

**DOKUZ EYLÜL ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ÜÇ SERBESTLİK DERECELİ ROBOT KOLUNUN**  
**POZİSYON KONTROLÜ**

**Gökmen KATIPOĞLU**

**Ağustos, 2013**

**İZMİR**

# **ÜÇ SERBESTLİK DERECELİ ROBOT KOLUNUN POZİSYON KONTROLÜ**

**Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü**

**Yüksek Lisans Tezi**

**Mekatronik Mühendisliği Ana Bilim Dalı Mekatronik Mühendisliği**

**Programı**

**Gökmen KATIPOĞLU**

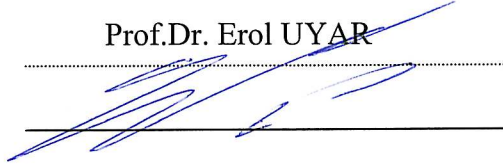
**Ağustos, 2013**

**İZMİR**

## YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

GÖKMEN KATIPOĞLU, tarafından PROF. DR. EROL UYAR yönetiminde hazırlanan "3 SERBESTLİK DERECELİ ROBOT KOLUNUN POZİSYON KONTROLÜ" başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Prof.Dr. Erol UYAR



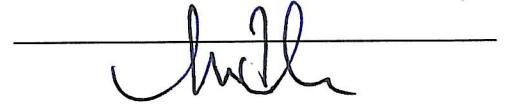
Yönetici

Doç. Dr. Zeli Kural



Jüri Üyesi

Doç. Dr. Musa Alcu



Jüri Üyesi

Prof.Dr. Ayşe OKUR

Müdür

Fen Bilimleri Enstitüsü

## TEŐEKKÜR

Öncelikle yüksek lisans eğitimim ve tez çalışmalarım sırasında bana her türlü desteęi sunan, yeni bilgiler öğrenmem ve yeni çevreler edinmem hususlarında desteklerini esirgemeyen sayın hocam Prof. Dr. Erol UYAR 'a teşekkürü bir borç bilirim.

Tez çalışmalarım ve yüksek lisans eğitimim sırasındaki çalışmalarımda bana yol gösteren değerli arkadaşlarım Ozan ENGİNOĞLU, Bertan KARAHODA, Lütfi MUTLU, Nail AKÇURA ve değerli kardeşlerim Ege Üniversitesi Elektrik-Elektronik Mühendislięi 3. Sınıf öğrencileri Mert ALTINTAŐ ve Onur DURAN 'a sonsuz teşekkürlerimi sunarım.

Son olarak eğitim hayatım boyunca bana maddi ve manevi desteęini esirgemeyen babam Mahmut KATIPOĞLU 'na teşekkürlerimi bir borç bilirim.

Gökmen KATIPOĞLU

# ÜÇ SERBESTLİK DERECELİ ROBOT KOLUNUN POZİSYON KONTROLÜ

## ÖZ

Robotik; verilen görevi yerine getirebilen fiziksel etkinliklerde bulunabilen, karar verebilen, insanların yerini alabileceği düşünülen araçları inceleyen bilim dalıdır. Aynı zamanda robotik, temel mühendislik dallarını ortak paydada bir araya getiren modern teknolojinin alanıdır. Robotların yapısını ve uygulama alanlarını anlamak için temel mühendislik dallarının bilgi ağını kullanmak gerekir (Bu mühendislikler; makine, elektrik-elektronik, bilgisayar ve endüstri mühendislikleridir). Robotik bu mühendislik dallarının beraberce geliştirdiği ve ilerlettiği çalışmaların ortak paydasıdır.

Bir robotun tasarlanmasında ve yapılmasındaki aşamalara baktığımızda, yukarıda yer alan mühendislik dallarının çalışmalarının ortak sonucu olduğunu görebiliriz.

Genel olarak düşündüğümüzde robot; önceden belirlenmiş görevleri yerine getirebilen mekatronik bir cihazdır. Bu tez çalışmasında da robotun (robot kolun) genel tasarımı ve kademeli olarak gelişimi ele alınmıştır.

Bu robot kol yazılan programa göre farklı amaçlar için de kullanılabilir. Örneğin; bir tasnif robotu olarak çalıştırılabilir. Mekanik yapısı bu iş için uygundur. Üzerinde bulunan beş farklı servo motor, robotun istenilen konuma hareketini kolaylaştırmıştır. Çünkü servo motorların en büyük avantajı; sürücü tarafından belirtilen açı miktarı kadar hareket etmesi ve hassasiyetinin yüksek olmasıdır. Yapılan çalışmada robot kolun ters kinematik denklemler sayesinde uzuvlarının istenilen açıda istenilen yere hareket etmesi hesaplanarak sağlanmıştır. Mikro-denetleyici olarak Arduino işlemcilerinden Atmega 2560 versiyonu tezin gerçekleşmesinde kullanıldı. Robot kolun farklı amaçlarda kullanılmasına izin veren ve farklı programların yazılmasını destekleyen farklı iletişim protokollerinin bulunması Atmega 2560'ın sağladığı en büyük avantajıdır. Kartın işlem yapma hızı robotun doğru şekilde çalışmasını sağlayıp uzuvların istediğimiz zamanda isteğimiz görevi yerine getirmesine olanak vermiştir. Örneğin; robotu tasnif robotu olarak denendiğinde, banttın geçen malzemeleri doğru anda banttın alıp belirlediğimiz yere

bırakmıştır. Home pozisyonuna geri dönerek yeni gelen malzemeler için beklemiştir. Bu kadar hassas işlem yapabilmesini sağlamak için servo motorlardan faydalanılmıştır. Bu motorların Atmega 2560 ile doğru bir biçimde haberleştirerek robotun doğru bir biçimde çalışması sağlanmıştır.

**Anahtar Kelimeler:** İleri kinematik, servo motor, arduino, mikro-denetleyici

# **POSITION CONTROL OF A THREE DEGREE OF FREEDOM ROBOT MANIPULATOR**

## **ABSTRACT**

Robotics is the area of science which analyzes appliances which are able to perform the desired task, to do physical acts and are supposed to replace man. Meanwhile Robotics is the shared technological area with all basic engineering fields. In order to have an understanding of robots' structure and fields of application, it is essential to use data network of basic engineering ( including: mechanical, electrics and electronics, computer and industrial engineering). Robotics is the common outgrowth progressed and developed by all these engineering fields.

In analyzing different stages of designing and making a robot one can see that it is the shared result of all engineering fields mentioned above.

Generally speaking, robot is a mechatronic device able to perform pre-determined tasks, in this study gradual development and general design of a hand robot is investigated.

This hand robot can be used for different intents. For instance it can perform as a classification robot. its mechanic structure allows this performance. Five different servo-assisted motors facilitate its movements in any particular way. Since biggest advantage of servo-assisted motors are the ability to make movements in the set angle by controller and high sensitivity. In this study , the movements of limbs in desired angles to desired points are computed using reverse kinematic equations. Atmega 2560 version of Arduino processors is applied as micro-processor. the main advantage of Atmega 2560 is allowing the robot hand to be used for different intents. also existence of different communication protocols which enables writing of different programs is another advantage. process speed of card makes it possible for the robot to perform better and lets the limbs to do desired action at the desired moment. for example if we decide to use it as a classification robot, it will be able to take out the material on the conveyor at the right moment and put them in the pre-

determined spot. In the Home position it waits for newly arriving material by going backward. servo-assisted motors are used to make delicate actions like this possible. these motors are corresponded properly by Atmega 2560 to allow proper performance for robot.

**Keywords:** Forward kinematic, servo motor, arduino, microcontroller



## İÇİNDEKİLER

	<b>Sayfa</b>
YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU .....	ii
TEŞEKKÜR.....	iii
ÖZ .....	iv
ABSTRACT.....	vi
ŞEKİLLER LİSTESİ .....	x

### **BÖLÜM BİR - ROBOTİĞİN TANIMI VE TARİHİ GELİŞİMİ..... 1**

1.1 Robotiğe Giriş .....	1
1.2 Genel Tanımlamalar.....	3
1.2.1 Konum .....	3
1.2.2 Yönelim .....	4
1.2.3 Genel Dönüşümler .....	5
1.2.4 Ardışık Dönüşümler.....	7
1.3 İleri Kinematiğin Ayarlanması.....	9

### **BÖLÜM İKİ - ELEKTRİK MOTORLARI ..... 12**

### **BÖLÜM ÜÇ -MİKRO-DENETLEYİCİ..... 13**

3.1 Mikro-denetleyicilerin Tanımı .....	13
3.2 Mikro-denetleyici içindeki RISC Mimarisi .....	14
3.3 Atmel Mikro-denetleyiciler.....	14

3.4 Arduino .....	15
3.4.1 Donanımı .....	16
3.4.2 Yazılımı .....	16
3.4.3 Geliştirme Ortamı (Arduino) .....	16
3.4.4 Arduino Bootloader (Optiboot) .....	16
3.4.5 Arduino Kütüphaneleri .....	17
<b>BÖLÜM DÖRT -PROGRAMIN İŞLEYİŞİ.....</b>	<b>18</b>
4.1 Program Algoritması.....	18
4.2 Robot Kolunun Görünümü.....	19
<b>BÖLÜM BEŞ -SONUÇ.....</b>	<b>22</b>
<b>KAYNAKLAR .....</b>	<b>23</b>
<b>EKLER.....</b>	<b>24</b>

## ŞEKİLLER LİSTESİ

	<b>Sayfa</b>
Şekil 1.1 Koordinat düzlemlerinin birbirine göre yönelimi .....	4
Şekil 1.2 Bir noktanın başka bir koordinat sistemine göre ifadesi.....	5
Şekil 1.3 Bir noktanın başka bir koordinat sistemine göre ifadesinde ardışıl dönüşüm metodu.....	8
Şekil 1.4 Çalışmada kullanılan manipülatörün modellenmesi.....	9
Şekil 2.1 Elektrik motorlarının sınıflandırılması .....	122
Şekil 3.1 Mikro-denetleyici yapısı ve bileşenleri.....	13
Şekil 3.2 Arduino Atmel mikro-denetleyicili hazır kartları .....	15
Şekil 4.1 Robot kolunun nesneyi tanınması.....	19
Şekil 4.2 Robot kolunun nesneyi kavraması .....	20
Şekil 4.3 Robot kolunun nesneyi taşıması .....	21

## BÖLÜM BİR

### ROBOTIĞIN TANIMI VE TARİHİ GELİŞİMİ

#### 1.1 Robotiğe Giriş

Robotik, günümüzde ağırlıklı olarak makine mühendisliği, elektronik mühendisliği ve mekatronik mühendisliği olmak üzere diğer mühendislik dallarının (uzay mühendisliği, uçak mühendisliği ve kontrol mühendisliği) ortak çalışma alanıdır. Robotik, isminin de beraberinde getirdiği robot teknolojisini geliştiren ve gerekli görülen alanlarda kullanılmasını sağlayan robot adını verdiğimiz elektro-mekanik cihazları üreten ve geliştiren bilim dalıdır.

Robotiğin tarihi gelişimine bakacak olursak Ktesibios robotik biliminin babası olarak kabul edilir; kendisi çağını aşan pek çok çalışmalar yapmıştır. Ktesibios kendinden öncede bilinen su saatini geliştirmiştir. Ktesibios'un su saati içine belli bir ritimle su dolan bir depodan oluşuyordu, depoya su doldukça, içindeki duba yükseliyordu. Dubanın ucundaki iğne ise, bir silindirin üzerine bu yükselmeyi işaretliyordu.

Görüldüğü gibi robotik mekanik temelli olarak başlayıp geçen yüzyıllar sonunda elektriğin bulunması ve elektrikle çalışan cihazların üretilmesi ve günümüzdeki çalışmalar ile beraber gelişmeye devam etmektedir. Gelişen teknolojiyle beraber üretilen robotların işlevleri geliştirilmiş olup bu sayede günlük hayatımızda daha fazla yer edinmeye başlamıştır. Çünkü robotların programda yazan görevleri direk olarak yerine getirmesi yapılan işlerde hatayı azaltmıştır. Gündelik birçok işimizi veya seri üretim yapılmak istenen her üretim yerinde robotlar kullanılmaktadır. Robot denildiğinde akla ilk gelen aracın insan gibi görünüşe sahip olan ve insan gibi hareket eden günümüzde popüler olan Honda Asimo gibi "Humanoid" robotlar olsa da robotun tanımı da işlevi de düşünülenenden farklıdır.

Robot; direkt insan kontrolünde çalışabildiği gibi programlanabilen verilen görevleri yerine getirebilen elektronik aksamı mekanik bir cihazdır. Robotlar günümüzde genellikle endüstriyel üretim yapan alanlarda kullanılır. Özellikle de

otomotiv endüstrisinde çok sayıda farklı işlevleri olan robotlar kullanılmaktadır. Genel olarak buradaki robotlar manipülatör şeklinde olan işlev yapan robotlardır. Uzuvlardan oluşan bu robotlar kaynak, parça montajı ve boya gibi işlevlerde kullanılmaktadır.

Robot kelimesi, ilk kez Karel Čapek 'in 1920 yılında yazdığı R.U.R.-Rossum's Universal Robots adlı (ve Türkçe'ye Halid Fahri tarafından R.U.R. - Alemşumul Suni Adamlar Fabrikası adıyla çevrilip, Osmanlıca olarak 1927 yılında Devlet Matbaası tarafından da yayınlanan) eserinde yer almış ve daha sonra tüm dünyada kullanılmaya başlanmıştır [1]. Karel Čapek 'in R.U.R. adlı oyununda otonom ve mekanik, ayrıca insanca duyguları bulunmayan yaratıklar olarak kullanılan robot kelimesi, daha sonra ki dönemlerde birçok bilim kurgu romanına konu olmuştur. Isaac Asimov'un robot serisi ile teknolojik açıdan tutarlı robot kavramını yaratmış ve robotların yapılma amacının insana hizmet etmek olduğunu belirtmiştir, bir robotun kendi amaçlarını insanların amaçlarına hiçbir zaman tercih edemeyeceğini koyduğu üç Robot Yasası'yla belirlemiştir. Bu robot yasaları şu anda insanlar ile robotlar arasında ki ahlaksal ve hukuksal ilişkinin temelini oluşturmaktadır.

#### Üç Temel Robot Etik Yasası;

1- Bir robot hiçbir biçimde insanoğluna zarar veremez; pasif kalmak suretiyle zarar görmesine izin veremez.

2- Bir robot kendisine insanlar tarafından verilen emirlere ilk kural ile çelişmediği sürece uymak zorundadır.

3- Bir robot birinci ve ikinci kurallar ile çelişmediği sürece kendi varlığını korumak zorundadır.

Robotlar sensörler ile çevresini algılayabilen, algıladıklarını yorumlayabilen, bunun sonucunda karar verebilen, karar sonucuna göre davranan, eylem olarak hareket organlarını çalıştıran veya durduran elektronik aygıttır. Yapılan tanıma göre paralel port ile bilgisayara bağlı ve klavyeden kontrol edilebilen bir araba robot sayılmamaktadır. Çünkü kendisi tek başına karar vermemekte ve bizim

yönlendirmelerimiz ile klavyeden kontrol edilmektedir. Ancak aynı araba sensörleri çalışıp algıladıklarını yorumlayabilmek için bilgisayarın mikroişlemcisini kullanabilseydi, kendi kararlarını alabildiği için, bizden bağımsız olarak davranabilen bir robottur sayılmış olacaktır.

Sanayi robotları konusunda ilk çalışma ve uygulamalar ABD de 1950’li yıllardan sonra başlamıştır. Bu yıllarda Denavith ve Hartenberg ’in geliştirdiği kinematik Hesaplama yöntemleri yanı sıra 1960 lı yıllarda Stanford Robot Enstitüsünde yapılan araştırmalar sonucu değişik tiplerde sanayi robotları geliştirilmiş ve sanayide uygulanmıştır. ABD ‘de Cincinnati Milcron, Westinghouse, Unimation firmalarının yanı sıra Japonya’da da Hitachi, Mitsubishi, Fanuc gibi firmalar kendi robotlarını geliştirerek üretimlerinde kullanmıştır. Bu şekilde Puma, Scara, Yasukawa – Motoman gibi kendine has yapısı olan manipülatör türleri ortaya çıkmıştır.

Genelde yere sabit olan ve belli bir “Çalışma Uzayına” sahip manipülatör şeklindeki açık zincir kinematikle çalışan robotlardan başka yerini sürekli değiştirerek operasyonlar yapabilen hareketli “Mobil Robotlar” da son yıllarda popüler hale gelmiştir.

Bu çalışmada sanayi de kullanılabilir şekilde üç serbestlik dereceli kartezyen tip bir robot manipülatörün tasarımı, ilk örnek imalatı ile kinematik denklemlerinin çıkarılması ve bilgisayar destekli kontrolü ele alınmıştır. Aşağıda bazı robotik tanımlamalar yapıldıktan sonra tasarlanan robotun kinematik hesapları ile programlaması açıklanacaktır.

## **1.2 Genel Tanımlamalar**

### ***1.2.1 Konum***

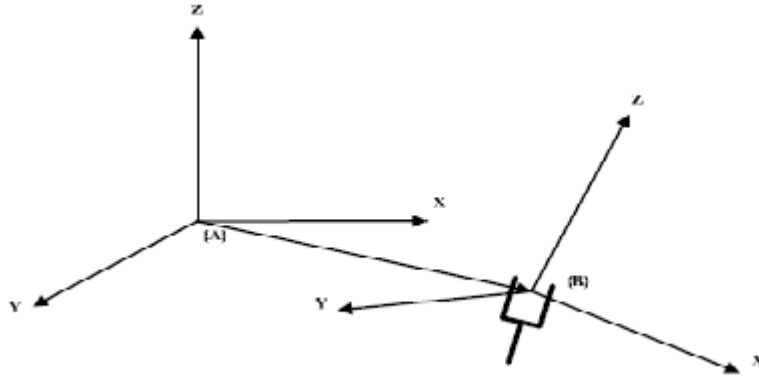
Bir nokta, koordinat sistemi tanımlamak suretiyle evrensel çerçeve içerisinde herhangi bir yere konumlandırılabilir. Üç boyutlu uzayda, bir nokta bu koordinat sistemlerinin merkezine göre tanımlanmış  $3 \times 1$  boyutlu bir vektörle gösterilebilir.

Örneğin evrensel çerçeve içerisinde buluna bir P noktasının {A} koordinat sistemine göre konumu  ${}^A P$  şeklinde bir vektörle gösterilir.  ${}^A P$ , P noktasının {A} koordinat sisteminin merkezine uzaklığını, (x, y, z) eksenlerinde sayısal olarak tanımlar.  ${}^A P$  vektörü matematiksel olarak aşağıdaki denklemdeki gibi gösterilir.

$${}^A P = \begin{bmatrix} {}^A p_x \\ {}^A p_y \\ {}^A p_z \end{bmatrix}$$

### 1.2.2 Yönelim:

Üç boyutlu uzayda, bir noktanın herhangi bir koordinat sistemine göre konumunun yanında yönelimi de tanımlanır. Yönelim, bir koordinat sisteminin başka bir koordinat sistemine göre dönme miktarıdır ve 3x3 boyutlu bir matrisle ifade edilir.



Şekil 1.1 Koordinat düzlemlerinin birbirine göre yönelimi

Şekil 1.1'deki gibi {B} koordinat sisteminin, {A} referans koordinat sisteme göre yönelimini (dönme miktarını) aşağıdaki gibi ifade edebiliriz.

$${}^A R_B = \begin{bmatrix} {}^A \hat{X}_B & {}^A \hat{Y}_B & {}^A \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}$$

Bu noktadan hareketle x, y ve z eksenlerindeki dönme dönüşümleri aşağıdaki gibidir:

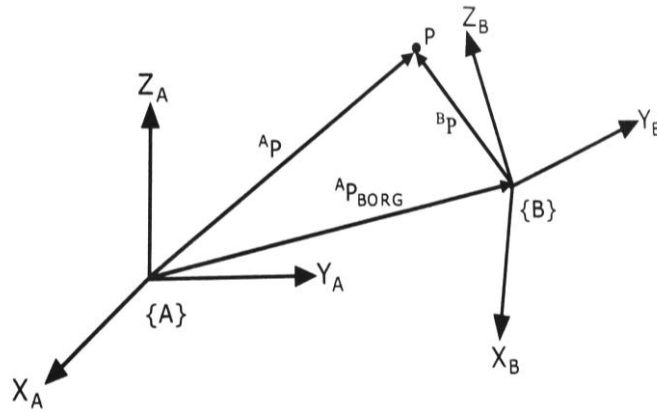
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### 1.2.3 Genel Dönüşümler

Bir koordinat sisteminde yer alan bir noktanın başka bir koordinat sistemine göre ifadesi aşağıdaki gibidir.



Şekil 1.2 Bir noktanın başka bir koordinat sistemine göre ifadesi



Yukarıdaki Şekil 1.2, hem yönelimleri hem de merkezleri farklı noktalarda bulunan iki koordinat sistemidir.

$${}^A P = {}^A_B R {}^B P + {}^A P_{BORG} \quad (1.1)$$

Eğer {A} koordinat sistemi ile {B} koordinat sisteminin yönelimleri aynı merkezleri farklı noktalarda ise bu durumda  ${}^A_B R = I$  olur ve denklem  ${}^A P = {}^A_B R {}^B P + {}^A P_{BORG}$  şeklinde ifade edilir. Aynı şekilde, eğer {A} koordinat sistemi ile {B} koordinat sisteminin merkezleri çakışık yönelimleri farklı ise  ${}^A P_{BORG} = 0$  olur ve denklem  ${}^A P = {}^A_B R {}^B P$  şeklinde ifade edilir.

Genel olarak  ${}^B P$  ile  ${}^A P$  arasındaki ilişki bir matrisle ifade edilmek istenir. Bunun için denklem

${}^A P = {}^A_B T {}^B P$  şeklinde yeni bir dönüşüm matrisi tanımlanır. Bu denklem daha açık bir şekilde yazılırsa aşağıdaki matris denklemi olarak ifade edilebilir:

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R & {}^A P_{BORG} \\ 000 & 1 \end{bmatrix} \begin{bmatrix} {}^B P \\ 1 \end{bmatrix}. \quad (1.2)$$

${}^A P = {}^A_B T {}^B P$  denkleminde  ${}^A P$ ,  ${}^B P$  ve  ${}^A P_{BORG}$  vektörlerine 1 ve  ${}^A_B R$  matrisine 0'lar eklenerek matris çarpımının doğru yapılması sağlanmaktadır. Üstteki çarpım gerçekleşirse  ${}^A P = {}^A_B R {}^B P + {}^A P_{BORG}$  denklemi sağlanır.

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R + {}^A P_{BORG} \\ 1 \end{bmatrix} \quad (1.3)$$

${}^A P$  ile  ${}^B P$  arasındaki ilişki, içerisinde dönme matrisi ve konum vektörünün bulunduğu 4x4 boyutunda bir matrisle ifade edilebilir. Bu matrise homojen dönüşüm matrisi denir ve aşağıdaki gibi gösterilir:

$${}^A_B T = \begin{bmatrix} {}^A_B R & {}^A P_{BORG} \\ 000 & 1 \end{bmatrix}.$$

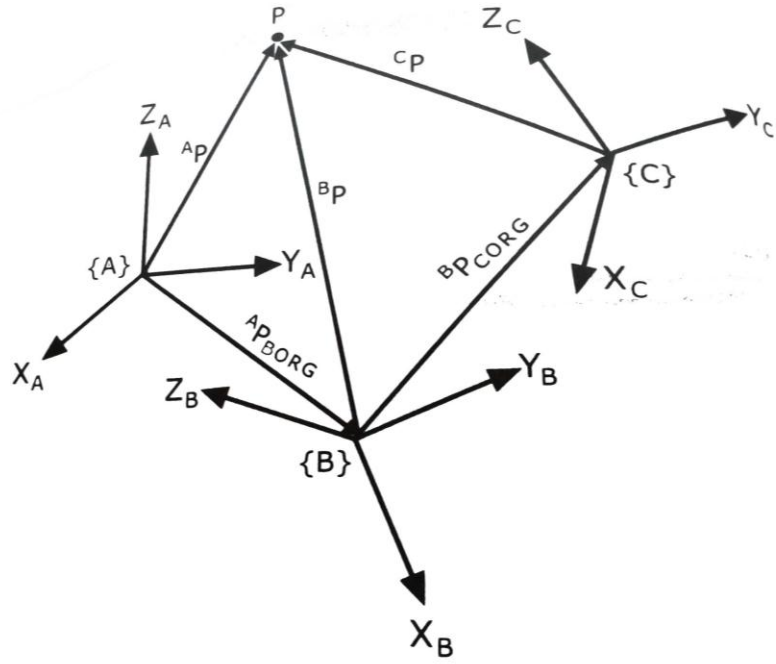
Denklemden gösterilen  ${}^A_B R$  3x3 boyutlu bir matris,  ${}^A P_{BORG}$  ise 3x1 boyutlu bir vektördür.  ${}^A_B T$  matrisini

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

şeklinde de gösterebiliriz. Denklemden,  $P_x, P_y, P_z$   ${}^A P_{BORG}$  vektörünün elemanlarını,  $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}$  ise  ${}^A_B R$  matrisinin elemanlarını temsil eder.

#### **1.2.4 Ardışık Dönüşümler**

İkiden fazla koordinat sistemini içeren sistemlerde, koordinat sistemlerinin birbirlerine göre konum ve yönelimleri ardışık koordinat sistemleri kullanılarak gerçekleştirilir. Aşağıdaki şekilde {C} koordinat sistemine göre tanımlanan P noktasının konumunu ve yönelimini {A} koordinat sistemine göre ardışık dönüşüm matrislerini kullanarak tanımlayalım.



Şekil 1.3 Bir noktanın başka bir koordinat sistemine göre ifadesinde ardışıl dönüşüm metodu

{C} koordinat sistemine göre tanımlanan P noktasının konum ve yöneliminin {A} koordinat sistemine göre ardışık dönüşüm matrisleri kullanılarak tanımlanması:

$${}^B P = {}^B C T {}^C P$$

şeklinde olur. Aynı şekilde, P noktasının {B} koordinat sistemine göre konumu  ${}^B P$  olduğundan, aynı P noktasının {A} koordinat sistemine göre konumu,

$${}^A P = {}^A B T {}^B P$$

olur. Bir önceki denklemdeki  ${}^B P$  ifadesini bu denklemde yerine koyarsak,

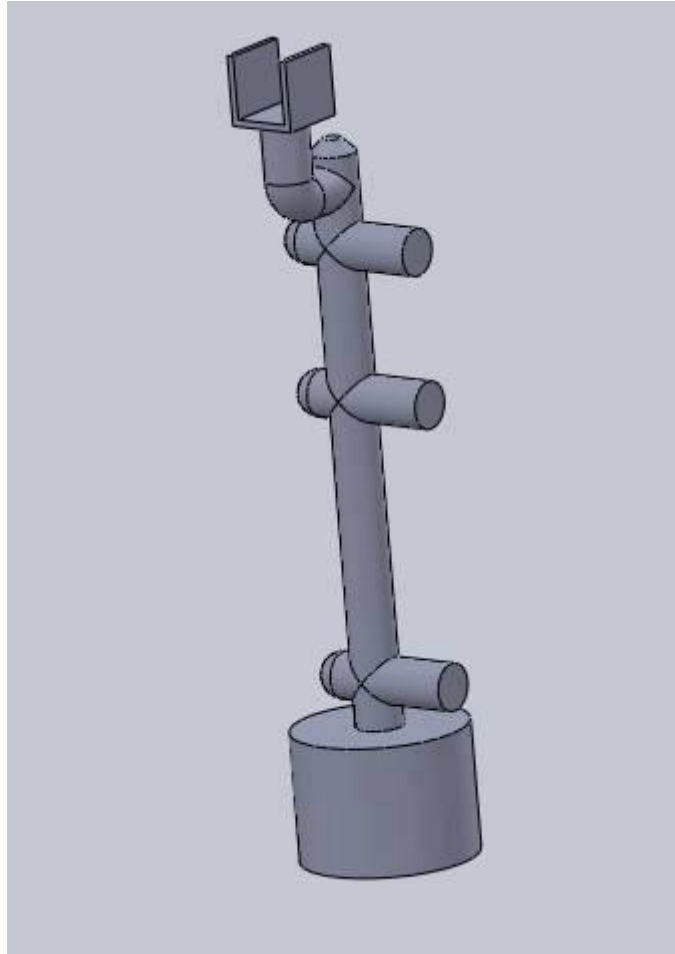
$${}^A P = {}^A B T ({}^B C T {}^C P)$$

$$= ({}^A B T {}^B C T) {}^C P = {}^A C T {}^C P \text{ olur.}$$

Denklemde  ${}^A C T = {}^A B T {}^B C T$  'dir.  ${}^A C T$  ifadesi daha açık bir şekilde aşağıdaki gibi yazılabilir:

$$\begin{aligned}
{}^A T_C &= {}^A T_B {}^B T_C = \begin{bmatrix} {}^A R_B & {}^A P_{BORG} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B R_C & {}^B P_{CORG} \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} {}^A R_B {}^B R_C & {}^A R_B {}^B P_{CORG} + {}^A P_{BORG} \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

### 1.3 İleri Kinematiğin Ayarlanması



Şekil 1.4 Çalışmada kullanılan manipulatörün modellenmesi

$${}^0 T_5 = {}^0 T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5$$

Manüplatörümüzün durum matrisleri şu şekildedir:

$${}^0_1T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} c90 & -s90 & 0 & 0 \\ s90 & c90 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(-90) & 0 & s(-90) & 0 \\ 0 & 1 & 0 & 0 \\ -s(-90) & 0 & c(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & L_3 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4_5T = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 1 & 0 & A \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c180 & -s180 & 0 \\ 0 & s180 & c180 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c90 & 0 & s90 & 0 \\ 0 & 1 & 0 & 0 \\ -s90 & 0 & c90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} 0 & 0 & 1 & L_4 \\ 0 & -1 & 0 & A \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Böylece;

$${}^0_5T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots$$

$$\begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & L_3 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & L_4 \\ 0 & -1 & 0 & A \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Manipülâtörün başlangıç konumu Şekil 1.4 teki gösterildiği dikey konumu olarak alınmıştır. Bu konumda uzuv açıları değerleri hesaplanırken  $\theta_1, \theta_2, \theta_3, \theta_4 = 0$  (sıfır) olarak alınır. Bu duruma göre hesaplanan sonuç matrisi:

$${}^0_{\xi}T = \begin{bmatrix} 1 & 0 & 0 & L_1 + L_2 + L_3 + L_4 \\ 0 & 1 & 0 & A \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ olur.}$$

Robotun boyutları ve kinematiği dönüşüm denklemleri yazıldıktan sonra ileri kinematik yöntemi ile uzuv açıları verilerek uç noktasının hesaplanan yere gidip gitmediği kontrol edilmiştir. Sonradan bu uzuv açıları referans değeri olarak verilerek robot çalıştırıldı ve uç noktalara ulaştığı görülmüştür. Teorik yoldan hesaplanan uç nokta ile referans açı verilip çalıştırılarak görülen uç noktanın birbirine yakın olduğu saptanmıştır. Robotun hareketi Arduino mikro-denetleyicisi ile yapılmıştır.

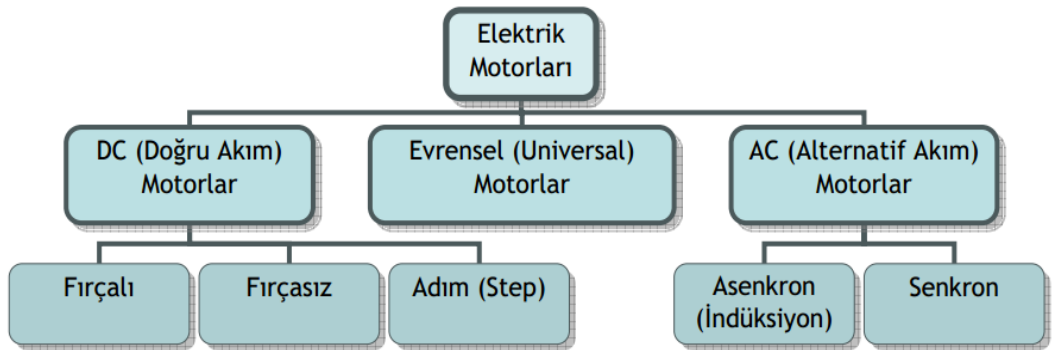
## BÖLÜM İKİ

### ELEKTRİK MOTORLARI

Günlük hayatımızda en çok ihtiyaç duyulan enerji çeşitlerinden birisi de mekanik enerjidir. Bir arabadan uçağa, fotokopi makinesinden çamaşır makinesine, oluşan hareketi algılayınca açılabilen alışveriş merkezi kapısından mutfak robotuna kadar her yerde hareket enerjisini hayatımızın hemen hemen her anında kullanıyoruz.

Elektrik motoru; şebekeden aldığı elektrik enerjisini farklı araçlar yardımı ile mekanik enerjiye dönüştüren araçtır. Her elektrik motoru biri sabit olan stator ve diğeri kendi çevresinde dönen rotor denen iki ana parçadan oluşur. Bu ana parçalar, manyetik akıyı ileten parçalar, yapı parçaları ve elektrik akımını ileten parçalar olmak üzere tekrar kısımlara ayrılır. Alternatif akım ile çalışan elektrik motorlarında rotor ve statorun manyetik akıyı ileten kısımları fuko akımlarından kaçınmak amacıyla tabakalandırılmış saçlardan yapılır. Temelde bir elektrik motoru manyetik alan değişimlerinden faydalanarak çalışır.

Elektrik motorlarını aşağıda bulunan şemada ki gibi temel gruplara ayırabiliriz. Motorlar ve türleri çok çeşitlidir ancak bu yazıda yalnızca doğru akım motorlarını (özellikle SERVO motorları) çeşitlerini inceliyoruz.



Şekil 2.1 Elektrik motorlarının sınıflandırılması

## BÖLÜM ÜÇ

### MİKRO-DENETLEYİCİ

#### 3.1 Mikro-denetleyicilerin Tanımı

Bir bilgisayar sisteminin tek bir tüm devre üzerinde birleştirilmiş modeline verilen ada mikro-denetleyici denir. Bir mikro-denetleyici içerisindeki yapı CPU (Ana işlemci), RAM (Kullanılabilir Hafıza), ROM (Programın Saklandığı bölüm), Timer (Zamanlayıcı), Adc (Analog Dijital Dönüştürücü), Pwm gibi çevre birimler, IO (Giriş Çıkış Birimleri) içinde olduğu paket halindeki bir yapıdır. Günümüzde mikro-denetleyicilerin yaygın olmasının sebebi kullanımın basit ve gerekli elektronik devresinin karmaşık olmamasındandır. Çünkü nerdeyse gerekli her şeyi içinde barındırmaktadır. Bir çok karmaşık işi basitleştirmek ya da otomatik kontrolü için mikro-denetleyiciler kullanılır. Örneğin en basit elektronik saatlerden otomatik çamaşır makinelerine, robotlardan fotoğraf makinelerine, LCD monitörlerden biyomedikal cihazlara ve endüstriyel otomasyondan elektronik bilet uygulamalarına kadar pek çok elektronik uygulamada mikro-denetleyiciler kullanım alanı bulmuşlardır.



Şekil 3.1 Mikro-denetleyici yapısı ve bileşenleri

Bir mikro-denetleyiciyle yapılan üç temel işlem;

1. Çevreden çeşitli sinyalleri almak,
2. Alınan sinyalleri önceden yüklenmiş fonksiyonlardan geçirmek,



3. Fonksiyonların sonucunu dış ortama vererek kullanışlı bir iş yapmak için kullanır.

### **3.2 Mikro-denetleyici İçindeki RISC Mimarisi**

İndirgenmiş komut setli bilgisayar (RISC), daha küçük ve basit yapıları destekleyen ve bunları yürütmek için neredeyse aynı miktarda zaman harcayan içinde bir CPU mikroişlemci bulunduran tasarımıdır. Bu fikir kodlamayı kolaylaştırmak üzere geleneksel CPU tasarımları içeren birçok özneliğin keşfi ile onları işleten programlar tarafından tanınmamasından esinlenildi. Hem de bu daha karmaşık özneliklerin çalışması için birçok işlemci devri gerektiriyor. Ek olarak CPU'nun hızı hafızayı kullanma hakkının artmasıyla bağlantılıydı. CPU'daki işlemlerin verimliliğini artırmak için yol gösteren bu bir takım teknikler aynı zamanda ulaşabildiği toplam hafıza miktarını azaltmaya da çalışıyordu. Bu aileden olan en yaygın mikroişlemciler ise ARM, Atmel, DEC Alpha, PA-RISC, SPARC, MIPS, ve IBM Power-PC diyebiliriz.

### **3.3 Atmel Mikro-denetleyiciler**

Atmel Corporation 1984'te kurulmuş bir yarıiletken üreticisidir. Ürün çeşitleri genel olarak 8051 türevleri, AT91SAM, AT91CAP, ARM tabanlı mikroişlemciler ve kendi AVR ve AVR32 mimarileri, radyo frekans cihazları, EEPROM ve Flash bellek cihazları ve birkaç uygulama spesifik üründen oluşmaktadır.

Atmel' in bünyesinde birçok farklı yapıda ve zenginlikte mikro-denetleyici barındırmaktadır.

Bunların arasından öne çıkanlar;

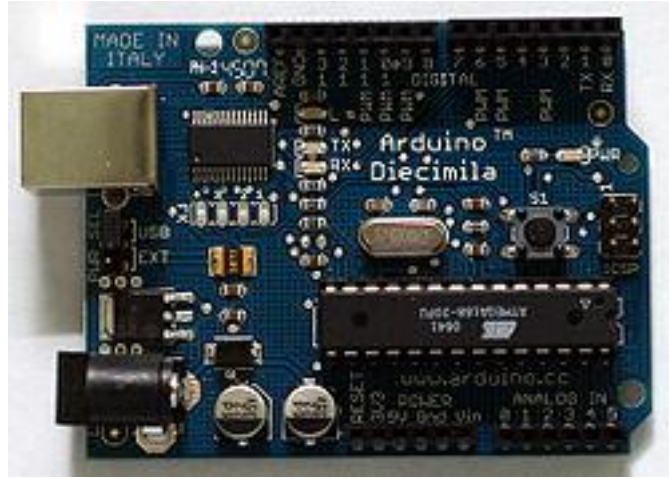
- AT91SAM, ARM/Cortex sistemi kullanılmıştır.
- AT91CAP, 32-bit ARM
- Atmel AVR, 8-bit mikro-denetleyici ailesi
- AVR32, family of 32-bit microcontrollers
- Atmega 8/16/32-bit mikro-denetleyici

- Atmel At94k
- Atmel AT89, 8-bit mikro-denetleyici

Projede kullanılan Mikro-denetleyici Atmel 2560'tır. Yapısı sayesinde yüksek performans ve düşük güçte çalışmaktadır. 256 KB ISP flash memory, 8KB SRAM, 4KB EEPROM, 86 general I/O pini, 32 kullanılabilir register, gerçek zamanlı sayıcı(RTC), 6 esnek sayıcı/zamanlayıcı, PWM, 4 farklı Usart ara-yüzü, 16 kanallı 10 bit A/D dönüştürücü ve Jtag ara-yüzü bulunmaktadır. Sistem 16 MHz kristalle ve 16 MIPS (1 İşlem için gereken clock sayısı) oranı ile çalışmaktadır. Sistemin besleme voltajı 4,5 ile 5 volt arasındadır.

### 3.4 Arduino

Arduino bilgi işlem dizgesinin değişik fonksiyonel birimleri (alt sistemleri) arasındaki iletişimi veya bu ara-yüzlere doğrudan bilgi sinyallerini gönderen ve açık kaynaklı bir tümleşik geliştirme ortam ve temel olarak bir mikro-denetleyici programlamak için gerekli bileşenleri kullanımı oldukça rahat halde bir araya getirmiş olan bir platformdur.



Şekil 3.2 Arduino Atmel mikro-denetleyicili hazır kartları

### **3.4.1 Donanımı**

Bu kartlar donanımsal olarak Atmel 'in mikro-denetleyicilerini kullanmaktadırlar. Her kartta en azından bir 5 voltluk regüle entegresi ve bir 16MHz kristal osilatör (bazılarında seramik rezonatör) bulunur. Mikro-denetleyiciye önceden bir bootloader programı yazılı olduğundan programlama için harici bir programlayıcıya ihtiyaç duyulmaz.

### **3.4.2 Yazılımı**

Arduino IDE kod editörü ve derleyici olarak görev yapan, aynı zamanda derlenen programı karta yükleme işlemini de yapabilen, her platformda çalışabilen Java programlama dilinde yazılmış bir uygulamaya sahiptir.

Arduino platformunda kullanılan temel bileşenlere bakacak olursak, bunlar:

- Arduino Geliştirme Ortamı (IDE)
- Arduino Bootloader (Optiboot)
- Arduino Kütüphaneleri
- AVR Dude (Arduino üzerindeki mikro-denetleyici programlayan yazılım)
- Derleyici (AVR-GCC)

### **3.4.3 Geliştirme Ortamı (Arduino)**

Arduino geliştirme ortamı **Processing** adlı dilin ortamına dayanıyor ve görünüş olarak da ona çok benziyor. Eğer Java biliyorsanız Arduino ortamı üzerinde istediğiniz değişiklikleri yapabilirsiniz.

### **3.4.4 Arduino Bootloader (Optiboot)**

Normalde bir mikro-denetleyiciyi programlamak için kullandığımız ürüne yönelik geliştirilmiş olan programlayıcıları kullanmamız gerekir. Bunun bir alternatifi de seri haberleşme üzerinden mikro-denetleyiciyi programlamamızdır. Tabi bunun için mikro-denetleyicimizin bir seri haberleşme modülüne sahip olması ve kendi program belleğini programlama özelliğine sahip olması gerekiyor. Bir de bu programlama

işini yapacak ufak bir programcığa ihtiyaç var. İşte bu programcığa “**bootloader**” deniliyor. Mikro-denetleyici çalışmaya başladıktan hemen sonra programlamaya başlamak için gerekli verilerin gelmesini bekliyor. Bu veriler gelmediyse doğrudan mikro-denetleyiciye yüklenmiş programı koşturmaya başlıyor. İşte Arduino ’da kullanılan bootloader ’ın ismi **OptiBoot**.

### ***3.4.5 Arduino Kütüphaneleri***

Arduino kütüphaneleri belki de Arduino ’nun bu kadar popüler olmasındaki en önemli bileşen. Bu kütüphaneler sayesinde mikro-denetleyicilerin ayrıntılı olarak bilmesek de kolayca programlayabiliyoruz. Arduino ile birlikte gelen standart kütüphaneler yanında internette gönüllüler tarafından geliştirilmiş birçok farklı kütüphane de bulmak mümkün. Arduino kütüphaneleri geliştirme ortamıyla beraber geliyor ve Arduino klasörleri içerisindeki “libraries” klasörünün altında yer alıyor. Buradaki kodlara göz atarak kütüphanelerin yapısını ve mikro-denetleyicinin modüllerinin nasıl programlandığını görebilirsiniz.

Projemizde Arduino ’nun Atmega 2560 kartı kullanılmıştır. Yazılım Arduino programında C tabanlı olarak gerçekleştirilmiştir. Arduino ’nun gerekli kütüphaneleri kullanılarak daha iyi bir kontrol sağlanmıştır.

## BÖLÜM DÖRT

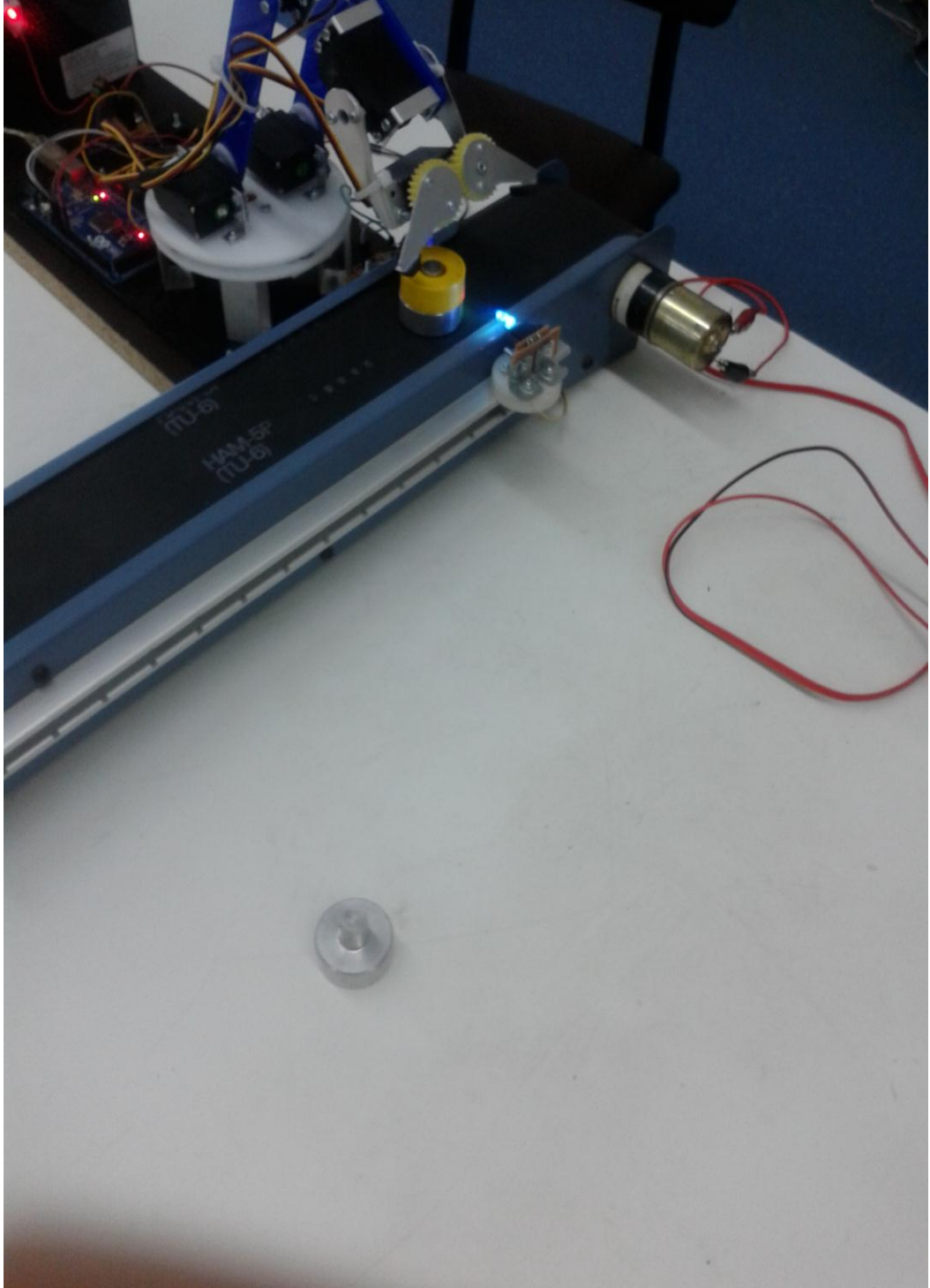
### PROGRAMIN İŞLEYİŞİ

#### 4.1 Program Algoritması

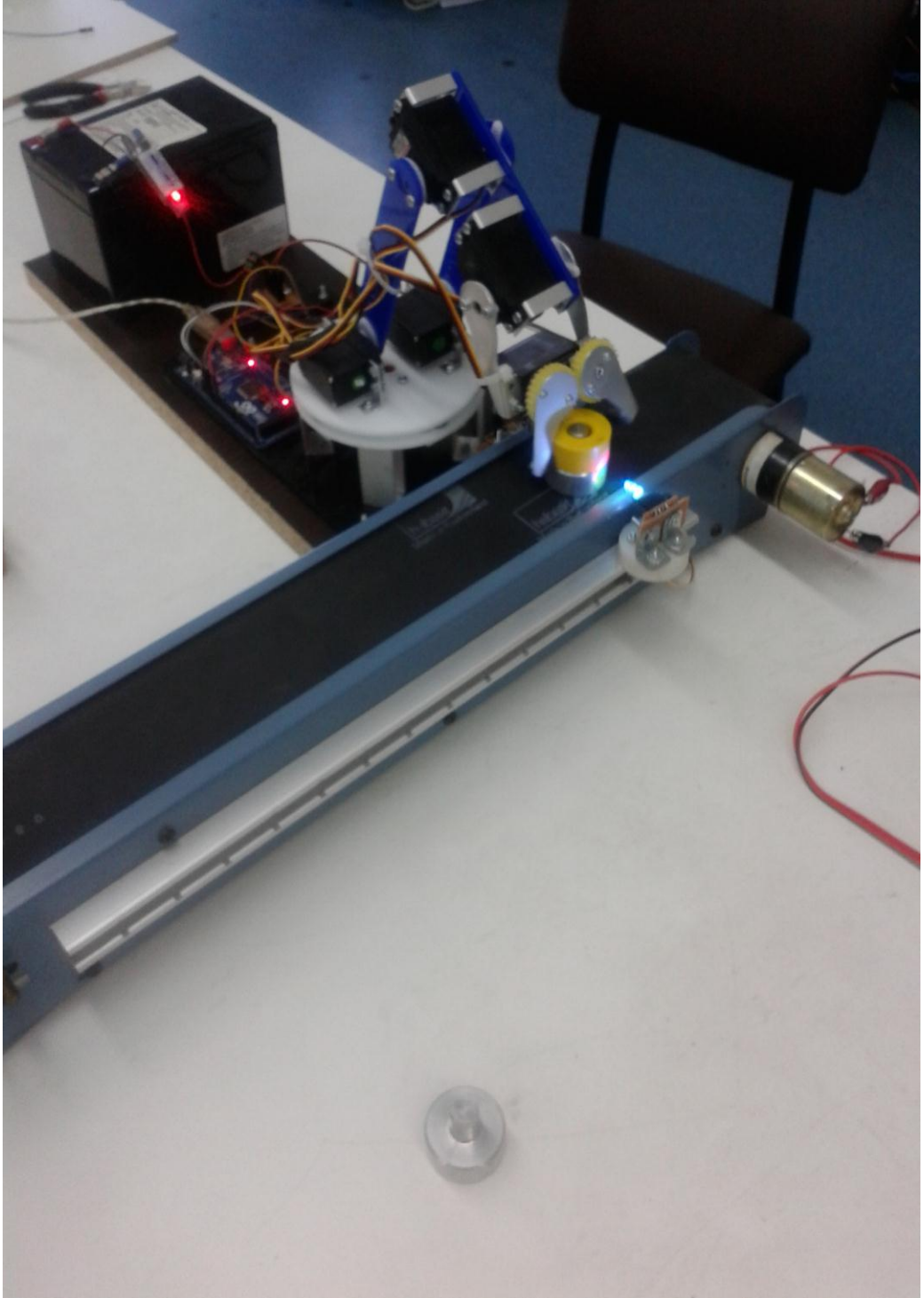
Program ilk başladığında servo kütüphanesi tanımlanmaya başlar bu kütüphanede servoların hangi timer birimini kullanacağı kaç servo çalıştırılacağı ve servo sinyalleri yer almaktadır. Daha sonra tanımlamalara geçilir Servo adlı tanımlamalarda servolar isimlendirilir. Servo x Derece ise servonun alacağı komutu belirtmektedir. Malzeme tanımlaması banttan geçen malzemenin robotun alacağı noktaya gelip gelmediğini ölçen analog bir sistemden alınan değer için tanımlanmıştır.

Loop döngüsüne gelirse bu döngüde sistem bilgisayardan başlama komutunu yani "S" harfini beklemektedir. Girilmediği sürece sensör kapalı kalacak ve servolar sabit durumunda bekleyecektir. S harfi girildiğinde programda if döngüsünün içine girilerek daha önceden inverse kinematikle konumları belirlenmiş noktaların hesapları yapılmış ve servo motorların derece kaç derece dönmesi hesaplanmıştır. Daha sonra programa girilerek yörünge oluşturmak için sıralı komutlar yazılmıştır. Böylelikle çabuk hızlı işlem yapılması sağlanmıştır. if else döngüsün içinde servo\_driver fonksiyonu çağırılmıştır. Bu fonksiyon aldığı servo açılarını servo motorun hızını ayarlayarak kendi içerisinde bir döngü ile servo motor o dereceye gelene kadar yavaş yavaş dereceyi olması gerek konuma getirip , kütüphaneyi çağırarak servoya gerekli sinyalleri göndermektedir. Daha sonra tekrar başa dönüp yeni malzemenin alım noktasına gelmesini beklemektedir. Eğer "D" harfine basılırsa da sistem baştaki haline dönmektedir.

## 4.2 Robot Kolunun Görünümü

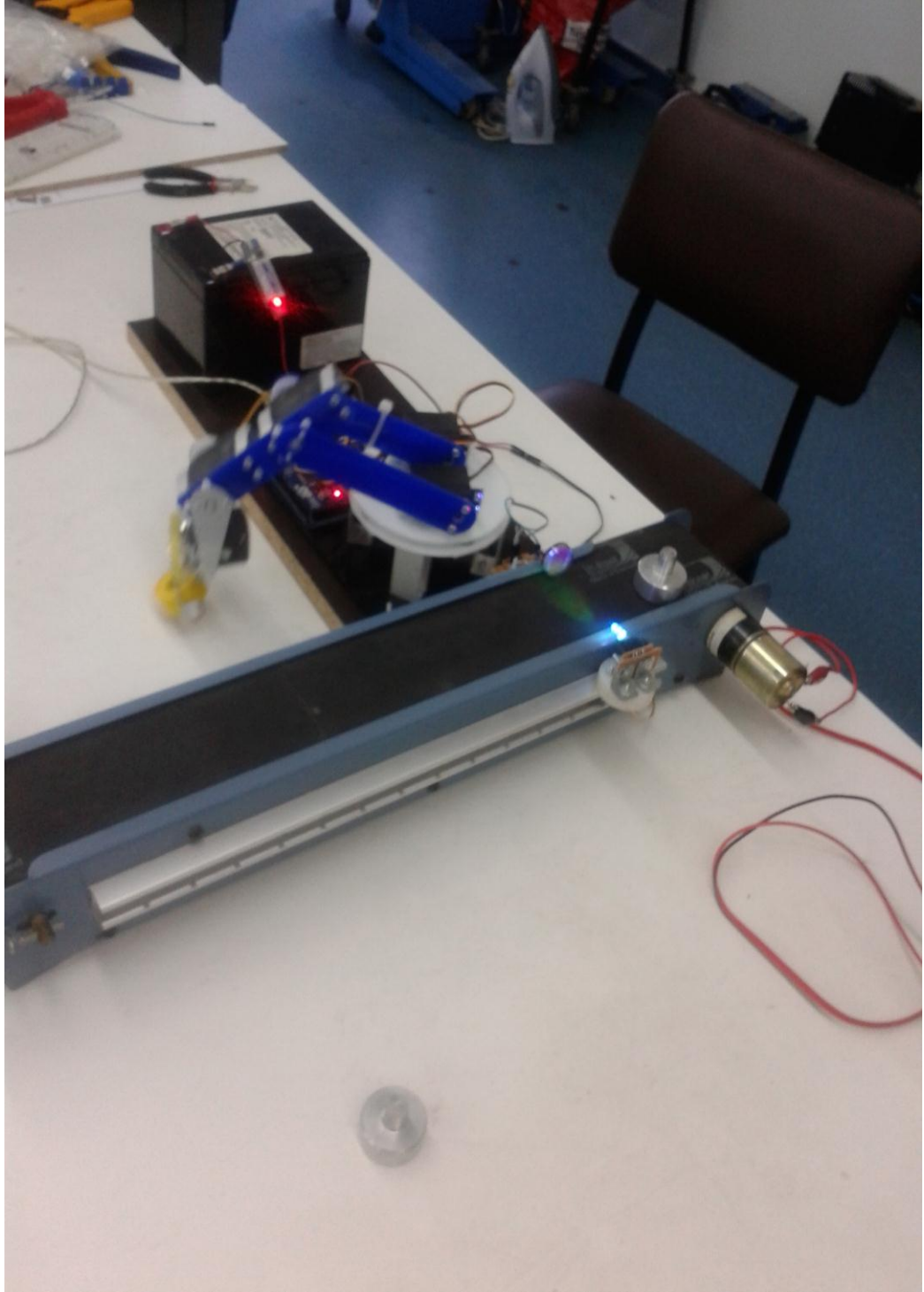


Şekil 4.1 Robot kolunun nesneyi tanınması



Şekil 4.2 Robot kolunun nesneyi kavraması





Şekil 4.3 Robot kolunun nesneyi taşıması



## BÖLÜM BEŞ

### SONUÇ

Çalışması ve kontrolü programlanan sistemin testleri laboratuvar ortamında denenmiştir. Bu amaçla klasifikasyon amaçlı yazılımı yapılan program gereği, hareketli bir bant üzerindeki ard arda gelen halka şeklindeki parçaların optik sensörün ve tutucunun (Gripper) işlem alanına girmesi ile tutma ve taşıma işlemi başlatılmıştır.

Değişik bant hızları ve parçaların hareket mesafeler değiştirilerek yapılan testler de manipülatörün oldukça başarılı sonuçlar verdiği saptanmıştır. Bu testler çekilen video gösterileri ile de sunulmuştur. Şekil.4.1, 4.2 ve 4.3 de manipülatörün değişik konumlarda ki fotoğrafları gösterilmiştir.

Robotlar sensörler ile çevresini algılayabilen, algıladıklarını yorumlayabilen, bunun sonucunda karar verebilen, karar sonucuna göre davranan, eylem olarak hareket organlarını çalıştıran veya durduran elektronik aygıttır. Bu tez çalışmasında tasarlanan robot kolu yukarıdaki tanıma uygun olarak bir üretim bandından geçen malzemeleri sensör yardımıyla algılayıp malzemenin tanımlanan kısmını alıp önceden tanımlanan noktaya bırakmaktadır.

Robotun boyutları ve kinematiği dönüşüm denklemleri yazıldıktan sonra ileri kinematik yöntemi ile uzuv açıları verilerek uç noktasının hesaplanan yere gidip gitmediği kontrol edilmiştir. Sonradan bu uzuv açıları referans değeri olarak verilerek robot çalıştırıldı ve uç noktalara ulaştığı görülmüştür. Teorik yoldan hesaplanan uç nokta ile referans açı verilip çalıştırılarak görülen uç noktanın birbirine yakın olduğu saptanmıştır. Robotun hareketi Arduino mikro-denetleyicisi ile yapılmıştır.

## KAYNAKLAR

Çamoğlu, D. (2011). *Bilgisayar kontrollü Robotik*, Ankara: Dikey Eksen

Bingül, Z. ve Küçük, S. (2009). *Robot kinematiği*, İstanbul: Birsen Yayınevi

Kurtoğlu, A. (2011). *Robot tekniği*, Basım yeri: Papatya Yayınları

Wikipedia (b.t). *Robotik*, 20 Haziran 2013, <http://tr.wikipedia.org/wiki/Robotik>

Wikipedia (b.t). *Elektrik motoru*, 20 Haziran 2013,  
<http://tr.wikipedia.org/wiki/Robotik>

wikipedia (b.t). *Servo motor*, 20 Haziran 2013,  
[http://tr.wikipedia.org/wiki/Servo\\_motor](http://tr.wikipedia.org/wiki/Servo_motor)

## **EKLER**

### **Kodlar**

#### ***Band Kontrol***

```
#include <Servo.h>
Servo servo1,servo2,servo3,servo4,servo5;
int
servo1derece=90,servo2derece=100,servo3derece=100,servo4derece=90,servo5derece=90;
int next_servo1derece=90, next_servo2derece=100, next_servo3derece=100,
next_servo4derece=90, next_servo5derece=90;
int servo_durum,secim_flag=1,acik=0,malzeme,i=146;
char buffer;

void setup() {
  servo1.attach(2);
  servo2.attach(3);
  servo3.attach(4);
  servo4.attach(5);
  servo5.attach(6);
  Serial.begin(9600);
  servo_driver(167,73,180,90,5); //1
  delay(500);
  servo_driver(78,54,97,103,5); //2
}

void loop() {
  if(Serial.available())
  { buffer=Serial.read();
  i=10;
  }
  malzeme = analogRead(0);
```

```

Serial.println(malzeme);

if(buffer=='S' or buffer=='s')
{
  if(malzeme>500)
  { i=i-10;
  delay(200);
  servo_driver(81,52,97,113,5); //3
  delay(20);
  ac_kapa(30);
  delay(20);
  servo_driver(78,54,97,103,5); //2
  //servo_driver(90,93,97,139,5); //4
  delay(20);
  servo_driver(76,53,125,106,5);
  delay(20);
  servo_driver(178,93,97,139,5); //1
  delay(20);
  servo_driver(178,38,132,139,10);
  delay(20);
  servo_driver(178,20,132,139,15);
  delay(200);
  servo_driver(178,7,146,130,15);
  delay(20);
  ac_kapa(100);
  servo_driver(167,73,180,90,5); //1
  delay(20);
  servo_driver(78,54,97,103,5); //2
  }
  }}

```

### ***Fonksiyon 1 (Aç Kapa)***

```
void ac_kapa(int next_servo5derece)
{ int islem_flag=1;
  while(islem_flag)
  {
    if(servo5derece < next_servo5derece)
      servo5derece++;
    else if (servo5derece > next_servo5derece)
      servo5derece--;

    servo5.write(servo5derece);

    delay(5);

    if(servo5derece==next_servo5derece)
      islem_flag=0;
  }
}
```

### ***Fonksiyon 2 (Servo\_Driver)***

```
void servo_driver(int next_servo1derece,int next_servo2derece,int
next_servo3derece,int next_servo4derece,int hiz)
{ int islem_flag=1;
  while(islem_flag)
  {
    if(servo1derece < next_servo1derece )
      servo1derece++;
    else if (servo1derece > next_servo1derece )
      servo1derece--;

    if(servo2derece < next_servo2derece )
```

```

servo2derece++;
else if (servo2derece > next_servo2derece )
servo2derece--;

if(servo3derece < next_servo3derece )
servo3derece++;
else if (servo3derece > next_servo3derece)
servo3derece--;

if(servo4derece < next_servo4derece)
servo4derece++;
else if (servo4derece > next_servo4derece)
servo4derece--;

servo1.write(servo1derece);
servo2.write(servo2derece);
servo3.write(servo3derece);
servo4.write(servo4derece);

delay(hiz);

if(servo1derece==next_servo1derece &&
servo2derece==next_servo2derece && servo3derece==next_servo3derece &&
servo4derece==next_servo4derece)
islem_flag=0;
}
}

```

### **Robot Kol Manuel Kontrol**

```
#include <Servo.h>
```

```
Servo servo1,servo2,servo3,servo4,servo5;
```

```

int
servo1derece=90,servo2derece=100,servo3derece=100,servo4derece=90,servo5derece=90;
int buffer_sayici=0,servo_derece,servo_no;

char buffer[11];

void setup() {
  servo1.attach(2);
  servo2.attach(3);
  servo3.attach(4);
  servo4.attach(5);
  servo5.attach(6);
  Serial.begin(9600);
  servo1.write(servo1derece);
  servo2.write(servo2derece);
  servo3.write(servo3derece);
  servo4.write(servo4derece);
}

void loop() {

  if(Serial.available())
  {
    buffer[buffer_sayici] = Serial.read();
    buffer_sayici++;
    if(buffer_sayici==2)
    {
      buffer[0]=buffer[0]-0x30;
      servo_no=buffer[0];

      buffer_sayici=0;

```

```

if(servo_no==1)
{  servo_derece=servo_driver(buffer[0],buffer[1],servo1derece);
  servo1derece=servo_derece; }
else if(servo_no==2)
{  servo_derece=servo_driver(buffer[0],buffer[1],servo2derece);
  servo2derece=servo_derece;}
else if(servo_no==3)
{  servo_derece=servo_driver(buffer[0],buffer[1],servo3derece);
  servo3derece=servo_derece;}
else if(servo_no==4)
{  servo_derece=servo_driver(buffer[0],buffer[1],servo4derece);
  servo4derece=servo_derece;}
else if(servo_no==5)
{  servo_derece=servo_driver(buffer[0],buffer[1],servo5derece);
  servo5derece=servo_derece;}

buffer[3] = Serial.read();

Serial.print("Servo1=");
Serial.println(servo1derece);
Serial.print("Servo2=");
Serial.println(servo2derece);
Serial.print("Servo3=");
Serial.println(servo3derece);
Serial.print("Servo4=");
Serial.println(servo4derece);
Serial.print("Servo5=");
Serial.println(servo5derece);
Serial.println("");
}
}

```



```
}
```

### ***Fonksiyon 1***

```
int servo_driver(int servo_no,int servo_yon,int past_servoderece)
{
  int servoderece;
  servoderece=past_servoderece;
  while(!Serial.available())
  {
    if(servoderece<180 && servo_yon=='A')
      servoderece++;
    else if (servoderece>0 && servo_yon=='D')
      servoderece--;
    else
      return(servoderece);

    if(servo_no==1)
      servo1.write(servoderece);
    else if(servo_no==2)
      servo2.write(servoderece);
    else if(servo_no==3)
      servo3.write(servoderece);
    else if(servo_no==4)
      servo4.write(servoderece);
    else if(servo_no==5)
      servo5.write(servoderece);

    delay(30);
  }

  return(servoderece);
}
```