

**KOCAELİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ**

**OTONOM PARALEL PARK YAPAN ARAÇ**

**MÜHENDİSLİK TASARIMI 3 PROJE RAPORU**

**180207004 AHMET ALTAY**

## İÇİNDEKİLER

|   |    |
|---|----|
| İÇİNDEKİLER .....   | 1  |
| 1. TANITIM .....  | 2  |
| 2. DONANIMSAL TASARIM .....                                       | 3  |
| 2.1 Platform .....  | 3  |
| 2.2 Elektronik Malzemeler .....                                   | 4  |
| 2.2.1 Ultrasonik Sensör .....                                     | 4  |
| 2.2.2 Servo Motor .....   | 5  |
| 2.2.3 Voltaj Regülatörü .....                                     | 6  |
| 2.2.4 L298N Motor Sürücü .....                                    | 7  |
| 2.2.5 STM32F407VG Geliştirme Kartı .....                          | 7  |
| 2.2.6 Geliştirme Kartına Elektronik Malzemelerin Bağlantısı ..... | 9  |
| 3. GÖMÜLÜ YAZILIM TASARIMI .....                                  | 10 |
| 3.1 STM32CubeMX .....   | 10 |
| 3.2 Atollic TrueSTUDIO .....                                      | 12 |
| 3.3 PARK SENARYOSU.....   | 13 |
| 4. SONUÇ.....   | 16 |
| KAYNAKÇA.....   | 17 |

## 1. TANITIM

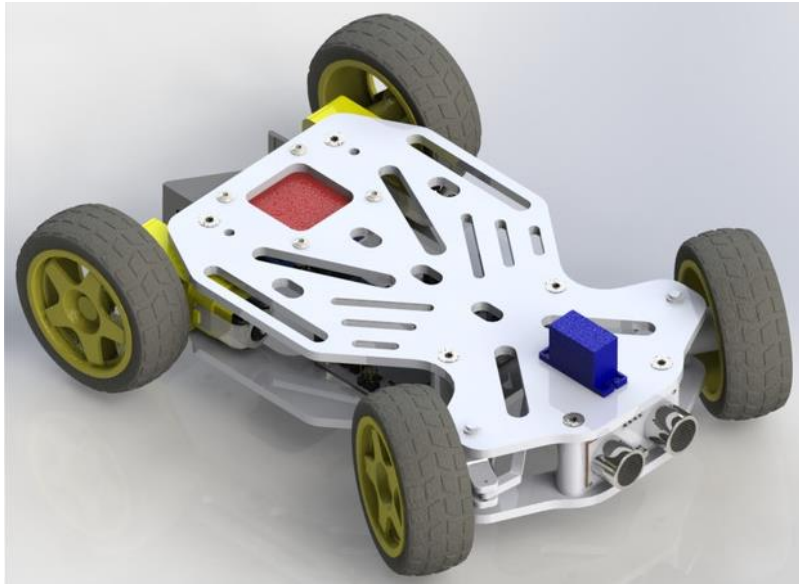
Kullanımdaki araç sayısının gün geçtikçe artması beraberinde çeşitli sorunları getirmiştir. Bu sorunlardan bir tanesi de araçların park etme problemidir. Araçlarda park etme sırasında genellikle küçük maddi kazalar meydana gelmektedir. Araçlara takılan park sensörleri ve geri görüş kameraları sürücüye park esnasında engel olup olmadığı bilgisini çeşitli ikazlarla vererek, park etme işini kolaylaştıran yarı otonom park etme yöntemleri olarak karşımıza çıkmaktadırlar.

Bu projede araba ve benzeri araçların otomatik olarak park etme problemi ele alınmıştır. Aracın kendi kendine park etmesini sağlayan sistem için bir prototip oluşturmak ve bu prototip sayesinde sürücülerin daha kolay bir şekilde araçlarını park edebilmelerini sağlamak çalışma kapsamında hedeflenmiştir. Geliştirilen donanım ve yazılım yardımıyla park edilebilecek kadar bir alanın olup/olmadığının hesaplanması, yeterli alan bulunduğu anda sürücünün herhangi bir müdahalesi olmadan aracın üzerindeki sensörler yardımıyla otomatik park edebilmesi sağlanmıştır.

## 2. DONANIMSAL TASARIM

### 2.1 Platform

Bu proje kapsamında öncelikle kullanılacak sensör ve geliştirme kartının yerleştirilmesi ve aynı zamanda araç olarak kullanılacak şekilde bir platform seçilmesi gerekmektedir. Gerçek araç gibi park işlemi gerçekleşeceğinden servo motor kullanılması gerekmektedir. Bunun için de platformumuzda tekerlerin dönüşü için gerekli boşluk bulunması ve ayrıca ön tekerlerin birbiriyle uyumlu şekilde dönmesi için birtakım parçalar gerekmektedir. Bu istekler doğrultusunda piyasada kullanılan normal platformlar bu proje için işlevsiz hale gelmektedir. Hazır bir araç şasesini kullanmak ise kullanılacak sensörler için yeterli alan barındırmaması ve kullanışsız olmasından dolayı tercih edilmemiştir. Şekillendirilebilme imkânı sayesinde istenilen platformu oluşturma ve uygun fiyatlı olarak parçaların temin edilebilmesinden dolayı 3D yazıcılardan çıkarabilen bir platforma ihtiyaç duyulmuştur. Yapılan araştırma sonucu proje için kullanılacak olan platform için en iyi seçenek seçilmiştir [1]. Yazıcıdan tek tek çıkarılıp alınan parçalar daha sonra manuel olarak birleştirilmiştir. Sonrasında tekerlek ve motor parçaları monte edilmiştir.



Şekil 1: Araç platformu

## 2.2 Elektronik Malzemeler

Kullanılan araç platformunun otonom bir şekilde paralel park yapabilmesi için sensörlere ihtiyaç vardır. Sensör, ortamdaki değişikliği algılayan ve diğer sistemdeki bazı çıkışlara yanıt veren bir alettir. Bir sensör, fiziksel bir olguyu, insan tarafından okunabilen bir ekrana dönüştüren veya daha fazla işlem için iletilen ölçülebilir bir analog voltaja (veya bazen dijital bir sinyale) dönüştürür. Bu projede kullanacağımız sensörleri tanıyalım.

### 2.2.1 Ultrasonik Sensör

Ultrasonik sensörler ya da diğer adıyla ultrasonik dönüştürücüler ses dalgalarını kullanarak mesafeyi ölçen bir sensör türüdür. Radarla çalışma mantığı aynıdır. Hedef nesneye ultrasonik dalgalar yollar ve geri yansımayı elektrik sinyaline dönüştürür. Adından da anlaşılacağı üzere ultrasonik yüksek ses demek. Aynı zamanda yolladığı ultrasonik dalgalar ise insanların duyabileceği ses frekansından daha hızlı bir frekansta hareket eder. Bu ses frekansı 20 kHz ile 500 kHz arasındadır. En çok bilinen ultrasonik sensör olan HC SR04'ün ultrasonik dalga frekansı 40kHz'dir.



Şekil 2: HC-SR04 ultrasonik sensör

Sensörün çalışma prensibi basittir. TRIG pininden sinyal verildiğinde 40 KHz frekansında bir ses dalgası sensör tarafından üretilir ve bu ses dalgası bir cisme çarpıp geri döndüğünde ECHO pini aktif hale gelir.

Sesin havada yayılma hızını bildiğimiz için TRIG pinine verilen sinyalden sonra ECHO pininin aktif olduğu zaman kadar ki süreyi ölçerek aradaki mesafeyi kolayca hesaplayabiliriz. Formülü aşağıdaki gibidir.

$$\text{Uzaklık (cm)} = \text{Zaman } (\mu\text{s}) / 58$$

### 2.2.2 Servo Motor

Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzendir. Servo motorlar, robot teknolojilerinde en çok kullanılan motor çeşidi olmakla birlikte, RC (Radio Control) uygulamalarda da kullanılmaktadırlar. RC Servo Motorlar ilk olarak uzaktan kumandalı model araçlarda kullanılmışlardır. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.

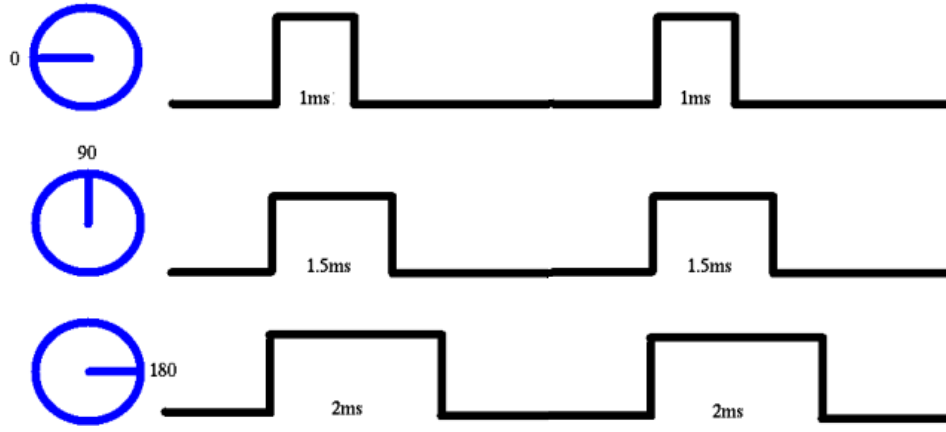
Servo motorlar içerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüyü ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır.



Şekil 3: SG90 servo motor

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolcünden veya uzaktan kumandadan sağlanabilmektedir. Servo, her 20 ms içerisinde bir pals değeri okumaktadır. Pals uzunluğu motorun dönüşünü belirlemektedir. Örnek olarak 1.5 ms'lik bir pals, motorun 90 derece pozisyonunu almasını sağlayacaktır.

Hareket etmeleri için gereken pals genişliklerinin minimumları ve maksimumları vardır ve bu değerler değişkendir. Fakat genellikle minimum pals genişliği 1 ms, maksimum pals genişliği ise 2 ms'dir.



Şekil 4: Duty cycle değerine göre servo motorun konumu

### 2.2.3 Voltaj Regülatörü

Voltaj regülatörü, girilen voltaj değerini istenilen çıkış voltaj değerlerine ayarlayan bir üründür. Regulator kelimesinin Türkçe karşılığı düzenleyici anlamındadır. Yani bu ürünler voltajı düzenlemektedirler. Ancak bu ürünler bahsedilen işlemleri kendi kapasiteleri kapsamında yapabilirler. Lm2596, LM317 vb. yüzlerce çeşit voltaj regüle eden entegreler bulunmaktadır. Her entegre belirli voltaj aralıklarında ve belirli amper değerlerine kadar çalışabilir.

Regülatör çeşitleri; giriş voltajına ne yapılması istendiğine göre değişiklik gösterirler. Başlıca çeşitler:

- Voltaj düşürücü (Step Down)
- Voltaj yükseltici (Step Up - Booster)
- Voltaj Sabitleyici

Voltaj düşürücüler belirli bir giriş voltajını kendi değerlerinin aşağısına indirirler. Örneğin 12V DC bir kaynaktan gelen enerjiyi 5V DC yapmak istiyorsak bir düşürücü kullanmamız gerekir. Bu projede de 9v pil tekerleri çok hızlı döndüreceğinden dolayı bir gerilim düşürücüye ihtiyaç duyulmuştur.

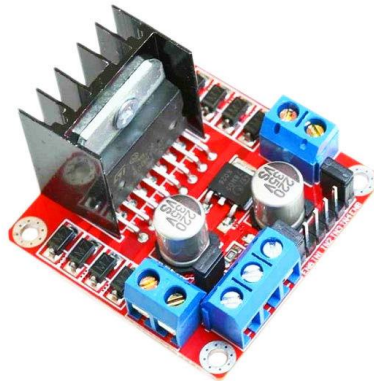


Şekil 5: LM2596 voltaj regülatörü

### 2.2.4 L298N Motor Sürücü Kartı

L298N, 15 uçlu Multiwatt ve PowerSO20 paketlerinde entegre bir monolitik devredir. Standart TTL mantık seviyelerini kabul etmek ve röleler, solenoidler, DC ve kademeli motorlar gibi endüktif yükleri sürmek için tasarlanmış yüksek voltajlı, yüksek akımlı çift tam köprülü bir sürücüdür. Cihazı giriş sinyallerinden bağımsız olarak etkinleştirmek veya devre dışı bırakmak için iki etkinleştirme girişi sağlanmıştır. Her köprünün alt transistörlerinin emitörleri birbirine bağlanır ve karşılık gelen harici terminal, bir harici algılama direncinin bağlantısı için kullanılabilir. Mantığın daha düşük bir voltajda çalışması için ek bir besleme girişi sağlanır.

İki kanallı olarak öne çıkan bu motor sürücü aynı zamanda kanal başına 2A akım veren bir yapıya sahiptir. Birçok farklı alanda devreler için etkin bir çözüm yöntemi sağlayan bu sistem, hazırlanan programın da nasıl kullanılması gerektiği doğrusunda kontrol imkânı sağlamaktadır.



Şekil 6: L298N motor sürücü kartı

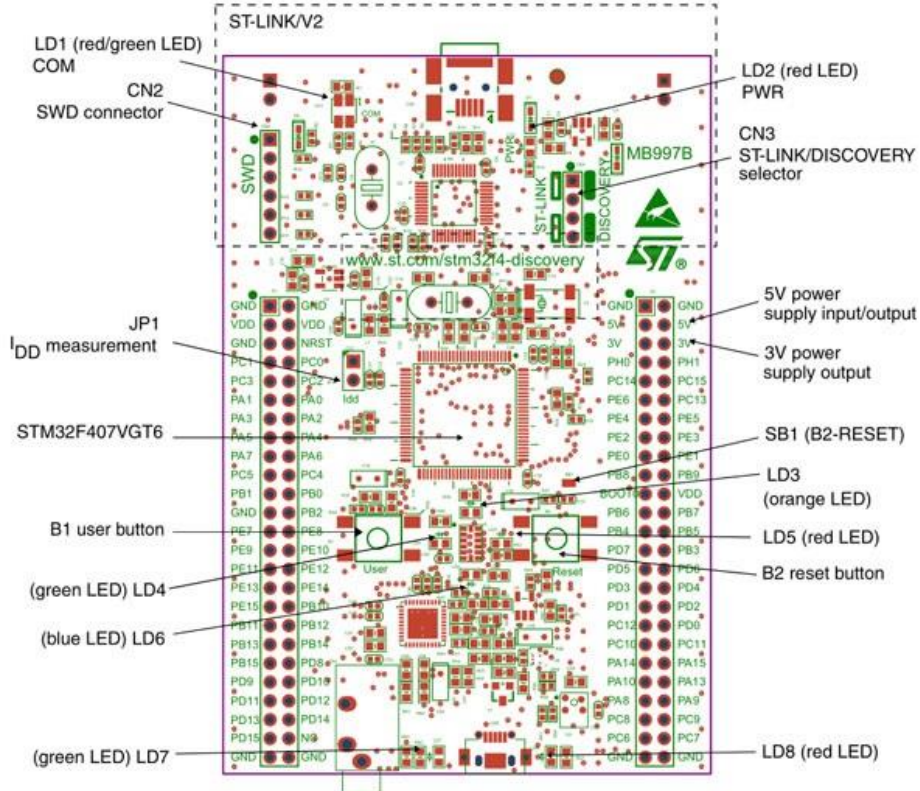
### 2.2.5 STM32F407VG Geliştirme Kartı

STM32F407, 168 MHz saat frekansına sahip 32-bit'lik bir mikrodeneleyicidir. Özellikle DSP uygulamalarında, içerisinde birçok alt sistemler olan yapıların ana beyni olarak kullanılmaktadır. Üzerinde STM32F4VG çekirdeğini barındırmaktadır. 3-eksen ivme ölçer, dahili ses girişi ve buna bağlı dijital analog dönüştürücü, 4 adet programlanabilir LED, 2 adet buton barındırır. ST-LINK V2 kısmında bulunan SWD pinleri ile başka ARM mimarileri de program yapılabilmektedir.



STM32F407 Teknik Özellikleri ise şunlardır.

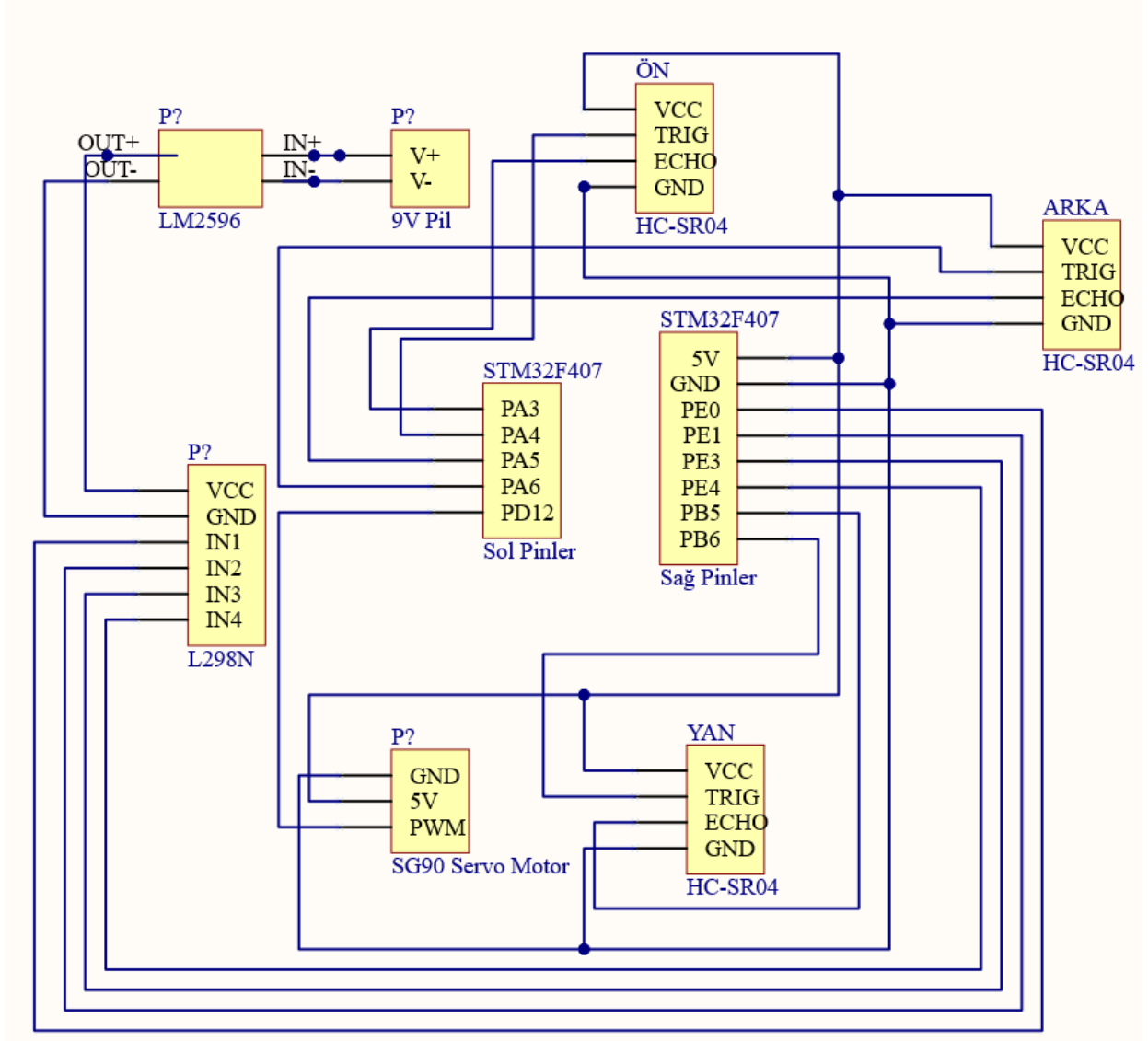
- 32-bit ARM Cortex-M4F çekirdeğine sahip STM32F407VGT6 mikrodenetleyicisi, 1 MB Flash ve 192 KB RAM
- Dahili ST-LINK/V2 JTAG Debugger
- USB veya harici kaynaktan doğrudan 5v ile çalışabilme
- 3 V ve 5 V'luk çıkış pinleri
- 3-eksen dijital ivmeölçer (LIS302DL)
- Omni-Directional Mikrofon (MP45DT02)
- D sınıfı yükselteçli ses sürücü çipi (CS43L22)
- Sekiz adet LED:
  - LD1 (kırmızı/yeşil) (USB haberleşmesi için)
  - 3v3 power on/off LEDi
  - Dört adet kullanıcı LEDi, LD3 (turuncu), LD4 (yeşil), LD5 (kırmızı) ve LD6 (mavi)
  - 2 adet USB OTG LEDi, LD7 (yeşil) ve LD8 (kırmızı)
- Bir adet Reset ve bir adet kullanıcı tanımlı buton
- USB OTG için mikro-AB konnektör.
- 100 pin'in tamamını kullanabilmeye imkân tanıyan çıkışlar.



Şekil 7: STM32F407 şematiği

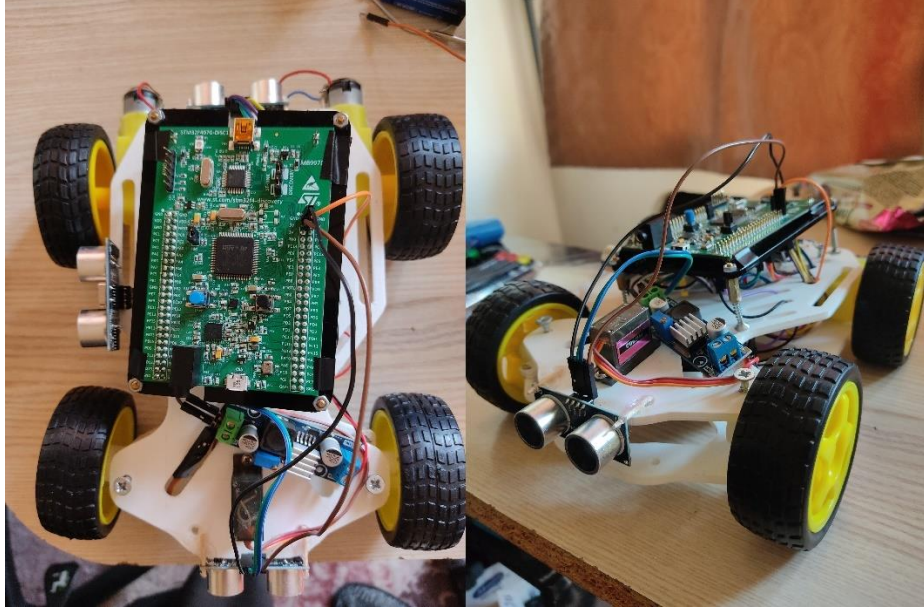
## 2.2.6 Geliştirme Kartına Elektronik Malzemelerin Bağlantısı

Bu proje kapsamında otonom paralel park sisteminin sağlanması ve aracın tasarımında kullanılmak üzere 3 adet HC-SR04 ultrasonik sensör, 1 adet SG90 servo motor, 1 adet L298N motor sürücü, 1 adet STM32F407VG geliştirme kartı ve 1 adet LM2596 voltaj düşürücü regülatör kartı kullanılmıştır. Bu elektronik malzemelerin geliştirme kartımız olan STM32F407VG kartına bağlantısı aşağıdaki Şekil 8’de gösterilmektedir.



Şekil 8: Bağlantı şeması

Bağlantı şemasına göre kartın bağlantısı yapıldıktan ve elimizdeki platformla birleştirdikten sonra aracımız Şekil 9'daki gibi görünmektedir.



Şekil 9: Parçalar monte edildikten sonra araç

### 3. GÖMÜLÜ YAZILIM TASARIMI

Elektronik elemanların STM32F407 geliştirme kartının hangi pinlerine bağlanacağını belirledikten sonra geliştirme kartında kullanacağımız pinleri aktif hale getirmemiz gerekmektedir. Bunun için bize kolaylık sağlayan STM32CubeMX programı kullanılmıştır. Pinler aktif edildikten sonra Atollic TrueSTUDIO üzerinden geliştirme kartından elektronik malzemelerin kontrol edilmesi için gerekli yazılım kısmı yazılmıştır.

#### 3.1 STM32CubeMX

STM32CubeMX, ST firması tarafından geliştirilmiş olup grafiksel ara yüze sahip yazılım yapılandırma aracıdır. Bu arayüz sayesinde genel amaçlı giriş-çıkış pinlerimizi, clock ayarlarımızı, güç tüketimi hesabımızı, MCU da bulunan gerekli çevre birimlerinin yapılandırılmasını (UART, SPI) ve ara katman yığınlarının (USB, TCP/IP) hazırlanması gibi süreçleri daha az efor harcayarak kısa bir zaman diliminde projelendirmemizi sağlar. CubeMX tabanında HAL (Hardware Abstraction Layer) yani Donanım Soyutlama Katmanı kütüphanelerini kapsayan bir proje dosyası oluşturur.

Bu özellikleri sebebiyle STM32CubeMX programı tercih edilmiştir. *Pinout & Configuration* kısmından Şekil 8'de gösterilen bağlantı için STM32F407VG üzerinde kullanılacak pinler aktif edilmiştir. Şekil 10'da ise kullanılacak olan pinlerin hangi modlarda kullanılacağı seçilmiştir.



Yapılan işlemlerin ardından proje ayarları kısmından projenin adı ve konumu belirlenmiş, derleyici olarak Atollic TrueSTUDIO seçilmiştir. Sonrasında *Generate Code* denilerek seçili pinler aktif olacak şekilde derleyiciye kodlar aktarılmıştır.

### 3.2 Atollic TrueSTUDIO

TrueSTUDIO temel olarak Eclipse IDE framework'ünü, GCC (GNU C / C++ Compiler) derleyicisini ve GDB (GNU Debugger Project) gibi bileşenleri içermektedir. Tamamıyla açık kaynaklı bileşenlere dayalıdır. Eclipse, C, C++, Java gibi dillerde de programlama yapmakta yaygın olarak kullanılan bir tümleşik geliştirme ortamıdır. Bu bileşenler dışında içerisinde Atollic ekibinin geliştirdiği bileşenler de yer almaktadır.

STM32CubeMX üzerinden kodları aktardıktan sonra elektronik malzemeler için tanımlama yapmamız gerekmektedir. Öncelikle HC-SR04 ultrasonik sensörün tanımlamaları yapılmıştır. Şekil 11'de bu tanımlamalar gösterilmektedir.

```
uint32_t Read_HCSR04_4() // sureyi tutacagiz
{
    uint32_t local_time = 0; //echodaki sureyi tutacagiz

    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET); // pini tetikledik
    DWT_Delay_us(10); // 10mikro saniye tetiklememiz lazimdi 10msaniye delay yaptik
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET); // pini kapattik

    while(!HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_5)); // A portunun 5.pinin okur deger gelince donguden cikar

    while(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_5)) // echodan deger gelince girer
    {
        local_time++;
        DWT_Delay_us(1); // 1 mikro saniye bekle
        // her mikro saniyede local time artar
    }

    return local_time;
}
```

Şekil 12: HC-SR04 için mesafe ölçüm fonksiyonu

Şekil 11'deki gibi bir fonksiyon tanımlanmış ve Trig pini SET edilerek tetiklenmiştir. 10ms'nin ardından Trig pini RESET edilerek kapanmıştır. Echo pinine Trigden çıkan bir değer geri gelince *local\_time* değişkeni arttırılmıştır. Ardından Şekil 12'deki bir eşitlik ile uzaklık cm cinsinden bulunmuştur.

```
time_3 = Read_HCSR04_3();
distance_3 = time_3 / 58;
```

Şekil 13: Santimetre cinsinden uzaklık hesabı

Servo motor için öncelikle seçilen Timer'ın ve kanalın aktif edilmesi gerekmektedir. Aktif edildikten sonra istenilen konuma göre değerler verilerek servo motor hareket ettirilebilir. Bu projede Timer4'ün 1.kanalı aktif edilmiştir ve verilen değerle servo motorun 90 derecede olmasını sağlamaktadır.

```
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
/* USER CODE END 2 */
TIM4->CCR1=1100;
HAL_Delay(1000);
```

Şekil 14: Servo motoru aktifleştirme

Motor sürücünün In1, In2, In3 ve In4 pinlerinin bağlı olduğu PE0, PE1, PE3 ve PE4 pinlerinin SET ve RESET olmasına göre teker hareket yönü belirlenmektedir. Bu işlemleri bir fonksiyon içinde gerçekleştirerek yazılan kodun temiz bir şekilde olması sağlanmıştır ve kod üzerinde karmaşıklık yaratmasının önüne geçilmiştir. Şekil 14'te hangi durumlarda ileri, geri veya durma pozisyonunda olacağı gösterilmiştir.

```
void Teker_geri()
{
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_RESET); // 1. tekerlegi ileri yonde dondur
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4,GPIO_PIN_RESET); // 2. tekerlegi ileri yonde dondur
}

void Teker_ileri()
{
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_SET); // 1. tekerlegi geri yonde dondur
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_SET); // 2. tekerlegi geri yonde dondur
}

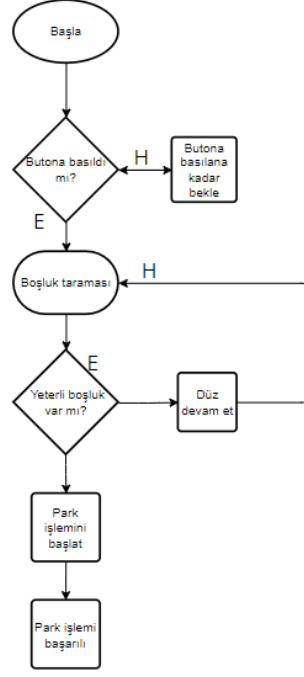
void Teker_dur()
{
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1,GPIO_PIN_RESET); // 1.tekerlegi durdur
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_RESET); // 2. tekerlegi durdur
}
```

Şekil 15: Teker hareketleri

### 3.3 Park Senaryosu

Gerekli tüm elektronik malzemeler yazılan kod üzerinde tanımlandıktan sonra sıradaki işlem park senaryosunun canlandırılmasıdır. Araç park işlemine PA0 pinine bağlı olan aynı zamanda STM32F407VG kartında bulunan *User Button*'a basıldıktan sonra başlayacak şekilde ayarlanmıştır. Park işlemi için oluşturulan algoritma basit bir şekilde oluşturulmuş akış diyagramı ile Şekil 16'te gösterilmektedir.





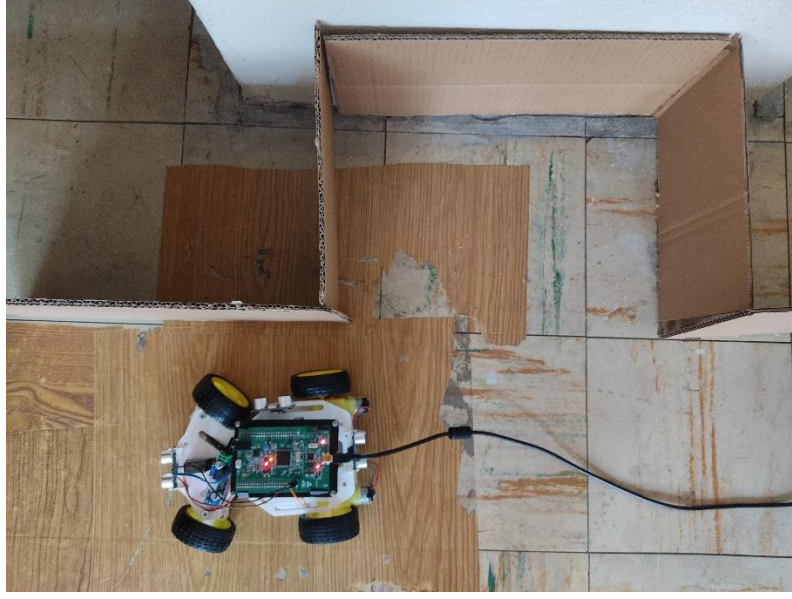
Şekil 16: Basitleştirilmiş akış diyagramı

Öncelikle araç park edebileceği yerin olup olmadığını kontrol etmektedir. Bu koşul sağlanmadığı durumda yoluna devam etmesi gerekmektedir.



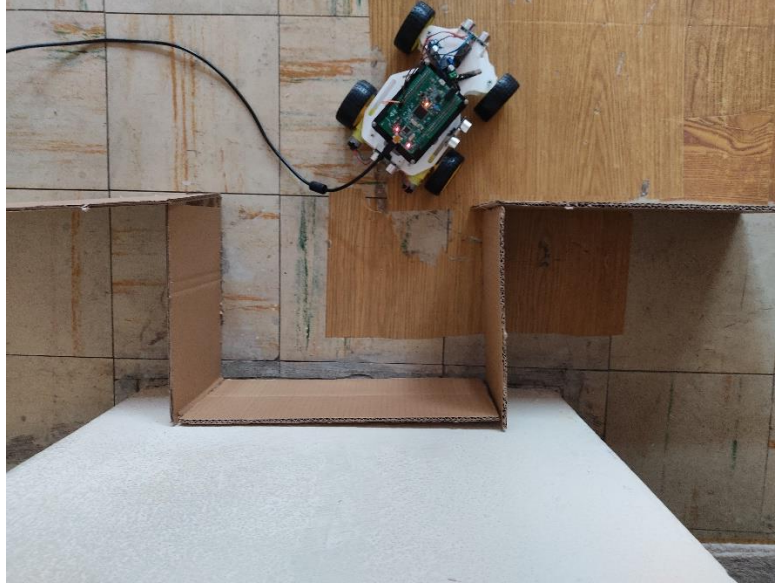
Şekil 17: Yeterli alan bulunmaması

Park yerinin araç boyutlarına uygun olması durumunda araç öncelikle durma durumuna geçmektedir. Ardından araç tekerleğini park yerine girecek konumda ayarlayarak servo motoru 0 dereceye getirip geri gelme durumuna geçmektedir.



*Şekil 18: Servo motor 0 derece*

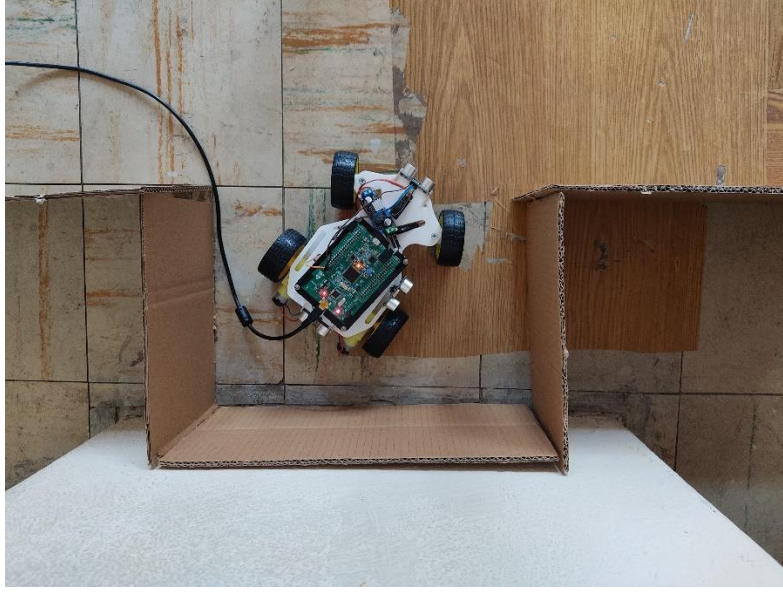
Yeteri kadar geri gelerek döndükten sonra servo motorun konumunu tekrar 90 dereceye getirerek park alanının içine girmektedir.



*Şekil 19: Servo motor 90 derecede geri gelme*

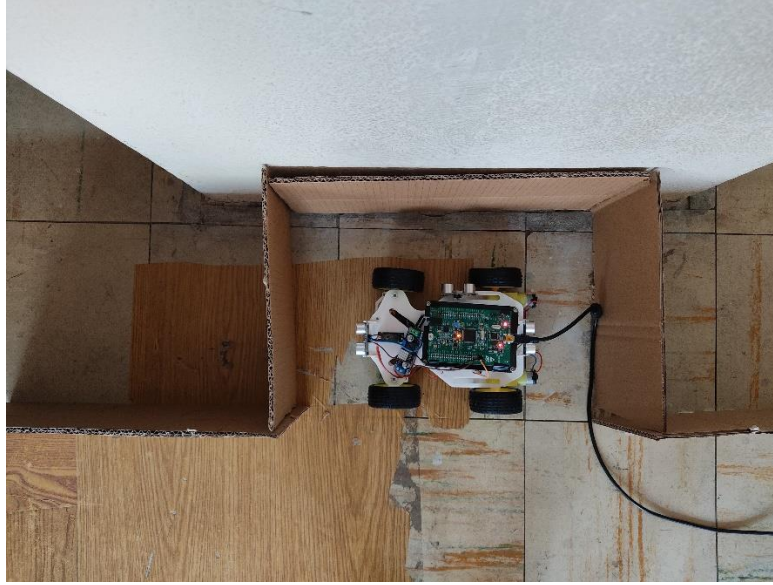
En son olarak servo motorun konumunu 180 dereceye getirip park yerine tam olarak giriş yaparak park işlemini gerçekleştirmektedir.





*Şekil 20: Servo motor 180 derecede park alanına tam giriş*

Bu işlemden sonra park işlemi tamamlanmış olup aracımız paralel parkı başarı ile yerine getirmiştir.



*Şekil 21: Park işleminin başarılı olması*

#### **4. SONUÇ**

Bu proje süresince araçların kendi başlarına park edebilecekleri uygun bir alan bulması ve bu park işlemini paralel ve otonom bir şekilde yapabilmesi üzerine bir araç prototipi geliştirilmiştir. Geliştirilen donanım ve kartta bulunan yazılım sayesinde prototip aracın sığabileceği bir park alanın tespitini yapması, yeterli alan bulunduğu ise herhangi bir müdahale olmadan araçta bulunan sensörler yardımıyla otomatik bir şekilde paralel park etmesi sağlanmıştır. Yapılan uygulama çalışmalarında farklı durumlar denenmiş ve bu durumlar için geliştirilen prototip başarılı olmuştur.

## KAYNAKÇA

[1] Grabcad, Matteo Parmeggiani, “3D Printed RC Car”, erişim: 20 Nisan 2022, <https://grabcad.com/library/3d-printed-rc-car-1>

[2] Yüzgeç, U. Ve Şahin, S., (2017), “Mobil Otonom Park Etme (MOPA) Uygulaması Geliştirilmesi”, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendislik Dergisi, c. 9, no. 2, s. 1-7, 2016.

[3] Çakıl, O.E., ARM CORTEX-M4 ile Mikrodenetleyici Programlama, Kodlab, 2021.

[4] Udemy, Muhammed Fatih Köseoğlu, “STM32F4 Discovery Kartı ile Arm Programlama Dersleri”, erişim: 26 Mart 2022, [www.udemy.com/course/stm32f4-discovery-kart-ile-arm-dersleri/](http://www.udemy.com/course/stm32f4-discovery-kart-ile-arm-dersleri/)