

**T.C**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**MEKATRONİK MÜHENDİSLİĞİ**

**UZAKTAN KONTROLLÜ ÇİZGİ İZLEYEN**

**TAŞIMA ARACI**

**Ümit DEMİR**

**Oğuzcan ÜNLÜ**

**Dr. Öğr. Üyesi İsmail TEMİZ**



**T.C**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**MEKATRONİK MÜHENDİSLİĞİ**

**UZAKTAN KONTROLLÜ ÇİZGİ İZLEYEN**

**TAŞIMA ARACI**

**Ümit DEMİR**

**(171215504)**

**Oğuzcan ÜNLÜ**

**(171216822)**

**Dr. Öğr. Üyesi İsmail TEMİZ**

**T.C**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği öğrencisi Ümit DEMİR ve Oğuzcan ÜNLÜ ’nün “UZAKTAN KONTROLLÜ ÇİZGİ İZLEYEN TAŞIMA ARACI” başlıklı bitirme projesi çalışması 22/06/2019 tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

[Dr. Öğr. Üyesi İsmail](http://abys.marmara.edu.tr/huseyin.yuce/) TEMİZ (Danışman)

Marmara Üniversitesi (İMZA)

………………………………………….

Marmara Üniversitesi (İMZA)

………………………………………….

Marmara Üniversitesi (İMZA)

**İÇİNDEKİLER**

[**İÇİNDEKİLER i**](#_bookmark0)

[**ÖZET iii**](#_bookmark1)

[**KISALTMALAR iv**](#_bookmark2)

[**ŞEKİL LİSTESİ v**](#_bookmark3)

[**TABLO LİSTESİ vii**](#_bookmark4)

1. [**BOLUM: GİRİŞ VE TEZİN AMACI 1**](#_bookmark5)
   1. [Endüstriyel Önemi 3](#_bookmark6)
   2. [Kullanım Alanları 5](#_bookmark7)
   3. [Literatür Özeti 5](#_bookmark11)
   4. [Tezin Amacı 6](#_bookmark12)
2. [**BOLUM: SİSTEMİN TASARIMI 7**](#_bookmark13)
   1. [Tasarımın Amacı 8](#_bookmark14)
   2. [Tasarımın Detayları 8](#_bookmark15)
   3. [Mekanik Tasarım 15](#_bookmark18)
   4. [Yazılımsal Tasarım 15](#_bookmark22)
      1. [İstemci 17](#_bookmark23)
      2. [Sunucu 18](#_bookmark24)
      3. [OTA (Over the Air) 19](#_bookmark25)
   5. [Elektronik Tasarım 21](#_bookmark26)
   6. [Haberleşme Tasarımı 22](#_bookmark30)
   7. [Tasarımda Kullanılan Malzemelerin Özellikleri 24](#_bookmark30)
   8. Bu [Bölümden Elde Edilen Çıktılar 27](#_bookmark42)
3. [**BOLUM: SİSTEM UYGULAMASI VE SONUÇLARI 28**](#_bookmark43)
   1. [Bağlantı Denemeleri 29](#_bookmark44)
   2. [Mekaniksel Uygulama 36](#_bookmark47)
   3. [Bölümden Çıkarılan Sonuçlar 38](#_bookmark48)
4. [**BOLUM: ÇALIŞMANIN GENEL SONUCU 38**](#_bookmark49)
   1. [Veri Akış Blok Şeması 38](#_bookmark44)
   2. [Çalışma Şeması 39](#_bookmark47)
   3. [Yazılımsal Akış Şeması 39](#_bookmark48)
   4. [Sonuç 40](#_bookmark48)
5. [**KAYNAKÇA 41**](#_bookmark50)
6. [**EKLER 43**](#_bookmark51)

[EK-1: Açık Kaynak Kodu 43](#_bookmark52)

1. [**ÖZGEÇMİŞ 53**](#_bookmark51)

**ÖZET**

Yapılan bu çalışmada, endüstriyel lojistik faaliyetlerinde kullanılabilecek, önceden belirlenmiş üretim alanlarında taşıma işlevine sahip, belirli bir rotası olan, Nesnelerin İnterneti sayesinde birbirleri ile kablosuz olarak haberleşebilen, çalıştığı ortamdaki insanların iş güvenliğini riske atmayan küçük ölçekli bir çizgi takip edebilen taşıma aracı prototipi tasarlanması hedeflenmiştir.

**Anahtar Kelimeler**: MQTT, Nesnelerin İnterneti, Internet of Things, Çizgi İzleyen Robot

**KISALTMALAR**

|  |  |
| --- | --- |
| **ASP** | :Etkin Sunucu Sayfaları |
| **CC** | **:**Bulut Bilişim |
| **ERP** | **:** Kurumsal Kaynak Planlaması |
| **DC** | **:**Doğru Akım |
| **HTTP** | **:**Hyper Text Transfer Protocol |
| **FTP** | : Dosya Transfer Protokolü |
| **IIoT** | **:**Indistrial Internet of Things |
| **IoT** | **:**Internet of Things |
| **J2EE** | :Java 2 Enterprise Edition |
| **J2ME** | :Java 2 Micro Edition |
| **MQTT** | **:**Message Queuing Telemetry Transport |
| **OTA** | :Over the Air |
| **PLC** | : Programmable Logic Controller |
| **RFID** | **:**Radyo Frekansı ile Tanımlama |
| **SCADA** | : Supervisory Control and Data Acquisation |
| **TCP/IP** | **:**İletim Protokolü Kontrolü / İnternet Protokolü |
| **Vb.** | **:**Ve benzeri |
| **Wi-Fi** | **:**Wireless Fidelity |
| **WWW** | **:**World Wide Web |

**ŞEKİL LİSTESİ**

Şekil 1.1- Nesnelerin İnterneti Katmanları 2

Şekil 1.2 - Endüstride IoT 4

Şekil 2.1 - Sistemin Tasarımı 7

Şekil 2.2 - Optik Sensör Çalışma Prensibi 9

Şekil 2.3 – Optik Sensör ile Çizgi Takip Eden Robotta Ana Bölümler 10

Şekil 2.4 – Optik Algılama Sensörleri 10

Şekil 2.5 – Diferansiyel Sürüş Yönteminde Motorların Hareket Verme Prensibi 13

Şekil 2.6 - Aracın Şase ve Motorlarının Üstten Görünümü 15

Şekil 2.7 – App Inventor 2 Programında Butonların Arduino Koduna Tanımlanması 22

Şekil 2.8 – Android Program Ekran Görüntüsü 23

Şekil 2.9 – Fırçalı DC Motor 25

Şekil 2.10 – L298N Motor Sürücü Kartı 25

Şekil 2.11 – HC-SR04 Ultrasonik Mesafe Sensörü 26

Şekil 2.12 – Pil Yuvası 26

Şekil 3.1 – Projenin Fritzing Şeması 28

Şekil 3.2 – Adım 1 29

Şekil 3.3 – Adım 2 30

Şekil 3.4 – Adım 3 30

Şekil 3.5 – Adım 4 31

Şekil 3.6 – Adım 5 32

Şekil 3.7 – Adım 6 32

Şekil 3.8 – Adım 7 33

Şekil 3.9 – Adım 8 34

Şekil 3.10 – Adım 9 35

Şekil 3.11 – Adım 10 35

Şekil 3.12 – Aracın Son Hali 36

Şekil 3.13 – Araç Pisti 36

Şekil 4.1 – Veri Akışı Blok Şeması 38

Şekil 4.2 – Tasarım Çalışma Şeması 39

Şekil 4.3 – Yazılımsal Akış Şeması 39

**TABLO LİSTESİ**

Tablo 2.1 – Çizgi Takip Yöntemlerinin Karşılaştırılması 11

Tablo 2.2 – Robot Sürüş Yöntemlerinin Karşılaştırılması 14

Tablo 2.3 – Sunucu CPU Kullanım Oranları 17

Tablo 3.1 – Hız-Zaman Grafiği 37

# BÖLÜM I

## Giriş ve Tezin Amacı

Günümüz değişen ve sürekli gelişen iş hayatında taşıma işlemlerinin insan gücüne gerek kalmadan yapılması artık olağan bir hale geldi. Artık ofis, ev, lokanta, işyeri, kafe ve fabrika tarzı ortamlarda eşya, araç, bilgi paylaşımını, herhangi bir insan gücü gerekmeksizin internet ve akıllı cihaz yardımıyla bir yerden bir yere nâkilini sağlamak daha basit hale geldi. Bu noktada İoT (İnternet of Things) yardımımıza koştu. Yapmış olduğumuz projemizde gömülü bir sistem olan ESP32 modülünü kullanarak bir mekânda belirlemiş olduğumuz yörüngede, belirlemiş olduğumuz bölgelerde akıllı cihazlarla bir yerden bir yere taşıma işlemini gerçekleştirdik. Projede ESP32 modülüne, istenen işlevleri yerine getirebilmesi ve yazılımının oluşturulması için Arduino IDE programı ve bu programa ait olan yazılım dili kullanılmıştır.

Nesnelerin İnterneti kavramı, 1999 yılında Radyo Frekansı ile Tanımlama (RFID) teknolojisinin P&G firması için sağladığı yararlarla ilgili bir sunumda Kevin Ashton tarafından ilk kez kullanılmıştır [1]. Ancak tarihteki ilk Nesnelerin İnterneti uygulaması, 1991 yılında Cambridge Üniversitesi'ndeki bir grup akademisyen tarafından kameralı bir sistem ile bir kahve makinesinin görüntülerinin internet üzerinden paylaşılmasıdır [2]. Bu sistem 22 Ağustos 2001 yılına dek kullanılmıştır [3].

Şekil 1.’de nesnelerin interneti organizmasının katmanları gösterilmiştir. IoT alanında henüz standart bir mimari yoktur. Literatürde çok sayıda farklı IoT mimari çerçevesi de bulunmaktadır. IEEE P2423 gibi henüz üzerinde çalışılan bir standart vardır. Şekil 1’de verilen katmanlar yaptığımız bu çalışmada ortaya çıkardığımız kendimize özgü bir çalışmadır. Nesnelerin interneti çekirdeği doğal çevreyi ve sıcaklık, konum, ağırlık, ışık şiddeti, nabız sayısı, tansiyon, sertlik, karbondioksit oranı, nem, ph değeri, ses şiddeti vs. gibi fiziksel büyüklüklerin bulunduğu ortamı ifade eder. Çekirdek katmanında her türlü ölçülebilir büyüklük ham halinde bulunur. Cihaz katmanında bu ham veriler algılanarak analog veya sayısal sinyallere dönüştürülürler. Doğadan alınan bu verilerin işlenmesi için iletilmesi gerekmektedir ve insan-makine, makine-makine iletişimi için gerekli olan RFID, Zigbee, 802.5.4, NFC, Kızılötesi, Bluetooth, ModBus, M-Bus, GPRS ve GSM, Bacnet, LPWAN, elektrik hattı taşıyıcıları, Ethernet gibi kablosuz ve kablolu iletişim altyapısı ve iletişim protokolleri iletişim katmanında yer alır. Daha sonra bu veriler ilgili iletişim protokolleriyle bilinç olarak adlandırılan veri işleme merkezine gönderilirler. Burada küçük çaptaki veri işleme işlemleri gömülü sistemler ile gerçekleşir. Daha büyük uygulamalarda ise bu veriler depolanmak üzere bulut bilişim sistemlerine iletilir. Burada depolanan veriler artan yığınlar halinde büyük veriyi oluştururlar. Verimliliğin arttırılabilmesi için bu büyük miktardaki verinin analiz edilmesi gerekmektedir ve bu da makine öğrenimi yöntemleri kullanılarak gerçekleştirilir. Güvenlik, gizlilik, kimliklendirme, tanıma gibi işlemler bu katmanda gerçekleştirilir [4].



**Şekil 1.1-** Nesnelerin İnterneti Katmanları

Nesnelerin interneti “nesnelerin kendi aralarında oluşturduğu, benzersiz bir şekilde adreslenmiş, dünya çapında bir ağda belirli bir protokol ile birbirleriyle kablosuz veya kablolu bağlantılarla iletişim halinde olmaları” şeklinde tanımlanmıştır. Ayrıca “Çeşitli haberleşme protokolleriyle birbirleri ile haberleşen ve birbirlerine bağlanıp bilgi alışverişi yapabilen akıllı bir ağ oluşturmuş aygıtlar sistemi” olarak da tanımlanabilir. Bu ağda bulunan aygıt ve algılayıcılar makine-makine ve insan-makine iletişimi kurabilen organizmalardır. Bu aygıtlar birçok algılayıcı ve çok sayıda kontrol sistemlerinden oluşan ağ ortamlarından oluşur. Algılayıcılardan gelen veriler bir merkezde toplanarak Büyük Veriyi (Big Data) meydana getirir ve bu veriler Bulut Bilişim (CC) diye adlandırılan sistemlerde depolanabilir. Daha sonra veriler işlenerek verimliliğin ve üretkenliğin artması hedeflenir. Nesnelerin internetinin amacı her zaman her yerden nesnelerin internete bağlanmasıdır [5] .

Bu sayede kolaylıkla çevremizdeki fiziksel olayların kontrol edilmesi ve bu olaylardan elde edilen veriler sayesinde her an nesnelerle iletişim halinde olmamız sağlanmaktadır. Bu fiziksel olaylar taşımacılık, alışveriş, akıllı evler gibi alanlarda ölçülebilir büyüklükler olabilir.

### Endüstriyel Önemi

Çizgi Takip Eden Robotlar günümüzde gelişen teknolojiyle birlikte endüstriyel alanda bir süredir kullanılmaktadır. Genellikle gelişen sanayinin oluşturduğu büyük sanayi kuruluşlarında lojistik ve otomasyon bölümleri içerisinde fazlaca gereksinim duyulan niteliksiz insan gücü ile yapılan taşıma işlemlerini bir süredir çizgi takip eden robotlar yapmaya başlamışlardır [6].

Bu tezde yapılan prototipte öğrenilen tasarım, yazılım ve kontrol sistemleri gibi tecrübeler büyük ölçekteki robotlarda ve endüstriyel uygulamalarda da kullanılabilmektedir. Çizgi Takip Eden Robotlar gelecekte sanayinin lojistik ve taşıma problemlerine çözüm getirerek endüstriyel anlamda otomasyon sistemlerinde kullanılmaya başlayacaktır.

IoT sensörler, aktüatörler, kontrol sistemleri ve makine ağı ile endüstride üretim ve tedarik zinciri ağlarının gerçek zamanlı optimizasyonu ile ilgilenir. Proses endüstrilerinde gelişmiş verimliliği ve güvenli dağıtım sistemini elde etmek için dijital kontrolörler kullanarak proses kontrollerini, hizmet bilgi sistemlerini ve operatör araçları otomatik hale getirir. Nesnelerin intertenin endüstrideki görevleri aşağıdaki gibidir [7].

► Gerçek zamanlı izleme ve süreçlerin kontrolü

► Özel iletişim ve internet teknolojileri ile akıllı makineleri, akıllı sensörleri, akıllı denetleyicileri görevlendirme

► Yüksek hassasiyetli otomasyon ve kontrol sayesinde güvelik, güvenilirlik ve güvenilebilirliği en üst seviyeye çıkartmak

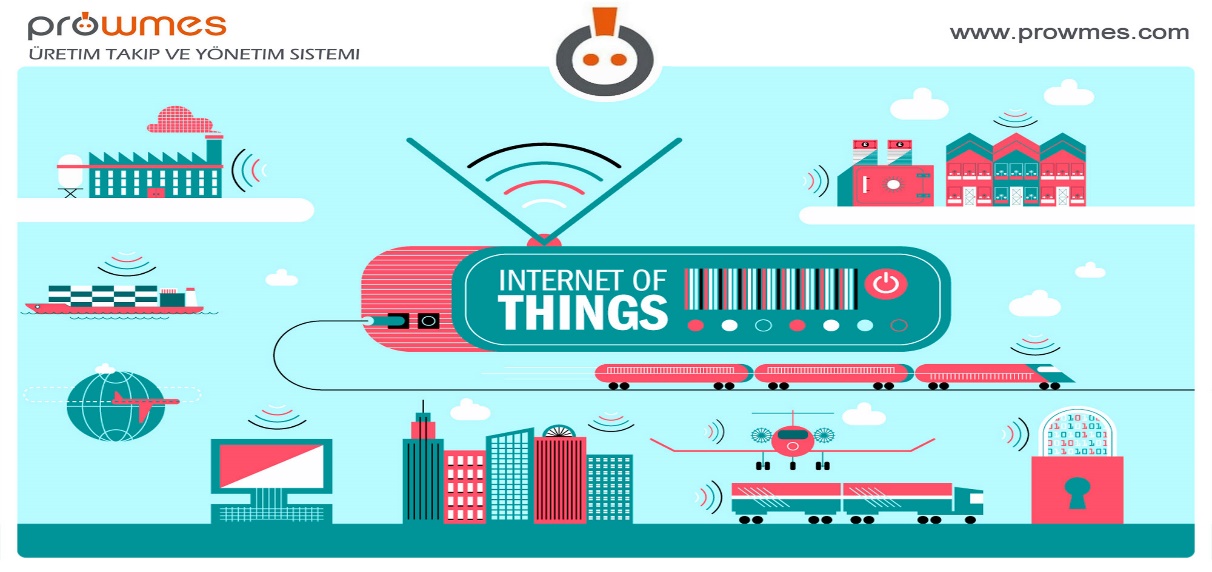
Nesnelerin internetinin mevcut endüstriyel otomasyon sistemleriyle birlikte kullanılmasının önemli avantajları vardır [8].

• İnternete bağlı akıllı üretim cihazları ağ üzerinden birbirleri ile iletişim kurarak üretimi kontrol eder ve insan etkisini en az seviyeye indirir

• Hatalar önceden tahmin edilerek elektriksel ve mekanik arızalar sebebiyle üretimde durma süreleri azaltılabilir

• Üretimde stok takibi otomatik gerçekleştirilerek ham madde eksikliği hızlı bir şekilde giderilir.

• Yöneticiler üretim hattı ve arızalarla ilgili bilgileri dünyanın her yerinden istediği zamanda gerçek zamanlı olarak alabilir ve bu bilgileri istendiği takdirde müşteriler ile paylaşabilir.



**Şekil-1.2** Endüstride IoT

### Kullanım Alanları

-Endüstride

-Enerji Sektöründe

-Medikal ve Sağlık Sistemlerinde

-Ulaşımda

-Çevre Analizinde

-Fabrikalarda

-Ofislerde

### Literatür Özeti

Bu tez çalışmasında; öncelikli olarak literatürde bulunan çizgi takip eden robot projeleri incelenmiş, daha sonra endüstriyel anlamda kullanılabilecek, taşıma bölgelerini tanıma özelliğine sahip, istasyonlar ile kablosuz haberleşme sayesinde iletişim halinde olabilen ve temel görevi olan çizgi takip etme işlemini devam ettirebilecek bir robot tasarım üzerine çalışılmıştır. Bu noktada çizgi takip etme işleminin hem endüstriyel ortamlarda çözüm amaçlı fikirler verebilmesi, hem de yazılımsal anlamda ihtiyaçları karşılayabilmesi ve yeni sistemleri destekleyebilmesi amaçlanmıştır [6].

Tasarım ve programlama esnasında ortam koşullarından meydana gelebilecek iş güvenliği hassasiyetleri de analiz edilmiştir. Prototip çalışmasında vizyon olarak öngörülen çözümler büyük ölçekli robotlarda fikirsel ve yöntemsel olarak uygulanabilir şekilde geliştirilmiştir.

IoT’nin temelleri 1990’ların başında Weiser, (1993) tarafından önerilen “Her Zaman Her Yerde Hesaplama (Ubiquitious Computing)” kavramına dayanır. Nesnelerin interneti terimini ilk defa kullanan kişi ise MIT RFID araştırma grubunda yer alan Ashton, (2002) olmuştur. Fabrikalarda özellikle üretim alanlarında RFID etiketi taşıyan çok fazla miktardaki malzemeler süreç yönetiminde çok büyük kolaylık sağlarlar. RFID okuyucularla otomatik olarak alınan bilgiler veri ağı ortamında saklanırken başka bir akıllı sistemi harekete geçirecek farklı bir süreci başlatabilir. Süreçler arasındaki ilişkiler Kurumsal Kaynak Planlaması (ERP) sisteminde yeni sipariş emirleri, cihaz durumu, üretim hattındaki gecikmeler gibi güncellenmiş bilgilere ışık tutar. Algılayıcı cihaz donanımı içinde bulunan mikro kontrol yongaları ve gömülü yazılımlar giderek daha çok fonksiyonun daha güvenilir olarak yerine getirilebilmesini sağlamaktadır [9]. IoT’de kullanılan akıllı algılayıcı cihazlar kendilerini tanıtma, ağ kurabilme ve topladıkları bilgileri depolama ve analiz yeteneğine sahip genel kullanıma açık bulut servislerine aktarma yeteneğine sahiptirler [10].Kullanıcıların bu servislere ulaşımı ve istedikleri analiz sonuçlarını alabilmeleri kolay kullanımlı web servisleri aracılığıyla gerçekleştirilmektedir. IoT konusundaki hızlı gelişmeler üretim kalitesi ve verimliliğini artırmaya çalışan şirketlerin ilgisini çekmiştir [11].Üretimde kablolu ağlar üzerinden haberleşen algılayıcı birimler, programlanabilir sayısal kontrol cihazları (PLC) ve gömülü sistemler kullanılmaktadır. Bu sistemler bir kontrol merkezinden SCADA (Supervisory Control and Data Acquisation) veya DCS (Distributed Control Systems) sistemleri ile kontrol edilirler ve büyük ölçüde yerel ağ sistemlerinden veya internetten bağımsızdırlar [12].

### Tezin Amacı

Bu tezin amacı; endüstriyel lojistik faaliyetlerinde kullanılabilecek, önceden belirlenmiş üretim alanlarında taşıma işlevine sahip, belirli bir rotası olan, IoT sayesinde birbirleri ile kablosuz olarak haberleşebilen, çalıştığı ortamdaki insanların iş güvenliğini riske atmayan küçük ölçekli bir çizgi takip edebilen taşıma aracı prototipi geliştirilmesidir.

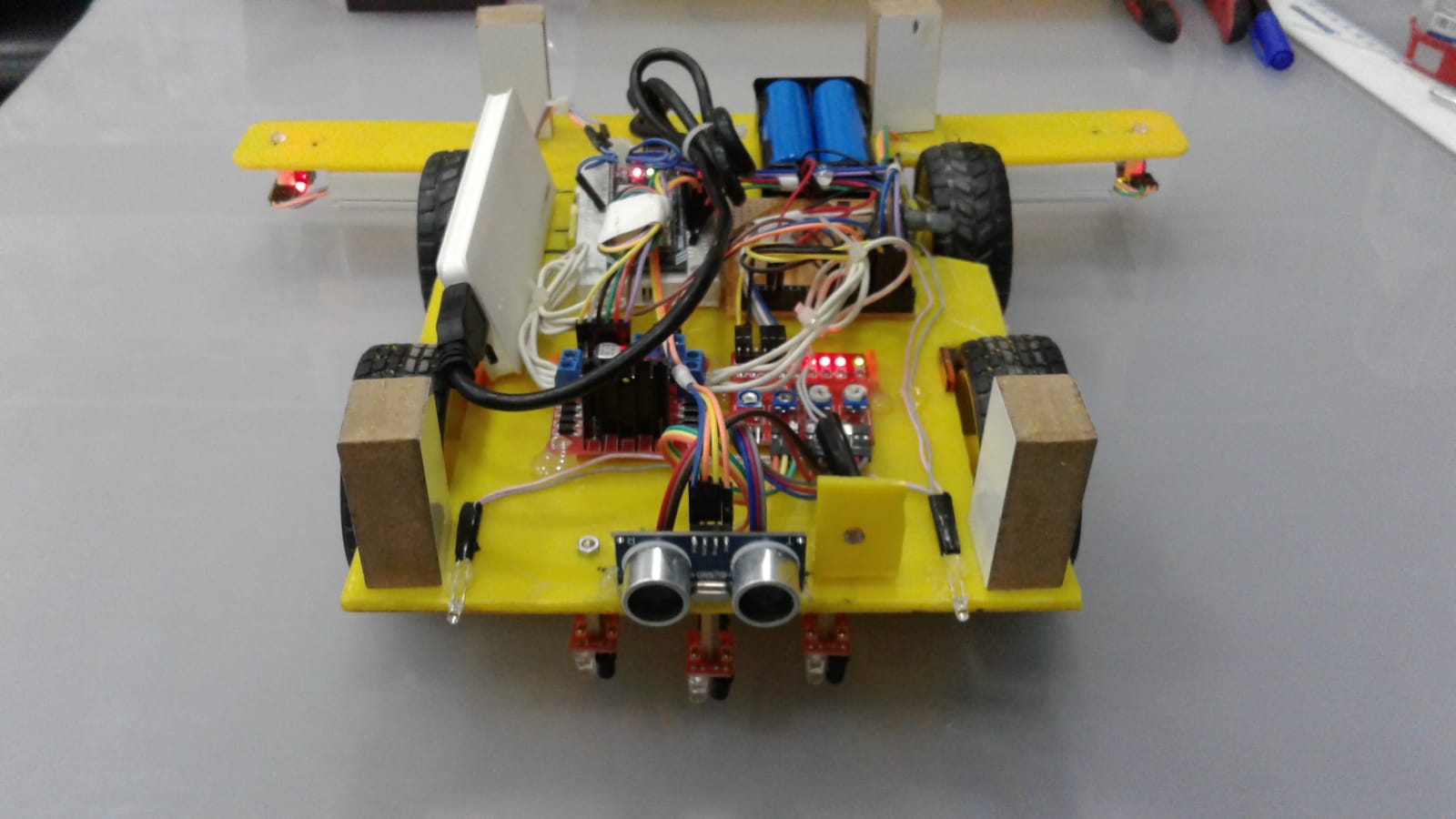
Burada asıl amaç ise; fabrika ve ofis gibi yerlerde insan gücünü minimuma indirgeyerek sistem sayesinde çalışanlar için zamandan ve emekten tasarruf sağlamaktır.

# BÖLÜM II

## Sistemin Tasarımı

Bu sistemde;

* 1 adet araç şase gövdesi,
* 1 adet ESP32 Modülü,
* 4 adet DC motor,
* 1 adet pil yuvası,
* 1 adet HC-SR05 ultrasonik mesafe sensörleri,
* 1 adet L298N çift kanallı motor sürücü kartı,
* 1 Adet Çizgi İzleyen Sensör Seti,
* Diğer ekipmanlar(jumper kablo, bağlantı için USB kablo gibi) kullanılmıştır.



**Şekil-2.1** Sistemin Tasarımı

### Tasarımın Amacı

Burada fayda sağlanmak istenilen konu; fabrika ve ofis gibi yerlerde insan gücünü minimuma indirgeyerek sistem sayesinde çalışanlar için zamandan ve emekten tasarruf sağlamaktır. Bu tasarımda bir otonom aracın hareket ettirilmesi ve bölge kontrolünün yanı sıra; HC-SR04 ultrasonik mesafe sensörleri ile gelen veri doğrultusunda aracın önüne bir engel veya insan çıktığında kazaları önlemek ve iş güvenliğini maksimum seviyeye çıkarmaktır.

### Tasarımın Detayları

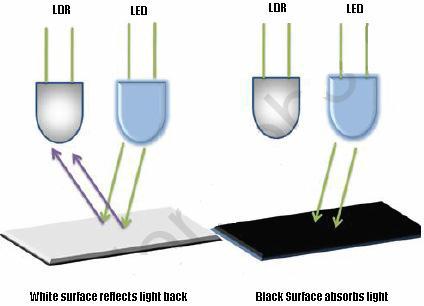
**Çizgi Takip Yönteminin Belirlenmesi**

Günümüz sanayiinde yaygın olarak kullanılan çizgi takip yöntemleri incelendiğinde karşımıza genellikle ihtiyaca yönelik veya robotun kullanılacağı çevre koşullarına uygun şekilde tasarımlar meydana getirilmiştir [13]. Temel olarak incelendiğinde 4 takip yönteminin yaygın olarak kullanıldığını görüyoruz [14]. Bunlar,

1. CCD Kamera Yöntemi ile Önceden Tanımlı Çizgi Takibi
2. Manyetik Algılama Yöntemi ile Önceden Tanımlı Çizgi Takibi
3. Ultrasonik Algılama Yöntemi ile Önceden Tanımlı Çizgi Takibi
4. Optik Algılama Yöntemi İle Önceden Tanımlı Çizgi Takibi

Biz bu projede Optik Algılama Yöntemi İle Önceden Tanımlı Çizgi Takibi’ ni kullandık. Şimdi aşağıda bu yöntemi anlatacağız.

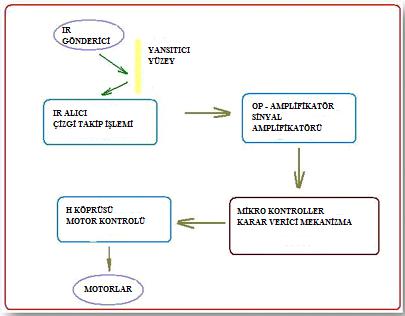
Optik algılama sensörleri ile çizgi takip yöntemlerinde kullanılan temel özellik kontrast farkından kaynaklanan ön tanımlı çizgiyi algılayabilen sensörlerin kullanımıdır [15]. Bu sensörler LDR ve LED özellikleri olan ve kontrast farkını bu iki ekipmanı bir arada barındırması sayesinde anlayabilen sensörlerdir [16].



**Şekil-2.2** Optik Sensör Çalışma Prensibi

LED den gönderilen optik sinyaller şekil 4.’de görüldüğü gibi siyah zemin üzerinde veya kontrast farkının olmadığı zeminlerde tamamen absorbe edilmektedir. Ancak kontrast farkının olduğu veya siyah zemin üzerinde beyaz çizgilerden oluşan bir parkurda LDR dan gönderilen sinyaller LED tarafından algılanmakta ve bu sayede çizgi takip işlemi gerçekleşmektedir [17].

Optik sensörlerle gerçekleşen çizgi takip etme yöntemlerinde diğer çizgi takip etme yöntemlerinde olduğu gibi alınan verilerin işlenmesi ve motorlara gönderilmesi aynı yöntemle olmaktadır. Şekil 5.’de gösterildiği gibi optik algılama sensörlerinden gelen sinyaller veri işleme biriminde yani robotun merkez işlemcisinde işlenmektedir. Sensörlerden gelen veriler robotun hareket yapısını oluşturacağından işlemciden motorlara hareket vermek üzere motor sürücü devresine merkez işlemci birimi tarafından gönderilmektedir.

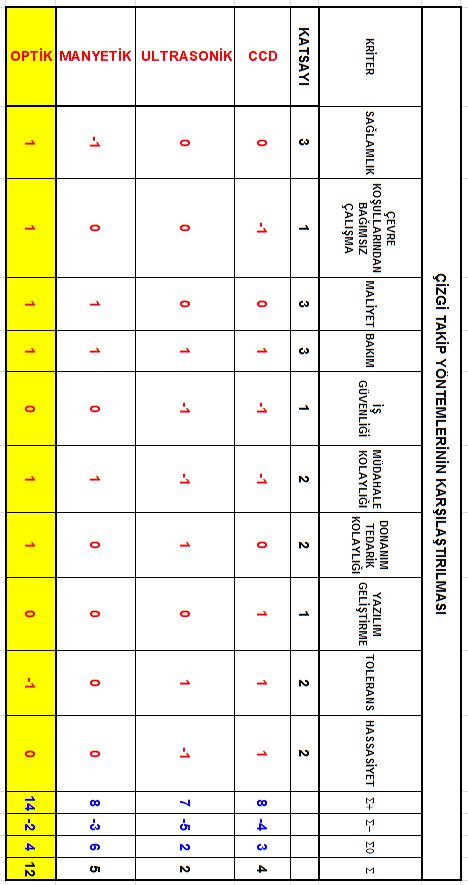


**Şekil-2.3** Optik Sensör ile Çizgi Takip Eden Robotta Ana Bölümler

Optik sensörler ile çizgi takip işlemi incelendiğinde; malzeme bulunabilirliği ve tedariği bakımından oldukça uygun, düşük maliyetli, endüstriyel çalışma koşullarından etkilenmeyen ancak hassasiyet bakımından ince çizgi algılamada düşük bir performansa sahip bir sensör yapısı görülmektedir. QTR sensörlerinin algılama mesafeleri led girişine bağlanan direnç değerleriyle değişkenlik göstermekte ve minimum 1mm maksimum 10 mm ile tanımlanmaktadır bu da çizgi algılama hassasiyetini belirli seviyede tutmaktadır. Endüstriyel ortamlarda çizgi takip etme işleminde çizgi kalınlıklarının istenildiği boyutlarda ayarlanabilir olması bu dezavantajla karşı karşıya kalma durumunu oldukça düşürmektedir.



**Şekil-2.4** Optik Algılama Sensörleri



**Tablo-2.1** Çizgi takip yöntemlerinin karşılaştırılması

**Tekerlekli Robotlar için Sürüş Yönteminin Belirlenmesi**

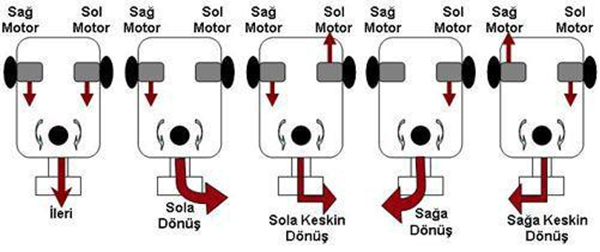
Tekerlekli robotların; hareket edecekleri ortam ve ortam şartları, sahip oldukları güç depolama birimleri, yapacakları iş ve kontrol mekanizmaları göz önüne alınarak uygulamaya en uygun hareket sistemleri belirlenmelidir [18]. Görüldüğü üzere hareket sisteminin belirlenmesinde rol oynayan birçok etken vardır. Hareket sistemi seçimi sırasında, sistemin genel yapısının yapılmak istenilen işi karşılayacak, pratik ve karmaşık olmaması beklenir.

Mevcut projemizde kullanılmak üzere değerlendirilen ve analiz edilen bazı sürüş teknikleri aşağıdaki verilmiştir.

1. Ackerman Sürüş yöntemi
2. Senkron sürüş yöntemi
3. Üç tekerlekli sürüş yöntemi
4. Diferansiyel sürüş yöntemi

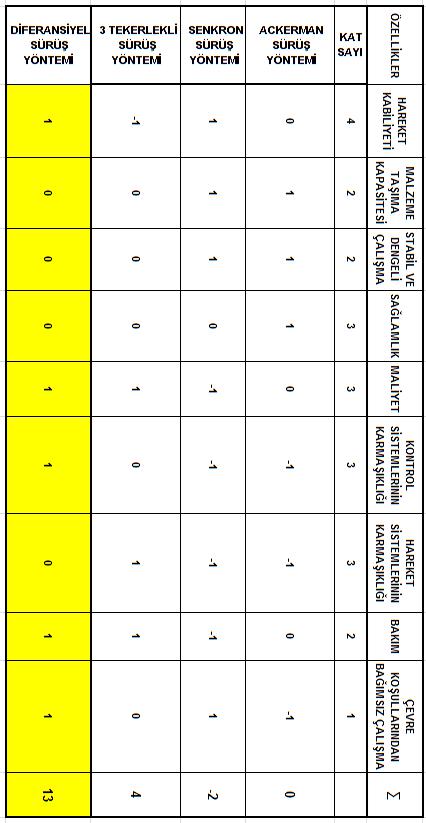
Biz bu projede Diferansiyel sürüş yöntemi ’ni kullandık. Şimdi aşağıda bu yöntemi anlatacağız.

Diferansiyel sürüş tekniği, yabancı literatürde genellikle yanlış olarak “differential drive system” olarak kullanılmaktadır. “differantial drive system” otomotiv mühendisliğinde, robotlarda kullanılandan farklı olan bir çeşit sürüş sistemine karşılık gelmektedir. Robotik uzmanları bu sürüş tekniği için “differential steering system” Kullanmaya özen göstermektedirler [19]. Bina içi gezgin robot uygulamalarında en sık kullanılan sürüş tekniklerinden biridir. Diferansiyel sürüş, günlük hayatımızdan da alışık olduğumuz bir tekniktir. Tekerlekli sandalyelerde uygulanan mantık ile aynıdır. İki birbirinden bağımsız teker aynı dingil üzerine oturtulmuştur. Tekerler birbirinden ayrı hızlarda dönebildikleri için; iki tekerlek aynı hızda döndüğü takdirde düz ilerleme hareketi gerçekleştirilir. Bu yöntem dar alanda manevra kabiliyetini arttırmaktadır. Diferansiyel sürüşte robotun iki tarafında bir birinden bağımsız hareket ettirilen iki motor bulunur. Bu motorların bağlı olduğu tekerlekler dışında, dengeyi sağlamak amacı ile ön ve arka kısımlara kastor (castor) tekerlekler veya roll on tekerler [20]. konulabilir. Birbirinden bağımsız iki ayrı motora bağlı yan tekerlekler ile robotun yönü ve hızı tayin edilir. Tekerlekler aynı yönde ve aynı hızda hareket ettiği takdirde, robot düz bir şekilde ilerleyecektir. Ve bu hız aşağıda denklemlerde verileceği gibi sağ ve sol motor hızlarının toplamının yarısı kadar olacaktır. Sağ ve sol motor hızlarının farklı değerlerde olması durumunda ise, robot hızı robotun orta noktasına göre yine sağ ve sol motorların toplamının yarısına eşit olduğu halde, robot yönünde değişiklik olacaktır. Bu yön değişimi şekil 7.’de görüldüğü gibi daha az dönen motor yönündedir ve iki motor arasındaki hız farkı ile doğru orantılıdır.



**Şekil-2.5** Diferansiyel Sürüş Yönteminde Motorların Hareket Verme Prensibi

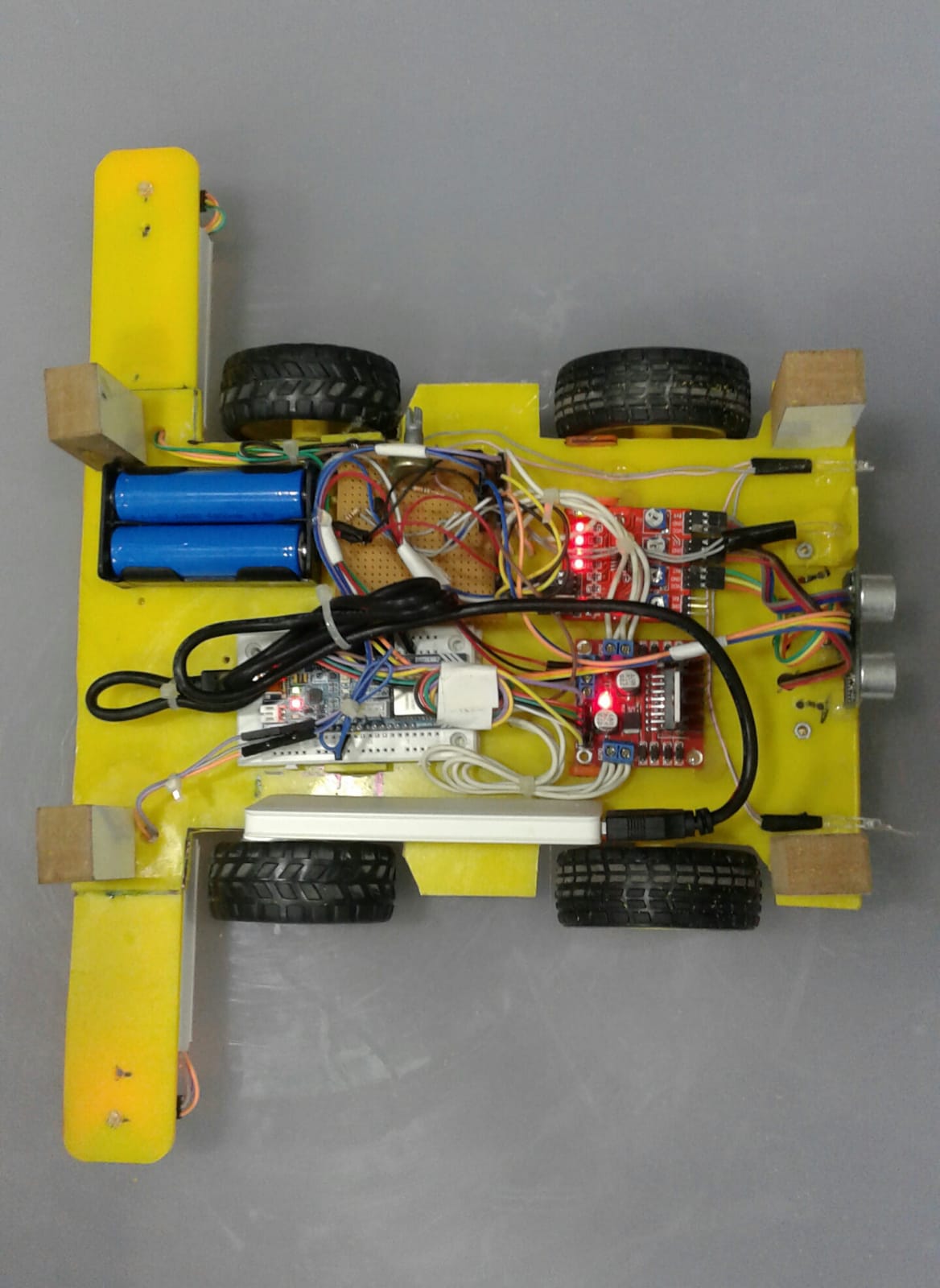
Yukarıda bahsetmiş olduğumuz sürüş yöntemlerini hareket edecekleri ortam ve ortam şartları, sahip oldukları güç depolama birimleri, yapacakları iş ve kontrol mekanizmaları göz önüne alınarak değerlendirip gerekli analizler yapıldığında ortaya çıkan karşılaştırma tablosu bize şu sonuçları vermektedir. Diferansiyel sürüş yöntemi bizim prototip robotu oluşturmak için ideal kapalı alanlarda kullanılmak üzere model olabilir [21]. Karşılaştırma tablosunda yapmış olduğumuzu değerlendirme indisleri arasında en efektif sonuçlara ulaştığımız yöntem diferansiyel sürüş yöntemidir.



**Tablo-2.2** Robot sürüş yöntemlerinin karşılaştırılması

### Mekanik Tasarım

Araç kitinin şase aksamını oluşturmada pleksi bir araba gövdesi kullanılmıştır. Diğer arabalar gibi 2 DC motor ile çalışmaktadır. Aracın ön kısmındaki DC motorlar ilerlemeyi ve yön kontrolünü sağlamaktadır. Bunlara ilave olarak; hem ESP32 yazılım kontrol kartına enerji vermesi için hem de araçtaki bir takım ekipmana enerji vermesi için pil yuvası kullanılmıştır.ESP32 Modülü için de ayrı bir yuva oluşturulmuştur. Projenin bitimine yakın aracın üstü pleksi malzemeyle kaplanmış ve dışarıdan gelebilecek belirli derecedeki fiziksel etmenlere karşı korunma sağlanmıştır. Bu da projenin hem mukavemetini arttırıp hem de yük taşıma kapasitesini arttırmıştır. Yapmış olduğumuz araç prototip olarak tasarlandığı için çok büyük yüklerde tam verim sağlamamaktadır. Taşınacak olan yüke göre mekanik tasarım ve kullanılan malzemeler değişmektedir.



**Şekil-2.6** Aracın Şase ve Motorlarının Üstten Görünümü

### Yazılımsal Tasarım

Bu projede ESP32 kartının yazılımının kontrolü için, Arduino’nun kendisine ait olan Arduino IDE’si kullanılmıştır. ESP32 gerekli komutlar vererek sistemin çalışması sağlanmıştır. Ayrıca bu sistemdeki malzemelerden gelen veriler de olduğu için, bu verilerin işleme tabi tutulması ve bu işlem sonucunun bir veriye dönüştürülmesi kısmı yine Arduino dilinde yazılan komutlar tarafından sağlanmaktadır. Aracın kontrol edilmesinde kullanılan kodların Arduino yazılımı EK-1 ‘ de verilmiştir.

**ULTRASONİK SENSÖR 29.1 DEĞERİNİN HESAPLANMASI**

**mesafe = (sure/2) / 29.1** (Ortam sıcaklığı sabit olarak 20 °C kabul edilir.)

Dalga Hızının Hesaplanması:

V (m/s) = 331 x

0 °C ‘de ses dalgasının hızı saniyede 331 m/s ‘dir.

V = 331 x = 342.91026069 m/s

mesafe = (sure/2) / 29.1

29.1 = = =

Yazdığımız kodda süre mikrosaniye, mesafe ise santimetre olarak alınmıştır.

29.1 =

V = 342.91026069 m/s (Dalga Hızı)

Bulduğumuz bu değerde metreyi santimetreye, saniyeyi ise mikrosaniyeye çevirmemiz gerekir.

1.Adım da metreyi santimetreye çeviriyoruz.

V = 342.91026069 x 100 = 34291.026069 cm / s

2.Adım da saniyeyi mikrosaniyeye çeviriyoruz.

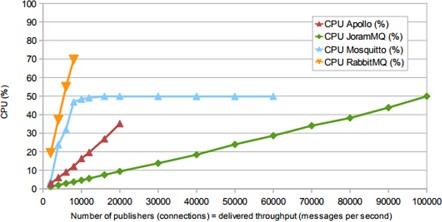
( 1 saniye = 1000000 µs )

V = 34291.026069 / 1000000 = 0.0342910261 cm / µs

3. Adım da ‘yi elde etmek için 0.0342910261 cm / µs değerini ters çevirmemiz gerekir.

= 29.162148636 µs / cm = 29.1 µs / cm olarak alınır.

Farklı dillerde yazılmış birkaç MQTT Brokerlar bulunmaktadır. Bunlardan en popüler olanları ActiveMQ, Apollo, JoramMQ, Mosquitto ve RabbitMQ olarak sıralanabilir.



**Tablo-2.3** Sunucu CPU Kullanım Oranları

CPU kullanımı göz önünde bulundurularak Mosquitto üzerinden çalışmalar yaygındır. Yazılımsal tasarım istemci ve sunucu olmak üzere iki konuda ele alınmaktadır. İstemci uzak yazılım güncellemeleri için OTA (Over the Air) Update kullanılmıştır.

### 2.4.1 İstemci (Client)

İstemci, bir ağ üzerinde, sunucu bilgisayarlardan hizmet alan kullanıcı bilgisayarlarıdır. Bilgiye erişim yetkileri sunucu tarafından belirlenir. Değişik çeşitleri olmakla birlikte önemli sınıflaması;

* Şişman/Zengin istemci: Yazılımın her seferinde sunucudan bütün istemcilere yüklenmesi ya da bir başka medyadan yüklenmesi gerekir. Örnek: Microsoft Outlook.
* Zayıf istemci: Sadece sunumla ilgili grafik birimleri ve onların denetim yazılımını içerir. İşle ilgili yazılım sunucudadır. Haberleşme kanalından gönderilen bilgi az olmasına dikkat edilir. Yazılımının bilgisayara yüklenmesi problemi yoktur. Zengin istemciye nazaran daha değişik bilgisayarlarda, hatta mobil gereçlerde de kullanılabilir. Örnek: J2EE/J2ME mimarisiyle yapılmış bir sitenin istemcisi.

### Sunucu (Server/Hosting)

Sunucu, bilgisayar ağlarında, diğer ağ bileşenlerinin (kullanıcıların) erişebileceği, kullanımına ve/veya paylaşımına açık kaynakları barındıran bilgisayar birimidir. Bir ağda birden fazla sunucu birim bulunabilir.

Çeşitleri de vardır: Shared Hosting (paylaşımlı barındırma), Co-Location, Reseller ve Dedicated (atanmış) Hosting.

* Paylaşımlı barındırmada bir sunucuda birden fazla alan barındırılabilir.
* Ortak merkezli sunucular, kendi sunucunuzu özel hazırlanmış veri merkezlerinde yüksek hızda hizmet bağlantısı ile kullanıma açmanızı sağlar. Reseller Hosting de belli sayıda alan (domain), disk ve bandwidth sınırları ile kullanıcıya sunulur. Paylaşımlı veya Atanmış sunuculardan farkları, reseller sunucuların daha çok sunucu hizmetini bir firmadan alıp, müşterilerine dağıtım yapan aracı küçük işletmelere hitap ediyor olmasıdır.
* Atanmış (Dedicated) Hosting'de, sunucu donanımı ve bağlantısı dahil tüm hizmetlerin sunucuyu kullanan şahıs veya firma tarafından karşılanmaktadır.
* Tüm sunucu çeşitlerinin Windows ve UNIX türevleri mevcuttur.

Windows firması tarafından Windows NT ile başlayan sunucu işletim sistemlerine, zamanla Windows 2000 Server, Windows 2003 Server, Windows 2008 Server, Windows 2012 Server ve Windows 2016 Server eklenmiştir. Kullanıcılar Windows işletim sistemine sahip bir sunucudan faydalanmak istediklerinde, Microsoft Windows firmasına lisans ücreti ödemeleri de gerekmektedir. Bu ücretler genellikle sunucu hizmetini sağlayan firma tarafından karşılanarak, son kullanıcının aylık ödeyeceği masrafa dâhil edilmektedir. Windows tabanlı sunucular, özellikle ASP gibi Windows işletim sistemine özgü programlama dilleriyle geliştirilen uygulamalar için kullanılmaktadır.

Linux tabanlı Centos, RedHat ve SUSE dağıtımları da sunucu pazarında kendine yer edinmişlerdir. Linux tabanlı sunucular, linux işletim sisteminin açık kaynak kodlu ve ücretsiz olması nedeniyle genellikle daha düşük maliyete sahiptir.

Ayrıca sunucuların hangi amaçla kullanılacağı da önemlidir. Örnek vermek gerekirse web hizmeti veren web sunucuları vardır. Donanım olarak büyük farklılıklar göstermemekle birlikte, sunucular kullanılacakları alana göre özel yazılımlarla

desteklenmektedir. Bunlara örnek olarak Apache, Microsoft IIS ve Abyss, Nginx ve Fastream IQ Web/FTP Server gösterilebilir. Dosya paylaşım, canlı yayın ve hatta çevrim içi oyun hizmeti verenleri de mevcuttur. Tüm bu sunucuların değişik sistem gereksinimleri vardır.

### OTA (Over the Air) Programlama

OTA (Over the Air) update, ESP modülüne bir seri bağlantısı noktası yerine Wi-Fi bağlantısını kullanarak ESP modülüne sabit yazılım yükleme işlemidir. Bu gibi işlevler modüle zor ve hiç erişiminin olmadığı durumlarda yararlı olabilir.

OTA 3 farklı şekilde yapılabilir. Bunlar;

* Arduino IDE’si yoluyla,
* WEB browser yoluyla,
* HTTP Server yoluyla.

Arduino IDE seçeneği öncelikle yazılım geliştirme aşamasında kullanılmak üzere tasarlanmıştır. Diğer iki seçenek, modülün bir web tarayıcısı ile manuel olarak veya bir http sunucusu kullanarak manuel olarak sağlanması için dağıtımdan sonrası için daha yararlı olacaktır.

Her durumda, ilk yazılım yüklemesi bir seri port üzerinden yapılmalıdır. OTA rutinleri doğru bir şekilde uygulanırsa, sonraki tüm yüklemeler kablosuz yapılabilir.

OTA işleminde saldırıya uğramamak için geliştirilen herhangi bir güvenlik protokolü yoktur. Güncellemelere yalnızca güvenilir kaynaktan izin verildiğinden emin olmak geliştiricinin sorumluluğundadır. Güncelleme tamamlandıktan sonra modül yeniden başlatılır ve yeni kod çalıştırılır. Geliştirici, modülde çalışan uygulamanın kapatıldığından ve güvenli bir şekilde yeniden başlatıldığından emin olmalıdır. Aşağıdaki bölüm OTA sürecinin güvenliği ve güvenliği ile ilgili ek bilgiler sağlar.

### OTA Sürecinin Güvenliği

Modülün yeni bir eskiz (sketch) ile güncellenmesi için kablosuz olarak yapılması gerekir. Bu, modülün kötü amaçla saldırıya uğrayabileceği ve başka bir kod yüklenme ihtimallerini ortaya çıkarmaktadır. Saldırıya uğramanın olasılığını azaltmak için, yüklemelerinizi bir şifreyle, belirli OTA bağlantı noktası vb. seçerek koruyabilirsiniz.

Güvenliği artıracak ArduinoOTA kütüphanesi ile sağlanan işlevselliği kontrol edelim:

1. **void** setPort(uint16\_t port);

2. **void** setHostname(**const char** \* hostname);

3. **void** setPassword(**const char** \* password);

Bunun gibi belli koruma işlevleri önceden oluşturulmuştur ve geliştirici tarafından herhangi bir ek kodlama gerektirmez. ArduinoOTA ve espota.py, karşıdan yüklemeyi doğrulamak için Digest-MD5 kullanıyor. Aktarılan verilerin bütünlüğü, ESP tarafında MD5 kullanılarak doğrulanır.

Kendi risk analizinizi yapın ve uygulamaya bağlı olarak hangi kütüphane işlevlerini uygulayacağınıza karar verin. Gerekirse, diğer koruma yöntemlerinin saldırıya maruz kalmamak üzere uygulanmasını göz önünde bulundurun, ör. Sadece belirli bir programa göre yüklemeler için modül açığa, OTA yalnızca kullanıcı vb. adanmış “Update” düğmesine basarak tetiklemek gibi senaryolar geliştirebilirsiniz.

### OTA Sürecinde Emniyet

OTA işlemi, yükleme sırasında ESP’nin kaynaklarını ve bant genişliğini alır. Ardından, modül yeniden başlatılır ve yeni bir eskiz (skecth) gerçekleştirilir. Mevcut ve yeni eskizinizin işlevselliğini nasıl etkilediğini analiz edin ve test edin.

ESP uzak bir yere yerleştirilirse ve bazı ekipmanları kontrol ediyorsanız, bu ekipmanın çalışması aniden güncelleme işlemi tarafından kesilirse ne olacağına dikkat etmeniz gerekir. Bu nedenle, güncellemeye başlamadan önce bu ekipmanın nasıl güvenli bir hale getirileceğine karar verin. Örneğin, modülünüz bahçe sulama sistemini sırayla kontrol ediyor olabilir. Bu sıra düzgün şekilde kapatılmazsa ve bir su vanası açık kalırsa, OTA bittikten ve modül yeniden başlatıldıktan sonra bu vana kapalı değilse, bahçeniz su basabilir.

Aşağıdaki işlevler ArduinoOTA kitaplığı ile sağlanır ve OTA’nın belirli aşamalarında veya OTA hatasında uygulamanızın işlevselliğini ele almayı amaçlar:

|  |
| --- |
| 1. **void** onStart(OTA\_CALLBACK(fn)); |
| 2. **void** onEnd(OTA\_CALLBACK(fn)); |
| 3. **void** onProgress(OTA\_CALLBACK\_PROGRESS(fn)); |
| 4. **void** onError(OTA\_CALLBACK\_ERROR(fn)); |
|  |
|  |

### 2.4.3.3 OTA İşlemi İçin Temel Gereksinimler

Skeç dosyasının iki katı büyüklüğünde flash chip hafızası

Aşağıdaki bölümlerde, OTA’nın yapılmasına ilişkin daha ayrıntılı bilgi ve özel yöntemler içermektedir.

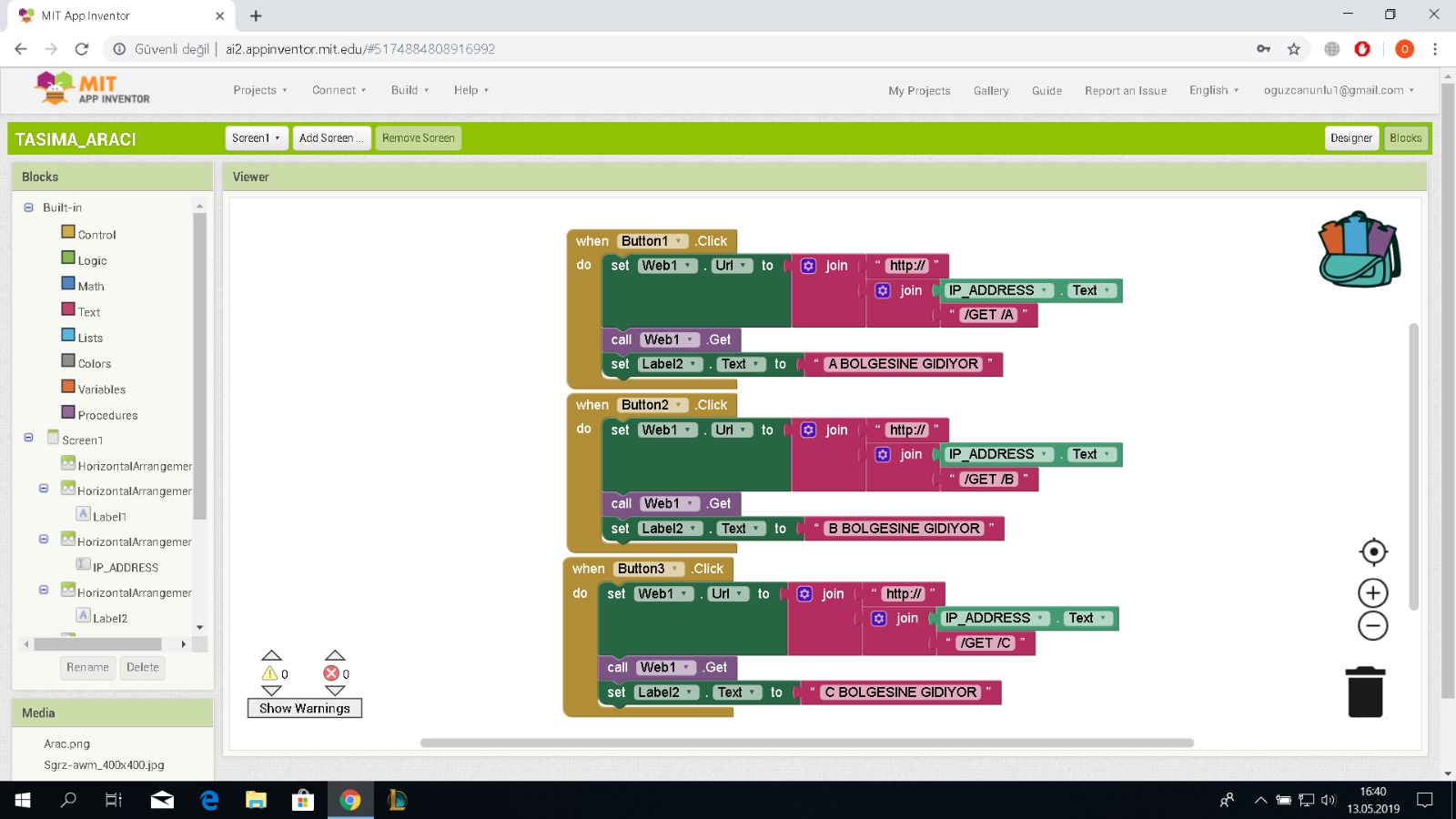
* 1. **Elektronik Tasarım**

Bu projedeki araç, kontrol ve motor sürüş devresi olmak üzere devreden oluşmaktadır. Ancak bu 2 baskı devreyi de kontrol eden ana merkez ESP32 yazılım kontrol kartıdır. Kontrol devresinde kullanılan malzemeler; HC-SR04 ultrasonik mesafe sensörleri, optik sensörler ve sensörlerin bağlandığı soketlerden oluşmaktadır. Motor sürüş devresinde ise L298N çift kanallı motor sürücü kartı, ve DC motorlar kullanılmıştır.

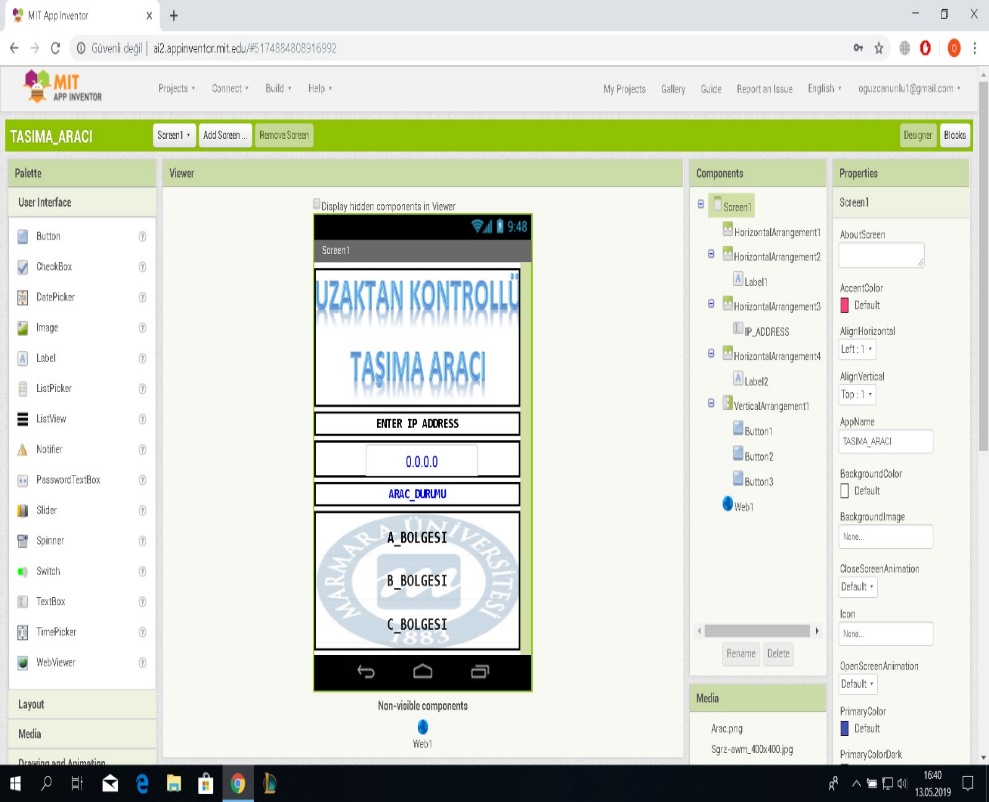
Bu 2 devrenin dışında bir de aracın mobil cihaz tarafından kontrol edilebilmesi için gerekli olan telefon bulunmaktadır. Bu telefon sayesinde aracın merkezi olan ESP32 yazılım kontrol kartı ile mobil cihaz arasındaki bağlantı sağlanmış olur. Bu sayede mobil cihazdaki uygulama ile aracın istenilen hareketi için butona basıldığında ESP32 yazılım kontrol kartının ilgili çıkış pini aktif edilir.

* 1. **Haberleşme Tasarımı**

Araç kitindeki malzemeler, verilerinin işlenmesi için ESP32 yazılım kontrol kartına bağlanmıştır. ESP32 yazılım kontrol kartında bu verileri işlemek için Arduino IDE programında yer alan yazılım dilindeki kodlar kullanmıştır. Bu kodların dışında bir de aracın mobil kontrolü için kodları bulunmaktadır. Araç uzaktan bir telefonla kontrol edilebilmesi için “MIT App Inventor 2” web sitesinden bir Android uygulama oluşturulur. Oluşturulan bu mobil uygulama içerisinde Wifi bağlantı kısmı eklenir ve ESP32 Modülü ile uygulama eşlenir. MIT App Inventor 2 web sitesinde yazılan programa blok şeması halinde kodlarla Arduino yazılım kartına veri gönderilir ve veri teslim alınır. Bu sayede aracın hareketi ve sensörlerden gelen veriler mobil uygulamadan takip edilebilir.



**Şekil-2.7** MIT App Inventor 2 Programında Butonların Arduino Koduna Tanımlanması



**Şekil-2.8** Android Program Ekran Görüntüsü

* 1. **Tasarımda Kullanılan Malzemelerin Özellikleri**

Bu bölümde, araç kitinin tasarımında kullanılan; ESP32 yazılım kontrol kartı, DC motor, pil yuvası, powerbank, HC-SR04 ultrasonik mesafe sensörü, L298N çift kanallı motor sürücü kartı gibi malzemelerin teknik özellikleri ve kullanım sebepleri gibi tanımlamalar yapılacaktır.

**ESP32 Geliştirme Kartı:** SparkFun şirketi tarafından giyilebilir cihazlar ve IoT (Nesnelerin İnterneti) projeleri için tasarlanmıştır. Son teknoloji entegreler, maksimum hız minimum güç güç tüketimi ile çalışır. ESP32 Wifi entegresi güç tasarrufu için belirlenen durumlarda periyodik olarak çalışır. Böylece pil ve batarya gerektiren projelerde sorun yaşamazsınız. Wifi protokolünün yanında Bluetooth ve RF gibi kablosuz haberleşme özellikleri ile uzaktan kontrol için alternatifler arttırılmıştır [22].

ESP32 Geliştirme Kartı Teknik Özellikleri:

-Çift çekirdekli Tensilica LX6 mikroişlemci

-240MHz'e kadar saat frekansı

-520kB dâhili SRAM

-Dâhili 802.11 b/g/n WiFi

-Dâhili çift modlu Bluetooth (klasik ve BLE)

-2.2-3.6V çalışma gerilimi

-2.5µA derin uyku akımı

-28 adet GPIO

-10 adet kapasitif dokunma bağlantısı

-Donanımsal şifreleme desteği (AES, SHA2, ECC, RSA-4096)

-4MB flaş bellek

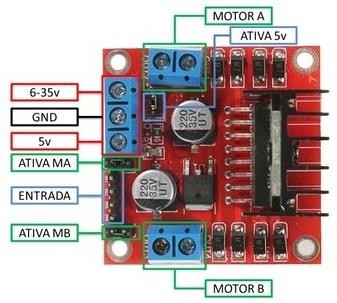
-Dâhili LiPo şarj devresi (JST soketli)

**DC Motor*:*** Bu araç kitinin ileri-geri hareket etmesini sağlayan motor çeşididir. Projede kullanılan modeli “fırçalı DC motor” dur. Bu motorun çalışma prensibi: Motorun ana milinin üzerinde bobinler bulunur ve motorun ana gövdesinin iç kısmında ise güçlü mıknatıslar yer alır. Şaft üzerinde yer alan bobinlere fırçalar (sanayi gibi ortamlarda “kömür” de denir) vasıtasıyla elektrik akımı uygulanır. Uygulanan elektrik akımıyla bobinlerde oluşan manyetik alan, mıknatısların manyetik alanıyla sürekli çakışacak şekilde bir etki gösterir ve bu sayede mil hareket etmiş olur.



**Şekil-2.9** Fırçalı DC Motor

**L298N Çift Kanallı Motor Sürücü Kartı:**Motorların hız ve yönlerini kontrol etmek amacıyla kullanılan bir çift H köprülü motor sürücü kartıdır. H-köprüleri ise düşük akımlar ile büyük akımların iletilmesini kontrol edebilmek amacıyla kullanılan devrelerdir. Motor kontrolü dışında ışıklandırma projelerinde LED gruplarının parlaklarının ayarlanması amacıyla da kullanılır [23].



**Şekil-2.10** L298N Motor Sürücü Kartı

**HC-SR04Ultrasonik Mesafe Sensörü:** Hc-Sr04 ultrasonik sensör sonar (Sound Navigation and Ranging ) kullanarak karşısındaki nesneye olan mesafesini hesaplayan bir input kaynağıdır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığını boyutunu elde etmemizi sağlar. Bu tür sensörlerin esin kaynağı yunuslar ve yarasalardır. Onlarda sonar ile iletişim kurar ve hareket eder.2cm ile 400cm arası mesafe en sağlıklı okuma yaptığı aralıktır. Üzerinde bir alıcı ve bir verici modül bulunur [24].



**Şekil-2.11** HC-SR04 Ultrasonik Mesafe Sensörü

**Pil yuvası***:* Bu araç kitinin birkaç elemanı için gereken enerjinin sağlanması için kullanılacak olan pillerin konulduğu yaylı şekildeki, pilleri sıkıca tutan ve üzerinde kullanım amacı için ortalama 10 cm. kadar kabloları bulunan aparattır. Pillerin rahat bir şekilde kullanılması ve değiştirilmesine kolaylık sağlar [25].



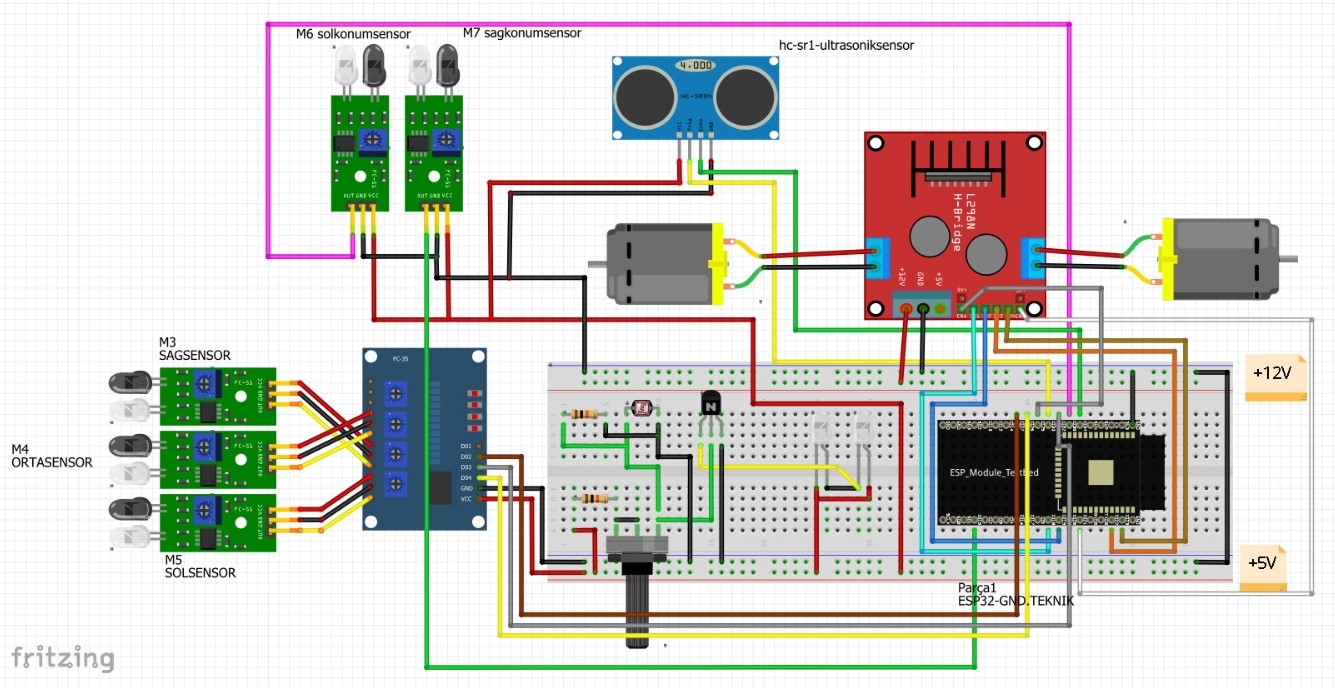
**Şekil-2.12** Pil Yuvası

* 1. **Bu Bölümden Elde Edilen Çıktılar**

Bu bölümde; aracın daha dayanıklı ve hafif olması için pleksi bir malzemeden araç kiti tasarımı yapılmıştır. Piller için ayrı bir pil yuvası yapılmıştır. Ayrıca araca eklenen ultrasonik mesafe sensörü sayesinde aracın önüne çıkabilecek fiziksel bir engelde araç kendini otomatik olarak durduracaktır. Bu sayede iş güvenliği maksimum seviyede olacaktır.

# BÖLÜM III

## Sistemin Uygulaması ve Sonuçları

****

**Şekil-3.1** Projenin Fritzing Şeması

Fritzing; Raspberry Pi, Arduino vb. elektronik tabanlı projelerde, prototipten üretime

geçerken üretim maliyetlerini azaltmayı ve çıkabilecek fiziksel sorunları en aza indirmeyi amaçlayan açık kaynak kodlu bir devre tasarım programıdır. Şekil 3.1’de sistemi gösteren Fritzing şeması gösterilmiştir. Şemada belirtildiği gibi araçta; ESP32 Geliştirme Kartı, 2 adet DC Motor, HC-SR04 Ultrasonik Mesafe Sensörü ve gösterilen diğer devre elemanları kullanılmıştır. Çizgi izleyen sensörler aracın çizgiyi takip etmesini sağlar. Diğer 2 sensör ise aracın konumunu tespit etmeye yarar.

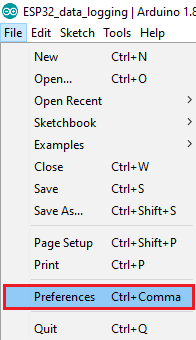
### Bağlantı Denemeleri

Bu kısımda ESP32 kartımızı Ardunio’nun kendi IDE programıyla nasıl programladığımızı belirttik. Şimdi bu işlemi tek tek nasıl yaptığımızı aşağıda anlatacağız:

**1. ESP32 Kartını Tanıtma**

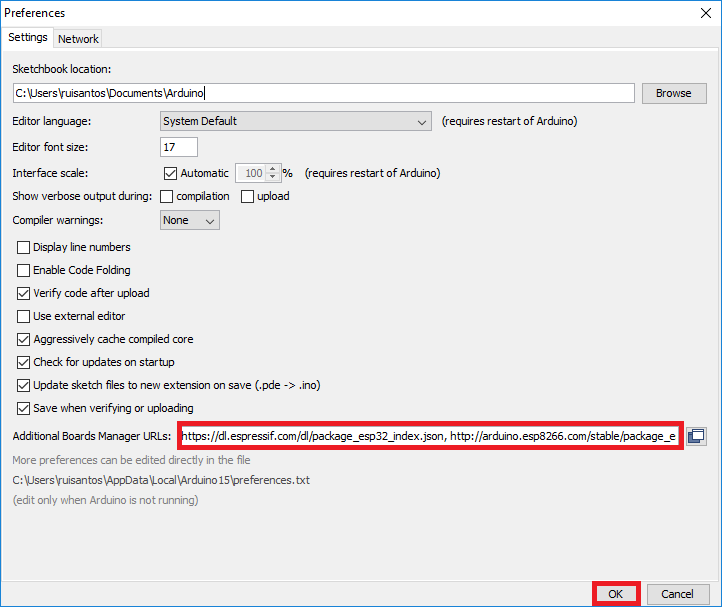
ESP32 kartını Arduino IDE'nize kurmak için aşağıdaki talimatları izleyin:

1) Arduino IDE'den tercihler penceresini açın. **Dosya** > **Tercihlere** git



**Şekil-3.2** Adım 1

2) Aşağıdaki şekilde gösterildiği gibi “Ek Kurul Yöneticisi URL'leri” alanına **https://dl.espressif.com/dl/package\_esp32\_index.json adresini**  girin. Ardından, “Tamam” düğmesine tıklayın:

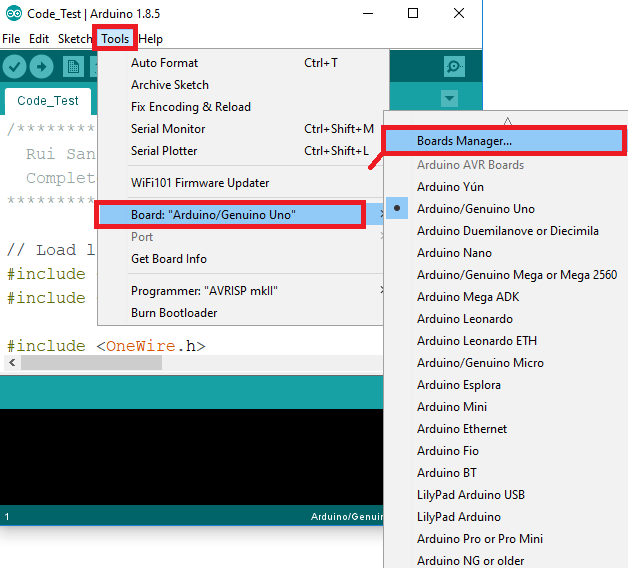


**Şekil-3.3** Adım 2

**Not:** ESP8266 panoları URL’niz zaten varsa, URL’leri virgül ile aşağıdaki gibi ayırabilirsiniz:

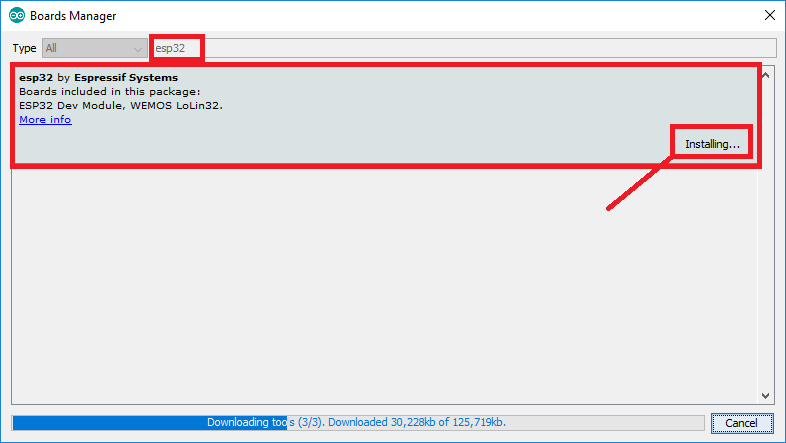
https://dl.espressif.com/dl/package\_esp32\_index.json, <http://arduino.esp8266.com/stable/package_esp8266com_index.json>

3) Kurul yöneticisini açın. Gidin **Araçlar** > **Kurulu** > **Kurulum Müdürü**



**Şekil-3.4** Adım 3

4) ESP32'yi arayın ve “ **ESP32 by Espressif Systems** ” için kurulum düğmesine basın:



**Şekil-3.5** Adım 4

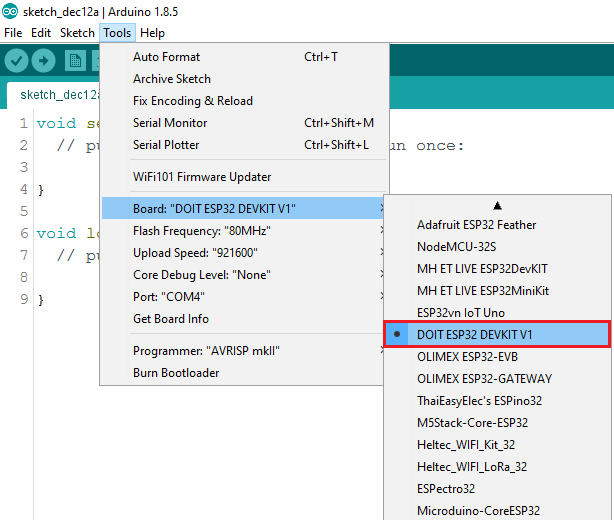
5) Kurulum tamamlandı.

**Kurulumu Test Etme**

ESP32 kartını bilgisayarınıza takın. Sonra şu adımları izleyin:

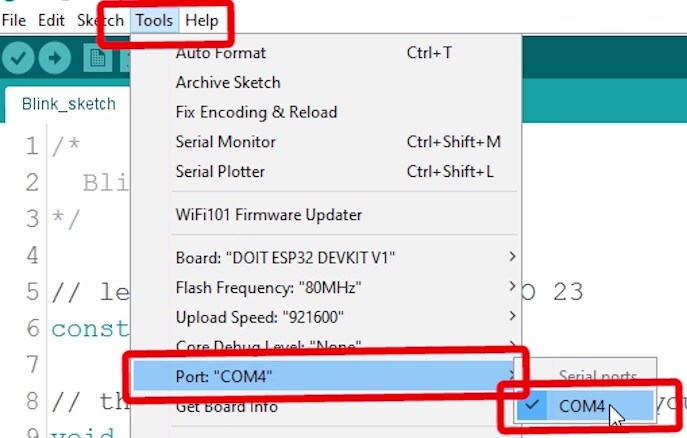
1) Arduino IDE'yi açın

2)  **Araçlar**  >  **Pano**  menüsünden **DOIT ESP32 DEVKIT V1**  seçin



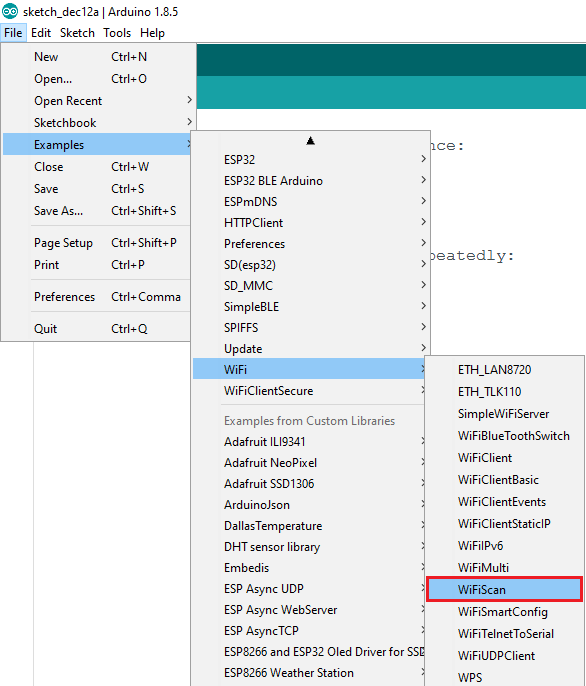
**Şekil-3.6** Adım 5

3) Portu seçin (Arduino [IDE'nizde](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers) COM Portunu göremiyorsanız, [ESP32 CP210x USB'yi UART Bridge VCP Sürücülerine](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers) yüklemeniz gerekir  ):



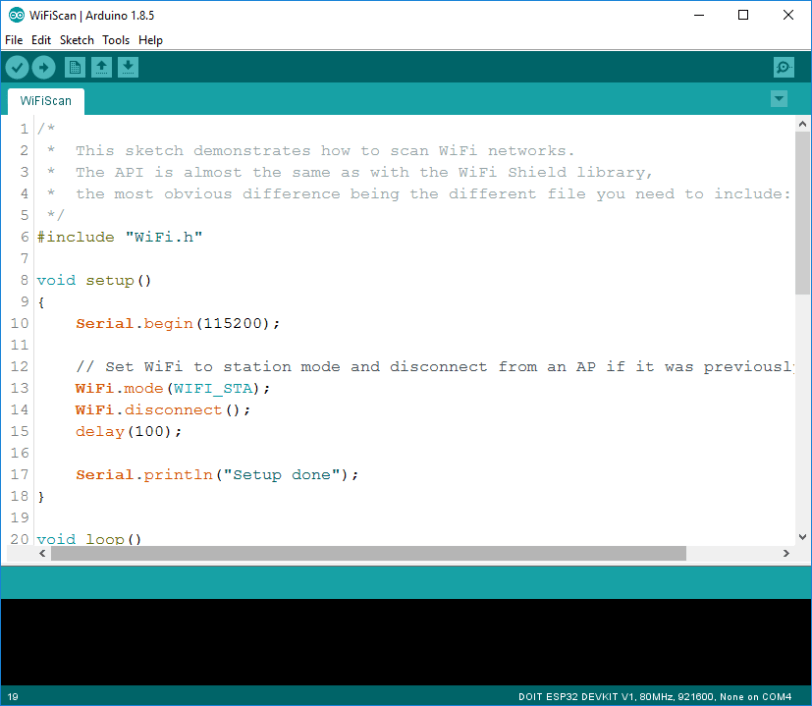
**Şekil-3.7** Adım 6

4) **Dosya** > **Örnekler** > **WiFi (ESP32)** > **WiFi Taraması** altında aşağıdaki örneği açın.



**Şekil-3.8** Adım 7

5) Yeni bir taslak açılır:

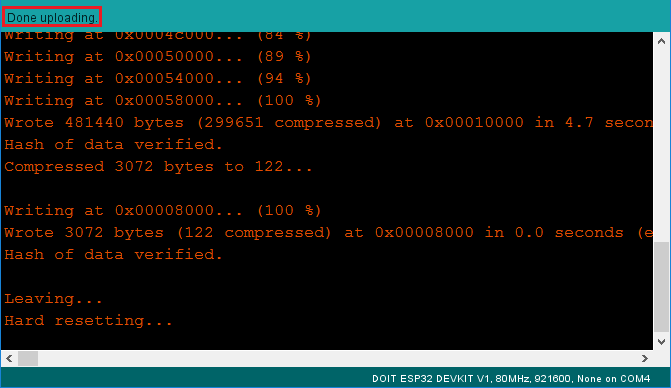


**Şekil-3.9** Adım 8

6)  Arduino IDE'deki **Yükle** düğmesine basın. Kod derlenir ve panonuza yüklenirken birkaç saniye bekleyin.

https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2016/12/arduino-ide-upload-button.png?resize=34%2C29&ssl=1

7) Her şey beklendiği gibi giderse, **Yükleme işlemi tamamlandı.**

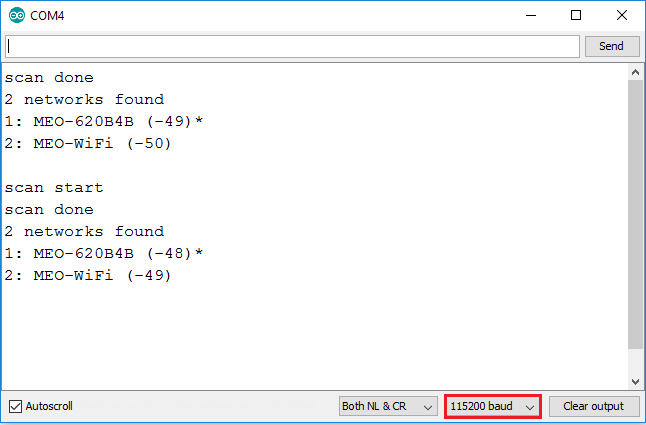


**Şekil-3.10** Adım 9

8) Arduino IDE Serial Monitor'ü 115200 baud hızında açın:

https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2016/12/mac-open-arduino-ide-serial-monitor.png?resize=38%2C29&ssl=1

9) ESP32 yerleşik **Etkinleştir**  düğmesine basın; ESP32'nizin  yakınında bulunan ağları görmelisiniz:



**Şekil -3.11** Adım 10

### Mekaniksel Uygulama

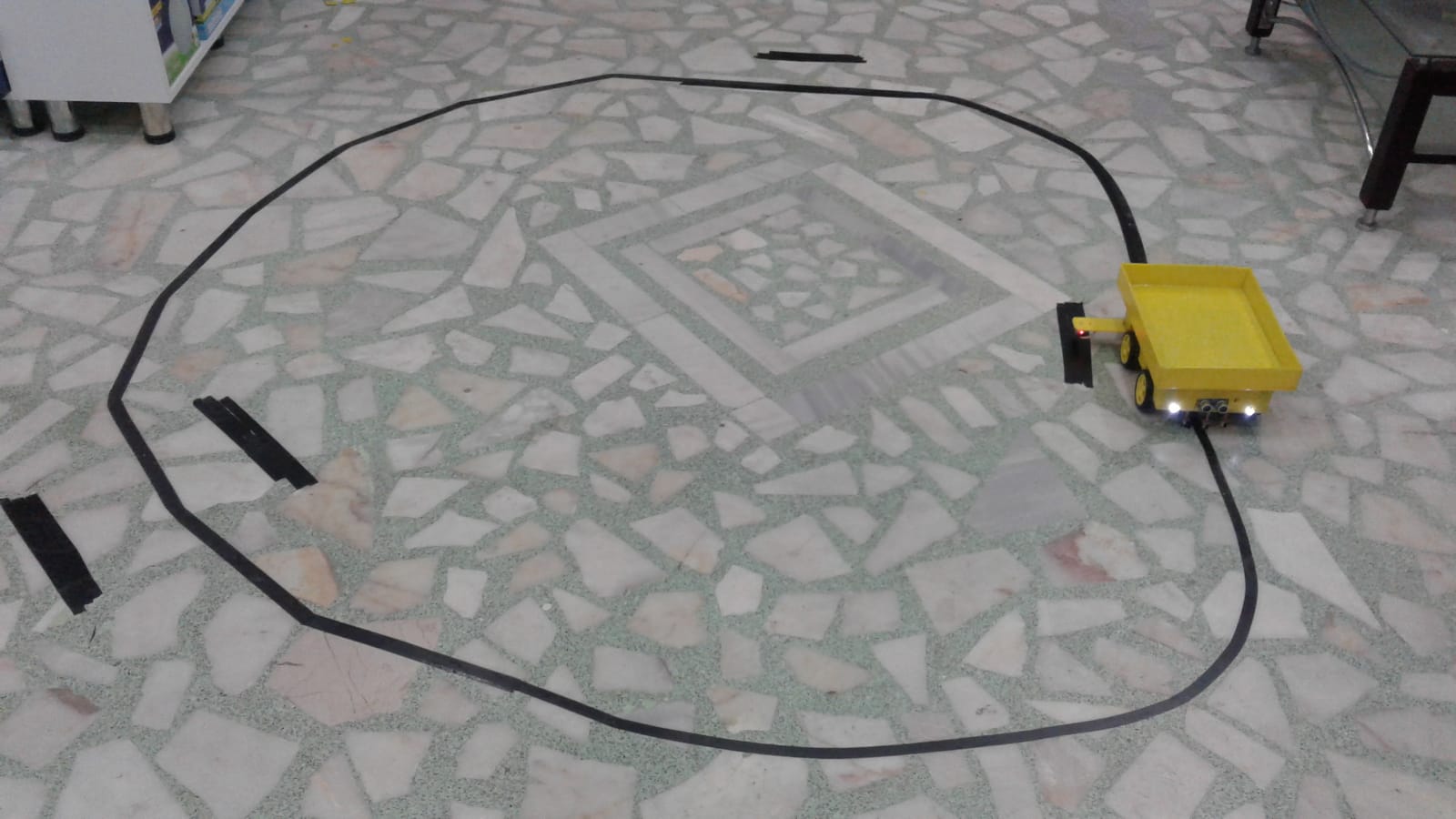
Mekaniksel uygulama gerçek işleyişteki prototipin son hali tamamlanmıştır. Araca gerekli testler yapılarak sistemin düzgün çalışıp çalışmadığı kontrol edilmiştir. Ayrıca aracın istenilen bölgelerde çalışıp çalışmadığı kontrolü yapılmıştır. Ayrıca güvenlik amaçlı ultrasonik sensörün hassas ölçüm yapıp yapmadığı kontrol edilmiştir.

Aşağıdaki şekilde aracın son hali gösterilmiştir.



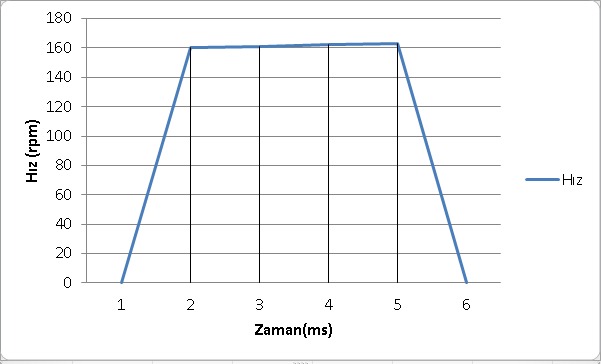
**Şekil-3.12** Aracın Son Hali

Aracın çalışmasını aşağıdaki şekilde kendimiz oluşturduğumuz pistte denedik. Yolları siyah bantla oluşturduk. Bu sayede ayrı bir pist yapmak yerine istediğimiz yerde pistimizi oluşturabiliriz.



**Şekil-3.13** Araç Pisti

Aracın Hız-Zaman grafiği Tablo-3.1’de verilmiştir. Bu grafiği açıklayacak olursak araca ilk hareketi verdiğimiz anda motorlar 1 ms de 160 rpm hıza ulaşır. Dışarıdan yük ve kuvvet etki etmediği sürece araç bu hızla ivmesini koruyarak gitmeye devam eder. Aracı durdurmak istediğimizde ise araç sabit bir ivmeyle 1 ms gibi düşük bir sürede durur.



**Tablo-3.1** Hız-Zaman Grafiği

### Bölümden Çıkarılan Sonuçlar

Sistemin son olarak uygulanmasında elde edilen tecrübelere göre ESP-8266 modülünün istediğimiz fonksiyonları yerine getiremediği belirlenmiştir. Ayrıca ESP-8266 modülünün if, while vb. döngülerde hız bakımından yetersiz olduğundan gereksinimlerimizi karşılayamadığını tespit ettik. Bu yüzden bu modülün yerine ESP32 modülünü kullanarak sorunu çözdük. Ve başarılı bir şekilde sistemi çalıştırmayı başardık.

# BÖLÜM IV

## Çalışmanın Genel Sonucu

Bu projenin sonucunda; proje amaçlarına ulaşılmış, sistemin çalışması değerlendirilmiş ve bu değerlendirmeler sonucu verimli bir çalışma elde edilmiştir. Yapılan projede kullanılan kodlarla ve programlarla, projenin amacı doğrultusunda gerçekleştirilmek istenen insan gücüne gerek kalmadan büyük fabrika tarzı yerlerde önceden belirlenmiş yörüngelerde ve bölgelerde malzeme transferinin sağlanması gerçekleştirilmiştir.

Projenin amacı doğrultusunda, öncelikle kullanılan her bir parçanın sağlamlık kontrolü yapılmıştır ve verimli çalıştığına kanaat getirilen parçalar kullanılmıştır. Daha sonra bu kullanılan parçaların her birinin teorik bilgileri, çalışma prensipleri ve çalışma koşulları araştırılıp bilgi sahibi olunmuştur. Bu parçaların gerekli olduğu anlarda çalışıp gerekli olduğu anlarda durmasını sağlayan kodlar, ESP32 adı verilen bir gömülü sistem sayesinde oluşturulmuştur. WEB üzerinden aracın kontrol edilmesi için bir web sayfası tasarlanmıştır. Hazırlanan bu web sayası ile Arduino IDE programında yazılmış olan kodların haberleşmesini sağlamak için ESP32 modülü kullanılmıştır.

### Veri Akış Blok Şeması

ESP32 GELİŞTİRME

KARTI

ANDROİD PROGRAM

ULTRASONİK ve

OPTİK SENSÖRLER

**Şekil-4.1** Veri Akışı Blok Şeması

Şekil 4.1’de projedeki sistemin haberleşme kısmında yapılan veri transferinin blok şeması belirtilmiştir. Proje için MIT App Inventor 2 web sitesinde tasarlanan Android uygulama programında aracın istediğimiz bölgeye gitmesini sağlayan bölge butonları oluşturulmuştur. Ardunio IDE’ sinde yazdığımız programa göre Ultrasonik ve Optik Sensörlerden gelen veriler değerlendirilir ve yorumlanır. Ve bu veriler ESP32 kartına gönderilir.ESP32 kartı istenilen şartlar sağlandığı takdirde yapmak istediğimiz işlemi yaparak görevini tamamlar. Ve işleme son verir.

### Çalışma Şeması

HC-SR04 ULTRASONİK

MESAFE SENSÖRLERİ

ESP32 *GND* TEKNİK

MODÜLÜ

L298N MOTOR

SÜRÜCÜ KARTI

KONUM SENSÖRLERİ

ÇİZGİ İZLEYEN SENSÖRLER

DC MOTORLAR

**Şekil-4.2** Tasarım Çalışma Şeması

HABERLEŞME

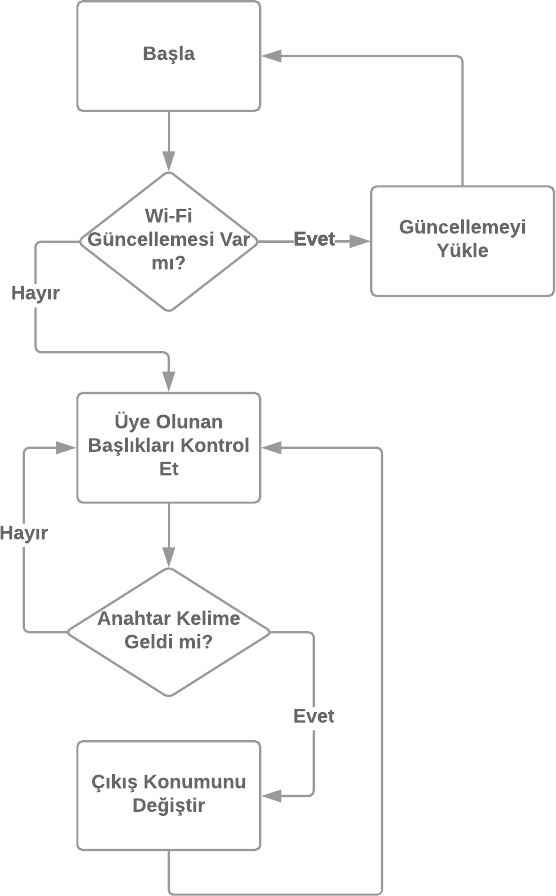
MATERYALLERİ

DİĞER EKİPMANLAR

Şekil 4.2’de projenin tasarımında yer alan her parçanın birbirleri ile olan etkileşimi şematik bir biçimde gösterilmiştir.

### Yazılımsal Akış Şeması (Algoritması)

Şekil 4.3’de sistemin yazılımsal akış şeması gösterilmiştir.



**Şekil-4.3** Yazılımsal Akış Şeması

### Sonuç

Endüstriyel taşıyıcı robotların sanayi ortamındaki otomasyon kullanımındaki yeri kaçınılmazdır. Otomasyon sistemlerinin varlığı hem verimli çalışmayı hem de bakım ve onarım maliyetlerini doğrudan etkilemektedir. Bu tez kapsamında yapılan proje aynı mantık doğrultusunda daha büyük şirketlerde, fabrikalarda bölümden bölüme malzeme, bilgi paylaşımının hızlı ve doğru yapılması istenen yerlerde kullanılabilir.

Mekanik ve elektronik çatısı tamamlanmış olan bu projede çizgi takibi uygulamasında çizgi tespiti için manyetik algılama ve ultrasonik algılama yöntemlerinin kullanılmasının bazı ortamlarda verimsiz olduğu, çizgi takibinin stabil yapılabilmesi amacı ile manyetik nesnelerden ve sonar sensörün algılayabileceği yapılardan yeterli derecede korunması gerektiği görülmüştür. Bu faktörler göz önüne alınarak, çizgi takibi sensör modülünün tasarlanmasında, kızılötesi led ve kızılötesi alıcılar kullanılmıştır. Endüstriyel taşıyıcı robotun daha adaptif bir yapıya kavuşması için, yol izleme kısmı çizgi takibi yerine, görüntü işleme metodu ile de gerçekleştirilebilir ancak görüntü işleme teknolojisinin maliyeti ve bakım uygulamaları oldukça zordur.

Robotun endüstriyel uygulamalarda kullanılabilecek bir modelinin yapılması esnasında elektronik kartları mekanik tasarımda değiştirilebilirlik adına robotun kolay ulaşılabilir bir bölgesinde olması ve tahrik sisteminin servo motorları yerine elektrik motoru veya adım motorları seçilerek yapılması bakım ve üretim maliyetlerini azaltmaktadır. Endüstriyel robot sabit bir enerji kaynağından beslenemeyeceğinden akülerin kolay sökülür takılır olması gerekmektedir. Robotun tahrik mekanizması arkadan olduğu için vagon sistemi veya robotun üst platformu düz yapılarak malzeme taşıma işlemi bu şekilde gerçekleştirilebilir

# KAYNAKÇA

1. Ashton, K., 2009. That 'Internet of Things' Thing. RFID Journal, <http://www.rfidjournal.com/articles/pdf?4986>). -4 Mart 2019
2. Armentia ve ark 2012(Armentia, J., C.-Mansilla, D., Ipiña, D. L., 2012. Fighting against Vampire Appliances through Eco-aware Things. Sixth International Conference on Innovative Mobile and Internet Services, Ubiquitous Computing.). -6 Mart 2019
3. Anonim, 2001. The Story of the Trojan Room Coffee Pot, http://www.cl.cam.ac.uk/coffee/qsf/timeline.html - (16.05.2016) -7 Mart 2019
4. Gökrem, L., and Bozuklu, M., "Nesnelerin İnterneti: Yapılan Çalışmalar ve Ülkemizdeki Mevcut Durum." *Gaziosmanpaşa Bilimsel Araştırma Dergisi* 13 (2016): 47-68. -8 Mart 2019
5. Hatton, M., Machina Research (2012), M2M Market Forecasts[,http://www](http://www/). webcitation.org/query?url=http%3A%2F%2Fwww.telecomengine.com%2Fsites%2F default%2Ffiles%2Ftemp%2FCEBIT\_M2M\_WhitePaper\_2012\_01\_11.pdf&date=20

-8 Mart 2019

1. İren, K. "Endüstriyel Çizgi Takip Eden Robot Cihazı Geliştirilmesi." (2013). -9 Mart 2019
2. “Nesnelerin İnterneti ve Endüstriyel Uygulamaları ” www.endustri40.com ,Web. -9 Mart 2019
3. Ercan, T., and Mahir K., "Endüstride Nesnelerin Interneti (IoT) Uygulamaları." Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi 16.3 (2016): 599-607.

10 Mart 2019

1. Atzori, L., Lera A., Morabito, G., 2010. Internet of things: A survey. Journal of Computer Networks, 54, 2787- 2805. -10 Mart 2019
2. Miorandi, D., Sicari, S., Pellegrini, F.D., Chlamtac, I., 2012. Internet of things: Vision, applications and research challanges. Journal of Ad Hoc Networks, 10, 1497-1516 15 Mart 2019
3. Li, D.X., Wu, H., Shangang, H., 2014. Internet of things in industries: A Survey. IEEE Transactions on Industrial Informatics, 10, 2233-2243. -15 Mart 2019
4. Ercan, Tuncay, and Mahir Kutay. "Endüstride Nesnelerin Interneti (IoT) Uygulamaları." Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi 16.3 (2016): 599-607.

-16 Mart 2019

1. Thomson, A., J., 2001, A Path Following System for Autonomus Robots with Minimal computing Power, Yüksek Lisans Tezi, University of Auckland, Auckland, 59 s. -16 Mart 2019
2. Borenstian,J. Everett, H.R., Feng, L., 1996, Sensors and Methods for Mobile Robot Positioning, The university of Michigan, 282 s. -17 Mart 2019
3. Bansal, N., Arora, S., Shiva, S., 2010, Line Follower Robot, Department of Physics Panjab University, Chandigarh, 30 s. Wilson, J., S., 2005, Sensor Technology Handbook, Newnes, Bulington, 690 s. -12 Nisan 2019
4. Wilson, J., S., 2005, Sensor Technology Handbook, Newnes, Bulington, 690 s.

-13 Nisan 2019

1. Wee, K. C., 2007, PID Control of Line Following Robot, Yüksek Lisans Tezi, Faculty of Electronics & Computer Engineering Universiti Teknikal Malaysia, 27 s. -13 Nisan 2019
2. Jones, J., L., Flynn, A., M., Seiger, B., 1998, Mobile Robots Inspiration to implementation, AK Peters, Natick, 486 s. -14 Nisan 2019
3. Lucas, G.W., 2001, A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators, <http://rossum.sourceforge.net/papers/DiffSteer/>. -6 Mayıs 2019
4. Jones, J., L., Flynn, A., M., Seiger, B., 1998, Mobile Robots Inspiration to implementation, AK Peters, Natick, 486 s. -10 Mayıs 2019
5. Belpaeme, T., 2004, Autonomous Systems Manipulation and Locomotion, Autonomous Systems. -11 Mayıs 2019
6. “Esp32 Teknik Özellikleri” Direnç.net, Web. -17 Mayıs 2019
7. “L298N H BRIDGE ÇİFT MOTOR SÜRÜCÜ KARTI” Robotpark, Web. -17 Mayıs 2019
8. “Hc-Sr04 Nedir?” Sinancanbayrak, Web. -18 Mayıs 2019
9. “Pil Yatağı” Ekşi Sözlük, Web. -18 Mayıs 2019

# EKLER

## EK-1: Açık Kaynak Kodu

#include <WiFi.h>

char\* ssid = "ZyXEL\_7A20";

char\* password = "26ts375unlu";

WiFiServer server(80);

#define motorR1 18

#define motorR2 19

#define motorL1 22

#define motorL2 23

#define solsensor 26

#define ortasensor 27

#define sagsensor 32

#define trigpin 33

#define echopin 34

#define sagkonumsensor 15

#define solkonumsensor 35

int sure;

int mesafe;

int baslangicTime;

int i;

int k;

void setup()

{

Serial.begin(115200);

/\*Serial.println("Restarting in 10 seconds");

delay(30000);

ESP.restart();\*/

pinMode(motorR1, OUTPUT);

pinMode(motorR2, OUTPUT);

pinMode(motorL1, OUTPUT);

pinMode(motorL2, OUTPUT);

pinMode(solsensor, INPUT);

pinMode(ortasensor, INPUT);

pinMode(sagsensor, INPUT);

pinMode(echopin, INPUT);

pinMode(trigpin, OUTPUT);

pinMode(sagkonumsensor, INPUT);

pinMode(solkonumsensor, INPUT);

attachInterrupt(digitalPinToInterrupt(echopin),detect,CHANGE);

digitalWrite(trigpin, LOW);

delay(10);

ledcSetup(0,5000,8); //sol motor hız ayarı

ledcAttachPin(21,0); //sol motor hız ayarı

ledcSetup(0,5000,8); //sag motor hız ayarı

ledcAttachPin(25,0); //sag motor hız ayarı

// We start by connecting to a WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

int value = 0;

int bolge = 0;

void loop(){

WiFiClient client = server.available(); // Gelen masterileri dinle

if (client) { // Eger bir masteri bulursan

Serial.println("new client"); // Seri baglanti noktasinda bir mesaj yazdir

String currentLine = ""; // İstemciden gelen verileri tutmak icin bir string yap.

while (client.connected()) { // masteri baglıyken döngü

if (client.available()) { // istemciden okunacak bayt varsa,

char c = client.read(); // sonra bir byte oku

Serial.write(c); // seri monitorde yazdir.

if (c == '\n') { // bayt bir yeni satir ise

if (currentLine.length() == 0) {

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println();

client.print("Click <a href=\"/A\">here</a> A bolgesine git<br>");

client.print("Click <a href=\"/B\">here</a> B bolgesine git<br>");

client.print("Click <a href=\"/C\">here</a> C bolgesine git<br>");

client.println();

// while döngüsünden çıkmak:

break;

} else { // eğer yeni bir hattınız varsa, currentLine öğesini temizleyin:

currentLine = "";

}

} else if (c != '\r') { // satır büyük karakterinden başka birşey varsa,

currentLine += c; // currentLine'nin sonuna ekle

}

if(currentLine.endsWith("GET /A")){

bolge = 1;

while(bolge == 1) {

Serial.println("while A");

olcumeBasla();

if(mesafe>20) {

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 1 && digitalRead(sagsensor) == 0 ){

Serial.println("while A.1");

ileri();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 0){

Serial.println("while A.2");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 1 ){

Serial.println("while A.3");

sag();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 0){

Serial.println("while A.4");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 1 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 0 ){

Serial.println("while A.5");

sol();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 0){

Serial.println("while A.6");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

}

}

}

else if(currentLine.endsWith("GET /B")){

bolge = 2;

while(bolge == 2) {

Serial.println("while B");

olcumeBasla();

if(mesafe>20){

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 1 && digitalRead(sagsensor) == 0 ){

Serial.println("while B.1");

ileri();

delay(60);

if(digitalRead(sagkonumsensor) == 1 && digitalRead(solkonumsensor) == 0 ){

Serial.println("while B.2");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 1 ){

Serial.println("while B.3");

sag();

delay(60);

if(digitalRead(sagkonumsensor) == 1 && digitalRead(solkonumsensor) == 0 ){

Serial.println("while B.4");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 1 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 0 ){

Serial.println("while B.5");

sol();

delay(60);

if(digitalRead(sagkonumsensor) == 1 && digitalRead(solkonumsensor) == 0 ){

Serial.println("while B.6");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

}

}

}

else if(currentLine.endsWith("GET /C")){

bolge = 3;

while(bolge == 3) {

Serial.println("while C");

olcumeBasla();

if(mesafe>20){

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 1 && digitalRead(sagsensor) == 0 ){

Serial.println("while C.1");

ileri();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 1 ){

Serial.println("while C.2");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 0 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 1 ){

Serial.println("while C.3");

sag();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 1 ){

Serial.println("while C.4");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

if(digitalRead(solsensor) == 1 && digitalRead(ortasensor) == 0 && digitalRead(sagsensor) == 0 ){

Serial.println("while C.5");

sol();

delay(60);

if(digitalRead(solkonumsensor) == 1 && digitalRead(sagkonumsensor) == 1 ){

Serial.println("while C.6");

dur();

bolge=0;

Serial.println("bolge=0");

}

}

}

}

}

else bolge=0;

}

}

// bağlantıyı kapat:

client.stop();

Serial.println("client disonnected");

}

}

void olcumeBasla()

{

baslangicTime = 0;

digitalWrite(trigpin,HIGH);

delayMicroseconds(10);

digitalWrite(trigpin,LOW);

sure = pulseIn(echopin,HIGH);

mesafe = (sure/2) / 29.1;

}

void detect ()

{

if(mesafe <= 20){

digitalWrite(motorR1, LOW);

digitalWrite(motorR2, LOW);

digitalWrite(motorL1, LOW);

digitalWrite(motorL2, LOW);

Serial.println("Engel algilandi");

}

}

void ileri()

{

digitalWrite(motorR1, HIGH);

digitalWrite(motorR2, LOW);

digitalWrite(motorL1, HIGH);

digitalWrite(motorL2, LOW);

for (i = 180; i <= 183; i += 1) { //sol motor hız ayarı

ledcWrite(0, i);

delay(20);

}

for (k = 180; k <= 183; k += 1) { //sag motor hız ayarı

ledcWrite(0, k);

delay(20);

}

}

void sol()

{

digitalWrite(motorR1, LOW);

digitalWrite(motorR2, HIGH);

digitalWrite(motorL1, HIGH);

digitalWrite(motorL2, LOW);

for (i = 215; i <= 217; i += 1) { //sol motor hız ayarı

ledcWrite(0, i);

delay(20);

}

for (k = 160; k <= 163; k += 1) { //sag motor hız ayarı

ledcWrite(0, k);

delay(20);

}

}

void sag()

{

digitalWrite(motorR1, HIGH);

digitalWrite(motorR2, LOW);

digitalWrite(motorL1, LOW);

digitalWrite(motorL2, HIGH);

for (k = 215; k <= 217; k += 1) { //sag motor hız ayarı

ledcWrite(0, k);

delay(20);

}

for (i = 160; i <= 163; i += 1) { //sol motor hız ayarı

ledcWrite(0, i);

delay(20);

}

}

void dur()

{

digitalWrite(motorR1, LOW);

digitalWrite(motorR2, LOW);

digitalWrite(motorL1, LOW);

digitalWrite(motorL2, LOW);

}

##  [oguzcanunlu1@gmail.com](mailto:oguzcanunlu1@gmail.com) Cep: (0541) 468 92 31

OĞUZCAN ÜNLÜ

Uyruğu : T.C

Doğum Yeri : İstanbul

Doğum Tarihi : 02/08/1995

Medeni Durum : Bekar

Askerlik : Yapıldı (Yedek Subay)

Ehliyet : B Sınıfı

Sigara : Kullanmıyorum

Adres : Güzelbahçe Mah. Merkez / DÜZCE

**Eğitim Durumu**

|  |
| --- |
|  |

2018 – Anadolu Üniversitesi, İşletme Fakültesi

İşletme Bölümü (Açıköğretim)

2016 – 2019 Marmara Üniversitesi, Teknoloji Fakültesi

(Yatay Geçiş) Mekatronik Mühendisliği Bölümü (GANO: 3,11/4)

2014 – 2016 Kırklareli Üniversitesi, Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü

2009 – 2013 Gaziosmanpaşa Mesleki ve Teknik Anadolu Lisesi, İstanbul Elektrik Elektronik Teknolojileri / Elektrik Tesisatları ve Pano Montörlüğü (Not: 73/100)

**Önyazı**

|  |
| --- |
|  |

Aktif olarak C# yazılım dilinde ve Asp.Net Core MVC üzerinde eğitim serileri izleyerek bu konu hakkında bilgi sahibi olmaya ve kendimi geliştirmeye çalışıyorum. Yazılım alanında kendimi geliştirmek için aktif olarak çalışmalarım devam etmekte.

# İş Deneyimleri

09.2018 – 12.2018 Saf Holland (Stajyer)

* Bakım Mühendisi

06.2018 – 08.2018 Saf Holland (Stajyer)

* Bakım Mühendisi

07.2017 – 08.2017 Şenol Soğutma (Stajyer)

* Pano ve PLC montajı, PLC yazılımı

# Yetkinlikler

|  |  |
| --- | --- |
| **Program** | **Seviye** |
| C# (ASP.NET MVC) | Orta |
| C | Temel |
| AutoCad | Orta |
| PLC Programlama | Orta |
| Excel | İyi |
| Matlab (Resim üzerinde Görüntü İşleme) | Temel |

**Yapılan Projeler**

|  |  |
| --- | --- |
| **Arduino IDE**  **IoT**  **MIT App Inventor** | **Bitirme Projesi:** Uzaktan Kontrollü Çizgi İzleyen Taşıma Aracı  Yapılan bu çalışmada, endüstriyel lojistik faaliyetlerinde kullanılabilecek, önceden belirlenmiş üretim alanlarında taşıma işlevine sahip, belirli bir rotası olan, nesnelerin interneti sayesinde birbiriyle kablosuz haberleşebilen, çalıştığı ortamdaki insanların iş güvenliğini riske atmayan küçük ölçekli bir çizgi takip edebilen taşıma aracı prototipi tasarlanmıştır. |

## Eğitim ve Sertifikalar

## 29.05.2010-21.02.2011 British Time

* Temel İngilizce

## 08.07.2019-14.07.2019 İİEnstitü

* + A1-A2 İngilizce

# 

# 08.07.2019-14.07.2019 İİEnstitü

* + Excel

## 22.07.2019 – 28.07.2019 İİEnstitü

* + B1-B2 İngilizce

## 22.07.2019 – 28.07.2019 İİEnstitü

* + İleri Excel

## Yabancı Dil

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Anlama** | **Konuşma** | **Yazılı Anlatım** |
| **Türkçe** | Anadil | Anadil | Anadil |
| **İngilizce** | Orta | Orta | Orta |

**Referanslar**

|  |  |
| --- | --- |
| **SEÇİL SEVİL** | Saf Holland – İnsan Kaynakları Müdürü |
|  | Telefon:0(532) 137 55 27 |
| **MEHMET ALİ KURNAZ** | Saf Holland – Elektrik Bakım Sorumlusu |
|  | Telefon:0(530) 080 59 81 |