T.C.

KARABÜK ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ

Quadruped Robot

Hazırlayan

Engin Aydın 2015010225067

Tez Danışmanı

Doç.Dr. İbrahim ÇAYIROĞLU

06-2019

KABUL VE ONAY

Engin AYDIN tarafından hazırlanan "Quadruped Robot başlıklı bu tezin Lisans Bitirme Tezi olarak uygun olduğunu onaylarım. 17/06/2019

Tez Danışmanı

.....Yrd.Doç.Dr. İbrahim ÇAYIROĞLU.....
Mekatronik Müh. Böl.Bşk.

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Mekatronik Mühendisliği Anabilim Dalında Lisans Bitirme Tezi olarak kabul edilmiştir. 17/06/2019

Tez Jürisi

Başkan: Yrd.Doç.Dr. İbrahim ÇAYIROĞLU Mekatronik Müh. Böl.Bşk. .....

Üye : Prof.Dr. Gökhan Gökoğlu .....

Üye : Dr.Öğr.Ü. Can Bülent FİDAN .....

KBÜ Mühendislik Fakültesi, Mekatronik Mühendisliği Mezuniyet Komisyonu ve Bölüm Başkanlığı bu tezi Lisans Bitirme Tezi olarak onamıştır. 17/06/2019

Doç.Dr. İbrahim ÇAYIROĞLU
Mekatronik Müh. Bölüm Bşk.

ÖNSÖZ

Öncelikle proje ve tez konusunda yardımlarını esirgemeyen danışmanım sayın Doç. Dr. İbrahim ÇAYIROĞLU başta olmak üzerine lisans eğitimim boyunca üzerimde emeği olan tüm öğretim üyelerine ve eğitim hayatım boyunca yanımda olan, desteklerini her daim hissettiğim aileme sonsuz teşekkürlerimi sunarım.

Engin AYDIN

İçindekiler

Şekiller Listesi

Simge Listesi

V: Volt

mA: Miliamper

KB: Kilobayt

MHz:

mm: Milimetre

g: Gram

kgf·cm:

dBm: Desibel (dB), sinyal gücünün miktarı. (Desibel x miliwatt)

MBps: Saniye başına megabit

KBps: Saniye başına kilobit

ÖZET

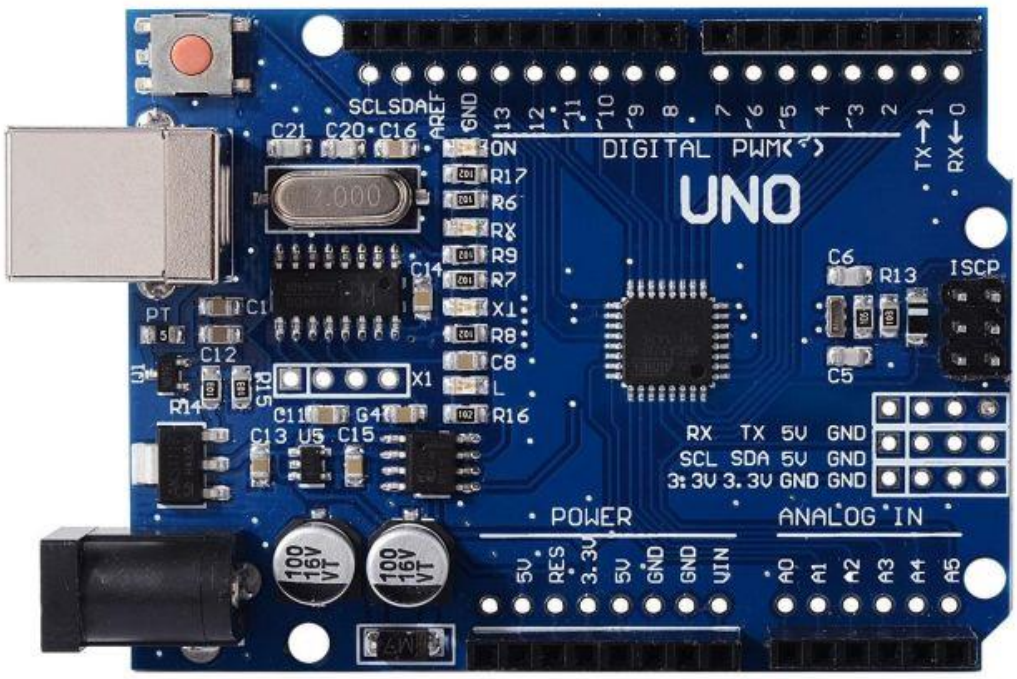
Bu projede günümüzde hızla önem kazanan insansız hava araçlarının geliştirilmesinin ve üretilmesinin eksiklerini gidermek adına ortaya çıkan insansız kara araçlarının tasarımında farklı bir yaklaşım sağlanması amaçlanmıştır.

Kara araçlarına esneklik kazandırmak adına palet veya tekerlek gibi elemanlarla hareket imkânı kısıtlı bir şekilde hareket ettirmek yerine eklemlerle desteklenmiş uzuvlarla daha geniş bir hareket uzayında hareket etmesi sağlanacaktır. Tasarlanacak robot hareketi sağlayan ana gövde ve istenilen göreve göre değiştirilecek bir başlıktan oluşmaktadır.

1. KULLANILAN MALZEMELER

Robotun hareketli mekanizmasını oluşturan gövdenin alt kısmının değiştirilmesi gövdenin üst kısmının değiştirilmesine oranla daha kısıtlı olacağından bu bölgedeki araçlar standartlaştırılmıştır. Hareketli kısımlar için servo motorlar yeterli olurken gövde kısmında yapılacak özelleştirmeler ile birden fazla sensör veyaa seri haberleşme elamanı bulunabilir.

1.1. Arduino Mikrodenetleyici



[1] 1.1 Arduino Uno R3

Bu projede Arduino Uno R3 klonu kullanılmıştır. R3 klonu üzerlerinde kullanılan elektronik devre elemanları ve tüm işlevselliği orijinal kart ile aynıdır. Ancak orjinal üründe DIP kılıfında bulunan yonga, bu üründe ise SMD kılıfında yer alır.

Projede kapsamında Arduino ile HC-06 bluetooth modülü kullanılarak servo motorların kontrolü sağlanmıştır. Servo motorların kontrolü için 4,5,8,9,10,11,12,13 numaralı pinler ve bluetooth modülünün kontrolü için 6 ve 7 numaralı pinler kullanılmıştır.

Kartı beslemek için ise 5V ve 2,1 A çıkış değerlerine sahip bir powerbank kullanılmıştır.

1.1.1. Arduino UNO R3'ün Teknik Özellikleri

Mikrodenetleyici: ATmega328

Çalışma Gerilimi: 5V

Giriş Gerilimi (önerilen): 7-12V

Giriş Gerilimi (limit): 6-20V

Dijital G/Ç Pinleri: 14 (6 tanesi PWM çıkışı)

Analog Giriş Pinleri: 6

Her G/Ç için Akım: 40 mA

3.3V Çıkış için Akım: 50 mA

Flash Hafıza: 32 KB (ATmega328) 0.5 KB kadarı bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Saat Hızı: 16 MHz

Uzunluk: 68.6 mm

Genişlik: 53.4 mm

Ağırlık: 25 g

1.1.2. Güç

Arduino UNO gücünü USB üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi batarya da olabilir. Adaptör kart üzerindeki 2.1mm merkez-pozitif güç soketinden veya GND (-) ve Vin (+) pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak USB'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

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

UNO 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.

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.

1.1.3. Hafıza

Atmega328P mikrokontrolcüsü 32 KB'lık flash belleğe sahiptir (0.5 KB kadarı bootloader tarafından kullanılmaktadır). 2 KB SRAM ve 1 KB EEPROM'u bulunmaktadır.

1.1.4. Giriş ve Çıkış

UNO üzerindeki 14 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. 6 tane analog giriş pini de bulunmaktadır. Bu analog giriş pinleride aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 20 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:

Seri Haberleşme, 0 (RX) ve 1 (TX): TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bu pinler doğrudan kart üzerinde bulunan Atmega16u2 USB-seri dönüştürücüsüne bağlıdır. Yani bilgisayardan karta program yüklenirken veya bilgisayar-UNO arasında karşılıklı haberleşme yapılırken de bu pinler kullanılır. O yüzden karta program yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.

Harici Kesme, 2 (interrupt 0) ve 3 (interrupt 1): Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir.

PWM, 3,5,6,9,10 ve 11: 8-bit çözünürülükte PWM çıkış pinleri olarak kullanılabilir.

SPI, 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK): Bu pinler SPI haberleşmesi için kullanılır.

LED, 13: UNO üzerinde 13. pine bağlı olan, kart üzerinde "L" harfi ile belirtilmiş dahili bir LED bulunmaktadır. Pin HIGH yapıldığında LED yanacak, LOW yapıldığında LED sönecektir.

Analog, A0-A5: UNO, 6 adet 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'tur. AREF pini veanalogReference() foksiyonu kullanılarak alt limit yükseltilip, üst limit düşürülebilir.

I2C, A4 veya SDA pini ve A5 veya SCL pini: Bu pinler I2C haberleşmesi için kullanılır.

AREF: Analog girişler için ölçüm referansı pini.

Reset: Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan reset butonu ile de yapılabilir.

1.1.5. Haberleşme

Arduino UNO'nun bilgisayarla, başka bir Arduino veya mikrodenetleyici ile haberleşmesi için bir kaç farklı seçenek vardır. Atmega328, 0 (RX) ve 1 (TX) pinleri üzerinden UART TTL (5V) seri haberleşme imkanı sunar. Kart üzerinde bulunan Atmega16u2 USB-seri dönüştürücü de bilgisayarda sanal bir seri port (COM portu) açarak Atmega328 ile bilgisayar arasında bir köprü kurar. Arduino IDE içerisinde yer alan seri monitör ile Arduino ile bilgisayar arasında metin temelli bilgilerin gönderilip alınmasını sağlar. USB-seri dönüştürücü ile bilgisayar arasında USB üzerinden haberleşme olduğu zaman kart üzerinde bulunan RX ve TX LED'leri yanacaktır.

UNO üzerinde donanımsal olarak bir adet seri port bulunmaktadır. Ancak SoftwareSerial kütüphanesi ile bu sayı yazılımsal olarak arttırılabilir.

Atmega328 aynı şekilde I2C ve SPI portlarıda sağlamaktadır. Arduino IDE içerisinde yer alan Wire kütüphanesi I2C kullanımını, SPI kütüphanesi de SPI haberleşmesini sağlamak için kullanılır.

1.1.6. Programlama

Arduino UNO kartı Arduino bilgisayar programı (Arduino IDE) ile programlanır. Program içerisinde Tools > Board sekmesi altında Arduino UNO'yu seçip programlamaya başlayabilirsiniz. Ayrıntılı bilgi için referans ve temel fonksiyonlar sayfasını inceleyebilirsiniz. Arduino UNO üzerindeki Atmega328 üzerine bootloader isimli özel bir yazılım yüklü gelir. Bu sayede kartı programlarken ekstra bir programlayıcı kullanmanıza gerek kalmaz. Haberleşme STK500 protokolü ile sağlanır.

Bootloader yazılımı bypass edilerek kart doğrudan mikrodenetleyicinin ICSP header kullanılarak ISP programlayıcı ile programlanabilir (referans).

Bootloader yazılımı gibi Atmega16u2 içerisindeki kaynak yazılım da açık kaynaklıdır. Bu yazılıma da DFU bootloader adı verilir. Atmel's FLIP software (Windows) veya DFU programmer (Mac OS X ve Linux) kullanılarak bu yazılım yeniden yüklenebilir. Veya Atmega328'de olduğu gibi 16u2'de ISP programlayıcı ile programlanabilir. Gerek Atmega328 gerekse 16u2 içerisindeki yazılımlar her zaman en güncel hali ile gönderilir. O yüzden mecbur kalmadıkça bu yazılımları değiştirmenize gerek yoktur.

1.1.7. USB Aşırı Akım Koruması

Arduino UNO üzerinden bulunan resetlenebilir sigorta, bilgisayarınızın USB portunu kısa devrelerden veya aşırı akım tüketimi durumlarından korumaktadır. Kart, USB portu üzerinden 500mA'den fazla akım çektiğinde otomatik olarak USB'den aldığı gücü koruma amacıyla kesmektedir. Fazla akım durumu veya kısa devre ortadan kaldırıldığında sigorta normal konuma döner ve tekrar bağlantı kurulur.

1.2. Servo Motor

[2] 1.2. Servo motor (MG996)

Robotun hareketini sağlamak amacıyla projede servo motorlar kullanılmıştır. Projede kullanılan servo motorlar Tower Pro markasının MG996 modelidir.

1.2.1. Tower Pro MG996 Servo Motorun Teknik Özellikleri

Büyüklük: 40.6 x 19.8 x 36.5mm,

Ağırlık: 50 gr,

Hız @4.8V: 0.15sn/60°,

Zorlanma Torku @4.8V: 9,4 kgf·cm,

Zorlanma Torku @6V: 13 kgf·cm,

Ölü bant genişliği: 5 µs,

Çalışma Akımı: 500mA - 900mA (6V)

Durma Akımı: 2.5 A (6V),

Çalışma Gerilimi: 4.8V ~ 7.2V,

1.2.2. Servo Motorların Çalışma Prensibi

Servo motorların 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 sürücüye ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolcüden veya uzaktan kumandadan sağlanabilmektedir. Servo, her 20 ms içerisinde bir pals değeri okumaktadır. Pals uzunluğu motorun dönüşünü belirlemektedir. Servolar hareket etmeleri için bir komut aldıklarında önce istenilen pozisyona hareket ederler, sonrasında ise o pozisyonda kalırlar. Servolar bulundukları pozisyonu korurken kendilerine dışarıdan bir güç uygulandığında bu güce direnirler. Bulundukları konumu sonsuza kadar koruyamazlar, pozisyonlarını koruyabilmeleri için palsin tekrar edilmesi gerekebilir. 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.

1.3. Bluetooth Modülü

[3] 1.3 Bluetooth modülü (HC-06)

Robotun kontrolünü sağlamak için bluetooth modül kullanılmıştır. Android cihaz ile iletişim halinde olup, gelen veri ile istenilen komutun algılanması ve buna göre istenilen eylemin gerçekleştirilmesi amaçlanmıştır.

HC-06 Bluetooth-Serial Modül Kartı, Bluetooth SSP(Serial Port Standart) kullanımı ve kablosuz seri haberleşme uygulamaları için tasarlanmıştırÇoğu bluetooth modülden farklı olarak master modunu da desteklemektedir.
Bluetooth 2.0'ı destekleyen bu kart, 2.4GHz frekansında haberleşme yapılmasına imkan sağlayıp açık alanda yaklaşık 10 metrelik bir haberleşme mesafesine sahiptir.

1.3.1. Teknik Özellikler

Bluetooth Protokolü: Bluetooth 2.0+EDR(Gelişmiş Veri Hızı)2.4GHz haberleşme frekansı
Hassasiyet: ≤-80 dBm
Çıkış Gücü: ≤+4 dBm
Asenkron Hız: 2.1 MBps/160 KBps
Senkron Hız: 1 MBps/1 MBps
Güvenlik: Kimlik Doğrulama ve Şifreleme
Çalışma Gerilimi: 1.8-3.6V(Önerilen 3.3V)
kım: 50 mA
Boyutları: 43x16x7mm

1.4. Güç Kaynağı

Sensörler için gerekli enerji mikrodenetleyici üzerindeki pinlerden sağlanabiliyorken robot üzerindeki motor sayısını fazla olmasından dolayı kart üzerinden gelen enerji motorlar için yetersiz kalacaktır. Mikro denetleyiciyi ve motorları aynı anda beslemek için iki çıkışlı bir powerbank kullanılmıştır.

1.4.1. Teknik Özellikleri

Kapasite: 4000mAh

Giriş değerleri: 5V 1,5 A

Çıkış değerleri: 5V 2,1 A

2. ÇALIŞMA PRENSİBİ

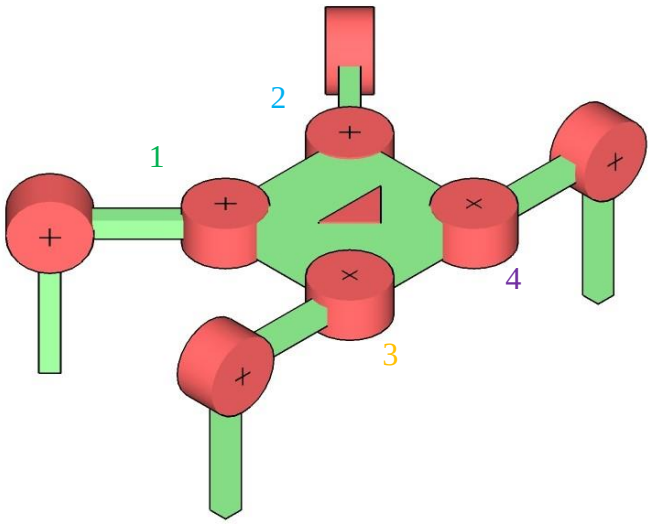
Robotun temel olarak çalışma mantığı önceden belirlenen, hareketlere özgü açı değişimlerinin kumandadan gelen bilgi doğrultusunda devreye sokulmasıdır. Bu başlık altında bu açı değişimlerine örnekler verilecek kod başlığı altında değinilen bilgilerin görselleştirilmesi sağlanacaktır.

Öncelikle hareketin sağlanmasında temel şartlardan biri robotun dengede kalmasıdır. Bu yüzden robotun hareketi sırasında bacaklarından biri havada olacağından diğer üç bacağın oluşturacağı yeni pozisyonun ağırlık merkezi gövdenin ağırlık merkezi üzerinde veya merkezine yakın olmalıdır. Bu denge şartı sağlandıktan sonra robot belirli eylemlere uygun daha önce tanımlanmış açı değerlerini istenilen süre ile devreye sokar. Üst paragrafta bahsedilen hareketlere özgü açı değişimleri şu şekilde gerçekleşir: Robot ilk bacağını kaldırarak ileri hareket ettirir daha sonra bu hareketi diğer üç bacak tekrar eder. Böylece tüm bacaklar bir adım önde olacaktır. Fakat bu durumda gövde pozisyonu geride kalacağından adımlar tamamlandıktan sonra gövdenin kendini ileri itmesi istenir. Bu hareket dizini tamamlandığında robot bir adım ilerlemiş olacaktır.

Bluetooth ile Kontrol başlığı altında belirtilen kumandaya tanımlı değerlerden ileri hareketin bir adımlık döngüsü alt başlık altında sunulmuştur.

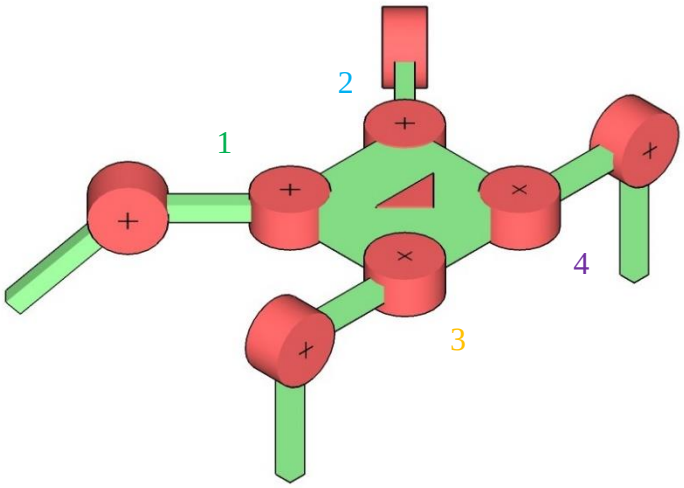
2.1. Bir Adımlık İleri Hareket

Robotun harekete başlamadan önceki açı değerleri.



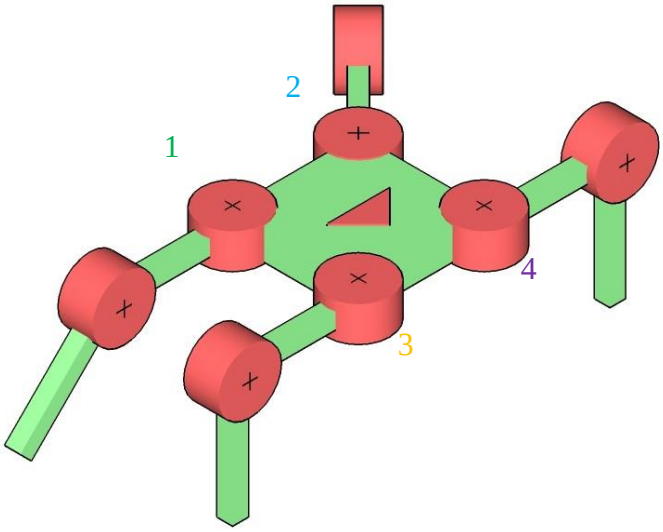
[4] 2.1. Başlangıç açı değerleri

Bacak 1'e ait uç servo ileri atılacağından 45 derece havaya kalkarak bacağın ileri gitmesine olanak tanır.



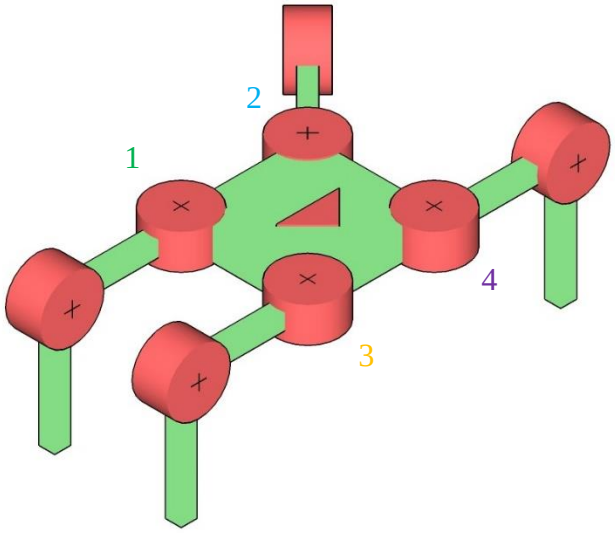
[5] 2.2. Bacak 1'e ait uç motorun hareketinin başlangıcı

Bacak 1'ait dip servo 45 derece dönere Bacak 1'i son pozisyona getirir.



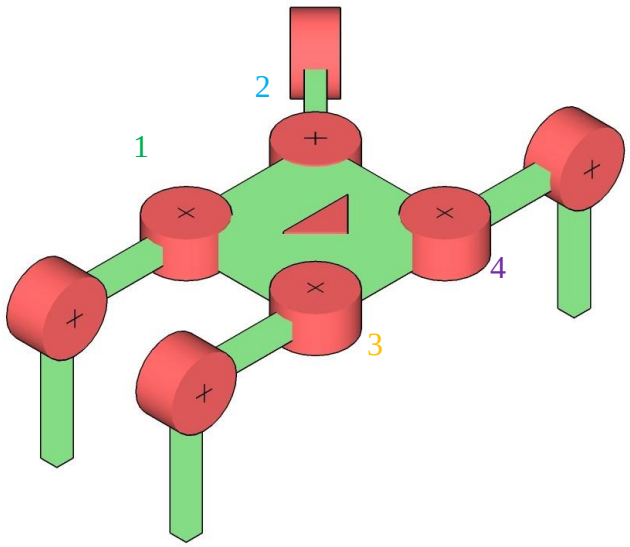
[6] 2.3. Bacak 1'e ait dip motorun hareketi

Bundan sonraki aşamalarda Bacak 1'i ilgilendirecek bir hareket olmadığı için Bacak 1'ait uç servo 45 derece dönerek zemine değer.

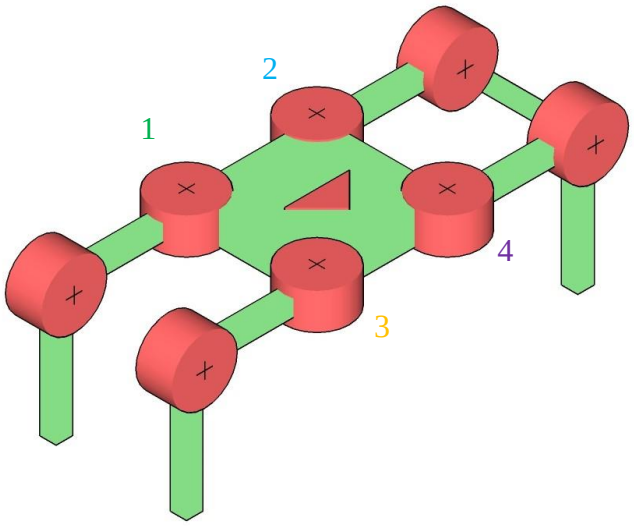


[7] 2.4. Bacak 1'e ait uç motorun hareketinin bitişi

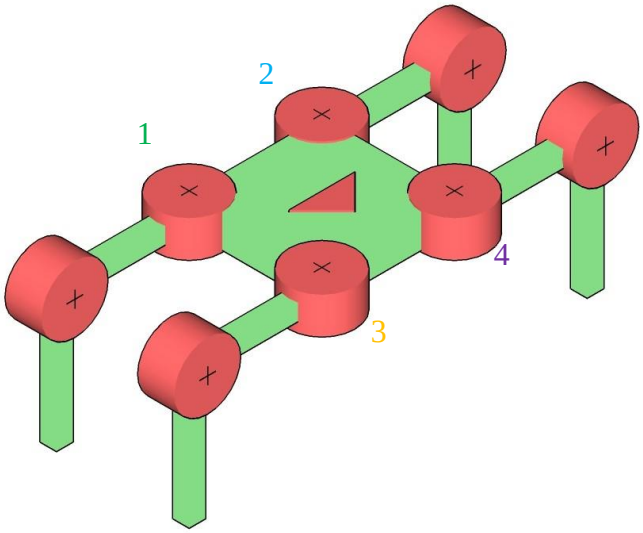
Bundan sonraki aşamalarda diğer üç bacakta aynı hareketi yapacağından diğer bacaklara ilişkin açıklamalar eklenmemiştir.



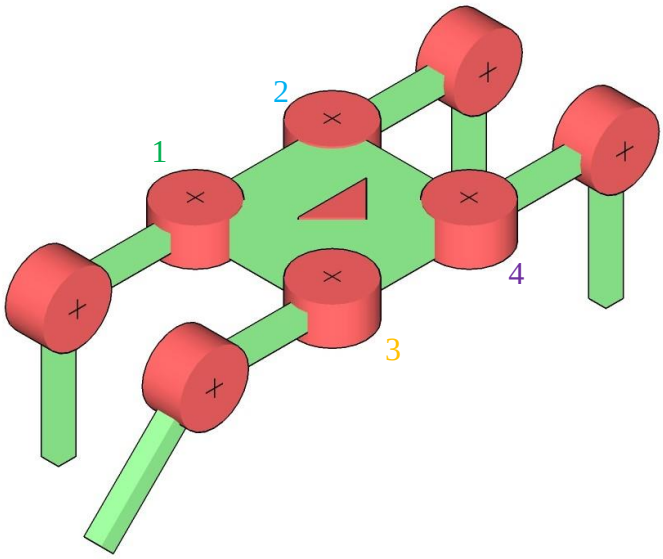
[8] 2.5. Bacak 2'e ait uç motorun hareketinin başlangıcı



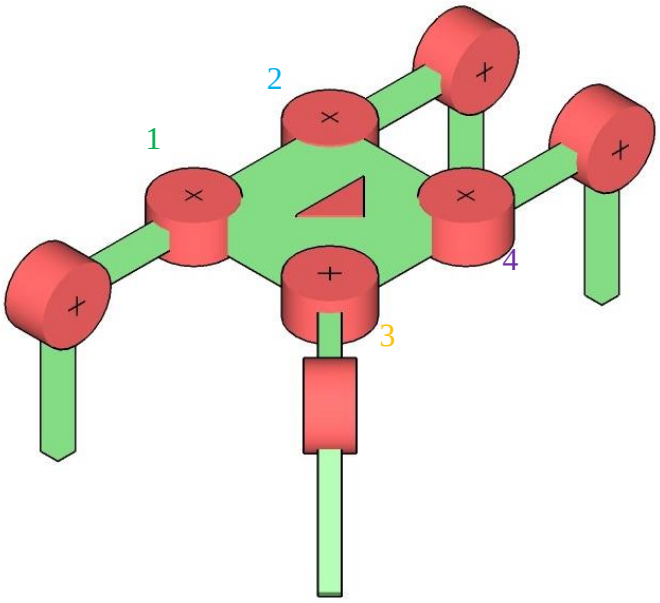
[9] 2.6. Bacak 2'e ait dip motorun hareketi



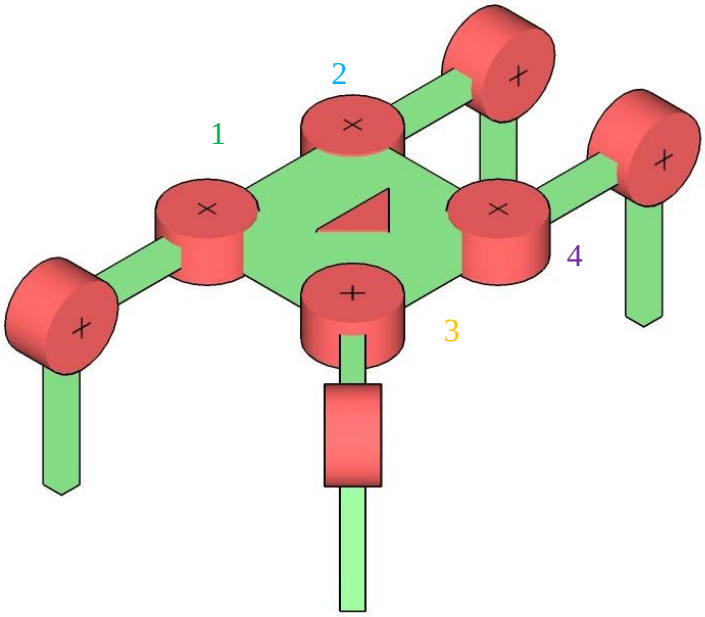
[10] 2.7. Bacak 2'e ait uç motorun hareketinin bitişi



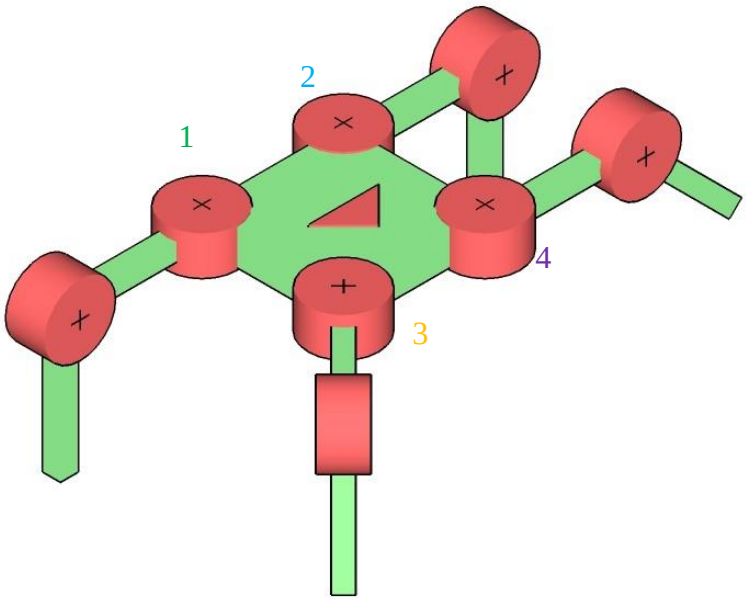
[11] 2.8. Bacak 3'e ait uç motorun hareketinin başlangıcı



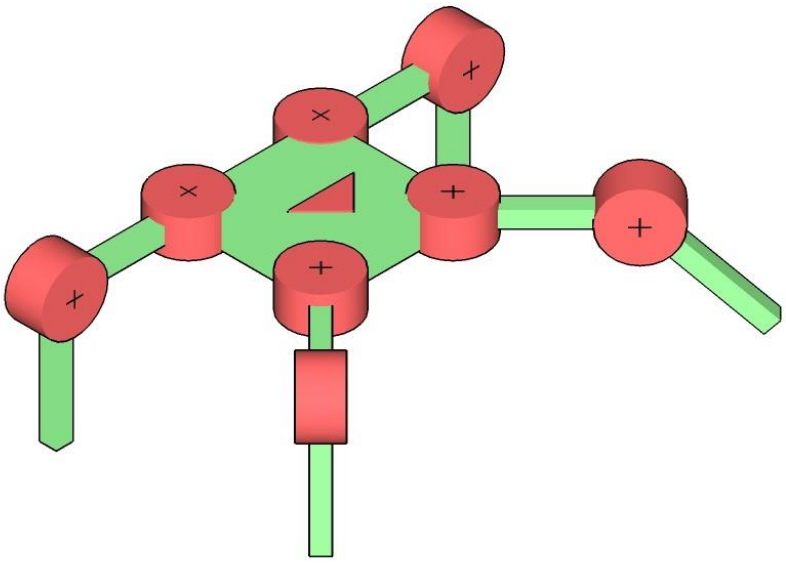
[12] 2.9. Bacak 3’e ait dip motorun hareketi



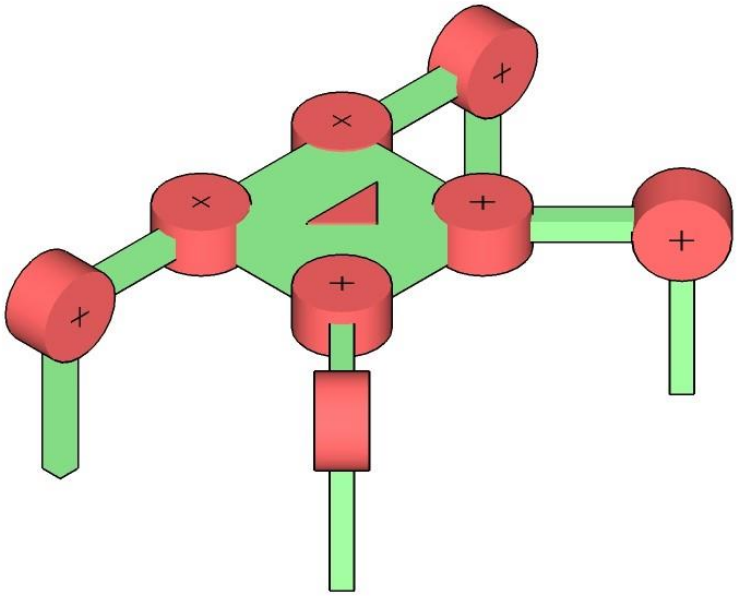
[13] 2.10. Bacak 3’e ait uç motorun hareketinin bitişi



[14] 2.11. Bacak 4'e ait uç motorun hareketinin başlangıcı

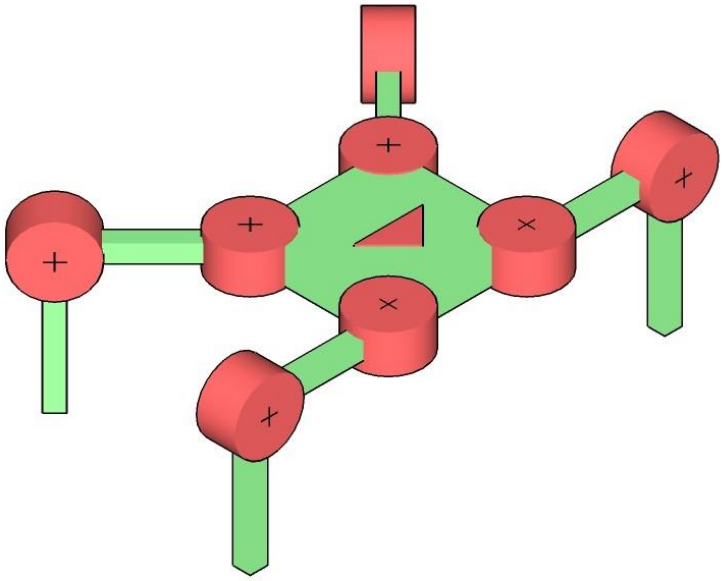


[15] 2.12. Bacak 4'e ait dip motorun hareketi



[16] 2.13. Bacak 4'e ait uç motorun hareketinin bitişi

Tüm bacaklar istenilen konuma geldikten sonra gövde geride kalacağından gövdeyi ileri itmek için dip motorlar yardımıyla son bir hareket yapılır.



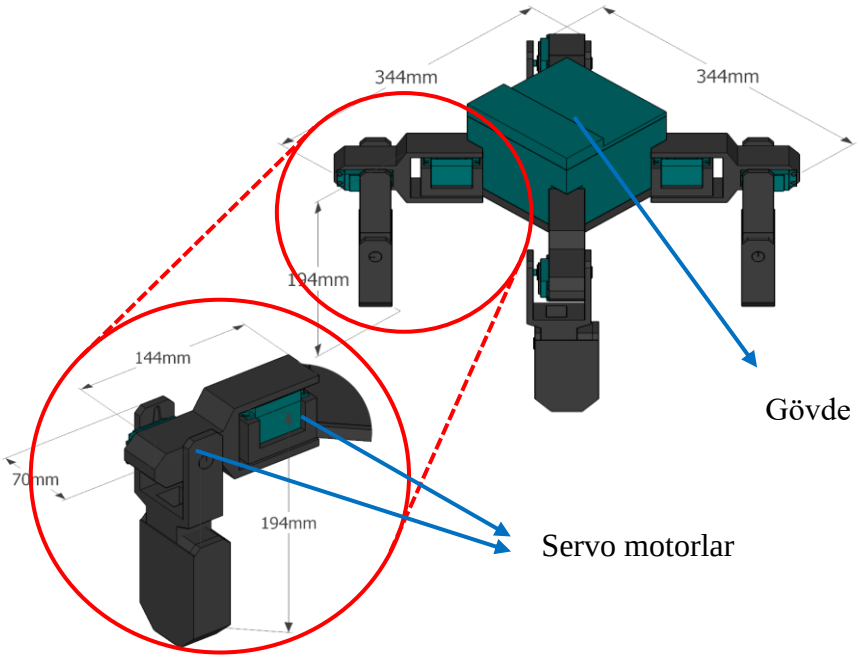
[17] 2.14. Dip motorların kendini itmesi

3. TASARIM

Robotun fiziksel yapısı hareketli alt gövde ve farklı amaçlar için tasarlanmış modüler üst gövdelerden oluşmakta. Alt kısım üst gövde yapılacak değişiklerden bağımsız ve sadece hareket odaklıdır. Üst kısım ise mikrodenetleyici, sensörler ve seri haberleşme modülleri gibi bileşenleri yer aldığı bir yapıdan oluşmaktadır. Bu iki kısmı oluşturan parçalar alt başlıklarda detaylandırılmıştır. Bunlara ek olarak motor gibi bileşenler ile mikrodenetleyici arasındaki bağlantılarda bu başlık altında ele alınmıştır.

3.1. Parçaların Bilgisayar Ortamında Tasarımı

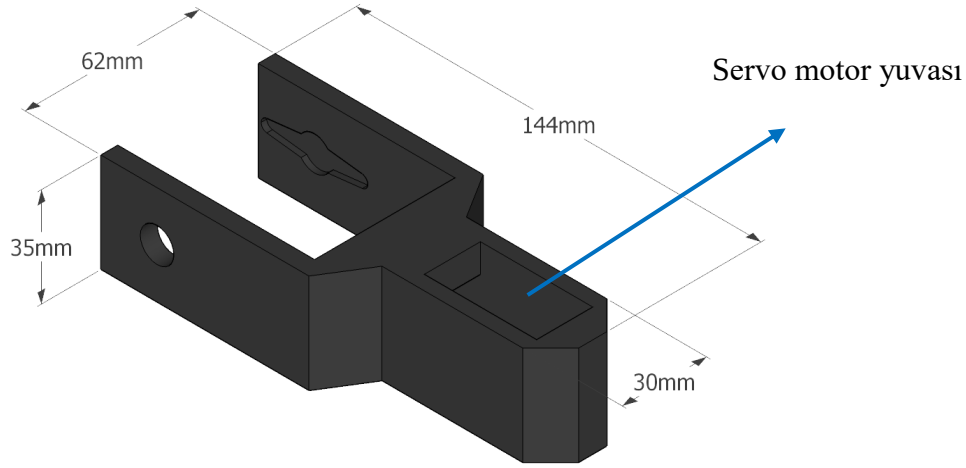
Parçaların bilgisayar ortamında tasarımında SketchUp programı kullanılmıştır. Aşağıdaki şekillerde gövde uzuvlarının ayrıntıları verilmiştir.



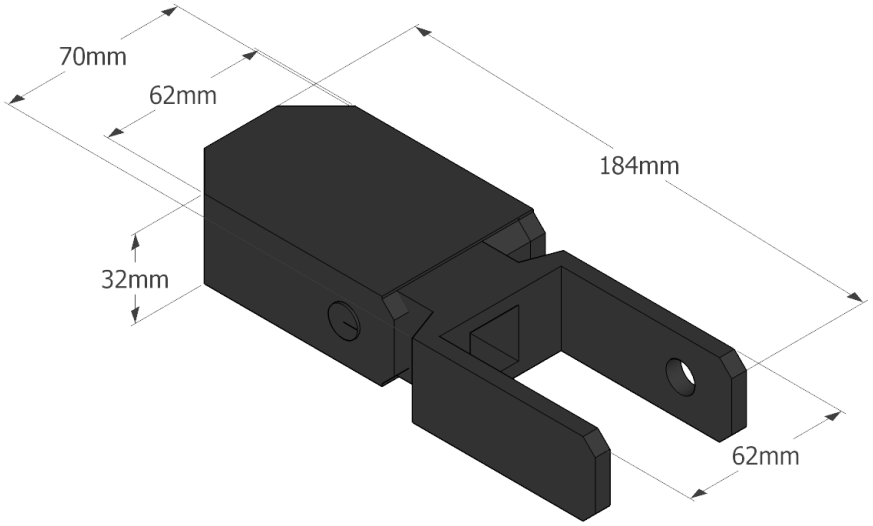
[18] 3.1. Parçaların montajlanmış hali

3.1.1. Bacak Boyutları

Bacaklar iki parçadan oluşur ilk parçanın başlangıcında gövdedeki servo motorun yerleşeceği yuva ve uç kısmında bir sonraki parçayı kontrol etmek için yerleştirilecek servonun yuvası yer alır.



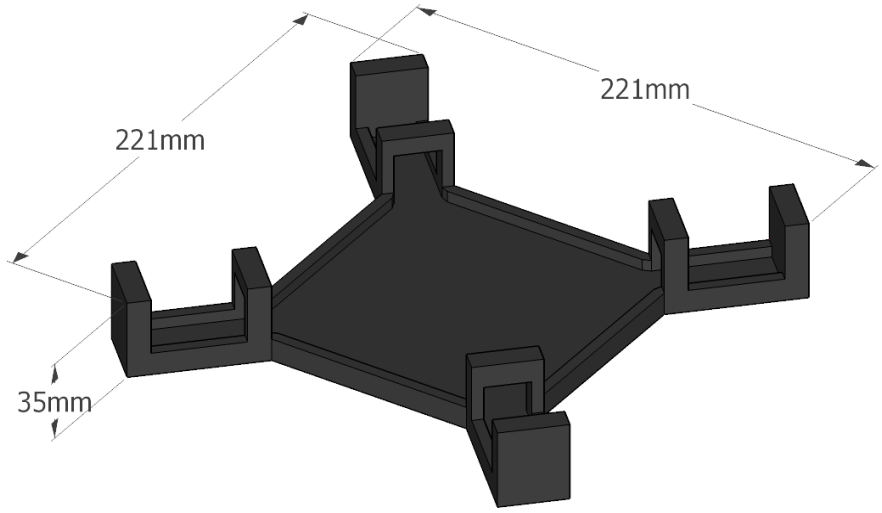
[19] 3.2. Bacak boyutları (1)



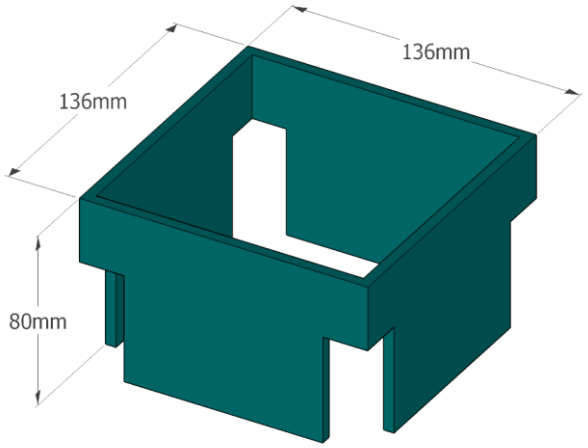
[20] 3.3. Bacak boyutları (2)

3.1.2. Gövde Boyutları

Gövde taban ve bu tabana yerleştirilen bir kasadan oluşmakta. Gövde tabanının köşelerine bacakların kontrolü için servo motorlar yerleştirilmiştir. Tabanın üzerine kasa yerleştirilmesinin amacı robot kontrolü için kullanılan mikrodenetleyici, bluetooth gibi elamanları çevre etkilerinden korunması ve oluşacak kablo ağını bir araya getirilmesidir.



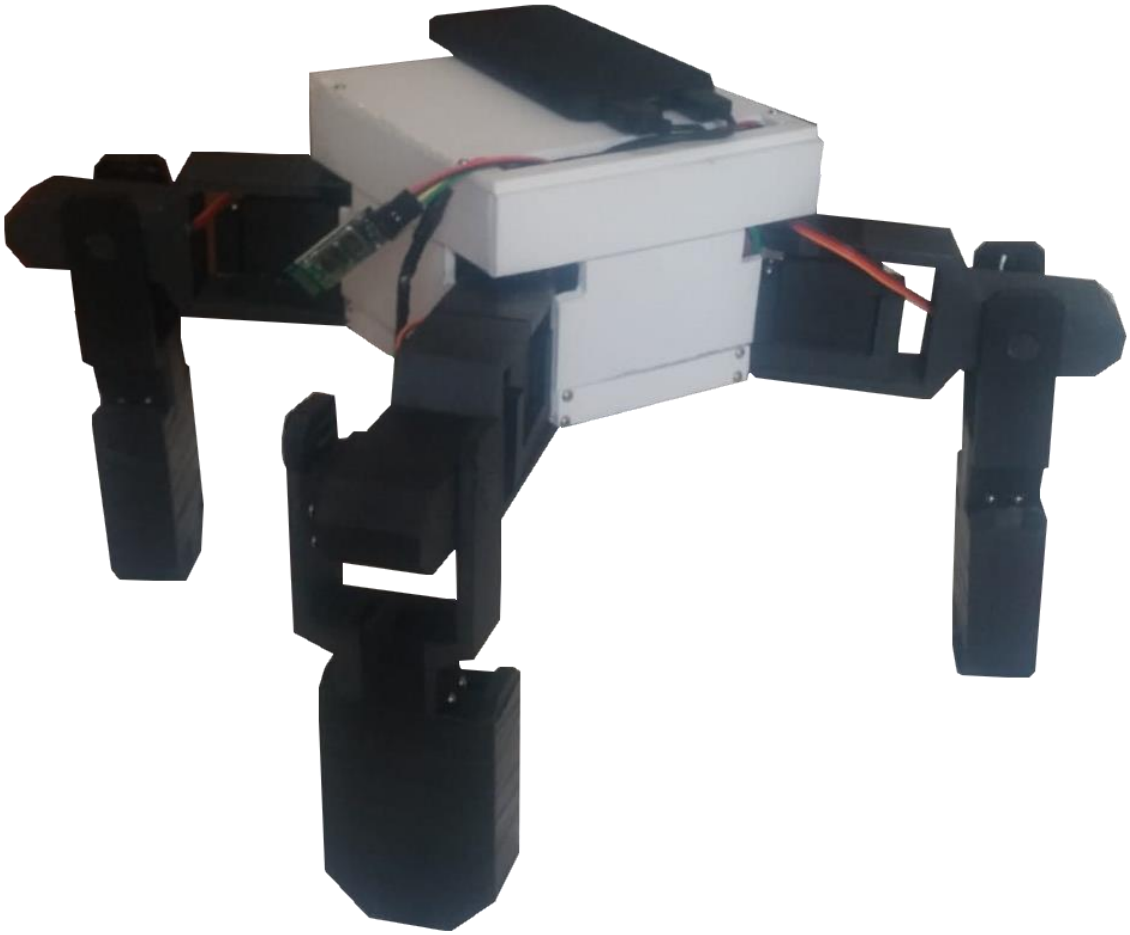
[21] 3.4. Gövde tabanı



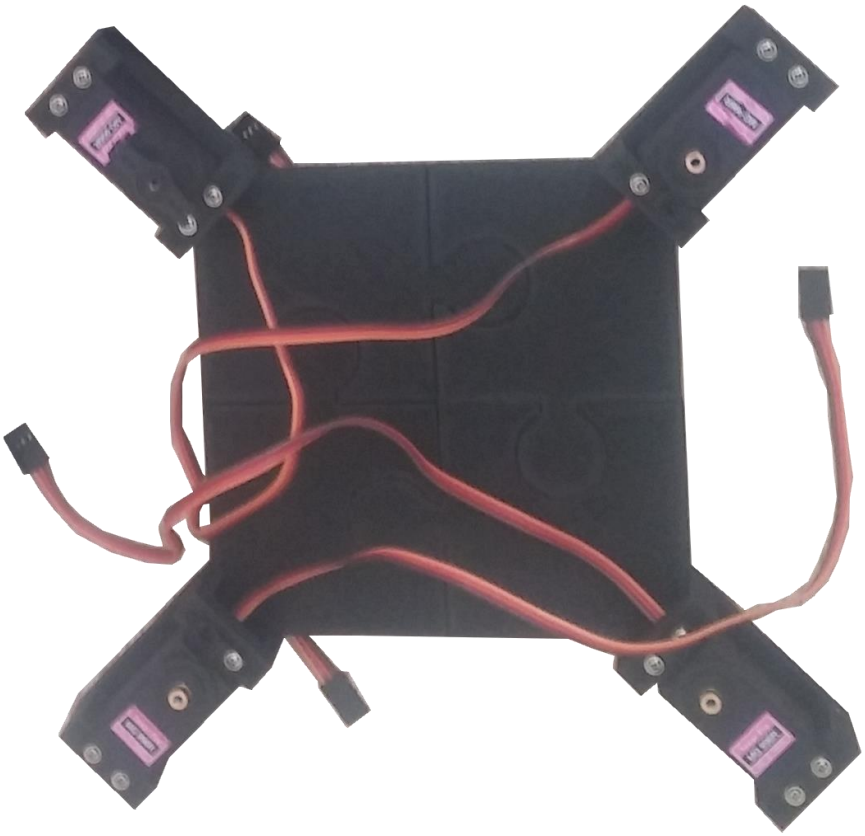
[22] 3.5. Gövde kasası

3.2. 3D Yazıcı ile Üretilen Parçalar

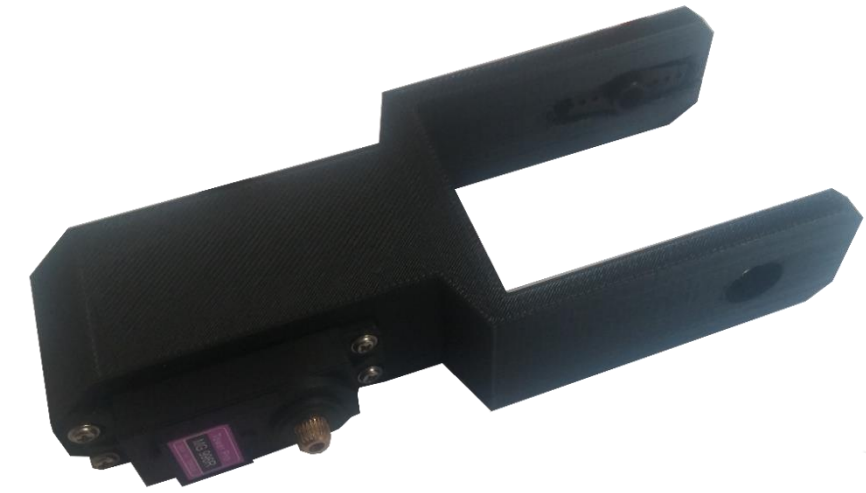
Bir önceki başlık altına belirtilen bilgisayar ortamında çizilen parçalar 3D yazıcıda %50 doluluk oranı ile ABS filamentle basılmıştır. Bu başlık altında basılan parçalara ait görsellere yer verilmiştir.



[23] 3.6. Parçaların montajlanmış hali (3D)



[24] 3.7. Gövde tabanı (3D)

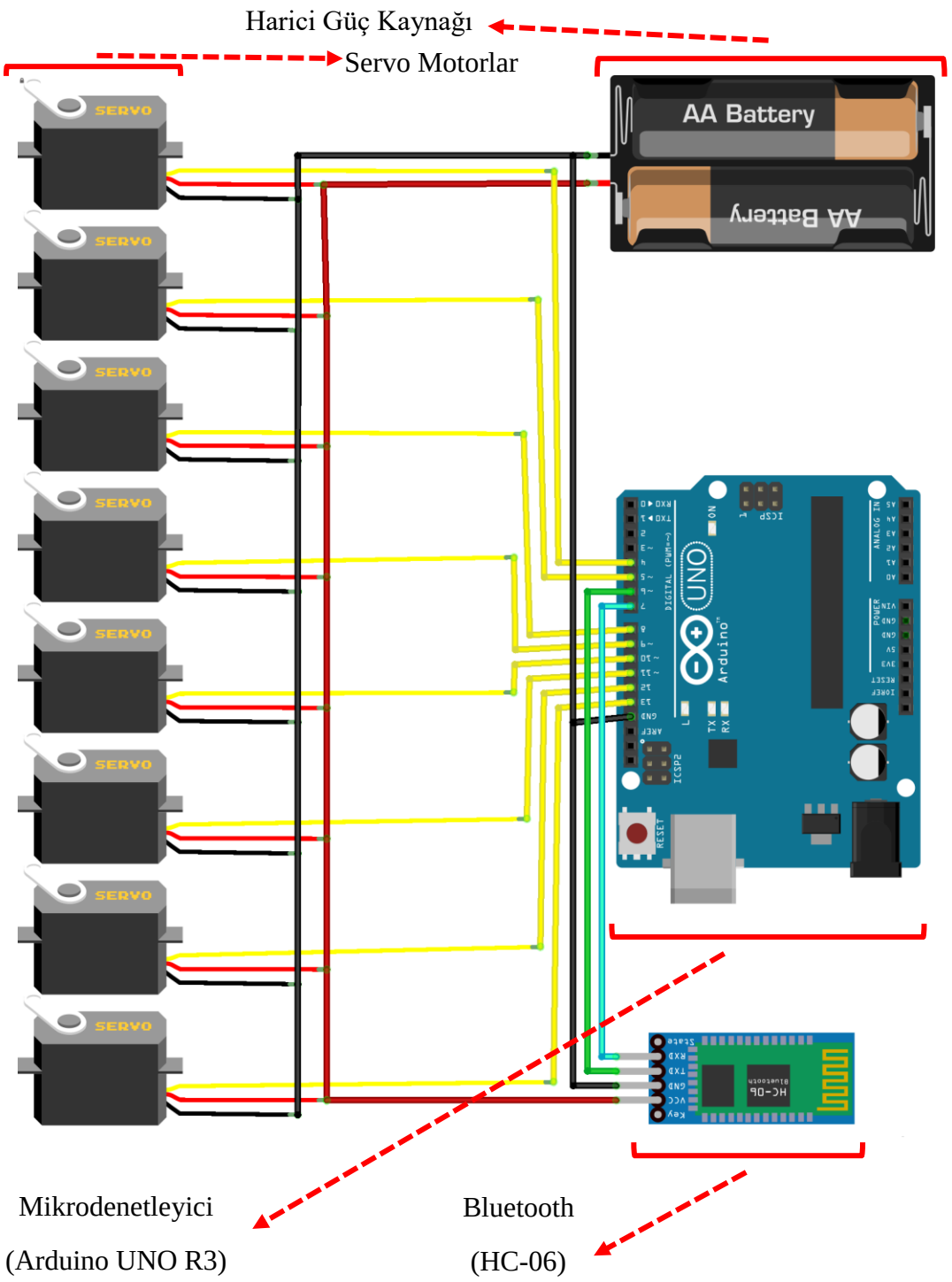


[25] 3.8. Bacak (3D)

3. 3. Devre Tasarımı

Daha önceki başlıklarda bahsedildiği gibi robotta hareket servo motorlar yardımıyla sağlanmıştır ve servo motorların kontrolü ise Arduino R3 kullanarak gerçekleştirilmiştir. Robotun yapımında sekiz adet servo motor kullanılmıştır. Servo motorların çektiği akım servo başlığında da belirtildiği gibi minimum 500 mA'dir. Arduino çıkışından alınacak maksimumu gerilim 50 mA olduğu için kart üzerinden doğrudan motorları beslemek mümkün olmayacaktır. Bu yüzden sisteme motorları sürmek için ekstra bir güç kaynağı bağlanması gerekmektedir. Şekilde göründüğü üzere harici güç kaynağından motorlar beslenmiş ve harici güç kaynağının GND çıkış ile kart üzerindeki GND çıkışı aynı hatta bağlanmıştır. Servo motorların sinyal kabloları kart üzerindeki 4, 5, 8, 9, 10 ,11, 12, 13 portlarına bağlanarak servo motorların kontrolü sağlanmıştır.

Devrede kullandığımız bir diğer eleman olan bluetooth modülünü kontrol etmek için öncelikler 6. Ve 7. pinlerine yazılım yardımıyla RX ve TX atamaları yapılmış ve modül üzerindeki RX ve TX portlarına bağlanmıştır. Bu bağlantının RX→TX ve TX→RX şeklinde olmasına dikkat edilmelidir.

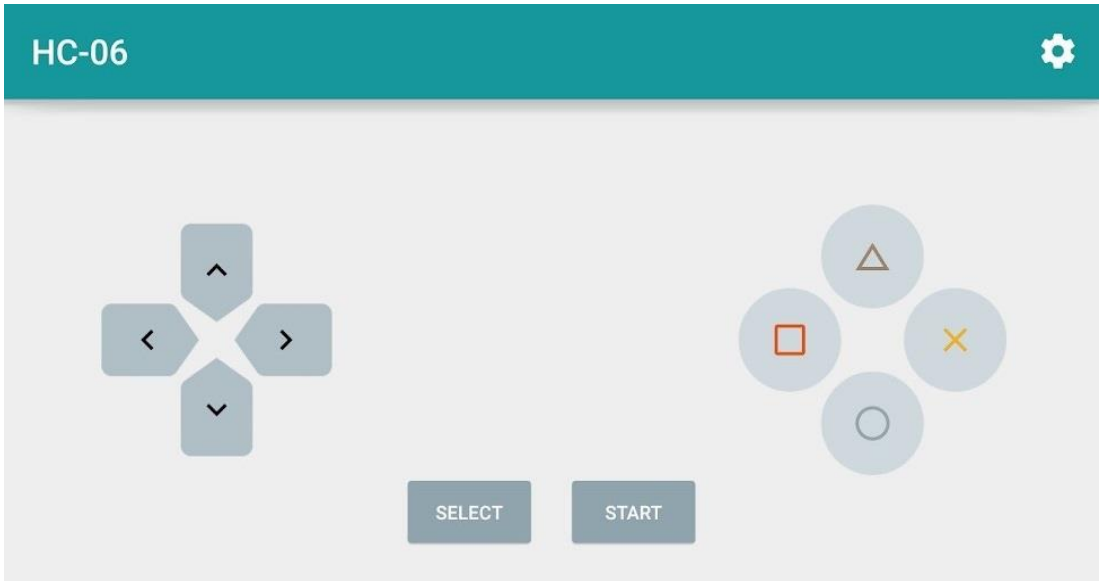


[26] 3.9. Devre tasarımı

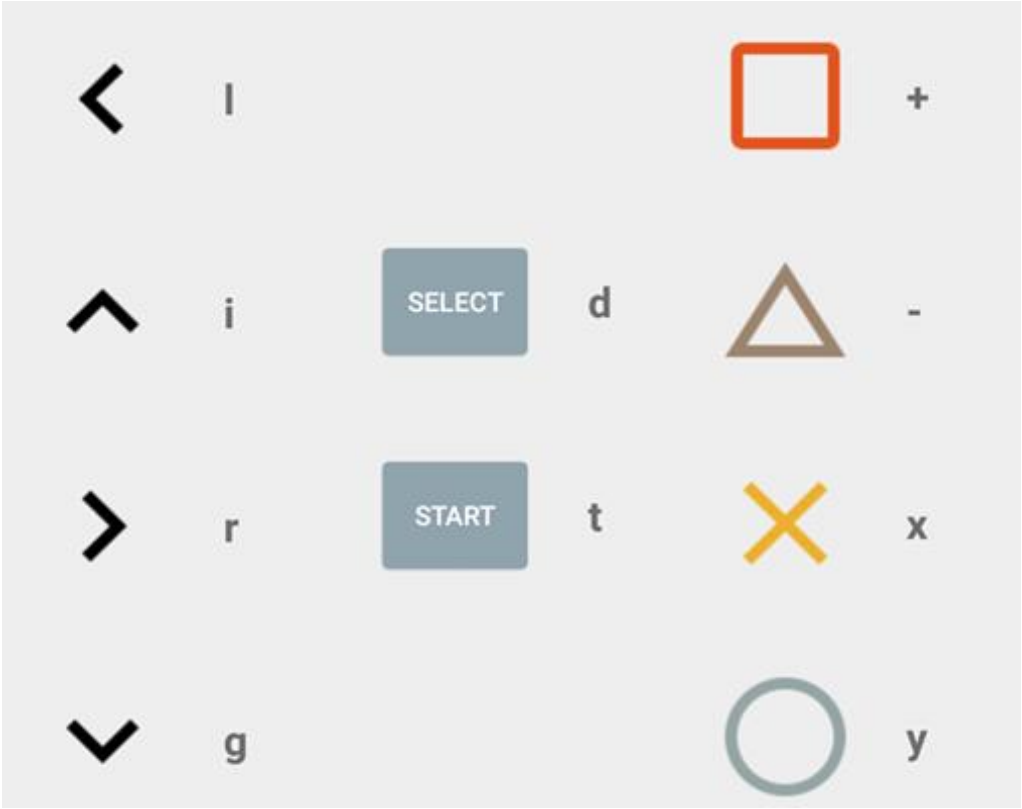
4. BLUETOOTH İLE KONTROL

Telefon ve Arduino Uno R3 mikrodenetleyicinin arasında iletişimi sağlamak için HC-06 bluetooth modülü kullanılmıştır. Mikrodenetleyicinin 6. ve 7. pinleri kart ile modül arasında seri haberleşmeyi sağlamak için kullanılmıştır. Robotu kumanda etmek için ise android işletim sistemine sahip bir akıllı telefon seçilmiştir. Bu proje üzerinde kullanılan uygulama Google Play üzerinden temin edilen Arduino Bluetooth Controller isimli uygulamadır.

Uygulama üzerindeki butonlara harf atamaları yapılarak istenilen eylem gerçekleştirmek istendiğinde bu butonlar basılması ile gerçekleşmesi sağlanmıştır. Sistem şu şekilde çalışır: Uygulama üzerinde bir butona basıldığı zaman buton üzerine kayıtlı olan harf bluetooth ile HC-06 modülüne gönderilir daha sonra modül ile mikrodenetleyici arasındaki seri haberleşme sayesinde buton değerinin devreye sokacağı kod döngesi karta bildirilmiş olur.



[27] 4.1. Kumanda ekranı



[28] 4.2. Kumanda atamaları

Üstte bahsedilen çalışma mantığına örnek olarak aşağıdaki ifadeler verilebilir. Kumandadan alınan değer data isimli değişkene atanmıştır. Bu değişken if komutu ile sorgulanarak eğer data değişkeni sorguda istenen değere uyuyorsa döngü içerisine alınmıştır.

Kumanda atamalarında belirtilen tuşların görevlerine alt başlıklarda değinilmiştir. Alt başlıklarda yer alan hareket kodları tanımlamasının ayrıntıları kod başlığı altında verilmiştir.

4.1. İleri Hareket

Robotun doğrusal olarak hareket etmesini sağlar.

[29] 4.3. İleri hareket (i)

```
if (data_received == 'i')
   {
     //**********İLERİ HAREKET KODLARI**********//
   }
```

4.2. Geri Hareket

Robotun gövdesi pozisyon değiştirmeden geriye hareketi sağlar.

[30] 4.4. Geri hareket (g)

```
if (data_received == 'g')
   {
    //**********GERİ HAREKET KODLARI**********//
   }
```

4.3. Sağa Hareket

Robotun gövdesi pozisyon değiştirmeden geriye hareketi sağlar.

[31] 4.5. Sağa hareket (r)

```
if (data_received == 'r')
   {
    //**********SAĞA HAREKET KODLARI**********//
   }
```

4.4. Sola Hareket

Robotun gövdesi pozisyon değiştirmeden geriye hareketi sağlar.

[32] 4.6. Sola hareket (l)

```
if (data_received == 'l')
   {
     //**********SOLA HAREKET KODLARI**********//
   }
```

4.5. Test

Test girdisi geldiğinde robot üzerindeki iletişimin sağlanıp sağlanmadığı, servo motorlara elektrik gelip gelmediği gibi durumları kontrol etmek için robota bir dizi hareket yaptırır.

[33] 4.7. Test (t)

```
if (data_received == 't')
   {
    //**********TEST HAREKETİ KODLARI**********//
   }
```

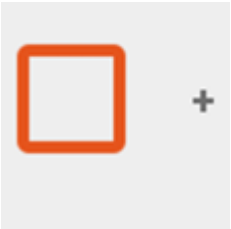
4.6. Dur

Servo değerleri robotun dengede olduğu açı durumlarına gelerek robotu hareketsiz bırakır.

[34] 4.8. Dur (d)

```
if (data_received == 'd')
   {
    //**********DUR HAREKETİ KODLARI**********//
   }
```

4.7. Sağa Kırk Beş Derece Dönüş Hareketi



[35]] 4.9. Sağa kırk beş derece dönme hareketi (+)

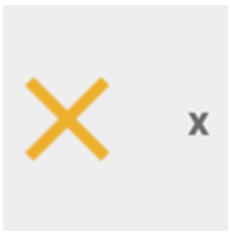
```
if (data_received == '+')
   {
    //*****SAĞA KIRK BEŞ DERECE DÖNME HAREKETİ KODLARI*****//
   }
```

4.8. Sol Kırk Beş Derece Dönüş Hareketi

[36] 4.10. Sola kırk beş derece dönme hareketi (-)

```
if (data_received == '-')
   {
     //*****SOLA KIRK BEŞ DERECE DÖNME HAREKETİ KODLARI*****//
}
```

4.9. Sağa On Beş Derece Dönüş Hareketi



[37] 4.11. Sağa on beş derece dönme hareketi (x)

```
if (data_received == 'x')
   {
     //*****SAĞA ON BEŞ DERECE DÖNME HAREKETİ KODLARI*****//
}
```

4.10. Sola On Beş Derece Dönüş Hareketi

[38] 4.12. Sola on beş derece dönme hareketi (y)

```
if (data_received == 'y')
   {
     //*****SOLA ON BEŞ DERECE DÖNME HAREKETİ KODLARI*****//
   }
```

4. KOD

Robotun fiziksel yapısında olduğu gibi yazılımında da gövdenin altını ve üstünü çalıştıracak kodlar ayrı ayrı değerlendirilir. Robotu hareket ettiren alt kısmını çalıştıracak kodların yazılmasında tek bir amaç güdüldüğü için bu kodlar robotun üst kısmındaki herhangi bir değişiklikten etkilenmeden her farklı gövde üstü eklentisi için aynı şekilde olacaktır. Robotun hareketli kısmına ait kodlar yazılırken dikkat edilmesi gereken en önemli faktör robotun ağırlık merkezinin bacaklarla olan ilişkisidir. Hareket boyunca robotun dört ayağından üçü mutlaka zemine değiyor olmalıdır aksi taktirde robot dengesini sağlayamayacağı için hareket edemeyecektir. Yere basan ayakların durumu da büyük önem taşımaktadır çünkü eklemlerin oluşturacağı üçgenin ağırlık merkezi ile robotun ağırlık merkezi ne kadar birbirine yakın olursa robot o kadar dengeli olur. Hareket kodları yazılırken robotun karşılaşa bileceği durumlarda göz önüne alınmalıdır. Sadece ileri giden bir robot proje için istenilenden uzak bir hedeftir. Aşağıdaki kodun içerisinde de belirtildiği gibi robota; ileri, geri, pozisyonunu koruyarak sağa yönelme, pozisyonunu koruyarak sola yönelme, kendi etrafında sola ve sağa dönme gibi hareket kodları tanıtılmış. Kodlar Arduino ile yazılmıştır.

4.1. Arduino ile Programlama

Arduino bir G/Ç kartı ve Processing/Wiring dilinin bir uygulamasını içeren geliştirme ortamından oluşan bir fiziksel programlama platformudur.

Arduino kartlarının donanımında bir adet Atmel AVR mikrodenetleyici (ATmega328, ATmega2560, ATmega32u4 gibi) ve programlama ve diğer devrelere bağlantı için gerekli yan elemanlar bulunur. Her Arduino kartında en azından bir 5 voltluk regüle entegresi ve bir 16MHz kristal osilator (bazılarında seramik rezonatör) vardır. Arduino

kartlarında programlama için harici bir programlayıcıya ihtiyaç duyulmaz, çünkü kart-taki mikrodenetleyiciye önceden bir bootloader programı yazılıdır.

Arduino geliştirme ortamı (IDE), Arduino bootloader (Optiboot), Arduino kütüphaneleri, AVRDude (Arduino üzerindeki mikrodenetleyici programlayan yazılım) ve derleyiciden (AVR-GCC) oluşur.

Arduino yazılımı bir geliştirme ortamı (IDE) ve kütüphanelerden oluşur. IDE, Java dilinde yazılmıştır ve Processing adlı dilin ortamına dayanmaktadır. Kütüphaneler ise C ve C++ dillerinde yazılmıştır ve AVR-GCC ve AVR Libc. ile derlenmiştir. Arduino kaynak kodlarına buradan ulaşabilirsiniz.

Optiboot bileşeni Arduino 'nun bootloader bileşenidir. Bu bileşen, Arduino kartlarının üzerindeki mikrodenetleyicinin programlanmasını sağlayan bileşendir.

Arduino 'nun bu kadar çok tercih edilmesini sağlayan en önemli bileşen ise mikrodenetleyici konusunda detaylı bilgi sahibi olmayı gerektirmeden herkesin programlama yapabilmesini sağlayan Arduino kütüphaneleridir. AVRDude bileşeni ise derlenen kodları programlamak için kullanılır.

4.2. Kodlar

```
//*****Servo kütüphanesi ve seri iletişim kütüphanelerinin eklenmesi *****//

#include <Servo.h>
#include <SoftwareSerial.h>

//**********Servoların tanıtılması**********//
```

```
SoftwareSerial bluetooth(7, 6);

Servo motor1;
Servo motor2;

Servo motor3;
Servo motor4;

Servo motor5;
Servo motor6;

Servo motor7;
Servo motor8;

void setup()
{

 //**********MOTOR PİN ATAMALARI**********//

 //***Yeşil(Bacak Bir)***//

 motor1.attach(12);
 motor2.attach(13);
```

```
//***Sarı(Bacak İki)***//

motor3.attach(8);
motor4.attach(9);

//***Mor(Bacak Üç)***//

motor5.attach(10);
motor6.attach(11);

//***Lacivert(Bacak Dört)***//

motor7.attach(4);
motor8.attach(5 );

//**********Başlangıç değerleri**********//

motor1.write(85);
motor2.write(150);

motor3.write(85);
motor4.write(10);

motor5.write(80);
```

```
  motor6.write(145);


  motor7.write(70);
  motor8.write(20);


  Serial.begin(9600);
  bluetooth.begin(9600);


}


void loop() {


if (bluetooth.available())
  {
    char data_received;
    data_received = bluetooth.read();


    if (data_received == 'i')
    {


      //**********İLERİ**********//


      //***Bacak Üç***//
```

```
motor6.write(145);
delay(150);
motor5.write(75);
delay(150);
motor6.write(85);
delay(150);
motor5.write(120);
delay(150);
motor6.write(145);
delay(500);

//***Bacak Bir***//

motor2.write(150);
delay(150);
motor1.write(43);
delay(150);
motor2.write(90);
delay(150);
motor1.write(85);
delay(150);
motor2.write(150);
delay(500);
```

```
//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(75);
delay(150);
motor8.write(80);
delay(150);
motor7.write(35);
delay(150);
motor8.write(20);
delay(500);

//***Bacak İki***//

motor4.write(10);
delay(150);
motor3.write(125);
delay(150);
motor4.write(85);
delay(150);
motor3.write(80);
delay(150);
motor4.write(10);
```

```
  delay(500);

  //***iTME***//

  motor1.write(43);
  motor3.write(125);
  motor5.write(80);
  motor7.write(70);
  delay(500);

}

if (data_received == 'g' )
{

  //**********GERİ**********//

  //***Bacak İki***//

  motor4.write(10);
  delay(150);
  motor3.write(80);
  delay(150);
  motor4.write(85);
```

```
delay(150);
motor3.write(125);
delay(150);
motor4.write(10);
delay(500);

//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(35);
delay(150);
motor8.write(80);
delay(150);
motor7.write(75);
delay(150);
motor8.write(20);
delay(500);

//***Bacak Bir***//

motor2.write(150);
delay(150);
motor1.write(85);
```

```
delay(150);
motor2.write(90);
delay(150);
motor1.write(43);
delay(150);
motor2.write(150);
delay(500);

//***Bacak Üç***//

motor6.write(145);
delay(150);
motor5.write(120);
delay(150);
motor6.write(85);
delay(150);
motor5.write(75);
delay(150);
motor6.write(145);
delay(500);

//***iTME***//

motor1.write(85);
```

```
  motor3.write(80);
  motor5.write(120);
  motor7.write(35);
  delay(500);

}

if (data_received == 'r')
{

  //**********SAĞ**********//

  //***Bacak Dört***//

  motor8.write(20);
  delay(150);
  motor7.write(75);
  delay(150);
  motor8.write(80);
  delay(150);
  motor7.write(120);
  delay(150);
  motor8.write(20);
  delay(500);
```

```
//***Bacak Üç***//

motor6.write(145);
delay(150);
motor5.write(30);
delay(150);
motor6.write(85);
delay(150);
motor5.write(75);
delay(150);
motor6.write(145);
delay(500);

//***Bacak İki***//

motor4.write(10);
delay(150);
motor3.write(80);
delay(150);
motor4.write(85);
delay(150);
motor3.write(45);
delay(150);
```

```
motor4.write(10);
delay(500);

//***Bacak Bir***//

motor2.write(150);
delay(150);
motor1.write(120);
delay(150);
motor2.write(90);
delay(150);
motor1.write(85);
delay(150);
motor2.write(150);
delay(500);

//***iTME***//

motor1.write(130);
motor3.write(75);
motor5.write(30);
motor7.write(75);
delay(500);
```

```
}

if (data_received == 'l')
{

 //**********SOL**********//

 //***Bacak Bir***//

 motor2.write(150);
 delay(150);
 motor1.write(85);

 delay(150);
 motor2.write(90);
 delay(150);
 motor1.write(120);
 delay(150);
 motor2.write(150);
 delay(500);

 //***Bacak İki***//
```

```
motor4.write(10);
delay(150);
motor3.write(45);
delay(150);
motor4.write(85);
delay(150);
motor3.write(80);
delay(150);
motor4.write(10);
delay(500);

//***Bacak Üç***//

motor6.write(145);
delay(150);
motor5.write(75);
delay(150);
motor6.write(85);
delay(150);
motor5.write(30);
delay(150);
motor6.write(145);
delay(500);
```

```
  //***Bacak Dört***//

  motor8.write(20);
  delay(150);
  motor7.write(120);
  delay(150);
  motor8.write(80);
  delay(150);
  motor7.write(75);
  delay(150);
  motor8.write(20);
  delay(500);

  //***iTME***//

  motor1.write(85);
  motor3.write(45);
  motor5.write(75);
  motor7.write(120);
  delay(500);

}

if (data_received == 'd')
```

```
{

 //**********DUR**********//

 //***Bacak Bir***//

 motor1.write(90);
 delay(150);
 motor2.write(150);
 delay(150);

 //***Bacak İki***//

 motor3.write(90);
 delay(150);
 motor4.write(10);
 delay(150);

 //***Bacak Üç***//

 motor5.write(80);
 delay(150);
 motor6.write(145);
 delay(150);
```

```
  //***Bacak Dört***//

  motor7.write(70);
  delay(150);
  motor8.write(20);
  delay(150);

}

if (data_received == 't')
{

  motor1.write(90);
  delay(150);
  motor3.write(90);
  delay(150);
  motor5.write(80);
  delay(150);
  motor7.write(70);
  delay(150);

  motor2.write(150);
  delay(250);
```

```
motor2.write(90);
delay(250);
motor2.write(150);
delay(250);

motor4.write(10);
delay(250);
motor4.write(70);
delay(250);
motor4.write(10);
delay(250);

motor5.write(80);
delay(250);
motor6.write(145);
delay(250);
motor6.write(95);
delay(250);
motor6.write(145);
delay(250);

motor8.write(20);
delay(250);
motor8.write(80);
```

```
  delay(250);
  motor8.write(20);
  delay(250);

}

if (data_received == '+')
{

  //**********KIRK BEŞ DERECE SAĞA DÖN**********//

  //***Bacak Bir***//

  motor2.write(150);
  delay(150);
  motor1.write(90);
  delay(150);
  motor2.write(90);
  delay(150);
  motor1.write(45);
  delay(150);
  motor2.write(150);
  delay(150);
```

```
//***Bacak Üç***//

motor6.write(145);
delay(150);
motor5.write(80);
delay(150);
motor6.write(85);
delay(150);
motor5.write(35);
delay(150);
motor6.write(145);
delay(150);

//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(70);
delay(150);
motor8.write(80);
delay(150);
motor7.write(35);
delay(150);
motor8.write(20);
```

```
delay(150);

//***Bacak İki***//

motor4.write(15);
delay(150);
motor3.write(90);
delay(150);
motor4.write(65);
delay(150);
motor3.write(45);
delay(150);
motor4.write(15);
delay(150);

//***Son Dönme***//

motor1.write(90);
delay(100);
motor3.write(90);
delay(100);
motor5.write(80);
delay(100);
motor7.write(70);
```

```
  delay(100);

}

if (data_received == 'x')
{

  //**********ON BEŞ DERECE SAĞA DÖN**********//

  //***Bacak Bir***//

  motor2.write(150);
  delay(150);
  motor1.write(90);
  delay(150);
  motor2.write(90);
  delay(150);
  motor1.write(75);
  delay(150);
  motor2.write(150);
  delay(500);

  //***Bacak Üç***//
```

```
motor6.write(145);
delay(150);
motor5.write(80);
delay(150);
motor6.write(85);
delay(150);
motor5.write(65);
delay(150);
motor6.write(145);
delay(500);

//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(70);
delay(150);
motor8.write(80);
delay(150);
motor7.write(55);

delay(150);
motor8.write(20);
delay(500);
```

```
//***Bacak İki***//

motor4.write(15);
delay(150);
motor3.write(90);
delay(150);
motor4.write(65);
delay(150);
motor3.write(75);
delay(150);
motor4.write(15);
delay(500);

//***Son Dönme***//

motor1.write(90);
delay(100);
motor3.write(90);
delay(100);
motor5.write(80);
delay(100);
motor7.write(70);
delay(100);
```

```
}

if (data_received == '-')
{

  //**********KIRK BEŞ DERECE SOLA DÖN**********//

  //***Bacak Bir***//

  motor2.write(150);
  delay(150);
  motor1.write(85);
  delay(150);
  motor2.write(90);
  delay(150);
  motor1.write(125);
  delay(150);
  motor2.write(150);
  delay(500);

  //***Bacak İki***//

  motor4.write(15);
  delay(150);
```

```
motor3.write(85);
delay(150);
motor4.write(65);
delay(150);
motor3.write(125);
delay(150);
motor4.write(15);
delay(500);

//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(70);
delay(150);
motor8.write(80);
delay(150);
motor7.write(115);
delay(150);
motor8.write(20);
delay(500);

//***Bacak Üç***//
```

```
  motor6.write(145);
  delay(150);
  motor5.write(80);
  delay(150);
  motor6.write(85);
  delay(150);
  motor5.write(120);
  delay(150);
  motor6.write(145);
  delay(500);

  //***Son Dönme***//

  motor1.write(90);
  delay(100);
  motor3.write(90);
  delay(100);
  motor5.write(80);
  delay(100);
  motor7.write(70);
  delay(500);
}
if (data_received == 'y')
{
```

```
//**********ON BEŞ DERECE SOLA DÖN**********//

//***Bacak Bir***//

motor2.write(150);
delay(150);
motor1.write(85);
delay(150);
motor2.write(90);
delay(150);
motor1.write(100);
delay(150);
motor2.write(150);
delay(500);

//***Bacak İki***//

motor4.write(15);
delay(150);
motor3.write(85);
delay(150);
motor4.write(65);
delay(150);
```

```
motor3.write(100);
delay(150);
motor4.write(15);
delay(500);

//***Bacak Dört***//

motor8.write(20);
delay(150);
motor7.write(70);
delay(150);
motor8.write(80);
delay(150);
motor7.write(85);
delay(150);
motor8.write(20);
delay(500);

//***Bacak Üç***//

motor6.write(145);
delay(150);
motor5.write(80);
delay(150);
```

```
      motor6.write(85);
      delay(150);
      motor5.write(95);
      delay(150);

      motor6.write(145);
      delay(500);

      //***Son Dönme***//

      motor1.write(90);
      delay(100);
      motor3.write(90);
      delay(100);
      motor5.write(80);
      delay(100);
      motor7.write(70);
      delay(500);

    }

  }

}
```

5. KAYNAKÇA

1. (**How to Program a Quadruped Robot with Arduino**, 2016)

[https://makezine.com/2016/11/22/robot-quadruped-arduino-program/]

2. (**Arduino Quadruped Robot**, 2016)

[https://www.instructables.com/id/Synopsis/]

3. (**ARDUINO QUADRUPED ROBOT – STALKER**, 2013)

[https://oscarliang.com/arduino-quadruped-robot-stalker/]

4. (**ESP8266 WIFI AP Controlled Quadruped Robot**, 2018)

[https://www.instructables.com/id/ESP8266-WIFI-AP-Controlled-Quadruped-Robot/]

5. (**Servo Motor Nedir? Çeşitleri ve Çalışma Prensibi**, 2015)

[https://maker.robotistan.com/rc-servo-motor-nedir/]

6. (Arduino UNO R3, 2018)

[https://maker.robotistan.com/arduino-uno/]

7. (**MG996R Digital Servo**, 2013)

[https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf]

8. (**Bluetooth ile İletişim**, 2016)

[https://gelecegiyazanlar.turkcell.com.tr/konu/arduino/egitim/arduino-201/bluetooth-ile-iletisim]

6. ÖZGEÇMİŞ

Engin AYDIN 1996 yılında Bursa'da doğdu; ilk ve orta öğretimini doğduğu şehirde Nezir Gencer İlköğretim okulunda tamamladı; Şehit Onbaşı Hakan Yutkun Anadolu Lisesini tamamladıktan sonra 2015 Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

İletişim Bilgileri

E-posta: engiin.aydin@gmail.com

Tel: 0534 631 75 79

MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ

QUADRUPED ROBOT

HAZIRLAYAN
ENGİN AYDIN
PROJE DANIŞMANI
DR.ÖĞR.ÜYESİ İBRAHİM ÇAYIROĞLU

GİRİŞ

Bu projede günümüzde hızla önem kazanan insansız hava araçlarının geliştirilmesinin ve üretilmesinin eksiklerini gidermek adına ortaya çıkan insansız kara araçlarının tasarımında farklı bir yaklaşım sağlanması amaçlanmıştır.

Kara araçlarına esneklik kazandırmak adına, palet veya tekerlek gibi elemanlarla hareket imkânı kısıtlı bir şekilde hareket ettirmek yerine eklemlerle desteklenmiş uzuvlarla daha geniş bir hareket uzayında hareket etmesi sağlanacaktır. Tasarlanacak robot hareketi sağlayan ana gövde ve istenilen göreve göre değiştirilecek bir başlıktan oluşmaktadır.

ÇALIŞMA PRENSİBİ

İstenilen yönde doğrusal hareket etmesi veya bulunduğu konumda istenilen bir açıyla dönmesi için her konuma özgü bir adımlık hareket veya bir derecelik dönüş için gerekli servo açılarının kodlarını önceden tanıtılacak ve kumanda ile belirli bir yöne gitmesi istenildiğinde bu değerler bir döngüye sokulup yön girdisi geldiği sürece istenilen harekete devam edecek.

Hareket sırasında uzuvlardan biri havada olacağından diğer üç uzvun gövdeyi dengede tutabilmek için eşkenar üçgen oluşturacak şekilde konum alması sağlanarak ağırlık merkezinin dengelenmesi amaçlanmıştır.

KULLANILAN MALZEMELER

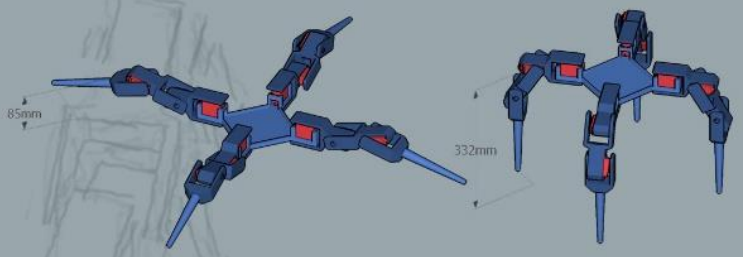
Mikrodenetleyici: MSP-EXPMSP430F5529-LP
Servo Motorlar: Tower Pro MG996
Bluetooth Modülü: HC06

TASARIM

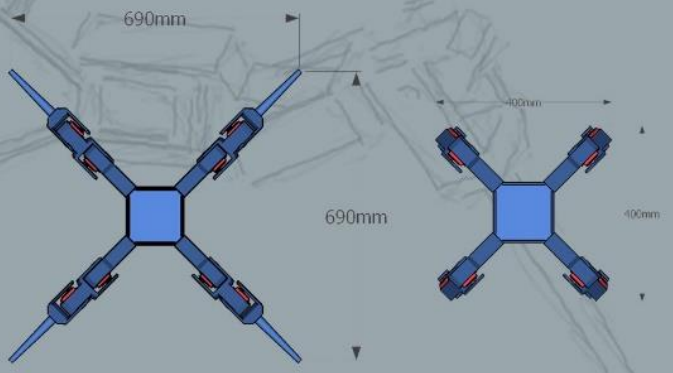
Tasarım genel haliyle mikrodenetleyici, pil ve sensörlerin bulunduğu gövdeden ve gövdeye bağlı dört uzuvdan oluşmaktadır. Hareket bu uzuvlar üzerindeki servo motorlar yardımı ile sağlanmaktadır.

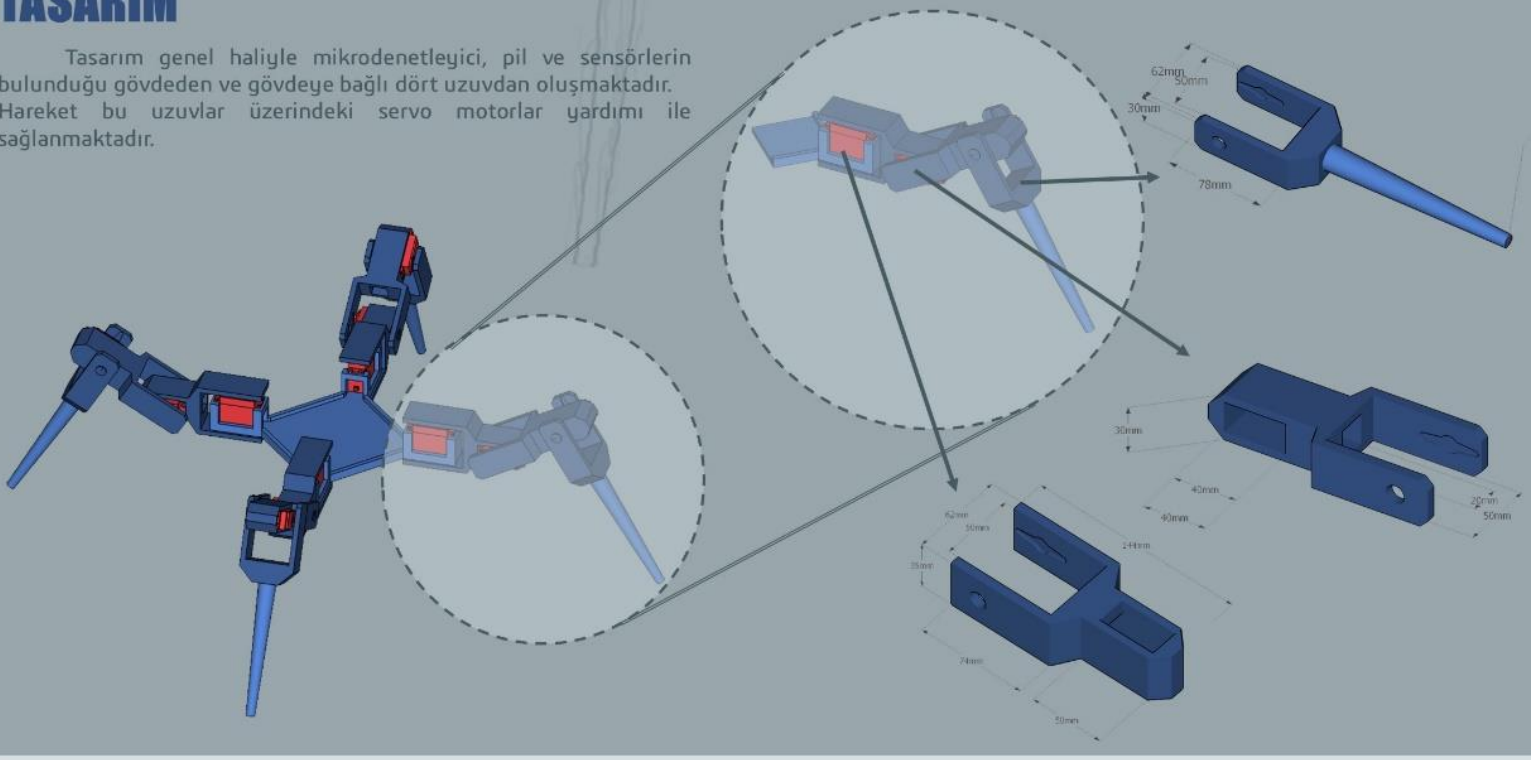
ÇALIŞMA SINIRLARI

Minimum ve maksimum çalışma yükseklikleri.



Minimum ve maksimum çalışma yüksekliklerinde genişlikler.





İletişim

Hazırlayan
Engin Aydın
engiin.aydin@gmail.com

Proje Danışmanı
Dr. Öğr. Üyesi İbrahim Çayıroğlu
icayiroglu@karabuk.edu.tr