

Clasificador de colores:

Diseño del sistema

15/04/2021

Santiago García

Elena Diez

Andriana Dovhanych

Remi García

Sara Aparicio

Sergio Grande

Escuela Técnica Superior de Ingeniería y Diseño Industrial

Universidad Politécnica de Madrid

Curso 2020-2021 - Grupo A-104

Introducción

Resumiendo el documento anterior, el propósito final de esta máquina será clasificar y empaquetar los caramelos, metidos en una tolva, según el color y número elegido por el usuario.

Para ello, el sistema se basará en cuatro pistones accionados por servomotores, un sensor de colores, una cinta transportadora y un servo para controlar la caída de los caramelos.

Ya se explicó el funcionamiento mecánico esperado en el anterior documento. En este se procederá a explicar la programación del proyecto.

Historia de usuario

Como usuario de este clasificador de caramelos, quiero poder introducir en una consola el color y número de caramelos que deseo, y que en caso de necesitar que se introduzcan más caramelos, deberá avisar para poder reabastecer la máquina.

Estructura general de la parte de Visual Studio C++

Como se ha comentado, el objetivo de la parte del ordenador será servir como interfaz entre la máquina y el usuario. El programa en Visual Studio deberá de ser capaz de preguntar al usuario sobre el color de los caramelos y la cantidad de estos, para luego pasar la información a la arduino uno que controlará los pistones que separarán los dulces. Otro de los requisitos que debe cumplir es que sea capaz de informar al usuario cuando no hay suficientes caramelos de un tipo escogido en la tolva.

Para recoger la información del usuario, el programa de C++ deberá de, mediante el uso de bucles, deberá de preguntarle que colores desea filtrar, y después qué número desean. Respecto a la última parte, al ser un añadido relativamente reciente, no se dispone aún de una forma definitiva de realizar este conteo. Una de las opciones iniciales planteadas sería contar los colores que salen de la tolva y se escanean en el sensor de colores , e ir añadiendo al valor de una variable que cuente el número de caramelos de un color. Finalmente, si el sensor de colores no lee ningún valor distinto del negro de la cinta, deberá de mandar un mensaje solicitando se se reabastezca la tolva.

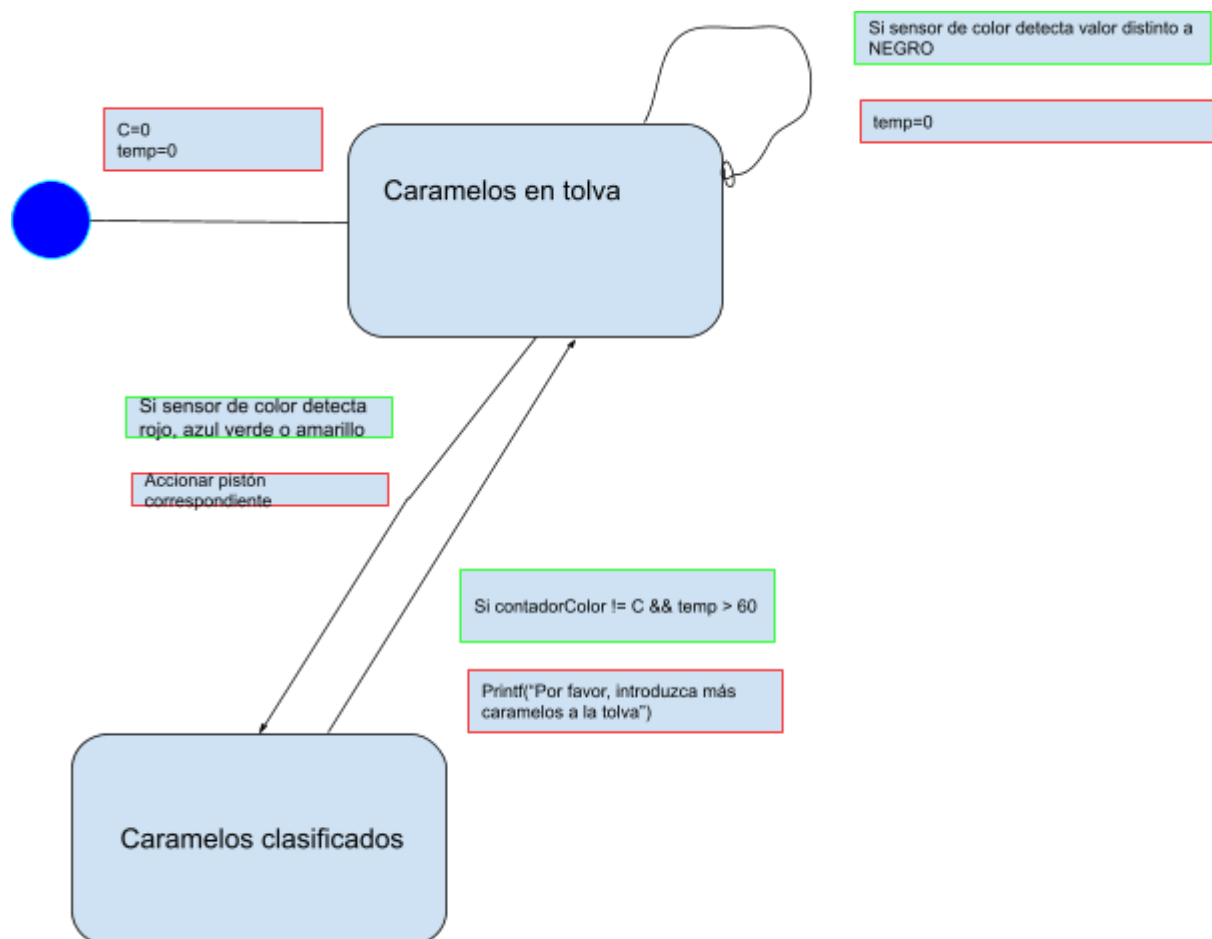
El programa que se muestra a continuación es el primero mencionado, que nos permite obtener del usuario el color y la cantidad de caramelos que desea.

```
#include <stdio.h>
#include <stdlib.h>
#define NCOLORES 4
int main(void) {
    char *colores[]= {"Rojo","Azul","Amarillo","Verde"};
    int cantidad[NCOLORES];
    int numero,i,j,color;
    // Inicializar la cantidad de cada color a cero
    for (i=0;i<NCOLORES;i++)
        cantidad[i]=0;
    do
    {
        printf ("Cuantos colores quiere clasificar:");
        scanf ("%d",&numero);
    } while (numero<0 || numero>NCOLORES);
    for (j=0;j<numero;j++) // Se repite tantas veces como colores quieran
    {
        for (i=0;i<NCOLORES;i++)
            printf ("%d - %s\n",i+1,colores[i]);
        do
        {
            printf ("Seleccione numero de color:");
            scanf ("%d",&color);
            if (cantidad[color-1]!=0)
                printf ("Ese color ya se ha seleccionado\n");
        } while (cantidad[color-1]!=0);
        printf ("Cantidad:");
        scanf ("%d",&cantidad[color-1]);
    }
    // Imprimir los resultados
    for (i=0;i<4;i++)
    {
        printf ("Color: %s Cantidad %d\n",colores[i],cantidad[i]);
    }
    return 0; }
```

Estructura general de la parte de Arduino

En el proyecto se usará la arduino uno con múltiples propósitos. Una vez lleguen los datos introducidos por el usuario en la consola, el servo colocado en la tolva irá soltando los caramelos de manera periódica. Estos caerán a una cinta, que estará continuamente desplazándose a velocidad constante, desde el momento en que se activa el servo por primera vez, hasta un minuto después de que se haya realizado la última lectura del sensor de color. Una vez pasen los caramelos por debajo del sensor de color, este deberá de leer el dulce y dar un valor, que será utilizado para activar uno de los cuatro pistones colocados más adelante en la cinta. Estos guiarán a los dulces hacia otros tubos que los llevará finalmente al envase.

Diseño gráfico



El programa que se muestra a continuación se trata del código para controlar el sensor de color TCS 3200, que realizará las medidas.

```
#define S0 4 // S0 a pin 4
#define S1 5 // S1 a pin 5
#define S2 6 // S2 a pin 6
#define S3 7 // S3 a pin 7
#define salidaTCS 8 // salidaTCS a pin 8

void setup() {
  pinMode(S0, OUTPUT); // pin 4 como salida
  pinMode(S1, OUTPUT); // pin 5 como salida
  pinMode(S2, OUTPUT); // pin 6 como salida
  pinMode(S3, OUTPUT); // pin 7 como salida
  pinMode(salidaTCS, INPUT); // pin 8 como salida

  digitalWrite(S0,HIGH); // establece frecuencia de salida
  digitalWrite(S1,LOW); // del modulo al 20 por ciento

  Serial.begin(9600); // inicializa monitor serie a 9600 bps
}

void loop() {
  digitalWrite(S2,LOW); // establece fotodiodos
  digitalWrite(S3,LOW); // con filtro rojo
  int rojo = pulseIn(salidaTCS, LOW); // obtiene duración de pulso de salida del sensor
  delay(200); // demora de 200 mseg

  digitalWrite(S2,HIGH); // establece fotodiodos
  digitalWrite(S3,HIGH); // con filtro verde
  int verde = pulseIn(salidaTCS, LOW); // obtiene duración de pulso de salida del sensor
  delay(200); // demora de 200 mseg

  digitalWrite(S2,LOW); // establece fotodiodos
  digitalWrite(S3,HIGH); // con filtro azul
  int azul = pulseIn(salidaTCS, LOW); // obtiene duración de pulso de salida del sensor
  delay(200); // demora de 200 mseg

  Serial.print("R:"); // muestra texto
  Serial.print(rojo); // muestra valor de variable rojo

  Serial.print("\t"); // espacio de tabulación

  Serial.print("V:"); // muestra texto
  Serial.print(verde); // muestra valor de variable verde

  Serial.print("\t"); // espacio de tabulación
```

```
Serial.print("A:"); // muestra texto
Serial.println(azul); // muestra valor de variable azul
    // y salto de línea
if ( rojo < 150 && verde > 300 && azul > 60){ // si valores dentro del rango
    Serial.println("ROJO"); // muestra texto
}
else if ( verde < 300 && rojo > 300 && azul > 80){ // si valores dentro del rango
    Serial.println("VERDE"); // muestra texto
}
else if ( azul < 50 && rojo > 200 && verde > 100){ // si valores dentro del rango
    Serial.println("AZUL"); // muestra texto
}
}
```