

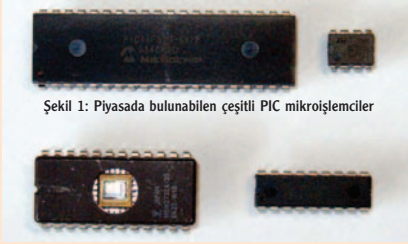


# Kendimiz Yapalım

Ömer Çayırpunar - ODTÜ Robot Topluluğu

## Mikroişlemcilerin Çevre Birimlerinin Kullanılması

Robotların beyinleri olan ve önceden programlanmış yönetim ve karar mekanizmalarını içeren mikroişlemciler birçok farklı birimden oluşmaktadır. Bu birimler aritmetik mantık birimleri(ALU), bellek, giriş çıkış portları ve yazmaçlar ile birlikte çeşitli alt çevresel birimlerdir. Bu çevresel birimlerden bazıları zamanlayıcılar, kesme mekanizmaları, analog-dijital çeviriciler (ADC), PWM ve seri haberleşme birimleri(UART) olabilmektedir. Bu ayki yazımızda mikroişlemcilerin sahip olduğu önemli çevre birimlerinin kullanılması ve bu birimlerle neler yapılabileceğimiz konusunda çeşitli uygulamalarımız olacak. Sizlerle zamanlayıcı alt birimini kullanarak bir saat ve analog-dijital çevirici birimini kullanarak ta bir pil test cihazı uygulaması yapacağız.

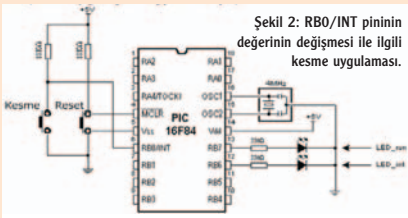


Şekil 1: Piyasada bulunabilen çeşitli PIC mikroişlemciler

### Kesme Çevre Biriminin kullanılması:

Kesmeler, mikroişlemcilerin ani cevap vermesi gereken bazı özel durumlarda kullanılan mekanizmalardır. Bu durumlar mikroişlemcinin RBO/INT pininin değerinin değişmesi, seri porttan bir bilgi alınması, yada zamanlayıcılardan birisinin taşmış (overflow) olması olabilir. Kullanılan mikroişlemci türüne ve büyüklüğüne bağlı olarak kesme durumlarının sayısı ve çeşidi değişebilir. Herhangi bir kesme olmadığın zaman bir mikroişlemci normal olarak kendi ana programı üzerinde sürekli bir döngü halinde çalışır ve hiçbir duraksama olmaz. Bir kesme oluştuğunda ise mikroişlemci kesmeyi fark eder, ana programın çalışmasını durdurur ve o kesmeye özel alt programı (interrupt routine) çalıştırmaya başlar. Bu alt programda oluşan kesme durumu analiz edilir ve bu durum karşısında mikroişlemcinin nasıl bir tepki vermesi gerektiği tanımlanır. Kesmenin analiz edilmesi gereklidir çünkü birden fazla kesme türü bulunduğu için mikroişlemcimizin bunlardan hangisine tepki vermesi gerektiği önceden belirtilmelidir. Kesme alt programı tamamlandıktan sonra mikroişlemci tekrar ana program döngüsünde kaldığı yerden devam etmeye başlar.

Şekil 2'deki devre şemasında mikroişlemcinin RBO/INT pininin değerinin değişmesi ile tetiklenen



Şekil 2: RBO/INT pininin değerinin değişmesi ile ilgili kesme uygulaması.

kesme mekanizması kullanılmıştır. Kesme adlı buton mikroişlemcinin RBO/INT pinine, LED\_RUN adlı LED PORTB.7 pinine, LED\_KESME adlı LED ise PORTB.6 pinine bağlanmıştır. Kesme olmadığında sadece LED\_RUN yanık durumda, LED\_KESME ise sönmük durumdadır. Kesme butonuna basıldığında ise bir kesme oluşur ve mikroişlemci kesmenin tanımlandığı kısımdaki kodu çalıştırmaya başlar. Bu kısımda ise bu sefer LED\_RUN söndürülür ve LED\_KESME yanmaya başlar. Mikroişlemciye yüklenecek PICBASIC-PRO(PBP) kodu ise şu şekildedir.

```

:*****KESME.BAS*****
LED_RUN VAR PORTB.7
LED_KESME VAR PORTB.6
TRISB = %00111111
OPTION_REG = %10000000
On Interrupt Goto KESME
INTCON = %10010000
PORTB = 0
ANADONGU:
HIGH LED_RUN
LOW LED_KESME
GOTO ANADONGU
DISABLE
KESME:
LOW LED_RUN
HIGH LED_KESME
PAUSE 500
INTCON.1 = 0
RESUME
ENABLE
END

```

### Zamanlayıcı Çevre Biriminin kullanılması:

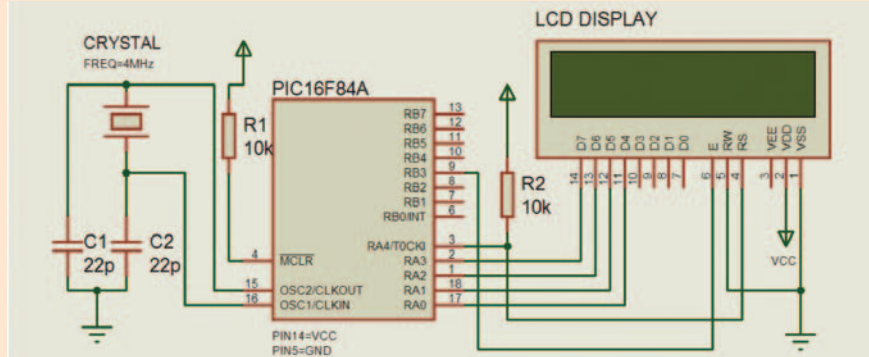
PIC mikroişlemcilerin tümünde TIMERO zamanlayıcısı bulunmaktadır. Bu zamanlayıcı bizden bağımsız olarak geri planda sürekli çalışır. Zamanlayıcı 8 bitlidir. Yani maksimum alabileceği değer 255 tir. 4 MHz lik bir osilatörün kullanıldığını varsayarsak TIMERO her 1 mikrosaniyede bir artar ve 256 mikrosaniye de bir başa döner. Bu zamanlayıcı süre aşımına uğradığında INTCON yazmacının 2.biti "1" olur. Eğer TMRO kesmesi etkinleştirilmiş ise bu bir kesme oluşturur. TIMERO'ın en güzel olan yanı ölçeklendirilebilmesidir. Yani zamanlayıcının artış aralıkları değiştirilerek kesme süresi uzatılabilir yada kısaltılabilir. TIMERO'ı 2,8,16,32,64,128 ve 256 ile ölçeklemek olanaklıdır. Örnek olarak eğer biz TIMERO'ı 256 ile ölçeklersek, 256x256 yani 65.535 mikrosaniyede bir kesme gerçekleşecektir. TIMERO'ın ölçeklendirilmesi OPTION\_REG yazmacıyla yapılmaktadır.

```

*****SAAT.BAS*****
PAUSE 250
SAAT VAR BYTE
DSAAT VAR BYTE
DAKİKA VAR BYTE
SANİYE VAR BYTE
TICKS VAR BYTE
UPDATE VAR BYTE
SAAT=0; DAKİKA=0; SANİYE=0; TICKS=0; UPDATE=1
OPTION_REG = %00000101 ;TIMERO ÖLÇEKLEME = 1/64
INTCON = $A0 ;TIMERO KESMESİ ETKİN
ON INTERRUPT GOTO KESME ;KESME ADRESİ
ANADONGU:
IF UPDATE = 1 THEN
LCDOUT $FE,1 ;EKRANI TEMİZLE
DSAAT = SAAT ;SAATİ 12'LİK SİSTEME ÇEVİR
IF (SAAT // 12) = 0 THEN
DSAAT = DSAAT + 12
ENDIF
IF SAAT < 12 THEN ;DEĞERLERİ LCD EKRAINA YAZ
LCDOUT DEC2 DSAAT, ":", DEC2 DAKİKA, ":", DEC2 SANİYE, " AM"
ELSE
LCDOUT DEC2 (DSAAT - 12), ":", DEC2 DAKİKA, ":", DEC2 SANİYE, " PM"
ENDIF
UPDATE = 0
ENDIF
GOTO ANADONGU:
DISABLE
KESME:
TICKS = TICKS + 1
IF TICKS < 61 THEN DEVAM ;TICK LER 61 OLURSA SANİYİYİ ARTIR
TICKS = 0
SANİYE = SANİYE + 1 ;SANİYELER 60 OLURSA DAKİKAYI ARTIR
IF SANİYE >= 60 THEN
SANİYE = 0
DAKİKA = DAKİKA + 1
IF DAKİKA >= 60 THEN ;DAKİKALAR 60 OLURSA SAATİ ARTIR
SAAT = SAAT + 1
IF SAAT >= 24 THEN SAAT = 0
ENDIF
ENDIF
UPDATE = 1 ;GÜNCELLEME TAMAMLANDI
DEVAM:
INTCON.2 = 0 ;KESME BAVRAĞINI TEMİZLE
RESUME
END

```

TIMERO zamanlayıcısını ve TMRO kesmesini kullanarak kolaylıkla dijital göstergeli bir saat yapabiliriz. Şekil 3 teki devre şemasını kullanarak yapacağımız devrede eğer mikroişlemcimize aşağıdaki SAAT.BAS başlıklı PICBASIC kodunu yüklersek saatimiz çalışmaya hazır olacaktır. Bu kod TIMERO sayacını her 16.384 milisaniyede bir süre aşımına uğrayacak şekilde yapılandırır ve süre aşımında da kesme rutinini çalıştırır. Kesme rutini ise her 61 çalışmasında saniyeyi bir artırır (61x16384 yaklaşık 1 saniyeye eşittir). Artan saniyeler 60 olduğunda dakika değişkeni, artan dakikalar 60 olduğunda da saat değişkeni bir artar. Daha sonra bu değişkenler uygun formatta LCD display ekranına yazılır.



Şekil 3: LCD göstergeli dijital saat devre şeması.

## Analog-dijital Çevirici Çevre biriminin Kullanılması:

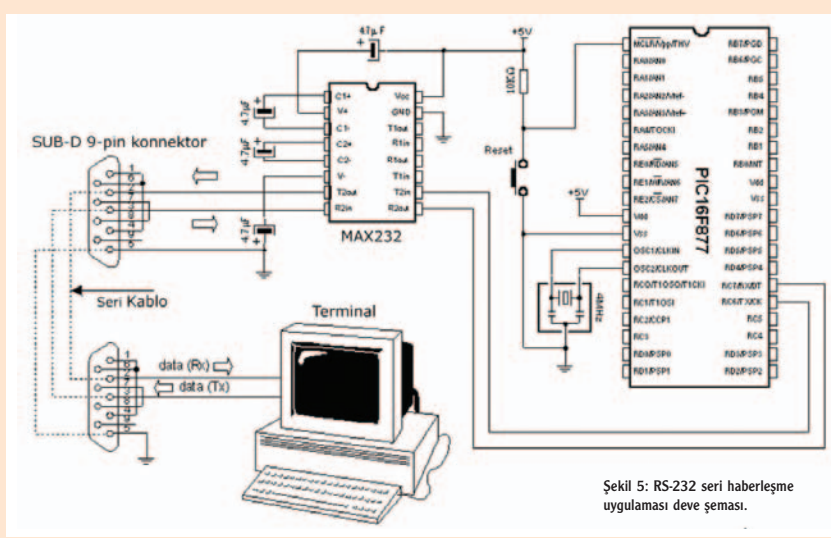
Analog-dijital çevirici alt birimi birçok mikroişlemcide bulunan alt birimlerden biridir. Bu birimi kullanarak mikroişlemcimizin dış dünyadaki analog sinyalleri anlamasını sağlayabiliriz. Çoğu mikroişlemci 10 bitlik çözünürlükle analogdan dijitalce çevirim yapabilmektedir. Bu ise eğer 5V'luk bir gerilim kaynağı kullanırsanız, 4.88V'luk bir hassasiyete eşittir.

Bu uygulamada kullanacağımız PIC16F877 mikroişlemcisinde 8 ayrı analog kanal vardır. Başka bir deyişle 8 farklı analog sinyali birden takip edebilirsiniz. Örneğin kanallardan birine sıcaklık sensörü, diğerine basınç sensörü, bir diğerineyse nem sensörü bağlayarak küçük bir meteoroloji istasyonu kurabilirsiniz. Yapabileceğiniz uygulamalar sizin hayal gücünüze bağlı. Fakat biz sadece tek bir kanal kullanarak basit bir kalem pil test cihazı uygulaması yapacağız. Bu uygulamada Şekil 4'teki devre şemasını kurmanız ve gerekli PBP kodunu mikroişlemcimize yüklememiz gerekiyor

```
*****ANALOG.BAS *****
DEFINE ADC_BITS 10      ;ADC ÇÖZÜNÜRLÜĞÜ 10 BIT
DEFINE ADC_CLOCK 0     ;ADC CLOCK = OSCILLATOR / 2
DEFINE LCD_DREG PORTC ;LCD TANIMLANMASI
DEFINE LCD_DBIT 0
DEFINE LCD_RSREG PORTC
DEFINE LCD_RSBIT 4
DEFINE LCD_EREG PORTC
DEFINE LCD_EBIT 5
ADC_DEGERI VAR WORD    ;GEREKLİ DEĞİŞKENLER
GERILIM VAR WORD
TRISA = %11111111      ; PORTA'NIN TAMAMI GİRİŞ
ADCON1 = %10000010    ;ADC AÇIK
ANADONGU:
  ADCIN 0, ADC_DEGERI ;PORTA.0 DAN GERİLİM
  DEĞERİNİ OKU
GERILIM = ADC_DEGERI*48/10 ; MİLVOLT'A ÇEVİR
LCDOUT $FE,1, #GERILIM, "mV " ;LCD EKCRANINA YAZ
IF GERILIM < 1500 THEN ;ALT SATIRA GEÇ
LCDOUT $FE,$C0, "BOS" ;DEĞER 1500 mV TAN
  KÜÇÜKSE "BOS"
ELSE
  LCDOUT $FE,$C0, "DOLU" ;BÜYÜKSE "DOLU" YAZ
ENDIF
PAUSE 250 ;ÖLÇÜMLER ARASINDAKİ
  BEKLEME
GOTO ANADONGU
```

## Seri Haberleşme Çevre Biriminin Kullanılması:

İki mikroşlemci arasında yada bir mikroşlemci ile bir bilgisayar arasında data transferi yapmanın en kolay yolu RS-232 arayüzünü kullanmaktır. RS-232 arayüzü sadece iki adet iletken kablo kullanarak 10 metre mesafe içerisinde seri asenkron veri transferini mümkün kılar. Eğer kullanacağımız mikroşlemci



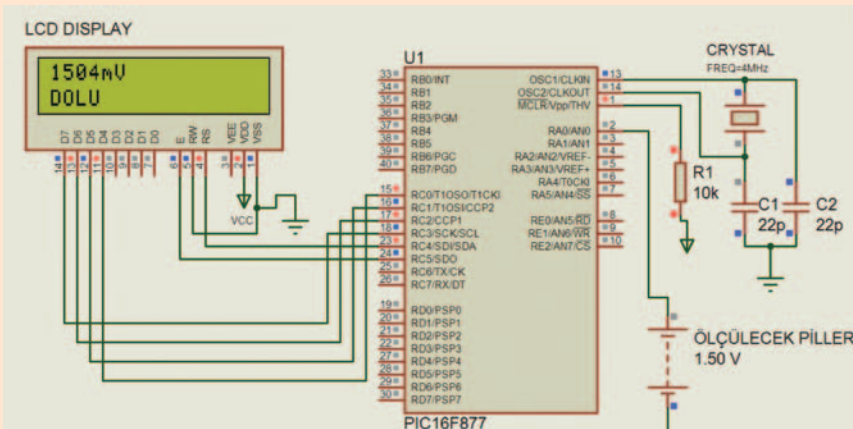
Şekil 5: RS-232 seri haberleşme uygulaması devre şeması.

seri haberleşme çevre birimini(UART) içerisinde bulunduyorsa RS-232 arayüzünü kolaylıkla kullanabiliriz.

Yapacağımız uygulamada kullanacağımız mikroşlemci olan PIC16F877 UART birimine sahiptir. Bu arabirim aynı TIMER0'da olduğu gibi geri planda seri veri alışverişini yaparken, PIC ana program üzerinde çalışmaya devam eder. Bizim bütün yapmamız gereken, hazırlayacağımız program senaryosunda belli aralıklarla uygun PBP komutlarını (HSERIN ve HSEROUT) çalıştırmaktır. Şekil 5'teki devre şemasında bir bilgisayar ile PIC16F877 mikroşlemcisi arasında kurulmuş bir seri haberleşme uygulaması gösterilmiştir. Mikroşlemcinin seri haberleşme giriş-çıkışları RS-232 seviyesinde olmadığı için bunları direkt olarak bilgisayarı seri portuna bağlayamayız. Mikroşlemci çıkışlarını bilgisayara bağlamadan önce uygun RS-232 gerilim seviyelerine dönüştürmemiz gerekir. Bu sebepten dolayı bu uygulamada MAX232 adlı RS-232 seviye çevirici entegresi kullanılmıştır. Bu entegre mikroşlemcinin çıkış seviyesi olan TTL'den RS-232'ye ve RS-232'den de TTL'ye gerilim dönüştürmelerini yapabilmektedir.

Seri haberleşme uygulamamızda mikroşlemcimize yüklenecek olan program UART.BAS adlı PBP kodudur. Bu programda seri porttan bilgi almak için HSERIN, seri porta bilgi göndermek için ise HSERO-UT komutları kullanılmıştır. Programımızın yaptığı

```
*****UART.BAS *****
BILGI VAR BYTE
TRISC = %10111111 ;:PORTC GİRİŞ ÇIKIŞLARI
  AYARLANIYOR
SPBRG = 25 ;:BAUD RATE 2400 Bps
RCSTA = %10010000 ;:SERI PORT BİLGİ ALIMI AKTİF
TXSTA = %00100000 ;:SERI PORT BİLGİ GÖNDERİMİ
  AKTİF
ANADONGU:
  HSERIN [WAIT("X"), BILGI];X DEĞERİNİ BEKLE,
  ;X TEN SONRA GELEN BİLGİYİ
  ;SAKLA
  HSEROUT [DEC BILGI] ;:BU BİLGİYİ ASCII
  FORMATINDA TEKRAR GÖNDER
  ;DENEMELER ARASINDAKİ
  BEKLEME
PAUSE 200
GOTO ANADONGU
```



Şekil 4: PİL test cihazı devre şeması.

işe gelince; programımız seri porttan gelen bilgiyi sürekli dinlemekte, eğer gelen bilgi "X" ise bunun ardından gelen bilgiyi "BILGI" adıyla kaydedip bilgisayara seri port üzerinden tekrar göndermektedir. PC tarafında ise veri göndermek ve almak için "Hyperterminal" programını kullanabiliriz. Bu uygulama için hyperterminal oturumunu COM1 üzerinden, 2400 bps(bits per second) hızında, 8 veri biti ve 1 dur biti şeklinde açmalıyız. Açılan terminal ekranında göndermek istediğiniz veriyi klavyeden tuşlayarak girebilirsiniz.

## PWM Çevre Birimi Kullanılması:

Mikroşlemcilerde bulunan bir diğer çevresel birim PWM (Pulse Width Modulation) yada darbe genişlik modülasyonu birimidir. Bu birim sayesinde dijital sinyaller analog sinyallere dönüştürülebilmektedir. PWM birimine sahip bir mikroşlemci PWM sinyalini üretebilmek için belirli bir pini kullanır. Örnek olarak, bu pin PIC 16F628'te PORTB.3 tür. PWM birimi çalıştırıldığında mikroşlemcinin ilgili pininden, önceden belirlenen bir frekansta bir kare dalga sinyali gönderilir. Bu sinyalin +5 V (iş yapma aralığı) ta ve 0 V (bekleme aralığı) ta kalma süreleri değiştirilerek ilgili pindeki gerilim de değiştirilmiş olur. Eğer iş yapma aralığı ve bekleme aralığı birbirine eşitse çıkış ana geriliminin %50 si olacaktır. Başka deyişle de %50'lik bir "Duty Cycle" üretilmiş olur. PBP'da PWM birimini kullanarak sinyal üretmek için kullanılan komut "HPWM" komutudur. Kullanılışı ise "HPWM kanal,Dutycycle,Frekans" şeklindedir. Burada kanal PIC'in üzerinde ki birinci yada ikinci PWM kanalı olabilir. Dutycycle çıkış sinyalinin gerilimidir. 8 bitliktir ve değeri 0 ile 255 arasında değıştikçe çıkış gerilimi de 0 V ile +5 V arasında orantılı olarak değışir. PWM birimini kullanarak ve uygun elektronik devrelerle PWM çıkışını güçlendirerek rahatlıkla bir lambamın parlaklığını değıştirilmiş yada bir elektrik motorunun dönüş hızını ayarlayabilirsiniz.

Aşağıdaki kod mikroşlemcinin birinci PWM kanalından, 2000 Hz frekansında ve 127/255 = %50 lik dutycycle da bir sinyal üretir. Çıkış gerilimi 5/2 = 2.5 Volt olur.

HPWM 1,127,2000

Yardım ve destek için <http://robot.metu.edu.tr/forum> adresi altındaki foruma ilet yazabilirsiniz.

**Dizeltme:** Aralık ayındaki yazımızda devre şemasında bulunan R14 adlı 4 Kohm'luk direncin değerini yanlış yazıldığını ve 1 Kohm olması gerektiğini anlaşılmıştır.

## Kaynaklar:

Odtü Robot Topluluğu sitesi: <http://www.robot.metu.edu.tr>  
Microchip, 16F84A Sata Sheet: <http://www.microchip.com>  
Microchip, 16F628A Sata Sheet: <http://www.microchip.com>  
Microchip, 16F87X Sata Sheet: <http://www.microchip.com>  
Hyperterminal Programı <http://www.hilgraeve.com>  
Yaşar Bodur, Adım Adım PICmicro Programlama, INFOGATE Yayınları, 2002  
Yaşar Bodur, PICBasic Pro ile PICmicro Programlama, INFOGATE Yayınları, 2002