

Carpeta de Campo

Thiago Arroyo

Proyecto: Selector de Caramelos

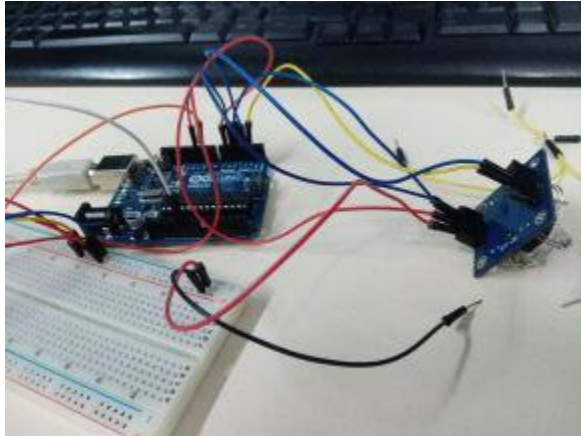
Año y división: 4°3

Año lectivo: 2021

Número del grupo: Grupo N°3

10/11/2021

Comencé buscando las conexiones del sensor RGB y su código correspondiente, el cual se encontró en el siguiente enlace diyodemag.com/projects/lolly_sorter, se hicieron las respectivas conexiones y se empezó a probar su funcionalidad.



17/11/2021

Continué con la programación del sensor y, con ayuda del profesor, logramos modificar el código para su funcionalidad manejando números en escala más chica que la predeterminada y se ajustaron un par de colores a modo de ejemplo de su funcionalidad.

```
#include <Servo.h>

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
Servo topServo;
Servo bottomServo;
int frequency = 0;
int color=0;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
```

```

    // Setting frequency-scaling to 20%
    digitalWrite(S0, HIGH);
    digitalWrite(S1, HIGH);
    topServo.attach(7);
    bottomServo.attach(8);
    Serial.begin(9600);
}
void loop() {

    color = readColor();

    delay(500);

    if(color == 1){
        Serial.println("Color: Verde");
    }
    if(color == 2){
        Serial.println("Color: Amarillo rancio");
    }
    if(color == 3){
        Serial.println("Color: Azul Arturo");
    }

    if(color == 4){
        Serial.println("Color: Maron Lo  Intento");
    }

    if(color == 5){
        Serial.println("Color: ROJO MEDIO MEDIO");
    }

    delay(500);
    color=0;
}
// Custom Function - readColor()
int readColor() {
    // Setting red filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
    // Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int R = frequency;
    // Printing the value on the serial monitor
    Serial.print("R= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");

```

```

delay(50);
// Setting Green filtered photodiodes to be read
digitalWrite(S2, HIGH);
digitalWrite(S3, HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int G = frequency;
// Printing the value on the serial monitor
Serial.print("G= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.print(" ");
delay(50);
// Setting Blue filtered photodiodes to be read
digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int B = frequency;
// Printing the value on the serial monitor
Serial.print("B= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.println(" ");
delay(50);

if(R==13 & G==12 & B ==13){
    color = 1; // Qcy
}
if(R==6 & G==7 & B ==9){
    color = 2; // Qcy
}

if(R==18 & G==15 & B ==10){
    color = 3; // Qcy
}

if(R==14 & G==17 & B ==13){
    color = 4; // Qcy
}

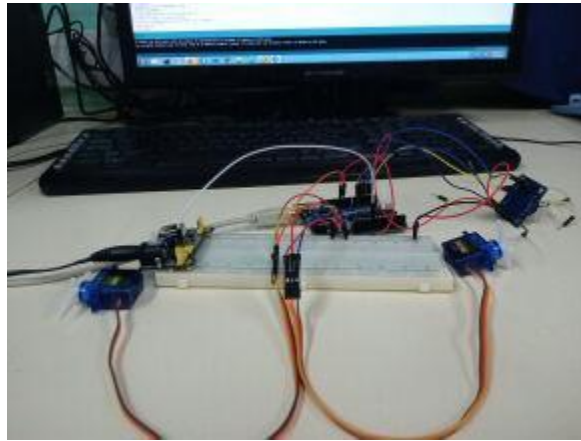
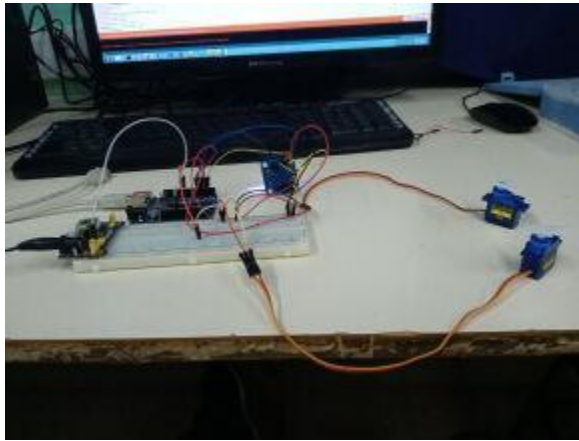
if(R==12 & G==15 & B ==12){
    color = 5; // Qcy
}

return color;
}

```

24/11/2021

Una vez terminado con la parte del sensor RGB, continué con la parte de los servo motores, conectando los mismos y modificando el código para cumplir con su funcionamiento, sin llegar a hacerlo por completo, pensamos con el profesor que fue una falla del servo motor, pero no llegamos a probar con otro.



Código

```
#include <Servo.h>

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
Servo topServo;
Servo bottomServo;
int frequency = 0;
int color=0;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  // Setting frequency-scaling to 20%
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  topServo.attach(10);
  bottomServo.attach(11);
}
```

```

    topServo.write(0);
    bottomServo.write(0);
    Serial.begin(9600);
}
void loop() {
    topServo.write(90);
    color = readColor();

    delay(1000);

    if(color == 0){
        Serial.println("Color: ");
        bottomServo.write(0);
        bottomServo.write(0);
        delay(1000);
    }

    if(color == 1){
        Serial.println("Color: Celeste");
        bottomServo.write(170);
        delay(1000);
    }

}

// Custom Function - readColor()
int readColor() {
    // Setting red filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
    // Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int R = frequency;
    // Printing the value on the serial monitor
    Serial.print("R= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");
    delay(50);
    // Setting Green filtered photodiodes to be read
    digitalWrite(S2, HIGH);
    digitalWrite(S3, HIGH);
    // Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int G = frequency;
    // Printing the value on the serial monitor

```

```

Serial.print("G= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.print(" ");
delay(50);
// Setting Blue filtered photodiodes to be read
digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int B = frequency;
// Printing the value on the serial monitor
Serial.print("B= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.println(" ");
delay(50);

bool carameloEncontrado = false;

if( (R>=15 && R<=24) && (G>=15 && G<=18) && (B>=8 && B<=12)){
    color = 1; // Rojo
    return color;
}

return 0;
}

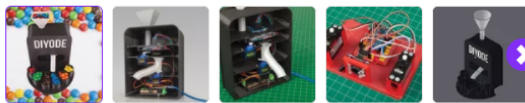
```

28/11/2021

Completé la carpeta de campo con el contenido poseído para la misma, procedí a realizar el [GitHub](#) con todos los detalles y archivos del proyecto.

Descargué el [modelo 3D](#) dado por el profesor y así utilizarlo para nuestro proyecto.

ARDUINO CONTROLLED COLOURED M&M'S LOLLY SORTING ROBOT



DOWNLOAD FREE

LIKE SAVE MAKE

★★★★★ ?

3.9k views 13 likes 70 downloads

3D MODEL DESCRIPTION

Do you save your favourite coloured M&M's to eat last, or are you constantly fighting over different colours with family and friends? Let us introduce you to the DIYODE Arduino Controlled Lolly Sorter, designed to stop the battle of the colours.

Designed by our Staff Writer Johann Wyss, and published in our magazine Issue 19, February 2019.

The full article can be found here: https://diyodemag.com/projects/lolly_sorter

3D PRINTER FILE INFORMATION

- > 3D design format: STL [Folder details](#)
- > Last update: 2019/06/03 at 10:20
- > Publication date: 2019/05/27 at 15:01

LICENSE

CC BY NC ND

TAGS

Modifiqué y guardé el código final para su uso en el proyecto

```
#include <Servo.h>

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
Servo topServo;
Servo bottomServo;
int frequency = 0;
int color=0;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  // Setting frequency-scaling to 20%
  digitalWrite(S0, HIGH);
  digitalWrite(S1, HIGH);
  topServo.attach(10);
```



```

    bottomServo.attach(11);
    topServo.write(0);
    bottomServo.write(0);
    Serial.begin(9600);
}
void loop() {
    topServo.write(90);
    color = readColor();

    delay(1000);

    if(color == 1){
        Serial.println("Color: Celeste");
        bottomServo.write(170);
        delay(1000);
    }

    color = 0;
}

// Custom Function - readColor()
int readColor() {
    // Setting red filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
    // Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int R = frequency;
    // Printing the value on the serial monitor
    Serial.print("R= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");
    delay(50);
    // Setting Green filtered photodiodes to be read
    digitalWrite(S2, HIGH);
    digitalWrite(S3, HIGH);
    // Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int G = frequency;
    // Printing the value on the serial monitor
    Serial.print("G= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");
    delay(50);
    // Setting Blue filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, HIGH);

```

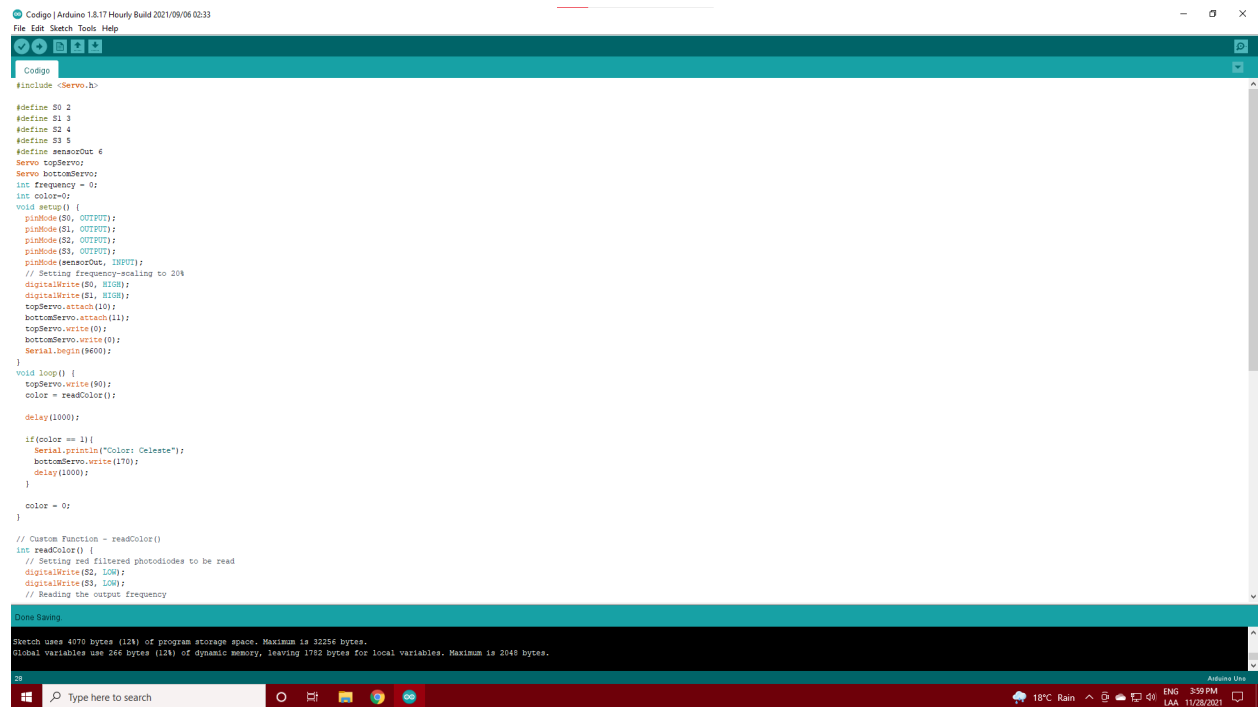
```

// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
int B = frequency;
// Printing the value on the serial monitor
Serial.print("B= "); //printing name
Serial.print(frequency); //printing RED color frequency
Serial.println(" ");
delay(50);

if( (R>=15 && R<=24) && (G>=15 && G<=18) && (B>=8 && B<=12)){
    color = 1; // Rojo
    return color;
}

return 0;
}

```



```

Codigo | Arduino 1.8.17 Hourly Build 2021/09/06 03:33
File Edit Sketch Tools Help

Codigo
#include <Servo.h>

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
Servo topServo;
Servo bottomServo;
int frequency = 0;
int color=0;
void setup() {
    pinMode(S0, OUTPUT);
    pinMode(S1, OUTPUT);
    pinMode(S2, OUTPUT);
    pinMode(S3, OUTPUT);
    pinMode(sensorOut, INPUT);
    // Setting frequency-scaling to 20K
    digitalWrite(S0, HIGH);
    digitalWrite(S1, HIGH);
    topServo.attach(10);
    bottomServo.attach(11);
    topServo.write(0);
    bottomServo.write(0);
    Serial.begin(9600);
}
void loop() {
    topServo.write(90);
    color = readColor();
    delay(1000);
    if(color == 1){
        Serial.println("Color: Celeste");
        bottomServo.write(170);
        delay(1000);
    }
    color = 0;
}

// Custom Function - readColor()
int readColor() {
    // Setting red filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
    // Reading the output frequency
}

```

Done Saving.

Sketch uses 4070 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 266 bytes (12%) of dynamic memory, leaving 1782 bytes for local variables. Maximum is 2048 bytes.

Windows taskbar: 18°C Rain, 3:59 PM, 11/28/2021

Realicé el análisis FODA en base a nuestro progreso y el proyecto en sí. Esto utilizando la plantilla publicada por el profesor vía classroom.

Análisis **FODA** aplicando en autoanálisis



Creé la portada del proyecto, rellenandola con la información básica sobre el proyecto, la institución, docente y, los integrantes.



Selector de Caramelos

Integrantes: Thiago Arroyo y Maximiliano Ortiz

Docente: Gonzalo Nicolas Consorti

Turno: Vespertino

Curso: 4°3