



**T.C.
SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**ÜRÜNLERİ RENGİNE GÖRE OTONOM
OLARAK AYIRAN ÜÇ EKSENLİ ROBOT
KOLU TASARIMI**

**EMİNE SELİN KOCAOĞLU
BİTİRME PROJESİ**

**Ocak-2020
KONYA
Her Hakkı Saklıdır**

Kontrol Edilecek Hususlar	Evet	Hayır
Sayfa yapısı uygun mu?	X	
Şekil ve çizelge başlık ve içerikleri uygun mu?	X	
Denklemler yazımları uygun mu?	X	
İç kapak, onay sayfası, Proje bildirimi, özet, abstract, önsöz ve/veya teşekkür uygun yazıldı mı?	X	
Proje yazımı; Giriş, Kaynak Araştırması, Materyal ve Yöntem (veya Teorik Esaslar), Araştırma Bulguları ve Tartışma, Sonuçlar ve Öneriler sıralamasında mıdır?	X	
Kaynaklar soyadı sırasına göre verildi mi?	X	
Kaynaklarda verilen her bir yayına proje içerisinde atıfta bulunuldu mu?	X	
Kaynaklar açıklanan yazım kuralına uygun olarak yazıldı mı?	X	
Proje içerisinde kullanılan şekil ve çizelgelerde kullanılan ifadeler Türkçe'ye çevrilmiş mi? (Latince ve Özel kelimeler hariçtir)	X	
Projenin içindekiler kısmı, proje içerisinde verilen başlıklara uygun hazırlanmış mı?	X	

Yukarıdaki verilen cevapların doğruluğunu kabul ediyorum.

Unvanı Adı SOYADI

İmza

Öğrenci : Emine Selin KOCAOĞLU

Danışman : Arş. Görevlisi Dr. Okan UYAR

BİTİRME PROJESİ KABUL VE ONAYI

Emine Selin KOCAOĞLU tarafından hazırlanan “Ürünleri Rengine Göre Otonom Olarak Ayıran Üç Eksenli Robot Kolu Tasarımı” adlı bitirme proje çalışması 08/01/2020 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği/oy çokluğu ile Selçuk Üniversitesi Teknoloji Fakültesi Elektrik Elektronik Mühendisliği bölümünde bitirme projesi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman

Arş. Görevlisi Dr. Okan UYAR

Üye

Prof. Dr. Mehmet ÇUNKAŞ

Üye

Prof. Dr. İsmail SARITAŞ

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mehmet ÇUNKAŞ
Elektrik Elektronik Mühendisliği
Bölüm Başkanı

PROJE BİLDİRİMİ

Bu projedeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve proje yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

İmza
Emine Selin KOCAOĞLU
08.01.2020

ÖZET
ÜRÜNLERİ RENGİNE GÖRE OTONOM OLARAK
AYIRAN ÜÇ EKSENLİ ROBOT KOLU TASARIMI

Emine Selin KOCAOĞLU

Selçuk Üniversitesi Teknoloji Fakültesi
Elektrik Elektronik Mühendisliği Bölümü

Danışman: Arş. Görevlisi Dr. Okan UYAR

2020, 50 sayfa

Jüri
Arş. Görevlisi Dr. Okan UYAR
Prof. Dr. Mehmet ÇUNKAŞ
Prof. Dr. İsmail SARITAŞ

Bitirme projesinin amacı 4 farklı renkte gelen ürünün rengini renk sensörü sayesinde algılayıp ilgili kutuya otonom olarak taşıyıp ve kutulara aktarılan ürünlerin sayısını ve rengini LCD ekran üzerinde gösteren gerektiği durumlarda uyarı yapan bir sistem olmasıdır. Kutuların belirlenen dolum sınırına ulaştığında “Kırmızı Kutu Doldu” gibi LCD üzerinden ve buzzer ile uyarı verip ürünler boşaltılıp sayacın sıfırlanması sağlanmaktadır. Test alanına ürün konulduğunu algılayacak sensör ve rengini algılayan sensörü sayesinde ürün ayırma işlemi başlamaktadır. Sistemin düzgün çalıştığını test etmek için bir test platformu geliştirilmiştir (Renkli bir parçayı ölçeklendirilmiş bir koordinattan alıp başka koordinattaki belli rengin toplandığı bir kutunun içine bırakma gibi). Proje’nin komutları PIC mikrodenetleyicisi kullanarak C dilinde yazılmıştır. Renk sensörü ve cisim algılama sensörü LDR ve LED’lerin kullanımı ile hazırlanmıştır.

Anahtar Kelimeler: Otonom, Robot kol, Sayaç, Sensör, Test platformu

ABSTRACT
THREE AXIS ROBOT ARM DESIGN CAN DIVIDE
COLORED OBJECT AUTONOMOUSLY

Emine Selin KOCAOĞLU

Selçuk University Faculty of Technology
Department of Electrical and Electronics Engineering

Advisor: Res. Assistant Dr. Okan UYAR
2020, 50 sayfa

Jüri
Res. Assistant Dr. Okan UYAR
Prof. Dr. Mehmet ÇUNKAŞ
Prof. Dr. İsmail SARITAŞ

The aim of the graduation project is to detect the color of the product coming in 4 different colors with color sensor and to carry it to the relevant box autonomously and to show the number and color of the products transferred to the boxes on the LCD screen. When the boxes reach the specified filling limit, “Red Box Filled” can be warned via LCD and buzzer and the products will be emptied and the counter will be reset. Product separation process starts with a sensor that detects the product has been placed in the test area and a sensor that detects the color. A test platform has been developed to test that the system is working properly (such as taking a colored part from a scaled coordinate and dropping it into a box where certain color in another coordinate is collected). The project's commands were written with C language using the PIC microcontroller. The color sensor and the object detection sensor are prepared using LDR and LEDs.

Keywords: Autonomous, Robot arm, Counter, Sensor, Test platform

ÖNSÖZ

Genel olarak robot kol uygulamasının amacı, insanlar tarafından ulaşımı ya da çalışma şartları zor olan ortamlarda ya da daha hızlı ve hatasız işlemler yapılmak istendiğinde, insan gücünün kullanılmasının tercih edilmediği ortamlarda bilgi toplama ve müdahale etmektir. Böylece işlemler daha sağlıklı, güvenilir ve yüksek performansta gerçekleşmektedir.

Diğer bir yandan renk sensörlerinin kullanım alanları çok fazla olmakla beraber paketleme endüstrisinde renk sensörleri ürünleri sınıflandırmada kullanılabilmektedir. Bunu yaparken de yüksek hızlı üretim talebini de karşılayabilmektedirler. Birden fazla renkte ürünlerin montajının yapıldığı tesislerde renk sensörleri ile kaliteli üretim garanti edilir.

Robot kol ve renk sensörünün birlikte kullanılmasıyla işlemlerin PIC üzerinde programla belirtmiş olduğum hareket servo motorlar yardımıyla gerçekleştirilmekte ve robot kol algıladığı cisim, o cisim için belirlenen kutuya bırakmaktadır. Günümüzde amacına göre birçok farklı alanda kullanılan bir projedir. Bu tez çalışmasında bitirme projesinin yapılma amacı yöntemleri ve elde edilen sonuçları yazılmıştır.

İlk olarak tez konusunu seçerken isteklerimi göz önünde bulundurup bana yardımcı olan tez danışmanım Arş. Görevlisi Dr. Okan UYAR hocama teşekkürlerimi sunarım. Son olarak tüm eğitim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen bu günlere gelmemde her türlü fedakarlığı gösteren ve her zaman yanımda olan sevgili aileme teşekkürlerimi bir borç bilirim.

Emine Selin KOCAOĞLU

İÇİNDEKİLER

PROJE BİLDİRİMİ	i
ÖZET	ii
ABSTRACT.....	iii
ÖNSÖZ	iv
İÇİNDEKİLER	Hata! Yer işareti tanımlanmamış.
SİMGELER VE KISALTMALAR	Hata! Yer işareti tanımlanmamış.i
1. GİRİŞ VE KAYNAK ARAŞTIRMASI	Hata! Yer işareti tanımlanmamış.
2. TEORİK ESASLAR.....	Hata! Yer işareti tanımlanmamış.
2.1. Mikrodenetleyiciler.....	Hata! Yer işareti tanımlanmamış.
2.2. Servo Motorlar	4
2.3. Sensörler.....	6
2.4. Projede Kullanılan Elektronik Bağlantılar ve Malzemeler	7
2.4.1. Osilatör Bağlantısı	7
2.4.2. MCLR Bağlantısı	7
2.4.3. ICSP (In Circuit Serial Programming) Bağlantısı	8
2.4.4. LCD Bağlantısı	8
2.4.5. Buzzer Bağlantısı.....	9
3. ARAŞTIRMA BULGULARI VE TARTIŞMA	10
3.1. Sensör Tasarımı.....	10
3.1.2. Deney ile veri elde etme	13
3.1.3. Sensör çalıştırma programı	14
3.2. Servo Motor Çalıştırma Deneyi	17
3.3. Devre Tasarımı.....	18
3.4. Robot Kol Tasarımı.....	20
4. SONUÇLAR VE ÖNERİLER	24
KAYNAKLAR	25
EK-1 Proje İçin Yazılan Program.....	26
ÖZGEÇMİŞ	40

SİMGELER VE KISALTMALAR

ADC: Analog-to-digital converter, analog dijital çevirici

CPU: Central Process Unit

DAC: digital to analog converter, dijital analog çevirici

ICSP: In Circuit Serial Programming, devre üzerinde seri programlama

LDR: light dependent resistor, ışığa duyarlı direnç

PWM: pulse width modulation, Sinyal Genişlik Modülasyonu

RAM: Random Access Memory, Rastgele Erişim Belleği

ROM: Read Only Memory, Sadece Okunan Bellek

V: VOLT

1. GİRİŞ VE KAYNAK ARAŞTIRMASI

Günümüzde endüstri 4.0. 'ın etkisiyle insan kas gücünün azalması istenmektedir. Robot Kollarının(Robotic Arm) endüstriyel üretim tesislerinin temel taşı haline gelmesi de bundandır. Hidrolik, pnömatik ve elektrikli, çeşitli motor sistemleriyle üretilen bu sistemler, endüstrinin birçok alanında kullanılmaktadır. Robot kolun ne yapacağı da çalışması kadar önemlidir. Günümüzde montaj, seri üretim gibi aşamalarda robot kol ile birlikte çalışması sağlanan sensörler de üretilmektedir. Bu bitirme projesinde amaçlanan da bu iki teknoloji ayrı olarak kullanılsa da projede birlikte kullanarak amaçlanan renk ayırma işlemini kolaylıkla gerçekleştirmesi için bir tasarım hazırlamaktır. Bazı sistemlerde renk ayırma işlemi bir başka makine ile gerçekleştirilirken bu projede sensörün istenilen yere örneğin bir konveyör bant sistemine eklenerek renk tanıma işlemi yaparak cisimleri istenilen yere koyma özelliği vardır. Bu sayede iki işlemin bir projede gerçekleştirilmesi sağlanır.

2. TEORİK ESASLAR

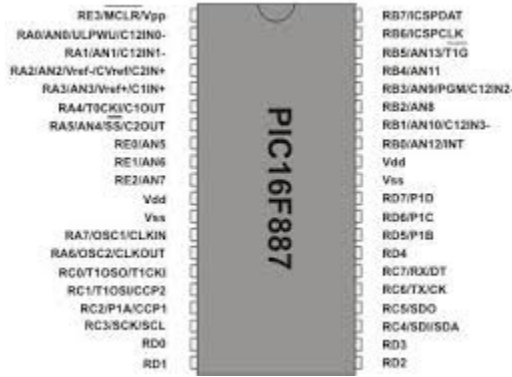
2.1. Mikrodenetleyiciler

Mikrodenetleyici programlanabilme, bir programı içerisinde depolayıp daha sonra çalıştırabilme özelliklerine sahip tek bir chip 'ten oluşan bilgisayardır. Bu özelliği mikrodenetleyicileri mikroişlemcilerden ayıran özelliğidir.

Mikrodenetleyicilerde bir CPU (Central Process Unit) , RAM (Random Access Memory) ROM (Read Only Memory) , input - output (giriş - çıkış I/O) uçları , seri ve paralel portlar , sayıcılar (counter) ve bazı mikrodenetleyicilerde de Analog 'dan Digital 'e (ADC) ya da Digital 'den Analog 'a (DAC) çeviriciler (konvertör) bulunur. Mikroişlemciler kullanılarak oluşturulan sistemlerde ise (örneğin kullandığımız bilgisayarlar) bu özelliklerin her biri için ayrı mikroişlemci kullanılır.

Projede kullandığım denetleyici microchip firmasının ürettiği PIC 16F887 mikrodenetleyicisidir. Seçilme sebepleri;

- Çalışma hızı daha yüksektir. (- 20 Mhz)
- Bellek yapısı daha geniştir. (256 byte)
- Giriş / çıkış port sayısı daha fazladır. (33)
- ADC çeviriciye sahiptir.



Şekil 2.1. 16F887'nin pin diyagramı

Çizelge 2.1. 16F887’de kullanılan pinler

OSC1/CLKIN	13	Kristal osilatör girişi/Harici osilatör kaynağı girişi
OSC2/CLKOUT	14	Kristal osilatör çıkışı
MCLR	1	Mikrodenetleyici için reset ucu
RA0/AN0	2	Analog giriş görevi yapar.
RA1/AN1	3	Analog giriş görevi yapar.
RA2/AN2/VREF-	4	Analog giriş veya negatif referans girişim görevi yapar.
RB0/INT	33	Harici kesme ucu görevi yapar.
RB1	34	Giriş/Çıkış ucu
RB2	35	Giriş/Çıkış ucu
RB3/PGM	36	Düşük seviye programlama girişi
RB4	37	G/Ç
RB5	38	G/Ç
RB6/PGC	39	Seri programlama girişi görevindedir.
RB7/PGD	40	Seri programlamada data girişi görevi yapar.
RC0/T1OSO/T1CKI	15	TIMER1 osilatör çıkışı/clock çıkışı görevi yapar.
RD4/PSP4	27	Data uçları
RD5/PSP5	28	
RD6/PSP6	29	
RD7/PSP7	30	
RE0/RD/AN5	8	Paralel slave porttan okuma kontrolü veya 5.analog girişi görevi yapar. LCD için kullanılmıştır.
RE1/WR/AN6	9	Paralel slave porttan yazma kontrolü veya 6.analog girişi görevi yapar. LCD için kullanılmıştır.
RE2/CS/AN7	10	Paralel slave porttan seçim kontrolü veya 7.analog girişi görevi yapar. LCD için kullanılmıştır.
VSS	12,31	Mikrodenetleyici için toprak seviyesini oluşturur.
VDD	11,32	Mikrodenetleyici için pozitif kaynak gerilimini oluşturur.

2.2 Servo Motorlar

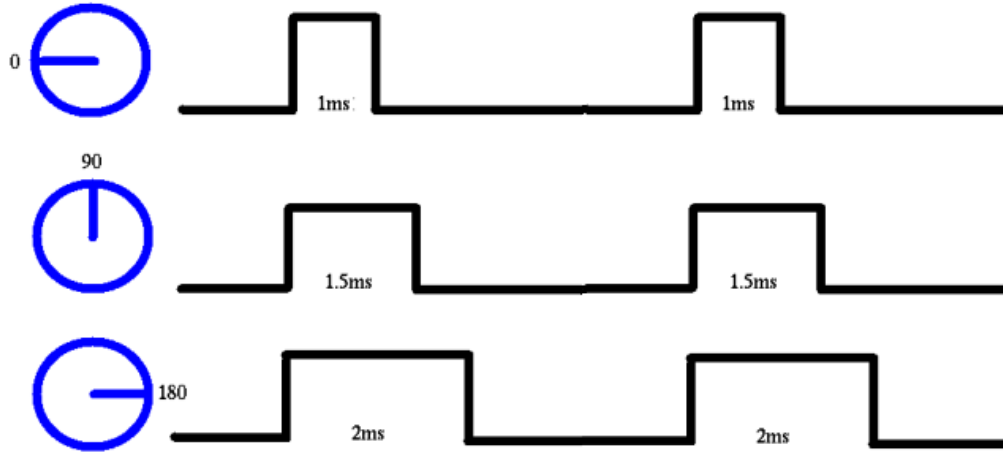
Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzendir. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.

Servo motorların 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 sürücüyü ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6Volt(V) gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.



Şekil 2.2. Servo motor

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolciden veya uzaktan kumandadan sağlanabilmektedir. Servo, her 20 ms (milisaniye) içerisinde bir pals değeri okumaktadır. Pals uzunluğu motorun dönüşünü belirlemektedir. Örnek olarak 1.5 ms'lik bir pals, motorun 90 derece pozisyonunu almasını sağlayacaktır (Nötr Pozisyon). Servolar hareket etmeleri için bir komut aldıklarında önce istenilen pozisyona hareket ederler, sonrasında ise o pozisyonda kalırlar. Servolar bulundukları pozisyonu korurken kendilerine dışarıdan bir güç uygulandığında bu güce direnirler. Bulundukları konumu sonsuza kadar koruyamazlar, pozisyonlarını koruyabilmeleri için palsin tekrar edilmesi gerekebilir. Hareket etmeleri için gereken pals genişliklerinin minimumları ve maksimumları vardır ve bu değerler değişkendir. Fakat genellikle minimum pals genişliği 1 ms, maksimum pals genişliği ise 2 ms'dir.



Şekil 2.3. Servo motor

Çok fazla marka ve çeşitte servo motor olsa da projede Tower Pro MG995 Servo Motor kullanıldı. Kullanılma sebepleri;

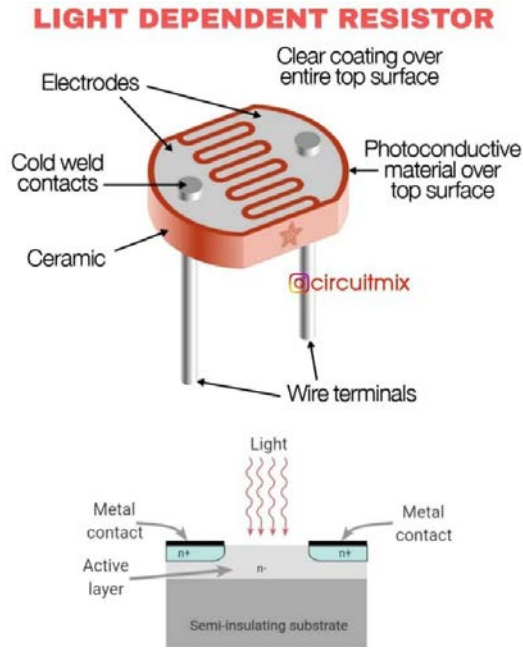
- Çalışma Voltajı: 4.8~ 6.6v
- Tutunma Torku: 9.4kg/cm (4.8v)
- Tutunma Torku: 11kg/cm (6v)
- Ve muadillerine göre daha uygun olmasıdır.

2.3. Sensörler

Tcs3200 , AMS – TSL257 – LF gibi renk ayırma amaçlarında da kullanılan ürünler yerine yapılan araştırmalar sonucunda renk sensöründe LDR ve LED kullanarak LDR üzerine düşen ışığın şiddeti ile değişen direnç değerleriyle bir çıkış sağlayarak mikrodenetleyiciye ADC aracılığıyla bilgi vermesi üzerine bir sistem yapıldı.

LDR ışığa duyarlı bir devre elemanıdır. Üzerine düşen ışık şiddeti ile ters orantılı bir çalışma prensibine sahiptir. Yani üzerine düşen ışık şiddeti arttıkça sahip olduğu direnç değeri azalır, ışık şiddeti azaldıkça sahip olduğu direnç değeri artar. LDR'ler sahip oldukları direnç değerlerinin değişmesi ile bir anahtarlama görevi görürler.

Projede LDR kullanarak hem renk hem de cisim algılama sensörü tasarlanmıştır.

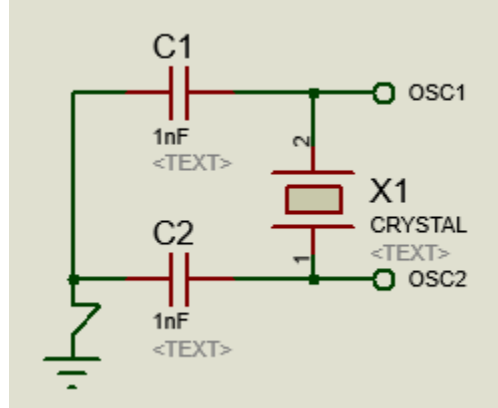


Şekil 2.4. LDR

2.4. Projede Kullanılan Elektronik Bağlantılar ve Malzemeler

2.4.1. Osilatör bağlantısı

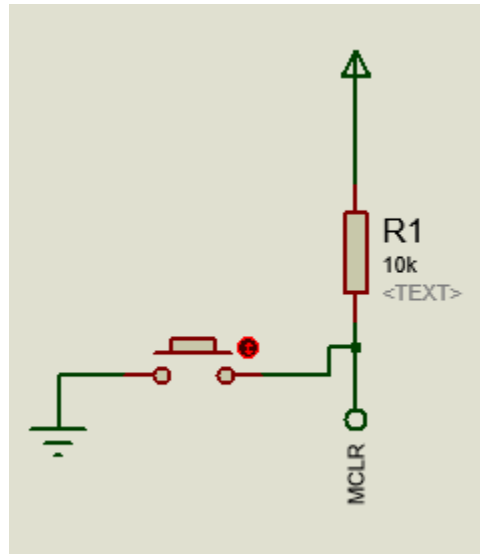
Osilatör, elektronik devrelerde kare, üçgen ve testere gibi sinyalleri üreten bir elektronik düzendir. En önemli özelliği sabit frekansta kalabilmesidir.



Şekil 2.5. Osilatör Devresi

2.4.2. MCLR bağlantısı

Mikrodenetleyiciyi gerektiği durumlarda resetleme(yenileme) yapılabilmek amacıyla kullanılan devredir. PIC 16F887 ‘de direk olarak 5V ile bağlanabileceği gibi devreye buton koyularak kontrolü de gerçekleştirilebilir.

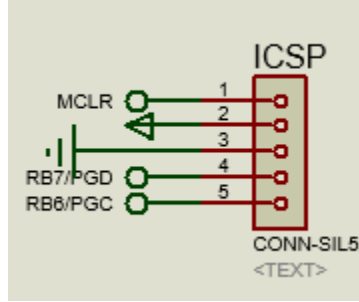


Şekil 2.6. MCLR Devresi

2.4.3. ICSP (In Circuit Serial Programming) bağlantısı

ICSP devre üzerinden programlamak için yapılan bağlantıdır. Bu mikrodenetleyiciyi programlarken, kod atmak için her seferinde yerinden söküp takmak yerine sadece kablo yardımıyla devre üstünden program atmaya yarar sağlar. Böylece mikrodenetleyici'nin bacaklarının zarar görmesini engellemiş olur ve bu zahmetli iş yapılmamış olur.

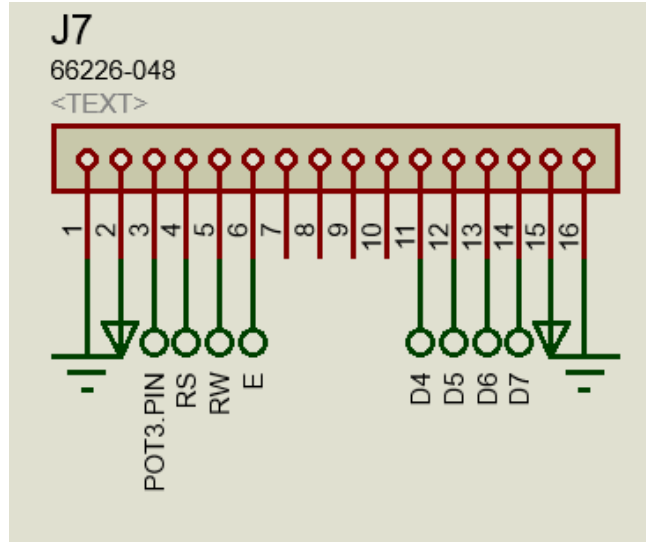
Proje kodlarını derlemekte kullanılan PIC KIT 3 derleyicisinin bağlantısına göre devre kartımda ICSP bağlantısı oluşturdum.



Şekil 2.7. ICSP Bağlantısı

2.4.4. LCD Bağlantısı

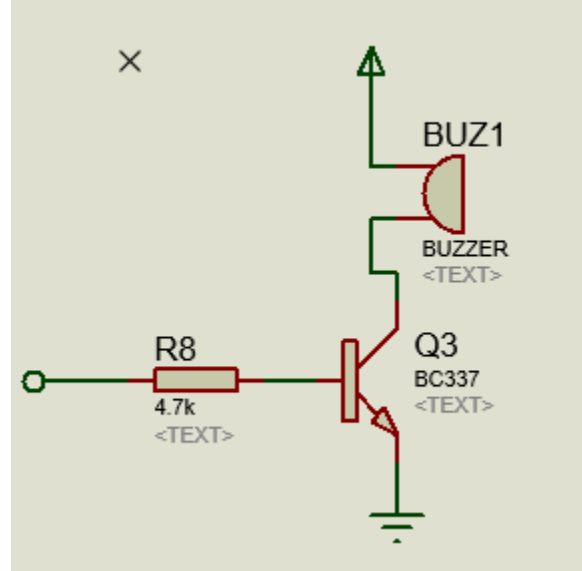
LCD panelleri robot projelerinde ya da otomasyon projelerinde kullanmak için bilgisayarınızın seri ya da paralel portundan veya bir PIC mikrodenetleyici kullanarak kontrol edilebilen ekranlardır. Üretilen LCD panellerin çoğunda tek sıra halinde 16 pin bulunur. Bu pinlerden ilk 14 tanesi kontrol için son iki tanesi ise eğer varsa arka ışık için kullanılır.



Şekil 2.8. LCD Bağlantısı

2.4.5. Buzzer bağlantısı

Buzzer verilen voltaja göre farklı ses sinyalleri sağlayan bir cihazdır. Maliyetleri az, üretimi basit, ve çok hafif yapıda olmalarından dolayı kullanım alanı çok geniştir. Hırsız alarmları, araçlarda uyarı veren çoğu sistem, bazı zil sesleri kısaca uyarı almak, korunmak, ayırım yapmak amaçlı her yerde kullanılabilir. Devrede kutu dolduğunu belirtmek amaçlı kesik kesik sesler çıkarması üzerine kodu yazılmıştır. Bir transistör yardımı ile sürülebilir.



Şekil 2.9. Buzzer Devresi

```
for(i = 0; i < 20; i++) //kaç kez bipleyeceğini belirliyor
{
    output_high(pin_c0);
    delay_ms(100);      //biplerin uzunluğunu belli ediyor değer
                        //azaldıkça tek ses elde edebiliriz.
    output_low(pin_c0);
    delay_ms(100);
}
```

Şekil 2.10. Buzzer Kodu

3. ARAŞTIRMA BULGULARI VE TARTIŞMA

3.1. Sensör Tasarımı

Sensörler gerçek hayatta gerçekleşen bir olayı hava, basınç ses, sıcaklık, ışık ve renk v.b. bilgisayar ortamına aktarmada ve bu verilerle işlemleri yapmamızı kolaylaştıran ürünlerdir. Projede gerçekleştirilen sensör tasarımında ihtiyaç olan durum 4 farklı rengi algılayabilmesidir. Bunu da önceki bölümlerde anlatılan renklerin LDR veya LDR ye seri bağlanan direnç üzerine düşen gerilim değeri sayesinde analog bir değer elde etmeye çalışılmıştır. Bu değeri dijital ortamda kullanabilmek için bir ADC(Analog digital converter) kullanılmıştır.

ADC birimlerinin en önemli özellikleri çözünürlükleri yani bit sayılarıdır. 10 bitlik bir ADC 0 ile 1023 arasında değerler üretebilmektedir. ADC biriminin bit sayısı arttıkça çözünürlüğü de artmaktadır. Devreye verilen gerilim PIC 5Volt(V) ile çalıştığı için 5V, ADC ye verilecek değerleri direnç üzerinden hesaplandığında direncin voltaj değeri 0-5V arasında olmuştur. 10 bitlik bir ADC kullanıldığı için ise;

$$5V/1023=0.0048875855327468 \quad (3.1.)$$

gibi bir ADC hassasiyeti değeri elde edilmiştir. Yani ADC birimi 4.8mV değerindeki voltaj değişimlerini hissedebilmektedir. Eğer ADC biriminden okunan değer 512 ise ADC biriminin girişinde bulunan voltaj değeri;

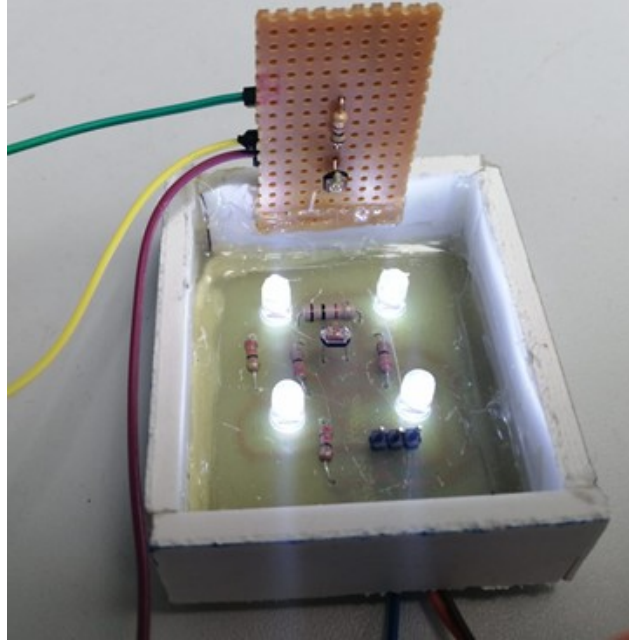
$$512*0.0048875855327468 \approx 2.5V \quad (3.2.)$$

olur. Bu durumda direncin üzerine düşen gerilim değeri bulunur.

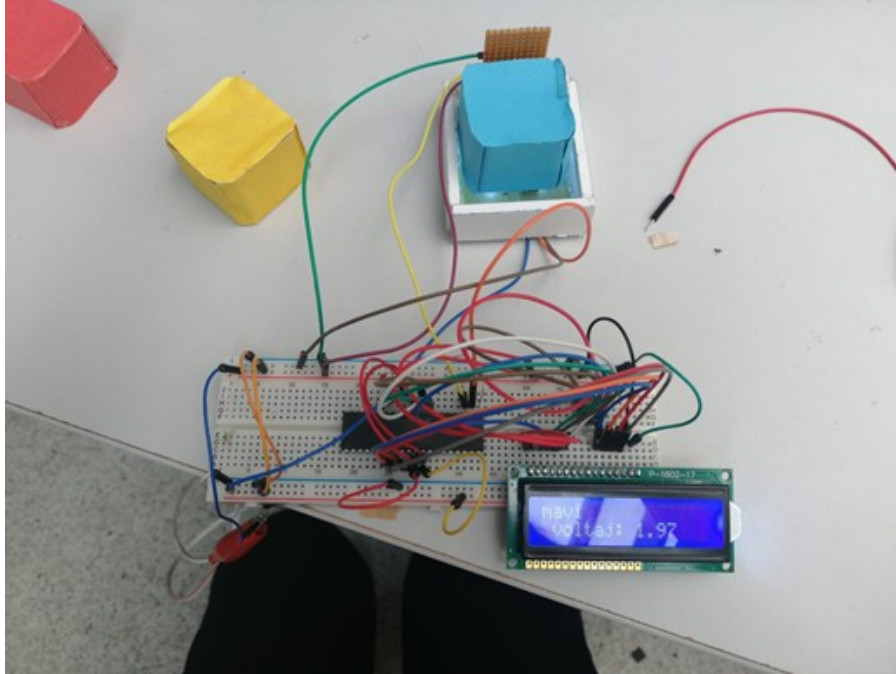
Tespit edilmesi gereken renkler ile deneme yapılarak her rengin voltaj aralığı tespit edildi. Kod yazarken bu aralıklar kullanılarak motorların çalışması düzenlendi.

Sensörü tasarlarken renklerin algılanması için devrede ışık olması amacıyla 4 adet led kullanıldı. LDR ye seri olarak bağlanan direnç üzerindeki voltaj değerleri izlendi.

Sensörde istenilen 4 renk dışında denenen renkli cisimleri de “ farklı bir cisim” ve “ cisim yok” şeklinde ayırt edebilmek adına bir LDR ve dirençten oluşan sadece cismi algılamak amaçlı bir sensör daha yapıldı.



Şekil 3.3. Renk ve cisim algılama sensörü devresi



Şekil 3.4. Renk ve cisim algılama sensörü devresi

3.1.2. Deney İle Veri Elde Etme

Projede istenen 4 farklı rengi algılamak için bu renklerde cisimler yapıldı. Her renkten alınacak verilerin hassasiyetini arttırmak amacıyla karanlık ve aydınlık olmak üzere iki farklı ortamda veriler alındı.

Çizelge 3.1. Renklerin Voltaj değerleri

	IŞIKLI ORTAM	KARANLIK ORTAM
KIRMIZI	1. 86V, 1.92 V	1.91V, 1.93V
MAVİ	1.95V, 2.08V	1.97V, 2.10V
SARI	2.35V, 2.47V	2.32V, 2.45V
SİYAH	1.51V, 1.52V	1.49V, 1.51V
CİSİM YOKKEN	2.02V, 2.03V	1.90V, 1.91V

Bu verilere göre

Cisim algılama: $(0.3 < \text{voltaj}1) \ \&\& \ (\text{voltaj}1 < 2.30)$ (cisim algılama sensöründeki veriler kullanıldı)

Kırmızı: $(1.80 < \text{voltaj}) \ \&\& \ (\text{voltaj} < 1.94)$

Mavi: $(1.95 < \text{voltaj}) \ \&\& \ (\text{voltaj} < 2.15)$

Sarı: $(2.30 < \text{voltaj}) \ \&\& \ (\text{voltaj} < 2.55)$

Siyah: $(1.4 < \text{voltaj}) \ \&\& \ (\text{voltaj} < 1.60)$

Bilinmeyen cisim: $(1.45 > \text{voltaj}) \ || \ (\text{voltaj} > 2.55)$ değerleri oluşturuldu.



Şekil 3.5. Renk ve cisim algılama sensörü renk ölçümü

3.1.3. Sensör çalıştırma programı

Sensör kısmında kullanılan kodlar;

```
#define LCD_ENABLE_PIN PIN_E2
#define LCD_RS_PIN PIN_E0
#define LCD_RW_PIN PIN_E1
#define LCD_DATA4 PIN_D4
#define LCD_DATA5 PIN_D5
#define LCD_DATA6 PIN_D6
#define LCD_DATA7 PIN_D7

#include <lcd.c>

void main()
{
    set_tris_a(0b11111111);
    set_tris_b(0b00000000);
    set_tris_c(0b00000000);
    int16 dijitalbilgi,dijitalbilgi1;
    float voltaj,voltaj1;
    setup_adc(ADC_CLOCK_DIV_32);
    SETUP_ADC_PORTS(sAN0,sAN1); //sAN0
    =RENK ALGILAYAN SENSOR sAN1= CİSİM
    ALIGLAYAN SENSÖR

    lcd_init();

    while(TRUE)
    {
        set_adc_channel(1); //sAN1= CİSİM
        ALIGLAYAN SENSÖR

        delay_us(20);
        dijitalbilgi1=read_adc();
```

```
        delay_ms(100);
        voltaj1=dijitalbilgi1*0.0048875855;
        delay_ms(100);
        if((0.3<voltaj1) && (voltaj1<2.30))
        {
            delay_ms(100);
            printf(LCD_PUTC,"\fCisim Algilandi");
            printf(lcd_putc"\nvoltaj: %f",voltaj);
            delay_ms(50);

            set_adc_channel(0); //sAN0 =RENK
            ALGILAYAN SENSOR

            delay_us(20);
            dijitalbilgi=read_adc();
            delay_ms(100);
            voltaj=dijitalbilgi*0.0048875855;
            delay_ms(1000);
            if((1.80<voltaj) && (voltaj<1.94))
            {
                lcd_putc("\fKup:kirmizi");
                printf(lcd_putc"\nvoltaj: %f",voltaj);
                delay_ms(500);
            }
            delay_ms(100);

            if((1.95<voltaj) && (voltaj<2.15))
            {
                lcd_putc("\fKup:mavi");
                printf(lcd_putc"\nvoltaj: %f",voltaj);
```



```

    delay_ms(500);
}

delay_ms(100);

if((2.30<voltaj)&& (voltaj<2.55))
{

    lcd_putc("\fKup:sari");
    printf(lcd_putc"\nvoltaj: %f",voltaj);
    delay_ms(500);
}

delay_ms(100);

if((1.4<voltaj)&& (voltaj<1.60))
{

    lcd_putc("\fKup:siyah");
    printf(lcd_putc"\nvoltaj: %f",voltaj);
    delay_ms(500);
}

delay_ms(100);

if((1.45>voltaj) || (voltaj>2.55))
{

    lcd_putc("\fBilinmeyen Cisim");
    printf(lcd_putc"\nvoltaj: %f",voltaj);
    delay_ms(1000);
}

else
{
    lcd_putc("\fCisim Yok");
    printf(lcd_putc"\nvoltaj1: %f",voltaj1);
    delay_ms(100);
}

}
}

```

3.2. Servo Motor Çalıştırma Deneyi

Önceki başlıklarda anlatıldığı gibi servo motor çalıştırmak için belirli mili saniyelerde bir motorun sinyal ucuna güç verip kesmek gereklidir. Servo motorlar uzun süre konumlarını koruyamadıkları için döngü kullanarak konumunu koruması sağlanır. Bu sayede motorların hareketi gerçekleştirilir. Deneme için aşağıdaki kod kullanılmıştır.

```
for( i=0;i<50;i++)    // 0 dereceye dönme
{
    output_high(pin_b1);
    delay_us(1000);
    output_low(pin_b1);
    delay_us(19000);
}
delay_ms(1000);

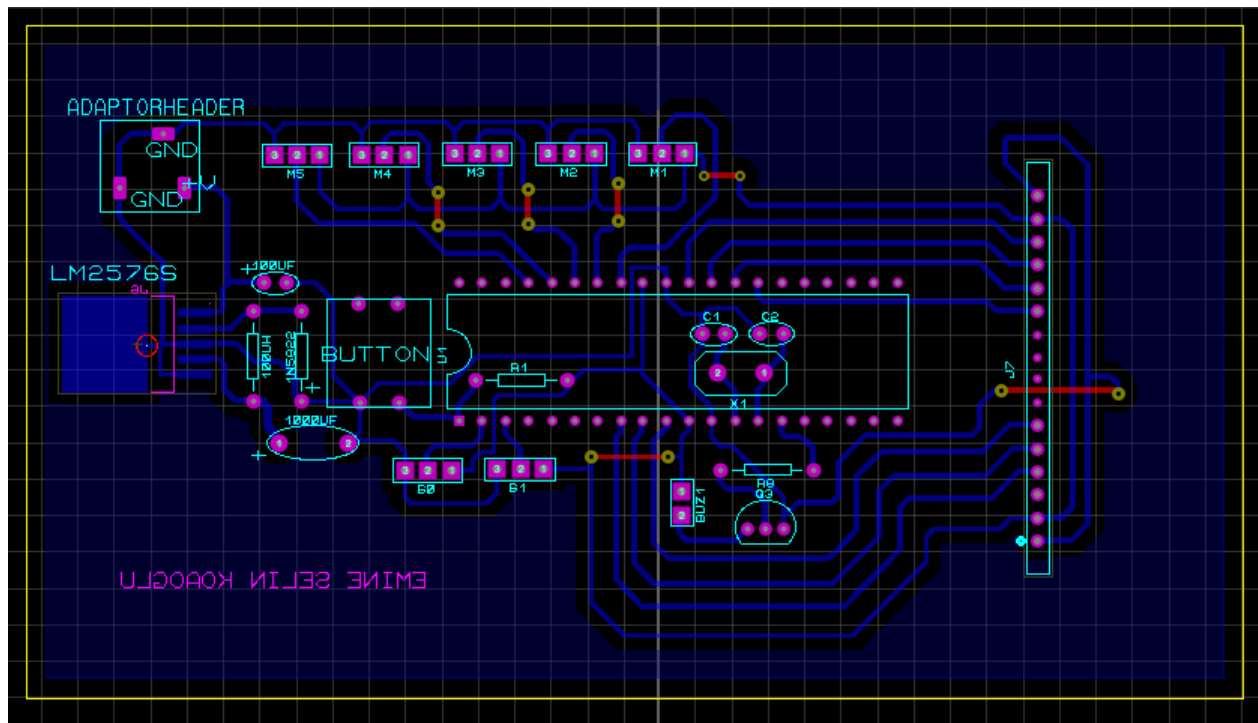
for( i=0;i<30;i++)    // 90 dereceye dönme
{
    output_high(pin_b3);
    delay_us(1500);
    output_low(pin_b3);
    delay_us(18500);
}
delay_ms(1000);

for( i=0;i<50;i++)    // 180 dereceye dönme
{
    output_high(pin_b2);
    delay_us(2000);
    output_low(pin_b2);
    delay_us(18000);
}
```

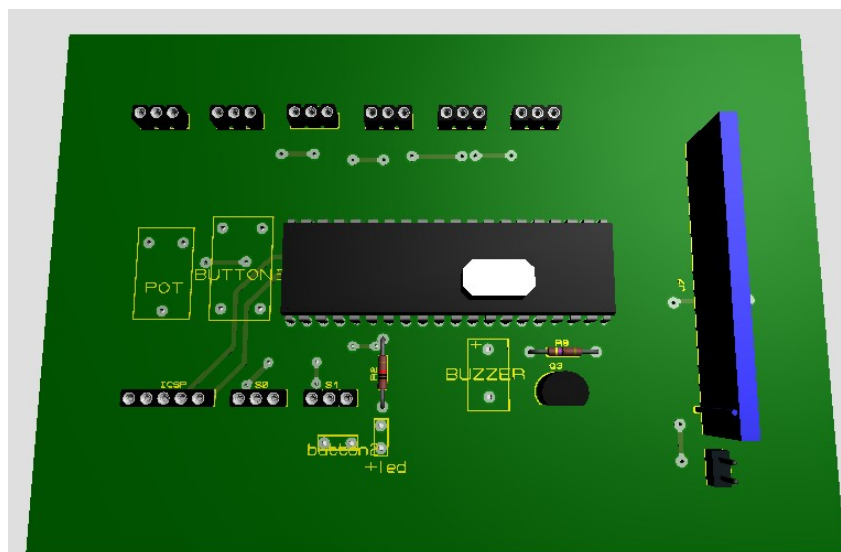
Robot koldaki motorları istenilen yönlerde kontrol etmek için delay_(); komutundaki mikrosaniye değerleri değiştirildi.

3.3. Devre Tasarımı

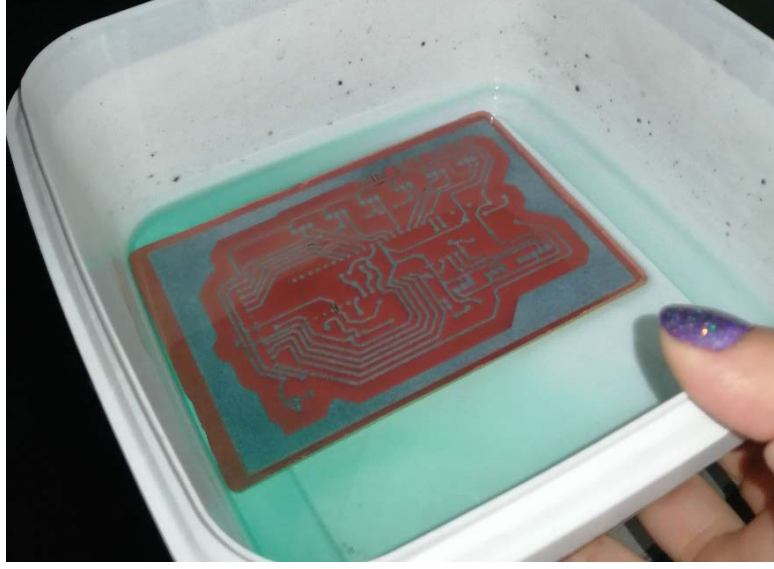
Devreyi tasarlarken 2 adet sensör giriş pinlerine ek olarak bir adaet butuon girişi kullanılma ihtimali için eklendi. 4 motor kullanılmasına rağmen 6 adet servo girişi, ICSP pinleri ve buzzer pini eklendi. Bu şekilde istenildiği taktirde fazla motor eklenme ve buton girişi ile kontrol etme imkanı sağlandı.



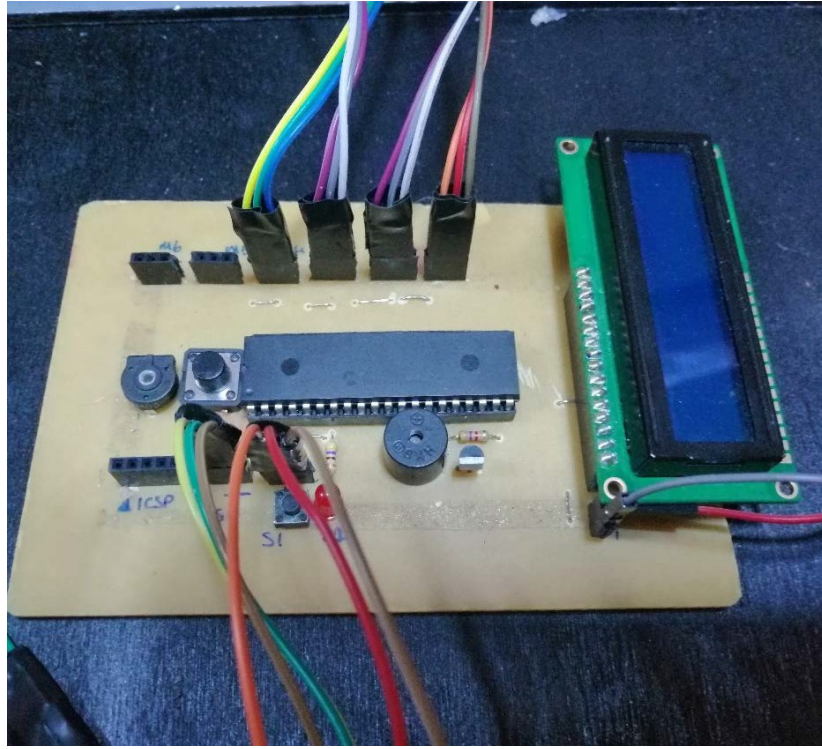
Şekil 3.6. Ana devre ARES çizimi



Şekil 3.7. Ana devre 3D görüntüsü



Şekil 3.8. Devre basım aşaması görüntüsü



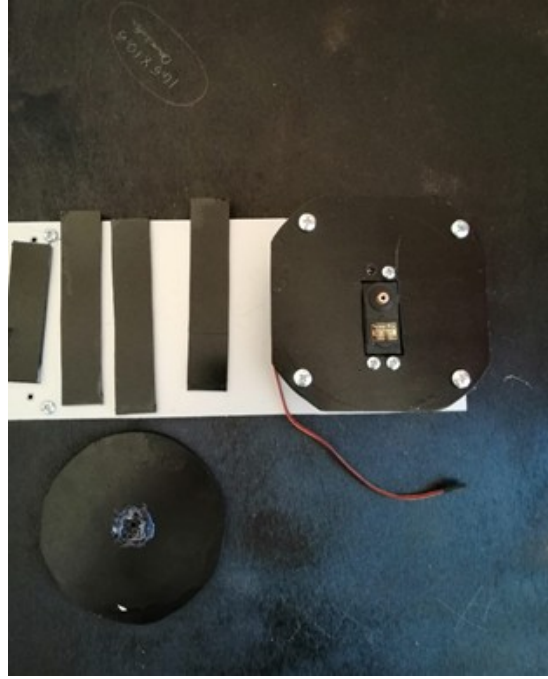
Şekil 3.9. Ana Devre

3.4. Robot Kol Tasarımı

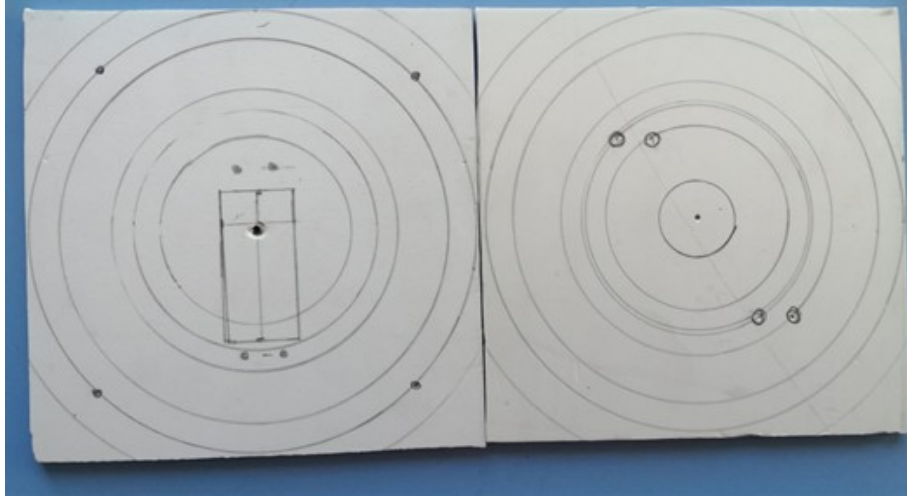
Projede, robot kolun mekanik tasarımında, servo motorları birleştirici olarak ve robot kol şeklinin alınması amacıyla dekota malzeme kullanılarak şekil verilmiştir.

PVC hammaddesinin içine kimyasallar homojen bir biçimde karıştırılarak çekme veya dökme hattında üretilen levhalardır. Bu işlemin yapılmasında ki amaç malzemenin yoğunluğunu düşürerek birim başına kullanılan PVC miktarını azaltmak, böylelikle kaynakları daha doğru kullanmaktır. Hafif bir malzemedir. Dolayısıyla işleme, taşıma ve istifleme rahatlığı sunmaktadır. Malzemenin özgül ağırlığı düşürüldüğü için de sert pvc ye göre çok daha ekonomiktir. Uzun ömürlüdür, taşınması, işlenmesi kolaydır. Yüzeyleri düzgündür. Darbeye dayanıklı malzemelerdir.

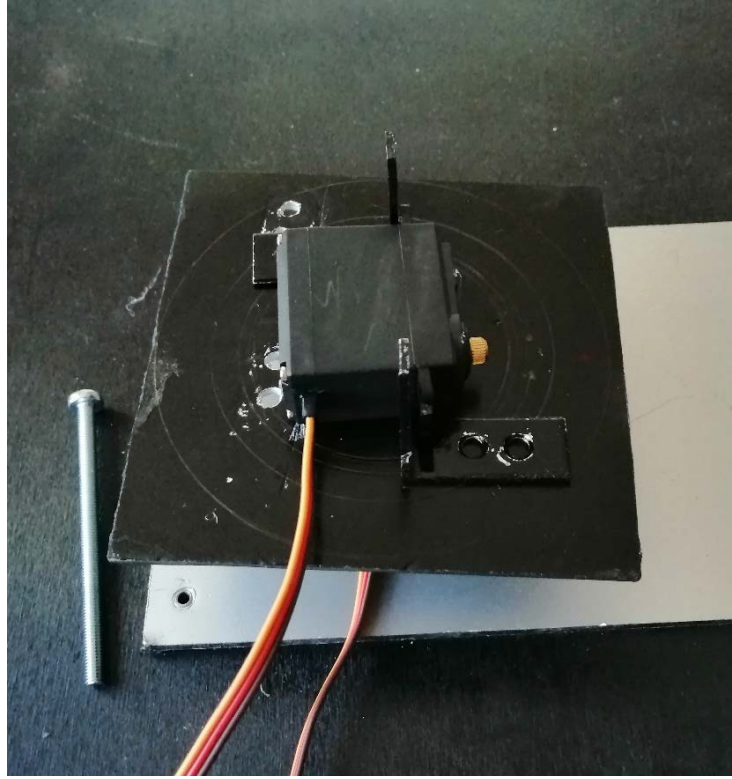
Robot kolun bulunduğu tabanı ince tahtadan yapıldı. Ve estetik açıdan uyumlu olması için parçalar siyah rengine boyandı.



Şekil 3.10. Robot kol parçaları



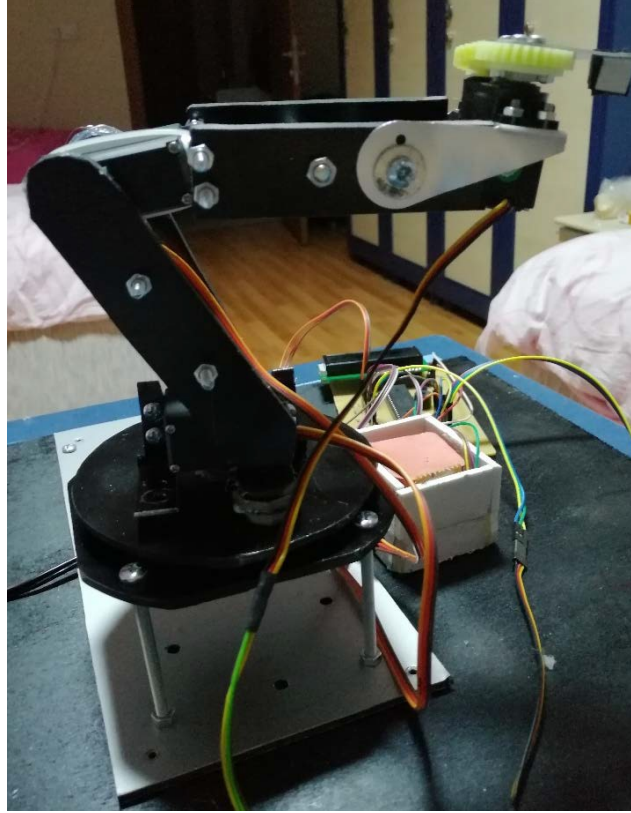
Şekil 3.11. Robot kol tabanının ölçülmüş parçaları



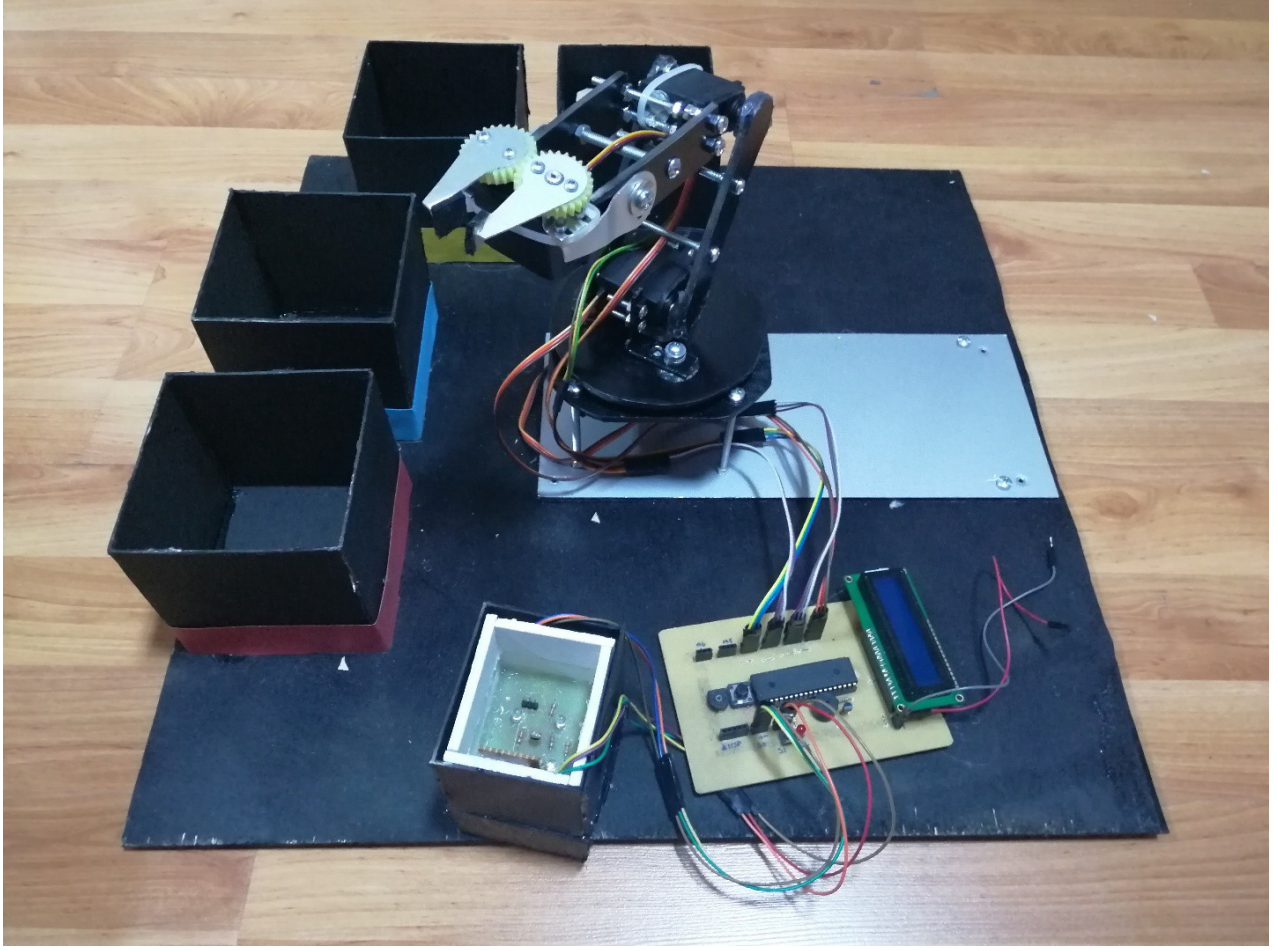
Şekil 3.12. Robot kol taban kısmı



Şekil 3.13. Robot kolun kol kısmı



Şekil 3.14. Robot kol tam hali



Şekil 3.15. Proje son hali

4. SONUÇLAR VE ÖNERİLER

Yapılan proje günümüzün gerekliliklerini karşılamak amacıyla insan kas gücünü azaltmak ve yerine robot kolların birçok işi gerçekleştirebilmesini sağlamaktır. Renk sensörü kullanarak büyük ve tek bu iş için kullanılan makinalardan kaçınmak ve maliyeti düşürmek hedeflenmiştir ve bu amaçla prototipi hazırlanmıştır. Projenin kontrol ve stabilizasyon yöntemleri üzerinde daha fazla çalışma yapıp geliştirilmesi sağlanabilmektedir. Sensörün dijital değerlerini çözünürlüğü artırarak daha fazla renk algılama aralığı oluşturulabilmektedir.

KAYNAKLAR

Serdar Çiçek, CCS C ile PIC Programlama, 2016, kitap

Özcan Çataltaş, Ö.Ç. Selçuk Üniversitesi, SÜ, Mikrodenetleyici Lab. Föyü, 2018

Editronikx, control de servomotor con pic, youtube,
<https://www.youtube.com/watch?v=5UgMIL9mRxU>, [ağustos,2019]

Robotiksisitem,MikrodenetleyiciNedir?, Google,
http://www.robotiksisitem.com/mikrodenetleyici_nedir_pic_ozellikleri.html , [ağustos,2019]

Tayfun Yağız Semiz, 11 Kasım 2015, Servo Motor Nedir? Çeşitleri ve Çalışma Prensipleri,
<https://maker.robotistan.com/rc-servo-motor-nedir/> [kasım, 2019]

EK-1 Proje İçin Yazılan Program

```

#define LCD_ENABLE_PIN PIN_E2

#define LCD_RS_PIN PIN_E0

#define LCD_RW_PIN PIN_E1

#define LCD_DATA4 PIN_D4

#define LCD_DATA5 PIN_D5

#define LCD_DATA6 PIN_D6

#define LCD_DATA7 PIN_D7


#include <lcd.c>


//B0,B1,B2,B3,B4,B5 = SERVO ÇIKIŞ PINLERİ
//A0,A1= SENSÖR GİRİŞLERİ
//CO= BUZZER ÇIKIŞ

unsigned int i; //for döngüleri için


int topmavikupsayisi =0; // kutulara giren kup
sayısını belirlemek için

int topsarikupsayisi =0;

int topkirmizikupsayisi =0;

int topsiyahkupsayisi =0;


void mavi(); //renklere göre yapılacak hareketleri
ayırmak için açılan ...

void sari();

void kirmizi();

void siyah();

void basangic();

void main()

{

    set_tris_a(0b11111111);

    set_tris_b(0b00000000);

    set_tris_c(0b00000000);

    int16 dijitalbilgi,dijitalbilgi1;

    float voltaj,voltaj1;

    setup_adc(ADC_CLOCK_DIV_32);

    SETUP_ADC_PORTS(sAN0,sAN1); //sAN0
=RENK ALGILAYAN SENSÖR sAN1= CİSİM
ALIGLAYAN SENSÖR

    lcd_init();

    baslangic();


    while(TRUE)

    {

        set_adc_channel(1); //sAN1= CİSİM
ALIGLAYAN SENSÖR

        delay_us(20);

        dijitalbilgi1=read_adc();

        delay_ms(100);

        voltaj1=dijitalbilgi1*0.0048875855;

        delay_ms(100);

        if((0.3<voltaj1) && (voltaj1<2.30))

        {

            delay_ms(100);

            printf(LCD_PUTC,"\fCisim Algilandi");

            printf(lcd_putc"\nvoltaj: %f",voltaj);

            delay_ms(50);

```

```

    set_adc_channel(0); //sAN0 =RENK
ALGILAYAN SENSOR

```

```

    delay_us(20);
    dijitalbilgi=read_adc();
    delay_ms(100);
    voltaj=dijitalbilgi*0.0048875855;
    delay_ms(1000);

    if((1.80<voltaj) && (voltaj<1.94))
    {

        lcd_putc("\fKup:kirmizi");
        printf(lcd_putc"\nvoltaj: %f",voltaj);
        delay_ms(500);

        kirmizi();
        baslangic();

    }

    delay_ms(100);

    if((1.95<voltaj) && (voltaj<2.15))
    {

        lcd_putc("\fKup:mavi");
        printf(lcd_putc"\nvoltaj: %f",voltaj);
        delay_ms(500);

        mavi();
        baslangic();

```

```

    }

    delay_ms(100);

    if((2.30<voltaj)&& (voltaj<2.55))
    {

        lcd_putc("\fKup:sari");
        printf(lcd_putc"\nvoltaj: %f",voltaj);
        delay_ms(500);

        sari();
        baslangic();
    }

    delay_ms(100);

    if((1.4<voltaj)&& (voltaj<1.60))
    {

        lcd_putc("\fKup:siyah");
        printf(lcd_putc"\nvoltaj: %f",voltaj);
        delay_ms(500);

        siyah();
        baslangic();

    }

    delay_ms(100);

    if((1.45>voltaj) || (voltaj>2.55))

```

```

{
    delay_us(19100);
}

    lcd_putc("\fBilinmeyen Cisim");
    printf(lcd_putc"\nvoltaj: %f",voltaj);
    delay_ms(1000);

}

}

else
{
    lcd_putc("\fCisim Yok");
    printf(lcd_putc"\nvoltaj1: %f",voltaj1);
    delay_ms(100);
}

}

}

void mavi()
{
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(900);
        output_low(pin_b3);

        delay_us(19100);
        delay_ms(1000);
        for( i=0;i<50;i++)
        {
            output_high(pin_b2);
            delay_us(800);
            output_low(pin_b2);
            delay_us(19200);
        }
        delay_ms(1000);
        for( i=0;i<50;i++)
        {
            output_high(pin_b1);
            delay_us(1700);
            output_low(pin_b1);
            delay_us(18300);
        }
        delay_ms(50);
        for( i=0;i<50;i++)
        {
            output_high(pin_b1);
            delay_us(1800);
            output_low(pin_b1);
            delay_us(18200);
        }
        delay_ms(50);
        for( i=0;i<50;i++)

```

```

{
    output_high(pin_b1);
    delay_us(1900);
    output_low(pin_b1);
    delay_us(18100);
}

delay_ms(1000);

for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(500);
    output_low(pin_b3);
    delay_us(19500);
}

delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(700);
        output_low(pin_b2);
        delay_us(19300);
    }

delay_ms(1000);
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(500);
        output_low(pin_b3);
        delay_us(19500);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1500);
        output_low(pin_b1);
        delay_us(18500);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b0);
        delay_us(1500);
        output_low(pin_b0);
        delay_us(18500);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(700);
        output_low(pin_b2);
        delay_us(19300);
    }
}

```

```

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1800);
    output_low(pin_b1);
    delay_us(18200);
}

delay_ms(1000);

for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(1000);
    output_low(pin_b3);
    delay_us(19000);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(500);
    output_low(pin_b2);
    delay_us(19500);
}

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1500);
    output_low(pin_b1);
    delay_us(18500);
}

delay_ms(1000);
topmavikupsayisi=topmavikupsayisi+1;

printf(lcd_putc"\fmavikupadeti:%u",topmavikup
sayisi);

delay_ms(2000);
if(topmavikupsayisi>=2)
{
    lcd_putc("\fmavi kutu doldu");
    printf(lcd_putc"\nLutfen bosaltin! ");

    for(i = 0; i < 20; i++) //kaç kez
bipleyeceğini belirliyor
    {
        output_high(pin_c0);

        delay_ms(100); //biplerin uzunluğunu
belli ediyor değer

        //azaldıkca tek ses elde
edebiliriz.

        output_low(pin_c0);
        delay_ms(100);
    }

    topmavikupsayisi =0;

```

```

        delay_ms(2000);

    }

}

void sari()
{
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(900);
        output_low(pin_b3);
        delay_us(19100);
    }

    delay_ms(1000);

    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(800);
        output_low(pin_b2);
        delay_us(19200);
    }
}

```

```

delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1700);
        output_low(pin_b1);
        delay_us(18300);
    }
    delay_ms(50);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1800);
        output_low(pin_b1);
        delay_us(18200);
    }

    delay_ms(50);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1900);
        output_low(pin_b1);
        delay_us(18100);
    }

    delay_ms(1000);

    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(500);
    }
}

```



```

    output_low(pin_b3);
    delay_us(19500);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(700);
    output_low(pin_b2);
    delay_us(19300);
}

delay_ms(1000);
for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(500);
    output_low(pin_b3);
    delay_us(19500);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1500);
    output_low(pin_b1);
    delay_us(18500);
}

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b0);
    delay_us(1200);
    output_low(pin_b0);
    delay_us(18300);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(700);
    output_low(pin_b2);
    delay_us(19300);
}

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1800);
    output_low(pin_b1);
    delay_us(18200);
}

delay_ms(1000);

for( i=0;i<30;i++)
{

```

```

    output_high(pin_b3);
    delay_us(1000);
    output_low(pin_b3);
    delay_us(19000);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(500);
    output_low(pin_b2);
    delay_us(19500);
}

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1500);
    output_low(pin_b1);
    delay_us(18500);
}

delay_ms(1000);
topsarikupsayisi=topsarikupsayisi+1;

printf(lcd_putc"\fsarikupadeti:%u",topsarikupsa
yisi);

delay_ms(2000);

    if(topsarikupsayisi==2)
    {

        lcd_putc("\fsari kutu doldu");
        printf(lcd_putc"\nLutfen bosaltin! ");

        for(i = 0; i < 20; i++) //kaç kez
bipleyeceğini belirliyor

        {

            output_high(pin_c0);

            delay_ms(100); //biplerin uzunluğunu
belli ediyor değer azaldıkca tek ses elde edebiliriz.

            output_low(pin_c0);
            delay_ms(100);

        }

        topsarikupsayisi =0;
        delay_ms(2000);

    }

}

void kirmizi()
{
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(900);
        output_low(pin_b3);
        delay_us(19100);
    }
}

```

```

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(800);
    output_low(pin_b2);
    delay_us(19200);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1700);
    output_low(pin_b1);
    delay_us(18300);
}

delay_ms(50);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1800);
    output_low(pin_b1);
    delay_us(18200);
}

delay_ms(50);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1900);
    output_low(pin_b1);
    delay_us(18100);
}

delay_ms(1000);

for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(500);
    output_low(pin_b3);
    delay_us(19500);
}

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(700);
    output_low(pin_b2);
    delay_us(19300);
}

delay_ms(1000);
for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(500);
    output_low(pin_b3);
    delay_us(19500);
}

```

```

    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1500);
        output_low(pin_b1);
        delay_us(18500);
    }

    delay_ms(1000);

    for( i=0;i<50;i++)
    {
        output_high(pin_b0);
        delay_us(1900);
        output_low(pin_b0);
        delay_us(18100);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(700);
        output_low(pin_b2);
        delay_us(19300);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1800);
        output_low(pin_b1);
        delay_us(18200);
    }

    delay_ms(1000);

    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(1000);
        output_low(pin_b3);
        delay_us(19000);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(500);
        output_low(pin_b2);
        delay_us(19500);
    }

    delay_ms(1000);

    for( i=0;i<50;i++)
    {
        output_high(pin_b1);

```

```

    delay_us(1500);
    output_low(pin_b1);
    delay_us(18500);
}

delay_ms(1000);

topkirmizikupsayisi=topkirmizikupsayisi+1;

printf(lcd_putc("\fkirm.kupadeti:%u",topkirmizikupsayisi);

    delay_ms(2000);
    if(topkirmizikupsayisi>=2)

    {

        lcd_putc("\fkirmizikutudoldu");
        printf(lcd_putc("\nLutfen bosaltin! ");

        for(i = 0; i < 20; i++) //kaç kez
bipleyeceğini belirliyor
        {

            output_high(pin_c0);

            delay_ms(100); //biplerin uzunluğunu
belli ediyor değer azaldıkca tek ses elde edebiliriz.

            output_low(pin_c0);
            delay_ms(100);

        }

        topkirmizikupsayisi =0;

delay_ms(2000);
    }

void siyah()
{
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(900);
        output_low(pin_b3);
        delay_us(19100);
    }

    delay_ms(1000);

    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(800);
        output_low(pin_b2);
        delay_us(19200);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1700);
        output_low(pin_b1);
        delay_us(18300);
    }
}

```

```

}
delay_ms(50);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1800);
    output_low(pin_b1);
    delay_us(18200);
}

delay_ms(50);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1900);
    output_low(pin_b1);
    delay_us(18100);
}

delay_ms(1000);

for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(500);
    output_low(pin_b3);
    delay_us(19500);
}

delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1500);
        output_low(pin_b1);
        delay_us(18500);
    }

delay_ms(1000);

for( i=0;i<50;i++)
{
    output_high(pin_b0);
    delay_ms(1000);
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(500);
        output_low(pin_b3);
        delay_us(19500);
    }

    delay_ms(1000);
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(500);
        output_low(pin_b3);
        delay_us(19500);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(700);
        output_low(pin_b2);
        delay_us(19300);
    }

    delay_ms(1000);
    for( i=0;i<30;i++)
    {
        output_high(pin_b3);
        delay_us(500);
        output_low(pin_b3);
        delay_us(19500);
    }

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b0);

```

```

    delay_us(900);
    output_low(pin_b0);
    delay_us(19100);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b2);
    delay_us(700);
    output_low(pin_b2);
    delay_us(19300);
}

delay_ms(1000);
for( i=0;i<50;i++)
{
    output_high(pin_b1);
    delay_us(1800);
    output_low(pin_b1);
    delay_us(18200);
}

delay_ms(1000);

for( i=0;i<30;i++)
{
    output_high(pin_b3);
    delay_us(1000);
    output_low(pin_b3);
    delay_us(19000);
}

    delay_ms(1000);
    for( i=0;i<50;i++)
    {
        output_high(pin_b2);
        delay_us(500);
        output_low(pin_b2);
        delay_us(19500);
    }

    delay_ms(1000);

    for( i=0;i<50;i++)
    {
        output_high(pin_b1);
        delay_us(1500);
        output_low(pin_b1);
        delay_us(18500);
    }

    delay_ms(1000);

    topsiyahkupsayisi=topsiyahkupsayisi+1;

    printf(lcd_putc"\fSiyahkupadeti:%u",topsiyahku
psayisi);

    delay_ms(2000);

    if(topsiyahkupsayisi==2)
    {

        lcd_putc("\fsiyah kutu doldu");

```

```

    printf(lcd_putc"\nLutfen bosaltin! ");

    for(i = 0; i < 20; i++) //kaç kez
bipleyeceğini belirliyor
    {

        output_high(pin_c0);

        delay_ms(100); //biplerin uzunluğunu
belli ediyor değer azaldıkca tek ses elde edebiliriz.

        output_low(pin_c0);

        delay_ms(100);

    }

    topsiyahkupsayisi =0;

    delay_ms(2000);

}

void baslangic()
{

    lcd_putc("\fbaslangic Konumu");

    for( i=0;i<50;i++) //b0 1.

    {

        output_high(pin_b0);

        delay_us(2600);

        output_low(pin_b0);

        delay_us(17400);

    }

```

```

delay_ms(1000);

for( i=0;i<50;i++) // b1 1.

{

    output_high(pin_b1);

    delay_us(1500);

    output_low(pin_b1);

    delay_us(18500);

}

delay_ms(1000);

for( i=0;i<50;i++) //b2 1.

{

    output_high(pin_b2);

    delay_us(500);

    output_low(pin_b2);

    delay_us(19500);

}

delay_ms(1000);

for( i=0;i<30;i++) //b3 2.

{

    output_high(pin_b3);

    delay_us(500);

    output_low(pin_b3);

    delay_us(19500);

}

delay_ms(1000);

}

```