

**KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
BÖLÜMÜ**

BİTİRME ÇALIŞMASI

**NESNELERİN İNTERNETİ TABANLI WEB ARAYÜZÜ
KONTROLLÜ MQTT İLE HABERLEŞEN AKILLI
İTFAİYE ARACI**

Halil İbrahim ÇETİN

210208020

Doç. Dr. Oğuzhan KARAHAN

KOCAELİ, 2025

ÖNSÖZ ve TEŞEKKÜR

Teknolojinin hızlı gelişimi, her sektörde olduğu gibi yangınla mücadele, arama-kurtarma ve acil müdahale gibi kritik alanlarda da önemli değişimlere neden olmuştur. Özellikle Nesnelerin İnterneti (IoT), yapay zeka ve uzaktan erişim teknolojilerinin yaygınlaşması sayesinde, daha akıllı ve daha güvenli sistemlerin geliştirilmesi artık mümkün hale gelmiştir. Bu değişim, hem kullanıcı deneyimini geliştirmekte hem de insan faktörünü azaltarak operasyonel verimliliği ve güvenliği artırmaktadır.

Son yıllarda uzaktan kontrol edilebilen araçlara olan ilginin artması, bu alanda yapılan projelerin çeşitlenmesine ve fonksiyonel olarak zenginleşmesine yol açmıştır. Artık yalnızca robotik sistemlerin değil, aynı zamanda geleneksel araçların da uzaktan yönetilebildiği; yönlendirme, sistem durumu takibi, canlı video izleme ve sensör verisi okuma gibi işlevlerin bir web arayüzü üzerinden yapılabildiği projeler ön plana çıkmaktadır.

Bu proje kapsamında, bir itfaiye aracının temel işlevlerini uzaktan kontrol edebilecek bir sistem tasarlanmış ve uygulanabilirliği test edilebilmiştir. Kullanıcı, internet bağlantısı olan herhangi bir cihazdan, araca ait web arayüzü üzerinden canlı video görüntüsünü izleyebilmekte, aracın yönünü belirleyebilmekte, siren ve su pompası gibi işlevleri çalıştırabilmekte ve aynı zamanda araç üzerindeki sıcaklık, nem gibi çevresel verileri anlık olarak takip edebilmektedir. Tüm bu sistem, MQTT protokolü ile iletişim sağlayan bir altyapı üzerine kurulmuştur. Sistemin tam manası ile tamamlanabilmesi için fiziksel bir araca ihtiyaç duymaktadır ve proje finalinde tamamen internetten kontrol edilebilen bir itfaiye aracı fiziksel olarak üretilecektir.

Bu konuda çalışma yapmama olanak sağlayan, çalışmanın her aşamasında ilgi ve desteğini eksik etmeyen değerli hocam Doç. Dr. Oğuzhan Karahan'a teşekkür ederim.

Haziran 2025, KOCAELİ

Halil İbrahim ÇETİN

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLOLAR DİZİNİ	v
SİMGELER DİZİNİ VE KISALTMALAR	vi
IOT TABANLI WEB ARAYÜZ KONTROLLÜ AKILLI İTFAİYE ARACI	vii
GİRİŞ	1
1. LİTERATÜR ARAŞTIRMASI	3
1.1 Giriş	3
1.2 Akıllı Araç Sistemleri ve Acil Müdahalede IoT Kullanımı.....	3
1.3 Web Arayüzü ve IoT Entegrasyonu	7
1.4 IoT Temelli Araç Takip ve Kontrol Uygulamaları.....	9
2. KULLANILAN TEKNOLOJİLER	11
2.1 Giriş	11
2.2 Yazılım	12
2.2.1 Python	12
2.2.2 HTML	13
2.2.3 PiServo.....	14
2.2.4 gpiozero.....	14
2.2.5 paho-mqtt	16
2.2.6 Owncast.....	17
2.3 Donanım	18
2.3.1 Raspberry Pi 5.....	18
2.3.2 SG90 Servo Motor	20
2.3.3 DC Motor.....	21
2.3.4 L298N DC Motor Sürücüsü.....	23
2.3.5 12 DC Su Pompası.....	24
2.3.6 ADS1115 ADC (Analog Dijital Dönüştürücü).....	25
2.3.7 Şamandıra Modülü.....	26
2.3.8 LED Diyot.....	27
2.3.9 PLA 3D Yazıcı Filamenti	28
2.3.10 Flame Sensör.....	29
2.3.11 Buzzer	31
2.4 Haberleşme ve Veri İletimi.....	32
2.4.1 MQTT(Message Queuing Telemetry Transport).....	32
2.4.2 VPS(Virtual Private Server)	33
2.4.3 Owncast.....	34
3. NESNELERİN İNTERNETİ (IOT).....	36
3.1 IoT Bağlantı Türleri.....	36
3.2 IoT Ağlarının Türleri	38
3.3 IoT ve Veri tabanları.....	39
4. AKILLI İTFAİYE ARACI TASARIMI ve gerçekleşmesi	39

4.1	Giriş	40
4.2	Uzaktan Kontrollü İtfaiye Araç Donanımsal Tasarımı.....	40
4.3	İtfaiye Aracı Elektronik Tasarımı	43
4.4	İtfaiye Aracı Yazılım Tasarımı	44
4.5	Web Arayüzü Yazılım Tasarımı	45
4.6	DeneySEL SonuÇlar	49
4.6.1	Web Arayüz Erişilebilirlik Testi.....	49
4.6.2	MQTT Broker Bağlantı Testi.....	49
4.6.3	Veri Gönderim Testi	50
4.6.4	Veri Alım Testi	52
4.6.5	Motor Fonksiyonellik Testi.....	52
4.6.6	Su Tahliye ve Depo Sızdırmazlık Testi	53
4.6.7	Şamandıra Ölçüm ve Algoritma Testi	54
4.6.8	Flame Sensör Veri Alımı ve Aktarımı Testi	55
4.6.9	Servo Kontrol Testi.....	55
4.6.10	Kamera Yayın Testi	56
5.	SONUÇLAR VE ÖNERİLER	58
	KAYNAKLAR	60

ŞEKİLLER DİZİNİ

Şekil 1.1 : Örnek araç modül ve çalışma şeması.....	4
Şekil 1.2 : Smart Firefighting.....	5
Şekil 1.3 : Tokyo Robotics Firefighter Bot.....	6
Şekil 1.4 : IoT tabanlı otomobil kontrol ve takip uygulaması	7
Şekil 1.5 : Bulut tabanlı veri yönetimi	9
Şekil 2.1 : HTML CSS ve Javascript çalışma mantığını anlatan bir görsel.....	14
Şekil 2.2 : Raspberry Pi 5.....	20
Şekil 2.3 : SG90 servo motor	21
Şekil 2.4 : DC motor	23
Şekil 2.5 : L298N DC motor sürücüsü.....	24
Şekil 2.6 : 12V DC su pompası.....	25
Şekil 2.7 : ADS1115 ADC	26
Şekil 2.8 : Şamandıra modülü	27
Şekil 2.9 : LED diyot	28
Şekil 2.10: PLA filament	29
Şekil 2.11: Flame sensör	31
Şekil 2.12: Piezo buzzer	32
Şekil 2.13: MQTT protokolü çalışma prensibi	33
Şekil 2.14: VPS'e ait örnek görsel	34
Şekil 2.15: Kamera verisi iletimine ait sistem mimarisi	35
Şekil 4.1 : İtfaiye aracı şasi tasarımı	40
Şekil 4.2 : Elektronik bağlantı şeması.....	41
Şekil 4.3 : İtfaiye aracı dış gövde tasarımları.....	43
Şekil 4.4 : 5V DC motor	43
Şekil 4.5 : 12V DC su pompası.....	44
Şekil 4.6 : Sistemin çalışma prensibine ait akış şeması	45
Şekil 4.7 : Sistem arayüz prototipi	46
Şekil 4.8 : Arayüz görüntüsü.....	46
Şekil 4.9 : MQTT bağlantısı ve gerekli topiclere subscribe işlemi.....	47
Şekil 4.10: Web arayüz erişilebilirlik testi sonucu.....	49
Şekil 4.11: Web arayüzü üzerinden MQTT bağlantı testi ve konsol kayıtları.....	50
Şekil 4.12: MQTT broker client bağlantı testi	50
Şekil 4.13: Web arayüzü veri gönderim testi konsol kayıtları	51
Şekil 4.14: Veri gönderim testi MQTT broker veri alım sonuçları.....	51
Şekil 4.15: Sistemin başarılı veri gönderimine ait geri bildirim kısmı	52
Şekil 4.16: Sistemin başarılı veri aldığına dair geri bildirim terminal çıktısı	52
Şekil 4.17: Motor çalışma algoritması	53
Şekil 4.18: Su deposu 3B tasarımı	53
Şekil 4.19: Su tahliye pompası kontrol algoritması	54
Şekil 4.20: Şamandıra ile su deposu doluluk oranı algılama algoritması	54
Şekil 4.21: Flame sensör alev yönü algılama ve arayüze gönderme algoritması.....	55
Şekil 4.22: Servo çalışma algoritması.....	56
Şekil 4.23: Servo fonksiyonellik testi	56
Şekil 4.24: Owncast çalışma testi	56
Şekil 4.25: Yayına ait durum istatistikleri.....	57
Şekil 4.26: OBS giden kare durum testi.....	57
Şekil 4.27: Kamera yayınına sağlayan , yerel ağ'a kamera verisini gönderen kod.....	57
Şekil 4.28: İtfaiye aracı nihai görünümü.....	59

TABLolar DİZİNİ

Tablo 4.1 : 3B parça tasarımları.....	42
Tablo 4.2 : Arayüze ait fonksiyon tablosu	47
Tablo 4.3 : Topic/İşlev tablosu.....	48
Tablo 5.1 : İtfaiye aracı fiziksel görünüşleri.....	59

SİMGELER DİZİNİ VE KISALTMALAR

IoT	: Internet of Thing (Nesnelerin İnterneti)
PWM	: Pulse Width Modulation
API	: Application Program Interface
LED	: Light Emitting Diode
I/O	: Input/Output (Giriş/Çıkış)
IDE	: Integrated Development Enviroment (Entegre Geliştirme Ortamı)
PLA	: Polilaktik Asit
Wi-Fi	: Wireless Fidelity
UART	: Universal Asynchronous Receiver/Transmitter
GPIO	:General Input/Output Pin
RC	:Remote Control (Uzaktan Kontrol)
GPS	:Global Positioning System
GSM	:Global System for Mobile Communications
DC	:Direct Current
JSON	: Javascript Object Nation
RAM	: Random Acces Memory
WAN	: Wide Area Network
LAN	: Local Area Network
ARM	: Acorn RISC Machine
MQTT	: Message Queuing Telemetry Transport
HTML	: HyperText Markup Language
CSS	: Cascading Style Sheets
MIT	: Massachusetts Institute of Technology
HTTPS	: HyperText Transfer Protocol Secure

IOT TABANLI WEB ARAYÜZ KONTROLLÜ AKILLI İTFAİYE ARACI

Halil İbrahim ÇETİN

Anahtar Kelimeler: Akıllı Araç , IoT, Uzaktan Kontrol , Otomobil

Özet: Akıllı cihazların günlük yaşamdaki etkisi her geçen gün artarken, bu teknolojilerin farklı sektörlerdeki yansımaları da dikkat çekici hale gelmiştir. Özellikle acil durum müdahale sistemlerinde, teknolojinin etkin kullanımı sayesinde hem güvenlik hem de verimlilik artışı sağlanmaktadır. Bu bağlamda geliştirilen web tabanlı kontrol sistemleri, uzak erişimle cihazların yönetilmesine imkan tanımakta ve insan müdahalesini minimize ederek daha hızlı aksiyon alınmasına olanak sunmaktadır.

Bu çalışmada, IoT tabanlı, uzaktan kontrol edilebilen bir itfaiye aracı prototipi tasarlanmıştır. Geliştirilen sistemde; bir web arayüzü üzerinden itfaiye aracı yönlendirilebilmekte, siren ve su pompası gibi ekipmanlar kontrol edilebilmekte ve araç üzerindeki sensörlerden alınan veriler anlık olarak kullanıcıya sunulmaktadır. Sistem; yön bilgisi, sıcaklık, gaz seviyesi gibi çeşitli sensör verilerini MQTT protokolü ile bir VPS (sanal sunucu) üzerinde kurulu olan MQTT broker’a iletmekte, kullanıcı ise bu verilere gerçek zamanlı olarak ulaşabilmektedir.

Ayrıca araç üzerine entegre edilen bir kamera aracılığıyla canlı görüntü yayını sağlanmış, bu yayın web arayüzünde gösterilerek operatörün aracı hem görsel hem de veri tabanlı şekilde kontrol etmesi mümkün hale getirilmiştir.

Proje kapsamında; elektronik, mekanik ve yazılım alanlarında disiplinler arası birçok teknoloji birlikte kullanılmıştır. Web arayüzü HTML, CSS ve JavaScript ile geliştirilmiş; MQTT iletişimi için gerekli sunucu altyapısı VPS üzerinde yapılandırılmıştır. Böylece kullanıcı, herhangi bir cihazdan sadece bir web tarayıcısı aracılığıyla itfaiye aracını uzaktan kontrol edebilmekte ve sistemin işleyişini gerçek zamanlı olarak takip edebilmektedir.

Bu proje, gerçek bir acil durum müdahale aracının temel işlevlerini simüle etmekte ve IoT tabanlı akıllı sistemlerin yangınla mücadele gibi kritik alanlarda nasıl kullanılabileceğine dair önemli bir örnek teşkil etmektedir.

GİRİŞ

Günümüzde IoT tabanlı sistemlerin kullanımı birçok alanda olduğu gibi afet yönetimi ve kamu hizmetlerinde de yaygınlaşmaktadır. Bu bağlamda geliştirilen akıllı sistemler, acil müdahale araçlarının daha verimli ve uzaktan yönetilebilir hale gelmesini sağlamaktadır. Özellikle yangın gibi anlık müdahale gerektiren durumlarda, teknoloji destekli çözümler zamanla yarışta büyük avantajlar sağlamaktadır.

Web arayüzü üzerinden kontrol edilebilen itfaiye aracı projesi, bu vizyon doğrultusunda hayata geçirilmiştir. Projenin temel amacı, uzaktan erişim sağlanabilen, yön kontrolü yapılabilen, su pompası ve siren gibi sistemleri yönetilebilen, aynı zamanda canlı kamera görüntüsü ve sensör verilerini kullanıcıya anlık olarak iletebilen bir prototip geliştirmektir.

Proje kapsamında ilk olarak, yangın durumlarında karşılaşılabilecek ihtiyaçlar analiz edilmiş ve bu ihtiyaçlara çözüm sunabilecek donanım ve yazılım bileşenleri araştırılmıştır. Literatür taraması sonucunda, benzer sistemlerin dünyada ve ülkemizde nasıl geliştirildiği incelenmiş, bu sistemlerin avantaj ve dezavantajları değerlendirilerek projeye yön verilmiştir.

İkinci aşamada, kullanılan komponentler ve teknolojiler belirlenmiş; yön kontrolü, su pompası ve siren sistemlerinin hangi modüllerle kontrol edileceği planlanmıştır. Kullanılan sensörlerin neden tercih edildiği, bu sensörlerin proje için sunduğu avantajlar ve ekonomik anlamda sağladığı katkılar analiz edilmiştir. Web arayüzünün geliştirilmesinde kullanılan yazılım dilleri (HTML, CSS, JavaScript) ve MQTT protokolü gibi iletişim teknolojilerinin seçim sebepleri detaylıca açıklanmıştır.

Üçüncü bölümde ise projenin Nesnelerin İnterneti (IoT) ile olan entegrasyonu detaylandırılmıştır. MQTT protokolü ile araç ile kullanıcı arasında düşük gecikmeli, hafif veri iletişimi sağlanması hedeflenmiştir. Aynı zamanda sensör verilerinin web

arayüzü üzerinden görselleştirilmesi ve analiz edilmesi için uygun veri tabanı ve bağlantı yöntemleri araştırılmış, uygun çözümler projeye entegre edilmiştir.

Son olarak projenin şu ana kadar tamamlanan kısımları aktarılmış; canlı video akışı, yön kontrolü, siren ve pompa sistemi gibi temel işlevlerin başarıyla gerçekleştirildiği gösterilmiştir. Web tabanlı kontrol paneli üzerinden yapılan testlerde alınan sonuçlar ve sistemin performans değerlendirmeleri raporlanmıştır. Projenin sonraki aşamalarında ise yapay zeka destekli yangın tespiti, GPS tabanlı konum bilgisi paylaşımı gibi ileri düzey geliştirmeler hedeflenmektedir.

Bu çalışma, afet müdahale araçlarında teknoloji kullanımına dikkat çekmekte ve bu alanda geliştirilebilecek daha büyük sistemler için önemli bir temel sunmaktadır.

1. LİTERATÜR ARAŞTIRMASI

1.1 Giriş

Günümüzde yangınla mücadelede teknolojik sistemlerin entegrasyonu giderek önem kazanmaktadır. Özellikle IoT (Nesnelerin İnterneti) tabanlı uzaktan kontrol sistemleri, itfaiye araçlarının daha etkin ve güvenli bir şekilde kullanılmasına olanak tanımaktadır. Bu projelerde web tabanlı kontrol sistemleri ile IoT teknolojileri bir araya getirilerek uzaktan erişim, gerçek zamanlı izleme ve veri toplama gibi imkânlar sağlanmaktadır.

IoT destekli itfaiye araçları, sensörlerden gelen verileri anlık olarak analiz ederek sürücü veya görevli personele bilgi sunabilir; ayrıca yangın söndürme sistemleri, kamera akışı, siren gibi kontroller web arayüzü üzerinden sağlanabilir. Bu teknolojiler, özellikle afet anlarında müdahale süresini kısaltarak can ve mal kaybını en aza indirme potansiyeline sahiptir.

Web tabanlı IoT sistemlerinin getirdiği temel avantajlar şunlardır:

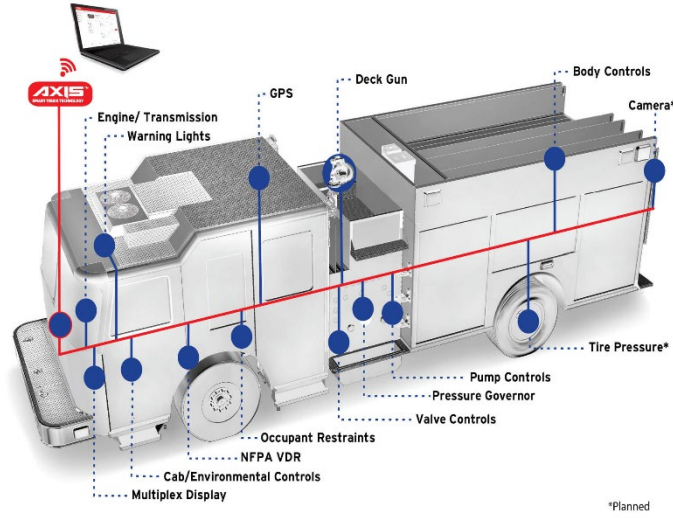
- 1) Gerçek zamanlı veri takibi: Araç konumu, sıcaklık, nem ve gaz sensörü gibi veriler anlık olarak izlenebilir.
- 2) Uzaktan kontrol: Su püskürtme sistemi, siren, uyarı lambaları gibi donanımlar uzaktan kontrol edilebilir.
- 3) Güvenli bağlantı: HTTPS üzerinden güvenli bir şekilde veri aktarımı sağlanarak sistem güvenliği artırılır.

1.2 Akıllı Araç Sistemleri ve Acil Müdahalede IoT Kullanımı

IoT teknolojilerinin acil durum araçlarına entegrasyonu, sadece otomobil sektörünü değil, aynı zamanda afet yönetim ve müdahale alanlarını da etkilemiştir. Özellikle

itfaiye araçlarında bu teknolojiler sayesinde sistemler çok daha etkili, kontrollü ve verimli hale gelmektedir.

Bu konudaki yapılan bu denli entegre bir sistemin web arayüzünden kontrol edilmesi şuan çok yaygın değildir ve proje ile birebir benzer yapıda bir uygulama henüz yoktur. Fakat her geçen gün gelişen IoT dünyasında benzer uygulamalar kullanılmaktadır. Bu gibi sistemlerin örnek araç modül ve çalışma şeması Şekil 1.1’de verilmiştir



Şekil 1.1 Örnek araç modül ve çalışma şeması

Bu uygulamaların bazılarından örnek vermek gerekirse şöyle bir liste ortaya çıkarabiliriz.

Smart Firefighting Initiatives – ABD : ABD’de bazı şehirlerde "Smart Firefighting" adı altında itfaiye sistemlerinin dijitalleşmesi üzerine çalışmalar yapılıyor. Bu sistemlerde:

- Yangın bölgelerine gönderilen araçların içi sensörlerle donatılıyor.
- Araçlar GPS ile takip ediliyor.
- Araç içindeki sistemler merkezden yönetiliyor.

Ancak, bu sistemler daha çok büyük ölçekli, profesyonel itfaiye sistemlerine yönelik; robotik kontrol ya da web arayüzü üzerinden doğrudan mini araç yönlendirme gibi özellikler yok [2]. Sistemin sitesinden alınan örnek görsele Şekil 1.2’den ulaşılabilir.



Şekil 1.2 Smart Firefighting

Tokyo Robotics Firefighter Bot : Tokyo Robotics Firefighting Robotları, özellikle tehlikeli ve insanların müdahale edemeyeceği yangın alanlarında kullanılmak üzere geliştirilen insansız araçlardır. Bu robotlar, yangın anında bina içlerine veya çökmek üzere olan yapılara girerek hem yangın söndürme hem de keşif işlemlerini gerçekleştirebilir. Tokyo İtfaiyesi ve çeşitli Japon araştırma enstitülerinin ortak çalışmalarıyla geliştirilen bu robotlar; yüksek ısıya dayanıklı yapıları, güçlü su püskürtme sistemleri ve otomatik yön bulma kabiliyetleriyle dikkat çeker. Robotlar, üzerlerindeki termal kameralar ve çevresel sensörler sayesinde alev kaynağını tespit edip doğrudan müdahalede bulunabilirler [1].

Bu sistemler genellikle ağır ve büyük yapıda olup otonom ya da yarı otonom olarak çalışırlar. Operatör tarafından uzaktan kumanda edilebilseler de çoğu görevlerini önceden belirlenen algoritmalara göre otomatik gerçekleştirebilirler. Web arayüzü üzerinden doğrudan yönlendirme, MQTT gibi IoT tabanlı kontrol altyapıları bu sistemlerde yaygın değildir. Daha çok endüstriyel düzeyde kablosuz iletişim sistemleri veya özel kontrol üniteleri kullanılır. Tokyo Robotics’in geliştirdiği bu sistemler, profesyonel itfaiye teşkilatlarında kullanım için uygundur ve gerçek zamanlı veri iletimi, termal görüntüleme ve koordinat paylaşımı gibi gelişmiş özellikler sunar.

Bu sisteme ait örnek görüntü Şekil 1.3'te görüldüğü gibidir.



Şekil 1.3 Tokyo Robotics Firefighter Bot

MIT - Web Controlled Robot Car : MIT'nin Web Kontrollü Robot Araç Projeleri, genellikle lisans ve lisansüstü düzeyde robotik, gömülü sistemler ve IoT derslerinde eğitim amacıyla geliştirilen çalışmalardır.

Bu projelerde öğrenciler, mini ölçekli robot araçları web tabanlı arayüzler üzerinden yönlendirebilir, gerçek zamanlı kamera görüntüsünü izleyebilir ve mesafe, sıcaklık gibi temel sensör verilerini MQTT veya HTTP protokolleri üzerinden çekebilirler. Araçlar genellikle Raspberry Pi, Arduino veya benzeri mikrodenetleyici platformları kullanılarak tasarlanır. Sistem, öğrencilerin hem yazılım hem donanım konularını deneyimlemeleri açısından önemli bir öğrenim aracıdır [3].

Ancak bu sistemler daha çok laboratuvar ortamına özgü ve temel kontrol mantığını öğretmeye yönelik tasarlandığı için, entegre bir uygulama senaryosu ya da özel bir görev teması içermez.

Örneğin, yangın söndürme, afet müdahalesi gibi gerçek dünya senaryolarına doğrudan uygulanabilir bir kapsamı bulunmamaktadır. İtfaiye teması, su püskürtme sistemleri, coğrafi sınırlama (geofencing) veya sahada kullanılabilecek güçlü bir mobil platform gibi özellikler bu projelerde yer almaz.

Bu nedenle, eğitim amaçlı olmalarının dışında, tam entegre IoT sistemlerine örnek teşkil etseler de gerçek zamanlı müdahale gerektiren profesyonel senaryolar için

yetersizdirler. Bu projelere benzer bir projeyi ben de MUHTAS-III için gerçekleştirdim ve bu gibi projelere örnek bir görsel olarak kendi projem Şekil 1.4'te verilmiştir.



Şekil 1.4 IoT tabanlı otomobil kontrol ve takip uygulaması [6]

1.3 Web Arayüzü ve IoT Entegrasyonu

Web tabanlı kontrol sistemleri, IoT cihazlarıyla etkileşim kurmak için genellikle MQTT (Message Queuing Telemetry Transport) ve HTTP gibi hafif protokolleri kullanır. Bu sistemlerde web sitesi, kullanıcı arayüzü olarak görev yapar ve cihazlara komut gönderebilir veya sensör verilerini gerçek zamanlı olarak alabilir [5].

MQTT Protokolü: Araç üzerindeki gömülü sistemler (örneğin Raspberry Pi), sensör verilerini MQTT broker'a yayınlar. Web arayüzü bu broker'a abone olarak sıcaklık, gaz seviyesi, yön bilgisi veya motor durumu gibi verileri anlık olarak kullanıcıya sunar. Ayrıca kullanıcı, web sayfasındaki butonlar aracılığıyla araca yön, siren veya su pompası komutu gönderebilir. Bu sayede doğrudan bir mobil uygulamaya ihtiyaç duymadan, cihaz herhangi bir tarayıcıdan yönetilebilir [6].

HTTP/REST API: Daha geniş kapsamlı veri sorgulama işlemlerinde HTTP protokolü kullanılır. Örneğin, geçmiş sürüş verileri veya olay kayıtları, web arayüzü üzerinden REST API'ler aracılığıyla alınabilir. Bu sistemler genellikle JavaScript ve WebSocket desteği ile geliştirilir.

WebSocket ve Gerçek Zamanlı Görüntü Aktarımı: WebSocket teknolojisi sayesinde, kullanıcının web sayfası ile araç arasında gerçek zamanlı iki yönlü iletişim sağlanır.

Bu sayede, yön komutları gecikmeden araca iletilir ve kamera görüntüsü aynı anda tarayıcıda izlenebilir[7].

Canlı Kamera Yayını: Araç üzerine entegre edilen kamera sistemi, MJPEG veya RTMP gibi protokollerle bir akış servisine gönderilir. Web sayfası bu yayını iframe veya canvas üzerinden göstererek canlı yayın sağlar. Böylece operatör, aracın konumunu sadece haritadan değil, görsel olarak da takip edebilir.

cihazlarla senkronize olarak kullanıcı deneyimini artırır. Örneğin, telefonunuzun yakındaki araç tarafından otomatik algılanması ve kontağa gerek kalmadan aracın çalıştırılması mümkündür.

Web tabanlı IoT sistemleri genellikle merkezi bir bulut altyapısı üzerinde çalışır. Bu yapı, verinin sadece gerçek zamanlı değil, aynı zamanda geçmişe dönük olarak da analiz edilmesini mümkün kılar.

Veri Depolama ve Erişim: Aracın konumu, yönü, sistem durumu ve sensör verileri gibi bilgiler veri tabanında saklanır. Kullanıcı, web sitesine giriş yaparak geçmiş görevler, araç performansı ve rota kayıtlarını görüntüleyebilir. Ayrıca sistem, bakım hatırlatmaları, arıza uyarıları ve görev geçmişi gibi fonksiyonları da sağlayabilir.

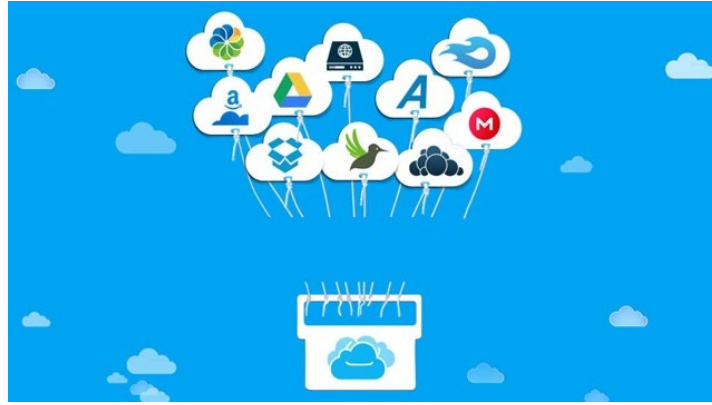
Güvenlik ve Kimlik Doğrulama: Kullanıcı oturumları için JWT veya OAuth tabanlı kimlik doğrulama mekanizmaları kullanılır. Web arayüzü ile yapılan işlemler log'lanır ve güvenli veri iletimi için HTTPS sertifikaları (örneğin Let's Encrypt ile) devreye alınır. Böylece hem kullanıcı hem de araç sistemi güvenli şekilde haberleşmiş olur.

Veri Depolama ve İşleme: Araçtan gelen konum, hız, yakıt durumu, sensör bilgileri gibi veriler, bulut sunucularında depolanır ve işlenir. Bu sayede kullanıcılar, geçmiş sürüş bilgilerini, yakıt tüketim raporlarını veya performans analizlerini detaylı bir şekilde inceleyebilir.

Gerçek Zamanlı Veri Erişimi: Kullanıcılar, internet bağlantısı olan her yerden araçlarının durumunu kontrol edebilir. Örneğin: Araç konumunu haritada görmek , hız

limitinin aşılması durumunda uyarı almak aracın bakım zamanı geldiğinde veya herhangi bir arıza tespit edildiğinde bildirim almak.

Güvenlik ve Yedekleme: Bulut sunucuları, verilerin güvenliğini sağlamak için şifreleme ve kimlik doğrulama gibi yöntemler kullanır. Ayrıca, veri kaybı durumunda yedekleme özellikleri ile kullanıcıların verilerine tekrar erişmesi sağlanır[8].Bulut tabanlı veri yönetimini anlatan görsel Şekil 1.5.'de verilmiştir.



Şekil 1.5 Bulut tabanlı veri yönetimi

1.4 IoT Temelli Araç Takip ve Kontrol Uygulamaları

IoT (Nesnelerin İnterneti) teknolojisi, araç takip ve kontrol sistemlerinde devrim yaratmıştır. Bu sistemler, sensörler, GPS, bulut bilişim ve mobil uygulamaların entegrasyonu sayesinde araç sahiplerine gelişmiş güvenlik, konfor ve verimlilik sağlar.

Güvenlik: IoT temelli araç kontrol sistemleri, araç güvenliğini artırmak için tasarlanmıştır. Bu sistemler, araç konumunu gerçek zamanlı olarak takip eder ve güvenlik tehditlerine karşı kullanıcıyı bilgilendirir.

Anlık Konum Takibi: GPS ve GSM teknolojisi sayesinde araç konumu sürekli olarak izlenir. Araç çalınması durumunda, kullanıcılar aracın yerini mobil uygulama üzerinden anında tespit edebilir. Bu, aracın hızlı bir şekilde geri alınmasına olanak tanır.

Hırsızlık Girişimi Uyarıları: Araçta bulunan sensörler, izinsiz giriş veya araç hareketini algıladığında hemen uyarı gönderir. Örneğin, kapıların zorlanması veya motorun izinsiz çalıştırılması durumunda kullanıcı, akıllı telefonuna bildirim alır.

Uzaktan Müdahale: Hırsızlık veya acil durumlarda, kullanıcılar mobil uygulama üzerinden aracı kilitleyebilir veya motoru devre dışı bırakabilir. Bu özellik, aracın güvende tutulmasına yardımcı olur.

Konfor : IoT temelli sistemler, araç sahiplerine günlük yaşamda büyük kolaylıklar ve konfor sağlar.

Uzaktan Araç Çalıştırma: Kullanıcılar, soğuk havalarda araçlarını uzaktan çalıştırarak motorun ve iç ortamın ısınmasını sağlayabilir. Bu, araç içine binış sırasında maksimum konfor sunar[9].

Klima ve Isıtma Kontrolü: IoT tabanlı sistemler, araç klimasını veya ısıtmasını uzaktan kontrol etme imkanı sunar. Bu özellik, yazın sıcak günlerinde aracın serinlemesini veya kışın önceden ısınmasını sağlar.

Akıllı Kilit Sistemleri: Araç kapıları, mobil uygulama üzerinden uzaktan kilitlenip açılabilir. Kullanıcı, aracının güvenliğini sağlamak için anahtara gerek duymadan bu işlemi gerçekleştirebilir [10].

Kişiselleştirme: Kullanıcılar, IoT uygulamaları aracılığıyla araç içi sistemleri kendi tercihleri doğrultusunda ayarlayabilir (örneğin, sürüş modları, koltuk pozisyonu, müzik çalma listeleri).

2. KULLANILAN TEKNOLOJİLER

2.1 Giriş

Bu proje, bir itfaiye aracının web tabanlı bir kullanıcı arayüzü üzerinden uzaktan kontrol edilebilmesini hedefleyen, MQTT protokolü ile çalışan bir IoT (Nesnelerin İnterneti) uygulamasıdır. Proje kapsamında geliştirilen sistemde yön kontrolü, siren ve su pompası gibi temel bileşenler web üzerinden kumanda edilebilmekte; ayrıca araç üzerindeki kamera yardımıyla canlı video akışı da sağlanmaktadır. Projenin merkezinde, Linux tabanlı Raspberry Pi 5 model bilgisayar yer almakta ve tüm donanım kontrolü bu platform üzerinden yürütülmektedir.

Sistem, istemci ve sunucu mimarisiyle çalışmaktadır. Kullanıcı tarafında, HTML, CSS ve JavaScript ile geliştirilmiş bir web arayüzü bulunmaktadır. Bu arayüz, kullanıcının aracın yönünü değiştirebilmesi, sireni açıp kapatabilmesi ve pompayı çalıştırabilmesi gibi işlevleri gerçekleştirebildiği bir kontrol paneli sunar. Web arayüzü ile sistem arasındaki veri alışverişi MQTT (Message Queuing Telemetry Transport) protokolü üzerinden gerçekleşmektedir. Bu sayede düşük bant genişliği ve gecikmeyle gerçek zamanlı kontrol sağlanmaktadır. MQTT broker olarak kullanıcıya ait bir VPS (Virtual Private Server) üzerinde kurulu Mosquitto sunucusu kullanılmaktadır. Bu yapı, sistemin dış servislere bağımlılığını azaltarak veri güvenliğini ve sistem sürekliliğini artırmaktadır.

Raspberry Pi 5 üzerinde çalışan Python tabanlı bir MQTT istemcisi, gelen komutları algılayarak ilgili GPIO pinleri üzerinden fiziksel bileşenlere iletir. Örneğin; aracın ileri hareket etmesi için motor sürücüsüne komut gönderilirken, siren çalıştırılmak istendiğinde ilgili çıkış aktif hale getirilir. Araca entegre edilen bir kamera modülü ile elde edilen görüntü, kullanıcıya canlı video yayını olarak ulaştırılır. Bu sayede kullanıcı, kontrol ettiği aracın bulunduğu ortamı gerçek zamanlı olarak görebilir.

Bu sistemin en önemli avantajlarından biri platformdan bağımsız olarak çalışabilmesidir. Web tabanlı arayüz sayesinde kullanıcı, herhangi bir işletim sistemi veya cihaz kısıtlaması olmaksızın sadece bir internet tarayıcısı ile araca erişebilir.

Ayrıca MQTT protokolü sayesinde sistemin ölçeklenebilirliği artırılmış ve ileride sensör verilerinin izlenmesi, olay temelli bildirimlerin sağlanması gibi özelliklerin kolaylıkla entegre edilebilmesinin önü açılmıştır.

Sonuç olarak, bu proje; IoT, MQTT, web teknolojileri ve gömülü sistemlerin birleşimiyle oluşturulmuş, gerçek zamanlı ve güvenli bir uzaktan kontrol altyapısı sunmaktadır. Özellikle afet durumlarında, uzaktan yönlendirilebilir itfaiye araçlarının geliştirilmesi için önemli bir prototip teşkil etmektedir.

2.2 Yazılım

2.2.1 Python

Python, genel amaçlı, yüksek seviye bir programlama dilidir ve son yıllarda büyük bir popülarite kazanmıştır. Guido van Rossum tarafından 1991 yılında geliştirilen Python, basit ve okunabilir bir syntax yapısına sahiptir, bu da onu özellikle öğrenme süreci için ideal bir dil haline getirir. Python, açık kaynaklı bir dil olup, geniş bir kullanıcı topluluğu tarafından desteklenmektedir.

Python'un popülaritesi, çeşitli alanlarda kullanılabilmesiyle büyük ölçüde artmıştır. Web geliştirme, veri analizi, yapay zeka, bilimsel hesaplama, otomasyon, oyun geliştirme ve daha birçok alanda Python yaygın olarak kullanılmaktadır. Python'un geniş kütüphane desteği, kullanıcıların birçok farklı görevi kolayca gerçekleştirebilmesini sağlar.

Özellikle Python'un, hızlı bir şekilde prototip geliştirme yapmaya olanak sağlayan dinamik tip sistemine sahip olması, yazılım geliştirme süreçlerini hızlandırır. Ayrıca, Python'un çeşitli platformlarda çalışabilme yeteneği, taşınabilirlik açısından da büyük bir avantaj sağlar.

Python ile arayüz kodlama da oldukça popülerdir. Python'un çeşitli arayüz kütüphaneleri (örneğin, Tkinter, PyQt, wxPython) sayesinde kullanıcılar masaüstü uygulamaları için grafiksel kullanıcı arayüzleri (GUI) geliştirebilirler. Bu arayüzler,

kullanıcıların uygulamalarıyla etkileşimde bulunmalarını sağlar ve Python'u masaüstü uygulamaları geliştirmek için ideal bir dil haline getirir.

2.2.2 HTML

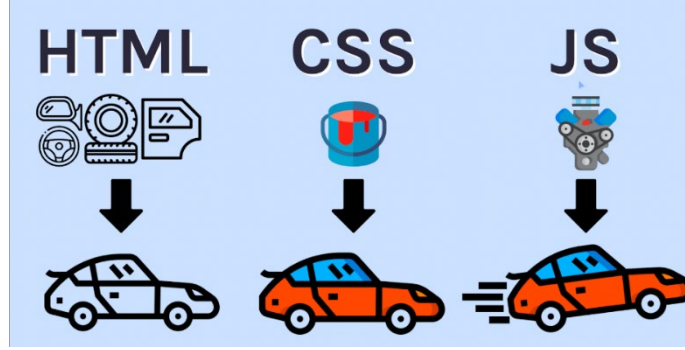
Web geliştirme teknolojilerinin temelini HTML (HyperText Markup Language), CSS (Cascading Style Sheets) ve JavaScript oluşturur. Bu üç teknoloji birlikte çalışarak kullanıcı etkileşimli, görsel olarak zengin ve dinamik web sayfalarının oluşturulmasını sağlar.

HTML, web sayfalarının iskelet yapısını oluşturan işaretleme dilidir. Sayfadaki başlıklar, paragraflar, butonlar, görseller ve bağlantılar gibi içerik öğeleri HTML aracılığıyla tanımlanır. HTML'in modern sürümü olan HTML5, multimedya öğeleri (video, ses vb.) ve yerel depolama gibi özelliklerle daha güçlü ve esnek bir yapı sunmaktadır.

CSS, web sayfalarının tasarım ve stil kısmını belirleyen dil olarak kullanılır. Yazı tipi, renk, arka plan, hizalama ve sayfa düzeni gibi görsel unsurlar CSS ile özelleştirilir. CSS3 sürümü ile birlikte animasyonlar, geçiş efektleri ve responsive (duyarlı) tasarım teknikleri yaygınlaşmış; bu da kullanıcı deneyimini artıran modern arayüzlerin önünü açmıştır.

JavaScript ise web sayfalarına etkileşim kazandıran, istemci taraflı bir programlama dilidir. Sayfa üzerindeki öğelerin kullanıcı girdilerine göre değişmesini sağlayarak dinamik içerik üretir. Örneğin, butona tıklanmasıyla bir motorun çalıştırılması, sensör verisinin güncellenmesi veya bir uyarı kutusunun gösterilmesi gibi işlemler JavaScript ile gerçekleştirilir. JavaScript'in modern sürümleri (ES6 ve sonrası), nesne yönelimli programlama, modüler yapı ve asenkron iletişim gibi güçlü özellikler sunar.[5]

Web tabanlı kontrol sistemlerinde HTML, CSS ve JavaScript'in birlikte kullanımı, platformdan bağımsız, erişimi kolay ve gerçek zamanlı kontrol sağlayan sistemlerin geliştirilmesine imkan tanımaktadır. Bu sayede kullanıcılar sadece bir web tarayıcısı üzerinden herhangi bir yazılım yüklemeksizin cihazlara erişebilir.Şekil 2.1'de HTML'e ait çalışma prensibi görselleştirilerek verilmiştir.



Şekil 2.1 HTML CSS ve Javascript çalışma mantığını anlatan bir görsel

2.2.3 PiServo

Piservo, Python'un mikroservis geliştirme kütüphanelerinden biridir. Piservo, kullanıcıların Python ile RESTful API'ler geliştirmesini sağlar ve basit, hızlı ve etkili bir şekilde mikroservisler oluşturmalarına imkan tanır. Piservo, HTTP protokolüne dayalı hizmetlerin hızlı bir şekilde oluşturulmasını sağlayarak, API endpoint'leri oluşturma ve yönetme süreçlerini kolaylaştırır. Kullanıcılar, çeşitli HTTP yöntemlerini (GET, POST, PUT, DELETE vb.) kullanarak veri alışverişi yapabilir ve bu işlemleri JSON formatında gerçekleştirebilirler. Piservo, platformlar arası uyumluluğu sayesinde, Windows, macOS ve Linux gibi birçok işletim sisteminde çalışabilir uygulamaların geliştirilmesini kolaylaştırır. Hem başlangıç düzeyinden ileri seviyeye kadar farklı beceri seviyelerine hitap eden Piservo, Python ile mikroservis geliştirmek isteyenler için ideal bir seçenektir.

2.2.4 gpiozero

gpiozero, Raspberry Pi için özel olarak geliştirilmiş bir Python kütüphanesidir ve GPIO (General Purpose Input/Output) pinlerini kolay ve hızlı bir şekilde kontrol etmeyi sağlar. Basit bir yapıya sahip olan bu kütüphane, hem yeni başlayanlar hem de

deneyimli kullanıcılar için ideal bir çözümdür. gpiozero, Raspberry Pi'nin elektronik bileşenlerle etkileşim kurmasını kolaylaştırarak, temel projelerden karmaşık uygulamalara kadar geniş bir kullanım alanı sunar.

gpiozero, LED'ler, butonlar, sensörler, motorlar ve daha birçok elektronik bileşenle çalışmak için önceden tanımlanmış sınıflar içerir. Örneğin, bir LED'i kontrol etmek için sadece birkaç satır Python kodu yeterlidir. Karmaşık zamanlama ve duruma bağlı işlemler, kütüphanenin sunduğu basit yöntemlerle hızlıca gerçekleştirilebilir.

Bu kütüphane, diğer GPIO kütüphanelerine kıyasla, daha okunabilir ve anlaşılır bir syntax sunar. Örneğin, bir LED'in yanıp sönmesi veya bir butona basıldığında belirli bir eylemin tetiklenmesi gibi görevler, minimal kod ile gerçekleştirilir. gpiozero, altında RPi.GPIO veya pigpio gibi düşük seviyeli kütüphaneleri kullanır, ancak kullanıcıları bu teknik detaylardan soyutlayarak programlamayı daha erişilebilir hale getirir.

Bir başka önemli özellik ise, gpiozero ile bileşenlerin durumlarını kolayca izleyebilmenizdir. Örneğin, bir butonun basılıp basılmadığını kontrol etmek için sürekli olarak bir döngü kullanmanıza gerek yoktur. Bunun yerine, kütüphane olay tabanlı bir sistem sunar. Bu sistem, donanımda meydana gelen değişikliklere otomatik olarak tepki veren geri çağırma (callback) fonksiyonlarını kullanır.

gpiozero ayrıca, donanımı simüle etme özelliği sunar. Fiziksel bir Raspberry Pi cihazına sahip olmadan, projelerinizi sanal ortamda test edebilirsiniz. Bu, özellikle uzaktan geliştirme ve eğitim projeleri için büyük bir avantaj sağlar.

Sonuç olarak, gpiozero, Raspberry Pi projeleri için güçlü, kullanıcı dostu ve esnek bir araçtır. Elektronik projelere hızlı bir başlangıç yapmak isteyenler veya daha önceki projelerini basitleştirmek isteyenler için mükemmel bir seçimdir. Geniş dokümantasyonu ve destek topluluğu sayesinde, her seviyeden geliştirici için öğrenmesi ve kullanması kolay bir kütüphanedir.

2.2.5 paho-mqtt

paho-mqtt, MQTT protokolü üzerinden mesajlaşma işlemlerini kolaylaştırmak için geliştirilen açık kaynaklı bir Python kütüphanesidir. Eclipse Vakfı tarafından geliştirilen bu kütüphane, istemci (client) tarafında MQTT iletişimini gerçekleştirmek amacıyla yaygın olarak kullanılmaktadır. Hafif yapısı ve kullanım kolaylığı sayesinde, özellikle IoT (Internet of Things) projelerinde ve gömülü sistemlerde tercih edilmektedir.

paho-mqtt kütüphanesi, MQTT protokolünün temel işlevlerini doğrudan desteklemektedir. Bunlar arasında:

- MQTT broker'a bağlantı kurma
- Belirli bir "topic" (konu) üzerinden mesaj yayınlama (publish)
- Topic'lere abone olma (subscribe)
- Gelen mesajları dinleme ve işleme
- Bağlantı yönetimi (yeniden bağlanma, bağlantı kopması vb.) gibi işlemler bulunmaktadır.

paho-mqtt ile çalışan bir istemci, oldukça basit bir kod yapısı ile MQTT broker'a bağlanabilir ve sensör verilerini gönderebilir ya da gelen komutlara göre hareket edebilir. Bu durum, paho-mqtt kütüphanesini hem hızlı prototipleme hem de gerçek zamanlı kontrol sistemleri için ideal bir çözüm haline getirir.

paho-mqtt, genellikle şu üç ana fonksiyon etrafında yapılandırılır:

- `on_connect()`: MQTT broker'a bağlantı sağlandığında çalışan fonksiyondur.
- `on_message()`: İstemci abone olduğu bir topic üzerinden mesaj aldığında tetiklenir.
- `on_disconnect()`: Bağlantı koptuğunda devreye giren fonksiyondur.

Bu olay tabanlı yapı, sistemin daha verimli çalışmasına katkı sağlar. Örneğin, bir sensörden gelen verilerin web arayüzüne iletilmesi veya bir kullanıcı tarafından web arayüzünden gönderilen komutun donanıma iletilmesi bu yapı sayesinde kolayca gerçekleştirilebilir.

Proje kapsamında, paho-mqtt kütüphanesi, Raspberry Pi cihazında çalıştırılarak hem istemci görevi görmekte hem de fiziksel donanımlarla MQTT üzerinden veri alışverişini sağlamaktadır. Örneğin, bir web arayüzü üzerinden verilen yön komutları MQTT broker'a gönderilmekte, Pi üzerindeki paho-mqtt istemcisi bu mesajları alarak motorları yönlendirmektedir. Aynı şekilde, sıcaklık, nem ya da batarya durumu gibi

bilgiler Pi üzerinde toplanarak yine bu kütüphane aracılığıyla MQTT üzerinden yayınlanmaktadır.

paho-mqtt, güvenlik açısından da çeşitli yapılandırmalara izin verir. TLS/SSL protokolüyle şifreli bağlantı kurulabilir, kullanıcı adı-parola doğrulaması yapılabilir. Bu özellikler, özellikle VPS sunucusu üzerinden uzaktan kontrol edilen sistemlerde veri güvenliğini artırmak için önemli bir avantaj sağlar.

Genel olarak, paho-mqtt kütüphanesi, Python dili ile MQTT tabanlı haberleşme gerçekleştirmek isteyen geliştiriciler için pratik, güçlü ve esnek bir araçtır. Donanım ve yazılım bileşenleri arasında kesintisiz veri aktarımı sağladığı için, IoT sistemlerinin temel yapı taşlarından biri olarak değerlendirilebilir.

2.2.6 Owncast

Owncast, kendi canlı video yayın platformunuzu kurmanıza olanak tanıyan açık kaynaklı bir yazılımdır. Twitch, YouTube Live gibi merkezi sistemlere alternatif olarak geliştirilen Owncast, kullanıcıların kendi sunucularında bağımsız bir şekilde yayın yapmalarını sağlar. Bu sayede yayıncılar, içerikleri üzerinde tam kontrol sahibi olurken, izleyiciler de reklamsız ve özgür bir deneyim yaşar.

Owncast, özellikle Raspberry Pi, VPS veya yerel sunucular gibi çeşitli donanımlarda çalışabilecek şekilde optimize edilmiştir. Yayıncı tarafında OBS Studio gibi yayın kullanılan yazılımlardan gönderilen RTMP (Real-Time Messaging Protocol) akışını alarak, bunu modern web tarayıcıları tarafından izlenebilecek HLS (HTTP Live Streaming) formatına dönüştürür. Bu da herhangi bir eklenti gerektirmeksizin tüm cihazlarda izlenebilir bir yayın sunar.

Kullanımı oldukça basittir: Owncast, kurulumu tamamlandıktan sonra bir web arayüzü ile birlikte gelir. Bu arayüz üzerinden yayın başlatma, sohbeti yönetme, akış kalitesini ayarlama gibi işlemler rahatlıkla gerçekleştirilebilir. Ayrıca sistem, gelen izleyici mesajlarını görüntüleme, moderasyon ve canlı sohbet gibi özellikleri de destekler.

Owncast'ın mimarisi, RTMP ile alınan canlı video akışını segmentlere bölerek HLS playlistleri oluşturur. Bu yöntem sayesinde yayınlar düşük gecikmeyle, kararlı bir

şekilde izlenebilir. Ayrıca CDN veya ters proxy (örneğin: Nginx veya Cloudflare Tunnel) ile daha geniş kitlelere erişmek mümkündür.

Proje kapsamında Owncast, Raspberry Pi üzerinde çalışan bir kamera akışının VPS üzerinde host edilen bir Owncast sunucusuna gönderilmesiyle kullanılabilir. Bu sayede, bir robot ya da IoT sisteminden alınan canlı görüntü, bir web sitesinde izleyicilere sunulabilir. Özellikle gerçek zamanlı izleme gerektiren uygulamalarda — örneğin bir itfaiye robotunun kamerayla izlenmesi gibi — düşük gecikme ve güvenli yayın imkânı sunar.

Owncast ayrıca, güvenlik açısından da gelişmiş özellikler sunar. HTTPS desteği, yayın anahtarı ile erişim kontrolü ve IP sınırlamaları gibi önlemler alınabilir. Yayınların dışa kapalı olması veya yalnızca belirli izleyicilere açık hale getirilmesi mümkündür.

Sonuç olarak, Owncast, kendi yayın sistemini kurmak isteyen bireyler ve ekipler için güçlü, özgür ve esnek bir çözümdür. Geniş konfigürasyon olanakları, topluluk desteği ve açık kaynak doğası sayesinde, merkezi platformlara bağımlı kalmadan yayın yapmanın modern bir yolunu sunar.

2.3 Donanım

2.3.1 Raspberry Pi 5

Raspberry Pi 5, Raspberry Pi Vakfı tarafından geliştirilen, kredi kartı büyüklüğünde bir bilgisayar olup, önceki modellerden daha güçlü performans özelliklerine sahip olarak piyasaya sürülmüştür. Raspberry Pi 5, özellikle işlemci ve grafik performansı açısından büyük bir yükselme sunar ve daha geniş bir uygulama yelpazesini destekleyebilir. Dört çekirdekli ARM Cortex-A76 işlemcisi, 1.8 GHz hızında çalışarak, önceki Raspberry Pi modellerine göre daha yüksek hızda işlem yapabilme yeteneği sunar. Bu geliştirme, cihazın çoklu görevlerde ve işlem yoğunluklu uygulamalarda daha verimli çalışmasını sağlar.[3]

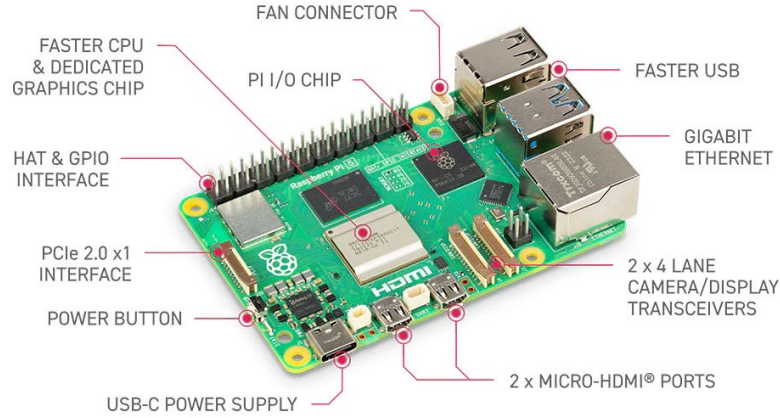
Raspberry Pi 5, daha fazla bellek kapasitesi ve daha hızlı veri işleme sunarak, 4 GB ve 8 GB LPDDR4-3200 RAM seçenekleri ile gelir. Bu bellek kapasitesi, daha büyük ve karmaşık projelerde kullanılabilirliği artırır ve cihazın çoklu uygulama çalıştırma kabiliyetini geliştirir. Ayrıca, USB 3.0 portlarının yanı sıra USB-C güç girişi, cihazın güç tüketimini optimize eder ve daha stabil bir enerji akışı sağlar.

Raspberry Pi 5, video ve grafik performansında da önemli bir iyileşme sunar. Çift 4K monitör desteği sağlayan iki adet micro-HDMI portu, 60 Hz çözünürlük ile çalışabilir ve 4K video oynatma yeteneği sunar. Bu özellik, medya projeleri ve grafik yoğun uygulamalar için büyük avantaj sağlar. Ayrıca, Raspberry Pi 5'te, grafik işlemcisi güçlendirilmiş olup, OpenGL 3.3 ve Vulkan desteği sunarak, daha gelişmiş görsel uygulamaların çalıştırılmasına olanak tanır.

Bağlantı özellikleri de önemli ölçüde iyileştirilmiştir. Gigabit Ethernet portu, daha hızlı internet bağlantıları sunarak veri transferini hızlandırır. Ayrıca, Bluetooth 5.0 ve Wi-Fi 6 (802.11ax) desteği ile daha hızlı ve güvenilir kablosuz bağlantılar sağlanır. Raspberry Pi 5, donanım hızlandırmalı video dekoderleri ile daha verimli medya oynatımı yapabilir, özellikle video akışı ve dijital medya projeleri için büyük fayda sağlar.

Raspberry Pi 5, ev otomasyonu, medya merkezi kurulumları, IoT projeleri ve hatta düşük maliyetli sunucu uygulamaları gibi geniş bir kullanım alanına sahiptir. Eğitim alanında da, öğrencilerin donanım ve yazılım geliştirme becerilerini artırmak için yaygın olarak kullanılır. Raspberry Pi 5, açık kaynaklı doğası ve geniş yazılım desteğiyle kullanıcıların farklı projeler geliştirmesine imkan tanır. Raspberry Pi OS ve diğer çeşitli işletim sistemleri, bu cihazı daha kullanıcı dostu hale getirir ve projelere başlamak için kolay bir başlangıç sağlar.

Raspberry Pi 5'in sunduğu yüksek performans, esneklik ve güçlü bağlantı seçenekleri, onu daha profesyonel projeler için de uygun hale getirmektedir. Düşük maliyetli olmasına rağmen gelişmiş özellikleri, Raspberry Pi 5'i hem hobilerle uğraşan kullanıcılar hem de profesyonel mühendisler için ideal bir platform yapmaktadır. Raspberry Pi 5'in görünümü Şekil 2.2'te gösterildiği gibidir.



Şekil 2.2 Raspberry Pi 5

2.3.2 SG90 Servo Motor

SG90, küçük boyutlu ve hafif yapısı ile popüler bir servo motor modelidir. Genellikle robotik projeler, RC araçlar, oyuncaklar ve maketlerde yaygın olarak kullanılır. SG90 servo motoru, düşük maliyeti ve kompakt tasarımı ile özellikle hobi projeleri ve eğitim amaçlı uygulamalarda tercih edilmektedir.

SG90 servo motoru, 180 derece dönme açısı sağlar ve bu açı, PWM (Pulse Width Modulation) sinyali ile hassas bir şekilde kontrol edilebilir. Bu servo motor, genellikle 5V güç kaynağı ile çalışır ve 4.8V ile 6V arasındaki gerilim aralığında en verimli şekilde çalışır. SG90, üç pimli bir bağlantıya sahiptir: pozitif güç (+), toprak (-) ve sinyal (PWM sinyali).

SG90'ın iç yapısı, plastik dişliler ve düşük güçlü bir motor içerir. Bu nedenle, düşük tork gerektiren uygulamalarda yüksek verimlilik sağlar. SG90, robotik projelerde özellikle mekanik hareketler için uygun olup, küçük ve hafif olmasından dolayı projelere rahatça entegre edilebilir. Bu özellikleri, SG90'ı özellikle küçük robot kolları, paneller, dronlar ve diğer mobil uygulamalar için popüler bir seçenek haline getirmektedir.

SG90 servo motoru, Arduino, Raspberry Pi ve diğer mikrodenetleyicilerle kolayca kontrol edilebilir. PWM sinyali aracılığıyla mikrodenetleyici, motorun pozisyonunu

belirleyerek hassas hareketler gerçekleştirilmesini sağlar. Bu sayede, servo motorun dönme açısı hassasiyetle kontrol edilebilir ve belirli görevler için gereken hareketler yapılabilir.

Ancak SG90 servo motorunun bazı sınırlamaları vardır. Özellikle düşük torku nedeniyle, yüksek ağırlık veya kuvvet gerektiren uygulamalarda verimliliği azalabilir. Ayrıca, plastik dişliler kullanıldığından, uzun süreli kullanımlarda aşınma ve performans kaybı yaşanabilir. SG90'ın dişlilerinin kırılabilir olması da, motoru daha ağır ve zorlu uygulamalarda kullanmak isteyenler için bir dezavantaj oluşturabilir.

Sonuç olarak, SG90 servo motoru, küçük ve orta ölçekli projeler için uygun fiyatlı ve kolay kullanılabilir bir seçenektir. Kompakt tasarımı ve düşük güç tüketimi ile, robotik ve elektronik projelerde, özellikle hareketli parçaların kontrolünde sıkça tercih edilen bir motor modelidir. Servo motorun görseli Şekil 2.3.'te verilmiştir.



Şekil 2.3 SG90 servo motor

2.3.3 DC Motor

DC motor (Doğru Akım Motoru), doğru akım (DC) elektrik ile çalışan ve genellikle hareket enerjisi üretmek için kullanılan elektrik motoru türüdür. DC motorları, basit yapıları ve kolay kontrol edilebilir özellikleri ile geniş bir kullanım alanına sahiptir. Bu motorlar, elektrik enerjisini mekanik enerjiye dönüştürerek, endüstriyel makinelerden, oyuncaklara, robotlara kadar pek çok uygulamada kullanılır.

DC motorun temel yapısı, bir rotor (dönme hareketini gerçekleştiren bölüm), bir stator (dönme hareketini sağlamak için sabit bir manyetik alan oluşturan bölüm) ve fırçalar (motorun elektriksel bağlantısını sağlayan bileşenler) gibi temel parçalardan oluşur. DC motor, elektriksel enerjiyi manyetik alanlarla etkileşime sokarak rotorun dönmesini sağlar. Bu dönme hareketi, motorun şaftına bağlı bir yükü hareket ettirir.

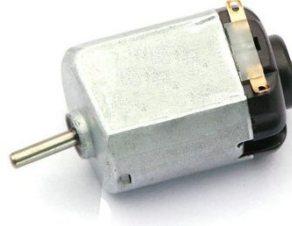
DC motorlar, genellikle iki temel bileşenle çalışır: stator ve rotor. Stator, motorun dış kısmında yer alırken, rotor motorun dönme hareketini yapan kısmıdır. Rotor, elektrik akımı tarafından yaratılan manyetik alanla etkileşime girerek dönmeye başlar. Bu hareket, rotorun ve stator arasındaki manyetik kuvvetin değişiminden kaynaklanır.

DC motorların bir diğer özelliği de hızlarının ve yönlerinin kolayca kontrol edilebilmesidir. Hız kontrolü, motorun besleme voltajı ile düzenlenebilir. Yön kontrolü ise, yön değiştiren elektrik akımının kullanılmasıyla yapılır. Bu nedenle, DC motorlar, hız ve yön kontrolünün önemli olduğu uygulamalarda tercih edilir. Motorun hızını değiştirmek için kullanılan yöntemler arasında voltajı değiştirmek, pulse-width modulation (PWM) tekniği veya direnç kullanmak sayılabilir.

DC motorların avantajları arasında yüksek verimlilik, kolay kontrol edilebilirlik, basit yapıları ve düşük maliyetleri bulunur. Ayrıca, düşük hızlarda bile tork üretme kapasitesine sahip olmaları, onları küçük robotlar, fanlar, elektrikli araçlar ve robotik kollar gibi uygulamalar için uygun hale getirir. DC motorlar, düşük hızlarda yüksek tork üretme kabiliyetine sahip oldukları için, genellikle güçlü hareketlerin gerektiği küçük ve orta ölçekli projelerde tercih edilir.

Ancak, DC motorların bazı dezavantajları da vardır. En belirgin dezavantajları, fırçalı motorlarda aşınma sorunlarıdır. Fırçaların zamanla aşındığı ve bakım gerektirdiği için, fırçasız DC motorlar gibi daha dayanıklı alternatifler de geliştirilmiştir. Ayrıca, DC motorların dönme hareketini sürekli sağlamak için doğru akım kaynağına ihtiyaçları vardır, bu da bazı uygulamalarda enerji verimliliği sorunlarına yol açabilir.

Sonuç olarak, DC motorlar, basit yapıları, verimli çalışmaları ve kolay kontrol edilebilir özellikleri ile robotik, otomasyon ve oyuncak endüstrisi gibi birçok alanda yaygın olarak kullanılır. Yüksek tork kapasitesi ve hız kontrolü gerektiren uygulamalar için ideal bir motor türüdür. DC motora ait görsel Şekil 2.4’te verilmiştir



Şekil 2.4 DC motor

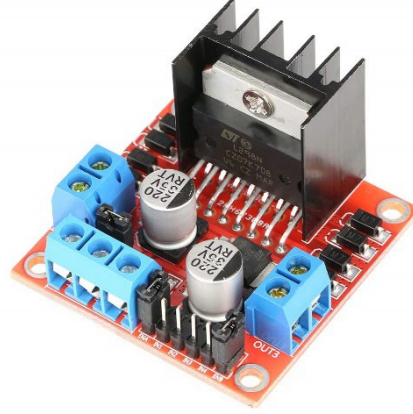
2.3.4 L298N DC Motor Sürücüsü

L298N, endüstriyel ve otomasyon projelerinde, motor sürücü olarak kullanılan bir DC motor sürücü entegresidir. Bu entegre, DC motorları veya step motorları kontrol etmek için tasarlanmıştır ve motorları çalıştırmak, hızlarını ayarlamak veya yönlerini değiştirmek için kullanılabilir. L298N, özellikle robotik projelerde, elektronik cihazlarda ve motor kontrol sistemlerinde tercih edilen bir çözümdür.

L298N, dört adet köprü (H-Bridge) çıkış devresi içerir. Bu yapı, motorun yönünü kontrol etme yeteneği sağlar ve motorun dönebilmesi için gerekli olan ileri ve geri akımları yönlendirebilir.[5] Bu sayede, motorların hızını ve yönünü hassas bir şekilde kontrol etmek mümkündür. L298N, 5V-35V arasında giriş voltajını kabul edebilir ve 2A'ye kadar motor akımı sağlayabilir.

Motor kontrolü için, L298N üzerinde bulunan iki adet çıkış kanalı ve her biri için birinci ve ikinci yarı-bridge'ler kullanılır. Bu kanallar, motorları ileri veya geri çalıştırmak için gereken elektrik sinyallerini sağlar.

Giriş Voltajı Aralığı: 5V-35V arasında çalışabilir.Çıkış Akımı: 2A'ye kadar sürekli akım sağlayabilir. Bu değer, motorun türüne ve kullanıldığı uygulamaya göre değişebilir.İleri, geri ve durma seçenekleri sunan çift yönlü motor kontrolü sağlar.Ancak, L298N'in bazı dezavantajları da vardırAnahtarlama düzenleri nedeniyle, verimlilik oranı %50-70 arasında değişebilir. Bu, bazı projelerde enerji kayıplarına neden olabilir.L298N'e ait görsel Şekil 2.6'da verilmiştir.



Şekil 2.5 L298N DC motor sürücüsü

2.3.5 12 DC Su Pompası

12V DC su pompası, düşük gerilimle çalışan, küçük çaplı sıvı transferi uygulamalarında yaygın olarak kullanılan bir elektromekanik bileşendir. Özellikle IoT projeleri, tarımsal otomasyon sistemleri, su seviyesi kontrolü ve robotik sulama sistemlerinde tercih edilen bu pompa, sıvıları belirli bir yönde transfer etmek üzere tasarlanmıştır.

Bu tip pompalar, doğrudan doğru akım (DC) ile çalışır ve çalışma mantığı, motorun dönme hareketini bir pervane veya diyafram aracılığıyla sıvı akışına dönüştürmesi esasına dayanır. 12V DC giriş voltajı ile çalıştığı için, genellikle batarya, adaptör veya harici güç kaynakları ile kolayca entegre edilebilir.

Teknik Özellikleri:

Çalışma Gerilimi: 6V – 12V DC aralığında çalışabilir, ancak maksimum performans genellikle 12V'ta elde edilir.

Akım Tüketimi: 0.5A – 1.5A arası, model ve yük durumuna bağlı olarak değişir.

Avantajları:

- Küçük boyutlu ve taşınabilir sistemlere kolay entegre edilir.
- Düşük gerilimle çalıştığı için güvenlidir.

Dezavantajları:

- Sürekli çalışma durumunda ısınabilir; bu nedenle uzun süreli kullanımda soğutma gerekebilir.
- Sadece temiz su ile kullanılmalıdır; partiküllü sıvılar, pompa içinde tıkanmalara neden olabilir.
- Debi ve basınç değerleri yüksek hacimli işler için yetersiz olabilir.

Proje kapsamında bu pompası , itfaiyenin su tazyik sisteminde kullanılmıştır. Web arayüzünden verilen MQTT komutu, Raspberry Pi üzerindeki GPIO pininden ilgili voltajı L298N üzerindeki emir pinine verir ve bu sayede L298N'in çıkış pinlerine bağlanan pompa çalışmaya başlar. 12V DC su pompası modülüne ait görsel Şekil 2.6'da verilmiştir.



Şekil 2.6 12V DC su pompası

2.3.6 ADS1115 ADC (Analog Dijital Dönüştürücü)

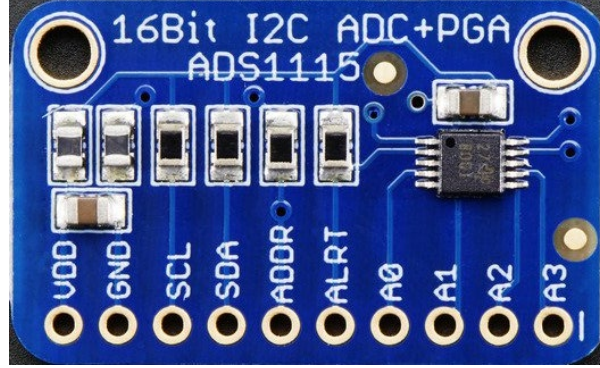
ADS1115, Texas Instruments tarafından üretilen, yüksek çözünürlüklü ve düşük güç tüketimli bir analog-dijital dönüştürücü (ADC) entegresidir. Raspberry Pi gibi yerleşik ADC'si bulunmayan platformlarda, analog sensör verilerini dijital forma dönüştürmek için kullanılır. 16-bit çözünürlük sunan ADS1115, voltajdaki küçük değişiklikleri dahi hassas şekilde algılayabilir.

ADS1115, I²C protokolü üzerinden haberleşir ve tek bir I²C adresinde birden fazla sensörden gelen analog veriyi okuyabilir. Bu yönüyle, çoklu sensör sistemleri için ideal bir çözümdür.

Teknik Özellikleri:

- Çözünürlük: 16-bit (0 – 65535 arası değer üretir)
- Giriş Sayısı: 4 adet tek uçlu giriş (A0, A1, A2, A3)
- Giriş Voltaj Aralığı: $\pm 6.144V$
- Haberleşme: I²C arayüzü

Proje kapsamında Analog verileri direkt okuyamayan Raspberry Pi 5'in SDA SCL pinlerine 2 adet ADS1115 ADC takılarak bu 8 analog pinin dördüne flame sensör , birine ise şamandıra takılmıştır. Şamandıra bağlantısı diferansiyel okuma yapılması amacıyla bir ucu 3.3V diğer ucu analog pine bağlanacak şekilde yapılandırılmıştır.ADS115 analog dijital dönüştürücüye ait görsel Şekil2.7'de verilmiştir.



Şekil 2.7 ADS1115 ADC

2.3.7 Şamandıra Modülü

Şamandıra modülü, sıvı seviyesini tespit etmek amacıyla kullanılan, basit fakat oldukça etkili bir mekanik seviye sensörüdür. Genellikle su tankları, depo sistemleri, sıvı kontrol uygulamaları ve otomatik dolum-boşaltım sistemleri gibi alanlarda tercih edilir. Çalışma prensibi, sıvı seviyesi belirli bir noktaya ulaştığında şamandıranın konum değiştirmesiyle birlikte bir anahtarın (reed switch) tetiklenmesine dayanır.

Bu modül, dijital (açık/kapalı) çıkış verir ve sıvı seviyesi belirlenen bir sınırı aştığında devreyi kapatır veya açar. Bu sayede mikrodenetleyiciye basit sinyaller göndererek sistemin otomatik kontrolünü sağlar.

Çalışma Prensipleri:

Şamandıra modülünün içinde yer alan reed switch, mıknatıslı yüzeyle karşılaştığında manyetik alan etkisiyle kontağını değiştirir. Bu değişim:

- Sıvı yükseldiğinde: Şamandıra yukarı kalkar ve anahtar kapanır (veya açılır – modelin NC/NO özelliğine göre)
- Sıvı azaldığında: Şamandıra aşağı iner ve anahtar eski durumuna döner

Ancak , proje kapsamında geliştirilen algoritma ile 0 iken 0V ; 1 iken 3.3V gerilim veren şamandıra modülünün ara değerleri diferansiyel olarak okunabilmiş ve yüzdelik dilim hesabı yapılarak depodaki su miktarı görselleştirilebilmiştir.Şamandıra modülüne ait görsel Şekil 2.8’de verilmiştir.

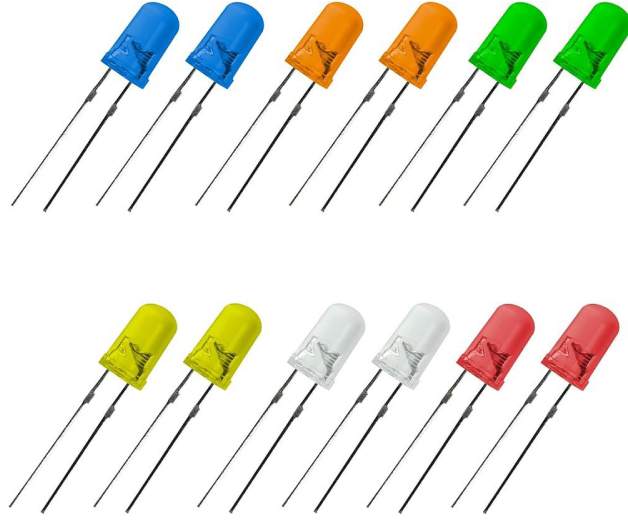


Şekil 2.8 Şamandıra modülü

2.3.8 LED Diyot

LED (Light Emitting Diode), ışık yayan bir yarı iletken diyottur. Elektrik akımının bir yarı iletkeninden geçerken ışık yaymasına dayanan bir prensibe çalışır. LED'ler, düşük enerji tüketimi, uzun ömür ve hızlı tepki süreleri gibi avantajlarla bilinir, bu nedenle modern aydınlatma ve elektronik cihazlarda yaygın olarak kullanılır. LED'ler, renk ve parlaklık gibi özelliklere göre farklı türlerde üretilir. LED'lerin yaydığı ışık, kullanılan

yarı iletken malzemeye göre deęiřir ve bu da farklı renklerin oluşmasını sağlar. Ayrıca, LED'ler sıcaklık üretmedięi için enerji verimlilięi açısından geleneksel ampullere göre çok daha etkilidir. LED'ler, aydınlatmadan ekranlara, otomobillerden televizyonlara kadar geniş bir kullanım alanına sahiptir. LED diyota ait görsel Şekil 2.9'da verilmiştir.



Şekil 2.9 LED diyot

2.3.9 PLA 3D Yazıcı Filamenti

PLA (Polylactic Acid), 3D yazıcılar için yaygın olarak kullanılan bir filament türüdür. Biyobozunur bir plastik olan PLA, genellikle mısır nişastası veya şeker kamışından elde edilen doğal hammaddelerden üretilir. Çevre dostu bir seçenek olan PLA, düşük erime sıcaklığı, düşük koku ve düşük termal çekilme gibi avantajlara sahiptir.

PLA filamentleri, 3D baskıda geniş bir kullanım alanına sahiptir. Hobi projelerinden prototiplere, endüstriyel parçalardan sanat eserlerine kadar birçok farklı uygulamada kullanılabilirler. İyi bir katman bağlantısı ve düzgün yüzey bitiři sağlarlar, bu da detaylı ve estetik olarak çekici baskılar elde etmeyi mümkün kılar. Filament hakkında örnek görsel Şekil 2.10.'de verilmiştir.



Şekil 2.10 PLA filament

PLA filamentlerinin bir diğer avantajı, geniş bir renk ve çeşitlilik yelpazesine sahip olmalarıdır. Bu, kullanıcıların istedikleri renkte ve özellikte filamenti seçmelerini sağlar. Ayrıca, PLA'nın düşük termal çekilme özelliği, 3D baskı sırasında parçanın çarpılmasını veya deformasyonunu azaltır, bu da baskıların daha istikrarlı olmasını sağlar.

Ancak, PLA filamentlerinin bazı sınırlamaları da vardır. Özellikle yüksek sıcaklık dayanımı ihtiyacı olan uygulamalarda PLA'nın sınırlı kullanımı bulunur. Ayrıca, PLA filamentleri, uzun süreli dış mekan kullanımı veya yüksek sıcaklıkta kullanım gibi zorlu koşullara dayanıklı olmayabilir.

PLA filamentler, 3D baskı dünyasında popüler bir seçenektir. Çevre dostu yapısı, geniş renk ve çeşitlilik seçenekleri ve düşük sıcaklıkta çalışma özelliği, kullanıcıların birçok farklı uygulama için PLA'yı tercih etmelerini sağlar.

2.3.10 Flame Sensör

Flame sensör (alev sensörü), ortamdaki alev veya yangın gibi ışık kaynaklarını algılamak için kullanılan bir tür algılayıcıdır. Genellikle yangın tespiti, güvenlik sistemleri ve yangına duyarlı otomasyon projelerinde kullanılır. Bu sensör, alevin yaydığı belirli dalga boylarındaki ışığı algılayarak çalışır. Özellikle 760nm ile 1100nm arası kızılötesi (IR) ışık dalga boylarına duyarlıdır. Bu dalga boyu aralığı, yanan bir alevin yaydığı ışığı temsil eder.

Flame sensörler, hızlı tepki verme özellikleri sayesinde, yangın algılama sistemlerinde erken uyarı amacıyla kullanılır. Genellikle bir fotodiyot ya da IR sensör içeren bu cihazlar, alevden gelen kızılötesi ışığı algılar ve bu bilgiyi bir çıkış sinyali olarak sunar. Bu çıkış sinyali, mikrodenetleyiciler tarafından okunarak bir alarm sistemini tetiklemek, bir motoru durdurmak veya başka güvenlik önlemlerini devreye sokmak için kullanılabilir.

Alev sensörleri dijital ve analog çıkışlara sahip olabilir. Dijital çıkış, belirli bir eşik üzerinde ışık yoğunluğunda sinyal üretirken; analog çıkış, algılanan ışık yoğunluğunu kademeli olarak iletebilir. Bu sayede kullanıcı, sensörün hassasiyetini daha hassas şekilde kontrol edebilir.

Flame sensörler genellikle aşağıdaki bileşenlerden oluşur:

IR alıcı diyot veya fototransistör: Alevden gelen kızılötesi ışığı algılayan temel bileşen.

Yükselteç ve karşılaştırıcı devre: Algılanan sinyali işleyerek dijital veya analog çıkış sinyali üretir.

Ayarlanabilir potansiyometre: Sensörün hassasiyetini manuel olarak ayarlamak için kullanılır.

Çıkış pinleri (DO ve AO): Dijital ve analog çıkışlar için bağlantı noktalarıdır.

Flame sensörlerin avantajları arasında düşük maliyet, basit kullanım ve güvenilir alev algılama yer alır. Küçük boyutları sayesinde birçok elektronik projeye entegre edilebilirler. Robotik uygulamalarda, yangın algılayan ve yangına müdahale eden sistemlerde yaygın olarak kullanılırlar.

Ancak bazı dezavantajları da mevcuttur. Örneğin, güneş ışığı veya yüksek IR ışınına sahip başka kaynaklar sensörü yanlış tetikleyebilir. Bu durum, dış ortamda veya yoğun ışıklı ortamlarda kullanımı sınırlayabilir. Ayrıca, duman veya cam gibi engeller, sensörün alevi algılamasını zorlaştırabilir. Bu nedenle, sensörün yerleşimi ve ortam koşulları dikkatlice planlanmalıdır.

Dezavantajları ise sınırlı ses kontrolü (aktif buzzer'lar için), belirli frekanslarda sabit ses üretme ve bazı ortamlarda rahatsız edici ses düzeyleri olabilir. Ayrıca, uzun süreli çalıştırıldığında sesli çevre kirliliği oluşturabilir. Proje kapsamında aktif buzzer kullanılmaktadır fakat pasif buzzer'a geçilme ihtimali de göz önünde bulundurulmaya devam etmektedir. Buzzer'a ait görsel Şekil 2.12'de verilmiştir.



Şekil 2.12 Piezo buzzer

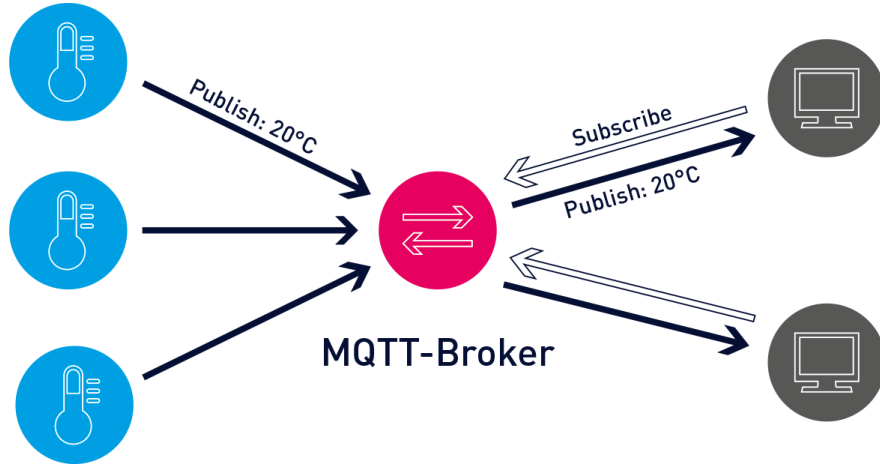
2.4 Haberleşme ve Veri İletimi

2.4.1 MQTT(Message Queuing Telemetry Transport)

MQTT, düşük bant genişliği ve düşük güç tüketimi gerektiren cihazlar arasında hafif yapısıyla veri iletişimini sağlayan bir mesajlaşma protokolüdür. İlk olarak IBM tarafından geliştirilen bu protokol, özellikle IoT (Nesnelerin İnterneti) uygulamalarında oldukça yaygın olarak kullanılmaktadır.

MQTT, istemci-sunucu modeline dayalı olarak çalışır. İstemciler (örneğin bir Raspberry Pi veya bir web arayüzü), bir MQTT broker aracılığıyla birbirleriyle haberleşir. Mesajlar belirli konular (topic) üzerinden yayınlanır ve bu konulara abone olan istemciler bu mesajları alır. Bu yapı sayesinde, sistemin her bileşeni birbirinden bağımsız olarak çalışabilir ve mesaj trafiği basit ve etkili bir biçimde yönetilir.

Proje kapsamında, kullanıcı web arayüzünden MQTT protokolü aracılığıyla komut gönderir. Bu komutlar, bir VPS üzerinde kurulu olan MQTT broker'a iletilir. Raspberry Pi üzerindeki istemci bu mesajları dinler ve alınan komutlara göre fiziksel donanımı (motorlar, LED'ler, pompalar) kontrol eder. Aynı zamanda, Pi üzerinden toplanan sensör verileri de MQTT üzerinden yayınlanarak kullanıcıya anlık bilgi aktarımı sağlanır. MQTT protokolünün çalışma sistemine ait örnek görsel Şekil 2.13'de verilmiştir.



Şekil 2.13 MQTT protokolü çalışma prensibi

Örnek görselde de görüldüğü üzere tüm bu sistemin çalışması için bir MQTT Broker'a ihtiyaç vardır. Bu sebeple projenin gerçekleştirilmesi için gerekli olan broker altyapısı ya sürekli çalışan bir bilgisayar ile olacak ya da VPS (Virtual Private Server) kiralatarak o bilgisayar üzerinden yapılacaktır.

Sistemin sektöre uymaması amaçlandığı için broker olarak VPS kiralatılmış ve bu VPS üzerinden gerekli veri iletim işlemleri yapılmıştır.

2.4.2 VPS (Virtual Private Server)

VPS, fiziksel bir sunucunun sanal bölümlere ayrılmasıyla oluşturulan, her birinin bağımsız bir sistem gibi çalışabildiği sanal sunuculardır. Bu sunucular, maliyet ve yönetim kolaylığı açısından hem bireysel geliştiriciler hem de kurumsal sistemler tarafından tercih edilmektedir.

Bu projede, VPS üzerinde bir MQTT broker (Mosquitto gibi) kurularak hem veri iletimi hem de sistem kontrolü merkezi hale getirilmiştir. Bu yapı, sistemin yerel ağ sınırlarının dışına çıkarak uzaktan erişilebilir hale gelmesini sağlar.

VPS sunucusu ayrıca, web arayüzünün barındırıldığı ortam olarak da kullanılmıştır. Böylece kullanıcılar internete bağlı herhangi bir cihazdan sadece tarayıcı kullanarak araca erişebilir, komut gönderebilir ve geri bildirim alabilir. Bu yaklaşım, sistemin taşınabilirliğini artırırken, herhangi bir yazılım kurulumuna gerek kalmadan geniş bir kullanıcı kitlesine erişim olanağı tanır. VPS'e ait örnek görsel Şekil 2.14'de verilmiştir.



Şekil 2.14 VPS'e ait örnek görsel

2.4.3 Owncast

Owncast, açık kaynak kodlu ve kendine barındırılabilir bir canlı video yayın sistemidir. Modern yayın ihtiyaçlarına yönelik geliştirilen bu yazılım, internet üzerinden gerçek zamanlı video akışını izleyicilere sunmak amacıyla kullanılmaktadır. Yayıncı tarafında alınan medya akışı, RTMP (Real-Time Messaging Protocol) üzerinden Owncast sunucusuna iletilir ve burada işlenerek HLS (HTTP Live Streaming) protokolü ile son kullanıcıya ulaştırılır. Bu mimari, sistemin düşük gecikme süresiyle geniş kitlelere yayın yapabilmesini mümkün kılar.

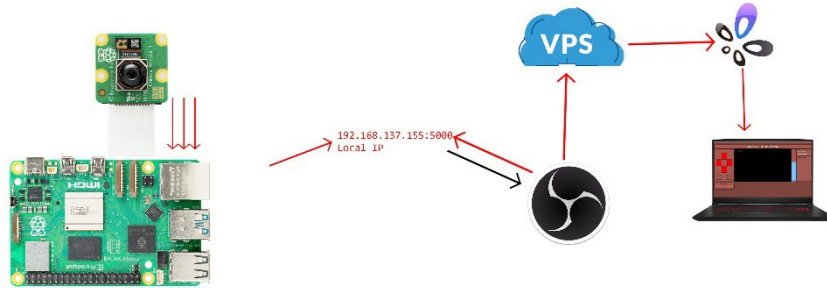
Owncast'ın en dikkat çeken özelliklerinden biri, herhangi bir üçüncü taraf servise ihtiyaç duymadan tamamen bağımsız çalışabilmesidir. Böylece, veri güvenliği, özelleştirme imkânı ve bant genişliği kontrolü tamamen sistem sahibine ait olur. Kullanıcılar, OBS Studio, ffmpeg gibi medya iletim araçlarını kullanarak yayınlarını kolaylıkla Owncast sunucusuna aktarabilir. Yayınların izlenmesi ise, sistemle birlikte gelen web arayüzü üzerinden gerçekleştirilir.

Owncast, yapılandırma dosyaları aracılığıyla tüm yayın ayarlarının düzenlenmesine imkân tanır. config.yaml dosyası üzerinden yayın başlığı, açıklaması, kalite profilleri ve güvenlik seçenekleri tanımlanabilir. Geniş API desteği ile sisteme harici kontrol paneli veya özel arayüzler entegre etmek de mümkündür.

Bu bitirme projesinde Owncast, geliştirilen uzaktan kontrollü itfaiye aracı sisteminin görsel takibini sağlamak amacıyla kullanılmıştır. Raspberry Pi cihazına bağlı olan kamera modülü, çevredeki görüntüyü sürekli olarak yakalamakta ve ffmpeg yazılımı aracılığıyla RTMP protokolü üzerinden VPS (Virtual Private Server) üzerinde çalışan Owncast sunucusuna aktarmaktadır.

Owncast, alınan bu video akışını gerçek zamanlı olarak işler ve izlenebilir HLS formatına dönüştürür. Daha sonra bu akış, projenin web arayüzüne gömülerek kullanıcıya sunulur. Kullanıcılar, web tarayıcıları üzerinden robotun çevresini canlı olarak izleyebilir ve duruma göre sistemde yer alan yönlendirme kontrollerini kullanarak robota komut gönderebilir.

Bu yapı sayesinde, herhangi bir bulut servisine bağlı kalmaksızın düşük gecikmeli, güvenli, bağımsız ve özelleştirilebilir bir video yayını gerçekleştirilmiş olur. Ayrıca Owncast'ın açık kaynaklı yapısı sayesinde sistemin tüm bileşenleri üzerinde tam kontrol sağlanarak, gerekirse projeye özel modifikasyonlar yapılabilmektedir. Proje kapsamında kamera verisinin web kontrol arayüzüne nasıl aktarıldığına dair gerekli sistem mimarisi görseli Şekil2.15'te verildiği gibidir.



Şekil 2.15 Kamera verisi iletimine ait sistem mimarisi

3. NESNELERİN İNTERNETİ (IOT)

Internet of Things (IoT), nesnelerin interneti olarak bilinen bir kavram ve teknolojidir. Bu teknoloji, farklı cihazların, sensörlerin ve diğer fiziksel nesnelerin internete bağlanmasını ve birbirleriyle iletişim kurmasını sağlar. IoT, bu nesneler arasındaki veri alışverişini ve işbirliğini kolaylaştırarak çeşitli endüstrilerde verimliliği artırır ve yeni olanaklar sunar. IoT'nin günlük yaşamda kullanımı oldukça yaygındır. Ev otomasyonu alanında, IoT cihazları ev sahiplerine akıllı termostatlar, güvenlik kameraları, akıllı aydınlatma sistemleri ve ev eşyaları gibi bir dizi cihazı internet üzerinden kontrol etme imkanı sunar. Bu sayede, ev sahipleri uzaktan evlerini izleyebilir, enerji kullanımını optimize edebilir ve güvenliklerini artırabilirler. Endüstriyel alanda da IoT'nin kullanımı giderek artmaktadır. Fabrikalarda, IoT sensörleri üretim ekipmanlarını izler, verimliliği artırır ve bakım gereksinimlerini belirler. Tarım sektöründe, IoT cihazları toprak nemini, hava koşullarını ve bitki büyümesini izleyerek çiftçilere daha verimli tarım yöntemleri geliştirme fırsatı sunar. Sağlık sektöründe, IoT cihazları hasta sağlığını izleyebilir, yaşlı bireylerin güvenliğini sağlayabilir ve hastanelerin operasyonlarını optimize edebilir. Bununla birlikte, IoT'nin yaygınlaşması beraberinde bazı endişeleri de getirir. Güvenlik, en önemli endişelerden biridir. IoT cihazlarının birçoğu, zayıf güvenlik önlemleri nedeniyle siber saldırılara ve veri ihlallerine karşı savunmasız olabilir. Ayrıca, IoT cihazlarının büyük miktarda kişisel veri toplaması, saklaması ve işlemesi, gizlilik endişelerini de artırır. Sonuç olarak, IoT teknolojisi günümüzde giderek önem kazanan ve birçok alanda kullanılan bir teknolojidir. Ancak, bu teknolojinin etkili bir şekilde kullanılabilmesi için güvenlik, gizlilik ve veri yönetimi gibi konuların dikkatle ele alınması gerekmektedir. İlerleyen yıllarda, IoT'nin daha da yaygınlaşması ve gelişmesi beklenmektedir, bu nedenle bu endişelerin çözülmesi büyük önem taşır.

3.1 IoT Bağlantı Türleri

IoT (Nesnelerin İnterneti), cihazların birbirleriyle veya internet üzerinden iletişim kurarak veri alışverişi yapmasını sağlayan bir teknoloji alanıdır. IoT cihazları genellikle çeşitli bağlantı türleri aracılığıyla bir ağa veya internete bağlanır ve bu bağlantılar, cihazların veri toplamasını, analiz etmesini ve kontrol etmesini sağlar.

Wi-Fi: Wi-Fi bağlantısı, IoT cihazlarının kablosuz bir yerel ağı bağlanmasını sağlar. Bu, evde veya işyerindeki Wi-Fi erişim noktalarına bağlanarak internete erişimi mümkün kılar. Wi-Fi bağlantısı, yüksek bant genişliği sağlar ve geniş kapsama alanı sunar, bu nedenle ev otomasyonu, akıllı cihazlar ve endüstriyel IoT uygulamalarında sıkça kullanılır.

Bluetooth: Bluetooth, kısa mesafe kablosuz iletişim sağlayan bir teknolojidir ve IoT cihazlarının birbirleriyle veya yakındaki cihazlarla etkileşime girmesini sağlar. Bluetooth, düşük güç tüketimi ve basit kurulumuyla öne çıkar ve genellikle kişisel cihazlar, sağlık izleme cihazları ve akıllı ev cihazları gibi uygulamalarda kullanılır.

Zigbee: Zigbee, düşük güç tüketimi ve düşük veri hızı gerektiren IoT uygulamaları için tasarlanmış bir kablosuz iletişim protokolüdür. Zigbee ağları, düşük maliyetli sensör ağları oluşturmak için kullanılabilir ve genellikle akıllı evler, endüstriyel otomasyon ve akıllı enerji sistemleri gibi alanlarda tercih edilir.

LoRaWAN: LoRaWAN (Uzun Menzilli Geniş Alan Ağı), uzun menzilli düşük güç tüketimli IoT cihazlarının iletişimini sağlayan bir protokoldür. LoRaWAN, sensörlerin düşük güç tüketimiyle uzak mesafelere iletişim kurmasını sağlar ve genellikle kentsel alanlarda akıllı şehir uygulamaları, tarım izleme ve endüstriyel IoT uygulamaları gibi alanlarda kullanılır.

Cellular (Mobil): Cellular bağlantıları, IoT cihazlarının mobil şebekelere (GSM, 3G, 4G, 5G) bağlanmasını sağlar. Bu, IoT cihazlarının herhangi bir yerde internete erişim sağlamasına olanak tanır, ancak yüksek enerji tüketimi ve abonelik maliyetleri gibi bazı dezavantajları vardır. Bu nedenle, özellikle hareket halindeki cihazlar veya uzak alanlarda kullanılan IoT uygulamaları için tercih edilir.

MQTT: MQTT (Message Queuing Telemetry Transport), düşük bant genişliği, düşük güç tüketimi ve güvenilir mesaj iletimi sağlayan bir mesajlaşma protokolüdür. Özellikle IoT (Nesnelerin İnterneti) uygulamaları için tasarlanmış olan MQTT, cihazlar arasında verilerin hızlı ve güvenli bir şekilde iletilmesine olanak tanır.

Protokol, yayıncı-abone modeline dayanır ve genellikle sensörler, cihazlar ve sunucular arasında iletişim için kullanılır.

3.2 IoT Ağlarının Türleri

IoT (Nesnelerin İnterneti) uygulamaları için kullanılan ağlar, genellikle cihazların birbirleriyle veya internet üzerinden iletişim kurmasını sağlayan çeşitli teknolojilerden oluşur. Bu teknolojiler, IoT cihazlarının belirli ihtiyaçlarına ve uygulama senaryolarına göre farklılaşabilir. İşte yaygın olarak kullanılan bazı IoT ağlarının türleri:

Kablosuz Alan Ağı (Wireless LAN - WLAN): WLAN, kablosuz yerel ağ teknolojilerini kullanarak IoT cihazlarının bir ağa bağlanmasını sağlar. Wi-Fi, en yaygın WLAN standardıdır ve genellikle ev otomasyonu, akıllı cihazlar ve işyeri ortamlarında kullanılır. WLAN, yüksek hızlı veri iletimi ve geniş kapsama alanı sağlar.

Düşük Güç Kablosuz Ağı (Low-Power Wireless Network - LPWN): LPWN, düşük güç tüketimi gerektiren IoT cihazları için tasarlanmıştır. Bu ağlar, cihazların uzun pil ömrü ile çalışmasını sağlar ve genellikle pil ile çalışan sensör ağları ve uzun menzilli IoT uygulamaları için kullanılır. LPWN teknolojileri arasında Zigbee, Z-Wave ve LoRa gibi protokoller bulunur.

Kişisel Alan Ağı (Personal Area Network - PAN): PAN, IoT cihazlarının yakın mesafede birbirleriyle iletişim kurmasını sağlar. Bluetooth, en yaygın PAN teknolojisidir ve genellikle kişisel cihazlar arasında veri paylaşımı veya kablosuz kulaklık gibi uygulamalarda kullanılır.

Geniş Alan Ağı (Wide Area Network - WAN): WAN, IoT cihazlarının geniş bir coğrafi alanda dağılmış olabileceği durumlar için tasarlanmıştır. Bu ağlar, mobil ağlar (GSM, 3G, 4G, 5G) veya uydu bağlantıları gibi teknolojileri kullanarak IoT cihazlarının internete bağlanmasını sağlar. WAN, uzaktan izleme, taşımacılık ve akıllı şehir uygulamaları gibi geniş kapsamlı IoT projelerinde kullanılır.

Yerel Alan Ağı (Local Area Network - LAN): LAN, belirli bir bina veya tesis içindeki IoT cihazlarının bir ağına bağlanmasını sağlar. Ethernet, en yaygın LAN teknolojisidir ve genellikle endüstriyel otomasyon ve akıllı bina sistemleri gibi uygulamalarda kullanılır.

3.3 IoT ve Veri tabanları

IoT cihazlarından gelen verilerin işlenmesi, depolanması ve yönetilmesi için veri tabanları kritik bir rol oynar. IoT uygulamaları genellikle büyük miktarda veri üretir ve bu verilerin etkili bir şekilde depolanması ve analiz edilmesi gereklidir. Veri tabanları, IoT verilerini düzenli bir şekilde depolamak ve sorgulamak için kullanılır.

SQL Veri tabanları (Structured Query Language): IoT cihazlarının topladığı yapılandırılmış veriler için SQL tabanlı veri tabanları yaygın olarak kullanılır. SQL veri tabanları, verilerin tutarlı bir şekilde depolanmasını sağlar ve verilerin hızlı bir şekilde sorgulanmasına olanak tanır.

NoSQL Veri tabanları: IoT uygulamaları, yapılandırılmamış veya yarı yapılandırılmış verilerle çalışabileceği için NoSQL veri tabanları da tercih edilir. Bu veri tabanları, verilerin esnek bir şekilde depolanmasını sağlar ve büyük veri analizleri için idealdir.

Bulut Tabanlı Veri tabanları: IoT cihazları verilerini genellikle bulut ortamında depolar. Bulut tabanlı veri tabanları, cihazlardan gelen verilerin merkezi bir yerde toplanmasını ve bu verilere internet üzerinden her yerden erişilmesini sağlar. AWS, Google Cloud, Microsoft Azure gibi bulut servis sağlayıcıları, IoT projeleri için veri tabanı çözümleri sunar.

4. AKILLI İTFAİYE ARACI TASARIMI VE GERÇEKLENMESİ

4.1 Giriş

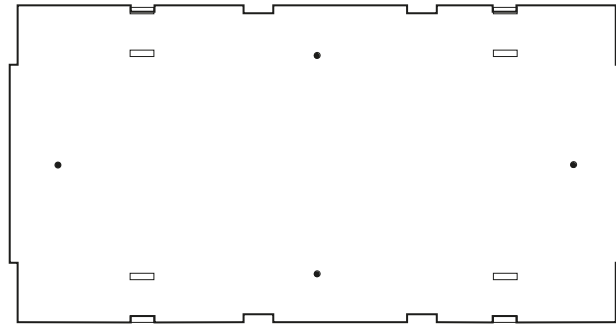
Proje yazılım ve donanım olarak iki kolda incelenecektir. Web arayüzü kısmı ve uzaktan kontrollü itfaiye araba olarak 2 kısıma sahip olacaktır.

İtfaiye arabası , içerisine elektronik komponent ve sensörlerin konumlandırılacağı şekilde yapılacağı için iç tasarımı olmayacaktır sadece dış işkelet şeklinde kullanılan bir kılıf şeklinde olacaktır. Bu sayede hem iç karışıklık ve kötü görüntünün önüne geçilecek hem de dışarısında bir araba iskeleti olduğu için daha güzel görünmesi amaçlanmıştır.

Web arayüzü kısmına geldiğimizde ise Adobe Dreamweaver üzerinden HTML kullanılarak kodlanacak olan uygulama haberleşme bağlantısının da mqtt protokolü üzerinden yapılması amaçlanmıştır.

4.2 Uzaktan Kontrollü İtfaiye Araç Donanımsal Tasarımı

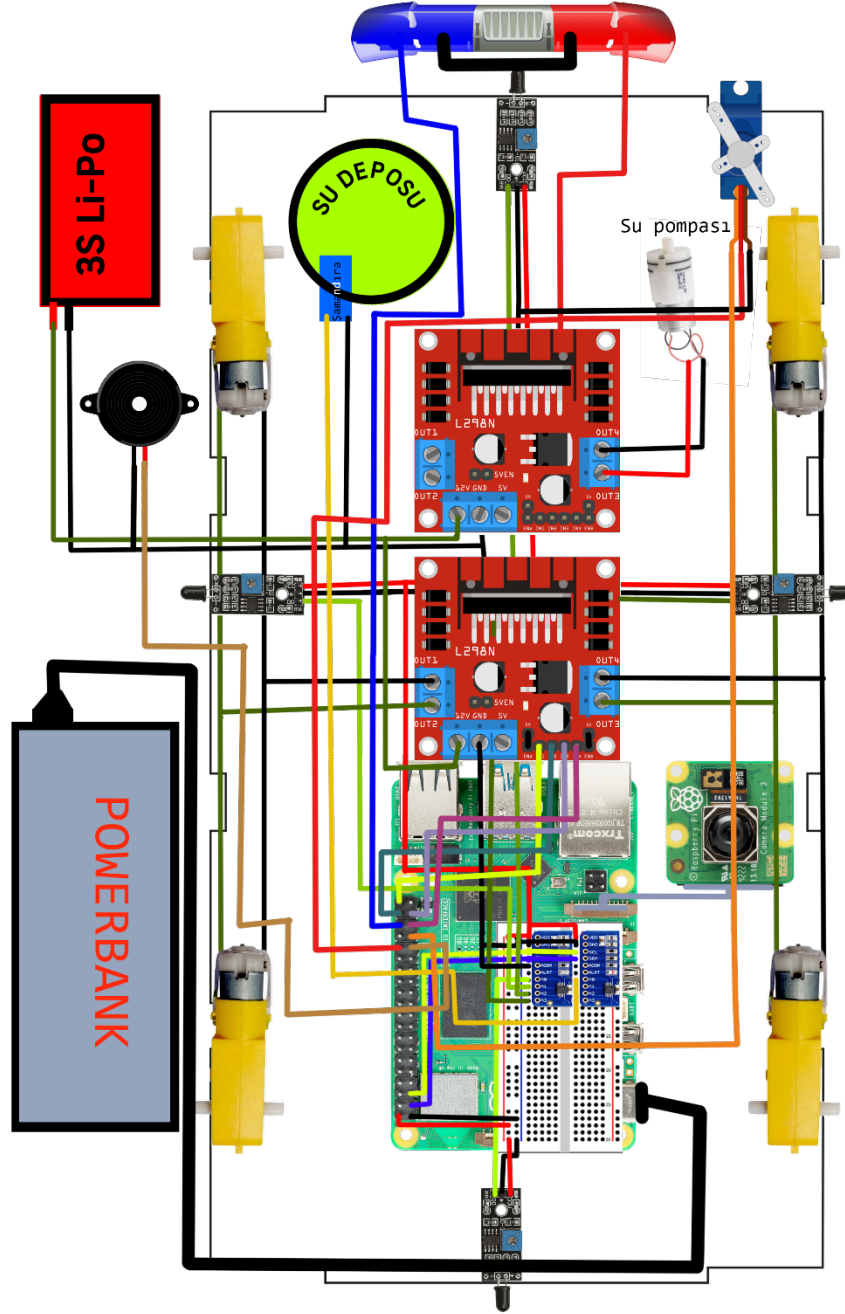
RC araç tasarımının şasisi tarafımda Adobe İllustrator programında çizilmiş akabinde lazer CNC ile maliyet ve ağırlık performansı göz önünde bulundurularak sunta malzemesinden kesilerek oluşturulmuştur. Araç şasi tasarımının vektörel çizimi Şekil4.1’de verildiği gibidir.



Şekil 4.1 İtfaiye aracı şasi tasarımı

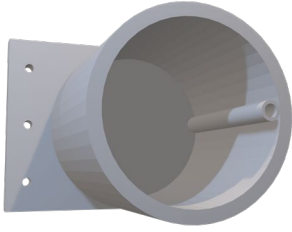
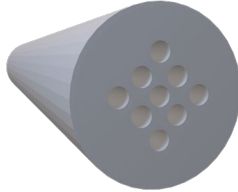

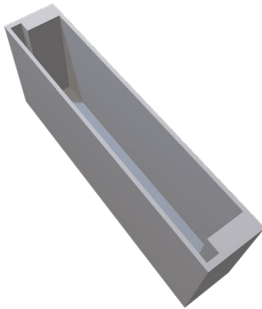
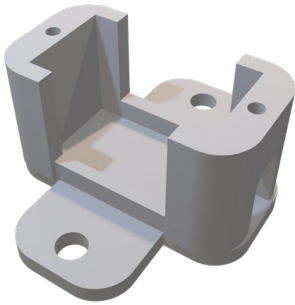
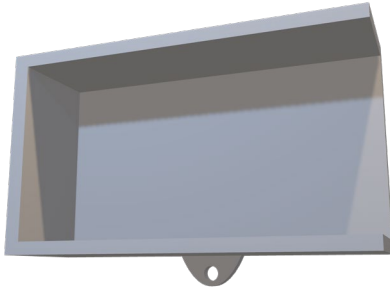
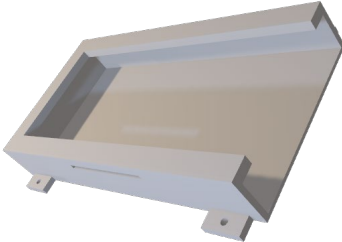
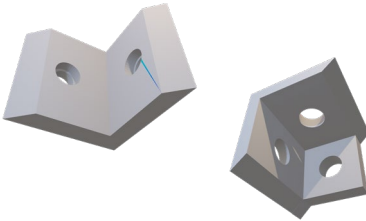
Daha sonrasında ise itfaiye aracının elektronik montajı bu şasi üzerine yapılmıştır. Tüm sistem , bir yüzey üzerine entegre edilmek istendiği için bu şasi yüzeyi üzerine

elektronik komponentler optimum şekilde yerleştirilmeye çalışılmıştır. Sisteme ait bağlantı şeması Şekil 4.2’de verilmiştir.



Şekil 4.2 Elektronik bağlantı şeması

İtfaiye aracının 3B tasarımları da itfaiye şöför kabini hariç tarafımca SolidWorks programıyla tasarlanmıştır. Bu tasarımlara ait parçalar Tablo4.1’de verilmiştir.

 <p>Su deposu tasarımı</p>	 <p>Su basıncını arttırmak için yapılan parça tasarımı</p>
 <p>12V DC su pompası tutucu tasarımı</p>	 <p>Siren Çakar kısmı difüzör tasarımı</p>
 <p>Servo tutucu tasarımı</p>	 <p>3S Li-Po Pil tutucu</p>
 <p>Powerbank tutucu tasarımı</p>	 <p>Bağlantı parçaları tasarımı</p>

Tablo 4.1 3B parça tasarımları

Daha sonrasın itfaiye aracının şasisi üzerine motajlanan elemanlar itfaiye aracına bir bütün araç görünümü verilmek amacıyla yine sunta malzemeden kesilmiş olan gövde parçaları ile çerçevelenmiş ve daha sonra görsel kozmetik olarak itfaiye aracına benzemesi amacı ile Adobe Illustrator programında , Türk itfaiye araçlarına benzer bir tasarım yapılmıştır. Tasarıma ait detaylar Şekil 4.3’te verilmiştir.

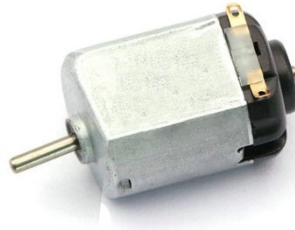


Şekil 4.3 İtfaiye aracı dış gövde tasarımları

Daha sonrasında bu tasarımlar sticker üzerine basılmış ve bu stickerlar sunta üzerine yapıştırılmıştır.Sonuç olarak itfaiye arabası tek ve entegre bir araba sistemi şekilde görünmeye başlamıştır.

4.3 İtfaiye Aracı Elektronik Tasarımı

RC aracın tekerlerini hareket ettirmek için DC Motor kullanılacaktır. Gerek kolay bulunumu gerekse de düşük voltajla çalışması sebebi ile 5V DC Motor seçilmiştir.İlgili komponente ait görsel Şekil 4.4’de verilmiştir.



Şekil 4.4 5V DC motor

Aracın siren sistemi ve diğer aydınlatmaları için LED diyotlar kullanılacaktır.Hem görsel olarak güzel gözükmesi hem de sisteme fazla yük binmemesi amaçlanmıştır.

Aracın su püskürtme sistemi için 12V DC Motor entegreli su pompası modülü seçilmiştir. Bu pompa GPIO pinleri üzerinden kontrol edilebilen röle kartı sayesinde olacaktır.İlgili komponente ait görsel Şekil 4.5’de verilmiştir.



Şekil 4.5 12V DC su pompası

Aracın sulama sistemindeki hareketleri SG90 servo ile sağlanacaktır.Servo dönme derecesini websitesi üzerinde bulunan arayüzden kullanıcı girdisi ile almaktadır.

Aracın hareket etmesini sağlayacak olan DC motorlara gerekli olan 12V güç 3S Li-Po batarya sayesinde verilecektir.

Araca ait elektronik komponent bağlantı şeması Şekil4.2.’de verildiği gibidir.

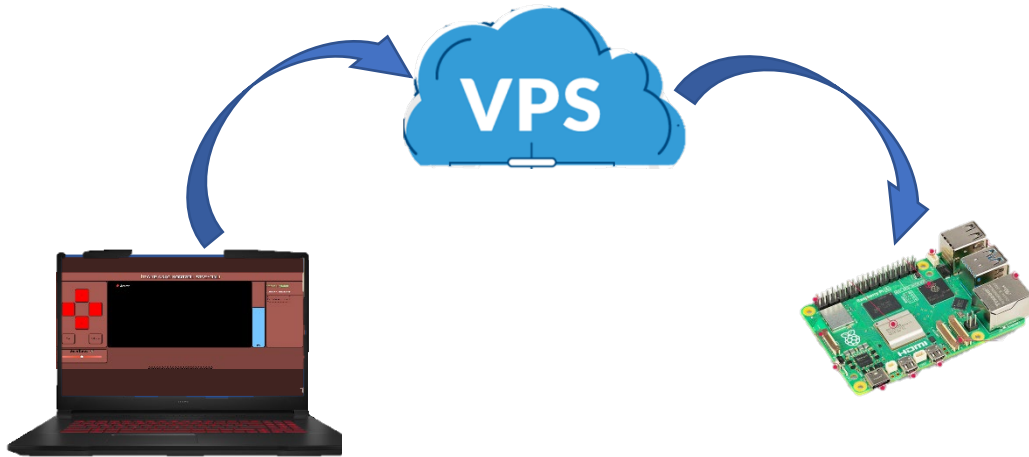
4.4 İtfaiye Aracı Yazılım Tasarımı

RC aracın yazılımı Visual Studio Code üzerinden python dili kullanılarak yazılacaktır.Python dilinin kullanılma sebebi Raspberry Pi 5 ile olan uyumluluğu ve kullanılacak olan sensör ve komponentler ile olan kütüphane uyumluluğudur.

Çok tercih edilen bir dil olması sebebi ile bir çok komponent ile ilgili kütüphane verisine sahip olması pythonu bu konuda bir adım öne çıkarıyor.Ayrıca broker bağlantısı için de gerekli işlemlerin nispeten daha kolay olması ve herhangi bir olumsuzlukta internetten yardımcı kaynak bulunma olasılığının da fazla olması öne çıkan sebeplerden birisi.

Genel olarak servolar için PiServo kütüphanesi , GPIO pinlerine bağılı olacak olan LED sistemleri ve su pompası modülü için gpiozero gibi popüler kütüphaneler kullanılması planlanmaktadır.

Sistem sensörlerden aldığı verileri brokera iletecek ve bu sayede kullanıcıya bildirecek ve aynı zamanda kontrol edilebilecektir. Genel sistem çalışma prensibine ait akış şeması Şekil 4.6’te verilmiştir.

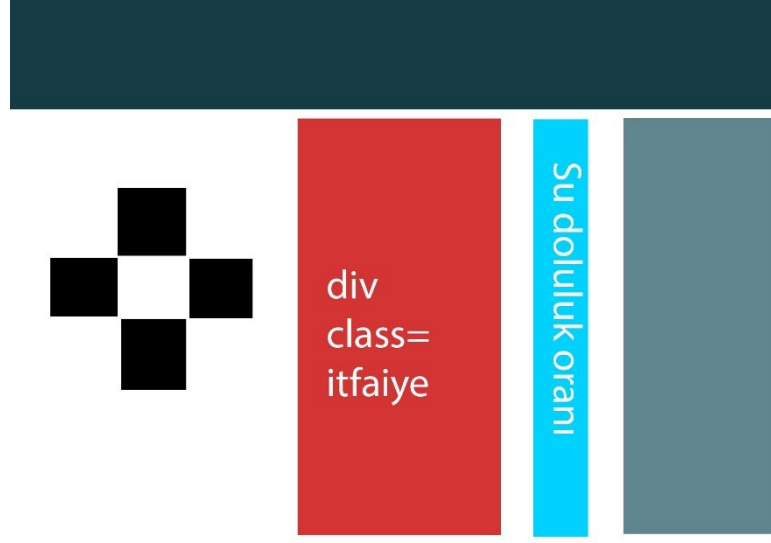


Şekil 4.6 Sistemin çalışma prensibine ait akış şeması

4.5 Web Arayüzü Yazılım Tasarımı

Web sayfası kısmında kontrol arayüzü olacak olan gerekli modüller olacaktır ve by web sayfası HTML dilinde Javascript ve CSS kullanılarak yazılacaktır. Web sayfasının tasarımı ve yazılımı tarafımca yapılacaktır.

UI dizaynı kolay ve basit bir tasarım ve kullanım kolaylığı prensibi ile yapılmıştır ve ilk prototipi Şekil4.4'te gösterildiği gibidir.



Şekil 4.7 Sistem arayüz prototipi

Web arayüzü genel olarak 3 kısımdan oluşmaktadır. Sol tarafta araç kontrol kısmı , orta kısımda araç kamera görüntüsü , sağ tarafta ise araç ile iletişim bilgileri (su doluluk oranı , verilerin brokera ulaşp ulaşmadığı vb.). Arayüzün Şekil4.7’de gösterilen prototipe göre HTML üzerinden tasarlanmış hali Şekil 4.8’te verilmiştir.



Şekil 4.8 Arayüz görüntüsü

Daha sonrasında websitesi yerel ağ yerine ortak ağ da olacağı için hosting ve domain adreslerini alınmış akabinde <http://halilibrahimcetin.com.tr> sitesinde yayına alınmıştır.

Arayüze ait gerekli kodların bulunduğu index.html sayfasından mqtt bağlantısı ve verilerin iletilmesi ile ilgili olan gerekli kod parçası Şekil 4.9’da verildiği gibidir.

```
<script>
// MQTT Broker bağlantısı ve abone olunacak topic'ler
const MQTT_BROKER = 'ws://178.208.187.184:9001';

const MQTT_TOPICS = ['sensor/veri', 'ileri/veri','sol/veri','sag/veri','geri/veri','siren/veri','pompa/veri','servo/veri'];

// MQTT Client bağlantısını oluştur
const client = mqtt.connect(MQTT_BROKER);

// Bağlantı başarılı olduğunda
client.on('connect', () => {
  console.log("MQTT'ye bağlandı!");
  let statusElement = document.getElementById("status");
  statusElement.innerText = "Status: Bağlandı";
  statusElement.style.textShadow = "2px 2px 4px rgba(0, 0, 0, 0.5)";
  statusElement.style.color = "lightgreen"; // Yeşil renk yap

  // Birden fazla topic'e abone ol
  MQTT_TOPICS.forEach(topic => {
    client.subscribe(topic, (err) => {
      if (err) {
        console.log('Topic aboneliği başarısız: ${topic}');
      } else {
        console.log('Topic'e abone olundu: ${topic}');
      }
    });
  });
});

// Mesaj alındığında
client.on('message', (topic, message) => {
  const veri = message.toString();
  console.log("Gelen Veri:", veri);
});
</script>
```

Şekil 4.9 MQTT bağlantısı ve gerekli topiclere subscribe işlemi

Arayüzde öne çıkan divlere ait detaylar Tablo4.2.’de verilmiştir.

Header	Arayüzde en üstte bulunan ve tüm genişliği kaplayan “İTFAİYE ARACI KONTROL İSTASYONU YAZAN” kısım
Kontrol	Araca ait kontrol edilebilen özelliklerin toplandığı kısımdır . Bu kısımdan siren aç kapa , yönlendirme , su pompası aç kapa , hortum yönü gibi değerler ayarlanabilir
Resim	Araç önü canlı görüntünün Owncast aracılığıyla yayınlanacağı kısımdır
Su seviye	Araç su haznesinde ne kadar su kaldığı ile alakalı animasyonlu bir görsele sahip gösterge bulunur
Veriler	Arayüz ve broker arasında bağlantı sağlandı mı , brokera veriler gidiyor mu , hangi veri hangi topic’e ulaştı,son giden başarılı veri gibi detayları içerir

Tablo 4.2 Arayüze ait fonksiyon tablosu

İtfaiye aracı ile websitesi arasında yapılan MQTT bağlantısında bulunan topicler Tablo 4.3'te verildiği gibidir.

Topic	Veri Tipi	İşlev
ileri/veri	Binary	Butona basıldığı sürece 1 verisi gider , Raspberry 1 verisi aldığı sürece motorların hepsine ileri emri verir
sag/veri	Binary	Butona basıldığı sürece 1 verisi gider , Raspberry 1 verisi aldığı sürece soldaki motorlara ileri , sağdaki motorlara geri emri verir
geri/veri	Binary	Butona basıldığı sürece 1 verisi gider , Raspberry 1 verisi aldığı sürece motorların hepsine geri emri verir
sol/veri	Binary	Butona basıldığı sürece 1 verisi gider , Raspberry 1 verisi aldığı sürece sağdaki motorlara ileri , soldaki motorlara geri emri verir
siren/veri	Binary	Butona basıldığında 1 değeri gönderilir ve tekrar basılana kadar değer sabit kalır.1 verisi geldiği sürece buzzer ve çakar devresi algoritmaya göre siren yapar
pompa/veri	Binary	Butona basıldığında 1 değeri gönderilir ve tekrar basılana kadar değer sabit kalır.1 verisi geldiği sürece pompa çalışarak borudan su tahliyesi sağlanır
servo/veri	Binary	Slider ile 0 ile 180 derece arasında veri gider. Giden veriye göre servo PWM sinyali , dereceyi 0 ile 180 derece olacak şekilde servoyu döndürür.
flame/veri	Float	Bu topic'de , diğer topiclere nazaran subscribe olan tarafa Raspberry değil websitesidir. Raspberry , 2 saniyede bir flame sensörlerinden ölçülen değerleri karşılaştırır ve en düşük değer alan sensör , sensörün baktığı yönü alev olarak algılayarak websitesine ,sensörün yönünü gönderir. Akabinde , arayüzün ilgili bölümünde alevin yönü görselleştirilir.
su/veri	Float	Bu topic'de , diğer topiclere nazaran subscribe olan tarafa Raspberry değil websitesidir. Raspberry , 2 saniyede bir şamandıra sensöründen gelen voltaj diferansiyelini ADS1115 tarafından alır ve algoritma bu veriyi 0 ile 100 arasında yorumlayarak websitesine gönderir. Websitesinde su derecesini gösteren div kısmında su seviyesindeki değişim ve anlık durum javascript sayesinde animasyonlu bir şekilde görselleştirilir.

Tablo 4.3 Topic/İşlev tablosu

4.6 Deneyisel Sonuçlar

Aracın tam işlevli bir şekilde çalıştığından emin olmak için birtakım testler yapılmıştır. Bu testlerin sistemin geliştirilmesine katkıda bulunması amaçlanmıştır.

4.6.1 Web Arayüz Erişilebilirlik Testi

Web arayüzü yerleşik ağda yapılan tüm sistem testlerinden geçtikten sonra ortak ağ'da <http://www.halilibrahimcetin.com.tr> üzerinde yayına alınmıştır ve sistemin tutarlı bir şekilde erişilebilir olduğunu test etmek amacı ile erişilebilirlik testine tabi tutulmuştur. Test sonucu Şekil 4.10'da verildiği gibidir.

```
PS C:\Users\halilibrahim> ping halilibrahimcetin.com.tr

Pinging halilibrahimcetin.com.tr [31.186.11.139] with 32 bytes of data:
Reply from 31.186.11.139: bytes=32 time=7ms TTL=52
Reply from 31.186.11.139: bytes=32 time=7ms TTL=52
Reply from 31.186.11.139: bytes=32 time=6ms TTL=52
Reply from 31.186.11.139: bytes=32 time=7ms TTL=52

Ping statistics for 31.186.11.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 7ms, Average = 6ms
PS C:\Users\halilibrahim> |
```

Şekil 4.10 Web arayüz erişilebilirlik testi sonucu

Şekil 4.10'da görüldüğü üzere sistem stabil bir şekilde paket kaybı olmadan giden verileri almış ve erişilebilir şekildedir.

4.6.2 MQTT Broker Bağlantı Testi

Web arayüzünden gönderdiğimiz verilerin Raspberry Pi tabanlı itfaiye aracına erişebilmesi için VPS üzerinde kurulan MQTT broker'a başarılı ve kesintisiz bir bağlantı yapması beklenmektedir. Sistem bu durumu kendi içinde sürekli kontrol etmekle beraber konsol kayıtlarından da bu bilgilere erişilebilmektedir. Sistem testlerine ait test sonuçları Şekil 4.11 ve Şekil 4.12'de verilmiştir.



Şekil 4.11 Web arayüzü üzerinden MQTT bağlantı testi ve konsol kayıtları

```

root@178-208-187-184:~# sudo tail -f /var/log/mosquitto/mosquitto.log
1748761581: Client auto-E4D6BB95-069B-B5F7-A0BC-D941489EA3C0 closed its connection.
1748762910: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748764711: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748766512: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748768313: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748770114: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748771915: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748773716: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.
1748775410: New client connected from ::ffff:88.238.194.155:35177 as mqttjs_c6c81256 (p2, c1, k60).
1748775517: Saving in-memory database to /var/lib/mosquitto/mosquitto.db.

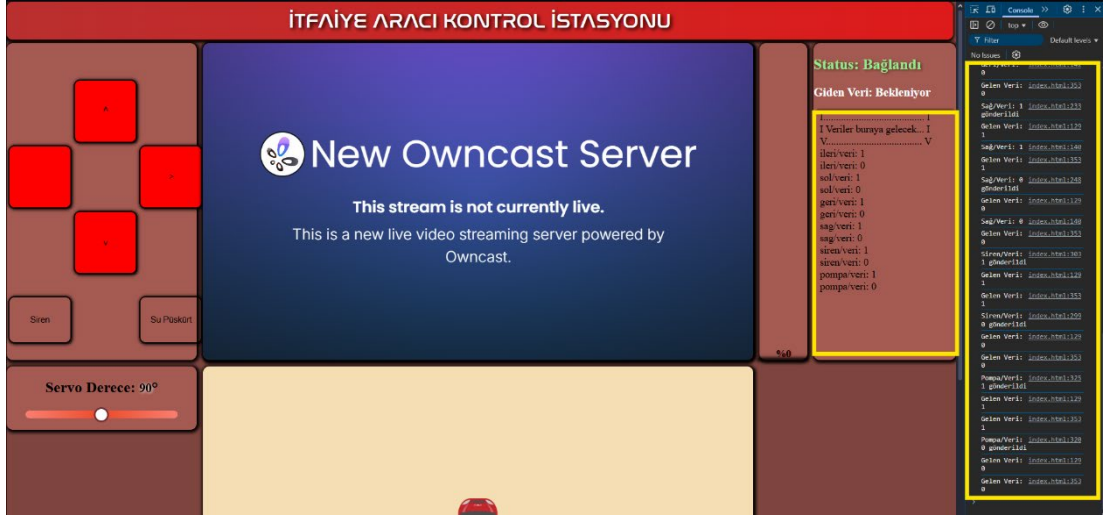
```

Şekil 4.12 MQTT broker client bağlantısı testi

Sistem hem client hem broker üzerinden test edilmiş ve başarılı bir şekilde mqtt bağlantısı yapabildiği sonucuna varılmıştır.

4.6.3 Veri Gönderim Testi

Sistemin sorunsuz şekilde bağlantı kurabildiğinden emin olunduktan sonra arayüz üzerinden broker'a veriler gönderilip bu verilerin ne sürede ve ne doğrulukta gönderildiği test edilmiştir. Sistem performansını doğrudan etkileyen bu test için bir çok topic'ten veri gönderilip broker üzerinden izlenilmiştir. Veri gönderim testine ait sonuçlar Şekil 4.13 ve 4.14'te verilmiştir.



Şekil 4.13 Web arayüzü veri gönderim testi konsol kayıtları

```

root@178-208-187-184:~# mosquitto_sub -v -t '#'
ileri/veri 1
ileri/veri 0
sol/veri 1
sol/veri 0
geri/veri 1
geri/veri 0
sag/veri 1
sag/veri 0
siren/veri 1
siren/veri 0
pompa/veri 1
pompa/veri 0

```

Şekil 4.14 Veri gönderim testi MQTT broker veri alım sonuçları

Testler sonucunda gönderilen veriler başarılı bir şekilde broker'a ulaştığı sonucuna varılmıştır. Ayrıca web arayüzü tasarlanırken başarı ile gönderilen verilerin "veriler" div'inde anlık olarak yayınlanması tasarlandığı için kullanıcıya verilerin iletildiği geribildirimi , sistem tarafından verilmektedir. Dolayısıyla her veri gönderim aşamasında sistem kendini test etmektedir. Sistemin bu mimarisine ait kod parçası Şekil 4.15'te verilmiştir.

```
// Mesaj alındığında
client.on('message', (topic, message) => {
  const veri = message.toString();
  console.log("Gelen Veri:", veri);

  // Topic
  if (topic === 'sensor/veri') {
    console.log("Sensör Verisi:", veri);
    document.getElementById("sensorVerisi").innerText = `${parseFloat(veri).toFixed(2)} °C`;
  } else if (topic === 'ileri/veri') {
    console.log("İleri/Veri:", veri);
  } else if (topic === 'sol/veri') {
    console.log("Sol/Veri:", veri);
  } else if (topic === 'sag/veri') {
    console.log("Sağ/Veri:", veri);
  } else if (topic === 'geri/veri') {
    console.log("Geri/Veri:", veri);
  }
});
```

Şekil 4.15 Sistemin başarılı veri gönderimine ait geri bildirim kısmı

4.6.4 Veri Alım Testi

Sistemin sorunsuz şekilde veri gönderebildiğinden emin olunduktan sonra broker üzerinden gelen verilerin Raspberry Pi'a ne sürede ve ne doğrulukta ulaştığı test edilmiştir. Sistem performansını doğrudan etkileyen bu test için bir çok topic'ten veri gönderilip Raaspberry üzerinden izlenilmiştir. Veri alım testine ait sonuçlar Şekil 4.16'da verilmiştir.

```
Running: deneme.py
İleri gidiliyor
İleri gitme işlemi bitti
Sola dönülüyor
Sola dönülme işlemi bitti
Geri gidiliyor
Geri gitme işlemi bitti
Sağa dönülüyor
Sağa dönülme işlemi bitti
```

Şekil 4.16 Sistemin başarılı veri aldığına dair geri bildirim terminal çıktısı

4.6.5 Motor Fonksiyonellik Testi

Arayüzden gelen ve aracın yönlendirmesini sağlayan verilerin fonksiyonelliği test edilmiştir. Ve algoritmaya göre “ileri/veri , geri/veri , sag/veri , sol/veri” topiclerinden gelen verilerin doğru çalıştığı ve istenilen amaca hizmet ettiği gözlemlenmiştir. Algoritmaya ait detaylar Şekil4.17’de verilmiştir

```

if topic == "sag/veri":
    sagpayload = msg.payload.decode()
    if sagpayload == "1":
        sag_geri.on()
        sag_ileri.off()
        sol_ileri.on()
        sol_geri.off()
        print("Sağa dönülüyor")
    elif sagpayload == "0":
        sag_geri.off()
        sag_ileri.off()
        sol_ileri.off()
        sol_geri.off()
        print("Sağa dönme işlemi bitti")
    else:
        pass

if topic == "ileri/veri":
    ileripayload = msg.payload.decode()
    if ileripayload == "1":
        sag_geri.off()
        sag_ileri.on()
        sol_ileri.on()
        sol_geri.off()
        print("İleri gidiliyor")
    elif ileripayload == "0":
        sag_geri.off()
        sag_ileri.off()
        sol_ileri.off()
        sol_geri.off()
        print("İleri gitme işlemi bitti")
    else:
        pass

if topic == "sol/veri":
    solpayload = msg.payload.decode()
    if solpayload == "1":
        sag_geri.off()
        sag_ileri.on()
        sol_ileri.off()
        sol_geri.on()
        print("Sola dönülüyor")
    elif solpayload == "0":
        sag_geri.off()
        sag_ileri.off()
        sol_ileri.off()
        sol_geri.off()
        print("Sola dönme işlemi bitti")
    else:
        pass

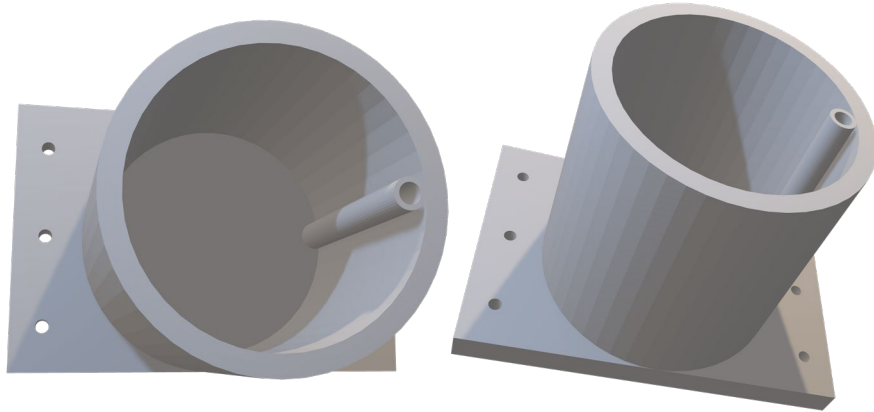
if topic == "geri/veri":
    geripayload = msg.payload.decode()
    if geripayload == "1":
        sag_geri.on()
        sag_ileri.off()
        sol_ileri.off()
        sol_geri.on()
        print("Geri gidiliyor")
    elif geripayload == "0":
        sag_geri.off()
        sag_ileri.off()
        sol_ileri.off()
        sol_geri.off()
        print("Geri gitme işlemi bitti")
    else:
        pass

```

Şekil 4.17 Motor çalışma algoritması

4.6.6 Su Tahliye ve Depo Sızdırmazlık Testi

İtfaiye aracının su deposu tarafımda SolidWorks üzerinden tasarlanmış ve 3B yazıcıda PLA+ filament kullanılarak üretilmiştir. Sızdırmazlık açısından iç kısmı PET malzemesi ile kaplanmıştır. Depo tasarımı Şekil4.18’de gösterilmiştir.



Şekil 4.18 Su deposu 3B tasarımı

Depodan 12V DC motor sayesinde tahliye edilecek su 3mm çapında tahliye borusu sayesinde tahliye edilecektir. Arayüz üzerinden kontrol edilebilirliği test edilmiştir.

Gerekli algoritmaya ait kod parçası Şekil 4.19’da verilmiştir.

```
if topic == "pompa/veri":
    pompapayload = msg.payload.decode()
    if pompapayload == "1":
        print("Pompa Açık")
        pompa.on()
    elif pompapayload == "0":
        print("Pompa Kapalı")
        pompa.off()
else:
    pass
```

Şekil 4.19 Su tahliye pompası kontrol algoritması

4.6.7 Şamandıra Ölçüm ve Algoritma Testi

Su deposunda bulunan su miktarının arayüz üzerinden görülmesi için anlık olarak depodaki su miktarının ölçülmesi gerekmektedir ve bu nedenle şamandıra sistemi kullanılmıştır. Kullanılan şamandıra modülü analog veri gönderemediği ve ayrıca Raspberry Pi’da ADC entegresi olmadığı için ADS1115 ADC üzerinden veri aktarılmıştır. Alınan veri 0 ile 100 arasında sıkıştırılarak depo doluluk yüzdesi arayüze gönderilmektedir. İlgili algoritmaya ait kod detayı Şekil4.20’de verilmiştir.

```
def map_range(x, in_min, in_max, out_min, out_max):
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min

def flame_sensor_and_samandira_loop():
    while True:
        try:
            front_sensor = AnalogIn(ads, ADS.P0)
            right_sensor = AnalogIn(ads, ADS.P1)
            left_sensor = AnalogIn(ads, ADS.P2)
            back_sensor = AnalogIn(ads, ADS.P3)
            samandira_sensor = AnalogIn(ads2, ADS.P0)
            samandiraraw = samandira_sensor.value
            su_seviyesi = max(0, min(100, (samandiraraw / 5500) * 100))
            ##print("Su seviyesi:", int(su_seviyesi))
            client.publish("su/veri", int(su_seviyesi))
```

Şekil 4.20 Şamandıra ile su deposu doluluk oranı algılama algoritması

Yapılan testler sonucunda alınan verinin stabil bir şekilde 0 ile 100 arasında sıkıştırıldığı gözlemlenmiştir.

4.6.8 Flame Sensör Veri Alımı ve Aktarımı Testi

Aracın etrafındaki alevlerin flame sensör aracılığı ile algılanması ve akabinde algoritma ile alevin yönünün belirlenip arayüze iletilmesi işleminin stabil çalışması test edilmiştir. Daha doğru bir sonuç elde edebilmek için flame sensörler analog çıkıştan ADC sayesinde okunmuş ve en az değere sahip olan flamse sensör verisinin yönünde alev algılayan bir algoritma geliştirilmiştir. Algoritmaya ait kod detayları Şekil 4.21.'de verilmiştir.

```
def flame_sensor_and_samandira_loop():
    while True:
        try:
            front_sensor = AnalogIn(ads, ADS.P0)
            right_sensor = AnalogIn(ads, ADS.P1)
            left_sensor = AnalogIn(ads, ADS.P2)
            back_sensor = AnalogIn(ads, ADS.P3)
            samandira_sensor = AnalogIn(ads2, ADS.P0)
            samandiraraw = samandira_sensor.value
            su_seviyesi = max(0, min(100, (samandiraraw / 5500) * 100))
            ##print("Su seviyesi:", int(su_seviyesi))
            client.publish("su/veri", int(su_seviyesi))

            values = [front_sensor.value, right_sensor.value, left_sensor.value, back_sensor.value]

            if front_sensor.value < min(back_sensor.value, right_sensor.value, left_sensor.value):
                payload = "^"
            elif right_sensor.value < min(back_sensor.value, left_sensor.value, front_sensor.value):
                payload = ">"
            elif back_sensor.value < min(front_sensor.value, left_sensor.value, right_sensor.value):
                payload = "v"
            elif left_sensor.value < min(front_sensor.value, back_sensor.value, right_sensor.value):
                payload = "<"
            else:
                payload = "-"
            client.publish("flame/veri", payload)

        except Exception as e:
            print("Sensör okuma hatası:", e)
            time.sleep(1)
```

Şekil 4.21 Flame sensör alev yönü algılama ve arayüze gönderme algoritması

4.6.9 Servo Kontrol Testi

İtfaiyenin su hortumunun yönelimini belirleyen SG90 servo motorun arayüz üzerinden anlık kontrol edilebilirliği test edilmiştir ve yapılan test sonucu servo motorun amacına uygun çalıştığı gözlemlenmiştir. Servo yönlendirmesine ait kod parçası ve test sonucu Şekil 4.22 ve Şekil 4.23'te verilmiştir.

```
def set_servo_angle(angle):
    angle = max(0, min(180, int(angle)))
    servo.value = (angle / 90) - 1
    print(f"Servo açısı: {angle}")

    if topic == "servo/veri":
        try:
            derece = float(payload)
            set_servo_angle(derece)
        except:
            pass
```

Şekil 4.22 Servo çalışma algoritması

```
Servo açısı: 68
Servo açısı: 79
Servo açısı: 87
Servo açısı: 92
Servo açısı: 96
Servo açısı: 97
Servo açısı: 99
Servo açısı: 101
```

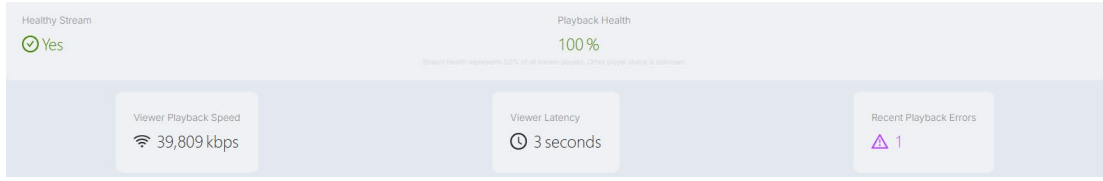
Şekil 4.23 Servo fonksiyonellik testi

4.6.10 Kamera Yayın Testi

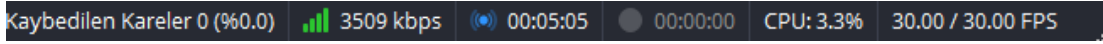
Kamera verisi owncast üzerinden arayüzde yayınlanmaktadır. Owncast OBS üzerinden giden kamera verisinin kare kaybı incelenmiş ve kare kaybına rastlanmamıştır. Test sonucu gecikmenin 3 -5 saniye olduğu gözlemlenmiştir. Gerekli testlere ait sonuçların görselleri aşağıda verildiği gibidir.

```
*** System restart required ***
Last login: Sun Jun  1 11:00:39 2025 from 88.238.194.155
root@178-208-187-184:~# cd owncast
root@178-208-187-184:~/owncast# ./owncast
INFO[2025-06-02T15:06:12Z] Owncast v0.2.3-linux-64bit (1d439a6430377fd792595fae265eede55ed79563)
INFO[2025-06-02T15:06:12Z] Web server is listening on port 8080.
INFO[2025-06-02T15:06:12Z] Configure this server by visiting /admin.
```

Şekil 4.24 Owncast çalışma testi



Şekil 4.25 Yayına ait durum istatistikleri



Şekil 4.26 OBS giden kare durum testi

```
@app.route('/')
def index():
    return render_template_string(HTML_PAGE)

def generate_frames():
    while True:
        frame = camera.capture_array()
        ret, buffer = cv2.imencode('.jpg', frame)
        frame = buffer.tobytes()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    # MQTT düğümünü başlat
    client.loop_start()

    # Sensor izleme threadini başlat
    flame_thread = threading.Thread(target=flame_sensor_and_samandira_loop)
    flame_thread.daemon = True
    flame_thread.start()

    # Flask uygulamasını başlat
    app.run(host='0.0.0.0', port=5000, debug=False)
```

Şekil 4.27 Kamera yayını sağlayan , yerel ağ'a kamera verisini gönderen kod parçası

Görüldüğü üzere sistem ilk başta yerel ağda kamera yayını başlatılır ve bu sayede Raspberry Pi'nin işlem gücü yetersizliğinden kaynaklanan problemin önüne geçmiş olarak bilgisayar üzerinden görüntü yayını başlatılır .Sistemin bu testten de geçer not aldığı gözlemlenmiştir.

5. SONUÇLAR VE ÖNERİLER

Sonuç olarak şasi üzerine dizilen elektronik komponentlerin Şekil4.2’de görüldüğü gibi elektronik kablo bağlantıları yapılmıştır.

Daha sonrasında stickera bastırılan itfaiye araç gövde tasarımları gövdeyi oluşturan suntalara yapıştırılmış ve gövde montajı şasi üzerine yapılmıştır.

Gövde montajı yapılan aracın elektronik motorlarına güç verecek olan 3S Li-Po batarya bir güvenlik önlemi olarak bir kill-switch’e bağlanmış ve sisteme entegresi bu şekilde yapılmıştır. Ayrıca pil herhangi bir olumsuz durumda ulaşılması en kolay yer olan tavan suntasına M3 vidalar sayesinde sabitlenen pil tutucu tasarımına sabitlenmiştir.

Sistemin ana görev bilgisayarı olan Raspberry Pi 5’e gücünü verecek olan powerbank de tavan suntasına powerbank tutucu tasarımı sayesinde sabitlenmiştir.

Montaj sonrası tamamlanan akıllı itfaiye aracı , mqtt bağlantısı testlerinden başarıyla geçmiştir ve sistem tam manası ile istenilen şekilde veri haberleşmesi yapabilmektedir.

Sistemde kamera versisinin arayüze aktarılması için ilk etapta Youtube API kullanılması planlanmıştır fakat güvenlik sebepleri ile gömülü koddaki canlı yayın ID’si sürekli değiştiği için sistemin performansını ve statüğünü bozmaktadır. Bu sebeple halihazırda mqtt haberleşmesini gerçekleştiren VPS üzerinden bir Owncast yayın sistemi kurularak statik ID’li bir canlı yayın sistemine geçilmiştir. Canlı yayında minimum gecikme yaşanması için tüm ayarlamalar dikkatle yapılmasına rağmen 3 -5 saniyelik bir gecikmenin aşığına inilememiştir. Sistem performansını etkilemeyen bir kusur olduğu için göz ardı edilmiştir fakat ilerleyen süreçlerde bu kısım geliştirilebilir.

Akıllı itfaiye aracının fiziksel görünümüne ait görüntüleri Şekil 4.29’da görüldüğü üzeredir.



Şekil 4.28 İtfaiye aracı nihai görünümü

KAYNAKLAR

- [1] The Japan Society of Mechanical Engineers , 2022 , “A firefighting robot system that can be applied to large-scale fires in petroleum complexes” , JSME , <https://www.jsme.or.jp/english/about/awards/2022-2/award2022-nt4> (Mart 2025)
- [2] Smart Firefighting Initiatives. (2024). *Smart Firefighting Technology*. <https://www.smartfirefighting.org/> (Haziran 2025).
- [3] MIT Robotics & IoT Lab. (2024). *Web Controlled Robot Car Projects*. Eriřim adresi: <https://web.mit.edu/robotics-iot-lab/> (Haziran 2025).
- [4] Raspberry Pi Foundation 15 Ekim 2021 , “ Raspberry Pi Documentation” , raspberrypi.com, <https://www.raspberrypi.com/documentation/> (Mart 2025)
- [5] İvam Novikov , 26 Kasım 2021 , “ What is MQTT ? All you need to know.” , Medium.com , <https://d0znpp.medium.com/what-is-mqtt-all-you-need-to-know-cda591a8586> (Mart 2025)
- [6] ÇETİN Halil İbrahim , 9 Mart 2025 , “IoT Tabanlı Otomobil Kontrol ve Takip Uygulaması” , Github , <https://en.wikipedia.org/wiki/HTML> (Nisan 2025)
- [7] Davemck , 6 Haziran 2016 , “HTML” , Wikipedia:Özgür Ansiklopedi , <https://en.wikipedia.org/wiki/HTML> (Nisan 2025)
- [8] Video Streaming Protocols. (2024). *MJPEG and RTMP Streaming*. Eriřim adresi: <https://www.videostreaming.com/protocols/> (Haziran 2025).
- [9] K. B. Krishna. (2017). IoT Based Vehicle Tracking and Security System. *International Journal of Engineering and Technology*, 9(5), 3747-3751.
- [10] R. S. Kulkarni. (2019). Smart Car System Using IoT for Safety and Convenience. *International Journal of Computer Applications*, 178(39), 23-28.