



T. C.

İSTANBUL GELİŞİM ÜNİVERSİTESİ

MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

**ÇOCUKLARA TEMEL ARAÇ EYLEMLERİNİ ÖĞRETEBİLMEK
AMACIYLA TASARLANMIŞ OTONOM VEYA SES KONTROLÜ İLE
HAREKET EDEN SES VE IŞIK ÇIKIŞLI ARAÇ**

Bitirme Çalışması 2

Öğrencinin Adı SOYADI

Suat KEKLİK

Danışman

Dr. Öğr. Üyesi Ercan AYKUT

İstanbul, 2024

TEZ TANITIM FORMU

Yazar Adı Soyadı	Suat KEKLİK
Tezin Dili	Türkçe
Tezin Adı	Çocuklara Temel Araç Eylemlerini Öğretebilmek Amacıyla Tasarlanmış Otonom Veya Ses Kontrolü İle Hareket Eden Ses Ve Işık Çıkışlı Araç
Fakülte	İstanbul Gelişim Üniversitesi Mühendislik ve Mimarlık Fakültesi
Bölüm	Elektrik-Elektronik Mühendisliği
Tezin Türü	Bitirme Çalışması 2
Tezin Tarihi	28.05.2024
Sayfa Sayısı	55 (Elli Beş Sayfa)
Tez Danışmanı	Dr. Öğr. Üyesi Ercan AYKUT
Dağıtım Listesi	Mühendislik ve Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği Bölümü

Suat KEKLİK

BEYAN

Bu tezin hazırlanmasında bilimsel ahlak kurallarına uyulduđu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduđu, kullanılan verilerde herhangi tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez olarak sunulmadığını beyan ederim.

Suat KEKLİK

26.05.2024

İSTANBUL GELİŞİM ÜNİVERSİTESİ
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

.....'ın
.....
..... adlı tez çalışması, jürimiz tarafından
Elektrik-Elektronik Mühendisliği Bölümü Bitirme tezi olarak kabul edilmiştir.

İmza

Başkan

Prof. Dr. Adı SOYADI

İmza

Üye

Prof. Dr. Adı SOYADI

(Danışman)

İmza

Üye

Prof. Dr. Adı SOYADI

İmza

Üye

Doç. Dr. Adı SOYADI

İmza

Üye

Dr. Adı SOYADI

(Var ise İkinci Danışman)

ÖZET

Ülkemizde ki trafik kazalarının gün geçtikçe artması ve neticesinde çok değerli olan insan hayatının olumsuz etkilenmesi söz konusudur. TÜİK verilerine göre ülkemizdeki kara motorlu taşıtları 26 milyon sayısını geçmiş, aynı kurum tarafından yayımlanmış verilere göre trafik kaza sayısı 23.571 ulaşmıştır. Bu kazalar 6.548 insan hayatının kaybedilmesine neden olmuştur. Bu kişilerin ise %10,1 oranı 0,-17 yaş aralığındaki kişilerden oluşmaktadır.

Yapılan çalışmalarda çocukluk çağında yapılan eğitimin sonraki eğitim aşamalarında olumlu dönüşlere neden olduğu gözlemlenmiştir. Bu nedenle birçok araştırmacı özellikle trafik kuralları ve trafik güvenliği eğitimlerinin çocukluk çağından itibaren başlanmasının uygun olduğunu belirtmiştir. Böylece ileri yaşlardaki eğitimlerin daha başarılı olacağını belirtmişlerdir. Bu projenin amacı trafikte temel basit araç davranışlarını öğretmektir. Bu amaç üzerine araştırma yapılarak; sağ dönmek, sola dönmek, geri gelmek, ileri gitmek ve durmak durumları incelenmiştir. Bu eylemlere ses ve ışık çıkışları eklenerek algıyı artırmak hedeflenmiştir. Ayrıca araç seyir halindeyken engel ile karşılaşması, ışık miktarına göre farların kapatılıp açılması incelemeye dahil edilmiştir. Bu başlıklar ile ilgili şoför personelden oluşan bir grup ile anket çalışması yapılması planlanmaktadır. Sonuçlar listelendikten sonra elemine edilecektir. Böylece çocuklara araçları yönlendirme komutlarını öğretmek ve hareketlerin anlaşılması amaçlanmıştır. Kullanım kolaylığı açısından cep telefonu üzerinden kontrol edilmesi düşünülmüştür.

Anahtar Kelimeler: Çocuk, Trafik, Araç Eylem

ABSTRACT

Traffic accidents in our country are increasing day by day and as a result, very valuable human life is negatively affected. According to TÜİK data, the number of land motor vehicles in our country has exceeded 26 million, and according to the data published by the same institution, the number of traffic accidents has reached 23,571. These accidents caused the loss of 6,548 human lives. 10.1% of these people are between the ages of 0-17.

Studies have shown that education in childhood causes positive outcomes in later educational stages. For this reason, many researchers have stated that it is appropriate to start traffic rules and traffic safety training from childhood. Thus, they stated that education at older ages would be more successful. The aim of this project is to teach basic simple vehicle behavior in traffic. By conducting research for this purpose; Turning right, turning left, coming back, going forward and stopping were examined. It is aimed to increase perception by adding sound and light outputs to these actions. Additionally, the vehicle encountering an obstacle while driving and turning off and on the headlights depending on the amount of light are included in the analysis. It is planned to conduct a survey on these topics with a group of driver personnel. It will be eliminated after the results are listed. Thus, it is aimed to teach children the commands to direct the vehicles and understand the movements. For ease of use, it is planned to be controlled via a mobile phone.

Keywords: Child, Traffic, Vehicle Action

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
SİMGELER.....	iv
KISALTMALAR	v
TABLolar LİSTESİ.....	vi
GRAFİKLER LİSTESİ	vii
ŞEKİLLER LİSTESİ.....	viii
GİRİŞ	1
BİRİNCİ BÖLÜM.....	3
KAYNAK ARAŞTIRMASI.....	3
1.1. Eğitim ve Trafik İlişkisi	4
1.1.1. Eğitim Nedir?.....	4
1.1.2. Trafik Nedir?	4
1.2. Çocuklara Yönelik Ülkemizde Gerçekleştirilen Trafik Eğitimleri.....	6
1.2.1. Sokakta İlk Adımlar Yol Güvenliği Eğitim Projesi	6
1.2.2. Çocukların Trafikte Güvenliği Projesi	6
1.3. Projede Kullanılan Malzemelerin Özellikleri	6
1.3.1. Arduino Elektronik Kontrol Kart Sistemi	6
1.3.2. HC-05 Bluetooth Modülü.....	11
1.3.3. HW-131 Güç Modülü	11
1.3.4. L298N Güç Modülü	12
1.3.5. HC-SR04 Ultrasonik Sensör Modülü.....	13
1.3.1. Sg90 Servo Motor	13
1.3.2. DC Motor.....	14
1.3.3. DF Player Mini Mp3 Player Modülü	15
1.3.4. Mikro SD Kart	16
1.3.1. Tasarımda Kullanılacak Diğer Devre Elemanları.....	17
1.4. Tasarım Aşaması	17
1.4.1. Projenin Komut Çalışma Prensiplerinin Oluşturulması	17
1.4.2. Projenin Mekanik Çalışma Prensiplerinin Oluşturulması	19
1.4.3. Projenin Tasarım Aşamasında Blok Diyagramın Çıkarılması	21
1.4.4. Bağlantı Şemasının Hazırlanması	23
1.4.5. Li-Ion Pili Projedeki Çalışma Yapısı	26
1.4.6. L298N Modülünün Projedeki Çalışma Yapısı	27
1.4.7. HW-131 Modülünün Projedeki Çalışma Yapısı	28
1.4.8. Ultrasonik Sensörün Projede Çalışması ve Mesafe Ölçümü	28
1.4.9. Sg90 Servo Motorun Projedeki Çalışma Yapısı	31
1.4.10. Mp3 Player Modülünün Projedeki Çalışma Yapısı	32
1.4.11. HC-05 Bluetooth Modülünün Projedeki Çalışma Yapısı.....	37
1.4.12. Fiziksel Montaj	38
1.4.13. Proje Maliyetinin Hesaplanması	38
SONUÇLAR VE ÖNERİLER.....	41
KAYNAKÇA.....	42
EKLER.....	A

SİMGELELER

A	:	Amper
°C	:	Celsius
cm	:	Centimeter
GHz	:	Gigahertz
KB	:	Kilobayt
KHz	:	Kilohertz
mA	:	Miliamper
MB	:	Megabayt
MHz	:	Megahertz
mm	:	Milimetre
R	:	Resistance
s	:	Saniye
T	:	Time Constant
V	:	Volt

KISALTMALAR

AC	:	Alternating Current –Asi Akım
DC	:	Direct Current –Doğru Akım
EEPROM	:	Electronically Erasable Programmable Read Only Memory– Elektrikle Silinebilir Programlanabilir Sadece Okunabilir Bellek
FAT	:	File Allocation Table - Dosya Ayırma Tablosu
GND	:	Ground - Topraklama
HID	:	Human Interface Device – İnsan Arayüz Cihazı
I/O	:	Input/Output – Giriş/Çıkış
ICSP	:	In - Circuit Serial Programming - Devre Üzerinde Seri Programlama
LDR	:	Light Dependent Resistor - Işığa Bağımlı Direnç
MEB	:	Milli Eğitim Bakanlığı
Mp3	:	Motion Pictures Experts Group Audio Layer 3 - Film Uzmanlar Grubu Ses Katmanı 3
PC	:	Personal Computer – Kişisel Bilgisayar
PWM	:	Pulse Width Modulation – Sinyal Genişlik Modülasyonu
RAM	:	Random Access Memory – Rastgele erişimli hafıza
SDHC	:	Secure Digital High Capacity - Yüksek Kapasiteli Güvenilir Dijital Kart
SPI	:	Serial Peripheral Interface – Seri Çevresel Arabirim
TTL	:	Transistor–Transistor Logic – Transistör - Transistör Mantık Devresi
TÜİK	:	Türkiye İstatistik Kurumu
UART	:	Universal Asynchronous Receiver - Transmitter – Evrensel Asenkron Alıcı-Verici
USB	:	Universal Serial Bus – Evrensel Seri Veri yolu
Wi-Fi	:	Wireless Fidelity – Kablosuz Bağlantı

TABLÖLAR LİSTESİ

Tablo 1 “Trafik Nedir” Soru Cevapları	5
Tablo 2 Arduino Model Özellik Tablosu	7
Tablo 3 Mesafe Ölçüm Test Tablosu	30
Tablo 4 Anket Sonuçlarına Göre Komut ve İşlem Tablosu.....	34
Tablo 6 Proje Kurulum ve Ömür Maliyeti Tablosu	39

GRAFİKLER LİSTESİ

Grafik 1 Zamana Göre Pil Şarj, Deşarj Grafiği	26
--	----

ŞEKİLLER LİSTESİ

Şekil 1	Arduino Mega Kartı Görünüşü.....	9
Şekil 2	Arduino Mega Devre Şeması.....	10
Şekil 3	HC-05 Bluetooth Modülü	11
Şekil 4	LM1117-XX Genel Devre Şeması.....	12
Şekil 5	L298N İle Tasarlanmış Örnek Motor Sürücü Devresi.....	12
Şekil 6	HC-SR04 Ultrasonik Sensör Çalışma Prensibi.....	13
Şekil 7	SG90 Micro Servo Motor Görüntüsü.....	14
Şekil 8	Uygulamamızda Kullanılan DC Motor Kiti.....	15
Şekil 9	Uygulamamızda Kullanılan DF Player Mp3 Modülü.....	16
Şekil 10	Mikro SD Kart ve Adaptörü	16
Şekil 11	Komut Akış Diyagramı	18
Şekil 12	Ackermann Yönlendirme Prensibi.....	19
Şekil 13	Ayna-Pinyon Dişli Sistemi.....	20
Şekil 14	Swedish Tekerleği.....	20
Şekil 15	Karma Sistem Dönüş.....	21
Şekil 16	Proje Blok Şeması.....	22
Şekil 17	Yükselteç Devresi	23
Şekil 18	Devremizin Bağlantı Şeması.....	25
Şekil 19	Arduino Bluetooth Control Programı.....	37
Şekil 20	Aracımızın Montajı Tamamlanmış Görüntü	38

GİRİŞ

Gelişen toplumumuzun yeni üyeleri çocuklarımız, çok meraklı olmalarının yanında oldukça kendini korumaya alma yönünde eyleme yönelir. Birçok çocuğumuz araç kullanım ve diğer özelliklerine oldukça ilgilidir. Lakin araç yapısının karmaşıklığı, kullanıcı kişinin davranış farklılığı, önceden belirlenmiş trafik kurallarının varlığı gibi nedenlerle eylemlerin anlamlarını öğrenmekte zorlanmaktadırlar. Aracın engel ile karşılaştığında ya da geri gelme durumunda nasıl eylemler yapacağını merak etmekte aynı zamanda zorlanmaktadırlar. Araçlar trafik kuralları nedeniyle eylemlerini gerçekleştirirken diğer araçlara veya yayalara gerçekleştirmek istedikleri eylem için bazı belirtiler yapmaları gerekmektedir. Örneğin sağ dönmek için sağ sinyalin yakılması gerekmektedir.

Araç kullanıcılarının zorunlu olarak yapması gereken bu trafik eylemlerini çocuklarımıza eğlenirken öğretmek için araç tasarımı yapılması planlanmıştır.

BİRİNCİ BÖLÜM

KAYNAK ARAŞTIRMASI

Bu bölümde, çocuklara bazı araç davranışlarını öğretmek amacıyla araştırma yapılmıştır. Temel araç davranışlarından sağ dönmek, sola dönmek, geri gelmek, ileri gitmek ve durmak durumları incelenmiştir. Ayrıca araç seyir halindeyken engel ile karşılaşması, ışık miktarına göre farların kapatılıp açılması incelemeye dahil edilmiştir. Bu başlıklar ile ilgili şoför personelden oluşan bir grup ile anket çalışması yapılması planlanmaktadır. Sonuçlar listelendikten sonra elemine edilecektir. Daha sonra komutlar çoğaltılacak ve ses dosyaları projeye uygun hale getirilecektir. Böylece çocuklara araçları yönlendirme komutlarını öğretmek ve hareketlerin anlaşılması amaçlanmıştır. Kullanım kolaylığı açısından komutları yazmak yerine ses ile iletilmesi planlanmaktadır. Bu işlem için; mobil olması ve kullanım alanı geniş olması nedeniyle cep telefonu kullanılması planlanmaktadır. Aracın enerji ihtiyacını karşılamak için, ekonomik ve pratik olması nedenleriyle şarj edilebilir özellikli pil kullanılması düşünülmektedir.

Sonraki bölümde ise önceki akademik ve uygulama çalışmaları incelenmiş tasarım aşaması bir sisteme oturtulmuştur. Akabinde iletişim konusuna giriş yapılmış olup elektronik iletişim özellikleri ve türlerine değinilmiştir. Projemizde kullanacağımız malzemeler incelenmiştir. Doğruluk karşılaştırması için bazı testler uygulanmıştır. Maliyet analizi yapılarak projenin bugünkü değeri hesaplanmıştır.

1.1. Eğitim ve Trafik İlişkisi

TÜİK(Türkiye İstatistik Kurumu) verilerine göre Nisan 2024 tarihi itibarıyla ülkemizde motorlu taşıtlar sayısı 29.561.690 adet olmuştur. Trafığe kayıtlı taşıtların %52,7'sini otomobil, %18,4'ünü motosiklet, %15,4'ünü kamyonet, %7,5'ini traktör, %3,3'ünü kamyon, %1,7'sini minibüs, %0,7'sini otobüs ve %0,3'ünü özel amaçlı taşıtların oluşturduğu belirtilmektedir (TÜİK,2024). Bir diğer konu ise trafik kazalarıdır. 2023 yılına ait kaza verileri incelendiğinde 23.571 adet kaza gerçekleşmiş olup bu kazalarda 6.548 kişi hayatını kaybetmiştir. Ölümler yaş gruplarına göre incelendiğinde, 0-17 yaş grubu toplam ölümlerin %10,1'ini, 18-24 yaş grubu %13,6'sını, 25-64 yaş grubu %56,3'ünü, 65 yaş ve üzeri ise %20,0'ını oluşturduğu görülmüştür. (TÜİK,2023)

Yukarıdaki bilgiler ışığında trafik kazalarını azaltmak için çalışma yapılması gerektiği görülmektedir. Yapılacak eğitimin küçük yaştan itibaren uygulanması algı verimini artıracak ve orta vadede sonucu olumlu yönde ekilecektir.

1.1.1. Eğitim Nedir?

Eğitim, belirlenen hedefler doğrultusunda bireylerin yaşantılarında, davranışlarında değişiklikler oluşturma sürecidir (Özdemir, 2004). Okul öncesi eğitim, çocuğun doğduğu günden temel eğitime başlandığı güne kadar geçen 0-6 yaş arasındaki dönemi kapsayan ve çocukların daha sonraki yaşamlarında çok önemli bir yeri olan, bedensel, sosyal, duygusal, zihinsel ve dil gelişimlerinin büyük ölçüde tamamlandığı, bu doğrultuda kişiliğin şekillendiği gelişim ve eğitim süreci olarak tanımlanabilir (Hatipoğlu, 2002). Araştırmalar, okul öncesi eğitimin çocuk gelişimini ve ondan sonraki eğitimi olumlu ya da olumsuz yönde etkilediğini ve sistemli bir okul öncesi eğitimin yaşamın diğer kademelerindeki başarıyı arttırdığını kesin olarak ortaya koymuştur (Tütüncü, 2001). Görüldüğü üzere eğitimin erken dönemde başlanması öğrenme etkisini artıracaktır.

1.1.2. Trafik Nedir?

Yasal koşulların belirlediği trafik kurallarını, kişinin yaşantısı sırasında doğal davranışlar şekline dönüştürebilmek, kişilerin can ve mal güvenliğini sağlamak için yapılması gereken çalışmaların tümüdür (Sönmez, 1992). Kişisel, sosyal, duygusal, zihinsel gelişimin ve kişiliğin şekillenmeye başladığı okul öncesi dönemde verilen trafik eğitimi son derece önemlidir. Emniyet kemeri takmamak, hız sınırının aşılması gibi genel trafik kurallarının ihlal edilmesi ilerleyen yaşlarda artık kalıplaşmış ve alışkanlık sonucu

sergilenen hareketler olmakta ve belli bir yaştan sonra kazanılan olumsuz davranışların eğitimle değiştirilmesi çok zor olmaktadır. Erken yaşlarda başlayan trafik eğitimi çocukların doğru ve güvenli davranış modellerini öğrenmesi, bunların benimsenip hayat tarzı haline getirilmesi açısından etkili olmaktadır. Bilgi ve beceriler ne kadar küçük yaşlarda öğrenilip uygulanırsa o derece kalıcı olmaktadır (Alisinanoğlu, 2009). Çocuklara verilen trafik eğitiminde göz önünde bulundurulması gereken en önemli şeylerden biri, çocuğun aldığı bu eğitimi, yetişkin yaşlarında yaşam tarzı olarak benimsemesini sağlayabilmektir (Pampal, 1997). Tanımlamaları yapılmıştır.

Çocuklarla yapılan bir çalışma aşağıdaki gösterilmiştir. Bu tablodaki verilere göre çocuklara “*Trafik Nedir*” sorusu sorulmuş ve cevaplar listelenmiştir.

Tablo 1 “Trafik Nedir” Soru Cevapları

Cevaplar	Sayı	Yüzde %
Yolda arabaların sıkışması	208	25,9
Arabaların gittiği yer	126	15,7
Bilmiyorum	111	13,8
Trafik ışıkları	111	13,8
Trafik kazası	47	5,8
Kurallara uymak	45	5,6
Yol	42	5,2
Araba kullanmak	39	4,9
Trafik polisi	36	4,5
Karşıya geçmek	19	2,4
Diğer	20	2,5

Kaynak: Hatipoğlu, (2011)

Çocukların trafik algısında %25,9’la ilk sırada, yolda arabaların sıkışması, yani trafik sıkışıklığı yer almaktadır. Bir sorun olan trafik sıkışıklığının çocukların trafik olgusunun temelini oluşturduğu görülmektedir. Çocukların %15,7’sinin trafik denilince akıllarına “*Arabaların gittiği yer*” yani yollar gelmektedir. Ne yazık ki %13,8’inin trafik hakkında bilgileri bulunmamaktadır. Dördüncü sırada verilen cevap “*trafik ışıkları*”, beşinci sırada ise “*trafik kazasıdır*”. Aynı trafik sıkışıklığı gibi bir sorun olan trafik kazasının da çocukların trafik algısına yerleştiği görülmektedir.

1.2. Çocuklara Yönelik Ülkemizde Gerçekleştirilen Trafik Eğitimleri

Yapılan araştırma çalışmasında Milli Eğitim Bakanlığı (MEB) tarafından Trafik Eylem Planı hazırlanmıştır. Bu eylem planında çocuklar için “Sokakta İlk Adımlar Yol Güvenliği Eğitim Projesi” ve “Çocukların Trafikte Güvenliği Projesi” gerçekleştirilmiştir.

1.2.1. Sokakta İlk Adımlar Yol Güvenliği Eğitim Projesi

Sokakta İlk Adımlar Yol Güvenliği Eğitim Projesi çalışması, ilk olarak 2002 yılında 10 yıllığına uygulamaya konulmuştur. İlk eğitim 2002-2003 eğitim döneminde başlanmıştır. İstanbul, İzmir, Ankara ve Bursa’da, belirlenen pilot okullarda geçirilmiştir. 2003-2012 yılları arasında nüfusun kazaya oranı yüksek olan 41 il daha proje kapsamına alınmış. Daha sonraki yıllarda ilgili müdürlük tarafından proje 3 yıl daha uzatılmış ve çalışmalar 2016 yılında tamamlanmış. (MEB, 2018). Sonraki yıllarda projenin yürütülüşüyle ilgili bir veri bulunamamıştır.

1.2.2. Çocukların Trafikte Güvenliği Projesi

Çocukların Trafikte Güvenliği Projesi, 2014 yılında Ankara’da başlatılmıştır. Bu proje kapsamında, ilkokul 2 ve 4. sınıflarında uygulanmıştır. Sonraki yıllarda, Ankara’da 32, Malatya’da 3, Kırıkkale’de 4, Karabük ve Adapazarı’nda 1’er okul olmak üzere toplamda 41 ilkokulda, 7.449 öğrenci ile uygulanmıştır (MEB, 2018). Sonraki yıllarda projenin yürütülüşüyle ilgili bir veri bulunamamıştır.

Görüldüğü üzere MEB tarafından çocuklara yönelik trafik eğitimleri bir süre uygulanmış ancak daha sonraki yıllarda devamlılığı sağlanmamıştır.

1.3. Projede Kullanılan Malzemelerin Özellikleri

1.3.1. Arduino Elektronik Kontrol Kart Sistemi

Arduino, donanım ve yazılım kısmından oluşan bir elektronik donanımdır. Yazılımı açık kaynak olup; herkes tarafından programlanabilir. Arduino sensörlerden aldığı bilgiyi alır ve bunu bir çıkışa dönüştürür. Kartta bulunan mikro denetleyiciye yaptırmak istediğiniz işi tarif ederek bir dizi talimat göndererek kartınıza ne yapması gerektiğini anlatabilirsiniz (Arduino.cc, 2018).

Farklı amaçlar ve uygulamalar için üretilmiş Arduino kartlarının bazıları aşağıdaki tablo 2’de özellikleriyle bildirilmiştir.

Tablo 2 Arduino Model Özellik Tablosu

Arduino Model Özellik Tablosu							
Model	UNO R4 Wi-Fi	UNO Mini	Zero	UNO Wi-Fi	Mega 2560	DUE	Giga R1 Wi-Fi
Mikro Denetleyici	Renesas RA4M1	ATmega328P	ATSAMD21G18	ATmega4809	ATmega2560	AT91SAM3X8E	STM32H747XI
Kablosuz İletişimi				Wi-Fi, Bluetooth			Wi-Fi, Bluetooth
Veri Şifreleme				ATECC608A			ATECC608A
Kablo Bağlantısı	USB	USB					HID, USB
Dijital I/O	14	14	20	14	54	54	76
Analog I/O	6/1	6	2	6	16	12/2	12/2
PWM	6	6	10	5	15	12	12
İletişim	UART, SPI	UART, SPI	UART, SPI		UART 4x, SPI		UART 4x, SPI 2x
Çalışma Gerilimi (V)	5	5	3,3	5	5	3,3	3,3
Giriş Gerilimi (V)	6-24	6-12V		6-20	7-12	7-12	6-24
Akım (mA)	8	20	7	20	20	800	8
Saat hızı (MHz)	48	16	48	16	16	84	480
Hafıza Kilobayt (KB)	RAM 32 FLASH 256	RAM 2 FLASH 32 EEPROM 1	RAM 32 FLASH 256	RAM 6 FLASH 48 EEPROM 0,25	RAM 8 FLASH 256 EEPROM 4	RAM 96 FLASH 512	RAM 1024 FLASH 2048
Ölçüleri (mm)	68/53	26/34	53/68	68/53	53/101	53/101	53/101

Kaynak: (Arduino, 2018)

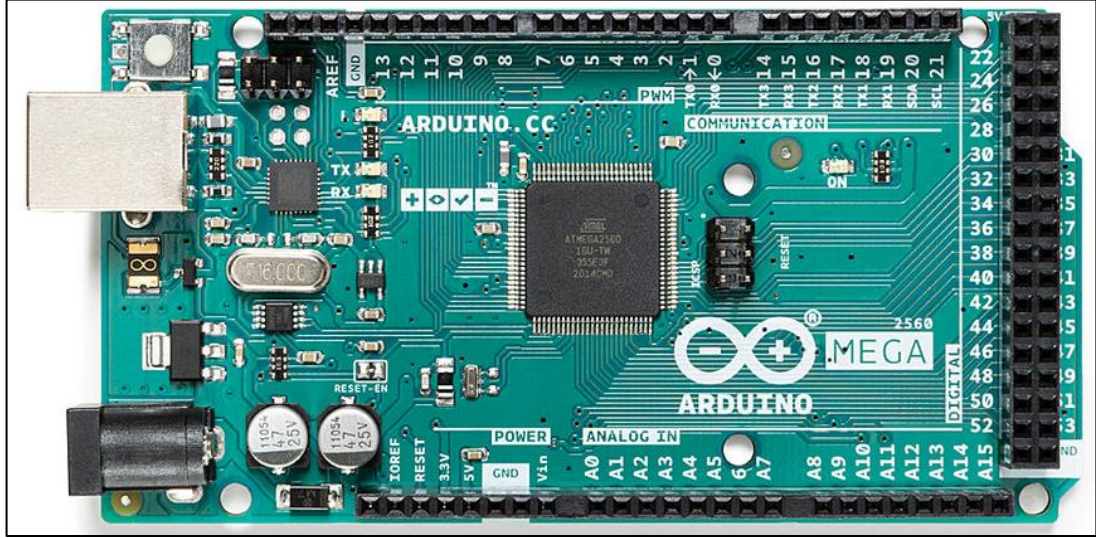
Tablo 2’de üzere bazı farklı Arduino kartlarının özellikleri tablo haline getirilmiştir. Modellerde; Renesas RA4M1, ATmega328P, ATSAM21G18, ATmega4809, ATmega2560, AT91SAM3X8E, STM32H747XI mikro denetleyiciler kullanılmıştır. Kablosuz iletişim için bazı modellerde Kablosuz Bağlantı (Wireless Fidelity - Wi-Fi), Bluetooth; kablolu iletişim için ise İnsan Arayüz Cihazı (Human Interface Device – HID), Evrensel Seri Veri yolu (Universal Serial Bus – USB) tipi bağlantı seçilmiştir. Sinyal giriş ve çıkışları için analog ve dijital portlar kullanılmış ve modellerine göre port 2 ile 76 arasında değişmektedir. Çalışma gerilimleri 3,3 ile 5 Volt arasında olmakla beraber giriş gerilimleri 6 ile 24 Volt arasında değişebilir. Kullanılan akım 7 ile 800 mA arasında değişebilir. Çalışma hızı ise 16 ile 480 MHz arasındadır. Farklı hafıza seçenekleri bulunmaktadır. Bunlar; Rastgele Erişimli Hafıza (Random Access Memory – RAM), Elektrikle Silinebilir Programlanabilir Sadece Okunabilir Bellek (Electrically Erasable Programmable Read Only Memory– EEPROM) hafıza türlerindedir.

Arduino Elektronik kontrol sisteminin çeşitli amaçlarda kullanılabilmesi için birçok amaçla tasarlanmış kontrol kartları ve yardımcı ürünleri bulunur. Kontrol kartları aşağıda açıklanmış olup yardımcı elemanlara, sensörler, motor sürücü devreleri, kamera devresi, örnek olarak verilebilir.

1.3.1.1. Arduino Mega

Arduino, processing /Wiring dilini kullanarak çevre elemanları ile temel çıkış uygulamalarını gerçekleştiren açık kaynaklı fiziksel programlama platformudur (Ö. Güngör, 2015). Arduino kart ucuzdur ve ihtiyaca uygun olarak satın alınan shield denilen çevre birimleri vasıtasıyla genişletilebilir. Arduino çok ucuz ve kolay erişilebilir bir kart olmasına karşın çok işlevseldir. Akademik ve öğrenci projeleri için ideal çözümler sağlar(H-S. Juang, K-Y. Lum, 2013). Arduino Mega, ATmega2560 işlemcisi temelli bir mikroişlemci kartıdır. Arduino Mega da 54 adet dijital giriş/çıkış pini bulunmaktadır ve bunların 14’ü PWM çıkışı olarak kullanılabilir. On altı adet analog giriş pini ve 4 adet Evrensel Asenkron Alıcı-Verici (Universal Asynchronous Receiver - Transmitter – UART) bulunmaktadır. 16 MHz kristal osilatörü, USB bağlantısı, bir güç girişi, bir Devre Üzerinde Seri Programlama (In - Circuit Serial Programming – ICSP) başlığı ve reset butonu kart üzerinde mevcuttur. Çalışma gerilimi olarak DC 7~12V ihtiyaç duymaktadır. Bu kart mikro denetleyicinin ihtiyaç duyduğu her şeyi üzerinde barındırır.

USB kablo ile bilgisayara kolayca bağlanabilir. Ayrıca istenirse Alternatif Akım- Doğru Akım (Alternating current - Direct current, AC-DC) adaptör veya pil vasıtasıyla da başlatılabilir (N.Ya'acob, 2016). Bir Arduino Mega mikro işlemci kartının üstten görünüşü aşağıdaki şekilde verilmiştir.



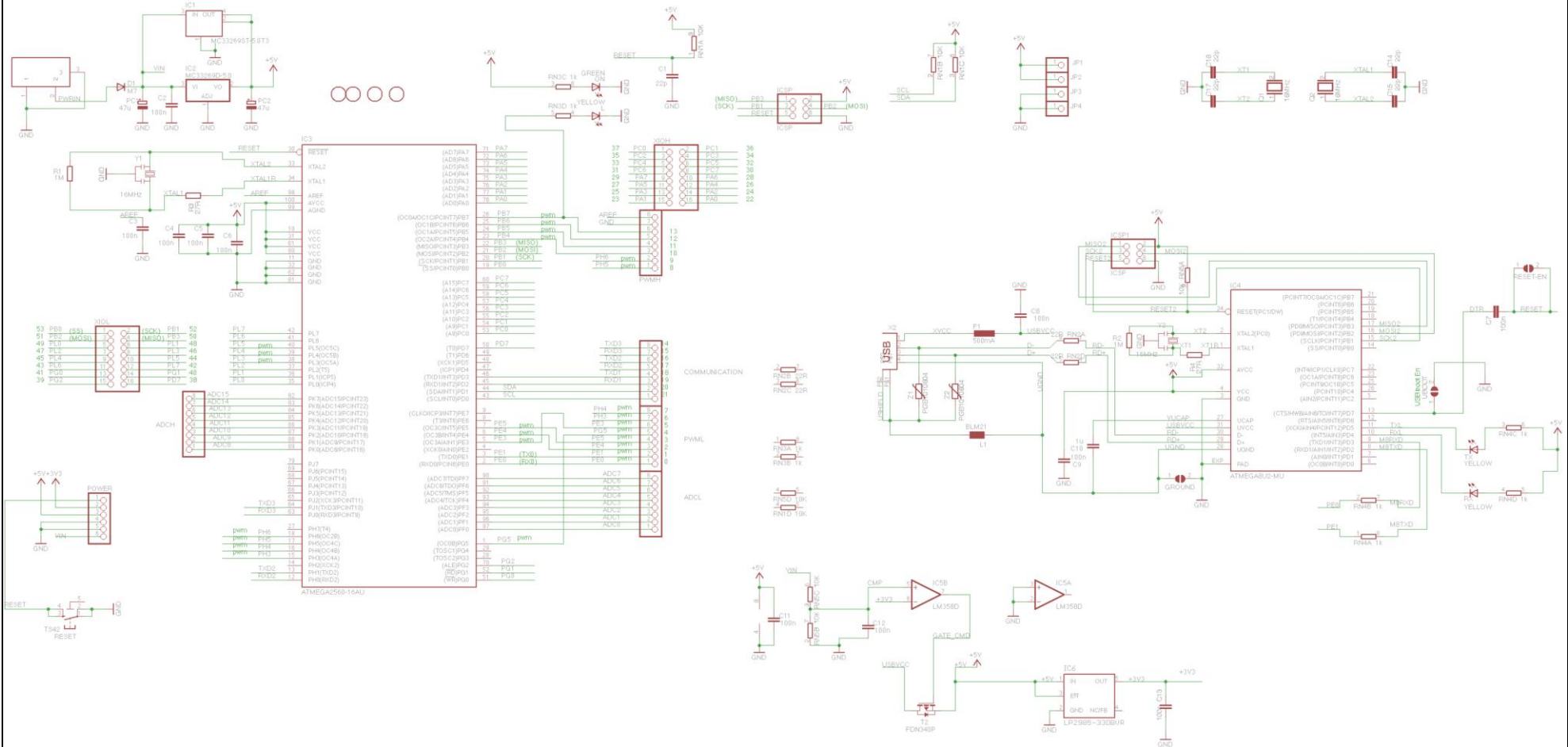
Şekil 1 Arduino Mega Kartı Görünüşü

Kaynak: (Arduino.cc,2024)

Arduino Mega kartı Atmega 2560 işlemci üzerine tasarımı yapılmıştır. Kartta bir adet adaptör besleme girişi, USB programlama girişi, reset tuşu, koruma devresi, saat devresinden oluşmaktadır. Arduino ide programı ile yazılım tasarlanabilir. Bu program ile pinler tanımlanarak işlemcinin çalışma düzeni belirlenir.

Aşağıda Arduino Mega kartının devre şeması görülmektedir. Üreticinin sitesinden alınan bu bilgilerle, farklı firmalar tarafından kopya kartlar üretilmektedir.

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

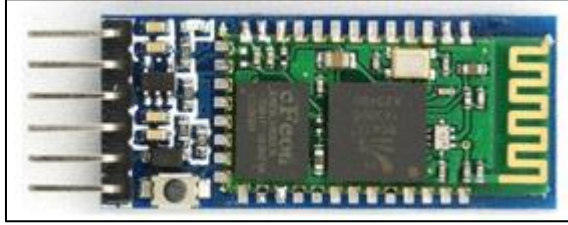


Kaynak: (Arduino.cc,2024)

1.3.2. HC-05 Bluetooth Modülü

Ferreira ve arkadaşları yaptıkları sistem kontrolünü Android sistemli akıllı telefonlara yüklenebilir uygulama yazılımı üzerinden sağlamışlardır. Akıllı telefon ile mikro denetleyici arası veri iletişimi HC-05 Bluetooth modülü kullanarak gerçekleştirmişlerdir (Ferreira &, 2020)

Bluetooth modülü olarak Bluetooth 2.0 destekleyen HC-05 kullanılmıştır. Bu modül 2.4 Giga hertz (GHz) frekans bandında haberleşme sağlamaktadır. 460800 baud hızına kadar iletişimi desteklemektedir. Uygulamada 9600 standart değeri kullanılmıştır (HC-05, 2010). Aşağıda HC-05 Bluetooth modülü resmi görülmektedir.



Şekil 3 HC-05 Bluetooth Modülü

Kaynak: (Components,2010)

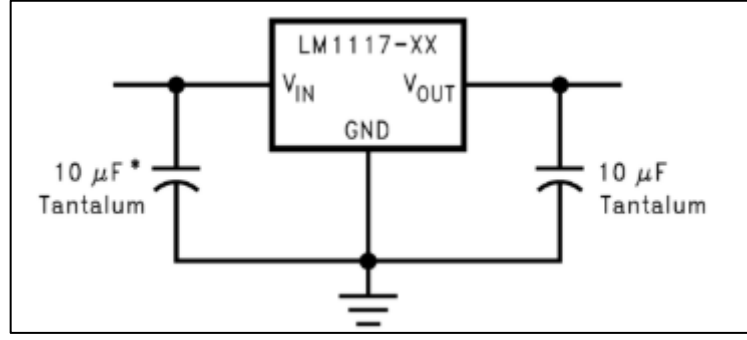
Robotik uygulamalarda oldukça sık kullanılan bu modül, mikro işlemci ile seri port üzerinde haberleşmektedir.

1.3.3. HW-131 Güç Modülü

MB102, Breadboard güç modülü'nün üzerinde, 3.3V ve 5V çıkışlarını seçmek için iki adet seçim düğmesi bulunur. Ayrıca modül üzerinde, çıkışlara bağlanacak olan pozitif ve negatif bağlantı noktaları da mevcuttur. LM1117 3,3 ve 5 V doğrusal voltaj regülatörü ile tasarlanmıştır (Texas instruments, 2023). Üretici katalog bilgisine göre 25°C sıcaklıkta;

LM1117 3,3 versiyonu, $4,75V < V_{giriş} < 12V$ arası giriş geriliminde, $0\text{ mA} < I_{çıkış} < 800\text{ mA}$ çıkış akımı,

LM1117 5,0 versiyonu, $6,5V < V_{giriş} < 12V$ arası giriş geriliminde $0\text{ mA} < I_{çıkış} < 800\text{ mA}$ çıkış akımı verebilmektedir. Aşağıdaki resimde LM1117-XX genel devre şeması gösterilmiştir.



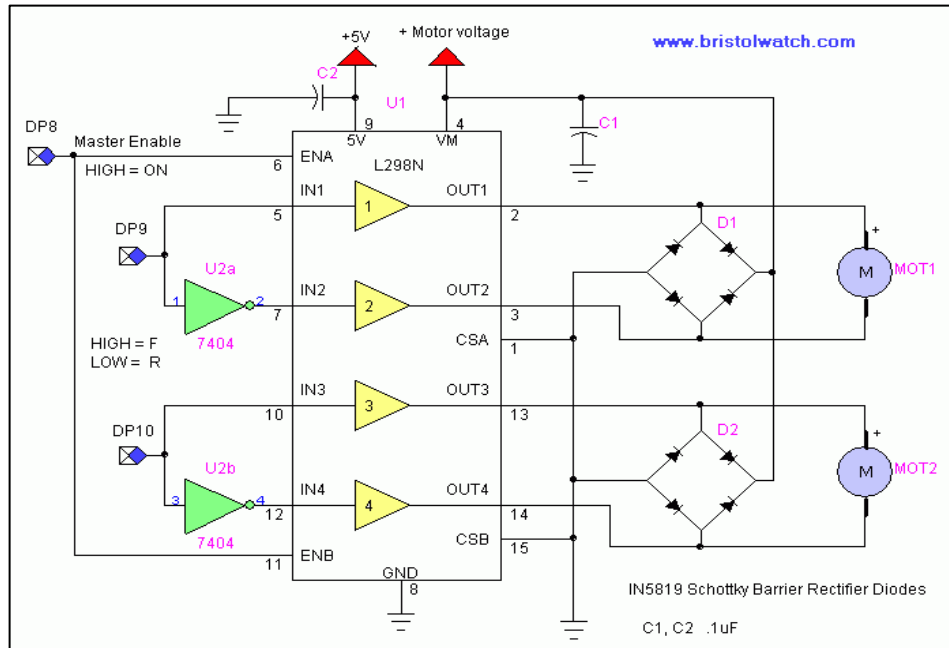
Şekil 4 LM1117-XX Genel Devre Şeması

Kaynak: (Texas instruments, 2023).

Yukarıdaki şekilde Texas instruments firması ürettiği LM1117-XX genel devre şeması gösterilmiştir. LM1117 entegresinin farklı sabit çıkışlı ve ayarlanabilir çıkışlı türleri olmasından dolayı şematik yapıda versiyon XX olarak gösterilmiştir.

1.3.4. L298N Güç Modülü

L298N; transistör - transistör mantık devresi (Transistor–transistor logic, TTL)sistemine göre çalışır. TTL sistemleri, ilk transistör mantık, ikinci transistör de yükseltme işlemini yapar. Endüktif yükleri sürmek için üretilmiştir. Bunlara; Röle, selenoid ve motor örnek olarak verilebilir. İki adet tam köprü sürücü devre elemanıdır. Maksimum çıkış akımı 2,5 Amper, çalışma gerilimi 5-50 Volt, tetikleme gerilimi -0,3-7 Volt 'tur. (ST Microelectronics, 2023).



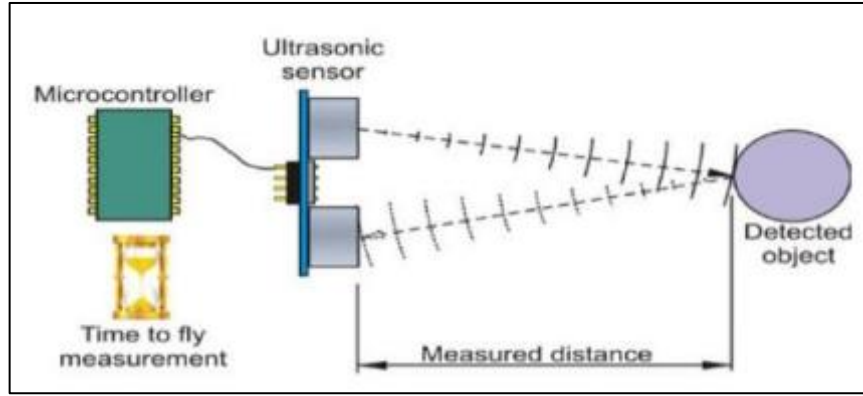
Şekil 5 L298N İle Tasarlanmış Örnek Motor Sürücü Devresi

Kaynak: (bristolwatch.com,2024)

Yukarıda L298N tipi entegresi ile tasarlanmış motor sürücü devresi görülmektedir. Devrede 7404 entegresi kullanarak entegrenin 2 adet NOT kapısı kullanılmıştır. Motorlar köprü diyot üzerinden devreye bağlanmıştır. Ayrıca modülde bir adet 7805 entegresi bulunur.

1.3.5. HC-SR04 Ultrasonik Sensör Modülü

Ultrasonik sensörler 1980'li yılların ortalarından itibaren otonom mobil robotların engellerden kaçınmaları ve harita oluşturmada kullanılmaktadır. Ultrasonik sensör üzerindeki verici tarafından ses dalgası yayılır. Bu ses dalgası engele çarpıp yansır. Yansıyan bu dalga ultrasonik sensör üzerindeki alıcı tarafından algılanması için geçen süre kullanılarak mesafe ölçülür (S. Kim, H. B. Kim, 2010).



Şekil 6 HC-SR04 Ultrasonik Sensör Çalışma Prensipleri

Kaynak: (V.A. Akpan & A.S. Eyefia, 2021)

Yukarıdaki şekilde görüldüğü üzere HC-SR04 mikro işlemci tarafından yönetilir. Ultrasonik sensör, tarafından üretilen ses dalgası verici ile gönderilir, engele çarpar ve sensörün alıcısı tarafından alınarak gönderilen dalga gidip gelme süresi kontrole iletilir.

1.3.1. Sg90 Servo Motor

Servo motor, sistemin yukarı-aşağı veya sağa-sola kontrol edilmesi gereken birçok uygulamada kullanılan geri bildirimli DC tipi motorlardan biridir. Servolar, robotikte son derece yararlıdır. Servo motor, düşük devirde yüksek Tork sağlar. Bir servo motordan çıkan üç kablo vardır. Bunlardan ikisi güç ve toprak, diğeri ise motora konum kontrol sinyalinin gönderildiği çıkıştır. Servo motor bir seri darbe (pulse) ile kontrol edilir. Bu darbenin uzunluğu servonun alacağı pozisyonu belirler (W. M. Kadir ve Ark, 2012). Aşağıdaki resimde tasarımıımızda kullanılan servo motorun resmi görülmektedir.



Şekil 7 SG90 Micro Servo Motor Görüntüsü

Kaynak: (Yazar Tarafından Çekilmiştir).

Resim de görülen mikro servo motorumuz 3,3-5 V DC gerilim ile çalışmaktadır. Üç adet kontrol kablosu bulunmaktadır. Bunlar; GND, V_{in} ve konumlandırma sinyal kablosudur. Yapısı içinde konumlandırılmış dişli sistemi ile mekanik olarak hızı yavaşlatılarak daha kontrollü bir hıza indirgenmiş ve sıkıştırılan bu güç motorumuzun torkunu güçlendirmek için kullanılmıştır. Uygulamalarda farklı açı ve yapıda olan mikro servo motorlar kullanılmaktadır.

1.3.2. DC Motor

DC Motor: Doğru akım elektrik enerjisini mekanik enerjiye dönüştürme işlemi yapmaktadır. Doğru akım motorlarının yönü ve şiddetleri zamanla değişime uğramaz. DC Motorların temel çalışma prensibi şu şekildedir; manyetik alan içerisinde geçen iletken bir tel üzerinden akım geçtiği zaman bu iletken telde bir hareket gerçekleşmesi üzerine dayanır. Bu içerisinde akım geçen iletken tel, eğer bir manyetik alan içerisine dâhil edilirse iletken bir kuvvet etkimektedir (Bülbül ve Sözbilir, 2017).

Yazılım Bileşeni Robotik kodlama alanı gelişme oldukça açık ve hemen hemen her yaş grubuna hitap eden çalışmaların yapıldığı bir alandır. Bundan dolayı dünya çapında kullanılan çok çeşitli yazılım programları geliştirilmektedir (Sancak, 2003). Aşağıda uygulamamız için seçtiğimiz DC motor kiti görülmektedir.



Şekil 8 Uygulamamızda Kullanılan DC Motor Kitleri

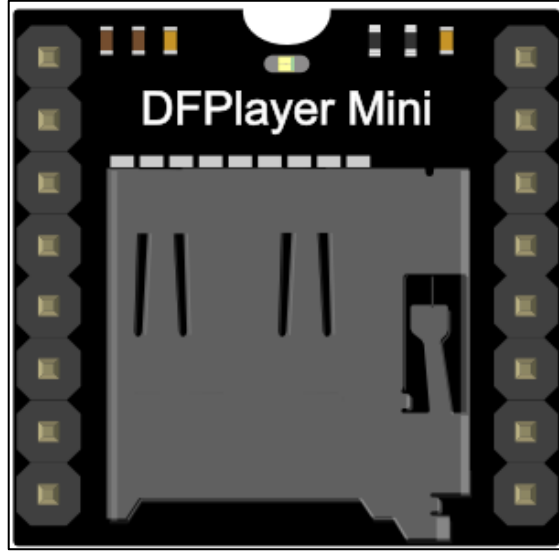
Kaynak: (Yazar Tarafından Çekilmiştir).

Birçok uygulamada kullanılan 6 V DC motor kit halinde redükte edilmiş ve tekerlekli olarak temin edilmektedir. Bu motorumuzda servo motorumuz gibi dişliler ile hızı düşürülerek dönüş torku artırılmıştır.

1.3.3. DF Player Mini Mp3 Player Modülü

DF player mini mp3 modülü, komutları seri port üzerinden almaktadır ve bundan dolayı Arduino, PC ve Bluetooth gibi kablosuz iletişimde kullanılmaktadır. Modül üzerinde mevcut olan buton girişleri ile de ek bir donanım olmadan kullanmak mümkündür. Çalışma gerilimi 3.3V- 5V'dur. (Yücel ve Ark., 2021). Modülün genel özellikleri ve üstün görünümü aşağıda belirtilmiştir.

- Örneklem oranları (KHz): 8/11.025/12/16/22.05/24/32/44.1/48
- 24 bit çıkışı, 90dB dinamik aralık desteği
- FAT16, FAT32 dosya sistemini tam olarak destekler,
- Maksimum 32G kart desteği,
- 100'e kadar klasörü destekler, her klasörde 255'e kadar şarkı bulunabilir
- 30 seviyeden oluşmuş ayarlanabilir ses seviyesi özelliklerine sahiptir.



Şekil 9 Uygulamamızda Kullanılan DF Player Mp3 Modülü
Kaynak: (wiki.dfrobot.com, 2016).

Üreticinin verisine göre projemizde kullanacağımız ses dosyalarının mp3 formatında olması gereklidir. Bu modül üzerinden direk ses çıkışı alınabilir. Mikro işlemci ile iletişimi için seri port kullanmak gereklidir.

1.3.4. Mikro SD Kart

Mikro SD kartların kapasiteleri 512 MB'tan 2 GB'a kadar, mikro SDHC kartların kapasitesi ise 4 ila 32 GB arasındadır. Mikro SDHC kart için teorik üst kapasite sınırı 32 GB'dır. Mikro-SDHC kartları dört hız sınıfında sahiptir. 2, 4, 6 ve 10. Sınıf ne kadar yüksek olursa kart o kadar hızlı ve aynı zamanda daha pahalı olur (René Durup, 2013). Aşağıda Projemizde kullanılan Mikro SD kartımız ve bilgisayardan veri yüklemek için gerekli adaptör görülmektedir.



Şekil 10 Mikro SD Kart ve Adaptörü
Kaynak: (Yazar Tarafından Çekilmiştir).

Yukarıda görüldüğü üzere projemizde kullanmak üzere 500 Mega Byte (MB) kapasiteye sahip Mikro SD kart ve bu kartımıza gerekli ses dosyalarını kopyalamak için gerekli ara adaptör görülmektedir.

1.3.1. Tasarımda Kullanılacak Diğer Devre Elemanları

Bu tasarımıımızda kullanılan diğer devre elemanları aşağıda listelenmiştir.

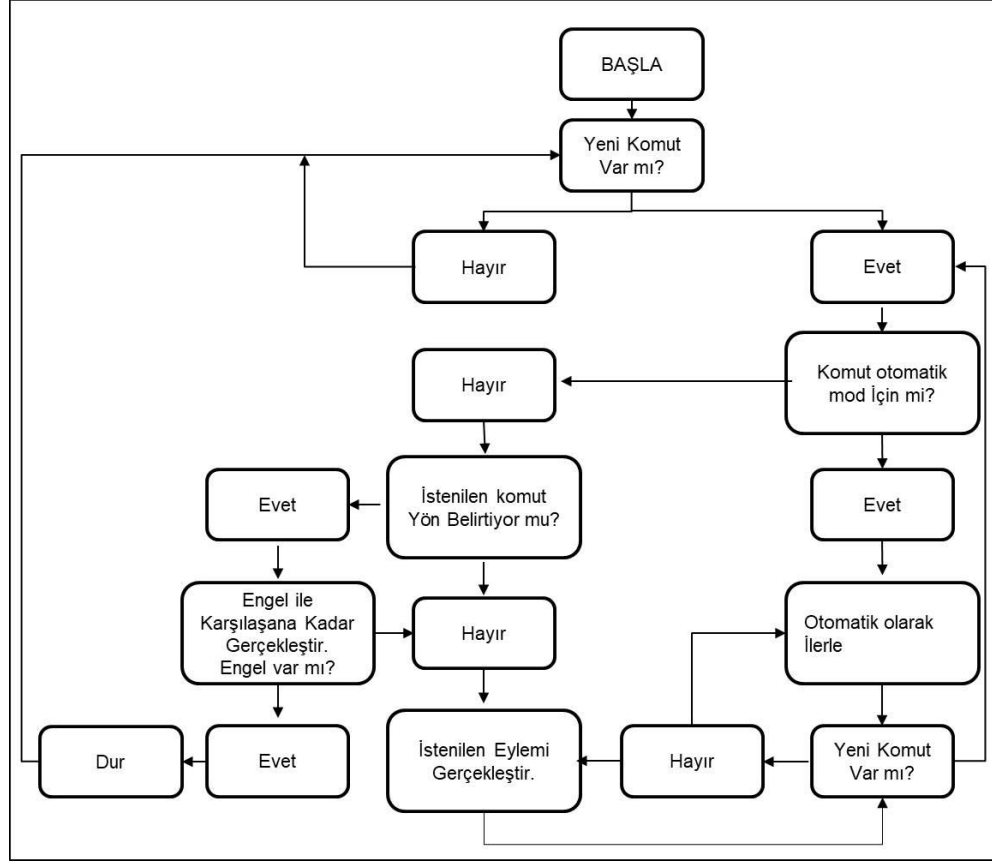
- Hazır olarak elde edilmiş çeşitli bağlantı kabloları
- Farklı renklerde LED diyot,
- Farklı değerlerde direnç,
- BC237 NPN Transistör,
- 1N5408 Diyot,
- LDR,
- Ses Çıkışı İçin Hoparlör'dir.

1.4. Tasarım Aşaması

Bu çalışmada ilk olarak Arduino UNO kullanılmış fakat proje üç adet seri port iletişimine ihtiyaç olması, giriş ve çıkış kontrollerinin yetersiz olması nedeniyle Arduino UNO kartının projeye uygun olmadığı görülmüştür. Arduino Mega kontrol kartının daha uygun olacaktır.

1.4.1. Projenin Komut Çalışma Prensibinin Oluşturulması

Projenin ilk olarak çalışma prensibi oluşturulmuştur. Çalışma prensibi aşağıda gösterilmiştir.



Şekil 11 Komut Akış Diyagramı

Kaynak: (Yazar Tarafından Oluşturulmuştur).

Yukarıda görüldüğü üzere araç, oluşturulan komutlara göre döngüye girmektedir. Eğer ki herhangi bir komut yoksa yeni bir komut oluşturulana kadar beklemektedir. Bir komut aldığında ise komutu ses ile kullanıcı kontrolünde mi kullanılacak yoksa otomatik konumda mı kullanılacak ayırımı yapılır. Ayırım sonrası komut gereği kullanıcı tarafından yönlendirilecekse komutun hareket içerip içermediği kontrol edilir. Kullanıcı tarafından oluşturulan komut farklı eylemler içerebilir. Bu eylemler ileri, geri, sağa, sola hareket olabileceği gibi farları yakmak ya da kapatmak gibi komutlarda olabilir. Buradaki ayırım aracın hareket halinde iken zarar görmemesi nedeniyle eklenmiştir. Eğer komutumuz yer değiştirme içermiyorsa sadece istenilen komutu gerçekleştirir ve yeni komutu beklemeye geçer. Sesli kontrol bölümünde yer değiştirme içeriyorsa araç komutu engel ile karşılaşana kadar gerçekleştirir ve yeni komut için beklemeye geçer.

Araç, otomatik konumda kullanıldığında ileri gider. Engelle karşılaştığında güvenli mesafede durur. Servo motor ve ultrasonik sensör yardımı ile sağ ve sol engel mesafesini ölçer. Dönüş hareket yönü, uzak olan engel yönüdür. Araç sola dönecekse; sağ motorlara ileri, sol motorlara geri gidecek şekilde güç verilir. Sağa dönecekse; sol motorlara ileri, sağ

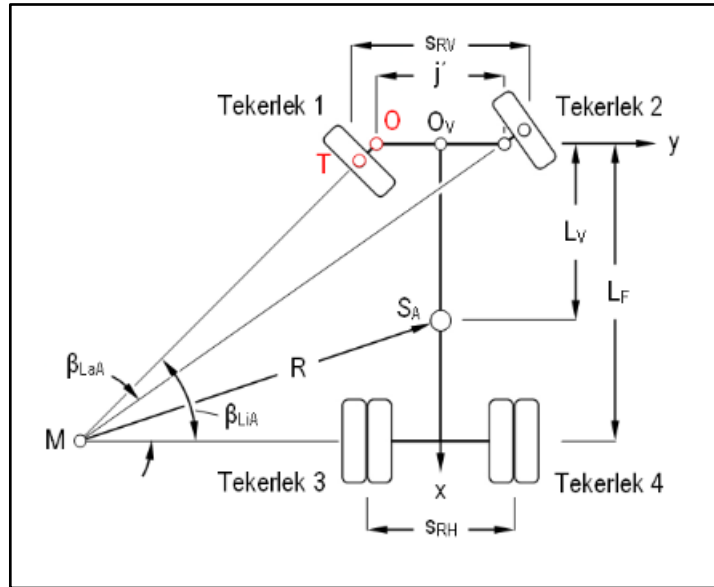
motorlara geri gidecek şekilde güç verilir. Bu dönüş yöntemi ile dönüş çapı küçültülmüştür. Dönüş sonrası araç engeli geçerek ilerlemeye devam eder.

1.4.2. Projenin Mekanik Çalışma Prensibinin Oluşturulması

Kara taşıtlarının yönlendirilmesi için birçok yöntem kullanılmaktadır. Bunlardan bazıları aşağıda listelenmiştir.

- Ackermann Yönlendirme Prensibi
- Ayna ve Pinyon Dişli Yapısı
- Swedish Tekerleği
- Omni Tekerleği

Günümüzde kara taşıtlarında Ackermann prensibi yaygın olarak kullanılmaktadır. Dönüşlerde iç tekerlek ve dış tekerlek açılarının farklı olması nedeniyle tasarlanmıştır. Mekanik yöntem ile tekerlekleri farklı açılarda döndürür. Aşağıda Ackermann prensibi ile ilgili görsel verilmiştir.



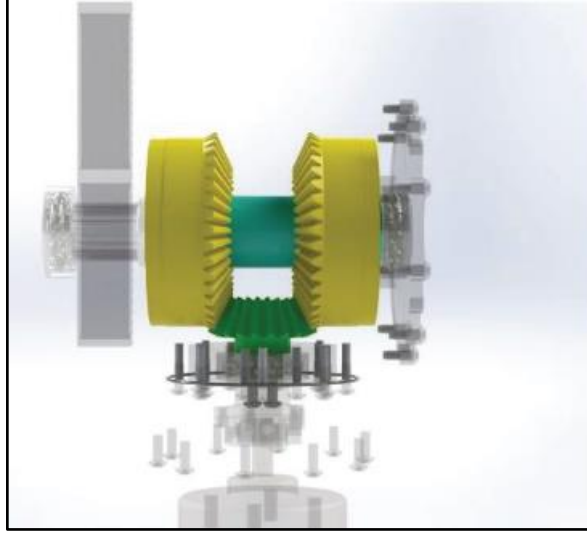
Şekil 12 Ackermann Yönlendirme Prensibi

Kaynak: (Topaç ve Ark, 2016)

Yukarıda Ackermann yönlendirme sistemini açıklamak için şekil verilmiştir. Şekilde görüldüğü üzere tekerlek merkezlerinden oluşturulan üçgen yardımıyla yöntem anlatılmıştır.

Ayna ve Pinyon dişli sistemiyle kullanılan bir başka döndürme sistemi genellikle paletli sistemlerde kullanılmaktadır. Bu yöntemde yönlendirmelerde mekanik yöntemler

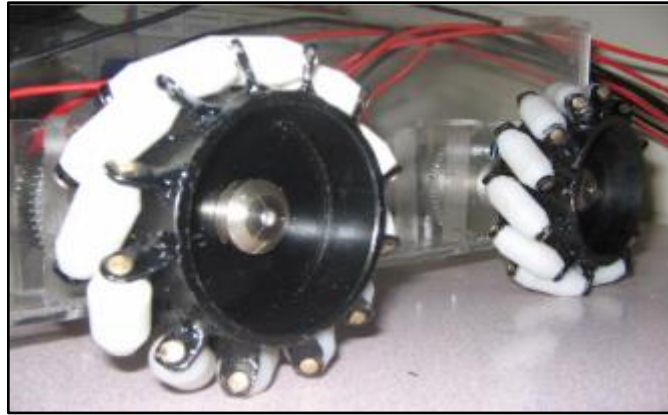
kullanılmıştır. Genelde iki farklı yöntem kullanılır. Dönölmek istenilen yöndeki paletler durdurulup karşı paletler döndürölür böylece araç kendi ekseni etrafında yarıçap oluşturarak döner. Diğer bir yöntemde ise dönmek istenilen yöndeki paletler geriye, diğer paletlerde ileri döndürölür. Bu yöntem ile aracın dönüş yarıçapı aracın yarıçapı kadardır. Böylece daha dar bir alanda araç dönebilir. Aşağıdaki bu sistem ile ilgili görsel eklenmiştir.



Şekil 13 Ayna-Pinyon Dişli Sistemi

Kaynak: (Tuncer D., Ulu, E.Y. , 2023)

Swedish ve Omni tekerleği tasarımlarında ise mekanik yönlendirme kullanılmaz. Her tekerlekte bir motor bulunur. Gidilmesi istenilen yöne göre motorlara yön verilir. Böylece araç yönelme işlemini radyan olarak değil doğrusal hareketler ile yapar. Lastiklerin yapısı sisteme göre tasarlanmıştır. Aşağıdaki şekilde Swedish tekerleğine ait bir görsel verilmiştir.

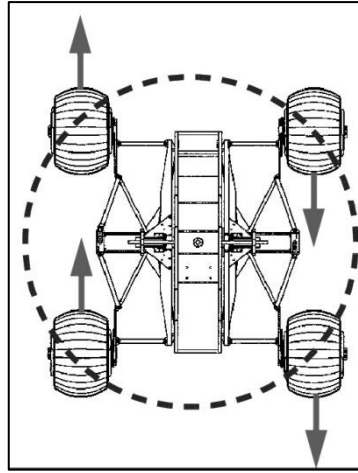


Şekil 14 Swedish Tekerleği

Kaynak: (Soygüder S., Alli, H., 2008)

Şekilde Swedish tekerleği yapısı görülmektedir. Yapısı gereği yönlenme ya da 45° açısı ile olur.

Tasarımda karma sistem kullanılmıştır. Her tekerleği çeviren bir adet motor kullanılmıştır. Motorlar yönetime göre hareket ettirilir. Bu yöntem ile dönüş yarıçapı aracın yarısı olarak kullanılır. Ayrıca tekerlekler standart lastik tekerlek kullanıldığı için maliyet düşürülmüştür. Dönüşler radyan olarak gerçekleştirilir. Aşağıda örnek şekil verilmiştir.



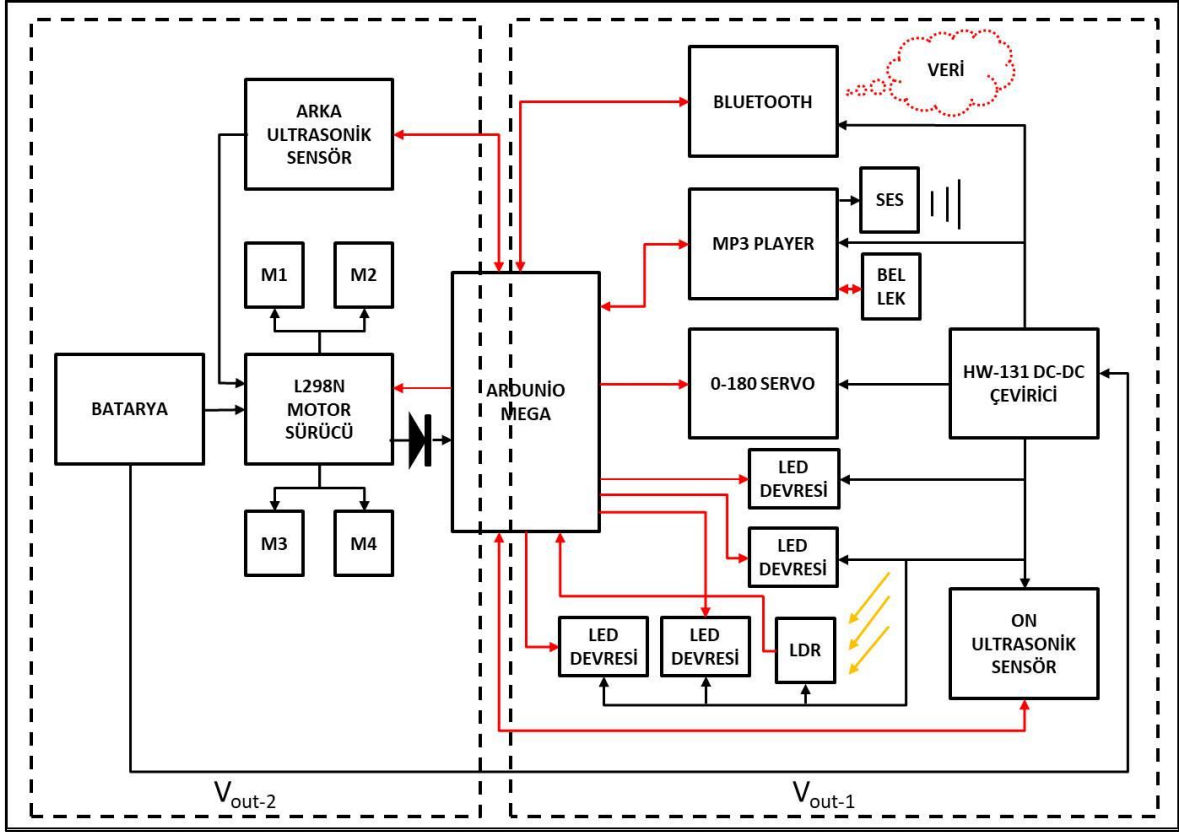
Şekil 15 Karma Sistem Dönüş

Kaynak: (Shamah, B., 1999)

Araç, otomatik konumda kullanıldığında ileri gider. Engelle karşılaştığında güvenli mesafede durur. Servo motor ve ultrasonik sensör yardımı ile sağ ve sol engel mesafesini ölçer. Dönüş hareket yönü, uzak olan engel yönüdür. Araç sola dönecekse; sağ motorlara ileri, sol motorlara geri gidecek şekilde güç verilir. Sağa dönecekse; sol motorlara ileri, sağ motorlara geri gidecek şekilde güç verilir. Bu dönüş yöntemi ile dönüş çapı küçültülmüştür. Dönüş sonrası araç engeli geçerek ilerlemeye devam eder.

1.4.3. Projenin Tasarım Aşamasında Blok Diyagramın Çıkarılması

Sonraki aşamada projemizde kullanacağımız donanımların blok şeması oluşturulmuştur. Aşağıdaki Şekil 14 de projenin birinci seviye blok şeması verilmiştir.



Şekil 16 Proje Blok Şeması

Kaynak: (Yazar Tarafından Oluşturulmuştur).

Şekil-14'deki kırmızı hatlar veri, siyah hatlar ise çalışma gerilimi için gerekli hatları göstermektedir. Bazı veri hatları tek yönde olurken bazıları ise çift yönlüdür.

Çift yönlü veri hatlarımız, ön ve arka ultrasonik sensörlerin data bağlantı hatları, Bluetooth iletişim hattı, mp3 player data hattıdır. Bluetooth, mp3 player ve bilgisayar bağlantı çıkışları seri haberleşme ile çalışmaktadır. Böylece Arduino Mega kartımızın yapısındaki tüm seri iletişim çıkışları kullanılmıştır. Tek yönlü sinyal çıkışlarımız ise, ön, arka, sağ ve sol LED çıkışları, servo motor sinyal çıkışı, motor sürücüsü çıkışları, LDR devresidir.

Şekilde görüldüğü üzere bir tane DC-DC çeviricisinin devreye yeterince akım sağlayamaması nedeniyle ikinci DC-DC çevirici olan HW-131 modülü eklenmiş ve devre güç dağılımı olarak iki bölüme ayrılmıştır. HW-131 DC-DC modülüne bağlı olan cihazlarımız;

- Bluetooth modülünü,
- Mp3 player modülünü,
- Servo motor,

- Ön ultrasonik sensör,
- LED devreleri,
- LDR devresini beslemektedir.

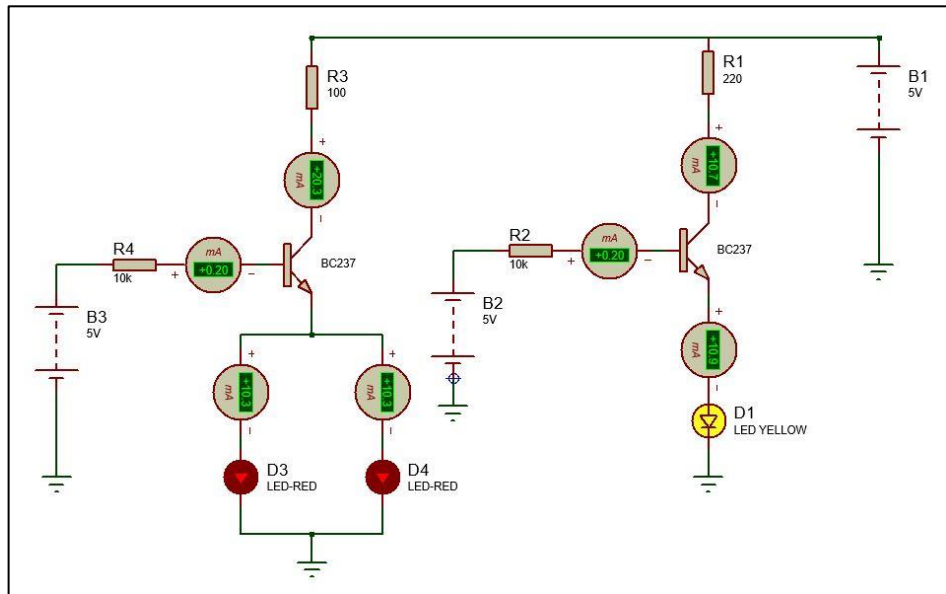
İkinci modülümüz ise L298N motor kontrol modülüne 7805 entegresi kullanılarak tümleşik olarak tasarlanmıştır. Bu modül ise;

- Aracın hareketini sağlayacak dört adet motoru,
- Arduino Mega kartımızı,
- Arka ultrasonik sensörümüzü beslemektedir.

Arduino üzerinden akım akmaması için L298N beslemesi ile Arduino Mega arasına bir adet 1N5408 üç amper geçirgenliğe sahip diyot eklenmiştir. Böylece akım kontrol kartımızın üzerinden geri akması engellenmiştir.

1.4.4. Bağlantı Şemasının Hazırlanması

Projenin sıradaki aşaması, bağlantı şemasının oluşturulmasıdır. İlk olarak LED'lerin Arduino üzerinden beslenmemesi için Proteus ile tasarlanmış yükseltici devreleri oluşturulmuştur. Çalışma akımlarını belirlemek için direnç değerlerini ayarlanmış ve Proteus simülasyonu ile geçen akımlar kontrol edilmiştir. Aşağıda tasarlanmış yükselteç devresi görülmektedir.

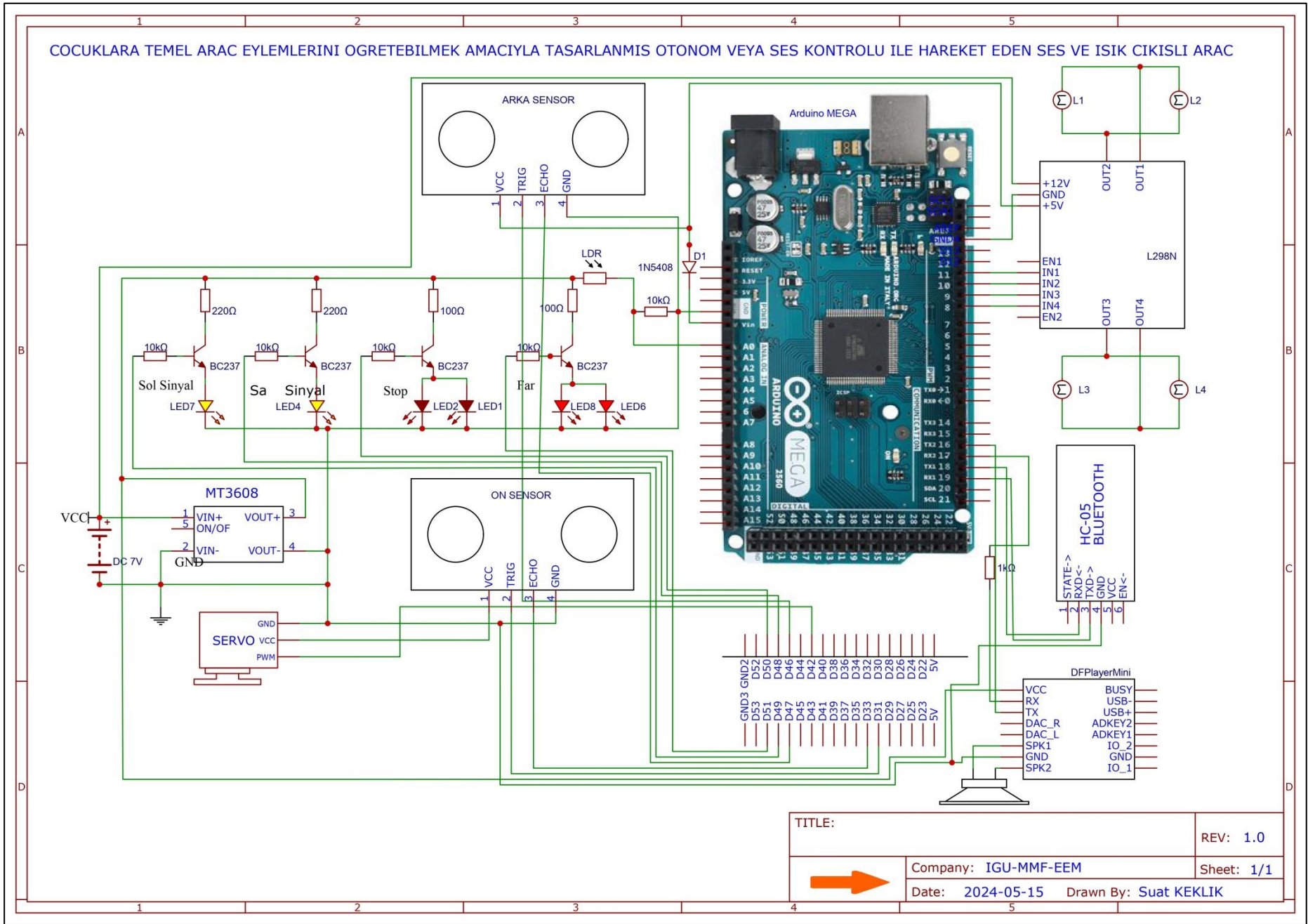


Şekil 17 Yükselteç Devresi

Kaynak: (Yazar Tarafından Oluşturulmuştur).

Şekil 15 de projemizdeki LED lambaların Arduino kartı üzerinden beslenmemesi için tasarlanan devre görülmektedir. Devrede görüldüğü üzere Beyz ucunda 0,20 mA, Emitör çıkışından ise 10,3 ila 10,9 mA akım geçmektedir. Burada görüldüğü üzere her LED için Arduino Mega kartından 10 mA daha az akım çekilmektedir. Arduino Mega kartı, verilere göre en fazla 500 – 800 mA akım geçirilmesine izin verilmektedir. Üreticinin bu verisi dikkate alındığında altı adet LED yaklaşık 60 mA daha az yük demektir. Bu işlem Arduino Mega üzerinde ki yükü %5-10 arasında azaltacaktır.

Sonraki aşamada, EasyEDA programı kullanılarak şekil 16’de görülen devre şeması oluşturulmuştur.



Şekil 18 Devrenin Bağlantı Şeması

Kaynak: (Yazar Tarafından Oluşturulmuştur).

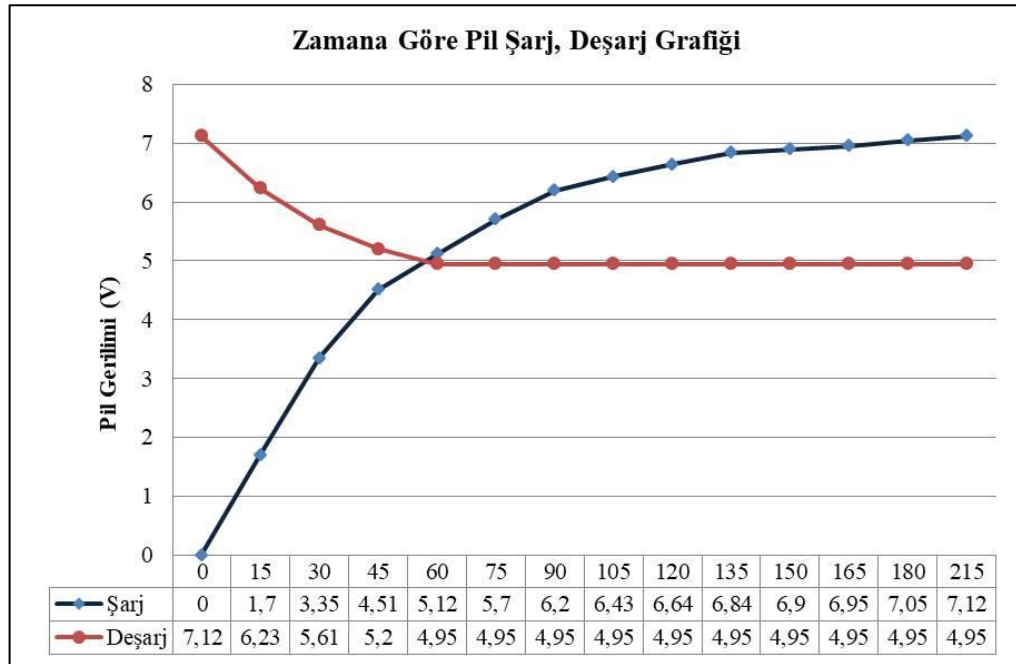
Yukarıdaki resimde de görüldüğü üzere mikro işlemcinin 8, 9, 10 ve 11 pinlerine motor kontrolü, mp3 player 16 ve 17, Bluetooth 18 ve 19, LDR A0, farlar 51, stop lamba 50, sol sinyal 49, sağ sinyal 48, ön ultrasonik sensör 30 ve 31, arka ultrasonik sensör 46 ve 47, servo motoru ise 42 nolu Arduino Mega pinine bağlanmıştır.

Modül ve diğer elektronik ekipmanların beslemeleri yukarıda belirtildiği gibi harici olarak alınmış böylece Arduino kartı üzerinden geçirilen akım azaltılmıştır.

1.4.5. Li-Ion Pilin Projedeki Çalışma Yapısı

Pilimiz kapalı kutu içinde modül olarak temin edilebilmektedir. Pil modülü üzerinde bir adet açma kapatma anahtarı ve bir adet USB Tip C şarj soketi bulunmaktadır. Pilimiz standart telefon şarj aletleri ile doldurulama bilir olarak tasarlanmıştır.

Aracımızın kullanım süresini ölçmek için pilimizin dolum ve boşalma süresi ölçülmesi gerekmektedir. Bu nedenle pilimiz tamamen boşaltılarak şarj edilmiştir. Belli aralıklarla pilimizin gerilimi ölçülmüştür. Pilimizin tamamen dolması halinde akım çekmeyecek ve üzerindeki şarj ışığı sönecektir. Pilimizin dolumu gerçekleştikten sonra aracımıza montajı gerçekleştirilmiştir. Araç otomatik konumda kullanılmıştır. Belirli aralıklarla pilin enerjisi ölçülmüştür. Elde edilen veriler aşağıdaki grafiğe aktarılmıştır.



Grafik 1 Zamana Göre Pil Şarj, Deşarj Grafiği

Kaynak: (Yazar Tarafından Test Ortamında Oluşturulmuştur).

Grafik de görüldüğü üzere pilimiz 0 V doldurulmaya başlanmış ve her 15 dakikada bir pil gerilimi ölçülmüştür. Pilimizde sırasıyla; 1,7 Volt, 3,35 Volt, 4,51 Volt, 5,12 Volt, 5,7 Volt, 6,2 Volt, 6,43 Volt, 6,64 Volt, 6,84 Volt, 6,9 Volt, 6,95 Volt, 7,05 Volt, 7,12 Volt değerleri ölçülmüştür.

Daha sonra pilimiz araca montajı yapılmıştır. Araç otomatik konumda çalıştırılmıştır. Her 15 dakikada pil gerilimi ölçülmüştür. Sırasıyla; 7,12 Volt, 6,23 Volt, 5,61 Volt, 5,2 Volt, 4,95 Volt olarak ölçülmüş. Yaklaşık 60 dakika otomatik konumda aracımız çalıştıktan sonra, gerilim 5 Voltun altına düştüğü için aracımız durmuştur. Bu durum yeni şarjın bu kademedan başlayacağı anlamına gelmektedir. Pilimiz şuan ki gerilim değerine kadar dolam süresi yaklaşık 60 dakikadır. Öyleyse yeni şarj süremiz yaklaşık olarak; $215 \text{ dakika} - 60 \text{ dakika} = 155 \text{ dakika}$ olarak hesaplanmıştır.

1.4.6. L298N Modülünün Projedeki Çalışma Yapısı

Üreticinin katalog bilgisine göre Arduino Mega kartı her çıkış pininden en fazla 40 mA akım verebilmektedir. Ancak araç üzerinde çalışan 4 adet DC motor yaklaşık 600 mA akım çekmektedir. Bu nedenle DC motorlar ve diğer modüller Arduino Mega kartı üzerinden beslenememektedir. Araç üzerindeki 4 adet DC Motor, L298N modülü ile sürülmektedir. Bu modül 2 kanallıdır. Aracımız üzerinde 4 adet DC motor kullanılmıştır. Her kanala 2 adet motor bağlanmıştır. Modülümüz kanal başına 2A akım sağladığı için sorun teşkil etmemektedir. Motor sürücü modülü, aracın istenilen yöne gitmesi veya dönmesi için bir aracıdır. Aracın ileri gitmesi, geri gelmesi, sağ veya sola dönmesi ya da durması yazılım ile komutlara göre belirlenmiştir. Modülümüzün üzerinde motor kontrollerinin haricinde 5 Volt çıkışı için tümleşik 7805 entegresi bulunmaktadır. Devremiz bütün olarak yüksek akım çekmesinden dolayı güç dağılımı ikiye ayrılmıştır. Bu modülümüz;

- Aracın hareketini sağlayacak dört adet motor,
- Arduino Mega kartı,
- Arka ultrasonik sensör beslemektedir.

L298N Motor Kontrol modülünün yazılım bölümünde ilk olarak sisteme tanıtmamız gerekmektedir. Pin bağlantısı şu şekilde sisteme tanıtılmıştır.

```
#define in1 8  
#define in2 9
```



```
#define in3 10  
#define in4 11
```

Tanımlı oluşturulmuştur. Daha sonra setup bölümünde bağlantıları çıkış pini olarak sisteme tanıtılmıştır.

```
pinMode(in1, OUTPUT);  
pinMode(in2, OUTPUT);  
pinMode(in3, OUTPUT);  
pinMode(in4, OUTPUT);
```

Başlangıç durumu, setup bölümünde kapalı olarak tanımlanmıştır.

```
digitalWrite(in1, 0);  
digitalWrite(in2, 0);  
digitalWrite(in3, 0);  
digitalWrite(in4, 0);
```

Böylece motor kontrol çıkışları işlemciye yazılım aracılığıyla tanımlanmıştır. Yazılımın ilgili yerlerinde gerçekleştirilmek istenilen komuta göre motor kontrol çıkışları, aktif ya da pasif edilmiştir. Böylece komuta uyum sağlanmıştır. Örneğin ileri git komutunun algılanması sonucunda in1 ve in3 çıkışları aktif hale getirilmiş, in2 ve in4 çıkışları ise pasif olarak tutulmuştur. Tüm kod dizimi projenin sonunda ek olarak verilmiştir.

1.4.7. HW-131 Modülünün Projedeki Çalışma Yapısı

Bu modülümüz Breadboard montajına uygun olarak tasarlanmıştır. Üzerindeki pinler sayesinde çıkış gerilimini 5 Volt ya da 3,3 Volt olarak alınabilir. Üretici katalog verisine göre cihaz 700 mA çıkış akımı verebilir. Bu modülümüz;

- Bluetooth modülünü,
- Mp3 player modülünü,
- Servo motor,
- Ön ultrasonik sensör,
- LED devreleri,
- LDR devresini beslemektedir.

1.4.8. Ultrasonik Sensörün Projede Çalışması ve Mesafe Ölçümü

Çalışmamda mesafe ölçmek için 2 adet HC-SR04 ultrasonik mesafe sensörü kullanılmıştır. Üretici katalog bilgilerine göre sensör 2 cm'den 400 cm'ye kadar 3mm hassasiyetle ölçüm yapabilmektedir. İki adet sensörden bir tanesi aracın ön tarafına servo motor üzerine monte

edilmiştir. Servo motor kullanılarak aracımızın önüne gelen engeli en kolay şekilde aşması tasarlanmıştır. Servo motorun dönüşü sayesinde aracın sağ ve sol engel mesafeleri ölçülmüş olup araç en uzak olan mesafeye gidecek şekilde programlanmıştır. Diğer ultrasonik sensör aracın arkasına monte edilmiştir. Bu sayede araç geri gelirken arkasındaki engeli de tespit edebilir. Projemizde aşağıdaki formül dizimi ile ses dönüş verisi işlenmiştir.

$V_{\text{hava}} = (331,5 + (0,6 * t))$ m/s formülünü kullanarak sesin havadaki hızını bulabiliriz (Wikipedia, 2019). Bu formülde V_{hava} ses dalgalarının havadaki yayılma hızını, t ortam sıcaklığını, m metre cinsinde mesafeyi ve s ise saniye cinsinden süreyi göstermektedir. Bu formülümüze göre 20 derece hava sıcaklığındaki ses hızımız;

$$V_{\text{hava}} = (331,5 + (0,6 * 20)) \text{ m/s}$$

$$V_{\text{hava}} = 343,5 \text{ m/s olur.}$$

Formülümüz ile bulduğumuz hızımızın birim saniyede kaç metre gittiğini hesaplarız. Ancak yazılımımızda hızımızı mikro saniye de gidilen santimetre birimine göre almamız gerekiyor.

Öyleyse;

$$V_{\text{hava}} = 343,5 \text{ m/s} = 0,3435 \text{ cm}/\mu\text{s}'\text{dir.}$$

Sensörümüzden çıkan ses dalgası gidip geri gelme süresine göre hesaplama yaptığımızdan engel ile olan mesafemizi ölçerken okunan değerimizi ikiye bölmemiz gerekmektedir. Böylece engele ses sinyalizmi engele çarpına kadar geçen süre ile mesafemizi hesaplayabiliriz. Burada hız formülünden yola çıkarsak ve mesafe değerini X harfiyle gösterirsek formülümüz;

$$X = V_{\text{hava}} * t'\text{'dir.}$$

Öyleyse aracımız ile engelimiz arasındaki mesafemizi sesin gidip gelme süresini eklersek;

$$X = (0,3435 \text{ cm}/\mu\text{s}) * (t / 2) \text{ olarak hesaplamalarımızda kullanabiliriz.}$$

Hesaplamalar sonrasında mesafe doğruluğu için test düzeneği kurulmuştur. Sensörden 35 cm uzaklığa koyulan engel ile birer saniye ara ile mesafe ölçülmüştür. Ölçümler sonucunda aşağıdaki tablo oluşturulmuştur.

Tablo 3 Mesafe Ölçüm Test Tablosu

Ölçüm No	Mesafe	Ölçülen	Fark	Hata Oranı
1	35 cm	33,96 cm	-1,04 cm	% 2,97
2	35 cm	33,84 cm	-1,16 cm	% 3,31
3	35 cm	34,05 cm	-0,95 cm	% 2,71
4	35 cm	33,85 cm	-1,15 cm	% 3,29
5	35 cm	34,03 cm	-0,97 cm	% 2,77
6	35 cm	33,97 cm	-1,03 cm	% 2,94
7	35 cm	34,10 cm	-0,9 cm	% 2,57
8	35 cm	34,08 cm	-0,92 cm	% 2,63
9	35 cm	34,12 cm	-0,88 cm	% 2,51
10	35 cm	33,89 cm	-1,11 cm	% 3,17

Kaynak: (Yazar Tarafından Test Ortamında Oluşturulmuştur).

Tablo 3’de görüldüğü üzere engel ile sensör arasındaki mesafe fiziki olarak ve yukarıda belirtilen formül kullanılarak sensör tarafından ölçülüp kontrolör tarafından hesaplanmıştır. Fiziki mesafe cetvel ile ölçülmüştür. Engel mesafesine en yakın değer 34,12 cm, en uzak değer ise 33,84 cm’dir. Hata oranı ise en az %2,51, en yüksek ise %3,97 olarak hesaplanmıştır. Hata oranlarının %4 ün altında kalması nedeniyle kabul edilebilir aralıktadır. Hassas ölçümler için farklı sensörler kullanılabilir.

Yazılım bölümünde sensörlerimizi ilk olarak sisteme tanıtmamız gerekmektedir. Ön bölümdeki sensör için pin bağlantısı;

```
#define on_trigger_pin 31  
#define on_echo_pin 30
```

Tanımlı oluşturulmuştur. Arka bölümdeki sensör için;

```
#define arka_trigger_pin 46  
#define arka_echo_pin 47
```

Oluşturulmuştur. Daha sonra sensör bağlantılarının yaptığı işlemler tanımlanmıştır.

```
pinMode(on_trigger_pin, OUTPUT);  
pinMode(on_echo_pin, INPUT);
```

Tanımlı oluşturulmuştur. Arka bölümdeki sensör için;

```
pinMode(arka_trigger_pin, OUTPUT);  
pinMode(arka_echo_pin, INPUT);
```

Oluşturulmuştur. Her iki sensör içinde setup bölümünde başlangıç durumu kapalı olarak tanımlanmıştır.

```
digitalWrite(on_trigger_pin, 0);  
digitalWrite(on_echo_pin, 0);  
digitalWrite(arka_trigger_pin, 0);  
digitalWrite(arka_echo_pin, 0);
```

Böylece sensörler mikro işlemciye yazılım aracılığıyla tanımlanmıştır. Yazılımın ilgili yerlerinde mesafeyi ölçmek için kullanılan bu sensörler ölçüm kodu ise, triger ve echo giriş çıkışlarını belirli bir süre açılmış ve işlem tamamlanınca tekrar kapatılmıştır. Böylece enerji tasarrufu yapılmaktadır. Tüm kod dizimi projenin sonunda ek olarak verilmiştir.

1.4.9. Sg90 Servo Motorun Projedeki Çalışma Yapısı

Geliştirilen robot üzerinde Sg90 180 derece dönebilen servo motor kullanılmıştır. Bu motor üzerine ultrasonik mesafe sensörü monte edilmiştir ve sensörün daha fazla alanı taraması amaçlanmıştır. Mesafe sensörü bu sayede daha fazla alanı taradığından dolayı gidilmesi gereken yöne daha doğru karar verilebilir. Projede otomatik konumda araç kendisi hareket etmekte ve engel ile karşılaştığı zaman servo motorumuz önce 45° ye döner ve ultrasonik sensör aracın sağındaki engele olan mesafeyi ölçer. Daha sonra 135° döner ve ultrasonik sensör aracın soldaki engele olan mesafesini ölçer. Servo motorunun çalışması için ilk olarak yazılıma kendi kütüphanesini eklemek gereklidir. Bunun için yazılımın en üst kütüphaneler bölümüne,

```
#include <Servo.h>
```

Satırını eklemek yeterlidir. Servo motorumuz diğer modüllerden farklı olarak setup bölümünde bağlantı pini tanıtılmıştır. Tanıtım ve başlangıç pozisyonu olarak kullanılan komutlar aşağıdaki gibidir.

```
sg90.attach(42);  
sg90.write(90);
```

Ölçüm için kullanımda sg90.write komutu ile açısı istenilen konuma getirilir. Servo 90° derecelik açıyı 0,2 saniyede dönebilmektedir. Bu nedenle istenilen yöne gelebilmesi için dönüş komutundan sonra hesaplanan süre kadar beklenmelidir. Bunun için;

```
delay(200);
```

Komutu kullanılabilir.

1.4.10. Mp3 Player Modülünün Projedeki Çalışma Yapısı

Projemizde kullandığımız mp3 modülümüz mikro işlemci ile seri port üzerinden iletişim kurmaktadır. Modülün çalışa bilmesi için bir adet mikro SD karta ihtiyaç duyulmaktadır. Modül, 32GB'a kadar boyutlardaki Micro SD kartlarını destekler. Oynatılacak ses dosyalarının boyutlarının küçük olması nedeniyle SD kart (500MB) kapasiteli olacak şekilde seçilmiştir. Modülü ilk olarak kütüphanesini eklememiz gereklidir.

```
#include "DFRobotDFPlayerMini.h"
```

Kütüphanesi eklenen modülü seri port üzerinden haberleştiği için yazılımda hangi portu kullanacağını belirtmemiz gereklidir. Bu işlem için aşağıdaki kod yazılıma eklenmiştir.

```
#define DFPLAYER_SERIAL Serial2
```

Modülün ismini sistemde oluşturmamız gereklidir. Ayrıca iletişim hızını setup bölümünde tanımlamamız gereklidir. Kodlar aşağıda belirtilmiştir.

```
DFRobotDFPlayerMini player;  
Serial2.begin(9600);
```

Bu komutlara göre modülümüzün adı player, iletişim hızı ise 9600 olarak belirlenmiştir. Ses ayarını setup kısmında tanımlamamız gereklidir. Ses skalası 0-30 arasında tanımlanabilir.

```
player.volume(30);
```

Player tarafından seslendirmesini istediğimiz mp3 dosyasını örnekteki kod ile çalıştırılabilir. Parantez içerisindeki sayı mp3 dosyasının numarasıdır.

```
player.play(18);
```

Ayrıca rastgele olarak birkaç ses dosyası arasından birini çalıştırmak isterseniz randomTrack komutu örnekteki gibi kullanılmalıdır. Böylece ilgili eylemi gerçekleştirirken 21 ila 29 nolu mp3 dosyalarının arasındaki bir tanesini rastgele çalıştıracaktır.

```
int randomTrack = random(21, 29);  
player.play(randomTrack);
```

Projemizde kullanacağımız ses dosyalarının belirlenmesi için yirmi kişi ile anket çalışması yapılmıştır. Bu kişilerin seçiminde şoför olması ve çocuk sahibi olması kriter olarak belirlenmiştir.

Komutlar, ilk olarak seçme - eleme işlemine tabi tutul daha sonra dil yapısı nedeniyle genişletilmiştir. Çalıştırılacak ses dosyaları grup olarak belirlenerek, <https://speechgen.io/> sitesi üzerinden mp3 dosyalarına çevrilmiştir. Ses dosyaları üreticinin belirttiği şekilde numaralandırılmış ve bu numaralar yazılıma işlenmiştir. Bu sayede komutu alan araç işlem basamaklarını sıra ile yapacak ve ilgili mp3 dosyasını çalıştıracaktır. Anket sonuçları hakkında özetlenmiş bilgi aşağıdaki tabloda belirtilmiştir.

Tablo 4 Anket Sonuçlarına Göre Komut ve İşlem Tablosu

<p>1 numaralı soruyu boşluğa ekleyerek cevaplandırınız. “Eğer çocuğunuzun verdiği komuta göre hareket eden bir arabası olsa için nasıl bir komut söylemesini istersiniz”. (Boşluğa “Soru 1” cümlesini yerleştiriniz). 2 numaralı soruyu, 1 numaralı soruya göre cevaplandırınız. “Eğer çocuğunuzun verdiği komuta göre hareket eden bir arabası olsa; soru 1 deki eylemi yaparken aracın nasıl cevap vermesini istersiniz”</p>				
Soru 1	Soru 1 Derlenmiş Komutlar	Genişletilmiş Komutlar	Soru 2 Derlenmiş Cevaplar	İşlemler
İleriye Gitmesi	git, go, ileri, ileri git, ileriye, ileriye git, ilerle, öne doğru git, öne git	Git, Go, İleri, İleri git, İleriye, İleriye git, İlerle, one dogru git, One dogru git, one git, One git, Öne dogru git, Öne git	Tamam. İleriye gidiyorum. (0001.mp3) Önümde engel var duruyorum. (0013.mp3)	0001.mp3 çalışır. Yeni bir komut alana kadar ya da engel ile karşılaşana kadar ilerler. Arka lamba açıksa kapatır. Engel ile karşılaştığında 0013.mp3 çalışır. Durur ve stop lambaları yanar.
Geri Gelmesi	gel, gel abi gel, gel gel, geri, geri gel, geri git, geriye doğru gel, geriye gel, geriye git, geriye doğru gel	Gel, Gel abi gel, Gel gel, Geri, Geri gel, Geri git, Geriye dogru gel, Geriye doğru gel, Geriye gel, Geriye git	Tamam. Geriye geliyorum. (0002.mp3) Arkamda engel var duruyorum. (0012.mp3)	0002.mp3 çalışır. Stop lambalar yanar. Yeni bir komut alana kadar ya da engel ile karşılaşana kadar geri gelir. Engel ile karşılaştığında 0012.mp3 çalışır. Durur ve stop lambaları yanmaya devam eder.
Sağa Dönmesi	Sağ, sağ sağ, sağa doğru dön, sağa dön, sağlı, sağlı dön	sag, Sag, sag sag, Sag sag, saga dogru don, Saga dogru don, saga don, Saga don, sagli, Sagli, sagli don, Sagli don, Sağ, Sağ sağ, Sağa doğru dön, Sağa dön, Sağlı, Sağlı dön	Tamam. Sağa dönüyorum. (0003.mp3)	0003.mp3 çalışır. Sağ sinyal yanar. 0,6 saniye sağa döner.
Sola Dönmesi	Sol, sol dön, sol sol, sola doğru dön, sola dön, sollu dön	Sol, sol don, Sol don, Sol dön, Sol sol, sola dogru don, Sola dogru don, Sola doğru dön, sola don, Sola don, Sola dön, sollu don	Tamam. Sola dönüyorum. (0004.mp3)	0004.mp3 çalışır. Sol sinyal yanar. 0,6 saniye sola döner.
Durması	dur, stop	Dur, Stop	Durdum. (0005.mp3)	Araç durur. 0005.mp3 çalışır.
Etrafında Dönmesi	dön, etrafında dön	don, Don, etrafında don, Etrafında dön	Tamam. Başım dönene kadar etrafımda döneceğim. (0006.mp3) Başım dönüyor duruyorum. (0007.mp3)	0006.mp3 çalışır. Tüm lambalar yanar. 2,5 saniye sağa döner. Durur. 2,5 saniye sola döner. Durur. 0007.mp3 çalışır. Tüm lambalar söner.
Farı Açması	far, far yak, farı aç, farı yak, farları aç, farları yak, lambaları aç, lambaları	Far, Far yak, Farı aç, Farı yak, fari ac, Fari ac, fari yak, Fari yak, Farları aç, Farları yak, farlari ac, Farlari ac, farlari yak, Farlari yak, Lambaları aç, Lambaları yak, lambalari ac, Lambalari ac, lambalari yak, Lambalari yak, Lambayı aç, Lambayı yak, Lambayı ac, lambayı ac,	Farlar açıldı. (0008.mp3)	0008.mp3 çalışır. Ön far yanar.

	yak, lambayı aç, lambayı yak, ön lambayı aç, ön lambayı yak	Lambayı yak, lambayı yak, On lambayı aç, on lambayı ac, On lambayı yak, on lambayı yak, Ön lambayı aç, Ön lambayı yak		
Farı Kapatması	farı kapat, farları kapat, lambaları kapat, lambayı kapat, ön lambayı kapat	Farı kapat, Fari kapat, fari kapat, Farları kapat, Farları kapat, farları kapat, Lambaları kapat, lambaları kapat, Lambaları kapat, Lambayı kapat, lambayı kapat, Lambayı kapat, on lambayı kapat, On lambayı kapat, Ön lambayı kapat	Farlar kapatıldı. (0009.mp3)	0009.mp3 çalışır. Ön far kapanır.
Geri Lambaları Açması	arka lamba aç, arka lamba yak, arka lambaları aç, geri lamba yak, stop aç, stop yak, stopları aç	arka lamba ac, Arka lamba ac, Arka lamba aç, Arka lamba yak, Arka lambaları aç, arka lambaları ac, Arka lambaları ac, arka lambaları yak, Arka lambaları yak, geri lamba ac, Geri lamba ac, Geri lamba yak, geri lambaları ac, Geri lambaları ac, Geri lambaları yak, geri lambaları yak, stop ac, Stop ac, Stop aç, Stop yak, Stopları aç, Stopları ac, stopları ac, Stopları yak, stopları yak	Arka lambalar açıldı. (0010.mp3)	0010.mp3 çalışır. Stop lamba yanar.
Geri Lambaları Kapatması	arka lamba kapat, arka lambaları kapat, geri lamba kapat, geri lambaları kapat, stop kapat	Arka lamba kapat, Arka lambaları kapat, arka lambaları kapat, Arka lambaları kapat, Geri lamba kapat, Geri lambaları kapat, geri lambaları kapat, Geri lambaları kapat, Stop kapat, stopları kapat, Stopları kapat	Arka lambalar kapatıldı. (0011.mp3)	0011.mp3 çalışır. Stop lamba kapanır.
Engeli Kendi Geçmesi	engelden geç, engelden otomatik geç, engeli geç, engeli kendin geç, engeli otomatik geç, oto, otomatik, otomatik geç	engelden gec, Engelden gec, Engelden geç, engelden otomatik gec, Engelden otomatik gec, Engelden otomatik geç, engeli gec, Engeli gec, Engeli geç, engeli kendin gec, Engeli kendin gec, Engeli kendin geç, engeli otomatik gec, Engeli otomatik geç, Oto, Otomatik, otomatik gec, Otomatik gec, Otomatik geç	Otomatik moda geçiyorum. (0019.mp3) Önümde engel var kendim geçeceğim. (0014.mp3) Engelleri tek tek aşıyorum. (0015.mp3) Tamam. Engeli geçiyorum. (0016.mp3) Ticari bekleme yapma. (0017.mp3)	0019.mp3 çalışır. Araç ileriye doğru gider. Eğer önüne engel gelirse durur. 0,1 saniye geri gelir. 0014-0017.mp3 dosyaları rastgele çalışır. Sensör 45°, sağa dönerek aracın sağ taraftan engel mesafesini ölçer. Sensör 135°, sola döner ve aracın engel ile sol mesafesini ölçer. Hangi yön daha uzaksa o tarafa döner. İleriye ilerler. Yeni bir komut alana kadar sürekli çalışır.
Erkek Çocuğa Övgü	Ali, Çiçi, Oğlum	ali, cici, çiçi, Oglum, oğlum, Oğlum	Çakır gözlüm benim. (0021.mp3) Hasta oluyorum bu adama. (0022.mp3) Hasta oluyorum bu adamın saçlarına. (0023.mp3) Very nice. (0024.mp3) Çok karizmatik. (0025.mp3) Çok çok yakışıklı. (0026.mp3) Boyuna posuna kurban olayım. (0027.mp3)	0021-0028.mp3 dosyaları rastgele çalışır.

			Çiçinin gözleri lacı, Yanakları ise sarı, Yoksa yoksa bu çocuk, Doğuştan fenerli mi? (0037.mp3)	
Kız Çocuğa Övgü	Kızım, Prenses	kızım, Kızım, kızım, prenses	Küçük prenses, bugün çok güzel. (0029.mp3) Very nice. (0030.mp3) Pamuk prensesim benim. (0031.mp3) Hanım kızım bugün pek de süslü. (0032.mp3) Benim canım kızım. Sen bir meleksin. (0033.mp3) Kızımın gülüşü, dünyalara bedel. (0034.mp3) Benim kızım, tüm güzellerden güzeldir. (0035.mp3) Elimi tutan en güzel el, bana bakan en güzel göz. (0036.mp3)	0029-0036.mp3 dosyaları rastgele çalışır.
Manuel Kontrol			Kontrol kaptan tarafından gerçekleştirilecek. (0020.mp3)	0020.mp3 çalışır.
Açılış			Merhaba. Yeniden oynamaya hazır mısın? Hadi başlayalım. (0018.mp3)	0018.mp3 çalışır. 0,5 saniye tüm lambalar açılır. 0,2 saniye tüm lambalar kapatılır. 0,5 saniye tüm lambalar açılır. 0,2 saniye tüm lambalar kapatılır. 0,5 saniye tüm lambalar açılır. 0,2 saniye tüm lambalar kapatılır. 0,5 saniye tüm lambalar açılır. Tüm lambalar kapatılır.

Kaynak: (Yazar Tarafından Anket Sonuçlarına Göre Oluşturulmuştur).

Yukarıdaki tabloda görüldüğü üzere iki farklı sonu hazırlanmış ve bu sorular ile on farklı basit komut, bir eğlence komutu, iki kız ve erkek çocuk için farklı övgüler, bir açılış veya karşılama söylevi, bir otomatik komut ve bir manuel komut geçişi belirten söylevi eklenmiştir. On farklı basit komut anketlerden uygun olanları seçilerek toplamda yüz elli üç farklı komut olarak derlenmiştir. Derlenen komutlar, yazılım temel dili ile Türkçe dil yapısının farklı olması nedeniyle üç yüz atmış beş komut olarak yazılıma eklenmiştir. İlgili komuta ve sorulara göre aracın vereceği cevaplar speechgen.io sitesinde mp3 ses dosyalarına çevrilmiş ve numaralandırılmıştır. Numaralandırılan otuz yedi söylem, Arduino IDE programı ile kodlanarak mikro denetleyici kartımıza yüklenmiştir. Ayrıca komutlara uygun olarak aracın hareket etmesi, ışıklarının yanması ya da kapanması için gerekli kod satırları yazılıma eklenmiştir.

1.4.11. HC-05 Bluetooth Modülünün Projedeki Çalışma Yapısı

Bu modül aracımızın komut iletişimini gerçekleştirmek için kullanılmıştır. Cep telefonuna yüklenen Arduino Bluetooth Control programı ile kullanılmaktadır. Telefonumuzun yerleşik Bluetooth ile aracımızın HC-05 modülünü eşleştirilmesi gerekmektedir. Program ile sesli komutumuz HC-05 alıcısına gönderilmekte ve alıcı bu komutu mikro işlemciye göndermektedir. Yazılım kısmında tanımlanan kodlar ile karşılaştırılarak istenilen işlem gerçekleştirilir.



Şekil 19 Arduino Bluetooth Control Programı

Kaynak: (Yazar Tarafından Ekran Görüntüsü Alınarak Oluşturulmuştur).

Bluetooth modülümüz Arduino kütüphanesine ekli olduğundan ek bir kütüphaneye ihtiyaç yoktur. Fakat seri haberleşme sağladığından dolayı aşağıdaki komut ile sisteme tanıtılması gereklidir.

```
Serial1.begin(9600);
```

Arduino Mega kartında seri haberleşme pinleri sabittir. Bu nedenle hangi seri port kullanılacaksa belirtilmesi yeterlidir. Yukarıdaki komutta serial1 portunun 9600 baud hızında iletişimde bulunduğu belirtilmiştir. Bluetooth aracılığı ile gelen bilgiyi okumak için yazılım bölümüne read komutu eklemek gereklidir. Bu komutu Bluetooth modülümüzün bağlı olduğu seri port ile eşleştirmemiz gereklidir.

```
char c = Serial1.read ();
```

Örnekteki kod ile modülden alınan bilgi c değişkenine atanmıştır. Böylece bilgi ayıklama işlemine geçilebilir.

1.4.12. Fiziksel Montaj

Elde edilen bilgiler neticesinde kullanılacak malzemeler temin edilerek montaj aşamasına geçilmiştir. İlk olarak aracın gövdesine motorlar yerleştirilmiş ve kablo bağıyla sabitlenmiştir. Daha sonra pil, HW-131 ve L298N devreleri yerleştirilmiştir. Tüm beslemeyi tek bir anahtardan kontrol etmek için pilin yanına anahtar yerleştirilmiştir. Güç kablolarının montajından sonra aracın üst şasesinin montajı yapılmıştır. Breadboard zemine sabitlenmiş ve Arduino Mega kartı distans kullanılarak yükseltilmiştir. Servo motorun düzeneği aracın ön bölümüne yerleştirilmiştir. Ön sensör servo üstüne takılmıştır. Arka sensör montajı yapıldıktan sonra LED'lerin montajına geçilmiştir. Breadboard üzerine DF Player ve Bluetooth modülleri yerleştirilmiştir. Hoparlör şase altından mıknatıs ile sabitlenmiştir. LED'lerimizin çalışması için yükseltici devreleri Breadboard üzerine yerleştirildikten sonra tüm kablolama işlemi gerçekleştirilmiştir. Aşağıda bulunan Şekil de aracın tüm montajı yapılmış durumu görülmektedir.



Şekil 20 Aracımızın Montajı Tamamlanmış Görüntü

Kaynak: (Fotoğrafın Çekimi Yazar Tarafından Yapılmıştır).

1.4.13. Proje Maliyetinin Hesaplanması

Projenin, Ar-Ge, kurulum ve proje ömrü boyunca tüketilecek değer; paranın şimdiki zaman değer yöntemine göre hesaplanarak aşağıdaki tablo oluşturulmuştur.

Tablo 5 Proje Kurulum ve Ömür Maliyeti Tablosu

	A	B	C = OKEK B (10 Yıl)	D	E	F = A x E	G = D x E	H = C x E
Kullanılan Malzeme	Projede Kullanılan Malzeme Miktarı (Adet)	Cihazın Ekonomik Ömrü (Yıl)	Cihaz Kullanım Ömrü Süresince Malzeme Kaç Kez Değişecek (Adet)	Ar-ge Çalışmalarında Kullanılan Malzeme (Adet)	Birim Maliyet	Malzemenin Şimdiki Zaman Değeri (TL)	Ar-ge Çalışmalarında Kullanılan Malzeme Toplam Değeri (TL)	Cihazın Ekonomik Ömrü Süresince Malzemenin Maliyeti (TL)
Li-ion pil Modülü	1	2	5	-	300	300	-	1.500
HW-131	1	5	2	-	100	100	-	200
L298N	1	5	2	1	70	70	70	140
Araç Şasesi	1	5	2	-	70	70	-	140
Motor	4	5	2	-	75	300	-	150
Arduino Mega	1	5	2	2	850	850	1.700	1.700
Ultrasonik Sensör	2	5	4	-	35	70	-	140
Bluetooth Modülü	1	5	2	2	120	120	240	240
Mp3 Modülü	1	5	2	-	75	75	-	150
Servo Motor	1	5	2	-	30	30	-	60
Diğer Elektronik Ekipman	1	5	2	1	100	100	100	200
Bakım Bedeli	1	1	10	-	200	200	-	2.000
Cihazın İlk Kurulum Maliyeti (TL)						2.155		
	Ar-ge Çalışmalarında Kullanılan Toplam Malzeme Değeri (TL)						2.110	
	Cihazın, Paranın Bugünkü Zaman Değerine Göre 10 Yıllık Maliyeti (TL)							6.620

Kaynak: (Yazar Tarafından Oluşturulmuştur).

Tablo 6’da projemizde kullanılan tüm malzemeleri listelenmiştir. Tablonun A sütünün da projemizde kullanılan parçaların adet sayısı belirtilmiştir. B sütünün da ise Sanayi Bakanlığı sitesinden alınan bilgiler doğrultusunda cihazlarımızın ekonomik ömürleri tabloya girilmiştir. C sütününde, ortak katların en küçüğü (OKEK) yöntemiyle projemizin ekonomik ömür hesaplanmış ve bu ömür içinde kaçar adet cihaz kullanılacağı belirtilmiştir. Tablonun D sütünün da projemizin Ar-Ge çalışmaları sırasında kullanılan ve zarar gören malzemelerin birim/adet miktarları belirtilmiştir. E sütünün da kullanılacak cihazların ilk edinim ücretleri adet bazında belirtilmiştir. F sütununda birimlerin tutarı belirtilmiştir. G sütünün da Ar-Ge çalışmaları sırasında kullanılan malzemelerin ekonomik değeri belirtilmiştir. H sütununda projenin ekonomik ömrü boyunca birim ücreti hesaplanmıştır. Tabloda oluşturulan bilgilere göre projemizin Ar-Ge çalışmaları sırasında kullanılan malzemenin toplam ekonomik değeri 2.110 TL’dir. Cihazımızın ilk kurulum ücreti ise 2.155 TL olarak hesaplanmıştır. Projenin ömrü sürecinde paranın bugünkü değerine göre 10 yıllık maliyeti 6.620 TL olarak hesaplanmıştır.

SONUÇLAR VE ÖNERİLER

Projemizde ilk olarak literatür taraması yapılmış, sonraki adımda ise çalışma prensibi oluşturulmuştur. Daha sonraki aşamada blok diyagramı oluşturulmuş ve bağlantı şeması çizilmiştir. Aracımızın yazılımı Arduino ide programı ile gerçekleştirilmiştir. Şematik görüntüsünü oluşturmak için EasyEDA programı seçilmiş, LED'lerin çalışması için gerekli akım için devre PROTEUS programı üzerinde oluşturulmuş ve simüle edilmiştir. Projemizde kullanılan sesler ve eylemlerin; mesleği şoförlük olan ve ebeveyn olan kişilerden seçilmesi komut, eylem ilişkisini kuvvetlendirmiştir. Seçilen ve çoğaltılan komutlar <https://speechgen.io/> sitesi üzerinden gerçekleştirilmiştir. Aracın zarar görmemesi içinde kullanılan sensörlerin güven sağlaması yapılmış ve %4 'ün altında hata payı saptanmıştır. Bu oran kabul edilebilir olarak değerlendirilmiştir. Projemizde kullanılan pil şarj ve deşarj süreleri test edilmiş ve 60 dakika kullanım için 155 dakika şarj edilmesi gerektiği belirlenmiştir. Projemizin fiziksel tamamlanmış hali eklenmiş böylece son durumu görsel olarak ifade edilmiştir. Son aşamada projemizin paranın bugünkü zaman değerine göre Ar-Ge, maliyet ve ömür hesabı yapılmış tabloda belirtilmiştir.

Ekonomik değer bakımından uygunluk sınırları içindeki bu araç, çocuklarımıza sağlayacağı fayda oldukça fazla olduğu düşünülmektedir. Çocuklarımızı eğlendirirken kendilerine aynı zamanda günümüz koşullarına göre araçların eylemlerini yönlendirme ve eylemlerini öğreteceği planlanmaktadır.

KAYNAKÇA

Addicore, (2024), *HW-131*, Erişim adresi: https://www.addicore.com/v/vspfiles/downloadables/Product%20Downloadables/Breadboard_Power_Module/Addicore%20YwRobot%20Breadboard%20Power%20Supply%20Module%20Schematic.pdf

Alisinanoğlu, F., Akduman, G., Güven, G., (2009). *Çocuklar ve Trafik Kazaları*. Ulaşım ve Trafik Güvenliği Dergisi, C.3, S.1, 16-19.

Arduino.cc, (2019), *Mega2560_Rev3e*, Erişim adresi: https://content.arduino.cc/assets/Pinout-Mega2560rev3_latest.pdf

Bülbül, M.Ş. ve Sözbilir, M. (2017). *Engelsiz STEM eğitimi*, Pegem Akademi, Erişim adresi: <https://avesis.atauni.edu.tr/yayin/7b9ff634-56c9-45f8-a024-7dddc14c4493/engelsiz-stem-egitimi>

Dfrobot, (2016). *Arduino için DFPlayer Mini MP3 Çalar*. Erişim adresi: https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299

Ferreira, F.M., Rúbio, G.D., Brandão, F.H., Mata, A.M., Avellar, N.B., Bonfim, J.P., Tonelli, L.G., Silva, T.X., Dutra, R.M., Petten, A.M., Vimieiro, C. (2020). *Robotic Orthosis For Upper Limb Rehabilitation*. 1. Uluslararası Aktüatör Teknolojisi Elektronik Konferansı: Malzemeler, Cihazlar ve Uygulamalar. Erişim adresi: <https://www.mdpi.com/journal/proceedings>

Hatipoğlu, S., (2002). *Okul Öncesi Çocuklarda Trafik Eğitiminin Gerekliliği*. Uluslararası Trafik ve Yol Güvenliği Kongresi Bildiriler Kitabı, 501-507.

HC-05. (2010). *Bluetooth to Serial Port Module*. Erişim adresi: https://components101.com/sites/default/files/component_datasheet/HC05%20Datasheet.pdf.

H-S. Juang, K-Y. Lum, (2013), *Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board*, 10th IEEE International Conference on Control and Automation (ICCA) Hangzhou

Instruments, T. (2023), *LM1117 800-mA, Low-Dropout Linear Regulator*, Erişim adresi: <https://www.ti.com/product/LM1117>

Maxim Integrated (2019). *+5V-Powered, Multichannel RS-232 Drivers/Receivers*. Erişim adresi: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX220-MAX249.pdf>.

MEB, (2018), *Trafik Genel Eğitim Planı*, Özel Öğretim Kurumları Genel Müdürlüğü, Erişim adresi: https://ookgm.meb.gov.tr/meb_iys_dosyalar/2018_05/02150850_01_Trafik_Eylem_PlanY.pdf

N.Ya'acob , S. F. S Adnan, A. L. Yusof, A. E. Azhar, N. F. Naim, N. M. Nur, A. Mahmon, (2016), *RFID Lab Management System Using Arduino Microcontroller Approach Associate With Webpage*, Journal of Scientific Research and Development

O. Güngör, (2015), *Kaçak Elektrik Kullanımının GSM Aracılığıyla Takibi*, Erişim adresi: <https://dergipark.org.tr/tr/pub/emobd/issue/5509/74718>

Ozdemir Ç., Cakiroglu M., Bayılmış C., Ekiz H., (2004). *Teknolojik Gelişme İçin Eğitimin Önemi ve İnternet Destekli Öğretimin Eğitimdeki Yeri*. The Turkish Online Journal of Educational Technology

Özdemir, S., (2010). *Türkiye’de Örgün Eğitim Sistemi İçerisinde Trafik Eğitiminin Durumu; Avrupa ve Dünya Ülkeleri Işığında Değiştirilmesi*. Yayımlanmamış Yüksek Lisans Tezi, Güvenlik Bilimleri Enstitüsü, Ankara.

Pampal, S., Hatipoğlu, S., Arıkan, E., (1997). *Ulaşım ve Trafik*. Ulaşım-Trafik Kongresi Bildiriler Kitabı, Ankara, 307.

Rene, D. (2013). *Android Mobile Computing Using Samsung Tablets and Smartphones Running, Android 2.3*

S. Hatipoğlu, (2011), *Okul Öncesi Çağı Çocuklarının Trafik Bilgi ve Algıları*, Pegem Eğitim ve Öğretim Dergisi, Erişim adresi: <https://dergipark.org.tr/en/download/article-file/209748>

S. Kim, H. B. Kim, (2010), *High Resolution Mobile Robot Obstacle Detection Using Low Directivity Ultrasonic Sensor Ring, Advanced Intelligent Computing Theories and Applications*. With Aspects of Artificial Intelligence: 6th International Conference on Intelligent Computing, ICIC 2010,

Sahamah, B., (1999), *Experimental Comparison of Skid Steering vs. Explicit Steering for a Wheeled Mobile Robot*, The Robotics Institute Carnegie Mellon University, Eriřim adresi: https://www.ri.cmu.edu/pub_files/pub1/shamah_benjamin_1999_1/shamah_benjamin_1999_1.pdf

Sancak. (2003). *Okulöncesi Eğitim Kurumlarına Devam Eden 6 Yaş Çocuklarına Sayı ve Şekil Kavramlarının Kazandırılmasında Bilgisayar Destekli Eğitim ile Geleneksel Eğitim Yöntemlerinin Karşılaştırılması*. Yüksek Lisans Tezi, Gazi Üniversitesi: Ankara

Soygüder, S., Alli, H.,(2008), *Çok Yönlü Tekerleklerle Sahip Bir Mobil Robotun Tasarımı ve Modelinin Gerçekleştirilmesi*, Celal Bayar Üniversitesi Fen Bilimleri Fakültesi, Eriřim adresi: <https://dergipark.org.tr/tr/download/article-file/45612>

Sönmez, V., (1991). *Okul Öncesi ve Temel Eğitimde Trafik Eğitiminin Önemi*. 1. Trafik Şurası Bildirileri, EGM Trafik Daire Başkanlığı.

Topaç, M.M., Duran, İ., Kuralay, N.S., (2016), *Deney Tasarımı Yaklaşımıyla, Dört Tekerlekten Çekişli Bir Taşıtın Direksiyon Trapezinin Kinematik Optimizasyonu*, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, Eriřim adresi: <https://mmtopac.com/admin/uploads/086e55000153693-5000320588-1-PB%202.pdf>

TUİK, (2023), *Karayolu Trafik Kaza İstatistikleri*, Eriřim adresi: <https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2023-53479>

TUİK, (2024), *Motorlu Kara Taşıtları*, Eriřim adresi: <https://data.tuik.gov.tr/Bulten/Index?p=Motorlu-Kara-Tasitlari-Nisan-2024-53456>

Tuncer, D., Ulu, E.Y., (2023), *Elektrikli Araçların Menzil Sorunu Çözümünde Önemli Bir Tasarım: Elektromekanik Rejeneratif Süspansiyon Sistemi*, Mühendis ve Makine, Eriřim adresi: <https://dergipark.org.tr/en/download/article-file/3653082>

Tütüncü, M., (2001). *Türkiye'de Örgün Eğitim Sistemi İçinde Trafik Eğitiminin Önemi*. Yayımlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

V.A. Akpan &A.S. Eyefia, (2021). *Development of an Enhanced Obstacle Detection System Using Arduino Mega 2560*. Eriřim adresi: https://www.researchgate.net/figure/The-HC-SR04-ultrasonic-sensors_fig2_361815336

W. M. Kadir, R.E. Samin , B.S. Ibrahim,(2012), *Internet controlled robotic arm.*, Proc Eng,

Wikipedia, (2019), *Ses Hızı*, Erişim adresi:
https://tr.wikipedia.org/wiki/Ses_h%C4%B1z%C4%B1

Yelmen, Y., (2010). *Türkiye'de Okul Öncesi Çocukların Trafik Bilgi ve Alışkanlıklarının Belirlenmesi*. Yayımlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

Yücel, E., Gökrem, L., Tuncer, N. (2021). *Okul Öncesi Eğitimde Kullanılabilecek Bir Robotik Materyal Oluşturulması*. Erişim adresi:
<https://dergipark.org.tr/en/download/article-file/1890338>

EKLER

```

#include <Arduino.h>
#include <SoftwareSerial.h>
#include "DFRobotDFPlayerMini.h"
#include <Servo.h>
#include "NewPing.h"

#define in1 8
#define in2 9
#define in3 10
#define in4 11
#define on_far 51
#define stop_lamba 50
#define sol_sinyal 49
#define sag_sinyal 48
#define DFPLAYER_SERIAL Serial2
#define on_trigger_pin 31
#define on_echo_pin 30
#define arka_trigger_pin 46
#define arka_echo_pin 47

DFRobotDFPlayerMini player;
String komut;
Servo sg90;
int pos = 0;
float engelmesafesi = 45;

int sure_on = 0, sure_arka = 0,
sure_sag = 0, sure_sol = 0,
mesafe_on = 0, mesafe_arka = 0,
mesafe_sag = 0, mesafe_sol = 0,
run = 0, otomatik = 0, yon = 0, ses
= 0;
unsigned long simdikizaman,
eskizaman = 0;

void setup()
{
  sg90.attach(42);
  sg90.write(90);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  digitalWrite(in1, 0);

```

```

  digitalWrite(in2, 0);
  digitalWrite(in3, 0);
  digitalWrite(in4, 0);

  pinMode(on_trigger_pin, OUTPUT);
  pinMode(on_echo_pin, INPUT);
  pinMode(arka_trigger_pin, OUTPUT);
  pinMode(arka_echo_pin, INPUT);
  digitalWrite(on_trigger_pin, 0);
  digitalWrite(on_echo_pin, 0);
  digitalWrite(arka_trigger_pin, 0);
  digitalWrite(arka_echo_pin, 0);

  pinMode(on_far, OUTPUT);
  pinMode(stop_lamba, OUTPUT);
  pinMode(sol_sinyal, OUTPUT);
  pinMode(sag_sinyal, OUTPUT);
  digitalWrite(on_far, 0);
  digitalWrite(stop_lamba, 0);
  digitalWrite(sol_sinyal, 0);
  digitalWrite(sag_sinyal, 0);

  Serial.begin(9600);
  delay(10);
  Serial1.begin(9600);
  delay(10);
  Serial2.begin(9600);
  delay(10);

  if (player.begin(Serial2))
  {
    player.volume(30);
    delay(100);
    player.play(18);
    digitalWrite(on_far, HIGH);
    digitalWrite(stop_lamba, HIGH);
    digitalWrite(sag_sinyal, HIGH);
    digitalWrite(sol_sinyal, HIGH);
    delay(500);
    digitalWrite(on_far, LOW);
    digitalWrite(stop_lamba, LOW);
    digitalWrite(sag_sinyal, LOW);
    digitalWrite(sol_sinyal, LOW);
    delay(200);

```

```

digitalWrite(on_far, HIGH);
digitalWrite(stop_lamba, HIGH);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(sol_sinyal, HIGH);
delay(500);
digitalWrite(on_far, LOW);
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, LOW);
delay(200);
digitalWrite(on_far, HIGH);
digitalWrite(stop_lamba, HIGH);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(sol_sinyal, HIGH);
delay(500);
digitalWrite(on_far, LOW);
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, LOW);
delay(200);
digitalWrite(on_far, HIGH);
digitalWrite(stop_lamba, HIGH);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(sol_sinyal, HIGH);
delay(500);
digitalWrite(on_far, LOW);
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, LOW);
delay(200);
digitalWrite(on_far, HIGH);
digitalWrite(stop_lamba, HIGH);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(sol_sinyal, HIGH);
delay(500);
digitalWrite(on_far, LOW);
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, LOW);
}
else
{
delay(10);
}
}

void loop()

```

```

{
simdikizaman = millis();
if (simdikizaman - eskizaman >=
1000)
{
eskizaman = millis();
}
int isik = analogRead(A0);
delay(50);
if (isik > 800 && otomatik==2 &&
(run==1 || run==0))
{
digitalWrite(on_far, LOW);
}
if (isik <= 800 && otomatik==2 &&
(run==2 || run==0))
{
digitalWrite(on_far, HIGH);
}
if (isik > 800 && otomatik==2 &&
(run==2 || run==0))
{
digitalWrite(on_far, LOW);
}
}
delay(100);
digitalWrite(on_trigger_pin, 1);
delayMicroseconds(10);
digitalWrite(on_trigger_pin, 0);
delayMicroseconds(10);
sure_on = pulseIn(on_echo_pin, 1);
delayMicroseconds(20);
digitalWrite(on_echo_pin, 0);
mesafe_on = (sure_on / 2) / 29.7;
if (mesafe_on < 0)
{
mesafe_on = 200;
}
if (mesafe_on > 200)
{
mesafe_on = 200;
}
}
delay(100);
digitalWrite(arka_trigger_pin, 1);
delayMicroseconds(10);
digitalWrite(arka_trigger_pin, 0);
delayMicroseconds(10);
}

```

```

sure_arka = pulseIn(arka_echo_pin,
1);
delayMicroseconds(20);
digitalWrite(arka_echo_pin, 0);
mesafe_arka = (sure_arka / 2) /
29.7;
if (mesafe_arka < 0)
{
mesafe_arka = 200;
}
if (mesafe_arka > 200)
{
mesafe_arka = 200;
}
delay(50);
if(mesafe_on < engelmesafesi &&
otomatik==1 && run==1 && yon ==1)
{
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(stop_lamba, 1);
delay(100);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(100);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
player.play(13);
delay(2500);
otomatik=0;
}
delay(10);
if (mesafe_arka < engelmesafesi &&
otomatik==1 && run==1 && yon ==2)
{
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(stop_lamba, 1);
delay(100);

```

```

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(200);
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
player.play(12);
delay(2500);
otomatik=0;
}
delay(10);
if(mesafe_on > engelmesafesi &&
otomatik==2 && run==2)
{
delay(10);
digitalWrite(stop_lamba, 0);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
delay(10);
if (mesafe_on < engelmesafesi &&
otomatik==2 && run==2)
{
delay(100);
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(stop_lamba, HIGH);
delay(100);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(200);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(30);
int randomTrack = random(14, 18);
player.play(randomTrack);

```

```

delay(3500);
sg90.write(45);
delay(200);
digitalWrite(on_trigger_pin, 1);
delayMicroseconds(5);
digitalWrite(on_trigger_pin, 0);
delayMicroseconds(5);
sure_sag = pulseIn(on_echo_pin,
1);
delayMicroseconds(10);
digitalWrite(on_echo_pin, 0);
mesafe_sag = ( sure_sag / 2) /
29.7;
if (mesafe_sag < 0)
{
mesafe_sag = 200;
}
if (mesafe_sag > 200)
{
mesafe_sag = 200;
}

delay(10);
sg90.write(135);
delay(200);
digitalWrite(on_trigger_pin, 1);
delayMicroseconds(5);
digitalWrite(on_trigger_pin, 0);
delayMicroseconds(5);
sure_sol = pulseIn(on_echo_pin,
1);
delayMicroseconds(10);
digitalWrite(on_echo_pin, 0);
mesafe_sol = ( sure_sol / 2) /
29.7;

if ( mesafe_sol < 0)
{
mesafe_sol = 200;
}
if ( mesafe_sol > 200)
{
mesafe_sol = 200;
}

delay(10);
sg90.write(90);

```

```

delay(200);
if(mesafe_sag < mesafe_sol)
{
delay(50);
player.play(4);
delay(200);
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, HIGH);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(600);
digitalWrite(sol_sinyal, LOW);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
else if(mesafe_sag >= mesafe_sol)
{
delay(10);
delay(50);
player.play(3);
delay(200);
digitalWrite(stop_lamba, LOW);
digitalWrite(sol_sinyal, LOW);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(600);
digitalWrite(sag_sinyal, LOW);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
}
olcum3();
}

void olcum3()
{
while (Serial1.available())

```

```

{
delay(100);
char c = Serial1.read();
if (c == '#')
{
break;
}
komut += c;
}
if (komut.length() > 0)
{
delay(10);
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
otomatik=1;
if (komut == "git" || komut ==
"go" || komut == "ileri git" ||
komut == "ileriye git" || komut ==
"öne git" || komut == "öne dogru
git" || komut == "one git" ||
komut == "one dogru git" || komut
== "ilerle" || komut == "ileri" ||
komut == "ileriye")
{
ileriye_git();
}
if (komut == "Git" || komut ==
"Go" || komut == "İleri git" ||
komut == "İleriye git" || komut ==
"Öne git" || komut == "Öne dogru
git" || komut == "One git" || komut
== "One dogru git" || komut ==
"İlerle" || komut == "İleri" ||
komut == "İleriye")
{
ileriye_git();
}
else if (komut == "gel abi gel"
|| komut == "gel gel" || komut ==
"gel" || komut == "geri" || komut
== "bak" || komut == "geri gel" ||
komut == "geri git" || komut ==
"geriye git" || komut == "geriye
gel" || komut == "geriye dogru

```

```

gel" || komut == "geriye dogru
gel")
{
geriye_gel();
}
else if (komut == "Gel abi gel"
|| komut == "Gel gel" || komut ==
"Gel" || komut == "Geri" || komut
== "Bak" || komut == "Geri gel" ||
komut == "Geri git" || komut ==
"Geriye git" || komut == "Geriye
gel" || komut == "Geriye dogru
gel" || komut == "Geriye dogru
gel")
{
geriye_gel();
}
else if (komut == "sağ sağ"
|| komut == "sağa dogru dön"
|| komut == "sağlı" || komut ==
"sağ" || komut == "sağa dön" ||
komut == "sağlı dön")
{
saga_don();
}
else if (komut == "sag sag"
|| komut == "saga dogru don"
|| komut == "sagli" || komut ==
"sag" || komut == "saga don" ||
komut == "sagli don")
{
saga_don();
}
else if (komut == "Sağ sağ"
|| komut == "Sağa dogru dön"
|| komut == "Sağlı" || komut ==
"Sağ" || komut == "Sağa dön" ||
komut == "Sağlı dön")
{
saga_don();
}
else if (komut == "Sag sag"
|| komut == "Saga dogru don"
|| komut == "Sagli" || komut ==
"Sag" || komut == "Saga don" ||
komut == "Sagli don")
{

```

```

saga_don();
}
else if (komut == "sol" || komut
== "sol sol" || komut == "sola
dogru don " || komut == "sola don"
|| komut == "sol don" || komut ==
"sollu don" || komut == "sola
dogru dön " || komut == "sola dön"
|| komut == "sol dön" || komut ==
"sollu dön")
{
sola_don();
}
else if (komut == "Sol" || komut
== "Sol sol" || komut == "Sola
dogru don " || komut == "Sola don"
|| komut == "Sol don" || komut ==
"Sollu don" || komut == "Sola
dogru dön " || komut == "Sola dön"
|| komut == "Sol dön" || komut ==
"Sollu dön")
{
sola_don();
}
else if (komut == "dur" || komut ==
"Dur" || komut == "Stop" || komut
== "stop")
{
dur();
}
else if (komut == "don" || komut
== "Don" || komut == "Dön" ||
komut == "dön" || komut ==
"etrafında don" || komut ==
"Etrafında don" || komut ==
"etrafında dön" || komut ==
"Etrafında dön")
{
etrafında_don();
}
else if (komut == "far" || komut
== "farları aç" || komut ==
"farlari ac" || komut == "fari ac"
|| komut == "farı aç" || komut ==
"lambayi ac" || komut == "lambayı
aç" || komut == "lambalari ac"
|| komut == "lambaları aç" || komut

```

```

== "on lambayi ac" || komut == "ön
lambayı aç")
{
far_ac();
}
else if (komut == "Far" || komut
== "Farları aç" || komut ==
"Farlari ac" || komut == "Fari ac"
|| komut == "Farı aç" || komut ==
"lambayi ac" || komut == "lambayı
aç" || komut == "Lambalari ac"
|| komut == "Lambaları aç" || komut
== "On lambayi ac" || komut == "Ön
lambayı aç")
{
far_ac();
}
else if (komut == "farları yak" ||
komut == "farlari yak" || komut ==
"fari yak" || komut == "farı yak"
|| komut == "lambayi yak" || komut
== "lambayı yak" || komut ==
"lambalari yak" || komut ==
"lambaları yak" || komut == "on
lambayi yak" || komut == "ön
lambayı yak")
{
far_ac();
}
else if (komut == "Farları yak" ||
komut == "Farlari yak" || komut ==
"Fari yak" || komut == "Farı yak"
|| komut == "Lambayi yak" || komut
== "Lambayı yak" || komut ==
"Lambalari yak" || komut ==
"Lambaları yak" || komut == "On
lambayi yak" || komut == "Ön
lambayı yak")
{
far_ac();
}
else if (komut == "fark" || komut
== "farlari kapat" || komut ==
"farları kapat" || komut == "fari
kapat" || komut == "farı kapat" ||
komut == "lambayi kapat" || komut
== "lambayı kapat" || komut ==

```



```

"lambalari kapat" || komut ==
"lambaları kapat" || komut == "on
lambayı kapat" || komut == "ön
lambayı kapat")
{
far_kapat();
}
else if (komut == "Fark" || komut
== "Farlari kapat" || komut ==
"Farları kapat" || komut == "Fari
kapat" || komut == "Farı kapat" ||
komut == "Lambayı kapat" || komut
== "Lambaları kapat" || komut ==
"Lambaları kapat" || komut == "On
lambayı kapat" || komut == "Ön
lambayı kapat")
{
far_kapat();
}
else if (komut == "s" || komut ==
"stop ac" || komut == "stop aç" ||
komut == "stoplari ac" || komut ==
"stopları aç" || komut == "arka
lamba ac" || komut == "arka lamba
aç" || komut == "arka lambalari
ac" || komut == "arka lambaları
aç" || komut == "geri lambalari ac"
|| komut == "geri lamba ac") {
stop_lamba_ac();
}
else if (komut == "stop yak" ||
komut == "stoplari yak" || komut
== "arka lamba yak" || komut ==
"arka lambalari yak" || komut ==
"geri lambalari yak" || komut ==
"geri lamba yak") {
stop_lamba_ac();
}
else if (komut == "s" || komut ==
"Stop ac" || komut == "Sstop aç"
|| komut == "Stoplari ac" || komut
== "Stopları aç" || komut == "Arka
lamba ac" || komut == "Arka lamba
aç" || komut == "Arka lambalari
ac" || komut == "Arka lambaları

```

```

aç" || komut == "Geri lambalari ac"
|| komut == "Geri lamba ac")
{
stop_lamba_ac();
}
else if (komut == "Stop yak" ||
komut == "Stoplari yak" || komut
== "Arka lamba yak" || komut ==
"Arka lambalari yak" || komut ==
"Geri lambalari yak" || komut ==
"Geri lamba yak")
{
stop_lamba_ac();
}
else if (komut == "sk" || komut ==
"stop kapat" || komut == "stoplari
kapat" || komut == "arka lamba
kapat" || komut == "arka lambalari
kapat" || komut == "arka lambaları
kapat" || komut == "geri lambalari
kapat" || komut == "geri lambaları
kapat" || komut == "geri lamba
kapat")
{
stop_lamba_kapat();
}
else if (komut == "Sk" || komut ==
"Stop kapat" || komut == "Stoplari
kapat" || komut == "Arka lamba
kapat" || komut == "Arka lambalari
kapat" || komut == "Arka lambaları
kapat" || komut == "Geri lambalari
kapat" || komut == "Geri lambaları
kapat" || komut == "Geri lamba
kapat")
{
stop_lamba_kapat();
}
else if (komut == "radar" || komut
== "Radar")
{
radar_test();
}
else if (komut == "oğlum" || komut
== "oglum" || komut == "cici" ||
komut == "çiçi" || komut ==
"ali" || komut == "Ali")

```

```

{
  rasgele_calistir();
}
else if (komut == "kızım" || komut
== "kizim" || komut == "Kızım" ||
komut == "kizim" || komut ==
"prensese" || komut == "Prensese")
{
  rasgele_calistir2();
}
else if (komut== "engeli gec"
||komut== "engeli geç" || komut==
"engelden gec" ||komut== "engelden
geç" || komut== "engeli kendin
geç" || komut== "engeli kendin
geç" || komut== "oto" || komut==
"otomatik gec" ||komut== "otomatik
geç" || komut== "engelden otomatik
geç" || komut== "engelden otomatik
geç" || komut== "engeli otomatik
geç" || komut== "engeli otomatik
geç" || komut== "Engeli gec"
||komut== "Engeli geç" || komut==
"Engelden gec" ||komut== "Engelden
geç" || komut== "Engeli kendin
geç" || komut== "Engeli kendin
geç" || komut== "Oto" || komut==
"Otomatik gec" ||komut== "Otomatik
geç" || komut== "Engelden otomatik
geç" || komut== "Engelden otomatik
geç" || komut== "Engeli otomatik
geç" || komut== "Engeli otomatik
geç")
{
  digitalWrite(stop_lamba, LOW);
  otomatik=2;
  delay(10);
}
}
manuel_oto();
komut = "";
}
void manuel_oto()
{
  delay(10);
  if(otomatik==2 && (run==1 ||
run==0))

```

```

{
  digitalWrite(in2, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  player.play(19);
  delay(3000);
  run=2;
}
delay(10);
if(otomatik==1 && (run==0 ||
run==2))
{
  digitalWrite(in2, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  player.play(20);
  delay(3000);
  run=1;
}
}

void ileriye_git()
{
  manuel_oto();
  yon=1;
  digitalWrite(stop_lamba, LOW);
  player.play(1);
  delay(2000);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}
void geriye_gel()//
{
  manuel_oto();
  yon=2;
  digitalWrite(stop_lamba, HIGH);
  player.play(2);
  delay(2000);
  digitalWrite(sag_sinyal, LOW);
  digitalWrite(sol_sinyal, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);

```

```

digitalWrite(in4, HIGH);
}
void saga_don()
{
manuel_oto();
digitalWrite(stop_lamba, LOW);
digitalWrite(sol_sinyal, LOW);
digitalWrite(sag_sinyal, HIGH);
player.play(3);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(600);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(sag_sinyal, LOW);
}

void sola_don()
{
manuel_oto();
digitalWrite(stop_lamba, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, HIGH);
player.play(4);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(600);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(sol_sinyal, LOW);
}

void dur()
{
manuel_oto();
digitalWrite(stop_lamba, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in1, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);

```

```

player.play(5);
}

void etrafinda_don()
{
manuel_oto();
player.play(6);
digitalWrite(sag_sinyal, HIGH);
digitalWrite(sol_sinyal, HIGH);
digitalWrite(stop_lamba, HIGH);
digitalWrite(on_far, HIGH);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);

delay(2500);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(250);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(2500);
player.play(7);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
digitalWrite(sag_sinyal, LOW);
digitalWrite(sol_sinyal, LOW);
digitalWrite(stop_lamba, LOW);
digitalWrite(on_far, LOW);
}

void far_ac()
{
manuel_oto();
player.play(8);
digitalWrite(stop_lamba, LOW);
digitalWrite(on_far, HIGH);
}

void far_kapat()
{

```

```

manuel_oto();
player.play(9);
digitalWrite(on_far, LOW);
}

void stop_lamba_ac()
{
manuel_oto();
player.play(10);
digitalWrite(on_far, LOW);
digitalWrite(stop_lamba, HIGH)
}

void stop_lamba_kapat()
{
manuel_oto();
player.play(11);
digitalWrite(stop_lamba, LOW);
}

void radar_test()
{
manuel_oto();
sg90.write(0);
delay(50);
for(int i=0;i<=180;i=i+15)
{
sg90.write(i);
delay(30);
}
for(int i=180;i>=0;i=i-15)
{
sg90.write(i);
delay(30);
}
}

void rasgele_calistir()
{
manuel_oto();
int randomTrack = random(21, 29);
player.play(randomTrack);
}

void rasgele_calistir2()
{
manuel_oto();
int randomTrack = random(29, 36);
player.play(randomTrack);
}

```

```

void tanimsiz()
{
manuel_oto();
player.play(37);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(4000);
}

```