

BMS 302 MİKROKONTROLCÜLER

MOTOR UYGULAMALARI

Prof.Dr. Mutlu AVCI

Çukurova Üniversitesi Biyomedikal Mühendisliği Bölümü

Bahar 2020



Dört çeşit motor ile hayatımızın pek çok aşamasında karşılaşıyoruz, bunlar:

- DC motor
- Fırçasız (brushless) DC motor
- Adım (stepper) motor
- Servo motor

DC Motorlar

- Dönüş yönü + - kutuplamasının yer deęiřtirmesiyle deęiřir.
- Hız kontrolü PWM ile yapılır.
- H-köprü (H-Bridge) motor sürücü entegreleri ile sürülür.

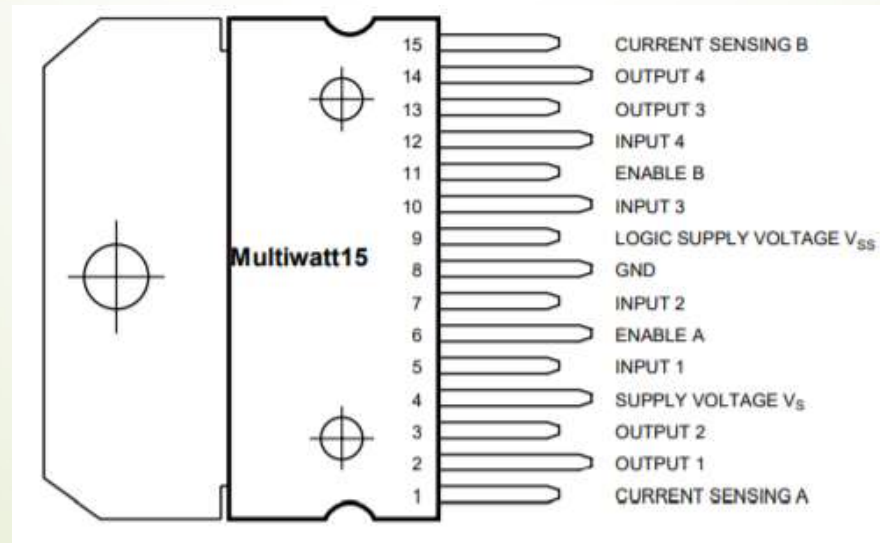
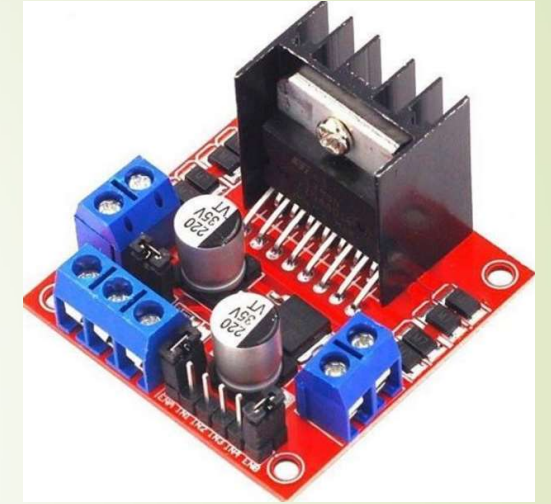
Bu entegreler uygulamaya ve motorun özelliklerine (akım çekmesi, PWM frekansı gibi) göre seçilir.

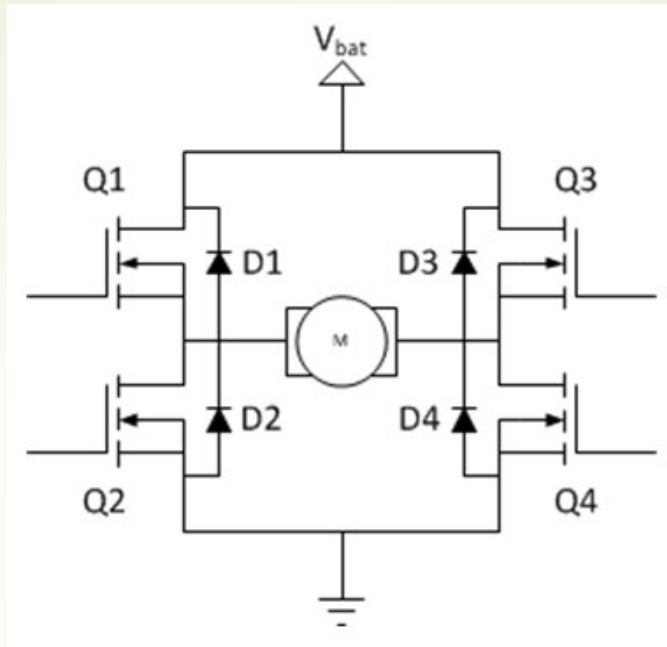


L298 H-Köprü Motor Sürücü



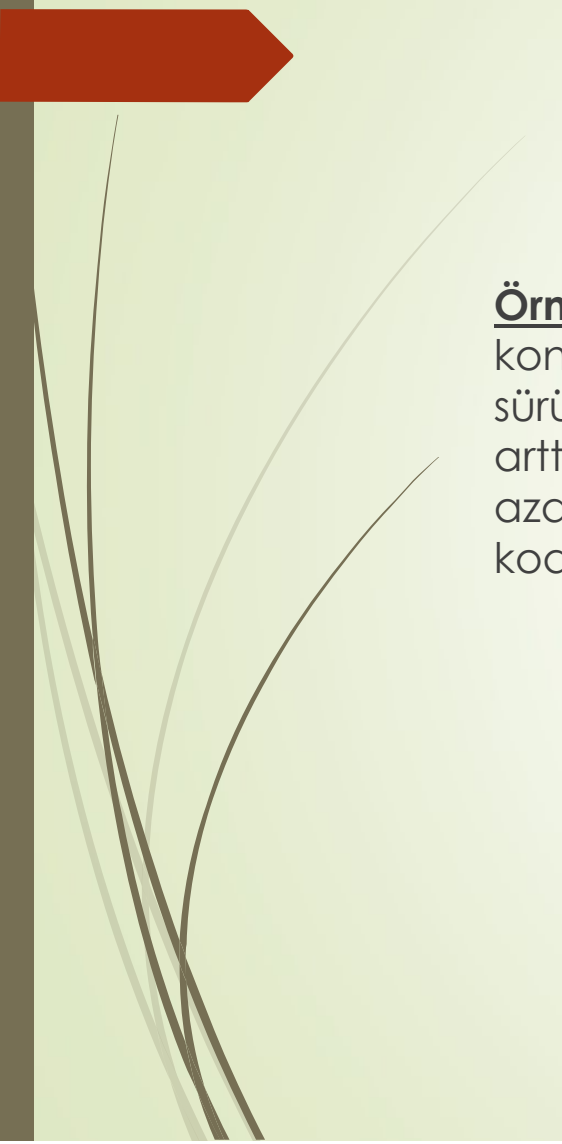
- Modül veya entegre halinde kullanılabilir.
- Aynı anda iki motor sürebilir.



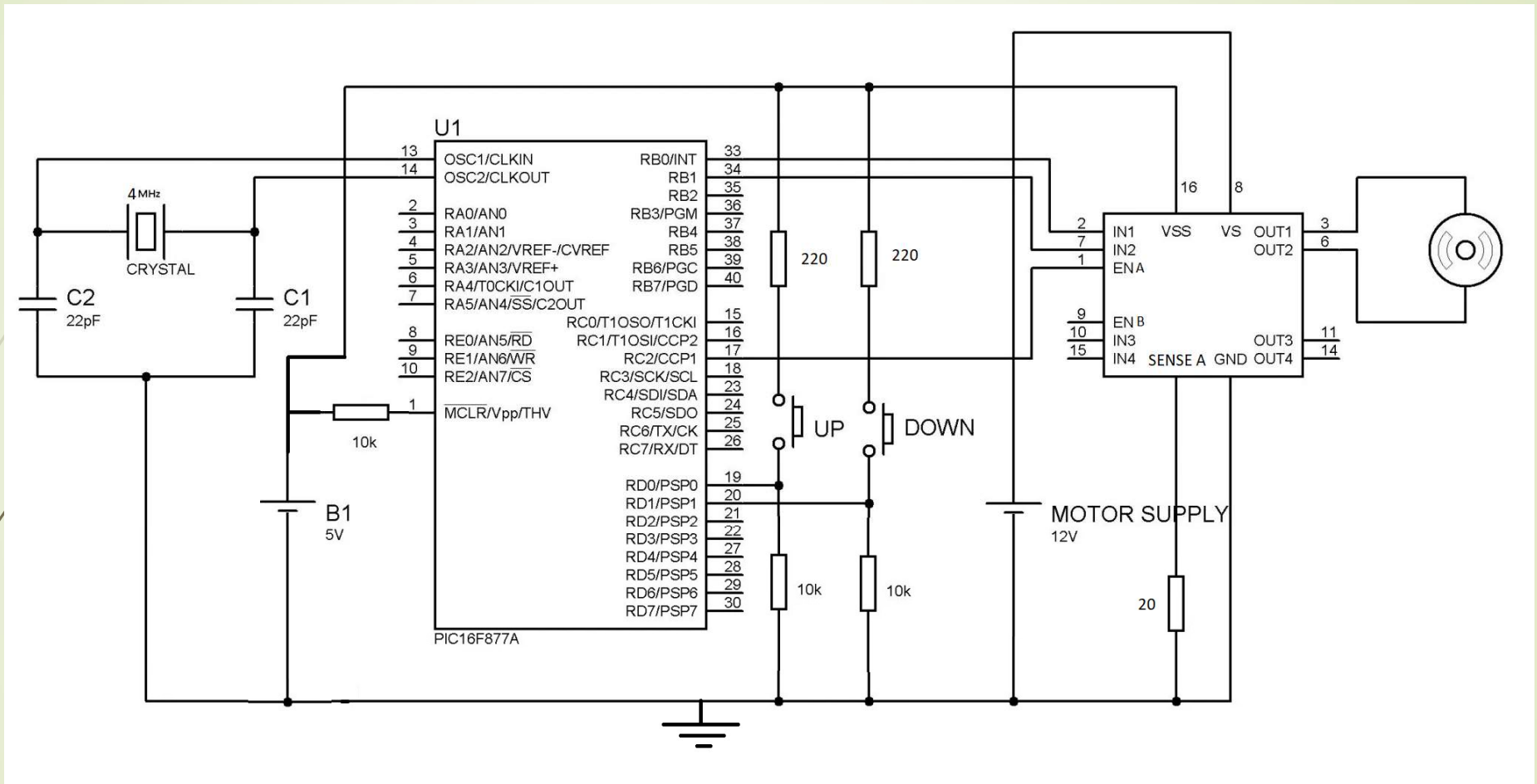


Q1	Q2	Q3	Q4	Motor Durumu
1	0	0	1	Saat yönünde döner
0	1	1	0	Saatın tersi yöne döner
1	0	1	0	Motor durur
0	0	0	0	Motor boşta

Yalnız A kısmı	Enable A	Enable B	IN 1	IN 2	Durum		
	1	0	1	0	İleri		
	1	0	0	1	Geri		
	1	0	Input 2	Input 1	Fren		
	0	0	X	X	Boşta		
Yalnız B kısmı	Enable A	Enable B	IN3	IN4	Durum		
	0	1	1	0	İleri		
	0	1	0	1	Geri		
	0	1	Input 4	Input 3	Fren		
	0	0	X	X	Boşta		
A ve B kısmı birlikte	Enable A	Enable B	IN1	IN2	IN3	IN4	Durum
	1	1	1	0	1	0	A→ İleri B→ İleri
	1	1	1	0	0	1	A→ İleri B→ Geri
	1	1	0	1	0	1	A→ Geri B→ Geri
	1	1	0	1	1	0	A→ Geri B→ İleri
	1	1	IN 2	IN1	IN 4	IN3	Fren
	0	0	X	X	X	X	Boşta



Örnek: PIC 16F877A mikrokontrolcüsü kullanılarak bir 12 V luk bir DC motorun hız kontrolü yapılmak istenmektedir. Bu DC motor 10 kHz PWM frekanslı olarak sürülmelidir. D0 ucuna bağlı bir push button ile hız maksimum hızın %1'i kadar arttırılmakta, D1 ucuna bağlı başka bir push buttona her bir basışta %1 azalmaktadır. Bu işlevi yerine getiren devreyi tasarlayıp CCS C program kodunu yazın.



CCP PWM Modu Hatırlatma

- CCP birimi PWM modu, istenen CCPX ucundan istenen görev çevirimine (duty cycle – doluluk oranı) sahip PWM sinyali elde etmek için kullanılır.
- PWM birimi Timer2 zamanlayıcısını kullanır. PIC16F877A'da iki adet CCP modülü olduğundan iki adet de PWM çıkış ucu vardır.
- Bu uçlar RC1/CCP2 ve RC2/CCP1'dir.
- PWM sinyali; denetleyici osilatör frekansı, Timer2 zamanlayıcı biriminin PR2 değeri ve bölme oranı değeri ile belirlenir.

- 
- Timer2 fonksiyonunda bulunan *postscaler* değerinin PWM sinyaline bir etkisi yoktur. PWM sinyalinin periyodu ve frekansı ;


$$TPWM = T_{komut} \times (\text{Timer2 Bölme Oranı}) \times (PR2 \text{ değeri} + 1)$$

$$T_{komut}(sn) = 1 / f_{komut}(hz)$$

$$f_{komut} = \text{Kontrolcü Osilatör Frekansı} / 4 = f_{osc} / 4$$

$$f_{PWM} = 1 / TPWM$$

şeklinde hesaplanır



➤ Örneğin ;

#use delay (clock=4000000)

setup_timer_2(T2_DIV_BY_1, PR2+1, 1);

komutları kullanılarak bir PWM sinyalinin frekansı 10 kHz e ayarlanmalıdır.

$f_{komut} = f_{osc}/4 = 1 \text{ Mhz}$

$TPWM = 1/f_{komut} \times (PR2+1) \times (\text{Timer2 bölme oranı}) = 1/10k$

$(1/(1 \text{ MHz})) \times (PR2+1) \times 1 = 0.1 \text{ ms}$

PR2=99

şeklinde hesaplanır.



SET PWMX DUTY() Fonksiyonu:

- PWM modunda oluşturulan sinyalin görev çevrim süresini (duty cycle) belirlemeye yarayan fonksiyondur. Kullanımı ;

set_pwmx_duty(değer);

SETUP_CCPX () Fonksiyonu

- Bu fonksiyon ile mikro kontrolcü içinde bulunan CCP modülü ile ilgili ayarlamalar yapılır. Kullanımı ;

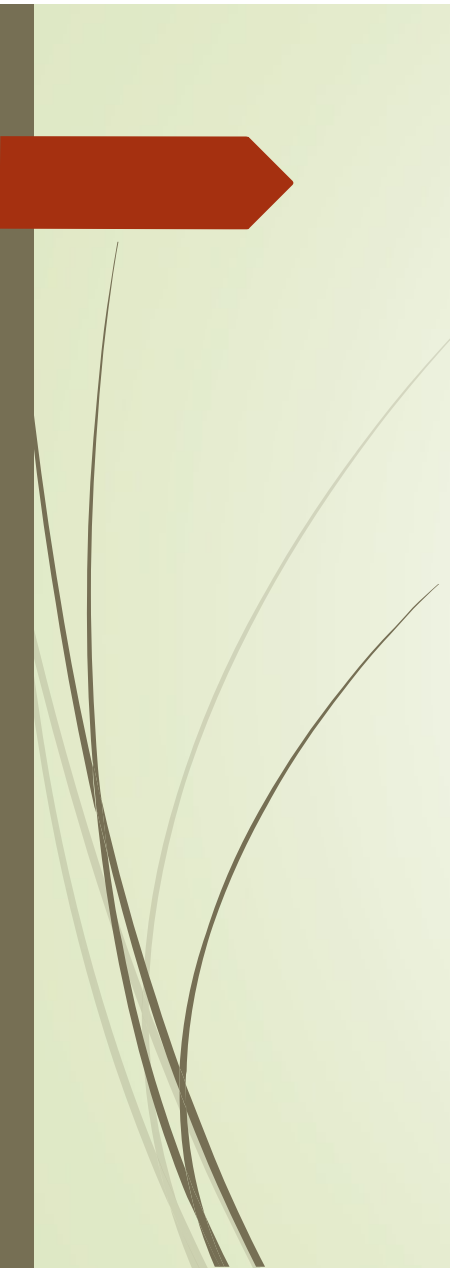
setup_ccp1 (mod) ;

setup_ccp2(mod);

şeklindedir.

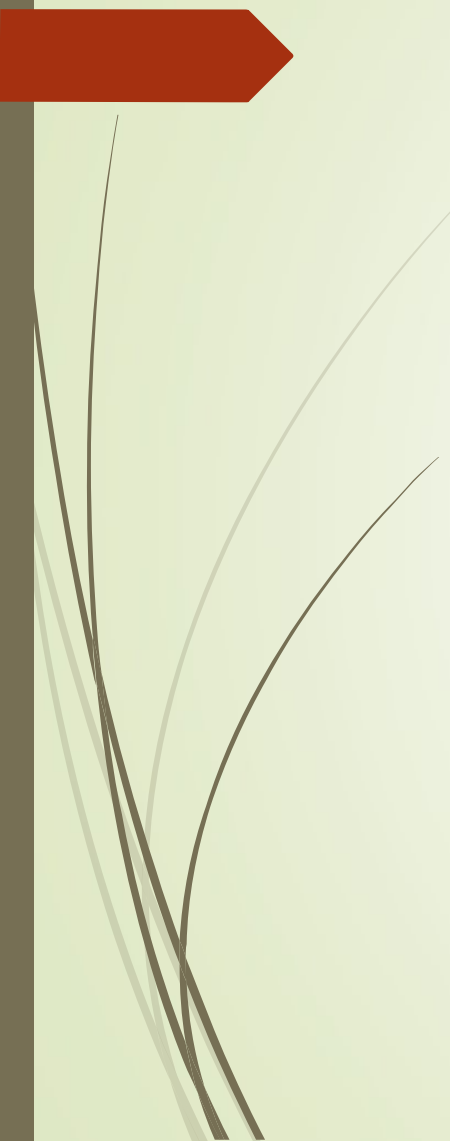
“mod” kısmına kullanılmak istenen moda göre sabit tanımlamalar yazılır.

- PWM modu için mod kısmına, **CCP_PWM** yazılır.
- CCP birimini devre dışı yapmak için **CCP_OFF** yazılır.



```
#include <prog.h>
#use delay (clock=4000000)
#use fast_io(C)
int8 i=0;
void main(){
  set_tris_c(0x00);
  setup_ccp1(CCP_PWM);
  setup_timer_2(T2_DIV_BY_1, 99, 1);
  output_high(pin_B0);
  output_low(pin_B1);

  while(1){
    if (input(pin_D0)){
      while(input(pin_D0));
      i++;
      if(i>=99) i=99;
      set_pwm1_duty(i);
    }
  }
```

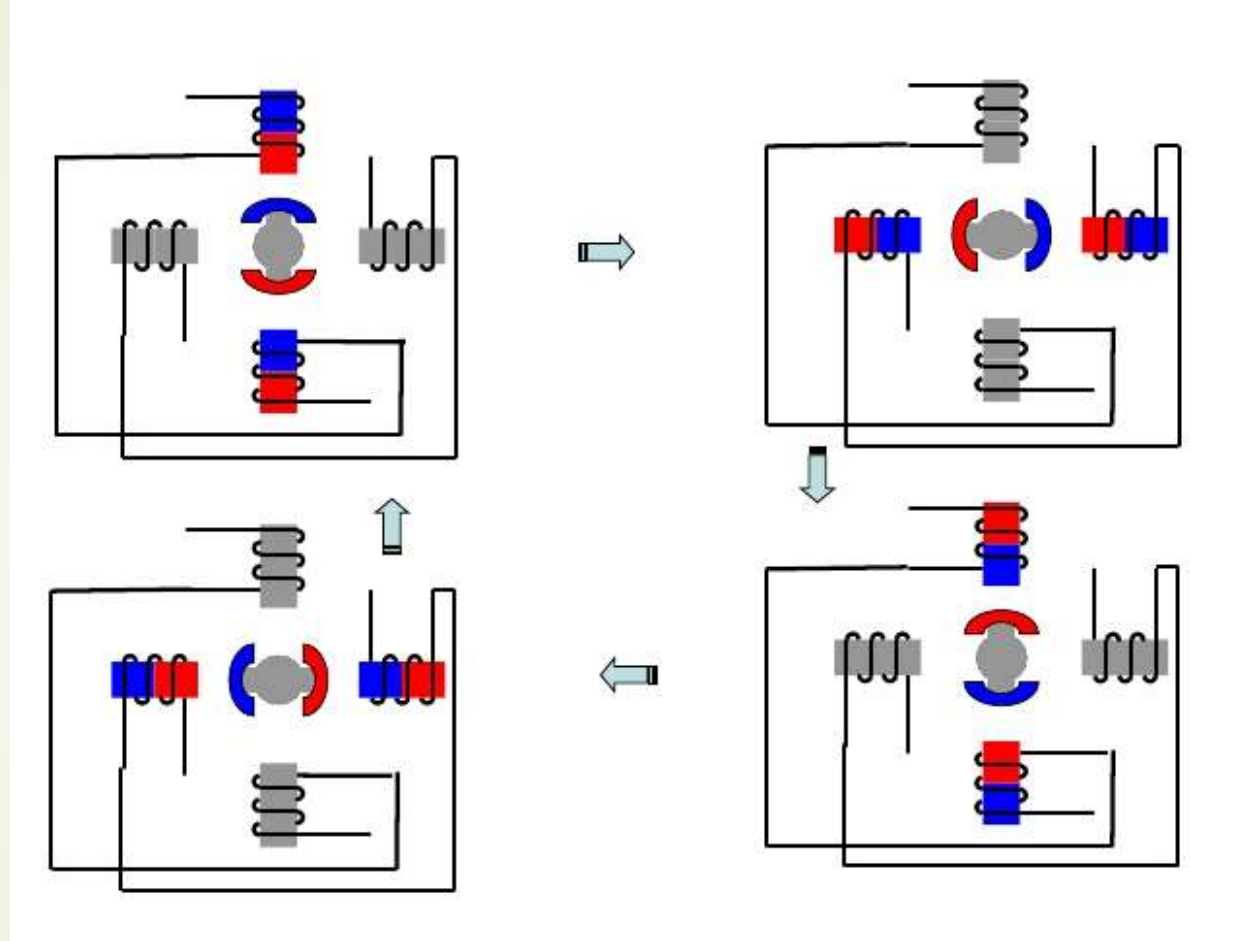


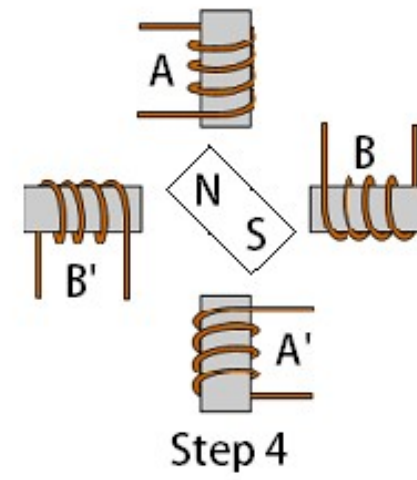
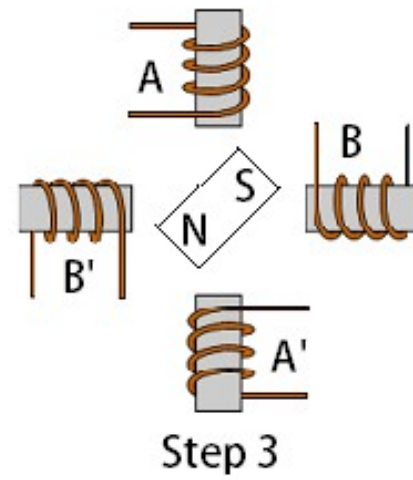
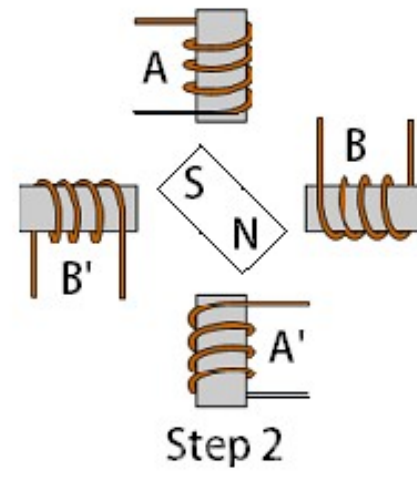
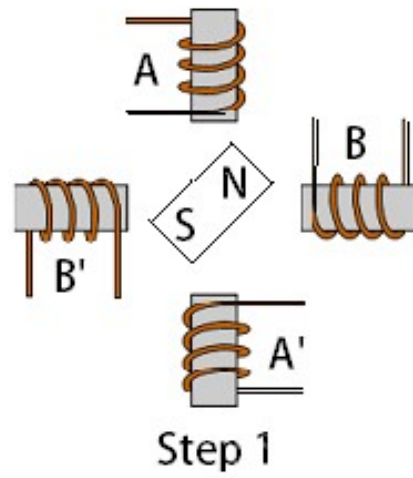
```
if (input(pin_D1)){  
  while(input(pin_D1));  
  i--;  
  if(i>=0) i=0;  
  set_pwm1_duty(i);  
}  
}  
}
```



Adım (Stepper) Motor

- Tam Adım
- Yarım Adım
- Mikro Adım





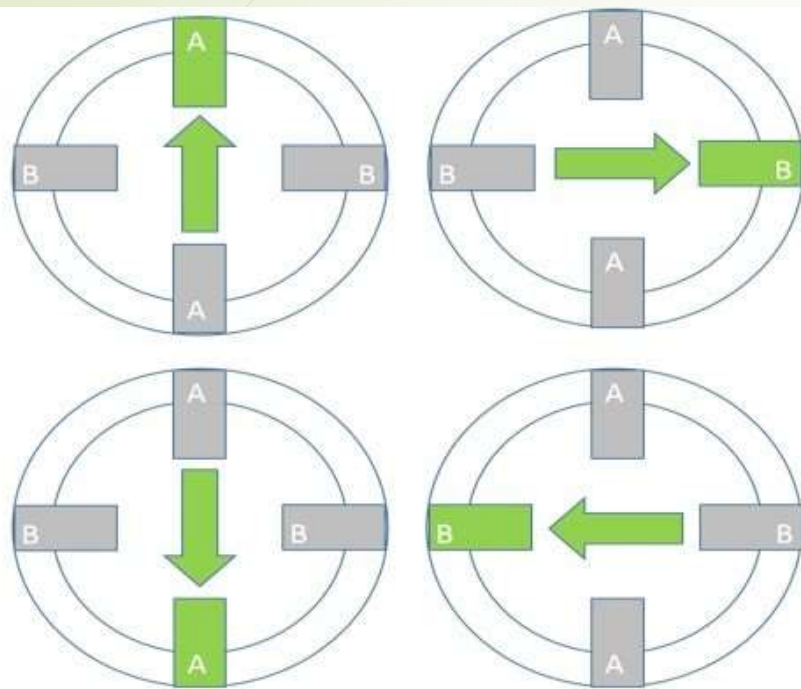


Fig 1 – One phase on – full step

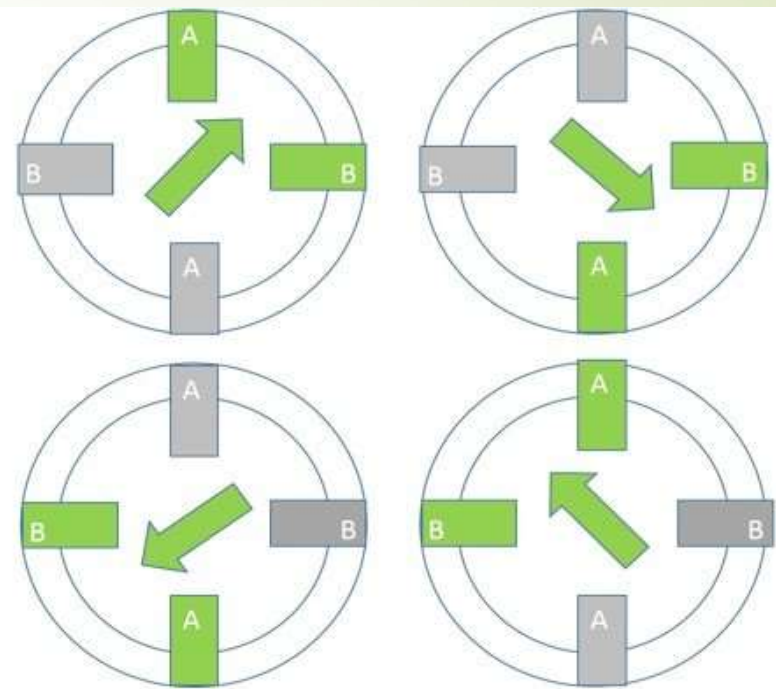


Fig2 – Two phase on – full step

Tam adım(tek faz modu)

STEP	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Tam adım (dual faz modu)

STEP	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

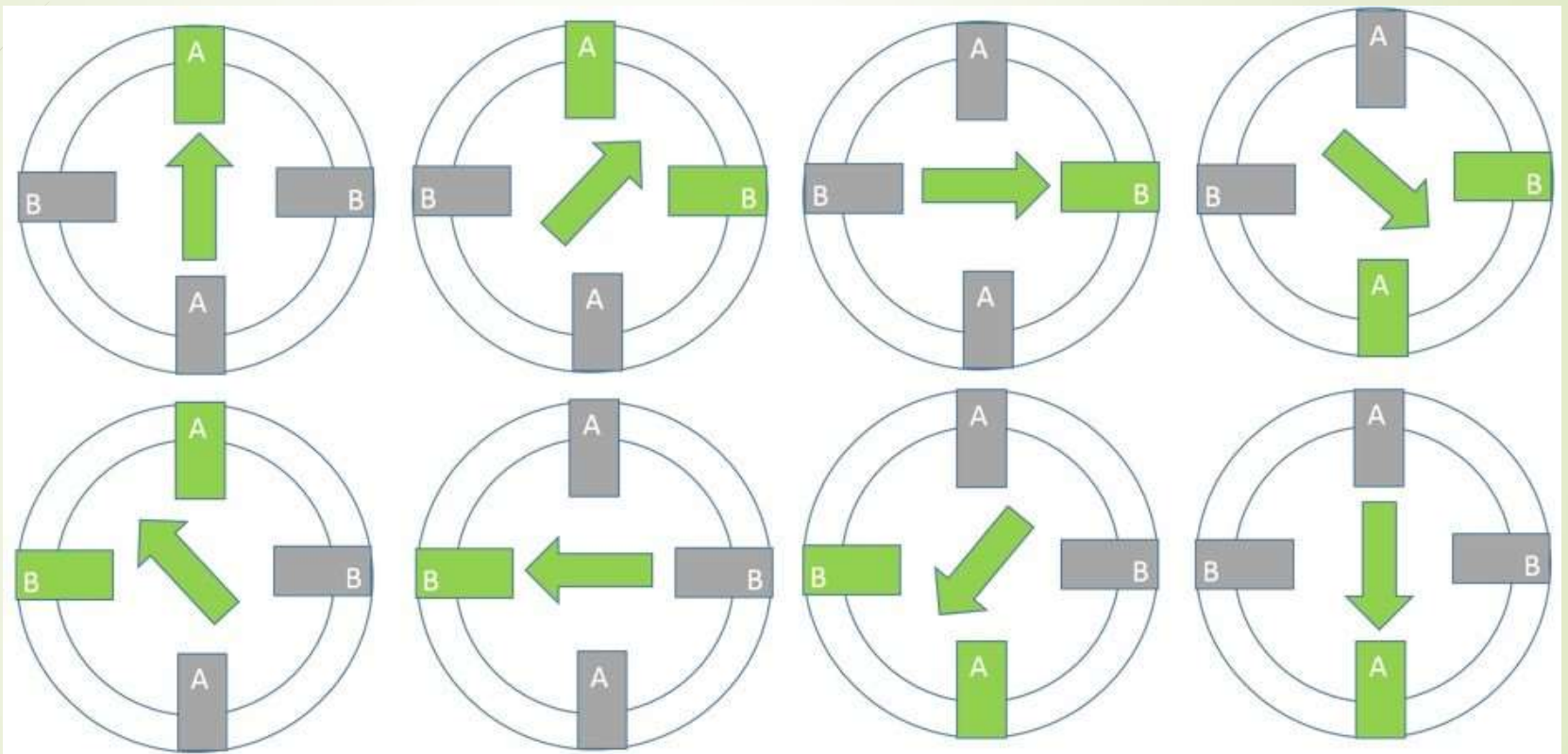




Fig3 - One-two phase on - half step

Yarım adım modu

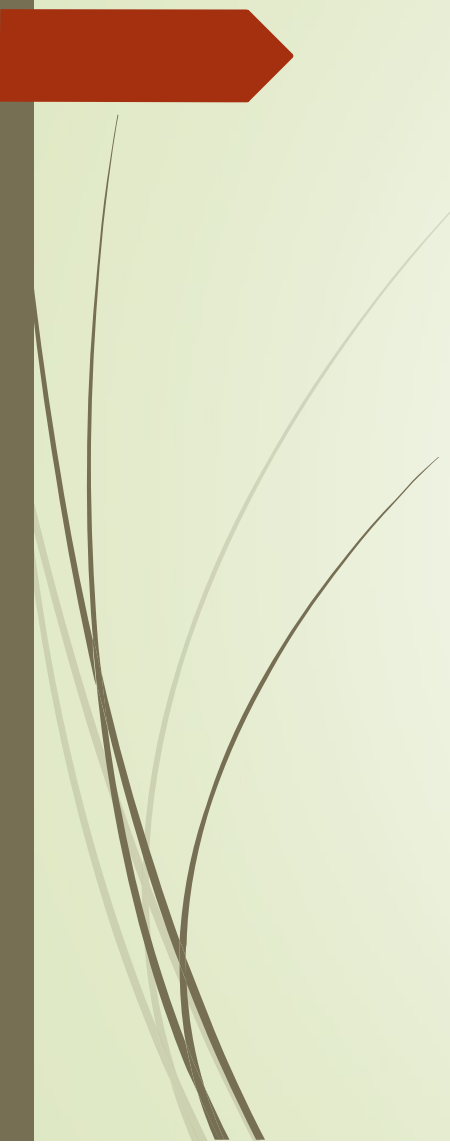
STEP	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1



Örnek: Bir PIC 16F877A mikrokontrolcüsü kullanarak yarım adım 5 V beslemeli adım motor süren, D0 ucundaki push button a her basıldığında saat yönünde bir adım, D1 ucundaki push button a her basıldığında tersi yönde bir adım döndüren, D2 ucundaki button ile dönmeyi durduran, devreyi tasarlayın ve CCS C programını yazın.



```
#include <prog.h>
Void main(){
int i=0;
int8 yarim_adim[]={1,3,2,6,4,12,8,9};
output_B(0);
while(1){
if(input(pin_D0)){
while(input(pin_D0));
i=(i+1)%8;
output_B(i);
}
```



```
if(input(pin_D1)){  
while(input(pin_D1));  
i=(i-1)%8;  
output_B(i);  
}  
if(input(pin_D2)){  
break;}  
}  
}
```