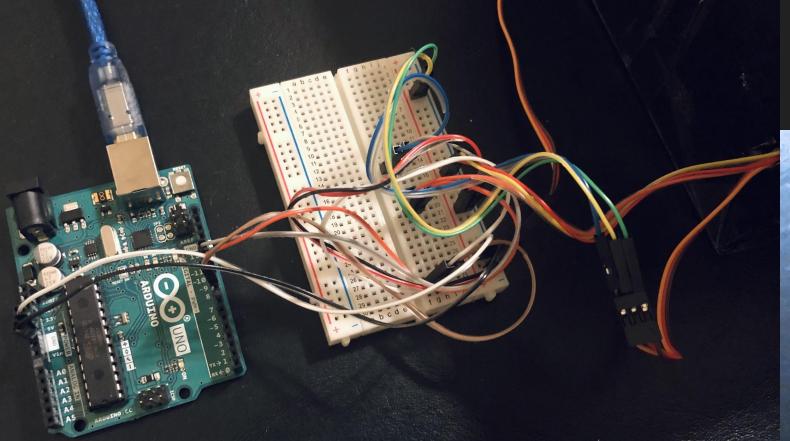
# Schematic

Materials:

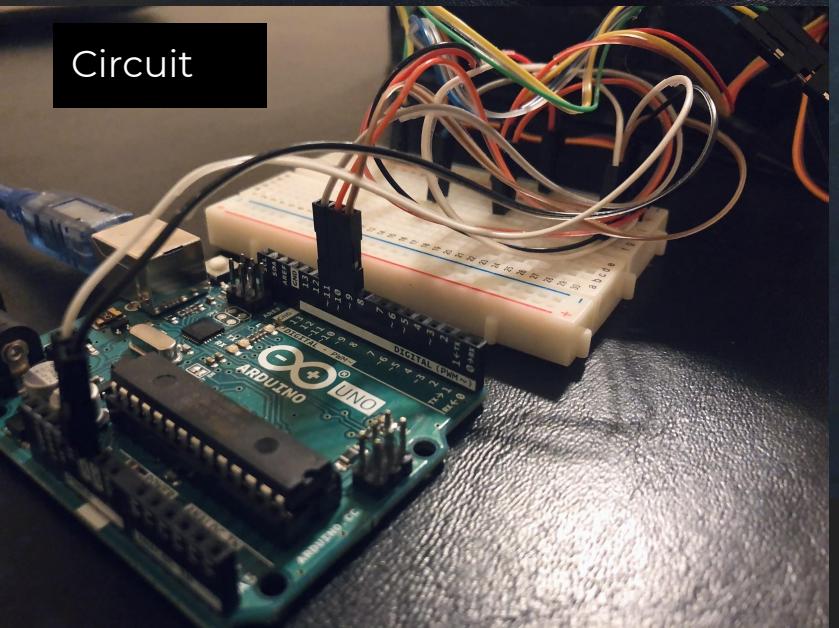
Arduino  
Servos

CDs  
Sequins  
Cardboard  
Paper Plates  
Glitter  
Foam  
Glass Sheet  
Frame  
Dichroic Film  
Glue  
Flashlight





Circuit



Enclosure

# light painting iii (2021)

Experiments by Annie Liu

Servo Speed

Slow



Fast

Lighting

To be continued

## Code

```
15 var serial;           // variable to hold an instance of the serialport library
16 var portName = 'COM5'; // rename to the name of your port
17 var inData;           // variable to store data
18 let val = 3;          // val is the servo speed setting value
19
20
21 function setup() {
22     serial = new p5.SerialPort();           // make a new instance of the serialport library
23     serial.on('list', printList);           // set a callback function for the serialport list event
24     serial.on('connected', serverConnected); // callback for connecting to the server
25     serial.on('open', portOpen);            // callback for the port opening
26     serial.on('data', serialEvent);         // callback for when new data arrives
27     serial.on('error', serialError);        // callback for errors
28     serial.on('close', portClose);          // callback for the port closing
29
30     serial.list();                      // list the serial ports
31     serial.open(portName);               // open a serial port
32
33     createCanvas(2000, 1000);             // create a 2000x1000 createCanvas
34     textSize(50);                     // making the text size large
35     textAlign("center");               // this is the font I want to use
36
37     sliderServo = createSlider (1, 5, 3); // create a slider with min val 1, max val 5, and default val 3
38     sliderServo.position(505,280);       // position of the slider (x,y)
39     sliderServo.style('width', '350px');   // width of the slider
40 }
41
42 // This is how we communicate to the Arduino.
43 function servoSlider_Arduino() {
44     serial.write(val);                  // write the value of the servo speed setting to serial
45 }
```

## Code

```
79  function draw() {
80      background(0);                                // black background
81      fill(255);                                    // white text
82      textSize(50);                                // Large text
83      text("light painting iii (2021)", 100, 120); // project title
84      textSize(25);                                // small text
85      text("Experiments by Annie Liu", 300, 170); // Light painting experiments by me !
86
87      textSize(40);                                // Larger text
88      fill(220, 46, 118);                           // pink text
89      text('Servo Speed', 100, 300);                // Servo Label
90      text('Lighting', 100, 400);                  // Lighting label
91
92      fill(255);                                    // white text
93      textSize(30);                                // smaller text
94      text('Slow', 400, 297);                      // slider Label
95      text('Fast', 900, 297);                      // slider Label
96      text('To be continued', 400, 400);           // Lighting Label
97      let currentVal = sliderServo.value();          // make currentVal the value of the servo slider
98      if (currentVal != val) {                      // if the currentVal isn't equal to val (that means val changed),
99          val = currentVal;                         // reassign val's value to currentval
100         servoSlider_Arduino();                   // tell Arduino to change speed of servo through serial input
101     }
102 }
```

## Code

```
10 #include <Servo.h>          // Include this library.  
11  
12 Servo servo;                // CD servo  
13 Servo servo2;               // sequin servo  
14 Servo servo3;               // sequin servo  
15  
16 // Variables  
17 int servo_delay = 20;       // Default delay.  
18 int servo2_delay = 20;       // Default delay.  
19 int servo3_delay = 20;       // Default delay.  
20  
21 // Set up !  
22 void setup() {  
23     Serial.begin(9600);       // Begin running serial monitor.  
24     Serial.setTimeout(10);    // Set the timeout for parseInt.  
25     servo.attach(9);         // Attach CD servo to pin 9.  
26     servo2.attach(10);        // Attach sequin servo to pin 10.  
27     servo3.attach(11);        // Attach CD servo to pin 11.  
28     servo.write(0);          // Send servo to position 0.  
29     servo2.write(0);         // Send servo to position 0.  
30     servo3.write(0);         // Send servo to position 0.  
31 }
```

## Code

```
52  
53 // Repeat !  
54 void loop() {  
55     // Send servos from position 0 to position 180.  
56     for(int i=0; i<180; i++){  
57         webSpeed();           // Update servo speed (the delays used).  
58         servo.write(i);      // Send servo to position i.  
59         delay(servo_delay); // Delay for servo_delay ms.  
60         servo2.write(i);    // Send servo2 to position i.  
61         delay(servo2_delay); // Delay for servo2_delay ms.  
62         servo3.write(i);    // Send servo3 to position i.  
63         delay(servo3_delay); // Delay for servo3_delay ms.  
64     }  
65  
66     // Send servos from position 180 to position 0.  
67     for(int i=180; i>=0; i--){ //  
68         webSpeed();           // Update servo speed (the delays used).  
69         servo.write(i);      // Send servo to position i.  
70         delay(servo_delay); // Delay for servo_delay ms.  
71         servo2.write(i);    // Send servo2 to position i.  
72         delay(servo2_delay); // Delay for servo2_delay ms.  
73         servo3.write(i);    // Send servo3 to position i.  
74         delay(servo3_delay); // Delay for servo3_delay ms.  
75     }  
76 }  
77
```

## Code

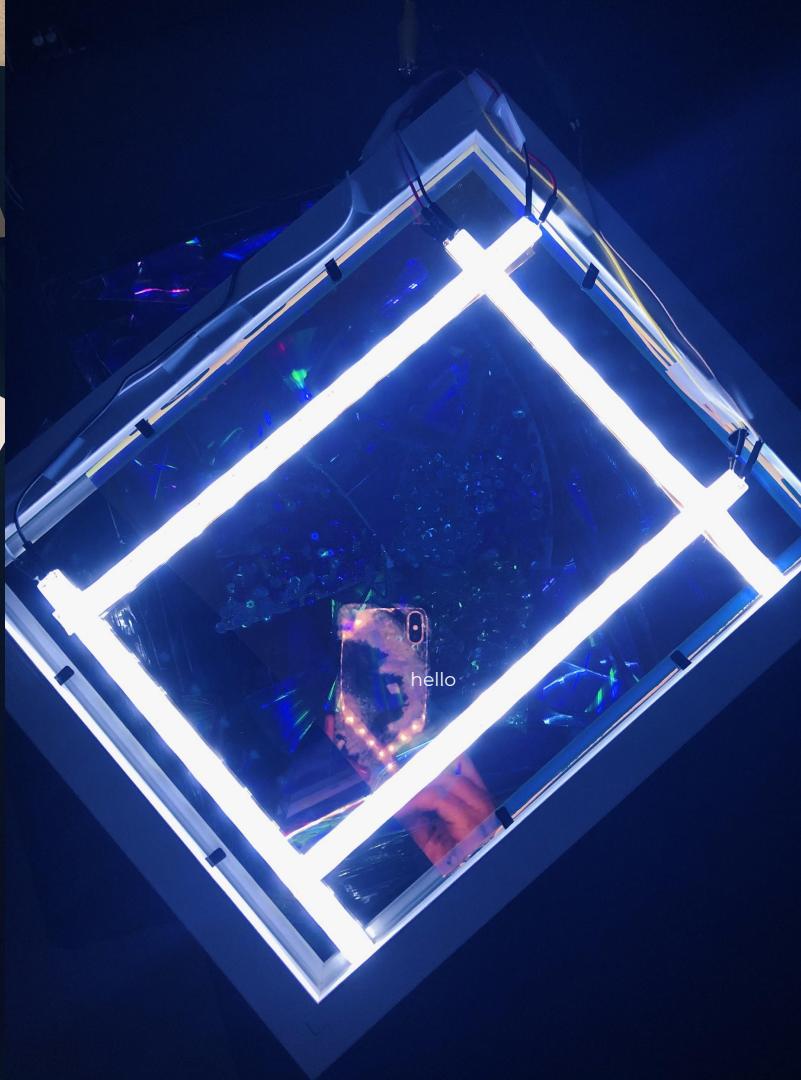
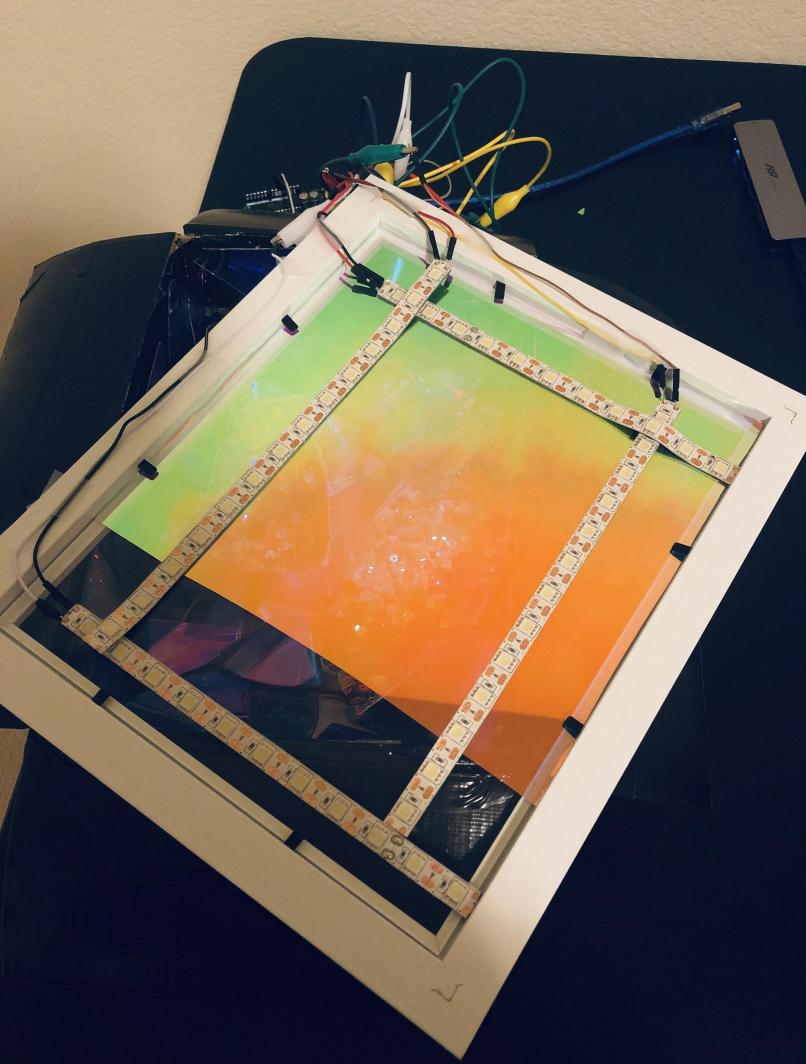
```
57
58 // Update the servo speed with input from the web.
59 void webSpeed() {
60     if(Serial.available() > 0) { // If there's serial data.
61         int inByte = Serial.read(); // Read it.
62         Serial.write(inByte); // Write it back to serial.
63         if (inByte == 1) { // 1 -> Slow Setting
64             // Set all delays to 40 ms.
65             servo_delay = 40;
66             servo2_delay = 40;
67             servo3_delay = 40;
68         } else if (inByte == 2) { // 2 -> Slow-Medium Setting
69             // Set all delays to 30 ms.
70             servo_delay = 30;
71             servo2_delay = 30;
72             servo3_delay = 30;
73         } else if (inByte == 3) { // 3 -> Medium Setting
74             // Set all delays to 20 ms.
75             servo_delay = 20;
76             servo2_delay = 20;
77             servo3_delay = 20;
78         } else if (inByte == 4) { // 4 -> Medium-Fast Setting
79             // Set all delays to 10 ms.
80             servo_delay = 10;
81             servo2_delay = 10;
82             servo3_delay = 10;
83         } else if (inByte == 5) { // 5 -> Medium-Fast Setting
84             // Set all delays to 5 ms.
85             servo_delay = 5;
86             servo2_delay = 5;
87             servo3_delay = 5;
88         }
89     }
90 }
```



Demo

more photos + videos















look at that !!!

servo says hi

Close Up