

olarak nitelendirilmekte bunun dışındakiler ise Analog cihazlar olarak nitelendirilmektedirler. Mikroişlemcilerin tamamı dijital cihazlardır.

Zaman zaman mikroişlemci bazlı bazı cihazların Analog cihazları kontrol etmesi istenir. Zaman zamanda Analog sinyallerin işlenip değerlendirilmesi için bu sinyallerin dijital sinyaller haline getirilmesi gerekir.

İşte bu gibi durumlarda Analog – Digital çeviriciler kullanılır. Yalnızca A/D çevirici olarak üretilmiş entegreler olduğu gibi bazı Pic'ler de bu çeviriciler seçimli olarak hazır bulunurlar. Burada **A/D** çeviricisi bulunan Pic'ler incelenecek ve kullanımları konusunda örnekler verilecektir.

Bu konuda en çok kullanılan Pic'lerden birisi **PIC16F877** entegresidir. 40 bacaklı olan bu entegrede 8 kanal ve 10 bit hassasiyette **A/D** çeviricisi bulunmaktadır.

A/D çevirim prensibi genelde bir kondansatörün belirli bir referans voltajı ile şarj edilipdeşarj için geçen sürenin sayılması esasına dayanır. Elde edilen süre bilgisi ise Analog değere karşılık elde edilmiş olan dijital değer olarak verilir.

Üzerinde **A/D** çevirici bulunan bir Pic kullanılırken yapılacak işlerin başında hangi bacakların ANALOG hangilerinin **DIGITAL** olarak kullanılacağına karar vermektir. Sonra A/D çevirici için referans voltajı kaynağının seçimi yapılmalıdır. İki türlü referans voltajı vardır. Birincisi Pic'in kendi VDD beslemesi (+5V) diğeri ise harici bir voltaj kaynağıdır. Tabiiki harici kaynaklar referans voltajının 5 Volt'tan farklı olması durumunda seçilmelidir.

Şimdi gelelim bu seçimleri yaptıktan sonra bunları Pic'e nasıl vereceğimize. A/D çeviricisi olan Pic'lerde **ADCON** register'i denilen 8 bitlik bir yazmaç bulunmaktadır. Genellikle bu yazmaç **ADCON1** adı ile isimlendirilir.

Şimdi **PIC16F877** nin **ADCON1** yazmaç'ını inceleyelim.

2: ADCON1 REGISTER (ADDRESS 9Fh)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

Bit-7: **A/D** Sonuç Format Seçme biti dir.

1 olur ise sonuç sağa hizalanmış , **ADRESH** nin 6. uç bitleri 0 olarak okunur.

0 olur ise sonuç sola hizalanmış, **ADRESL** nin alt bitleri 0 olarak okunur.

Bit 6-4 arası kullanılmaz ve 0 olarak okunur.

Bit -3-0 arası **PCFG3 – PCFG0** A/D portu ayarlama kontrol bitleridir. İşte bu bitleri ayarlayarak portların seçimleri yapılır. Aşağıdaki tabloya bakınız.

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	Kanal/Refs
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	-	-	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	½

Şimdi tablo üzerinde biraz kafa yoralım.

Şayet **PCFG3:PCFG0** bitlerini **0000** olarak verir isek bu durumda **RA0-RA3 , RA5, RE0-RE2** bacaklarının tamamı **ANALOG** olarak ayarlanmış olacak ve artı referans Voltajı VDD den eksi referans voltajı ise VSS yani **GND** den alınacaktır.

Şimdi diyelim ki bize 3 adet Analog giriş lazım diğerleri Digital olabilir. +5V ve **GND** de referans voltajları olarak kullanılacak. Bu durumda tabloya bakar isek 0100 değeri tam istediğimiz ayarlamayı yapabiliyor. Tablonun en son hanesinde ilk değer kaç Analog giriş olduğunu / işaretinden sonraki değer ise harici referans voltaj girişi adedini gösteriyor. Seçtiğimiz değerde burası 3/0 olarak görülmektedir. Anlamı 3 adet Analog giriş ve sıfır referans voltajı var demek. Zaten biz Pic'in kendi voltaj girişlerini kullanmak istediğimizden gerçektende bize sıfır referans girişi lazımdır.

Pic'in voltaj girişleri referans voltajı olarak seçildiğinde Pic'e giren Analog voltaj değerinin 5 Volt'u aşmamasına dikkat edilmelidir.

Bu şekilde tablonun kullanılmasını anlattıktan sonra bu aşamada tüm girişlerin Digital seçilmesi konusunda değinmek isterim. Analog girişlerin Digital olarak kullanılması istenir ise tablodan sağ tarafta 0/0 değerinin bulunduğu satırda **PCFG3:PCFG0** değeri 0111 (Desimal 7) olarak görülür. Demek ki **ADCON1=7** veya **ADCON1=%0111** dediğimiz zaman **Pic 17F877** nin Analog girişleri iptal edilerek normal Digital giriş çıkışa dönüştürülmektedir. A/D çeviricisi bulunan Piclerde aksine bir komut bulunmaz ise ilk açılışta A/D pin'ler Analog olarak açılırlar. Dolayısıyla bu pinleri Digital kullanmak isterseniz mutlaka **Adcon1=7** komutunu vermeniz gerekir.

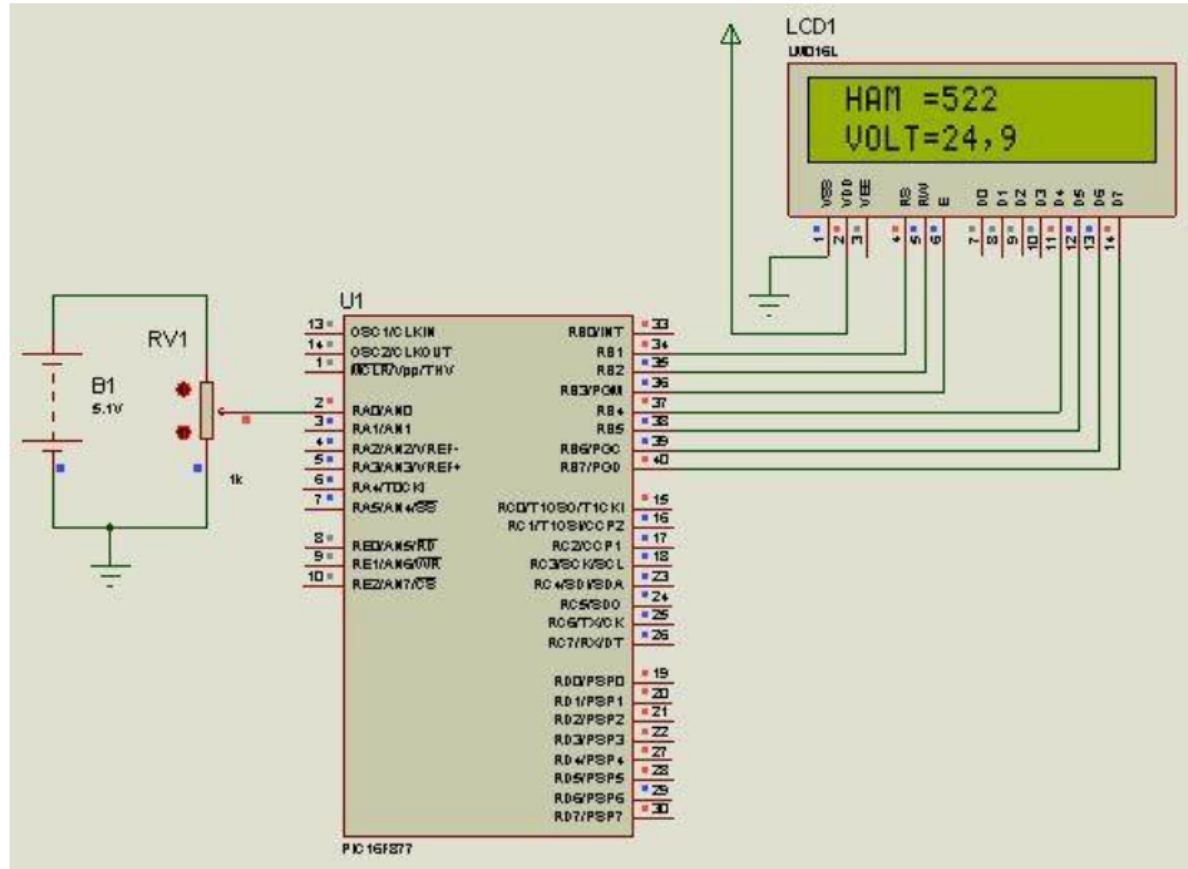
Bu kadar bilgi bir örnek üzerinde çalışmamız için yeterlidir. Şimdi bir örnek program yapalım.

Örnek Program:

Bu programımız **AN0** yani **PORTA.0** bacağına bağlayacağımız bir potansiyometre üzerinden voltaj okuması yapılmasını sağlayacaktır.

Bu durumda 1 adet Analog giriş gerekiyor. **PCFG3:PCFG0** değeri olarak tablodan **1110** değerini seçiyoruz. Bu seçenek tek Analog giriş (**AN0-RA0**) verdiği gibi Pic in voltaj

girişlerini de referans voltajı olarak kullanmamızı sağlamaktadır. Devremiz aşağıdaki gibi olacaktır.



Şimdi Programımızı yazalım

```

\*****
\*   Name       : ADC.BAS
\*   Author      : [ETE]
\*   Notice     : Copyright (c) 2005 [ETE]
\*               : All Rights Reserved
\*   Date       : 20.04.2005
\*   Version    : 1.0
\*   Notes      : *
\*               :
\*****
TRISA=%00000001
TRISB=0
TRISC=0
TRISD=0
\-----
@ DEVICE pic16F877
@ DEVICE pic16F877, WDT_on
@ DEVICE pic16F877, PWRT_ON
@ DEVICE pic16F877, PROTECT_OFF
@ DEVICE pic16F877, XT_OSC
\-----
DEFINE LCD_DREGPORTB 'LCD data bacakları hangi Porta bağlı?
DEFINE LCD_DBIT      4 'LCD data bacakları hangi bitten başlıyor?
DEFINE LCD_EREGPORTB 'LCD Enable Bacağı Hangi Porta bağlı?
DEFINE LCD_EBIT      3 'LCD Enable Bacağı Hangi bite bağlı ?

```



```

define LCD_RWREG          PORTB  'LCD R/W Bacağı Hangi Porta bağılı?
define LCD_RWBIT          2      'LCD R/W Bacağı Hangi bite bağılı ?
DEFINE LCD_RSREG          PORTB  'LCD RS Bacağı Hangi Porta bağılı ?
DEFINE LCD_RSBIT          1      'LCD RS bacağı Hangi Bite bağılı ?
DEFINE LCD_BITS            4      'LCD 4 bit mi yoksa 8 bit olarak bağılı?
DEFINE LCD_LINES          2      'LCD Kaç sıra yazabiliyor
\-----
DEFINE ADC_BITS            10     'A/D çevirim sonucu kaç bit olacak
DEFINE ADC_CLOCK           3      'Clock kaynağı (3=rc)
DEFINE ADC_SAMPLEUS        100   'Örnekleme zamanı mikro saniye cinsinden.
\-----
ADCON1=%10001110 '7. bit 1 yapıldı 10 bit sonuç almak için.
\-----
HAM    var word  'ADC den okunan ham Digital değer.
VOLT   var word  '16 bit değişken tipi seçtik kullanacağımız değer 10 bit
        olacak.
Mvolt  var byte
\-----
Low PORTB.2 ' LCD R/W line Low (W), şemada direkt GND ye bağlanabilir.
LCDout $FE,1 ' LCD de CLS yapar
pause 200 ' LCD nin açılması için gerekli süredir.
\-----

```

BASLA:

ADCIN 0,HAM '0 nolu kanaldan Analog değeri oku ve RAW değişkenine aktar.

BAK: IF ADCON0.2=1 THEN BAK 'Çevirme işlemi tamamlanınca Adcon0.2=0 olacak.

lcdout \$FE,1," HAM =",# HAM

Ham=ham+1 'Hesap kolaylığı açısından Ham değerini bir artırdık

'Okunan değer 0-5 volt için 0-1024 olacağından okunacak değer başına volt değeri

' $5/1024 = 0,0048828$ olacaktır. Sayı çok küçük olduğundan bu değeri 1000 ile

'çarpmalıyız. O halde $(Okuma/V) = 4,8828$ bu değeri 256 ile çarparsak

' $4,8828 \times 256 = 1250$ değerini buluruz. Bu değeri kullanarak 32 bit işlem yapabiliriz.

'1250 değeri $(5/1024) \times 256$ değerine karşılık gelmektedir.

'sonuçta ADC HAM değerini bu sayı ile çarpıp 256'ya bölersek sonucu elde etmiş oluruz.

'ancak sonuç 32 bitlik sınırdan olduğundan bu işi 32 bit çarpma ile yapıyoruz ve orta

'baytı aldığımızda ise zaten sayının 256'ya bölünmüş halini aldığımızdan başka bir

'işlem yapmadan sonucu elde etmiş oluruz.

```
Volt=(ham */ 1250)/100 'Ham ile 1250 yi 32 bit olarak çarp ve 100'böl  
Mvolt=Volt // 10 'Mvolt= Volt MOD 10
```

'Burada yeni bir komut veya işlem şekli görüyorsunuz. MOD alma yani (//) kalan bulma

'Bir sayını başka bir sayıya bölünmesinden sonra kalan miktarı bulma işlemine MOD

'alma denir. Örnek $W=A//1000$, A'yı 1000'e böl kalanı W değişkenine koy anlamındadır.

'bizde yukarıda Mili volt değerini bulmak için volt (10 ile çarpılmış halini) 10'a

'bölüp kalanı Mvolt değişkenine yerleştiriyoruz.

```
Volt=Volt/10
```

```
LCDOUT $FE,$C0," VOLT=",#VOLT,"",# Mvolt  
PAUSE 500  
GOTO BASLA
```

ADC işlemlerinde ölçümler genelde çok kararlı olmaz. Bir biri arkasına yapılan ölçümlerde ufak tefek farklılıklar görülebilir. Bu nedenle çoğunlukla ortalama alma metodu kullanılır. Aynı yerde birden fazla ölçüm alınır ve sonuçlar toplanır. Sonuçta kaç adet ölçüm yapılmış ise toplam, o sayıya bölünür. Örnek vermek gerekir ise;

```
For I=1 To 10  
ADCIN 0,HAM  
TOPLAM=TOPLAM+HAM  
NEXT I  
HAM=TOPLAM/10
```

Sonuçta Ham değeri 10 adet ölçümün ortalamasını gösterecektir ve oldukça kararlı bir değer olacaktır.

ADC çevirme işlemlerinde en önemli husus elde edilen Ham değerini istenen değere çevrilmesi için oluşturulacak formülün bulunmasıdır. Örneğin bir sıcaklık ölçümü yapıyorsunuz ve bir ısı [sensörü](#) değerini ölçüyorsunuz. 8 bitlik bir işlemde **0-255** arası değerler , 10 bitlik bir işlemde **0-1023** arası değerler bulacaksınız. Peki bu değerleri nasıl sıcaklık değeri olarak göstereceksiniz. İşin önemli noktası burası.

Hemen şu hesaplamayı yapmalısınız. 8 bit hesaplama için , Sensör den okunabilecek en yüksek değer 255 olacaktır. Peki bu değer kaç derece sıcaklığa eşdeğerdir?. Yani Sensör ün gösterebileceği maksimum sıcaklık ne olacaktır. Bunu sensör bilgilerinden (bilgi formu veya üretici bilgilerinden) alacaksınız. Diyelim ki 120 dereceye eş değerdir. O halde birim okuma başına düşen sıcaklık değeri $120/255 = 0,470$ derece olacaktır.

Bunu Pic de kullanabilmek için $0,470 \times 100 = 47$ olarak bir değer elde edilecektir. Demek ki ben okuduğum değeri 47 ile çarpıp sonucu 100 e bölersem sıcaklığı bulabilirim.

O halde sensör den okunan 134 dijital değerinin sıcaklık karşılığı ne olacaktır diye sorduğumuzda formülümüz bize;

$Sıcaklık = (134 \times 47) / 100 = 62,98$ derece olarak bulunacaktır.

ADC konusunda söylenecek son söz olarak üzerinde ADC çevirici bulunan her Pic farklı özellikler taşıyabilir. En azından Analog pinlerinin ayarlanması işlemi farklı olabilir. Bu nedenle farklı Pic ler ile çalışırken mutlaka bilgi formlarında belirtilen ADC özelliklerine göz atmanızı tavsiye ediyorum.

Diğer bir husus da program başında verilen ADC DEFINE parametrelerinin ayarlanmasıdır. Bunlar;

DEFINE ADC_BITS 10 'A/D çevirim sonucu kaç bit olacak
DEFINE ADC_CLOCK 3 'Clock kaynağı (3=rc)
DEFINE ADC_SAMPLEUS 100 'Örnekleme zamanı mikro saniye cinsinden.

Sırası ile tekrar gözden geçirirsek;

DEFINE ADC_BITS 10 ADC çevirim sonucunun kaç bitlik olacağını ayarlamaktadır. Sonuç 8 bit ise buraya 8 , 10 bit ise buraya 10 yazılmalıdır. Tabii ki kullanılacak değişken tespit edilen Bit değerine uygun olmalıdır.

DEFINE ADC_CLOCK 3 'Clock kaynağı (3=rc) Adc çevrim işlemlerinde her bit karşılığı bir Clock palsı ile belirlenir. Clock kaynağı olarak Pic in osilatörü veya dahili RC osilatör kullanılır. ADC işlemlerinde Bit başına çevirim süresi yaklaşık 1,6 mikro saniye olmalıdır. Pic in kendi osilatörü kullanılacak ise bu süreyi tutturmak için sistemi çalıştıran kristal frekansına göre bir hesap yapıp gerekli parametre bulunmalı ve buraya yazılmalıdır. Şimdi örnek bir hesap yapalım.

Bu hesapta kullanabileceğimiz formül şöyledir;

Tçevirim= X/Fosc. Burada X değeri 2 veya 8 veya 32 olabilmektedir. Fosc ise kristal frekansı olup MHZ cinsindendir. **Tçevirim=1,6 us.** Olduğuna göre şimdi diyelim ki kristal frekansımız 4 Mhz ve Pic'in osilatörü nü kullanacağız. Hesaba göre ;

(00) Tcevirim=2/4 = 0,5 us çıkar 1,6 dan çok küçük olduğu için uygun değildir.

(01) Tcevirim=8/4 = 2 us çıkar 1,6 dan büyük ve yakın olduğu için kullanılabilir.

(10) Tcevirim=32/4 = 8 us çıkar 1,6 dan çok büyük olduğu için uygun değildir.

Kullanılabilir olan 8 değerinin Define komutundaki karşılığı (%01=1) 1 dir. O halde komut

DEFINE ADC_CLOCK 1 'olacaktır.

Aynı hesabı **20 MHz** için yapar isek;

(00) Tcevirim=2/20 = 0,1 us çıkar 1,6 dan çok küçük olduğu için uygun değildir.

(01) Tcevirim=8/20 = 0,4 us çıkar 1,6 dan çok küçük olduğu için uygun değildir.

(10) Tcevirim=32/20 = 1,6 us çıkar 1,6 ile aynı olduğundan çok uygundur.

