



Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü

EE-302

Mikroişlemciler

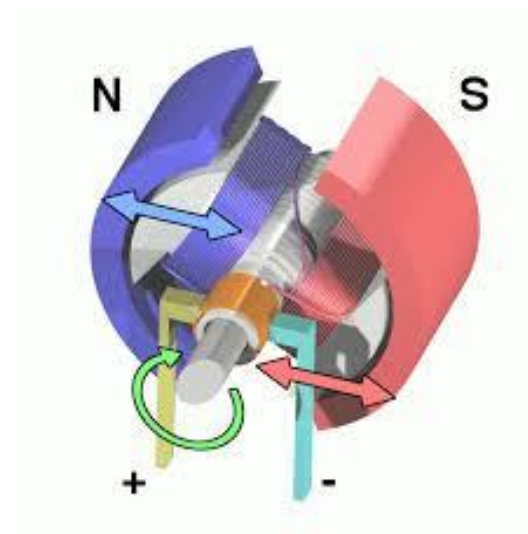
Motor Kontrol Uygulamaları

11. Hafta

Prof. Dr. Mehmet DEMİRTAŞ

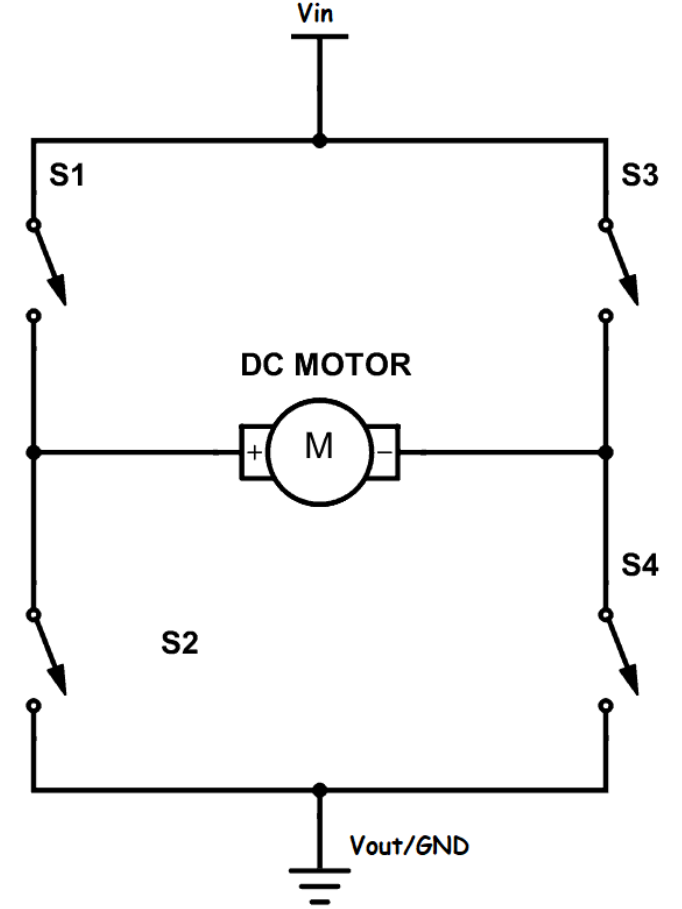
Motor Kontrol Uygulamaları

- Bu bölümde DC motor, step (adım) motor ve R/C tipi servo motorlar hakkında bilgi verilmiş ve ardından her tip motor için birer uygulama gerçekleştirilmiştir.
- **DC MOTOR**
- DC motorlar endüstride uzun zamandan beri çok kullanılan motor tipidir. DC motorlar manyetik alan içine içinden akım geçen telin konulmasıyla meydana gelen kuvveti kullanma prensibiyle dönmeyi sağlarlar. Diğer elektrik motor türleri de bu temel prensibi değişik donanım özellikleriyle kullanır. DC motorlar, sabit bir mıknatıs ve içinde bir rotor içerirler.



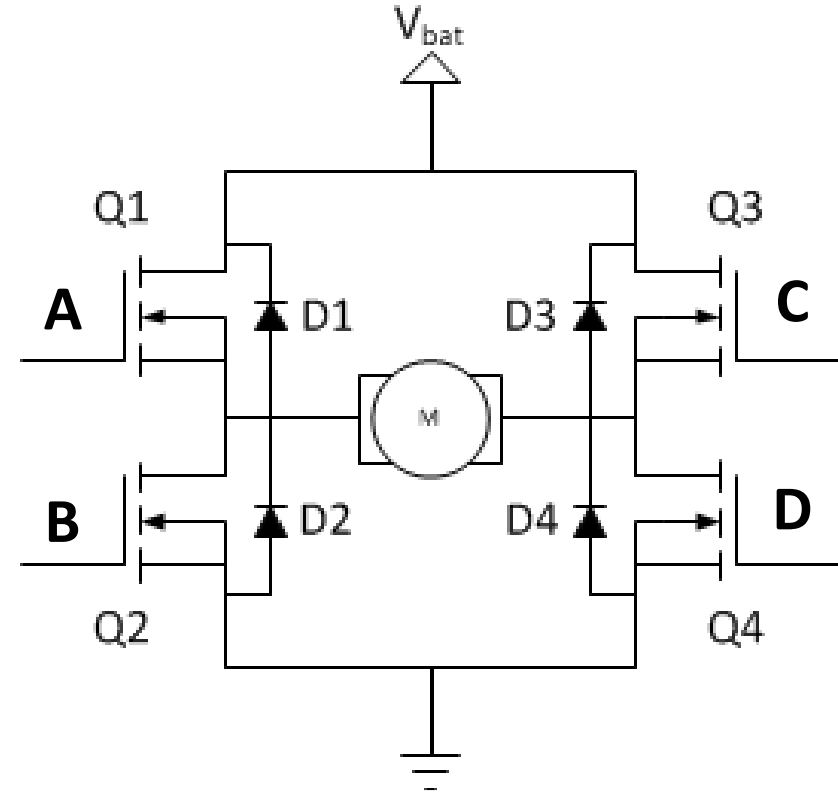
DC Motor

DC motorları direkt olarak mikro denetleyici çıkışı ile süremeyiz. Bu nedenle sürücü devreler kullanılır. DC motorların sürücü ve yön kontrolü için H-köprü devreleri kullanılır. Köprü devreleri genelde transistör ile yapılır. Fakat hem yön hem de hız kontrolü için köprü devrelerinde transistör yerine MOSFET kullanılabilir. Şekil 'de verilen MOSFET'li, bir H-Köprü devresi ile kontrol uçlarına Tablo'da verilen lojik değerler uygulandığında istenen ileri, geri, dur, serbest konumları elde edilir. Yine A ve D veya B ve C MOSFET çiftine uygulanan **gate sinyali** PWM mantığına göre kontrol edilirse motorun hızı da kontrol edilmiş olur.



DC Motor

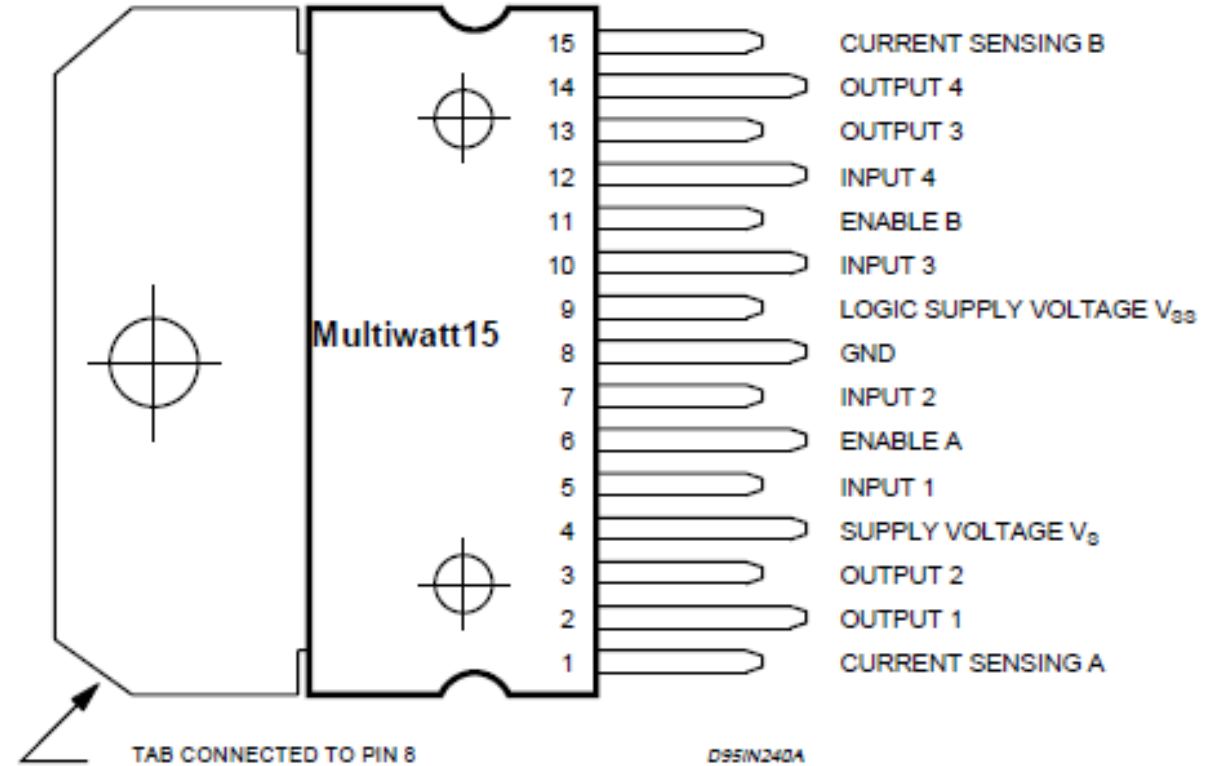
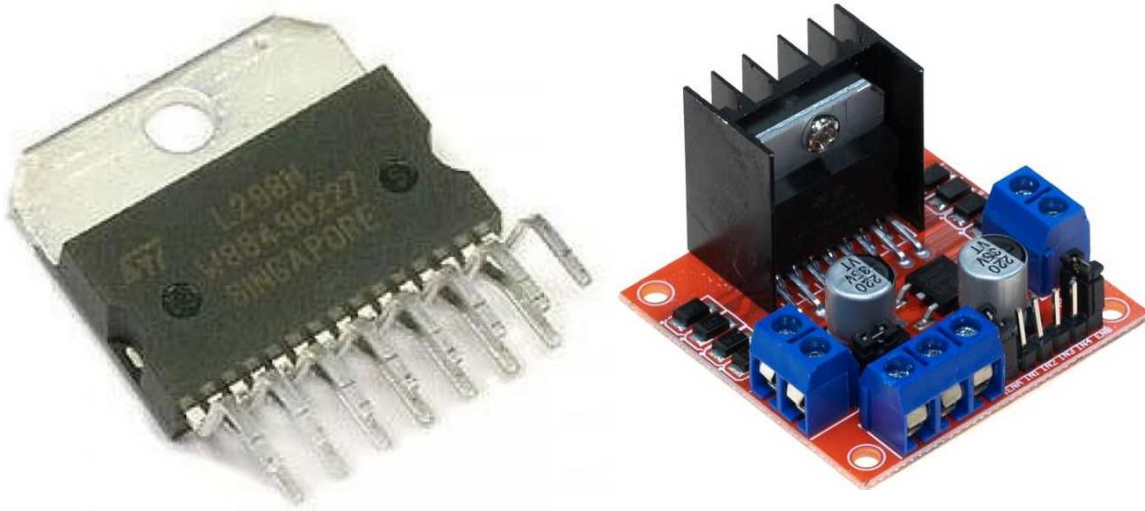
Şekil'deki diyotlar, DC motor tarafından oluşabilecek ters EMK'nin (Elektro Motor Kuvveti) devre elemanlarına zarar vermesini önlemek için kullanılmıştır. Bu iş için BA159 hızlı diyotu gibi diyot modelleri tercih edilir. Tüm MOSFET'ler TTL/CMOS çıkışları ile sürülemez. Sadece bazı MOSFET'ler TTL/CMOS uyumludur. Bu nedenle ya TTL/CMOS uyumlu MOSFET kullanmak gerekli ya da MOSFET sürücü entegreleri kullanmak gereklidir. MOSFET sürücü entegreler olarak MAX620/621, ICL7667, MAX8552 gibi sürücü entegreler kullanılabilir. Fakat çoğu DC motor kontrol uygulamasında Şekil'de verilen H-köprü sürücü devrelerini içinde barındıran entegre elemanlar kullanılır. L293, L298, L6201, L6202, L6203 veya LMD18200 gibi entegreler DC motor sürücü için kullanılabilir. Bu bölümde DC motor kontrolü için L298 entegresini inceleyeceğiz.



A	B	C	D	Motor Durumu
1	0	0	1	Saat yönünde döner
0	1	1	0	Saat yönünün tersine döner
1	0	1	0	Motor durur (OFF)
0	0	0	0	Motor boşta

L298 DC Motor Sürücü Entegresi

L298 DC motor sürücü entegresi, içinde iki adet transistörlü sürücü devre barındırır. Sürücü devrelerin ilki A, ikincisi B olarak isimlendirilmiştir. Böylece aynı entegre ile 2 adet motor kontrol edilebilir. Çalışma gerilimi 46V'a s. Çıkış akımı ise 4 Amper'e kadar çıkmaktadır. Entegre aynı zamanda aşırı ısı korumasına sahiptir.



L298 DC Motor Sürücü Entegresi

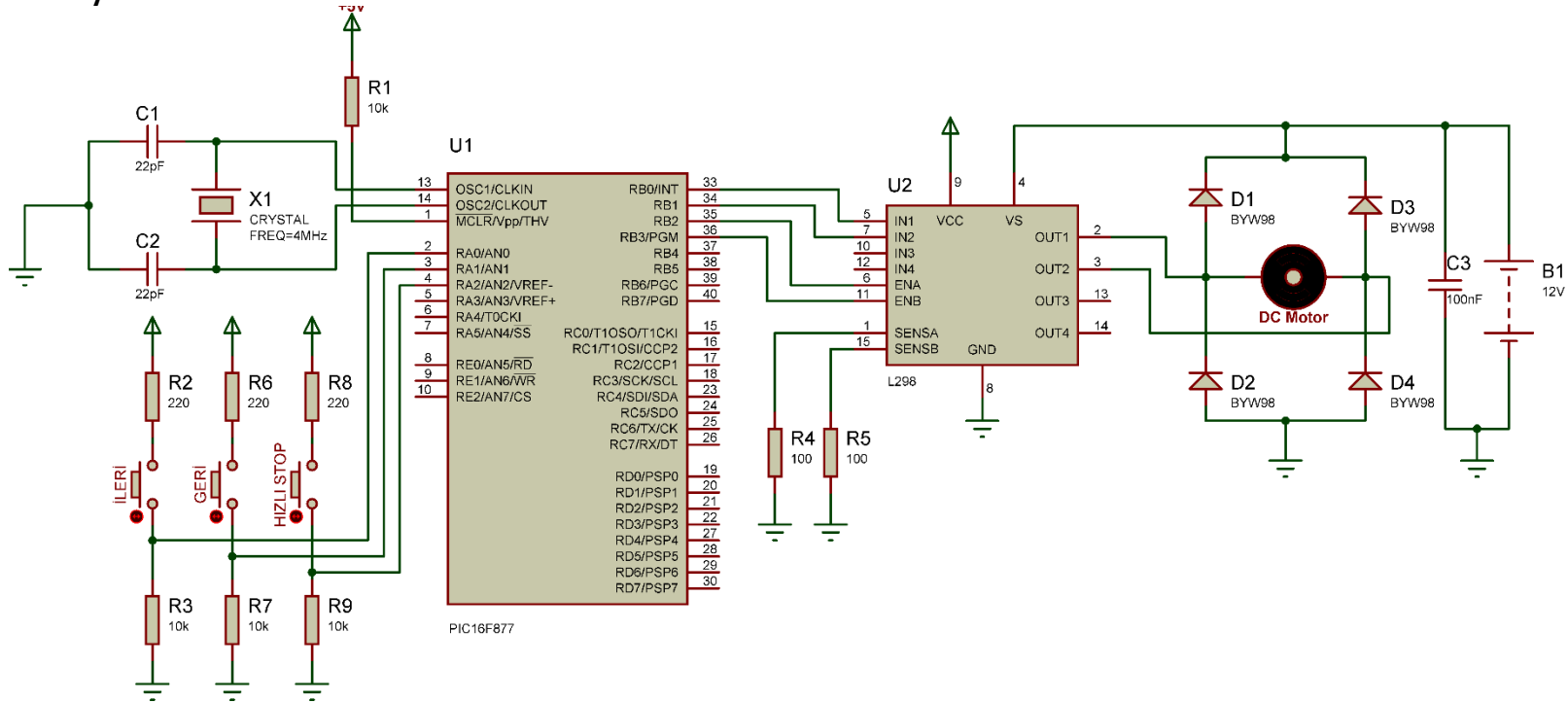
Bu entegrenin pin isimleri soldaki tabloda verilmektedir. L298 entegresinden Input girişlerine verilen lojik değerler (0,1) ile motorun ileri, geri, stop veya boşta olma durumları belirlenir. Bu durumlar sağdaki tablo'da verilmiştir.

Pin No	Pin İsmi	Açıklama
1,15	Sense A, Sense B	Bu pin'ler şaseye bir direnç ile bağlanır. Motordan geçen akımı kontrol etmek için kullanılır.
2,3	Out1, Out2	A köprü devresinin çıkış uçları.
4	Vs	Motor beslemesinin + ucu. Bu uç 100nF'lık kondansatör ile şase ucuna bağlanmalıdır.
5,7	Input1, Input2	A köprü devresinin TTL kontrol uçları.
6,11	Enable A, Enable B	Entegre içindeki köprü devrelerini (A ve B) aktif etme pin'leri.
8	GND	Şase ucu.
9	VSS	L298 entegresi besleme gerilim ucu. Bu uç 100nF'lık kondansatör ile şase ucuna bağlanmalıdır.
10,12	Input3, Input4	B köprü devresinin TTL kontrol uçları.
13,14	Out3, Out4	B köprü devresinin çıkış uçları.

Köprü Devresi	Girişler	Durum	Köprü Devresi	Girişler	Durum
A Köprü Devresi	Enable A=High Enable B=Low Input1=High Input2=Low	İLERİ	B Köprü Devresi	Enable A=High Enable B=Low Input1=High Input2=Low	İLERİ
	Enable A=High Enable B=Low Input1=Low Input2=High	GERİ		Enable A=High Enable B=Low Input1=Low Input2=High	GERİ
	Enable A=High Enable B=Low Input1=Input2	HIZLI STOP		Enable A=High Enable B=Low Input1=Input2	HIZLI STOP
	Enable A=Low Enable B=Low Input1=X Input2=X	MOTOR BOŞTA		Enable A=Low Enable B=Low Input1=X Input2=X	MOTOR BOŞTA

DC motor Uygulaması

Bu uygulamada 12V'luk bir DC motorun yön kontrolü L298 entegresi kullanılarak gerçekleştirilmiştir. Uygulamada L298 entegresinin Köprü-A kısmı kullanılmıştır. A port'una bağlanan 3 buton ileri, geri ve hızlı durdurma işlemleri içindir. L298'in 1 numaralı ucuna bağlanan direnç değeri küçültülerek motordan geçen akım arttırılabilir direnç değeri arttırılarak motordan geçen akım azaltılabilir. Devrede kullanılan diyotlar hızlı diyotlardır. Kendi uygulamanızda kullandığınız motor voltajı ve motorun çekeceği akıma göre diyotlar kullanmayı unutmayınız. Kullandığınız motorun + besleme voltajı için Vs ucuna uygun gerilim vermeyi unutmayınız.



Örnek Programlar (DC Motor Uygulaması)

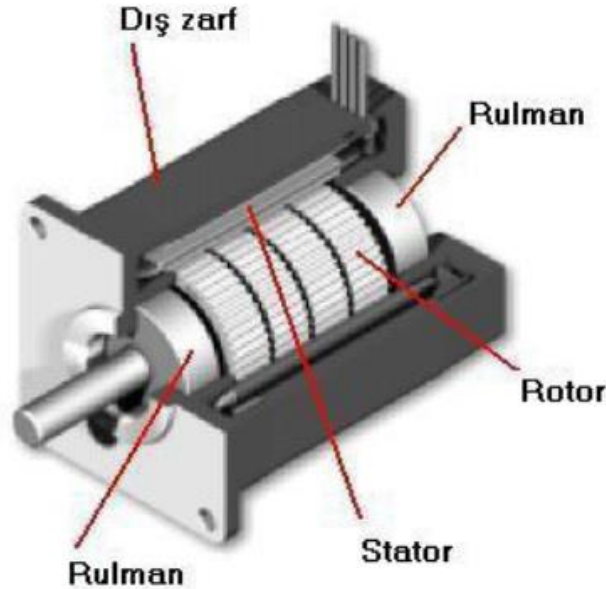
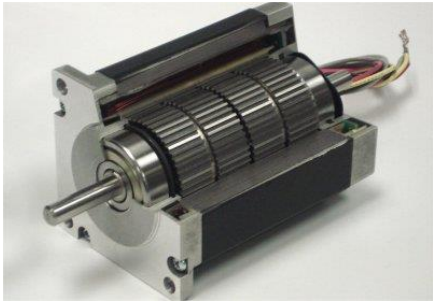
```
1  #include <16f877.h>      // Kullanılacak denetleyicinin başlık dosyası
2  #fuses XT,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPPD
3  #use delay (clock=4000000) // Gecikme fonksiyonu için kullanılacak osi
4
5  // Giriş ve çıkış pinlerine isim atanıyor
6  #define buton_ileri pin_a0
7  #define buton_geri  pin_a1
8  #define buton_stop  pin_a2
9  #define input1      pin_b0
10 #define input2      pin_b1
11 #define enable_a    pin_b2
12 #define enable_b    pin_b3
13 /***** ANA PROGRAM FONKSİYONU*****/
14 void main ( )
15 {
16     setup_psp(PSP_DISABLED);      // PSP birimi devre dışı
17     setup_timer_1(T1_DISABLED);   // T1 zamanlayıcısı devre dışı
18     setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
19     setup_adc_ports(NO_ANALOGS);   // ANALOG giriş yok
20     setup_adc(ADC_OFF);           // ADC birimi devre dışı
21     setup_CCP1(CCP_OFF);          // CCP1 birimi devre dışı
22     setup_CCP2(CCP_OFF);          // CCP2 birimi devre dışı
23 }
```

Örnek Programlar (DC Motor Uygulaması)

```
24 output_high(enable_a); // Köprü A seçili
25 output_low(enable_b); // Köprü B pasif
26 while(1) // Sonsuz döngü
27 {
28     if (input(buton_ileri)) // İLERİ Butonuna basılsı ise
29     {
30         output_high(input1); // L298 Input1 girişi lojik-1
31         output_low(input2); // L298 Input2 girişi lojik-0
32     }
33     if (input(buton_geri)) // GERİ Butonuna basılsı ise
34     {
35         output_low(input1); // L298 Input1 girişi lojik-0
36         output_high(input2); // L298 Input1 girişi lojik-1
37     }
38     if (input(buton_stop)) // HIZLI STOP Butonuna basılsı ise
39     {
40         output_low(input1); // L298 Input1 girişi lojik-0
41         output_low(input2); // L298 Input1 girişi lojik-0
42         // Hızlı stop için burada her iki girişte lojik-1 olabilirdi.
43         // Önemli olan Hızlı Stop için, her iki girişin aynı seviyede olması
44     }
45 }
46 }
```

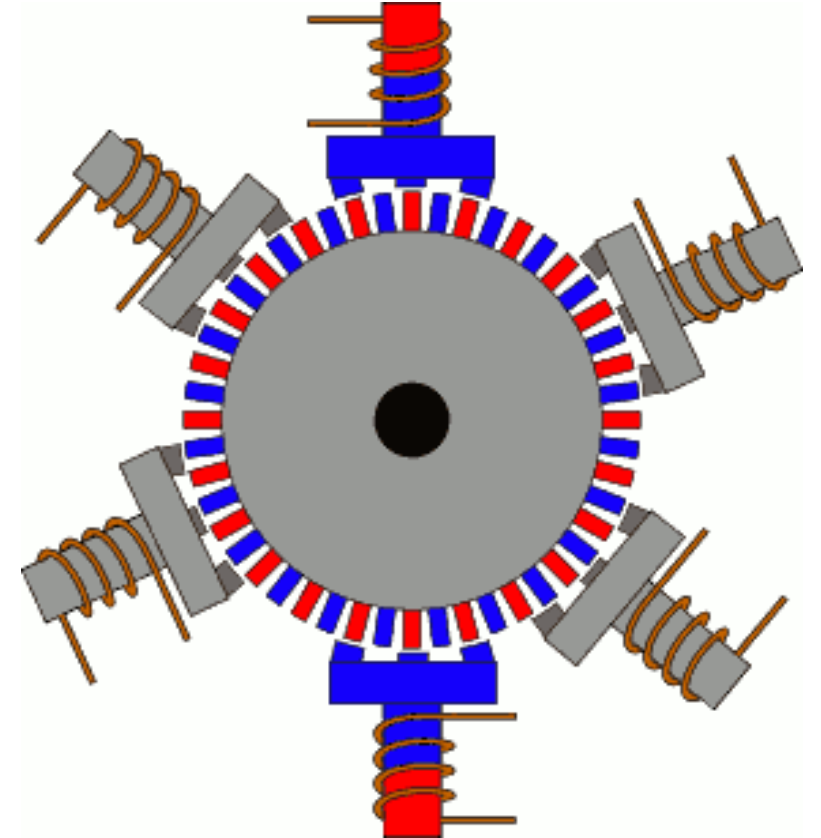
Step (Adım) Motor

Step (adım) motorlar çok yönlü, kalıcı ömürlü birçok uygulamada kullanılan motorlardır. Step motorlarda diğer motor türleri gibi elektrik enerjisini, mekanik enerjiye çevirirler. Diğer türlerden farkı, dijital kontrollü olması ve her gönderilen pals için belli açıda dönmesidir. Bu özelliğinden dolayı bilgisayar destekli uygulamalarda sıkça kullanılmaktadır. Temelde, sabit mıknatis içeren rotor (hareketli kısım) ve rotorun hareketini sağlayan statordan (sabit kısım) oluşmaktadır.



Step (Adım) Motor

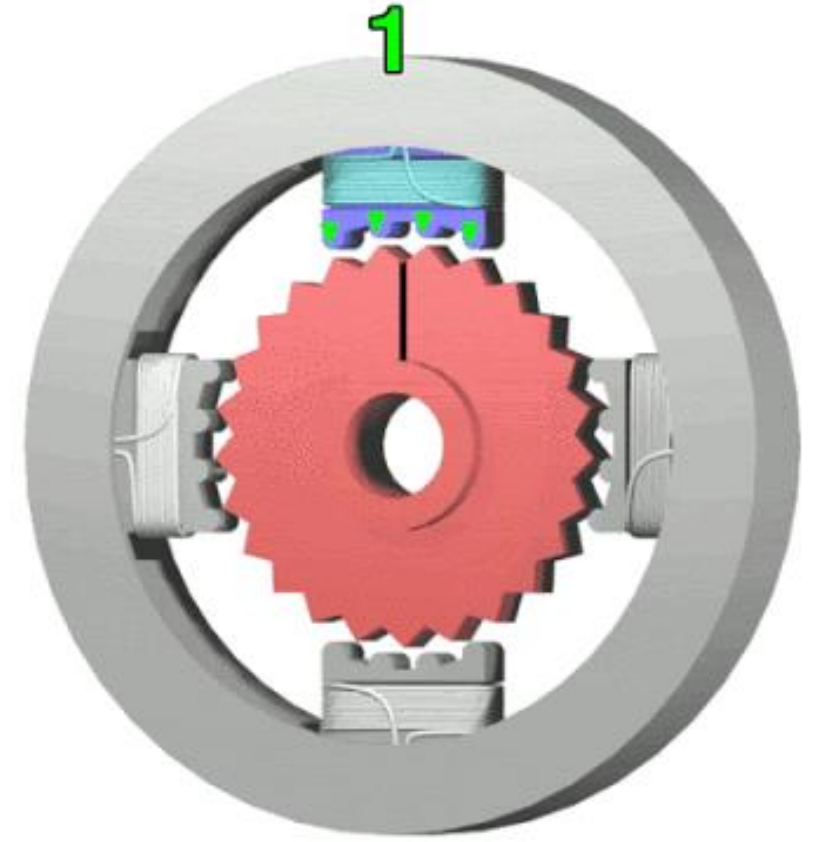
Step motorlar sabit mıknatıslı ve karma mıknatıslı yapılara sahiptir. Sabit mıknatıslı step motorlar mıknatısın manyetik alan içinde hareket etmesi prensibine göre çalışır. Sargılara uygulanan sinyallere göre rotor hareket eder. Şekil'de görülen stator sargılarından (bobin) gerçek step motorlarında çok miktarda bulunmaktadır. Sargılara uygulanan sinyallerden dolayı sargılar içinden geçen akımın yönüne göre sargılarda N veya S kutbu oluşur. Rotor sargısı da bu duruma göre döner. Step motorlar genellikle kare dalga palsler ile enerjilendirilir. Rotorda oluşan tork, bobinlere (sargılar) uygulanan akımla doğru orantılıdır. Uygulanan akım ne kadar fazla ise, oluşan manyetik alan o kadar fazla olur, dolayısıyla tork da artar. Bobin sargıları doyuma ulaşınca akım ve gerilim arttırılsa dahi tork artmaz.



Step (Adım) Motor

Karma step motorlarda (Hybrid stepper motors) rotor, iki tane aynı doğrultuda olan silindirin paslanmaz çelik şafta (mil) monte edilmesi ile oluşur. Böylece rotorun bir ucu kuzey (N) diğer ucu güney polaritesini (S) belirtir. Rotor silindiri dişlere sahiptir. Bu tip motorlarda rotor diş sayısı ile sargı kutup sayısı birbirinden farklı olabilir. Bu motora daha küçük açılarda hareket yapma imkânı tanır.

Adım motorlar genelde 4, 5 ve 6 uçlu olarak piyasada bulunmaktadır. En fazla olarak 5 uçlu adım motorlar kullanılmaktadır. 5 uçlu (kabloya) sahip adım motorlarda bir uç ortak uçtur, 6 uçlu adım motorlarda ise 2 ortak uç vardır. Ortak uçlar ya + beslemeye, ya da toprak hattına bağlanır. Sonuçta kalan 4 uç bobin-sargı (stator) uçlarıdır. Bobin uçlarına + besleme veya toprak seviyesi uygulanır. Bobin uçlarını bulmak için sargı uçlarının direnci ölçülmelidir. Daha sonra da deneme yöntemi ile doğru sargı uçları bulunabilir.



Step (Adım) Motor

Motora bir pals uygulandığında, rotorun hareket edebileceği açı miktarına adım açısı (step angle) denir. Piyasadan step (adım) motor alırken adım açısına göre alınır. Örneğin $1,8^\circ$ adım açısına sahip adım motor 360° için 200 adım hareket ettirilir. Adım açısı ne kadar küçük olursa motor o kadar hassas kullanılabilir. Step motora sinyal uygulandığında belli bir zaman sonra tepki verir. Bu zaman miktarına adım cevabı (adım response) denir. Motorun 1 saniyede yapabileceği maksimum adım sayısına ise adımlama oranı (stepping rate) denir.

Step motoru sürmek için her bobine sırasıyla enerji verilerek dönme hareketi sağlanabilir (**Tek faz uyartım yöntemi**). Fakat bu yöntemden ziyade genelde **İki faz uyartım yöntemi** kullanılır. İki faz uyartım yöntemi için 2 farklı sürme biçimi vardır. Bunlar "**Tam Adım**" ve "**Yarım Adım**" sürme teknikleridir. **Tam adım** yönteminde motor sargılarının ikisi aynı anda enerjilendirilir. Bu sayede tek faz uyartım yöntemine göre yaklaşık 1,5 kat daha fazla tork elde edilir. Fakat çekilen akımda yaklaşık 2 katına çıkmaktadır. **Yarım adım** yönteminde ise tek faz ve tam adım sürme adımları ard arda uygulanır ve böylece rotor her enerjilendirmede yarım adım döner. Bu sayede 30'lik dönme açısına sahip bir step motoru, yarım adım sürme yöntemiyle $1,8$ derecelik dönme açısına sahip olarak döndürebiliriz. Yarım adım sürmede motor daha hassas açılarla hareket ettirilir.

Step (Adım) Motor

Adım	Uç-1	Uç-2	Uç-3	Uç-4
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

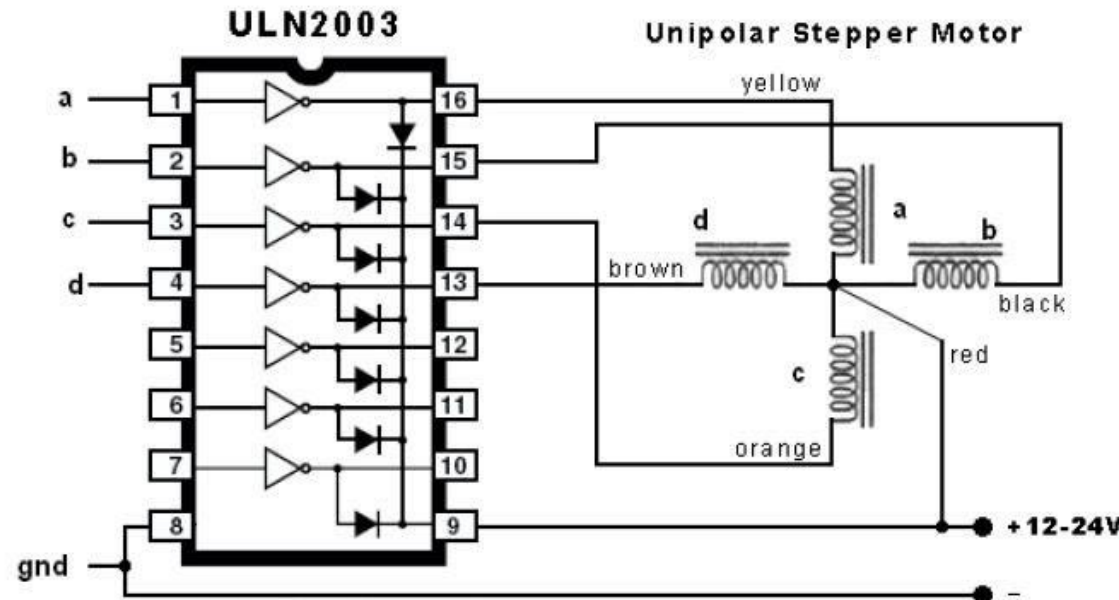
Tablo-20.4. Tam adım sürme.

Adım	Uç-1	Uç-2	Uç-3	Uç-4
1	1	0	0	0
2	1	1	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

Tablo-20.5. Yarım adım sürme.

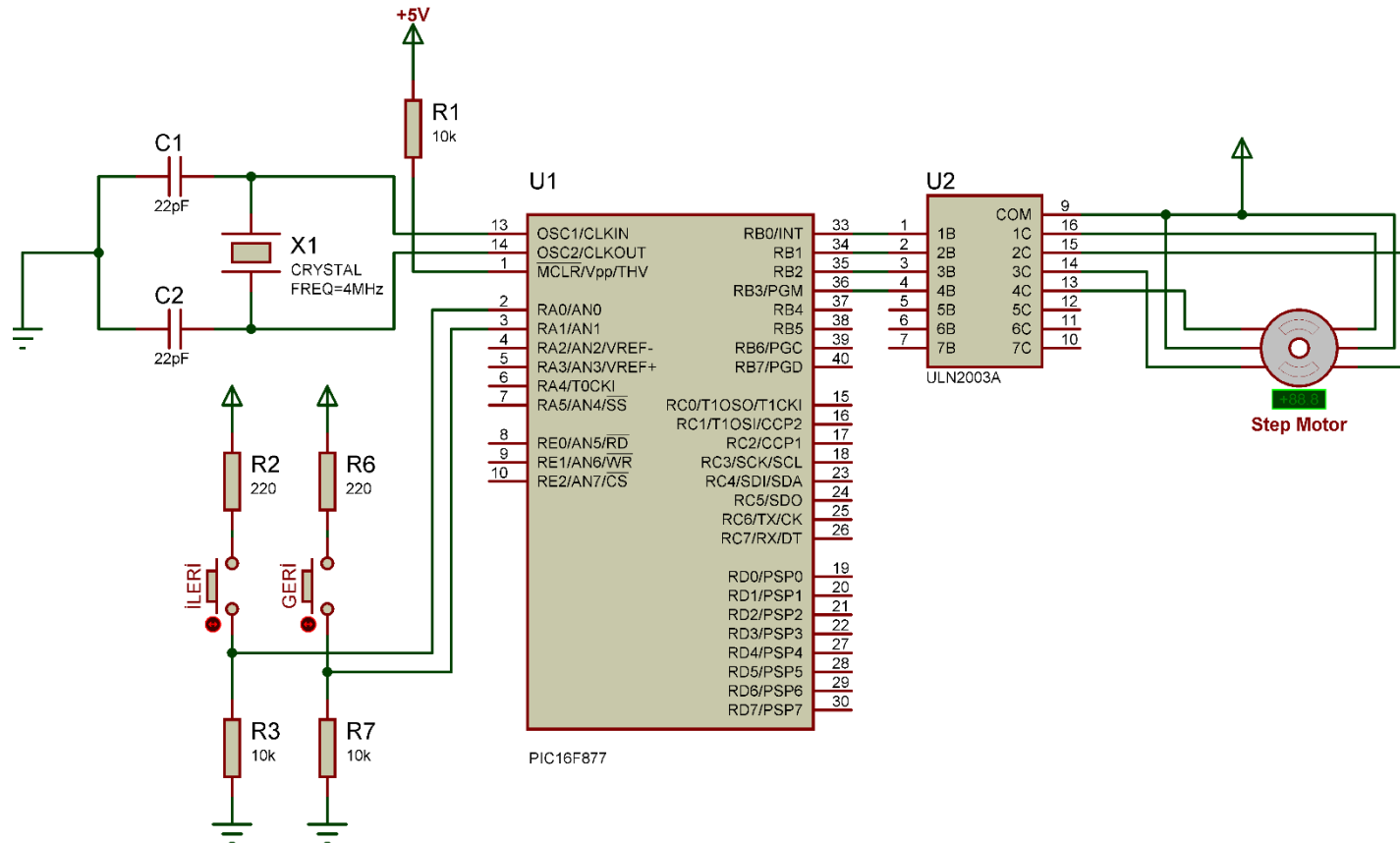
Step (Adım) Motor

Adım motor sürme işlemlerinde PIC denetleyici tarafından lojik-1 veya lojik-0 değerleri bobinlere uygulanır. Fakat PIC denetleyiciler en fazla 25 mA akım verebilirler. Bu akım değeri adım motoru sürmek için yeterli değildir. Bu sebeple denetleyici ucundan gelen sinyal akım yükseltme işlemine tabi tutulmalıdır. Bu işlem için transistörlerden faydalanabilir. Fakat bunun yerine tümleşik entegrelerde kullanılabilir. ULN 2003 entegresi step motor sürücü entegresi olarak yaygın şekilde kullanılmaktadır. Yüksek akım (kanal başına 500mA) ve yüksek gerilim (max. 50V) verebilen darlington transistör dizisi barındıran entegredir.



Step motor Uygulaması ULN-2003

Bu uygulamada 5V'luk bir step motor, ULN2003A entegresi ile sürülmüştür. Devrede RA0 ve RA1 girişlerine ILERI ve GERI butonları bağlanmıştır. Butonlara bir kez basıldığında step motor sadece 1 adım ileri veya geri gider. Program step motorun yarım adım metoduna göre adım atacak şekilde yazılmıştır.



Örnek Programlar (Step Motor Uygulaması)

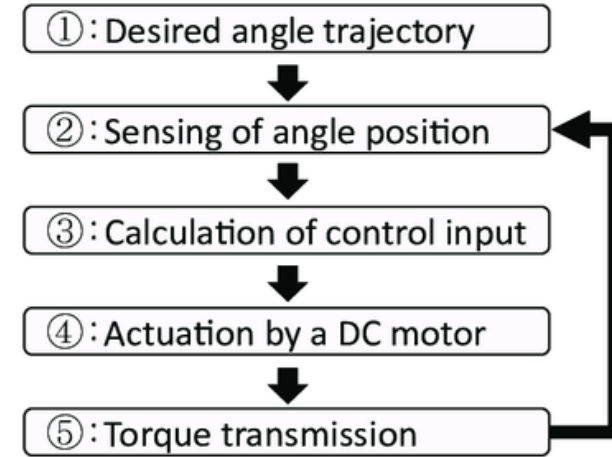
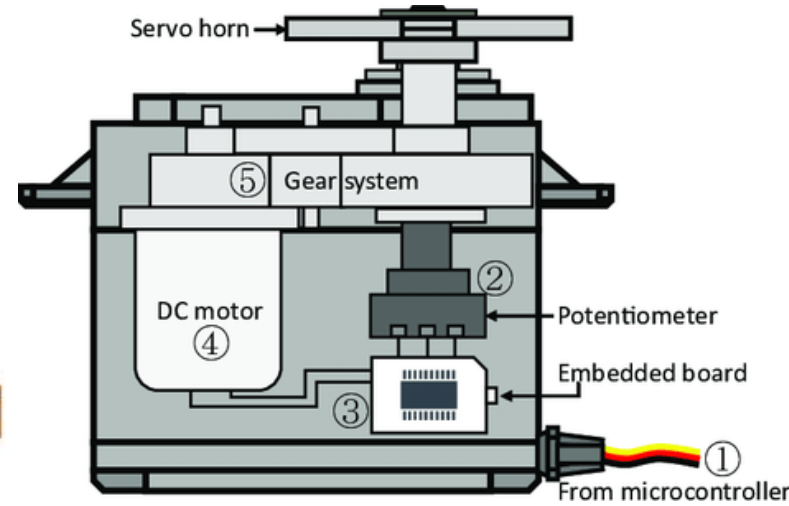
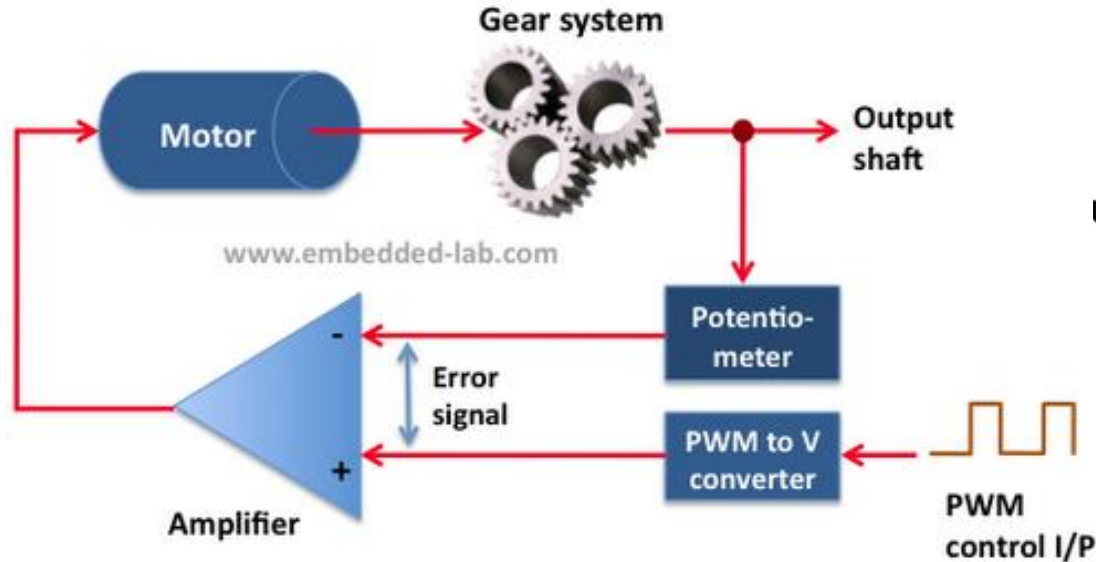
```
1  #include <16f877.h>
2  #fuses XT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPU
3  #use delay (clock=4000000)
4  // Giriş ve çıkış pinlerine isim atanıyor
5  #define buton_ileri pin_a0
6  #define buton_geri pin_a1
7
8  int i=0,hiz=1;
9  const int yarim_adim[]={0x01,0x03,0x02,0x06,0x04,0x0C,0x08,0x09}; // 1
10 /***** ANA PROGRAM FONKSİYONU*****/
11 void main ( )
12 {
13     setup_psp(PSP_DISABLED);           // PSP birimi devre dışı
14     setup_timer_1(T1_DISABLED);        // T1 zamanlayıcısı devre dışı
15     setup_timer_2(T2_DISABLED,0,1);    // T2 zamanlayıcısı devre dışı
16     setup_adc_ports(NO_ANALOGS);       // ANALOG giriş yok
17     setup_adc(ADC_OFF);                // ADC birimi devre dışı
18     setup_CCP1(CCP_OFF);               // CCP1 birimi devre dışı
19     setup_CCP2(CCP_OFF);               // CCP2 birimi devre dışı
20
21     output_b(0x00);
22
23     while(1) // Sonsuz döngü
```

Örnek Programlar (Step Motor Uygulaması)

```
23 while(1) // Sonsuz döngü
24 {
25     if (input(buton_ileri)) // İLERİ Butonuna basıldı ise
26     { output_b(yarim_adim[i]); // Step motor 1 adım ileri
27       while(input(buton_ileri)); // Basılan tuş bırakılana kadar bekle
28       delay_ms(hiz); // Adımlar arası bekleme süresi
29
30       if (i==7) // i değişkeni 7 olunca i değeri -1 olsun
31           i=-1;
32       i++; // i değişkenini 1 arttır
33     }
34
35     if (input(buton_geri)) // GERİ Butonuna basıldı ise
36     {
37         if (i==0) // i değişkeni 0 ise i değeri 8 olsun
38             i=8;
39         i--; // i değişkenini 1 azalt
40         output_b(yarim_adim[i]); // Step motora 1 adım geri
41         while(input(buton_geri)); // Basılan tuş bırakılana kadar bekle
42         delay_ms(hiz); // Adımlar arası bekleme süresi
43     }
44 }
45 }
```

Servo Motorlar

R/C Servo Motor, DC akımla çalışan ve istenilen açı aralığında dönen motor yapısıdır. R/C , Radio Controlled anlamına gelir. Servo motorlar DC Motorların temel mantığını kullanırlar. Fakat buna karşın elektronik pozisyon kontrol devresi ve elektronik şaft gibi ekstra bileşenleri vardır. Servo motor şaftın kaç derece ve hangi hızda döndüğünü algılar ve girişe bunu geri besleme olarak verir. Motorun pozisyonunu algılamak için rotora takılı bir potansiyometre bulunur. Bu potansiyometreden gelen analog değer ile inputtaki sinyal karşılaştırılır ve output olarak motorun yeni pozisyonu kontrol edilir.



Servo Motorlar

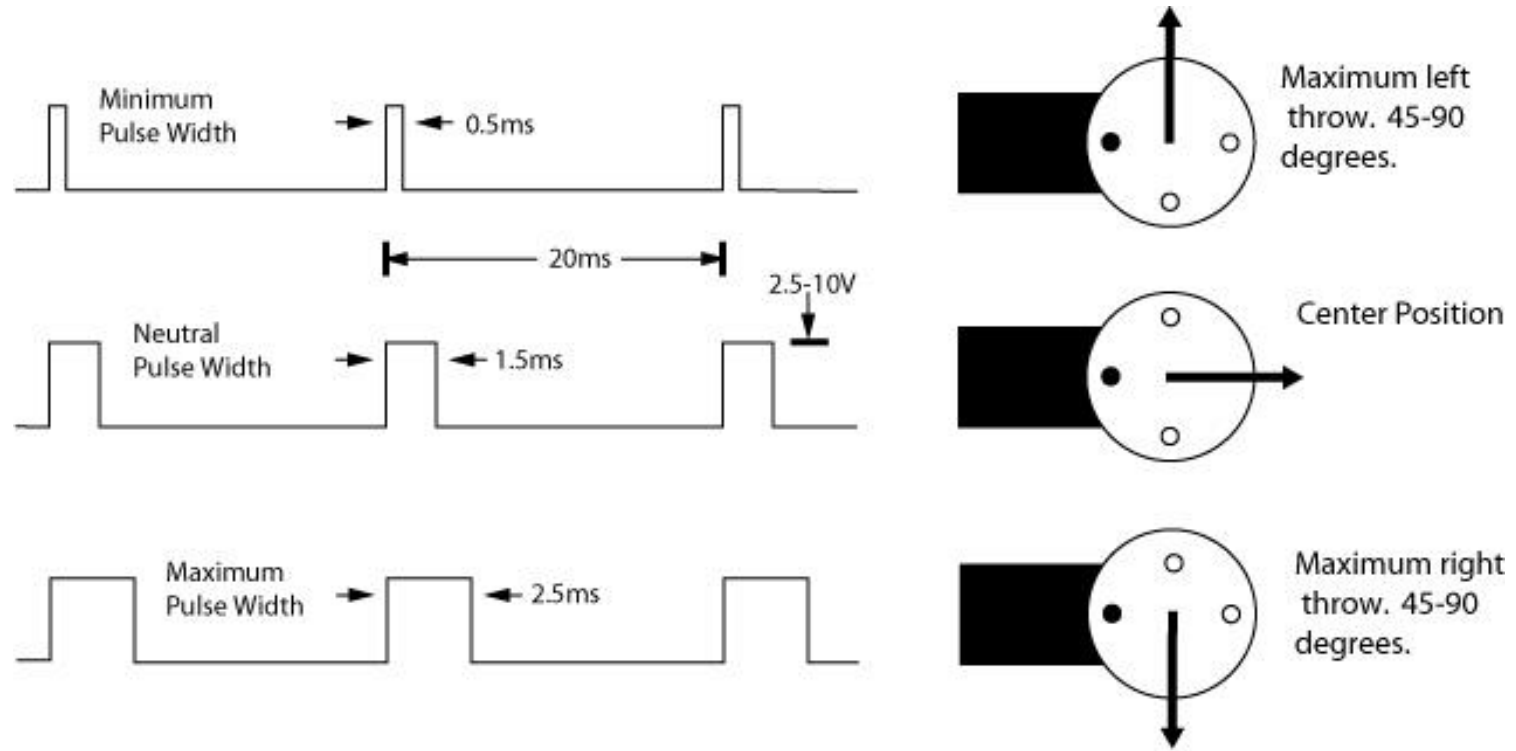
R/C tipi servo motorlarda enkoder yoktur. Bunun yerine şafta bağlanmış ve dönüşü algılayan potansiyometre vardır. Bu tip servo motorlar PWM (Puls Width Modulation - Darbe Genişlik Modülasyonu) tekniği ile çalışırlar. Kontrol ucuna gelen PWM sinyalinin görev çevrimine (duty cycle) göre belli açılarda dönme yaparlar. Bu tip servo motorlara R/C tip servo motor denmesinin nedeni genelde radyo frekansı ile uzaktan kontrol edilerek hobi amaçlı araba, uçak, helikopter vb. yapımlarında kullanılmasındandır. R/C kısaltması Radio Controlled kelimelerinin baş harflerinden meydana gelmiştir.

R/C servo motorlar derece esasına göre dönerler. PWM sinyallerini yorumlayarak gerekli dönme derecelerini hesaplar ve o değer ölçüsünde dönüş yaparlar. Kontrol sinyali aynı kaldığı müddetçe konumlarını korurlar. R/C servo motorlar 0° ile 180° arasında dönme gerçekleştirirler. Bu dönme dereceleri PWM sinyalinin görev çevriminin genişliğine göre değişir.



Servo Motorlar

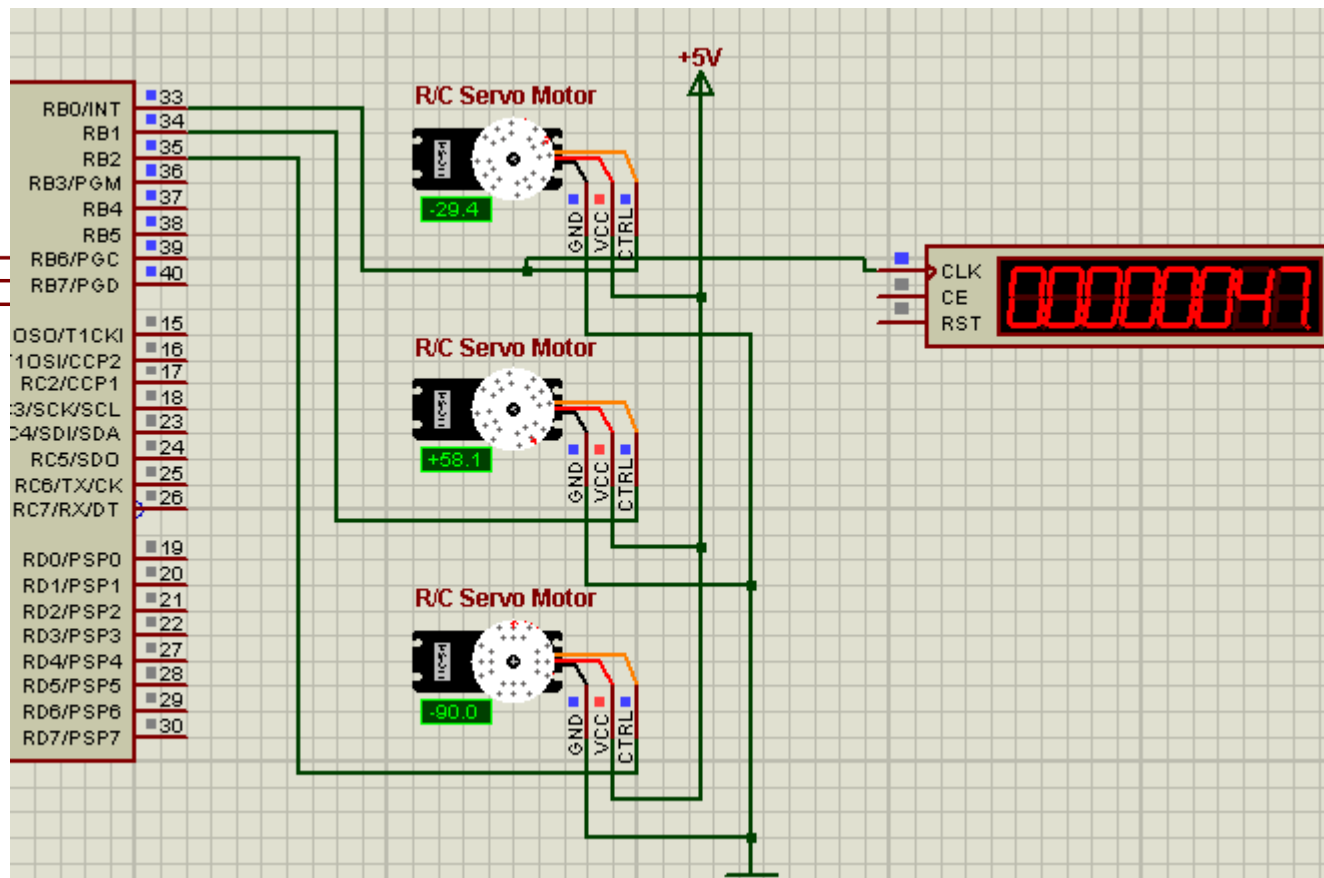
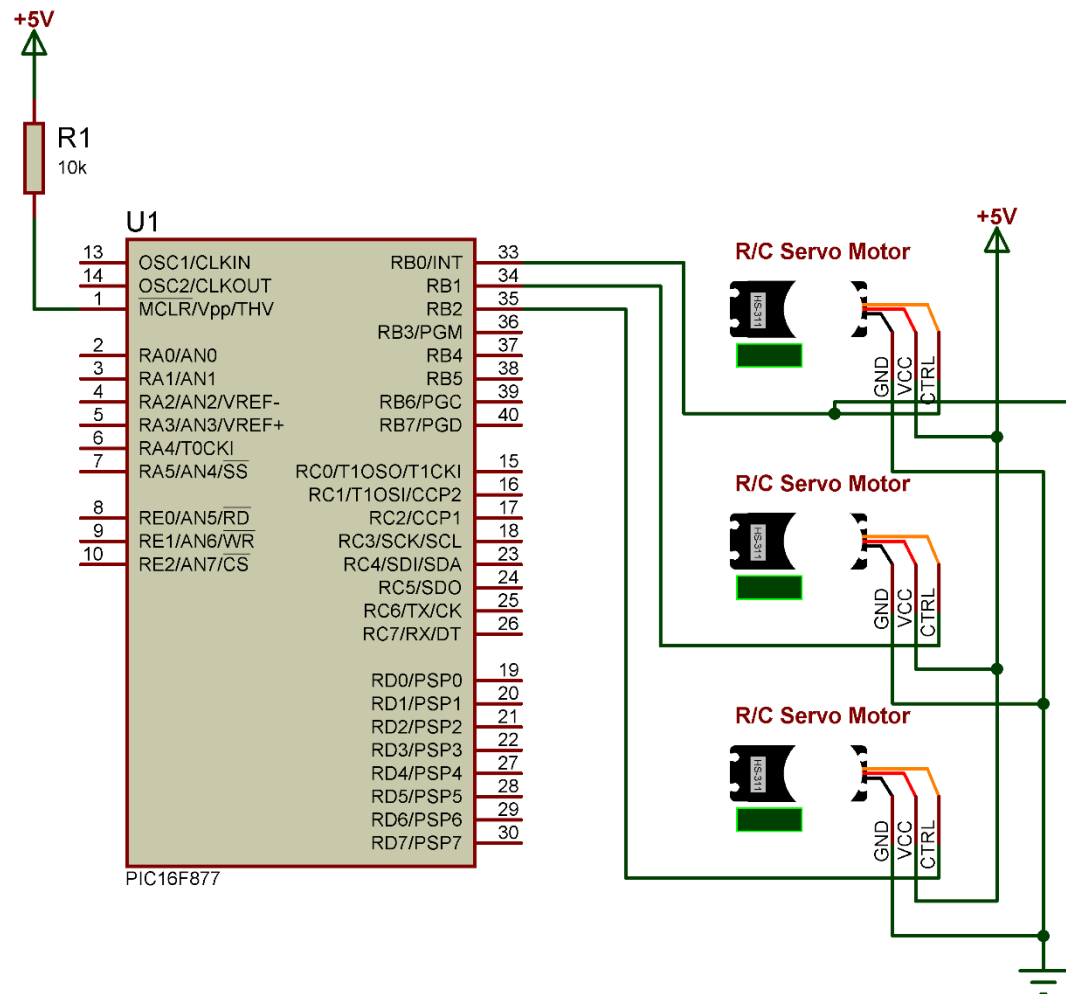
Dönmenin gerçekleşmesi için PWM sinyalinin frekansı 50 Hz olmalıdır. Yani R/C servo motorların kontrol sinyalinin periyodu 20 ms olmalıdır. 0° ile 180° arasındaki dönüş dereceleri ise PWM sinyalinin görev çevriminin (duty cycle). yaklaşık 0,5 ms ile 2,5 ms arasında değiştirilmesi suretiyle elde edilir. PWM sinyalinde 1,5 ms'lik görev çevrimi motoru merkez konuma (0°), 0,5 ms'lik görev çevrimi motoru tam sol (90° sola) konuma, 2,5 ms'lik görev çevrimi motoru tam sağ (90° sağ) konuma getirir. R/C servo motorların besleme gerilimleri ise DC 4,8 V ile 6 V arasında olmalıdır.



Servo Motor Uygulaması

Bu uygulamada 3 adet R/C servo motorun aynı anda ayrı ayrı derecelerde dönmeleri sağlanmıştır. R/C servo motorların kontrolü için gerekli PWM sinyali PIC denetleyici içinde bulunan CCP modülünde PWM birimi ile sağlanabilir. Ama çoğu PIC'de 1 veya 2 adet CCP modülü bulunmakta veya hiç bulunmamaktadır. Bu da fazla sayıda R/C servo motorun kontrolü için yeterli değildir. Bu nedenle oluşturacağınız bir algoritma fazla sayıda servo motoru kontrol edebilmeniz gerekmektedir. Bu uygulamada oluşturulan bir algoritma ile 3 adet R/C servo motor aynı anda farklı derecelerde kontrol edilebilmektedir. İstenirse daha fazla R/C servo motor da kullanılabilir.

Servo Motor Uygulaması



Örnek Programlar (Servo Motor Uygulaması)

```
1  /****
2      PIC16F877 İle R/C Servo Motor Uygulaması
3  *****/
4  #include <16f877.h>      // Kullanılacak denetleyicinin başlık dosyası
5
6  #fuses HS,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPPD
7
8  #use delay (clock=2000000) // Gecikme fonksiyonu için kullanılacak c
9
10 // R/C Servo motor dönme açı değerleri
11 const int8 servo_derece_1[]={8,9,10,11,12,13,14,15,16,17,18};
12 const int8 servo_derece_2[]={18,17,16,15,14,13,12,11,10,9,8};
13 const int8 servo_derece_3[]={8,12,18,8,12,18,8,12,18,8,12};
14
15 int i=0,pwm=0,duty_0=0,duty_1=0,duty_2=0;
16 int16 zaman=0; // 16 bitlik değişken tanımlanıyor
17
18 #int_timer0 // Timer0 taşma kesmesi
19 void kesme ()
```

Örnek Programlar (Servo Motor Uygulaması)

```
18  #int_timer0 // Timer0 taşma kesmesi
19  void kesme ()
20  {
21      set_timer0(113); // TMR0 kaydedicisine 113 değeri yükleniyor
22      if (pwm==0)      // Eğer PWM değişkeni 0 ise
23      {
24          output_high(pin_b0); // RB0 çıkışı lojik-1
25          output_high(pin_b1); // RB1 çıkışı lojik-1
26          output_high(pin_b2); // RB2 çıkışı lojik-1
27      }
28
29      if (pwm>=duty_0) output_low(pin_b0);
30      if (pwm>=duty_1) output_low(pin_b1);
31      if (pwm>=duty_2) output_low(pin_b2);
32
33      zaman++; // zaman değişkenini 1 arttır
34
35      // Servo motor dönüş adımları arası bekleme süresi için
36      if (zaman>17350) // 17350x114,4µsn=1.984.840µsn, yaklaşık 2msn
37      {
```

Örnek Programlar (Servo Motor Uygulaması)

```
36  if (zaman>17350) // 17350x114,4µsn=1.984.840µsn, yaklaşık 2msn
37  {
38      zaman=0; // zaman değişkenini sıfırla
39      i++; // i değişkeni değerini 1 arttır
40      if(i==11) // Eğer i değeri 11 ise-Tüm adımlar bitti ise
41          i=0; // i değişkenini sıfırla
42  }
43
44  pwm++; // pwm değişkenini 1 arttır
45  if (pwm>=173) // pwm değeri 173'den büyük ise
46      pwm=0; // pwm değişkenini sıfırla
47  }
48
49  /***** ANA PROGRAM FONKSİYONU*****/
50
51  void main ()
52  {
53      setup_psp(PSP_DISABLED); // PSP birimi devre dışı
54      setup_timer_1(T1_DISABLED); // T1 zamanlayıcısı devre dışı
55      setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
56      setup_adc_ports(NO_ANALOGS); // ANALOG giriş yok
57      setup_adc(ADC_OFF); // ADC birimi devre dışı
```

Örnek Programlar (Servo Motor Uygulaması)

```
58      setup_CCP1(CCP_OFF);           // CCP1 birimi devre dışı
59      setup_CCP2(CCP_OFF);           // CCP2 birimi devre dışı
60
61
62      setup_timer_0(RTCC_INTERNAL | RTCC_DIV_4); // Timer0 ayarları belirtiliyor
63      set_timer0(113); // TMR0 kaydedicisine 113 değeri yükleniyor
64
65      enable_interrupts(int_timer0); // Timer0 taşma kesmesi aktif
66      enable_interrupts(global);     // Aktif edilen tüm kesmelere izin ver
67
68      output_b(0x00); // İlk anda B portu çıkışı sıfırlanıyor
69
70      while(1) // Sonsuz döngü
71      {
72          duty_0=servo_derece_1[i]; // 1. R/C servo PWM görev saykılı
73          duty_1=servo_derece_2[i]; // 2. R/C servo PWM görev saykılı
74          duty_2=servo_derece_3[i]; // 3. R/C servo PWM görev saykılı
75      }
76  }
```

Kaynaklar

- CCS C Programlama Kitabı, Serdar Çiçek, Altaş Yayıncılık
- Mikroelektronika C programlama e-kitabı «<https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c>»
- Mikroişlemciler Lab. EE-304 dersi deney föyleri, Prof. Dr. Mehmet DEMİRTAŞ