

Deneyap Teknoloji Atölyeleri

İleri Robotik – Lise

Yazarlar

Doç. Dr. Cüneyt Yılmaz

Dr. Öğr. Üyesi Mehmet Emin Aktan

Prof. Dr. Vasfi Emre Ömürlü

Dr. Öğr. Üyesi Mehmet İşcan

İçindekiler

İçindekiler	2
Öğretim Kılavuzu	4
İLERİ ROBOTİK DERSİ ÖĞRETİM KILAVUZU	4
İLERİ ROBOTİK SİSTEMLERE GİRİŞ	7
İLERİ ROBOTİK DERS PLANI	8
DERSİN İŞLEYİŞİ İLE İLGİLİ ÖNEMLİ HUSUSLAR	8
KAYNAKÇA	8
1. BÖLÜM: Robot Kavramı, İleri Robot Teknolojilerine Giriş ve IMU Sensörü	9
1. GÖZLE VE UYGULA	10
1.1. Gözle: Robot Nedir?	10
1.2. Gözle: Ders Kapsamında Gerçekleştirilecek Robotlar	11
1.3. Gözle: Mekanizma ve Robot Kavramları	14
1.4. Gözle: IMU (Inertial Measurement Unit) Sensörü Nedir?	18
1.5. Uygula: IMU Sensörü Değerlerinin Ekrana Yazdırılması	18
1.6. Uygula: IMU Sensör Değerlerinin Pan-Tilt Açısı Değerlerine Dönüştürülmesi	19
1.7. Uygula: IMU Sensörü ile DC Motor Kontrolü	20
2. TASARLA VE ÜRET	23
2.1. Tasarla: Denge çubuğu mekanizması	23
2.2. Üret: Denge çubuğu mekanizması	23
3. DEĞERLENDİR	23
4. İLAVE ETKİNLİK	24
Kaynakça	24
2. BÖLÜM: İki Serbestlik Dereceli Robot Mekanizması	25
1. GÖZLE VE UYGULA	26
1.1. Gözle: Kumanda Kolu Nedir?	26
1.2. Uygula: Deneyap Kart / Deneyap Kart 1A Üzerinde Kumanda Kolu Çalıştırılması	27
1.3. Uygula: Deneyap Kart / Deneyap Kart 1A ile Çift Servo Motor Kontrolü	29
1.4. Uygula: İki Motorlu Bir Sistemin Kumanda Kolu ile Kontrolü	32
1.5. Uygula: IMU Sensörü ile Çift Servo Motor Konum Kontrolü	35

1.6. Uygula: Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü	36
2. TASARLA VE ÜRET	42
2.1. Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü	42
3. DEĞERLENDİR	47
4. İLAVE ETKİNLİK.....	48
3. Bölüm: Üç Serbestlik Dereceli Robot Mekanizması.....	49
1. GÖZLE VE UYGULA.....	50
1.1. Gözle: Üç Serbestlik Dereceli Robotların Yapısı ve İşlevleri	50
1.2. Uygula: Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü	50
2. TASARLA VE ÜRET	64
2.1. Tasarla: Üç Serbestlik Dereceli Robot için Çalışma Ortamı	64
2.2. Üret: Üç Serbestlik Dereceli Robot için Çalışma Ortamı	64
3. DEĞERLENDİR.....	65
4. İLAVE ETKİNLİK	65
4. BÖLÜM: Beş Serbestlik Dereceli Robot Mekanizması	66
1. GÖZLE VE UYGULA.....	67
1.1. Gözle: Beş Serbestlik Dereceli Robotların Yapısı ve İşlevleri	67
1.2. Uygula: Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü	67
2. TASARLA VE ÜRET	84
2.1. Tasarla: Beş Serbestlik Dereceli Robot için Çalışma Ortamı.....	84
2.2. Üret: Beş Serbestlik Dereceli Robot için Çalışma Ortamı.....	84
3. DEĞERLENDİR.....	85
4. İLAVE ETKİNLİK	85

Öğretim Kılavuzu

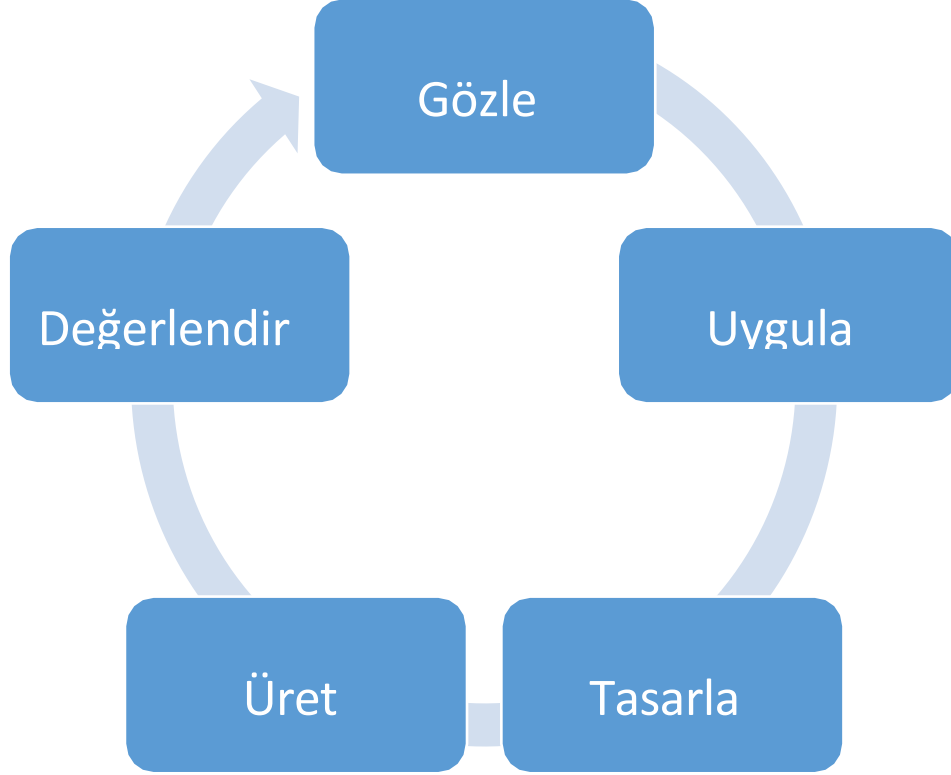
İLERİ ROBOTİK DERSİ ÖĞRETİM KILAVUZU

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** olarak özgün bakış açıları ile **yeni değerler oluşturmak** ve **sistematiik 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 kullanılabilir. 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.



Şekil 1 Öğrenme Döngüsü

Gözle: Gözle adımıında eğitimden aktif rol almaktadır. Bu adım iki aşamadan oluşmaktadır. Birinci aşamada, eğitimden öğrencilerin o hafta işlenecek konu hakkında bilgi edinmelerini sağlayarak, konuya olan ilgilerini ve motivasyonlarını güçlendirmelidir. Bu noktada eğitimden 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 eğitimden, örnek bir robotik uygulamayı yine ilgi çekici görsel ve videolarla açıklar. Eğitimden öğrencilerle diyalog içerisinde olması ve soru-cevaplar ile öğrencilerin konunun içerisine çekilmesi beklenmektedir. Ayrıca eğitimden, öğrencilerin yorum yapmasını ve sorular sormasını teşvik etmelidir.

Uygula: Bu bölümde eğitimden, 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 uygulama adımıyla 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 üretilmektedir. 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.

İLERİ ROBOTİK DERS PLANI

LİSE İÇERİĞİ	
Robot Kavramı ve İleri Robot Teknolojilerine Giriş	1. hafta
İki Serbestlik Dereceli Robot Mekanizması	2. hafta
Üç Serbestlik Dereceli Robot Mekanizması	3. hafta
Üç Serbestlik Dereceli Robot Mekanizması	4. hafta
Beş Serbestlik Dereceli Robot Mekanizması	5. hafta
Proje Şenliği Hakkında Bilgiler ve Raporlama	6. hafta
Proje Şenliği	7. hafta

DERSİN İŞLEYİŞİ İLE İLGİLİ ÖNEMLİ HUSUSLAR

- Öğitmenlerin, derslerin gözle aşamasında verilen örnekleri çoğaltması, görsel ve videolar ile bu aşamanın zenginleştirilmesi gerekmektedir. Sadece verilen içerikle yetinilmemelidir.
- Her bir derste yapılacak uygulamalarda kullanılacak malzemeler derslerin giriş kısmında belirtilmiştir. Bunlar 1 grup için gerekli malzemelerdir. Gruptaki öğrenci sayısı öğretmen 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. Bölüm içeriğinde yer alan Deneyap Kart bilgisayar bağlantısı ve gerekli programların ve kütüphanelerin yüklenmesi ile ilgili kısımlar daha önce Deneyap Kartı'nı kullanmamış öğrenciler içindir. Önceki eğitimlerde bu kısmı uygulayan gruplarda Deneyap Kart / Deneyap Kart 1A'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](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, İleri Robot Teknolojilerine Giriş ve IMU Sensörü

Ön Bilgi:

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

Haftanın Kazanımları:

- Öğrenciler robot kavramını tanımlar, robot çeşitleri ile ilgili günlük hayattan örnekler verir.
- Öğrenciler robotiğin temel kavramlarını tanır ve kavrar.
- Öğrenciler 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.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A'nın temel özelliklerini kavrar.
- Öğrenciler analog ve dijital sinyalleri kavrar. Bunlarla ilgili örnekler verir.
- Öğrenciler Deneyap Kart IDE kurulumunu yapar ve gerekli kütüphaneleri yükler.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinde IMU sensörü çalıştırır.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinde DC motor çalıştırır.

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. Deneyap Kart / Deneyap Kart 1A 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:

Malzeme Listesi	
Deneyap Kart / Deneyap Kart 1A	1 adet
L298N motor sürücü	1 adet
DC Motor (6V Sarı Motor)	1 adet
Tekerlek	1 adet
Breadboard	1 adet
Jumper Kablo	20+ Adet
Mikro USB Bağlantı Kablosu	1 adet
Harici 5V USB Batarya / 9V Pil / 2li 18650 Pil,	1 adet
Abeslang	Yeterli Kadar

Haftanın İşlenişi:

Gözele: Deneyap Kart / Deneyap Kart 1A'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: Robot Nedir?

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 iletme 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.

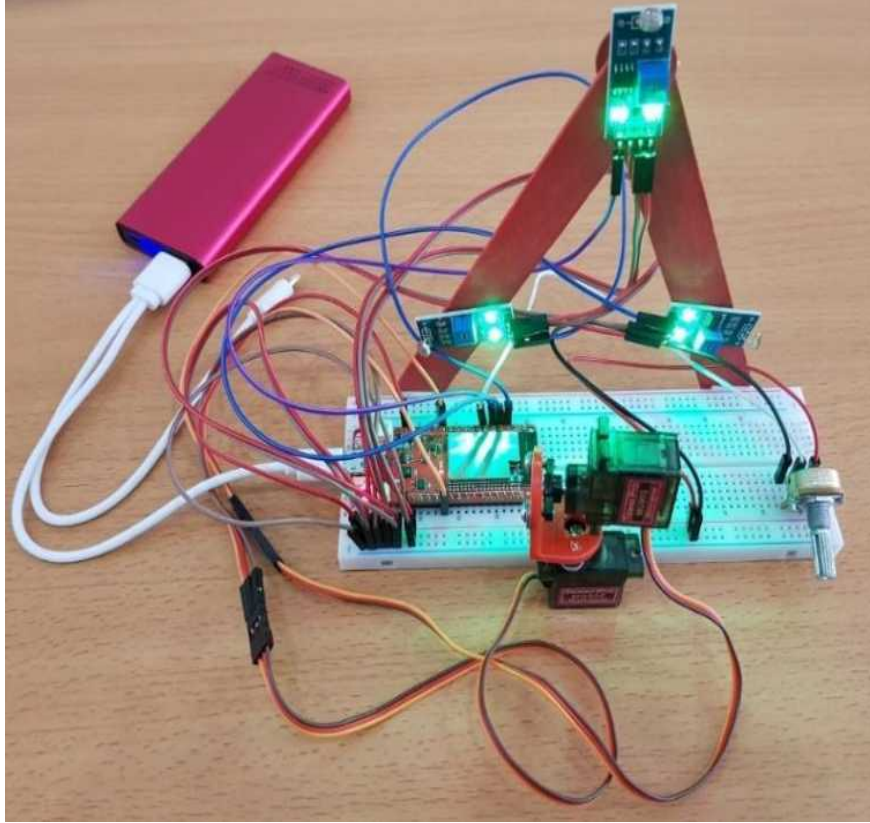
19. ve 20. yüzyıllarda elektronik ve bilgisayar teknolojilerinin hızla gelişmesi ile robotik sistemlerde de ciddi değişim ve gelişmeler yaşanmıştır. Robot tanımı da çeşitli otorite ve ülkeler tarafından farklı şekilde yapılmaktadır. Robotun yaygın bir şekilde kabul gören genel tanımı Amerikan Robot Enstitüsü 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 aksamaların 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?

1.2. Gözle: Ders Kapsamında Gerçekleştirilecek Robotlar

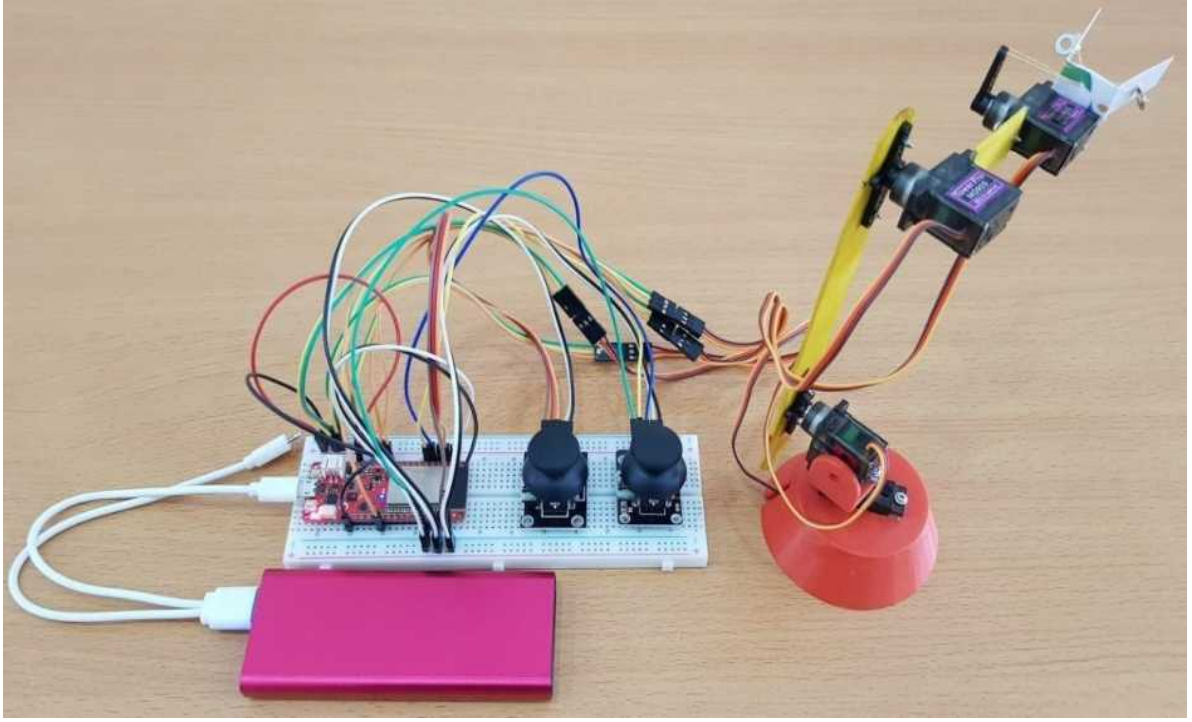
Ders kapsamında hem robotlar hem de robotların alt sistemleri üzerine çalışmalar yapılacaktır. Robotların alt sistemlerinin neler olduğu konusunda bir fikir alışverişi öğrenciler ve öğretmen arasında gerçekleştirilebilir. Paralel robotlara göre teknolojisi, tekniği, teorisi ve üretimi daha kolay olmasından 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’ de böyle bir robot iki eksenli olarak gösterilmektedir ve İleri Robotik dersinde üretilecektir.



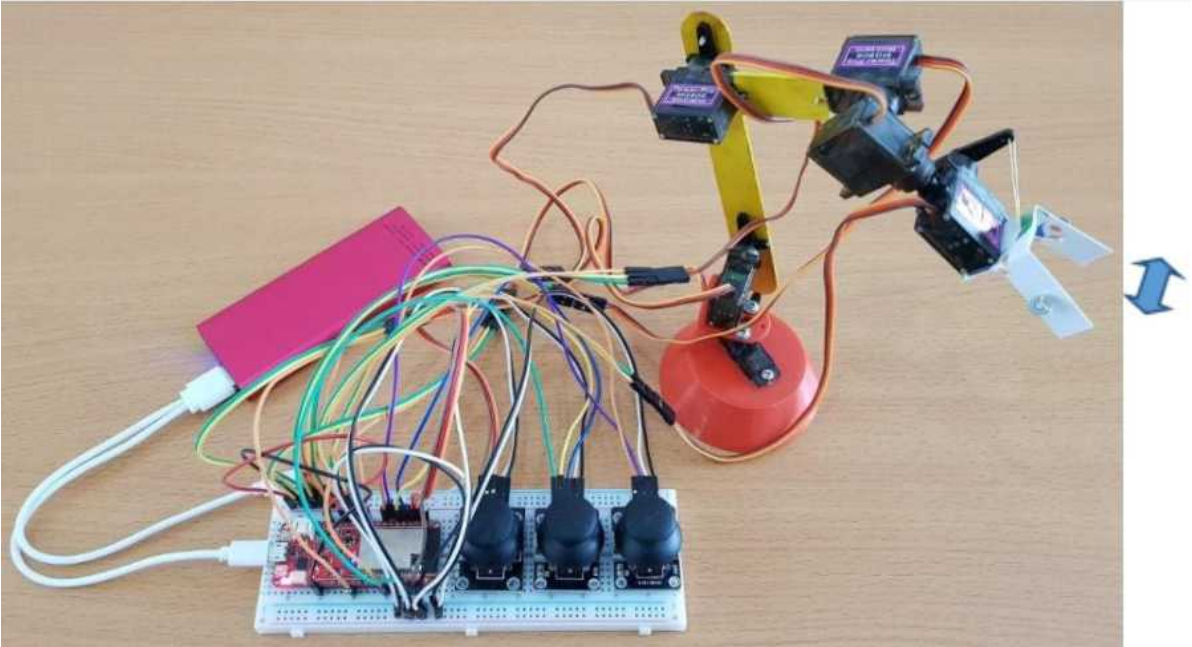
Resim 1 İki Eksenli Seri Robot Sistemi

Üç eksenli robotlarda ise iki eksenli robottan farklı olarak bir aktif eksen daha bulunmaktadır. Üç eksenli robotlar hem kendi dikey eksen 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 2’de üretilecek olan robotik sistem görülmektedir.



Resim 2 Üç Eksenli Seri Robot Sistemi

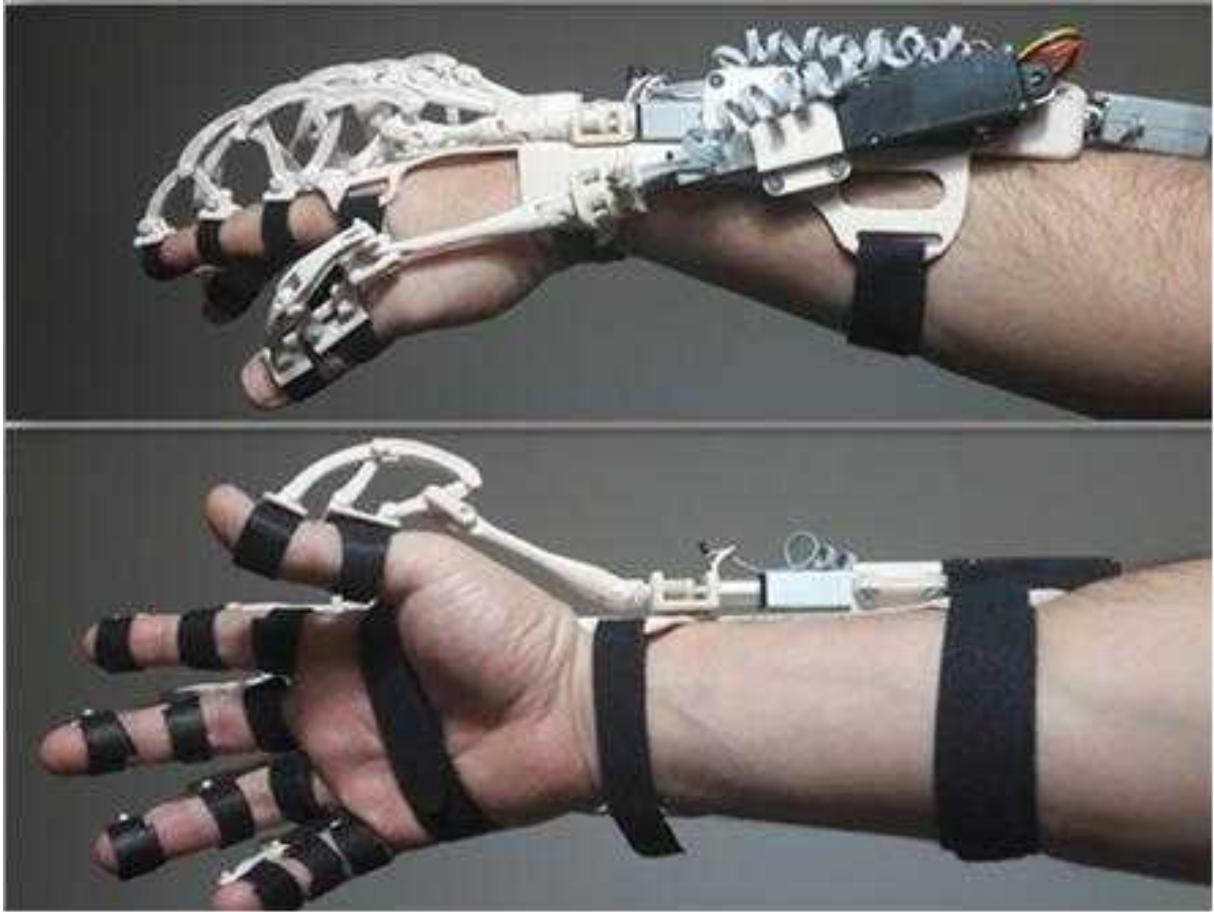
Ders kapsamında üretilecek olan son seri robot beş eksenli bir robottur. Üç eksenli robottan farklı olarak üzerinde bir de bilek motoru bulunmaktadır. Dikey eksen etrafında dönme, omuz, dirsek, bilek ve artı olarak tutma mekanizması olacaktır. Resim 3'te üretilecek olan robot sistemi görülmektedir.



Resim 3 Beş Eksenli Seri Robot Sistemi

1.3. Gözle: Mekanizma ve Robot Kavramları

Her robot, içerisinde çeşitli mekanizmalar içerir. Fakat her mekanizma robot değildir. Örneğin, Resim 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 4 Bir İnsan-Makine Arayüzü Olarak El Dış İskelet Sistemi (Ahmadipour, 2018)

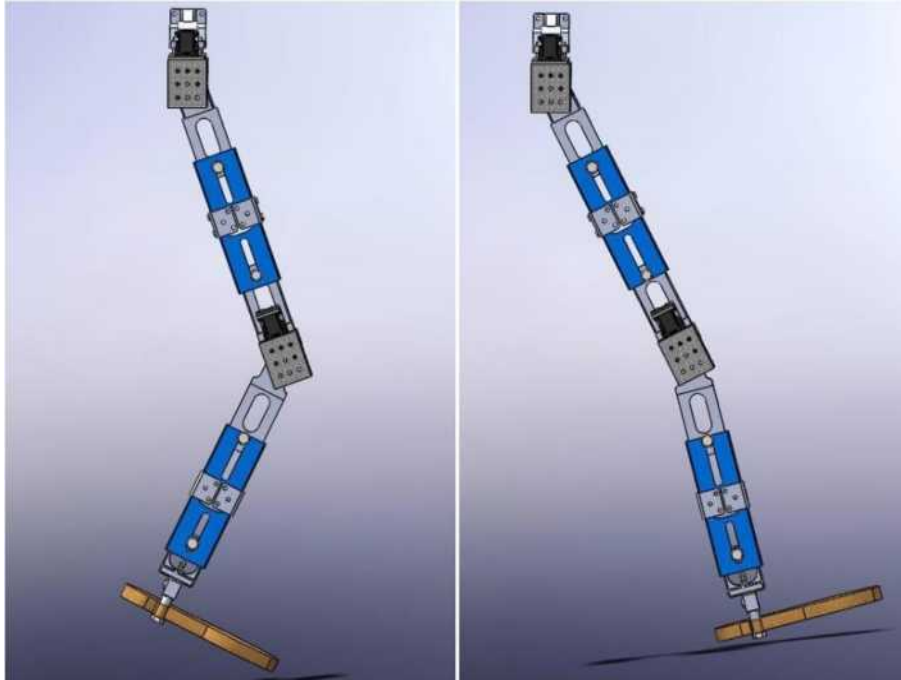
Vücudumuz da mükemmel bir robotik sistem olarak düşünülebilir. Parmaklar, omuz, kol, dirsek, bilek, diz, kalça, boyun gibi uzuvlar ve eklemler fiziksel olarak karmaşık, çok serbestlik dereceli bir robotik sistemdir. Bunların beyin tarafından kontrolü yoluyla vücudumuz çok karmaşık hareketleri yapabilmektedir.

1.3.1. 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 eklemlerinizi 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 düşünersek 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 5).



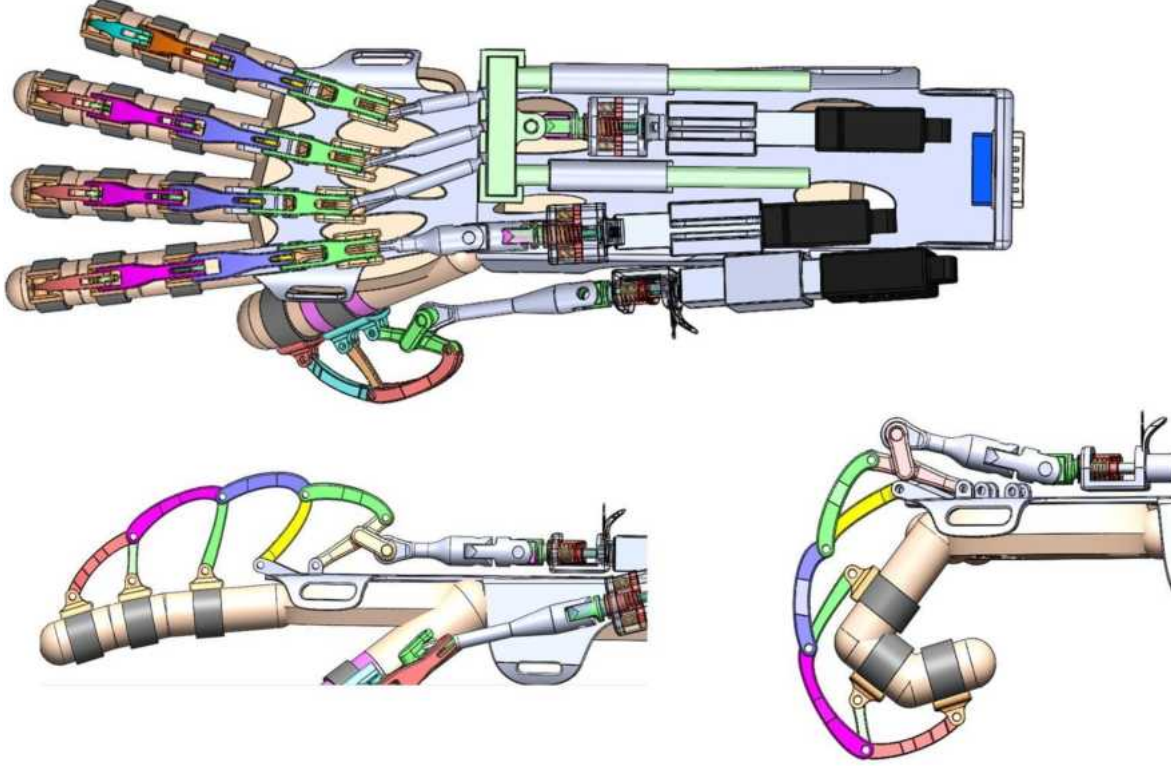
Resim 5 Seri Robot Örneği Olarak Bacak Dış İskelet Robotu Katı Modeli (Eken, 2018)

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 eksenini 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 6'da verilen uçuş simülatorünü inceleyelim. Simülator üzerinde bulunan kullanıcı simülatorün 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 6 Bir Paralel Robot Mekanizması (Ghorbani, 2018)

Bir diğ er robot tipi ise melez (hibrit) robot sistemleridir. Bu tip robotlar  zerinde hem paralel hem de seri eklem yapıları bulunmaktadır. Resim 7’de bir dıř iskelet el robotu verilmiřtir. Bu sistem incelendiğinde birbirine paralel durumda olan parmaklar g r lmektedir. Her bir parmak  zerinde de seri eklemler bulunmaktadır. El uzvu b t n olarak d ř n ld ğ nde  zerinde 2 ve 3 adet seri eklem bulunan beř tane birbirine paralel parmaktan oluřan melez bir robot sistemidir.



Resim 7 Melez (Seri + Paralel) Bir Robot Sistemi Katı Modeli (Ahmadipour, 2018)

1.3.2. Serbestlik Derecesi

Robot sistemleri ve mekanizmalardaki en  nemli kavramlardan birisi de serbestlik derecesidir. Serbestlik derecesi, bir robot veya mekanizmanın ne kadar sınırlı bir hareket kabiliyetine sahip olduğ n n ve aynı zamanda ne kadar bağımsız tahriğ e ihtiya  duyduğ n n bir g stergesidir.  rnek olarak bir sarkaç d ř n ld ğ nde bu sarkacın yaptığı řey nedir? Sallanmak. Bu sallanma hareketinin kontrols z olduğ  d ř n ld ğ nde bu bir serbestlik dereceli pasif bir harekettir. Bir serbestlik derecelidir   nk  tek bir eksen etrafında sadece sağ  ve sola salınmaktadır. Bu hareketi sadece bir a ı değışkeni ile ifade edebiliriz. Sallanma hareketi serbest olarak ger ekleřtiğinden (bir motor tarafından kontrol edilmediğinden) dolay  pasif hareket yapmaktadır. Bu sarkacın hareket eklemine bir motor bağılanarak sallanma a ısı kontrol edildiğ nde ise aktif hareket elde edilmiř olur. Sadece bir motor kullanıldığından bu sistem bir serbestlik dereceli olarak ifade edilmektedir.

Buna benzer olarak  nceki b l mlerde tanıtılan mekanizma ve robot sistemleri de d ř n ld ğ nde, bunların da hareket kabiliyetleri sınırlıdır.  rneğ n Resim 1’deki sistemi d ř n rsek, iki motor, iki a ı hareketi yapmaktadır. Yani iki serbestlik derecelidir. Elin sadece iřaret parmağının ucu d ř n l rse, elin sabit tutulduğ  bir durumda sadece iřaret parmağının ucunun hareket ettirilebilmesi i in parmak  zerindeki    boğ mun 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 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.

1.4. 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 Kart / Deneyap Kart 1A üzerinde dahili olarak 3 eksenli bir dönüş ölçer ve 3 eksenli bir açısal ivme ölçer olmak üzere toplam 6 eksenli bir IMU sensörü bulunmaktadır.

1.5. Uygula: IMU Sensörü Değerlerinin Ekrana Yazdırılması

Bu uygulamada Deneyap Kart / Deneyap Kart 1A üzerindeki IMU sensöründen verilerin nasıl alındığı ve kullanıma hazır hale getirildiği gösterilmektedir.

Uygulama Adımları

Adım 1: Deneyap Kart / Deneyap Kart 1A bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.



```
1 #include "lsm6dsh.h"
2 LSM6DSH IMU;
3
4 void setup() {
5   Serial.begin(115200);
6   IMU.begin();
7 }
8
9 void loop() {
10  Serial.print("accel_x: ");
11  Serial.println(IMU.readFloatAccelX());
12  Serial.print("accel_y: ");
13  Serial.println(IMU.readFloatAccelY());
14  Serial.print("accel_z: ");
15  Serial.println(IMU.readFloatAccelZ());
16  delay(500);
17 }
18
19
20
21
22
23
24
25
```

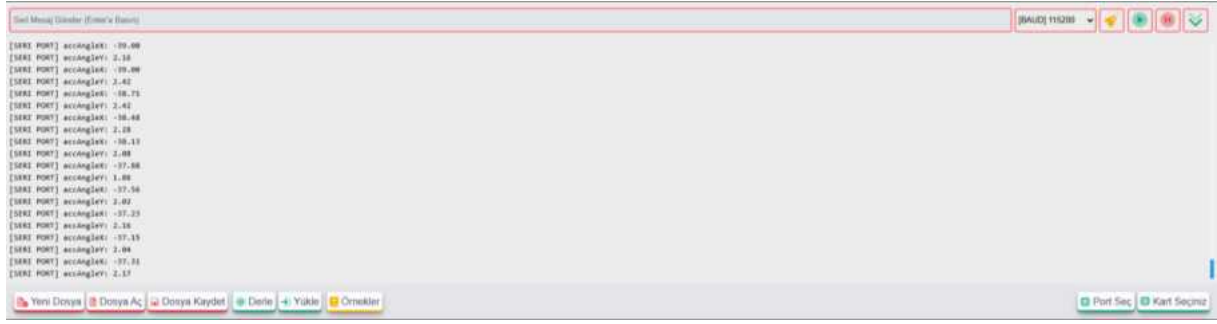
Resim 8 IMU Sensör Değerlerinin Ekrana Yazdırılması Kodu

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

```
1 #include "lsm6dsm.h"
2 LSM6DSM IMU;
3
4 void setup() {
5     Serial.begin(115200);
6     IMU.begin();
7 }
8
9 void loop() {
10     float accAngleX;
11     float accAngleY;
12
13     accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2)
14         + pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
15
16     accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2)
17         + pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
18
19     Serial.print("accAngleX: ");
20     Serial.println(accAngleX);
21     Serial.print("accAngleY: ");
22     Serial.println(accAngleY);
23 }
24
25
```

Bilgi Notu:

Resim 11’de gösterildiği gibi, Arduino PC uygulamasının Serial Monitor’ünde bu pan-tilt açısı değerleri gözlemlenmektedir.



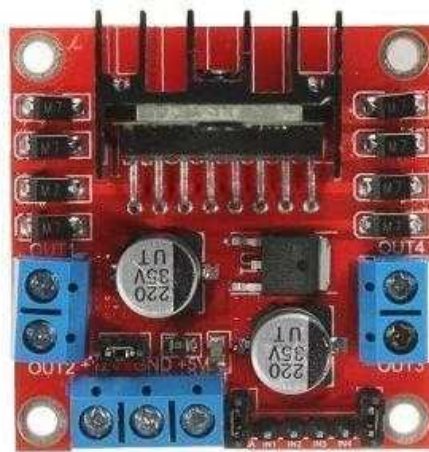
Resim 11 Pan-Tilt Açı Değerlerinin Seri Dinleme'de Gözlemlenmesi

1.7. 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
Motor A	IN1, IN2	Motor A, Giriş Pinleri
	OUT1, OUT2	Motor A, Çıkış Pinleri
	ENA	Motor A, Hız Pini
Motor B	IN3, IN4	
	OUT3, OUT4	
	ENB	
Güç	12V	Motor Beslemesi (5 – 46V ama 6 – 24V önerilir.)
	GND	Ortak GND
	5V	5V Gerilim Çıkışı

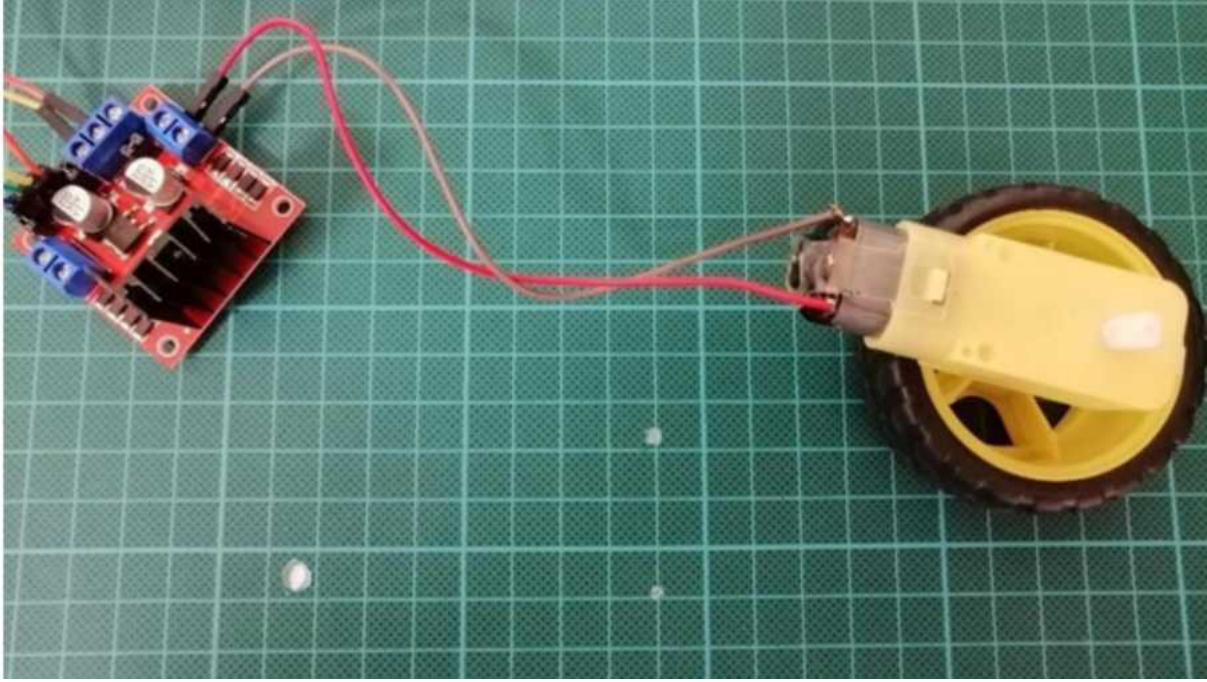
Tablo 1 L298N Motor Sürücü Kartı Pin Listesi



Resim 12 L298N Motor Sürücü Kartı

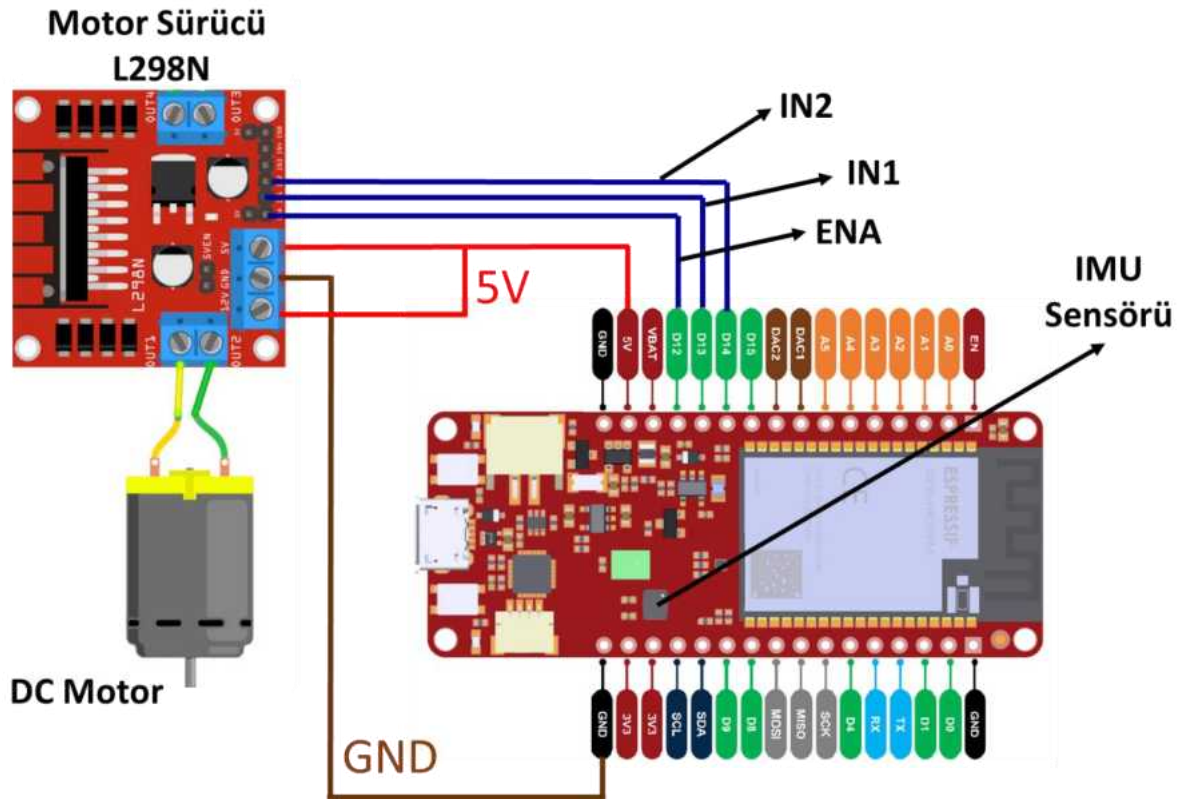
Uygulama Adımları

Adım 1: Erkek – erkek jumper kabloların birer ucu kesilerek DC motora lehimlenir. 1 adet tekerlek DC motorun miline takılır (Resim 13).

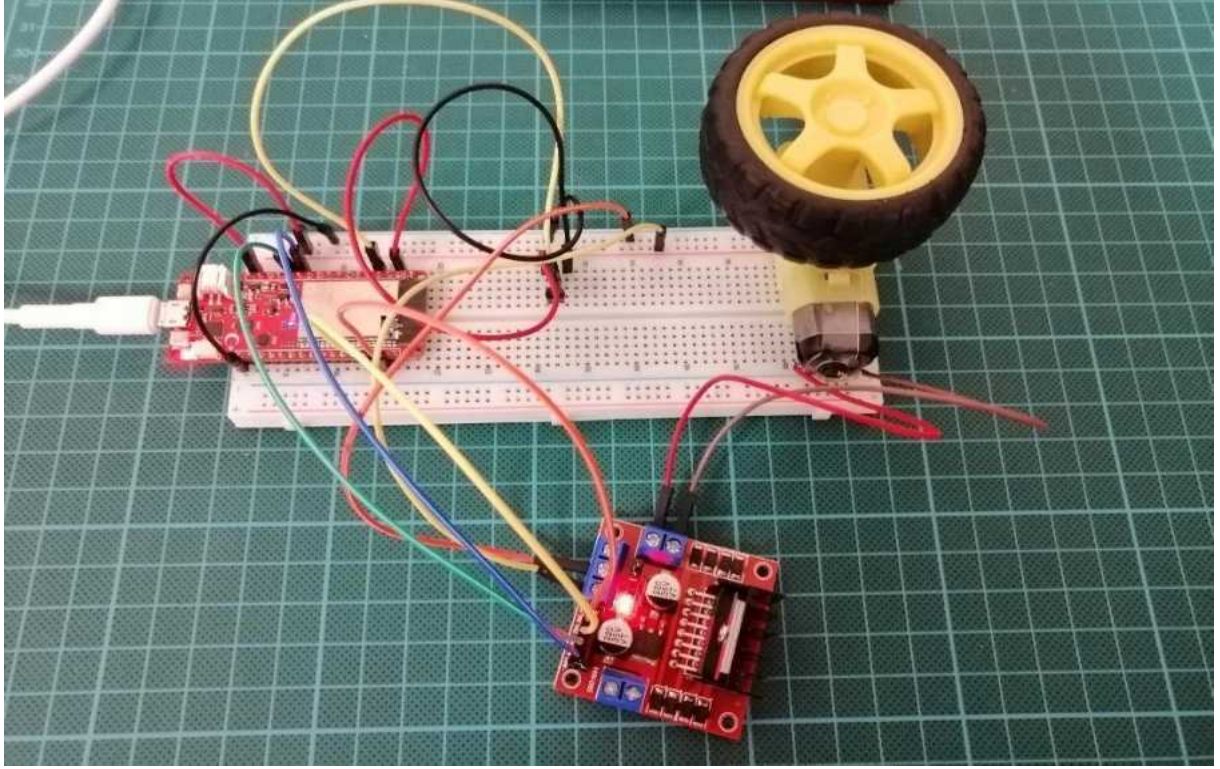


Resim 13 Kabloların Motora Lehimlenmesi ve Tekerleğin Montajı

Adım 2: Kurulacak Devre (Resim 14).



Resim 14 IMU Sensörü ile DC Motor Kontrolü Uygulama Devre Şeması



Resim 15 Deneyap Kart / Deneyap Kart 1A, Motor Sürücü ve Motor Bağlantısı

Adım 3: Uygulama kodları Arduino IDE üzerinde yazılarak Deneyap Kart / Deneyap Kart 1A'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

```
1 #include "lsm6dsm.h"
2 #define MOT_ENA D12 // Motor hız kontrol pini
3 #define MOT_DIR1 D13 // Motor yon secme pin1
4 #define MOT_DIR2 D14 // Motor yon secme pin2
5
6 float angle_x=0.; //x açısı
7 int hiz=0;
8 LSM6DSM IMU; //Deneyap Geliştirme Kartı üzerinde bulunan IMU sensörü modeli
9
10 void setup() {
11     IMU.begin();
12     pinMode(MOT_ENA, OUTPUT); // Motor hız kontrol pini cikis olarak ayarlandı
13     pinMode(MOT_DIR1, OUTPUT); // Motor yon secme pin1 cikis olarak ayarlandı
14     pinMode(MOT_DIR2, OUTPUT); // Motor yon secme pin2 cikis olarak ayarlandı
15 }
16
17 void loop() { /*açı hesaplama denklemi (sqrt: karekök alma işlemi, pow: üs alma işlemi, PI: pi (π) sayısı)*/
18     angle_x = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
19     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
20     hiz= abs(angle_x);
21     if(angle_x > 0) { /* İleri yon */
22         digitalWrite(MOT_DIR1, HIGH);
23         digitalWrite(MOT_DIR2, LOW);
24     } else if(angle_x < 0) { /* Geri yon */
25         digitalWrite(MOT_DIR1, LOW);
26         digitalWrite(MOT_DIR2, HIGH);
27     }
28
29     digitalWrite(MOT_ENA, LOW);
30     delay(100/hiz);
31     digitalWrite(MOT_ENA, HIGH);
32     delay(100/hiz);
33 }
```

Resim 16 IMU Sensörü ile DC Motor Kontrolü Uygulama Kodu

Adım 4: Kodlar yazılıp karta yüklendikten sonra Deneyap Kart / Deneyap Kart 1A'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.
- 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 derste temel kavramlar ve genel robot sistemleri konusunda bilgi verilmiştir. Robot tipleri, serbestlik dereceleri konusunda örnekler üzerinden paylaşımlarda bulunulmuştur. IMU sensörü ile DC Motor kontrolü yapılmıştır.

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

- 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?

- 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

Bir güneş enerjisi santralinde, güneş panellerinin güneşi takip etmesi için nasıl bir sistem tasarlanması gerekir? Bu sistemi çizerek tasarlayınız ve gerekli serbestlik derecesini hesaplayınız. Öğitmenler, robotlarla ilgili aşağıda verilen linklerdeki videolar üzerinden öğrencilerle beyin fırtınası yaparak konu hakkında bilgi seviyelerini artırabilir. Öğitmenler 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 interface 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: İki Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A'nın temel özelliklerini bilir.

Haftanın Kazanımları:

- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinden kumanda kolu kurulumunu ve programlamasını uygular.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinden servo motor kurulumunu ve programlamasını uygular.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinden kumanda kolu ile servo motorları kontrol eder.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A 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 eklemlili) bir robotun mekanik ve elektronik donanım çalışmalarını ve Deneyap Kart / Deneyap Kart 1A 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:

Malzeme Listesi	
Deneyap Kart / Deneyap Kart 1A	1 adet
Jumper Kablo	30+ Adet
Mikro USB Bağlantı Kablosu	1 adet
Harici 5V USB Batarya / 9V Pil / 2li 18650 Pil,	1 adet
Abeslang	Yeterli Kadar
LDR sensörü	3 Adet
Potansiyometre	1 Adet
Kumanda kolu (Joystick)	1 Adet
Servo Motor	2 Adet
Çift taraflı silikon bant/silikon tabancası ve makas	1 Adet
El Feneri veya Telefon Feneri	1 Adet
L – Köşebent	1 Adet

Haftanın İşlenişi:

Gözle: Deneyap Kart / Deneyap Kart 1A 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.

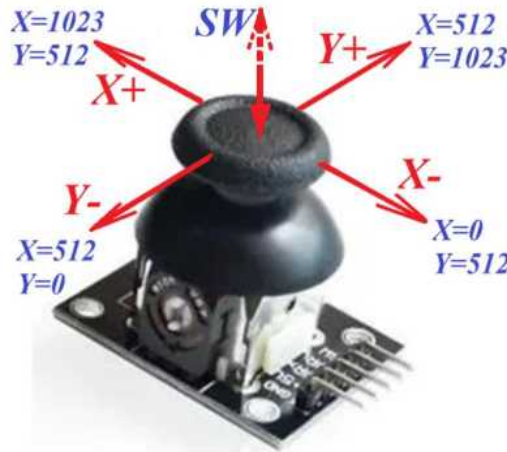
1. GÖZLE VE UYGULA

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 5'te 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 17 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.

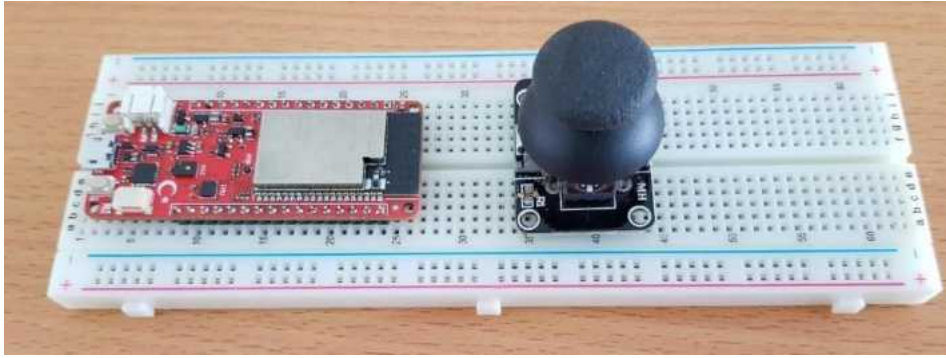
1.2. Uygula: Deneyap Kart / Deneyap Kart 1A Üzerinde Kumanda Kolu Çalıştırılması

Bu uygulamada X ve Y yönlerinde hareket ettirilebilen ve üzerinde buton bulunan bir kumanda kolu in Deneyap Kart / Deneyap Kart 1A'ya bağlantıları yapılacak ve kumanda kolu değerlerinin serial monitör üzerinde gösterilecektir. İleriki bölümlerde kumanda kolu sıklıkla kullanılacak olup, bağlantıları benzer şekilde yapılacaktır.

Adım 1: Deneyap Kart / Deneyap Kart 1A Breadboard üzerine yerleştirilir. Resim 18'de gösterildiği gibi, kumanda kolu tabanına çift taraflı yapışkan silikon bantlar yerleştirilir ve breadboard üzerine yapıştırılır. Resim 18'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 tercihinə bırakılmıştır.

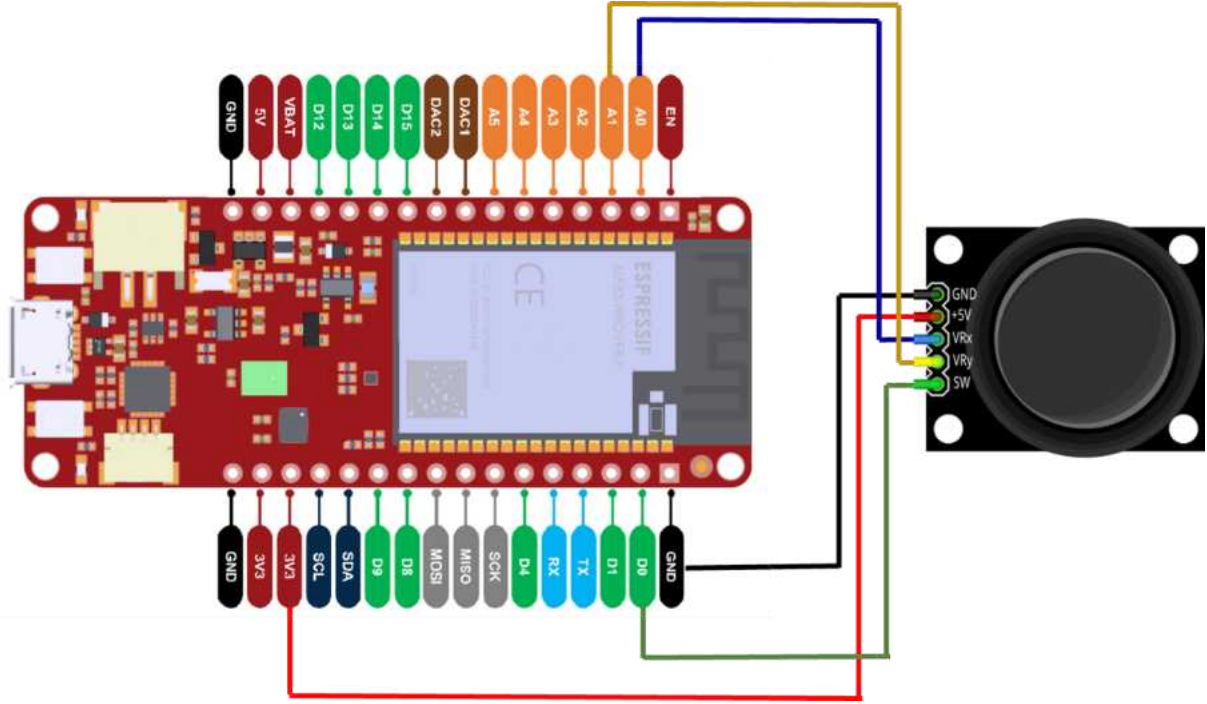


Resim 18 Çift Taraflı Silikon Bant, Makas ve Kumanda Kolu Tabanında Çift Taraflı Yapışkan Silikon Bant



Resim 19 Breadboard Üzerine Konumlandırılmış Kumanda Kolu.

Adım 2: Uygulama için gerekli Deneyap Kart / Deneyap Kart 1A ve kumanda kolu bağlantıları Resim 20'de verilmiştir. Kumanda kolunun güç bağlantıları (+5V ve GND) Deneyap Kart / Deneyap Kart 1A'nın 3V3 ve GND girişlerine yapılacaktır. Deneyap Kart / Deneyap Kart 1A'nın analog girişleri 3.3V'a kalibre edildikleri için kumanda kolu 5V yerine 3.3V ile beslenmelidir. Kumanda kolunun VRx ve VRy çıkışları Deneyap Kart / Deneyap Kart 1A'nın A0 ve A1 girişlerine, SW çıkışı ise D0 girişine bağlanır.



Resim 20 Kumanda Kolu Pinlerinin Deneyap Kart / Deneyap Kart 1A Bağlantıları

Adım 3: Deneyap Kart / Deneyap Kart 1A micro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve Deneyap Kart / Deneyap Kart 1A'na yüklenir. Gerekli kodlar aşağıda verilmiştir.

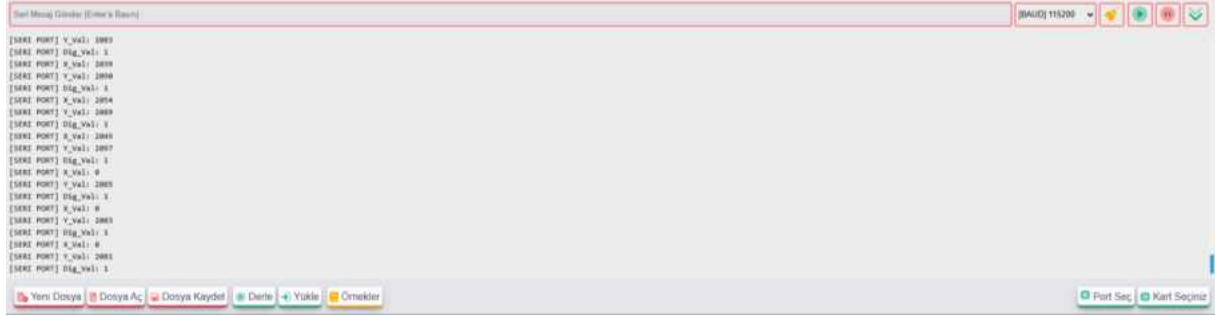
```
1 // Kumanda kolu butonunun dijital pini ve X-Y hareketlerinin analog pinleri için değişken tanımlamaları
2 #define SW_pin D0
3 #define X_pin A0
4 #define Y_pin A1
5
6 // Program içerisinde kullanılacak değişkenler de tanımlanır.
7 int X_Val, Y_Val, Dig_Val;
8
9 /* Başlangıç düzenlemeleri aşağıdaki gibi yapılır. Değişkenlerin ekrana yazdırılacağı Serial Monitör
10 örnekleme frekansı ve analog/dijital pinlerin giriş, çıkış ayarlamaları yapılır. */
11 void setup() {
12   Serial.begin(115200); // Seri portun 115200 baudrate hızında başlatılması.
13   pinMode(SW_pin, INPUT_PULLUP); /* Switch (anahtar) değişkeni mod'u HIGH' e ayarlanır.
14   Butona basıldığında LOW değerini alır.*/
15   digitalWrite(SW_pin, HIGH);
16   // kumanda kolu eksen değişken değerlerinin alınacağı 2 pin'in INPUT (giriş) olarak tanımlanması.
17   pinMode(X_pin, INPUT);
18   pinMode(Y_pin, INPUT);
19 }
20
21 /* Kumanda kolu fonksiyonlarını algılamak için dijital ve analog girişlerden elde edilen
22 değerlerin genel döngü (loop) içerisinde atama ve ekrana yazdırma işlemleri yapılır. */
23 void loop() {
24   X_Val = analogRead(X_pin);
25   Y_Val = analogRead(Y_pin);
26   Dig_Val = digitalRead(SW_pin);
27   Serial.print("X_Val: ");
28   Serial.println(X_Val);
29   Serial.print("Y_Val: ");
30   Serial.println(Y_Val);
31   Serial.print("Dig_Val: ");
32   Serial.println(Dig_Val);
33   delay(100); /* Burada bu delay (gecikme) fonksiyonu içindeki milisaniye değeri artırılarak
34   serial monitor'deki akış yavaşlatılabilir ve öğrencilerin kumanda kolu değerlerini daha
35   rahat okumaları sağlanabilir. */
36 }
```

Resim 21 Deneyap Kart / Deneyap Kart 1A Üzerinde Kumanda Kolu Çalıştırılması Kodu

Uyarı:

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

Adım 4: Serial Monitor ekranında kumanda kolu fonksiyonları gözlemlenmelidir (Resim 22).



Resim 22 Serial Monitörde 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 Kart / Deneyap Kart 1A ile Çift Servo Motor Kontrolü

Adım 1: Deneyap Kart / Deneyap Kart 1A Breadboard üzerine yerleştirilir. Resim 23'te 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 24'te 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.

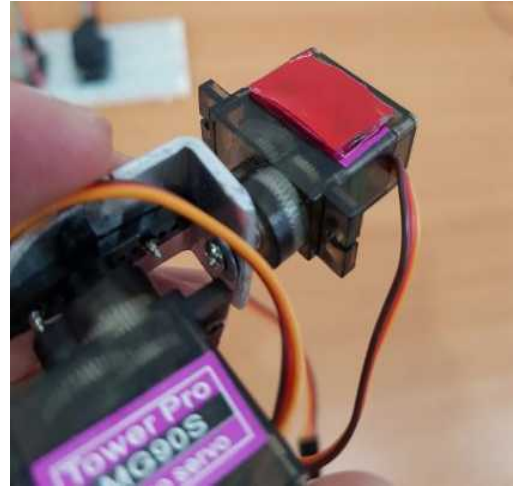


Resim 23 Çift Motor Düzenğinde Kullanılan Parçalar



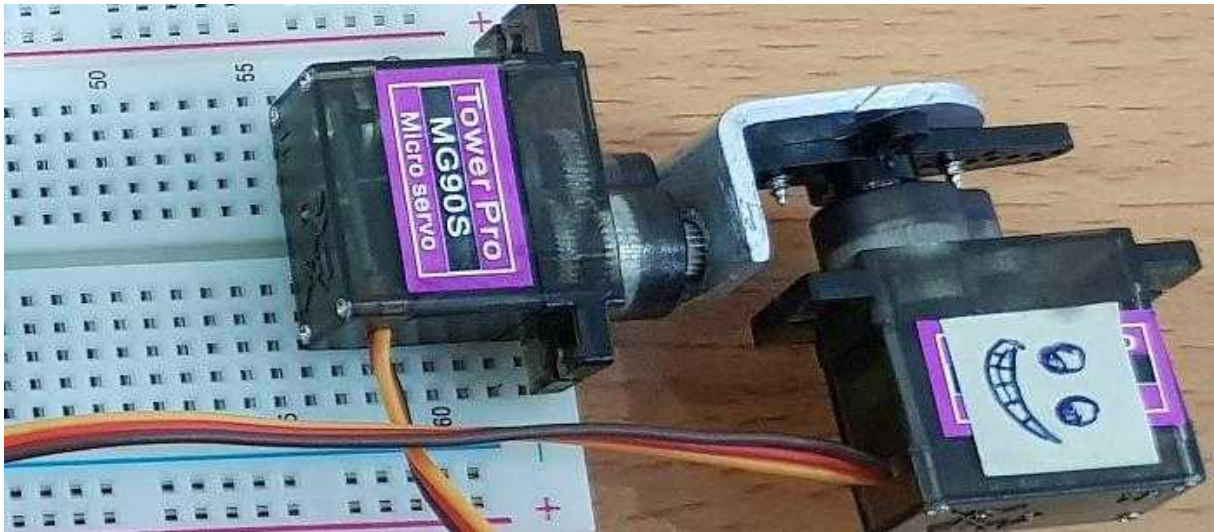
Resim 24 3B Yazıcı ile Basılmış L-Profil Kullanılarak Hazırlanmış Çift Motor Düzeneği

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 25). Burada da isteğe bağlı olarak, çift taraflı silikon bant yerine silikon tabancası kullanılabilir.



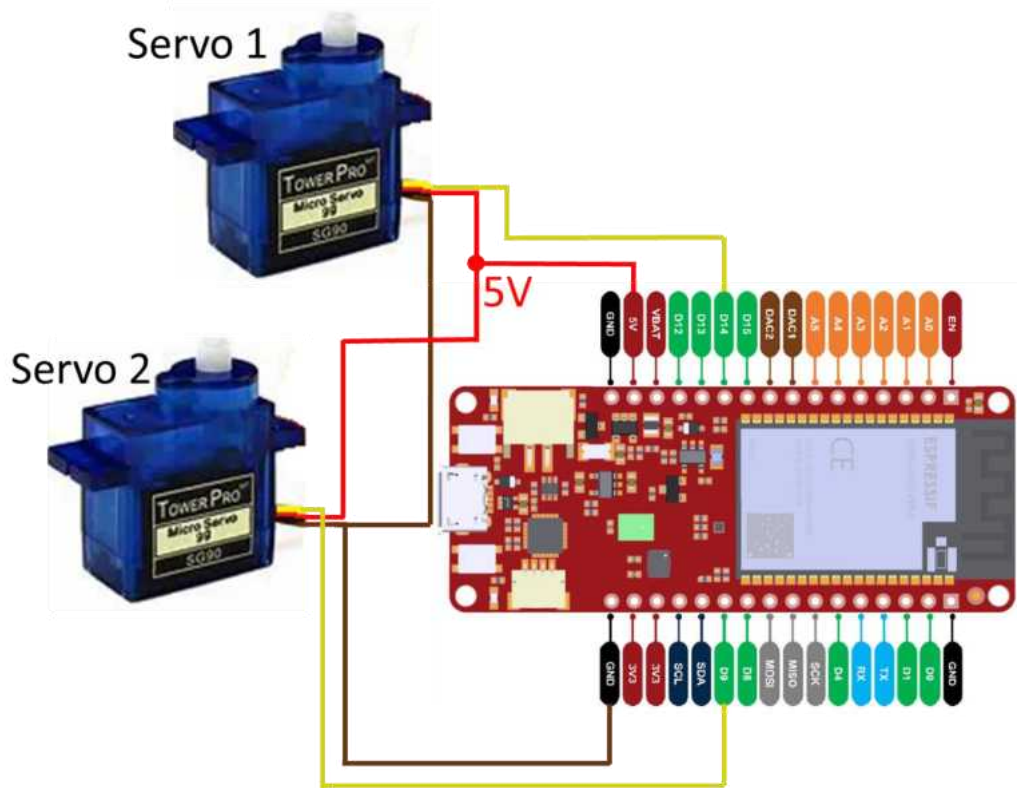
Resim 25 Çift Taraflı Silikon Bant ve Etiket Yapıştırılması

Adım 3: Çift motor düzeneğinin silikon (veya silikonlandırılmış) bantlı yüzeyi breadboard'un diğer boş tarafına yerleştirilir ve motor üzerine nazıkçe baskı yapılarak bir süre beklenir (Resim 26).

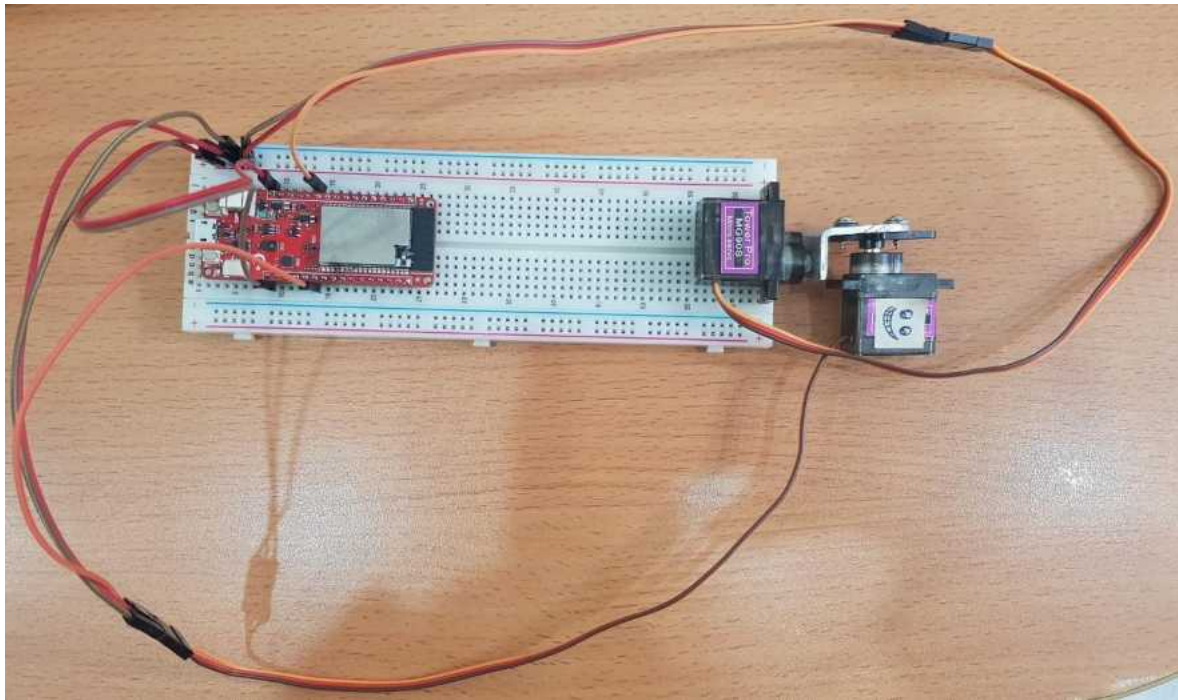


Resim 26 Çift Motor Düzeneğinin Breadboard Üzerine Yerleştirilmesi

Adım 4: Uygulama için gerekli Deneyap Kart / Deneyap Kart 1A ve servo motor bağlantıları Resim 27’de verilmiştir. Servo motorların güç bağlantıları (+5V ve GND) Deneyap Kart / Deneyap Kart 1A’nın 5V ve GND girişlerine yapılacaktır. Motorların kontrol girişleri de Deneyap Kart / Deneyap Kart 1A’nın D9 ve D14 pinlerine bağlanacaktır (Resim 28).



Resim 27 Çift Servo Motorun Deneyap Kart / Deneyap Kart 1A Bağlantıları



Resim 28 Servo Motorların Deneyap Kart / Deneyap Kart 1A Bağlantısı

Adım 5: Deneyap Kart / Deneyap Kart 1A bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```
1 #include <Deneyap_Servo.h>
2 #define SERVOPIN1 D14
3 #define SERVOPIN2 D9
4
5 Servo myservo1, myservo2; // Servo motorların tanımlanması.
6 int pos = 0; //İlk pozisyon. Her 2 motor için ortak değişken.
7
8 // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
9 void setup() {
10     myservo1.attach(SERVOPIN1);
11     myservo2.attach(SERVOPIN2,1);
12 }
13
14 /* Servo motorların kontrol girişlerine kontrol sinyallerin gönderilmesi işlemi
15 genel döngü (loop) içerisine yazılan kodlar ile yapılır. Aşağıdaki kod yazılarak,
16 motorlara tek bir pos değişkeni ile seri bir demo hareket yaptırılmaktadır. */
17 void loop() {
18     for (pos = 0; pos <= 180; pos += 1) {
19         // Motorlara pos değerinin yollanması
20         myservo1.write(pos);
21         myservo2.write(pos);
22         delay (10); // Her bir döngüde 10 milisaniye bir gecikme ilave edilmesi.
23     }
24
25     for (pos = 180; pos >= 0; pos -= 1) {
26         // Motorlara pos değerinin yollanması
27         myservo1.write(pos);
28         myservo2.write(pos);
29         delay (10); // Her bir döngüde 10 milisaniye bir gecikme ilave edilmesi.
30     }
31 }
```

Resim 29 Deneyap Kart / Deneyap Kart 1A ile Çift Servo Motor Kontrolü Kodu

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.

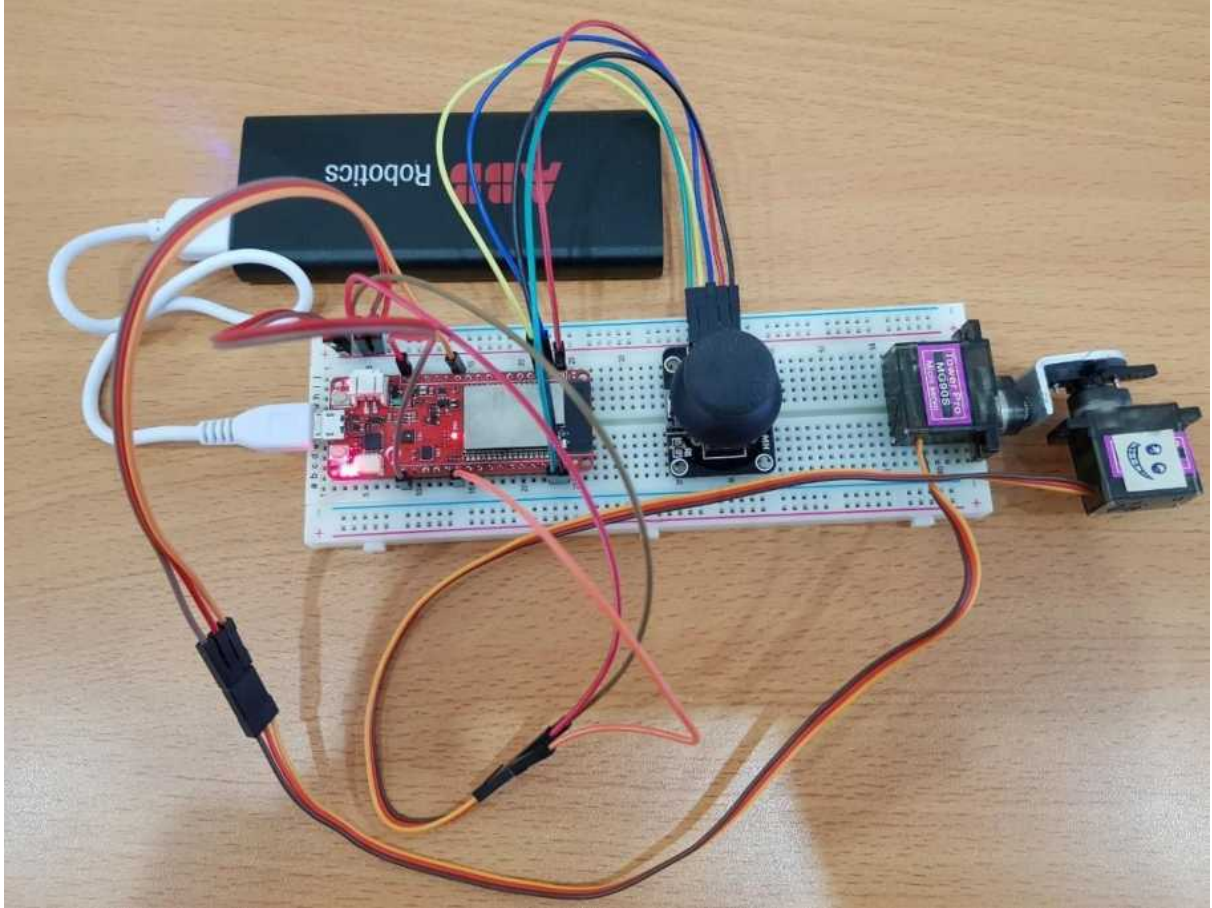
Adım 7: Programın Deneyap Kart / Deneyap Kart 1A'ya yüklemesi bittiğinde, her iki motorun da 0-180°'lik açı aralığında sürekli olarak ileri-geri döndüğü gözlemlenmelidir.

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

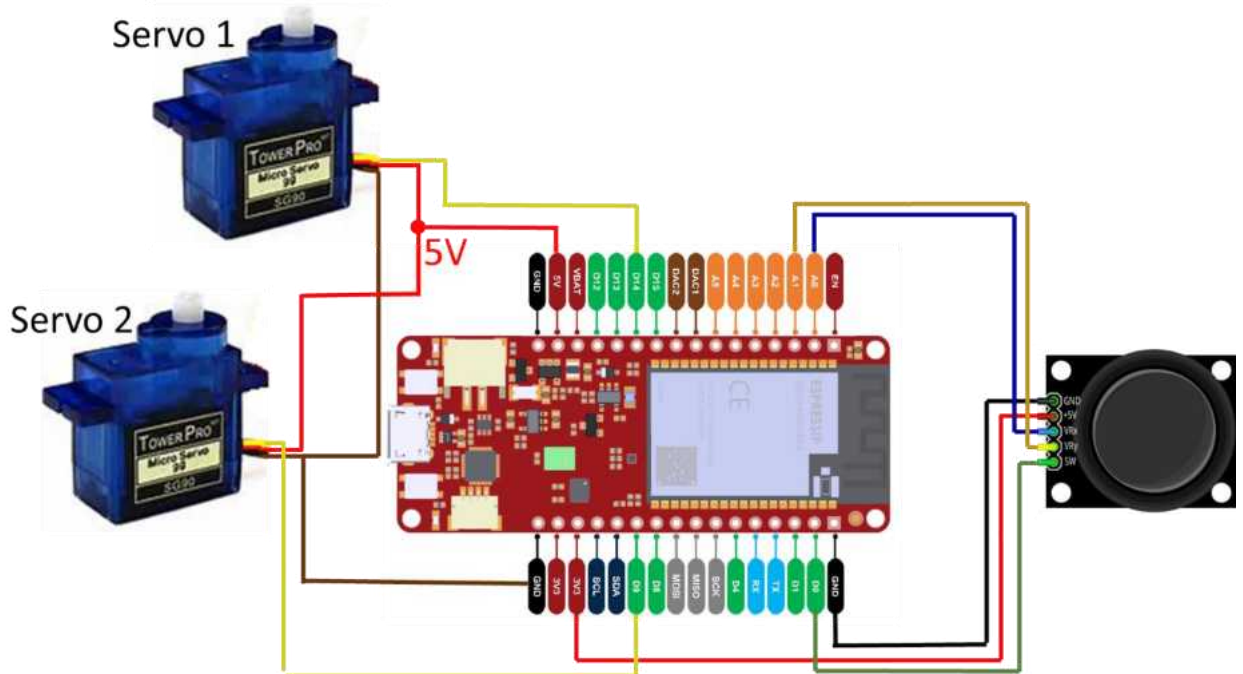
Bu uygulamada, hemen önceki iki uygulamada yapılan sistemlerin bir araya getirilmesi ve çalıştırılması aşamaları verilmiştir. Bir kumanda kolu ile 2 servo motorun hareketi kontrol edilecektir.

Uygulama Adımları

Adım 1: Bir önceki uygulamada gerçekleştirilen deney düzeneğine bir adet kumanda kolu eklenir (Resim 30). Uygulama için gerekli Deneyap Kart / Deneyap Kart 1A, servo motor ve kumanda kolu bağlantıları Resim 31'de verilmiştir. Kumanda kolunun güç bağlantıları Deneyap Kart / Deneyap Kart 1A'nın 3V3 ve GND girişlerine, VRx ve VRy çıkışları sırası ile Deneyap Kart / Deneyap Kart 1A'nın A0 ve A1 girişlerine, buton çıkışı ise D0 girişine bağlanır. Burada, harici güç kaynağı, Deneyap Kart / Deneyap Kart 1A 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 30 Kumanda Kolu ve Servo Motorların Bir Breadboard Üzerinde Jumper Kablolar ile Deneyap Kart / Deneyap Kart 1Ana Bağlanması ve Harici Bir Batarya ile Çalıştırılması



Resim 31 Deneyap Kart / Deneyap Kart 1A, Servo Motor ve Kumanda Kolu Bağlantıları

Adım 2: Deneyap Kart / Deneyap Kart 1A bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```

1 #include <Deneyap_Servo.h>
2 #define SERVOPIN1 D14
3 #define SERVOPIN2 D9
4 #define SW_pin D0
5 #define X_pin A0
6 #define Y_pin A1
7
8 Servo myservo1, myservo2; // Servo motorların tanımlanması
9 int X_Val, Y_Val; // kumanda kolu eksen değişkenleri.
10 int pos1=0, pos2=0; // Motorların pozisyon değişkenleri.
11 int dly = 10; // Genel döngüde ilave edilen gecikme için kullanılan değişken.
12 int stp = 1; /* Motorların dönüş açısının değiştirilmesi durumunda her döngüde kullanılan adım değişkeni.
13 Örneğin servo 1 pozitif yöne dönsün istendiğinde her döngüde, Pos1+=stp olacaktır (kod aşağıda). */
14 int Dig_Val = 0; // Buton değeri
15
16 // Servo motorların açısı sınır değerleri. Ne oldukları değişken isimlerinden bellidir.
17 int teta1min=0;
18 int teta1max=180;
19 int teta2min=0;
20 int teta2max=180;
21
22 // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
23 void setup() {
24 // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
25 myservo1.attach(SERVOPIN1);
26 myservo2.attach(SERVOPIN2,1);
27 pinMode(SW_pin, INPUT_PULLUP); /* Switch (anahtar) değişkeni mod'u HIGH' e ayarlanır.
28 Butona basıldığında LOW değerini alır. */
29 digitalWrite(SW_pin, HIGH);
30 // kumanda kolu eksen değişken değerlerinin alınacağı 2 pin'in INPUT (giriş) olarak tanımlanması.
31 pinMode(X_pin, INPUT);
32 pinMode(Y_pin, INPUT);
33 }
34
35 void loop() {
36 // kumanda kolu X ve Y değişiminin okunması.
37 X_Val = analogRead(X_pin);
38 Y_Val = analogRead(Y_pin);
39 Dig_Val=digitalRead(SW_pin);
40
41 // kumanda kolu butonuna her basıldığında, servo motor dönüş hızlarını belirleyen dly değişkenine değerler atanması.
42 if (Dig_Val<1) {
43 if (dly==5) {
44 dly=10;
45 } else {
46 if(dly==10) {
47 dly=20;
48 } else if {
49 (dly==20) dly=5
50 }
51 }
52 delay (500); // Butona basış tespit edildikten sonra 500 ms bir gecikme ilave edilmesi.
53 }
54
55 // Motorların limitleri dahilinde dönüşlerine izin verilmesi:
56 if (X_Val<1000 && pos1>teta1min) pos1-=stp;
57 if (X_Val>3000 && pos1<teta1max) pos1+=stp;
58 if (Y_Val<1000 && pos2>teta2min) pos2-=stp;
59 if (Y_Val>3000 && pos2<teta2max) pos2+=stp;
60
61 // Motorlara dönüş komutlarının verilmesi.
62 myservo1.write(pos1);
63 myservo2.write(pos2);
64 delay (dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
65 }

```

Resim 32 İki Motorlu Bir Sistemin Kumanda Kolu ile Kontrolü Kodu

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.

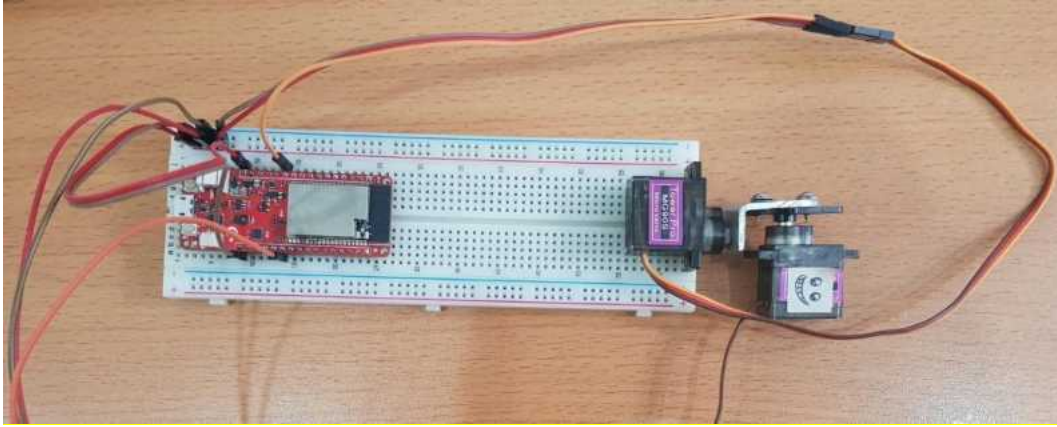
Adım 3: Programın Deneyap Kart / Deneyap Kart 1A'ya yüklemesi bittiğinde motorların kumanda kolu ile kontrolünün yapılabildiği gözlemlenmelidir.

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

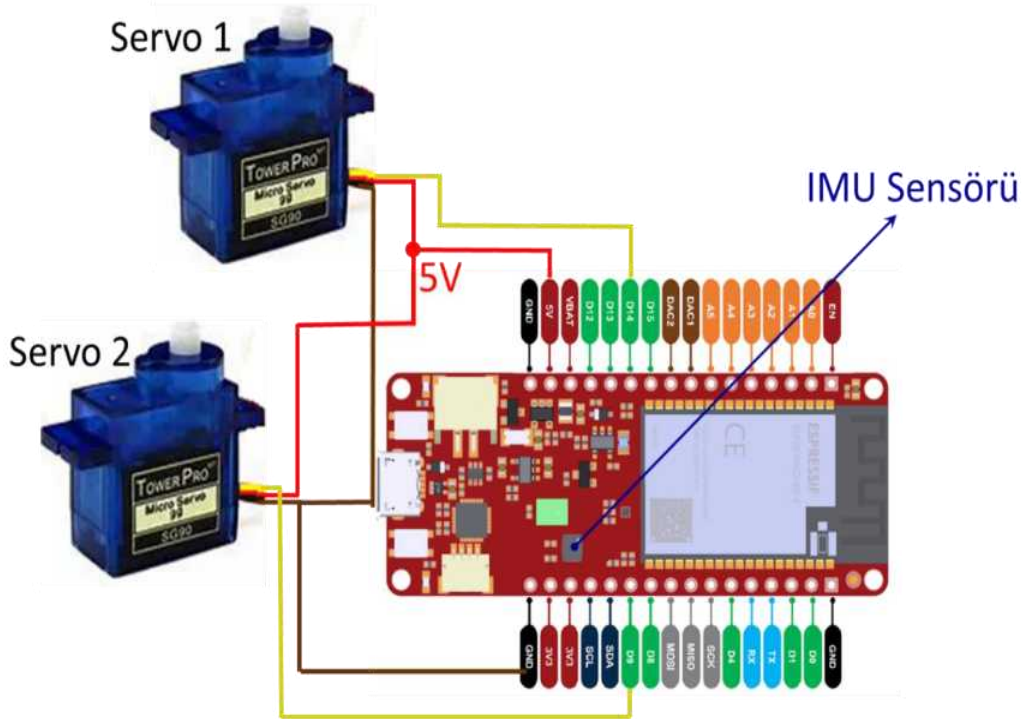
Bu bölümde Deneyap Kart / Deneyap Kart 1A üzerinde bulunan IMU sensörü ile iki adet servo motorun hareket kontrolü yapılacaktır.

Uygulama Adımları

Adım 1: Bir önceki uygulamada gerçekleştirilen deney düzeneğindeki kumanda kolu bağlantıları sökülerek sadece servo motorlar bırakılır. (Resim 33). Uygulama için gerekli Deneyap Kart / Deneyap Kart 1A ve servo motor bağlantıları Resim 34'te verilmiştir.



Resim 33 IMU Sensörü ile Çift Servo Motor Konum Kontrolü Uygulaması



Resim 34 IMU Sensörü ile Çift Servo Motor Konum Kontrolü Bağlantı Şeması

Adım 2: Deneyap Kart / Deneyap Kart 1A bir mikro USB kablo ile bilgisayara bağlanır. Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```

1 #include <Deneyap_Servo.h>
2 #include "lsm6dsm.h"
3
4 #define SERVOPIN1 D14
5 #define SERVOPIN2 D9
6
7 // Kart üzerindeki ivme ölçerin tanımlanması.
8 LSM6DSM IMU;
9 // Servo motorların tanımlanması
10 Servo myservo1;
11 Servo myservo2;
12 // X ve Y eksenlerindeki ivme değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
13 float accAngleX=0.;
14 float accAngleY=0.;
15 // Motorların dönüş açısı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
16 int pos1 = 90;
17 int pos2 = 90;
18
19 void setup() {
20     IMU.begin();
21     // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
22     myservo1.attach(SERVOPIN1);
23     myservo2.attach(SERVOPIN2,1);
24 }
25
26 void loop() {
27     // İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
28     accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
29     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
30     accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
31     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
32
33     // Motorlara belirlenen X ve Y eğim açılarının uygulanması:
34     pos1=90+int(accAngleX);
35     pos2=90+int(accAngleY);
36
37     myservo1.write(pos1);
38     myservo2.write(pos2);
39     delay(10); // Genel döngüye 10 milisaniyelik bir gecikme uygulanması.
40 }

```

Resim 35 IMU Sensörü ile Çift Servo Motor Konum Kontrolü Kodu

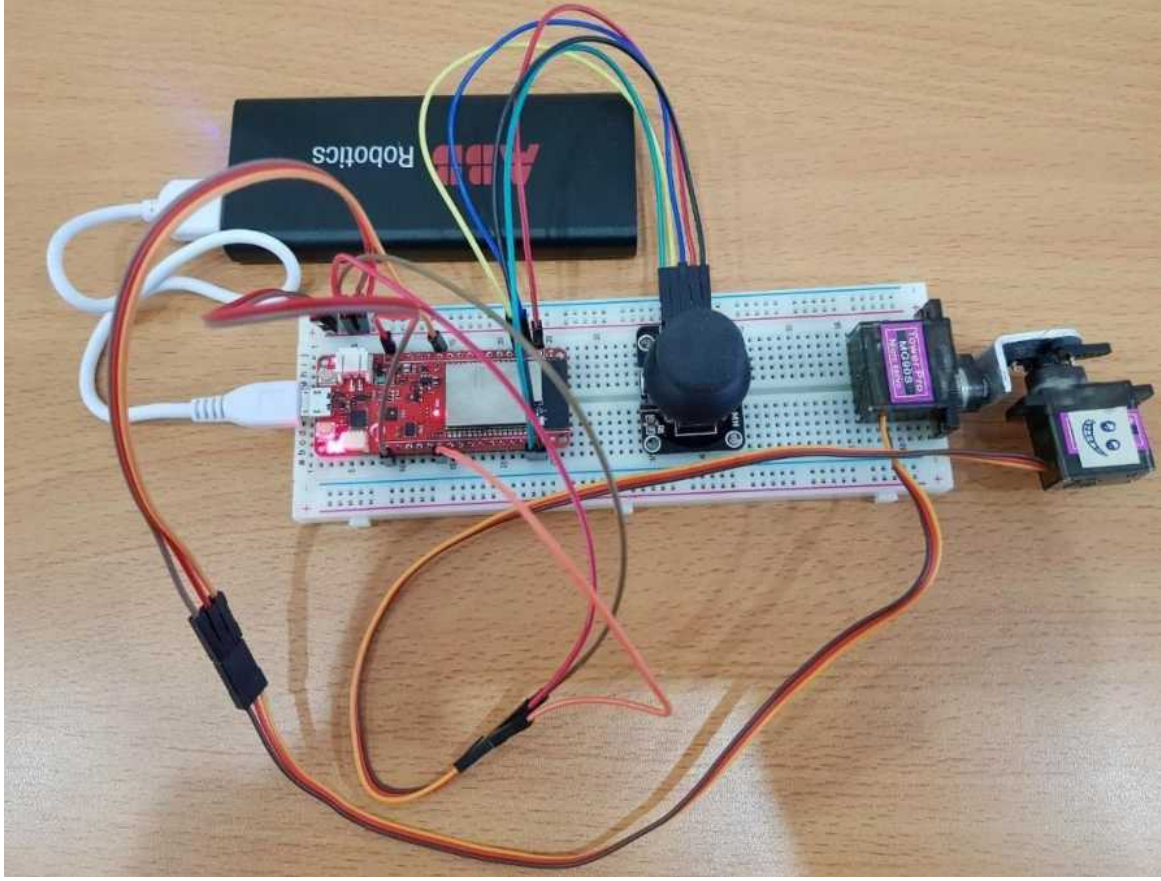
Adım 3: Programın Deneyap Kart / Deneyap Kart 1A'ya yüklemesi bittiğinde motorların IMU sensöründen gelen değerlere göre davranış gösterdiği görülecektir. Basit bir gimbal uygulaması olarak, çift motorlu mekanizmanın gülen yüzü her zaman yukarı yöne bakacaktır.

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

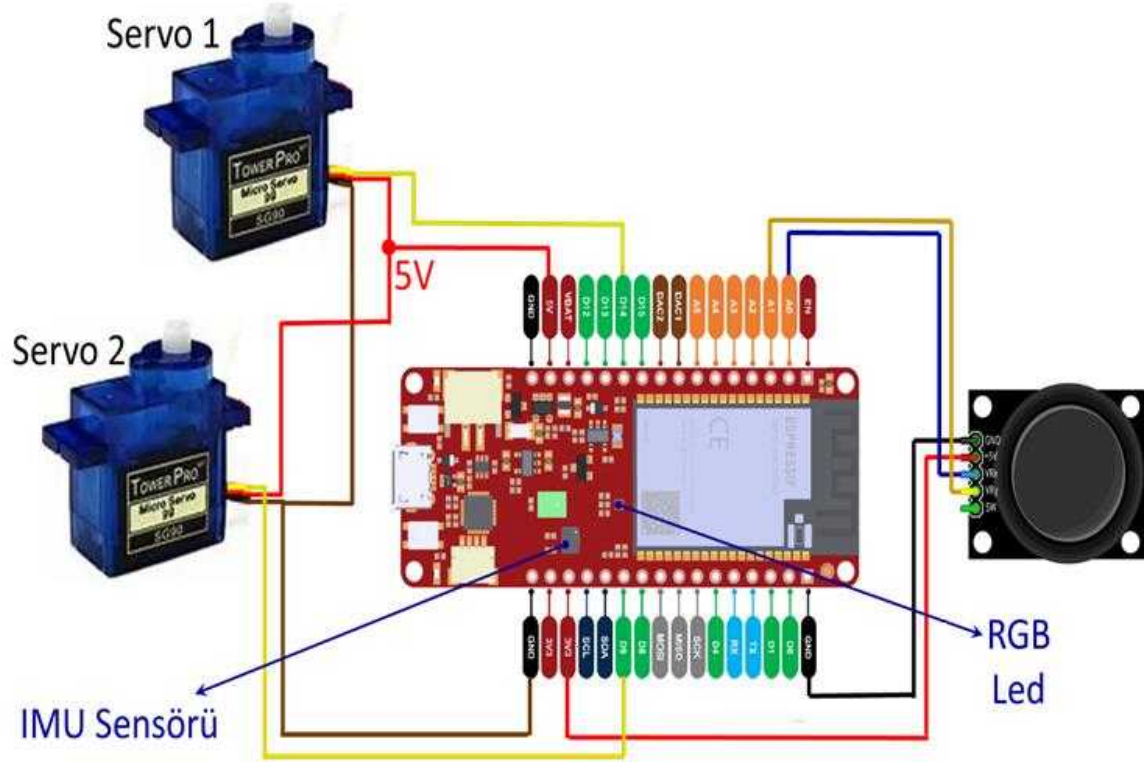
Bu uygulamada servo motorların hem kumanda kolu hem de Deneyap Kart / Deneyap Kart 1A üzerindeki IMU sensörü ile kontrolü ve kart üzerindeki RGB ledlerin kullanımı gösterilecektir.

Uygulama Adımları

Adım 1: Deneyap Kart / Deneyap Kart 1A, kumanda kolu ve servo motorlar breadboard üzerine yerleştirilir. Servo motorların ve Kumanda kolunun Deneyap Kart / Deneyap Kart 1A'ya bağlantıları yapılır (Resim 36). Uygulama için gerekli Deneyap Kart / Deneyap Kart 1A ve servo motor bağlantıları Resim 37'de verilmiştir.



Resim 36 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Uygulaması



Resim 37 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Bağlantı Şeması

Adım 2: Arduino PC uygulaması üzerinde program yazılır, kaydedilir, derlenir ve Deneyap Kart / Deneyap Kart 1A yüklenir. Bu uygulamada kumanda kolu üzerindeki buton yerine Deneyap Kart / Deneyap Kart 1A üzerinde USB portun solunda bulunan buton (BUILTIN_KEY) ve yine kartın üzerindeki renkli led kullanılacaktır.

```
1 #include <Deneyap_Servo.h>
2 #include "lsm6dsm.h"
3
4 // Dijital servo pin bağlantıları:
5 #define SERVOPIN1 D14
6 #define SERVOPIN2 D9
7
8 // kumanda kolu eksen analog bağlantı pinleri
9 #define X_pin A0
10 #define Y_pin A1
11
12 int X_Val, Y_Val; // kumanda kolu eksen değişkenleri
13 int dly = 5; // Genel döngüdeki gecikme değişkeni
14 int step = 1; // Servo motorların dönüş durumunda her bir döngüdeki dönüş adımı
15 int X_jys=0, Y_jys=0; // kumanda kolu eksen değişkenlerinden elde edilen motor dönüş açıları
16
17 // Motor dönüş açı alt ve üst limitleri:
18 int totalmin=0;
19 int totalmax=180;
20 int totalmin=0;
21 int totalmax=180;
22
23 // İvme ölçerden elde edilen X ve Y eksenlerindeki açısal dönüşler:
24 float accAngleX=0;
25 float accAngleY=0;
26
27 // IMU'nun tanımlanması
28 LSM6DSM IMU;
29
30 // Servo motorların tanımlanması
31 Servo myservo1;
32 Servo myservo2;
33
34 // Motorların dönüş açı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
35 int pos1 = 90;
36 int pos2 = 90;
37
38 void setup() {
39   IMU.begin();
40
41   // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
42   myservo1.attach(SERVOPIN1);
43   myservo2.attach(SERVOPIN2,1);
44
45   // Deneyap kartı üzerindeki butonun giriş olarak tanımlanması
46   pinMode(BUILTIN_KEY, INPUT);
47
48   // kumanda kolu pinlerinin giriş olarak tanımlanması
49   pinMode(X_pin, INPUT);
50   pinMode(Y_pin, INPUT);
51
52   // RGB led pinlerinin çıkış olarak tanımlanması.
53   pinMode(LED0, OUTPUT);
54   pinMode(LED1, OUTPUT);
55   pinMode(LED2, OUTPUT);
56   digitalWrite(LED0, LOW); //Kırmızı led açık (aktif)
57   digitalWrite(LED1, HIGH); // Yeşil led sönük (pasif)
58   digitalWrite(LED2, HIGH); // Mavi led sönük (pasif)
59 }
60
```

Resim 38 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Kodu I. Kısım – Tanımlamalar ve Setup

```

61 /*İvme-ölçer verilerinin alınması, pan-tilt açısı değerlerinin hesaplanması, joystick verileri,
62 buton verileri, RGB led renk değişimleri ve motorların kontrol edilmesi işlemleri
63 genel döngü (loop) içerisinde yapılır.*/
64 void loop() {
65     // kumanda kolu X ve Y değişiminin okunması.
66     X_Val = analogRead(X_pin);
67     Y_Val = analogRead(Y_pin);
68
69     uint8_t buttonState = digitalRead(BUILTIN_KEY); // Buton durumu
70
71     /* Kart üzerindeki butona her basıldığında, servo motor dönüş hızlarını belirleyen
72     dly değişkenine değerler atanması ve buna göre kart üzerindeki RGB led'lerinin değiştirilmesi.*/
73     if(buttonState == LOW) { /* Butona basılma durumu*/
74         if (dly==5) {
75             dly=10;
76             digitalWrite(LED_R, HIGH);
77             digitalWrite(LED_G, LOW); //Yeşil led aktif
78         } else if(dly==10) {
79             dly=20;
80             digitalWrite(LED_G, HIGH);
81             digitalWrite(LED_B, LOW); //Mavi led aktif
82         } else if (dly==20) {
83             dly=5;
84             digitalWrite(LED_R, LOW); //Kırmızı led aktif
85             digitalWrite(LED_B, HIGH);
86         }
87         delay(300); // Butona basış tespit edildikten sonra 300 ms bir gecikme ilave edilmesi.
88     }
89
90     // kumanda kolundan elde edilen dönüş açılarının limitler dahilinde uygulanması.
91     if (X_Val<1000 && pos1>teta1min) X_jys-=stp;
92     if (X_Val>3000 && pos1<teta1max) X_jys+=stp;
93     if (Y_Val<1000 && pos2>teta2min) Y_jys-=stp;
94     if (Y_Val>3000 && pos2<teta2max) Y_jys+=stp;
95
96     // İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
97     accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
98     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
99     accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
100     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
101
102     // Motor dönüş açıları:
103     pos1=90+int(accAngleX)+X_jys;
104     pos2=90+int(accAngleY)+Y_jys;
105
106     myservo1.write(pos1);
107     myservo2.write(pos2);
108
109     delay(dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
110 }

```

Resim 39 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Kodu II. Kısım – Loop

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; 750 değerinin girilmesi gerekmektedir. Dolayısıyla burada (3000/180) değeri “writeMicroseconds” fonksiyonunu kullanırkenki kalibrasyon katsayısıdır. Uygulamanın kodlarını “writeMicroseconds” fonksiyonuna göre tekrar düzenleyip çalıştıralım.

$$45 \times (3000/180) = 750$$

```

1 #include <Deneyap_Servo.h>
2 #include "lsm6dsm.h"
3
4 // Dijital servo pin bağlantıları:
5 #define SERVOPIN1 D14
6 #define SERVOPIN2 D9
7
8 // kumanda kolu eksen analog bağlantı pinleri
9 #define X_pin A0
10 #define Y_pin A1
11
12 int X_Val, Y_Val; // kumanda kolu eksen değişkenleri
13 int dly = 5; // Genel döngüdeki gecikme değişkeni
14 // *writeMicroseconds fonksiyonuna girilen değer normal açı değerinden çok yüksek olduğundan
15 stp değeri 10 kat fazla girilir. */
16 int stp = 10; // Servo motorların dönüş durumunda her bir döngüdeki dönüş adımı
17 int X_jys=0, Y_jys=0; // kumanda kolu eksen değişkenlerinden elde edilen motor dönüş açıları
18
19 // Motor dönüş açı alt ve üst limitleri:
20 int teta1min=0;
21 int teta1max=3000; // 180 dereceye karşılık gelmektedir.
22 int teta2min=0;
23 int teta2max=3000; // 180 dereceye karşılık gelmektedir.
24
25 // İvme ölçerden elde edilen X ve Y eksenlerindeki açısal dönüşler:
26 float accAngleX=0.;
27 float accAngleY=0.;
28
29 // IMU'nun tanımlanması
30 LSM6DSM IMU;
31
32 // Servo motorların tanımlanması.
33 Servo myservo1;
34 Servo myservo2;
35
36 // Motorların dönüş açı değişkenlerinin tanımlanması ve ilk değerlerinin atanması.
37 int pos1 = 1500; // 90 dereceye karşılık gelmektedir.
38 int pos2 = 1500; // 90 dereceye karşılık gelmektedir.
39
40 void setup() {
41     IMU.begin();
42
43     // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
44     myservo1.attach(SERVOPIN1);
45     myservo2.attach(SERVOPIN2,1);
46
47     // Deneyap kartı üzerindeki butonun giriş olarak tanımlanması
48     pinMode(BUILTIN_KEY, INPUT);
49
50     // kumanda kolu pinlerinin giriş olarak tanımlanması
51     pinMode(X_pin, INPUT);
52     pinMode(Y_pin, INPUT);
53
54     // RGB Led pinlerinin çıkış olarak tanımlanması.
55     pinMode(LED_R, OUTPUT);
56     pinMode(LED_G, OUTPUT);
57     pinMode(LED_B, OUTPUT);
58     digitalWrite(LED_R, LOW); //Kırmızı led açık (aktif)
59     digitalWrite(LED_G, HIGH); // Yeşil led sönmük (pasif)
60     digitalWrite(LED_B, HIGH); // Mavi led sönmük (pasif)
61 }
62

```

Resim 40 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Kodu Microseconds I. Kısım – Tanımlamalar ve Setup

```

63 //İvme-ölçer verilerinin alınması, pan-tilt açısı değerlerinin hesaplanması, joystick verileri,
64 buton verileri, RGB led renk değişimleri ve motorların kontrol edilmesi işlemleri
65 genel döngü (loop) içerisinde yapılır.*/
66 void loop() {
67     // kumanda kolu X ve Y değişiminin okunması.
68     X_Val = analogRead(X_pin);
69     Y_Val = analogRead(Y_pin);
70
71     uint8_t buttonState = digitalRead(BUILTIN_KEY); // Buton durumu
72
73     /* Kart üzerindeki butona her basıldığında, servo motor dönüş hızlarını belirleyen
74     dly değişkenine değerler atanması ve buna göre kart üzerindeki RGB led'lerinin değiştirilmesi.*/
75     if(buttonState == LOW) { /* Butona basılma durumu*/
76         if (dly==5) {
77             dly=10;
78             digitalWrite(LED_R, HIGH);
79             digitalWrite(LED_G, LOW); //Yeşil led aktif
80         } else if(dly==10) {
81             dly=20;
82             digitalWrite(LED_G, HIGH);
83             digitalWrite(LED_B, LOW); //Mavi led aktif
84         } else if (dly==20) {
85             dly=5;
86             digitalWrite(LED_R, LOW); //Kırmızı led aktif
87             digitalWrite(LED_B, HIGH);
88         }
89         delay(300); // Butona basış tespit edildikten sonra 300 ms bir gecikme ilave edilmesi.
90     }
91
92     // kumanda kolundan elde edilen dönüş açılarının limitler dahilinde uygulanması.
93     if (X_Val<1000 && pos1>teta1min) X_jys--stp;
94     if (X_Val>3000 && pos1<teta1max) X_jys++stp;
95     if (Y_Val<1000 && pos2>teta2min) Y_jys--stp;
96     if (Y_Val>3000 && pos2<teta2max) Y_jys++stp;
97
98     // İvme verisinden, X ve Y eğim açılarının trigonometrik olarak belirlenmesi.
99     accAngleX = atan(IMU.readFloatAccelX() / sqrt(pow(IMU.readFloatAccelY(), 2) +
100     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
101     accAngleY = atan(-1 * IMU.readFloatAccelY() / sqrt(pow(IMU.readFloatAccelX(), 2) +
102     pow(IMU.readFloatAccelZ(), 2)+0.001)) * 180 / PI;
103
104
105     // Motor dönüş açıları:
106     pos1=1500+int(accAngleX*3000./180.)+X_jys;
107     pos2=1500+int(accAngleY*3000./180.)+Y_jys;
108
109     // Motorlara dönüş komutlarının verilmesi.
110     myservo1. writeMicroseconds(pos1);
111     myservo2. writeMicroseconds(pos2);
112
113     delay(dly); // bu gecikme işlemi ile motorların dönüş hızları tayin edilir.
114 }

```

Resim 41 Kumanda Kolu ve IMU Sensörü ile Servo Motor Kontrolü Kodu Microseconds II. Kısım – Loop

Uygula: Donanım ve Yazılımın Denenmesi (Test Edilmesi):

Program Deneyap Kart / Deneyap Kart 1A'ya 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 Kart / Deneyap Kart 1A 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

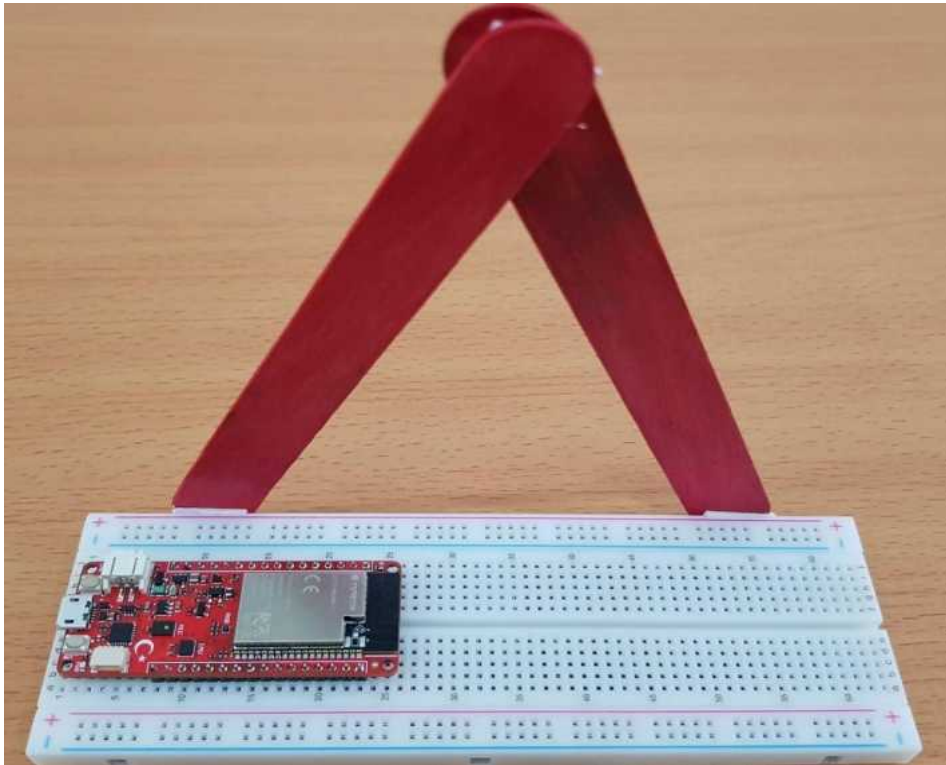
Bu uygulamada ışık sensörlerinden alınan ortam ışığı değerlerine göre 2 motorlu bir sistemin hareket kontrolü yapılacaktır. Böyle bir sistem, çeşitli amaçlar için kullanılabilir. Örneğin:

- 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.

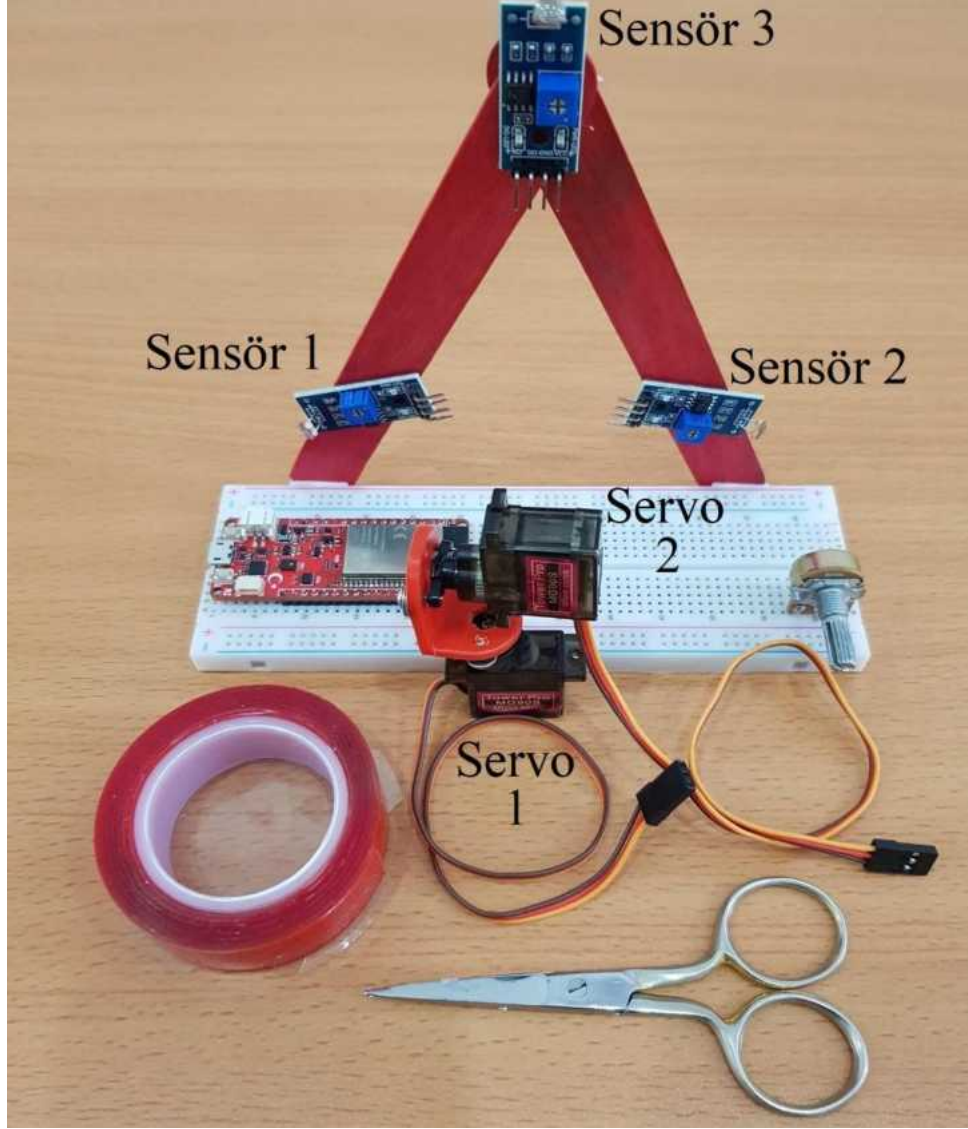
Tasarım ve Üretim Adımları:

Adım 1) Deneyap Kart / Deneyap Kart 1A Breadboard üzerine yerleştirilir ve 2 adet dil çubuğu da yine Resim 42’de gösterildiği gibi bir üçgen oluşturacak şekilde birbirlerinin uçlarına yapıştırılır. Diğer uçları da breadboard’un bir kenarına yapıştırılır. Yapıştırma işlemi için ağaç tutkalı, silikon tabanca veya çift taraflı yapışkan bant kullanılabilir. Bu yapıştırma işlemi yöntemi, eğitmenin ve öğrencilerin tercihinə 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.



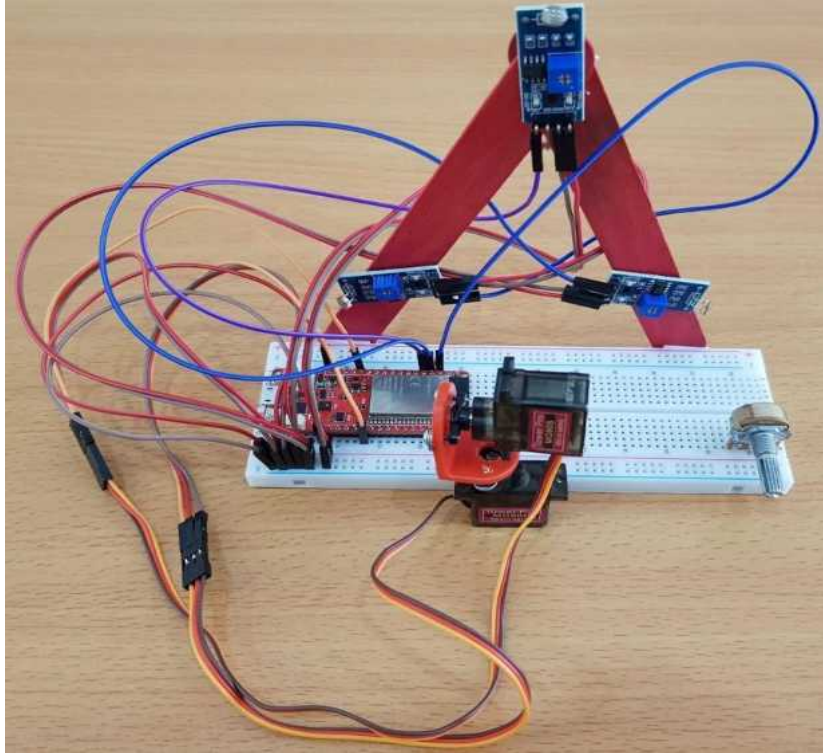
Resim 42 Deneyap Kart / Deneyap Kart 1A ve Dil Çubuklarının Breadboard Üzerine Yerleştirilmesi

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 43'de 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.

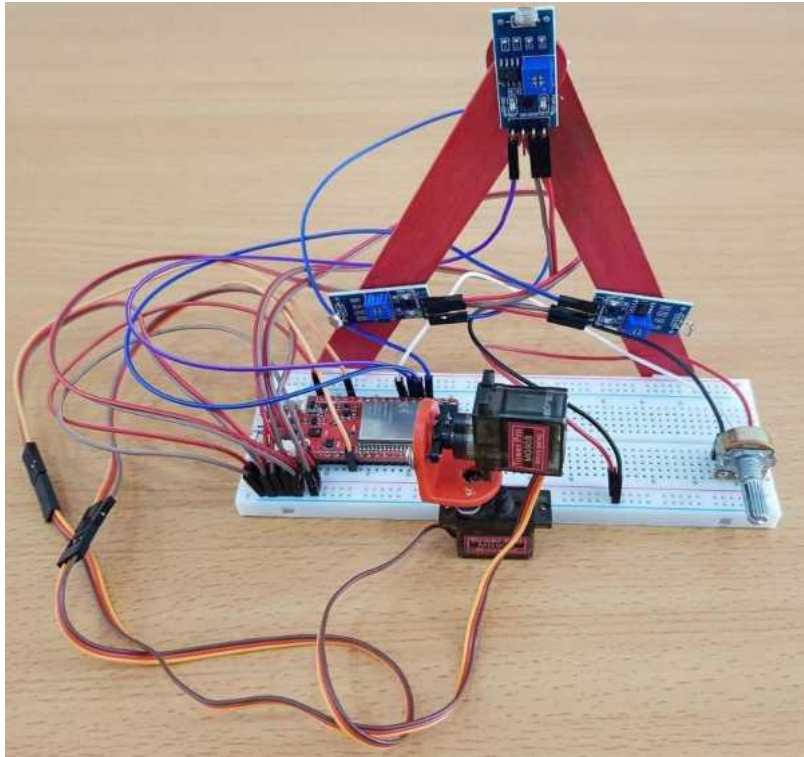


Resim 43 Çift Motor Düzeneği ve Işık Sensörlerinin Sisteme Yerleştirilmesi

44



Resim 46 Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü Uygulaması Işık Sensörü Bağlantıları



Resim 47 Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü Uygulaması Potansiyometre Bağlantısı

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

Adım 4: Deneyap Kart / Deneyap Kart 1A bilgisayara bağlanır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

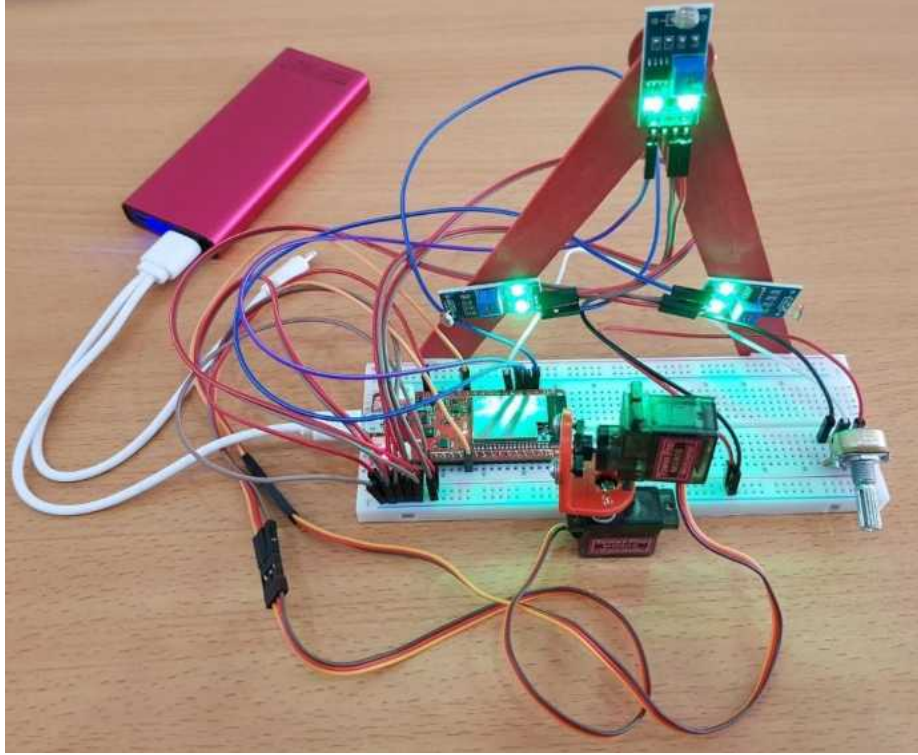
```

1 #include <Deneyup_Servo.h>
2
3 // Dijital servo pin bağlantıları:
4 #define SERVOPIN1 D9
5 #define SERVOPIN2 D14
6
7 // Işık sensörleri analog bağlantıları:
8 #define L1_pin A0
9 #define L2_pin A1
10 #define L3_pin A2
11
12 // Potansiyometre analog bağlantısı:
13 #define Pot_pin A3
14
15 // Motor dönüş açısı alt ve üst limitleri:
16 int totalmin=0;
17 int totalmax=180;
18 int tota2min=0;
19 int tota2max=180;
20
21 //Accelerometre X ve Y eksenleri etrafı açısı değişkenleri
22 int accAngleX=0;
23 int accAngleY=0;
24 int conavali=0;
25
26 int coef; // Potansiyometreden elde edilen katsayı değişkeni.
27 int valin1, valin2, valin3; // Işık sensörlerinden hesaplanan değerlerin atanacağı değişkenler.
28
29 //Servo motor dönüş açısı değişkenleri
30 int pos1 = 0;
31 int pos2 = 0;
32 int cih1=0;
33 int cih2=0;
34 int cih3=0;
35
36 // İvme ölçer verisinde oluşan noise'ı (gürültüyü) azaltmak (hatta yok etmek) için "moving average"
37 // algoritmasının uygulanmasında kullanılan değişkenler.
38 int i=0;
39 int n=10;
40 int32_t tot1=0.;
41 int32_t tot2=0.;
42 int32_t tot3=0.;
43
44 // Servo motorların tanımlanması.
45 Servo myservo1, myservo2;
46
47 void setup() {
48     // Servo motorların hangi dijital pinlere bağlı olduklarının belirtilmesi.
49     myservo1.attach(SERVOPIN1);
50     myservo2.attach(SERVOPIN2,1);
51 }
52
53 void loop() {
54     coef = analogRead(Pot_pin); //Potansiyometre çıkışının bir katsayı değişkenine atanması.
55
56     // Işık sensörlerinden hesaplanan değerlerin atandığı değişkenler.
57     valin1 = 4095-analogRead(L1_pin);
58     valin2 = 4095-analogRead(L2_pin);
59     valin3 = 4095-analogRead(L3_pin);
60
61     // Işık sensörü verisindeki noise'ı (gürültüyü) azaltmak için "moving average" (hareketli ortalama) yöntemi:
62     if (i<n) {
63         i++;
64     }
65     tot1=int(((i-1)*tot1+valin1)/i);
66     tot2=int(((i-1)*tot2+valin2)/i);
67     tot3=int(((i-1)*tot3+valin3)/i);
68
69     // Işık sensörlerinden hesaplanan verilerin kullanılması ile hesaplanan servo motor dönüş açıları:
70     pos1=90+(tot2-tot1)*((2048-coef)/2048.);
71     pos2=120+(0.5*(tot1+tot2)-tot3)*((2048-coef)/2048.);
72
73     // Motorlara dönüş komutlarının verilmesi.
74     myservo1.write(pos1);
75     myservo2.write(pos2);
76
77     delay(5); // Genel döngü içerisinde 5 milisaniye gecikme.
78 }

```

Resim 48 44 Işık Sensörleri ile 2 Servo Motorlu Sistemin Pan-Tilt Hareket Kontrolü Kodu

Adım 5: Program Deneyap Kart / Deneyap Kart 1A'ya yüklendiğinde motorların ışık sensörleri ve potansiyometre aktivitelerine göre davranış gösterdiği görülecektir. Resim 49'da 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 49 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 örnekteki 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 örnekteki 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.

4. 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.

3. Bölüm: Üç Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A'nın temel özelliklerini bilir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinde servo motor kullanımı bilgisine sahiptir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A 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 Kart / Deneyap Kart 1A 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 eklemlili) ve uç kısmında kısaç 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:

Malzeme Listesi	
Deneyap Kart / Deneyap Kart 1A	1 adet
Breadboard	1 adet
Jumper Kablo	30+ Adet
Mikro USB Bağlantı Kablosu	1 adet
Harici 5V USB Batarya / 9V Pil / 2li 18650 Pil,	1 adet
Abeslang	Yeterli Kadar
LDR sensörü	3 Adet
Potansiyometre	1 Adet
Kumanda kolu (Joystick)	2 Adet

Servo Motor	4 Adet
Çift taraflı silikon bant/silikon tabancası ve makas	1 Adet
El Feneri veya Telefon Feneri	1 Adet
3B Yazıcı ve Filament	Yeteri Kadar
Tutakl, Yapıştırıcı, Silikon, Çift Taraflı Bant	Yeteri Kadar
Esnek Plastik Levha	Yeteri Kadar
M2 veya M3 rondela (pul)	15 Adet

Haftanın İşlenişi:

Gözle: Bundan önceki haftalarda uygulaması yapılan modüllerin Deneyap Kart / Deneyap Kart 1A 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 Kart / Deneyap Kart 1A 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

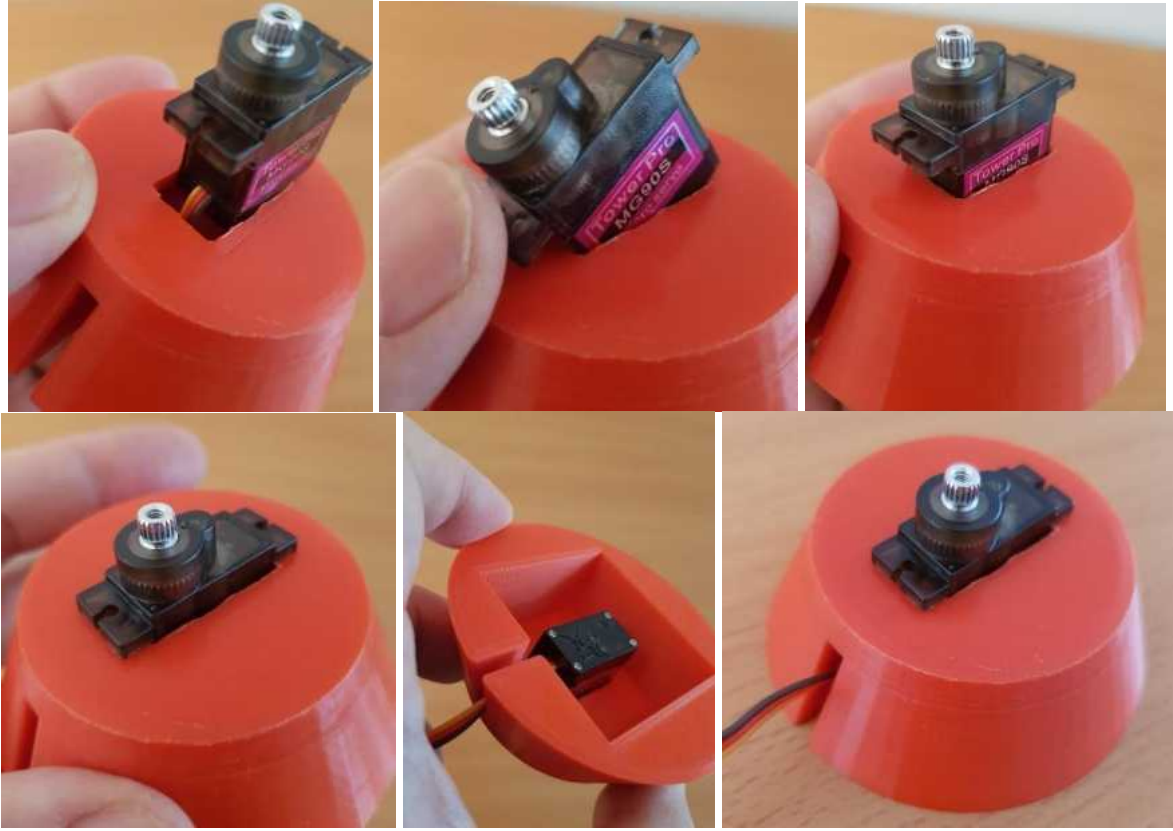
Dersin başlangıcında, eğitmen öğrencilerin 3 serbestlik dereceli robotlar ile ilgili tartışmalarını, ilk dersin konularını hatırlamalarını sağlamalıdır. Üç serbestlik dereceli robotların yapısı ve kullanım alanları üzerine öğrencilerin beyin jimnastiği yapmaları, onların derse ilgi ve katılımlarını artıracaktır. İnternet 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 ulaşılabilir.

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

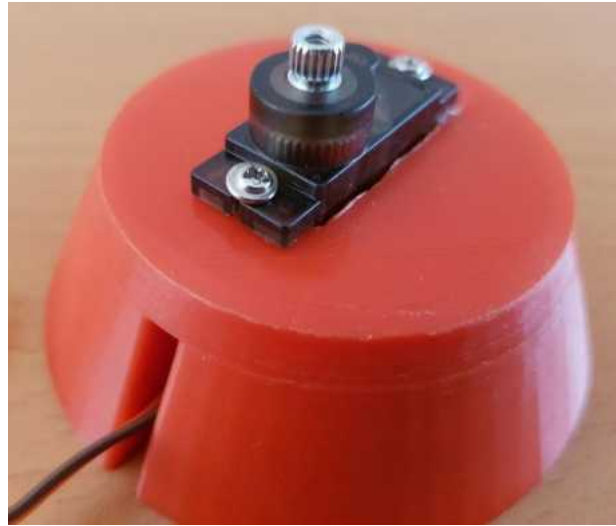
1.2. Uygula: Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü

Uygulama Adımları:

Adım 1: 1 numaralı servo motor 3B yazıcıdan basılmış taban parçasına Resim 50'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 51). Eğer 3 boyutlu yazıcı konusunda tecrübesizlik nedeniyle bu taban parçasının üretilmesinde zorlukla karşılaşırsa, buna alternatif olarak, elde halihazırda mevcut plastik veya pleksiglas malzemeden yapılmış dikdörtgen bir taban parçası üzerine taban motorunun yapıştırılması yoluna da gidilebilir.

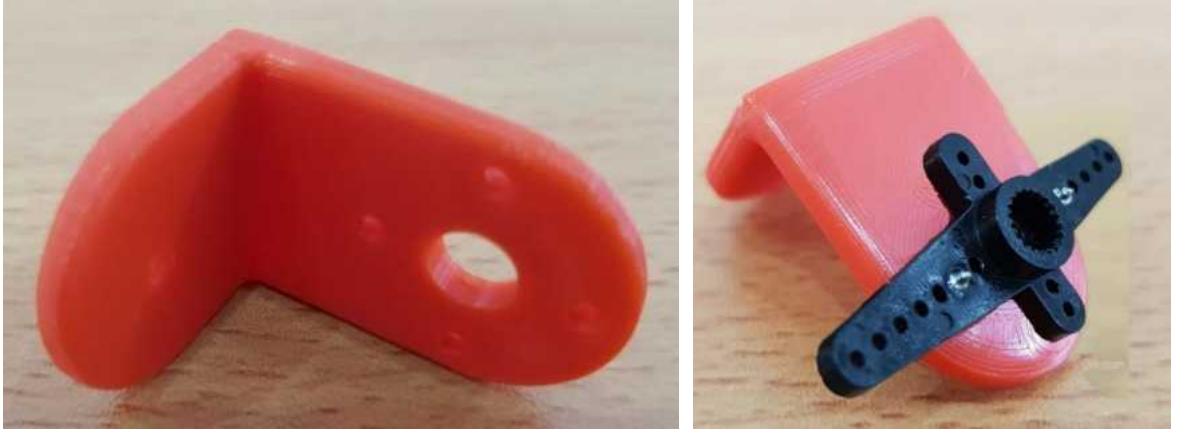


Resim 50 Servo Motorun 3B Yazıcıdan Basılmış Taban Parçasına Yerleştirilmesi



Resim 51 Taban Parçaya Vidalanmış Servo Motor

Adım 2: İki numaralı servo motoru bağlamak için 3B yazıcıda basılmış bir L-profil kullanılır (Resim 52). Burada servo motor plastiklerinden T şeklinde olanı kullanılır. T şeklinde plastik bulunamazsa diğerleri de kullanılabilir.



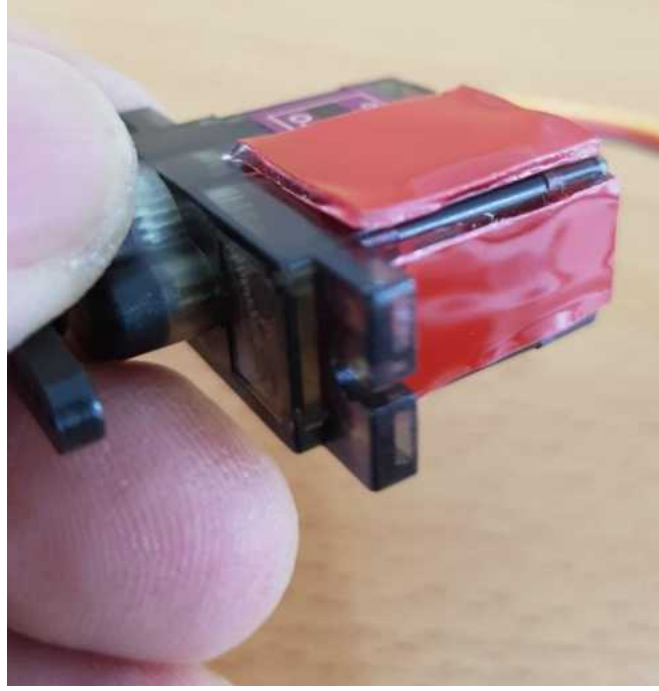
Resim 52 3B-Yazıcıda Basılan L-Profil ve Üzerine Vidalanmış Servo Motor Plastik Parçası

Adım 3: Hazırlanan L-profil, taban motorunun (Motor 1) miline Resim 53'te gösterildiği gibi yerleştirilir.



Resim 53 Hazırlanan L-Profil Parçasının Taban Motoru Miline Yerleştirilmesi

Adım 4: İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılır (Resim 54). Burada, önceki bölümlerdekine benzer olarak, silikon tabanca ile sıkılan yapıştırıcı silikon malzemede kullanılabilir. Bu, eğitmen ve öğrencinin tercihinine bırakılmıştır.



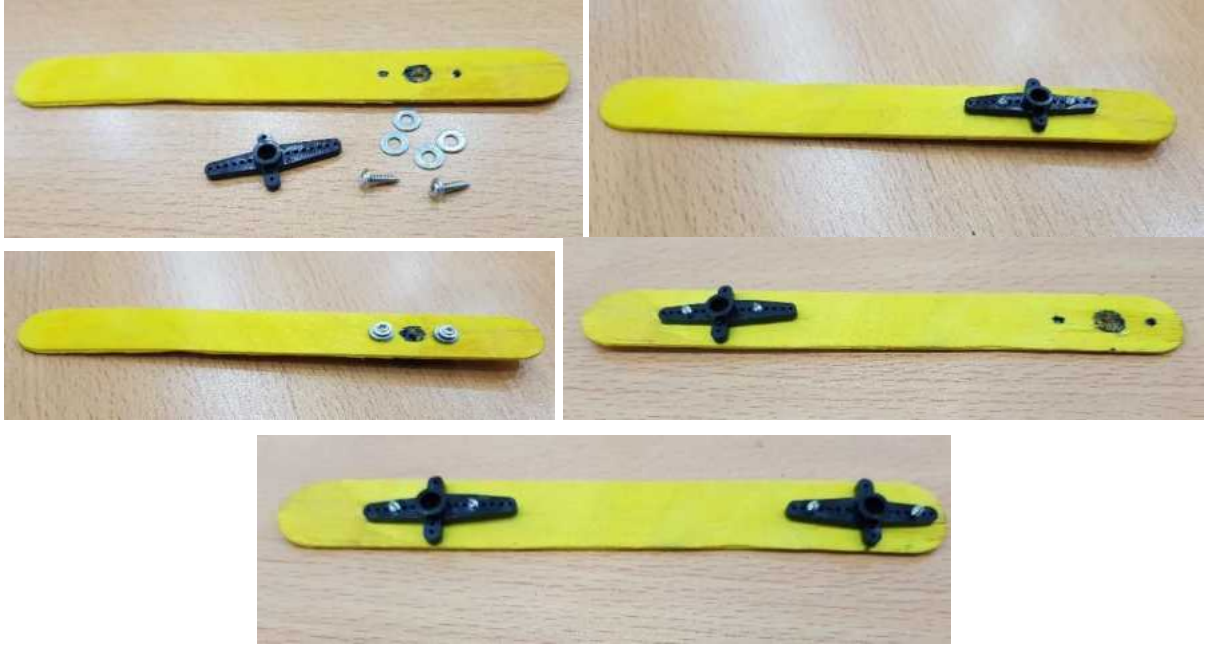
Resim 54 İki Numaralı Servo Motorun İki Ayrı Yüzeyine Çift Taraflı Silikon Bant Yapıştırılması

Adım 5: İki numaralı servo motora yapıştırılmış çift taraflı silikon bantın kırmızı naylonları çıkartılır ve servo motor L-profil parçanın üzerine yapıştırılır (Resim 55).



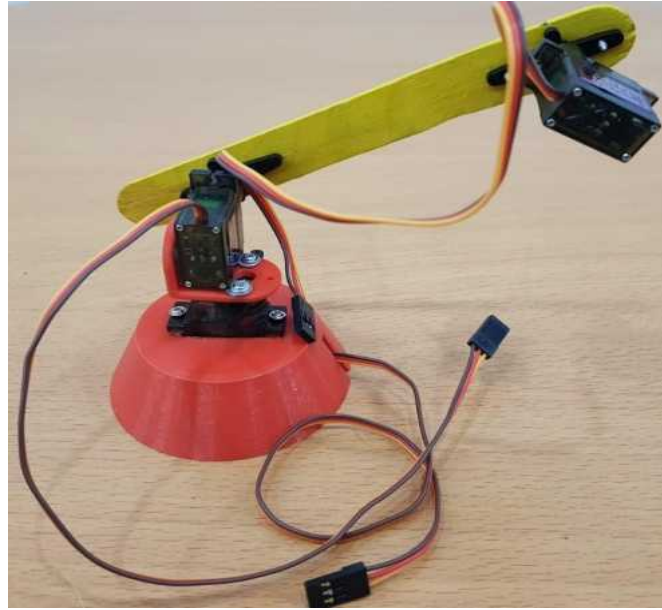
Resim 55 İki Numaralı Servo Motorun L-Profil Parça Üzerine Konumlandırılması

Adım 6: Bir adet abeslang Resim 56'dadki 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. Öğitmenler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, öğitmenlere ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.



Resim 56 Abeslanglara Servo Motor T-Plastik Parçalarının Takılması

Adım 7: Hazırlanmış olan abeslang Resim 57’de gösterildiği gibi 2 numaralı servo motora takılır. Burada hareket açısı 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.



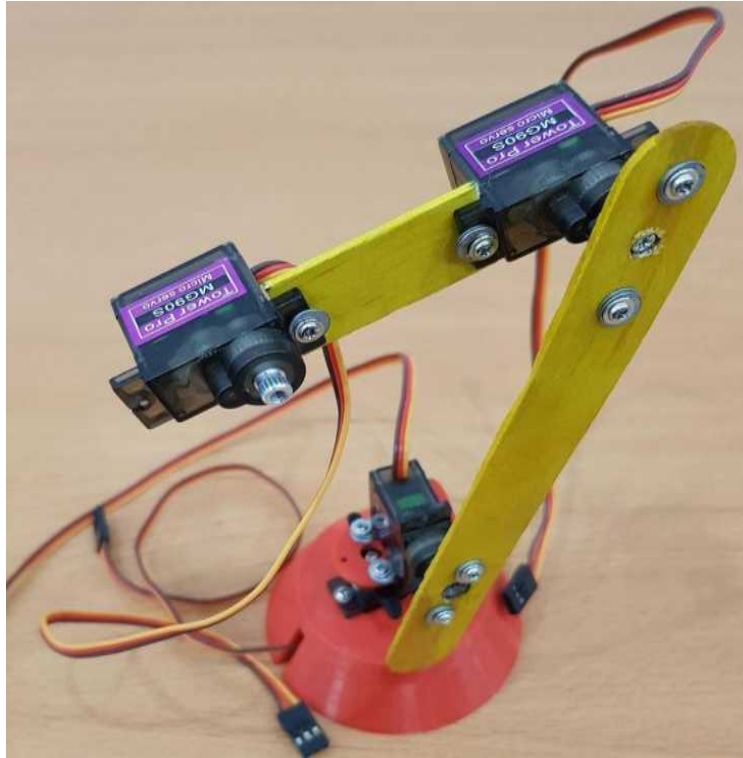
Resim 57 Abeslangın 2 Numaralı Servo Motora ve 3 Numaralı Motorun Abeslanga Monte Edilmesi

Adım 8: Başka bir abeslangtan kısa bir parça kesilir (Resim 58). Eğitimciler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, eğitimciler ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.



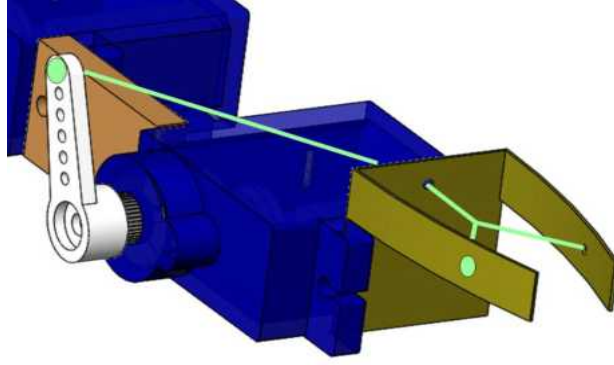
Resim 58 Kısa Bir Parçanın Abeslangtan Kesilmesi ve Hazırlanması

Adım 9: Hazırlanmış olan kısa parça ile 3 ve 4 numaralı servo motorlar birbirlerine bağlanır (Resim 59).

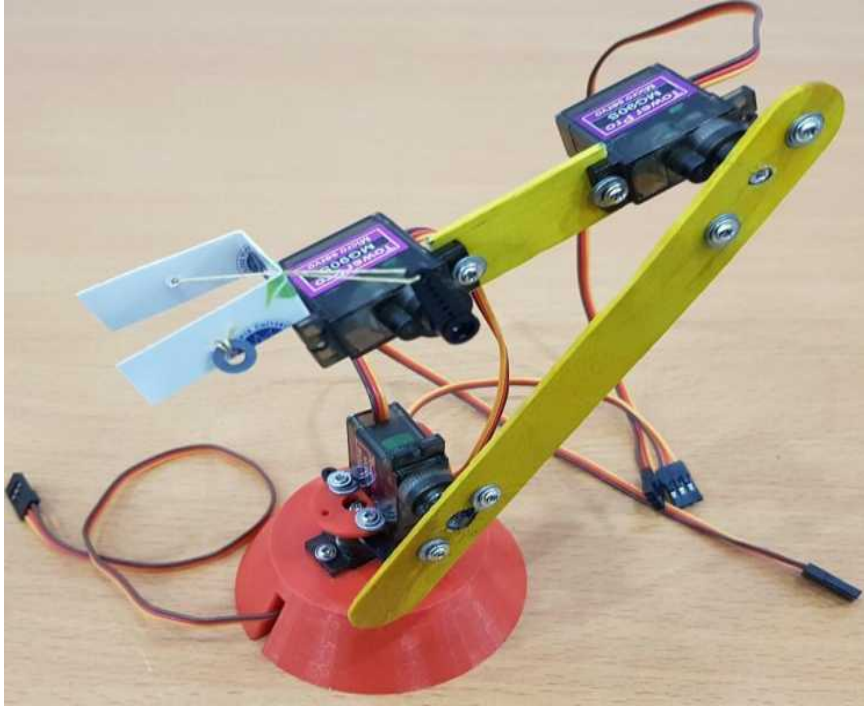


Resim 59 Kısa Parça ile 3 ve 4 Numaralı Servo Motorların Birbirlerine Bağlanması

Adım 10: Tutucu (Gripper) tasarımında (Resim 60) 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 60'ta gösterildiği gibi servo motora yapıştırılır. Yine resimde gösterildiği gibi naylon ip de tutucu ile motor arasında bağlanır. Motorun dönme yönüne göre tutucu elinin aç/kapa hareketini yaptığı gözlemlenir. Bu motorlu tutucu mekanizma Resim 61'de gösterildiği gibi robot manipülatöre takılır.

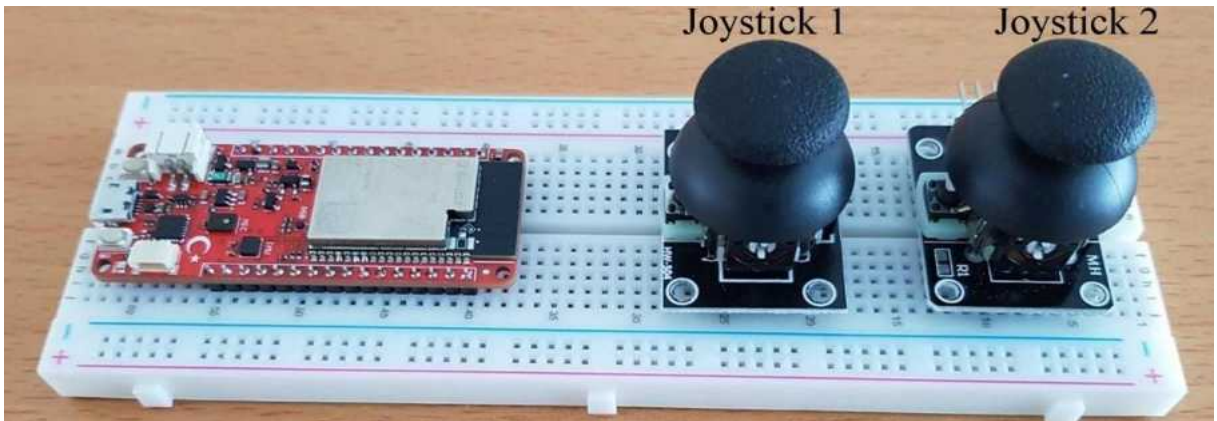


Resim 60 Tutucu tasarımı



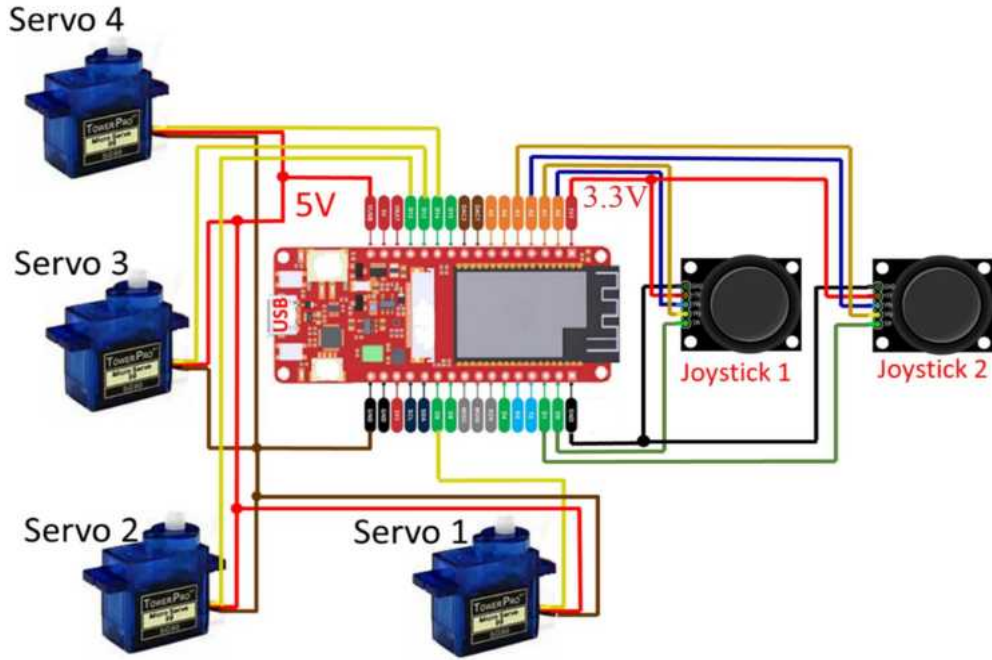
Resim 61 Tutucusu Yerleştirilmiş Robot Manipulator

Adım 11: Deneyap Kart / Deneyap Kart 1A breadboard üzerine yerleştirilir. Kumanda kollarının alt tabanlarına çift taraflı silikon bant yerleştirilir ve breadboard üzerine Resim 62’de gösterildiği gibi yapıştırılır. Silikon bant lehim çıkıntıları nedeniyle 2 veya 3 olarak uygulanabilir.

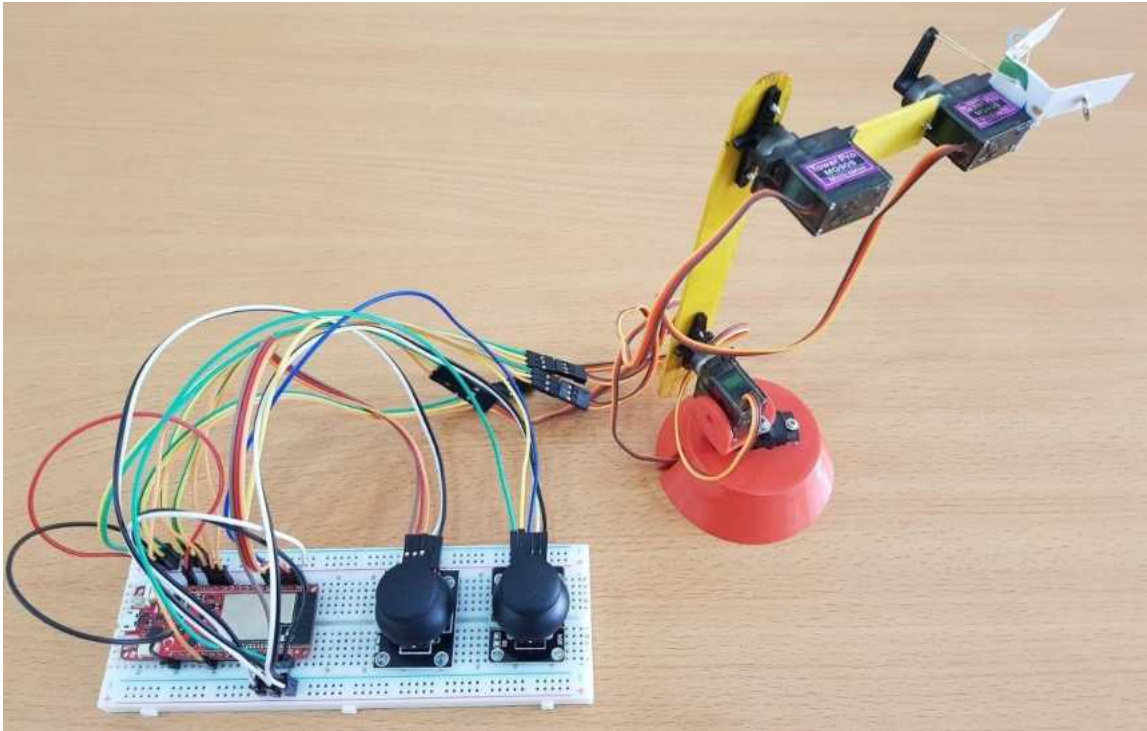


Resim 62 Breadboard Üzerine Konumlandırılmış Kumanda Kolları

Adım 12: Kumanda kollarının ve servo motorların Deneyap Kart / Deneyap Kart 1A'ya bağlantıları Resim 5.14'te verilmiştir.



Resim 63 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Devre Şeması



Resim 64 Üç Serbestlik Dereceli Robot Manipulatörün Bağlantılarının Yapılmış Genel Fotoğrafı

Adım 13: Deneyap Kart / Deneyap Kart 1A'nın bilgisayarla bağlantısı yapılır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```

1 #include <Deneyap_Servo.h>
2
3 // Dijital kumanda kollarının button pin bağlantıları:
4 #define SW1_pin D0
5 #define SW2_pin D1
6
7 // kumanda kollarının eksen analog bağlantı pinleri
8 #define X1_pin A0
9 #define Y1_pin A1
10 #define X2_pin A2
11 #define Y2_pin A3
12
13 // Dijital servo pin bağlantıları:
14 #define SERVOPIN1 D9
15 #define SERVOPIN2 D12
16 #define SERVOPIN3 D13
17 #define SERVOPIN4 D14
18
19 // Motor dönüş açısı alt ve üst limitleri:
20 // Aşağıdaki teta değerleri 0-3000 arasında kalibre edilmiştir. 3000 değeri 180 dereceye karşılık gelmektedir.
21 int teta1min=0;
22 int teta1max=3000;
23 int teta2min=0;
24 int teta2max=3000;
25 int teta3min=0;
26 int teta3max=3000;
27 int teta4min=1500;
28 int teta4max=2000;
29
30 int valin1, valin2, valin3, valin4;
31 int valcalc1=1500; //Başlangıç açısı değerleri
32 int valcalc2=1500;
33 int valcalc3=1500;
34 int valcalc4=2000;
35 int jystp=20;
36 // kumanda kollarıyla motor hareketleri sağlandığında her bir döngüde istenen açı değeri
37 (bu yazılımda 3000 değerinin 360 dereceye karşılık geldiğini unutmayın. */
38
39 bool playmode=false; // Robotun tekrarlama (oynama) modu değişkeni. Değer false olduğunda, öğretim modunda demektir.
40 int ButtonMode=0; // Buton modu değişkeninin tanımlanması. Sıfır olduğunda, öğretim modunda demektir.
41 unsigned long CurrentMillis = millis();
42 // millis() fonksiyonu, şu anki saati milisecond cinsinden vermektedir. Zamanlayıcı olarak bu fonksiyonu kullanmaktayız.
43 // Herhangi bir anda, o anki zamanı bir long değişkenine atayabiliriz.
44 long PreMillis=0; // Bir evvelki zaman (milisaniye cinsinden)
45 int NumMemory=0; // Hafıza sayacı (Maksimum 50 olabilir.
46 Deneyap kartının kapasitesine bağlı olarak bu sınır 50'den yukarıda olabilir.) */
47 int PreNumMemory=0; // Bir evvelki moddaki hafıza değeri.
48 int Step=-1; // Öğretim modunda, servo pozisyonlarını hafızaya alırken kullanılan bir değişken.
49 // Servo pozisyonlarının hafızaya alındığı 50'lik seriye sahip değişkenler.
50 int joint1[50];
51 int joint2[50];
52 int joint3[50];
53 int joint4[50];
54 int maxarray=49; //Hafıza hatası vermeyi önlemek için 49 değerine sahip bir limit değişkeni.
55 int prehiz=10; // Bir önceki hız değerinin atanacağı değişken tanımlaması.
56 int normalhiz=10; // Normal hız değişkeni
57 // hiz=1 çok yavaş, 5 yavaş, 10 normal, 15 hızlı, 20+ çok hızlı
58 int hiz=10; // Şu anki hız değişkeni
59 int prepos1, prepos2, prepos3, prepos4; // bir evvelki motor konumları.
60 int pos11, pos21, pos31, pos41; // Motor konumları ile ilgili ikincil (yardımcı) değişkenler.
61
62 Servo myservo1, myservo2, myservo3, myservo4; // Servo motorların tanımlanması.
63
64 void setup()
65 {
66     myservo1.attach(SERVOPIN1);
67     //Eğer <ESP32Servo.h> kütüphanesi kullanırsanız, bu ve aşağıdaki 3 satırda parantez içerisindeki
68     //2'inci parametreyi silin, tek parametre yeterlidir.
69     myservo2.attach(SERVOPIN2,1);
70     myservo3.attach(SERVOPIN3,2);
71     myservo4.attach(SERVOPIN4,3);
72

```

Resim 65 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Kodu – I

```

73 // Joystik butonlarının pinlerinin tanımlanması (Normalde HIGH).
74 pinMode(SW1_pin, INPUT_PULLUP);
75 pinMode(SW2_pin, INPUT_PULLUP);
76 // kumanda kolu pinlerinin giriş olarak tanımlanması
77 pinMode(X1_pin, INPUT);
78 pinMode(Y1_pin, INPUT);
79 pinMode(X2_pin, INPUT);
80 pinMode(Y2_pin, INPUT);
81 // Joystik butonlarının pinlerinin değerleri normalde HIGH' a atandı.
82 digitalWrite(SW1_pin, HIGH);
83 digitalWrite(SW2_pin, HIGH);
84 // Tercihe bağlı olarak, Seri port'a "Ready" yazdırılması (baudrate= 9600)
85 Serial.begin(115200);
86 Serial.println("Ready.");
87 }
88
89 void loop() {
90   CurrentMilliSec = millis(); // Şu anki zaman değeri (ms)
91   Button(); // Kendi yazdığımız Button fonksiyonunun çağırılması.
92   if (playmode==false && Step==1){ // Öğretme modunda iken,
93     ReadAnalog(); // Analog kanalların okunması için kendi yazdığımız ReadAnalog fonksiyonunun çağırılması.
94     // Servo motorlara dönüş komutlarının verilmesi:
95     myservo1.writeMicroseconds(valcalc1);
96     myservo2.writeMicroseconds(valcalc2);
97     myservo3.writeMicroseconds(valcalc3);
98     myservo4.writeMicroseconds(valcalc4);
99     delay(int(300./Hiz)); // Hız kontrolü için genel döngü içerisindeki gecikme ayarı.
100  }
101  else{
102    playit(); //tekrarlama (veya oynama) modunda iken kendi yazdığımız Play fonksiyonunun çağırılması.
103  }
104 }
105
106 /* Kumanda kollarının ürettiği analog sinyallerin okunduğu ve motor açı değerlerinin bu sinyallere
107 göre değiştirildiği fonksiyon aşağıda verilmiştir. */
108 void ReadAnalog() {
109   // Analog pinlere giren kumanda kolu değerlerinin okunması.
110   valin1 = analogRead(Y1_pin);
111   valin2 = analogRead(X1_pin);
112   valin3 = analogRead(X2_pin);
113   valin4 = analogRead(Y2_pin);
114
115   // Belli sınırlar dahilinde, Joystik değerlerine göre servo motor dönüş açılarının tayini.
116   if (valin1<1000 && valcalc1>teta1min) valcalc1-=jystp;
117   if (valin1>3000 && valcalc1<teta1max) valcalc1+=jystp;
118   if (valin2<1000 && valcalc2>teta2min) valcalc2+=jystp;
119   if (valin2>3000 && valcalc2<teta2max) valcalc2-=jystp;
120   if (valin3<1000 && valcalc3>teta3min) valcalc3-=jystp;
121   if (valin3>3000 && valcalc3<teta3max) valcalc3+=jystp;
122   if (valin4<1000 && valcalc4>teta4min) valcalc4-=jystp;
123   if (valin4>3000 && valcalc4<teta4max) valcalc4+=jystp;
124 }
125
126 /*Buton fonksiyonu oldukça kapsamlıdır. Burada 2 adet buton ile robot hareket modları tayin edilmektedir.
127 Öğretme modunda iken kumanda kolu 1'in butona tek tıklama ile mevcut motor açılarını hafızaya alma işlemi
128 yapılabilmekte, öğretme modundan tekrarlama moduna veya tersine yine Kumanda kolu 1'in butonuna çift tıklayarak
129 geçilebilmektedir. Bu uygulamada da motorlara hızlı, normal ve yavaş olmak üzere 3 ayrı seviye tayin edilmiştir.
130 Bu hız seviyeleri arasındaki geçişler Kumanda kolu 2'nin butonuna basılarak gerçekleştirilmektedir.
131 Belirtilen buton aktivitelerini gerçekleştiren kodlar aşağıda verilmiştir. */
132 void Button() { // Butona basılma durumu
133   if (playmode==false){ // Eğer öğrenme modunda iken,
134     if (digitalRead(SW1_pin) == 0){
135       delay(100);
136       if (digitalRead(SW1_pin) == 1) { // butona tıklanmış ise,
137         if (ButtonMode == 0) {
138           ButtonMode = 1; // İlk buton tek tık tespiti.
139           PreMilliSec = CurrentMilliSec;
140         } else if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec < 500)) {
141           ButtonMode = 2; // İlk buton çift tık tespiti.
142         }
143       }
144     }
145   }

```

Resim 66 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Kodu – II

```

145 v if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec > 600)) {
146     saveit(); // İlk buton tek tık tespit edildiğinde, motor konumları hafızaya alınır.
147     NumMemory += 1;
148     if (NumMemory > maxarray) NumMemory = maxarray;
149     ButtonMode = 0;
150     Serial.print("Saved: "); Serial.println(NumMemory);
151 v } else if (ButtonMode == 2 && (CurrentMilliSec - PreMilliSec > 600)) {
152 v     if ((NumMemory > 1 || PreNumMemory > 1) && playmode == false) {
153         Serial.println("Playmode...");
154         playmode = true; // Çift tık tespit edildiğinde tekrarlama (oynama) moduna geçilir.
155         ButtonMode = 0;
156         if (NumMemory < 2) NumMemory = PreNumMemory; else PreNumMemory = NumMemory;
157         // Eğer hafızaya herhangi bir konum alınmamış olup ta çift tık yapılırsa,
158         // bir önceki oynatılmış konum bilgilerinin tekrar kullanılması için devreye alınması.
159         // Motor açılı konumlarının hafızaya alınması:
160         joint1[0] = pos11;
161         joint2[0] = pos21;
162         joint3[0] = pos31;
163         joint4[0] = pos41;
164
165         Step = NumMemory - 1;
166         ReadAnalog();
167         prepos1 = valcalcl1;
168         prepos2 = valcalcl2;
169         prepos3 = valcalcl3;
170         prepos4 = valcalcl4;
171     }
172 }
173 v if (ButtonMode < 3 && CurrentMilliSec - PreMilliSec > 1000) {
174     ButtonMode = 0; // Tekrar öğretme moduna geçiş.
175 }
176 v } else if (ButtonMode < 3 && digitalRead(SW1_pin) == 0) {
177     ButtonMode = 3;
178     Serial.println("Stopping."); // Tekrarlamayı (Oynamayı) durdur.
179     ReadAnalog();
180     NumMemory += 1;
181     // Durulan konumun hafızanın en son basamağına kaydedilmesi.
182     joint1[NumMemory - 1] = valcalcl1;
183     joint2[NumMemory - 1] = valcalcl2;
184     joint3[NumMemory - 1] = valcalcl3;
185     joint4[NumMemory - 1] = valcalcl4;
186 }
187 // İkinci butona her tık yapıldığında robot hareket hızı değiştirilir.
188 // Normal, yavaş, hızlı olmak üzere 3 ayrı hız için, aşağıdaki kod yazılmıştır.
189 // Diğer detaylar, önceki bölümlerdeki kodlarda anlatıldığı gibidir.
190 v if (digitalRead(SW2_pin) == 0) {
191     if (prehiz == hiz) {
192 v         if (hiz == normalhiz) {
193             hiz = 2 * normalhiz;
194             jystp = 40;
195             Serial.println("High speed.");
196 v             if (playmode == false) {
197                 delay(300);
198             }
199 v         } else {
200 v             if (hiz == (2 * normalhiz)) {
201                 hiz = normalhiz / 2;
202                 jystp = 10;
203                 Serial.println("Low speed.");
204 v                 if (playmode == false) {
205                     delay(300);
206                 }
207 v             } else if (hiz == (normalhiz / 2)) {
208                 hiz = normalhiz;
209                 jystp = 20;
210                 Serial.println("Normal speed.");
211 v                 if (playmode == false) {
212                     delay(300);
213                 }
214             }
215         }
216     }

```

Resim 67 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Kodu – III

```

217 } else{
218     prehiz=hiz;
219 }
220 }
221
222 // Mevcut servo motor açılarının hafızaya kaydedildiği fonksiyon aşağıda verilmiştir.
223 // Detaylar, önceki bölümlerde verilen kodlardaki gibidir.
224 void saveit() {
225     joint1[NumMemory] = valcalc1;
226     joint2[NumMemory] = valcalc2;
227     joint3[NumMemory] = valcalc3;
228     joint4[NumMemory] = valcalc4;
229
230     prepos1=valcalc1;
231     prepos2=valcalc2;
232     prepos3=valcalc3;
233     prepos4=valcalc4;
234
235     if (NumMemory==1){
236         pos1=joint1[0];
237         pos2=joint2[0];
238         pos3=joint3[0];
239         pos4=joint4[0];
240     }
241 }
242
243 //Hafızaya kaydedilen servo motor konumlarının tekrarlatıldığı fonksiyon aşağıda verilmiştir.
244 // Detaylar, önceki bölümlerde verilen kodlardaki gibidir.
245 void playit()
246 {
247     float pos1=prepos1;
248     float pos2=prepos2;
249     float pos3=prepos3;
250     float pos4=prepos4;
251
252     float dif1=abs(joint1[Step]-pos1);
253     float dif2=abs(joint2[Step]-pos2);
254     float dif3=abs(joint3[Step]-pos3);
255     float dif4=abs(joint4[Step]-pos4);
256     float maxdif=0.0;
257
258     // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için, aşağıda kullanılmak üzere,
259     // dönüş açı mesafelerinin farklara bölüştürülmesi.
260     if (dif1>maxdif) maxdif=dif1;
261     if (dif2>maxdif) maxdif=dif2;
262     if (dif3>maxdif) maxdif=dif3;
263     if (dif4>maxdif) maxdif=dif4;
264     maxdif=maxdif/hiz;
265     if (maxdif<1.0) maxdif=1.0;
266
267     float stp1=(joint1[Step]-pos1)/maxdif;
268     float stp2=(joint2[Step]-pos2)/maxdif;
269     float stp3=(joint3[Step]-pos3)/maxdif;
270     float stp4=(joint4[Step]-pos4)/maxdif;
271     for (int i=1;i<=int(maxdif);i++){
272         // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için dönüş açı mesafeleri
273         // yukarıda tanımlanmış farklara bölüştürülüp servo motorlara uygulanmıştır.
274         pos1=pos1+stp1;
275         pos2=pos2+stp2;
276         pos3=pos3+stp3;
277         pos4=pos4+stp4;
278         if (i==int(maxdif)){
279             pos1=joint1[Step];
280             pos2=joint2[Step];
281             pos3=joint3[Step];
282             pos4=joint4[Step];
283         }
284         myservo1.writeMicroseconds(int(pos1));
285         myservo2.writeMicroseconds(int(pos2));
286         myservo3.writeMicroseconds(int(pos3));
287         myservo4.writeMicroseconds(int(pos4));
288

```

Resim 68 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Kodu – IV

```

289     prepos1=pos1;
290     prepos2=pos2;
291     prepos3=pos3;
292     prepos4=pos4;
293     delay(10);
294
295     if (ButtonMode<3) Button();
296     else delay(5);
297 }
298 v if (ButtonMode==3 && Step==NumMemory-1) { // Oynatma modunda durdurma işlemi.
299     NumMemory = 0;
300     Step=-1;
301     ButtonMode = 0;
302     playmode=false;
303     Serial.println("Stopped.");
304 v } else {
305     Step+=1;
306     if (Step>NumMemory-1) Step=0;
307     // Hafıza aşıldığında, ilk adıma dönüş.
308     // Burada istenirse "Step=NumMemory-1;" de yazılabilir ve hafızada sürekli en son adımda kalınır.
309 }
310 }

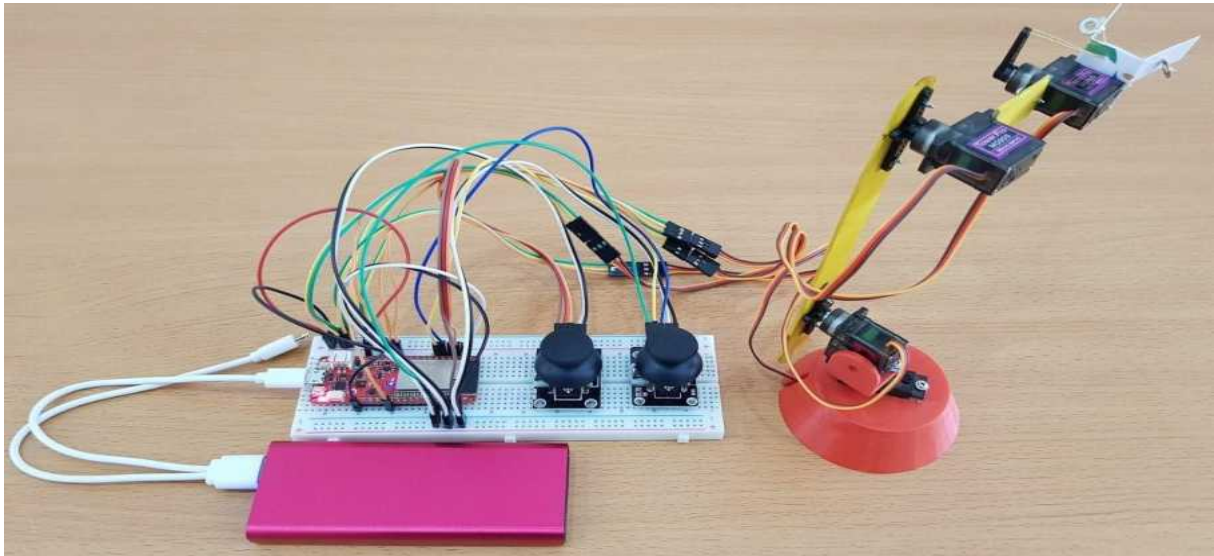
```

Resim 69 Üç Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin İki Kumanda Kolu ile Kontrolü Kodu – V

Adım 14: Program Deneyap Kart / Deneyap Kart 1A'ya yüklendikten sonra motorların kumanda kolu ile kontrolü yapılabilmekte, öğretme ve tekrarlatma modları denenebilmektedir (Resim 70).

Uyarı:

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

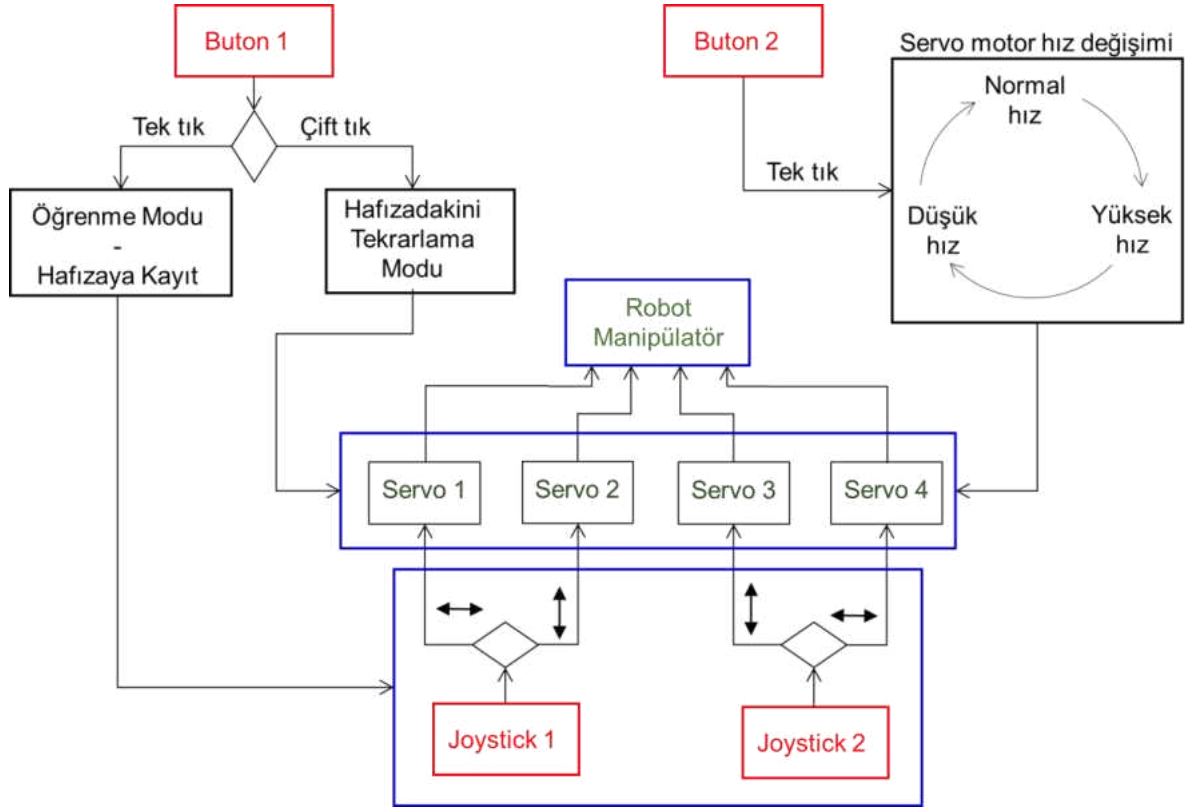


Resim 70 Tüm Sistemin Harici Bir Batarya ile Test Edilmesi

Robotun çalışma algoritması, Resim 71'ta 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.



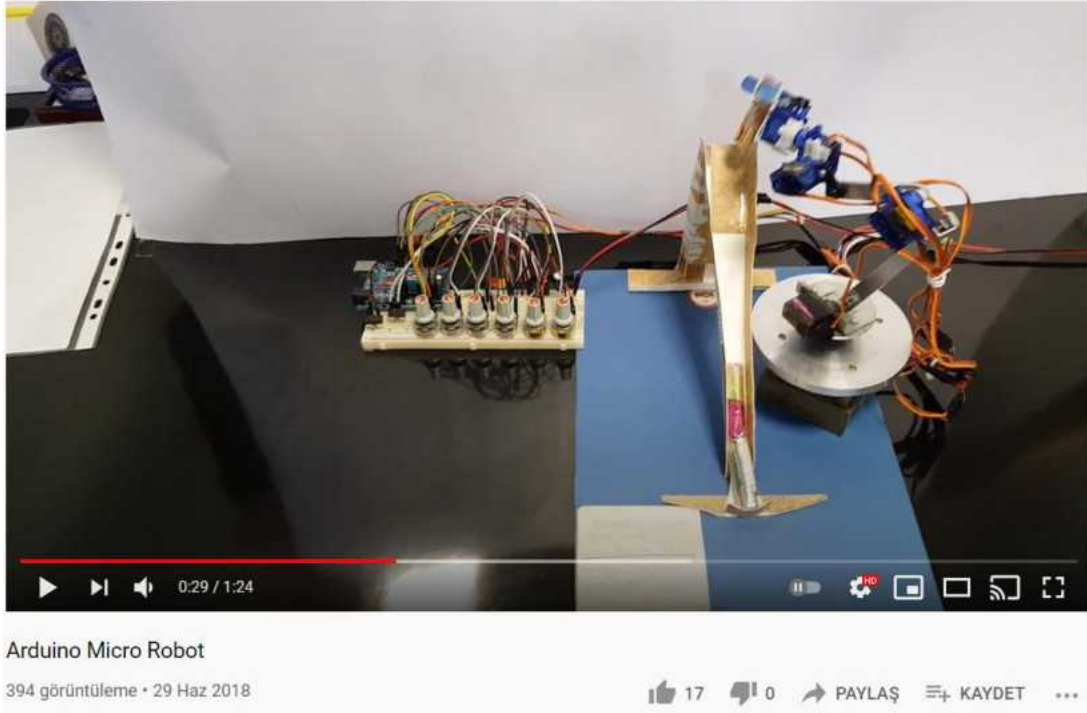
Resim 71 Robotun Çalışma Algoritmasını Gösteren Akış Diyagramı.

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 72).

Video Linki: <https://www.youtube.com/watch?v=QpYnd2ucZTY>



Resim 72 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, eğitime ve öğrencilerin hayal gücüne bırakılmıştır. Yapılan üretimler eğitmenler tarafından kontrol edilerek eksikler giderilir.

NOT:

Bu son TASARLA VE ÜRET bölümünde zaman sıkıntısı (yetiştirilememe) 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 yaptırılabilir.

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.

4. BÖLÜM: Beş Serbestlik Dereceli Robot Mekanizması

Ön Bilgi:

- Öğrenciler temel programlama bilgisine sahiptir.
- Öğrenciler temel elektrik bilgisine sahiptir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A'nın temel özelliklerini bilir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A üzerinde servo motor kullanımı bilgisine sahiptir.
- Öğrenciler Deneyap Kart / Deneyap Kart 1A 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 Kart / Deneyap Kart 1A 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 eklemlili) ve uç kısmında kısaç 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:

Malzeme Listesi	
Deneyap Kart / Deneyap Kart 1A	1 adet
Breadboard	1 adet
Jumper Kablo	40+ Adet
Mikro USB Bağlantı Kablosu	1 adet
Harici 5V USB Batarya / 9V Pil / 2li 18650 Pil,	1 adet
Abeslang	Yeterli Kadar
Potansiyometre	1 Adet
Kumanda kolu (Joystick)	3 Adet
Servo Motor	6 Adet

Çift taraflı silikon bant/silikon tabancası ve makas	1 Adet
3B Yazıcı ve Filament	Yeteri Kadar
Tutakl, Yapıştırıcı, Silikon, Çift Taraflı Bant	Yeteri Kadar
Esnek Plastik Levha	Yeteri Kadar
M2 veya M3 rondela (pul)	15 Adet

Haftanın İşlenişi:

Gözle: Bundan önceki haftalarda uygulaması yapılan sistemlerin Deneyap Kart / Deneyap Kart 1A üzerinde kullanımlarının ve temel kodlamalarının kısaca üzerinden geçilerek öğrencilere hatırlatılması.

Tasarla: Beş serbestlik dereceli robot mekanizmasının tasarlanması ve Deneyap Kart / Deneyap Kart 1A 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 Kart / Deneyap Kart 1A ile kontrolünün yapılması ve test edilmesi.

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

1. GÖZLE VE UYGULA

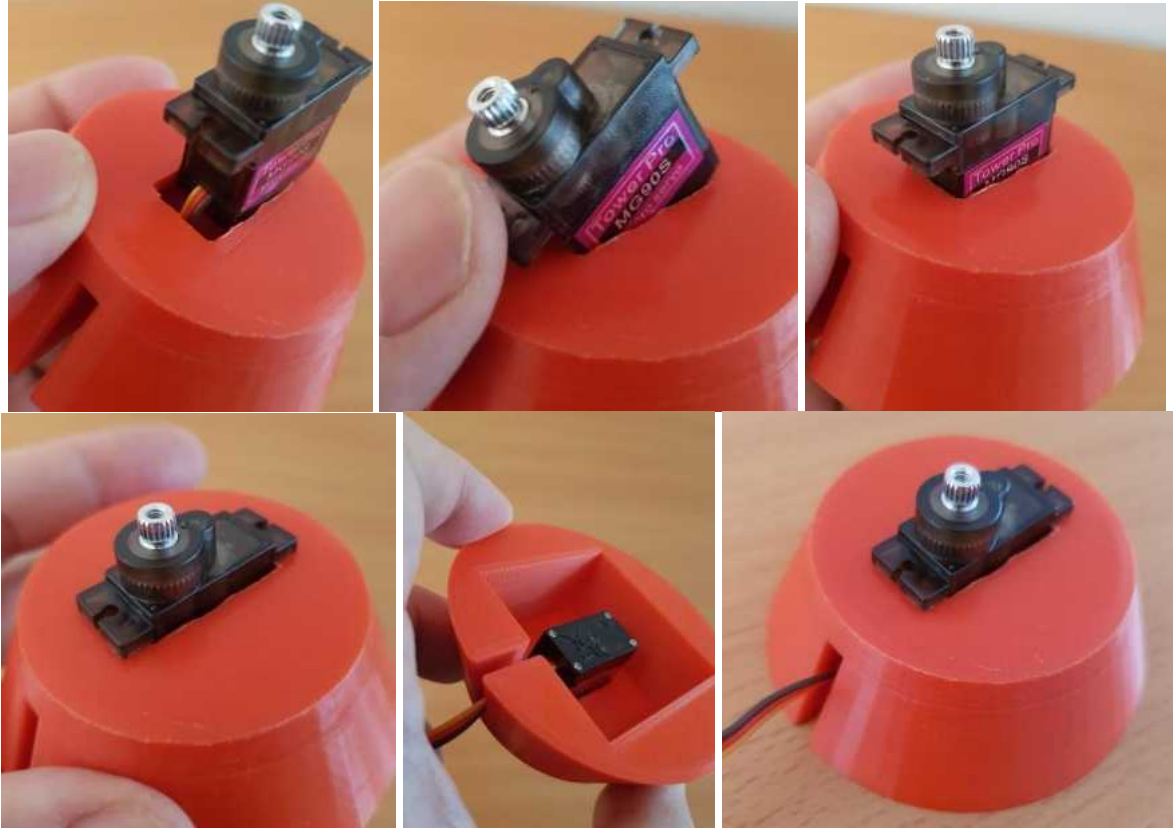
1.1. Gözle: Beş Serbestlik Dereceli Robotların Yapısı ve İşlevleri

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 73’te 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 74).

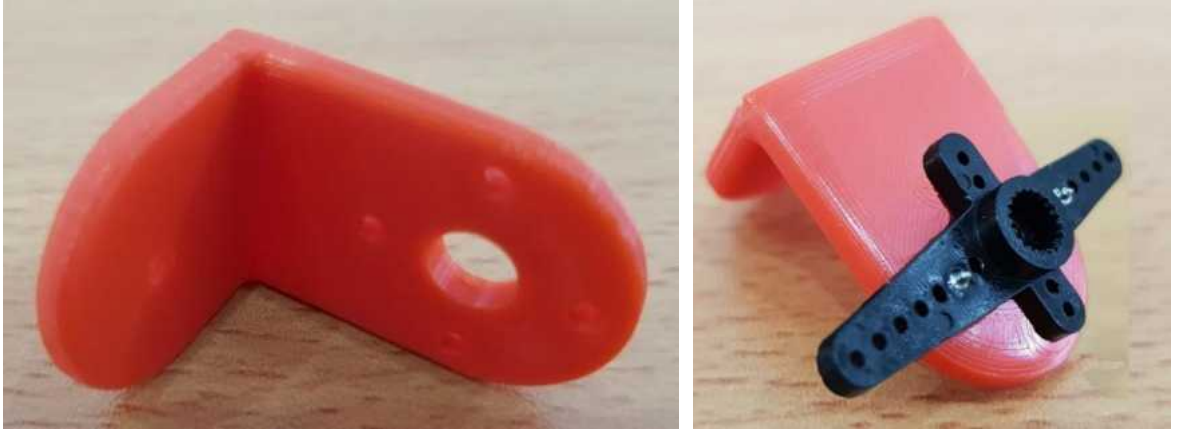


Resim 73 Servo Motorun 3B Yazıcıdan Basılmış Taban Parçasına Yerleştirilmesi



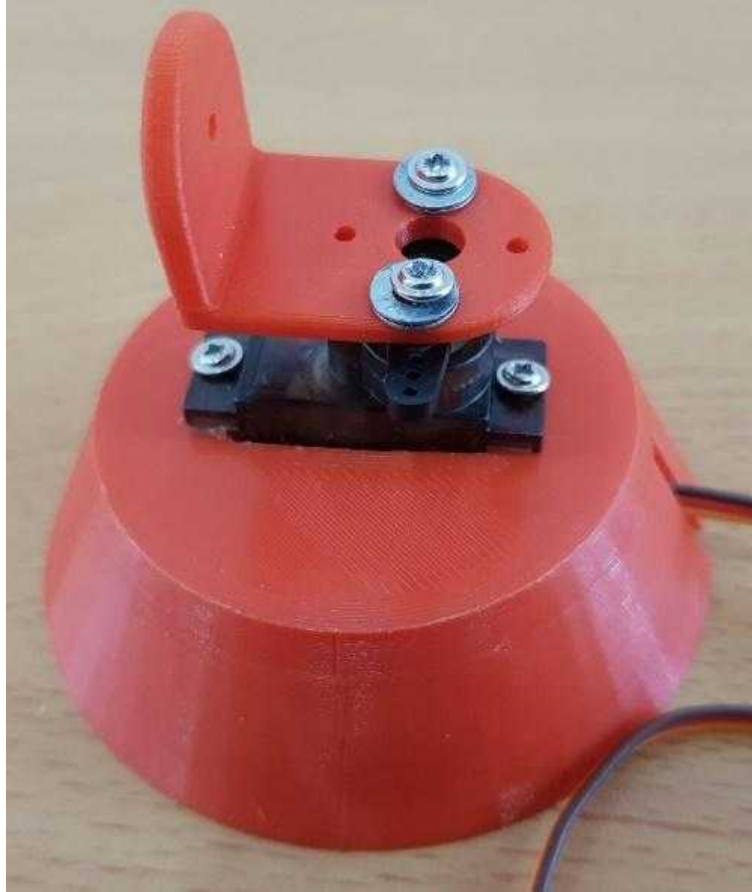
Resim 74 Taban Parçaya Vidalanmış Servo Motor.

Adım 2: İki numaralı servo motoru bağlamak için 3B yazıcıda basılmış bir L-profil kullanılır (Resim 75). Burada servo motor plastiklerinden T şeklinde olanı kullanılır. T şeklinde plastik bulunamazsa diğerleri de kullanılabilir.



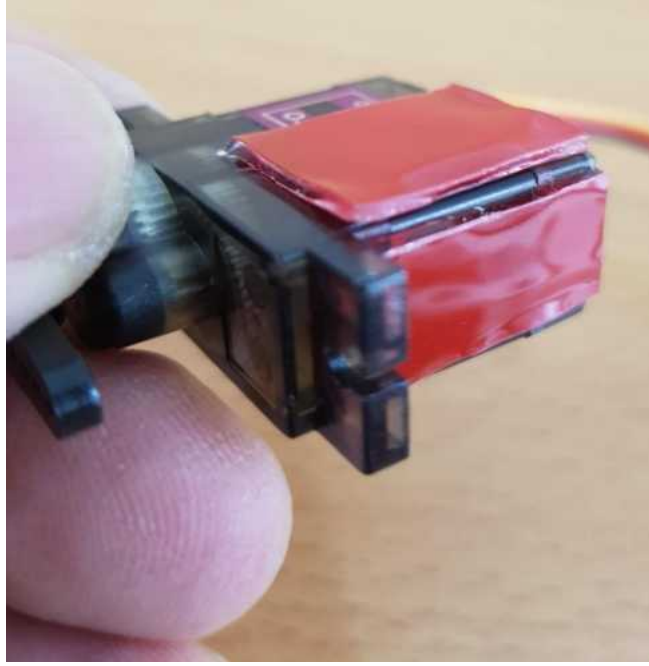
Resim 75 3B-Yazıcıda Basılan L-Profil ve Üzerine Vidalanmış Servo Motor Plastik Parçası

Adım 3: Hazırlanan L-profil, taban motorunun (Motor 1) miline Resim 76'da gösterildiği gibi yerleştirilir.



Resim 76 Hazırlanan L-Profil Parçasının Taban Motoru Miline Yerleştirilmesi

Adım 4: İki numaralı servo motorun iki ayrı yüzeyine çift taraflı silikon bant yapıştırılır (Resim 77). Burada, önceki bölümlerdekine benzer olarak, silikon tabanca ile sıkılan yapıştırıcı silikon malzemede kullanılabilir. Bu, eğitmen ve öğrencinin tercihinine bırakılmıştır.



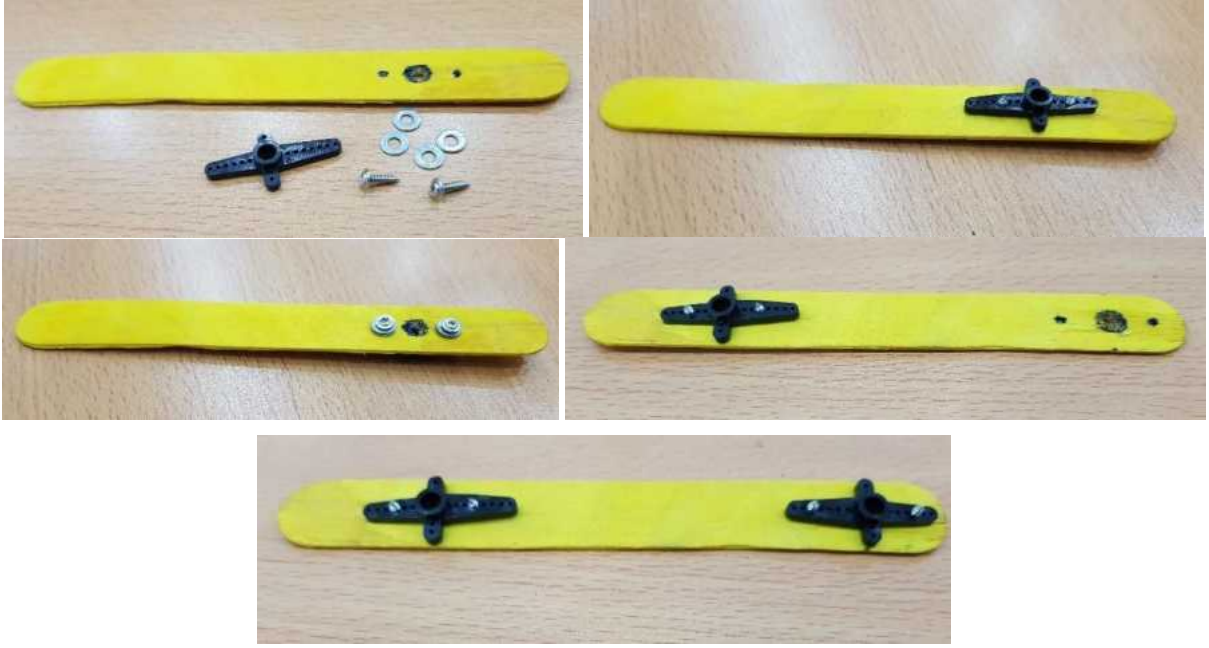
Resim 77 İki Numaralı Servo Motorun İki Ayrı Yüzeyine Çift Taraflı Silikon Bant Yapıştırılması

Adım 5: İki numaralı servo motora yapıştırılmış çift taraflı silikon bantın kırmızı naylonları çıkartılır ve servo motor L-profil parçanın üzerine yapıştırılır (Resim 78).



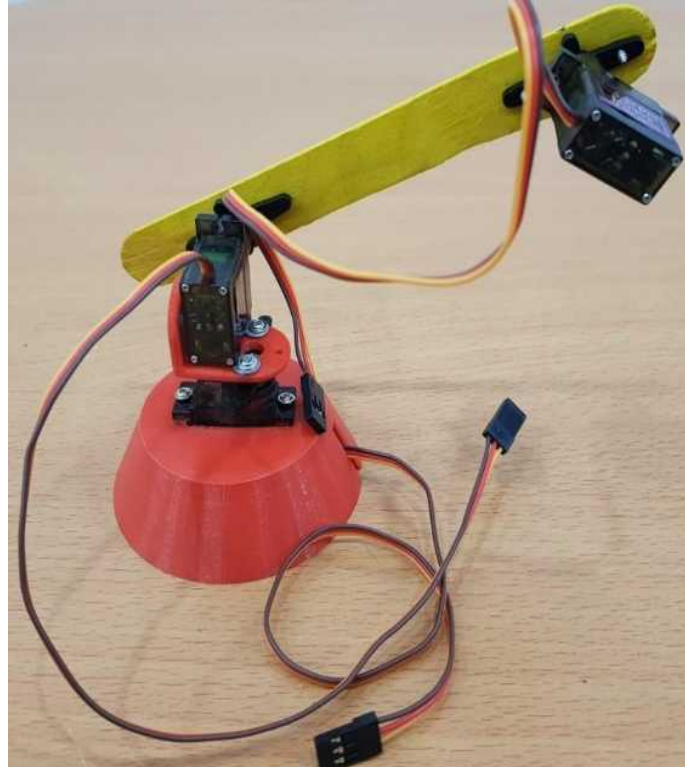
Resim 78 İki Numaralı Servo Motorun L-Profil Parça Üzerine Konumlandırılması.

Adım 6: Bir adet abeslang Resim 79'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. Öğitmenler, dilerlerse, bu ahşap çubuklar yerine, kendilerinin kolayca tasarlayabilecekleri çok basit dikdörtgen çubukları 3 boyutlu yazıcıdan üretme yoluna da gidebilirler. Buradaki seçim, öğretmenlere ve öğrencilere bırakılmakla birlikte, ahşap çubukların kullanımı tavsiye edilmektedir.



Resim 79 Abeslanglara Servo Motor T-Plastik Parçalarının Takılması

Adım 7: Hazırlanmış olan abeslang Resim 80’de gösterildiği gibi 2 numaralı servo motora takılır. Burada hareket açısı 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.



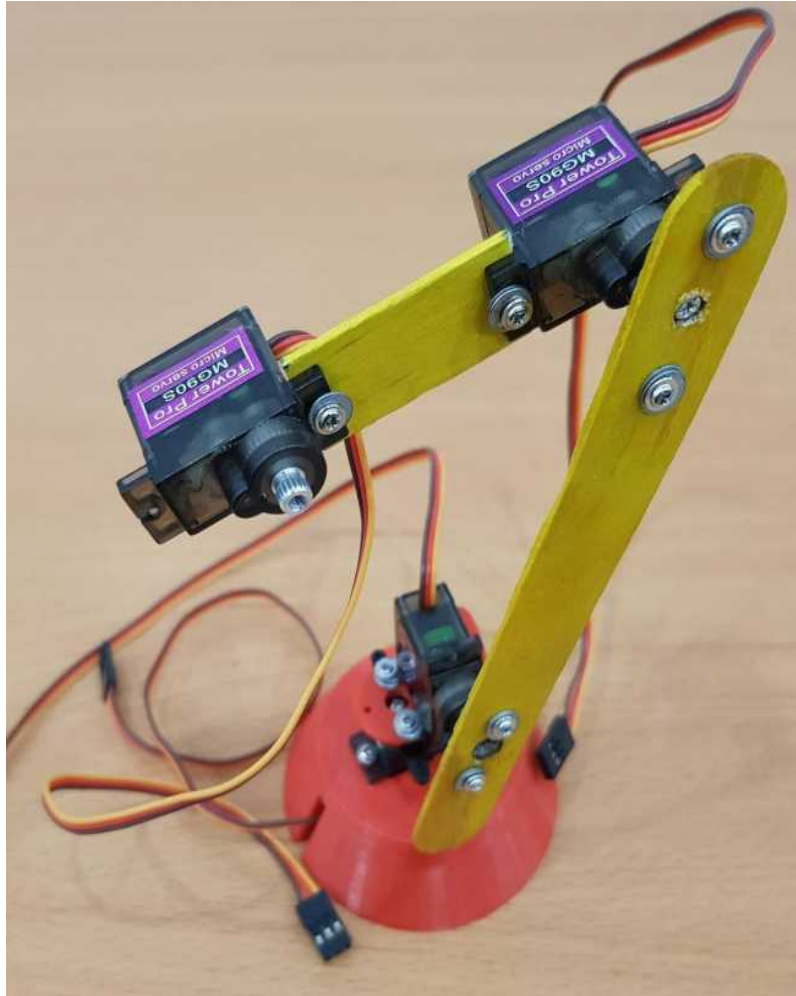
Resim 80 Abeslangın 2 Numaralı Servo Motora ve 3 Numaralı Motorun Abeslanga Monte Edilmesi

Adım 8: Başka bir abeslangtan kısa bir parça kesilir (Resim 81).



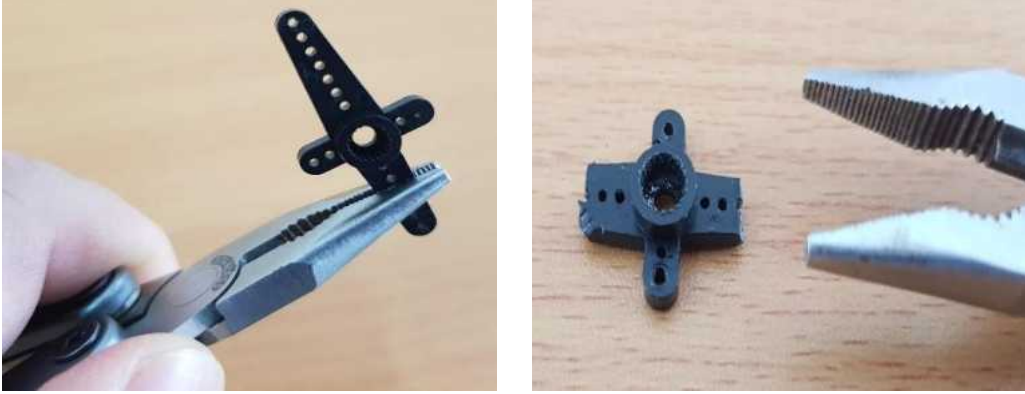
Resim 81 Kısa Bir Parçanın Abeslangtan Kesilmesi ve Hazırlanması

Adım 9: Hazırlanmış olan kısa parça ile 3 ve 4 numaralı servo motorlar birbirlerine bağlanır (Resim 82).



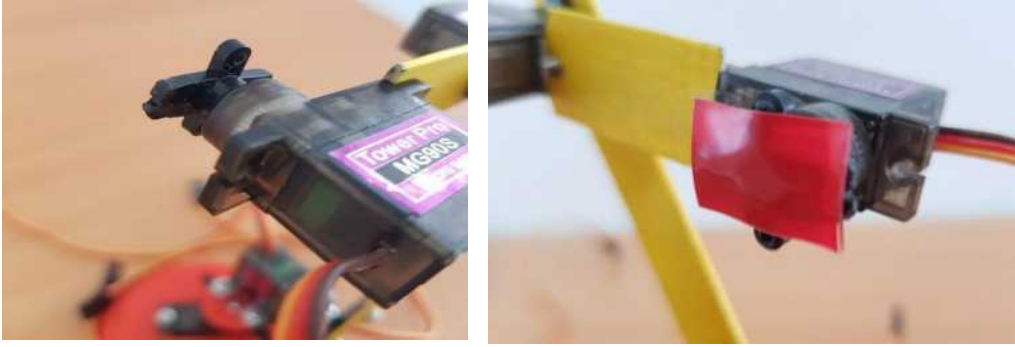
Resim 82 Kısa Parça ile 3 ve 4 Numaralı Servo Motorların Birbirlerine Bağlanması

Adım 10. Servo motorlarla birlikte gelen T-plastik parçanın uzun kenarları kırılarak kısaltılır (Resim 83). Böyle 2 adet parça hazırlanır.



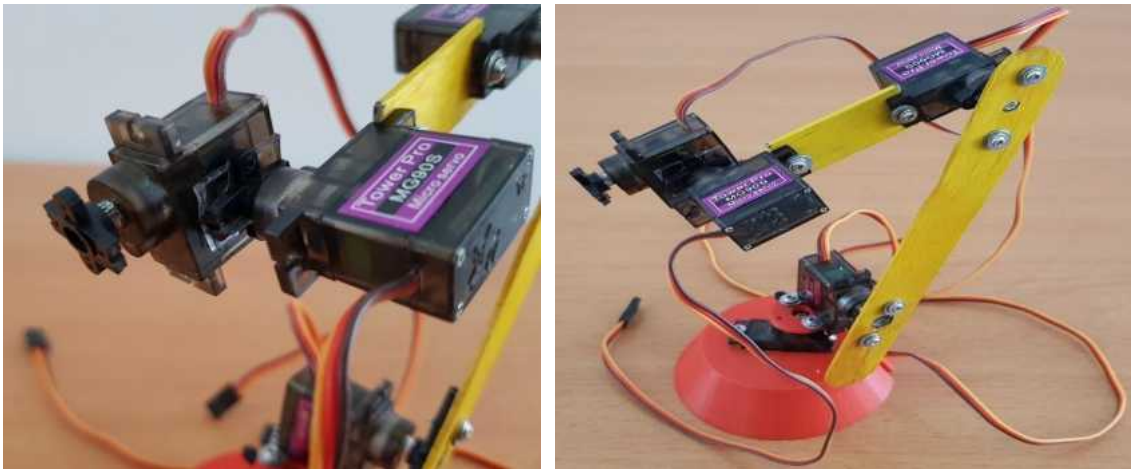
Resim 83 T-Plastik Parçanın Uzun Kenarları Kırılarak Elde Edilen Parça.

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 84).



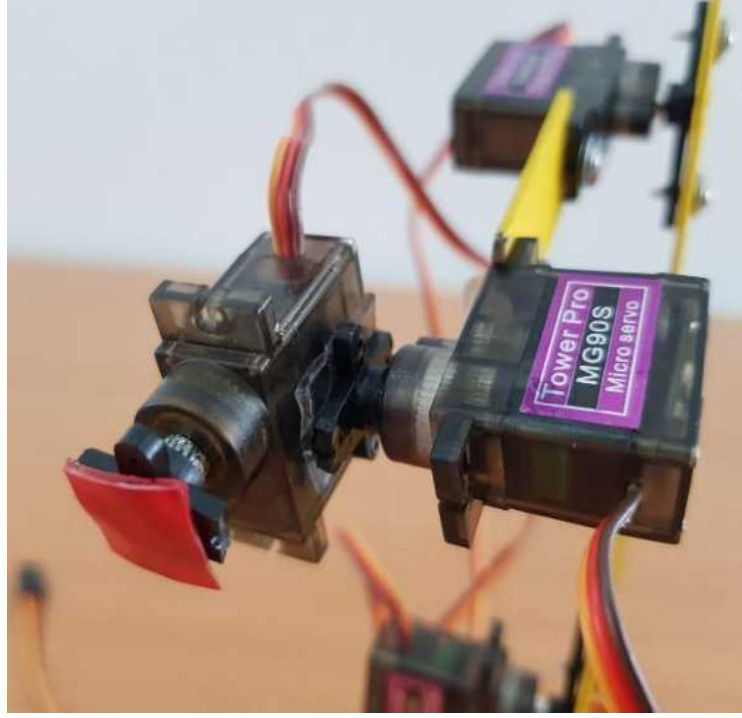
Resim 84 Dördüncü Motorun Milindeki Kısa Plastik Parça ve Çift Taraflı Silikon Bant.

Adım 12. İkinci küçük plastik parça, 5 numaralı 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 85).



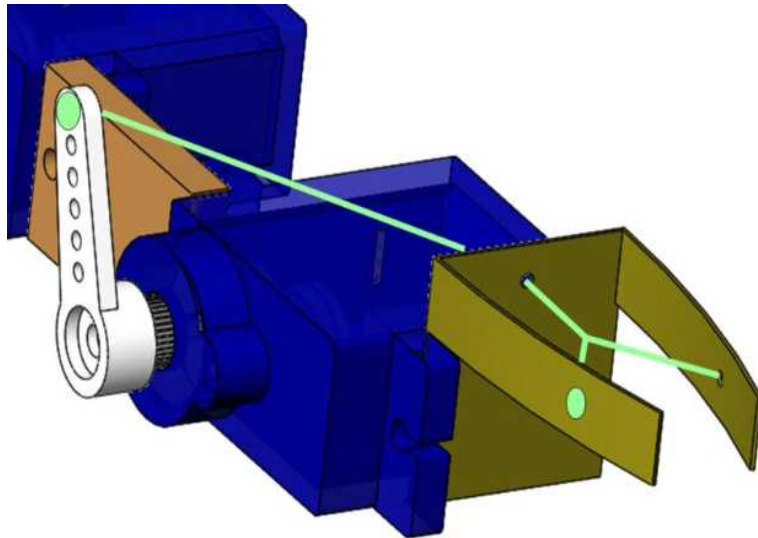
Resim 85 Beşinci Motorun Konumlandırılması.

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 86).



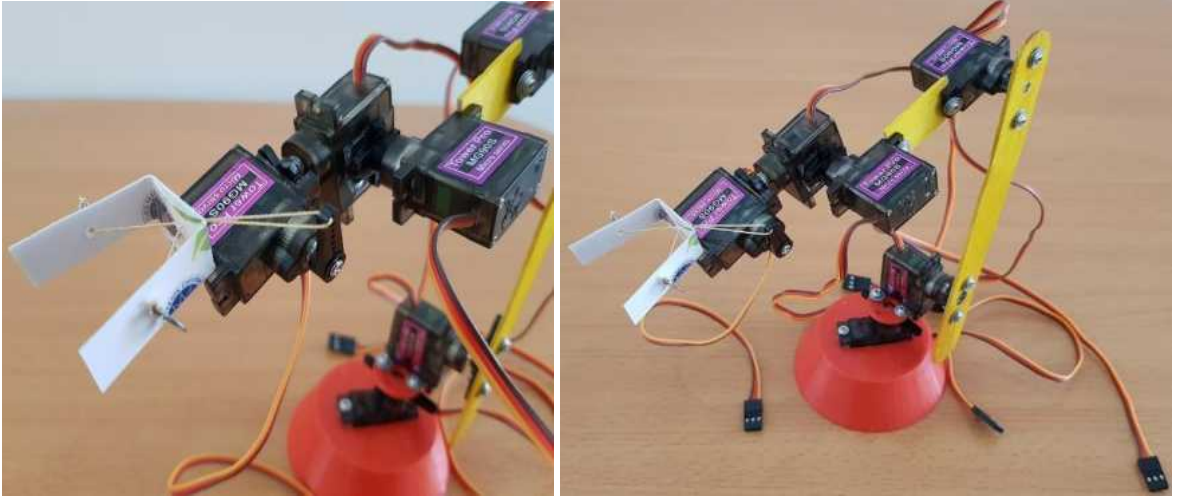
Resim 86 Beş Numaralı Motorun Milindeki Plastik Parçaya Çift Taraflı Silikon Bant Yapıştırılması

Adım 14: Tutucu (Gripper) tasarımında (Resim 87) 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 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.



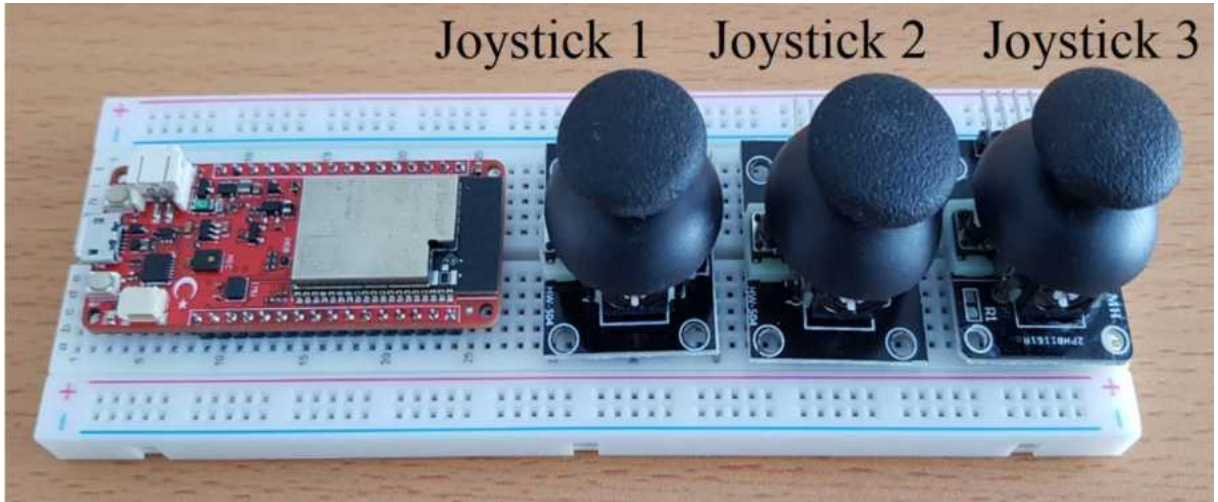
Resim 87 Tutucu Tasarımı

Adım 15. Bir parça çift taraflı silikon bant, 5 numaralı servo motorun milindeki plastik parçanın yüzeyine yapıştırılır (Resim 88).



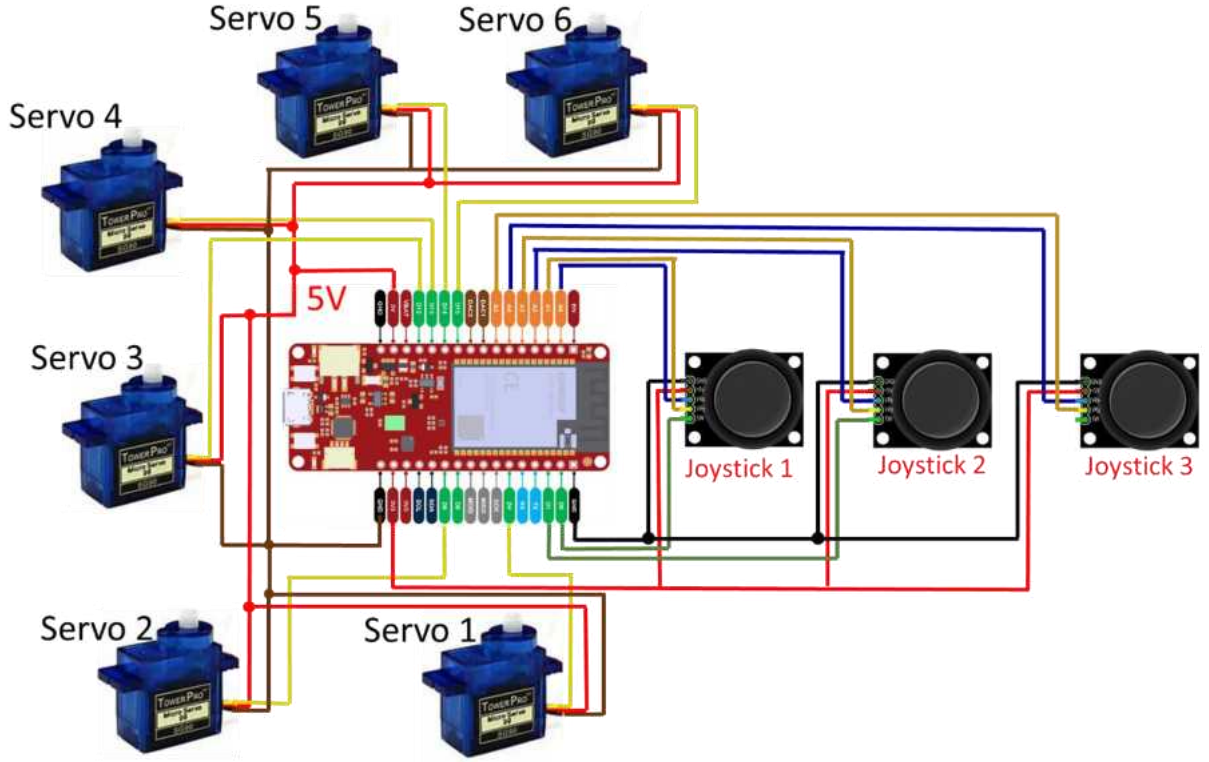
Resim 88 Tutucusu Yerleştirilmiş Robot Manipulator.

Adım 16: Deneyap Kart / Deneyap Kart 1A breadboard üzerine yerleştirilir. Kumanda kollarının alt tabanlarına çift taraflı silikon bant yerleştirilir ve breadboard üzerine Resim 89’da gösterildiği gibi yapıştırılır.

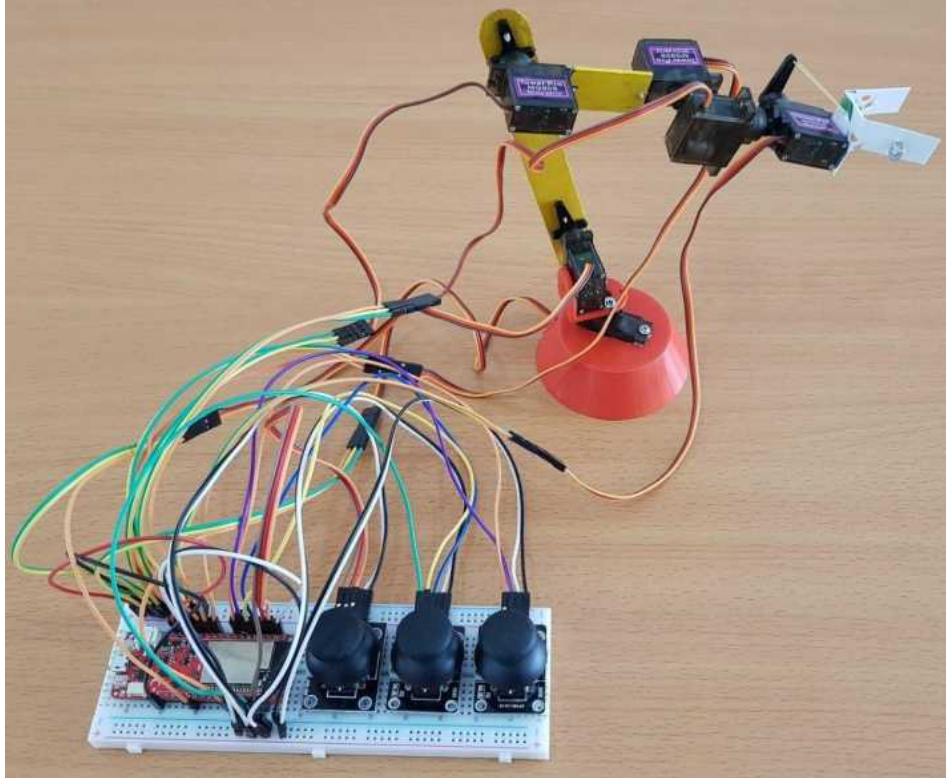


Resim 89 Breadboard Üzerine Konumlandırılmış Kumanda Kolları

Adım 17: Kumanda kolları ve servo motorların Deneyap Kart / Deneyap Kart 1A’ya bağlantıları Resim 90’da verilmiştir.



Resim 90 : Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Devre Şeması



Resim 91 : Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Uygulaması

Adım 18: Deneyap Kart / Deneyap Kart 1An bilgisayarla bağlantısı yapılır. Arduino IDE uygulaması üzerinde program yazılır, kaydedilir, derlenir ve karta yüklenir.

```

1 #include <Deneyap_Servo.h>
2
3 // Dijital kumanda kollarının button pin bağlantıları:
4 #define SW1_pin D0
5 #define SW2_pin D1
6
7 // kumanda kollarının eksen analog bağlantı pinleri
8 #define X1_pin A0
9 #define Y1_pin A1
10 #define X2_pin A2
11 #define Y2_pin A3
12 #define X3_pin A4
13 #define Y3_pin A5
14
15 // Dijital servo pin bağlantıları:
16 #define SERVOPIN1 D4
17 #define SERVOPIN2 D9
18 #define SERVOPIN3 D12
19 #define SERVOPIN4 D13
20 #define SERVOPIN5 D14
21 #define SERVOPIN6 D15
22
23 // Motor dönüş açısı alt ve üst limitleri:
24 // Aşağıdaki teta değerleri 0-3000 arasında kalibre edilmiştir. 3000 değeri 180 dereceye karşılık gelmektedir.
25 int teta1min=0;
26 int teta1max=3000;
27 int teta2min=0;
28 int teta2max=3000;
29 int teta3min=500;
30 int teta3max=2500;
31 int teta4min=0;
32 int teta4max=3000;
33 int teta5min=0;
34 int teta5max=3000;
35 int teta6min=1500;
36 int teta6max=2000;
37
38 // Motor dönüş açıları için değişkenlerin tanımlanması
39 int valin1, valin2, valin3, valin4, valin5, valin6;
40 //Başlangıç açısı değerleri
41 int valcalc1=1500;
42 int valcalc2=1500;
43 int valcalc3=1500;
44 int valcalc4=1500;
45 int valcalc5=1500;
46 int valcalc6=2000;
47
48 int jystp=20;
49 // kumanda kollarıyla motor hareketleri sağlandığında her bir döngüde istenen açı değeri
50 // (bu yazılımda 3000 değerinin 360 dereceye karşılık geldiğini unutmayın.
51
52 bool playmode=false; // Robotun tekrarlama (oylama) modu değişkeni. Değer false olduğunda, öğretim modunda demektir.
53 int ButtonMode=0; // Buton modu değişkeninin tanımlanması. Sıfır olduğunda, öğretim modunda demektir.
54 unsigned long CurrentMilliSec = millis(); // millis() fonksiyonu, şu anki saati milisecond cinsinden vermektedir.
55 //Zamanlayıcı olarak bu fonksiyonu kullanmaktayız. Herhangi bir anda, o anki zamanı bir long değişkenine atayabiliriz.
56 long PreMilliSec=0; // Bir evvelki zaman (milisaniye cinsinden)
57 int NumMemory=0; // Hafıza sayacı (Maksimum 50 olabilir.
58 // Deneyap kartının kapasitesine bağlı olarak bu sınır 50'den yukarıda olabilir.)
59 int PreNumMemory=0; // Bir evvelki moddaki hafıza değeri.
60 int Step=-1; // Öğretim modunda, servo pozisyonlarını hafızaya alırken kullanılan bir değişken.
61 // Servo pozisyonlarının hafızaya alındığı 50'lik seriye sahip değişkenler.
62 int joint1[50];
63 int joint2[50];
64 int joint3[50];
65 int joint4[50];
66 int joint5[50];
67 int joint6[50];
68 int maxarray=49; //Hafıza hatası vermeyi önlemek için 49 değerine sahip bir limit değişkeni.
69 int prehiz=10; // Bir önceki hız değerinin atanacağı değişken tanımlaması.
70 int normalhiz=10; // Normal hız değişkeni
71 // hiz=1 çok yavaş, 5 yavaş, 10 normal, 15 hızlı, 20+ çok hızlı
72 int hiz=10; // Şu anki hız değişkeni

```

Resim 92 Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Kodu – I

```

73 int prepos1, prepos2, prepos3, prepos4, prepos5, prepos6; // bir evvelki motor konumları.
74 int pos11, pos21, pos31, pos41, pos51, pos61; // Motor konumları ile ilgili ikincil (yardımcı) değişkenler.
75
76 Servo myservo1, myservo2, myservo3, myservo4, myservo5, myservo6; // Servo motorların tanımlanması.
77
78 void setup() {
79     myservo1.attach(SERVOPIN1);
80     myservo2.attach(SERVOPIN2,1);
81     myservo3.attach(SERVOPIN3,2);
82     myservo4.attach(SERVOPIN4,3);
83     myservo5.attach(SERVOPIN5,4);
84     myservo6.attach(SERVOPIN6,5);
85
86     // Joystick butonlarının pinlerinin tanımlanması (Normalde HIGH).
87     pinMode(SW1_pin, INPUT_PULLUP);
88     pinMode(SW2_pin, INPUT_PULLUP);
89     // kumanda kolu pinlerinin giriş olarak tanımlanması
90     pinMode(X1_pin, INPUT);
91     pinMode(Y1_pin, INPUT);
92     pinMode(X2_pin, INPUT);
93     pinMode(Y2_pin, INPUT);
94     pinMode(X3_pin, INPUT);
95     pinMode(Y3_pin, INPUT);
96
97     // Joystick butonlarının pinlerinin değerleri normalde HIGH' a atandı.
98     digitalWrite(SW1_pin, HIGH);
99     digitalWrite(SW2_pin, HIGH);
100    // Tercihe bağlı olarak, Seri port'a "Ready" yazdırılması
101    Serial.begin(115200);
102    Serial.println("Ready.");
103 }
104
105 void loop() {
106     CurrentMilliSec = millis(); // Şu anki zaman değeri (ms)
107     Button(); // Kendi yazdığımız Button fonksiyonunun çağırılması.
108     if (playmode==false && Step==1){ // Öğretme modunda iken,
109         ReadAnalog(); // Analog kanalların okunması için kendi yazdığımız ReadAnalog fonksiyonunun çağırılması.
110         // Servo motorlara dönüş komutlarının verilmesi:
111         myservo1.writeMicroseconds(valcalc1);
112         myservo2.writeMicroseconds(valcalc2);
113         myservo3.writeMicroseconds(valcalc3);
114         myservo4.writeMicroseconds(valcalc4);
115         myservo5.writeMicroseconds(valcalc5);
116         myservo6.writeMicroseconds(valcalc6);
117
118         delay(int(300./Hız)); // Hız kontrolü için genel döngü içerisindeki gecikme ayarı.
119     } else {
120         playit(); //tekrarlama (veya oynama) modunda iken kendi yazdığımız Play fonksiyonunun çağırılması.
121     }
122 }
123 // kumanda kollarının ürettiği analog sinyallerin okunduğu ve motor açılış değerlerinin bu sinyallere göre
124 // değiştirildiği fonksiyon aşağıda verilmiştir.
125
126 void ReadAnalog() {
127     // Analog pinlere giren kumanda kolu değerlerinin okunması.
128     valin1 = analogRead(Y1_pin);
129     valin2 = analogRead(X1_pin);
130     valin3 = analogRead(X2_pin);
131     valin4 = analogRead(Y2_pin);
132     valin5 = analogRead(X3_pin);
133     valin6 = analogRead(Y3_pin);
134
135     // Belli sınırlar dahilinde, Joystick değerlerine göre servo motor dönüş açılarının tayini.
136     if (valin1<1000 && valcalc1>teta1min) valcalc1-=jystp;
137     if (valin1>3000 && valcalc1<teta1max) valcalc1+=jystp;
138     if (valin2<1000 && valcalc2>teta2min) valcalc2-=jystp;
139     if (valin2>3000 && valcalc2<teta2max) valcalc2+=jystp;
140     if (valin3<1000 && valcalc3>teta3min) valcalc3-=jystp;
141     if (valin3>3000 && valcalc3<teta3max) valcalc3+=jystp;
142     if (valin4<1000 && valcalc4>teta4min) valcalc4-=jystp;
143     if (valin4>3000 && valcalc4<teta4max) valcalc4+=jystp;
144     if (valin5<1000 && valcalc5>teta5min) valcalc5-=jystp;

```

Resim 93 Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Kodu – II

```

145 if (valin5>3000 && valcalc5<teta5max) valcalc5+=jystp;
146 if (valin6<1000 && valcalc6>teta6min) valcalc6-=jystp;
147 if (valin6>3000 && valcalc6<teta6max) valcalc6+=jystp;
148 }
149 // Buton fonksiyonu oldukça kapsamlıdır. Burada 2 adet buton ile robot hareket modları tayin edilmektedir.
150 // Öğretme modunda iken kumanda kolu 1'in butona tek tıklama ile mevcut motor açılarını hafızaya alma
151 // işlemi yapılabilmekte, öğretme modundan tekrarlama moduna veya tersine yine Kumanda kolu 1'in butonuna
152 // çift tıklayarak geçilebilmektedir. Bu uygulamada da motorlara hızlı, normal ve yavaş olmak üzere 3 ayrı
153 // seviye tayin edilmiştir. Bu hız seviyeleri arasındaki geçişler Kumanda kolu 2'nin
154 // butonuna basılarak gerçekleştirilmektedir. Belirtilen buton aktivitelerini gerçekleştiren kodlar aşağıda verilmiştir.
155 void Button() { // Butona basılma durumu
156   if (playmode==false){ // Eğer öğrenme modunda iken,
157     if (digitalRead(SW1_pin) == 0){
158       delay(100);
159       if (digitalRead(SW1_pin) == 1) { // butona tıklanmış ise,
160         if (ButtonMode == 0) {
161           ButtonMode = 1; // İlk buton tek tık tespiti.
162           PreMilliSec = CurrentMilliSec;
163         } else if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec < 500)) {
164           ButtonMode = 2; // İlk buton çift tık tespiti.
165         }
166       }
167     }
168     if ((ButtonMode == 1) && (CurrentMilliSec - PreMilliSec > 600)) {
169       saveit(); // İlk buton tek tık tespit edildiğinde, motor konumları hafızaya alınır.
170       NumMemory += 1;
171       if (NumMemory>maxarray) NumMemory= maxarray;
172       ButtonMode = 0;
173       Serial.print("Saved: "); Serial.println(NumMemory);
174     } else if (ButtonMode == 2 && (CurrentMilliSec - PreMilliSec > 600)) {
175       if ((NumMemory>1 || PreNumMemory>1) && playmode==false){
176         Serial.println("Playmode...");
177         playmode = true; // Çift tık tespit edildiğinde tekrarlama (oynama) moduna geçilir.
178         ButtonMode = 0;
179         if (NumMemory<2) NumMemory=PreNumMemory; else PreNumMemory=NumMemory;
180         // Eğer hafızaya herhangi bir konum alınmamış olup ta çift tık yapılırsa, bir önceki oynatılmış
181         // konum bilgilerinin tekrar kullanılması için devreye alınması.
182         // Motor açılı konumlarının hafızaya alınması:
183         joint1[0]=pos11;
184         joint2[0]=pos21;
185         joint3[0]=pos31;
186         joint4[0]=pos41;
187         joint5[0]=pos51;
188         joint6[0]=pos61;
189
190         Step=NumMemory-1;
191         ReadAnalog();
192         prepos1=valcalc1;
193         prepos2=valcalc2;
194         prepos3=valcalc3;
195         prepos4=valcalc4;
196       }
197     }
198     if (ButtonMode<3 && CurrentMilliSec - PreMilliSec > 1000) {
199       ButtonMode = 0; // Tekrar öğretme moduna geçiş.
200     }
201   }
202   else if (ButtonMode<3 && digitalRead(SW1_pin) == 0) {
203     ButtonMode = 3;
204     Serial.println("Stopping."); // Tekrarlamayı (Oynamayı) durdur.
205     ReadAnalog();
206     NumMemory+=1;
207     // Durulan konumun hafızanın en son basamağına kaydedilmesi.
208     joint1[NumMemory-1] = valcalc1;
209     joint2[NumMemory-1] = valcalc2;
210     joint3[NumMemory-1] = valcalc3;
211     joint4[NumMemory-1] = valcalc4;
212     joint5[NumMemory-1] = valcalc5;
213     joint6[NumMemory-1] = valcalc6;
214   }
215   // İkinci butona her tık yapıldığında robot hareket hızı değiştirilir.
216   //Normal, yavaş, hızlı olmak üzere 3 ayrı hız için, aşağıdaki kod yazılmıştır.

```

Resim 94 Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Kodu – III

```

217 //Diğer detaylar, önceki bölümlerdeki kodlarda anlatıldığı gibidir.
218 if (digitalRead(SW2_pin) == 0) {
219     if (prehiz==hiz) {
220         if (hiz==normalhiz) {
221             hiz=2*normalhiz;
222             jystp=40;
223             Serial.println("High speed.");
224             if (playmode==false) {
225                 delay(300);
226             }
227         } else {
228             if (hiz==(2*normalhiz)) {
229                 hiz=normalhiz/2;
230                 jystp=10;
231                 Serial.println("Low speed.");
232                 if (playmode==false) {
233                     delay(300);
234                 }
235             } else if (hiz==(normalhiz/2)) {
236                 hiz=normalhiz;
237                 jystp=20;
238                 Serial.println("Normal speed.");
239                 if (playmode==false) {
240                     delay(300);
241                 }
242             }
243         }
244     }
245 } else {
246     prehiz=hiz;
247 }
248 }
249 // Mevcut servo motor açılarının hafızaya kaydedildiği fonksiyon aşağıda verilmiştir.
250 void saveit() {
251     joint1[NumMemory] = valcalc1;
252     joint2[NumMemory] = valcalc2;
253     joint3[NumMemory] = valcalc3;
254     joint4[NumMemory] = valcalc4;
255     joint5[NumMemory] = valcalc5;
256     joint6[NumMemory] = valcalc6;
257
258     prepos1=valcalc1;
259     prepos2=valcalc2;
260     prepos3=valcalc3;
261     prepos4=valcalc4;
262     prepos5=valcalc5;
263     prepos6=valcalc6;
264
265     if (NumMemory==1){
266         pos1=joint1[0];
267         pos2=joint2[0];
268         pos3=joint3[0];
269         pos4=joint4[0];
270         pos5=joint5[0];
271         pos6=joint6[0];
272     }
273 }
274
275 // Hafızaya kaydedilen servo motor konumlarının tekrarlatıldığı fonksiyon aşağıda verilmiştir.
276 // Detaylar, önceki bölümlerde verilen kodlardaki gibidir.
277 void playit()
278 {
279     float pos1=prepos1;
280     float pos2=prepos2;
281     float pos3=prepos3;
282     float pos4=prepos4;
283     float pos5=prepos5;
284     float pos6=prepos6;
285
286     // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için dönüş açısı
287     // mesafeleri yukarıda tanımlanmış farklara bölüştürülüp servo motorlara uygulanmıştır.
288     float dif1=abs(joint1[Step]-pos1);

```

Resim 95 Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Kodu – IV

```

289 float dif2=abs(joint2[Step]-pos2);
290 float dif3=abs(joint3[Step]-pos3);
291 float dif4=abs(joint4[Step]-pos4);
292 float dif5=abs(joint5[Step]-pos5);
293 float dif6=abs(joint6[Step]-pos6);
294 float maxdif=0.0;
295
296 if (dif1>maxdif) maxdif=dif1;
297 if (dif2>maxdif) maxdif=dif2;
298 if (dif3>maxdif) maxdif=dif3;
299 if (dif4>maxdif) maxdif=dif4;
300 if (dif5>maxdif) maxdif=dif5;
301 if (dif6>maxdif) maxdif=dif6;
302
303 maxdif=maxdif/hiz;
304 if (maxdif<1.0) maxdif=1.0;
305
306 float stp1=(joint1[Step]-pos1)/maxdif;
307 float stp2=(joint2[Step]-pos2)/maxdif;
308 float stp3=(joint3[Step]-pos3)/maxdif;
309 float stp4=(joint4[Step]-pos4)/maxdif;
310 float stp5=(joint5[Step]-pos5)/maxdif;
311 float stp6=(joint6[Step]-pos6)/maxdif;
312 for (int i=1;i<=int(maxdif);i++){
313     // Motor dönüşlerinin olabildiğince düzgün ve sürekli hissedilmesi için
314     // dönüş açısı mesafeleri yukarıda tanımlanmış farklara bölüştürülüp servo motorlara uygulanmıştır.
315     pos1=pos1+stp1;
316     pos2=pos2+stp2;
317     pos3=pos3+stp3;
318     pos4=pos4+stp4;
319     pos5=pos5+stp5;
320     pos6=pos6+stp6;
321
322     if (i==int(maxdif)){
323         pos1=joint1[Step];
324         pos2=joint2[Step];
325         pos3=joint3[Step];
326         pos4=joint4[Step];
327         pos5=joint5[Step];
328         pos6=joint6[Step];
329     }
330     myservo1.writeMicroseconds(int(pos1));
331     myservo2.writeMicroseconds(int(pos2));
332     myservo3.writeMicroseconds(int(pos3));
333     myservo4.writeMicroseconds(int(pos4));
334     myservo5.writeMicroseconds(int(pos5));
335     myservo6.writeMicroseconds(int(pos6));
336
337     prepos1=pos1;
338     prepos2=pos2;
339     prepos3=pos3;
340     prepos4=pos4;
341     prepos5=pos5;
342     prepos6=pos6;
343
344     delay(10);
345     if (ButtonMode<3) Button();
346     else delay(5);
347 } if (ButtonMode==3 && Step==NumMemory-1) { // Oynatma modunda durdurma işlemi.
348     NumMemory = 0;
349     Step=-1;
350     ButtonMode = 0;
351     playmode=false;
352     Serial.println("Stopped.");
353 } else {
354     Step+=1;
355     if (Step>NumMemory-1) Step=0;
356     // Hafıza aşıldığında, ilk adıma dönüş. Burada istenirse "Step=NumMemory-1;" de
357     // yazılabilir ve hafızada sürekli en son adımda kalınır.
358 }
359 }

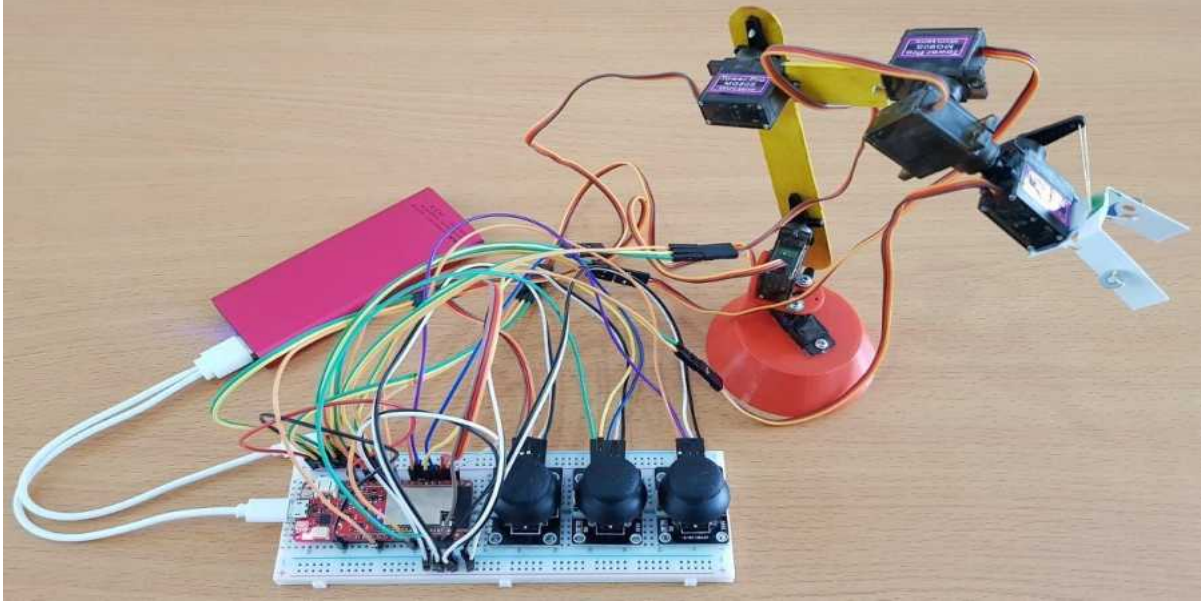
```

Resim 96 Beş Serbestlik Dereceli ve 1 Tutuculu Bir Sistemin Üç Kumanda Kolu ile Kontrolü Kodu – V

Adım 19: Program Deneyap Kart / Deneyap Kart 1A'ya yüklendikten sonra motorların kumanda kolu ile kontrolü yapılabilmekte, öğretme ve tekrarlatma modları denenebilmektedir (Resim 97).

Uyarı:

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

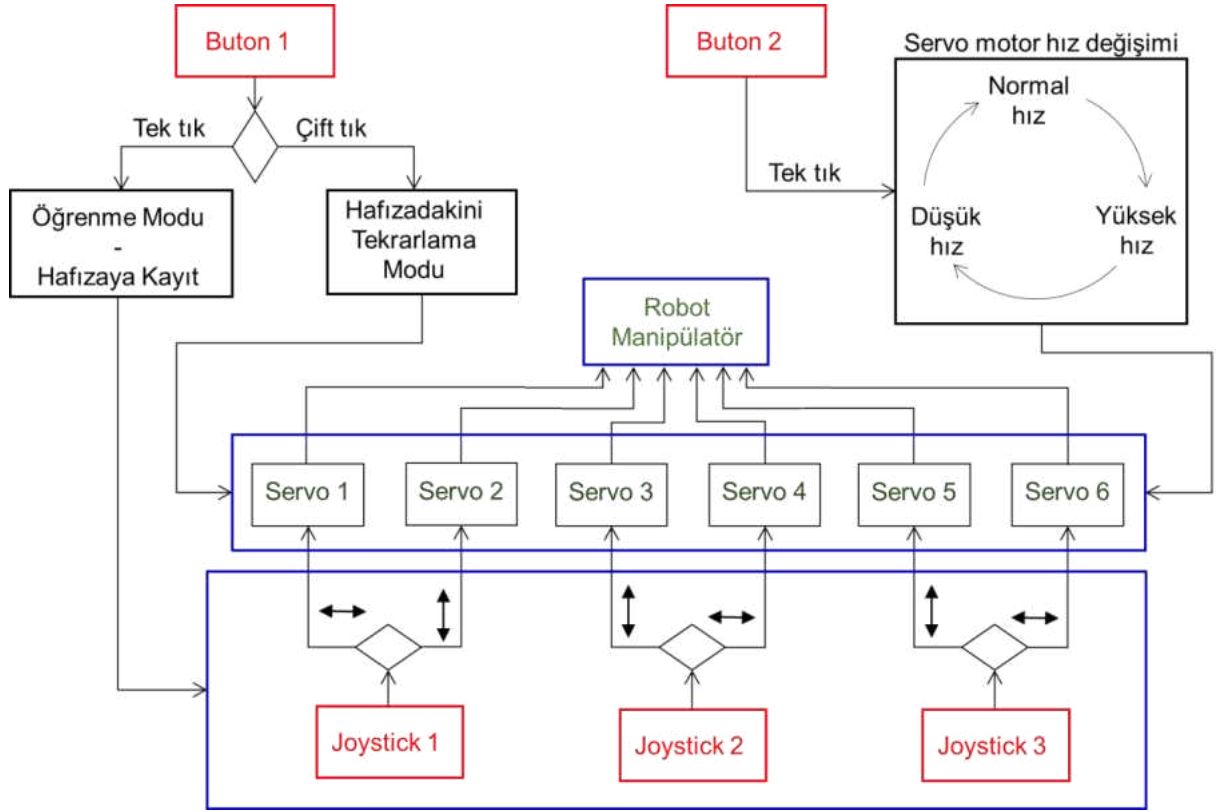


Resim 97 Tüm Sistemin Harici Bir Batarya ile Test Edilmesi

Robotun çalışma algoritması, Resim 98' 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 (oyunatma 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 oyunatma modlarında iken ikinci kumanda kolunun butonuna tek tıklama yapıldığında, bir hız modundan diğerine geçilebilmektedir.



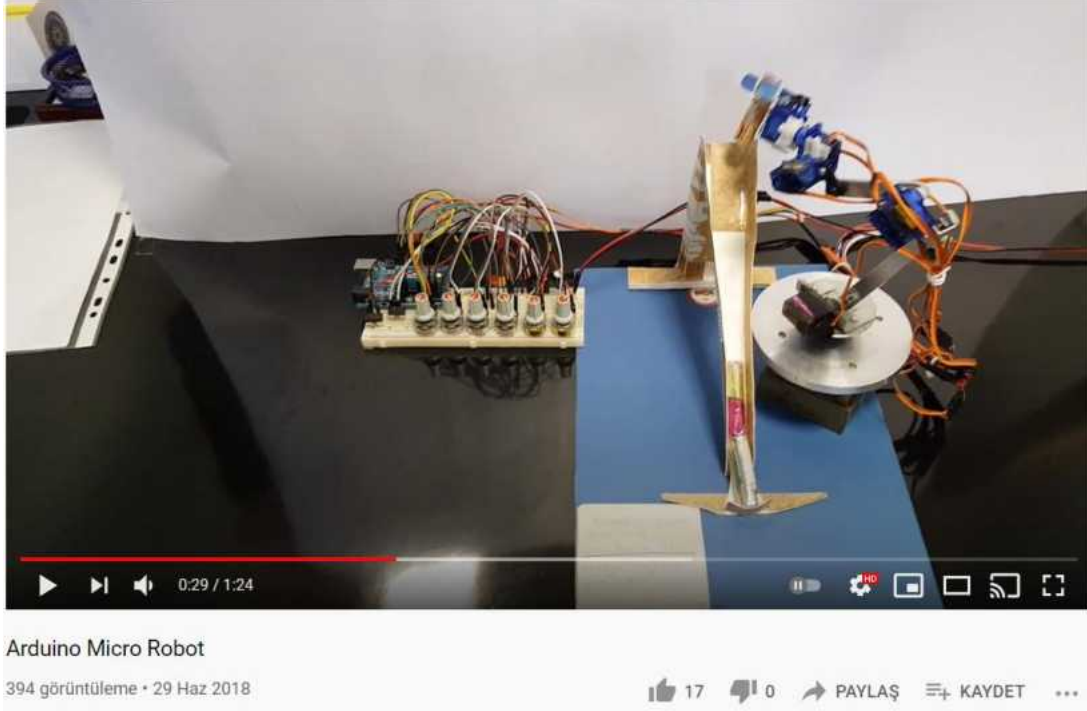
Resim 98 Robotun Çalışma Algoritmasını Gösteren Akış Diyagramı

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 97).

Video Linki: <https://www.youtube.com/watch?v=QpYnd2ucZTY>



Resim 99 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, eğitime 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:
 - o 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>
 - o 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=lwE6VIPTcFg>

4. İLAVE ETKİNLİK

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

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.