

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

LabVIEW İLE ADIM MOTORU HIZ KONTROLÜ

MORTEZA ABEDİNİ

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK PROGRAMI**

**DANIŞMAN
Y. DOÇ. DR. TUNCAY UZUN**

İSTANBUL, 2014

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LabVIEW İLE ADIM MOTORU HIZ KONTROLÜ

Morteza ABEDİNİ tarafından hazırlanan tez çalışması 14.01.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Y. Doç. Dr. Tuncay UZUN
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Y. Doç. Dr. Tuncay UZUN
Yıldız Teknik Üniversitesi

Prof. Dr. Herman SEDEF
Yıldız Teknik Üniversitesi

Prof. Dr. Hadi SARUL
Yıldız Teknik Üniversitesi

ÖNSÖZ

Projede, LabVIEW yazılımı kullanılarak bir adım motorunun görsel hız, adım, yön ve moment kontrolü gerçekleştirilmiştir. Kişisel bilgisayar ile dışarıdan haberleşebilen bu sistem, mikrodenetleyici, elektronik sürücü devresi, adım motoru ve ilgili yazılımlarının gerçekleştirilmesiyle oluşturulmuştur.

Birçok kaynağı inceleyip, prototipini tasarladığım bu devrede ilgili kaynaklar ve tezle ilgili verdiği diğer desteklerinden dolayı bana yardımcı olan Sayın Y. Doç. Dr. Tuncay UZUN 'a ve yazılım kısmında bana büyük destekçi olan Uğur ÜLKER'e sonsuz teşekkürlerimi sunuyorum.

Ocak, 2014

Morteza ABEDINI

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....	viii
KISALTMA LİSTESİ	ix
ŞEKİL LİSTESİ.....	x
ÇİZELGE LİSTESİ	xiii
ÖZET.....	xiv
ABSTRACT	xv
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	1
1.3 Orijinal Katkı.....	2
BÖLÜM 2	
ADIM MOTORLARI.....	3
2.1 Hareket Denklemleri, Dinamik ve Statik Denge.....	6
2.2 Hareket Denklemlerinin Çözümü.....	9
2.3 Elektrikle Tahrikte Kuvvet ve Momentler.....	10
2.3.1 Reaksiyon Kuvvet veya Momentleri.....	10
2.3.2 Potansiyel Kuvvet ve Momentleri	10
2.4 İşletme Karakteristikleri	11
2.4.1 Tahrik Edilen Makinaların İşletme Karakteristikleri	11
2.4.1.1 Açısal Hız ile Hiperbolik Olarak Değişen Yük Momenti Karakteristiği (Soyma-Moment karakteristiği)	11
2.4.1.2 Açısal Hıza Bağlı Olmayan Yük Momenti Karakteristiği (Kren-Moment Karakteristiği).....	13
2.4.1.3 Açısal Hız ile Orantılı Değişen Yük Momenti Karakteristiği (Kalender-Moment Karakteristiği)	14

2.4.1.4	Açısal Hızın Karesi ile Orantılı Değişen Yük Momenti Karakteristiği (Vantilatör-Moment Karakteristiği)	14
2.5	Adım Motoru ve Kontrolü	16
2.5.1	Bir Hareket Kontrol Sisteminin Bileşenleri	18
2.5.2	Yapılandırma, Prototip ve Geliştirme İçin Yazılım.....	19
2.5.3	Hareket Tipleri	21
2.5.3.1	Tek-Eksen, Noktadan Noktaya Hareket(Single-Axis, Point-to-Point Motion) 21	
2.5.3.2	Koordineli Çoklu Eksen Hareketi(Coordinated Multiaxis Motion) 21	
2.5.3.3	Karma Hareket.....	22
2.5.3.4	Kontrollü Hareket	22
2.6	Adım Motorların Kullanım Alanları	26
2.6.1	Adım Motorların Avantajları.....	27
2.6.2	Adım Motorların Dezavantajları	28
2.7	Adım Motorlarına Ait Bazı Önemli Kavramlar	29
2.7.1	Adım Tepkisi / Tek Adım Tepkisi (Single Step Response)/ Cevap Süresi 29	
2.7.2	Adım Oranı (Stepping Rate / Starting Pulse Rate)	30
2.7.3	Adım Açısı (Step Angle)	30
2.7.4	Adım Açısı Doğruluğu (Step Angle Accuracy)	30
2.7.5	Çözünürlük.....	30
2.7.6	Hedefi Aşırmak (Overshoot)	31
2.7.7	Adım Motor Moment Parametreleri.....	31
2.7.8	Moment - Hız Grafikleri.....	34
2.7.9	Etiket Voltajının Anlamı	34
2.8	Adım Motor Çeşitleri	34
2.8.1	Sabit Mıknatıslı (Permanent Magnet, PM) Adım Motorları	35
2.8.2	Değişken Relüktanslı (Variable Reluctance, VR) Adım Motorları	36
2.8.3	Hibrid Adım Motorları	37
2.9	Adım Motor Çalışma Prensipleri	39
2.9.1	Çift Kutuplu ve Tek Kutuplu Adım Motoru Ne Demektir?.....	40
2.9.2	Adım Motorlarında Uçların Bulunması	41
2.9.3	Adım Motorların Uyartımı	42
2.9.4	Tek-Faz Uyartım(1 Fazlı Tam Adımlı Sürüş)	42
2.9.5	İki-Faz Uyartım(2 Fazlı Tam Adımlı Sürüş)	43
2.9.6	Karma Uyartım(2 Fazlı Yarım Adımlı Sürüş).....	44
2.10	Adım Motorların Denetimi	45
2.10.1	Açık Döngü Denetim	45
2.10.2	Kapalı Döngü Denetim	46
2.11	Adım Motorların Çalıştırılma Şekilleri ve Teknikleri.....	47
2.11.1	Başla-Dur Adımlama Oranı.....	48
2.11.2	Düzgün Hız.....	48
2.11.3	Rampalama.....	48
2.11.4	Diğer Uyartım Yöntemleri	49
2.11.5	İki Fazlı Motorların Çalışma Şekilleri	49

2.11.6	Üç Fazlı Motorların Çalışma Şekilleri	50
2.11.7	Dört Fazlı Motorların Çalışma Şekilleri	51
2.12	Adım Motor Sürücü Devreleri Yapısı ve Çalışması	51
2.12.1	NPN Transistörlü Tek Kutuplu(Unipolar) Adım Motoru Sürücü Devresi 53	
2.12.2	ULN2003 Entegresi İle Tek Kutuplu Adım Motor Sürücü Devresi	54
2.12.3	MOSFET Transistörlü Unipolar Motor Sürücü Devresi	54
2.12.4	Transistörlü İki Kutuplu(Bipolar) Adım Motoru Sürücü Devresi	55
BÖLÜM 3		
LabVIEW İLE ADIM MOTORU KONTROL SİSTEMİNİN DONANIMI		56
3.1	Adım Motor Kontrolünde Kullanılan Donanım ve Yazılım	56
3.2	PIC18F2520 Mikrodenetleyicisi İle Tek Kutuplu Adım Motoru Kontrol Devresi	57
3.3	Mikrodenetleyici Birimi (PIC18F2520)	58
3.3.1	Teknik Özellikleri	59
3.4	RS-232 Sürücü Devresi (Max232).....	61
3.5	Motor Sürücü Devresi (ULN2803)	62
3.6	Adım Motoru	62
3.7	Sistemde Kullanılan Motor Sürücü Kartı	63
BÖLÜM 4		
LabVIEW İLE ADIM MOTORU KONTROL SİSTEMİNİN YAZILIMI		66
4.1	LabVIEW Yazılımı ve Kontrol Uygulamaları	66
4.1.1	Denetleyici Alan Ağı (DAA)	68
4.1.2	Sistemin Genel Yapısı	69
4.1.3	LabVIEW Kullanıcı Ara yüzünün Tanıtılması	70
4.1.4	Ön Paneli Oluşturmak	70
4.1.5	LabVIEW'da For ve While Döngüsü.....	73
4.1.6	For Döngüsü.....	73
4.1.7	While Döngüsü.....	74
4.1.8	Formül Düğümleri	74
4.1.9	Bir XY Grafiğini Daire Çizme İçin Kullanma	77
4.1.10	Paketleme.....	78
4.2	Örnek Uygulamalar	79
4.2.1	PID ve LabVIEW kullanarak Sıcaklık Kontrolü	79
4.2.2	Değişken Frekanslı Bir Sinüs Dalgası Oluşturmak	81
4.2.3	Sayıları Karşılaştırma	82
4.2.4	Sıcaklık Analizi	85
4.3	Gerçekleştirilen Sistemin Kişisel Bilgisayar LabVIEW Yazılımı	87
4.4	Gerçekleştirilen Sistemin Mikrodenetleyici C Yazılımı	91
4.4.1	Yazılımın Ekran Görüntüsü ve Akış Diyagramı	110
BÖLÜM 5		
SONUÇ VE ÖNERİLER		113

KAYNAKLAR	115
EK-A	
DEVRENİN PCB ŞEMASI.....	118
EK-B	
PIC18F2520 MİKRODENETLEYİCİ.....	121
EK-C	
MAX232 ENTEGRESİ	125
EK-D	
ULN2803 ENTEGRESİ	128
EK-E	
SİSTEMDE KULLANILAN ADIM MOTORU	133
ÖZGEÇMİŞ	135

SİMGE LİSTESİ

ωn	Açısal Hız
T_m	Dönme Momenti(Tork)
P	Güç
I	Akım
S	Hız
F	Kuvvet
r	Yol
V	Gerilim
$N.m$	Tutma Momenti(Newton metre)
N/Kg	Newton/Kilogram
μm	10^{-6} metre
$K\Omega$	10^3 Ohm
mA	10^{-3} Amper
W	Watt
Hz	Hertz
MHz	10^6 Hertz
$^\circ$	Derece

KISALTMA LİSTESİ

A/D	Analog-to-Digital converter
BJT	Bipolar Junction Transistor
CCW	Counter Clockwise
CNC	Computer Numerical Control
CSMA/CR	Carrier Sense Multiple Access with Collision Resolution
CTRL	Control
CW	Clockwise
DAA	Denetleyici Alan Ağı
DAQ	Data Acquisition
DC	Direct Current
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPL	Graphical Programming Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
LabVIEW	Laboratory Virtual Instruments For Engineering Workbench
LED	Light Emitting Diode
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MSSP	Master Synchronous Serial Port
PIC	Peripheral Interface Controller
PID	Proportional, Integral, Derivative
PLC	Programmable Logic Controller
PPS	Puls Per Second
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
SW	Switch
VI	virtual instrument

ŞEKİL LİSTESİ

Sayfa

Şekil 2. 1 Elektrik motorları.....	6
Şekil 2. 2 Basit tahrik sistemi.....	7
Şekil 2. 3 Kontraplak makinesi şematik	12
Şekil 2. 4 Soyma-Moment karakteristiği.....	12
Şekil 2. 5 Bir krenin kinematik diyagramı	13
Şekil 2. 6 Kren-Moment karakteristiği.....	13
Şekil 2. 7 Kalender-Moment karakteristiğine ait değişimler	14
Şekil 2. 8 Vantilatör-Moment karakteristiği.....	15
Şekil 2. 9 Bir vantilatörün hakiki yük momenti karakteristiği	16
Şekil 2. 10 Bir krenin hakiki yük momenti karakteristiği	16
Şekil 2. 11 Adım motoru	17
Şekil 2. 12 Adım motorunun kesiti	17
Şekil 2. 13 Bir hareket kontrol sisteminin bileşenleri.....	18
Şekil 2. 14 Hareket kontrol sistemi geliştirme süreci	19
Şekil 2. 15 NI MAX hareket kontrol sisteminin ekran görüntüsü.....	19
Şekil 2. 16 NI Motion yardımcısı.....	20
Şekil 2. 17 LabVIEW'da tek-eksen, noktadan noktaya hareket.....	21
Şekil 2. 18 LabVIEW'da koordineli çoklu eksenli hareket	21
Şekil 2. 19 LabVIEW'da karma hareket	22
Şekil 2. 20 LabVIEW'da kontrollü hareket.....	22
Şekil 2. 21 NI Instruments yapımı sürücü devresi ve adım motorları örneği	23
Şekil 2. 22 NI Instruments yapımı 4 eksenli sürücü devresi.....	24
Şekil 2. 23 Adım ve yön değiştirici mikro sürücüler	25
Şekil 2. 24 DS3041 adım motoru sürücüsü	25
Şekil 2. 25 ROB-09402 adım motoru sürücüsü	26
Şekil 2. 26 Adım motoru kontrol blok diagramı	56
Şekil 2. 27 Adım motorlarının adım tepkisi.....	29
Şekil 2. 28 Adım motorun rotor pozisyonu-zaman eğrisi	31
Şekil 2. 29 Motor mil bağlantısı.....	33
Şekil 2. 30 Moment – Hız grafiği.....	34
Şekil 2. 31 Sabit mıknatıslı adım motoru	35
Şekil 2. 32 4fazlı bir adım motor.....	36
Şekil 2. 33 3fazlı değişken relüktanslı adım motor	36

Şekil 2. 34 Hibrid adım motorunun rotor yapısı.....	37
Şekil 2. 35 Hibrid adım motorunun stator sargıları.....	38
Şekil 2. 36 Adım motoru temel prensip şeması	39
Şekil 2. 37 Çift kutuplu adım motoru.....	40
Şekil 2. 38 Tek kutuplu adım motoru	40
Şekil 2. 39 Bipolar adım motorunun uçları	41
Şekil 2. 40 Tek kutuplu adım motorunun uçları	42
Şekil 2. 41 8 uçlu adım motor.....	42
Şekil 2. 42 Tek fazlı uyartım	43
Şekil 2. 43 İki fazlı uyartım	44
Şekil 2. 44 Karma uyartım	45
Şekil 2. 45 Karma uyartım	46
Şekil 2. 46 Kapalı döngü denetimi	47
Şekil 2. 47 İki fazlı adım motor sargıları ve çalışma modları	49
Şekil 2. 48 Üç fazlı adım motor sargıları ve çalışma modları	50
Şekil 2. 49 Dört fazlı adım motor sargıları ve çalışma modları.....	51
Şekil 2. 50 Adım motor sürücü devresinin blok diyagramı	52
Şekil 2. 51 NPN transistörlü tek kutuplu adım motoru sürücü devresi.....	53
Şekil 2. 52 ULN2003 entegresi ile unipolar adım motoru sürücü devresi	54
Şekil 2. 53 MOSFET transistörlü tek kutuplu adım motoru sürücü devresi.....	55
Şekil 2. 54 Tansistörlü çift kutuplu adım motoru sürücü devresi	55
Şekil 3. 1 Sürücü devresi	58
Şekil 3. 2 PIC18F2520 mikrodenetleyicisinin bacakları	59
Şekil 3. 3 Max232 entegresi	61
Şekil 3. 4 ULN2803 entegresi	62
Şekil 3. 5 Adım motorunun sürücü devresine bağlantı şekli	63
Şekil 3. 6 Motor sürücü kartı.....	64
Şekil 3. 7 Adım motor denetim donanımının baskılı devresi	65
Şekil 3. 8 Adım motor denetim donanımı devresinin simülasyonu	65
Şekil 4. 1 Sistemin genel yapısı.....	69
Şekil 4. 2 “Ön Panel” ve “Blok Diyagram” görüntüleri	70
Şekil 4. 3 Ön panel	70
Şekil 4. 4 Ön panel	71
Şekil 4. 5 “Fonksiyon Paleti” ve “Kontrol Paleti”	72
Şekil 4. 6 Blok diyagram	73
Şekil 4. 7 Sıcaklık kontrol diyagramı	79
Şekil 4. 8 Ön panel	80
Şekil 4. 9 Blok diyagram	80
Şekil 4. 10 Ön panel	81
Şekil 4. 11 Blok diyagram	82
Şekil 4. 12 Ön panel	83
Şekil 4. 13 Blok diyagram	83
Şekil 4. 14 Blok diyagram	84
Şekil 4. 15 Blok diyagram	75
Şekil 4. 16 Formül düğümü blok diyagramı	75

Şekil 4. 17 Formül düğümü	76
Şekil 4. 18 Formül düğümü	76
Şekil 4. 19 Ön panel	77
Şekil 4. 20 Blok diyagram	78
Şekil 4. 21 Ön panel	85
Şekil 4. 22 Blok diyagram	86
Şekil 4. 23 LabVIEW ön panel görünümü.....	88
Şekil 4. 24 LabVIEW blok diyagram görünümü	88
Şekil 4. 25 Blok diyagram time out kısmı	89
Şekil 4. 26 Blok diyagram hız ayar kısmı	89
Şekil 4. 27 Blok diyagram adım ayar kısmı	90
Şekil 4. 28 Blok diyagram moment ayar kısmı	90
Şekil 4. 29 Blok diyagram güç ayar kısmı	91
Şekil 4. 30 Yazılımın ekran görüntüsü.....	111
Şekil 4. 31 Motor sürücüsü devresinin akış diyagramı	112

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2. 1 Tek faz uyarım çizelgesi (CW)	48
Çizelge 2. 2 İki faz uyarım çizelgesi.....	48
Çizelge 2. 3 Yarım adım uyarım çizelgesi.....	48
Çizelge 4. 1 DAA mesajları	69

LabVIEW İLE ADIM MOTORU HIZ KONTROLÜ

Morteza ABEDINI

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Y. Doç. Dr. Tuncay UZUN

Bu çalışmada adım motorunun hız, dönme yönü, adım durumu ve momentunun kontrolü için uygun bir yöntem gerçekleştirilmiştir. Bu parametrelerin kontrolü, LabVIEW programı ve sürücü kartı yardımıyla gerçekleştirilmiştir. Adım motoru, sürücü kartın bilgisayarla bağlantısını seri port üzerinden ve gerekli LabVIEW VI' ını tasarladıktan sonra dikkatli bir şekilde kontrol edilir. Bunun için önce sistemin akış diyagramı oluşturmuş ve sonra da ona göre LabVIEW blok diyagramı tasarlanmıştır. Bu yöntem, adım motorun kontrol şeklinin değiştirilebilme yeteğine sahiptir. Bu nedenle önerilen yöntem, adım motorlarının, farklı alanlarda örneğin CNC makinesi gibi, kullanılmasını sağlar.

Anahtar Kelimeler: Adım Motoru, LabVIEW, Sürücü

STEPPER MOTOR SPEED CONTROL WITH LabVIEW

Morteza ABEDINI

Department of Electronics and Communications Engineering

MSc. Thesis

Adviser: Assist. Prof. Dr. Tuncay UZUN

This paper describes an appropriate method for control of stepper motor from viewpoints of velocity, direction of rotation, step status and torque. Control of these parameters is performed by LabVIEW software and driver card. The stepper motor can be controlled accurately by configuring the driver controller card over the serial port on computer and writing the appropriate program in block diagram of LabVIEW using the card drivers. At the start of this proposed method, the control flowchart is created and then the graphical programming is accomplished in LabVIEW environment according to flowchart control. This method is capable to change the configuration of control. Thus, the proposed method can be used for control of stepper motor that used in different machinery such as CNC machines.

Keywords: Stepper Motor, LabVIEW, Driver

1.1 Literatür Özeti

Elektrik motorlar kabaca, doğru akım ve alternatif akım motorları diye ikiye ayrılabilir. 1880'lerde ilk olarak doğru akım bulunduğundan ilk motorlar da bu akımla çalışanlar olmuştur. 1890'larda alternatif akımın bulunmasıyla indüksiyon motoru adıyla bilinen bir alternatif akım motoru basitliğinden dolayı yaygınlaşmıştır. İlk ticari motor 1880'lerde, elektrik ve manyetizmada 1820'lerin başlarında yapılan temel keşiflerden yarım asır sonra ortaya çıkmıştır. İlk defa 1821'de Michael Faraday, bir mıknatıs ve hareketli telden sürekli bir dönme hareketi elde edebileceğini göstermiştir.

Adım motorların, endüstriyel ve elektronik uygulamalarda özellikle 90'lı yıllarda bilgisayar sistemlerinin de yaygınlaşmasıyla birlikte, kullanımı oldukça artmıştır. En basitinden, bilgisayarımızdaki CD okuyucusunda ve hard diskler de bu teknolojiye başvurulmuştur. Adım motorlar, girişlerine uygulanan lojik sinyalleri dönme hareketine çevirirler. İstedığınız yönde ve derecede döndürebileceğiniz adım motorlar, hassas hareketleri sayesinde, birçok cihazda konum kontrolü amacıyla kullanılmaktadır.

1.2 Tezin Amacı

Bu çalışmada adım motorunun hızı, dönme yönü, adım durumu(tam adım veya yarım adım) ve momentinin kontrolü görsel olarak sağlanmıştır. Bu amaç için LabVIEW grafiksel kontrol yazılımı kullanılmıştır.

1.3 Orijinal Katkı

Bu çalışmada adım motorun kontrolü görsel olarak LabVIEW programı ile yapılmıştır. LabVIEW programının kütüphanesinde hazır ölçme ve kontrol araçları yer almaktadır. Bu tezin gerçekleştirilmesi sırasında bu araçlar incelenmiş ve bazı araçların kullanımına ilişkin örnekler verilmiştir. LabVIEW programının üreticisi NI Instrument tarafından verilen örneklerde veya diğer firmaların örneklerine bakıldığında hazırlanan adım motor sürücü devreleri, kullanıcı tarafından programlanarak kullanılmaya uygun olarak yapıldığı görülmüştür.

Bu çalışmada tasarlanan LabVIEW sanal cihazına (VI, Virtual Instrument) benzer bir örneğe rastlanmadığından, bu bitirme tezinin kendi türündeki ilk örnek olduğu düşünülebilir. Bu doğrultuda LabVIEW VI'ları da genel kapsamlı ve çoğunlukla motor dönme hızı ve yönünün kontrolü için hazırlanmış, ancak bu bitirme tezinde yön ve hız kontrolü dışında adım tipi (yarım adım, tam adım) kontrolü ve moment kontrolü de yapmaktadır. Moment kontrolü ağır veya hafif yüklerin aynı sistemde çalışabilmesini sağlamaktadır ve hafif yüklerde yüksek momentte çalışmaktan kaçınılmıştır. Çünkü ağır yükler için motorun kullandığı güç iki katına çıkarak hem tüketimi artırmakta, hem de motorun daha fazla ısınmasına neden olmaktadır. Bu sistemin yazılımında, adım motorun konumu her an tespit edilebilmektedir ve anında yön değişimi uygulanarak motorun gecikmesiz olarak, yön değişimi sağlanmıştır. Ayrıca kontrol LabVIEW yazılımı kullanılarak yapıldığı için oldukça güzel ve kullanımı kolay bir görsel ara yüze sahiptir. Bu sayede adım motor kontrolü çok hassas, hızlı, hatasız ve güvenilir bir hale getirilmiştir. Sistemin donanımında kişisel bilgisayarla dışarıdan seri iletişimle çalışabilen bir mikrodenetleyici kullanılarak, akıllı bir izleme ve kontrol gerçekleştirilmiştir.

BÖLÜM 2

ADIM MOTORLARI

Otomasyonun doğuşu 1980'li yıllardan itibaren sanayi devriminin ardından gerçekleşmiştir. Bu dönemlerdeki ilkel otomasyon sistemlerinin, insan gücünden alınıp makine gücüne aktarılması düşünülmüştür. İşlerin makinelere aktarılması sanayi sektörlerini hızlandırmış daha çok ilgi doğurmasına neden olmuştur. Daha sonra da seri üretim anlayışının da sanayide ortaya çıkmasıyla birlikte otomasyon sistemlerine duyulan ihtiyaç biraz daha artmaya başlamıştır.

Bilgisayarın icadı, elektronik, elektrik vb. alanlardaki hızlı gelişmeler, otomasyon gelişmesini de hızlandırmıştır. Bu dönemde otomasyon sistemleri ilkelikten kurtulup çok daha kapsamlı süreçler için hazırlanmaya başlanmışlardır. Otomasyon önceleri genellikle üretime yardım için kullanıyorken artık üretimin yanında üretimin kontrolünü de üstlenmiştir. Milenyum çağıyla birlikte otomasyon sistemleri altın çağına girmiş ve artık geleceğin olmazsa olmazı haline gelmiştir.

Artık bir üretim tesisi için kurulan otomasyon sistemi, tek başına tüm fabrikanın denetim, analiz ve dokümantasyon işlemlerini üstlenebilecek kapasitededir. Otomasyonun özellikleri işletmede kolaylık sağlar. Üretim kontrol ve personel giderlerinde sanayiciye tasarrüfa sağlar. Sürekli işlerde istikrarlı şekilde çalışma sağlar ve çalışması istenildiği takdirde kesintisiz çalışır. Otomasyon sistemlerinde kullanılan gereçler belli bir standartlara tabii tutularak üretilip kullanılmaktadır ki IEC61131, IEEE488, IEEE802 ve IEEE1394 gibi standartları buna örnek verebiliriz.

Motor elektrik enerjisini mekanik enerjisine çeviren bir makinadır. Elektrik kaynağına bağlandığı zaman bir mili döndürerek bir hareket meydana getirir. Günlük hayatımızda

milyonlarca motor kullanılır. Evde bulunan elektrik süpürgesi, çamaşır makinası, dikiş makinası ve karıştırıcıları hep motorlar çalıştırır. Modern bir fabrikada ise, bantların çalıştırılmasında, baskı makinalarında, değirmenlerde hep elektrik motoru vardır. Motorlar genellikle tüm yüklendikleri zaman tam verimlidirler. Büyük motorlar, küçük motorlardan, yüksek hız motorları düşük hız motorlarından daha verimlidir. Beygir gücünün küçük bir kesiminden, 5000 beygir gücüne veya daha büyük güce sahip motorlar mevcuttur.

Elektrik motorları kabaca, doğru akım ve alternatif akım motorları diye ikiye ayrılabilir. 1880'lerde ilk olarak doğru akım bulunduğundan ilk motorlar da bu akımla çalışanlar olmuştur. 1890'larda alternatif akımın bulunmasıyla indüksiyon motoru adıyla bilinen bir alternatif akım motoru basitliğinden dolayı yaygınlaşmıştır. Daha sonra, özellikle senkron ve üniversal motor olmak üzere, diğer tür alternatif akım motorları gelişmiş ve kullanılmıştır. Günümüzde alternatif akım yaygın olmasına karşılık, doğru akım motorlarının uygulamada önemli bir yeri vardır. Özellikle değişik hızların gerekli olduğu, asansör, el aleti ve konveyörlerde doğru akım motorları kullanılır.

İlk ticari motor 1880'lerde, elektrik ve manyetizmada 1820'lerin başlarında yapılan temel keşiflerden yarım asır sonra ortaya çıkmıştır. 1820'de Hans Christian Oersted elektrik akımının bir pusula iğnesini hareket ettirdiğini gözlemledi. İlk defa 1821'de Michael Faraday, bir mıknatıs ve hareketli telden sürekli bir dönme hareketi elde edebileceğini gösterdi. Ayrıca 1831'de, at nalı biçimindeki bir mıknatısın ağzına yerleştirdiği döner bir bakır diskle, elektromanyetik indüklemeyi buldu. 1832'de Fransız Hippolyte Pixii ilk indüklemeli makineyi gerçekleştirdi, bu elle döndürülen manyeto elektrik bir üreteçti. Bir Amerikalı demirci olan Thomas Davenport 1837'de elektrik motoru için ilk patenti aldı. Ancak her buluş gibi bunun da uzun bir gelişme devresine ihtiyacı vardı. 1860 ve 1870'lerde çok çalışmalar yapıldı. Hatta bu sırada doğru akım jeneratörünün tersine çalışabileceği, yani başka bir doğru akım jeneratörü tarafından bir doğru akım motoru gibi çalıştırılabileceği anlaşıldı. 1865'de Maxwell, elektromanyetik dalgalara ilişkin kuramını yayınladı. 1871'de Belçikalı Zenobe Gramme, bilimler akademisine alternatörün atası olan dinamoyu sundu. 1873'de Viyana Sergisi'nde, mutlu bir rastlantı Gramme'in dinamosunun tersinir bir makine olduğunu

ve motor(senkron) olarak kullanılabilceğini açığa çıkardı.1887 Maxwell'in kuramı H. Hertz tarafından deneysel olarak kanıtlandı. 1888'lerin sonuna doğru akım motorları büyük miktarlarda imal edilmeye başlandı. Büyük şehirlerdeki ulaşım ihtiyacı motorlar için yeni imkânlar ortaya koydu. Atla çekilen arabaların elektrik motorlarıyla da hareket ettirilebileceği fikri çıktı. Bu arada elektrik motorlarının teorisinde de gelişmeler kaydedildi. Bütün bunlara rağmen, 1880 ortalarında, doğru akımın ancak birkaç kilometrelik mesafeye dağıtılabilmesi ve elektrik motorunun pahalı olması, kullanımın gelişmesini engelleyen en önemli faktörlerdi.

Bu sırada alternatif akımın uzak yerlerde ekonomik olacağı anlaşıldı. 1888'de İtalyan Galileo Ferrari birbirinden 90° faz farkı olan iki alternatif manyetik alanının sabit dönen bir manyetik alan olarak görülebileceğini gösterdi. Ayrıca, bir tek akımın, fazları farklı iki manyetik alan doğuran fazları farklı iki akıma ayrılabilceğini gösterdi. Dönen tek bir manyetik alanın motorun rotasında bir dönme meydana getirdiğini müşahade etti. Böylece ilk indüksiyon motoru doğmuşsa da bunun sanayideki uygulamasını Ferrari devam ettirememiştir. 1888-1896 arasında Nikola Tesla bağımsız olarak geliştirdiği indüksiyon motorunun patentlerini aldı. George Westinghouse, Tesla'nın patentlerini alarak ileri çalışmalar yaptırdı ise de, ekonomik kriz ve elektrik akımı tekniğinin gelişmeleri bu çalışmaları engelledi. 1890'lardan itibaren çeşitli ülkelerde alternatif akım motorları ortaya çıktı. Çok fazlı elektrik motorları geliştirildi. Böylece elektrik motorunun temelleri atılmış oldu. Bundan sonraki gelişmeler genellikle esası değıştirmeden yapılan geliştirmeler oldu. Elektrik motorları yapılış itibarıyla başlıca iki kısımdan meydana gelir; **Stator:** Sabit olan kısımdır. Manyetik alan elde etmeye yarayan bir mıknatıstır. Motorların bir kısmında bu parça elektromıknatıstır. Bu elektromıknatıstan akım geçtiği zaman, kutuplar arasında manyetik alan meydana gelir.

Rotor: Mıknatısın kutupları arasında dönme hareketi yapabilen bir makara sistemidir. Elektrik motorları kullanma maksatlarına göre çeşitli büyüklükte ve türde imal edilir.

Genellikle bunlar iki grupta toplanabilir.

1. Elektromanyetik makinalar: Bu gruba indüksiyon, senkron, doğru akım, doğru akım çok fazlı kolektörlü motorlar girebilir:

2. Manyetik motorlar (makinalar): Bu gruba ise manyetik dirençli eşzamanlı motorlar, kesiklik motorları, selenoidler ve röleler girer.

Elektromanyetik motorların hacimleri büyüdükçe verimleri artar. Manyetik motorlarda verim artışı için hacim küçülür [1] - [3].



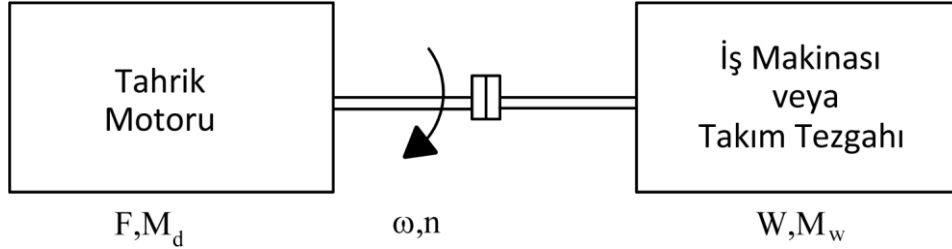
Şekil 2. 1 Elektrik motorları

Elektrikle tahrikte hareket olayları aşağıdaki gibidir;

2.1 Hareket Denklemleri, Dinamik ve Statik Denge

Elektrik motoru tahrik sisteminin çevirici elemanı olup iş makinaları (tahrik edilen) için gerekli cer kuvveti veya döndürme momentini sağlar. Mil, kavrama, kasnak ve kayış, dişli, dişli takımı vb. bir elemandan meydana gelen ara transmisyon veya ayar mekanizmasının görevi, elektrik motoru milindeki hareketi (cer kuvveti veya döndürme momenti) iş makinesi miline aktarmak ve aynı zamanda da ayar fonksiyonunu yerine getirmektedir. Ara transmisyon mekanizması olarak sadece mil ve kavrama kullanılması halinde hareket motor milinden tahrik miline aynen aktarılır. Lineer motor dışında bütün elektrik motorları ve ekseri iş makineleri ve takım tezgâhları döner hareketlidir. Taşıt araçları, kaldırma makineleri ve takım tezgâhlarında ise hem doğrusal, hem de döner hareket yapan kısımlar bulunur. Hem doğrusal hem de döner hareket eden kütlelere sahip bu gibi karmaşık sistemlerin hareket olayını en kısa yoldan

hesaplayabilmek için, sistemin yalnız doğrusal hareket eden veya yalnız döner hareket eden eşdeğer sisteme indirgenmesi gerekir. İndirgeme işlemi daha ileriki bölümlerde incelenecek olup, etütlerimizi elektrik motorunun iş makinesine mil ve kavrama üzerinden doğrudan bağlandığı sistemde yürütelim. Böyle bir sistemin blok diyagramı Şekil 2. 2’de verilmiştir.



Şekil 2. 2 Basit Tahrik Sistemi

Doğrusal hareket eden ve eşdeğer kütlesi m olan bir sistemin hareket olayının ana denklemi, 2.Newton kanunu'na göre,

$$F - W = m \cdot a, F - W = m \frac{dv}{dt} \quad (2.1)$$

Şeklinde yazılabilir. Burada, F cer kuvveti, W toplam mukavemet kuvvetidir. 2.Newton kanununa göre döner hareketli bir sistemin hareket olayının ana denklemi,

$$M_d - M_w = J \cdot \alpha, \\ M_d - M_w = J \frac{d\omega}{dt} \quad (2.2)$$

Şeklinde yazılabilir. Burada M_d döndürme momentini, M_w mukavemet momentini gösterir. Doğrusal ve döner hareket için elde edilen (2.1) ve (2.2) no'lu diferansiyel denklemlere elektrikle tahrikin ana denklemleri denir.

Atalet momentinin savurma momentinden hesaplanan değeri ile açısal hızın $\omega=2\pi n/60$

ifadesinden hesaplanan $\frac{d\omega}{dt} = \frac{2\pi}{60} \frac{dn}{dt}$ değeri (2.2) denkleminde yerine konulursa,

denklemin yeni şekli,

$$M_d - M_w = \frac{GD^2}{4g} \frac{2\pi}{60} \frac{dn}{dt}, M_d - M_w = \frac{GD^2}{375} \frac{dn}{dt} \quad (2.3)$$

Şeklinde elde edilir. (2.1) ve (2.2) denklemlerinin analizinden aşağıdaki sonuçlar elde edilir.

1.Durum

$$F - W = m \frac{dv}{dt} = 0, \quad F = W, \quad \frac{dv}{dt} = 0, \quad v = \text{sabit}$$

$$M_d - M_w = J \frac{d\omega}{dt} = 0, \quad M_d = M_w, \quad \frac{d\omega}{dt} = 0, \quad \omega = \text{sabit}$$

Bu durumda ivme sıfır olduğundan hız sabittir. Sistem sükûnette ise sükûnetini muhafaza eder, harekette ise sabit hızla hareketine devam eder. Bu şekilde hareket eden bir sistem statik denge halindedir ve hareket problemi de statikteki yöntemler yardımıyla çözülebilir.

2.Durum

$$F - W = m \frac{dv}{dt} > 0, \quad \frac{dv}{dt} = \alpha > 0, \quad F = W + m \frac{dv}{dt}$$

$$M_d - M_w = J \frac{d\omega}{dt} > 0, \quad \frac{d\omega}{dt} = \alpha > 0, \quad M_d = M_w + J \frac{d\omega}{dt} > 0$$

Bu halde sistem pozitif ivmeye sahiptir. Tahrik sistemi sükûnette ise harekete geçer, harekette ise zamanla artan hızla hareketine devam eder (hızlanma hareketi). Tahrik sistemlerine yol vermede bu farkın, dolayısıyla ivmenin pozitif olması şarttır. Bu hareket şeklinde F cer kuvveti, W cismin harekete karşı gösterdiği direnme kuvveti ile $m \left(\frac{dv}{dt} \right)$ dinamik kuvvetin toplamına eşittir. Bu koşullar altında hareket eden bir cisim dinamik denge halindedir. Benzer durum döner hareketli tahrik sistemi için de geçerlidir.

3.Durum

$$F - W = m \frac{dv}{dt} < 0, \quad \frac{dv}{dt} = \alpha < 0, \quad F = W + m \frac{dv}{dt} < 0$$

$$M_d - M_w = J \frac{d\omega}{dt} < 0, \quad \frac{d\omega}{dt} = \alpha < 0, \quad M_d = M_w + J \frac{d\omega}{dt} < 0$$

Bu durumda, sistem hareket halinde ise, hareketine azalan hızla devam eder. Sabit hızla çalışan bir sistemin devre dışı bırakılması veya motor çalışmadan fren çalışmaya geçilmesi halinde bu hareket şekli söz konusudur.

2.2 Hareket Denklemlerinin Çözümü

Doğrusal veya döner hareketli tahrik sistemlerinin hızı veya hız-zaman diyagramları (2.1) ve (2.2) no.lu ifadelerden aşağıdaki gibi bulunabilir.

$$\frac{dv}{dt} = \frac{F-W}{m}, \quad v = v_0 + \int \frac{F-W}{m} dt \quad (2.4)$$

$$\frac{d\omega}{dt} = \frac{Md - Mw}{J}, \quad \omega = \omega_0 + \int \frac{Md - Mw}{J} dt \quad (2.5)$$

Bu bağıntılarda v_0 ve ω_0 başlangıç hızlarıdır. Tahrik sistemlerinin çoğunda eşdeğer kütle ve atalet momenti hareket esnasında hıza bağlı olmayıp sabit kalır. Fakat tahrik eden ve tahrik edilen ünitelere ait kuvvet ve momentler ise sabit olmayıp çoğu zaman hıza, bazen yola ve çalışma koşullarına göre değişir. Bundan dolayı (2.4) ve (2.5) bağıntılarından hız denklemlerinin bulunabilmesi için,

$$F = f(v), \quad \text{Çer kuvveti – hız,}$$

$$W = f(v), \quad \text{Mukavemet-hız,}$$

$$Md = f(\omega), \quad \text{Döndürme momenti-açısal hız,}$$

$$Mw = f(\omega), \quad \text{Yük momenti-açısal hız,}$$

Karakteristiklerine ait matematiksel ifadelerinin bilinmesi ve bu denklemlerde yerine konarak entegrallerinin alınması gerekir. Fakat çoğu zaman bu karakteristikler deneysel yolla bulunduğu ve matematiksel ifadelerin çıkarılması güç olduğundan, bu gibi hallerde hareket problemlerinin çözümü için grafik metotlar kullanılır.

2.3 Elektrikle Tahrikte Kuvvet ve Momentler

Tahrik edilen tezgâh ve makinelerin harekete karşı gösterdiği yük veya yük momentleri faydalı iş ile sürtünme işine isabet eden iki kısımdan oluşur. Faydalı işe isabet eden kısım istenilen teknolojik şekil verme işlemine yarar. Sürtünme işi özellikle tahrik edilen sistemin veriminden hesaplanır. Örneğin kaldırma makinelerinde sürtünme işi denklemlerde ek bir ağırlık olarak hesaba katılabilir. G yükünün h yüksekliğine kaldırılması sırasında sürtünme işinin bu yükü G_o kadar arttırdığı düşünülebilir. Böylece kaldırma esnasında faydalı iş, $A_f = G.h$, sürtünme işiyle birlikte toplam iş, $A_t = (G+G_o).h$ olup kaldırma makinelerinin verimi ve sürtünme işine tekabül eden ağırlığı,

$$G_o = \frac{(1-\eta)}{\eta} G, \quad \eta = \frac{A_f}{A_t} = \frac{G}{G+G_o} \quad (2.6)$$

Şeklinde elde edilir. Tahrik edilen makinelerin sürtünme kuvvet veya momentleri daima cer kuvveti veya döndürme momentine zıt yöndedir. Faydalı yük veya yük momentleri ise genel olarak iki gruba ayrılır:

2.3.1 Reaksiyon Kuvvet veya Momentleri

Elastik olmayan cisimlerin presle kesilmesi, şekil verilmesi, haddeden çekilmesi veya sıvanmasında meydana gelen yük veya yük momentleri ile sürtünme kuvvet ve momentleri bu gruba girer. Bu kuvvet ve momentler daima döndürme momentine zıt yöndedir.

2.3.2 Potansiyel Kuvvet ve Momentleri

Yer çekimi kuvvetlerini yenmeye çalışan tüm kuvvet ve momentler, elastik cisimlere presle şekil verilmesi, presle kesilmesi için gerekli olan kuvvet ve momentler bu gruba girer. Bu tür kuvvet ve momentler işaret değiştirebilir. Sonuç olarak, yük veya yük momentlerinin tipleri ve tahrik motorunun çalışma durumlarına bağlı olarak, elektrikle tahrikin ana denklemlerinin en genel hali,

$$\mp F \pm W = m \frac{dv}{dt}, \quad \mp M_d \pm M_w = J \frac{d\omega}{dt}$$

Şeklinde yazılabilir.

2.4 İşletme Karakteristikleri

Bu tahrik sisteminin gerçekleştirilmesi teknik, ekonomik ve mali etütleri gerektirir. Tahrik sistemi için en uygun motor cinsinin seçimi, motor gücünün tespit edilmesi, motorun teknik özelliklerinin tespiti, motor için lüzumlu hız, yol verme, ölçü ve kumanda sisteminin seçimi teknik etütler arasında yer alır.

Tahrik sistemlerinde en uygun motor ve tahrik sisteminin seçimi için elektrik motorları ile iş makinesi ve takım tezgâhlarının işletme karakteristiklerinin bilinmesi gerekir.

2.4.1 Tahrik Edilen Makinaların İşletme Karakteristikleri

Yapı, çalışma tarzı ve gördükleri işler bakımından birbirinden farklı olan bu tezgâh ve makinelerin işletme özellikleri de farklıdır. Genellikle farklı metotlarla bulunan işletme özellikleri, başlıca faydalı iş ile sürtünme cisme isabet eden iki kısımdan oluşur.

Sürtünmeler dışında faydalı işe isabet eden ideal $M_w=f(\omega)$ Yük momenti - Açısal hız karakteristiklerine göre, tahrik edilen makineleri genel olarak dört teorik gruba ayırarak incelemek mümkündür.

2.4.1.1 Açısal Hız ile Hiperbolik Olarak Değişen Yük Momenti Karakteristiği (Soyma-Moment karakteristiği)

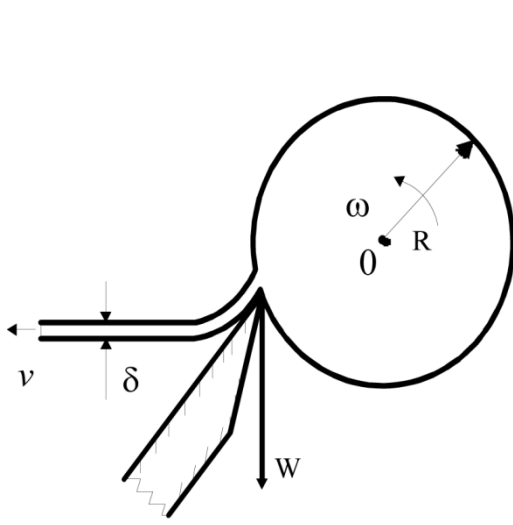
Bu gruba giren makinelerde yük momenti açısal hız ile ters orantılı, yani hiperbolik olarak değişir. Orantı sabiti K ile gösterilirse, yük momenti,

$$M_w = \frac{k}{\omega} \quad (2.7)$$

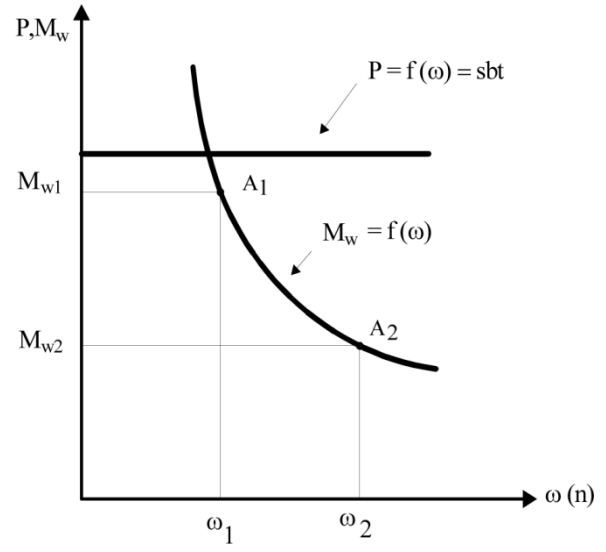
Şeklinde yazılabilir. Her türlü soyma ve sarma makineleri, kâğıt, kumaş ve iplik sarma makineleri ile torna tezgâhları bu gruba giren makineler arasında sayılabilir.

Şekil 2. 3’de, bir kontra plak makinesinin şematik diyagramı verilmiştir. Silindirik ağacın kesit boyunca homojen olduğu kabul edilirse, δ sabit talaş kalınlığı için W deformasyon mukavemeti de sabit olur. Talaş kaldırma hızı ve yük momenti ifadeleri aşağıdaki gibi elde edilir.

$$v = \omega \cdot R, \quad M_w = W \cdot R, \quad M_w = \frac{W \cdot v}{\omega}, \quad K = W \cdot v, \quad M_w = \frac{k}{\omega}$$



Şekil 2. 3 Kontrplak makinesi şematik



Şekil 2. 4 Soyma-Moment

diyagramı

karakteristiği

Şekil 2. 4'te, Soyma-Moment karakteristiğine ait değişimler verilmiştir. Bu karakteristik üzerinde A1 ve A2 gibi iki farklı çalışma noktasına ait yük momenti yazılıp oranlanırsa,

$$\frac{M_{w1}}{M_{w2}} = \frac{\omega_2}{\omega_1} = \frac{n_2}{n_1}$$

İfadesi elde edilir. Bu tip iş makinelerinde güç değişmeyip sabit kalır. Güç ifadesinde moment yerine (2.7)'deki değeri yazılırsa,

$$P = \omega \cdot M = \omega \cdot \frac{K}{\omega} = K = \text{sabit} \quad (2.8)$$

Elde edilir. Kâğıt ve kumaş makinelerinde sıkı ve üniforma bir top elde etmek için, kâğıt veya kumaş makara üzerine sabit bir germe kuvveti ile sarılır. Sarma işi ile soyma işi arasındaki fark, birincisinde zamanla çap büyürken, ikincisinde küçülmesidir. Bu tip makinelerde yarıçap değişirken çevresel hızı sabit tutabilmek için devir sayısı ayarı gereklidir. Düşük devir sayılarında yük aşırı değerlere ulaşır. Bu nedenle ayar sırasında motorun aşırı olarak zorlanmasını önlemek için motorla iş makinesi arasında çok kademeli ayar mekanizması konur ve elektriksel devir sayısı ayarı ile mekanik ayar birlikte yapılır.

2.4.1.2 Aısal Hıza Baęlı Olmayan Yk Momenti Karakteristięi (Kren-Moment Karakteristięi)

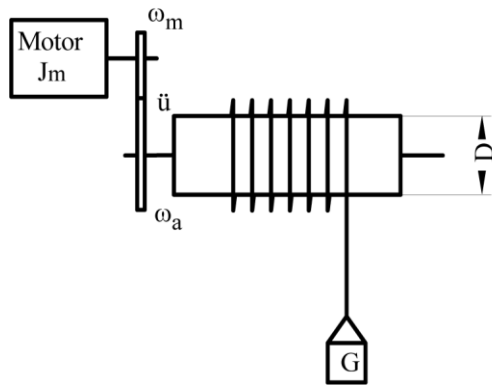
Bu gruba giren makinelerde yk momenti aısal hız veya devir sayısına baęlı olmayıp sabittir.

$$M_w = K = \text{sabit} \quad (2.9)$$

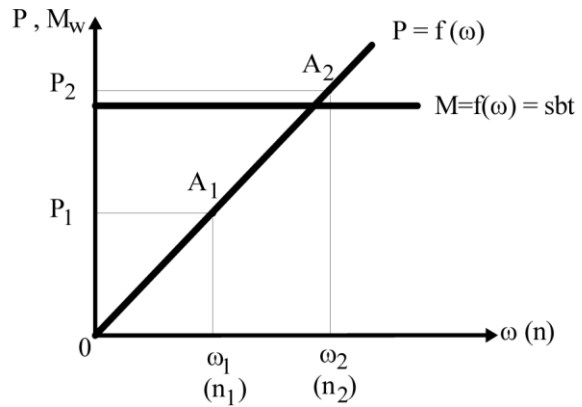
Bu gruba her eřit kaldırma makineleri (vinler, krenler, asansrler), sabit bir ykseklіęe su basan pompalar, transport kayışları, elevatrler vb. makineler girer. ekil 2. 5'de kinematik diyagramı verilen krenin tambur milindeki faydalı yk momenti,

$$M_w = G \frac{D}{2} = K = \text{sabit}$$

dir. ekil 2. 6'da ise bu gruba giren iř makinelerine ait moment ve g deęiřimleri gsterilmiřtir.



ekil 2. 5 Bir krenin kinematik diyagramı



ekil 2. 6 Kren-Moment karakteristięi

G ise,

$$P = \omega \cdot M = K\omega \quad (2.10)$$

eklinde elde edilir. İki farklı alıřma noktası iin gler oranlanırsa,

$$\frac{P_1}{P_2} = \frac{\omega_1}{\omega_2} = \frac{n_1}{n_2}$$

İfadesi elde edilir. Kren-Moment karakteristięine haiz iř makinelerinde, yk momenti iřaret deęiřtirebilir.

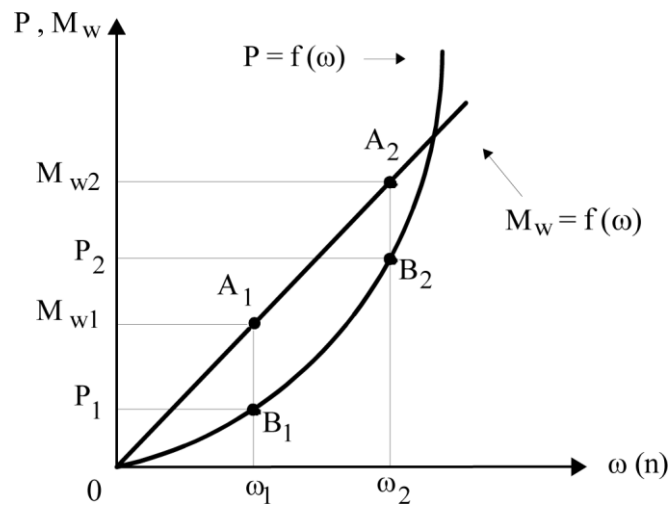
2.4.1.3 Aısal Hız ile Orantılı Deęiřen Yk Momenti Karakteristięi (Kalender-Moment Karakteristięi)

Bu tip makinelerde yk momenti hızla, g ise hızın karesiyle orantılıdır. Orantı sabiti K ile gsterilirse, yk momenti ve g,

$$M_w = K \cdot \omega \quad (2.11)$$

$$P = \omega \cdot M = K \cdot \omega^2 \quad (2.12)$$

řeklinde yazılabilir. Her trl kumař ve kęit perdahlama ve parlatma makineleri (kalenderleri), hadde tezghları bu gruba girer.



řekil 2. 7 Kalender-Moment karakteristięine ait deęiřimler

řekil 2. 7'de Kalender-Moment karakteristięine ait deęiřimler verilmiř olup, iki farklı alıřma noktası iin yk momentleri ve gler oranlanırsa,

$$\frac{M_{w1}}{M_{w2}} = \frac{\omega_1}{\omega_2} = \frac{n_1}{n_2}, \quad \frac{P_1}{P_2} = \left(\frac{\omega_1}{\omega_2}\right)^2 = \left(\frac{n_1}{n_2}\right)^2$$

ifadeleri elde edilir.

2.4.1.4 Aısal Hızın Karesi ile Orantılı Deęiřen Yk Momenti Karakteristięi (Vantilatr-Moment Karakteristięi)

Santrifj etkisiyle alıřan btn makineler; vantilatr, aspiratr ve merkezka pompalar, gemi pervanesi, kimya ve dięer sanayi kollarında kullanılan karıřtırma

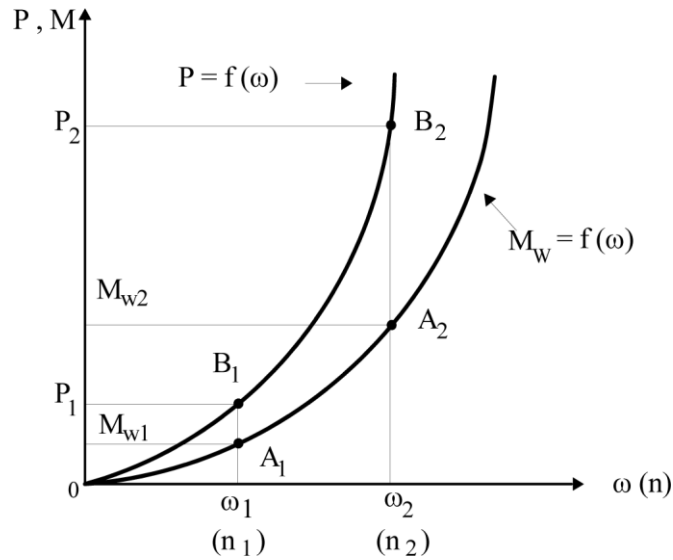
tesisleri bu tip karakteristiğe sahiptir. Bu tip makinelerde yük momenti, açısal hızın karesiyle, güç ise küpü ile orantılı değişir ve aşağıdaki bağıntılar yazılabilir.

$$M_W = K \cdot \omega^2 \quad (2.13)$$

$$P = K \cdot \omega^3 \quad (2.14)$$

Şekil 2. 8'de Vantilatör-Moment karakteristiğine ait iş makinelerinde yük momenti ve gücün açısal hıza bağlı olarak değişimleri verilmiştir. İki farklı çalışma noktası için yük momenti ve güç değerleri yazılarak oranlanırsa, aşağıdaki bağıntılar elde edilir.

$$\frac{M_{W1}}{M_{W2}} = \left(\frac{\omega_1}{\omega_2} \right)^2 = \left(\frac{n_1}{n_2} \right)^2, \quad \frac{P_1}{P_2} = \left(\frac{\omega_1}{\omega_2} \right)^3 = \left(\frac{n_1}{n_2} \right)^3$$



Şekil 2. 8 Vantilatör-Moment karakteristiği

Hava dâhilinde yüksek hızla hareket eden cisimler, örneğin elektrikli ulaşım sistemlerinde hava mukavemetleri de hızın karesiyle orantılı değişir. İş makinelerinin hakiki işletme karakteristikleri çoğu zaman hakiki işe isabet bir veya birkaç teorik karakteristikle sürtünme işine isabet eden diğer bir karakteristiğin toplamından meydana gelebilir.

Makineler dâhilinde sürtünme ve sürtünme kayıpları, hız, sıcaklık derecesi, yağlama malzemesi, sürtünen kısımların pürüzlük derecesi vb. faktörlere bağlı olmakla beraber, tahrik sistemlerinde normal çalışma ve ayar sahaları için değişmeyip sabit kaldığı kabul edilir. Örneğin bir vantilatörün hakiki yük momenti karakteristiği, sürtünme işine isabet

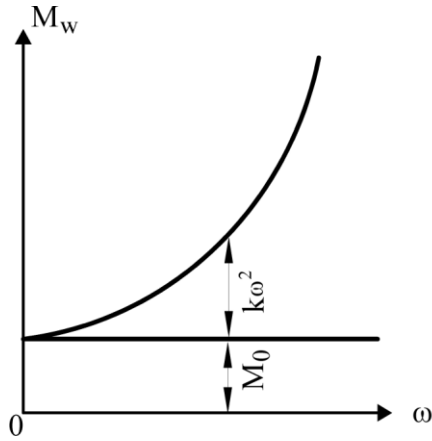
eden sabit bir M_0 srtnme momenti ile teorik vantilatr yk karakteristięinin toplamından oluřur. řekil 2. 9'da bir vantilatrn hakiki yk momenti karakteristięi verilmiř olup, matematiksel ifadesi,

$$M_w = M_0 + K. \omega^2 \quad (2.15)$$

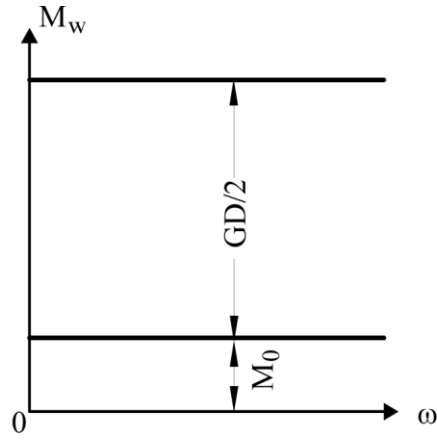
řeklinde yazılabilir. Kren-Moment karakteristięi halinde srtnmeler ek bir G_0 aęırlıęı ile hesaba katılabilir. Bu durumda hakiki Kren-Moment karakteristięinin matematiksel ifadesi,

$$M_w = G_0 \frac{D}{2} + G \frac{D}{2} = (G + G_0)D/2 = \text{sabit} \quad (2.16)$$

řeklinde yazılabilir [4].



řekil 2. 9 Bir Vantilatrn hakiki yk momenti karakteristięi



řekil 2. 10 Bir Krenin hakiki Yk momenti karakteristięi

2.5 Adım Motoru ve Kontrol

Adım motoru, adından da anlařılacaęı gibi adım adım hareket eden yani sargılarından birinin enerji ilenmesi ile sadece 1 adım hareket eden ve elektrik enerjisini dnme hareketine eviren elektro-mekanik bir cihazdır veya bařka bir deęiřle adım motorları, giriřlerine uygulanan darbe dizilerine karřılık analog dnme hareketi yapabilen elektromekanik elemanlardır. Bu adımın ka derece olacaęı motorun tasarımına baęlıdır.

Adım motoru, Elektrik enerjisi alındıęında rotor ve buna baęlı řaft, sabit aısal birimlerde (adım-adım) dnmeye bařlar. Adım motorları, ok yksek hızlı anahtarlama

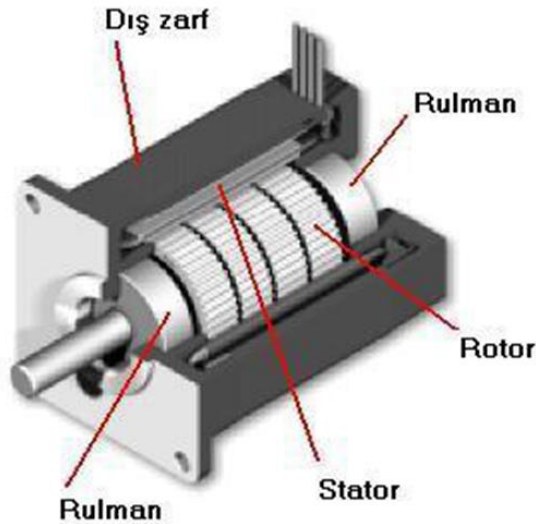
özelliğine sahip bir sürücüye bağlanır ki buna adım motor sürücüsü ismi verilir. Bu sürücü bir kodlayıcı, bilgisayar veya PLC'den giriş darbeleri alır. Alınan her giriş darbesinde motor, bir adım ilerler. Adım motorları, bir turundaki adım sayısı ile anılır.

Örnek olarak 400 adımlık bir adım motoru, bir tam dönüşünde (360°) 400 adım yapar. Bu durumda bir adımın açısı $360/400 = 0,9^\circ$ derecedir. Bu değer, adım motorunun hassasiyetinin bir göstergesidir. Bir devirdeki adım sayısı yükseldikçe adım motoru hassasiyeti ve dolayısı ile maliyeti artar. Şekil 2. 11'de bir adım motoru tipik olarak gösterilmiştir.



Şekil 2. 11 Adım motoru

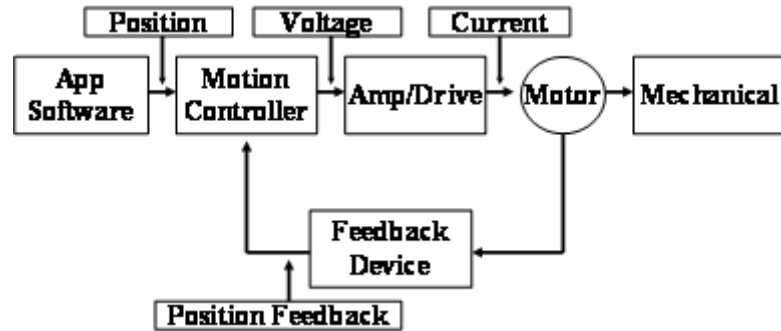
Ayrıca Şekil 2. 12'de adım motorunun kesit görüntüsü verilmiştir[5].



Şekil 2. 12 Adım motorunun kesiti

2.5.1 Bir Hareket Kontrol Sisteminin Bileşenleri

Şekil 2.13 'de bir hareket kontrol sisteminin farklı bileşenleri göstermektedir.



Şekil 2. 13 Bir hareket kontrol sisteminin bileşenleri

Uygulama yazılımı(App. Software): Hedef konumlarını ve hareket kontrol profillerini komuta edebilmek amacıyla uygulama yazılımı kullanılır.

Hareket kontrolörü(Motion Control): Bu bölüm sistemin beyni olarak görev yapmaktadır, yani istenen hedef konumlarını ve hareket profilleri olarak motorların takip etmesi için yörüngeler oluşturur, ancak farklı çıkışlar üretir örneğin servo motorlar için ± 10 V sinyali ve adım motorları için ise adım ve yön darbe dizini üretir.

Amplifikatörler veya sürücüler(Amplifiers): kontrolörden komutları alıp ve motoru sürmek veya çevirmek için gerekli akımı üretir.

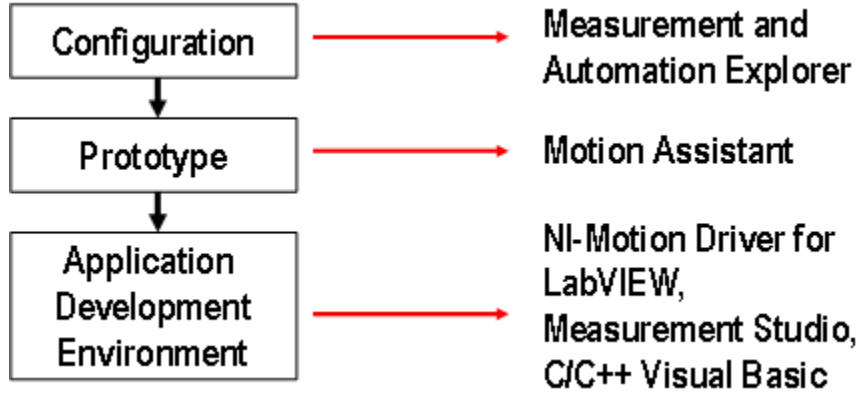
Motor: Motorlar mekanik enerjiyi elektrik enerjisine çevirmek ve istenilen hedef konuma taşınmak için gerekli momenti üretirler.

Mekanik Elemanları(Mechanical elements): Motorlar mekanik moment sağlamak üzere tasarlanmıştır. Bunlar lineer slaytlar ve robotik kollar olabilirler.

Geri besleme Cihazı veya Pozisyon Sensörlü(Feedback device or position sensor): bir pozisyon geri besleme cihazı bazı hareket kontrol uygulamaları için gerekli olmayabilir(adım motorları kontrolü gibi) ancak servo motorlar için hayati öneme sahiptir. Geri besleme cihazı, genellikle döndürme kodlayıcı olmak üzere, motor pozisyonunu algılar ve kontrolöre bilgi verir. Böylece döngüyü hareket kontrolü için kapatmış olur.

2.5.2 Yapılandırma, Prototip ve Geliştirme İçin Yazılım

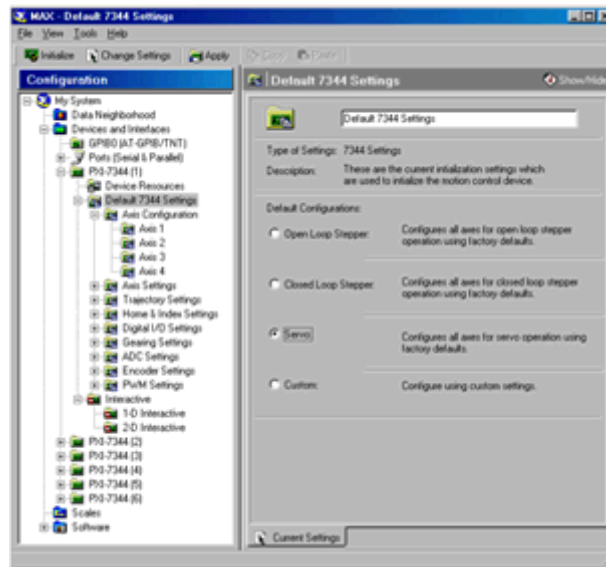
Uygulama yazılımı üç ana kategoriye ayrılır: Yapılandırma, ilk örnek ve uygulama geliştirme ortamı(ADE). Şekil 2.14'de hareket kontrol sistemi işlemi için tasarlanmış programlama süreci gösterilmiştir.



Şekil 2. 14 Hareket kontrols geliştirme süreci

Yapılandırma

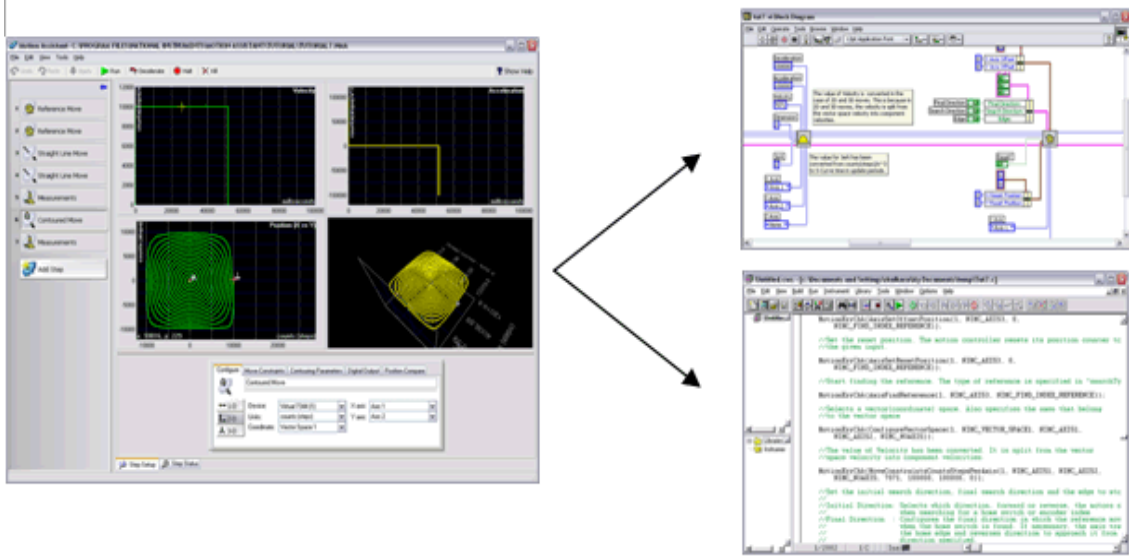
yapılacak ilk şeylerden biri, sistemimizi yapılandırmaktır. Bunun için, National Instruments, Ölçüm ve Otomasyon Explorer(MAX) 'i sunuyor. Burada hareket kontrolü ile birlikte diğer tüm National Instruments donanımları da yapılandırılabilir. Hareket kontrolü için, MAX sistemi programlamadan önce sistem işlevselliğini doğrulamada yardımcı olmak üzere bir interaktif test ve ayar paneli sunuyor.



Şekil 2. 15 NI MAX hareket kontrol sisteminin ekran görüntüsü

Prototip

Eğer sisteminizi kon figüre ettiyseniz, ilk örnek ve uygulama geliştirmeye başlayabilirsiniz. Bu aşamada, hareket kontrol profillerini oluşturup ve sistem üzerinde deneyerek, istediğiniz amaca hizmet edeceğinden emin olabilirsiniz. Bu amaçla National Instruments NI Motion Yardımcısı diye bir sitem geliştirmiştir, diğer taraftan, size görevleri yerine getirmek için LabVIEW, C veya Visual Basic gibi standart programlama dillerini kullanmak için bir köprü oluşturmaktadır. NI Motion Yardımcısı uygulamalarınızı prototipini hızlıca oluşturmak ve sonra bunlardan LabVIEW, C veya Visual Basic gibi standart programlama dillerinin formatında çıkış almak olanağı sunmaktadır.



Şekil 2. 16 NI Motion yardımcısı

Geliştirme

Prototip aşamasından sonra, bir sonraki adım son uygulama kodunu geliştirmektir. Bunun için, LabVIEW, C veya Visual Basic gibi bir ADE sürücü düzeyinde yazılımı kullanmak gerekiyor, örneğin National Instruments hareket denetleyicisi için, NI-Motion sürücü yazılımını kullanılabilir.

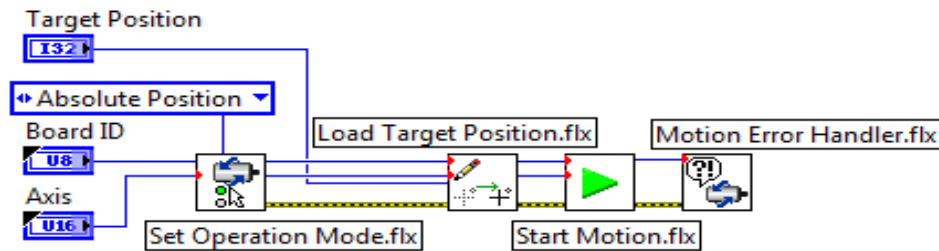
NI-Motion sürücü yazılımı Windows veya LabVIEW gerçek zamanlı OS NI hareket kontrolörleri ile iletişim kurmak için kullanabileceğiniz işlevleri içerir. NI-Motion da kolayca hareket sistemi yapılandırma ve ayar yardımcı MAX uygulaması mevcuttur. Windows olmayan sistemler için, NI hareket kontrol donanım DDK kılavuzunu kullanarak kendi sürücünüzü geliştirebilirsiniz.

2.5.3 Hareket Tipleri

Aşağıda hareket tiplerine göre NI Instrument firması tarafından uygulanan yazılım ve donanım örnekleri verilmiştir.

2.5.3.1 Tek-Eksen, Noktadan Noktaya Hareket(Single-Axis, Point-to-Point Motion)

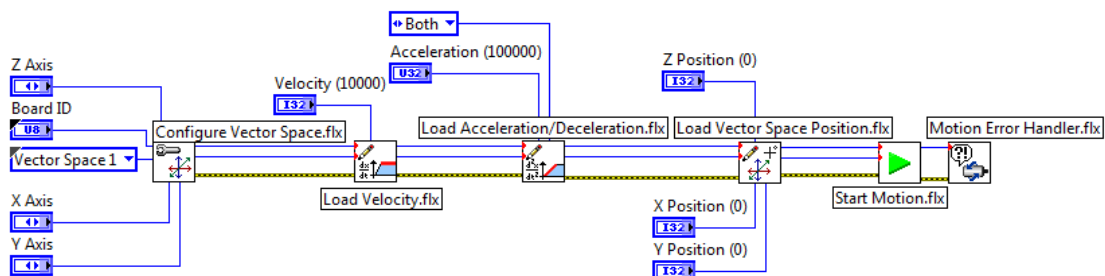
En sık kullanılan profillerden biri basit, tek eksenli, noktadan noktaya bir harekettir ki bunun için eksenin taşınması gereken konumun bilinmesi gerekir. Genellikle istenilen harekette aynı zamanda hız ve ivmenin bilinmesini gerektirir(genellikle varsayılan ayarı ile sağlanan). Şekil 2.17 'de LabVIEW'da belirli bir hız ve ivme kullanılarak, tek eksenli bir taşımının nasıl olduğu gösterilmiştir.



Şekil 2. 17 LabVIEW'da tek-eksen, noktadan noktaya hareket

2.5.3.2 Koordineli Çoklu Eksen Hareketi(Coordinated Multiaxis Motion)

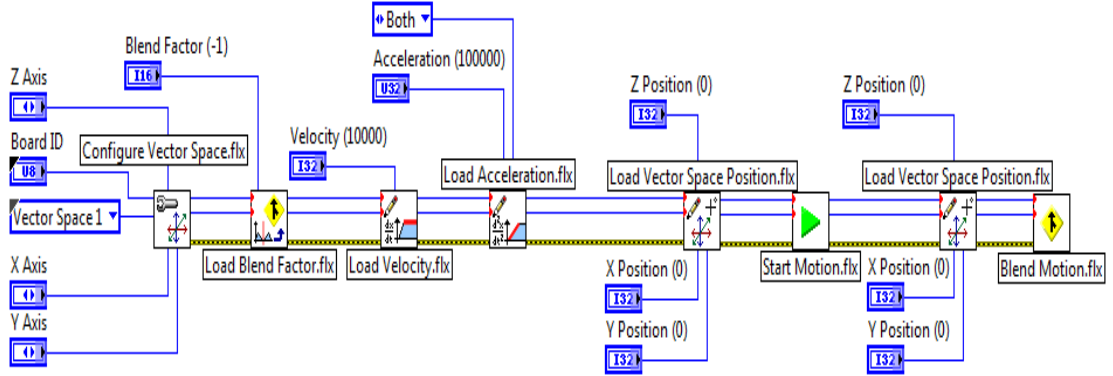
Bir başka hareket türü koordineli çoklu eksenli hareket veya vektör hareketidir. Bu hareket genellikle noktadan noktaya hareket ancak 2D veya 3D boyutlarda gerçekleşir. Vektör harekette X, Y ve / veya Z eksenlerinde nihai pozisyonların bilinmesi gerekir. Şekil 2.18 'de LabVIEW kullanarak üç eksenli hareket gerçekleştirilmiştir.



Şekil 2. 18 LabVIEW'da koordineli çoklu eksenli hareket

2.5.3.3 Karma Hareket

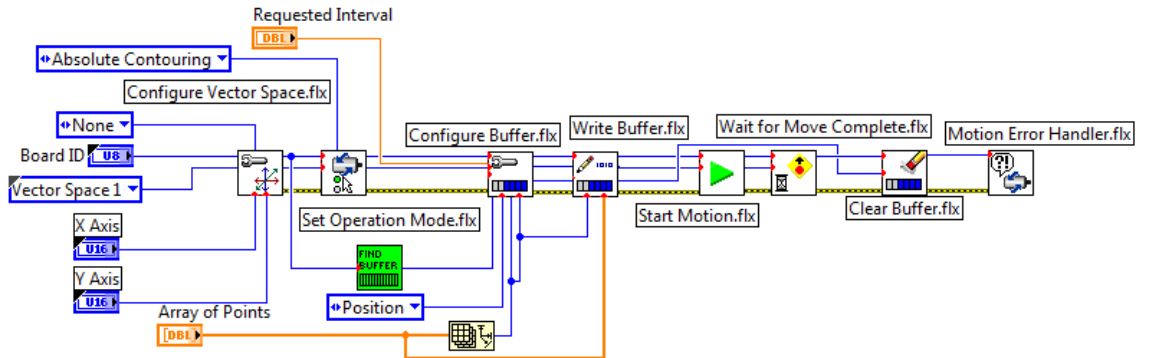
Şekil 2.19 'da NI-Motion sürücü yazılım örneği olarak LabVIEW ile iki vektör arasında karma hareketini göstermektedir.



Şekil 2. 19 LabVIEW'da karma hareket

2.5.3.4 Kontrollü Hareket

Şekil 2. 20'de NI-Motion sürücü yazılım örneği olarak LabVIEW ile bir kontrollü hareket gösterilmiştir [6].



Şekil 2. 20 LabVIEW'da kontrollü hareket

Aşağıda NI Instruments firması tarafından farklı hareket tipleri için programlana bilen donanım örnekleri verilmiştir.



Şekil 2. 21 NI Instruments yapımı sürücü devresi ve adım motorları örneği

Bu sistemde kullanılan adım motorları ve sürücü kodlayıcı sınıfı(encoder) özellikleri aşağıdaki gibidir [7].

Örnekte kullanılan adım motorları

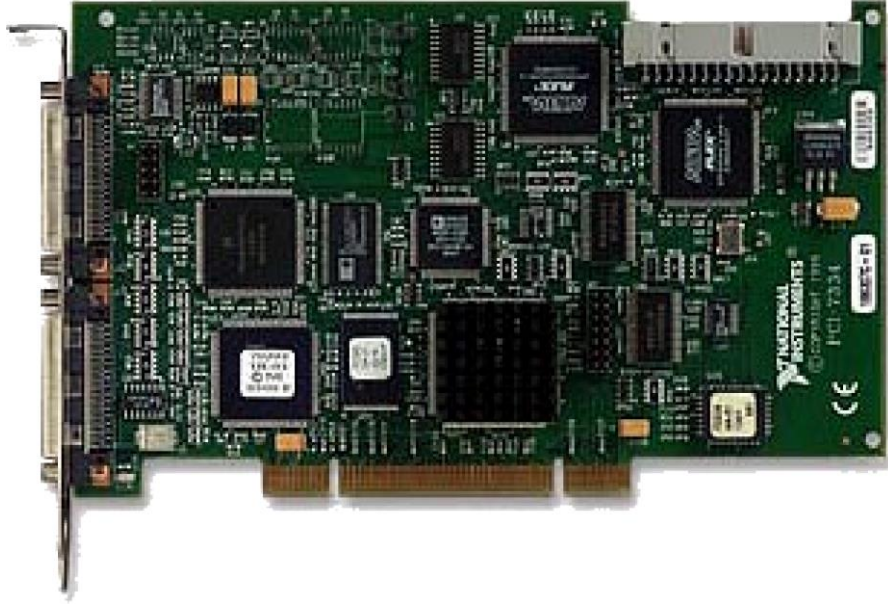
- NEMA 17, 23 ve 34 şasi boyutları
- 1710 oz-in.'e kadar ($12,1 \text{ N} \cdot \text{m}$) tutma momenti
- 3000 rpm maksimum hız
- 1.8 derece adım açısı
- Yüksek performans için P7000 sürücüler ile uyumlu

Örnekte kullanılan kodlayıcı

- 1000 devir/dakika çözünürlük
- NEMA 23 ve 34 motor uyumluluğu
- Düşük profil 1 in. (25.4 mm) kaliteli tasarımı ve kolay montaj
- Endüstriyel ortamlarda kullanılma yeteneği

4 eksenli NI PCI-7334 sürücü devresi

Bu örnekte NI Instruments tarafından üretilen 4 eksenli bir adım motor sürücüsü devresi gösterilmiştir.



Şekil 2. 22 NI Instruments yapımı 4 eksenli sürücü devresi

Devrenin özellikleri aşağıdaki gibidir [8].

- 4-eksen step motor kontrol panosu
- Çeyrek encoder veya analog geri besleme
- Diğer NI ölçüm ürünleri ile güçlü senkronizasyon için RTSI Bus
- Koordineli, çok eksenli hareket kontrolü için 3D doğrusal ekleme yapma
- ± 10 V serisi ile dört adet 12-bit ADC

LAM Teknolojies Tarafından Üretilen Sürücüler

LAM Teknolojies firması tarafından üretilen adım motoru sürücülerinde, geleneksel adım ve yön sinyalleri kullanılarak geniş çapta çift kutuplu adım motoru kontrolü yapılmıştır.

Bu sürücülerde olağanüstü geniş güç aralığı ile 18VDC 'dan 240VDC'ye kadar gerilim ve 0,3Arms'den 10Arms'e kadar akım aralılarında kullanılabilirler. Bu sürücülerin çoğunda AC trafo kullanıldığı için doğrudan AC güç kaynağına da bağlana bilirler. Aşağıdaki resimde bu ürünlerin bazıları gösterilmiştir.



Şekil 2. 23 Adım ve yön değıştirici mikro sürücüler

Bu firmanın DS30 serisinde ürettiğı sürücülerden örnek olarak DS3041 aşağıda incelenmiştir. Adım motorlarının konum ve hız kontrolü bu sürücülerini kullanarak çok dikkatli yapıla bilir. Ayrıca yapı itibariyle kullanıcı tarafından kolaylıkla programlanabilirler.

DS3041 bir programlanabilir çift kutuplu adım motoru sürücüsüdür ve 50Vdc, 1.4Arms(2Apk) olarak çıkış özelliklerine sahiptir [9].



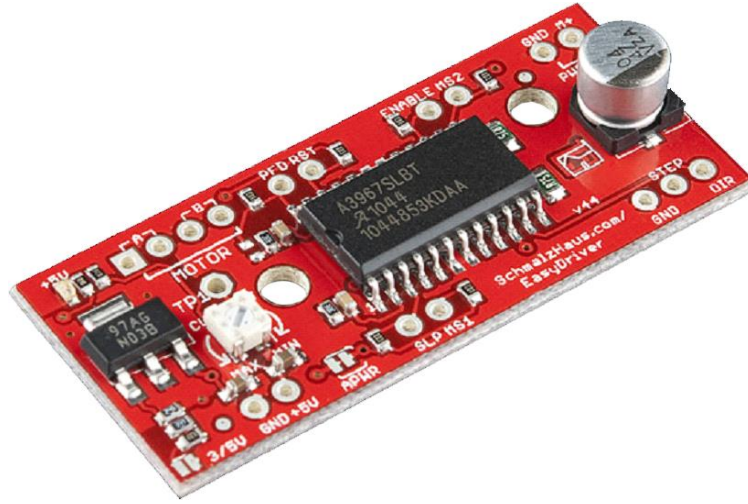
Şekil 2. 24 DS3041 adım motoru sürücüsü

Sürücünün teknik bilgileri bu şekildedir;

- Besleme gerilimi 18 – 50 Vdc
- Faz akım değeri 0,3 – 1,4 Arms
- UDP Commander programı üzerinden kullanıcı tarafından programlanabilir
- 3000 devir dakika(rpm), 1/128 çözünürlüklü adım motoru sürme performansı

SparkFun Electronics Tarafından Üretilen Sürücüler

SparkFun Electronics firması tarafından üretilen ve programlanabilir çift kutuplu adım motoru sürücüsü olan ROB-09402, genellikle robot endüstrisinde kullanılır [10].



Şekil 2. 25 ROB-09402 adım motoru sürücüsü

- Sürücüde kullanılan mikrodenetleyici A3967SLB
- Besleme gerilimi 7 – 30 Vdc
- Faz akım değeri 0,15 – 0,75 Arms
- 4, 6 ve 8 kablolu adım motorlarıyla uyumlu

2.6 Adım Motorların Kullanım Alanları

Adım motorları, girişlerine uygulanan darbe dizilerine karşılık (bu sayısal giriş olarak da ifade edilebilir), analog dönme hareketi yapabilen elektromanyetik elemanlardır. Bu özellikleri nedeniyle “sayısal makine” olarak da tanınan adım motorları, sayısal sistemlerde kullanılırken büyük kolaylıklar sağlarlar.

Adım motorları, adından da anlaşılacağı gibi belirli adımlarla hareket ederek rotorun açısal konumunu değiştirirler. Bu adımlar, motor sargılarına uygun sinyaller gönderilerek kontrol edilir. Herhangi bir uyarımda, rotorun yapacağı hareketin ne kadar olacağı, motorun adım açısına bağlıdır. Adım açısı, motorun yapısına bağlı olarak 90, 45, 18, 7,5, 1,8, ... Derece veya çok daha değişik açılarda olabilir. Motora uygulanacak sinyallerin frekansı değiştirilerek motorun hızı da kontrol edilebilir. Adım motorlarının dönüş yönü ise, uygulanan sinyallerin sırası değiştirilerek, saat ibresi yönünde (CW) veya saat ibresinin tersi yönünde (CCW) olabilir.

Adım motorları uzun yıllardır var olmalarına rağmen ticari olarak kullanılmaları ancak 1960'lı yıllarda yüksek seviyeli doğru akımları anahtar ayabilen transistörlerin üretimine başlanmasıyla yaygınlaşmıştır. 1970'li yıllardan beri dijital elektronikteki ve mikrodenetleyici teknolojisindeki gelişmelerle birlikte adım motorlarının kullanımı giderek cazipleşmekte ve tüm dünyada bu motorların üretim ve uygulamalarıyla ilgili geliştirme çalışmaları yapılmaktadır.

Günümüzde adım motorları endüstride birçok kontrol sistemlerinde, hassas konum kontrolü yapmak amacıyla kullanılmaktadır. En çok yazıcılar (printer), çiziciler (plotter), disket sürücüler (floppy driver), hard disk sürücüler (hard disk driver), kart okuyucular... Vb. gibi bilgisayar çevre cihazlarında bu elemanlardan yararlanılmaktadır. Ayrıca sayısal kontrol sistemlerinde, CNC tezgâhlarında, süreç (process) kontrol sistemlerinde, robot teknolojisinde (milimetrik hareketlerin kontrolünde) ve uzay endüstrisine ait birçok sistemde adım motorları tahrik elemanı olarak yer almaktadır.

2.6.1 Adım Motorların Avantajları

Adım motorlarının bu kadar çok kullanım alanı bulmasının nedeni bu motorların bazı avantajlara sahip olmasıdır. Bu avantajlar aşağıdaki gibi sıralanabilir.

- Adım motorları sayısal giriş işaretlerine cevap verirler, bu nedenle mikrodenetleyici veya bilgisayarlarla kontrol için ideal elemanlardır.
- Adım motorların hangi yöne döneceği, devir sayısı, dönüş hızı gibi değerlerin mikrodenetleyici veya bilgisayar yardımı ile kontrol edilebilmesi, her an bu

motorların dönüş yönü, hızı ve konumunun bilinebilmesini sağlamaktadır. Bu özelliklerinden dolayı adım motorları ile çok hassas konum kontrolü yapılabilir.

- Adım motorların sayısal girişlere cevap vermesi, geri beslemeye ihtiyaç duyulmaksızın açık çevrim çalıştırılabilmesini sağlamaktadır. Yani açık çevrim çalıştırılan bir adım motoru ile hız, ivme ve konum kontrolü daha basit ve daha az maliyetle gerçekleştirilebilir. Böylece alışılmış kararsızlık problemlerinin de önüne geçilmiş olur.
- Adım motorları, giriş işaretlerinin frekansına bağlı olarak çok geniş bir hız aralığında sürülebilirler.
- Adım motorları, herhangi bir hasara yol açmadan defalarca durdurulup çalıştırılabilirler. (Sürerken aniden durdurma ya da aniden ters yönde sürme isteğine karşı mükemmel cevap verebilirler)
- Adım motorları, aşırı yüklenmeden hasar görmezler, oldukça dayanıklıdırlar.
- Adım motorların, her yeni adımla artan (katlanmış) konum hataları yoktur.
- Adım motorları, mekanik yapısı basit olduğundan bakım gerektirmezler.

Yağlanma ve kirlenme problemleri yoktur.

2.6.2 Adım Motorların Dezavantajları

Adım motorların bütün bu avantajlarına karşılık bazı dezavantajları vardır ki aşağıdaki gibi sıralanabilir.

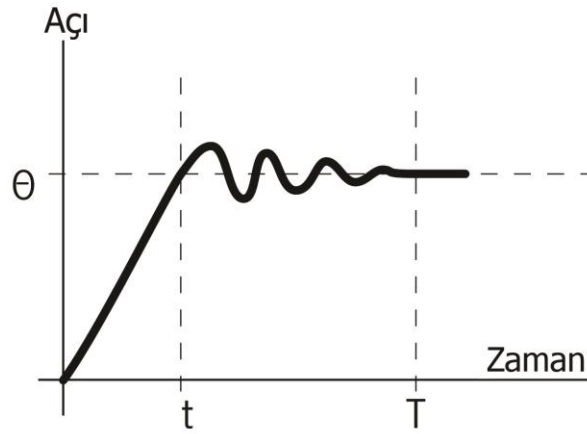
- Adım açıları sabit olduğundan rotordan alınan hareket sürekli değil darbelidir.
- Klasik sürücülerle kullanıldıklarında verimleri düşüktür.
- Adım cevapları nispeten büyük aşım ve salınımlıdır.
- Yüksek eylemsizlikle yüklerde yetenekleri sınırlıdır.
- Sürtünme kaynaklı yükler, hata katlanmış olmasa dahi açık çevrim çalışmada konum hatası meydana getirebilirler.
- Elde edilebilecek çıkış gücü ve momentleri sınırlıdır.

- İyi kontrol edilmezse tınlışım meydana gelebilir.
- Oldukça yüksek hızlarda çalıştırmak pek kolay değildir [11].

2.7 Adım Motorlarına Ait Bazı Önemli Kavramlar

Adım motorlarına ait bazı önemli kavramlar bu bölümde anlatılmıştır.

2.7.1 Adım Tepkisi / Tek Adım Tepkisi (Single Step Response)/ Cevap Süresi



Şekil 2. 26 Adım motorlarının adım tepkisi

Motor fazlarından biri uyarılmış durumdaysa, motor kararlı bir adım konumundadır. Bu fazın uyarıtımı kesilip yeni bir faz uyarılırsa rotor bir adım atacaktır. Rotor konumunun zamana göre değişimi “tek adım tepkisi” olarak tanımlanır. Bir başka deyişle motorun girişine bir komut darbesi (sinyali) uygulandıktan sonra motorun adımlara cevap vermek için gerekli olarak aldığı zamana (T) “tek adım tepkisi”, “adım tepkisi” ya da “cevap süresi” denir. Bu süre hem motor parametrelerine hem de motorun sürücü devresine bağlıdır ve yaklaşık olarak milisaniye civarlarındadır.

Tek adım tepkisi, motorun adım hareketinin hızını, tepkinin aşım ve salınım miktarını, adım açısının hassaslığını veren önemli bir parametredir. Adım motorlarından maksimum performans elde edebilmek için tek adım tepkisindeki aşım ve salınımların azaltılması ve yerleşme zamanının kısaltılması gerekmektedir. Bu nedenle tek adım tepkisinin iyileştirilmesi adım motorlarının kontrolünde çok büyük öneme sahiptir.

Motora uygulanan giriş darbeleri ile çıkış hareketi arasındaki senkronizasyonu bozmamak için, sürme esnasında iki darbe arasındaki süre, cevap süresinden daha kısa

olmamalıdır. Dolayısıyla adım motorunun cevap süresinin kısaltılabilmesi, motorun daha hızlı adım atabilmesini sağlayacaktır.

Adım motorlarının cevap davranışlarında dikkat edilmesi gereken diğer bir nokta da yaptıkları aşımın(overshoot) ve osilasyonun miktarıdır. Adım motorları, bilgisayar sistemlerinde veya bilgisayarla kontrolü gerektiren hassas sistemlerde kullanıldığında, aşım ve osilasyon sistemi kötü yönde etkileyen ve önemli boyutta hatalara sebep olan bir davranıştır.

2.7.2 Adım Oranı (Stepping Rate / Starting Pulse Rate)

Bir saniyede rotorun yapabildiği adım sayısıdır. Bu adım sayıları, tipik olarak saniyede 300 ila 800 arasındadır.

2.7.3 Adım Açısı (Step Angle)

Motora bir tek darbe uygulandığında rotorun döndüğü açıdır. Yapımcı firmalar, değişen sayılarda rotor ve stator sargıları ile çeşitli açılarda adım motorları yapmaktadırlar.

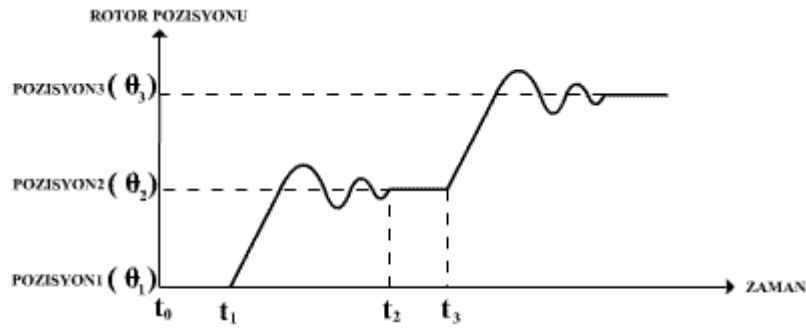
2.7.4 Adım Açısı Doğruluğu (Step Angle Accuracy)

Rotorun yaptığı her bir adımdaki hata miktarını gösterir. Bu parametre genellikle bir yüzde ile verilir. Rotor bir adım döndüğünde meydana gelen toplam hata oranını gösteren bu parametrenin yazılması önemlidir. Bu hata değeri birikmiş(birikimli) değildir. Yani rotorun yaptığı her adım ile bu hata miktarı toplanarak gitmez. Rotorun gerçek hızı adım açısı ve adım hızına bağlıdır.

2.7.5 Çözünürlük

Çözünürlük, dönen motorlar için adım açısı(derece), lineer motorlar için ise adım uzunluğu(mm) olarak tanımlanır. Bu sabit değer, üretim sırasında tespit edilen bir büyüklüktür. Bir adım motorunun adım büyüklüğü, çeşitli uyartım metotları ile değiştirilebilir. Mesela yarım adım çalışmada, adım büyüklüğü normal değerinin yarısına indirilir.

2.7.6 Hedefi Aşırmak (Overshoot)



Şekil 2. 27 Adım motorun rotor pozisyonu-zaman eğrisi

Adım motora gönderilen darbelerin frekansı arttırılırsa, adım motorun adımları hızlanır ancak adımlarının hızlanmasıyla adım motor istenmeyen bir karakteristik gösterir. Buna “overshoot” denir. Bu durum yukarıdaki şekilde görülmektedir. Motorun rotoru “Pozisyon1” ve “t₀” anında durmaktadır. “t₁” anında stator sargısına uygun güç uygulanmıştır. Bu uygulama rotorun “Pozisyon2” ye dönmesini sağlar. Rotor “Pozisyon2” de tam olarak durmaz çünkü bir dönme momenti vardır. Bundan dolayı dönmek isteyecektir. Rotor biraz daha döner. Buna “overshoot” denir. İki kutup arasındaki manyetik ilişkiden dolayı rotor tekrar geri döner ama yine bir miktar “overshoot” meydana gelir. Rotorun hareketi kesin ve net olmayacağından tam durması istenen yerde duramayacaktır. Kutupların manyetik etkisinden dolayı geri gelip yerinde durması gerekirken fazla dönerek tekrar istenmeyen bir durum meydana gelir. Yani rotor basit harmoni hareket ya da sönümlü osilasyon yapar. Neticede motora darbe uygulandıktan sonra rotorun durması bir miktar zaman alır. Rotorun durması için geçen zamana “oturma(yerleşme) zamanı (settling-time)” denir. Bazı devrelerde rotorun durmasının çabuk olması istenir. Bunun yapılabilmesi için bazı işlemlere ihtiyaç vardır. Bu işleme “sönümlleme(damping)” denir [11] - [15].

2.7.7 Adım Motor Moment Parametreleri

Dönme momentine 'moment' denir. Moment, elektrik devrelerindeki elektrik potansiyeline, milin açısal hızı da elektrik akımına benzetilebilir. Nasıl elektrik devrelerindeki direnç potansiyel etkisinde akımı sınırlarsa, mekanik devrelerde de sürtünme kuvvetleri milin moment etkisinde açısal hızına sınır getirir. Bir elektrik devresinde akü, potansiyel kaynağıdır aynı şekilde mekanik sistemlerde de motor,

moment kaynağıdır. Motor mili, moment etkisinde dönmeye zorlanır. Bir aküden çekilen akımı sıfırdan itibaren arttırsak, akünün iç direncinden dolayı akım arttıkça akü voltajı düşer. Benzer olarak bir motorun milinden alınan devir arttıkça motorun verdiği moment düşer.

Elektrik devrelerinde akım ve gerilimin çarpımı, düzgün harekette ise bir nesneyi etkileyen kuvvet ile nesnenin hızının çarpımı gücü verirken, motorda moment ile açısal hızın çarpımı mekanik gücü verir.

$$P = I * V \text{ (Güç = Akım * Gerilim)} \quad (1.1)$$

$$P = F * S \text{ (Güç = Kuvvet * Hız)} \quad (1.2)$$

$$P = Tm * \omega n \text{ (Mekanik Güç = Moment * Açısal Hız)} \quad (1.3)$$

$$Tm = F * r \text{ (Dönme Momenti [Moment] = Kuvvet * Yol)} \quad (1.4)$$

Bir motordan, herhangi bir devirde beklenen momentin büyüklüğü, motorun gücü ile orantılıdır. Sabit güç altında itme kuvveti artarken hız düşer, hız artarken ise itme kuvveti düşer ve dolayısıyla moment da düşer.

Adım motor kataloglarında dört çeşit moment parametresine rastlanılır. Bu parametreler genellikle moment-hız grafikleriyle birlikte kataloglarda sunulur.

a. Yüksüz ve Enerjisiz Tutma Momenti (Detent Torque)

b. Yüksüz ve Enerjili Tutma Momenti (Holding Torque)

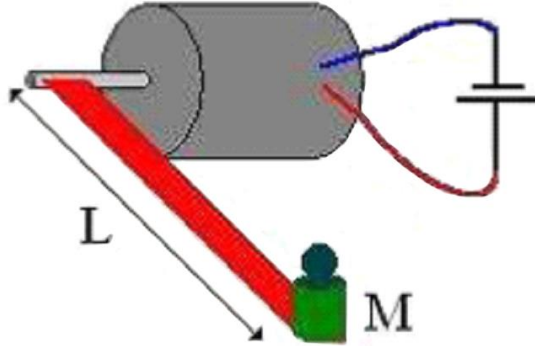
c. Kalkıştaki Maksimum Yük Momenti (Pull in Torque)

d. Sürekli Rejimdeki Maksimum Yük Momenti (Pull Out Torque)

a. Yüksüz ve Enerjisiz Tutma Momenti (Detent Torque) : Detent Torque, motor enerjisiz iken rotor stator dişlerindeki etkileşimden dolayı oluşan moment demektir. Diğer bir ifadeyle sargılara besleme gerilimi uygulanmıyorken, adım motor şaftını döndürmek için gerekli olan maksimum moment miktarı demektir. Bu parametre adım motor tipine göre değişir.

b. Yüksüz ve Enerjili Tutma Momenti (Holding Torque) : Holding Torque, “tutma moment” ya da “statik moment” olarak da bilinir. Sargılardan sadece birisinden nominal değerde akım akıtılıyorken milden alınan moment demektir. Diğer bir ifadeyle

enerjili halde motor duruyorken üretilen maksimum moment miktarıdır. Tutma momenti bir adım motorunun en temel moment karakteristiğidir ve pratik olarak şu şekilde ölçülebilir.



Şekil 2. 28 Motor mil bağlantısı

Motor miline şekilde görüldüğü gibi ağırlıksız bir çubuk bağlanır ve çubuğun yatayla sıfır derece açı yapması sağlanır. Motor sargısından nominal değerde akım akıtılır. Şekildeki yeşil ağırlık çubuğun en ucuna oturtulur ve değişik ağırlıktaki kütlelerle denenerek hangi ağırlıktan sonra motor mili sapma yapıyor tespit edilir. Çubuğun boyu L ve Kütlenin ağırlığı M olması durumunda;

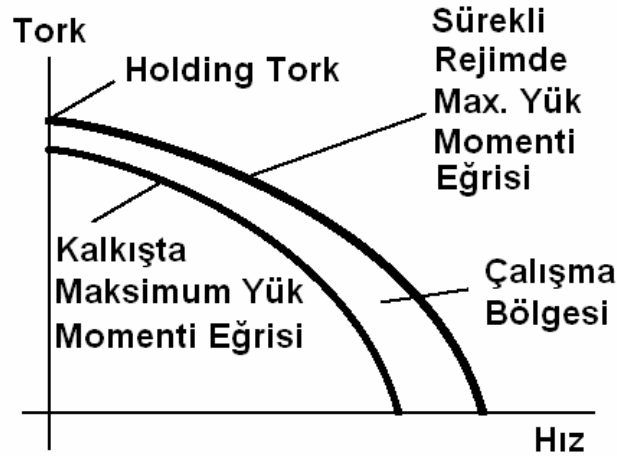
$$\text{Tutma Momenti} = (N.m) = L (m) \times M (Kg) \times 9,81 (\text{Yerçekimi İvmesi}) (N/Kg) \quad (1.5)$$

Örneğin Enerjili motorda, 10cm lik çubuk ucuna 250gr lık ağırlık konduğunda mil aşağı ya da yukarıya kaymıyorsa oluşan moment $T = 0,1 \times 0,25 \times 9,81 = 0,245 \text{ N.m}$ dir. Sonucun Amerikan sistemindeki karşılığı: $1\text{N.m}=141,6 \text{ ozf inc}$ ise $T=34,7 \text{ ozf inc}$ (onz forse inc) olur.

c. Kalkıştaki Maksimum Yük Momenti (Pull in Torque) Özellikle açık döngülü sistemlerde duran bir sistemi istenen pozisyona getirebilmek için motora uygulanan uyartım darbelerinin motor tarafından hiç kaçırılmadan takip edilmesini sağlamak çok önemlidir. Fakat uygulanan uyartım sinyallerin sıklığı, motorun miline bağlı yükü sıfır hızından itibaren kaldırıp hızlandırmasına izin vermeyebilir. Bu yüzden adım motorları için, kalkışta maksimum yük momenti eğrileri tanımlanır.

d. Sürekli Rejimdeki Maksimum Yük Momenti (Pull Out Torque) Bu parametre de motor hızına bağlı olarak değişeceğinden sürekli rejimde maksimum yük momenti eğrisinden bahsedilir.

2.7.8 Moment - Hız Grafikleri



Şekil 2. 29 Moment – Hız grafiği

Sürekli rejimde maksimum yük momenti/ hız eğrisi herhangi bir sabit dönüş hızında, rotor hareketinin giriş darbe dizisiyle olan senkronizasyonunu bozmadan ve rotorun durmasına neden olmadan sürekli halde motor miline uygulanabilecek maksimum yük momentini verir. Bu moment aynı zamanda, söz konusu hızda motorda meydana gelecek maksimum moment anlamına da gelmektedir.

Klasik motorlarda bu eğriye karşılık gelebilecek bir karakteristik yoktur. Maksimum yük momenti eğrisi çalışma noktalarını göstermediği gibi bir transfer fonksiyonu eğrisi de değildir. Sadece, çalışma bölgesini sınırlar. Bu eğrinin sınırladığı bölge içinde herhangi bir noktada motor giriş darbe dizilerini kaybetmeden ve durma tehlikesi olmadan ilgili hız ve yük momenti ile çalışır. Sınırların dışına çıkıldığında bu durum değişebilir.

2.7.9 Etiket Voltajının Anlamı

Etikette belirtilen voltaj rotor hareketsizken tutma momentinin oluşturulması için gerekli voltaj değeridir. Bu voltaj değeri etiket akımına bölünürse ilgilenilen fazın sıcak sargı direnci bulunmuş olur. Etiket akımının üstüne çıkılmaması gerekir [16].

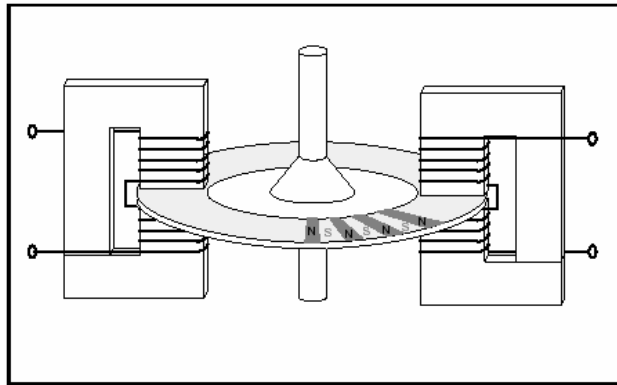
2.8 Adım Motor Çeşitleri

Kullanımda olan birçok elektrik motorunda olduğu gibi adım motorları da makinenin yapısına ve çalışmasına göre sınıflandırılabilir. Adım motorları için ilk günlerden bu yana

birçok deęişik tasarım yapılmıştır. Bu motorların kayda deęer bir moment üretebilmeleri için hem rotor hem de statorları önemli miktarda manyetik akı taşıyabilecek yapıda olmalıdır. Bu nedenle de olabildiğince çok sayıda demir diş ile çevrili olmaları gerekmektedir. Günümüzde hem bu özellięi taşıyıp, hem de ticari bakımdan kullanılabilecek nitelikteki adım motorların başlı çaları Deęişken Relüktanslı, Sabit Mıknatıslı ve Hibrid adım motorlarıdır. Temelde bunlar aynı türden cihazlardır, ama farklı yapım yöntemlerinden kaynaklanan farklı karakteristikleri belli bir uygulama için adım motor seçilirken önem kazanır.

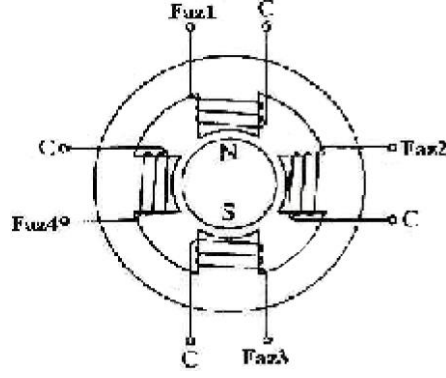
Dięer tür adım motorları, bu üç tip adım motorların varyasyonları olarak tanımlanabilirler. Adım motorların en belirgin özellięi, girişine uygulanan uyarma darbelerinin, “adım” olarak tam tanımlanmış rotor konumlarına dönüştürmesidir. Rotorun hassas konumlama yapması, hareketli ve sabit parçalar üzerindeki demir dişlerin manyetik olarak karşılıklı gelmesi, dizilmesi ile mümkündür.

2.8.1 Sabit Mıknatıslı (Permanent Magnet, PM) Adım Motorları



Şekil 2. 30 Sabit mıknatıslı adım motoru

Sabit mıknatıslı adım motorlarındaki rotor, sabit mıknatıstan oluşur. Bu tip motorların rotorlarında diş bulunmaz. Yukarıdaki şekilde de görüldüğü gibi iki farklı yapıda (“disk” ya da “tin-can” yapısı) rotor yapıları mevcuttur. Stator yapıları, deęişken relüktanslı adım motorların stator yapılarıyla aynıdır.



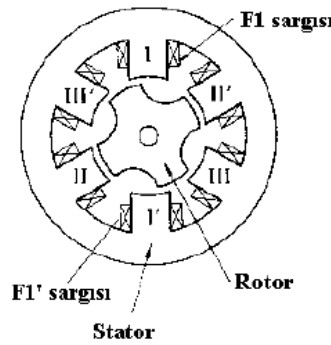
Şekil 2. 31 4fazlı bir adım motor

Basit olarak 4-fazlı sabit mıknatıslı bir adım motor örneği yukardaki şekilde gösterilmiştir. İki kutuplu sabit mıknatıslı rotor, 4(dört) kutuplu stator içinde döner. Burada C ile adlandırılan terminal, her bir fazın birer uçlarının birleştirilerek güç kaynağının pozitif ucuna bağlandığı ortak uçtur. Eğer fazlar Faz1, Faz2, Faz3, Faz4 sırasıyla uyarılırsa; rotor saat ibresi yönünde (CW) hareket edecektir. Bu motorda, adım açısının 90° olduğu açıkça görülmektedir.

Küçük boyutlu sabit mıknatıslı rotorda çok sayıda kutup oluşturmak zor olduğundan bu tip motorlarda adım genişliği $30-90$ derece arasındadır. Bu tür motorların momenti diğerlerine göre oldukça düşüktür.

2.8.2 Değişken Relüktanslı (Variable Reluctance, VR) Adım Motorları

Değişken relüktanslı adım motorları, sabit mıknatıslı adım motorları gibi bir kaç kutbu olan statora sahiptirler. Bu motorları sabit mıknatıslı adım motorlarından ayıran en önemli fark rotorlarında sabit mıknatısın olmamasıdır. Ayrıca rotorları da dişli yapıdadır.



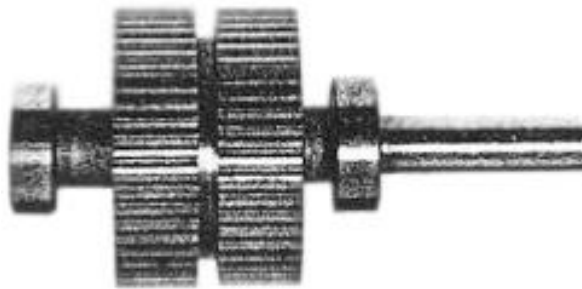
Şekil 2. 32 3fazlı değişken relüktanslı adım motor

Yukarıdaki şekilde üç-fazlı değişken relüktanslı adım motorun kesit görünüşü verilmiştir. Bu tür motorlarda rotor ve stator dişlerinin sayısı aynı değildir. Şekildeki motorda rotor 4, stator 6 dişe sahiptir. Bir faz uyarıldığında asıl askıyı sadece iki rotor dişi taşır. Bu arada diğer rotor diş çifti ise uyarılmamış stator dişlerinin tam ortasındadır.

Eğer uyarılan fazlarda değişiklik yapılırsa, yeni uyarılan stator fazıyla karşı karşıya gelecek olan rotor dişleri bu dişler olacaktır. Değişken relüktanslı motorun stator ve rotor dişleri arasındaki hava aralığı, küçük çaplı rotordan mümkün olduğunca büyük moment üretebilmek ve konumlamada yüksek doğruluk elde edebilmek için mümkün mertebe küçük tutulmalıdır. Hava aralığı diğerinden biraz daha büyük olan iki adım motora aynı değerde manyetik kuvvet uygulandığında, küçük hava aralığı yüksek momenti üreten manyetik akıyı verecektir ve ayrıca rotora dışarıdan bir kuvvet uygulandığında küçük aralığın denge pozisyonundan uzaklaşmasının da az olacağı açıktır. Modern motorlarda bu aralık 30 ila 100 μm 'dir. Bu tip motorlarda rotor, manyetize olmamalı hafif ve küçük boyutlu yapılmalıdır. Rotor ölçülerinin küçük yapılması eylemsizlik momentinin de küçük olmasını sağlar. Böylece uygulanan gerilimin meydana getireceği moment sebebiyle rotor çok çabuk hareketlenir. Değişken relüktanslı adım motorların başlama, durma ve adımları, sabit mıknatıslı adım motorlarından daha hızlıdır.

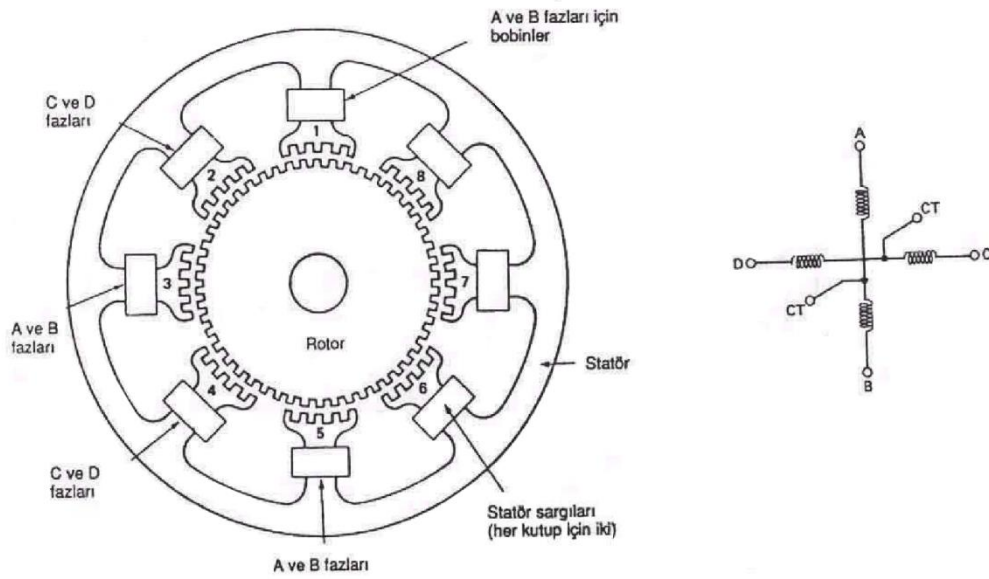
2.8.3 Hibrid Adım Motorları

Hibrid adım motorların rotorunda sabit mıknatıs bulunur. Bu tip motorlara hibrid denmesinin sebebi, motorun değişken relüktanslı ve sabit mıknatıslı motorların çalışma prensiplerinin bir kombinasyonu ile çalışmasıdır.



Şekil 2. 33 Hibrid adım motorunun rotor yapısı

Rotorun her bir ucuna, tipik olarak üzerinde elli diş bulunan dişli türü göbekler vardır. Göbekler amaçlı olarak ayarsız yapılır, böylece mıknatısın kuzey ucundaki dişler, güney ucundakiyle 180° faz dışı kalır. Şekildeki fotoğraf, rotor yapısını net olarak göstermektedir. Hibrid adım motorlarında tipik olarak 8 stator kutbu bulunur. Her kutupta bulunan diş sayısı iki ila altı arasındadır. Ayrıca rotorun istenen konuma gelmesini sağlamak üzere mıknatıs akısının ilgili kutuplar üzerinden akışını desteklemek veya engellemek amacıyla stator kutuplarına sargılar da ilave edilmiştir.

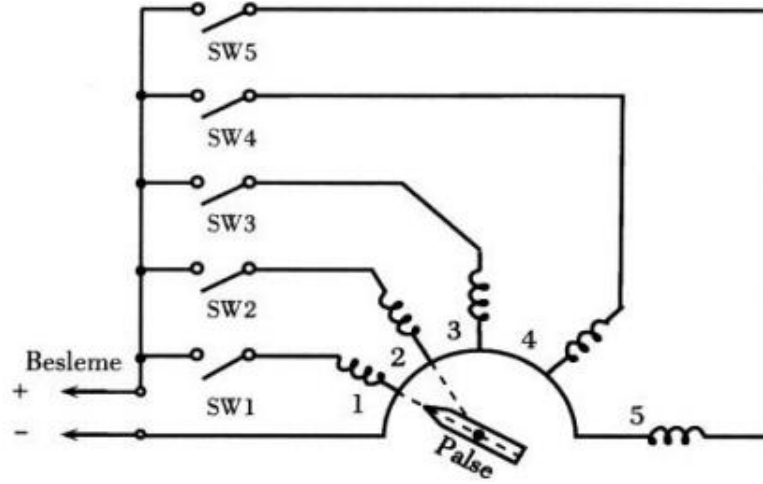


Şekil 2. 34 Hibrid adım motorunun stator sargıları

Hibrid adım motorlarında iki farklı sargı kullanılır. Her bir sargı (faz), sekiz stator kutbundan dördünü dolaşır. A ve B sargıları 1, 3, 5, 7 kutupları üzerinde ise, C ve D sargıları 2, 4, 6, 8 kutuplarındadır. Her faza ait yakın kutuplar birbirleriyle zıt yönde sarılmışlardır. Öncekilerde olduğu gibi, Hibrid adımlı bir motorun rotoru da, stator tahrik akımlarının uygun bir şekilde sıralanmasıyla adım adım hareket ettirilir. Bu tip motorların rotoru sabit mıknatıslı olduğundan, her zaman için bir tutma veya kalıntı momenti vardır. Motor, düşük hızla ivmelendiği takdirde saniyede 30.000 adıma kadar hızlarda çalışabilir. Tüm adım motorların içinde hibrid tip, her türlü doğrusal ve açısal konumlama sisteminde en yaygın kullanılan tiptir [12], [17], [18].

2.9 Adım Motor Çalışma Prensipleri

Adım motorları bir dizi kısa elektrik akımıyla hareket ederler. Stator (hareketsiz kısım) birbirine dik manyetik alan üreten iki ayrı bobinden oluşur. Bu bobinlere sırayla elektrik akımı verilerek statorun içerisinde döndürme etkisine sahip bir manyetik alan oluşması sağlanır. Statorun içindeki rotor (hareketli kısım) bobinler tarafından sırayla oluşturulan manyetik alanla polarize olarak döner. Her bir elektrik akım darbesi (pulse) rotorun belli bir açı kadar (bir adım) dönmesine neden olur. Eğer güç sadece bir bobine verilirse manyetik alanın etkisiyle rotor sabitlenecektir, bu da motorun durdurulmasında kullanılır. Aşağıdaki şekilde bir adım motorun temel prensip şeması verilmiştir.

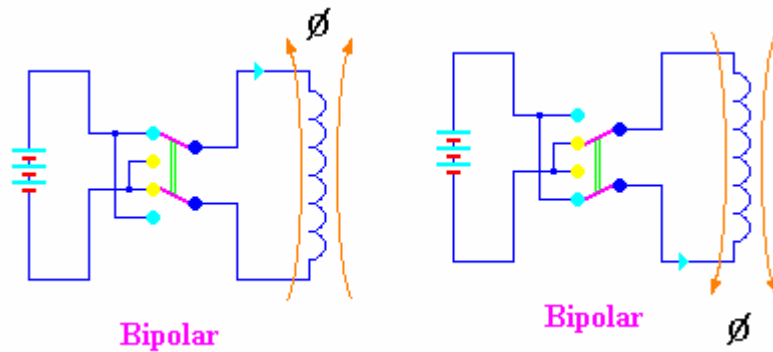


Şekil 2. 35 Adım motoru temel prensip şeması

Adım motor bir daire içinde elektromanyetik alanların dönüşü ile ifade edilebilir. 1 nolu anahtar kapandığında (SW1) rotor kendiliğinden 1. Elektromanyetik alan ile aynı hizaya gelecektir. Bundan sonra 1 nolu anahtar açılıp 2 nolu anahtar kapatılırsa sabit mıknatıs 2. elektromanyetik alanın karşısına gelecektir. Bu olaylar sırayla tekrarlanırsa rotor bir daire içinde düzgün şekilde döner. Bu şekilde verilen elektrik akım vurgularının frekansı motorun dönme hızını belirler. Hızlı ivmelenme sonucunda adım motorda kayma meydana gelebilir bunun önlenmesi için ivmelenme sırasında vurgu sıklığı ayarlanmalıdır.

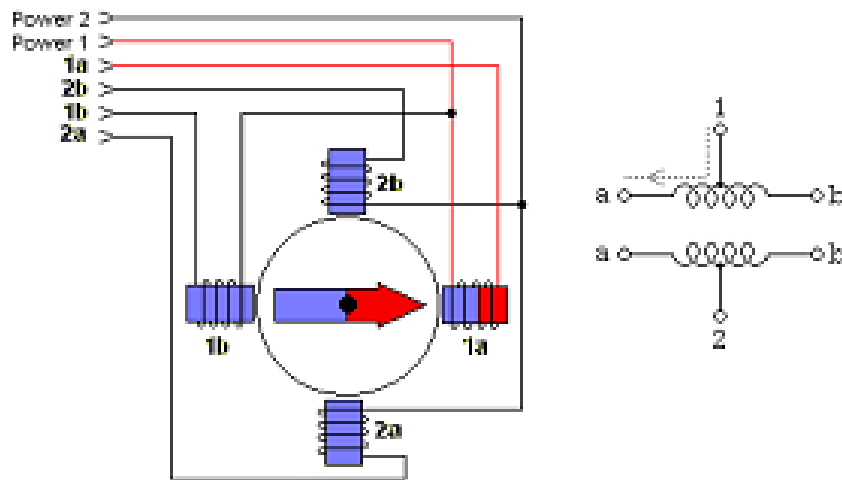
2.9.1 Çift Kutuplu ve Tek Kutuplu Adım Motoru Ne Demektir?

Bir bobinin oluşturmuş olduğu manyetik akının yönü, bobinin sarım yönüne ve bobinin içinden geçen akımın yönüne bağlıdır. Motor gibi elektromanyetik prensiple çalışan makinelerde, şiddeti ve yönü değişen manyetik alanlara ihtiyaç duyulur. Manyetik akının yönü bobinden geçen akımın yönü ile değiştirilebilir. Ancak bazı uygulamalarda akım yönünü değiştirmek yerine bobinin sargı yönü değiştirilmek istenir. Tabi ki bu işlem sargıyı söküp ters yönde sarmayı gerektirir. Ancak biri birine göre ters sarılmış iki sargı kullanılırsa, manyetik akının yönü içinden akım geçen sargının sarım yönüne bağlı olacaktır.



Şekil 2. 36 Çift kutuplu adım motoru

Çift kutuplu iki yönlü beslenen anlamına gelir ve bipolar adım motoru, iki yönde de akım akabilen motor demektir. Bipolar motorların sadece 4 ucu vardır. Unipolar tek yönlü besleme olup sargılardan sadece tek yönde akım akıtılır demektir.



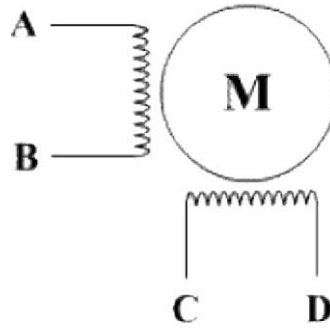
Şekil 2. 37 Tek kutuplu adım motoru

Yukarıda verilen şekillerde manyetik akının nasıl değiştiği görülmektedir. Şekillerden de anlaşılacağı üzere, tek kutuplu motorda daha çok uç (6 adet) vardır. İki motor tipinin birbirine göre avantajlarına bakacak olursak, bipolar motorlar akım yönünün değişmesi ihtiyacından dolayı daha karmaşık sürücü devreleri ister. Unipolar motorlarda sürücü daha basittir.

Aynı boyutlardaki çift kutuplu ve tek kutuplu motordan, tek kutuplu olanında aynı sargı alanına 2 bobin yerleştirildiğinden tel çapı küçük seçilir hâlbuki çift kutuplu olana daha kalın telden sargı konabilir. Kalın sargı daha çok akım dolayısı ile daha çok moment demektir. En yaygın olarak 2 fazlı adım motorları kullanılmaktadır. Bunun anlamı 2 fazlı, çift kutuplu motorlarda 2 sargı, tek kutuplu motorlarda da 4 sargı olacak demektir.

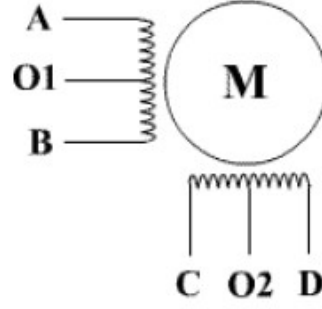
2.9.2 Adım Motorlarında Uçların Bulunması

Adım motorlarımızda uçları ya kataloğuna bakarak ya da basit metotlarla bulabiliriz.



Şekil 2. 38 Bipolar adım motorunun uçları

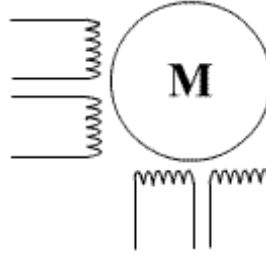
Örneğin elimizde 4 ucu olan bir adım motor olduğunu farz edelim. Bir ölçü aleti ile uçlar arasında direnç ölçümü yaparsak ohm metreden sadece A - B arasında ya da C - D arasında bir direnç değeri okuyabiliriz. A - C, A - D ya da B - C, B - D uçlarının birbirleri ile bağlantıları olmadığı için ohm metrede hiç bir değer okuyamayız (ohm metre sonsuz direnç gösterir). Bu yol vasıtasıyla hangi uçların birbiri ile bağlantılı olduğunu yani motor bobinlerinin karşılıklı uçlarını belirleyebiliriz.



Şekil 2. 39 Tek kutuplu adım motorunun uçları

Eğer elimizdeki motorun 6 ucu varsa, bu kez yine bir ölçü aleti yardımıyla uçlar arasında direnç ölçümü yaparak hangi üç ucun birbiri ile bağlantılı olduğunu buluruz. Şekilde O1 ve O2 ortak uçlardır. Dolayısıyla A – B arasında ölçülen direnç A - O1 ya da B – O1 arasında ölçülen direncin 2 katıdır. Aynı şekilde C – D arasında ölçülen direnç C – O2 ya da D – O2 arasında ölçülen direncin 2 katıdır.

NOT: Bazı motorlarda tek kutuplu sargıda iki bobinin ortak ucu dışarıya tek kablo ile çıkarılabilir. Bu durumda adım motorun 5 ucu mevcuttur.



Şekil 2. 40 8 uçlu adım motor

Bazı tek kutuplu adım motorların sargılarından ayrı ayrı uçlar çıkarılabilir. Bu durumda da adım motorun 8 ucu mevcuttur. Yukarıdaki şekilde 8 uçlu bir adım motorun bağlantı uçları görülmektedir. Yapılması gereken yine bir ölçü aleti yardımıyla hangi uçların birbiri ile bağlantılı olduğunu belirlemektir [12], [19].

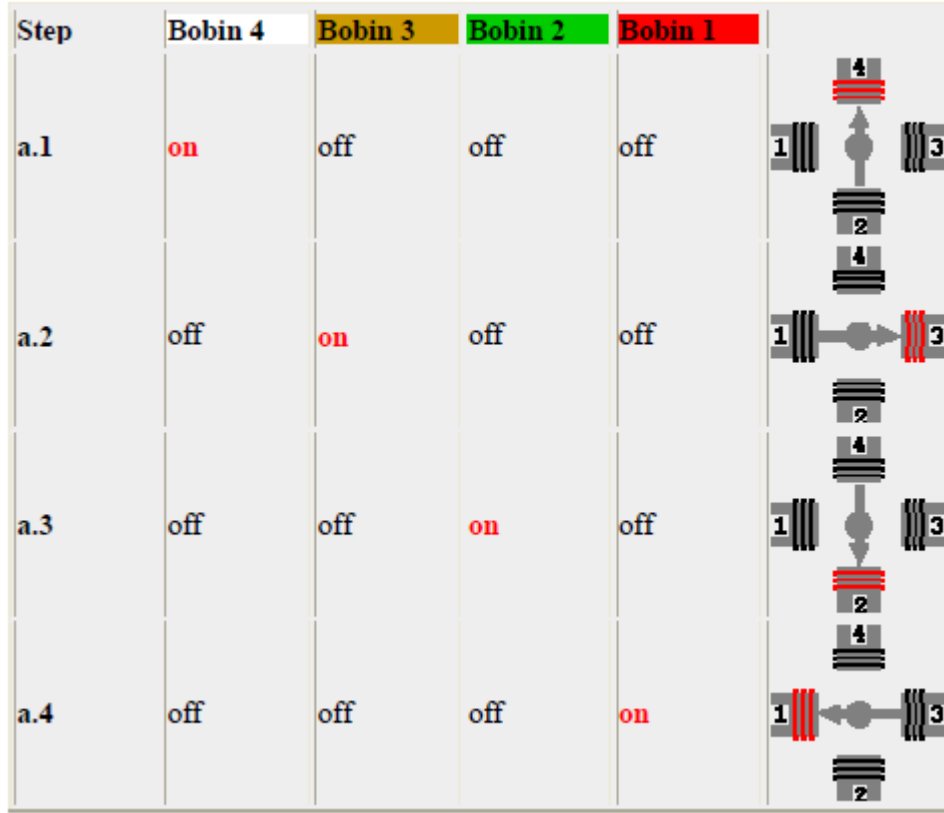
2.9.3 Adım Motorların Uyartımı

Adım motorların uyartıları bu şekildedir.

2.9.4 Tek-Faz Uyartım(1 Fazlı Tam Adımlı Sürüş)

Motor sargılarının sadece birinin uyartıldığı uyartım cinsine tek-faz uyartımı adı verilir. Aşağıdaki şekilde 4-fazlı adım motoru için tek-faz uyartım sırasındaki fazların durumu

görülmektedir. Bu uyarım metodunda rotor her bir uyarım sinyali için tam adımlık bir hareket yapmaktadır. Uyarım dönüş yönüne bağlı olarak sıra ile yapılır.



Şekil 2. 41 Tek fazlı uyarım

2.9.5 İki-Faz Uyarım(2 Fazlı Tam Adımlı Sürüş)

Motor sargılarının ikisinin sıra ile aynı anda uyarıldığı uyarım cinsine iki-faz uyarımı adı verilir. Aşağıdaki şekilde 4-fazlı adım motoru için iki-faz uyarım sırasındaki fazların durumu görülmektedir. İki faz uyarımda rotorun geçici durum tepkisi tek-faz uyarıma göre daha hızlıdır. Fakat burada güç kaynağından çekilen güç iki katına çıkmaktadır.

Step	Bobin 4	Bobin 3	Bobin 2	Bobin 1	
b.1	on	on	off	off	
b.2	off	on	on	off	
b.3	off	off	on	on	
b.4	on	off	off	on	

Şekil 2. 42 İki fazlı uyarım

2.9.6 Karma Uyarım(2 Fazlı Yarım Adımlı Sürüş)

Bu uyarım yönteminde tek-faz uyarımı ile iki-faz uyarımı art arda uygulanır. Burada rotor her bir uyarım sinyali için yarım adımlık bir hareket yapmaktadır. Aşağıdaki şekilde fazların uyarım sırası görülmektedir. Bu uyarım metodunda adım açısı yarıya düştüğünden adım sayısı iki katına çıkmaktadır [20], [21].

Step	Bobin 4	Bobin 3	Bobin 2	Bobin 1	
a.1	on	off	off	off	
b.1	on	on	off	off	
a.2	off	on	off	off	
b.2	off	on	on	off	
a.3	off	off	on	off	
b.3	off	off	on	on	
a.4	off	off	off	on	
b.4	on	off	off	on	

Şekil 2. 43 Karma uyarım

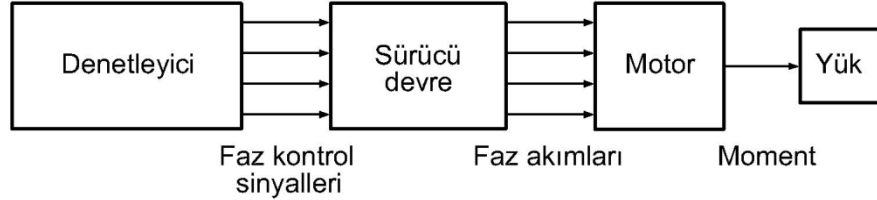
2.10 Adım Motorların Denetimi

Adım motorların 2 tip denetim vardır. Bunlar şu şekildedir;

2.10.1 Açık Döngü Denetim

Şekil 2. 43'de açık döngü denetim için blok diyagramı görülmektedir. Sayısal kontrol sinyalleri denetleyici tarafından üretilir ve sürücü devre tarafından yükseltilip adım motorunun sargılarına uygulanır. Eğer denetleyici olarak mikrodenetleyici veya bilgisayar kullanılırsa bu elemanların getirdiği esnekliklerden dolayı aynı denetleyici ile

farklı adım motorları kontrol edilebilir. Kontrol edilecek adım motorları 3, 4 veya daha farklı faz sayısına sahip olabilir. Ayrıca kullanılacak uyartım metodu için tek-fazlı, iki-fazlı veya yarım adım uyartılarından herhangi biri seçilebilir. Bu uyartım metotlarından hangisinin kullanılacağı daha önce de açıklandığı gibi motorun kullanılacağı sisteme bağlıdır.

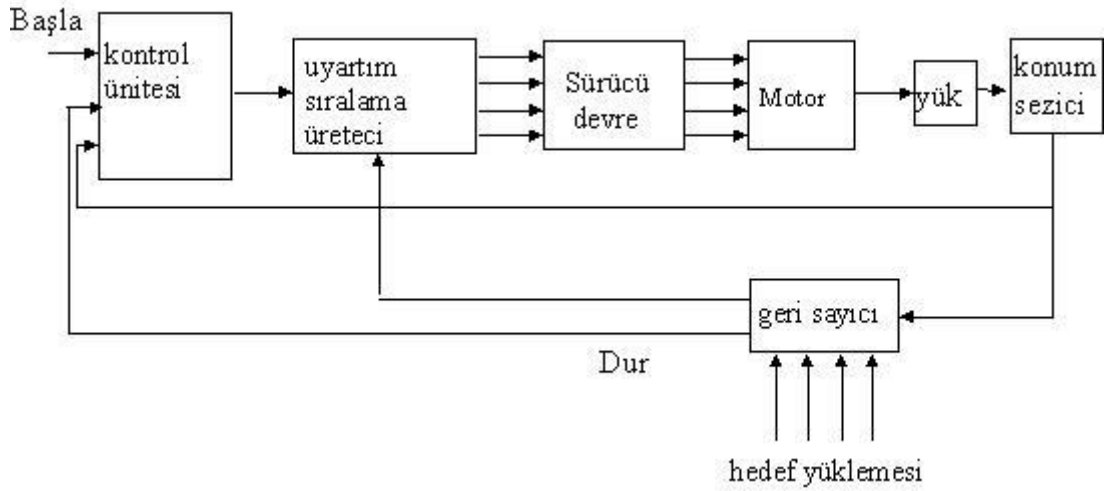


Şekil 2. 44 Karma uyartım

Denetleyici tasarlanırken motorun cinsi ve yükün durumu göz önünde bulundurulmalıdır. Bu sırada meydana gelen sınırlamalar kalıcı veya geçici durum sınırlamaları olabilir. Açık döngülü denetimde motorun konumu bilinmediğinden dolayı motorun gönderilen bütün adım komutlarını yerine getirdiği varsayılmaktadır. Eğer uyartım hızı çok yüksek ise, motor adım komutlarından bir kısmını yerine getiremeyebilir. Bu durumda kalıcı bir hata meydana gelir. Bu tür hataların meydana gelmemesi için motor yükünün en büyük olduğu durum göz önüne alınarak hata yapılmayan en yüksek hız belirlenip, bu hızın üzerindeki hızlarda uyartım yapılmamalıdır.

2.10.2 Kapalı Döngü Denetim

Kapalı döngü sistemlerde ani rotor konumu sezilerek denetim birimine iletilir. Her adım komutu için bir önceki komutun gerçekleştirildiği adım bilgisi alınarak uygulanır. Bu nedenle motor ile denetleyici arasında herhangi bir adım kaybı olmaz. Kapalı döngü denetime bir örnek aşağıdaki şekilde gösterilmiştir.



Şekil 2. 45 Kapalı döngü denetimi

İlk olarak geri sayıcıya hedef konum yüklenir. Daha sonra başla komutu verilerek adım komutlarının sıralayıcıya uygulanması sağlanır. Adım komutlarına bağlı olarak motor adım hareketi yapmaya başlar. İlk adım tamamlanınca, konum sezici geri sayıcıyı ve denetim birimlerini uyarır ve geri sayıcı değeri bir azalır. Eğer bu denetim açık döngülü yapılırsa, geri sayıcı adım komutlarının sayısını yine saklar fakat komutun uygulanıp uygulanmadığı bilinmez. Konum sezici, denetim birimine yeni adım komutu üretimi için sinyal gönderir. Ağır yükler için adım komutları arası sürenin daha büyük olması nedeniyle adım komutlarının art arda gelmesi istenmez.

Yüke göre hız ayarlaması yapılır ve motor hedef konuma gelene kadar bu olaylar tekrarlanır. Adım motoru hedef konuma gelince denetim birimi dur komutu ile uyarılarak yeni adım komutu üretilmesi engellenir. Kapalı döngü sistemi, adım motorunu yük durumunu da göz önüne alarak uyartım sürelerini ayarlar ve en uygun hız profilinde çalıştırır [17], [20], [22].

2.11 Adım Motorların Çalıştırılma Şekilleri ve Teknikleri

Adım motorlar çalışmalarında olduğu gibi uyartımda da fazla esnekliğe sahiptirler. Bu esneklik, maksimum çıkış güç, maksimum etki, maksimum tepki ve minimum giriş gücünde olmaktadır.

2.11.1 Başla-Dur Adımlama Oranı

Motor sargılarının sadece birisinin uyarıldığı uyartım cinsine tek-faz (Single Coil) uyartım denir. Uyartım CW (saat yönü) için 1000.0100.0010.0001 şeklinde CCW (saat yönünün tersi) için 0001.0010.0100.1000 şeklinde olmalıdır.

Çizelge 2. 1 Tek faz uyartım çizelgesi (CW)

Adım	Faz1	Faz 2	Faz3	Faz 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

2.11.2 Düzgün Hız

Motor sargılarının ikisinin sıra ile aynı anda uyarıldığı uyartım şekline denir. İki faz uyartımda rotorun geçici durum tepkisi tek-faz uyartıma göre daha hızlıdır ancak burada güç kaynağından çekilen güç iki katına çıkmıştır.

Çizelge 2. 2 İki faz uyartım çizelgesi

Adım	Faz1	Faz 2	Faz3	Faz 4
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

2.11.3 Rampalama

Bu uyartım biçiminde tek faz ve iki faz art arda uygulanır. Burada rotor her bir uyartım sinyali için yarım adımlık bir hareket yapmaktadır. Bu uyartım biçimi sayesinde, örneğin fabrika çıkışı 2 derece olan bir motorun adım açısını 1 dereceye düşürmüş oluruz.

Çizelge 2. 3 Yarım adım uyartım çizelgesi

Adım	Faz1	Faz 2	Faz3	Faz 4
1	1	0	0	1
2	1	0	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1

7	0	0	0	1
8	1	0	0	1

2.11.4 Diğer Uyarım Yöntemleri

Adım motorların uyarım metotları faz sayısına göre şöyle sıralanabilir.

1 – İki fazlı motorlarda;

a) İki faz uyarım biçimi b) İki faz düzeltme biçimi

2 – Üç fazlı motorlarda;

a) Üç faz uyarım biçimi b) Üç faz düzeltme biçimi

3 – Dört fazlı motorlarda ya da orta ucu (müşterek ucu) kullanılan iki fazlı motorlarda;

a) Dört faz uyarım biçimi b) Dört faz düzeltme biçimi kullanılır.

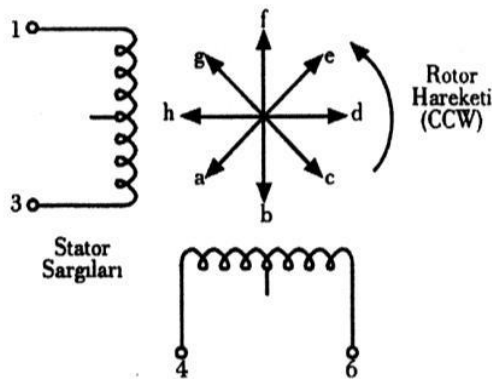
Not: Çizelgelerde adım motorun sargılarına uygulanacak gerilimin yönüne göre rotorun hareketi,

CW: Saat ibresi yönünde

CCW: Saat ibresi tersi yönünde döndüğünü ifade etmektedir.

2.11.5 İki Fazlı Motorların Çalışma Şekilleri

Bazı adım motorlarında Şekil 2.45'de görüldüğü gibi iki faz sargısı (stator sargısı) bulunur. Şekilde gösterildiği gibi her iki sargının da (fazın) orta merkez ucu olduğuna dikkat edilmelidir. Bu motor her bir uyarım sargısının yarısı bir faz gibi uyarımla çalıştırılacak olursa dört fazlı bir motor olarak çalışabilir.



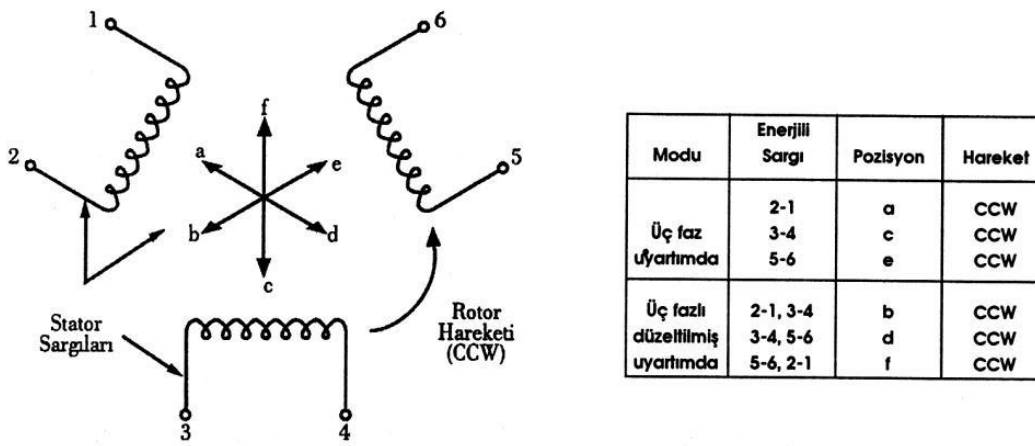
Modu	Enerjili Sargı	Pozisyon	Hareket
İki faz uyarımda	3-1	f	Index
	6-4	h	CCW
	1-3	b	CCW
	4-6	d	CCW
İki fazlı düzeltilmiş uyarımda	3-1,6-4	g	Index
	1-3,6-4	a	CCW
	1-3,4-6	c	CCW
	3-1,4-6	e	CCW

Şekil 2. 46 İki fazlı adım motor sargıları ve çalışma biçimleri

a) İki faz uyarıtım biçimi: Bu çalışma şeklinde sargılara gerilim, dış uçlardan ve yönü değiştirilerek uygulanılır. Bunun sonucunda rotor, Şekil 2.45'deki çizelgede verildiği aralıklarda ve yönde dönecektir.

b) İki faz düzeltme biçimi: Bu çalışma şeklinde yine orta uçlar kullanılmaz. Ancak her iki sargıda uygun fazlı gerilimler uygulandığında Şekil 2. 45'de verilen pozisyon ve yönde dönecektir.

2.11.6 Üç Fazlı Motorların Çalışma Şekilleri



Şekil 2. 47 Üç fazlı adım motor sargıları ve çalışma biçimleri

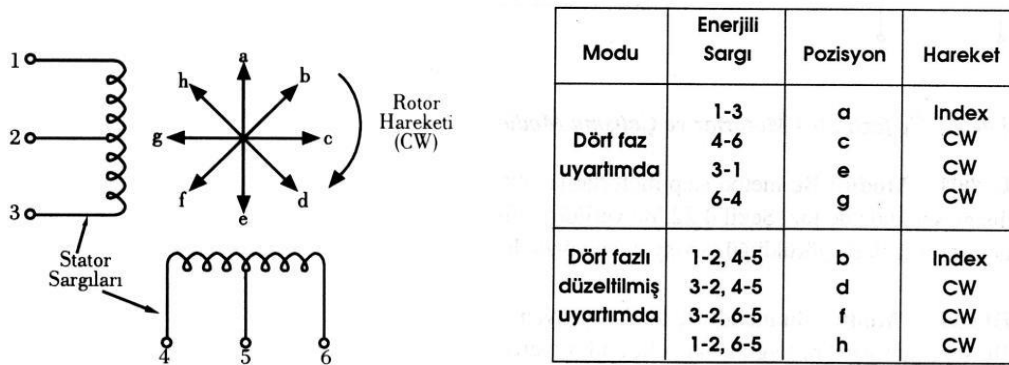
Üç fazlı adım motorlar bağımsız üç sargıdan meydana gelir. Üç fazlı motorun, uyarıtım ve düzeltme biçimlerinde saat ibresinin tersi yönünde 60° adımlarla hareket etmesi için Şekil 2.46 'da verilen çizelgede belirtilen sargılara sırayla gerilim uygulanmalıdır.

a) Üç faz uyarıtım biçimi: Bu metot adım motorlarda çok kullanılan bir sistemdir. Bu çalışma şeklinde üç faz, Şekil 2. 46 'da verildiği gibi sırayla polarılır. Bunun sonucu rotor tabloda görüldüğü pozisyon ve yönde hareket eder.

b) Üç faz düzeltme biçimi: Bu biçimde üç fazdan yan yana olan ikisi aynı anda polarılır. Bu sargılara gerilim uygulanır. Adım adım gerçekleşen dönme pozisyonu ve yönü aynı tabloda gösterilmiştir (Şekil 2. 46).

2.11.7 Dört Fazlı Motorların Çalışma Şekilleri

Dört fazlı motorlar bağımsız dört sargıdan meydana gelir. Ancak daha önceden açıklandığı gibi müşterek uçlu iki sargıya sahip iki fazlı motor, dört fazlı motor gibi çalıştırılabilir. Bu şekilde dört fazlı bir motor gibi çalıştırılan adım motor Şekil 2. 47'de verilmiştir. Dört fazlı motorun, uyarım ve düzeltme biçiminde saat ibresi yönünde 90° adımlarla hareket etmesi için Şekil 2. 47'de verilen çizelgede belirtilen sargılara sırayla gerilim uygulanmalıdır [11], [16].

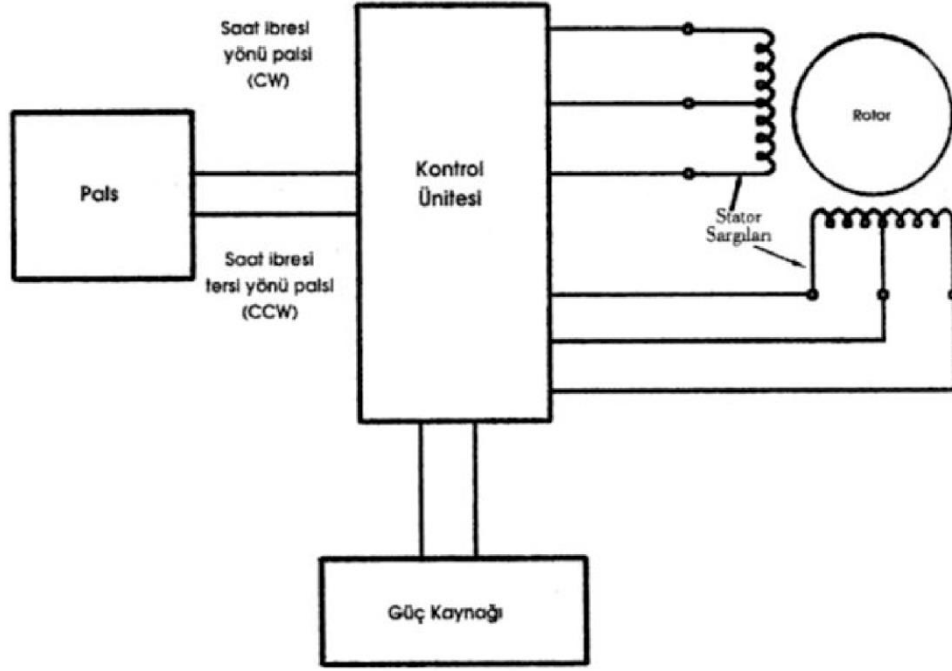


Şekil 2. 48 Dört fazlı adım motor sargıları ve çalışma biçimleri

2.12 Adım Motor Sürücü Devreleri Yapısı ve Çalışması

Adım motorları istenilen yönde ve hızda çalıştırmak istendiğinde sargılarına belli bir sırada darbeler uygulanmalıdır. Adım motorun kaç adım atacağı uygulanan darbelere bağlıdır. Fazlara uygulanacak darbeler (gerilimler) basit olarak bir anahtarlama sistemi ile yapılabilir. Bu işlemi yapan devrelere sürücü devresi veya kontrolör denir. Günümüzde elektronik devreler ile bu işlem çok kolay bir şekilde yapılmaktadır. Adım motorların ve kullanılacak yerin özelliğine göre hazırlanmış mikrodenetleyici kontrollü sürücü kartları mevcuttur. Bu kartlar sayesinde adım motorların istenilen hızda ve istenilen hassasiyette çalıştırmak mümkündür.

Bir adım motor sürücü devresinin blok diyagramı Şekil 2. 48 'da verilmiştir.

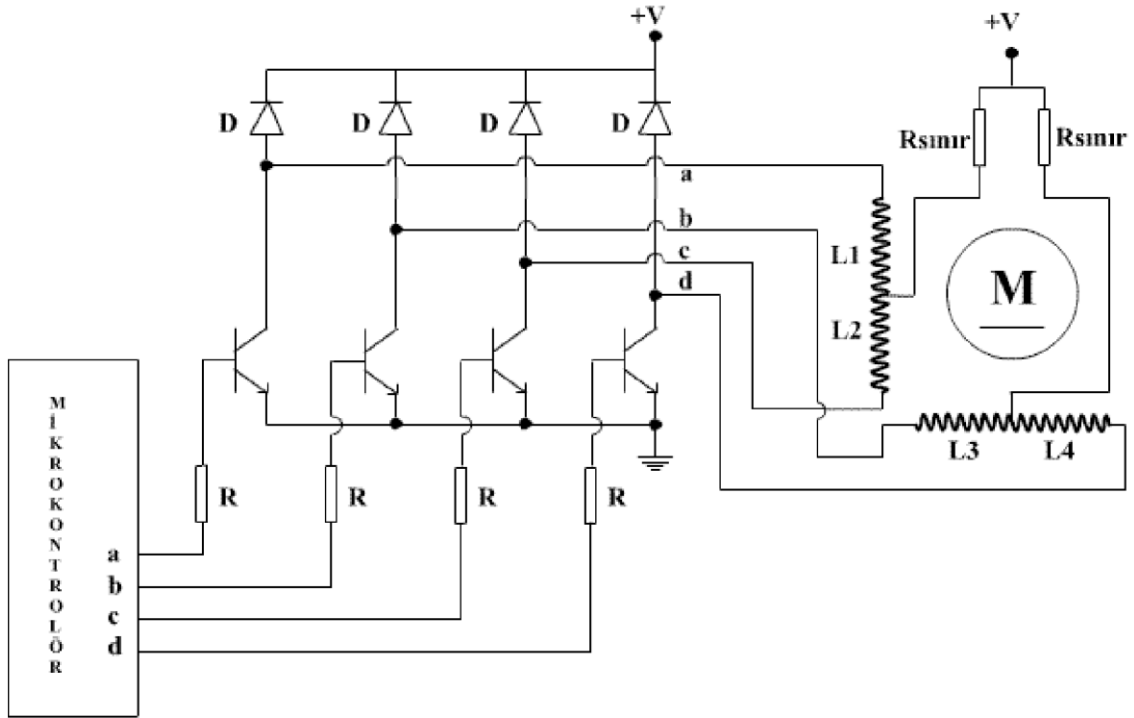


Şekil 2. 49 Adım motor sürücü devresinin blok diyagramı

Adım motorların sürülebilmesi için 2 temel noktaya dikkat etmek gerekmektedir. Bunlardan birincisi motorun bağlanacağı sürücü devresinin olmasıdır. İkincisi ise bu sürücü devresi yardımıyla motorun doğru sargılarına gerekli tetiklemeleri gönderebilmektir.

Sürücüyü tetiklemek için elektromekanik anahtarlar kullanabileceğimiz gibi bilgisayarın seri veya paralel portunu uygun bir yazılımla kullanabiliriz. Ayrıca günümüzde sanayide kullanılan adım motorlar için mikrodenetleyici kontrollü sürücüler ve bu işler için özel olarak tasarlanmış PLC'leride bulunmaktadır.

2.12.1 NPN Transistörlü Tek Kutuplu(Unipolar) Adım Motoru Sürücü Devresi



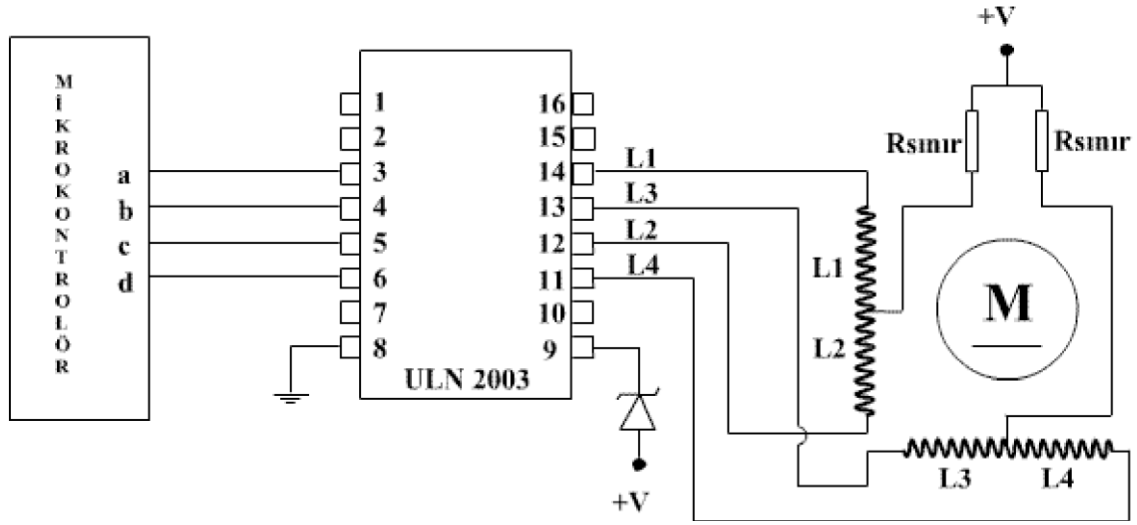
Şekil 2. 50 NPN transistörlü tek kutuplu adım motoru sürücü devresi

Devrenin amacı, motorun ortak kablolarına, motor çalışma gerilimini uyguladıktan sonra, motorun diğer uçlarını, belli bir sırada toprağa göndererek, bir adım hareketi elde etmektir. Transistorun Base ucuna, mikro kontrolörden +5V geldiğinde, transistor tetiklenir ve Collector-Emetor iletme geçer. Dolayısıyla transistorun Collector ucuna bağlı kablolar toprağa ulaşır.

Devredeki diyotlar, motorun üretmiş olduğu ters elektro motor kuvvetinden, transistorları korumak için kullanılmışlardır. Yani, motor adım attıktan sonra oluşacak ters elektro motor kuvveti, bu diyotlar sayesinde emilir. Kullanılan diyotun değerinin herhangi bir önemi yoktur.

Devredeki NPN transistorlar anahtarlama elemanı (switch) olarak kullanılmışlardır ve bu transistorların modelleri, motor için gerekli olan akıma bağlıdır. Yaklaşık 5 Amper civarı akımlar için TIP141 modeli, daha küçük akımlar için ise BC108 veya BC547B tipi transistorlar kullanılabilir. Transistorlar BC547 seçilirse $R=4.7K\Omega$ ve $D=1N4001$ seçilebilir. Motor akımını sınırlamak için, R sınır direnci motor sargılarına seri olarak bağlanabilir.

2.12.2 ULN2003 Entegresi İle Tek Kutuplu Adım Motor Sürücü Devresi



Şekil 2. 51 ULN2003 entegresi ile unipolar adım motoru sürücü devresi

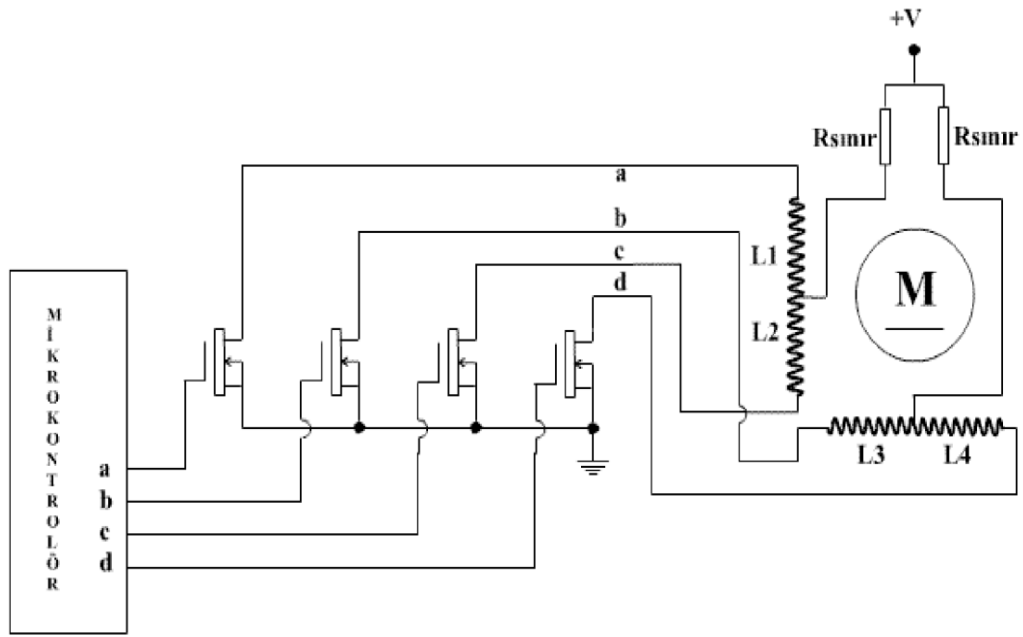
Step motor sürücü devresi olarak kullanılan ULN2003 entegresi içerisinde 7 adet NPN transistor ve dâhili diyot bulunmaktadır. Bir önceki devrede transistor bacaklarıyla uğraşmak yerine, bu entegreyi kullanmak oldukça kolaylık sağlayacaktır.

Şekildeki devreyi kurduktan sonra, mikro kontrolörden herhangi bir bacağı +5V uyguladığımızda, karşısındaki bacak toprağa ulaşacaktır. ULN2003 entegresi ile 500mA kadar akım çeken motorlar çalıştırılabilir. +12V'a kadar entegreye besleme gerilimi verilebilir.

2.12.3 MOSFET Transistörlü Unipolar Motor Sürücü Devresi

Step motor sürücü elemanı olarak, MOSFET güç transistörleri kullanmak daha iyi bir seçimdir çünkü MOSFET'lerde, BJT transistörlerde olduğu gibi ikinci bir kırılma noktası yoktur. Ayrıca anahtarlama hızları da yüksektir.

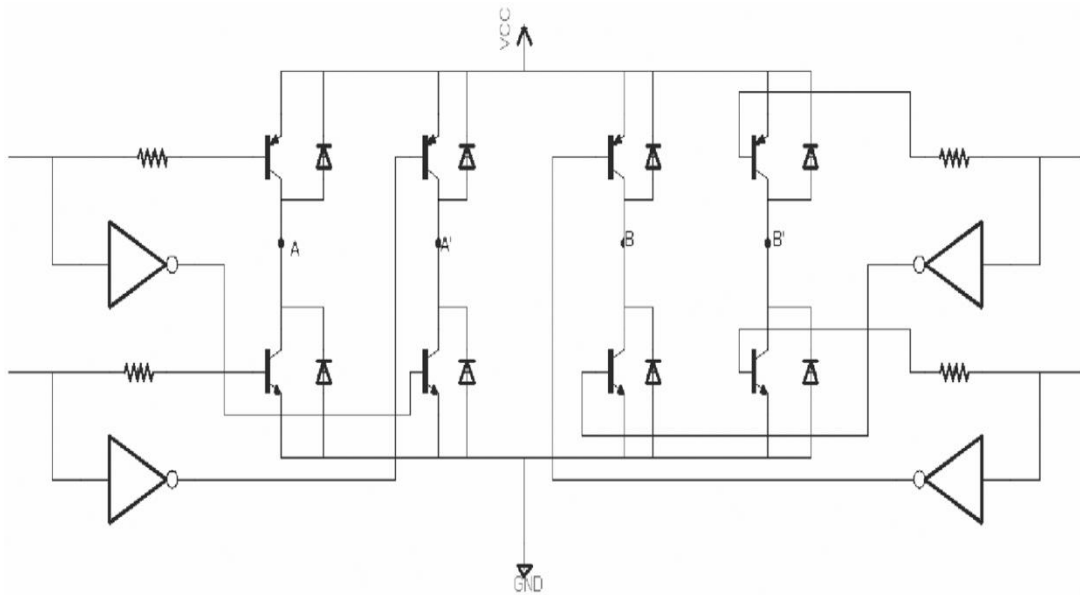
Şekilde de görüldüğü gibi MOSFET'lerin giriş dirençleri yüksek olduğu için, kontrol devresine doğrudan bağlanabilirler. Devredeki güç transistörleri (MOSFET) yerine IRLI520N ya da UN40AF ya da BD522 model transistörler kullanılabilir. Bu transistörlerin Drain ve Source bacakları arasında diyot olması nedeniyle, MOSFET'leri korumak için dışarıdan diyot bağlanmasına gerek yoktur. Bu durumda yeni bağlantı şeması aşağıdaki gibi olacaktır.



Şekil 2. 52 MOSFET transistörlü tek kutuplu adım motoru sürücü devresi

2.12.4 Transistörlü İki Kutuplu(Bipolar) Adım Motoru Sürücü Devresi

Bipolar motorlarda ayrı bobinler olup bu bobinlere verilen gerilim yönünün değiştirilmesi gerekmektedir. Bipolar adım motorlarının sürüşü bundan dolayı daha karışıktır. Bu tip motorları sürmek için genellikle H-köprüsü devreleri kullanılmaktadır [8], [16].



Şekil 2. 53 Transistörlü çift kutuplu adım motoru sürücü devresi

LabVIEW İLE ADIM MOTORU KONTROL SİSTEMİNİN DONANIMI

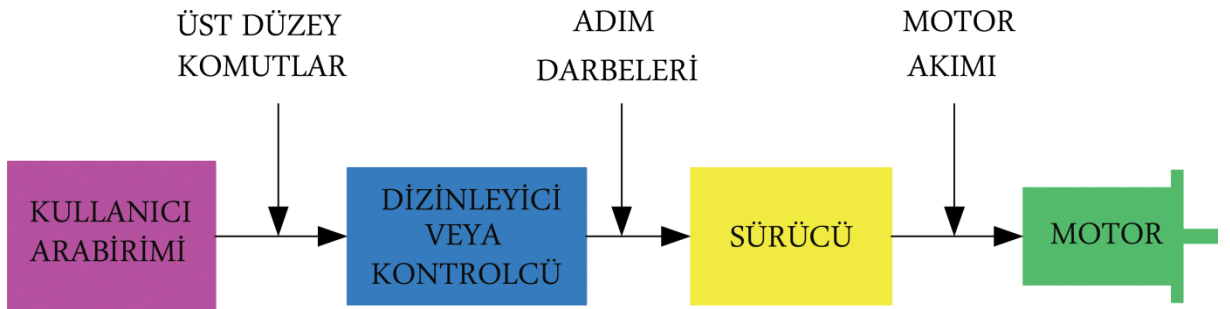
Adım motorların devrede kullanılabilmesi için sürücü devreleri mevcuttur. Bu devreler motorun çeşidine göre değişiklik göstermektedir.

3.1 Adım Motor Kontrolünde Kullanılan Donanım ve Yazılım

Adım motorları, sürüş bilgisinin gönderildiği bir sistem ile bu bilgiyi adım motora uygulayabilecek sürücü devreler ile sürülebilir. Dolayısıyla gerçekleşen sistem üç kısımdan oluşmaktadır:

- Mikrodenetleyicili donanım kısmı ki sürücü işlevini yapmaktadır
- Mikrodenetleyicinin assembler dilindeki yazılımı
- LabVIEW ortamında hazırlanmış VI

En temel olarak sistem ele alınırsa öncelikle mikrodenetleyici kullanan devreye bir programlayıcı aracılığıyla assembler kaynak kodu yüklenir ve daha sonra ise devre RS–232 kablosu aracılığıyla bilgisayar üzerinden LabVIEW programına erişir. Şekil 2.24'de adım motor kontrolünün, blok diyagramı gösterilmiştir.



Şekil 2. 54 Adım motoru kontrol blok diyagramı

Yukarıdaki şekilde kullanıcı arabirimi bir bilgisayar, PLC veya bir bilgisayar terminali olabilir.

Dizinleyici bir mikrodnetleyicidir ki sürücü devresi için dizin darbeleri veya yönlü sinyal üretmekle sorumludur. Sürücü devresi dizinleyiciden gelen komut sinyallerini güç sinyaline dönüştürerek motorun sargılarını tahrik eder. Motor sürücü devrelerinin birçok çeşitleri var ve farklı akım seviyeleri için farklı devreler tasarlanmaktadır, dolayısıyla bir sürücü devresi her türlü adım motorunu çalıştırmak için uygun olmaya bilir [5].

3.2 PIC18F2520 Mikrodnetleyicisi İle Tek Kutuplu Adım Motoru Kontrol Devresi

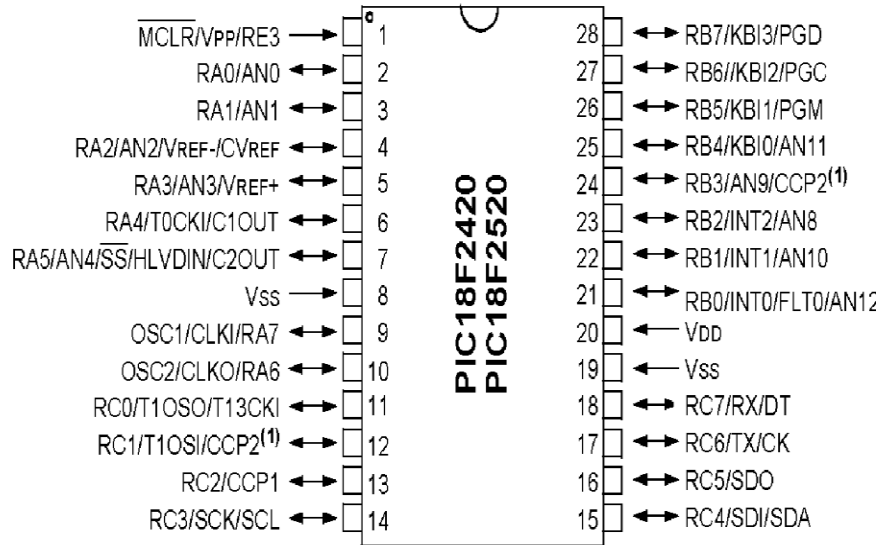
Bu bitirme tezinde motor sürücü devresi, PIC18F2520 mikrodnetleyici kullanılarak yapılmıştır ve adım motoru olarak Mitsumi M42SP-7 tek kutuplu motor kullanılmıştır. PIC18F2520 entegre devresi küçük tek kutuplu adım motorlarını sürmek için tasarlanmıştır. Bu entegre ile +50V gerilime kadar çalışan ve 500mA akım çeken adım motorları kontrol edilebilir. Entegre devre çıkışları step motora direk olarak bağlanabilir. Devrenin bilgisayar ile bağlantısı Max232 entegresi ile sağlanmaktadır. Adım motorunu sürmek için de ULN2803 entegresi kullanılmıştır.

Bu bitirme tezinde iki farklı yazılım tasarlanmıştır. Bunlardan birisi PIC mikrodnetleyicisinde kullanılan yazılımdır ki C dilinde yazılmıştır ve diğeri adım motorunun sürücü kartı üzerinden bilgisayarda görsel olarak kontrol edilebilmesi için LabVIEW programında dizayn edilmiş VI'dır. Motorun dönmesi için PIC18F2520 mikrodnetleyicisinin adım girişine bilgisayardan LabVIEW programı yardımıyla seri port üzerinden sinyal gönderilmesi gerekmektedir.

Sürücü özellikleri aşağıdaki gibidir.

- Max 50V motor voltajı
- Max 500mA faz akımı
- Adım ayarı, Yön ayarı, Moment ayarı ve Hız ayarı biçimlerinde çalışabilme
- Devrenin kontrolü görsel olarak LabVIEW programıyla seri port üzerinden yapılmaktadır

yüklenebilmektedir. Çalışma hızı DC-40 MHz olup, içerisinde üç adet zamanlayıcı ve 10 bitlik bir A/D çevirici bulunmaktadır.



Şekil 3. 2 PIC18F2520 mikrodeneleyicisinin bacakları

3.3.1 Teknik Özellikleri

- 32K Flash program bellek
- 1536 bayt RAM bellek
- 256 byte EEPROM bellek
- 32 bit genişliğinde komutlar
- Kesme (interrupt) kaynağı
- 25 giriş-çıkış portları
- 3 tane zamanlayıcı devresi
- 2 tane capture/compare/PWM devresi
- 13 tane 10 bit analog-sayısal çevirici
- Master Senkron SSP seri port (MSSP)
- SPI ve I2C bus özelliği
- Paralel/Slave port

- USART seri iletişim
- 25 mA port çıkış akımı
- Bekçi köpek devresi (Watchdog circuit)
- Uyku modu

PIC18F2520 mikrodnetleyicisi 40 MHz kadar bir saat hızında çalışabilir ve RISC (Reduced Instruction Set Computer) İndirgenmiş Komut Kümeli Mimari yapısına sahiptir. Ayrıca bu mikrodnetleyicinin 19 tane kesme (interrupt) kaynağı bulunur. PIC18F2520 mikrodnetleyici bacak tanımları aşağıdaki çizelgede verilmiştir [23].

Çizelge 3. 1 PIC18F2520 mikrodnetleyici bacak tanımları

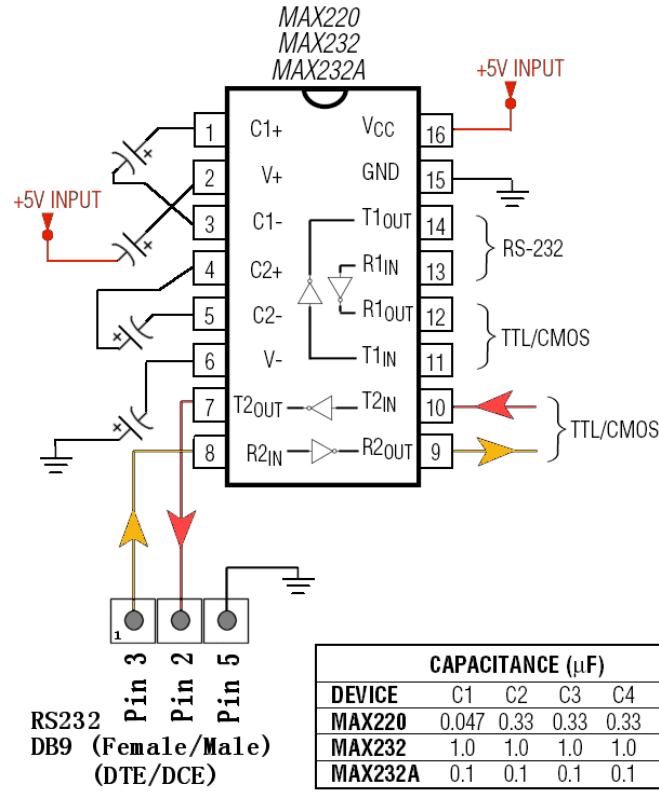
BACAK İSMİ	BACAK NO	TANIMI
MCLR/VPP/RE3	1	Reset girişi, Programlama girişi
RA0/AN0	2	PORTA bit 0, Analog girişi 0
RA1/AN1	3	PORTA bit 1, Analog girişi 1
RA2/AN2/VREF-/CVREF	4	PORTA bit 2, Analog girişi 2
RA3/AN3/VREF+	5	PORTA bit 3, Analog girişi 3
RA4/T0CKI/C1OUT	6	PORTA bit 4, TMRO saat girişi
RA5/AN4/SS/HLVDIN/C2OUT	7	PORTA bit 5, SSP slave seçme pini, Analog girişi 4
VSS	8, 19	Toprak
OSC1/CLKI/RA7	9	Osilatör girişi
OSC2/CLKO/RA6	10	Osilatör girişi
RC0/T1OSO/T13CKI	11	PORTC bit 0, Zamanlayıcı 1 osilatör çıkışı, Zamanlayıcı 1 saat girişi
RC1/T1OSI/CCP2	12	PORTC bit 1, Zamanlayıcı 1 osilatör girişi, Capture2 girişi/ Compare2 çıkışı, PWM2 çıkışı
RC2/CCP1	13	PORTC bit 2, Capture1 girişi/ Compare1 çıkışı, PWM1 çıkışı
RC3/SCK/SCL	14	PORTC bit 3, SPI ve I ² C seri saat giriş-çıkış
RC4/SDI/SDA	15	PORTC bit 4, SPI giriş, I ² C seri saat giriş-çıkış
RC5/SDO	16	PORTC bit 5, SPI veri çıkış
RC6/TX/CK	17	PORTC bit 6, USART transmit, sync, saat
RC7/RX/DT	18	PORTC bit 7, USART giriş, sync, veri
VDD	20	Pozitif Kaynak
RB0/INT0/FLT0/AN12	21	PORTB bit 0, Dış kesme (interrupt) pin0, Gelişmiş PWM arıza girişi, Analog girişi 12
RB1/INT1/AN10	22	PORTB bit 1, Dış kesme (interrupt) pin1, Gelişmiş PWM arıza girişi, Analog girişi 10
RB2/INT2/AN8	23	PORTB bit 2, Dış kesme (interrupt) pin2, Gelişmiş PWM arıza girişi, Analog girişi 8
RB3/AN9/CCP2	24	PORTB bit 3, Gelişmiş PWM arıza girişi, Analog girişi 9, Capture2 girişi/ Compare2 çıkışı, PWM2 çıkışı

RB4/KBI0/AN11	25	PORTB bit 4, Interrupt-on değişimi pin10, Analog girişi 11
RB5/KBI1/PGM	26	PORTB bit 5, Interrupt-on değişimi pin11, Seri Programlama modu girişi (ICSP)
RB6//KBI2/PGC	27	PORTB bit 6, Interrupt-on değişimi pin12. ICSP ve ICD çalışması için seri yürütme (ICSP) saat girişi
RB7/KBI3/PGD	28	PORTB bit 7, Interrupt-on değişimi pin13, ICSP ve ICD çalışması için seri yürütme veri çıkışı-girişi

3.4 RS-232 Sürücü Devresi (Max232)

Sürücü devresi ile bilgisayar arası seri bağlantıyı kurabilmek için Max232 entegresi kullanılmıştır. Bu entegrenin bacak tanımları şekil 3. 3 'de gösterilmiştir. Max232 entegresinin özellikleri aşağıdaki gibidir [24].

- Çalışma gerilimi +5 Vdc
- Çalışma hızı 120 Kbit/s
- İki adet verici ve iki adet alıcısı var
- Çalışma akımı 8mA dir



Şekil 3. 3 Max232 entegresi

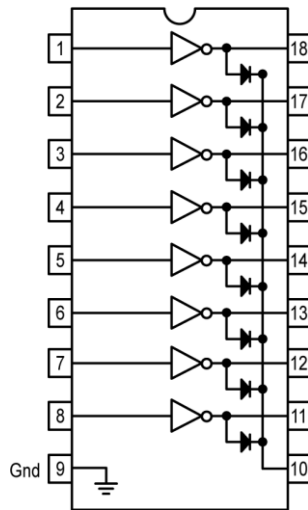
3.5 Motor Sürücü Devresi (ULN2803)

Adım motorunu sürmek için devrede ULN2803 entegresi kullanılmıştır. Bu entegrenin özellikleri aşağıdaki gibidir.

- Çalışma gerilimi +30 Vdc 'ye kadar
- Çıkış gerilimi max +50 Vdc 'ye kadar
- Entegrenin motora her faz üzerinden verebileceği maksimum akım 500mA
- Entegrenin çalışma sıcaklığı 0 ~ +70 °C

ULN2803 entegrenin bacak tanımları aşağıdaki şekilde gösterilmiştir [24].

PIN CONNECTIONS



Şekil 3. 4 ULN2803 entegresi

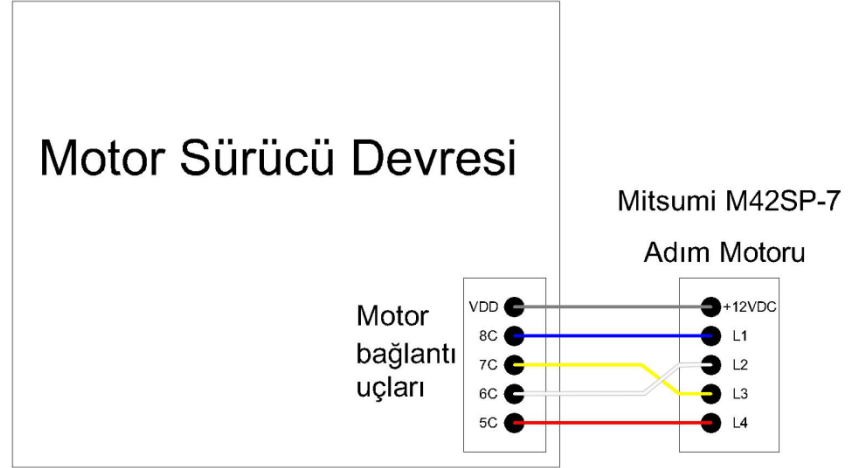
3.6 Adım Motoru

Bu projede Mitsumi M42SP-7 unipolar motoru kullanılmıştır. Adım motoru +12 VDC ve +24 VDC gerilimleri altında çalışabiliyor. Projedeki çalışma gerilimi +12 VDC olduğundan motor 12 voltta sürülmektedir. Motorun +12 VDC de özellikleri aşağıdaki gibidir [25].

- Çalışma gerilimi +10,8 ~ +13,2 VDC
- Çektiği akım 259mA
- Faz sayısı 4 adet
- Motorun DC direnci 50Ω/±7% faz başına

- Adım açısı 7,5°
- Uyartım sistemi 2-2 faz (tek kutuplu)

Aşağıda adım motorunun devreye bağlantı şekli gösterilmiştir. Motordan çıkan kablolar beş farklı renktedir ve bağlantı sırası aşağıdaki şekilde gibidir.

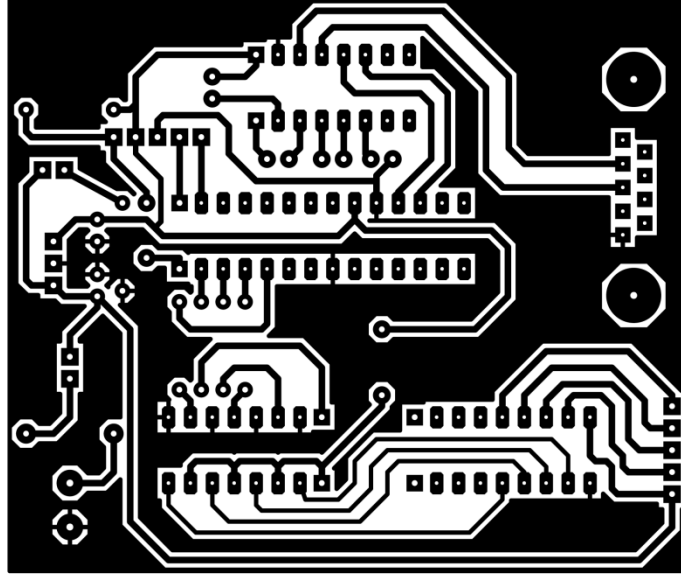


Şekil 3. 5 Adım motorunun sürücü devresine bağlantı şekli

3.7 Sistemde Kullanılan Motor Sürücü Kartı

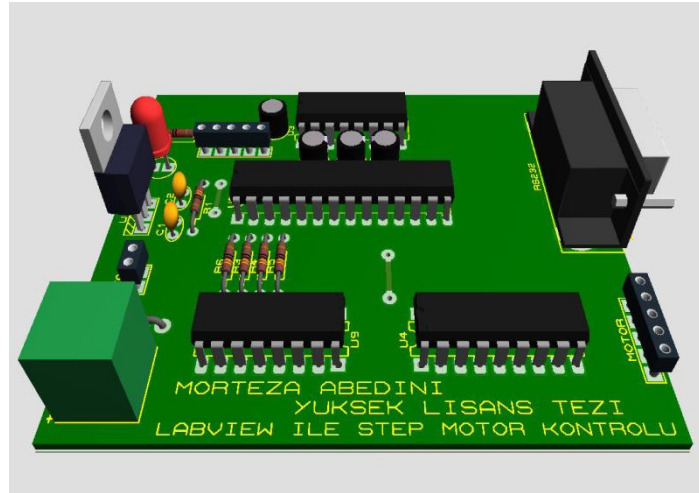
Sistemimizde kullanılan adım motoru sürücü kartı aşağıdaki gibidir. Sistemin çalışma şekli şöyledir; Seri port yardımıyla bilgisayardan LabVIEW programından motoru sürmek için bilgiler mikrodenetleyiciye gönderilmektedir. Seri port Max232 entegresiyle bilgisayarla bağlantı kuruyor ve PIC18F2520 ile LabVIEW arasındaki bilgi alış verişini sağlıyor. PIC bilgisayardan aldığı komutlar doğrultusunda adım motorunu ULN2803 entegresi üzerinden programlandığı gibi sürecektir.

Adım motor denetim donanımının baskılı devresi (PCB) aşağıdaki gibidir.



Şekil 3. 7 Adım motor denetim donanımının baskılı devresi

Devrenin şeması ve baskılı devresi Proteus programında oluşturulmuş ve şematik simülasyonu şekil 3. 8 'deki gibidir.



Şekil 3. 8 Adım motor denetim donanımı baskılı devresinin simülasyonu

LabVIEW İLE ADIM MOTORU KONTROL SİSTEMİNİN YAZILIMI

4.1 LabVIEW Yazılımı ve Kontrol Uygulamaları

LabVIEW (Laboratuary Virtual Instruments For Engineering Workbench); veri toplama ve cihaz kontrolü açısından sanayide, akademik çalışmalarda, araştırma laboratuvarlarında kullanılmaya yönelik geliştirilmiş bir programlama dilidir. LabVIEW; Windows, Mac OSX veya Linux işletim sistemleri altında çalışabilen güçlü, esnek veri analiz ve cihaz kontrol sistemidir.

LabVIEW, geleneksel programlama dillerindeki gibi satır satır yazılan bir dil değildir. Kolayca kullanılabilen grafiksel programlama ortamıdır; veri toplama (DAQ), veri analiz etme ve sonuçların sunulması konularında gerekli olan tüm araçlara sahiptir. “G” olarak ifade edilen bu grafiksel programlama dili ile programlarımızı bir grafiksel block diyagram kullanarak hazırlayabiliriz.

Ölçme ve enstrümantasyon odaklı olarak geliştirilen LabVIEW bugün bir yazılım üretme platformu olarak diğer yapısal ve nesne tabanlı programlama dillerinin hemen hemen her yeteneğine sahiptir. Hali hazırda sunulmuş modüler yapıdaki birçok fonksiyonu ve süratli(Express) VI'ları ile programlama sürecini şaşırtıcı derecede basitleştirir ve hızlandırır. National Instrument firması tarafından üretilmiş ve LabVIEW ile bütünleşmiş çalışabilen çeşitli donanımlar bir çok mühendislik uygulamalarında veri toplama, analizi ve sunumu aşamalarında çok hassas ölçümler yapmayı sağlar. Ayrıca birçok donanım üreticisi de LabVIEW sürücülerini ürünle birlikte vermektedir. LabVIEW grafiksel programlama dili teknolojisi (GPL) ile programlamada yeni bir dönemi başlatmıştır. GPL metin tabanlı kodlama yerine tamamen sembolleştirilmiş komut

setine sahiptir. Dolayısı ile komut ezberleme zorluğunu ortadan kaldırmıştır. Programcı paletlerden ihtiyacı olan fonksiyonları alarak bir akış şeması oluşturur gibi yazılım üretebilmektedir, hatta Express VI'lar ile sadece girişleri ve çıkışları bağlamak suretiyle onlarca programı dakikalar içerisinde üretebilir. Böyle bir işlem, diğer programlama dilleri ile günlerce sürebilir.

Bilgisayar simülasyonlarından faydalanmak eğitim öğretim yöntemlerinden birisidir. Bu amaçla günümüzde üretilmiş ve kullanılan birçok benzetim yazılımları mevcuttur. Bu yazılımlar gerçek hayattaki bileşenleri taklit edip onların karakteristiklerini sergileyen, böylece kullanıcıya düşük maliyetlerle yeni uygulamaları gerçekleştirme imkânı tanıyan ara yüzlerdir. 1980'lerin başında National Instrument firması tarafından geliştirilen LabVIEW, G programlama dili tabanlı bir grafiksel program geliştirme ortamıdır (Basher ve Isa, 2006). LabVIEW, cihaz kontrolü ve veri toplama işlemleri için çok güçlü bir programlama yazılımıdır. Bu yazılım sayesinde kontrol alanındaki pek çok sistemlerin denetimi yapılabilmektedir. Bunun yanında eğitim amaçlı elektronik deney düzeneklerin denetim ve yönetimine ilişkin çalışmalarda yapılmıştır (Higa vd. 2002) (Vento 1988). Bu çalışmalar bilgisayar ile tek bir elektronik düzeneğin kontrolünü amaçlamaktadırlar. LabVIEW ile program yazmak yerine kendi VI'larınızı (virtual instrument: sanal çalgı aleti) yaratırsınız. Yazılım sistemimizin aktif kontrolünü sağlayan kullanıcı ön panelini kolaylıkla oluşturur, bulduğunuz çözüm için gerekli blok diyagramları istediğiniz şekilde oluşturabiliriz.

LabVIEW'i kullanabileceğimiz işlerden bazıları aşağıdaki gibidir:

- Kalp atışı simülasyonu
- Dondurma yapım sürecini kontrol etme
- Uzay mekiğinde hidrojen gazı sızıntısını belirleme
- Bebek deve kuşlarının beslenme alışkanlıklarının izleme
- Güç kalitesini analiz etmek için güç sistemlerini modelleme
- Laboratuvar farelerinde deneyin fiziksel etkilerini ölçme
- Servo ve step motor hareketini kontrol etme

- Bilgisayarlardaki ve diğer elektronik cihazlardaki devre kartlarını test etme
- Sanal gerçeklik sistemlerinde hareketi simülasyonu

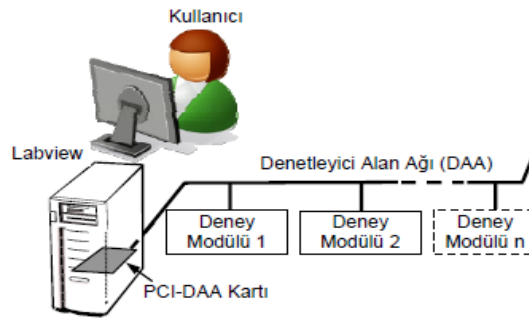
4.1.1 Denetleyici Alan Ağı (DAA)

DAA protokolü, 1980'lerin ortalarında Alman firması olan Robert Bosch tarafından, otomotiv uygulamalarında güçlü bir seri veri iletiminin oluşturulması amacıyla tasarlanmıştır (Kiencke, 1994). 1993 yılında ISO tarafından uluslararası bir standart olarak kabul edilmiştir. Denetleyici Alan Ağı protokolü, otomotiv sektöründe olduğu gibi endüstri sektöründe de yaygın olarak kullanılmaktadır. Örneğin, tarımsal makineler, tıbbi makineler ve otomasyon sistemleri içerisinde Denetleyici Alan Ağı haberleşme protokolü kullanılmaktadır (Lawrenz, 1995). Tekstil üretim sistemleri, paketleme kontrol sistemleri, robot kontrol sistemleri gibi daha birçok sistem içerisinde Denetleyici Alan Ağı protokolü popüleritesini arttırmıştır. İçerisinde Denetleyici Alan Ağı modülü bulunan ilk bütünleşmiş 1989 yılında Intel Corp. tarafından piyasaya çıkarılmıştır. Bu tarihten itibaren, Siemens, Motorola, Philips ve Microchip gibi büyük firmalar Denetleyici Alan Ağı entegrelerini üretmeye başlamışlardır.

Microchip firması, 2005 yılında toplam 200 milyona yakın Denetleyici Alan Ağı modüllü entegre satarak üretici firmalar arasında ikinci sıraya oturmuştur. DAA 2.0A protokolü, uzunluğu 88 ila 108 bit arasında değişen mesajların CSMA/CR (Carrier Sense Multiple Access with Collision Resolution) erişim metoduna uygun olarak iletilmesi prensibine dayanır. Her mesajın 11bit uzunluğunda niteliğini ve aynı zamanda sayısal değeri itibarıyla önceliğini belirleyen öntakısı (Identifier) vardır. Nitelik ile kastedilen, kullanıcı tarafından mesaja anlamlı bir sayısal değer verilmesidir. 11 bit ile 211 değişik nitelik ve öncelik tanımlaması yapılabilmektedir. DAA 2.0B sürümü için bu değer 229 dur. Sayısal olarak diğerlerinden düşük değeri olan mesajın yüksek önceliği vardır. İki mesajın aynı anda farklı kaynaklardan iletmeye çalışılması durumunda önceliği yüksek olan mesaj ortama erişme hakkına sahiptir. Denetleyici Alan Ağı protokolü ile ilgili detaylı bilgiler (Intel, 1993; Philips, 1991) referanslarda bulunabilir. Bu çalışmada DAA için tasarlanmış olan “MicroNET” yüksek seviye protokolü kullanılmıştır.

4.1.2 Sistemin Genel Yapısı

Tasarımı gerçekleştirilen sistem genel olarak donanım ve yazılım olmak üzere iki kısımdan meydana gelmektedir. Donanım kısmını kendi içerisinde yine iki bölümde incelemek gerekir. Bunlar sunucu bilgisayarı ve Elektronik Deney modülleridir. Sunucu bilgisayarı üzerinde Elektronik deney modülleriyle haberleşmeyi sağlayacak bir PCI DAA kartı mevcuttur. Elektronik deney modüllerinin her birisi ise diyot ve transistör deney setleri ile bu setleri kontrol eden DAA modüllerinden oluşmaktadır.



Şekil 4. 1 Sistemin genel yapısı

Yazılım kısmı ise LabVIEW kullanılarak geliştirilen bilgisayar üzerindeki kullanıcı arayüzü ve elektronik deney modülleri üzerindeki mikro-denetleyici yazılımından meydana gelmektedir. LabVIEW kullanılarak modüller için hazırlanmış ara yüz sayesinde modüllere Denetleyici Alan Ağı üzerinden deney setlerini kumanda edecek mesajlar gönderilerek deney setleri üzerinde değişiklikler yapılmaktadır DAA üzerinde kullanılan mesajların listesi Çizelge 4. 1 'da verilmiştir.

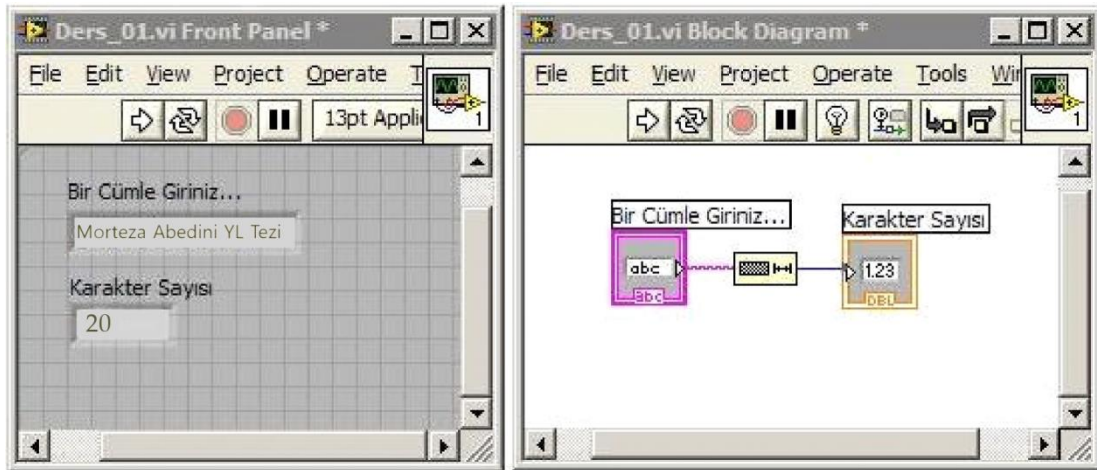
Çizelge 4. 1 DAA mesajları

Mesaj No (Hex)	Sembolik komut adı	Tanım – Mesaj Yönü
010..020, ...1F0*	Boot loader Start*	Bağlantı oluşturur (sunucudan modüle)
011..021, ...1F1	Monitor	Çıkış Port durumlarını sunucu yazılımına gönderir. (modülden sunucuya).
012..1F2	Write	Giriş Port durumlarını modül yazılımına gönderir (sunucudan modüle)
013..1F3	Set Param	Port izlemesini başlatır. (sunucudan modüle)
014..1F4	Reset	Modülü resetler (sunucudan modüle)
015..1F5		Boş
..		
01F..1FF		

* Komutlar her deney modülü için ayrı ayrı üretilir.

4.1.3 LabVIEW Kullanıcı Ara yüzünün Tanıtılması

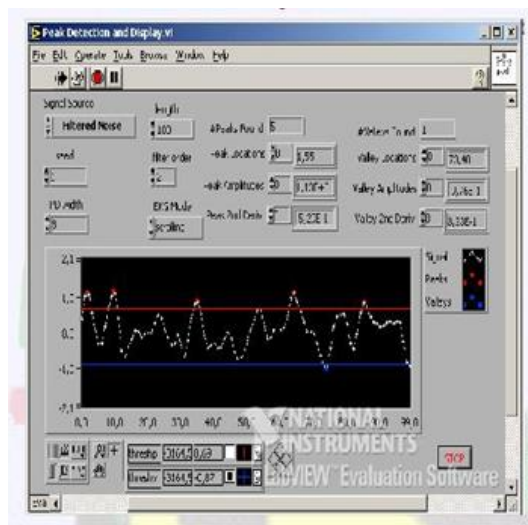
LabVIEW ekranı “Block Diagram” ve “Front Panel” olmak üzere iki ana kısımdan oluşur. Sırası ile bu iki kısım blok diyagram ve ön panel olarak adlandırılır. Blok diyagram görsel programlama dillerinde kod yazma bölümüne, ön panel ise kullanıcı ara yüzünün hazırlandığı formlara karşılık gelmektedir.



Şekil 4. 2 “Ön Panel” ve “Blok Diyagram” görüntüleri

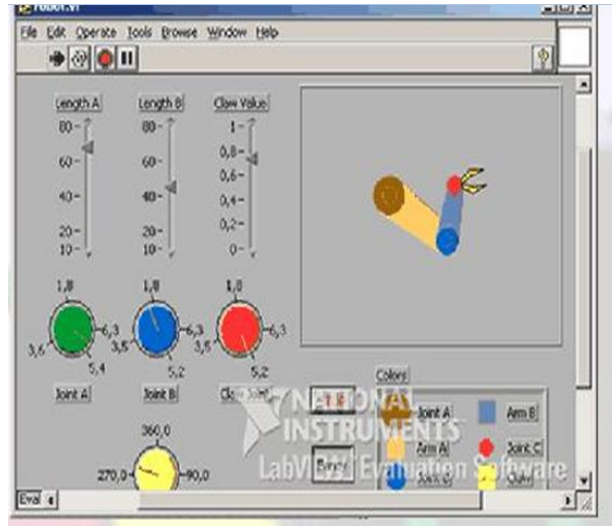
Ön panelde ve blok diyagramda kullanılan iki önemli araç paleti vardır. Ön panelde kullanılan palete “Kontrol Paleti”, blok diyagramda kullanılan palete ise “Fonksiyon Paleti” adı verilmiştir.

4.1.4 Ön Paneli Oluşturmak



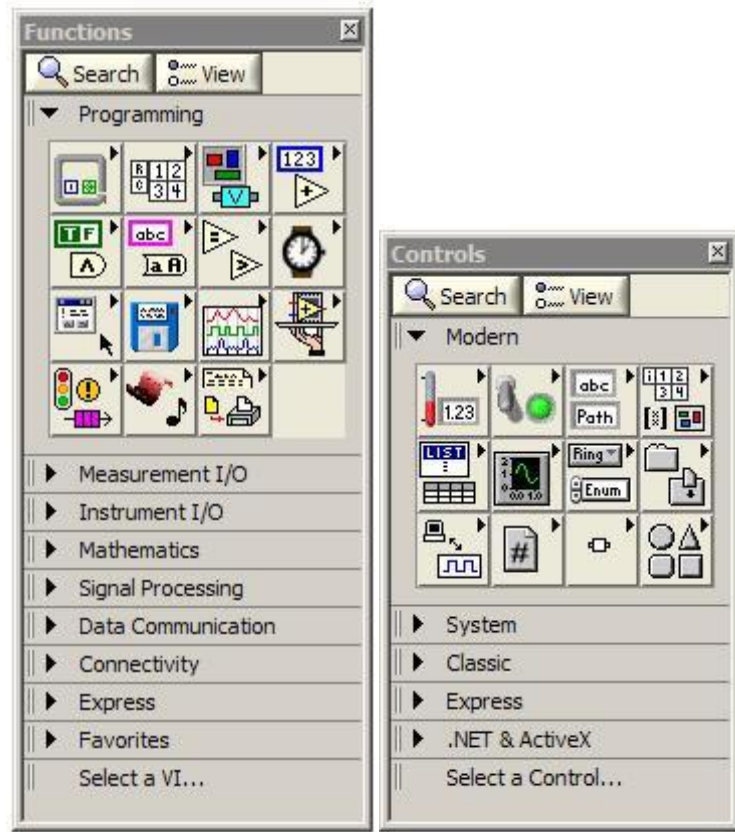
Şekil 4. 3 Ön panel

Ön panel bir sanal enstrümanın kullanıcı ara yüzüdür. Genellikle, ön panelde oluşturulan giriş ve çıkışların görevlerini yerine getirebilmesi için öncelikle ön panel daha sonra blok diyagram oluşturulur. VI'ın ön paneline kontrol paletinden seçilecek sayısal göstergeler, ölçekler, metreler, termometreler, LED'ler, çizelgeler, grafikler ve daha fazlası yerleştirilebilir. Her şey tamamlandığında çalışan VI, bir anahtarı tıklayarak, bir sürgüyü oynatarak, grafiğe zoom yaparak veya klavyeden bir değer girerek, ön panelden kontrol edilebilir.



Şekil 4. 4 Ön panel

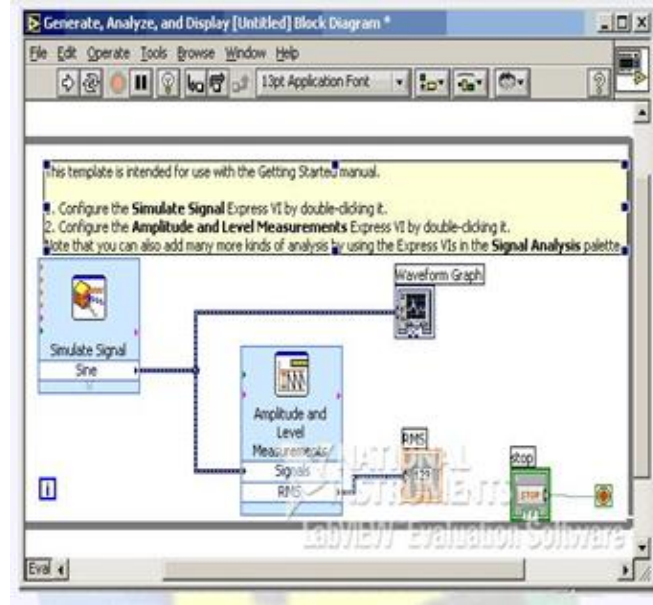
Sanal enstrümanın interaktif giriş ve çıkış terminalleri olan kontrol ve göstergeler ile ön panel oluşturulabilir. Kontroller topuzlu düğmeler, basmalı butonlar, kadran ve diğer giriş sürücüleridir. Göstergeler ise grafik LED ve diğer gösterim elemanlarıdır. Kontroller enstrümanın giriş sürücülerini canlandırır ve VI'ın blok diyagramına veri sağlar. Göstergeler ise enstrümanın çıkış sürücülerini canlandırır ve blok diyagramın işlediği ya da ürettiği verileri görüntüler. Kontrol panelini görüntülemek için View menüsünden "Controls Palette" veya "Functions Palette" seçeneği seçilir.



Şekil 4. 5 “Fonksiyon Paleti” ve “Kontrol Paleti”

Ön panel veya Blok Diyagram üzerinden yukarıdaki şekilde gösterilen paletlerden biri açılacaktır (Ön Panel ‘de Controls ve Blok Diyagram ‘da Functions açılacaktır). Palette bulunan herhangi bir kontrol veya gösterge ön panele yerleştirilebilir. LabVIEW’da kontrol paletindeki ön panel nesneleri, onların tipik kullanımı şeklinde düzenlenmiştir. Örneğin bir vites kolu anahtar, bir kontrol elemanı, LED ise gösterge elemanıdır.

Bazı alt paletlerde hem gösterge hem de kontrol elemanlarını kapsar. Örneğin; nümerik alt paleti nümerik bir kontrol ile nümerik göstergiyi kapsamaktadır. Bir kontrolü bir göstergeye çevirebilmek için, nesne üzerine sağ tıklayıp, kısa yol menüden “Change to indicator” seçilmelidir.



Şekil 4. 6 Blok diyagram

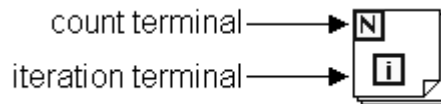
Ön panel yapıldıktan sonra, ön panel nesnelerini kontrol etmek için fonksiyonların grafiksel karşılıklarını kullanarak kodlar eklenmelidir. Blok diyagram bu grafiksel kodları kapsamaktadır.

4.1.5 LabVIEW'da For ve While Döngüsü

VI'daki tekrarlayan işlemleri kontrol etmek için For döngüsü ile While döngüsü kullanılır. For döngüsü özel olarak birkaç defa, While döngüsü özel şartın sağlanacağı ana kadar yerine getirilir. Bu döngüleri Functions menüsü altında Structs&Constants'da bulabilirsiniz.

4.1.6 For Döngüsü

Aşağıdaki şekilde görüldüğü gibi For döngüsü, kendi sınırları içinde kodu uygular, sayım terminalinde içerdiği değeri sayıcıya eşitleyen sayıcı zamanlayıcısının bir toplamıdır.



Tekrar terminali, tamamlanmış döngünün geçerli değerini gösterir. N-1'e kadar birinci tekrarda 0, ikinci tekrarda 1 vs. Sayım terminaline 0 koyarsanız döngü işlemez.

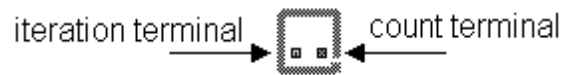
For döngüsü aşağıdaki formülle bulunur.

For i=1 to N-1

Alt diyagram uygulaması

4.1.7 While Döngüsü

Aşağıdaki şekilde görüldüğü gibi While döngüsü sınırı içerisinde boolean değerinin, şartlı terminalin FALSE (Yanlış) sağlayacağı alt diyagramı içerir. LabVIEW her bir tekrardan sonra şartlı terminal değerini gözden geçirir. Eğer değer TRUE (Doğru) ise sıradaki tekrar meydana gelir. Şartlı terminalin değeri FALSE (Yanlış) olursa ve eğer siz onu böyle bırakırsanız döngü bir kez tekrarlanır.



While döngüsünün tekrar terminali For döngüsünün aynısını yapar.

While döngüsü aşağıdaki formülle bulunur.

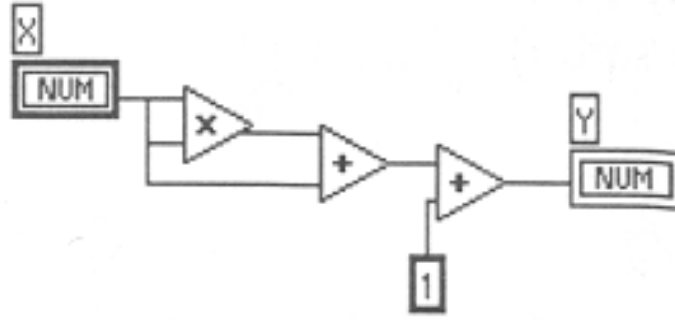
Do

Alt diyagram uygulaması

While koşulu doğru

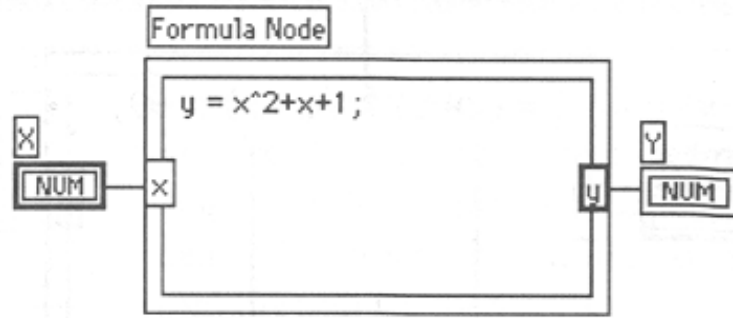
4.1.8 Formül Düğümleri

Formül düğümleri, blok diyagrama direk olarak cebirsel formülleri girebilmek için kullanılan büyüklükleri değiştirebilen kutulardır. Bu özellik karmaşık formüllerin çözümünde çok kullanışlıdır. Örneğin, $Y = x^2 + x + 1$ eşitliğini ele alalım. Eğer, LabVIEW aritmetik fonksiyonları kullanılarak çözümlenirse, onun blok diyagramı aşağıdaki gibi olur.



Şekil 4. 7 Blok diyagram

Aynı eşitliği, formül düğümünü kullanarak şekildeki gibi kullanabilirsiniz.

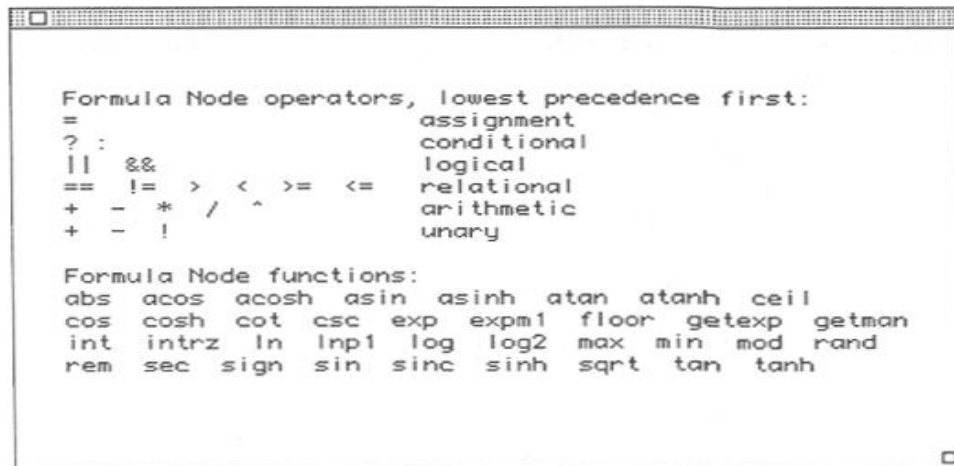


Şekil 4. 8 Formül düğümü blok diyagramı

Formül düğümü ile direk olarak tamamlanmış formül ya da formüller, tamamlanmış blok diyagramın alt kısmına girilir. Bazen şekilde kutu içine de formüller girilir. Formül düğümünün girdi çıktı terminalini oluşturmak için düğüm kenarında **Add Input** ya da **Add Output** seçilir. Ardından kutulara değişken tanımlamaları girilir. Adların limiti iki karakterdir ve duyarlı durumdadır. Her formül (;) ifadesi sonlandırılmalıdır.

Functions menüsünden **Structs**'ı seçerek formül düğümünü blok diyagrama yerleştirirsiniz.

Aşağıdaki işlem ve fonksiyonlar formül düğüm içinde kullanılabilir.



Şekil 4. 9 Formül düğümü

Aşağıdaki örnekte formül düğümü içinde şartlı dallanmayı deneyebilirsiniz. Şekilde olduğu gibi, Alistirma 1. 3'e benzer kod parçaları farz edelim, öyle ki eğer x pozitif ise onun karekökünü bulsun ve y'ye aktarsın. Eğer x negatif ise, kod y'ye -99'u aktarsın.

if ($x \geq 0$) then

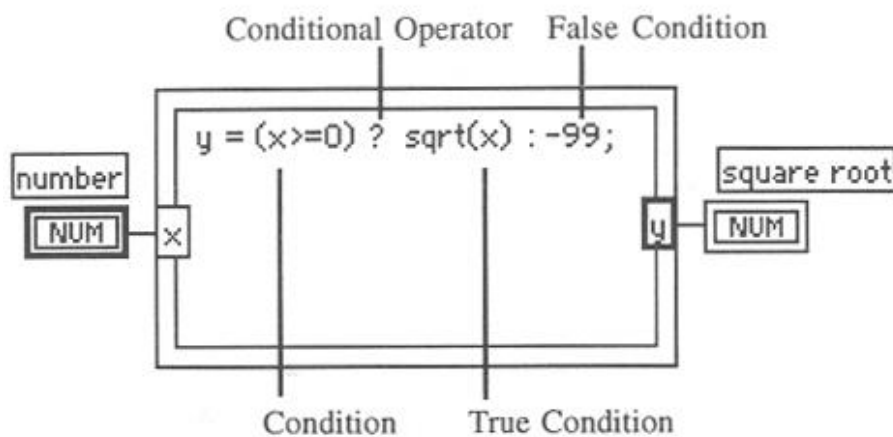
$$y = \text{sqrt}(x)$$

else

$$y = -99$$

end if

Aşağıdaki şekilde olduğu gibi, formül düğümünü kullanarak parça kodu uygulayabilirsiniz.



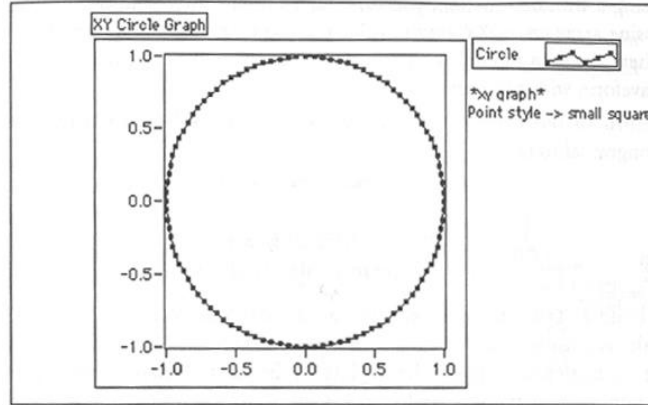
Şekil 4. 10 Formül düğümü

4.1.9 Bir XY Grafiğini Daire Çizme İçin Kullanma

Bağımsız x ve y düzenekleri kullanarak XY grafiğinde daire çizecek VI kuracaksınız.

Ön Panel

1. Yeni bir ön panel açınız. Gösterilen paneli yeniden oluşturacaksınız.

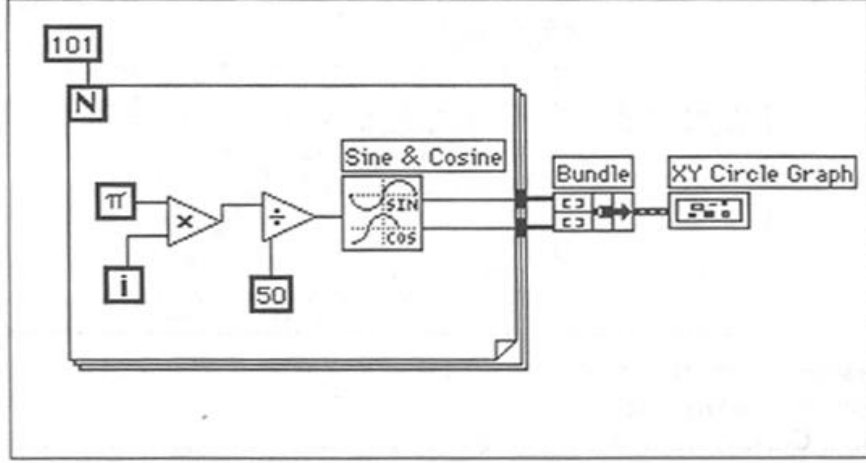


Şekil 4. 11 Ön panel

2. Panel penceresine bir XY daire grafiği (**Array&Cluster** menüsü) yerleştiriniz. Grafiği XY daire grafiği diye adlandırınız.
3. Grafiği bir köşeden ayar çubuğuyla çekerek genişletiniz. Çizim bölgesini hemen hemen kare yapmaya çalışınız.
4. Grafiğe pop- up menüsünden **Show ► Legend**'i seçiniz. Legend'i sol taraftan yeniden ayarlayınız ve daire etiketini adlandırma çubuğuyla giriniz. Yazı üzerindeki çizgiye tıklayınız ve **Point Style** paletinden küçük kareyi seçiniz. Ondan sonra Color paletinde yeni bir çizim rengi seçiniz.

Blok Diyagram

1. Blok diyagramı şekildeki gibi oluşturunuz.



Şekil 4. 12 Blok diyagram



Sine&Cosine fonksiyonu (**Arithmetic** menüsü). Bu örnekte bir sinüs dalga devresini ve bir kosinüs dalga devresini temsil eden noktalardan oluşan bir düzenek kurmak için bu fonksiyonu kullanacaksınız.



Bundle fonksiyonunu (**Array&Cluster** menüsü). Sinüs düzeneğine karşı kosinüs düzeneğini çizmek için sinüs ve kosinüs düzeneğini birleştirir.



Pi Sabiti (**Structs&Constants** menüsü).

Bundle fonksiyonunu kullanarak, bir daire oluşturan XY grafiğinde tek daire kosinüs sırasına karşı tek daire sinüs dizisini çizebilirsiniz. XY grafiği, daire gibi planlanmış verinin çok değerli bir fonksiyon olduğu ya da verinin tek formu olmayan bir zaman tabanıyla dalga formu olduğu durumlarda faydalıdır.

2. Ön panele geri dönün ve VI'yı çalıştırın. MYWORK.LLB'de **Graph Circle.vi** olarak saklayın. Tebrikler.

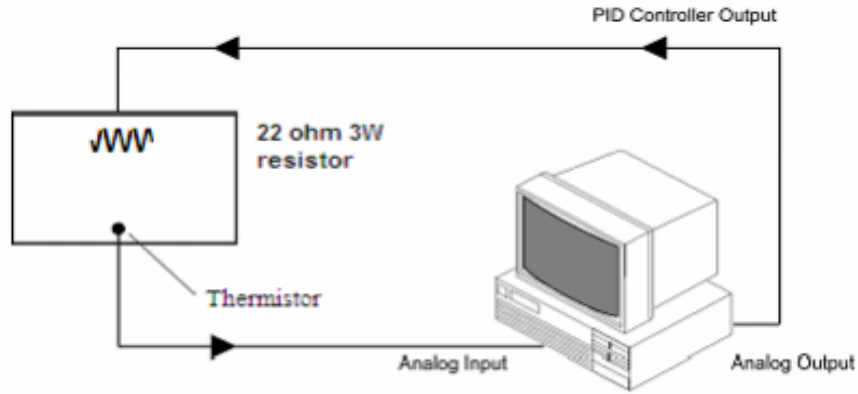
4.1.10 Paketleme

LabVIEW'in grafik ve kartlarını kullanarak veri örgüsünü gösterebiliriz. Tablolar etkileşimsel olarak, bir noktayı (veya noktaları) işaretleyerek, eski verilere yenilerini eklerler, böylece parçada şu anki değeri önceki değer ile birlikte görülebilir. Grafikler ise tamamen bir veri topluluğunu gösterir. LabVIEW, iki tür grafik sağlar; dalga formu ve XY grafikleri.

Dalga formu grafiđi, sadece zamana gre deđiřken dalga biimleri gibi, x eksenine gre dzgn bir řekilde dađılmıř noktalarla tek deđerli fonksiyonları birleřtirir. XY grafiđi genel amalıdır, kartezyen grafiđi ok deđerli fonksiyonların planını izmenize izin verir, rneđin dairesel řekiller. Y eksenini bir x eksenine bađlar. Tablo ve grafiklerin grnmne yazı ve palet kullanarak deđiřtirebilirsiniz. Verinize uyum sađlaması iin lleri de deđiřtirebilirsiniz. Hem tablolar hem de grafikler birden fazla noktadan geerler. Veri eřitleri karıřık olabilir, bunun iin kendi VI'larını yazarken bu nitedeki rneklere bir kalıp olarak bařvurabiliriz [27] - [32].

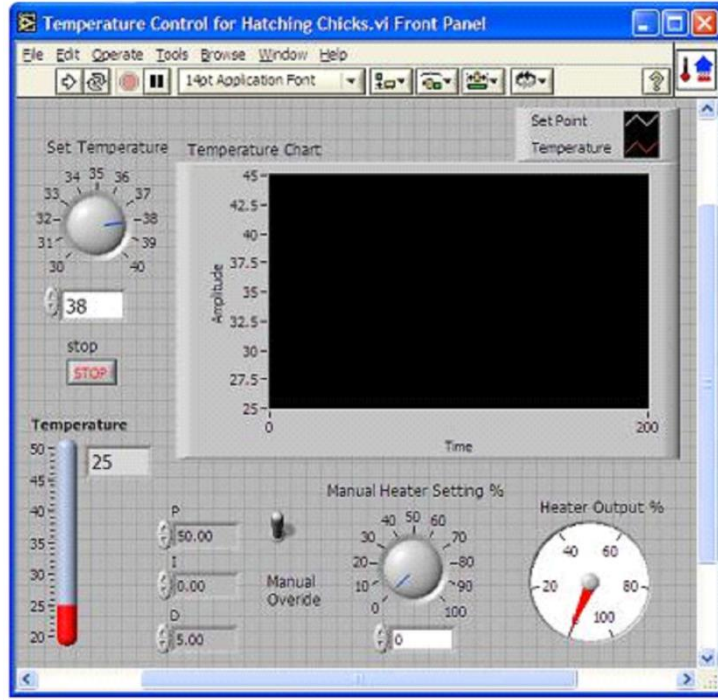
4.2 rnek Uygulamalar

4.2.1 PID ve LabVIEW kullanarak Sıcaklık Kontrol



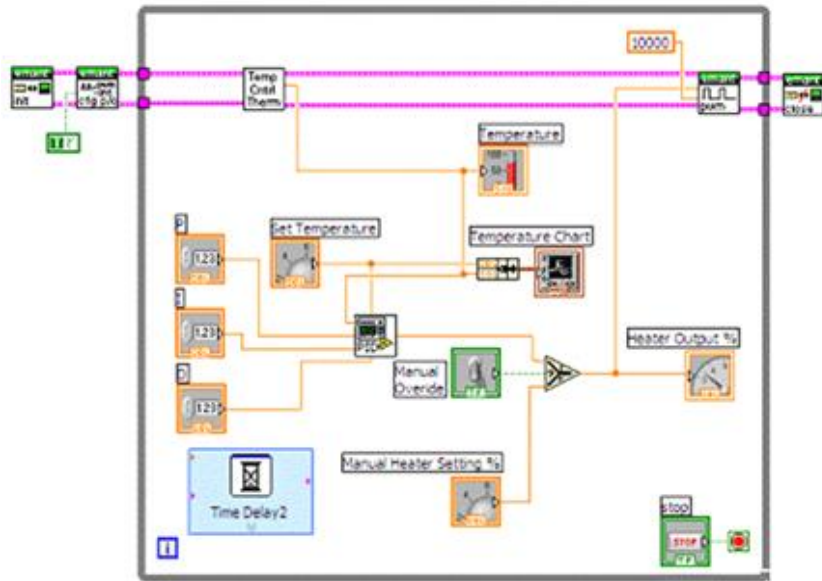
řekil 4. 13 Sıcaklık kontrol diyagramı

Burada tavuk retim iftliđinde daha verimli kuluka sresi iin sıcaklıđı 37,7° ve 38,3° derece arasında tutabilecek bir sistem tasarlanmaktadır. Sıcaklık lm ve kvetin sıcaklıđını artırma iřlemi 22 Ω , 3W ılık direnle yapılmıřtır. Burada PID kontrolr kullanılmıřtır. LabVIEW front paneli ařađıdaki gibidir.



Şekil 4. 14 Ön panel

LabVIEW Block Diyagramı

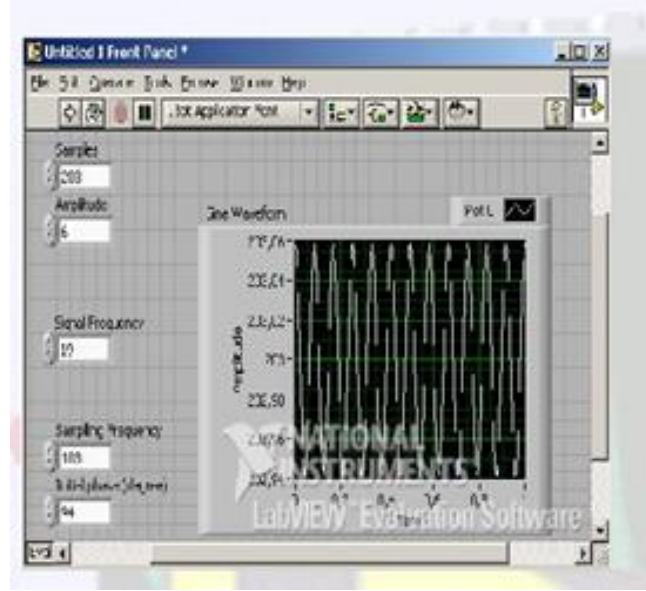


Şekil 4. 15 Blok diyagram

VI termistör tarafından ölçülen sıcaklığı celsiusa çevirir. Değer ölçüldükten sonra PID çıkışını değiştirmek için set edilen değerle karşılaştırılır. PID PWM çıkışını sürmek için çıkışı 0 ve %100 arasındadır. VI aynı zamanda ısıtıcının çıkışını manuel kontrol etmemiz için de kullanılabilir. VI'yı başlatmak için run butonuna basmamız gerekmektedir. PWM

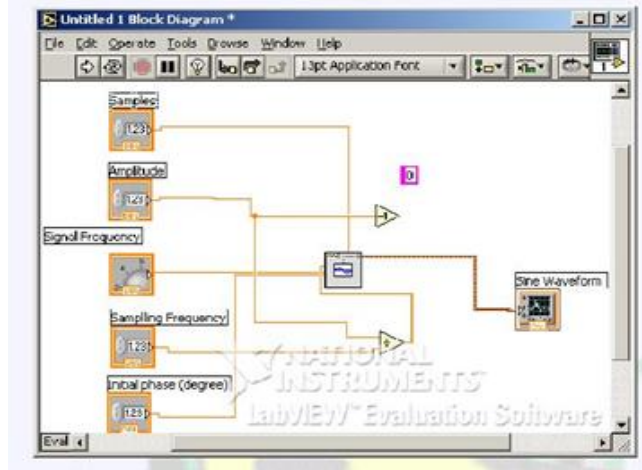
'i 0'a ayarlanır ve ölçülen sıcaklığa T_{amb} denir, daha sonra PWM'i 100'e ayarlanır ölçülen sıcaklığa T_{max} denilir. Okunan değer gözlemlendikten sonra eğer doğru değer veriyorsa ısıtıcı PID kontrolüne bırakılır. VI'yı durdurmak için STOP a basılır.

4.2.2 Değişken Frekanslı Bir Sinüs Dalgası Oluşturmak



Şekil 4. 16 Ön panel

Öncelikle yukarıdaki gibi bir panel düzenlenmesi gerekmektedir. Ön panel kaç tane örnekleme noktası oluşturulacağını, genliği, analog frekansı, oluşturulacak sinüs dalgasının başlangıç fazını ve bu sinyalin hangi frekansta örnekleneceğini girebileceğiniz kontroller içermektedir. Sinyal frekansı üzerine gelip sağ tıklandığında açılan kısa yol menüsünden Replace seçeneğini işaretlenmesi gerekir. Açılan paletten Numeric Control alt panelini açarak, sayısal kontrolü dönel bir sayısal düğmeye çevrilmesi gerekmektedir. Etiketleme aracını kullanarak dönel düğmenin maksimum değeri 100 girilir. Ctrl ve E tuşları ile veya Windows menüsünden "Show Block Diagram" seçilerek blok diyagramını görüntülenmelidir.



Şekil 4. 17 Blok diyagram

Blok diyagrama yukarıdaki fonksiyonları ekleyerek, yapılan bağlantıları incelenir. **Sine Wave VI**'ı çift tıkladığında bu VI'nın iç yapısı görülebilir. Sine wave VI All Function paletinde Analyze>> Waveform Generation alt paletlerinden bulunabilir. Bu örnekte enstrüman 100 Hz. 'de örneklenen 10 Hz. Frekansında bir sinüs dalgasının 200 noktasını oluşturmaktadır. Sinüs dalgası sanal enstrümanına giriş yapmadan önce, sinyal frekansının örnekleme frekansına bölüldüğü gözlemlenir. Bunun sebebi sinüs dalgası enstrümanı sinyalinin sayısal frekansına ihtiyaç duymasındır. Tekrar ön panele dönerek, Run Continuous butonuna basarak sanal enstrümanı sürekli çalıştırdığımızda frekans düğmesini çevirerek sinüzoidal dalgasının nasıl değiştiğini gözlemleyebiliriz.

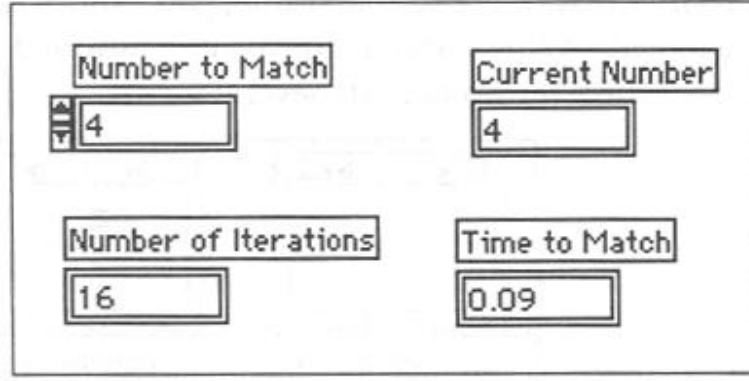
4.2.3 Sayıları Karşılaştırma

Sizin özelleştirdiğiniz sayı ile rasgele seçilmiş bir sayıyı karşılaştırma zamanını hesaplayan bir VI kurunuz.

Ön Panel

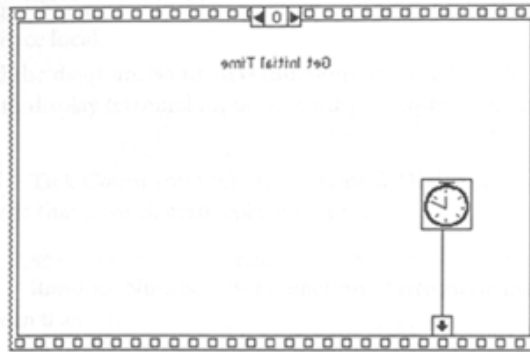
Yeni bir ön panel açın.

Ön paneli gösterilen şekildeki gibi kurun.

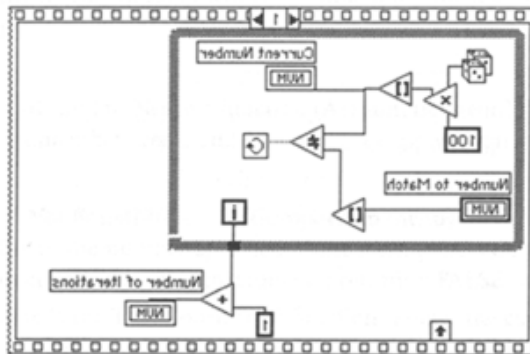


Şekil 4. 18 Ön panel

1. Karşılaştırma için sayı, cari sayı ve döngü sayısı 0 olan kararlılığı pop – up menüsünden Format&Precision'u seçerek değiştirin. Ondalık noktanın sağ tarafına rakam gösterilmeyecek şekilde karar rakamları için sıfırı giriniz.
2. Diyagram penceresini açarak şekildeki gibi kurun. Dizi yapısının üç ayrı çerçevesini kurmalısınız.



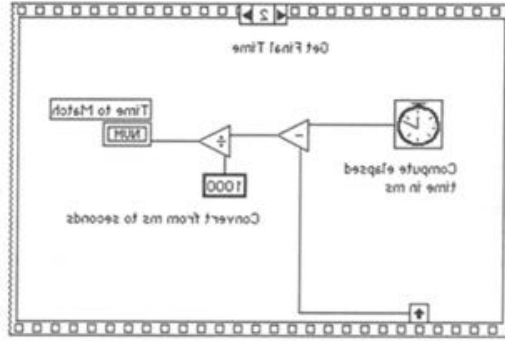
Frame 0



Frame 1

Şekil 4. 19 Blok diyagram

2.



Şekil 4. 20 Blok diyagram

3. Diyagram penceresinden dizi yapısını (Structs&Constants menüsü) yerleştirin. Pozisyon aletleri yardımıyla yapıyı bir köşesinden büyütün. Pop - up menüsünden Add Frame After seçerek çerçeve kenarından yeni çerçeve oluşturun.

4. Pop-up menüsünden Add Sequence Local ile 0. çerçevenin alt sınırından yerel dizi oluşturun. Yerel dizi boş kare olarak ortaya çıkar.

5. Diyagramı kurun. Yeni fonksiyonlar aşağıda sıralanmıştır. Girdi ve çıktılar terminalin çalışmasında yardım penceresinden kullanımından emin olun.



Tick Count (ms) fonksiyonu (**Time&Dialog** menüsü). Çalıştırıldığında itibaren geçen süreleri verir.



Random Number (0–1) fonksiyonu (**Arithmetic** Menüsü). 0 ile 1 arası rasgele sayı verir.



Multiply fonksiyonu (**Arithmetic** menüsü). Burada, rasgele olarak 100 ile çarpılır. Yani, 0,0 ile 100,0 arasında rasgele sayı verir.



Round to Nearest fonksiyonu (**Arithmetic** menüsü). 0 ile 100 arası rasgele olarak en yakın tam sayıya tamamlar.



Not Equal fonksiyonu (**Comparison** menüsü). Rasgele sayı ile özelleştirilmiş sayıyı ön panelde karşılaştırır ve eşit değilse TRUE (Doğru) ve eşitse FALSE (Yanlış) verir.

Çerçevede, **Tick Count (ms)** fonksiyonu cari zamanı milisaniyelerle verir. 1. çerçevede, VI While döngüsü özel ve **Random Number (0-1)** karşılaştırmayan fonksiyon olarak çalışır. 2. çerçevede, **Tick Count (ms)** yeni zamanı milisaniyelerle verir. VI geçen zamanı hesaplamak için yeni zamandan eski zamanı (0. çerçeveden dizi boyunca geçen) çıkarır.

6. Ön panele geri döner ve  tıklamayla açılır. Bu düğmenin çalışması için çalışır vaziyette olmalısınız. Işığı açmak, ön panelde elde edilmiş sayıları görebilmek için yavaşlatır.

7. VI kontrol ve çalışması için karşılaştırma sayısındaki sayıya girin. Hız kazandırmak için, çalışma ışığını  söndürebilirsiniz.

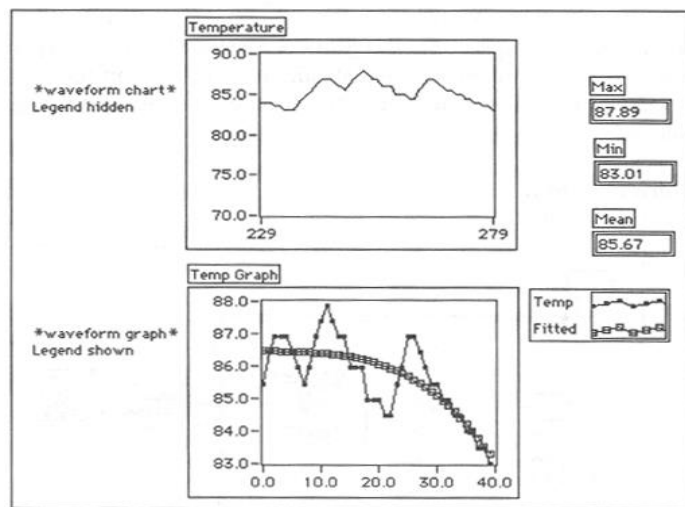
8. **Save** kullanarak MYWORK.LLB'de **Time to Match.vi** olarak saklayın ve VI'yı kapatın.

4.2.4 Sıcaklık Analizi

10 saniye içinde her 0,25 saniyede sıcaklık ölçen bir VI oluşturacaksınız. Toplama süresince, VI bir dalga formu tablosunun üzerindeki gerçek zamandaki ölçümleri gösterir. Toplama tamamlandıktan sonra, VI veriyi ile bir grafik üzerindeki en iyi uyan eğriyi çizer ve minimum, maksimum ve ortalama sıcaklıkları hesaplar.

Ön Panel

1. Yeni bir panel açın ve gösterilen paneli çizin.



Şekil 4. 21 Ön panel

2. 70,0'dan 90,0'a uzanacak şekilde tabloyu tekrar ölçeklendirin. Aynı zamanda, ölçeğinizin grafiğin her iki ekseninde olduğundan da emin olun.

Sıcaklık tablosu, sıcaklığı alındığı gibi gösterir. Toplama bittikten sonra, VI veriyi ve sıcaklık grafiğinde en iyi uyan eğriyi çizer. Ortalama, maksimum ve minimum sıcaklıkların her birini söylendiği sıraya göre gösterir.

1. Blok diyagramını gösterildiği gibi oluşturun. Bu fonksiyonların girdilerinin ve çıktılarının size gösterilmesi için yardım penceresini kullandığınıza emin olun. Aksi takdirde, büyük olasılıkla yanlış kutupları birbirine bağlarsınız.

Şekil 4. 22 Blok diyagram

 **Thermometer VI** (VI... menüsü veya eğer oluşturamazsanız **Tutorial** paleti). Bir sıcaklık ölçümüne geri döndürür.

 **Wait (ms)** fonksiyonu (**Time&Dialog** menüsü) her 0,25 saniyede (250 ms) yerine getirmek için For döngüsüne neden olur.



Array Max&Min fonksiyonu (**Array&Cluster** menüsü). Maksimum ve minimum sıcaklığı toplama boyunca geri döndürür.



Mean&Standard Deviation VI (**Analysis ► Statistics** menüsü). Sıcaklık ölçmelerini ortalamasını geri döndürür.



Curve Fit VI (**Analysis ► Statistics** menüsü). Sıcaklık dizisine karşı polinoma uygun bir sırayı geri döndürür. Bu alıştırmmanın üçü sıra olarak kullanılır.



Build Array fonksiyonu (**Array&Cluster** menüsü). Sıcaklık dizisinden ve en uygun diziden bir küme oluşturur. Bir köşeyi pozisyon aleti ile çekerek fonksiyon için girdilerin sayısını artırabilirsiniz.

For döngüsü 40 kez gerçekleşir. Wait (ms) fonksiyonu her tekrarın, 250 ms'de olmasına neden olur. VI sıcaklık ölçmelerini For döngüsü sınırlarında oluşturulmuş bir dizinin içerisinde saklar. For döngüsü tamamlandıktan sonra, dizi çeşitli düğümlere uzanır. Array Max&Min fonksiyonu, maksimum ve minimum sıcaklığı geri getirir. Mean VI, ortalama sıcaklık değerlerini geri getirir. Curve Fit VI, sıcaklık dizisindeki noktalar için en uygununu belirler. Build Array fonksiyonu, bir dizi içinde çok planlı grafikler için veri toplar.

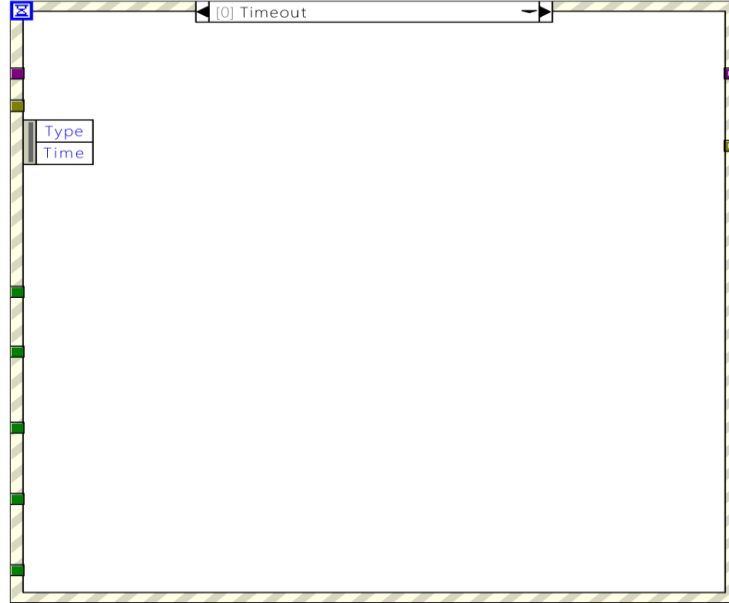
2. Ön panele geri dönün ve VI'yı çalıştırın.

3. VI'yı saklayın ve kapatın. MYWORK.LLB'de Temperature Analysis.vi olarak adlandırın.

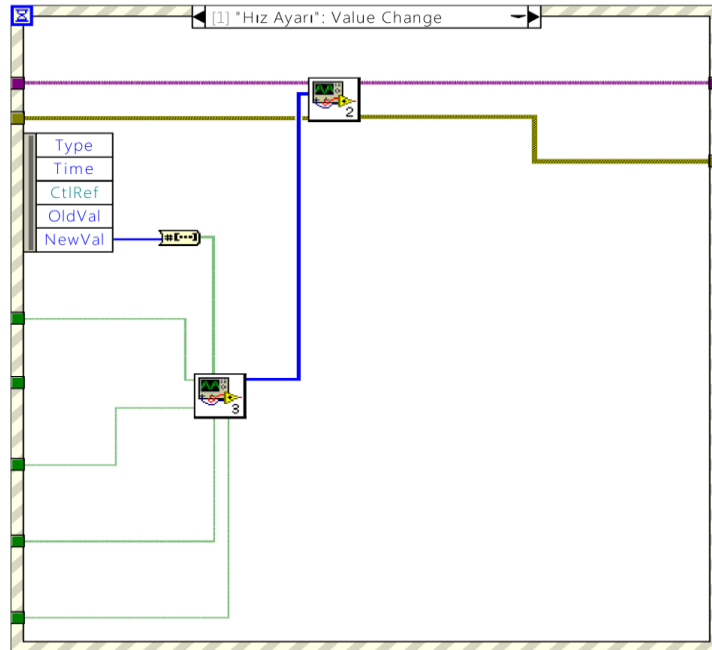
4.3 Gerçekleştirilen Sistemin Kişisel Bilgisayar LabVIEW Yazılımı

Bu bitirme çalışmasında adım motoru PIC18F2520 mikrodenetleyicisi ile kontrol edilmiştir. PIC içindeki yazılım MPLAB IDE v8.36 programıyla hazırlanmıştır ve C dili ile yazılmıştır. Adım motor denetim yazılımında hazırlanan C yazılımına uygun olarak LabVIEW VI'yı dizayn edilmiştir. Projeye ait VI aşağıdaki gibidir.

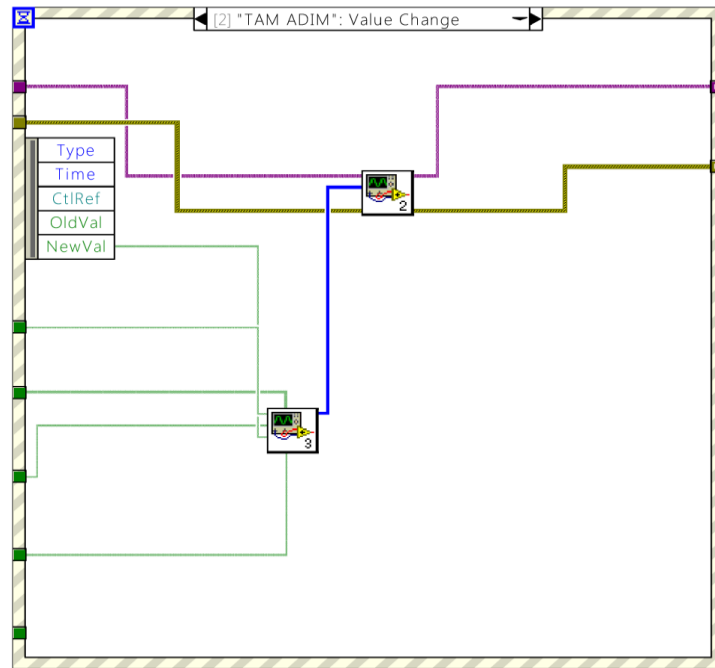
Şekil 4. 23 'de LabVIEW yazılımının ön paneli gösterilmiştir. Ayrıca şekil 4. 24 'de yazılımın blok diyagramı gösterilmiştir ki blok diyagramının alt bölümleri aşağıdaki gibidir.



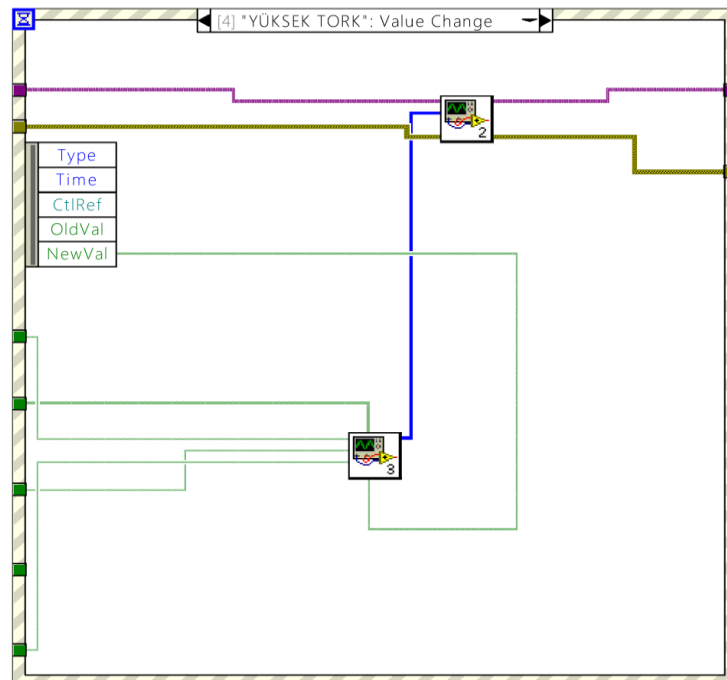
Şekil 4. 25 Blok Diyagram Time Out kısmı



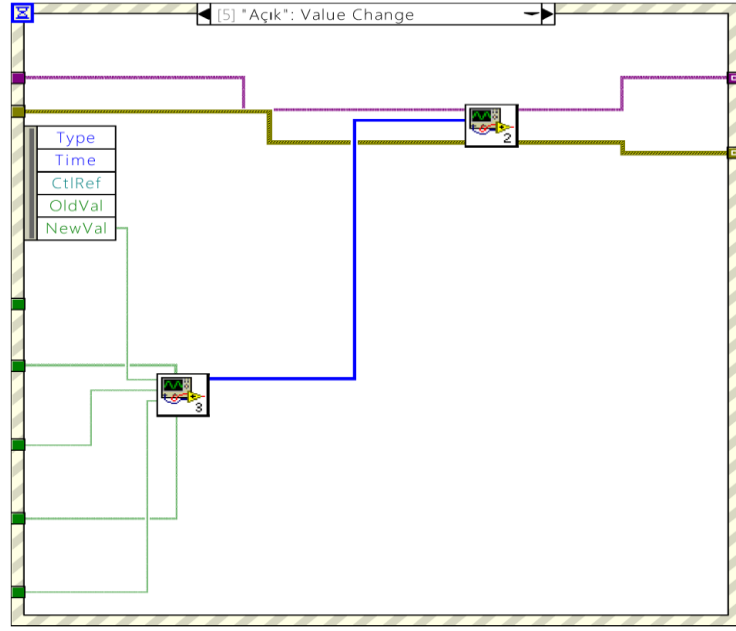
Şekil 4. 26 Blok Diyagram hız ayar kısmı



Şekil 4. 27 Blok Diyagram adım ayar kısmı



Şekil 4. 28 Blok Diyagram moment ayar kısmı



Şekil 4. 29 Blok Diyagram güç ayar kısmı

Motoru sürmek için önce LabVIEW 'da oluşturulan VI 'yün üst kısmında bulunan "çalıştır" düğmesine basarak çalıştırıp sonra ön panelde ayarlanan güç düğmesini "Açık" pozisyonuna getirerek programı devreye sokacağız ki bu aşamada yeşil ışık yanmaya başlayacak ve penceredeki "Veri" bölümünde Start kelimesi yazılacaktır. Bundan sonra ön paneldeki ayar düğmeleri ile oynayarak motorun momenti, Adım tipi, Dönme yönü ve Dönme hızını ayarlaya biliriz.

Her zaman güç düğmesini kapalı konumuna getirirsek de motor duracak ve "Veri" bölümünde Stop kelimesi yazacaktır, ayrıca gücün üstündeki yeşil ışık da sönecektir ve ardından da LabVIEW çalıştır düğmesini devre dışı bırakmalıyız.

4.4 Gerçekleştirilen Sistemin Mikrodenetleyici C Yazılımı

Adım motor denetim donanımında kullanılan PIC mikrodenetleyicisi için C dilinde hazırlanan yazılım aşağıdaki gibidir.

MAIN.c

```
#include "hardware.h"
#include "functions.h"
void main()
{
```

```

initHardware ();
while( true ){
delay_ms(10);
restart_wdt();
}
}

```

18F2520.h

```

////////// Standard Header file for the PIC18F2520 device //////////
#define PIC18F2520
#nolist
////////// Program memory: 16384x16 Data RAM: 1536 Stack: 31
////////// I/O: 25 Analog Pins: 10
////////// Data EEPROM: 256
////////// C Scratch area: 00 ID Location: 200000
////////// Fuses: LP,XT,HS,RC,EC,EC_IO,H4,RC_IO,PROTECT,NOPROTECT
////////// Fuses:
BROWNOUT_NOSL,BROWNOUT_SW,NOBROWNOUT,BROWNOUT,WDT1,WDT2.WDT4
////////// Fuses:
WDT8,WDT16,WDT32.WDT64,WDT128,WDT,NOWDT,BORV20,BORV27,BORV42
////////// Fuses:
BORV45,PUT,NOPUT,CPD,NOCPD,NOSTVREN,STVREN,NODEBUG,DEBUG
////////// Fuses:
NOLVP,LVP,WRT,NOWRT,WRTD,NOWRTD,IESO,NOIESO,FCMEN,NOFCMEN
////////// Fuses:
PBADEN,NOPBADEN,CCP2B3,CCP2C1,WRTC,NOWRTC,WRTB,NOWRTB,EBTR
////////// Fuses:
NOEBTR,EBTRB,NOEBTRB,CPB,NOCPB,LPT1OSC,NOLPT1OSC,MCLR,NOMCLR
////////// Fuses: XINST,NOXINST,INTRC,INTRC_IO,WDT256,WDT512.WDT1024,WDT2048
////////// Fuses: WDT4096,WDT8192.WDT16384,WDT32768
//////////
////////// I/O
// Discrete I/O Functions: SET_TRIS_x(), OUTPUT_x(), INPUT_x(),
// PORT_x_PULLUPS(), INPUT(),
// OUTPUT_LOW(), OUTPUT_HIGH(),

```

```

//          OUTPUT_FLOAT(), OUTPUT_BIT()
// Constants used to identify pins in the above are:
#define PIN_A0 31744
#define PIN_A1 31745
#define PIN_A2 31746
#define PIN_A3 31747
#define PIN_A4 31748
#define PIN_A5 31749
#define PIN_A6 31750
#define PIN_A7 31751
#define PIN_B0 31752
#define PIN_B1 31753
#define PIN_B2 31754
#define PIN_B3 31755
#define PIN_B4 31756
#define PIN_B5 31757
#define PIN_B6 31758
#define PIN_B7 31759
#define PIN_C0 31760
#define PIN_C1 31761
#define PIN_C2 31762
#define PIN_C3 31763
#define PIN_C4 31764
#define PIN_C5 31765
#define PIN_C6 31766
#define PIN_C7 31767
#define PIN_E3 31779
//////////////////////////////////// Useful defines
#define FALSE 0
#define TRUE 1
#define BYTE int8
#define BOOLEAN int1
#define getc getch

```

```

#define fgetc getch
#define getchar getch
#define putc putchar
#define fputc putchar
#define fgets gets
#define fputs puts

//////////////////////////////////// Control
// Control Functions: RESET_CPU(), SLEEP(), RESTART_CAUSE()
// Constants returned from RESTART_CAUSE() are:
#define WDT_TIMEOUT    7
#define MCLR_FROM_SLEEP 11
#define MCLR_FROM_RUN  15
#define NORMAL_POWER_UP 12
#define BROWNOUT_RESTART 14
#define WDT_FROM_SLEEP  3
#define RESET_INSTRUCTION 0

//////////////////////////////////// Timer 0
// Timer 0 (AKA RTCC) Functions: SETUP_COUNTERS() or SETUP_TIMER_0(),
//                               SET_TIMER0() or SET_RTCC(),
//                               GET_TIMER0() or GET_RTCC()
// Constants used for SETUP_TIMER_0() are:
#define RTCC_INTERNAL  0
#define RTCC_EXT_L_TO_H 32
#define RTCC_EXT_H_TO_L 48
#define RTCC_DIV_1    8
#define RTCC_DIV_2    0
#define RTCC_DIV_4    1
#define RTCC_DIV_8    2
#define RTCC_DIV_16   3
#define RTCC_DIV_32   4
#define RTCC_DIV_64   5
#define RTCC_DIV_128  6
#define RTCC_DIV_256  7

```

```

#define RTCC_OFF    0x80
#define RTCC_8_BIT  0x40
// Constants used for SETUP_COUNTERS() are the above
// constants for the 1st param and the following for
// the 2nd param:
//////////////////////////////////// WDT
// Watch Dog Timer Functions: SETUP_WDT() or SETUP_COUNTERS() (see above)
//          RESTART_WDT()
// WDT base is 4ms
//
#define WDT_ON      0x100
#define WDT_OFF     0

//////////////////////////////////// Timer 1
// Timer 1 Functions: SETUP_TIMER_1, GET_TIMER1, SET_TIMER1
// Constants used for SETUP_TIMER_1() are:
// (or (via |) together constants from each group)
#define T1_DISABLED    0
#define T1_INTERNAL    0x85
#define T1_EXTERNAL    0x87
#define T1_EXTERNAL_SYNC 0x83
#define T1_CLK_OUT      8
#define T1_DIV_BY_1     0
#define T1_DIV_BY_2     0x10
#define T1_DIV_BY_4     0x20
#define T1_DIV_BY_8     0x30
//////////////////////////////////// Timer 2
// Timer 2 Functions: SETUP_TIMER_2. GET_TIMER2. SET_TIMER2
// Constants used for SETUP_TIMER_2() are:
#define T2_DISABLED    0
#define T2_DIV_BY_1    4
#define T2_DIV_BY_4    5
#define T2_DIV_BY_16   6

```

```

//////////////////////////////////// Timer 3
// Timer 3 Functions: SETUP_TIMER_3, GET_TIMER3, SET_TIMER3
// Constants used for SETUP_TIMER_3() are:
// (or (via |) together constants from each group)
#define T3_DISABLED      0
#define T3_INTERNAL      0x85
#define T3_EXTERNAL      0x87
#define T3_EXTERNAL_SYNC 0x83
#define T3_DIV_BY_1      0
#define T3_DIV_BY_2      0x10
#define T3_DIV_BY_4      0x20
#define T3_DIV_BY_8      0x30

//////////////////////////////////// CCP
// CCP Functions: SETUP_CCPx, SET_PWMx_DUTY
// CCP Variables: CCP_x, CCP_x_LOW, CCP_x_HIGH
// Constants used for SETUP_CCPx() are:
#define CCP_OFF          0
#define CCP_CAPTURE_FE    4
#define CCP_CAPTURE_RE    5
#define CCP_CAPTURE_DIV_4  6
#define CCP_CAPTURE_DIV_16 7
#define CCP_COMPARE_SET_ON_MATCH 8
#define CCP_COMPARE_CLR_ON_MATCH 9
#define CCP_COMPARE_INT    0xA
#define CCP_COMPARE_INT_AND_TOGGLE 0x2
#define CCP_COMPARE_RESET_TIMER 0xB
#define CCP_PWM            0xC
#define CCP_PWM_PLUS_1     0x1c
#define CCP_PWM_PLUS_2     0x2c
#define CCP_PWM_PLUS_3     0x3c
#define CCP_USE_TIMER3     0x100
long CCP_1;

```

```

#byte CCP_1 =          0xfbe
#byte CCP_1_LOW=        0xfbe
#byte CCP_1_HIGH=       0xfb
long CCP_2;
#byte CCP_2 =          0xfbb
#byte CCP_2_LOW=        0xfbb
#byte CCP_2_HIGH=       0xfbc

//////////////////////////////////// SPI
// SPI Functions: SETUP_SPI, SPI_WRITE, SPI_READ, SPI_DATA_IN
// Constants used in SETUP_SPI() are:
#define SPI_MASTER      0x20
#define SPI_SLAVE       0x24
#define SPI_L_TO_H      0
#define SPI_H_TO_L      0x10
#define SPI_CLK_DIV_4    0
#define SPI_CLK_DIV_16   1
#define SPI_CLK_DIV_64   2
#define SPI_CLK_T2       3
#define SPI_SS_DISABLED  1
#define SPI_SAMPLE_AT_END 0x8000
#define SPI_XMIT_L_TO_H  0x4000

//////////////////////////////////// UART
// Constants used in setup_uart() are:
// FALSE - Turn UART off
// TRUE  - Turn UART on
#define UART_ADDRESS      2
#define UART_DATA         4
#define UART_AUTODETECT   8
#define UART_AUTODETECT_NOWAIT 9
#define UART_WAKEUP_ON_RDA 10
#define UART_SEND_BREAK   13

//////////////////////////////////// COMP
// Comparator Variables: C1OUT, C2OUT

```

```

// Constants used in setup_comparator() are:
#define A0_A3_A1_A3 0xffff04
#define A0_A3_A1_A2_OUT_ON_A4_A5 0xfcf03
#define A0_A3_A1_A3_OUT_ON_A4_A5 0xbc05
#define NC_NC_NC_NC 0x0ff07
#define A0_A3_A1_A2 0xffff02
#define A0_A3_NC_NC_OUT_ON_A4 0x9ef01
#define A0_VR_A1_VR 0x3ff06
#define A3_VR_A2_VR 0xcff0e
#define CP1_INVERT 0x0000010
#define CP2_INVERT 0x0000020
#define C1OUT = 0xfb4.6
#define C2OUT = 0xfb4.7

//////////////////////////////////// VREF
// Constants used in setup_vref() are:
//
#define VREF_LOW 0xa0
#define VREF_HIGH 0x80
// Or (with |) the above with a number 0-15
#define VREF_F5 0x40
#define VREF_COMP 0x10
//////////////////////////////////// VREF
// Constants used in setup_low_volt_detect() are:
//
#define LVD_LVDIN 0x1F
#define LVD_45 0x1E
#define LVD_42 0x1D
#define LVD_40 0x1C
#define LVD_38 0x1B
#define LVD_36 0x1A
#define LVD_35 0x19
#define LVD_33 0x18

```

```

#define LVD_30 0x17
#define LVD_28 0x16
#define LVD_27 0x15
#define LVD_25 0x14
#define LVD_24 0x13
#define LVD_23 0x12
#define LVD_22 0x11
#define LVD_21 0x10
#define LVD_TRIGGER_BELOW 0
#define LVD_TRIGGER_ABOVE 0x80
//////////////////////////////////// INTERNAL RC
// Constants used in setup_oscillator() are:
// First param:
#define OSC_31KHZ 0
#define OSC_125KHZ 0x10
#define OSC_250KHZ 0x20
#define OSC_500KHZ 0x30
#define OSC_1MHZ 0x40
#define OSC_2MHZ 0x50
#define OSC_4MHZ 0x60
#define OSC_8MHZ 0x70
#define OSC_16MHZ 0x4060
#define OSC_32MHZ 0x4070
// The following may be OR'ed in with the above using |
#define OSC_TIMER1 1
#define OSC_INTRC 2
#define OSC_NORMAL 0
// The following may be OR'ed in with the above using |
#define OSC_IDLE_MODE 0x80
#define OSC_31250 0x8000
#define OSC_PLL_ON 0x4000
#define OSC_PLL_OFF 0
// A second optional parameter may be used with this part to fine

```

```

// tune the speed (signed int,0-31)
// Result may be (ignore all other bits)
#define OSC_STATE_STABLE 4
#define OSC_STATE_EXT_RUNNING 8
////////////////////////////////////////// ADC
// ADC Functions: SETUP_ADC(), SETUP_ADC_PORTS() (aka SETUP_PORT_A),
//      SET_ADC_CHANNEL(), READ_ADC()
// Constants used for SETUP_ADC() are:
#define ADC_OFF      0      // ADC Off
#define ADC_CLOCK_DIV_2  0x100
#define ADC_CLOCK_DIV_4  0x04
#define ADC_CLOCK_DIV_8  0x01
#define ADC_CLOCK_DIV_16 0x05
#define ADC_CLOCK_DIV_32 0x02
#define ADC_CLOCK_DIV_64 0x06
#define ADC_CLOCK_INTERNAL 0x07      // Internal 2-6us
// The following may be OR'ed in with the above using |
#define ADC_TAD_MUL_0    0x00
#define ADC_TAD_MUL_2    0x08
#define ADC_TAD_MUL_4    0x10
#define ADC_TAD_MUL_6    0x18
#define ADC_TAD_MUL_8    0x20
#define ADC_TAD_MUL_12   0x28
#define ADC_TAD_MUL_16   0x30
#define ADC_TAD_MUL_20   0x38
// Constants used in SETUP_ADC_PORTS() are:
#define NO_ANALOGS  0x0F  // None
#define ALL_ANALOG  0x00  // A0 A1 A2 A3 A5 B2 B3 B1 B4 B0
#define AN0_TO_AN11 0x03  // A0 A1 A2 A3 A5 B2 B3 B1 B4
#define AN0_TO_AN10 0x04  // A0 A1 A2 A3 A5 B2 B3 B1
#define AN0_TO_AN9  0x05  // A0 A1 A2 A3 A5 B2 B3
#define AN0_TO_AN8  0x06  // A0 A1 A2 A3 A5 B2
#define AN0_TO_AN4  0x0A  // A0 A1 A2 A3 A5

```

```

#define AN0_TO_AN3 0x0B // A0 A1 A2 A3
#define AN0_TO_AN2 0x0C // A0 A1 A2
#define AN0_TO_AN1 0x0D // A0 A1
#define AN0 0x0E // A0
#define AN0_TO_AN11_ANALOG 0x03 //!old only provided for compatibility
#define AN0_TO_AN10_ANALOG 0x04 //!old only provided for compatibility
#define AN0_TO_AN9_ANALOG 0x05 //!old only provided for compatibility
#define AN0_TO_AN8_ANALOG 0x06 //!old only provided for compatibility
#define AN0_TO_AN4_ANALOG 0x0A //!old only provided for compatibility
#define AN0_TO_AN3_ANALOG 0x0B //!old only provided for compatibility
#define AN0_TO_AN2_ANALOG 0x0C //!old only provided for compatibility
#define AN0_TO_AN1_ANALOG 0x0D //!old only provided for compatibility
#define AN0_ANALOG 0x0E //!old only provided for compatibility

// The following may be OR'ed in with the above using |
#define VSS_VDD 0x00 // Range 0-Vdd
#define VREF_VREF 0x30 // Range VrefL-VrefH
#define VREF_VDD 0x20 // Range VrefL-Vdd
#define VSS_VREF 0x10 // Range 0-VrefH
// Constants used in READ_ADC() are:
#define ADC_START_AND_READ 7 // This is the default if nothing is specified
#define ADC_START_ONLY 1
#define ADC_READ_ONLY 6
////////////////////////////////////// INT
// Interrupt Functions: ENABLE_INTERRUPTS(), DISABLE_INTERRUPTS(),
// CLEAR_INTERRUPT(), INTERRUPT_ACTIVE(),
// EXT_INT_EDGE()
//
// Constants used in EXT_INT_EDGE() are:
#define L_TO_H 0x40
#define H_TO_L 0
// Constants used in ENABLE/DISABLE_INTERRUPTS() are:
#define GLOBAL 0xF2C0

```

```

#define INT_RTCC          0xF220
#define INT_TIMER0        0xF220
#define INT_TIMER1        0x9D01
#define INT_TIMER2        0x9D02
#define INT_TIMER3        0xA002
#define INT_EXT           0xF210
#define INT_EXT1          0xF008
#define INT_EXT2          0xF010
#define INT_RB            0xFFF208
#define INT_AD            0x9D40
#define INT_RDA           0x9D20
#define INT_TBE           0x9D10
#define INT_SSP           0x9D08
#define INT_CCP1          0x9D04
#define INT_CCP2          0xA001
#define INT_BUSCOL        0xA008
#define INT_LOWVOLT       0xA004
#define INT_COMP          0xA040
#define INT_EEPROM        0xA010
#define INT_OSCF          0xA080
#list

```

HARDWARE.h

```

#ifndef _HARDWARE_
#define _HARDWARE_
#include "18f2520.h"
#use delay(clock=4000000,restart_wdt)
#fuses INTRC_IO,NOMCLR //,WDT
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7,parity=n,DISABLE_INTS)
#include "defines.h"
struct _port_a{
int engine:4;
int not_used:4;
};

```

```

struct _port_a PORTA;
#ifdef __pch__
#locate PORTA = 0xf80
#else
#locate PORTA = 5
#endif

int1 timer1status = FALSE;
void startTimer1 (int16 value)
{
    set_timer1(value);
    if ( timer1status EQ FALSE ) {
        timer1status = TRUE;
        enable_interrupts(INT_TIMER1);
    }
}

void stopTimer1()
{
    if ( timer1status EQ TRUE ) {
        timer1status = FALSE;
        disable_interrupts(INT_TIMER1);
    }
}
#endif

```

FUNCTIONS.h

```

#ifndef _FUNCTIONS_
#define _FUNCTIONS_
#include "hardware.h"
#include "defines.h"
#include "engine_driver.h"
/*
mode speed16 target16
mode

```

```

LSB start dircetion  step          HSB
stop
*/
#include <avr/io.h>
void initHardware ()
{
    setup_oscillator( OSC_4MHZ );
    enable_interrupts(INT_RDA);
    disable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);
    setup_timer_1(T1_INTERNAL | T1_DIV_BY_1 );
    set_timer1(0);//periot);
    set_tris_a(0);
    rs232_errors = 0;
}

int getch_time()
{
    while ( !kbhit()){
        delay_us(10);
    }
    return getchar();
}

void test_pin_voltage ( int read )
{
    PORTA = read;
}

void test_from_hyper ( int read )
{
    unsigned int16 temp;
    if ( read EQ 0 ) {
        stopEngine();
    }
    else if ( read EQ 1 ) {

```

```

startEngine(DIRECTION_CLOCK,STEP_HALF,0);
}
else if ( read EQ 2 ) {
startEngine(DIRECTION_ANTI_CLOCK,STEP_HALF,0);
}
else if ( read EQ 3 ) {
startEngine(DIRECTION_CLOCK,STEP_FULL,0);
}
else if ( read EQ 4 ) {
startEngine(DIRECTION_ANTI_CLOCK,STEP_FULL,0);
}
else if ( read < 10 ) {
temp = (read - 5 ) * 10000;
startEngine(DIRECTION_ANTI_CLOCK,STEP_FULL,temp);
}
}
int read_numeric ()
{
    int8 read;
    read = getchar ();
    putchar(read);
    read -= 48;
    return read;
}
#inline void read_all ()
{
    int8 command;
    int8 speed;
    command = getchar ();
    speed = getch_time ();
    runEngine(command, speed);
}
#int_rda

```

```

void rs232_readRead()
{
  disable_interrupts(INT_RDA);
  //test_pin_voltage (read_numeric() );
  //test_from_hyper(read_numeric());
  read_all();
  enable_interrupts(INT_RDA);
}
#endif

```

ENGINE_DRIVE

```

#ifndef _STEP_ENGINE_
#define _STEP_ENGINE_
#define POWER_DOWN 0
#define POWER_UP 1
#define DIRECTION_ANTI_CLOCK 0
#define DIRECTION_CLOCK 1
#define STATUS_IDLE 0
#define STATUS_RUN 1
#define STEP_HALF 0
#define STEP_FULL 1
#define MOMENT_NORMAL 0
#define MOMENT_DOUBLE 1
#define MAX_SPEED 440
#define MIN_SPEED 15
int steps[8] = {1,3,2.6,4,12.8,9};
int step_size = 8;
int1 direction = DIRECTION_CLOCK;
int step_index = 0;
int1 step_type = STEP_HALF;
int16 timeout = 0;
int1 moment = MOMENT_NORMAL;
int16 wait_time = 0;
void stopEngine();

```

```

inline int get_inc_step()
{
    int value;
    if ( step_index EQ step_size ) {
        step_index = 0;
    }
    if ( step_type EQ STEP_FULL ) {
        if ( moment EQ MOMENT_NORMAL ) {
            value = steps[step_index];
            step_index++;
        }
        else {
            step_index++;
            value = steps[step_index];
        }
    }
    else {
        value = steps[step_index];
    }
    step_index ++;
    return value;
}

inline int get_dec_step ()
{
    signed int value;
    if ( step_index EQ -1 ) {
        step_index = step_size - 1;
    }
    if ( step_type EQ STEP_FULL ) {
        if ( moment EQ MOMENT_DOUBLE ) {
            value = steps[step_index];
            step_index--;
        }
    }
}

```

```

else {
    step_index--;
    value = steps[step_index];
}
}
else {
    value = steps[step_index];
}
if ( step_index EQ -1) {
    step_index = step_size - 2;
}
step_index --;
return value;
}

#inline unsigned int16 calc_timer_load_value(int8 speed)
{
    int16 load[10]={20000,25000,30000,35000,40000,45000,50000,55000,60000,62230};
    int32 temp;
    temp=load[speed];
    return temp;
}

int get_next_step ()
{
    if ( direction EQ DIRECTION_CLOCK ) {
        return get_inc_step();
    }
    else {
        return get_dec_step();
    }
}

void startEngine(int1 dir,int1 step,int16 speed)
{
    stopTimer1();

```

```

direction = dir;
step_type = step;
timeout = speed;
step_index = 0;
startTimer1(timeout);
}
#inline void runEngine(int8 command,int8 speed)
{
int1 status;
int1 k;
k=0;
stopTimer1();
status = command & 0x01;
if ( status EQ POWER_DOWN ) {
stopEngine ();
printf("STOP");
}
else {
if ( speed EQ 0 ) {
stopEngine();
return;
}
printf("START");
direction = (command & 0x02) >> 1;
step_type = (command & 0x04) >> 2;
moment = (command & 0x08) >> 3;
timeout = calc_timer_load_value(speed);
step_index = 0;
PORTA = get_next_step();
startTimer1(timeout);
}
}
void stopEngine()

```

```

{
PORTA.engine = 0;
stopTimer1();
}
#endif
void timer_1_out ()
{
if ( wait_time EQ 0 ) {
set_timer1(timeout);
}
else {
stopTimer1();
delay_ms(wait_time);
startTimer1(timeout);
}
PORTA = get_next_step();
}
#endif

```

Yazılım aşağıda anlatılan kısımlardan oluşmaktadır;

Main.c : Driverların ve portların ilk değerlerini yükler ve sonsuz döngü içinde bekler.

18f2520.h: Bu bölümde sistemde kullanılan PIC mikrodenetleyicisi programda tanıtılmıştır.

Hardware.h : Bu bölüm, PIC 'in standart driverinin üzerine koyulan katmandır.

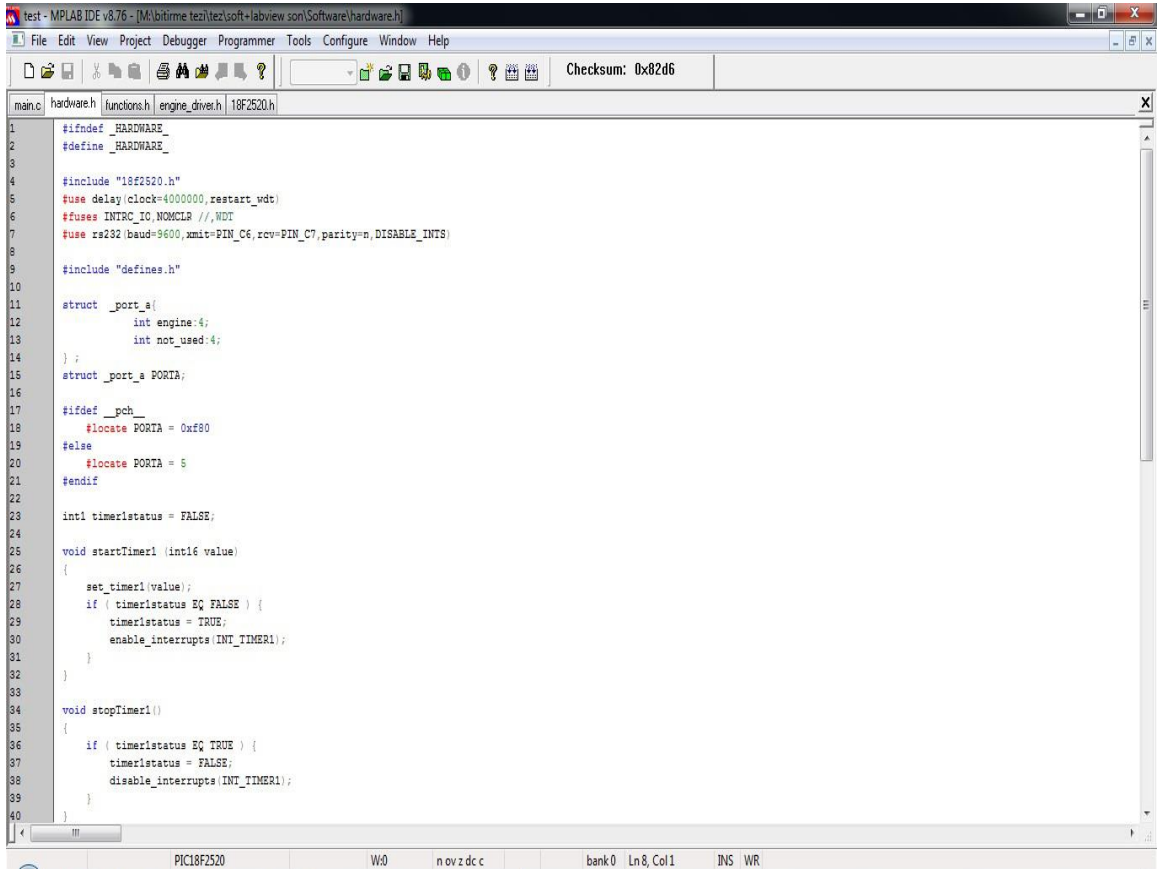
Function.h : Bu kısım, seri portu okur ve sistem için gerekli fonksiyonları tutar.

Engine_Drive: Seri porttan gelen komutları çözer , zaman ayarlamasını yapar ve port değerlerini ayarlar.

Defins : Program içinde kullanılacak tanımlamalar bulunur.

4.4.1 Yazılımın Ekran Görüntüsü ve Akış Diyagramı

Yazılımın ekran görüntüsü aşağıdaki gibidir.



```
1 #ifndef _HARDWARE_
2 #define _HARDWARE_
3
4 #include "18F2520.h"
5 #use delay(clock=4000000, restart_wdt)
6 #fuses INTRC_IO, NOMCLR //, WDT
7 #use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, parity=n, DISABLE_INTS)
8
9 #include "defines.h"
10
11 struct _port_a{
12     int engine:4;
13     int not_used:4;
14 };
15 struct _port_a PORTA;
16
17 #ifdef __pch__
18     #locate PORTA = 0xf80
19 #else
20     #locate PORTA = 5
21 #endif
22
23 int1 timer1status = FALSE;
24
25 void startTimer1 (int16 value)
26 {
27     set_timer1(value);
28     if ( timer1status EQ FALSE ) {
29         timer1status = TRUE;
30         enable_interrupts(INT_TIMER1);
31     }
32 }
33
34 void stopTimer1()
35 {
36     if ( timer1status EQ TRUE ) {
37         timer1status = FALSE;
38         disable_interrupts(INT_TIMER1);
39     }
40 }
```

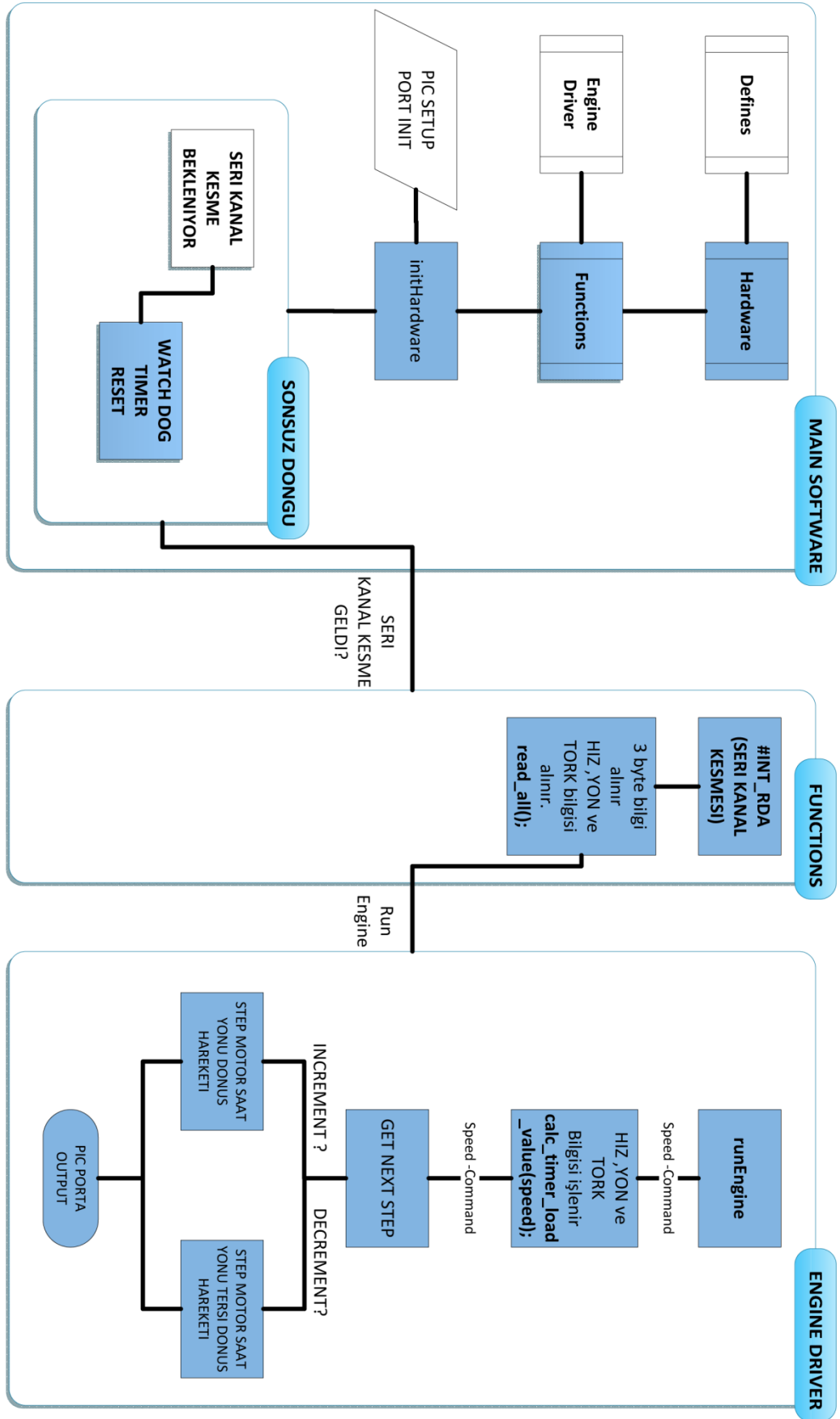
Şekil 4. 30 Yazılımın ekran görüntüsü

Devrenin akış diyagramı şekil 4. 31 'de verilmiştir. Akış diyagramı yazılım algoritmasını üç bölümde göstermektedir.

Main software: entegre donanımının, değişkenlerin, fonksiyonların tanımlandığı ve yazılıma dahil edildiği kısımdır. Bu kısımda sonsuz döngü yazılımı ile entegre bekletilir. İşlemci beklemeyi seri kanal kesmesi ile terk eder ve altprogramlara dallanır.

Functions: Sonsuz döngüde bekleyen entegre seri kanal kesmesi gelmesiyle bu bölüme dallanır ve seri haberleşmeyi gerçekleştirip gelen bilgileri saklar ve motoru sürececek olan üçüncü kısma dallanır.

Engine drive: Seri kanal kesmesinde alınan bilgiler hız, yön, moment bileşenlerini ayırıp işler ve saklar. Daha sonra entegrenin önceden belirtilen port'undan motoru sürececek olan işareti motor sürücüyeye gönderir [16], [19], [33]-[35].



Şekil 4. 31 Adım motor denetim donanımı yazılımının akış diyagramı

SONUÇ VE ÖNERİLER

Günümüzde adım motorları hayatımızın vazgeçilmez bir parçası haline gelmişlerdir. Bilgisayar teknolojisinden örneğin yazıcılardan uzay teknolojisine, kullanılan uydulara kadar kullanım alanları gelişmiştir.

Gerçekleştirilen sistemde LabVIEW programı kontrol amaçlı kullanıldığı için, programın görsel bir ara yüze sahip olmasından dolayı, adım motorunun kontrolü gayet basit ve hızlı bir şekilde yapılabilmektedir. Dolayısıyla bu sistemin çalışma şeklinin anlaşılmasını kolaylaştırır. Bu sistem ile kontrol edilen her hangi bir mekanizmanın çalıştırılması için uzman olmayı gerektirmez. Bilgisayarla çalışmayı bilen birisi kolaylıkla bu sistemle çalışmayı kısa sürede öğrenebilir. İleride ihtiyaç duyulabilecek değişiklikleri grafik programlamayla yapmak daha kolay olur. Örneğin eğer bu sistem, bir sağ-sol dönme hareketi yapan ve belli bir mesafe kat eden bir mekanizmanın kontrolü için kullanılırsa, motorun adım hızı kontrolü ve adım yönünü değiştirmek bizim için yeterli olacaktır.

Bu sistemde adım motoru kullanıldığından dolayı motorun konumu her an tespit edilebilmektedir ve anında yön değişimi uygulanarak motorun gecikmesiz olarak, yön değişimi sağlanmıştır. Ayrıca daha önce de bahsedildiği gibi adım motorların bakıma ihtiyaçları olmadığı için, bu sistem ara vermeden uzun süreler kullanılabilir.

Bu sistemde kullanılan elektronik malzemeler kolaylıkla bulunabilir, üstelik kullanılan mikrodenetleyicinin fiyatı da çok düşüktür.

Sistemin bir diğer başarılı yönü yük altında kolaylıkla devreye girebilmesi için moment ayarının yapılabilmesidir. Ağır yükler altında hatasız devreye girebilmesi için yüksek moment ayarında eş zamanlı olarak çift fazdan sargılar tahrik edilmiştir.

Sistemin bir görsel programla kontrol edilmesi adım motorunun hızlı, hatasız ve güvenilir bir şekilde kontrolünü sağlamaktadır. Bununla beraber, farklı yönlerde lineer veya dairesel olarak değişik hızlarda çalışması gereken bilgisayar kontrollü makinalarda kolaylıkla kullanılabilir.

KAYNAKLAR

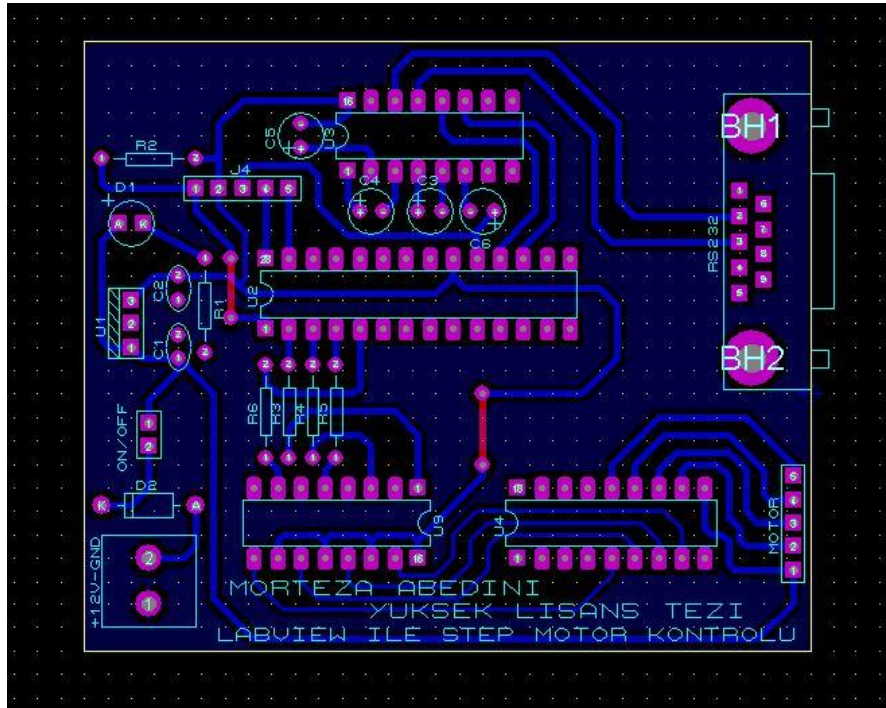
- [1] Altunsaçlı, A., (2003). "Elektrik makineleri", Color Ofset, İskenderun.
- [2] Altunsaçlı, A., (2003). "Elektrik Motorları ve Sürücüleri", Color Ofset, İskenderun.
- [3] Ansiklopedi, Elektrik motoru, http://tr.wikipedia.org/wiki/Elektrik_motoru, 03 Ağustos 2012.
- [4] Sarul M.H., (2012-2013). Güz dönemi ders notları, Elektrik makinalarının kontrolü, Yıldız Teknik Üniversitesi.
- [5] Doğan, İ., (2011). "PIC Adım Motor Kontrolü", Bileşim Yayınları Teknik Kitaplar.
- [6] Fundamentals of Motion Control, <http://www.ni.com/white-paper/3367/en/>, 17 Ocak 2014.
- [7] NI Stepper Motors, <http://sine.ni.com/nips/cds/view/p/lang/en/nid/203934>, 17 Ocak 2014.
- [8] NI Stepper Motors, <http://sine.ni.com/nips/cds/view/p/lang/en/nid/7191>, 17 Ocak 2014.
- [9] NI Stepper Motors, <http://www.lamtechnologies.com/Product.aspx?lng=EN&idp=stpdvckd>, 17 Ocak 2014.
- [10] Easy Driver Stepper Motor Drive, <https://www.sparkfun.com/products/10267>, 17 Ocak 2014.
- [11] Benjamin C. Kuo, (1974). "Theory and applications of step motors", West Publishing Company, College & School Division.
- [12] ACarnley, P., (2007). "Stepping Motors : a guide to theory and practice", 4th Edition, Reprinted.
- [13] Athani, V., (1974). "Stepper Motors : Fundamentals, Applications And Design", New Age International.
- [14] Krause, P. ve Wasynczuk, O. ve Pekarek, S., (2012). "Stepper Motors, Electromechanical Motion Devices Digital Object Identifier" :10.1002/9781118316887.ch9 Psge(s): 415-450 Copyright Year : 2012.

- [15] Industrial Circuits Application Note, Stepper Motor Basics, <http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf>, 09 Ağustos 2012.
- [16] Albert C. L., "Step Motor System Design Handbook", Magna Physics Publishing, 1 Ağu 2001.
- [17] Crnosija, P. ve Kuzmanovic, B. ve Ajdukovic, S., "Microcomputer implementation of optimal algorithms for closed-loop control of hybrid stepper motor drives Industrial Electronics", IEEE, Transactionson Volume: 47 , Issue: 6.
- [18] Types of Stepper Motors (detailed) ,<http://zone.ni.com/devzone/cda/ph/p/id/287>, 03 Ağustos 2012.
- [19] Arango, B. ve Soori, P.K. ve Talukder, P. (2012). "Stepper motor drives for robotic applications", IEEE, Power Engineering and Optimization Conference (PEDCO) Melaka, 2012 IEEE International, Malaysia.
- [20] Teknik Bilgiler Merkezi, Adım Motorları, <http://www.teknomerkez.net/sayfa.php?git=92>, 05 Ağustos 2012.
- [21] Stepper Motors and Control : Part1- Unipolar Stepper Motor and Control , <http://www.stepperworld.com/Tutorials/pgUnipolarTutorial.htm>, 03 Ocak 2013.
- [22] Stepper Motor System Basics,<http://www.ams2000.com/pdf/step101.pdf>, 09 Ağustos 2012.
- [23] PIC18F2420/2520/4420/4520 Data Sheet - Microchip,<http://ww1.microchip.com/downloads/en/devicedoc/39631e.pdf> , 10 Ocak 2013.
- [24] LMAX232. MAX232I (Rev. L) - Texas Instruments, <http://www.ti.com/lit/ds/symlink/max232.pdf>, 10 Ocak 2013.
- [25] ULN2803 ULN2804 Octal High Voltage, High Current Darlington, <http://www.engr.usask.ca/classes/EE/392/DataSheets/ULN2803.pdf>, 10 Ocak 2013.
- [26] Stepping Motors M42SP-7 - Datasheet Catalog,http://pdf.datasheetcatalog.com/datasheets/90/502505_DS.pdf, 10 Ocak 2013.
- [27] Jeffrey T., (2007). "LabVIEW for Everyone: Graphical Programming Made Easy and Fun", 3rd Edition.
- [28] Gray W. ve Jennings, J./R., (2006). "LabVIEW Graphical Programing", 4th Edition, McGraw Hill Professional.
- [29] Stamps, D., (2012). "Learn LabVIEW 2010/2011 Fast: A Primer for Automatic Data Acquisition", SDC Publications.

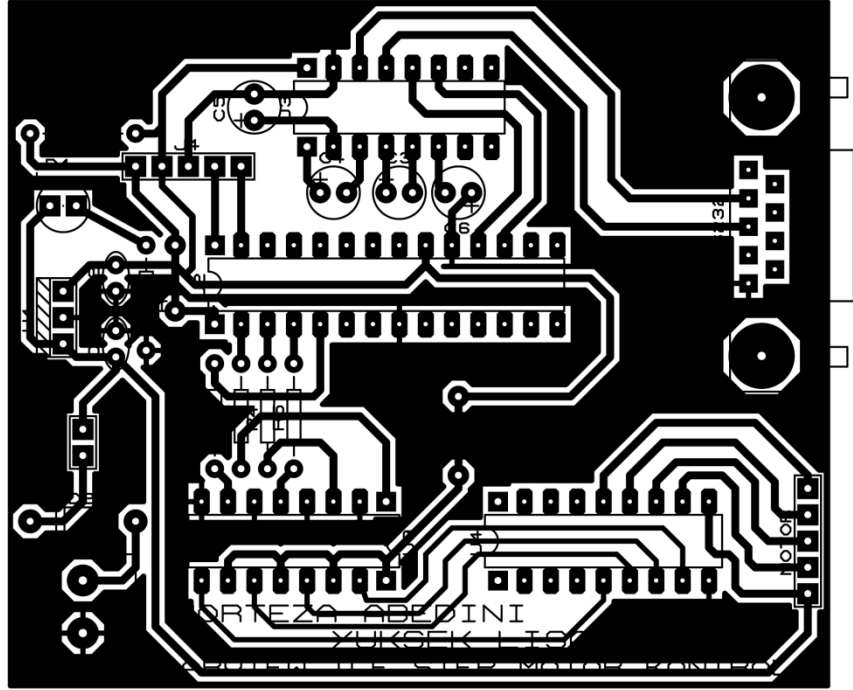
- [30] Elektronik Deney Modüllerinin LabVIEW ile Kontrolü, <http://edergi.sdu.edu.tr/index.php/utbd/article/viewFile/1708/1931.pdf>, 03 Aralık 2013.
- [31] LabVIEW, <http://en.wikipedia.org/wiki/LabVIEW>, 03 Ocak 2013.
- [32] Analog Electronics With LabVIEW, <http://ebookbrowse.net/labview-book-analog-electronics-with-labview-pdf-d104940674>, 03 Ocak 2013.
- [33] Doğan, İ., (2011). "PIC C ile Motor Kontrol Projeleri", Bileşim Yayınları Teknik Kitaplar.
- [34] Stepper Motor Drives, <http://sine.ni.com/nips/cds/view/p/lang/en/nid/3308>, 10 Ocak 2013.
- [35] Uzun, T., (2005). "Mikroişlemci Sistemleri", TOR Ofset.

DEVRENİN PCB ŞEMASI

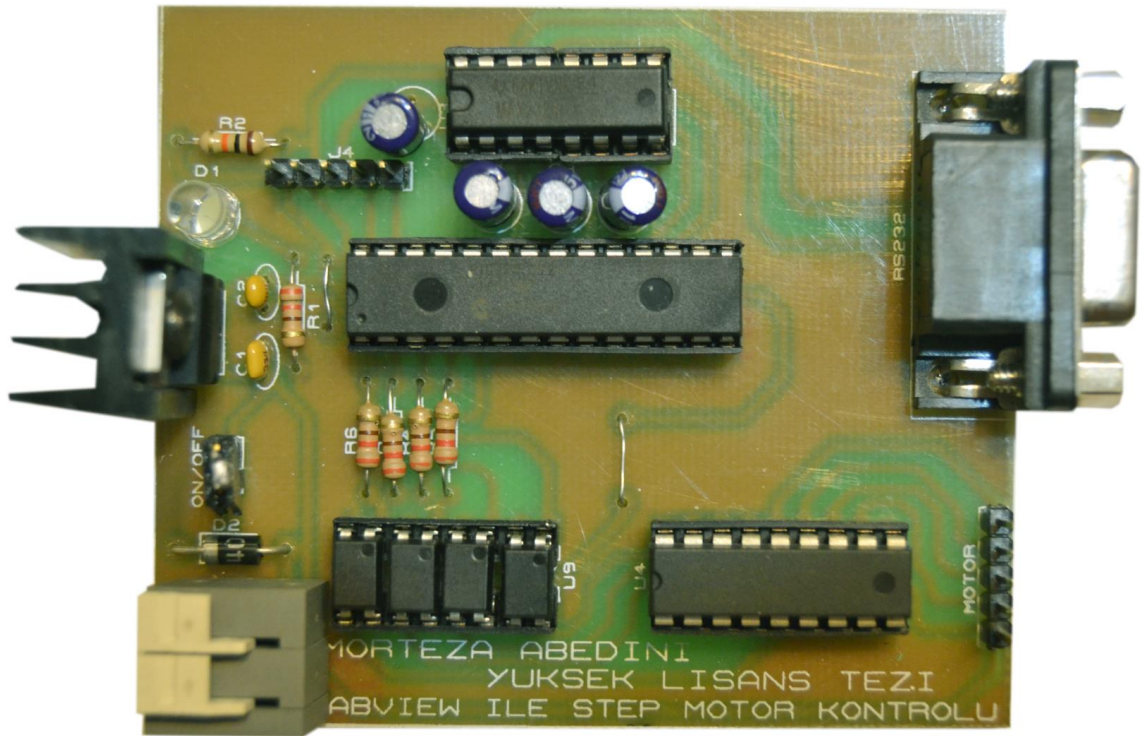
Devrenin PCB şeması aşağıdaki gibidir



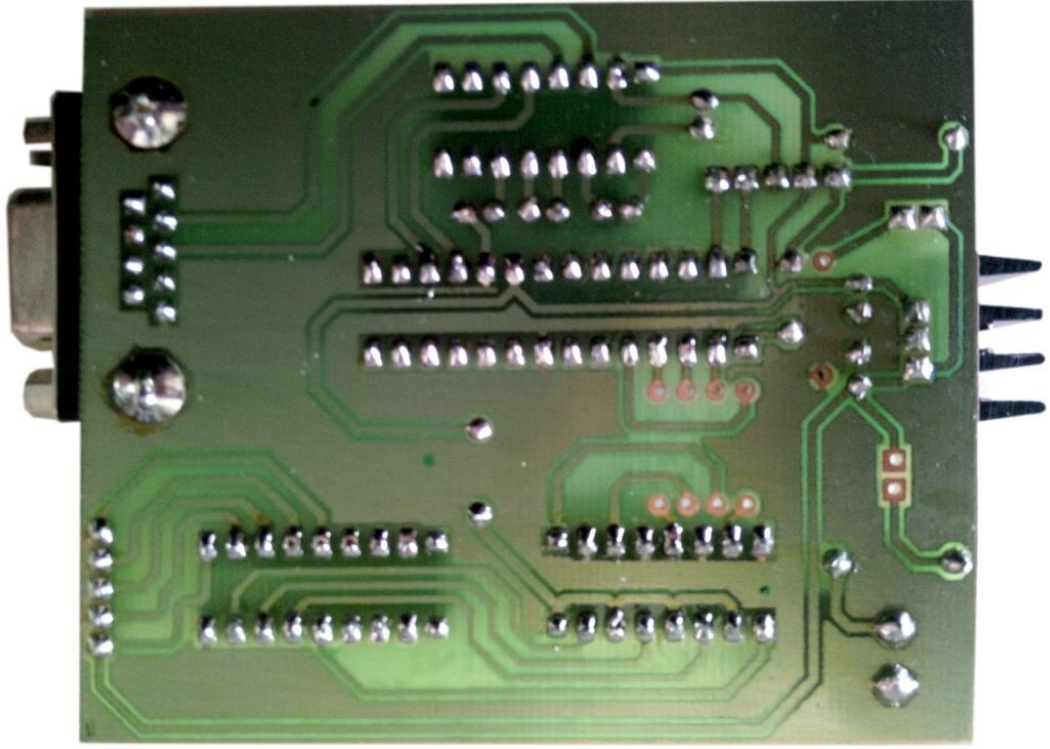
Şekil EK A. 1 Motor sürücü kartının Proteus şeması



Şekil EK A. 2 Motor sürücü kartının PCB devresi



Şekil EK A. 3 PCB üstten görünüm



Şekil EK A. 4 PCB alttan görünüm

PIC18F2520 MİKRODENETLEYİCİ

Bu mikrodnetleyici Microchip firmasının ürünüdür ve veri sayfası(datasheet) aşağıdaki gibidir.



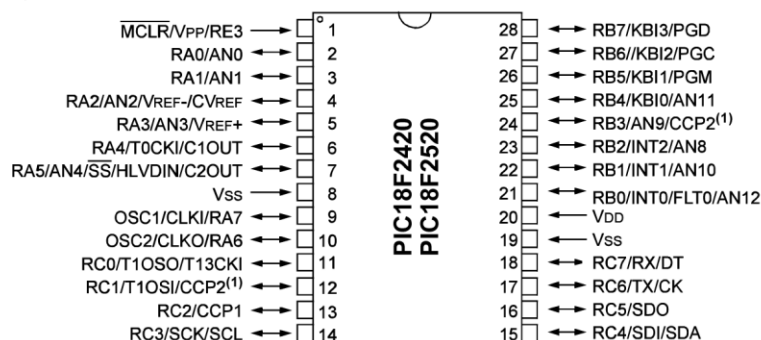
**PIC18F2420/2520/4420/4520
Data Sheet**

28/40/44-Pin Enhanced Flash
Microcontrollers with 10-Bit A/D
and nanoWatt Technology

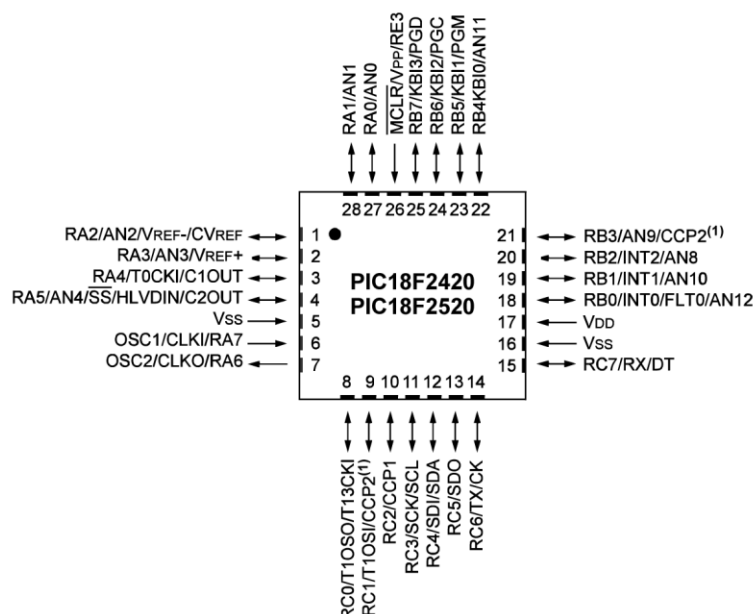
PIC18F2420/2520/4420/4520

Pin Diagrams

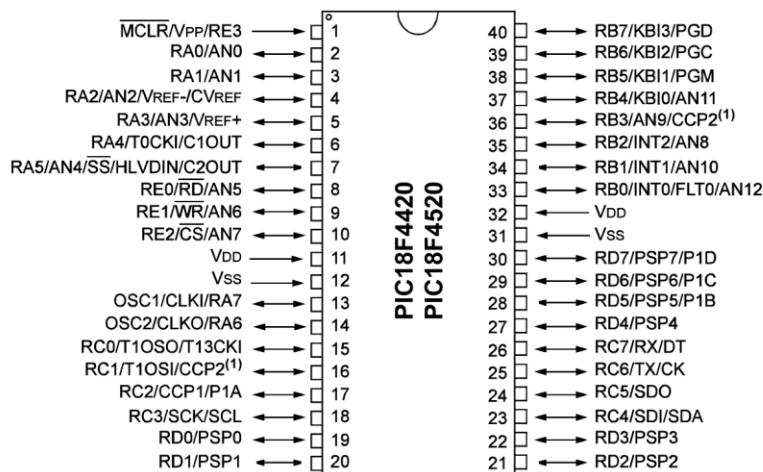
28-Pin SPDIP, SOIC



28-Pin QFN



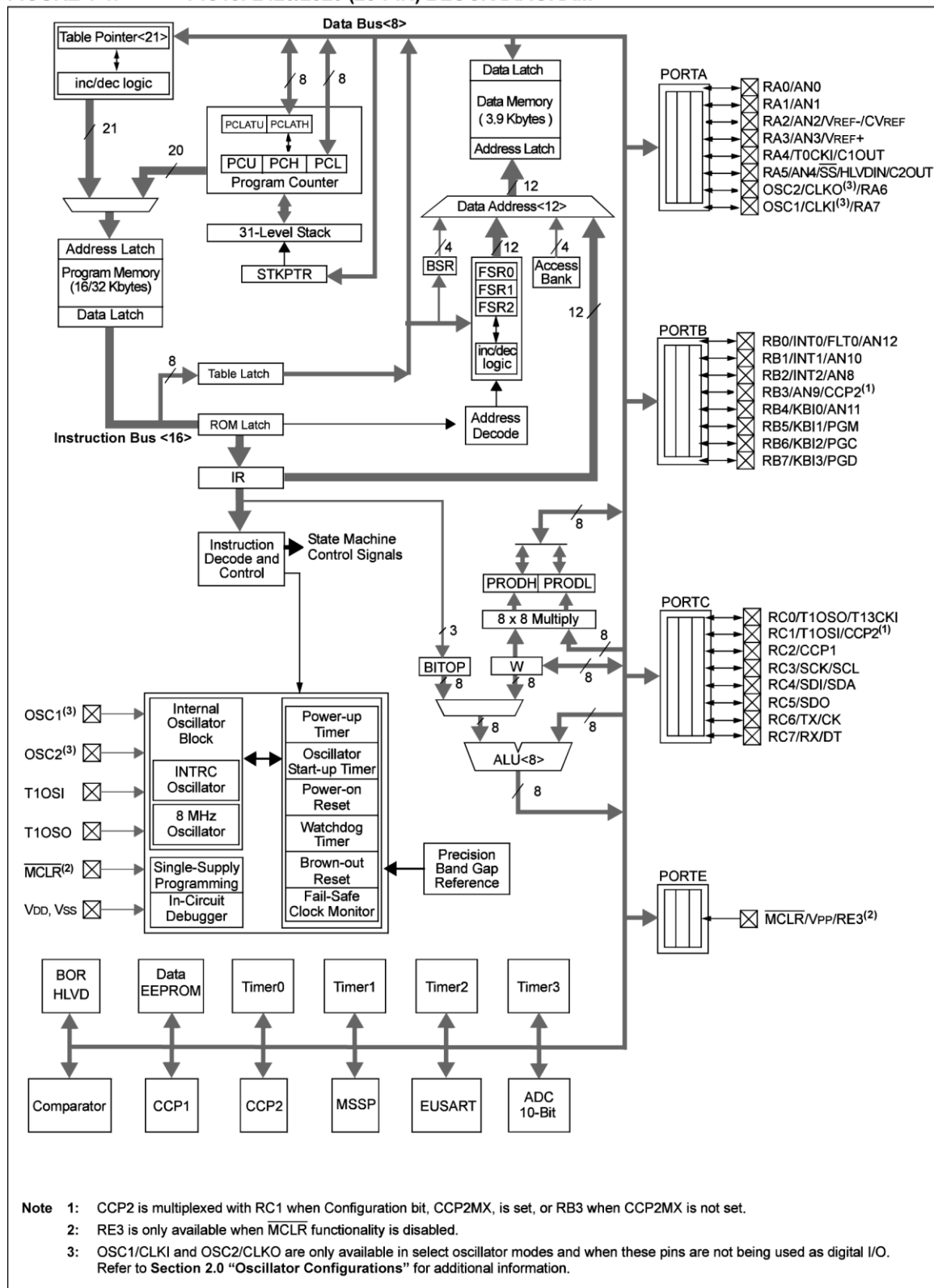
40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

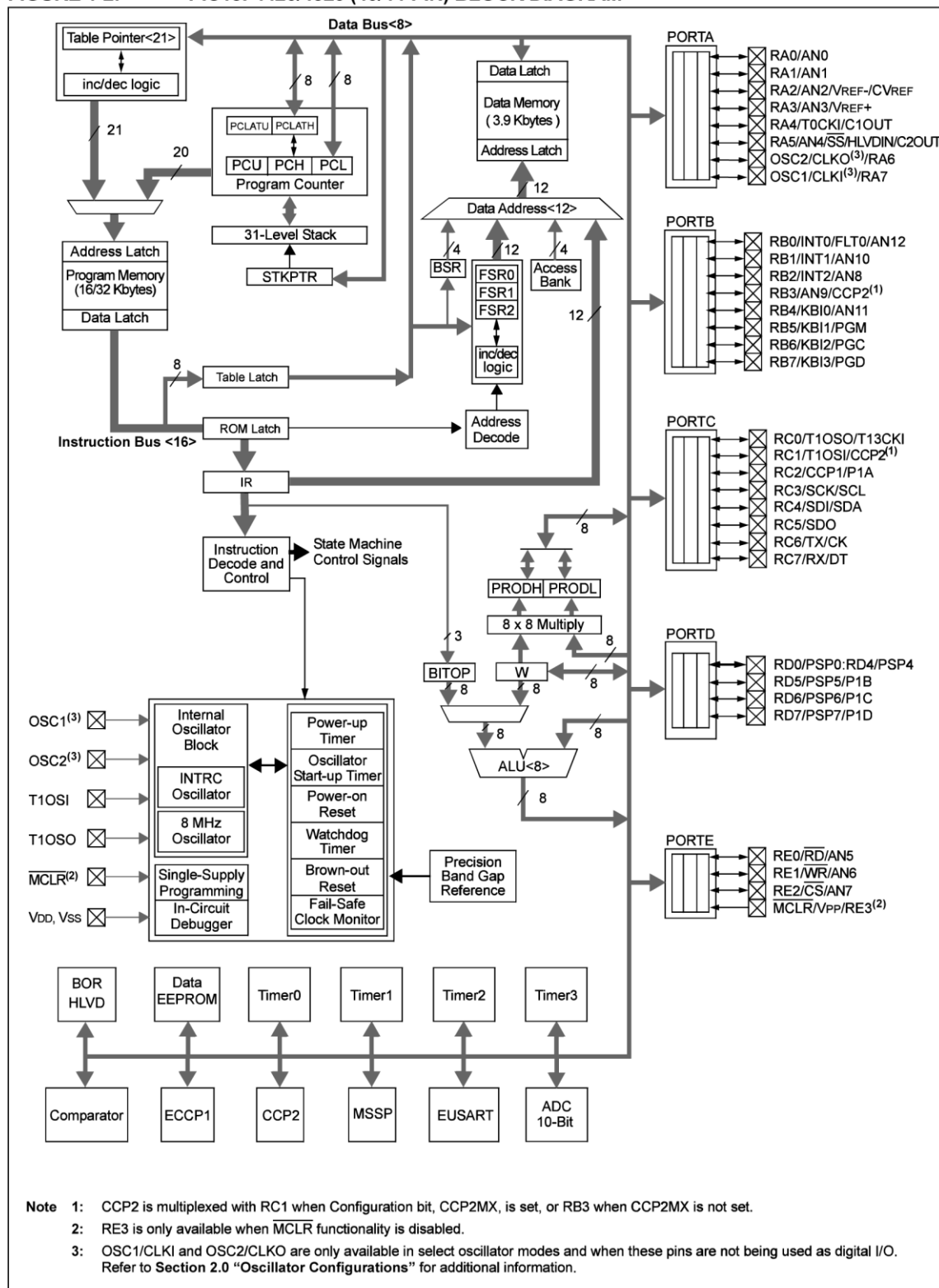
PIC18F2420/2520/4420/4520

FIGURE 1-1: PIC18F2420/2520 (28-PIN) BLOCK DIAGRAM



PIC18F2420/2520/4420/4520

FIGURE 1-2: PIC18F4420/4520 (40/44-PIN) BLOCK DIAGRAM

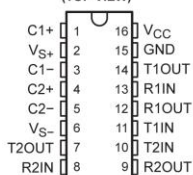


MAX232 ENTEGRESİ

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)

description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
-40°C to 85°C	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

MAX232, MAX232I

DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

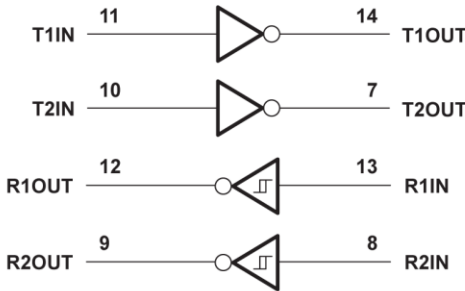
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



TEXAS
INSTRUMENTS

POST OFFICE BOX 655300 • DALLAS, TEXAS 75265

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)[†]

Input supply voltage range, V_{CC} (see Note 1)	–0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	–0.3 V to –15 V
Input voltage range, V_I : Driver	–0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	–0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Operating virtual junction temperature, T_J	150°C
Storage temperature range, T_{stg}	–65°C to 150°C

[†] Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.

2. Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\text{max}) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

3. The package thermal impedance is calculated in accordance with JEDEC 51-7.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN, T2IN)	2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			± 30	V
T_A	Operating free-air temperature	MAX232		0	70
		MAX232I		–40	85
					°C

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP [‡]	MAX	UNIT
I_{CC} Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10	mA

[‡] All typical values are at $V_{CC} = 5$ V and $T_A = 25^\circ\text{C}$.

NOTE 4: Test conditions are C1–C4 = 1 μF at $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$.



TEXAS
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

3

ULN2803 ENTEGRESİ

MOTOROLA

**Octal High Voltage,
High Current Darlington
Transistor Arrays**

The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open-collector outputs and free wheeling clamp diodes for transient suppression.

The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.

MAXIMUM RATINGS ($T_A = 25^{\circ}\text{C}$ and rating apply to any one device in the package, unless otherwise noted.)

Rating	Symbol	Value	Unit
Output Voltage	V_O	50	V
Input Voltage (Except ULN2801)	V_I	30	V
Collector Current – Continuous	I_C	500	mA
Base Current – Continuous	I_B	25	mA
Operating Ambient Temperature Range	T_A	0 to +70	$^{\circ}\text{C}$
Storage Temperature Range	T_{stg}	–55 to +150	$^{\circ}\text{C}$
Junction Temperature	T_J	125	$^{\circ}\text{C}$

$R_{\theta JA} = 55^{\circ}\text{C/W}$
Do not exceed maximum current limit per driver.

ORDERING INFORMATION

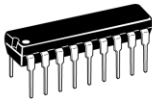
Device	Characteristics		
	Input Compatibility	$V_{CE}(\text{Max})/I_C(\text{Max})$	Operating Temperature Range
ULN2803A	TTL, 5.0 V CMOS	50 V/500 mA	$T_A = 0 \text{ to } +70^{\circ}\text{C}$
ULN2804A	6 to 15 V CMOS, PMOS		

Order this document by ULN2803/D

**ULN2803
ULN2804**

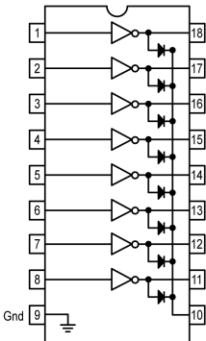
**OCTAL PERIPHERAL
DRIVER ARRAYS**

**SEMICONDUCTOR
TECHNICAL DATA**



A SUFFIX
PLASTIC PACKAGE
CASE 707

PIN CONNECTIONS



ULN2803 ULN2804

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$, unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Output Leakage Current (Figure 1) ($V_O = 50\text{ V}$, $T_A = +70^\circ\text{C}$) ($V_O = 50\text{ V}$, $T_A = +25^\circ\text{C}$) ($V_O = 50\text{ V}$, $T_A = +70^\circ\text{C}$, $V_I = 6.0\text{ V}$) ($V_O = 50\text{ V}$, $T_A = +70^\circ\text{C}$, $V_I = 1.0\text{ V}$)	All Types All Types ULN2802 ULN2804	I_{CEX}	— — — —	— — — —	100 50 500 500	μA
Collector–Emitter Saturation Voltage (Figure 2) ($I_C = 350\text{ mA}$, $I_B = 500\text{ }\mu\text{A}$) ($I_C = 200\text{ mA}$, $I_B = 350\text{ }\mu\text{A}$) ($I_C = 100\text{ mA}$, $I_B = 250\text{ }\mu\text{A}$)	All Types All Types All Types	$V_{CE(sat)}$	— — —	1.1 0.95 0.85	1.6 1.3 1.1	V
Input Current – On Condition (Figure 4) ($V_I = 17\text{ V}$) ($V_I = 3.85\text{ V}$) ($V_I = 5.0\text{ V}$) ($V_I = 12\text{ V}$)	ULN2802 ULN2803 ULN2804 ULN2804	$I_{I(on)}$	— — — —	0.82 0.93 0.35 1.0	1.25 1.35 0.5 1.45	mA
Input Voltage – On Condition (Figure 5) ($V_{CE} = 2.0\text{ V}$, $I_C = 300\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 200\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 250\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 300\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 125\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 200\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 275\text{ mA}$) ($V_{CE} = 2.0\text{ V}$, $I_C = 350\text{ mA}$)	ULN2802 ULN2803 ULN2803 ULN2803 ULN2804 ULN2804 ULN2804 ULN2804	$V_{I(on)}$	— — — — — — — —	— — — — — — — —	13 2.4 2.7 3.0 5.0 6.0 7.0 8.0	V
Input Current – Off Condition (Figure 3) ($I_C = 500\text{ }\mu\text{A}$, $T_A = +70^\circ\text{C}$)	All Types	$I_{I(off)}$	50	100	—	μA
DC Current Gain (Figure 2) ($V_{CE} = 2.0\text{ V}$, $I_C = 350\text{ mA}$)	ULN2801	h_{FE}	1000	—	—	—
Input Capacitance		C_I	—	15	25	pF
Turn–On Delay Time (50% E_I to 50% E_O)		t_{on}	—	0.25	1.0	μs
Turn–Off Delay Time (50% E_I to 50% E_O)		t_{off}	—	0.25	1.0	μs
Clamp Diode Leakage Current (Figure 6) ($V_R = 50\text{ V}$)	$T_A = +25^\circ\text{C}$ $T_A = +70^\circ\text{C}$	I_R	—	—	50 100	μA
Clamp Diode Forward Voltage (Figure 7) ($I_F = 350\text{ mA}$)		V_F	—	1.5	2.0	V

ULN2803 ULN2804

TEST FIGURES

(See Figure Numbers in Electrical Characteristics Table)

Figure 1.

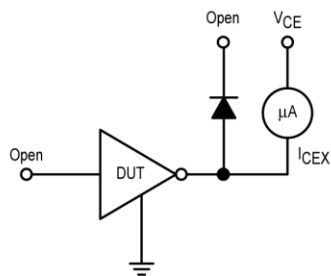


Figure 2.

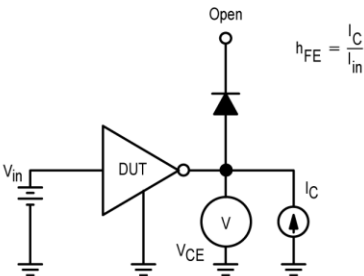


Figure 3.

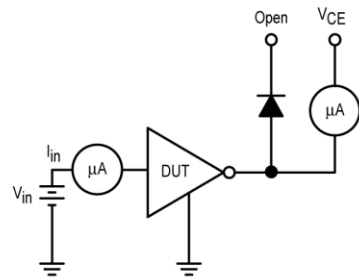


Figure 4.

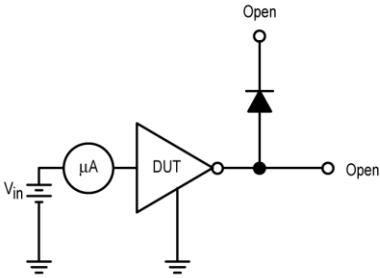


Figure 5.

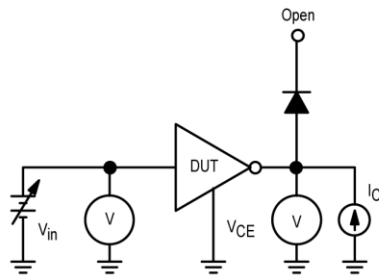


Figure 6.

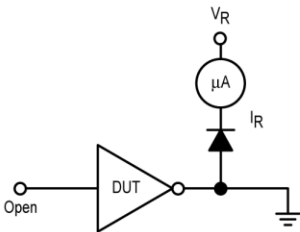
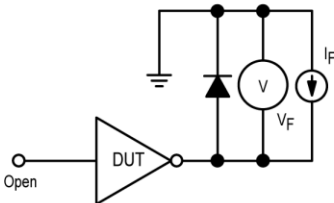


Figure 7.



ULN2803 ULN2804

TYPICAL CHARACTERISTIC CURVES – $T_A = 25^{\circ}\text{C}$, unless otherwise noted
Output Characteristics

Figure 8. Output Current versus Saturation Voltage

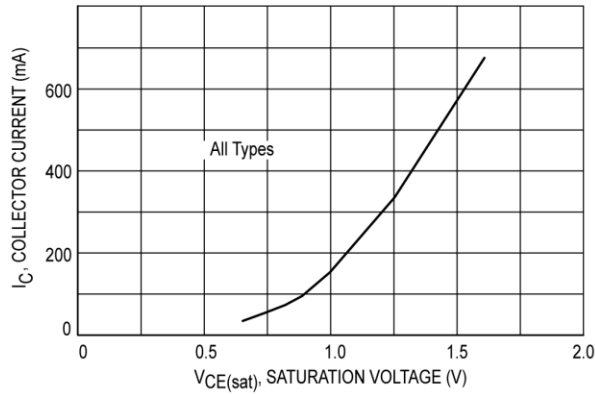
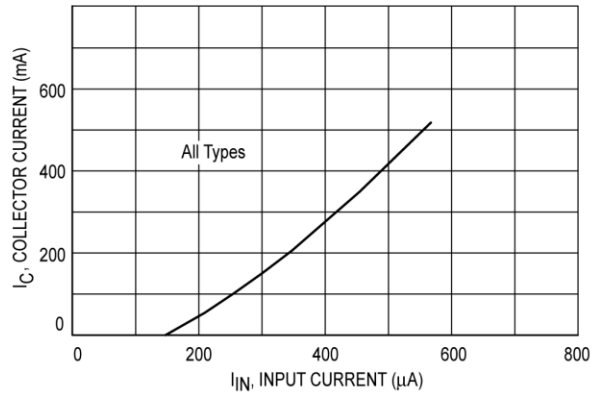


Figure 9. Output Current versus Input Current



Input Characteristics

Figure 10. ULN2803 Input Current versus Input Voltage

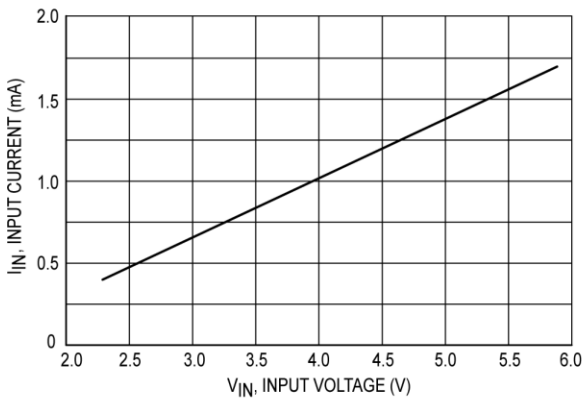


Figure 11. ULN2804 Input Current versus Input Voltage

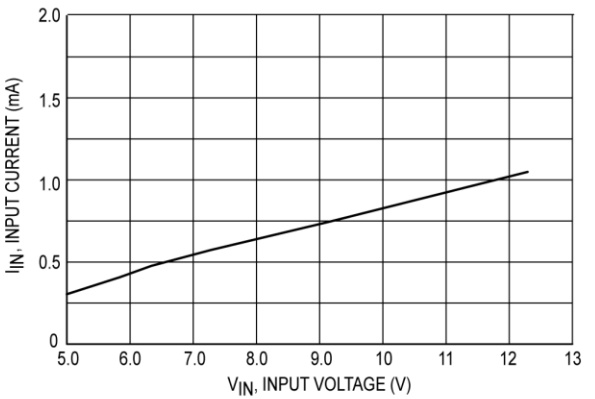
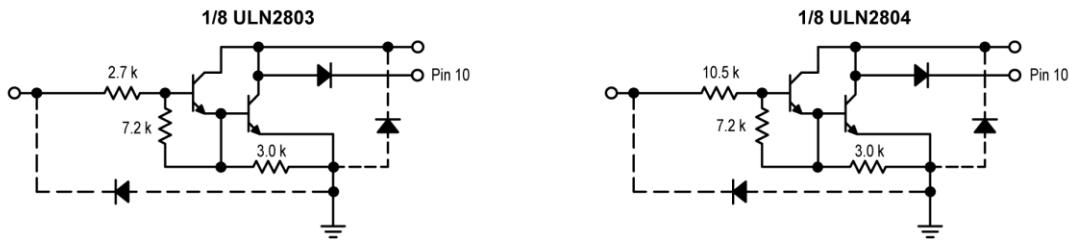
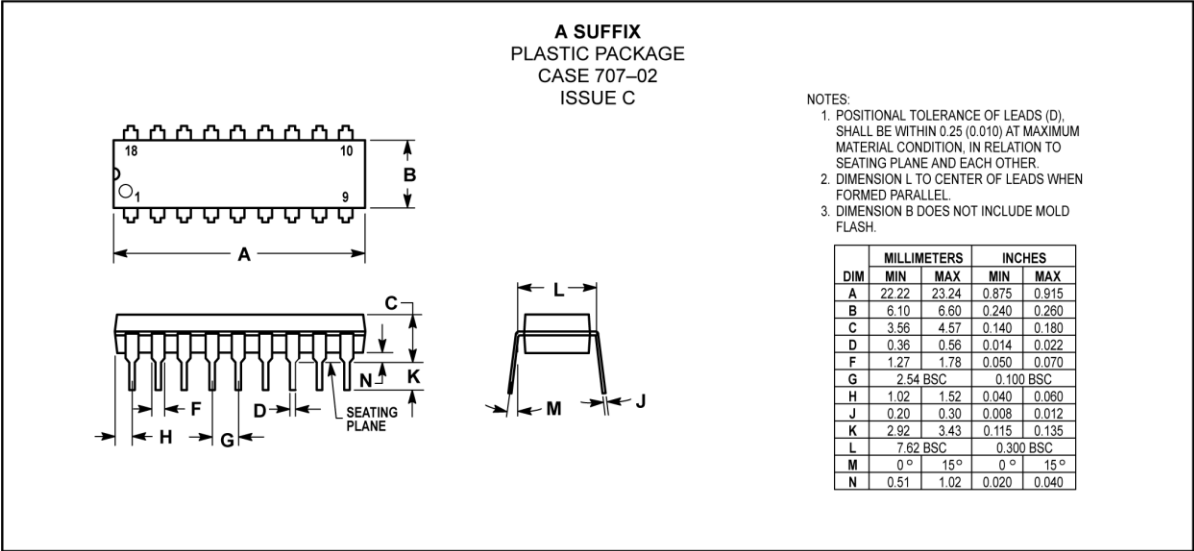


Figure 12. Representative Schematic Diagrams



ULN2803 ULN2804

OUTLINE DIMENSIONS



ESİSTEMDE KULLANILAN ADIM MOTORU

MITSUMI

Stepping Motors M42SP-7

Stepping Motors

OUTLINE

Standard type motor "M42SP-7" which belongs to the outer diameter 42mm model series. Despite 7.5° of step angle, and excitation method that are all in common with "M42SP-5", its body has been reduced to 15.7mm in thickness. This type is the most suited to any compact electronic apparatuses.



FEATURES

1. High output torque.
2. Superior running quietness and stability.
3. Step angle : 7.5°.
4. Excellent responsiveness acquired.

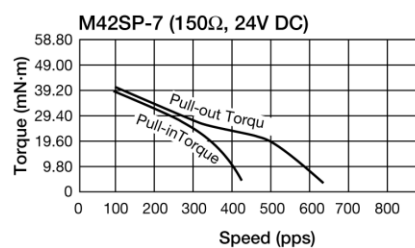
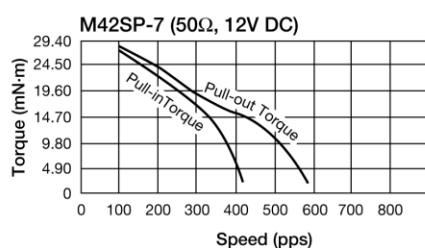
USES

Printers, typewriters, word processors, facsimiles, and such.

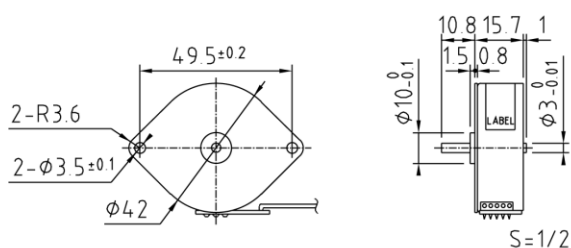
SPECIFICATIONS

Items	M42SP-7	
Rated Voltage	DC 12V	DC 24V
Working Voltage	DC 10.8~13.2V	DC 21.6~26.4V
Rated Current/Phase	259mA	173mA
No. of Phase	4 Phase	4 Phase
Coil DC Resistance	50Ω/phase±7%	150Ω/phase±7%
Step Angle	7.5°/step	7.5°/step
Excitation Method	2-2 Phase excitation (Unipolar driving)	
Insulation Class	Class E insulation	Class E insulation
Holding Torque	49.0mN·m	52.9mN·m
Pull-out Torque	23.5mN·m/200pps	33.3mN·m/200pps
Pull-in Torque	19.6mN·m/200pps	29.4mN·m/200pps
Max. Pull-out Pulse Rate	600pps	650pps
Max. Pull-in Pulse Rate	420pps	430pps

CHARACTERISTICS



DIMENSIONS



Unit : mm, General tolerance : ±0.5

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Morteza ABEDINI
Doğum Tarihi ve Yeri : 1979-Iran
Yabancı Dili : İngilizce
E-posta : mabedini@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektrik-Haberleşme	Azade Eslami	2002
Lise	Matematik	Taleghani	1997

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2012	Arup Mühendislik	Proje koordinatörü
2010	BGM Endüstriyel A.Ş.	Proje Mühendisi
2009	Net Mühendislik	QC/QA Inspector