



Proyecto final: Brazo robotico para clasificador de colores

Datos principales

Universidad Autónoma de Baja California

Facultad de Ciencias

Nombre: Romo Jiménez Samuel Saul

Matricula: 364946

Carrera: Lic. en Ciencias Computacionales

Materia: Sistemas Empotrados

Profesora: Dra. Eloísa del Carmen García Canseco

Fecha: 3 de Junio de 2024

Objetivos

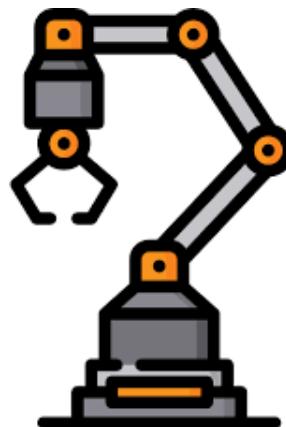
Algunos de los objetivos principales de este proyecto son los siguientes:

- Implementar los conceptos y herramientas aprendidas a lo largo del curso para el desarrollo de un proyecto con una aplicación real.
- Explorar las posibilidades de los actuadores para aplicaciones relacionadas con robótica industrial y control de procesos mediante la clasificación de colores.

Introducción

La motivación de este proyecto es la de adentrarse en la robótica y el control de procesos industriales mediante un proyecto simple con Arduino.

El proyecto está basado en 2 implementaciones que ya existen en internet. Una de ellas es el propio clasificador de colores para 4 tipos de colores, utilizando casi los mismos materiales. El otro es el brazo robótico, el cual posee una estructura más sólida debido a que está atornillado y soldado.



Materiales

Algunos de los materiales utilizados para el proyecto, junto con sus respectivos precios, son los siguientes:

Materiales estructurales:

- Madera balsa (200 pesos)
- Esferas de colores (20 pesos)



Materiales de Hardware:

- 5 servomotores SG90 (200 pesos)
- 4 potenciómetros 5 kOhms (32 pesos)
- Arduino nano/uno (650 pesos)
- Sensor de color TCS34725 (150 pesos)
- Driver de servomotores PCA9685 (270 pesos)
- Pantalla OLED 0.96 SSD (150 pesos)
- Jumpers, Breadboards, extensiones (100 pesos)



Figura 1.1: Materiales principales para el proyecto.

Nota: El total del costo de todos los materiales para el proyecto fue de 1772 pesos.

Materiales de Software:

- Arduino IDE

Funcionamiento

Para el funcionamiento del proyecto entero, es necesario entender como funcionan sus partes principales. Una de las partes principales son los servomotores. Estos cuentan con un motor DC que va convertir la señal eléctrica en movimiento mecánico. Dicho motor esta conectado a un controlador que obtiene la posición actual del servomotor con el Encoder y envía señales al motor para ejecutar pulsos que pondrán al servomotor en movimiento. Este también cuenta con algunos engranajes que hacen que el movimiento sea mas controlado.

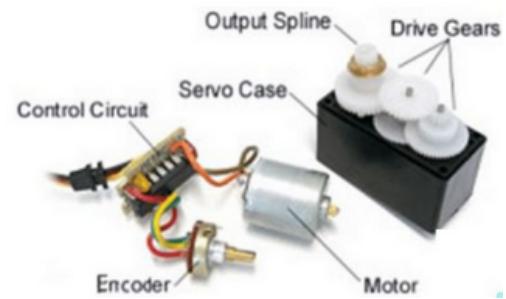


Figura 2.1: Componentes del servomotor.

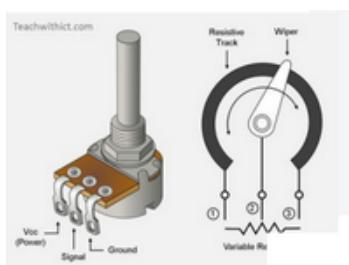


Figura 2.2: Componentes del potenciómetro.

Para enviar las señales de las posiciones actuales en las que el usuario desea que se mueva el brazo, se utilizaron potenciómetros, los cuales pueden interpretarse como resistencias variables cuya terminal central es la que contiene los datos de la posición actual del potenciómetro.

El sensor de color para el clasificador de colores utiliza las entradas SDA y SCL para enviar los datos del color actual que esta frente a el en formato de RGB. Este también cuenta con un LED que permite visualizar correctamente el color en la oscuridad.

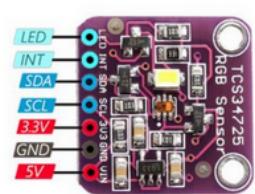


Figura 2.3:
Componentes del sensor de color.

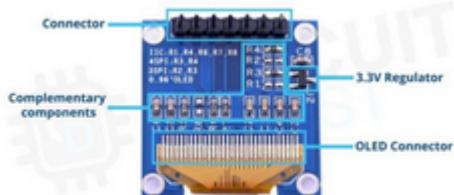


Figura 2.4: Componentes del OLED.

Por ultimo, el Driver de los servomotores utiliza las mismas entradas mencionadas anteriormente, solo que este tiene mas entradas para la fuente de poder externa que alimentara los servomotores y posee 16 entradas para el control de 16 servomotores diferentes.

La pantalla OLED cuenta con entradas para la corriente y, al igual que el sensor de color, este también cuenta con las entradas SDA y SCL para recibir información del Arduino.

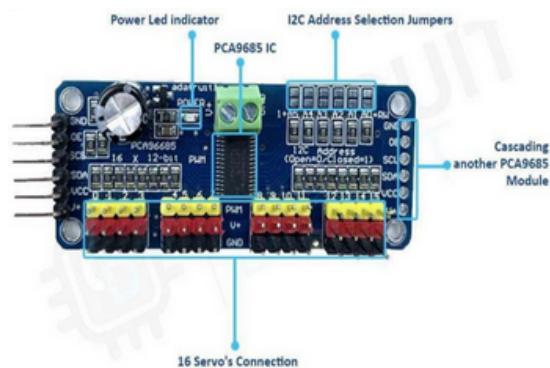


Figura 2.5: Componentes del Driver de servomotores.

Desarrollo del proyecto

Para el desarrollo del brazo robotico, se tomo en cuenta el diagrama de la simulacion de Tinkercad propuesta en la primera presentación, con algunas modificaciones. Las modificaciones mas importantes a esta simulacion son que los servomotores están conectados a un Driver que a su vez esta conectado al Arduino, el control de servomotores se hace mediante potenciómetros y ya no se utiliza una pantalla LCD para mostrar la posición actual.

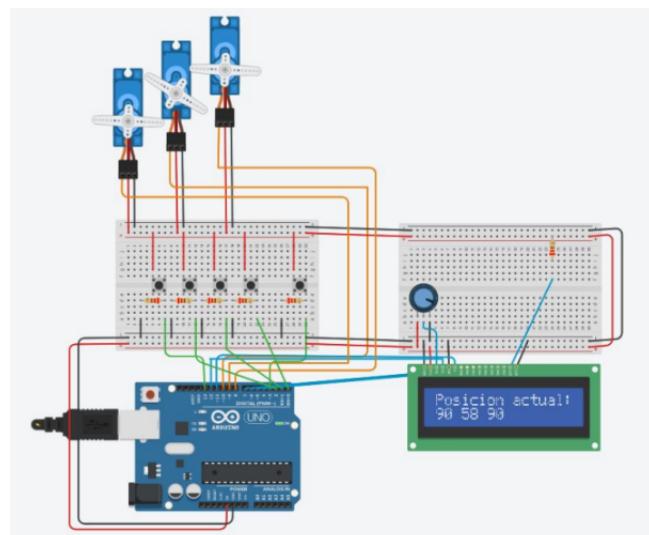


Figura 3.1: Simulación de Tinkercad.

Lo primero que haremos sera realizar las conexiones necesarias para el funcionamiento de los potenciómetros. Para ello, cada nodo lateral de los mismos estará conectado a negativo y positivo respectivamente (el orden no importa, debido a que esto solo influye en la polaridad del potenciómetro y la dirección en la que se tendrán que dar vuelta). El nodo central de los potenciómetros es el que cuenta con la señal que necesitamos, por lo que dichos nodos los conectaremos a los puertos analógicos del Arduino nano. También utilizaremos la fuente de poder proporcionada por esta placa para alimentar a los potenciómetros.

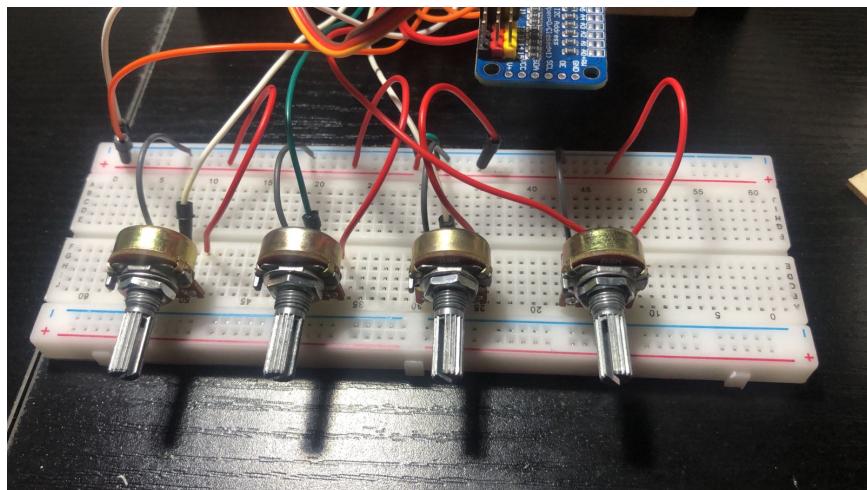


Figura 3.2: Arreglo de potenciómetros para el control del brazo.

Posteriormente, realizamos las conexiones necesarias para el funcionamiento del brazo. Para ello, conectamos las salidas de todos los potenciómetros a las entradas análogas del arduino nano y conectamos las entradas del Driver de servomotores a las ultimas entradas análogas del arduino, esto debido a que este arduino no cuenta con entradas SCL ni SDA. Posteriormente conectamos los servomotores a utilizar al Driver correspondiente, junto con una fuente de poder externa.

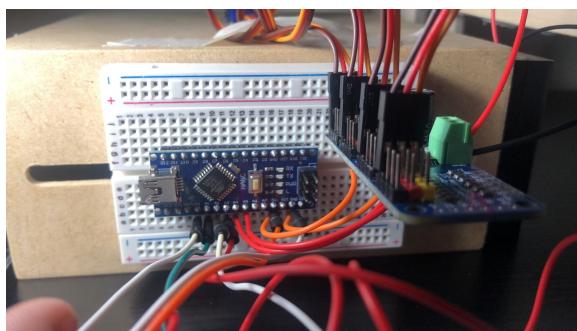


Figura 3.3: Conexiones entre el arreglo de potenciómetros, Arduino Naano y el Driver de servomotores.

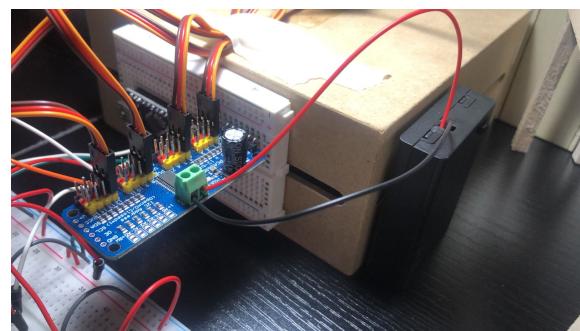


Figura 3.4: Conexiones para el Driver.

A continuación, se muestra el código necesario para la configuración de los pines hasta ahora:

```
// Include Adafruit PWM Library
#include <Adafruit_PWMServoDriver.h>

#define MIN_PULSE_WIDTH      650
#define MAX_PULSE_WIDTH      2350
#define FREQUENCY             50

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

// Define Potentiometer Inputs

int controlA = A0;
int controlB = A1;
int controlC = A2;
int controlD = A3;

// Define Motor Outputs on PCA9685 board

int motorA = 0;
int motorB = 4;
int motorC = 8;
```

```

int motorD = 12;

void setup()
{
    pwm.begin();
    pwm.setPWMFreq(FREQUENCY);
}

```

Una vez realizada esta parte, lo único que necesitamos realizar es pegar las piezas que hemos cortado en madera balsa a los servomotores. Para la base del brazo, se cortó una pieza cuadrada para sostener el segundo servomotor y lograr que pueda moverse tanto horizontal como verticalmente. Para la siguiente parte, se cortó una pieza que sostendrá dos servomotores en la misma dirección, para que el brazo pueda extenderse o contraerse. Por último se cortaron 3 piezas para la mandíbula. La primera sostiene el servomotor de la mandíbula y se conecta con el resto del brazo, la segunda sostiene una pinza para que el servomotor pueda sujetar cosas y la tercera es la otra parte de la pinza, la cual está pegada con silicona a la parte mecánica del servomotor. A continuación se muestran algunas figuras tanto del boceto inicial del brazo como del proceso de construcción del mismo.

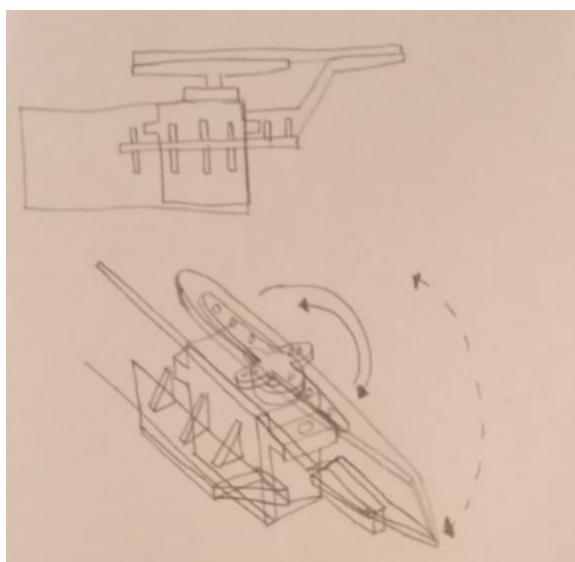


Figura 3.5: Bocetos para la mandíbula del brazo.

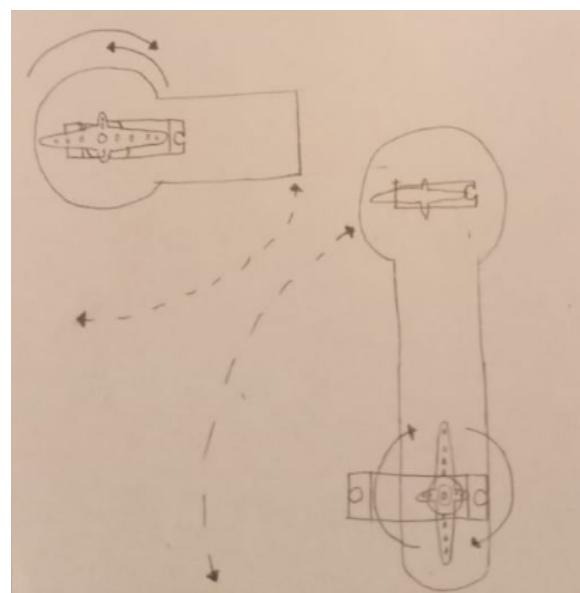


Figura 3.6: Bocetos para la estructura del brazo.



Figura 3.7: Base del brazo.

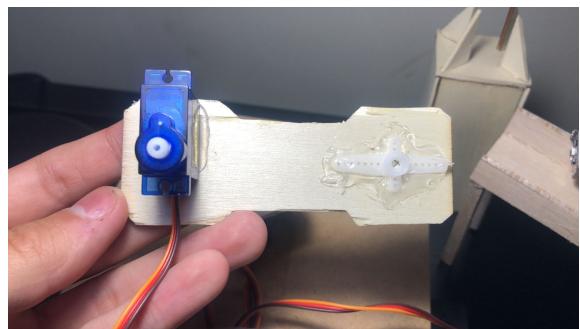


Figura 3.8: Primera parte del brazo.

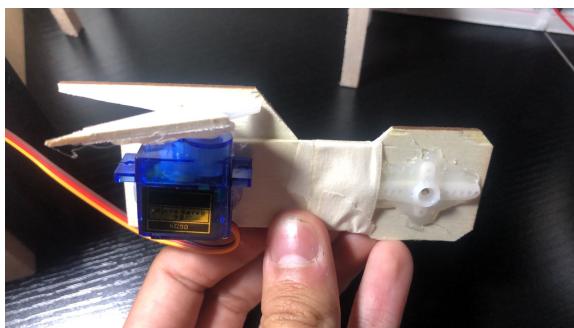


Figura 3.9: Mandíbula interior del brazo.

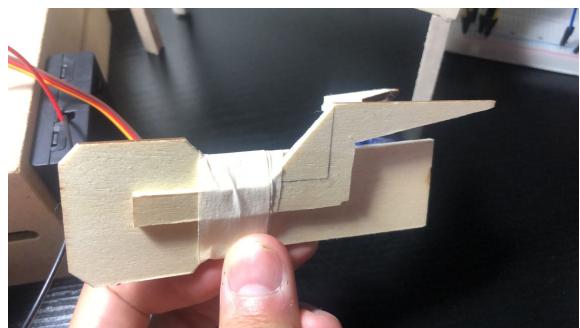


Figura 3.10: Mandíbula exterior del brazo.

A continuación, se muestra el código utilizado para mover los servomotores, junto con las respectivas estructuras de los brazos.

```
void moveMotor(int controlIn, int motorOut)
{
    int pulse_wide, pulse_width, potVal;

    // Leer los valores del potenciómetro
    potVal = analogRead(controlIn);

    // Convertir el ancho de los pulsos
    pulse_wide = map(potVal, 0, 1023, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
    pulse_width = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);

    //Controlar el motor
    pwm.setPWM(motorOut, 0, pulse_width);
```

```
}
```

Por ultimo, agregamos un código que verifique constantemente si un motor debe moverse o no en función de los potenciómetros. El armazón final del brazo se muestra en la siguiente figura.

```
void loop() {  
  
    //Control Motor A  
    moveMotor(controlA, motorA);  
  
    //Control Motor B  
    moveMotor(controlB, motorB);  
  
    //Control Motor C  
    moveMotor(controlC, motorC);  
  
    //Control Motor D  
    moveMotor(controlD, motorD);  
  
}
```

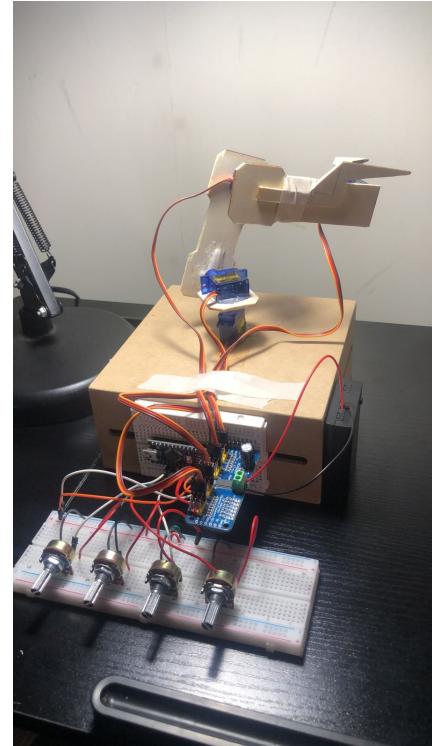


Figura 3.11: Construcción total de todo el brazo robotico.

Posteriormente, para las estructuras que necesitaremos para el funcionamiento del clasificador de colores, simplemente cortamos algunas piezas de madera de balsa para la rampa (3 rectángulos y 4 pilares con un tarugo de madera) y para el pilar que sostendrá las pelotas que tomara el brazo (4 rectángulos para el pilar, un cuadrado para la base y 4 triángulos para la estructura que sostiene las pelotas). Después cortamos un circulo y varios triángulos de cartón y los acomodamos con silicona y cinta como se muestra en la ultima figura, esto nos va a permitir realizar la estructura del clasificador que tendrá el servomotor debajo.



Figura 3.12: Rampa para el clasificador.



Figura 3.13: Estante para las esferas de colores.



Figura 3.14: Cilindro para la clasificación.

Una vez realizadas estas piezas, pegamos un servomotor a una placa de madera de balsa y colocamos el cilindro de cartón sobre el servomotor.

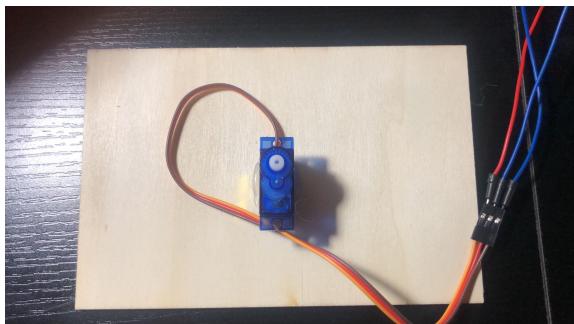


Figura 3.15: Servomotor para mover el clasificador.

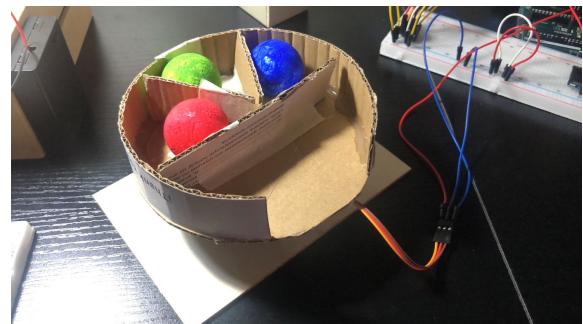


Figura 3.16: Clasificador montado en el servomotor.

Posteriormente, para la parte electrónica del clasificador, simplemente conectamos el sensor de color a la corriente proporcionada por el Arduino Uno y conectamos sus respectivos pines de datos a las ultimas entradas análogas que nos proporciona el mismo. Esto debido a que las entradas SCL y SDA serán

utilizadas por la pantalla OLED. Posteriormente conectamos la pantalla OLED a la corriente y a los pines SCL y SDA del Arduino y conectamos el servomotor a la corriente. El pin de los datos del servomotor sera conectado al pin digital 10 del Arduino.

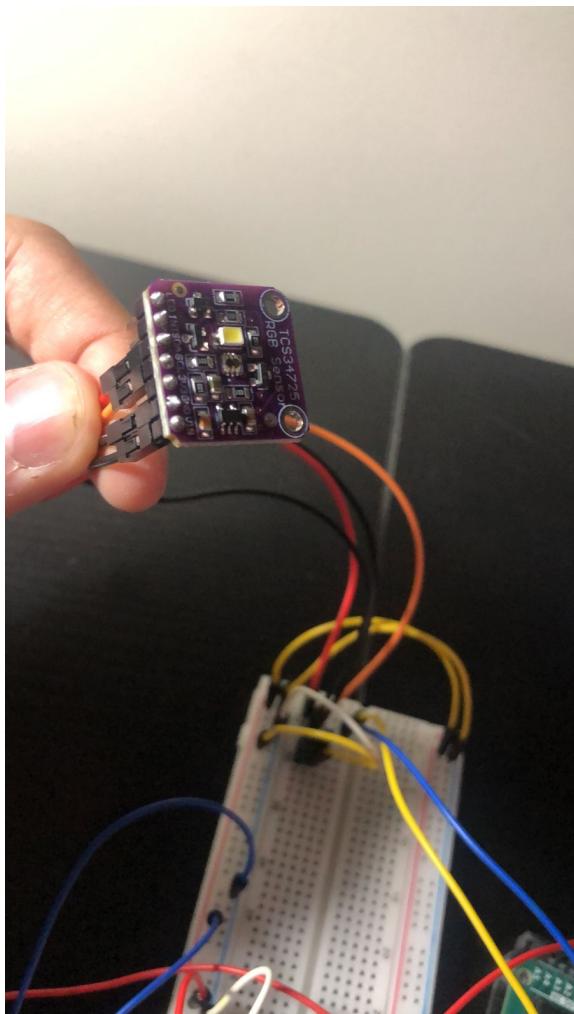


Figura 3.17: Sensor de color conectado al circuito principal del clasificador de colores.

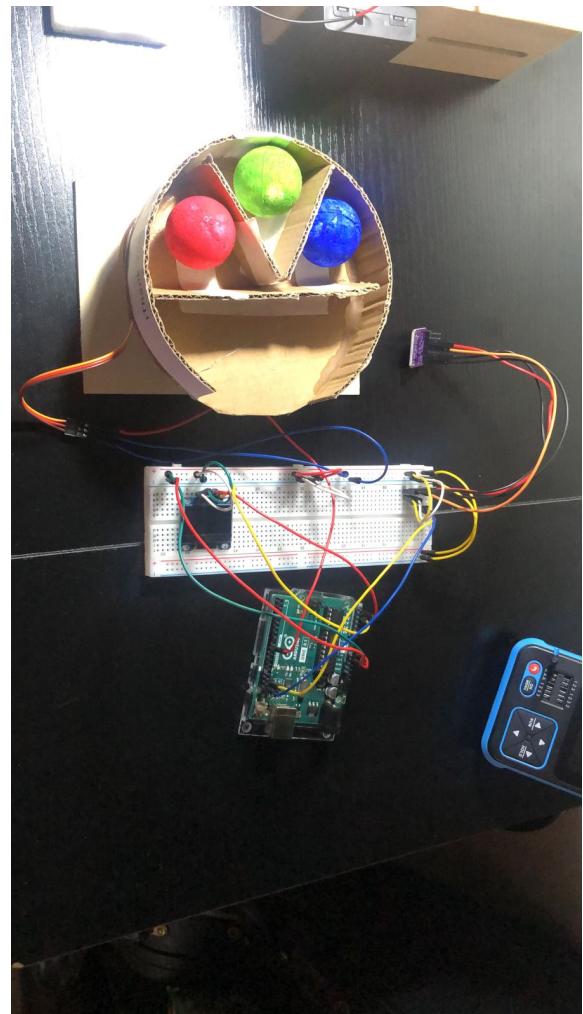


Figura 3.18: Arreglo final para el clasificador de colores.

En cuanto al código del clasificador de colores, primero incluimos las librerías necesarias para el funcionamiento del sensor y de la pantalla OLED. Posteriormente definimos algunas variables para configurar la pantalla OLED y lograr hacer el test inicial para determinar si funciona correctamente. Por ultimo, creamos una variable para el funcionamiento del sensor y otra para el funcionamiento y control del servomotor en función del sensor.

```

//Librerias principales
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//Variables para el funcionamiento del OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET     -1
#define SCREEN_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, _);

//Variables para probar la pantalla
#define NUMFLAKES      10
#define LOGO_HEIGHT    16
#define LOGO_WIDTH     16

//Librerias para el sensor y el servomotor
#include <Adafruit_TCS34725.h>
#include <Servo.h>

//Variables para el sensor y el servomotor
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_60MS,
                                             TCS34725_GAIN_1X);
Servo miservo;

```

Una vez definidas las librerías y variables, realizamos la configuración inicial para la pantalla, el sensor y el servomotor en la función `setup()`.

```

void setup() {
  Serial.begin(115200);
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;); // Para ciclar la comprobacion
  }
  tcs.begin(); // Iniciar el sensor

```

```
    miservo.attach(9); // Iniciar el servomotor  
}
```

Una vez realizada la configuración inicial, continuamos con el código que se repetirá en la función `loop()`. En el, definimos algunas variables para los valores RGB del sensor, así como una variable tipo cadena para desplegar el color actual en la pantalla OLED. Posteriormente, realizamos condicionales para determinar si el sensor se encuentra con un objeto rojo, verde o azul respectivamente. En dichas condicionales, se compara si las diferencias entre los valores RGB pasan umbrales que determinan dichos colores. Posteriormente se despliegan los datos que se obtienen en el momento mediante la pantalla OLED y la consola serial.

```
void loop() {  
    //Variables para detectar mediante RGB  
    float red, green, blue;  
    tcs.getRGB(&red, &green, &blue);  
  
    int R = int(red);  
    int G = int(green);  
    int B = int(blue);  
  
    //Variable para desplegar el color en pantalla  
    String color = "";  
  
    //Condicional para determinar si un objeto es rojo  
    if((R-G > 35)&(R-B > 35)){  
        color = "RED";  
        miservo.write(30);  
    }  
  
    //Condicional para determinar si un objeto es verde  
    if((G-R > 10)&(G-R > 30)){  
        color = "GREEN";  
        miservo.write(90);  
    }  
}
```

```

    //Condicional para determinar si un objeto es azul
if((B-G > 20)&(B-R > 40)){
    color = "BLUE";
    miservo.write(150);
}

//Se limpia y se configura la pantalla OLED
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);

//Se muestran los datos actuales detectados por el sensor
display.print("R: ");
display.print(int(red));
display.print(" G: ");
display.print(int(green));
display.print(" B: ");
display.println(int(blue));

//Se muestra el color en funcion de las condicionales
display.print("Color: ");
display.print(color);
display.setTextSize(2);
display.setCursor(40,10);
display.display();

//Se muestra lo mismo que desplegamos en pantalla en la consola
Serial.print("R: ");
Serial.print(int(red));
Serial.print(" G: ");
Serial.print(int(green));
Serial.print(" B: ");
Serial.print(int(blue));
Serial.print(" Color: ");
Serial.print(color);

```

```
    Serial.println();  
}  

```

Por ultimo, el proyecto debería quedar de la siguiente forma, en donde el estante de esferas de colores se encuentra cerca del brazo para que lo pueda alcanzar, al igual que la rampa y el sensor de colores. Al final de la rampa debe colocarse el clasificador y a la derecha la pantalla OLED para que pueda mostrarse el color que se esta clasificando en el momento.

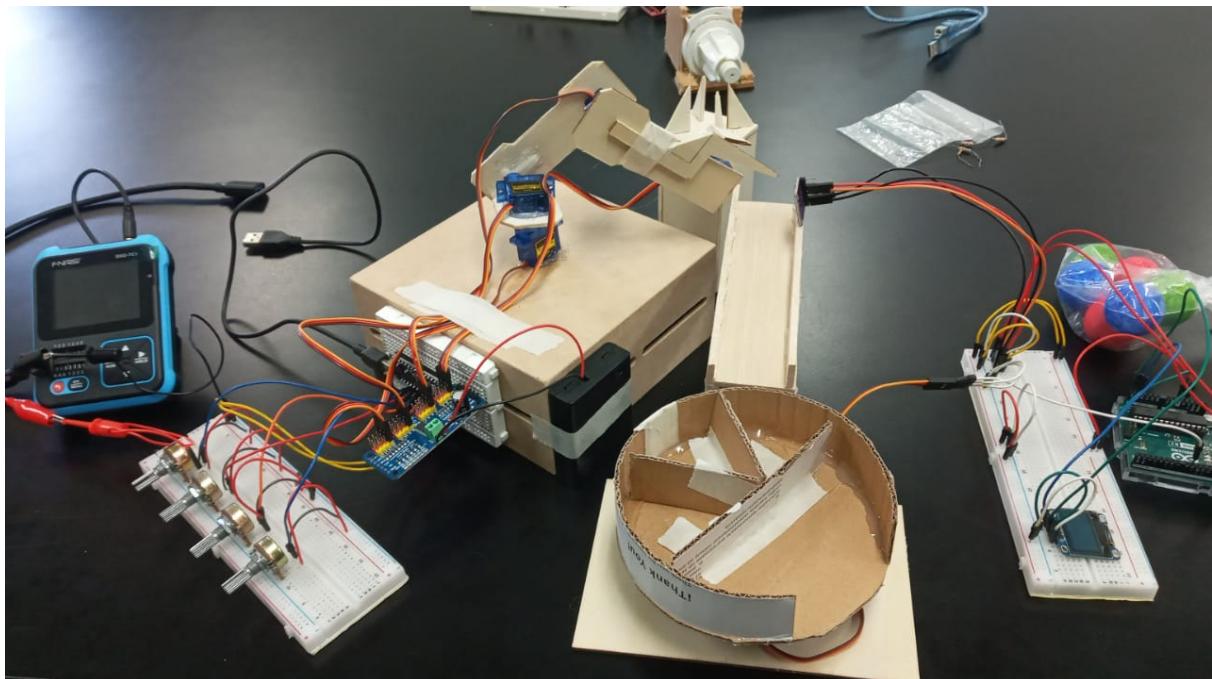


Figura 3.19: Brazo robotico acomodado junto al clasificador de colores.

A continuación, se muestra un video de su funcionamiento.

https://prod-files-secure.s3.us-west-2.amazonaws.com/53f49191-d144-4242-b55e-b751c7f03c0d/6290bea4-53c6-4d29-9361-e8c3be7f5dc1/WhatsApp_Video_2024-06-01_at_11.39.58_AM.mp4

Figura 3.20: Video de la ejecución del proyecto completo.

Conclusiones

Este proyecto me fue de mucha ayuda para adentrarme en cuestiones de sistemas empotrados más relacionadas a actuadores y me ayudó a complementar mi formación de computólogo con conocimientos más prácticos.

En algunos momentos fue muy estresante debido a las limitaciones que puede proporcionar la tarjeta de Arduino UNO, por lo que es necesario comprar más dispositivos para completar el proyecto.

Algunas cosas que se podrían mejorar son la disminución de la inercia de la rueda del clasificador, una estructura más sólida para el brazo, al igual que para el deslizador del clasificador. También podría agregarse un sistema para control de pelotas, en el que se lleve la cuenta de cada pelota para tomar decisiones de producción en función de ello.

Referencias

A continuación, se muestran algunos videos de proyectos que se tomaron como referencia para el desarrollo del mismo.

- https://www.youtube.com/watch?v=JFFHzGBWSE4&ab_channel=Hacktuber
- https://www.youtube.com/watch?v=9G8JasPzUk8&ab_channel=Maker101
- https://www.youtube.com/watch?v=ADJGxOrEZAM&t=12s&ab_channel=NamuLab

●

https://www.youtube.com/watch?v=cdYLOKGhvdU&ab_channel=⚡VictorInstala⚡
%9A%A1

●

<https://www.youtube.com/watch?v=JFFHzGBWSE4>

●

<https://www.youtube.com/watch?v=n18FmtGQdJ8>