

T.C.
ÖMER HALİSDEMİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ

FIRÇALI DC MOTOR SÜRÜCÜ TASARIMI

Davut YÜKSEKKAYA
130607039

BİTİRME TEZİ

Yrd. Doç. Dr. Mehmet Kürşat YALÇIN

NİĞDE, 2017

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

Bölümümüz 130607039 numaralı öğrencisi Davut YÜKSEKKAYA 'nın, FIRÇALI DC MOTOR SÜRÜCÜ TASARIMI 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. Mehmet Kürşat YALÇIN

Üye : Yrd. Doç. Dr. İlyas KACAR

Üye : Öğr. Gör. Mehmet Ali EROĞLU

Tezin savunulduğu Tarih: 19/06/2017

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 V.
Doç. Dr. Murat BARUT

DOĞRULUK BEYANI

Bitirme tezi olarak sunduğum bu çalışmayı tüm akademik kurallara ve Ömer Halisdemir Üniversitesi Yayın Etiği Komisyonu Yönergesi'ne uygun olarak gerçekleştirdiğimi ve sunduğumu; bu kurallar ve ilkelere aykırı hiç bir yol ve yardıma başvurmaksızın bizzat hazırladığımı beyan ederim.

Tezimle ilgili yaptığım beyana aykırı bir durum saptanırsa ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımı bildiririm. 17/06/2017

İmza

Davut YÜKSEKKAYA

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesi sürecinde beni yönlendiren, teşvik eden ve kıymetli bilgi, birikim, tecrübeleriyle desteğini esirgemeyen değerli danışmanım Sayın Yrd. Doç. Dr. Mehmet Kürşat YALÇIN' a, yardımlarını benden esirgemeyen Bauyrzhan ANARBAY' e sonsuz teşekkür ve saygılarımı sunarım.

Lisans eğitimim boyunca beraber çalıştığımız Halil Bahadır AKYILDIZ, Muhammed Ali ÇALIŞIR ve diğer çalışma arkadaşlarıma teşekkürü bir borç bilirim.

Hayatım boyunca maddi manevi her anlamda desteğini benden esirgemeyen aileme sonsuz teşekkür ederim.

ÖZET

FIRÇALI DC MOTOR SÜRÜCÜ TASARIMI

Davut YÜKSEKKAYA

Ömer Halisdemir Üniversitesi

Mühendislik Fakültesi

Mekatronik Mühendisliği Bölümü

Danışman: Yrd. Doç. Dr. Mehmet Kürşat YALÇIN

Bu projede, fırçalı bir DC motorun hız ve pozisyon kontrolü PID kontrol algoritması kullanılarak gerçek zamanda gerçekleştirilmiştir. Sistem girişine uygulanan bir referans girişi kontrol algoritması ile istenilen referans değerinde tutulmaya çalışılmıştır. Sistem için gerekli olan kodlar Arduino IDE ortamında yazılmıştır. Ayrıca sistem, bir motora bağlanarak MATLAB ortamında matematiksel modellenmesi yapıp transfer fonksiyonu üzerinden sistemin durum uzayı modeli çıkarılmış olup sistemin davranışı, gerçek sistem ile grafiksel karşılaştırılması yapılmıştır.

Anahtar Kelimeler: motor sürücü, pozisyon kontrolü, hız kontrolü

İÇİNDEKİLER

BİTİRME TEZİ KABUL VE ONAY BELGESİ.....	i
DOĞRULUK BEYANI	ii
TEŞEKKÜR.....	iii
ÖZET.....	iv
İÇİNDEKİLER	v
KISALTMALAR	vii
ÖNSÖZ	viii
BÖLÜM I.....	1
1.1. Giriş.....	1
1.2. Çalışmanın Amacı.....	1
1.3. Konunun Önemi	2
1.4. İş - Zaman Çizelgesi.....	2
BÖLÜM II.....	5
2.1. Literatür Taraması.....	5
BÖLÜM III	6
3.1. Sürücü Tasarımında Kullanılan Elektronik Kartlar	6
3.2. Motor Sürücü Elektronik Kartların Çizimi	9
3.3. Motor Sürücü Devre Şeması	12
3.4. DC MOTORUN SİSTEM DİNAMİĞİ.....	23
3.4.1. DC Motorun Transfer Fonksiyonu.....	23
3.4.2. MATLAB ile DC Motorun Transfer Fonksiyonunu Çıkarma	25
3.5. PID Kontrol Algoritması.....	28
3.5.1. Oransal (Proportional) Terim	31
3.5.2. İntegral (Integral) Terimi	32
3.5.3. Türev (Derivative) Terimi.....	32
3.5.4. PID Kontrol Algoritması.....	33
3.5.5. PID Parametrelerinin Belirlenmesi	33
3.5.5.1. Ziegler-Nichols Metodu	34
3.5.5.2. Deneme-Yanılma Yoluyla Parametrelerin Bulunması.....	35
3.6. Sistemin Kontrolü	37
3.6.1. Kontrolöre CPU Üzerinden Referans Gönderme.....	41
BÖLÜM IV	44
4.1. Bütçe Planı	44

BÖLÜM V	45
5.1. Sonuç.....	45
5.2. Motor Sürücünün Entegre Edildiği Farklı Platformlar	49
KAYNAKÇA.....	52
ÖZGEÇMİŞ	54

KISALTMALAR

A	: Amper
AA	: Alternatif Akım
AC	: Alternative Current
CCW	: Counter Clock Wise
CW	: Clock Wise
DA	: Doğru Akım
DC	: Direct Current
ID	: Identification
PCB	: Printed Circuit Board
PID	: Proportional Integral Derivative
PWM	: Pulse Width Modulation
RPM	: Revolutions Per Minute
SMD	: Surface Mount Device
TTL	: Transistor Transistor Logic
V	: Volt

ÖNSÖZ

DC motorlar kullanımı kolay olduğundan endüstride sıkça kullanılmaktadır. Fakat yalnızca DC motor istenilen ihtiyacı karşılamamaktadır. İhtiyaç duyulan eksikliklere cevap verebilmesi için bir kontrolör dahilinde kullanılması gerekmektedir. Piyasada birçok kontrolör bulunsa da çoğunun çalışma stili aynıdır. Tasarımı pek karmaşık olmadığından genellikle PID kontrolör tercih edilmektedir.

Endüstrideki eksiklikler göz önünde bulundurularak bir DC motor sürücü tasarımına gidilmesi düşünüldü. Fakat piyasadaki diğer kontrolörlere göre dizayn edilen bu kontrolör maliyet açısından oldukça karlıdır. Piyasadaki diğer motor sürücülerin hemen hemen çoğunda bulunan birçok özellik bu motor sürücüsüne de eklenmiştir. Hatta kendi projelerimizde kullanılmak üzere piyasadaki sürücülerde bulunmayan bir takım özellikler de bu sürücüye eklenmiştir. İstenilen şekilde modifiye edilebilen bu sürücünün ilerde geliştirilebilir olduğu açıkça görülmekte olup daha sonradan dizayn edilmesi düşünülen fırçasız motor sürücünün de temelini oluşturmaktadır. Eksik görülen noktalar giderildikten sonra fırçasız motorlar için bir üst versiyonunun tasarımına gidilmesi düşünülmektedir.

BÖLÜM I

1.1. Giriş

DC motorlar, direkt olarak DC güç kaynağından beslenebilen elektrik makineleridir. Kullanım alanları oldukça geniş olmakla beraber AC motorlara göre daha ucuzdurlar. Yapılarının basit oluşunun yanı sıra kullanımları da oldukça basit olduğundan sıkça kalkış, durma, frenleme ve devir yönü değişimi gerektiren çeşitli endüstriyel uygulamalarda yaygın bir şekilde kullanılmaktadır. Son yıllarda teknolojik gelişmelerle birlikte ev aletleri uygulamalarında, düşük güçlü ve düşük maliyet istenen ayarlanabilir hız gereken yerlerde yaygın bir kullanım alanı bulmuştur. Geniş kullanım alanı bulmasının diğer bir sebebi de AC motorlarına göre kontrolünün daha kolay olmasıdır. AC motor sürücüleri ile kıyaslandığında, DC motor sürücü devrelerinin basit ve ucuz olması, ayarlanabilir hız uygulamalarında DC motorunu ön plana çıkarmıştır. Fakat istenilen şekilde çalıştırılabilmesi için bir kontrolöre ihtiyaç duyarlar. Bu kontrolörün görevi sistemin çıkışını, sisteme girilen referans değerinde tutmaya çalışmaktır. Böyle bir kontrolörü dizayn etmenin birçok yöntemi bulunmaktadır. Bu yöntemler içerisinde en yaygın olanı PID kontrol algoritması ile kontrolör dizaynıdır. Bu çalışmada da PID kontrol algoritması kullanılarak bir fırçalı DC motor sürücü tasarımı yapılmıştır.

1.2. Çalışmanın Amacı

Bu çalışmada PID kontrol algoritması kullanılarak sabit mıknatıslı bir DC motorun konum ve hız kontrolü yapılmıştır. Ayrıca motorların çektiği akım ölçülerek, herhangi bir zorlanma durumunda tespiti yapıp gerekli tedbirlerin alınması amaçlanmıştır. Öncelikli olarak sürücünün kodları Arduino IDE ortamında yazılıp sonrasında MATLAB ortamında sürücünün bilgisayardan kontrolü için bir grafiksel arayüz hazırlanmıştır. Çalışma sonucunda elde edilen veriler MATLAB ortamında çizdirilerek grafiksel olarak sonuçlar gözlemlenmiştir.

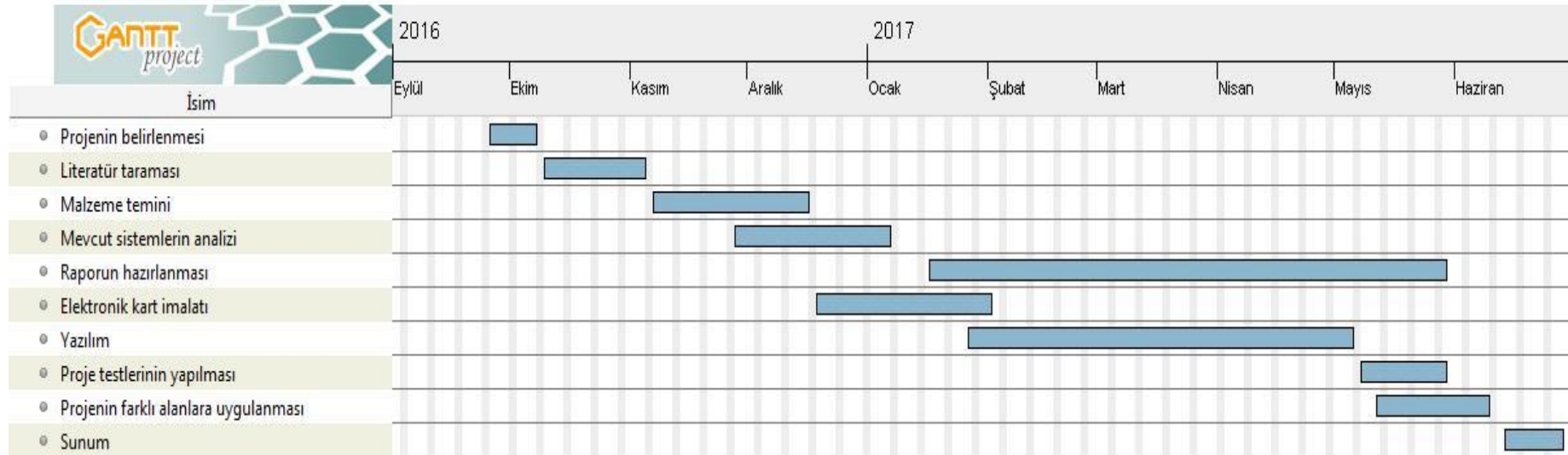
1.3. Konunun Önemi

PID kontrol tekniği, dinamik sistemlerin kontrolünde yaygın olarak kullanılmaktadır. Dinamik kontrollerin %85' i PID kontrol tabanlıdır. PID kontrol tekniği, uygulamadaki basitliğinden dolayı çeşitli endüstriyel işlemlerde kullanılmaktadır. İlk uygulamaları pnömatik sistemlerde, vakumlama aletlerinde ve katı durum analog elektroniğinde görülmektedir. Daha sonra mikroproseslerin dijital uygulanması kullanılmaya başlanmıştır. Belirli bir ayarlama kriterine göre PID kontrol parametrelerinin belirlenmesi, PID kontrolörünün tasarımının en önemli konularındandır. PID kontrolör yapısının basit olması ve kararlı çalışmasının yanında matematiksel modele ihtiyaç duyması ve doğrusal olmayan sistemlerin kontrolünde başarılı olamaması bu kontrolörün dezavantajlarındandır. Bu projede de PID kontrol algoritması kullanılarak sistemin kendisine girilen bir referans değerine sistemin pozisyon ve hız kontrolü yapılarak oturması sağlanmıştır.

1.4. İş - Zaman Çizelgesi

Proje için planlanan iş paketleri ve iş - zaman çizelgesi şekil 1.1 'de verilmiştir. Planlanan iş paketleri herhangi bir aksaklık olmadan başarı ile gerçekleştirilmiştir. Şekil 1.1 'de görüldüğü üzere proje seçimine Eylül ayının sonunda başlanmıştır. Yaklaşık iki hafta süren karar verme evresinden sonra motor sürücü tasarımına karar verildi. Ardından konu ile alakalı gerekli araştırmalar ve literatür taramasına başlandı. Yaklaşık bir aylık bir süreçten sonra gerekli bilgiler edinildiğinden dolayı malzeme seçimine başlandı. Malzemelerin bir kısmı yurtiçinden bir kısmı ise yurtdışından sipariş edildi. Bundan dolayı malzeme tedarik süresi uzun sürmüş olup yaklaşık 1,5 ay civarındadır. Malzemelerin siparişi verildiğinde konu ile alakalı yapılan çalışmalar ve projeler incelenmeye başlandı. Daha önce yapılan projelerin artı ve eksileri belirlenerek, bu projeye eklenecek özellikler belirlendi. Aynı zamanda malzemelerin gelmesinin ardından hemen gerekli devre breadboard üzerinde kurulum analizi yapıldıktan sonra, elektronik kartın imalatına başlandı. İmalat süresi yaklaşık olarak 1,5 ay sürmüştür. Elektronik kartın imalatıyla beraber aynı zamanda raporun yazımına başlanmış olup, gerekli yerlerde küçük notlar alınmıştır. Rapor yazma işlemi bir ana sıkıştırılmayıp uzun bir süre

içerisinde projeye beraber yazılmıştır. Elektronik kartın imalatı bitmeden, sistemin yazılımı yazılmaya başlandı. Projenin en önemli kısmı olan bu evre yaklaşık olarak 3,5 ay sürmüştür. Yazılımı tamamlanan proje test aşamasına geçirilip gerekli tablo ve grafikler çıkarılıp karşılaştırılması yapılmıştır. Aynı zamanda proje yalnızca test aşamasında kalmayıp farklı projelere entegre edilip, başarılı bir şekilde sonuçlandırılmıştır. Haziran ayında ise yaklaşık 2 hafta gibi bir sürede sunum için gerekli hazırlıklar yapıldı.



Şekil 1.1. İş – Zaman Çizelgesi

BÖLÜM II

2.1. Literatür Taraması

Yapılan araştırmalarda elde edilen bilgilere göre, fırçalı DC motor sürücülerini için çeşitli algoritmalar kullanılmakta olup üzerinde çalışmalar da yapılmaktadır. Yapılan araştırma sonuçları aşağıda verilmiştir.

John J. Schachte ve Geoffrey A. Ross (1979) motor miline bir enkoder bağlayarak motor hızını gösteren bir sayaca bağlayıp, hız ve pozisyon hesaplamaları için sayaçtaki sayıyı periyodik olarak örneklemişlerdir. Wang Zhongmin ve Ye Hong (2000) otomobil endüstrisinde yükselen talebi karşılamak için DC motor sürücüsünü bir çip içine entegre ederek yüksek verimlilik, basit konfigürasyon ve aşırı voltaja karşı korumayı amaçlamaya yönelik çalışmalar yapmıştır. Damon Germanton ve Menahem Lehr (1991) DC motora sahip bir elektrikli alete bağladığı bir düzenek ile sistem üzerindeki basınçla orantılı olarak çıkış gerilimi üretilip, hız ve torkunu kontrol etmişlerdir. Harashima ve arkadaşları (1982) oransal P denetleyicisini DC motorun konum denetleyicisini yük altında sabit tutmaya çalışmıştır. Aström ve Wittenmark (1984) sürtünmenin doğrusal olmayan etkilerinin uyarlanabilir şekilde telafi edildiği bir kontrol şeması önererek, bu sürtünmenin hesaba katıldığı zaman motor torkunun yaklaşık olarak bir sabit katsayılı doğrusal model ile tanımlanabileceğini ve bu model için bir kontrolör geliştirilebileceğini belirtmişlerdir. Bunun sonucunda endüstriyel robotlarda bulunduğu gibi birçok hassas işlemlerde kullanılabilecektir.

PID denetleyicisi kolay tasarım ve karmaşık hesaplama işlemleri olmadığından dolayı endüstride sıklıkla kullanılmaktadır. Bununla birlikte, başarılı bir endüstriyel uygulama yapmak için PID kontrolörünün genellikle önceden bir manuel olarak ayarlanmaya ihtiyacı vardır (Chun-Fei Hsu, Bore Kuen Lee, 2011).

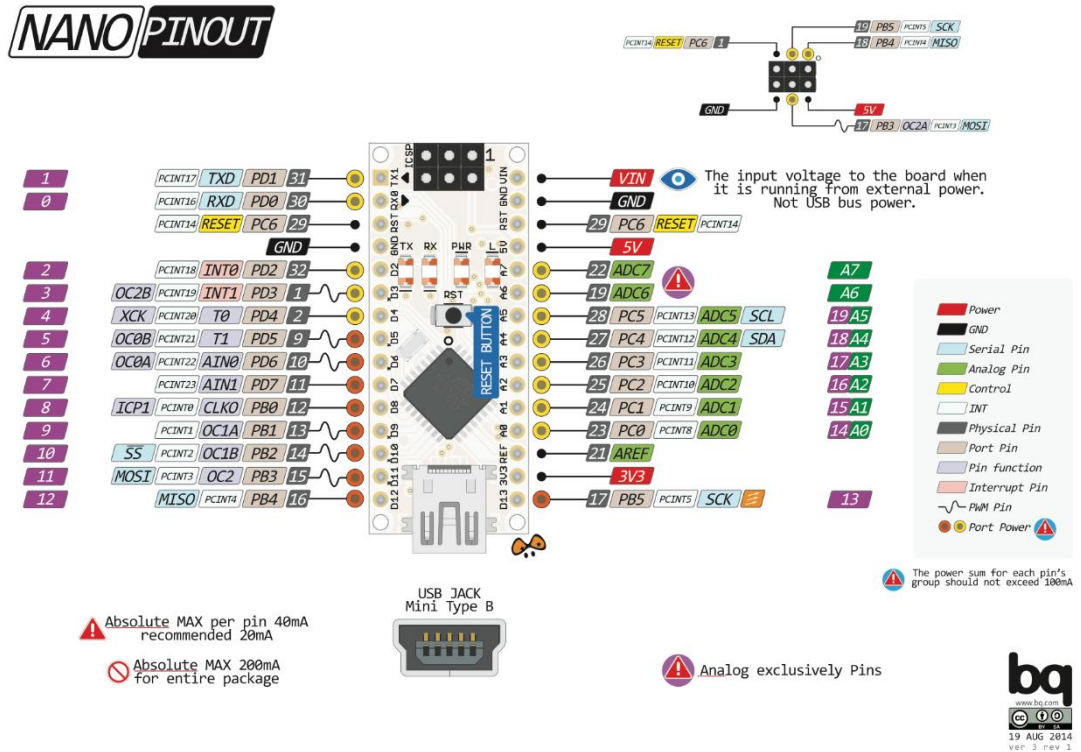
Fuse H. (2005) sinüs fonksiyonu ile çalıştırılan ve rotorun konumunu tespit etmek için statora göre kaydırmalı olarak monte edilmiş elemanlara sahip bir DC motorunda, motoru çalıştırırken açma aşamalarını değiştirme ve frenlemeden kaçınarak başlangıç torkundaki bir azalmayı ortadan kaldırmayı amaçlamaya yönelik çalışmalar yapmıştır.

BÖLÜM III

3.1. Sürücü Tasarımında Kullanılan Elektronik Kartlar

Bu çalışmada, projeyi kısa sürede tamamlayabilmek için hazır shieldler kullanılmıştır. Ancak incelendiğinde bu shieldlerin fazla karmaşık yapısı olmadığı, yalnızca uğraştırıcı olduğu görülmektedir. Süre sıkıntısı olmadığı zamanlarda bu shieldlerin tümünün aynı PCB üzerinde basılıp daha küçük ebatlarda ve daha kararlı çalışabilecek bir sürücünün yapılması mümkündür.

Bu projede kullanılan sabit mıknatıslı fırçalı DC motoru sürebilmek için Arduino Nano kartı kullanılmıştır. Bu kartın tercih edilme sebebi, sistem için gerekli olan dijital ve analog giriş çıkışların yeterli sayıda üzerinde mevcut olmasının yanı sıra kart mekanik boyut açısından küçük olduğundan fiziksel olarak az yer kaplamaktadır.



Şekil 3.1. Arduino Nano Pin Giriş ve Çıkışları

Şekil 3.1 'de Arduino Nano kartının dijital ve analog giriş-çıkışları ile PWM çıkışlarının haricinde diğer pinlerin sahip olduğu özellikler ve kullanılabileceği diğer fonksiyonlar da görülmektedir.

Bu kartın sahip olduğu teknik özellikler tablo 3.1 'de verilmiştir.

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Tablo 3.1. Arduino Nano Teknik Özellikler

Motor sürücünün tasarımı sırasında KORMAS firmasının 631 034 13 kodlu, çift kanallı 1500 pulse/tur veren optik enkoderli fırçalı DC motoru kullanılmıştır. Bu motor 24 volt ile çalışmakta olup, 15:1 redüksiyon oranında 230 rpm ve 140 wattır.

Enkoderler dönme hareketini dijital sinyallere dönüştürüp (öteleme hareketini algılayan özel lineer enkoderler de bulunmaktadır), hareket hakkında birtakım

bilgiler veren bir geri besleme elemanıdır. Bu geri besleme elemanları mutlak ve artımlı olmak üzere iki kısma ayrılmaktadır.

Mutlak enkoderlerin artımlı enkoderlerden tamamen farklı bir çalışma mantığı vardır. Bu enkoderler genellikle konum bilgisi almak için kullanılırlar ve her pozisyonun enkoder içerisinde belirli bir şekilde kodlanmış durumu mevcuttur. Herhangi bir güç kesintisinde enkoder mekanik olarak hareket ettirilmesi durumunda bile enkodere besleme yapıldığında bulunduğu pozisyonu bilir.

Artımlı enkoderlerin ise mekanik ve manyetik çeşitleri bulunmakla beraber genellikle optik olanları tercih edilmektedir. Optik enkoderler genellikle diskler üzerine açılmış yarıklara karşılıklı fototransistör yerleştirilmesi sonucu alınan sinyalin kesilmesi ile elde edilen bilgiler dahilinde kullanılırlar. Bu tip enkoderlerin tek kanallı, iki kanallı ve üç kanallı olanları mevcuttur.

Tek kanallı olan artımlı enkoderler yalnızca pozisyon ve hız bilgisi vermektedirler. Fakat iki kanallı olanlar pozisyon ve hız bilgisinin yanında ilave olarak yön bilgisi de vermektedir. Üçüncü kanalda ise yalnızca tek yarık bulunmakla beraber yalnızca enkoderin tur sayısı hakkında bilgi verir.

Bu çalışmada çift kanallı optik artımlı enkodere sahip motor kullanılmakta olup, sürücü de çift kanala sahip olan motorlar için dizayn edilmiştir. Fakat sürücü kısa süreli bir çalışma ile herhangi bir mekanik veya elektronik müdahale yapılmadan yalnızca var olan kodlarda küçük bir değişiklik sonucunda tek kanallı ve üç kanallı enkoderlere de uyumlu hale getirilebilir.

Sürücünün tasarımında Arduino IBT-2 half-bridge güç katı kullanılmıştır. Bu güç katı aracılığı ile motor 6 volt ile 27 volt arasında beslenebilmektedir. Yüksek performansa sahip olan bu güç katı maksimum 43 ampere kadar çıkabilmektedir. Bu özelliğinin yanı sıra motorun CW ve CCW yönlerinde kontrolü için tek pin kullanmayıp ayrı ayrı iki pin üzerinden kontrol edilebilme gibi avantajı da bulunmaktadır. Ayrıca bu güç katı üzerinde bulunan iki pin üzerinden motorun çektiği akım da hesaplanabilmektedir.

Sürücü üzerindeki komponentleri daha stabil besleyebilmek için bir mini ayarlanabilir LM2596 voltaj regülatörü kullanılmıştır. Bu regülatör kartı 4 volt ile

35 volt arasındaki gerilimi 1.25 volt ile 30 volt arasında istenilen değere dönüştürüp, çıkış pininden maksimum 3 amper verebilmektedir. Bu projede regülatör kartı 5 volta ayarlanıp gerekli komponentlerin beslemesi yapılmaktadır.

Sürücünün birden fazla sürücü ile iletişime geçebilmesi için RS-485 haberleşme protokolü kullanılmıştır. Bu haberleşme için ise MAX485 hazır shieldi kullanılmaktadır. Bu shield 32 cihaz arasında haberleşme yapabilme kapasitesine sahip olduğundan motor sürücü slave modda 32 farklı sürücü ile haberleşebilmektedir.

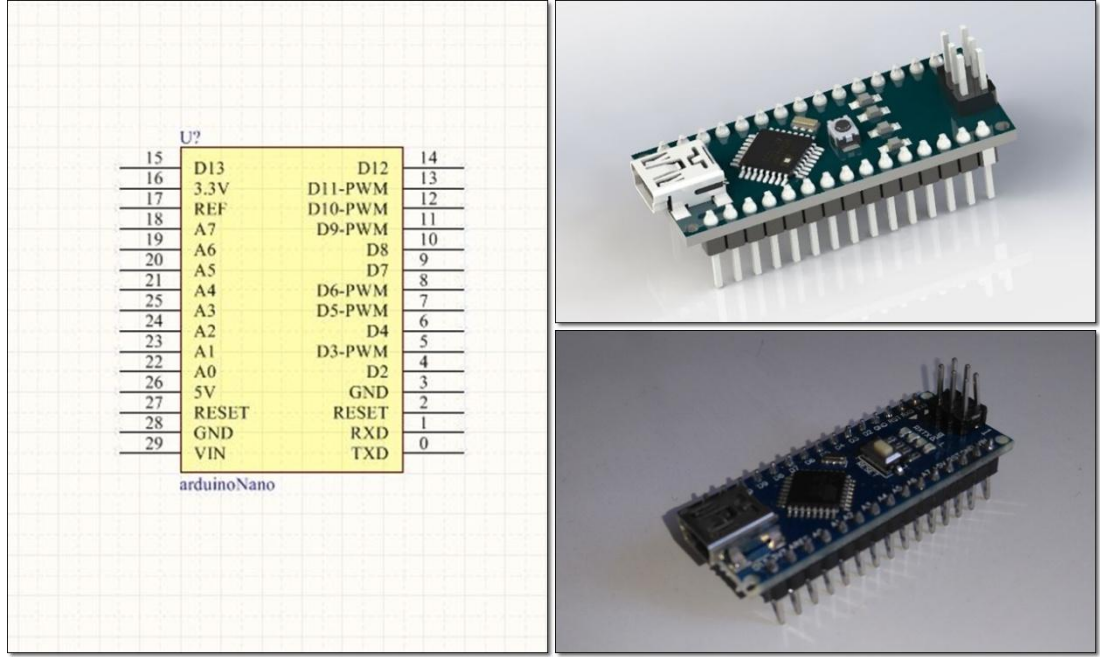
3.2. Motor Sürücü Elektronik Kartların Çizimi

Sürücüde kullanılan komponentlerin hazır elektronik kartlardan oluştuğu daha önce yukarda belirtilmişti. Dolayısıyla motor sürücü dizayn edilirken, şematik ve PCB çizimleri olmadığından bu kartların çizimi motor sürücü projesi dahilinde yapılmıştır.

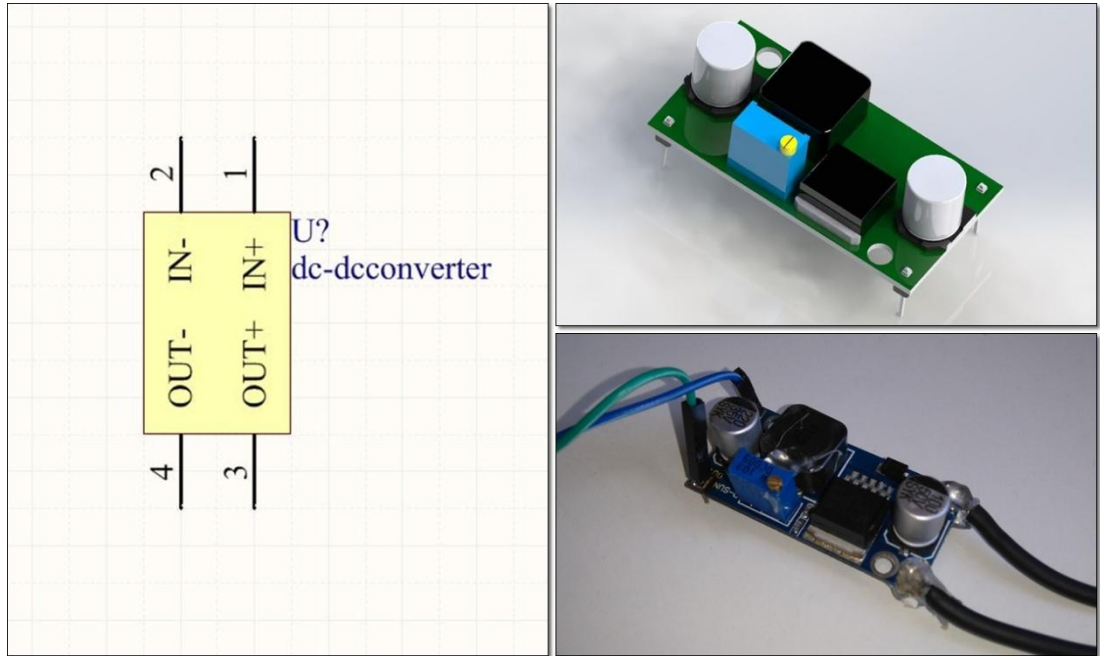
Bu projenin elektronik çizim ve tasarım çalışması Altium Designer 13.1 programı kullanılarak yapılmıştır. Elektronik kartların 3D çizimleri ise Dassault firmasının Solidworks isimli programı kullanılarak yapılmıştır.

Elektronik kartların mekanik ölçüleri hassas olabilmesi için kumpas kullanılarak alınmıştır. Bu ölçülerin hassasiyetinin yüksek olması çok önemlidir. Çünkü mekanik olarak kartın dış ölçüleri ile pinler arası mesafeden kaynaklanan çok küçük bir hata, o kartın PCB imal edildikten sonra yerine oturmasını engelleyecektir.

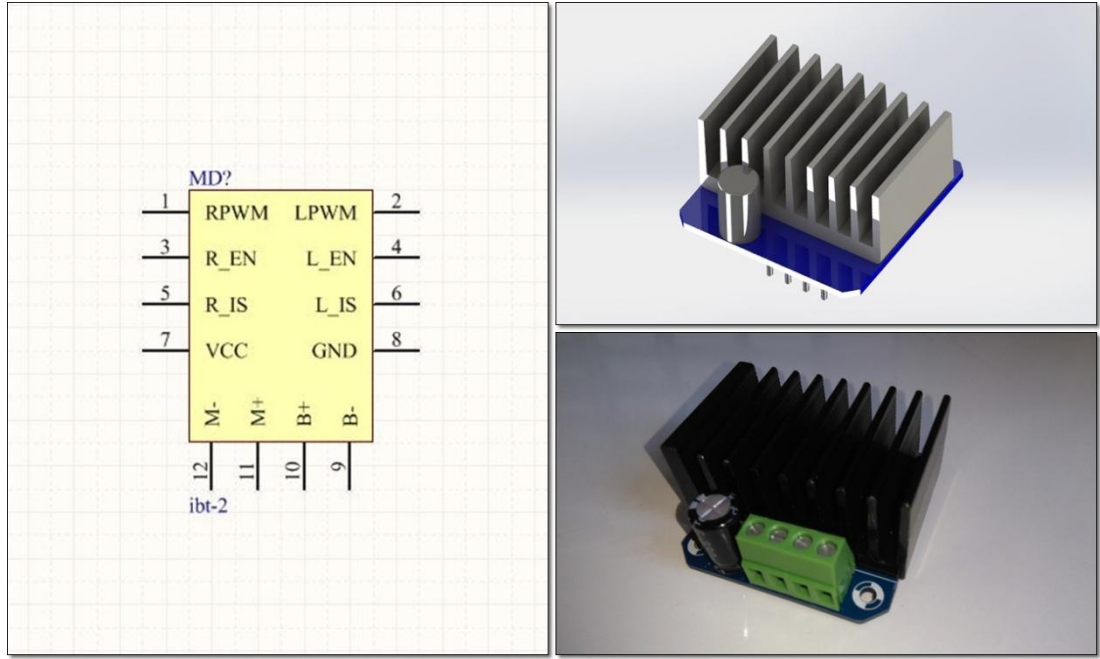
Motor sürücü projesi dahiline kullanılan komponentlerin şematik, 3D ve gerçek görüntüleri aşağıda verilmiştir.



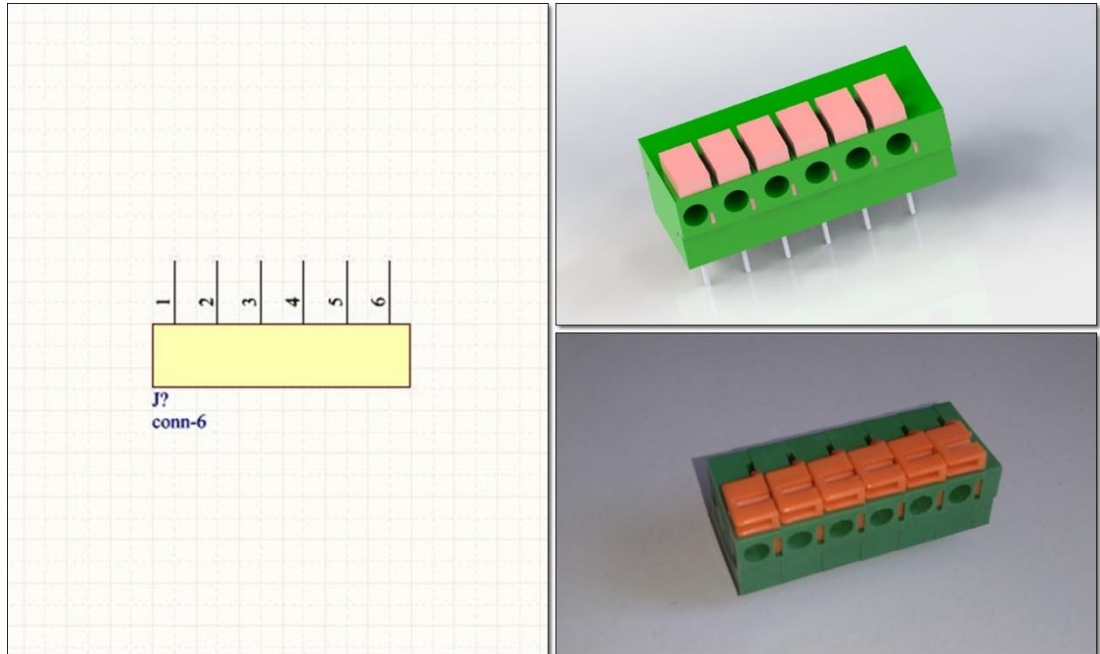
Şekil 3.2. Arduino Nano Şematik, 3D ve Gerçek Görüntüsü



Şekil 3.3. DC-DC Dönüştürücü Şematik, 3D ve Gerçek Görüntüsü

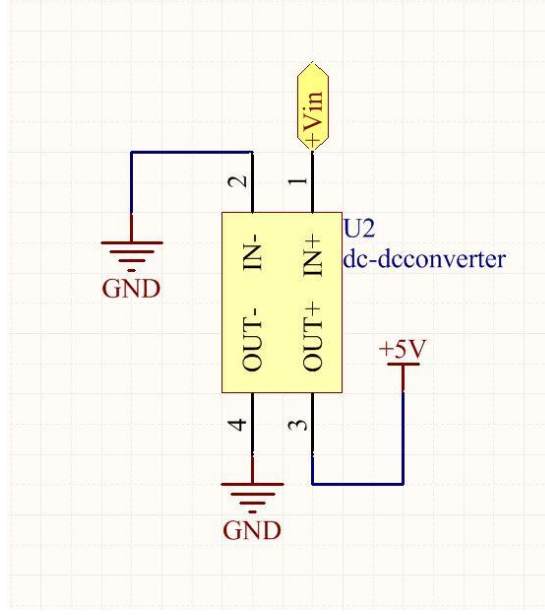


Şekil 3.4. IBT-2 Şematik, 3D ve Gerçek Görüntüsü



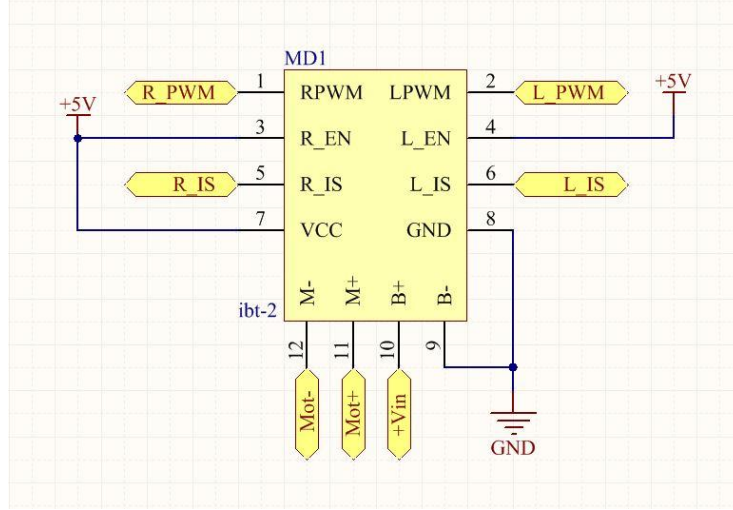
Şekil 3.5. Yaylı Klemens Şematik, 3D ve Gerçek Görüntüsü

3.3. Motor Sürücü Devre Şeması



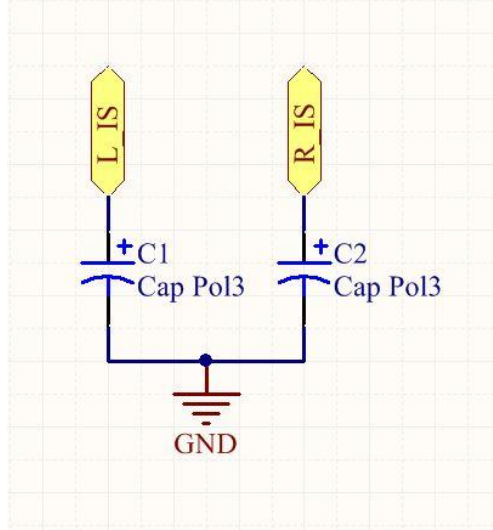
Şekil 3.6. Yaylı Klemens Şematik Görünümü

DC-DC dönüştürücünün çıkışı, üzerinde bulunan trimpot ile +5V değerine sabitlenerek trimpotun üzeri sıcak silikon ile kapatıldı. Bunun yapılma nedeni ise karta daha sonradan herhangi bir fiziksel müdahale yapılmasının sonucunda bütün motor sürücü kartının yanmasına sebep olmasından kaynaklanmaktadır. Bunun dışında kartta meydana gelebilecek herhangi bir mekanik şok, titreşim, farklı cisimlerin teması vs. bu ayarı değiştirebilmesinden de kaynaklanmaktadır. Motor sürücü kartının tüm beslemesi bu dönüştürücünün girişinden ve çıkışından gerçekleştirilmektedir. Giriş gerilimi bu kart için +24V olarak belirlenmiştir.



Şekil 3.7. IBT-2 Şematik Görünümü

Motorun güç katı olarak kullanılan IBT-2 nin üzerindeki pinler yukarda görülmektedir. Toplam 12 pini bulunmaktadır (normalde 8 pin ve 4 klemensi bulunan bu komponent, motor sürücü kartına uyarlanarak klemensler sökölüp yerine pin takılmıştır).

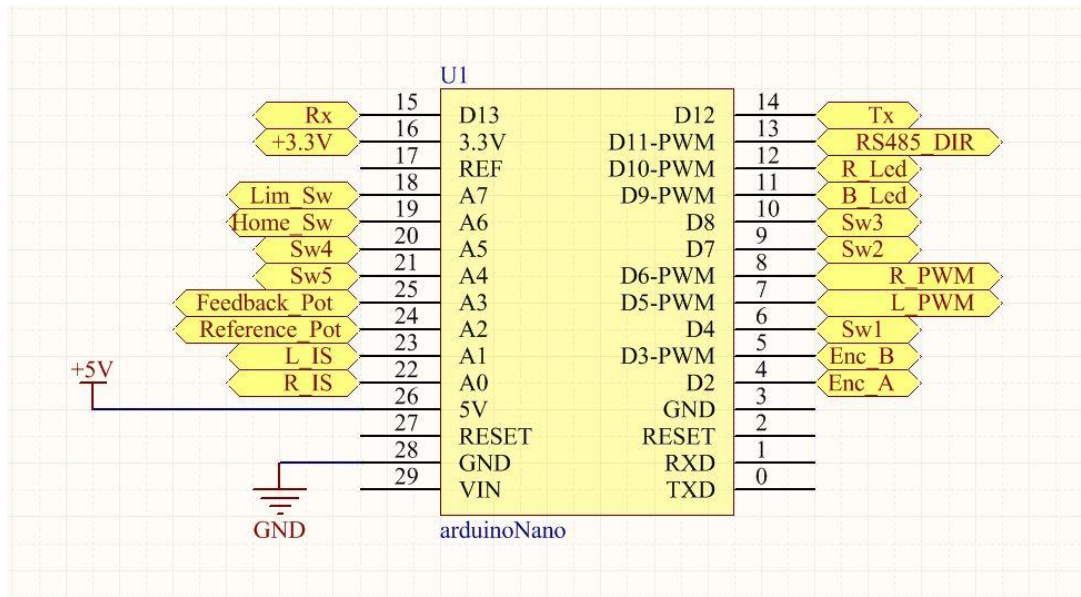


Şekil 3.8. Kapasitörler

R_PWM ve L_PWM olarak isimlendirilen bu iki pin, motorun sağ ve sol yöne doğru dönmesini sağlamaktadır. Bu pinler arduino nanonun PWM çıkışı olan iki pine bağlanmaktadır. R_EN, L_EN ve VCC olarak isimlendirilen pinler sırasıyla sağ half-bridge (BTN7971B), sol half-bridge (BTN7971B) ve güç katının

kullandığı 74HC244D çipinin beslemesidir. Bunlar TTL çalışmakta olup +5V ile beslenmektedir. Bu pinlerin beslemeleri DC-DC dönüştürücüden alınmaktadır. R_IS ve L_IS ise akım ölçümünde kullanılan pinlerdir. Bu pinlerin çıkışı, akımı filtreleme amaçlı PCB üzerine yerleştirilen 1 uF 'lık 1206 paketlik SMD kapasitörlere bağlanmaktadır. Ayrıca bu iki pinden daha iyi ölçüm yapabilmek için IBT-2 kartı üzerindeki 0805 paketlik 103 SMD RX ve RX dirençleri, 102 SMD ve 0805 paket dirençleri ile değiştirildi. Bu şartlar altında arduinonun L_IS ve R_IS pinlerinden gerilim başına 8.5 A akım ölçümü yapılmaktadır.

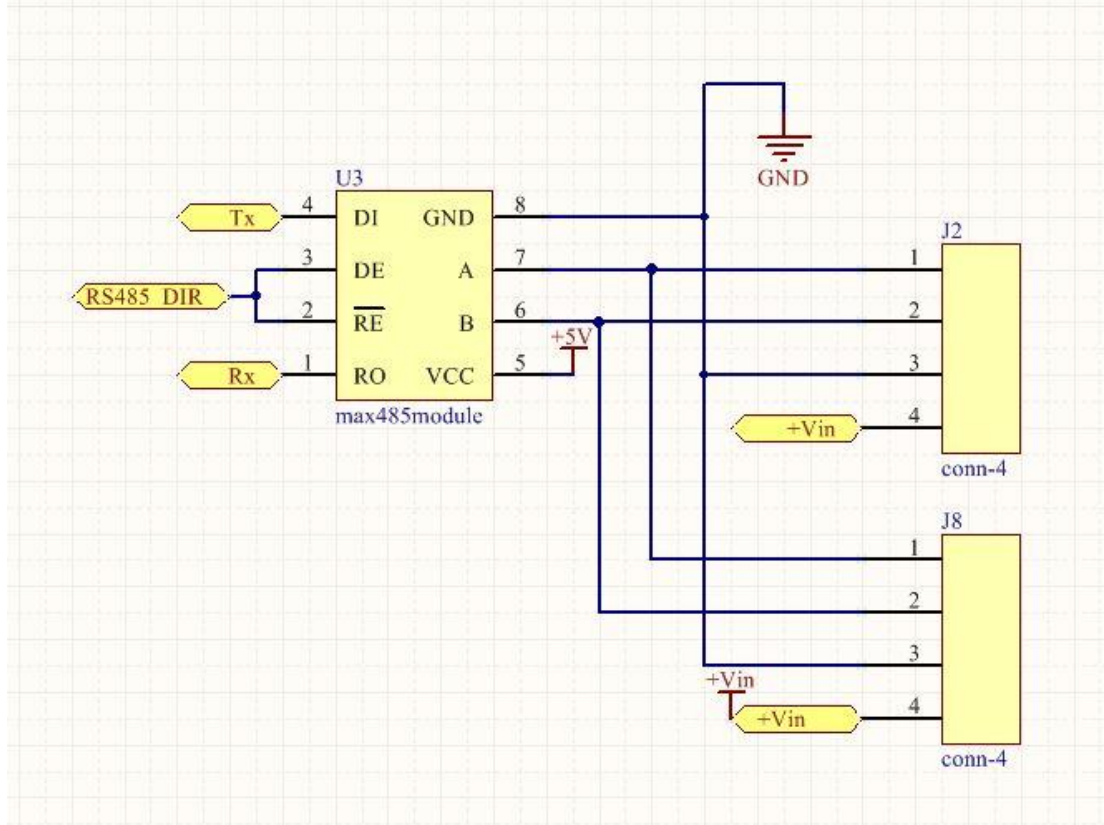
Şekil 3.9 ise arduino kartının bağlantı şeması görülmektedir. Motor sürücü devresinin tamamı, karmaşıklığı engellemek amacıyla etiketleme yöntemi ile çizildiğinden pin isimlerinden arduinonun pinlerinin nereye gidileceği anlaşılmaktadır. Zaten diğer komponentlerin pinleri anlatılırken arduinonun hangi pinine bağlanılacağı da anlatılmıştır.



Şekil 3.9. Arduino Nano Şematik Görünümü

Birden fazla motor sürücüsüne ihtiyaç duyulan sistemler için eklenen RS-485 haberleşme protokolü sayesinde 32 adet slave modda çalışabilecek motor sürücüsü kullanılabilir. Bu motor sürücüleri 1200 metrelik bir mesafeki masterden komutları alıp onları çalıştırabilir. Bu projedeki haberleşme ise Maxim firmasının üretmiş olduğu MAX485 shieldi ile gerçekleştirilmiştir.

Shield üzerinde bulunan DI pini arduino kartının üzerinde bulunan TX pinine, RO pini ise arduinodaki RX pinine bağlanması gerekir. Fakat bu projede SoftwareSerial kullanıldığından sanal olarak oluşturulan D12 ve D13 pinleri TX ve RX olarak kullanılmıştır. SoftwareSerial ile haberleşmeye neden ihtiyaç duyulduğu, çalışma stili ve nasıl programlandığı ise ilerleyen kısımlarda anlatılacaktır.

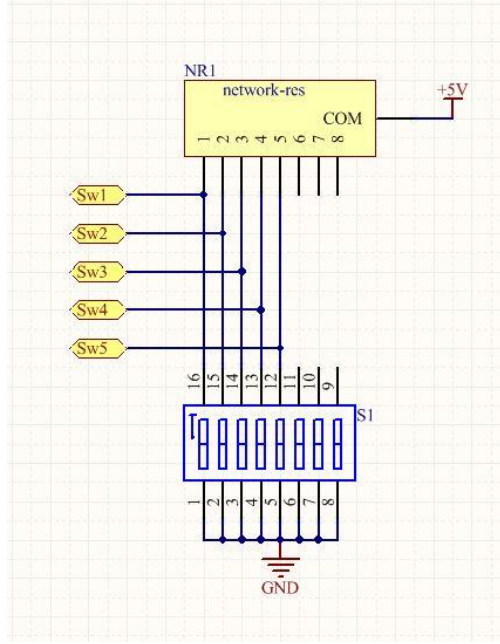


Şekil 3.10. MAX485 Şematik Görünümü

Shield üzerindeki DE ve RE pinleri ise kısa devre yapılarak, arduinonun D11 numaralı pinine bağlanmıştır. DE ve RE pinleri, şayet işlemciden bilgi gönderilecekse HIGH yapılır ve TX pini üzerinden veri transferi yapılır. Bilgi transfer işlemi yapıldıktan hemen sonra bu pin LOW değerine çekilir ve sistem bilgi alma moduna geçer. Herhangi bir bilgi geldiğinde ise onu otomatik alma modunda olduğundan bu veri alacaktır.

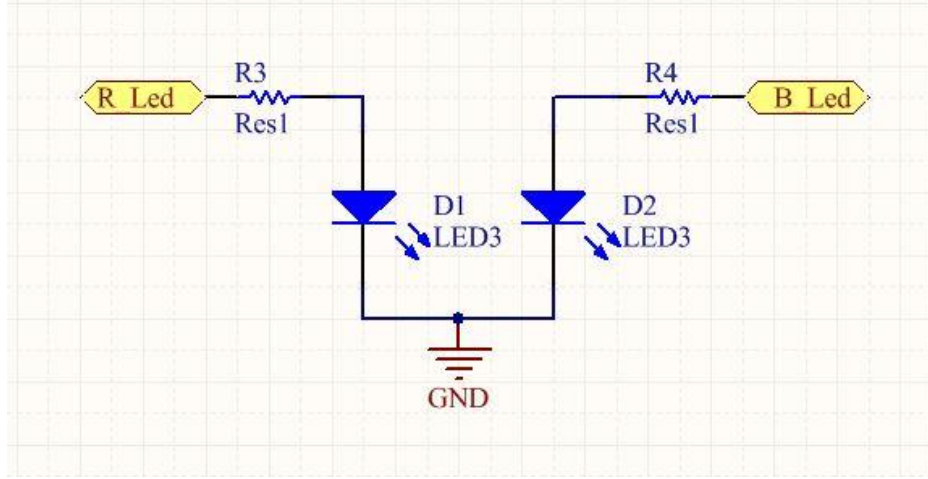
VCC pini ise DC-DC dönüştürücünün çıkışına bağlanıp +5V ile beslenmektedir. Geriye kalan A ve B pinleri ise bu kartın çıkış pinleridir. Yani haberleşme esnasında verilerin gönderilip alındığı pinlerdir.

MAX485 modülünün çıkışında olan GND, A ve B pinleri birer klemense gönderilerek ve yanına da bir +24V ilave edilerek diğer sürücüler ile arasındaki iletişimin kurulmasını sağlamaktadır. Klemense gönderilmesinin nedeni ise kablolama esnasında her bir işlemin diğer bütün işlemlerden bağımsız olarak yapılmasını daha sonradan eklenecek kabloların bu kablolamaları etkilememesini sağlamaktır.



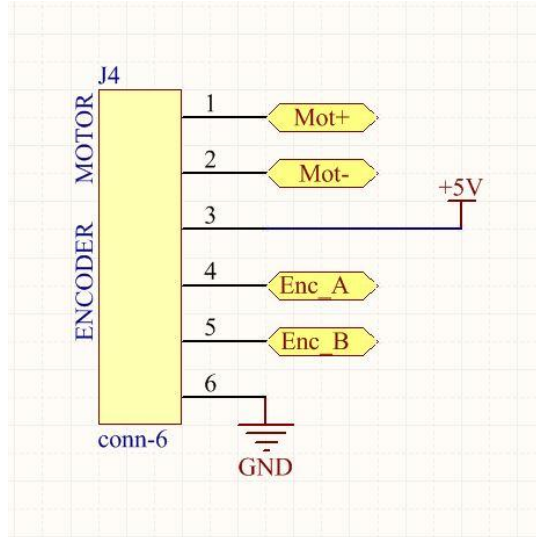
Şekil 3.11. Network Direnç ve Dip-Switch Şematik Görünümü

Master konumunda bulunan işlemcinin slave modunda olan sürücülerle sağlıklı bir iletişim kurulabilmesi için her bir sürücüye bir ID tanımlaması yapıldı. Bu ID tanımlaması her bir sürücüye bağlanan dip-switchler ile kullanıcı tarafından belirlenmektedir. Bu dip-switchler bir network direncine pull-up bağlanarak arduino nano kartının D4, D7, D8, A5 ve A4 pinlerine girilmiştir. Bu ID tanımlaması sürücüye ilk enerji verildiğinde switchler okunup programa girişi yapılmaktadır. Bu sayede master, slaveler arasında sağlıklı bir iletişim kurup bir karmaşıklık yaşamayacaktır.



Şekil 3.12. Ledler

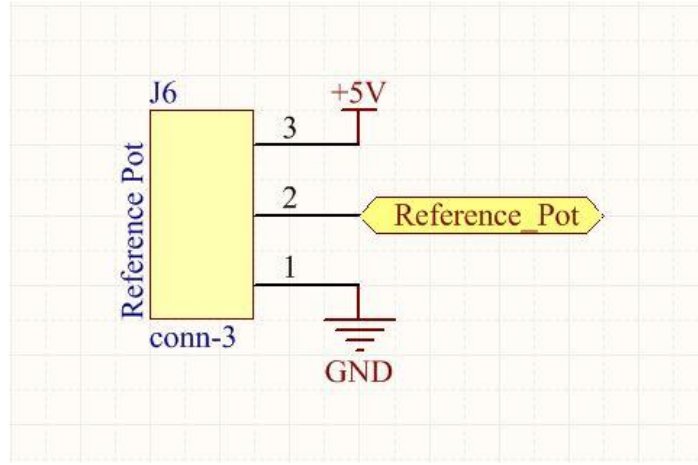
Sürücü üzerine yerleştirilmiş iki adet 0805 paketinde biri kırmızı diğeri mavi SMD led yerleştirilmiştir. Bu ledlerin arduino üzerindeki D9 ve D10 pinlerine girişi yapılmıştır. Bu ledler, sürücü ile kullanıcı arasında bir takım bilgileri aktarmak amaçlı kullanılmıştır. Örneğin sürücü ilk açılışta hazır olduğuna dair bir blink mesajı vermektedir. Bunun dışında mod değişimi, referans girişi veya bir hata mesajı bu ledler sayesinde kullanıcıya aktarılabilir.



Şekil 3.13. Enkoder ve Motor Klemensleri Şematik Görünümü

Sürücü üzerinde motor girişinin yapıldığı iki adet klemens bulunmaktadır. Motorun her iki ucu bu klemenslere bağlanmaktadır. Motor girişinin hemen

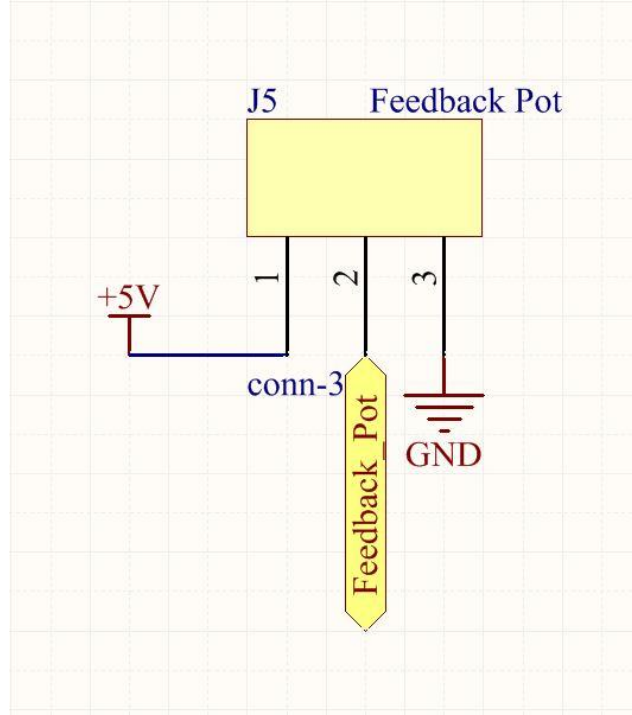
yanında sürücüye geri besleme elemanı olan enkoder giriş klemensleri bulunmaktadır. Bu klemenslerin birinin girişi DC-DC dönüştürücünün çıkışına bağlanmış olup +5V verip enkoderi beslemektedir. Klemensin biri de enkoderin GND bağlantısıdır. Geriye kalan iki klemens ise enkoderin A ve B kanallarıdır. Enkoder kanallarının ve motorun kablolarının yönü herhangi bir sıkıntı oluşturmayıp, ters bağlantı yapıldığı takdirde yalnızca motorun dönüş yönünde bir değişim göstermektedir. Fakat bu değişimde sürücünün çalışması için bir sıkıntı oluşturmamaktadır. Çünkü bunun önlemi yazılımsal olarak alınmıştır. Kablolar ters bağlandığı zaman bile yazılımsal olarak sürücü bunu değiştirmektedir.



Şekil 3.14. Referans Potansiyometresi Şematik Görünümü

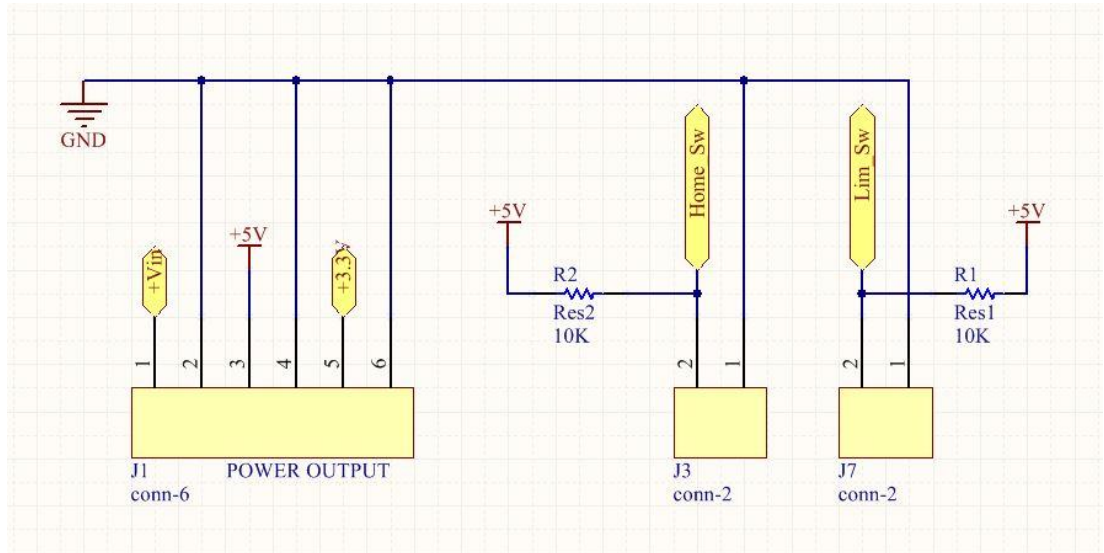
Sürücü üzerinde bir adet referans potansiyometresi için giriş bulunmaktadır. Toplam 3 klemense girişi bulunan bu potansiyometrenin beslemesi için bir adet +5V, bir adet GND ve arduinonun A2 girişine bağlanan bir veri klemensi bulunmaktadır.

Motordan yapılan geri beslemenin enkoder ile değilde potansiyometre ile yapıldığı durumlar için sürücü üzerine bir adet geribesleme potansiyometre girişi bulunmaktadır.



Şekil 3.15. Geri Besleme Potansiyometresi Şematik Görünümü

Bu potansiyometrenin de toplam 3 pini bulunmaktadır. Bunlardan biri DC-DC dönüştürücünün çıkışına bağlanan +5V, GND ve arduionun A3 pinine bağlanan veri hattıdır.



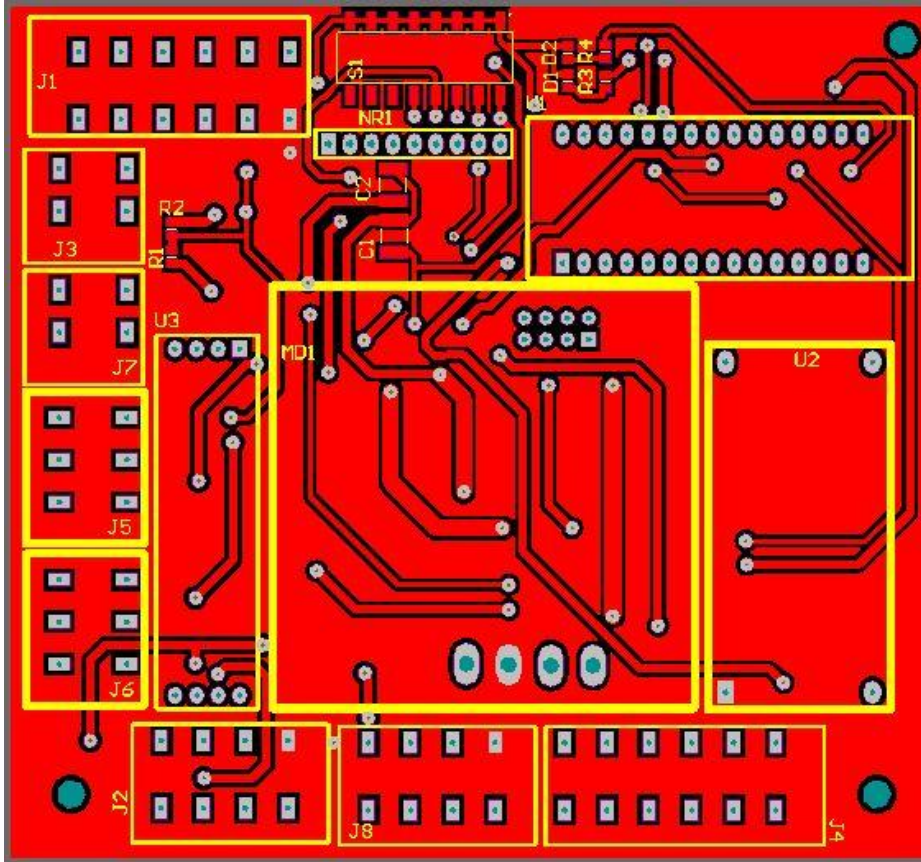
Şekil 3.16. Güç, Home Switch ve Limit Switch Şematik Görünümü

Sürücü üzerinde bir adet home switch ve limit switch bulunmaktadır. Bunların kullanılacağı yerde ayrıca mikrodenetleyiciye giriş olarak verilmesine gerek kalmadan direkt sürücü üzerinden okunabilmektedir. 2 adet 10K SMD 0805 paketlik dirençlerle pull-up bağlanan bu girişler arduinonun A6 ve A7 numaralı pinlerine bağlanmaktadır.

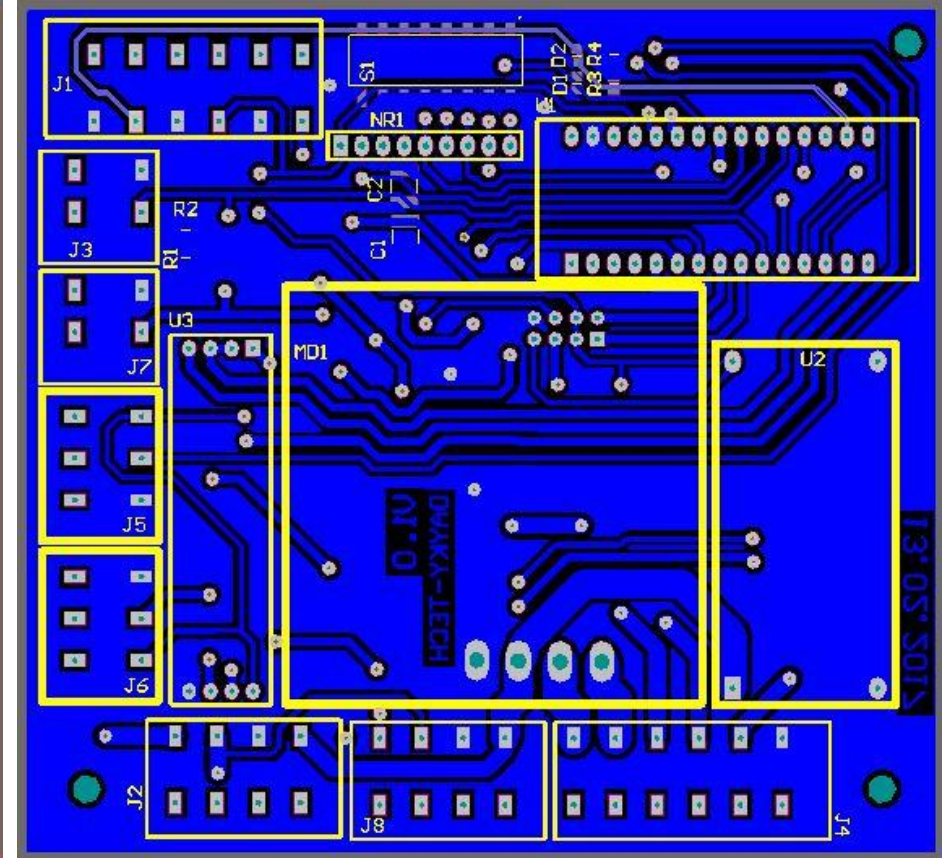
Ayrıca sürücü üzerinde keyfi olarak bırakılmış bir adet +12V, bir adet +5V ve bir adet +3.3V klemens ve bunların yanında GND klemensleride mevcuttur. Bir güç çıkışına ihtiyaç duyulan yerlerde (örneğin sürücü ile beraber kullanılacak bir sensör olabilir) bu klemensler kullanılabilir.

Devre şeması anlatıldığı gibi yapıldıktan sonra, devrenin baskı devre için 3D tasarıma geçilmesi gerekir. Herhangi bir devre tasarım programında şematik çizimi yapılan devre (bu projede Altium Designer kullanılmıştır), PCB tasarımı da tamamlandıktan sonra istenilen baskı devre yöntemi ile basılabilir. Sürücü kartı çift taraflı PCB 'den imal edilmiş olduğundan layerler arası geçişler vialarla gerçekleştirilmiştir. Via iletkenliği ise kısa kısa kesilen antigron kabloların çift taraflı lehimlenmesi sonucu sağlanmıştır. Bununla beraber baskı devre yapılmadan önce PCB 'nin alt yüzeyi tamamen GND ile, üst yüzeyi ise +24V ile kaplanmıştır. Bu baskı yöntemi sayesinde kabloların yüzey alanı artırıldığından, motorun çekeceği yüksek akıma karşı yanma riski ortadan kaldırılmış olur.

Motor sürücünün genel devre şeması şekil 17 'de, PCB baskı şeması ise şekil 3.18 ve şekil 3.19 'da verilmiştir.



Şekil 3.18. Motor Sürücü Üst Yüzey PCB Görünümü



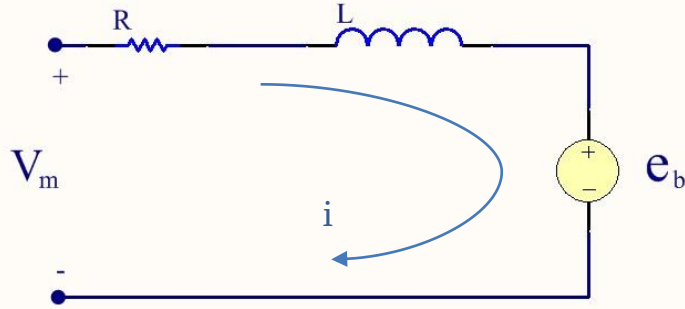
Şekil 3.19. Motor Sürücü Alt Yüzey PCB Görünümü

3.4. DC MOTORUN SİSTEM DİNAMİĞİ

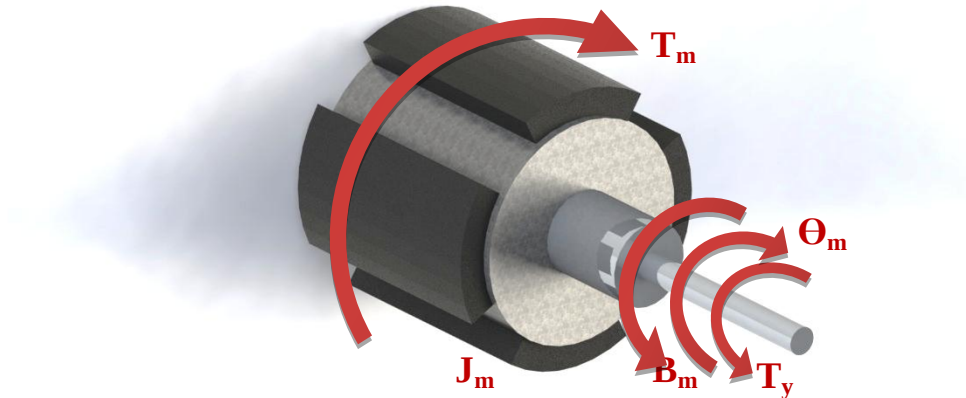
3.4.1. DC Motorun Transfer Fonksiyonu

Transfer fonksiyonu, dinamik bir sistemin girişi ile çıkışı arasındaki ilişkiyi gösteren bağıntıdır. Gerçek sistemlerin kontrol uygulamalarını analitik inceleyebilmek için matematiksel modelini çıkarmaya ihtiyaç duyulur. Bu matematiksel model ise genellikle transfer fonksiyonu üzerinden gerçekleştirilir.

Bir DC motoru direnci ve sarımlarından dolayı hem elektriksel olarak, üzerindeki yük ve ataletinden dolayı ise hem de mekanik olarak modellenmek zorundadır.



Şekil 3.20. DC Motor Elektriksel Devre Şeması



Şekil 3.21. DC Motor Mekanik Çizimi

$$V_m = i \cdot R + L \cdot \frac{di}{dt} + K \cdot \omega_m$$

$$T_m = j_m \cdot \ddot{\theta}_m + B_m \cdot \dot{\theta}_m + T_y$$

Yukarıda elde edilen denklemlerin Laplace Dönüşümü yapılarak;

$$V_m = I(s) \cdot (R + L \cdot s) + K \cdot s \cdot \theta_m(s)$$

$$T_m(s) = K \cdot I(s)$$

$$T_m(s) = j_m \cdot s^2 \cdot \theta_m(s) + B_m \cdot s \cdot \theta_m(s) + T_y$$

denklemleri elde edilir.

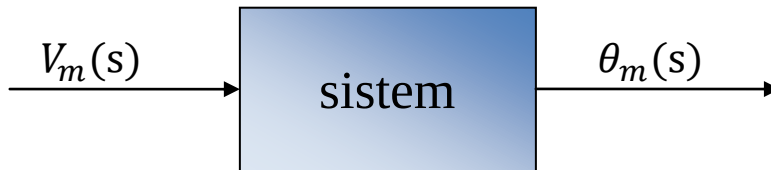
$$K \cdot I(s) = (j_m \cdot s^2 + B_m \cdot s) \cdot \theta_m(s) + T_y$$

Bu denklemde, motor yükünün olmadığı varsayılarak $T_y = 0$ alınırsa ;

$$K \cdot I(s) = (j_m \cdot s^2 + B_m \cdot s) \cdot \theta_m(s)$$

olur.

$$V_m = I(s) \cdot (R + L \cdot s) + K \cdot s \cdot \theta_m(s)$$



Sistemin girişine uygulanan PWM (V_m)sonucunda, sistemin çıkışında (θ_m) açısal konum alınır.

$$\frac{j_m \cdot s^2 + B_m \cdot s}{K} \cdot (L \cdot s + R) \cdot \theta_m(s) + K \cdot s \cdot \theta_m(s) = V_m$$

Transfer fonksiyonu;

$$TF = \frac{\theta_m(s)}{V_m(s)}$$

olduğundan, sistemin transfer fonksiyonu,

$$\frac{\theta_m(s)}{V_m(s)} = \frac{K}{(j_m \cdot s^2 + B_m \cdot s) \cdot (L \cdot s + R) + K^2 \cdot s}$$

olur. Buradan,

$$\frac{\theta_m(s)}{V_m(s)} = \frac{K}{s[(j_m \cdot s + B_m) \cdot (L \cdot s + R) + K^2]}$$

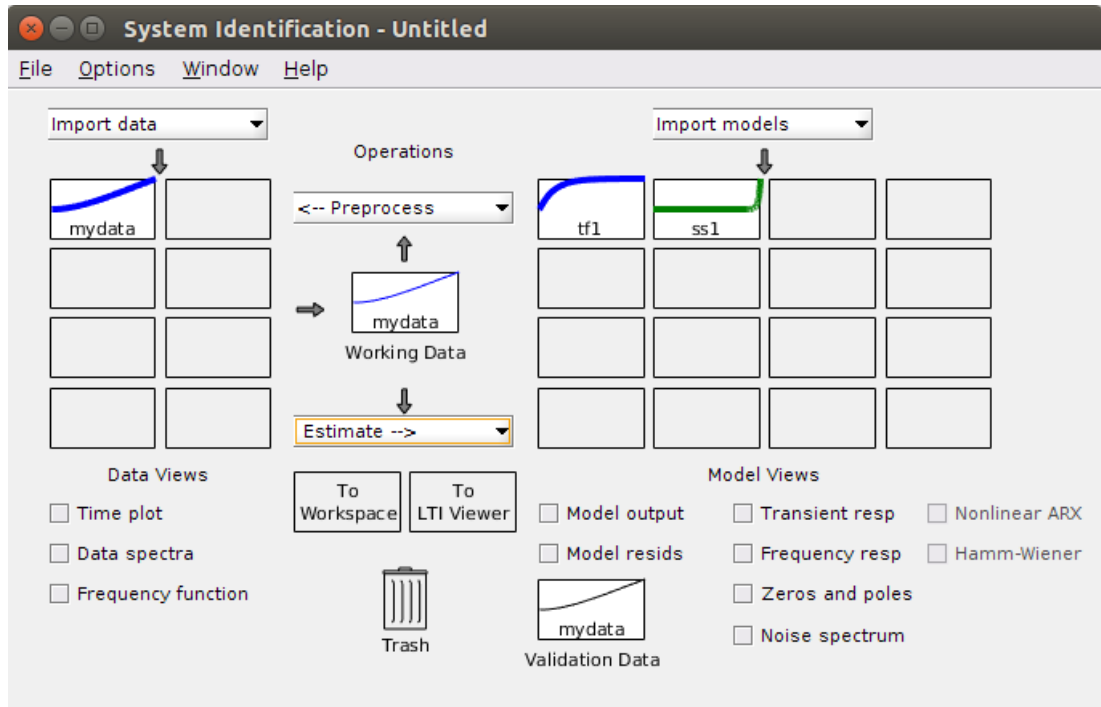
denklemini elde edilir.

3.4.2. MATLAB ile DC Motorun Transfer Fonksiyonunu Çıkarma

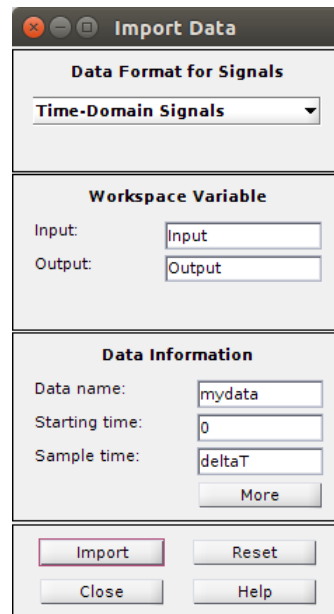
Motor sürücü bilgisayara bağlanıp MATLAB üzerinden bir COM açıldı. Test motorunun karakteristiğini belirleyebilmek için kısa süreliğine tam performans çalıştırılarak 200 adet veri alındı. MATLAB ile bu veriler bir diziye atanarak kaydedildi. Sonra MATLAB programının Command Window bölümüne,

```
ident
```

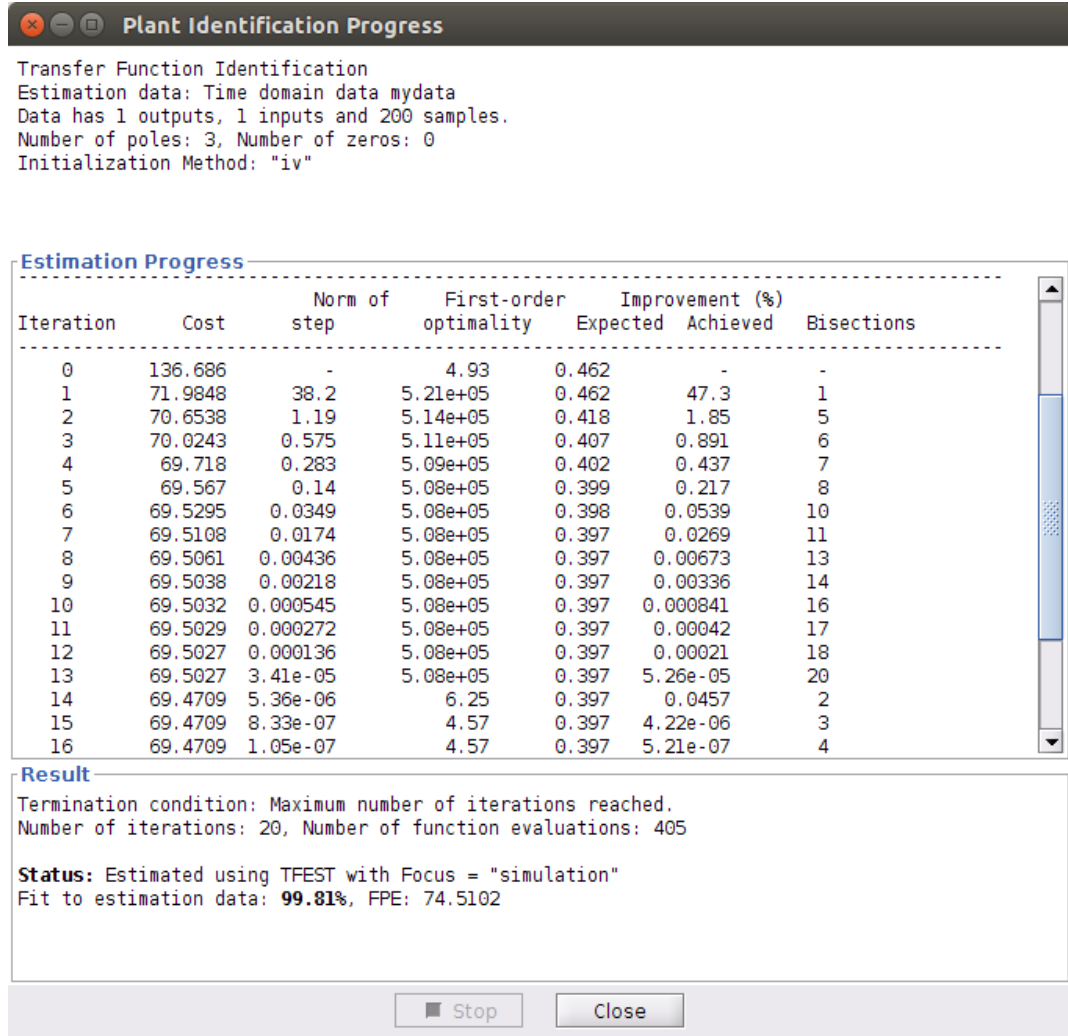
komutu girilerek System Identification Toolbox penceresi açılıp veriler Import Data penceresinden buraya aktarılır.



Şekil 3.22. System Identification Penceresi



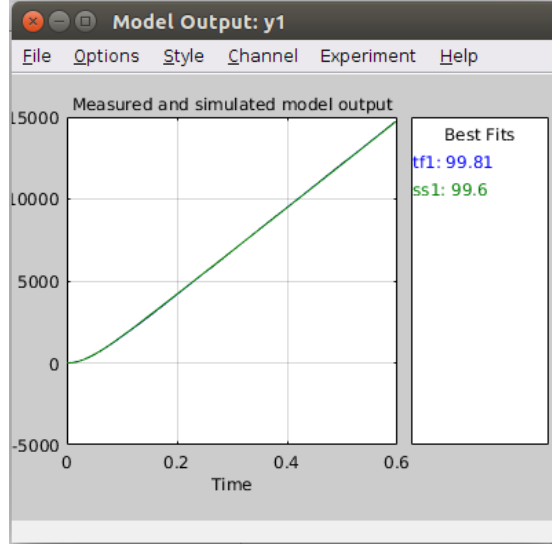
Şekil 3.23. Import Data Penceresi



Şekil 3.24. Plant Identification Penceresi

Bundan sonraki adımda ise sistemin kutup ve sıfır sayısı girilir. MATLAB bu bilgilere göre Plant Identification Progress penceresinden bir transfer fonksiyonu çözümlemesi yaparak sonucu göstermektedir. Bu toolbox ile yalnızca transfer fonksiyonu değil durum uzayı modeli vs. gibi modeller de üretilebilmektedir.

Model Output seçeneğinden ise üretilen modelin gerçek sistemle ne kadar uyduğu yüzdesi görülebilir.



Şekil 3.25. Model Output Penceresi

Şekil 3.25 'den de görüldüğü üzere MATLAB' in ürettiği transfer fonksiyonu modeli gerçek sistemle %99.81 oranında, durum uzayı modeli ise %99.6 oranında uyuşmaktadır.

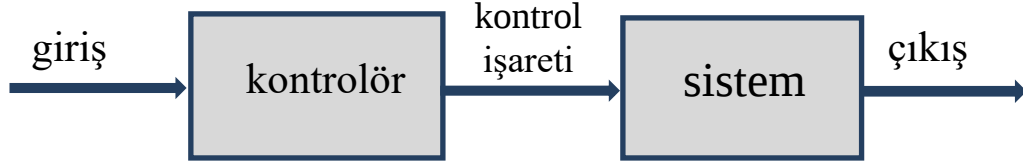
3.5. PID Kontrol Algoritması

Otomatik kontrol teorisinde birçok kontrol algoritması bulunmasına rağmen PID kontrol algoritması kullanım kolaylığı sebebiyle endüstride sıkça kullanılmaktadır. Diğer kontrol algoritmalarında olduğu gibi PID kontrol algoritmasının temelinde de geri beslemeli kontrol sistemleri yatmaktadır. Bilindiği üzere kontrol sistemleri çalışma prensipleri bakımından ikiye ayrılmaktadır.

- Açık Çevrim Sistemler
- Kapalı Çevrim Sistemler

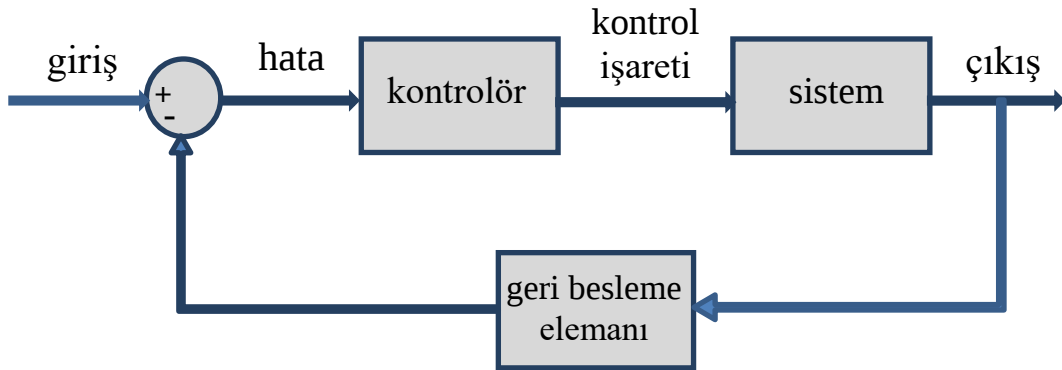
Açık çevrim kontrol sistemleri, kendisine referans olarak girilen kontrol işareti sayesinde sistemin çıkışının kontrol edildiği fakat bu çıkışın sistemin girişi ile bir bağlantısının bulunmadığı sistemlerdir. Biraz daha açıklamak gerekirse, sistemin çıkışının istenilen duruma gelebilmesi için sisteme bir kontrol işareti girilir. Bu işaret girildikten sonra sistemin çıkışının istenilen değere gelip gelmediği kontrol edilmez. Bu nedenle sistem çıkışının bozucu dış etkiler, gürültü vs. gibi fiziksel

engeller sebebiyle istenilen değere oturmama olasılığı oldukça yüksektir.



Şekil 3.26. Açık Çevrim Sistemin Blog Diyagramı

Kapalı çevrim kontrol sistemleri ise, açık çevrim sistemlere ek olarak sistem çıkışının bir geri besleme elemanı ile sistem girişine tekrar verilip referans değeri ile farkı alındıktan sonra elde edilen hata bir kontrolörde değerlendirilip o hatanın minimuma indirilerek sistem çıkışının sürekli olarak kontrol edildiği sistemlerdir.

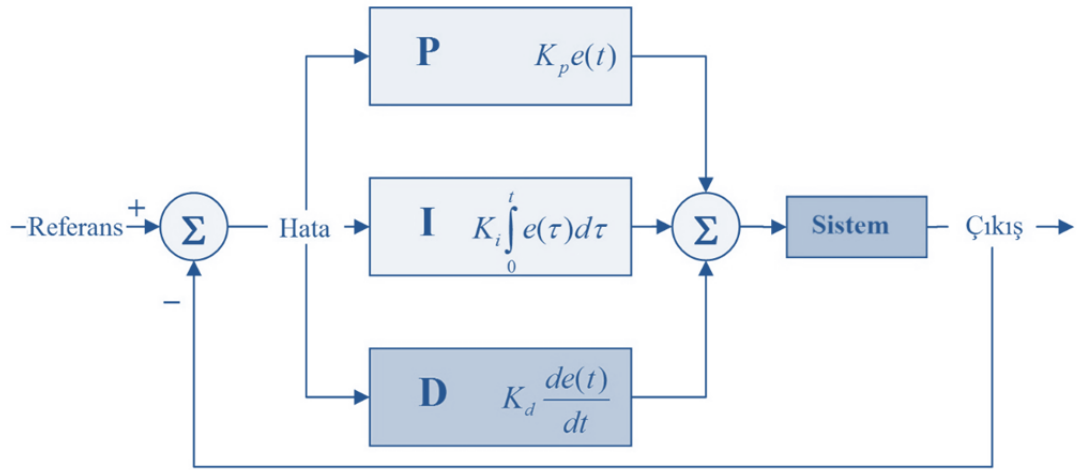


Şekil 3.27. Kapalı Çevrim Sistemin Blog Diyagramı

Kapalı çevrim sistemler, açık çevrim sistemlere göre oldukça hassastırlar. Sistem kontrolcüsü doğru bir şekilde dizayn edildiği takdirde oldukça kararlı bir yapıya sahiptirler. Fakat sistem ileri teknoloji ile donatıldığından açık çevrim sistemlere göre daha erken bozulmaları sebebiyle düzenli olarak mekanik ve elektronik bakımları yapılmalıdır. Ayrıca kapalı çevrim sistemler, açık çevrim sistemlere göre maliyetlidir.

Kapalı çevrim kontrol sistemlerinde, sistemi kontrol eden bir kontrolörden bahsedilmiştir. Bu kontrolörün tasarımında çeşitli yöntemlerin yanı sıra, endüstride de sıkça karşılaşılan PID kontrol algoritması kullanılmaktadır.

PID aslında bir kontrol geri bildirim mekanizmasıdır. Bir PID kontrolcü, ölçülü bir süreç içinde değişen ve istenilen referans noktası ile bulunduğu değer arasındaki farkı alarak bir hata sinyali hesaplar. Oluşan bu hataya göre PID kontrolcüsü, hatayı en aza indirmeye yönelik bir etki yapar ve çıkışa gönderir. Bu şekilde hata minimuma indirilene kadar çıkıştan girişe sürekli geri besleme ile hatalar belirlenir ve kontrolcü etkisini çıkışa göndererek hatayı azaltır.



Şekil 3.28. PID Kontrol Algoritması Blok Diyagramı

PID kontrol algoritması 3 farklı parametreyi içerir. Bu parametrelerden P parametresi oransal, I parametresi integral ve D parametresi de türev olarak adlandırılmaktadır.

Şekil 3.28 'den de görüldüğü üzere PID kontrolörü dizayn edilirken sistemde oluşan hata değeri bir katsayı ile çarpılıp, aynı hatanın integral ve türevi de alınarak farklı katsayılar ile çarpılması sonucu toplanıp sisteme giriş olarak verilmektedir. Öyleyse sistemde oluşan hataya $e(t)$, referans girişine $r(t)$, sistem girişine $u(t)$ ve sistem çıkışına da $y(t)$ denilecek olursa,

$$e(t) = r(t) - y(t)$$

$$u(t) = K_P \cdot e(t) + K_I \cdot \int e(t) \cdot dt + K_D \cdot \frac{de(t)}{dt}$$

Bilindiği üzere bir fonksiyonun integralinin Laplace dönüşüm fonksiyonu $\left(\frac{1}{s}\right)$ ifadesi ile çarpılacak şekilde, fonksiyonun türevinin Laplace dönüşüm fonksiyonu ise (s) ifadesi ile çarpılarak yazılabilir. Öyleyse PID kontrolörünün transfer fonksiyonu,

$$U(s) = \left(K_P + \frac{K_I}{s} + K_D \cdot s \right) \cdot E(s)$$

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D \cdot s$$

olur.

PID kontrol algoritmasında öncelikle hata tanımlaması yapılmalıdır.

1	Referans = İstenilen değer
2	Gelen = Şu an ki konum
3	Hata = Referans - Gelen

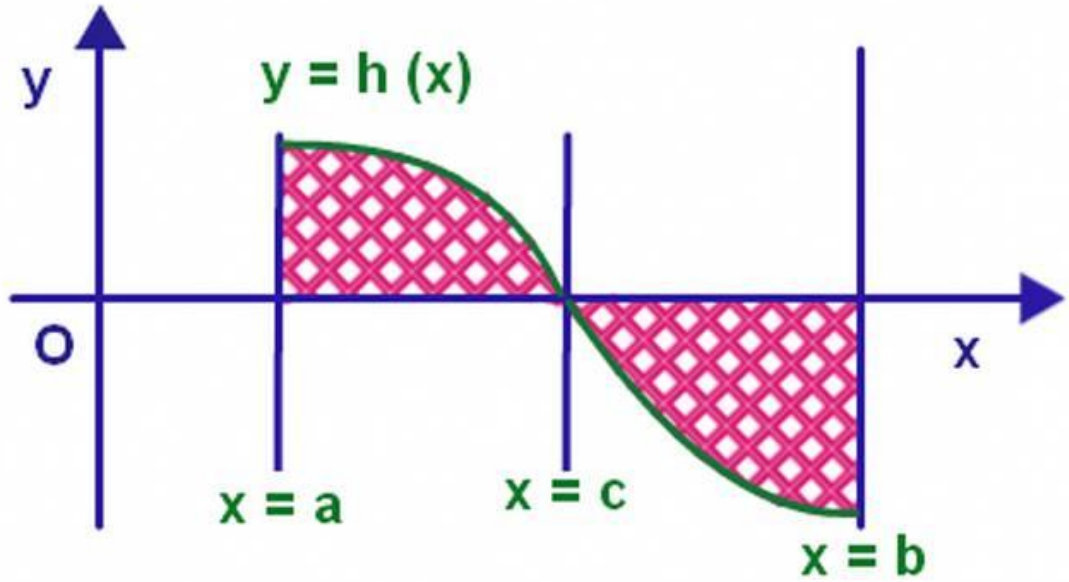
3.5.1. Oransal (Proportional) Terim

Oransal terim, sistemde var olan hatayı bir katsayı ile çarparak bu hatayı minimize etmeyi hedefler. Fakat hata istenilen değere yaklaştıkça sistemin çıkışında osilasyon görülme ihtimalinin artması bu terimin dezavantajlarından. Bu yüzden çok yüksek seçilmemesi gerekir.

1	$P = K_P \cdot \text{Hata}$
---	-----------------------------

3.5.2. İntegral (Integral) Terimi

İntegral grafikteki eğrinin altında kalan alanı bulmak için kullanılır. PID kontrol algoritmasında sistemin referans değerine giderken yaptığı hata değerini bulmak için kullanılmaktadır (Şekil 3.29). Herbir Δt (PID fonksiyonuna her girdiğinde geçen zaman) çevrimindeki hata bir K_i katsayısıyla çarpılarak toplanır. İntegralin çok yükselmesini engellemek için sınırlandırmak gerekir. Sürekli toplandığı için integral çok artarsa tekrar azalmasını beklemek sistemin yavaşlamasına hatta bazen kontrolörün bufferının şişmesine neden olacağından sistemdeki bağlantıyı bitirecektir. Bu yüzden integrali sınırlamak sistemin toparlanmasını sağlayacaktır.



Şekil 3.29. İntegrali Alınan Bir Grafik

1	$I = I + (K_i * Hata * dt)$
---	-----------------------------

3.5.3. Türev (Derivative) Terimi

Türev terimi, sistemdeki iki örnek arasındaki zamanı hesaplar. Eğer hatada bir değişim olmadıysa türev sıfır olur.

```
1 Hata_1 = eski_hata (bir önceki hatanın değeri)
2 HD = Hata - Hata_1
3 D = (Kd * HD) / dt
```

3.5.4. PID Kontrol Algoritması

Daha öncede anlatıldığı gibi PID kontrol algoritması K_p , K_i ve K_d gibi üç farklı parametreden oluşmaktadır. Bu katsayılar deneme yanılma yoluyla bulunabileceği gibi birtakım metodlarla da bulunabilmektedir. Bu projede Ziegler-Nichols yöntemi kullanılarak bu katsayılar belirlenmiştir. Bu yöntemin nasıl kullanıldığı ilerde anlatılacaktır.

```
1 Hata = Referans - Gelen
2 HD = Hata - Hata_1
3 P = Kp * Hata
4 I = I + (Ki * Hata * dt)
5 D = (Kd * HD) / dt
6 PID = P + I + D
7 Hata_1 = Hata
```

3.5.5. PID Parametrelerinin Belirlenmesi

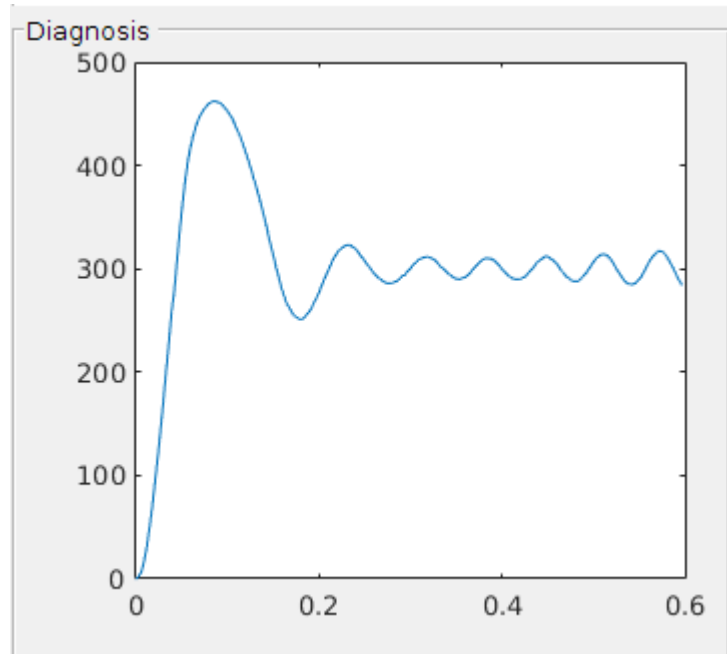
PID kontrol algoritmasının K_p , K_i ve K_d parametrelerini belirleyebilmenin birden fazla yöntemi bulunmaktadır. Bu parametreler sabit olmayıp, motor sürücünün entegre edildiği sisteme göre değişkenlik göstermektedir. Bu motor sürücüyü sonradan kullanmak isteyenlere yardımcı olabilmek için bu proje kapsamında, sürücünün bu parametrelerinin farklı iki yöntemle nasıl belirleneceği anlatılacaktır.

İlk metot, Ziegler-Nichols Metodu diğer metot ise deneme yanılma yoluyla parametreleri belirlemedir. Ziegler-Nichols Metodu, sistem için belirlediği parametreler bazen o sistem için en iyi katsayılar olmayabilir. Bu katsayılar referans alınarak üzerinde biraz değişiklik yapılabilir. Bu değişikliğin nasıl yapılacağı ise deneme-yanılma yolunda anlatılacaktır. Zaten Ziegler-Nichols Metodu'nun belirlediği parametreler direkt sisteme uymazsa ondan sonra yapılan işlemler deneme-yanılma olmaktadır.

Ziegler-Nichols Metodu, açık çevrim Ziegler-Nichols Metodu ve kapalı çevrim Ziegler-Nichols Metodu olmak üzere iki şekilde yapılabilmektedir. Bu proje dahilinde açık çevrim kullanılmayıp, kapalı çevrim Ziegler-Nichols Metodu kullanılmış olup aşağıda da anlatılanlar kapalı çevrim Ziegler-Nichols Metodu için geçerlidir.

3.5.5.1. Ziegler-Nichols Metodu

Bu yöntem geri beslemeli sistemin, referans değeri olarak birim basamak girişi ile sürekli osilasyonda test edilmesine dayanır. Başlangıçta PID kontrol algoritmasında K_i ve K_d parametreleri sıfır yapılarak devre dışı bırakılır. Bu şekilde kontrolörün yalnızca K_p parametresi ile çalışması sağlanır. Sistemin girişine referans değeri olarak birim basamak uygulanır ve K_p parametresi sıfırdan başlatılıp devamlı artırılarak sistem çıkışının osilasyon yaptığı değer tespit edilir. Sistemin osilasyona girdiği K_p değerine K_u , osilasyon periyoduna ise P_u denilecek olursa tablo 3.1 'den PID parametreleri bulunur.



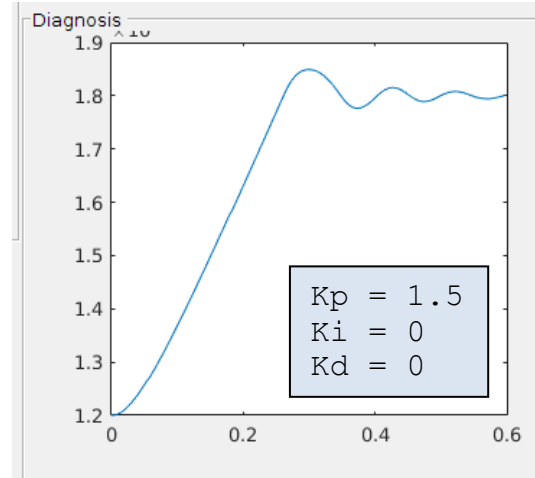
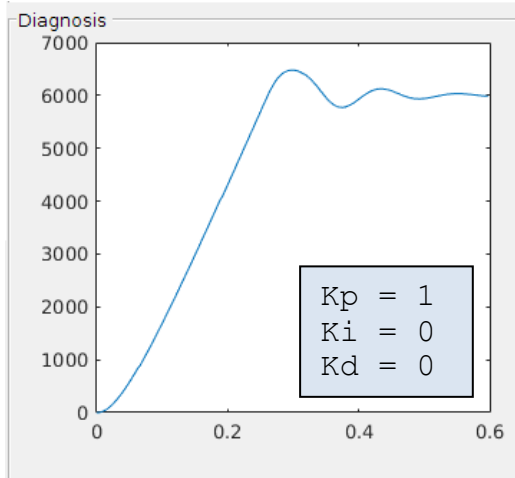
Şekil 3.30. Osilasyona Girmiş Sistem Grafiği

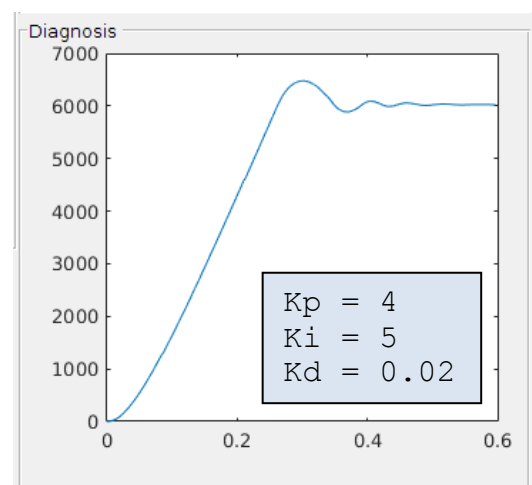
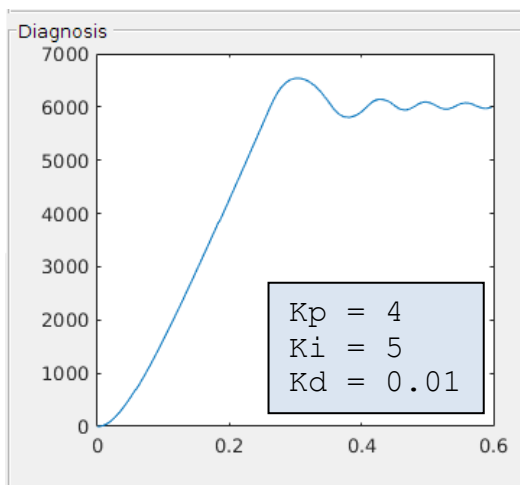
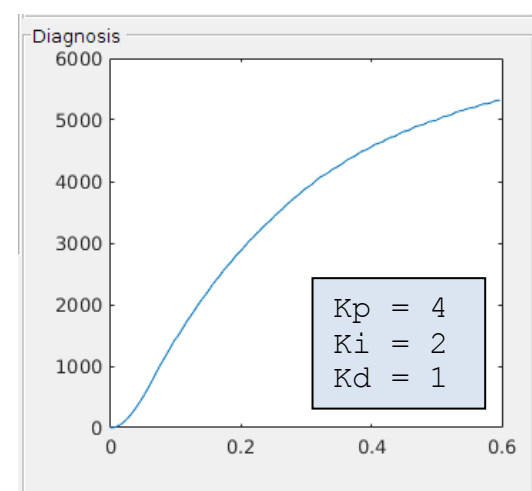
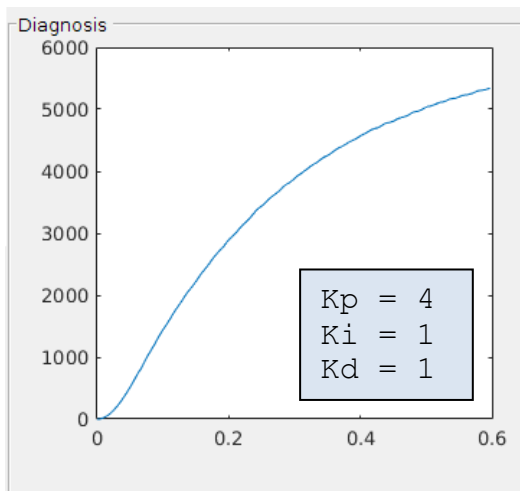
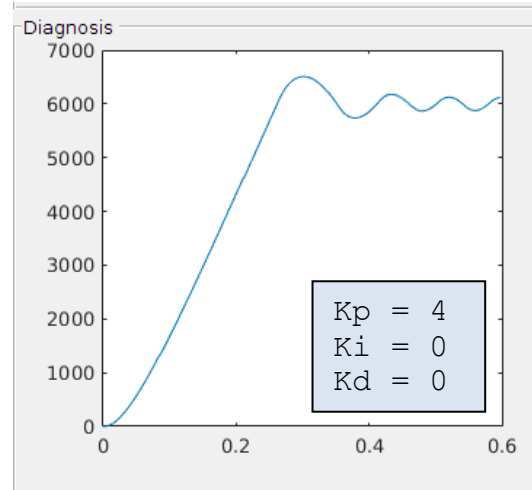
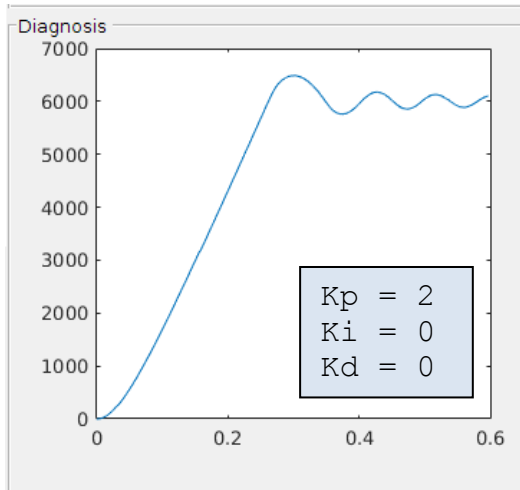
Kontrolör	Kp	Ki	Kd
P	$K_u / 2$	-	-
PI	$K_u / 2.2$	$P_u / 1.2$	-
PID	$K_u / 1.7$	$P_u / 2$	$P_u / 8$

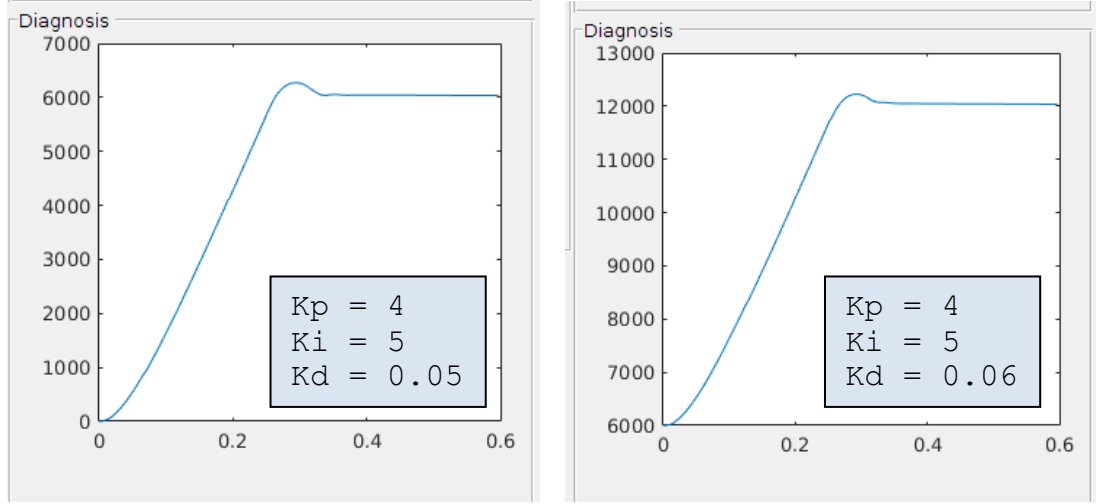
Tablo 3.1. Ziegler-Nichols Tablosu

3.5.5.2. Deneme-Yanılma Yoluyla Parametrelerin Bulunması

Deneme yanılma yoluyla PID parametrelerini bulmak biraz sıkıcı ve zaman alabilir. Fakat bu parametrelerin ne işe yaradığını bilmek bu işi daha da kolaylaştıracaktır. Yukarda bu parametrelerin ne işe yaradığı anlatılmıştır. İlk olarak Ki ve Kd parametreleri sıfır yapılarak Kp parametresi yavaş yavaş artırılır. Sonrasında sistem için uygun Kp değeri bulunup, bu parametre sabit tutularak Ki değeri artırılır. Sonrasında ise eğer ihtiyaç varsa Kd parametresi eklenir.







Şekil 3.31. Aynı Sistemin Değişik PID Parametrelerine Karşılık Verdiği Sistem Cevabı

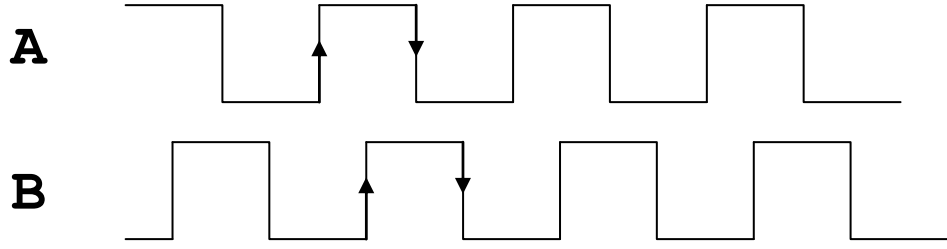
3.6. Sistemin Kontrolü

Bu proje kapsamındaki kodlar Arduino IDE ortamında, arduinoya özgü C programlama dili tabanlı Processing dili kullanılarak yazılmıştır.

Programlamadaki karmaşıklığı gidermek amacıyla sürekli olarak kullanılacak olan fonksiyonlar bir kütüphane oluşturularak basitleştirildi. Programın tamamında toplam 3 adet kütüphane oluşturulup bunlar kullanılmıştır.

```
1 #include <PinChangeInt.h>
2 #include "PID.h"
3 #include "DEFINITIONS.h"
```

İlk başta kullanılan PinChangeInt.h kütüphanesi, enkoderin okuyabilmek için oluşturulmuştur. Bu kütüphane sayesinde enkoderler 4x modda okunabilmektedir. Yani bu demek oluyor ki aralarında 90 derece faz farkı bulunan her iki kanalın hem yükselen kenarına hem de düşen kenarına bakılarak işlem yapıldığıdır. Şekil 3.32 incelendiğinde daha iyi anlaşılacaktır.



Şekil 3.32. İki Kanallı Enkoder Darbeleri

A Change	A	B	Pozisyon
	1	0	++
	0	1	++
	1	1	--
	0	0	--

B Change	A	B	Pozisyon
	1	1	++
	0	0	++
	0	1	--
	1	0	--

Tablo 3.2. Enkoder Yönü Tespiti

```

1 | attachPinChangeInterrupt(encPinA, isrForEncPinA, CHANGE);
2 | attachPinChangeInterrupt(encPinB, isrForEncPinB, CHANGE);

```

PinChangeInterrupt, her iki enkoder için de ayrı ayrı 4x modda okuyabilmek için InterruptServiceRoutine'leri CHANGE olarak ayarlandı.

İkinci kütüphane olan PID.h ise PID kontrol algoritmasının yazıldığı kütüphanedir. İçerisinde PID parametreleri hesaplanmaktadır. Bu parametrelerin nasıl hesaplandığı daha önce anlatılmıştı.

Arduinoya giriş yapılabilecek maksimum PWM değeri 255, minimum PWM değeri ise -255 olduğundan bunlar öncelikle sınırlandırıldı. Böyle bir önlemin alınması programın daha stabil çalışmasını sağlayacaktır.

```

1 float outMax = 255;
2 float outMin = 255;
3
4 if (uI > outMax)
5     uI = outMax;
6 else if (uI < outMin)
7     uI = outMin;

```

PID kontrol algoritmasının temel mantığı yukarda anlatılmıştı. Yukardaki teorik bilgiler dahilinde gerçek sistem için kontrol algoritması oluşturularak PID.h kütüphanesine eklenmiştir.

```

1 uP = kP * error;
2 uI += kI * error * deltaT / 1000.0;
3 uD = kD * (error - error_1) * 1000.0 / deltaT;
4
5 error_1 = error;
6 u = uP + uI + uD;

```

Geriye kalan ve son kütüphane olan DEFINITIONS.h kütüphanesi ise kontrolöre gönderilen komutların listesinin yazıldığı yerdir. Bu liste aşağıda yer alıp, isimlerden bu komutların hangi görevi icra ettikleri anlaşılmaktadır. İlk tabloda yer alan komutlar, sürücünün kullanılabileceği modları temsil etmektedir. Birden fazla modda çalışabilen bu sürücü önce modu aktif edilip daha sonra görev komutları girilmesi gerekmektedir.

0	MANUAL_MODE
1	POSITION_MODE
2	PROFILE_POSITION_MODE
3	VELOCITY_MODE
4	PROFILE_VELOCITY_MODE
5	CURRENT_MODE
6	HOMING_MODE

Tablo 3.3. Kontrolör Mod Değişim Tablosu

Çalışması istenen mod seçildikten sonra aşağıdaki komutlar kullanılabilir.

1	COM_REL_VEL_REF
2	COM_ABS_VEL_REF
3	COM_REL_POS_REF
4	COM_ABS_POS_REF
5	COM_SET_POS_GAINS
6	COM_SET_VEL_GAINS
7	COM_PRINT_LONG_VEL_REPORT
8	COM_PRINT_SHORT_VEL_REPORT
9	COM_PRINT_LONG_POS_REPORT
10	COM_PRINT_SHORT_POS_REPORT
11	COM_PRINT_NO_REPORT
12	COM_SEND_DATA_FOR_MATLAB
13	COM_SET_U
14	COM_PRINT_HELP
15	COM_GET_POSITION_LONG
16	COM_LOG_POSITION
17	COM_LOG_CURRENT
18	COM_TEST_MOTOR_CONNECTION
19	COM_GET_LOG_SIZE_INTEGER
20	COM_GET_LOOPRATE_UINT8
21	COM_CHANGE_USER_POS_DIR
22	COM_SET_LOOPRATE
23	COM_SET_PRINTING_PERIOD
24	COM_RESET_ENCODER_POS
25	COM_GET_POS_GAINS
26	COM_GET_VEL_GAINS
27	COM_REL_PROF_POS_REF
28	COM_ABS_PROF_POS_REF
29	COM_SET_DRIVER_STATE
30	COM_SET_OPERATING_MODE
31	COM_REL_MAX_PROF_POS_REF
32	COM_ABS_MAX_PROF_POS_REF

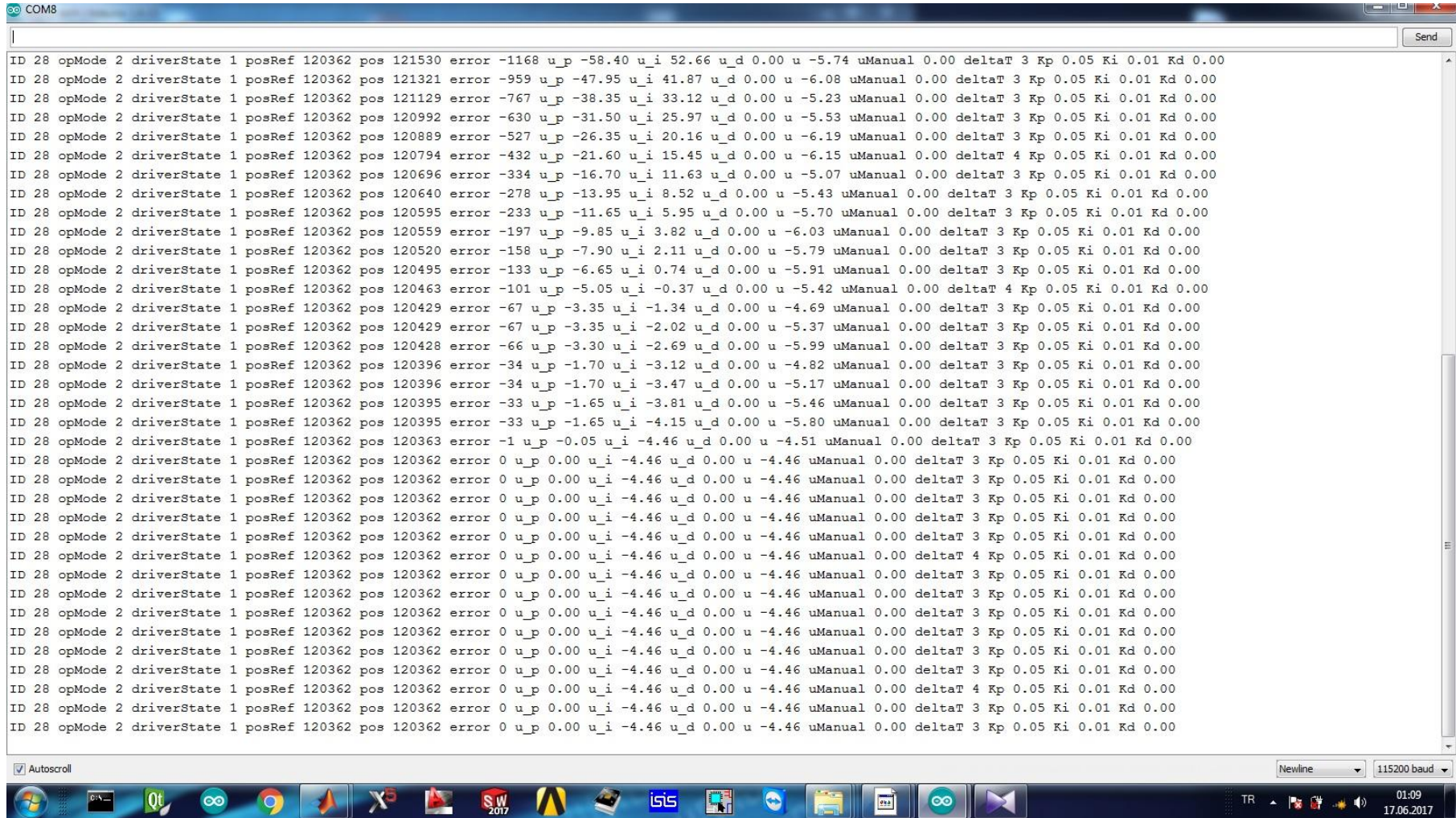
Tablo 3.4. Kontrolör Komut Listesi

Motor sürücünün üzerine entegre edildiği güç katı sayesinde akım ölçümü yapılabileceğinden bahsedilmişti. Bu ölçümün ise aşağıda verildiği şekilde yapılmaktadır.

```
1   rIs = analogRead(R_IS);
2   lIs = analogRead(L_IS);
3
4   rIsFloat = rIs / 1023.0 * 5.0 * 8.5;
5   lIsFloat = lIs / 1023.0 * 5.0 * 8.5;
6   totalCurrent = rIsFloat - lIsFloat;
```

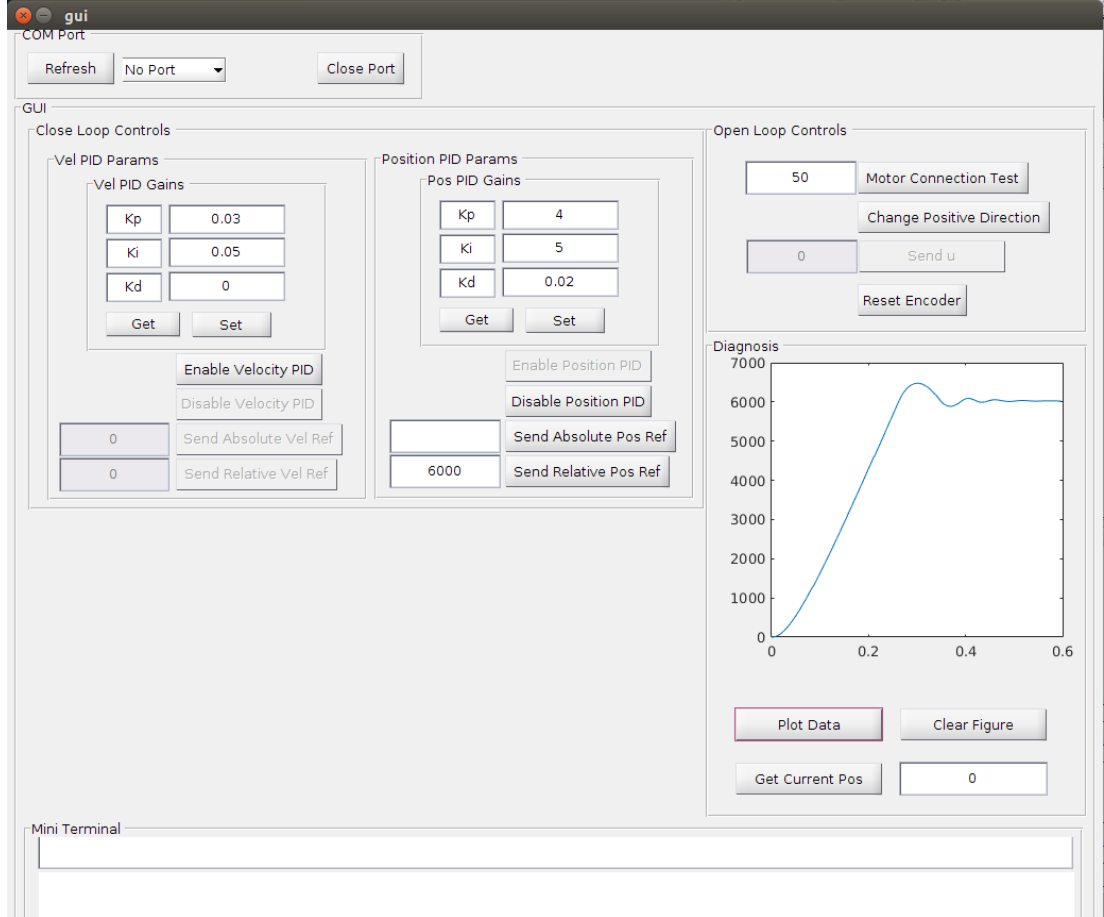
3.6.1. Kontrolöre CPU Üzerinden Referans Gönderme

Kontrolöre referans göndermek için birtakım kodlamalar yapıldı. Bu kodlar yukarda verilmiştir. Herhangi bir seri monitörden sürücüye bağlanıp referans gönderilebilir. Arduinonun seri monitörden alınmış bir görüntüsü aşağıda verilmiştir.



Şekil 3.33. Kontrolör Durumunu Gösteren Seri Monitör

Kontrolöre seri monitörden referans gönderildiği gibi MATLAB ile hazırlanmış bir arayüz sayesinde de referans gönderilebilmektedir. Arayüzün seri monitöre göre bir avantajı bulunmaktadır. Bu da şekil 3.34 de görüldüğü üzere bu arayüzün, sistemin cevabının bir grafiğini veriyor olmasıdır.



Şekil 3.34. Motor Sürücü GUI Görüntüsü

BÖLÜM IV

4.1. Bütçe Planı

Proje kapsamında yapılan harcamaların listesi aşağıda verilmiştir.

NO	MALZEME ADI	ADET	BİRİM FİYATI (TL)	FİYAT (TL)
1	Arduino Nano	1	9,66	9,66
2	Arduino BTS7960B	1	33,00	33,00
3	TTL-RS485 Modül	1	9,32	9,32
4	DC-DC Voltaj Regülatörü	1	4,83	4,83
5	10K 8+1 Network Direnç	1	0,33	0,33
6	8 Pin Dip Switch	1	1,95	1,95
7	6 ‘ lı Yaylı Klemens	2	2,04	4,08
8	4 ‘lü Yaylı Klemens	2	1,58	3,16
9	3 ‘lü Yaylı Klemens	2	1,22	2,44
10	2 ‘li Yaylı Klemens	2	0,86	1,72
11	10x10 Bakır Plaket	1	2,00	2,00
12	1x40 Dişi Header	2	0,38	0,76
13	SMD Kırmızı Led	1	0,13	0,13
14	SMD Mavi Led	1	0,21	0,21
15	SMD 330 Ohm Direnç	2	0,01	0,02
16	SMD 10K Direnç	2	0,01	0,02
17	SMD 1 uF Kondansatör	2	0,08	0,16
TOPLAM				73,79

Tablo 4.1. Motor Sürücü Giderleri

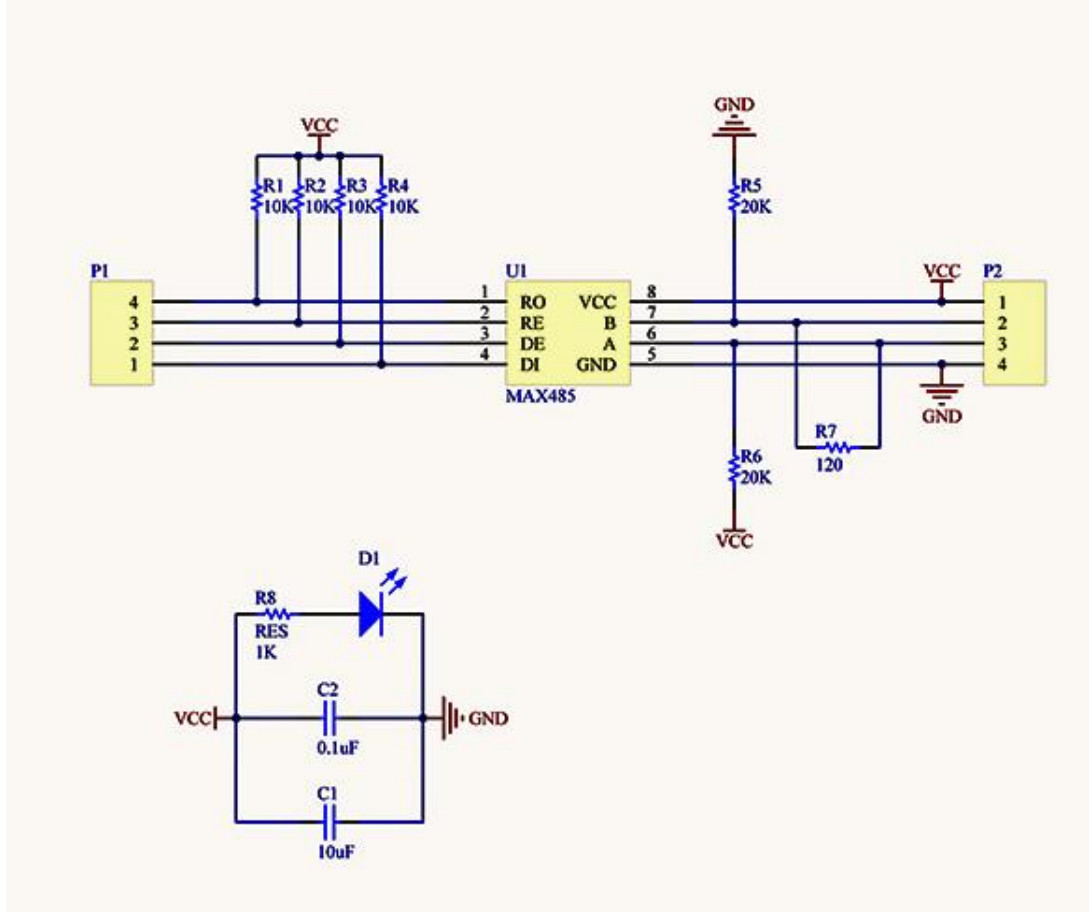
BÖLÜM V

5.1. Sonuç

Fırçalı DC Motor Sürücü tasarımı yapılmış olup başarılı bir şekilde imalatı gerçekleştirilmiştir. Enkoder ve potansiyometre ile geri beslemesi yapılabilen her türlü fırçalı DC motoru sürebilen bu motor farklı işlemcilerde ve güç katı entegre edilerek başarılı olunduğu görülmüştür. Pozisyon ve hız kontrolü yapabilen bu sürücü dört farklı projede kullanılarak farklı platformlarda test edilmiştir. Motor sürücünün entegre edildiği diğer platformların fotoğrafları aşağıda verilmiştir.

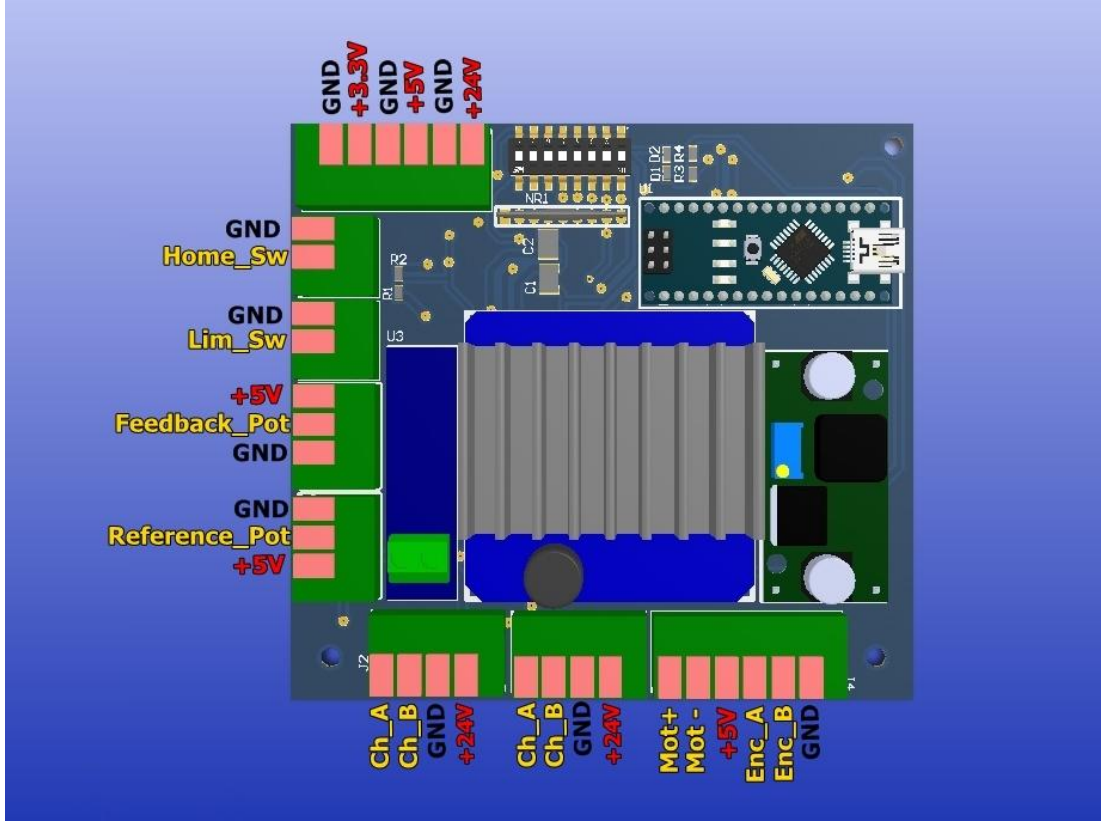
Sonradan home switch, limit switch ve farklı sensörlerin takılabileceği fonksiyonlar eklenmiştir. Bir GUI üzerinden kontrolü de yapılmış olup sistem cevabı grafiksel olarak çıkartılmıştır.

Projenin bir fonksiyonu çalışmayıp geriye kalan fonksiyonların tümü sorunsuz bir şekilde çalışmaktadır. Slave modda çalıştırılmak istenilen motor sürücüsü, verileri başarılı bir şekilde gönderdiği osiloskoptan tespit edilmiştir. Fakat karşı taraftan bu veriler çekilmek istendiğinde sıkıntı yaşanmıştır. Bunun ise hazır kullanılan MAX-485 modülünden kaynaklandığı düşünülmektedir. Çünkü master konumunda olan entegrede aşırı ısınma problemine rastlanmıştır. Modül üzerindeki çip değiştirilerek tekrar testleri yapılmış olup başarılı olunamamıştır. Fakat haberleşme modülünü imalatı yapıldığı zaman bu problemin ortadan kalkacağı düşünülmektedir. Bu yüzden haberleşmede kullanılacak olan şema Şekil 5.1 'de verilmiştir.



Şekil 5.1. RS-485 Devre Şeması

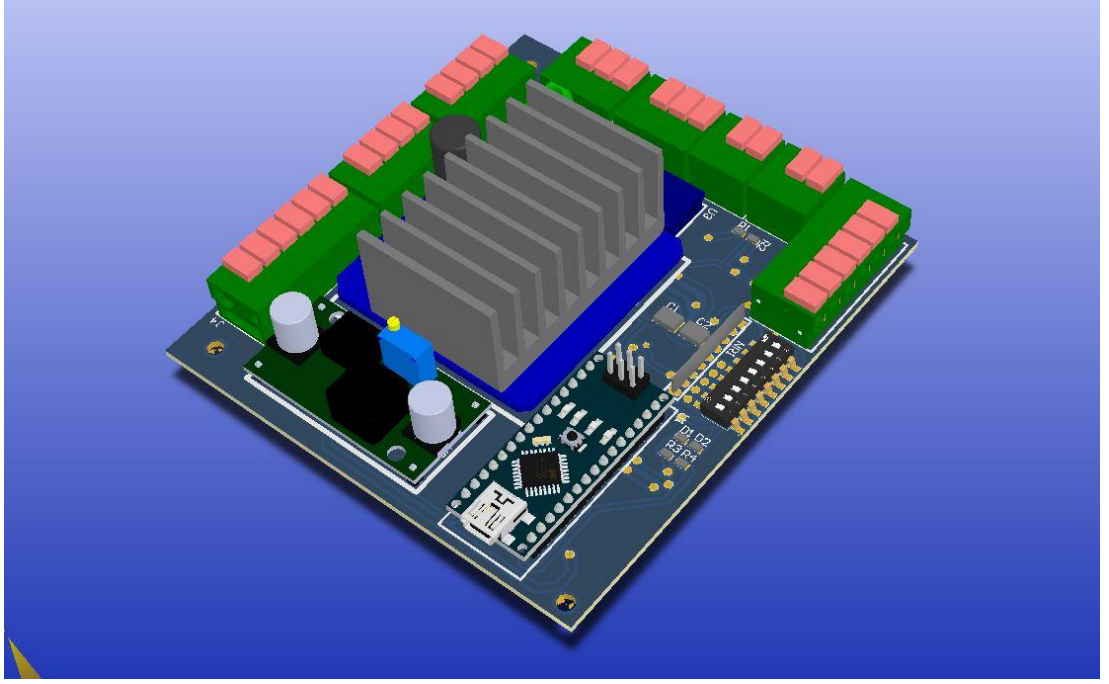
Bu devre başarılı bir şekilde motor sürücüsüne uygun ölçülerde imal edildiği takdirde haberleşme problemi ortadan kalkacağı düşünülmektedir. Haberleşme kodları sürücü için yazılmış olup kullanıma hazır bulunmaktadır.



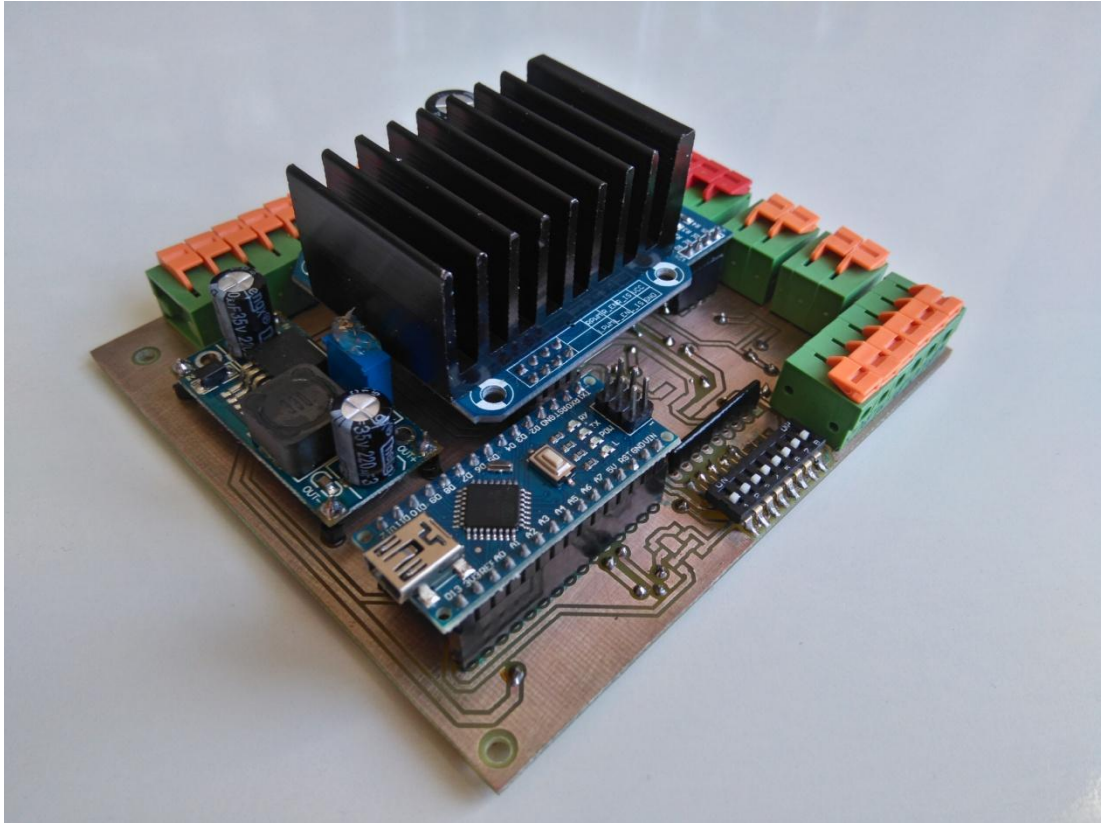
Şekil 5.2. Motor Sürücü Pin Giriş ve Çıkışları

Şekil 5.2 'de sürücüye kabloları bağlamak için giriş ve çıkışlar gösterilmiştir. Bu giriş ve çıkışlara dikkat edilmesi gerekmektedir. Yanlış bir bağlantıda sürücü düzgün çalışmayabilir ve hatta güç bağlantısı hatalarında ise kart tamamen yanabilir.

Şekil 5.3 ve şekil 5.4 'te ise motor sürücüsünün tasarım ve gerçek görüntüsü verilmiştir.

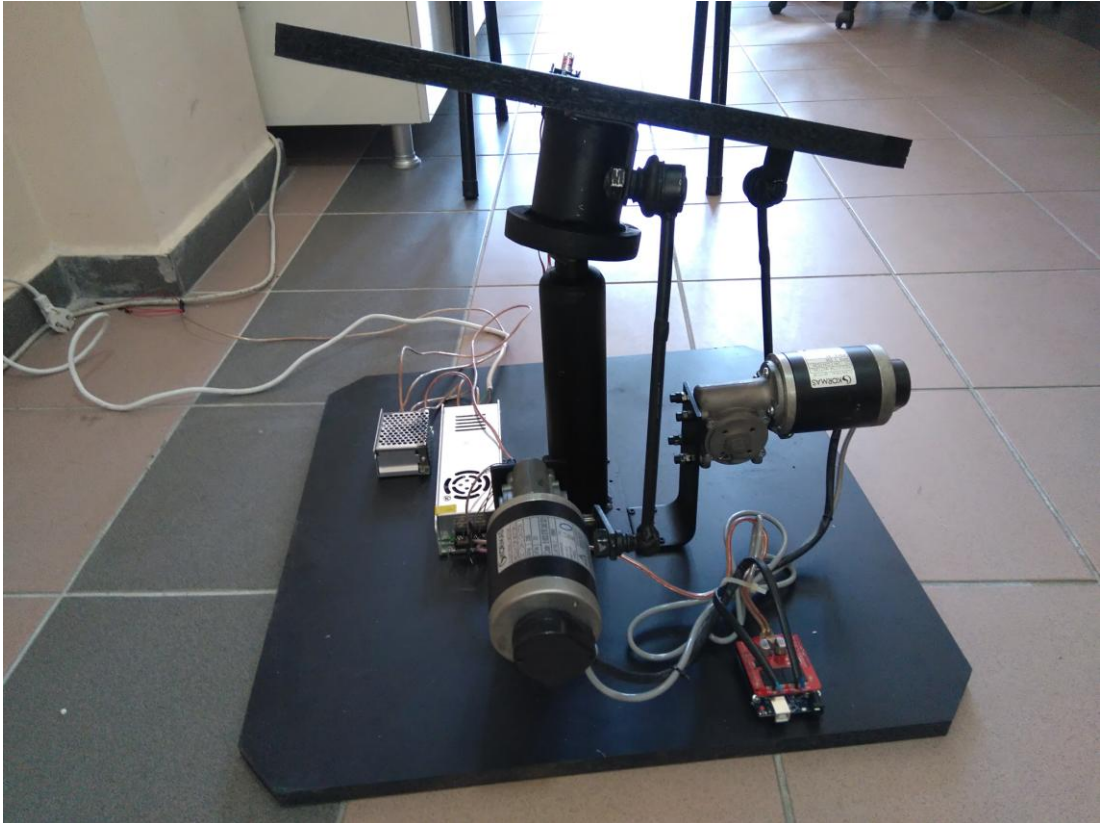


Şekil 5.3. Motor Sürücü 3D Görünümü

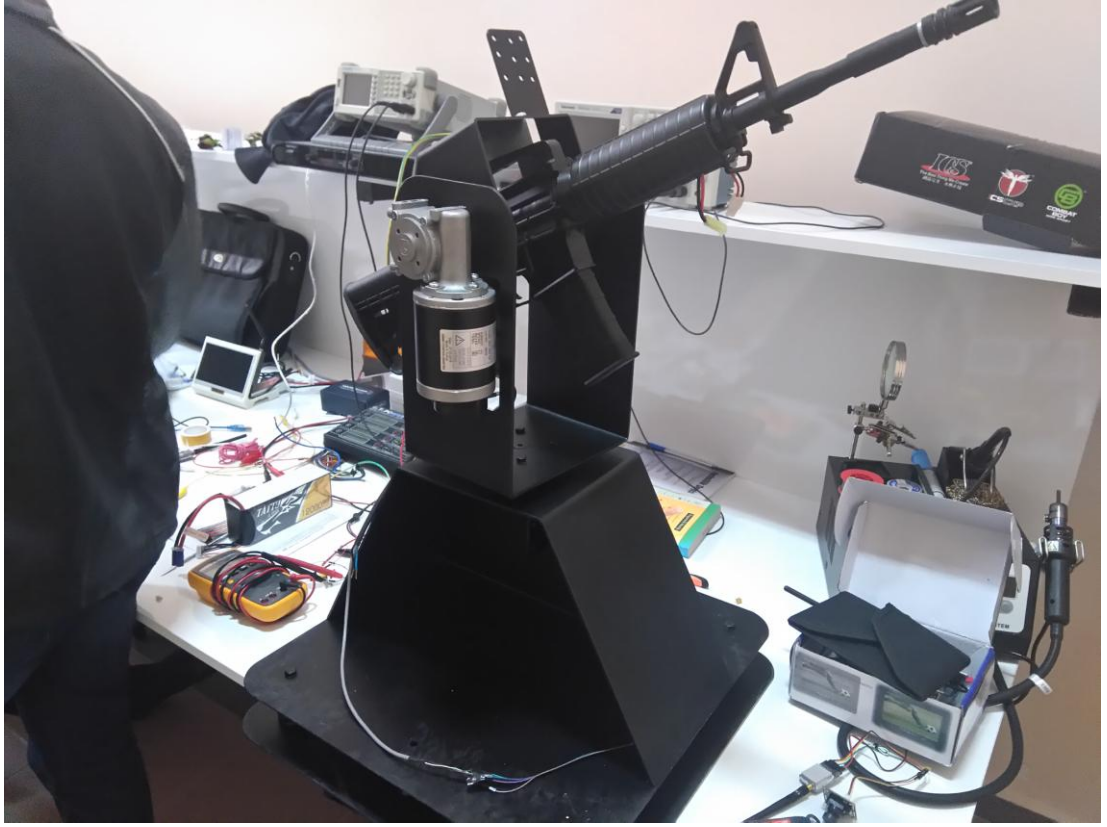


Şekil 5.4. Motor Sürücü Gerçek Görüntüsü

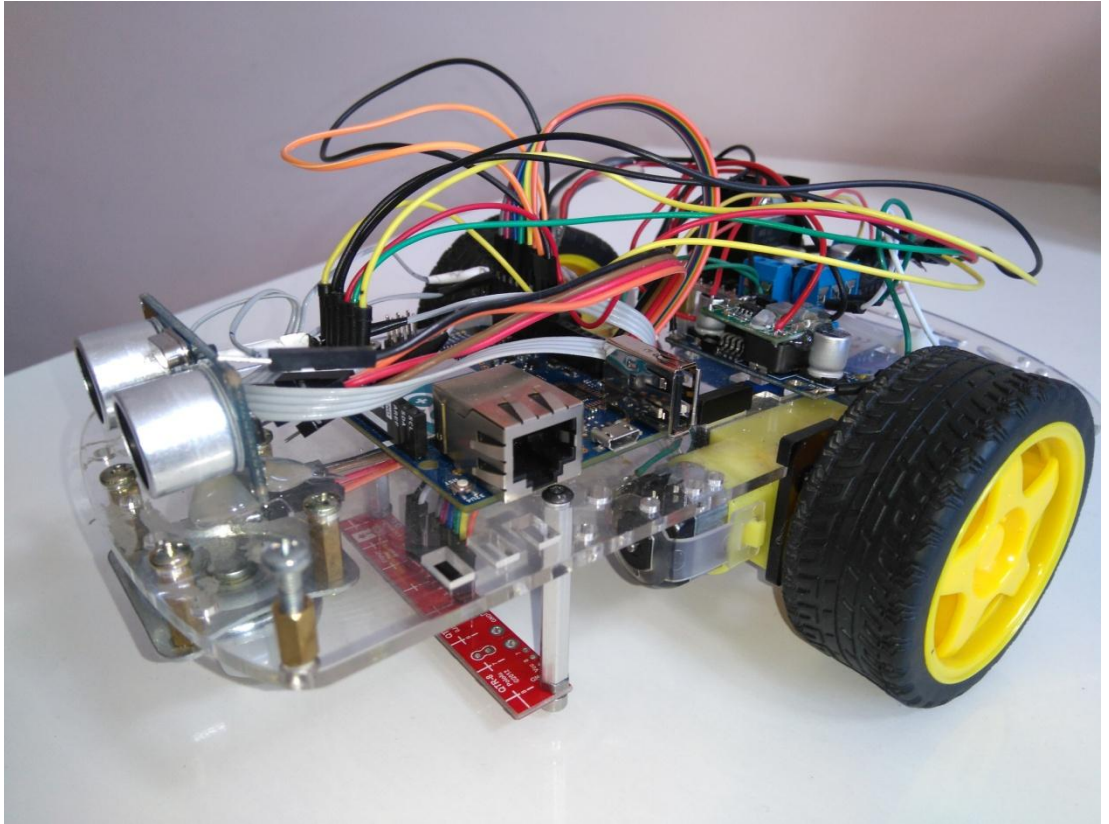
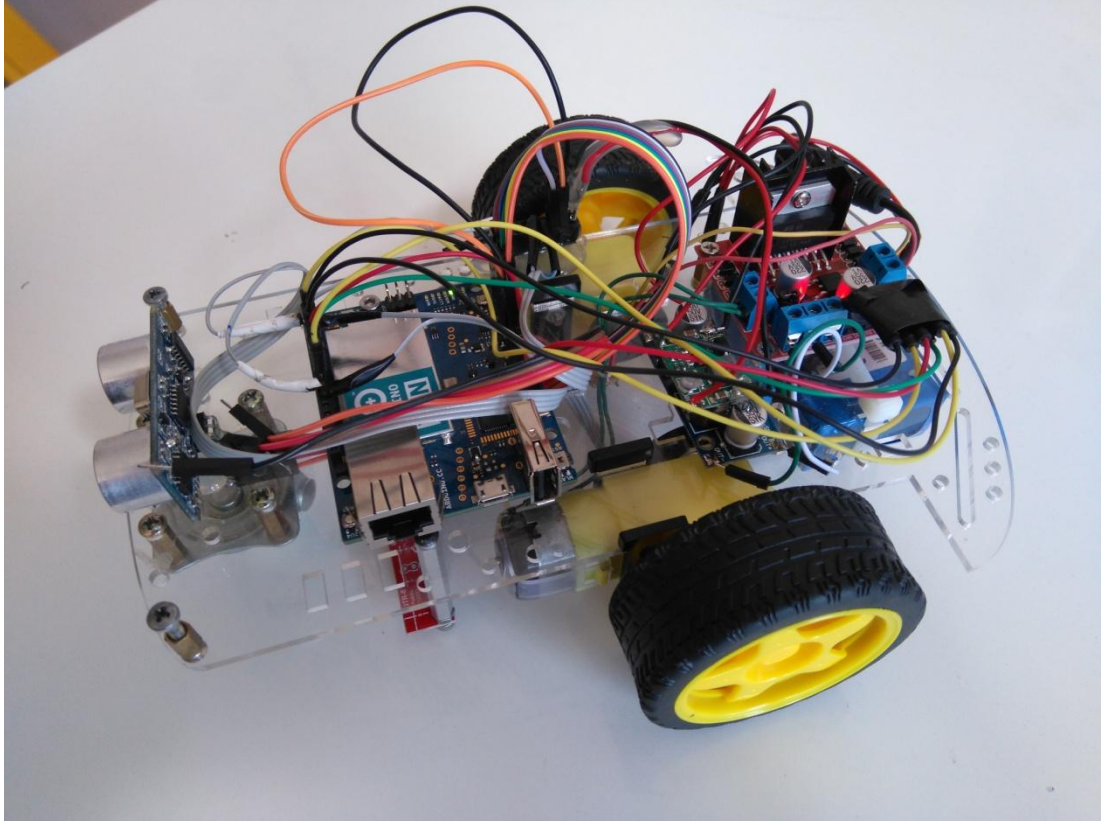
5.2. Motor Sürücünün Entegre Edildiği Farklı Platformlar



Şekil 5.5. 2 DOF Steward Stabilizasyon Platformu



Şekil 5.6. Hedef Takibi Yapan Silah Sistemi



Şekil 5.7. Çok Fonksiyonlu Mobil Robot



Şekil 5.8. Hedef Takibi Yapan Kamera Sistemi

KAYNAKÇA

1. Robert A. Paz, “The Design of the PID Controller”, Klipsch School of electrical and computer engineering June 12,2001
2. Rahul Malhotra, Tehbeer Kaur, gurpreet Singh deol “dc motor control using fuzzy logic controller ” international journal of advanced engineering sciences and Technologies, 2011
3. Fatih Köse, Kaplan Kaplan, H. Metin Ertunç “Otomatik Kontrol Ulusal Toplantısı”, TOK2013, 26-28 Eylül 2013, Malatya
4. Wang Zhongmin, Ye Hong “Xi'an Jiaotong University”, China, 2000
5. Germanton, D , Lehr, M “Variable Speed DC motor Controller Apparatus Particularly Adapted for Control of Portable-Power Tools”, 1991
6. F. Harashima, S. Kondo, "A design method for digital speed control system of motor drives", *Conf. Rec. PESC.*, pp. 289-297, 1982.
7. R. Krishnan, "Selection criteria for servo motor drives", *Proc. IEEE IAS Annu. Meeting*, pp. 301-308, 1986.
8. Department of Electrical Engineering, Chung Hua University, Hsinchu, 300 Taiwan, Republic of China, 2011
9. <https://www.arduino.cc/en/Main/ArduinoBoardNano>
10. <http://www.projehocam.com/pid-kontrol-algoritmasi-nedir/#content-anchor>
11. <http://www.projehocam.com/pid-kontrol-algoritmasi-nedir/>

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Davut YÜKSEKKAYA

Doğum Yeri : Siverek

Doğum Yılı : 10.04.1994



Haberleşme Bilgileri

Adres : Bağlar Mahallesi 20. Sokak No:34 Siverek / ŞANLIURFA

Telefon : 0507 005 78 33

E-posta : davutyuksekkaya@gmail.com