



PIC16F876-PIC16F877 PWM MODÜLÜNÜN KULLANIMI

Emre YAVUZ - Ağustos 2009

PIC16F87X'te PWM MODÜLÜNÜN KURULUMU

Bu makalemizde PIC16F876-PIC16F877 mikrodnetleyicilerinde PWM modülünün kullanımını anlatacağım.

PWM, (Pulse-Width Modulation), yani darbe genişliği modülasyonu, sabit çıkış verebilen entegrelere görev süresi istenilen değere göre ayarlı bir kare dalga verdirerek analog çıkış elde etme olayıdır.

Örneğin mikrodnetleyicilerden biz 5 volt veya 0 voltluk çıkışlar alabiliyoruz. fakat 4 voltluk bir çıkışa ihtiyacımız var diyelim, bunu transistör vb. ek devre kullanarak elde edebiliriz yada mikrodnetleyiciden 4 birimi 5 volt, 1 birimi de 0 volt olan frekansı çok düşük olmayan bir kare dalga verdirirsek bu pratikte 4 volt olarak algılanır ve bizi ek bir donanımdan kurtarmış olur. Bu 5 volt olan kısım dalganın görev süresidir. Kısacası, PWM ile, bu görev süresini, yani darbenin genişliğini ayarlayarak çıkışın analog değerini ayarlamış oluruz.

Bunu program içerisinde kendi PWM modülümüzü yazıp(gecikmeler kullanıp görev süresini ayarlayarak) yapabileceğimiz gibi, hazır PWM modülüne sahip mikrodnetleyicilerin bu modüllerinden faydalanarak da yapabiliriz.

PIC16F87X serisi mikrodnetleyiciler iki adet PWM modülüne sahiptir. Bunlar Portc'nin birinci ve ikinci bacaklarındadırlar. Bu mikrodnetleyicilerin PWM modüllerini kullanabilmek için çeşitli ayarları yapmak gerekir.

Aşağıdaki komutlar bu kurulumda gerekli olan genel komutlardır.

```
BANKSEL TRISC ;portc'yi çıkış olarak ayarlıyoruz
CLRF TRISC    ;sadece pwm çıkışı alacağımız rc1 ve rc2'yi de
                ;çıkış olarak ayarlayabilirdik
```

```
BANKSEL CCP1CON
```

```
; RC2 birinci PWM çıkışıdır
; RC2'den PWM çıkış almak için CCP1CON'u 2. ve 3. bitlerini
; 1 yapmamız gerek 0. ve 1. bitlerinin ne olduğu önemli değil
```

```
BSF CCP1CON,2
```

```
BSF CCP1CON,3
```

```
; RC1'den PWM çıkışı almak için CCP2CON'a da aynı işlemi uyguluyoruz
```

```
BSF CCP2CON,2
```

```
BSF CCP2CON,3
```

```
; Daha sonra PR2 registerına değer atıyoruz
; bu değer dalganın frekansını belirlemeye yarıyor.
```

```
BANKSEL PR2
```

```
MOVLW 0xFF ; 255 değeri atadık
```

```
MOVWF PR2 ; bu değeri gerektiği şekilde ayarlamamız gerek.
```

; Timerın prescaler değerini T2CON registerının 0. ve 1. bitleriyle ayarlıyoruz
 ; 16 değeri için 11, 4 için 01, 1 için de 00 değerlerini 1. ve 0. bitlere yazıyoruz
 ; prescaler değeri de dalga frekansına etki ediyor

BANKSEL T2CON

BSF T2CON,0

BSF T2CON,1 ; prescaler değerini 16 olarak ayarladık

; daha sonra aynı registerın 6-3 bitleriyle oynayarak
 ; postscaler değerini ayarlıyoruz bu değer verdiğimiz çıkışla ilgili frekans vb.
 ; gibi bir işe yaramıyor

MOVF T2CON,W

ANDLW 0x07

IORLW 0x00

MOVWF T2CON

CLRF CCPR1L

CLRF CCPR2L

; çıkışı başlatmak için TMR2ON bitine bir değeri atamamız gerekiyor

BSF T2CON, TMR2ON

; daha sonra CCPR1L ve CCPR2L registerlarına ve
 ; CCP1CON ve CCP2CON registerlarına değerler atayarak
 ; PWM çıkışlarını duty cyclelerini ayarlıyoruz

MOVLW D'30'

MOVWF CCPR1L ; PORTC,2 deki modülü kontrol eder

MOVLW D'40'

MOVWF CCPR2L ; PORTC,1 deki modülü kontrol eder

Dalgaın frekansı

$PWM \text{ periyodu} = [(PR2) + 1] \times 4 \times TOSC \times (TMR2 \text{ prescaler değeri})$

formülünden hesaplanabilir, yani Osilatör frekansını 20 MHZ kabul edersek yukarıdaki değerlerle (PR2=h'FF', TMR2 nin prescaleri 16)

$PWM \text{ Periyodu} = [255 + 1] \times 4 \times (1/20000000) \times 16$

PWM periyodunu 81.9 mili saniye frekansını ise

$f=1/T$ formülünden 1.22 KHZ olarak hesaplarız.

Görev süresi ise ;

$PWM \text{ Duty Cycle} = (CCPR1L:CCP1CON<5:4> \cdot TOSC \cdot (TMR2 \text{ prescaler değeri})$

formülü kullanılarak hesaplanır.

Yani yukarıdaki gibi CCPR1L registerına 30 değeri vermişsek, görev süresi;
 $DC = 120 \times (1/20000000) \times 16 = 96$ mikro saniye olarak bulunur.

Burada 120 değerini, PWM çıkışının CCPR1L registerı ve CCP1CON<5:4> bitlerinin birleşiminden oluşan 10 bitlik bir değere göre hareket ettiğini göz önüne alarak hesapladık.
Yani

CCPR1L'ye 30 değeri atadık yani binary olarak b'00011110' CCPCON<5:4> de b'00' olduğu için bu 10 bitlik çıkışımız b'0001111000' şeklini aldı bunun da decimal karşılığı 120'dir. Bunun da analog değeri yaklaşık olarak $(120/1023) \times 5 \text{ volt} = 0.6$ 'dır. Ama diyelim ki PIC'ın çıkışı bir motor sürücü entegresinin inputuna bağlı o zaman hesabımızı ona göre yapmamız gerekir, yani motora gelen değer yaklaşık olarak $(120/1023) \times 12 \text{ volt} = 1.4$ volt olacaktır.

Yani istediğiniz çıkışı elde etmek için bu 10 bitin tamamını göze alarak hesabınızı yapmanız gerekir.

Özetlemek gerekirse; öncelikle PWM modüllerinin kurulum ayarlarını yapıyoruz. Daha sonra frekans değerini ihtiyacımıza göre ayarlıyoruz. Örneğin bir motor sürmek istiyorsak frekansımız orta bir değerde olmalı, bu değerleri deneme yanılma bulmanız gerekir. Son olarakta görev süresini istediğimiz şekilde ayarlayarak analog çıkış elde ediyoruz..

Kaynaklar

PIC16f87X datasheet

Bağlantılar

<http://robot.ee.hacettepe.edu.tr/>