

T.C.
TURGUT ÖZAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

UZAKTAN EĞİTİM İÇİN HİBRİT EĞİTİM ROBOTU TASARIMI

YÜKSEK LİSANS TEZİ

Hazırlayan

Mahmut KASAP

Tez Danışmanı

Yrd. Doç. Dr. Özgür YILMAZ

Ankara-2015

T.C.
TURGUT ÖZAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

UZAKTAN EĞİTİM İÇİN HİBRİT EĞİTİM ROBOTU TASARIMI

YÜKSEK LİSANS TEZİ

Hazırlayan

Mahmut KASAP

Tez Danışmanı

Yrd. Doç. Dr. Özgür YILMAZ

Ankara-2015

Bilimsel Etik Bildirim Sayfası

Turgut Özal Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı beyan ederim.

08/09/2015

Mahmut KASAP

ONAY

Mahmut KASAP tarafından hazırlanan “UZAKTAN EĞİTİM İÇİN HİBRİT EĞİTİM ROBOTU TASARIMI” başlıklı bu çalışma, 08/09/2015 tarihinde yapılan savunma sınavı sonucunda *oybirliđi* ile başarılı bulunarak jürimiz tarafından *Elektrik ve Bilgisayar Mühendisliđi Anabilim Dalında Yüksek Lisans* tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Özgür YILMAZ (Başkan)

Doç. Dr. Yıldıray YALMAN

Yrd. Doç. Dr. Ömer ASAL (Gazi Üniversitesi)

ÖNSÖZ

Bu tez, uzaktan eğitim alanında temel sorunlardan biri olarak görülen eğitici ve öğrencinin mekansal bağımlılığından kaynaklanan etkileşim zorluklarına farklı ve genel bir çözüm üretmek amacıyla yazılmıştır.

Tez yapım aşamasında hedef olarak belirlediğimiz, her yararlanıcının ülke içi imkanlar ile projeyi gerçekleştirilebilecek olması, bizi her ne kadar kısıtladıysa da bu amaca ulaşabildiğimize inanıyorum.

Eğitimim süresince, bilgisi, tecrübesi ve çalışmaları ile bana yol gösteren, sadece ilmi, bilgisi, tecrübesi ve bilim adamı kimliği ile değil, hoşgörüsü, en zor zamanlarında bile desteğini esirgememesi ve insanlığı ile de örnek aldığım, öğrencisi olmaktan gurur duyduğum değerli hocam Yrd. Doç. Dr. Özgür YILMAZ'a,

Proje ve tezin ortaya çıkmasında ilmi, desteği, yol gösterici fikirleri ve çalışmaları için değerli hocam sayın Yrd. Doç. Dr. Mehmet Can ŞAHİN'e,

Projemizin gerçekleşmesinde en büyük zorluklardan biri olan kaynak konusunda 001-10-2014 numaralı bilimsel araştırma projesi ile bizi destekleyen, çalışmalarımız için bizi ödüllendiren, bilgisini, yardımını ve zamanını bizden esirgemeyen Turgut Özal Üniversitesi ve yine üniversitemiz Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü ve Bilimsel Araştırmalar Koordinatörlüğü yönetimine, hocalarına, çalışanlarına,

Projemizin yapım aşamasında 114E554 kodlu proje ile bizi destekleyen TÜBİTAK'a,

Aynı zamanda robotun kesimi, montajı ve testlerinde yardım eden öğrenci arkadaşlarımızdan Sezai Furkan PÜR, Muhammet Mustafa ÖZCAN ve İbrahim Ethem GÜRSOY'a,

Bu günlere gelmemde büyük emeği olan değerli aileme, çalışmalarım esnasında büyük sabır ve fedakarlıkla her türlü zorluğa rağmen beni destekleyen yardım eden sevgili eşim Dr. Servet Seçkin TUNÇ KASAP'a yine çalışmalarımın ötürü baba ilgisinden feragat eden biricik oğlum Ali Erol KASAP'a minnet ve teşekkürlerimi sunmayı bir borç bilirim.

ÖZET

KASAP, Mahmut. Uzaktan Eğitim İçin Hibrit Eğitim Robotu Tasarımı, Yüksek Lisans Tezi, Ankara, 2015.

Eğitim alanındaki çalışmalar uzaktan eğitimin, ömür boyu eğitime katkısının büyük olduğunu göstermektedir. BigBlueButton projesi, senkron uzaktan eğitim için geliştirilmiş, iyi sonuçlar veren, açık kaynak ve yaygın kullanılan bir uygulamadır. Telepresence (televarlık), bir kişinin teknoloji sayesinde bulunduğu yerden başka bir yerde eş zamanlı iş yapabilme kabiliyeti olarak ifade edilmektedir [1]. Telepresence robot; diğer ortamlarda olma hissini verebilmek için duyulara (algılayıcılara-sensörlere) ve o ortamda iş yapabilme kabiliyeti sağlayabilmek için uzuvlara (eyleyicilere-manipülatörlere) sahip bir televarlık olarak tanımlanmaktadır [1]. Günümüzde bu sistemler ayrı şekilde mevcut olsalar da, uzaktan eğitim sistemi ile entegre, yerel ortamda eğitim faaliyetlerini gözlemleyebilecek, lise ve üzeri düzey eğitim materyallerini kullanabilecek, öğrencilerden girdi alabilecek, gezgin televarlık robot bulunmamaktadır. Böyle bir televarlık robot öğrencilerin uzaktan eğitim süresince dikkat seviyelerini yükseltmek için gereklidir. Ayrıca, böyle bir televarlık robot ulaşılması zor bölgelerde eğitim ve engelli öğrencilere evde eğitim için de gereklidir. Çünkü bunun için nitelikli personele ihtiyaç vardır. Bu personelin eğitimi, ulaşımı zor bölgelerde istihdamı ve bu personelin giderleri ciddi miktarda maddi kaynak gerektirmektedir.

Çalışma kapsamında, evlerde ve okullarda kullanılabilen, senkron uzaktan eğitim sistemi ile eş zamanlı görüntü, ses ve ders materyali paylaşabilen, öğrencilerin yerel ortamdan veri girişine destek sunan gezgin televarlık robot tasarladık. Bu platform kolay gerçekleştirilebilir yapısı ve maliyetinin az oluşu ile, eğitimde eşit imkanların ve kaynakların gerekli yerlere götürülmesini bunun yanında uzaktan eğitimdeki etkileşim sorununun giderilmesini mümkün kılacaktır. Proje kapsamında geliştirilen robot tasarımının ve teknolojilerin tümü açık kaynak olarak paylaşılmaktadır, bu sayede bundan sonra geliştirilecek olan diğer bilimsel çalışmalarda da (makine görmesi, işitmesi, otonom seyrüsefer vb.) bu platformun kullanılması hedeflenmektedir.

Anahtar Sözcükler

1. senkron uzaktan eğitim
2. televarlık
3. robot
4. açık kaynak
5. öğrenci veri girişı



ABSTRACT

KASAP, Mahmut. Hybrid Education Robot Design For Distance Education, Master Thesis, Ankara, 2015.

Education studies show that distance education has made great contributions to lifelong education. BigBlueButton project is developed for synchronous distance learning. The project is open source and it widely used with success. Telepresence is defined as a system that provides a person who is in a distant place to have an ability of synchronous acting in local place through the technology [1]. Telepresence robot is defined as an object which has senses (sensors) for giving the sense of being in a distant place and has actuators (manipulators) for an ability of synchronous acting in local place [1]. Today, distance education and telepresence robot systems are separately available but there aren't any systems that are able to observe training activities in the local environment, use the high school or higher level educational materials, receive input from students. A telepresence robot designed for this purpose is expected to improve the attentional focus of students taking a distance education. Also, such a telepresence robot is necessary for the education in inaccessible regions and the home schooling of students with disabilities. Because for this there is a need for expert educators. Training of staff, employment in inaccessible regions and the expenses of staff require significant financial resources.

In this study, we design a mobile telepresence robot that can be used in home and school, which has an ability of sharing course material, sound and image with synchronous distance learning system, offering support students to input data from local media. This platform with easily implemented structure and minimal cost should make it possible to deliver equal opportunities and resources to necessary places in education, additionally eliminate the interaction problems of distant education. In this project all of the developed design and the technology of the robot is shared as open source. In this way it is targeted to be used this platform in other scientific studies (computer vision, voice recognition, autonomous navigation, etc.) which will be developed from now on.

Key Words

1. synchronous distance learning
2. telepresence
3. robot
4. open source
5. input from students



İÇİNDEKİLER

ÖNSÖZ.....	i
ÖZET.....	ii
ABSTRACT.....	iv
İÇİNDEKİLER	vi
SİMGELER VE KISALTMALAR	x
TABLolar	xii
ŞEKİLLER	xiii
GİRİŞ	1

BİRİNCİ BÖLÜM

HEduBot SİSTEMİNİN PLANLANMASI

1.1. İLGİLİ ÇALIŞMALAR	4
1.1.1. Senkron Uzaktan Eğitim Sistemi Olarak BigBlueButton.....	4
1.1.1.1. Kısa Tarihçe	4
1.1.1.2. BigBlueButton Teknik Özellikler	5
1.1.2. Mobil Televarlık Robotları İle İlgili Literatür	6
1.1.2.1. Mevcut Bazı Televarlık Robotların Teknik Özellikleri	13
1.1.2.2. Genel Olarak Mevcut Sistemlerin Ortak Özellikleri	33
1.1.2.3. Tasarımımızı Mevcut Sistemlerden Ayırıcı Özellikler	33
1.1.2.4. Televarlık Robotların Uygulama Alanları	33
1.2. AÇIK KAYNAK MOBİL TELEVARLIK İÇİN GEREKSİNİMLER.....	35
1.2.1. Mekanik Aksam Gereksinimleri	36
1.2.1.1. Yük Taşıma Kapasitesi.....	36
1.2.1.2. Yüksek Manevra Kabiliyeti	36
1.2.1.3. Kolay Uygulanabilirlik.....	36

1.2.1.4. Özelleştirilebilirlik	37
1.2.1.5. Düşük Maliyetlilik	37
1.2.2. Elektronik Aksam Gereksinimleri	37
1.2.2.1. Açık Kaynaklılık	37
1.2.2.2. Temin Edilebilirlik	38
1.2.2.3. Geniş Geliştirme Kitlesine Sahiplik	38
1.2.3. Yazılım Gereksinimleri	38
1.2.3.1. Açık Kaynaklılık	38
1.2.3.2. Temin Edilebilirlik	39
1.2.3.3. Düşük Maliyetlilik	39
1.2.3.4. Geniş Geliştirme Kitlesine Sahiplik	39
1.2.3.5. Verimlilik	39

İKİNCİ BÖLÜM

YÖNTEM

2.1. TASARIM	40
2.1.1. Mekanik	40
2.1.2. Elektronik	43
2.1.3. Sensörler	44
2.1.4. Elektronik Bileşenler	45
2.1.5. Yazılım	47
2.1.6. HEduBot Haberleşme Mimarisi	48
2.2. SINIF ORTAMINDA HEduBot	49

ÜÇÜNCÜ BÖLÜM













HEduBot TEKNİK ÖZELLİKLERİ










3.1. BOY VE AĞIRLIK	53
3.2. HIZ	54

3.3. ÇALIŞMA SÜRESİ	56
3.4. SENSÖR ÖZELLİKLERİ	58
GELECEK ÇALIŞMALAR.....	59
SONUÇ.....	60
TEŞEKKÜR	62
KAYNAKÇA	63
EKLER.....	71
8.1. GEREKLİ ARAÇLAR VE FİYATLARI	71
8.2. TEKNİK ÇİZİMLER	73
8.2.1. Mekanik Çizimler	73
8.2.2. Elektronik Çizimler.....	80
8.3. MONTAJ VE ÇALIŞTIRMA	81
8.3.1. Montaj.....	81
8.3.2. Çalıştırma.....	114
8.4. UYGULAMA KODLARI.....	122
8.4.1.Kullanıcı Uygulaması	122
8.4.1.1. Main.java.....	122
8.4.1.2. Client.java	124
8.4.1.3. Joystick.java.....	126
8.4.1.4. JoystickData.java.....	128
8.4.1.5. Reader.java.....	129
8.4.1.6. Writer.java.....	130
8.4.1.7. Constants.java.....	130
8.4.1.8. JoystickWindow.java.....	131
8.4.2. Robot Bilgisayar Uygulaması.....	139
8.4.2.1. Main.java.....	139

8.4.2.2. ArduinoMegaHelper.java.....	141
8.4.2.3. ArduinoUnoHelper.java	143
8.4.2.4. Dispatcher.java	144
8.4.2.5. Reader.java.....	146
8.4.2.6. Writer.java.....	146
8.4.2.7. Server.java.....	147
8.4.2.8. Constants.java.....	148
8.4.2.9. HedubotHelper.java	148
8.4.3. Robot Uygulaması	149
8.4.3.1. MotorKontrol.ino	149
8.4.3.2. SensorVeri.ino	152
ÖZGEÇMİŞ.....	156

SİMGELER VE KISALTMALAR

Simge	Açıklama
	Robot
	Yerel Kullanıcı
	Öğrenci
	Projektör
	Yazıcı
	PC
	Ultrasonik mesafe sensörü
	Ateş sensörü
	IR engel sensörü
	Isı ve nem sensörü
	Hareket sensörü
	Motor sürücü
	Sensör shield
	Arduino UNO
	Arduino Mega
	Motor

	LED
	Kamera
	Tablet
	LiPo batarya
	Robot PC
	İnternet
	Kullanıcı PC
	Joystick
	LCD
Kısaltma	Açıklama

MRP

Mobil Robotic Telepresence

ROS

Robot Operating System

TABLÖLAR

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri.....	28
Tablo 2: Hız hesaplaması deneylerinde elde edilen sonuçlar	55
Tablo 3: Sarf malzeme adet ve fiyatları	71
Tablo 4: Araç gereç adet ve fiyatları.....	73



ŞEKİLLER

Şekil 1: Tasarım ve uygulaması geliştirilen HEduBot'un bir görünümü	2
Şekil 2: BigBlueButton ekranından bir görüntü.....	6
Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler.....	7
Şekil 4: Tasarım için seçilen mekanik aksam bileşenleri.....	41
Şekil 5: Robot üzerinde fark sürüş sistemi.....	42
Şekil 6: Platform katmanları	42
Şekil 7: Tasarım için kullanılan elektronik aksam bileşenleri	44
Şekil 8: Sistemde kullanılan sensörler ve sensör değerleri test ekranı.....	45
Şekil 9: Sistemde kullanılan elektronik bileşenler	46
Şekil 10: HEduBot sisteminin genel bağlantı yapısı.....	47
Şekil 11: HEduBot haberleşme mimarisi.....	49
Şekil 12: HEduBot'un çalışmasını özetleyen diyagram	50
Şekil 13: Sınıf ortamında HEduBot	50
Şekil 14: HEduBot pilot kullanım testi	51
Şekil 15: Eğitimden ekranından bir görüntü	51
Şekil 16: HEduBot öğrenci kullanım testi.	52
Şekil 17: HEduBot ağırlık değerlerinin ölçümü.....	54
Şekil 18 HEduBot hız testi.....	56
Şekil 19: HEduBot'un farklı açılardan görünümü.....	61
Şekil 20: Tablet katmanı teknik çizimi	74
Şekil 21: Kart katmanı teknik çizimi	75
Şekil 22: Bilgisayar katmanı teknik çizimi	76
Şekil 23: Sensör ve ekran yan panelleri teknik çizimi	77
Şekil 24: Motor ve teker katmanı teknik çizimi.....	78
Şekil 25: Kamera katmanı ve pan tilt teknik çizimi.....	79
Şekil 26: Elektronik bağlantı şeması.....	80
Şekil 27: Lazer kesim için levhaların yerleştirilmesi.....	81
Şekil 28: Lazer kesim işleminin tamamlanması.....	82
Şekil 29: HEduBot için gerekli malzemelerin kontrolü.....	82

Şekil 30: Motor ve teker katmanının oluşturulmasında kart katı tutucu civata yerleşimi.....	83
Şekil 31: Motor tutucu montajı	84
Şekil 32: Motor ve teker katmanının oluşturulmasında oval masa gönyesi montajı .	84
Şekil 33: Bilgisayar katmanında oval masa gönyesi montajı.....	85
Şekil 34: Motor ve teker katmanının oluşturulmasında sarhoş teker montajı (a)	85
Şekil 35: Motor ve teker katmanının oluşturulmasında sarhoş teker montajı (b)	86
Şekil 36: Bilgisayar katmanı düz gönye montajı	86
Şekil 37: Motor ve teker katmanına motor tutucuların montajı (a)	87
Şekil 38: Motor ve teker katmanına motor tutucuların montajı (b)	87
Şekil 39: Motor ve teker katmanına motor tutucuların montajı (c)	88
Şekil 40: Plastik aralayıcıların kartlara montajı	88
Şekil 41: Kartların kart katına montajı.....	89
Şekil 42: Sensörlerin silikon ile yan panellere montajı (a)	89
Şekil 43: Sensörlerin silikon ile yan panellere montajı (b)	90
Şekil 44: Sensörlerin silikon ile yan panellere montajı (c)	90
Şekil 45: Sensörlerin silikon ile yan panellere montajı (d)	91
Şekil 46: Sensörlerin silikon ile yan panellere montajı (e)	91
Şekil 47: IR sensörlerin montajı.....	92
Şekil 48: Teker flanş montajı	92
Şekil 49: Teker motor montajı	93
Şekil 50: Motor ve teker katmanının tamamlanmış hali	93
Şekil 51: Isı ve nem sensörü montajı	94
Şekil 52: LiPo güç kablosu montajı	94
Şekil 53: Açma kapama düğmesinin hazırlanışı (a).....	95
Şekil 54: Açma kapama düğmesinin hazırlanışı (b)	95
Şekil 55: Açma kapama düğmesinin hazırlanışı (c).....	96
Şekil 56: Reset düğmesi	96
Şekil 57: Kart katının motor teker katmanına montajı.....	97
Şekil 58: Step down voltaj regülatör kartının 5v'a ayarlanması.....	98
Şekil 59: Sensör montajı (a).....	98
Şekil 60: Sensör montajı (b).....	99

Şekil 61: Sensör montajı (c).....	99
Şekil 62: Sensör montajı (ç).....	100
Şekil 63: Sensör ve motor testi (a).....	100
Şekil 64: Sensör ve motor testi (b).....	101
Şekil 65: Sensör ve motor testi (c).....	101
Şekil 66: Hata düzeltme ve kontrol (a).....	102
Şekil 67: Hata düzeltme ve kontrol (b).....	102
Şekil 68: Bilgisayar katmanı montajı.....	103
Şekil 69: Pan tilt montajı (a).....	103
Şekil 70: Pan tilt montajı (b).....	104
Şekil 71: Servo başlık montajı (a).....	104
Şekil 72: Servo başlık montajı (b).....	105
Şekil 73: Servo başlık montajı (c).....	105
Şekil 74: Kamera montajı (a).....	106
Şekil 75: Kamera montajı (b).....	106
Şekil 76: Gijon ve kamera katmanı montajı.....	107
Şekil 77: Tablet katmanı montajı.....	108
Şekil 78: Artık Gijonların kesimi (a).....	108
Şekil 79: Artık gijonların kesimi (b).....	109
Şekil 80: Artık gijonların kesimi (c).....	109
Şekil 81: Artık gijonların kesiminin tamamlanması.....	110
Şekil 82: Tablet montajı.....	111
Şekil 83: Servo bağlantısı.....	112
Şekil 84: USB çoklayıcı montajı.....	113
Şekil 85: Tamamlanan HEduBot.....	114
Şekil 86: MotorKontrol.ino yüklemesi.....	115
Şekil 87: SensorVeri.ino yüklemesi.....	115
Şekil 88: Robot bilgisayarında "lib" klasörünün yerleşimi.....	116
Şekil 89: Robot bilgisayarında PATH ayarının yapılması.....	116
Şekil 90: VirtualBox uygulamasında BigBlueButton sunucunun çalıştırılması (a).....	117
Şekil 91: VirtualBox uygulamasında BigBlueButton sunucunun çalıştırılması (b).....	118
Şekil 92: BigBlueButton ekranı.....	119

Şekil 93: Kullanıcı bilgisayarında "lib" klasörünün yerleşimi.....	120
Şekil 94: Kullanıcı bilgisayarında PATH ayarlarının yapılması	120
Şekil 95: Kullanıcı bilgisayarında joystick ekranı	121



GİRİŞ

Robotların eğitimdeki olası uygulamalarını inceleyen çalışmalar mevcuttur [2,3]. Genel olarak robot teknolojilerinin, anthropomorphism (cansız varlıkların canlı olarak algılanması) sayesinde öğrencilerin derse olan ilgisini artırdığı ve aynı zamanda derse katılmalarına engel olacak fobilerini azalttığı düşünülmektedir [2]. Her ne kadar robot teknolojilerinin olası fonksiyonlarından bahsedilse de eğitimde robot kullanan çalışma sayısı çok azdır. Bu çalışmaların çoğu ilkökul öncesi çocukları hedef aldığından eğitime destek, sadece uzak kullanıcının sınıfa sesli görüntülü bağlantısı, sanal bir yüz veya el aracılığıyla derse katılması ile yapılmıştır. Bu yaklaşımlarda eksiklik sözel olmayan dersler için yüksek öğretimde etkili ders materyali paylaşımıdır. BigBlueButton ile beyaz tahta paylaşımı, masaüstü paylaşımı, sunum-doküman paylaşımı, aynı doküman üzerinde ortak çalışma, öğrenci ve eğitici arasında çoklu web kamera paylaşımı benzeri yetenekler eş zamanlı uzaktan eğitimde kullanılabilir. Çalışma kapsamında uzaktan eğitim için tasarlanmış BigBlueButton yazılımı ile mobil televarlık robot teknolojisi birleştirilmiş, açık kaynaklı (Tasarımın tamamı ayrıntılı olarak ekte paylaşılmıştır.) bir robot-kullanıcı iletişim uygulaması geliştirilerek faydalanıcıların kullanımına sunulmuştur. Aynı zamanda geliştirilen robot tasarımlarında görülen yerel kullanıcı veri girişi eksikliği için grafik tablet entegrasyonu (Şekil 9 (a)) ile çözüm önerisi sunulmuştur.

Bu çalışmada uzaktan eğitimde insan etkileşimi eksikliği ve buna bağlı motivasyon eksikliği sorunlarına bir çözüm önerisi olmak üzere HEduBot (Şekil 1) adlı uzaktan eğitim robotu tasarlanmıştır. HEduBot, uzak eğiticinin sınıfta bir öğretmenin yapabildiği şekilde sunum yapma ve öğrenciden cevap alma gibi temel etkileşimli ders işlemlerini, uzaktan yapabilmesi amacıyla tasarlanmaktadır. Uzak eğiticinin yüzü HEduBot ekranında sunulacak ve HEduBot'un sınıfta gezinmesi sağlanarak, uzaktan eğitimde bir türlü aşılamayan insan etkileşimi eksikliği giderilmeye çalışılacaktır. Öğrencilerin derse katılımını arttırmak amacıyla robot, öğrenci veri girişlerini alabilecek bir şekilde tasarlanmaktadır. Bu sayede öğrencinin televarlık robot ile etkileşime girmesi sağlanacak, derse katılımı teşvik edilecektir.



Şekil 1: Tasarım ve uygulaması geliştirilen HEduBot'un bir görünümü

BİRİNCİ BÖLÜM

HEduBot SİSTEMİNİN PLANLANMASI

Çalışmanın nihai hedefine ulaşması aşamasında aşağıdaki süreçler sırasıyla tamamlanmıştır.

- **Ön çalışmalar:** Alanyazında incelemeler yapılarak proje isterlerinin (requirements) belirlenmesi ve uygun çözümlerin araştırılması.
- **Platform mekanik yapısının tasarımı ve gerçekleştirilmesi:** Mekanik aksamda kullanılacak parçaların belirlenmesi, temini, yapının teknik çizimlerinin yapılması, teknik çizim üzerinde sanal montajların gerçekleştirilmesi, sonrasında teknik çizimde belirtilen parçaların CNC veya lazer ile kesilmesi ve son montajı.
- **Platform elektronik yapısının tasarımı ve gerçekleştirilmesi:** Platformda kullanılacak elektronik bileşenlerin belirlenmesi, bağlantı şemalarının oluşturulması, temini ve bileşenlerin mekanik aksama montajı.
- **Haberleşme sistemi tasarımı ve programlanması:** İnternet üzerinden kullanıcı-robot ve robot-kullanıcı arası bilgi aktarımı sağlanması amacıyla, sunucu ve istemci mimarisine sahip özel bir yazılımın tasarımı, uygun protokollerin oluşturulması ve gerçekleştirilmesi.
- **Yönlendirme sistemi tasarımı ve programlanması:** Robot ve kamera sistemlerinin yönlendirilmesi ve sürüş sistemlerinin tasarımı. Bu sistemlerin kullanıcıların tam ihtiyacını karşılayacak şekilde, gürbüz ve kolay kullanılabilir hale getirilmesi.
- **Veri giriş sisteminin belirlenmesi ve platforma entegrasyonu:** Veri girişi amacıyla kullanılacak sistemlerin belirlenmesi, temini ve montajı.
- **Uzaktan eğitim sisteminin projeye entegrasyonu:** Projede senkron uzaktan eğitim sistemi kullanılması. Kullanılacak olan sistemin sunucusunun hazırlanması ve haberleşme yapısının belirlenmesi.
- **Testler ve geribildirimsel iyileştirmeler:** Geliştirilen platformun testlerinin kullanıcılar ile gerçekleştirilerek kullanıcı dönüşlerine göre iyileştirmeler yapılması. Kullanıcı memnuniyetinin sağlanması.

- **Dokümantasyon ve web sayfası:** Proje dokümantasyonu düzenlenerek bunun web üzerinden yayımlanması. Projenin ayrıntılı dokümantasyonu, kullanıcı kitlesinin geniş olmasını sağlayacaktır. Proje bu sayede geliştirme ve eklenti kitlesine de sahip olacaktır.

1.1. İLGİLİ ÇALIŞMALAR

Eş zamanlı uzaktan eğitim yazılımları içerisinde BigBlueButton, sadece açık kaynak olması sebebi ile değil, üzerine kurulduğu platform, kullandığı diğer üçüncü parti uygulamaların açık kaynak olması ve geniş topluluk desteği ile öne çıkmaktadır. 1.1.1. Senkron Uzaktan Eğitim Sistemi Olarak BigBlueButton başlığı altında bu uygulamaya genel bir bakış yapılması amaçlanmıştır.

Hem araştırma hem de ticari amaçlarla gerçekleştirilmiş televarlık robot tasarımları mevcuttur. Kullanıldığı alanları göstermek ve uzaktan eğitim amacıyla gerçekleştirmek istediğimiz robot tasarımı için gereksinimleri netleştirmek için bu tasarımların bir kısmı 1.1.2 Mobil Televarlık Robotları İle İlgili Literatür başlığı altında incelenecektir.

1.1.1. Senkron Uzaktan Eğitim Sistemi Olarak BigBlueButton

BigBlueButton projesi 2013 yılında Mozilla tarafından ödüllendirilmiş [4], gelişimi açısından örnek bir açık kaynak projedir. Projenin tarihçesi ve yapısı özet olarak şu şekildedir;

1.1.1.1. Kısa Tarihçe [5]

BigBlueButton projesi açık kaynak eş zamanlı çevrimiçi eğitim için 2007 yılında Carleton Üniversitesinde Dr. Tony Bailetti tarafından başlatılan bir projedir.

Dr. Bailetti'nin yöneticisi olduđu Teknoloji ve İnovasyon programı bu tarihe kadar eğitim amaçlı ticari bir uygulama kullanmıştır.

Dr. Bailetti'nin öğrencisi olan Richard Alam tez çalışması olarak bu konuyu ele almıştır. Red5 (açık kaynaklı medya sunucusu) projesi üzerinde BigBlueButton gerçekleştirilmeye başlanmıştır bu aşamada diğerk açık kaynak ve büyük projeler incelenmiş topluluđu genişletme amacıyla uygun stratejiler geliştirilmiştir.

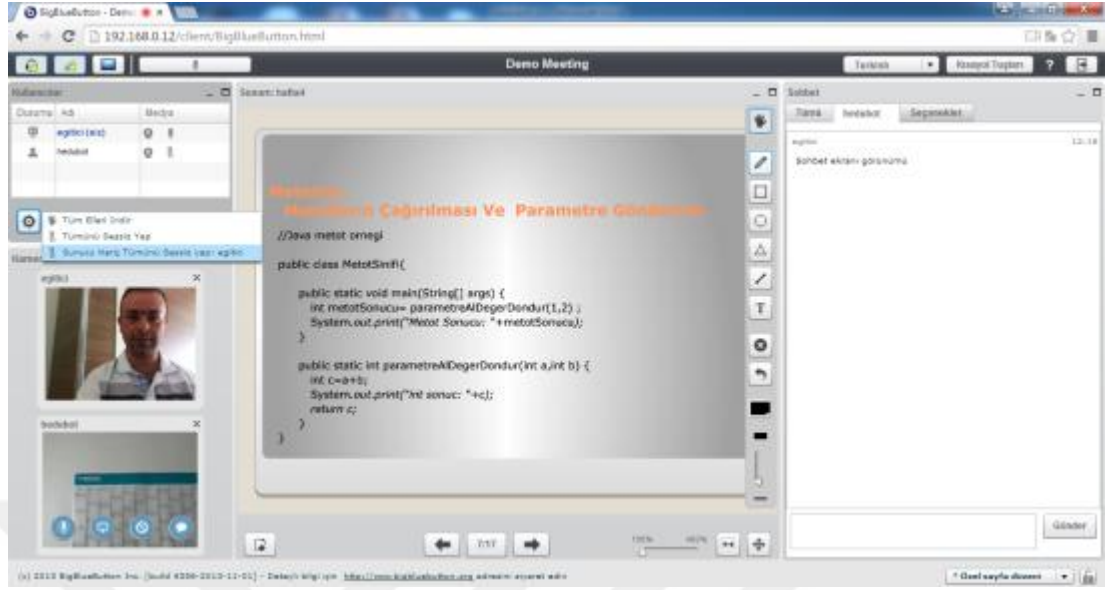
Proje büyümeye başlayınca topluluđa ve kullananlara teknik destek vermek amacıyla bir şirket kurulmuş ve tam zamanlı çalışanlar işe alınmıştır.

2008'de mesajlaşma, ses paylaşımı, sunum ve video paylaşımı ile başlayan proje, 2009'da masaüstü paylaşımı ve VoIP desteđi vermeye başlamış, 2010'da beyaz tahta desteđi eklenmiş, 2011'de kaydetme ve çalma desteđi sunulmaya başlanmıştır.

2012 Haziran ayı itibari ile BigBlueButton projesi 60000 den fazla indirilmiştir.

1.1.1.2. BigBlueButton Teknik Özellikler

Sınıf yönetimi sayesinde, kullanıcı oturumu sonlandırma, söz hakkı isteme, sunucu belirleme ve deđiştirme işlemleri yapılabilmektedir. Sunum sistemi ile farklı türlerde dokümanları sınıf ortamında sunulabilir, tablet ve benzeri araçlarla sunumların zenginleştirilmesi sağlanabilir. İki taraflı ve çoklu sesli ve görüntülü iletişim yeteneđi ve VoIP desteđi mevcuttur. Kolay kullanım için arayüz yerleşim düzenleme desteđi vardır. Yazılı sohbet, oturum tekrarlarını izleyebilme, beyaz tahta ve masaüstü paylaşımı özellikleri mevcuttur (Şekil 2) [6].



Şekil 2: BigBlueButton ekranından bir görüntü.

Şekil 2 de görülen BigBlueButton ekranında, sol üstte katılımcılar paneli bulunmaktadır; el kaldırma, sunucu değiştirme ve benzeri işlemler bu panelden yapılmaktadır. Sol altta kameralar ekranı görülmektedir; katılımcı kameraları burada yer alır. Ortada paylaşım ekranı bulunmaktadır; beyaz tahta, sunum, ofis dokümanları gibi belge paylaşımları buradan gerçekleşir. Sağda sohbet paneli yer almaktadır, yazılı sohbet buradan yapılmaktadır.

1.1.2. Mobil Televarlık Robotları İle İlgili Literatür

Televarlık robot; diğer ortamlarda olma hissini verebilmek için duyulara (algılayıcılara-sensörlere) ve o ortamda iş yapabilme kabiliyeti sağlayabilmek için uzuvlara (eyleyicilere-manipülatörlere) sahip bir televarlık olarak tanımlanmaktadır [1].

Mevcut gezgin robot televarlıklar ile ilgili 2012 yılında Kristoffersson A., Coradeschi S., ve Loutfi A. tarafından yapılmış bir literatür çalışması vardır [7]. Geçen süre zarfında faydalanılan bu çalışmada olmayan bazı yeni robotlar tasarlanmış, yine çalışmada bahsedilen robotların bir kısmı da güncellenmiştir.

Yapılan özetle bazı yeni robotlar ile mevcut robotların teknik bilgileri güncellenerek sunulmaktadır.

Şekil 3 Hata! Başvuru kaynağı bulunamadı. içerisinde robot tasarımlarının resimleri, devamında bu tasarımların kısa açıklamaları ve Tablo 1'de bu robotların teknik özellikleri verilmiştir.

Mevcut bazı tasarımlar ve bu tasarımların özetinde kullanılacak olan bazı özel terimler aşağıda verilmiştir.

- **Pilot kullanıcı:** Uzaktan bir bilgisayar arayüzü üzerinden robota bağlanan kişidir. Pilot insan bilgisayar etkileşimi ile robotu diğer insanların bulunduğu alanda hareket ettirebilir.
- **Yerel kullanıcı:** Robot ile aynı fiziksel konumda olan kullanıcı. Yerel kullanıcılar pilot ile etkileşim sırasında özgür hareket ederler pilot robot vasıtası ile onları ziyaret eder.
- **Yerel çevre:** Robot ile yerel kullanıcının bulunduğu ortam.



PRoP (a)



Griaaff (b)



QB (c)



Texai (ç)



Beam (d)



VGo (e)



PEBBLES (f)



MantaroBot
Classic (g)

Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler.



Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)



VGo (Güncel) (u)
[38]



Beam Pro (ü) [39]



Beam+ (v) [40]



MantaroBot Classic2
(y) [41]



MantaroBot
TeleMe2 (z) [41]



iRobot Ava 500 (a1)
[42]



Double (b1)
[43]



InTouch Vita (c1)
[45]



ROSCO (ç1)
[46]



Chesster ve Kasparov
(d1)[47]



RambleBot (e1)
[48]



Dennis Telepresence
Robot (f1)
[49]



My Telepresence
Robot (g1)
[50]



Caiba.net Telepresence
Robot in Oculus Rift
(ğ1)
[51]



MIDbot (h1)
[52]



Mandy (ı1)
[53]

Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)



TELESAR V(i1)
[54]



Roti (j1)
Telerobotics and [55]



**Simple Telepresence
Robot (k1)**
[56]



ROVIO (l1)
[57]



Teleance (m1)
[58]



TeRK (n1) [59]



Helios (o1) [60]



**LOUDEST
TELEPRESENCE
ROBOT EVER (ö1)**
[61]



**Taptic Toys iPad
robot (p1)**
[62]



E-One (r1) [63]



DORA (s1) [64]



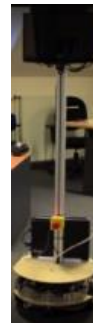
Sparky (ş1) [65]



RUDEBOT (t1)
[66]



MH-2 (u1) [67]



MITRO (ü1) [68]



ORGIBOT (v1) [69]

Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)



**A DIY
Telepresence
Robot (y1) [70]**



LUNA (z1) [71]



Anybots QA (a2) [72]



MG-500 (b2) [73]



R.BOT (c2) [74]



HOSPI-Rimo (ç2) [75]



**Low-Cost Video Chat
Robot (d2) [76]**



Telemba (e2) [77]



**MantaroBot
TableTop
TeleMe (f2)
[41]**



**Kubi (g2)
[78]**



**PadBot (ğ2)
[79]**



**iRobot ConnectR (h2)
[80]**



**OCULUS (i2)
[81]**



Oculus Prime (i2) [82]



RP2W (j2) [83]



PatrolBot (k2) [84]

Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)



Gridbots GSR-1
(12) [85]



RAGE (m2) [86]



**TELEPRESENCE
ROBOT (n2)** [87]



Webot (o2) [88]



**Smart
Telepresence
Robot (ö2)** [89]



**A mobile teleoperated
robot (p2)** [90]



MOBI (r2) [91]



TR-7 (s2) [92]



**Telepresence /
Social Robot (ş2)**
[93]



Robonaut 1 (t2) [94]



OLPC Telepresence
(u2) [95]



**Tabletop Telepresence
Robot (ü2)** [96]



Weelooms (v2)
[97]



**NIC ETS South
Telepresence Robot**
(y2) [98]

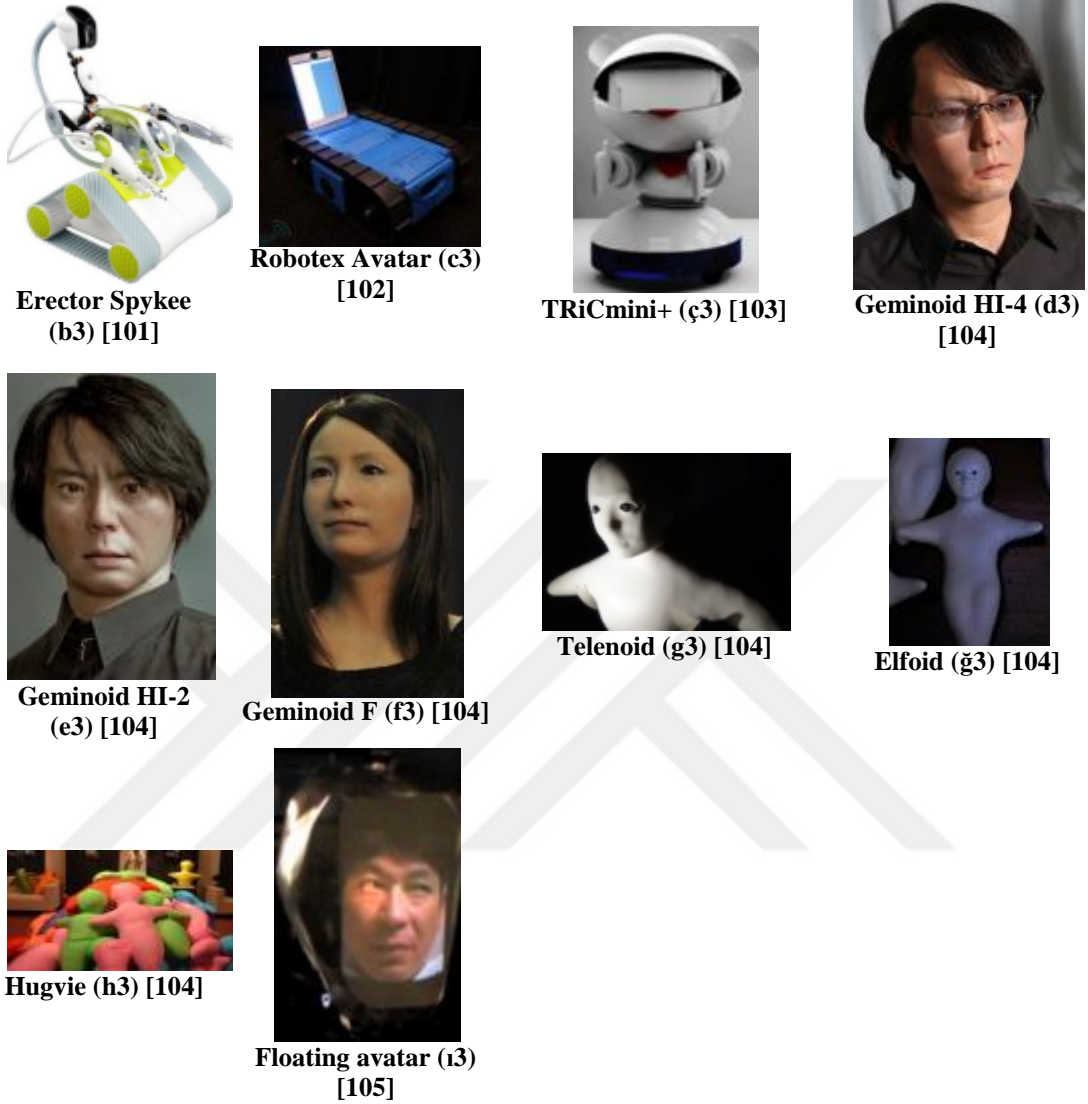


**MAYA Budget
Telepresence Rover**
(z2) [99]



MOBITON (a3) [100]

Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)



Şekil 3: Mevcut bazı televarlık robot tasarımlarına ait resimler (devamı)

1.1.2.1. Mevcut Bazı Televarlık Robotların Teknik Özellikleri

Bu bölümde önceki çalışmalarda görülen ve güncel araştırmalar ile tespit edilen mevcut birçok televarlık robotun teknik özellikleri özetlenmiştir.

1.1.2.1.1. PRoP [8,9]: 1.5 metre uzunluğundadır. Yönlendirmede joystick ve klavye kullanılır. Kamera, LCD monitör, 2 DOF robot kol ve el'in üzerinde lazer işaretçisi bulunmaktadır. Bu sayede elin konumu pilot tarafından görülebilir. Pilot

robotun kafa kısmı için pan tilt ve zoom hareketleri yapabilir. Günlük dosyası tutmaktadır (Şekil 3 (a)).

1.1.2.1.2. Giraff [10]: Ekranı 14.1" olan insan boylarında ve boynu eğilebilen bir MRP (Mobil Robotic Telepresence) sistemidir. Pilota yerel çevrenin geniş açı görüntüsü ulaştırılmaktadır. Ev ortamında çalışması için tasarlanmıştır. Eşik ve engelleri düşük risk ile geçebilir. Robot fare veya dokunmatik pad ile sürülebilir (Şekil 3 (b)).

1.1.2.1.3. QB [11]: Yüksekliği elle ayarlanabilir. 2 tekerlekli öz denge sistemine sahip bir platformu vardır. 5 mega pixel ön kamera ve yere bakan kamerası vardır. Lazer işaretçisi ile nokta tayini yapılır. Windows ve Machintosh işletim sistemlerinde herhangi bir tarayıcı üzerinden kumanda edilebilir [12] (Şekil 3 (c)).

1.1.2.1.4. Texai Protatip [14]: 19" dokunmatik ekran, pan tilt geniş açılı kamera, 2 lazer mesafe sensörü vardır. Polikarbon tampon mevcuttur. Tam şarjla sekiz saat çalışabilir. ROS işletim sistemi kullanılmaktadır. Ticari olarak sunulan konferans sistemine sahiptir. Willow Garage tarafından geliştirilmiştir [15] (Şekil 3 (ç)).

1.1.2.1.5. Beam [16]: 17" ekran, 6 adet mikrofondan oluşan ses yönü algılama dizisi, biri öne, biri navigasyon için yere bakan 2 adet yüksek çözünürlüklü dijital geniş açılı yakınlaştırmalı kamerası vardır. Şarjının tüm iş günü yeterli olduğu belirtilmiştir. Şarja takıldığını gösteren ledleri vardır. Sistem Windows ve Machintosh ile pilotaj olabilmektedir (Şekil 3 (d)).

1.1.2.1.6. VGo [17,18]: Windows veya Machintosh'da fare yardımı veya klavye ile pilotaj olabilmektedir. 5x zoom yapabilen kamerası vardır. Verizon 4G LTE ve WiFi desteklemektedir. Sabit yüksekliği ve 6 ve 12 saatlik iki pil seçeneği vardır (Şekil 3 (e)).

1.1.2.1.7. PEBBLES [19]: Okula gidemeyecek durumdaki çocuklar için dizayn edilmiştir. Çocuk yaşlarına göre iki tipi vardır. Büyük yaştaki çocukların grup olarak çalışabilmesi için çift ekranlı yapı vardır. Ekranlardan biri dosya paylaşımı için diğeri iletişim içindir. Pilot robot elini kumanda edebilmektedir (Şekil 3 (f)).

1.1.2.1.8. MantaroBot Classic ve MantaroBot TeleMe [20]: TeleMe tablet ve akıllı telefonların bir kısmının baş olarak kullanımına olanak sağlamaktadırlar. Pilot robotun baş kısmına pan tilt yaptırabilir, yüksekliği değiştirebilir. Lazer işaretçisi bulunmaktadır. Skype eklentisi kontrol için ve video konferans sistemi olarak kullanılmaktadır (Şekil 3 (g ve ğ)).

1.1.2.1.9. Jazz [21,22]: Web üzerinden ulaşılabilir. Telefon ve PC ile kullanımı arayüz sayesinde mümkündür. Üç boyutlu bir imleç ile tıkla ve git şeklinde yönetilmektedir. Engeller için ikaz sistemi vardır. Şarj istasyonuna otomatik yönelme için 2 m kadar yaklaşması gerekmektedir (Şekil 3 (h)).

1.1.2.1.10. iRobot Ava [23]: Robot üzerinde kafa olarak telefon, tablet ve monitör kullanılabilir. Şarj istasyonuna otomatik yönelme yapabilir, yüksekliği ayarlanabilir (Şekil 3 (ı)).

1.1.2.1.11. 9th Sense Helo ve 9th Sense Telo [24]: İki sisteminde Skype üzerinden bağlantı ve bilgisayar klavyesi ile kontrol özellikleri vardır. USB bağlantı noktaları ile çevre birimler eklenebilir. Helo daha kompakt bir tasarıma sahiptir. Kafa olarak Samsung Galaxy Tab kullanır (Şekil 3 (i ve j)).

1.1.2.1.12. Double [25]: Jiroskop ve ivme ölçer destekli özdenge sistemi vardır. Durunca açılan ayak standı sayesinde enerji tüketimi düşürülmüştür. Kafa ve pilot kumandası olarak iPad kullanılır, yüksekliği 100 ile 150 cm arasında değiştirilebilir (Şekil 3 (k)).

1.1.2.1.13. mObi [26]: Ballbot teknolojisi kullanan ilk ticari robot olarak tanıtılmaktadır ve Carnegie Mellon Üniversitesi Robotik Enstitüsü tarafından

geliştirilmiştir. Robot küre teker üzerinde durmaktadır. Etrafını 3 boyut olarak taramaktadır (Şekil 3 (l)).

1.1.2.1.14. RP-7 ve RP-VITA [27,28]: InTouchHealth firmasının sunduğu FDA (US Food and Drug Administration) onaylı sağlık alanında kullanılabilen sistemlerdir. RP-7 iki yüksek çözünürlüklü kamera ile video konferans sistemine sahiptir. RP-VITA 2012 yazında InTouchHealth firmasının iRobot firması ile birlikte açıkladığı, geliştirilmiş hareket sistemine sahip, gerçek zamanlı hasta bilgilerini aktarma kabiliyetine sahip, iPad ile hızlı ve kolay kullanılabilir bir sistemdir (Şekil 3 (m ve n)).

1.1.2.1.15. MeBot V4 [29]: Mebot V4 engelleri ve kenarları algılayabilir. Baş ve kolları 3 DOF hareket ettirebilir, baş hareketleri pilotun baş hareketleri ile yapılabilir. 3D fare ile robot kontrol edilebilir (Şekil 3 (o)).

1.1.2.1.16. iRobiQ [30]: 5 mimik yapabilir, video ve fotoğraf çekebilir, evdeki elektronik eşyalara bağlanabilir, telefonlara cevap verebilir. Çocuklarla şarkı söylemek, karaoke yapmak gibi aktiviteler yanında ev izleme, obje seçme, zaman ve hava durumu söyleme, hikaye anlatma gibi yazılımsal kabiliyetleri vardır (Şekil 3 (ö)).

1.1.2.1.17. Engkey [31,32]: Çok yönlü hareket edebilir. Ultrasonik mesafe ölçümü, 3 DOF kolu ve 2 DOF boynu vardır. İlkokul yaşındaki çocuklara derste uzak öğretici ile dil eğitimi amaçlı kullanılmaktadır. Eğiticinin yüz hareketlerini algılayarak robot üzerinde bulunan sanal karaktere mimik olarak uyarlayabilir (Şekil 3 (p)).

1.1.2.1.18. PeopleBot [33]: Robust P3-DX platformuna kurulmuştur. Dokunmatik ekranı vardır. Çeşitli objeleri tutmak için 2DOF el ve lazer sensörler, IR ve sonar sensörleri vardır. Sonar sensörler lazerden daha yakını takip etmek içindir. Sensörler sayesinde otonom hareket edebilmektedir (Şekil 3 (r)).

1.1.2.1.19. Botiful [34]: Android telefon ile birlikte çalışabilmektedir. 3 tekerli mini şasi üzerindedir. Telefonu yukarı aşağı hareket ettirebilme yeteneği vardır. Sykpe entegrasyonu yapılmıştır (Şekil 3 (s)).

1.1.2.1.20. Romo [35]: Apple tarafından üretilmiş 30-pin ve lighting portları olan 4. ve 5. nesil cihazlar ile çalışabilir. Tüm Apple cihazları ile kontrol edilebilir. Paletli mini şasi üzerindedir. Yüz takibi, eğitim görevleri, eylem programlama, nesne takibi, mimik, rota takibi yapabilir (Şekil 3 (ş)).

1.1.2.1.21. Lego Mind Storms Telepresence Robot [36]: Skype destekli bir tabletin Lego Mind Storms şasi üzerine eklenmesi ile yapılmış bir robottur. Renk ile kodlanmış eylemleri gerçekleyebilir (Şekil 3 (t)).

1.1.2.1.22. VGo (Güncel) [37,38]: Kaliteli fotoğraf çekimi için flaşlı kamerası otomatik veya manüel olarak tilt hareketi ve 5X zoom yapabilir Kullanımda değilken kamera aşağıya bakar. 6" ekranı vardır. 4 mikrofonu vardır. Düşük ışık düzeyi için ışığı vardır. Yerel kontrol için dokunmatik pad vardır. Gömülü kablosuz ağa sahiptir. Ayrıca hoparlörü vardır. Güvenlik için önde sensörler bulunmaktadır (Şekil 3 (u)).

1.1.2.1.23. Beam Pro [39]: 17" ekran, 6 adet mikrofondan oluşan ses yönü algılama dizisi, biri öne, biri navigasyon için yere bakan 2 adet geniş açı kamerası vardır. Sekiz saat çalışma süresi vardır. İki kanal ağ bağlantısı vardır ve seçmeli 4G kart yuvası vardır. Şarj istasyonu ile şarj edilmektedir. Hızı 2mph (~0.8m/sn) dir (Şekil 3 (ü)).

1.1.2.1.24. Beam+ [40]: 10" ekran, 4 adet mikrofondan oluşan ses yönü algılama dizisi, biri 2 adet HDR kamerası vardır. İki saat arama süresi vardır. 2.4 veya 5 GHz kablosuz ağ bağlantısı seçeneği vardır. Şarj istasyonu ile şarj edilmektedir. Hızı 2mph (~0.8m/sn) dir (Şekil 3 (v)).

1.1.2.1.25. MantaroBot TableTop TeleMe, Classic2 ve TeleMe2 [41]:

MantaroBot TableTop TeleMe masaüstü telepresence robottur. Üzerinde Apple iPad, iPhone veya Samsung Galaxy TAB kullanılabilir ve lazer işaretçisine sahiptir (Şekil 3 (f2)). Classic2 ile pilot robotun baş kısmına bulunan kamerayı pan tilt yaptırabilir, baş kısmında dokunmatik bilgisayar veya tablet seçeneği sunulmaktadır, yüksekliği değiştirebilir. Lazer işaretçisi bulunmaktadır. Skype eklentisi kontrol için ve video konferans sistemi olarak kullanılmaktadır (Şekil 3 (y)). MantaroBot TeleMe 2 ile Apple iPad, iPhone, Android veya Windows tablet kullanılabilir. TeleMe 2 ile Skype veya FaceTime, Google Hangouts, Cisco, Polycom, Vidyo, benzeri konferans sistemleri de kullanılabilir (Şekil 3 (z)).

1.1.2.1.26. iRobot Ava 500 [42]:

Cisco tarafından sunulan kurumsal sınıf telepresence sistemi vardır. Cisco TelePresence EX60 1080p30 çözünürlükte, 6Mbps hızında, H.323 / SIP standart üstünde video konferans sağlamaktadır. HTTPS ile şifrelenerek kurumsal güvenlik sağlanır. Gömülü Cisco Aironet 1600 kablosuz erişim noktası sayesinde kesintisiz ve kurumsal sınıf bir ağ bağlantısı sağlar. Cloud Service yardımıyla robot hakkında tüm bilgiler görüntülenebilir. iPad ve iPhone ile kumanda edilmektedir (Şekil 3 (a1)).

1.1.2.1.27. Double (Güncel) [43]:

Jiroskop ve ivme ölçer destekli özdenge sistemi vardır. Durunca açılan ayak standı sayesinde enerji tüketimi düşürülmüştür. Kafa olarak iPad Air 2 kullanılır. Pilot kumandası olarak iPad, iPhone, Mac veya Windows PC kullanılır. Mac ve PC için Google Chrome veya Firefox tarayıcı gereklidir. Apple donanımları için iOS 7 veya iOS 8 gereklidir, şarj için isteğe bağlı istasyon alınabilir. Yüksekliği uzaktan ayarlanabilir. Arkayı ve alt tarafı kamera yardımı ile gösterebilecek bir yapısı vardır (Şekil 3 (b1)).

1.1.2.1.28. InTouch Vita [44,45]:

InTouchHealth firmasının sunduğu FDA (US Food and Drug Administration) onaylı sağlık alanında kullanılabilen sistemdir. RP-VITA; AutoDrive özelliği sayesinde direk kullanım veya otonom hareket kabiliyetine sahiptir. iRobot firmasına ait iRobot Ava platformunda, geliştirilebilir bir tele sağlık sistemidir (Şekil 3 (c1)).

1.1.2.1.29. ROSCO [46]: İnternet tarayıcısı ile kullanılabilen, 3 boyutlu yazıcı ile parçaları hazırlanabilen, Arduino ve Adafruit motor sürücü ile hazırlanmış, Samsung akıllı telefon ile kullanılan, düşük maliyetli bir robottur (Şekil 3 (ç1)).

1.1.2.1.30. Chesster ve Kasparov [47]: Avustralya ulusal müzesi için hazırlanmış, yarı otomatik mobil bir robottur. Çok yönlü kamera ile internet üzerinden müze eğitimi, galeri ve panoramik video sağlayabilir. Standart bir bilgisayar, web kamera ve ağ yardımı ile öğrencilere video sohbet sağlar. Bu sayede öğrenciler eğitici ile etkileşime geçebilir. Bir robota birden fazla öğrenci bağlanabilir (Şekil 3 (d1)).

1.1.2.1.31. RambleBot [48]: Sağlam alüminyum şasi üzerine lastik palet kullanılarak yapılmış, çeşitli arazi ve ev şartlarına uygun bir platformdur. Tek şarj ile 24-48 saat arası çalışabilir. Akıllı telefon yuvası vardır. Robot kol sistemi bulunmaktadır. Tarayıcı üzerinden kontrol edilebilir. Görüşmeler için Skype kullanır (Şekil 3 (e1)).

1.1.2.1.32. Dennis Telepresence Robot [49]: Bir tank şasisi üzerine ters mühendislik uygulanarak servo motor sürücüler ile inşa edilmiş robot platformudur (Şekil 3 (f1)).

1.1.2.1.33. My Telepresence Robot [50]: Nexus 7 tablet, Arduino UNO ile hazırlanmış bir robottur. Google+ Hangouts ile ses ve video paylaşımı yapmaktadır. Ni-Mh batarya kullanılmıştır (Şekil 3 (g1)).

1.1.2.1.34. Caiba.net Telepresence Robot in Oculus Rift [51]: Arduino ve mini Windows PC kullanan platform üzerindeki kameranın pan-tilt hareketini kullanıcı, Oculus Rift gözlük sayesinde yaptırabilir. Pil ömrü yaklaşık 3 saattir ve bir oyun kumandası kullanarak hareket ettirilebilir. Fiyatı yaklaşık 500 dolardır (Oculus Rift hariç fiyat verilmiştir.) (Şekil 3 (ğ1)).

1.1.2.1.35. MIDbot [52]: Skype veya benzeri bir sohbet uygulamasıyla kullanılabilir ultra düşük maliyetli bir robot platformudur. Üstüne 1.5 kiloya kadar bilgisayar yerleştirilir. Cisim tutmak için kol mevcuttur. Platformun arkası alçalış yüksek olabilir (Şekil 3 (h1)).

1.1.2.1.36. Mandy [53]: Android tabanlı robot kontrol sistemi, Android Nexus 7 üzerine kurulu kullanıcı platformu, 3 teker KiWi sürüş sistemine sahiptir. Omni teker kullanılmaktadır. Google Hangout Extension ile kontrol edilmektedir (Şekil 3 (i1)).

1.1.2.1.37. TELESAR V [54]: İnsansı televarlık robottur. Robotun gördüğü, duyduğu pilota iletilir. Aynı zamanda robotun tuttuğu cisimleri ve sıcaklıklarını da pilot hissedebilir (Şekil 3 (i1)).

1.1.2.1.38. Roti [55]: Nesneleri ve engelleri algılaması için ultrasonik sensörler vardır. Monitör, ses sistemi ve fark sürüş sistemine sahiptir. Uzak eğitici derse bu robot yardımıyla görüntülü katılabilmektedir (Şekil 3 (j1)).

1.1.2.1.39. Simple Telepresence Robot [56]: Üzerinde Kinect kamera ve sensörü olan, VNC sayesinde uzak bağlantı yapılabilen, Max/Rex adında seri olarak kontrol edilebilen bir robot platformu üzerine kurulu televarlık robottur. Üzerinde mikrofon dizisi bulunmaktadır (Şekil 3 (k1)).

1.1.2.1.40. ROVIO [57]: Tarayıcıdan kontrol edilebilen, evde hareket noktaları işaretlenerek otonom hareket edebilen, otonom şarja yönelebilen, çok yönlü hareket sistemine sahip, LED aydınlatmalı bir televarlık robot sistemidir (Şekil 3 (l1)).

1.1.2.1.41. Telegance [58]: Arduino Mega kart ve Mac Mini elektronik altyapısına sahip, Skype video konferans sistemi kullanan ve kamera pan-tilt hareketi yapabilen bir televarlık robottur (Şekil 3 (m1)).

1.1.2.1.42. Telepresence Robot Kit (TeRK) [59]: Eğitim amaçlı, geniş geliştirme kitlesine ve geliştirme için hazır parçalara sahip, hobi ve ev kullanımına uygun açık kaynaklı yazılım kütüphanesi olan bir sistemdir. Sistem 2010 dan itibaren VEXPro adıyla ticari olarak sunulmaktadır (Şekil 3 (n1)).

1.1.2.1.43. Helios [60]: iPhone ile kullanılabilen bir televarlık robottur. Fark sürüşü sistemine sahiptir. Telefonu yukarı aşağı hareket ettirebilir (Şekil 3 (o1)).

1.1.2.1.44. LOUDEST TELEPRESENCE ROBOT EVER [61]: Ucuz tasarım amaçlanmıştır. Arduino, relay ve iki adet matkap ile ahşap şasi üzerine inşa edilmiştir. Skype kullanılmaktadır (Şekil 3 (ö1)).

1.1.2.1.45. Taptic Toys iPad robot [62]: Segway ile iPad 2 kullanılarak hazırlanmış bir televarlık robottur. FaceTime ile haberleşme imkanı sağlar (Şekil 3 (p1)).

1.1.2.1.46. E-One [63]: Sanal dadı olması için tasarlanmış robot platformudur. Tek şarjla 4 saat çalışabilir. Sekiz ultrasonik mesafe sensörü vardır (Şekil 3 (r1)).

1.1.2.1.47. DORA (Dexterous Observational Roving Automaton) [64]: Kullanıcı 6 serbestlik derecesi ile kafa hareketlerini Oculus Rift gözlük sayesinde yaptırabilir (Şekil 3 (s1)).

1.1.2.1.48. Sparky [65]: Kendin yap felsefesi ile geliştirilmiş bir robottur. 2 versiyonu vardır. Güncel versiyonunda şasi için Vex parçaları kullanılmıştır. 12V 7Ah batarya mevcuttur. Mac Mini ile yönetilmektedir. Bilgisayar ile servo motor kontrolü için elektronik kart kullanılmaktadır. Skype konferans sistemi kullanılmaktadır (Şekil 3 (ş1)).

1.1.2.1.49. RUDEBOT [66]: 3A motorlar ile tek şarjla 15 saat kullanılabilir. Mac Mini ve motor kontrol kartı bulunmaktadır. Ahşap şasisi vardır (Şekil 3 (t1)).

1.1.2.1.50. MH-2 [67]: Omuz üstünde taşınan teller ve servolar ile kontrol edilen bir robottur. Kollarda yedi serbestlik dereceli, kafa da 3 serbestlik dereceli, gövde iki ve ekstra bir serbestlik derecesi ile nefes alma hareketini yapabilmektedir (Şekil 3 (u1)).

1.1.2.1.51. MITRO [68]: 6" tekerler ile fark sürüş sistemine sahiptir. Lazer mesafe sensörü vardır. 14" LCD ekranı vardır. 160 cm yüksekliğindedir. HD web kamerası ve hareket kabiliyetini arttırmak için fisheye kamera vardır. Kinect sensör ve kamerası vardır. Tam otonom hareket kabiliyetine sahiptir (Şekil 3 (ü1)).

1.1.2.1.52. ORIGIBOT [69]: Açık kaynaklıdır. Arduino ile kontrol edilen 5 serbestlik derecesinde hareket ettirilebilen bir şasiye sahiptir. PC, Android ve iOS ile kontrol edilebilir. Üzerinde Android tablet vardır. İki parmaklı robot kola sahip bir platformdur. 4 saat kullanılabilir (Şekil 3 (v1)).

1.1.2.1.53. A DIY Telepresence Robot [70]: Kendin yap felsefesi ile hazırlanmış bir robottur. Tripod üzerine notebook eklenmiş ahşap bir şasi, 2 adet 12V DC motor ile sürülmektedir. Skype kullanmaktadır (Şekil 3 (y1)).

1.1.2.1.54. Luna [71]: RoboDynamics firması tarafından üretilen 5'2" boyunda robottur. 8" LCD dokunmatik ekrana sahiptir. 3000 dolar fiyatı vardır. 2 GHz işlemcisi, nVidia9000M grafik işlemcisi, 8 GB yükseltilebilir belleği ve WiFi özelliği vardır. Ayrıca 3 mikrofonluk ses dizisi, 3D PrimeSense sensörü vardır. 4-8 saat arası çalışma süresi, 7 adet genişleme portunun her birinin yanında USB, 12V ve 5V güç yuvaları vardır. Üzeride LunaOS açık kaynak işletim sistemi ile Poly Linux ve ROS paketleri kullanılmaktadır. Taşıma kulpu bulunur (Şekil 3 (z1)).

1.1.2.1.55. Anybots QA [72]: 5 feet uzunluğunda, 5 megapiksel kamerası olan bir platformdur. 2009 yılı fiyatları 15000 dolar ile 25000 dolar arasındadır. Ağırlığı 35 pound'dur (Şekil 3 (a2)).

1.1.2.1.56. MG-500 [73]: Hızı 1.5 m/s den düşüktür. 10 saat veya 8 km şarj yeterliliği vardır. Bir adet 2D lazer, 5 adet ultrasonik sensör, balık göz PTZ kamera, dört adet pasif infrared dedektör ve opsiyonel mikrofonu vardır (Şekil 3 (b2)).

1.1.2.1.57. R.BOT [74]: 30 lbs'den düşük ağırlıktadır. Ekran olarak Android cihazlar kullanılabilir. Fiyatı 450 dolar olarak belirlenmiştir. Yüz mimiklerini tanımlayıp mimik yapabilmektedir. Bu şekilde şarkı söyleme, video oynatma gibi komutlar uygulayabilir. Saatte bir mil hız yapabilir (Şekil 3 (c2)).

1.1.2.1.58. HOSPI-Rimo [75]: Medikal kullanım için hazırlanmış bir robottur. 27 ultrasonik sensörü ile otonom olarak bina içinde konum değiştirebilir. HD kalite video chat yapılabilir (Şekil 3 (ç2)).

1.1.2.1.59. Low-Cost Video Chat Robot [76]: iRobot şasisi geri mühendislikle değiştirilerek hazırlanmış bir robottur. Maliyeti 500 dolar civarındadır. Kaynak kodları paylaşılmaktadır (Şekil 3 (d2)).

1.1.2.1.60. Telemba [77]: 170 dolar maliyetle Japonyalı robotik bilimciler tarafından geliştirilmiştir. Roomba temizlik robotu şasisi üzerine Android tablet yerleştirilerek oluşturulmuştur (Şekil 3 (e2)).

1.1.2.1.61. Kubi [78]: Masaüstü televarlık robottur. Üzerinde tablet kullanılabilir. Bluetooth yardımıyla Kubi ile tablet arasında bağlantı kurulur. Herhangi bir tarayıcı ve video uygulaması ile kubi app adlı uygulama sayesinde bağlanabilir veya kendi uygulaması olan one app adlı uygulamayı kullanan bir cihazla bağlantı sağlanabilir (Şekil 3 (g2)).

1.1.2.1.62. PadBot [79]: Ticari amaçlı geliştirilen bir projedir. Robotta baş olarak iPad veya Android Pad kullanabilir. Kullanıcılar hem Android hem de Apple tablet ve telefonlar ile robotu kullanabilirler. Önünde ve altında bulunan sensörler ile engel ve boşluk tanımlayabilir. İzinli bağlantılara otomatik cevap verebilir. Dakikada 44 metre gidebilir. 5 kg ağırlığındadır (Şekil 3 (ğ2)).

1.1.2.1.63. iRobot ConnectR [80]: iRobot firması tarafından geliştirilen ticari amaçlı bir sanal ziyaret robotudur (Şekil 3 (h2)).

1.1.2.1.64. OCULUS [81]: Xaxxon firması tarafından geliştirilen ticari bir çoklu kullanım platformudur. Üzerine taşınabilir bilgisayar bağlanarak sistem uzaktan kontrol edilebilir bir robot haline dönüşmektedir. Donanımı ve yazılımı açık kaynaklıdır. Tarayıcıdan ve akıllı telefonda kontrol edilebilir. Otomatik şarj istasyonu bağlantısı ile şarj olmaktadır. Ek aksesuarları vardır (Şekil 3 (i2)).

1.1.2.1.65. Oculus Prime [82]: Xaxxon firması tarafından geliştirilen ticari robottur. 4 tekerli şasiye sahiptir. Tarayıcıdan ve akıllı telefonda kontrol edilebilir. Otomatik şarj istasyonu bağlantısı ile şarj olmaktadır. Otonom navigasyon ve haritalama yapabilir. Açık kaynak kontrol yazılımı vardır. Odometri hesabını dahili gyro ve enkoder sistemi ile yapar. Farklı eklenti versiyonları vardır (Şekil 3 (i2)).

1.1.2.1.66. RP2W [83]: Üç farklı satış versiyonu bulunmaktadır. 2 çeker şasiye sahip özel yazılımı olan bir platformdur. Baz fiyatı 3,402.22 dolar olarak belirtilmiştir. Motor, enkoder, LED, PC, sensör ve şarj istasyonu paketleri eklenebilir (Şekil 3 (j2)).

1.1.2.1.67. PatrolBot [84]: Robot engelli kolluk kuvvetlerinin güvenlik hizmetlerine devam edebilmesi için tasarlanmıştır. Kolları vardır ve dört tekerli bir platform olarak kullanılabilir. Gelecekte askeri görevler için de kullanılması planlanmaktadır. Gözetleme, devriye görevleri, 911 çağrılarını yanıtlama gibi fonksiyonları gerçekleştirebileceği düşünülmektedir (Şekil 3 (k2)).

1.1.2.1.68. Gridbots GSR-1 [85]: Hindistan'ın ilk televarlık robotu olarak tanıtılmaktadır. Görsel SLAM ile navigasyon, geçiş noktalarını takip, otomatik şarja yönelme, 4 saat çalışma süresi, 30 kg taşıma kapasitesi olduğu belirtilmiştir (Şekil 3 (12)).

1.1.2.1.69. RAGE [86]: Remote Access Grid Explorer (RAGE) R&D Magazine tarafından 2002 R&D100 ödülünü almış bir televarlık robottur. Standart bilgisayarlar tarafından kullanılan açık kaynak bir video konferans sistemi kullanmaktadır. 8 ultrasonik mesafe sensörü vardır (Şekil 3 (m2)).

1.1.2.1.70. TELEPRESENCE ROBOT [87]: Elektrikli tekerlekli sandalye şasisi üzerinde bulunan bir sistemdir. Yön tespiti için elektronik pusula sistemi bulunmaktadır (Şekil 3 (n2)).

1.1.2.1.71. Webot [88]: Robot kontrol sayfası istenilen web sitesine konulabilen; Linux, Windows, Mac OS için kontrol yazılımı olan; 12 saate kadar çalışma süresi hesaplanmış bir platformdur (Şekil 3 (o2)).

1.1.2.1.72. Smart Telepresence Robot [89]: Engel ve çukur tanıma, otonom navigasyon ve otomatik şarja yönelme özellikleri olan ticari bir robot olduğu belirtilmektedir (Şekil 3 (ö2)).

1.1.2.1.73. A mobile teleoperated robot [90]: Scott Settembre adlı öğrencinin yüksek lisans projesi olarak tasarladığı bir televarlık robottur. İki kolu bulunmaktadır. Robot otonom davranışları otomatik olarak veya pilot kontrolü ile aktif hale gelecek şekilde planlanmıştır (Şekil 3 (p2)).

1.1.2.1.74. MOBI [91]: Graham Smith tarafından tasarlanmış bir televarlık sistemidir. Sanatçının sergisinde kullanılmıştır. Çocuklar için olan küçük bir versiyonu da MOBI jr. olarak adlandırılmıştır (Şekil 3 (r2)).

1.1.2.1.75. TR-7 [92]: Robot Android tablet kullanmaktadır. Açık kaynak altyapısı vardır. Pilot Linux bilgisayar ile robotu kontrol edebilir. Düşük maliyetli olması planlanmıştır (Şekil 3 (s2)).

1.1.2.1.76. Telepresence / Social Robot [93]: Robotta elektrikli tekerlekli sandalye motorları kullanılmıştır. Kinect, 2 adet ultrasonik, 3 adet IR sensör ve enkoder bulunmaktadır (Şekil 3 (ş2)).

1.1.2.1.77. Robonaut 1 [94]: Kırk üç serbestlik dereceli NASA tarafından geliştirilen bir robottur. Dokunuş hissini vermek için kuvvet ve dokunsal geri bildirim eldiveni vardır. İnsan gövdesi ve dört teker şasisi bulunmaktadır. Nesnelere takip etme ve pilot eylemlerini sezgisel olarak gerçekleştirme özelliklerine sahiptir (Şekil 3 (t2)).

1.1.2.1.78. OLPC Telepresence [95]: Her çocuk için bir bilgisayar projesi (OLPC) ile üretilen ucuz bilgisayar ve iRobot Create kullanılarak oluşturulmuş, düşük maliyetli bir robottur (Şekil 3 (u2)).

1.1.2.1.79. Tabletop Telepresence Robot [96]: Sparkfun Magician robot şasisi üzerine kurulmuş, Arduino, Adafruit motor sürücü ve Pololu Wixel ile hazırlanmış düşük maliyetli bir telepresence robottur. iPhone Facetime ile haberleşme sağlar (Şekil 3 (ü2)).

1.1.2.1.80. Weelooms [97]: Android ile kontrol edilen bir telepresence robottur. Dört tekerlekli şasisi vardır (Şekil 3 (v2)).

1.1.2.1.81. NIC ETS South Telepresence Robot [98]: Oval iki tekerlekli bir şasi üzerinde kamera tripodlu ile tablet kullanılarak oluşturulmuş bir robottur (Şekil 3 (y2)).

1.1.2.1.82. MAYA Budget Telepresence Rover [99]: RoboRealm yazılımı ile kontrol edilen Roomba şasisi üzerine kurulu ve video konferans için Skype kullanan bir robottur. Kol mekanizması vardır (Şekil 3 (z2)).

1.1.2.1.83. MOBITON [100]: Arduino Mega ADK kart, Atmega386 işlemci, 4 adet IR mesafe sensörü ile düşme önleyici yapıya sahiptir. 800mAh kapasiteli LiPo pili vardır. USB ile programlama geliştirme sürümü de mevcuttur (Şekil 3 (a3)).

1.1.2.1.84. Erector Spykee [101]: Video ve ses kaydetme özelliği olan bir platformdur. 9.6 V pil ile çalışmaktadır (Şekil 3 (b3)).

1.1.2.1.85. Robotex Avatar [102]: Temel modelinin fiyatı 299 dolar olarak tanıtımı yapılmıştır. iPad mini robot üzerindedir. iPad veya iPhone ile kontrol edilebilmektedir. Açık kaynak olacağı belirtilmektedir (Şekil 3 (c3)).

1.1.2.1.86. TRiCmini+ [103]: Robot üzerinde bluetooth haberleşme yeteneği olan bir Arduino kart ile tablet kullanılır. Çok yönlü hareket kabiliyeti olan üç adet teker bulunmaktadır. Robotun üç adet ultrasonik sensörü vardır. Saniyede 12 cm hızla ilerler ve üzerinde 12V LiFePO4 batarya bulunur. Ağırlığı 3.6 kg dır (Şekil 3 (ç3)).

1.1.2.1.87. Geminoid HI-4, Geminoid HI-2, Geminoid F, Telenoid, Elfoïd ve Hugvie [104]: Erkek görünümlü robot modelleri Hiroshi Ishiguro adlı insandan esinlenerek oluşturulmuş insansı telepresence robotlardır. HI-4 (Şekil 3 (d3)) 16, HI-2 (Şekil 3 (e3)) ise 50 serbestlik dereceli ve pnömatik aktüatörlüdür. Geminoid F (Şekil 3 (f3)) ise HI-2 ile aynı altyapıda fakat 12 serbestlik derecesine ve pnömatik aktüatöre sahip kadın görünümlü model robottur. Telenoid (Şekil 3 (g3)) minimal insan özelliklerine sahip robottur. Elfoïd (Şekil 3 (ğ3)) cep telefonu büyüklüğünde Telenoid benzeri minimal özelliklerde bir robottur. Hugvie (Şekil 3 (h3)) ise sarılmak ve tutmak için geliştirilmiş bir telepresence robottur.

1.1.2.1.88. Floating avatar [105]: Sony Bilgisayar Bilimleri Laboratuvarı tarafından geliştirilmiş. Keşif balonu şeklinde uçan bir telepresence robottur. Pilot yüzü balon üzerine yansıtılır (Şekil 3 (13)).

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri

Sistem Adı	Ticari	Uygulama	Manüplatör	Sensör	Referans
PROP	Hayır	Araştırma	Lazer 2DOF el kol	Yok	[8,9]
Giraff	Evet	Yaşlı Bakım	Yok	Yok	[10]
QB	Evet	Ofis	Lazer pointer	Platform stabilizasyonu için hareket sensörü	[11,12,13]
Texai	Hayır	Ofis	Yok	Lazer uzaklık sensörü, tampon	[14]
Beam	Evet	Ofis	Yok	Yok	[16]
VGo	Evet	Ofis	Yerel kullanım elliği	Engel çukur algılama sensörü	[13,17,18]
PEBBLES	Hayır	Eğitim	El	Yok	[19]
MantaroBot Classic	Evet	Ofis	Lazer pointer	Kızılötesi engel algılama, uç algılama, 3 akseloremetre	[20]
MantaroBot TeleMe	Evet	Ofis	Lazer pointer	Kızılötesi engel algılama, uç algılama, 3 akseloremetre	[20]
Jazz Connect	Evet	Ofis	Yok	Engel algılama, 8 ultrasonik sensör, 4 IR sensör ve otonom navigasyon için 30 m mesafeli telemetrik lazer (opsiyonel)	[21,22]
iRobot Ava	Evet	Sağlık	Var	Otonom navigasyon için lazer, sonar, 2D/3D resimleme, engel, çukur ve çarpma sensörlü tampon, çok yönlü hareket	[23]
9th sense Helo ve Telo	Evet	Belirsiz	Yok	Çevre birimleri bağlantı noktası var	[24]

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri (devamı)

Double	Evet	Belirsiz	Yok	Denge için jiroskop ve ivmeölçer	[25]
mObi	Hayır	Araştırma	Yok	3D tarama	[26]
RP7	Evet	Sağlık	Yok	Çok yönlü hareket. engel algılama,30'dan fazla kızılötesi sensör	[27,28]
MeBot	Hayır	Araştırma	3 DOF kol ve boyun	Engel ve çukur algılama	[29]
iRobiQ	Evet	Belirsiz	Kol	Belirsiz	[30]
Engkey	Evet	Eğitim	3 DOF kol,2 DOF boyun	Çok yönlü hareket, ultrasonik mesafe sensörü	[31]
PeopleBot	Evet	Araştırma	Robust P3-DX platform	2DOF IR, lazer, sonar, otonom navigasyon	[33]
Botiful	Evet	Sosyal	Yok	Samsung Galaxy telefon sensörleri	[34]
Romo	Evet	Sosyal	Yok	Apple iPhone telefon sensörleri	[35]
Lego Mind Storms Telepresence Robot	Hayır	Deneysel	Yok	Yok	[36]
VGo(Güncel)	Evet	Ofis	Yok	Engel algılama	[37]
Beam Pro	Evet	Ofis	Yok	Yok	[39]
Beam+	Evet	Ofis	Yok	Ses yönü algılama için mikrofon	[40]
MantaroBot Classic2	Evet	Ofis	Boyun, lazer işaretçisi	Kullanılan platform sensörleri	[41]
MantaroBot TeleMe2	Evet	Ofis	Yok	Kullanılan platform sensörleri	[41]
iRobot Ava 500	Evet	Ofis	Yok	iRobot Ava platformu sensörleri	[42]
Double	Evet	Ofis	Ayak standı	Jiroskop ve ivmeölçer	[43]
InTouch Vita	Evet	Sağlık	Yok	iRobot Ava platformu sensörleri	[44]
ROSCO	Hayır	Belirsiz	Yok	Yok	[46]
Chesster ve Kasparov	Hayır	Müze Gezisi	Yok	Engel tanıma	[47]

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri (devamı)

RambleBot	Evet	Belirsiz	Robot kol	Kullanılan platform sensörleri	[48]
Dennis Telepresence Robot	Hayır	Hobi	Yok	Yok	[49]
My Telepresence Robot	Hayır	Hobi	Yok	Ultrasonik mesafe sensörü	[50]
Caiba.net Telepresence Robot in Oculus Rift	Hayır	Hobi	Boyun	Yok	[51]
MIDbot	Evet	Belirsiz	Robot kol, yükseklik değiştirme	Yok	[52]
Mandy	Evet	Belirsiz	Yok	Kullanılan platform sensörleri	[53]
TELESAR V	Hayır	Araştırma	Robot kol	Dokunma ve sıcaklık	[54]
Roti	Evet	Eğitim	Yok	Ultrasonik mesafe sensörü	[55]
Simple Telepresence Robot	Hayır	Belirsiz	Yok	Kinect ve mikrofon dizisi	[56]
ROVIO	Evet	Belirsiz	Boyun	True track	[57]
Telegance	Hayır	Araştırma	Boyun	Yok	[58]
Telepresence Robot Kit (TeRK)	Evet	Belirsiz	Belirsiz	Belirsiz	[59]
Helios	Evet	Belirsiz	Boyun	Kullanılan platform sensörleri	[60]
LOUDEST TELEPRESENCE ROBOT EVER	Hayır	Hobi	Yok	Yok	[61]
Taptic Toys iPad robot	Hayır	Hobi	Yok	Kullanılan platform sensörleri	[62]
E-One	Evet	Çocuk Bakımı	Yok	Ultrasonik mesafe sensörü	[63]
DORA (Dexterous Observational Roving Automaton)	Hayır	Araştırma	Boyun	Yok	[64]
Sparky	Hayır	Hobi	Yok	Yok	[65]
RUDEBOT	Hayır	Hobi	Yok	Yok	[66]
MH-2	Hayır	Araştırma	Kollar, baş ve gövde	Yok	[67]
MITRO	Hayır	Araştırma	Yok	Lazer mesafe sensörü, Kinect	[68]

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri (devamı)

ORIGIBOT	Hayır	Hobi	Robot kol	Kullanılan platform sensörleri	[69]
A DIY Telepresence Robot	Hayır	Hobi	Yok	Yok	[70]
Luna	Evet	Belirsiz	Yok	3 lü mikrofon dizisi ve 3D PrimeSense sensörü	[71]
Anybots QA	Evet	Belirsiz	Bilinmiyor	Bilinmiyor	[72]
MG-500	Evet	Belirsiz	Yok	2D lazer, 5 adet ultrasonik sensör ve 4 pasif IR dedektör	[73]
R.BOT	Evet	Belirsiz	Boyun	Kullanılan platform sensörleri	[74]
HOSPI-Rimo	Evet	Medikal	Yok	Ultrasonik mesafe sensörü	[75]
Low-Cost Video Chat Robot	Hayır	Hobi	Yok	iRobot platform sensörleri	[76]
Telemba	Evet	Belirsiz	Yok	Roomba platform sensörleri	[77]
Kubi	Evet	Belirsiz	Boyun	Kullanılan platform sensörleri	[78]
MantaroBot TableTop TeleMe	Evet	Ofis	Boyun, lazer işaretçisi	Kullanılan platform sensörleri	[41]
PadBot	Evet	Belirsiz	Yok	Engel ve çukur algılama	[79]
iRobot ConnectR	Evet	Belirsiz	Yok	Yok	[80]
OCULUS	Evet	Belirsiz	Yok	Yok	[81]
Oculus Prime	Evet	Belirsiz	Yok	Gyro ve enkoder	[82]
RP2W	Evet	Belirsiz	Yok	Belirsiz	[83]
PatrolBot	Hayır	Güvenlik	Robot kol	Yok	[84]
Gridbots GSR-1	Evet	Belirsiz	Bilinmiyor	Bilinmiyor	[85]
RAGE	Hayır	Araştırma	Yok	8 adet ultrasonik mesafe sensör	[86]
TELEPRESENCE ROBOT	Hayır	Hobi	Yok	Elektronik pusula	[87]
Webot	Evet	Belirsiz	Yok	Bilinmiyor	[88]
Smart Telepresence Robot	Evet	Belirsiz	Yok	Engel ve çukur tanıma, otonom navigasyon	[89]
A mobile teleoperated robot	Hayır	Araştırma	Robot kol	Bilinmiyor	[90]

Tablo 1: Mevcut bazı televarlık robotlar ve özellikleri (devamı)

MOBI	Evet	Sanat	Yok	Bilinmiyor	[91]
TR-7	Hayır	Araştırma	Yok	Yok	[92]
Telepresence / Social Robot	Hayır	Araştırma	Yok	Kinect, 2 adet ultrasonik sensör, 3 adet IR sensör	[93]
Robonaut 1	Hayır	Araştırma	43 DOF robot kol, dokunsal geri bildirim eldiveni	Dokunuş kuvvet sensörleri	[94]
OLPC Telepresence	Hayır	Hobi	Yok	iRobot Create sensörleri	[95]
Tabletop Telepresence Robot	Hayır	Hobi	Yok	iPhone sensörleri	[96]
Weelooms	Ticari	Bilinmiyor	Yok	Android platform sensörleri	[97]
NIC ETS South Telepresence Robot	Hayır	Hobi	Yok	Yok	[98]
MAYA Budget Telepresence Rover	Hayır	Hobi	Yok	Roomba platform sensörleri	[99]
MOBITON	Evet	Belirsiz	Boyun	4 adet IR mesafe sensörü, düşme önleyici sensörler	[100]
Erector Spykee	Evet	Belirsiz	Belirsiz	Belirsiz	[101]
Robotex Avatar	Evet	Belirsiz	Yok	Kullanılan platform sensörleri	[102]
TRiCmini+	Hayır	Araştırma	Robot kol	3 ultrasonik sensör	[103]
Geminoid HI-4	Hayır	Araştırma	16 DOF aktüatör	Çeşitli sensörler	[104]
Geminoid HI-2	Hayır	Araştırma	50 DOF aktüatör	Çeşitli sensörler	[104]
Geminoid F	Hayır	Araştırma	50 DOF aktüatör	Çeşitli sensörler	[104]
Telenoid	Hayır	Araştırma	9 DOF aktüatör	Yok	[104]
Elfoid	Hayır	Araştırma	Yok	Yok	[104]
Hugvie	Hayır	Araştırma	Titreşim	Yok	[104]
Floating avatar	Hayır	Araştırma	Pervane	Bilinmiyor	[105]

1.1.2.2. Genel Olarak Mevcut Sistemlerin Ortak Özellikleri

İncelenen robotların ortak özellikleri şu şekilde sıralanabilir:

1. Manipülator olarak boyun, kol ve lazer pointer mekanizmaları kullanılmaktadır.
2. Sensör olarak IR, ultrasonik mesafe sensörleri, lazer, sonar, jiroskop, ivme ölçer ve dokunsal sensörler kullanılmıştır.
3. Şasi yapısı olarak mevcut sistemlerde çoğunlukla 2 tekerli özdenge sistemleri veya 3 tekerlekli hareket sistemleri tercih edilmektedir.

1.1.2.3. Tasarımımızı Mevcut Sistemlerden Ayırıcı Özellikler

Yapılan geniş kapsamlı alanyazın çalışmasına göre tasarımımızı diğer tasarımlardan ayıran başlıca özellikler şu şekilde sıralanabilir:

1. Açık kaynaklı, yapımı kolay ve düşük maliyetli olacaktır. Parça listesi, ayrıntılı montaj kılavuzu ve kaynak kodlar ekte paylaşılmıştır. Böylelikle isteyen herkes kendisi için bu sistemi kolaylıkla kurabilecektir. Her ne kadar açık kaynaklı televarlık robot projeleri bulunsa da, yükseköğretimde kullanılması amaçlanan, yurt içi üretimi gözetilen ve doküman paylaşımı yapabilen bir proje yoktur.
2. Mobil olarak yerel ortamdan veri girişi alınabilecek ve çok sayıda dosya türü üzerinde ortak çalışma yapılabilecektir.
3. Diğer sensörlere ek olarak; hareket sensörü, ateş sensörü, nem, sıcaklık sensörü bulunacaktır.
4. En yaygın kullanılan uzaktan eğitim sistemlerinden olan BigBlueButton ile birlikte çalışacaktır.

1.1.2.4. Televarlık Robotların Uygulama Alanları

Günümüzde televarlık robotların uygulamalarında görülen alanlar şu şekilde özetlenebilir.

1.1.2.4.1. Ofis Ortamları

Günümüzde şirketler için seyahat giderleri önemli bir masraf kalemi oluşturmaktadır. Aynı zamanda seyahat için tercih edilen yöntemlerin karbon salınımı nedeniyle çevreye büyük zararları olmaktadır. Özetle görülen robotların bir kısmı ofis ortamlarında test edilmiştir. HEduBot tüm bu robotlardan farklı olarak ofis ortamında doküman paylaşımına ve yerel kullanıcı veri girişine olanak vermektedir.

1.1.2.4.2. Sağlık Birimleri

Sağlık alanında kullanılan ve onay alan robotlar mevcuttur. Farklı alanlarda uzman doktorların sistem üzerinden birbirleri ile konsültasyon yaparak hastalara daha kısa zamanda daha iyi hizmet verebilecekleri kabul görmektedir.

1.1.2.4.3. Evde Yaşlı Bakımı

Yaşlı bakımında kullanılacak olan robotların diğer robotlara nazaran daha anlaşılır ve kolay kullanılabilir tasarlanması gerekmektedir. Yaşlı bakımında telepresence robotların, ilaç kullanımı, acil durum yardım sistemi, hastalık-durum takibi ve evde takip alanlarında kullanılması mümkündür. Hali hazırda yaşlı bakımı için kullanılmakta olan robot tasarımı mevcuttur.

1.1.2.4.4. Okullar

Hastaneden veya evden çıkamayacak durumda engelli çocuklar için kullanılan telepresence robotlar mevcuttur. Bunun yanında bazı telepresence robotlar çocuklar ve öğretmenler arası iletişimde çeşitli çalışmalarda denenmiştir. Özetle bu sistemler çocuklar ve öğretmenler arası ders iletişimde kullanılabilir. Günümüzde eğitim amacıyla kullanılan telepresence robotlar mevcuttur.

1.1.2.4.5. Genel Amaçlı Ortamlar

Açık kaynaklı robot kontrol uygulamaları ve farklı video konferans sistemleri ile genel amaçlı telepresence robotlar geliştirmek mümkündür. Bu televarlık sistemler farklı alanlarda değişikliğe gidilmeden kullanılabilirler. İncelenen robotlar içerisinde genel kullanıma uygun robotlar bulunmaktadır.

1.2. AÇIK KAYNAK MOBİL TELEVARLIK İÇİN GEREKSİNİMLER

Araştırmacıların, sanayi kuruluşlarının ve eğitimcilerin bu platformu inşa etmeleri ve kullanabilmeleri için tasarladığımız robot platformunun maliyetini düşük tutmak (Detaylı maliyet tablosu ekte verilmiştir.) ve yurt içinden temin edilebilir bileşenlerden oluşturmak başlıca amaçlarımız olmuştur.

Çalışmada kullanılmış olan tüm bileşenlerin açık kaynak olarak paylaşılması felsefesi sayesinde, platform yararlananların ihtiyacına göre özelleştirilebilir hale gelecektir. Platform bileşenlerinin tamamının yazılımı, mekanik tasarımı ve donanım özellikleri, çalışma sonunda kamuya açık şekilde sunulacaktır. Platformda kullanılacak mekanik aksamların teknik çizimleri, elektronik aksamların devre şemaları ve yazılımın kaynak kodu açık (Open Source) olacaktır (Yazılımın kaynak kodu ekte verilmiştir.). Robotun birleştirilmesi ve sistemin ayağa kaldırılması için ayrıntılı bir kullanım kılavuzu ekte sunulmuştur.

Geliştirme sürecinde ortaya çıkan prototip kullanıcı memnuniyeti için denenmelidir. Her deneme sonunda kullanıcıların görüşleri alınarak eksiklikler tamamlanmalı ve gerekirse bu eksiklikler için sürecin farklı adımları tekrar işletilmelidir. Bu tekrarlanan süreç sayesinde oluşturulacak platform iyileştirilecek ve sonunda nihai halini alacaktır.

Projemizin amacı doğrultusunda oluşturduğumuz robot platformunun isterlerini şu şekilde özetlenebilir:

1.2.1. Mekanik Aksam Gereksinimleri

Mekanik aksam isterleri yük taşıma, manevra, uygulama, özelleştirilebilirlik ve maliyet başlıkları altında incelenecektir.

1.2.1.1. Yük Taşıma Kapasitesi

Özelleştirilebilir bir platformun ne kadar yük taşıma kapasitesi olacağı önemlidir. Bu platformlar sadece sensör benzeri elektronik bileşenleri taşıyabildiği gibi eklenecek diğer bileşenler ile özelleştirilebilir olmalıdır. Bu yüzden yük konusunda esneklik sağlayacak bir platform tasarlanmalı, aynı zamanda büyüklüğe ve ağırlığa da dikkat edilmelidir. Çünkü iç ortamlarda kapılardan rahat geçiş gereksinimi ve düşük enerji tüketimi ihtiyacı vardır.

1.2.1.2. Yüksek Manevra Kabiliyeti

Robotun kullanıcı tarafından rahatlıkla yönlendirilebilmesi ve iç ortamlarda kapı ve eşiklerden geçebilmesi gerekmektedir. Aynı zamanda uygulanabilirliği ve diğer istelere uyum da göz önünde bulundurularak hareket mekanizması belirlenmelidir. Bilinen en yüksek manevra kabiliyeti çok yönlü hareket (Omni Wheel) sistemleri ile sağlanmaktadır. Fakat bu şekilde üretilecek olan bir robot platformunun kullanılabilirlik, taşıma kapasitesi ve de maliyet sorunları olacaktır. Alternatif olarak fark sürüşü adı verilen ve yine yüksek manevra kabiliyetine sahip maliyeti düşük sürüş sistemi mevcuttur.

1.2.1.3. Kolay Uygulanabilirlik

Robotun üretilmesindeki tüm basamakların yararlanıcının kendi imkanları ile gerçekleyebilmesi, montajının ağır teknik bilgi gerektirmemesi ve parçaların kolay

temini önemlidir. Türkiye'den temin edilebilen parçalarla robotun gerçekleştirilmesi aynı zamanda detay teknik bilgiye sahip olmayan araştırmacıların da eldeki şemalar ve teknik dokümanlar ile parçaları temin ederek yapım ve montajını gerçekleştirebilmesi amaçlanmıştır.

1.2.1.4. Özelleştirilebilirlik

Robotun şasisi üzerine yeni donanımlar eklenebilmesi amacıyla standart yuvalar ve delikler açılması robotun diğer sensör ve donanımlarla entegrasyonunu kolaylaştıracaktır. Bunun yanında robotun şasisinin biçimlendirilmesi kolay olmalıdır.

1.2.1.5. Düşük Maliyetlilik

Robot için alınacak olan mekanik aksamın dikkatle seçilmesi gerekmektedir. Bu seçim işleminde maliyetin düşük olmasının önemini yanında temin edilebilir olması da göz önünde bulundurulmuştur.

1.2.2. Elektronik Aksam Gereksinimleri

Elektronik aksam istekleri açık kaynaklılık, temin edilebilirlik ve geniş geliştirme kitlesine sahiplik başlıkları altında incelenecektir.

1.2.2.1. Açık Kaynaklılık

Kullanılan tüm elektronik kartların devre şemalarının açık olması önem arz etmektedir. Bu sayede platform konusunda araştırma geliştirme faaliyeti sürdüreceği olan yararlanıcılar devre yapılarını ihtiyaçlarına göre özelleştirebileceklerdir.

1.2.2.2. Temin Edilebilirlik

Kullanılan tüm elektronik aksamın hazır ve bileşen bazlı olarak Türkiye'den rahatlıkla temin edilebilir olması dikkate alınmıştır ve proje sonunda internet yansılarını üzerinden proje yararlanıcılarına bilgi verilecektir. Bunun yanında tercih sebebi olarak fiyatı uygun seçenekler değerlendirilmelidir.

1.2.2.3. Geniş Geliştirme Kitlesine Sahiplik

Kullanılan elektronik kartlar için internet üzerinde forumlar ve proje sitelerinde destek yeterliliği aranmıştır. Bu sayede yararlanıcılar, sorun yaşamaları halinde bu sitelerden destek alabilecekler ve çeşitli örnekler ile öğrenme ve geliştirme süreçlerini hızlandıracaklardır.

1.2.3. Yazılım Gereksinimleri

Yazılım istekleri açık kaynaklılık, temin edilebilirlik, düşük maliyetlilik, geniş geliştirme kitlesine sahiplik ve verimlilik başlıkları altında incelenecektir.

1.2.3.1. Açık Kaynaklılık

Geliştirilen tüm yazılımın, kullanılan tüm üçüncü parti kütüphanelerin kaynak kodlarının açık olmasına dikkat edilmiştir. Bu sayede programlama konusunda araştırma geliştirme faaliyeti sürdüren yararlanıcılar, kaynak kodlarını ihtiyaçlarına göre özelleştirebileceklerdir.

1.2.3.2. Temin Edilebilirlik

Proje sonunda geliştirilen yazılımın ve kullanılan tüm üçüncü parti kütüphanelerin internet üzerinden yansılar ile temin edilebilmesi sağlanacaktır.

1.2.3.3. Düşük Maliyetlilik

Kullanılan tüm üçüncü parti kütüphanelerin düşük maliyetli olmasına dikkat edilmiştir. Burada daha iyi bir seçenek olarak tamamıyla ücretsiz açık kaynak kütüphaneler öncelikli olarak tercih edilmiştir.

1.2.3.4. Geniş Geliştirme Kitlesine Sahiplik

Kullanılan yazılımlarda internet üzerinde forumlar ve proje sitelerinde destek yeterliliği aranmıştır. Bu sayede yararlanıcılar sorun yaşamaları halinde bu sitelerden destek alabilecekler ve çeşitli örnekler ile öğrenme ve geliştirme süreçlerini hızlandıracaklardır.

1.2.3.5. Verimlilik

Tasarlanmış olan yazılım mimarisi ile haberleşme hızının yüksek tutulması sağlanmıştır. Ağ üzerinden kullanıcı komutlarının gönderilmesi ve robot tarafından bu komutların işlenmesindeki gecikmeler minimum düzeyde tutulmalıdır. Bu yüzden gönderilen ve alınan veri paketleri gerekirse sıkıştırılacak ve paket boyutları belirli sınırlara tabi tutulacaktır.

İKİNCİ BÖLÜM

YÖNTEM

Uzaktan eğitimde karşılaşılan sorunlar alanyazında incelendiğinde, genellikle altyapı ve teknoloji eksikliği, yüksek maliyetler, kaliteli ders içeriği eksikliği ile ilgili tespitlerle karşılaşmaktadır [106]. Öte yandan ilginçtir ki bunların tamamlandığı kurumlarda da sorunlar devam etmektedir. Uzaktan eğitimde çok kapsamlı online, bilgisayar destekli ev veya sınıf ortamları kullanıldığında bile görülmektedir ki öğrenciler derslere katılmamakta, dersleri takip etmemektedirler. Bunun nedeni olarak da temel kişiler-arasılık (essential intersubjectivity) eksikliği, bir başka deyişle insan etkileşimi eksikliği gösterilebilir. İnsan etkileşimi eksik kaldığında, bütün gerekli bilgiyi, ders materyallerini ve öğretimsel süreçleri sunsanız bile öğrenciler o sınıfta veya o ortamda olmamayı tercih etmektedirler [107]. Bu konuda Peters O. [108]'de Rusçada uzaktan eğitim kavramında kullanılan “uzak” kelimesinin etimolojik kökenlerini incelediğinde “zaochny” kelimesine ulaşmış ve bu kelimenin anlamının “göz teması olmadan” anlamına geldiğini görmüştür. İnsan eksikliği ve/veya göz teması eksikliği uzaktan eğitimde yaşanan eğitimsel sorunların önemli bir kısmının kaynağını tek başına oluşturmaktadır.

2.1. TASARIM

Hazırlanan robot tasarımı mekanik, elektronik, sensörler, elektronik bileşenler ve yazılım olarak incelenecektir.

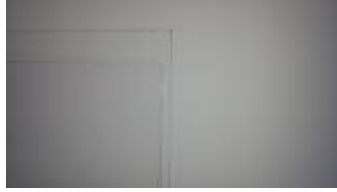
2.1.1. Mekanik

Mekanik tasarım için temini kolay ve montaja hazır bileşenler üzerinde durulmuştur. Platform şasisinin, mdf Levha (Şekil 4 (a)) ve pleksi Levha (Şekil 4 (b)) olması düşünülmüştür bu sayede şasi güçlü ve hafif olacaktır aynı zamanda bu levhaları CNC veya lazer ile işleme de kolaydır. Teker olarak 9 cm çaplı Pololu

teker düşünölmüştür (Şekil 4 (d)). Motor bağlantıları için ise sac motor tutucu (Şekil 4 (c)) ve alüminyum flanş düşünölmüştür (Şekil 4 (ç)). Katmanlar ve paneller arası bağlantı, oval masa gönyeleri, cıvatalar, plastik aralayıcılar, gijonlar ve kelepçeler ile (Şekil 4 (f)) sağlanmış, sonrasında üç boyutlu tasarım ortaya çıkarılmıştır.



Mdf Levha (a)



Pleksi Levha (b)



Motor Tutucu (c)



Flanş (ç)



9 cm Teker (d)



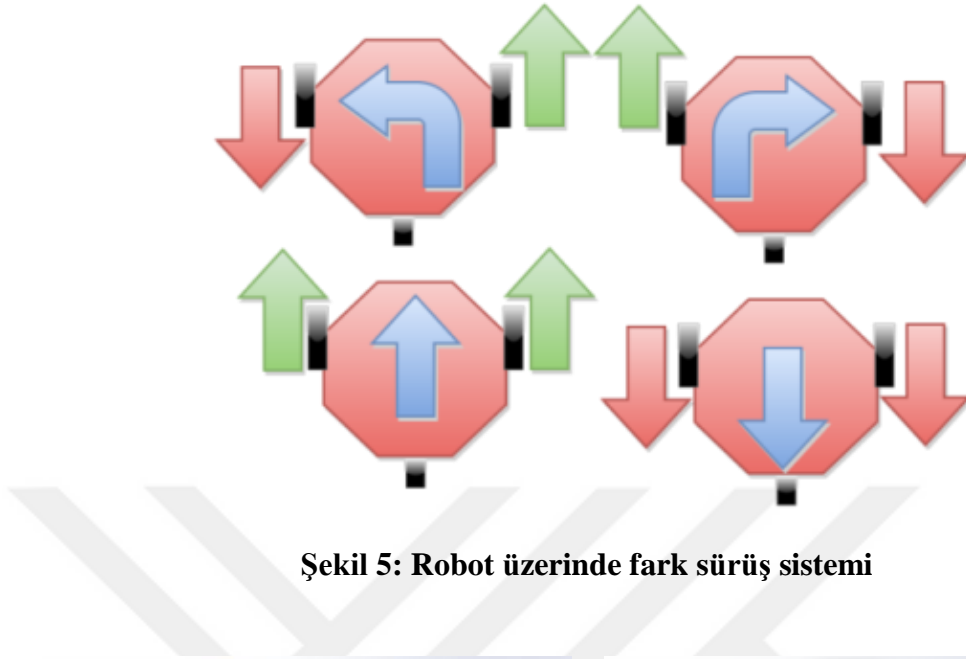
Sarhoş Teker (e)



Bağlantı elemanları (f)

Şekil 4: Tasarım için seçilen mekanik aksam bileşenleri

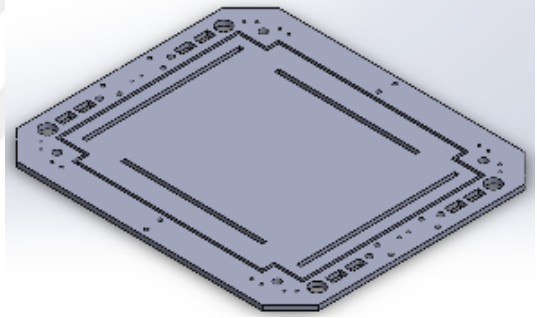
Sürüş sistemi olarak fark sürüşü sistemi, kolay uygulanabilir olması, düşük maliyeti, yüksek manevra ve yük taşıma özellikleri ile isterleri karşılamaktadır. Bu sistemde iki tekerin hız farkı, robotun yönünü belirlemektedir. Robotun dengede durması için ise bir adet sarhoş teker kullanılmaktadır (Şekil 5). Ekte kullanılan malzemelerin detaylı listesi mevcuttur.



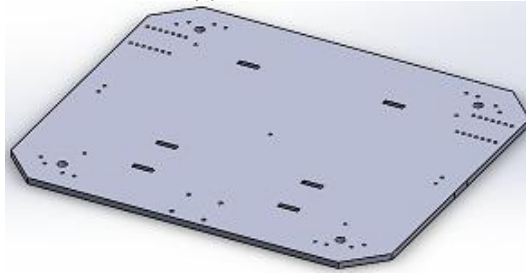
Şekil 5: Robot üzerinde fark sürüş sistemi



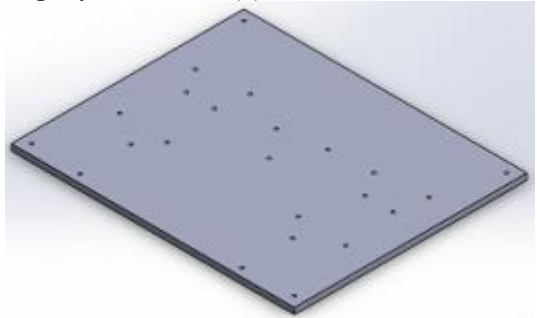
Tablet Katmanı (a)



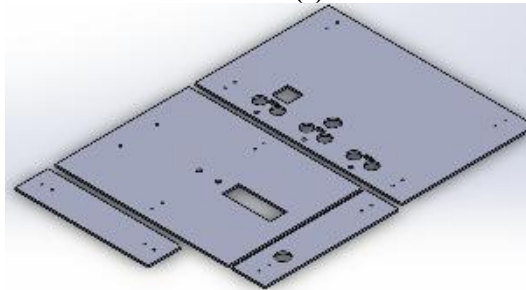
Bilgisayar Katmanı (b)



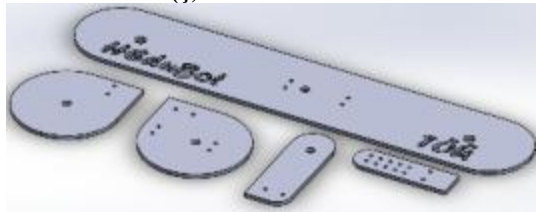
Motor ve Teker Katmanı (c)



Kart Katmanı (ç)



Sensör ve Ekran Yan Panelleri (d)



Kamera Katmanı ve Pan Tilt (e)

Şekil 6: Platform katmanları

Yapı olarak, çok katlı özelleştirilebilir yapı düşünülmüştür. Robotun çok katlı olması ileride eklenebilecek yeni donanımlara da izin verecek ve özelleştirilebilir olma isterini karşılayacaktır. Kamera Katmanı ve Pan Tilt 3mm, diğer tüm katmanlar 6mm kalınlığındadır ve gijon olarak tabir edilen 1m uzunluğunda dış açılmış M8 ebatlarında mil sayesinde birbirlerine istenilen aralıklarla yerleştirilebilir. Kat kesiminde CNC kullanılacak ise deliklerde 3mm'lik uç kullanılması gerekmektedir. Delikler içten, dış çevre ise dıştan kesilmelidir. Motor ve Teker Katmanı (Şekil 6 (c)) ile Bilgisayar Katmanı (Şekil 6 (b)) arasında 20cm'lik aralık olmalıdır. Yan paneller bu ölçüye göre hazırlanmıştır (Şekil 6 (d)). Sonrasında Kart Katmanı (Şekil 6 (ç)) oluşan bu kısım içerisine üzerine elektronik kartlar monte edilerek eklenmektedir. Yan Paneller üzerine sensörler, LCD, acil ve reset butonları eklenerek, robot şasisi tamamlanır. İstenilen yüksekliklere Tablet Katmanı (Şekil 6 (a)), Kamera Katmanı ve Pan Tilt (Şekil 6 (e)). eklenerek robot şasisi kullanıma hazır hale getirilir. Montaj işleminin detayları, tüm katların teknik çizimleri ve ölçüleri ekte verilmiştir.

2.1.2. Elektronik

Elektronik aksamda tamamen açık kaynak devre şeması çizimli ve ihtiyaca göre geliştirilen yazılım sayesinde çoklu kullanılacak Arduino Mega ve Arduino Uno kullanılması kararlaştırılmıştır (Şekil 7(a,b)). Motor sürücü olarak 12A akıma kadar desteği olan, geniş çalışma voltajına sahip (5.5-24V) bir motor sürücü seçilmiştir (Şekil 7(c)). Motor için ise 80 devirli enkoderli, redüktörlü ve 18 kg/cm torklu DC motor düşünülmüştür (Şekil 7(e)). Arduino Mega sayesinde daha çok port kullanımı ile doğru orantılı olarak sensör ve manipülatör bazlı genişletilir bir sistem kurmak mümkün olacaktır. Sensör bağlantılarının kolay sağlanabilmesi için Arduino Mega Sensor Shield kullanılmıştır (Şekil 7(ç)). Sensörlerin akım ihtiyacı için LiPo pilden gelen 11.1V'luk doğru akımı 5V'a indirecek bir adet voltaj düşürücü regülatör kart kullanılmıştır (Şekil 7(d)). Arduino Mega Sensor Shield kullanımı isteğe bağlıdır. Sistemin genel bağlantı yapısı Şekil 10'da görülmektedir. Sistemin detaylı bağlantı şeması (Şekil 26) ve kullanılan malzemeler ektedir.



Arduino Mega (a)

Arduino UNO (b)

12 A Motor Sürücü (c)

Arduino Mega Sensor Shield
(ç)Voltaj Düşürücü Regülatör
(d)4 A 12 V Enkoderli Redüktörlü
DC 80 RPM Motor (e)**Şekil 7: Tasarım için kullanılan elektronik aksam bileşenleri**

2.1.3. Sensörler

Çoğu mobil televarlık sistemi pilotun yerel ortamda olma hissini sağlamak amacıyla sensörlerle desteklenmiştir. HEduBot eğitim amacının yanında diğer projelerde de kullanılabilmesi için çeşitli sensörler ile donatılmıştır.

Sensör değerlerini yerel kullanıcıya test amaçlı gösterebilmek için bir adet 16x2 LCD yerleştirilmiştir (Şekil 8 (a)). Robot üzerinde ortam sıcaklığı ve nem bilgilerini ölçmek için bir adet ısı ve nem ölçüm sensörü (Şekil 8 (b)) vardır. Yangın durumunda alevlerin yönünü belirlemek için üç adet ateş sensörü (Şekil 8 (c)) bulunmaktadır. Çevresindeki cisimleri algılayabilmesi için öne üç ve arkaya üçer adet olmak üzere altı adet ultrasonik mesafe sensörleri (Şekil 8 (ç)) ve öne üç adet kızılötesi engel sensörleri (Şekil 8 (d)) ile navigasyon desteği sunulabilir. Yerel ortamdaki hareketi algılayabilmesi için 1 adet hareket sensörü (Şekil 8 (e)) yerleştirilmiştir. Sistemin genel bağlantı yapısı Şekil 10'da görülmektedir. Sistemin detaylı bağlantı şeması (Şekil 26) ve kullanılan malzemeler ektedir.

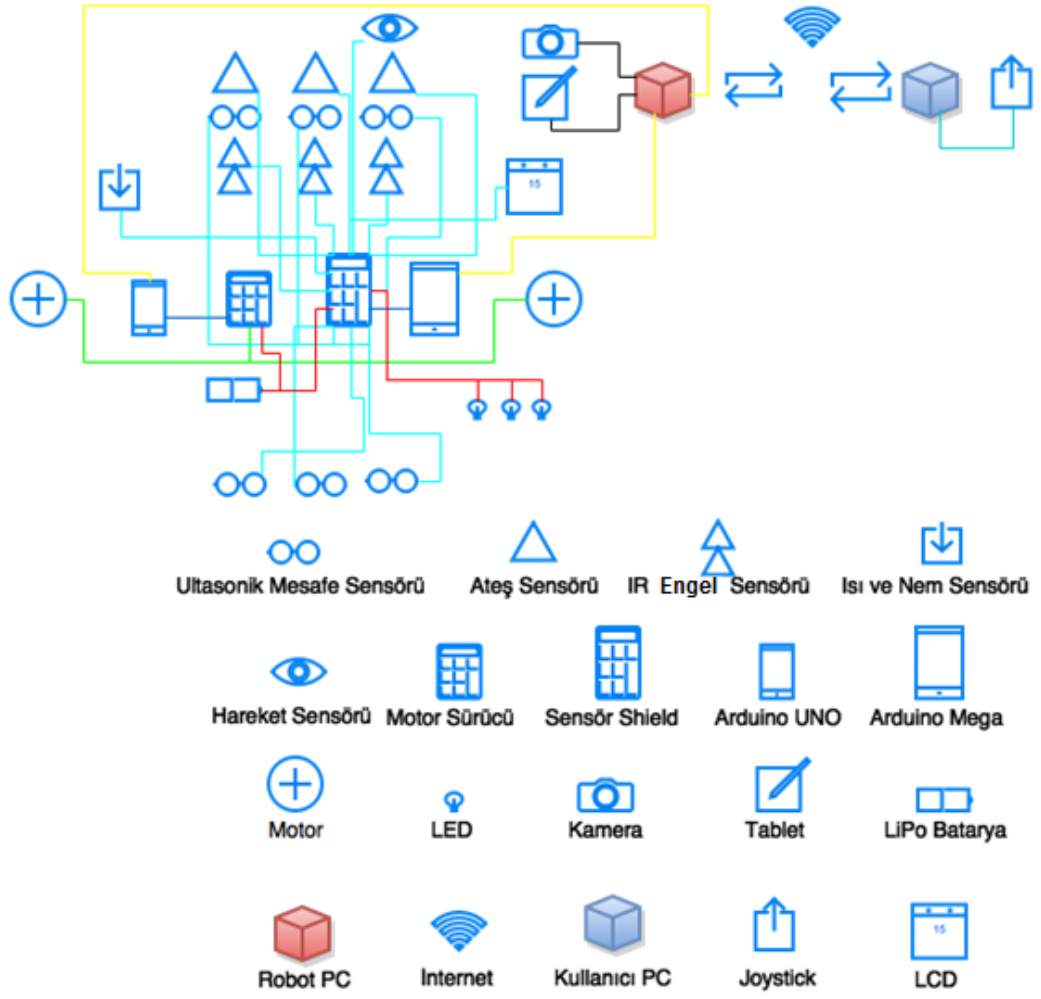


Şekil 8: Sistemde kullanılan sensörler ve sensör değerleri test ekranı

2.1.4. Elektronik Bileşenler

Pilotun yerel ortamı izleyebilmesi için kamera gereklidir (Şekil 9 (g)). Kamera pan tilt hareketlerinin sağlanabilmesi için servo motorlar kullanılmıştır (Şekil 9 (b)). Batarya olarak 5100mAh 11.1 V değerlerine sahip LiPo batarya (Şekil 9 (ç)), batarya şarjı için ise Imax B6AC şarj cihazı kullanılmaktadır (Şekil 9 (d)). Düşük şarj uyarısı için bir adet alarmlı gösterge kullanılmıştır (Şekil 9 (c)). Platform üzerinde elektronik kartlar ile bağlantı, haberleşme, kamera-tablet verilerinin alınması amacıyla taşınabilir bilgisayar kullanılmıştır. Bu bilgisayar aynı zamanda ağ üzerinden kullanıcı komutlarının alınmasını, sensör bilgilerinin kullanıcıya gönderilmesini ve kısmi güç beslemesini de sağlamaktadır (Şekil 9 (e)). Pilot kullanıcının robotu kontrol edebilmesi için bir adet joystick kullanılmıştır (Şekil 9 (f)). Aynı zamanda robot üzerindeki taşınabilir bilgisayarın USB portlarını çoklamak amacıyla bir adet 4 çıkışlı USB port çoklayıcısı kullanılmıştır (Şekil 9 (ğ)). Tasarımımızın ayırıcı özelliklerinden, öğrenci ve eğitmenin ortak çalışması için kullanılan bir adet grafik tablet mevcuttur (Şekil 9 (a)). Sistemin genel bağlantı yapısı Şekil 10'da görülmektedir. Burada kullanılan elektronik bileşenler tavsiye niteliğindedir. İhtiyaçlar doğrultusunda farklı marka model bileşenler kullanılabilir. Sistemin detaylı bağlantı şeması (Şekil 26) ve kullanılan malzemeler ektedir.

**Tablet (a)****Standart 180° Servo (b)****Düşük şarj göstergesi (c)****5100mAh Lipo Batarya (ç)****Lipo ve NiMh Şarj Aleti (d)****Taşınabilir Bilgisayar (e)****Joystick (f)****Kamera (g)****USB Port Çoklayıcısı (ğ)****Şekil 9: Sistemde kullanılan elektronik bileşenler**



Şekil 10: HEduBot sisteminin genel bağlantı yapısı

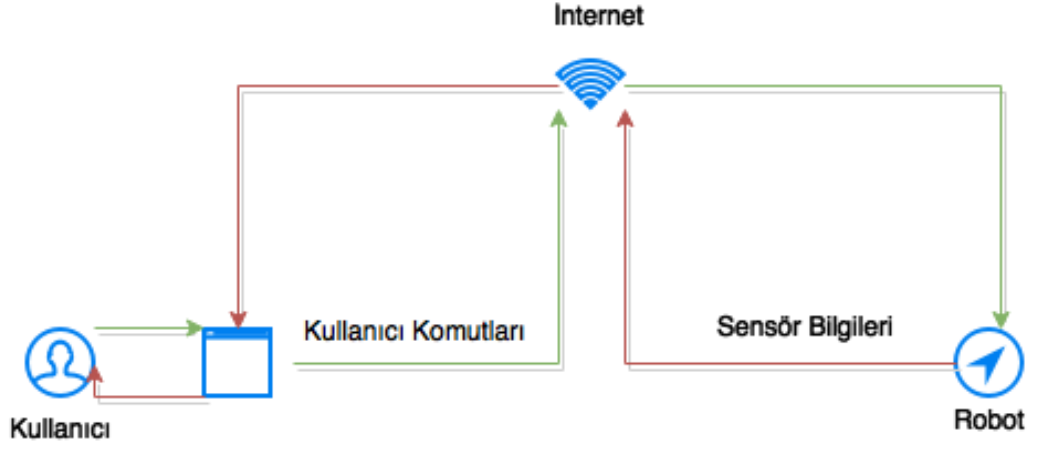
2.1.5. Yazılım

Java programlama dili ve Eclipse IDE ile robot iletişimi için bir sunucu istemci sistemi tasarlanmıştır. Java programlama dilinin tercih edilmesinin sebebi platform bağımsız olması ve çok sayıda açık kaynak ek kütüphanelerinin bulunmasıdır. BigBlueButton ise yine açık kaynak koda sahip ve aynı zamanda senkron uzaktan eğitimde tercih edilen bir sistemdir. Tüm bu sistemler Windows veya Ubuntu Linux çekirdeği üzerinde çalıştırılabilir vaziyettedir. Robot yönetim arayüzü senkron uzaktan eğitim sistemi ile birlikte kullanılabilir ve platform

bağımsız olarak derlenebilir. Robot üzerinde bulunan Arduino kartlar AVR C veya C++ kodları kullanılarak programlanmaktadır. Arduino IDE geliştirme aracı ile programlanabilir ve ücretsizdir. (Uygulama kodları kütüphane dosyaları ve geliştirme uygulamaları ek DVD içerisinde yer almaktadır.)

2.1.6. HEduBot Haberleşme Mimarisi

HEduBot haberleşmek için ağ üzerinden hareket talebi (motion request) ve sensör cevabı (sensor response) paradigmasını kullanmaktadır bu sayede robot anlık komutları alırken yerel ortam bilgilerini de anlık olarak kullanıcıya iletmektedir. Gönderilen hareket talebi komutu "jy://joystickx,joysticky,joystickh" alınan sensör cevabı ise sensörlerden okunan sayısal değerlerin ";" yardımıyla birleştirilmesi ile "hareket;solateş;ortaateş;sağateş;solengel;ortaengel;sağengel;solonmesafe;ortaonmesafe;sagonmesafe;solarkamesafe;ortaarkamesafe;sagarkamesafe" şeklindedir. Robotun bu protokole uygun mesaj oluşturan farklı programlarla kontrol edilmesi de mümkündür. Anlık talep cevap gecikmeleri maksimum 200 milisaniye şeklinde görülmektedir. Robot hareketlerinin pürüzsüz bir şekilde sağlanabilmesi için bu gecikme minimum düzeyde tutulmalıdır. Burada zorluk ağ bağlantısının optimum düzeyde sağlanabilmesidir. Pilot robota ulaşmak için robot bilgisayarının kullanıcı tarafından belirlenmiş sunucu dinleme portu ve robot bilgisayarın yer aldığı IP adresi bilgisini kontrol programı çalışma parametresi olarak belirtmelidir. Robot bilgisayarı bir yerel ağa bağlı ise ilgili porta gelen talepler ağ yöneticisi tarafından robot bilgisayarın yerel IP'sine yönlendirilmelidir ve kullanıcı robot bilgisayarın internete çıkış IP'sini çalıştırma parametresi olarak girmelidir. Temel olarak haberleşme mimarisi Şekil 11'de görülebilir. Haberleşme uygulamalarının kaynak kodları ekte verilmiştir.

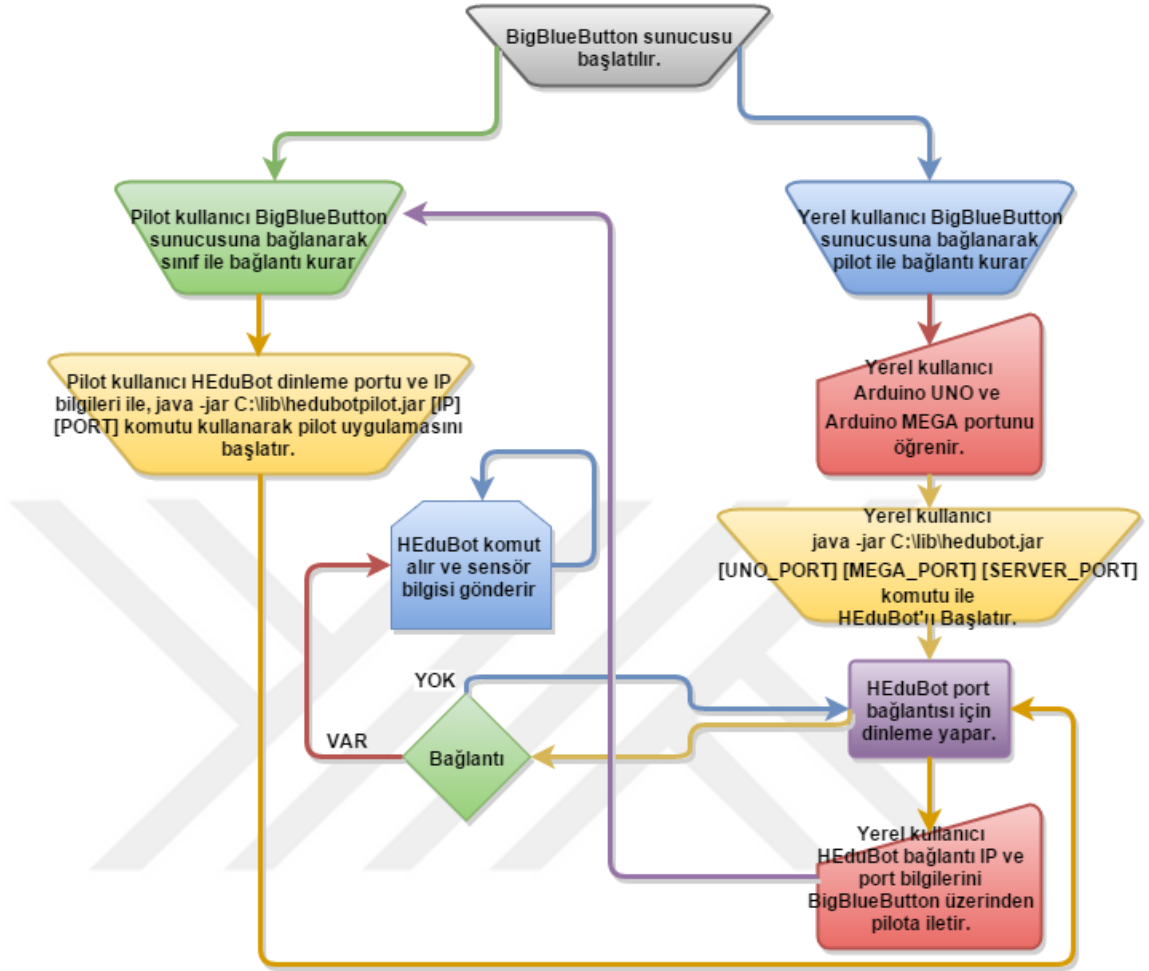


Şekil 11: HEduBot haberleşme mimarisi

2.2. SINIF ORTAMINDA HEduBot

Sınıf ortamında HEduBot'un hazır hale getirilmesi için yerel kullanıcı ve pilot ekte bulunan yönergeyi takip etmelidir. HEduBot'un özet çalışma diyagramı Şekil 12'de verilmiştir. Buna göre önce yerel kullanıcı HEduBot'u kullanıma hazır hale getirmeli ve çalışma bilgilerini pilota iletmelidir sonrasında pilot kontrol uygulamasını başlatır.

Yerel ortamda öğrenciler ve yerel kullanıcı HEduBot ile birlikte bulunmaktadır.(Şekil 13).



Şekil 12: HEduBot'un çalışmasını özetleyen diyagram

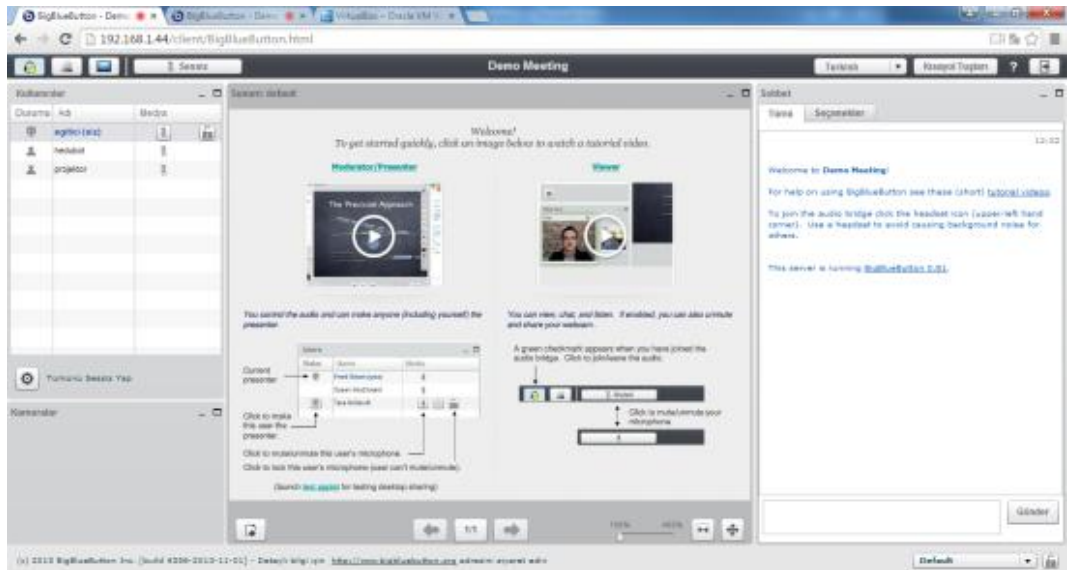


Şekil 13: Sınıf ortamında HEduBot

Pilot HEduBot'a bağlandıktan sonra üzerindeki sensörlerin bilgilerini okuyabilir, robotu ve kamerayı yönlendirebilir. BigBlueButton aracılığı ile robot kamerasından yerel ortamı izleyebilir. Doküman paylaşabilir. Sohbet açabilir. Kullanıcılardan beyaz tahta veya doküman üzerinden veri girişi alabilir, birden fazla eğitmen ile ders yürütebilir (Şekil 14 ve 15).



Şekil 14: HEduBot pilot kullanım testi. Eğitmen BigBlueButton ve HEduBot ile uzaktan eğitim verirken.



Şekil 15: Eğitmen ekranından bir görüntü



Şekil 16: HEduBot öğrenci kullanım testi. Öğrenci ile uzak eğiticinin BigBlueButton ve HEduBot grafik tableti aracılığıyla ortak ders materyali üzerinde çalışması.

Tüm bu faaliyetler diğer öğrenciler tarafından sınıf bilgisayarına veya kablosuz olarak robot bilgisayarına bağlı projektör yardımı ile görüntülenebilmektedir (Şekil 13 ve 16). Pilot dilerse internet üzerinden paylaşım açılacak olan yerel bir yazıcı ile yerel ortama belge gönderebilir ve yapılacak olan çalışmaları gözlemleyebilir. HEduBot'un sınıf ortamında rahatça hareket edebilmesi için sıralar arasında geçişine uygun boşluklar bulunmalıdır. Ayrıca sınıf içerisinde optimum düzeyde kablosuz ağ bağlantısının bulunması gerekmektedir. HEduBot'un detaylı kurulum ve kullanım kılavuzu ektedir.

ÜÇÜNCÜ BÖLÜM

HEduBot TEKNİK ÖZELLİKLERİ

HEduBot için teknik özellikler belirlenirken kullanılan donanımın teknik özellikleri referans alınmıştır ve yaklaşık kullanım değerleri bu yolla hesaplanmıştır.

3.1. BOY VE AĞIRLIK

Robotun boyu 75,5cm, ağırlığı ise tüm sistemler üstünde iken 8,6kg'dır (Şekil 17). Kullanılan gijonlar plastik gijon ile değiştirilerek ağırlık düşürülebilir. Aynı zamanda boy robotun yapımı esnasında değiştirilebilir.



Şekil 17: HEduBot ağırlık değerlerinin ölçümü

3.2. HIZ

Kullanılan motor 12V ile 80 RPM yapan 131:1 dişli oranı olan yükte 5A akım çeken bir motordur [109]. Teker ise 9 cm çaplıdır. Sert teker olduğu için çapında değişim olmamaktadır [110]. Yüksüz değerler ile robotun teknik olarak hızı;

Teker için $\text{çap}=9\text{cm}$

Motor için 80RPM devir dakika;

Boşta hesaplama:

$$\text{Çevre}=2\pi r$$

$$\text{Çevre}=2 \times 4,5 \times 3,14$$

$$\text{Çevre}=28.26\text{cm}$$

$$\text{Dakikada Alınan Yol}=80 \times 28,26$$

$$\text{Dakikada Alınan Yol}=2260,8$$

$$\text{Saniyede Alınan Yol}=2260,8/60$$

$$\text{Saniyede Alınan Yol}=37,68\text{cm'dir.}$$

Robotun yükteki deneysel hızını hesaplamak için 5m uzunluğunda düz bir parkur oluşturulmuş, robot tam hızda bu parkuru 10 defa tamamlamıştır (Şekil 18). Tutulan değerler ile hesaplanan ortalama hız saniyede 33.51cm şeklindedir (Tablo 2).

Tablo 2: Hız hesaplaması deneylerinde elde edilen sonuçlar

Deney No	Mesafe	Süre	Hız
1	500cm	15,17sn	32,95
2	500cm	15,20	32,89
3	500cm	14,95	33,44
4	500cm	15,35	32,57
5	500cm	14,71	33,99
6	500cm	14,80	33,78
7	500cm	14,97	33,40
8	500cm	14,59	34,27
9	500cm	15,02	33,28
10	500cm	14,47	34,55
ORTALAMA	500cm	14,92	33,51



Şekil 18 HEduBot hız testi

3.3. ÇALIŞMA SÜRESİ

Kullanılan DC motor boşta bekleme esnasında 350mA yükte ise yani 18kg çekerken 5A akım çekmektedir [109]. HEduBot tartı ağırlığı 8,6 kg dır. Ağırlık ile çekilen yük doğru orantı ile hesaplandığı taktirde, robot ağırlığı ile motorlar yaklaşık 2.38A akım çekmektedir.

3 adet kızılötesi MZ80 engel sensörü vardır. Bu sensörlerin bir tanesi çalışma esnasında 10 mA akım çekmektedir [111].

6 adet HCSR04 ultrasonik mesafe ölçüm sensörü vardır. Bu sensörlerin bir tanesi çalışma esnasında 15 mA akım çekmektedir [112].

1 adet HW8002 hareket sensörü vardır. Bu sensörlerin bir tanesi çalışma esnasında 400 μ A akım çekmektedir [113].

Isı ve nem sensörü olarak DHT21 kullanılmıştır. Ölçme ve beklemede ortalama 1.1 mA akım çekmektedir [114].

Sistemde 2 adet standart 180^olik servo motor bulunmaktadır. Bu servo motorlar yükte 800 mA akım çekmektedir [115].

Arduino UNO [116] ve Mega 2560 [117] 500mA USB akım korumasına sahiptir. Bu akımdan fazlasının çekilmesi Arduino üzerindeki bir sigorta ile durdurulmuştur ve USB bağlantısı kesilir. Bu yüzden bu sensörler ve servo motorların kullanılması için ek güç kaynağına ihtiyaç vardır. Bu güç kaynağı ihtiyacı da yine 5000mA'lık LiPo pilden voltaj düşürücü regülatör kartı ile sağlanacaktır.

Tam yükte hesaplama:

DC motorlar yükte 2,38A	=2,38 A
MZ80 akım değeri 10mA \times 3	=30 mA
HCSR04 akım değeri 15mA \times 6	=90 mA
HW8002 akım değeri	=400 μ A
DHT21 akım değeri	=1.1 mA
Servo akım değeri 800mA \times 2	=1,6A
Toplam	\approx 4.101A

Güç kaynağımız saat boyunca 5000mA verebilir. 4,1A'lik tüketim ile tam yükte yaklaşık 1,2 saat kullanılabilir.

Yüksüz hesaplama:

DC motorlar boşta 350mA \times 2	=700 mA
MZ80 akım değeri 10mA \times 3	=30 mA
HCSR04 akım değeri 15mA \times 6	=90 mA
HW8002 akım değeri	=400 μ A
DHT21 akım değeri	=1,1 mA
LED akım değeri 20mA \times 3	=60 mA
Servo tam yükte 4mA \times 2	=8mA

Toplam $\approx 888\text{mA}$

Güç kaynağımız saat boyunca 5000mA verebilir. 888mA'lik tüketim ile boşta yaklaşık 5,6 saat kullanılabilir.

Ortalama alındığında tek şarj ile yaklaşık 3,4 saat boyunca robotu kullanmak mümkündür. Bu süre aynı teknik değerde LiPo pillerin paralel bağlı kullanımını ile pil sayısı ile doğru orantılı artacaktır. Örneğin 2 adet 11.1V 5000mA LiPo kullanıldığında süre iki katına yani 6,8 saate çıkacaktır.

3.4. SENSÖR ÖZELLİKLERİ

MZ80 ile 15^0 'lik alanda 3cm ile 80cm arası uzaklıkta bulunan engeller algılanabilir. 2ms (milisaniye) tepki süresi vardır [111].

HCSR04 ile 15^0 'lik alanda 2cm ile 400 cm arası uzaklıkta bulunan cisimlerin uzaklıkları algılanabilir. Tepki süresi uzaklığın değerine göre değişmektedir [112].

HW8002 ile x ekseninde 69^0 , y ekseninde 62.5^0 'lik açıdaki hareketler algılanabilir [113].

DHT21 ile -40 dan 80C^0 'ye kadar sıcaklık ölçümü ve 0-100% arası nem ölçümü 0,1 çözünürlükle $\pm 1\%$ nem $\pm 0,2 \text{C}^0$ ısı duyarlılık ile alınabilir [114].

Bu sensörler kullanılarak otonom bir şekilde çarpışma olmadan hareket eden robot navigasyon yazılımının tasarımı başka bir çalışmanın konusudur ve yakın zamanda başlaması planlanmaktadır.

GELECEK ÇALIŞMALAR

Çalışmamızın odak noktası, uzaktan eğitim için tasarlanacak düşük maliyetli ve kolay uygulanabilir bir televarlık robot platformu tasarlamak olmuştur.

Gelecek çalışmalarda tasarlanan bu platforma, otonom seyrüsefer ve şarj, rota takibi, ortam haritalama, konum bulma, bağlantı kopması durumunda otonom davranışlar sergileme, mimik ve jest yapabilme, bağlantı yapısına ve fiziki koşullara uygun seyrüsefer algoritması seçimi gibi özellikler eklenecektir.

Aynı zamanda bu güncelleştirmelerin yapılması esnasında HEduBot'un uzaktan eğitime etkisi konusunda çalışmaların yapılması doğru olacaktır. Bu çalışmalar, gerçek ders ortamında, uygun süreli, yaygın ve objektif olmalıdır çünkü geri dönüşsel sonuçları HEduBot'un gelecek güncellemelerine de yol gösterici olacaktır. Çalışmalarda doğru bilimsel sonuçlara ulaşabilmek için, eğitim-uzaktan eğitim, ölçme ve değerlendirme alanlarında uzman bir kadro ile uygun bir yapı oluşturularak, disiplinler arası çalışma izlenilmesi gerekliliği görülmektedir.

Aynı zamanda HEduBot'un bina içerisinde diğer katlara da ulaşabilmesi ihtiyacı öngörülmektedir. Bunun için merdiven veya asansör kullanımı gerekmektedir. Bunun yanında yerel kullanıcıların robot bilgisayarını daha aktif ve rahat kullanabilmesi önemlidir. Bu isterleri gerçekleştirmek için HEduBot'un mekanik tasarımında güncelleştirmeler yapılacaktır.

SONUÇ

Bu çalışmada, uzaktan eğitimde öğrenci-öğretmen etkileşimi sorununa bir çözüm önerisi olarak, hibrit eğitim robotu HEduBot (Şekil 19) tasarımı açık kaynaklı olarak sunulmuştur.

Yerel imkanlarla yararlanıcıların bu projeden faydalanabilmesi isteri, özelleştirilebilir olması isteri ve optimal ağ bağlantısının kullanım esnasında sürekli sağlanması gerekliliği çalışma esnasında karşılaşılan başlıca zorluklar olarak sıralanabilir.

Bunun yanında HEduBot'un bu ilk sürümünde yerel kullanıcıya olan ihtiyaç, sonraki sürümlerde daha da azaltılmalı, pilot kullanıcıya kullanım esnasında daha fazla destek ve kullanım kolaylığı sağlanmalıdır.

Yapılan geniş alanyazın çalışmasında, televarlık robotlarda görülen sunum paylaşabilme, kullanıcıdan el yazısı ile veri girişi alabilme, çoklu telekonferans yapabilme gibi önemli eksikliklerin giderilmesi gerekliliği bundan sonra geliştirilecek olan araştırma amaçlı, genel amaçlı veya ticari amaçlı platformlara da yol gösterici niteliktedir.

HEduBot tasarımının açık kaynak olması ve düşük maliyeti, (Kaynak kod ve maliyet tablosu ekte verilmiştir.) yararlanıcıların rahatlıkla kendi kurumları için bu platformu kullanmalarına olasılık sağlayacaktır. İhtiyaçlar doğrultusunda tasarımının özelleştirilebilir olması, robot görmesi, yapay zeka ve benzeri gelecek bilimsel çalışmaların da önünü açacaktır. Çalışmada her kesimden insanın rahatlıkla bu robotu geliştirebilmesi için geniş geliştirme kitlesine sahip açık kaynak teknolojilerin kullanılmasına dikkat edilmiştir.

Uzaktan eğitim kavramında “uzak” özelliğinin negatif etkilerini azaltmak, eğitimin kalitesini arttırmak aynı zamanda da yaygınlaştırmakta bir çözüm önerisi sunmak bu projenin temel motivasyonları olmuştur.

Gelecek çalışmalarda bu platformun uzaktan eğitimde eğitici ve öğrenciler tarafından değerlendirilmesi, geribildirimsel iyileştirmeler yapılarak sonuçlarının yayınlanması düşünülmektedir.



Şekil 19: HEduBot'un farklı açılardan görünümü

TEŐEKKÜR

Çalıőmamızda 001-10-2014 numaralı bilimsel araőtırma projesi (BAP) ile bizi destekleyen Turgut Özal Üniversitesi'ne,

Projemizin yapım aőamasında 114E554 kodlu proje ile bizi destekleyen Türkiye Bilimsel ve Teknolojik Araőtırma Kurumu (TÜBİTAK)'na teőekkür ederiz



KAYNAKÇA

- [1] **Minsky, M.** (1980). "Telepresence", OMNI Magazine, Sf. 45-52.
- [2] **Chang, C.W., Lee, J.H., Chao, P.Y., Whang, C.Y., Chang, G.D.** (2010). "Exploring the Possibility of Using Humanoid Robots as Instructional Tools for Teaching a Second Language in Primary School". Educational Technology & Society, 13 (2), Sf. 13–24.
- [3] **Mubin O., Stevens C. J., Shahid S., Mahmud A.A., Dong J.J.** (2013). A review of the applicability of robots in education. Technology for Education and Learning, 1, Sf. 209-0015.
- [4] **Url-1** <<https://www.whitehouse.gov/blog/2013/06/25/mozilla-ignite-challenge-winner-announced>>, alındığı tarih: 28.08.2015.
- [5] **Url-2** <<http://bigbluebutton.org/history/>>, alındığı tarih: 04.07.2015.
- [6] **Url-3** <<http://bigbluebutton.org/>>, alındığı tarih: 04.07.2015.
- [7] **Kristoffersson, A., Coradeschi, S., ve Loutfi, A.** (2013). A Review of Mobile Robotic Telepresence. Advances in Human-Computer Interaction, Volume 2013, Article ID 902316, 17 pages.
- [8] **Paulos, E. ve Canny, J.** (1998) Designing personal tele-embodiment,"in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98), pp. 3173–3178, May 1998
- [9] **Paulos, E. ve Canny, J.** (2001). Social tele-embodiment: understanding presence, Autonomous Robots, vol. 11, no. 1,Sf. 87–95.
- [10] **Url-4** <<http://www.giraff.org/about-giraff/?lang=en>>, alındığı tarih: 04.07.2015.
- [11] **Url-5** <<https://www.anybots.com/products/>>, alındığı tarih: 04.07.2012.
- [12] **Gibstein, M.**, CES coverage anybots QB telepresence robot hands on. Tech Tracker Tracking Today's Tech.
<http://techtrackr.com/2011/01/12/ces-coverage-anybots-qb-telepresence-robot-hands-on/>, alındığı tarih: 12 Ocak 2011.
- [13] **Tsui, K. M., Desai, M., Yanco, H. A. ve Uhlik, C.** (2011). "Exploring use cases for telepresence robots," in Proceedings of the 6th ACM/IEEE

International Conference on Human-Robot Interaction (HRI '11), Sf. 11–18, March 2011.

- [14] **Url-6** <<http://www.willowgarage.com/pages/texai>>, alındığı tarih: 08.2012.
- [15] **Url-7** <<http://ros.org>>, alındığı tarih:08.2012.
- [16] **Url-8** <<http://suitabletech.com>>, alındığı tarih:08.2012.
- [17] **Url-9** <<http://www.vgo.com>>, alındığı tarih:08.2012.
- [18] **Url-10**
<http://www.vgocom.com/sites/default/files/vgo_user_guide_v1.5.0.pdf>, alındığı tarih:08.2012.
- [19] **Url-11** <<http://www.ryerson.ca/pebbles/>>, alındığı tarih:08.2012.
- [20] **Url-12** <<http://mantarobot.com>>, alındığı tarih:08.2012.
- [21] **Url-13** <<http://gostai.com>>, alındığı tarih: 08.2012.
- [22] **Url-14** <<http://www.urbiforge.org>>, alındığı tarih: 12.2012.
- [23] **Url-15** <<http://www.irobot.com/ava>>, alındığı tarih: 08.2012.
- [24] **Url-16** <<http://9thsense.com/>>, alındığı tarih: 08.2012.
- [25] **Url-17** <<http://www.doublerobotics.com/>>, alındığı tarih: 12.2012.
- [26] **Url-18** <<http://www.bnrobotics.com/#!/robots/c14gd/>>, alındığı tarih: 11.07.2015.
- [27] **Url-19** <<http://intouchhealth.com>>, alındığı tarih: 08.2012.
- [28] **Url-20** <<http://www.intouchhealth.com/ITHFAQs.pdf>>, alındığı tarih: 08.2012.
- [29] **Adalgeirsson, S. O. ve Breazeal, C.** (2010). "MeBot: a robotic platform for socially embodied presence" ,5th ACM/IEEE International Conference on Human-Robot Interaction (HRI '10), pp. 15–22,Osaka, Japan,March 2010.
- [30] **Url-21** <<http://www.robotshop.com/blog/en/irobiq-a-fun-and-interactive-robot-on-sale-at-robotshop-589>>, alındığı tarih: 11.07.2015.
- [31] **Sangseok, Y., Jongju, S., Daijin, K., Chang Gu, K., Munsang, K., Mun-Taek, C.** (2011). "Engkey: Tele-education Robot", Social Robotics Lecture Notes in Computer Science Volume 7072, 2011, Sf. 142-152
- [32] **Url-22** <<http://www.roboticstoday.com/robots/engkey>>, alındığı tarih: 08.08.2015

- [33] **Url-23** <<http://www.mobilerobots.com/ResearchRobots/PeopleBot.aspx>>, alındığı tarih: 08.08.2015
- [34] **Url-24** <<https://www.kickstarter.com/projects/1452620607/botiful-telepresence-robot-for-android/description>>, alındığı tarih: 08.08.2015
- [35] **Url-25** <<http://www.romotive.com/>>, alındığı tarih: 08.08.2015
- [36] **Url-26** <<http://makezine.com/2012/12/21/lego-mindstorms-telepresence-robot/>>, alındığı tarih: 08.08.2015
- [37] **Url-27** <<http://www.vgocom.com/how-does-vgo-work>>, alındığı tarih: 08.07.2015
- [38] **Url-28** <<http://www.vgocom.com/>>, alındığı tarih: 08.07.2015
- [39] **Url-29** <<https://www.suitabletech.com/beampro/>>, alındığı tarih: 08.07.2015
- [40] **Url-30** <<https://www.suitabletech.com/beam-plus/>>, alındığı tarih: 08.07.2015
- [41] **Url-31** <<http://mantarobot.com>>, alındığı tarih: 10.07.2015
- [42] **Url-32** <<http://www.irobot.com/For-Business/Ava-500.aspx>>, alındığı tarih: 10.07.2015
- [43] **Url-33** <<http://www.doublerobotics.com/>>, alındığı tarih: 11.07.2015
- [44] **Url-34** <<http://www.intouchhealth.com/our-solutions/patient-access/intouch-vita/>>, alındığı tarih: 08.08.2015
- [45] **Url-35** <<http://www.intouchhealth.com/our-solutions/patient-access/>>, alındığı tarih: 08.08.2015
- [46] **Url-36** <<http://www.3ders.org/articles/20131031-rosco-low-cost-open-source-3d-printable-diy-rover.html>>, alındığı tarih: 08.08.2015
- [47] **Url-37** <http://www.blooloop.com/features/telepresence-robots-facilitating-access-enhancin/31103#.VaT4M_ntmko>, alındığı tarih: 08.08.2015
- [48] **Url-38** <<http://www.smashingrobotics.com/ramblebot-an-affordable-telepresence-robot-for-the-family/>>, alındığı tarih: 11.08.2015
- [49] **Url-39** <<http://www.edmstudio.com/dennis/index.html>>, alındığı tarih: 11.08.2015
- [50] **Url-40** <<http://telepresenceroobot.blogspot.com.tr/>>, alındığı tarih: 11.08.2015

- [51] **Url-41** <<http://armdevices.net/2014/10/10/caiba-net-telepresence-robot-in-oculus-rift/>>, alındığı tarih: 12.08.2015
- [52] **Url-42** <http://usb.brando.com/midbot_p01839c035d15.html>, alındığı tarih: 12.08.2015
- [53] **Url-43** <<http://www.andymark.com/product-p/am-2515.htm>>, alındığı tarih: 12.08.2015
- [54] **Url-44** <<http://techcrunch.com/2011/11/07/video-telepresence-robot-avatar-telesar-v/>>, alındığı tarih: 12.08.2015
- [55] **Url-45** <http://robot.kaist.ac.kr/?page_id=360>, alındığı tarih: 12.08.2015
- [56] **Url-46** <<http://www.zagrosrobotics.com/shop/custom.aspx?recid=59>>, alındığı tarih: 12.08.2015
- [57] **Url-47** <<http://www.wowwee.com/en/support/rovio>>, alındığı tarih: 12.08.2015
- [58] **Url-48** <<http://beatty-robotics.com/introducing-telegance/>>, alındığı tarih: 12.08.2015
- [59] **Url-49** <https://www.ri.cmu.edu/research_project_detail.html?project_id=565&menu_id=261>, alındığı tarih: 12.08.2015
- [60] **Url-50** <<https://www.kickstarter.com/projects/1452229737/helios-evolve-your-iphone>>, alındığı tarih: 12.08.2015
- [61] **Url-51** <<http://hackaday.com/2012/03/30/loudest-telepresence-robot-ever/>>, alındığı tarih: 12.08.2015
- [62] **Url-52** <<http://www.robots-dreams.com/2011/06/ipad2-on-wheels-makes-an-impressive-remote-telepresence-robot-video.html/>>, alındığı tarih: 12.08.2015
- [63] **Url-53** <<http://www.element14.com/community/community/news/blog/2011/12/27/e-one-a-telepresence-robot>>, alındığı tarih: 12.08.2015
- [64] **Url-54** <<http://spectrum.ieee.org/automaton/robotics/robotics-hardware/upenn-dora-platform>>, alındığı tarih: 12.08.2015
- [65] **Url-55** <<https://gomistyle.wordpress.com/sparky/>>, alındığı tarih: 12.08.2015

- [66] **Url-56**<<http://hackedgadgets.com/2013/01/29/telepresence-on-the-cheap/>>, alındığı tarih: 12.08.2015
- [67] **Url-57**<<http://www.gizmag.com/mh2-telepresence-robot/22860/>>, alındığı tarih: 12.08.2015
- [68] **Url-58**< <http://swarmlab.unimaas.nl/mitro/>>, alındığı tarih: 12.08.2015
- [69] **Url-59**< <http://blog.atmel.com/2015/02/17/origibot-is-an-open-source-telepresence-robot-with-an-articulated-arm/>>, alındığı tarih: 13.08.2015
- [70] **Url-60**<<http://spectrum.ieee.org/geek-life/hands-on/a-diy-telepresence-robot>>, alındığı tarih: 13.08.2015
- [71] **Url-61**<<http://walyou.com/personal-telepresence-robot/>>, alındığı tarih: 13.08.2015
- [72] **Url-62**<http://www.telepresenceoptions.com/2009/01/new_telepresence_robot_forum_an/>, alındığı tarih: 13.08.2015
- [73] **Url-63**<<http://www.neobotix-robots.com/telepresence-robot-mg-500.html>>, alındığı tarih: 13.08.2015
- [74] **Url-64**<<http://www.telepresencerobots.com/russias-synergy-swan-wins-price>>, alındığı tarih: 14.08.2015
- [75] **Url-65**<<http://techcrunch.com/2011/09/27/hospi-rimo-meet-panasonics-cute-assistance-robot/>>, alındığı tarih: 14.08.2015
- [76] **Url-66**<<http://procrastineering.blogspot.com.tr/2011/02/low-cost-video-chat-robot.html>>, alındığı tarih: 14.08.2015
- [77] **Url-67**<<http://spectrum.ieee.org/automaton/robotics/home-robots/telemba-telepresence-robot>>, alındığı tarih: 14.08.2015
- [78] **Url-68**<<https://www.revolverobotics.com/using-kubi/>>, alındığı tarih: 14.08.2015
- [79] **Url-69**<<https://www.indiegogo.com/projects/padbot-a-telepresence-robot-cool-fun#/story>>, alındığı tarih: 14.08.2015
- [80] **Url-70**< <http://store.irobot.com/shop/index.jsp?categoryId=3311370>>, alındığı tarih: 14.08.2015
- [81] **Url-71**<<http://www.xaxxon.com/oculus>>, alındığı tarih: 17.08.2015

- [82] **Url-72**<<http://www.xaxxon.com/oculusprime/shop>>, alındığı tarih: 17.08.2015
- [83] **Url-73**<<http://www.superdroidrobots.com/shop/item.aspx/rp2w-two-way-remote-presence-robot-open-style/1046/#options>>, alındığı tarih: 17.08.2015
- [84] **Url-74**<<http://www.gizmag.com/telepresence-law-enforcement-robot/24354/>>, alındığı tarih: 18.08.2015
- [85] **Url-75**<<http://www.smashingrobotics.com/telepresence-robots-reviewed-part-3/>>, alındığı tarih: 18.08.2015
- [86] **Url-76**<<https://www.nersec.gov/news-publications/nersec-news/nersec-center-news/2002/berkeley-lab-s-rage-telepresence-robot-captures-r-and-d100-award>>, alındığı tarih: 18.08.2015
- [87] **Url-77**<<http://www.swanrobotics.com/projects/telepresence-robot/>>, alındığı tarih: 18.08.2015
- [88] **Url-78**<<https://wicron.com/en/projects/webot>>, alındığı tarih: 18.08.2015
- [89] **Url-79**<http://www.sastrarobotics.com/products/smart_telepresence/>, alındığı tarih: 18.08.2015
- [90] **Url-80**<http://www.cse.buffalo.edu/~ss424/ms_progress_page.html>, alındığı tarih: 18.08.2015
- [91] **Url-81**<<http://www.telepresenceart.com/gsmith/Artworks.html>>, alındığı tarih: 18.08.2015
- [92] **Url-82**<<http://www.thehindu.com/todays-paper/tp-national/tp-kerala/state-joins-the-robot-league/article5040656.ece>>, alındığı tarih: 18.08.2015
- [93] **Url-83**<http://forums.trossenrobotics.com/robots.php?project_id=30#ad-image-0>, alındığı tarih: 20.08.2015
- [94] **Url-84**<<http://robonaut.jsc.nasa.gov/R1/sub/telepresence.asp>>, alındığı tarih: 20.08.2015
- [95] **Url-85**<<http://www.instructables.com/id/OLPC-Telepresence/>>, alındığı tarih: 20.08.2015
- [96] **Url-86**<<http://www.peterklemperer.com/blog/2011/10/15/tabletop-telepresence-robot/>>, alındığı tarih: 20.08.2015

- [97] **Url-87**<<http://www.mistywest.com/portfolio-items/weelooms/>>, alındığı tarih: 20.08.2015
- [98] **Url-88**<https://www.youtube.com/watch?v=qMnw_dheMrs>, alındığı tarih: 20.08.2015
- [99] **Url-89**<<http://www.instructables.com/id/MAYA-Telepresence-Rover/>>, alındığı tarih: 20.08.2015
- [100] **Url-90**<http://mobiton.odcaf.com/technical_details.html>, alındığı tarih: 23.08.2015
- [101] **Url-91**<<http://blog.dugnorth.com/2009/03/erector-spykee-spy-robot.html>>, alındığı tarih: 23.08.2015
- [102] **Url-92**<<http://dhcompliant2.com/ces-2013-robotex-avtar-the-telepresence-robot/>>, alındığı tarih: 23.08.2015
- [103] **Yi-Shin, C., Yeh-Liang, H. ve Jun-Ming, L.** (2013) "TRiCmini+ – telepresence robot for interpersonal communication for older adults". Chinese Society of Mechanical Engineers,34(2), 121-127, 2013
- [104] **Url-93**<<http://www.geminoid.jp/en/robots.html>>, alındığı tarih: 23.08.2015
- [105] **Url-94**<<http://www.engadget.com/2011/05/16/sony-makes-floating-head-telepresence-avatars-a-reality-sean-co/>>, alındığı tarih: 23.08.2015
- [106] **Tuncer, M., TAŞPINAR, M.** (2008) Sanal Ortamda Eğitim Ve Öğretimin Geleceği Ve Olası Sorunlar, Sosyal Bilimler Dergisi, 2008
- [107] **Peters, R.** (1966). Ethics and Education, London: Allen & Unwin.
- [108] **Peters, O.** (1967). 'Distance teaching and industrial productions: a cooperative interpretation in outline', in D.Sewart, D. Keegan and B. Holmberg
- [109] **Url-95**<<https://www.pololu.com/product/1447>>, alındığı tarih: 28.08.2015
- [110] **Url-96**<<https://www.pololu.com/product/1435>>, alındığı tarih: 29.08.2015
- [111] **Url-97**<<http://robotus.net/wp-content/uploads/2012/07/MZ80-endustriyel-kizilotesi-sensor.pdf>>, alındığı tarih: 29.08.2015
- [112] **Url-98**<http://elec Freaks.com/estore/download/EF03085-HC-SR04_Ultrasonic_Module_User_Guide.pdf>, alındığı tarih: 29.08.2015
- [113] **Url-99**<<http://oomlout.com/parts/PIRS-01-datasheet.pdf>>, alındığı tarih: 29.08.2015

- [114] **Url-100**<<http://www.electrodragon.com/w/images/6/6f/DHT21.pdf>>, alındığı tarih: 29.08.2015
- [115] **Url-101**<<http://www.robotistan.com/PowerHD-Standart-Analog-Servo-Motor-HD-3001HB,PR-3222.html>>, alındığı tarih: 29.08.2015
- [116] **Url-102**<<https://www.arduino.cc/en/Main/arduinoBoardUno> >, alındığı tarih: 29.08.2015
- [117] **Url-103**<<https://www.arduino.cc/en/Main/arduinoBoardMega2560>>, alındığı tarih: 29.08.2015
- [118] **Url-104**<<http://www.tcmb.gov.tr/wps/wcm/connect/tcmb+tr/tcmb+tr/main+menu/istatistikler/doviz+kurlari/gosterge+niteligindeki+merkez+bankasi+kurlari>>, alındığı tarih: 29.08.2015
- [119] **Url-107**<<https://code.google.com/p/bigbluebutton/wiki/BigBlueButtonVM>>, alındığı tarih: 28.08.2015
- [120] **Url-108**<<https://www.virtualbox.org/wiki/Downloads>>, alındığı tarih: 28.08.2015
- [121] **Url-105**<<https://theuzo007.wordpress.com/2013/10/26/joystick-in-java-with-jinput-v2/>>, alındığı tarih: 30.08.2015
- [122] **Url-106**<<http://electronics.stackexchange.com/questions/19669/algorithm-for-mixing-2-axis-analog-input-to-control-a-differential-motor-drive>>, alındığı tarih: 30.08.2015

EKLER

Robotun teknik çizimleri, montaj kılavuzu, kaynak kodları ve tez dokümanı <http://ozguryilmazresearch.net> adresinde paylaşılmıştır. Ayrıca bazı videolar ile kullanım ile ilgili ayrıntılı açıklamalar görselleştirilmiştir.

8.1. GEREKLİ ARAÇLAR VE FİYATLARI

Gerekli araçların adet ve fiyatları bkz. Tablo 3 verilmiştir. Fiyatlarda TL birimi KDV dahil, diğer birimler KDV hariç verilmiştir. Kurlar 28.08.2015 tarihinde USD için 2,9199 TL, EURO için 3,2936 TL'dir [118]. Robotun yapımına sarf edilen malzemeler için toplam fiyat kurların belirlendiği tarihte 2462,5 TL KDV dahil, USD cinsinden 843,35 dolar KDV dahil olarak hesaplanmıştır. bkz. Tablo 3. Robotun yapımında kullanılan aletlerin fiyatları bkz. Tablo 4 484.5 TL KDV dahi, 1 USD cinsinden 165,96 dolar KDV dahil olarak hesaplanmıştır.

Tablo 3: Sarf malzeme adet ve fiyatları

Sıra No	Adı	Adet	Fiyat	Birim
1	Orjinal Arduino UNO R3	1	18,5	EURO
2	Orjinal Arduino Mega 2560 R3 (Yeni Versiyon)	1	38,6	EURO
3	Pololu 12V 37mm 80 Rpm Enkoderli 131:1 Redüktörlü DC Motor	2	39,7	USD
4	Pololu VNH5019 Arduino Uyumlu Çift Motor Sürücü Kartı	1	49,6	USD
5	37D mm Metal Motor Tutucu Paketi (Bağlantı Aparatı) - Pakette 2 Adet	1	19,12	TL
6	Arduino Mega Sensör Shield (Opsiyonel)	1	7,1	USD
7	6mm Motor Bağlantı Elemanı Çifti (M3 Sabitleme Vida Delikli)	1	7,9	USD
8	Pololu 90x10mm Tekerlek Çifti (Siyah)	1	9,9	USD
9	180° Dönebilen Standart Servo Motor	2	11,75	USD
10	HC-SR04 Ultrasonik Mesafe Sensörü	6	2,9	USD
11	80cm Menzilli Kızılötesi Sensör - MZ80	3	8	USD
12	DHT21 Isı ve Nem Sensörü	1	7,5	USD
13	Ateş Algılayıcı Sensör Kartı (Flame Sensor)	3	2,5	USD
14	HC-SR501 Ayarlanabilir IR Hareket Algılama Sensörü - PIR	1	3	USD

Tablo 3: Sarf malzeme adet ve fiyatları (devamı)

15	Adan Bye USB Kablosu/ Yazıcı Kablosu	2	1,5	USD
16	11,1V Lipo Batarya 5100mAh (Adet arttırılarak ve paralel bağlanarak çalışma süresi arttırılabilir.)	1	53	USD
17	16x2 IIC/I2C/TWI Seri LCD Ekran Mavi	1	7	USD
18	Mini Ayarlanabilir 3A Voltaj Düşürücü Regülatör Kartı Step Down	1	2,1	USD
19	1S-8S LiPo Buzzer Voltaj Ölçer Alarmlı	1	19,5	TL
20	Breadboard	1	2,05	USD
21	Mini Breadboard (Opsiyonel)	1	0,9	USD
22	Casper N200 Atom 3735 1.83ghz-1gb-16gb-10.1" -W8.1 Notebook Bilgisayar	1	599	TL
23	Logitech c170 Kamera	1	85,46	TL
24	4 Port USB Trust Çoklayıcı	1	12,9	TL
25	Uc-Logic Lapazz A5 grafik tablet	1	99	USD
26	Logitech Extreme 3D Pro Joystick	1	35,8	USD
27	12 mm M3 Sert Plastik Aralayıcı Paketi - 24 Adet -	1	2,55	TL
28	M3 Somun - 50 Adet	4	1,15	TL
29	M3 Pul - 50 Adet	4	1,27	TL
30	M3 50mm YSB Yıldız Silindirik Baş Metrik Vida - 50 Adet - (15mm)	4	1,18	TL
31	Lötfett Lehim Pastası	1	3,5	USD
32	Soldex 0.75 mm 200 gr Lehim Teli (% 60 SN / % 40 Pb)	1	32,45	TL
33	40 Pin Ayrılabilen Dişi-Erkek M-F Jumper Kablo-300 mm	2	3,1	USD
34	40 Pin Ayrılabilen Erkek-Erkek M-M Jumper Kablo-300 mm	1	3,3	USD
35	40 Pin Ayrılabilen Dişi-Dişi F-F Jumper Kablo-300 mm	1	2,4	USD
36	NURSAN oval masa gönyesi	25	12,5	TL
37	2 adet 3 mm kalınlık 1 m ² pleksiglas levha veya 1 adet 3 mm kalınlıkta pleksiglas levha ve 6mm kalınlıkta 1m ² mdf levha	1	100	TL
38	Gijon ebat m8 uzunluk 1m	4	10	TL
39	Somun ebat m8-50 Adet	1	10	TL
40	Pul ebat m8-50 Adet	1	10	TL
41	Pleksi levha 1m ²	1	100	TL
42	Mdf levha 1m ²	1	44	TL
43	Açma kapama ve reset düğmesi	1	2	TL
44	LiPo dişi erkek jack	1	5	TL
45	Cat5 1m kablo	1	4	TL

Tablo 4: Araç gereç adet ve fiyatları

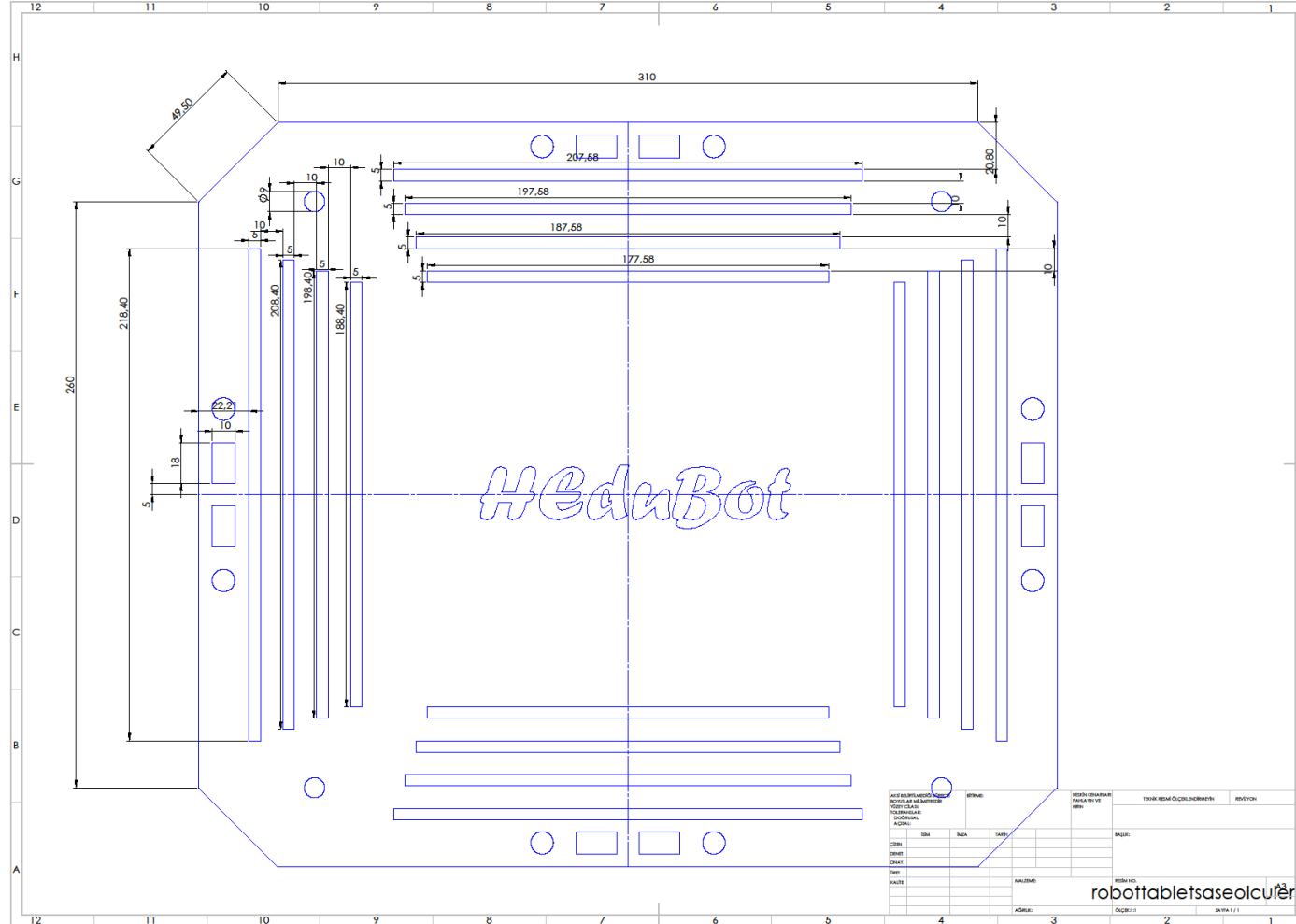
Sıra No	Adı	Miktar Adet	Fiyat	Birim
1	iMAX B6AC Adaptörlü Profesyonel Li-Po, Ni-Mh Şarj Aleti - Balancer (50W)	1	55	USD
2	Havya	1	10,39	USD
3	Havya Standı	1	2	USD
4	Mercekli Yardımcı El - Third Hand	1	15,58	USD
5	Lehim Pompası (Lehim Sökücü)	1	6,68	USD
6	80 Watt Silikon Tabancası G-766	1	16,32	USD
7	Karga Burun Pense	1	8,31	USD
8	Yan Keski	1	10,24	USD
9	Proskit 9 Parça Tornavida Seti	1	7,42	USD
10	4,8 V Şarjlı Vidalama Promo Blister + 24 Parça Aksesuar Seti	1	29,99	TL

8.2. TEKNİK ÇİZİMLER

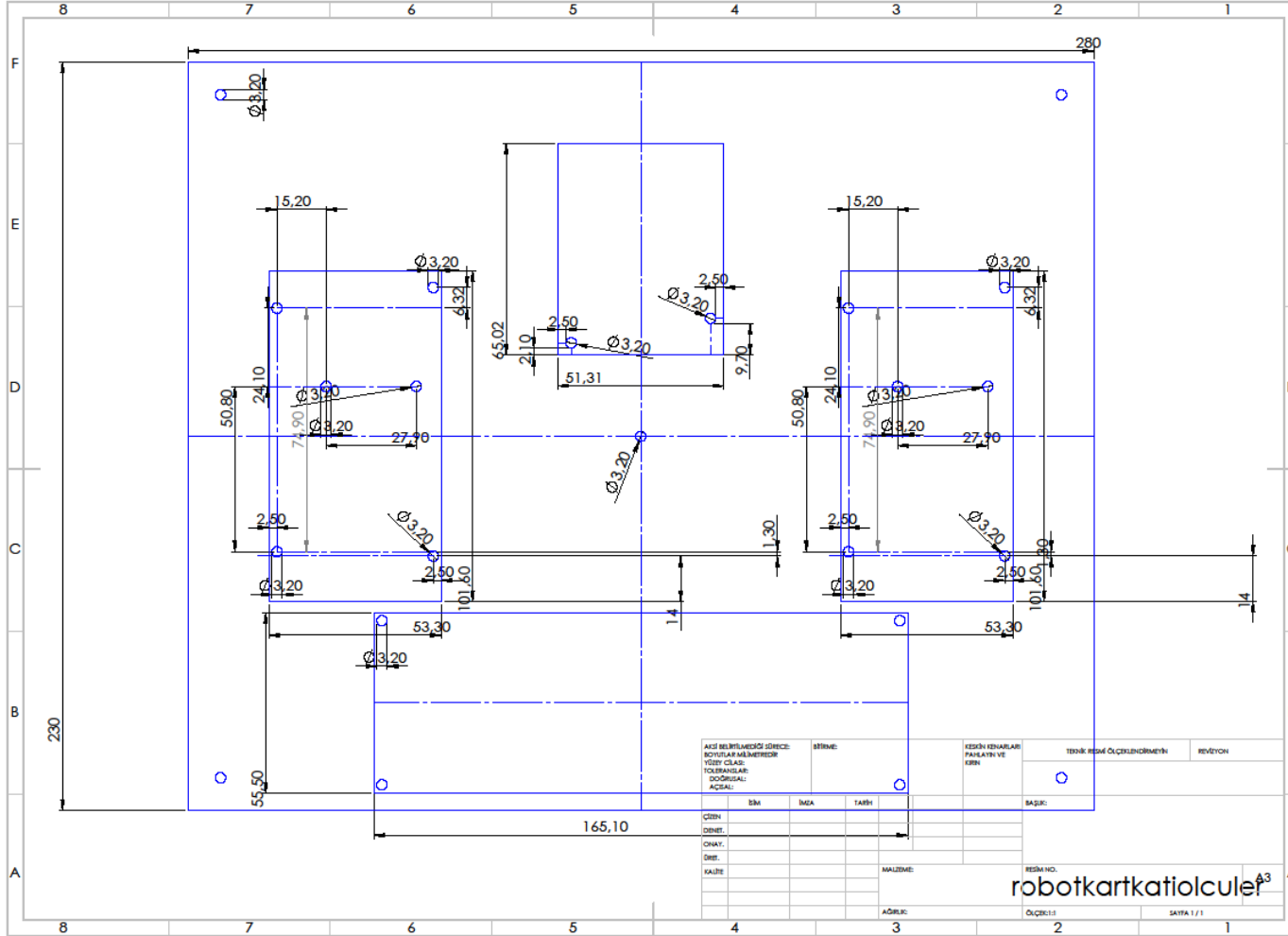
Teknik çizimler, mekanik ve elektronik olarak iki bölüm halinde verilecektir (Şekil 20,21,22,23,24, 25 ve 26). Verilen çizimler gerçek boyutlu olarak çalışmanın ekinde bulunan DVD içerisinde görülebilir.

8.2.1. Mekanik Çizimler

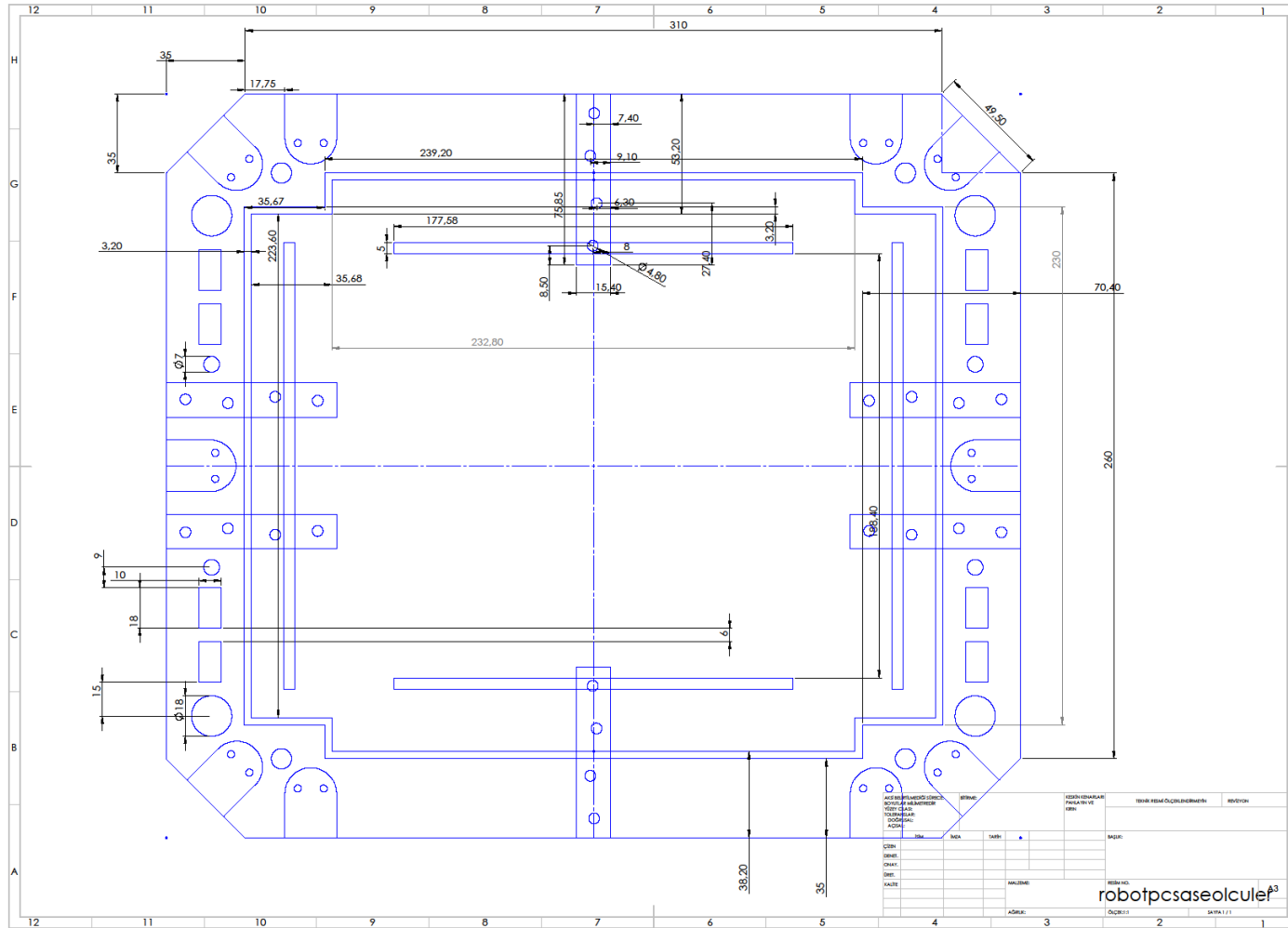
Mekanik olarak robotta kullanılan katmanların teknik çizimleri bu bölümde verilmektedir.

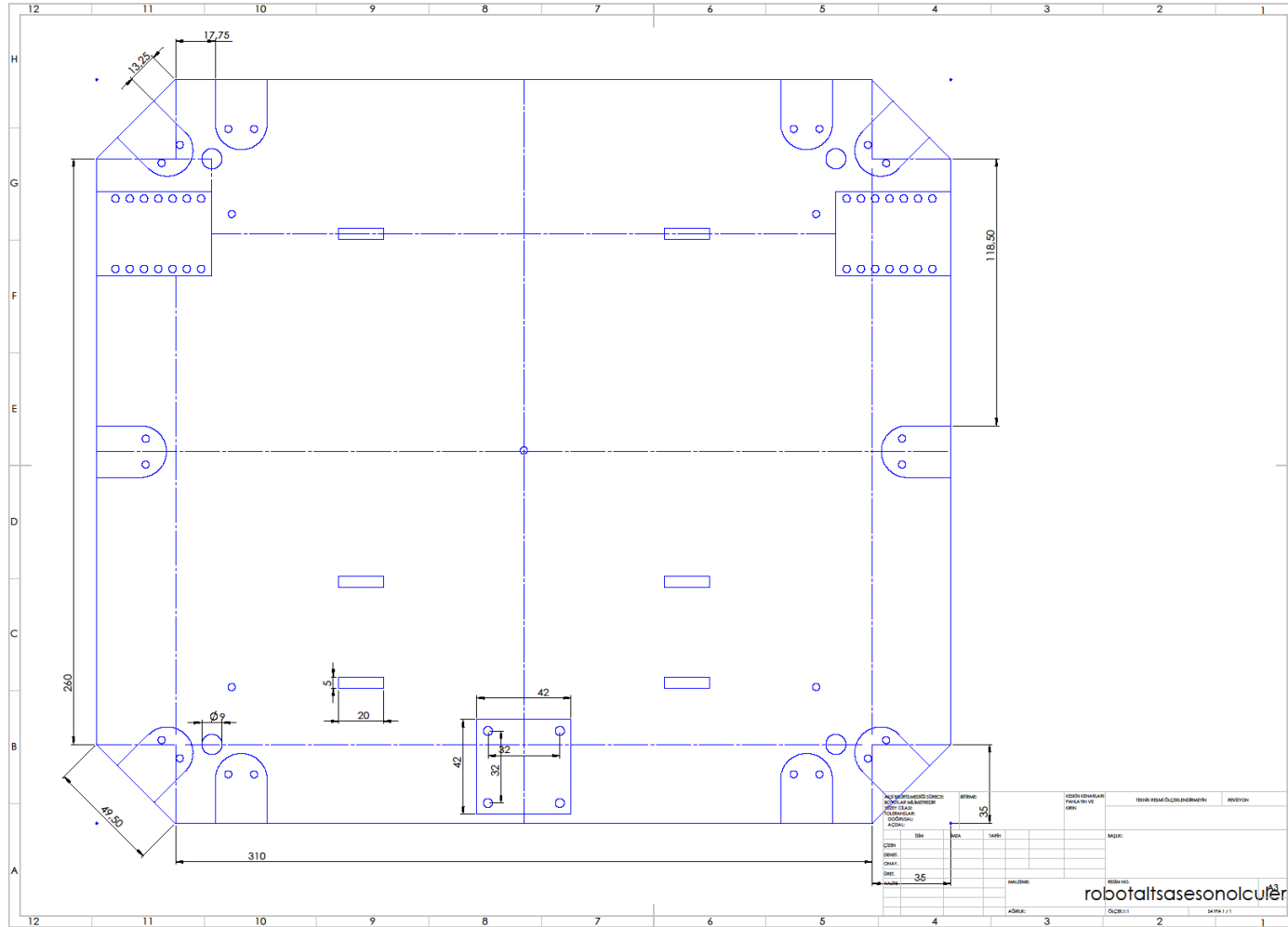


Şekil 20: Tablet katmanı teknik çizimi

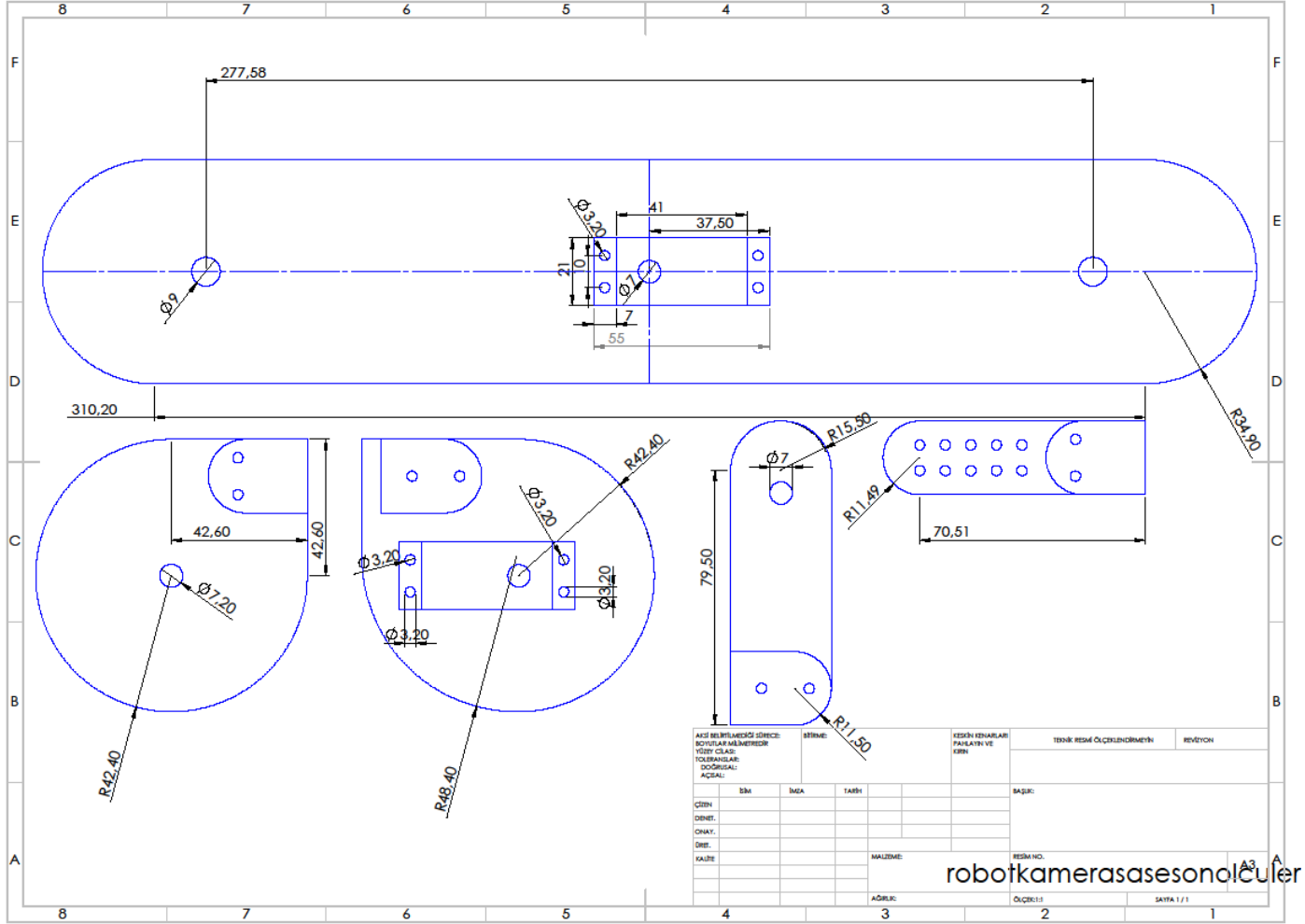


Şekil 21: Kart katmanı teknik çizimi



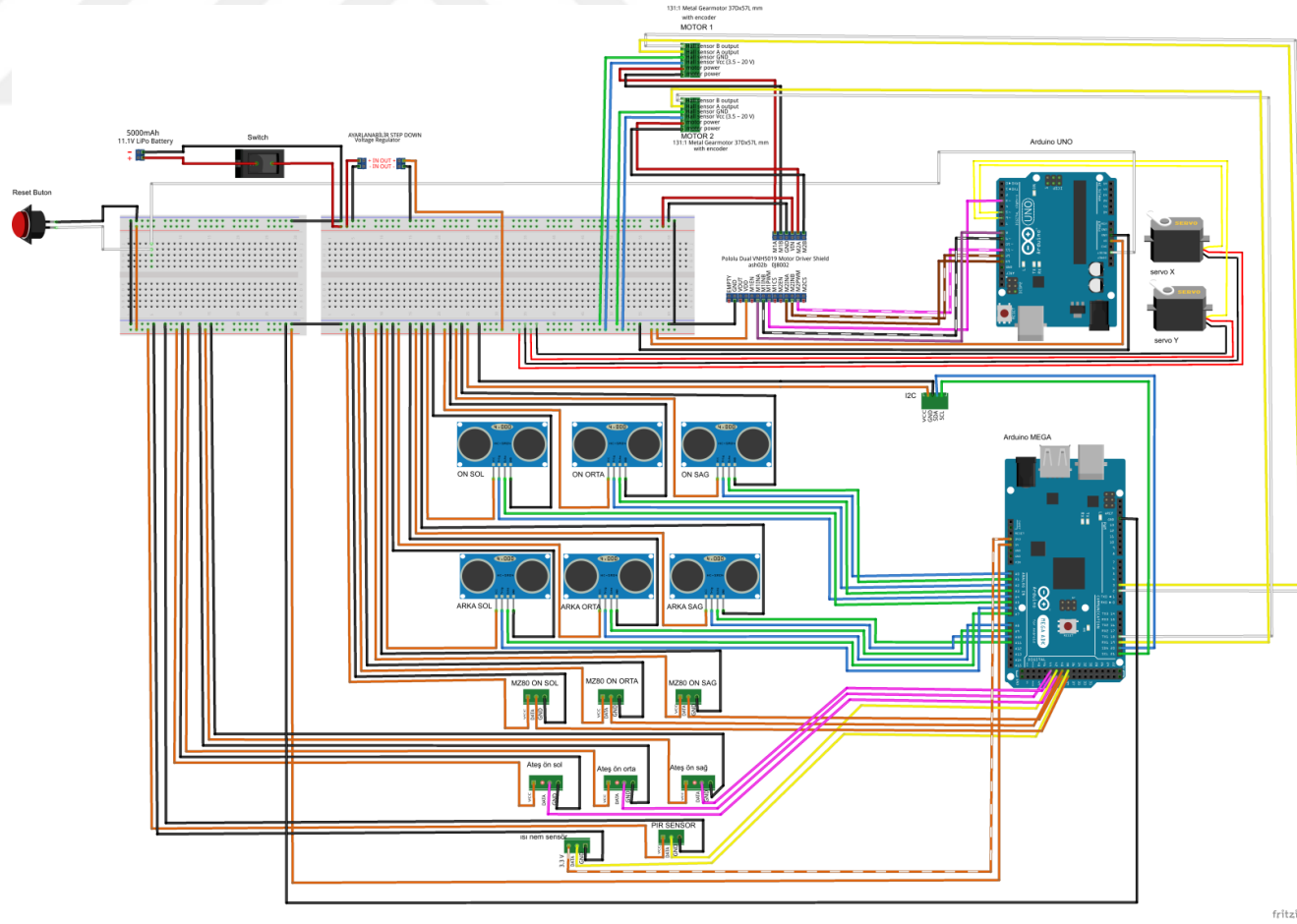


Şekil 24: Motor ve teker katmanı teknik çizimi



Şekil 25: Kamera katmanı ve pan tilt teknik çizimi

8.2.2. Elektronik Çizimler



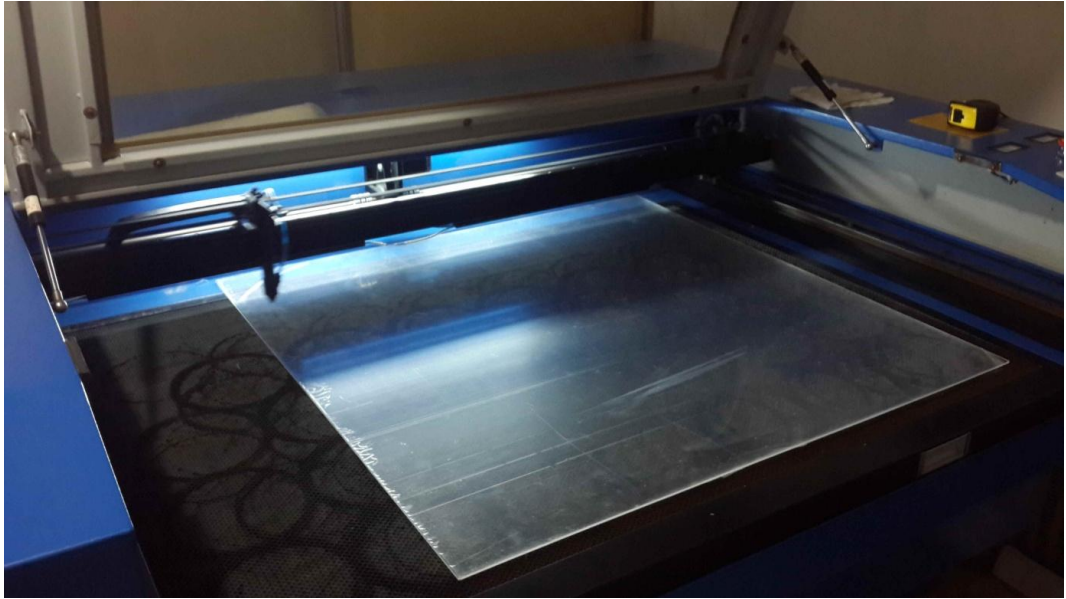
Şekil 26: Elektronik bağlantı şeması

8.3. MONTAJ VE ÇALIŞTIRMA

Robotun hazırlanması montaj ve çalıştırma başlıkları altında incelenecektir.

8.3.1. Montaj

Öncelikle ekte verilen teknik çizimlerdeki katmanlardan tablet, bilgisayar, motor ve teker katmanı 6mm mdf veya pleksi levhadan kesilir. Sensör ve ekran yan panelleri, kamera katmanı ve pan tilt 3mm pleksi levhadan kesilir. Bu işlem lazer veya CNC kesim ile yapılabilir (Şekil 27 ve 28). Kesim elde yapılacak ise ek DVD içerisinde verilen gerçek ölçülü teknik çizimden dikkatli bir şekilde yapılması gerekmektedir.



Şekil 27: Lazer kesim için levhaların yerleştirilmesi



Şekil 28: Lazer kesim işleminin tamamlanması

HEduBot montajı için gerekli malzemeler ekte kullanılan parçalar ve araç gereç listesinde verilmiştir. Bu araçlar montaj esnasında hazır bulunmalıdır (Şekil 29).



Şekil 29: HEduBot için gerekli malzemelerin kontrolü

6 mm kalınlıktaki motor ve teker katmanının oluřturulması için katmanın merkezindeki ve 8mm'lik gijon deliklerinin i taraflarındaki deliklere m3 (3mm) apta 5cm'lik cıvatalar yerleřtirilir (Őekil 30).



Őekil 30: Motor ve teker katmanının oluřturulmasında kart katı tutucu cıvata yerleřimi

Motor tutucular ile motorlar birleřtirilir. Bunun için motor tutucular ile birlikte gelen m3 cıvatalar kullanılabilir. Motor millerinin Őekil 30'daki gibi ũst tarafta olmasına dikkat edilmelidir.



Şekil 31: Motor tutucu montajı

Oval masa gönyeleri kenarlardaki deliklere gelecek şekilde m3 çapta 15mm'lik uzunlukta cıvatalar ile bağlanır. Aynı işlem bilgisayar katmanında da yapılmalıdır (Şekil 32 ve 33).



Şekil 32: Motor ve teker katmanının oluşturulmasında oval masa gönyesi montajı

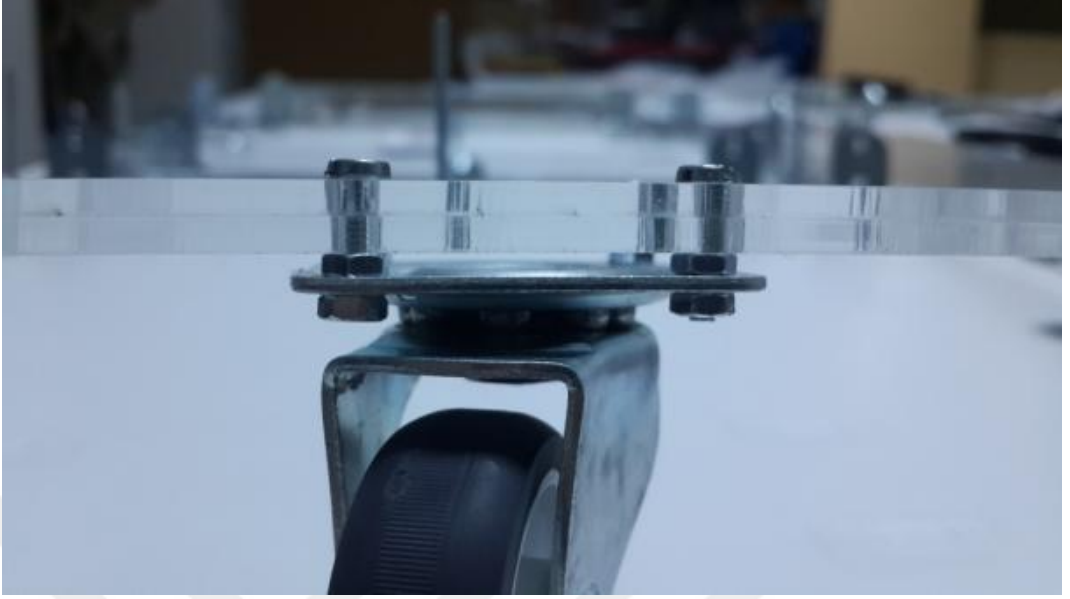


Şekil 33: Bilgisayar katmanında oval masa gönyesi montajı

Robot şasisine sarhoş teker, şasiden ayırmak amaçlı ek bir somun ve 15mm'lik m3 çapta 4 adet cıvata yardımıyla eklenir (Şekil 34 ve 35).

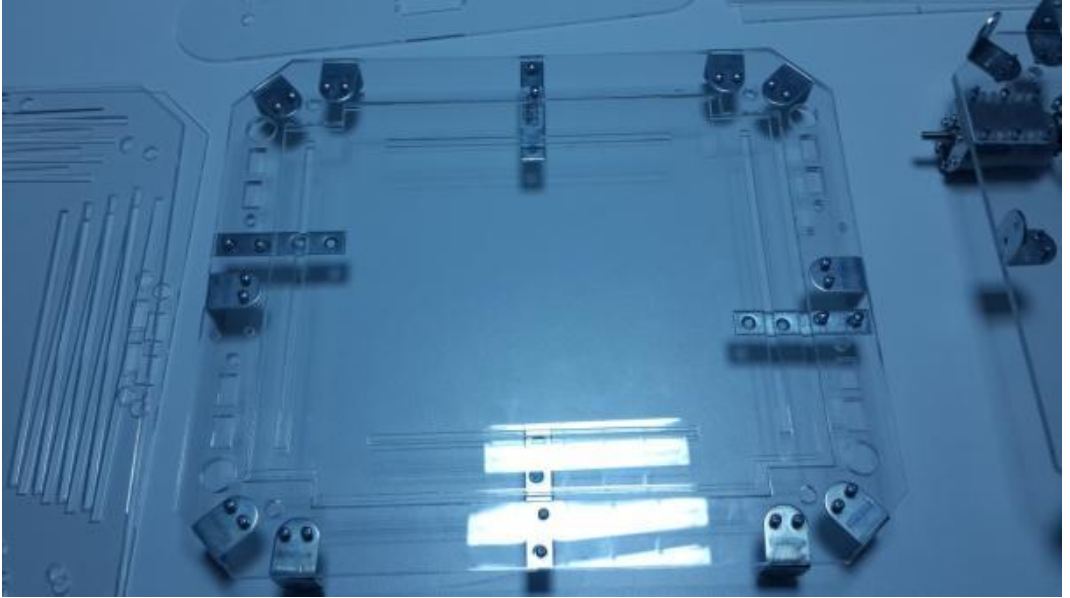


Şekil 34: Motor ve teker katmanının oluşturulmasında sarhoş teker montajı (a)



Şekil 35: Motor ve teker katmanının oluşturulmasında sarhoş teker montajı (b)

Bilgisayar katmanında kapağı tutması için düz gönye 15mm'lik m3 cıvatarla tutturulur (Şekil 36).

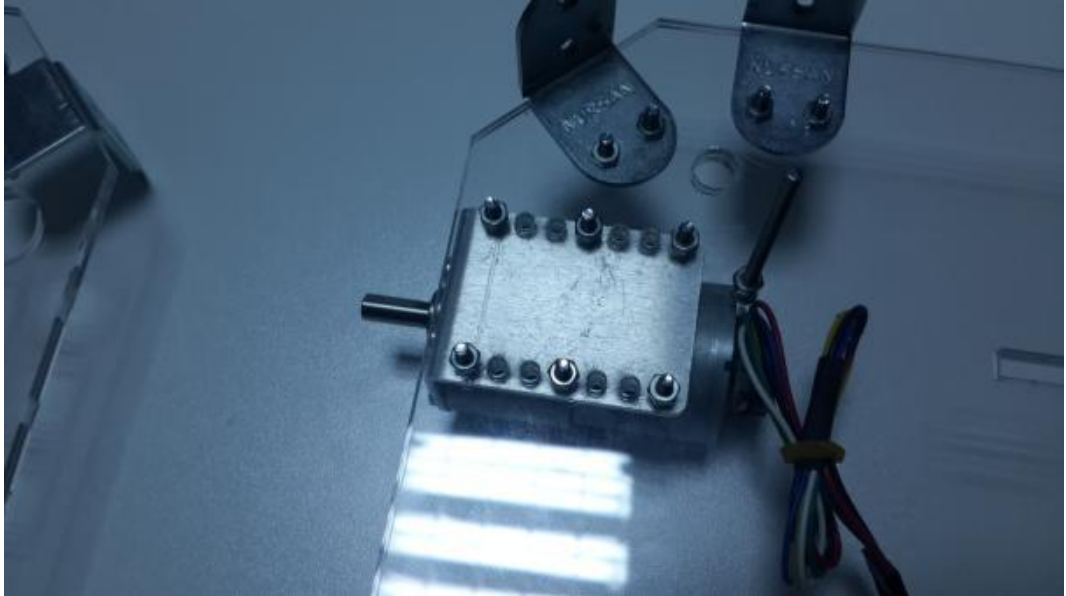


Şekil 36: Bilgisayar katmanı düz gönye montajı

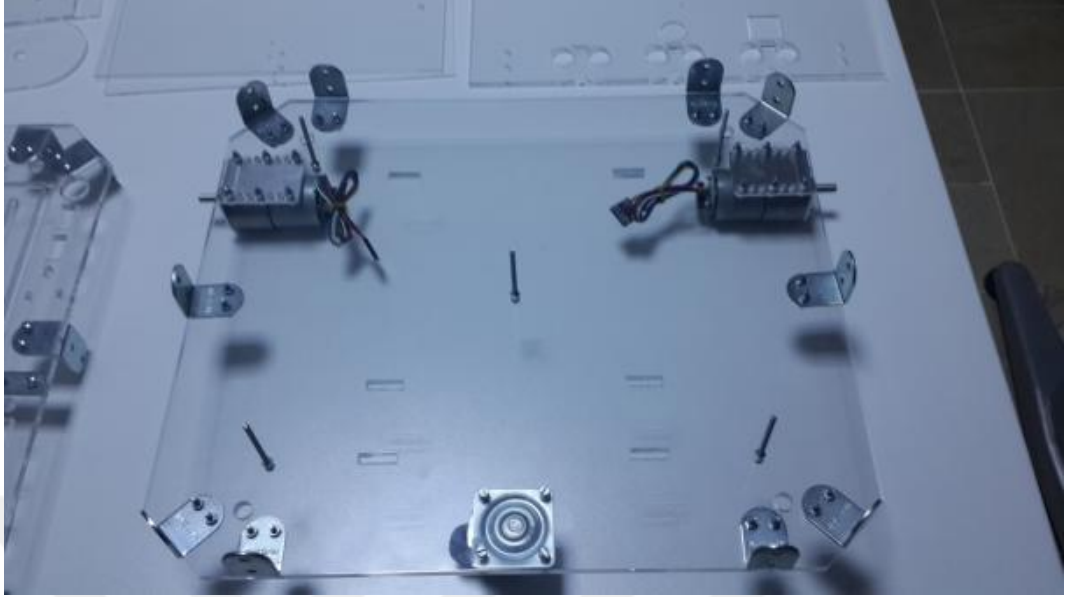
Daha önce birleřtirilen motor tutucular řasiye m3, 15mm'lik cıvatalar ile tutturulur (řekil 37, 38 ve 39).



řekil 37: Motor ve teker katmanına motor tutucuların montajı (a)



řekil 38: Motor ve teker katmanına motor tutucuların montajı (b)



Şekil 39: Motor ve teker katmanına motor tutucuların montajı (c)

Plastik ayırıcı ile Arduino kartlar, motor sürücü ve voltaj düşürücü kart 5cm'lik m3 civata ile tutturulur (Şekil 40).



Şekil 40: Plastik aralayıcıların kartlara montajı

Daha sonra kartlar ve breadboard kart katına monte edilir. Breadboard için alt kısmında bulunan yapışkan yüzey montaj için kullanılır (Şekil 41).



Şekil 41: Kartların kart katına montajı

Ultrasonik mesafe sensörleri, ateş sensörleri, hareket sensörü ve 16x2 LCD ekran silikon yardımı ile sensör ve ekran yan panellerindeki yuvalarına yerleştirilir (Şekil 42, 43, 45 ve 46).

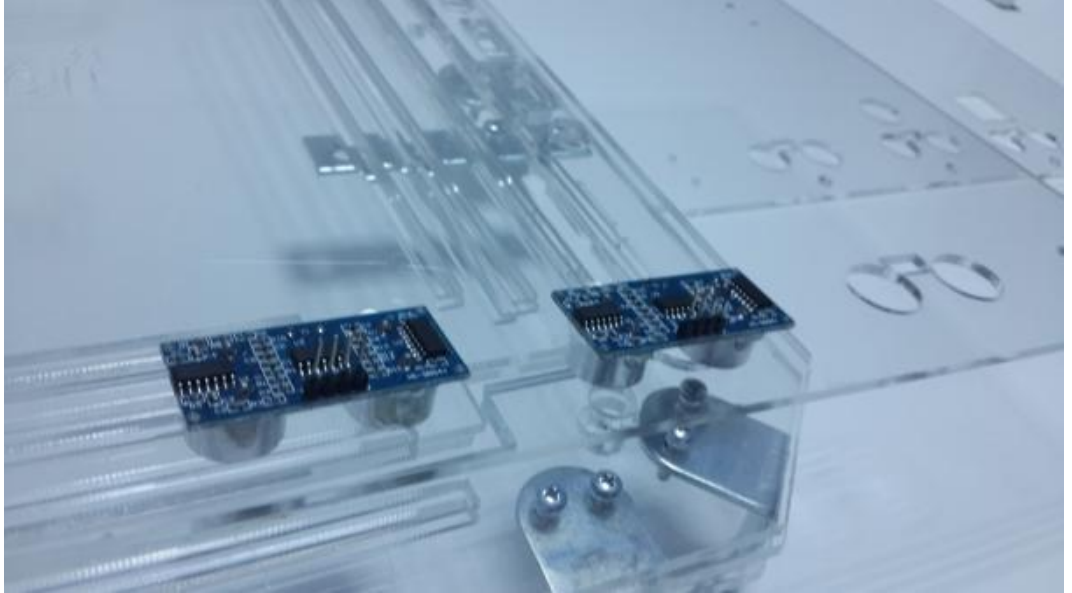


Şekil 42: Sensörlerin silikon ile yan panellere montajı (a)

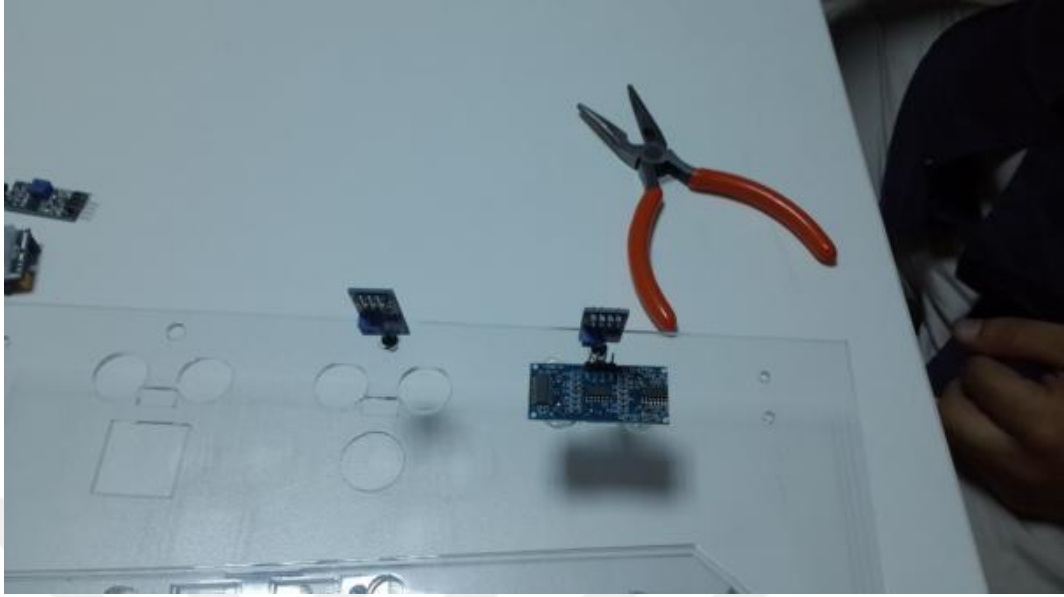


Şekil 43: Sensörlerin silikon ile yan panellere montajı (b)

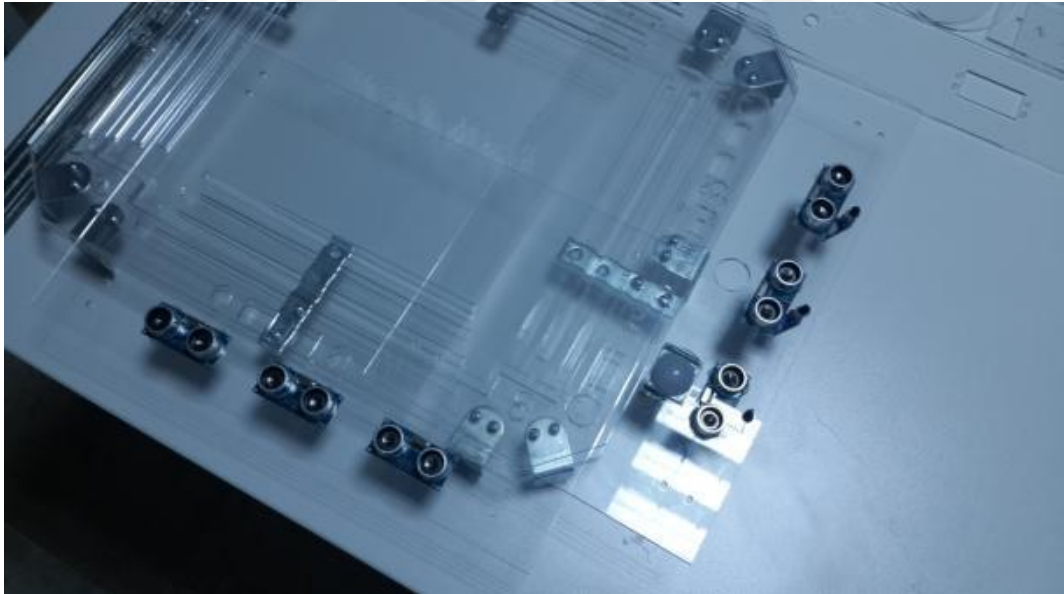
Ultrasonik sensörlerin pinlerinin yan panellere dik olmasına dikkat edilmelidir (Şekil 44).



Şekil 44: Sensörlerin silikon ile yan panellere montajı (c)

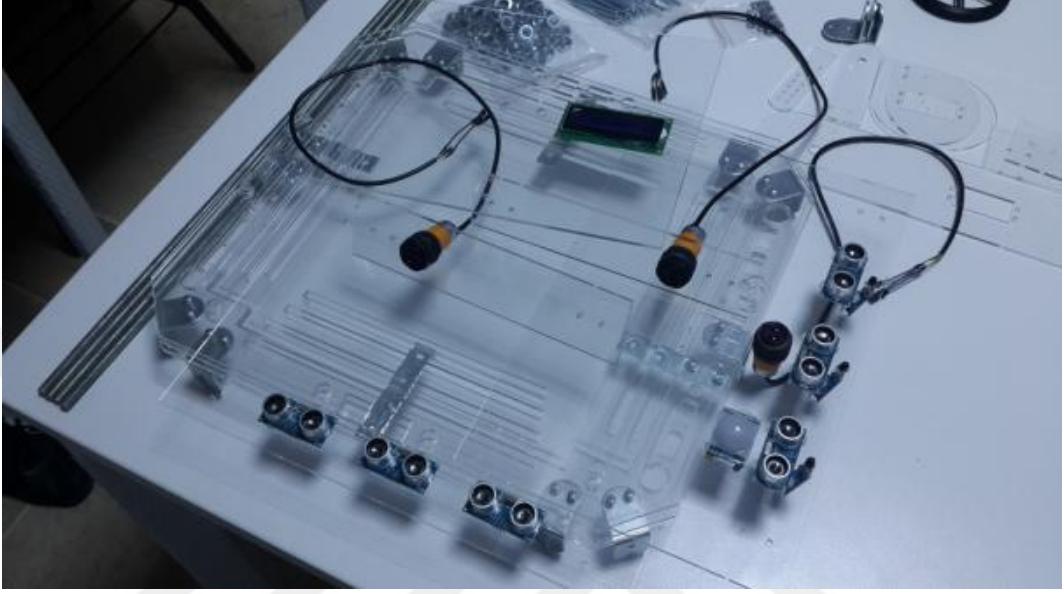


Şekil 45: Sensörlerin silikon ile yan panellere montajı (d)



Şekil 46: Sensörlerin silikon ile yan panellere montajı (e)

Daha sonra IR (Kızılötesi) engel sensörleri ön ve yan panellere yerleştirilir ve kendi tutucusu yardımı ile sıkıştırılır (Şekil 47).



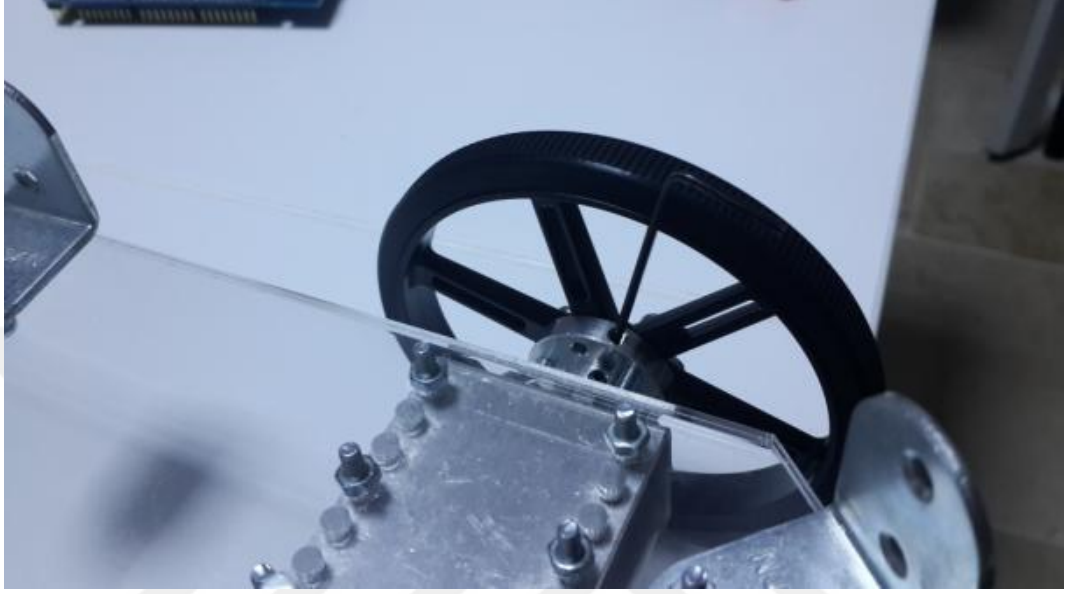
Şekil 47: IR sensörlerin montajı

Tekerlere flanşlar 15mm'lik m3 cıvatalar ile tutturulur (Şekil 48).

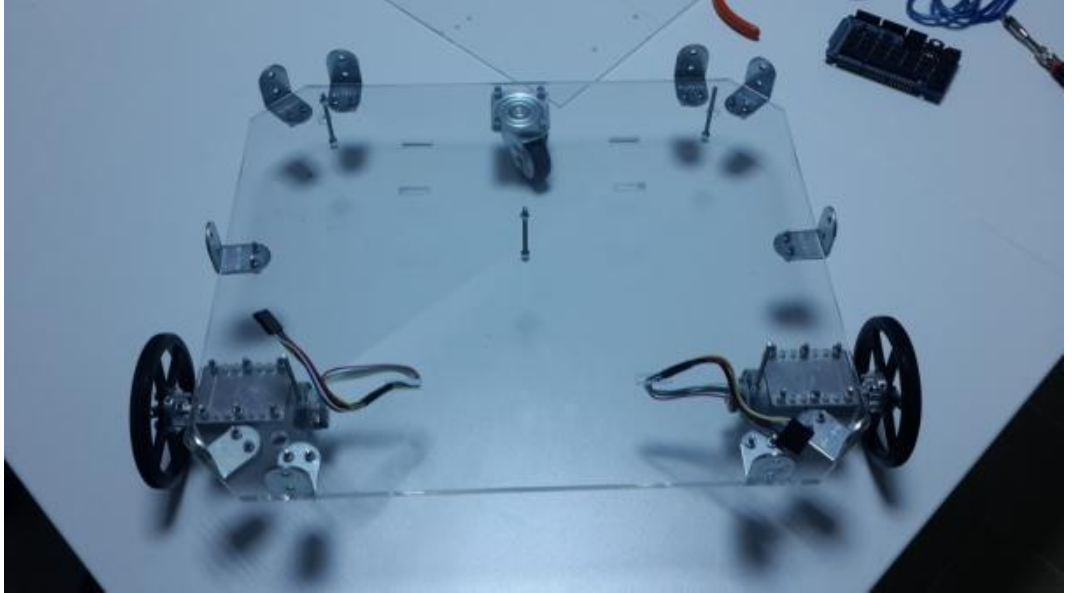


Şekil 48: Teker flanş montajı

Daha sonra tekerler setskurlu vidalar ile motor şaftına tutturulur (Şekil 49 ve 50).

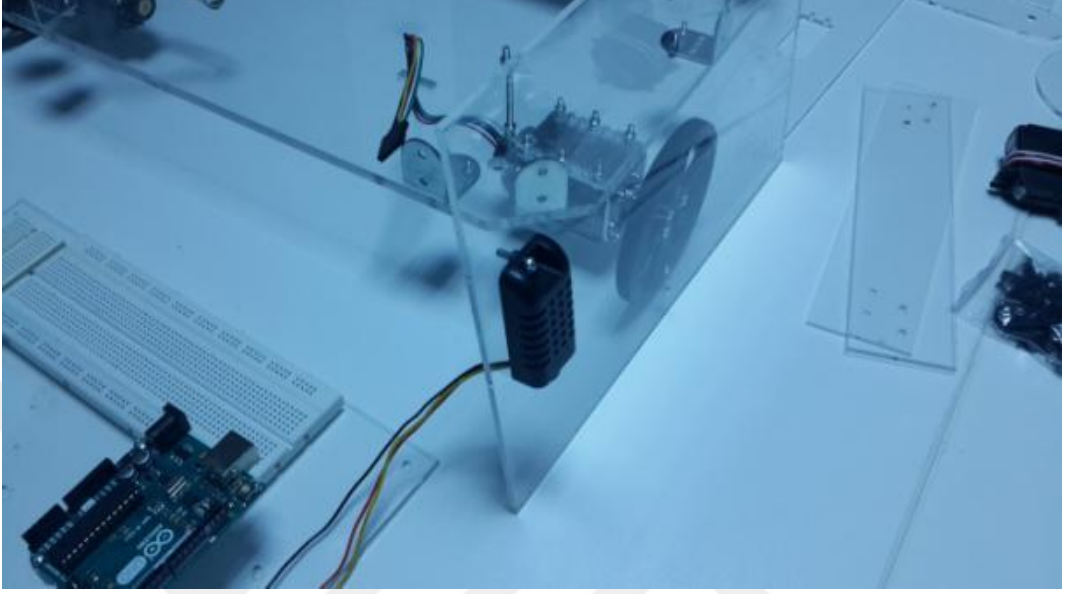


Şekil 49: Teker motor montajı



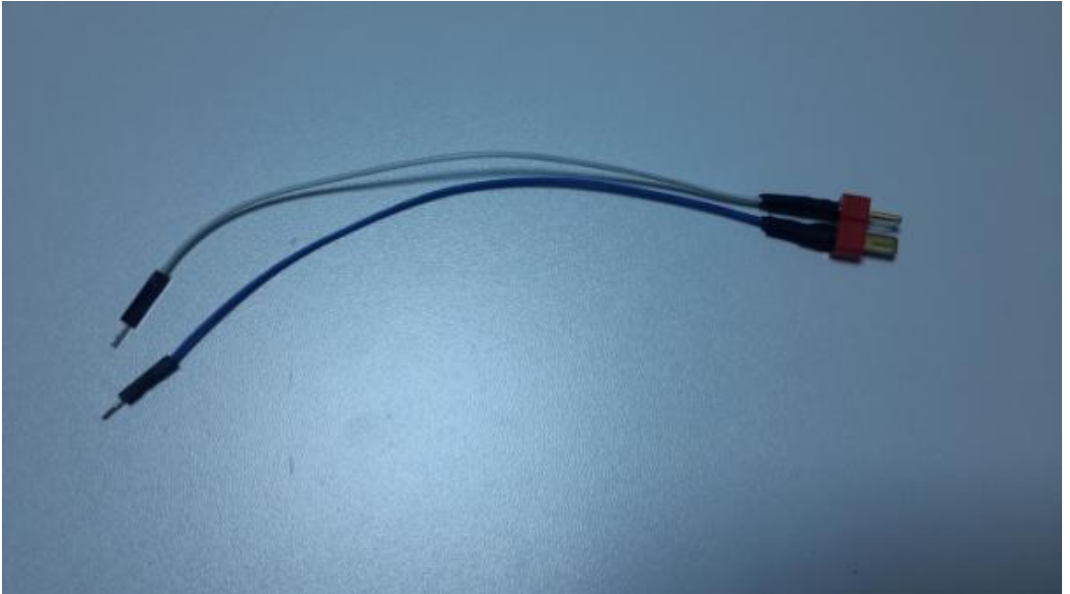
Şekil 50: Motor ve teker katmanının tamamlanmış hali

Isı ve nem sensörü de sensör ve ekran yan paneli üzerindeki deliklere yerleştirilir (Şekil 51).



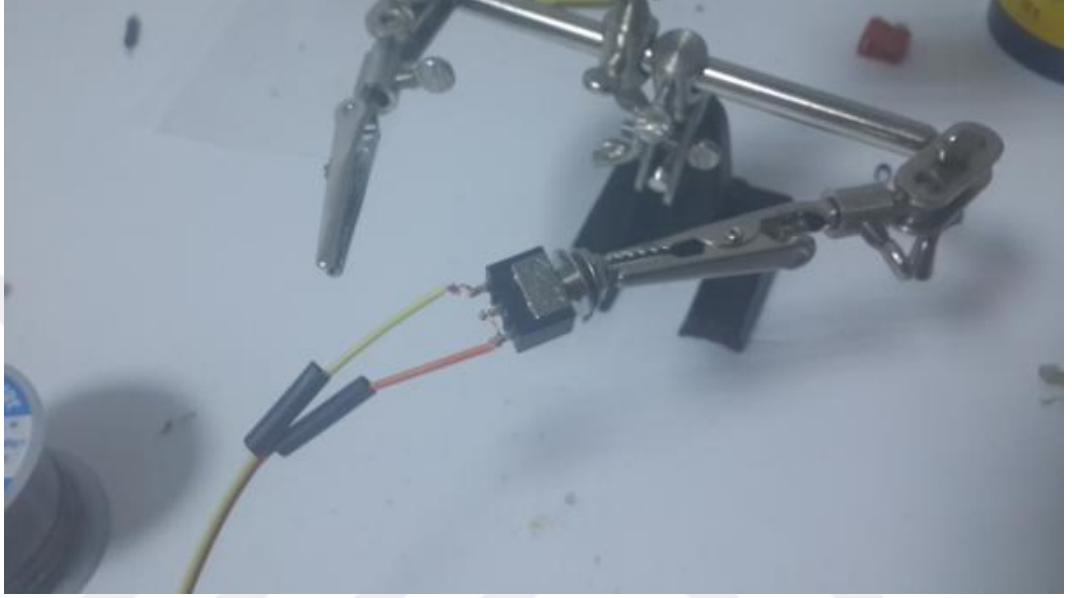
Şekil 51: Isı ve nem sensörü montajı

Güç için LiPo fişi erkek-pin bağlantısı yapabilecek şekilde hazırlanır (Şekil 52).

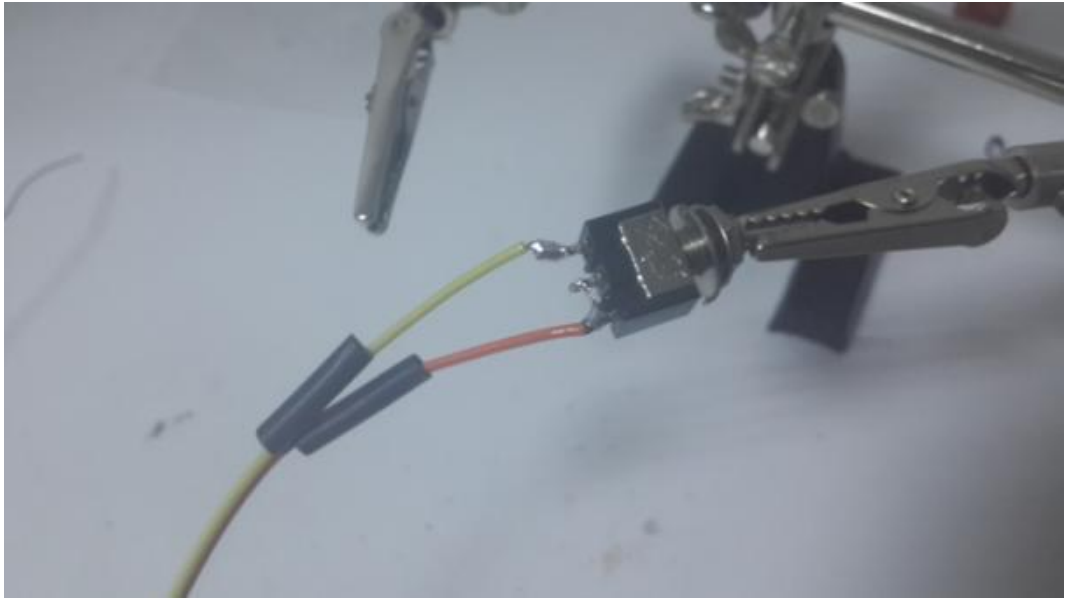


Şekil 52: LiPo güç kablosu montajı

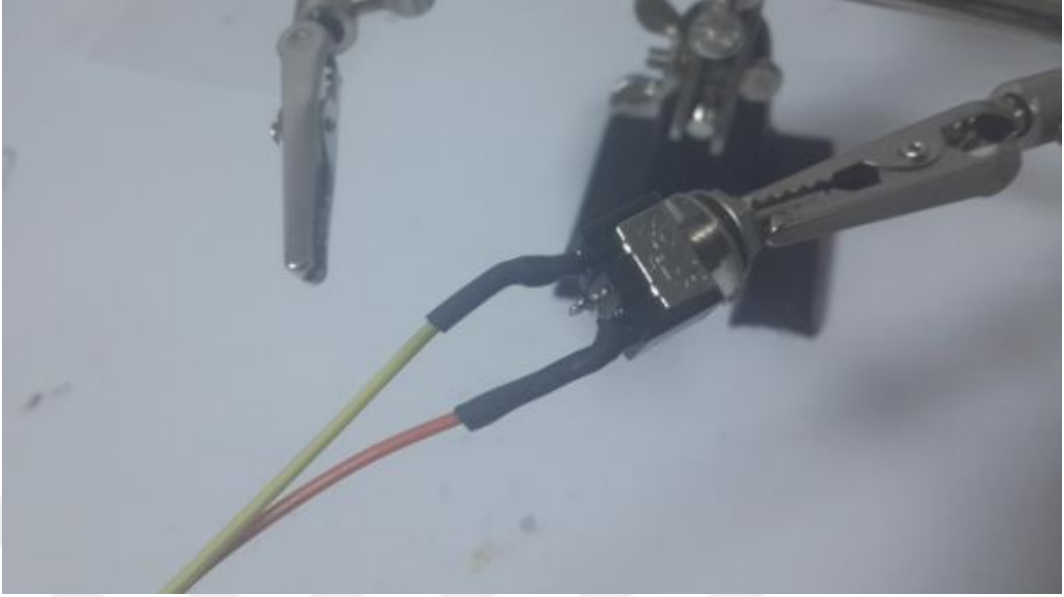
Açma kapama düğmesi ve yeniden başlatma düğmesi erkek pinleri olacak şekilde hazırlanır. Bu bağlantılarda makaron adı verilen yalıtım malzemesi kullanımına dikkat ederek kısa devre önlenmelidir (Şekil 53, 54, 55 ve 56).



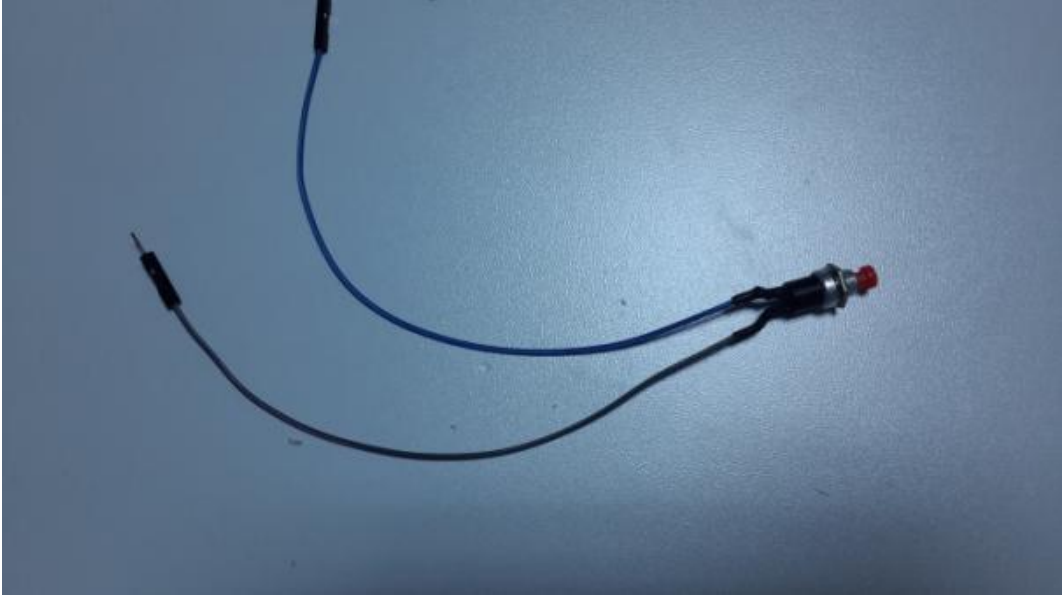
Şekil 53: Açma kapama düğmesinin hazırlanışı (a)



Şekil 54: Açma kapama düğmesinin hazırlanışı (b)

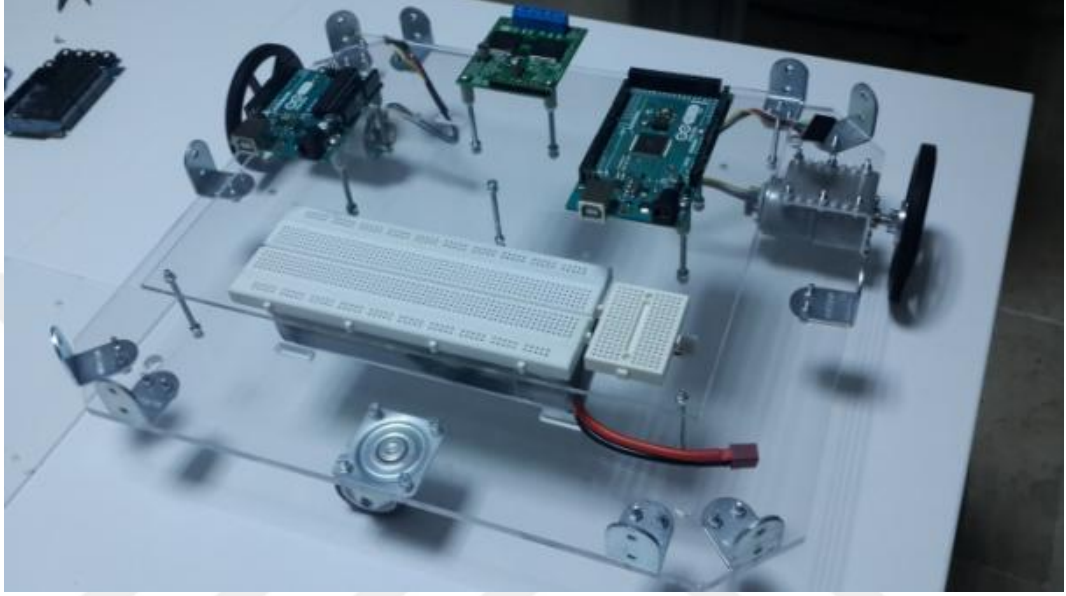


Şekil 55: Açma kapama düğmesinin hazırlanışı (c)



Şekil 56: Reset düğmesi

Kart katmanı motor teker katmanı üzerindeki yere oturtulur, sensörlerin bulunduğu yan paneller açık ve alt şasi yanlarına uygun şekilde yerleştirilir (Şekil 57).



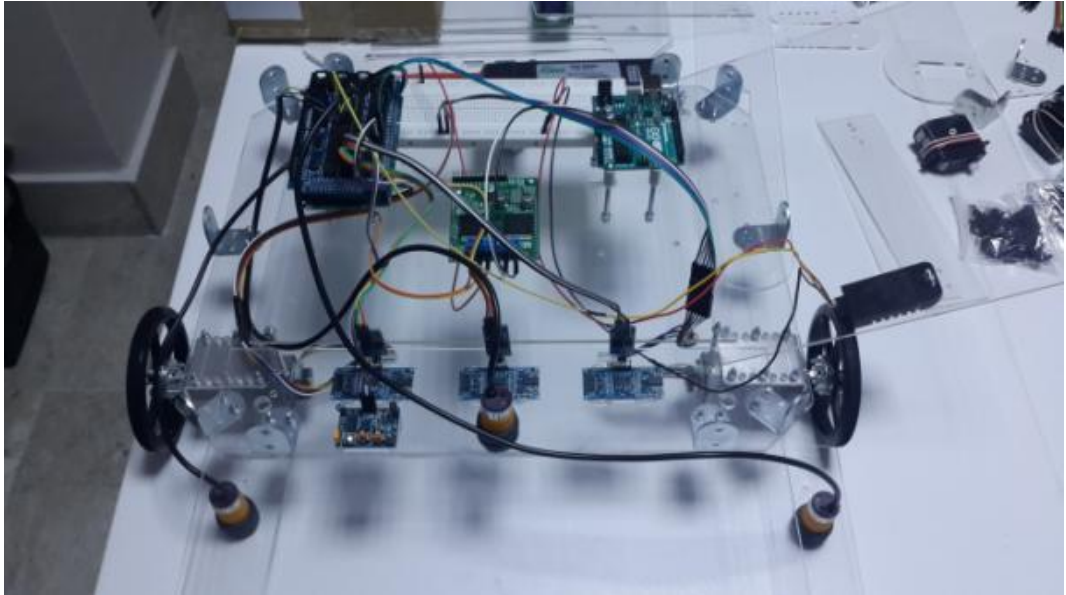
Şekil 57: Kart katnının motor teker katmanına montajı

Ayarlanabilir step down voltaj regülatör kartı LiPo pilden gelen 12V voltajı 5V'a düşürecek şekilde ayarlanır ve kart katmanının üzerine yerleştirilir (Şekil 58). LiPo ile yapılan tüm çalışmalarda ters voltaj, kısa devre, yanlış şarj ve benzeri tehlikelere dikkat etmek gereklidir aksi takdirde size ve donanımınıza zarar verebilir. Bu nedenle şarj işleminde ve bağlantı işleminde gerekli güvenlik önlemleri mutlaka alınmalıdır.

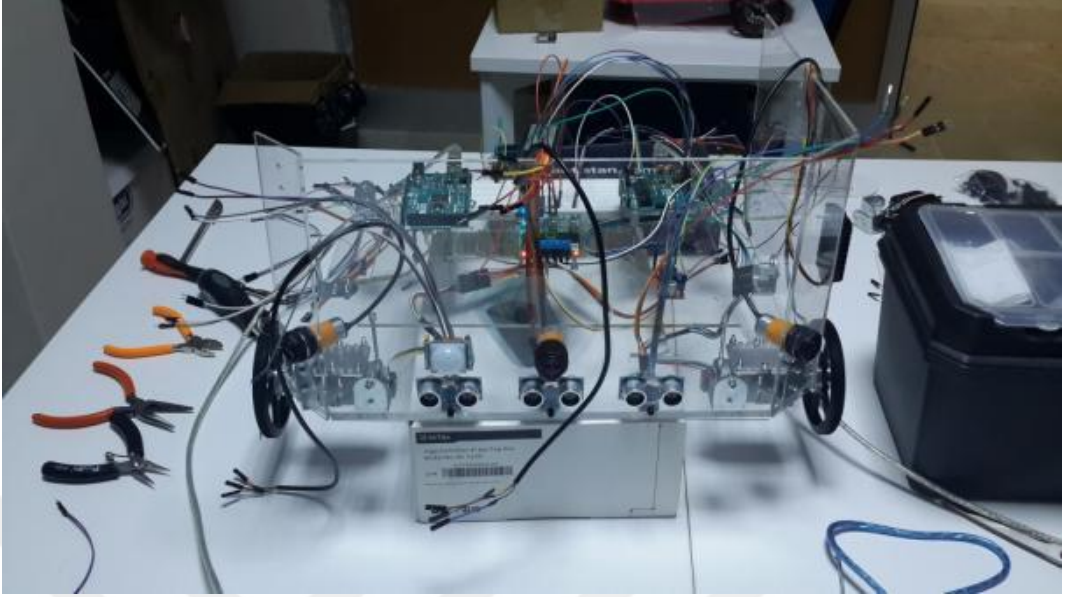


Şekil 58: Step down voltaj regülatör kartının 5v'a ayarlanması

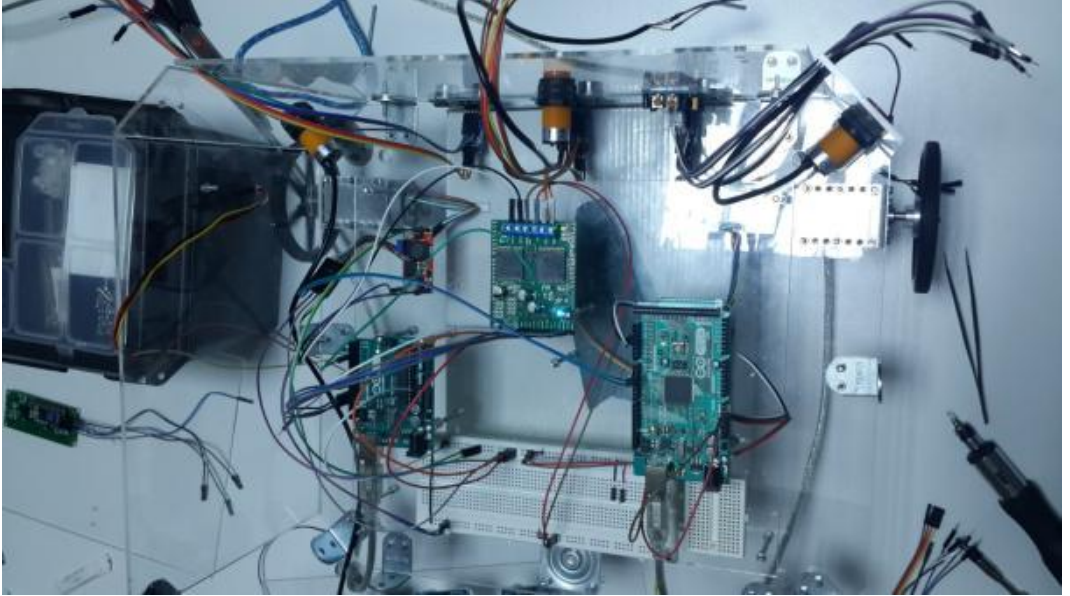
Ekte bulunan bağlantı şeması yardımı ile bağlantı işlemi yapılır (Şekil 59, 60, 61 ve 62).



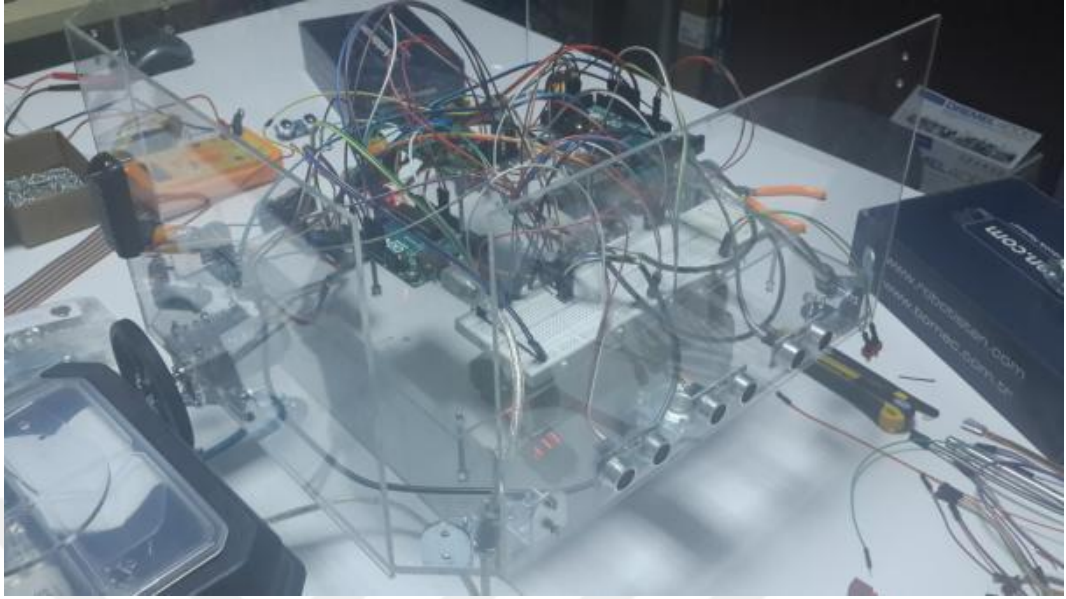
Şekil 59: Sensör montajı (a)



Şekil 60: Sensör montajı (b)



Şekil 61: Sensör montajı (c)



Şekil 62: Sensör montajı (ç)

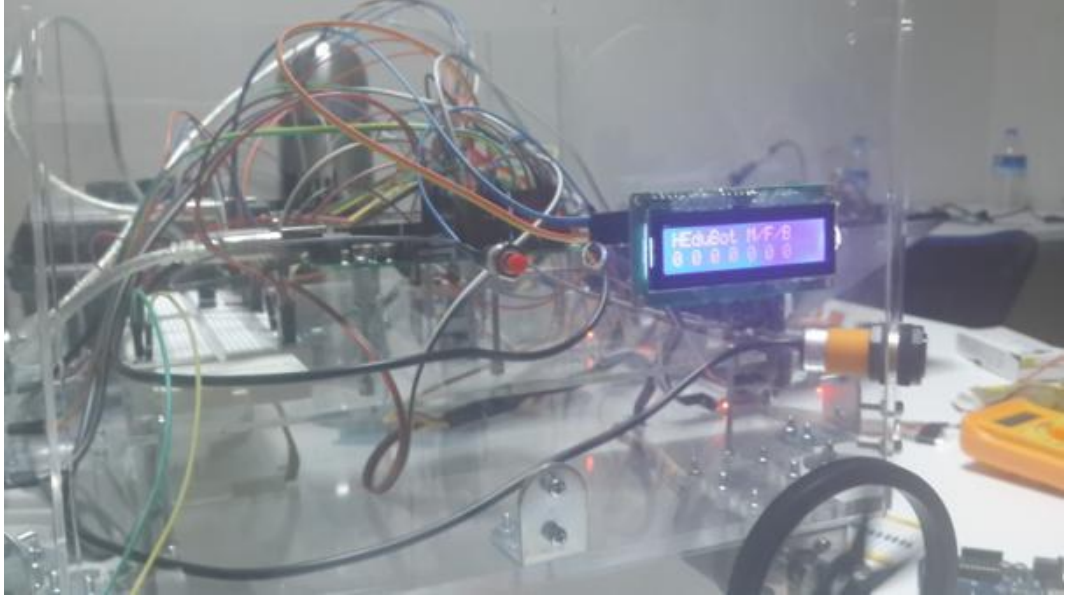
Sensör bağlantılarını ve motor bağlantılarını tamamladıktan sonra ekte verilen "SensorVeri.ino" ve "MotorKontrol.ino" uygulamaları ile bağlantılar kontrol edilerek varsa hatalı bağlantılar düzeltilir (Şekil 63, 64 ve 65).



Şekil 63: Sensör ve motor testi (a)



Şekil 64: Sensör ve motor testi (b)



Şekil 65: Sensör ve motor testi (c)

Bilgisayar katmanı oval masa gönyeleri ve m3 çapta 15mm cıvatalarla monte edilir varsa hatalı bağlantılar kontrol edilerek düzeltilir (Şekil 66, 67 ve 68).



Şekil 66: Hata düzeltme ve kontrol (a)

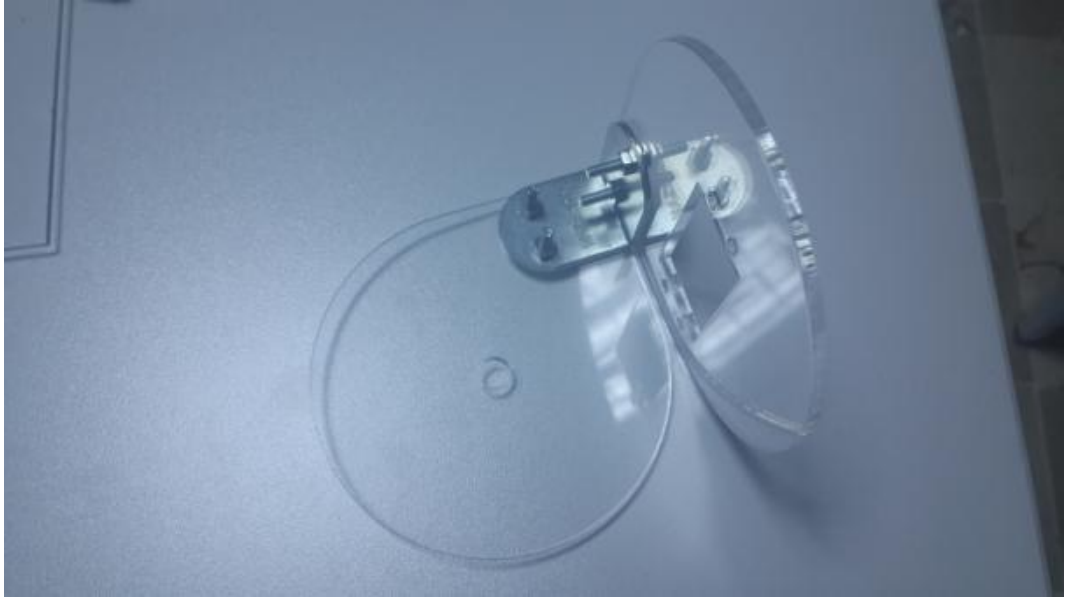


Şekil 67: Hata düzeltme ve kontrol (b)



Şekil 68: Bilgisayar katmanı montajı

Sonrasında pan tilt montajı yapılabilir. Kamera katmanı ve pan tilt montajında m3 çapta 15mm uzunlukta civata kullanılır (Şekil 69 ve 70).



Şekil 69: Pan tilt montajı (a)



Şekil 70: Pan tilt montajı (b)

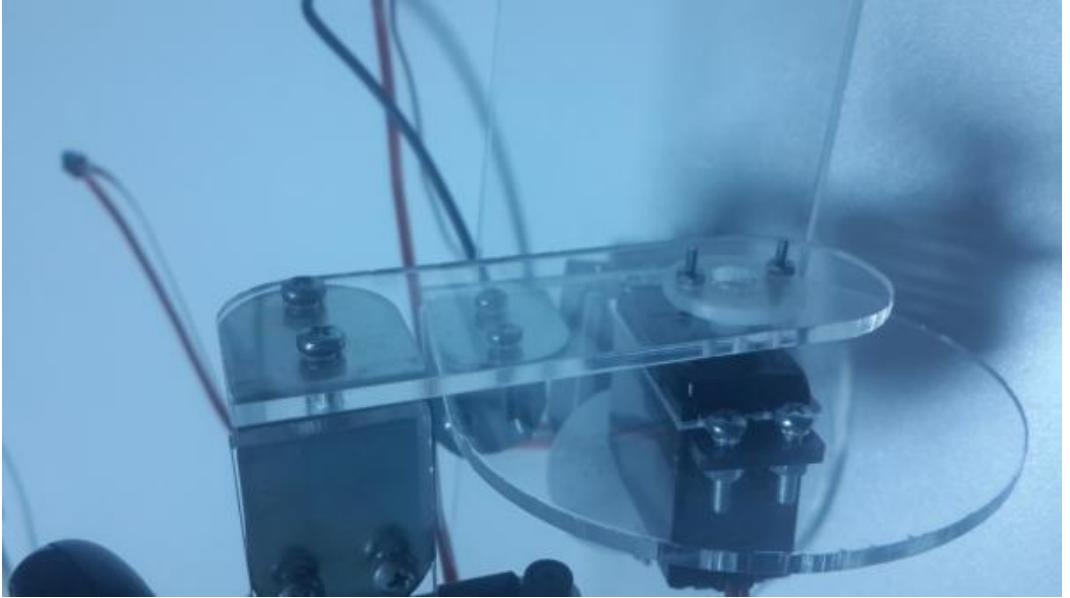
Servoların plastik başlıkları pan tilt mekanizmasının ilgili yerlerine matkapla delik açılarak yerleştirilir (Şekil 71, 72 ve 73).



Şekil 71: Servo başlık montajı (a)

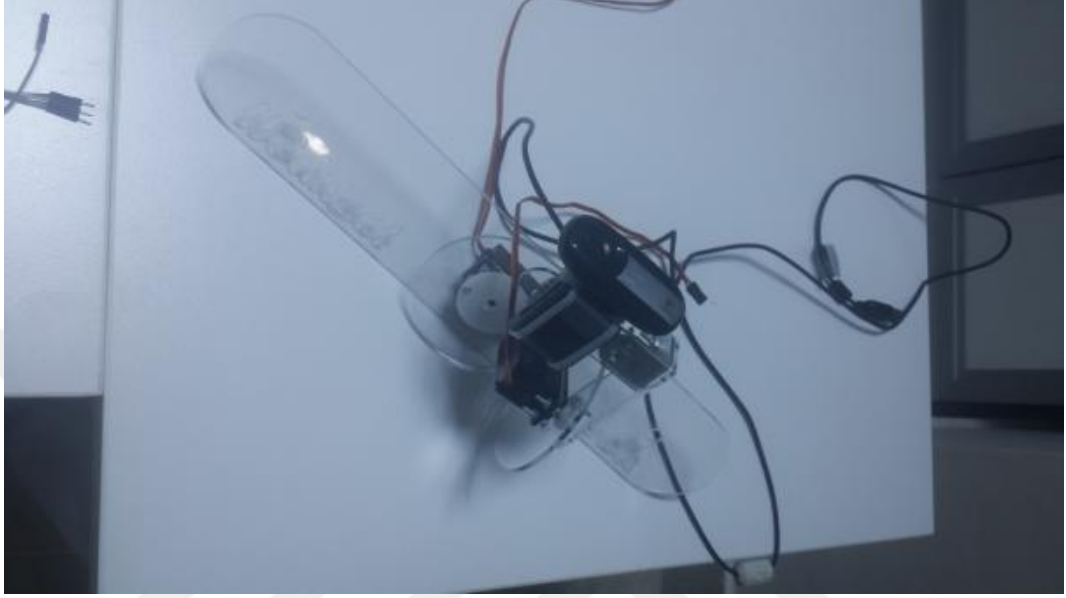


Şekil 72: Servo başlık montajı (b)



Şekil 73: Servo başlık montajı (c)

Pan tilt mekanizması üzerine kamera kelepçe yardımı ile takılarak kamera katmanı ve pan tilt tamamlanır (Şekil 74 ve 75).



Şekil 74: Kamera montajı (a)



Şekil 75: Kamera montajı (b)

Alt kısma m8 (8mm) gijonlar köşelerdeki deliklere yine m8 (8mm) somun ve pullar yardımı ile sabitlenir ve kamera katmanı ve pan tilt ön taraf gijonlarına takılarak istenilen yükseklikte ayarlanır (Şekil 76).



Şekil 76: Gijon ve kamera katmanı montajı

Sonrasında tablet katmanı istenilen yükseklikte ayarlanıp gijonlara m8 pul ve somunlar ile sabitlenir (Şekil 77).



Şekil 77: Tablet katmanı montajı

Robotun üstü örtülerek artan gijon varsa demir testeresi veya avuç taşlama benzeri araçlar ile kesilir. Bu işlem sırasında gözlük ve eldiven mutlaka kullanılmalıdır (Şekil 78, 79, 80 ve 81).



Şekil 78: Artık Gijonların kesimi (a)



Şekil 79: Artık gijonların kesimi (b)



Şekil 80: Artık gijonların kesimi (c)



Şekil 81: Artık gijonların kesiminin tamamlanması

Tablet katmanı üzerine tablet yerleřtirilerek sabitlenir (řekil 82).



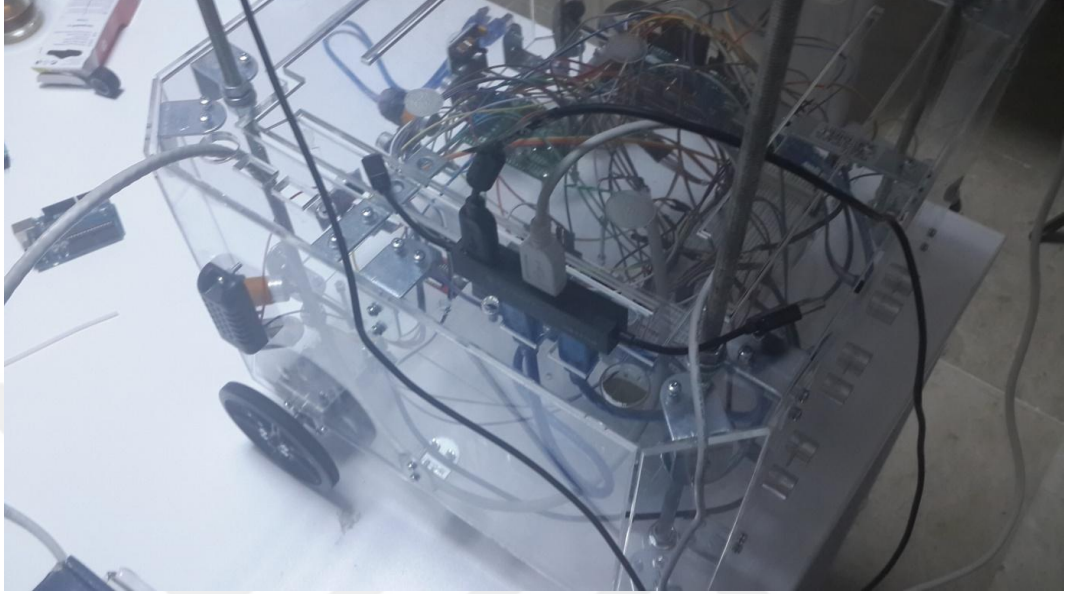
řekil 82: Tablet montajı

Servo X ve servo Y bağlantısı yapılarak pan tilt mekanizması montajı tamamlanır (Şekil 83).



Şekil 83: Servo bağlantısı

Eğer gerekiyorsa USB çoklayıcısı sisteme eklenir (Şekil 84).



Şekil 84: USB çoklayıcı montajı

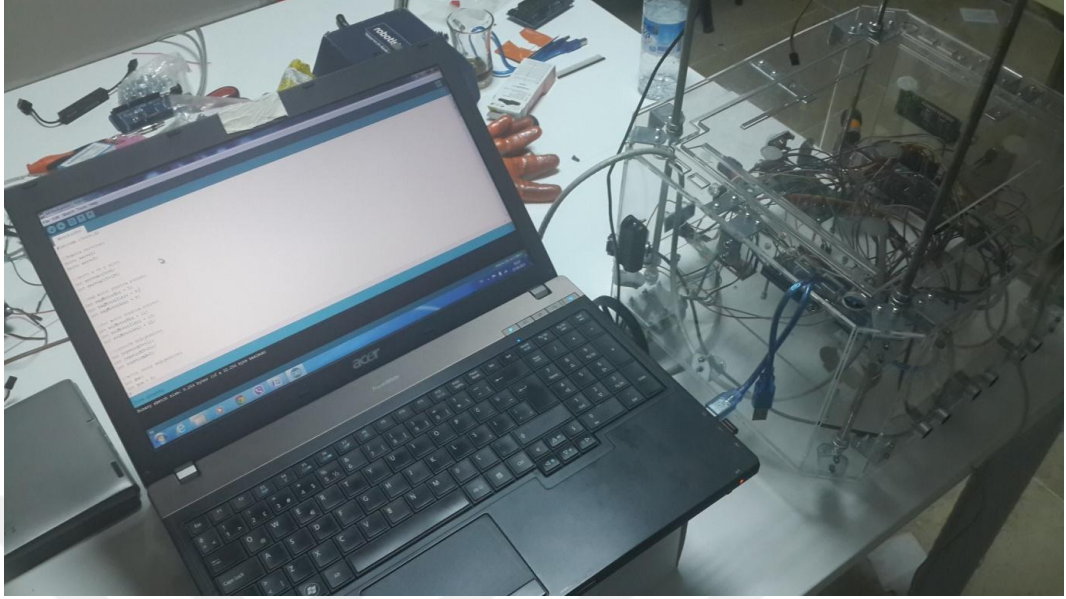
Bilgisayar montajı ve cırt cırt kayış benzeri bir bağlantı şekliyle sabitlenmesi yapılarak HEduBot montajı tamamlanır. Sonrasında isteğe bağlı olarak kablolar kelepçe yardımı ile sabitlenir ve HEduBot montajı tamamlanır (Şekil 85).



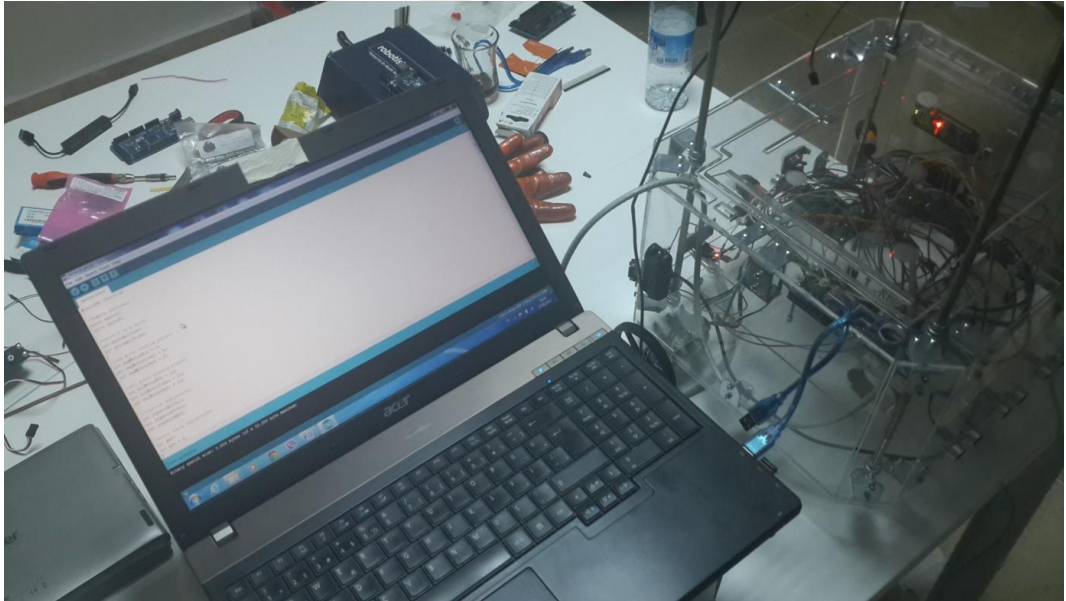
Şekil 85: Tamamlanan HEduBot

8.3.2. Çalıştırma

Montaj işlemi tamamlandıktan sonra ekte kaynak kodları bulunan Arduino UNO için yazılmış olan "MotorKontrol.ino" (Şekil 86) uygulaması ve Arduino Mega için hazırlanmış olan "SensorVeri.ino" (Şekil 87) uygulaması kartlara Arduino IDE aracılığı ile derlenerek yüklenir. Derlemede ek DVD içerisinde bulunan "arduinolib" klasörü Arduino IDE içerisinde kütüphane olarak tanıtılmalıdır.



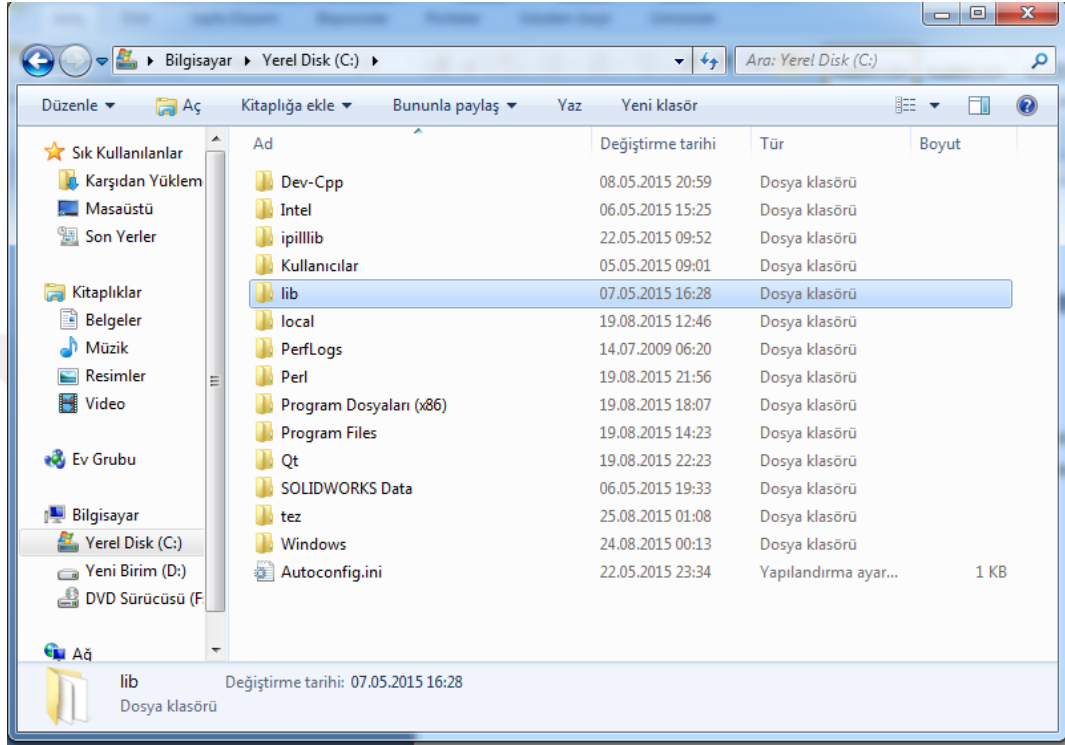
Şekil 86: MotorKontrol.ino yüklemesi



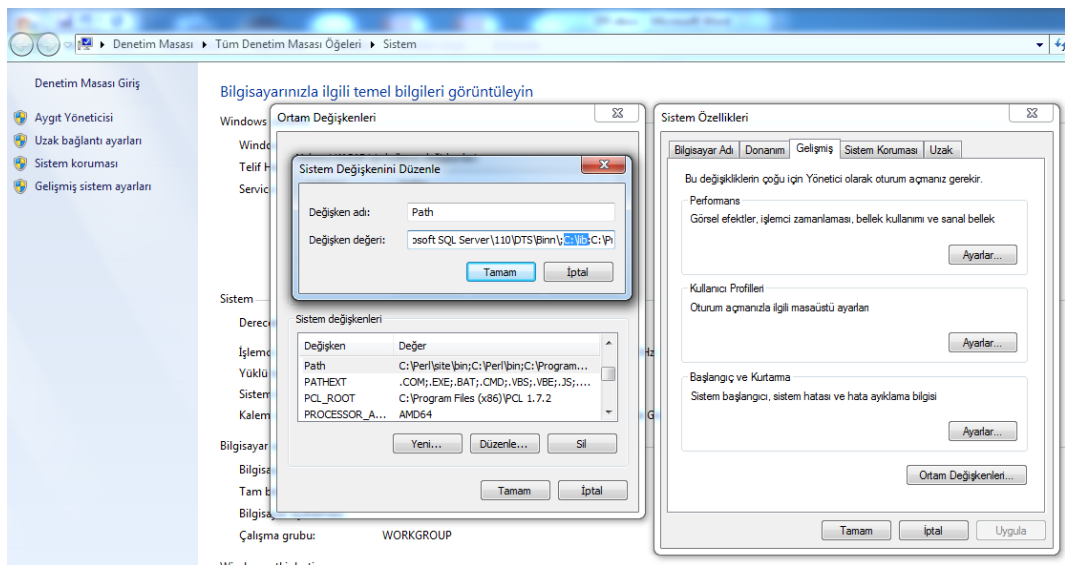
Şekil 87: SensorVeri.ino yüklemesi

Sınıf ortamında HEduBot, ekteki bağlantı şemasına uygun şekilde güç bağlantıları yapılarak yerel kullanıcı tarafından hazır hale getirilir. Pilot ve robot bilgisayarlarında Java SE Runtime Environment (JRE) kurulmuş olmalıdır. Robot bilgisayarı üzerinde Windows 32bit veya 64bit sistem platformuna bağlı olarak ek

CD içerisinde bulunan "lib" klasörü "C:\\" dizinine kopyalanır ve "PATH" ortam değişkeni altına bu dizinin yolu "C:\lib" şeklinde eklenir (Şekil 88 ve 89).

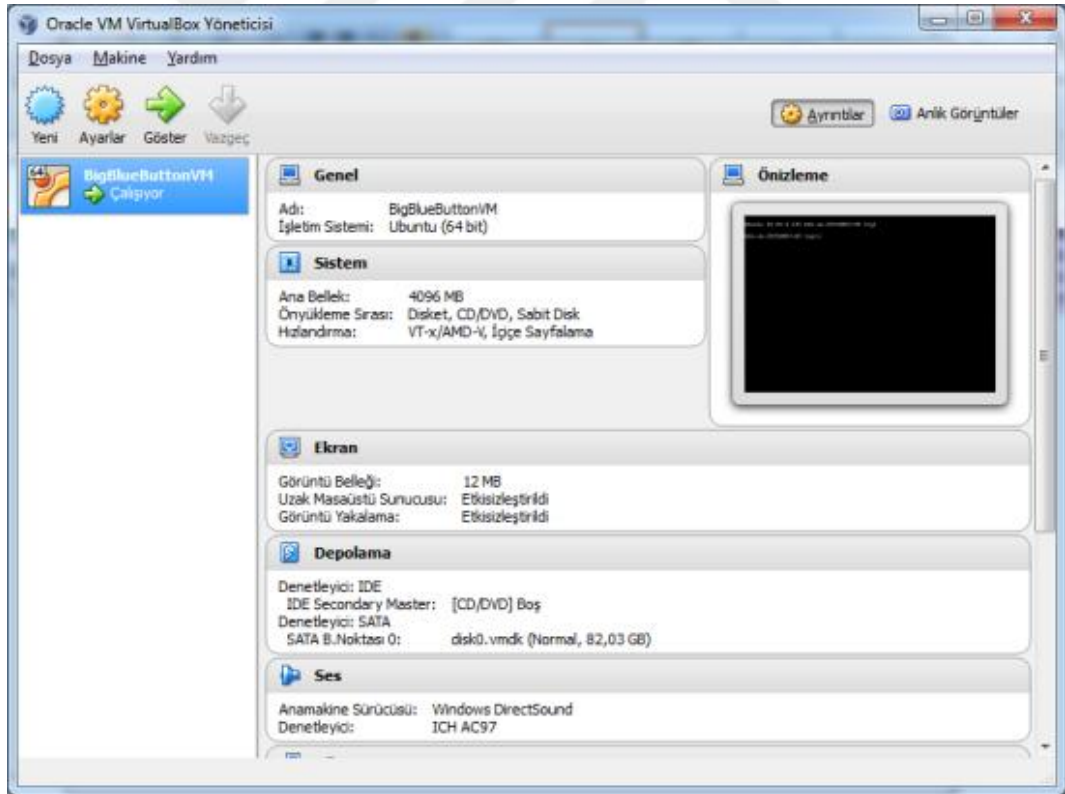


Şekil 88: Robot bilgisayarında "lib" klasörünün yerleşimi

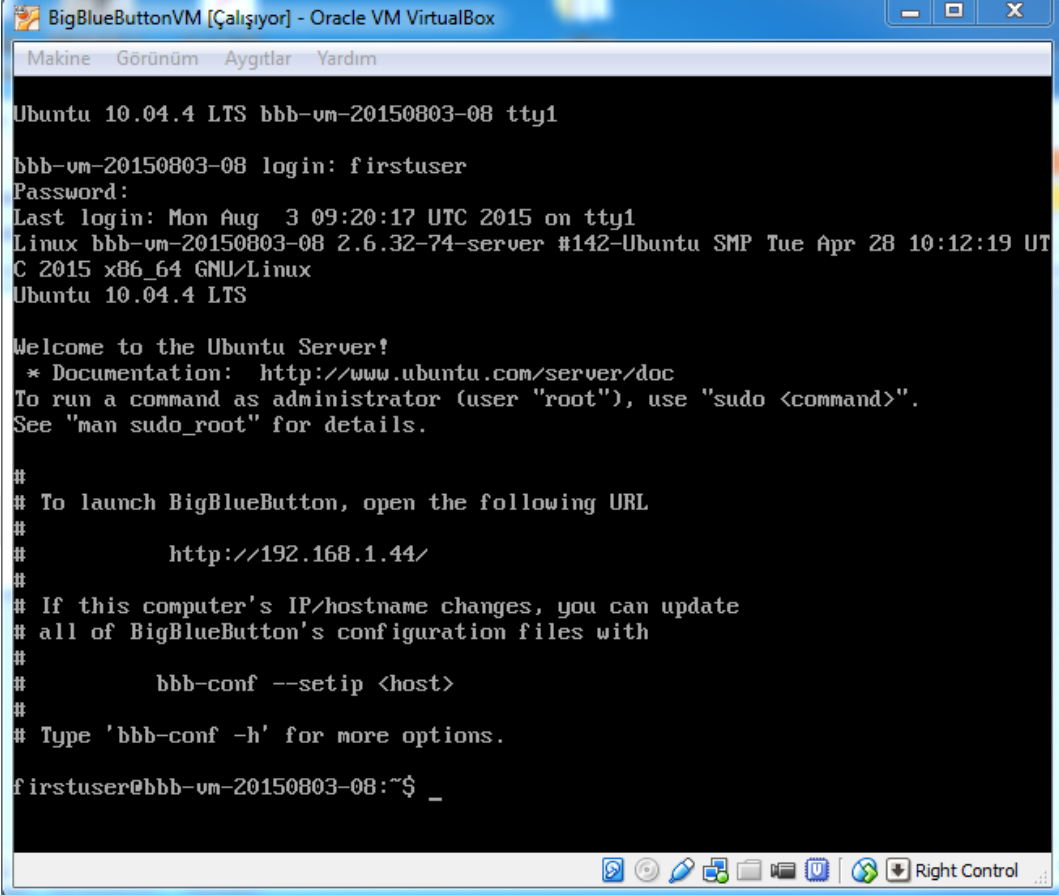


Şekil 89: Robot bilgisayarında PATH ayarının yapılması

Arduino UNO ve Arduino Mega bağlantı portları Arduino IDE üzerinden öğrenilerek, komut istemi veya terminalden "java -jar C:\lib\hedubot.jar [UNO_PORT] [MEGA_PORT] [SERVER_PORT]" komutunda [UNO_PORT] alanına Arduino UNO portu, [MEGA_PORT] alanına Arduino Mega portu ve [SERVER_PORT] alanına sunucu dinleme portu yazılarak giriş yapılır. Açılışta log olarak gösterilen IP ve sunucu portu pilota iletilmelidir. Bunun için BigBlueButton üzerinden pilota mesaj yollanabilir. Proje için kullanılacak BigBlueButton sunucusu sanal makine üzerinde kurulu vaziyette [119]'dan indirilebilir. Oracle firmasına ait ücretsiz ve açık kaynaklı sanal bilgisayar uygulaması olan VirtualBox, [120]'den ücretsiz temin edilebilir aynı zamanda üzerinde bu sanal makine kullanılabilir. Bu sayede kullanıcıların üst düzey teknik bilgiye sahip olmaları gerekmemektedir (Şekil 90 ve 91).



Şekil 90: VirtualBox uygulamasında BigBlueButton sunucunun çalıştırılması (a)



```

BigBlueButtonVM [Çalışıyor] - Oracle VM VirtualBox
Makine Görünüm Aygıtlar Yardım

Ubuntu 10.04.4 LTS bbb-vm-20150803-08 tty1
bbb-vm-20150803-08 login: firstuser
Password:
Last login: Mon Aug  3 09:20:17 UTC 2015 on tty1
Linux bbb-vm-20150803-08 2.6.32-74-server #142-Ubuntu SMP Tue Apr 28 10:12:19 UT
C 2015 x86_64 GNU/Linux
Ubuntu 10.04.4 LTS

Welcome to the Ubuntu Server!
 * Documentation: http://www.ubuntu.com/server/doc
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

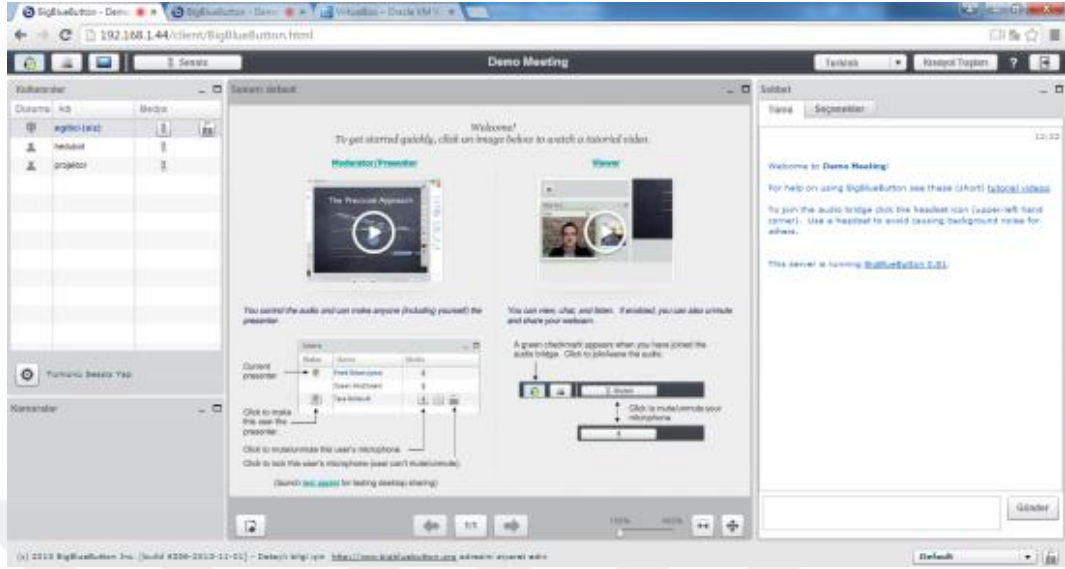
#
# To launch BigBlueButton, open the following URL
#
#     http://192.168.1.44/
#
# If this computer's IP/hostname changes, you can update
# all of BigBlueButton's configuration files with
#
#     bbb-conf --setip <host>
#
# Type 'bbb-conf -h' for more options.

firstuser@bbb-vm-20150803-08:~$ _

```

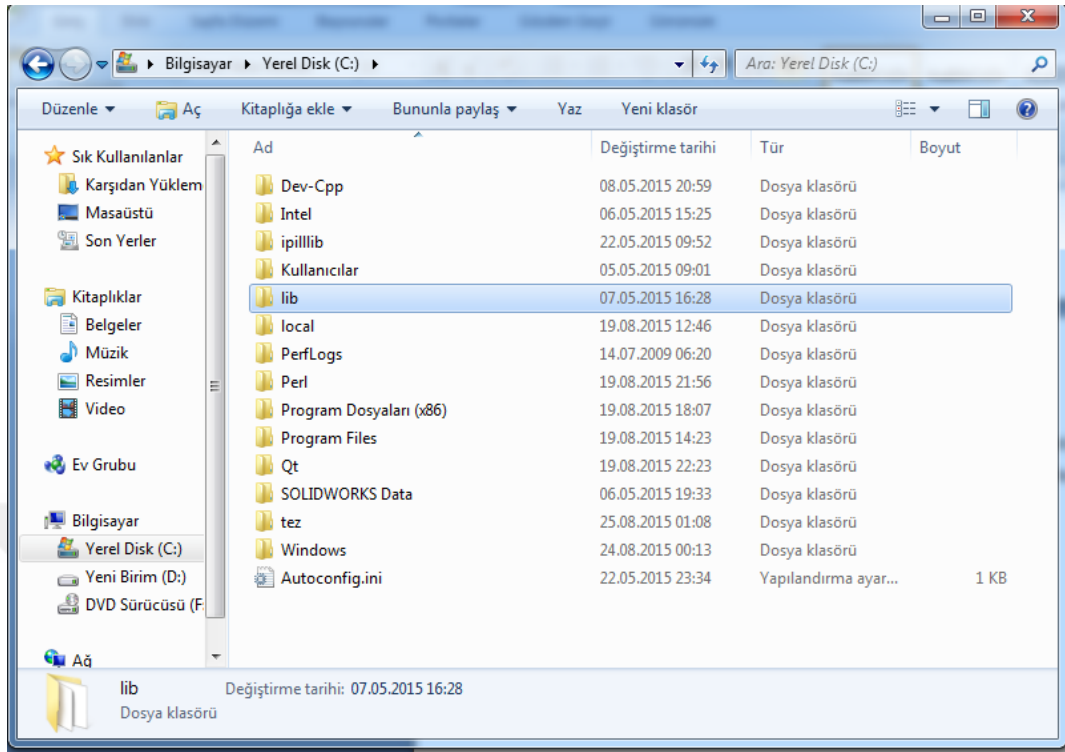
Şekil 91: VirtualBox uygulamasında BigBlueButton sunucunun çalıştırılması (b)

Sanal makine üzerinde BigBlueButton sunucusu başlatıldıktan sonra BigBlueButton sunucusunun bağlantı adresine eğitici, robot ve sınıf bilgisayarlarının tarayıcısı üzerinden bağlantı yapılır (Şekil 92).

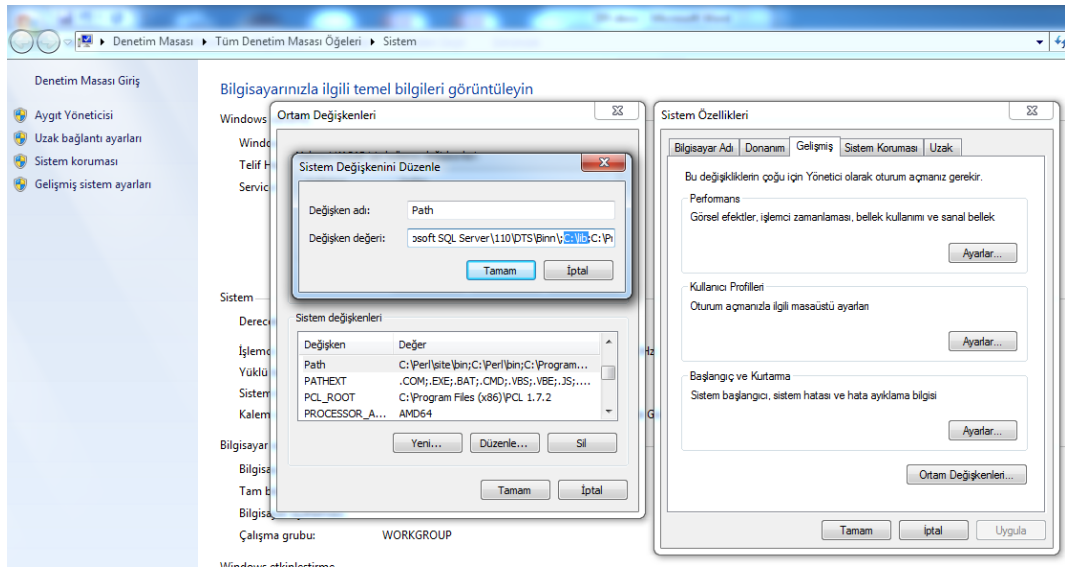


Şekil 92: BigBlueButton ekranı

Sınıf bilgisayarına bağlı projektör başlatılır. Pilot bilgisayarı üzerinde Windows 32bit veya 64bit sistem platformuna bağlı olarak ek CD içerisinde bulunan "lib" klasörü "C:" dizinine kopyalanır ve "PATH" ortam değişkeni altına bu dizinin yolu "C:\lib" şeklinde eklenir (Şekil 93 ve 94).



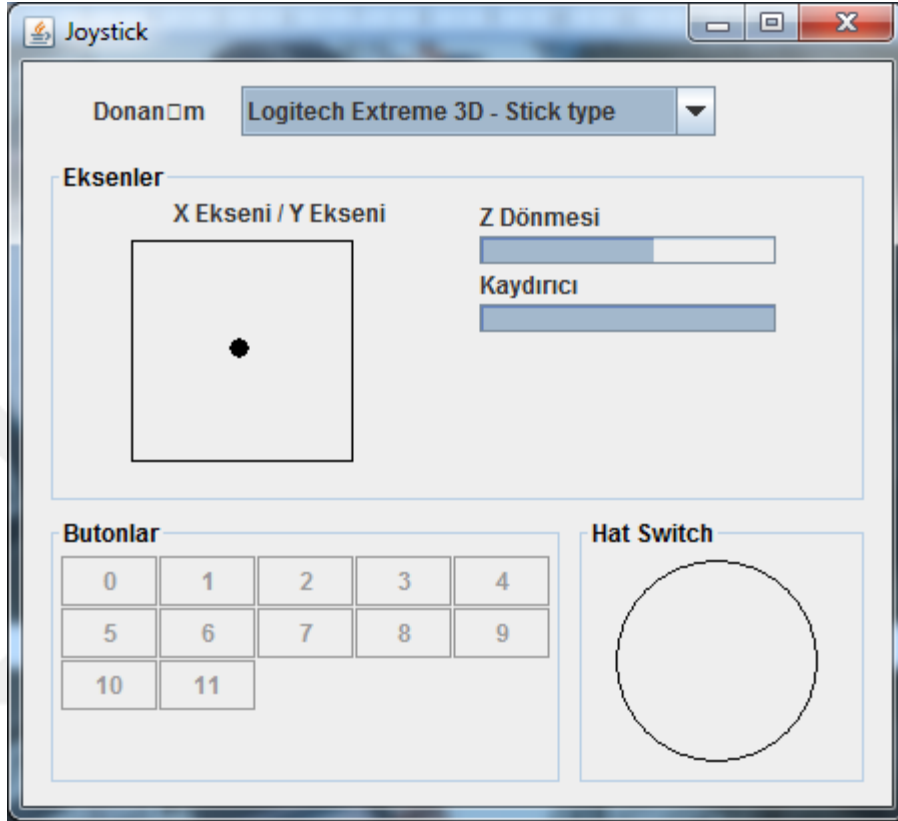
Şekil 93: Kullanıcı bilgisayarında "lib" klasörünün yerleşimi



Şekil 94: Kullanıcı bilgisayarında PATH ayarlarının yapılması

Daha sonra komut istemi veya terminalden, "java -jar C:\lib\hedubotpilot.jar [IP] [PORT]" komutunda [IP] alanına HEduBot bağlantı IP adresi [PORT] alanına bağlantı portu yazılarak giriş yapılır. Pilot joystick kontrol penceresini görecektir. Bu

pencerde kumandayı seçip robotu kontrol etmeye başlayabilir. Konsol ekranından sensör bilgileri takip edilebilir.



Şekil 95: Kullanıcı bilgisayarında joystick ekranı

Pilot robota bağlantı yaptıktan sonra kontrol pilota geçmektedir. Pilot öğrencilere sunuculuk hakkı verebilir, öğrencilere robotu götürerek dilediği öğrenciden tablet ile veri girişi ve sunum alabilir. Aynı zamanda sesli, yazılı ve görüntülü olarak öğrenciler ile iletişime geçebilir, masaüstü paylaşımı ve sunum yapabilir. Kamerayı yönlendirebilir. Tüm bu haberleşme öğrenciler tarafından projektör üzerinden ve robot bilgisayarı üzerinden görüntülenebilir. Acil durumlarda yerel kullanıcı HEduBot üzerindeki acil anahtarını kapatarak robot hareketini durdurabilir. Aynı zamanda olası hatalar için robot üzerinde bir adet reset anahtarı da bulunmaktadır.

8.4. UYGULAMA KODLARI

Uygulama kodları ekte bulunan DVD içerisinde "kaynak kodlar" klasöründedir, projede kullanılan diğer kütüphaneler ile birlikte verilmiştir. Kütüphaneler platform tipi ile isimlendirilmiş klasörün altında "lib" olarak adlandırılmıştır. Örneğin kurulacak sistemde 32 bit Microsoft Windows 7 işletim sistemi var ise "win32" klasörü altında bulunan "lib" klasörü kullanılmalıdır. Arduino uygulamaları için kullanılan kütüphaneler "arduinolib" klasörü içerisinde.

8.4.1. Kullanıcı Uygulaması

Kullanıcı bilgisayarına yüklenecek uygulamayı oluşturan sınıflar burada verilmektedir. Her bir başlık bir sınıfı temsil etmektedir. Bu sınıflar oluşturulup, ekte bulunan DVD içerisindeki kullanılacak olan platforma uygun "lib" klasörü proje kütüphanesi olarak tanıtılarak kullanıcı uygulaması derlenebilmektedir.

8.4.1.1. Main.java

```
package org.hedubot.base;

import org.hedubot.client.Client;
import org.hedubot.controller.Joystick;
import org.hedubot.controller.JoystickData;

public class Main {
    public static int JOYSTICK_X=0;
    public static int JOYSTICK_Y=0;
    public static int JOYSTICK_Z=0;
    public static int JOYSTICK_S=0;
    public static int JOYSTICK_HAT=0;
    public static String MESSAGE="jy://511,511,0";
    public static String RESPONSE="";

    public synchronized static int getJOYSTICK_X() {
        return JOYSTICK_X;
    }

    public synchronized static void setJOYSTICK_X(int jOYSTICK_X) {
        JOYSTICK_X = jOYSTICK_X;
    }
}
```

```

    }

    public synchronized static int getJOYSTICK_Y() {
        return JOYSTICK_Y;
    }

    public synchronized static void setJOYSTICK_Y(int jOYSTICK_Y) {
        JOYSTICK_Y = jOYSTICK_Y;
    }

    public synchronized static int getJOYSTICK_Z() {
        return JOYSTICK_Z;
    }

    public synchronized static void setJOYSTICK_Z(int jOYSTICK_Z) {
        JOYSTICK_Z = jOYSTICK_Z;
    }

    public synchronized static int getJOYSTICK_S() {
        return JOYSTICK_S;
    }

    public synchronized static void setJOYSTICK_S(int jOYSTICK_S) {
        JOYSTICK_S = jOYSTICK_S;
    }

    public synchronized static int getJOYSTICK_HAT() {
        return JOYSTICK_HAT;
    }

    public synchronized static void setJOYSTICK_HAT(int jOYSTICK_HAT) {
        JOYSTICK_HAT = jOYSTICK_HAT;
    }

    public synchronized static String getMESSAGE() {
        return MESSAGE;
    }

    public synchronized static void setMESSAGE(String mMESSAGE) {
        MESSAGE = mMESSAGE;
    }

    public synchronized static String getRESPONSE() {
        return RESPONSE;
    }

    public synchronized static void setRESPONSE(String rRESPONSE) {
        RESPONSE = rRESPONSE;
    }

    public static void main(String[] args) {
        System.out.print("Usage java -jar hedubotpilot.jar [IP] [PORT] please type
HEduBot IP and PORT than press Enter");
        Client c = null;
        try{
            c=new Client(args[0],Integer.parseInt(args[1]));
        }catch(Exception e){

```



```

        throw e;
    }
    Joystick j=new Joystick();
    j.execute();
    JoystickData jd=new JoystickData();
    jd.execute();
    c.execute();

    while(true){
    }
}
}

```

8.4.1.2. Client.java

```

package org.hedubot.client;

import java.io.IOException;
import java.net.ConnectException;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;

import javax.swing.SwingWorker;

import org.hedubot.base.Main;
import org.hedubot.io.Reader;
import org.hedubot.io.Writer;

public class Client extends SwingWorker<String, String> {
    private String host;
    private int port;
    private Socket ioSocket;
    public Client(String host, int port) {
        this.host = host;
        this.port = port;
    }
    @Override
    protected String doInBackground() {
        try {
            ioSocket = new Socket();
            test:
            while(!ioSocket.isConnected()){
                try {
                    System.out.println("Connecting... to "+host+":"+port);

                    ioSocket.connect(new InetSocketAddress(host, port), 0);
                    ioSocket.setSoTimeout(0);
                } catch (ConnectException e) {
                    continue test;
                } catch (SocketException s) {

```

```

        continue test;
    }
}
while (ioSocket.isConnected()) {
    Writer mw = new Writer();
    try {
mw.writeOutputStream(ioSocket.getOutputStream(),
                    Main.getMESSAGE());
    } catch (IOException e) {
        e.printStackTrace();
    }
    Reader mr = new Reader();
    try {
        Main.setRESPONSE(mr.readInputStream(ioSocket
            .getInputStream()));
        if (Main.getRESPONSE() != null
            || !Main.getRESPONSE().equals("")) {
            System.out.println("Response from server "
                + Main.getRESPONSE());
        } else {
            System.out.println("Response is null or empty");
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
} catch (UnknownHostException e) {
    e.printStackTrace();
    return e.getClass() + " " + e.getMessage();
} catch (IOException e) {
    System.out.println("IOException e ");
    e.printStackTrace();
    return e.getClass() + " " + e.getMessage();
}
return host;
}
protected void done() {
    try {
        if (ioSocket != null) {
            ioSocket.close();
            ioSocket = null;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

8.4.1.3. Joystick.java [121]

```

package org.hedubot.controller;

import java.awt.Dimension;
import java.awt.FlowLayout;
import java.util.ArrayList;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JToggleButton;
import javax.swing.SwingWorker;
import net.java.games.input.Component;
import net.java.games.input.Component.Identifier;
import net.java.games.input.Controller;
import net.java.games.input.ControllerEnvironment;
import org.hedubot.base.Main;
import org.hedubot.windows.JoystickWindow;

public class Joystick extends SwingWorker<String, String>{

    final JoystickWindow jw;
    private ArrayList<Controller> foundControllers;

    public Joystick() {
        jw = new JoystickWindow();
        jw.setVisible(true);
        foundControllers = new ArrayList<>();
        searchForControllers();
    }

    private void searchForControllers() {
        Controller[] controllers = ControllerEnvironment.getDefaultEnvironment().getControllers();

        for(int i = 0; i < controllers.length; i++){
            Controller controller = controllers[i];

            if (
                controller.getType() == Controller.Type.STICK ||
                controller.getType() == Controller.Type.GAMEPAD ||
                controller.getType() == Controller.Type.WHEEL ||
                controller.getType() == Controller.Type.FINGERSTICK
            )
            {
                foundControllers.add(controller);
                jw.addControllerName(controller.getName() + " - " + controller.getType().toString() + "
type");
            }
        }
    }

    public int getAxisValueInPercentage(float axisValue)

```

```

{
    return (int)(((2 - (1 - axisValue)) * 100) / 2);
}
public int getAxisValueInAnalog(float axisValue)
{
    return (int)(((2 - (1 - axisValue)) * 1024) / 2);
}
protected String doInBackground() throws Exception {
    if(foundControllers.isEmpty()){
        jw.addControllerName("No controller found!");
        return "No controller found!";
    }

    while(true)
    {
        int selectedControllerIndex = jw.getSelectedControllerName();
        Controller controller = foundControllers.get(selectedControllerIndex);
        if( !controller.poll() ){
            jw.showControllerDisconnected();
            break;
        }
        int xAxisPercentage = 0;
        int yAxisPercentage = 0;
        JPanel axesPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 25, 2));
        axesPanel.setBounds(0, 0, 200, 190);
        JPanel buttonsPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 1, 1));
        buttonsPanel.setBounds(6, 19, 246, 110);
        Component[] components = controller.getComponents();
        for(int i=0; i < components.length; i++)
        {
            Component component = components[i];
            Identifier componentIdentifier = component.getIdentifier();

            if(componentIdentifier.getName().matches("[0-9]*$")){
                boolean isItPressed = true;
                if(component.getPollData() == 0.0f)
                    isItPressed = false;
                String buttonIndex;
                buttonIndex = component.getIdentifier().toString();
                JToggleButton aToggleButton = new JToggleButton(buttonIndex, isItPressed);
                aToggleButton.setPreferredSize(new Dimension(48, 25));
                aToggleButton.setEnabled(false);
                buttonsPanel.add(aToggleButton);
                continue;
            }

            if(componentIdentifier == Component.Identifier.Axis.POV){
                float hatSwitchPosition = component.getPollData();
                jw.setHatSwitch(hatSwitchPosition);
                continue;
            }

            if(component.isAnalog()){
                float axisValue = component.getPollData();
                int axisValueInPercentage = getAxisValueInPercentage(axisValue);
                int axisValueInAnalog=getAxisValueInAnalog(axisValue);
            }
        }
    }
}

```

```

        if(componentIdentifier == Component.Identifier.Axis.X){
            Main.JOYSTICK_X=axisValueInAnalog;
            xAxisPercentage = axisValueInPercentage;
            continue;
        }
        if(componentIdentifier == Component.Identifier.Axis.Y){
            Main.JOYSTICK_Y=axisValueInAnalog;
            yAxisPercentage = axisValueInPercentage;
            continue;
        }
        if(componentIdentifier == Component.Identifier.Axis.RZ){
            Main.JOYSTICK_Z=axisValueInAnalog;
        }
        if(componentIdentifier == Component.Identifier.Axis.SLIDER){
            Main.JOYSTICK_S=axisValueInAnalog;
        }
        JLabel progressBarLabel = new JLabel(component.getName().replace("ý", "1"));
        JProgressBar progressBar = new JProgressBar(0, 100);
        progressBar.setValue(axisValueInPercentage);
        axesPanel.add(progressBarLabel);
        axesPanel.add(progressBar);
    }
}

jw.setControllerButtons(buttonsPanel);
jw.setXYAxis(xAxisPercentage, yAxisPercentage);
jw.addAxisPanel(axesPanel);

try {
    Thread.sleep(25);
} catch (InterruptedException ex) {
    ex.printStackTrace();
}
}

return null;
}
}

```

8.4.1.4. JoystickData.java

```

package org.hedubot.controller;

import java.util.concurrent.TimeUnit;
import org.hedubot.base.Main;
import org.hedubot.client.Client;
import org.hedubot.utils.Constants;
import javax.swing.SwingWorker;

public class JoystickData extends SwingWorker<String, String> {

    public JoystickData(){

```

```

    }
    @Override
    protected String doInBackground() throws Exception {
        while (true) {
            if (Constants.SEND_JOYSTICK) {
                Main.setMESSAGE("jy://" + Main.JOYSTICK_X + "," + Main.JOYSTICK_Y + "," + Main.JOYSTICK_HAT);
            } else {
                break;
            }
        }
        return "Joystick control stopped";
    }

    protected void done() {
        return;
    }
}

```

8.4.1.5. Reader.java

```

package org.hedubot.io;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import org.hedubot.utils.Constants;

public class Reader {
    public String readInputStream(InputStream is) throws IOException{
        InputStreamReader isr=new InputStreamReader(is, "UTF-8");

        BufferedReader br = new BufferedReader(isr);
        StringBuffer sb = new StringBuffer();
        boolean eom = false;
        int c = 0;
        c = br.read();
        if (c == -1) {
            return null;
        }
        if (c != Constants.START_MESSAGE_CHARACTER) {
            return null;
        }
        while (!eom) {
            c = br.read();
            if (c == -1) {
                return null;
            }
            if (c == Constants.END_MESSAGE_CHARACTER) {

```

```

        c = br.read();
        if (c >= 0)
            if (c != Constants.LAST_MESSAGE_CHARACTER) {
                return null;
            }
            eom = true;
        } else {
            sb.append((char) c);
        }
    }
    return sb.toString();
}
}

```

8.4.1.6. Writer.java

```

package org.hedubot.io;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import org.hedubot.utils.Constants;

public class Writer {
    public void writeOutputStream(OutputStream os,String message) throws IOException{
        OutputStreamWriter osw= new OutputStreamWriter(os, "UTF-8");

        BufferedWriter bw=new BufferedWriter(osw);
        bw.write(Constants.START_MESSAGE_CHARACTER);
        bw.write(message);
        bw.write(Constants.END_MESSAGE_CHARACTER);
        bw.write(Constants.LAST_MESSAGE_CHARACTER);
        bw.flush();
    }
}

```

8.4.1.7. Constants.java

```

package org.hedubot.utils;

public class Constants {
    public static int PORT=4000;
    public static boolean SEND_JOYSTICK=true;
    public static boolean SERVER_IS_RUN=false;
    public static String STOP_MESSAGE="jy://511,511,0";
    public static final char START_MESSAGE_CHARACTER = '\u000b';
    public static final char END_MESSAGE_CHARACTER = '\u001c';
    public static final char LAST_MESSAGE_CHARACTER = '\r';
}

```

```

    public static final int SERVER_TIMEOUT=200;
    public static final int CLIENT_TIMEOUT=200;
    public static final int ARDUINO_SERVER_TIME_OUT = 200;
    public static final int ARDUINO_DATA_RATE = 9600;
    public static boolean ARDUINO_SERVER_IS_RUN=true;
}

```

8.4.1.8. JoystickWindow.java [121]

```

package org.hedubot.windows;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics2D;
import java.awt.Point;
import javax.swing.BorderFactory;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.LayoutStyle;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.UIManager;
import javax.swing.WindowConstants;
import javax.swing.border.TitledBorder;

import net.java.games.input.Component;

import org.hedubot.base.Main;
//VS4E -- DO NOT REMOVE THIS LINE!
public class JoystickWindow extends JFrame {
    private static final long serialVersionUID = 1L;
    private JComboBox<String> jComboBox_controllers;
    private JLabel jLabelXYAxis;
    private JPanel jPanelAxes;
    private JPanel jPanelButtons;
    private JPanel jPanelHatSwitch;
    private JPanel jPanelXYAxis;
    private JPanel jPanel_forAxis;
    private JLabel lblDonanim;

    public JoystickWindow() {
        initComponents();
    }

    private void initComponents() {
        jPanelAxes = new JPanel();
        jLabelXYAxis = new JLabel();
        jPanelXYAxis = new JPanel();
        jPanel_forAxis = new JPanel();
        jPanelButtons = new JPanel();

```



```

jPanelHatSwitch = new JPanel();
jComboBox_controllers = new JComboBox<String>();
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
setTitle("Joystick");
Point p = new Point(400, 0);
setLocation(p);
jPanelAxes.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Eksenler", TitledBorder.LEADING,
TitledBorder.TOP, null, Color.BLACK));
jLabelXYAxis.setText("X Ekseni / Y Ekseni");
jPanelXYAxis.setBorder(BorderFactory
.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanelXYAxis.setPreferredSize(new java.awt.Dimension(111, 111));
GroupLayout jPanelXYAxisLayout = new GroupLayout(jPanelXYAxis);
jPanelXYAxis.setLayout(jPanelXYAxisLayout);
jPanelXYAxisLayout.setHorizontalGroup(jPanelXYAxisLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
109, Short.MAX_VALUE));
jPanelXYAxisLayout.setVerticalGroup(jPanelXYAxisLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
109, Short.MAX_VALUE));
GroupLayout jPanel_forAxisLayout = new GroupLayout(jPanel_forAxis);
jPanel_forAxis.setLayout(jPanel_forAxisLayout);
jPanel_forAxisLayout.setHorizontalGroup(jPanel_forAxisLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
202, Short.MAX_VALUE));
jPanel_forAxisLayout.setVerticalGroup(jPanel_forAxisLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
0, Short.MAX_VALUE));
GroupLayout jPanelAxesLayout = new GroupLayout(jPanelAxes);
jPanelAxes.setLayout(jPanelAxesLayout);
jPanelAxesLayout
.setHorizontalGroup(jPanelAxesLayout
.createParallelGroup(GroupLayout.Alignment.LEADING)
.addGroup(
jPanelAxesLayout
.createSequentialGroup()
.addGroup(
jPanelAxesLayout
.createParallelGroup(
GroupLayout.Alignment.LEADING)
.addGroup(
jPanelAxesLayout
.createSequentialGroup()

```

```

        .addGap(58,
                58,
                58)
        .addComponent(
                JLabelXYAxis))
        .addGroup(
                JPanelAxesLayout
                .createSequentialGroup()
                .addGap(37,
                        37,
                        37)
                .addComponent(
                        JPanelXYAxis,
                        GroupLayout.PREFERRED_SIZE,
                        GroupLayout.DEFAULT_SIZE,
                        GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(
                LayoutStyle.ComponentPlacement.RELATED,
                GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
        .addComponent(jPanel_forAxis,
                GroupLayout.PREFERRED_SIZE,
                GroupLayout.DEFAULT_SIZE,
                GroupLayout.PREFERRED_SIZE)
        .addContainerGap());
        JPanelAxesLayout.setVerticalGroup(jPanelAxesLayout
                .createParallelGroup(GroupLayout.Alignment.LEADING)
                .addGroup(
                        JPanelAxesLayout

```

```

                                                                                   .createSequentialGroup()
.addComponent(jLabelXYAxis)
                                                                                   .addPreferredGap(
LayoutStyle.ComponentPlacement.RELATED)
                                                                                   .addComponent(jPanelXYAxis,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
                                                                                   .addGap(0,
Short.MAX_VALUE))
                                                                                   .addComponent(jPanel_forAxis, GroupLayout.DEFAULT_SIZE,
                                                                                   GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE));
    jPanelButtons.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Butonlar", TitledBorder.LEFT,
TitledBorder.TOP, null, Color.BLACK));
    GroupLayout jPanelButtonsLayout = new GroupLayout(jPanelButtons);
    jPanelButtons.setLayout(jPanelButtonsLayout);
    jPanelButtonsLayout.setHorizontalGroup(jPanelButtonsLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
248, Short.MAX_VALUE));
    jPanelButtonsLayout.setVerticalGroup(jPanelButtonsLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
112, Short.MAX_VALUE));
    jPanelHatSwitch.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Hat Switch", TitledBorder.LEFT,
TitledBorder.TOP, null, Color.BLACK));
    GroupLayout jPanelHatSwitchLayout = new GroupLayout(jPanelHatSwitch);
    jPanelHatSwitch.setLayout(jPanelHatSwitchLayout);
    jPanelHatSwitchLayout.setHorizontalGroup(jPanelHatSwitchLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
121, Short.MAX_VALUE));
    jPanelHatSwitchLayout.setVerticalGroup(jPanelHatSwitchLayout
.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0,
0, Short.MAX_VALUE));
    jComboBox_controllers
        .addActionListener(new java.awt.event.ActionListener() {
            @Override
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                jComboBox_controllersActionPerformed(evt);
            }
        });
    lblDonanim = new JLabel("Donanim");
    GroupLayout layout = new GroupLayout(getContentPane());
    layout.setHorizontalGroup(
        layout.createParallelGroup(Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
            .addContainerGap()

        .addGroup(layout.createParallelGroup(Alignment.TRAILING)
            .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addComponent(jPanelButtons,
            GroupLayout.PREFERRED_SIZE,
            GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(ComponentPlacement.RELATED)

        .addComponent(jPanelHatSwitch,
            GroupLayout.DEFAULT_SIZE,
            GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addComponent(jPanelAxes,
            GroupLayout.DEFAULT_SIZE, 399, Short.MAX_VALUE))
            .addContainerGap()
            .addGroup(layout.createSequentialGroup())

        .addPreferredGap(ComponentPlacement.RELATED, 20, Short.MAX_VALUE)
            .addComponent(lblDonanim)
            .addGap(18)

        .addComponent(jComboBox_controllers,
            GroupLayout.PREFERRED_SIZE, 237,
            GroupLayout.PREFERRED_SIZE)
            .addGap(88)))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(Alignment.LEADING)
            .addGroup(layout.createSequentialGroup())
            .addContainerGap(GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)

        .addGroup(layout.createParallelGroup(Alignment.BASELINE)
            .addComponent(jComboBox_controllers,
            GroupLayout.PREFERRED_SIZE,
            GroupLayout.PREFERRED_SIZE)

            .addComponent(lblDonanim))
            .addPreferredGap(ComponentPlacement.UNRELATED)
            .addComponent(jPanelAxes,
            GroupLayout.PREFERRED_SIZE,
            GroupLayout.DEFAULT_SIZE,
            GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(Alignment.LEADING, false)
            .addComponent(jPanelButtons,
            GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanelHatSwitch,
            GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap()
    );
    getContentPane().setLayout(layout);
    pack();
    Dimension d = new Dimension(450, 410);

```

```

        setSize(d);
    }
    private void jComboBox_controllersActionPerformed(
        java.awt.event.ActionEvent evt)    { // GEN-
FIRST:event_jComboBox_controllersActionPerformed
        jPanelButtons.removeAll();
        jPanelButtons.repaint();
        jPanel_forAxis.removeAll();
        jPanel_forAxis.repaint();
    }

    public int getSelectedControllerName() {
        return jComboBox_controllers.getSelectedIndex();
    }

    public void addControllerName(String controllerName) {
        jComboBox_controllers.addItem(controllerName);
    }

    public void showControllerDisconnected() {
        jComboBox_controllers.removeAllItems();
        jComboBox_controllers.addItem("Controller disconnected!");
    }

    public void setXYAxis(int xPercentage, int yPercentage) {
        Graphics2D g2d = (Graphics2D) jPanelXYAxis.getGraphics();
        g2d.clearRect(1, 1, jPanelXYAxis.getWidth() - 2,
            jPanelXYAxis.getHeight() - 2);
        g2d.fillOval(xPercentage, yPercentage, 10, 10);
    }

    public void setControllerButtons(JPanel buttonsPanel) {
        jPanelButtons.removeAll();
        jPanelButtons.add(buttonsPanel);
        jPanelButtons.validate();
    }

    public void setHatSwitch(float hatSwitchPosition) {
        int circleSize = 100;
        Graphics2D g2d = (Graphics2D) jPanelHatSwitch.getGraphics();
        g2d.clearRect(5, 15, jPanelHatSwitch.getWidth() - 10,
            jPanelHatSwitch.getHeight() - 22);
        g2d.drawOval(20, 22, circleSize, circleSize);

        int smallCircleSize = 10;
        int upCircleX = 65;
        int upCircleY = 17;
        int leftCircleX = 15;
        int leftCircleY = 68;
        int betweenX = 37;
        int betweenY = 17;
        int x = 0;
        int y = 0;
        g2d.setColor(Color.blue);
        if (Float.compare(hatSwitchPosition, Component.POV.OFF) == 0) {
            Main.JOYSTICK_HAT = 0;
            return;
        }
    }

```

```

    }
    if (Float.compare(hatSwitchPosition, Component.POV.UP) == 0) {
        x = upCircleX;
        y = upCircleY;
        Main.JOYSTICK_HAT = 1;
    } else if (Float.compare(hatSwitchPosition, Component.POV.DOWN) == 0) {
        x = upCircleX;
        y = upCircleY + circleSize;
        Main.JOYSTICK_HAT = 5;
    } else if (Float.compare(hatSwitchPosition, Component.POV.LEFT) == 0) {
        x = leftCircleX;
        y = leftCircleY;
        Main.JOYSTICK_HAT = 7;
    } else if (Float.compare(hatSwitchPosition, Component.POV.RIGHT) == 0) {
        x = leftCircleX + circleSize;
        y = leftCircleY;
        Main.JOYSTICK_HAT = 3;
    } else if (Float.compare(hatSwitchPosition, Component.POV.UP_LEFT) == 0) {
        x = upCircleX - betweenX;
        y = upCircleY + betweenY;
        Main.JOYSTICK_HAT = 8;
    } else if (Float.compare(hatSwitchPosition, Component.POV.UP_RIGHT) == 0) {
        x = upCircleX + betweenX;
        y = upCircleY + betweenY;
        Main.JOYSTICK_HAT = 2;
    } else if (Float.compare(hatSwitchPosition, Component.POV.DOWN_LEFT) == 0)
    {
        x = upCircleX - betweenX;
        y = upCircleY + circleSize - betweenY;
        Main.JOYSTICK_HAT = 6;
    } else if (Float.compare(hatSwitchPosition, Component.POV.DOWN_RIGHT) ==
0) {
        x = upCircleX + betweenX;
        y = upCircleY + circleSize - betweenY;
        Main.JOYSTICK_HAT = 4;
    }
    g2d.fillOval(x, y, smallCircleSize, smallCircleSize);
}

public void addAxisPanel(JPanel axesPanel) {
    jPanel_forAxis.removeAll();
    jPanel_forAxis.add(axesPanel);
    jPanel_forAxis.validate();
}

public JComboBox<String> getjComboBox_controllers() {
    return jComboBox_controllers;
}

public void setjComboBox_controllers(JComboBox<String> jComboBox_controllers) {
    this.jComboBox_controllers = jComboBox_controllers;
}

public JLabel getjLabelXYAxis() {
    return jLabelXYAxis;
}

```

```
public void setjLabelXYAxis(JLabel jLabelXYAxis) {
    this.jLabelXYAxis = jLabelXYAxis;
}

public JPanel getjPanelAxes() {
    return jPanelAxes;
}

public void setjPanelAxes(JPanel jPanelAxes) {
    this.jPanelAxes = jPanelAxes;
}

public JPanel getjPanelButtons() {
    return jPanelButtons;
}

public void setjPanelButtons(JPanel jPanelButtons) {
    this.jPanelButtons = jPanelButtons;
}

public JPanel getjPanelHatSwitch() {
    return jPanelHatSwitch;
}

public void setjPanelHatSwitch(JPanel jPanelHatSwitch) {
    this.jPanelHatSwitch = jPanelHatSwitch;
}

public JPanel getjPanelXYAxis() {
    return jPanelXYAxis;
}

public void setjPanelXYAxis(JPanel jPanelXYAxis) {
    this.jPanelXYAxis = jPanelXYAxis;
}

public JPanel getjPanel_forAxis() {
    return jPanel_forAxis;
}

public void setjPanel_forAxis(JPanel jPanel_forAxis) {
    this.jPanel_forAxis = jPanel_forAxis;
}

public static long getSerialVersionUID() {
    return serialVersionUID;
}

public JLabel getLblDonanim() {
    return lblDonanim;
}

public void setLblDonanim(JLabel lblDonanim) {
    this.lblDonanim = lblDonanim;
}
}
```

8.4.2. Robot Bilgisayar Uygulaması

Robot bilgisayarına yüklenecek uygulamayı oluşturan sınıflar burada verilmektedir. Her bir başlık bir sınıfı temsil etmektedir. Bu sınıflar oluşturulup, ekte bulunan DVD içerisindeki kullanılacak olan platforma uygun "lib" klasörü proje kütüphanesi olarak tanıtılarak kullanıcı uygulaması derlenebilmektedir.

8.4.2.1. Main.java

```
package hedubot.base;

import hedubot.io.ArduinoMegaHelper;
import hedubot.io.ArduinoUnoHelper;
import hedubot.server.Server;
import hedubot.utils.Constants;
import java.io.IOException;
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.util.Enumeration;

public class Main {
    public static ArduinoUnoHelper unoah = null;
    public static ArduinoMegaHelper megaah = null;
    public static String megamesage = "";
    public static String unomesage = "";
    public static String UNO_PORT = "";
    public static String MEGA_PORT = "";
    public static int SERVER_PORT;

    public static String getUNO_PORT() {
        return UNO_PORT;
    }

    public static void setUNO_PORT(String uNO_PORT) {
        UNO_PORT = uNO_PORT;
    }

    public static String getMEGA_PORT() {
        return MEGA_PORT;
    }

    public static void setMEGA_PORT(String mEGA_PORT) {
        MEGA_PORT = mEGA_PORT;
    }

    public static int getSERVER_PORT() {
        return SERVER_PORT;
    }
}
```



```

public static void setSERVER_PORT(int sSERVER_PORT) {
    SERVER_PORT = sSERVER_PORT;
}

public static ArduinoUnoHelper getUnoah() {
    return unoah;
}

public static void setUnoah(ArduinoUnoHelper unoah) {
    Main.unoah = unoah;
}

public static ArduinoMegaHelper getMegaah() {
    return megaah;
}

public static void setMegaah(ArduinoMegaHelper megaah) {
    Main.megaah = megaah;
}

public static String getMegamesage() {
    return megamesage;
}

public static void setMegamesage(String megamesage) {
    Main.megamesage = megamesage;
}

public static String getUnomesage() {
    return unomesage;
}

public static void setUnomesage(String unomesage) {
    Main.unomesage = unomesage;
}

public static void main(String[] args) throws IOException {
    System.out
        .print("Usage java -jar hedubot.jar [UNO_PORT] [MEGA_PORT]
[SERVER_PORT] please type HEduBot parameters than press Enter");
    try {
        setUNO_PORT(args[0]);
        setMEGA_PORT(args[1]);
        setSERVER_PORT(Integer.parseInt(args[2]));
    } catch (Exception e) {
        throw e;
    }
    Enumeration<NetworkInterface> n = NetworkInterface
        .getNetworkInterfaces();
    for (; n.hasMoreElements();) {
        NetworkInterface e = n.nextElement();
        System.out.println("NIC display name is " + e.getDisplayName());
        Enumeration<InetAddress> a = e.getInetAddresses();
        for (; a.hasMoreElements();) {
            InetAddress addr = a.nextElement();
            System.out.println("IP " + addr.getHostAddress());

```

```

    }
}

if (Main.getMegaah() == null) {
    Main.setMegaah(new ArduinoMegaHelper());
    ;
    Main.getMegaah().initialize(Main.getMEGA_PORT());
}

Server s = new Server(getSERVER_PORT());
Constants.SERVER_IS_RUN = true;
s.execute();
while (true) {
}
}
}

```

8.4.2.2. ArduinoMegaHelper.java

```

package hedubot.io;

import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import hedubot.base.Main;
import hedubot.utils.Constants;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.util.Enumeration;

public class ArduinoMegaHelper implements SerialPortEventListener {
    SerialPort serialPort;

    private BufferedReader input;
    private OutputStream output;

    private static final int DATA_RATE = 9600;

    public void initialize(String port) {

        CommPortIdentifier portId = null;
        Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();

        while (portEnum.hasMoreElements()) {
            CommPortIdentifier currPortId = (CommPortIdentifier) portEnum
                .nextElement();
            if (currPortId.getName().equals(port)) {
                portId = currPortId;
                break;
            }
        }
    }
}

```

```

    if (portId == null) {
        System.out.println("Could not find Mega COM port.");
        return;
    }

    try {
        serialPort = (SerialPort) portId.open(this.getClass().getName(),
            Constants.ARDUINO_TIME_OUT);
        serialPort.setSerialPortParams(DATA_RATE, SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
        serialPort.disableReceiveTimeout();
        serialPort.enableReceiveThreshold(1);
        input = new BufferedReader(new InputStreamReader(
            serialPort.getInputStream()));

        output = serialPort.getOutputStream();
        serialPort.addEventListener(this);
        serialPort.notifyOnDataAvailable(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public synchronized void close() {
    if (serialPort != null) {
        serialPort.removeEventListener();
        serialPort.close();
    }
}

public synchronized void serialEvent(SerialPortEvent oEvent) {
    if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
        try {
            if (input.ready()) {
                String inputLine = input.readLine();
                Main.setMegamesage(inputLine);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

public synchronized void writeData(String message) {
    try {
        byte[] m = message.getBytes();
        output.write(m);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

8.4.2.3. ArduinoUnoHelper.java

```

package hedubot.io;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import hedubot.base.Main;
import hedubot.utils.Constants;
import hedubot.utils.HedubotHelper;
import java.util.Enumeration;

public class ArduinoUnoHelper implements SerialPortEventListener {
    SerialPort serialPort;

    private BufferedReader input;
    private OutputStream output;

    private static final int DATA_RATE = 9600;

    public void initialize(String port) {

        CommPortIdentifier portId = null;
        Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();

        while (portEnum.hasMoreElements()) {
            CommPortIdentifier currPortId = (CommPortIdentifier) portEnum
                .nextElement();
            if (currPortId.getName().equals(port)) {
                portId = currPortId;
                break;
            }
        }

        if (portId == null) {
            System.out.println("Could not find Uno COM port.");
            return;
        }

        try {
            serialPort = (SerialPort) portId.open(this.getClass().getName(),
                Constants.ARDUINO_TIME_OUT);
            serialPort.setSerialPortParams(DATA_RATE, SerialPort.DATABITS_8,
                SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
            serialPort.disableReceiveTimeout();
            serialPort.enableReceiveThreshold(1);
            input = new BufferedReader(new InputStreamReader(
                serialPort.getInputStream()));
            output = serialPort.getOutputStream();
            serialPort.addEventListener(this);
            serialPort.notifyOnDataAvailable(true);
        } catch (Exception e) {

```

```

        HedubotHelper hh = new HedubotHelper();
        hh.stopHedubotAndClosePort();
        e.printStackTrace();
    }
}

public synchronized void close() {
    if (serialPort != null) {
        serialPort.removeEventListener();
        serialPort.close();
    }
}

public synchronized void serialEvent(SerialPortEvent oEvent) {
    if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
        try {
            if (input.ready()) {
                String inputLine = input.readLine();
                Main.unomesage = inputLine;
                System.err.println("----->Eyleyici mesajı "
                    + Main.unomesage);
            }
        } catch (Exception e) {
            HedubotHelper hh = new HedubotHelper();
            hh.stopHedubotAndClosePort();
            e.printStackTrace();
        }
    }
}

public synchronized void writeData(String message) {
    try {
        byte[] m = message.getBytes();
        output.write(m);
    } catch (Exception e) {
        e.printStackTrace();
        HedubotHelper hh = new HedubotHelper();
        hh.stopHedubotAndClosePort();
    }
}

}
}

```

8.4.2.4. Dispatcher.java

```

package hedubot.io;

import hedubot.base.Main;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

```

```

import java.net.Socket;
import javax.swing.SwingWorker;

public class Dispatcher extends SwingWorker<String, String> {
    private InputStream is;
    private OutputStream os;

    public Dispatcher(Socket socket) throws IOException {
        this.is = socket.getInputStream();
        this.os = socket.getOutputStream();
    }

    @Override
    protected String doInBackground() throws Exception {
        while (true) {

            Reader mr = new Reader();
            String message = "";
            try {
                message = mr.readInputStream(is);
                if (message != "" && !message.equals("")) {
                    System.err.println(">>>>>>>>>>" + message);
                    if (Main.getUnoah() == null) {
                        Main.setUnoah(new ArduinoUnoHelper());
                    }
                    Main.getUnoah().initialize(Main.getUNO_PORT());
                    Main.getUnoah().writeData(message + "\n");
                }
            } catch (IOException e) {
                e.printStackTrace();
                return e.getClass() + " " + e.getMessage();
            }
            Writer mw = new Writer();
            try {
                // System.err.println("<<<<<<<<<<" + Main.getMegamesage());
                mw.writeOutputStream(os, Main.megamesage);
            } catch (IOException e) {
                e.printStackTrace();
                return e.getClass() + " " + e.getMessage();
            }
        }
    }

    @Override
    protected void done() {
        return;
    }
}

```

8.4.2.5. Reader.java

```

package hedubot.io;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import hedubot.utils.Constants;

public class Reader {
    public String readInputStream(InputStream is) throws IOException {
        InputStreamReader isr = new InputStreamReader(is, "UTF-8");
        BufferedReader br = new BufferedReader(isr);
        StringBuffer sb = new StringBuffer();
        boolean eom = false;
        int c = 0;
        c = br.read();
        if (c == -1) {
            return null;
        }
        if (c != Constants.START_MESSAGE_CHARACTER) {
            return null;
        }
        while (!eom) {
            c = br.read();
            if (c == -1) {
                return null;
            }
            if (c == Constants.END_MESSAGE_CHARACTER) {
                c = br.read();
                if (c >= 0)
                    if (c != Constants.LAST_MESSAGE_CHARACTER) {
                        return null;
                    }
                eom = true;
            } else {
                sb.append((char) c);
            }
        }
        return sb.toString();
    }
}

```

8.4.2.6. Writer.java

```

package hedubot.io;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStream;

```

```

import java.io.OutputStreamWriter;
import hedubot.utils.Constants;

public class Writer {
    public void writeOutputStream(OutputStream os, String message)
        throws IOException {
        OutputStreamWriter osw = new OutputStreamWriter(os, "UTF-8");
        BufferedWriter bw = new BufferedWriter(osw);
        bw.write(Constants.START_MESSAGE_CHARACTER);
        bw.write(message);
        bw.write(Constants.END_MESSAGE_CHARACTER);
        bw.write(Constants.LAST_MESSAGE_CHARACTER);
        bw.flush();
    }
}

```

8.4.2.7. Server.java

```

package hedubot.server;

import hedubot.io.Dispatcher;
import hedubot.utils.Constants;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.SwingWorker;

public class Server extends SwingWorker<String, String> {
    private int port;
    private String ip;
    private ServerSocket serverSocket;

    private SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy hh:MM");

    public Server(int port) {

        this.port = port;
    }

    @Override
    protected String doInBackground() throws Exception {
        try {
            serverSocket = new ServerSocket(port);
            System.out.println("Server started listening port is: " + port
                + " at " + sdf.format(new Date()));
            while (true) {
                if (Constants.SERVER_IS_RUN) {
                    Socket socket = serverSocket.accept();
                    socket.setSoTimeout(0);
                    Dispatcher d = new Dispatcher(socket);
                }
            }
        }
    }
}

```



```

        d.execute();
    } else {
        serverSocket.close();
        break;
    }
}
} catch (IOException e) {
    e.printStackTrace();
    return e.getClass() + " " + e.getMessage();
}
return "Server Started on IP: " + ip + " Port: " + port + " at "
    + sdf.format(new Date());
}

@Override
protected void done() {
    try {
        if (serverSocket != null)
            serverSocket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

8.4.2.8. Constants.java

```

package hedubot.utils;

public class Constants {
    public static boolean SERVER_IS_RUN=false;
    public static final char START_MESSAGE_CHARACTER = '\u000b';
    public static final char END_MESSAGE_CHARACTER = '\u001c';
    public static final char LAST_MESSAGE_CHARACTER = '\r';
    public static final int ARDUINO_TIME_OUT = 100;
    public static final int ARDUINO_DATA_RATE = 9600;
}

```

8.4.2.9. HedubotHelper.java

```

package hedubot.utils;

import hedubot.base.Main;
import hedubot.io.ArduinoUnoHelper;

public class HedubotHelper {
    public void stopHedubotAndClosePort() {

```

```

        if (Main.unoah == null) {
            Main.unoah = new ArduinoUnoHelper();
            Main.unoah.initialize(Main.getUNO_PORT());
        }
        if (Main.unoah != null) {
            Main.unoah.writeData("jy://511,511,0\n");
            Main.unoah.close();
            Main.unoah = null;
        }
        System.gc();
    }
}

```

8.4.3. Robot Uygulaması

Robot üzerinde bulunan Arduino kartlara yüklenecek dosyalar burada verilmektedir. Bu dosyalar oluşturulup, ekte bulunan DVD içerisindeki "arduinolib" klasörü proje kütüphanesi olarak tanıtılarak robot Arduino uygulamaları derlenebilmektedir. 8.4.3.1. MotorKontrol.ino Arduino UNO kartına, 8.4.3.2. SensorVeri.ino Arduino MEGA kartına yüklenmelidir.

8.4.3.1. MotorKontrol.ino [122]

```

#include <Servo.h>
//kamera servoları
Servo servoX;
Servo servoY;
//servo x ve y açısı
int servoAciX=90;
int servoAciY=180;
//sağ motor yönetim pinleri
int sagMotorHiz = 3;
int sagMotorIleri = 8;
int sagMotorGeri = 9;
//sol motor yönetim pinleri
int solMotorHiz = 11;
int solMotorIleri = 13;
int solMotorGeri = 12;
//joystick değişkenleri
int joystickX=511;
int joystickY=511;
int joystickH=0;
//motor sürüş değişkenleri
int guc;

```

```

int yon = 0;
int solMotor;
int solMotorOlceklenen = 0;
float solMotorOlcek = 0;
int sagMotor;
int sagMotorOlceklenen = 0;
float sagMotorOlcek = 0;
float makMotorOlcek = 0;
int joystickDurmaDegeri = 10;
//sunucuya gelen komut
String komut;

void setup() {
  pinMode(sagMotorHiz, OUTPUT);
  pinMode(sagMotorIleri, OUTPUT);
  pinMode(sagMotorGeri, OUTPUT);
  pinMode(solMotorHiz, OUTPUT);
  pinMode(solMotorIleri, OUTPUT);
  pinMode(solMotorGeri, OUTPUT);
  servoX.attach(5);
  servoY.attach(6);
  servoX.write(90);
  servoY.write(180);
  Serial.begin(9600);
}

void loop() {
  while (Serial.available()>0) {
    char gelenKarakter = (char)Serial.read();
    if (gelenKarakter == '\n') {
      Serial.flush();
      if(komut.startsWith("jy://")){
        int virgulIndisi = komut.indexOf(',');
        String ilkDeger = komut.substring(5, virgulIndisi);
        komut = komut.substring(virgulIndisi+1);
        virgulIndisi = komut.indexOf(',');
        String ikinciDeger = komut.substring(0, virgulIndisi);
        String ucuncuDeger = komut.substring(virgulIndisi+1);
        joystickX=ilkDeger.toInt();
        joystickY=ikinciDeger.toInt();
        joystickH=ucuncuDeger.toInt();
        if(joystickH==3){
          --servoAciX;
          servoAciX=servoAciX<0?0:servoAciX;
        }
        else if(joystickH==7){
          ++servoAciX;
          servoAciX=servoAciX>180?180:servoAciX;
        }
        else if(joystickH==5){
          ++servoAciY;
          servoAciY=servoAciY<0?0:servoAciY;
        }
        else if(joystickH==1){
          --servoAciY;
          servoAciY=servoAciY>180?180:servoAciY;
        }
      }
    }
  }
}

```

```

else if(joystickH==6){
  servoAciX=90;
  servoAciY=180;
}
servoX.write(servoAciX);
servoY.write(servoAciY);
delayMicroseconds(100);
analogSurus();
komut="";
}
}
else{
  komut.concat(gelenKarakter);
}
}
}

void analogSurus(){
  guc = (512-joystickY)/2;
  yon = -(512-joystickX)/2;
  solMotor = guc+yon;
  sagMotor = guc-yon;
  solMotorOlcek = solMotor/255.0;
  solMotorOlcek = abs(solMotorOlcek);
  sagMotorOlcek = sagMotor/255.0;
  sagMotorOlcek = abs(sagMotorOlcek);
  makMotorOlcek = max(solMotorOlcek,sagMotorOlcek);
  makMotorOlcek = max(1,makMotorOlcek);
  solMotorOlceklenen = constrain(solMotor/makMotorOlcek,-255,255);
  sagMotorOlceklenen = constrain(sagMotor/makMotorOlcek,-255,255);
  if(abs(solMotorOlceklenen)>joystickDurmaDegeri){
    if (solMotorOlceklenen > 0){
      digitalWrite(solMotorIleri, HIGH);
      digitalWrite(solMotorGeri, LOW);
      analogWrite(solMotorHiz, abs(solMotorOlceklenen));
    }
    else{
      digitalWrite(solMotorIleri, LOW);
      digitalWrite(solMotorGeri, HIGH);
      analogWrite(solMotorHiz, abs(solMotorOlceklenen));
    }
  }
  else{
    digitalWrite(solMotorIleri, LOW);
    digitalWrite(solMotorGeri, LOW);
  }
  if(abs(sagMotorOlceklenen)>joystickDurmaDegeri){
    if (sagMotorOlceklenen > 0){
      digitalWrite(sagMotorIleri, HIGH);
      digitalWrite(sagMotorGeri, LOW);
      analogWrite(sagMotorHiz, abs(sagMotorOlceklenen));
    }
    else{
      digitalWrite(sagMotorIleri, LOW);
      digitalWrite(sagMotorGeri, HIGH);
      analogWrite(sagMotorHiz, abs(sagMotorOlceklenen));
    }
  }
}

```

```

}
else{
    digitalWrite(sagMotorIleri, LOW);
    digitalWrite(sagMotorGeri, LOW);
}
}
}

```

8.4.3.2. SensorVeri.ino

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#include <DHT.h>

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
const int trigPin1 = A0;
const int echoPin1 = A1;
NewPing mesafeSensoruOnSag(trigPin1, echoPin1, 200);
const int trigPin2 = A2;
const int echoPin2 = A3;
NewPing mesafeSensoruOnOrta(trigPin2, echoPin2, 200);
const int trigPin3 = A4;
const int echoPin3 = A5;
NewPing mesafeSensoruOnSol(trigPin3, echoPin3, 200);
const int trigPin4 = A10;
const int echoPin4 = A11;
NewPing mesafeSensoruArkaSag(trigPin4, echoPin4, 200);
const int trigPin5 = A8;
const int echoPin5 = A9;
NewPing mesafeSensoruArkaOrta(trigPin5, echoPin5, 200);
const int trigPin6 = A6;
const int echoPin6 = A7;
NewPing mesafeSensoruArkaSol(trigPin6, echoPin6, 200);
DHT dht;
unsigned long sonDHTIstegi= 0;
float isi = 0.0;
float nem = 0.0;
int kirmiziLed1=5;
int beyazLed=6;
int kirmiziLed2=7;
int hareketSensoru=42;
int atesSensoruSol=43;
int atesSensoruOrta=44;
int atesSensoruSag=45;
int engelSensoruSol=39;
int engelSensoruOrta=40;
int engelSensoruSag=41;
String sensorler;
unsigned long sonEkranDegisimiZamani= 0;
int ekranPeriyodu=5000;
int ekran=1;

```

```

void setup(){
  lcd.begin(16,2);
  Serial.begin(9600);
  pinMode(hareketSensoru,INPUT);
  pinMode(atesSensoruSol,INPUT);
  pinMode(atesSensoruOrta,INPUT);
  pinMode(atesSensoruSag,INPUT);
  pinMode(engelSensoruSol,INPUT);
  pinMode(engelSensoruOrta,INPUT);
  pinMode(engelSensoruSag,INPUT);
  pinMode(trigPin1,OUTPUT);
  pinMode(echoPin1,INPUT);
  pinMode(trigPin2,OUTPUT);
  pinMode(echoPin2,INPUT);
  pinMode(trigPin3,OUTPUT);
  pinMode(echoPin3,INPUT);
  pinMode(trigPin4,OUTPUT);
  pinMode(echoPin4,INPUT);
  pinMode(trigPin5,OUTPUT);
  pinMode(echoPin5,INPUT);
  pinMode(trigPin6,OUTPUT);
  pinMode(echoPin6,INPUT);
  pinMode(kirmiziLed1,OUTPUT);
  pinMode(beyazLed,OUTPUT);
  pinMode(kirmiziLed2,OUTPUT);
  dht.setup(38);
  lcd.backlight();
}

void loop()
{
  int hareketDegeri=digitalRead(hareketSensoru);
  int solAtesDegeri=digitalRead(atesSensoruSol);
  int ortaAtesDegeri=digitalRead(atesSensoruOrta);
  int sagAtesDegeri=digitalRead(atesSensoruSag);
  int solEngelDegeri=digitalRead(engelSensoruSol);
  int ortaEngelDegeri=digitalRead(engelSensoruOrta);
  int sagEngelDegeri=digitalRead(engelSensoruSag);
  int sagOnMesafe=mesafeOku(mesafeSensoruOnSag);
  int ortaOnMesafe=mesafeOku(mesafeSensoruOnOrta);
  int solOnMesafe=mesafeOku(mesafeSensoruOnSol);
  int sagArkaMesafe=mesafeOku(mesafeSensoruArkaSag);
  int ortaArkaMesafe=mesafeOku(mesafeSensoruArkaOrta);
  int solArkaMesafe=mesafeOku(mesafeSensoruArkaSol);
  if (millis() - sonDHTIstegi >= dht.getMinimumSamplingPeriod())
  {
    nem = dht.getHumidity();
    isi = dht.getTemperature();
    sonDHTIstegi = millis();
  }
  if (millis() - sonEkranDegisimiZamani >= ekranPeriyodu)
  {
    if(ekran==1){
      ekran=2;
    }
    else if(ekran==2){
      ekran=3;
    }
  }
}

```

```

    }
    else if(ekran==3){
        ekran=1;
    }
    sonEkranDegisimiZamani=millis();
}
if(ekran==1){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HEduBot M/F/B");
    lcd.setCursor(0,1);
    lcd.print(hareketDegeri);
    lcd.setCursor(2,1);
    lcd.print(!solAtesDegeri);
    lcd.setCursor(4,1);
    lcd.print(!ortaAtesDegeri);
    lcd.setCursor(6,1);
    lcd.print(!sagAtesDegeri);
    lcd.setCursor(8,1);
    lcd.print(!solEngelDegeri);
    lcd.setCursor(10,1);
    lcd.print(!ortaEngelDegeri);
    lcd.setCursor(12,1);
    lcd.print(!sagEngelDegeri);
}
else if(ekran==2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HEduBot Temp. Hum.");
    lcd.setCursor(0,1);
    lcd.print(isi);
    lcd.setCursor(6,1);
    lcd.print(nem);
}
else if(ekran==3){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(sagOnMesafe);
    lcd.setCursor(4,0);
    lcd.print(ortaOnMesafe);
    lcd.setCursor(9,0);
    lcd.print(solOnMesafe);
    lcd.setCursor(0,1);
    lcd.print(sagArkaMesafe);
    lcd.setCursor(4,1);
    lcd.print(ortaArkaMesafe);
    lcd.setCursor(9,1);
    lcd.print(solArkaMesafe);
}
analogWrite(kirmiziLed1,5);
analogWrite(beyazLed,20);
analogWrite(kirmiziLed2,50);
sensorler.concat(hareketDegeri);
sensorler.concat(";");
sensorler.concat(!solAtesDegeri);
sensorler.concat(";");
sensorler.concat(!ortaAtesDegeri);

```

```
sensorler.concat(";");
sensorler.concat(!sagAtesDegeri);
sensorler.concat(";");
sensorler.concat(!solEngelDegeri);
sensorler.concat(";");
sensorler.concat(!ortaEngelDegeri);
sensorler.concat(";");
sensorler.concat(!sagEngelDegeri);
sensorler.concat(";");
sensorler.concat(sagOnMesafe);
sensorler.concat(";");
sensorler.concat(ortaOnMesafe);
sensorler.concat(";");
sensorler.concat(solOnMesafe);
sensorler.concat(";");
sensorler.concat(sagArkaMesafe);
sensorler.concat(";");
sensorler.concat(ortaArkaMesafe);
sensorler.concat(";");
sensorler.concat(solArkaMesafe);
sensorler.concat(";");
sensorler.concat(isi);
sensorler.concat(";");
sensorler.concat(nem);
sensorler.concat(";");
Serial.println(sensorler);
sensorler="";
}
```

```
int mesafeOku(NewPing sensor) {
    unsigned int uS = sensor.ping();
    int cm = uS/US_ROUNDTRIP_CM;
    return cm;
}
```


ÖZGEÇMİŞ

1983 Aksaray doğumlu Mahmut KASAP, 2006 yılında Ahmet Yesevi Uluslararası Türk-Kazak Üniversitesi Bilgisayar Mühendisliği bölümünden mezun olmuştur. Tez Tour, Akgün Yazılım, NKR yazılım, SYS yazılım firmalarında yazılım uzmanı olarak görev yaptıktan sonra, 2010 yılında Gazi Üniversitesi Uzaktan Eğitim Meslek Yüksekokulu'nda Öğretim Görevlisi olarak göreve başlamış, 2013-2014 eğitim öğretim yılında Gazi Üniversitesi Kazan Meslek Yüksekokulu'na atanmıştır, halen görevine devam etmektedir. 36 farklı derste öğretmenlik yapmış olan Mahmut KASAP, evli ve bir çocuk babasıdır.