



Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü

EE-302

Mikroişlemciler

CCP Modülü(Capture-Compare-PWM) Uygulamaları

10. Hafta

Prof. Dr. Mehmet DEMİRTAŞ

CCP BİRİMİNE GENEL BAKIŞ

- CCP birimi içinde 3 değişik modda çalışan yani 3 değişik modül içeren bir birimdir. Bu çalışma modları, Capture (yakalama), Compare (karşılaştırma) ve PWM (Puls Genişlik Modülasyonu) modudur. PIC16F877 içinde 2 adet CCP birimi vardır. Bunlar CCP1 ve CCP2'dir. Her CCP birimi, 16 bit'lik capture kaydedicisi olarak, 16 bit'lik compare kaydedicisi veya 10 bit'lik PWM Master/Slave görev çevrimi kaydedicisi olarak çalışan 16 bit'lik kaydediciye sahiptir. CCP birimi donanımı sayesinde program komutlarını kullanmadan yakalama (capture), karşılaştırma (compare) ve PWM sinyali üretme işlemleri kolayca yapılabilir. Bu işlemler için her CCP birimi çalışma modu denetleyici içinde bulunan bir zamanlayıcı/sayıcı birimini kaynak olarak kullanır. Tablo 1'de bu CCP biriminin kullandığı zamanlayıcı/sayıcı kaynakları verilmiştir.

CCP BİRİMİNE GENEL BAKIŞ

CCP Modu	Kullandığı Zamanlayıcı
Capture	Timer1
Compare	Timer1
PWM	Timer2

CCPx birimini CCP1CON ve CCP2CON kaydedicileri kontrol etmektedir. İki kaydedicinin yapısı ayındır. Şekil 2'de CCPxCON kaydedici bit'leri ve bit'lerin görevleri verilmiştir. Şekilde bit isimlerindeki x yerine CCP1CON kaydedicisi için 1, CCP2CON kaydedicisi için 2 numarası gelmektedir.

CCP1CON Register

	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
CCP1CON	-	-	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
								Bit name

Bit 3-0 **CCPxM3:CCPxM0** = CCPx modu seçme bit'leri

Bit 7-6 = Kullanılmayan pin'ler. Lojik-0 olarak okunur.

Bit 5-4 **CCPxX-CCPxY** = PWM düşük değerlikli bit'leri.
(Bu bit'ler Capture ve Compare modlarında kullanılmaz.)

PWM görev çevriminin düşük değerlikli 2 bit'inin değerini belirtir. 10 bit'lik görev çevrimi değerinin yüksek değerlikli 8 bit'lik kısmı ise CCPRxL kaydedicisinde bulunur.

0000 = Capture/Compare/PWM Kapalı.

0100 = Capture mod, her düşen kenarda.

0101 = Capture mod, her yükselen kenarda.

0110 = Capture mod, her dördüncü yükselen kenarda.

0111 = Capture mod, her onaltıncı yükselen kenarda.

1000 = Karşılaştırma modunda Eşleşme olayı meydana geldiğinde CCPx pin'i "1" olsun ve CCPx kesmesi çağrılsın.

1001 = Karşılaştırma modunda Eşleşme olayı meydana geldiğinde CCPx pin'i "0" olsun ve CCPx kesmesi çağrılsın.

1010 = Karşılaştırma modunda Eşleşme olayı meydana geldiğinde CCPx pin'i durumu değişmesin ve CCPx kesmesi çağrılsın.

1011 = Karşılaştırma modunda Eşleşme olayı meydana geldiğinde CCPx pin'inin durumu değişmesin ve Timer1 resetlensin.

11xx = PWM modu.

CAPTURE MODU

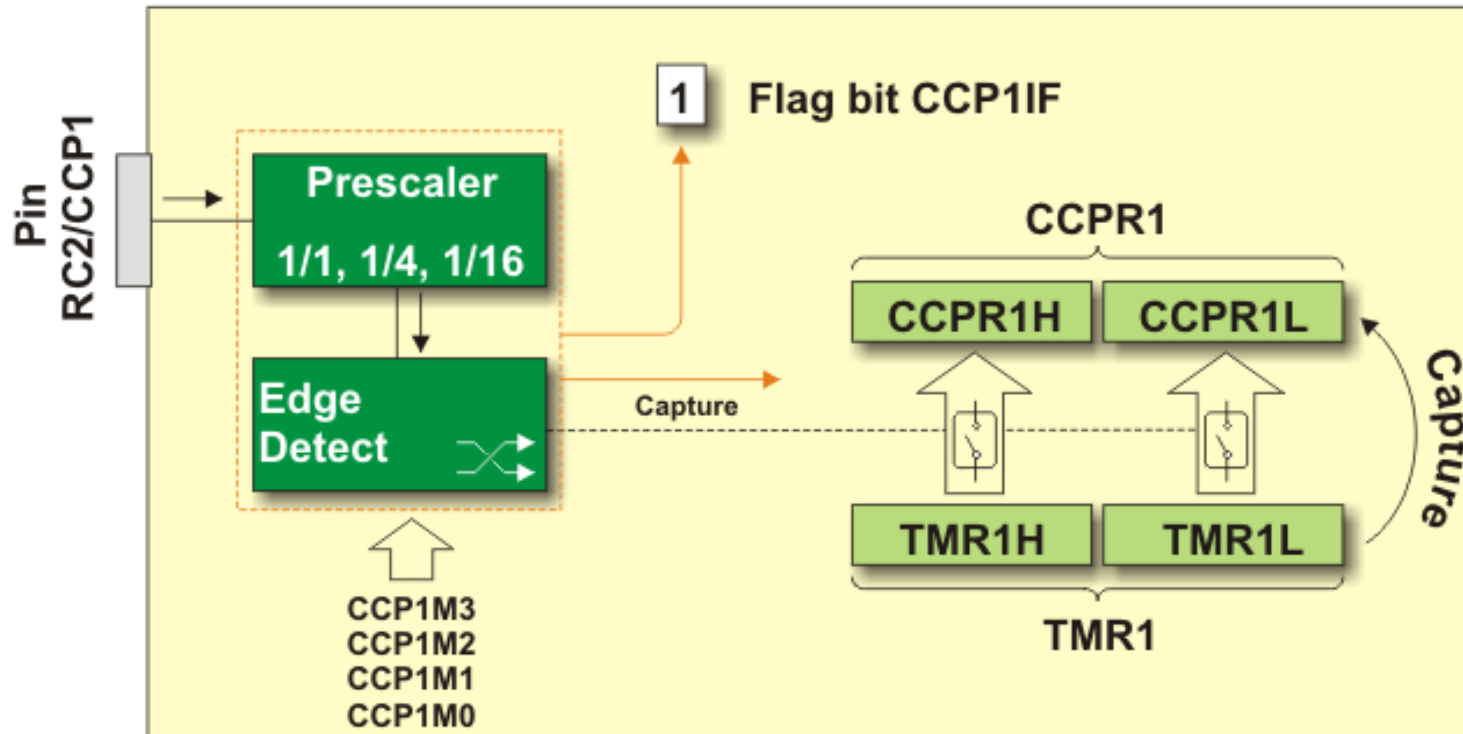
- CAPTURE modunda belirlenen RCx/CCPx (CCP1 birimi için RC1/CCP1 pin'i, CCP2 birimi için RC2/CCP2 pin'i) pin'inden gelen clock sinyali aşağıda verilen şartlara göre değerlendirilir. Bu şartlardan biri sağlandığında (yakalama olayı meydana geldiğinde) TMR1 kaydedicisi içeriği CCPRxH:CCPRxL kaydedicisine aktarılır (Kaydedilecek değer 16 bit'lik olduğundan iki adet 8 bit'lik kaydedicide saklanır) ve bir kesme oluşur. İkinci bir yakalama olayı meydana geldiğinde yine o anki TMR1 değeri CCPRxH:CCPRxL kaydedicisine aktarılır, eski CCPRxH:CCPRxL değeri silinmiş olur.
- Sinyalin her düşen kenarında
- Sinyalin her yükselen kenarında
- Sinyalin her 4. yükselen kenarında
- Sinyalin her 16. yükselen kenarında

CAPTURE MODU

- CCPRxH kaydedicisi 16 bit'lik değerin yüksek değerlikli 8 bit'ini, CCPRxL kaydedicisi ise 16 bit'lik değerin düşük değerlikli 8 bit'ini içerir. CCPx pin'inden sinyal alımı için bu ucun giriş olarak yönlendirilmesi gerekmektedir. Tanımlamalarda yer alan "X " yerine kullanılan CCP biriminin numarası yazılır.
- Not: CCS C'de CCPx birimi ile işlem yaparken esneklik sağlaması açısından denetleyici başlık dosyasında CCPR1 kaydedicisi CCP_1 (16 bit'lik), CCPR2 kaydedicisi CCP_2 (16 bit'lik) olarak tanımlanmıştır. Ayrıca 16 bit'lik CCPR1 kaydedicisinin 8 bit'lik yüksek değerlikli kısmını tutan CCPR1H kaydedicisi CCP_1_HIGH, düşük değerlikli 8 bit'ini tutan CCPR1L kaydedicisi CCP_1_LOW olarak tanımlanmıştır. Aynı şekilde CCPR2H kaydedicisi CCP_2_HIGH, CCPR2L kaydedicisi CCP_2_LOW olarak tanımlanmıştır. Bu tanımlamaları kullanarak bu kaydedicilere istenen değerler direkt olarak yazılabilir.

CAPTURE MODU

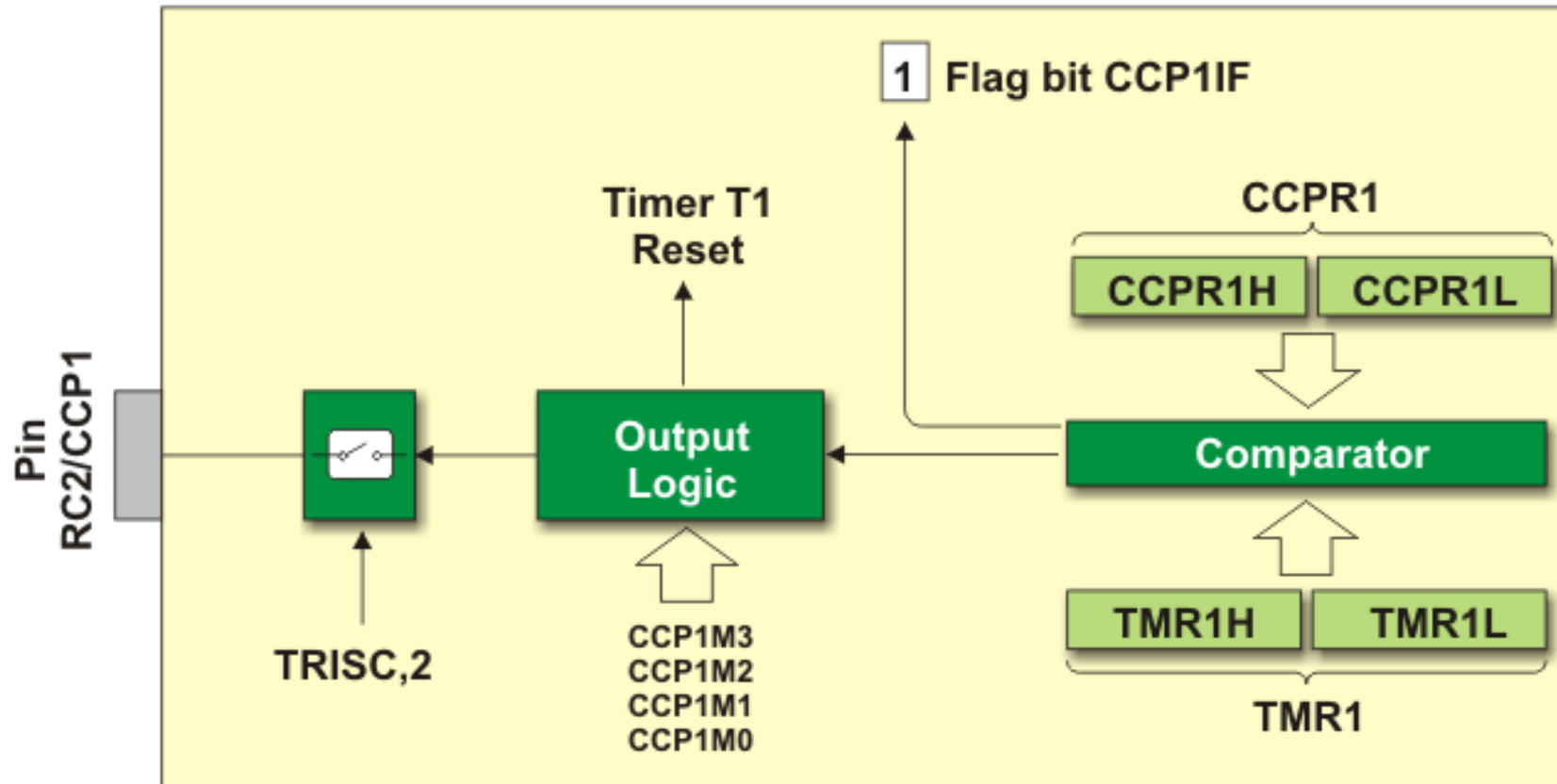
- Aşağıda bazı örnekler verilmiştir.
- CCP_1=600; //CCPR1 kaydedicisi içeriğine 600 değeri yüklenmiştir.
- CCP_1_HIGH=0x45; // CCPR1H kaydedicisi 8 bit 45h değeri yüklenmiştir.



COMPARE MODU

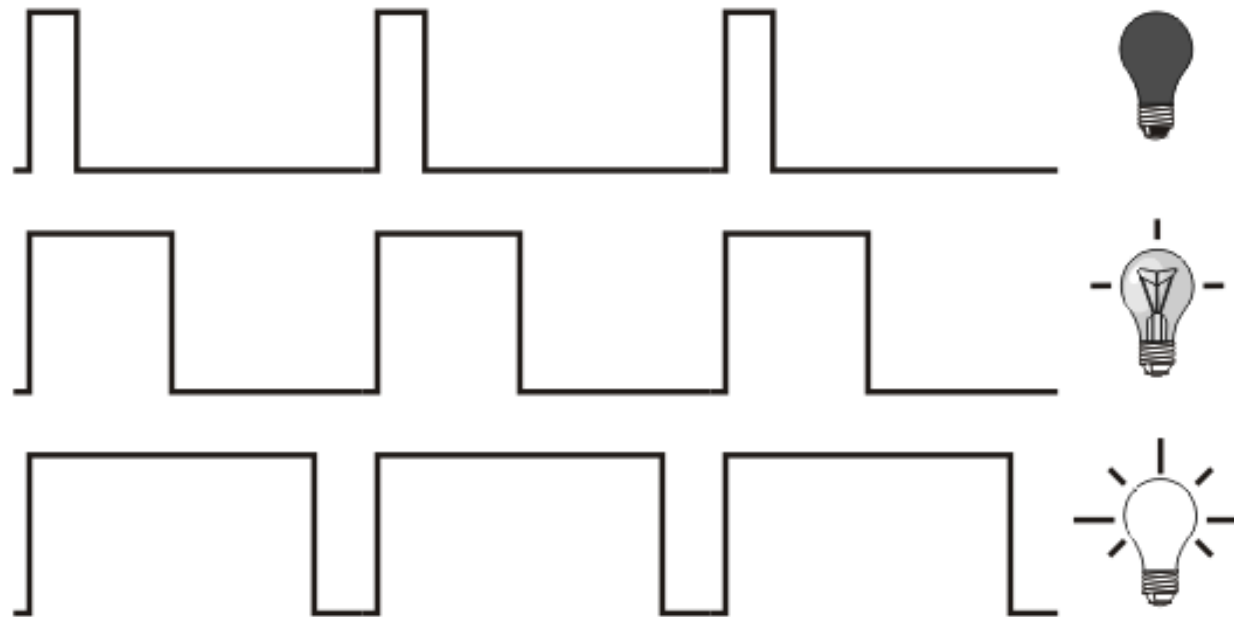
- CCP biriminin COMPARE modunun çalışma prensibi şöyledir. CCPRx kaydedicisinin içindeki değer ile Timer1 kaydedicisi olan TMR1'in içindeki değer sürekli olarak karşılaştırılır. Karşılaştırma işlemi sonucunda eşleşme meydana gelirse CCPx kesmesi meydana gelir ve CCPxCON kaydedicisinde belirtilen bit ayarları ile aşağıda belirtilen 4 olaydan istenen biri gerçekleştirilebilir. Şekil-3'te COMPARE modunun blok diyagramı görülmektedir. CCPx ifadesindeki x yerine hangi CCP modülü kullanılıyorsa onun numarası yazılır. Örneğin CCP2 modülü için x yerine 2 yazılmalıdır.
- CCPx pin'i lojik-1 olsun, kesme meydana gelsin
- CCPx pin'i lojik-0 olsun, kesme meydana gelsin
- CCPx pin'inin durumu değişmesin, kesme meydana gelsin
- CCPx pin'inin durumu değişmesin, kesme meydana gelsin ve Timer1 resetlesin.

COMPARE MODU



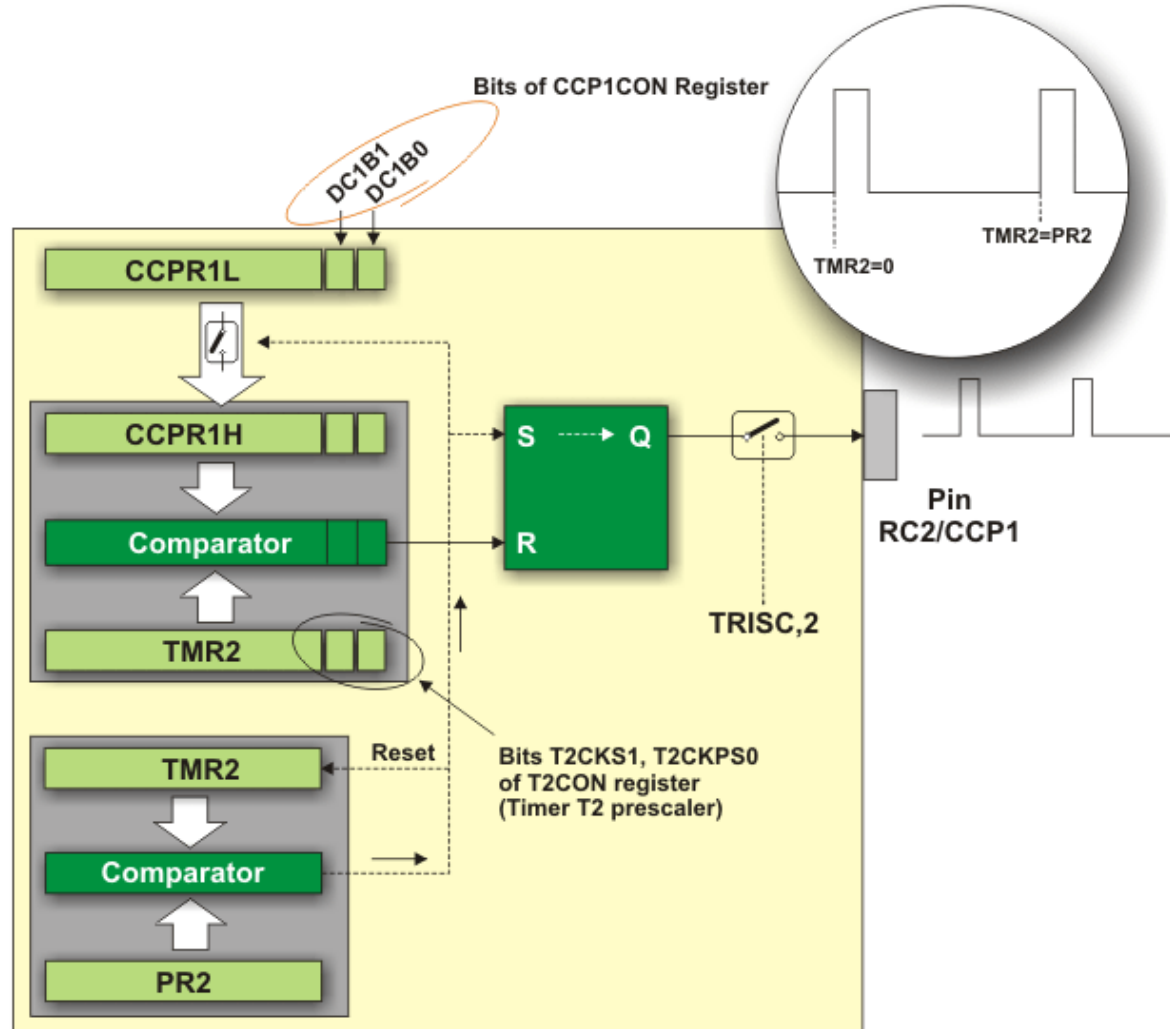
PWM MODU

- CCP biriminin PWM modu, istenen CCPx ucundan istenen görev çevrimine (duty cycle - doluluk oranı) sahip PWM sinyali elde etmek için kullanılır. PWM birimi Timer2 zamanlayıcısını kullanır. PIC16f877'de iki adet CCP modülü olduğundan 2 adet de PWM çıkış ucu vardır. PWM modunda, PWM çıkışı olarak kullanılan RC1/T10SI/CCP2 ve RC2/CCP1 pin'leri çıkış olarak yönlendirilmelidir.



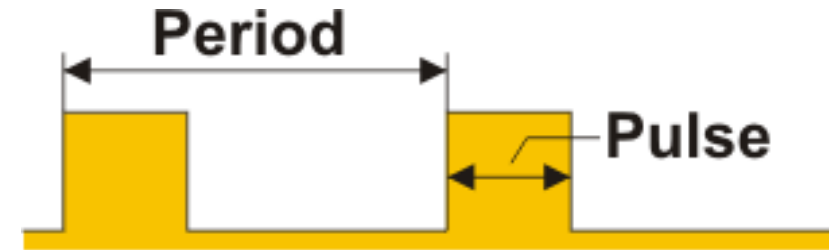
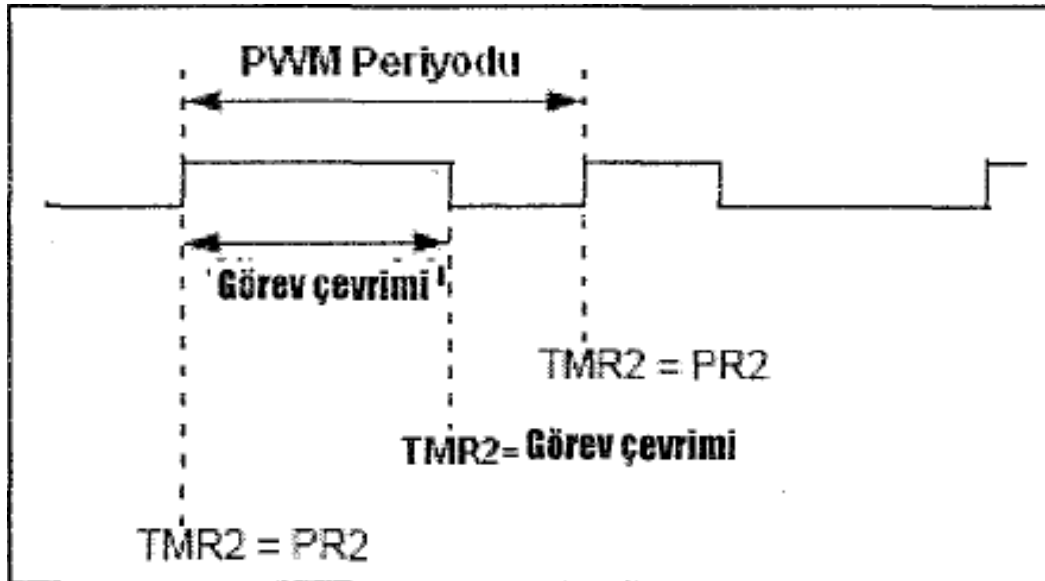
PWM MODU

- Aşağıda Şekil 5'te PWM modunun çalışmasının blok diyagramı verilmiştir.



PWM MODU

- Blok şemada da görüldüğü gibi oluşturulacak PWM sinyali denetleyici osilatör frekansına, PR2 değerine ve Timer2 zamanlayıcı/sayıcı biriminin bölme oranı değeri ile belirlenir. Aşağıda oluşturulan PWM sinyalinin, periyod ve frekans hesaplanması için kullanılacak formül ile oluşturulan bir PWM dalga şekli verilmiştir.
- PWM sinyalinin periyodu ve frekansı:
- $T_{pwm} = T_{osc} * 4 * (PR2 + 1) * (TMR2 \text{ bölme oranı}) = 1 / f_{osc} * 4 * (PR2 + 1) * (TMR2 \text{ bölme oranı})$



PWM MODU

- PWM dalga seklinde görüldüğü gibi PWM sinyalin bir görev çevrimi - doluluk oranı (duty cycle) bölümü vardır. Bu bölüm PWM sinyalinde lojik-1 (High) olan süredir. Bu görev çevrimi süresi PWM periyodundan uzun olamaz. PWM modülü 10 bit'lik bir çözünürlüğe sahip PWM sinyali üretebilir. Bu 10 bit'lik değerin 8 bit'i kısmı CCPRxL kaydedicisinde, düşük değerlikli 2 bit'i ise CCPxCON kaydedicisinin 5 ve 4. bit'leri ile ifade edilir. Böylece 10 bit'lik değer tanımlanmış olur. PWM sinyalinin görev çevrimi periyodu (süresi) aşağıdaki formül ile hesaplanır. Formüldeki istenen değerler program vasıtası ile değiştirilerek PWM sinyalinin görev çevrimi süresi değiştirilmiş olur.
- PWM sinyalinin görev çevrimi (duty cycle) periyodu;
- $T_{pwm_duty} = (CCPR1L:CCP1CON<5:4>) * T_{osc} * (TMR2 \text{ bölme oranı})$

PWM MODU

- PWM sinyalinin görev çevrimi süresi PWM sinyalinin periyodundan daha büyük olmaması gereklidir. Programlamada bu hususu göz önüne alarak görev çevrimi süresi değeri tespit edilmelidir.
- PWM modülünde oluşturulan PWM sinyalinin bir de çözünürlük değeri vardır. PWM çözünürlüğü, PWM sinyalinin kaç adet eşit parçada değiştirilebileceğini belirtir. PWM çözünürlüğü aşağıda verilen formül ile hesaplanır. PWM çözünürlüğü en az 2, en fazla 10 bit olabilir.

$$\text{PWM Çözünürlüğü} = \frac{\log \frac{F_{\text{OSC}}}{F_{\text{PWM}}}}{\log 2} \text{ bit}$$

CCS Programında kullanılan CCP birimi komutları

- **SETUP_CCPX () FONKSİYONU**

- Bu fonksiyon ile PIC donanımı içinde bulunan (her modelde yoktur) CCP modülü ile ilgili ayarlamalar yapılır.
- `setup_ccp1(mod); , setup_ccp2(mod); , setup_ccp3(mod); ...`
- CCP modülü ister Capture, ister Compare, isterse PWM modunda çalıştırılır. Fonksiyondaki "mod" kısmına aşağıda verilen sabit tanımlamalar yazılır. Bu tanımlamalar yazılırken hangi mod (Capture, Compare, PWM) kullanılacaksa o moda uygun tanımlamalar kullanılır. Fonksiyondaki x kısmına kullanılan CCP modülü birimi numarası yazılır.
- **CCP birimini devre dışı bırakmak için:**
- `CCP_OFF = CCP birimi devre dışı`

CCS Programında kullanılan CCP birimi komutları

- **Capture Mode**
- CCP_CAPTURE_FE
- CCP_CAPTURE_RE
- CCP_CAPTURE_DIV_4
- CCP_CAPTURE_DIV_16

CCS Programında kullanılan CCP birimi komutları

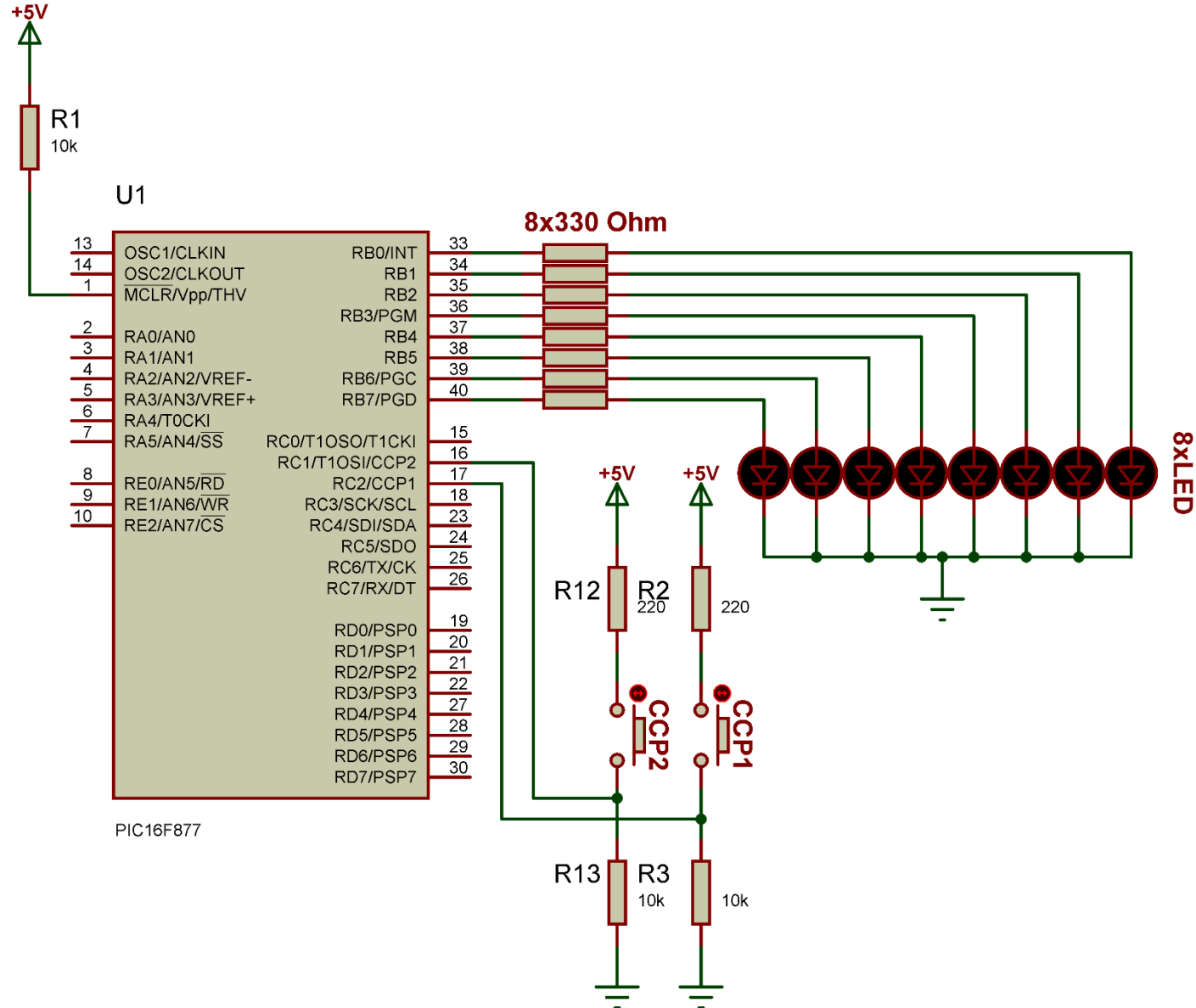
- **Compare Mode**
- CCP_COMPARE_SET_ON_MATCH = Eşleşme olayı meydana geldiğinde CCPx pin'i " 1" olsun, kesme meydana gelsin.
- CCP_COMPARE_CLR_ON_MATCH = Eşleşme olayı meydana geldiğinde CCPx pin'i "0" olsun, kesme meydana gelsin.
- CCP_COMPARE_INT = Eşleşme olayı meydana geldiğinde sadece kesme meydana gelsin, CCPx değişmesin.
- CCP_COMPARE_RESET_TIMER = Eşleşme olayı meydana geldiğinde kesme meydana gelsin, CCPx pin'ini durumu değişmesin ve Timer1 sıfırlansın.
- **PWM Mode**
- CCP_PWM = PWM modunu aç.

CCS Programında kullanılan CCP birimi komutları

- **SET_PWMX_DUTY () Fonksiyonu**

- PWM modunda oluşturulan PWM sinyalinin görev çevrimi süresini belirlemeye yarayan bir fonksiyondur. Fonksiyonda "deger" kısmına 10 bit'lik sabit veya değişken yazılabilir. Bu "deger" kısmı PWM görev çevrimi hesaplama formülünde bulunan (CCPR1L:CCP1CON<5:4> kısmının karşılığıdır. "deger" kısmına 8 bit'lik bir değer girildiğinde derleyici yazılan bu değeri 0 kullanarak kaydırır ve 10 bit'lik değere çevirir. Fonksiyondaki x kısmına kullanılan CCP modülü birimi numarası yazılır.
- `set_pwm1_duty(deger);`, `set_pwm2_duty(deger);`,
- `set_pwm3_duty(deger); ...`
- `set_pwm1_duty(500);` // PWM görev çevrimi süresi 500 olarak ayarlanıyor.

Örnek Programlar (Capture Mod Uyg.)



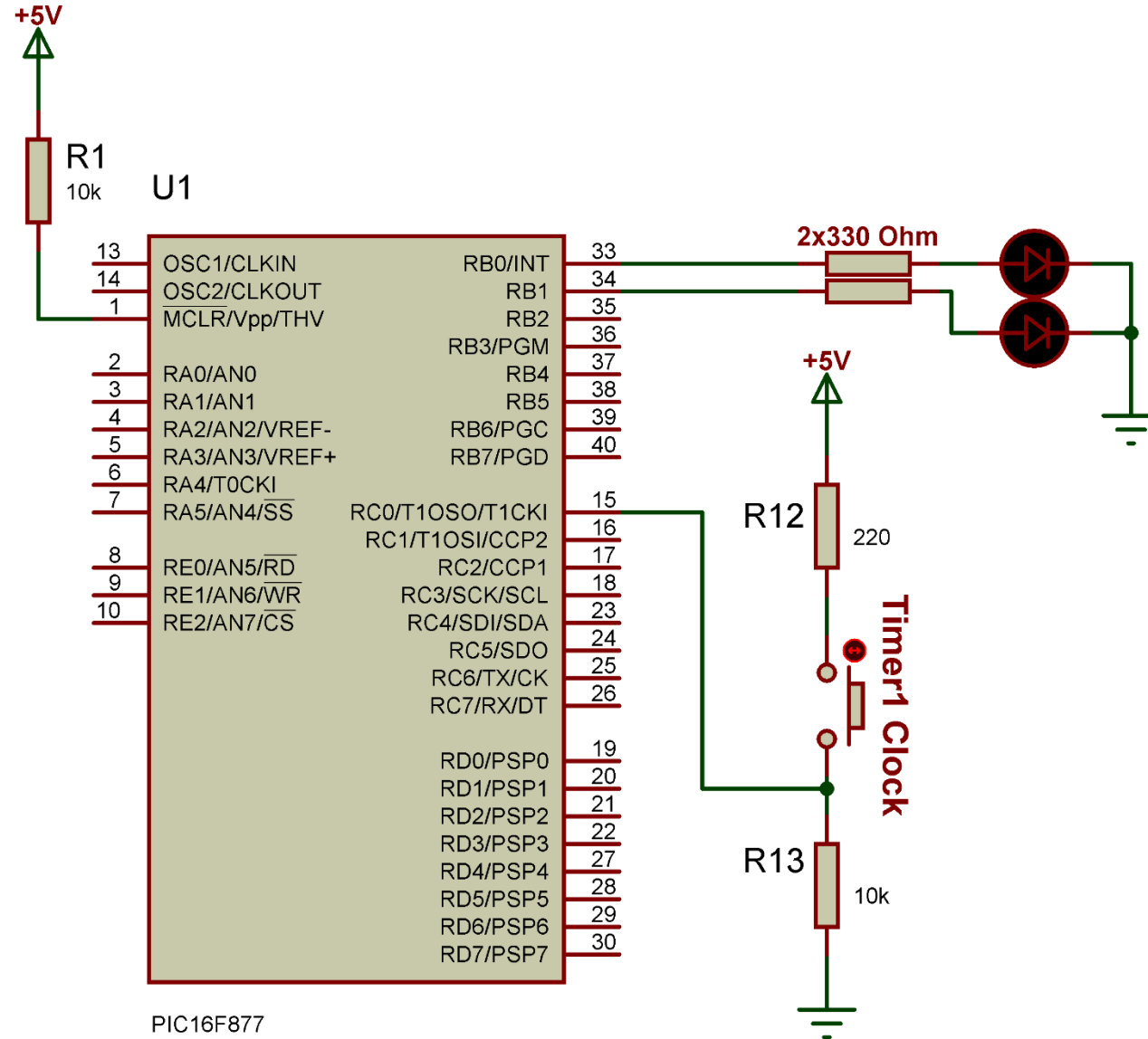
Örnek Programlar (Capture Mod Uyg.)

```
1  #include <16f877.h>
2  #fuses XT,NOWDT
3  #use delay (clock=4000000)
4  #use fast_io(b)
5  #use fast_io(c)
6
7  int i=0; // Tamsayı tipinde değişken tanımlanıyor
8  //***** CCP1 KESMESİ *****
9  #INT_CCP1 // INT_CCP1 kesmesi fonksiyonu
10 void yakala1_kesmesi ()
11 {
12     i++; // i değerini 1 arttır
13     output_b(i); // i değerini B portuna gönder
14 }
15 //***** CCP2 KESMESİ *****
16 #INT_CCP2 // INT_CCP2 kesmesi fonksiyonu
17 void yakala2_kesmesi ()
18 {
19     if(i==0) // Eğer i değeri sıfırsa
20         i=1; // i değeri içeriği 1 olsun
21     i--; // i değerini 1 azalt
22     output_b(i); // i değerini B portuna gönder
23 }
```

Örnek Programlar (Capture Mod Uyg.)

```
24 void main ( )
25 { setup_psp(PSP_DISABLED);          // PSP birimi devre dışı
26   setup_timer_2(T2_DISABLED,0,1);   // T2 zamanlayıcısı devre dışı
27   setup_adc_ports(NO_ANALOGS);      // ANALOG giriş yok
28   setup_adc(ADC_OFF);               // ADC birimi devre dışı
29
30   set_tris_b(0x00);                // B portu komple çıkış
31   set_tris_c(0x07);                // RC0, RC1 ve RC2 pini giriş 00000111
32   output_b(0x00);                  // İlk anda B portu çıkışı sıfırlanıyor
33
34   enable_interrupts(INT_CCP1);      // INT_CCP1 kesmesi aktif yapılıyor
35   enable_interrupts(INT_CCP2);      // INT_CCP2 kesmesi aktif yapılıyor
36   enable_interrupts(GLOBAL);        // Aktif edilen kesmelere izin ver
37   setup_ccp1(CCP_CAPTURE_DIV_4);    // CCP1 kesmesi her 4 yükselen kenarda 1
38   setup_ccp2(CCP_CAPTURE_RE);       // CCP2 kesmesi her yükselen kenarda 1 ya
39
40   CCP_1_HIGH=0x00;                 // CCPR1H kaydedicisi sıfırlanıyor
41   CCP_1_LOW=0x00;                  // CCPR1L kaydedicisi sıfırlanıyor
42
43   CCP_2_HIGH=0x00;                 // CCPR2H kaydedicisi sıfırlanıyor
44   CCP_2_LOW=0x00;                  // CCPR2L kaydedicisi sıfırlanıyor
45   while(1);                        // Sonsuz döngü
46 }
```

Örnek Programlar (Compare Mod Uyg.)



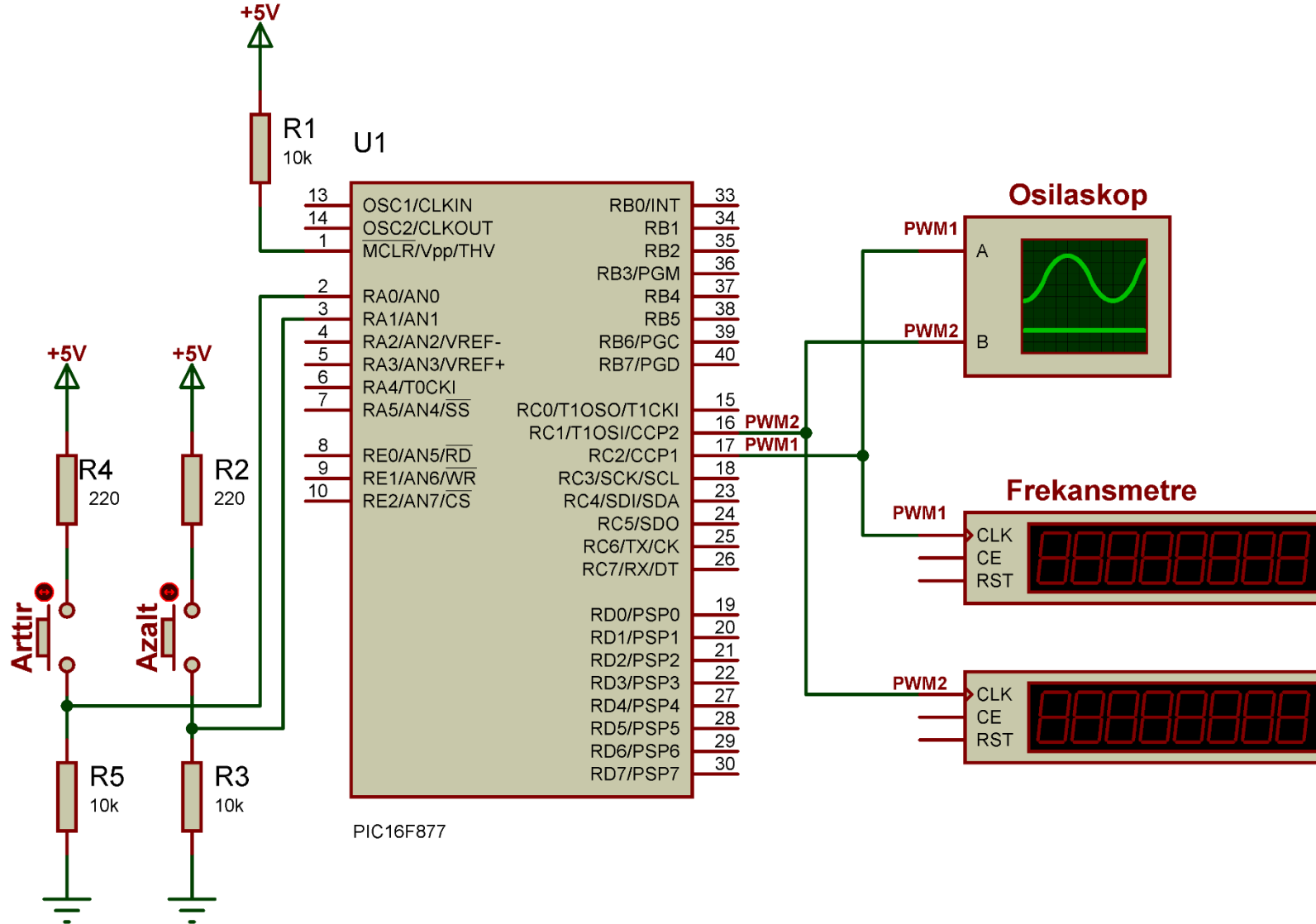
Örnek Programlar (Compare Mod Uyg.)

```
1  #include <16f877.h>
2  #fuses XT,NOWDT
3  #use delay (clock=4000000)
4  #use fast_io(b)
5  #use fast_io(c)
6  //***** CCP1 KESMESİ *****
7  #INT_CCP1    // INT_CCP1 kesmesi fonksiyonu
8  void karsilastir1_kesmesi ()
9  {
10     output_toggle(pin_b0); // RB0 çıkışı tersleniyor
11 }
12 //***** CCP2 KESMESİ *****
13 #INT_CCP2    // INT_CCP2 kesmesi fonksiyonu
14 void karsilastir2_kesmesi ()
15 {
16     set_timer1(0); // TMR1 içeriği sıfırlanıyor
17     output_toggle(pin_b1); // RB1 çıkışı tersleniyor
18 }
19 void main ( )
20 {
21     setup_psp(PSP_DISABLED); // PSP birimi devre dışı
22     setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
23     setup_adc_ports(NO_ANALOGS); // ANALOG giriş yok
24     setup_adc(ADC_OFF); // ADC birimi devre dışı
```


Örnek Programlar (Compare Mod Uyg.)

```
25 set_tris_b(0x00); // B portu komple çıkış
26 set_tris_c(0x01); // RC0 pini giriş
27 output_b(0x00); // İlk anda B portu çıkışı sıfırlanıyor
28
29 enable_interrupts(INT_CCP1); // INT_CCP1 kesmesi aktif yapılıyor
30 enable_interrupts(INT_CCP2); // INT_CCP2 kesmesi aktif yapılıyor
31 enable_interrupts(GLOBAL); // Aktif edilen kesmelere izin ver
32
33 setup_ccp1(CCP_COMPARE_INT); // CCP1 birimi COMPARE modunda kullanılacak
34 setup_ccp2(CCP_COMPARE_INT); // CCP2 birimi COMPARE modunda kullanılacak
35 setup_timer_1(T1_EXTERNAL_SYNC | T1_DIV_BY_1); // Timer1 ayarları yapılıyor
36
37 CCP_1_HIGH=0x00; // CCPR1H kaydedicisi sıfırlanıyor
38 CCP_1_LOW=0x05; // CCPR1L kaydedicisine 05h değeri yükleniyor.
39
40 CCP_2_HIGH=0x00; // CCPR2H kaydedicisi sıfırlanıyor
41 CCP_2_LOW=0x0A; // CCPR2L kaydedicisine 0Ah değeri yükleniyor.
42
43 set_timer1(0); // TMR1 içeriği sıfırlanıyor
44
45 while(1); // Sonsuz döngü
46 }
```


Örnek Programlar (PWM Modu Uyg_1)



Örnek Programlar (PWM Modu Uyg_1)

```
1  #include <16f877.h>
2  #fuses XT, NOWDT
3  #use delay (clock=4000000)
4  #use fast_io(a)
5  #use fast_io(c)
6  int i=5; // Tamsayı tipinde değişken tanımlanıyor
7
8  void main ( )
9  {
10     setup_psp(PSP_DISABLED);           // PSP birimi devre dışı
11     setup_timer_1(T1_DISABLED);        // T1 zamanlayıcısı devre dışı
12     setup_adc_ports(NO_ANALOGS);        // ANALOG giriş yok
13     setup_adc(ADC_OFF);                 // ADC birimi devre dışı
14
15     set_tris_a(0x03); // RA0 ve RA1 pinleri giriş
16     set_tris_c(0x00); // RC0, RC1 ve RC2 pini ÇIKIŞ
17
18     setup_ccp1(CCP_PWM); // CCP1 birimi PWM çıkışı için ayarlandı
19     setup_ccp2(CCP_PWM); // CCP2 birimi PWM çıkışı için ayarlandı
20
21     setup_timer_2(T2_DIV_BY_16, 170, 1); // Timer2 ayarları yapılıyor
22
23     set_pwm1_duty(i); // PWM1 çıkışı görev saykılı belirleniyor
24     set_pwm2_duty(i); // PWM2 çıkışı görev saykılı belirleniyor
```

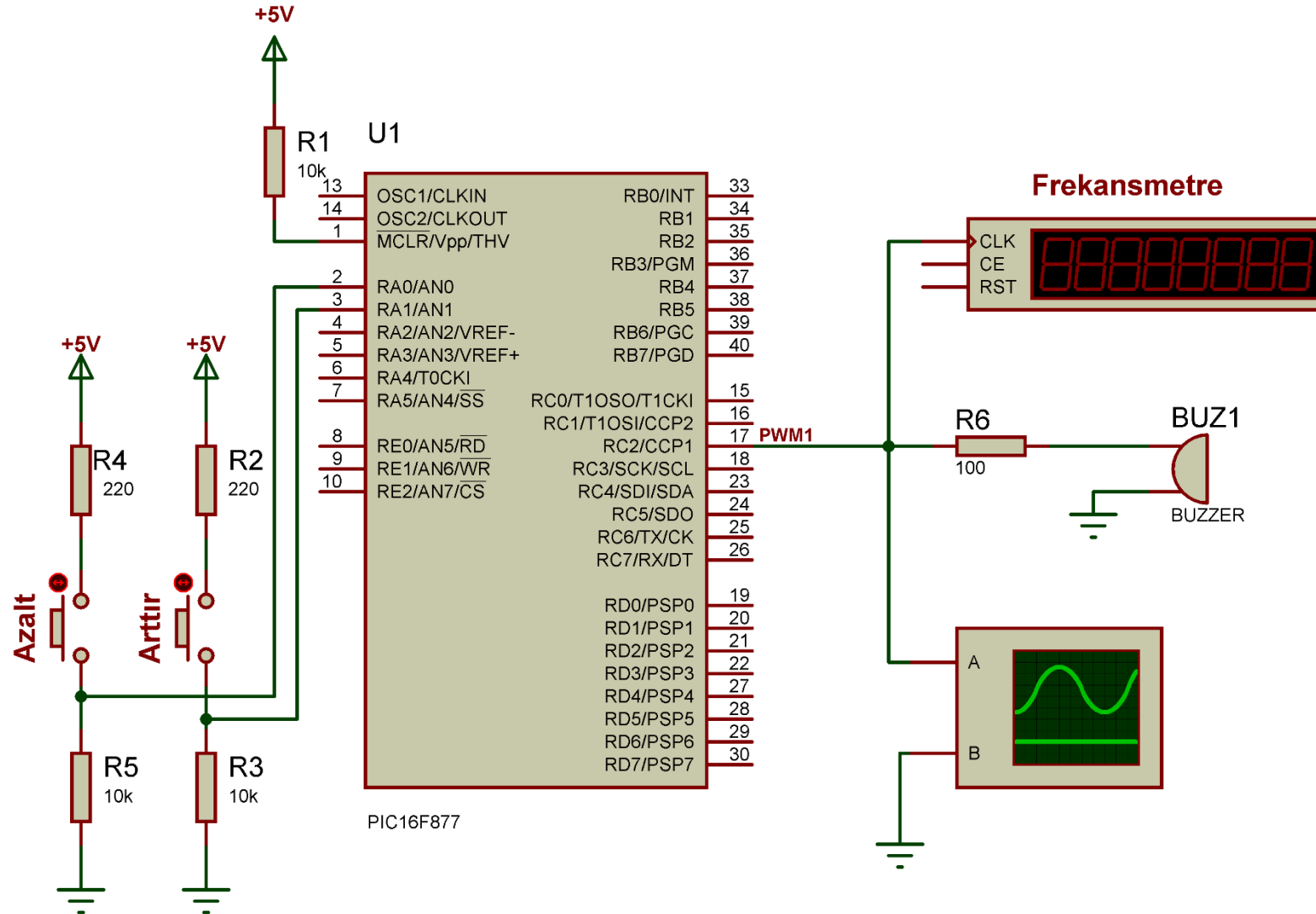
Örnek Programlar (PWM Modu Uyg_1)

```
25 while(1) // Sonsuz döngü
26 {
27     if (input(pin_a0)) // Eğer RA0 girişine bağlı butona basılırsa
28     {
29         delay_ms(20); // Buton arkı önleme gecikmesi
30         while(input(pin_a0)); // Basılı butondan el çekile kadar bekle
31         i+=3; // i=i+3 anlamındadır.
32
33         if (i>=170) // i değeri 170'dan büyükse i değeri 170 olsun
34             i=170;
35
36         set_pwm1_duty(i); // PWM1 çıkışı görev saykılı belirleniyor
37         set_pwm2_duty(i); // PWM2 çıkışı görev saykılı belirleniyor
38     }
39
```

Örnek Programlar (PWM Modu Uyg_1)

```
40 if (input(pin_a1)) // Eğer RA1 girişine bağlı butona basılırsa
41 {
42     delay_ms(20); // Buton arkı önleme gecikmesi
43     while(input(pin_a1)); // Basılı butondan el çekile kadar bekle
44     i-=3; // i=i-3 anlamındadır.
45
46     if (i<5) // i değeri 5'den küçük ise i değeri 10 olsun
47         i=5;
48
49     set_pwm1_duty(i); // PWM1 çıkışı görev saykılı belirleniyor
50     set_pwm2_duty(i); // PWM2 çıkışı görev saykılı belirleniyor
51 }
52 }
53 }
54 }
```

Örnek Programlar (PWM Modu Uyg_2)



Örnek Programlar (PWM Modu Uyg_2)

```
1  #include <16f877.h>
2  #fuses XT,NOWDT
3  #use delay (clock=4000000)
4  #use fast_io(a)
5  #use fast_io(c)
6  int i=65; // Tamsayı tipinde değişken tanımlanıyor
7
8  void main ( )
9  {
10     setup_psp(PSP_DISABLED);           // PSP birimi devre dışı
11     setup_timer_1(T1_DISABLED);        // T1 zamanlayıcısı devre dışı
12     setup_adc_ports(NO_ANALOGS);        // ANALOG giriş yok
13     setup_adc(ADC_OFF);                 // ADC birimi devre dışı
14
15     set_tris_a(0x03); // RA0 ve RA1 pinleri giriş
16     set_tris_c(0x00); // C portu komple çıkış
17
18     setup_ccp1(CCP_PWM); // CCP1 birimi PWM çıkışı için ayarlandı
19     setup_timer_2(T2_DIV_BY_16,i,1); // Timer2 ayarları yapılıyor
20
21     set_pwm1_duty(50); // PWM1 çıkışı görev saykılı belirleniyor
22
23     while(1) // Sonsuz döngü
```

Örnek Programlar (PWM Modu Uyg_2)

```
22 while(1) // Sonsuz döngü
23 {
24     if (input(pin_a0)) // Eğer RA1 girişine bağlı butona basılırsa
25     { delay_ms(20); // Buton arkı önleme gecikmesi
26         while(input(pin_a0)); // Basılı butondan el çekile kadar bekle
27         i+=5; // i=i+5 anlamındadır.
28
29         if (i>250) // i değeri 250'dan büyükse i değeri 250 olsun
30             i=250;
31         setup_timer_2(T2_DIV_BY_16,i,1); // Timer2 ayarları yapılıyor
32     }
33     if (input(pin_a1)) // Eğer RA0 girişine bağlı butona basılırsa
34     { delay_ms(20); // Buton arkı önleme gecikmesi
35         while(input(pin_a1)); // Basılı butondan el çekile kadar bekle
36         i-=5; // i=i-5 anlamındadır.
37
38         if (i<65) // i değeri 65'den küçük ise i değeri 65 olsun
39             i=65;
40         setup_timer_2(T2_DIV_BY_16,i,1); // Timer2 ayarları yapılıyor
41     }
42 }
43 }
```

Kaynaklar

- CCS C Programlama Kitabı, Serdar Çiçek, Altaş Yayıncılık
- Mikroelektronika C programlama e-kitabı «<https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c>»