



Universidad Internacional
SECUNDARIA . BACHILLERATO . LICENCIATURA . INGENIERÍA . POSGRADO

23-5-2023

Clasificador de colores

Luis Alberto López Pedroza

Aníbal Jesús Aguilar Cardoso

Microcontroladores

Ing. Martin Omar Vázquez Reyes

Universidad Internacional

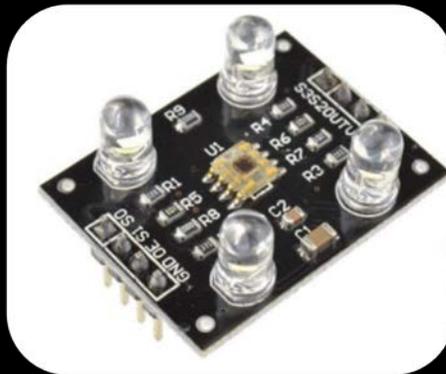
Contenido

| | |
|--------------------|----|
| Introducción | 2 |
| Material..... | 3 |
| Desarrollo..... | 4 |
| Código | 4 |
| LabVIEW | 19 |
| Impresión 3D | 20 |
| Ensamble | 24 |
| Problemas..... | 31 |
| Conclusión | 33 |
| Referencias..... | 33 |

Introducción

Para la realización del proyecto final se escogió hacer un clasificador de colores que con el sensor de colores detectara el color de la caja y así lo clasificara al lugar donde debe ir y además que tuviera una parte que se pueda controlar manual y esto se lograría haciendo la serie de comunicación entre el programa de Arduino y el programa de LABVIEW.

Para el proyecto final el elemento clave fue el sensor de color ya que si no se detecta el color no podría clasificar las cajas a su lugar correspondiente. Un sensor de color detecta el color del material. Este sensor suele detectar el color en la escala RGB. Este sensor puede clasificar el color como rojo, azul o verde. Estos sensores también están equipados con filtros para rechazar la luz IR y UV no deseada.



Para detectar el color de un material se necesitan tres tipos principales de equipos. Una fuente de luz para iluminar la superficie del material, una superficie cuyo color debe detectarse, y receptores capaces de medir las longitudes de onda reflejadas.

Los sensores de color contienen un emisor de luz blanca para iluminar la superficie. Tres filtros con sensibilidad de longitud de onda de 580 nm, 540 nm y 450 nm para medir las longitudes de onda de los colores rojo, verde y azul, respectivamente.

Material

Para este proyecto final los materiales que se utilizaron fueron los siguientes;

- 6 sensores ultrasónicos
- 2 Arduino Uno
- Arduino Mega
- 2 Protoboard
- Sensor de colores
- 2 servomotores
- Motorreductor
- Puente H
- Jumpers

Desarrollo

Para la realización del proyecto final se utilizaron diferentes cosas para la construcción de la banda y hacer que se lograra clasificar las cosas y lo primero que se hizo fue que se realizaron las pruebas de los códigos por separado para probar como funcionaria cada material de los sistemas del clasificador de colores para al final juntar todos los códigos y tener el programa completo que controlara el clasificador de colores.

Código

El primer código que se probó fue el código para el sensor de colores para poder detectar el color y se utilizó el siguiente código para indicarnos el color de las cajas.

```
sensor_de_color
delay(200);

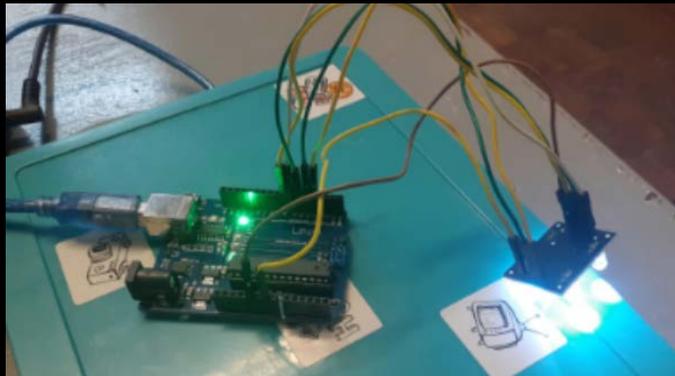
// Read Blue Pulse Width
bluePW = getBluePW();
// Delay to stabilize sensor
delay(200);

// Print output to Serial Monitor
Serial.print("Red PW = ");
Serial.print(redPW);
Serial.print(" - Green PW = ");
Serial.print(greenPW);
Serial.print(" - Blue PW = ");
Serial.println(bluePW);
}

// Function to read Red Pulse Widths
int getRedPW() {
// Set sensor to read Red only
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);
// Define integer to represent Pulse Width
int PW;
// Read the output Pulse Width
PW = pulseIn(sensorOut, LOW);
// Return the value
return PW;
}
```

```
// Function to read Blue Pulse Widths
int getBluePW() {
  // Set sensor to read Blue only
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  // Define integer to represent Pulse Width
  int PW;
  // Read the output Pulse Width
  PW = pulseIn(sensorOut, LOW);
  // Return the value
  return PW;
}
```

Con este código pudimos saber donde conectar cada entrada y salida digital en el Arduino para poder comprobar si leía de manera correcta los colores de los objetos poniéndole diferentes objetos para ver que colores indicaba y nos mostraba solo tres valores de colores el verde, azul y rojo y dependiendo el color bajaba o subía la intensidad de los tres colores mencionados.



El segundo código que se realizó fue el código para poder mover el motor a pasos y para hacerlo mover se probaron varios códigos ya que hubo dificultades para hacerlo funcionar y con este código se logró mover.

```

motor_a_pasos_impresora_3d
// Definición de pines
const int pinEnableA = 2; // Pin de habilitación del motor A
const int pinInput1A = 3; // Pin de entrada 1 del motor A (IN3)
const int pinInput2A = 4; // Pin de entrada 2 del motor A (IN4)

void setup() {
  // Configurar los pines como salidas
  pinMode(pinEnableA, OUTPUT);
  pinMode(pinInput1A, OUTPUT);
  pinMode(pinInput2A, OUTPUT);
}

void loop() {
  // Girar hacia adelante durante 2 segundos
  moveForward();
  delay(2000);

  // Girar hacia atrás durante 2 segundos
  moveBackward();
  delay(2000);
}

void moveForward() {
  digitalWrite(pinInput1A, HIGH);
  digitalWrite(pinInput2A, LOW);
  analogWrite(pinEnableA, 135); // Control de velocidad del motor A (0-255)
}

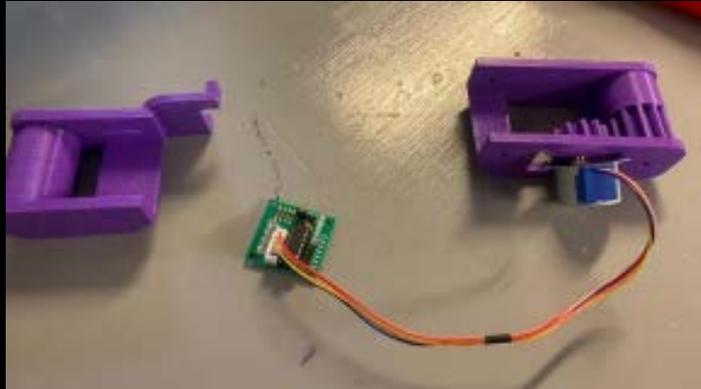
```

```

void moveBackward() {
  digitalWrite(pinInput1A, LOW);
  digitalWrite(pinInput2A, HIGH);
  analogWrite(pinEnableA, 135); // Control de velocidad del motor A (0-255)
}

```

Con este código pudimos saber dónde conectar cada entrada y salida digital en el Arduino para poder hacer que se moviera el motor a pasos pero después de que se logro que funcionara el motor a pasos y de que se instaló en la banda transportadora se hicieron pruebas y el motor no funcionaba se trababa por eso se decidió cambiar por un motorreductor y el código del motorreductor se agrego directamente al programa final ya que se siguió intentado hacer que se moviera muchas veces por al final no funciono y por eso se cambió antes de entregarlo.



El siguiente código que se probó fue el código para hacer mover el servomotor con este código no hubo ningún problema, solo hubo problemas al momento de conectarlo al programa de LabVIEW por eso se decidió utilizar dos Arduino uno para el programa de Arduino para hacer mover el servomotor y el otro Arduino fue para el programa de LabVIEW.

```
ayuda_ciclista
#include <Servo.h>

int pos = 0;

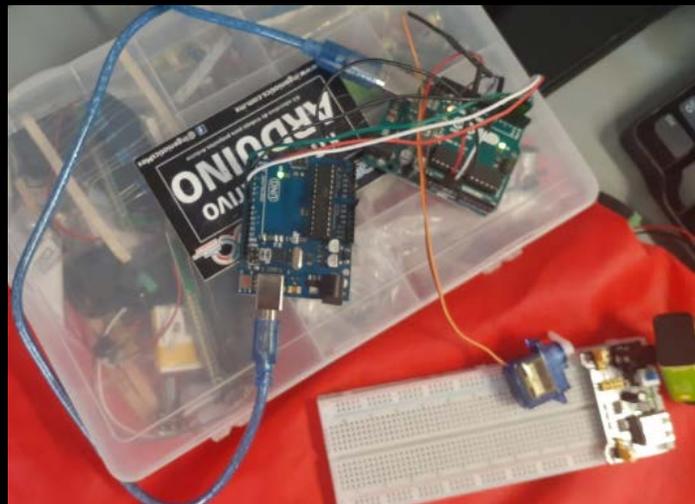
Servo servo_9;
int pwm = 2;//0
int modo = 5;//selector de modo
int vpwm = 0;//0
int vmodo = 0;
void setup()
{
  servo_9.attach(9, 500, 2500);
  pinMode(pwm, INPUT);
  pinMode(modo, INPUT);
}

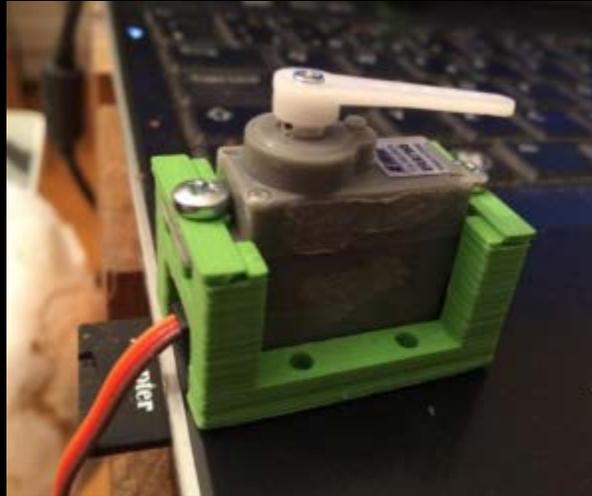
void loop()
{
  va = analogRead(a);
  vmodo = digitalRead(modo);

  if (vmodo == 1){
    Serial.println();
    servo_9.write();
  }
}
```

```
}
else{
  for (pos = 0; pos <= 180; pos += 1) {
    // tell servo to go to position in variable 'pos'
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    // tell servo to go to position in variable 'pos'
    servo_9.write(pos);
    // wait 15 ms for servo to reach the position
    delay(15); // Wait for 15 millisecond(s)
  }
}
}
```

En las siguientes imágenes nos muestra como conectamos el servomotor y como utilizamos dos Arduino unos para que uno controlara el programa de Arduino y el otro para controlar el programa de LabVIEW.





El ultimo código que se probó fue el código para poder detectar las cajas con los sensores ultrasónicos y este código sirve como un actuador o activador al momento en que detecte una caja este activará el sensor de color o los servomotores dependiendo del sensor ultrasónico para que haga la función de empujar la caja o de leer el color.

```
sensores_2_xd
// Librería para la comunicación con los sensores ultrasónicos
#include <NewPing.h>

// Pines del sensor 1
const int trigPin1 = 2;
const int echoPin1 = 3;

// Pines del sensor 2
const int trigPin2 = 4;
const int echoPin2 = 5;

// Pines del sensor 3
const int trigPin3 = 6;
const int echoPin3 = 7;

// Distancia mínima para contar una caja
const int distanciaMinima = 5;

// Contadores de cajas
int contadorAzules = 0;
int contadorAmarillas = 0;
int contadorVerdes = 0;

// Inicialización de los sensores ultrasónicos
NewPing sensor1(trigPin1, echoPin1);
NewPing sensor2(trigPin2, echoPin2);
NewPing sensor3(trigPin3, echoPin3);
```

```

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Lectura de distancia del sensor 1
  unsigned int distancia1 = sensor1.ping_cm();
  if (distancia1 <= distanciaMinima) {
    contadorAzules++;
    Serial.println("Se detectó una caja azul");
  }

  // Lectura de distancia del sensor 2
  unsigned int distancia2 = sensor2.ping_cm();
  if (distancia2 <= distanciaMinima) {
    contadorAmarillas++;
    Serial.println("Se detectó una caja amarilla");
  }

  // Lectura de distancia del sensor 3
  unsigned int distancia3 = sensor3.ping_cm();
  if (distancia3 <= distanciaMinima) {
    contadorVerdes++;
    Serial.println("Se detectó una caja verde");
  }

  // Esperar un tiempo antes de la siguiente lectura
  delay(50);
}

```

Por ultimo se juntaron todos los códigos para crear el programa completo que se dividió en dos partes ya que hubo problemas con el funcionamiento de algunos sistemas del clasificador de colores ya que se saturó el Arduino mega y la primera parte del código controlara la parte de la banda transportadora, de los sensores y de los servomotores todo esto está conectado en el Arduino mega.

codigo_completo

```
// Motor reductor con modulo puente h
const int pinEnableA = 2; // Pin de habilitación del motor A
const int pinInput1A = 3; // Pin de entrada 1 del motor A (IN3)
const int pinInput2A = 4; // Pin de entrada 2 del motor A (IN4)

//lectura de sensor ultrasonico 1
const int trigPin1 = 6; // Pin de Trigger del primer sensor ultrasónico
const int echoPin1 = 7; // Pin de Echo del primer sensor ultrasónico

//lectura de sensor ultrasonico 2
const int trigPin2 = 8; // Pin de Trigger del primer sensor ultrasónico
const int echoPin2 = 9; // Pin de Echo del primer sensor ultrasónico

//lectura de sensor ultrasonico 3
const int trigPin3 = 10; // Pin de Trigger del primer sensor ultrasónico
const int echoPin3 = 11; // Pin de Echo del primer sensor ultrasónico

//servomotor 1
#include <Servo.h>
Servo servoMotor1; // Crear un objeto de la clase Servo
//servomotor 2
Servo servoMotor2; // Crear un objeto de la clase Servo
```

```
// Definimos los pines del sensor de color
const int S0 = 14;
const int S1 = 15;
const int S2 = 16;
const int S3 = 17;
const int OUT = 18;

int coloractual = 0;

//labview
const int botonModoPin = 25;
const int botonServo10Pin = 26;
const int botonServo180Pin = 27;
const int botonServo20Pin = 28;
const int botonServo2180Pin = 29;
const int botonMotorAtrasPin = 30;
const int botonMotorAdelantePin = 31;

int modo = 0; // 1: manual, 0: automatico
int servo_1_0 = 0;
int servo_1_180 = 0;
int servo_2_0 = 0;
int servo_2_180 = 0;
int motoratras = 0;
int motoradelante = 0;
```

```

void setup() {
  // Configuramos los pines del motor como salida
  pinMode(pinEnableA, OUTPUT);
  pinMode(pinInput1A, OUTPUT);
  pinMode(pinInput2A, OUTPUT);

  // Inicializamos el puerto serial para permitir la comunicación con el usuario
  Serial.begin(9600);

  //sensor ultrasonico 1
  pinMode(trigPin1, OUTPUT); //pin como salida
  pinMode(echoPin1, INPUT); //pin como entrada
  digitalWrite(trigPin1, LOW); //Inicializamos el pin con 0

  //sensor ultrasonico 2
  pinMode(trigPin2, OUTPUT); //pin como salida
  pinMode(echoPin2, INPUT); //pin como entrada
  digitalWrite(trigPin2, LOW); //Inicializamos el pin con 0

  //sensor ultrasonico 3
  pinMode(trigPin3, OUTPUT); //pin como salida
  pinMode(echoPin3, INPUT); //pin como entrada
  digitalWrite(trigPin3, LOW); //Inicializamos el pin con 0

  //servomotor 1
  servoMotor1.attach(12); // Adjuntar el servo al pin 12

  //servomotor 2
  servoMotor2.attach(13); // Adjuntar el servo al pin 12

  // Inicializamos los pines del sensor de color
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);

  // Configuramos la frecuencia de trabajo del sensor a 20%
  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);

  //botones digitales de labview
  pinMode(botonModoPin, INPUT_PULLUP);
  pinMode(botonServo10Pin, INPUT_PULLUP);
  pinMode(botonServo1180Pin, INPUT_PULLUP);
  pinMode(botonServo20Pin, INPUT_PULLUP);
  pinMode(botonServo2180Pin, INPUT_PULLUP);
  pinMode(botonMotorAtrasPin, INPUT_PULLUP);
  pinMode(botonMotorAdelantePin, INPUT_PULLUP);
}

```

```

void loop() {
  //iniciamos los servos en 0 grados
  servoMotor1.write(0);    // Enviar el ángulo al servo
  servoMotor2.write(0);    // Enviar el ángulo al servo

  //leemos valores de el sensor de color
  // Leemos el valor del color rojo
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  int red = pulseIn(OUT, LOW);
  // Leemos el valor del color verde
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  int green = pulseIn(OUT, LOW);
  // Leemos el valor del color azul
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  int blue = pulseIn(OUT, LOW);
  delay(250);
}

```

```

//Sensor ultrasonico 1 distancia de la caja a 6 cm
long t1; //timepo que demora en llegar el eco
long d1; //distancia en centimetros
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);    //Enviamos un pulso de 10us
digitalWrite(trigPin1, LOW);
t1 = pulseIn(echoPin1, HIGH); //obtenemos el ancho del pulso
d1 = t1/59;                //escalamos el tiempo a una distancia en cm
/*
Serial.print("Distancial: ");
Serial.print(d1);        //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
*/

//Sensor ultrasonico 2 distancia de la caja a 6 cm
long t2; //timepo que demora en llegar el eco
long d2; //distancia en centimetros
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);    //Enviamos un pulso de 10us
digitalWrite(trigPin2, LOW);
t2 = pulseIn(echoPin2, HIGH); //obtenemos el ancho del pulso
d2 = t2/59;                //escalamos el tiempo a una distancia en cm
/*

```

```

/*
Serial.print("Distancia2: ");
Serial.print(d2);      //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
*/

//Sensor ultrasonico 3 distancia de la caja a 7 cm
long t3; //timepo que demora en llegar el eco
long d3; //distancia en centimetros
digitalWrite(trigPin3, HIGH);
delayMicroseconds(10);      //Enviamos un pulso de 10us
digitalWrite(trigPin3, LOW);
t3 = pulseIn(echoPin3, HIGH); //obtenemos el ancho del pulso
d3 = t3/59;      //escalamos el tiempo a una distancia en cm
/*
Serial.print("Distancia3: ");
Serial.print(d3);      //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
*/

```

```

//botones
modo = digitalRead(botonModoPin);
servo_1_0 = digitalRead(botonServo10Pin);
servo_1_180 = digitalRead(botonServo1180Pin);
servo_2_0 = digitalRead(botonServo20Pin);
servo_2_180 = digitalRead(botonServo2180Pin);
motoratras = digitalRead(botonMotorAtrasPin);
motoradelante = digitalRead(botonMotorAdelantePin);

if (red>=177 && red <=360 && green >=133 && green <=290 && blue >=99 && blue <=196){
  coloractual = 1;
  Serial.println("El cubo es color azul");
}
else if (red>=239 && red <=390 && green >=148 && green <=323 && blue >=171 && blue <=315){
  coloractual = 2;
  Serial.println("El cubo es color verde");
}
else if (red>=88 && red <=199 && green >=84 && green <=197 && blue >=137 && blue <=244){
  coloractual = 3;
  Serial.println("El cubo es color amarillo");
}/*
else {
  Serial.println(coloractual);
}*/

```

```

//movimiento de la banda
// Girar hacia adelante
//atras();

//automatico 0
if(modo == 0){
  //color azul
  if(coloractual = 1){
    servoMotor1.write(180);
    servoMotor1.write(0);
  }
  //color verde
  else if(coloractual = 2){
    servoMotor2.write(180);
    servoMotor2.write(0);
  }
}
else{
  if (servo_1_0 == HIGH) {
    // Botón servo 1 a posición 0 grados
    servoMotor1.write(0);
  }
}

```

```

if (servo_1_180 == HIGH) {
  // Botón servo 1 a posición 180 grados
  servoMotor1.write(180);
}

if (servo_2_0 == HIGH) {
  // Botón servo 2 a posición 0 grados
  servoMotor2.write(0);
}

if (servo_2_180 == HIGH) {
  // Botón servo 2 a posición 180 grados
  servoMotor2.write(180);
}

if (motoratras == HIGH) {
  // Botón motor hacia atrás
  atras();
}

if (motoradelante == HIGH) {
  // Botón motor hacia adelante
  adelante();
}
}
}

```

```

}

void adelante() {
  digitalWrite(pinInput1A, HIGH);
  digitalWrite(pinInput2A, LOW);
  analogWrite(pinEnableA, 45); // Control de velocidad del motor A (0-255)
}

void atras() {
  digitalWrite(pinInput1A, LOW);
  digitalWrite(pinInput2A, HIGH);
  analogWrite(pinEnableA, 45); // Control de velocidad del motor A (0-255)
}

```

La segunda parte del código sirve para el sistema de contadores de las cajas de cada sección de los colores donde nos indicará cuantas cajas hay en cada sección y todo esto estará conectado en un tercer Arduino que servirá para este código del contador.

```

sensor_ultrasonico
//contadores cajas
int contadorazul = 0;
int contadorverde = 0;
int contadoramarrillo = 0;

const int contazul = 10;
const int contverde = 11;
const int contamarillo = 12;

//lectura de sensor ultrasonico 1
const int trigPin1 = 2; // Pin de Trigger del primer sensor ultrasónico
const int echoPin1 = 3; // Pin de Echo del primer sensor ultrasónico

//lectura de sensor ultrasonico 2
const int trigPin2 = 4; // Pin de Trigger del primer sensor ultrasónico
const int echoPin2= 5; // Pin de Echo del primer sensor ultrasónico

//lectura de sensor ultrasonico 3
const int trigPin3 = 6; // Pin de Trigger del primer sensor ultrasónico
const int echoPin3 = 7; // Pin de Echo del primer sensor ultrasónico

void setup() {
  // Inicializamos el puerto serial para permitir la comunicación con el usuario
  Serial.begin(9600);
}

```

```

    //envio de datos contadores
    pinMode(contazul, OUTPUT);
    pinMode(contverde, OUTPUT);
    pinMode(contamarillo, OUTPUT);

    //sensor ultrasonico 1
    pinMode(trigPin1, OUTPUT); //pin como salida
    pinMode(echoPin1, INPUT); //pin como entrada
    digitalWrite(trigPin1, LOW); //Inicializamos el pin con 0

    //sensor ultrasonico 2
    pinMode(trigPin2, OUTPUT); //pin como salida
    pinMode(echoPin2, INPUT); //pin como entrada
    digitalWrite(trigPin2, LOW); //Inicializamos el pin con 0

    //sensor ultrasonico 3
    pinMode(trigPin3, OUTPUT); //pin como salida
    pinMode(echoPin3, INPUT); //pin como entrada
    digitalWrite(trigPin3, LOW); //Inicializamos el pin con 0
}

```

```

void loop() {
    //Sensor ultrasonico 1 distancia de la caja a 4 cm
    long t1; //timepo que demora en llegar el eco
    long d1; //distancia en centimetros
    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
    digitalWrite(trigPin1, LOW);
    t1 = pulseIn(echoPin1, HIGH); //obtenemos el ancho del pulso
    d1 = t1/59; //escalamos el tiempo a una distancia en cm
    /*
    Serial.print("Distancia1: ");
    Serial.print(d1); //Enviamos serialmente el valor de la distancia
    Serial.print("cm");
    Serial.println();
    */

    //Sensor ultrasonico 2 distancia de la caja a 4 cm
    long t2; //timepo que demora en llegar el eco
    long d2; //distancia en centimetros
    digitalWrite(trigPin2, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
    digitalWrite(trigPin2, LOW);
    t2 = pulseIn(echoPin2, HIGH); //obtenemos el ancho del pulso
    d2 = t2/59; //escalamos el tiempo a una distancia en cm
    /*
    Serial.print("Distancia2: ");
    Serial.print(d2); //Enviamos serialmente el valor de la distancia
    Serial.print("cm");
    Serial.println();
    */
}

```

```

Serial.print("cm");
Serial.println();
*/

//Sensor ultrasonico 3 distancia de la caja a 4 cm
long t3; //timepo que demora en llegar el eco
long d3; //distancia en centimetros
digitalWrite(trigPin3, HIGH);
delayMicroseconds(10); //Enviamos un pulso de 10us
digitalWrite(trigPin3, LOW);
t3 = pulseIn(echoPin3, HIGH); //obtenemos el ancho del pulso
d3 = t3/59; //escalamos el tiempo a una distancia en cm
/*
Serial.print("Distancia3: ");
Serial.print(d1); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
*/

if (d1 <= 4){
  contadorazul = contadorazul + 1;
  digitalWrite(contazul, HIGH);
  Serial.println( contadorazul, "numero de cajas azules");
}/*

```

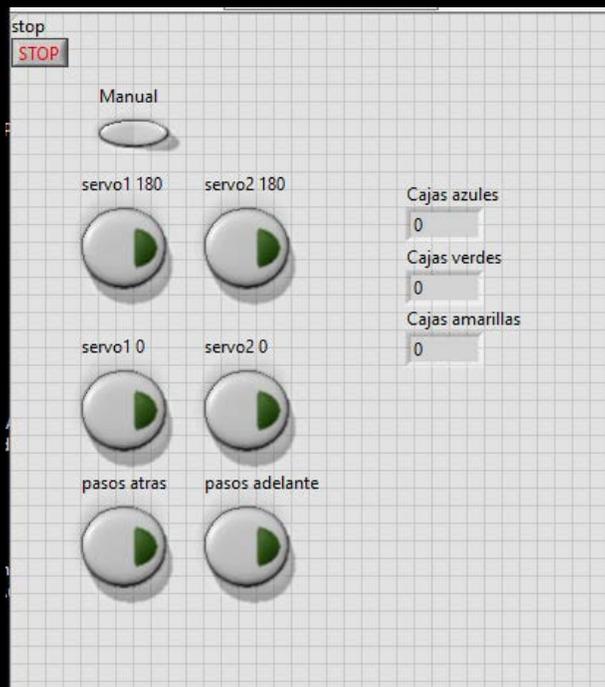
```

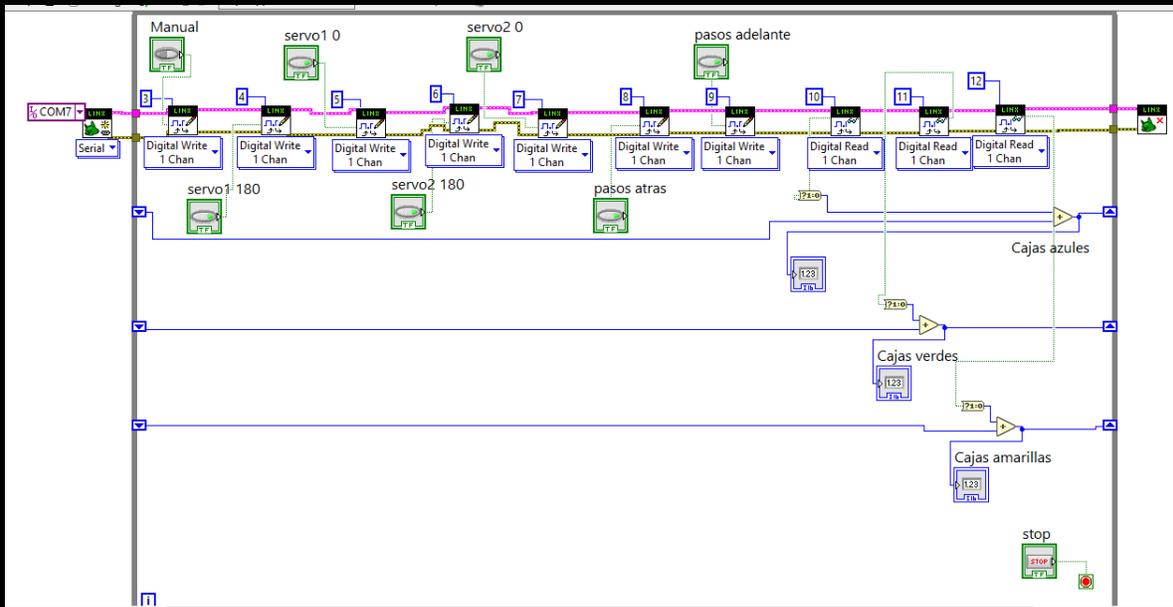
if (d2 <= 4){
  contadorverde = contadorverde + 1 ;
  digitalWrite(contverde, HIGH);
  Serial.println( contadorverde, "numero de cajas verdes");
}
if (d3 <= 4){
  contadoramarillo = contadoramarillo + 1;
  digitalWrite(contamarillo, HIGH);
  Serial.println( contadoramarillo, "cajas amarillas");
}
}

```

LabVIEW

En el programa de LabVIEW se hizo la parte manual del clasificador de colores en el programa se diseño para poder controlar algunas partes de la banda transportadora y esto funciona activando el botón Manual para cambiar el sistema de la banda transportador de automático a manual como solo activar el botón. Al activar el botón se puede control con los botones la banda transportadora si queremos que vaya hacia adelante o hacia atrás y también podemos controlar los dos servomotores para empujar las cajas a sus secciones. Además de controlar el clasificador de colores de manera manual también tenemos una parte donde aparece el conteo de las cajas que caen de cada color en sus secciones indicándonos cuantas cajas están en la sección azul o en la verde.





Impresión 3D

Para la elaboración de la estructura del clasificador de colores se decidió imprimir las piezas utilizando la impresión 3D que es el proceso de creación de objetos mediante el depósito de capas de material unas sobre otras. La impresión 3D se denomina fabricación aditiva (AM) en lugar de los métodos sustractivos tradicionales, como el fresado CNC, cuando se utiliza para la producción industrial.

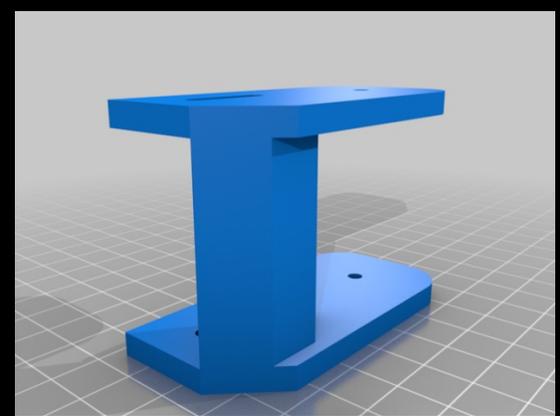
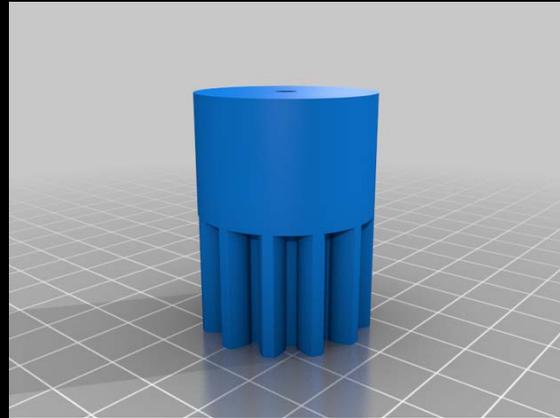
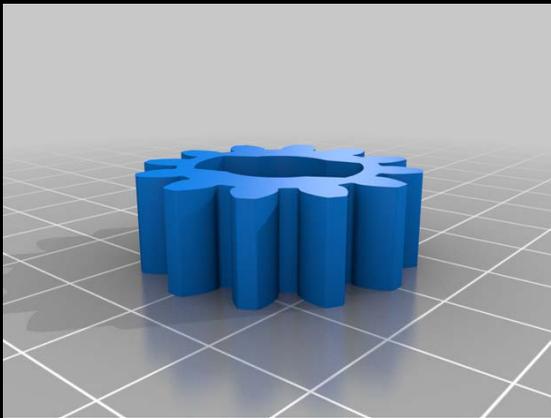


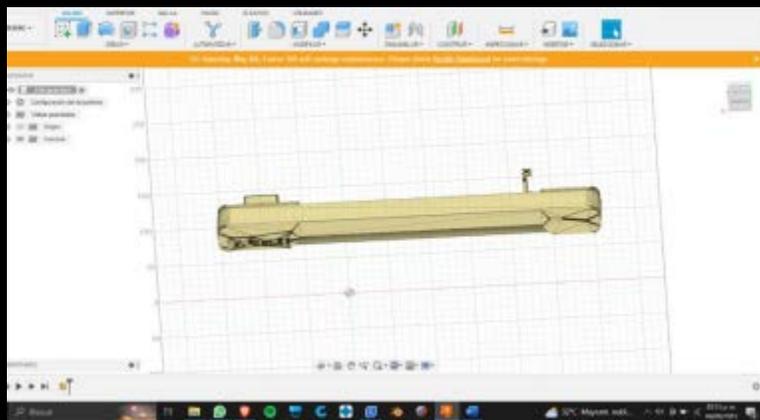
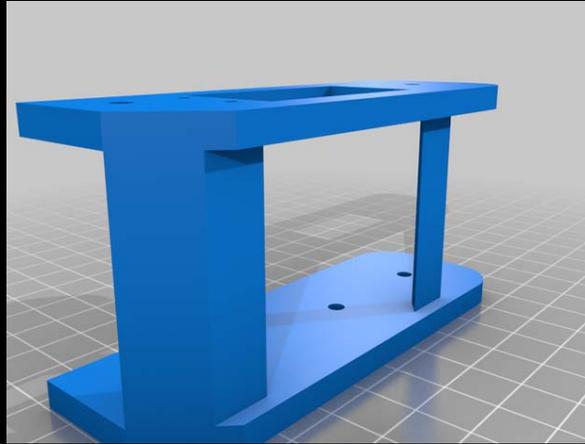
Un modelo digital en 3D se corta en cientos de capas finas mediante un software específico para exportarlo en formato de código G. Este formato de impresión 3D es un lenguaje que la impresora 3D lee para saber con precisión cuándo y dónde depositar el material.

Cada capa corresponde a la forma 2D exacta de una sección o rebanada del objeto. Por ejemplo, si se imprimiera en 3D una pirámide, la primera capa (la inferior) sería un cuadrado plano, y la última capa (en la parte superior) sería un pequeño punto.

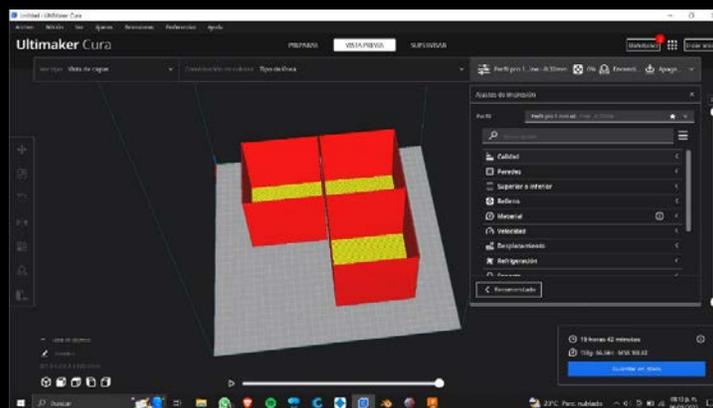
Las capas se imprimen consecutivamente en 3D de una en una hasta obtener el objeto completamente impreso.

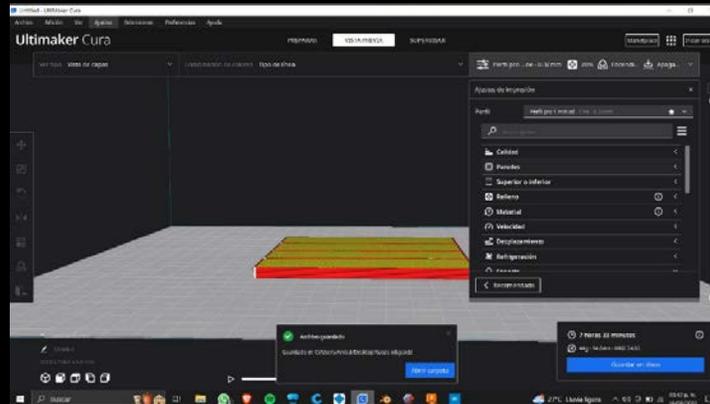
Lo primero que se tuvo que realizar fue el diseño de las piezas de la banda transportadora para saber el tamaño y la forma en la que va a estar la banda para el momento del ensamblaje y para el momento en el que se le colocara los demás componentes para el clasificador de colores.



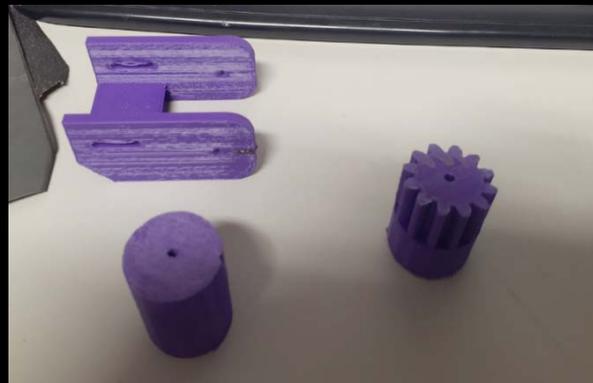


Además de diseñar las piezas de la banda también se diseñaron otras cosas que fueron necesarias para la estructura del clasificador de colores como las cajas, el soporte de la banda y también los soportes para los sensores.



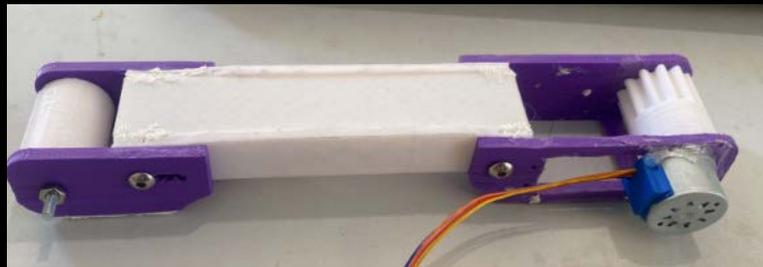
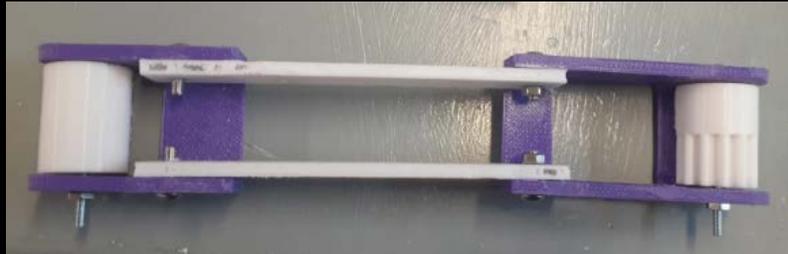


Luego se mandaron a imprimir los diseños en una impresora 3D y la impresora tardó de imprimir todas las piezas en un aproximado de dos días, aun contando que se tuvieron que volver a imprimir algunas piezas ya que el diseño no salió como debería o se tuvo que cambiar las medidas de las piezas para que pudiera funcionar.



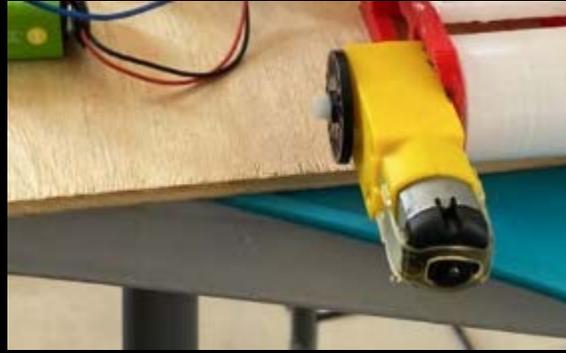
Ensamble

Lo primero que se ensamblo fue la banda transportadora para poder comprobar que funcionara con normalidad y que no hubiera ningún problema al momento de probar la banda con el motor a pasos.



Al momento de probar la banda con el motor a pasos se descubrió que tenía fallas ya que la banda se trababa y no giraba y para resolver eso se tuvo que volver a imprimir algunas piezas de la banda.

Luego de volver a probar la banda hubo problemas ya que cuando poníamos las cajas en la banda dejaba de moverse y para resolver ese problema se tuvo que cambiar el motor a pasos por un motorreductor con el puente H para que así la banda pudiera funcionar.

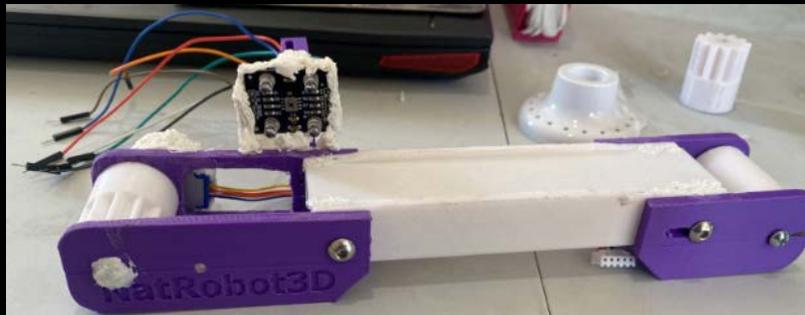


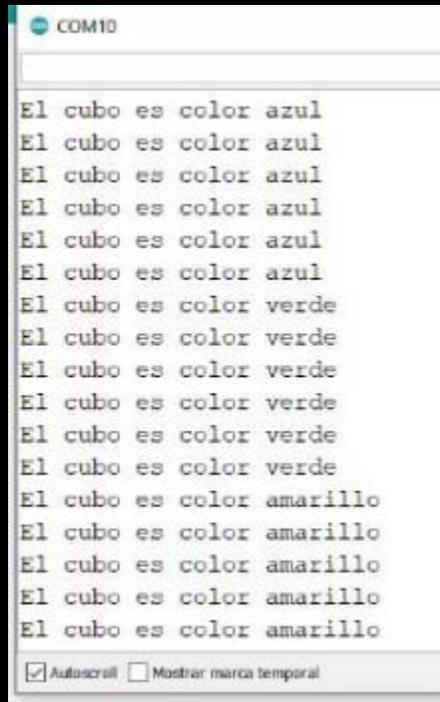
Después de que las cajas se imprimieran se tuvieron que forrar con hojas de colores Azul, Verde y Amarillo para que el sensor de colores pueda detectarlos y clasificarlos.



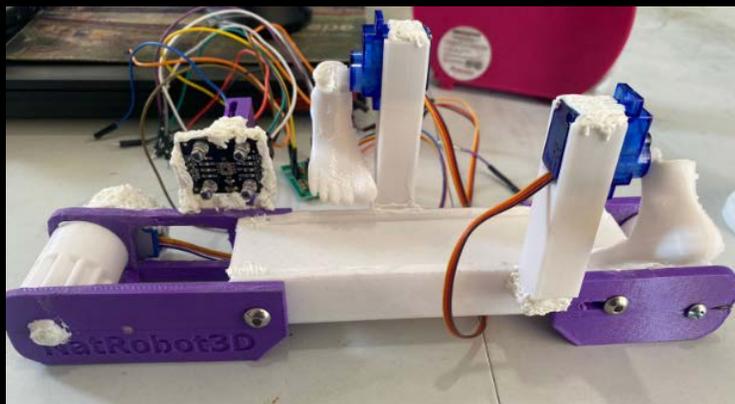


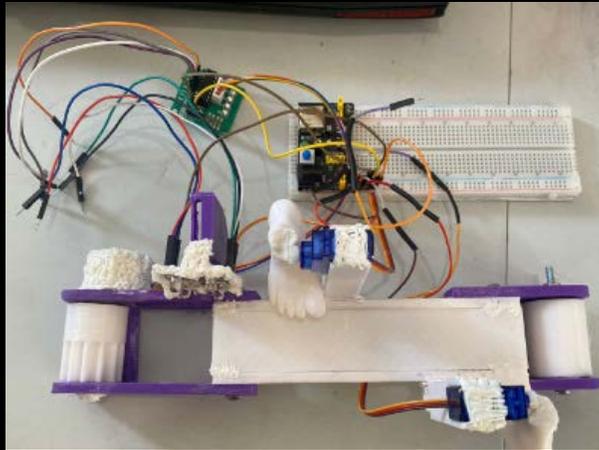
Luego se coloco el sensor de colores en la banda transportadora para poder hacer las pruebas de que el sensor si detectara los colores de las cajas al momento en que pasaban.



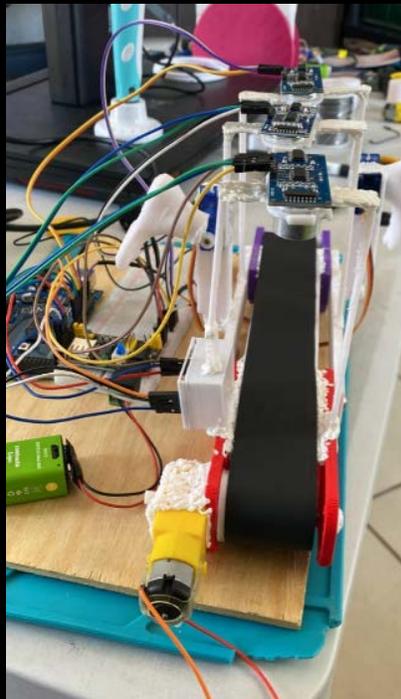


Luego se le coloco a la banda transportadora los servomotores con los pies que se imprimieron y estos van a servir que cuando se indique que la caja es azul se empuje al lado derecho, si es verde se empuja al lado izquierdo o si es amarillo dejar que se vaya recto, pero esto depende de a que lugar se desea que vaya la caja del color.

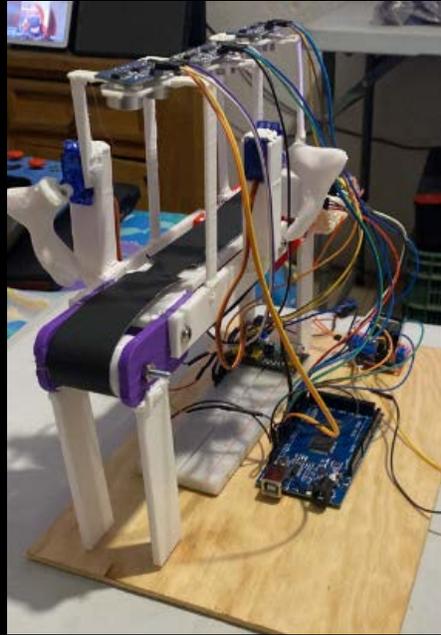




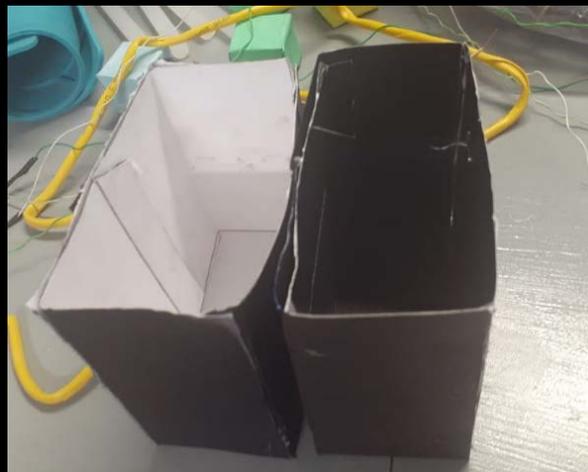
Luego se colocaron tres sensores ultrasónicos en la banda transportadora y se colocaron en cada sección donde está el sensor de color y donde están los servomotores para que cuando detecten la caja se active el sensor o los servomotores.



Luego se le colocaron los soportes de la banda transportadora para que la banda no se trabe y puede moverse con fluidez y también para que las cajas puedan caer en su lugar indicado.

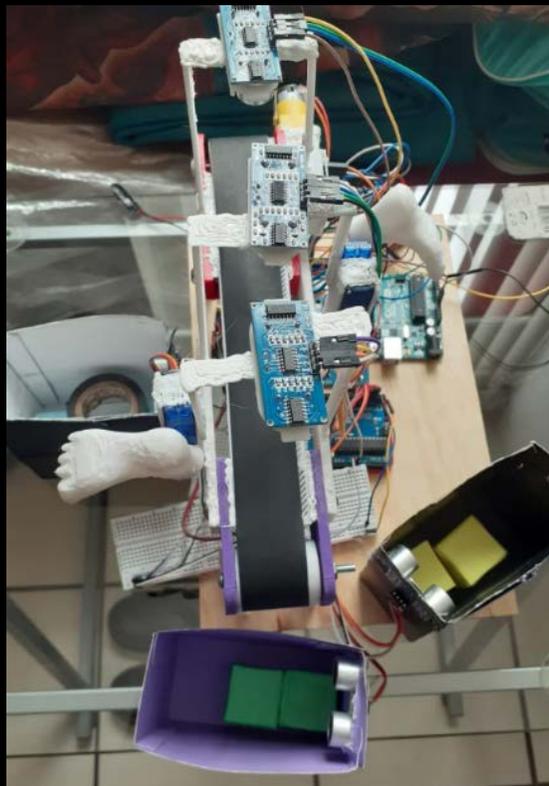


Luego se hicieron unas cajas con cartulina para que ahí cayeran las cajas de colores y se le colocaron un sensor ultrasónico para contar cuantas cajas caen en la sección azul, en la verde o en la amarilla.





Al final así es como quedaría la estructura de todo el clasificador de colores ensamblado para que pueda funcionar de manera correcta y así estaría completamente construido el proyecto final.



El sistema de la banda transportadora iría conectado en el Arduino Mega ya que se necesita muchas entradas y salidas digitales y en el Arduino Mega es donde va a ir el código completo de Arduino.

El programa de LabVIEW ira en un Arduino Uno que estará conectado con el Arduino Mega donde ahí se podrá hacer todo el sistema donde se podrá cambiar de manual a automático del clasificador de colores donde se podrá controlar de manera manual la banda y los servomotores.

Habrá un segundo programa de Arduino donde estará la parte del contador de cajas de los sensores ultrasónicos que están en las cajas de cartulina y este programa estará en un segundo Arduino uno ya que al momento de tener todo en el Arduino mega este se saturo y ya no pudieron funcionar los contadores de cajas de colores por eso se escogió poner todo el sistema de los contadores en un segundo Arduino uno para que pudiera funcionar bien y este también ira conectado con el Arduino uno con el sistema de LabVIEW.

Problemas

Para la elaboración del clasificador de colores hubo muchos problemas que provocaron que no funcionaran algunos sistemas del clasificador que se tuvieron que solucionar y en esta parte se mencionaran algunos de los problemas que ocurrieron y sus soluciones:

- Hubo saturación en el Arduino Mega y esto hacia que el sistema de contadores de cajas no indicara nada ósea que no contaba ningún objeto. Esto se solucionó agregando otro Arduino uno donde se pasó todo el código y los cables del contador de cajas.
- Hubo problemas con el motor a pasos de la banda ya que se trababa o las cajas eran muy pesadas que no las podía mover. Esto se soluciono cambiando el motor a pasos por un motorreductor.
- Hubo problemas con los servomotores para hacerlos funcionar de manera manual con el programa de LabVIEW. Este se soluciono colocando dos Arduino uno donde solo iría el código de Arduino y otro donde iría el programa de LabVIEW.

- Hubo problemas al momento de juntar todos los códigos en uno solo ya que algunas partes del código no funcionaban o funcionaban mal. Esto se tuvo que solucionar haciendo de nuevo el código para así poder arreglarlo mejor.
- Hubo problemas con el lector del sensor de colores para que nos indicara el color exacto de la caja y también para que detectara el color amarillo ya que solo había tres colores rojo, azul y verde y solo cambia el valor de su intensidad. Esto se solucionó modificando el código con if para que nos apareciera el color exacto de las cajas.
- Hubo problemas al momento de mostrar la cantidad de cajas que los sensores ultrasónicos leían ya que aparecían toda clase de símbolos excepto el conteo de las cajas. Esto se solucionó pasando el código del contador a otro Arduino uno solo para el contador de cajas.
- Hubo problemas con los tres sensores ultrasónicos colocados en la banda ya que los últimos dos no indicaban que la caja estaba ahí ya que en el código hacían falta partes del código para funcionaran los sensores. Esto se solución agregando el resto del código que faltaba.
- Hubo problemas para calibrar la banda para que fuera mas lenta ya que si iba mas lenta dejaba de funcionar la banda y también si no iba lenta no había tiempo para que los servomotores empujaran las cajas, ni tampoco habría tiempo para que el sensor de colores leyera el color de la caja.

Conclusión

Como conclusión podemos decir que el proyecto final tuvo demasiados problemas que hicieron que hubiera un gran retraso en la forma en la que se avanzaba para poder terminar aunque se logro terminar de construir la estructura y de conectar los circuitos no se logró cumplir con el objetivo del proyecto final ya que aunque la mayoría de los sistemas se lograron que funcionaran, no se logro que el proyecto funcionara al 100% ya que no se logro calibrar el motor para que fuera mas lento para que el sensor de colores pudiera leer el color de las cajas y para que el servomotor pudiera empujar las cajas a sus secciones y esto fue porque al momento en el que hacíamos que fuera lenta la banda transportadora y le poníamos una caja la banda se trababa y dejaba de funcionar por eso no se logro cumplir el objetivo del proyecto.

Referencias

(s.f.). Obtenido de <https://www.3ds.com/es/make/guide/process/3d-printing>

(s.f.). Obtenido de <https://www.thingiverse.com/thing:3915136>

(s.f.). Obtenido de <https://robots-argentina.com.ar/didactica/arduino-reconocer-colores-con-el-modulo-tcs230/>

(s.f.). Obtenido de <https://electrositio.com/sensor-de-color-funcionamiento-y-aplicaciones/>