**RENKLERİNE GÖRE ÜRÜN AYIKLAMA ROBOTU**

**HAZIRLAYAN**

**ADI: MEHMET**

**SOYADI: ÖZEK**

İçindekiler

1. [Projenin Amacı 4](#_bookmark0)
2. [Sistemin Tasarlanması 4](#_bookmark1)
   1. [Karar Mekanizması 4](#_bookmark2)
   2. [Fritzing Şeması 4](#_bookmark3)
3. [Kullanılan Malzemeler 5](#_bookmark5)
   1. [Arduino Uno R3 5](#_bookmark6)
   2. [3D Baskı Stl Parçaları 6](#_bookmark8)
   3. [TCS34725 Rgb Renk Algılama Sensörü 7](#_bookmark10)
   4. [SG90 Servo Motor 8](#_bookmark12)
   5. [Jumper Kabloları 6](#_bookmark14)
4. [Otomasyonun Çalışması 9](#_bookmark22)
5. [Arduino Kodu 10](#_bookmark23)
6. [Visual Basic Arayüz Kodu 13](#_bookmark24)
7. [Projeden Görüntüler 16](#_bookmark33)
8. [Visual Basic Arayüz Tasarımı 20](#_bookmark33)
9. [Var Olan Projeler 21](#_bookmark29)
   1. [Örnek 21](#_bookmark30)
   2. [Örnek 21](#_bookmark31)
   3. [Örnek 22](#_bookmark32)
10. [Projemin Diğerlerinden Farkı 22](#_bookmark33)
11. [Projeden Çıkardıklarımız 23](#_bookmark33)
12. [Kaynakça 24](#_bookmark35)

**Şekiller Tablosu**

[Şekil 1 (Fritzing Şeması) 4](#_bookmark4)

[Şekil 2 (Arduino Uno R3) 5](#_bookmark7)

[Şekil 3 (3D Baskı Stl Parçaları) 6](#_bookmark9)

[Şekil 4 (TCS34725 Rgb Renk Algılama Sensörü) 7](#_bookmark11)

[Şekil 5 (SG90 Servo Motor) 8](#_bookmark13)

[Şekil 6 (Jumper Kabloları) 9](#_bookmark15)

[Şekil 7 (Projeden Görüntü(1)) 16](#_bookmark25)

[Şekil 8 (Projeden Görüntü(2)) 17](#_bookmark26)

[Şekil 9 (Projeden Görüntü(3)) 18](#_bookmark27)

[Şekil 10 (Projeden Görüntü(4)) 19](#_bookmark28)

[Şekil 11 (Projeden Görüntü(5)) 20](#_bookmark28)

# Projenin Amacı

Projenin amacı, Belirlediğimiz ürünleri renklerine göre ayırıp bunları bilgisayar ekranında hangi renge aitse kaç adet olduğunu ve tüm ürünlerin toplam miktarını ekranda yazdırmasıdır.

# Sistemin Tasarlanması

# Öncelikle mekanik kısmın parçalarını 3d yazıcıdan çıkartıp hazırlıyoruz.

# -Tüm parçaları hazırladıktan sonra montajına başlıyoruz. Önce dış parçaları bir tutkal tabancası kullanarak monte ediyoruz.

# - Daha sonra çok amaçlı yapıştırıcı kullanarak iki servo motoru platformlarına yapıştırıp montajını yapıyoruz.

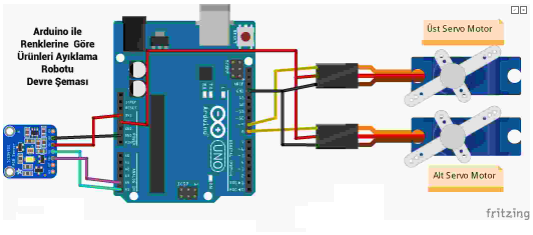
# - Bundan sonra yine bir yapıştırıcı kullanarak kılavuz rayı alt servo motorun yanı sıra üst servo motor için gereken destek ve platformu bağlıyoruz.

# - Ardından, Arduino'ya 5V adaptörle güç sağlamak için bir anahtar ve güç girişi takıyoruz ve üçüncü platformda renk sensörünü takıyoruz.

## **Karar Mekanizması**

Önceden huninin içine yerleştirdiğimiz pulları ara-yüz uygulamasından cihazı çalıştır butonuna tıkladığınızda sistem çalışmaya başlayacaktır. Üstteki servo motor hareket ederek pulu hazneye alır ve renk sensörünün hizasına getirir. sensör pulun rengini tespit ederek alttaki servo motora hangi bardağa yöneleceğinin açı bilgisini gönderir. Üstteki servo motor pulu ilerletir ve bardağa gönderir. Arayüz programında ise, ayrışan pulların hangi renkte ne kadar olduğu anlık olarak ekrana yazdırılır. Bütün pullar ayıklandığında arayüz programı sistemi otomatik olarak kapatır ve ekrana bilgilendirme mesajı gönderir.

## **Fritzing Şeması:**



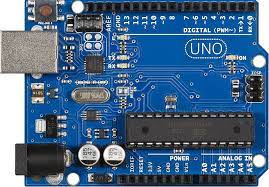
*Şekil 1 (Fritzing Şeması)*

# Kullanılan Malzemeler

## **Arduino Uno R3**

Arduino Uno ‘nun 14 tane dijital giriş / çıkış pini vardır. Bunlardan 6 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 6 adet analog girişi, bir adet 16 MHz kristal osilatörü, USB bağlantısı, power jakı (2.1mm), ICSP başlığı ve reset butonu bulunmaktadır. Arduino Uno bir mikrodenetleyiciyi desteklemek için gerekli bileşenlerin hepsini içerir. Arduino Uno ‘yu bir bilgisayara bağlayarak, bir adaptör ile ya da pil ile çalıştırabilirsiniz.

* Mikrodenetleyici: ATmega328P
* Çalışma Gerilimi: 5V
* Giriş Gerilimi (önerilen): 7-12V
* Giriş Gerilimi (limit): 6-20V
* Dijital G/Ç Pinleri: 14 (6 tanesi PWM çıkışı)
* Analog Giriş Pinleri: 6
* Her G/Ç için Akım: 40 mA
* 3.3V Çıkış için Akım: 50 mA
* Flash Hafıza: 32 KB (ATmega328) 0.5 KB kadarı bootloader
* SRAM: 2 KB (ATmega328)
* EEPROM: 1 KB (ATmega328)
* Saat Hızı: 16 MHz



*Şekil 2 (Arduino Uno R3)*

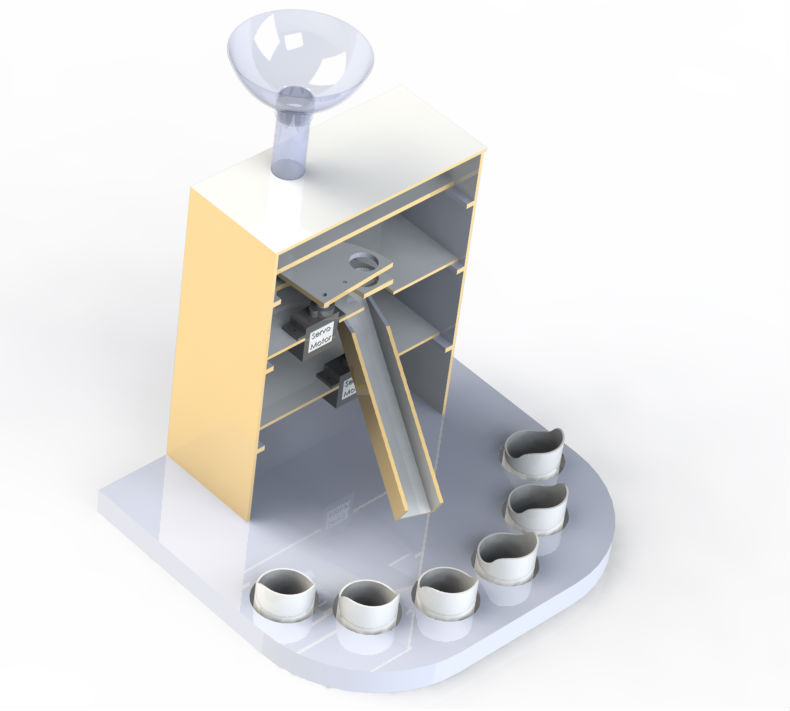
## **3D STL Baskı Parçaları**

STL en yaygın kullanılan 3D baskı dosyası formatıdır. Kısa ve kolay tanımıyla STL dosyaları 3D modellerin hacim bilgisini saklar. Bu format üç boyutlu bir nesnenin hacim bilgisini yalnızca herhangi bir renk, doku veya malzeme bilgisi içermeksizin saklanmasına ve aktarılmasına imkan vermektedir.

STL formatlı 3 boyutlu model daha sonra dilimleyici (Slicer) program kullanılarak katmanlara bölünür. Bu katmanların oluşturulabilmesi için gerekli iş yolu bilgisi ise Dilimleyici tarafından Gcode’ a dönüştürülür ve üç boyutlu yazıcıya aktarılır.

STL dosya formatı 3D modellerin üçgenler ile tanımlanmasıdır. Yukarıda görüldüğü gibi bir katı model üçgenler halinde kaydedilir.

Katı model bilgisini saklayabilen ve 3D Baskıya uygun halde aktarabilen 30’dan fazla dosya formatı bulunmaktadır. Bunun yanında en yaygın olarak kullanılan format STL’dir. İnternette STL olarak kaydedilmiş pek çok hazır modele ulaşabilir, neredeyse tüm modern CAD yazılımlarından bu formatla çıktı alabilirsiniz



*Şekil 3 (STL Parçalarının Birleştirilmiş Hali)*

## 

## **TCS34725 RGB Renk Algılama Sensörü**

Çip üzerine entegre edilmiş ve renk algılama fotodiyotlarına lokalize edilmiş bir IR engelleme filtresi, gelen ışığın IR spektrum bileşenini en aza indirir ve renk ölçümlerinin doğru bir şekilde yapılmasını sağlar. Filtre, insanlar IR'yi görmediğinden çoğu sensörden daha doğru bir renk elde etmenizi sağlar. Sensör, ayarlanabilir entegrasyon süresi ve kazancı ile inanılmaz 3,800,000: 1 dinamik aralığa sahiptir ve böylece koyu renkli camın arkasında kullanım için uygundur.

3.3V regülatör devresi bulunur, böylece 3-5VDC ile kesime güç verebilirsiniz ve 3.3V veya 5V mantıkla kullanılabilmeleri için I2C pinleri için seviye değiştirme yapabilirsiniz.

**Teknik Özellikler:**

* Boyutlar: 20.44 mm x 20.28 mm
* Ağırlık: 3.23 gr
* I2C adresi: 0x29



*Şekil 4 (TCS34725 RGB Renk Sensörü)*

## **SG90 Servo Motor**

## [Servo motorların](https://www.robotistan.com/servo) içerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüye ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.

**

*Şekil 5 (Hareket Algılama Sensö**rü(HC-SR50)*

## **Jumper Kabloları**

Bu tarz bağlantı kabloları, özellikle devre tahtası ile Arduino, EasyPIC gibi geliştirme kartlarının bir arada kullanıldığı devreler için oldukça uygundur. Uçlarında dişi ve erkek girişlerin olduğu üç çeşidi bulunmaktadır.

* Erkek-erkek,
* Erkek-dişi ve
* Dişi-dişi.

Bağlantı yapacağımız girişlere göre, bu çeşitlerden uygun olanları seçebiliriz. Ayrıca, farklı çeşitleri bir arada kullanmamız, bu kabloları birbirine bağlayarak, daha uzun iletim kabloları oluşturmamıza da olanak sağlar.



*Şekil 5 (Jumper Kabloları)*

# Otomasyonun Çalışması

Başlangıçta, hunide tutulan renkli pullar, üst servo motoruna takılı platforma düşer. Visual basic ile geliştirilen arayüz programı ile ürünlerin anlık takibi yapılmaktadır. Uygulamayı çalıştırıp. Arduinonun bağlı olduğu portu seçerek cihaza bağlan butonuna tıklayıp. Pulları huni içerisine bırakıp ve cihazı çalıştır butonuna tıkladığınızda sistem çalışmaya başlayacaktır. Üstteki servo motor hareket ederek pulu hazneye alır ve renk sensörünün hizasına getirir. sensör pulun rengini tespit ederek alttaki servo motora hangi bardağa yöneleceğinin açı bilgisini gönderir. Üstteki servo motor pulu ilerletir ve bardağa gönderir. Arayüz programında ise, ayrışan pulların hangi renkte ne kadar olduğu anlık olarak ekrana yazdırılır. Bütün pullar ayıklandığında arayüz programı sistemi otomatik olarak kapatır ve ekrana bilgilendirme mesajı gönderir.

# Arduino Kodu

#include <Wire.h>

#include <Servo.h> //Gerekli kütüphane tanımları

#include "Adafruit\_TCS34725.h"

Servo ustServo;

Servo altServo;

int renk = 0; //değişkenler tanımlandı

int veri, durum = false;

Adafruit\_TCS34725 sensor = Adafruit\_TCS34725(TCS34725\_INTEGRATIONTIME\_50MS, TCS34725\_GAIN\_1X); //renk sensörü tanımlandı

void setup() {

Serial.begin(9600);

if (sensor.begin()) { //sensör kontrolü

} else {

while (1); // Dur!

}

ustServo.attach(9); //üst serbo 9.pin ayarlandı

altServo.attach(8); //alt sevo 8.pin ayarlandı

ustServo.write(125); //üst servo 125 derece ayarlandı

delay(10);

}

void loop() {

veri = Serial.read(); //ardunio dan veri okundu

delay(10);

if ( veri == '1') //arayüzden calıstır butonu ile 8 değeri atanıyor

{

durum = true;

}

if ( veri =='0')

{

durum = false; //arayüzden durdur butonu ile 9 değeri atanıyor

}

if ( durum == true)

{

makine\_baslat(); //durum doğru olursa makine başlat fonksiyonu çağrılır.

}

if ( durum == false)

{

makine\_durdur(); //durum yanlıs olursa makine durdur fonksiyonu cağrılır.

}

}

void makine\_baslat()

{

delay(500);

for(int i = 125; i > 68; i--) {

ustServo.write(i); //üst servo sensöre göre ayarlandı

delay(5);

}

delay(500);

renk = renk\_oku(); //renk fonksiyonu çağrıldı

delay(50);

switch (renk) {

case 1:

altServo.write(10);

Serial.println(1);

break;

case 2:

altServo.write(25);

Serial.println(2);

break;

case 3:

altServo.write(50);

Serial.println(3);

break;

case 4: // RENK BELİRLENDİ ALT SERVO DERECESİ AYARLANDI

altServo.write(70);

Serial.println(4);

break;

case 5:

altServo.write(93);

Serial.println(5);

break;

case 6:

altServo.write(120);

Serial.println(6);

break;

default:

Serial.println(0);

break;

}

delay(300);

for(int i = 68; i > 25; i--) {

ustServo.write(i); // ürün alt servoya aktarıldı.

delay(5);

}

delay(200);

for(int i = 25; i < 125; i++) { // üst başlangıç derecesine döndü.

ustServo.write(i);

delay(5);

}

renk=0;

}

void makine\_durdur()

{

ustServo.write(ustServo.read()); // arayüzden durdur butonuna basıldı

altServo.write(altServo.read()); // son dönen açıda bekle

}

// renk okuma fonksiyonu

int renk\_oku() { // renk okuma fonksiyonu

uint16\_t clearcol, red, green, blue; // Renk değerleri için tanımlamalar

float average, r, g, b; // renk hesaplamaları için tanımlamalar

sensor.getRawData(&red, &green, &blue, &clearcol);

average = (red+green+blue)/3; //Renk degerlerini topla ve ortalamasını al.

r = red/average;

g = green/average; //renkleri ortalamaya böl

b = blue/average;

delay(50);

if((r < 1.5)&& (r > 1.35)) /\*&& ((g > 0.74) && (g < 0.85))&& ((b < 0.9)&& (b > 0.7)))\*/{

renk = 1;

}

if((r < 1.34)&& (r > 1.15))/\* && ((g > 1.24) && (g < 1.34))&&((b < 0.52&& (b > 0.42))))\*/{

renk = 2;

}

if(((r < 1)&& (r > 0.70)) && ((g > 1.40) && (g < 2))/\*&& ((b < 1)&& (b > 0.54))\*/){

renk = 3;

}

if(((r < 1.05)&& (r > 0.40)) &&/\*((g > 1.24) && (g < 1.39))&& \*/((b < 2.24)&& (b > 1.14))){ //değerlere göre renk döndürüyor

renk = 4;

}

if(((r < 2.2)&& (r > 1.8))/\*&& ((g > 0.5) && (g < 0.65))&& ((b < 0.55)&& (b > 0.4))\*/){

renk = 5;

}

if(((r < 1.79)&& (r > 1.51))&& ((g > 0.76) && (g < 1.2))/\*&&((b < 0.56&& (b > 0.45)))\*/){

renk = 6;

}

return renk;

}

# VİSAUL BASİC ARAYÜZ KODU

Imports System.IO

Imports System

Imports System.ComponentModel

Imports System.Threading

Imports System.IO.Ports

Public Class Form1

Dim veri1 = 0

Dim veri2 = 0

Dim veri3 = 0 ' RENK DEĞERLERİ İÇİN DEĞİŞKENLER TANIMLANDI ; '

Dim veri4 = 0

Dim veri5 = 0

Dim veri6 = 0

Dim portlar As Array

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

portlar = IO.Ports.SerialPort.GetPortNames()

For i = 0 To UBound(portlar)

ComboBox1.Items.Add(portlar(i))

Next

ComboBox1.Text = ComboBox1.Items.Item(0) 'COMBOBOX'A PORT SAYISI YAZDIRILDI '

Button2.Enabled = False

Button3.Enabled = False

Button4.Enabled = False

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click ' BAĞLAN BUTONU '

TextBox1.Text = 0

TextBox2.Text = 0

TextBox3.Text = 0

TextBox4.Text = 0

TextBox5.Text = 0

TextBox6.Text = 0

TextBox7.Text = 0

SerialPort1.PortName = ComboBox1.Text

SerialPort1.BaudRate = 9600 'SECİLEN PORTA BAGLANTI YAPILIYOR'

SerialPort1.DataBits = 8

SerialPort1.Parity = IO.Ports.Parity.None

SerialPort1.StopBits = IO.Ports.StopBits.One

SerialPort1.Open()

Button1.Enabled = False

Button2.Enabled = True

Button4.Enabled = True

Button3.Enabled = False

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click ' BAĞLANTI KES BUTONU '

SerialPort1.Close()

Button1.Enabled = True

Button2.Enabled = False

Button4.Enabled = False

Button3.Enabled = False

End Sub

Private Sub cmbPort\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

If SerialPort1.IsOpen = False Then

SerialPort1.PortName = ComboBox1.Text ' BAĞLANTI AÇIKKEN DEĞİŞİM OLMAZ '

Else

MsgBox("BAĞLANTIYI, KAPALIYKEN DEĞİŞTİRİNİZ", vbCritical)

End If

End Sub

Dim carpanx As Integer = 1

Dim carpany As Integer = 1

Dim sonportsayisi As Integer = 0

Private Sub Timer1\_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

portlar = IO.Ports.SerialPort.GetPortNames()

If sonportsayisi <> portlar.Length Then

ComboBox1.Items.Clear()

For i = 0 To UBound(portlar)

ComboBox1.Items.Add(portlar(i))

Next

ComboBox1.Text = ComboBox1.Items.Item(0)

sonportsayisi = portlar.Length

End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click 'CALISTIR BUTONU '

SerialPort1.Write("1")

Button4.Enabled = False

Button3.Enabled = True

Button1.Enabled = False

Button2.Enabled = False

'Timer1.Start()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click 'DURDUR BUTONU '

SerialPort1.Write("0")

Button1.Enabled = False

Button2.Enabled = True

Button4.Enabled = True

Button3.Enabled = False

'Timer1.Stop()

End Sub

Dim kontrol As Integer = 0 'SAYI DEĞİSKENİ TANIMLANTI

Private Sub SerialPort1\_DataReceived(sender As Object, e As SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

Try

Dim receiveddata As Integer = SerialPort1.ReadLine() 'porta yazdırdığımız renk okuma case 'i okunuyor değeri dönüyor '

Me.Invoke(Sub()

If receiveddata = 1 Then

veri1 = veri1 + 1

TextBox1.Text = veri1

TextBox8.Text = "pembe"

kontrol = 0

ElseIf receiveddata = 2 Then

veri2 = veri2 + 1

TextBox2.Text = veri2

TextBox8.Text = "sarı"

kontrol = 0

ElseIf receiveddata = 3 Then

veri3 = veri3 + 1

TextBox3.Text = veri3

TextBox8.Text = "yesil"

kontrol = 0

ElseIf receiveddata = 4 Then

veri4 = veri4 + 1

TextBox4.Text = veri4

TextBox8.Text = "mavi"

kontrol = 0

ElseIf receiveddata = 5 Then

veri5 = veri5 + 1

TextBox5.Text = veri5

TextBox8.Text = "kırmızı"

kontrol = 0

ElseIf receiveddata = 6 Then

veri6 = veri6 + 1

TextBox6.Text = veri6

TextBox8.Text = "turuncu"

kontrol = 0

ElseIf receiveddata = 0 Then

kontrol += 1

TextBox8.Text = "Ürün Yok"

End If

If kontrol > 0 Then

SerialPort1.Write("0")

Button1.Enabled = False

Button2.Enabled = True

Button4.Enabled = True

Button3.Enabled = False

MsgBox(" İşlem Tamamlandı ! ")

End If

TextBox7.Text = veri1 + veri2 + veri3 + veri4 + veri5 + veri6

End Sub)

Catch ex As Exception

End Try

End Sub

# End Class

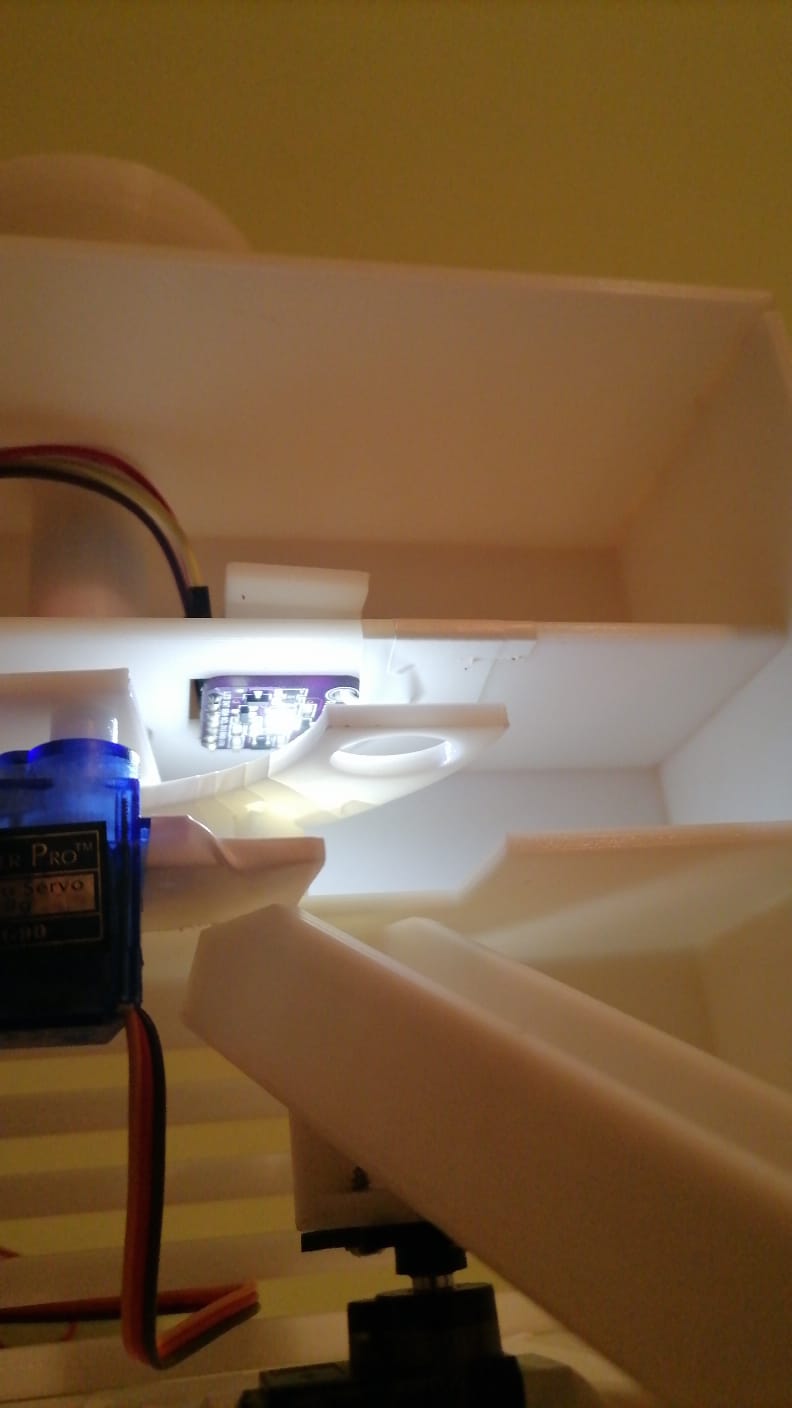
# PROJEDEN GÖRÜNTÜLER

# 

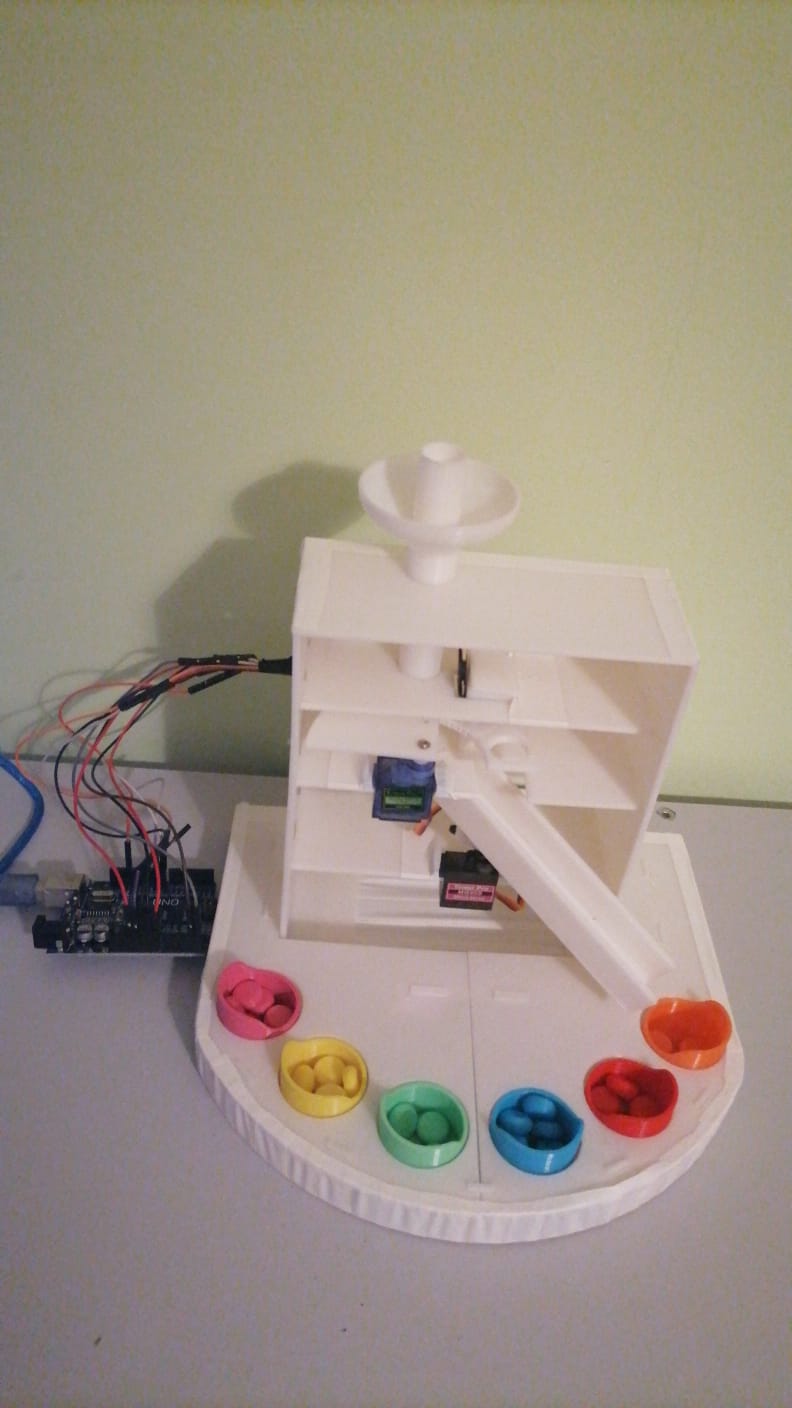
*Şekil 6 (Projeden Görüntü(1))*



*Şekil 7 (Projeden Görüntü(2))*



*Şekil 8 (Projeden Görüntü(3))*

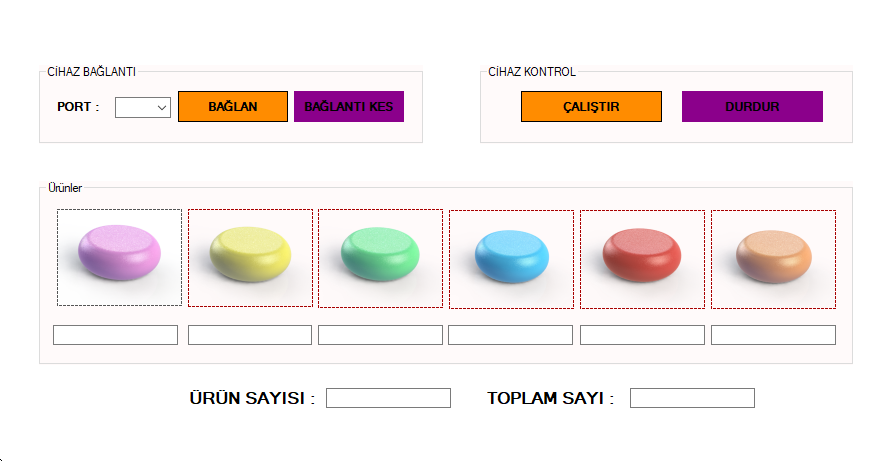


*Şekil 9 (Projeden Görüntü(4))*

1. **VİSAUL BASİC ARAYÜZ TASARIMI**

Port kısmının olduğu comboBoxtan arduinonun bağlandığı portu seçip

BAĞLAN butonuna tıklayıp arayüz ile arduino bağlantısını gerçekleştiriyoruz. Son olarak ÇALIŞTIR butonuna tıklayıp sistemi çalıştırıyoruz.

****

*Şekil 10 (Projeden Görüntü(5))*

# Var Olan Projeler

## Örnek

Sistemde renkli şekerler karışık olarak hazneden boşaltılıyor ve renk sensörü yardımıyla rengi belirlenip gerekli olan bardağa boşaltılıyor. Basit görülmesine rağmen müthiş bir mekanik ve programlama gerektiriyor. Hazır olarak yapmak basit ama düşünen gerçekten iyi düşünmüş. Şeker ayırma otomasyonu fabrikalarda ve birçok üretim tesisinde kullanılan sistemlere benzer ister bir ürün bandı olsun isterseniz bu şekilde bir huni olsun hiç fark etmez mantık aynıdır.

**Malzemeler:**

1- Arduino UNO  
2- ADJD-S311-CR999 renk sensörü  
3- Servo motor  
4- QRE1113 IR sensör  
5- Mekanik için 3d printer baskısı

Örnek Linki: <https://www.projehocam.com/arduino-seker-ayirma-otomasyonu/>

## Örnek

Makine 2 ana aşamaya ayrılabilir.

En üstte, sıralanmamış Skittles için rezervuarı bulabilirsiniz. Bu rezervuarın altında, kukaların toplanmasından ve renklerinin kontrol edileceği renk sensörünün önüne yerleştirilmesinden sorumlu besleme mekanizması bulunmaktadır. Besleme mekanizması, Skittles'ın hazne içinde sıkışmadan besleme mekanizmasının açıklığından güzelce akmasını sağlamak için haznenin içeriğini sallayan bir kol içerir (maalesef yine de bazen oluyor). Kukanın rengi bulunduğunda, makinenin ikinci aşaması devreye girer. Test tüpü tutucusu, kukanın düşmesi için sağ tüpü besleme mekanizmasının önüne hizalar. Bu döngü daha sonra rezervuar boşalana kadar tekrar edilebilir.

**Malzemeler:**

Arduino nano

1×TCS 34725 RGB sensörü

3×220 ohm dirençler

1×teller

1×2.1x5.5mm jak fişi

Örnek Linki: <https://hackaday.io/project/168885-color-sorting-machine>

## Örnek

Makine, renk sensörünü kullanarak optik ölçümler yaparak M & Ms ve Skittles'ı renge göre sıralar. Tek tek parçalar normal bir şekle ve benzer boyutlara sahip olduğu sürece her tür renkli nesneyi sıralamak için değiştirilebilir. 300 g'lık bir Skittles / M & Ms torbasını ayırmak yaklaşık 2-3 dakika sürer ve mekanizma saniyede yaklaşık 2 parça ayırır. Makine 250 mm çapında ve yaklaşık 300 mm yüksekliğindedir.

Haznede iken, üst besleme tüpünün girişinde tıkanmayı önlemek için şeker parçaları sürekli olarak karıştırılır. Tarayıcı, birbirinden 90 derece aralıklı dört yuvaya sahip küçük bir tekerlekten oluşur. Bir parça yuvaya girdikten sonra step motordan güç alan tekerlek 90 derece dönüyor. RGB sensörü daha sonra her biri 30 milisaniye süren üç ölçüm alır. Arduino denetleyicisi daha sonra öğenin rengini belirler (referans verilerine göre) ve parçayı doğru kaba yönlendirmek için çıkış tüpünü (ayrıca bir kademeli motor kullanarak) konumlandırır. Çıkış borusu hedef konumuna ulaşmadan hemen önce, çark parçayı düşürmek için 90 derece daha döner. İşlem daha sonra tekrar eder. İşlem sırasında, makineyi çevreleyen LED şerit kullanılarak görsel geri bildirim sağlanır.

Örnek Linki: <https://willemm.nl/mm-skittles-sorting-machine/>

# Projemin Diğerlerinden Farkı

Projemin diğerlerinden farkı, TCS34725 Rgb Renk Algılama Sensörünü kullanarak ürünleri renklerini ayırt edip sistemi kontrol ettiğimiz visaul basicde tasarladığımız arayüze verileri yollayıp istediğimiz o bilgileri ekranda yazdırıyoruz.

# Projeden Çıkardıklarımız

# Altta görüldüğü gibi arduino kodunun bir kısmında kırmızıyla gösterilen ilk başta yazdığımız kodları çıkardık. Bunun nedeni ise renkleri algılamada ve alt servonun çalışmasında probleme neden olduğu için çıkardık.

if((r < 1.5)&& (r > 1.35)) /\*&& ((g > 0.74) && (g < 0.85))&& ((b < 0.9)&& (b > 0.7)))\*/{

renk = 1;

}

if((r < 1.34)&& (r > 1.15))/\* && ((g > 1.24) && (g < 1.34))&&((b < 0.52&& (b > 0.42))))\*/{

renk = 2;

}

if(((r < 1)&& (r > 0.70)) && ((g > 1.40) && (g < 2))/\*&& ((b < 1)&& (b > 0.54))\*/){

renk = 3;

}

if(((r < 1.05)&& (r > 0.40)) &&/\*((g > 1.24) && (g < 1.39))&& \*/((b < 2.24)&& (b > 1.14))){ //değerlere göre renk döndürüyor

renk = 4;

}

if(((r < 2.2)&& (r > 1.8))/\*&& ((g > 0.5) && (g < 0.65))&& ((b < 0.55)&& (b > 0.4))\*/){

renk = 5;

}

if(((r < 1.79)&& (r > 1.51))&& ((g > 0.76) && (g < 1.2))/\*&&((b < 0.56&& (b > 0.45)))\*/){

renk = 6;

}

# Kaynakça

<https://bicisim.com/blog/stl-dosyasi-nedir-ve-nasil-kullanilir>

<https://www.direnc.net/tcs34725-rgb-renk-sensoru>

<https://maker.robotistan.com/rc-servo-motor-nedir/>,

<https://devreokulu.com/BaslangicBaglanti.html>

<https://interestingengineering.com/video/make-your-very-own-arduino-based-color-candy-sorting-machine>

<https://howtomechatronics.com/projects/arduino-color-sorter-project/>

<https://iotdesignpro.com/projects/iot-based-colour-sorting-machine-using-esp8266-and-thingspeak>

<https://create.arduino.cc/projecthub/user421848217/how-to-make-color-sorting-machine-8278c9>

<https://circuitdigest.com/microcontroller-projects/arduino-color-sorter-machine-using-tcs3200-color-sensor>