

**T.C.**  
**KARABÜK ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**MEKATRONİK MÜHENDİSLİĞİ**



**LİSANS BİTİRME TEZİ**

**ROBOTLARDA GÖRÜNTÜ İŞLEME TEKNİKLERİ VE**  
**UYGULANMALARININ İNCELENMESİ**

**BİTİRME TEZİ**

**Hazırlayanlar**

Erhan ERDOĞAN	2012010226071
Kamil ŞİMŞEK	2012010226039
Burak KAPUSIZ	2012010225059

**Tez Danışmanı**

**Yrd.Doç.Dr. Aytül BOZKURT**

**KARABÜK-2017**

# İÇİNDEKİLER

	<u>Sayfa</u>
İÇİNDEKİLER.....	1
KABULVE ONAY.....	3
ÖNSÖZ.....	4
TEŞEKKÜR.....	4
ÖZET.....	5
SUMMARY.....	6
ŞEKİLLER LİSTESİ.....	7
1.GİRİŞ.....	8
2.KURAMSAL ALTYAPI.....	9
2.1 Robot Kolda Kullanılan Malzemelerin Tanıtımı .....	9
2.1.1Servo Motorlar.....	9
2.1.2Dc Motorlar.....	12
2.1.3Motor Sürücü Mikrodenetleyici.....	17
2.1.4Bluetooth Modülü.....	20
2.1.5Arduino Mikrodenetleyici.....	21
2.1.6Devre Şeması.....	26
2.1.7Li-po Piller.....	28
2.2 Görüntü İşlemede Kullanılan Malzemelerin Tanıtımı .....	30
2.2.1Kameralar.....	30
3. ROBOT KOL.....	32
3.1 Robot Kol Tanıtımı.....	32
3.2 Robot kol kinematiği.....	41
3.3 Düz Kinematik.....	41
3.4 Ters Kinematik.....	42
3.5 Robot Kol Tasarımı.....	42
3.5.1 Mekanik Kısım Tasarımı.....	42
3.5.2 Mekanik Kısım Montajı.....	43
3.6 Android Programlama.....	47
3.7 Arduino Kodları.....	47
4. GÖRÜNTÜ İŞLEME.....	50
4.1 Görüntü İşleme Tanımı.....	50

4.2 Görüntü Yakalama .....	50
4.3 Sayısal Görüntü.....	51
4.4 Görüntü İşleme Adımları.....	52
4.5 Seri Port İle Haberleşme.....	53
4.6 Görüntü İşleme Teknikleri.....	54
4.7 Matlab Programına Web Kamerasının Tanıtılması.....	57
4.8 Matlab Görüntü İşleme Kodları.....	59
4.8.1 Matlab Script ile İleri ve Ters Kinematik Hesaplamalar.....	74
4.8.2 Matlab İle Nesne Takibi.....	74
5. ROBOTLARDA GÖRÜNTÜ İŞLEME UYGULAMALARI.....	80
5.1Tasarım ve İmalat Uygulamaları.....	81
5.2Savunma ve Güvenlik Uygulamaları.....	82
5.3Tıp Alanında Uygulamaları.....	84
5.4Mimari Uygulamaları.....	84
5.5Harita ve Jeodezi Uygulamaları.....	85
6.GENEL SONUÇLAR.....	86
KAYNAKLAR.....	87
ÖZGEÇMİŞ.....	88

### KABUL VE ONAY

Erhan ERDOĞAN, Kamil ŞİMŞEK, Burak KAPUSUZ.. tarafından hazırlanan  
“ROBOTLARDA GÖRÜNTÜ İŞLEME TEKNİKLERİ VE UYGULANMALARININ  
İNCELENMESİ” başlıklı bu tezin Lisans Bitirme Tezi olarak uygun olduğunu onaylarım.

25/05/2017

  
Tez Danışmanı

Yrd. Doç. Dr. Aytül BOZKURT

..

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Mekatronik Mühendisliği Anabilim  
Dalında Lisans Bitirme Tezi olarak kabul edilmiştir. 25/05/2017

#### Tez Jürisi

Başkan: Yrd. Doç. Dr. İbrahim ÇAYIROĞLU

Üye : Yrd.Doç. Dr. Aytül BOZKURT

Üye :

Prof. Dr. Gökhan Göktaş

KBÜ Mühendislik Fakültesi, Mekatronik Mühendisliği Mezuniyet Komisyonu ve Bölüm  
Başkanlığı bu tezi Lisans Bitirme Tezi olarak onamıştır. .25./05./2017

Yrd.Doç.Dr.İbrahim ÇAYIROĞLU  
Mekatronik Müh. Bölüm Başkanı

## ÖNSÖZ

**Robot**, otonom veya önceden programlanmış görevleri yerine getirebilen elektro-mekanik bir cihazdır. **Robotlar** doğrudan bir operatörün kontrolünde çalışabildikleri gibi bağımsız olarak bir bilgisayar programının kontrolünde de çalışabilir. Robotik, Makine Mühendisliği, Endüstri Mühendisliği, Elektrik ve Elektronik Mühendisliği ve Bilgisayar Mühendisliği disiplinlerinin ortak çalışma alanıdır. Bunların yanı sıra robotik, matematik ve fizik bilimlerinden de yararlanır. Robotik bütün dünyada olduğu gibi ülkemizde de büyük ilgi görmektedir. Robotik'e yakın bir konu Mekatronik'tir. Üniversitelerde **Mekatronik** adıyla Robotik Bölümleri açılmakta, ön lisans ve lisans düzeyinde eğitimler verilmektedir.

Robot teknolojisi, çağımız gelişim süreci içinde gelişen birçok bilimsel ve teknolojik olguların, robot adını verdiğimiz teknolojik ürünler üzerinde bütünleşmesi ve uygulamasını içerir.

Hayatımızı kolaylaştıran robotların yanında bir de bunun öncesi olan iş yaşamımızı daima kolaylaştıran endüstriyel robotlar var. Endüstriyel robotların bir vida sıkımından kutuları istiflemeye kadar birçok alanda kullanımı hem zamandan, hem iş gücünden hem de maddi olarak kazanım sağlar.

## TEŞEKKÜR

Öncelikle tez konusunu seçerken bizim sadece teoride değil uygulama konusunda da gelişmemizi sağlayan, iş hayatımızda fazlasıyla işimize yarayacak olan görüntü işleme, robot kol ve seri haberleşme hakkında araştırma yapmamıza, bilgi edinmemize yardımcı olan Tez Danışmanımız Yrd. Doç. Dr. Aytül BOZKURT Hocamıza göstermiş olduğu ilgi ve emeklerinden dolayı çok teşekkür ederiz. Eğitim hayatımız boyunca yanımızda olan maddi manevi hiçbir desteği bizlerden esirgemeyen ailelerimize teşekkürü borç biliriz.

## ÖZET

Bu projedeki amaç 4 eksenli hareket kabiliyetine sahip robot kullanılarak matlab aracılığıyla görüntü işlenerek renkleri ayırma işlemi yapmaktır. Bu projede otomasyon sistemlerinde kullanılabilen bir robot kolun elektrik devreleri, mekanik yapıları ve yazılımı hakkında bilgiler yer almaktadır.

Proje deneysel bir çalışma olup kamera tarafından algılanıp matlab tarafından tanımlanan rengine göre malzemenin taşınması gereken yerin açısal değerleri ayarlanıp motorları konumlandırarak robot kolu hareket ettirmek üzere gerçekleştirilmiştir. Robotun mekanik aksamında servo motorlara yük binmemesi için ağırlık olarak hafif olan pleksiglas malzeme ve servo motorun dişli takımları kullanılmıştır. Robotun imalatında kullanılan parçaların teknik resim çizimleri projede yer almaktadır. Robotun elektronik kartı ve robotun kontrolünü gerçekleştiren yazılımda projede bulunmaktadır. Projede mikro denetleyici olarak Atmel AVR mikro denetleyiciye sahip Arduino Uno kullanılmaktadır. Arduino kartları üzerinde Atmel firmasının üretmiş olduğu Atmega serisi 8 bit mikro denetleyiciler bulunuyor. Bu mikro denetleyicilerin de PIC'ler ile aynı kategoride olduğunu söyleyebiliriz. Bunun dışında Arduino donanımlarında başka özel bir komponent bulunmuyor. Kartlara göre farklılık göstermekle beraber bir başka 8 bit mikro denetleyici de Arduino ile bilgisayar arasındaki USB iletişimi için kullanılıyor.

Arduino'yu bu kadar popüler kılan özelliklerinden “kolay kullanımı” oluşturan en büyük etken geliştirme ortamı. Geliştirme ortamı Processing Programlama dili/ortamına dayanıyor. Arka planda Atmega mikro denetleyicileri için kullanılan GNU GCC derleyicisi ve derlenen programların mikro denetleyiciye yüklenmesinde kullanılan AVR-Dude yazılımı bulunuyor. Geliştirme ortamı kodları derleyip kolayca mikro denetleyiciye yüklemenizi sağlıyor. Arduino kütüphaneleri birçok işlemi donanım seviyesine inmeden yani mikro denetleyicinin kaydedicileri üzerinde işlemler yapmaya gerek olmadan yapmanızı sağlıyor. Bu avantajlarından yararlanarak projede Arduino Uno kullanmaya karar kılınmıştır.

## **SUMMARY**

Robot is an electro-mechanical device capable of performing autonomous or pre-programmed tasks. While they can directly work in the control of an operator, also work under the control of a computer program independently. Robotics is the common workplace of the disciplines; Mechanical, Industry, Electrics-Electronics and Computer Engineering. Beside all of these, Robotics benefits from the science of mathematics and physics. Robotics has a great interest in our country as well as in the whole world. Mechatronics is a similar field to Robotics. Robotics departments are opened under the name of Mechatronics at universities as undergraduate and graduate programs.

Robot technology includes the application and integration of the technological products, which are named as robot. Those products have had many scientific and technological phenomena that develop within the process of modernization.

Besides the robots that ease our lives, also there are industrial robots which make our work life easier. From tightening a screw to stacking a box, the industrial robot usage supplies a recovery of time, workforce and finance.



## ŞEKİLLER LİSTESİ

(Şekil 2.1 SG-90 servo motor modeli).....	9
(Şekil 2.2 Servo motor yapısı).....	10
(Şekil 2.3 Sanayi tipi servo motorların yapısı).....	11
(Şekil 2.4 DC motor modeli).....	12
(Şekil 2.5 Akım geçen iletkenin manyetik alan içindeki durumları).....	15
(Şekil 2.6 DC motor yapısı).....	15
(Şekil 2.7 Sargı yapısı).....	16
(Şekil 2.8 DC motor endüktör yapısı).....	16
(Şekil 2.9 DC motor parçaları).....	17
(Şekil 2.10 Mikrodenetleyici modeli).....	18
(Şekil 2.11 L293D datasheet).....	19
(Şekil 2.12 Bluetooth modeli).....	21
(Şekil 2.13 Arduino programlama arayüzü).....	24
(Şekil 2.14 Arduino Uno pinleri ).....	26
(Şekil 2.15 Fritzing örneği).....	27
(Şekil 2.16 Lipo pil modeli).....	28
(Şekil 3.1 Robot kol benzetimi).....	32
(Şekil 3.2 Sanayi robotları).....	34
(Şekil 3.3 Kartezyen robot kinematiği).....	36
(Şekil 3.4 Silindirik robot kinematiği).....	37
(Şekil 3.5 Küresel robot kinematiği).....	37
(Şekil 3.6 Scara robot kinematiği).....	38
(Şekil 3.7 Mafsallı robot kinematiği).....	39
(Şekil 3.8 Paralel robot modeli).....	40
(Şekil 3.9 Mekanik kısım tasarımı).....	42
(Şekil 3.10 Mekanik kısım montajı).....	43
(Şekil 3.11 Mıt App Inverto).....	47
(Şekil 4.1 Görüntü yakalama ve sayısallaştırma aşamaları).....	51
(Şekil 4.2 Piksel imgesi).....	52
(Şekil 4.3 Görüntü işleme adımları).....	52
(Şekil 4.4 Görüntü işleme teknikleri).....	54

(Şekil 4.5 Görüntü karşılaştırılması).....	54
(Şekil 4.6 BAYRAK).....	55
(Şekil 4.7 JPG dosya özellikleri gösterimi).....	56
(Şekil 4.8 Resim kayıt seçimi).....	56
(Şekil 4.9 Görüntü formatlarına piksellere göre sınıflandırılması).....	56
(Şekil 4.10 Matlab guide arayüz).....	59
(Şekil 5.1 Soldan sağa doğru kayan pencere bölümlemesi).....	81
(Şekil 5.2 Akıllı etiket ile elde edilen sonuçlar).....	82
(Şekil 5.3 Geliştirilen sistem doğrulama modu parmak izi).....	83
(Şekil 5.4) Plaka karakterlerinin ayrıştırılması.....	83
(Şekil 5.5 Geliştirilen mimari modelleme sistemi ile nesne tanıma).....	84
(Şekil 5.6 Harita ve jeodezi uygulamaları).....	85

## 2.KURAMSAL ALTYAPI

### 2.1 Robot Kolda Kullanılan Malzemelerin Tanıtımı

#### 2.1.1 Servo Motorlar

Servo motor, elektrik kontrolüyle gerçekleştirilen servo sistemin hareketini sağlayan birimdir. Pozisyon ve hız kontrolünün gerektiği uygulamalarda, pozisyon, hız bilgisi, geri besleme ile bir karar verme ünitesine gönderilerek sistemin davranışı kontrol edilmektedir. İçinde adım motor kullanılmayan kapalı devre sistemler de servo sistem diye adlandırılmaktadır. Bu yüzden hız kontrolü yapılan basit bir asenkron motor da servo motor olarak adlandırılabilir. Servo sistemlerde çeşitli yapıda farklı elektrik motorları kullanılabilir. Servo motorları normal motorlardan farklı kılan, çok geniş bir hız komutunu yerine getirecek şekilde tasarlanmış olmalarıdır. Bir servo motor şu özelliklere sahip olmalıdır;

- Geniş bir hız yelpazesinde kararlı olarak çalışabilmelidir.
- Devir sayısı, hızlı ve düzgün bir şekilde değiştirilebilmelidir.
- Küçük boyuttan büyük moment elde edilebilmelidir.

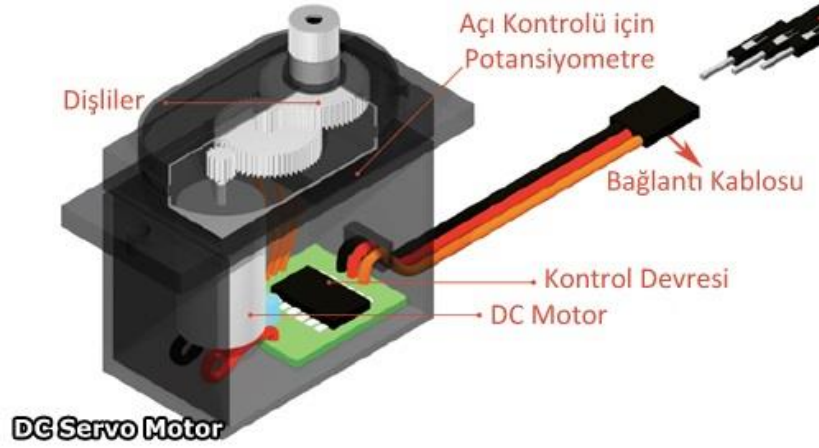


(Şekil 2.1 SG-90 servo motor modeli)

#### Servo Motorun Yapısı

Servo motorlar bir motor ve geri besleme ünitesinden meydana gelir. Motor DC ya da AC olabilir. Geri besleme ünitesi de bir takometer ya da enkoder (kodlayıcı) olabilir.

Servo motorlar, 1 d/d' lik hız bölgelerinin altında bile kararlı çalışabilen, hız ve moment kontrolü yapan motorlardır. Servo motorlar ile diğer motorlar arasındaki birinci fark, çok hızlı ivmelenme ve frenleme yapabilmeleridir. Bunun için döndürme momentinin büyük, eylemsizlik momentinin küçük olması gerekir.



(Şekil 2.2 Servo motor yapısı)

### Servo Motor Çeşitleri

Servo motorlar, AC servo motorlar ve DC servo motorlar olarak ikiye ayrılmaktadır.

#### DC Servo Motor

DC servo motorlar, çok küçük güçlerden çok büyük güçlere kadar imal edilirler (0,05 HP' den 1000 HP' ye kadar). DC servo motorların rotorları; uzun, disk ve çan şeklinde olabilmektedir. Disk rotorlu servo motorlar, kısa ve hafif oldukları için robot mafsallarında hareket elemanı olarak kullanılmaktadırlar. İnce ve uzun rotorlu servo motorlarda ise boyutlar çok küçük olduğundan monte işlemi kolaydır. Yüksek bir dönme momenti ve aşırı yüklene bilirlilik elde etmek için özellikle şiddetli bir manyetik alan oluşturulmalıdır. DC servo motorun en büyük dönme momenti, düşük hızlı çalışmalar sırasında ve en küçük devir sayısında ise yaklaşık anma dönme momentinin dört katıdır. Yapısında endüktör, endüvi, gövde, fırça ve kolektör bulunur.

#### DC servo motor çeşitleri

DC servo motorlar, bir servo sistemde çalışırken endüvisinden ya da kutuplarından kontrol edilir. Kutuplar gerilim kaynağından ya da akım kaynağından beslenir. Her iki tür uygulama farklı bir hız-moment karakteristiğinin ortaya çıkmasına sebep olur. Endüstride daha çok dört tip servo motor kullanılır. Bunlar:

- Alan kontrollü-sabit endüvi gerilim beslemeli servo motor
- Endüvi kontrollü-sabit alan beslemeli servo motor
- Sabit mıknatıslı- endüvi kontrollü servo motor

## AC servo motor

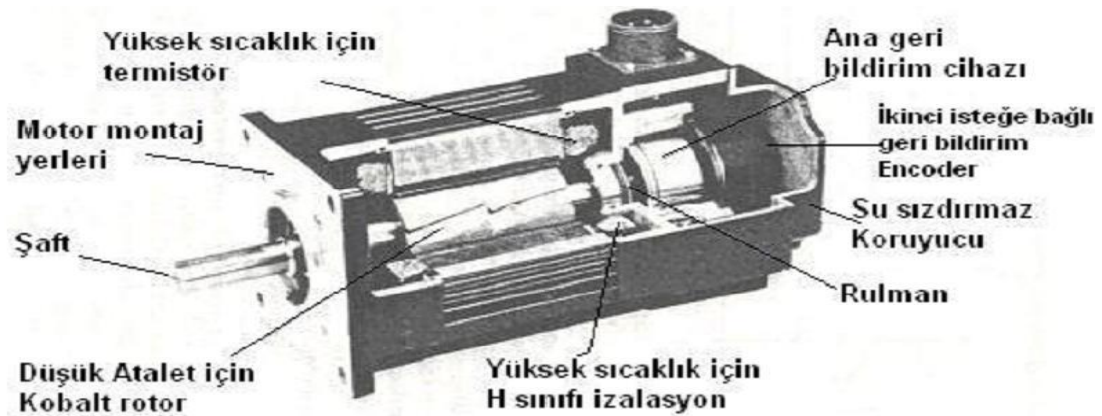
Bu tip servo motorlar, genellikle iki fazlı sincap kafesli asenkron tipi motorlardır. İki fazlı asenkron motorlar, büyük güçlü yapılmakla birlikte çoğunlukla otomatik kontrol sistemlerinde servo motorlar olarak kullanılmak amacı ile küçük güçlü yapılır. Fırça ve kollektör olmadığından arıza yapma ihtimalleri az, bakımları kolaydır.

## AC servo motorun yapısı

Büyük ve güçlü AC servo motorlar iki ya da üç fazlı olarak üretilmektedir. Bu tip motorların rotorları, doğal mıknatıslı ya da kısa devre çubuklu şeklinde olmaktadır. İki ya da üç fazlı servo motorların çalışma prensibi, senkron tipi motorlara çok benzemektedir. Üç fazlı servo motorların hız kontrolü, darbe frekans çevirici devresi üzerinden darbe genişlik modülasyonu (PWM) ile yapılmaktadır. Küçük ve güçlü AC servo motorlar ise minik boyutlu olarak iki faz ile çalışabilecek şekilde üretilirler.

Bunların içyapısında aralarında, 90° elektriksel açı yapacak şekilde yerleştirilmiş iki bobin ve sincap kafesine benzer rotor vardır. Bunların rotorları, savrulma ve atalet momentlerinin küçük olabilmesi için uzun, çapları ise küçük yapılır. İki fazlı servo motorlar hava yolu sistemlerinde kullanılmaktadır.

Üç fazlı servo motor yüksek güçlü servo sistemlerin uygulama alanlarında kullanılmaktadır. 3 fazlı asenkron motor yapı olarak dayanıklı olmakla beraber doğrusal olmayan bir özelliğe sahiptir ve bundan dolayı kontrolü karmaşıktır. Genel olarak AC servo motorların kullanım alanları; robotlar, radarlar, nümerik kontrollü makinelerde (CNC), otomatik kaynak makinelerinde, pres makinelerinde, paketleme makinelerinde, yarı iletken üretim ünitelerinde, yüksek hızlı çip yerleştiricilerinde, tıbbi cihazlarda, anten sürücülerinde vb. yerlerde kullanılır.



(Şekil 2.3 Sanayi tipi servo motorların yapısı)

### 2.1.2 Doğru Akım(DC) Motorları

Elektrik enerjisini, mekanik enerjiye çeviren elektrik makinasına doğru akım motoru denir. Doğru akım motorlarına DA motor veya DC motor da denilmektedir. Buna bağlı olarak doğru akım makinası, doğru akım genaratörü ya da doğru akım motoru olarak da geçmektedir. Doğru akım motorları herhangi bir iletkeni doğru akım uygulandığında iletken, sabit bir manyetik alan meydana getirir. N ve S kutuplarından meydana gelen bu sabit manyetik alan etki alanının içindeki iletken cisimlere ya da farklı manyetik alanlara sabit mıknatısın gösterdiği etkiyi göstermektedir. İletken cisimleri çeker, aynı kutuplu manyetik alanları iter; farklı kutuplu manyetik alanları çeker. N kutbundan S kutbuna doğru meydana gelen bu kuvvete manyetik akı adı verilmektedir. DC motor statorun yapısına göre sabit mıknatıslı ve elektromıknatıslı olarak DC motor üreticileri tarafından, DC motor imalatı yapılır. Elektromıknatıslı ve sabit mıknatıslı arasında stator dışında yapıları açısından farklılık yoktur. Doğru akım motorları, duran ve dönen kısımlardan meydana gelir. Doğru akım makineleri dönen kısım (endüvi), duran kısım (endüktör), yatak, kapak, fırça ve kolektörden oluşmaktadır.



(Şekil 2.4 DC motor modeli)

#### Endüktör (Duran Kısım)

Görevi manyetik alan meydana getirmektir. Endüktör sargısı DA makinesinin gövdesinde bulunur, vida veya somunlarla gövdeye tutturulur. Doğru akım makinesinin özelliğine göre endüktör sargısı yapısal değişiklikler göstermektedir. Küçük güçlü DA makinelerinde ve pilli oyuncakta daimi mıknatıs, endüktör olarak görev yapmaktadır. Doğru akım motorlarında kutup sayısı, alternatif akım makinelerinde olduğu gibi hız, indüklenen gerilim ve akımın frekansına bağlı değildir. Burada kutup sayısı makinenin gücüne ve devir sayısına göre değişmektedir. Endüktör, makinenin gücüne (büyüklüğüne, çapına) ve devir sayısına göre 2, 4, 6, 8 veya daha çok kutuplu olur.

### **Endüvi (Dönen Kısım)**

DA makinelerinde dönen, mekanik enerjinin alındığı kısımdır. Doğru akım makinesinin yapısına göre çeşitli ebatlarda yapılmaktadır. Endüvi üzerinde kolektör ve preslenmiş sac paket bulunur. Sac üzerindeki emaye yalıtkanlı iletkenlerden akım geçtiğinde motor olarak çalışır yani dönmektedir. Manyetik alan içindeki endüvi dışarıdan bir kuvvetle döndürülürse DA gerilim üretir yani dinamo görevi yapar.

Periyodik aralıklarla endüvinin bakımının yapılması, kollektör yüzeyinin temizlenmesi gerekir. Büyük, güçlü doğru akım makineleri yüksek akım çekmektedir. Bundan dolayı kollektöre iki ya da daha fazla fırça ile doğru gerilim uygulanır. Endüvi üzerinde kolektör vardır ve bakır dilimlerden oluşmaktadır. Endüvide bulunan iletkenler bu dilimlere lehimlenerek bağlanır.

### **Kollektör**

Kollektör sayesinde doğru akım motorlarında endüviye gerilimin iletilmesi sağlanır. Statorda üretilen alternatif gerilim kolektör tarafından doğrultulur ve kolektör üzerine temas eden fırçalar yardımı ile dış devreye iletilir. Kollektör birbirinden tek tek yalıtılmış sert bakır levhalardan oluşmaktadır. Bu levhalar bakalit dökümü veya mekanik bir sıkıştırma yöntemi ile birleştirilmiştir. Endüvide üretilen alternatif gerilim, kollektör tarafından doğrultulur ve kollektör üzerine temas eden fırçalar yardımıyla dış devreye iletilmektedir. Kollektör dilimleri yüksek ısıya dayanıklı sert bakırdan yapılmaktadır.

Yalıtkanlığı sağlamak için bakır dilimlerin arasına 0,5 –1,5 mm yalıtkan maddeler kullanılır. Yalıtkan maddeler, dilimler arasındaki gerilim farkına göre değişim gösterir. Kollektör dilimlerine değen fırçalar, bağlama elemanlarını oluşmaktadır. Doğru akım motorlarında önemli yeri olan kollektörler, çok sayıda arıza veren elemanıdır.

### **Fırça ve Fırça Yatağı**

Dış devrede ki akımı, fırçalar yardımı ile endüviye ileten sistemde doğru akım motorları kullanılır. İletimi iyi sağlayan ve aşınmayı asgariye indiren fırça kullanılır. Yani aşınmayı fazlaştıran saf bakır kullanılmaz. Kollektör ile elektriksel bağlantıyı sağlayan kömürden yapılmış fırçalar, fırça tutucuları ve fırça taşıyıcıları monte edilmiştir. Fırçalar, makinanın akım ve gerilimine göre sert, orta, yumuşak karbonlardan imal edilmektedir. Bazen iletkenliği arttırmak için karbon bakır alaşımlı yapırlardır. Fırça boyları eşit olmalı, aynı tür fırçalar bir motorda kullanılmalıdır.

Fırçalar yatay ve dikey biçimde imal edilirler. Küçük ve güçlü motorlarda fırçalar, yalıtılmış, kapak üzerine açılmış yuvalara konulur. Güçlü ve büyük olan motorlarda fırça yuvası dökümden ve sacdan yapılmış fırça tutucularına montajı yapılmalıdır. Fırça tutucuları, fırçaların kolektör üzerinde durmalarını ve bu durumun işletme sırasında devamlılığını sağlar. Kollektör üzerine oturan fırça, kollektör yüzeyine fırça tutucuları ve yay düzeneği sayesinde itilmektedir. Eğik ve dik olmak üzere çeşit fırça tutucuları vardır. Fırça taşıyıcıları, fırça tutucuları ile birlikte montajı yapılmaktadır.

### **Yataklar ve Kapaklar**

Doğru akım motorlarında kendi kendine yağlanan metal yataklar veya bilyeli yataklar kullanılır. Bu yataklar, gürültüsüz az bir kayıpla motorun hareket kısmının bir eksen etrafında rahat dönmesini sağlar. Bilezikli ve rulmanlı yataklar doğru akım motorlarında kullanılır. Küçük ve orta güçlü motorlarda sürtünmenin ve kayıpların az olması değişiminin kolay olması açısından rulmanlı yatak kullanılır. Bu yatağın dezavantajı:

- Çok gürültülü olmasıdır.
- Arızaları fazladır.
- Bakım gerektirirler.

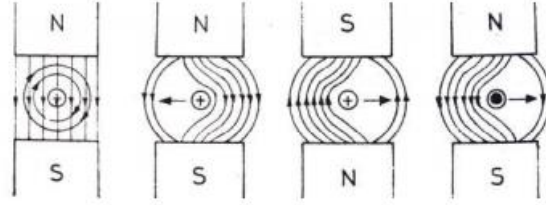
Yataklarda meydana gelen aşınmalar ve sürtünmeler, komütasyonun bozulmasına ve arızalara yol açar. DC motor parçaları kapaklar, vantilatör, ayaklar, taşıma kancası ve kablo klemensleri gibi parçalarıda bulunmaktadır.

### **DA Motorunun Çalışması**

Doğru akım motoru, içinden akım geçen iletkenin manyetik ortam dışına itilmesi prensibine göre çalışır. Motorlarda manyetik alanı endüktör oluşturmaktadır. İçinden akım geçen iletkenler ise endüvi üzerinde bulunur. Endüvi üzerindeki iletkenlere fırça ve kolektör yardımıyla gerilim uygulanır. Böylece endüvi üzerindeki iletkenlerden akım geçer ve manyetik alan oluşur. Endüviden geçen akım, manyetik alan oluşturur ve kutuplarda oluşan manyetik alanı bozmaktadır. Bu durumda endüvi reaksiyonu oluşur. Bu istenmeyen durumu ortadan kaldırmak için yardımcı kutup kullanılır. Bazı küçük güçlü motorlarda yardımcı kutup olmayabilir. Endüktör sargısının manyetik alanı (N-S), endüvide üzerinde manyetik alan oluşturan iletken veya iletken demetini dışa doğru itmektedir.



Bu itilme, mil etrafında dönmeyi meydana getirir. İletkenden geçen akım yön değiştirirse itilme yönü de değişir. İtilme yönünün değişmesi motorun dönüş yönünü de değiştirir.



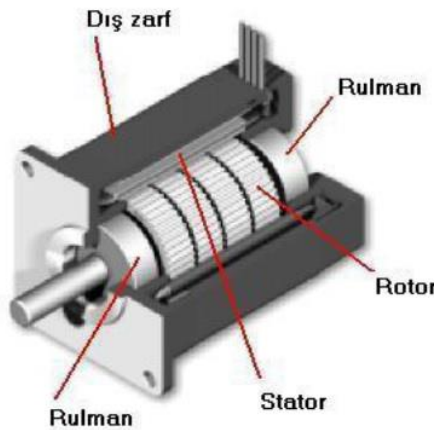
(Şekil 2.5 Akım geçen iletkenin manyetik alan içindeki durumları)

### DA Motorlarında Uyarım

Doğru akım motorlarının çalışması için endüktörde manyetik alan bulunması ve endüviden akım geçmesi gerekir. Endüktöre doğru akım uygulandığında manyetik alan yani mıknatısiyet oluşturmaktadır. Doğru akım motorunun endüktör sargısının manyetik alan oluşturmak için dışarıdan çektiği akıma uyarım akımı denir. Endüktör sargısının uyarım geriliminin kaç volt olacağı motor etiketi üzerinde belirtilir.

### DA Motorlarının Sargı Yapıları ve Özellikleri

Çalışma prensipleri aynı olmakla birlikte farklı yapıda doğru akım motorları da kullanılmaktadır. Bu motorları birbirlerinden ayıran en büyük fark endüktör sargılarının yapılarında görülür. İşletmelerde üç farklı yapıda doğru akım motoru kullanılmaktadır. Bunlar; şönt, seri ve kompunt motorlardır.



(Şekil 2.6 DC motor yapısı)

### **Şönt Motor Endüktör Yapısı**

Uyartım sargısının endüvi sargısına paralel olarak bağlandığı doğru akım motorlarıdır. Şönt motorun devir sayısı, yük ile çok fazla değişmemektedir. Motorun devir sayısı, kaynak gerilimi veya endüvi akımı ile kontrol edilebilir. Yol alma anındaki momentleri düşüktür. Motor boşta çalışırken de devir sayısı normal değerindedir. Motorun maksimum verimde çalışması için motorun sabit kayıplarının endüvi kayıplarına eşit olması gerekir. Motorun üreteceği moment, endüvi akımıyla doğru orantılı olarak artar. Yüksek kalkınma momenti ve sabit devir sayısı istenen uygulamalarda kullanılır. Vantilatör, aspiratör ve tulumbalar, kâğıt fabrikaları, dokuma tezgâhları, gemi pervaneleri, matbaa makineleri ve asansörler bu motorun kullanım alanlarındandır.



(Şekil 2.7 Sargı yapısı)

### **Seri Motor Endüktör Yapısı**

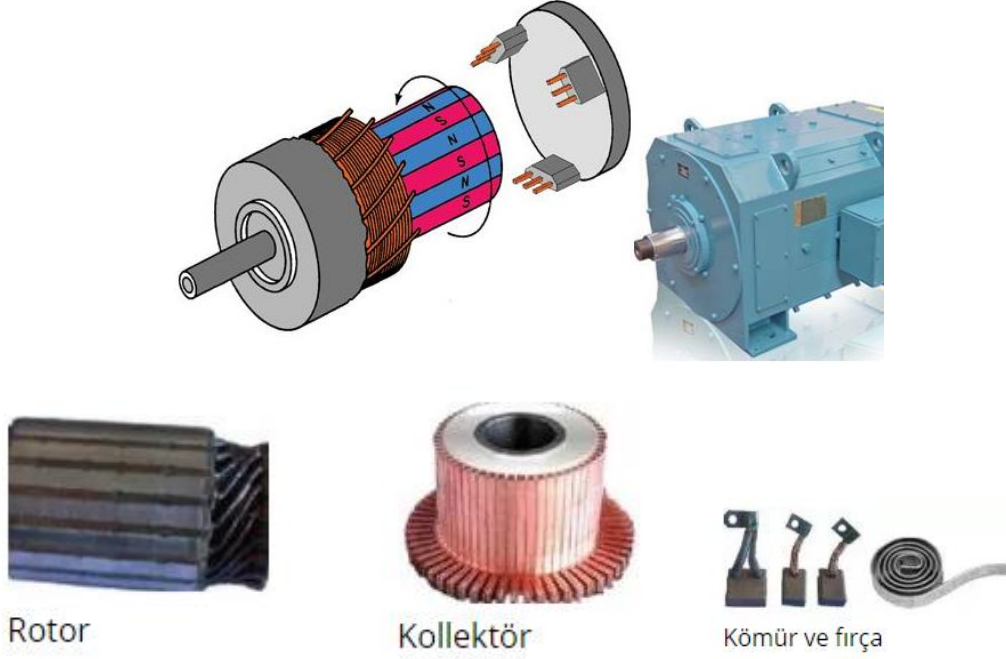
Uyartım sargısı endüvi sargısının birbirine seri olarak bağlandığı doğru akım motoru çeşididir. Motor, yüklendikçe devir sayısı hızla düşer. Bunun nedeni, yük akımının aynı zamanda uyartım akımı olmasıdır. Seri motorun yol alma momenti oldukça yüksektir. Motor boşta çalıştığında moment değeri oldukça küçük bir değer alır. Dolayısıyla devir sayısı, tehlikeli bir şekilde yükselebilir. Bu yüzden seri motor boşta çalıştırılmamalıdır.



(Şekil 2.8 DC motor endüktör yapısı )

## Kompunt Motor

Kompunt motor da endüktör sargısı olarak şönt ve seri olmak üzere iki tip sargı bulunur. Kompunt motor, istenirse şönt veya seri motor olarak da çalışabilir. Ancak şönt sargı endüviye paralel, seri sargı endüviye seri bağlandığında kompunt bağlantı yapılır.



(Şekil 2.9 DC motor parçaları)

### 2.1.3 Motor Sürücüleri

Motor sürücü devreleri güç elektroniği devrelerinin en önemli uygulama alanlarından biridir. Amaçları hız, pozisyon ya da moment kontrolü yapmaktır. Proses motor sürücü tasarımını belirleyen en önemli faktördür. Motor sürücü sistemlerinin dışında prosesi kontrol eden bir dış çevrim bulunmaktadır. Bu dış çevrimin zaman sabiti çok büyüktür. Bu yüzden motor sürücü sisteminin hassasiyeti ve hızlılığı çok önemli değildir. Motor sürücü sistemleri motor ve onu kontrol eden güç elektroniği dönüştürücüsünden oluşur. Motor: DC, asenkron ve senkron motorlar olabilmektedir. Bunun dışında motor miline bağlı bir hız ve pozisyon sensörleri bağlanır. Güç elektroniği devresi ise ana güç katı ve yardımcı elektronik elamanlardan oluşmaktadır. Motor sürücü sistem elemanlarının seçimindeki kriterler şunlardır:

- Motor ile yük arasındaki uyumluluk
- Motor seçimindeki termal konular

- Motor ile güç elektroniği konverteri arasındaki uyumluluk
- Yarı iletken güç elemanlarının akım değeri
- Nominal gerilim değeri



(Şekil 2.10 Mikrodenetleyici modeli)

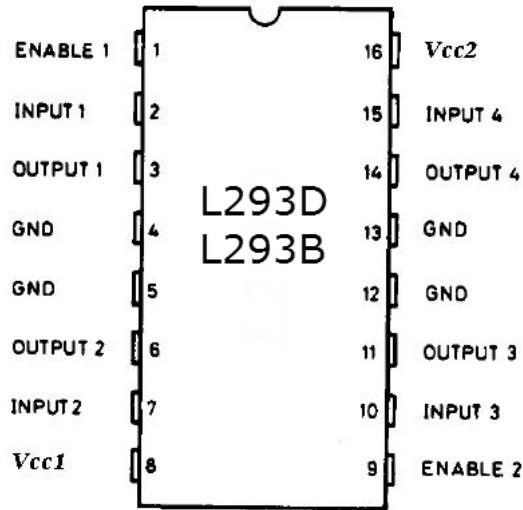
#### Giris reaktörleri

- Harmonikleri azaltırlar. Motor sürücülerin girişinde bulunan doğrultucu devreleri şebekeden doğrusal olmayan akım çekerler. Çekilen akım yüksek miktarda harmonik içerir. Motor sürücünün girişine eklenecek giriş reaktörü sürücünün çektiği akımdaki ani değişimleri ortadan kaldırarak oluşan harmonikleri azaltmaktadır.
- Sürücüyü yüksek gerilimlerden korur.
- Ani akım değişimlerini engellerler.
- Sistem bileşenlerinin ömrünü uzatırlar.

#### Çıkış reaktörleri

- Motor sürücüler üzerinden beslenen motorların çalışma sıcaklığı normal şebeke gerilimi ile beslenen motorlardan daha yüksektir. Bunun nedeni motor sürücünün çıkışında darbe genişlik modülasyonu nedeniyle yüksek frekanslı harmonikleri oluşmasıdır. Bu harmonikler motorda ek kayıplar oluşmasına ve motor çalışma sıcaklığının artmasına neden olurlar. Motor sürücünün çıkışında bir reaktör kullanılması durumunda harmonikler azalacağından motorun çalışma sıcaklığı da düşecektir. Ayrıca kayıpların azalması sistemin verimini arttıracaktır.

- Motor sürücünün PWM işaretini oluşturan evirici kısmının anahtarlama frekansı genel olarak birkaç kHz'tir. Bu anahtarlama frekansı insan kulağının duyma sınırları içerisinde. Bu nedenle motor sürücüleri üzerinden beslenen motorlarda yüksek frekansta istenmeyen sesler duyulur. Reaktörler yüksek frekans bileşenlerini önemli ölçüde ortadan kaldırdıklarından motorların daha sessiz çalışmalarını sağlarlar.
- Motor sürücü ile motor arasında bulunan kabloda veya motorda kısa devreler oluşabilir. Motor sürücülerinde genel olarak kısa devrelere karşı koruma sistemleri bulunmaktadır. Ancak bazı kısa devre durumlarında akım bir anda çok yüksek değerlere ulaşabileceğinden koruma sistemi devreye girinceye kadar motor sürücünün evirici kısmındaki yarıiletken anahtarlar zarar görebilmektedir. Reaktörler bir kısa devre anında akımın bir anda artmasına engel olacağından motor sürücüsünün kısa devre koruma sisteminin devreye girmesine zaman tanır ve sürücüyü korurlar.
- Bazı uygulamalarda motor sürücü, motorun hemen yanına yerleştirilmez. Bu durumda motor ile motor sürücü arasında uzun bir kablo kullanılır. Kablonun endüktansı ve kapasitesi nedeniyle motorun terminallerinde anlık yüksek gerilimler oluşur. Bu durum motorun ömrünü kısaltır. Motor sürücü çıkış reaktörleri bu durumu engelleyerek motorun ömrünü uzatır.



(Şekil 2.11 L293D datasheet)

### 2.1.4 Bluetooth Modülü

Bluetooth kısa mesafeli haberleşmeler için geliştirilmiş, 2,4 – 2,48 GHz ISM bandını kullanan bir haberleşme protokolüdür. Bluetooth modülleri arasındaki iletişim mesafesi eğer arada bir engel yoksa yaklaşık 20 metredir. Geliştirilen yeni teknolojiler ile beraber bu mesafe yaklaşık 100 metreye kadar arttırılmıştır. Bu yeni geliştirilen Bluetooth modülleri henüz Arduino projelerinde kullanılmamaktadır. Arduino projelerinde genellikle HC-05 veya HC-06 Bluetooth modülleri kullanılmaktadır. Biz de projelerimizde bu Bluetooth modüllerini kullanmaktayız.

HC-05 ve HC-06 Bluetooth modülleri özellik olarak hemen hemen birbirinin aynısıdır. Tek fark, HC-05 hem kendisine gelen bağlantı isteklerine cevap verirken hem de başka Bluetooth cihazlarına bağlantı isteği yollayabilmesidir. HC-06 Bluetooth modülü ise yalnızca kendisine gelen bağlantı isteklerini cevaplayabilir, başka bir Bluetooth modülüne bağlantı isteği yollayamamaktadır. Kısacası HC-05 hem yönetici hem de köle modunda çalışabilirken, HC-06 sadece köle modunda çalışabilmektedir.

HC-05 ve HC-06 Bluetooth modüllerinin ortak özellikleri aşağıda verilmiştir.

- 2,4 GHz haberleşme frekansı (ISM)
- Hassasiyet:  $\leq -80$  dBm
- Çıkış gücü:  $\leq +4$  dBm
- Asenkron hız: 2,1 MBps / 160 KBps
- Senkron hız: 1 MBps / 1 MBps
- Çalışma gerilimi: 1,8 - 3,6 V (Önerilen 3,3 V)
- Akım: 50 mA
- Kimlik doğrulama ve şifreleme

Bluetooth modülü satın alınırken dikkat edilmesi gereken noktalar vardır. Projede kullanım kolaylığı için breakout'a (kılıf) sahip Bluetooth modülü seçilmesi gerekir. Breakout kablolamada kolaylık sağlamaktadır. Projelerde kullanımı kolaylaştırmak için Arduino üzerine direkt takılabilen Bluetooth Shield'leri de bulunmaktadır.



(Şekil 2.12 Bluetooth modeli)

Bluetooth modülünün üzerinde VCC, GND, Rx ve Tx olmak üzerine 4 adet pin bulunmaktadır. Bu pinlerden VCC ve GND modülü beslemek için kullanılır. Arduino tarafından yollanan komutlar Bluetooth modülü tarafından alınabilmesi için, Arduino'nun Tx pini Bluetooth modülünün Rx ayağına takılmalıdır. Aynı şekilde Bluetooth'a gelen mesajların Arduino'ya aktarılması için, Arduino'nun Rx pini Bluetooth modülünün Tx pinine takılması gerekmektedir. Bluetooth modülü her ne kadar 3,3 volt ile beslense bile Rx ve Tx pinlerindeki gerilim Arduino tarafından 5 volt düzeyine çekilebilmektedir. Bazı Bluetooth modülleri için 3,3 volt gerilimin üstü cihaza zarar verebildiği için bu pinlerin voltaj bölücü ile devreye bağlanmalıdır.

### 2.1.5 Arduino Mikrodenetleyici

Arduino bir giriş çıkış veya processing/wiring dilinin bir uygulamasını içeren geliştirme kartıdır. İtalyan Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino ve David Mellis bu işin temellerini atmışlardır. Açık kaynak kodlu geliştirilen ve isteyen herkesin baskı devrelerini indirerek kendi devrelerini basabilecekleri, dilerlerse şık bir görüntüye sahip hazır basılmış ve bileşenleri yerleştirilmiş halde alabilecekleri, esnek, kolay kullanımlı donanım ve yazılım tabanlı bir fiziksel programlama denetleyicisidir.

Arduino tek başına çalışan interaktif nesneler geliştirmek için kullanılabileceği gibi bilgisayar üzerinde çalışan yazılımlara da bağlanabilir. Arduino geliştirme kartı üzerindeki mikroişlemci (AtmegaXX), USB portu, analog ve sayısal giriş/çıkış birimleri bulunur.

Programlama için C/C++ dili kullanılır ve Java üzerinde geliştirilmiş geliştirme, derleme, yükleme özelliklerine sahip Arduino IDE ile programlanır. Donanım özellikleri arduino çeşidine göre farklılıklar gösterse de temel olarak bir arduino şu özellikleri taşımaktadır;

- ATmega8, ATmega168, ATmega328 mikroişlemci
- 5 voltluk regüle entegresi
- 16MHz kristal osilator ya da seramik rezonatör
- Flash memory
- SRAM
- EEPROM

### **Arduino İle İlgili Temel Kavramlar**

**Giriş Voltajı:** Kart için önerilen voltaj aralığıdır. Kart maksimum voltaj aralığından az daha fazla voltajla da beslenebilir. Aklımızda bulunması gereken püf nokta ise Li-Po piller 3.7V desteklemektedir. Piyasadaki Arduino çeşitleri de bu voltaj değerini desteklemektedir. Dolayısıyla Arduinolar 3.7V Li-Po pillerle doğrudan beslenebilir.

**Mikroişlemci:** Mikroişlemci, bir Arduino'nun beynidir. Arduino geliştirme kartı çeşitli AVR mikroişlemciye dayalıdır, hepsinin de kendine has fonksiyonları ve özellikleri bulunmaktadır.

**Sistem Voltajı:** Mikroişlemcinin çalıştığı voltajdır. Bu kartın uyumluluğu için özellikle 5V'tan 3.3V'a geçildiğinde önemli bir faktördür. Bağlanmak istenilen dış sistem ne olursa olsun her zaman elimizde bulunan işlemcinin voltaj seviyesiyle eşleştirme gereği duyulacaktır.

**Saat Hızı:** Mikroişlemcinin hıza bağlı frekans aralığıdır ve komutları çalıştırma hızıyla bağlantılıdır. Ayrıca bazı beklentiler oluşabilir. Çoğu ATmega mikroişlemci 3V'ta iken saat hızı 8MHz kaldırırken çoğu 5V'ta 16MHz'de çalışmaktadır.

**Dijital I/O:** Arduinodaki dijital Giriş/Çıkış'ların sayısıdır. Bunların her biri giriş ya da çıkış olarak bazısı da PWM olabilecek şekilde tasarlanmıştır.



**Analog Giriş:** Arduinodaki analog girişlerin toplam sayısıdır. Analog pinler ‘A’ harfi ve yanındaki numaralarla isimlendirilir ve ATmega yonga içindeki Analog to Digital Converter(ADC) aracılığıyla analog değerleri okumayı sağlar. Analog girişler eğer ihtiyaç olursa daha fazla Giriş/Çıkış olarak kullanılabilir.

**PWM:** PWM sinyal üretebilme kapasitesi olan dijital Giriş/Çıkış’ların sayısıdır. PWM sinyalleri analog çıkışlar gibidir. Arduino’yu analog voltajını 0 ve sistem voltajı arasında değişme imkânı tanır.

**UART:** Arduino’nun desteklediği, birbirinden ayrı seri bağlantı çizgileri sayısıdır. Çoğu Arduino kartta dijital Giriş/Çıkış pinleri 0 ve 1 serilerin 2 katıdır ve alınan pinler seri programlama portuyla paylaşılmıştır. Bazı Arduino kartları birden fazla UART’a sahiptir ve birden fazla seri portu bir kerede destekleyebilir. Bütün arduino kartlar programlama için en az bir UART’a sahiptirler. Fakat bazılarında pinlere erişilebilir değildir.

**Flash Memory:** Taslakların depolanabileceği maksimum mevcut hafızadır. Bütün bu hafızanın tamamı kullanılabilir değildir, zira küçük bir kısmı bootloader tarafından kullanılmaktadır.

**Bootloader:** Eğer mikroişlemci Arduino’nun beyni olarak nitelendirilirse, bootloader kişiliği olarak değerlendirilebilir. Bootloader ATmega’nın içinde bulunur ve seri port aracılığıyla donanım programlaması için yükleme yapılmasını sağlar. Çünkü farklı Arduino kartları farklı mikroişlemciler ve programlama arayüzleri kullanmaktadır. Her birinde farklı bootloader programı vardır. Bootloader için olan kaynak kodları, Arduino dağıtımı içinde bulunabilir. Bütün Arduino bootloaderlar Arduino IDE yazılımından kod yüklenmesini sağlar.

## Arduino IDE



(Şekil 2.13 Arduino programlama arayüzü)

Arduino kullanımı ve programlaması kolay bir platform olarak tanınmaktadır. Bu özelliğini kullanımı kolay bir IDE ve C++ dilinde programlanabilir olmasına borçludur. Arduino açık kaynak kodludur ve birçok geliştirme kütüphanesi ücretsiz olarak kullanılabilir. Temel işlemler için ihtiyaç duyulacak birçok kütüphane Arduino projesine dâhil edilmiştir. Arduino IDE Windows, Linux, Macintosh OSX işletim sistemlerine destek vermektedir. Programlama için C++ dilinin sahip olduğu nesneye dayalı programlama avantajlarına sahiptir. Arduino programlanırken kullanıcıdan iki metodu gerçekleştirmesi beklenmektedir. Bu metotlardan birisi cihaz çalışmaya başladığında sadece bir kez çalıştırılır. Bu metodun imzası “void setup()” olarak belirlenmiştir ve tanımlama, ilk değer atama gibi işlemler bu metod içinde yapılır. Gerçekleşmesi gereken diğeri ise “void loop()” metodudur. Bu metod içinde kullanıcı cihaz çalıştığı sürece yapılmasını istediğini işlemleri cihaz çalıştığı süre boyunca bu metod çağırılır. Kullanıcı iki metodu gerçekleştikten sonra kod dosyasını Arduino IDE ile derleyip, cihaza yükleyebilmektedir.

## Arduino Uno

- USB seri iletişim dönüştürücü olarak FTDI entegre kullanılmıyor ama dönüştürme işini ATmega8U2 yapıyor. Bu durum, daha hızlı aktarım olarak sağlıyor. Yani aslında dönüştürücü entegresi kullanıldığında herhangi bir USB bağlantılı cihaz olarak algılanırken, bu dönüştürücünün işinin yazılımla halledilmiş olması, takılan

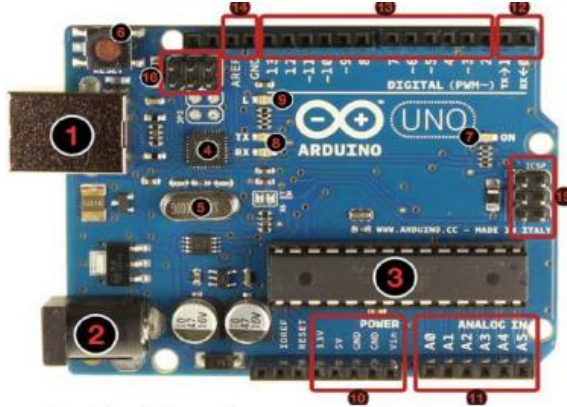
bilgisayar tarafından, tek para bir USB tak kullan aygıtı olarak algılanmasını saęlıyor.

Linux ve Max için driver yüklemek gerekmezken Windows için “inf” dosyası gerekiyor. FTDI chip’i kullanılmamış olmasının verdiği başka bir özellik de Arduino Uno’yu sisteme klavye, mouse, joystick vs. olarak tanıtabilme ve sistemde bu şekilde kullanabilme kolaylığıdır.

- 14 adet dijital giriş / çıkış bu çıkışlardan 6 tanesi PWM çıkışı olarak kullanılır. Bunun ile beraber 6 adet analog giriş bulunmaktadır.
- Çok popüler olmasından dolayı Arduino Uno ile yapılmış proje örnekleri çokça bulunabilmektedir.
- Üzerinde hem 3.3V hem de 5V çıkışı mevcuttur.
- Haberleşme ve güç için hem USB portu hem de DC adaptör girişı bulunmaktadır.
- Kolay bağlantı için tüm çıkışlar dişı pin header şeklindedir.

### **Arduino İle Yapılabilecekler**

- Kolay bir şekilde çevresiyle etkileşime girebilen sistemler tasarlanabilir.
- Açık kaynaklı bir geliştirme platformudur. Bu yüzden kullanıcı istedięi şekilde geliştirebilir, eklemeler çıkarmalar yapabilir.
- Arduino kütüphaneleri ile mikrodenetleyiciler kolaylıkla programlanabilir.
- Analog ve dijital girişleri sayesinde analog ve dijital veriler işlenebilir.
- Sensörlerle çalışabilir olduğundan sensörlerden gelen veriler kullanılabilir.
- Dış dünyaya çıktılar (ses, ışık, hareket vs.) üretilebilir.
- Kart ile robotik ve elektronik uygulamaları kolayca gerçekleştirilebilir.



Yukarıdaki şekilde görülen ATmega328 mikrodnetleyicisi kullanan Arduino Uno Kartı çeşididir. 14 dijital giriş/çıkış pini bulunur, bunlardan 6'sı PWM çıkışı olarak kullanılabilir ve dijital dalga genişliği ayarlanarak analog değerler elde edilebilir. 6 analog giriş pinine sahiptir. 16 MHz kristal osilatörü, USB bağlantısı, 2.1mm güç girişi, ICSP başlığı ve reset butonu vardır. Mikroişlemciyi destekleyecek herşeye sahiptir. Çalıştırmak için DC 7~12V güç kaynağına bağlamak yeterlidir

- 1 : USB jakı
- 2 : Power jakı (7-12 V DC)
- 3 : Mikrodnetleyici ATmega328
- 4 : Haberleşme çipi
- 5 : 16 MHz kristal
- 6 : Reset butonu
- 7 : Power ledi
- 8 : TX / NX ledleri
- 9 : Led
- 10 : Power pinleri
- 11 : Analog girişleri
- 12 : TX / RX pinleri
- 13 : Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)
- 14 : Ground ve AREF pinleri
- 15 : ATmega328 için ICSP
- 16 : USB arayüzü için ICSP

İşlemci Özellikleri:  
Flash: 32KB (0.5KB bootloader için kullanılır)  
SRAM: 2KB  
EEPROM: 1KB  
Saat frekansı: 16MHz

(Şekil 2.14 Arduino Uno pinleri )

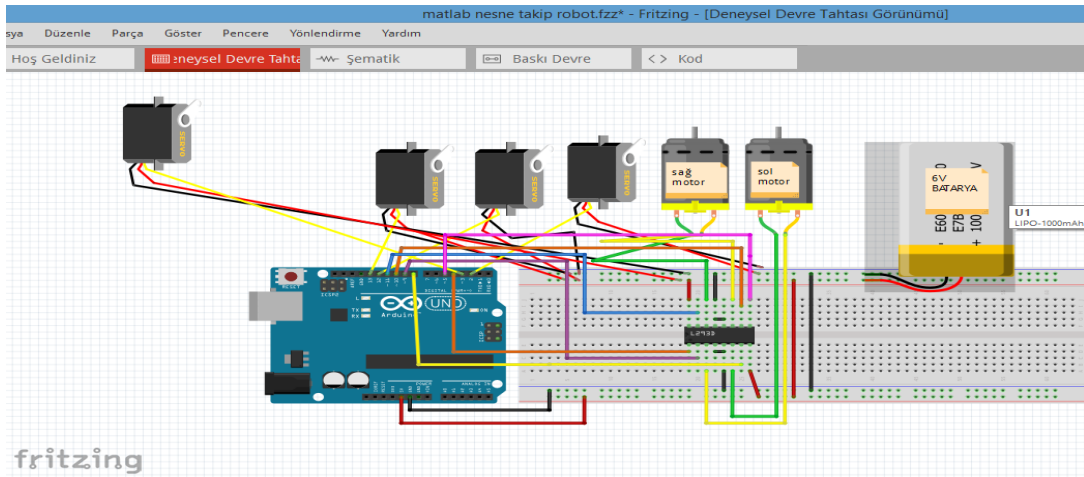
### 2.1.6 Devre Şeması

Teknik elemanlar hızlı sanayileşmenin, ekonomik, sosyal ve kültürel kalkınmanın en önemli faktörüdür. Hızlı ve sürekli üretim, yeterli montaj, işletmeye alma, bakım ve onarım teknik elemanlarının aynı dili kullanmaları ile sağlanır. Yapılan işin istenen özelliklerde olması teknik elemanların elektrik devre şemalarını eksiksiz okuma ve bunu birebir uygulamalarıyla sağlanır. Bundan dolayı soğutma, iklimlendirme ve elektrik devre elemanlarının sembollerinin çizimlerinde ülkelerin standart sembolleri gösterilir. Bu sembollerin birbirlerine birleştirilmesinin sonucunda elektrik devre şemaları elde edilir.

Çevremizdeki elektronik cihazların her yeni gün, insan ihtiyacına daha hızlı ve daha kolay yanıt verecek modelleri tasarlanmaktadır. Tasarlanan her yeni model boyutları ve işlevleri bakımından bir önceki modellerine göre daha üstün özelliklere sahiptir. Bu cihazlarda kullanılan malzeme teknolojisinin sürekli gelişmesi bu yenilenmeyi hızlandırmaktadır. Elektronik teknolojisinde yaşanan gelişmelerle beraber elektrik devreleri, elektrik sinyallerinin işleme özeline göre analog ve sayısal sistemler olarak ayrılmaktadır. Zaman eksenine göre sonsuz sayıda değer aralığına sahip analog elektrik sinyallerinin her anında tepki gösterebilen devre elemanları “Analog Devre Elemanları” olarak adlandırılabilir. Analog devre elemanları direnç, kondansatör, bobin, diyot ve transistör gibi yarı iletkenlerden meydana gelmektedir.

Çeşitli analog ve sayısal elektronik sistemlerde farklı boyutlarda ve elektriksel özelliklerde karşımıza çıkan elektrik elektronik alanındaki bu devre elemanlarını tanımak, ihtiyaca uygun malzeme seçim yapmak ve kullanabilmektir. Açık kaynak kodlu bir devre tasarım programı olan Fritzing; Arduino ve diğer elektronik tabanlı projelerde prototipten üretime geçerken üretim maliyetlerini azaltmayı ve çıkabilecek fiziksel sorunları en aza indirmeyi amaçlayan bir programdır. Projeleriniz üretime geçmeden önce prototip hazırlamakta kullanılmaktadır. Fritzing Linux, Windows, Mac OS X işletim sistemlerinde çalışabilir.

Programın arayüzü kullanmayı çok basit kılan bir tasarıma sahiptir. Diğer elektronik programlarda olduğu gibi sürükle-bırak mantığı ile malzemeleri proje alanına yerleştiriyorsunuz ve bir breadboard üzerinde kablo yardımıyla malzemeleri birbirine bağlıyorsunuz. Eğer hatalı bir bağlantı yaparsanız program sizi renklerle uarmaktadır. Bunun yanı sıra isterseniz “Breadboard-Schematic-PCB” menüsü ile 3 tür proje görüntüsü arasında geçiş yapabiliyorsunuz. Yani program size şema ve baskı devre çıkarmanız için de yardımcı olmaktadır. Tabi ki profesyonel bir şekilde çıkartmamaktadır ama siz programın devreyi düzenlemesinde biraz yardımcı olursanız program güzel seçenekler sunabilmektedir. Tasarım sonunda PDF’e aktararak daha sonra herhangi bir yazıcıdan asetata, kâğıda vs. çıktı almanızı mümkün kılmaktadır. Programın en büyük eksiği simülasyon özelliğine sahip olmaması, ancak sağladığı Arduino , shield ve modül kütüphanesi desteği programı bu alanda çalışma yapanlar için oldukça kullanışlı hale getirmektedir. Simülasyon oluşturma yönünde de çalışmaların olduğu ve yakın zamanda yeni modellerinde bu özelliğinin de kullanıcılara sunulabileceği söylenmektedir.



(Şekil 2.15 Fritzing örneği)

### 2.1.7 Lipo Piller

Lityum-iyon polimer pil olarak adlandırılmaktadır. Lityum-iyon pilin daha gelişmiş bir tasarımıdır ve aynı karakteristikleri taşımaktadır. Bu tür pillerde hafıza etkisi yoktur ve kullanılmadıkları zamanlardaki enerji kayıpları yavaştır. Uygunsuz kullanımları halinde tehlikeli olabilirler. Eğer gerekli önlemler alınmaz ise diğer pil türlerine göre ömürleri daha kısa olabilmektedir.



(Şekil 2.16 Lipo pil modeli)

### Kullanım Alanları

Lityum-iyon pil çeşitleri dizüstü bilgisayarlar, cep telefonlar, müzik çalarlar ve daha birçok taşınabilir dijital cihazlarda kullanılmaktadırlar. Lipo piller küçük boyutlarına rağmen yüksek enerji gereksinimlerini karşılayabildikleri için RC kontrollü elektrikli modellerde kullanılmaktadırlar. Ayrıca Boeing tipi yolcu uçaklarında dahi bu tarz piller kullanılmaktadır.

### Lipo Pillerin Özellikleri

Lipo piller hücrelerden oluşmaktadırlar. Tek bir hücrenin boş hali 3V dolu hali ise 4.2V olmalıdır. Lipo pillerin hücreleri seri(S) veya paralel(P) bağlı olabilir. Bu piller bağlantı şekline göre adlandırılırlar. Örneğin; 2 hücre seri bağlı ise 2S, 3 hücre seri bağlı ise 3S, 2 hücre paralel bağlı ise 2P, 3 hücre seri bağlı ve bunlara paralel bağlı bir takım daha var ise 3S2P olarak adlandırılırlar. Bunları daha detaylı açıklayacak olursak seri bağlı

demek voltaj değerimizin arttığı anlamındadır. Paralel bağlı da aynı voltaj değerinde verilen akımın arttığını göstermektedir.

Örneğin; aynı özelliklere sahip hücreler ayrıca Lipo piller üzerinde kapasite olarak adlandırılan C değeri bulunmaktadır. Özellikleri 2200mAh 3S1P 20C bir pilin üzerinden giderek teorik olarak anlatırsak;

- Bu pil sabit 2,2 amper akım çeken bir sisteme bağlandığında yaklaşık olarak 1 saat kullanılabilir. Yani başlangıçtaki akım değeri olan 2,2A 1 saatte durmaksızın verebileceği akım değeridir.
- $2,2A \times 20 C = 44A$  bu pil en fazla 44A akım verebilmektedir. Buradaki C değeri ne kadar büyük ise pilin kapasitesi yani verebileceği akım değeri o kadar büyüktür..
- Lipo pillerde 'burst C' olarak adlandırılan değer ise pilin aşırı yüklenmede dayanabileceği anlık akımı göstermektedir.

### **Lipo Pillerde Kullanımında Dikkat Edilmesi Gerekenler**

- Lipo piller kesinlikle kapasite değerlerinin üzerinde kullanılmamalıdır. Örneğin bir quadrotor da havada ne kadar süre kalınmak istendiği hesaplanıp buna uygun bir motor seçilmelidir. Lipo pillerden verebilecekleri max akımdan daha fazlası istendiğinde yangın ve benzeri olaylar ile karşılaşılabilir.
- Lipo pillerin uç kabloları kesinlikle kısa devre edilmemelidir.
- Lipo piller için özel olarak üretilmiş balanslı şarj aletlerini kullanarak şarj edilmelidir.
- Lipo piller kullanılırken hücre başına gerilim değeri 3V un altına düştüğünde hücresi ölebilir ve bir daha kullanılamamaktadır. RC modellerde programlı ESC'ler bu durumu engellemektedir.

### **Lipo Pillerin Saklama Koşulları:**

- Lipo piller hücre başına yarı gerilimle (3.6V-3.75V) saklanmalıdır.
- Lipo piller serin kuru ve ışıktan uzak bir ortamda saklanmalıdır.

- Lipo pilleri saklamak için özel saklama kapları da mevcuttur.

## **2.2 Görüntü İşlemede Kullanılan Malzemelerin Tanıtımı**

### **2.2.1 Kamera**

Fotoğraf makinesinde çekilen resimleri art arda hareket haline alınması fikriyle oluşturulmuş makinedir. Kamera ile çekilmiş filmler, film gösterme makinasıyla aynı hızla perdeye aksettirildiğinden, filmi hareketli olarak takip etmek mümkün olmaktadır.

Kamerada, fotoğraf makinasındaki gibi bir karanlık oda, ışığı toplayıp görüntüyü sağlayan mercekler (objektif) ve ışığa dayalı film bulunur. Kamerada uzun şeritler halinde sarılmış filmler kullanılır. Filmin kenarları rahat akışı sağlamak için delinmiştir. Her kamerada temel olarak şu parçalar bulunur:

#### **Mercek**

Görüntüyü küçülterek filme ileten kısımdır. Gelişmiş kameralarda birden fazla mercek bulunur. Bu mercek topluluğuna objektif denir. Kameranın değeri objektifin niteliği ile ölçülür. Her objektifin belli bir odak uzaklığı vardır ve mm olarak verilir. Kutu tipi dışındaki kameralarda objektif ayarlanabilir merceklerden meydana gelir. Mercekleri ayarlamaktan maksat, değişik uzaklıktaki görüntüleri film üzerine net olarak aktarabilmektir.

#### **Diyafram**

Kameranin film üzerine düşen ışık miktarını ayarlayan kısımdır. Işığın geçebileceği alanı daraltıp genişleterek ışık miktarını ayarlar. Film üzerinde net bir görüntü meydana gelmesi için film üzerine ne çok, ne az, ihtiyaç kadar ışık düşmesi gerekir.

Fazla ışık filmi yakar, azı ise görüntü ortaya çıkmaz. Diyafram açığını kendi ayarlayanlar dışındaki kameralarda çeşitli miktardaki ışık için ayar ölçüleri konmuştur.

#### **Obtüratör**

Işığın makinaya çok kısa bir süre için girmesini sağlayan ve bu süreyi ayarlayan bir mekanizmadır. Çeşitli şekilleri vardır. Işığın filmin üzerine yeteri kadar düşmesinden



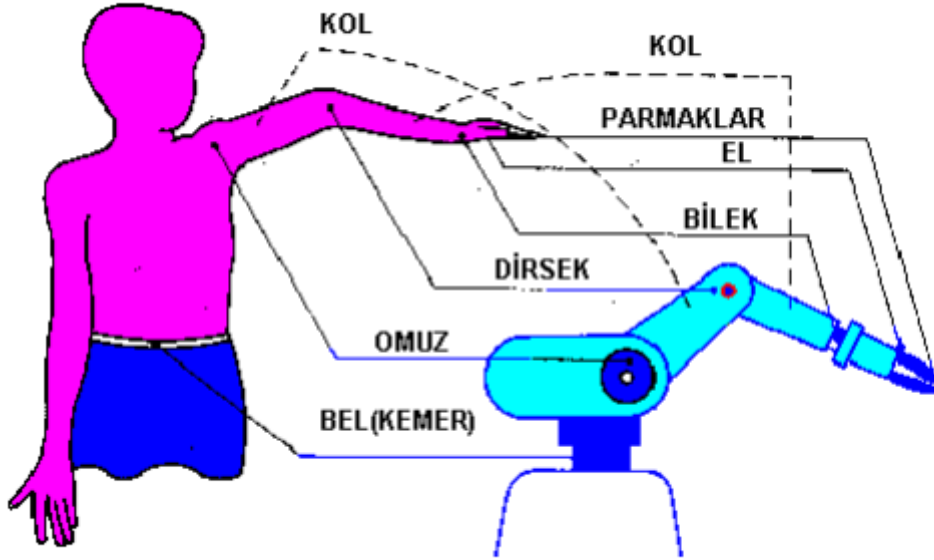
başka görüntünün hareketli olup olmamasına bağlı olarak yeterli süre düşmesi de istenir. Hareket eden cisimlerde hareketin hızı ile doğru orantılı olarak obtüratörün de hızı artar. Hareketli görüntülerde konum devamlı değiştiğinden film üzerinde net bir görüntü meydana getirebilmek için çok kısa bir an için obtüratörün açılıp kapanması lazımdır. Bunu da obtüratörün mekanizması sağlar. Kamerada, kullanılmış ve kullanılmamış filmlerin toplandığı kasetler de mevcuttur. Filmin kamera içindeki hareketini dişli tamburlar sağlar. Dişli tamburlar kamera içindeki küçük bir motorla döndürülür. Cihazın mekanizması filmin kullanılmamış bir bölümünün objektif kapağı önünde bir süre durmasını sağlar. Bu sırada kapak açılır ve görüntü tespit edilir. Sonra çekilen film yerini kullanılmamış filme bırakır. Bu işlem sürekli olarak ve saniyede 24 defa çekecek şekilde devam eder. Kameraların çeşitlerine göre hızı saniyede 1 ile 100-150 defaya kadar değişebilir. Fakat standart olarak saniyede 24 defa kabul edilmiştir.

### **Orbiratör**

Örtücü, yarım daire kelebek kanadı veya eski kameralarda olduğu gibi silindir biçimli de olabilir. Günümüzde kullanılan örtücüler çoğunlukla kelebek kanadı veya yarım daire biçimindedir. Örtücünün görevi dişli tamburun hareketine uygun olarak objektifin önünü açıp kapamaktır. Yarım daire biçiminde olanlar her turda, kelebek kanadı biçiminde olan ise yarım turda bir görüntü kaydederler. Bir örtücüde açık olan kısım ne kadar büyükse açık kalma süresi o kadar fazladır. Normal kameralarda açıklık 130°-180° arasında değişir. Bazı kameralarda ise 230°'ye kadar örtücüler bulunmaktadır.

### 3 ROBOT KOL

#### 3.1 Robot Kol Tanıtımı



(Şekil 3.1 Robot kol benzetimi)

#### Robot Kol ile insan benzetimi

Tanımdaki terimlerin detaylı olarak açıklamaları aşağıdaki gibidir:

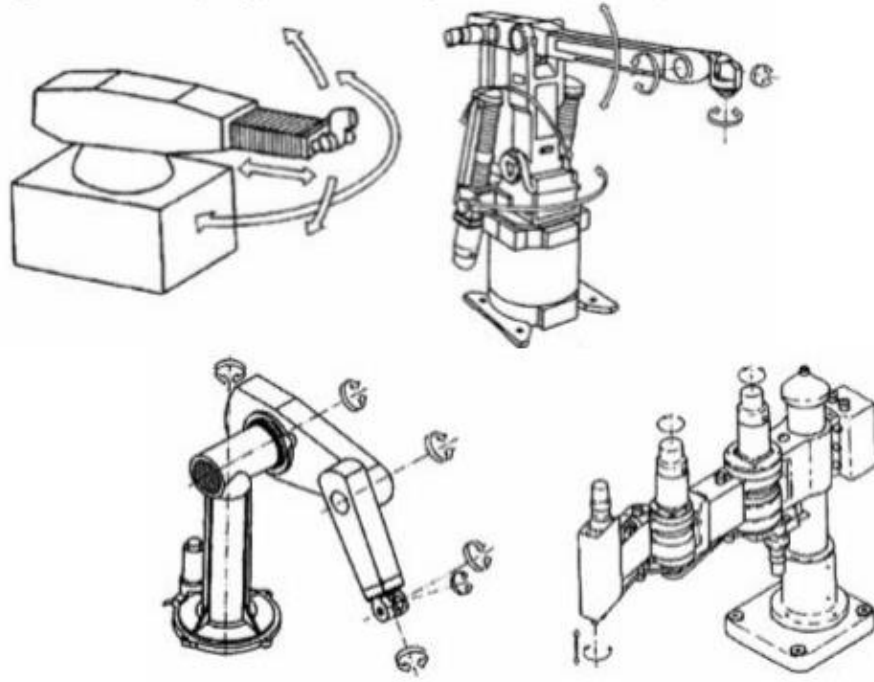
- **Manipülâtör:** Robotun hareketli kısımlarına denir.
- **Yeniden programlanabilir:** Fiziksel değişiklikler olmadan programlanmış hareketleri veya yardımcı fonksiyonları değiştirilebilen.
- **Çok amaçlı:** Fiziksel değişikliklerle farklı bir uygulamaya adapte edilebilme yeteneği.
- **Fiziksel değişiklikler:** Programlama kasetleri, ROM'lar vb. gibi değişiklikler hariç mekanik yapının veya kontrol sisteminin değiştirilmesi.
- **Eksen:** Lineer veya dönel modda robot hareketini belirtmek için kullanılan yön.
- **Tamlık:** Tamlık, çalışma hacmi içinde istenen bir noktaya, robotun bilek sonunu götürebilme yeteneğidir. Uzaysal çözülüm, robotun çalışma hacmini bölebileceği en küçük hareket artışıdır. Robotun tamlığı uzaysal çözülüm cinsinden ifade edilebilir çünkü hedef bir noktaya ulaşabilme yeteneği her eklem hareketi için

robotun kontrol artımlarını ne derece tanımlayabildiğine bağlıdır. Tamlik, verilen bir hedef noktaya erişebilmek için robotun programlanabilme kapasitesiyle ilişkilidir.

- **Tekrarlanabilirlik:** Tekrarlanabilirlik, uzayda robota önceden öğretilen bir noktaya robotun, bileğini veya bileğine eklenen end efektörünü götürebilme yeteneğidir. Robotun öğretilen bir noktaya göre tekrarlanan hareketlerinin sonucunda, robot uç noktası ile öğretilen nokta arasında oluşabilecek maksimum hata miktarıdır. Genel amaçlı robotlarda tekrarlanabilirlik değerinin 0,1 mm ila 0,2 mm olması yeterli olabilmektedir. Özel olarak ark kaynağı uygulaması düşünülürse tekrarlanabilirlik değerinin kaynakta kullanılacak tel çapının yarısından küçük olması istenir.
- **Yük taşıma kabiliyeti ve hız:** Maksimum yük taşıma kapasitesi, robotun minimum hızında tekrarlanabilirlik değerini koruyarak taşıyabileceği maksimum yük değeridir. Nominal yük taşıma kapasitesi de robotun maksimum hızda tekrarlanabilirlik değerini koruyarak taşıyabileceği maksimum yük miktarıdır. Bu yük taşıma kapasitesi değerleri taşınan malzemenin boyut ve şekline bağlıdır. Bu tanımlamayla robot bir otomatik makinedir ve belki de birçok tartışmanın çıktığı alandır. Robot olarak kabul edilebilecek birçok makine, çevre ne kadar sınırlı olursa olsun, çevreden alınan bilgiye cevap verebilmelidir. Robot, cevabı yorumlayacak ve gereken değişikliği yapacaktır.

Robotların daha önceki otomatik makinelerden **farkları:**

- Mafsal hareketleri çok daha fazladır.
- Daha hızlı iş görür.
- Vidaları yanlış yer e monte etme gibi herhangi bir yanlış harekette bulunulmaz.
- Kendi işlemlerinin doğruluğunu kontrol edebilir.
- Gerektiğinde ana sisteme bağlı olmaksızın kendi bölümlerine ilişkin çalışmaları kendi başlarına sürdürebilir. İmal edilen parçaları, kendi kendine tasnif edip yükler. Parçayı, ait olduğu bölümlerine götürerek yerleştirebilir.



(Şekil 3.2 Sanayi robotları)

Robot kelimesi Slavca'daki zorunlu iş, angarya ve işçi kelimelerinden türetilmiş bir kelimedir. İlk olarak eski Çekoslovakya'lı yazar Karel Capek tarafından yazılan oyun "Rossum'un Evrensel Robotları" (1921) isimli oyununda kullanılmıştır. Teknolojik anlamda günümüz teknolojisine en yakın robot ise 1960lar'ın ortalarında George Devol tarafından geliştirilmiştir. Bu ilk bilgisayar kontrollü mekanik kol daha önceleri 1940'lı yıllarda güvenlik camının arkasında bir insan tarafından kontrol edilerek radyoaktif malzemeleri taşıyan ilk atasından esinlenerek geliştirilmiş olup, kontrol insandan alınıp bilgisayara devredilmiştir. 1970ler'in başlarında Devol'un ortağı Engleberger'in Japonya'ya yaptığı ziyaret sonrasında o yıllarda yoğun şekilde üretim tesisleri kuran Japon'lar ilk olarak bu keşfin endüstriyel geleceğini kavrayarak lisanslı olarak üretilmeye başlamış ve sonrasında tüm Japon robot firmalarının önünü açmıştır. Başlangıçta yeniden programlamanın zor olması ve parça konumu tam olarak bilinmediğinde insanlarla rekabet edemeyen robotlar sadece sprej boyama ve kaynak işlerinde kullanılmışlardır. Scara robotlarının ortaya çıkmasıyla elektronik endüstrisinde baskılı devre kartlarına bileşenlerinin montajı işi robotların hâkimiyetine girmiştir.

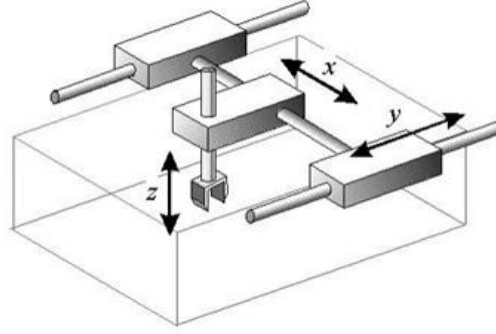
Etrafında olup bitenleri programında belirtilen sınırlar dâhilinde algılayıp komutları yerine getiren robotlar karar verme yeteneğinden yoksundurlar. Bu yüzden günümüzde artık endüstrinin birçok alanında faaliyet gösteren robotlarla ilgili başlıca araştırma

konuları, robotun kısmen kendi kararlarını verip uygulamaya koymasını sağlayacak olan “yapay zekâ” ve bu kararları vermesi yolunda algı sınırlarını genişletecek olan görüntü algılama ve işleme sistemleri üzerine yoğunlaşmıştır. ISO 8373’e göre ise robot, “Endüstriyel uygulamalarda kullanılan, sabit veya hareketli olabilen, üç veya daha fazla programlanabilir eksenle sahip, otomatik kontrollü, yeniden programlanabilir çok amaçlı manipülatör” olarak tanımlanmıştır. Bu tanım da yer alan yeniden programlanabilme kavramı, fiziksel değişiklikler olmadan programlanmış hareketleri veya yardımcı fonksiyonları değiştirilebilen anlamına gelirken, çok amaçlı olması fiziksel değişikliklerle farklı uygulamalara adapte edilebilme yeteneğini tanımlamaktadır.

### **Hareket Kabiliyetlerine Göre Robot Kolların Sınıflandırılması**

Robotlar eksen sayıları, kontrol tipleri ve mekanik yapılarına göre sınıflandırılır. Endüstriyel robotların mekanik yapılarına göre sınıflandırılmaları görülmektedir. Endüstriyel robotlarda önem arz eden başlıca kriterler hassasiyet, tekrarlılık, yük taşıma kabiliyeti ve erişim alanıdır. Hassasiyet, çalışma hacmi içinde istenen bir noktaya robotun götürebilme yeteneğidir. Uzaysal çözülüm, robotun çalışma hacmini bölebileceği en küçük hareket miktarıdır. Böylece robotun hassasiyeti hedef noktaya ulaşabilmek için yaptığı eklem hareketlerinde kontrol artımlarını ne kadar küçük tanımlayabildiği şeklinde açıklanabilir. Tekrarlılık, uzayda robota önceden öğretilen bir noktaya robotun eklenen bir takımın götürülmesinde tekrarlanan hareketler sonunda öğretilen nokta ile ulaşılan nokta arasında oluşabilecek maksimum hata miktarıdır. Genel amaçlı robotlarda bu değer 0,05-0,2 mm arasında değişmektedir. Maksimum yük taşıma kapasitesi, robotun en düşük hızında tekrarlılık değerlerini koruyarak taşıyabileceği maksimum yük değeri iken, nominal yük taşıma kapasitesi de robotun maksimum hızda tekrarlılık değerlerini koruyarak taşıyabileceği maksimum yük miktarıdır. Bu yük taşıma kapasitesi değerlerinde, yükün geometrik şekli ve boyutlarıyla, ağırlık merkezinin robot kolunun uç kısmına göre pozisyonunun önemi unutulmamalıdır. Robot erişim alanı ya da çalışma zarfı, robot kolunun uç kısmının ulaşabildiği en uzak noktaların oluşturduğu hacimdir. Endüstriyel robotlar uygulama tipine göre farklılık göstermekle beraber genellikle bu bölge içinde kalan tüm noktalara ulaşabilirler.

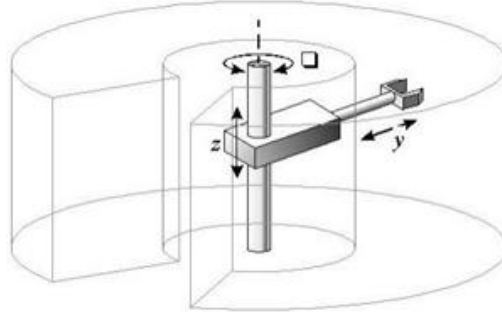
## Kartezyen Robotlar



(Şekil 3.3 Kartezyen robot kinematiği)

Sadece tutma ve taşıma yeteneği olan bu robot tipi X,Y,Z, eksenlerinde doğrusal olarak hareket etme yeteneğine sahiptirler. Basit bir yapıya sahip oldukları için hareketlerin planlanması çok kolaydır. Bu tür robotlarda; pozisyon hesaplamaları, robot uç elemanının bulunduğu pozisyon, mafsalların o anda olduğu yerde bulunduğundan çok kolaydır. Eğilme ve bükülme işlemlerini gerçekleştiremez. Çalışma alanları kare veya dikdörtgen prizma şeklindedir. Yük taşıma kapasitesi diğer robot türlerine göre daha büyüktür. İnsan gücünün taşıma kapasitesini aşan yüklerin taşınmasında kullanılır. Bu nedenle genellikle yük araçlarına, yükleme ve boşaltma işlerinde, fabrikalar da ağır yükleri taşımak amacı ile fabrikaların tavanlarına monta edilerek kullanımı yaygındır. Islak, nemli, rutubetli çalışma ortamlarında kullanılabilir. Küçük güçte olanları pnömatik olarak tahrik sistemine sahiptir. Büyük güç gereken yerlerde hidrolik tahrikli olan kartezyen robotlar kullanılır. Bunların yağ sızdırma problemleri olduğu için temizliğin önem arz ettiği ortamlarda pnömatik tahrikli olanlar tercih edilir. Hava tahrikli olan robot tipinde basınçlı hava ve havanın kontrolüne ihtiyaç olduğu için yatırım maliyetleri daha ucuz olup işletim maliyetleri de düşüktür. Büyük güçte yapılan kartezyen robotların tahrik sistemleri elektrik motorları veya hidrolik tahrik sistemleri ile sağlanmaktadır.

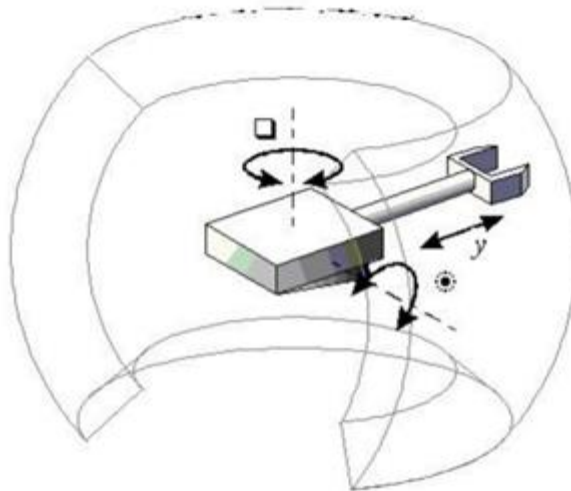
### Silindirik Robot Kolları



(Şekil 3.4 Silindirik robot kinematiği)

Silindirik robot kollar da kendi etrafında dönebilen bir mafsal ve bunun üzerinde bulunan X,Y,Z düzleminde doğrusal hareket edebilen kollardan oluşmaktadır. Esnek olmayan silindirik bir koordinat sistemine sahiptirler. Kartezyen robot kola göre hareket serbestliği daha geniştir. Çalışma alanı içindeki noktalara ulaşımı çok iyidir. Hareket kabiliyetinin az olmasından dolayı programlanması kolaydır. Robot kolun çalışma alanı silindirik koordinat sisteminde hareket eden kolların uzunluğuna bağlı olarak değişmektedir. Robotun kullanım alanı ve yük taşıma kapasitesine göre hidrolik, pnömatik veya elektrik tahrikli olarak kullanılmaktadır. Silindirik robot kollar nemli, rutubetli ve tozlu ortamlarda, deniz altı, uzay gözlem araçlarında ve nokta kaynağı işlerinde yaygın olarak kullanılmaktadır.

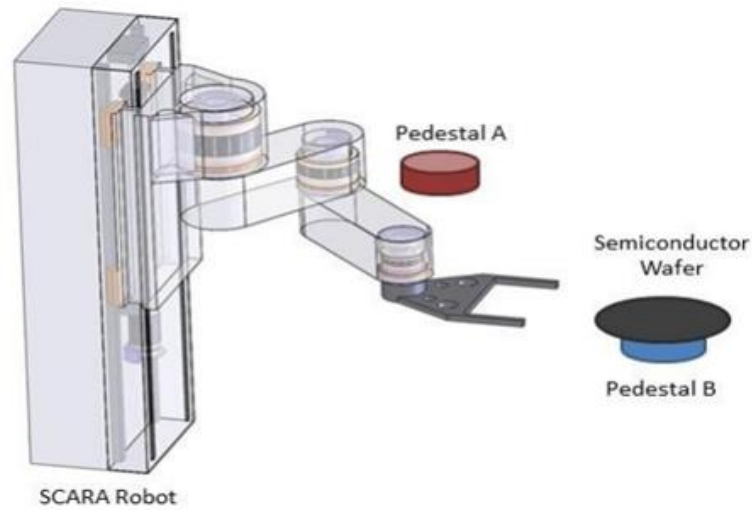
### Küresel Robot Kolları



(Şekil 3.5 Küresel robot kinematiği)

Bel, omuz ve dirsek mafsallarından oluşan bir yapıya sahiptirler. Bel ve omuz mafsalı kendi etrafında dönme hareketi yapabilirken, dirsek mafsalı kola uzama ve kısalma hareketi yaptırmaktadır. Hareket alanı olarak silindirik bir koordinat sistemine sahiptir. Kol yapılarından dolayı eklemlili robot kollarına benzemektedirler. Kinematik yapıları kartezyen ve silindirik robot kollara göre daha karmaşıktır. Çalışma şeklinin zihinde canlandırılması zor olduğu için programlama ve kontrolü de zordur. Çalışma alanının büyüklüğü kolların büyüklüğüne bağlıdır. Hidrolik tahrik sistemine sahip olan küresel robot kollar eğme, bükme işlerinde, kameralı izleme işlerinde kullanılmaktadır. Ayrıca sarkaç robot olarak ta küçük bir moment ile hareketlerini devam ettiren bu robotlar kaynak ve yapıştırma işlemlerinde kullanılmaktadır.

### Scara Robot Kolları



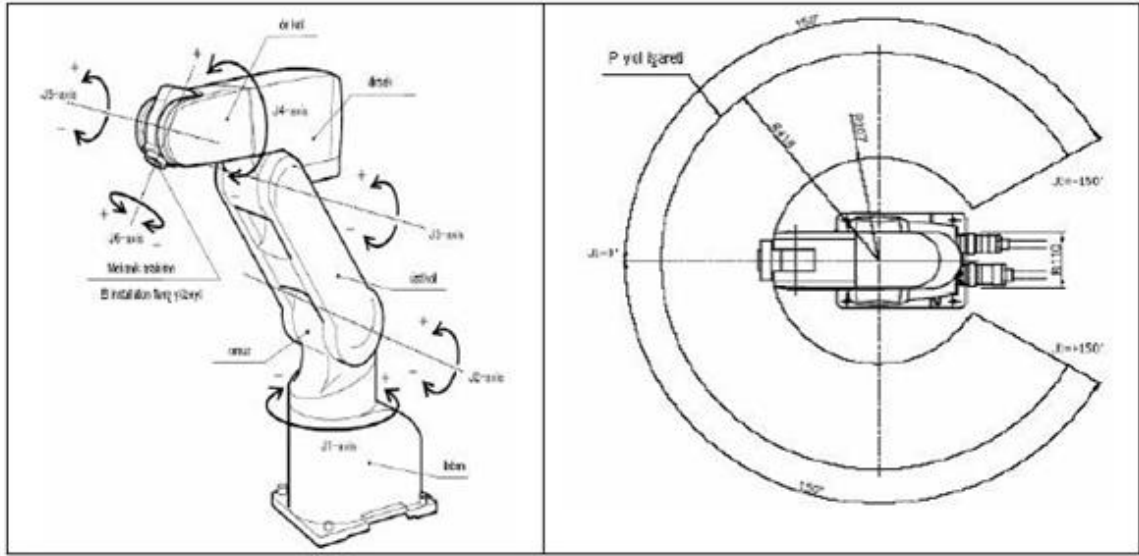
(Şekil 3.6 Scara robot kinematiği)

İki eklem yerinde elektrik motoru ve aşağı yukarı hareket edebilen pnömatik koldan oluşmuştur. Eklemlerdeki elektrik motorları eksenlerin kendi etrafında dönmesini sağlamaktadır. Tutucu ağzın bulunduğu kol pnömatik tahrikli olup Z ekseninde hareket etme kabiliyetine sahiptir. Buda robot kola esnek hareket imkânı sağlamaktadır. Hız ve konum performansı çok iyi olduğundan dolayı bu robot kol en çok elektronik sanayinde, elektronik kartlara malzemelerin montajını gerçekleştirmek için kullanılmaktadır.

Tutma ve taşıma işlerinde maliyetinin ucuz olmasından ve programlanmasının kolay olmasından dolayı şu anda sanayide en çok kullanılan robot olmuştur.



## Mafsallı Robot Kollar



(Şekil 3.7 Mafsallı robot kinematiği)

İnsan kolunun hareketlerini taklit etmeye en yakın robot koldur. Üretim sistemlerinde diğer kolların hareket kabiliyetlerinin sınırlı olmasından dolayı mafsal sayısı genellikle 5 veya 6 adet olan robot kollara ihtiyaç duyulmuştur. Bu tip robot kollarda her mafsal ayrı ayrı kontrol edilebilen servo motorlardan oluşmaktadır. Mafsallarda bulunan motorlar 12–24 V. gerilim ile beslenmektedir. Hareket esneklikleri en yüksek olan robot kollardır. Kol üzerinde bulunan her eklem X,Y,Z eksenlerinde üç boyutlu hareket yapabilmektedir. Çalışma alanı içerisinde tanımlanan bir noktaya en kısa yoldan ve kısa zamanda ulaşım imkânı tanımaktadır. Robotun hedef pozisyonlara yaklaşımı mafsal hareketi veya doğrusal X,Y,Z koordinatları doğrultusunda hareket ederek gerçekleşmektedir. Diğer robot türlerine göre karmaşık bir yapıya sahip olup, programlanması da diğerlerine göre zordur. Her mafsal program içerisinden sınırlandırılan belirlenmiş bir alan içerisinde hareket edebilmektedir. Bu da robotun güvenli bir çalışma ortamı ve alan içerisinde bulunan diğer parçalara çarparak zarar vermesini önlemekte ve hedef noktaya, robotun daha kısa zamanda ulaşmasını sağlamaktadır. Yapılacak uygulamanın niteliğine göre robot kolun eksen sayısı tercihi yapılmalıdır.

## Paralel Robotlar



(Şekil 3.8 Paralel robot modeli)

Çoğu çağdaş robot açık kinematik zincir yapısına sahip iken, paralel robotlar kapalı kinematik zincir yapısına sahiptirler. Altı serbestlik derecesine sahip ilk paralel manipülatör 1965 yılında Stewart tarafından ortaya çıkarıldığından Stewart-Platformu olarak da anılır. Kapalı zincir yapısı paralel robota pek çok avantaj sağlamaktadır. Birden fazla paralel bağlantı sayesinde daha hafif yapılarla maksimum yükler taşınabilir. Eklem yapısı seri olmadığından hatalar eklemeli değildir. Daha rijit bir yapı sunduğundan yüksek hızlarda pozisyonlama hassasiyeti bakımından avantaj sağlamaktadır.

### Hareket Kabiliyetlerine Göre Robot Kolların Karşılaştırılması

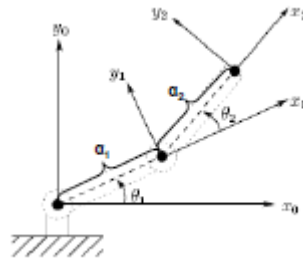
En çok insan koluna benzeyen ve hareket kabiliyeti en iyi olan mafsallı robotlar üstünlük sağlamaktadır. Ana sanayide standart bir ürün olarak kullanılan mafsallı robot kolları tercih edilmektedir. Görüntü işleme prosesinde iş parçasını insan kolu gibi tutabilen “Mafsallı robot kolları” kullanılması tercih edilir. Mafsallı robot kolları X,Y,Z de ilerlediği gibi X,Y ve Z de rotasyon hareket kabiliyetine de sahiptir. Eğer proses daha basit ve X,Y ve Z rotasyona ihtiyaç yok ise “Kartezyen robot kolları” da kullanılmaktadır.

### 3.2 Robot Kol Kinematiği

Robot kolu, eklemlerle birbirine bağlanmış katı cisimlerin açık uçlu kinematik zinciri olarak tanımlanabilir. Zincirin bir ucu sabit bir zemine bağlı iken, diğeri sonlandırıcıya bağlıdır. Robot kolu kinematiği; katı cisimlerin kinematik zinciri olarak kabul edilen bu yapının hareketlerinin incelenmesi ve ulaşılacak konumun analizi ile ilgili çalışmaları ve yöntemleri içerir. Robot kolunun hareketi, sahip olduğu her bir hareketli uzvun hareketlerinin toplamından oluşur. Robot kolu kinematiği incelenirken, robot kolu düz kinematiği veya ters kinematiği yöntemleri kullanılmaktadır. Bugüne kadar yapılan çoğu robot kolu uygulamasında, ters kinematik yöntemi kullanılmıştır. Kinematik hesaplarda genellikle fiziksel engeller, moment ve kayıplar göz ardı edilmektedir.

### 3.3 Robot Kolu Düz Kinematiği

Robot kolu düz kinematiği; robotun oturduğu tabanı, koordinat sisteminin merkezi noktası kabul ederek, robot kolu uç noktasının koordinat sisteminde konumunun bulunması olarak tanımlanabilir. Robot kolu eklem noktalarının açısal pozisyonlarının belirlenmesi, bazı algılayıcılarla sağlanabilir. Bu algılayıcıların verdiği bilgilerle aşağıdaki şekilde gösterilen iki mafsallı robotun 1 ve 2 numaralı açıları robotun uç noktasının koordinat bilgilerinin belirlenmesinde kullanılmaktadır.

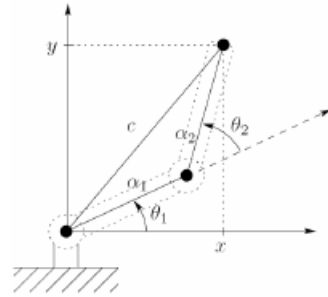


Aşağıdaki denklemde iki eklemlili bu robot kolunun düz kinematik denklemleri olarak tanımlanmıştır.

$$\begin{bmatrix} x_2 \cdot x_0 & y_2 \cdot x_0 \\ x_2 \cdot y_0 & y_2 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}$$

### 3.4 Robot Kolu Ters Kinematiği

Robot kolu ters kinematiği; robot kolunun uç noktasının koordinat bilgilerinden hareketle şekilde verilen robot kolunun 1 ve 2 numaralı açılarının bulunması şeklinde ifade edilebilir. Robot kolunun uç noktasının verilen koordinat bilgileriyle açısının bulunmasında, şekilde de görüleceği üzere iki farklı çözüm bulunabilir. Bunlara dirsek yukarı ve dirsek aşağı çözümleri diye bilinmektedir. Açılarının bulunmasında robot kolunun eklem sayısına göre, bazen bir çözüm veya sonsuz çözüm de bulunabilmektedir.

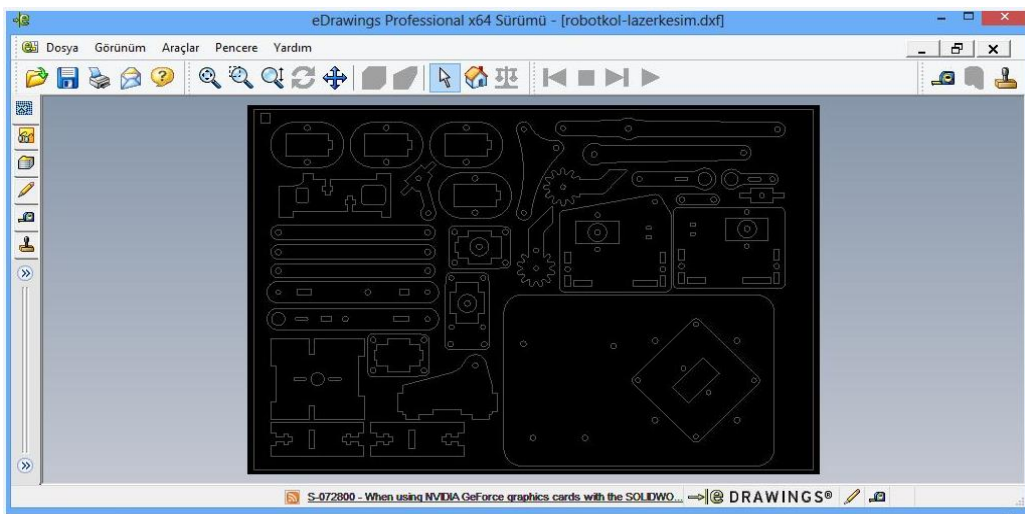


$$\sin(\theta_2) = \pm\sqrt{1 - D^2}$$

### 3.5 Robot Kol Tasarımı

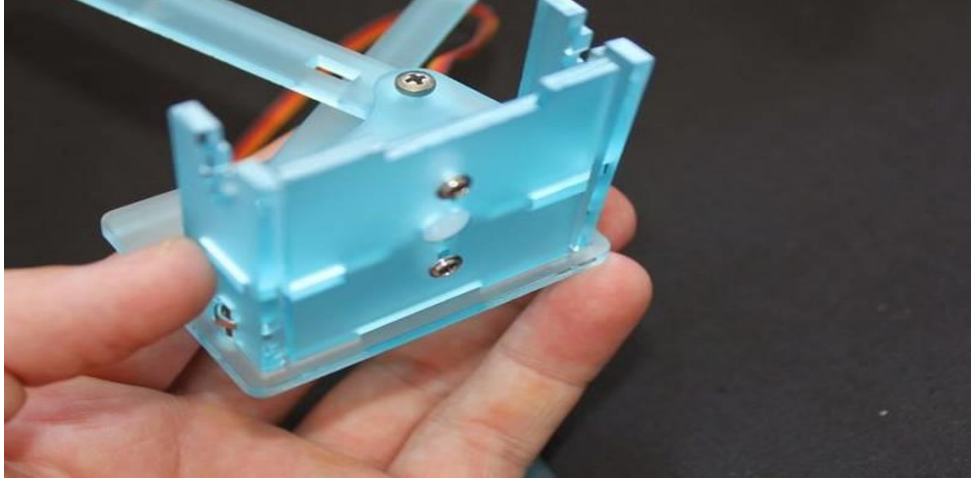
#### 3.5.1 Mekanik Kısım Tasarımı

Mekanik kısım tasarımını Autocad programı kullanılarak 1 boyutlu olarak parça çizimi yapılmıştır ve Lazer kesim için çıktı alınmıştır.

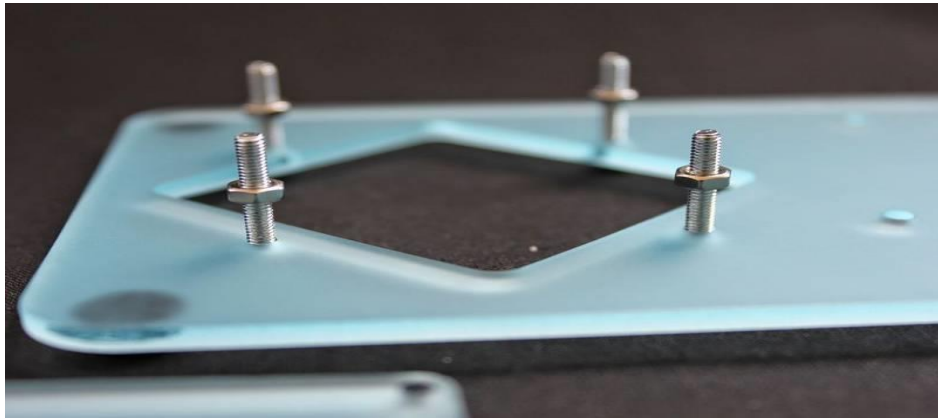
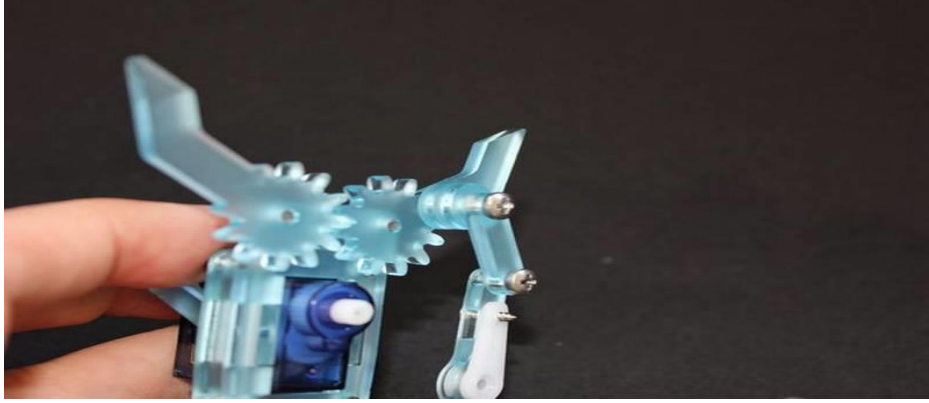


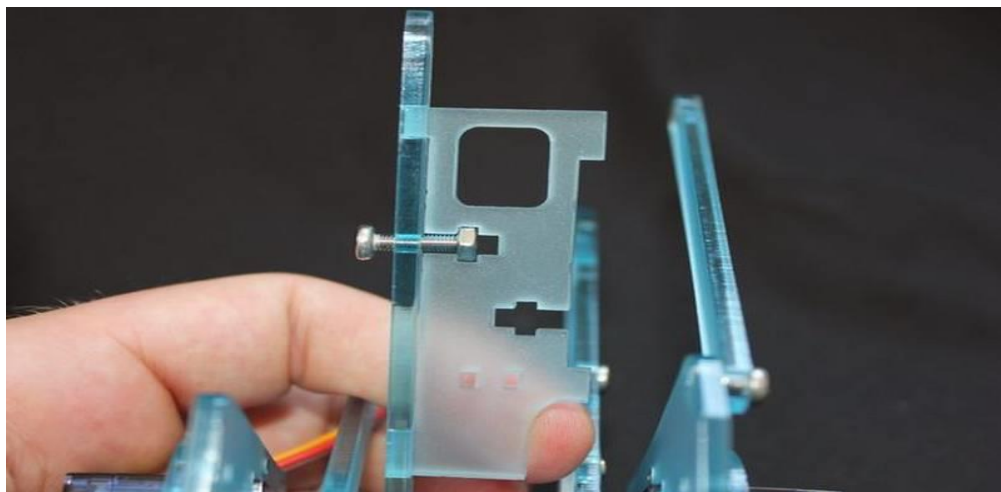
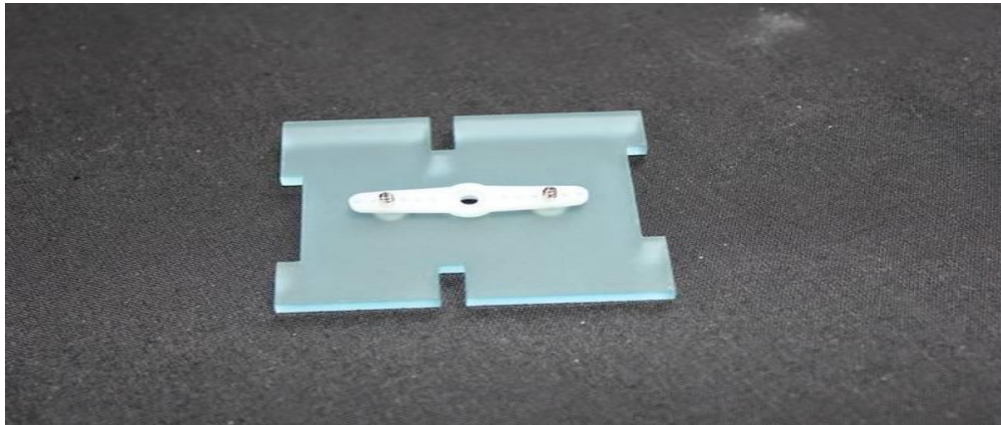
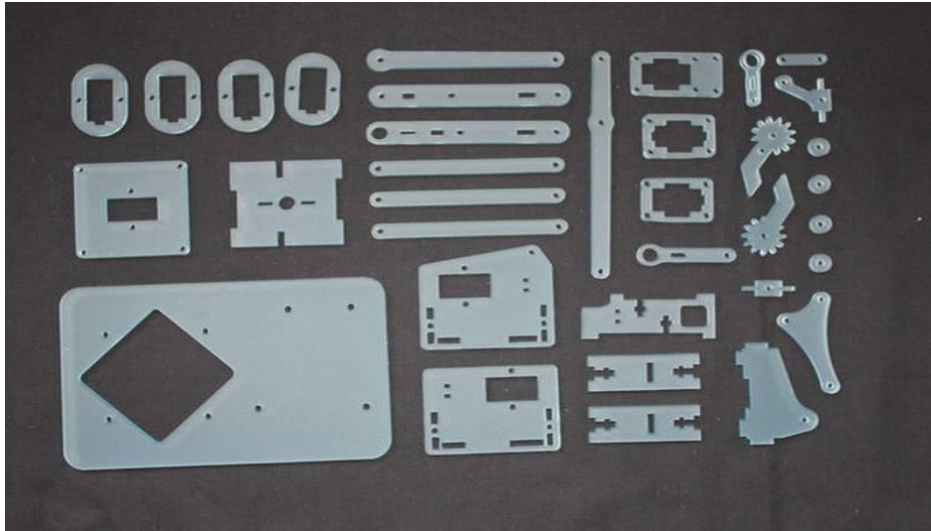
(Şekil 3.9 Mekanik kısım tasarımı)

### 3.5.2 Mekanik Kısım Montajı

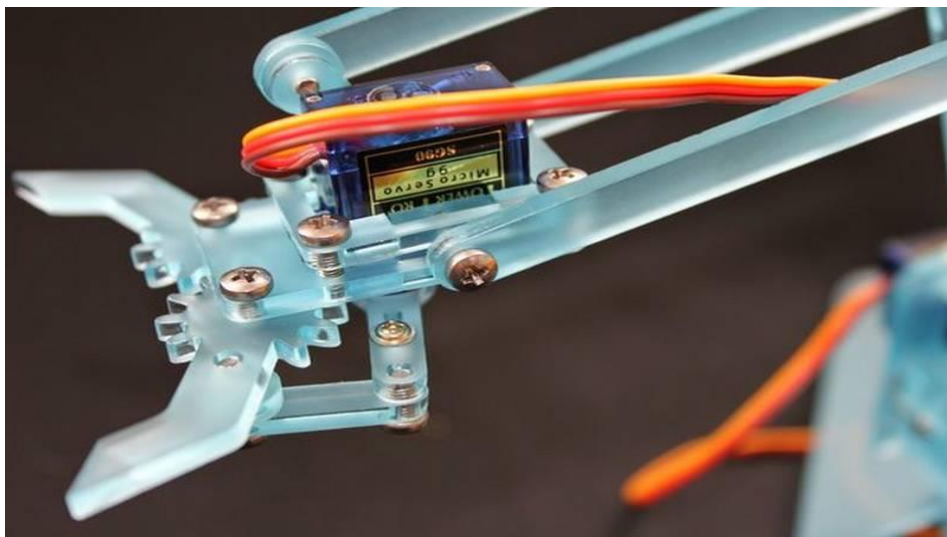
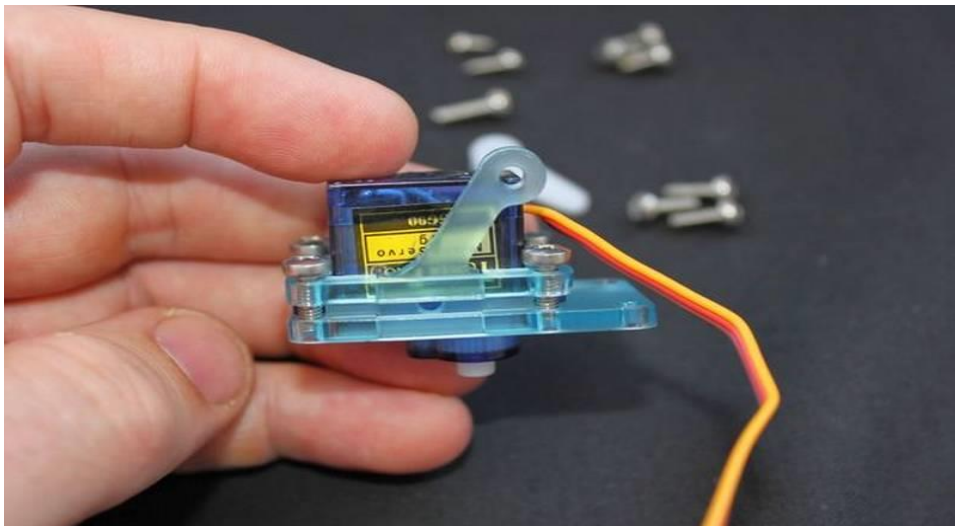
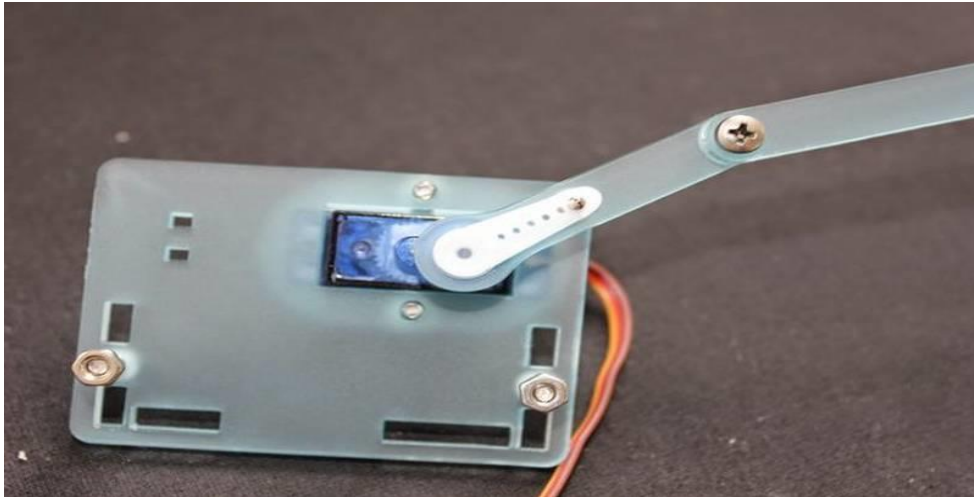


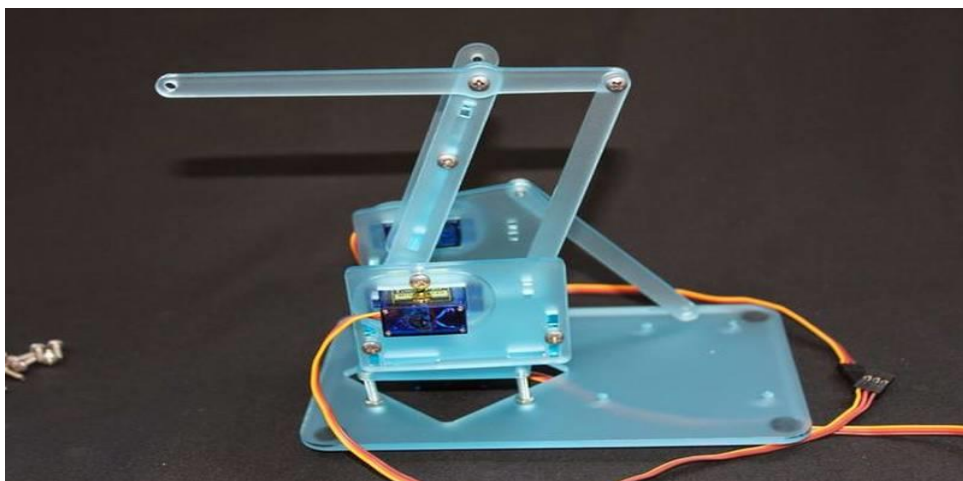
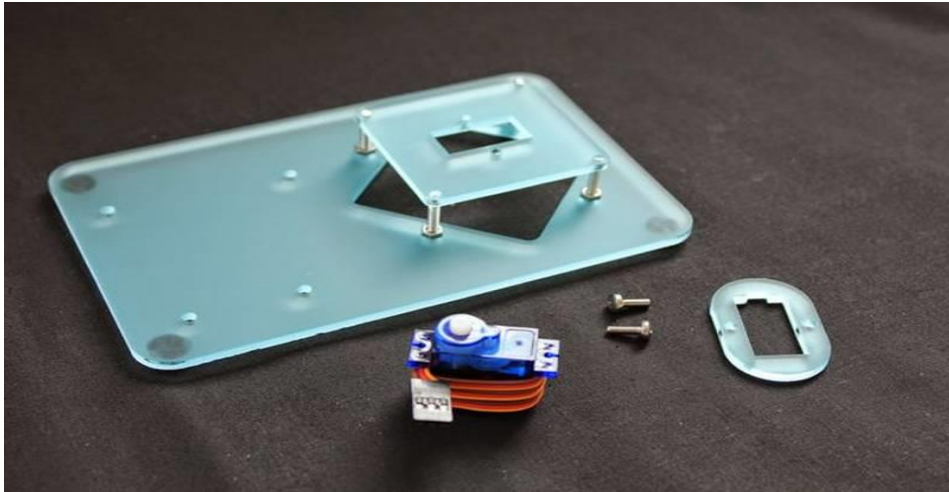
(Şekil 3.10 Mekanik kısım montajı)







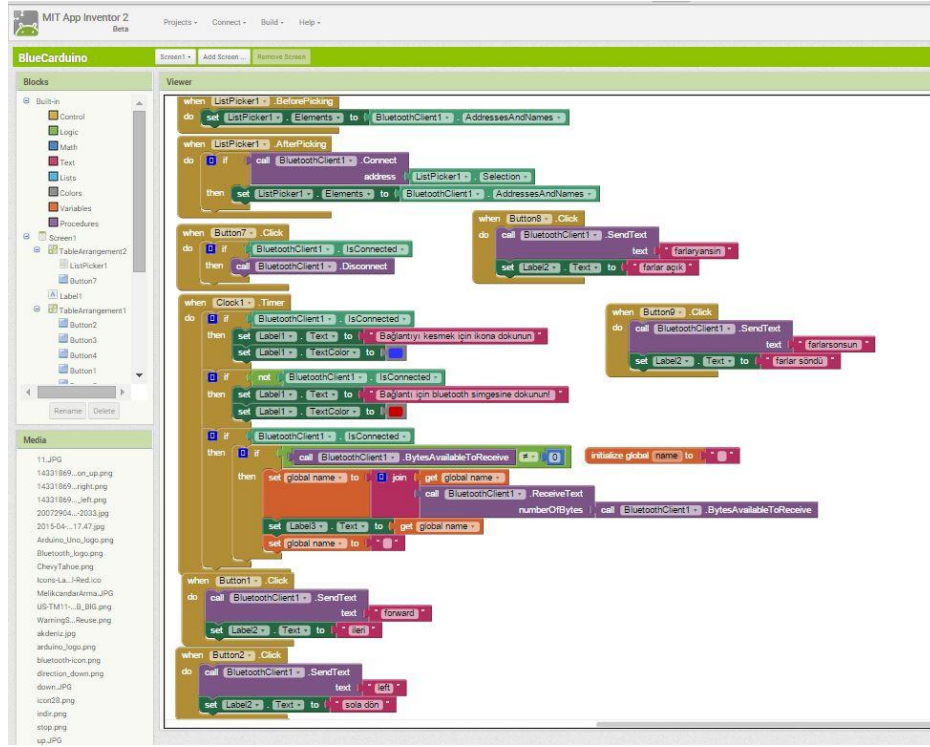






### 3.6 Android Programlama

Mit App Inventor kullanarak android blok programlama yapılmıştır.



(Şekil 3.11 Mit App Inverto)

### 3.7 Arduino Kodları

```
//motor pinleri
const int sag_ileri=11;
const int sag_geri=10;
const int sol_ileri=9;
const int sol_geri=;
const int ena=5;
const int enb=4;
const int motor_hiz = 200; // motor hızı
int led = 13;
void setup()
{
  Serial.begin(9600);
```

```

pinMode(led,OUTPUT);
pinMode(sag_ileri,OUTPUT);
pinMode(sag_geri,OUTPUT);
pinMode(sol_ileri,OUTPUT);
pinMode(sol_geri,OUTPUT);
pinMode(ena, OUTPUT);
pinMode(enb, OUTPUT);
}

void loop()
{
    int val;
    if (Serial.available())
    {
        val=Serial.read();
        if(val=='L')
        {
            analogWrite(ena,motor_hiz);
            analogWrite(enb,motor_hiz);
            digitalWrite(sag_ileri,HIGH);
            digitalWrite(sag_geri,LOW);
            digitalWrite(sol_ileri,LOW);
            digitalWrite(sol_geri,HIGH);
            digitalWrite(led,LOW);
        }
        else if(val=='R')
        {
            analogWrite(ena,motor_hiz);
            analogWrite(enb,motor_hiz);
            digitalWrite(sag_ileri,LOW);
            digitalWrite(sag_geri,HIGH);
            digitalWrite(sol_ileri,HIGH);
            digitalWrite(sol_geri,LOW);
            digitalWrite(led,LOW);
        }
    }
}

```

```

}
else if(val=='C')
{
digitalWrite(sag_ileri,LOW);
digitalWrite(sag_geri,LOW);
digitalWrite(sol_ileri,LOW);
digitalWrite(sol_geri,LOW);
digitalWrite(led,HIGH);
}
else if(val=='F')
{
    analogWrite(ena,motor_hiz);
    analogWrite(enb,motor_hiz);
digitalWrite(sag_ileri,HIGH);
digitalWrite(sag_geri,LOW);
digitalWrite(sol_ileri,HIGH);
digitalWrite(sol_geri,LOW);
digitalWrite(led,HIGH);
}
else if(val=='B')
{
    analogWrite(ena,motor_hiz);
    analogWrite(enb,motor_hiz);
digitalWrite(sag_ileri,LOW);
digitalWrite(sag_geri,HIGH);
digitalWrite(sol_ileri,LOW);
digitalWrite(sol_geri,HIGH);
digitalWrite(led,HIGH);
}
else
digitalWrite(led,LOW);
}
}

```

## 4.GÖRÜNTÜ İŞLEME

### 4.1 Görüntü İşleme Tanımı

Görüntü işleme, sinyal işleme kapsamında yer alan ve hızlı gelişen önemli bir alandır. Görüntü işleme, daha çok mevcut görüntüleri işlemek, diğer bir ifadeyle mevcut resim ve grafikleri, değiştirmek, düzenlemek ya da iyileştirmek için kullanılır. Görüntü işleme, sayısal bir resim hâline getirilmiş olan gerçek yaşamdaki görüntülerin bir giriş resmi olarak işlenerek o resmin özelliklerinin ve görüntüsünün değiştirilmesinde sonucunda yeni bir resim oluşturulmasıdır. Projede yazılım ile robot kolu belirli bir uzaklıktaki noktadaki malzemeyi algılayarak ana renklerine Kırmızı, Yeşil, Mavi, (RGB) göre farklı noktalara taşımayı sağlayacaktır.

### Görüntü İşleme

- İki boyutlu bir görüntünün bilgisayar yardımı ile işlenmesidir.
- Çıktı genelde yeni bir görüntüdür. Görüntü kalitesi artırılır veya istenen şekle getirilir.
- Durağan görüntüler (resimler, fotoğraflar) dışında video ve animasyonlar üzerinde de işlem yapılabilmektedir.

### 4.2 Görüntü Yakalama

Robot görüşü sistemleri, ışık yoğunluğu değişen optik görüntüler üzerinde işlem yaparlar ve en başta elde edilen bilgiyi elektriksel işarete dönüştürülür. Bu işlem, ortamın bir ortam ekipmanlarla taramasıyla sağlanır. Bir ortamdaki görüntüyü tarama, ortamdaki her noktanın (piksel) ışık yoğunluğunun metodik olarak belirlenmesi ve kaydedilmesi olayıdır. Bir nokta(piksel) tarayıcı aracın bir seferde inceleyebildiği en küçük alan olarak tanımlanabilir.

Projede görüntü alımı aşamasında A4 TECH PK635K Webcam kamerasından gerçek zamanlı olarak görüntüler alınmıştır. Elde edilen görüntüler 320x240 imge çözünürlüğünde olup RGB formata sahiptir. Kameradan alınan görüntüler ilk olarak bir renk süzgecinden geçirilerek Matlab ile RGB (Red-Green-Blue) tespiti yapılarak robot kolun hareketi sağlanmıştır.

### 4.3 Sayısal Görüntü

#### Renkli Görüntü

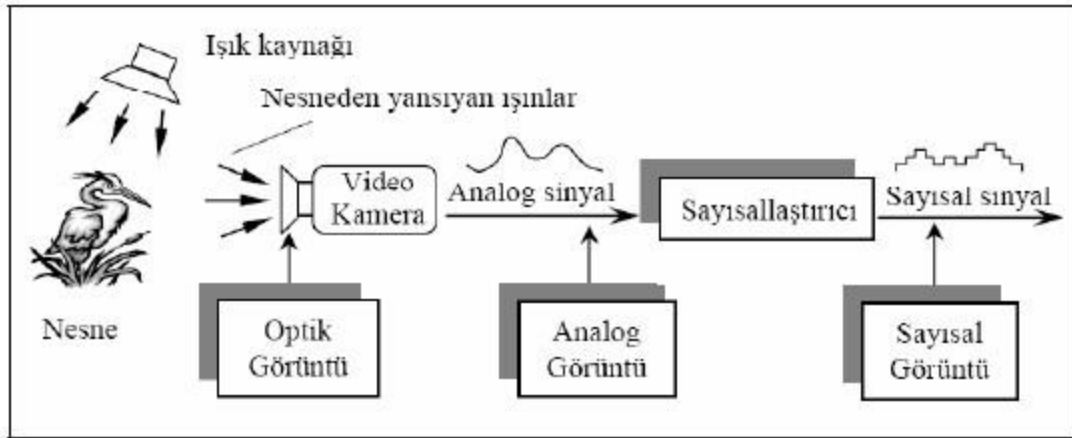
Üç fonksiyonla gösterilebilir.

$R(x,y)$ : kırmızı bileşen

$G(x,y)$ : yeşil bileşen

$B(x,y)$ : mavi bileşen

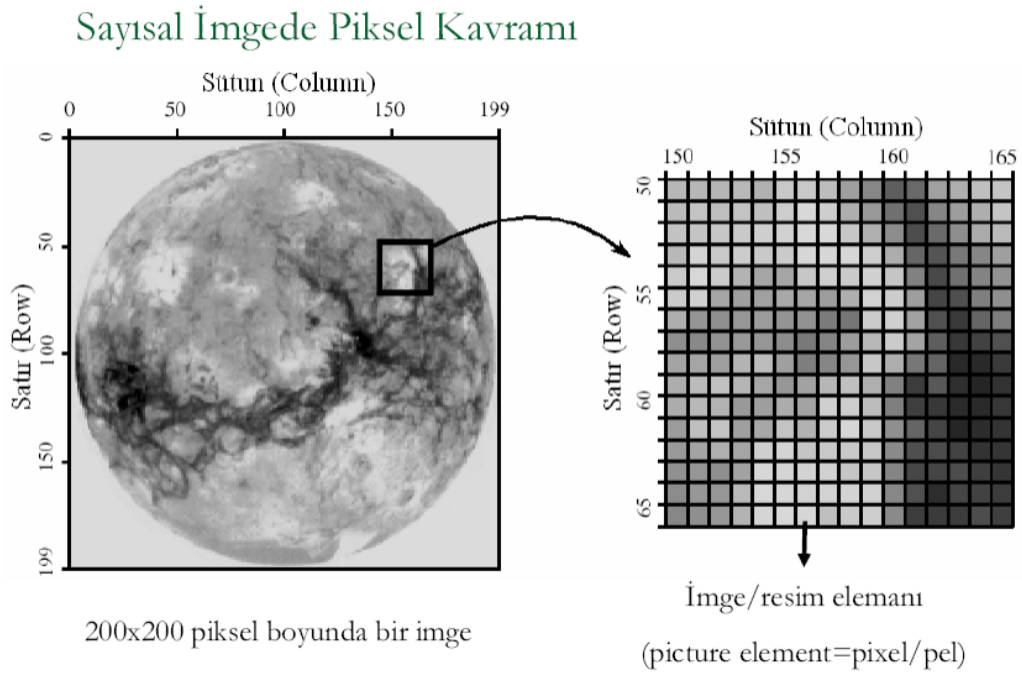
Sayısal görüntü işleme, sayısal görüntülerin bilgisayar yardımı ile işlenmesidir. Sayısal görüntü iki boyutlu bir fonksiyon olarak tanımlanır ve **MATLAB** görüntüleri birer matris olarak tanımlayıp bu işlemleri matris üzerinde yaparak görüntü işleme algoritmaları gerçekleştirir. Görüntü işleme teknikleri ile sayısal görüntü verileri kullanılarak iyileştirilmiş görüntü veya daha farklı görüntüler elde edilebilmekte ve nesne tanıma işlemleri gerçekleştirilebilmektedir. Sayısal görüntü, sayısal değerlerden oluşan ve bilgisayar ortamında görüntülenebilen görüntü olup oluşturulma aşamaları aşağıdaki (Şekil 4.1) de gösterilmiştir.



(Şekil 4.1 Görüntü yakalama ve sayısallaştırma aşamaları)

$I(x, y)$  gibi bir fonksiyonla temsil edilen analog bir görüntü veya resimde  $I$  bir şiddet birimi (örneğin parlaklık)  $x$  ve  $y$  ise görüntünün yatay ve dikey eksenindeki koordinatlarına karşı düşen değerlerdir.

Sayısal görüntü ise, bu analog görüntünün  $M$  sütun ve  $N$  satırdan oluşacak şekilde gösterilmesi sonucu elde edilir. Satır ve sütunun kesiştiği her bölgeye **piksel** adı verilir. Sonuç olarak, sayısal görüntüye çevrilen resimde  $N \times M$  adet piksel bulunur. Sayısal görüntü (imge)'de piksel kavramına ait gösterimi (Şekil 4.2)'de gösterilmiştir.



(Şekil 4.2 Piksel imgesi)

#### 4.4 Görüntü İşleme Adımları

İlk olarak sayısallaştırılan resmi geriye çevirme ile gürültüyü temizleme diğer filtremeler ile de uygulanarak gerekli algoritmalarla yeni görüntüler elde edilmesi görüntü işlemede tüm işlemler, resim üzerindeki en küçük resim elemanı olan pikseller üzerinde gerçekleştirilmektedir. Aşağıda basit bir görüntü işleme adımı (Şekil 4.3) de gösterilmiştir.



(Şekil 4.3 Görüntü işleme adımları)

#### 4.5 Seri Port İle Haberleşme

Yine isminden de anlaşılacağı gibi seri olarak bilgilerin aktarıldığı bir porttur. Veriler bir hat üzerinden zamansal olarak sıra ile yollanarak istenilen veriler sıra ile karşıya bit bit ulaştırılır. Burada zamanlama yani frekans uyumu çok önemli bir yer almaktadır bunun için protokol kavramı oluşmuştur. Aynı zamanda çift yönlü iletişim ile seri iletişim paralel iletişimin önüne geçerek daha az kablo ile maksimum mesafeye veri ulaştırır hale gelmiştir. Veriler TTL boyutundan çıkarak lojik seviyelerini +10V ile -10V arasında düzenlenmiştir.

Aşağıda bir byte'lık bir verinin seri olarak aktarılması görülmektedir;

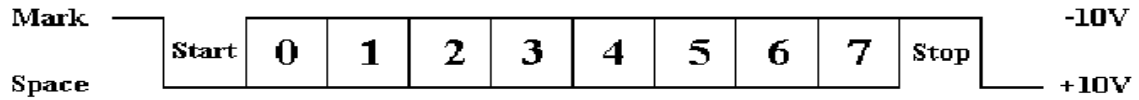


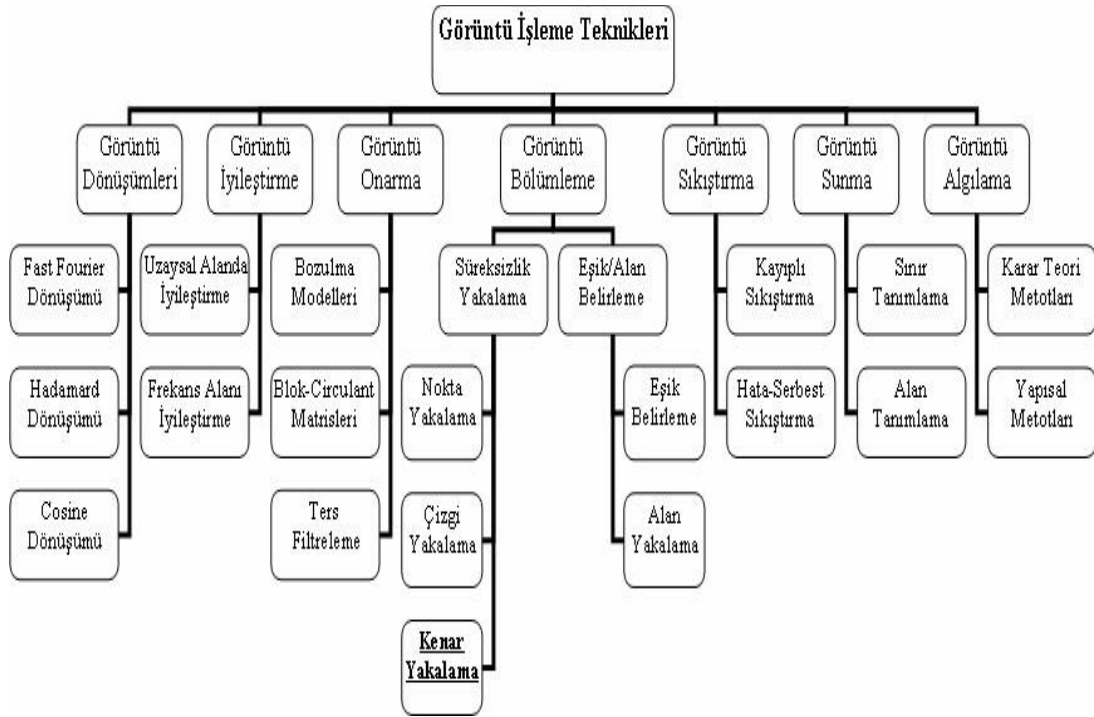
Figure 5 : RS-232 Logic Waveform

Görüldüğü gibi bir bytelık verinin sekiz ayrı biti lojik 0 -10V ve lojik 1'de +10V olarak belirli periodlarla karşıya aktarılırken aktif low olarak başlangıç manasına gelen START biti ve sonunda da aktif high olarak STOP biti bulunmaktadır buda protokolün bittiğini belirtir. Seri port parametrelerine dikkat edilecek olunursa gönderilen verinin bit uzunluğu 5 ile 8 bit arasında değişebildiği gibi verinin hızı ve veri kontrolü için çift yada tek party biti kullanılabilmektedir. Fakat kullanım kolaylığı ve gereksinimler sonucunda paritsiz, 9600 baud (baud= sn gönderilen bit sayısı  $9600/8=1200$  byte sn yollanabiliyor.) hızında ve 8 bitlik veri protokolü kabul görmüştür.

Seri Haberleşmenin Ana mantığı bu şekildedir. Arduino ile Matlab Arasında seri haberleşmeyi ArduinoIO kütüphanesini indiriyoruz ilk olarak arduino indirdiğimiz

kütüphanenin içinde pde isimli dosyayı arduino lib içine kaydediyoruz ve arduino kartımızın içine adioes isimli kütüphane kodlarını yüklüyoruz. Daha sonra matlab programında kodları açıyoruz ve ilk olarak install arduino-matlab isimli dosyayı kurulumunu yapıyoruz sonra matlab gui içinde yazdığımız kodları arduino ile seri iletişim içine sokarak yolluyoruz ve arduino takılı olduğu com numarasını tanımlamaktayız.

#### 4.6 Görüntü İşleme Teknikleri

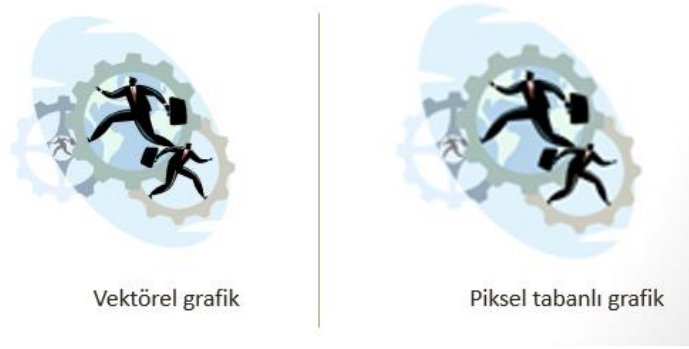


(Şekil 4.4 Görüntü işleme teknikleri)

#### Görüntü formatları

Piksel tabanlı grafiklerde ‘Grafik piksellerden oluşur’ Vektör tabanlı grafiklerde ise ‘Grafik matematiksel fonksiyonlardan oluşmaktadır’





(Şekil 4.5 Görüntü karşılaştırılması)

Grafik tasarımlarında kullanılan yazılımlar birbirleri ile dosya alışverişinde bulunarak çalışır. Bir resmi veya vektörel çizimi oluşturduğumuz uygulama programından bağımsız hale getirip sayfa düzenleme programına ya da vektörel programlara aktarmak için farklı bir formatta kaydedilmesi gerekir.

Photoshop programının oluşturacağı dosya PSD uzantılı

- Corel DRAW- CDR
- Fireworks- PNG
- Illustrator- AL

### **PICT**

PICT formatı bütün programların ortak kullandığı dosya formatıdır. Bu format herhangi bir uygulama programına aktarıldığında resim bilgisi sayfaya dâhil olur.

### **EPS**

EPS formatı hemen hemen bütün çizim ve sayfa düzenleme programları tarafından desteklenir. EPS ayrıca vektörel programların (Illustrator ve CorelDRAW) sayfa düzenleme programlarına dosya aktarım formatıdır.

### **TIFF**

TIFF formatı bilgisayarlar arası ortak bir dosya formatıdır. Tüm programlar tarafından desteklenir. Bu formatta kayıtlı dosyalar, herhangi bir uygulama programında sayfa içine alındığında görüntünün ve zeminin renk değerlerini azaltma ve değiştirme olanağı verir.

Örneğin, farklı renklerde kullanacağımız bir görüntü ya da logoyu TIFF formatla kaydedip sayfaya yapıştırdığımızda renklerini değiştirebilirsiniz.

### **JPEG**

JPEG formatı özellikle sıkıştırma verimliliği nedeni ile internet ortamında oldukça sık kullanılmaktadır. Genellikle kayıplı sıkıştırma ile elde edilen JPEG dosyaları

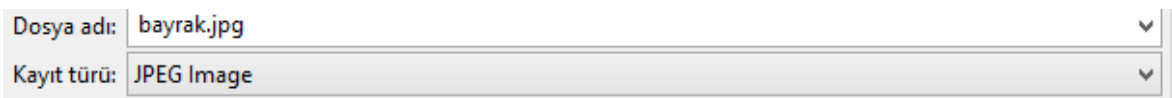
kullanılmaktadır. JPEG ve JPG formatlarının ortak özelliği gerçek renk değerlerini içermesidir. Bu nedenle foto grafik ‘çizgisel/grafiksel olmayan’ görüntüleme için kullanılır.



(Şekil 4.6) BAYRAK

Ad	2.JPG
Öge türü	JPG Dosyası
Klasör yolu	C:\Kullanıcılar\KAMIL\Masaüstü
Oluşturma tarihi	20.5.2017 03:51
Değiştirme tarihi	6.11.2014 06:13
Boyut	80,7 KB
Öznitelikler	A
Kullanılabilirlik	Çevrimdışı kullanılabilir
Çevrimdışı durumu	
Kendisiyle Paylaşılan	

(Şekil 4.7 JPG dosya özellikleri gösterimi)



(Şekil 4.8 Resim kayıt seçimi)

## PNG

PNG (Portable Network Graphics) formatı patentsizdir. PNG kayıpsız Wave Table sıkıştırma yöntemini kullanır. Şu anda mevcut olmayan kayıpsız gerçek renk ve saydamlık bilgilerini içeren resim kalitesini internet’e taşımayı amaçlamaktadır. Ayrıca sıkıştırma için değişik filtreleme algoritmaları sıkıştırma öncesi kullanılabilir.

Görüntü işleme formatları için kullanılan yöntemler farklı şekillerde sınıflandırılabilir. İşleme giren piksellere göre bir sınıflandırma yapıldığında üç farklı türde görüntü işleme yöntemi

vardır. Genel görüntü formatlarının basit bir kıyaslaması aşağıdaki (Şekil4.9) de gösterilmiştir.

Görüntü Türü	PNG (Kayıpsız)	JPEG (Kayıpsız)	JPEG (Kalite: 100)	JPEG (Kalite: 70)	JPEG (Kalite: 40)
Rastgele	34.4 MB	37.9 MB	22.6 MB	6.3 MB	3.9 MB
Manzara	18.5 MB	22.5 MB	8.8 MB	1.7 MB	1.1 MB
Kapalı Mekan	12.6 MB	18.0 MB	6.0 MB	1.0 MB	545 KB

(Şekil 4.9 Görüntü formatlarına piksellere göre sınıflandırılması)

#### 4.7 Matlab Programına Web Kamerasının Tanıtılması

Aşağıdaki bilgisayara bağlı olan kameranın özelliklerine ortaya çıkarmak ve üzerinde işlem yapabilmek için gerekli matlab kodları ve örnek yazılım aşağıda verilmiştir.

#### İmaqhwinfo

Bilgisayar desteklediği formatı

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

>> imaqhwinfo

ans =

    InstalledAdaptors: {'gentl' 'gige' 'matrox' 'winvideo'}
    MATLABVersion: '8.1 (R2013a)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '4.5 (R2013a)'

fx >>
```

#### imaqhwinfo('winvideo')

Kamera bilgileri

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

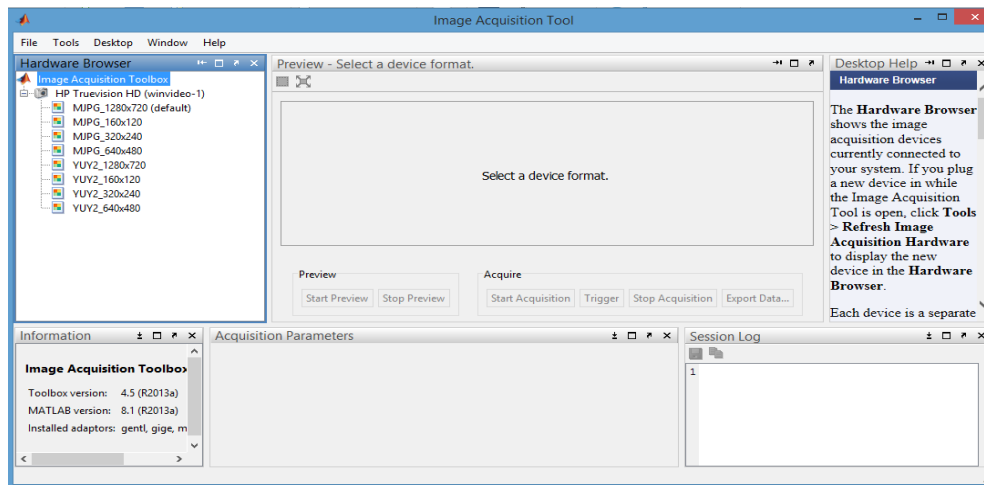
>> imaqhwinfo

ans =

    InstalledAdaptors: {'gentl' 'gige' 'matrox' 'winvideo'}
    MATLABVersion: '8.1 (R2013a)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '4.5 (R2013a)'
```

## İmaqtool

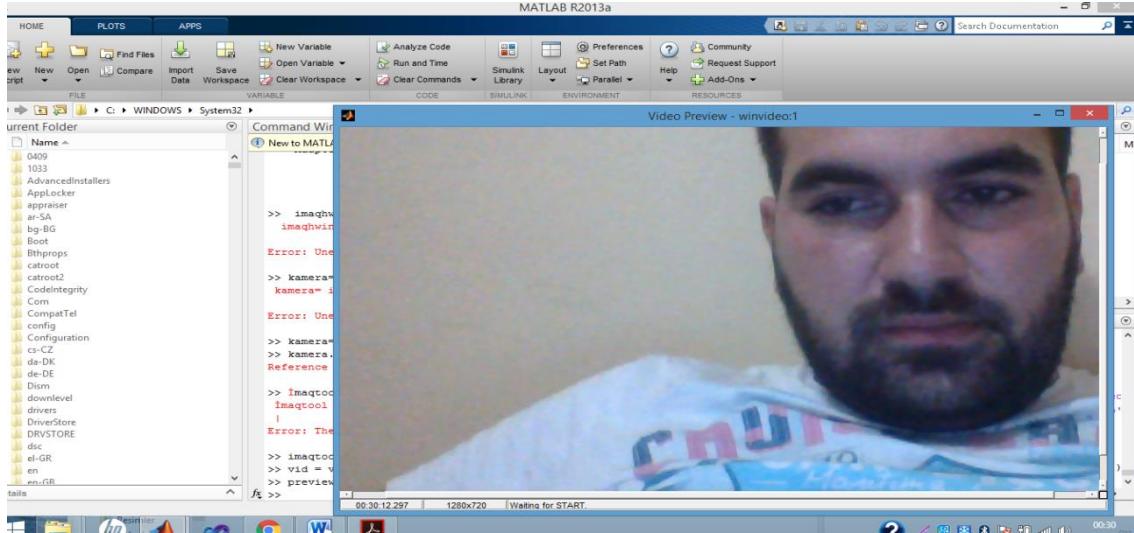
Kamera desteklediğinin bulunduğu diğer formatlar



**vid = videoinput('winvideo');**

Değişkeni vid diye tanımlanır ve ardından kamera sürümümüzü değişkene tanımlamaktadır.

**preview(vid);**

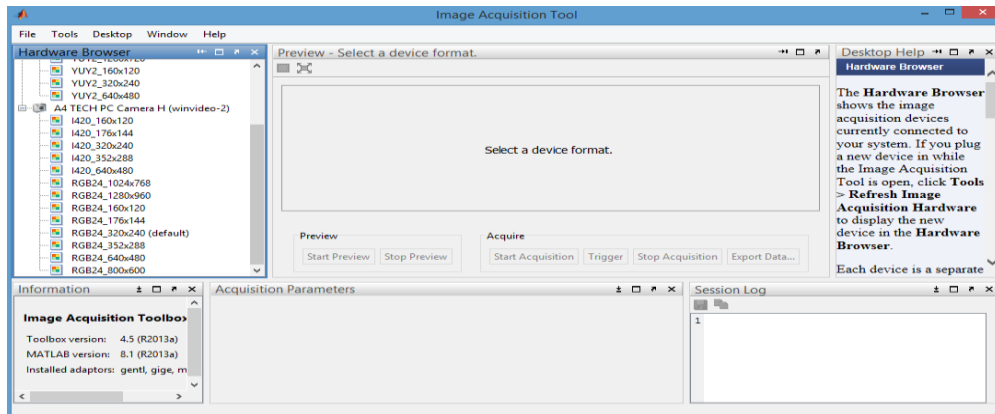


### stop(video)

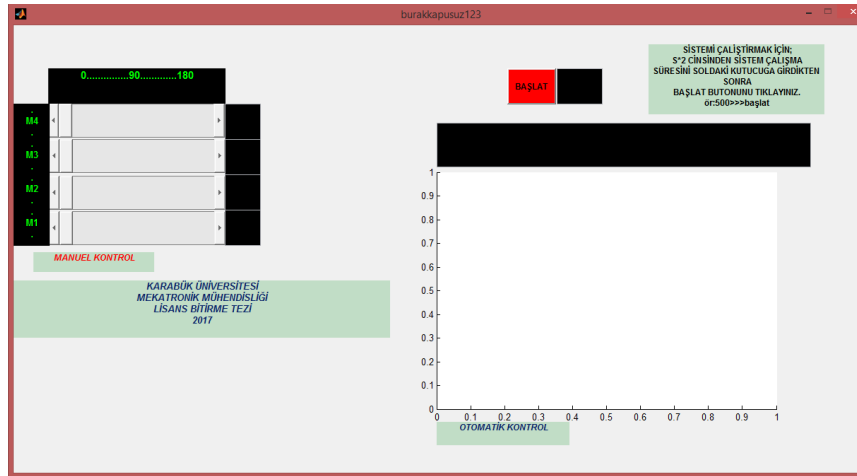
Görüntü Almayı durdurmak için kullanılır. Not; Bilgisayara Dahili kamerayı taktıktan sonra ilk olarak kamera driver kurulur sonra matlab programında bu kameradan görüntü almak için kod yazılır;

**vid = videoinput('winvideo',2,'RGB24\_320x240');**

Bu kodlarda 2 diye belirtilmesi gereklidir. 1.Bilgisayarımızın kendi kamerasıdır. RGB24\_320\*240 diye belirtilen kısımda imaqtool koduyla bulunur. Kameranin varsayılan(default) olduğu kısımdır.



## Matlab Guide Arayüz



(Şekil 4.10) Matlab guide arayüz

### 4.8 Matlab Görüntü İşleme Kodları

```
function varargout = omereviz(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @omereviz_OpeningFcn, ...
    'gui_OutputFcn', @omereviz_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function omereviz_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);
```

```

function varargout = omereviz_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
clear all;
global a;          matlab'ta a değişkeni tanımlaması yapılmıştır.
a=arduino('COM3'); arduino ile seri haberleşmenin com bağlantısı yapılmıştır.
servoAttach(a,9); % m4 motor 4'ü arduino digital 9 numaralı bacağına bağladık
servoAttach(a,8); % m3 motor 3'ü arduino digital 8 numaralı bacağına bağladık
servoAttach(a,7); % m2 motor 2'ü arduino digital 7 numaralı bacağına bağladık
servoAttach(a,6); % m1 motor 1'ü arduino digital 6 numaralı bacağına bağladık
function motor4_Callback(hObject, eventdata, handles)
global a;
sliderVal1=get(hObject,'Value');
k1=round(180*sliderVal1);
set(handles.motoryazi4,'String',num2str(k1));
servoWrite(a,9,k1);
% --- Executes during object creation, after setting all properties.
function motor4_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
% --- Executes on slider movement.
function motor3_Callback(hObject, eventdata, handles)
global a;
sliderVal2=get(hObject,'Value');
k2=round(180*sliderVal2);
set(handles.motoryazi3,'String',num2str(k2));
servoWrite(a,8,k2);
function motor3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to motor3 (see GCBO)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
function motor2_Callback(hObject, eventdata, handles)

```

```

global a;
sliderVal3=get(hObject,'Value');
k3=round(180*sliderVal3);
set(handles.motoryazi2,'String',num2str(k3));
servoWrite(a,7,k3);
function motor2_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
% --- Executes on slider movement.
function motor1_Callback(hObject, eventdata, handles)
global a;
sliderVal4=get(hObject,'Value');
k4=round(180*sliderVal4);
set(handles.motoryazi1,'String',num2str(k4));
servoWrite(a,6,k4);
function motor1_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
function motoryazi4_Callback(hObject, eventdata, handles)
function motoryazi4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function motoryazi3_Callback(hObject, eventdata, handles)
function motoryazi3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function motoryazi2_Callback(hObject, eventdata, handles)

```



```

function motoryazi2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function motoryazi1_Callback(hObject, eventdata, handles)
function motoryazi1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in dugme1.
function dugme1_Callback(hObject, eventdata, handles)
global a;
j=str2num(get(handles.yazi1,'String'))    edittext değişkenine yazı yazdırılmıştır.
m2='Baglanti saglandi.MOTORLAR STABİLİZE EDİLİYOR'; program
başladığında edittext'yazılacak yazı gösterimi yapılmıştır.
set(handles.yazi2,'String',m2)
servoWrite(a,9,0); %    motor4 0 derece de başlattık
servoWrite(a,8,60); %    motor3 60 derece de başlattık
servoWrite(a,7,0); %    motor2 0 derece de başlattık
servoWrite(a,6,100); %    motor1 100 derece de başlattık
pause(3)                3 saniye bekletme süresi
m3='STABİLİZASYON SAGLANDI.KULLANİMA HAZİR';    bağlantı sağlandığında
edittext yazılacak kısım gösterilmiştir.
set(handles.yazi2,'String',m3)
kirmizi=0.25;            rgb değerlerini tanımlanmıştır.
yesil=0.08;
mavi=0.16;
vid = videoinput('winvideo',2,'RGB24_320x240'); 2. Dahili kamerayı başlattık vid
değişkeni tanımlanmıştır.
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb') kamera başlatma kodları yazılmıştır.

```

```
vid.FrameGrabInterval = 2;  
start(vid)  
while(vid.FramesAcquired<=j)    j değışkeni tanımlanmış ve edittext değeri girip  
kameranın açık kalma süresini ayarlanmıştır.
```

**kırmızı rengi tanımlama kodları yazılıp matris şeklinde tanımlama yapıp kırmızı kutuyla tanımlama yapılmıştır.**

```
data = getsnapshot(vid);  
diff_kirmizi = imsubtract(data(:,:,1), rgb2gray(data));  
diff_kirmizi = medfilt2(diff_kirmizi, [3 3]);  
diff_kirmizi= im2bw(diff_kirmizi,kirmizi);  
diff_kirmizi = bwareaopen(diff_kirmizi,300);  
brk = bwlabel(diff_kirmizi, 8);  
stats_kirmizi = regionprops(brk, 'BoundingBox', 'Centroid','Area');
```

**mavi rengi tanımlama kodlarını yazılıp matris şeklinde tanımlama yapıldı ve rengi mavi kutuyla tanımlama yapılmıştır.**

```
diff_mavi = imsubtract(data(:,:,3), rgb2gray(data));  
diff_mavi = medfilt2(diff_mavi, [3 3]);  
diff_mavi= im2bw(diff_mavi,mavi);  
diff_mavi = bwareaopen(diff_mavi,300);  
bmw = bwlabel(diff_mavi, 8);  
stats_mavi = regionprops(bmw, 'BoundingBox', 'Centroid','Area');
```

**yeşil rengi tanımlama kodlarını yazılıp matris şeklinde tanımlama yapıp yeşil kutuyla tanımlama yapılmıştır.**

```
diff_yesil = imsubtract(data(:,:,2), rgb2gray(data));  
diff_yesil = medfilt2(diff_yesil, [3 3]);  
diff_yesil= im2bw(diff_yesil,yesil);  
diff_yesil = bwareaopen(diff_yesil,300);  
bby = bwlabel(diff_yesil, 8);  
stats_yesil = regionprops(bby, 'BoundingBox', 'Centroid','Area');
```

hold on

```
for object = 1:length(stats_kirmizi)
```

```

bk = stats_kirmizi(object).BoundingBox;
xk = stats_kirmizi(object).Centroid;
ak = stats_kirmizi(object).Area;
rectangle('Position',bk,'EdgeColor','r','LineWidth',2)
plot(xk(1),xk(2), '-m+')
if(ak>15) %*****kirmizi cisim lokasyonu ****
    %*****
    %alma islemi*****
    servoWrite(a,9,90);
    for m4=90:-1:0
        servoWrite(a,9,m4);
        pause(0.0001);
        if m4==0
            break
        end
    end
    for m3=90:-1:0
        servoWrite(a,8,m3);
        pause(0.0001);
        if m3==0
            break
        end
    end
    for m2=0:+1:70
        servoWrite(a,7,m2);
        pause(0.0001);
        if m2==70
            break
        end
    end
    servoWrite(a,6,70);
    pause(1);
    %*****kaldirma islemi*****

```

```

        for m2=70:-1:0
servoWrite(a,7,m2);
pause(0.0001);
if m2==0
break
end
end
for m3=0:+1:90
servoWrite(a,8,m3);
pause(0.0001);
if m3==90
break
end
end

        for m4=0:+1:90
servoWrite(a,9,m4);
pause(0.0001);
if m4==90
break
end
end

        end
%***birakma islemi*****
        for m3=90:-1:0
servoWrite(a,8,m3);
pause(0.0001);
if m3==0
break
end
end
        for m2=0:+1:70
servoWrite(a,7,m2);
pause(0.0001);

```

```

        if m2==70
            break
        end
        end
        servoWrite(a,6,100);
        pause(1);
servoWrite(a,9,90);
        %baslangic konumları
        servoWrite(a,8,90); %
servoWrite(a,7,0); %
servoWrite(a,6,100); %

    end
    end
    for object = 1:length(stats_mavi)
        bm = stats_mavi(object).BoundingBox;
        xm = stats_mavi(object).Centroid;
        am = stats_mavi(object).Area;
        rectangle('Position',bm,'EdgeColor','b','LineWidth',2)
        plot(xm(1),xm(2), '-m+')
        if(am>15) %mavi cisim lokasyonu
            for m4=90:-1:0
                servoWrite(a,9,m4);
                pause(0.0001);
                if m4==0
                    break
                end
            end
            for m3=90:-1:0
                servoWrite(a,8,m3);
                pause(0.0001);
                if m3==0
                    break
                end
            end
        end
    end
end

```

```

        end
    end
    for m2=0:+1:70
        servoWrite(a,7,m2);
        pause(0.0001);
        if m2==70
            break
        end
    end
    servoWrite(a,6,70);
    pause(1);

    %*****kaldirma islemi*****
    for m2=70:-1:0
        servoWrite(a,7,m2);
        pause(0.0001);
        if m2==0
            break
        end
    end
    for m3=0:+1:90
        servoWrite(a,8,m3);
        pause(0.0001);
        if m3==90
            break
        end
    end
    for m4=0:+1:45
        servoWrite(a,9,m4);
        pause(0.0001);
        if m4==45
            break
        end
    end

```

```

        end

        %***birakma islemi*****

        for m3=90:-1:0
            servoWrite(a,8,m3);
            pause(0.0001);
            if m3==0
                break
            end
        end

        end

        for m2=0:+1:70
            servoWrite(a,7,m2);
            pause(0.0001);
            if m2==70
                break
            end
        end

        end

        servoWrite(a,6,100);
        pause(1);
        servoWrite(a,9,90);

        %baslangic konumları
        servoWrite(a,8,90); %
        servoWrite(a,7,0); %
        servoWrite(a,6,100); %

        end

        end

        for object = 1:length(stats_yesil)
            by = stats_yesil(object).BoundingBox;
            xy = stats_yesil(object).Centroid;
            ay = stats_yesil(object).Area;
            rectangle('Position',by,'EdgeColor','g','LineWidth',2)
            plot(xy(1),xy(2), '-m+')
        end

        if(ay>15)    %yesil cisim lokasyonu
                    %alma islemi*****

```

```

% servoWrite(a,9,90);
for m4=90:-1:0
servoWrite(a,9,m4);
pause(0.0001);
if m4==0
break
end
end

for m3=90:-1:0
servoWrite(a,8,m3);
pause(0.0001);
if m3==0
break
end
end

for m2=0:+1:70
servoWrite(a,7,m2);
pause(0.0001);
if m2==70
break
end
end

servoWrite(a,6,70);
pause(1);
% *****kaldirma islemi*****
for m2=70:-1:0
servoWrite(a,7,m2);
pause(0.0001);
if m2==0
break
end
end

for m3=0:+1:90

```



```

servoWrite(a,8,m3);
pause(0.0001);
if m3==90
break
end
    end
    for m4=0:+1:120
servoWrite(a,9,m4);
pause(0.0001);
if m4==120
break
end
    end
    %***birakma islemi*****
    for m3=90:-1:0
servoWrite(a,8,m3);
pause(0.0001);
if m3==0
break
end
    end
    for m2=0:+1:70
servoWrite(a,7,m2);
pause(0.0001);
if m2==70
break
end
    end
servoWrite(a,6,100);
pause(1);
servoWrite(a,9,90);
Baslangic konumlarına geri dön tekrar
servoWrite(a,8,90);

```

```

        servoWrite(a,7,0);
        servoWrite(a,6,100);
    end
end
hold off program sonu
end
stop(vid); kamerayı durdur
flushdata(vid); kamerayı kapat
clear all; temizle
function yazi1_Callback(hObject, eventdata, handles)
function yazi1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yazi2_Callback(hObject, eventdata, handles)
function yazi2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

#### 4.9.1 Matlab Script ile İleri ve Ters Kinematik Hesaplamalar

Master koldan alınan eklem açılarına veya yazılımla oluşturulan slave kolun eklem açılarına göre konum vektörlerinin bulunması gerekir. Bu amaçla yazılımda bize ileri ve ters kinematik hesaplar gerekir. İleri ve ters kinematik hesaplamaların LabVIEW 'in kendi fonksiyonları ile yapılması uzun ve karmaşık olacağından yine LabVIEW 'in altında çalışan MATLAB scripti ile yapılır. MATLAB scripti programın blok diyagramına fonksiyon paletinden eklenir. LabVIEW altında çalışan MATLAB scripti ileri ve ters kinematik hesapları yapabilmek için arka planda MATLAB programını çalıştırarak scripte yazılan MATLAB komut dizisini çalıştırarak LABVIEW programından aldığı eklem bilgilerine karşılık ileri kinematik, uç işlevcisinin koordinat bilgilerine göre ters kinematik sonuçlarını üretir. İleri ve ters kinematik hesaplamaların yapılabilmesi için öncelikle Slave kolun matematiksel modeli çıkartılıp ileri ve ters kinematik eşitlikleri bulunmalıdır. Bu nedenle öncelikle ileri ve ters kinematik modelleri ve konum vektörleri bulunur.

#### 4.9.2 Matlab İle Nesne Takibi

```
vid = videoinput('winvideo',2);    dahili kamera seçimi.
set(vid, 'FramesPerTrigger', Inf);  rgb renk ayarlamaları yapılmıştır.
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 3;
s = serial('COM3','BaudRate',9600);  matlab seri haberleşirme kısmı.
fopen(s);
start(vid) kamerayı başlat
i=1;    i değişkeni tanımlandı
q=1;    q değişkeni tanımlandı
c='A';
min=5500; 5500 – 10500 arası s^2 degeri tanımlama
max=10500;
while(i<=300)
i=i+1;
obj_found=0;
data = getsnapshot(vid);
```

```

[y x c]=size(data);
x1=x/2;
x2=x1-40;
x3=x1+40;
diff_im = imsubtract(data(:,:,1), rgb2gray(data));
diff_im = medfilt2(diff_im, [3 3]);
diff_im = im2bw(diff_im,0.18);
diff_im = bwareaopen(diff_im,300);
bw = bwlabel(diff_im, 8);
stats = regionprops(bw, 'all');
imshow(data)
hold on
for object = 1:length(stats)
    obj_found=1;
    bb = stats(object).BoundingBox;
    bc = stats(object).Centroid;
    ar=stats(object).Area;
    rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
    plot(bc(1),bc(2), '-m+')
    pix((((ar<=max)&(ar>=min))*2)+((ar>max)*4)+((ar<min)*5));
    if (bc(1)>x2)&(bc(1)<x3)
        disp('center')
        switch (pix)
            case 20
                r=1;
            case 4
                r=4;
            case 5
                r=5;
        end
    else if bc(1)<x2
        r=2;
        disp('left')
    end
end

```

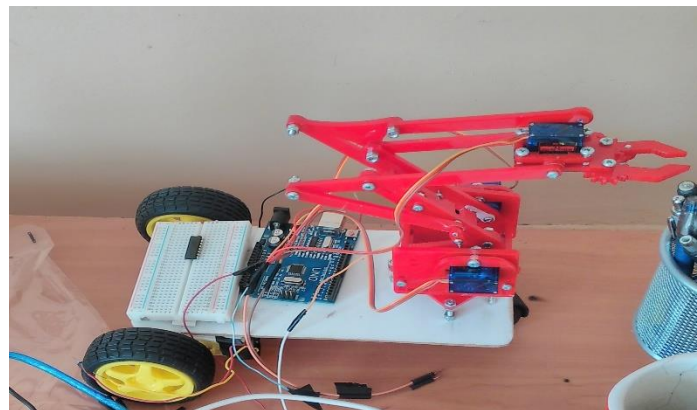
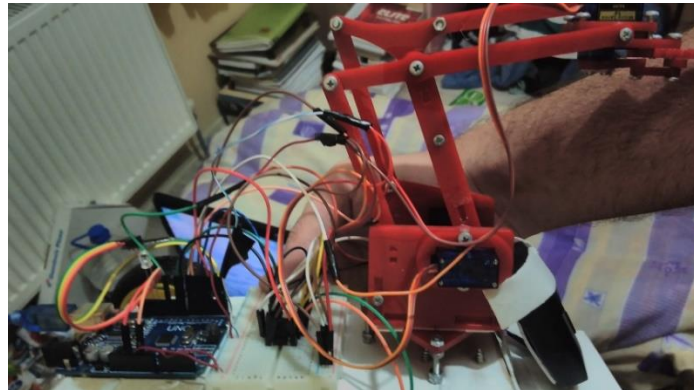
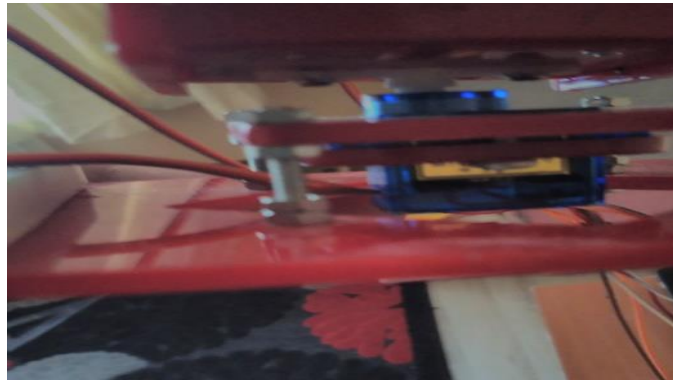
```

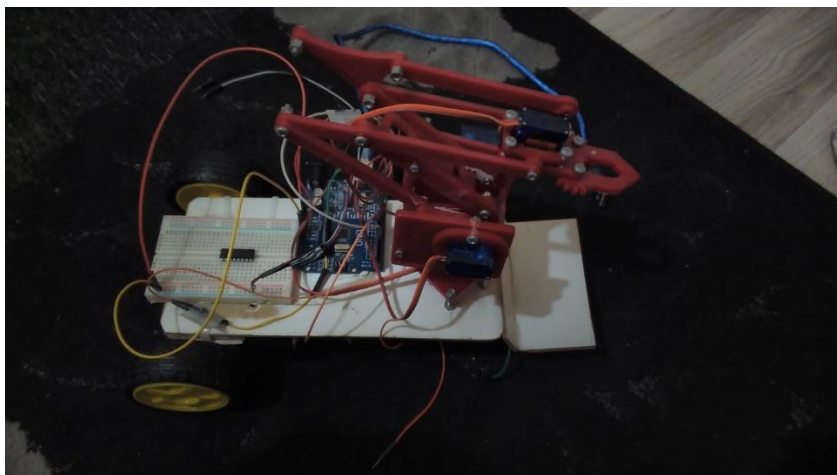
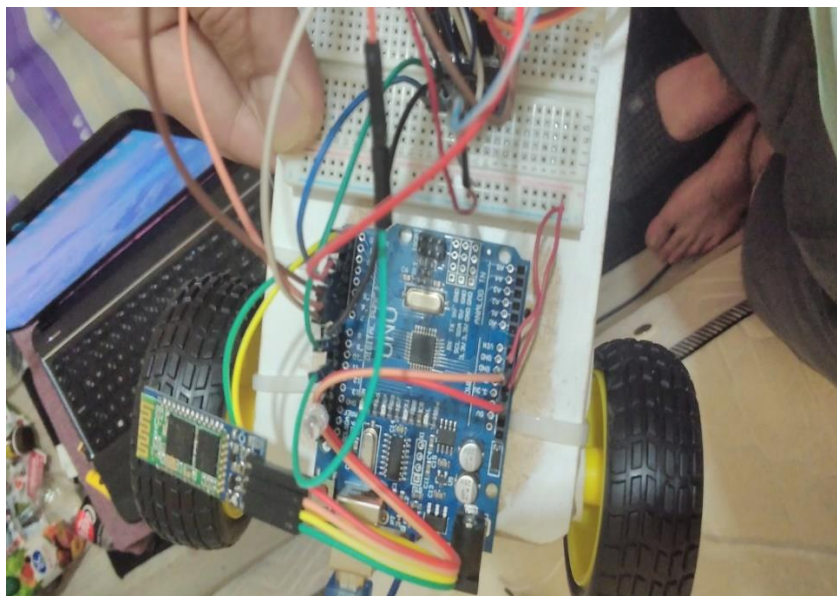
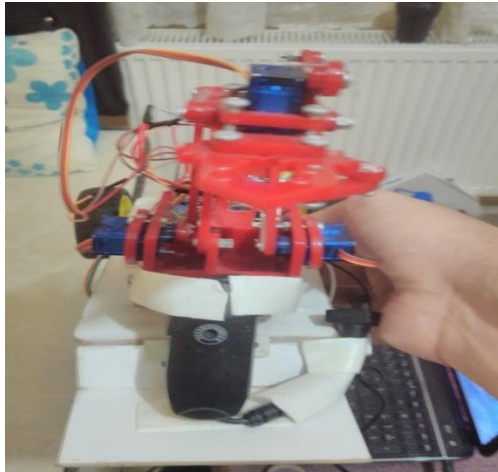
else if bc(1)>x3
    r=3;
    disp('right')
else
end
end
end
y1=1:y;
plot(x2,y1);
plot(x3,y1);
switch(r)
    case 1
        fwrite(s,'C');
    case 2
        fwrite(s,'L');
    case 3
        fwrite(s,'R');
    case 4
        fwrite(s,'B');
    case 5
        fwrite(s,'F');
    otherwise
        fwrite(s,'N');
    end
end
if obj_found == 0
    fwrite(s,'C');
    disp('no image')
end
hold off
end
fwrite(s,'C');
stop(vid);

```

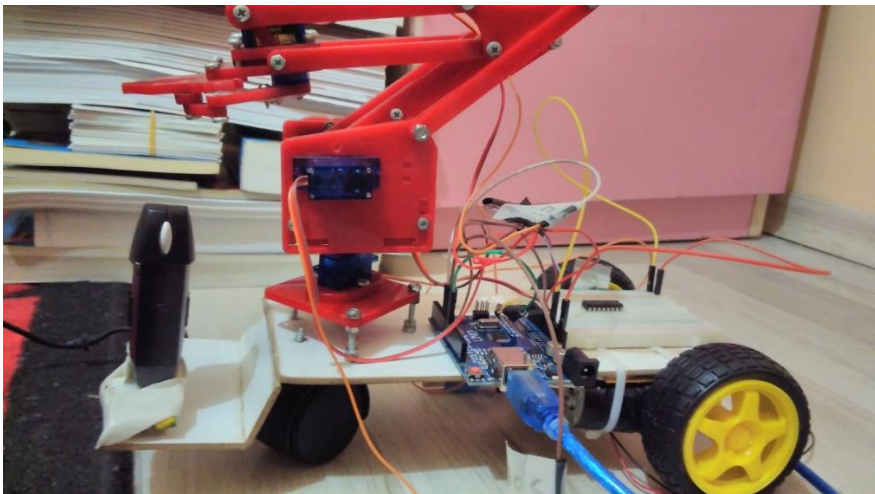
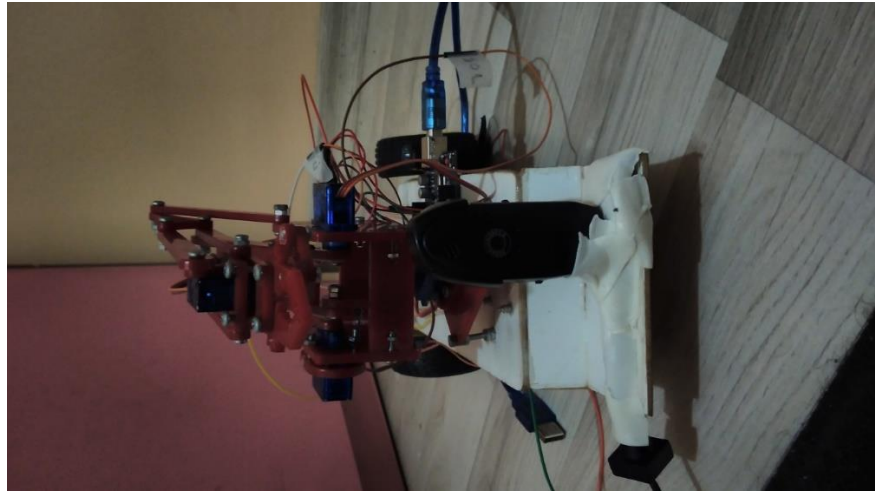
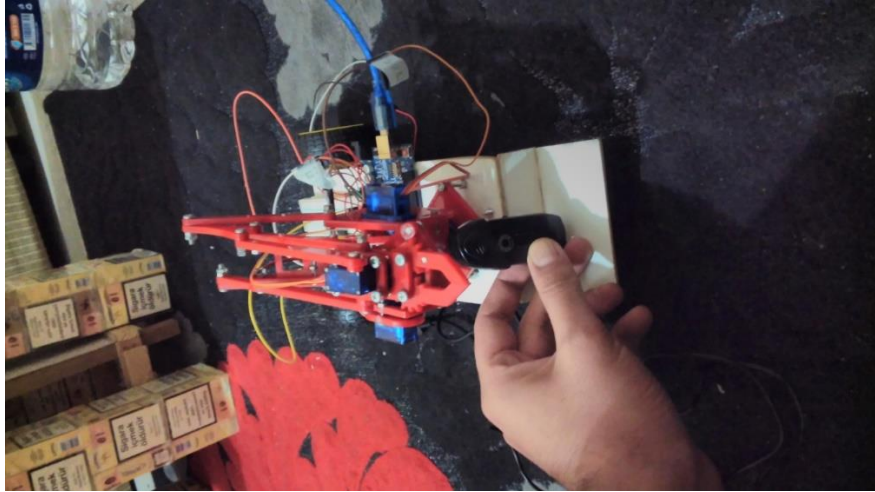
```
flushdata(vid);  
fclose(s);  
clear all
```

## Mekanik Kısım

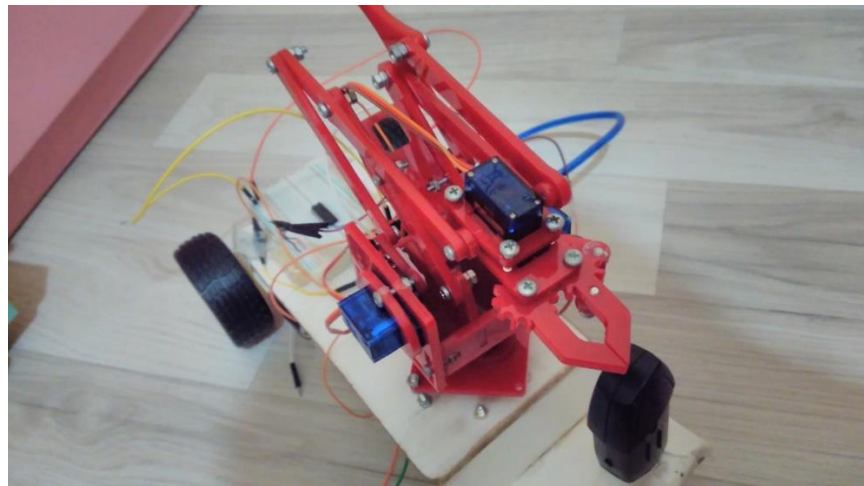












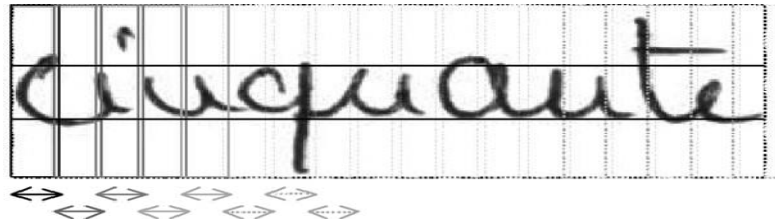
## 5. ROBOTLARDA GÖRÜNTÜ İŞLEME UYGULAMALARI

Gelişmiş ve gelişmekte olan ülkelerde, gelişen endüstriyel sanayi ile birlikte üretilen ürünler için; yüksek kalite, düşük maliyet ve zamandan kazanç aramaktadır. Bu nedenle günümüzde birçok sanayi kuruluşu için insan gücü yerine makine gücü tercih edilmektedir. Günümüz endüstriyel sanayi uygulamalarında; yüksek yük miktarı, sürekli çalışma, zamandan kazanç ve düşük hata gibi nedenlerden dolayı robotlar ve robot kolları yaygın olarak kullanılmaktadır. Özellikle savunma ve endüstriyel sanayi teknolojilerinde yaşanan gelişmeler; hızlı üretim, iş güvenliği, ürün kalitesi ve zaman gibi önemli gereksinimlerden dolayı makine gücü kullanımını zorunlu kılmakta ve bununla birlikte insanların yerini robotlar almıştır. Bu gelişmelerden dolayı robotlarda özellikle görüntü işlemenin büyük katkısı olmuştur.

## 5.1 Tasarım ve İmalat Uygulamaları

Kontrol sistemlerinde tasarım sürecinden imalat sürecine kadar birçok alanda görüntü işleme kullanılmıştır. Görüntü işleme uygulandığı alana göre kalite, pratiklik, zaman tasarrufu gibi yeni iş alanlarının doğmasına neden olmuştur. Aynı zamanda görüntü işleme imalat ve imalat öncesinde yeni ürün tasarımlarına da yardımcı olarak üretim maliyetlerini de düşürür.

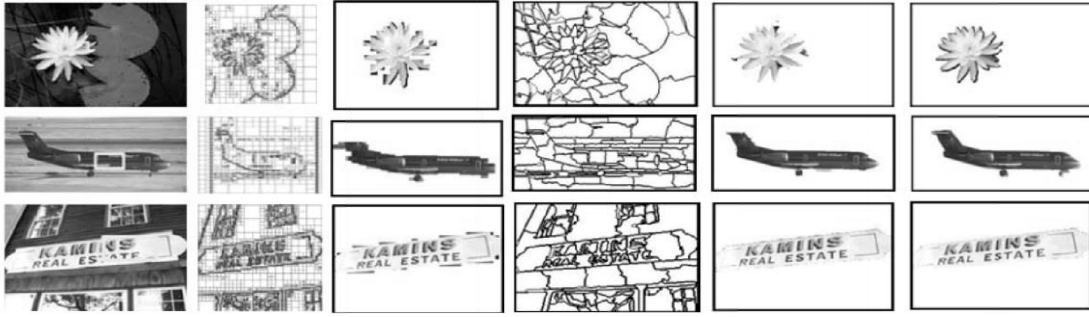
1-) Serbest el yazılarının tanımlanarak sayısal ortama aktarılma çabaları, el yazısı ile yazılmış kitapların elektronik olarak saklanabilmesini ve imzaların bilgisayar ortamında doğrulanmasını sağlayacaktır. El yazısı tanıma sistemi geliştirmiştir. Kullandığı ilk yaklaşım, sözcüğün soldan sağa doğru taranması sözcüklerin özellik vektör dizilerinin tanımlanması ile gerçekleştirilmiş ve sözcüklere biçim veren vuruş düzenini belirlemek için takım yolları geliştirilmiştir. Geliştirilen her bir yol tanımlanarak tasarlanan grafik model ile yapılmıştır.



(Şekil 5.1 Soldan sağa doğru kayan pencere bölümlemesi)

Her bir kelime için çevrimiçi ve çevrim dışı olmak üzere her iki sinyal özelliklerinden faydalanarak **IRONOFF** 'Çift yönlü veri tabanı sistemi' veri tabanından yaklaşık 30.000 kelime kullanmıştır. Sözcüklerin özellikleri ise tasarlanan sistemde çevrim dışı tanıma sistemi içerisinde karşılaştırmalar için tutulmuştur. Çevrimdışı tanıma sistemleri sözcükleri anlık olarak değil, veri tabanından alarak incelemektedir. Yapılan çalışmada çevrimiçi sistemin, çevrim dışı sisteme göre tanıma oranının yüksek olduğu görülmüştür. Fakat çalışmada yapılan deneysel sonuçlarda her iki sistemin birbirini bütünleyici olarak daha verimli çalıştıkları görülmüştür. Geliştirilen bu el yazısı sistemi imza doğrulamada kullanılabileceği gibi diğer birçok el yazısı tanımlarında kullanılabilmektedir.

2- Su altı görüntüleri, akustik kamera ile elde edilmektedir. Görüntüsü alınan nesnelerin yön ve mesafe ölçümünde su altı sonar sistemi ile beraber düzlemsel bir ızgara kullanılmıştır. Düzlemsel ızgara, aynı zamanda su altı kamerasının kalibrasyonunda da kullanılmaktadır. Çalışmada görüntü mesafelerinden maksimum komşuluk ilişkisi ile su altı üç boyutlu sahnenin yeniden yapılandırılmasında kullanılacak sistem için yeni bir metot geliştirilmiştir. Geliştirilen sistem ile su altından belirli yön ve mesafeleri bilinen nesnelerden alınan görüntüler bilgisayar vasıtasıyla işlenerek üç boyutlu yeniden yapılandırma gerçekleştirilmiştir. Görüntülerin etiketlenmesi, işlenen görüntü içerisinde yer alan nesnelerin seçilmesini ve ayrıştırılmasını sağlamaktadır. Etiketleme ile görüntü üzerinde istenmeyen alanlar elimine edilebilmekte, bu sayede işlemler daha hızlı yapılabilmektedir.



(Şekil 5.2 Akıllı etiket ile elde edilen sonuçlar)

## 5.2 Savunma ve Güvenlik Uygulamaları

Görüntü işleme savunma ve güvenlik alanlarında genel olarak uzaktan nesne tanıma ve çözümleme üzerine kullanılmaktadır. Uzaktan elde edilen görüntülerin anlık olarak incelenmesi, yorumlanması ve değerlendirilmesi aşamalarında kullanılan görüntü işleme bu alanda birçok konuya yardımcı olabilecek bir çalışma alanı oluşturmıştır. Görüntü işleme kullanarak otomatik bir parmak izi tanıma sistemi geliştirmiştir. İlk aşamada parmak izi okuyucu yardımıyla ya da dosya yoluyla alınan parmak izi görüntülerine görüntü işleme uygulanarak, küçük parçalara bölünmüş ve üzerinde işlem yapılacak alan arka plandan ayırmıştır. Daha sonra ayrılan görüntüler, gri seviye olarak incelenmiş ve görüntülerden referans noktaları elde edilmiştir. Sistem hızını ve performansını arttırmak için parmak izi resimleri üzerindeki istenmeyen unsurlar temizlenmiştir. Temizlenip iyileştirilen resimlere

bölgesel ikili dönüşüm uygulanarak siyah beyaz seviyeye çevrilmiştir. Çevrilen görüntülere inceltme uygulanarak, özellik noktaları olarak adlandırılan uç, çatal noktalar ve bunlarla ilgili parametreler bulunmuş ve yalancı özellik noktaları elenmiştir.

Son olarak da elde edilen son veriler karşılaştırma algoritmaları ile karşılaştırılmıştır. Çalışmada, sunulan adımların tamamını gerçekleştiren bir yazılım geliştirilmiştir.



(Şekil 5.3 Geliştirilen sistem doğrulama modu parmak izi)

Hem tanıma, hem de onaylama ve doğrulama modu da çalışan sistem, 100 parmak izinin bulunduğu bir veri tabanında başarıyla test edilmiştir. Parmak izinde, nokta, yaka sonu, ada, çatal, kısa yaka, kesişim, yaka ve destek noktaları olmak üzere birçok özellik noktaları bulunmaktadır. Fakat en çok kullanılan uç ve çatal noktalarıdır. Otomatik parmak izi teşhis sistemlerinin performanslarının artırılması, kullanılacak parmak izi görüntülerinin kalitesinin artırılmasına bağlıdır. Diğer bir çalışmada motorlu araçlar için Türk plaka standartlarına uyan plaka tanıma sistemi geliştirmişlerdir. Sistem video kameradan alınan anlık görüntüler üzerinde çalışmaktadır. Görüntülerden koparılan plaka bölgesi içerisinde görüntü işleme teknikleri kullanılarak plaka karakterleri yer tespiti ve karakter tanımı yapılmıştır.



(Şekil 5.4) Plaka karakterlerinin ayrıştırılması

Bozuk görüntüleri netleştirmek için histogram eşitleme uygulanmıştır. Histogram eşitleme renk değerleri düzgün dağılımlı olmayan resimler için ideal bir iyileştirme yöntemidir.

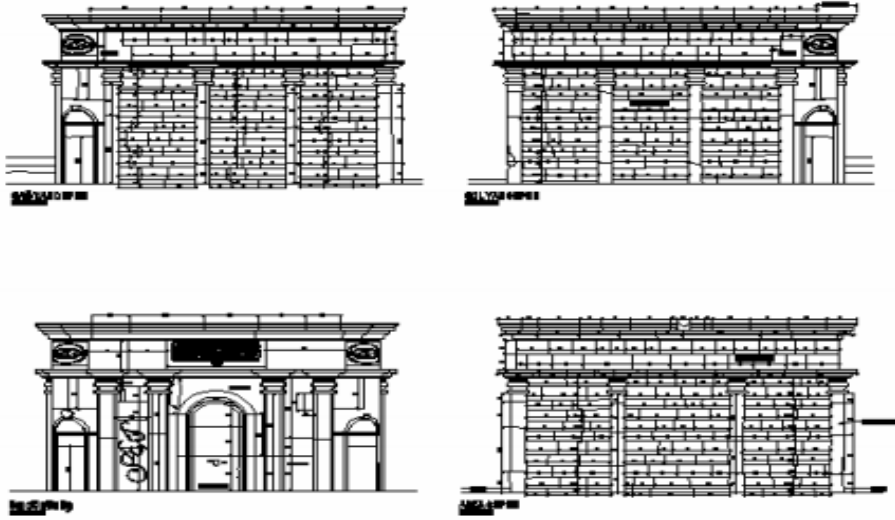
Karakter tespiti ve doğrulanması için plakalarda kullanılan 22 harf ve 10 rakamını tanımaya eğitilmiş yapay sinir ağları kullanılmıştır. Görüntü işleme teknikleri ve yapay sinir ağları kullanarak geliştirilmiş plaka tanıma sisteminde ortalama %89 oranında tanıma başarısı elde etmişlerdir.

### **5.3 Tıp Alanında Uygulamaları**

Görüntü işleme, tıp alanında kanserli hücrelerin tespiti, doku analizi ve patolojik araştırma gibi medikal bazı önemli alanlarda sıkça kullanılmaktadır. En yaygın kullanımı, tıbbi görüntülerin incelenmesi ve görüntülerdeki nesnelerin ayrıştırılmasıdır. Bununla birlikte dünyada tıbbi görüntüler için kullanılan görüntüleme standartının ismi DICOM'dur. İçeriğinde sadece görüntülenen bölge ile ilgili renk kodlaması değil aynı zamanda hastanın bilgilerini ve raporları ihtiva eder. Kullanılan görüntüleme cihazına göre farklılık gösterse bile genellikle yüksek radyometrik çözünürlüğe sahip görüntü bilgilerini barındırır.

### **5.4 Mimari Uygulamaları**

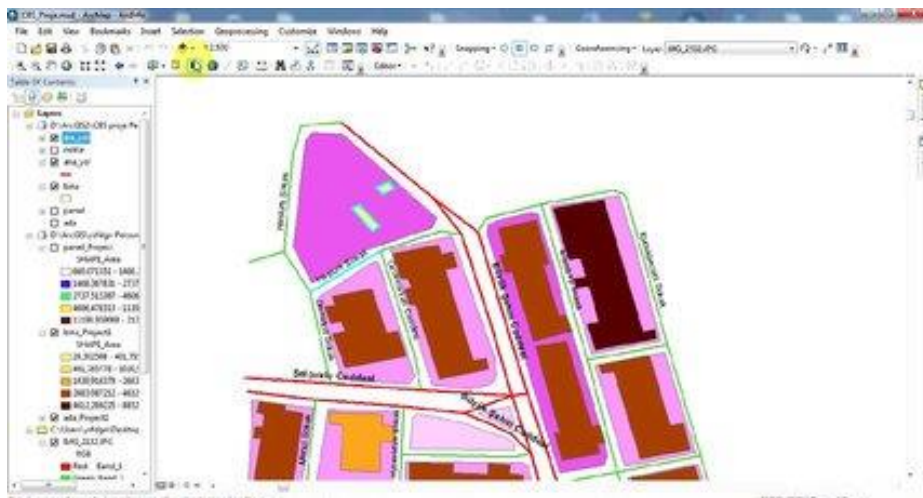
Yapımı tamamlanmış günümüz yapılarını ve eski dönemlerden kalan tarihi arkeolojik yapıların onarılması ve modellerinin sayısal olarak saklanması günümüz teknolojisinde mümkün olmaktadır. Özellikle eski mimarilerin onarılmasında sayısal ortama aktarılma süreci, yapıların detaylı incelenmesinde gerekli bir aşama olarak kullanılmaktadır.



(Şekil 5.5 Geliştirilen mimari modelleme sistemi ile nesne tanıma)

### 5.5 Harita ve Jeodezi Uygulamaları

Bugünkü bilgisayar teknolojisi olanakları, arazi ile bağlantılı her tür bilginin saklanabildiği coğrafi bilgi sistemi sunmaktadır. Yerin şekil, büyüklük alanlarını zamana bağlı olarak değişimlerini inceleyen ve bunları üç boyutlu bir koordinat sisteminde tanımlamaya yönelik bilimsel çalışmalar yürüten bilim dalına Jeodezi denmektedir. Uzaktan algılamada kullanılan görüntü işleme uygulamaları sayısal görüntü işleme uygulamalarının değişik adımlarını içermektedir. Uzaktan algılama görüntülerine uygulanan işlemler genel olarak ön işlemler, görüntü iyileştirme, görüntü zenginleştirme ve son olarak da görüntü sınıflandırma olarak ele alınabilir.



(Şekil 5.6 Harita ve jeodezi uygulamaları)

## 6. GENEL SONUÇLAR

Görüntü işleme ile beş eksenli robot kol projesinin mekanik aksamından elektronik devre tasarımı, kullanılan malzemeler ve dikkat edilmesi gerekenler tüm aşamaları bitirme tezinde açıklanmaya çalışılmıştır. Robot kol çalışmasında ilk denememiz olan bu projeyi tamamen hatasız yaptığımız söylenemez elimizdeki imkânlar dâhilinde şuan bu şekle getirilen proje daha da ilerletilebilir. Gözden kaçırılan noktalar ise servo motor seçimleridir.

Tork değerlerinin yeterli olduğu fakat plastik dişli servo motorlar kullanıldığı için sistem ağırlığının yoğun olduğu merkez ve alt kolları taşıyan servo motorlarda ısınma ve titreme problemleri yaşanmıştır.

Bu noktalardaki servo motorlar metal dişli servo motorlarla değiştirilerek sorun giderilmeye çalışılmıştır. Sorun baya giderilmiş olsa da servoların kalitesinden dolayı ufak titremeler görülmeye devam etmektedir.

Problem yaşanan bir başka nokta ise servo-dc motorların besleme problemidir. Servo motorların her biri 500mA akım çekmektedir. Servo motorlardan sadece birisi sürüldüğünde bir problem yaşanmayacaktır fakat sistemde 4 Servo motor ve 2 Dc motor kullanıldığı için minimum 3A e kadar akım ihtiyaç vardır her motor devreye girdiğinde akım probleminden dolayı motorlarda titreme olayı gözlenebilir. Kullanılan servo ve dc motora göre ihtiyaç duyulan güç kaynağı hesaplanarak ona göre bir besleme yöntemi seçilmelidir.



## KAYNAKLAR

Devrim Çamoğlu – Elektronik – 2013 Baskı –Dikeyksen Yayınları

Serkan Ayyıldız - Kendi Robotunu Kendin Yap - 2008 Baskı - Altaş Yayınları

Devrim Çamoğlu – Arduino – 2014 Baskı –Dikeyksen Yayınları

Suat Özdemirci – Robotik – 2014 Baskı –Dikeyksen Yayınları

Devrim Çamoğlu – Mikrodenetleyiciler ile Elektronik – 2013 Baskı–Dikeyksen Yayınları

Coşkun Taşdemir – Arduino – 2012 Baskı – Dikeyksen Yayınları

Coşkun Taşdemir – Arduino Uygulama Kitabı – 2014 Baskı – Dikeyksen Yayınları

<http://www.instructables.com/>

<http://arduinoturkiye.com/>

<http://pighixxx.com>

<http://tr.wikipedia.org/wiki/Arduino>

<http://www.mathworks.com/>

## ÖZGEÇMİŞ

Burak KAPUSIZ Ankara da doğdu. İlk ve Orta öğrenimini İstanbul Kuleli Okulunda tamamladı. Kastamonu Kuzeykent Süper Lisesinden mezun olduktan sonra 2012 yılında Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

### İletişim Bilgileri

Adres: Zafer Mahallesi No:14  
Bahçelievler/İSTANBUL  
E-posta: burak\_kapusuz\_@hotmail.com  
Tel: 541 390 80 19

Erhan ERDOĞAN 1991'de Kayseri'nin Kocasinan ilçesinde doğdu; ilköğrenimini Necip Fazıl İlköğretim Okulu'nda ve ortaöğrenimini yine aynı şehirde Kadir Has Ortaokulu'nda tamamladı; Samsun Endüstri Meslek Lisesi, Bilişim Teknolojileri Bölümü'nden mezun olduktan sonra 2012 yılında Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

### İletişim Bilgileri

Adres: Saitbey Mahallesi 100.Yıl Bulvarı No:94  
İlkadım / SAMSUN  
E-posta: eerhanerdoğan@hotmail.com  
Tel: 538 541 86 65

Kamil ŞİMŞEK 1993'de Konya'nın Akşehir ilçesinde doğdu; ilk ve orta öğrenimini 75. Yıl İlköğretim Okulun'da tamamladı; Seyyid Mahmut Hayrani Anadolu Lisesinden mezun olduktan sonra 2012 yılında Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

### İletişim Bilgileri

Adres: Alanyurt Mahallesi No: 37  
Akşehir/ KONYA  
E-posta: simsekamil@gmail.com  
Tel: 534 479 81 62