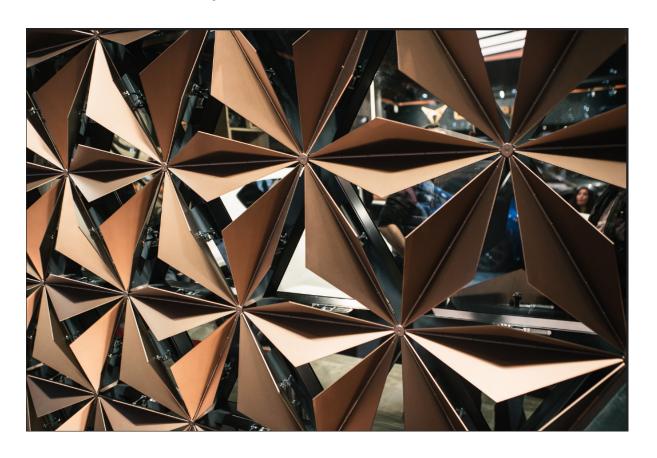
Proyecto arte cinético



Indice

1.Descripción	3
2.Electrónica	3
3.Montaje	4
4.Programación	5
5.Incidencias	7
1.Incidencia	7
2.Incidencia	7
6.Materiales	7
7.Enlaces de interés	8

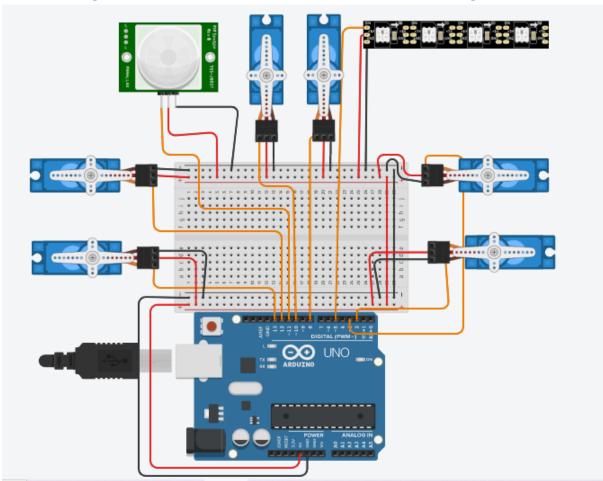
1.Descripción

Vamos a realizar un proyecto de arte cinético en conjunto con el bachillerato de artes, donde la idea principal se trata de un hexágono que tendrá un dibujo representado algo bueno y cuando se abra el hexágono, se vea otro hexágono detrás con un dibujo que represente algo malo. La idea es hacerlo relacionado con el tema de redes sociales y ciberbullying donde nosotros vamos a realizar la parte mecánica y electrónica, y los de arte la parte artística.

Mi idea para el hexágono es dividirlo en 6 triángulos curvos y cuando se pase por delante de él se abra uno a uno creando un efecto de ola y se vea el hexágono de atrás. También se añadirá al hexágono una tira de luces leds para que se ilumine el hexágono en verde cuando se vea la parte buena y en rojo la parte mala.

2.Electrónica

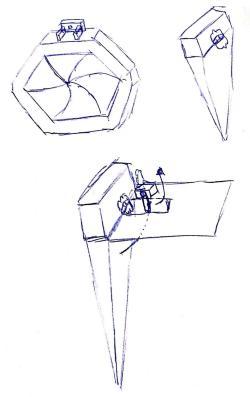
Con Tinkercad vamos a realizar el esquema electrónico y ver el conexionado,hemos utilizado 6 servomotores cada uno de ellos con su horn, un sensor PIR, una tira de leds y se alimentara el arduino con una batería powerbank. A continuación vemos como se conectaria cada componente al arduino:



3.Montaje

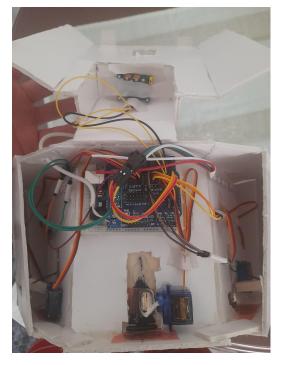
Cada triángulo estará unido a la estructura con una bisagra para posibilitar el giro de estos, y el mecanismo es muy sencillo se fijará un cable un extremo al triángulo y el otro extremo al horn del servomotor, este girara 180 grados y hará el movimiento de apertura y cierre. El sensor PIR se situará en la parte de arriba de la estructura y la tira de leds en el interior en un fondo donde se encontrará todo el cableado y el arduino para que no quede visible.

A continuación vamos a ver unos bocetos de la idea:



Y esta es una foto del montaje final:





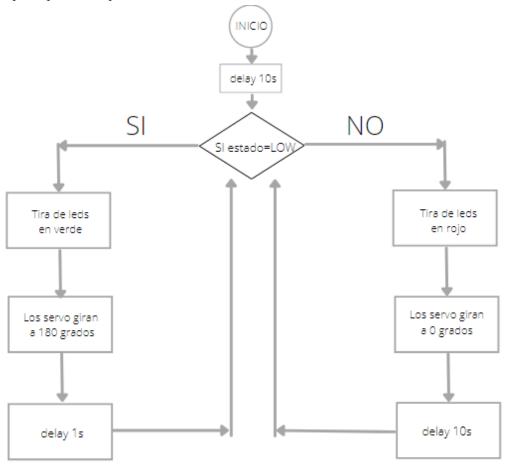
4.Programación

Código explicativo del programa:

```
#include <Adafruit_NeoPixel.h> // Incluímos la librería para la tira de leds
#include <Servo.h> // Incluímos la librería para poder controlar el servo
#define PINLEDRGB 0 //pin data 0
#define NUMPIXELS 40 //numero de leds en la tira
Servo motorl; // DECLARAR LOS 6 SERVO
Servo motor2;
Servo motor3;
Servo motor4:
Servo motor5;
Servo motor6;
int PIR=2; //pin de entrada PIR
int pirState=LOW; //de inicio no hay movimiento
int estado=0; //variable para asignar la lectura del sensor PIR
int leo=0; //variable para controlar el "for" de leds verde
Adafruit NeoPixel pixels (NUMPIXELS, PINLEDRGB, NEO GRB + NEO KHZ800);
void setup() {
pinMode(PIR,INPUT); //El pin 2 lo asignamos como entrada para la señal del sensor
 delay(10000); //delay de calibracion del PIR de 10 segundos
motorl.attach (4); // adjuntar cada servo a un pin
motor2.attach (6);
motor3.attach (5);
motor4.attach (13);
motor5.attach (9);
motor6.attach (8);
pixels.begin(); //Inicializamos el objeto "pixels"
pixels.clear(); //limpiamos el color anterior
void loop() {
 estado=digitalRead(PIR); //lee valor de la variable estado
 if (estado==LOW) { //si no detecta movimiento
if (estado==LOW && leo==0) { //si detecta movimiento y la variable leo es igual a 0 se enciende las luces led en verde
 pixels.clear(); //limpiamos el color anterior
for (int i = 0; i < NUMPIXELS; i++) {</pre>
    pixels.setPixelColor(i, pixels.Color(0, 50, 0)); // Brillo en verde
    pixels.show(); // Mostramos y actualizamos el color del pixel de nuestra cinta led RGB
```

```
delay(50); // tiempo con el que se enciende cada led de la tira
  leo=1; //igualamos la variable a 1 para que no vuelva a entrar y no entrar en bucle el encendido de leds
 }}
motor5.write(130); //se cierra los servomotores con un delay
delay(150);
motor4.write(145);
delay(150);
motor3.write (175);
delay(150);
motor2.write (175);
delay(200);
motorl.write(175);
delay(200);
motor6.write(180);
delay(1000);
if (pirState==HIGH) {    //si previamente estaba encendido
 pirState=LOW; //no hay movimiento
else{ //si detecta movimiento
pixels.clear(); //limpiamos el color anterior
for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(50, 0, 0)); // Brillo en rojo
    pixels.show(); // Mostramos y actualizamos el color del pixel de nuestra cinta led RGB
    delay(10); // tiempo con el que se enciende cada led de la tira
leo=0; //igualamos a 0 la variable para que pueda volver a entrar en el encendido de leds verde cuando se cierre
 motor6.write(0); //se abre los servomotores con un delay
delay(200);
motorl.write(0);
delay(200);
motor2.write (0);
delay(150);
motor3.write (0);
delay(150);
motor4.write (0):
delay(150);
motor5.write(0);
delay(10000);
if (pirState==LOW) { //si previamente estaba apagado
 pirState=HIGH; //hay movimiento
 111
```

Diagrama principal de bloques:



5.Incidencias

1.Incidencia

El sensor PIR tiene un amplio cono de detección entonces coloque 2 rectángulos de cartón del mismo material que la estructura al lado del sensor para hacerlo más direccional, y solo se active si pasas al frente de este (se puede ver en la foto del montaje).

2.Incidencia

El sensor pir requiere un tiempo de preparación para comenzar a tomar lecturas de forma adecuada. Esto se debe a que tiene que ocurrir las adaptaciones a las condiciones de operación del ambiente donde fue instalado. Durante ese tiempo del sensor "aprende" a reconocer el estado de reposo o el estado de movimiento del ambiente, por eso coloque un delay de 10 segundos en el setup().

6.Materiales

Cartón pluma, madera, 6 bisagras, compás, tijeras, cortadora de madera eléctrica, pistola de silicona caliente, cúter, cables, cinta aislante, 6 servomotores, sensor PIR, tira de 40 leds, arduino y bateria powerbank

7.Enlaces de interés

- $\hbox{-$\underline{$https://hetpro-store.com/TUTORIALES/sensor-pir-con-arduino/\#:$\sim:text=Calibraci\%C3\%B3n\%20del\%20sensor\%20PIR\&text=Durante\%20ese\%20tiempo\%20del\%20sensor,los\%2010\%20a\%2060\%20segundos.}$
- -https://www.youtube.com/watch?v=vUpl8OMknNg
- -https://www.youtube.com/watch?v=UbHiXc6DH8c
- -https://www.youtube.com/watch?v=t_FS993-tME