

T.C.  
MİLLÎ EĞİTİM BAKANLIĞI



# MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN  
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON  
TEKNOLOJİLERİ

MİKRODENETLEYİCİ-3

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

# İÇİNDEKİLER

AÇIKLAMALAR .....	ii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1.....	3
1. RÖLE KONTROLÜ .....	3
1.1. Mekanik Röle.....	3
1.2. Işık Kontrollü Röle .....	4
1.3. Doğru Akım Motoru Kontrolü .....	5
1.3.1. DA Motorları Hakkında Genel Bilgi.....	5
UYGULAMA FAALİYETLERİ .....	10
ÖLÇME VE DEĞERLENDİRME.....	23
ÖĞRENME FAALİYETİ-2.....	24
2. ADIM Motoru kontrolü .....	24
2.1. Adım Motoru Hakkında Genel Bilgiler .....	24
2.1.1. Adım Motorlarının Temel Özellikleri .....	25
2.1.2. Adım Derecesi .....	25
2.1.3. Uçlarının Tespiti .....	25
2.1.4. Çalışma Prensibi .....	26
2.2. Adım Motoru Sürme Metotları.....	26
2.2.1. Adım Motorunun 1 Fazlı Sürülmesi .....	27
2.2.2. Adım Motorunun 2 Fazlı Sürülmesi .....	28
2.2.3. Adım Motorunun 1 ve 2 Fazlı Sürülmesi.....	28
2.2.4. Adım Motorunun Pozisyon Kontrolü .....	30
UYGULAMA FAALİYETLERİ .....	31
ÖLÇME VE DEĞERLENDİRME.....	36
MODÜL DEĞERLENDİRME.....	37
CEVAP ANAHTARLARI .....	38
KAYNAKÇA.....	62

# AÇIKLAMALAR

<b>KOD</b>	<b>523EO0367</b>
<b>ALAN</b>	<b>Endüstriyel Otomasyon Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Ortak alan</b>
<b>MODÜLÜN ADI</b>	<b>Mikrodenetleyici-3</b>
<b>MODÜLÜN TANIMI</b>	Mikrodenetleyici ile sürücü elemanlarını kontrol etmek için programlama becerisinin kazandırıldığı bir öğrenme materyalidir.
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	Mikrodenetleyici -2 modülünü almış olmak.
<b>YETERLİK</b>	Mikrodenetleyici ile sürücü elemanların kontrol işlemlerini yapmak.
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Mikrodenetleyici ile sürücü elemanlarının kontrol işlemlerini hatasız olarak yapabileceksiniz. <b>Amaçlar</b> <b>1.</b> Mikrodenetleyici ile DA motorunun ON-OFF kontrolü hatasız olarak yapabileceksiniz. <b>2.</b> Mikrodenetleyici ile adım motoru kontrolünü hatasız olarak yapabileceksiniz.
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Mikrobilgisayar Laboratuvarı <b>Donanım:</b> Bilgisayar, mikrodenetleyici programlama kartı, çeşitli doğru akım motorları, doğru akım motoru sürücü devresi, çeşitli adım motorları, adım motoru sürücü devresi
<b>ÖLÇME VE DEĞERLENDİRME</b>	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

# GİRİŞ

## **Sevgili Öğrenci,**

Mikrodenetleyici 3 modülü ile endüstriyel otomasyon teknolojileri alanında en çok kullanılan mikrodenetleyici programlamaya yönelik bilgi ve teknolojiye ait temel yeterlilikleri kazanacaksınız.

Bu modülü başarılı bir şekilde tamamladığınızda mekanik rölenin, ışık kontrollü rölenin, doğru akım ve adım motorlarının çalışma prensiplerini kavrayarak detaylı bilgiye sahip olacaksınız. Ayrıca bu röle ve motorları mikrodenetleyici ile kontrol etmek için gerekli programları yazarak kendinizi geliştireceksiniz.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

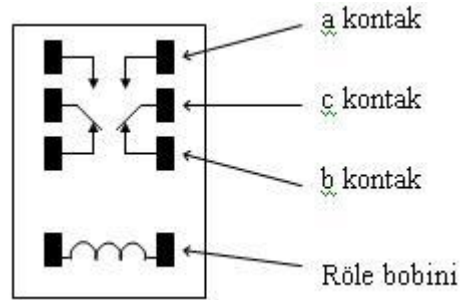
Mikrodenetleyici ile DA motorunun ON-OFF kontrolü hatasız olarak yapabileceksiniz.

## ARAŞTIRMA

- Ø Mekanik röle, ışık kontrollü röle ve doğru akım motorları hakkında bir araştırma yaparak rapor hâline getiriniz.

## 1. RÖLE KONTROLÜ

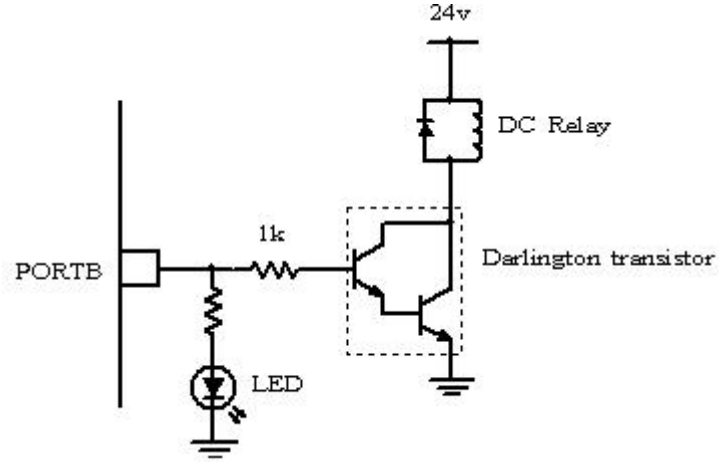
Bu bölümde PIC ile röle kontrolünün nasıl yapıldığını öğreneceğiz. Elektronik devrelerde küçük rölelerin çok çeşitli tipleri kullanılmaktadır. Büyük elektrik güçleri olan devrelerde ise güç röleleri kullanılmaktadır. Ayrıca bobin ile kontak arasında hiç bir bağlantı olmayan tamamen ışık ile kontrol eden röleler de vardır. Bunlara SSR ( Solid State Relays ) adı verilir. Normal rölelerde a, b ve c harfleri kontak noktalarıdır ve çok amaçlı kullanılmaktadırlar. Kontrol ünitelerinde ihtiyaç sayısına göre kontak bulunduran röleler tavsiye edilir. Rôlenin iç şekli ve kontak noktalarını gösteren şemalar Şekil 1.1’de gösterilmiştir.



Şekil 1.1: Rôlenin iç şekli ve kontak noktalarının şeması

### 1.1. Mekanik Röle

Rôlenin çalışması için bir LED’den daha fazla bir akım gerekmektedir. Bu nedenle Mikrodenetleyicinin uçlarına doğrudan röle bağlayamayız. PIC ile bir röle kontrolü yapmak istediğimizde Şekil 1.2’de görüldüğü gibi transistör kullanmalıyız.



Şekil 1.2: Rölenin kumandası

Röle elektromekanik bir anahtardır. Küçük bir plastik kutu içerisinde yerleştirilmiş olan bir **bobin** ve bobinin çekim kuvveti ile hareket eden bir anahtardan ibarettir. Bobin uçlarına +12 V gerilim uygulayınca bobin tepesinde bulunan L şeklindeki metal parçayı kendine doğru çeker. Bir eksen üzerinde hareket eden bu metal parçanın diğer kısmı ise kendisine bağlı olan plastik malzemeyi iter. Plastik malzemenin üzerinde metal bir kontak mevcuttur. Bu kontak, gerilim uygulanmadan önce temas ettiği kontakta ayrılır, diğer kontakta temas eder. Bizler, gerilimin bir ucunu plastik üstündeki kontakta, diğerini de diğer iki kontakta birine bağlayarak, anahtarlama olayını gerçekleştireceğiz.



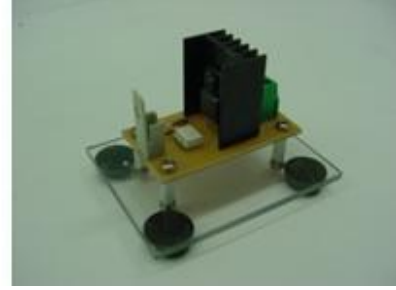
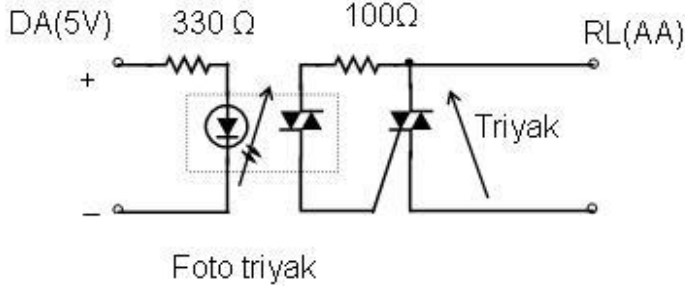
Şekil 1.3: Mekanik röle ve bacak numaraları

## 1.2. Işık Kontrollü Röle

SSR (Solid State Relays-Işık kontrollü röle) bobin ile kontak arasında hiç bir bağlantı olmayan tamamen ışık ile kontrol eden rölelerdir. AA güç kontrolü için oldukça kullanışlıdır. Giriş ve çıkış birimleri tamamen birbirinden ayırılır ve sadece foto ışık ile birbirlerine etki



ederler. Sadece foto triyakin LED'ini ON off yapmak suretiyle AA gücünü kolaylıkla kontrol edebiliriz. Bu nedenle PIC portları ile bu LED'i kontrol etmek suretiyle AA gücünü de kontrol etmiş oluruz. Görüldüğü gibi bu röle tamamen elektronik malzemelerden oluşmaktadır. Mekanik bir kontak olmadığı için oksitlenme söz konusu değildir ve kısa süreli ON-OFF uygulamalarında rahatlıkla kullanılır.



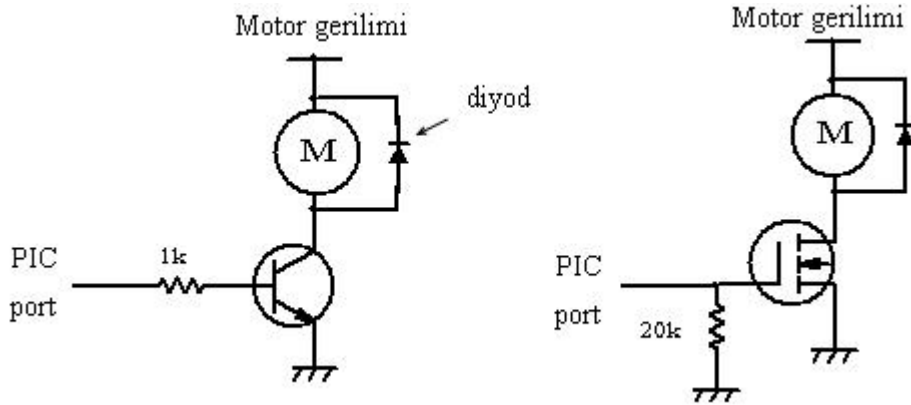
Şekil 1.4: SSR ( Solid State Relay )

### 1.3. Doğru Akım Motoru Kontrolü

#### 1.3.1. DA Motorları Hakkında Genel Bilgi

##### 1.3.1.1. Transistör ve FET ile DA Motor Kontrolü

Transistör ve FET ile DA motor kontrolü yapabiliriz. Aşağıda bu kontrollerin nasıl yapılabileceğini gösteren şekiller görülmektedir.

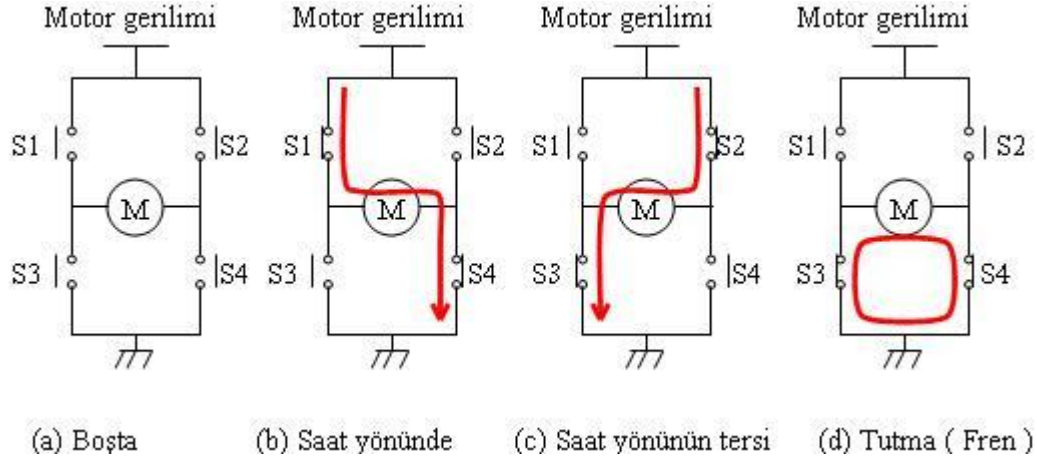


Şekil 1.5: DA Motor Kontrol Metodları

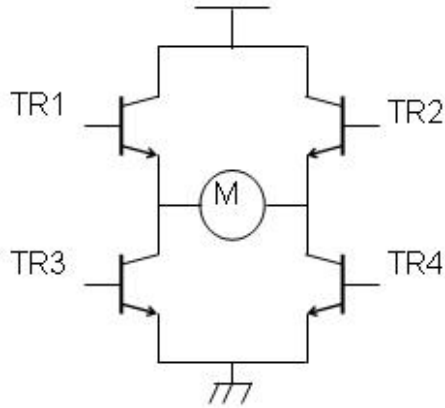
Şekil. 1.5'de DA motorlarının kontrol metodları görülmektedir. Eğer PIC 'in portundan gelen sinyal 1 (5V) ise motor dönmeye başlar. Eğer PIC' in portundan gelen sinyal 0 (0V) ise motor durur. Bu methodla motorda sadece bir yönde dönme sağlanır.

### 1.3.1.2. H Köprü Devresi

Motorun her iki yönde de dönmesini istediğimiz zaman Şekil 1.6'daki devreyi kullanmalıyız.



Şekil 1.6: H köprü devresi



SW1	SW2	SW3	SW4	Açıklama
TR1	TR2	TR3	TR4	
OFF	OFF	OFF	OFF	Boşta
ON	OFF	OFF	ON	Saat yönünde
OFF	ON	ON	OFF	Saat yönü tersi
OFF	OFF	ON	ON	Tutma ( Fren )

Şekil 1.7: Transistörle yapılmış H köprü devresi

### 1.3.1.3. Motor sürücü

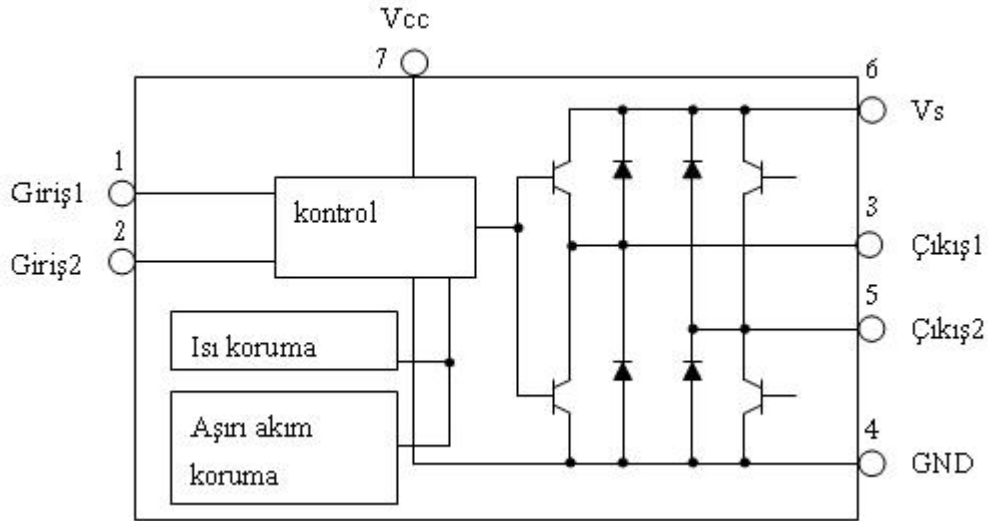
Motor sürücüsü DA motorlarda devir yönü değiştirmek için tam köprü sürücüye sahiptir. Bu devre elemanı ile DA motoru saat yönünde, saat yönünün tersi yönde, boşta ve fren durumunda kolaylıkla kontrol edebiliriz.

### Özelliđi (Motor sürücü TA7257P)

- (a) Çıkış akımı ortalama 1.5A, maksimum 4.5A'dır.
- (b) 4 fonksiyon moda sahiptir. (İleri, Geri, Boşta, Fren) Bütün bunları 2 ucunu besleyerek 2 lojik sinyalle gerçekleştirebiliriz
- (c) Aşırı akım, kısa devre ve ısıya karşı koruma vardır.
- (d) İşletim gerilimi :  $V_{cc} = 6 - 18V$ ,  $V_s = 0 - 18V$



Şekil 1.8: TA7257P Entegresi



Şekil 1.9: Blok diyagramı (TA7257P)

PIN No.	Sembol	Fonksiyon Açıklaması
1	IN1	Giriş ucu
2	IN2	Giriş ucu
3	OUT1	Çıkış ucu
4	GND	Toprak ucu
5	OUT2	Çıkış ucu
6	$V_s$	Motor sürücüsü için gerilim ucu
7	$V_{cc}$	Lojik devrenin çalışması için gerilim ucu

Şekil 1.10: TA7257P'nin pin uçları

Giriş 1	Giriş 2	Çıkış 1	Çıkış 2	Çalışma şekli
1	1	H	H	Fren
0	1	L	H	İleri
1	0	H	L	Geri
0	0	L	L	Boşta

Şekil 1.11: TA7257P'nin fonksiyonları

### 1.3.1.4. DA Motorlarında Hız Kontrolü

DA motorlarında hızı kontrol etmek için iki yöntem vardır. Bunlardan birisi gerilim kontrolü diğeri ise darbe genişlik kontrolüdür.



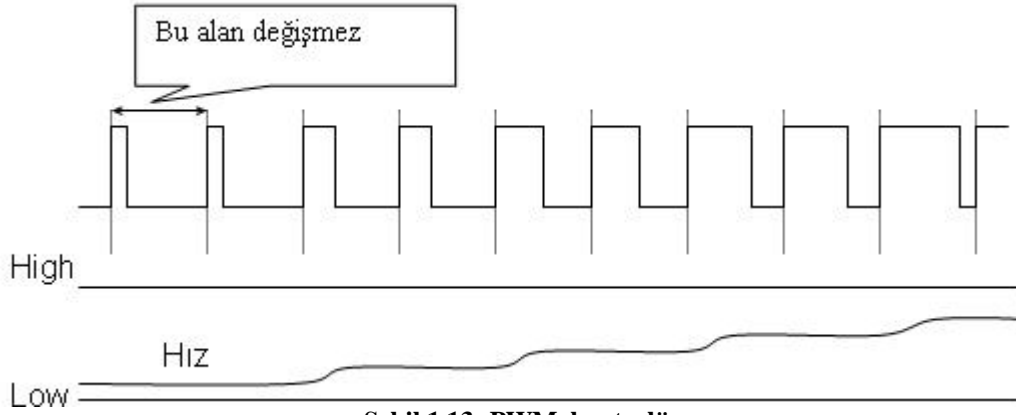
Şekil 1.12: Gerilim kontrolü

#### (a) Gerilim kontrolü

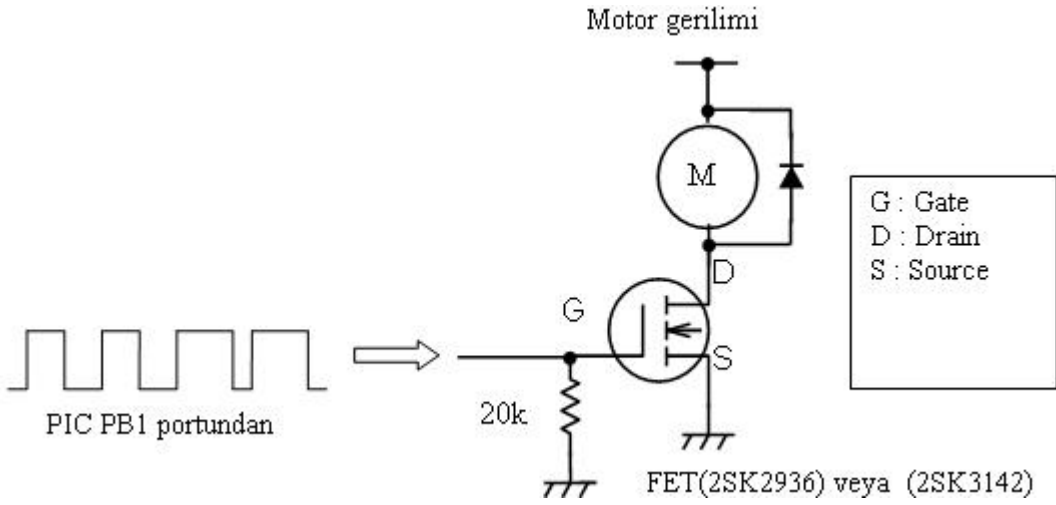
Gerilim kontrolü analog bir kontroldür (Şekil 1.12). Bu yöntem genel olarak kullanılan bir metottur. Bu metodu kullanmak için elektrik amplifikatörüne ihtiyaç vardır, ancak verimli ve kullanışlı bir metot değildir. Bu metodu mikrodenetleyiciler ile kullanamayız.

#### (b) PWM kontrol (Darbe genişlik kontrolü )

PWM metodu darbe genişliğini kontrol ederek motorun hızını kontrol etmektedir. PWM yöntemi tamamen dijital bir kontroldür. PWM ile kare dalga sinyalin 1 veya 0 olma durumlarını kontrol ederiz. Sadece motor hızını değil lambanın yanma parlaklığını da kontrol edebiliriz (Şekil 1.14).



Şekil 1.13: PWM kontrolü



Şekil 1.14: PIC ile PWM kontrolü

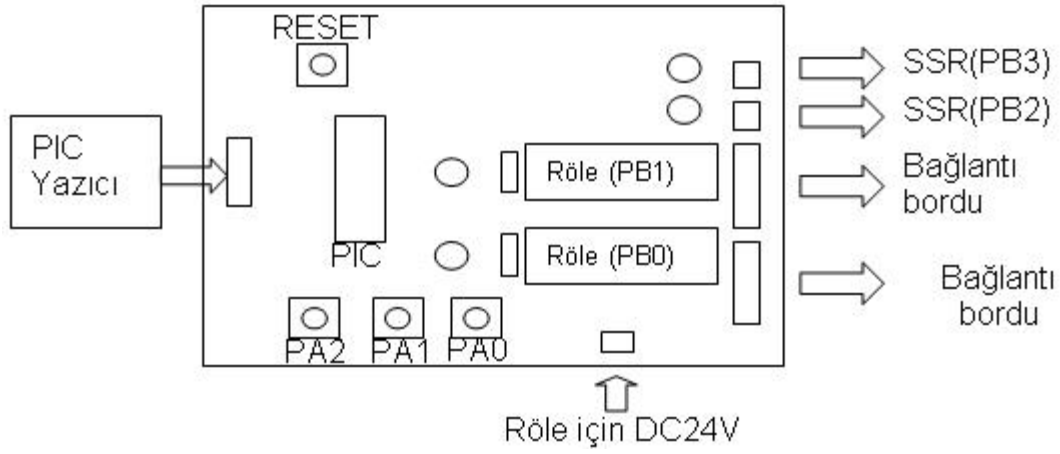
Şekil 1.14’de görüldüğü gibi bu şekilde DA motorunu kontrol etmek için FET’e ihtiyaç vardır. Çünkü FET’in anahtarlama özelliği iyidir ve PWM için en uygun özelliğe sahiptir.

## UYGULAMA FAALİYETİ-1

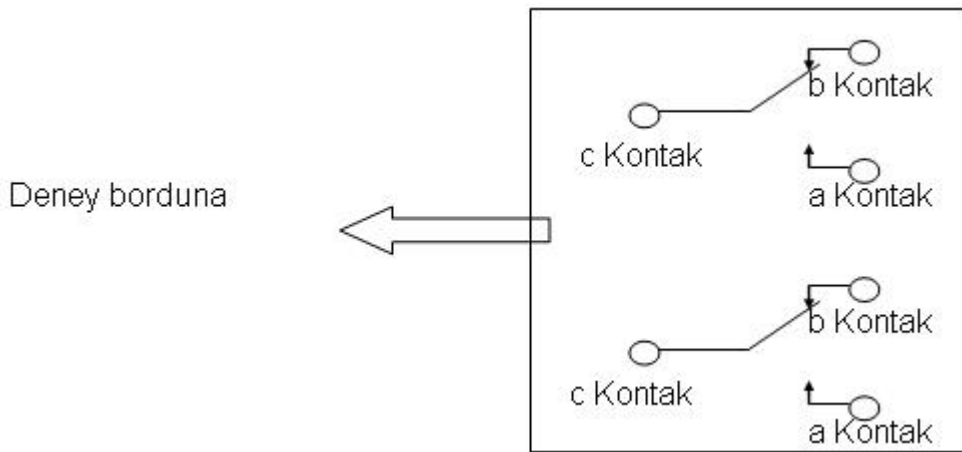
### 1. Kullanılan Araç ve Gereçler

a) PIC Programlayıcı Devresi	1
b) Uygulama bordsu	1
c) Bağlantı bordsu	1
d) Güç kaynağı	1
e) AC kablo	1
f) Soketli lamba duyu	1

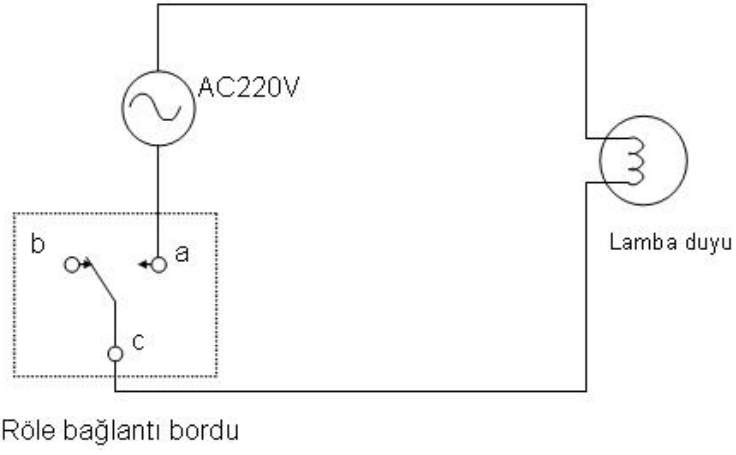
### 2. Röle Uygulama Devreleri



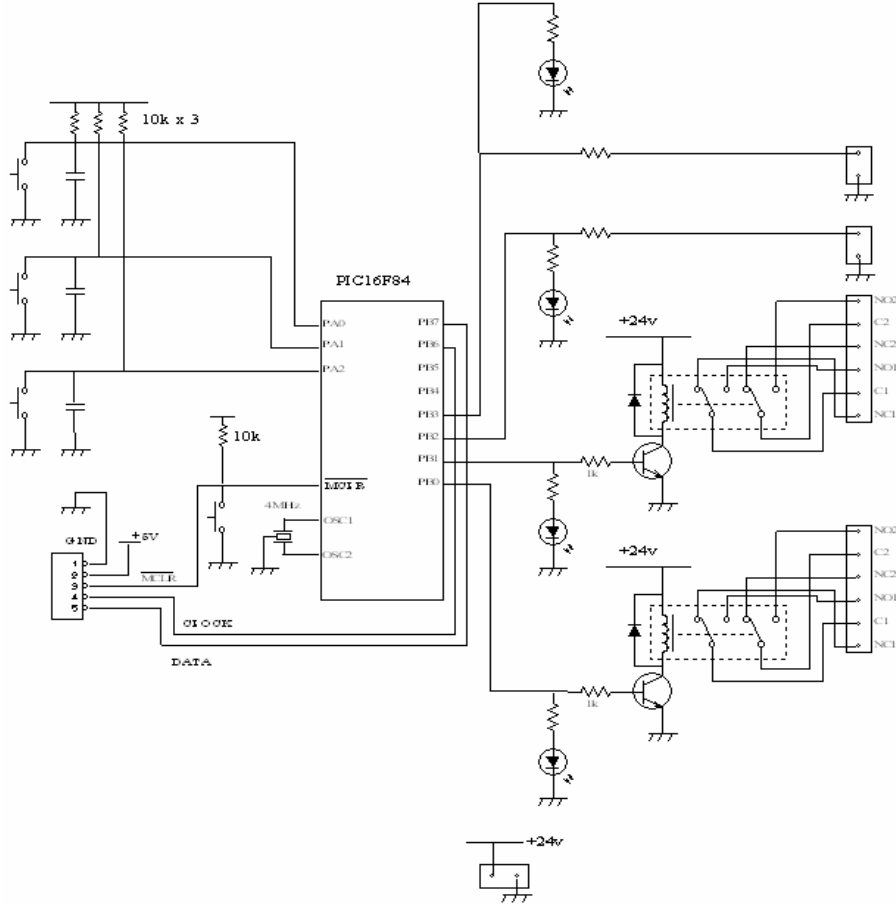
Şekil 1.15: Röle deney bordsunun görünüşü



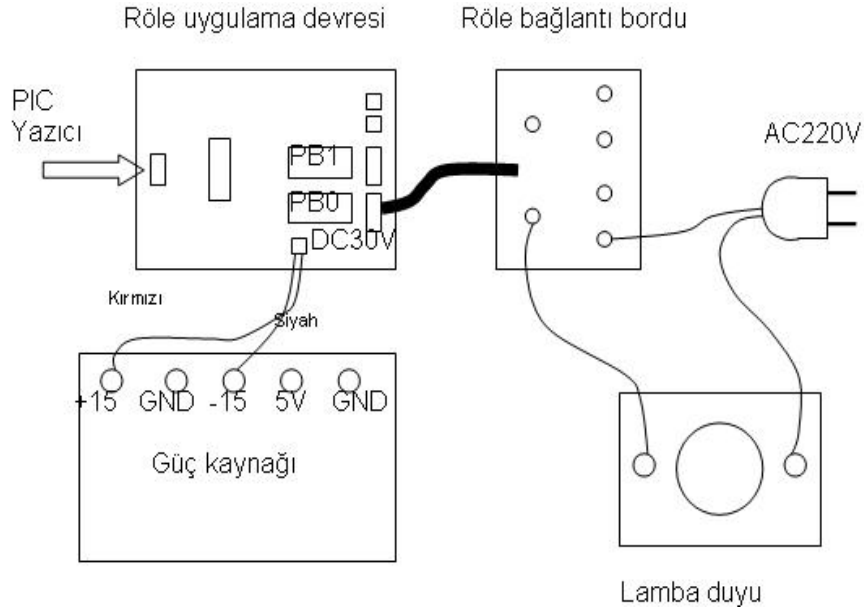
Şekil 1.16: Bağlantı bordsu görünüşü



Şekil 1.17: Uygulama devre şeması



Şekil 1.18: Röle deney bordsu devre şeması



Şekil 1.19: Uygulamanın iletken bağlantı şeması

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> <li>• Şekil 1.19'daki bağlantıyı kurunuz.</li> <li>• Devreyi Öğretmeninize kontrol ettiriniz.</li> </ul> <p><b>Program Uygulaması-1</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığı sürece lambanın yanmasını sağlayan, elimizi çektiğimiz an ise lambanın sönmesini sağlayan programı yazınız.</li> <li>• Programı mikrodenetleyiciye yükleyiniz.</li> <li>• Devreye enerji verip devre çalışmasını gerçekleştiriniz</li> </ul> <p><b>Program Uygulaması-2</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığında lambanın yanmasını sağlayan PA1 butonuna basıldığında ise lambanın sönmesini sağlayan programı yazınız.</li> <li>• Programı mikrodenetleyiciye yükleyiniz.</li> <li>• Devreye enerji verip devre çalışmasını gerçekleştiriniz.</li> </ul>	<ul style="list-style-type: none"> <li>• Bağlantı kablolarına 220V gelmediğinden emin olunuz.</li> <li>• İletkenleri kısa devre etmemeye özen gösteriniz.</li> <li>• 220V kablolarının bağlanacağı klemenslerin ve duyun vidalarını iyice sıkınız.</li> </ul>

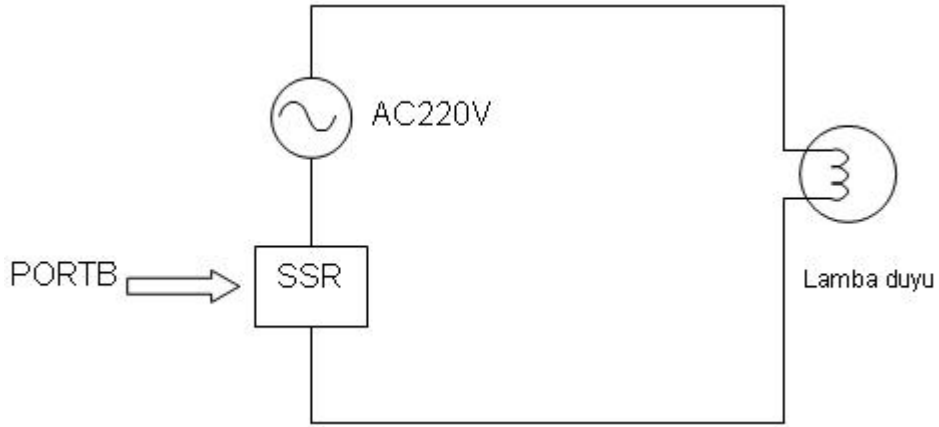


## UYGULAMA FAALİYETİ-2

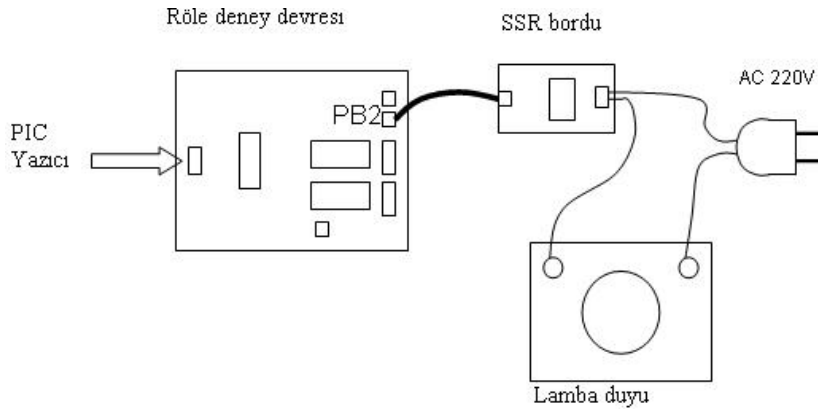
### 1. Kullanılan Araç ve Gereçler

- a) PIC Programlayıcı Devresi 1
- b) Uygulama bordu 1
- c) SSR board 1
- d) AA için kablo 1
- e) Soketli lamba duyu 1

### 2. Röle Uygulama Devreleri



Şekil 1.20: Uygulama devre şeması



Şekil 1.21: Deneyin iletken bağlantı şeması

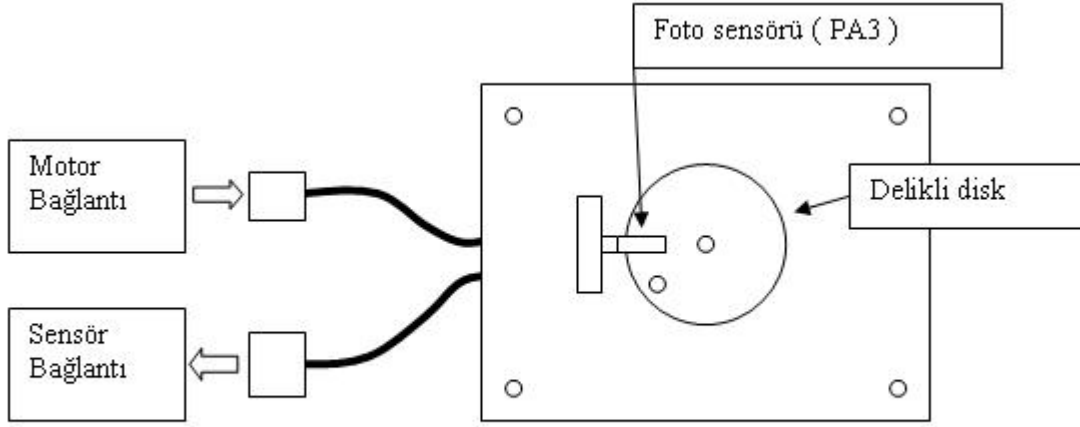
İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> <li>• Şekil 1.21'deki bağlantıyı kurunuz.</li> <li>• Devreyi Öğretmeninize kontrol ettiriniz.</li> </ul> <p><b>Program Uygulaması-3</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığı sürece lambanın yanmasını sağlayan elimizi çektiğimiz an ise lambanın sönmesini sağlayan programı yazınız.</li> <li>• Programı mikrodenetleyiciye yükleyiniz.</li> <li>• Devreye enerji verip devre çalışmasını gerçekleştiriniz</li> </ul> <p><b>Program Uygulaması-4</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığında lambanın yanmasını sağlayan PA1 butonuna basıldığında ise lambanın sönmesini sağlayan programı yazınız.</li> <li>• Programı mikrodenetleyiciye yükleyiniz.</li> <li>• Devreye enerji verip devre çalışmasını gerçekleştiriniz.</li> </ul> <p><b>Program Uygulaması-5</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığında lambanın 0.5 saniye aralıklarla yanıp sönmesini, PA1 butonuna basıldığında da lambanın sürekli sönmesini sağlayan programı yazınız.</li> <li>• Programı mikrodenetleyiciye yükleyiniz.</li> <li>• Devreye enerji verip devre çalışmasını gerçekleştiriniz.</li> </ul>	<ul style="list-style-type: none"> <li>• Bağlantı kablolarına 220V gelmediğinden emin olunuz.</li> <li>• İletkenleri kısa devre etmemeye özen gösteriniz.</li> <li>• 220V kablolarının bağlanacağı klemenslerin ve duyun vidalarını iyice sıkınız.</li> </ul>

## UYGULAMA FAALİYETİ-3

### 1. Kullanılan Araç ve Gereçler

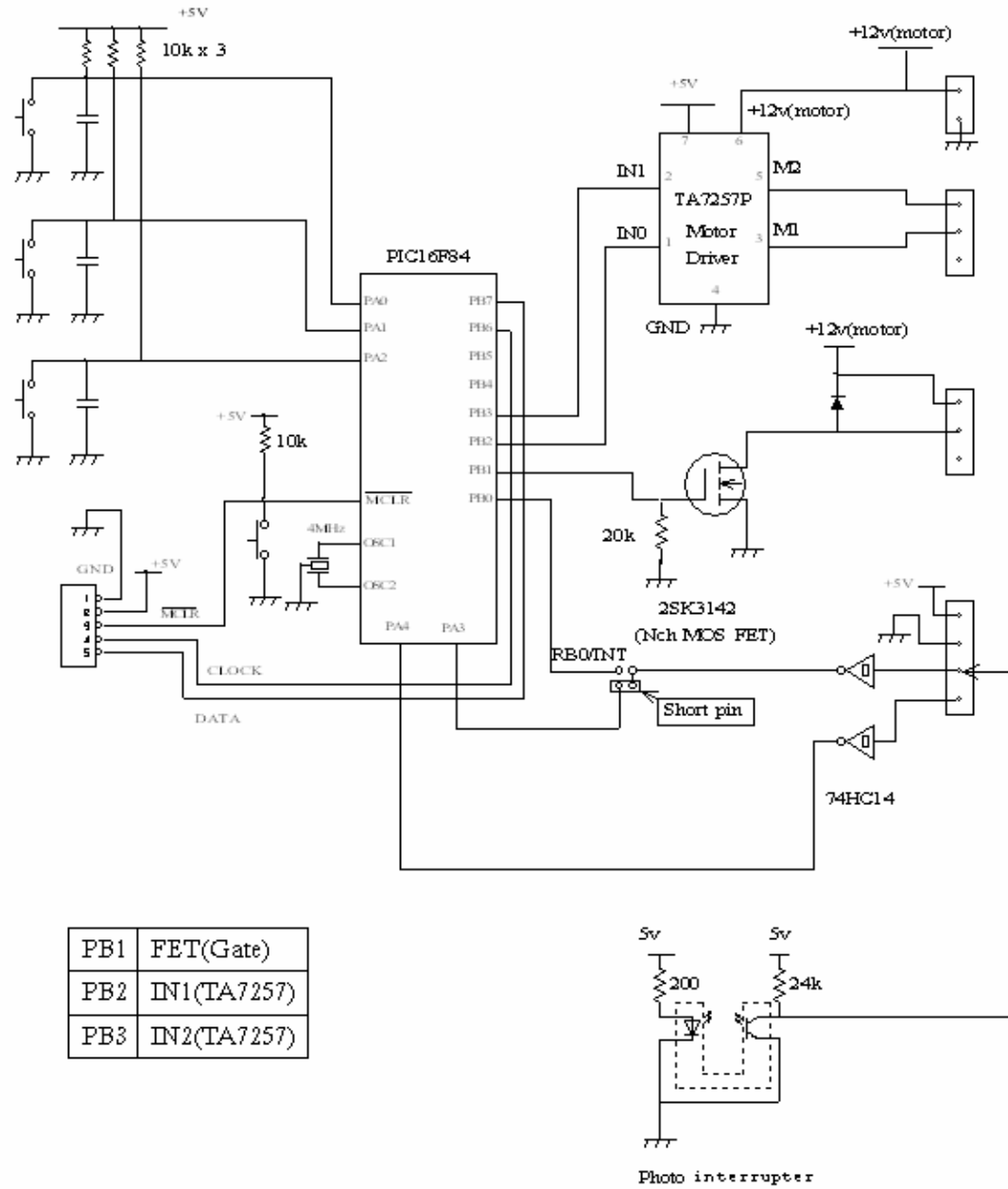
- a) PIC Programlayıcıyı Devresi 1
- b) DA Motor Eğitim Seti 1
- c) DA Motor 1
- d) Foto sensör 1
- e) Bağlantı Kabloları 2

### 2. Uygulama Devreleri

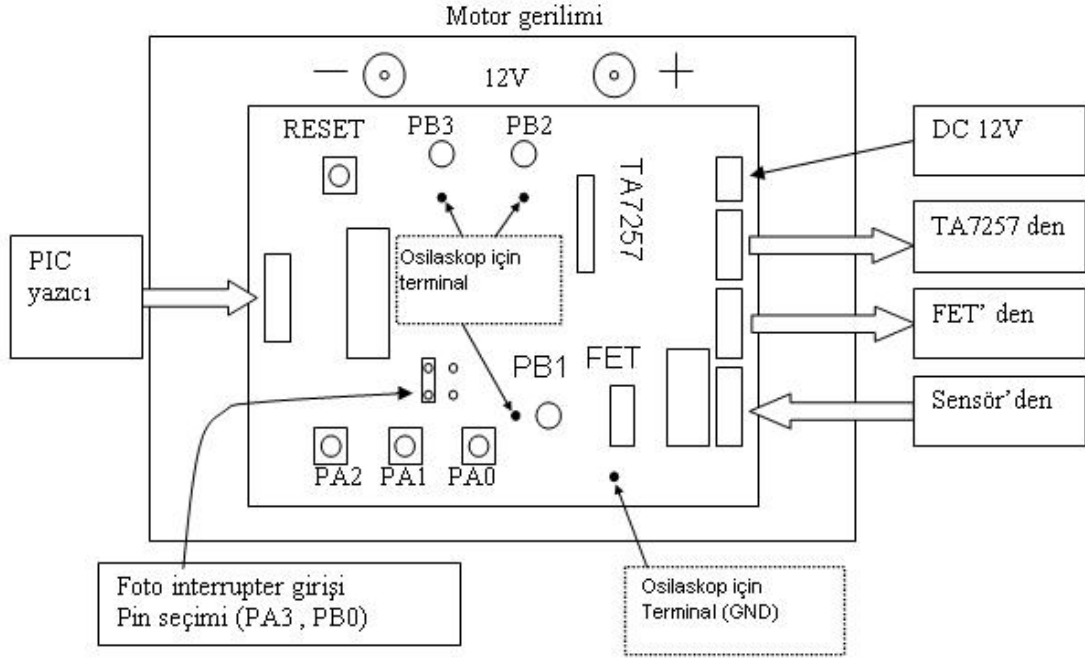


Redüktörlü DA motor ( TSUKASA TG23A-SG 12V )  
Foto sensörü (T oshiba TP507A )

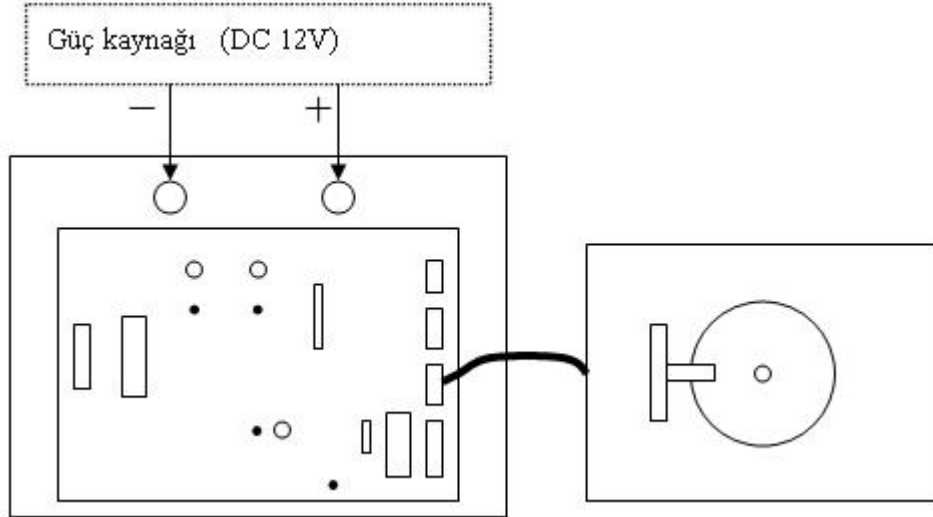
Şekil 1.22: Sensörlü DA motor



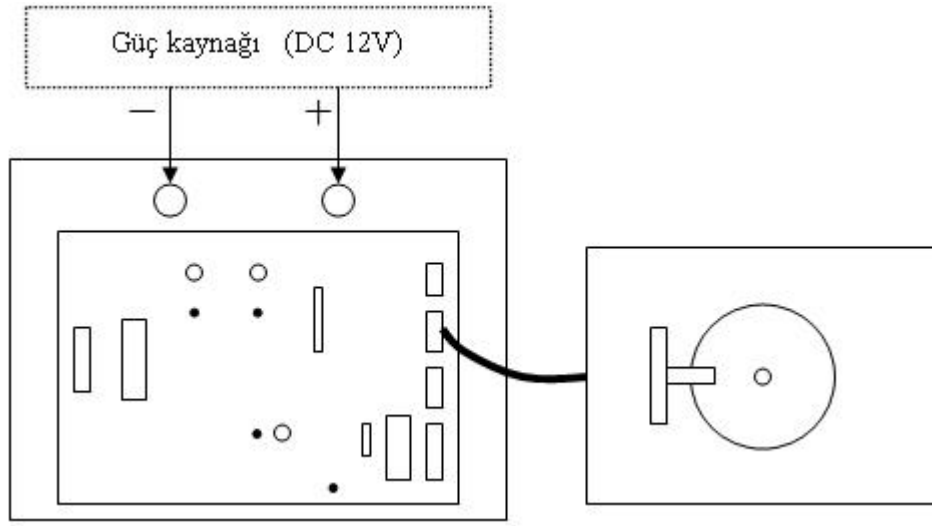
Şekil 1.23: DA motor eğitim setinin devre şeması



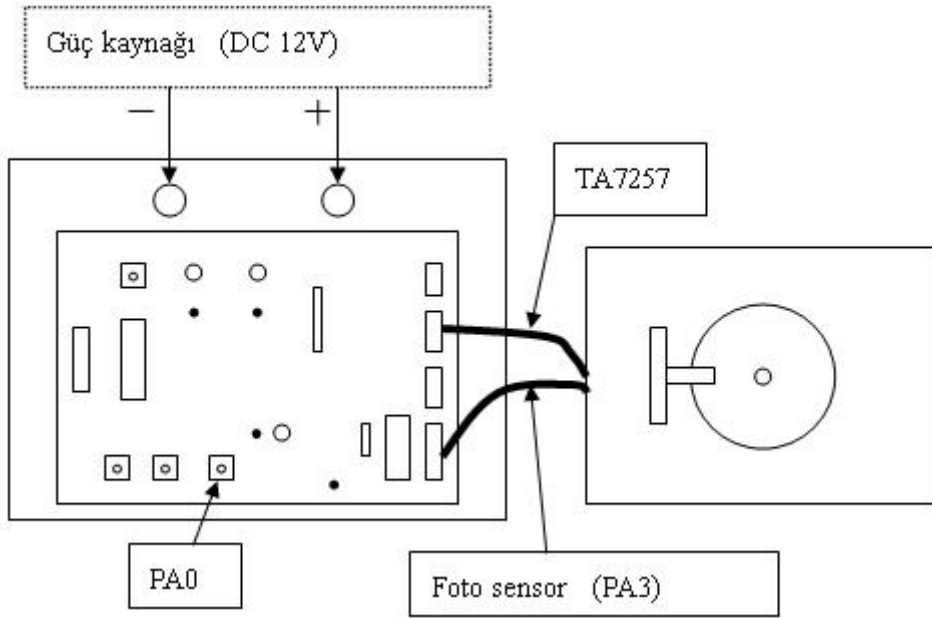
Şekil 1.24: DA motor eğitim seti üst görünüş



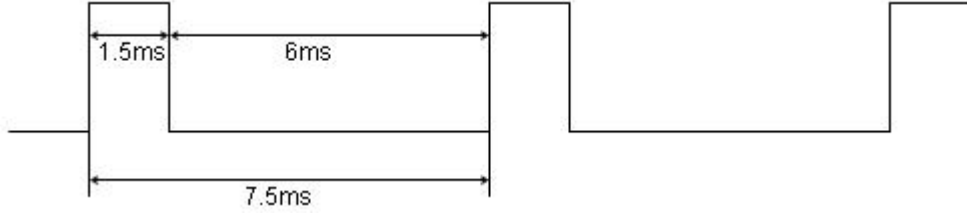
Şekil 1.25: Doğru akım motorunun FET ile bağlantısı



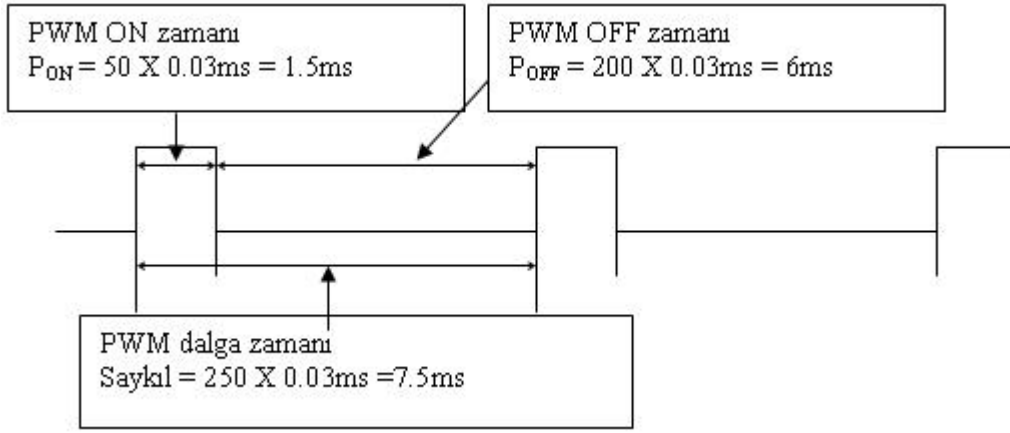
Şekil 1.26: Doğru akım motorunun TA7257 ile bağlantısı



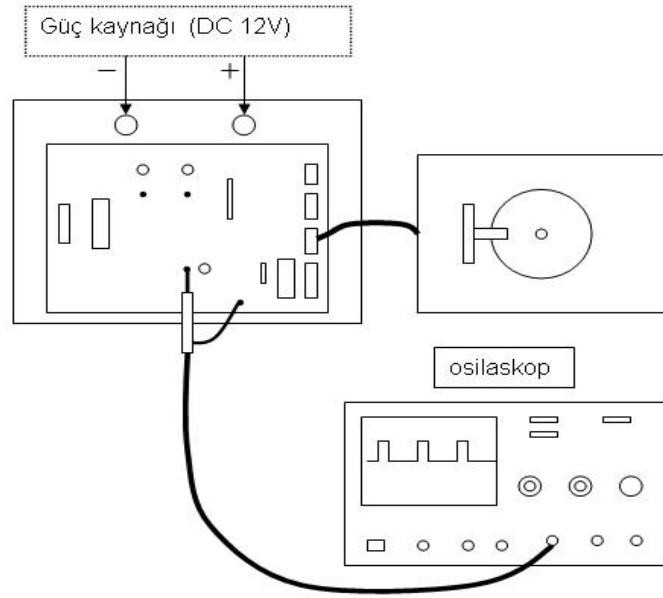
Şekil 1.27: Uygulama 8'in iletken bağlantısı



Şekil 1.28: DA motorun Pwm dalgası



Şekil 1.29: Programda PWM değeri



Şekil 1.30: Uygulama -9'un osiloskop ile bağlantısı

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> <li>• Doğru Akım motorunun Şekil 1-25'deki gibi FET ile bağlantısını sağlayınız.</li> <li>• Devreyi Öğretmeninize kontrol ettiriniz.</li> </ul> <p style="text-align: center;"><b>Program Uygulaması-6</b></p> <ul style="list-style-type: none"> <li>• PA0 butonuna basıldığında motoru döndüren, PA1 butonuna basıldığında ise motoru durduran programı yazınız.</li> <li>• Programı programlayıcı ile yükleyiniz. Çalışmasını gerçekleştiriniz.</li> </ul> <p style="text-align: center;"><b>Program Uygulaması-7</b></p> <ul style="list-style-type: none"> <li>• Doğru Akım motorunun Şekil 1.26'daki gibi TA7257 ile bağlantısını sağlayınız.</li> <li>• Açıklama-1'e uygun olarak programı yazıp PIC'e yükleyiniz.</li> <li>• Çalışmasını gerçekleştiriniz.</li> </ul> <p style="text-align: center;"><b>Program Uygulaması-8</b></p> <ul style="list-style-type: none"> <li>• Açıklama-2'ye uygun olarak Şekil 1.27 için gerekli programı yazınız.</li> <li>• Programı PIC'e yükleyip devrede deneyiniz.</li> </ul> <p style="text-align: center;"><b>Program Uygulaması-9</b></p> <ul style="list-style-type: none"> <li>• Doğru Akım motorunun Şekil 1-25'deki gibi FET ile bağlantısını sağlayınız.</li> <li>• PIC'in PB1 çıkışından Şekil 1-28'deki gibi bir dalga ile motoru çalıştıran bir program yazınız.</li> <li>• Hesaplamalar için açıklama-3'ü okuyunuz.</li> <li>• Programı PIC'e yükleyip sonucu osilaskop ile izleyiniz.</li> <li>• Osilaskop görüntüsünü Şekil 1.31'e çiziniz.</li> </ul> <p style="text-align: center;"><b>Program Uygulaması-10</b></p> <ul style="list-style-type: none"> <li>• Şekil 1-27'deki devre bağlantısını kurunuz ve açıklama-4'de istenilenlere göre bir program yazınız.</li> <li>• Programı PIC'e yükleyip çalışmasını deneyiniz.</li> </ul>	<ul style="list-style-type: none"> <li>• İletkenleri kısa devre etmemeye özen gösteriniz.</li> <li>• Açıklamalara uygun olarak programı yazıp yine açıklamalarda yazılan çalışmayı sırasıyla gerçekleştiriniz.</li> </ul>



### **Açıklama-1:**

- (a) Butonlara basılı değilse>> Motor durur(fren modu) ve giriş butonunu beklemededir.
- (b) PA0 butonuna basılırsa>> Motor saat yönünde döner CW(clockwise).
- (c) PA1 butonuna basılırsa>> Motor saat yönünün tersi yönde döner CCW(counter clockwise)
- (d) PA2 butonuna basılırsa>>Motor durur(fren modu) ve giriş butonunu beklemededir.

Not : Doğru akım motorlarında eğer motorun dönüş yönünü değiştirmek istiyorsak önce PA2 (stop) butonuna basılmalı daha sonra diğer butonları kullanmalıyız.

### **Açıklama-2**

- (a) PA0 butonuna basılırsa DA motor 10 tur saat yönünde döner. (Motorun dönüş sayısını PA3 ucuna bağlı bulunan foto sensör ile tesbit etmekteyiz.)
- (b) 10 tur sonunda DA motor 1 saniye durur ve bu defa da DA motor saat yönünün tersi yönünde 10 tur döner ve yine 1 saniye durur.
- (c) Bu işlem sürekli tekrarlanır.

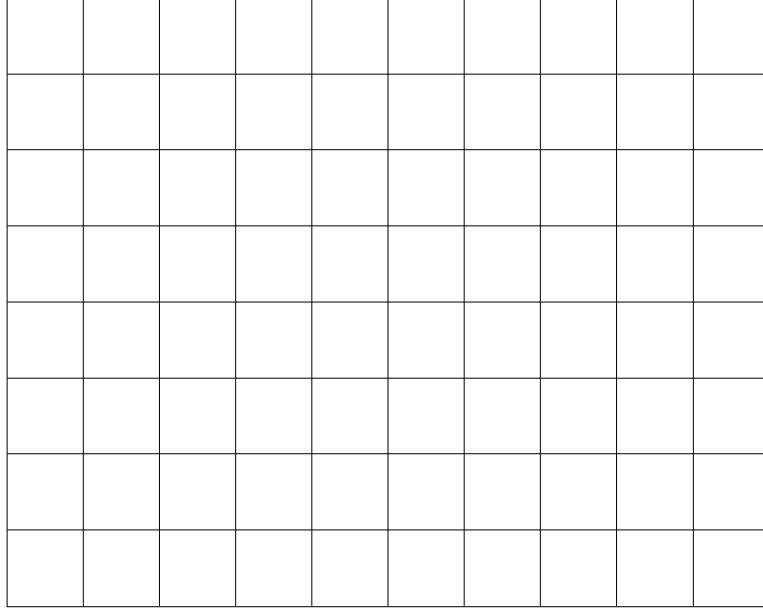
### **Açıklama-3**

(a) DA motor Şekil 1-28'deki dalga konumuna göre dönmektedir. Bir dalga boyu 7.5 ms. dir. ON zamanı 1.5ms ve OFF zamanı 6 ms.'dir. Bu data bilgileri PB1 ucuna gönderilerek motorun dönmesi sağlanır.

(b) PWM saykılının nümerik toplam değeri (250) dir. Bunun ON zamanı için (50) nümerik değer ayırırsak OFF zamanı içinde  $(250) - (50) = (200)$  nümerik değer buluruz.

### **Açıklama-4**

- (1) DA motor dururken PA0 butonu başlatma butonudur.  
(Dönüş hızını kendiniz belirleyebilirsiniz.)
- (2) PA1 butonu dönüş hızının artması için butondur.
- (3) PA2 butonu dönüş hızının azalması için butondur.
- (4) DA motoru dönerken, PA0 butonu durdurma butonudur.
- (5) PWM programının çıktısını osilaskop ile doğrulayınız.



**Şekil 1.31: Program Uygulaması-9 için Osiloskop Çizimi**

## ÖLÇME VE DEĞERLENDİRME

### A. OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Elektrik enerjisinin gelmesi ile kontakları konum değiştiren eleman aşağıdakilerden hangisidir?  
A) Transistör      B) SSR      C) Mekanik Röle      D) FET
2. Bir rölede hareketli kontak aşağıdakilerden hangisi ile isimlendirilir?  
A) b      B) d      C) c      D) a
3. PIC portu ile mekanik röleyi kontrol etmek için aşağıdaki elektronik elemanlardan hangisi röle sürücü olarak kullanılamaz?  
A) Transistör      B) Led      C) FET      D) MOSFET
4. Mekanik gürültü sorunu olmayan yarı iletken röle aşağıdakilerden hangisidir?  
A) Transistör      B) SSR      C) SRS      D) FET
5. DA motorun dönüş yönünü ayarlamak için kullanılan devre aşağıdakilerden hangisidir?  
A) Weston köprüsü      B) Yükseltici devresi  
C) H köprüsü      D) Empedans Uygunlaştırıcı
6. DA motorun hız kontrolünü gerçekleştirmek için kullanılan yöntem aşağıdakilerden hangisidir?  
A) FM      B) AM      C) SSM      D) PWM

### DEĞERLENDİRME

Eğer soruları doğru olarak yanıtlatabiliyorsanız bir sonraki öğrenme faaliyetine geçebilirsiniz. Yanlış cevaplarınız varsa ilgili bilgileri tekrar ediniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Mikrodenetleyici ile adım motoru kontrolünü hatasız olarak yapabileceksiniz.

## ARAŞTIRMA

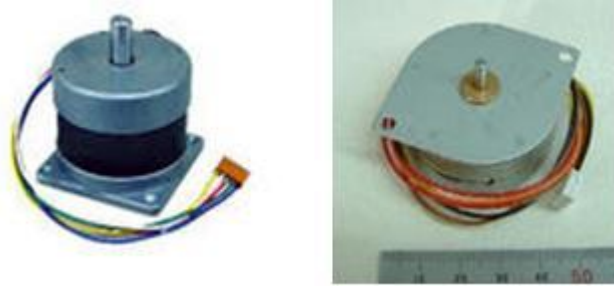
Ø Adım motorları hakkında bir araştırma yaparak rapor hâline getiriniz.

## 2. ADIM MOTORU KONTROLÜ

### 2.1. Adım Motoru Hakkında Genel Bilgiler

Adım motorları fırçasız DA motorları sınıfına girmektedir ve yazıcılarda, küçük ölçekli CNC tezgâhlarında, disket sürücülerde, diğer ufak çaplı uygulamalarda ve az da olsa hız kontrolleri gereken yerlerde kullanılırlar. DA motorlarında en fazla arıza gösteren yerler fırça kısımlarıdır. Bu nedenle fırçasız motorlar fırçalı motorlara göre daha uzun ömürlüdürler. Bu yüzden endüstriyel kontrol sistemleri gibi yerlerde arıza fazla olması istenmediğinden dolayı fırçalı motorlar tercih edilmezler.

Normal DA motorları ile adım motorlarını karşılaştırdığımızda adım motorlarının donanımları daha karmaşık olduğu görülür. Uygulamada hız ve adım kontrolü istenmiyorsa DA fırçalı motorlar daha iyi sonuç verirler. Ancak hız kontrolü veya pozisyon kontrolünün gerekli olduğu uygulamalarda adım motorlarını kullanmak gerekir. Bu doğrultuda adım motorları geri besleme olmadan hız kontrolünde veya pozisyon kontrolünde asla kayma ve şaşma göstermezler. Eğer geri besleme gerektiren bir motor kullanmış olsaydık bu defa da geri besleme ünitesi motorun maliyetinden daha fazla yük getirebilirdi.



Şekil 2.1: Adım motoru

### 2.1.1. Adım Motorlarının Temel Özellikleri

- (a) Dönüş açısı giriş faz sayısı ile orantılıdır.
- (b) Dönüş hızı giriş faz oranı ile orantılıdır. ( Faz frekansıyla )
- (c) Bazı torklar kendi kendine de oluşabilmektedir, çünkü iç yapısında kalıcı bir mıknatıs kullanılmaktadır.
- (d) Yüksek tork değerlerine sahiptirler ve hafiftirler.
- (e) Küçük açılı, yüksek verimli ve ucuzdurlar.
- (f) Bakıma ihtiyaçları yoktur, çünkü fırçasız bir DA motordur.

### 2.1.2. Adım Derecesi

Yapacağımız uygulamaya göre seçeceğimiz adımlı motor özellikleri çok önemlidir. Motorun her tam adımda döneceği derece miktarı kesin olarak bilinmelidir. Motorun yarım adım işletiminde adım veya dönme miktarı iki katı olacaktır. Ve her adım derecesi yarıya inecektir. Üzerinde adımın kaç derece olduğu yazılmayan motorlar için bu adım sayısı dikkatli bir şekilde elle döndürülerek sayılmalıdır. Bir tam dönüşteki adım sayısını tesbit ettikten sonra dairenin açısı olan 360'a bölersek her bir adımın kaç derece olduğunu bulmuş oluruz. Genel standart olarak: 0.72, 1.8, 3.6, 7.5, 15, ve hatta 90 derece bulabiliriz. Her adımın derecesi motorun kararlılığı ile ilgilidir. Eğer motorun üzerinde sadece adım sayısı veya tur sayısı yazıyorsa bu adım sayısını veya tur sayısını 360'a bölmeliyiz ki bir adımın kaç dereceye tekabül ettiğini bulabilelim.

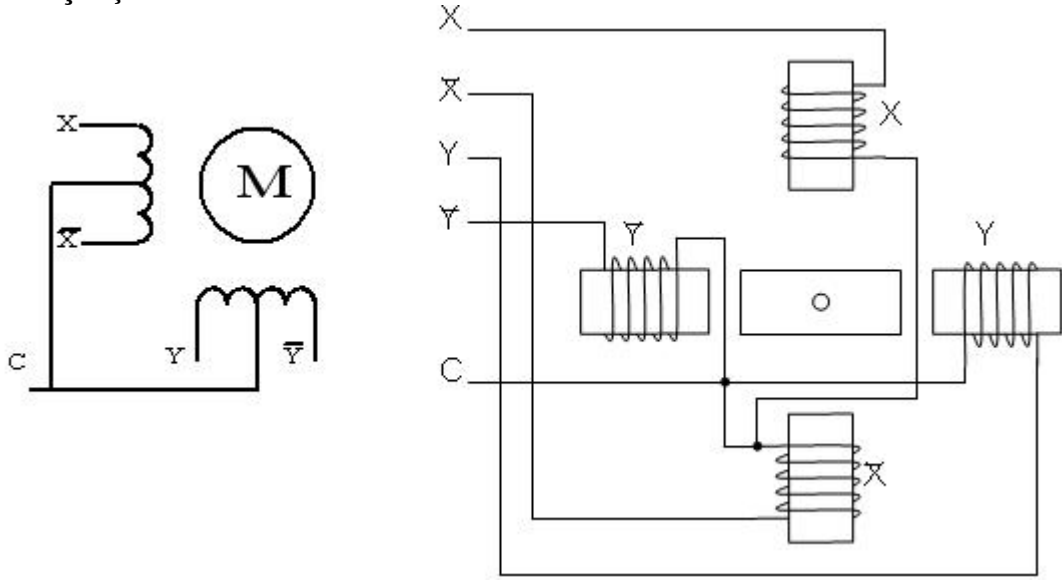
### 2.1.3. Uçlarının Tespiti

Adım motorlarının uç bağlantıları ile ilgili elimizde bir katalog veya doküman yok ise uçlarını ohmmetre ile bir kaç ölçme ve bir kaç deneme ile (en fazla 16 deneme) bulabiliriz.

Adım motorlarının 4 adet bobini vardır. Her bobinden birer uç çıkartılmıştır. Bir adet de ortak uç olmak üzere 5 uç mevcuttur. Ohm metre ile ölçümümüzde ortak uç ile diğer 4 bobin ucu arasında sabit bir direnç değeri vardır. Böylece ortak uç bulunabilir. Bu uca motorun çalışma gerilimi uygulanır. Kalan 4 ucu ise transistör veya FET gibi anahtarlama uçlarına bağlarız. Bağlantımız doğru ise motor dönmeye başlar, yanlış ise motor dönmez ve titreme yapar. Bu durumda 4 uçtan bazılarının yerini değiştirmemiz gerekir. Doğru uçları bulup motor dönene kadar bu işleme devam edilir ki 4 ucun 16 adet kombinasyonu vardır ve en fazla 16 deneme sonucunda gerçek uçları bulmuş oluruz.

Eğer adım motorumuz 6 uçlu ise bunun anlamı mevcut olan 4 bobin ikişerli gruplara ayrılarak birer ortak uç çıkartılmış demektir. 2 bobin ucu 1 ortak uç olmak üzere 3 uçlu iki grup vardır ve bunlar arasında elektriksel bir bağ bulunmamaktadır. Avometre ile 3'lü grupları birbirinden ayırmak oldukça kolaydır. Ayrılan bu gruplarda sabit direnç gösteren uç ortak uçtur. Sonuç olarak bulunan iki ortak uç birleştirilmek suretiyle 5 uçlu adımlı motor elde edilmiş olur. Zaten 5 uçlu adımlı motorlarda bu ortak uçlar içeride yapılmıştır. Bundan sonra yapılacak işler 5 uçlu ile aynıdır.

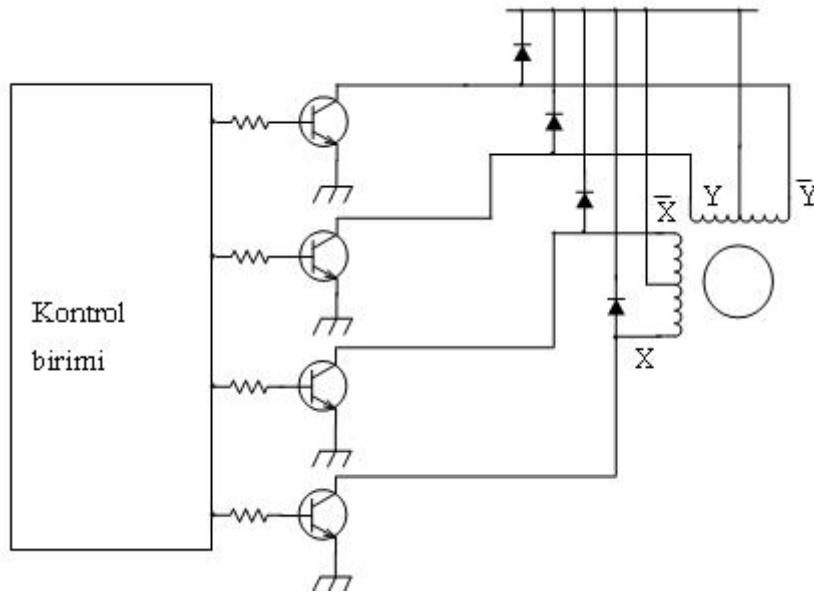
### 2.1.4. Çalışma Prensibi



Şekil 2.2: Adım motorlarının iç yapısı

Çok kutuplu (Unipolar) adım motorları 4 fazlı fırçasız motorlardır ve tipik olarak 5 veya 6 uçlu olarak yapılırlar. Eğer adım motoru 5 uçlu ise ortadaki uçlar motorun içerisinde bağlanmışlar ve ortak uç olarak dışarıya çıkarılmışlardır.

### 2.2. Adım Motoru Sürme Metotları



Şekil 2.3: Adım motoru sürücü devresi

Şekil 2-3’de görüldüğü gibi anahtarlama transistör kullanılmış, ancak kontrol ünitesi olarak transistörlerin açılıp kapanmasını sağlayan kısım gösterilmemiştir. Anahtarların kontrolü için ihtiyaç olan çıkış sinyallerini üreten bu birim genel olarak bilgisayar programları ile hazırlanıp bir ara birimle sürülmektedir.

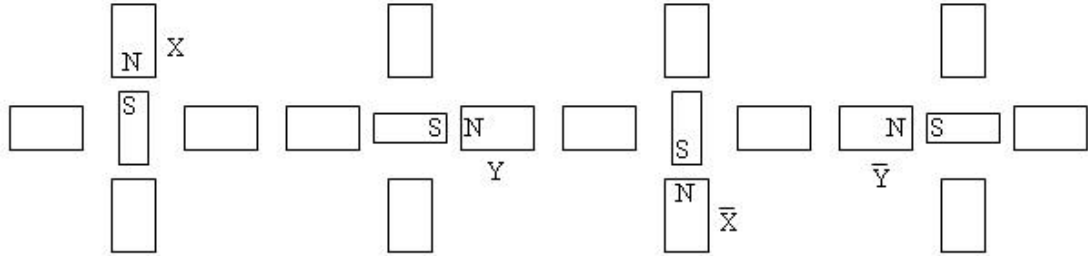
Şekil 2-4’de görüldüğü gibi adım motorunun çalışması için bobinleri kullanarak manyetik alanlar oluşturulmalıdır.

### 2.2.1. Adım Motorunun 1 Fazlı Sürülmesi

Bu metotta sürekli olarak bir fazlı düzenli bir gerilim vermeliyiz. Her bobin sıra ile enerjilendiğinde mıknatıslanarak zıt kutuplu rotoru kendine doğru çeker ve 1 adım döner.

Örneğin 1 adımı 1.8 derece olan adım motor 1 adımda 1.8 derece döner. Eğer Şekil 2.4’deki tabloda görülen 4 değer de sırayla verilirse tablo sonunda adım motor 4 adım atmış olur. Motorun tam bir tur dönmesi için (yani 360 dereceyi tamamlaması için)  $360/1.8= 200$  adım dönmesi ve bu tablonun  $200/4=50$  kez çağrılması gerekir.

Bu metotta tek bobin enerjilendiği için motor milini döndüren kuvvet 2 fazlı sürülen motorun torkundan daha zayıftır.



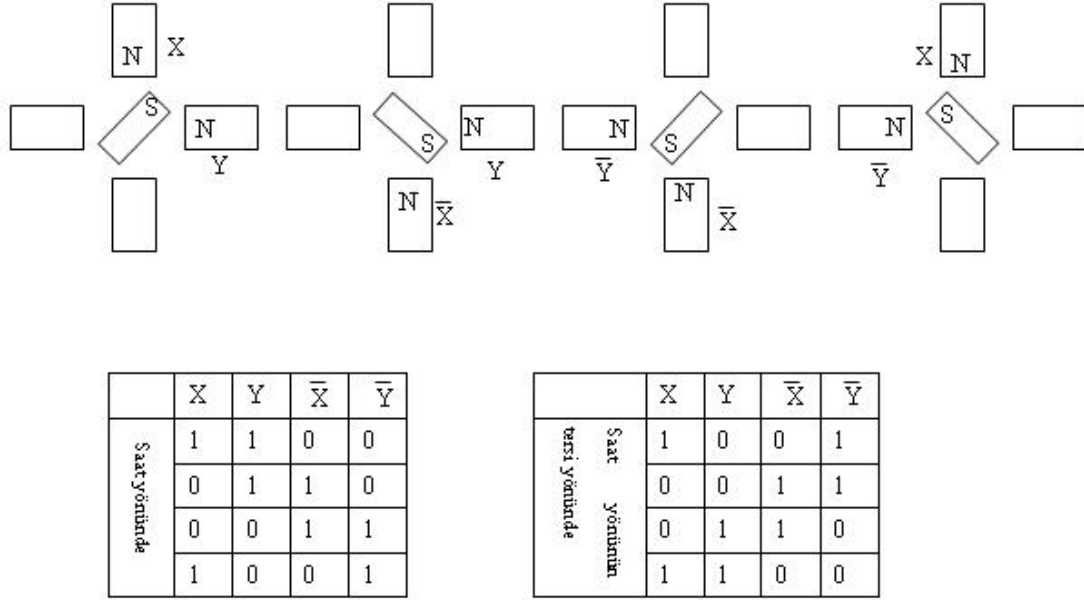
	X	Y	$\bar{X}$	$\bar{Y}$
Saat yönünde	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1

	X	Y	$\bar{X}$	$\bar{Y}$
Saat yönünün tersi yönde	0	0	0	1
	0	0	1	0
	0	1	0	0
	1	0	0	0

Şekil 2.4: Ünipolar adım motorunun 1 fazlı olarak sürülmesi

Bu metodla çalışma prensibini daha iyi anlayabiliriz. Ancak bu metod adım motorların tork ve durma karakteristikleri iyi olmadığından çok fazla kullanılmaz.

### 2.2.2. Adım Motorunun 2 Fazlı Sürülmesi



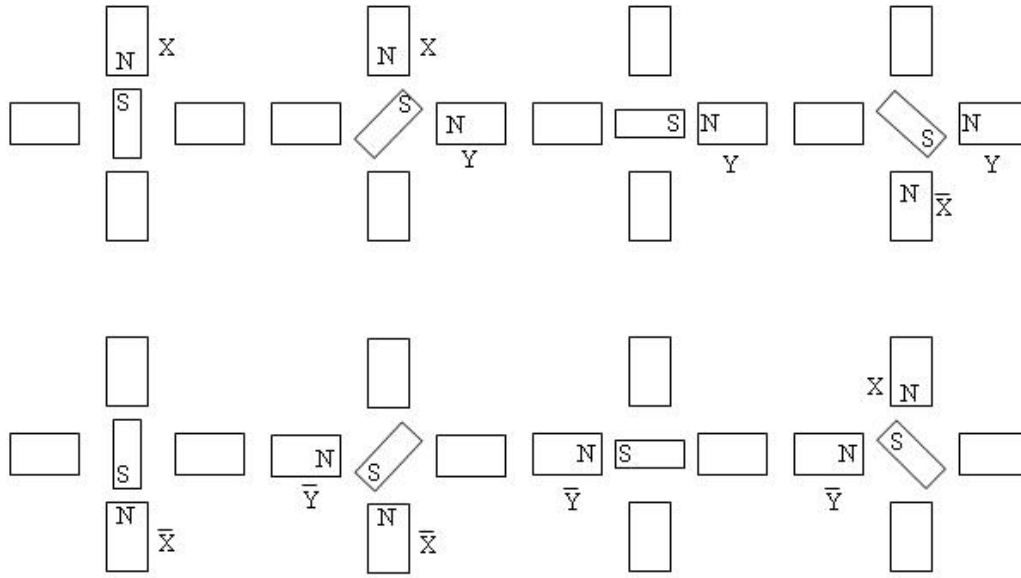
Şekil 2.5: Ünipolar adım motorunun 2 fazlı olarak sürülmesi

Bu metotta her iki bobin sıra ile enerjilendiğinde beraber mıknatıslanarak zıt kutuplu rotoru eşit kuvvetle kendilerine doğru (iki bobinin ortasına ) çekerek rotorun 1 adım dönmesini sağlarlar. Bu metotta motorun rotoru 2 bobin tarafından çekildiği için tork daha yüksektir. Örneğin bir robot kolunun kontrolü adımli motorla yapıldığında torkun yüksek olması robot kolunun daha fazla yükü kaldırmasını sağlar. Genellikle bu metod kullanılmaktadır çünkü torkları yüksek ve durma karakteristikleri iyidir.

### 2.2.3. Adım Motorunun 1 ve 2 Fazlı Sürülmesi

Bu metotta bobinler önce tek sonra çift enerjilenerek rotoru yarım adım döndürürler.





	X	Y	$\bar{X}$	$\bar{Y}$
Saat yönünde	1	0	0	0
	1	1	0	0
	0	1	0	0
	0	1	1	0
	0	0	1	0
	0	0	1	1
	0	0	0	1
	1	0	0	1

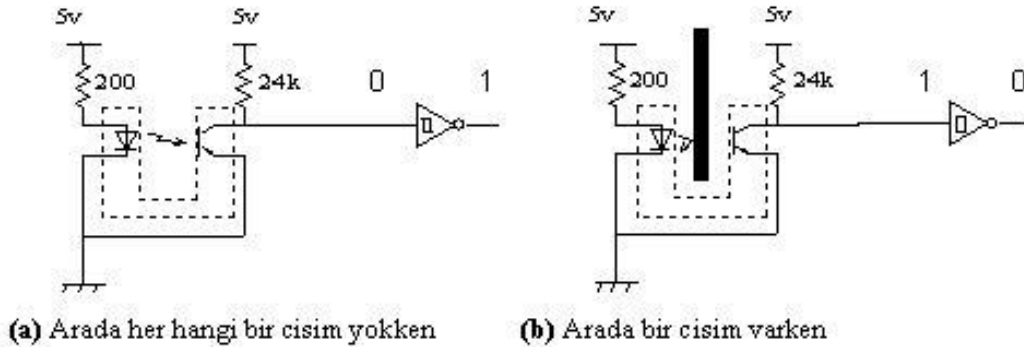
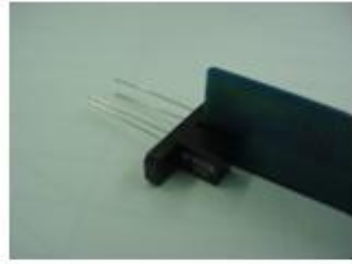
	X	Y	$\bar{X}$	$\bar{Y}$
Saat yönünün tersi yönünde	1	0	0	1
	0	0	0	1
	0	0	1	1
	0	0	1	0
	0	1	1	0
	0	1	0	0
	1	1	0	0
	1	0	0	0

Şekil 2.6: Unipolar adım motorunun 1-2 fazlı olarak sürülmesi

## 2.2.4. Adım Motorunun Pozisyon Kontrolü

Adım motoru pozisyon kontrolü için foto sensörler kullanılabilir. Şekil 2.7’de foto sensör iç yapısı görülmektedir.

### Foto sensörler



Şekil 2.7: Foto sensörler

Foto sensör LED ile foto transistörün birleşiminden meydana gelmektedir. Çıkışı schmitt-trigger not kapısına bağlanır. Not kapısı çıkışı arada her hangi bir cisim yok iken “1” (Şekil a), arada bir parça var ise “0” (Şekil b) olmaktadır.

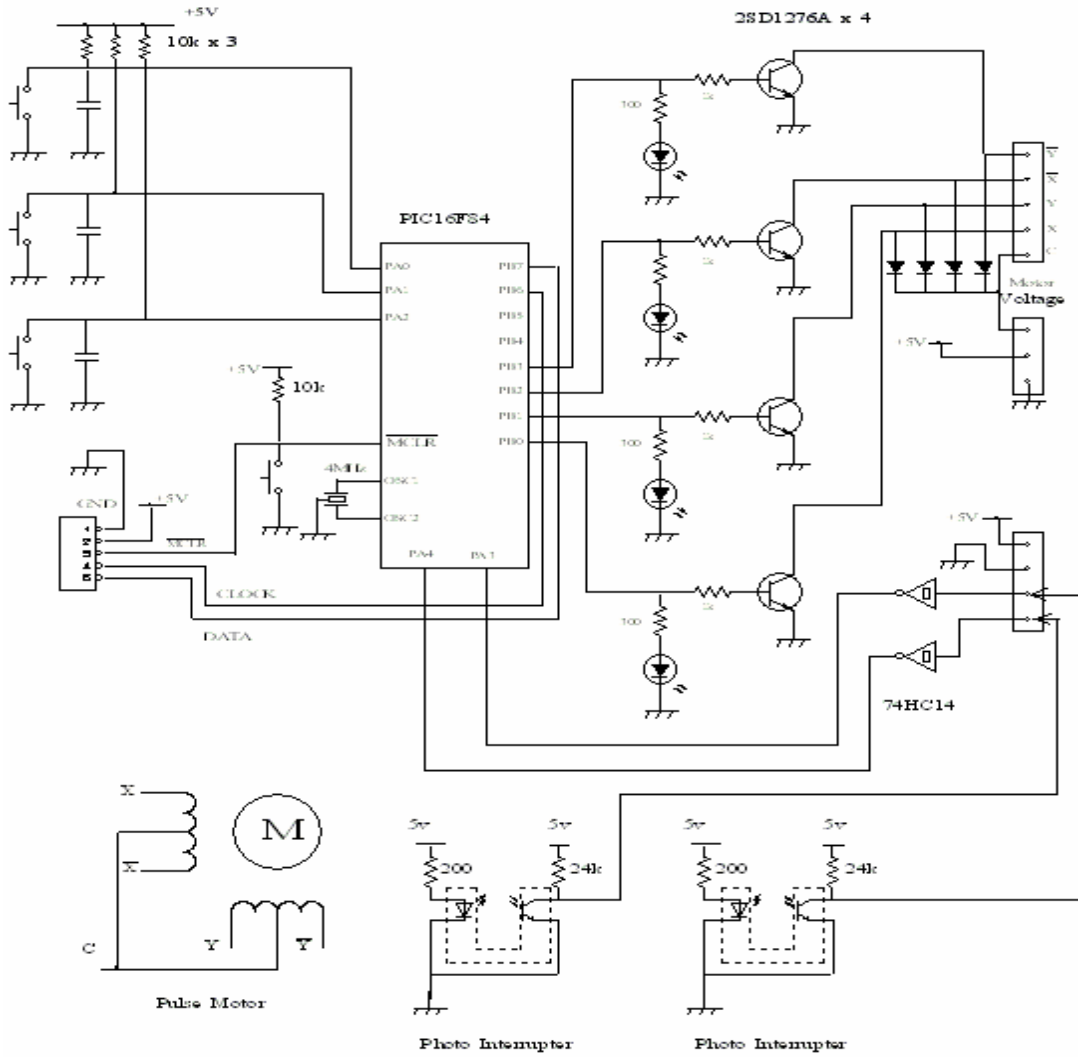
## UYGULAMA FAALİYETİ-1

### 1. Kullanılan Araç ve Gereçler

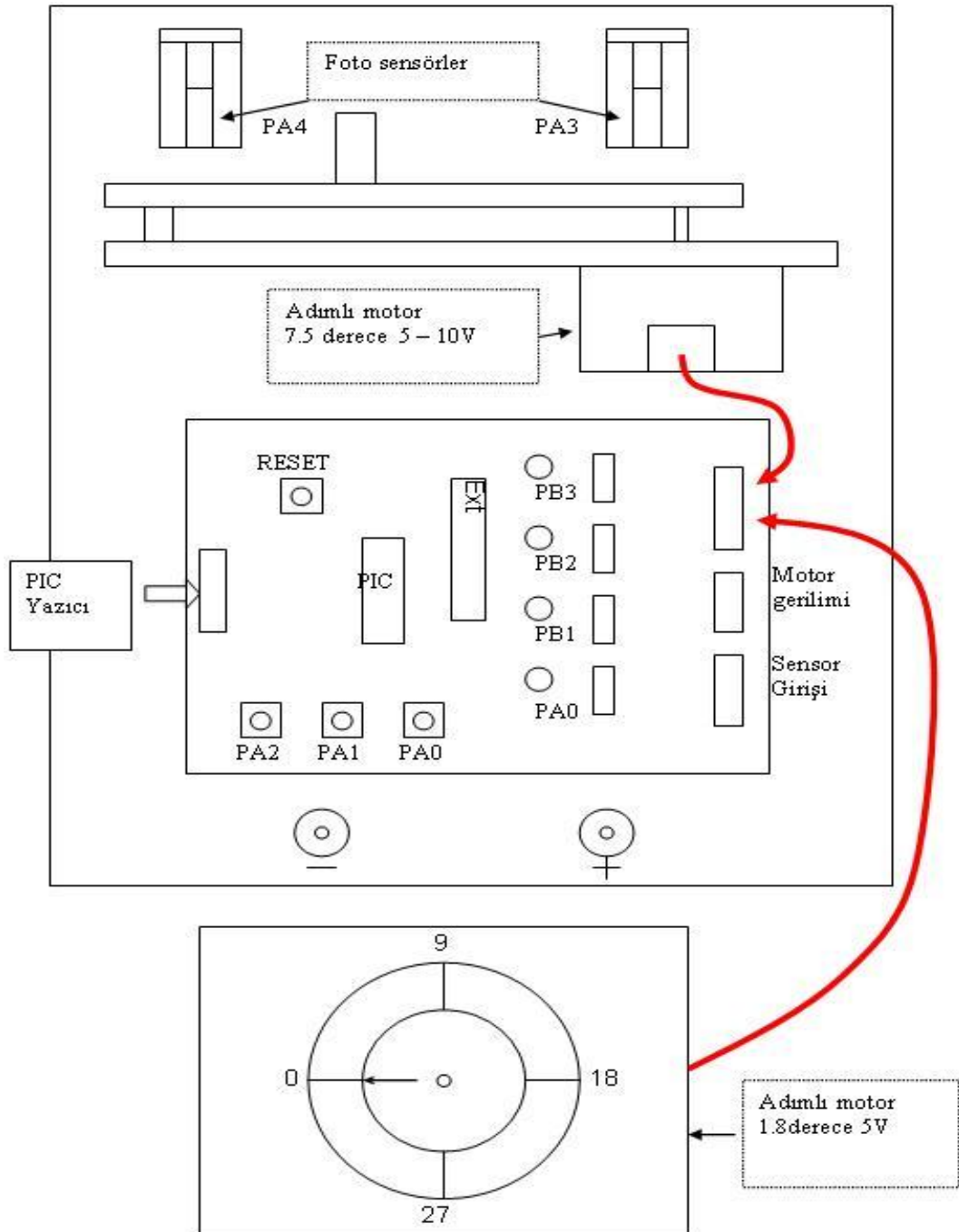
- a) PIC Programlayıcıyı Devresi 1
- b) Adım Motoru Eğitim Seti 1
- c) Adım Motoru 1
- d) Bağlantı Kabloları 2

### 2. Uygulama Devreleri

Uygulama için gerekli devreler diğer sayfada verilmiştir.



Şekil 2.8: Adım motoru uygulama devre şeması



Şekil 2.9: Eğitim uygulama seti üst görünüşü

<b>İşlem Basamakları</b>	<b>Öneriler</b>
<p>Ø Şekil 2.8'deki deney setini kullanarak Şekil 2.9'da gösterilen 1.8 derece 5V adım motorunu sete takınız.</p> <p>Ø Devreyi Öğretmeninize kontrol ettiriniz.</p> <p><b>Program Uygulaması-1</b></p> <p>Ø PA0 butonuna basılırsa adım motorunu 1 faz saat yönünde döndüren program yazınız.</p> <p>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</p> <p><b>Program Uygulaması-2</b></p> <p>Ø PA0 butonuna basılırsa adım motorunu 2 faz saat yönünde döndüren programı yazınız.</p> <p>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</p> <p><b>Program Uygulaması-3</b></p> <p>Ø PA0 butonuna basılırsa adım motorunu 1-2 faz saat yönünde döndüren programı yazınız.</p> <p>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</p> <p><b>Program Uygulaması-4</b></p> <p>Ø Açıklama-1'deki koşullar için gereken programı yazınız.</p> <p>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</p> <p><b>Program Uygulaması-5</b></p> <p>Ø Adım motorunu 360 derece döndüren programı yazınız.</p> <p>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</p>	<p>Ø İletkenleri kısa devre etmemeye özen gösteriniz.</p> <p>Ø 4. program için açıklama-1'de yazılan çalışmayı sırasıyla gerçekleştiriniz.</p>

#### **Açıklama-1**

PA0 butonuna basılırsa

>> Adım motoru saat yönünde 2 faz döner.

PA1 butonuna basılırsa

>> Adım motoru saat yönünün tersinde 2 faz döner.

PA2 butonuna basılırsa

>> Adım motoru durur.

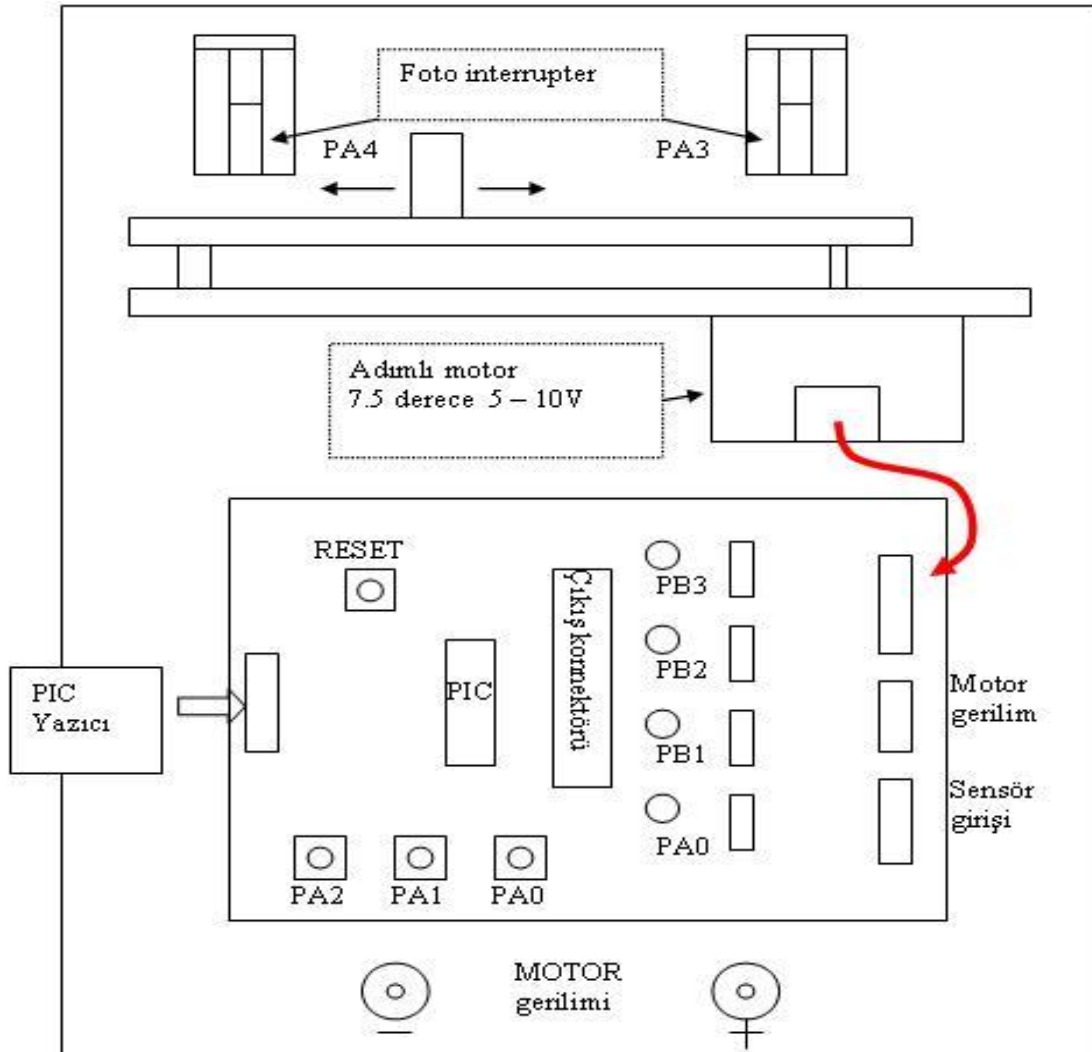
## UYGULAMA FAALİYETİ-2

### 1. Kullanılan Araç ve Gereçler

- |                              |   |
|------------------------------|---|
| a) PIC Programlayıcı Devresi | 1 |
| b) Adım Motoru Eğitim Seti   | 1 |
| c) Adım Motoru               | 1 |
| d) Foto sensör               | 1 |
| e) Bağlantı Kabloları        | 2 |

### 2. Uygulama Devreleri

Uygulama için gerekli devreler diğer sayfada verilmiştir.



Şekil 2.10: Step motor eğitim seti

<b>İşlem Basamakları</b>	<b>Öneriler</b>
<ul style="list-style-type: none"><li>Ø Şekil 2.10'daki bağlantıyı kurunuz.</li><li>Ø Bağlantıları öğretmeninize kontrol ettiriniz.</li></ul> <p><b>Program Uygulaması-6</b></p> <ul style="list-style-type: none"><li>Ø Açıklama-1'de istenen çalışmaları gerçekleştiren programı yazınız.</li><li>Ø Programı PIC'e yükleyip devre çalışmasının doğruluğunu kontrol ediniz.</li></ul>	<ul style="list-style-type: none"><li>Ø İletkenleri kısa devre etmemeye özen gösteriniz.</li><li>Ø 6. program için açıklama-1'de yazılan çalışmayı sırasıyla gerçekleştiriniz.</li></ul>

#### **Açıklama-1:**

- a) PA0 başlatma butonuna basıldığında adım motoru saat yönünde dönmeye başlar.
- b) Bant sola doğru hareket ederken bant üzerindeki parça PA4'deki foto sensöre gelince motor 1 saniye durur ve sonra saat yönünün tersinde dönmeye başlar.
- c) Bant sağa doğru hareket ederken bant üzerinde parça PA3'deki foto sensöre gelince motor 1 saniye durur ve tekrar saat yönünde dönmeye başlar. Bu işlem devamlı tekrar eder.

## ÖLÇME VE DEĞERLENDİRME

### OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Geri besleme olmaksızın pozisyon kontrolü istenen uygulamalarda aşağıdaki motorlardan hangisini kullanmak uygun olur?  
A) AC motor      B) Fırçalı DC Motor      C) Lineer Motor      D) Adım Motoru
2. Adım motorunun dönüş hızı aşağıdakilerden hangisine bağlıdır?  
A) Faz sayısına      B) Faz Frekansına  
C) Sargı çapına      D) Sargı Uzunluğuna
3. Bir adım motorunun adım açısı 7,5 derece ise motorun bir tam turu tamamlaması için kaç adım hareket etmesi gereklidir?  
A) 12      B) 24      C) 48      D) 360
4. Aşağıdakilerden hangisi adım motoru sürme metodlarından değildir?  
A) 1 fazlı      B) 2 fazlı  
C) 3 fazlı      D) 1-2 fazlı
5. Adım motorunun 2 fazlı sürülmesinin 1 fazlı sürülmesine göre en önemli üstünlüğü aşağıdakilerden hangisidir?  
A) Daha hızlı çalışması      B) Daha yüksek tork değerinin olması  
C) Daha kararlı olması      D) Daha az akım çekmesi

### DEĞERLENDİRME

Eğer soruları doğru olarak yanıtlatabiliyorsanız bu öğrenme faaliyetini tamamlayabilirsiniz. Yanlış cevaplarınız varsa ilgili bilgileri tekrar ediniz.



# MODÜL DEĞERLENDİRME

## YETERLİK ÖLÇME

Modülde yaptığınız uygulamaları aşağıdaki tabloya göre değerlendiriniz.

<b>Açıklama:</b> Aşağıda listelenen yeterliklere sahipseniz Evet sütununa, değilseniz Hayır sütununa X işareti yazınız.		
<b>Değerlendirme Ölçütleri</b>	<b>Evet</b>	<b>Hayır</b>
Ø Mikrodenetleyici ile mekanik röleyi kontrol edebiliyor musun?		
Ø Mikrodenetleyici ile ışık kontrollü röleyi kontrol edebiliyor musun?		
Ø DA motorunu ileri-geri kontrol edebiliyor musun?		
Ø DA motorunun hız kontrolünü PWM ile gerçekleştirebiliyor musun?		
Ø Mikrodenetleyici kontrolü ile adım motorunu istenilen adım kadar döndürüp durdurabiliyor musun?		
Ø Adım motoru bir yönde dönerken foto sensörden gelen bilgiye göre motoru durdurup diğer yöne doğru dönmesini sağlayabiliyor musun?		

## DEĞERLENDİRME

Cevaplarınızın tümü Evet ise modül hedeflerine ulaştığınız demektir. Hayır cevaplarınız var ise ilgili öğrenme faaliyetlerini tekrar ediniz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

### 1.1. Uygulama Faaliyeti -1 Cevap Anahtarı

#### 1.1.1. Uygulama-1

LIST P=16F84,R=DEC ;PIC 16F84 kullanılacaktır  
INCLUDE "P16F84.INC" ; P16F84.INC komut kütüphanesini programa ekler

```
BSF STATUS,RP0 ;Bank 1'e geçilir
MOVLW H'1F'
MOVWF TRISA ;PORTA'nın bütün portları giriş seçilir
CLRF TRISB ; PORTB'nin bütün portları çıkış seçilir
BCF STATUS,RP0 ; Bank 0'a geçilir
CLRF PORTB ; PORTB'yi temizle

ON BTFSZ PORTA,0 ;PA0 Butonuna basıldı mı? (PA0 sıfır oldu mu?)
GOTO OFF ;Hayır PA0 sıfır olmadı, OFF etiketine git
MOVLW H'01' ;Evet PA0 sıfır oldu, W B H'01'
MOVWF PORTB ; PORTB B W
GOTO ON ; ON etiketine git
OFF MOVLW H'00' ; W B H'00'
MOVWF PORTB ;PORTB B W
GOTO ON ; ON etiketine git
STP GOTO STP
END
```

#### 1.1.2. Uygulama -2

LIST P=16F84,R=DEC ;PIC 16F84 kullanılacaktır  
INCLUDE "P16F84.INC" ; P16F84.INC komut kütüphanesini programa ekler

```
BSF STATUS,RP0 ;Bank 1'e geçilir
MOVLW H'1F'
MOVWF TRISA ;PORTA'nın bütün portları giriş seçilir
CLRF TRISB ; PORTB'nin bütün portları çıkış seçilir
BCF STATUS,RP0 ; Bank 0'a geçilir
CLRF PORTB ; PORTB'yi temizle

PA0 BTFSZ PORTA,0 ;PA0 ON?(PA0 << Low)
GOTO PA1 ;PA0 ON >> PA1 etiketine git
MOVLW H'00' ;PA0 OFF >> PB0 = 00H
MOVWF PORTB ; Lamba sönmük
GOTO PA0
```

```

PA1  BTFS    PORTA,1    ;PA1 ON? (PA1 << LOW)
      GOTO    PA0SW     ;PA1 ON >> PA0 etiketine git
      MOVLW   H'01'     ;RA1 OFF >> PORTB = 01H
      MOVWF   PORTB     ;Lamba yanar
      GOTO    PA1SW
STP  GOTO    STP
      END

```

## UYGULAMA FAALİYETİ -2 CEVAP ANAHTARI

### 1.2.1. Uygulama -3

```

LIST      P=16F84,R=DEC ;PIC 16F84 kullanılacaktır
INCLUDE   "P16F84.INC" ; P16F84.INC komut kütüphanesini programa ekler

BSF       STATUS,RP0   ;Bank 1'e geçilir
MOVLW    H'1F'
MOVWF    TRISA        ;PORTA'nın bütün portları giriş seçilir
CLRF     TRISB        ; PORTB'nin bütün portları çıkış seçilir
BCF      STATUS,RP0   ; Bank 0'a geçilir
CLRF     PORTB        ; PORTB'yi temizle

ON  BTFS    PORTA,0    ;PA0 Butonuna basıldı mı? (PA0 sıfır oldu mu?)
      GOTO    OFF      ;Hayır PA0 sıfır olmadı, OFF etiketine git
      MOVLW   H'04'    ;Evet PA0 sıfır oldu, W B H'04'
      MOVWF   PORTB    ; PORTB B W
      GOTO    ON       ; ON etiketine git

OFF MOVLW   H'00'      ; W B H'00'
      MOVWF   PORTB    ;PORTB B W
      GOTO    ON       ; ON etiketine git

STP  GOTO    STP
      END

```

### 1.2.2. Uygulama -4

```

LIST      P=16F84,R=DEC ;PIC 16F84 kullanılacaktır
INCLUDE   "P16F84.INC" ; P16F84.INC komut kütüphanesini programa ekler

BSF       STATUS,RP0   ;Bank 1'e geçilir
MOVLW    H'1F'
MOVWF    TRISA        ;PORTA'nın bütün portları giriş seçilir

```

```

CLRF      TRISB      ; PORTB'nin bütün portları çıkış seçilir
BCF       STATUS,RP0 ; Bank 0'a geçilir
CLRF      PORTB     ; PORTB'yi temizle

PA0 BTFSS   PORTA,0  ;PA0 ON?(PA0 << Low)
      GOTO   PA1     ;PA0 ON >> PA1 etiketine git
      MOVLW  H'00'   ;PA0 OFF >> PB0 = 00H
      MOVWF  PORTB   ; Lamba sönmük
      GOTO   PA0

PA1 BTFSS   PORTA,1  ;PA1 ON? (PA1 << LOW)
      GOTO   PA0SW   ;PA1 ON >> PA0 etiketine git
      MOVLW  H'01'   ;RA1 OFF >> PORTB = 01H
      MOVWF  PORTB   ;Lamba yanar
      GOTO   PA1SW

STP GOTO    STP
      END

```

### 1.2.3. Uygulama-5

```

LIST      P=16F84,R=DEC
          INCLUDE "P16F84.INC"

COUNT1  EQU    H'10'
COUNT2  EQU    H'11'

          BSF    STATUS,RP0
          MOVLW  H'1F'
          MOVWF  TRISA
          CLRF   TRISB
          BCF    STATUS,RP0
          CLRF   PORTB

PA0SW    BTFSS   PORTA,0
          GOTO   PA1SW
          MOVLW  H'00'
          MOVWF  PORTB
          GOTO   PA0SW

PA1SW    BTFSS   PORTA,1
          GOTO   PA0SW
          MOVLW  H'04'
          MOVWF  PORTB
          CALL   TIMER
          MOVLW  H'00'
          MOVWF  PORTB

```

```

CALL    TIMER
GOTO   PA1SW

STP     GOTO   STP

TIMER   ; 0.5S TIMER
        MOVLW D'250'    ;1CLOCK
        MOVWF COUNT2    ;1CLOCK
DLY1    MOVLW D'200'    ;1CLOCK
        MOVWF COUNT1    ;1CLOCK
DLY2    GOTO    $+1      ;2CLOCK
        GOTO    $+1      ;2CLOCK
        GOTO    $+1      ;2CLOCK
        NOP           ;1CLOCK
        DECFSZ COUNT1,1 ;1(2)CLOCK
        GOTO    DLY2    ;2CLOCK
        DECFSZ COUNT2,1 ;1(2)CLOCK
        GOTO    DLY1    ;2CLOCK
        RETURN      ;2CLOCK
        END

```

### UYGULAMA FAALİYETİ -3 CEVAPANAHTARI

#### 1.3.1. Uygulama-6

```

LIST    P=16F84,R=DEC
INCLUDE "P16F84.INC"

BSF     STATUS,RP0
MOVLW   H'1F'
MOVWF   TRISA
CLRF    TRISB
BCF     STATUS,RP0
CLRF    PORTB

PA0SW   BTFSS   PORTA,0
        GOTO   PA1SW
        MOVLW   H'00'
        MOVWF   PORTB    ;Motor durur
        GOTO   PA0SW

PA1SW   BTFSS   PORTA,1
        GOTO   PA0SW
        MOVLW   H'02'
        MOVWF   PORTB    ;FET çıkışındaki motor çalışır (döner)
        GOTO   PA1SW

STP     GOTO   STP
        END

```

### 1.3.2. Uygulama -7

```
LIST P=16F84,R=DEC
INCLUDE "P16F84.INC"
BSF STATUS,RP0
CLRF TRISB
MOVLW B'00011111'
MOVWF TRISA
BCF STATUS,RP0

BEKE MOV LW B'00001100' ;motor için fren bilgisi
MOVWF PORTB ;motor frende bekler
BTFSS PORTA,0 ;PA0 girildi mi?
GOTO ILERI ;PA0 on >> goto ILERI
BTFSS PORTA,1 ; PA1 girildi mi?
GOTO GERI ; PA1 on >> goto GERI
GOTO BEKLE ;PA0 VE PA1 girilmezse>>goto BEKLE
ILERI MOV LW B'00001000' ; DC motor için ileri (saat yönü) bilgisi
MOVWF PORTB ; motor saat yönünde döner
BTFSS PORTA,2 ;PA2 butonu girildi mi?
GOTO BEKLE ;PA2 on >> goto BEKLE
GOTO ILERI

GERI MOV LW B'00000100' ; DC motor için geri (saat yönü tersi) bilgisi
MOVWF PORTB ; motor saat yönünün tersinde döner
BTFSS PORTA,2 ; PA2 butonu girildi mi?
GOTO STOP ; PA2 on >> goto BEKLE
GOTO GERI

STP GOTO STP
END
```

### 1.3.3. Uygulama-8

```
LIST P=16F84,R=DEC
INCLUDE "P16F84.INC"
TUR EQU H'10'
COUNT1 EQU H'11'
COUNT2 EQU H'12'

BSF STATUS,RP0
CLRF TRISB
MOVLW B'00011111'
MOVWF TRISA
BCF STATUS,RP0

BEKLE MOV LW B'00001100' ;motor için fren bilgisi
```

```

                MOVWF    PORTB    ;motor frende beklemededir
                BTFSC   PORTA,0   ;PA0 girildi mi?
                GOTO    BEKLE     ;PA0 girilmezse BEKLE etiketine git
BASLA          CALL    STOP      ;PA0 girilirse 1saniye bekle
                CLRF    TUR       ;motor TUR = 0
SAATYON        MOVLW   B'00001000' ; DC motor için saat yönü bilgisi
                MOVWF   PORTB    ;motor saat yönünde döner
                BTFSC   PORTA,3   ; Foto sensörden gelen bilgi = 0 ise 1 satır
                ;atla
                CALL    ARTIR     ; Foto sensörden gelen bilgi = 1 ise
                ;Iartıracak alt programı çağır
                MOVLW   D'10'     ; motor tur sayısı = 10
                SUBWF   TUR,W     ; motor tur sayısı= 10 oldu mu?
                BTFSS   STATUS,Z  ; (Z=1 ?) işlem sonucu sıfır ise Zero biti 1
                ;olur
                GOTO    SAATYON   ;tur sayısı 10 olmamış,saat yönünde
                ;dönmeye devam et
                CALL    STOP      ;tur sayısı 10 olduğu için 1 saniye bekle
                CLRF    TUR       ; motor TUR = 0
TERSYON        MOVLW   B'00000100' ; DC motor için saat yönü tersi bilgisi
                MOVWF   PORTB    ; motor saat yönünün tersinde döner
                BTFSC   PORTA,3   ; Foto sensörden gelen bilgi = 0 ise 1 satır
                ;atla
                CALL    ARTIR     ; Foto sensörden gelen bilgi = 1 ise
                ;Iartıracak alt programı çağır
                MOVLW   D'10'     ; motor tur sayısı = 10
                SUBWF   TUR,W     ; motor tur sayısı= 10 oldu mu?
                BTFSS   STATUS,Z  ; (Z=1 ?) işlem sonucu sıfır ise Zero biti 1
                ;olur
                GOTO    TERSYON   ;tur sayısı 10 olmamış,saat yönünün tersinde
                ;dönmeye devam et
                GOTO    BASLA     ;tur sayısı 10 olduğu için tekrar başlaya git

ARTIR          INCF    TUR,F     ;motor tur sayısını 1 artır
DISK_BEKLE    BTFSC   PORTA,3   ;Foto sensörden gelen bilgi =1 oldu mu?
(motor diski çevirerek foto sensörün diskteki delikten kurtulması beklenir.Aksi halde
motorun dönüş hızından daha hızlı olan mikrodenetleyici bu delik daha dönüp geçmeden
birden fazla tur sayacaktır.)
                GOTO    DISK_BEKLE
                RETURN           ;alt programdan çık

STOP          MOVLW   B'00001100' ;motor için fren bilgisi
                MOVWF   PORTB    ;motor frende beklemededir
                CALL    TIMER    ;0.5 saniyelik TIMER alt programı çağrılır
                CALL    TIMER    ;0.5 saniyelik TIMER alt programı çağrılır
                RETURN           ;alt programdan çık

```

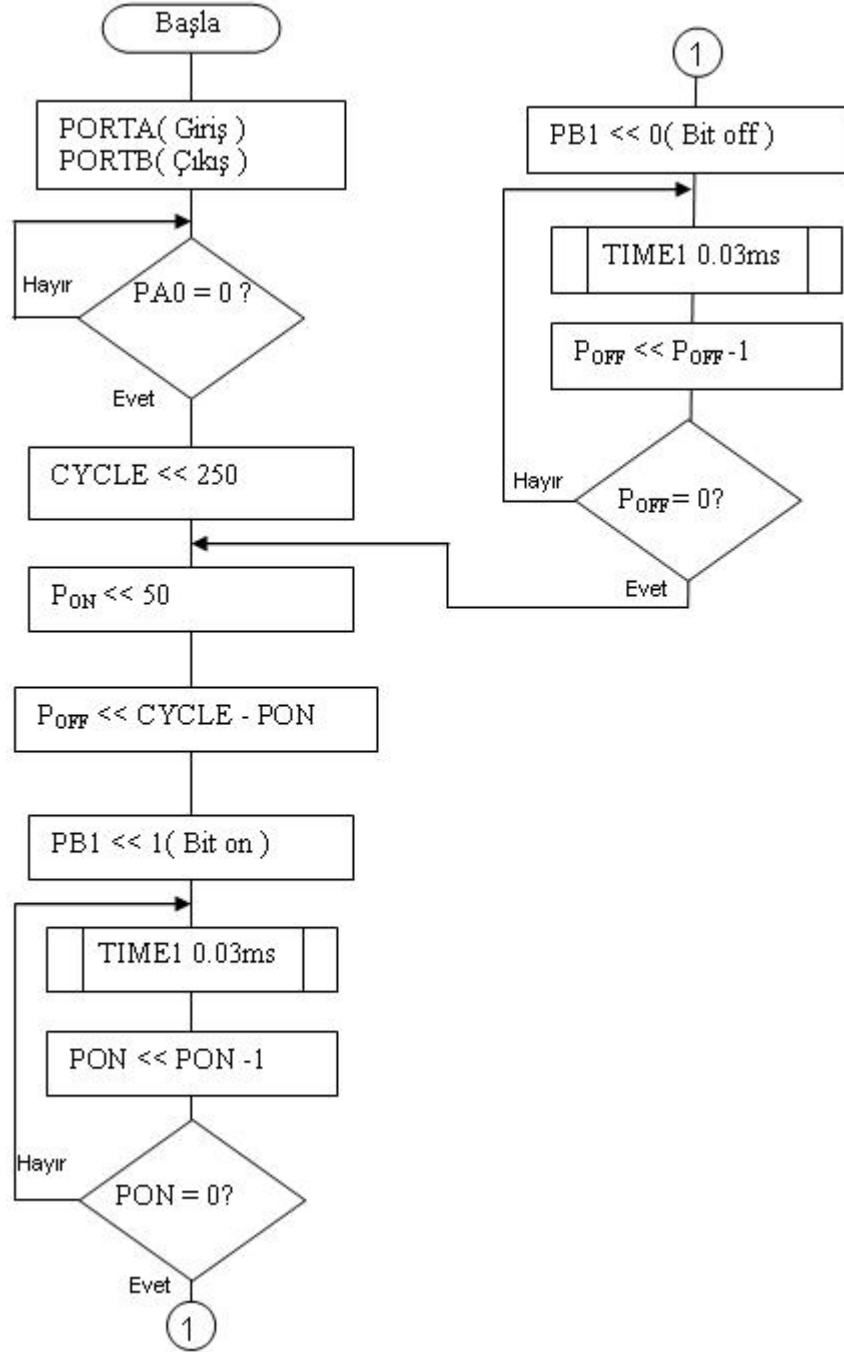
```
;0.5 saniyelik zaman alt programıdır
TIMER      MOVLW    D'200'
            MOVWF    COUNT2
DLY1       MOVLW    D'250'
            MOVWF    COUNT1
DLY        GOTO     $+1
            GOTO     $+1
            GOTO     $+1
            DECFSZ   COUNT1,F
            GOTO     DLY
            DECFSZ   COUNT2,F
            GOTO     DLY1
            RETURN

            END
```



### 1.3.4. Uygulama-9

Akış diyagramı



## Uygulama-9'un Program listesi

```
*****
;
;   Bu program PWM programıdır.
;*****
                LIST      P=16F84,R=DEC
                INCLUDE   "P16F84.INC" ;file include
SAYKIL         EQU       H'10'
PON            EQU       H'11'
POFF           EQU       H'12'
COUNT1       EQU       H'13'
COUNT2       EQU       H'14'

                BSF       STATUS,RP0
                MOVLW    B'00011111'
                MOVWF    TRISA
                CLRF     TRISB
                BCF      STATUS,RP0
                CLRF     PORTB

MAIN
                MOVLW    D'250'          ;Wreg << PWM cycle = 250
                MOVWF    SAYKIL         ;SAYKIL << W reg

PWM
                MOVLW    D'50'          ;Wreg << PWM ON time = 50
                MOVWF    PON            ;P_ON << W reg
                SUBWF    SAYKIL,W       ;Wreg << PWM saykıl - PWM ON
                MOVWF    POFF          ;P_OFF(PWM OFF time) << Wreg
;*****
; PWM on time 0.03ms * 50 = yaklaşık 1.5ms
;*****
                BSF      PORTB,1        ;PORTB bit1 = 1
PWM0N          CALL     TIME1           ;0.03ms gecikme
                DECFSZ  PON,F          ;P_ON-1 >> P_ON = 0? Eğer P_ON=0 ise atla
                GOTO    PWM0N         ; Eğer P_ON=0 değilse PWM0N' a git
;*****
; PWM off time 0.03ms * 200 = yaklaşık 6ms
;*****
                BCF     PORTB,1        ;PORTB bit1 = 0
PWMOFF        CALL     TIME1           ;0.03ms gecikme
                DECFSZ  POFF,1        ;P_OFF-1 >> P_OFF =0?
                GOTO    PWMOFF        ; Eğer P_OFF=0 değilse PWMOFF'a git
                GOTO    PWM           ; Eğer P_OFF=0 ise PWM' e git

TIME1          ; yaklaşık 0.03ms gecikme
;*****
;* 1+1+1*(1+1+2*(2+2+2+1+1+2))+1+1+2)=32us
;*****
```

```

                MOVLW    D'1'          ; 1clock
                MOVWF   COUNT1        ; 1clock
DLY1           MOVLW    D'2'          ; 1clock
                MOVWF   COUNT2        ; 1clock
DLY2           GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                NOP                    ; 1clock
                DECFSZ  COUNT2,1      ; 1(2)clock
                GOTO     DLY2         ; 2clock
                DECFSZ  COUNT1,1      ; 1(2)clock
                GOTO     DLY1         ; 2clock
                RETURN                ; 2clock
                END

```

### 1.3.5. Uygulama-10

```

                LIST      P=16F84,R=DEC
                INCLUDE   "P16F84.INC"

SAYKIL        EQU       H'10'
YEDEK_OFF     EQU       H'11'
YEDEK_ON      EQU       H'12'
PWM_ON        EQU       H'13'
COUNT1       EQU       H'14'
COUNT2       EQU       H'15'

                BSF      STATUS,RP0
                CLRF     TRISB
                MOVLW    B'00011111'
                MOVWF   TRISA
                BCF      STATUS,RP0
                CLRF     PORTB

MAIN          BTFSC     PORTA,0      ;başlatmak için PA0 butonuna girildi mi?
                GOTO     MAIN        ;girilmezse MAIN etiketine git
                CALL     TIME        ;siviçte oluşan gerilim dalgalanmasını
mikrodenetleyicinin okumaması için zaman çağrılır
                BTFSS    PORTA,0      ;PA0 butonundan el çekildi mi?
                GOTO     MAIN        ;PA0 butonu bırakılmazsa MAIN'e git
                MOVLW    D'250'      ;PWM saykıl sayısı = 250 >> W reg
                MOVWF   SAYKIL       ;W reg >> SAYKIL
                MOVLW    D'125'      ;ilk değer= 125(PWM on time) >> Wreg
                MOVWF   PWM_ON       ;Wreg >> PWM_ON

KONTROL       BTFSS    PORTA,0      ;PA0 butonuna girilmezse bir satır atla
                GOTO     STOP        ;PA0 butonuna girilirse STOP'a git

```

```

                BTFS    PORTA,1    ; PA1 butonuna girilmezse bir satır atla
                GOTO    HIZLAN    ; PA0 butonuna girilirse HIZLAN'a git
                BTFS    PORTA,2    ; PA2 butonuna girilmezse bir satır atla
                GOTO    YAVAS     ; PA0 butonuna girilirse YAVAS'a git

PWM            MOVF    PWM_ON,W   ; PWM_ON >> Wreg(PWM on time)
                MOVWF   YEDEK_ON  ; Wreg >> YEDEK_ON
                SUBWF   SAYKIL,W   ; SAYKIL - PWM_ON (Wreg) >> Wreg
                MOVWF   YEDEK_OFF  ; Wreg >> YEDEK_OFF (PWM off time)

                MOVLW   B'00000010' ; PWM için on bilgisi >> Wreg
                MOVWF   PORTB      ; PWM için on bilgisi için PB1=1 olur
PON            CALL    TIME1      ; 1 nümerik saykıl= 0.03ms için zaman
                DECFSZ  YEDEK_ON,F ; YEDEK_ON - 1 >> YEDEK_ON eğer
                YEDEK_ON=0 olursa 1 satır atla
                GOTO    PON        ; YEDEK_ON=0 değilse PON'a git
                MOVLW   B'00000000' ; PWM için off bilgisi >> Wreg
                MOVWF   PORTB      ; PWM için on bilgisi için PB1=0 olur
POFF           CALL    TIME1      ; 1 nümerik saykıl= 0.03ms için zaman
                DECFSZ  YEDEK_OFF,1 ; YEDEK_OFF - 1 >> YEDEK_OFF eğer
                YEDEK_OFF=0 olursa 1 satır atla
                GOTO    POFF       ; YEDEK_OFF=0 değilse POFF'a git
                GOTO    KONTROL    ; butonları KONTROL'e git

HIZLAN        INCF    PWM_ON,F    ; PWM_ON + 1 >> PWM_ON (max = 250)
                MOVLW   D'250'    ; Maksimum saykıl (250) >> Wreg
                SUBWF   PWM_ON,W   ; PWM_ON - Wreg >> Wreg
                BTFS    STATUS,Z   ; Wreg = 0 olduysa (Zero bit=0) 1 satır atla
                GOTO    PWM        ; PWM'e git
                DECF    PWM_ON,F   ; PWM_ON (250) - 1 = PWM_ON (249)
                GOTO    PWM        ; PWM'e git

YAVAS        DECF    PWM_ON,F    ; PWM_ON - 1 >> PWM_ON (min -0)
                BTFS    STATUS,Z   ; PWM_ON = 0 olduysa (Zero bit=0) 1 satır
                atla
                GOTO    PWM        ; PWM'e git
                INCF    PWM_ON,F   ; PWM_ON (0) + 1 = PWM_ON (1)
                GOTO    PWM        ; PWM'e git
STOP         GOTO    STOP

```

```

TIME1          ; yaklaşık 0.03ms'lik zaman programı
;*****
; * 1+1+1*(1+1+2*(2+2+2+1+1+2))+1+1+2)+1+2=32us *
;*****

```

```

                MOVLW    D'1'          ; 1clock
                MOVWF    COUNT1        ; 1clock
DLY1           MOVLW    D'2'          ; 1clock
                MOVWF    COUNT2        ; 1clock
DLY2           GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                NOP                    ; 1clock
                DECFSZ   COUNT2,1     ; 1(2)clock
                GOTO     DLY2          ; 2clock
                DECFSZ   COUNT1,1     ; 1(2)clock
                GOTO     DLY1          ; 2clock
                RETURN                   ; 2clock

```

TIME ; yaklaşık 40ms'lik zaman programı

```

;*****
;* 1+1+20*(1+1+20*(2+2+2+1+1+2))+1+1+2)+1+2=40126us *
;*****

```

```

                MOVLW    D'20'         ; 1clock
                MOVWF    COUNT1        ; 1clock
DL1           MOVLW    D'200'         ; 1clock
                MOVWF    COUNT2        ; 1clock
DL2           GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                GOTO     $+1           ; 2clock
                NOP                    ; 1clock
                DECFSZ   COUNT2,1     ; 1(2)clock
                GOTO     DL2           ; 2clock
                DECFSZ   COUNT1,1     ; 1(2)clock
                GOTO     DL1           ; 2clock
                RETURN                   ; 2clock
                END

```

#### 1.4. Ölçme Soruları Cevap Anahtarı

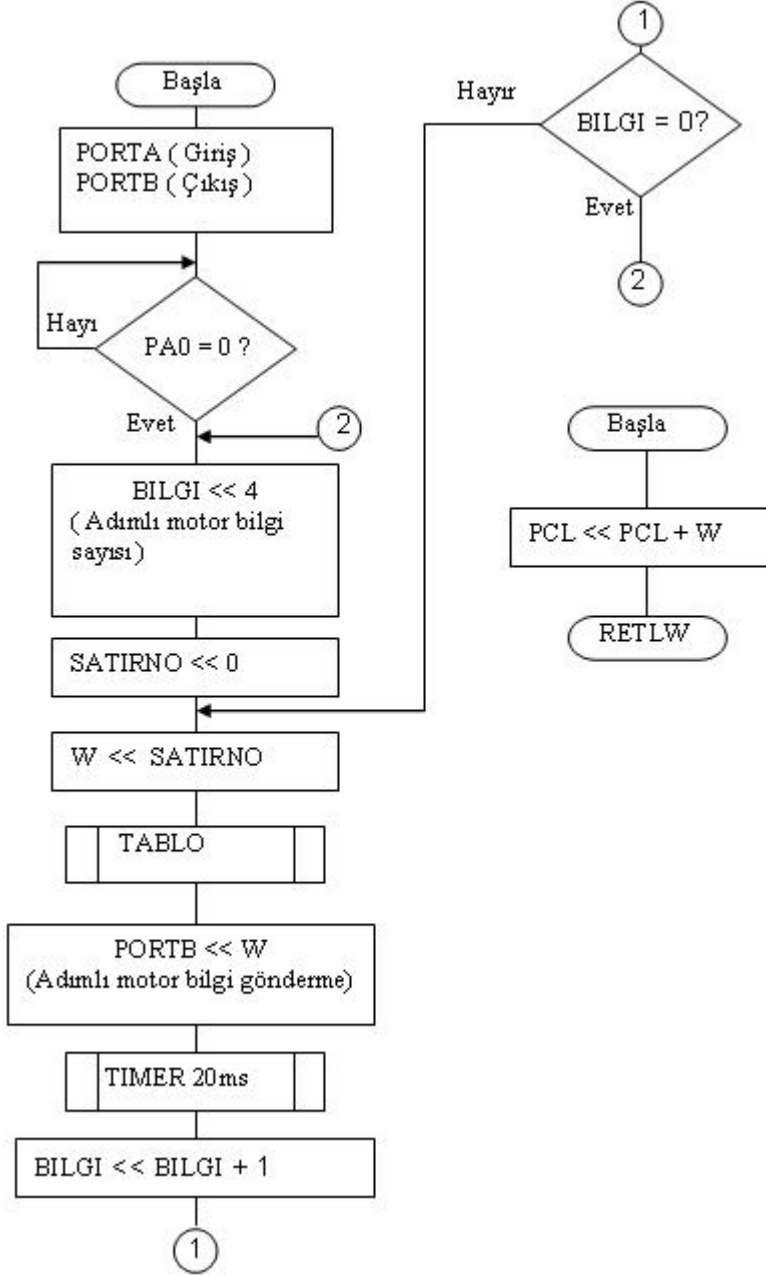
1-c 2-c 3-b 4-b 5-c 6-d

## ÖĞRENME FAALİYETİ -2 CEVAP ANAHTARLARI

### 2.1. Uygulama Faaliyeti -1 Cevap Anahtarı

#### 2.1.1. Uygulama -1

Programın akış diyagramı



```
*****
;*
;* RETLW komutunu kullanarak 1 fazlı adımlı motor kontrolü
;*
```

```

;*****
LIST      P=16F84,R=DEC
INCLUDE "P16F84.INC"
CNT1     EQU      H'10'      ; RAM ADRES
CNT2     EQU      H'11'      ;
BILGI    EQU      H'12'      ;
SATIRNO  EQU      H'13'      ;

        BSF      STATUS,RP0  ; Bank1'e geç
        MOVLW    B'11111'
        MOVWF    TRISA
        CLRF     TRISB        ; PORTB tamamı çıkış
        BCF      STATUS,RP0  ; Bank0'a geç
        CLRF     PORTB        ; PORTB = 0

SW       BTFSC   PORTA,0      ;PORTA Butonu basıldı mı ?
        GOTO    SW
LOOP     MOVLW   D'4'          ; tablodaki bilgi sayısı= 4
        MOVWF   BILGI         ; BILGI = 4
        CLRF   SATIRNO        ; SATIRNO = 0
LOOP1    MOVF    SATIRNO,W     ;W << SATIRNO
        CALL   TABLO          ; SATIRNO bilgi 0'dan 3'e
        MOVWF  PORTB          ;PORTB << W (motor bilgisi)
        CALL   TIMER          ; 20ms gecikme
        INCF   SATIRNO,F      ; SATIRNO << SATIRNO +1
        DECFSZ BILGI,F        ; BILGI << BILGI -1 = 0 ?
        ;
        GOTO   LOOP1
        GOTO   LOOP
STP      GOTO   STP

```

```

;*****
; adımlı motoru 1 fazlı sürmek için gerekli bilgilerin tablosu
;*****
TABLO    ADDWF   PCL,F        ;PCL ( program counter )+ W >>PCL
        RETLW   B'00000001'   ;1. adım bilgisi >> W (PCL+0)
        RETLW   B'00000010'   ;2. adım bilgisi >> W (PCL+1)
        RETLW   B'00000100'   ;3. adım bilgisi >> W (PCL+2)
        RETLW   B'00001000'   ;4. adım bilgisi >> W (PCL+3)

```

TIMER ;yaklaşık 5ms zamanlayıcı

```

;*****
;* 1+1+2*(1+1+250*(2+2+2+1+1+2))+1+1+2)+1+2=5017us *

```

\*\*\*\*\*

```
DLY1    MOVLW    D'2'           ; 1clock
        MOVWF   CNT1          ; 1clock
        MOVLW   D'250'        ; 1clock
        MOVWF   CNT2          ; 1clock
DLY2    GOTO    $+1           ; 2clock
        GOTO    $+1           ; 2clock
        GOTO    $+1           ; 2clock
        NOP                      ; 1clock
        DECFSZ  CNT2,1        ; 1(2)clock
        GOTO    DLY2          ; 2clock
        DECFSZ  CNT1,1        ; 1(2)clock
        GOTO    DLY1          ; 2clock
        RETURN                   ; 2clock
        END
```

### 2.1.2. Uygulama -2

\*\*\*\*\*

```
.*
.*
.* RETLW komutunu kullanarak 2 fazlı adımli motor kontrolü
.*
.*
*****
```

```
        LIST      P=16F84,R=DEC
        INCLUDE "P16F84.INC"
CNT1    EQU      H'10'        ; RAM ADRES
CNT2    EQU      H'11'        ;
BILGI   EQU      H'12'        ;
SATIRNO EQU      H'13'        ;

        BSF      STATUS,RP0   ; Bank1'e geç
        MOVLW   B'11111'
        MOVWF   TRISA
        CLRF    TRISB         ; PORTB tamamı çıkış
        BCF     STATUS,RP0    ; Bank0'a geç
        CLRF    PORTB         ; PORTB = 0

SW      BTFSZ   PORTA,0       ;PORTA Butonu basıldı mı ?
        GOTO    SW
LOOP    MOVLW   D'4'           ; tablodaki bilgi sayısı= 4
        MOVWF   BILGI         ; BILGI = 4
        CLRF    SATIRNO       ; SATIRNO = 0
LOOP1   MOVF    SATIRNO,W      ;W << SATIRNO
        CALL    TABLO         ; SATIRNO bilgi 0'dan 3'e
        MOVWF   PORTB         ;PORTB << W (motor bilgisi)
        CALL    TIMER         ; 20ms gecikme
```



```

                INCF     SATIRNO,F    ; SATIRNO << SATIRNO +1
                DECFSZ   BILGI,F     ; BILGI << BILGI -1 = 0 ?
                                ;
                GOTO     LOOP1
                GOTO     LOOP
STP             GOTO     STP

```

```

;*****
; adımlı motoru 2 fazlı sürmek için gerekli bilgilerin tablosu
;*****

```

```

TABLO         ADDWF    PCL,F      ;PCL ( program counter )+ W >>PCL
                RETLW   B'0000011' ;1. adım bilgisi >> W (PCL+0)
                RETLW   B'00000110' ;2. adım bilgisi >> W (PCL+1)
                RETLW   B'00001100' ;3. adım bilgisi >> W (PCL+2)
                RETLW   B'00001001' ;4. adım bilgisi >> W (PCL+3)

```

```

TIMER                                               ;yaklaşık 5ms zamanlayıcı

```

```

;*****
;* 1+1+2*(1+1+250*(2+2+2+1+1+2))+1+1+2)+1+2=5017us      *
;*****

```

```

                MOVLW   D'2'        ; 1clock
                MOVWF   CNT1        ; 1clock
DLY1           MOVLW   D'250'      ; 1clock
                MOVWF   CNT2        ; 1clock
DLY2           GOTO    $+1          ; 2clock
                GOTO    $+1          ; 2clock
                GOTO    $+1          ; 2clock
                NOP                    ; 1clock
                DECFSZ  CNT2,1      ; 1(2)clock
                GOTO    DLY2        ; 2clock
                DECFSZ  CNT1,1      ; 1(2)clock
                GOTO    DLY1        ; 2clock
                RETURN                ; 2clock
                END

```

### 2.1.3. Uygulama -3

```
*****
;
;*
; * RETLW komutunu kullanarak 1-2 fazlı adımlı motor kontrolü
;*
*****
                LIST      P=16F84,R=DEC
                INCLUDE "P16F84.INC"
CNT1            EQU      H'10'      ; RAM ADRES
CNT2            EQU      H'11'      ;
BILGI           EQU      H'12'      ;
SATIRNO         EQU      H'13'      ;

                BSF      STATUS,RP0  ; Bank1'e geç
                MOVLW   B'11111'
                MOVWF   TRISA
                CLRF    TRISB        ; PORTB tamamı çıkış
                BCF     STATUS,RP0   ; Bank0'a geç
                CLRF    PORTB        ; PORTB = 0

SW              BTFSB   PORTA,0      ;PORTA Butonu basıldı mı ?
                GOTO   SW
LOOP            MOVLW   D'8'         ; tablodaki bilgi sayısı= 8
                MOVWF  BILGI        ; BILGI = 8
                CLRF   SATIRNO      ; SATIRNO = 0
LOOP1          MOVF    SATIRNO,W     ; W << SATIRNO
                CALL   TABLO        ; SATIRNO bilgi 0'dan 3'e
                MOVWF  PORTB        ; PORTB << W (motor bilgisi)
                CALL   TIMER        ; 20ms gecikme
                INCF   SATIRNO,F    ; SATIRNO << SATIRNO +1
                DECFSZ BILGI,F      ; BILGI << BILGI -1 = 0 ?
                GOTO   LOOP1
                GOTO   LOOP
STP            GOTO   STP

*****
; adımlı motoru 1-2 fazlı sürmek için gerekli bilgilerin tablosu
*****
TABLO          ADDWF   PCL,F        ;PCL ( program counter )+ W >>PCL
                RETLW  B'00000001' ;1. adım bilgisi >> W (PCL+0)
                RETLW  B'00000011' ;2. adım bilgisi >> W (PCL+1)
                RETLW  B'00000010' ;3. adım bilgisi >> W (PCL+2)
                RETLW  B'00000110' ;4. adım bilgisi >> W (PCL+3)
                RETLW  B'00000100' ;5. adım bilgisi >> W (PCL+4)
                RETLW  B'00001100' ;6. adım bilgisi >> W (PCL+5)
                RETLW  B'00001000' ;7. adım bilgisi >> W (PCL+6)
```

```
RETLW    B'00001001'    ;8. adım bilgisi >> W (PCL+7)
```

```
TIMER                                ;yaklaşık 5ms zamanlayıcı
```

```
*****  
; * 1+1+2*(1+1+250*(2+2+2+1+1+2))+1+1+2=5017us *  
*****
```

```
        MOVLW    D'2'        ; 1clock  
        MOVWF   CNT1        ; 1clock  
DLY1    MOVLW    D'250'      ; 1clock  
        MOVWF   CNT2        ; 1clock  
DLY2    GOTO     $+1         ; 2clock  
        GOTO     $+1         ; 2clock  
        GOTO     $+1         ; 2clock  
        NOP                    ; 1clock  
        DECFSZ  CNT2,1      ; 1(2)clock  
        GOTO     DLY2        ; 2clock  
        DECFSZ  CNT1,1      ; 1(2)clock  
        GOTO     DLY1        ; 2clock  
        RETURN                ; 2clock  
        END
```

#### 2.1.4. Uygulama-4

```
LIST          P=16F84,R=DEC  
              INCLUDE "P16F84.INC"  
COUNT1      EQU    H'10'    ; MEMORY ADDRESS  
COUNT2      EQU    H'11'  
BILGI        EQU    H'12'  
SATIRNO      EQU    H'13'  
  
INIT          BSF     STATUS,RP0  
              CLRF   TRISB  
              MOVLW  B'11111'  
              MOVWF  TRISA  
              BCF    STATUS,RP0  
  
KONTROL      CLRF   PORTB  
              BTFSS  PORTA,0    ;PA0 butonu girilmezse 1 satır atla  
              GOTO   ILERI      ; PA0 on >> goto ILERI  
  
              BTFSS  PORTA,1    ; PA1 butonu girilmezse 1 satır atla  
              GOTO   GERI       ;PA1 on >> goto GERI  
              GOTO   STOP       ;hiçbir buton yok >> goto KONTROL  
  
ILERI        MOVLW  D'4'        ; adımlı motor bilgi sayısı= 4
```

```

MOVWF BILGI           ; BILGI = 4
CLRF  SATIRNO         ; SATIRNO = 0
LOOP1 MOVF  SATIRNO,W  ; SATIRNO >> W
CALL  TABLO1         ; motor ileri bilgileri sırayla alınır
MOVWF PORTB          ; W(bilgi) >> PORTB
CALL  TIMER          ; 20ms zaman gecikme mknatıslanma için
BTFSS PORTA,1        ; PA1 butonu girilmezse 1 satır atla
GOTO  GERI           ; PA1 on >> goto GERI
BTFSS PORTA,2        ; PA2 butonu girilmezse 1 satır atla
GOTO  KONTROL        ; PA2 on >> goto KONTROL
INCF  SATIRNO,F       ; SATIRNO +1 >> SATIRNO
DECFSZ BILGI,F       ; BILGI -1 >> BILGI = 0 olursa 1 satır atla
GOTO  LOOP1
GOTO  ILERI

GERI  MOVLW D'4'      ; adımlı motor bilgi sayısı= 4
MOVWF BILGI          ; BILGI = 4
LOOP2 CLRF  SATIRNO   ; BILGI = 0
MVF  SATIRNO,W      ; BILGI >> W
CALL  TABLO2         ; motor geri bilgileri sırayla alınır
MOVWF PORTB          ; W(bilgi) >> PORTB
CALL  TIMER          ; 20ms zaman gecikme mknatıslanma için
BTFSS PORTA,0        ; PA0 butonu girilmezse 1 satır atla
GOTO  ILERI         ; PA0 on >> goto ILERI
BTFSS PORTA,2        ; PA2 butonu girilmezse 1 satır atla
GOTO  KONTROL        ; PA2 on >> goto KONTROL
INCF  SATIRNO,F       ; SATIRNO +1 >> SATIRNO
DECFSZ BILGI,F       ; BILGI -1 >> BILGI = 0 olursa 1 satır atla
GOTO  LOOP2
GOTO  GERI

STP   GOTO  STP

TABLO1 ADDWF PCL,F    ;PCL(program counter) + W >>PCL
RETLW B'00000011'   ; 1. adım bilgisi >> W (PCL+0)
RETLW B'00000110'   ; 2. adım bilgisi >> W (PCL+1)
RETLW B'00001100'   ; 3. adım bilgisi >> W (PCL+2)
RETLW B'00001001'   ; 4. adım bilgisi >> W (PCL+3)

TABLO2 ADDWF PCL,F    ;PCL(program counter) + W >>PCL
RETLW B'00001001'   ; 1. adım bilgisi >> W (PCL+0)
RETLW B'00001100'   ; 2. adım bilgisi >> W (PCL+1)
RETLW B'00000110'   ; 3. adım bilgisi >> W (PCL+2)
RETLW B'00000011'   ; 4. adım bilgisi >> W (PCL+3)

TIMER ; about 5ms Timer

```

```

;*****
;* 1+1+2*(1+1+250*(2+2+2+1+1+2))+1+1+2)+1+2=5017us      *
;*****
DLY1      MOVLW D'2'          ; 1clock
          MOVWF COUNT1      ; 1clock
          MOVLW D'250'      ; 1clock
DLY2      MOVWF COUNT2      ; 1clock
          GOTO $+1          ; 2clock
          GOTO $+1          ; 2clock
          GOTO $+1          ; 2clock
          NOP              ; 1clock
          DECFSZ COUNT2,1   ; 1(2)clock
          GOTO DLY2        ; 2clock
          DECFSZ COUNT1,1   ; 1(2)clock
          GOTO DLY1        ; 2clock
          RETURN           ; 2clock

          END

```

### 2.1.5. Uygulama-5

```

LIST      P=16F84,R=DEC
          INCLUDE "P16F84.INC"
COUNT1   EQU    H'10'
COUNT2   EQU    H'11'
BILGI     EQU    H'12'
SATIRNO   EQU    H'13'
ADIM      EQU    H'14'

          BSF     STATUS,RP0
          MOVLW  B'11111'
          MOVWF  TRISA
          CLRF   TRISB
          BCF    STATUS,RP0
          CLRF   PORTB

SW        BTFSC  PORTA,0
          GOTO   SW
          MOVLW  D'200'
          MOVWF  ADIM
LOOP      MOVLW  D'4'          ; W = 4
          MOVWF  BILGI        ; BILGI = 4
          CLRF   SATIRNO      ; SATIRNO = 0
LOOP1     MOVF   SATIRNO,W    ; SATIRNO >> W
          CALL   TABLO        ; SATIRNO 0'dan 3'e
          MOVWF  PORTB        ; W(motor bilgisi) >> PORTB

```

```

CALL    TIMER           ; 20ms zaman
DECFSZ  ADIM,F
GOTO    DEVAM
GOTO    STOP

DEVAM   INCF    SATIRNO,F    ; SATIRNO +1 >> SATIRNO
DECFSZ  BILGI,F ; BILGI -1 >> BILGI = 0 değilse 1 satır atla
GOTO    LOOP1
GOTO    LOOP

STOP    CLRF    PORTB
GOTO    STOP

TABLO   ADDWF   PCL,F        ;PCL(program counter) + W >>PCL
RETLW   B'0000011'        ;1. motor bilgisi >> W (PCL+0)
RETLW   B'0000110'        ; 2. motor bilgisi >> W (PCL+1)
RETLW   B'00001100'       ; 3. motor bilgisi >> W (PCL+2)
RETLW   B'00001001'       ; 4. motor bilgisi >> W (PCL+3)

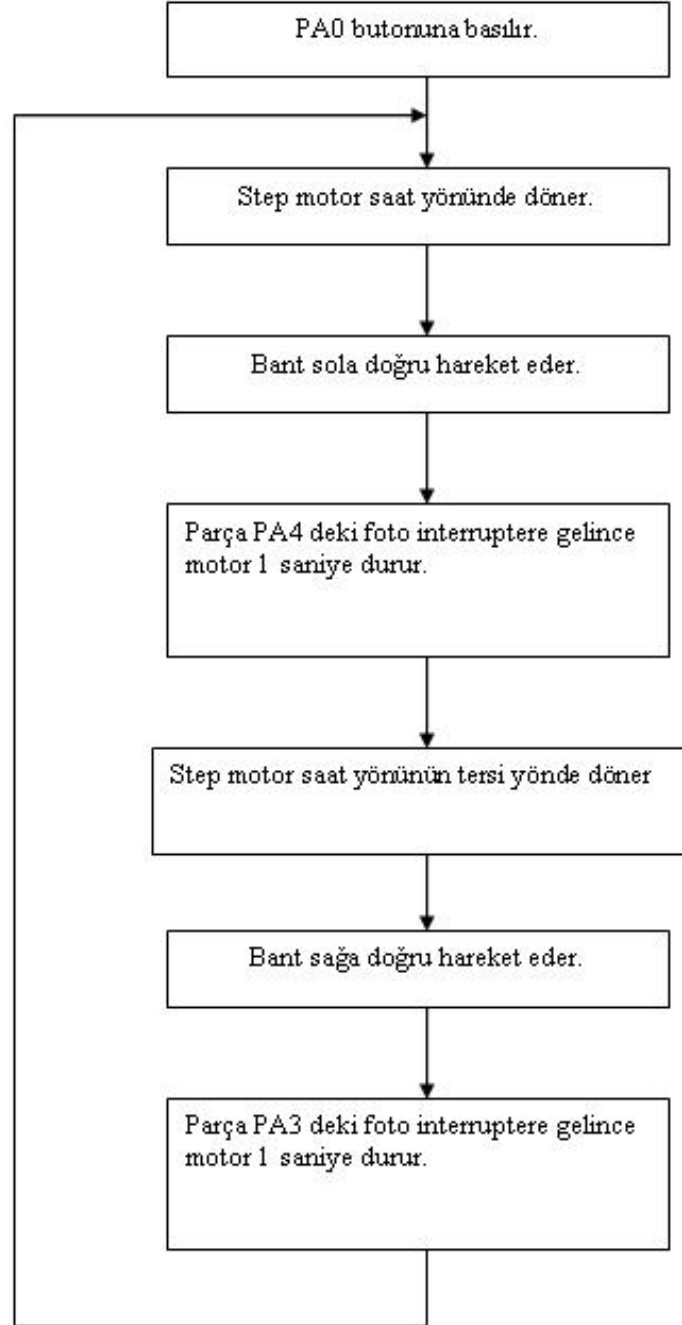
TIMER           ; about 5ms Timer
;*****
;* 1+1+2*(1+1+250*(2+2+2+1+1+2)+1+1+2)+1+2=5017us      *
;*****
DLY1      MOVLW  D'2'        ; 1clock
          MOVWF  COUNT1      ; 1clock
          MOVLW  D'250'      ; 1clock
          MOVWF  COUNT2      ; 1clock
DLY2      GOTO   $+1         ; 2clock
          GOTO   $+1         ; 2clock
          GOTO   $+1         ; 2clock
          NOP           ; 1clock
          DECFSZ COUNT2,1    ; 1(2)clock
          GOTO   DLY2       ; 2clock
          DECFSZ COUNT1,1    ; 1(2)clock
          GOTO   DLY1       ; 2clock
          RETURN          ; 2clock

          END

```

## 2.2. Uygulama Faaliyeti -2 Cevap Anahtarı

### 2.2.1. Uygulama -6



```

LIST          P=16F84,R=DEC
INCLUDE      "P16F84.INC"
COUNT1     EQU      H'10'
COUNT2     EQU      H'11'
BILGI       EQU      H'12'
SATIRNO     EQU      H'13'
TCOUNT      EQU      H'14'
BSF         STATUS,RP0
CLRF        TRISB
MOVLW      B'11111111'
MOVWF      TRISA
BCF         STATUS,RP0
BEKLE      CLRF        PORTB          ;PORTB=0
           BTFSS      PORTA,0        ;Eğer PA0 girilmezse 1 satır atla
           GOTO      FORWARD        ;PA0 on >> goto ILERI
           GOTO      BEKLE          ;no SW >> goto BEKLE
ILERI      CALL      ZAMAN          ; 1s bekleme süresi
FLOOP     MOVLW      D'4'           ; adımlı motor bilgi sayısı= 4
           MOVWF     BILGI          ; BILGI= 4
           CLRF     SATIRNO         ; SATIRNO = 0
LOOP1     MOVF      SATIRNO,W       ; SATIRNO >> W
           CALL     TABLO1          ; SATIRNO 0'dan 3'e
           MOVWF     PORTB          ; W(bilgi) >> PORTB
           CALL     TIMER           ; 20ms zaman
           BTFSS   PORTA,4         ; PA4, 1 ise 1 satır atla
           GOTO     GERI            ; PA4 on(0) >> goto GERI
           BTFSS   PORTA,1         ; PA1 girilmezse 1 satır atla
           GOTO     BEKLE          ; PA1 on >> goto BEKLE
           INCF    SATIRNO ,F       ; SATIRNO +1 >> SATIRNO
           DECFSZ  BILGI ,F        ; BILGI-1 >> BILGI = 0 ise 1 satır atla
           GOTO     LOOP1
           GOTO     FLOOP
GERI      CALL      ZAMAN          ; 1s bekleme süresi
RLOOP    MOVLW      D'4'           ; adımlı motor bilgi sayısı= 4
           MOVWF     BILGI          ; BILGI = 4
           CLRF     SATIRNO         ; SATIRNO = 0
LOOP2    MOVF      SATIRNO,W       ; SATIRNO >> W
           CALL     TABLO2          ; SATIRNO 0'dan 3'e
           MOVWF     PORTB          ; W(bilgi) >> PORTB
           CALL     TIMER           ; 20ms zaman
           BTFSS   PORTA,3         ; PA3, 1 ise 1 satır atla
           GOTO     ILERI          ; PA3 on(0) >> goto ILERI
           BTFSS   PORTA,1         ; PA1, 1 ise 1 satır atla
           GOTO     BEKLE          ; PA1 on >> goto BEKLE
           INCF    SATIRNO ,F       ; SATIRNO +1 >> SATIRNO
           DECFSZ  BILGI ,F        ; BILGI -1 >> BILGI = 0 then skip

```



```

                GOTO      LOOP2
                GOTO      RLOOP
STP             GOTO      STP
ZAMAN          MOVLW     B'00000000'
                MOVF      PORTB
                MOVLW     D'25'
                MOVWF     TCOUNT
TLOOP          CALL      TIMER
                DECFSZ    TCOUNT,F
                GOTO      TLOOP
                RETURN
TABLO1         ADDWF     PCL,F      ;PCL(program counter) + W >>PCL
                RETLW     B'00000011' ;1. motor bilgisi >> W (PCL+0)
                RETLW     B'00000110' ;2. motor bilgisi >> W (PCL+1)
                RETLW     B'00001100' ;3. motor bilgisi >> W (PCL+2)
                RETLW     B'00001001' ;4. motor bilgisi >> W (PCL+3)
TABLO2         ADDWF     PCL,F      ;PCL(program counter) + W >>PCL
                RETLW     B'00001001' ;1. motor bilgisi >> W (PCL+0)
                RETLW     B'00001100' ;2. motor bilgisi >> W (PCL+1)
                RETLW     B'00000110' ;3. motor bilgisi >> W (PCL+2)
                RETLW     B'00000011' ;4. motor bilgisi >> W (PCL+3)
TIMER
;*****
;* 1+1+20*(1+1+200*(2+2+2+1+1+2))+1+1+2)+1+2=40125us      *
;*****
                MOVLW     D'20'      ; 1clock
                MOVWF     COUNT1     ; 1clock
DLY1           MOVLW     D'40'      ; 1clock
                MOVWF     COUNT2     ; 1clock
DLY2           GOTO      $+1         ; 2clock
                GOTO      $+1         ; 2clock
                GOTO      $+1         ; 2clock
                NOP          ; 1clock
                DECFSZ    COUNT2,1   ; 1(2)clock
                GOTO      DLY2       ; 2clock
                DECFSZ    COUNT1,1   ; 1(2)clock
                GOTO      DLY1       ; 2clock
                RETURN          ; 2clock

                END

```

#### 2.4. Ölçme Soruları Cevap Anahtarı

1-d 2-b 3-c 4-c 5-b

## KAYNAKÇA

- Ø İŞBİLEN Turgay, Mustafa GÜNEŞ, Selim GÜLÇEN, “**Mikrodenetleyici Uygulamaları**”, EOT-JICA, İzmir, Ekim 2003.
- Ø [http://www.darkhardware.com/st.php?u=articles/at\\_atx\\_tetikleme-01](http://www.darkhardware.com/st.php?u=articles/at_atx_tetikleme-01)