

T.C.

NİĞDE ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

MEKATRONİK MÜHENDİSLİĞİ

2015-2016 EÖY LİSANS BİTİRME TEZİ PROJESİ

HEDEF TESPİTİ ve HAREKET TAKİBİ

Cevahir GÜNDÜZ

120607323

Engin AYDIN

120607338

Hanife DEMİRKAYA

120607302

DANIŞMAN

Yrd. Doç. Dr. İlyas KACAR

NİĞDE, 2016

T.C.
NİĞDE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ
BİTİRME TEZİ KABUL VE ONAY BELGESİ

Bölümümüz 120607323 numaralı öğrencisi Cevahir GÜNDÜZ'ün, 120607338 numaralı öğrencisi Engin AYDIN'ın, 120607302 numaralı öğrencisi Hanife DEMİRKAYA'nın "Hedef Tespiti ve Hareket Takibi" başlıklı Bitirme Tezi çalışması aşağıdaki jüri üyeleri tarafından Mekatronik Mühendisliği Bölümü'nde Bitirme Tezi olarak Oy Birliği/Oy Çokluğu ile kabul edilmiştir.

Danışman: Yrd. Doç. Dr. İlyas KACAR
İmzası

Üye: Unvanı Adı Soyadı
İmzası

Üye: Unvanı Adı Soyadı
İmzası

Tezin Savunulduğu Tarih: / /

Bitirme Tezi dersi kapsamında yapılan bu çalışma, ilgili jüriler tarafından değerlendirme sonucunda Mekatronik Mühendisliği Bölümü'nde Bitirme Tezi çalışması olarak kabul edilmiştir. / /

İmza
Bölüm Başkanı
Yrd. Doç. Dr. Mehmet Kürşat YALÇIN

T.C.
NİĞDE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
DOĞRULUK BEYANI

Bitirme tezi olarak sunduğumuz bu çalışmayı tüm akademik kurallara ve Niğde Üniversitesi Yayın Etiği Komisyonu Yönergesine uygun olarak gerçekleştirdiğimizi ve sunduğumuzu; bu kurallar ve ilkelere aykırı hiç bir yol ve yardıma başvurmaksızın bizzat hazırladığımızı beyan ederiz.

Tezimizle ilgili yaptığımız beyana aykırı bir durum saptanırsa ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımızı bildiririz. / /

İmza
Cevahir Gündüz

İmza
Engin Aydın

İmza
Hanife Demirkaya

TEŞEKKÜR

Öncelikle bu projede bizden hiçbir yardımını esirgemeyen değerli danışman hocamız Yrd. Doç. Dr İlyas KACAR’a, Makine Mühendisliği Bölümü hocalarımızdan Yrd. Doç. Dr Serkan TOROS’a, tüm çalışmalarımızda gerek laboratuvar ortamında sağladıkları imkânlar, gerekse bilgi alma aşamasında yine bize destek olan Niğde Üniversitesi Mekatronik Mühendisliği Bölümü hocalarımıza bütün yardımlarından dolayı teşekkür ederiz.

Sanayi danışmanımız SKY Robot Teknolojileri Sanayi Ticaret Ltd. Şti’ ne proje kapsamında sağladığı imkânlar için teşekkür ederiz.

Eğitim hayatımız boyunca bize desteklerini esirgemeyen bölüm başkanımız Sayın Yrd. Doç. Dr. Mehmet Kürşat YALÇIN’ a desteklerinden dolayı teşekkür ederiz.

Ayrıca projemize “TUBİTAK 2241-A Sanayi Odaklı Lisans Bitirme Tezi Destekleme Programı” kapsamında destek olan Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna teşekkür ederiz.

Özellikle hayatımızın her aşamasında her konuda bizim yanımızda olan ailelerimize sonsuz teşekkürlerimizi sunarız.

Cevahir Gündüz

Engin Aydın

Hanife Demirkaya

ÖZET

Günümüzde savunma sanayisinde gerçekleşecek yenilikler tüm dünyanın ilgi odağı haline gelmiştir. Askeri alanda daha pratik ve gelişmiş bir savunma gerçekleştirmek bütün dünyayı cezbetmekle birlikte en çokta ülkemize fayda sağlayacaktır.

Bu proje ile hedef takip ve güdüm sistemlerinin bütünleştirilmesi vasıtasıyla etkili bir savunma amaçlanmıştır. Hedef takip sistemi üzerindeki kamera ile hedef tespit edilecek ve güdüm sistemi ile de hedefe savunma gerçekleştirilecektir.

Güdüm sistemi iki adet adım motor, hedef takip sistemi ise iki adet servo motor, Pan/tilt hareket eksenleri ve bir adet kameradan oluşmaktadır. Bu sistemlerin kontrolü bilgisayar üzerinden seri haberleşen Arduno Mega 2560 ve hazırladığımız ara yüz programı ile sağlanmıştır. Ara yüz programı Microsoft Visual C# programı üzerinde yazılmış, tüm sürücülerin kontrolü ve kameradan alınan görüntünün işlenmesi sağlanmıştır. Güdüm sisteminin tasarımı Solidworks programı ile ve mekanik analizleri ise Ansys programı ile gerçekleştirilmiştir.

Anahtar kelimeler: Microsoft Visual C#, Step ve Servo Motor, Arduino Mega 2560, Güdüm Sistemi, Hedef Takip Sistemi

ABSTRACT

Nowadays, innovations in defense industry have been the focus of attention of the whole world. Performing more practical and advanced defense in military has attracted the whole world also it will provide the most benefit to our country.

In this project, it is aimed to obtain effective defense by integrating the target tracking system and the guidance system. Targets will be identified by camera on the tracking system and targets will be attacked by the guidance system

Guidance system consists of two step motors, while the tracking system consists of two servo motors, pan / tilt axes and one camera. Control of the whole system is carried out by serial communication with Arduino Mega 2560 card and interface program that we wrote. Interface program is written by using Microsoft Visual C # program. Driven of motors and acquiring photo frames from camera are obtained by interface program. The design of guidance system is drawn by using SolidWork software and mechanical analysis are carried out by using Ansys software.

Keywords : Microsoft Visual C # , stepper and servo motor , Arduino Mega 2560, Guidance System, Target Tracking System

İÇİNDEKİLER

TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT.....	vi
İÇİNDEKİLER	vii
ŞEKİLLER LİSTESİ	ix
TABLO LİSTESİ.....	x
ÖNSÖZ	xi
BÖLÜM 1 GİRİŞ.....	1
1.1 Giriş	1
1.2 Çalışmanın Amacı	2
1.3 Konunun Önemi	2
1.4 Proje için planlanan iş paketleri ve gerçekleştirme düzeyleri.....	3
BÖLÜM 2 LİTERATÜR ÖZETİ	5
2.1 Literatür Özeti	5
BÖLÜM 3 KONTROL ALGORİTMASI ve SİSTEM TASARIMI	7
3.1 Kontrol Kodları	7
3.1.1 Adım motor kontrolü ve ara yüz tasarımı.....	7
3.1.2 Görüntü alma ve görüntü işleme.....	7
3.1.3 Arayüz programının kullanımı.....	8
3.2 Güdüm Sistemi Tasarımı.....	8
3.3 Analiz Sonuçları	10
3.3.1 Statik yapısal analiz	10
3.3.2 Yorulma analizi.....	13
3.3.3 Rijit dinamik analiz.....	16
3.3.4 Şekil optimizasyonu.....	21

3.3.5 Harmonik analiz.....	22
BÖLÜM 4 BÜTÇE PLANI	24
4.1 Bütçe ve Gerekçesi	24
BÖLÜM 5 SONUÇ.....	26
5.1 Sonuç	26
5.5.1 Hedef takip sistemi	26
5.5.2 Güdüm sistemi	27
5.5.3 Kontrol panosu.....	28
5.5.4 İleride yapılması planlanan çalışmalar	29
KAYNAKLAR	30
EKLER.....	33
EK 1: Microsoft Visual C# Ana Program	33
EK 2: Arduino Programı	45
EK 3: Kullanılan Elektronik Sürücü Kartın Blok Şeması	53
EK 4: Microsoft Visual C# Programı Ara Yüzünde Alınan Görüntünün Eksenleri	54
ÖZGEÇMİŞ	55
1	55
2	56
3	57

ŞEKİLLER LİSTESİ

Şekil 1.1. Tasarlanan sistemin şeması.....	1
Şekil 3.1. Microsoft Visual C# ara yüz formu	8
Şekil 3.2. Güdüm sisteminin çeşitli perspektif görünüşleri	9
Şekil 3.3. Güdüm sistemi tasarımının boyutları	9
Şekil 3.4. Kuvvet ve ankastre bağlantı uygulanma yeri.....	11
Şekil 3.5. Mesh	11
Şekil 3.6. Von-Misses gerilme analizi sonucu.....	12
Şekil 3.7. Total deformasyon	12
Şekil 3.8 Soderberg diyagramı.....	13
Şekil 3.9 Uygulanan yüklemenin zamana göre değişimi.....	14
Şekil 3.10. Kuvvet ve ankastre bağlantı uygulanma yerleri	14
Şekil 3.11. Yorulma ömrü.....	15
Şekil 3.12. Soderberg’e göre yorulma emniyet katsayısı	15
Şekil 3.13. “Fatigue sensitivity” diyagramı	16
Şekil 3.14. Analizde kullanılan bağlantı noktaları.....	17
Şekil 3.15. Uygulanan kuvvet ve moment değerleri.....	18
Şekil 3.16. Güdüm sisteminde 4 numaralı bağlantıya gelen toplam kuvvetler	19
Şekil 3.17. Güdüm sisteminde 2 numaralı bağlantıya gelen toplam kuvvetler	20
Şekil 3.18. Güdüm sisteminde 4 numaralı bağlantıya gelen toplam momentler	20
Şekil 3.19. Güdüm sisteminde 2 numaralı bağlantıya gelen toplam momentler	21
Şekil 3.20. Kuvvet ve ankastre bağlantı uygulanma yerleri	21
Şekil 3.21. Modelde gerilmeye maruz kalmayan kısım.....	22
Şekil 3.22 Frekans Genlik Grafiği	23
Şekil 5.1. Hedef takip sistemi önden ve arkadan görünüşü	26
Şekil 5.2 Güdüm Sisteminin Görünüşü	27
Şekil 5.3. Kontrol panosu ve devre elemanlarının yerleşimi	28
Şekil 5.4 Sistemin genel görünümü	28

TABLO LİSTESİ

Tablo 1.1 İş- zaman çizelgesi (Gant Chart).....	4
Tablo 3.1 Yapısal çelik malzeme özellikleri	10
Tablo 3.2 Sistemde analizin uygulandığı parçalar	17
Tablo 3.3 Model analizin mod değerleri	22
Tablo 4.1 Bütçe Planı	24

ÖNSÖZ

Hazırlanılan bu proje genel olarak savunma sanayisi alanını kapsayan bir projedir. İlk olarak ilgili alan ve çalışmanın amacından bahsedilmiş olup konunun önemi özetlenmiştir.

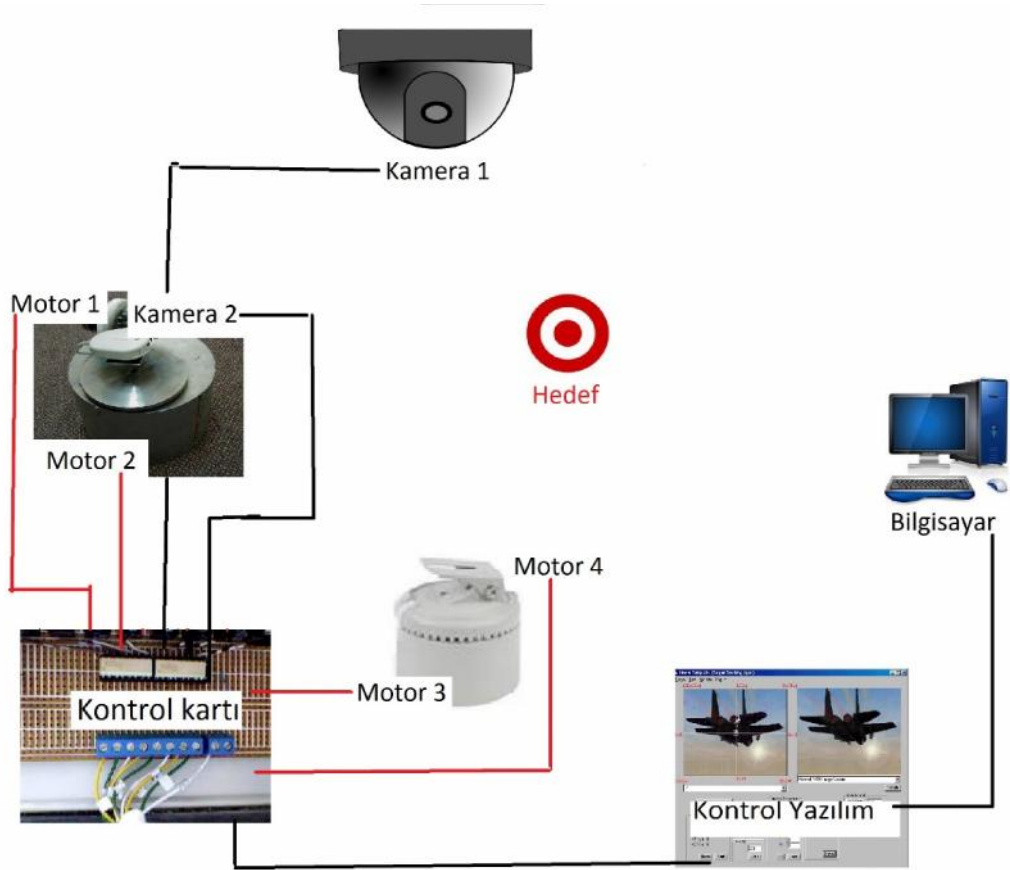
Daha sonra, Niğde Üniversitesi tez yazım kuralları dikkate alınarak, giriş, literatür taraması, bütçe hesabı, tasarım ve analizlerinin yapılması gibi kısımlar detaylıca açıklanmıştır. Projede kullanılan tüm yazılım kodları Ek halinde bu belgenin sonuna eklenmiştir.

Daha sonra yapılacak geliştirme çalışmaları için detaylı bir açıklayıcı belge olduğu kanaatindeyiz.

BÖLÜM 1

1.1 Giriş

Günümüz teknolojisinde gün geçmeden birçok yenilikler gerçekleşmekte her yeni üretilen ürünler kendinden öncekini gölgede bırakarak yoluna devam etmektedir. Bu gelişen teknolojinin bu projede bizi ilgilendiren kısmı, savunma sanayisindeki yeniliklerdir. Savunma alanında, diğer alanlarda olduğu gibi insan gücüne olan ihtiyacın azaltılması önemlidir. Böylece hem daha atik ve pratik bir mekanizma geliştirilmiş olacak hem de insanlar savunma alanında daha az zarar görecektir.



Şekil 1.1. Tasarlanan sistemin şeması

Bu proje ile hem hedef tespiti hem de güdüm sistemi ile de tespit edilmiş olan hareketli nesneye karşı savunma gerçekleştirilmiştir. Sistem içerisinde;

- küresel koordinatlarda hareket eden, tahriki adım motorların gerçekleştirdiği hazır olarak bulunan güdüm sistemi,
- iki adet servo motor ve bunların Pan/tilt hareket eksenlerinden oluşan hedef takip sistemi bulunmaktadır.

Adım ve servo motorların kontrolü Arduino Mega 2560 üzerinden oluşturulan kodlarla gerçekleştirilmiştir. Hedef takip sistemi, üzerinde bulundurduğu kamera vasıtasıyla hareketli nesneyi algılayarak adım ve servo motorların dönüşünü hareketli nesneye yönlendirmektedir. Bahsedilen sistemin tasarlanan hali Şekil 1.1 de verilmiştir.

1.2 Çalışmanın Amacı

Yapılan bu çalışmada tezin başlığı, Hedef Tespiti ve Hareket Takibi olarak belirlenmiştir. Projenin amacı, şekilden ve mesafeden bağımsız olarak hedefin hareket yörüngesinin görüntü işleme teknikleri vasıtasıyla belirlenmesi, bu yörüngelerde hareket edecek bir mekanik sistem tasarlamak ve oluşturulan prototip ile hareketi gerçekleştirmektir. Bu projede daha önce yapılan hedef takip sistemi ve benzeri projelere ek olarak güdüm sistemi yer almaktadır. Böylece hem hedef tespit edilecek hem de tespit edilen hedefe karşı savunma gerçekleştirilecektir.

1.3 Konunun Önemi

Son zamanlarda dünya üzerinde birçok ülke arasında iç çatışmalar olmakta ve gelişmiş ülkeler olası savaş ve saldırılara karşı yeni savunma stratejileri geliştirmektedir. Bu proje ile savunma sanayisinde insan gücünün en aza indirilerek daha dayanıklı bir sistem oluşturmak amaçlanmıştır ve projenin gerçekleşmesi ile en başta ülkemiz adına büyük yenilik sağlanmış ve büyük bir başarı elde edilmiştir.

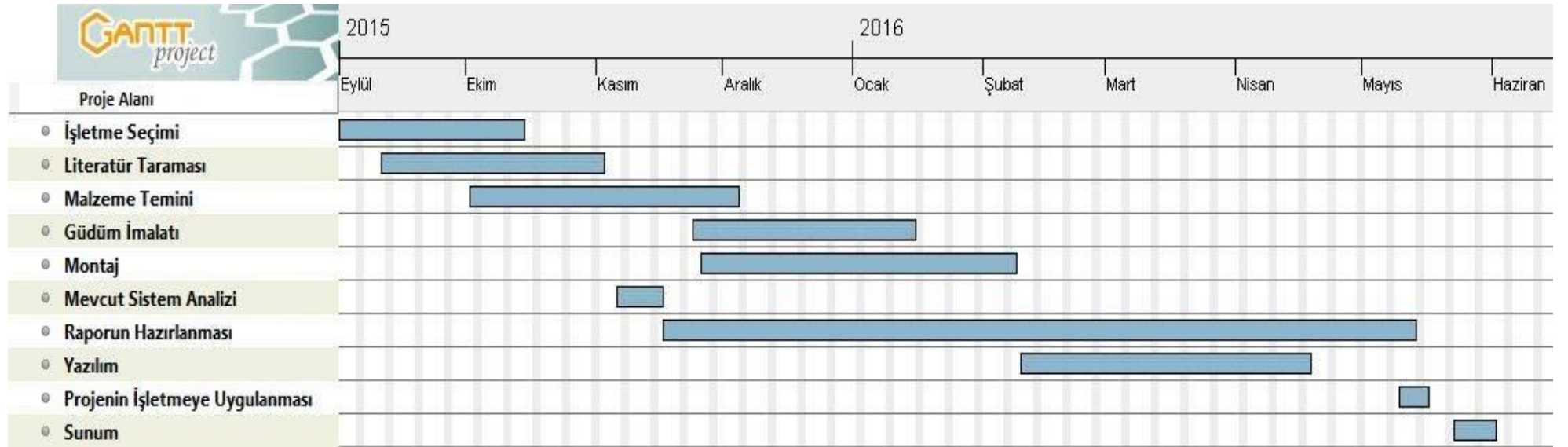
1.4 Proje için planlanan iş paketleri ve gerçekleşme düzeyleri

Proje için planlanan iş paketleri ve iş-zaman çizelgesi Tablo 1.1 de verilmiştir. Planlanan iş paketleri herhangi bir aksaklık olmaksızın başarı ile gerçekleştirilmiştir.

Tablo 1.1 de verilen iş-zaman çizelgesinde de görüldüğü üzere projenin hayata geçmesi için ilk adımlara Eylül-2015 itibariyle başlanmıştır.

Malzeme temini literatür taraması başladıktan bir ay sonra başlamış ve aralık ayında son bulması planlanmıştır ancak mekanik sistemdeki bazı problemler (motorların arızalanması, talep edilen firmanın stoklarında bulunmaması vs.) nedeniyle malzeme alımı Mayıs-2016 tarihine kadar devam etmiştir. Malzeme teminine başlanmasıyla birlikte mevcut durum analizi de yapılmıştır. Aynı zamanda rapor hazırlıklarına da başlanmış ve proje tam anlamıyla faaliyete girene kadar yani mayıs ayının ortalarına kadar rapor hazırlıklarına devam edilmiştir. Gerekli malzemeler temin edildikten sonra hedef takip sisteminin imalat ve montajına başlanmış ve 3 aylık bir süre sonunda hedef takip sistemi, teoriden kurtulup faaliyete geçirilmiştir. Hedef takip sisteminin imalat ve montajı bittikten sonra yazılım aşamasına geçilmiş ve bu aşamaya da 3 aylık bir zaman ayrılmış, yazılım hazırlama işlemi nisan ayında son bulmuştur. İmalat, montaj ve yazılımların bitip projenin hazır olmasıyla mayıs ayında işletmede uygulanması aşamasına geçilmiş ve mayıs ayında tamamlanmış olan proje süreci, haziran ayında yapılan sunum ile son bulmuştur.

Tablo 1.1 İş- zaman çizelgesi (Gant Chart)



BÖLÜM 2

2.1 Literatür Özeti

Yapılan araştırmalarda elde edilen bilgilere göre, hedef takip sistemleri çeşitli algoritmalara göre kullanılmaya devam etmektedir. Güdüm sistemleri de benzer şekilde çeşitli tiptedirler.

Basit ve arkadaşları (2015) nesnenin hareket yönüne göre kamera hareketini sağlayan bir çalışma yapmıştır [1]. Zhang ve arkadaşları (2015) kamera ile üst üste binen insan yüzlerinden kimlik tespitini yapan bir çalışma yapmıştır [2]. Chen ve arkadaşları (2010) birden fazla kamera ile hedefin tespitine yönelik çalışma yapmıştır [3]. Doyle ve arkadaşları (2014) Pan/tilt hareketi yapan kamera ile hedef takip ve tespiti çalışması yapmıştır [4]. Elbouz ve arkadaşları (2015) veri tabanında kayıtlı olan nesne ile kameranın algıladığı nesne arasında kimlik tespitine yönelik çalışma yapmıştır [5]. Shtarkalev ve arkadaşları (2015) doppler radarlardan gelen ancak üst üste binmiş birden fazla hareketli görüntünün tek tek ayırt edilmesini sağlamıştır [6]. Figueroa ve arkadaşları (2013) kamera ile robotik kolun eklem yerlerinin ve açılarının tespitine yönelik çalışma yapmıştır [7]. Crétual Armel and Chaumette (2001) Pan/tilt hareketi yapan kamera ile hedef takip ve tespiti çalışması yapmıştır [8]. Yao ve arkadaşları (2010) hedefin takibinde mobil olan kameranın nereye konumlanacağını belirlemiştir [9]. Kerr ve arkadaşları (1999) ölçüm sistemlerinde, hedef belirlenmesinde ve navigasyon sistemlerinde kullanılmak üzere geliştirdikleri küresel koordinatlı, çift motorlu mekanik ve kameralı sistemleri sayesinde, 1-20 m arasında 0,1 mm hassasiyette; 10 m'den daha uzak mesafelerde çok daha hassas ölçümler yapabilmişlerdir [10]. Surucu ve arkadaşları (2015) İnsansız Hava Araçlarında(İHA) çeşitli nesnelerin takibi ile görsel takip ve denetim sağlamak amacıyla çalışma yapmıştır [11]. Cai ve arkadaşları (2015) tek darbeli radar sistemi ile hedef tespit çalışması yapmıştır [12]. Ahn ve arkadaşları (2009) uçuş plânında sanal bir yapı oluşturmak için nokta hedeflenmesine yönelik çalışma yapmıştır [13]. Gordon ve arkadaşları (1999) Bayes desen eşleştirme tekniği ile hedef tespitine yönelik çalışma

yapmıştır [14]. Morril ve arkadaşları (1961) kontrol sistemlerinin, hedef tespiti ve üç boyutlu hedef izleme üzerine etkisini incelemiştir [15]. Lee ve arkadaşları (2015) sonlu darbe cevabı ile çoklu hedef izleme algoritması oluşturmaya çalışmıştır [16]. Gaxiola ve arkadaşları (2016) hedef takibi için uyarlanabilir dinamik korelasyona dayalı güvenilir bir algoritma çalışması yapmıştır [17]. Wu ve arkadaşları (2016) doppler radar ölçümlerinden yararlanarak çoklu hedef takibi için çalışma yapmıştır [18]. Liu ve arkadaşları (2016) bu çalışmasında birden fazla kamera ile grup halindeki(kuş sürüsü vb.) nesnelerin üç boyutlu izlenmesine yönelik çalışma yapmıştır [19]. Shao ve arkadaşları (2015) küresel hedefe ulaşmayı sağlayan koordinasyon şeması ile hedef takibine yönelik çalışma yapmıştır [20]. He ve arkadaşları (2015) çalışmasında ağırlıklı korelasyon filtresine dayalı kızılötesi hedef takibi sağlamıştır [21]. Zhang ve arkadaşları (2015) bu çalışmasında birden fazla kamera ağı ile geniş alanda hedef takibi gerçekleştirmiştir [22]. Zawadzki ve arkadaşları (2015) pan/tilt kamera ile hareketli bir cismin gerçek zamanlı görüntü analizine yönelik çalışma yapmıştır [23]. Wang ve arkadaşları (2007) karmaşık çözünürlüklerde kablosuz ağlarla görsel hedef takibine yönelik çalışma yapmıştır [24].

BÖLÜM 3

3.1 Kontrol Kodları

Sistemin tüm kodları Arduino programı ve Microsoft Visual C# programında yazılmıştır. Microsoft Visual C# da yazılan kodlar EK 1 de, Arduino programında yazılan kodlar ise EK 2 de verilmiştir.

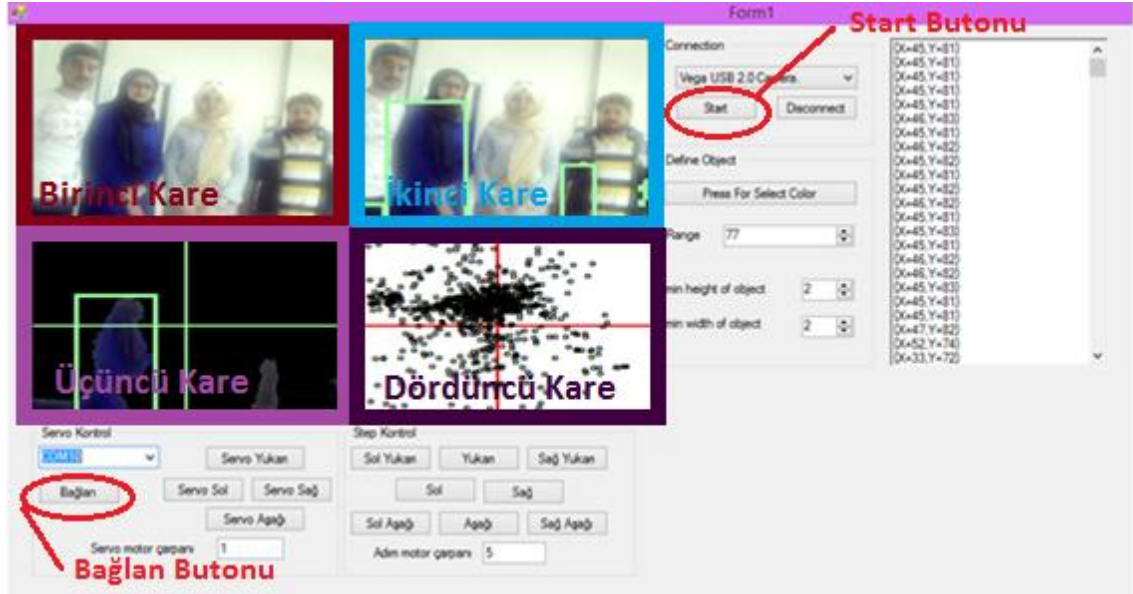
3.1.1 Adım motor kontrolü ve ara yüz tasarımı

Çalışmaya ilk olarak adım motorlarının kontrolü ile başlanmıştır. Arduino programı üzerinde adım motor kodu yazılıp çalıştırılmıştır. Ancak proje iki sistemden oluşacağından çoklu kontrol sağlamak için yalnızca Arduino programı yeterli olmayacaktır. Bu gereksinimden dolayı Microsoft Visual C# programında ara yüz oluşturmak ihtiyacı oluşmuştur. Microsoft Visual C# üzerinde oluşturulan ara yüz sayesinde sistemin kontrolü tek bir program üzerinde daha kolay ve pratik olarak gerçekleştirilmiştir.

3.1.2 Görüntü alma ve görüntü işleme

Projenin en önemli kısımlarından bir diğeri de; kameradan görüntü almak ve bu görüntü üzerinde işlem yapmaktır. Kameradan görüntü alınması ve alınan görüntü üzerinde işlem yapılması, Microsoft Visual C# programında yazılarak oluşturulmuştur. Oluşturulan ara yüz ekran görüntüsü 3.1 de verilmiştir. 3.1 de;

- birinci karede kameradan alınan görüntü olduğu gibi aktarılmakta,
- ikinci karede bu görüntü üzerindeki nesneler tanınarak nesnelerin dış kısımları kutucuklara alınmakta,
- üçüncü karede sadece kutucuklardan en büyüğü hedef olarak seçilerek kutunun ortasının koordinat değerleri elde edilmekte ve
- dördüncü karede de, bu koordinat değerleri x-y koordinat sistemi üzerinde noktalar halinde işaretlenmektedir.



Şekil 3.1. Microsoft Visual C# ara yüz formu

Ara yüz formundaki 1, 2 ve 3 olmak üzere ilk üç kare “videosourceplayer” ve dördüncü kare “picturebox”tır.

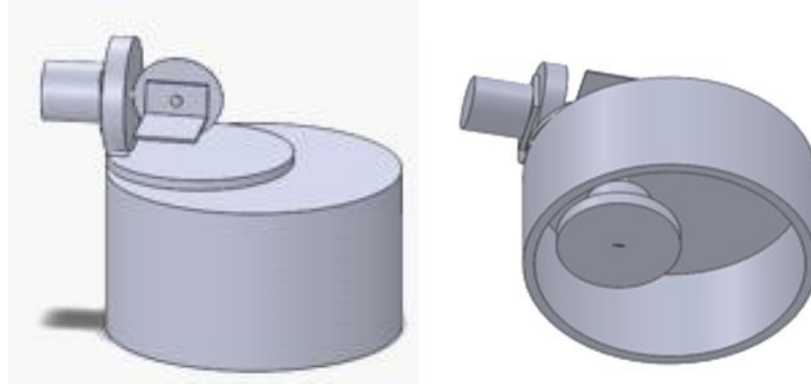
3.1.3 Arayüz programının kullanımı

Microsoft Visual C# programı ile oluşturulan ara yüz formunda ilk olarak “Bağlan” butonu ile serial porta bağlantı sağlanmıştır.

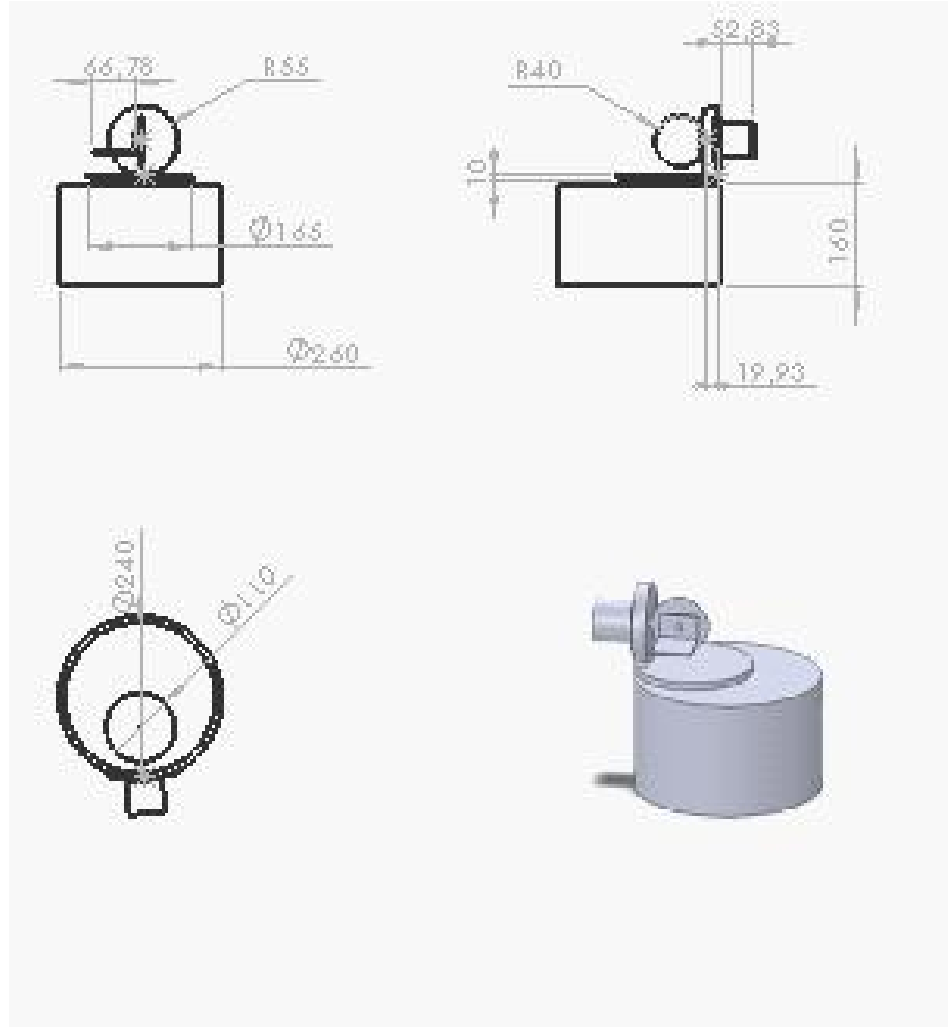
Connection kısmından kamera seçimi yapıp “Start(başlat)” butonuna basılarak kamera aktif hale getirilmiştir.

3.2 Güdüm Sistemi Tasarımı

Güdüm sistemi tasarımı, Solidworks programında üç boyutlu ve perspektif görünüş olarak çizilmiştir. Sistemin perspektif görünüşleri Şekil 3.2 de ve boyutları Şekil 3.3 de verilmiştir.



Şekil 3.2. Güdüm sisteminin çeşitli perspektif görüşleri



Şekil 3.3. Güdüm sistemi tasarımının boyutları

3.3 Analiz Sonuçları

Bu kısımda, tasarlamış olduğumuz sistemin mekanik analizleri yapılmıştır.

3.3.1 Statik yapısal analiz

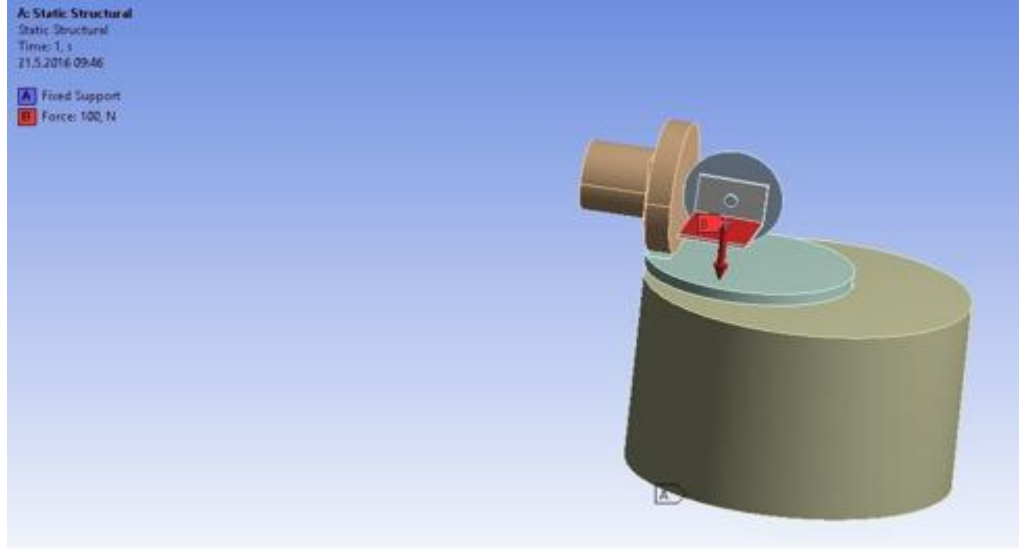
Statik analiz: Yüklerin sabit olduğunu veya tam değerlerine ulaşınca kadar çok yavaş bir şekilde uygulandığını ve daha sonra zamana göre sabit kaldığını varsayar. Bu varsayım nedeniyle, uyarılan sistemin hızı ve ivmesi dâhil olmak üzere tüm eylemsizlik etkilerinin ihmal edilebilir olduğu kabul edilmelidir. Bu nedenle statik analizler sabit stresler ve yer değiştirmeler üretir.

Kullanılan malzemenin özellikleri: Güdüm Sistemi tamamen çelik malzemeden imal edilmiş olup özellikleri aşağıdaki tabloda görülmektedir.

Tablo 3.1 Yapısal çelik malzeme özellikleri

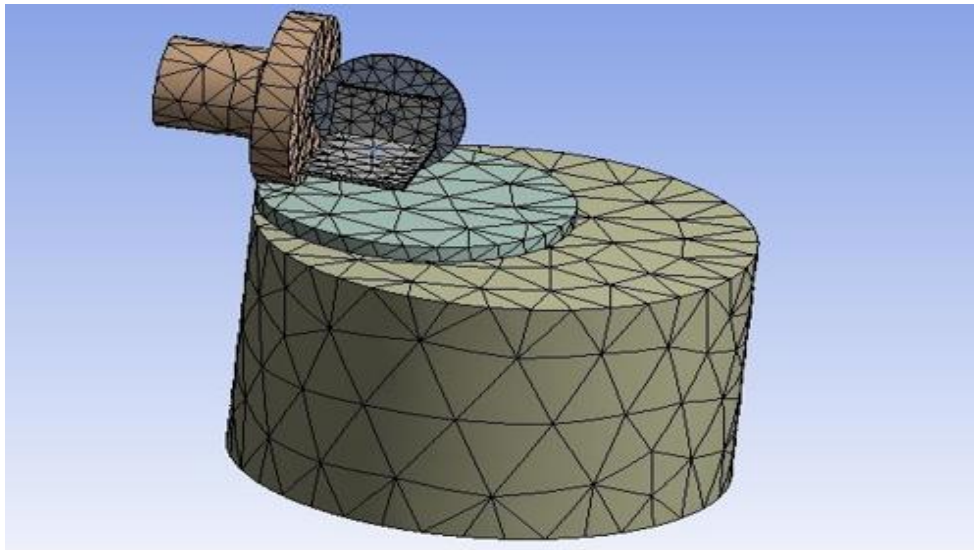
Malzeme özellikleri	Çelik
Elastiklik modülü	$2E+11 \text{ Pa}$
Poisson oranı	0,3
Akma mukavemeti	$2,5E+0,8$
Yoğunluk	7850 kgm^{-3}
Kopma mukavemeti	$4,6E+0,8$

Şekil 3.4 de güdüm sistemine sınır şartı olarak, yere bağlandığı noktaya ankastre bağlantı ve kameralık kısmına -y yönünde 100 N kuvvet uygulanmıştır.



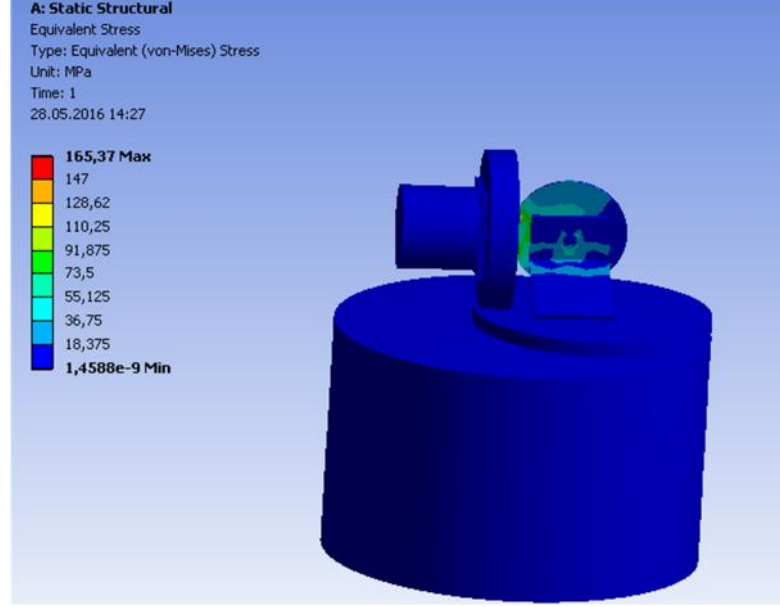
Şekil 3.4. Kuvvet ve ankastre bağlantı uygulanma yeri

Sistem; Şekil 3.5 de görüleceği gibi meshleme işlemi ile küçük parçalardan oluşan temel elemanlara ayrılmıştır.



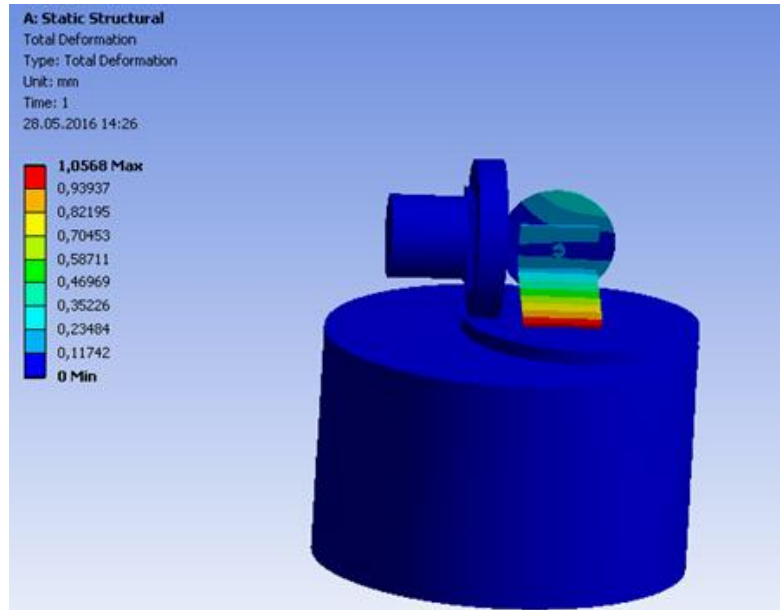
Şekil 3.5. Mesh

Şekil 3.6 da analizin sonuçları görülmekte olup, en büyük gerilme $165,37 \text{ MPa}$ değerindedir. Bu sonuç; çeliğin akma gerilmesinden küçük olduğu için güdüm sisteminde herhangi bir hasar olmayacaktır.



Şekil 3.6. Von-Misses gerilme analizi sonucu

Şekil 3.7 de maksimum deformasyon değeri $1,0568 \text{ mm}$ olarak görülmüştür.



Şekil 3.7. Total deformasyon

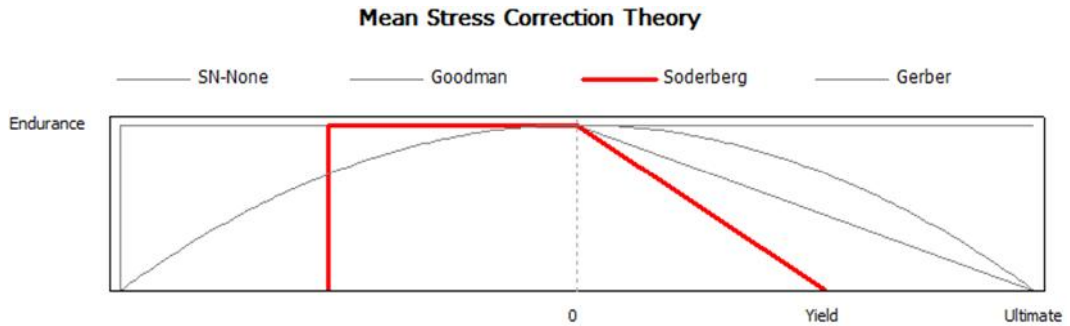
3.3.2 Yorulma analizi

Parçanın; uygulanmış olan yükleme etkisi altındaki çalışma ömrü (ne kadar süre sonra kırılacağı) veya sonsuz ömürde olması için en fazla ne kadar yükleme yapılmış olması gerektiği, kırılmanın nerede olacağı, yorulma sınırının ne olduğu, güvenlik katsayısının ne olduğu sorularının cevapları, yorulma analizi ile elde edilebilmektedir.

Bir malzemenin değişken dinamik yükler altında (akma gerilmesine ulaşmayan bir gerilme durumunda bile) belirli bir süre sonunda kırılmasına yorulma hasarı denir. Özel durumlar dışında çoğunlukla yorulmaya yol açan gerilme seviyesi malzemenin akma dayanımından düşüktür. Soderberg eğrisi düşük sünekliğe sahip malzemeler için kullanışlıdır. Güdüm sistemi üzerinde yapılan yorulma analizinde Soderberg eğrisi esas alınmıştır. Formülü ise Denklem3.1 ve Şekil 3.8 de gösterilmiştir.

$$\frac{\sigma_A}{S_E} + \frac{\sigma_M}{S_Y} = 1$$

Denklem3.1

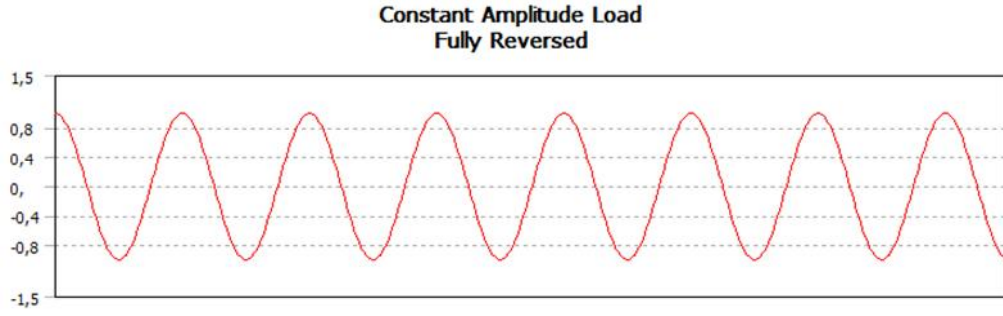


Şekil 3.8 Soderberg diyagramı

Yapıya 100 N'luk periyodik bir kuvvet uygulanmıştır ($F=100*\sin(t)$ N). Analizde aşağıdaki yükleme parametreleri kullanılmıştır:

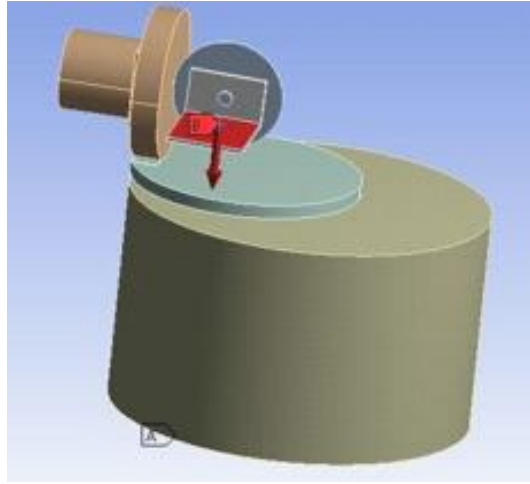
- Ortalama yük : $F_m = \frac{F_{\max} + F_{\min}}{2} = 0 \text{ N}$
- Yük aralığı: $\Delta F_r = F_{\max} - F_{\min} = 200 \text{ N}$
- Yük genliği : $F_a = \frac{F_r}{2} = \frac{F_{\max} - F_{\min}}{2} = 100 \text{ N}$
- Yük oranı (Yükleme oranı) : $R = \frac{F_{\min}}{F_{\max}} = 1$

Uygulanan yüklemenin zamana göre değişimi Şekil 3.9 da verilmiştir.



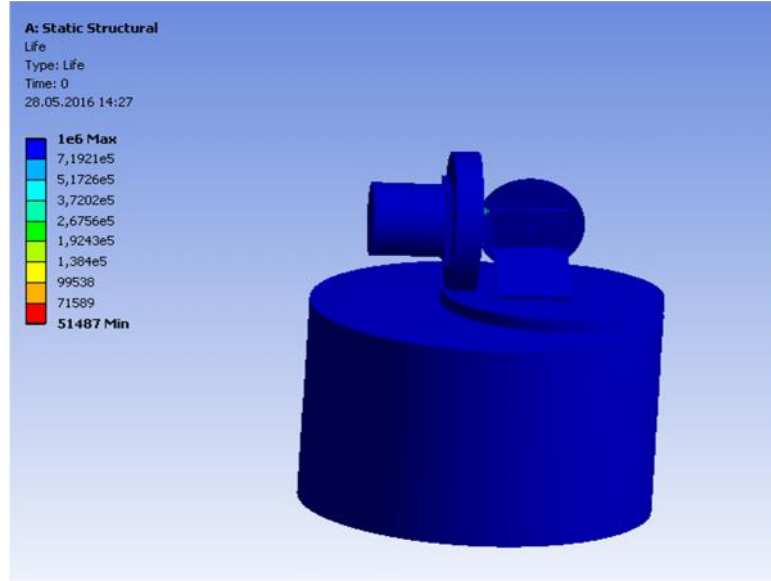
Şekil 3.9 Uygulanan yüklemenin zamana göre değişimi

Analiz sonucunda Şekil 3.10 da görüleceği üzere, ömür değerinin sonsuz ömür değeri olduğu anlaşılmaktadır. Ansys programında $1e6$ çevrim değeri sonsuz ömür anlamına gelmektedir. Şekil 3.10 da; modele sınır şartı olarak, yere bağlandığı noktaya ankastre bağlantı ve kameralık kısmına y yönünde 100 N kuvvet uygulanmıştır.



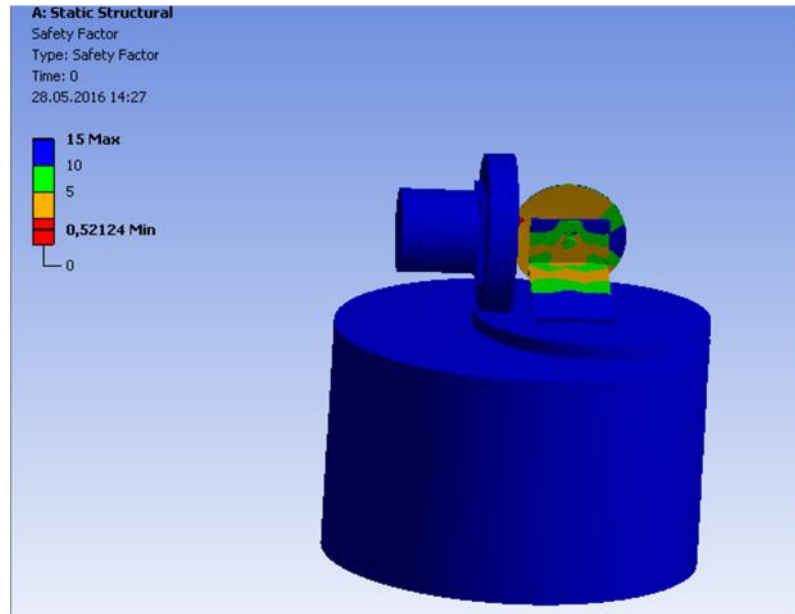
Şekil 3.10. Kuvvet ve ankastre bağlantı uygulanma yerleri

Şekil 3.11 de verilen analiz sonuçlarına göre minimum ömrünün 51487 çevrim olduğu görülmüştür. Buna göre bu ömürden sonra yorulma hasarı başlayacaktır.



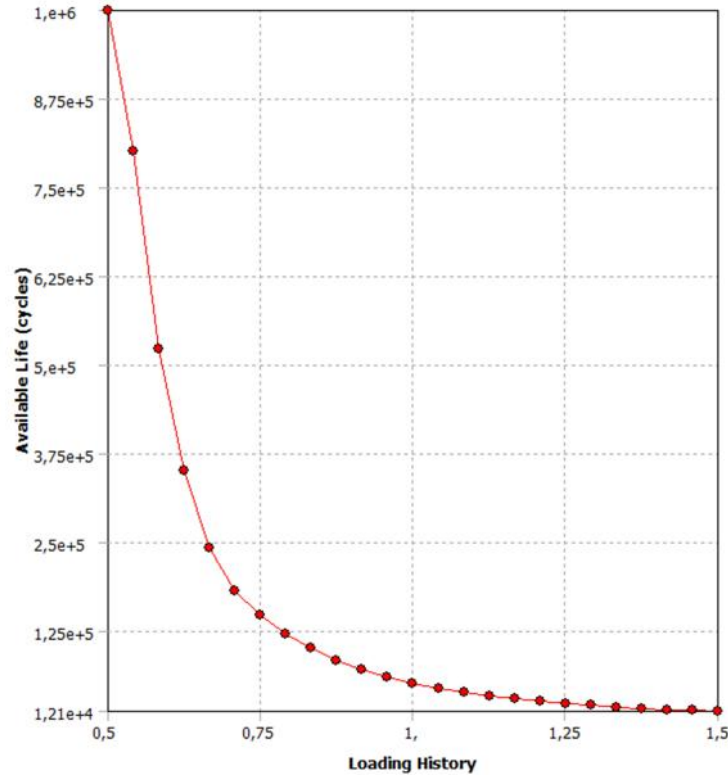
Şekil 3.11. Yorulma ömrü

Şekil 3.12 de, modelin mevcut gerilme halinin Soderberg'e göre kıyaslanarak bulunan yorulma emniyet katsayısının 0,52124 olduğu görülmüştür. Emniyet katsayısında 1 den küçük değerler, parçanın tasarım ömrüne ulaşmadan yorulma hasarına uğrayacağını göstermektedir. Bu değere göre malzemede yüklemekten dolayı 51487 çevrimden sonra yorulma hasarı gözlenecektir.



Şekil 3.12. Soderberg'e göre yorulma emniyet katsayısı

Fatigue Sensitivity diyagramı: Bu diyagram; yükleme değerinin değişmesi halinde yorulma ömrünün ne olacağını gösterir. Şekil 3.13 de görüldüğü gibi yük arttıkça yapının ömrü azalmaktadır. Uygulanan yük 1,5 kat arttırılınca ömür değeri $1,21 \cdot 10^4$ çevrime düşmektedir.

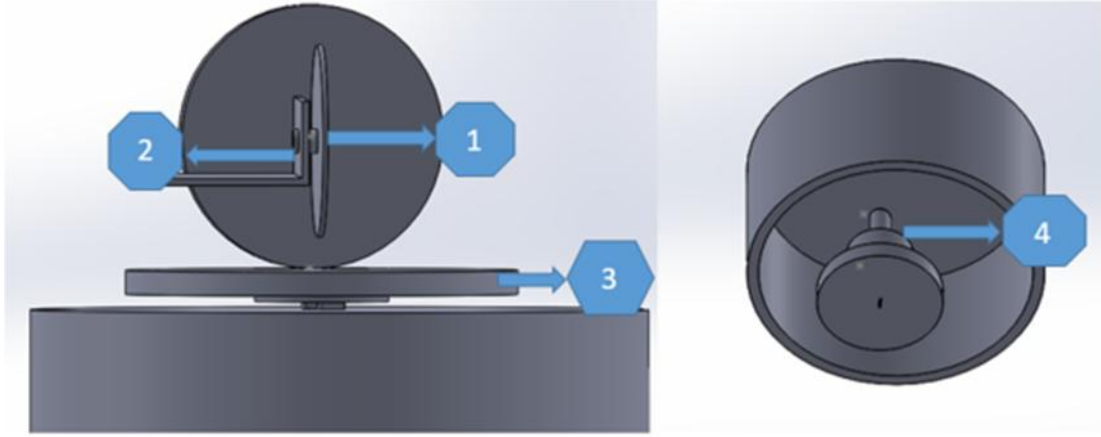


Şekil 3.13. “Fatigue sensitivity” diyagramı

3.3.3 Rijit dinamik analiz

Rijit dinamik analizi; yapının bütün elemanlarının rijit olduğu kabul edilerek yapılan bir analiz türüdür. Bu nedenle yapıda gerilme ve şekil değiştirme değerleri okunamaz. Bu analiz, yapıya uygulanan öteleme-dönme, hız-ivme, yerçekimi gibi etkiler neticesinde; yapının bağlantı noktalarına gelecek olan sadece kuvvet ve moment değerlerinin tespiti amacıyla yapılan bir analiz türüdür.

Şekil 3.14 de; sistemin analizinin yapıldığı bağlantı noktaları görülmektedir.



Şekil 3.14. Analizde kullanılan bağlantı noktaları

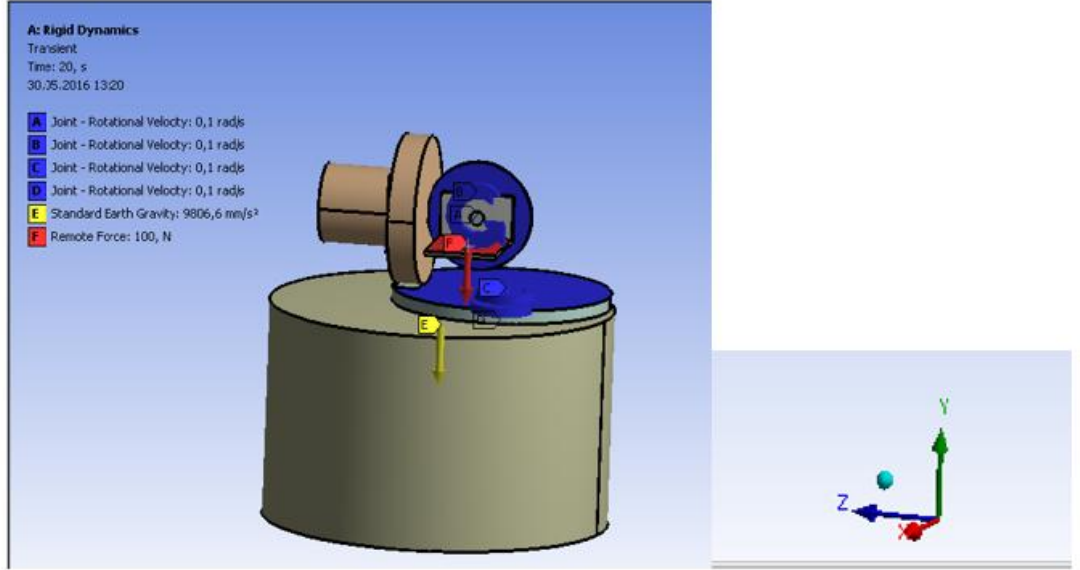
Bağlantı noktalarında, hangi parçaların birbirine bağlı oldukları Tablo 3.2 de açıklanmıştır.

Tablo 3.2 Sistemde analizin uygulandığı parçalar

Bağlantı noktaları	Bağlanan elemanlar
1	Üst adım motor
2	Kameralık ve üst adım motor arası
3	Döner (üst) tabla ve alt motor
4	Alt adım motor ve döner tabla

Sisteme; 0,1 Nmm moment, - y yönünde 100N kuvvet ve saatin tersi yönünde $\omega=0,1$ rad/s açısal hız uygulanıp, dönüş eksenleri z eksenine hizalayıp 60 sn süre aralığında analiz edildikten sonra, sistemin yukarıdaki bağlantı tablosunda gösterilen bağlantılarında oluşan moment ve kuvvet değerleri elde edilmiştir.

Şekil 3.15 de sisteme uygulanan moment, kuvvet, yer çekimi değerleri ve yerleri; ayrıca yapılan analizdeki koordinat sistemi gösterilmiştir.



Şekil 3.15. Uygulanan kuvvet ve moment değerleri

3.3.3.1 Rijit dinamik analizde kuvvet dağılımının yorumlanması

$\omega=0,1 \text{ rad/sn}$ açısal hıza sahip olan yapının bir tam tur ($2\pi \text{ radyan}$) dönebilmesi için 60 sn'lik analiz zamanı gerekmektedir. Bu hesaplamalar aşağıdaki denklemlerde

Denklem 3.2,

Denklem

3.3, Denklem 3.4, Denklem 3.5, Denklem 3.6 da belirtilmiştir.

$$\theta = \omega \cdot t$$

Denklem 3.2

$$2\pi = \omega \cdot t$$

Denklem 3.3

$$\omega = \frac{2\pi}{T}$$

Denklem 3.4

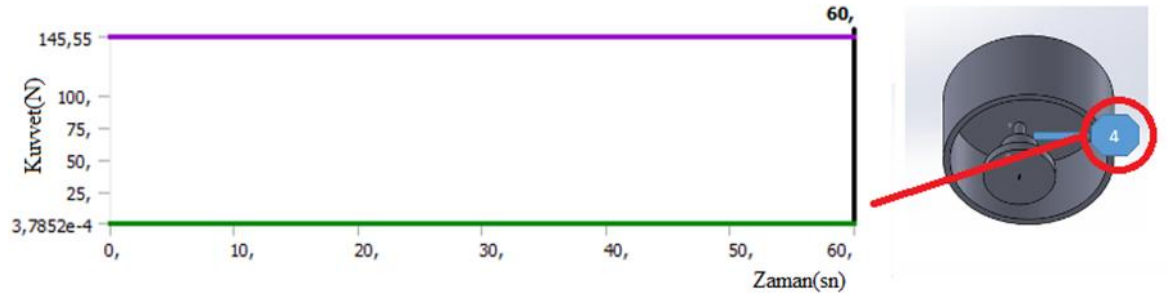
$$\omega = 2\pi$$

Denklem 3.5

$$\theta = 0.1 \times 60 = 6 \text{ rad}$$

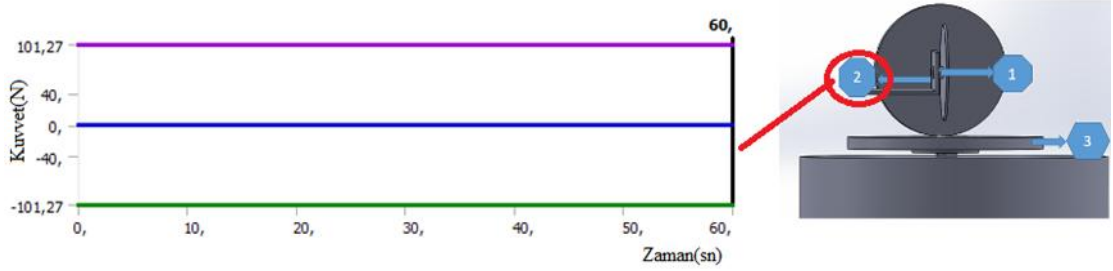
Denklem 3.6

Şekil 3.16 da görüldüğü gibi en yüksek kuvvet değeri Y ekseninde ve 145,55 N görülmüştür.



Şekil 3.16. Güdüm sisteminde 4 numaralı bağlantıya gelen toplam kuvvetler

Şekil 3.17 de görüldüğü gibi en yüksek kuvvet değeri Z ekseninde ve 101,27 N görülmüştür.

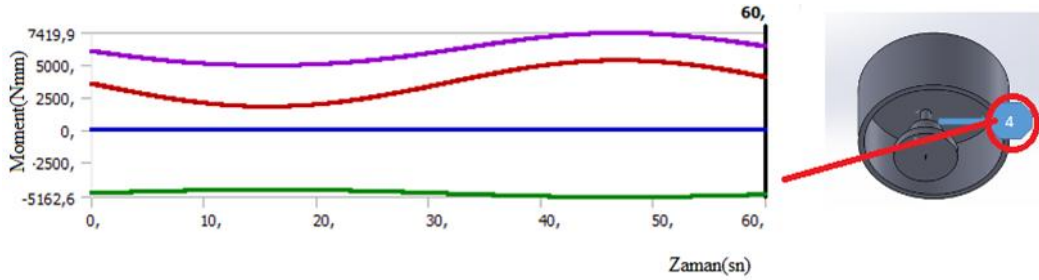


Şekil 3.17. Güzüm sisteminde 2 numaralı bağlantıya gelen toplam kuvvetler

3.3.3.2 Rijit dinamik analizde moment dağılımının yorumlanması

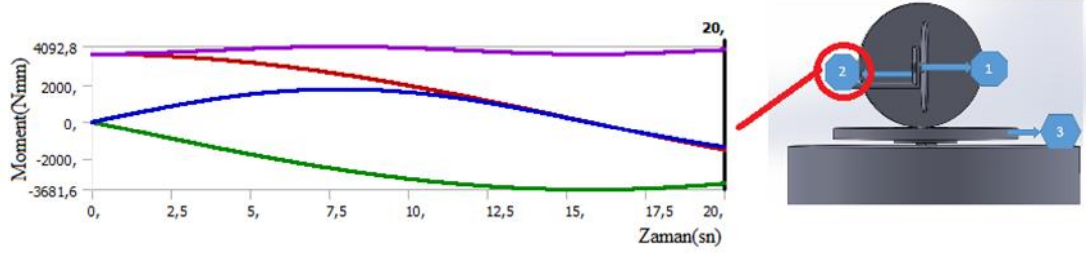
Güzüm sisteminin; bağlantılarının eksenlerinde oluşacak olan momentin zamana göre değişim değerleri analiz edilmiştir.

Şekil 3.18 de görüldüğü gibi X ekseninde moment dağılımı artış göstermiştir ve Z ekseninde moment görülmemiştir. Toplam moment maksimum 7419,9 Nmm'dir.



Şekil 3.18. Güzüm sisteminde 4 numaralı bağlantıya gelen toplam momentler

Şekil 3.19 da görüldüğü gibi X ve Y eksenlerinde moment değeri düşüş göstermiş ve toplam moment değeri $4092,8 \text{ Nmm}$ değerindedir.

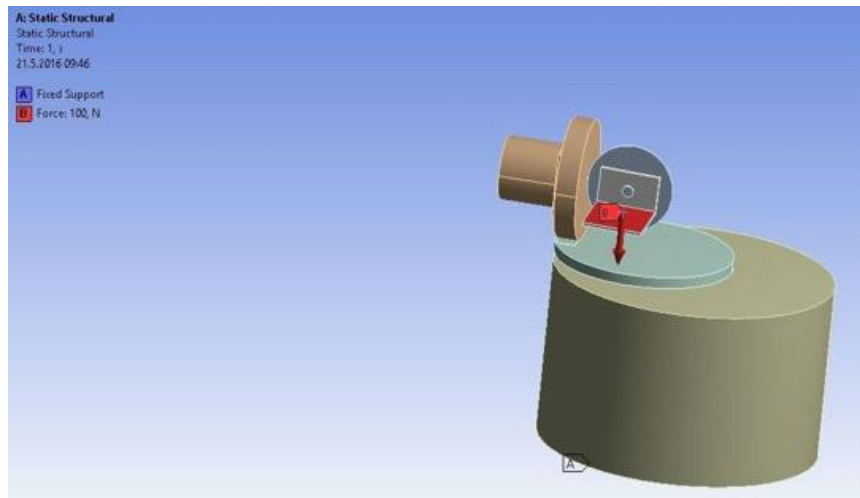


Şekil 3.19. Güdüm sisteminde 2 numaralı bağlantıya gelen toplam momentler

3.3.4 Şekil optimizasyonu

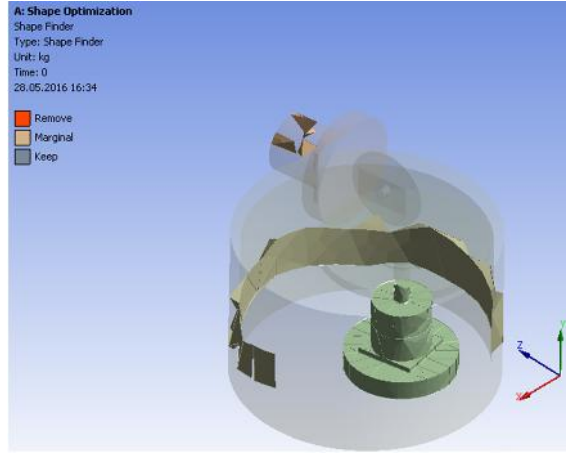
Şekil optimizasyonu; tasarlanan bir geometri üzerine yükleme ve sınır şartları uygulandığında, üzerine gerilme gelmeyen kısımların tespit edilerek tasarımdan çıkartılması işlemidir. Böylece malzemeden tasarruf edilerek verimliliğin artması ve maliyetin düşürülmesi sağlanacaktır.

Şekil 3.20 de görüldüğü gibi sisteme $-y$ yönünde 100 N yükleme ve sınır şartları uygulanmıştır.



Şekil 3.20. Kuvvet ve ankastre bağlantı uygulanma yerleri

Şekil 3.21 de Ansys analiz sonucuna göre sistemde üzerine gerilme gelmeyen kısımlar tespit edilmiştir.



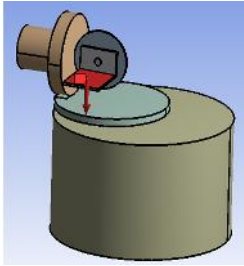
Şekil 3.21. Modelde gerilmeye maruz kalmayan kısım

3.3.5 Harmonik analiz

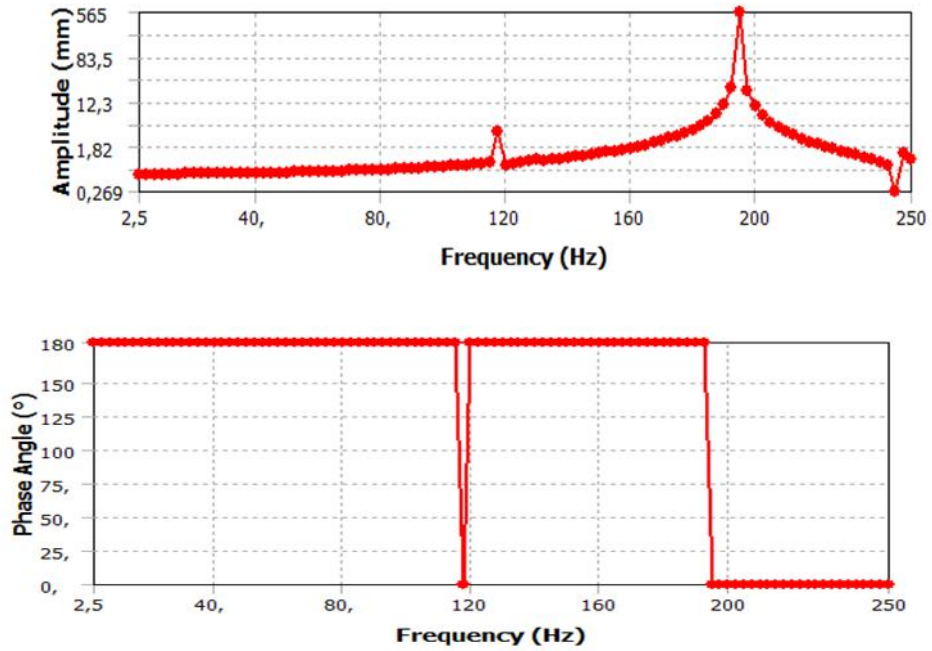
Harmonik analizin amacı; belirli nokta veya noktalardan belirli frekanslarda yapıyı zorlamak ve tüm yapının uygulanan bu zorlamalara karşı genlik cevabını (tepkisini) tespit etmektir.

İlk önce sistemin modal analizi yapılarak doğal frekans değerleri bulunmuştur. Tablo 3.3 de modal analizin 6 mod değeri için doğal frekans değerleri görülmektedir. Daha sonra harmonik analiz yapılarak genlik-frekans ve faz-frekans grafikleri elde edilmiştir.

Tablo 3.3 Model analizin mod değerleri

Mod	Frekans Değerleri	
1	117,45 Hz	
2	131,56 Hz	
3	132, 84 Hz	
4	194,9 Hz	
5	204,05 Hz	
6	246,02 Hz	

Analiz sonucunda Şekil 3.22 de görüldüğü gibi frekans-genlik ve frekans-faz açısı grafikleri elde edilmiştir. Şekildeki grafikte görüldüğü gibi 117,45 Hz frekansında (bu değer sistemin birinci mod frekansıdır) sistemin genliği 4,74 mm değerine, 194,9 Hz frekansında (bu değer sistemin ikinci mod frekansıdır) ise sistemin genliği 565 mm değerine çıkmıştır. Buradan da, yapının bu frekans değerlerinde (117,45 Hz ve 194,9 Hz) çalıştırılmaması gerektiği anlaşılmaktadır.



Şekil 3.22 Frekans Genlik Grafiği

BÖLÜM 4

4.1 Bütçe ve Gerekçesi

Proje kapsamında güdüm sistemi hazır olarak bulunmakta ve projenin diğer mekanik kısımları satın alınmıştır. Satın alınan ürünler ve fiyatları Tablo 4.1 de belirtilmiştir.

Tablo 4.1 Bütçe Planı

	Temin Edilen Ürünler	Ürün Fiyatı (TL)	KDV'li Miktar	Toplam	BÜTÇE TOPLAM
Laboratuvar Sarf Malzemesi Alımları	Havya	14,85	248,96		2124,59
	Havya Standı	7,60			
	Soldex Lehim Teli	47,2			
	3 Adet Tower Pro Servo Motor	43,55			
	3 Adet Servo Motor MG 996(Metal Dişli)	106,5			
	Çelik Cıvata	4,50			
	Çelik Somun	0,60			
	1 Adet Pnömatik Düz Bağlama (10mm)	11,88			
	1 Adet Pnömatik Düz Bağlama (8mm)	11,88			
Bilgisayar Sarf Malzemesi Alımları	Mini Pan Tilt Kiti	22,51	627,51		
	HP 12 A Muadil Toner	283,2			
	4GB Kingstone flashdisk	9,99			
	Harici Hard Disk 1TB	236			
	4 GB DDR3 1333 Mhz Kingston Ram	75,81			
Elektrik-Elektronik Sarf	Arduino Mega 2560	153,99			
	Arduino Motor Sürücü Shield	18,57			

	6 Kanal Usb Servo	66,85	1010,62	
	2Eksenli Joyistick Kart	25,32		
	L293D(Entegre)	5,40		
	1N5819 Diyot	0,13		
	Jumper Kablo(diři-erkek, erkek-erkek)	21,74		
	Muhtelif Elektrik Malzemesi - Güç kaynağı 5 volt gerilim ve 20A akım - Sürücü devreleri - Montaj giderleri	718,62		
Kurtasiye ve Tez	Isısal cilt	112,50	172,50	
	Fotokopi tek yön	60		
Ulaşım	Ulaşım Giderleri Kargo ve Otobüs Ücretleri	65	65	

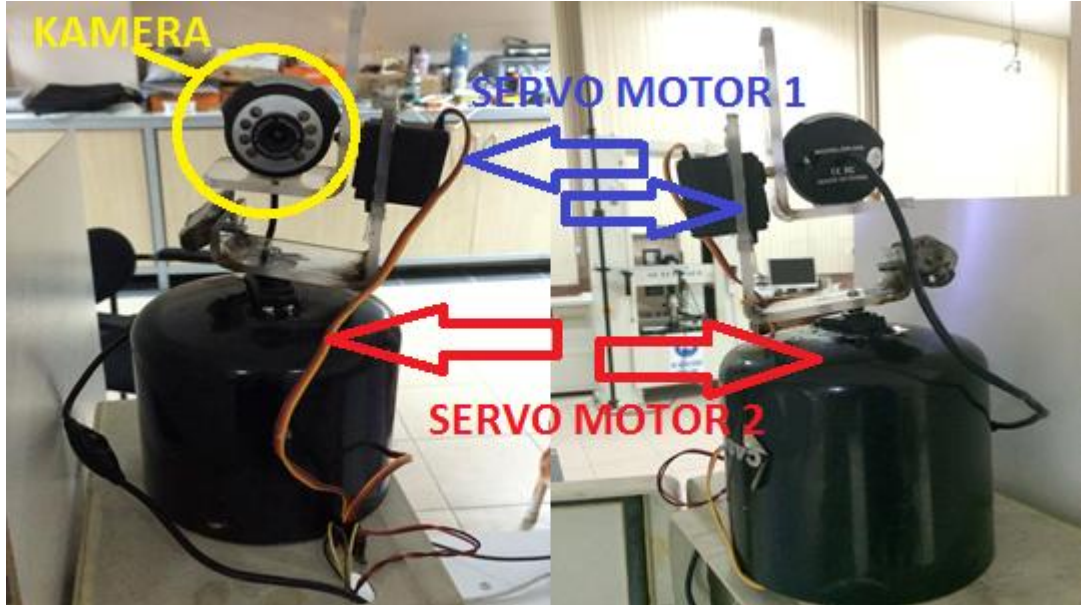
BÖLÜM 5

5.1 Sonuç

Hedef tespiti ve hareket takibi adlı bu projede Microsoft Visual C# ve Arduino programlarında yazılan yazılımlarla kameradan alınan görüntü ile hedef takip sistemi ve güdüm sisteminin aynı yönde hedefe doğru hareket etmeleri sağlanmıştır. Mekanik aksamda güdüm sistemi hazır olarak bulunmakta ve hedef takip sistemi laboratuvar ortamında, servo motor ve 3D printer ile yapılan pan tilt hareket eksen montajının yapılmasıyla tamamlanmıştır.

5.5.1 Hedef takip sistemi

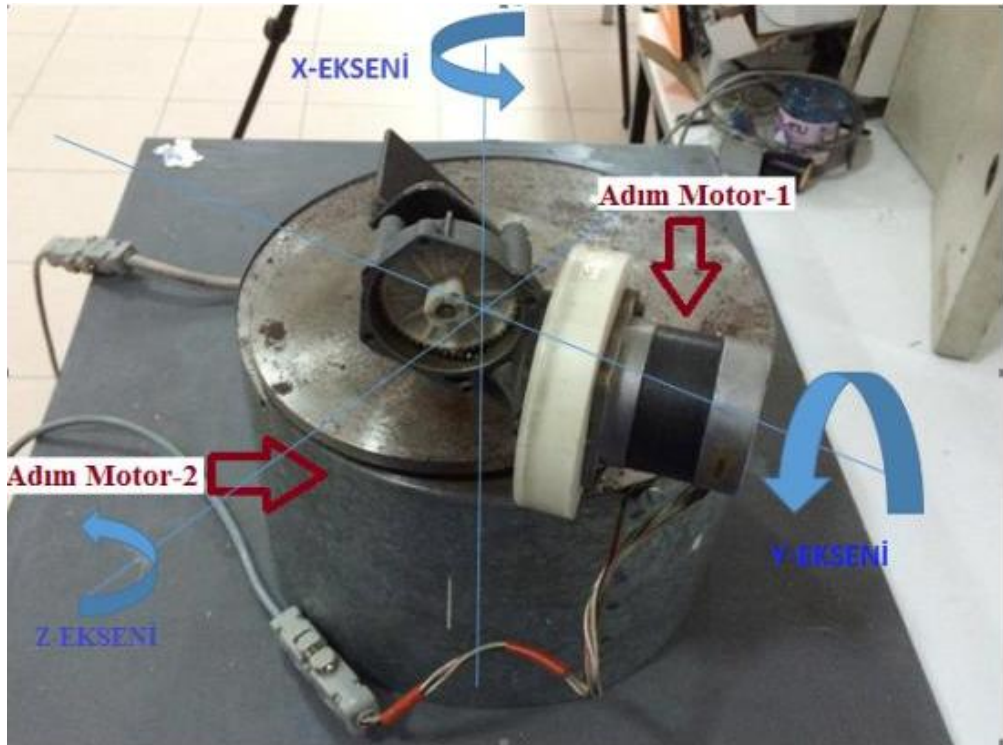
İki adet servo motor, pan/tilt hareket eksen ve kameradan oluşan hedef takip sisteminin önden ve arkadan görünüşleri Şekil 5.1 deki gibidir.



Şekil 5.1. Hedef takip sistemi önden ve arkadan görünüşü

5.5.2 Gdm sistemi

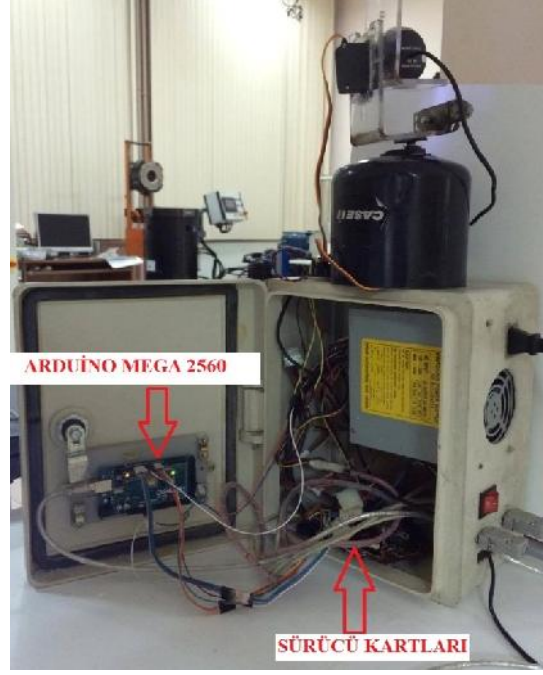
Gdm sistemi **Hata! Bařvuru kaynaęı bulunamadı.** de grldę gibi iki adet adım motordan oluřmakta ve kresel koordinatlarda hareket etmektedir. Birinci adım motor řekilde aıka grlmekte fakat ikinci adım motor sistemin alt kısmında olduęu iin grlmemektedir. Ancak birinci adım motorun dnen tablası net olarak grlmekte ve ikinci adım motoru tařımaktadır.



řekil 5.2 Gdm Sisteminin Grnř

5.5.3 Kontrol panosu

Sistem için Şekil 5.3 de görülen bir kontrol panosu hazırlanarak, elektronik devrelerin bu pano içerisine yerleşimi yapılmıştır. Güç kaynağı 5 V gerilim ve 20 A akım değerine sahiptir. Şekil 5.3 de görüldüğü gibi Arduino Mega 2560 panonun kapağına monte edilmiş olup, sürücü devreler ise panonun içerisinde bulunmaktadır.



Şekil 5.3. Kontrol panosu ve devre elemanlarının yerleşimi

Sistemin tamamı Şekil 5.4 de görüldüğü gibidir.



Şekil 5.4 Sistemin genel görünümü

5.5.4 İleride yapılması planlanan çalışmalar

Bu proje kapsamında hedeflenen iş paketleri başarı ile gerçekleştirilmiştir. Ancak sistemin ticari hale gelebilmesi için daha hızlı ve ergonomik bir yapıya sahip olması gerekmektedir. Bu nedenle bu projenin devamı niteliğindeki ileriki çalışmalarda mekanik sistemin hareket hızının artırılması için daha hızlı kontrol edilebilen sürücü kartları ve daha hızlı motorların kullanılması planlanmaktadır. Ayrıca farklı takip algoritmalarının uygulanarak kıyaslamalarının yapılması, optimizasyonu planlanmaktadır..

KAYNAKLAR

- [1] Basit A, Dailey M. N., Moonrinta J., Laksanacharoen P., “Joint Localization and Target Tracking with a Monocular Camera”, Accepted Manuscript, 2015 Vol: 00, pp.1-23
- [2] Zhang S., Zhu Y., Chowdhury A., “Tracking Multiple Interacting Targets in a Camera Network”, Computer Vision and Image Understanding, 2015 Vol: 134, pp.64-73
- [3] Chen C., Yao Y., Page D., Abidi B., Koschan A., Abidi M., “Camera Handoff and Placement for Automated Tracking Systems with Multiple Omnidirectional Cameras”, Computer Vision and Image Understanding, 2010 Vol: 114, pp.179-197
- [4] Doyle D. D., Jennings A. L., Black J.T., “Optical Flow Background Estimation for Real-Time Pan/Tilt Camera Object Tracking”, Measurement, 2014 Vol: 48, pp.195-207
- [5] Elbouz M., Alfalou A., Brosseau C., Yahia N. B. H., Alam M. S., “Assessing The Performance of a Motion Tracking System Based on Optical Joint Transform Correlation”, Optics Communications, 2015 Vol: 349, pp.65-82
- [6] Shtarkalev B., Mulgrew B., “Multistatic Moving Target Detection in Unknown Coloured Gaussian Interference”, Signal Processing, 2015 Vol: 115, pp.130-143
- [7] Figueroa N. B., Schmidt F., Ali H., Mavridis N., “Joint Origin Identification of Articulated Robots with Marker-Based Multi Camera Optical Tracking Systems”, Robotics and Autonomous Systems, 2013 Vol: 61, pp.580-592
- [8] Créatual A., Chaumette F., “Application of Motion-Based Visual Servoing to Target Tracking”, The International Journal of Robotics Research, 2001 Vol: 11, pp.878-890
- [9] Yao Y., Chen C., Koschan A., Abidi M., “Adaptive Online Camera Coordination for Multi-Camera Multi-Target Surveillance”, Computer Vision and Image Understanding 2010 Vol: 114, pp.463-474

- [10] Kerr D., Marouf B., West T.C., “An Inexpensive Pointing and Tracking System using TV Cameras and an Optical Fibre Beacon”, *Mechatronics*, 1999 Vol: 9, pp.919-927
- [11] Surucu D., Surucu M., Koksall K., Hacıotlu R., “Visual Tracking and Control of Unmanned Aerial Vehicle”, *Signal Processing and Communications Applications Conference*, 2015 Vol: 23, pp. 1849-1852
- [12] Cai F., Fan H., Song Z., Fu Q., “Difference Beam Aided Target Detection in Monopulse Radar”, *Chinese Journal of Aeronautics*, 2015 Vol: 28, pp.1485-1493
- [13] Ahn C., Kim Y., “Point Targeting of Multisatellites via a Virtual Structure Formation Flight Scheme”, *Journal of Guidance, Control, and Dynamics*, 2009 Vol: 32, pp.1330-1344
- [14] Gordon N., Salmond D., “Bayesian Pattern Matching Technique for Target Acquisition”, *Journal of Guidance, Control, and Dynamics*, 1999 Vol:22, pp.68-77
- [15] Morrill C.S., Davies B.L., “Target Tracking and Acquisition in Three Dimensions Using a Two-Dimensional Display surface”, *Journal of Applied Psychology*, 1961 Vol: 45, pp.214-221
- [16] Lee C.J., Pak J.M., Ahn C.K., Min K.M., Shi P., Lim M.T., “Multi-Target FIR Tracking Algorithm for Markov Jump Linear Systems Based on True-Target Decision-Making”, *Neurocomputing*, 2015 Vol: 168, pp.298-307
- [17] Gaxiola L.N., Diaz-Ramirez V.H., Tapia J.J., Garcia-Martinez P., “Target Tracking With Dynamically Adaptive Correlation”, *Optics Communications*, 2016 Vol: 365, pp.140-149
- [18] Wu w., Jiang j., Liu W., Feng X., Gao L., Qin X., “Augmented State GM-PHD Filter With Registration Errors for Multi-Target Tracking by Doppler radars”, *Signal Processing*, 2016 Vol: 120, pp.117-128
- [19] Liu Y., Wang S., Chen Y.Q., “Automatic 3D Tracking System for Large Swarm of Moving Objects”, *Pattern Recognition*, 2016 Vol: 52, pp.384-396
- [20] Shao Q.M., Cinar A., “Coordination Scheme and Target Tracking for Distributed Model Predictive Control”, *Chemical Engineering Science*, 2015 Vol: 136, pp.20-26

- [21] He Y.J., Li M., Zhang J., Yao Y.P., “Infrared Target Tracking Via Weighted Correlation Filter”, *Infrared Physics & Technology*, 2015 Vol: 73, pp.103-114
- [22] Zhang S., Zhu Y., Rot-Chowdhury A., “Tracking Multiple Interacting Targets in a Camera Network”, *Computer Vision and Image Understanding*, 2015 Vol: 134, pp.64-73
- [23] Zawadzki A., Gorgon M., “Automatically Controlled Pan/Tilt Smart Camera with FPGA Based Image Analysis System Dedicated to Real-Time Tracking of a Moving Object”, *Journal of Systems Architecture*, 2015 Vol: 61, pp.681-692
- [24] Wang X., Wang s., “Collaborative signal processing for target tracking in distributed wireless sensor networks”, *Journal of Parallel and Distributed Computing*, 2007 Vol: 67, pp.501-515

EKLER

EK 1: Microsoft Visual C# Ana Program

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge;
using AForge.Video.DirectShow;
using System.Drawing.Imaging;
using System.Threading;
using System.IO;

namespace trackk
{
    public partial class f21 : Form
    {

        public int v { get; set; } public int z { get; set; } public int y { get; set; } public int x { get; set; } string d = "";
        private FilterInfoCollection videoDevices;
        private FilterInfoCollection CaptureDevice;
        private VideoCaptureDevice FinalFrame;
        EuclideanColorFiltering filter = new EuclideanColorFiltering();
        Color color = Color.Black;
        GrayscaleBT709 grayscaleFilter = new GrayscaleBT709();
        BlobCounter blobCounter = new BlobCounter();
        int range = 120;
        public f21()
        {
            InitializeComponent();

            blobCounter.MinWidth = 2;
            blobCounter.MinHeight = 2;
            blobCounter.FilterBlobs = true;
            blobCounter.ObjectsOrder = ObjectsOrder.Size;
            try
            {
                // enumerate video devices
                videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
```

```

        if (videoDevices.Count == 0)
            throw new ApplicationException();

        // add all devices to combo
        foreach (FilterInfo device in videoDevices)
        {
            camerasCombo.Items.Add(device.Name);
        }

        camerasCombo.SelectedIndex = 0;
    }
    catch (ApplicationException)
    {
        camerasCombo.Items.Add("No local capture devices");
        videoDevices = null;
    }

    Bitmap b = new Bitmap(320, 240);
    // Rectangle a = (Rectangle)r;
    Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
    Graphics g2 = Graphics.FromImage(b);
    pen1 = new Pen(Color.FromArgb(255, 0, 0), 3);
    g2.Clear(Color.White);
    g2.DrawLine(pen1, b.Width / 2, 0, b.Width / 2, b.Width);
    g2.DrawLine(pen1, b.Width, b.Height / 2, 0, b.Height / 2);
    pictureBox1.Image = (System.Drawing.Image)b;
}

private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
{
    Bitmap objectsImage = null;
    Bitmap mImage = null;
    mImage = (Bitmap)image.Clone();
    filter.CenterColor = Color.FromArgb(color.ToArgb());
    filter.Radius = (short)range;

    objectsImage = image;
    filter.ApplyInPlace(objectsImage);

    BitmapData objectsData = objectsImage.LockBits(new Rectangle(0, 0, image.Width, image.Height),
        ImageLockMode.ReadOnly, image.PixelFormat);
    UnmanagedImage grayImage = grayscaleFilter.Apply(new UnmanagedImage(objectsData));
    objectsImage.UnlockBits(objectsData);

    blobCounter.ProcessImage(grayImage);
    Rectangle[] rects = blobCounter.GetObjectRectangles();

    if (rects.Length > 0)
    {

```

```

        foreach (Rectangle objectRect in rects)
        {
            Graphics g = Graphics.FromImage(mImage);
            using (Pen pen = new Pen(Color.FromArgb(160, 255, 160), 5))
            {
                g.DrawRectangle(pen, objectRect);
            }

            g.Dispose();
        }
    }

    image = mImage;
}

private void videoSourcePlayer3_NewFrame(object sender, ref Bitmap image)
{
    Bitmap objectsImage = null;

    // set center color and radius
    filter.CenterColor = Color.FromArgb(color.ToArgb());
    filter.Radius = (short)range;
    // apply the filter
    objectsImage = image;
    filter.ApplyInPlace(image);

    // lock image for further processing
    BitmapData objectsData = objectsImage.LockBits(new Rectangle(0, 0, image.Width, image.Height),
        ImageLockMode.ReadOnly, image.PixelFormat);

    // grayscaling
    UnmanagedImage grayImage = grayscaleFilter.Apply(new UnmanagedImage(objectsData));

    // unlock image
    objectsImage.UnlockBits(objectsData);

    // locate blobs
    blobCounter.ProcessImage(grayImage);
    Rectangle[] rects = blobCounter.GetObjectRectangles();

    if (rects.Length > 0)
    {
        Rectangle objectRect = rects[0];

        // draw rectangle around detected object
        Graphics g = Graphics.FromImage(image);

        using (Pen pen = new Pen(Color.FromArgb(160, 255, 160), 5))
        {

```

```

        g.DrawRectangle(pen, objectRect);
    }
    g.Dispose();
    int objectX = objectRect.X + objectRect.Width / 2 - image.Width / 2;
    int objectY = image.Height / 2 - (objectRect.Y + objectRect.Height / 2);
    ParameterizedThreadStart t = new ParameterizedThreadStart(p);
    Thread aa = new Thread(t);
    aa.Start(rects[0]);
}
Graphics g1 = Graphics.FromImage(image);
Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
g1.DrawLine(pen1, image.Width / 2, 0, image.Width / 2, image.Width);
g1.DrawLine(pen1, image.Width, image.Height / 2, 0, image.Height / 2);
g1.Dispose();
}

int sayici = 0;

int xMax = 0;
int yMax = 0;

void p(object r)
{
    try
    {

        Bitmap b = new Bitmap(pictureBox1.Image);
        Rectangle a = (Rectangle)r;
        Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
        Graphics g2 = Graphics.FromImage(b);
        pen1 = new Pen(color, 3);
        // Brush b5 = null;
        SolidBrush b5 = new SolidBrush(color);
        // g2.Clear(Color.Black);

        Font f = new Font(Font, FontStyle.Bold);
        int x = 15;
        int y = 25;

        g2.DrawString("o", f, b5, a.Location);
        g2.Dispose();
        pictureBox1.Image = (System.Drawing.Image)b;
        this.Invoke((MethodInvoker)delegate
        {
            richTextBox1.Text = a.Location.ToString() + "\n" + richTextBox1.Text + "\n";
        });
    }
}

```



```

x = a.Location.X;
y = a.Location.Y;

xMax = pictureBox1.Size.Width;
yMax = pictureBox1.Size.Height;

sayici += 1;

if (sayici >= 0)//0 sayisi arttıkça motorun hızı yavaşlar 0 dayken arduinodaki hız neyse o
{
    sayici = 0;

    if (x > (xMax / 2) + 10)
    {
        serialPort1.WriteLine("x");// servo motoru sağa dönderen harfi yolla
        // serialPort1.WriteLine("d");
        for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
        {
            button12.PerformClick(); //adım motoru sol
        }
    }
    if (x < (xMax / 2) - 10)
    {
        serialPort1.WriteLine("y");//servo motoru sola dönderen harfi yolla
        for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
        {
            button14.PerformClick(); //adım motor sağ
        }
    }
    if (y > (yMax / 2) + 10)
    {
        serialPort1.WriteLine("v");//servo motoru yukarı dönderen harfi yolla
        for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
        {
            button10.PerformClick(); // adım motor yukarı
        }
    }
    if (y < (yMax / 2) - 10)
    {
        serialPort1.WriteLine("z");//servo motoru aşağı dönderen harfi yolla
        for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
        {
            button16.PerformClick(); // adım motor aşağı
        }
    }
}

});
}

```

```

        catch (Exception)
        {
            Thread.CurrentThread.Abort();
        }

        Thread.CurrentThread.Abort();
    }

    private void button1_Click(object sender, EventArgs e)//Kamerayı Açma start
    {

        videoSourcePlayer1.SignalToStop();
        videoSourcePlayer1.WaitForStop();
        videoSourcePlayer2.SignalToStop();
        videoSourcePlayer2.WaitForStop();
        videoSourcePlayer3.SignalToStop();
        videoSourcePlayer3.WaitForStop();
        // videoDevices = null;
        VideoCaptureDevice videoSource = new
        VideoCaptureDevice(videoDevices[camerasCombo.SelectedIndex].MonikerString);
        videoSource.DesiredFrameSize = new Size(320, 240);
        videoSource.DesiredFrameRate = 1;

        videoSourcePlayer1.VideoSource = videoSource;
        videoSourcePlayer1.Start();
        videoSourcePlayer2.VideoSource = videoSource;
        videoSourcePlayer2.Start();
        videoSourcePlayer3.VideoSource = videoSource;
        videoSourcePlayer3.Start();
        //groupBox1.Enabled = false;
    }

    private void f21_FormClosing(object sender, FormClosingEventArgs e)
    {

        videoSourcePlayer1.SignalToStop();
        videoSourcePlayer1.WaitForStop();
        videoSourcePlayer2.SignalToStop();
        videoSourcePlayer2.WaitForStop();
        videoSourcePlayer3.SignalToStop();
        videoSourcePlayer3.WaitForStop();
        groupBox1.Enabled = true;
    }

    private void button2_Click(object sender, EventArgs e)//Kamerayı Kapatma Disconnet
    {

        videoSourcePlayer1.SignalToStop();
        videoSourcePlayer1.WaitForStop();
        videoSourcePlayer2.SignalToStop();

```

```

        videoSourcePlayer2.WaitForStop();
        videoSourcePlayer3.SignalToStop();
        videoSourcePlayer3.WaitForStop();
    }

    private void button3_Click(object sender, EventArgs e)
    {

        colorDialog1.ShowDialog();
        color = colorDialog1.Color;
    }

    private void numericUpDown1_ValueChanged(object sender, EventArgs e)
    {
        range = Convert.ToInt32(numericUpDown1.Value);
    }

    private void numericUpDown2_ValueChanged(object sender, EventArgs e)
    {
        blobCounter.MaxWidth = Convert.ToInt32(numericUpDown2.Value);
    }

    private void numericUpDown3_ValueChanged(object sender, EventArgs e)
    {
        blobCounter.MinWidth = Convert.ToInt32(numericUpDown3.Value);
    }

    private void button5_Click(object sender, EventArgs e) // servo motor 2 nin sola hareketi
    {
        for (int i = 0; i <= Convert.ToInt32(textBox2.Text); i++)
        {
            serialPort1.WriteLine("y");
        }
    }

    private void button6_Click(object sender, EventArgs e) // servo motor 1 nin yukarı hareketi
    {
        for (int i = 0; i <= Convert.ToInt32(textBox2.Text); i++)
        {
            serialPort1.WriteLine("z");
        }
    }

    private void button7_Click(object sender, EventArgs e) // servo motor 2 nin sağa hareketi
    {
        for (int i = 0; i <= Convert.ToInt32(textBox2.Text); i++)
        {

```

```

        serialPort1.WriteLine("x");
    }
}

private void button8_Click(object sender, EventArgs e)// servo motor 1 nin aşağı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox2.Text); i++)
    {
        serialPort1.WriteLine("v");
    }
}

private void button9_Click(object sender, EventArgs e)// adım motorun Sol Yukarı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("ad");
    }
}

private void button10_Click(object sender, EventArgs e)// adım motor 1 in Yukarı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("a");
    }
}

private void button11_Click(object sender, EventArgs e)// adım motorun Sağ Yukarı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("ab");
    }
}

private void button12_Click(object sender, EventArgs e)// adım motorun 2 nin Sola hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("d");
    }
}

private void button14_Click(object sender, EventArgs e)// adım motorun 2 nin Sağa hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("b");
    }
}

```

```

    }
}

private void button15_Click(object sender, EventArgs e)//adım motorun Sol Aşağı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("cd");
    }
}

private void button16_Click(object sender, EventArgs e)// adım motor 1 in Aşağı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("c");
    }
}

private void button17_Click(object sender, EventArgs e)//adım motorun Sağ Aşağı hareketi
{
    for (int i = 0; i <= Convert.ToInt32(textBox1.Text); i++)
    {
        serialPort1.WriteLine("cb");
    }
}

private void button4_Click(object sender, EventArgs e)// Bağlan butonu
{
    serialPort1.PortName = comboBox1.Text;
    if (!serialPort1.IsOpen)
        serialPort1.Open();
    MessageBox.Show("Bağlandı");

    serialPort1.WriteLine("");
    // serialPort1.Close();
}

private void f21_Load(object sender, EventArgs e)
{
    comboBox1.Items.Add("COM1");
    comboBox1.Items.Add("COM2");
    comboBox1.Items.Add("COM3");
    comboBox1.Items.Add("COM4");
    comboBox1.Items.Add("COM5");
    comboBox1.Items.Add("COM6");
    comboBox1.Items.Add("COM7");
    comboBox1.Items.Add("COM8");
}

```

```

comboBox1.Items.Add("COM9");
comboBox1.Items.Add("COM10");
comboBox1.Items.Add("COM11");
comboBox1.Items.Add("COM12");
comboBox1.Items.Add("COM13");
comboBox1.SelectedIndex = 9; // yazılan rakam Formda bir üst rakam olarak çıkar
}

static double[,] Normalize(double[,] mat)
{
    int width = mat.GetLength(0);
    int height = mat.GetLength(1);
    int nPixels = width * height;

    double sum = 0;
    double max = double.MinValue;
    double min = double.MaxValue;
    var grayLevels = new Dictionary<double, int>();

    foreach (var g in mat)
    {
        sum += g;
        if (g > max) max = g;
        if (g < min) min = g;
        if (!grayLevels.ContainsKey(g)) grayLevels[g] = 0;
        ++grayLevels[g];
    }
    double avg = sum / nPixels;
    double range = max - min;

    var I = new double[width, height];

    // how to normalize?

    return I;
}

private void button5_mousedown(object sender, MouseEventArgs e) // Mouse a basılı tuttuğumuz sürece butonların çalışması
{
    timer1.Enabled = true;
}

private void timer1_Tick_1(object sender, EventArgs e) // Mouse a basılı tuttuğumuz sürece butonların çalışması için gereken zamanlayıcı
{
    button5.PerformClick();
}

private void button5_mouseup(object sender, MouseEventArgs e) // Mouse a basmayı bıraktığımız an durması
{
    timer1.Enabled = false;
}

```

```

    }

    private void timer2_Tick(object sender, EventArgs e)
    {
        button7.PerformClick();
    }

    private void button7_MouseUp(object sender, MouseEventArgs e)
    {
        timer2.Enabled = false;
    }

    private void button7_MouseDown(object sender, MouseEventArgs e)
    {
        timer2.Enabled = true;
    }

    private void button6_MouseDown(object sender, MouseEventArgs e)
    {
        timer3.Enabled = true;
    }

    private void button6_MouseUp(object sender, MouseEventArgs e)
    {
        timer3.Enabled = false;
    }

    private void button8_MouseDown(object sender, MouseEventArgs e)
    {
        timer4.Enabled = true;
    }

    private void button8_MouseUp(object sender, MouseEventArgs e)
    {
        timer4.Enabled = false;
    }

    private void timer3_Tick(object sender, EventArgs e)
    {
        button6.PerformClick();
    }

    private void timer4_Tick(object sender, EventArgs e)
    {
        button8.PerformClick();
    }

    private void f21_KeyDown(object sender, KeyEventArgs e)//Butonları Klavyenin yön tuşlarına atama
    {
        if (e.KeyData == Keys.Down)

```

```
{
    button8.PerformClick();
}
if (e.KeyData == Keys.Up)
{
    button6.PerformClick();
}
if (e.KeyData == Keys.Left)
{
    button5.PerformClick();
}
if (e.KeyData == Keys.Right)
{
    button7.PerformClick();
}
}
}
}
```


EK 2: Arduino Programı

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

int pozisyonX=0;
int pozisyonY=0;

char Char;
String x, y, z, i, j;
int val1, val2, val3, val4, val5;           //Adım motorların pinleri
int motorPin1 = 8;
int motorPin2 = 9;
int motorPin3 = 10;
int motorPin4 = 11;
int motorPin5 = 2;
int motorPin6 = 3;
int motorPin7 = 4;
int motorPin8 = 38;

int delayTime = 4;
Servo myservo1;           // create servo object to control a servo
// twelve servo objects can be created on most boards
Servo myservo2;

int pos1 = 0;             // variable to store the servo position
int pos2 = 0;

void setup()
{
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  pinMode(motorPin5, OUTPUT);
  pinMode(motorPin6, OUTPUT);
  pinMode(motorPin7, OUTPUT);
  pinMode(motorPin8, OUTPUT);
  myservo1.attach(6);      // attaches the servo on pin 9 to the servo object
  myservo2.attach(5);      // attaches the servo on pin 9 to the servo object
  Serial.begin (9600);
}
void loop()
{
  if(Serial.available(>0)
  {

    // Char=Serial.readStringUntil(';');
```

```

Char=Serial.read();
if(Char=='a')
{
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
    delay(delayTime);
}
if(Char=='b')
{
    digitalWrite(motorPin5, HIGH);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin8, LOW);
    delay(delayTime);
    digitalWrite(motorPin5, LOW);
    digitalWrite(motorPin6, HIGH);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin8, LOW);
    delay(delayTime);
    digitalWrite(motorPin5, LOW);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin7, HIGH);
    digitalWrite(motorPin8, LOW);
    delay(delayTime);
    digitalWrite(motorPin5, LOW);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin8, HIGH);
    delay(delayTime);
}
if(Char=='c')
{
    digitalWrite(motorPin1, LOW);

```

// 1. Adım motorun ileri sürülüşü

// 2. Adım motorun ileri sürülüşü

// 1. Adım motorun GERİ sürülüşü

```

digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(delayTime);
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
delay(delayTime);
}

```

```

if(Char=='d') // 2. Adım motorun GERİ sürülüşü

```

```

{
digitalWrite(motorPin8, HIGH);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, HIGH);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, HIGH);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, HIGH);
delay(delayTime);
}

```

```

if(Char=='ab') // 1. ve 2. Adım motorun İLERİ sürülüşü

```

```

{
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);

```

```

digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, HIGH);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, HIGH);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin7, HIGH);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, HIGH);
delay(delayTime);
}
if(Char=='ad') // 1. Adım motorun İLERİ ve 2. Adım motorun GERİ sürülüşü
{
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin8, HIGH);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);

```

```

digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, HIGH);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, HIGH);
digitalWrite(motorPin5, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
digitalWrite(motorPin8, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin5, HIGH);
delay(delayTime);
}
if(Char=='cb') // 1. Adım motorun GERİ ve 2. Adım motorun İLERİ sürülüşü
{
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, HIGH);
digitalWrite(motorPin5, HIGH);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, HIGH);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, HIGH);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, HIGH);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, LOW);

```

```

digitalWrite(motorPin7, HIGH);
digitalWrite(motorPin8, LOW);
delay(delayTime);
digitalWrite(motorPin1, HIGH);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin5, LOW);
digitalWrite(motorPin6, LOW);
digitalWrite(motorPin7, LOW);
digitalWrite(motorPin8, HIGH);
delay(delayTime);
}

if(Char=='cd') // 1. Adım motorun GERİ ve 2. Adım motorun GERİ sürülüşü
{
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
    digitalWrite(motorPin8, HIGH);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin5, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin8, LOW);
    digitalWrite(motorPin7, HIGH);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin5, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin8, LOW);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin6, HIGH);
    digitalWrite(motorPin5, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin8, LOW);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin6, LOW);
}

```

```

digitalWrite(motorPin5, HIGH);
delay(delayTime);
}

if(Char=='z') // 1. ve 2. Adım motorların bütün pinlerini SIFIRLA

{
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);
digitalWrite(motorPin1, LOW);
digitalWrite(motorPin2, LOW);
digitalWrite(motorPin3, LOW);
digitalWrite(motorPin4, LOW);

}

if(Char=='y') // 1. SERVO SAĞA
{
myservo1.write(pozisyonX+3); // tell servo to go to position in variable 'pos'
pozisyonX+=3; // waits 15ms for the servo to reach the position
}
if(Char=='x') // 1. SERVO SOLA
{
myservo1.write(pozisyonX-3); // tell servo to go to position in variable 'pos'
pozisyonX-=3;
}

if(Char=='z') // 1. SERVO YUKARI
{
myservo2.write(pozisyonY+3); // tell servo to go to position in variable 'pos'
pozisyonY+=3;
}
if(Char=='v') // 1. SERVO AŞAĞI
{
myservo2.write(pozisyonY-3); // tell servo to go to position in variable 'pos'
pozisyonY-=3;

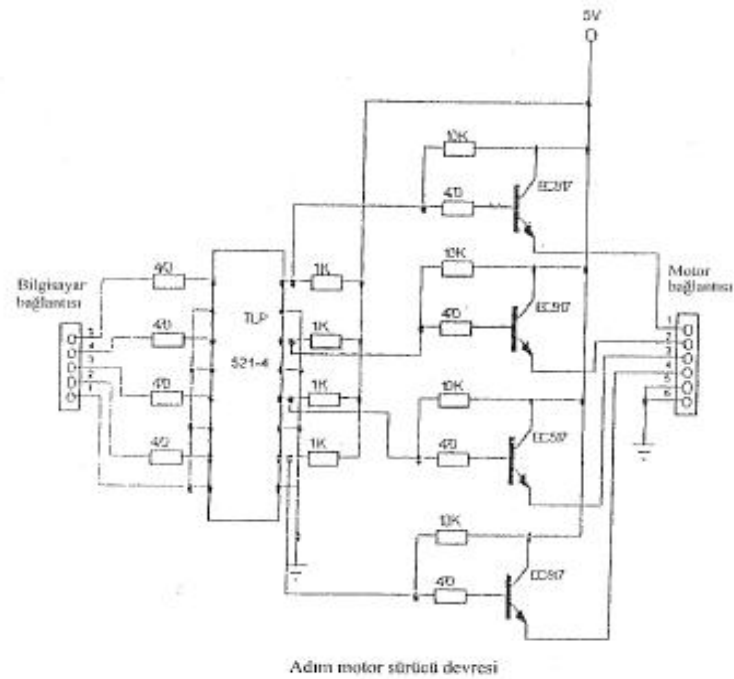
}
} // if serialkontrolünün parantezi
} // void setup in parantezi

```

Tablo Ek Arduino’un işlem gerçekleştirdiği komutlar

Comport vasıtasıyla Arduino’ya gönderilen komutlar	Arduino’ya gelen komutların yaptığı işlem
“z”	1.Servo motorun YUKARI yönde hareketini sağlıyor.
“v”	1.Servo motorun AŞAĞI yönde hareketini sağlıyor.
“y”	2.Servo motorun SOL yönde hareketini sağlıyor.
“x”	2.Servo motorun SAĞ yönde hareketini sağlıyor.
“a”	1. Adım motorun YUKARI yönde hareketini sağlıyor.
“c”	1. Adım motorun AŞAĞI yönde hareketini sağlıyor.
“b”	2. Adım motorun SAĞ yönde hareketini sağlıyor.
“d”	2. Adım motorun SOL yönde hareketini sağlıyor.
“ad”	Adım motorların SOL YUKARI yönde hareketini sağlıyor.
“ab”	Adım motorların SAĞ YUKARI yönde hareketini sağlıyor.
“cd”	Adım motorların SOL AŞAĞI yönde hareketini sağlıyor.
“cb”	Adım motorların SAĞ AŞAĞI yönde hareketini sağlıyor.

EK 3: Kullanılan Elektronik Sürücü Kartın Blok Şeması

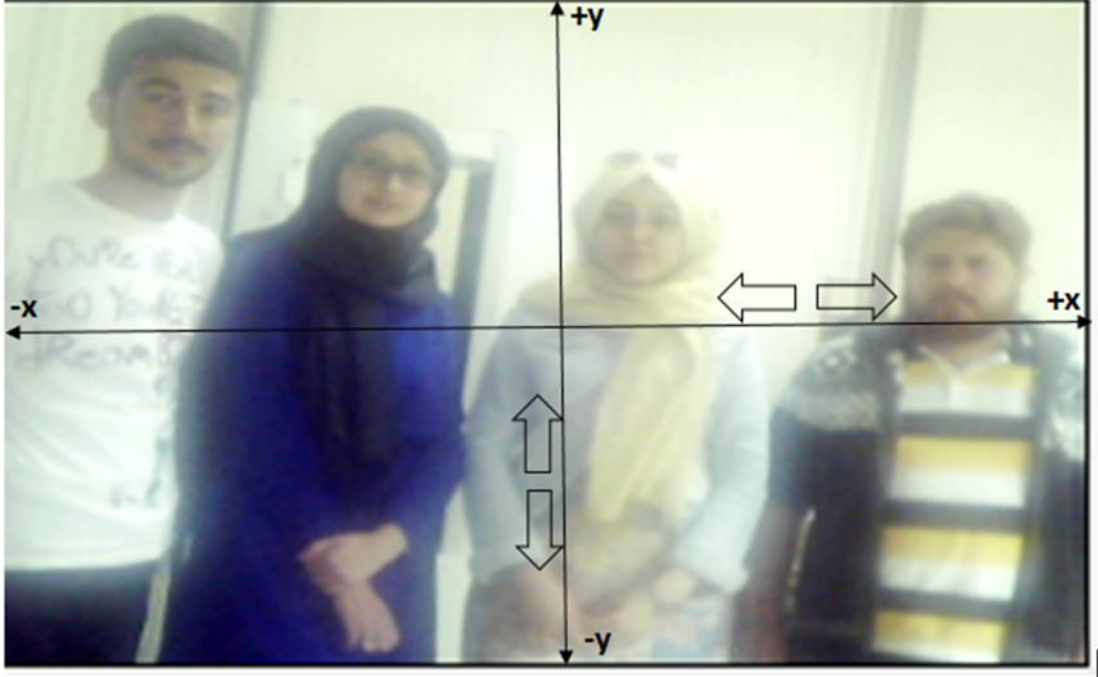


TLP 521-4 (Opto isolator)



Şekil Ek Sürücü Kart Şeması

EK 4: Microsoft Visual C# Programı Ara Yüzünde Alınan Görüntünün Eksenleri



Şekil Ek Görüntüdeki Eksenler

ÖZGEÇMİŞ

1

Kimlik Bilgileri

Adı Soyadı: Cevahir Gündüz

Baba Adı: Seyfi

Anne Adı: Ünzile

Doğum Yeri: İskenderun

Doğum Yılı: 30.10.1994



İlk ve orta öğrenimini İskenderun'da tamamladıktan sonra 2012 yılında Niğde Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümünde lisans öğrenimine başladı. 2016 yılında lisans öğrenimine devam etmektedir.

Haberleşme Bilgileri

Adres: Konacık Mah. Liman Sok. No:120 Arsuz/ Hatay

Telefon: (+90) 530 500 25 08

E-posta: cevahir_gunduz@hotmail.com

ÖZGEÇMİŞ

2

Kimlik Bilgileri

Adı Soyadı: Engin Aydın

Baba Adı: Muammer

Anne Adı: Lütfiye

Doğum Yeri: Melikgazi

Doğum Yılı: 12.03.1993



İlk ve orta öğrenimini Kayseri’de tamamladıktan sonra 2012 yılında Niğde Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümünde lisans öğrenimine başladı. 2016 yılında lisans öğrenimine devam etmektedir.

Haberleşme Bilgileri

Adres: Turgut Reis Mah. Turgut Reis Cad. Eras Sitesi B Blok 9/15 Kocasinan/Kayseri

Telefon: (+90) 506 791 3180

E-posta: enginaydin93@hotmail.com

ÖZGEÇMİŞ

3

Kimlik Bilgileri

Adı Soyadı: Hanife Demirkaya

Baba Adı: Ayhan

Anne Adı: Perihan

Doğum Yeri: Kırıkkale

Doğum Yılı: 15.03.1994



İlk ve orta öğrenimini Kırıkkale’de tamamladıktan sonra 2012 yılında Niğde Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümünde lisans öğrenimine başladı. 2016 yılında lisans öğrenimine devam etmektedir.

Haberleşme Bilgileri

Adres: Yeni Mah. Sağlık Cad. Yunus Emre Apt. No: 30/15 Kırıkkale

Telefon: (+90) 546 739 4674

E-posta: hdk.71@hotmail.com