

# **IMHren TXANDAKAKO INGENIARITZA ESKOLA**

**ESCUELA DE INGENIERÍA  
EN ALTERNANCIA DEL IMH**

## **PROYECTO TECNOLOGÍAS INDUSTRIALES**

**Autores:** Alberto Michelena  
Ibai Konde

Mario Arrue

**Fecha entrega:** Elgoibar, 16/01/2020

**Asignatura:** Tecnologías Industriales



# Índice

1. Introducción.....	5
2. Microcontrolador Arduino.....	6
2.1. <i>Control de la cinta transportadora y conteo de piezas</i> .....	6
2.2. <i>Clasificador por colores</i> .....	10
3. Visualización de datos con Tableau.....	13
4. Conclusiones.....	14
5. Referencias.....	15
6. Bibliografía.....	16

## Índice de ilustraciones

Ilustración 1: Primera estación.....	5
Ilustración 2: Segunda estación.....	5

## **RESUMEN**

El siguiente informe es redactado a petición del Máster en Fabricación Digital del IMH, en la asignatura “Tecnologías Industriales” con el objetivo de profundizar en las tecnologías vistas en clase. El proyecto propone el desarrollo de una solución tecnológica para la clasificación y conteo de productos alimentarios, introduciendo diversas tecnologías como microcontroladores utilizando Arduino, la visualización de datos con Tableau, lectura de bases de datos tipo SQL y visión artificial con Open CV.

## 1.INTRODUCCIÓN

Para el desarrollo del proyecto se ha utilizado una maqueta constituida por dos estaciones. En la primera estación mostrada en la Ilustración 1 se ha utilizado una cinta transportadora y una cámara Raspberry para detectar los productos defectuosos utilizando la librería Open CV de Phyton y dando un aviso al operario para eliminar el producto defectuoso y contando la cantidad de cada color. Además, se ha utilizado un Arduino para controlar el movimiento de la cinta y detenerlo en función de la presencia del tanque de almacenamiento.



*Ilustración 1: Primera estación*

Para la segunda segunda estación se ha utilizado una maqueta presentada por Littlefrenchkev [1] con la que se ha realizado la clasificación por colores y el conteo de los productos. En la Ilustración 2 se muestra la maqueta utilizada para la clasificación.



*Ilustración 2: Segunda estación*

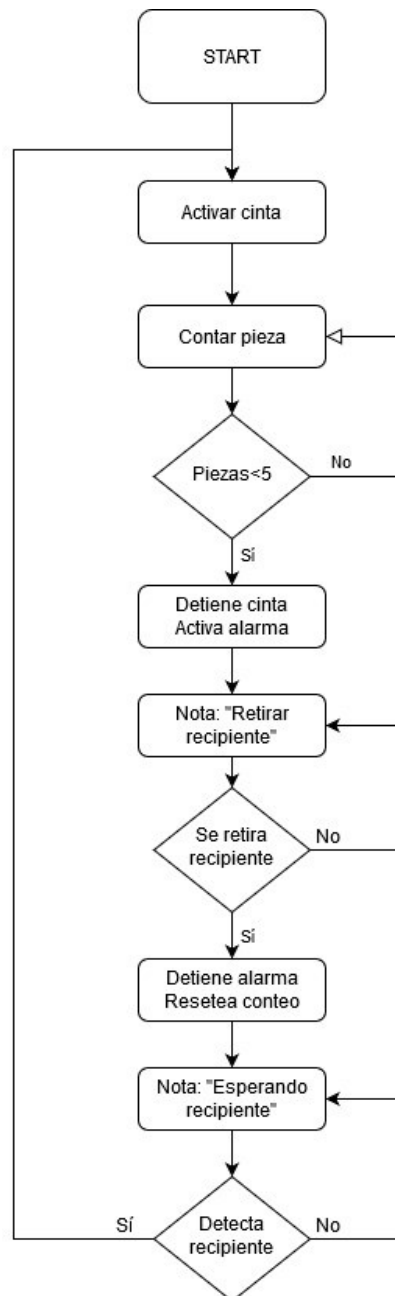
En los siguientes apartados se detallan las soluciones desarrolladas por el grupo para el proyecto.

## 2.MICROCONTROLADOR ARDUINO

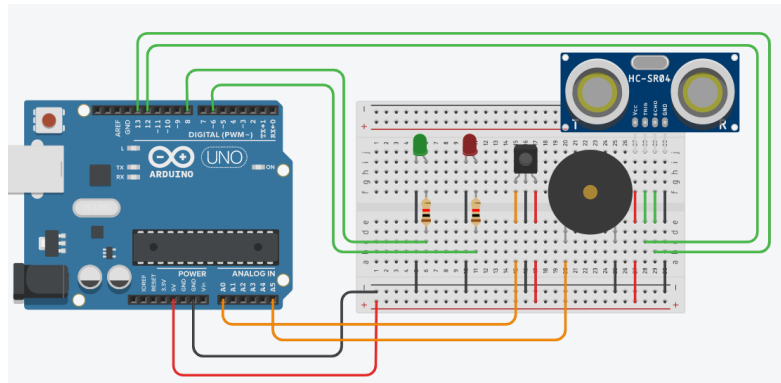
En el siguiente apartado se describe el código desarrollado para el control de la cinta transportadora en la primera estación y la clasificación de piezas para la segunda estación.

### 2.1.CONTROL DE LA CINTA TRANSPORTADORA Y CONTEO DE PIEZAS

El control de la cinta transportadora se ha realizado mediante un Arduino Uno R3 de manera que la cinta se encuentre en movimiento siempre que el depósito esté en su posición y no sobrepase el número de piezas necesarias. La secuencia de operaciones de la cinta es la siguiente:



En la siguiente imagen se muestra el esquema eléctrico de la estación 1.



En las siguientes líneas de código se puede apreciar la declaración de las variables del programa.

```
const int pinAdc = A0;    //el pin Analógico que recibe la señal del sensor
int count=0;             //inicializamos el contador
int led=8;               //el pin Digital donde esta el LED de lectura
int led2=6;              //el pin Digital donde esta el LED que simula la cinta
int pit=A5;              //el pin Analógico donde esta la alarma
int range=300;           //valor que activa el count
long sum = 0;            //inicializamos una variable de lectura del sensor
int res=5;               //El número de piezas que queremos contar antes de retirarlas
const int trigPin=12;    //Inicializar el emisor del sensor de proximidad
const int echoPin=13;    //Inicializar el receptor del sensor de proximidad
long duration;           //Variable tiempo sensor proximidad
int distance;            //Variable distancia sensor proximidad
int dis=6;
```

A continuación se muestra la función setup que se ejecutará una única vez. En ella se configuran las distintas salidas y entradas del Arduino.

```
// En el setup se especifican las condiciones iniciales
void setup(){
  Serial.begin(115200);
  Serial.println("Contador de piezas de Alkorta");
  pinMode(led, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(pit,OUTPUT);
  digitalWrite(led,HIGH);
  digitalWrite(led2,HIGH);
  analogWrite(pit,0);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

La función loop está constituida en primer lugar por varias líneas de código en las que se realiza el conteo de piezas mediante un micrófono tal y como se puede ver a continuación.

```
void loop()
{
    sum=analogRead(pinAdc);    //Se lee el valor del sensor y se guarda en sum
    if(sum>range)              //Si sum es mayor que el valor establecido previamente como
                                //límite quiere decir que ha caído una pieza
    {
        count+=1;              //El contador de piezas se incrementa en 1
        digitalWrite(led,LOW);  //Se apaga el LED
        Serial.print(count);
        if(count==1)Serial.print(" pieza"); else Serial.print(" piezas");
        Serial.println("");
    }
}
```

Una vez que el conteo de piezas llega al valor de cinco, se detiene la cinta y da un aviso mediante una alarma para retirar el recipiente. Una vez retirado, se detiene la alarma y espera hasta que el recipiente se encuentra en su posición, comenzado de nuevo el conteo.

```
if (count==5){    //Cuando el contador llega al límite de piezas que queremos salta la alarma
                  //que nos avisa que tenemos que retirarlas

    digitalWrite(led2,LOW);    //Se apaga el LED2
    Serial.println("Retirar piezas");
    distance=calcdist();

    while(distance<dis){    //Parte del código que emite un pitido hasta que se quita el recipiente
        analogWrite(pit,150);
        delay(500);
        analogWrite(pit,0);
        delay(500);
        distance=calcdist();
    }
    analogWrite(pit,0);
    count=0;
    while(distance>dis)
    {
        Serial.println("Esperando recipiente");
        distance=calcdist();
    }
}
```

A continuación se muestra la función utilizada para medir la distancia recorrida por la onda de ultrasonido hasta el recipiente. Esta distancia se utilizará para detectar la presencia del recipiente.



```
int calcdist()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

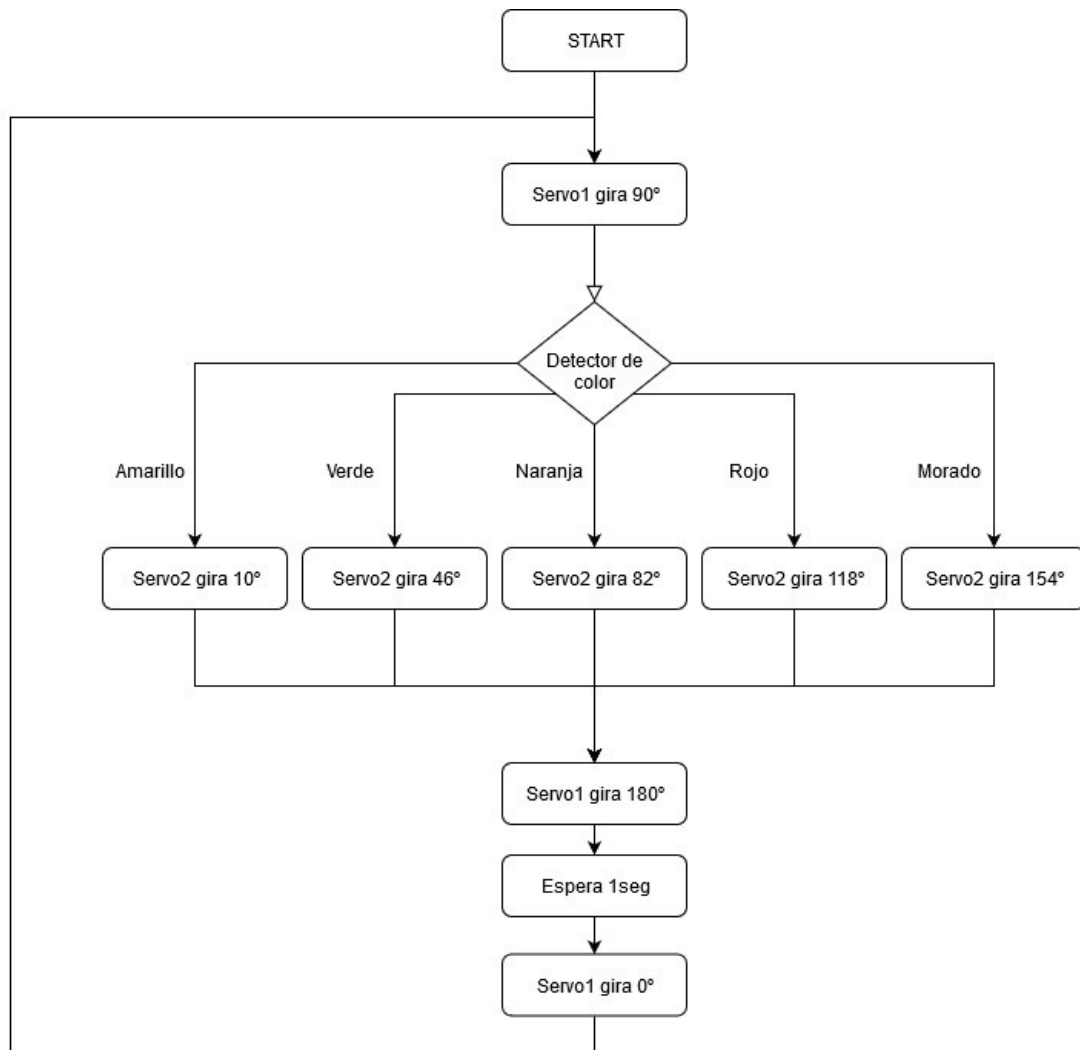
    duration=pulseIn(echoPin, HIGH);
    distance=duration*0.034/2;
    return (distance);
}
```

En el siguiente [enlace](#) se puede ver un video explicativo del la estación 1 y el archivo donde se encuentra todo el código.

## 2.2.CLASIFICADOR POR COLORES

El control del clasificador de colores se ha realizado mediante un Arduino Uno R3 detectando el color de cada Skittle mediante un sensor de colores y clasificando cada uno mediante dos servomotores.

La secuencia de operaciones del clasificador por colores es el siguiente:



En las siguientes líneas se describe el código utilizado y en el siguiente [enlace](#) se puede ver un video explicativo de la estación 2 y el archivo donde se encuentra todo el código.

En primer lugar, se han declarado las variables del programa y establecido los márgenes RGB de cada color. Al mismo tiempo se ha realizado el setup del sensor RGB.

```
Servo servomotor1;
Servo servomotor2;
uint16_t total_r, total_g, total_b, total_c;
//Límites de cada color
uint16_t black_limit_values[] = {0, 200, 0, 200, 0, 200};
uint16_t yellow_limit_values[] = {167, 220, 183, 251, 77, 102};
uint16_t green_limit_values[] = {55, 70, 115, 175, 47, 67};
uint16_t orange_limit_values[] = {140, 170, 70, 100, 40, 70};
uint16_t red_limit_values[] = {90, 140, 50, 100, 70, 125};
uint16_t purple_limit_values[] = {0, 30, 0, 35, 0, 35};
int grado=0;
float sum;
// setup RGB sensor
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_24MS, TCS34725_GAIN_1X);
```

En la función setup se han configurado las salidas del Arduino para los dos servos.

```
void setup() {
  Serial.begin(9600);
  while (!Serial) {}
  Serial.println("Red,Green,Blue");
  servomotor1.attach(4);//Configuración salidas servo
  servomotor2.attach(5);
}
```

A continuación en la función loop se inicia la secuencia de operaciones. En primer lugar, se coloca el servo en posición de recoger pieza. Después, gira 90° preparándola para la lectura de color, guardando el valor RGB de cada Skittle.

```
void loop() {
  servomotor1.write(0); //Servo 1 a 0
  delay(1000);
  servomotor1.write(90); //Servo 1 posición lectura
  delay(500);
  // wait for proximity and color sensor data
  if (tcs.begin()) { //check if sensor found
    Serial.println("Found sensor");
    // Obtención de datos RGB
    tcs.getRawData(&total_r, &total_g, &total_b, &total_c);
    Serial.print(total_r);
    Serial.print(',');
    Serial.print(total_g);
    Serial.print(',');
    Serial.print(total_b);
    Serial.println();
    delay (2000);
  }
}
```

Una vez obtenidos los valores RGB se comparan con los márgenes establecidos para cada color y en función de ello se asigna un grado de giro para el servo dos.

```
//Clasificación de color en función del RGB
if ((total_r >= yellow_limit_values[0]) && //check for yellow
    (total_r < yellow_limit_values[1]) &&
    (total_g >= yellow_limit_values[2]) &&
    (total_g < yellow_limit_values[3]) &&
    (total_b >= yellow_limit_values[4]) &&
    (total_b < yellow_limit_values[5]))
{Serial.print("Amarillo");
  grado=10;
}
```

Por último, se utilizan los grados calculados para girar el servo 2 permitiendo clasificar cada color en un recipiente.

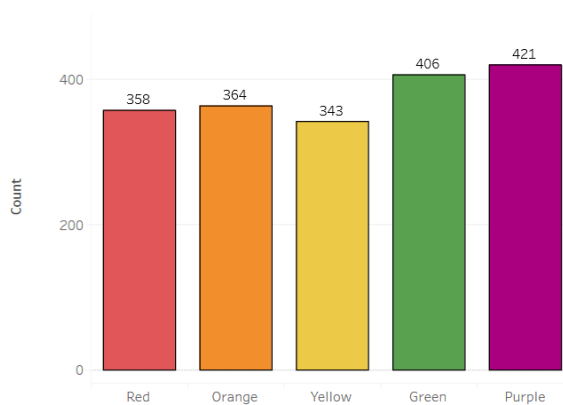
```
Serial.println();
servomotor2.write(grado); // Giro posicionador
delay (500);
servomotor1.write(180); // Deposición Skittles
delay(1000);
```

### 3. VISUALIZACIÓN DE DATOS CON TABLEAU

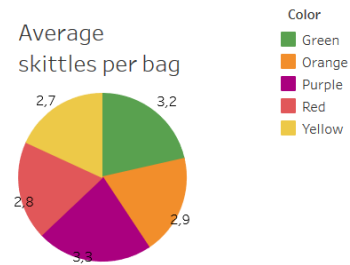
Para la parte de visualización de datos, se ha partido de un fichero .csv resultado de un estudio sobre la distribución de los colores de Skittles. Mediante la aplicación de Tableau se ha importado el fichero y se han realizado diversos gráficos sobre el estudio.

Por un lado, en el siguiente gráfico se ha mostrado el total de Skittles detectados para cada color mediante un gráfico de barras. Por otro lado, en el gráfico de tarta se puede apreciar la media de Skittles que se puede encontrar de cada color en un paquete.

Skittles Color War!!!! Which color will win?



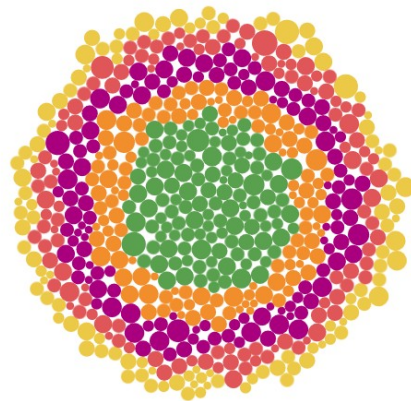
Average  
skittles per bag



Por otro lado, en la siguiente tabla se ha representado la medición realizada por cada sujeto del experimento, mostrando el número de Skittles de cada color y el total encontrados en cada bolsa. Por último, en la derecha de la imagen se muestra la distribución de colores.

Skittles per package

Name	Green	Orange	Purple	Red	Yellow	Total general
Aaliyah	4	3	2	2	4	15
Aaron	3	2	3	3	4	15
Abigail	2	0	4	2	3	11
Adalyn	4	3	3	3	2	15
Addison	4	5	1	4	1	15
Addysen	5	5	3	3	0	16
Adeline	3	2	2	5	2	14
Aiden	5	4	5	1	2	17
Alexander	3	3	3	3	3	15
Amelia	5	2	2	2	3	14
Andrew	3	1	1	4	4	13
Anna	2	3	5	3	0	13
Anne-Marie	2	4	4	3	1	14
Aria	4	2	3	4	2	15
Arianna	1	3	6	2	3	15



En el siguiente [enlace](#) se puede ver la publicación del tablero realizada vía web.

#### **4.CONCLUSIONES**

En conclusión, a través del presente trabajo se han puesto en práctica distintas tecnologías 4.0 estudiadas en la asignatura y se han plasmado en un proyecto real. En el se han conseguido construir dos maquetas que se han utilizado para realizar el conteo de piezas y control de una cinta transportadora y la clasificación de productos basados en un sensor de color. Por último, utilizando la aplicación de Tableau se ha realizado un dashboard publicado vía web resumiendo mediante representaciones gráficas un estudio sobre la distribución de los colores de un conjunto de paquetes de Skittles.

## 5.REFERENCIAS

[1] Littlefrenchkev. Colour Sorting Machine <<https://www.littlefrenchkev.com/colour-sorting-machine>> [Consulta: 14/01/2020]

## **6.BIBLIOGRAFÍA**

[1] IMH - Centro Avanzado de Fabricación. <<https://www.imh.eus/es>> [Consulta: 14/01/2020]