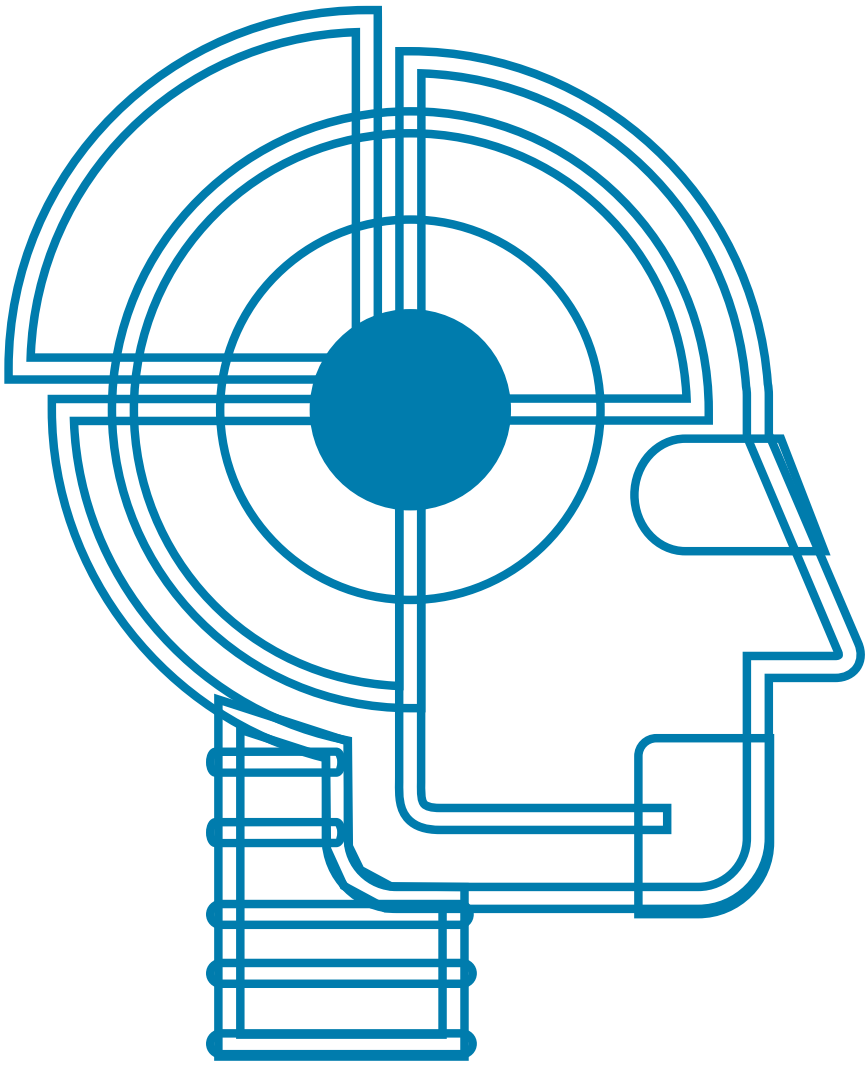
#MİLLİ TEKNOLOJİ HAMLESİ

DENEYAP TÜRKİYE

İLERİ ROBOTİK

LİSE



T.C. SANAYİ VE TEKNOLOJİ BAKANLIĞI | T.C. GENÇLİK VE SPOR BAKANLIĞI | TÜBİTAK | TÜRKİYE TEKNOLOJİ TAKIMI

TÜBİTAK Deneyap Kitapları 24

İLERİ ROBOTİK
LİSE

Doç. Dr. Cüneyt YILMAZ
Dr. Öğr. Üyesi Mehmet Emin AKTAN
Prof. Dr. Vasfi Emre ÖMÜRLÜ
Arş. Gör. Mehmet İŞCAN

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2021

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel materyaller, TÜBİTAK'tan yazılı izin alınmadan
tümüyle veya kısmen çoğaltılamaz ve yayımlanamaz.
Kitabın PDF formatındaki elektronik nüshasına
https://yayinlar.tubitak.gov.tr/deneyap-atolyesi adresinden ulaşılabilir.
TÜBİTAK Deneyap Kitapları DENEYAP TÜRKİYE Projesi kapsamında hazırlanmıştır.

ISBN 978-605-312-544-0
Yayıncı Sertifika No: 47703

Yayım Tarihi: 2024

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editör: Dr. İpek PİRPİROĞLU GENCER
Düzelti: Dr. Mustafa ORHAN
Telif İşleri Sorumlusu: Havva Hilal KAÇAR

TÜBİTAK Bilim ve Toplum Başkanlığı
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 96 50
e-posta: deneyap@tubitak.gov.tr
https://yayinlar.tubitak.gov.tr/deneyap-atolyesi

DENEYAP

Teknoloji Atölyeleri

İLERİ ROBOTİK

LİSE

Doç. Dr. Cüneyt YILMAZ
Dr. Öğr. Üyesi Mehmet Emin AKTAN
Prof. Dr. Vasfi Emre ÖMÜRLÜ
Arş. Gör. Mehmet İŞCAN

TÜBİTAK

İçindekiler

İçindekiler 2
Öğretim Kılavuzu 5
İLERİ ROBOTİK DERSİ ÖĞRETİM KILAVUZU 5
Dersin İşleyişi ile İlgili Önemli Hususlar 11
Kaynakça 12
1. Bölüm: Robot Kavramı ve İleri Robot Teknolojilerine Giriş 13
1. ROBOT NEDİR? 13
2. DERS KAPSAMINDA GERÇEKLEŞTİRİLECEK ROBOTLAR 15
3. MEKANİZMA VE ROBOT KAVRAMLARI 17
3.1. Genel Robot Tipleri 18
3.2. Serbestlik Derecesi 20
4. DEĞERLENDİR 21
5. İLAVE ETKİNLİK 22
2. Bölüm: Işık Sensörleri ile Servo Motor Kontrolü 23
1. GÖZLE VE UYGULA 24
1.1. Gözle: Deneyap Geliştirme Kartı 24
1.2. Uygula: Deneyap Geliştirme Kartı Arduino IDE Kütüphanesinin Yüklenmesi 25
1.3. Gözle: LDR (Light Dependent Resistor - Işığa Duyarlı Direnç) Nedir? 28
1.4. Uygula: LDR Sensörünün Dijital Çıkışı ile Servo Motor Kontrolü 29
1.5. Uygula: LDR Sensörlerinin Analog Çıkışları ile Servo Motor Kontrolü 35
2. TASARLA VE ÜRET 37
2.1. Tasarla: Ateş Ölçer Cihazı 37
2.2. Üret: Ateş Ölçer Cihazı 38
3. DEĞERLENDİR 38
4. İLAVE ETKİNLİK 39
3. Bölüm: Mesafe ve IMU Sensörleri ile Servo ve DC Motor Kontrolü 40
1. GÖZLE VE UYGULA 41
1.1. Gözle: Ultrasonik Mesafe Sensörü Nedir? 41
1.2. Uygula: Ultrasonik Sensör ile Servo Motor Kontrolü 41
1.3. Gözle: IMU (Inertial Measurement Unit) Sensörü Nedir? 46
1.4. Uygula: IMU Sensörü Değerlerinin Ekrana Yazdırılması 47
1.5. Uygula: IMU Sensör Değerlerinin Pan-Tilt Açı Değerlerine Dönüştürülmesi 48
1.6. Uygula: IMU Sensörü ile DC Motor Kontrolü 49
2. TASARLA VE ÜRET 53

2.1. Tasarla: Denge çubuğu mekanizması 53
2.2. Üret: Denge çubuğu mekanizması..... 53
3. DEĞERLENDİR 54
4. İLAVE ETKİNLİK..... 55
4. Bölüm: İki Serbestlik Dereceli Robot Mekanizması..... 56
1. GÖZLE VE UYGULA 57
1.1. Gözle: Kumanda Kolu Nedir?..... 57
1.2. Uygula: Deneyap Geliştirme Kartı Üzerinde Kumanda Kolu Çalıştırılması 58
1.3. Uygula: Deneyap Geliştirme Kartı ile Çift Servo Motor Kontrolü..... 61
1.4. Uygula: İki Motorlu Bir Sistemin Kumanda Kolu ile Kontrolü..... 66
1.5. Uygula: IMU Sensörü ile Çift Servo Motor Konum Kontrolü..... 69
1.6. Uygula: Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü..... 71
2. TASARLA VE ÜRET..... 78
2.1. Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü..... 78
3. DEĞERLENDİR..... 85
5. İLAVE ETKİNLİK..... 86
5. Bölüm: Üç Serbestlik Dereceli Robot Mekanizması 87
1. GÖZLE VE UYGULA..... 88
1.1. Gözle: Üç Serbestlik Dereceli Robotların Yapısı ve İşlevleri 88
1.2. Uygula: Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü 88
2. TASARLA VE ÜRET.....106
2.1. Tasarla: Üç Serbestlik Dereceli Robot için Çalışma Ortamı 106
2.2. Üret: Üç Serbestlik Dereceli Robot için Çalışma Ortamı 106
3. DEĞERLENDİR..... 107
4. İLAVE ETKİNLİK..... 107
6. Bölüm: Beş Serbestlik Dereceli Robot Mekanizması..... 108
1. GÖZLE VE UYGULA.....109
1.1. Gözle: Beş Serbestlik Dereceli Robotların Yapısı ve İşlevleri..... 109
1.2. Uygula: Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü 109
2. TASARLA VE ÜRET.....130
2.1. Tasarla: Beş Serbestlik Dereceli Robot için Çalışma Ortamı 130
2.2. Üret: Beş Serbestlik Dereceli Robot için Çalışma Ortamı..... 130
3. DEĞERLENDİR..... 131
İLAVE ETKİNLİK..... 131
7. Bölüm: Yarışma Hakkında Bilgiler ve Raporlama 132

1. KAPSAM VE AMAÇ 132
2. TANIMLAR VE GENEL BİLGİLER 133
2.1. Tanımlar 133
2.2. Genel Bilgiler 133
3. KURALLAR VE KISITLAMALAR 133
3.1. Yarışma Kuralları 133
3.2 Yarışma Ortamı Kısıtlamaları 134
3.3. Tasarım Kısıtlamaları 135
8. Bölüm: Mekanik/Elektronik Tasarım ve Montaj 137
1. MEKANİK MONTAJDA DİKKAT EDİLMESİ GEREKENLER 137
2. ELEKTRONİK MONTAJDA DİKKAT EDİLMESİ GEREKENLER 138
9. Bölüm: Sistem Entegrasyonu ve Yazılım 139
1. YAZILIM AŞAMASINDA DİKKAT EDİLMESİ GEREKENLER 139

Öğretim Kılavuzu

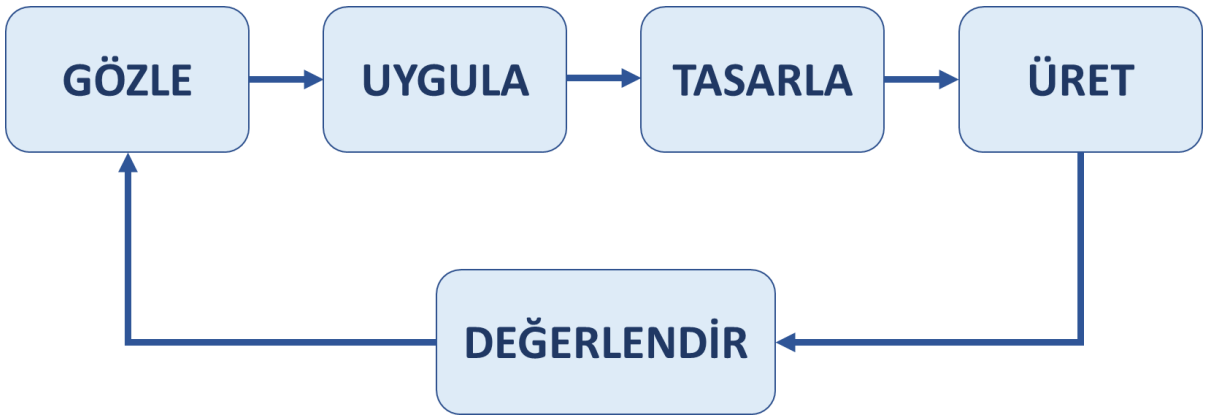
İLERİ ROBOTİK DERSİ ÖĞRETİM KILAVUZU

Öğrenme kavramı, çeşitli yaşantı ve tecrübeler sonucunda bireylerin davranışında meydana gelen uzun süreli değişimleri ifade etmektedir (Özkalp, 2005). Burada davranışta meydana gelen değişimlerin uzun süreli olması önemlidir. Bireylerin yaşamı boyunca davranışlarını etkileyen birçok faktör bulunmaktadır. Bu tür uzun süreli davranış değişikliğini ifade eden öğrenmeler sayesinde bireylerin kapasiteleri gelişmektedir. Buna bağlı olarak bireyler daha önce yapamadıkları şeyleri yapabilir hale gelmektedir. Öğrenme sonucunda bireyler hayata farklı anlamlar yükleyebilir, evreni ve kendilerinin evrendeki konumlarını daha farklı tanımlayabilir. Son yıllarda birçok bilim insanı ve kuruluş ideal eğitim modeli üzerinde çalışmış, eğitim ortamlarında öğrenciler için hangi temel yaşam becerilerinin büyük önem taşıdığını belirlemek için çaba sarf etmiştir. Bu beceriler yaygın olarak 21. yüzyıl becerileri olarak adlandırılmış ve bu beceriler için temel bir çerçeve tasarlanmaya çalışılmıştır. Bu noktada Ekonomik İşbirliği ve Kalkınma Örgütü (OECD) 21. yüzyıl becerilerini dönüştürücü beceriler olarak görmüş ve bunları üç kategoride sınıflandırmıştır: Problem çözme süreçlerinde **sorumluluk** alarak özgün bakış açıları ile **yeni değerler oluşturmak** ve **sistematik düşünme becerileri** ile problemleri çözebilmek (OECD, 2018).

Öğretme, bireyin davranışlarında uzun süreli değişiklikler meydana getirebilecek eylemlerdir. Bireylerin davranışlarında meydana gelen değişimin, bireyin kendi yaşantılarının bir ürünü olması beklenmektedir (Can, 2005). Bu nedenle öğrenme ve öğretme tasarımlarında bireylerin olayları deneyimlemesini sağlamak, farklı problemlere karşı çözüm üretebilme kabiliyetlerini geliştirmek ana hedef olmalıdır.

Bilgi, işlemsel düşünmenin ve motor becerilerinin geliştirilmesi için çeşitli yaklaşımlar kullanılabilir. Bu dersin içeriğinde donanım ve yazılım tabanlı tasarım ve uygulamaya yönelik etkinlikler yer almaktadır. Öğrencilerin robotik mekanizmaların tasarımı, bu tasarımlara uygun sensörler, tahrik elemanları ve diğer bileşenlerin seçimi ve entegrasyonu ile belirli görevleri yerine getirmek için gerekli olan yazılımların oluşturulması aşamalarını gerçekleştirmeleri beklenmektedir. Böylelikle öğrencilere robotik alanına yönelik bilişsel ve duyuşsal becerilerin kazandırılarak bu alana ilişkin davranışlarında uzun süreli değişiklikler oluşturulması hedeflenmiştir. Bahsi geçen becerilerin kazandırılmasında iki farklı yaklaşım kullanılabilmektedir. Bunlardan birincisinde eğitmen aktiftir ve öğrencilere anlatım yolu ile bilgi aktarımı gerçekleştirir. Öğrenci dinleyici konumundadır. Diğer yaklaşımda ise öğrenci aktif konumdadır. Eğitmen, öğrencinin keşfederek bilgiyi elde etmesini sağlayacak ortamı sunar. Belirli probleme yönelik uygun soruları sorup, alınan cevaba göre öğrenciyi yönlendiren taraf eğitmendir. Eğitmenin aktif, öğrencinin pasif olduğu yaklaşım 21. yüzyıl becerilerine yönelik kazanımların sağlanması noktasında yetersizdir (Papert, 1991). Öğrencinin aktif, eğitmenin yardımcı olduğu yaklaşımda ise temel kavramlar noktasında eksik olan öğrencilerin, bilgilerini üst seviye taşıma noktasında

zorluklar yaşadıkları görülmektedir (Mayer, 2004). Bu nedenlerden dolayı, ders kapsamında her iki yaklaşım da gerektiği yerlerde kullanılacaktır. Öğrencilere temel kavramların ve metotların aktarılması aşamasında anlatım yöntemi, öğrencilerden bir probleme yönelik çözüm üretmeleri beklendiğinde ise yönlendirici model uygulanacaktır. Bu amaçla İleri Robotik dersi kapsamında Resim 1'de verilen Gözle, Uygula, Tasarla, Üret ve Değerlendir (GUTÜD) öğrenme döngüsü (Çetin, İ., Üçgül, M., Top, E., Yükseltürk, E. 2021; Üçgül, M, Çetin, İ., Yükseltürk, E. & Top, E., 2021) kullanılacaktır.



Resim 1. Öğrenme Döngüsü

Gözle: Gözle adımında eğitmen aktif rol almaktadır. Bu adım iki aşamadan oluşmaktadır. Birinci aşamada, eğitmen öğrencilerin o hafta işlenecek konu hakkında bilgi edinmelerini sağlayarak, konuya olan ilgilerini ve motivasyonlarını güçlendirmelidir. Bu noktada eğitmen geçmiş konuları özetleyebilir ve öğrencilerin ilgisini çekecek örnek görsel ve videolar paylaşabilir. Geçmişte bu alanda önemli çalışmalar yapmış El-Cezerî, Leonardo da Vinci, Tesla gibi kişilerin hayatlarından, çalışmalarından ve geliştirdikleri makinalardan bahsedilebilir. Burada hikâyelendirme şeklinde bir anlatım kullanılabilir. Örneğin doğru akım ve alternatif akım kavramlarının tanıtımında Tesla ve Edison arasında geçen olayların anlatımı öğrencilerin ilgisini çekebilir. İkinci aşamada eğitmen, örnek bir robotik uygulamayı yine ilgi çekici görsel ve videolarla açıklar. Eğitmenin öğrencilerle diyalog içerisinde olması ve soru-cevaplar ile öğrencilerin konunun içerisine çekilmesi beklenmektedir. Ayrıca eğitmen, öğrencilerin yorum yapmasını ve sorular sormasını teşvik etmelidir.

Uygula: Bu bölümde eğitmen, o hafta için belirlenen uygulamaların aynısını veya benzerini öğrencilerle birlikte uygular. Örneğin, hazırlanmış iki serbestlik dereceli bir denge sistemi, verilen yöntem adımlarının uygulanmasıyla gerçekleştirilir ve test edilir. Burada eğitmen bazı adımları öğrencilere vermeyerek kendilerinin çözüm bulmasını isteyebilir.

Tasarla: Bu bölümde öğrenciler aktif rol üstlenir. Eğitmen, rehber pozisyonundadır ve öğrencilerin takıldıkları noktalarda destek vererek motivasyonlarının yüksek tutulmasını sağlamalıdır. Bununla birlikte, eğitmenin öğrencilere verdiği destek dengeli olmalı, aşırı yardımdan sakınılmalı ve kritik noktaları öğrencilerin çözmesi sağlanmalıdır. Tasarla aşamasının uygulandığı en önemli aşama

yarışmaya hazırlık aşamasıdır. Yarışmadan önceki derslerde, genellikle küçük tasarımlar veya uygulanan robotik mekanizmaların kullanımına yönelik tasarımlar söz konusu olacaktır. Tasarla aşamasının bazılarında verilen bir probleme yönelik çözüm istenirken, bazılarında uygula adımında gerçekleştirilen robotik sistemlerin verilen bir görevi yerine getirebilmesi için gerekli çalışma ortamının tasarlanması yer almaktadır.

Üret: Bu bölümde de öğrenciler aktif rol üstlenir. Eğitmen rehber pozisyonundadır. Eğitmen öğrencilere gereken yerlerde desteğini dengeli bir şekilde verecektir. Bu aşamada öğrencilerin *Tasarla* aşamasında ortaya koydukları tasarım planını kullanarak üretim gerçekleştirmeleri istenecektir.

Değerlendir: Değerlendirme aşamasında amaç, öğrenci başarısının notlandırılması değil, öğrencinin yukarıda belirtilen süreçlerden geçtikten sonra, öğrendiklerini ve deneyimlediklerini düşünmesi ve kritik etmesini sağlamaktır. Bu aşamada eğitmene büyük görev düşmektedir. Eğitmen, öğrencilerle interaktif bir şekilde durum değerlendirmesi yapmalı ve akabinde o haftaki eğitimin öğrencilere yaptığı katkı konusunda fikir sahibi olmalıdır. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Bu haftaki uygulamaları kendi cümlelerinizle tanımlayınız/ifade ediniz.
- Uygulama ve tasarımlarda ne tür yöntemler kullandınız? Neden?
- Tasarım ve üretim aşamalarında ne tür problemler yaşadınız ve bunların çözümü için neler yaptınız?
- Çözüm yöntemleriniz bu problemleri gidermekte başarılı oldu mu?
- Grup arkadaşınızla anlaşamadığınız noktalar oldu mu? Olduysa bunların üstesinden gelmek için neler yaptınız?

Öğrencilerden buradaki tüm sorulara cevap vermeleri beklenmemelidir. Öğrenciler, bunlardan, o haftaki uygulama türlerine bağlı olarak, uygun olanlarına cevap verebilirler. Eğitmenin tercihine bağlı olarak öğrencilerden yazılı olarak da cevaplar alınabilir. Eğitmen isterse, bu cevapları bir tartışma ortamında genel olarak alıp, durumla ilgili genel izlenim de elde edebilir.

İLERİ ROBOTİK SİSTEMLERE GİRİŞ

Robot sistemleri, özünde insansı mekanizmaların bir taklidi olarak düşünülse de günümüzde otonom kara, hava, deniz vb. araçlarını da içine alan geniş bir yelpazede yer almaktadır. Bu dersin amacı;

1. Robot sistemlerini ve özelliklerini tanıtmak,
2. Temel robotik mekanizmaları tanıtmak,
3. Çeşitli robotik sistemlerin geliştirilmesini sağlamak,
4. Öğrencilerin robotik sistemlere ilişkin düşünme, tasarlama, üretme ve programlama becerilerini geliştirmek,
5. Öğrencilerin sorumluluk alarak özgün tasarımlar gerçekleştirmelerini ve bu tasarımlar ile etik kurallara uyarak rekabetçi bir ortamda yarışmalarını sağlamaktır.

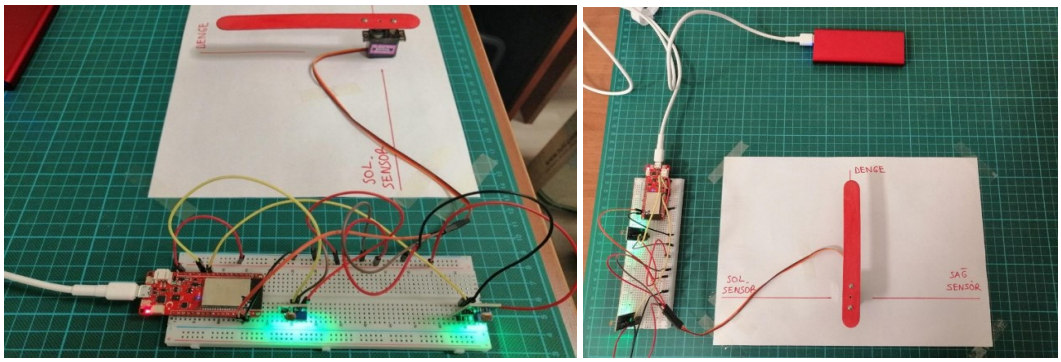
Bu amaçlara ulaşmak için aşağıdaki hedefler birer araç olarak kullanılacaktır.

1. Robot çeşitlerinin ve farklı robot yapılarının tanıtılması,
2. Temel robot öğelerinin, tahrik elemanlarının ve algılayıcıların çalışma prensiplerinin ve kullanım yöntemlerinin öğretilmesi,
3. Farklı serbestlik derecelerine sahip seri robotların üretilmesi, programlanması ve belli senaryolar çerçevesinde hareket ettirilmesi,
4. Öğrencilerin verilen problemlere yönelik özgün tasarımlar yapmasının sağlanması,
5. Bir yarışma konusu üzerinden, öğrencilerin sorumluluk alarak grup çalışması yapmalarının sağlanması ve bu sayede öğrencilerin özgün tasarım, planlama ve programlama kabiliyetlerinin geliştirilmesi.

İleri robotik dersi, farklı robot sistemlerinin tasarımı, entegrasyonu, kontrolü ve testleri ile yarışma konusuna yönelik özgün robot tasarımı olmak üzere iki ana aşamadan oluşmaktadır. Dersin birinci kısmında, robot sistemlerinde kullanılan sensörler, motorlar, sürücüler, kontrolcüler ve mekanik parçalar kullanılarak seri eklem yapısına sahip robotlar üretilecektir. Dersin ikinci kısmında ise, belirli sayıda üyeden oluşan gruplarla (2 kişilik gruplar önerilmektedir) yarışma konusuna yönelik tasarım, üretim, montaj ve test faaliyetleri yürütülecektir. Her iki aşamada da 3 boyutlu yazıcıdan üretilen parçalar kullanılacaktır. Her bir derste sistemin tasarım ve kurulumu, robot programlama detayları, robot kontrolü için gerekli ekipmanların tanıtımı ve entegrasyonu ile robotun testleri yapılacaktır. Uygulamalar sırasında öğrencilerin sadece yönergeleri takip etmesi değil, tasarım ve üretim aşamalarında kendi düşünce ve tecrübelerini kullanarak özgün ürünler ortaya koyması beklenmektedir. Gerek ders aşamalarında gerekse yarışma hazırlıklarında verilen kodların yanında öğrencilerin, kendi çıkarımlarından hareketle kodları geliştirmeleri istenmektedir. Eğitim ve yarışma aşamalarında, eğitmenlerin grupları seviye, yetkinlik ve kabiliyet bakımından uygun ve dengeli bir şekilde oluşturmaları beklenmektedir.

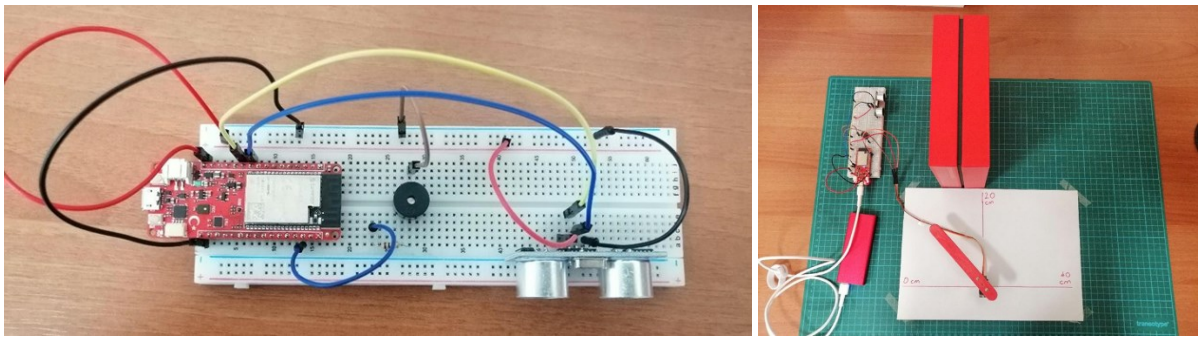
İleri robotik dersi kapsamında işlenecek konular hafta bazlı olarak aşağıda özetlenmiştir.

1. Hafta (Bölüm 1: Robot teknolojilerine örneklerle ve teorik seviyede giriş ve Bölüm 2: Işık sensörü ile servo motor kontrolü): Robot teknolojilerine örneklerle ve teorik seviyede bir giriş gerçekleştirilecektir. Robot kavramı, tarihçesi, robotların sahip olması gereken özellikler, bir robotu oluşturan birimler ve robot çeşitleri tanıtılacaktır. Sanayide ve günlük kullanımda karşılaşılan robotlar örnekler üzerinden incelenecektir. Dersin devamında tek bir servo motor üzerinden, robotlarda kullanılan farklı sensörlerin tanıtımı, kullanımı ve Deneyap Geliştirme Kartı'na entegrasyonu uygulanacaktır. Bu bölümde amaç, öğrencilerin ışık sensöründen alınan dijital ve analog sinyaller ile bir servo motoru hareket ettirebilmesini sağlamaktır.



Resim 2. İkinci hafta uygulamasından görüntüler

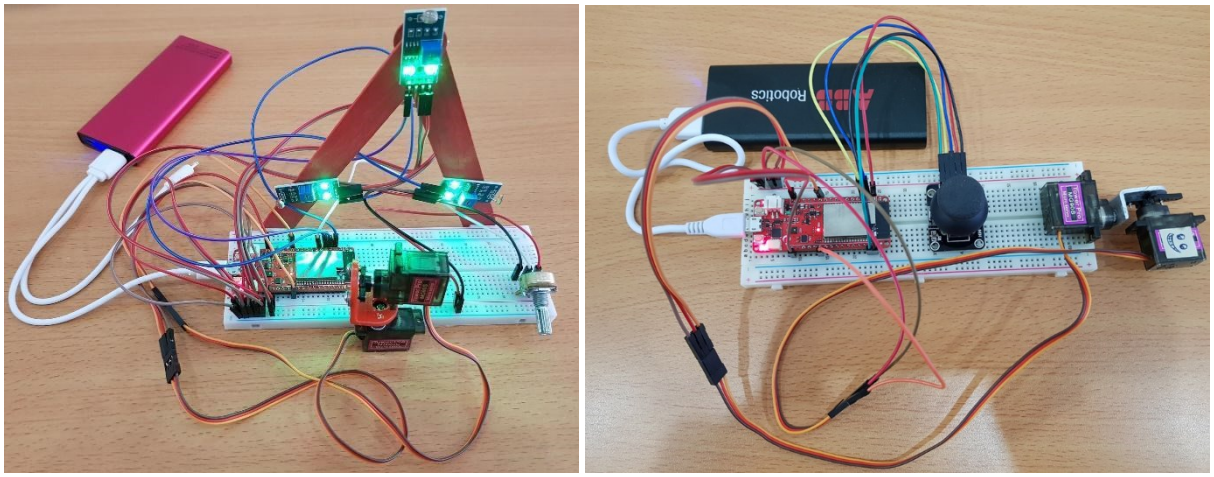
2. Hafta (Bölüm 3: Mesafe ve IMU sensörü ile servo motor ve DC motor kontrolü): Bu haftaki uygulamalarla öğrencilerin ultrasonik mesafe sensöründen ve Deneyap Geliştirme Kartı üzerinde bulunan 6 eksen IMU sensöründen alınan analog sinyaller ile bir servo ve DC motoru hareket ettirebilmesini ve bir buzzer üzerinden ses çıkarmasını sağlamak amaçlanmaktadır.



Resim 3. Üçüncü hafta uygulamalarından görüntüler

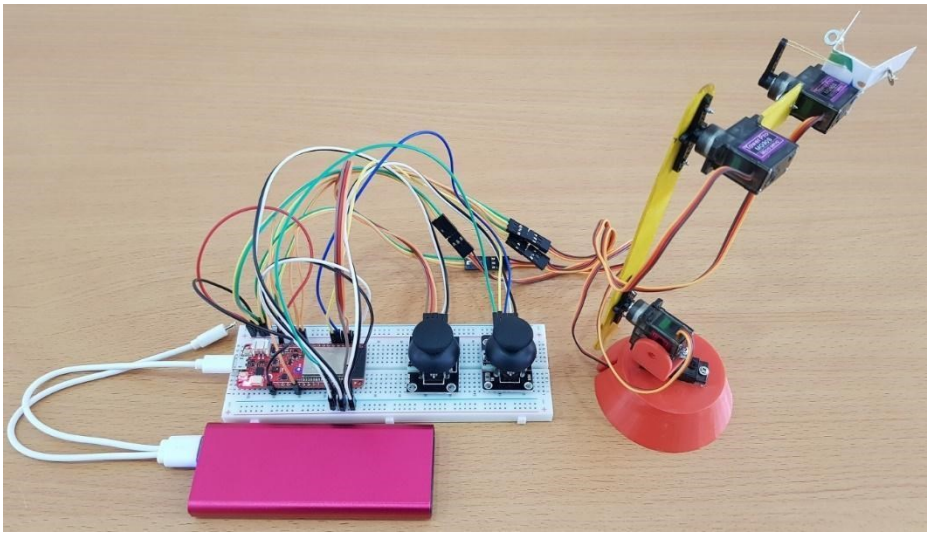
3. Hafta (Bölüm 4: İki eksenli robot kontrolü): Bu bölümde amaç, öğrencilerin Deneyap Geliştirme Kartı üzerinde kumanda kolu (joystick), ivme-ölçer ve ışık sensörlerinden alınan sinyaller ile servo motorları otomatik olarak kontrol etmeyi eğlenceli uygulamalar ile öğrenmelerini sağlamaktır. Öğrenciler Deneyap Geliştirme Kartı üzerinden kumanda kolu kurulumunu ve programlamasını, servo motor kurulumunu ve programlamasını, kumanda kolu ile servo motorları kontrol etmeyi,

IMU sensöründen ve ışık sensörlerinden alınan geri besleme sinyalleri ile servo motorları otomatik olarak kontrol etmeyi öğrenip deneyimleyeceklerdir.



Resim 4. Dördüncü hafta uygulamalarından görüntüler

4. ve 5. Hafta (Bölüm 5: Üç eksenli robot kontrolü): Bu bölümde amaç, öğrencilerin 3 serbestlik dereceli bir robot manipülatörün mekanik, elektronik ve yazılım çalışmalarını eğlenceli uygulamalar ile öğrenmelerini sağlamaktır. Öğrenciler, 3-serbestlik dereceli basit bir robot manipülatörün mekanik ve elektronik donanımını oluşturabilirler, Deneyap Geliştirme Kartı üzerinde 2 adet kumanda kolu bağlantı ve kullanımını öğrenip, bu robot manipülatör mekanizmasını 2 adet kumanda kolu aracılığıyla kontrol etmek ve bir görevi öğretebilmek amacıyla gerekli yazılımı Deneyap Geliştirme Kartı üzerinde hazırlayabilirler. Öğrenciler, bu 3 serbestlik dereceli robotun mekanik, elektronik ve yazılım hazırlıklarını tamamladıktan sonra bu sistemi kullanarak çeşitli robotik uygulamaları yaptırabilirler.



Resim 5. Üç serbestlik dereceli robotun monte edilmiş hali

6. Hafta (Bölüm 6: Beş eksenli robot kontrolü): Bu bölümde amaç, öğrencilerin 5 serbestlik dereceli bir robot manipülatörün mekanik, elektronik ve yazılım çalışmalarını eğlenceli uygulamalar ile öğrenmelerini sağlamaktır. Öğrenciler, 5-serbestlik dereceli basit bir robot manipülatörün mekanik ve elektronik donanımını oluşturabilirler, Deneyap Geliştirme Kartı

üzerinde 3 adet kumanda kolu bağlantı ve kullanımını öğrenip, bu robot manipülatör mekanizmasını 3 adet kumanda kolu aracılığıyla kontrol etmek ve bir görevi öğretebilmek amacıyla gerekli yazılımı Deneyap Geliştirme Kartı üzerinde hazırlayabilirler.

7. Hafta (Bölüm 7: Yarışma Hakkında Bilgiler ve Raporlama): Bu haftada, 12. haftada yapılması planlanan yarışma konusunun ilan edilmesi, teknik şartnamenin öğrencilerle paylaşılması, yarışmanın kısıtlarının ve kurallarının öğrencilerle paylaşılması gerçekleştirilecektir. Proje grupları oluşturulacak ve öğrencilerin olası tasarımlar üzerine tartışma yapmaları sağlanacaktır. Öğrencilerin belirtilen kural ve kısıtlamalara göre bir robot tasarımı yapmaları istenecektir. Bu aşamada Proje Tasarım Raporu'nun öğrenciler doldurulması sağlanacaktır. Tasarım raporları eğitmen tarafından gün sonunda toplanarak puanlandırılacaktır. Puanlandırma detayları Bölüm 7'de verilmiştir.

8. Hafta (Bölüm 8: Mekanik/Elektronik Tasarım ve Montaj): Bu hafta öğrenciler tasarladıkları robotların imalat ve montaj aşamalarına başlarlar. Eğitmen, Bölüm 8'de yer alan hususları öğrencilerle paylaşarak bu hususları dikkate almalarını sağlar.

9. Hafta (Bölüm 9: Sistem Entegrasyonu ve Yazılım): Bu hafta öğrenciler tasarlayıp ürettikleri robotların entegrasyonlarını tamamlayıp yazılımlarını oluşturmaya başlarlar. Eğitmen, Bölüm 9'da yer alan hususları öğrencilerle paylaşarak bu hususları dikkate almalarını sağlar.

10. Hafta: Bu hafta öğrenciler yarışma için tasarladıkları araçların sistem entegrasyonlarını bitirip sistem deneme ve testlerini yaparlar ve yarışmaya son hazırlıklarını tamamlarlar.

11. Hafta: Bu hafta öğrenciler ve eğitmenler yarışma sahasını, platformu hazırlar ve denemeler için hazır hale getirir. Geliştirdikleri robotları platform üzerinde deneyerek eksiklikleri tamamlarlar.

12. Hafta: Yarışma haftasında, öğrenciler önceden bilgilendirilerek, yarışma yönergesine göre, hazırlıklarını tamamlayıp, belirtilen sıralama dahilinde, araçlarının performanslarını sergilerler. Performans derecelerine göre yarışma dereceleri ilan edilir.

Dersin İşleyişi ile İlgili Önemli Hususlar

- Eğitmenlerin, derslerin gözle aşamasında verilen örnekleri çoğaltması, görsel ve videolar ile bu aşamanın zenginleştirmesi gerekmektedir. Sadece verilen içerik ile yetinilmemelidir.
- Her bir derste yapılacak uygulamalarda kullanılacak malzemeler derslerin giriş kısmında belirtilmiştir. Bunlar 1 grup için gerekli malzemelerdir. Gruptaki öğrenci sayısı eğitmen tarafından belirlenmekle birlikte, önerilen öğrenci sayısı 2'dir.
- Her bir hafta sonunda ilave etkinlikler verilmiştir. Haftalık konuları erken bitiren grupların bu ilave etkinlikleri yapmaları istenebilir.
- 4, 5 ve 6. bölümlerdeki uygulamalarda 3 boyutlu yazıcıda üretilmesi gereken parçalar bulunmaktadır. Bu parçaların üretimi zaman alacağından, ders gününden önce üretilmesi gerekmektedir. Önerilen uygulama, ders dönemi başlangıcında grup sayısına göre tüm baskı alınacak parçaların üretilerek hazır hale getirilmesidir.
- 2. bölümde yer alan Deneyap Geliştirme kartının bilgisayara bağlantısı ve gerekli programların ve kütüphanelerin yüklenmesi ile ilgili kısımlar daha önce Deneyap

Geliştirme Kartı'nı kullanmamış öğrenciler içindir. Önceki eğitimlerde bu kısmı uygulayan gruplarda Deneyap Geliştirme Kartı'nın tanıtıldığı kısımlar atlanarak LDR uygulamasına geçilmelidir.

Kaynakça

Can, Ş. (2005). *Öğretme-Öğrenmede İpuçları ve Pekiştireçlerin Rolü*. Muğla Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, (14), 97-109.

Çetin, İ., Üçgül, M., Top, E. & Yükseltürk, E. (2021). *Robotik ve Kodlama Lise*. TÜBİTAK Yayınları, Ankara.

Özkalp, E. (2005) *Davranış Bilimlerine Giriş.* Anadolu Üniversitesi Yayınları, ISBN 975-06-0095-9.

Organisation for Economic Co-operation and Development (OECD) (2018). The future of education and skills: Education 2030. https://www.oecd.org/education/2030/E2030%20Position%20Paper%20(05.04.2018).pdf

Üçgül, M., Çetin, İ., Yükseltürk, E. & Top, E. (2021). *Robotik ve Kodlama Ortaokul.* TÜBİTAK Yayınları, Ankara.

Papert, S. (1991). *Situating constructionism.* In I. Harel & S. Papert (Eds.), Constructionism (pp. 1–11). NJ: Ablex.

1. Bölüm: Robot Kavramı ve İleri Robot Teknolojilerine Giriş

Haftanın Kazanımları:

- Öğrenciler robot kavramını tanımlar, robot çeşitleri ile ilgili günlük hayattan örnekler verir.
- Öğrenciler robotiğin temel kavramlarını tanır ve kavrar.
- Öğrenciler ders kapsamında üretilecek 1, 2, 3 ve 5 serbestlik dereceli robotları tanır, inceler, değerlendirir ve günlük hayattaki uygulamalarını fark eder.
- Öğrenciler mekanizma ve robot arasındaki farkı tanımlar.
- Öğrenciler robotları farklı kategoriler altında sınıflandırır ve robot tiplerinden örnekler verir.
- Öğrenciler robotlarda serbestlik derecesi kavramını kavrar ve robotik bir sistemin serbestlik derecesini hesaplar.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin İleri Robotik dersinin kapsamı, derste gerçekleştirilecek çalışmalar, robot tipleri ve robotlarda serbestlik derecesi konularında bilgi sahibi olmalarıdır. Öğrenciler öncelikle mekanizma ve robot kavramını öğreneceklerdir. Aradaki farkı anladıktan sonra, robotların sınıflandırması üzerine değerlendirmeler yapılacak, eğitmenin yönlendirmesi ile kendi anlayışlarına göre sınıflandırma yapacak ve daha sonra literatürdeki sınıflandırmalar üzerine konuşulacaktır. Bu sınıflandırmalardaki farklı robotların örnekleri eğitmen tarafından gösterilecek ve öğrencilerle bunları değerlendirecektir. Ders kapsamında işlenecek robotların tanıtılması ve üzerine konuşulması ile ders sonlandırılacaktır.

1. ROBOT NEDİR?

Eğitmen, ilk haftaki dersin başlangıcında, öğrencilerin robotik ile ilgili kavramlar üzerinde tartışmalarını, aşağıdaki gibi sorulara öncelikle cevap vermelerini sağlamalıdır:

- Robot nedir? Robotlar hareketli olmak zorunda mıdır?
- İlk robotu kim geliştirmiştir? Robotların tarihsel gelişimi nasıl olmuştur?
- Robotların kullanıldıkları endüstriyel alanlar nelerdir?
- Robotların özellikleri nelerdir? Mekanik yapıları nasıldır?
- Hangi tür robotlar vardır?
- Robotlar nasıl programlanır?
- Günlük yaşamda karşılaştığınız robot örnekleri nelerdir?

Bu sorular eğitmen tarafından çeşitlendirilmelidir. Öğrencilerin verdiği cevaplara göre yönlendirmeler yapılabilir. Bu sorular ile öğrencilerin derse ilgileri, motivasyonları ve katılımları artırılmış olacaktır.

Eğitmen ayrıca bu ders içeriğinde verilmemiş, fakat faydalı olabileceğini düşündüğü çevrimiçi kaynaklardan temin edebileceği ilave görselleri ve videoları öğrenciler ile paylaşmalıdır. Öğrencilere sadece bu ders içeriğinde geçen paragrafları bire bir iletmek kesinlikle uygun değildir. Bu ilk derste eğitmene büyük görev düşmektedir.

Robotik alanındaki ilk çalışmalar sibernetik alanının kurucusu olarak kabul edilen Ebû'l İz İsmail İbni Rezzaz El Cezerî (d. 1136, Cizre, Şırnak; ö. 1206, Cizre) tarafından gerçekleştirilmiştir. Lakabını yaşadığı şehirden alan El Cezerî, öğrenimini Camia Medresesi'nde tamamlayarak, fizik ve mekanik alanlarında yoğunlaşmış ve pek çok ilke ve buluşa imza atmıştır. El Cezerî'nin yaptığı otomatik makineler günümüz robotik, mekanik, yapay zekâ ve sibernetik bilimlerinin temel taşlarını oluşturmaktadır. İlk robotu çalıştırdığı kabul edilen El Cezerî'nin Leonardo da Vinci'ye de ilham kaynağı olduğu düşünülmektedir.

Robot kelimesi ise ilk olarak 1921'de Çek oyun yazarı Karel Čapek tarafından "*Rossum'un Evrensel Robotları*" adlı bilim kurgu hikâyesinde geçmiştir. Çek dilinde "hizmetkâr" kelimesine karşılık gelen "robota" dan türetilmiştir. Sonrasında 1941 yılında Isaac Asimov "Robot" kelimesinden robot teknolojisiyle ilgili tüm alanları içeren bir tanım olarak kabul edilen "Robotik" kelimesini türetmiştir. Asimov, "Runaround" isimli hikayesinde robotların üç yasasını yazmıştır. Bunlar;

1- Bir robot bir insana zarar veremez ya da kayıtsız kalarak bir insanın zarar görmesine neden olamaz.

2- Birinci yasayla çatışmamak koşuluyla, bir robot insanlar tarafından verilen emirlere uymak zorundadır.

3- Birinci ve ikinci yasayla çatışmamak koşuluyla bir robot kendi varlığını korumalıdır.

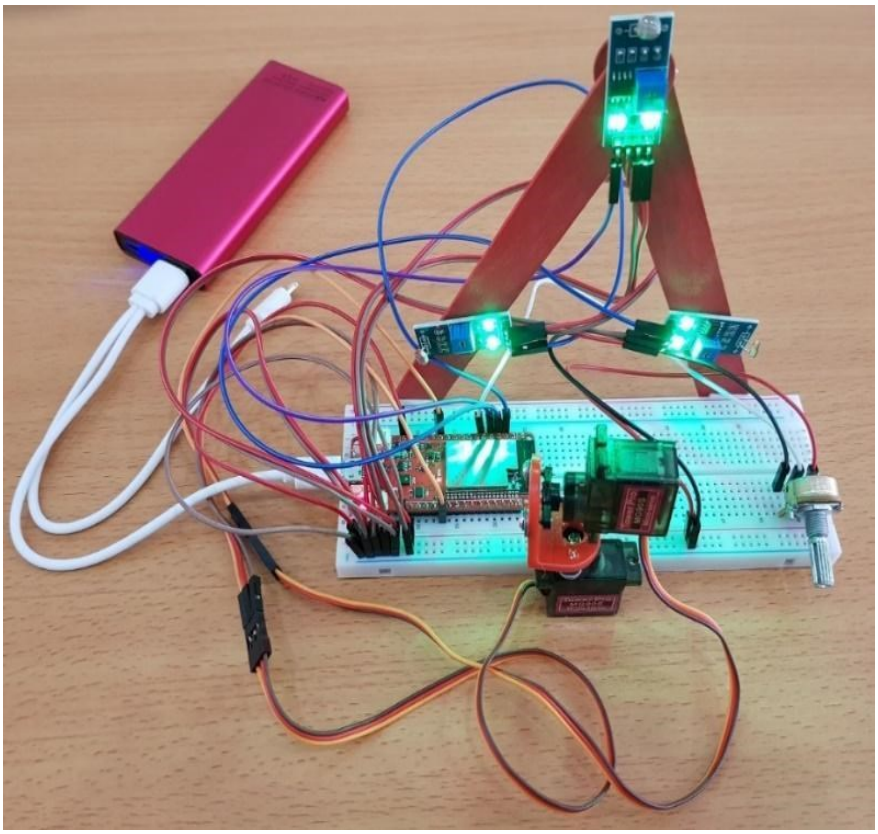
19. ve 20. yüzyıllarda elektronik ve bilgisayar teknolojilerinin hızla gelişmesi ile robotik sistemlerde de ciddi değişim ve gelişmeler yaşanmıştır. Robot tanımı da çeşitli otorite ve ülkeler tarafından farklı şekilde yapılmaktadır. Robotun yaygın bir şekilde kabul gören genel tanımı Amerikan Robot Enstitüsü tarafından yapılmıştır. Buna göre robot, "malzemelerin, parçaların ve araçların hareket ettirilebilmesi için tasarlanmış olan çok fonksiyonlu ve programlanabilir manipülatör veya farklı görevleri yerine getirebilmek için değişken programlı hareketleri gerçekleştirebilen özel araç" olarak tanımlanmaktadır. Bazı otoriteler robotların algılama ve karar verme yeteneklerine sahip olması gerektiğini savunurken, bazıları bunun elzem olmadığını, yeniden programlanabilen mekanik aksamların da robot sayılabileceğini belirtmişlerdir.

Genel olarak bir robotun yapısına bakıldığında mafsallarla bağlı katı cisimlerden (linkler) oluşan mekanik bir yapı, mekanik yapıya bağlı ve işlemleri gerçekleştiren uç işlevci, mekanizmaya hareket veren tahrik elemanları, kendi durumu ve çevresi hakkında bilgi toplamak için kullanılan sensörler ve tüm sistemi denetlemekten sorumlu kontrolcü içermektedir. Örnek olarak insan vücudu incelendiğinde alt ve üst uzuvlarımızı robotik bir mekanizmaya benzetebiliriz. Buna göre kol ve bacaklarımızı, bunların dönme noktalarını robotik bir sistemin hangi parçaları olarak tanımlayabiliriz? Bu uzuvlar arasında ne vardır? Kolumuzu hareket ettiren nedir? Hareket

komutları nereden gelir? Kolumuz veya bacağımızı hareket ettirirken hangi verileri kullanırız? Bu verilerin hassasiyetleri nedir?

2. DERS KAPSAMINDA GERÇEKLEŞTİRİLECEK ROBOTLAR

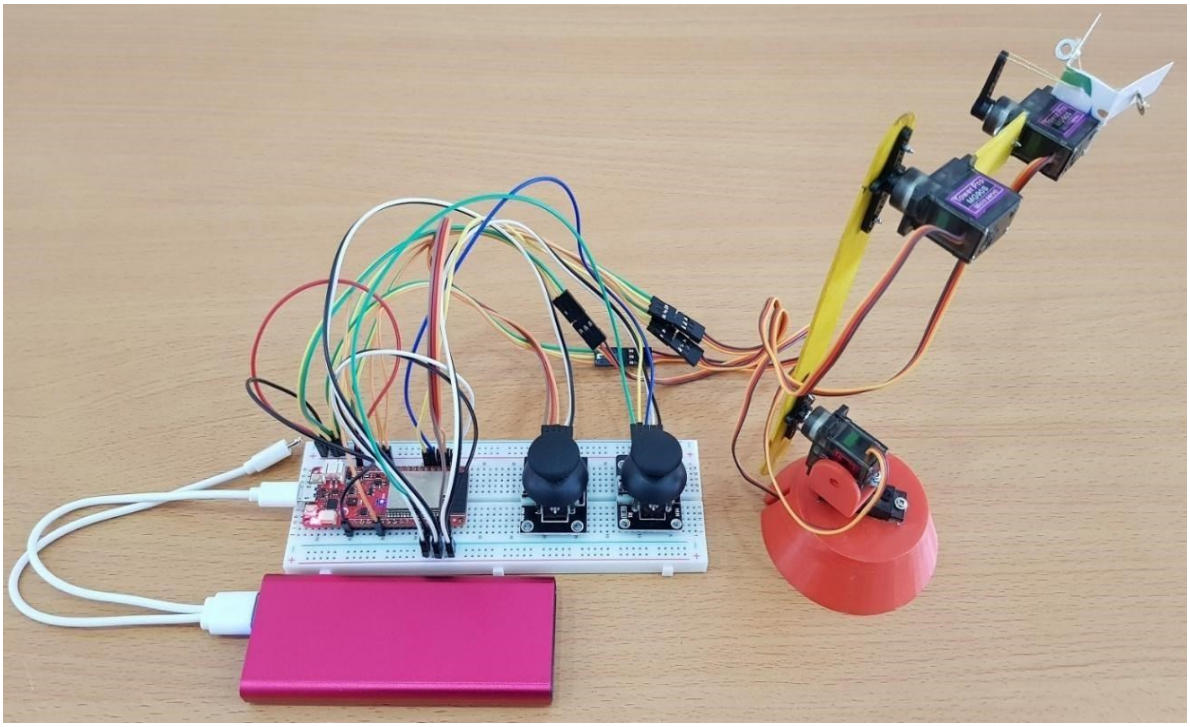
Ders kapsamında hem robotlar hem de robotların alt sistemleri üzerine çalışmalar yapılacaktır. Robotların alt sistemlerinin neler olduğu konusunda bir fikir alışverişi öğrenciler ve öğretmen arasında gerçekleştirilebilir. Paralel robotlara göre teknolojisi, tekniği, teorisi ve üretimi daha kolay olmasından dolayı genelde giriş seviyesindeki robot derslerinde seri robotlar tercih edilir. Seri veya paralel robot ayrımı, Bölüm 3.1'de detaylı olarak anlatılacaktır. Ders kapsamında, 2, 3 ve 5 serbestlik dereceli robotlar ile çalışılacak, programlanacak üretilecek ve entegre edilip çalıştırılacaktır. Temel olarak 2 eksenli robot üzerinde iki tane aktif motor bulunmaktadır. Yani, uzayda bir noktayı işaret etmek için bu robot kullanılabilir. Örneğin gökyüzünde bulunan bir uyduyu parmağınızla işaret etmek isterseniz, kolunuzu yukarı doğru kaldırmak ve uyduya doğru yatay dönmek üzere iki temel hareket yaparsınız. Burada iki serbestlik dereceli bir hareket yeterli olacaktır. Diğer bir örnek olarak havada bulunan bir insansız araç yerdeki bir noktayı gözlemek için yine iki temel hareket yapar. Bunlardan birisi dikey eksen etrafında dönme, diğeri de yatay eksen etrafında dönmedir. Resim 1.1' de böyle bir robot iki eksenli olarak gösterilmektedir ve İleri Robotik dersinde üretilecektir.



Resim 1.1. İki eksenli seri robot sistemi

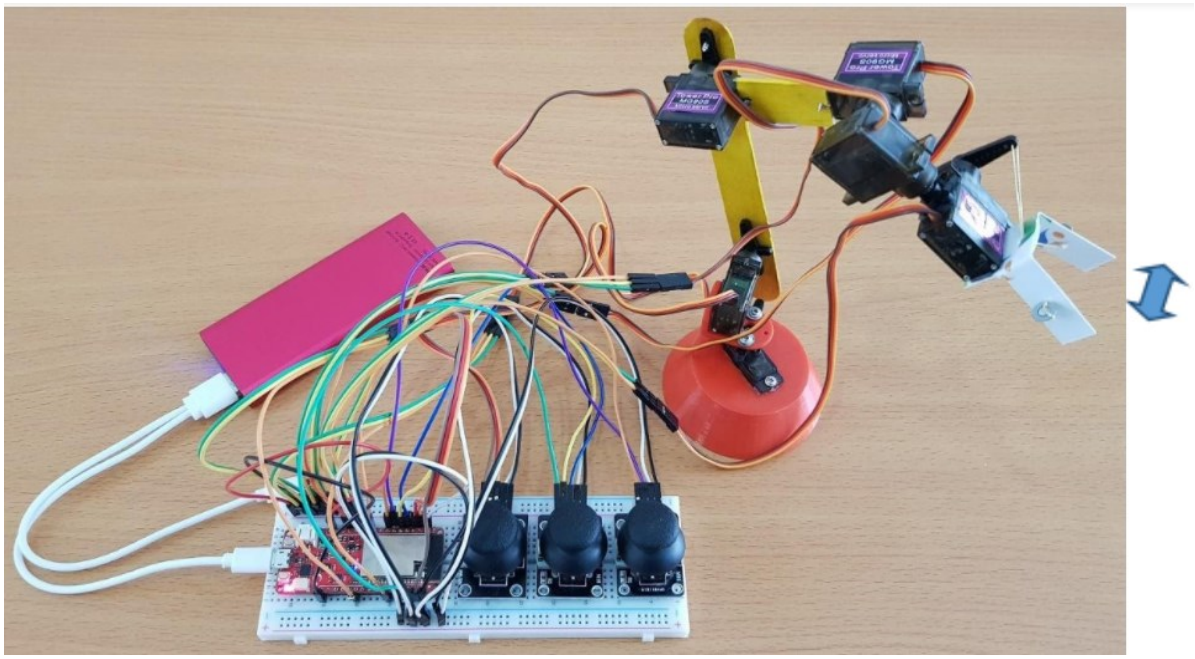
Üç eksenli robotlarda ise iki eksenli robottan farklı olarak bir aktif eksen daha bulunmaktadır. Üç eksenli robotlar hem kendi dikey ekseni etrafında dönebilmekte hem de omuz ve dirsek motorları ile bir kol gibi hareket edebilmektedir. Ders kapsamında geliştirilecek robotik sistem üzerinde

nesneleri tutmaya yarayan bir uç işlevci de bulunmaktadır. Basit bir el mekanizması olarak da düşünülebilir. Resim 1.2'de üretilecek olan robotik sistem görülmektedir.



Resim 1.2. Üç eksenli seri robot sistemi

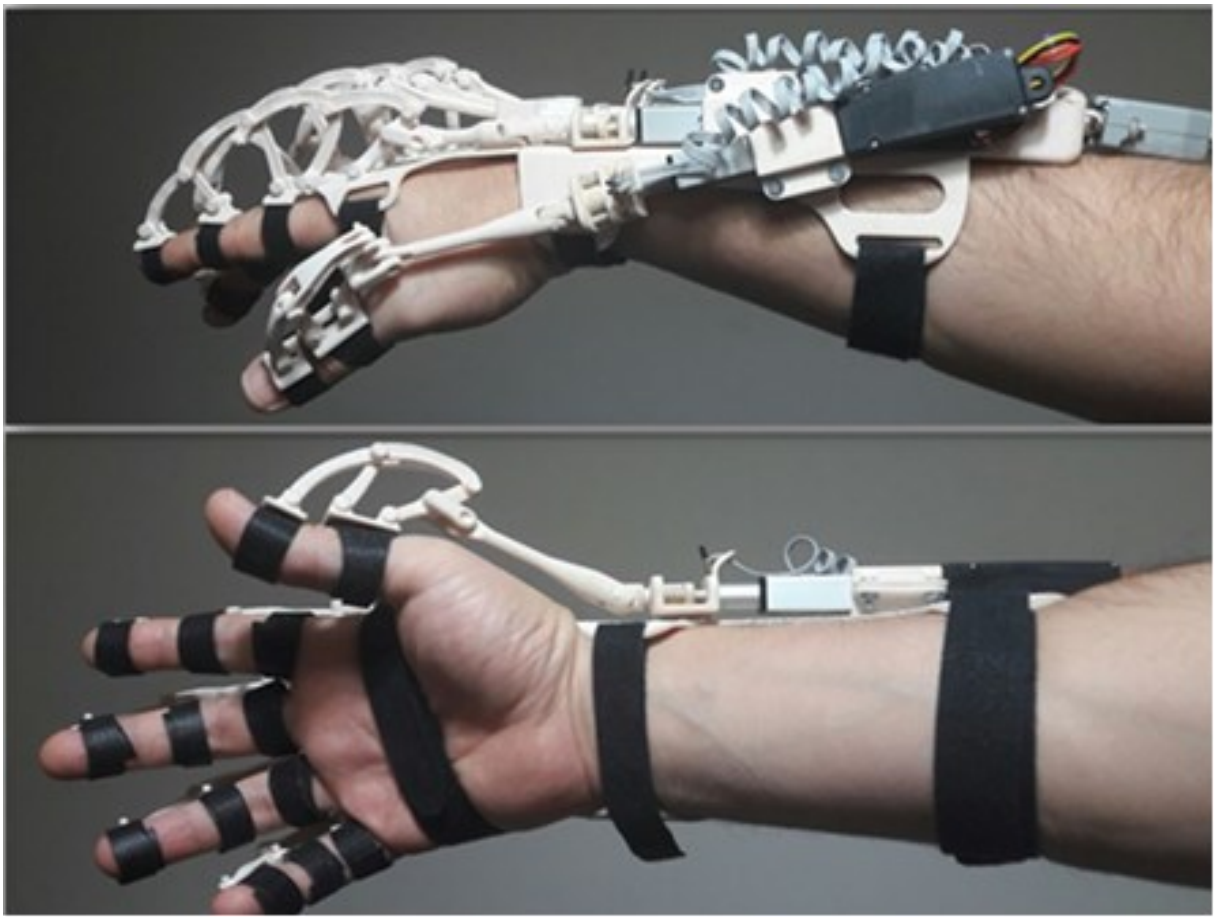
Ders kapsamında üretilecek olan son seri robot beş eksenli bir robottur. Üç eksenli robottan farklı olarak üzerinde bir de bilek motoru bulunmaktadır. Dikey eksen etrafında dönme, omuz, dirsek, bilek ve artı olarak tutma mekanizması olacaktır. Resim 1.3'te üretilecek olan robot sistemi görülmektedir.



Resim 1.3. Beş eksenli seri robot sistemi

3. MEKANİZMA VE ROBOT KAVRAMLARI

Her robot, içerisinde çeşitli mekanizmalar içerir. Fakat her mekanizma robot değildir. Örneğin, Resim 1.4'te bir el dış iskelet mekanizması görülmektedir. Bu karmaşık bir robot sistemidir ve alt mekanizmalar içermektedir. Önkol üzerine yerleştirilmiş doğrusal hareket üreten motorlar, her bir parmak üzerine yerleştirilmiş olan dört çubuk mekanizmalarını hareket ettirmekte, bu mekanizmalar da parmakları hareket ettirmektedir. Burada başparmağı hareket ettirmek için bir doğrusal motor, bir ara çubuk, bir dört çubuk mekanizması ve bir son çubuk kullanılmıştır. Tek motor ile parmak açılıp kapatılabilmektedir. Yine işaret parmağı için, bir motor, bir ara çubuk, her bir boğumda birer tane olmak üzere üç tane dört çubuk mekanizması ve bir son çubuk kullanılmıştır. Sonuç olarak, robot sistemleri alt mekanizmalardan, yani hareketi aktaran mekanik kol ve eklem gruplarından oluşmaktadır. Bu mekanik hareket grupları, mekanizmalar, motorlar ve yazılım yoluyla programlanabilir olduğunda artık bir robottan bahsediyoruz demektir.



Resim 1.4 Bir insan-makine arayüzü olarak el dış iskelet sistemi (Ahmadipour, 2018)

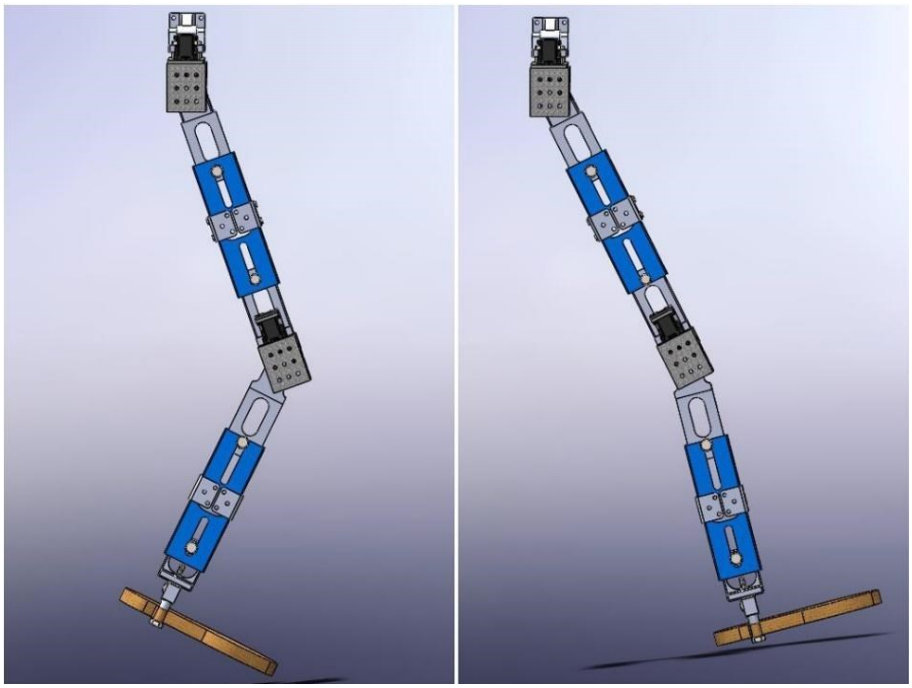
Vücudumuz da mükemmel bir robotik sistem olarak düşünülebilir. Parmaklar, omuz, kol, dirsek, bilek, diz, kalça, boyun gibi uzuvlar ve eklemler fiziksel olarak karmaşık, çok serbestlik dereceli bir robotik sistemdir. Bunların beyin tarafından kontrolü yoluyla vücudumuz çok karmaşık hareketleri yapabilmektedir.

3.1. Genel Robot Tipleri

Eğitmen burada aşağıdaki sorularla öğrencileri yönlendirebilir:

- Robotlar sizce nasıl sınıflandırılabilir? Robotları örneklerle sınıflandırmaya çalışınız.
- Seri ve paralel robot ne demektir?
- Tahrik tipine göre robotları nasıl sınıflandırırsınız? Elektromekanik, vb…
- Bağlantı tipine göre robotları nasıl sınıflandırırsınız? Dönel, vb.
- Kontrol tipine göre robotları nasıl sınıflandırırsınız? Noktadan noktaya, vb.
- Vücudunuzu düşündüğünüzde eklemleriniz nasıl, hangi tahrikle ve hangi eksenlerde hareket etmektedir?
- Bu robot tipleri hangi ihtiyaçlardan doğmuş olabilir? Örnekler veriniz.
- Farklı tipteki robotlarda geliştirilmiş veya geliştirmeye açık ne gibi özellikler olabilir?
- Teknolojik olarak gelişmeye açık ama şu ana kadar yapılamamış ne tip robotlar aklınıza geliyor?

Robotlar farklı şekillerde sınıflandırılmaktadır. Kullanım alanlarına göre endüstriyel robotlar, tıbbi robotlar, ev robotları, servis robotları, hobi robotları, uzay robotları, eğlence robotları vb. olarak sınıflandırılabilir. Bunun yanında tahrik tipine bağlı olarak hidrolik, elektrikli, pnömatik veya karma tahrikli robotlar bulunmaktadır. Burada en genel mânâda seri veya paralel olarak ayıracağız. Bir de bunların birleşimi olarak hibrit (melez) bir robot sisteminden bahsedebiliriz. Seri, kelime anlamı olarak uç uca sıralanmış veya bağlanmış bir yapıyı işaret etmektedir. Seri robot dediğimizde de robotun alt uzuvlarının seri bağlanmış halini düşünmeliyiz. Mesela kollarımızı veya bacaklarımızı düşünürsek bunlar seri mekanizmalardır. Neden? Çünkü kol, omuzdan itibaren üst kol, alt kol ve el olarak seri yani uç uca bağlanmış bir robot sistemidir. Yine bacaklar da kasıktan itibaren üst bacak ve alt bacak ve ayak olmak üzere seri bağlıdır (Resim 1.5).



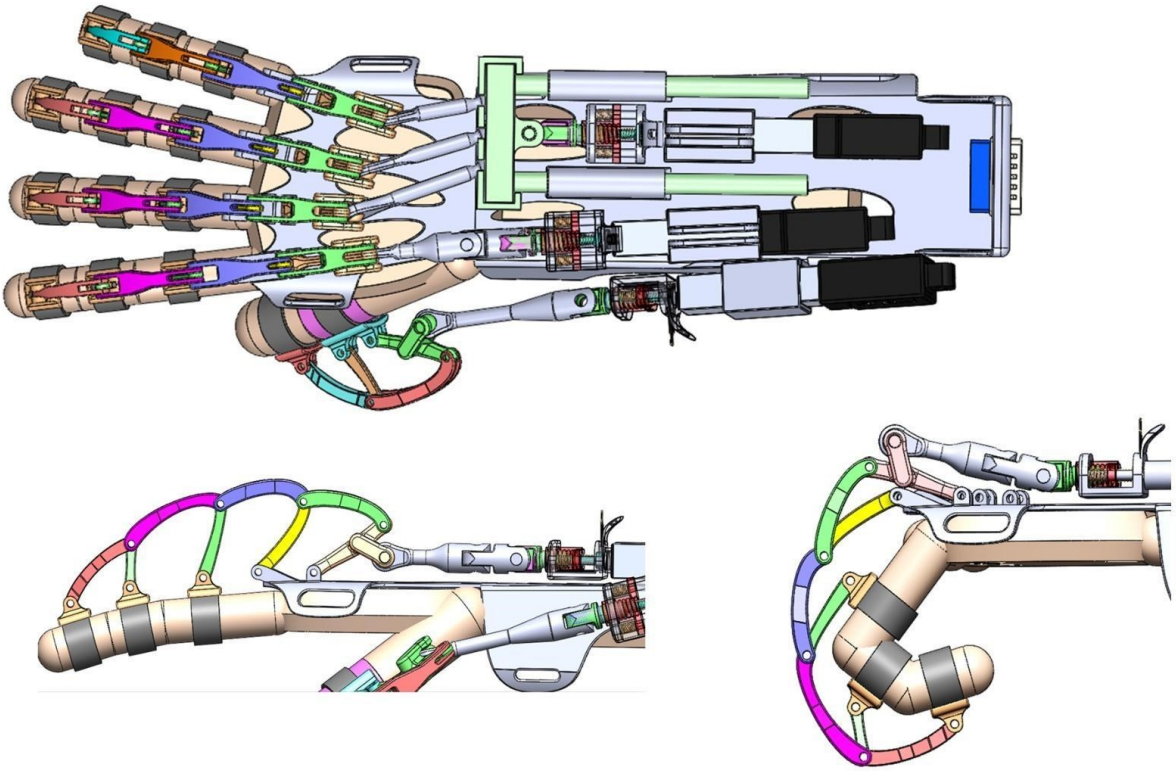
Resim 1.5. Seri robot örneği olarak bacak dış iskelet robotu katı modeli (Eken, 2018)

Paralel robot mekanizmaları bir diğer robot tipidir. Paralel kavramı, birbirinin yanında bulunan, üzerlerindeki noktaların birbirlerine eşit uzaklıkta olduğu doğru ya da düzlemleri ifade etmektedir. Örneğin iki kolumuzu ileri doğru uzattığımızda kollarımız birbirine paralel durumdadır. İki kolunuzla kocaman bir panda ayıcık taşıdığınızı düşünürseniz kollarınızı paralel kullanıyorsunuz demektir. Gözlerimizin görme ekseni de birbirine paraleldir. Ayakta hazır ol vaziyetinde dururken de bacaklarımız ve kollarımız birbirine göre paralel konumdadır. Bunun gibi farklı uzuvların paralel çalıştırılıp bir noktada birleştiği robotlar paralel robotlardır. Resim 1.6'da verilen uçuş simülatörünü inceleyelim. Simülatör üzerinde bulunan kullanıcı simülatörün altında bulunan altı bacak tarafından hareket ettiriliyor. Altı uzuv da paralel olarak konumlanıyor ve kabin üzerinde bulunan kişiyi 3 boyutlu uzayda sağa sola, yukarıya aşağıya, ileriye ve geriye doğrusal veya açısal hareket ettirmek için kullanılıyor. Buna benzer bir örnek de üst gövdemizin bacaklarımız tarafından taşınmasıdır. Üst vücudumuz iki bacağımızın oluşturduğu paralel bir yapı tarafından taşınmaktadır.



Resim 1.6 Bir paralel robot mekanizması (Ghorbani, 2018)

Bir diğer robot tipi ise melez (hibrit) robot sistemleridir. Bu tip robotlar üzerinde hem paralel hem de seri eklem yapıları bulunmaktadır. Resim 1.7'de bir dış iskelet el robotu verilmiştir. Bu sistem incelendiğinde birbirine paralel durumda olan parmaklar görülmektedir. Her bir parmak üzerinde de seri eklemler bulunmaktadır. El uzvu bütün olarak düşünüldüğünde üzerinde 2 ve 3 adet seri eklem bulunan beş tane birbirine paralel parmaktan oluşan melez bir robot sistemidir.



Resim 1.7 Melez (Seri + Paralel) bir robot sistemi katı modeli (Ahmadipour, 2018)

3.2. Serbestlik Derecesi

Robot sistemleri ve mekanizmalardaki en önemli kavramlardan birisi de serbestlik derecesidir. Serbestlik derecesi, bir robot veya mekanizmanın ne kadar sınırlı bir hareket kabiliyetine sahip olduğunun ve aynı zamanda ne kadar bağımsız tahriğe ihtiyaç duyduğunun bir göstergesidir. Örnek olarak bir sarkaç düşünüldüğünde bu sarkacın yaptığı şey nedir? Sallanmak. Bu sallanma hareketinin kontrolsüz olduğu düşünüldüğünde bu bir serbestlik dereceli pasif bir harekettir. Bir serbestlik derecelidir çünkü tek bir eksen etrafında sadece sağa ve sola salınmaktadır. Bu hareketi sadece bir açı değişkeni ile ifade edebiliriz. Sallanma hareketi serbest olarak gerçekleştiğinden (bir motor tarafından kontrol edilmediğinden) dolayı pasif hareket yapmaktadır. Bu sarkacın hareket eklemine bir motor bağlanarak sallanma açısı kontrol edildiğinde ise aktif hareket elde edilmiş olur. Sadece bir motor kullanıldığından bu sistem bir serbestlik dereceli olarak ifade edilmektedir.

Buna benzer olarak önceki bölümlerde tanıtılan mekanizma ve robot sistemleri de düşünüldüğünde, bunların da hareket kabiliyetleri sınırlıdır. Örneğin Resim 1.1'deki sistemi düşünürsek, iki motor, iki açı hareketi yapmaktadır. Yani iki serbestlik derecelidir. Elin sadece

işaret parmağının ucu düşünülürse, elin sabit tutulduğu bir durumda sadece işaret parmağının ucunun hareket ettirilebilmesi için parmak üzerindeki üç boğumun da hareket ettirilmesi gerekmektedir. Her bir boğum, parmağa dik bir eksen etrafında, sadece dönme hareketi yapmaktadır. İşaret parmağının ele bağlantı noktasındaki boğuma bakıldığında ise hem sağa sola hem yukarı aşağı hareket ettiği görülür. Bu durumda işaret parmağı dört serbestlik derecelidir. Resim 1.7'deki sistemin sadece işaret parmağına bakılacak olursa, bu sistemin işaret parmağını dışarıdan hareket ettirmeye yarayan (hareket bozukluğu olan, rehabilitasyona muhtaç hastalar için) bir mekanizma olduğu görülür. Burada işaret parmağının bükülme hareketi sadece bir motor tarafından gerçekleştirilmekte ve parmağın nesneleri kavraması sağlanmaktadır. İşaret parmağının toplamda dört serbestlik derecesi vardır fakat buradaki uygulamada 4 eklem ayrı ayrı kontrol edilmemekte, tek bir motor ile bükülme hareketi yaptırılmaktadır. Bu da bir serbestlik derecesine karşılık gelmektedir. Buradaki yapı ile işaret parmağının sadece bir eklemi kontrol edilmek istenirse, dört serbestlik derecesi elde edilmeyecek ve sadece bir serbestlik dereceli bir mekanizma elde edilmiş olacaktır.

4. DEĞERLENDİR

Bu hafta yapılan derste temel kavramlar ve genel robot sistemleri konusunda bilgi verilmiştir. Robot tipleri, serbestlik dereceleri konusunda örnekler üzerinden paylaşımlarda bulunulmuştur. Dersin sonunda öğrencilerle bu hafta yapılan uygulamalar hakkında konuşulur ve aşağıdaki sorular üzerinden tartışmalar gerçekleştirilir. Kısa sorulardan oluşan bir anket veya sınavla öğrencilerin bilgi seviyesi ölçülüp beraber değerlendirilebilir

- Bir arabanın motorundan tekerlere kadar giden sistemi düşündüğümüzde ne tür mekanizmalar vardır? Bunlardan herhangi biri robot olarak düşünülebilir mi?
- Bilekten itibaren bir eli düşündüğünüzde kaç serbestlik derecesi vardır?
- Uzayda ve bir yüzey üzerinde bir plaka düşünürseniz, buna kaç türlü hareket yaptırabilirsiniz? Buna göre uzayda ve bir yüzey üzerinde bir plakanın serbestlik derecesi (plakanın bir referans noktasına göre konumunu tarif etmek için gerekli değişken sayısı) nedir?
- Bir kuşun kanadındaki temel hareket düşünüldüğünde bu yapı kaç serbestlik derecelidir?
- İnsanların doğduğu ve öldüğü zaman vücutlarındaki kemik sayısı neden farklıdır? Bu neye işaret eder?
- İnsanlara uygulanan rehabilitasyon hareketlerinin serbestlik derecesi ile bir ilgisi var mıdır? Örnekler veriniz?

5. İLAVE ETKİNLİK

Bir güneş enerjisi santralinde, güneş panellerinin güneşi takip etmesi için nasıl bir sistem tasarlanması gerekir? Bu sistemi çizerek tasarlayınız ve gerekli serbestlik derecesini hesaplayınız.

Eğitmenler, robotlarla ilgili aşağıda verilen linklerdeki videolar üzerinden öğrencilerle beyin fırtınası yaparak konu hakkında bilgi seviyelerini artırabilir. Eğitmenler farklı videolar ve görseller ile bu kısmı zenginleştirmelidir.

İnsansı robot: https://www.youtube.com/watch?v=uhND7Mvp3f4

Böcek robotlar: https://www.youtube.com/watch?v=Fq8KB1CyA8g

Amfibik robotlar: https://www.youtube.com/watch?v=CkZszsl2aIU

Uçan robotlar: https://www.youtube.com/watch?v=zMi5v2KznU4

Paralel robotlar: https://www.youtube.com/watch?v=Ni4WkEA49qE&t=77s

Kaynakça

Ahmadipour, N. M. (2018). *Developing a haptic hand ınterface with force feedback from virtual environment and controlling it via electromyography signals*. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi.

Eken, H. & Vural, B. (2018). *Designing an assistive lower limb exoskeleton mechanism*. Lisans Bitirme Çalışması Raporu, Yıldız Teknik Üniversitesi.

Ghorbani, L. & Ömürlü, V.E. (2018). Forward kinematics of a 6x6 UPU parallel mechanism by ANFIS method. 2018 6th International Conference on Control Engineering & Information Technology (CEIT), pp. 1-6.

2. Bölüm: Işık Sensörleri ile Servo Motor Kontrolü

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.

Haftanın Kazanımları:

- Öğrenciler Deneyap Geliştirme Kartı'nın temel özelliklerini kavrar.
- Öğrenciler analog ve dijital sinyalleri kavrar. Bunlarla ilgili örnekler verir.
- Öğrenciler Deneyap Geliştirme Kartı'nın kurulumunu yapar ve gerekli kütüphaneleri yükler.
- Öğrenciler Deneyap Geliştirme Kartı üzerinden servo motor çalıştırır.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde ışık sensörü çalıştırır.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin ışık sensöründen alınan dijital ve analog sinyaller ile bir servo motoru hareket ettirebilmesini sağlamaktır.

Kullanılacak Malzemeler:

1. Deneyap Geliştirme Kartı, 1 adet
2. LDR sensörü, 2 adet
3. Servo Motor, 1 adet
4. Breadboard, 1 adet
5. Farklı renklerde erkek-erkek ve erkek-dişi jumper kabloları, 20+ adet
6. Mikro USB kablosu, 1 adet
7. Harici 5V USB Batarya, 1 adet
8. Çift taraflı silikon bant ve makas
9. Abeslang, 1 adet
10. El feneri (cep telefonunun feneri de kullanılabilir), 1 adet

Haftanın İşlenişi:

Gözle: Deneyap Geliştirme Kartı'na ilgili eleman bağlantılarının yapılması ve kodların gösterilmesi

Uygula: Yönergeye uygun şekilde tüm sensör bağlantılarının yapılması ve kodların yazılması

Tasarla: Sıcaklık sensörü ile ateş ölçüm cihazı tasarımı

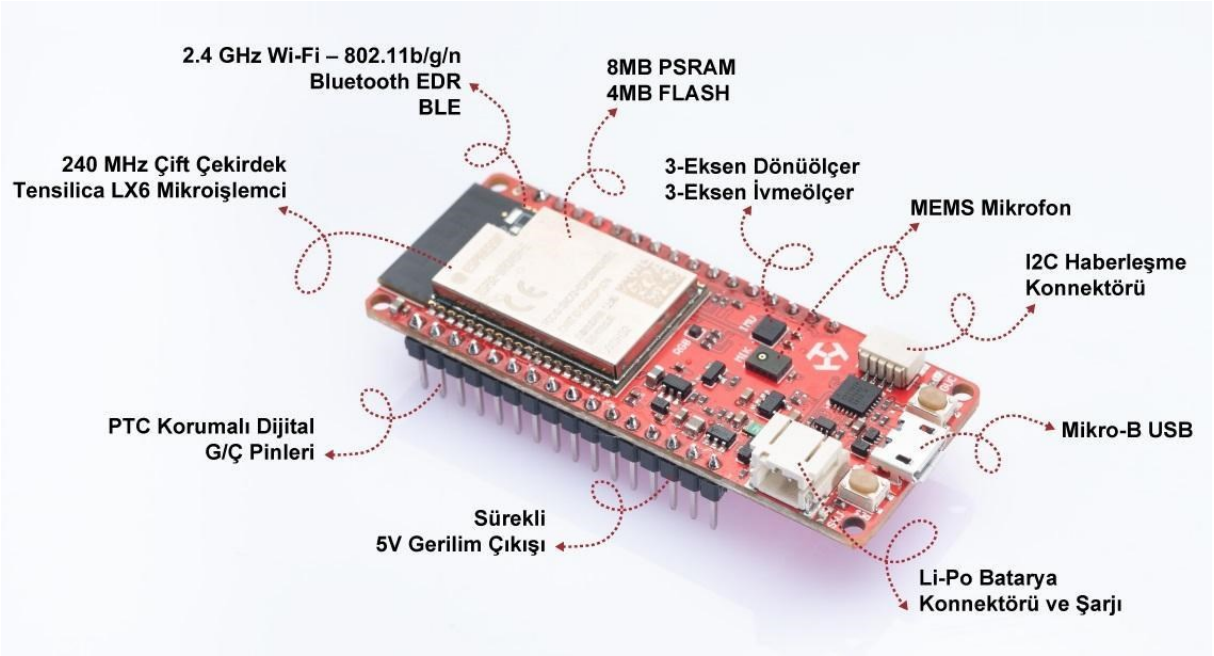
Üret: Tasarlanan ateş ölçüm cihazının üretilmesi

Değerlendir: Deney sonuçlarının ve yapılabilecek farklı uygulamaların tartışılması

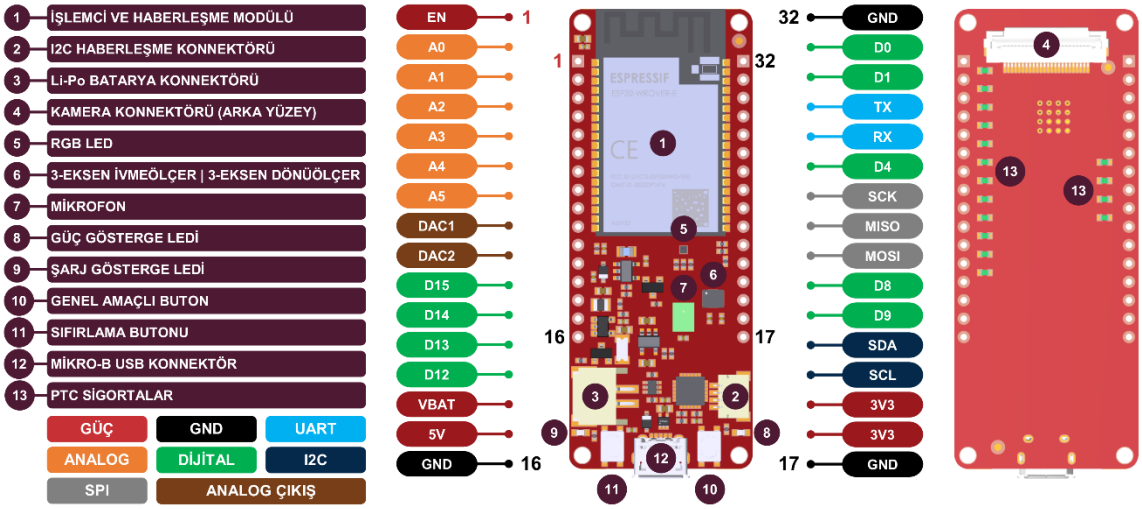
1. GÖZLE VE UYGULA

1.1. Gözle: Deneyap Geliştirme Kartı

Deneyap geliştirme kartı ESP32 tabanlı işlemciye sahip, üzerinde analog ve dijital giriş çıkış pinleri bulunan, I2C, Wi-Fi ve Bluetooth haberleşme modüllerine sahip bir geliştirme kartıdır. Kart üzerinde dâhili bir mikrofon, ivmeölçer ve RGB LED bulunmaktadır. Resim 2.1'de Deneyap Geliştirme Kartı'nın genel görünümü ve sahip olduğu donanımlar görülmektedir. Resim 2.2'de ise Deneyap Geliştirme Kartı'nın giriş çıkış pinleri verilmiştir.



Resim 2.1. Deneyap Geliştirme Kartı genel görünümü ve donanımlar



Resim 2.2. Deneyap Geliştirme Kartı pin bağlantıları

Kırmızı ile gösterilen pinler güç giriş ve çıkışlarını ifade etmektedir. GND olarak belirtilen siyah portlar topraklama amacıyla kullanılmaktadır. Kahverengi portlar (DAC1 ve DAC2) analog çıkışlardır. Turuncu portlar analog girişler, yeşil portlar ise dijital giriş ve çıkışlardır. Dijital portların giriş veya çıkış ayarı kod tanımlamaları üzerinden yapılmaktadır. Mavi ile gösterilen TX ve RX pinleri UART haberleşme için kullanılan alıcı ve verici portlarıdır. Lacivert ile gösterilen SDA ve SCL pinleri I2C haberleşmesi için kullanılmaktadır. Gri ile gösterilen pinler ise SPI haberleşme için kullanılmaktadır.

1.2. Uygula: Deneyap Geliştirme Kartı Arduino IDE Kütüphanesinin Yüklenmesi

Deneyap Geliştirme Kartı'nın Arduino IDE arayüzü üzerinde kullanılabilmesi için çeşitli yükleme işlemleri yapılmalıdır.

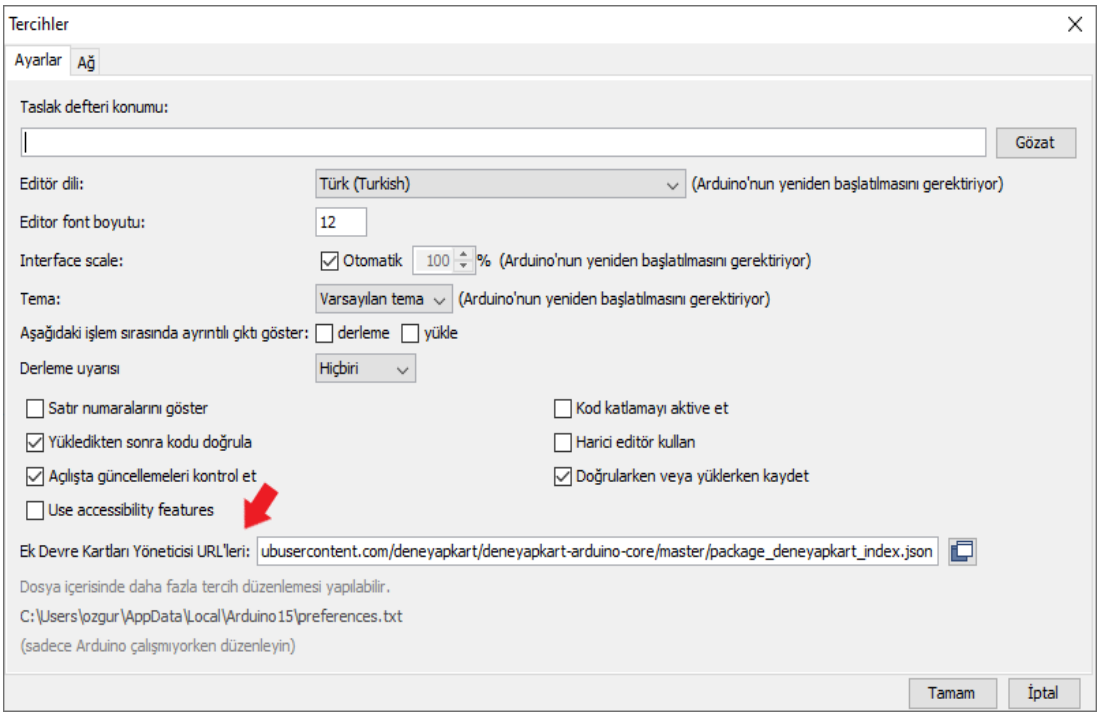
- Deneyap Geliştirme Kartı, Arduino IDE ile uyumlu olarak çalışmaktadır. Arduino IDE'nin en güncel versiyonu işletim sisteminize uygun olarak seçilerek yüklenir.



Resim 2.3. Arduino IDE yükleme ekranı

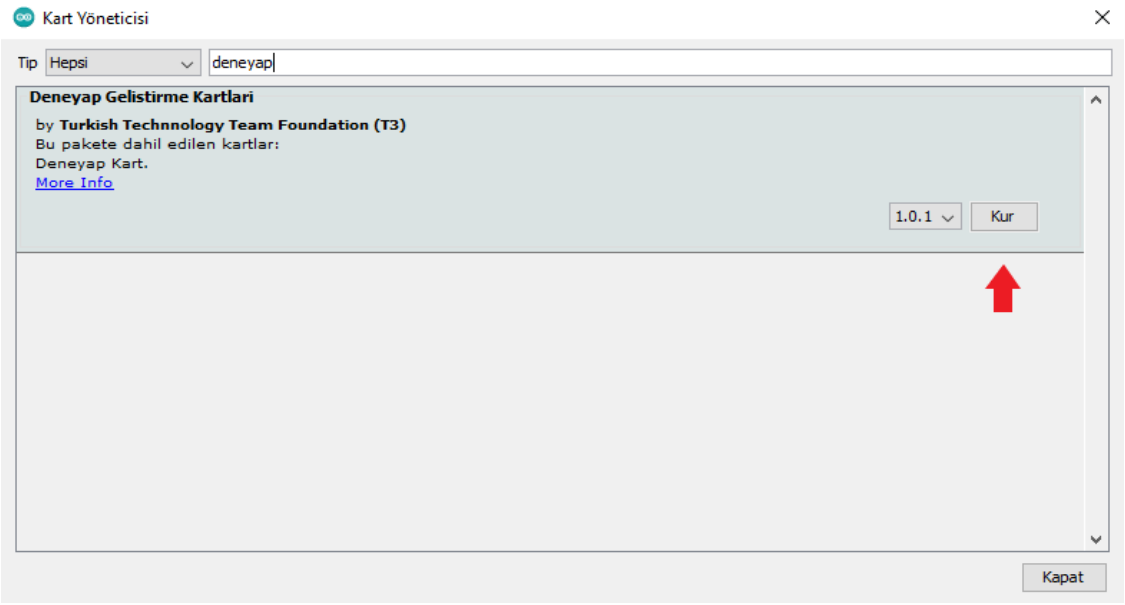
- Kurulumdan sonra Arduino IDE çalıştırılır. Dosya (File) –> Tercihler (Preferences) adımını takip ederek açılan pencerede Ayarlar (Settings) sekmesinde bulunan Ek Devre Kartları Yöneticisi URL'leri (Additional Boards Manager URLs) kısmına, aşağıda paylaşılan JSON dosyasına ait adresi kopyalayınız.

- **HTTPS://RAW.GİTHUBUSERCONTENT.COM/DENEYAPKART/DENEYAPKART-ARDUİNO-CORE/MASTER/PACKAGE_DENEYAPKART_İNDEX.JSON**



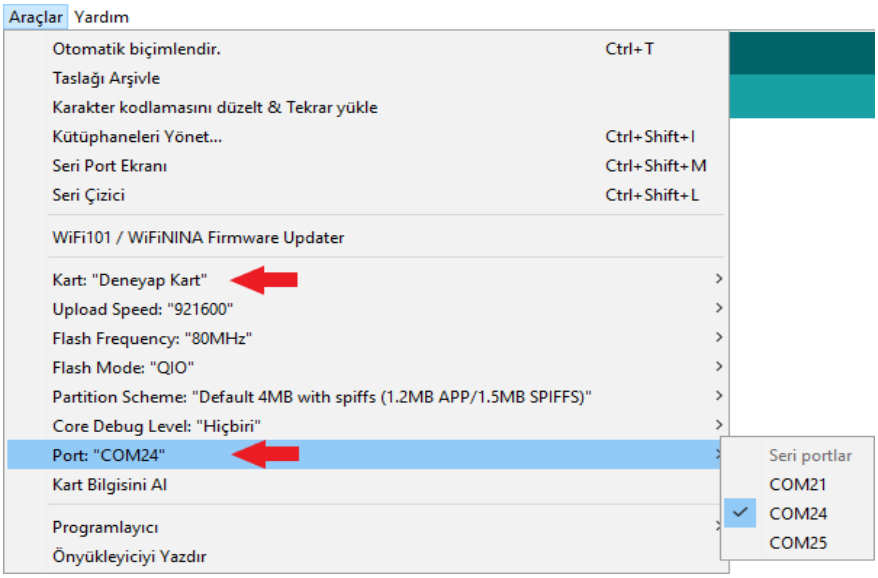
Resim 2.4. URL adresinin girildiği ekran

- Ana ekranda **"Araçlar"** sekmesinde Kart başlığı seçilerek **"Kart Yöneticisi"**ne tıklanır. Çıkan ekranda en üstteki arama satırına **"Deneyap Geliştirme Kartları"** yazılır ve karta ilişkin kütüphane **"Kur"** butonuna basılarak yüklenir. Yükleme tamamlandıktan sonra pencere kapatılır.

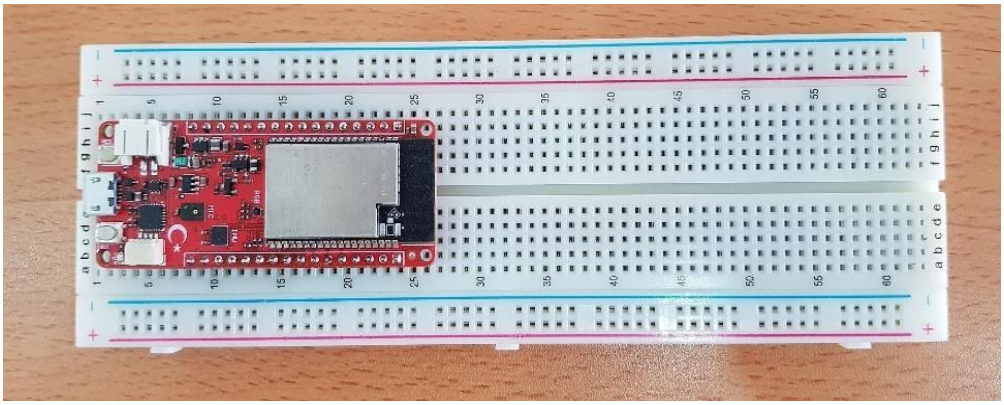


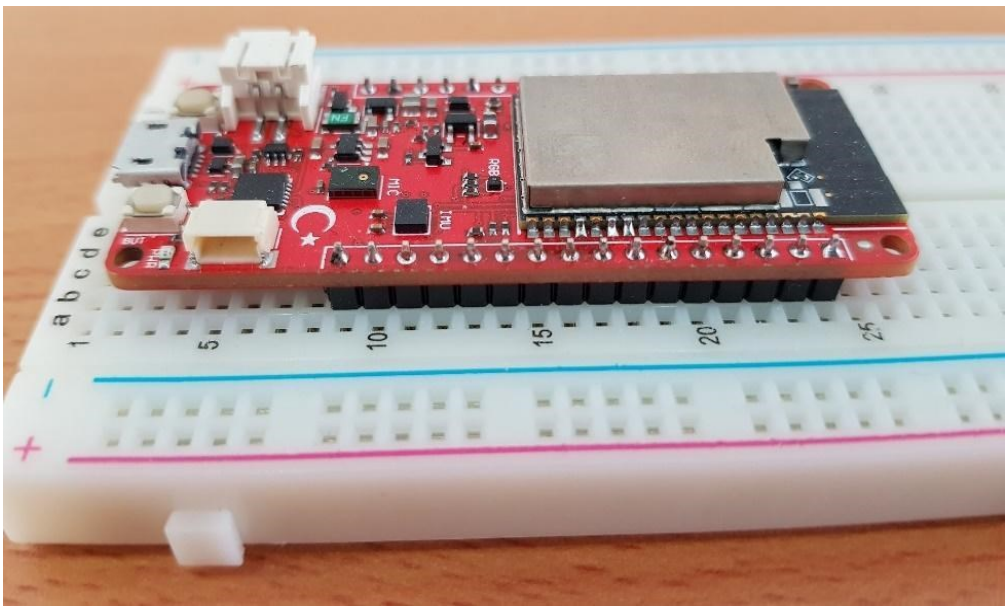
Resim 2.5. Deneyap Geliştirme Kartı kütüphanesi yükleme ekranı

- Ana ekranda yine **"Araçlar"** sekmesi altında **"Kart"** başlığı içerisinde **"Deneyap Geliştirme Kartları"** altında **"Deneyap Geliştirme Kart"** seçilir. "**Port**" başlığı içerisinden de kartın bağlı olduğu port numarası seçilir.



Resim 2.6. Kart ve port seçim ekranı





Resim 2.7. Breadboard üzerinde Deneyap Geliştirme Kartı



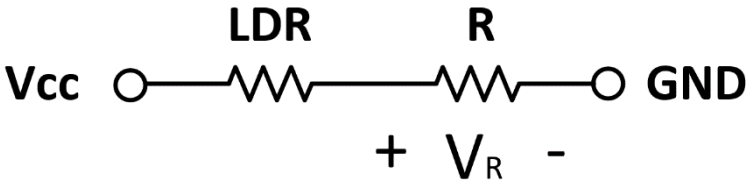
Resim 2.8. Deneyap Geliştirme Kartı'nın bilgisayar bağlantısı

1.3. Gözle: LDR (Light Dependent Resistor - Işığa Duyarlı Direnç) Nedir?

Bir robotun sahip olduğu en önemli özelliklerden birisi çevresini algılayabilmesidir. Dış ortamda bulunan ısı, ışık, ses, mesafe, nem, eğim, kuvvet gibi büyüklükler sensörler ile algılanarak gerçekleştirilmesi amaçlanan işe uygun olarak kullanılırlar. Ortamdaki ışık miktarının algılanması için LDR'ler kullanılmaktadır. LDR'lerin bazı kullanım alanları aşağıda verilmiştir;

- Gece yanıp, sabah sönen sokak ve bahçe lambaları,
- Otomatik olarak güneşe yönelen güneş enerjisi panelleri,
- Işığa yönelen robot uygulamaları.

Ortamdaki ışık şiddetinin algılanabilmesi için ışık miktarına bağlı olarak değeri değişen direnç kullanılmaktadır. LDR'nin yapısına göre üzerine düşen ışık miktarı arttıkça direnç değeri düşmekte veya artmaktadır. Bu uygulamada kullanılacak olan LDR'nin değeri ortamdaki ışık miktarı arttıkça düşmektedir. Resim 2.9'da LDR'li bir gerilim bölücü devresi görülmektedir. Burada V_{cc} pozitif besleme gerilimi, R direnç değeri, V_R direnç üzerinde düşen gerilim değeri ve GND topraklama girişidir.



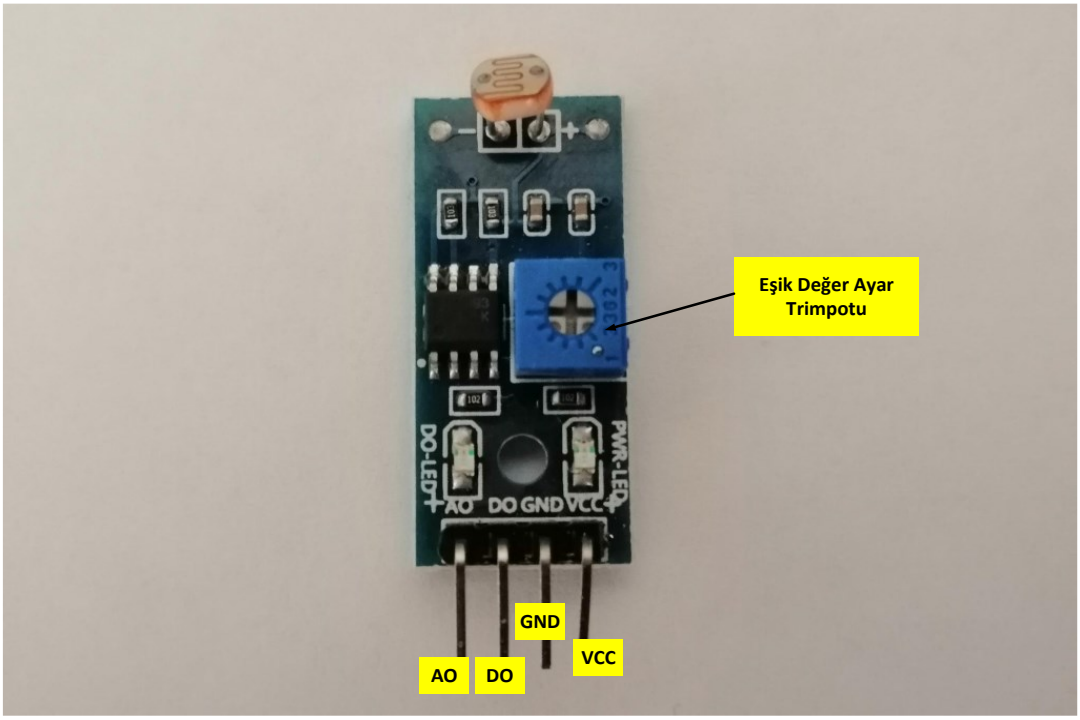
Resim 2.9. LDR'li gerilim bölücü devresi

Bu devrede ortamdaki ışık miktarına bağlı olarak LDR değeri değiştikçe R direnci üzerine düşen gerilim de değişmektedir. Böylelikle ortamdaki ışık şiddetine bağlı olarak değişen bir gerilim elde edilmektedir. Bu gerilim değeri aşağıdaki denklem ile tespit edilir.

$$V_R = \frac{V_{cc}R}{LDR + R}$$

Burada LDR'nin değeri arttıkça (ortamdaki ışık azaldıkça) paydanın değeri büyüyecek ve V_R değeri düşecektir. Bu analog gerilim değeri mikroişlemci üzerinde kullanılabilir.

Uygulamada kullanılacak olan LDR sensörünün yapısı Resim 2.10'da görülmektedir. Sensör üzerinde 4 adet pin bulunmaktadır. Bunlardan V_{cc} pinine sensörün besleme gerilimi uygulanmaktadır. Normalde 5V olarak belirtilen besleme gerilimi, DeneYap kartına ile kullanımına özgü olarak 3.3V olmalıdır, çünkü DeneYap kartın lojik pinleri 5V'a töleranslı değildir. GND pinine işlemcinin topraklama pini bağlanır. DO pini dijital çıkış vermektedir. Ortamdaki ışık şiddeti belli bir eşik değerinin üzerine çıktığında HIGH, eşik değerin altına düştüğünde ise LOW değerini almaktadır. Bu eşik değer sensör üzerinde bulunan trimpot ile ayarlanmaktadır. AO pini ise ortamdaki ışık şiddetini analog olarak vermektedir. Bu yukarıda bahsedilen direnç değişimlerine göre sağlanmaktadır. AO çıkışındaki gerilim değerine göre ortamdaki ışık miktarı tespit edilmektedir. Sensöre enerji verildiğinde PWR-LED ışığı yanmaktadır. DO çıkışı HIGH olduğunda ise DO-LED ışığı yanmaktadır.



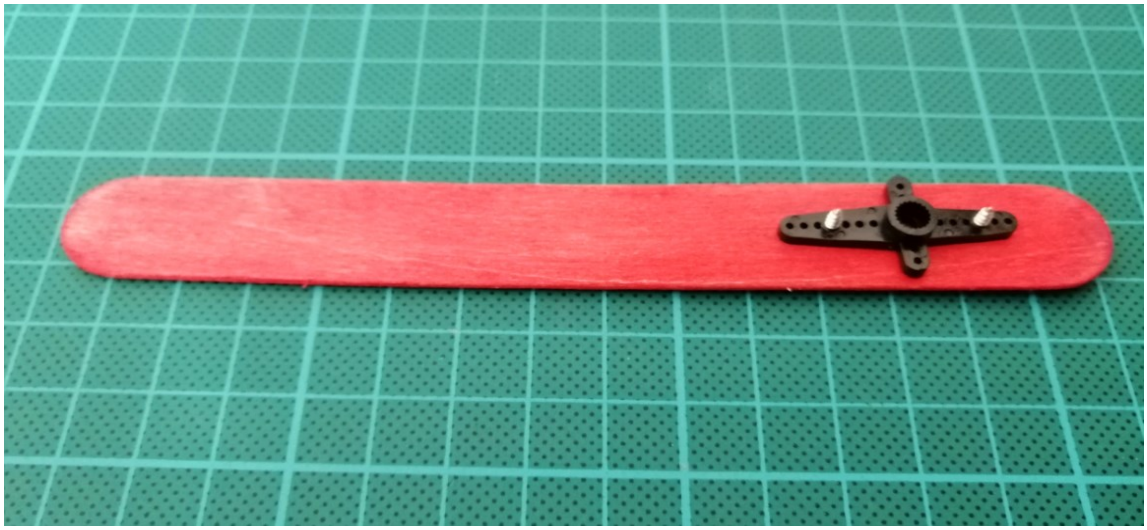
Resim 2.10. Uygulamada kullanılacak LDR sensörü

1.4. Uygula: LDR Sensörünün Dijital Çıkışı ile Servo Motor Kontrolü

Bu uygulamada 2 adet LDR sensörü üzerinde bulunan dijital çıkışlar ile bir servo motorun hareketi sağlanacaktır. Servo motor üzerine vidalanacak olan bir abeslang gösterge ibresi olarak kullanılacaktır.

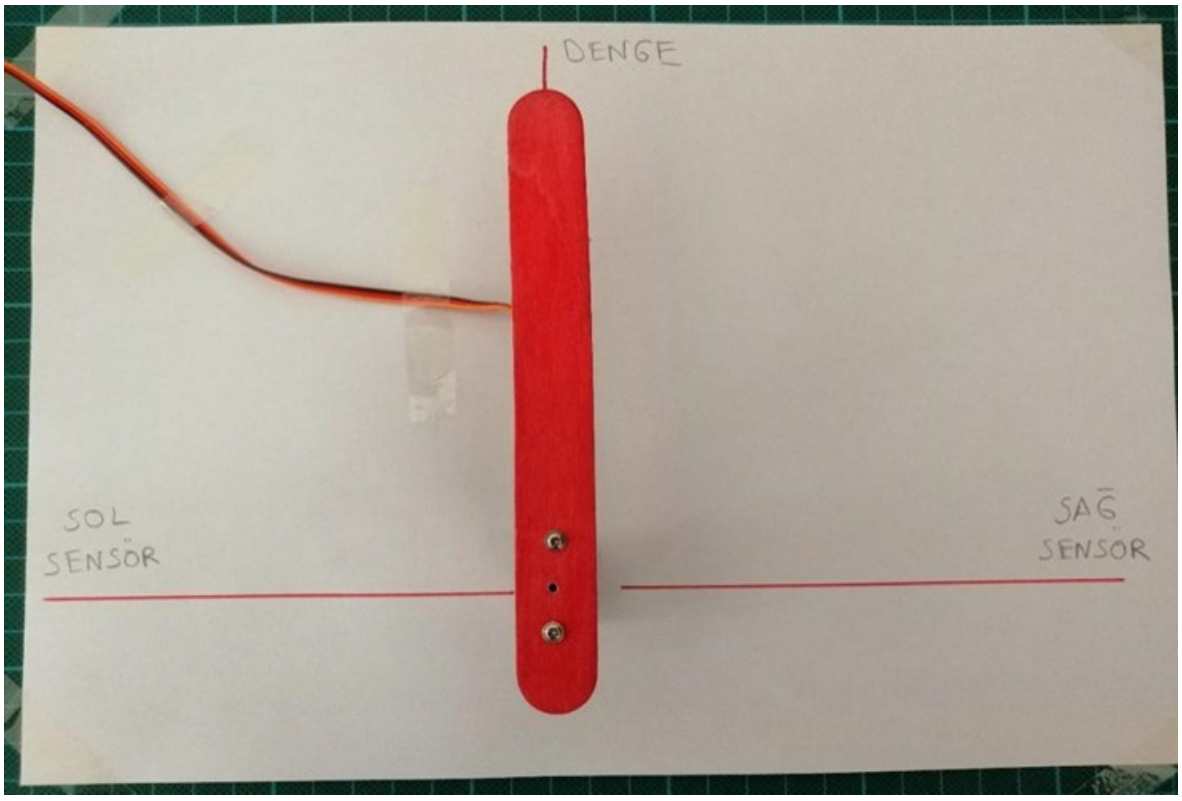
Uygulama Adımları

Adım 1: Servo motor bağlantı aparatı bir Abeslang üzerine vida ile monte edilir (Resim 2.11).

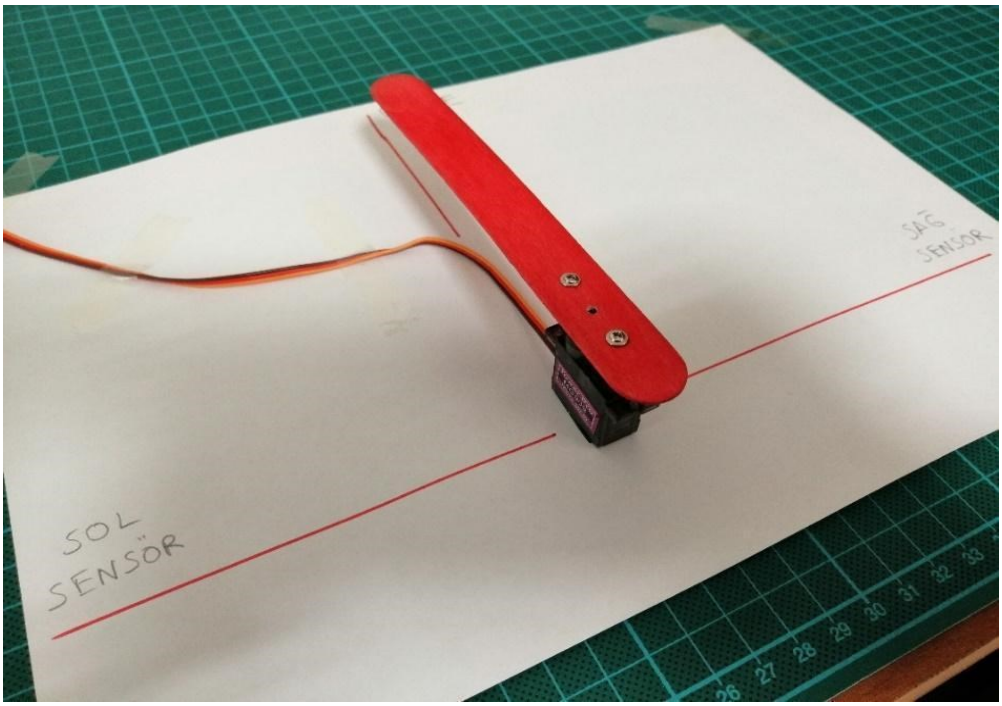


Resim 2.11. Servo motor bağlantı aparatına vidalanmış abeslang

Adım 2: Servo motor bağlantı aparatına bağlanmış abeslang motor üzerine yerleştirilir. Bir A4 kâğıdı üzerine Sağ Sensör, Sol Sensör ve Denge yazılır (Resim 2.12). Servo motor bir çift taraflı bant ile kâğıt üzerine yapıştırılır. Kâğıt ise zemine çift taraflı bant ya da şeffaf bant ile sabitlenir (Resim 2.13). Uygulamada sağ LDR sensörü üzerine düşen ışık eşik değerini aştığında ibre saat yönünde 90 derece dönecektir. Sol LDR sensörü aktif duruma geçtiğinde ise saat yönünün tersi yönünde 90 derece dönecektir. Her iki LDR de aktif veya pasif olduğunda ise ibre Denge konumuna gelecektir.

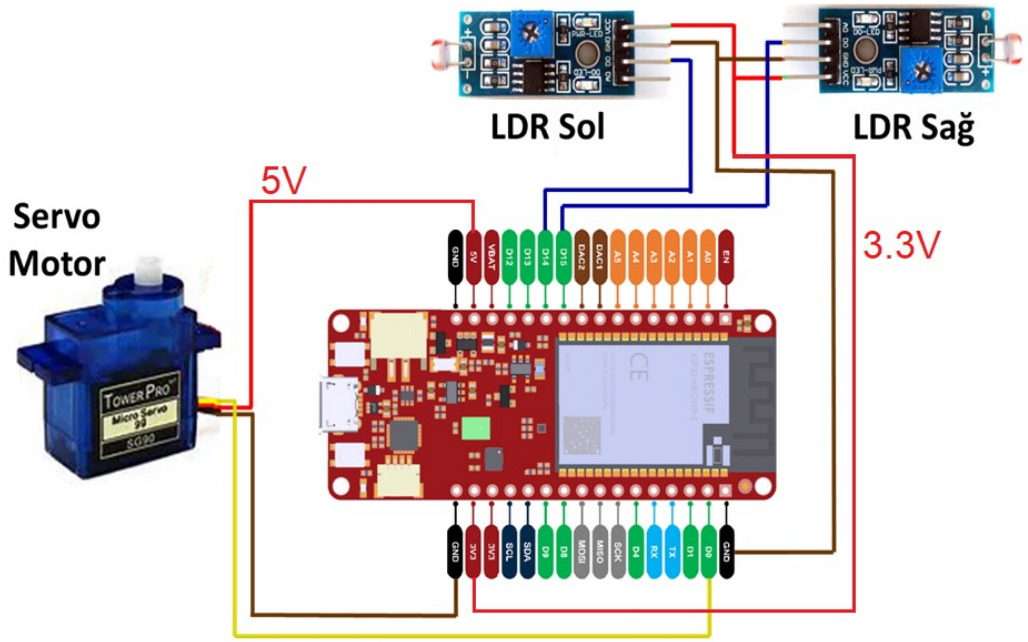


Resim 2.12. Uygulamada kullanılacak kağıdın hazırlanması

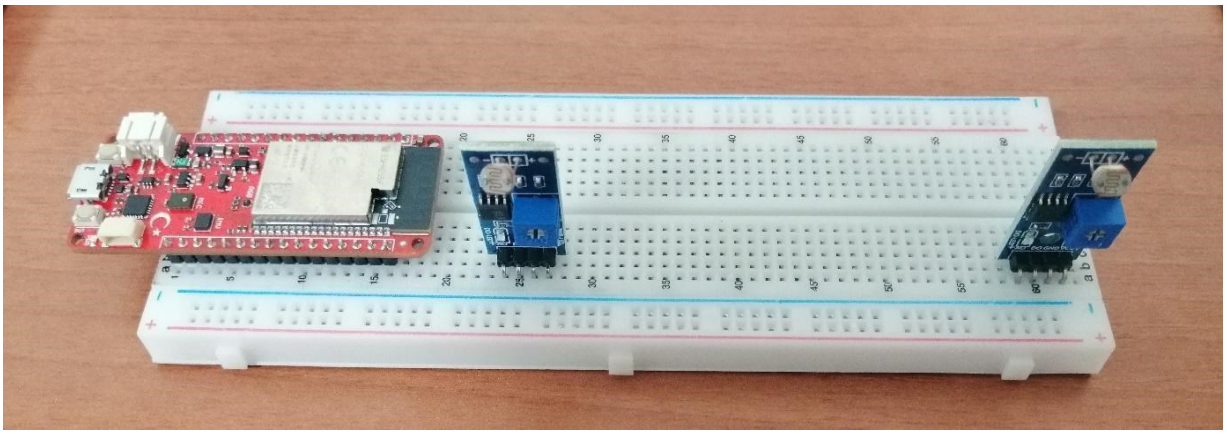


Resim 2.13. Uygulama düzeneği

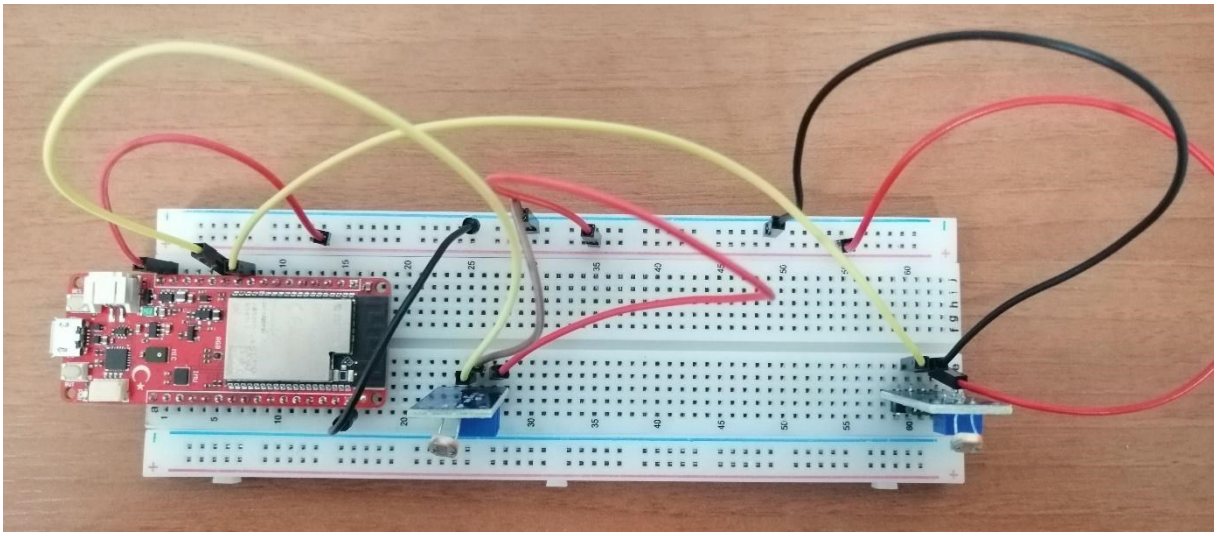
Adım 3: Uygulama için gerekli motor, sensör ve Deneyap Geliştirme Kartı bağlantıları Resim 2.14'te verilmiştir. Breadboard üzerine Deneyap Geliştirme Kartı ve LDR sensörleri yerleştirilir (Resim 2.15). Sensörlerin güç bağlantıları (Vcc ve GND) Deneyap Geliştirme Kartı'nın VUSB ve GND girişlerine yapılacaktır. Bunun için breadboard üzerindeki güç bağlantılarının yapıldığı şeritler kullanılabilir. Sol ve sağ LDR sensörlerinin DO çıkışları Deneyap Geliştirme Kartı'nın sırası ile D14 ve D15 girişlerine bağlanır (Resim 2.16).



Resim 2.14. Uygulama 1 pin bağlantıları



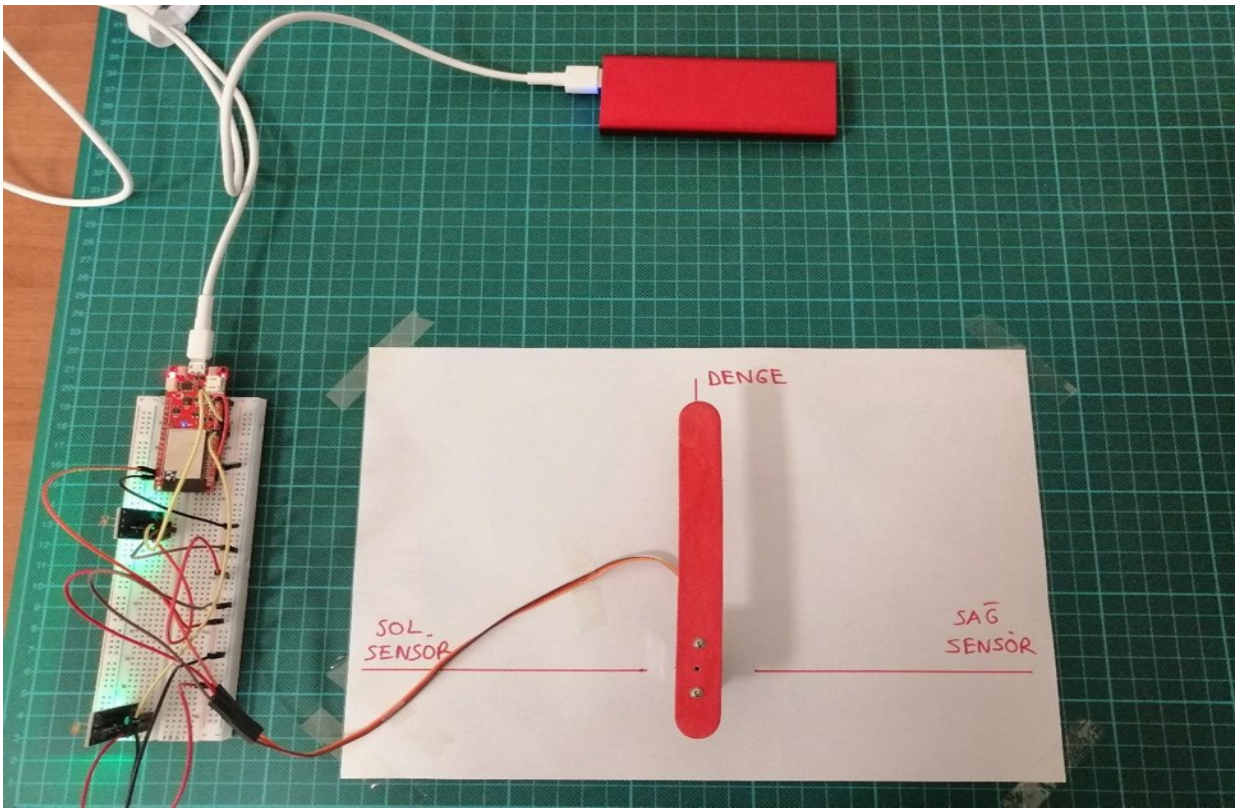
Resim 2.15. Deneyap Geliştirme Kartı'nın ve LDR sensörlerinin breadboard üzerindeki konumları



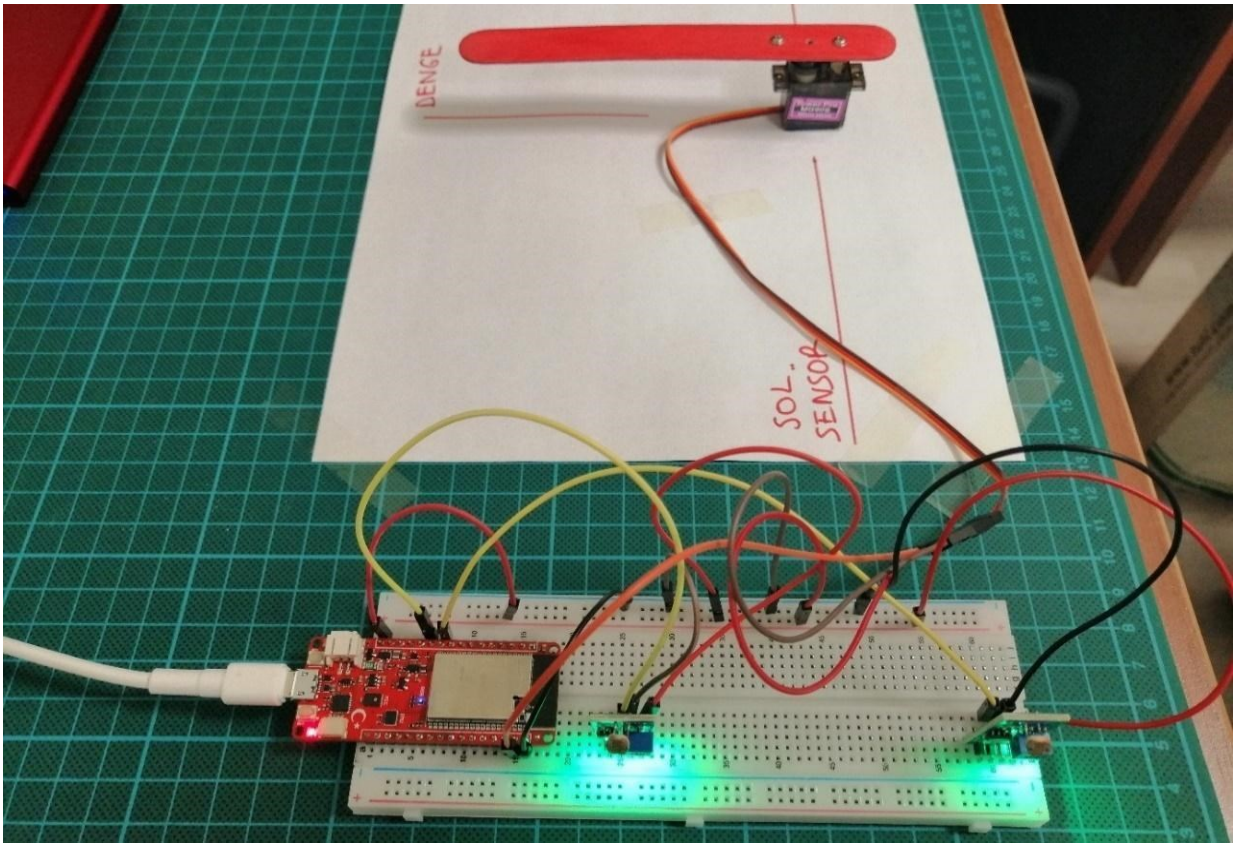
Resim 2.16. Deneyap Geliştirme Kartı'nın ve LDR sensörlerinin kablo bağlantıları

Adım 4: Servo motorun bağlantıları yapılır. Motorun kırmızı kablosu Vcc'ye, kahverengi kablosu GND'ye bağlanır. Kontrol sinyalinin gönderileceği turuncu kablo ise Deneyap Geliştirme Kartı'nın D0 pinine bağlanır.

Uygulama 1 için tüm bağlantıların yapılmış hali Resim 2.17 ve Resim 2.18'de verilmiştir.



Resim 2.17. Uygulama 1 genel görünüm



Resim 2.18. Uygulama 1 genel görünüm

Adım 5: Uygulama kodları Arduino IDE üzerinde yazılarak Deneyap Geliştirme Kartı'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

İlk olarak Deneyap Geliştirme Kartı ve servo motor kütüphaneleri yüklenir. Deneyap Geliştirme Kartı üzerinde ESP32 işlemcisi bulunduğundan servo motor kütüphanesi bu işlemciye uygun olmalıdır.

```
#include <Deneyap_Servo.h>          //Servo motor kütüphanesinin programa eklenmesi
```

Motor ve sensörlere ilişkin pin tanımlamaları yapılır. LDR'lerin dijital çıkışları kullanılacağından Deneyap Geliştirme Kartı'nın dijital pinleri tanımlanmalıdır. Ardından değişken tanımlamaları yapılır.

```
#define SERVOPIN D0                 //servo motorun bağlandığı D0 pininin tanımlaması
#define L1_pin D14                  //sağ ışık sensörünün bağlandığı D14 pininin tanımlaması
#define L2_pin D15                  //sol ışık sensörünün bağlandığı D15 pininin tanımlaması
Servo myservo;                  //servo motora ilişkin değişken tanımlaması
int sensor1, sensor2;               //LDR'lerin değişkenlerinin tanımlanması
int pos = 0;                        //servo motor pozisyonuna ilişkin değişken tanımlaması
```

Setup bloğu içerisinde başlangıç ayarlamaları yapılır.

```
myservo.attach(SERVOPIN);
pinMode(L1_pin, INPUT);
pinMode(L2_pin, INPUT);
```

"void loop" içerisine her bir döngüde yapılması gereken işlemler yazılır.

```
 sensor1 = digitalRead(L1_pin);     //sağ ışık sensöründen gelen dijital verinin okunması
 sensor2 = digitalRead(L2_pin);     //sol ışık sensöründen gelen dijital verinin okunması
```

LDR'ler üzerine düşen ışık miktarlarının birbirlerine göre karşılaştırılması ve servo motor üzerine bağlı ibrenin ışık miktarı fazla olan sensör tarafına doğru dönmesinin sağlanması için gerken kodlar yazılır.

```
 if (sensor1>sensor2)               //sağ LDR'ye düşen ışık > sol LDR'ye düşen ışık
  {
    pos=0;
    myservo.write(pos);             //pozisyon değeri 0 derece olarak motora gönderilir.
  }
```

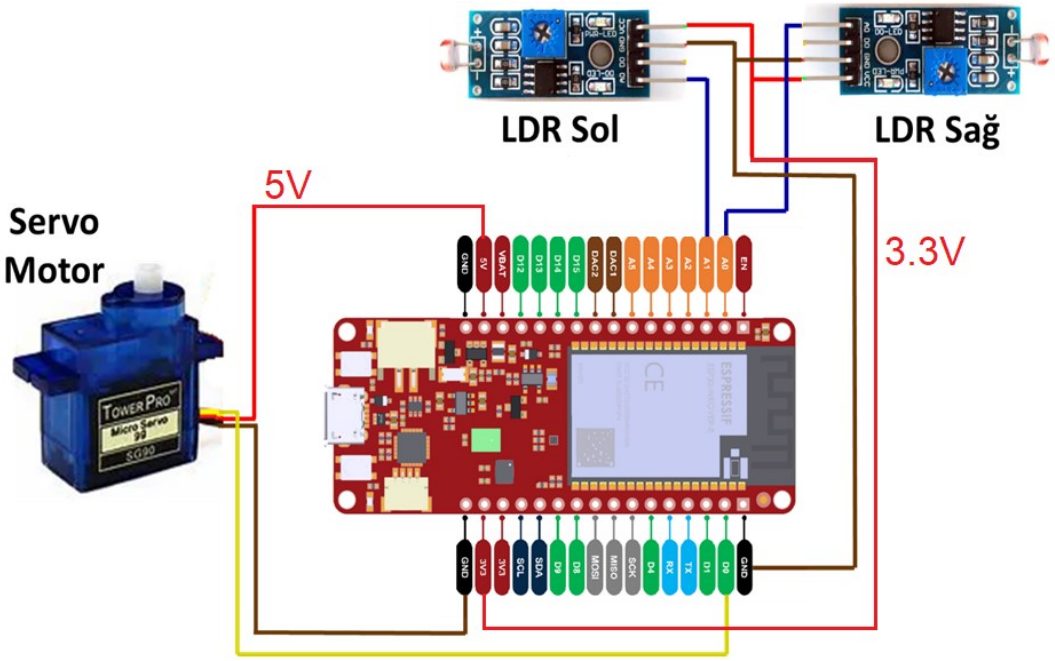
```
  else if(sensor2>sensor1)          //sol LDR'ye düşen ışık  > sağ LDR'ye düşen ışık
  {
    pos=180;
    myservo.write(pos);             //pozisyon değeri 180 derece olarak motora gönderilir.
  }
  else if(sensor2=sensor1)          //sağ LDR'ye düşen ışık = sol LDR'ye düşen ışık
  {
    pos=90;
    myservo.write(pos);             //pozisyon değeri 90 derece olarak motora gönderilir.
  }

delay(5);
}
```

Adım 6: Kodlar yazılıp karta yüklendikten sonra sağ ve sol LDR'lere el feneri veya cep telefonu feneri tutulduğunda ibrenin sağa ve sola dönüşü gözlemlenmelidir.

NOT: LDR'lerin eşik değerleri ortamdaki ışık miktarına göre trimpotlar üzerinden ayarlanmalıdır. El feneri ışığını tutmadan önce trimpotu DO-LED ışığı sönene kadar saat yönünün tersine çeviriniz.

1.5. Uygula: LDR Sensörlerinin Analog Çıkışları ile Servo Motor Kontrolü

Önceki bölümlerde LDR sensörlerinin dijital ve analog çıkış verebildiğinden bahsedilmişti. Analog çıkışta okunan değer ortamdaki ışık şiddetini vermektedir. Bu uygulamada Uygulama 1'den farklı olarak gösterge ibresinin ara değerleri de göstermesi sağlanacaktır. Sağ ve sol LDR sensörlerinin her ikisine düşen ışık yüzdesine bağlı olarak ibre konumlandırılacaktır. Bir önceki uygulamadan farklı olarak sağ ve sol LDR sensörleri sırası ile A0 ve A1 analog girişlerine bağlanacaktır (Resim 2.19).



Resim 2.19. LDR'lerin ve servo motorun Deneyap Geliştirme Kartı'na bağlantısı (Uygulama 2)

Uygulama 2 için gerekli Deneyap Geliştirme Kartı kodları aşağıda verilmiştir.

```
#include <Deneyap_Servo.h>
#define SERVOPIN D0                    //servo motor bağlantısı
#define L1_pin A0                      //sağ ışık sensörü bağlantısı, Analog Giriş 0
#define L2_pin A1                      //sol ışık sensörü bağlantısı, Analog Giriş 1
Servo myservo;
int sensor1, sensor2;
int pos = 0;
void setup() {
  myservo.attach(SERVOPIN);
  pinMode(L1_pin, INPUT);
  pinMode(L2_pin, INPUT);
}
void loop() {
  sensor1 = (4095-analogRead(L1_pin))*90/4095;      //ışık miktarının analog değer olarak tespiti
  sensor2 = (4095-analogRead(L2_pin))*90/4095;
```

```
if (sensor1>sensor2)                    //sağ LDR'ye düşen ışık > sol LDR'ye düşen ışık
{
  pos=(sensor1-sensor2)+90;             //LDRlere düşen ışık miktarı farkına göre konum belirlenir.
  myservo.write(pos);
}
else if(sensor2>sensor1)
{
  pos=-1*(sensor2-sensor1)+90;
  myservo.write(pos);
}
delay(5);
}
```

Kodlar yazılıp karta yüklendikten sonra sağ ve sol LDR'lere el feneri veya cep telefonu feneri tutulduğunda ibre sağa ve sola dönecektir. Bir önceki uygulamadan farklı olarak ibrenin tam sağa veya tam sola değil, LDR'ler üzerine düşen ışık dağılımına göre ara değerleri alacak şekilde döndüğü gözlemlenmelidir.

2. TASARLA VE ÜRET

2.1. Tasarla: Ateş Ölçer Cihazı

Sağlıklı bir insanın vücut sıcaklığı 36,5 °C – 37,5 °C arasındadır. Vücut herhangi bir hastalık etkenine karşı savaştığında (virüs, bakteri vb.) vücut sıcaklığında yükselme meydana gelir. Vücut sıcaklığının 38 °C'yi aştığı durumlarda hastalık şüphesi oluşur ve 39 °C üzerinde kişinin doktora görünmesi gerekir. Günümüzde tüm dünyayı etkisi altına alan Covid-19 pandemisinde de en belirgin hastalık semptomlarından birisi yüksek ateştir. Bu nedenle çeşitli ateş ölçüm cihazları ile kişilerin vücut sıcaklıkları ölçülmekte, şüpheli durumlarda kişiler toplu ortamlara sokulmamakta, detaylı tahlil ve testler için sağlık kuruluşlarına gönderilmektedir.

Bu ders kapsamında öğrencilerden bir ateş ölçüm cihazı tasarlamaları beklenmektedir. Bu işlem için bir sıcaklık sensörü kullanılacaktır. Cihazın tasarımında aşağıdaki kriterler göz önüne alınmalıdır;

- Sıcaklık sensörünün analog veya dijital çıkış vermesine göre uygun giriş belirlenmelidir.
- Sıcaklık sensörünün marka ve modeline göre ölçüm aralığı, her bir derecelik sıcaklık değişimine karşılık çıkıştaki değişim miktarı gibi parametreler belirlenmelidir.
- Sıcaklık sensörünün monte edileceği aparatın tasarımı yapılmalıdır. Burada 3 boyutlu yazıcı ile baskı yapılabileceği gibi abeslanglar da kullanılabilir.

- Sensör değerlerinin gözlenmesi için bu dersteki uygulama kapsamında üretilen gösterge yapısı kullanılabilir. Bunun haricinde öğrencilerin kendi tasarlayacakları farklı yapılar da kullanılabilir.

2.2. Üret: Ateş Ölçer Cihazı

Bu aşamada bir önceki bölümde tasarlanan ateş ölçüm cihazının üretimi yapılır. Sıcaklık sensörü aparata monte edilmeden önce sıcaklık değerlerini doğru ölçüp ölçmediği test edilir. Bunun için ortam sıcaklığı değeri ölçülürken sensör üzerine parmak değdirilerek sıcaklığın yükseldiği görülebilir.

Uyarı:

Sensör testleri esnasında çakmak, kibrit vb. tehlikeli araçlar kullanmayınız.

3. DEĞERLENDİR

Bu hafta yapılan uygulamalarda LDR sensöründen analog ve dijital sinyallerin alınması ve bir servo motorun nasıl sürüleceği gösterilmiştir.

Ders sonunda öğrencilerle bu hafta yapılan uygulamalar hakkında konuşulur ve aşağıdaki sorular üzerinden tartışmalar gerçekleştirilir.

- Ortamdaki ışık miktarının ölçümü için başka ne tür sensörler kullanılabilir?
- Tasarım süreci ile ilgili bir problem yaşadınız mı? Yaşadıysanız bunu nasıl çözdünüz?
- Devrelerin fiziki kurulumunda zorlandığınız yerler oldu mu?
- Bu derste bahsi geçen tek serbestlik dereceli sistemlerin pratik hayattaki uygulamaları (veya endüstriyel uygulamaları) neler olabilir? Örnek uygulamalar:
    - Geniş bir alanda (örneğin bir işyerinde) ışığa duyarlı sağa sola hareket edebilen bir güvenlik kamera kontrol sistemi geliştirilebilir. Gece karanlıkta o alana elinde bir el feneri ile giren bir hırsız, bu sistem sayesinde tespit edilir ve kamera da o hırsıza yönlendirilip net olarak görüntüsü alınır ve ayrıca alarm da aktif hale getirilir.
    - Bir sınır karakolunda, ışığa duyarlı sağa sola hareket edebilen bir silah sistemi geliştirilebilir. Gece karanlıkta sınır hattından terörist veya düşman tarafından yapılan ateşli bir saldırıya anında karşılık verilebilir. Işık sensörleri ilk ateş edilen konuma yönlendirilip karşı ateş edildiğinde bu çok etkili bir savunma hareketi olur. Karakolda herkes uyuduğu anda böyle bir sistem harekete geçerek hayat kurtarır.
    - Sıcaklık ölçümü ihtiva eden sistemler, endüstriyel alan oldukça yaygındır. Hemen hemen her endüstriyel sistemin sıcaklık ölçümüne bağlı bir mekanik yapısı mevcuttur. Genellikle soğutma fanları sıcaklık ölçümü değerlerine göre çalışmaktadır. Örneğin bir dizüstü bilgisayarın soğutma sistemi incelendiğinde (hatta dışarıdan sesini dinleyerek), bilgisayarın fan hızının sıcaklık durumuna göre

değiştiği gözlemlenir. Burada amaç, fanı sadece gerektiği anlarda gerektiği hızlarda çalıştırarak hem gürültüyü hem de pil sarfiyatını minimuma indirmektir.

4. İLAVE ETKİNLİK

Kalan zamana göre öğrencilerden ortam sıcaklığına göre dönüş hızı değişen bir vantilatör yapmaları istenir. Burada kullanılan servo motorların açı sınırlaması olduğu için motorlar tam tur döndürülmemeli, belli bir açı içerisinde ileri-geri dönüş hızlarının ortam sıcaklığına göre değiştirilmesi istenmelidir.

3. Bölüm: Mesafe ve IMU Sensörleri ile Servo ve DC Motor Kontrolü

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.

Haftanın Kazanımları:

- Öğrenciler Deneyap Geliştirme Kartı üzerinde ultrasonik mesafe sensörü çalıştırır.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde IMU sensörü çalıştırır.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde DC motor çalıştırır.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde zil çalıştırır.
- Öğrencileri motor kontrol parametrelerinin etkilerini deneyimler.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin ultrasonik mesafe sensöründen ve Deneyap Geliştirme Kartı üzerinde bulunan 6 eksen IMU sensöründen alınan analog sinyaller ile bir servo ve DC motoru hareket ettirebilmesini ve bir zil üzerinden ses çıkarmasını sağlamaktır.

Kullanılacak Malzemeler:

1. Deneyap Geliştirme Kartı, 1 adet
2. Ultrasonik mesafe sensörü, 1 adet
3. Servo Motor (TowerPro SG90), 1 adet
4. DC Motor, 1 adet
5. L298N motor sürücü, 1 adet
6. Tekerlek, 1 adet
7. Zil, 1 adet
8. 100Ω direnç, 1 adet
9. Breadboard, 1 adet
10. Farklı renklerde erkek-erkek ve erkek-dişi jumper kabloları, 20+ adet
11. Mikro USB kablosu, 1 adet
12. Harici 5V USB Batarya, 1 adet
13. Çift taraflı silikon bant ve makas
14. Abeslang, 1 adet

Haftanın İşlenişi:

Gözle: Deneyap Geliştirme Kartı'na ilgili eleman bağlantılarının yapılması ve kodların gösterilmesi

Uygula: Yönergeye uygun şekilde tüm bağlantıların yapılması ve kodların yazılması

Tasarla: Verilen kriterlere göre denge çubuğu tasarımının yapılması

Üret: Tasarlanan denge çubuğunun üretilmesi

Değerlendir: Deney sonuçlarının ve yapılabilecek farklı uygulamaların tartışılması

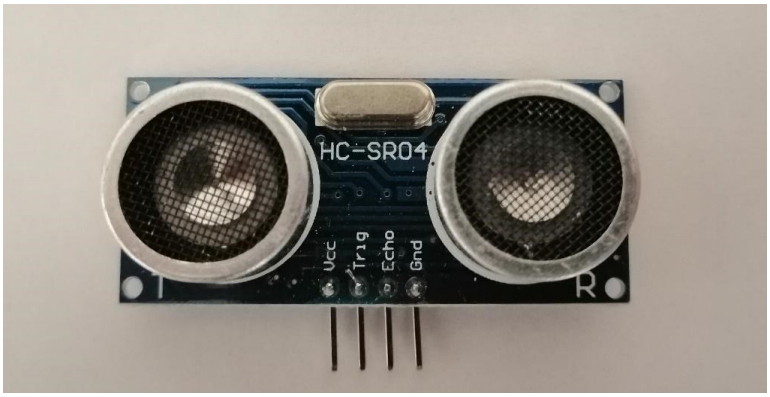
1. GÖZLE VE UYGULA

1.1. Gözle: Ultrasonik Mesafe Sensörü Nedir?

Ses dalgalarını kullanarak mesafe ölçmeye yarayan sensör türlerinden olan Ultrasonik mesafe sensörleri, mesafe ölçümü ve çevredeki nesnelerin tespiti amacıyla kullanıldığı robot uygulamalarında önemli bir yere sahiptirler. Radarlarla tamamen aynı mantığa sahip olan bu sensörler yunuslar ve yarasalardan esinlenilerek geliştirilmiştir. Yunuslar ve yarasalar da ses dalgası göndererek karşılarında olan cisimlerin uzaklığını tespit etmektedirler. Bu sensörler insan kulağının duyabileceği frekanstan (20 KHz) daha yüksek frekansta ses sinyalleri yayarlar. Geri dönen sinyalin geliş süresine göre karşıdaki cisme olan mesafe hesaplanır.

Bu uygulamada HC-SR04 model ultrasonik sensör kullanılacaktır. Bu sensörde 40 KHz frekanslı 8 kare dalga yapısında bir ses sinyali üretilerek karşıya gönderilmektedir. 8 adet kare dalga olmasının sebebi ortamdaki diğer seslerden ayrılmasıdır. Karşıdaki cisme çarpıp geri dönen bu ses sinyali alıcı tarafından alınır. Sinyalin gidişi ve gelişi arasında geçen süre cismin uzaklığına bağlıdır. Ses sinyalinin hızının 343 m/s olduğu ve mesafenin hız ile zamanın çarpımı olduğu bilindiğinden basit matematiksel işlemler ile cismin sensöre olan uzaklığı kolaylıkla hesaplanmaktadır. HC-SR04 sensörü ile 2-200 cm arası mesafeler ölçülebilmektedir.

Resim 3.1'de HC-SR04 sensörü görülmektedir. Sensör üzerinde 4 pin bulunmaktadır. Bunlardan Vcc pinine sensörün besleme gerilimi (5V) uygulanmaktadır. GND pinine işlemcinin topraklama pini bağlanır. Trig pini ses dalgasını göndermek için kullanılmaktadır. Bu pin tetiklendiğinde ses dalgası karşıya gönderilmektedir. Echo pini ise gönderilen sinyal geri geldiğinde bir kare sinyal üretir. Bu sinyalin periyodu, iletilen sinyalin algılanması için geçen süre ile orantılıdır.



Resim 3.1. HC-SR04 ultrasonik sensör

1.2. Uygula: Ultrasonik Sensör ile Servo Motor Kontrolü

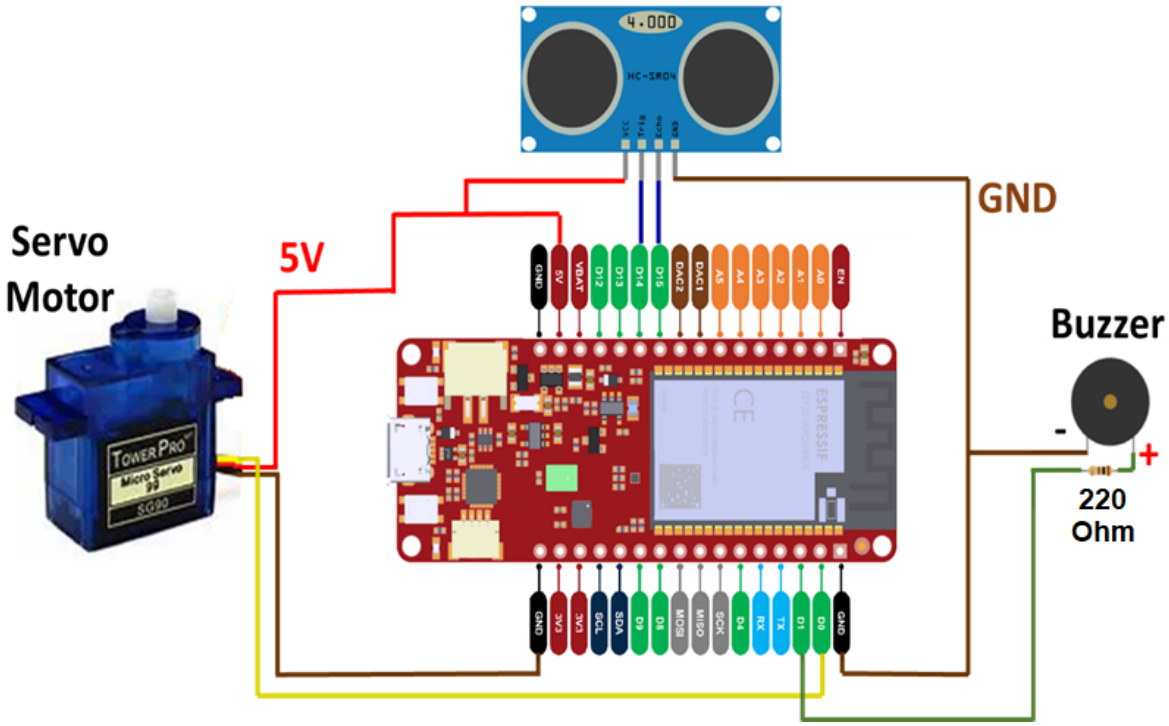
Bu uygulamada ultrasonik sensörden alınan mesafe bilgisine göre bir gösterge ibresi hareket ettirilecektir. Sensör ile cisim arasındaki mesafe belirli bir değerin altına düştüğünde zil ses çıkaracaktır. Zil, ses elde etmek amacıyla kullanılan bir elemandır. Zili günlük hayatta birçok yerde

görebiliriz. Araçlardaki park sensörleri, gaz kaçağı durumlarında uyarı veren sistemler, yangın alarmları, barkod okuma sistemleri gibi uygulamalarda zil kullanılmaktadır.

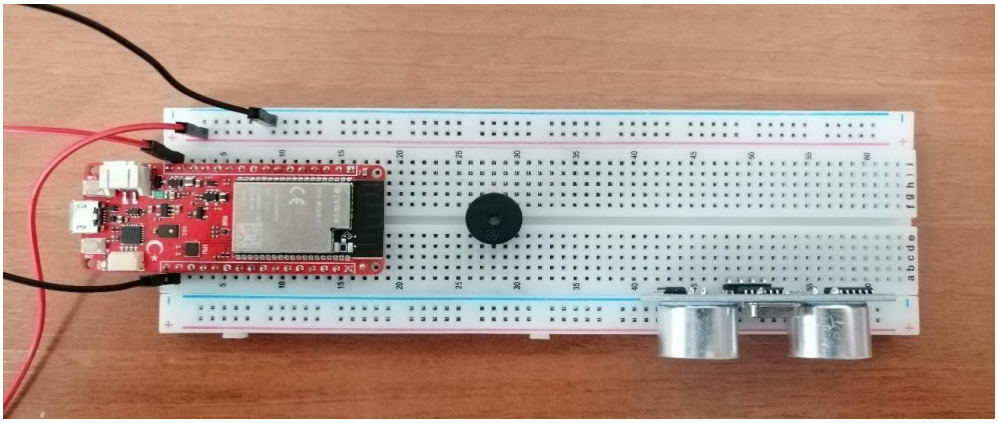
Uygulama Adımları

Adım 1: Geçen haftaki Uygulama 1 ve 2'de kullanılan düzenek bu uygulama için de kullanılacaktır. Sadece kâğıt üzerindeki yazılar cm cinsinden mesafe olarak değiştirilmelidir.

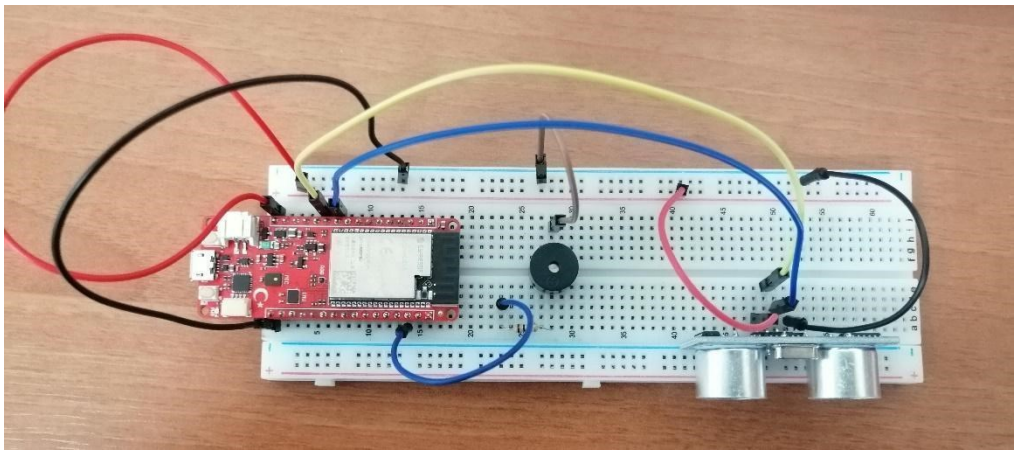
Adım 2: Uygulama için gerekli motor, sensör ve Deneyap Geliştirme Kartı bağlantıları Resim 3.2'de verilmiştir. Breadboard üzerine Deneyap Geliştirme Kartı, ultrasonik sensör, zil ve direnç yerleştirilir (Resim 3.3). Sensörün güç bağlantıları (Vcc ve GND) Deneyap Geliştirme Kartının 5V ve GND girişlerine yapılacaktır. Ultrasonik sensörün TRIG ve ECHO pinleri sırası ile Deneyap Geliştirme Kartı'nın D14 ve D15 pinlerine bağlanır. Zilin + ucu direnç üzerinden geçirilerek Deneyap Geliştirme Kartı'nın D1 pinine bağlanır. Zilin – ucu ise GND'ye bağlanır. (Resim 3.4).



Resim 3.2. Uygulama 1 bağlantı şeması



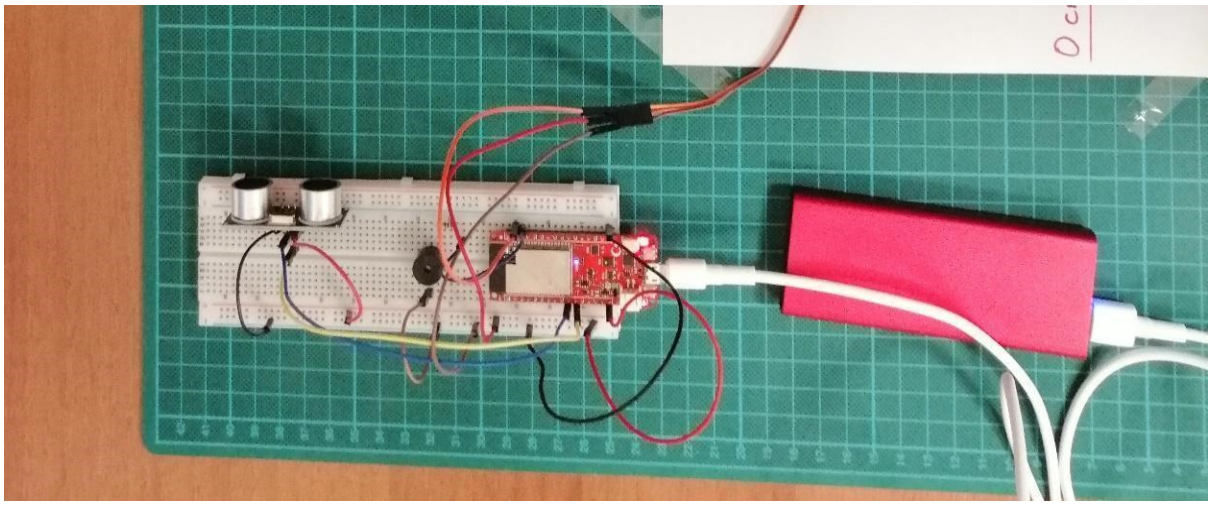
Resim 3.3. Ultrasonik sensörün ve zilin breadboard üzerindeki yerleşimi



Resim 3.4. Ultrasonik sensörün ve zilin Deneyap Geliştirme Kartı'na bağlantısı

Adım 3: Servo motorun bağlantıları yapılır. Motorun kırmızı kablosu Vcc'ye, kahverengi kablosu GND'ye bağlanır. Kontrol sinyalinin gönderileceği turuncu kablo ise Deneyap Geliştirme Kartı'nın D0 pinine bağlanır.

Uygulama 1 için tüm bağlantıların yapılmış hali Resim 3.5'te verilmiştir.



Resim 3.5. Uygulama 1 için eleman bağlantıları

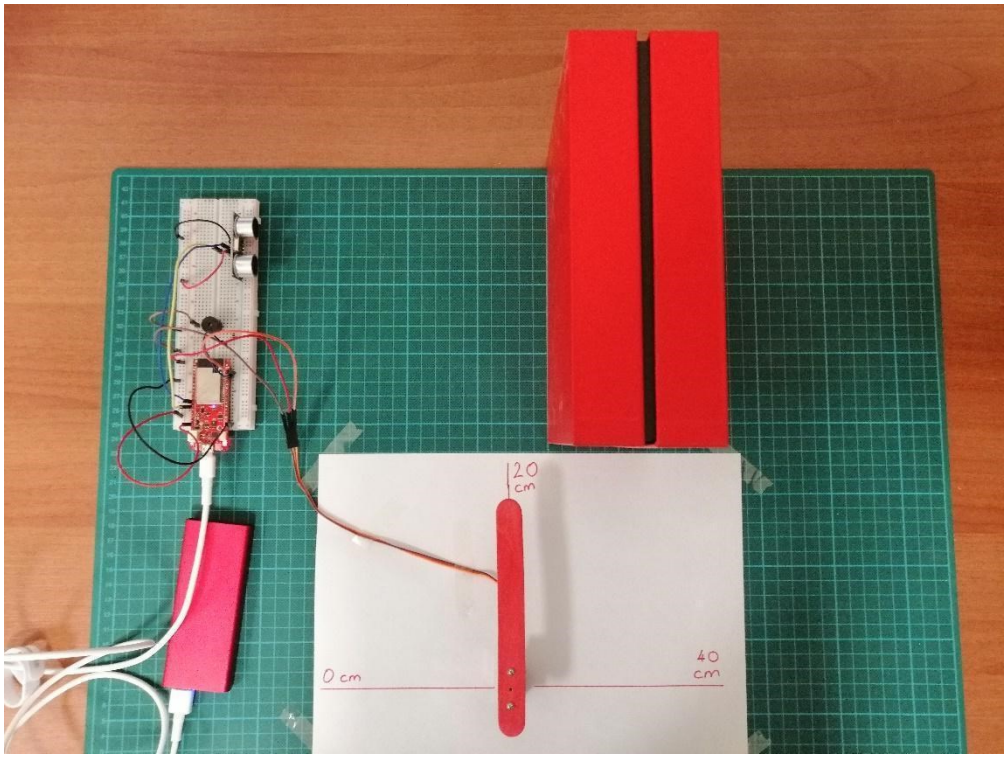
Adım 4: Uygulama kodları Arduino IDE üzerinde yazılarak Deneyap Geliştirme Kartı'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

```
#include <Deneyap_Servo.h>

#define SERVOPIN1 D0          //servo motor bağlantısı
#define trig_pin D14          //ultrasonik sensör trigger pin bağlantısı
#define echo_pin D15          //ultrasonik sensör echo pin bağlantısı
#define buzzer_pin D1         //zil bağlantısı

Servo myservo;

int pos = 0;
```

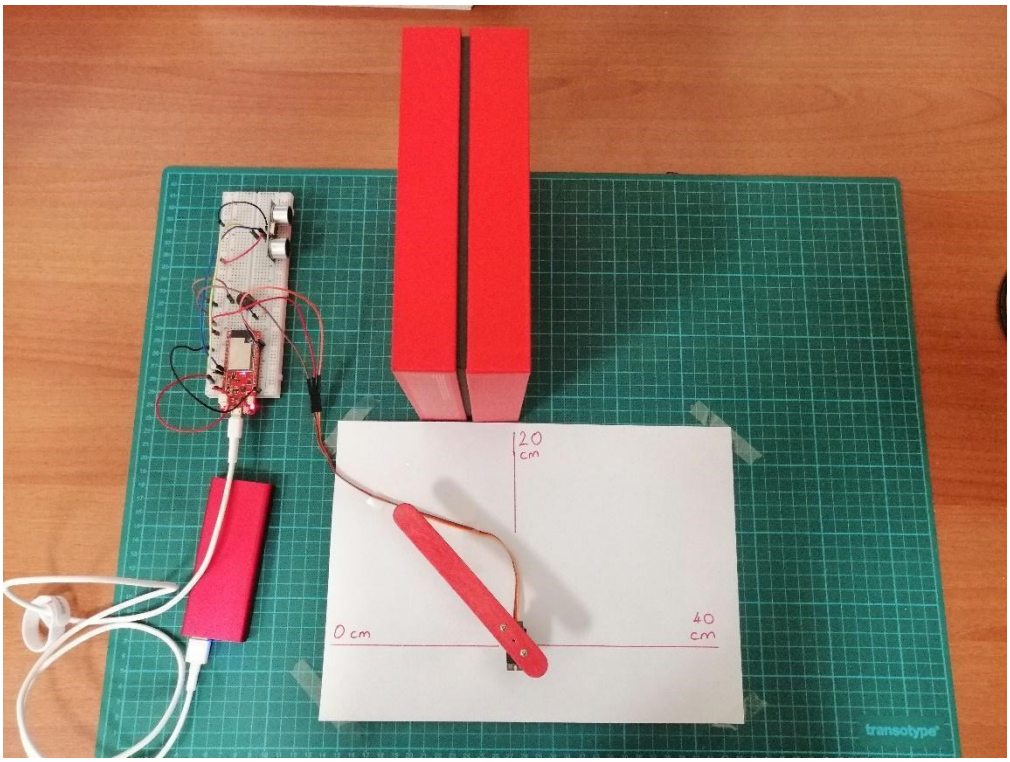
```
long sure;
long uzaklik;

void setup() {
  myservo.attach(SERVOPIN1);
  Serial.begin(115200);          //serial monitör üzerinde uzaklık değerini görmek için
  pinMode(trig_pin, OUTPUT);          //ultrasonik sensör trigger pinini çıkış olarak ayarlıyoruz
  pinMode(echo_pin,INPUT);          //ultrasonik sensör echo pinini giriş olarak ayarlıyoruz
  pinMode(buzzer_pin,OUTPUT);          //zil pinini çıkış olarak ayarlıyoruz
}

void loop() {
  digitalWrite(trig_pin, LOW);
  delayMicroseconds(5);
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
//gönderilen sinyalin echo pinine geliş süresinin tespiti
  sure = pulseIn(echo_pin, HIGH);
  //ses hızına göre uzaklığın hesaplanması (gidiş geliş olduğu için 2'ye bölünmektedir.)
  uzaklik= sure /29.1/2;

  if(uzaklik < 10)                    //uzaklık değeri 10 cm'den küçükse zil aktif
  digitalWrite(buzzer_pin, HIGH);
  if(uzaklik >= 10)                    //uzaklık değeri 10 cm'den büyükse zil pasif
  digitalWrite(buzzer_pin, LOW);

  Serial.print("Uzaklık");                    //uzaklık değeri serial monitör ekranına yazdırılıyor.
  Serial.print(uzaklik);
```

```
  Serial.println("CM");
//uzaklık değerine göre (max. 40 cm) ibrenin hareketinin sağlanması
  pos=uzaklik*(180/40);
  myservo.write(pos);
  delay(5);
}
```

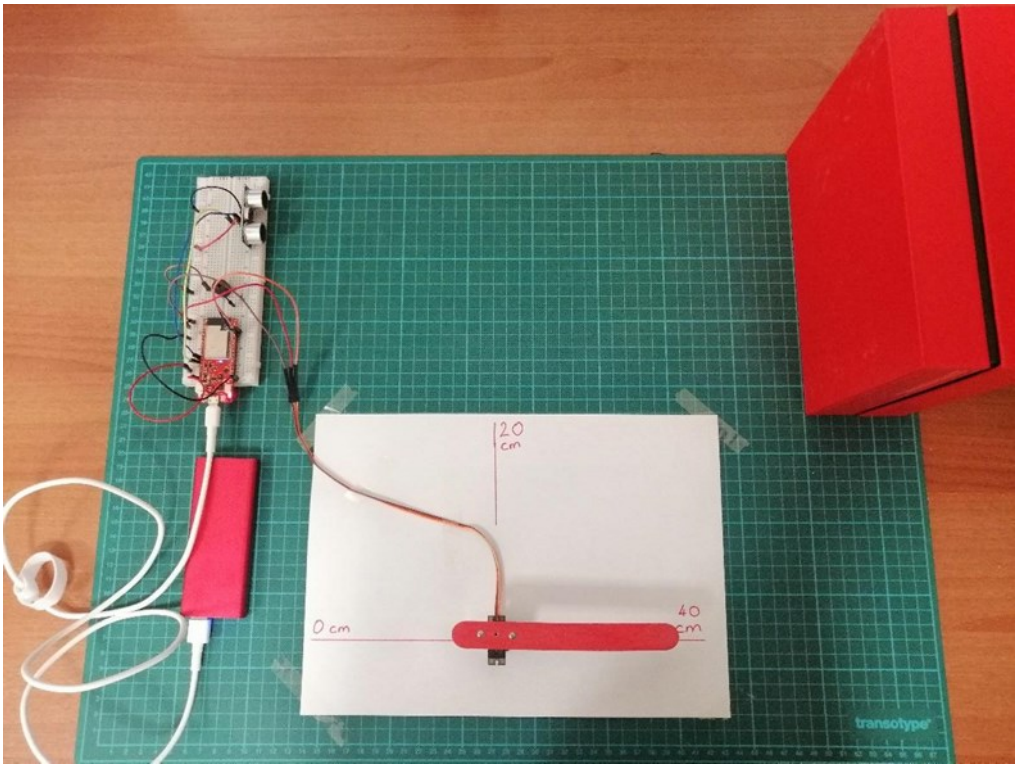
Adım 5: Kodlar yazılıp karta yüklendikten sonra ultrasonik sensörün önüne bir cisim yerleştirilir ve ileri geri hareket ettirilir. İbrenin de sensör ile cisim arasındaki mesafeye göre hareket ettiği gözlemlenir. Sensör ile cisim arasındaki mesafe 10 cm'nin altına düştüğünde zil ses çıkaracaktır.



Resim 3.6. Mesafe 20 cm iken ibrenin konumu



Resim 3.7. Mesafe 10 cm iken ibrenin konumu (zil aktif)



Resim 3.8. Mesafe 40 cm iken ibrenin konumu

1.3. Gözle: IMU (Inertial Measurement Unit) Sensörü Nedir?

IMU sensörleri robotik projelerinde sıklıkla kullanılan ve 3 boyutlu uzayda ivme ve dönüş ölçen sensörlerdir. Bir mobil robotun, bir insansız hava veya deniz aracının iki veya üç boyutlu uzaydaki

yönelimi ve ivmesi bu sensörler ile ölçülebilmektedir. Deneyap Geliştirme Kartı üzerinde dahili olarak 3 eksenli bir dönüş ölçer ve 3 eksenli bir açısal ivme ölçer olmak üzere toplam 6 eksenli bir IMU sensörü bulunmaktadır.

1.4. Uygula: IMU Sensörü Değerlerinin Ekrana Yazdırılması

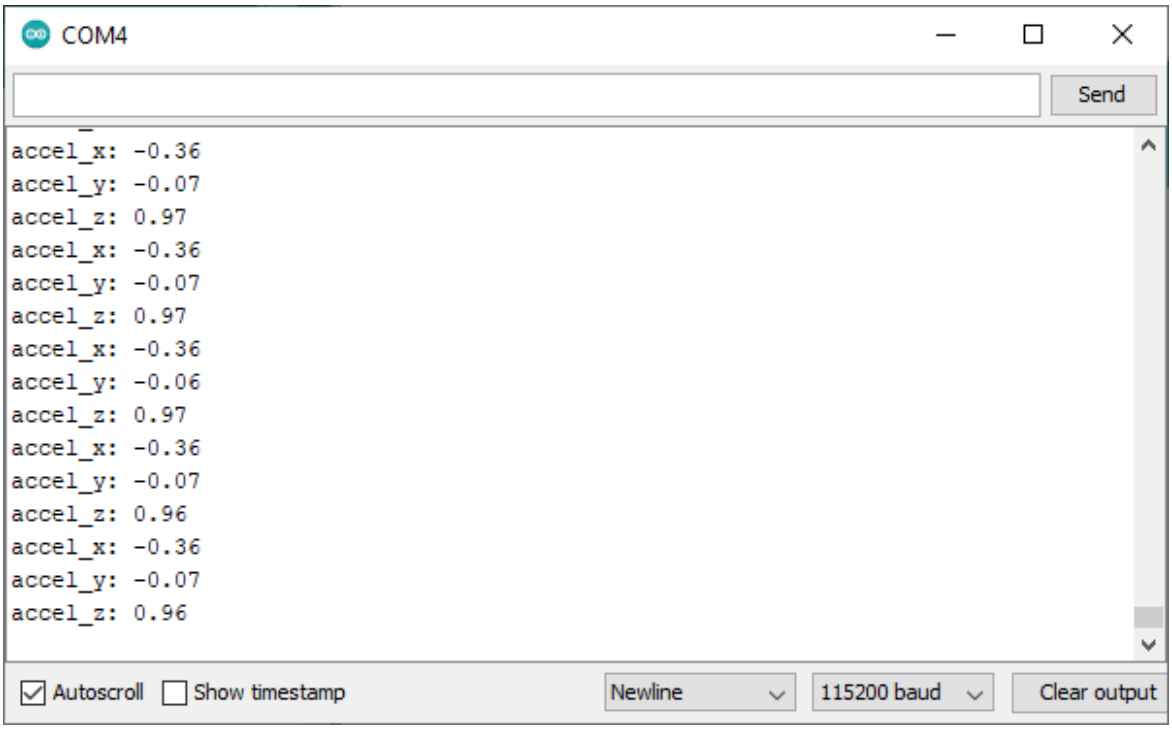
Bu uygulamada Deneyap Geliştirme Kartı üzerindeki IMU sensöründen verilerin nasıl alındığı ve kullanıma hazır hale getirildiği gösterilmektedir.

Uygulama Adımları

Adım 1: Deneyap Geliştirme Kartı bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include "lsm6dsm.h"

LSM6DSM IMU;

void setup()
{
  Serial.begin(115200);
  IMU.begin();
 }

void loop()
{
  Serial.print("accel_x: ");
  Serial.println(IMU.readFloatAccelX());
  Serial.print("accel_y: ");
  Serial.println(IMU.readFloatAccelY());
  Serial.print("accel_z: ");
  Serial.println(IMU.readFloatAccelZ());
  delay(500);
}
```

Adım 2: Programın Deneyap Geliştirme Kartı'na yüklemesi bittiğinde Arduino PC uygulamasının Serial Monitor penceresinde her 3-eksen için IMU sensör değerleri Resim 3.9' daki gibi gözlemlenebilmektedir. Deneyap Geliştirme Kartı'nın 3 boyutlu uzayda açısı değiştirildikçe bu değerlerin de değiştiği gözlemlenecektir.



Resim 3.9. IMU sensörü değerlerinin Serial Monitor' de gözlemlenmesi.

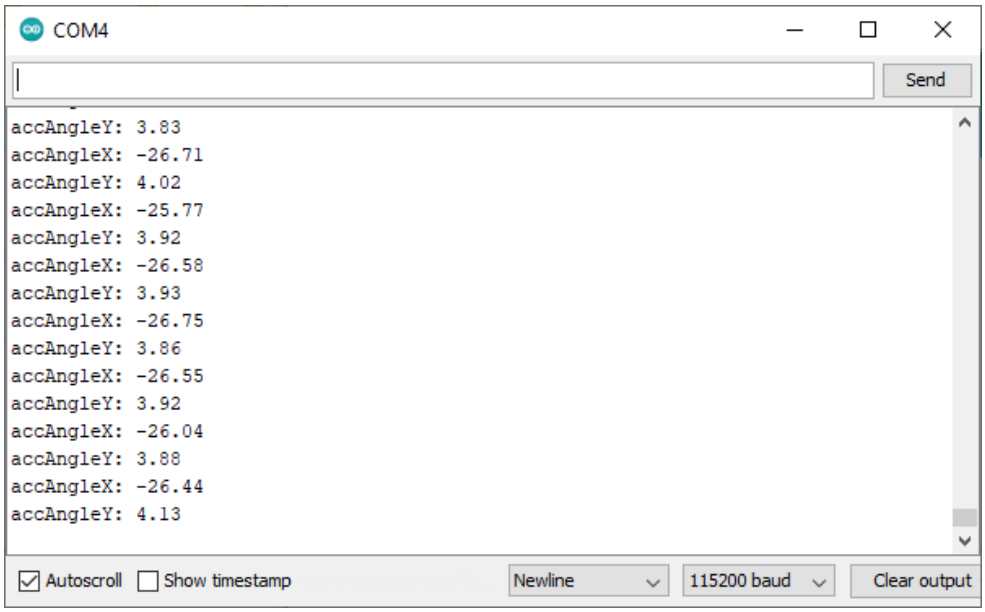
1.5. Uygula: IMU Sensör Değerlerinin Pan-Tilt Açı Değerlerine Dönüştürülmesi

Pan-tilt açılarının IMU sensör verisine göre hesaplanması aşağıda verilen kod ile sağlanmaktadır.

```
float accAngleX;
float accAngleY;

accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;

accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;

Serial.print("accAngleX: ");
Serial.println(accAngleX);
Serial.print("accAngleY: ");
Serial.println(accAngleY);
```

Bilgi Notu:
Kod içerisindeki "0.001" değerleri, olası herhangi bir tekillik (sıfıra bölme) durumunu bertaraf etmek için kullanılmıştır.

Resim 3.10'da gösterildiği gibi, Arduino PC uygulamasının Serial Monitor' ünde bu pan-tilt açı değerleri gözlenebilmektedir.

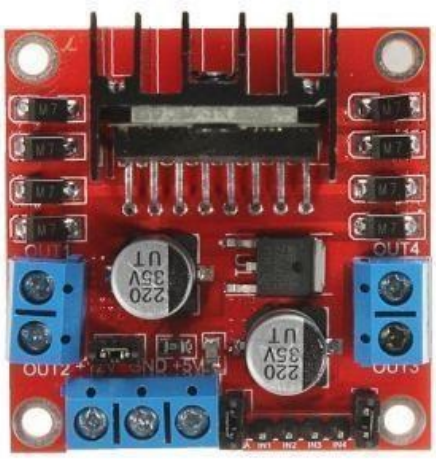


Resim 3.10. Açı değerlerinin Serial Monitor' de gözlemlenmesi.

1.6. Uygula: IMU Sensörü ile DC Motor Kontrolü

L298N motor sürücü entegresi dijital sinyaller ile çalışan ve röle, motor gibi endüktif yükleri kontrol edebilen 4 ampere kadar akım verebilen bir motor sürücü entegresidir. Bu entegre ile 2 adet motor sürülebilmektedir. L298 sürücü kartı incelendiğinde üzerinde çeşitli giriş ve çıkışlar bulunmaktadır. Bu giriş çıkışlar aşağıdaki tabloda verilmiştir.

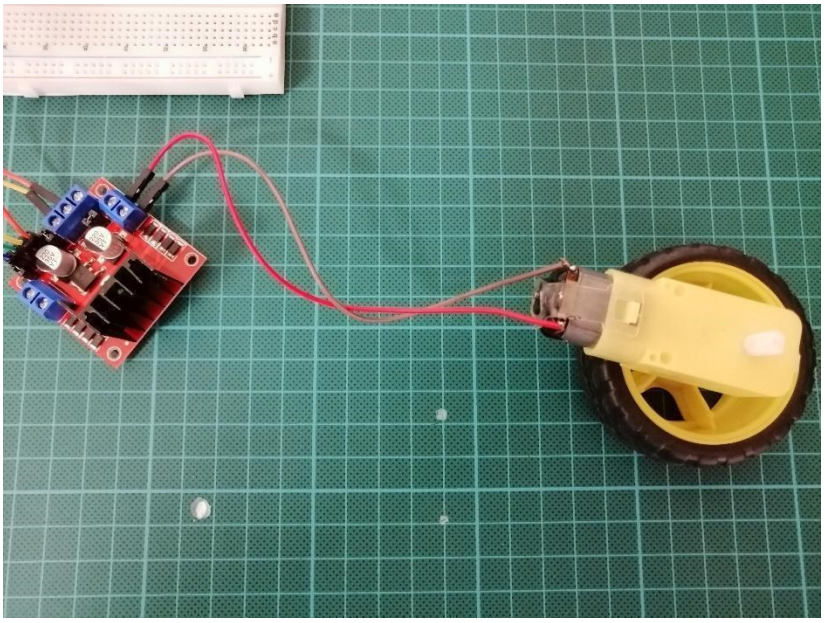
Pin Adı	**Görevi**
OUT1, OUT2	1. motor girişleri
OUT3, OUT4	2. motor girişleri
ENA	1.motor hız kontrolü
ENB	2.motor hız kontrolü
IN1, IN2	1.motor yön kontrolü
IN3, IN4	2.motor yön kontrolü
12V, GND	Motor güç beslemesi
5V	5V gerilim çıkışı



Resim 3.11. L298N motor sürücü kartı

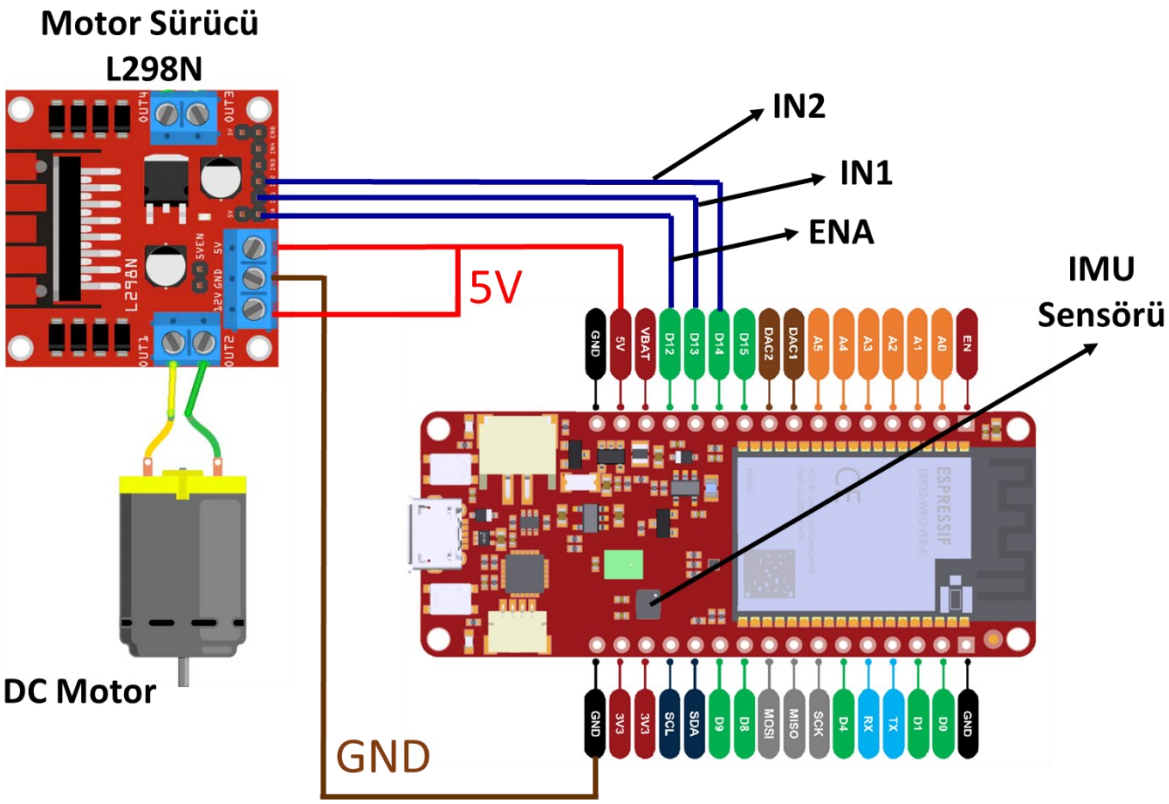
Uygulama Adımları

Adım 1: Erkek – erkek jumper kabloların birer ucu kesilerek DC motora lehimlenir. 1 adet tekerlek DC motorun miline takılır (Resim 3.12).

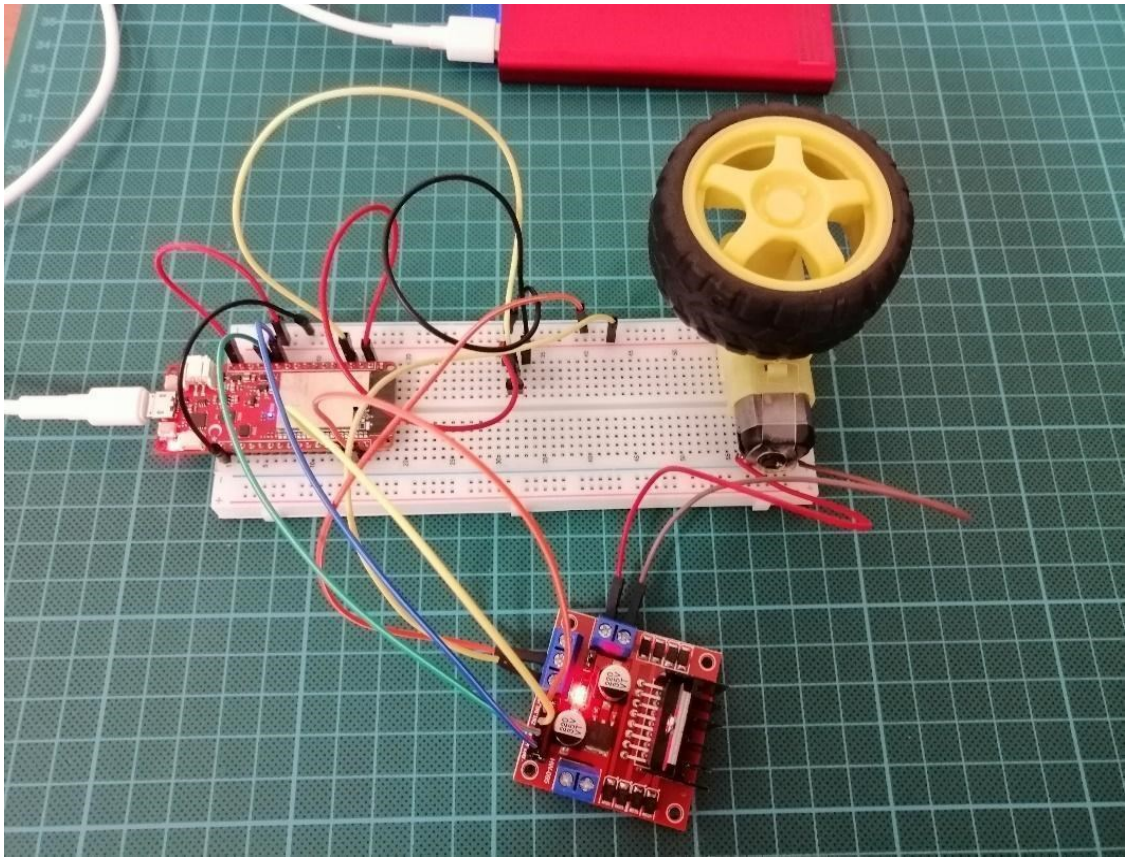


Resim 3.12. Kabloların motora lehimlenmesi ve tekerleğin montajı

Adım 2: Resim 3.13'teki devre kurulur.



Resim 3.13. Uygulama bağlantı şeması



Resim 3.14. Deneyap Geliştirme Kartı'nın, motor sürücünün ve motorun bağlantısı

Adım 3: Uygulama kodları Arduino IDE üzerinde yazılarak Deneyap Geliştirme Kartı'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

```
#include "lsm6dsm.h"
#define MOT_ENA   D12     // Motor hız kontrol pini
#define MOT_DIR1  D13     // Motor yon secme pin1
#define MOT_DIR2  D14     // Motor yon secme pin2

float angle_x=0.;                 //x açısı

int hiz=0;
LSM6DSM IMU;                      //Deneyap Geliştirme Kartı üzerinde bulunan IMU sensörü modeli
void setup() {
IMU.begin();
  pinMode(MOT_ENA, OUTPUT);       // Motor hız kontrol pini cikis olarak ayarlandi
  pinMode(MOT_DIR1, OUTPUT);      // Motor yon secme pin1 cikis olarak ayarlandi
  pinMode(MOT_DIR2, OUTPUT);      // Motor yon secme pin2 cikis olarak ayarlandi
}
void loop() {
//açı hesaplama denklemi (sqrt: karekök alma işlemi, pow: üs alma işlemi, PI: pi (π) sayısı)
angle_x = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) + pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
hiz= abs(angle_x);
   if(angle_x > 0)            // Ileri yon
  {
    digitalWrite(MOT_DIR1, HIGH);
    digitalWrite(MOT_DIR2, LOW);
  }
  else if(angle_x < 0)        // Geri yon
  {
    digitalWrite(MOT_DIR1, LOW);
```

```
    digitalWrite(MOT_DIR2, HIGH);
  }


  digitalWrite(MOT_ENA, LOW);
  delay(100/hiz);
  digitalWrite(MOT_ENA, HIGH);
  delay(100/hiz);
}
```

Adım 4: Kodlar yazılıp karta yüklendikten sonra Deneyap Geliştirme Kartı'nın takılı olduğu breadboard'un eğimi değiştirilir ve servo motora bağlı olan ibrenin eğime bağlı olarak değiştiği görülür.

2. TASARLA VE ÜRET

2.1. Tasarla: Denge çubuğu mekanizması

Bu ders kapsamında öğrencilerden bir denge çubuğu tasarlaması beklenmektedir. Orta kısmından motora sabitlenmiş bir çubuğun sürekli yere paralel halde tutulması, dış müdahaleler sonucunda açısı değişse bile tekrar paralel konuma gelmesi gerekmektedir. Tasarlama aşamasında aşağıdaki kriterler dikkate alınmalıdır;

- Motorun açısı değiştirilse bile çubuk yine yere paralel hale gelmelidir. Bunun sağlanabilmesi için çubuğun her iki ucuna ultrasonik sensör bağlanarak çubuğun uçlarının yer ile arasındaki mesafe ölçülebilir.
- IMU sensör ile sisteme ek bilgi sağlanabilir.
- Çubuk, 3 boyutlu yazıcı ile üretilebileceği gibi abeslangların uç uca eklenmesi ile de oluşturulabilir.

2.2. Üret: Denge çubuğu mekanizması

Bir önceki adımda tasarlanan denge çubuğu mekanizmasının üretimi yapılır. Mekanizmanın testi için düz bir yüzey seçilir ve çubuğa yapılacak dış müdahalelere rağmen sürekli yüzeye paralel durduğu gözlenir. Motorun açısı değiştirildiğinde de yine çubuk yere paralel konuma gelmelidir.

3. DEĞERLENDİR

Bu hafta yapılan uygulamalarda ultrasonik sensör üzerinden mesafe ölçümü, IMU sensörü ile DC motor kontrolü ve zil kullanımı gösterilmiştir.

Ders sonunda öğrencilerle bu hafta yapılan uygulamalar hakkında konuşulur ve aşağıdaki sorular üzerinden tartışmalar gerçekleştirilir.

- Bugünkü uygulamada ultrasonik mesafe sensörünün çalışma mantığı açıklandı ve kullanıldı. Sizce kızılötesi mesafe sensörünün çalışma prensibi nasıldır?
- Tasarım süreci ile ilgili bir problem yaşadınız mı? Yaşadıysanız bunu nasıl çözdünüz?
- Devrelerin fiziki kurulumunda zorlandığınız yerler oldu mu?
- Bu derste bahsi geçen Ultrasonik ve IMU sensörlü sistemlerin pratik hayattaki uygulamaları (veya endüstriyel uygulamaları) neler olabilir? Örnek uygulamalar:
    - Şu anki piyasada mevcut araçların ekseriyetinde ultrasonik sensörler kullanılmaktadır. Bu sensörler sayesinde, aracın etrafında belli bir alanda herhangi bir nesnenin olup olmadığı kolayca tespit edilir. Araçlardaki sistemler, sürücüyü sesli olarak bilgilendirmektedir. Bu sistemler, ayrıca otonom araçlarda da kullanılmaktadır. Otonom araçlarda, kontrol sistemi bu sensörlerden aldığı verileri kullanarak aracın etrafında bir nesne olup olmadığını tespit eder. Bu sistemler, özellikle otonom park edişlerde çok faydalı olmaktadır. Yalnız, burada dikkat edilmelidir ki, bu sistemler otonom sürüş için yeterli değildir, özellikle görüntü işleme tabanlı başka sistemlere de ihtiyaç duyulur.
    - Endüstriyel robotik sistemlerin güvenlik alan kontrolünde ultrasonik sensörler kullanılabilir. İnsanların girmesi tehlikeli alanlarda uygun yerlere yerleştirilmiş bu sensörler sayesinde o alanlara bir giriş ihlali tespit edilip mekanik sistemlerin yavaşlatılması ve hatta durdurulması sağlanarak herhangi bir kazanın önüne geçilmiş olunur.
    - Yine endüstriyel robotik sistemlerde, normal işleyişlerinde endüstriyel ürünlerin konumları ile ilgili bilgiler bu sensörler aracılığıyla alınabilir. Örneğin, bir konveyör üzerinden gelen paketleri alıp bir araca yükleyecek bir robot göz önüne alınsın. Ultrasonik sensör aracılığıyla gelen paket tespit edilip, robota sinyal yollanır ve robot paketi konumundan alıp araca yüklemiş olur.
    - IMU sensörlerinin uygulama alanlarından bir tanesi, iki tekerlekli bir araç olabilir. Yakın geçmişte icat edilmiş iki tekerlekli kendi kendine dengeli elektrik skuterlerinde bu sensörler kullanılmaktadır. Alışveriş merkezlerinde bu tür araçları kullanan güvenlik personellerine rastlanmaktadır. Pek çok kişi bu araçları denemiştir. Bu araçlardan her ne kadar bir Amerikan başkanı düşmüş olsa da, bunların üzerinden düşmek çok zordur. Bu araçlara bu özelliği veren, IMU sensörleridir.
    - IMU sensörleri, günümüzde kamera gimballarında çok yaygın bir şekilde kullanılmaktadır. Bu sensörlerden alınan geri besleme sinyalleri ile motorlar

kontrol edilerek denge sistemi çalıştırılmış olur, böylece kamera görüntüsü çok net olarak alınmış olur.

- Titreşim kontrollü hedeflemeli silah sistemlerinde de IMU sensörleri kullanılmaktadır. Bir askeri helikopterin pilotunun kaskına yerleştirilmiş IMU sensörleri sayesinde pilotun kafasını çevirdiği yöne silah sistemi yönlendirilebilir ve böylece bir helikopter bir çatışmada çok etkin bir şekilde kullanılmış olur.
- IMU sensörlerinin kullanılabileceği bir başka uygulama, bir teknede motor kontrollü bir yemek masası olabilir. Teknedeki masanın altına yerleştirilmiş bir IMU sensörü sayesinde dalgalı su üzerindeki teknenin hareketleri algılanıp masaya pan-tilt hareketleri veren motorlar kontrol edilerek masanın dengeli bir şekilde konumlanması sağlanır ve böylece üzerindeki tabak, bardak vs'nin emniyetli bir şekilde üzerinde tutulması sağlanmış olur.

4. İLAVE ETKİNLİK

Uygulama 1'de zilin belli bir mesafenin altında (örneğin 10 cm) sürekli bir ses sinyali üretmesi sağlanmıştı. Zilin bu belli mesafenin azalması ile orantılı olarak sıklaşan bip sesi çıkarması için gerekli olan kod düzenlemelerini yaparak deneyiniz. Örneğin 10 cm ile 7 cm arasında saniyede 1 bip sesi, 7 cm ile 5 cm arasında saniyede 2 bip sesi, 5 cm ile 3 cm arasında saniyede 3 bip sesi, 3 cm'nin altında ise sürekli ses çıkmasını sağlayınız.

4. Bölüm: İki Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap geliştirme kartı'nın temel özelliklerini bilir.

Haftanın Kazanımları:

- Öğrenciler Deneyap Geliştirme Kartı üzerinden kumanda kolu kurulumunu ve programlamasını uygular.
- Öğrenciler Deneyap Geliştirme Kartı üzerinden servo motor kurulumunu ve programlamasını uygular.
- Öğrenciler Deneyap Geliştirme Kartı üzerinden kumanda kolu ile servo motorları kontrol eder.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde IMU sensöründen alınan geri besleme sinyalleri ile servo motorları kontrol eder.
- Öğrenciler, farklı motorların, robotun uç noktasının hareketindeki etkilerini deneyimler.
- Öğrenciler robotlarda çalışma uzayı kavramını tanımlar.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin 2 serbestlik dereceli (2 döner eklemli) bir robotun mekanik ve elektronik donanım çalışmalarını ve Deneyap Geliştirme Kartı üzerinde kumanda kolu, IMU, ışık sensörleri ve potansiyometreden alınan geri besleme sinyalleri ile kontrol etmek üzere yazılım (kodlama) çalışmalarını eğlenceli uygulamalar ile yaparak öğrenmelerini sağlamaktır.

Kullanılacak Malzemeler:

1. Deneyap Geliştirme Kartı, 1 adet
2. Kumanda kolu (Joystick), 1 adet
3. LDR sensörü, 3 adet
4. Potansiyometre, 1 adet
5. Servo Motor, 2 adet
6. L-köşebent parçası (Motorları birbirlerine monte etmek için), 1 adet
7. Abeslang çubuklar, en az 2 adet
8. Breadboard, 1 adet
9. Farklı renklerde erkek-dişi ve erkek-erkek jumper kabloları, 30+ adet
10. Mikro USB kablosu
11. Harici 5V USB Batarya.

12. Çift taraflı silikon bant ve makas (buna başka bir seçenek olarak, bir silikon tabancası ile yapıştırıcı silikon malzeme kullanımı yönüne gidilebilir).
13. El feneri (cep telefonunun feneri de kullanılabilir), 1 adet

Haftanın İşlenişi:

Gözle: Deneyap Geliştirme Kartı üzerinde ilgili modüllerin bağlantılarının ve ilgili programlama kodlarının öğrencilere gösterilmesi.

Uygula: Yönergeye uygun şekilde tüm montaj ve bağlantıların yapılması, kodların yazılması

Tasarla: İlgi çekici ve eğlenceli bir uygulamanın donanımının tasarlanması ve programının bu uygulamaya uygun olarak yazılması.

Üret: Tasarlanan yapının gerçekleştirilmesi.

Değerlendir: Öğrenciler ile bu uygulama konusunda tartışma ve değerlendirmelerin yapılması.

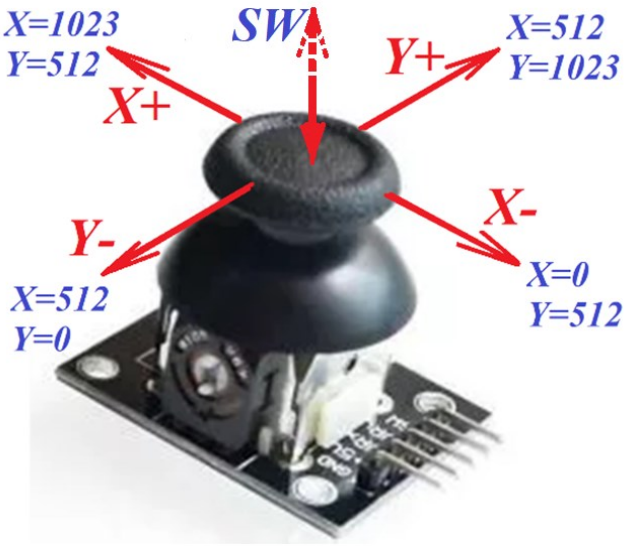
1. GÖZLE VE UYGULA

1.1. Gözle: Kumanda Kolu Nedir?

Dersin başlangıcında eğitmen, öğrencilerin 2 serbestlik dereceli robotlar ile ilgili tartışmalarını, önceki haftalardaki derslerin konularını hatırlamalarını sağlamalıdır. İki serbestlik dereceli robotların yapısı ve kullanım alanları üzerine öğrencilerin beyin jimnastiği yapmaları, onların derse ilgi ve katılımlarını artıracaktır.

Bu bölümdeki uygulamalarda, kumanda kolu, IMU, ışık sensörü ve potansiyometre kullanımı söz konusudur. Eğitmen, önceki bölümlerde bahsedilmiş olan IMU, ışık sensörü ve potansiyometre kullanımları ile ilgili öğrencilere kısa hatırlatmalar yapmalıdır.

Uygulamada kullanılacak kumanda kolu modülü, Resim 4.1’ de gösterilmiştir. Piyasada genellikle XY kumanda kolu modülü olarak da adlandırılmaktadır. Başta Arduino projeleri olmak üzere, oyun arayüzleri, robotik kol ve motor kontrolü uygulamalarında kullanılmaktadır.



Resim 4.1. XY Kumanda kolu Modülü

Kumanda kolu üzerinde 2 adet potansiyometre ve 1 adet buton bulunmaktadır. Kumanda kolu ileri geri ve sağa sola itilerek bu potansiyometre değerlerinde değişim sağlanır. Ayrıca 5 adet de pin bulunmaktadır. Bunlar; GND (topraklama bağlantısı), +5V (güç bağlantısı), VRx (X eksenindeki hareket değerini veren analog çıkış), VRy (Y eksenindeki hareket miktarını veren analog çıkış) ve SW (buton sinyal çıkışı)'dır.

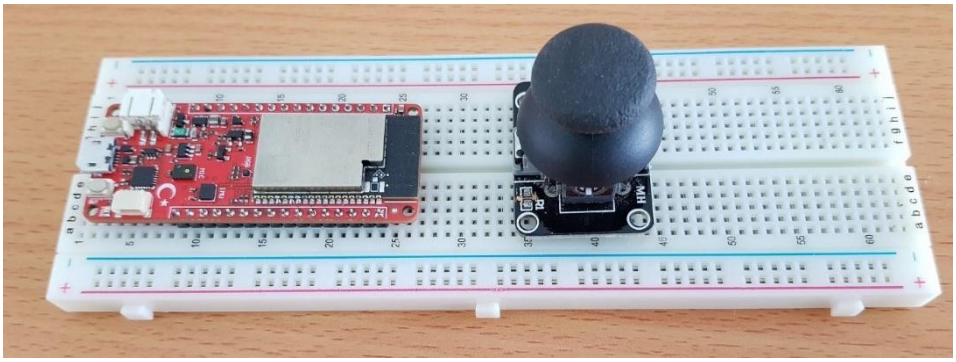
1.2. Uygula: Deneyap Geliştirme Kartı Üzerinde Kumanda Kolu Çalıştırılması

Bu uygulamada X ve Y yönlerinde hareket ettirilebilen ve üzerinde buton bulunan bir kumanda koluin Deneyap Geliştirme Kartı'na bağlantıları yapılacak ve kumanda kolu değerlerinin serial monitör üzerinde gösterilecektir. İleriki bölümlerde kumanda kolu sıklıkla kullanılacak olup, bağlantıları benzer şekilde yapılacaktır.

Adım 1: Deneyap Geliştirme Kartı Breadboard üzerine yerleştirilir. Resim 4.2'de gösterildiği gibi, kumanda kolu tabanına çift taraflı yapışkan silikon bantlar yerleştirilir ve breadboard üzerine yapıştırılır. Resim 4.2'den farklı olarak, silikon bant lehim çıkıntılarının üzerine değil de, iki kat olarak ortadaki boşluğa da yerleştirilebilir. Bu uygulamaya farklı bir seçenek olarak da, kumanda kolu tabanına silikon tabancası ile silikon malzemenin sıkılması da eğitmenin tercihine bırakılmıştır.

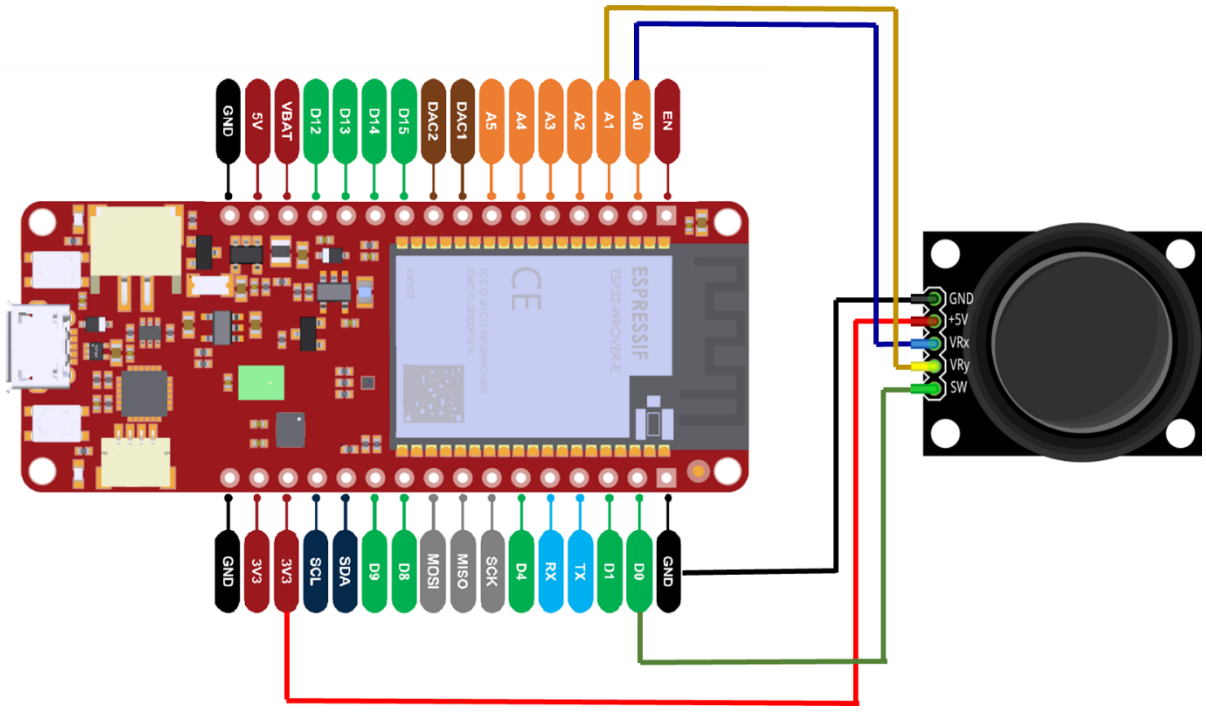


Resim 4.2. Çift taraflı silikon bant, makas ve kumanda kolu tabanında çift taraflı yapışkan silikon bant



Resim 4.3. Breadboard üzerine konumlandırılmış kumanda kolu.

Adım 2: Uygulama için gerekli Deneyap Geliştirme Kartı ve kumanda kolu bağlantıları Resim 4.4'te verilmiştir. Kumanda kolunun güç bağlantıları (+5V ve GND) Deneyap Geliştirme Kartı'nın 3V3 ve GND girişlerine yapılacaktır. Deneyap Geliştirme Kartı'nın analog girişleri 3.3V'a kalibre edildikleri için kumanda kolu 5V yerine 3.3V ile beslenmelidir. Kumanda kolunun VRx ve VRy çıkışları Deneyap Geliştirme Kartı'nın A0 ve A1 girişlerine, SW çıkışı ise D0 girişine bağlanır.



Resim 4.4. Kumanda kolu pinlerinin Deneyap Geliştirme Kartı bağlantıları

Adım 3: Deneyap Geliştirme Kartı micro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve Deneyap Geliştirme Kartı'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

Uyarı:
Yükleme esnasında kart ile bağlantı yapılamıyorsa, bağlantı anında kartın “Reset” butonuna bir kere basılmalıdır.

Programın en başında Deneyap Geliştirme Kartı’nın kütüphanesi tanımlanır. Ardından kumanda kolu butonunun dijital pini ve X-Y hareketlerinin analog pinleri için değişken tanımlamaları yapılır.

```
#include "deneyap.h"
#define SW_pin D0
#define X_pin A0
#define Y_pin A1
```

Program içerisinde kullanılacak değişkenler de tanımlanır.

```
int X_Val, Y_Val, Dig_Val;
```

Başlangıç düzenlemeleri aşağıdaki gibi yapılır. Değişkenlerin ekrana yazdırılacağı Serial Monitör örnekleme frekansı ve analog/dijital pinlerin giriş, çıkış ayarlamaları yapılır.

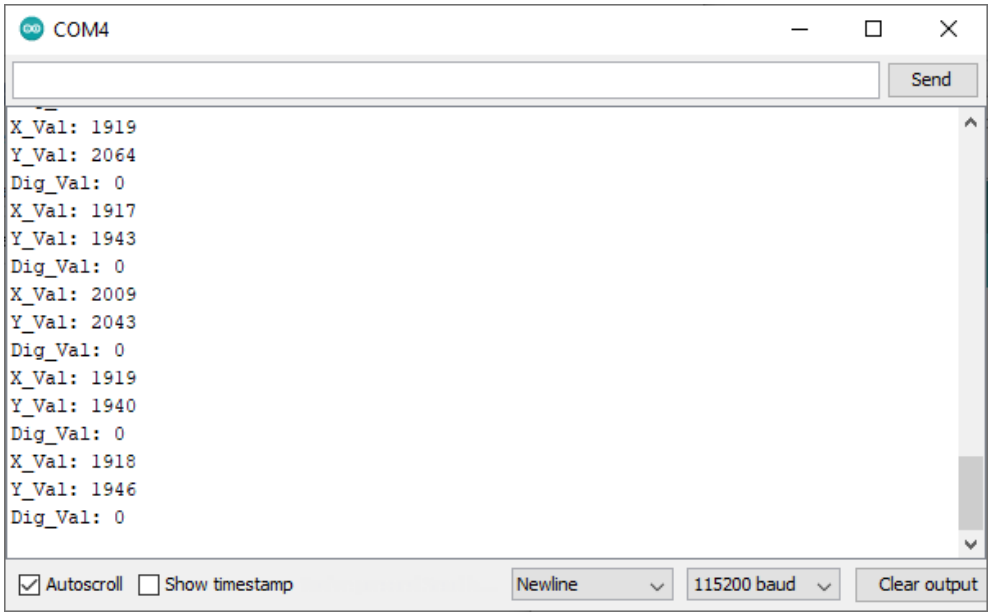
```
void setup ()
{
  Serial.begin(115200); // Seri portun 115200 baudrate hızında başlatılması.
  pinMode(SW_pin, INPUT_PULLUP); // Switch (anahtar) değişkeni mod’u HIGH’ e
ayarlanır.  Butona basıldığında LOW değerini alır.
   digitalWrite(SW_pin, HIGH);
   // kumanda kolu eksen değişken değerlerinin alınacağı 2 pin’in INPUT (giriş) olarak
tanımlanması.
  pinMode(X_pin, INPUT);
  pinMode(Y_pin, INPUT);
}
```

Kumanda kolu fonksiyonlarını algılamak için dijital ve analog girişlerden elde edilen değerlerin genel döngü (loop) içerisinde atama ve ekrana yazdırma işlemleri yapılır. Aşağıdaki kod, ekrana değerler yazdırılması için uygundur. Bununla birlikte, istenildiğinde, aşağıdaki koddan elde edilen değerlerin “Serial Plot”dan da izlenmesi mümkündür. Bu, eğitmen ve öğrencilerin tercihine bırakılmıştır.

```
void loop ()
{
  X_Val = analogRead(X_pin);
  Y_Val = analogRead(Y_pin);
  Dig_Val =digitalRead(SW_pin);
  Serial.print("X_Val: ");
```

```
        Serial.println(X_Val);
        Serial.print("Y_Val: ");
        Serial.println(Y_Val);
        Serial.print("Dig_Val: ");
        Serial.println(Dig_Val);
        delay(100); // Burada bu delay (gecikme) fonksiyonu içindeki milisaniye değeri
```
artırılarak serial monitor'deki akış yavaşlatılabilir ve öğrencilerin kumanda kolu değerlerini daha rahat okumaları sağlanabilir.
```
    }
```

Adım 4: Serial Monitor ekranında kumanda kolu fonksiyonları gözlemlenmelidir (Resim 4.5).



Resim 4.5. Serial monitorde kumanda kolu değerlerinin gözlemlenmesi

Uyarı:
Serial monitör üzerinde anlamsız karakterler görünmesi durumunda sağ alt kenarda bulunan baud ayarından uygun baud değeri seçilmelidir.

1.3. Uygula: Deneyap Geliştirme Kartı ile Çift Servo Motor Kontrolü

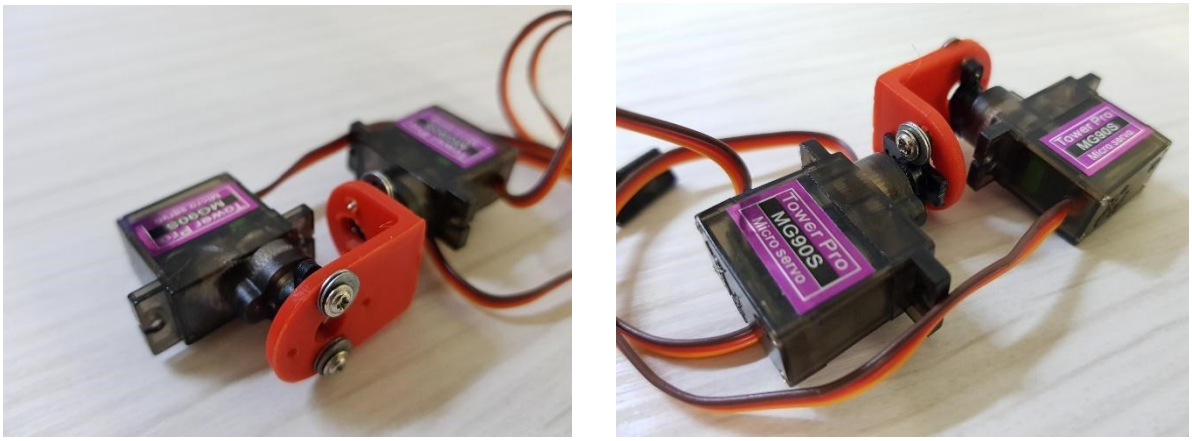
Adım 1: Deneyap Geliştirme Kartı Breadboard üzerine yerleştirilir. Resim 4.6'da gösterilen

- 2 adet servo motor,
- 1 adet gülen yüz resmi çizilmiş küçük bir etiket,
- servo motor T-plastik parçası,
- servo motor vidaları,
- minik yıldız tornavida,
- L-profil parça

hazırlanır. Resim 4.7'de gösterildiği gibi, motorlar, L-profil parçaya vidalar ve T-plastik parça kullanılarak monte edilir. L-profil parça alüminyum parçadan bükülerek yapılacağı gibi bir 3B yazıcıdan da basılıp kullanılabilir.



Resim 4.6. Çift motor düzeneğinde kullanılan parçalar



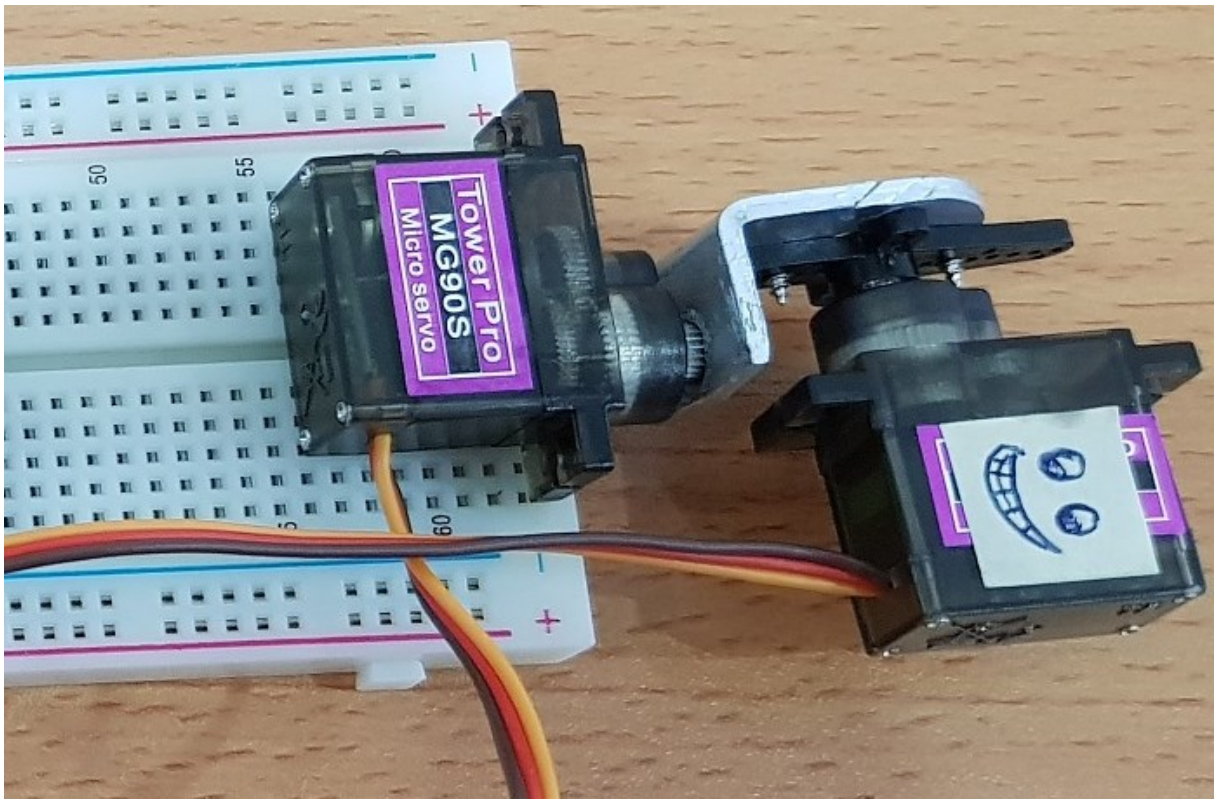
Resim 4.7. 3B yazıcı ile basılmış L-profil kullanılarak hazırlanmış çift motor düzeneği

Adım 2: Motorlardan birinin alt kısmına çift taraflı yapışkanlı silikon bant yapıştırılır. Diğer motorun üst kısmına ise üzerinde gülen yüz bulunan etiket resimde gösterildiği gibi yapıştırılır (Resim 4.8). Burada da isteğe bağlı olarak, çift taraflı silikon bant yerine silikon tabancası kullanılabilir.



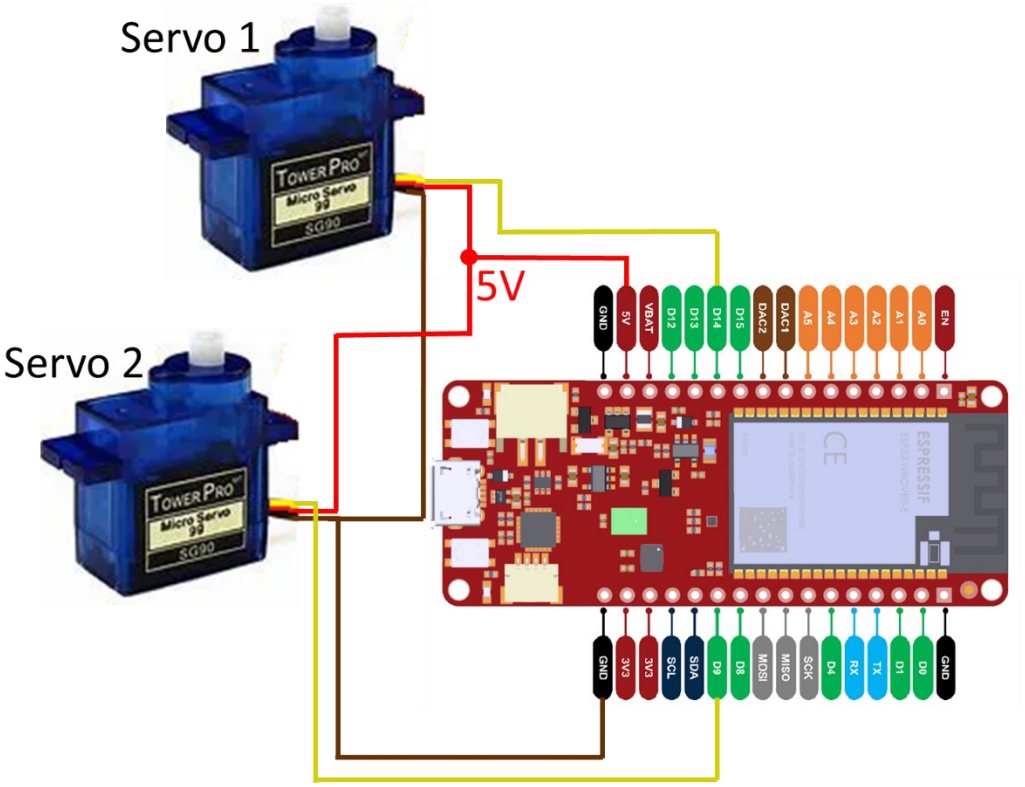
Resim 4.8. Çift taraflı silikon bant ve etiketin yapıştırılması

Adım 3: Çift motor düzeneğinin silikon (veya silikonlandırılmış) bantlı yüzeyi breadboard'un diğer boş tarafına yerleştirilir ve motor üzerine nazikçe baskı yapılarak bir süre beklenir (Resim 4.9).

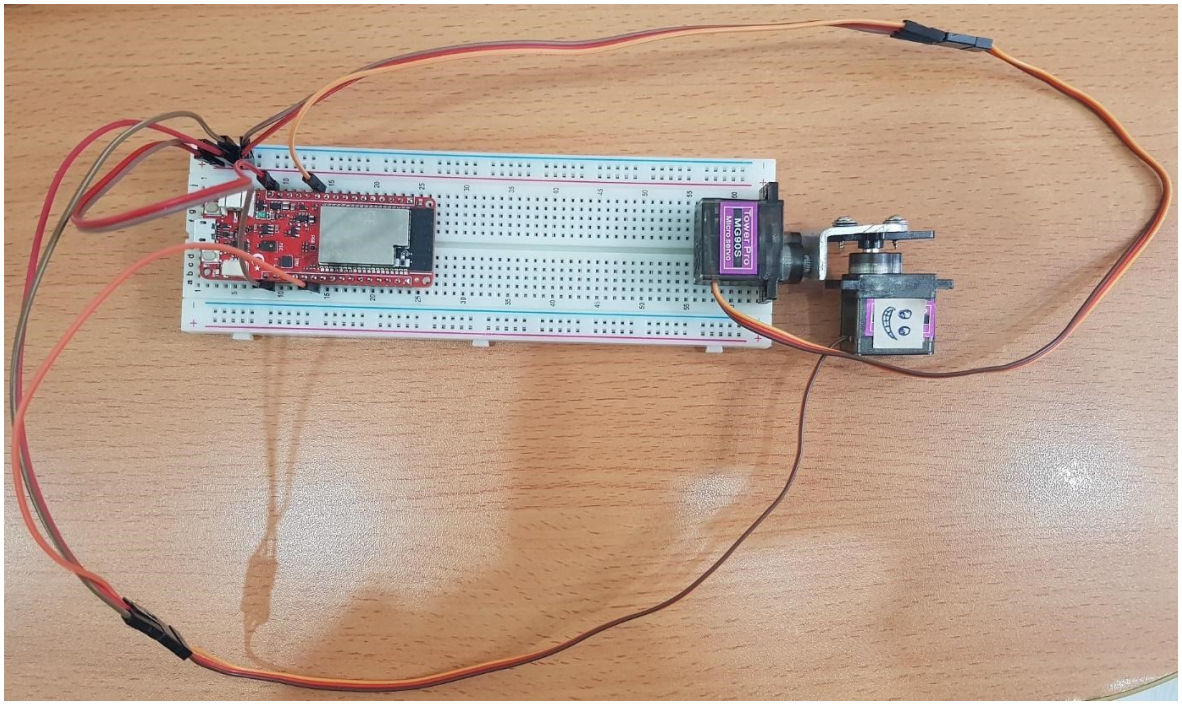


Resim 4.9. Çift motor düzeneğinin breadboard üzerine yerleştirilmesi

Adım 5: Uygulama için gerekli Deneyap Geliştirme Kartı ve servo motor bağlantıları Resim 4.10'da verilmiştir. Servo motorların güç bağlantıları (+5V ve GND) Deneyap Geliştirme Kartının 5V ve GND girişlerine yapılacaktır. Motorların kontrol girişleri de Deneyap Geliştirme Kartının D9 ve D14 pinlerine bağlanacaktır (Resim 4.11).



Resim 4.10. Çift servo motorun Deneyap Geliştirme Kartı bağlantıları



Resim 4.11. Servo motorların Deneyap Geliştirme Kartı bağlantısı

Adım 6: Deneyap Geliştirme Kart bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include <Deneyap_Servo.h>
#define SERVOPIN1 D14
#define SERVOPIN2 D9

Servo myservo1, myservo2; // Servo motorların tanımlanması.

int pos = 0; //İlk pozisyon.  Her 2 motor için ortak değişken.

// Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
void setup()
{
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1);
}
```

Servo motorların kontrol girişlerine kontrol sinyallerin gönderilmesi işlemi genel döngü (loop) içerisine yazılan kodlar ile yapılır. Aşağıdaki kod yazılarak, motorlara tek bir pos değişkeni ile seri bir demo hareket yaptırılmaktadır.

```
void loop()
{
   for (pos = 0; pos <= 180; pos += 1) {
          // Motorlara pos değerinin yollanması
          myservo1.write(pos);
          myservo2.write(pos);
          delay (10); // Her bir döngüde 10 milisaniye bir gecikme ilave edilmesi.
    }
    for (pos = 180; pos >= 0; pos -= 1) {
          // Motorlara pos değerinin yollanması
          myservo1.write(pos);
          myservo2.write(pos);
          delay (10); // Her bir döngüde 10 milisaniye bir gecikme ilave edilmesi.
   }
}
```

Bilgi Notu:
Motorların kararlı çalışabilmesi için her konum bilgisi gönderimi işleminden sonra 10 milisaniye bekleme verilmiştir. Bu bekleme değerinin seviyesine göre motorlarda hız kontrolü de yapılabilir.

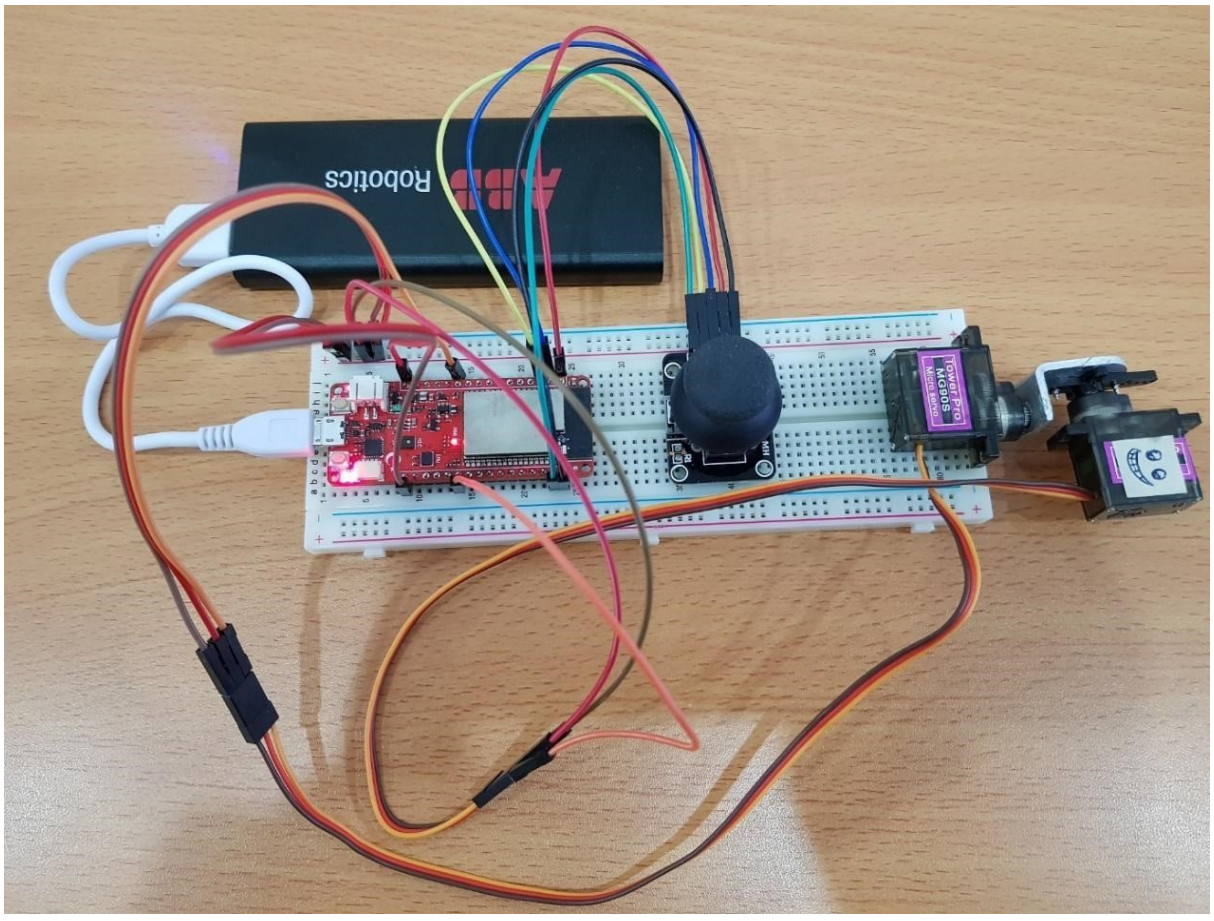
Adım 7: Programın Deneyap Geliştirme Kartına yüklemesi bittiğinde, her iki motorun da 0-180°'lik açı aralığında sürekli olarak ileri-geri döndüğü gözlemlenmelidir.

1.4. Uygula: İki Motorlu Bir Sistemin Kumanda Kolu ile Kontrolü

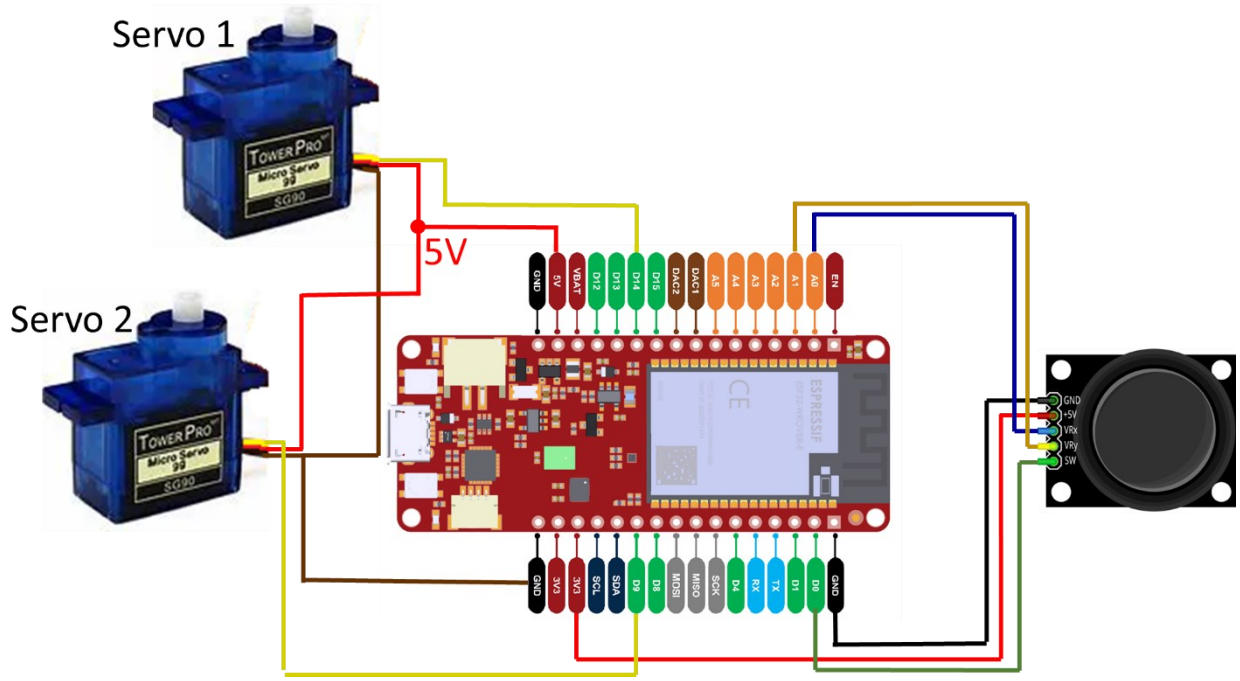
Bu uygulamada, hemen önceki iki uygulamada yapılan sistemlerin bir araya getirilmesi ve çalıştırılması aşamaları verilmiştir. Bir kumanda kolu ile 2 servo motorun hareketi kontrol edilecektir.

Uygulama Adımları

Adım 1: Bir önceki uygulamada gerçekleştirilen deney düzeneğine bir adet kumanda kolu eklenir (Resim 4.12). Uygulama için gerekli Deneyap Geliştirme Kartı, servo motor ve kumanda kolu bağlantıları Resim 4.13'te verilmiştir. kumanda kolunun güç bağlantıları Deneyap Geliştirme Kartının 3V3 ve GND girişlerine, VRx ve VRy çıkışları sırası ile Deneyap Geliştirme Kartının A0 ve A1 girişlerine, buton çıkışı ise D0 girişine bağlanır. Burada, harici güç kaynağı, Deneyap kartına kod yükledikten sonra sistemi bilgisayardan bağımsız herhangi bir yerde çalıştırmak için kullanılabilir. Ayrıca, bilgisayarların USB portlarından yeterli akım değerleri alınamadığı için, denemelerin harici bir güç kaynağı ile yapılması kuvvetle tavsiye edilir.



Resim 4.12. Kumanda kolu ve servo motorların bir breadboard üzerinde jumper kablolar ile Deneyap Geliştirme Kartına bağlanması ve harici bir batarya ile çalıştırılması.



Resim 4.13. Deneyap Geliştirme Kartı, servo motor ve kumanda kolu bağlantıları

Adım 2: Deneyap Geliştirme Kartı bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include "deneyap.h"
#include <Deneyap_Servo.h>
#define SERVOPIN1 D14
#define SERVOPIN2 D9
#define SW_pin D0
#define X_pin A0
#define Y_pin A1

Servo myservo1, myservo2; // Servo motorların tanımlanması

int X_Val, Y_Val; // kumanda kolu eksen değişkenleri.
int pos1=0, pos2=0; // Motorların pozisyon değişkenleri.
int dly = 10; // Genel döngüde ilave edilen gecikme için kullanılan değişken.
int stp = 1; // Motorların dönüş açısının değiştirilmesi durumunda her döngüde kullanılan adım değişkeni.  Örneğin servo 1 pozitif yöne dönsün istendiğinde her döngüde, Pos1+=stp olacaktır (kod aşağıda).
int Dig_Val = 0; // Buton değeri

// Servo motorların açı sınır değerleri.  Ne oldukları değişken isimlerinden bellidir.
int teta1min=0;
int teta1max=180;
int teta2min=0;
int teta2max=180;
```

```
void setup()
{
  // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1);

  pinMode(SW_pin, INPUT_PULLUP); // Switch (anahtar) değişkeni mod'u HIGH' e ayarlanır.  Butona basıldığında LOW değerini alır.
  digitalWrite(SW_pin, HIGH);
  // kumanda kolu eksen değişken değerlerinin alınacağı 2 pin'in INPUT (giriş) olarak tanımlanması.
  pinMode(X_pin, INPUT);
  pinMode(Y_pin, INPUT);
 }

void loop()
{
  // kumanda kolu X ve Y değişiminin okunması.
  X_Val = analogRead(X_pin);
  Y_Val = analogRead(Y_pin);
  Dig_Val=digitalRead(SW_pin);

 // kumanda kolu butonuna her basıldığında, servo motor dönüş hızlarını belirleyen dly değişkenine değerler atanması.
  if (Dig_Val<1) {
    if (dly==5) dly=10;
    else {
      if(dly==10) dly=20;
      else if (dly==20) dly=5;
    }

  delay (500);  // Butona basış tespit edildikten sonra 500 ms bir gecikme ilave edilmesi.
  }
  // Motorların limitleri dahilinde dönüşlerine izin verilmesi:
  if (X_Val<1000 && pos1>teta1min) pos1-=stp;
  if (X_Val>3000 && pos1<teta1max) pos1+=stp;
  if (Y_Val<1000 && pos2>teta2min) pos2-=stp;
  if (Y_Val>3000 && pos2<teta2max) pos2+=stp;

  // Motorlara dönüş komutlarının verilmesi.
  myservo1.write(pos1);
  myservo2.write(pos2);

 delay (dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
}
```

Burada motorların hareket hızları, döngü içerisindeki dly milisaniye bekleme değişkeninin kumanda kolu butonuna her basıldığında değiştirilmesi ile ayarlanmaktadır. Hızlı, normal ve yavaş olmak üzere üç hız seviyesi tanımlanmıştır. Hız kontrolünde, daha küçük dly değeri, daha yüksek hız anlamına gelmektedir. Burada, öğrencilere çeşitli dly değerlerini deneyimleri tavsiye edilir.

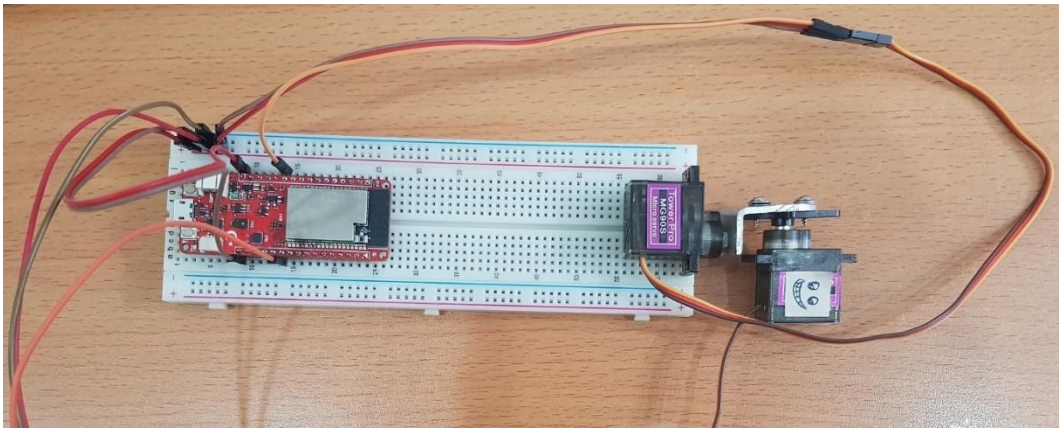
Adım 3: Programın Deneyap Geliştirme Kartına yüklemesi bittiğinde motorların kumanda kolu ile kontrolünün yapılabildiği gözlemlenmelidir.

1.5. Uygula: IMU Sensörü ile Çift Servo Motor Konum Kontrolü

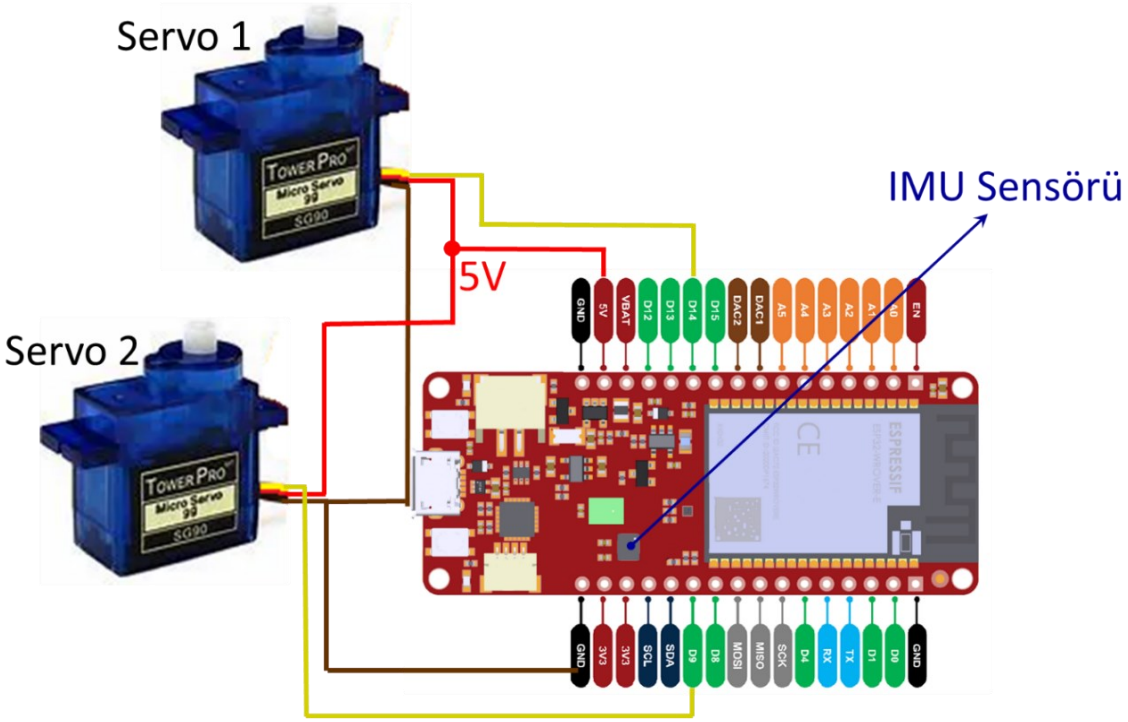
Bu bölümde Deneyap Geliştirme Kartı üzerinde bulunan IMU sensörü ile iki adet servo motorun hareket kontrolü yapılacaktır.

Uygulama Adımları

Adım 1: Bir önceki uygulamada gerçekleştirilen deney düzeneğindeki kumanda kolu bağlantıları sökülerek sadece servo motorlar bırakılır. (Resim 4.14). Uygulama için gerekli Deneyap Geliştirme Kartı ve servo motor bağlantıları Resim 4.15'te verilmiştir.



Resim 4.14. Uygulama 4 deney düzeneği



Resim 4.15. Uygulama 4 bağlantı şeması

Adım 2: Deneyap Geliştirme Kartı bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include <Deneyap_Servo.h>
#include "lsm6dsm.h"

#define SERVOPIN1 D14
#define SERVOPIN2 D9

LSM6DSM IMU; // Kart üzerindeki ivme ölçerin tanımlanması.

// Servo motorların tanımlanması
Servo myservo1;
Servo myservo2;

// X ve Y eksenlerindeki ivme değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
float accAngleX=0.;
float accAngleY=0.;
// Motorların dönüş açı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
int pos1 = 90;
int pos2 = 90;


void setup()
{
  IMU.begin();
  // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
```

```
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1);
 }

void loop()
{
// İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;

// Motorlara belirlenen X ve Y eğim açılarının uygulanması:
  pos1=90+int(accAngleX);
  pos2=90+int(accAngleY);

  myservo1.write(pos1);
  myservo2.write(pos2);
  delay(10); // Genel döngüye 10 milisaniyelik bir gecikme uygulanması.
}
```

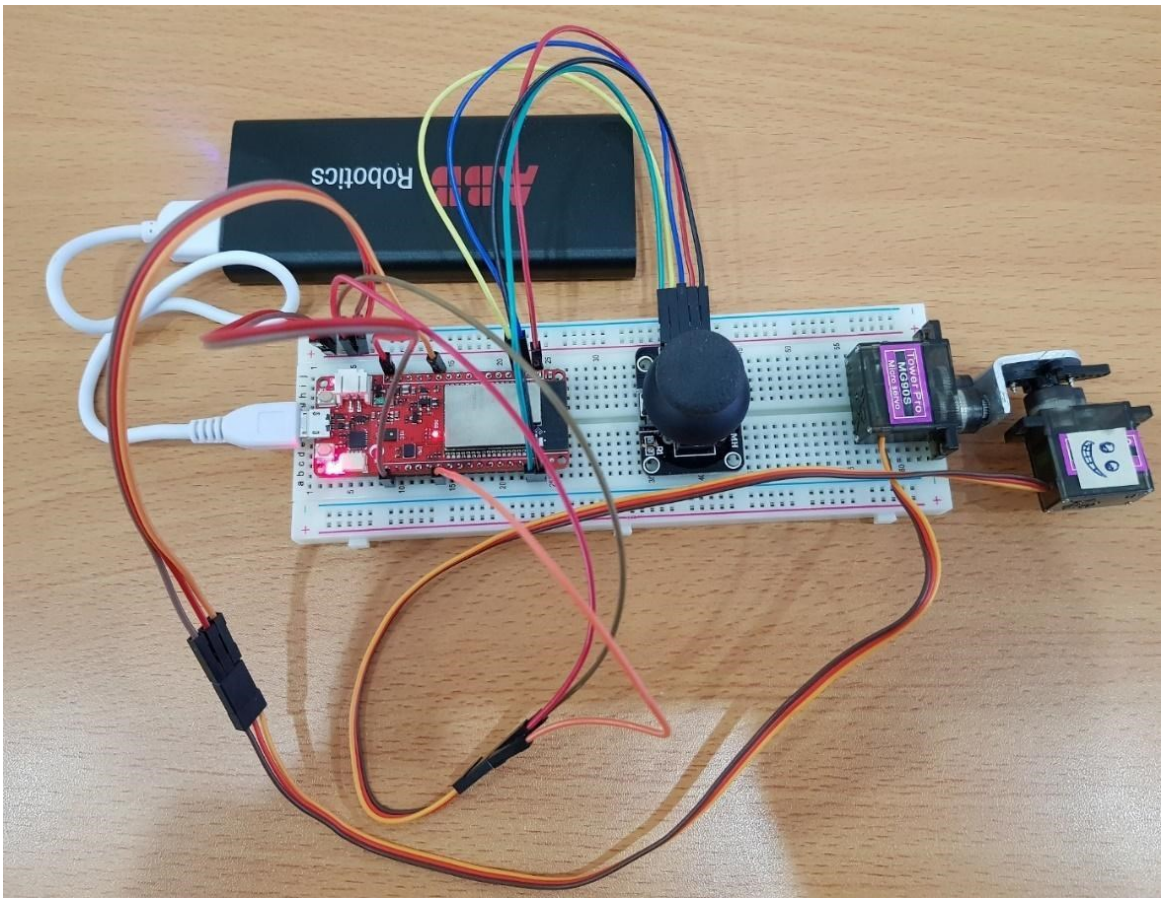
Adım 3: Programın Deneyap Geliştirme Kartına yüklemesi bittiğinde motorların IMU sensöründen gelen değerlere göre davranış gösterdiği görülecektir. Basit bir gimbal uygulaması olarak, çift motorlu mekanizmanın gülen yüzlü yüzeyi her zaman yukarı yöne bakacaktır.

1.6. Uygula: Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü

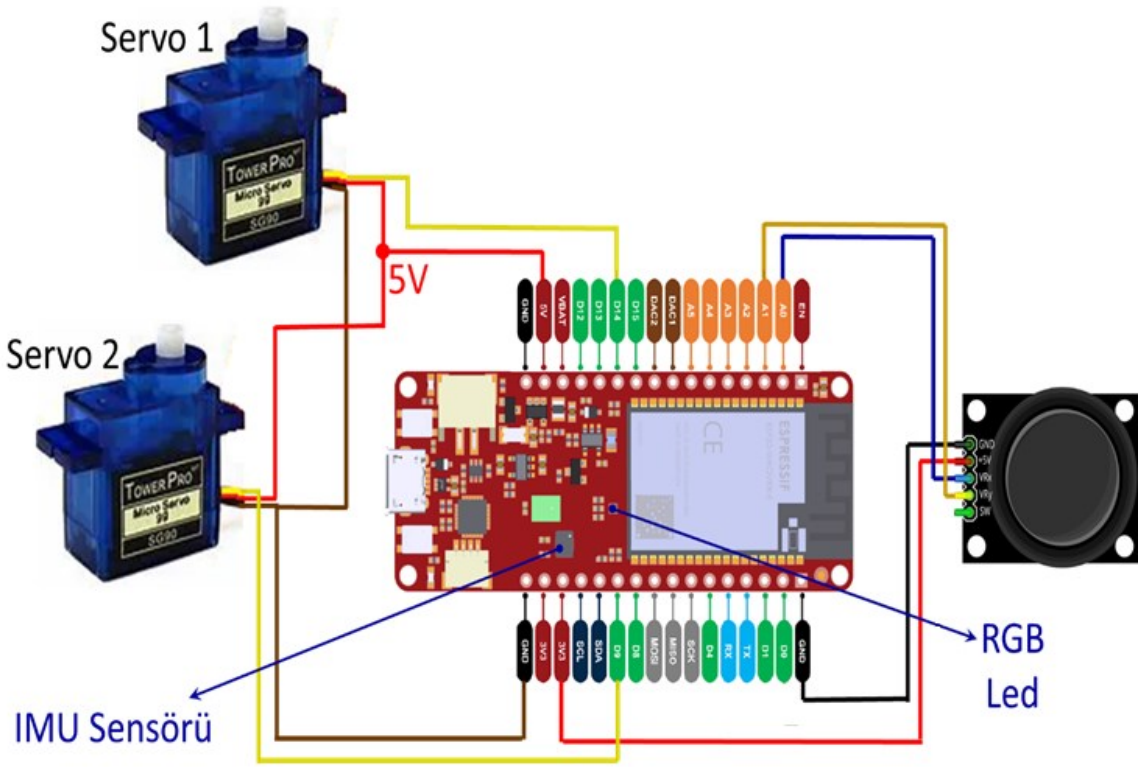
Bu uygulamada servo motorların hem kumanda kolu hem de Deneyap Geliştirme Kartı üzerindeki IMU sensörü ile kontrolü ve kart üzerindeki RGB ledlerin kullanımı gösterilecektir.

Uygulama Adımları

Adım 1: Deneyap Geliştirme Kartı, kumanda kolu ve servo motorlar breadboard üzerine yerleştirilir. Servo motorların ve Kumanda kolunun Deneyap Geliştirme Kartına bağlantıları yapılır (Resim 4.16). Uygulama için gerekli Deneyap Geliştirme Kartı ve servo motor bağlantıları Resim 4.17'de verilmiştir.



Resim 4.16. Uygulama 5 deney düzeneği



Resim 4.17. Uygulama 5 bağlantı şeması

Adım 2: Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve Deneyap Geliştirme Kartına yüklenir. Bu uygulamada kumanda kolu üzerindeki buton yerine Deneyap Geliştirme Kartı üzerinde USB portun solunda bulunan buton (BUILTIN_KEY) ve yine kartın üzerindeki renkli led kullanılacaktır.

```
#include <Deneyap_Servo.h>
#include "lsm6dsm.h"

// Dijital servo pin bağlantıları:
#define SERVOPIN1 D14
#define SERVOPIN2 D9
// kumanda kolu eksen analog bağlantı pinleri
#define X_pin A0
#define Y_pin A1

int X_Val, Y_Val;  // kumanda kolu eksen değişkenleri
int dly = 5;  // Genel döngüdeki gecikme değişkeni
int stp = 1; // Servo motorların dönüş durumunda her bir döngüdeki dönüş adımı
int X_jys=0, Y_jys=0;  // kumanda kolu eksen değişkenlerinden elde edilen motor dönüş
açıları

// Motor dönüş açı alt ve üst limitleri:
int teta1min=0;
int teta1max=180;
int teta2min=0;
int teta2max=180;

// İvme ölçerden elde edilen X ve Y eksenlerindeki açısal dönüşler:
float accAngleX=0.;
float accAngleY=0.;


LSM6DSM IMU; // IMU' nun tanımlanması

// Servo motorların tanımlanması.
Servo myservo1;
Servo myservo2;

// Motorların dönüş açı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
int pos1 = 90;
int pos2 = 90;

void setup()
{
  IMU.begin();
// Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1);
```

```
  pinMode(BUILTIN_KEY, INPUT); // Deneyap kartı üzerindeki butonun giriş olarak tanımlanması
  // kumanda kolu pinlerinin giriş olarak tanımlanması
  pinMode(X_pin, INPUT);
  pinMode(Y_pin, INPUT);

  // RGB Led pinlerinin çıkış olarak tanımlanması.
  pinMode(LEDR, OUTPUT);
  pinMode(LEDG, OUTPUT);
  pinMode(LEDB, OUTPUT);
  digitalWrite(LEDR, LOW);  //Kırmızı led açık (aktif)
  digitalWrite(LEDG, HIGH); // Yeşil led sönük (pasif)
  digitalWrite(LEDB, HIGH); // Mavi led sönük (pasif)
 }
```

İvme-ölçer verilerinin alınması, pan-tilt açı değerlerinin hesaplanması, joystik verileri, buton verileri, RGB led renk değişimleri ve motorların kontrol edilmesi işlemleri genel döngü (loop) içerisinde yapılır.

```
 void loop()
 {
  // kumanda kolu X ve Y değişiminin okunması.
  X_Val = analogRead(X_pin);
  Y_Val = analogRead(Y_pin);

  uint8_t buttonState = digitalRead(BUILTIN_KEY);       // Buton durumu

   // Kart üzerindeki butona her basıldığında, servo motor dönüş hızlarını belirleyen dly değişkenine değerler atanması ve buna göre kart üzerindeki RGB led'lerinin değiştirilmesi.
  if(buttonState == LOW)                    // Butona basılma durumu
  {
   if (dly==5) {
    dly=10;
    digitalWrite(LEDR, HIGH);
    digitalWrite(LEDG, LOW);              //Yeşil led aktif
   }
   else {
    if(dly==10) {
     dly=20;
     digitalWrite(LEDG, HIGH);
     digitalWrite(LEDB, LOW);  //Mavi led aktif
    }
    else if (dly==20) {
     dly=5;
     digitalWrite(LEDR, LOW);  //Kırmızı led aktif
     digitalWrite(LEDB, HIGH);
    }
```

```
      }
      delay(300); // Butona basış tespit edildikten sonra 300 ms bir gecikme ilave edilmesi.
    }
    // kumanda kolundan elde edilen dönüş açılarının limitler dahilinde uygulanması.
    if (X_Val<1000 && pos1>teta1min) X_jys-=stp;
    if (X_Val>3000 && pos1<teta1max) X_jys+=stp;
    if (Y_Val<1000 && pos2>teta2min) Y_jys-=stp;
    if (Y_Val>3000 && pos2<teta2max) Y_jys+=stp;

    // İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
    accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
    accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;

    // Motor dönüş açıları:
    pos1=90+int(accAngleX)+X_jys;
    pos2=90+int(accAngleY)+Y_jys;

    myservo1.write(pos1);
    myservo2.write(pos2);

    delay(dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
  }
```

Alternatif Bir Programlama Yöntemi: Servo Motor Kontrolünde "writeMicroseconds" Fonksiyonunun Kullanımı

Bundan önceki uygulamalarımızda servo motor kontrolü için "write" fonksiyonunun kullanımı gösterilmişti. Bu bölümde, aynı işlevi gören farklı bir fonksiyon olan "writeMicroseconds" fonksiyonunun kullanımı gösterilecektir. "write" ve "writeMicroseconds" fonksiyonları arasındaki fark, "write" fonksiyonuna girdiğimiz açı değeri doğrudan derece cinsinden iken, "writeMicroseconds" fonksiyonuna girilen değer 0 ile 3000 arasında olmaktadır. Burada 3000 değeri 180 dereceye karşılık gelmektedir. Örneğin 45 derecelik bir açı için, "writeMicroseconds" fonksiyonuna;

$$45 \times \left(\frac{3000}{180}\right) = 750$$

değerinin girilmesi gerekmektedir. Dolayısıyla burada $\left(\frac{3000}{180}\right)$ değeri "writeMicroseconds" fonksiyonunu kullanırkenki kalibrasyon katsayısıdır. 5 numaralı uygulamanın kodlarını "writeMicroseconds" fonksiyonuna göre tekrar düzenleyip çalıştıralım.

```
#include <Deneyap_Servo.h> // Eğer bu kütüphane çalışmaz ise <ESP32Servo.h>
```
kütüphanesi denenebilir.

```
#include "lsm6dsm.h"

// Dijital servo pin bağlantıları:
#define SERVOPIN1 D14
#define SERVOPIN2 D9
// kumanda kolu eksen analog bağlantı pinleri
#define X_pin A0
#define Y_pin A1

int X_Val, Y_Val; // kumanda kolu eksen değişkenleri
int dly = 5; // Genel döngüdeki gecikme değişkeni
// writeMicroseconds fonksiyonuna girilen değer normal açı değerinden çok yüksek olduğundan stp değeri 10 kat fazla girilir.
int stp = 10; // Servo motorların dönüş durumunda her bir döngüdeki dönüş adımı
int X_jys=0, Y_jys=0; // kumanda kolu eksen değişkenlerinden elde edilen motor dönüş açıları

// Motor dönüş açı alt ve üst limitleri:
int teta1min=0;
int teta1max=3000;              // 180 dereceye karşılık gelmektedir.
int teta2min=0;
int teta2max=3000;
float accAngleX=0.;
float accAngleY=0.;

LSM6DSM IMU;

// Servo motorların tanımlanması.
Servo myservo1;
Servo myservo2;

// Motorların dönüş açı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
int pos1 = 1500;                        // 90 dereceye karşılık gelmektedir.
int pos2 = 1500;

void setup()
{
  IMU.begin();
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1); //Eğer <ESP32Servo.h> kütüphanesi kullanırsanız, bu satırda parantez içerisindeki 2'inci parametreyi silin, tek parametre yeterlidir.

  pinMode(BUILTIN_KEY, INPUT); // Deneyap kartı üzerindeki butonun giriş olarak tanımlanması
  pinMode(X_pin, INPUT);
  pinMode(Y_pin, INPUT);
```

```
  // RGB Led pinlerinin çıkış olarak tanımlanması.
  pinMode(LEDR, OUTPUT);
  pinMode(LEDG, OUTPUT);
  pinMode(LEDB, OUTPUT);

  digitalWrite(LEDR, LOW);  //Kırmızı led açık (aktif)
  digitalWrite(LEDG, HIGH); // Yeşil led sönük (pasif)
  digitalWrite(LEDB, HIGH); // Mavi led sönük (pasif)
 }

void loop()
{
  // kumanda kolu X ve Y değişiminin okunması.
  X_Val = analogRead(X_pin);
  Y_Val = analogRead(Y_pin);

  uint8_t buttonState = digitalRead(BUILTIN_KEY); // Buton durumu

   // Kart üzerindeki butona her basıldığında, servo motor dönüş hızlarını belirleyen dly
değişkenine değerler atanması ve buna göre kart üzerindeki RGB led'lerinin değiştirilmesi.
  if(buttonState == LOW)
  {
   if (dly==5) {
    dly=10;
    digitalWrite(LEDR, HIGH);
    digitalWrite(LEDG, LOW); //Yeşil led aktif
   }
   else {
    if(dly==10) {
     dly=20;
     digitalWrite(LEDG, HIGH);
     digitalWrite(LEDB, LOW);  //Mavi led aktif
    }
    else if (dly==20) {
     dly=5;
     digitalWrite(LEDR, LOW); //Kırmızı led aktif
     digitalWrite(LEDB, HIGH);
    }
   }
   delay(300); // Butona basış tespit edildikten sonra 300 ms bir gecikme ilave edilmesi.
  }
  // kumanda kolundan elde edilen dönüş açılarının limitler dahilinde uygulanması.
  if (X_Val<1000 && pos1>teta1min) X_jys-=stp;
  if (X_Val>3000 && pos1<teta1max) X_jys+=stp;
  if (Y_Val<1000 && pos2>teta2min) Y_jys-=stp;
  if (Y_Val>3000 && pos2<teta2max) Y_jys+=stp;
```

```
        // İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
        accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
        accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;

        // Motor dönüş açıları (writeMicroseconds fonksiyonu için revize edildi):
        pos1=1500+int(accAngleX*3000./180.)+X_jys;
        pos2=1500+int(accAngleY*3000./180.)+Y_jys;

        // Motorlara dönüş komutlarının verilmesi.
        myservo1. writeMicroseconds(pos1);
        myservo2. writeMicroseconds(pos2);

        delay(dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
      }
```

Uygula: Donanım ve Yazılımın Denenmesi (Test Edilmesi):

Program Deneyap Geliştirme Kartına yüklendikten sonra motorların IMU sensörü ve kumanda kolu aktivitelerine göre davranış gösterdiği görülecektir. Bu uygulamada, kumanda kolu ile pan-tilt mekanizması hareket ettirilirken, IMU sensörü ile de duruşu dengelenmektedir. Motorların hızları, döngü içerisindeki "dly" bekleme değişkeninin Deneyap Geliştirme Kartı üzerinde bulunan butona her basıldığında değiştirilmesi ile ayarlanmaktadır. Hızlı, normal ve yavaş olmak üzere üç hız seviyesi tanımlanmıştır.

2. TASARLA VE ÜRET

2.1. Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü

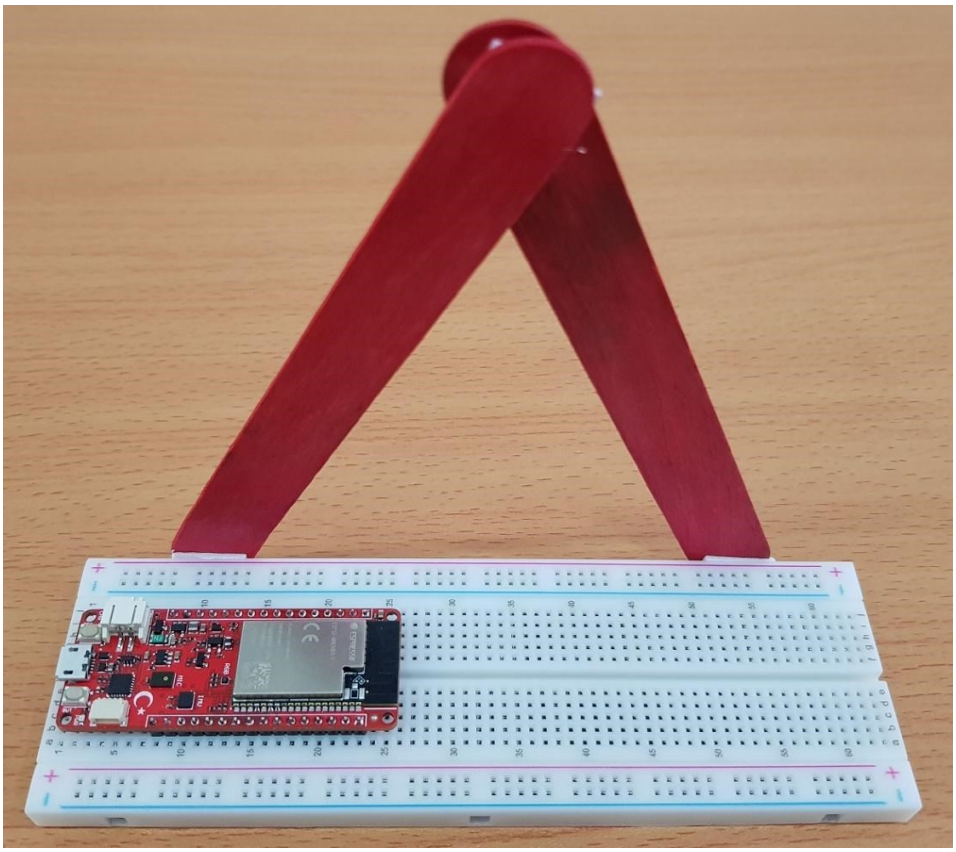
Bu uygulamada ışık sensörlerinden alınan ortam ışığı değerlerine göre 2 motorlu bir sistemin hareket kontrolü yapılacaktır. Böyle bir sistem, çeşitli amaçlar için kullanılabilir. Örneğin:

- Güneş enerjisi sistemlerinin verimini artırmak için panellerin güneşe yöneltilmesi uygulamasında,
- Güvenlik amaçlı olarak istenmeyen bir gölge oluşması durumunu algılama ve mekanizma üzerine bağlı bir kameranın bu yöne döndürülmesi uygulamasında,
- Askeri amaçlı olarak, karanlık ortamda meydana gelen ani ışık oluşumlarını (örneğin silah ateşlenmesi) ve konumlarını tespit etme uygulamalarında kullanılabilir.

Bu kısımdaki tasarım ve üretim adımları eğitmen için verilmiştir. Öğrencilerden bu tip bir sistemi tasarlayıp üretmeleri istenir.

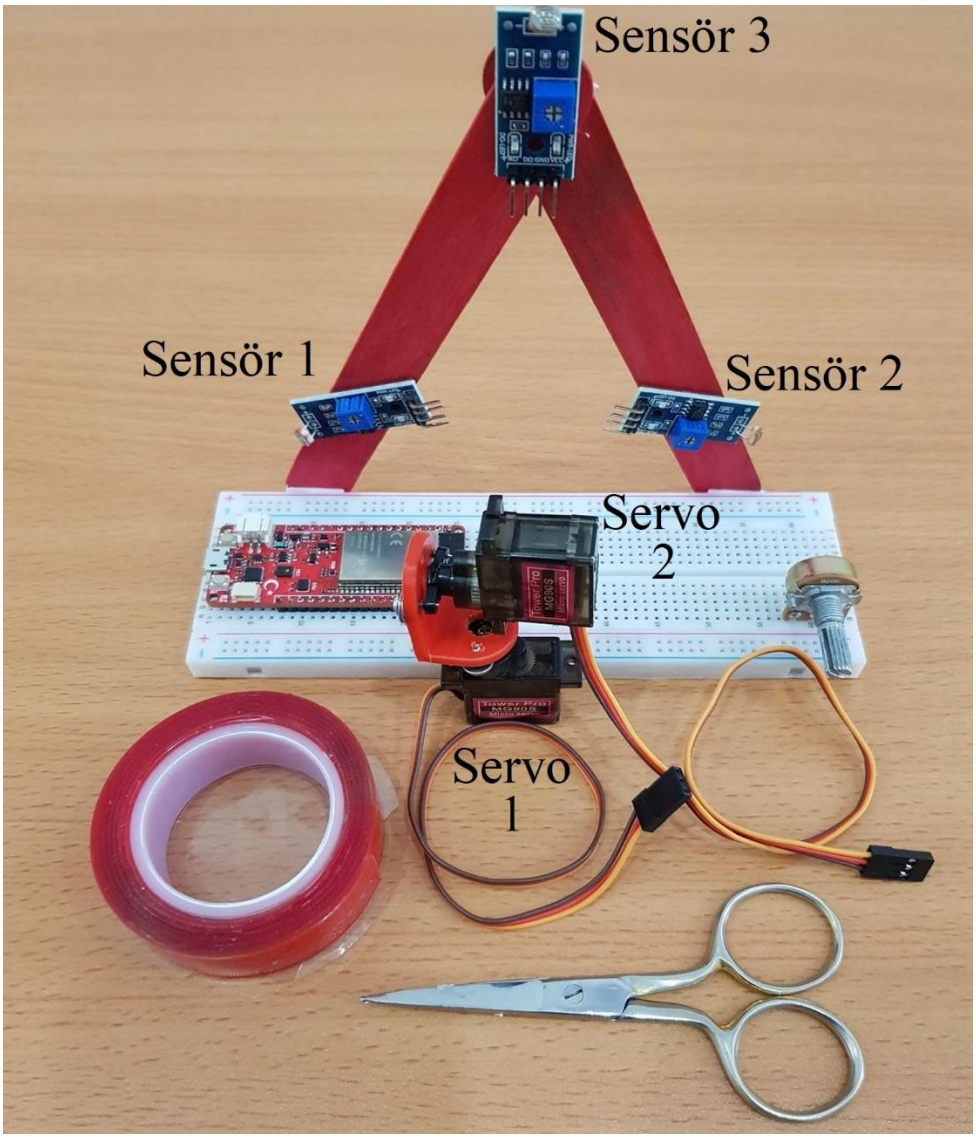
Tasarım ve Üretim Adımları:

Adım 1) Deneyap Geliştirme Kartı Breadboard üzerine yerleştirilir ve 2 adet dil çubuğu da yine Resim 4.18'de gösterildiği gibi bir üçgen oluşturacak şekilde birbirlerinin uçlarına yapıştırılır. Diğer uçları da breadboard'un bir kenarına yapıştırılır. Yapıştırma işlemi için ağaç tutkalı veya silikon tabanca veya çift taraflı yapışkan bant kullanılabilir. Bu yapıştırma işlemi yöntemi, eğitmenin ve öğrencilerin tercihine bırakılmıştır. Resimdeki sistemde, çift taraflı yapışkan bant kullanılmıştır. Ağaç tutkalı kullanıldığında, tutkalın kuruma süresi göz önüne alınıp basit bir planlama yapılmalıdır. Yapıştırma işlemi yapıldığında bekleme sürecinde diğer adımlara (kart bağlantıları, yazılım çalışmaları gibi) geçilebilir ve bu kuruma süresi verimli bir şekilde kullanılmış olur.



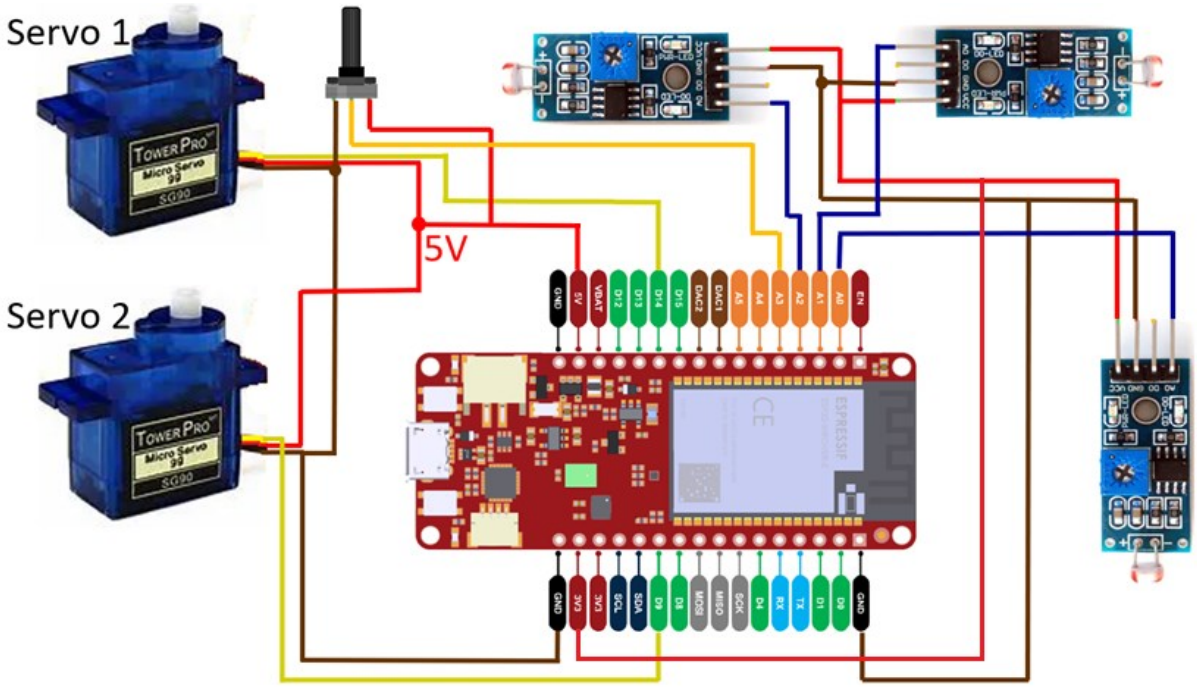
Resim 4.18. Deneyap Geliştirme Kartı ve dil çubuklarının breadboard üzerine yerleştirilmesi

Adım 2: Önceki uygulamalardakine benzer bir çift motor düzeneği, bu sistemde de kullanılmaktadır. Bu çift motor düzeneğinde taban motorunun bir kenarı breadboard'un ön kenarının ortasına Resim 4.19'da gösterildiği gibi yapıştırılır. Üç adet ışık sensörü de üçgen şeklinde konumlandırılmış dil çubukları üzerine şekilde gösterildiği gibi yapıştırılır. Yapıştırma işlemlerinde çift taraflı yapışkan silikon bant kullanılabilir.

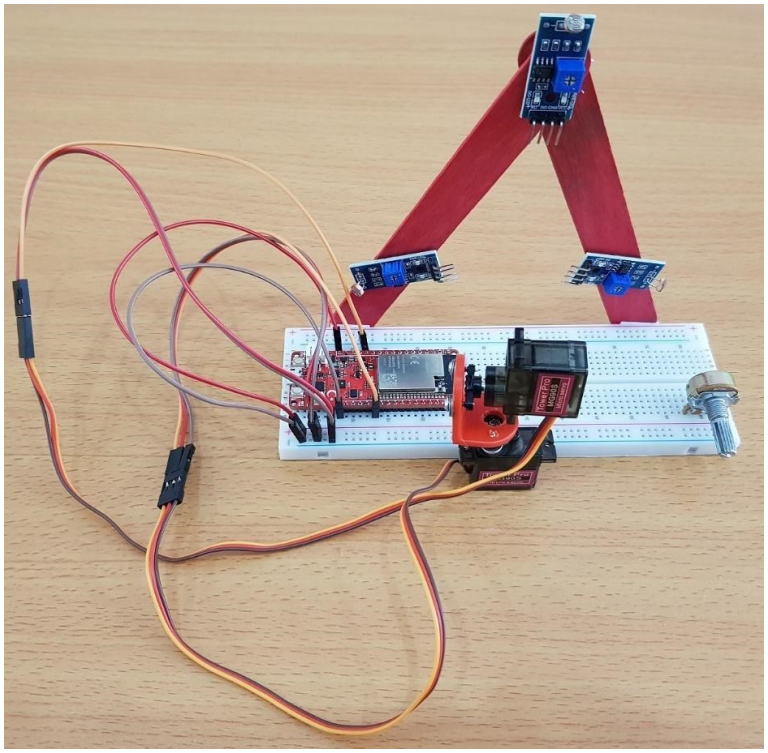


Resim 4.19. Çift motor düzeneği ve ışık sensörlerinin sisteme yerleştirilmesi.

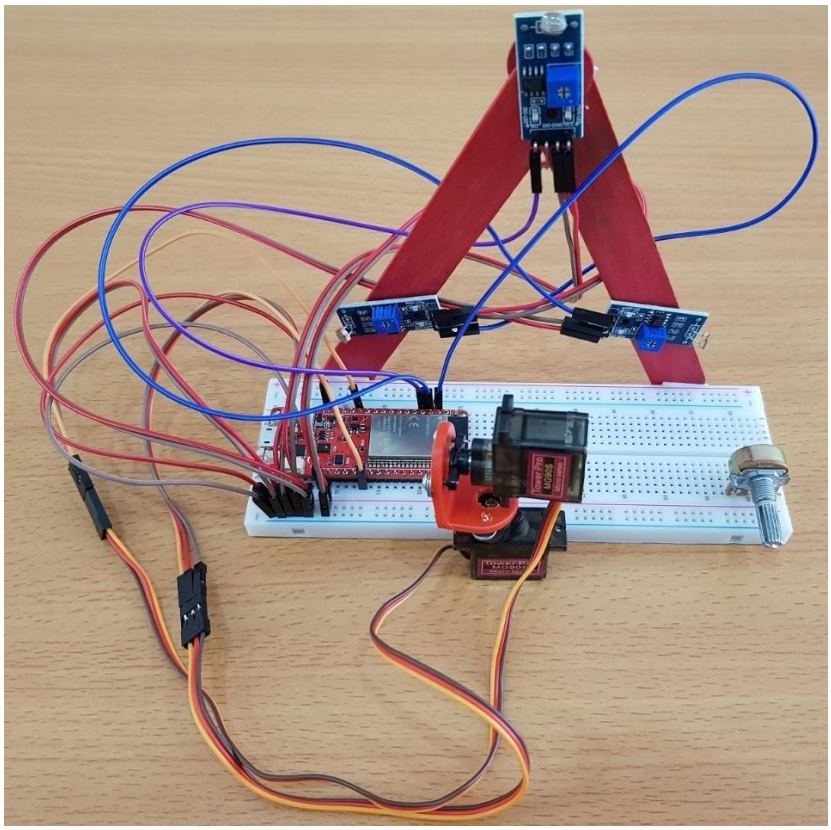
Adım 3: Çift motor düzeneğinin, ışık Sensörlerinin ve potansiyometrenin Deneyap Geliştirme Kartına bağlantıları yapılır. Gerekli bağlantı şeması Resim 4.20'de verilmiştir. Taban motoru şekilde "Servo 1" olarak belirtilen motordur.



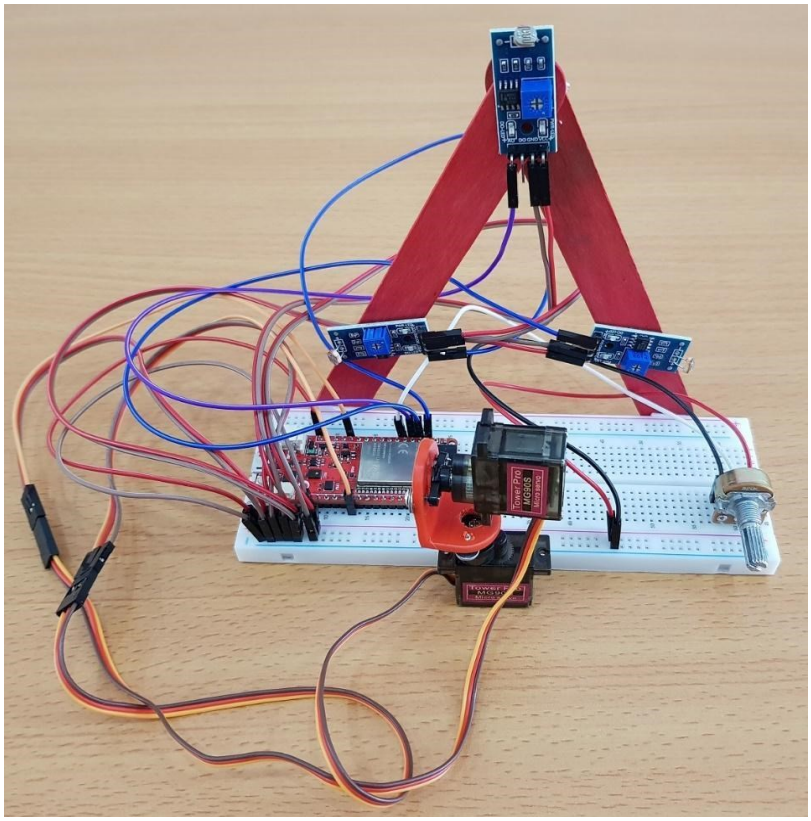
Resim 4.20. Sistemin bağlantı şeması.



Resim 4.21. Çift motor düzeneğinin Deneyap Geliştirme Kartına bağlantıları



Resim 4.22. Işık sensörlerinin Deneyap Geliştirme Kartına bağlantıları.



Resim 4.23. Potansiyometrenin Deneyap Geliştirme Kartına bağlantısı.

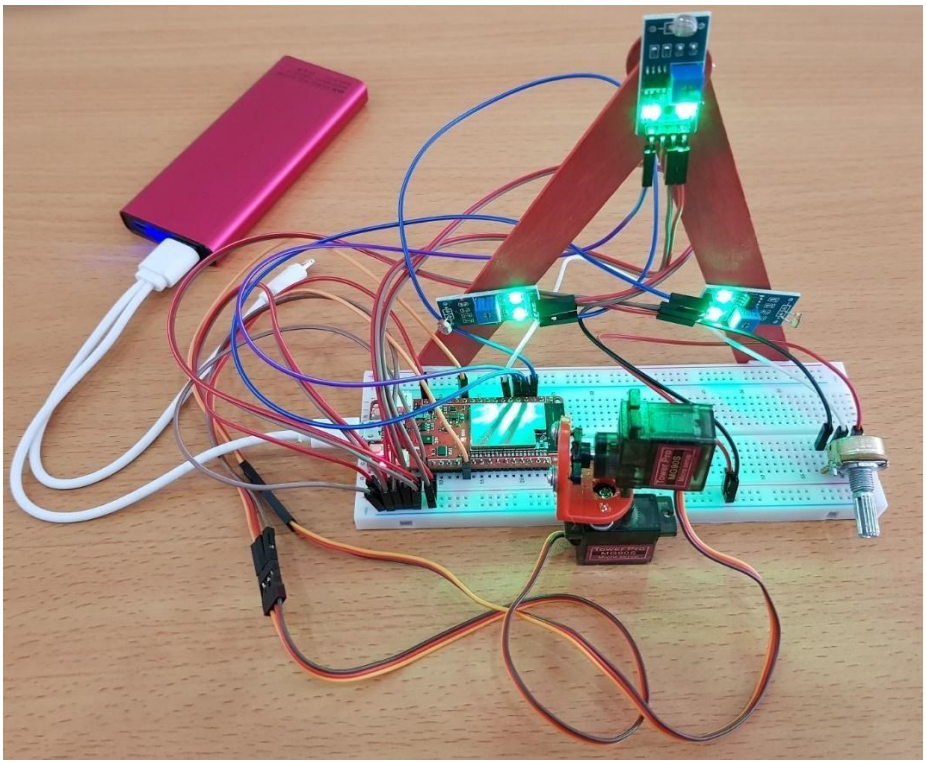
Bu uygulamada, potansiyometre kullanımının amacı, ışık sensör çıkışlarını ortamın aydınlık derecesine göre manuel olarak kalibre etmektir.

Adım 4: Deneyap Geliştirme Kartı bilgisayara bağlanır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include <Deneyap_Servo.h>

// Dijital servo pin bağlantıları:
#define SERVOPIN1 D9
#define SERVOPIN2 D14

// Işık sensörleri analog bağlantıları:
#define L1_pin A0
#define L2_pin A1
#define L3_pin A2
// Potansiyometre analog bağlantısı.
#define Pot_pin A3

// Motor dönüş açı alt ve üst limitleri:
int teta1min=0;
int teta1max=180;
int teta2min=0;
int teta2max=180;
//Accelerometre X ve Y eksenleri etrafı açı değişkenleri
int accAngleX=0;
int accAngleY=0;
int conavail=0;

int coef;  // Potansiyometreden elde edilen katsayı değişkeni.
int valin1, valin2, valin3; // Işık sensörlerinden hesaplanan değerlerin atanacağı değişkenler.
int pos1 = 0; //Servo motor dönüş açı değişkenleri
int pos2 = 0;
int clb1=0;
int clb2=0;
int clb3=0;

// İvme ölçer verisinde oluşan noise'i (gürültüyü) azaltmak (hatta yok etmek) için "moving average" algoritmasının uygulanmasında kullanılan değişkenler.
int i=0;
int n=10;
int32_t tot1=0.;
int32_t tot2=0.;
int32_t tot3=0.;

// Servo motorların tanımlanması.
Servo myservo1, myservo2;
```

```
void setup()
{
  // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
  myservo1.attach(SERVOPIN1);
  myservo2.attach(SERVOPIN2,1);
}

void loop()
{
  coef = analogRead(Pot_pin);  //Potansiyometre çıkışının bir katsayı değişkenine
atanması.

  // Işık sensörlerinden hesaplanan değerlerin atandığı değişkenler.
  valin1 = 4095-analogRead(L1_pin);
  valin2 = 4095-analogRead(L2_pin);
  valin3 = 4095-analogRead(L3_pin);

 // Işık sensörü verisindeki noise'i (gürültüyü) azaltmak için "moving average" (hareketli
ortalama) yöntemi:
 if (i<n) {
   i++;
 }
 tot1=int(((i-1)*tot1+valin1)/i);
 tot2=int(((i-1)*tot2+valin2)/i);
 tot3=int(((i-1)*tot3+valin3)/i);

 // Işık sensörlerinden hesaplana verilerin kullanılması ile hesaplanan servo motor
dönüş açıları:
 pos1=90+(tot2-tot1)*((2048-coef)/2048.);
 pos2=120+(0.5*(tot1+tot2)-tot3)*((2048-coef)/2048.);

 // Motorlara dönüş komutlarının verilmesi.
 myservo1.write(pos1);
 myservo2.write(pos2);

 delay(5);  // Genel döngü içerisinde 5 milisaniye gecikme.
}
```

Adım 5: Program Deneyap Geliştirme Kartına yüklendiğinde motorların ışık sensörleri ve potansiyometre aktivitelerine göre davranış gösterdiği görülecektir. Resim 4.24'te görüldüğü gibi sistem harici bir batarya ile çalıştırılıp test edilebilir. Sistemi test ederken bir el feneri ya da bir cep telefonunun flaş ışığı kullanılabilir. Sistemin davranışı (motorların ışığa karşı reaksiyonu) devredeki potansiyometre ile kalibre edilebilir.



Resim 4.24. Sistemin harici batarya ile çalıştırılması.

3. DEĞERLENDİR

Öğrenciler ile bu hafta yapılan uygulamalar konusunda görüşmeler ve değerlendirmeler yapılmalı ve yeterli kazanımları elde edip etmedikleri belirlenmelidir. Aşağıdaki örnek sorular üzerinden tartışmalar gerçekleştirilebilir:

- İki serbestlik dereceli bir robotun özellikleri nelerdir?
- Tasarım süreci ile ilgili bir problem yaşadınız mı? Yaşadıysanız bunu nasıl çözdünüz?
- Devrelerin fiziki kurulumunda zorlandığınız yerler oldu mu?
- Bu derste bahsi geçen iki serbestlik dereceli robotların pratik hayattaki uygulamaları (veya endüstriyel uygulamaları) neler olabilir? Örnek uygulamalar:
    - Önceki tek motorlu sistemlere verilen örnektekine benzer olarak, geniş bir alanda (örneğin bir işyerinde) ışığa duyarlı pan-tilt hareketleri yapabilen bir güvenlik kamera kontrol sistemi geliştirilebilir. Gece karanlıkta o alana elinde bir el feneri ile giren bir hırsız, bu sistem sayesinde tespit edilir ve kamera da o hırsıza üç boyutlu olarak yönlendirilip net olarak görüntüsü alınır ve ayrıca alarm da aktif hale getirilir.
    - Yine önceki tek motorlu sistemlere verilen örnektekine benzer olarak, bir sınır karakolunda, ışığa duyarlı pan-tilt hareketleri yapabilen bir silah sistemi geliştirilebilir. Gece karanlıkta sınır hattından terörist veya düşman tarafından yapılan ateşli bir saldırıya anında karşılık verilebilir. Işık sensörleri ilk ateş edilen konuma yönlendirilip karşı ateş edildiğinde bu çok etkili bir savunma hareketi olur. Karakolda herkes uyuduğu anda böyle bir sistem harekete geçerek hayat kurtarır.

- Güneş ışığından en verimli bir şekilde enerji sağlamak amacıyla, güneş panellerinin duruşunu değiştiren motorlu pan-tilt sistemleri ışık sensörleri ile kontrol edilebilir ve böyle paneller tam olarak güneşe yöneltilerek güneşten en fazla enerji alımı sağlanmış olur.
- İki serbestlik dereceli (yani iki motorlu) gimbal sistemleri günümüzde çok yaygın olarak kullanılmaktadır. IMU sensörlerinden alınan geri besleme sinyalleri ile iki motor kontrol edilerek pan-tilt hareketleri yaptırılıp denge sistemi çalıştırılmış olur, böylece kamera görüntüsü çok net olarak alınmış olur.
- Titreşim kontrollü hedeflemeli silah sistemlerinde pan-tilt hareketi kabiliyetine sahip platformlar kullanılabilir.
- Bir teknede dalga ve titreşimlerden etkilenmeyen bir pan-tilt hareket mekanizmasına sahip bir masa tasarlanabilir.

5. İLAVE ETKİNLİK

Bu aşamada, öğrencilerden iki serbestlik dereceli farklı ve yeni bir sistem uygulamasının tasarımı, üretimi ve uygulaması yaptırılabilir.

Örnek uygulama: İki serbestlik dereceli olarak, tek motorlu küçük bir asansör sistemi ve üzerinde yine tek motorlu bir itme mekanizması olan robotik bir sistem tasarlanabilir. Asansör düşük seviyede iken, bir sensör ile platformda bir nesne olduğunu tespit ettiğinde yukarı doğru o nesneyi çıkartır, itme mekanizması ile platformdan yukarıda istenilen bir yere iter, sonra tekrar aşağı seviyeye iner ve bir sonraki nesnenin yerleştirilmesini bekler. Bu işlem aynı şekilde tekrarlanır.

Burada başka ne tür uygulamalar yaptırılacağı hususu, eğitimciye ve öğrencilerin hayal gücüne bırakılmıştır.

5. Bölüm: Üç Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap Geliştirme Kartının temel özelliklerini bilir.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde servo motor kullanımı bilgisine sahiptir.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde kumanda kolu kullanımı bilgisine sahiptir.

Haftanın Kazanımları:

- Öğrenciler, 3 serbestlik dereceli bir robot manipülatörün mekanik ve elektronik donanımını oluşturmayı bilir.
- Öğrenciler, bir robot manipülatörü 2 adet kumanda kolu ile kontrol etmeyi ve bir görevi öğretebilmek için gerekli yazılımı Deneyap Geliştirme Kartı üzerinde kodlamayı bilir.
- Öğrenciler, 3 serbestlik dereceli robot üretimi için gerekli mekanik tasarım ve üretim, elektronik tasarım ve kodlama işlemlerini yapar.
- Öğrenciler, 3 serbestlik dereceli robot ile çeşitli uygulamalar yapar.
- Öğrenciler, serbestlik derecesi artışının çalışma uzayına etkisini gözlemler ve deneyimler.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin 3 serbestlik dereceli (3 döner eklemli) ve uç kısmında kıskaç ele (gripper) sahip bir robot manipülatörün mekanik, elektronik ve yazılım çalışmalarını eğlenceli uygulamalar ile öğrenmelerini sağlamaktır. Bu robot manipülatör uygulamasında, kumanda kolu modüllerinin kullanımı ile açık çevrim bir kontrol çalışması da amaçlanmıştır.

Kullanılacak Malzemeler:

1. Deneyap Geliştirme Kartı, 1 adet
2. Servo motor, 4 adet
3. XY Kumanda kolu Modülü, 2 adet
4. Breadboard, 1 adet
5. Çeşitli renklerde erkek-dişi ve erkek-erkek jumper kabloları, 30+ adet
6. M2 veya M3 rondela (pul), 15 adet
7. Tahta tutkalı, 1 adet
8. Abeslang, 1 paket
9. Esnek plastik levha
10. Naylon dikiş ipi
11. Mikro USB kablosu
12. Harici 5V USB Batarya
13. Çift taraflı silikon bant ve makas

14. 3B Yazıcı ve Filament

Haftanın İşlenişi:

Gözle: Bundan önceki haftalarda uygulaması yapılan modüllerin Deneyap Geliştirme Kartı üzerinde kullanımlarının ve temel kodlamalarının kısaca üzerinden geçilerek öğrencilere hatırlatılması.

Uygula: Yönergeye uygun şekilde tüm montaj ve bağlantıların yapılması, kodların yazılması

Tasarla: Üç serbestlik dereceli robot mekanizmasının tasarlanması ve Deneyap Geliştirme Kartı programının bu robotun kullanımı için yazılması.

Üret: Tasarlanan donanımın gerçekleştirilmesi.

Değerlendir: Öğrenciler ile bu uygulama konusunda tartışma ve değerlendirmelerin yapılması.

1. GÖZLE VE UYGULA

1.1. Gözle: Üç Serbestlik Dereceli Robotların Yapısı ve İşlevleri

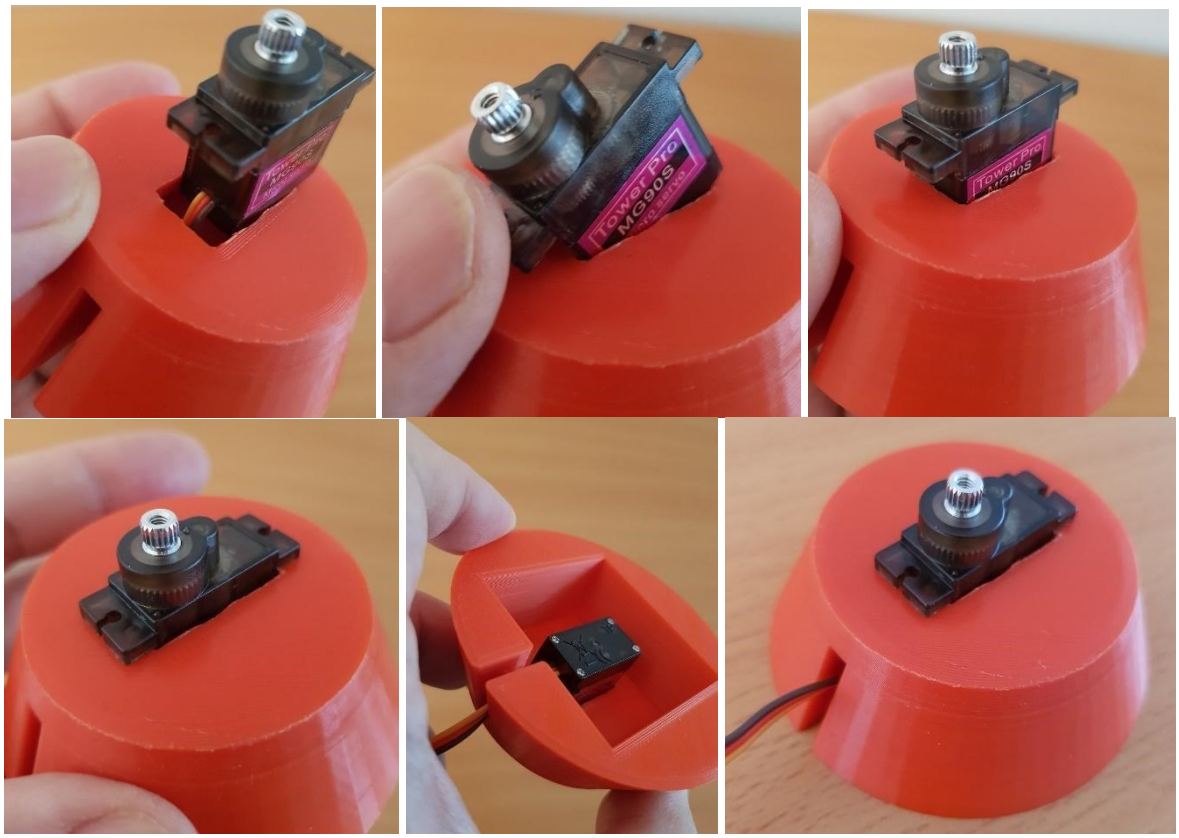
Dersin başlangıcında, eğitmen öğrencilerin 3 serbestlik dereceli robotlar ile ilgili tartışmalarını, ilk dersin konularını hatırlamalarını sağlamalıdır. Üç serbestlik dereceli robotların yapısı ve kullanım alanları üzerine öğrencilerin beyin jimnastiği yapmaları, onların derse ilgi ve katılımlarını artıracaktır. Internet ortamında 3 serbestlik dereceli çeşitli robotlar görülmektedir. Bunların resim ve videolarına yine internet ortamında ulaşılmaktadır. Örnek bir 3 serbestlik dereceli robot uygulaması “https://www.youtube.com/watch?v=ceLIAkFdSEM” linkinde gösterilmektedir. Daha çeşitli uygulamalara da yine internet ortamında ulaşılabilmektedir.

Bu bölümde üç dönel ekleme ve bir tutucu ele sahip bir robot kol tasarımı, üretimi ve uygulaması yapılacaktır.

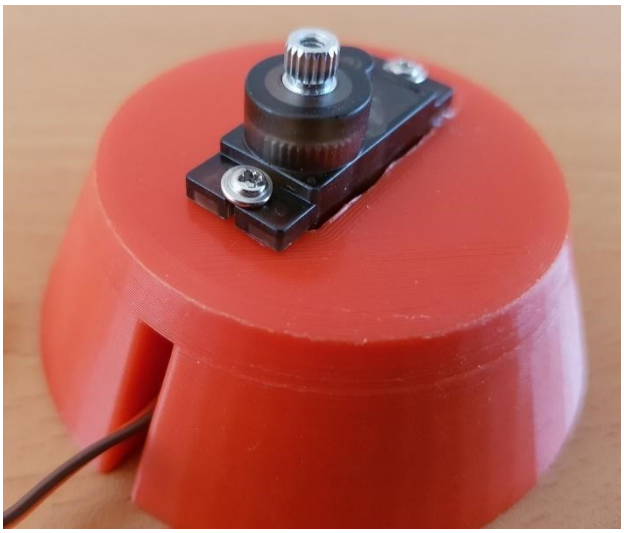
1.2. Uygula: Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü

Uygulama Adımları:

Adım 1: 1 numaralı servo motor 3B yazıcıdan basılmış taban parçasına Resim 5.1’de gösterildiği gibi dikkatlice yerleştirilir. Bu aşamada servo motorun kablosuna dikkat edilmelidir. Kabloda meydana gelebilecek bir deformasyon motorun çalışmamasına neden olabilmektedir. Motor yerleştirildikten sonra taban parçasına vidalanır (Resim 5.2). Eğer 3 boyutlu yazıcı konusunda tecrübesizlik nedeniyle bu taban parçasının üretilmesinde zorlukla karşılaşılırsa, buna alternatif olarak, elde halihazırda mevcut plastik veya pleksiglas malzemeden yapılmış dikdörtgen bir taban parçası üzerine taban motorunun yapıştırılması yoluna da gidilebilir.



Resim 5.1. Servo motorun 3B yazıcıdan basılmış taban parçasına yerleştirilmesi



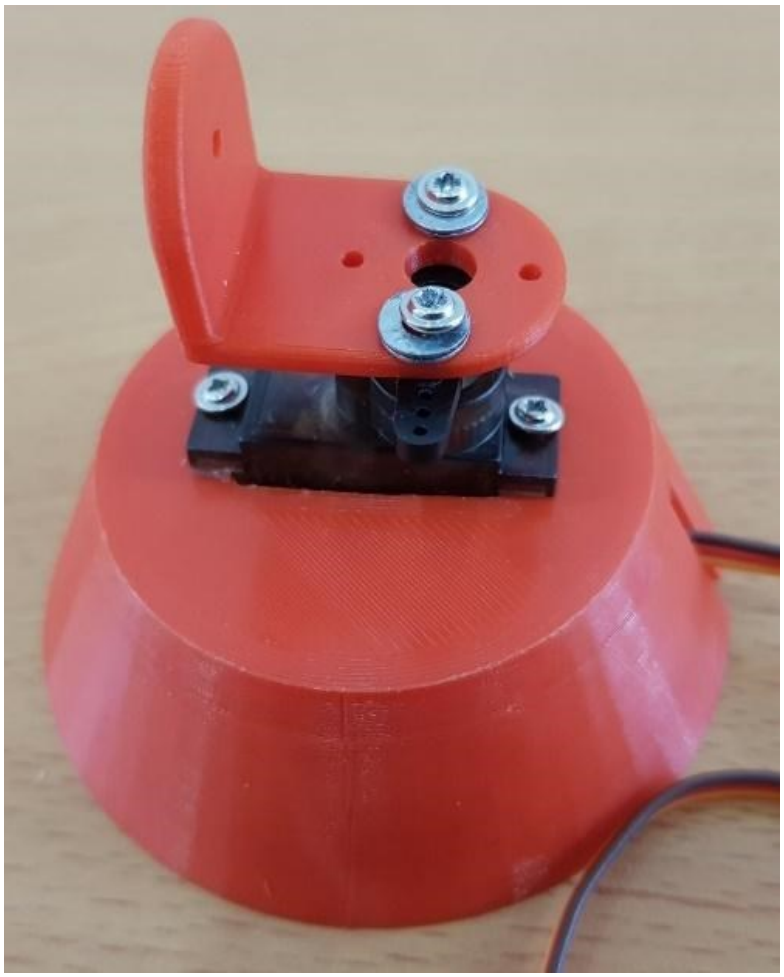
Resim 5.2. Taban parçaya vidalanmış servo motor

Adım 2: İki numaralı servo motoru bağlamak için 3B yazıcıda basılmış bir L-profil kullanılır (Resim 5.3). Burada servo motor plastiklerinden T şeklinde olanı kullanılır. T şeklinde plastik bulunamazsa diğerleri de kullanılabilir.



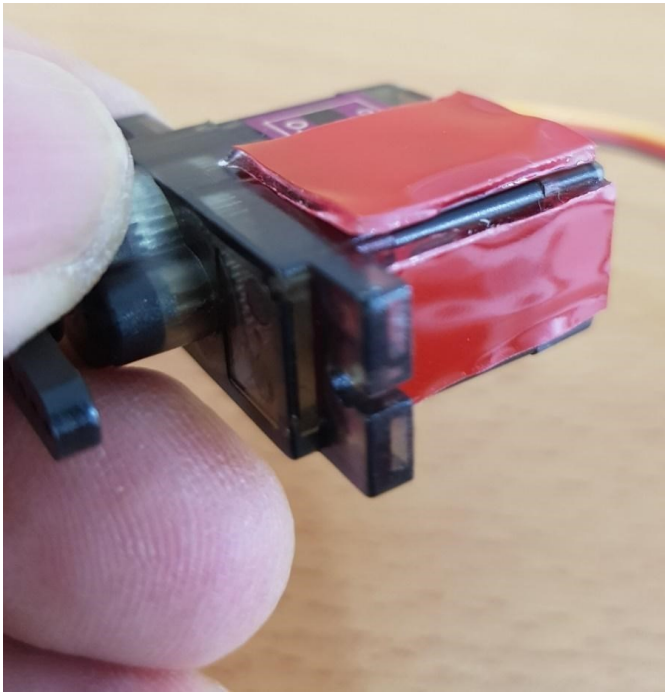
Resim 5.3. 3B-yazıcıda basılan L-profil ve üzerine vidalanmış servo motor plastik parçası

Adım 3: Hazırlanan L-profil, taban motorunun (Motor 1) miline Resim 5.4'te gösterildiği gibi yerleştirilir.



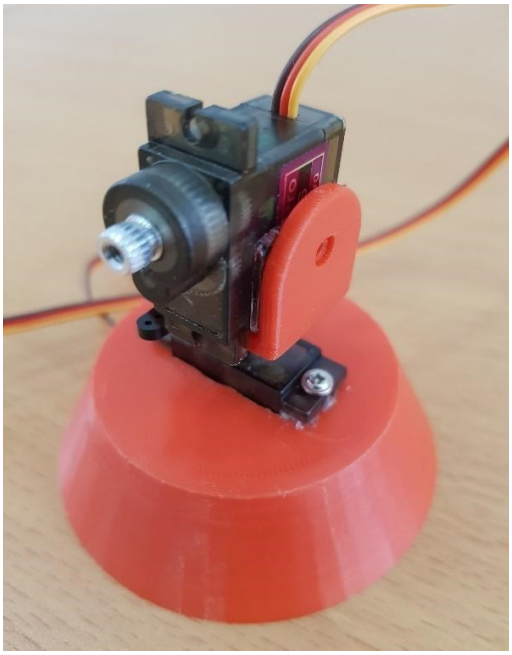
Resim 5.4. Hazırlanan L-profil parçanın taban motoru miline yerleştirilmesi

Adım 4: : İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılır (Resim 5.5). Burada, önceki bölümlerdekine benzer olarak, silikon tabanca ile sıkılan yapıştırıcı silikon malzemede kullanılabilir. Bu, eğitmen ve öğrencinin tercihine bırakılmıştır.



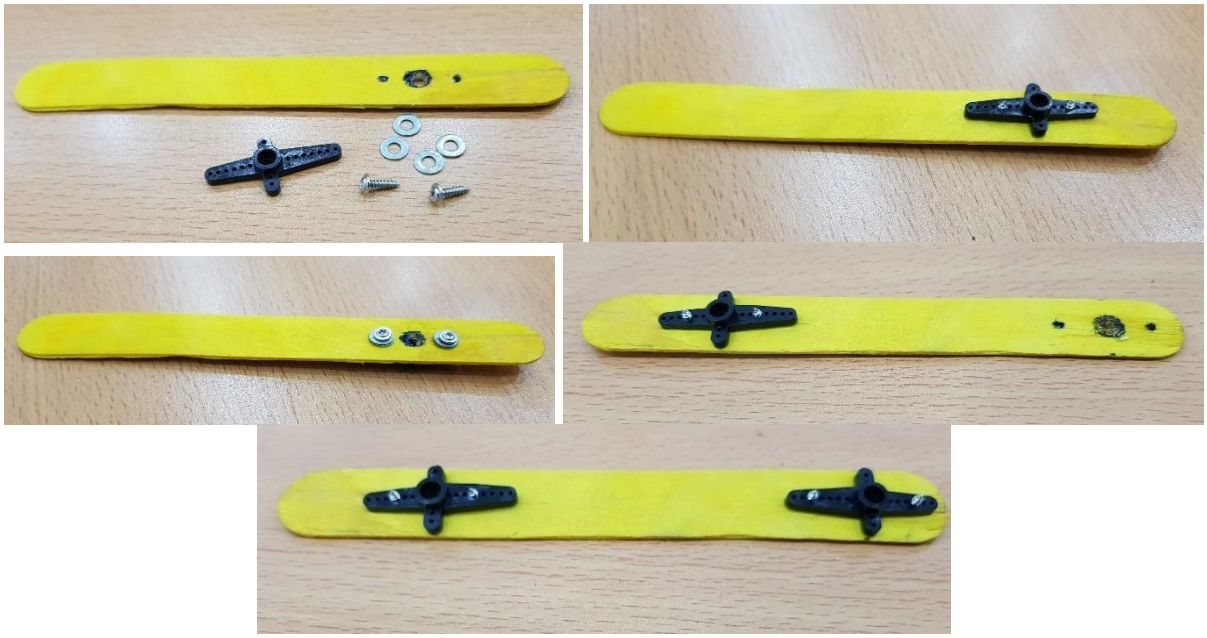
Resim 5.5. İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılması

Adım 5: İki numaralı servo motora yapıştırılmış çift taraflı silikon bantın kırmızı naylonları çıkartılır ve servo motor L-profil parçanın üzerine yapıştırılır (Resim 5.6).



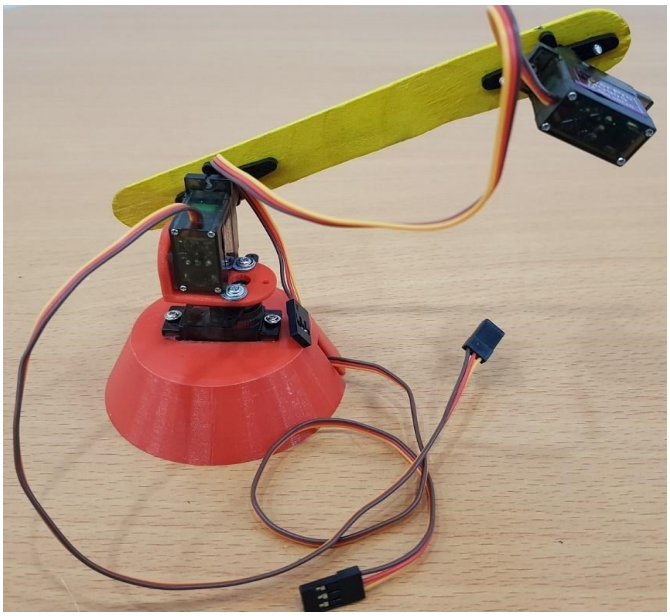
Resim 5.6. İki numaralı servo motorun L-profil parça üzerine konumlandırılması

Adım 6: Bir adet abeslang Resim 5.7'deki gibi hazırlanır. Abeslangların uçlarına ince bir alet ile kolayca delik delinebilmektedir, Delikler delinirken abeslangların çatlatılmamasına dikkat edilmelidir. Eğer mümkünse, bu delikler bir matkap yardımıyla delinmelidir. Resimdeki yapıda bir adet abeslang kullanılmıştır. Daha rijit bir sistem için birden fazla abeslang tutkal ile üst üste yapıştırılarak da kullanılabilir. Eğitmenler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, eğitmenlere ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.



Resim 5.7. Abeslanglara servo motor T-plastik parçalarının takılması

Adım 7: Hazırlanmış olan abeslang Resim 5.8'de gösterildiği gibi 2 numaralı servo motora takılır. Burada hareket açı aralıkları deneme yanılma yolu ile belirlenerek abeslang yerleştirilir ve motor mili vidası takılarak sıkıştırılır. Abeslangın diğer tarafına da yine şekilde gösterildiği gibi 3 numaralı servo motor yerleştirilir.



Resim 5.8. Abeslangın 2 numaralı servo motora ve 3 numaralı motorun abeslanga monte edilmesi

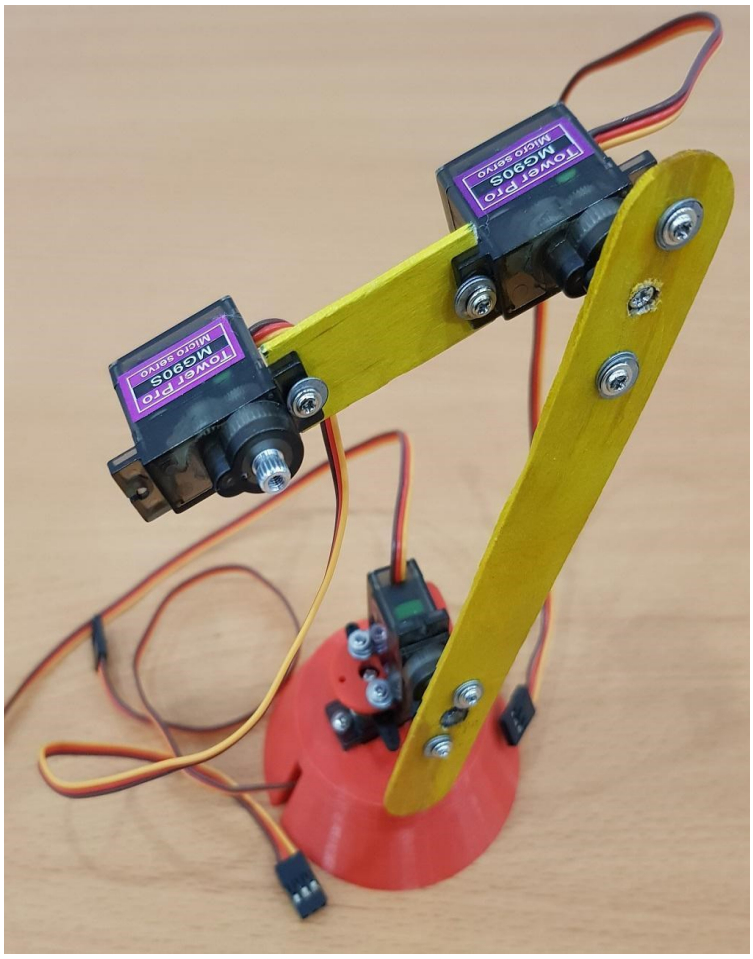
Adım 8: Başka bir abeslangtan kısa bir parça kesilir (Resim 5.9). Eğitmenler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, eğitmenlere ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.





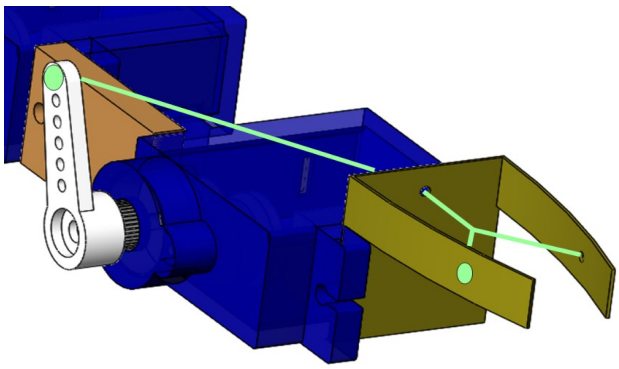
Resim 5.9. Kısa bir parçanın abeslangtan kesilmesi ve hazırlanması

Adım 9: Hazırlanmış olan kısa parça ile 3 ve 4 numaralı servo motorlar birbirlerine bağlanır (Resim 5.10).

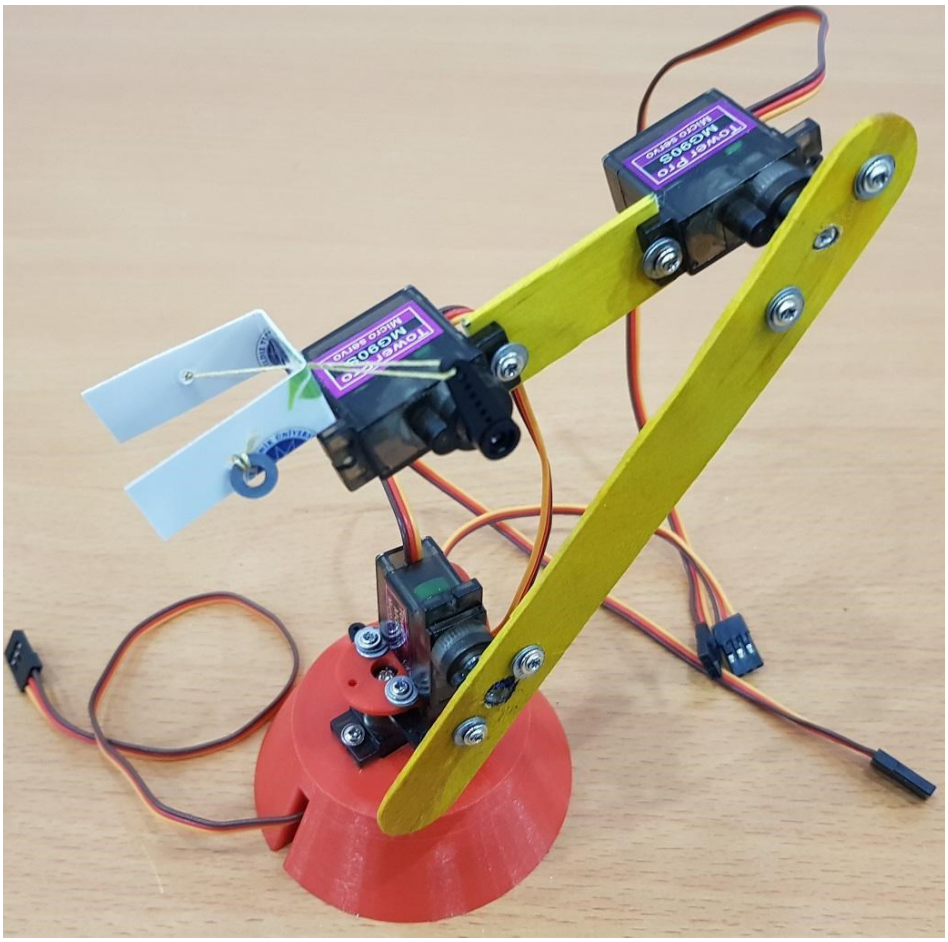


Resim 5.10. Kısa parça ile 3 ve 4 numaralı servo motorların birbirlerine bağlanması

Adım 10: Tutucu (Gripper) tasarımında (Resim 5.11) bir servo motor, esnek plastik malzemeden kesilmiş ve bükülmüş tutucu parçası ve açı değişimi işlemini gerçekleştirmek için servo motorun hareketini tutucuya aktaracak naylon ip kullanılmaktadır. Esnek plastik kart, tutucu şeklinde kesilerek bükülür ve Resim 5.11'da gösterildiği gibi servo motora yapıştırılır. Yine resimde gösterildiği gibi naylon ip de tutucu ile motor arasına bağlanır. Motorun dönme yönüne göre tutucu elinin aç/kapa hareketini yaptığı gözlemlenir. Bu motorlu tutucu mekanizma Resim 5.12'de gösterildiği gibi robot manipülatöre takılır.

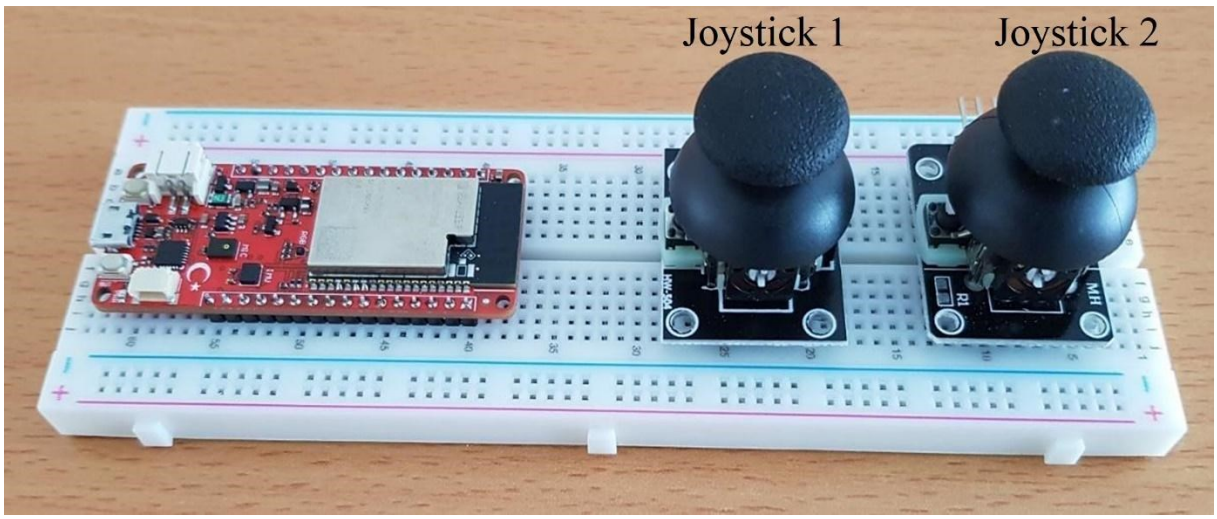


Resim 5.11. Tutucu tasarımı



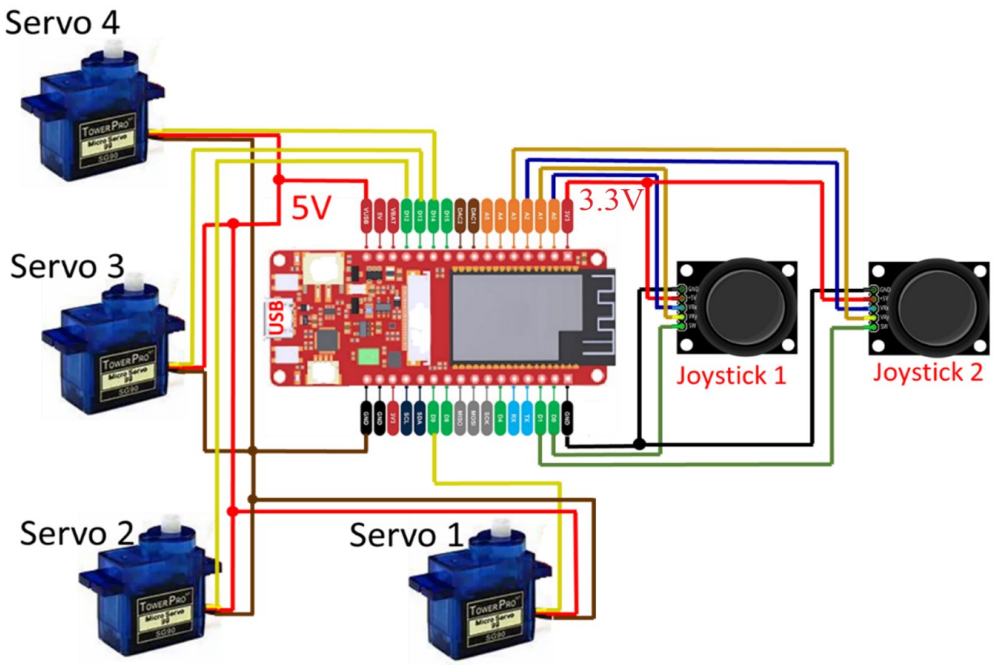
Resim 5.12. Tutucusu yerleştirilmiş robot manipülatör

Adım 11: Deneyap Geliştirme Kartı breadboard üzerine yerleştirilir. Kumanda kollarının alt tabanlarına çift taraflı silikon bant yerleştirilir ve breadboard üzerine Resim 5.13'de gösterildiği gibi yapıştırılır. Silikon bant lehim çıkıntıları nedeniyle 2 veya 3 olarak uygulanabilir.

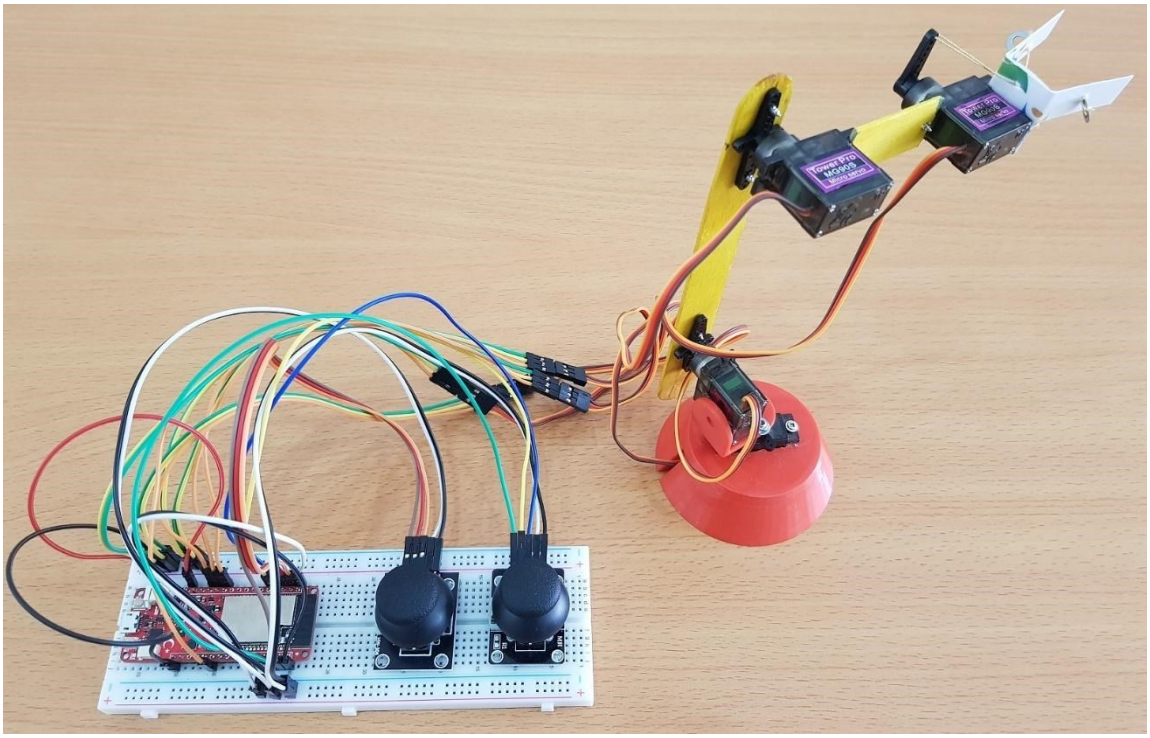


Resim 5.13. Breadboard üzerine konumlandırılmış kumanda kolları

Adım 12: Kumanda kollarının ve servo motorların Deneyap Geliştirme Kartına bağlantıları Resim 5.14'te verilmiştir.



Resim 5.14. Uygulama 1 devre şeması



Resim 5.15. Üç serbestlik dereceli robot manipülatörün bağlantılarının yapılmış genel fotoğrafı

Adım 13: Deneyap Geliştirme Kartının bilgisayarla bağlantısı yapılır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include "deneyap.h"
#include <Deneyap_Servo.h>// Eğer bu kütüphane çalışmaz ise <ESP32Servo.h> kütüphanesi denenebilir.
// Dijital kumanda kollarının button pin bağlantıları:
#define SW1_pin D0
#define SW2_pin D1

// kumanda kollarının eksen analog bağlantı pinleri
#define X1_pin A0
#define Y1_pin A1
#define X2_pin A2
#define Y2_pin A3

// Dijital servo pin bağlantıları:
#define SERVOPIN1 D9
#define SERVOPIN2 D12
#define SERVOPIN3 D13
#define SERVOPIN4 D14
// Motor dönüş açı alt ve üst limitleri:
//Aşağıdaki teta değerleri 0-3000 arasında kalibre edilmiştir.  3000 değeri 180 dereceye karşılık gelmektedir.
int teta1min=0;
int teta1max=3000;
int teta2min=0;
int teta2max=3000;
int teta3min=0;
int teta3max=3000;
int teta4min=1500;
int teta4max=2000;

int valin1, valin2, valin3, valin4;
int valcalc1=1500;//Başlangıç açı değerleri
int valcalc2=1500;
int valcalc3=1500;
int valcalc4=2000;
int jystp=20; // kumanda kollarıyla motor hareketleri sağlandığında her bir döngüde istenen açı değeri (bu yazılımda 3000 değerinin 360 dereceye karşılık geldiğini unutmayın.

bool playmode=false; // Robotun tekrarlama (oynama) modu değişkeni.  Değer false olduğunda, öğretme modunda demektir.
int ButtonMode=0; // Buton modu değişkeninin tanımlanması.  Sıfır olduğunda, öğretme modunda demektir.
unsigned long CurrentMilliSec = millis(); // milis() fonksiyonu, şu anki saati milisecond cinsinden vermektedir.  Zamanlayıcı olarak bu fonksiyonu kullanmaktayız.  Herhangi bir anda, o anki zamanı bir long değişkenine atayabiliriz.
```

```
long PreMilliSec=0; //  Bir evvelki zaman (milisaniye cinsinden)
int NumMemory=0; // Hafıza sayacı (Maksimum 50 olabilir.  Deneyap kartının kapasitesine bağlı olarak bu sınır 50'den yukarıda olabilir.)
int PreNumMemory=0; // Bir evvelki moddaki hafıza değeri.
int Step=-1; // Öğretme modunda, servo pozisyonlarını hafızaya alırken kullanılan bir değişken.
// Servo pozisyonlarının hafızaya alındığı 50'lik seriye sahip değişkenler.
int joint1[50];
int joint2[50];
int joint3[50];
int joint4[50];
int maxarray=49;  //Hafıza hatası vermeyi önlemek için 49 değerine sahip bir limit değişkeni.
int prehiz=10;  // Bir önceki hız değerinin atanacağı değişken tanımlaması.
int normalhiz=10; // Normal hız değişkeni
// hiz=1 çok yavaş, 5 yavaş, 10 normal, 15 hızlı, 20+ çok hızlı
int hiz=10; // Şu anki hız değişkeni
int prepos1, prepos2, prepos3, prepos4; // bir evvelki motor konumları.
int pos11,pos21,pos31,pos41; // Motor konumları ile ilgili ikincil (yardımcı) değişkenler.

Servo myservo1, myservo2, myservo3, myservo4; // Servo motorların tanımlanması.

void setup()
{
 myservo1.attach(SERVOPIN1); //Eğer <ESP32Servo.h> kütüphanesi kullanırsanız, bu ve aşağıdaki 3 satırda parantez içerisindeki 2'inci parametreyi silin, tek parametre yeterlidir.
 myservo2.attach(SERVOPIN2,1);
 myservo3.attach(SERVOPIN3,2);
 myservo4.attach(SERVOPIN4,3);

 // Joystik butonlarının pinlerinin tanımlanması (Normalde HIGH).
 pinMode(SW1_pin, INPUT_PULLUP);
 pinMode(SW2_pin, INPUT_PULLUP);
 // kumanda kolu pinlerinin giriş olarak tanımlanması
 pinMode(X1_pin, INPUT);
 pinMode(Y1_pin, INPUT);
 pinMode(X2_pin, INPUT);
 pinMode(Y2_pin, INPUT);
 // Joystik butonlarının pinlerinin değerleri normalde HIGH' a atandı.
 digitalWrite(SW1_pin, HIGH);
 digitalWrite(SW2_pin, HIGH);
// Tercihe bağlı olarak, Seri port'a "Ready" yazdırılması (baudrate= 9600)
 Serial.begin(115200);
 Serial.println("Ready.");
 }

void loop() {
```

```
  CurrentMilliSec = millis(); // Şu anki zaman değeri (ms)
  Button(); // Kendi yazdığımız Button fonksiyonunun çağırılması.
  if (playmode==false && Step==-1){ // Öğretme modunda iken,
    ReadAnalog();  //  Analog kanalların okunması için kendi yazdığımız ReadAnalog
fonksiyonunun çağırılması.
    // Servo motorlara dönüş komutlarının verilmesi:
    myservo1.writeMicroseconds(valcalc1);
    myservo2.writeMicroseconds(valcalc2);
    myservo3.writeMicroseconds(valcalc3);
    myservo4.writeMicroseconds(valcalc4);
    delay(int(300./hiz)); // Hız kontrolü için genel döngü içerisindeki gecikme ayarı.
  }
  else{
    playit(); //tekrarlama (veya oynama) modunda iken kendi yazdığımız Play
fonksiyonunun çağırılması.
  }
}
```

Kumanda kollarının ürettiği analog sinyallerin okunduğu ve motor açı değerlerinin bu sinyallere göre değiştirildiği fonksiyon aşağıda verilmiştir.

```
void ReadAnalog() {
  // Analog pinlere giren kumanda kolu değerlerinin okunması.
  valin1 = analogRead(Y1_pin);
  valin2 = analogRead(X1_pin);
  valin3 = analogRead(X2_pin);
  valin4 = analogRead(Y2_pin);

  // Belli sınırlar dahilinde, Joystik değerlerine göre servo motor dönüş açılarının tayini.
  if (valin1<1000 && valcalc1>teta1min) valcalc1-=jystp;
  if (valin1>3000 && valcalc1<teta1max) valcalc1+=jystp;
  if (valin2<1000 && valcalc2>teta2min) valcalc2+=jystp;
  if (valin2>3000 && valcalc2<teta2max) valcalc2-=jystp;
  if (valin3<1000 && valcalc3>teta3min) valcalc3-=jystp;
  if (valin3>3000 && valcalc3<teta3max) valcalc3+=jystp;
  if (valin4<1000 && valcalc4>teta4min) valcalc4-=jystp;
  if (valin4>3000 && valcalc4<teta4max) valcalc4+=jystp;
}
```

Buton fonksiyonu oldukça kapsamlıdır. Burada 2 adet buton ile robot hareket modları tayin edilmektedir. Öğretme modunda iken kumanda kolu 1'in butona tek tıklama ile mevcut motor açılarını hafızaya alma işlemi yapılabilmekte, öğretme modundan tekrarlama moduna veya tersine yine Kumanda kolu 1'in butonuna çift tıklayarak geçilebilmektedir. Bu uygulamada da motorlara hızlı, normal ve yavaş olmak üzere 3 ayrı seviye tayin edilmiştir. Bu hız seviyeleri arasındaki geçişler Kumanda kolu 2'nin butonuna basılarak gerçekleştirilmektedir. Belirtilen buton aktivitelerini gerçekleştiren kodlar aşağıda verilmiştir.

```
void Button() { // Butona basılma durumu
  if (playmode==false){ // Eğer öğrenme modunda iken,
  if (digitalRead(SW1_pin) == 0){
    delay(100);
    if (digitalRead(SW1_pin) == 1) { // butona tıklanmış ise,
       if (ButtonMode == 0)
       {
         ButtonMode = 1; // İlk buton tek tık tespiti.
         PreMilliSec = CurrentMilliSec;
       }
       else if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec < 500))
       {
         ButtonMode = 2; // İlk buton çift tık tespiti.
       }
     }
    }
    if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec > 600)) {
       saveit(); // İlk buton tek tık tespit edildiğinde, motor konumları hafızaya alınır.
       NumMemory += 1;
       if (NumMemory>maxarray) NumMemory= maxarray;
       ButtonMode = 0;
       Serial.print("Saved: "); Serial.println(NumMemory);
    }
    else if (ButtonMode == 2 && (CurrentMilliSec - PreMilliSec > 600)) {
      if ((NumMemory>1 || PreNumMemory>1) && playmode==false){
       Serial.println("Playmode...");
       playmode = true; // Çift tık tespit edildiğinde tekrarlama (oynama) moduna geçilir.
       ButtonMode = 0;
       if (NumMemory<2) NumMemory=PreNumMemory; else
PreNumMemory=NumMemory; // Eğer hafızaya herhangi bir konum alınmamış olup ta
çift tık yapılırsa, bir önceki oynatılmış konum bilgilerinin tekrar kullanılması için devreye
alınması.
       // Motor açı konumlarının hafızaya alınması:
       joint1[0]=pos11;
       joint2[0]=pos21;
       joint3[0]=pos31;
       joint4[0]=pos41;

       Step=NumMemory-1;
       ReadAnalog();
       prepos1=valcalc1;
       prepos2=valcalc2;
       prepos3=valcalc3;
       prepos4=valcalc4;
     }
    }
    if (ButtonMode<3 && CurrentMilliSec - PreMilliSec > 1000) {
```

```
    ButtonMode = 0; // Tekrar öğretme moduna geçiş.
   }
  }
  else if (ButtonMode<3 && digitalRead(SW1_pin) == 0) {
   ButtonMode = 3;
   Serial.println("Stopping."); // Tekrarlamayı (Oynamayı) durdur.
   ReadAnalog();
   NumMemory+=1;
   // Durulan konumun hafızanın en son basamağına kaydedilmesi.
   joint1[NumMemory-1] = valcalc1;
   joint2[NumMemory-1] = valcalc2;
   joint3[NumMemory-1] = valcalc3;
   joint4[NumMemory-1] = valcalc4;
  }
```

// İkinci butona her tık yapıldığında robot hareket hızı değiştirilir. Normal, yavaş, hızlı olmak üzere 3 ayrı hız için, aşağıdaki kod yazılmıştır. Diğer detaylar, önceki bölümlerdeki kodlarda anlatıldığı gibidir.

```
  if (digitalRead(SW2_pin) == 0) {
   if (prehiz==hiz) {
    if (hiz==normalhiz) {
     hiz=2*normalhiz;
      jystp=40;
     Serial.println("High speed.");
     if (playmode==false) {
      delay(300);
     }
    } else {
     if (hiz==(2*normalhiz)) {
      hiz=normalhiz/2;
      jystp=10;
      Serial.println("Low speed.");
      if (playmode==false) {
       delay(300);
      }
     }
     else if (hiz==(normalhiz/2)) {
      hiz=normalhiz;
      jystp=20;
      Serial.println("Normal speed.");
      if (playmode==false) {
       delay(300);
      }
     }
    }
   }
  }
  else{
```

```
    prehiz=hiz;
  }
}
```

Mevcut servo motor açılarının hafızaya kaydedildiği fonksiyon aşağıda verilmiştir. Detaylar, önceki bölümlerde verilen kodlardaki gibidir.

```
void saveit() {
  joint1[NumMemory] = valcalc1;
  joint2[NumMemory] = valcalc2;
  joint3[NumMemory] = valcalc3;
  joint4[NumMemory] = valcalc4;

  prepos1=valcalc1;
  prepos2=valcalc2;
  prepos3=valcalc3;
  prepos4=valcalc4;

  if (NumMemory==1){
    pos11=joint1[0];
    pos21=joint2[0];
    pos31=joint3[0];
    pos41=joint4[0];
  }
}
```

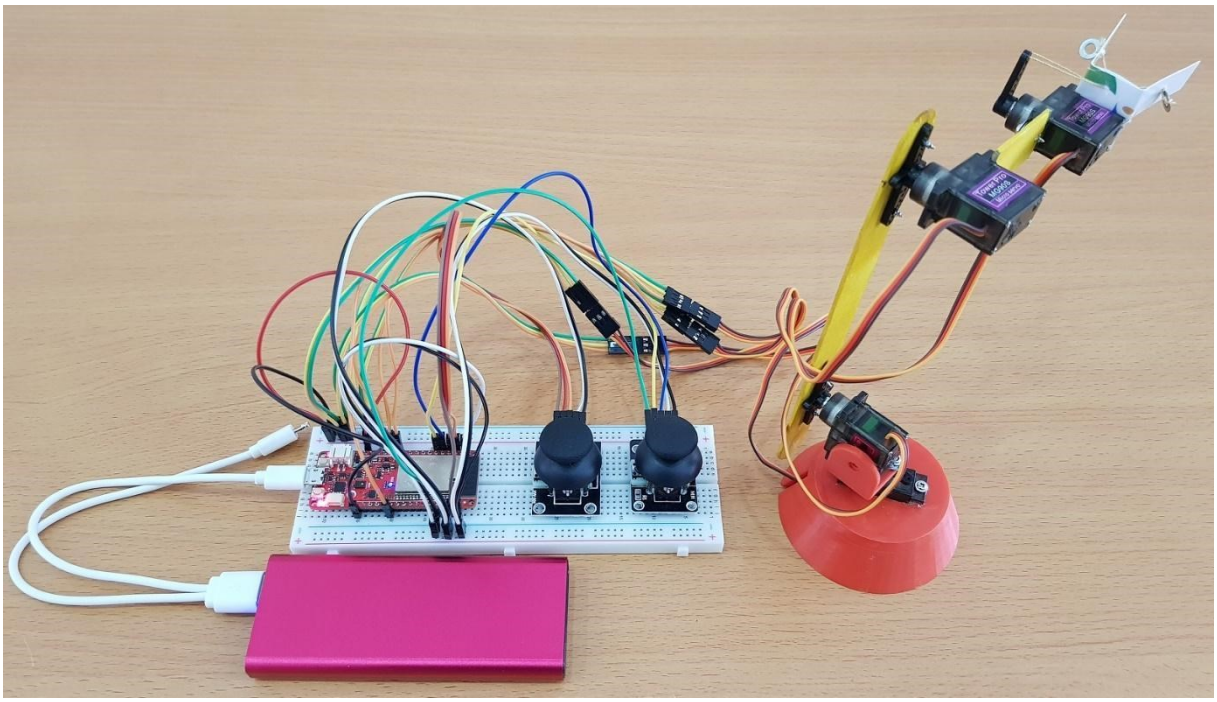
Hafızaya kaydedilen servo motor konumlarının tekrarlatıldığı fonksiyon aşağıda verilmiştir. Detaylar, önceki bölümlerde verilen kodlardaki gibidir.

```
void playit()
{
  float pos1=prepos1;
  float pos2=prepos2;
  float pos3=prepos3;
  float pos4=prepos4;

  float dif1=abs(joint1[Step]-pos1);
  float dif2=abs(joint2[Step]-pos2);
  float dif3=abs(joint3[Step]-pos3);
  float dif4=abs(joint4[Step]-pos4);
  float maxdif=0.0;

  // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için, aşağıda
kullanılmak üzere, dönüş açı mesafelerinin farklara bölüştürülmesi.
  if (dif1>maxdif) maxdif=dif1;
  if (dif2>maxdif) maxdif=dif2;
  if (dif3>maxdif) maxdif=dif3;
  if (dif4>maxdif) maxdif=dif4;
```

```
        maxdif=maxdif/hiz;
        if (maxdif<1.0) maxdif=1.0;

        float stp1=(joint1[Step]-pos1)/maxdif;
        float stp2=(joint2[Step]-pos2)/maxdif;
        float stp3=(joint3[Step]-pos3)/maxdif;
        float stp4=(joint4[Step]-pos4)/maxdif;
        for (int i=1;i<=int(maxdif);i++){// Motor dönüşlerinin olabildiğince düzgün ve sürekli
hissedilmesi için dönüş açı mesafeleri yukarıda tanımlanmış farklara bölüştürülüp servo
motorlara uygulanmıştır.
          pos1=pos1+stp1;
          pos2=pos2+stp2;
          pos3=pos3+stp3;
          pos4=pos4+stp4;
          if (i==int(maxdif)){
            pos1=joint1[Step];
            pos2=joint2[Step];
            pos3=joint3[Step];
            pos4=joint4[Step];
          }
          myservo1.writeMicroseconds(int(pos1));
          myservo2.writeMicroseconds(int(pos2));
          myservo3.writeMicroseconds(int(pos3));
          myservo4.writeMicroseconds(int(pos4));

          prepos1=pos1;
          prepos2=pos2;
          prepos3=pos3;
          prepos4=pos4;
          delay(10);

          if (ButtonMode<3) Button();
          else delay(5);
        }
        if (ButtonMode==3 && Step==NumMemory-1) { // Oynatma modunda durdurma işlemi.
          NumMemory = 0;
          Step=-1;
          ButtonMode = 0;
          playmode=false;
          Serial.println("Stopped.");
        }
        else{
          Step+=1;
          if (Step>NumMemory-1) Step=0; // Hafıza aşıldığında, ilk adıma dönüş.  Burada
istenirse “Step=NumMemory-1;” de yazılabilir ve hafızada sürekli en son adımda kalınır.
        }
      }
```

Adım 14: Program Deneyap Geliştirme Kartına yüklendikten sonra motorların kumanda kolu ile kontrolü yapılabilmekte, öğretme ve tekrarlatma modları denenebilmektedir (Resim 5.16).

Uyarı:
Motorların USB girişine aşırı yüklenmesini engellemek için harici bir batarya ile test edilmesi gerekmektedir. Aksi takdirde bilgisayarın USB portu zarar görebilir.



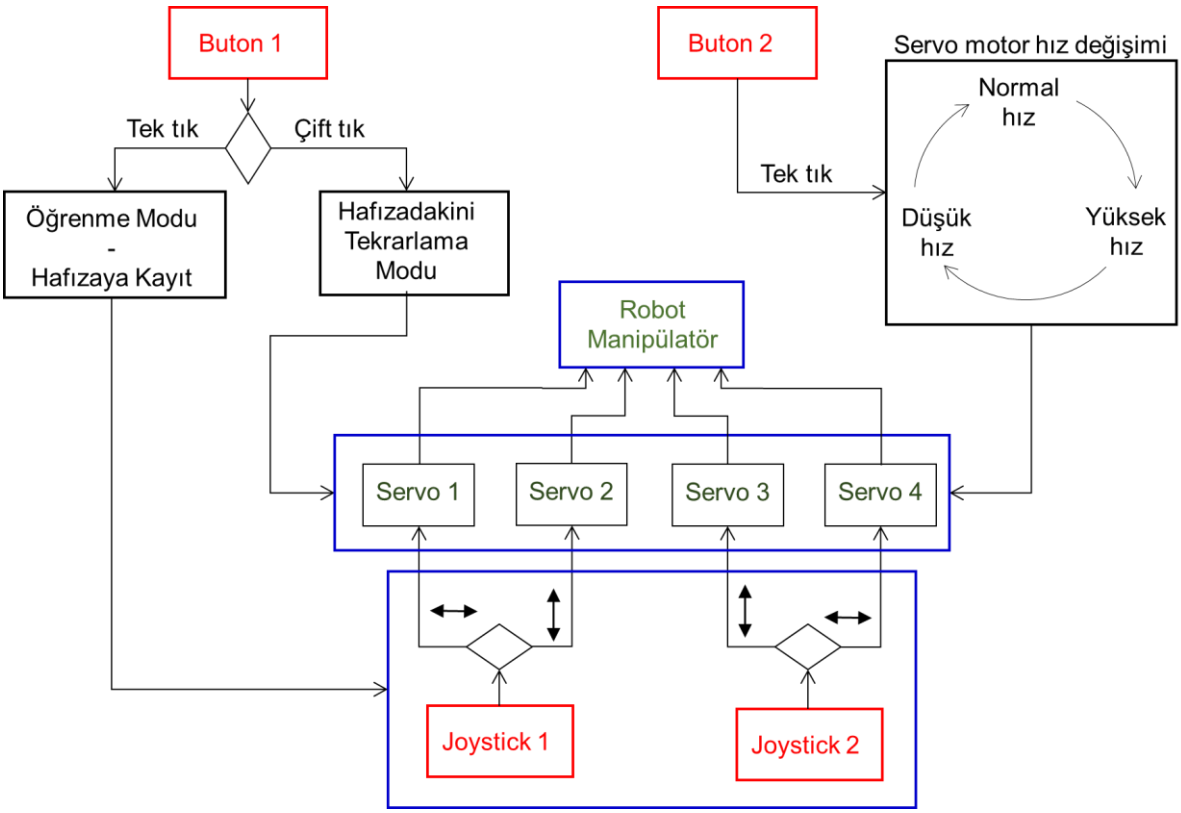
Resim 5.16. Tüm sistemin harici bir batarya ile test edilmesi

Robotun çalışma algoritması, Resim 5.17'de verilen blok diyagramında gösterilmektedir. Birinci kumanda kolu (soldaki) sağa sola hareket ettirildiğinde, taban servo motoru dönme hareketini gerçekleştirecektir. Birinci kumanda kolu ileri geri hareket ettirildiğinde ise iki numaralı servo motor hareket edecektir. İkinci kumanda kolu, ileri geri hareket ettirildiğinde, 3 numaralı servo motor dönme hareketini gerçekleştirecektir. İkinci kumanda kolu sağa sola hareket ettirildiğinde ise tutucu el kapanıp açılacaktır.

Birinci kumanda kolunun butonuna tek tıklama yapıldığında, o anki servo motor konumları hafızaya alınır (öğrenme modu). Bu her tekrarlandığında değişmiş her konum sıralı hafızaya alınır. Birinci kumanda kolunun butonuna çift tık yapıldığında, robot hafızadaki hareketleri sürekli olarak tekrarlayacaktır (oynatma modu). Aynı kumanda kolunun butonuna tekrar tek tık yapıldığında da robot yavaşlayıp duracaktır. Herhangi bir yeni konum bilgisi hafızaya girildiği anda önceki konum bilgilerinin hepsi silinmektedir. Eğer yeni konum bilgisi girilmez ve tekrardan birinci kumanda kolu butonuna çift tık yapılırsa robot hafızasındaki konum bilgilerine uygun olarak hareketleri sürekli tekrarlayacaktır.

Robotun hız kontrolü, ikinci kumanda kolu butonuna tek tıklama yapılması suretiyle gerçekleştirilmektedir. Her tıklamada eklem hızları değişecektir. Yavaş, normal ve hızlı olmak

üzere üç hız modu bulunmaktadır. Robot öğrenme veya oynatma modlarında iken ikinci kumanda kolunun butonuna tek tıklama yapıldığında, bir hız modundan diğerine geçilebilmektedir.



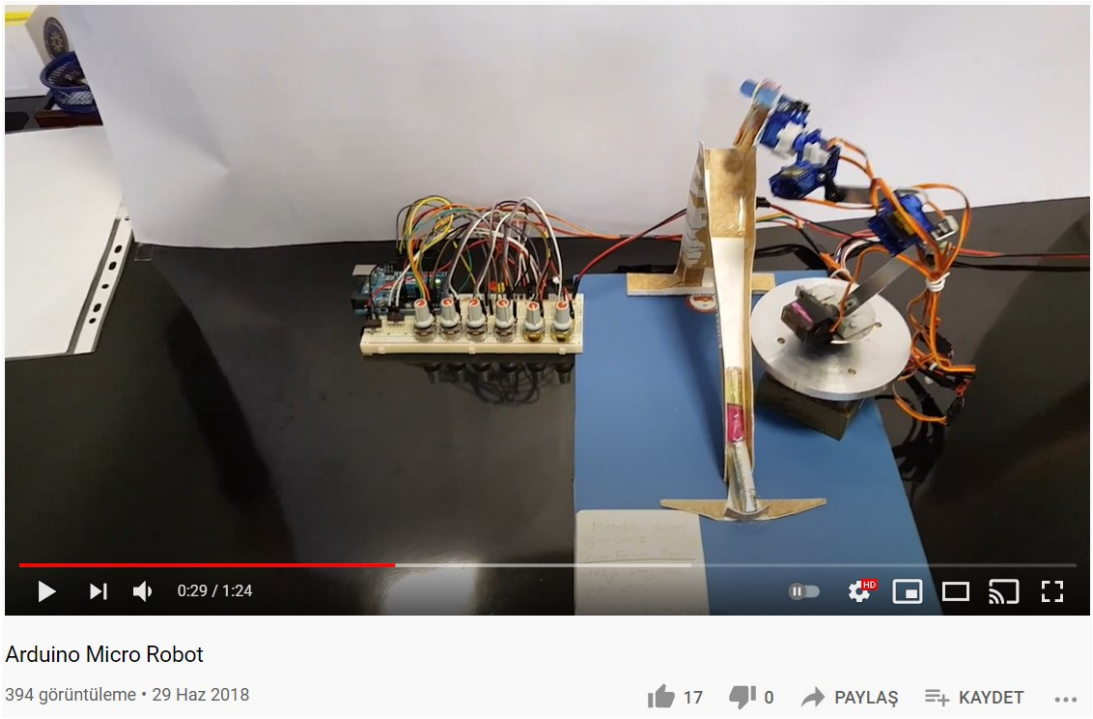
Resim 5.17. Robotun çalışma algoritmasını gösteren akış diyagramı.

2. TASARLA VE ÜRET

2.1. Tasarla: Üç Serbestlik Dereceli Robot için Çalışma Ortamı

Eğitmen, öğrencilerden ürettikleri robotla belli bir periyodik işlevi yerine getirebileceği veya sensörlerle otomatik olarak belli bir görevin yerine getirilebileceği bir sistemi tasarlamalarını ister. Örneğin, küçük parçaları bir noktadan alıp bir kutuya bırakma işlemi yaptırılabilir. Başka bir örnek olarak, aşağıdaki videodakine benzer bir uygulama yaptırılabilir (Resim 5.18).

Video Linki: HTTPS://WWW.YOUTUBE.COM/WATCH?V=QPYND2UCZTY



Resim 5.18. İnternet ortamında yayınlanmış örnek bir uygulama

Bu uygulamada 5 serbestlik dereceli robot kullanılmakta olmakla birlikte 3 serbestlik dereceli robotlara da kolaylıkla uyarlanabilir. Videoda görülen kızak modeli için, etrafımızda kolaylıkla bulabileceğimiz paket karton malzemeleri yeterli olacaktır. Robotun kaldırdığı objeler kartondan rulo şeklinde yapılmış ve saydam bantla kaplanmış hafif parçalardır.

2.2. Üret: Üç Serbestlik Dereceli Robot için Çalışma Ortamı

Öğrencilerden tasarla aşamasında yaptıkları tasarıma göre basit bir çalışma ortamını (yukarıda verilen örnekteki karton kaykay ve küçük silindirik parçalar gibi) üretmeleri istenir. Bu çalışma ortamının ne ve nasıl olacağı, eğitmene ve öğrencilerin hayal gücüne bırakılmıştır. Yapılan üretimler eğitmenler tarafından kontrol edilerek eksikler giderilir.

NOT:
Bu son TASARLA VE ÜRET bölümünde zaman sıkıntısı (yetiştirememe) durumu ile karşılaşılabilir. Bu herhangi bir sorun teşkil etmemelidir. Burada, yapılanları olduğu gibi bırakıp bir sonraki bölüme devam edilmesi tavsiye edilir.

3. DEĞERLENDİR

Öğrenciler ile bu hafta yapılan uygulamalar konusunda görüşmeler ve değerlendirmeler yapılmalı ve yeterli kazanımları elde edip etmedikleri belirlenmelidir. Aşağıdaki örnek sorular üzerinden tartışmalar gerçekleştirilebilir:

- Üç serbestlik dereceli bir robotun özellikleri nelerdir?
- Tasarım süreci ile ilgili bir problem yaşadınız mı? Yaşadıysanız bunu nasıl çözdünüz?
- Devrelerin fiziki kurulumunda zorlandığınız yerler oldu mu?
- Bu derste bahsi geçen üç serbestlik dereceli robotların pratik hayattaki uygulamaları (veya endüstriyel uygulamaları) neler olabilir? Örnek uygulamalar:
    - İmalat endüstrisinde üç serbestlik dereceli robotlara rastlanmaktadır. Bu tür robotların çok çeşitli amaçlı kullanımlarına rastlanmaktadır. Örneğin, bir konveyörden gelen paketlerin alınıp bir araca yüklenmesi, birkaç parçanın bir araya getirilip montajının yapılması, bir parçanın başka bir parçaya kaynak yapılması gibi.
    - İki serbestlik dereceli gimbal sistemlerinden daha da yaygın olan üç serbestlik dereceli gimbal sistemleri günümüzde çok yaygın olarak kullanılmaktadır. IMU sensörlerinden alınan geri besleme sinyalleri ile üç motor kontrol edilerek tam bir denge sistemi gerçekleştirilmiş olur, böylece kamera görüntüsü çok daha net olarak alınmış olur.
    -

4. İLAVE ETKİNLİK

Bu kısımda, mevcut zamana bağlı olarak, istenildiğinde, öğrencilerden üç serbestlik dereceli farklı ve yeni bir sistem uygulamasının tasarımı, üretimi ve uygulaması yaptırtılabilir.

Örnek uygulama: Eğlenceli bir uygulama olarak, dans eden bir robot uygulaması yapılabilir. Kulağa hoş gelen bir müzik ile senkronize robota ritmik hareketler öğretilir ve müzik eşliğinde hareketleri tekrarlatılır. Yalnız herhangi bir yanlış anlaşılma olmaması için, şunu belirtmeliyiz ki, bu uygulamada herhangi bir ses tespiti yapmadan, sadece robota doğrudan periodik hareketler öğretilmesi suretiyle bu uygulama yapılabilir. Aşağıda linki verilen Youtube videosundaki gibi, çok hoş görsel bir şölen yapılabilir:

https://www.youtube.com/watch?v=UGIMJofyn54

Burada başka ne tür uygulamalar yaptırılacağı hususu, eğitimciye ve öğrencilerin hayal gücüne bırakılmıştır.

6. Bölüm: Beş Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap Geliştirme Kartının temel özelliklerini bilir.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde servo motor kullanımı bilgisine sahiptir.
- Öğrenciler Deneyap Geliştirme Kartı üzerinde kumanda kolu kullanımı bilgisine sahiptir.
- Öğrenciler, 3 serbestlik dereceli robot yapısını bilir.

Haftanın Kazanımları:

- Öğrenciler, çeşitli amaçlı uygulamalar için 5 serbestlik dereceli bir robot manipülatörün mekanik ve elektronik donanımını oluşturabilir.
- Öğrenciler, bir robot manipülatörü 3 adet kumanda kolu ile kontrol etmeyi ve bir görevi öğretebilmek için gerekli yazılımı Deneyap Geliştirme Kartı üzerinde kodlayabilir.
- Öğrenciler, 5 serbestlik dereceli robot üretimi için gerekli mekanik tasarım ve üretim, elektronik tasarım ve kodlama işlemlerini yapabilir.
- Öğrenciler, 5 serbestlik dereceli robot ile çeşitli uygulamalar yapabilir.
- Öğrenciler, serbestlik derecesi artışının çalışma uzayına etkisini gözlemler ve deneyimler.

Haftanın Amacı:

Bu bölümde amaç, öğrencilerin 5 serbestlik dereceli (5 döner eklemli) ve uç kısmında kıskaç ele (gripper) sahip bir robot manipülatörün mekanik, elektronik ve yazılım çalışmalarını eğlenceli uygulamalar ile öğrenmelerini sağlamaktır. Ayrıca, bu robot manipülatör uygulamasında, kumanda kolu modüllerinin kullanımı ile açık çevrim bir kontrol çalışması amaçlanmıştır.

Kullanılacak Malzemeler:

1. Deneyap Geliştirme Kartı, 1 adet
2. Servo motor, 6 adet
3. XY Kumanda kolu Modülü, 3 adet
4. Breadboard, 1 adet
5. Çeşitli renklerde erkek-dişi ve erkek-erkek jumper kabloları, 40+ adet
6. M2 veya M3 rondela (pul), 15 adet
7. Tahta tutkalı, 1 adet
8. Abeslang, 1 paket
9. Esnek plastik levha
10. Naylon dikiş ipi
11. Mikro USB kablosu

12. Harici 5V USB Batarya
13. Çift taraflı silikon bant ve makas
14. 3B Yazıcı ve Filament

Haftanın İşlenişi:

Gözle: Bundan önceki haftalarda uygulaması yapılan sistemlerin Deneyap Geliştirme Kartı üzerinde kullanımlarının ve temel kodlamalarının kısaca üzerinden geçilerek öğrencilere hatırlatılması.

Tasarla: Beş serbestlik dereceli robot mekanizmasının tasarlanması ve Deneyap Geliştirme Kartı programının bu robotun kullanımı için yazılması. Burada istenirse, bu bölüm' deki adımlar sıra ile uygulanabilir veya bir önceki bölümde yapılmış olan 3 serbestlik dereceli robotu modifiye ederek 5 serbestlik dereceli bir robot manipülatörün oluşturulması yoluna da gidilebilir. Bu bölüm için gerekli zaman, önceki bölümden daha kısa olacağı için, öğrencilerin ilave bir uygulama yapabilmesi için yeterli bir zamana sahip olabilecekleri ön görülmektedir.

Üret: Tasarlanan donanımın gerçekleştirilmesi.

Uygula: Üretilen donanımın Deneyap Geliştirme Kartı ile kontrolünün yapılması ve test edilmesi.

Değerlendir: Öğrenciler ile bu uygulama konusunda tartışma ve değerlendirmelerin yapılması.

1. GÖZLE VE UYGULA

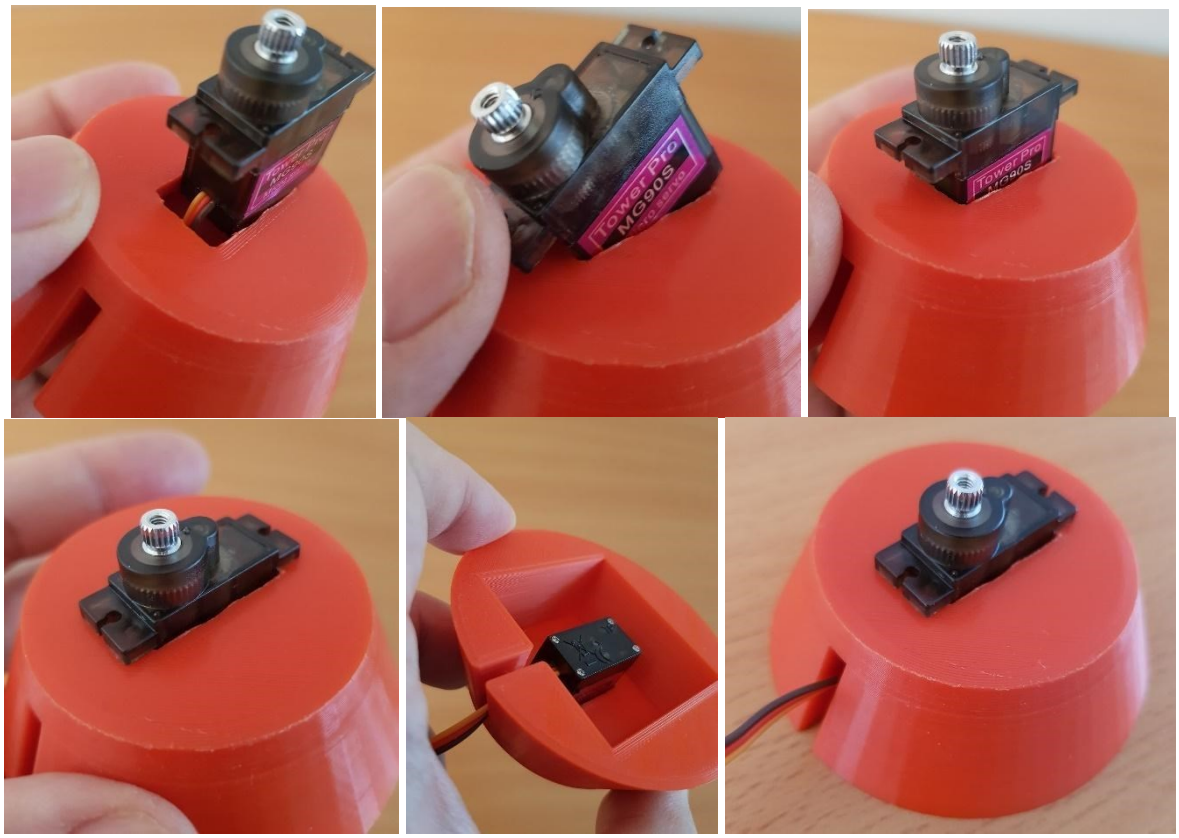
1.1. Gözle: Beş Serbestlik Dereceli Robotların Yapısı ve İşlevleri

Dersin başlangıcında eğitmen öğrencilerin 5 serbestlik dereceli robotlar ile ilgili tartışmalarını, ilk dersin konularını hatırlamalarını sağlamalıdır. Özellikle, neden 3 serbestlik dereceli yerine 5 ve daha yüksek serbestlik dereceli robotlara ihtiyaç duyulmaktadır, bu konuda öğrencilerle beyin jimnastiği yapılmalıdır. Bu aşamada internetteki örnek 5 serbestlik dereceli robotlar öğrencilere gösterilebilir.

1.2. Uygula: Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü

Uygulama Adımları:

Adım 1: 1 numaralı servo motor 3B yazıcıdan basılmış taban parçasına Resim 6.1'de gösterildiği gibi dikkatlice yerleştirilir. Bu aşamada servo motorun kablosuna dikkat edilmelidir. Kabloda meydana gelebilecek bir deformasyon motorun çalışmamasına neden olabilmektedir. Motor yerleştirildikten sonra taban parçasına vidalanır (Resim 6.2).



Resim 6.1. Servo motorun 3B yazıcıdan basılmış taban parçasına yerleştirilmesi



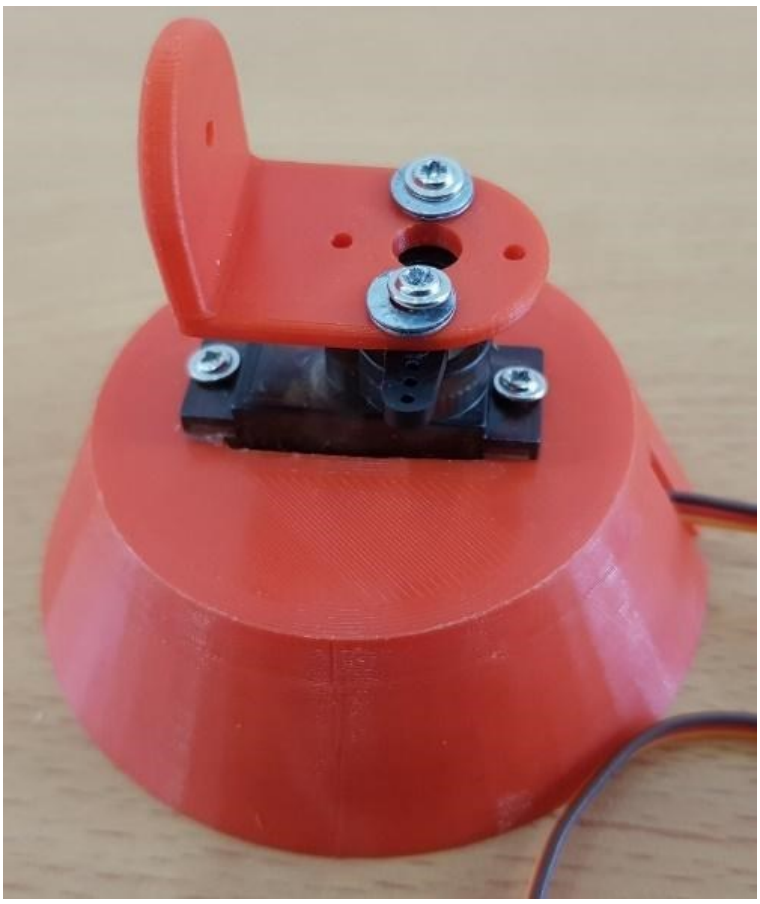
Resim 6.2. Taban parçaya vidalanmış servo motor.

Adım 2: İki numaralı servo motoru bağlamak için 3B yazıcıda basılmış bir L-profil kullanılır (Resim 6.3). Burada servo motor plastiklerinden T şeklinde olanı kullanılır. T şeklinde plastik bulunamazsa diğerleri de kullanılabilir.



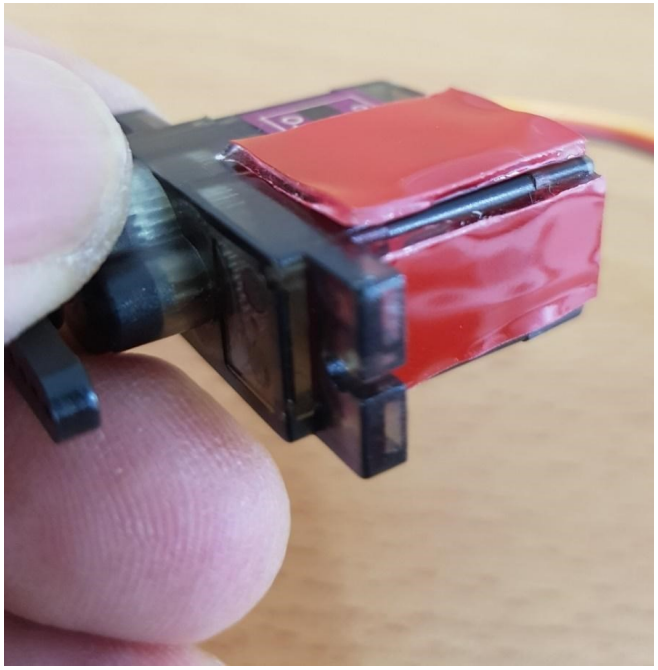
Resim 6.3. 3B-yazıcıda basılan L-profil ve üzerine vidalanmış servo motor plastik parçası

Adım 3: Hazırlanan L-profil, taban motorunun (Motor 1) miline Resim 6.4'te gösterildiği gibi yerleştirilir.



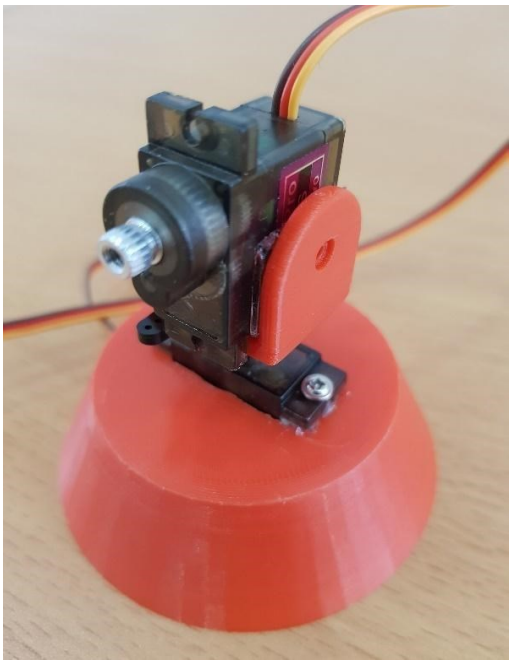
Resim 6.4. Hazırlanan L-profil parçanın taban motoru miline yerleştirilmesi

Adım 4: İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılır (Resim 6.5). Burada, önceki bölümlerdekine benzer olarak, silikon tabanca ile sıkılan yapıştırıcı silikon malzemede kullanılabilir. Bu, eğitmen ve öğrencinin tercihine bırakılmıştır.



Resim 6.5. İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılması

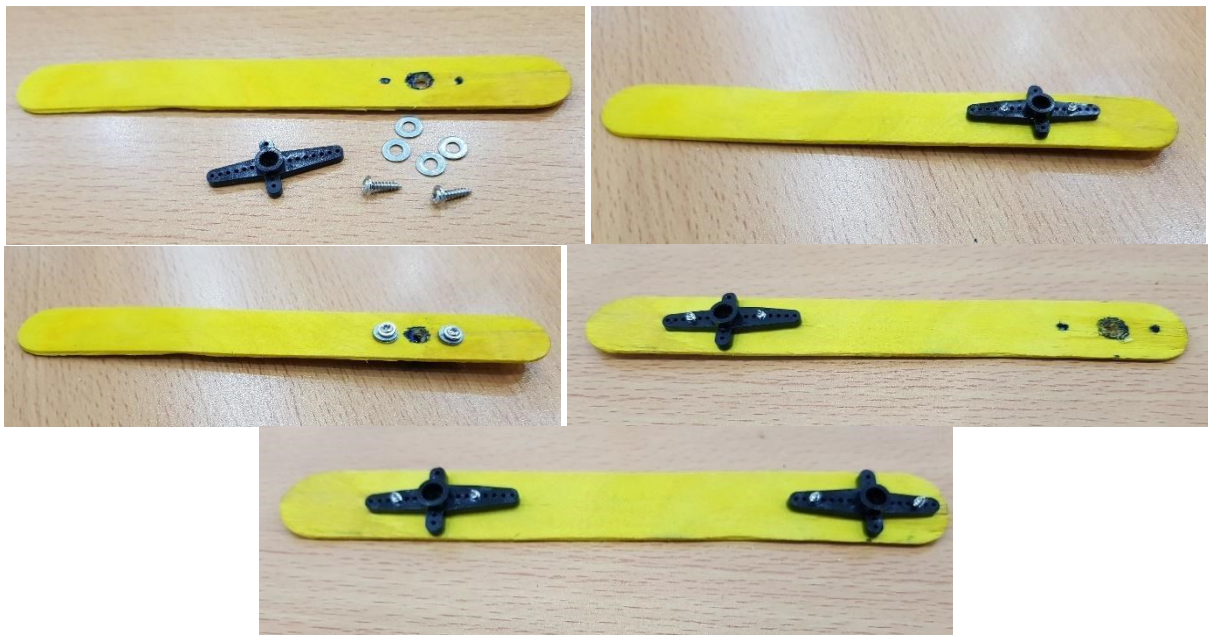
Adım 5: İki numaralı servo motora yapıştırılmış çift taraflı silikon bantın kırmızı naylonları çıkartılır ve servo motor L-profil parçanın üzerine yapıştırılır (Resim 6.14).



Resim 6.6. İki numaralı servo motorun L-profil parça üzerine konumlandırılması.

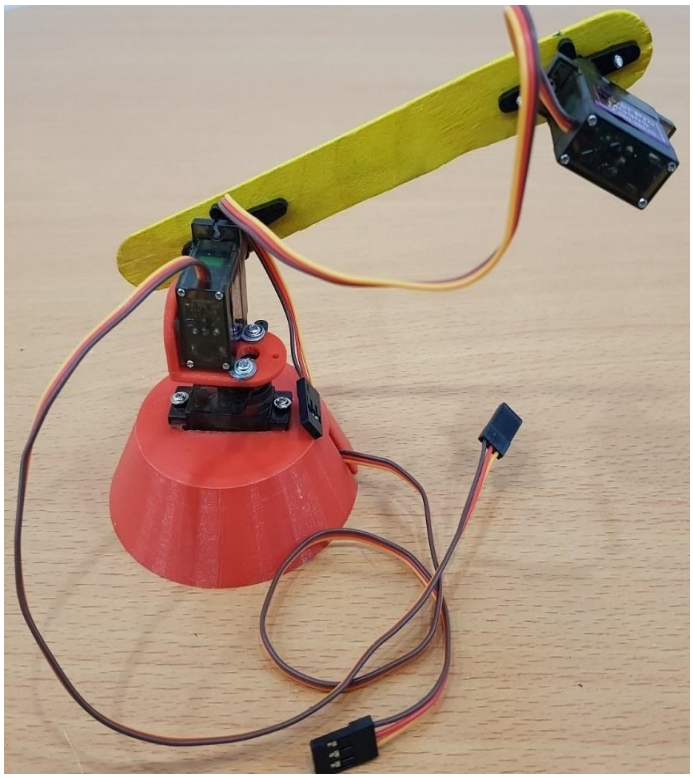
Adım 6: Bir adet abeslang Resim 6.7'deki gibi hazırlanır. Abeslangların uçlarına ince bir alet ile kolayca delik delinebilmektedir, Delikler delinirken abeslangların çatlatılmamasına dikkat edilmelidir. Eğer mümkünse, bu delikler bir matkap yardımıyla delinmelidir. Resimdeki yapıda bir adet abeslang kullanılmıştır. Daha rijit bir sistem için birden fazla abeslang tutkal ile üst üste yapıştırılarak da kullanılabilir. Eğitmenler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin

kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, eğitmenlere ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.



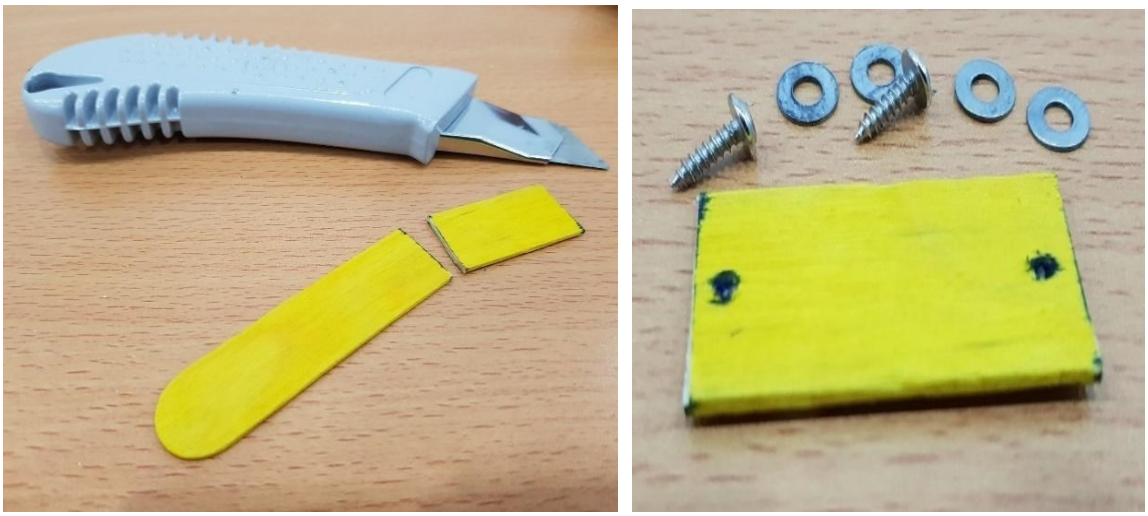
Resim 6.7. Abeslanglara servo motor T-plastik parçalarının takılması

Adım 7: Hazırlanmış olan abeslang Resim 6.8'de gösterildiği gibi 2 numaralı servo motora takılır. Burada hareket açı aralıkları deneme yanılma yolu ile belirlenerek abeslang yerleştirilir ve motor mili vidası takılarak sıkıştırılır. Abeslangın diğer tarafına da yine şekilde gösterildiği gibi 3numaralı servo motor yerleştirilir.



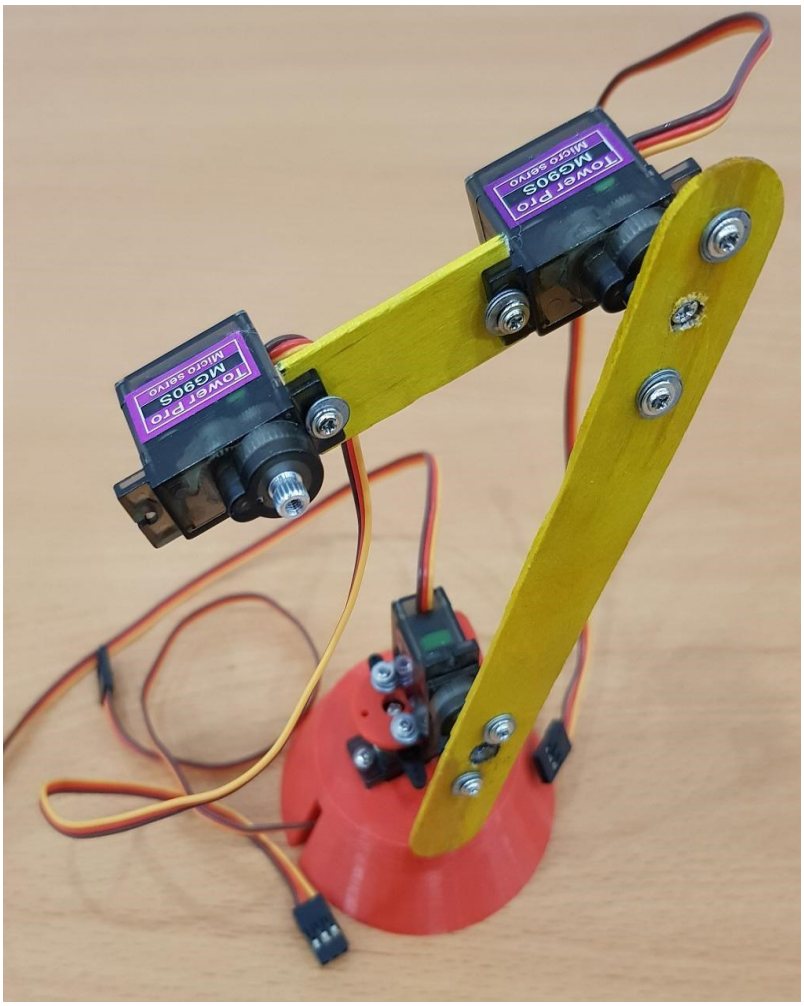
Resim 6.8. Abeslangın 2 numaralı servo motora ve 3 numaralı motorun abeslanga monte edilmesi

Adım 8: Başka bir abeslangtan kısa bir parça kesilir (Resim 6.9).



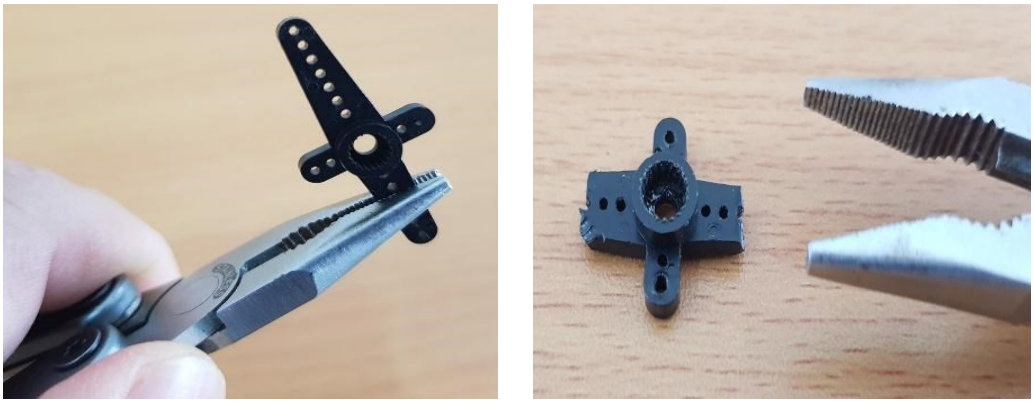
Resim 6.9. Kısa bir parçanın abeslangtan kesilmesi ve hazırlanması

Adım 9: Hazırlanmış olan kısa parça ile 3 ve 4 numaralı servo motorlar birbirlerine bağlanır (Resim 6.10).



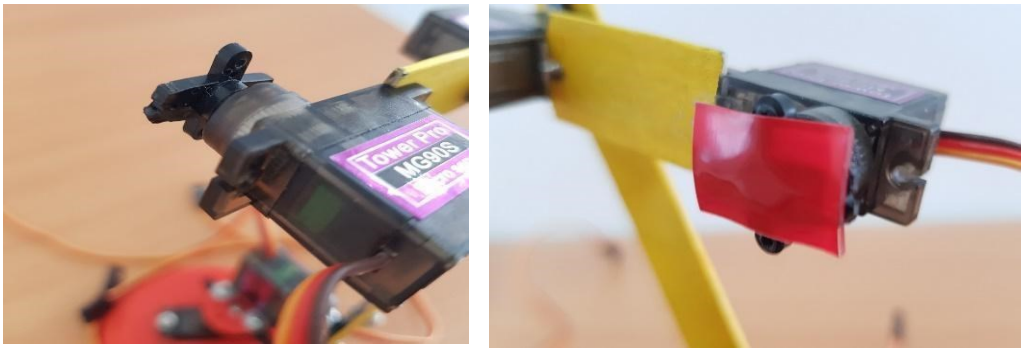
Resim 6.10. Kısa parça ile 3 ve 4 numaralı servo motorların birbirlerine bağlanması

Adım 10. Servo motorlarla birlikte gelen T-plastik parçanın uzun kenarları kırılarak kısaltılır (Resim 6.11). Böyle 2 adet parça hazırlanır.



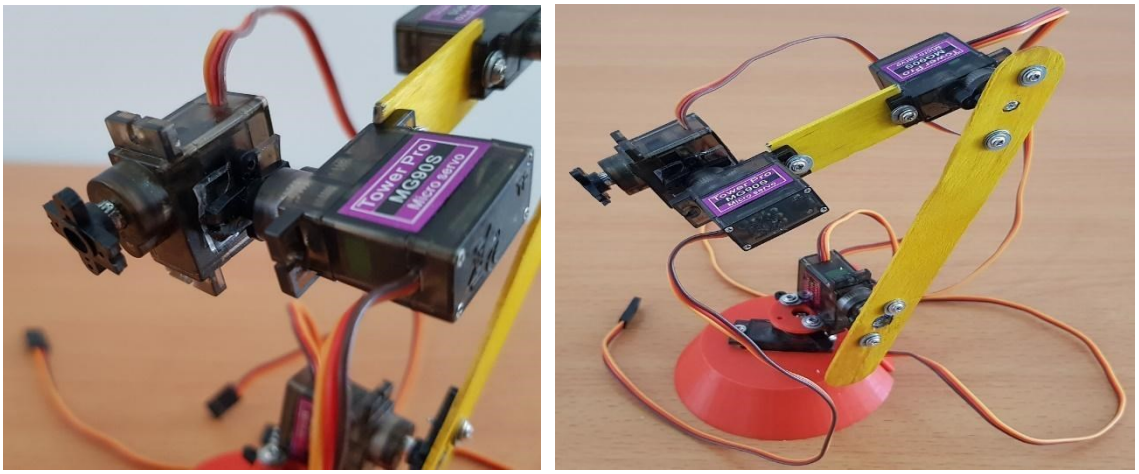
Resim 6.11. T-plastik parçanın uzun kenarları kırılarak elde edilen parça.

Adım 11. Elde edilen kısa plastik parça 4 numaralı servo motor miline takılır ve bir parça çift taraflı silikon bant (tercihen uygun başka bir malzeme, örneğin silikon tabanca ile sıkılan silikon yapıştırıcı), plastik parçanın yüzeyine yapıştırılır (Resim 6.12).



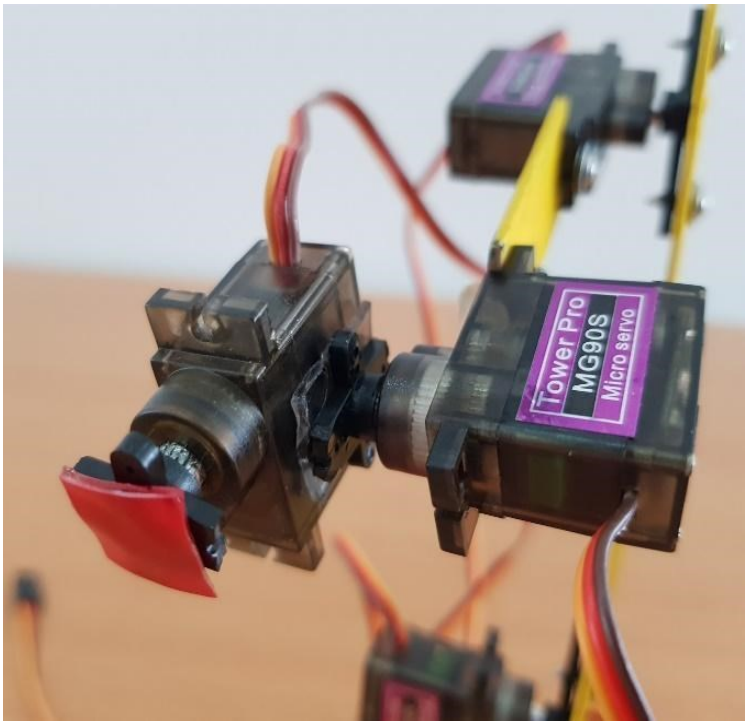
Resim 6.12. Dördüncü motorun milindeki kısa plastik parça ve çift taraflı silikon bant.

Adım 12. İkinci küçük plastik parça, 5 numralı servo motorun miline takılır. Çift taraflı silikon bandın kırmızı plastiği çıkartılıp 5 numaralı servo motor yan yüzeyinden 4 numaralı servo motorun milindeki plastiğin yüzeyine yapıştırılır (Resim 6.13).



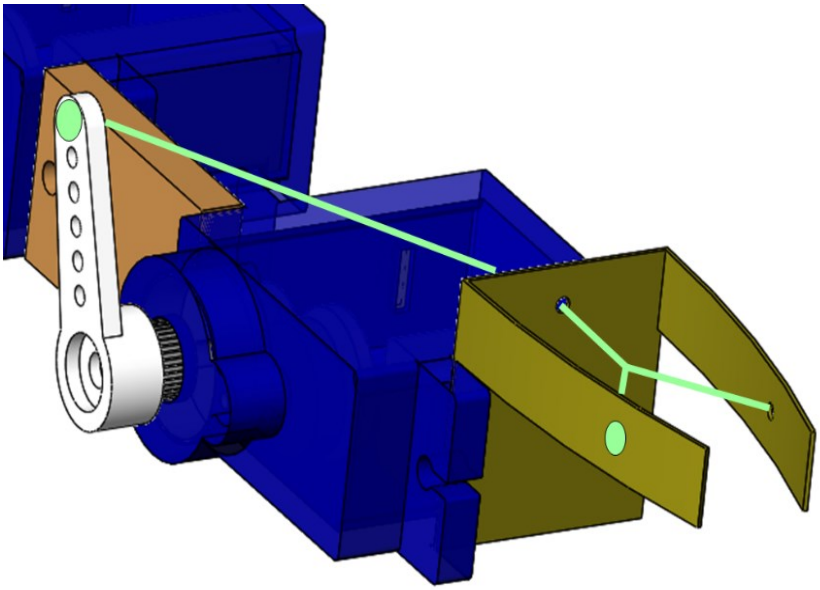
Resim 6.13. Beşinci motorun konumlandırılması.

Adım 13. Bir parça çift taraflı silikon bant (tercihen uygun başka bir malzeme, örneğin silikon tabanca ile sıkılan silikon yapıştırıcı), 5 numaralı servo motorun milindeki plastik parçanın yüzeyine yapıştırılır (Resim 6.14).



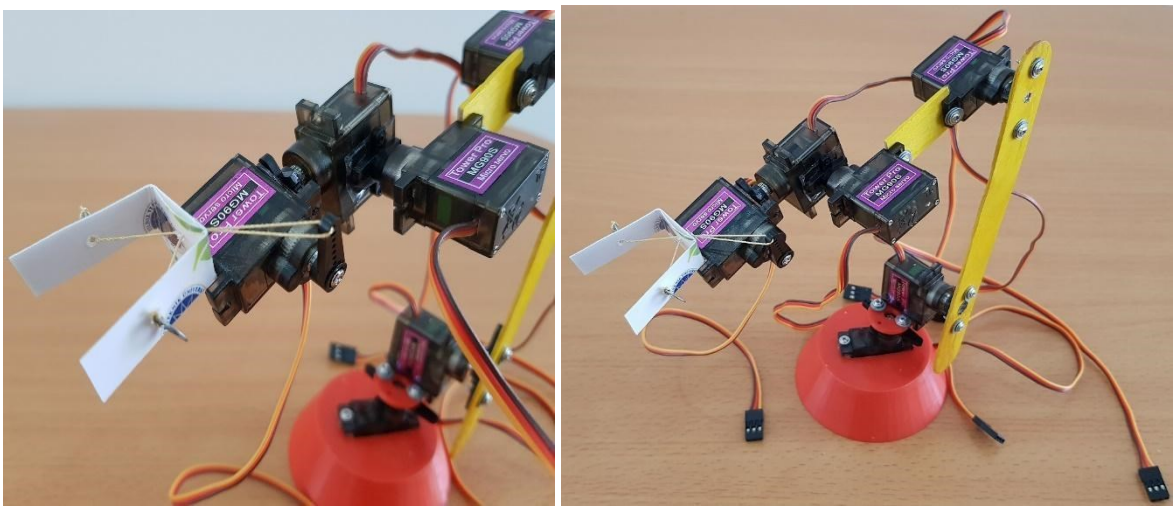
Resim 6.14. Beş numaralı motorun milindeki plastik parçaya çift taraflı silikon bant yapıştırılması

Adım 14: Tutucu (Gripper) tasarımında (Resim 6.15) bir servo motor, esnek plastik malzemeden kesilmiş ve bükülmüş tutucu parçası ve açı değişimi işlemini gerçekleştirmek için servo motorun hareketini tutucuya aktaracak naylon ip kullanılmaktadır. Esnek plastik kart, tutucu şeklinde kesilerek bükülür ve Resim 6.10'da gösterildiği gibi servo motora yapıştırılır. Yine şekilde gösterildiği gibi naylon ip de tutucu ile motor arasına bağlanır. Motorun dönme yönüne göre tutucu elinin aç/kapa hareketini yaptığı gözlemlenir.



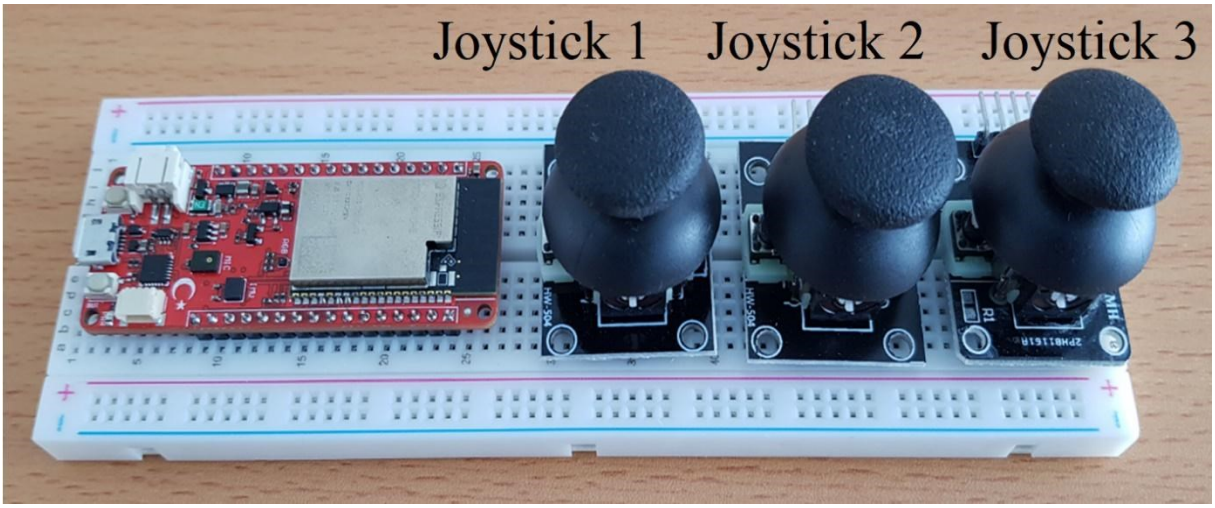
Resim 6.15. Tutucu tasarımı

Adım 15. Bir parça çift taraflı silikon bant, 5 numaralı servo motorun milindeki plastik parçanın yüzeyine yapıştırılır (Resim 6.16).



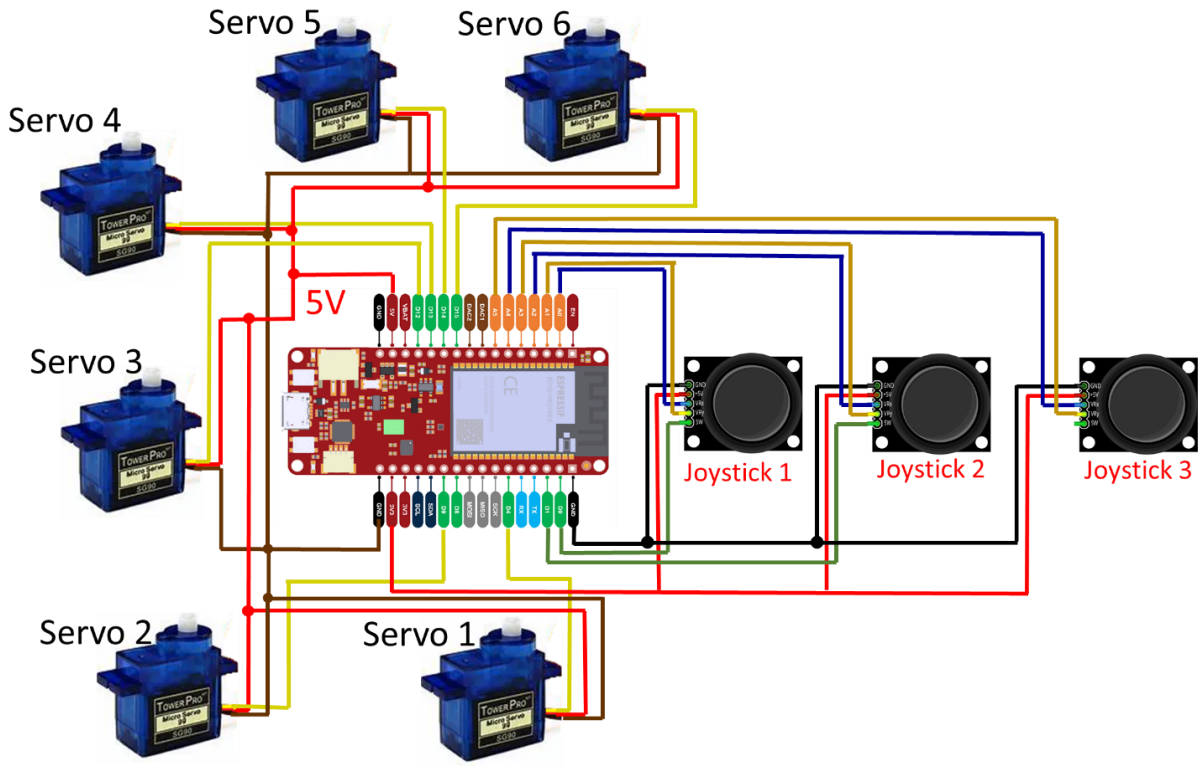
Resim 6.16. Tutucusu yerleştirilmiş robot manipülatör.

Adım 16: Deneyap Geliştirme Kartı breadboard üzerine yerleştirilir. Kumanda kollarının alt tabanlarına çift taraflı silikon bant yerleştirilir ve breadboard üzerine Resim 6.17'de gösterildiği gibi yapıştırılır.

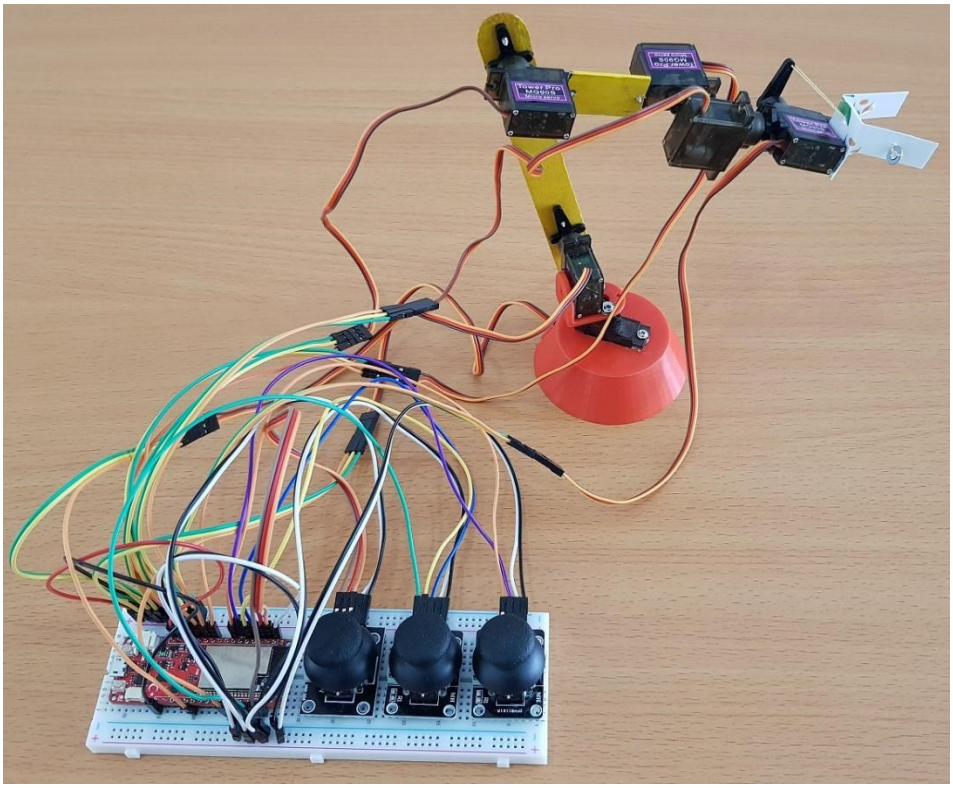


Resim 6.17. Breadboard üzerine konumlandırılmış kumanda kolları

Adım 17: Kumanda kolları ve servo motorların Deneyap Geliştirme Kartına bağlantıları Resim 6.18'te verilmiştir.



Resim 6.18. Uygulama 1 devre şeması



Resim 6.19. Beş serbestlik dereceli robot manipülatörün bağlantılarının yapılmış genel fotoğrafı

Adım 18: Deneyap Geliştirme Kartın bilgisayarla bağlantısı yapılır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
#include "deneyap.h"
#include <Deneyap_Servo.h>// Eğer bu kütüphane çalışmaz ise <ESP32Servo.h> kütüphanesi denenebilir.
// Dijital kumanda kollarının button pin bağlantıları:
#define SW1_pin D0
#define SW2_pin D1

// kumanda kollarının eksen analog bağlantı pinleri
#define X1_pin A0
#define Y1_pin A1
#define X2_pin A2
#define Y2_pin A3
#define X3_pin A4
#define Y3_pin A5

// Dijital servo pin bağlantıları:
#define SERVOPIN1 D4
#define SERVOPIN2 D9
#define SERVOPIN3 D12
#define SERVOPIN4 D13
#define SERVOPIN5 D14
#define SERVOPIN6 D15

// Motor dönüş açı alt ve üst limitleri:
//Aşağıdaki teta değerleri 0-3000 arasında kalibre edilmiştir.  3000 değeri 180 dereceye karşılık gelmektedir.
int teta1min=0;
int teta1max=3000;
int teta2min=0;
int teta2max=3000;
int teta3min=500;
int teta3max=2500;
int teta4min=0;
int teta4max=3000;
int teta5min=0;
int teta5max=3000;
int teta6min=1500;
int teta6max=2000;

// Motor dönüş açıları için değişkenlerin tanımlanması
int valin1, valin2, valin3, valin4, valin5, valin6;
int valcalc1=1500;//Başlangıç açı değerleri
int valcalc2=1500;
int valcalc3=1500;
int valcalc4=1500;
```

```
int valcalc5=1500;
int valcalc6=2000;

int jystp=20; // kumanda kollarıyla motor hareketleri sağlandığında her bir döngüde istenen açı değeri (bu yazılımda 3000 değerinin 360 dereceye karşılık geldiğini unutmayın.

bool playmode=false; // Robotun tekrarlama (oynama) modu değişkeni.  Değer false olduğunda, öğretme modunda demektir.
int ButtonMode=0; // Buton modu değişkeninin tanımlanması.  Sıfır olduğunda, öğretme modunda demektir.
unsigned long CurrentMilliSec = millis(); // milis() fonksiyonu, şu anki saati milisecond cinsinden vermektedir.  Zamanlayıcı olarak bu fonksiyonu kullanmaktayız.  Herhangi bir anda, o anki zamanı bir long değişkenine atayabiliriz.
long PreMilliSec=0; // Bir evvelki zaman (milisaniye cinsinden)
int NumMemory=0; // Hafıza sayacı (Maksimum 50 olabilir.  Deneyap kartının kapasitesine bağlı olarak bu sınır 50'den yukarıda olabilir.)
int PreNumMemory=0; // Bir evvelki moddaki hafıza değeri.
int Step=-1; // Öğretme modunda, servo pozisyonlarını hafızaya alırken kullanılan bir değişken.
// Servo pozisyonlarının hafızaya alındığı 50'lik seriye sahip değişkenler.
int joint1[50];
int joint2[50];
int joint3[50];
int joint4[50];
int joint5[50];
int joint6[50];
int maxarray=49; //Hafıza hatası vermeyi önlemek için 49 değerine sahip bir limit değişkeni.
int prehiz=10; // Bir önceki hız değerinin atanacağı değişken tanımlaması.
int normalhiz=10; // Normal hız değişkeni
// hiz=1 çok yavaş, 5 yavaş, 10 normal, 15 hızlı, 20+ çok hızlı
int hiz=10; // Şu anki hız değişkeni
int prepos1, prepos2, prepos3, prepos4, prepos5, prepos6; // bir evvelki motor konumları.
int pos11,pos21,pos31,pos41,pos51,pos61; // Motor konumları ile ilgili ikincil (yardımcı) değişkenler.

Servo myservo1, myservo2, myservo3, myservo4 myservo5, myservo6; // Servo motorların tanımlanması.

void setup()
{
  myservo1.attach(SERVOPIN1); //Eğer <ESP32Servo.h> kütüphanesi kullanırsanız, bu ve aşağıdaki 5 satırda parantez içerisindeki 2'inci parametreyi silin, tek parametre yeterlidir.
  myservo2.attach(SERVOPIN2,1);
  myservo3.attach(SERVOPIN3,2);
  myservo4.attach(SERVOPIN4,3);
```

```
  myservo5.attach(SERVOPIN5,4);
  myservo6.attach(SERVOPIN6,5);

  // Joystik butonlarının pinlerinin tanımlanması (Normalde HIGH).
  pinMode(SW1_pin, INPUT_PULLUP);
  pinMode(SW2_pin, INPUT_PULLUP);
 // kumanda kolu pinlerinin giriş olarak tanımlanması
  pinMode(X1_pin, INPUT);
  pinMode(Y1_pin, INPUT);
  pinMode(X2_pin, INPUT);
  pinMode(Y2_pin, INPUT);
  pinMode(X3_pin, INPUT);
  pinMode(Y3_pin, INPUT);

  // Joystik butonlarının pinlerinin değerleri normalde HIGH' a atandı.
  digitalWrite(SW1_pin, HIGH);
  digitalWrite(SW2_pin, HIGH);
  // Tercihe bağlı olarak, Seri port'a "Ready" yazdırılması
  Serial.begin(115200);
  Serial.println("Ready.");
 }
 void loop() {
  CurrentMilliSec = millis(); // Şu anki zaman değeri (ms)
  Button(); // Kendi yazdığımız Button fonksiyonunun çağırılması.
  if (playmode==false && Step==-1){ // Öğretme modunda iken,
   ReadAnalog(); //  Analog kanalların okunması için kendi yazdığımız ReadAnalog
fonksiyonunun çağırılması.
   // Servo motorlara dönüş komutlarının verilmesi:
   myservo1.writeMicroseconds(valcalc1);
   myservo2.writeMicroseconds(valcalc2);
   myservo3.writeMicroseconds(valcalc3);
   myservo4.writeMicroseconds(valcalc4);
   myservo5.writeMicroseconds(valcalc5);
   myservo6.writeMicroseconds(valcalc6);

   delay(int(300./hiz)); // Hız kontrolü için genel döngü içerisindeki gecikme ayarı.
  }
  else{
   playit(); //tekrarlama (veya oynama) modunda iken kendi yazdığımız Play
fonksiyonunun çağırılması.
  }
 }
 // kumanda kollarının ürettiği analog sinyallerin okunduğu ve motor açı değerlerinin bu
sinyallere göre değiştirildiği fonksiyon aşağıda verilmiştir.
 void ReadAnalog() {
  // Analog pinlere giren kumanda kolu değerlerinin okunması.
  valin1 = analogRead(Y1_pin);
```

```
  valin2 = analogRead(X1_pin);
  valin3 = analogRead(X2_pin);
  valin4 = analogRead(Y2_pin);
  valin5 = analogRead(X3_pin);
  valin6 = analogRead(Y3_pin);

  // Belli sınırlar dahilinde, Joystik değerlerine göre servo motor dönüş açılarının tayini.
  if (valin1<1000 && valcalc1>teta1min) valcalc1-=jystp;
  if (valin1>3000 && valcalc1<teta1max) valcalc1+=jystp;
  if (valin2<1000 && valcalc2>teta2min) valcalc2+=jystp;
  if (valin2>3000 && valcalc2<teta2max) valcalc2-=jystp;
  if (valin3<1000 && valcalc3>teta3min) valcalc3-=jystp;
  if (valin3>3000 && valcalc3<teta3max) valcalc3+=jystp;
  if (valin4<1000 && valcalc4>teta4min) valcalc4-=jystp;
  if (valin4>3000 && valcalc4<teta4max) valcalc4+=jystp;
  if (valin5<1000 && valcalc5>teta5min) valcalc5-=jystp;
  if (valin5>3000 && valcalc5<teta5max) valcalc5+=jystp;
  if (valin6<1000 && valcalc6>teta6min) valcalc6-=jystp;
  if (valin6>3000 && valcalc6<teta6max) valcalc6+=jystp;
}
```

Buton fonksiyonu oldukça kapsamlıdır. Burada 2 adet buton ile robot hareket modları tayin edilmektedir. Öğretme modunda iken kumanda kolu 1'in butona tek tıklama ile mevcut motor açılarını hafızaya alma işlemi yapılabilmekte, öğretme modundan tekrarlama moduna veya tersine yine Kumanda kolu 1'in butonuna çift tıklayarak geçilebilmektedir. Bu uygulamada da motorlara hızlı, normal ve yavaş olmak üzere 3 ayrı seviye tayin edilmiştir. Bu hız seviyeleri arasındaki geçişler Kumanda kolu 2'nin butonuna basılarak gerçekleştirilmektedir. Belirtilen buton aktivitelerini gerçekleştiren kodlar aşağıda verilmiştir.

```
void Button() { // Butona basılma durumu
  if (playmode==false){ // Eğer öğrenme modunda iken,
  if (digitalRead(SW1_pin) == 0){
    delay(100);
    if (digitalRead(SW1_pin) == 1) { // butona tıklanmış ise,
       if (ButtonMode == 0)
       {
         ButtonMode = 1; // İlk buton tek tık tespiti.
         PreMilliSec = CurrentMilliSec;
       }
       else if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec < 500))
       {
         ButtonMode = 2; // İlk buton çift tık tespiti.
       }
     }
   }
    if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec > 600)) {
       saveit(); // İlk buton tek tık tespit edildiğinde, motor konumları hafızaya alınır.
```

```
      NumMemory += 1;
      if (NumMemory>maxarray) NumMemory= maxarray;
      ButtonMode = 0;
      Serial.print("Saved: "); Serial.println(NumMemory);
    }
    else if (ButtonMode == 2 && (CurrentMilliSec - PreMilliSec > 600)) {
     if ((NumMemory>1 || PreNumMemory>1) && playmode==false){
      Serial.println("Playmode...");
      playmode = true; // Çift tık tespit edildiğinde tekrarlama (oynama) moduna geçilir.
      ButtonMode = 0;
      if (NumMemory<2) NumMemory=PreNumMemory; else
PreNumMemory=NumMemory; // Eğer hafızaya herhangi bir konum alınmamış olup ta
çift tık yapılırsa, bir önceki oynatılmış konum bilgilerinin tekrar kullanılması için devreye
alınması.
      // Motor açı konumlarının hafızaya alınması:
      joint1[0]=pos11;
      joint2[0]=pos21;
      joint3[0]=pos31;
      joint4[0]=pos41;
      joint5[0]=pos51;
      joint6[0]=pos61;

      Step=NumMemory-1;
      ReadAnalog();
      prepos1=valcalc1;
      prepos2=valcalc2;
      prepos3=valcalc3;
      prepos4=valcalc4;
     }
    }
    if (ButtonMode<3 && CurrentMilliSec - PreMilliSec > 1000) {
     ButtonMode = 0; // Tekrar öğretme moduna geçiş.
    }
  }
  else if (ButtonMode<3 && digitalRead(SW1_pin) == 0) {
    ButtonMode = 3;
    Serial.println("Stopping."); // Tekrarlamayı (Oynamayı) durdur.
    ReadAnalog();
    NumMemory+=1;
    // Durulan konumun hafızanın en son basamağına kaydedilmesi.
    joint1[NumMemory-1] = valcalc1;
    joint2[NumMemory-1] = valcalc2;
    joint3[NumMemory-1] = valcalc3;
    joint4[NumMemory-1] = valcalc4;
    joint5[NumMemory-1] = valcalc5;
    joint6[NumMemory-1] = valcalc6;
  }
```

```
  // İkinci butona her tık yapıldığında robot hareket hızı değiştirilir.  Normal, yavaş, hızlı
olmak üzere 3 ayrı hız için, aşağıdaki kod yazılmıştır.  Diğer detaylar, önceki
bölümlerdeki kodlarda anlatıldığı gibidir.
  if (digitalRead(SW2_pin) == 0) {
    if (prehiz==hiz) {
      if (hiz==normalhiz) {
        hiz=2*normalhiz;
          jystp=40;
        Serial.println("High speed.");
        if (playmode==false) {
          delay(300);
        }
      } else {
        if (hiz==(2*normalhiz)) {
          hiz=normalhiz/2;
          jystp=10;
          Serial.println("Low speed.");
          if (playmode==false) {
            delay(300);
          }
        }
        else if (hiz==(normalhiz/2)) {
          hiz=normalhiz;
          jystp=20;
          Serial.println("Normal speed.");
          if (playmode==false) {
            delay(300);
          }
        }
      }
    }
  }
  else{
    prehiz=hiz;
  }
}
```

Mevcut servo motor açılarının hafızaya kaydedildiği fonksiyon aşağıda verilmiştir.

```
void saveit() {
   joint1[NumMemory] = valcalc1;
   joint2[NumMemory] = valcalc2;
   joint3[NumMemory] = valcalc3;
   joint4[NumMemory] = valcalc4;
   joint5[NumMemory] = valcalc5;
   joint6[NumMemory] = valcalc6;
```

```
    prepos1=valcalc1;
    prepos2=valcalc2;
    prepos3=valcalc3;
    prepos4=valcalc4;
    prepos5=valcalc5;
    prepos6=valcalc6;


    if (NumMemory==1){
      pos11=joint1[0];
      pos21=joint2[0];
      pos31=joint3[0];
      pos41=joint4[0];
      pos51=joint5[0];
      pos61=joint6[0];

    }
  }
```

Hafızaya kaydedilen servo motor konumlarının tekrarlatıldığı fonksiyon aşağıda verilmiştir. Detaylar, önceki bölümlerde verilen kodlardaki gibidir.

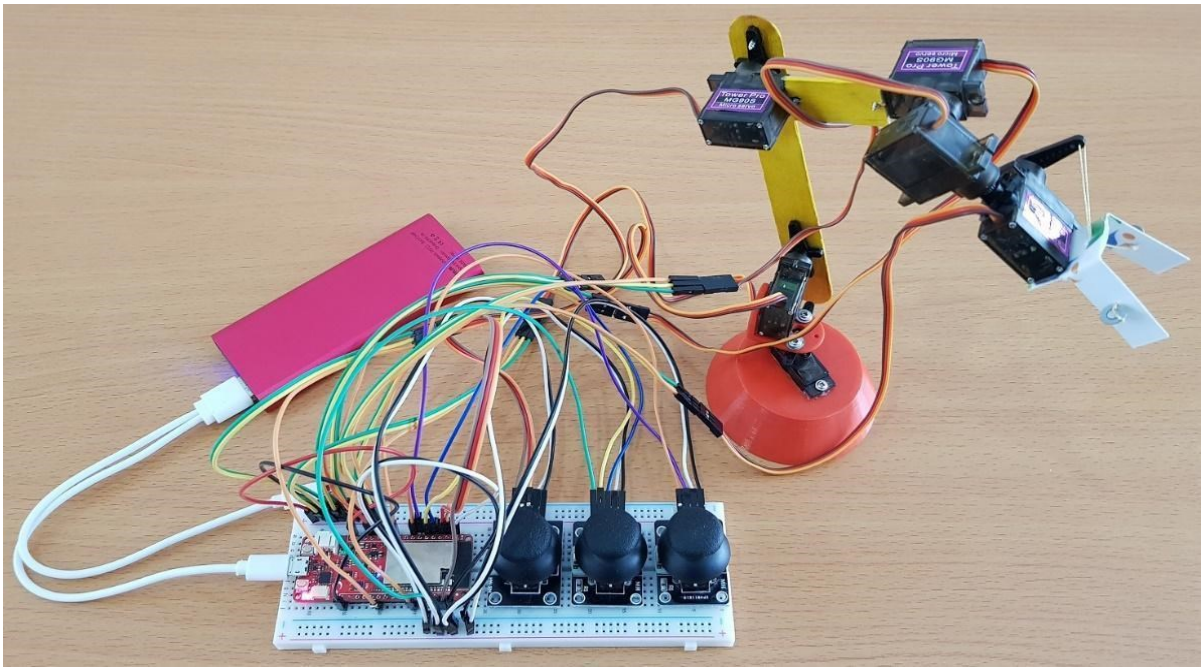
```
  void playit()
  {
    float pos1=prepos1;
    float pos2=prepos2;
    float pos3=prepos3;
    float pos4=prepos4;
    float pos5=prepos5;
    float pos6=prepos6;

    // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için dönüş açı
mesafeleri yukarıda tanımlanmış farklara bölüştürülüp servo motorlara uygulanmıştır.
    float dif1=abs(joint1[Step]-pos1);
    float dif2=abs(joint2[Step]-pos2);
    float dif3=abs(joint3[Step]-pos3);
    float dif4=abs(joint4[Step]-pos4);
    float dif5=abs(joint5[Step]-pos5);
    float dif6=abs(joint6[Step]-pos6);

    float maxdif=0.0;

    if (dif1>maxdif) maxdif=dif1;
    if (dif2>maxdif) maxdif=dif2;
    if (dif3>maxdif) maxdif=dif3;
    if (dif4>maxdif) maxdif=dif4;
    if (dif5>maxdif) maxdif=dif5;
    if (dif6>maxdif) maxdif=dif6;
```

```
maxdif=maxdif/hiz;
if (maxdif<1.0) maxdif=1.0;

float stp1=(joint1[Step]-pos1)/maxdif;
float stp2=(joint2[Step]-pos2)/maxdif;
float stp3=(joint3[Step]-pos3)/maxdif;
float stp4=(joint4[Step]-pos4)/maxdif;
float stp5=(joint5[Step]-pos5)/maxdif;
float stp6=(joint6[Step]-pos6)/maxdif;
for (int i=1;i<=int(maxdif);i++){ // Motor dönüşlerinin olabildiğince düzgün ve sürekli
```
hissedilmesi için dönüş açı mesafeleri yukarıda tanımlanmış farklara bölüştürülüp servo motorlara uygulanmıştır.
```
  pos1=pos1+stp1;
  pos2=pos2+stp2;
  pos3=pos3+stp3;
  pos4=pos4+stp4;
  pos5=pos5+stp5;
  pos6=pos6+stp6;

  if (i==int(maxdif)){
    pos1=joint1[Step];
    pos2=joint2[Step];
    pos3=joint3[Step];
    pos4=joint4[Step];
    pos5=joint5[Step];
    pos6=joint6[Step];

  }

  myservo1.writeMicroseconds(int(pos1));
  myservo2.writeMicroseconds(int(pos2));
  myservo3.writeMicroseconds(int(pos3));
  myservo4.writeMicroseconds(int(pos4));
  myservo5.writeMicroseconds(int(pos5));
  myservo6.writeMicroseconds(int(pos6));


  prepos1=pos1;
  prepos2=pos2;
  prepos3=pos3;
  prepos4=pos4;
  prepos5=pos5;
  prepos6=pos6;

  delay(10);
```

```
      if (ButtonMode<3) Button();
      else delay(5);
    }
    if (ButtonMode==3 && Step==NumMemory-1) { // Oynatma modunda durdurma işlemi.
      NumMemory = 0;
      Step=-1;
      ButtonMode = 0;
      playmode=false;
      Serial.println("Stopped.");
    }
    else{
      Step+=1;
      if (Step>NumMemory-1) Step=0; // Hafıza aşıldığında, ilk adıma dönüş.  Burada
istenirse “Step=NumMemory-1;” de yazılabilir ve hafızada sürekli en son adımda kalınır.
    }
  }
```

Adım 14: Program Deneyap Geliştirme Kartına yüklendikten sonra motorların kumanda kolu ile kontrolü yapılabilmekte, öğretme ve tekrarlatma modları denenebilmektedir (Resim 6.20).

Uyarı:
Motorların USB girişine aşırı yüklenmesini engellemek için harici bir batarya ile test edilmesi gerekmektedir. Aksi takdirde bilgisayarın USB portu zarar görebilir.



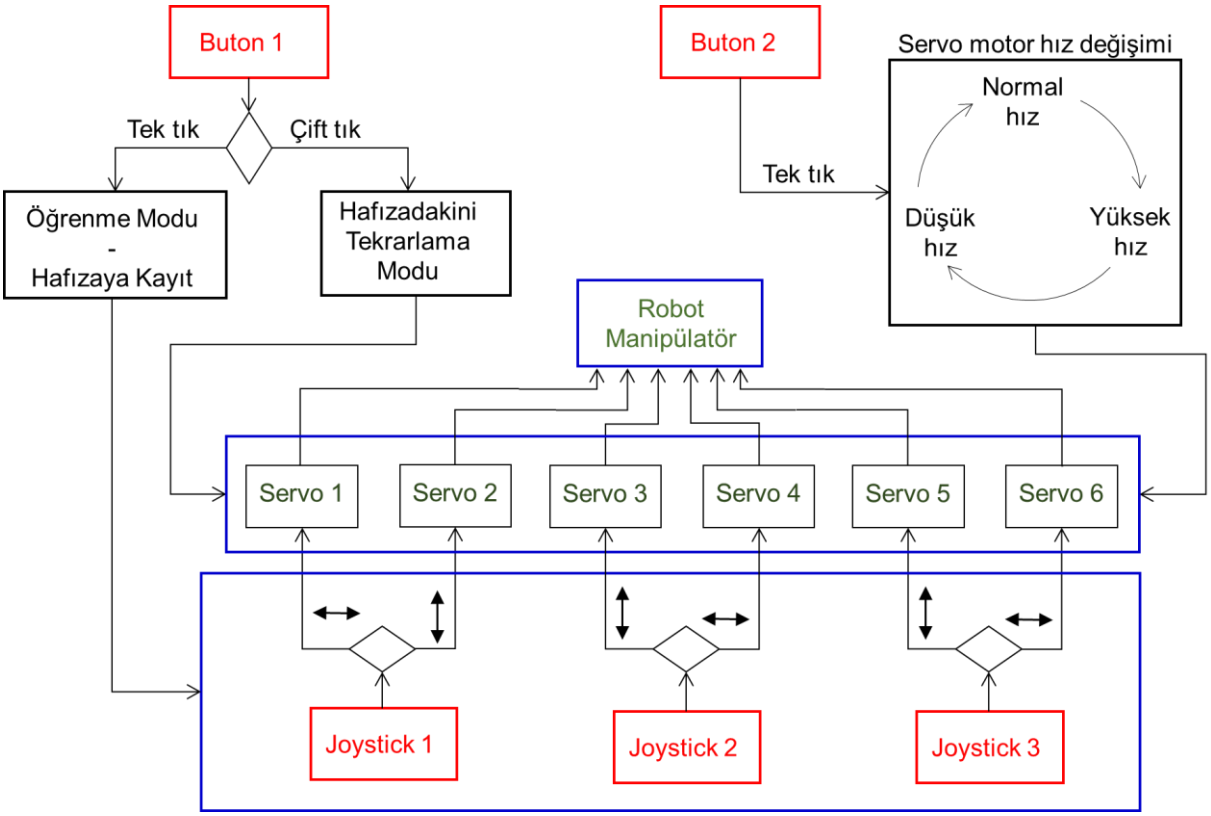
Resim 6.20. Tüm sistemin harici bir batarya ile test edilmesi

Robotun çalışma algoritması, Resim 6.21’ de verilen blok diyagramında gösterilmektedir. Birinci kumanda kolu (soldaki) sağa sola hareket ettirildiğinde, taban servo motoru dönme hareketini gerçekleştirecektir. Birinci kumanda kolu ileri geri hareket ettirildiğinde ise iki numaralı servo

motor hareket edecektir. İkinci kumanda kolu, ileri geri hareket ettirildiğinde, 3 numaralı servo motor dönme hareketini gerçekleştirecektir. İkinci kumanda kolu sağa sola hareket ettirildiğinde 4 numaralı servo motor hareket verecektir. Aynı şekilde, üçüncü kumanda kolu, ileri geri hareket ettirildiğinde 5 numaralı, sağa sola hareket ettirildiğinde de 6 numaralı (tutucu, gripper) servo motor hareket verecektir.

Birinci kumanda kolunun butonuna tek tıklama yapıldığında, o anki servo motor konumları hafızaya alınır (öğrenme modu). Bu her tekrarlandığında değişmiş her konum sıralı hafızaya alınır. Birinci kumanda kolunun butonuna çift tık yapıldığında, robot hafızadaki hareketleri sürekli olarak tekrarlayacaktır (oynatma modu). Aynı kumanda kolunun butonuna tekrar tek tık yapıldığında da robot yavaşlayıp duracaktır. Herhangi bir yeni konum bilgisi hafızaya girildiği anda önceki konum bilgilerinin hepsi silinmektedir. Eğer yeni konum bilgisi girilmez ve tekrardan birinci kumanda kolu butonuna çift tık yapılırsa robot hafızasındaki konum bilgilerine uygun olarak hareketleri sürekli tekrarlayacaktır.

Robotun hız kontrolü, ikinci kumanda kolu butonuna tek tıklama yapılması suretiyle gerçekleştirilmektedir. Her tıklamada eklem hızları değişecektir. Yavaş, normal ve hızlı olmak üzere üç hız modu bulunmaktadır. Robot öğrenme veya oynatma modlarında iken ikinci kumanda kolunun butonuna tek tıklama yapıldığında, bir hız modundan diğerine geçilebilmektedir.



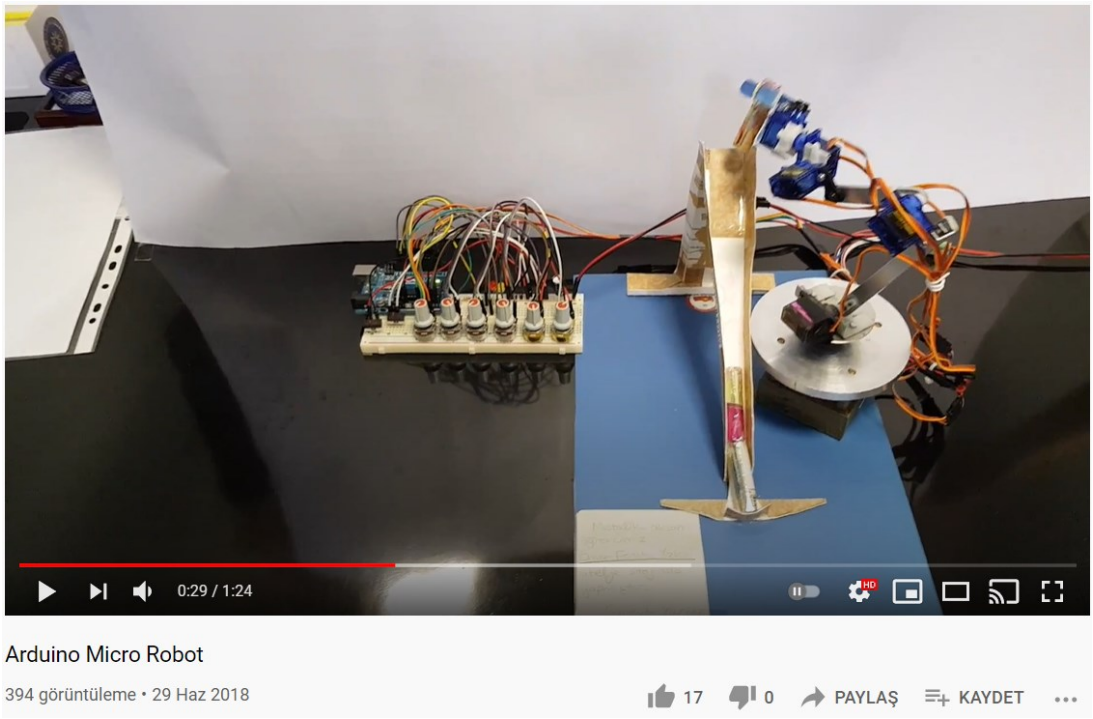
Resim 6.21. Robotun çalışma algoritmasını gösteren akış diyagramı.

2. TASARLA VE ÜRET

2.1. Tasarla: Beş Serbestlik Dereceli Robot için Çalışma Ortamı

Eğitmen, öğrencilerden ürettikleri robotla belli bir periyodik işlevi yerine getirebileceği veya sensörlerle otomatik olarak belli bir görevin yerine getirilebileceği bir sistemi tasarlamalarını ister. Örneğin, küçük parçaları bir noktadan alıp bir kutuya bırakma işlemi yaptırılabilir. Başka bir örnek olarak, aşağıdaki videodakine benzer bir uygulama yaptırılabilir (Resim 6.22).

Video Linki: HTTPS://WWW.YOUTUBE.COM/WATCH?V=QPYND2UCZTY



Resim 6.22. İnternet ortamında yayınlanmış örnek bir uygulama

Videoda görülen kızak modeli için, etrafımızda kolaylıkla bulabileceğimiz paket karton malzemeleri yeterli olacaktır. Robotun kaldırdığı objeler kartondan rulo şeklinde yapılmış ve saydam bantla kaplanmış hafif parçalardır.

2.2. Üret: Beş Serbestlik Dereceli Robot için Çalışma Ortamı

Öğrencilerden tasarla aşamasında yaptıkları tasarıma göre basit bir çalışma ortamını (yukarıda verilen örnekteki karton kaykay ve küçük silindirik parçalar gibi) üretmeleri istenir. Bu çalışma ortamının ne ve nasıl olacağı, eğitmene ve öğrencilerin hayal gücüne bırakılmıştır. Yapılan üretimler eğitmenler tarafından kontrol edilerek eksikler giderilir.

3. DEĞERLENDİR

Öğrenciler ile bu hafta yapılan uygulamalar konusunda görüşmeler ve değerlendirmeler yapılmalı ve yeterli kazanımları elde edip etmedikleri belirlenmelidir. Aşağıdaki örnek sorular üzerinden tartışmalar gerçekleştirilebilir:

- Beş serbestlik dereceli bir robotun özellikleri nelerdir?
- Tasarım süreci ile ilgili bir problem yaşadınız mı? Yaşadıysanız bunu nasıl çözdünüz?
- Devrelerin fiziki kurulumunda zorlandığınız yerler oldu mu?
- Bu derste bahsi geçen beş serbestlik dereceli robotların pratik hayattaki uygulamaları (veya endüstriyel uygulamaları) neler olabilir? Örnek uygulamalar:
    - İmalat endüstrisinde beş serbestlik dereceli robotlara rastlanmaktadır. Bu tür robotların çok çeşitli amaçlı kullanımlarına rastlanmaktadır. Örneğin, bir konveyörden gelen paketlerin alınıp bir araca yüklenmesi, birkaç parçanın bir araya getirilip montajının yapılması, bir parçanın başka bir parçaya kaynak yapılması, boya yapılması gibi:
        - https://www.youtube.com/watch?v=46Hs8LK645U
        - https://www.youtube.com/watch?v=9GLy2Kywp70
        - https://www.youtube.com/watch?v=uYRWHSIJ3Xc
    - Robot kolların eğlence endüstrisinde bile kullanımları vardır. Örneğin, uç elemanı olarak bir veya iki koltuğa sahip bir robot, bu koltukta oturan kişiye lunapark eğlencesi sunabilir:
        - https://www.youtube.com/watch?v=bSdA_oq1EgU
        - https://www.youtube.com/watch?v=2_SEJHxlmWw
        - https://www.facebook.com/UNILADTech/videos/robotic-arm-ride/718927789661986/
        - https://www.youtube.com/watch?v=IwE6VIPTcFg

İLAVE ETKİNLİK

Bu kısımda, mevcut zamana bağlı olarak, istenildiğinde, öğrencilerden üç serbestlik dereceli farklı ve yeni bir sistem uygulamasının tasarımı, üretimi ve uygulaması yaptırtılabilir.

Örnek uygulama: Gerçekleştirilen 5 serbestlik dereceli robotun, belli bir noktada belli bir nesneyi tespit ettiği anda o nesneyi oradan alıp bir toplama kutusuna bırakmasını sağlayacak bir sistem oluşturulabilir. Burada dikkat edilmesi gereken husus, nesnenin belli bir noktada mevcut olup olmadığını robotun tespit etmesine olanak sağlayacak bir sensör kullanımı gerekmektedir.

Burada başka ne tür uygulamalar yaptırılacağı hususu, eğitimciye ve öğrencilerin hayal gücüne bırakılmıştır.

7. Bölüm: Yarışma Hakkında Bilgiler ve Raporlama

1. KAPSAM VE AMAÇ

Bu bölümde İleri Robotik dersi kapsamında 12. Haftada yapılacak olan yarışmanın teknik detayları bulunmaktadır. Yarışmanın genel amaç ve hedefleri, öğrenciler yönünden kazanımları ve eğitmenlerin dikkat etmesi gereken hususlar açıklanmıştır. **Yarışmacıların bu bölümü dikkatle okumaları gerekmektedir**. Bu kısımdaki yarışma modeli örnek olarak verilmiş olup, farklı uygulamalar da eklenerek değiştirilebilir.

Yarışmada amaç, ikişer kişilik takımların, tasarlayacakları kablolu kumandalı (sürücülü), 2 kontrollü ve bir avare tekerlekli, üzerinde robot kolu bulunan bir araç ile yarışma alanına bırakılmış 10 adet topu toplamaları ve top biriktirme alanına bırakmalarıdır. **En kısa sürede topları toplayan yarışmayı kazanacaktır**.

Bu hafta eğitmen öğrencilere yarışma kapsamında gerçekleştirilecek robotun tasarım kısıtları hakkında bilgi verir ve öğrencilerin tasarımlarını planlamalarını ve gerçekleştirmelerini sağlar. Proje grupları oluşturulur ve öğrencilerin tasarımları ile ilgili tartışmaları yapmaları sağlanır. Proje aşamaları ve tasarımlar planlanırken proje tasarım raporunun da doldurulması sağlanır. Proje tasarım raporlarının ders sonunda toplanacağı belirtilir. Toplanan tasarım raporları aşağıdaki kriterlere göre puanlandırılır;

- Rapor 100 puan üzerinden değerlendirilecektir.

- 25 puan yazım dili, rapor organizasyonu ve görselleştirmeler üzerinden verilecektir.

- Proje Tasarım Raporundaki 1, 2, 3, 4 ve 5 nolu sorular 15'er puan üzerinden değerlendirilecektir.

- Proje Tasarım Raporundaki 6. soru için puanlama yapılmayacak olup, bu soru üzerinden beyin fırtınası yapılması sağlanmalıdır.

- **Rapor toplam puanı, yarışma genel puanının %20'sini oluşturacaktır. Öğrencilere bunun bilgisi verilmelidir**.

2. TANIMLAR VE GENEL BİLGİLER

2.1. Tanımlar

Takım: İki kişiden oluşan yarışma grubu

Sınıf: 20 kişilik eş zamanlı eğitim grubu

Araç: Biri avare, ikisi kontrollü tekerlekli, üzeri robotlu, hareketli sistem

Platform: Yarışmanın gerçekleşeceği düz, alçıpan yüzey

Kumanda: Uzaktan kablolu olarak aracı yönlendiren, robotu yönlendiren, cihaz

Sürücü: Araç/robot kullanıcısı

Orta hat çizgisi: Aracın yarışmaya başlayacağı çizgi

Top biriktirme alanı: Toplanan topların biriktirileceği alan

Yarışma alanı: Yarışmanın gerçekleşeceği yüzey

2.2. Genel Bilgiler

1. Yarışma Resim 7.1'de verilen alanda gerçekleşir.
2. Alan 120 cm x 250 cm boyutlarında, kenarlarında 10 cm alçıpan duvar olan bir bölgedir.
3. Top biriktirme alanı 40 cm x 40 cm boyutunda, üstü açık, yanları tamamen kapalı bölgedir.
4. Temel problemler
    a. Hız
    b. Hassasiyet
    c. Planlama
    d. Motivasyon
    e. Konumlama kontrolü
    f. Aracın seri cevap vermesi amacıyla programlama becerisi

3. KURALLAR VE KISITLAMALAR

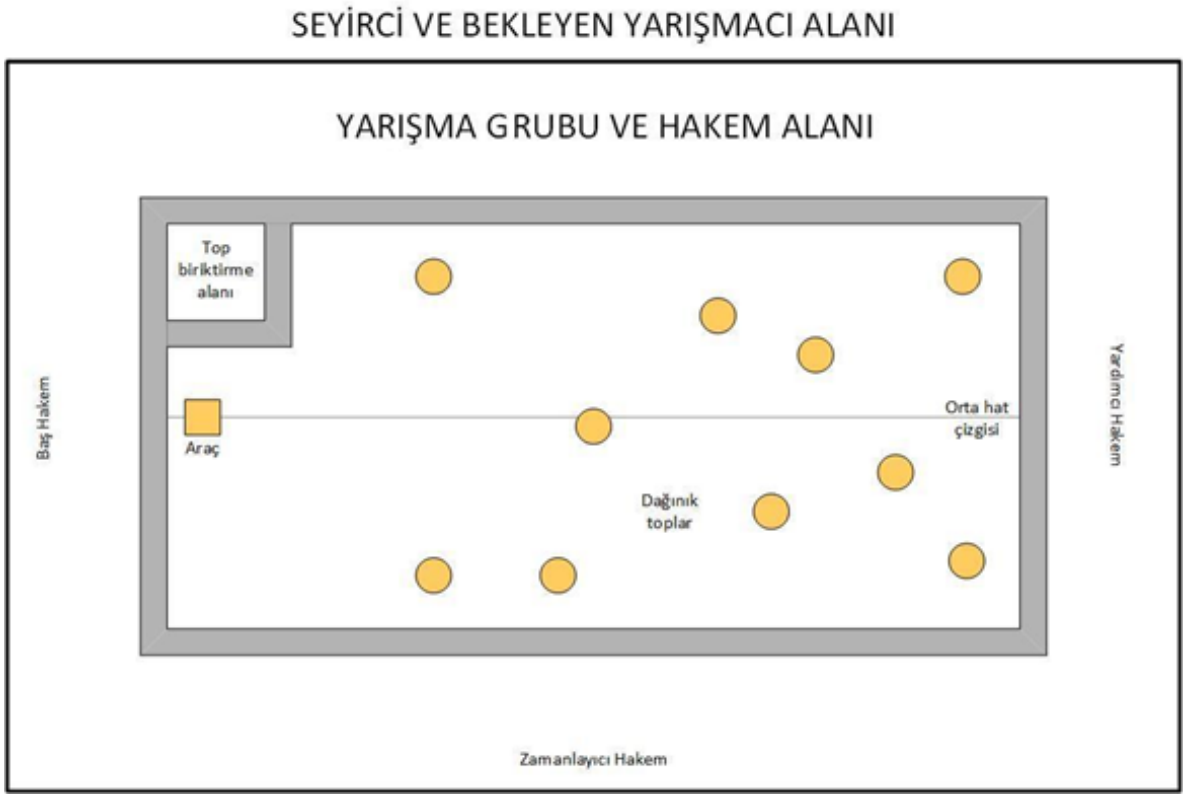
3.1. Yarışma Kuralları

1. Yarışmacıların, seyircilerin yarışma alanına müdahale etmeleri ve girmeleri yasaktır.
2. Her bir araç yalnızca bir adet kumanda ile idare edilmelidir.
3. Araçlar kablolu olarak kumanda edileceklerdir.
4. 10 adet top, her yarışmacı yarışma alanına aracını yerleştirdikten sonra, aracın önünden, yüzeyden 10 cm yukarıdan bırakılır. Dışarıya çıkan top olursa ayrı olarak tekrar aynı yer ve yükseklikten bırakılır.

5. Topları almak için en az iki serbestlik dereceli bir toplayıcı robot kol aracın üstüne yerleştirilmelidir.
6. Araç, toplar durmadan yarışmaya başlamaz.
7. Toplar teker teker toplanmalı ve top biriktirme alanına bırakılmalıdır.
8. Araç, yarışmaya kontrollü tekerleri yakın duvara 20 cm mesafeden, âvâre tekerleği orta çizgi üzerinde, yarışma alanı bitimine bakar şekilde başlar.
9. Top alma sırası ve planlaması yarışan ekibe bırakılmıştır.
10. Her grubun aracı ve robotu bir yarışmacı tarafından kablolu olarak kontrol edilecektir.
11. En kısa sürede topları toplayan kazanır.

3.2 Yarışma Ortamı Kısıtlamaları

1. Seyirciler, yarışma yüzeyinden en az iki metre mesafede yarışmayı seyrederler
2. Yarışan ekip ve hakemler, yarışma yüzeyi ve seyirciler arasındaki bölgede bulunurlar.
3. Yarışma Resim 7.1'de verilen alanda gerçekleşir.



Resim 7.1. Yarışma alanının krokisi

3.3. Tasarım Kısıtlamaları

1. Bir robot bir adet Deneyap Geliştirme Kartı ile kontrol edilmelidir.
2. Bir robot üzerinde robotun hareketi için en fazla 2 motor bulunabilir.
3. Bir robot üzerinde en fazla 2 tekerlek ve 1 avare tekerlek bulunabilir.
4. Robotlar kablolu bir kumanda ile kontrol edilebilir.
5. Kumanda, bir breadboard üzerine yerleştirilmiş kumanda kollarından oluşmalıdır.
6. Bir robot sadece bir kişi tarafından, tek bir kumanda ile kontrol edilebilir.
7. Top alma mekanizmasında (robot kol) en fazla 5 adet servo motor kullanılabilir.
8. Bir robot için en fazla 500 gram PLA filament kullanılabilir.
9. Robot mekanizmasında en fazla 30 adet abeslang kullanılabilir.

Öğrencilerin bu kısıtlara göre geliştirecekleri top toplayıcı mobil robotu planlayıp tasarlamaları sağlanır. Bu aşamada verilen Proje Tasarım Raporu'nu doldurmaları istenir. Raporlar ders sonunda toplanarak notlandırılır. Yarışmada eşit durumda olan yarışmacıların tasarım raporuna verilen nota göre sıralanacağı belirtilir.

PROJE TASARIM RAPORU

GRUP ADI ve LOGOSU:

PROJE EKİBİ:

Adı Soyadı	**Görevi**

1. Projenin amacı nedir ve bu amaca ulaşmak için gerçekleştirmeyi planladığınız hedefler nelerdir?

2. Proje geliştirme aşamasında karşılaşabileceğiniz problemler hakkında öngörüleriniz nelerdir? Bu problemleri nasıl çözmeyi düşünüyorsunuz?

3. Proje kapsamında gerçekleştirmeyi planladığınız robotun mekanik tasarımı hakkında bilgi veriniz. Katı modelleme programı ile temel tasarımınızı yaparak bu bölüme ekleyiniz. Robotun topları alması için geliştireceğiniz mekanizma hakkında ayrıntılı bilgi veriniz.

4. Proje kapsamında gerçekleştirmeyi planladığınız robotta kullanmayı planladığınız sensörler hakkında bilgi veriniz. Hangi sensörü hangi amaçla kullanacağınızı açıklayınız.

5. Yarışma görevini yerine getirmek için kurguladığınız yazılımın algoritmasını çizerek açıklayınız.

6. Tasarım kısıtlamaları olmasaydı robotunuza hangi eklemeleri yapardınız?

8. Bölüm: Mekanik/Elektronik Tasarım ve Montaj

Sistem entegrasyonu, tasarlanmış ve üretilmiş olan mekanik yapının elektronik donanımlarla birleştirilerek yazılım yapılmaya uygun hale getirilmesinden oluşmaktadır. Sistemin hali hazırda kullanılabilmesi için montajı yapılmış mekanik parçalar elektronik devreler vasıtası ile sürülür. Bu sürülme işleminin içinde yazılım elektroniği kontrol ederek çeşitli komutlarla motor/aktüatörlerin çalıştırılmasını sağlamaktadır. Ancak sistem entegrasyonu yapılırken montaj/elektronik/yazılım üçlüsünün birlikte çalıştırılması zorunluluğu doğmaktadır.

Bu açıdan değerlendirildiğinde bu üç ayrı konunun aynı sistem üzerinde çalıştırılması için bazı hususlara dikkat edilmesi gerekmektedir. Bu raporun içerisinde hangi hususlara dikkat edileceği detaylandırılarak anlatılmaktadır.

1. MEKANİK MONTAJDA DİKKAT EDİLMESİ GEREKENLER

Mekanik montaj, robot veya araç parçalarının bir araya getirilerek kullanıma hazır halde birleştirilmesinden oluşmaktadır. Aşağıda verilen hususlar mekanik parçalarla birlikte kullanımda dikkat edilmesi gereken maddelerdir:

- Mekanik montaj parçaları elektronik ve motor parçaları ile birleştirilmeden önce montaj testine sokulmalıdır.
- Montaj testinde bütün mekanik parçaların birleştirilerek hareket edip edilmediğine dikkat edilmelidir.
- Mekanik parçalar birleştirildikten sonra elektronik komponentlerden önce motorların sisteme bağlanması yapılmalıdır.
- Motorlar sisteme bağlandıktan sonra mekanik parçalarla birlikte hareket edip etmediğine dikkat edilmelidir.
- Eğer sistem bu kademeden sonra istenen hareketleri yapabiliyorsa elektronik devre kartı sisteme eklenebilir.
- Elektronik devre kartı sisteme eklendiğinde mekanik hareketlerin tamamının yapılıp yapılamadığı kontrol edilmelidir.
- Eğer hareket tamamen yapılabiliyorsa sistemin yazılımı ile testlerine geçiş yapılabilir.

2. ELEKTRONİK MONTAJDA DİKKAT EDİLMESİ GEREKENLER

Elektronik montaj aşaması, elektronik bileşenlerin ve kabloların motorlarla ve diğer elektronik elemanlarla birleştirilmesinden oluşmaktadır. Aşağıda elektronik montaj aşamasında dikkat edilmesi gereken hususlar verilmiştir. Eğitmen tüm yarışma hazırlıkları boyunca öğrencilerle birebir ilgilenmeli ve onlara yol göstermeli, öğrencileri ve takımları ayırmadan konu hakkında bütün çabasını göstermelidir.

- Elektronik bileşenleri sırayla sisteme bağlayınız. Örneğin motorları ayrı sensörleri ayrı bir sırada bağlama işlemine alınız.
- Motorların kablolarının doğru takıldığından emin olunuz. Motor adedi kadar kablonun elektronik devreye bağlı olduğunu kontrol ediniz.
- Sensör bağlantılarında devre kartına olan kablolamayı kontrol edin. Kontrol edilmiş her kabloyu bir yere not edin ve işlemlerinize devam edin.
- Potansiyometre, direnç ve diğer elektronik parçaların bağlantılarını motor ve sensör bağlantılarından sonra yapınız.
- Tüm elektronik donanımları bağladıktan sonra motorları hareket ettirerek deneyiniz. Mekanik montaja bağlama işlemi yapılmadan bunun yapılması önemlidir.
- Eğer motorlar elektronik komponentler ile sürülebiliyorsa sensörlerin testine geçiniz.
- Sensörlerden alınan değerlerin doğru olup olmadığını kontrol ediniz.
- Sensörler doğru veri veriyorsa bu sefer motor/sensör ikilisini beraber çalıştırınız.
- Eğer motor ve sensör ikilisi doğru çalışıyorsa bu kademeden sonra mekanik montajla birleştirme işlemine geçebilirsiniz.

9. Bölüm: Sistem Entegrasyonu ve Yazılım

Bu hafta eğitmen, öğrencilerden bir önceki haftada üretilmiş olan yarışmaya yönelik robotun mekanik – elektronik bileşen uyumluluğunu ve sistemin çalışma performansını test ederek yazılımını oluşturmalarını ister.

1. YAZILIM AŞAMASINDA DİKKAT EDİLMESİ GEREKENLER

Yazılım aşaması, sistemin motor, sensör ve diğer bileşenlerinin belirli bir iş için programlanmasından oluşmaktadır. Aşağıda belirtilen hususlar, kodların yazılması aşamasında dikkat edilmesi gerekenlerdir.

- Öncelikle kodların yüklenebilmesi için Deneyap Geliştirme Kartı'nın bilgisayara bağlı olduğundan emin olunuz.
- USB kablo üzerinden Deneyap Geliştirme Kartı ile bilgisayar bağlantısı yapıldıktan sonra bir test kodu çalıştırınız. Test kodunu karta yükleyebildiğinizden emin olunuz.
- Tüm program bir main() fonksiyonu içinde olmamalıdır. Yapacağınız işlemler için ayrı ayrı fonksiyon tanımları yapınız.
- Kullanacağınız genel değişkenler ve sabitler için bir kütüphane oluşturunuz ve programın her yerinde (global) bu değişkenleri kullanmaya çalışınız.
- Yapacağınız projeye başlamadan önce, motorları belli açılarda sürmeyi deneyiniz. Özellikle bir tam tur ileri veya bir tam tur geri adım attırmaya çalışınız.
- Motorların fonksiyonları yazıldıktan sonra sensör denemelerine geçebilirsiniz. Sensörlerden okuma yapma için ADC modülü ya da dijital girişleri kullanınız.
- Sensör doğrulaması için elinizle sensörlerin konumları veya giriş parametrelerinin değiştirerek uygulamada bunları okuyabildiğinizi görünüz.
- Seri haberleşme portu ile karttaki bütün verileri ve değişkenleri görselleştirebilirsiniz.
- Eğer sensörler doğru bir şekilde çalıştırılabiliyorsa motor/sensör ikilisini aynı anda çalıştırmayı deneyiniz. Bir tam tur ileri/geri işlemini yaparak sensörlerden doğru değerler aldığınızdan emin olunuz.
- IMU sensörü için SPI haberleşmesinin kullanıldığını unutmayınız.
- Eğer sensör/motor çifti doğru bir şekilde çalıştırılabildiyse buna dair bir fonksiyon tanımı yapınız ve bu kodu çalıştırınız.
- Bu kademelerden sonra eğer gerekliyse dijital giriş/çıkış pinlerinden okuma yazma işlemleri yapmaya çalışınız.

- Eğer kodların hepsi doğru bir şekilde çalıştırılabildiyse artık projenizde kullanmanız gereken algoritmayı tasarlamaya geçebilirsiniz.
- Algoritma tasarımı için gereken fonksiyonların hepsi, yukarıda verilen maddeleri gerçekleştirdiyseniz hazır olmak zorundadır.
- Eğer algoritmanız bittiyse elektronik komponentleri mekanik montaja entegre edip algoritmanızı bir daha deneyiniz.
- Eğer bir işlemde problem çıkıyorsa öncelikle yazılım tarafında hataları arayınız. Eğer yazılım kısmında problem olmadığını düşünüyorsanız elektronik bileşenleri kontrol ediniz.

  Yazılımla ilgili belirtilen hususlarda bütün sistemlerin entegrasyonu ile alakalı detaylı bilgi elde etmeniz mümkündür.

  Aşağıda temel bilgisayar programlama ile alakalı tavsiyeler verilmektedir:

- Programlama dilinde değişken ve fonksiyon işlemlerini anlamlı olacak şekilde seçiniz. Örneğin (motorSur(), motorDur(), int sensor1Veri, …)
- Uygulamanızda işlemler gecikebileceği için delay() komutunu kullanmayı unutmayınız.
- Bütün değişkenlerin hepsini main() fonksiyonu dışında tanımlayınız.
- Fonksiyonların hepsini main() fonksiyonunun üstünde tanımlayınız.
- Her yazdığınız kodun altına bir boşluk bırakınız.
- Her yazdığınız kodun üstüne yorum satırı olarak ne iş yaptığını ve nerede kullanıldığını yazınız.
- Eğer döngü kullanıyorsanız içsel değişkenin arttırıldığından veya azaltıldığından emin olun.
- Eğer koşul ifadeleri kullanıyorsanız bu koşul ifadelerinin doğru değişkenlerden oluştuğuna dikkat ediniz.

İLERİ ROBOTİK

LİSE

DENEYAP TÜRKİYE