

SEKTÖR ZAMANLARINA GÖRE ZAMAN AŞIMLARINI KAYDEDEN HAFIZALI SAYAÇ TASARIMI

Semih ŞENOL¹ Mehmet YAKUT² Murat SÖNMEZ³ İsmet KANDİLLİ⁴

^{1,2,3}Elektronik ve Haberleşme Mühendisliği Bölümü,

Kocaeli Üniversitesi, 41040, Veziroğlu Yerleşkesi, İzmit/Kocaeli

⁴Endüstriyel Elektronik Bölümü, Kocaeli Üniversitesi, 41500, Karamürsel/Kocaeli

¹semihsenol@gmail.com ²myakut@kou.edu.tr

³mrsonmez@kou.edu.tr ⁴kandilli@kou.edu.tr

Anahtar sözcükler: Mikrodenetleyici, Hafızalı Sayaç

ÖZET

In this study microcontroller based sector timing counter is designed and realised. These type of devices are used for performance evaluation of the personnel in industrial environment in which the working area is large. Thus the performance of the personnel can be observed by the employee, collecting and distributing the goods in product line.

1. GİRİŞ

Günümüzde mikrodenetleyicili sistemlerin kullanımı zirve noktasına ulaşmıştır. Bu gelişim içerisinde bu cihazları anlayıp kullanmak zorunlu hale gelmiştir. Bilindiği gibi mikroişlemci ile mikrodenetleyici arasındaki fark mikrodenetleyicinin, mikroişlemciyi yapısında bulundurmasının yanı sıra çevresel birimleri de içermesidir. Bu çevresel birimler eeprom, veri ve program belleği, giriş-çıkış arabirimi gibi işlevsel birimlerdir. Kuşkusuz bu birimlerin tek bir aygıtta birleşmesi kullanıcı açısından büyük kolaylık sağlayacaktır. Zira piyasada kolaylıkla bulunan veya kendinizin de yapabileceği ufak bir programlama devresi ve PC arabirim programı ile bu cihazları programlayıp kullanmak mümkündür. Makine dili olan "Assembly"nin yanı sıra "Basic" ve "C/C++" gibi dilleri destekleyen derleyicileri kullanarak programlar yazmak da mümkündür. İşte tüm bu kolaylıklar sayesinde mikroişlemciler yerine mikrodenetleyiciler kullanmak daha cazip bir hâl almıştır.

Bu çalışmada PC16F877 ile yapılmış LCD'li ve hafızalı bir sayaç devresinden tasarımı öngörülmektedir. Bu tip aygıtlar fabrikalarda ve üretimi, geniş alana yayılmış işletmelerde, dağıtım yapan elemanların çalışma hızını ölçmek için

kullanılmaktadır. Öncelikle bağımsız bir kullanıcı her sektör için zamanlarını kaydeder. Ardından cihaz dağıtım elemanına verilir. Kullanıcı tek bir tuşa basarak zamanı başlatır ve turu bitene kadar her sektörde bu tuşa tekrar basarak varsa zaman aşımının hafızaya kaydedilmesini sağlar. Böylece turu tamamlayan elemanın hangi sektörde ne kadar geç kaldığı öğrenilebilir. Bu değerlere bakan işveren, elemanın ne kadar verimli çalıştığını ve hangi sektörlerde performans düşüklüğü yaşadığını görebilir. Böylece maksimum verimde çalışılmış olunur.

2. MİKRODENETLEYİCİNİN ÖZELLİKLERİ

Hafızalı sayaç devresinde kullanılan mikrodenetleyici PIC16F877 14-Bitlik komutları işleyebilen bir işlemciye sahiptir. Çalışma hızı maksimum 20 Mhz'dir. Endüstriyel olanlarında bu değer 40 Mhz'e kadar çıkabilir. Bu çalışma hızı aslında gerçek çalışma hızı değildir. Aygıt çalışma esnasında bu değeri 4'e bölerek kullanır. Eğer 4 Mhz'lik bir kristal kullanıyorsanız komut işleme hızınız $4 \text{ Mhz}/4 = 1 \text{ Mhz}$ yani $1/1 \text{ Mhz} = 1 \mu\text{s}$ 'lik saat hızınız olacaktır.

Program belleği 8Kb, veri belleği 368-Byte ve eeprom'u da 256-Byte'dır. Bu üç bellek arasındaki farkı; Program belleği, herhangi bir derleyici ile yazdığınız programınızı kaydettiğiniz yerdir. Aygıt çalışırken bu bellekten sadece okunur. Program belleğinin değiştirilmesi sadece programlayıcı devre ile aygıtı programlarken olur. Veri belleği, programınızda tanımladığınız değişkenlerin işlendiği alandır. Bu alanda aygıt çalışırken veriler işlenir, programınıza göre gerekli düzeltmeler yapılır, gerekiyorsa çıkışlara verilir veya veri belleğinde

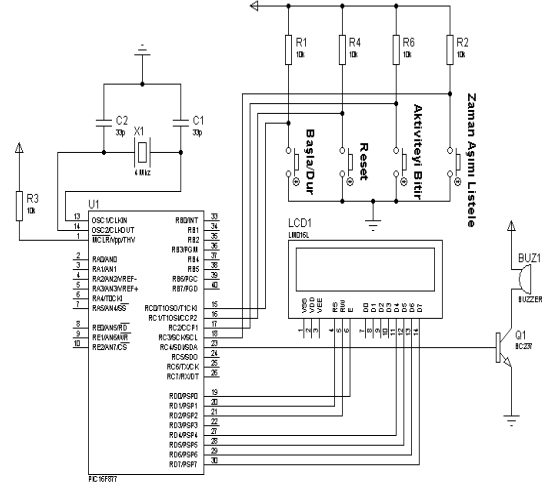
tutulur. Eğer verinin saklanması isteniyorsa, kalıcı bellek olan eeproma yazılır. Veri belleğindeki değişkenler aygıtta güç geldiği sürece tutulduğu için buradaki bilgiler kalıcı değildir. Aygıt yeniden başlatıldığında programdaki değişkenler ilk değerleri ile birlikte buraya yazılırlar. Program çalıştıkça değerleri değiştirilir. Bu nedenle aygıt kapandıktan sonra da güncel bilgilerin saklanması isteniyorsa bu bilgiler eeproma yazılır. Eeprom belleğine yazılan veriler silmedikçe orada kalır. Eeprom belleği ve komut işleme hızı bizim için önemli niceliklerdir. Çünkü bu devre hafızalı sayaç olarak çalışacaktır. Bu nedenle eeprom büyüklüğü ve zamanlama kesinliği önem taşır.

Burada, 4 Mhz'lik kristal kullanılmıştır. Bu nedenle 1 Mhz'lik (1µs) komut işleme hızı vardır. Mikrodenetleyici içerisindeki timer'lerden 8-Bitlik olanı zamanlama için kullanılmıştır. Bu timer'ın seçilmesinin en önemli nedeni, "prescaler" olarak adlandırılan "ön bölücü" değerinin çok seçenekli olmasıdır. Prescaler timer'ın ne kadar zamanda, bir bir artacağını belirleyen bir büyüklüktür.

Bu timer zamanlamada kullanılacağı için 1sn'nin çok önemi olacaktır. Bu değeri mümkün olan en hassas şekilde tutturmak gerekmektedir. Bu nedenle prescaler 64 olarak ayarlandı. Böylece $64 \times 1\mu s = 64\mu s$ 'de bir, timer bir artacak şekilde getirilmiş oldu. Timer 8-Bit olduğu için $256 \times 64\mu s = 16.384 \mu s$ 'lik bir gecikme elde edilmiş oldu. $1sn = 1.000.000 \mu s$ olduğuna göre $1.000.000 / 16.384 = 61,03515625$ kere timer kesmesi gelirse yaklaşık 1 saniyelik süre bilgisini elde etmiş oluruz. Burada 0,03515625'lik kısmı saymazsak yaklaşık 61 kere timer kesmesinin gelmesi 1sn'ye karşılık gelmektedir. Bu işlemler sırasında 61 kere timer kesmesinin geldiğini anlayıp saniyeyi bir arttırmak, saniyenin 60 olduğunu anlayıp da dakikayı bir arttırmak gereklidir.

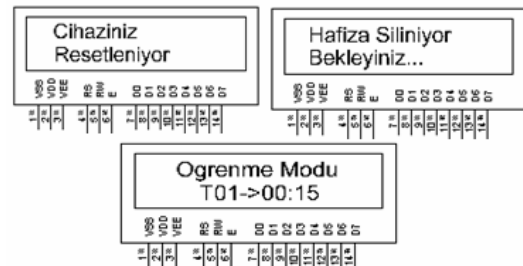
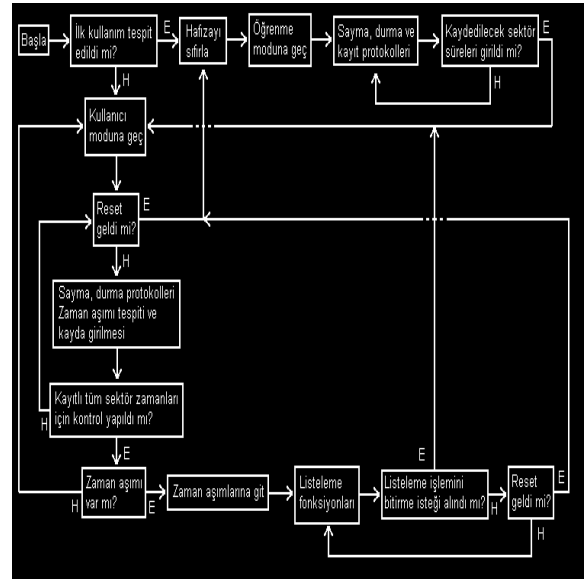
3. HAFIZALI SAYAÇ TASARIMI

Şekil-1'de hafızalı sayaç devresi verilmiştir. Devrenin genel çalışması esnasında süre bilgisi LCD'de gözükmektedir. Önceden kaydedilmiş sektör bilgileri de güncel süre bilgisinin yanında gözükmektedir. Bu esnada varsa zaman aşımaları listelenebilir. Aygıt ve zaman aşımaları resetlenebilir. Bir sektörde geç kalınırsa sesli uyarı da verilmektedir. Böylece kullanıcı hızlanıp açığını kapatabilir. Devre ilk örnek olarak tasarlandığından düzeltme ve eklemeler yapılabilir. Örneğin zaman aşımalarını ve aygıtı resetlemek için şifre sormak, tüm sektör zamanlarını toplayarak genel bir parkur süresi çıkartmaktadır. Kullanıcı bir sektörde geç kalmışsa bile sonradan hızlanıp farkı kapatabiliyor mu kontrol edilebilir. Şekil-2.'de Sistemin algoritması görülmektedir.



Şekil-1. Hafızalı Sayaç Devresi

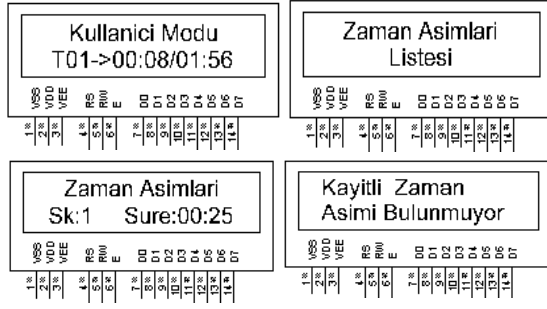
Şekil-2. Sistemin Algoritması



Şekil-3. LCD 'de Öğrenme Modu ve Resetlenmesi

Devre ilk kez kullanıldığında hafıza olarak kullanılan eeprom belleğinin ilgili kısımları sıfırlanır. Ardından öğrenme moduna girilir ve sektör zamanları kaydedilir. Şekil-3'de öğrenme modu ve LCD de cihazın resetlenmesi gözükmektedir. Ardından cihaz kullanıcı moduna geçiş yapar ve kullanıcının sektör zamanları ile kayıtlı zamanlar

karşılaştırılır. Burada zaman aşımı olursa sesli uyarı verilir. Şekil-4’de kullanıcı modu aşağıda verilmiştir.



Şekil-4. LCD ‘de Zaman aşımı

Zaman aşımının olduğu tespit edilir ve sayma sonunda kullanıcının zamanı, kayıtlı zamandan çıkarılarak eeproma kaydedilir. Sektör zamanlarının sayımı bittikten sonra zaman aşımı varsa zaman aşımaları listesi görüntülenir. Zaman aşımı yoksa ilk sektör zamanına geri dönülerek beklemeye geçilir. Şekil-4’ de zaman aşımaları listesi verilmiştir.

Bu tip LCD’lerde Türkçe karakterler görüntülenememektedir. Zaman aşımaları listesi sektör zamanları sayılırken de görüntülenebilir. Kayıtlı zaman aşımı yoksa ekrana “Kayıtlı zaman aşımı bulunmuyor” gelecektir.



Şekil-5. LCD ‘de Kayıtların Silinmesi Durumu

Zaman aşımaları listesinde iken reset tuşuna basılırsa, zaman aşımı kayıtlarını silmek isteyip istemediğinizi soran bir mesaj ve ardından bir kere daha reset’e basılırsa kayıtların silindiğini gösteren ibareler ekrana basılır.

Eğer kullanıcı modunda reset’e basılırsa, cihazın resetlenip resetlenmeyeceğini soran bir ifade ekranda görüntülenir. Bir kere daha reset’e basarsanız cihaz her şeyi ile birlikte resetlenir. Herhangi bir anda aktiviteyi bitir butonu ile ilgili işlem den vazgeçebilirsiniz.

Cihazda PIC16F877 gibi bir mikrodenetleyici kullanıldığı için, hafıza sınırlarınca Türkçe yönergeler yazılarak kullanımın kolaylaştırılması amaçlanmıştır. Ülkemizde bu tip aygıtlar çoğunlukla İngilizce olduğu için programlamanın ana dilimizi kullanarak yapması uygun görülmüştür.

256-Byte eepromda sıkıntı çekilmemiştir. Aşağıda eeprom belleğinin nasıl kullanıldığını gösteren bir tablo-1’de görülmektedir.

0x00	Status değişkeni (olacak ve olmuş durumların saklandığı değişken)
0x01	Sektör 1 dakika verisi
...	
0x50	Sektör 25 saniye verisi
0x51	Zaman aşımaları için bellek alanı
...	(Bir sektör gecikmesinde sektör numarası, dakika ve saniye farkı kaydedilir)
0x7D	Zaman aşımaları bellek alanı sonu (maksimum 25 sektör için)
0xFF	Kaydedilmiş sektör sayısı tutulur.

Tablo-1. Eeprom Belleğinin Kullanımı

Toplamda 127-Byte kullanılmıştır. 129-Byte boş alan mevcut olduğundan sektör sayısını daha da arttırmak mümkündür.

Aygıtta 7 parçalı gösterge kullanılmaması yazılım ve yönergeler açısından esneklik sağlamıştır. Zira LCD’nin kendi belleği olduğu için, bir kere ekrana yazılacak veriyi yollamak yeterli olmaktadır. 7 parçalı göstergelerde ise bastırma işlemi tarama mantığına dayanmakta ve yazılımda belirli aralıklardadır. Bu aralıklar insan gözünün fark edemeyeceği bir hızda olmalıdır ve göstergeler tazelenmelidir. Bu nedenle program döngüsünde display’de görüntülenen veri sürekli tazelenmelidir. Bu zaman kaybı hassas zamanlama yapan mikrodenetleyicilerde istenmeyen bir zaman kaybıdır. Bu gibi durumlarda bu işlemler birbirinden ayrılarak farklı işlemcilerle ya da saat çiplerine verilmelidir. Burada LCD kullanmak çok daha elverişlidir.

LCD haberleşmesi 8, 4 ve 1 bit ile yapılabilmektedir. 1 bit zorunlu olmadıkça pek tercih edilmez. 8 bit kullanım ise, çok pin işgal ettiği için genellikle tercih edilmemektedir. Bu nedenle burada 4 bitlik haberleşme kullanılmıştır. LCD’nin güç bağlantıları yapıldıktan sonra parlaklık ayarı da yapılmalıdır. Bunun için V_{EE} ucu kullanılır. V_{DD} ucu +5V ve V_{SS} ucu da GND olarak bağlanmalıdır. V_{EE} den gerilim değerine göre parlaklık değişecektir.

Burada kullanılan buzzer DC tipte bir buzzer’dır. Yani üzerine DC gerilim verince ses verir. Bir de AC tip buzzer’lar vardır. Bu tiplerin ses çıkarması için üzerine belirli frekansta AC işaret uygulamak gerekir. Bu devrede belirli bir frekansta işaret üretmemek için DC tipte buzzer kullandık. Aslında PIC16F877’nin PWM çıkışı kullanarak bu tip bir işaret üretmek de mümkündür. Alternatif olarak tüm devreler

kullanılarak işaret üretmek mümkündür. Bu yöntemle devrenin boyutları ve maliyetleri arttıracığından burada tercih edilmeyen bir yöntem olacaktır. Tamamen donanımsal olan bu modül işlemciye yüklenmediği için çalışma hızını etkilememektedir.

4. SONUÇ

Mikrodenetleyici kullanılarak yapılan hafızalı sayaç tasarımı ile kullanıcı modülleri verilmiştir. Zaman ve performans durumları değerlendirme ve denetleme için çoğu işletmelere önerilebilir. Devre, yapısı itibarıyla küçültülmeye ve yazılımsal olarak da genişletilmeye uygundur. Sektör zamanlarını denetlemek için hafızalı sayaç tasarımı ile verimlilik değerlendirilmiştir. Önerilen tasarım ölü zaman ve performans analizi için kullanılabilir.

KAYNAKLAR

- [1] Bates M, Interfacing PIC microcontrollers embedded design by interactive simulation., Newnes, 2006.
- [2] Uzun O, Mikroişlemciler ve assembler programlama, Alfa, 1994.
- [3] PIC16F877, www.microchip.com

- [4] Çolak İ., Bayındır R., “PIC 16F877 ile DA Motoru Hız Kontrolü”, Pamukkale Üniversitesi Mühendislik Fakültesi, Mühendislik Bilimleri Dergisi, Cilt 11, Sayı 2, 2005.Chang C. M.,
- [5] Sahin H, Dayanık A, Altınbasak C, PIC programlama teknikleri ve 16F877A

Özgeçmişler:

Semih Şenol, Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü 2006 yılı mezunlarından olup, halen aynı üniversitenin Fen Bilimleri Enstitüsünde yüksek lisans çalışmalarına devam etmektedir. İlgili alanları otomasyon, algılayıcı sistemleri ve endüstriyel kontrol ve veri toplama sistemleridir.

Mehmet Yakut, 1966 İzmit doğumlu olup, 1987 yılından beri Kocaeli Üniversitesinde görevlidir. Yüksek Lisansını Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Müh. Bölümünde 1992 yılında “adım motorların kapalı çevrim hız denetimi” konusunda tamamlamıştır. Doktora çalışmasını 2002 yılında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümünde “vektör kuantalama ile görüntü sıkıştırma” konusunda tamamlamıştır. İlgili alanları görüntü işleme ve sıkıştırma, gömülü sistemler ve kablosuz algılayıcılardır.