



# Enkazlarda İnsan Tespiti ve Görüntü Aktarımı Yapan Küçük Robotlar

Yazılım Mühendisliği

Dönem Projesi

Alphan SARIDOĞAN

ORCID 0009-0000-4115-5875

Proje Danışmanı: Doç. Dr. Sıla ÖVGÜ KORKUT

Ocak 2024

# Enkazlarda İnsan Tespiti ve Görüntü Aktarımı Yapan Küçük Robotlar

## ÖZ

Bu çalışma, bir arama-kurtarma robotunun tasarım ve geliştirilmesi sürecini ele alarak, robotun mevcut yeteneklerini ve potansiyel geliştirmeleri değerlendiriyor. Robot, böceklerden ilham alarak geliştirilen altı bacaklı bir sistem üzerine kurulmuş ve tripod yürüyüş prensibini benimsemiştir. Yürüme ve dönme yetenekleri ile wifi bağlantısı ve görüntü aktarımı gibi temel özelliklere sahip olan robot, PLA(Biopolimer Polilaktik Asit) baskı malzemesinin sınırlamaları nedeniyle yerde kayma sorunu yaşamaktadır. Bu sorunun giderilmesi için kaymaz bir malzeme kullanımı önerilmekte ve mevcut tasarımın daha fazla stabilite kazanması üzerinde çalışmalar önerilmektedir. Ayrıca, uzanma ve tırmanma yetenekleri gibi ek fonksiyonlar eklenerek robotun işlevselliği genişletilmelidir. Bu çalışma, arama-kurtarma operasyonlarında kullanılacak daha güçlü, esnek ve akıllı robotların geliştirilmesine yönelik bir adım olup, gelecekteki çalışmalara ilham kaynağı olabilir.

**Anahtar Sözcükler:** Arama-kurtarma robotu, hexapod, tripod yürüyüş prensibi

# Small Robots Performing Human Detection and Image Transmission in Disaster Debris

## Abstract

This study focuses on the design and development process of a search and rescue robot, evaluating its current capabilities and potential improvements. The robot is built upon a six-legged system inspired by insects, adopting the tripod walking principle. Equipped with basic features such as walking, turning, wifi connectivity, and image transmission, the robot faces challenges related to sliding on the ground due to limitations of the PLA(Polylactic acid) printing material. To address this issue, the use of a non-slip material is recommended, and efforts to enhance the current design's stability are suggested. Furthermore, additional functionalities such as reaching and climbing abilities should be incorporated to expand the robot's functionality. This study serves as a step towards the development of more robust, flexible, and intelligent robots for search and rescue operations, potentially inspiring future research endeavors.

**Keywords:** Search and rescue robot, hexapod, tripod walking principle

# Teşekkür

Bu proje sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam sayın Doç. Dr. Sıla Övgü Korkut'a sonsuz teşekkür ve saygılarımı sunarım.

Yüksek lisans eğitimim boyunca yardım, bilgi ve tecrübeleri ile bana sürekli destek olan başta Prof. Dr. Aytuğ Onan olmak üzere Yazılım Mühendisliği bölümündeki tüm hocalarıma teşekkür ederim.

Eğitimime devam etmem için beni teşvik eden ve her türlü imkânı sağlayan şirketim Intecon Information Technology & Consulting'e teşekkürü bir borç bilirim.

Çalışmalarım boyunca yardımları ve manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme ve İrem Yürük'e de teşekkür ederim.

# İçindekiler

Öz .....	i
Abstract .....	ii
Şekiller Listesi .....	v
Tablolar Listesi .....	vi
Kısaltmalar Listesi .....	vii
<b>1 Giriş .....</b>	<b>1</b>
<b>2 Materyal ve Metot .....</b>	<b>4</b>
2.1. Donanım .....	5
2.1.1. Merkezi Kontrol .....	5
2.1.2. Hareket Mekanizmaları.....	9
2.1.3. Haberleşme ve Görüntü iletme Mekanizması .....	11
2.2. Tasarım .....	13
2.2.1 Hareket Tasarımı .....	15
<b>3 Sonuç.....</b>	<b>18</b>
<b>Kaynaklar .....</b>	<b>21</b>
<b>Ekler .....</b>	<b>24</b>

# Şekiller Listesi

Şekil 2.1: Arduino Mega .....	6
Şekil 2.2: SG90 RC Mini (9gr) Servo Motor .....	9
Şekil 2.3: Robotun Bacağı.....	10
Şekil 2.4: PCA9685 16 Kanal I2C PWM/Servo Sürücü Kartı.....	11
Şekil 2.5: ESP32-CAM WiFi Bluetooth Geliştirme Kartı .....	12
Şekil 2.6: ESP32-CAM ve OV2640 Kamera Modül .....	13
Şekil 2.7: Robot bacağının anatomik yapısı.....	14
Şekil 2.7: Tipik bir böcek bacağının anatomik yapısı.....	14
Şekil 2.8: Robot genel tasarım.....	15
Şekil 2.9: Drosophila melanogaster yürüyüş yapısı .....	16
Şekil 2.10: Robotun senkron çalışan tripod yürüme mekanizması .....	16
Şekil 2.11: Robotun senkron çalışan tripod yürüme simülasyonu .....	17
Şekil 3.1: Prototip 1 .....	19
Şekil 3.2: Prototip 1 .....	20

# Tablolar Listesi

Tablo 2.1: Kullanılan malzemeler listesi. ....	5
--	---

# Kısaltmalar Listesi

PLA	Biopolimer Polilaktik Asit
PWM	Pulse Width Modulation
I/O	Giriş Çıkış pinleri



# Bölüm 1

## Giriş

Günümüzde, robot teknolojisi giderek karmaşıklaşmakta ve çok çeşitli alanlarda kullanım bulmaktadır. Bu çeşitlenme, özellikle biyolojik sistemlerden ilham alan ve doğadan öğrenen robot tasarımları ile dikkat çekmektedir. Bu bağlamda, enkaz bölgelerine müdahale ve kurtarma operasyonlarını daha etkin hale getirmek amacıyla geliştirilen robotik çözümler, biyomimikri prensiplerinden güç alarak önemli bir ilerleme kaydetmektedir. Bu makalede, biyolojik sistemlerden esinlenen robot tasarımlarının enkaz bölgelerindeki kurtarma operasyonlarında nasıl etkin bir rol oynayabileceği üzerinde durulacak ve bu alandaki son teknolojik gelişmeler ışığında incelenecektir. Biyomimikri odaklı robot teknolojisinin, enkaz altındaki canlıları hızlı bir şekilde tespit etme, görüntüleme ve kurtarma süreçlerini optimize etme potansiyeli, bu alandaki araştırma ve geliştirme çabalarını yönlendiren önemli bir itici güç haline gelmiştir.

Robot teknolojilerinin ilerlemesiyle birlikte, robot çeşitliliği giderek artmaktadır. Bu çalışmada, canlı varlıkların sağlığını olumsuz etkileyebilecek veya ölümcül sonuçlara yol açabilecek ortamlarda kullanılan bir mobil robotun tasarımı ele alınmıştır. Bu robotun temel amacı, çevresindeki nesnelere müdahale etmek ve ortam hakkında bilgi toplamaktır. Bu projenin temel hedefi, biyolojik ilhamdan yola çıkarak geliştirilen robot teknolojisi ile enkaz bölgelerine müdahalede bulunmak ve kurtarma operasyonlarını daha etkin hale getirmektir. Robotlar, otonom veya önceden programlanmış görevleri yerine getirebilen elektromekanik cihazlardır. Kullanıcı ile etkileşimli çalışabildikleri gibi önceden belirlenmiş görevleri de yazılı kodlar çerçevesinde icra edebilirler. Robotlar, insan benzeri bir yapıya sahip olmak yerine çeşitli şekiller alabilen ve görev gereksinimlerine uyum sağlayabilen cihazlardır. Endüstriyel üretimde, özellikle otomotiv sektöründe, robotlar yaygın olarak

kullanılmaktadır. Kol şeklindeki robotlar, parça entegrasyonu, boya işlemleri, kaynak gibi işlemleri gerçekleştirebilirler [1].

Robot teknolojisinin tarihsel evriminde, yapay zekâ önemli bir odak noktası olmuştur. Özellikle mobil robotların karar mekanizmalarının işlevselliği, robotik alanın gelişimini etkileyen kritik konulardan biridir. Mobil robotlar genellikle yapay zekâ ile tasarlanır, ancak bazı durumlarda operatör yardımı olmadan çalışmaları önemlidir. Özellikle bilinmeyen veya ulaşılması zor ortamlar için tasarlanan robotlar, üzerlerindeki çeşitli sensörlerle bu ortamlar hakkında operatöre bilgi iletmelidir.

Bu sensörler arasında ısı, mesafe, sıcaklık, gaz, ivme, basınç, eğim, konum ve ışık gibi çeşitli algılayıcılar bulunmaktadır. Bu algılayıcıların ölçüm verileri, mobil robot üzerindeki mikroişlemci tarafından matematiksel olarak hesaplanır, gerekli dönüşümler yapılır ve operatörün anlayabileceği anlamlı verilere dönüştürülür.

Mobil robotlarda kullanılan kameralar da çeşitli özelliklerle donatılmıştır. Bu kameralar, anlık görüntüleri kaydederek kullanıcıyla iletişim halinde oldukları sürece bilgi aktarır. Kullanıcı, bu görüntüler aracılığıyla robotu kontrol eder ve yönlendirir. Kameralar, hareket algılayıcıları ve gece görüşü gibi özelliklerle kullanıcının ortam keşfinde yardımcı olur.

Boyut sınırlaması olmayan robotlar için daha gelişmiş kameralar tercih edilirken, küçük boyutlardaki mobil robotlarda mikro işlemci ve algılayıcı sayısı sınırlı olabilir. Bu durum, mobil robotun daha basit hareketlere ve programlamaya sahip olmasına neden olabilir. Ebatları küçüldükçe, robot üzerine eklenen her birim ağırlığı artırabilir ve manevra yeteneğini azaltabilir. Bu nedenle, küçük robotlarda daha sınırlı kapasiteli mikro işlemcilerin kullanılması, algılayıcı verilerinin harici bir bilgisayar veya mikrodenetleyici üzerinde işlenerek robotun işlem kapasitesini optimize etmeyi amaçlar.

Mobil robot yapılarından doğan bir sonuç olarak, kablosuz iletişim kullanıcı ile robot arasında vardır. Bu iletişim genellikle radyo dalgaları üzerinden gerçekleşir. Mobil robotun kullanım alanına göre iletişim kuralları çeşitlilik gösterir. Özellikle erişimi zor ve neredeyse imkânsız bölgelerde kullanılan küçük boyutlu robotlarda görüntü ve veri aktarımı kritik bir öneme sahiptir. Kablosuz iletişim teknolojileri, mobil robotlarla etkileşimde bulunmak için kullanılır.

Chen ve arkadaşlarına göre; "Biological inspiration has spawned a wealth of solutions to both mechanical design and control schemes in the efforts to develop agile legged machines." ifadesiyle, böcek bacaklarının soyut anatomisinden esinlenerek tasarlanan esnek bacak mekanizması ve doğal hareket üretebilen bir planlama algoritması sunulmaktadır [2]. Pavan ve arkadaşlarına göre; "To escape danger or catch prey, running vertebrates rely on dynamic gaits with minimal ground contact." ifadesiyle, böceklerin üçlü adım atma yöntemini vurgulayarak, bu tür bir hareketin dikey zemin tırmanma yeteneğini hızlandırabileceğini ortaya koymaktadır [3]. Vidoni ve arkadaşlarına göre; "Legged walking and climbing robots have recently achieved important results and developments, but they still need further improvement and study." ifadesiyle, örümcek bacaklarının yapısını dikkate alarak, ayak kuvveti ve tork dağılımını inceleyerek biyomimikri prensiplerini kullanmaktadır [4]. Kemal ve arkadaşlarına göre; "Time is very significant for living things that are left under the rubble after earthquakes." ifadesiyle, deprem sonrası enkaz altında kalan canlıların hızlı bir şekilde tespit edilmesi ve kurtarılması için geliştirilen bir robot prototipini tanıtmaktadır [5]. Bu prototip, özellikle dar alanlardan geçiş yapabilen ve çeşitli sensörlerle çevresini izleyen bir tasarıma dayanmaktadır. Bekir Kuzu'ya göre; "Robotların günlük hayatımızdaki önemi artmaktadır." ifadesiyle, genel olarak robotların günlük hayattaki önemine vurgu yaparak, kullanım alanlarının sürekli olarak genişlediğini ifade etmektedir [6]. Bu tez çalışması kapsamında, bir kamera entegrasyonu ve paletli hareket mekanizmasıyla engelli arazide ilerleme yeteneği sağlayan bir robot tasarımını ele almaktadır.

Bu kaynaklar, biyolojik sistemlerden ilham alınarak geliştirilen robot teknolojilerinin, enkaz bölgelerindeki kurtarma operasyonlarını daha etkili hale getirmek adına çeşitli yöntem ve stratejilerle incelendiğini göstermektedir. Bu bağlamda, biyomimikri prensiplerinin ve doğadan alınan ilhamların, robotik sistemlerin hareket kabiliyetini ve etkinliğini artırmak için önemli bir rol oynadığı vurgulanmaktadır. Bu çeşitli yaklaşımların bir araya gelmesi, robot teknolojisinin kurtarma operasyonlarına daha etkin bir şekilde hizmet etmesine olanak tanımaktadır.

## Bölüm 2

### Materyal ve Metot

Bu bölüm, doğadan ilham alarak geliştirilen bir 6 bacaklı robotun tasarım ve uygulama sürecini içermektedir. Robot, Arduino Mega 2560 R3 (Klon) mikrodenetleyici, 18 adet SG90 RC Mini (9gr) servo motor, ESP32-CAM WiFi Bluetooth geliştirme kartı, OV2640 kamera modülü, PCA9685 16 kanal I2C PWM/Servo sürücü kartı (Klon), ve 12V 5A batarya gibi çeşitli bileşenler kullanılarak oluşturulmuştur. Robotun bacak hareketlerini doğadaki 6 bacaklı böceklerden esinlenerek her eklemde servo motor olacak şekilde tasarlanmıştır ve bu sayede robotun uzanabilmesi, engelleri aşabilmesi ve basit tırmanışlar yapabilmesini sağlamaktır.

Arduino Mega 2560 R3 (Klon) mikrodenetleyici, projenin kontrol merkezi olarak işlev görmekte olup, 18 adet SG90 RC Mini servo motorlar, bacak eklemlerinin hareketini sağlamaktadır. PCA9685 16 kanal I2C PWM/Servo sürücü kartı, servo motorların senkronizasyonunu sağlayarak robotun daha hızlı, sağlam ve dengeli adımlar atmasına olanak tanımaktadır. ESP32-CAM WiFi Bluetooth geliştirme kartı ve OV2640 kamera modülü, robotun çevresini gözlemlemesini ve bu görüntüleri anlık olarak bir telefona aktarmasını mümkün kılmaktadır.

## 2.1. Donanım

Donanım, robotu oluşturan tüm fiziksel birimlerdir. Robotun yapım aşamasında Tablo 2.1 de belirtilen parçalar kullanılmıştır. Her parça için açıklamaları kendi başlıklarında verilmiştir.

Tablo 2.1: Kullanılan malzemeler listesi.

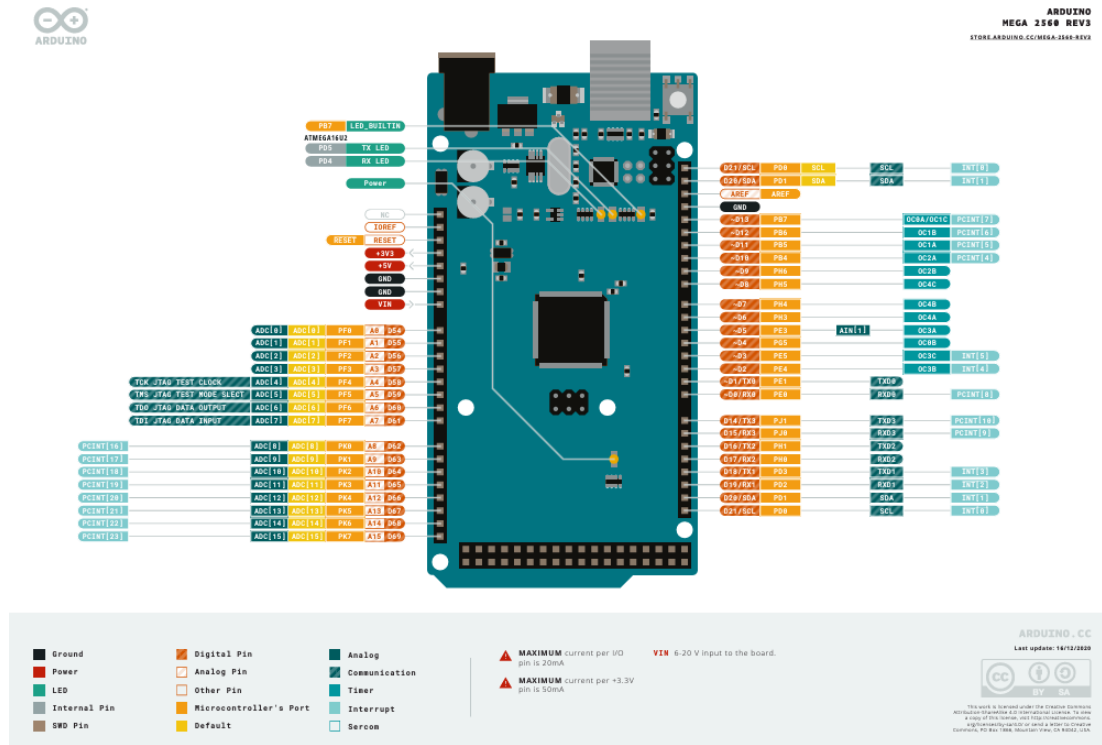
<i>Malzeme</i>	<i>Model</i>	<i>Adet</i>
Arduino	Mega 2560 R3 (Klon)	1
Servo Motor	SG90 RC Mini (9gr)	18
Wifi-Bluetooth	ESP32-CAM	1
Kamera Modül	OV2640	1
Servo Sürücü Kartı	PCA9685 I2C PWM	1
Batarya	12V 5A	1

### 2.1.1. Merkezi Kontrol

Arduino Mega'nın tercih edilmesindeki temel neden, projenin servo motorlarını kontrol etmek için ihtiyaç duyulan fazla sayıda PWM (Pulse Width Modulation) pinine olan ihtiyacı karşılamasıdır. Arduino Uno gibi diğer Arduino modellerinde az sayıda PWM pininin bulunması, projenin karmaşıklığını artırabilirdi. Ancak, Arduino Mega 2560 R3, 16 adet PWM pinine sahip olmasıyla öne çıkar.

Servo motorlar, belirli bir pozisyonu elde etmek için PWM sinyalleri kullanır. Dijital pinlerin 0 ila 5 volt arasında sadece iki durumu (0 veya 5V) kontrol edebildiği düşünüldüğünde, bu servo motorların kontrolü için yetersiz olabilir. PWM tekniği,

belirli bir zaman diliminde voltajın değiştirilmesi yoluyla analog bir etki yaratır. Arduino Mega'nın 16 adet PWM pinine sahip olması, servo motorların daha hassas bir kontrolünü mümkün kılar. PWM sinyalleri, 0 ile 5 volt arasındaki voltajı 0 ile 255 arasında bir değere bölerek, servo motorlara belirli bir derecede dönme komutu iletilmesini sağlar. [7]



Şekil 2.1: Arduino Mega [8]

### Teknik Özellikler:

- Mikrodenetleyici Atmega2560
- Çalışma Gerilimi 5V
- Giriş Gerilimi (önerilen) 7-12V
- Giriş Gerilimi (limit) 6-20V
- Dijital I/O Pinleri 54 (15 tanesi PWM çıkışı)
- Analog Giriş Pinleri 16
- Her I/O için Akım 40 mA
- 3.3V Çıkış için Akım 50 mA

- Flash Hafıza 256 KB (Atmega2560) 8 KB kadarı bootloader tarafından kullanılmaktadır
- SRAM 8 KB (ATmega2560)
- EEPROM 4 KB (ATmega2560)
- Saat Hızı 16 MHz
- Uzunluk 101.6 mm
- Genişlik 53.4mm
- Ağırlık 36 g

Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasındır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Mega kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. Vin pini veya güç soketi üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağılır.

Güç pinleri aşağıdaki gibidir:

- VIN: Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
- 5V: Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
- 3V3: Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
- GND: Toprak pinleridir.

Mega üzerindeki 54 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. 16 tane analog giriş pini de bulunmaktadır. Bu analog giriş pinleri de aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 70 tane dijital giriş çıkış pini vardır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maks. 40mA giriş ve

çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir. [8] [7] [9]

- Seri Haberleşme, Serial: 0 (RX) ve 1 (TX), Serial1: 19 (RX) ve 18 (TX), Serial2: 17 (RX) ve 16 (TX), Serial3: 15 (RX) ve 14 (TX) : TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Pin 0 ve 1 doğrudan kart üzerinde bulunan Atmega16u2 usb-seri dönüştürücüsüne bağlıdır. Yani bilgisayardan karta kod yüklerken veya bilgisayar-mega arasında karşılıklı haberleşme yapılırken de bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.
- Harici Kesme, 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2): Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için attachInterrupt() fonksiyon sayfasını inceleyebilirsiniz.
- PWM, 2-13 ve 44-46: 8-bit çözünürlükte PWM çıkış pinleri olarak kullanılabilir.
- SPI, 53 (SS), 51 (MOSI), 50 (MISO), 52 (SCK): Bu pinler SPI haberleşmesi için kullanılır.
- LED, 13: Mega üzerinden 13. pine bağlı olan dahili bir led bulunmaktadır. Pin HIGH yapıldığında led yanacak, LOW yapıldığında led sönecektir.
- Analog, A0-A15: Mega 16 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur. AREF pini ve analogReference() fonsiyonu kullanılarak alt limit yükseltip, üst limit düşürülebilir.
- I2C, 20 veya SDA pini ve 21 veya SCL pini: Bu pinler I2C haberleşmesi için kullanılır.
- AREF: Analog giriş için referans pini.
- Reset: Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.

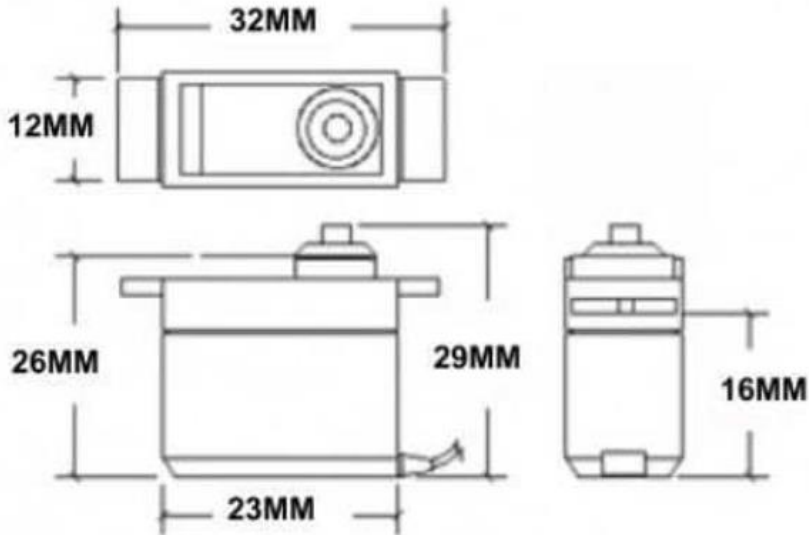


## 2.1.2. Hareket Mekanizmaları

Projenin temelini oluşturan hareket mekanizmaları, her bir bacakta kullanılan 3 adet servo motor ile sağlanmaktadır. Toplamda 6 bacak bulunması, projenin ileri seviye hareket kabiliyeti ve adaptasyon yeteneği ile donatılmasına olanak tanımaktadır. Her bir servo motor, robotun bacak eklemlerini kontrol etmek ve özelleştirilmiş hareketler gerçekleştirmek için tasarlanmıştır.

Projede kullanılan SG90 RC Mini (9gr) servo motorlar, kompakt tasarımı ve hafif yapısıyla dikkat çeker. Bu servo motorlar, 0 ila 180 derece arasında dönebilen, birçok uygulamada tercih edilen standart bir modeldir. Projede 18 adet servo motor kullanılması, her bir bacağın bağımsız hareket edebilmesi ve robotun çeşitli zorlu yüzeylerde adaptasyon yeteneğini artırabilmesi için stratejik bir tercihi temsil etmektedir.

SG90 servo motorlarının 9 gram ağırlığı, projenin genel ağırlığını optimize ederken, 0.12 saniyelik tepki süresi ve yüksek tork değeri, robotun hızlı ve etkili hareket etmesini sağlar. Bu özellikler, projenin hedeflerine ulaşması ve karmaşık ortamlarda başarılı bir şekilde gezinmesi için gerekli olan performansı sağlar.



Şekil 2.2: SG90 RC Mini (9gr) Servo Motor [10]

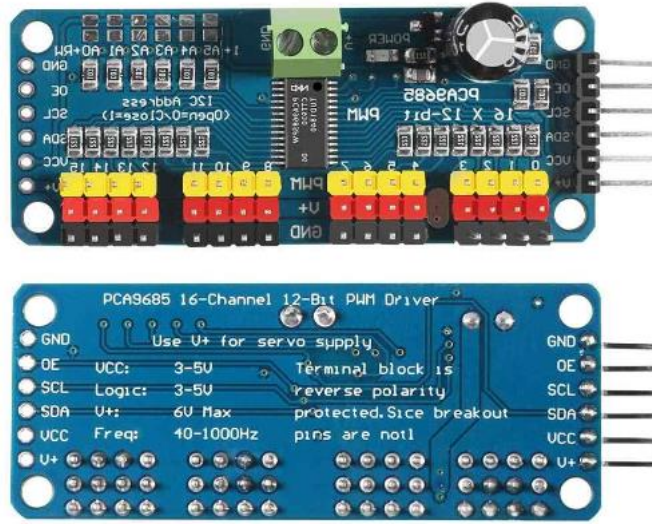
### SG90 Servo Motor Özellikleri:

- Boyutlar: 23.1 x 12.2 x 29 mm
- Ağırlık: 9 g
- Çalışma gerilimi: 4.8 - 6.0 VDC
- Hız @4.8V: 0.1 sn/60°
- Zorlanma Torku @6V: 1.8 kg.cm
- Dişli kutusu: Plastik
- Dönüş açısı: 0-180°
- Çalışma PWM sinyali: 500-2400  $\mu$ s
- Kablo Uzunluğu: 15 cm



Şekil 2.3: Robotun Bacağı

Servo motorların 0-5 volt arasında gerilimle çalıştığı düşünüldüğünde, projede kullanılan PCA9685 16 kanal I2C PWM/Servo sürücü kartı, servo motorların senkronizasyonunu sağlamak ve kontrol etmek adına önemli bir bileşen olarak belirir. Bu kart, 16 ayrı kanalda PWM sinyali üretebilir, bu sayede her bir servo motorun hassas bir şekilde kontrol edilmesini sağlar. Ayrıca Arduino Megada, Uno'dan ya da diğer modellerden daha fazla PWM pin olsa da yine de 18 servo motorun tamamının bağlanabileceği sayıda pin yoktur. Hem gerekli pin sayısını sağlamak hem de ileride olabilecek olası geliştirmeler için lazım olabilecek pin sayısını sağlamak için PCA9685 16 Kanal I2C PWM/Servo Sürücü Kartı kullanılmaktadır.



Şekil 2.4: PCA9685 16 Kanal I2C PWM/Servo Sürücü Kartı [10]

### 2.1.3. Haberleşme ve Görüntü İletme Mekanizması

Projenin haberleşme ve görüntü iletimi mekanizmasında ESP32-CAM WiFi Bluetooth Geliştirme Kartı ve OV2640 Kamera Modülü kullanılmıştır. Bu entegre sistem, projenin temel özelliklerini güçlendirmek ve kullanıcı ile robot arasında etkileşimi sağlamak amacıyla tercih edilmiştir. ESP32-CAM, hem kamera modülü hem de WiFi

ve Bluetooth bağlantılarını tek bir kart üzerinde birleştirerek, projenin karmaşıklığını azaltmış ve enerji verimliliğini artırmıştır.

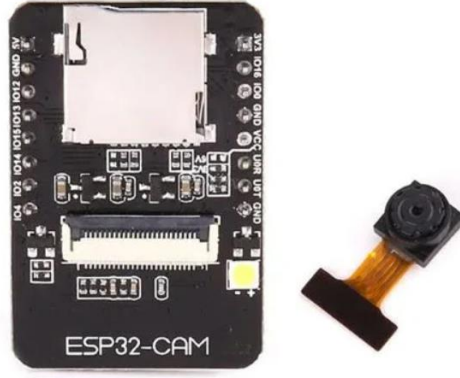
ESP32-CAM WiFi Bluetooth geliştirme kartı, ESP32 tabanlı bir mikrodenetleyici ve kamera modülünü içeren entegre bir karttır. Bu kart, WiFi ve Bluetooth iletişim protokollerini destekler ve aynı anda bu iki iletişim yöntemini kullanabilir. Bu özellik, robotun kablosuz ağlara bağlanmasını ve uzaktan kontrol edilmesini sağlar. Ayrıca, Bluetooth bağlantısı sayesinde yakın mesafelerdeki kontrol uygulamaları için de kullanılabilir.



Şekil 2.5: ESP32-CAM WiFi Bluetooth Geliştirme Kartı

ESP32-CAM'nin WiFi ve Bluetooth yetenekleri, robotun kablosuz bir şekilde ağa bağlanmasını sağlar. Bu özellik, robotun uzaktan kontrol edilmesine ve canlı görüntü akışının bir mobil cihaza aktarılmasına olanak tanır. Bu modülün seçilmesinin nedeni budur.

OV2640 kamera modülü, yüksek çözünürlüklü görüntüler ve video kaydı sağlayan bir CMOS sensöre sahiptir. Bu modül, projenin temel amacı olan robotun çevresini görsel olarak algılamasını sağlar. Görüntüler, ESP32-CAM kartı tarafından işlenir ve istemci cihazlara aktarılır.



Şekil 2.6: ESP32-CAM ve OV2640 Kamera Modül

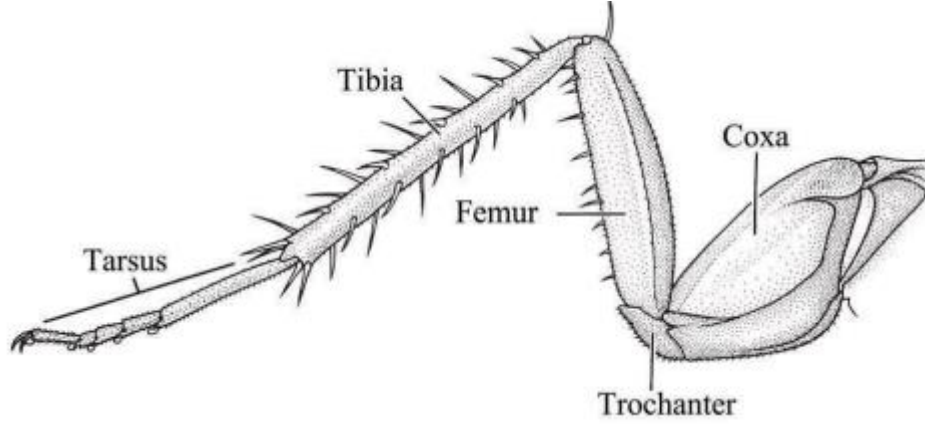
## 2.2 Tasarım

Hayvanların doğal çevreye hızla uyum sağlama yeteneği, mühendislerin ve robotik uzmanlarının, işlevsel mobil makineler inşa etme çabalarında dikkatini çekmiştir. Ayaklı sistemler, teorik olarak, sürekli temas yollarına ihtiyaç duymamaları nedeniyle geleneksel tekerlekli ve paletli tasarımlara göre engebeli arazileri daha iyi geçme potansiyeline sahiptir; ayrıca biyonik prototipleri doğada şaşırtıcı yeteneklere sahiptir. [11] Genel olarak, daha fazla bacaklı robotlar, düzensiz arazilere daha yüksek uyum sağlamaları anlamına gelir, ancak kaçınılmaz olarak daha karmaşık bir kontrol sistemine yol açar. Tüm ayaklı sistemler içinde, altı bacaklı robotlar, yeterli uyum sağlama ve kontrol karmaşıklığı arasında mantıklı bir denge gibi görünmektedir. [12] [13] Bu proje, altı bacaklı böceklerden esinlenilmiş bir robot tasarımını içermektedir çünkü, sayıca daha az bacaklar denge ve gücü azaltırken sayıca çok bacaklar ağırlığı ve enerji tüketimini arttırmaktadır.

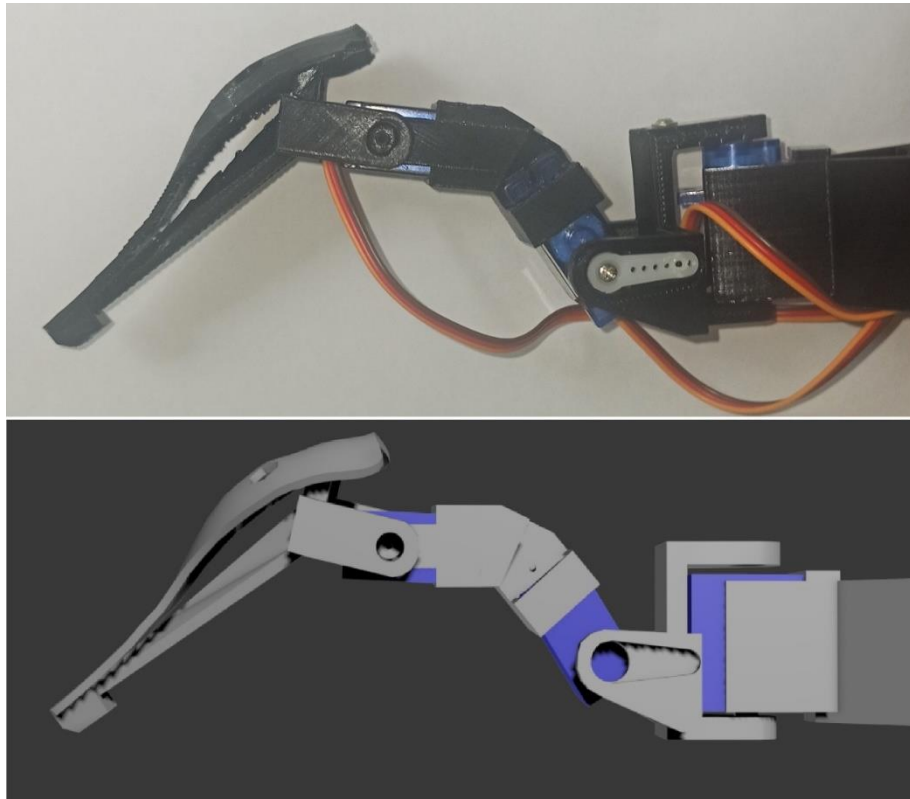
Altı bacak olması, çevik hareketleri ve zorlu arazilerde dolaşabilme yetenekleri, kurtarma operasyonları gibi acil durum müdahalelerinde önemli avantajlar sunabilir.

Böcek benzeri bir robot tasarımı, böceklerin biyolojik özelliklerini taklit ederek enkaz bölgelerinde daha etkili bir şekilde hareket edebilme potansiyeline sahiptir. Böcek bacaklarının esnek yapısı, dar alanlardan geçiş yapma yeteneği ve dikey yüzeylere tırmanabilme özellikleri, robotun enkaz altındaki karmaşık ortamlarda manevra yapmasını kolaylaştırabilir. Bu belirgin özellikleri, bacaklarındaki sofistike kas-iskelet

sistemine oldukça büyük ölçüde dayanmaktadır. [14] Tipik olarak, her böcek bacağı, eklemlerle birbirine bağlı beş temel segmentten oluşur ve bunlar, proksimalden distale şu şekildedir: coxa, trokanter, femur, tibia ve tarsus, Şekil 2.7'de tasvir edildiği gibi.



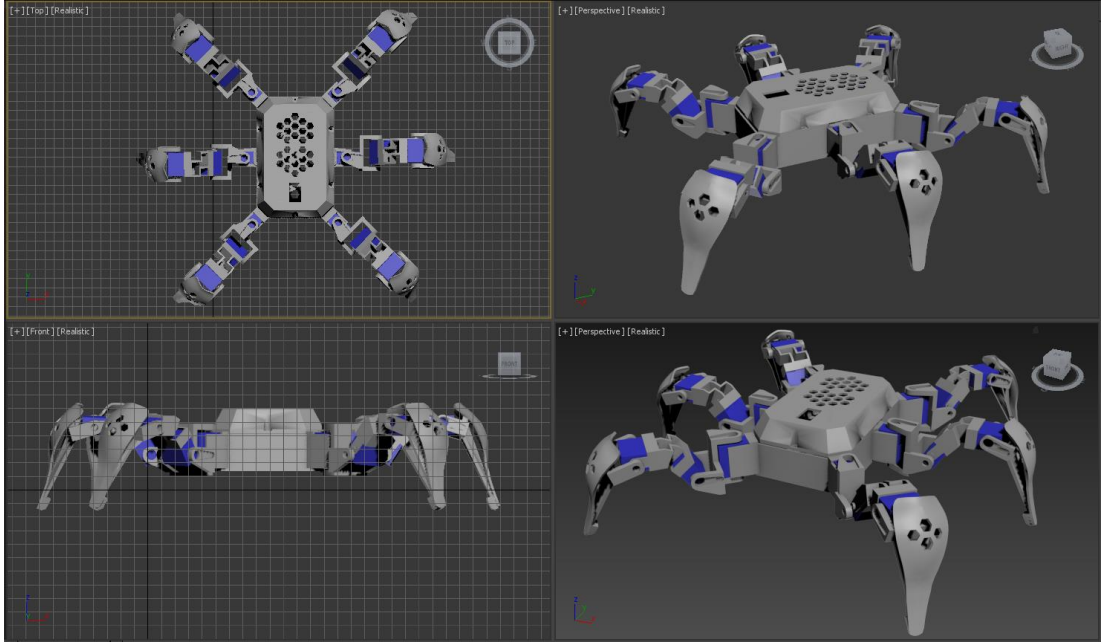
Şekil 2.7: Tipik bir böcek bacağının anatomik yapısı [15]



Şekil 2.7: Robot bacağının anatomik yapısı

Bu şekilde tasarlanan robotlar, enkaz altındaki mahsur kalan insanları daha hızlı ve etkili bir şekilde tespit edebilir, görüntüleyebilir ve kurtarma ekiplerine önemli bilgiler

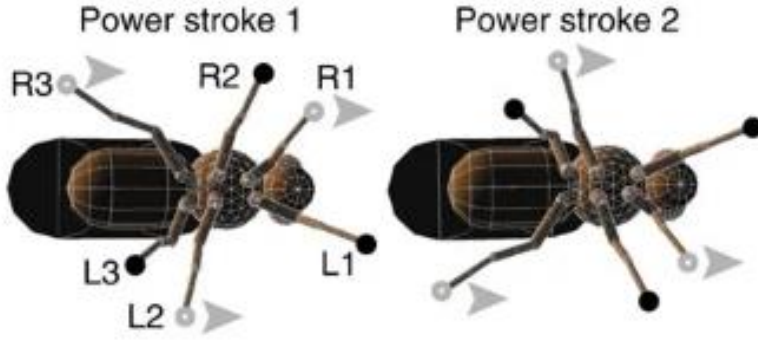
sağlayabilir. Böcek benzeri robotlar, biyolojik sistemlerden ilham alarak doğal çevre koşullarına daha iyi uyum sağlayabilir ve bu da kurtarma operasyonlarını daha başarılı kılabilir. Bu nedenle, örümcek şeklinden esinlenilmiş bir robot tasarımı, enkaz bölgelerindeki kurtarma operasyonlarında daha etkili bir çözüm sunma potansiyeline sahiptir.



Şekil 2.8: Robot genel tasarım

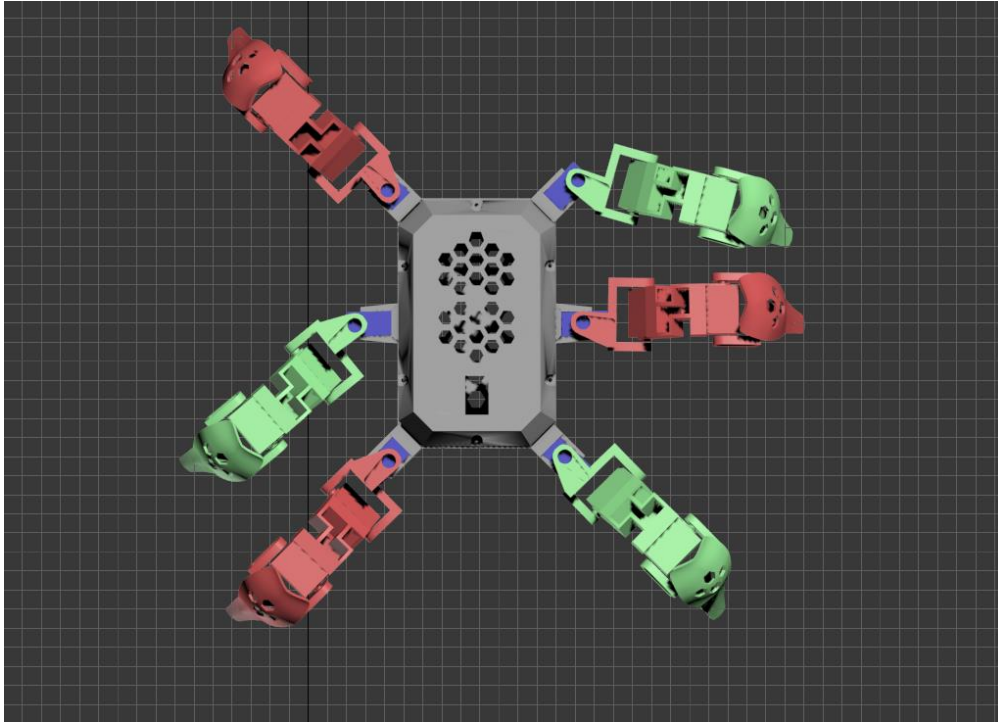
## 2.2.1 Hareket Tasarımı

Robotun hareket mekanizması tasarlanırken, böceklerin hareket stratejilerinden ilham alındı ve özellikle *Drosophila melanogaster* gibi böceklerin tripod yürüyüş ilkesi benimsendi. Altı bacaklı robot, her biri üçer bacak olmak üzere iki set halinde düzenlenmiş ve bu setler arasında senkronize bir hareketle dengiyi koruyan bir sistem üzerine kuruldu. Her bacak seti, birbirine yakın konumda bulunan üç bacak içerir ve bu bacaklar, böceklerin doğal tripod yürüyüşünden esinlenerek koordineli bir şekilde hareket eder. Bu tasarım, robotun zorlu arazilerde dengeli ve hızlı bir şekilde hareket etmesini sağlamak için tripod prensiplerini benimseyerek, böceklerin doğadaki başarılı lokomotor stratejilerinden etkilenmiştir. [16] [17] [18]



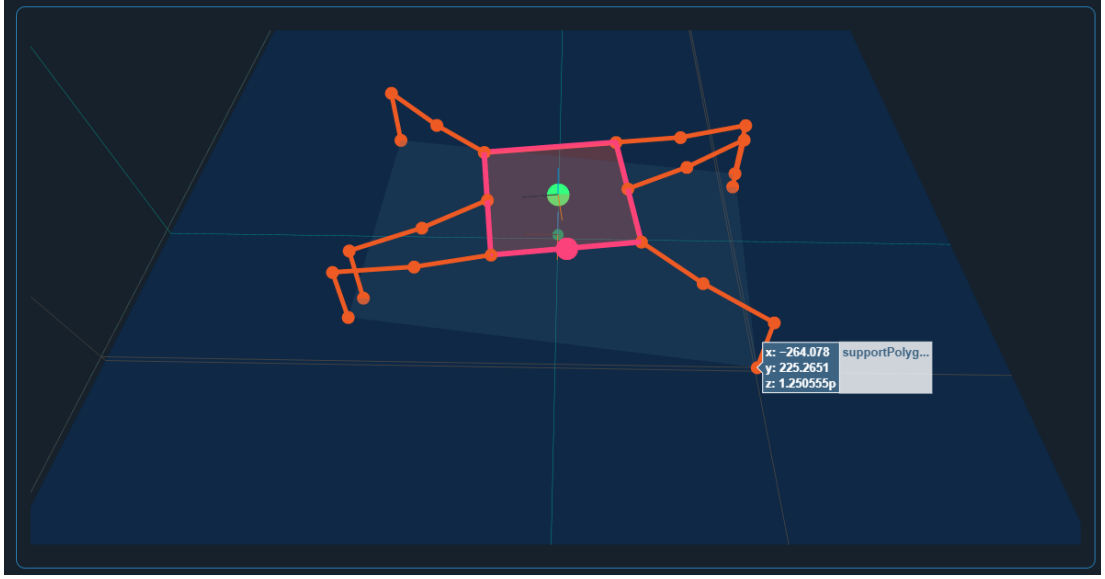
Şekil 2.9: Drosophila melanogaster yürüyüş yapısı [19]

Klasik tripod yürüyüşünde lokomotor döngüsü başına iki güç vuruşu vardır. Her güç vuruşu sırasında üç bacak yüzeydedir (duruş aşaması, siyah daireler), diğer üç bacak ise yüzeyin dışındadır (salınım aşaması, gri daireler). Gri ok uçları hareket yönünü gösterir. [20] [19]



Şekil 2.10: Robotun senkron çalışan tripod yürüme mekanizması



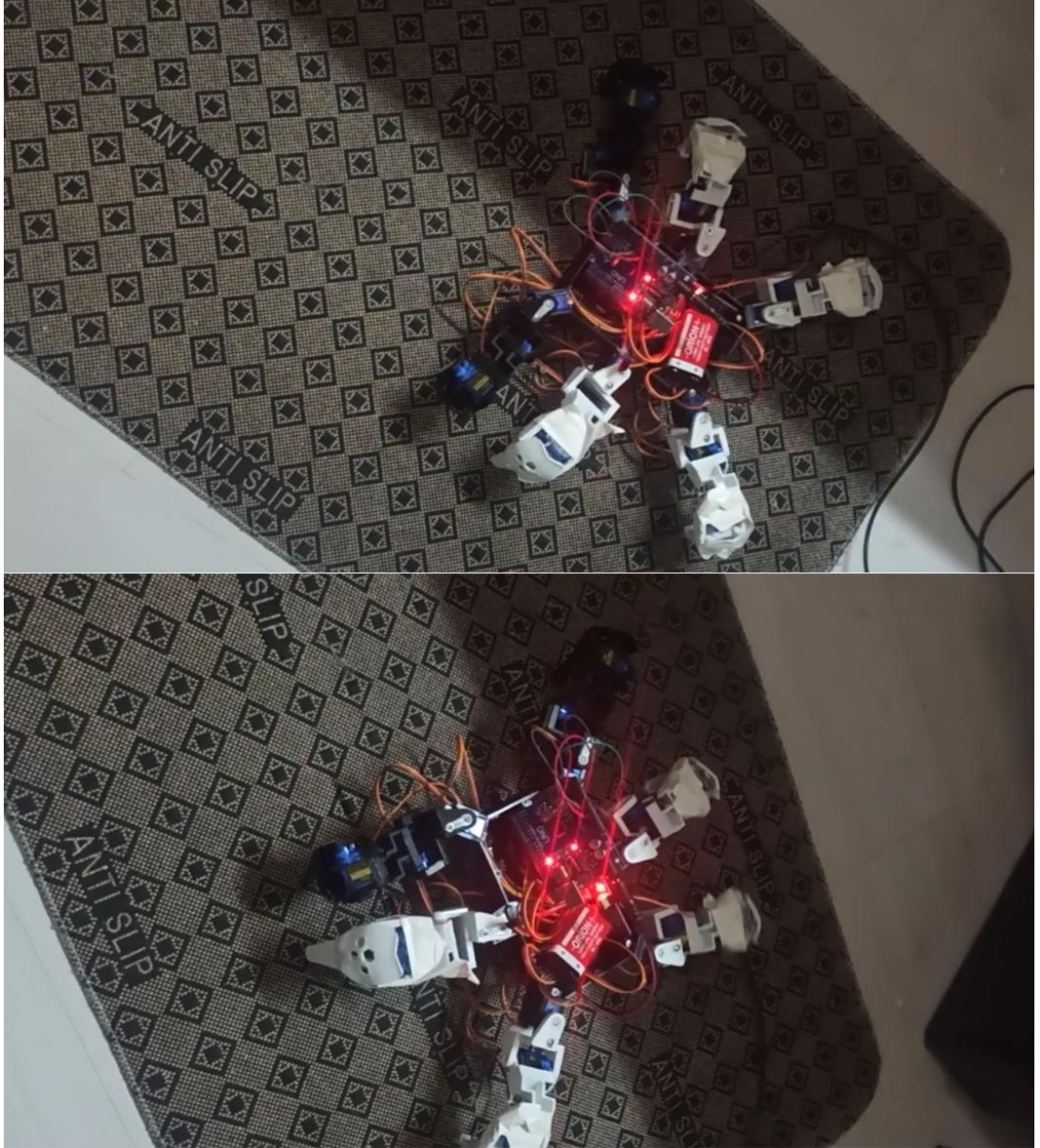


Şekil 2.11: Robotun senkron çalışan tripod yürüme simülasyonu [21]

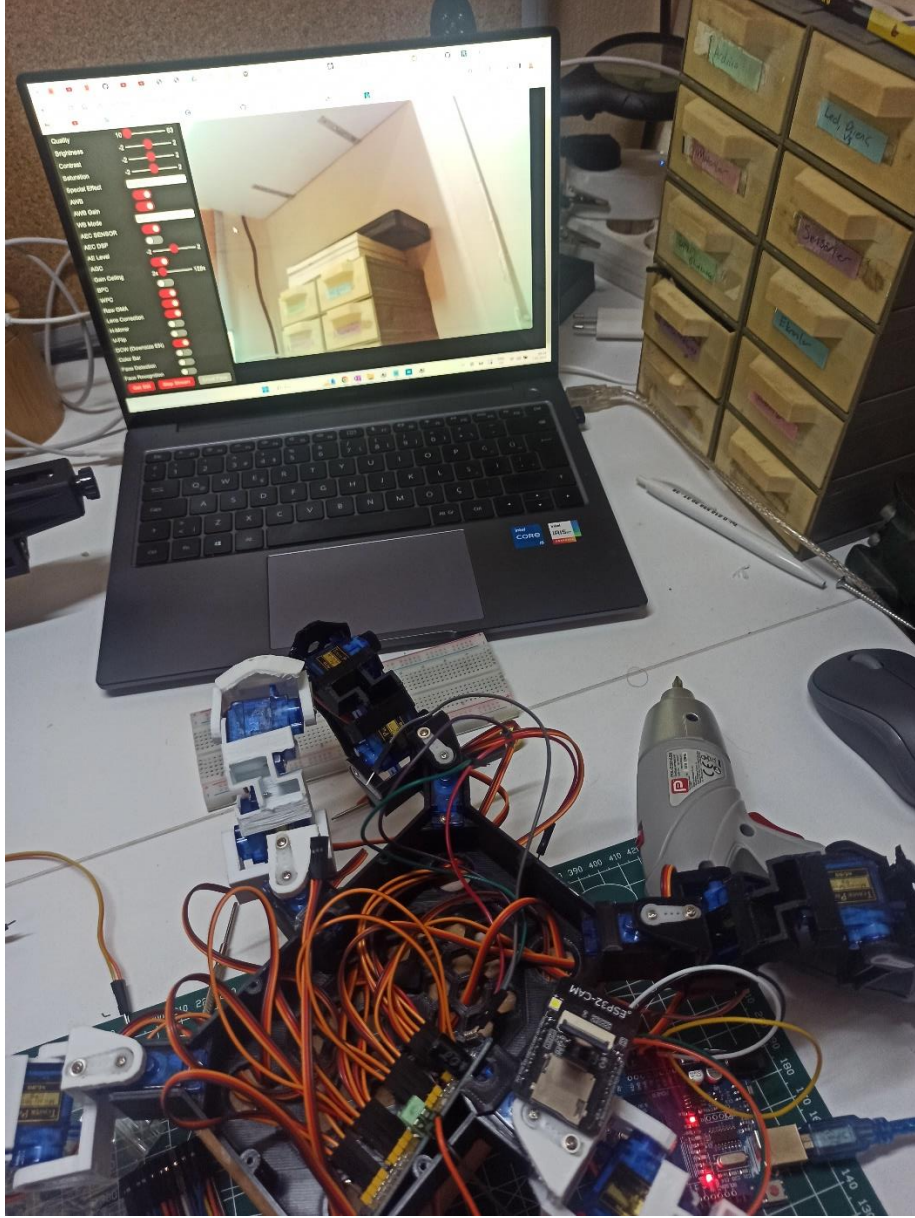
## Bölüm 3

### Sonuç

Robotumuz şu anda yürüme ve dönme hareketlerini başarıyla gerçekleştirebilmekte ve wifi ağına bağlanarak görüntü aktarımı yapabilmektedir. Ancak, PLA baskı malzemesinin yüzey kaymasına ve hareket kısıtlamalarına neden olduğu tespit edilmiştir. Bu sorunu çözmek için, yerde temas eden bölgelerin silikon benzeri kaymaz bir malzeme ile değiştirilmesi önerilmektedir. (Bkz. Şekil 3.1) Prototip sonraki versiyonu hafif ama dayanıklı bir malzemedan yapılmalıdır. Kullanılan malzeme testler sırasında kırılma bükülme gibi sorunlar çıkarmıştır. Her kullanıcının rahatça kullanması için bir telefon uygulaması oluşturulmalıdır. Ayrıca, harici bir SIM kart modülü ile internete bağlanabilme özelliği eklenmesi, robotun wifi ya da bluetooth bağlantısının kopması durumunda ekstra önlemler alabilmesi açısından önemlidir. Yürüme mekanizması üzerinde daha fazla çalışma yapılmalı ve prototipin stabilitesi artırılmalıdır. Fonksiyonlara ek olarak, uzanma ve tırmanma yetenekleri gibi özelliklerin eklenmesi düşünülmelidir, çünkü robotun anatomisi bu tür eklemeleri destekleyebilecek şekildedir. Ayrıca, arama kurtarma çalışmaları sırasında güvenlik ve çevresel koşulları anlamak adına oksijen, gaz, sıcaklık ölçer gibi sensörlerin eklenmesi gerekmektedir. Ürün, tam bir kullanılabilir ürüne dönüştükten sonra mesafe ölçer, engel algılayıcı, termal kamera gibi özelliklerin eklenerek yapay zekâ ile entegre hale getirilmesi düşünülmelidir. Böylece, bir robot sürüsü kendi başlarına enkaz içerisindeki yaralıları tespit edip hızlıca ilgili ekiplere bilgi verebilir. Bu proje kapsamında yapılan çalışmalar ile oluşturulmuş Prototip Şekil 3.1 ve Şekil 3.2'te sunulmuştur.



Şekil 3.1: Prototip 1



Şekil 3.2: Prototip 1

# Kaynaklar

- [1] «Wikipedia,» 18 January 2024. [Çevrimiçi]. Available: <https://tr.wikipedia.org/wiki/Robot>. [Erişildi: 29 January 2024].
- [2] J. Chen, Y. Liu, J. Zhao, H. Zhang ve H. Jin, «Biomimetic Design and Optimal Swing of a Hexapod Robot Leg,» %1 içinde *Journal of Bionic Engineering 11*, China, Harbin Institute of Technology, 2014, pp. 26-35.
- [3] P. Ramdya, R. Thandiackal, R. Cherney, T. Asselborn ve R. Benton, «Climbing favours the tripod gait over alternative,» *Nature Communications*, 2017.
- [4] R. Vidoni ve A. Gasparetto, «Efficient force distribution and leg posture for a bio-inspired spider robot,» %1 içinde *Robotics and Autonomous Systems*, Italy, Elsevier, 2011, pp. 142-150.
- [5] K. Kedikli, B. Yıldız, K. Toka ve A. Genç, «Performing of Reptile Robot Prototype with Radio Frequency Remote Control for the Exploration After the Earthquake Ruins,» *GAZİ JOURNAL OF ENGINEERING SCIENCES*, pp. 41-48, 2022.
- [6] E. Kuzu, «KABLOSUZ KONTROL EDİLEBİLEN MOBİL ARAŞTIRMA VE MÜDEHALE ROBOTU,» 2016.
- [7] T. Hirzel, «Arduino Docs,» [Çevrimiçi]. Available: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>.

- [8] «Arduino Docs,» [Çevrimiçi]. Available: <https://docs.arduino.cc/resources/pinouts/A000067-full-pinout.pdf>.
- [9] «Robotistan Robot Market,» [Çevrimiçi]. Available: [https://www.robotistan.com/orjinal-arduino-mega-2560-r3-yeni-versiyon-1?language=tr&h=5d6ece6d&\\_sgm\\_campaign=fcs\\_c12e1a8d78000&\\_sgm\\_source=470%7Cproduct&\\_sgm\\_action=search&\\_sgm\\_term=arduino+mega&\\_sgm\\_pinned=false](https://www.robotistan.com/orjinal-arduino-mega-2560-r3-yeni-versiyon-1?language=tr&h=5d6ece6d&_sgm_campaign=fcs_c12e1a8d78000&_sgm_source=470%7Cproduct&_sgm_action=search&_sgm_term=arduino+mega&_sgm_pinned=false).
- [10] «Maker Robotistan,» [Çevrimiçi]. Available: <https://maker.robotistan.com/rc-servo-motor-nedir/>.
- [11] H. T. J. C. J. S. L. C. Zhang W P, «BioDKM: Bio-inspired domain knowledge modeling method for humanoid delivery robots' planning.,» %1 içinde *Expert Systems with Applications*, 2012, pp. 663-672.
- [12] A. A. S. D. M. C. Li Y, «Abigaille II: Toward,» *Robotica*, pp. 79-89, 2012.
- [13] Q. R. D. Lewinger W A, «Neurobiologically-based control system for an adaptively walking hexapod.,» *Industrial Robot: An International Journal* , pp. 258-263, 2011.
- [14] L. R. S. D. K. F. Klaassen B, «Biomimetic walking robot SCORPION: Control and modeling.,» *Robotics and Autonomous Systems*, pp. 69-79.
- [15] C. P. S. Gullan P J, %1 içinde *The Insects: An Outline of Entomology*, USA, Wiley-Blackwell, 2005.

- [16] A. B. T. D. M. S. H. & B. Wosnitza, «A. Inter-leg coordination in the control of walking speed in *Drosophila*,» *J. Exp. Biol.*, pp. 480-491, 2013.
- [17] D. M. Wilson, «Insect walking,» *Annu. Rev. Entomol.*, pp. 103-122, 1966.
- [18] R. J. & T. M. S. Full, «Mechanics of six-legged runners,» *J. Exp. Biol.*, pp. 129-146, 1990.
- [19] R. J. B. R. & T. L. H. Full, «Leg design in hexapedal runners.,» *J. Exp. Biol.*, pp. 369-390, 1991.
- [20] L. W. T. & B. Reinhardt, «R. Dynamics and kinematics of ant locomotion: do wood ants climb on level surfaces?,» *J. Exp. Biol.*, pp. 2426-2435, 2009.
- [21] [Çevrimiçi]. Available: <https://hexapod.netlify.app/walking-gaits>.

# Ekler



```
#include <SoftwareSerial.h>

SoftwareSerial Bluetooth(12, 9);

#include <Servo.h>

#include "esp_camera.h"

#include <WiFi.h>

#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

const char* ssid = "MEHMETSIRRI";

const char* password = "alphan944194";

void startCameraServer();

int Received = 0;

int DELAY = 5;

int MODE = 0;

bool Impair_start = false;

int FM1 = 0;
```

int FM2 = 0;

int FM3 = 0;

int FM4 = 0;

int FM5 = 0;

int FM6 = 0;

int FM7 = 0;

int FM8 = 0;

int home\_Leg1\_Mot1 = 75;

int home\_Leg1\_Mot2 = 90;

int home\_Leg1\_Mot3 = 100;

int home\_Leg2\_Mot1 = 70;

int home\_Leg2\_Mot2 = 90;

int home\_Leg2\_Mot3 = 115;

int home\_Leg3\_Mot1 = 70;

int home\_Leg3\_Mot2 = 90;

int home\_Leg3\_Mot3 = 100;

```
int home_Leg4_Mot1 = 105;
```

```
int home_Leg4_Mot2 = 90;
```

```
int home_Leg4_Mot3 = 110;
```

```
int home_Leg5_Mot1 = 75;
```

```
int home_Leg5_Mot2 = 90;
```

```
int home_Leg5_Mot3 = 90;
```

```
int home_Leg6_Mot1 = 110;
```

```
int home_Leg6_Mot2 = 100;
```

```
int home_Leg6_Mot3 = 115;
```

```
Servo Leg1_Mot1, Leg1_Mot2, Leg1_Mot3;
```

```
Servo Leg2_Mot1, Leg2_Mot2, Leg2_Mot3;
```

```
Servo Leg3_Mot1, Leg3_Mot2, Leg3_Mot3;
```

```
Servo Leg4_Mot1, Leg4_Mot2, Leg4_Mot3;
```

```
Servo Leg5_Mot1, Leg5_Mot2, Leg5_Mot3;
```

```
Servo Leg6_Mot1, Leg6_Mot2, Leg6_Mot3;
```

```
void setup() {  
  
    Serial.begin(115200);  
  
    Serial.setDebugOutput(true);  
  
    Serial.println();  
  
    camera_config_t config;  
  
    config.ledc_channel = LEDC_CHANNEL_0;  
  
    config.ledc_timer = LEDC_TIMER_0;  
  
    config.pin_d0 = Y2_GPIO_NUM;  
  
    config.pin_d1 = Y3_GPIO_NUM;  
  
    config.pin_d2 = Y4_GPIO_NUM;  
  
    config.pin_d3 = Y5_GPIO_NUM;  
  
    config.pin_d4 = Y6_GPIO_NUM;  
  
    config.pin_d5 = Y7_GPIO_NUM;  
  
    config.pin_d6 = Y8_GPIO_NUM;  
  
    config.pin_d7 = Y9_GPIO_NUM;  
  
    config.pin_xclk = XCLK_GPIO_NUM;
```

```
config.pin_pclk = PCLK_GPIO_NUM;

config.pin_vsync = VSYNC_GPIO_NUM;

config.pin_href = HREF_GPIO_NUM;

config.pin_sscb_sda = SIOD_GPIO_NUM;

config.pin_sscb_scl = SIOC_GPIO_NUM;

config.pin_pwdn = PWDN_GPIO_NUM;

config.pin_reset = RESET_GPIO_NUM;

config.xclk_freq_hz = 20000000;

config.pixel_format = PIXFORMAT_JPEG;
```

```
if(psramFound()){

    config.frame_size = FRAMESIZE_UXGA;

    config.jpeg_quality = 10;

    config.fb_count = 2;

} else {

    config.frame_size = FRAMESIZE_SVGA;

    config.jpeg_quality = 12;

    config.fb_count = 1;

}
```

```

#if defined(CAMERA_MODEL_ESP_EYE)

    pinMode(13, INPUT_PULLUP);

    pinMode(14, INPUT_PULLUP);

#endif

esp_err_t err = esp_camera_init(&config);

if (err != ESP_OK) {

    Serial.printf("Camera init failed with error 0x%x", err);

    return;

}

sensor_t * s = esp_camera_sensor_get();

if (s->id.PID == OV3660_PID) {

    s->set_vflip(s, 1);

    s->set_brightness(s, 1);

    s->set_saturation(s, -2);

}

s->set_framesize(s, FRAMESIZE_QVGA);

```

```

#if          defined(CAMERA_MODEL_M5STACK_WIDE)           ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)

s->set_vflip(s, 1);

s->set_hmirror(s, 1);

#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

  delay(500);

  Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");

Serial.print(WiFi.localIP());

Serial.println(" to connect");

Serial.begin(9600);

Bluetooth.begin(9600);

```

Leg1\_Mot1.attach(1);

Leg1\_Mot2.attach(2);

Leg1\_Mot3.attach(3);

Leg2\_Mot1.attach(4);

Leg2\_Mot2.attach(5);

Leg2\_Mot3.attach(6);

Leg3\_Mot1.attach(7);

Leg3\_Mot2.attach(8);

Leg3\_Mot3.attach(9);

Leg4\_Mot1.attach(10);

Leg4\_Mot2.attach(11);

Leg4\_Mot3.attach(12);

Leg5\_Mot1.attach(13);

Leg5\_Mot2.attach(14);

Leg5\_Mot3.attach(15);



```
Leg6_Mot1.attach(16);

Leg6_Mot2.attach(17);

Leg6_Mot3.attach(18);

set_home_pos();

}

void loop() {

if(Bluetooth.available(>0)

{

Received = Bluetooth.read();

if (Received == 0)

{

MODE = 0;

Impair_start = false;

FM1 = 0;

FM2 = 0;

FM3 = 0;
```

```
FM4 = 0;

FM5 = 0;

FM6 = 0;

FM7 = 0;

FM8 = 0;

}

if (Received == 1)

{

    MODE = 1;

}

if (Received == 2)

{

    MODE = 2;

}

if (Received == 3)

{

    MODE = 3;

}

if (Received == 4)

{

    MODE = 4;
```



```
if(MODE == 1){  
    move_frwd();  
}
```

```
if(MODE == 2){  
    move_bwd();  
}
```

```
if(MODE == 3){  
    move_right();  
}
```

```
if(MODE == 4){  
    move_left();  
}
```

```
if(MODE == 5){  
    rotate_left();  
}
```

```
if(MODE == 6){  
    rotate_right();  
}
```

```
delay(DELAY);  
}
```

```
void set_home_pos()  
{  
    Leg1_Mot1.write(home_Leg1_Mot1);  
    Leg1_Mot2.write(home_Leg1_Mot2);  
    Leg1_Mot3.write(home_Leg1_Mot3);  
}
```

Leg2\_Mot1.write(home\_Leg2\_Mot1);

Leg2\_Mot2.write(home\_Leg2\_Mot2);

Leg2\_Mot3.write(home\_Leg2\_Mot3);

Leg3\_Mot1.write(home\_Leg3\_Mot1);

Leg3\_Mot2.write(home\_Leg3\_Mot2);

Leg3\_Mot3.write(home\_Leg3\_Mot3);

Leg4\_Mot1.write(home\_Leg4\_Mot1);

Leg4\_Mot2.write(home\_Leg4\_Mot2);

Leg4\_Mot3.write(home\_Leg4\_Mot3);

Leg5\_Mot1.write(home\_Leg5\_Mot1);

Leg5\_Mot2.write(home\_Leg5\_Mot2);

Leg5\_Mot3.write(home\_Leg5\_Mot3);

Leg6\_Mot1.write(home\_Leg6\_Mot1);

Leg6\_Mot2.write(home\_Leg6\_Mot2);

Leg6\_Mot3.write(home\_Leg6\_Mot3);

}

```
void move_frwd(){  
  
    if (FM1 <=10){  
  
        Leg1_Mot2.write(home_Leg1_Mot2 - FM1);  
  
        Leg1_Mot3.write(home_Leg1_Mot3 - FM1);  
  
        Leg3_Mot2.write(home_Leg3_Mot2 - FM1);  
  
        Leg3_Mot3.write(home_Leg3_Mot3 - FM1);  
  
        Leg5_Mot2.write(home_Leg5_Mot2 - FM1);  
  
        Leg5_Mot3.write(home_Leg5_Mot3 - FM1);  
  
        FM1++;  
  
    }  
  
}
```

```
if (FM2 <= 30)  
  
{  
  
    Leg1_Mot1.write(home_Leg1_Mot1 - FM2);  
  
    Leg3_Mot1.write(home_Leg3_Mot1 - FM2);  
  
    Leg5_Mot1.write(home_Leg5_Mot1 + FM2);  
  
}
```

```
    FM2++;  
  
}  
  
if (FM2 > 20 && FM3 <=10){  
  
    Leg1_Mot2.write(home_Leg1_Mot2 + FM3);  
  
    Leg1_Mot3.write(home_Leg1_Mot3 + FM3);  
  
    Leg3_Mot2.write(home_Leg3_Mot2 + FM3);  
  
    Leg3_Mot3.write(home_Leg3_Mot3 + FM3);  
  
    Leg5_Mot2.write(home_Leg5_Mot2 + FM3);  
  
    Leg5_Mot3.write(home_Leg5_Mot3 + FM3);  
  
    FM3++;  
  
}
```

```
if (FM2 >= 30)  
  
{  
  
    Leg1_Mot1.write(home_Leg1_Mot1 + FM4);  
  
    Leg3_Mot1.write(home_Leg3_Mot1 + FM4);  
  
    Leg5_Mot1.write(home_Leg5_Mot1 - FM4);  
  
    FM4++;  
  
}
```



```
    Impair_start = true;

}

if(FM4 >= 30) {

    FM1 = 0;

    FM2 = 0;

    FM3 = 0;

    FM4 = 0;

}

if (Impair_start){

    if (FM5 <=10){

        Leg2_Mot2.write(home_Leg2_Mot2 - FM5);

        Leg2_Mot3.write(home_Leg2_Mot3 - FM5);

        Leg4_Mot2.write(home_Leg4_Mot2 - FM5);

        Leg4_Mot3.write(home_Leg4_Mot3 - FM5);

        Leg6_Mot2.write(home_Leg6_Mot2 - FM5);

        Leg6_Mot3.write(home_Leg6_Mot3 - FM5);

        FM5++;

    }

}
```

```
}
```

```
if (FM6 <= 30)
```

```
{
```

```
Leg2_Mot1.write(home_Leg2_Mot1 - FM6);
```

```
Leg4_Mot1.write(home_Leg4_Mot1 + FM6);
```

```
Leg6_Mot1.write(home_Leg6_Mot1 + FM6);
```

```
FM6++;
```

```
}
```

```
if (FM6 > 20 && FM7 <=10){
```

```
Leg2_Mot2.write(home_Leg2_Mot2 + FM7);
```

```
Leg2_Mot3.write(home_Leg2_Mot3 + FM7);
```

```
Leg4_Mot2.write(home_Leg4_Mot2 + FM7);
```

```
Leg4_Mot3.write(home_Leg4_Mot3 + FM7);
```

```
Leg6_Mot2.write(home_Leg6_Mot2 + FM7);
```

```
Leg6_Mot3.write(home_Leg6_Mot3 + FM7);
```

```
FM7++;
```

```
}
```

```
if (FM6 >= 30)
{
    Leg2_Mot1.write(home_Leg2_Mot1 + FM8);
    Leg4_Mot1.write(home_Leg4_Mot1 - FM8);
    Leg6_Mot1.write(home_Leg6_Mot1 - FM8);
    FM8++;
}
if(FM8 >= 30) {
    Impair_start = false;
    FM5 = 0;
    FM6 = 0;
    FM7 = 0;
    FM8 = 0;
}
}
}
```

```
void move_bwd(){
```

```
if (FM1 <=10){  
  
    Leg1_Mot2.write(home_Leg1_Mot2 - FM1);  
  
    Leg1_Mot3.write(home_Leg1_Mot3 - FM1);  
  
    Leg3_Mot2.write(home_Leg3_Mot2 - FM1);  
  
    Leg3_Mot3.write(home_Leg3_Mot3 - FM1);  
  
    Leg5_Mot2.write(home_Leg5_Mot2 - FM1);  
  
    Leg5_Mot3.write(home_Leg5_Mot3 - FM1);  
  
    FM1++;  
  
}
```

```
if (FM2 <= 30)  
  
{  
  
    Leg1_Mot1.write(home_Leg1_Mot1 + FM2);  
  
    Leg3_Mot1.write(home_Leg3_Mot1 + FM2);  
  
    Leg5_Mot1.write(home_Leg5_Mot1 - FM2);  
  
    FM2++;  
  
}
```

```
if (FM2 > 20 && FM3 <=10){  
  
    Leg1_Mot2.write(home_Leg1_Mot2 + FM3);  
  
    Leg1_Mot3.write(home_Leg1_Mot3 + FM3);  
  
    Leg3_Mot2.write(home_Leg3_Mot2 + FM3);  
  
    Leg3_Mot3.write(home_Leg3_Mot3 + FM3);  
  
    Leg5_Mot2.write(home_Leg5_Mot2 + FM3);  
  
    Leg5_Mot3.write(home_Leg5_Mot3 + FM3);  
  
    FM3++;  
  
}
```

```
if (FM2 >= 30)  
  
{  
  
    Leg1_Mot1.write(home_Leg1_Mot1 - FM4);  
  
    Leg3_Mot1.write(home_Leg3_Mot1 - FM4);  
  
    Leg5_Mot1.write(home_Leg5_Mot1 + FM4);  
  
    FM4++;  
  
    Impair_start = true;  
  
}
```

```
if(FM4 >= 30) {  
  
    FM1 = 0;
```

```
FM2 = 0;

FM3 = 0;

FM4 = 0;

}
```

```
if (Impair_start){
```

```
    if (FM5 <=10){
```

```
        Leg2_Mot2.write(home_Leg2_Mot2 - FM5);
```

```
        Leg2_Mot3.write(home_Leg2_Mot3 - FM5);
```

```
        Leg4_Mot2.write(home_Leg4_Mot2 - FM5);
```

```
        Leg4_Mot3.write(home_Leg4_Mot3 - FM5);
```

```
        Leg6_Mot2.write(home_Leg6_Mot2 - FM5);
```

```
        Leg6_Mot3.write(home_Leg6_Mot3 - FM5);
```

```
        FM5++;
```

```
    }
```

```
    if (FM6 <= 30)
```

```
    {
```

```
Leg2_Mot1.write(home_Leg2_Mot1 + FM6);  
  
Leg4_Mot1.write(home_Leg4_Mot1 - FM6);  
  
Leg6_Mot1.write(home_Leg6_Mot1 - FM6);  
  
FM6++;  
  
}
```

```
if (FM6 > 20 && FM7 <=10){  
  
    Leg2_Mot2.write(home_Leg2_Mot2 + FM7);  
  
    Leg2_Mot3.write(home_Leg2_Mot3 + FM7);  
  
    Leg4_Mot2.write(home_Leg4_Mot2 + FM7);  
  
    Leg4_Mot3.write(home_Leg4_Mot3 + FM7);  
  
    Leg6_Mot2.write(home_Leg6_Mot2 + FM7);  
  
    Leg6_Mot3.write(home_Leg6_Mot3 + FM7);  
  
    FM7++;  
  
}
```

```
if (FM6 >= 30)  
  
{  
  
    Leg2_Mot1.write(home_Leg2_Mot1 - FM8);
```

```
Leg4_Mot1.write(home_Leg4_Mot1 + FM8);

Leg6_Mot1.write(home_Leg6_Mot1 + FM8);

FM8++;

}

if(FM8 >= 30) {

    FM5 = 0;

    FM6 = 0;

    FM7 = 0;

    FM8 = 0;

}

}

}
```

```
void rotate_left(){
```

```
if (FM1 <=10){

    Leg1_Mot2.write(home_Leg1_Mot2 - FM1);

    Leg1_Mot3.write(home_Leg1_Mot3 - FM1);

    Leg3_Mot2.write(home_Leg3_Mot2 - FM1);
```



```
Leg3_Mot3.write(home_Leg3_Mot3 - FM1);  
  
Leg5_Mot2.write(home_Leg5_Mot2 - FM1);  
  
Leg5_Mot3.write(home_Leg5_Mot3 - FM1);  
  
FM1++;  
  
}
```

```
if (FM2 <= 30)
```

```
{  
  
Leg1_Mot1.write(home_Leg1_Mot1 - FM2);  
  
Leg3_Mot1.write(home_Leg3_Mot1 - FM2);  
  
Leg5_Mot1.write(home_Leg5_Mot1 - FM2);  
  
FM2++;  
  
}
```

```
if (FM2 > 20 && FM3 <=10){
```

```
Leg1_Mot2.write(home_Leg1_Mot2 + FM3);  
  
Leg1_Mot3.write(home_Leg1_Mot3 + FM3);  
  
Leg3_Mot2.write(home_Leg3_Mot2 + FM3);  
  
Leg3_Mot3.write(home_Leg3_Mot3 + FM3);
```

```
Leg5_Mot2.write(home_Leg5_Mot2 + FM3);  
  
Leg5_Mot3.write(home_Leg5_Mot3 + FM3);  
  
FM3++;  
  
}
```

```
if (FM2 >= 30)
```

```
{  
  
Leg1_Mot1.write(home_Leg1_Mot1 + FM4);  
  
Leg3_Mot1.write(home_Leg3_Mot1 + FM4);  
  
Leg5_Mot1.write(home_Leg5_Mot1 + FM4);  
  
FM4++;  
  
Impair_start = true;  
  
}
```

```
if(FM4 >= 30) {
```

```
FM1 = 0;  
  
FM2 = 0;  
  
FM3 = 0;  
  
FM4 = 0;  
  
}
```

```
if (Impair_start){

    if (FM5 <=10){

        Leg2_Mot2.write(home_Leg2_Mot2 - FM5);

        Leg2_Mot3.write(home_Leg2_Mot3 - FM5);

        Leg4_Mot2.write(home_Leg4_Mot2 - FM5);

        Leg4_Mot3.write(home_Leg4_Mot3 - FM5);

        Leg6_Mot2.write(home_Leg6_Mot2 - FM5);

        Leg6_Mot3.write(home_Leg6_Mot3 - FM5);

        FM5++;

    }

    if (FM6 <= 30)

    {

        Leg2_Mot1.write(home_Leg2_Mot1 - FM6);

        Leg4_Mot1.write(home_Leg4_Mot1 - FM6);

        Leg6_Mot1.write(home_Leg6_Mot1 - FM6);

        FM6++;

    }

}
```

```
if (FM6 > 20 && FM7 <=10){  
  
    Leg2_Mot2.write(home_Leg2_Mot2 + FM7);  
  
    Leg2_Mot3.write(home_Leg2_Mot3 + FM7);  
  
    Leg4_Mot2.write(home_Leg4_Mot2 + FM7);  
  
    Leg4_Mot3.write(home_Leg4_Mot3 + FM7);  
  
    Leg6_Mot2.write(home_Leg6_Mot2 + FM7);  
  
    Leg6_Mot3.write(home_Leg6_Mot3 + FM7);  
  
    FM7++;  
  
}
```

```
if (FM6 >= 30)  
  
{  
  
    Leg2_Mot1.write(home_Leg2_Mot1 + FM8);  
  
    Leg4_Mot1.write(home_Leg4_Mot1 + FM8);  
  
    Leg6_Mot1.write(home_Leg6_Mot1 + FM8);  
  
    FM8++;  
  
}
```

```
if(FM8 >= 30) {
```

```

    Impair_start = false;

    FM5 = 0;

    FM6 = 0;

    FM7 = 0;

    FM8 = 0;

    }

    }

}

void rotate_right(){

    if (FM1 <=10){

        Leg1_Mot2.write(home_Leg1_Mot2 - FM1);

        Leg1_Mot3.write(home_Leg1_Mot3 - FM1);

        Leg3_Mot2.write(home_Leg3_Mot2 - FM1);

        Leg3_Mot3.write(home_Leg3_Mot3 - FM1);

        Leg5_Mot2.write(home_Leg5_Mot2 - FM1);

        Leg5_Mot3.write(home_Leg5_Mot3 - FM1);

        FM1++;

    }

```

```
if (FM2 <= 30)
{
    Leg1_Mot1.write(home_Leg1_Mot1 + FM2);
    Leg3_Mot1.write(home_Leg3_Mot1 + FM2);
    Leg5_Mot1.write(home_Leg5_Mot1 + FM2);
    FM2++;
}
```

```
if (FM2 > 20 && FM3 <=10){
    Leg1_Mot2.write(home_Leg1_Mot2 + FM3);
    Leg1_Mot3.write(home_Leg1_Mot3 + FM3);
    Leg3_Mot2.write(home_Leg3_Mot2 + FM3);
    Leg3_Mot3.write(home_Leg3_Mot3 + FM3);
    Leg5_Mot2.write(home_Leg5_Mot2 + FM3);
    Leg5_Mot3.write(home_Leg5_Mot3 + FM3);
    FM3++;
}
```

```
if (FM2 >= 30)
{
    Leg1_Mot1.write(home_Leg1_Mot1 - FM4);
    Leg3_Mot1.write(home_Leg3_Mot1 - FM4);
    Leg5_Mot1.write(home_Leg5_Mot1 - FM4);
    FM4++;
    Impair_start = true;
}
if(FM4 >= 30) {
    FM1 = 0;
    FM2 = 0;
    FM3 = 0;
    FM4 = 0;
}
```

```
if (Impair_start){
```

```
    if (FM5 <=10){
```

```
Leg2_Mot2.write(home_Leg2_Mot2 - FM5);  
  
Leg2_Mot3.write(home_Leg2_Mot3 - FM5);  
  
Leg4_Mot2.write(home_Leg4_Mot2 - FM5);  
  
Leg4_Mot3.write(home_Leg4_Mot3 - FM5);  
  
Leg6_Mot2.write(home_Leg6_Mot2 - FM5);  
  
Leg6_Mot3.write(home_Leg6_Mot3 - FM5);  
  
FM5++;  
  
}
```

```
if (FM6 <= 30)  
{  
  
Leg2_Mot1.write(home_Leg2_Mot1 + FM6);  
  
Leg4_Mot1.write(home_Leg4_Mot1 + FM6);  
  
Leg6_Mot1.write(home_Leg6_Mot1 + FM6);  
  
FM6++;  
  
}
```

```
if (FM6 > 20 && FM7 <=10){  
  
Leg2_Mot2.write(home_Leg2_Mot2 + FM7);  
  
Leg2_Mot3.write(home_Leg2_Mot3 + FM7);
```



```
Leg4_Mot2.write(home_Leg4_Mot2 + FM7);  
  
Leg4_Mot3.write(home_Leg4_Mot3 + FM7);  
  
Leg6_Mot2.write(home_Leg6_Mot2 + FM7);  
  
Leg6_Mot3.write(home_Leg6_Mot3 + FM7);  
  
FM7++;  
  
}
```

```
if (FM6 >= 30)
```

```
{  
  
Leg2_Mot1.write(home_Leg2_Mot1 - FM8);  
  
Leg4_Mot1.write(home_Leg4_Mot1 - FM8);  
  
Leg6_Mot1.write(home_Leg6_Mot1 - FM8);  
  
FM8++;  
  
}
```

```
if(FM8 >= 30) {
```

```
FM5 = 0;  
  
FM6 = 0;  
  
FM7 = 0;  
  
FM8 = 0;  
  
}
```

```
    }  
}  
  
void move_right(){  
  
    if (FM1 <= 20){  
  
        Leg1_Mot2.write(home_Leg1_Mot2 + FM1);  
  
        Leg1_Mot3.write(home_Leg1_Mot3 + FM1);  
  
  
        Leg2_Mot2.write(home_Leg2_Mot2 + FM1);  
  
        Leg2_Mot3.write(home_Leg2_Mot3 - FM1*2);  
  
  
        Leg3_Mot2.write(home_Leg3_Mot2 + FM1);  
  
        Leg3_Mot3.write(home_Leg3_Mot3 + FM1);  
  
  
        Leg4_Mot2.write(home_Leg4_Mot2 - FM1);  
  
        Leg4_Mot3.write(home_Leg4_Mot3 + FM1*2);  
  
  
        Leg5_Mot2.write(home_Leg5_Mot2 + FM1);  
  
        Leg5_Mot3.write(home_Leg5_Mot3 - FM1);  
  
    }  
}
```

```

Leg6_Mot2.write(home_Leg6_Mot2 - FM1);

Leg6_Mot3.write(home_Leg6_Mot3 + FM1*2);

FM1++;

}

if(FM1 >= 20 && FM2 <= 20){

Leg2_Mot2.write(home_Leg2_Mot2 + FM1 + FM2);

Leg2_Mot3.write(home_Leg2_Mot3 - FM1*2 + FM2*2);

Leg4_Mot2.write(home_Leg4_Mot2 - FM1 + FM2);

Leg4_Mot3.write(home_Leg4_Mot3 + FM1*2 - FM2*2);

Leg6_Mot2.write(home_Leg6_Mot2 - FM1 + FM2);

Leg6_Mot3.write(home_Leg6_Mot3 + FM1*2 - FM2*2);

FM2++;

}

if(FM2 >= 20 && FM3 <= 20){

Leg1_Mot2.write(home_Leg1_Mot2 + FM1 - FM3);

Leg1_Mot3.write(home_Leg1_Mot3 + FM1 - FM1);

```

```
Leg3_Mot2.write(home_Leg3_Mot2 + FM1 - FM3);
```

```
Leg3_Mot3.write(home_Leg3_Mot3 + FM1 - FM3);
```

```
Leg5_Mot2.write(home_Leg5_Mot2 + FM1 - FM3);
```

```
Leg5_Mot3.write(home_Leg5_Mot3 - FM1 + FM3);
```

```
FM3++;
```

```
}
```

```
if(FM3 >= 20){
```

```
    FM1 = 0;
```

```
    FM2 = 0;
```

```
    FM3 = 0;
```

```
}
```

```
}
```

```
void move_left(){
```

```
    if (FM1 <= 20){
```

```
        Leg4_Mot2.write(home_Leg4_Mot2 + FM1);
```

```
Leg4_Mot3.write(home_Leg4_Mot3 + FM1);
```

```
Leg5_Mot2.write(home_Leg5_Mot2 + FM1);
```

```
Leg5_Mot3.write(home_Leg5_Mot3 - FM1*2);
```

```
Leg6_Mot2.write(home_Leg6_Mot2 + FM1);
```

```
Leg6_Mot3.write(home_Leg6_Mot3 + FM1);
```

```
Leg3_Mot2.write(home_Leg4_Mot3 - FM1);
```

```
Leg3_Mot3.write(home_Leg4_Mot3 + FM1*2);
```

```
Leg2_Mot2.write(home_Leg2_Mot2 + FM1);
```

```
Leg2_Mot3.write(home_Leg2_Mot3 - FM1);
```

```
Leg1_Mot2.write(home_Leg1_Mot2 - FM1);
```

```
Leg1_Mot3.write(home_Leg1_Mot3 + FM1*2);
```

```
FM1++;
```

```
}
```

```
if(FM1 >= 20 && FM2 <= 20){
```

```
Leg5_Mot2.write(home_Leg5_Mot2 + FM1 + FM2);
```

```
Leg5_Mot3.write(home_Leg5_Mot3 - FM1*2 + FM2*2);
```

```
Leg3_Mot2.write(home_Leg3_Mot2 - FM1 + FM2);
```

```
Leg3_Mot3.write(home_Leg3_Mot3 + FM1*2 - FM2*2);
```

```
Leg1_Mot2.write(home_Leg1_Mot2 - FM1 + FM2);
```

```
Leg1_Mot3.write(home_Leg1_Mot3 + FM1*2 - FM2*2);
```

```
FM2++;
```

```
}
```

```
if(FM2 >= 20 && FM3 <= 20){
```

```
Leg6_Mot2.write(home_Leg6_Mot2 + FM1 - FM3);
```

```
Leg6_Mot3.write(home_Leg6_Mot3 + FM1 - FM1);
```

```
Leg4_Mot2.write(home_Leg4_Mot2 + FM1 - FM3);
```

```
Leg4_Mot3.write(home_Leg4_Mot3 + FM1 - FM3);
```

```
Leg2_Mot2.write(home_Leg2_Mot2 + FM1 - FM3);
```

```
Leg2_Mot3.write(home_Leg2_Mot3 - FM1 + FM3);
```

```
    FM3++;  
  
}  
  
if(FM3 >= 20){  
  
    FM1 = 0;  
  
    FM2 = 0;  
  
    FM3 = 0;  
  
}  
  
}
```