

T.C

FIRAT ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

EET-302 ve EET-344 MİKROİŞLEMCİLER LABORATUVARI DENEY FÖYÜ

Doç. Dr. Melih Cevdet İNCE

DENEYLER

Deney_1: PIC PROG DEKA V5 Eğitim Seti Kullanımı Ve Programlanması

Deney_2: Program yazma, derleme, pic'e yükleme, Port okuma yazma, pic16f84a ve pic16f877a da aynı programı çalıştırmak için gerekenler, döngü kavramı,7 parçalı göstergeye yazma

Deney_3: Program yazma, derleme, pic'e yükleme, 2 tane7 parçalı göstergeye yazma, tablodan okuma, step motor sürme

Deney_4: 2 tane7 parçalı gösterge ile 00-99 ileri-geri sayıcı, eksik tamamlamak için serbest çalışma.

Deney_5: ADC(Analog Digital Converter), RB0 Harici kesmesi

Deney_6: TIMER0'ın sayaç olarak kullanılması, TIMER0 Kesmesi

Deney_7: Yazılımla pwm üretilmesi

Deney_8: Yazılımla pwm üretilmesi yeni bir program

Deney_9: 8085 Simülatör Kullanımı

Deney_10: 8085 Simülatör Kullanımı-2

NOT:

□ Deneyler her hafta programda belirtilen saatte mesleki yazılım uygulamaları laboratuvarında yapılacaktır, deney başlamadan 15 dk. önce laboratuvarda hazır bulununuz.

Deney föyünde bulunan bilgilere, deneye gelmeden önce çalışarak geliniz.

Deney föyü olmayanlar laboratuvara alınmayacaktır.

🗆 Herkes kendi grubunda ve saatinde gelecektir. Kendi grubunda gelmeyenler diğer gruplarla alınmayacaktır.

DENEY 1:PIC PROG DEKA V5 EĞİTİM SETİ KULLANIMI VE PROGRAMLANMASI

Konular: Deneylerde kullanılacak olan eğitim setinin incelenmesi. Pic komutlarının bilgisayarda yazılması,derlenmesi ve pic'e yüklenmesi için gerekli olan programların öğrenilmesi

Eğitim Seti Kullanımı

PIC PROG DEKA V5 deney seti ile 18, 28 ve 40 pin'li PDIP paket tipine sahip PIC'leri PC'nizin USB port'undan gönderilen elektrik sinyalleri ile programlanabilmektedir. Program biter bitmez programlanan PIC otomatik olarak çalısmaya baslayacaktır.

Egitim Setinin Kullanım Amacı

1. **Microchip** firmasının ürünü olan **PIC** mikrodenetleyicilerin bir baska cihaza ihtiyaç duymadan, sorunsuz ve güvenilir biçimde programlamasını yapmak.

2. Programlanan bir PIC mikrodenetleyiciyi yerinden sökmeden, uygulama modunda otomatik olarak çalıstırmak ve program fonksiyonlarının incelenmesini saglamak.

PIC PROG DEKA V5, çift taraflı baskı teknigi ile üretilmis 160x160 mm boyutlarında PCB üzerine yerlestirilmis devre elemanların meydana gelmis bir PIC programlama ve deneme devresidir. V5 deney seti üzerindeki devreleri tanımanız, bu seti dogru olarak kullanabilmeniz için gereklidir. Sekil- 1'de PCB üzerinde yerlesik olarak bulunan elektronik elemanların tanınması için yazılanların kolayca görülebilmesi ARES programında çizilmis baskı devre semasının elemanlı yüz baskısı görülmektedir. Sekil-2'de de V5 üzerindeki devreleri incelemek amacıyla 20 bölüme ayrılarak verilen PCB'nin resimsel görünüşü görülmektedir.



Sekil-1: PIC PROG DEKA V5 Deney Setinin bölümlere ayrılmış görüntüsü.



Sekil-2: PIC PROG DEKA V5 Deney Setinin resimsel görünüşü

Besleme Girisi ve Devresi

V5 deneme setine güç kaynagı jak girisi (J21), ON/OFF anahtarı ve güç LED'inin bulundugu kısımdır. Bu set ile birlikte verilen 12V DC güç kaynagının fisini bu jack girisine takınız. ON/OFF yaylı anahtarı basılıyken devreye enerji verilmis olur ve kırmızı LED yanar. Devrenin enerjisini kesmek için ikinci defa yaylı anahtara basmak gerekir.



Programlama Devresi ve Soketi

Programlanarak monte edilen 16F628A PIC mikrodenetleyici, FT232BL entegresinin seri iletişim protokolüne çevirdigi kodlarını programlayacagınız PIC'e ulastırmak için kullanılır. Ayrıca PIC'in programlanması için MCLR pin'ine +13V'luk bir gerilimi PIC'in uygun ucuna yönlendirilmesini yapar. Programlanacak olan PIC'e uygulanması gereken data ve clock girisleri de bu mikrodenetleyici vasıtasıyla yapılır. Tüm bu isleri PIC içerisindeki özel bir program yürütmektedir.



Seçilebilir B Portu Çıkısları

Kart üzerindeki deneme soketlerinden birisine yerlestirilmis olan bir PIC'in B portu 4 farklı displaye yönlendirilerek kullanılabilmektedir. Bu display'ler alfanümerik LCD, 2x7 segment nümerik LED display, 5x7 dot matris LED display ve 8 adet normal LED'den olusmaktadır. Aşağıdaki Sekil de görüldügü gibi SW1 DIP anahtarı ve ULN2003 sürücü entegresi ile bu display'ler kontrol edilmektedir.



2x7 Segment LED Display

Set üzerinde denenecek olan PIC'lerin B portundan gönderilen verileri ondalık sayı biçiminde göstermeye yarayan devredir. Sekilde devresi görülmektedir. 2 adet ortak katotlu 7 segmentli display'den hangisinin aktif edilecegi PIC'in RA0 ve RA1 pinlerinden gönderilen verilerle belirlenir. ULN2003 entegresi gönderilen bu veri için bir buffer görevi üstlenir. Böylece displayden daha yüksek akım geçirilebilir ve segmentleri daha parlak yanması saglanır. ULN2033 gibi bir sürücü kullanılmasaydı segmentlerin ortak katot akımı PIC'in 25 mA lik source akımıyla sınırlı kalacagından yeterli parlaklık saglanamayacaktı. Hangi dijit aktif edilirse o dijit bir rakam gösterir. Hangi rakamın gösterilecegi ise PIC'in RB0~RB7 pinlerinden gönderilen uygun kodlanmıs verilerle saglanır. İstege baglı olarak displayde bir veya iki rakamlı sayılar gösterilir. Görülmesi istenmeyen dijit pasif (karartılmıs) durumda tutulur. Dijitlerin ortak katodunu RA0 ve RA1 uçlarına baglantısını saglamak için SW1 DIP anahtarının 7SEG-1 ve 7SEG-2 çıkısları ON konumuna getirilmelidir. Bu anda diger display çıkışlarıyla çakısma olmaması için DIP anahtarın diger kontaklarının tamamı OFF konumunda bulunmasına dikkat edilmelidir.



5x7 Dot Matris LED

5 sütun 7 satır biçiminde düzenlenmis LED'lerden olusan dot matris LED display'in bulundugu devredir. Satır verileri RB0~RB7 pinlerinden gönderilir. Sütunlar ise RC0~RC4 pinlerinden gönderilir. Dot matris display'in sütunlarını kontrol edebilmek için SW1 DIP anahtarının 4, 5, 6, 7, 8 kontakları ON konumuna getirilmelidir. Bu anda LED ve 7segment display ile karısma olmaması için DIP anahtarın diger kontakları OFF konumunda olmasına dikkat edilmelidir.



LED Gösterge

Denenecek olan PIC'lerin B Portu çıkıslarındaki dijital (TTL 5V) çıkısları görmek için kullanılır. LED'ler high-aktif yanacak biçimde baglantısı yapılmıs ve 470 ohm dirençlerle akım sınırlandırılması yapılmıstır. PortB Çıkıslarını LED'lere yönlendirmek için "Port seçme anahtarının(SW1)" PORTB-LED kontagı ON konumuna diğer kontakları OFF konumuna alınmalıdır.



PORTB Seçme Anahtarı

Denenecek olan PIC'lerin B portu çıkısları seçime baglı olarak 7 segment display'e, LCD'ye, veya LED'lere yönlendirilebilmektedir. Aşağıdaki Sekilde görüldügü gibi port çıkıslarına baglı olan displaylerin kontrol girisleri PIC'in A portundan yapılmaktadır. DIP anahtarın ilgili kontagı kapatıldığında istenilen kontrol ucu display'e baglanmıs olur. Bundan sonra da PIC'in A portundan gönderilen "1" veya "0" verileri displayleri kontrol edebilecek duruma getirmis olur.



PIC Programlama ve Deneme Soketleri

V5 egitim setinin ortasına 3 adet soket monte edilmistir. Microchip'in ürettigi farklı pin sayısına sahip olan DIP tipi PIC'leri devre üzerindeyken programlamak ve daha sonra da denemek için kullanılan soketlerdir. Bu soketler 18, 28 ve 40 pinli PDIP tipi PIC mikro denetleyicileri yerlestirmeniz içindir. Çalısmak istediginiz PIC mikro denetleyicileri bu soketlerden uygun olanına PIC'in sol üst kösesi 1 nolu pin olacak sekilde yerleştirerek çalısmaya baslayabilirsiniz.

Her soketin yan taraflarında pin adları yazılmıstır. Böylece ölçme gerektiginde ilgili pin kolaylıkla bulunabilmesi saglanmıstır.



4 ve 20 MHz kristal osilatör

Denenecek PIC'lerin seçime baglı olarak iki farklı kristal (4 MHz, 20 MHz) osilatörlerle çalıştırılabilmesi bir jumper kullanılmaktadır. JP2 kısa devre jumperi üst tarafa takıldığında PIC 4 MHz, asagı tarafa takıldığında 20 Mhz ile çalıştırılır.



PORTA Giris/Çıkıs Buton ve LED'leri

Bu devrede A portunun 6 pini (RA0~RA5) dijital giris olarak kullanılabilecek biçimde düzenlenmistir. Yine PORTA'nın 4 pini de dijital çıkıs olarak kullanılabilecek biçimde 4 adet LED bulunmaktadır.



Pull-Up/Pull-Down Anahtarı

Pull-UP, Pull-DOWN anahtarı deneme soketleri üzerine takılan PIC'lerin A portunun pinlerini farklı biçimde organize eder. Bilindigi gibi pull-up yapılmıs butona basılmadıgında ilgili pine uygulanan gerilim 5V'tur (lojik "1"). Butona basıldığında ise ilgili pin girisine uygulanan gerilim 0V tur (lojik"0"). Pull-DOWN durumunda ise butona basılmadığında ilgili pin girisine 0V (lojik "0"), basıldığında ise 5V'luk (lojik "1") gerilim uygulanmaktadır.

PORT A, B, C, D, E Giris/Çıkıs Pinleri

V5 deney seti ile gelistirilebilecek projelerde farklı girisler oldugunda gerekli esnekligi sağlamak amacıyla deneme soketlerine baglanan PIC'lerin pin çıkısları 10'lu gruplar halinde disi precision pinlerle PCB'nin sag tarafına yerlestirilmistir. 1. ve 10. pinler VCC ve GND için ayrılmıstır. Diger 8 pin, 8 bit'lik port pinlerine ayrılmıstır.



PİC KOMUTLARININ BİLGİSAYARDA YAZILMASI, DERLENMESİ VE PİC'E YÜKLENMESİ

Bir PIC'e program atabilmek için aşağıda verilen işlem adımlarının yapılması gereklidir.

- 1) Masaüstünde boş bir metin belgesi açınız.(Mouse sağ tuş/yeni/metin belgesi)
- 2) Metin belgesine PIC programını her komut bir satırda olacak şekilde yazınız.

ÖR: MOVLW H'00' MOVWF TRISB

- Metin belgesinde programı yazdıktan sonra Dosya/Farklı kaydet dedikten sonra dosya adı kısmına kaydetmek istediğiniz dosya ismini yazarak yanına .asm yazınız. ÖR: uygulama1.asm
- 4) Kaydet dedikten sonra ekrana .asm uzantılı dosya gelecektir.
- 5) Assembeler diline çevirilmiş olan ASM uzantılı programınızı PIC'e aktarmak için HEX uzantılı makine diline çevirmeniz gerekmektedir.
- 6) MPASM programını çalıştırınız.

🔉 MPASM v5.06 - Microchip Technology, Inc. —								
Source File Name:	Source File Name:							
F:\mikroişlemciler\MPA	SM Suite\MCP250> Bro	wse						
Options:	Options: MICROCHIP							
Radix: • Default • Default • Default • Default • All Messages • Octal • Cotal • Cotal • Cotal • Cotal • Default • Operation • Operation • Operation • Operation • Operation • Operation • Operation • Operation • Operation • Default • Default • INHX88 • INHX88 • INHX88 • INHX82 • Object File • Object File • Operation • O								
Extra Options:								
x Exit Assemble ✓ Assemble ✓ Save Settings on Exit ? Help								

- 7) Program ayarları yukarıdaki şekilde görüldüğü gibi yapılmalıdır.
- 8) Browse seçeneğinden .ASM uzantılı dosyanızı seçiniz.
- Processor kısmında kullanacağınız PIC'i seçmelisiniz. Deney setlerinde PIC16F877A olduğu için 16F877A denetleyicisini seçiniz
- 10) Eğer yazdığınız programda bir hata yok ise derleme işlemi başarı ile yapılmıştır. Aşağıdaki şekil A'da olduğu gibi rapor sayfası yeşil renk olacaktır ve errors:0 diyecektir.



- 11) Fakat yazdığınız programda bir yazım hatası var ise Şekil B de olduğu gibi rapor kırmızı olacaktır ve hata sayısını gösterecektir.
- 12) Yapmış olduğunuz hataları görmek için masaüstüne gelen .ERR uzantılı dosyayı Birlikte aç/Not defteri seçenekleri ile açarsanız hata yaptığınız satırları ve hatanızı görebilirsiniz.

- 13) Hatalarınızı düzeltmek için ya not defterinde yazdığınız programı tekrar açarak hataları düzelttikten sonra tekrar farklı kaydet ile .asm uzantılı kaydedin. Ya da direk .ASM uzantılı dosyanızı Birlikte aç/not defteri ile tekrar açarak gerekli düzenlemeleri yaparak kaydediniz.
- 14) Hataları düzelttikten sonra tekrar MPASM ile derleme yapınız.
- 15) Bu işlemleri derleme sonunda rapor yeşil renk alana kadar tekrar ediniz.
- 16) Derleme işlemi başarı ile yapıldıktan sonra masaüstüne .HEX uzantılı dosya gelecektir.
- 17) Setinizin güç ve USB kablosunu bağlayınız.
- 18) Makine diline çevrilmiş olan bu dosyayı PIC'e yüklemek için MicroPro programını kullanınız.



- 19) MicroPro programını açtığınızda yukarıdaki ekran açılacaktır.
- 20) Programı yüklemeden önce setin bilgisayara aktif şekilde bağlandığından emin olunuz. Sol tarafta eğer COM x yazıyor ise setiniz bağlanmamıştır. Bu durum da program yüklenemez. Ancak COM 3 gibi bir değer yazıyor ise ve yan tarafında board connected yazıyor ise bağlantı sağlanmıştır.
- 21) Sağ alt köşede chip selection kısmından kullandığınıc PIC'i seçiniz.
- 22) Load butonuna basarak .HEX uzantılı dosyanızı seçiniz. Yukarıdaki şekilde görüldüğü gibi makine diline çevrilmiş olan komutlar ekrana gelecektir.
- 23) PIC'i programlamak için program butonuna tıklandıgında aşağıda sekildeki uyarı penceresi gelir. Bu pencerede, sigortaların default (programın belirlediği degerler) degerlere ayarlandıgını, bu degerleri degistirip degistirmek istemediginiz sorulur. Programlamaya baslamadan önce bu degerleri kendi seçiminize göre ayarlamak istiyorsanız YES butonuna basmanız gerekir. Programın kabul ettigi degerlerin aynen yazılmasını isterseniz NO butonuna basmanız gerekir.

Confirm	n			×
?	The FUSES are se Do you wish to ee	et to the def dit the fuses	fault erased valu	ues. nming?
	Yes	No	Cancel	

- 24) Sigorta degerlerini kaynak program içerisinde yazdıysanız sekil 40'daki confirm penceresi ekrana gelmeden direkt olarak programlamaya geçilecektir.
- 25) Ayarları değişmek için YES butonuna bastığınızda aşağıdaki ekran gelmektedir. Ayarları aşağıda görüldüğü gibi yapınız.

o 📙 I 🖓 🛄 ╤ I SURUCU	_ARAYUZ		
USE Edit			×
	Disabled 💌	PWRTE	Disabled 💽 🔍
BODEN	Disabled 👻	LVP	Disabled 🗾
CPD	Disabled 🗸	WRT Enable	Enabled
DEBUG	Disabled 🗸	Oscillator	×T •
Code Protect	Disabled 👻		

- 26) OK butonuna bastığınızda program aktarma işlemi yapılacaktır.
- 27) Set üzerinde yapmanız gereken değişiklikler her program için farklılık gösterebilir. Bunun için her bir uygulama üzerindeki açıklamaları dikkatlice okuyunuz.

Programlama Yapılırken Dikkat Edilecek Hususlar

- 1) Eger micropro donarsa, klavyenin CTRL+ALT+DELETE tuslarına birlikte basılır. Ekrana gelen "Windows Görev yöneticisi"nden micropro yazılımı seçilip "Görevi Sonlandır" dügmesine tıklanarak program sonlandırılır. Program daha sonra tekrar çalıstırılır.
- 2) V5 egitim seti üzerine takılan PIC'lerin programlanması RB6, RB7 pinlerine gönderilen CLOCK ve DATA sinyalleri ile yapılmaktadır. Programlarınızı denerken PORTB çıkıs veya giris olarak kullanıldığı durumlarda bu portu herhangi bir modüle baglamıs olabilirsiniz. Bu durumda micropro yazılımından programı gönderdiğinizde program yazma hatası ile karsılasabilirsiniz. Bu gibi hata ile karsılasmamak için **PORTB seçme DIP anahtarlarını** programlama esnasında OFF konumuna alıp, programlama bittikten sonra gerekenleri ON konumuna alınmalıdır.
- 3) MikroPro'yu açtığınızda setin COM port görünmediyse COM x yazan yere çift tıklayarak setin bağlı olduğu COM numarasını giriniz. Bu numarayı öğrenmek için Denetim masası/ Donanım ve ses/ Aygıt Yöneticisi/Bağlantı noktaları'nı açınız.
- 4) Program yüklenirken bağlantı hatası verir ise USB kablosunu başka girişlere takarak deneyiniz. Bu şekilde de sorun çözülmez ise seti değiştiriniz.

MicroPro' nun Kullanılması

File Menüsü

Bu menüden yükleme (Load), kopyalama (Save), başka bir dosyayla birleştirme (Merge), yenileme (Refresh), ekranı temizleme (Clear) gibi işlemler yapılabilir.



Load: Bu seçenek veya Load butonu PIC'e yazdırılacak olan programı yüklemek için kullanılır.

Save: Bu seçenek veya buton ekranda bulunan HEX program kodlarıyla birlikte sigorta (FUSE) ve EEPROM bellekte bulunan verileri dosya olarak kopyalamak için kullanılır.

Merge: MicroPro'ya yüklenmiş bir program verilerine başka bir program verileriyle birleştirmek için kullanılır. Yeni program verileri bos veri adreslerine yüklenir. Bos olmayan adreslerdeki veriler değiştirilmeden bırakılır. Örnek olarak, bu özellik bootloader ve program verilerinin birleştirilmesi için kullanılabilir.

Refresh: Bu seçenek veya Refresh butonu program belleğindeki dosyanın yeniden yüklenmesini sağlar.

Clear: Bu seçenek program belleğindeki verileri siler ve ekranı boşaltır.

Programmer: Bu seçenek elinizde bulunan programlayıcı software'i seçmeniz için kullanılır. Örnegin K149-A programlayıcınız varsa K149-A seçmelisiniz. V5 üzerindeki programlayıcı K182 versiyonu olduğundan K182 seçmelisiniz. Bu işlem normal olarak MicroPro açıkken programlayıcıyı bilgisayarınıza bağladığınızda otomatikman seçilir. **Port:** Bu seçenekle COM port seçimini değiştirebilmek için bir iletişim penceresi açılır. Seri port MicroPro ekranının sol alt kösesindeki COMx yazısı üzerine çift tıklamak suretiyle de yapılabilir. COM port seçimi doğru yapıldıysa açılan pencerede COM kelimesinin hemen yanında numarası da görülecektir. Port seçimi doğru yapılmadığında COMx görülecek ve hata mesajı görüntülenecektir. Bazen COM port doğru seçilmiş görülse bile program donanımı bulamayabilir. Bu durumda İletişim penceresinde görülen portta başka bir donanım bağlı olabilir. Örneğin pencerede görüldüğü halde siz donanımı COM2'ye bağlamış olabilirsiniz.

Turn Serial Off: MicroPro programı kapatılmadan seri portu başka bir donanımın kullanılabilmesi için seri portu serbest bırakmak için kullanılır.

Recent Files: Bu seçenek listede bulunan en son kullanılan programlardan bir tanesini yüklemek için kullanılır.

MPLAB: MPLAB seçildiğinde start ve path olarak iki seçim yapılabilir. MicroPro ekranından MPLAB programını çalıstırmak için Start seçilir. Path ise MPLAB programının klasör yolunu (path) belirlemek için kullanılır.

Programmer Menüsü

Bu menüde ise PIC'i programlama, okuma, karşılaştırma, silme, sigortaları belirleme ve osilatör seçme işlemleri yapılmaktadır. Bu işlemler aşağıda görüldüğü MicroPro ekranından da yapılabilir.

Program: Micropro belleğine yüklenmiş olan program kodlarını PIC'e yüklenmesi için USB portuna gönderir. NOT: Micropro'nun bazı versiyonlarında seri porta gönderir (Örnegin K149A versiyonu hem seri hem de USB portundan programlama yapabilmektedir.)

Read: Chip selector listesinden uygun PIC seçildikten sonra read butonuna tıklandıgında PIC'te yazılı olan program okunur ve MicroPro ekranında görüntüler. İstenirse bu kodlar kaydedilebilir. Eğer PIC kod korumalı (code protected) ise ekranda 0000 verileri görüntülenir.



Verify: Chip selector listesinden uygun chip seçildikten sonra verify read butonuna tıklandığında PIC'te yazılı olan program ile MicroPro ekranındaki program karşılaştırır. Karşılaştırma neticesinde hatalar varsa bu bir pencere içerisinde görüntülenir.

Blank: Menüden blank seçildiginde veya blank butonuna tıklandıgında sekil-A'daki pencere açılır. "Erase Check" seçilirse donanım üzerindeki sokete takılı bulunan PIC'in bos olup olmadıgını kontrol eder, sekil 3 B'deki gibi pencerede silinmiş ve silinmemiş olan yerleri belirtir. Erase chip seçilirse PIC içerisindeki kodları siler, sekil -C'deki gibi silindiğini belirten pencere görülür. Silinmiş bir PIC tekrar okunduğunda ekranda görülen veriler 3FFF'dir.



Reset programmer: Bu seçenek programlayıcı soketine yakılı olan PIC'in MCLR ucunu 200 mS süreyle 0V (Low) yapar. Bu PIC'in reset yapmasına neden olur. Eger reset islemi basarılı olduysa Micropro programında "**Board connected**" mesajı görülür.

Fuses: Bu seçenek PIC'lerin sigorta (fuse) olarak adlandırılan bazı donanımsal özelliklerinin belirlenmesinde kullanılır. PIC programlandığında bu sigorta bilgileri PIC'in bu is için ayrılmıs özel adreslerine yazılır. Fuse seçenekleri, osilatör tipi, watchdog timer'ın açık veya kapalı tutulması, power-up timer'ın aktif veya pasif durumda olusu gibi donanımsal bilgileri içerir. Sigorta seçeneklerinin neler oldugu PIC'ten PIC'e degisir. Bu nedenle kullanılan PIC'in datasheet'i veya bu konuda hazırlanmıs kitaplara basvurmanız tavsiye edilir.

DENEY-2:MİKRODENETLEYİCİ UYGULAMALARI

Konular: program yazma, derleme, pic'e yükleme, Port okuma yazma, pıc16f84a ve pıc16f877a da aynı programı çalıştırmak için gerekenler, döngü kavramı,7 parçalı göstergeye yazma

UYGULAMA-1 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız. Burada verilen programlar pıc16f84a için yazılmıştır. Ancak biz deney setimizdeki hazır kurulu devreyi ve 16f877amikrodenetleyicisini kullanacağız. Böyle basit programlar biraz değişiklikle 16f877a da da çalışır. İlk iki satırı aşağıdaki şekilde değiştiriniz. Böylece yazdığınız program deney setinizdeki 16f877a üzerinde çalışacaktır. Bu iki satır sözde komut olup derleyici içindir.

LIST P=16F877A ;PIC ' in tanıtılması

INCLUDE "P16F877A.INC"; MPASM klasörü içinde bu dosyayı bulunuz, açınız ne işe yaradığını araştırınız Yazdığınız programı uyg1.asm adıyla kaydediniz. MPASMWIN programını kullanarak uyg1.asm'yi derleyiniz. İşlemci türünü 16f877a olarak seçmeyi unutmayınız. Hatasız derlendi ise micropro programını kullanarak uyg1.hex dosyasını deney setindeki mikrodenetleyiciye yükleyiniz. Yüklenmesi bittiğinde portb0'a bağlı led yanacaktır. Portb seçme anahtarı turuncu dip switch portb-led konumunda olmalıdır.

LIST P=16F84A INCLUDE "P16F84A.INC" ;****** AYARLAR ********** CLRF PORTB STATUS.5 BSF CLRF TRISB BCF STATUS.5 ;****** ANA PROGRAM ****** BSF PORTB.0 DONGU GOTO DONGU END

UYGULAMA-2 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız. Uygulama2 de istenenin yapılması için pıc16f877a'da ADCON1registerine O6H yazılması unutulmamalıdır. Bunu da eklersek program aşağıdaki gibi olur. Öncekinde yaptığınız gibi uzantısı. asm olacak şekilde örneğin uyg2.asm adıyla kaydediniz, MPASM kullanarak bu dosyayı derleyiniz. Micropro. exe yi kullanarak deney setindeki 16f877a' ya yükleyiniz. Bu program RA0 butonuna basıldığında RB0'a bağlı ledi yakar.

```
P=16F877A
      LIST
      INCLUDE
                    "P16F877A.INC"
STATUS,5
      BSF
      MOVLW
                    H'00'
                                 ;PORTb nin 8 pininide çıkış yapıyoruz
      MOVWF
                    TRISB
      MOVLW
                    H'06'
                                 ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
      MOVWF
                    ADCON1
                                 ;PORTA nın 6 pininide giriş yapıyoruz
      MOVLW
                    H'3F'
      MOVWF
                    TRISA
      BCF
                    STATUS,5
:****** ANA PROGRAM ******
      CLRF PORTB ;PORTB silindi
OKU
      BTFSS PORTA,0
                                  ; porta nin 0.biti test ediliyor 1 ise alt satır atlanacak
      GOTO OKU
      BSF
             PORTB,0
                                  ; led yakıldı
      GOTO OKU
      END
```

Pull-down anahtarı down konumunda ise(down olmalı) prog.yüklendikten sonra rb0 ledi yanmaz, ra0 a bağlı butona basılırsa rb0 a bağlı led yanar yanık kalır söndürmek için reset butonuna basılmalıdır. Aslında istenen budur.

Pull-down anahtarı pull konumunda ise program yüklendikten sonra rb0 ledi yanar, yanık kalır, söndürmek için resetlemek gerekir, resetleyince söner tekrar yanar çünkü pull konumunda anahtar 5 v a çekilmiştir

ÖDEV1:Bu programda gerekli değişiklikleri yaparak RA0'a basılıp basılı tutuldukça RB0'daki ledi yakan, RA0 bırakıldığında RB0'daki ledi söndüren program haline getiriniz. Hemen yapamazsanız araştırıp haftaya deneyiniz.

Ödev1 in cevabı OKU etiketini CLRF satırına almaktır. Pull-down anahtarı down konumundayken RA0'a basılıp basılı tutulduğunda CLRF PORTB satırına her gelindiğinde portb çok kısa süre için sönecektir ancak süre çok kısa olduğundan bu durum gözlenemeyecektir. Bu kısa süreyi hesaplayınız.

Başka bir çözüm ana programı aşağıdaki gibi yazmak olabilir. Deneyiniz. Başka çözümler üretiniz.

```
;******* ANA PROGRAM ******
BAS CLRF PORTB
OKU
BTFSS PORTA,0
GOTO OKU
BSF PORTB,0
BTFSC PORTA,0
GOTO OKU
GOTO OKU
GOTO BAS
END
```

UYGULAMA-3 Bu program RA0'a bağlı butona **basılıp bırakılınca** PORTB'ye bağlı 8 led de ikilik tabanlı sayıları 1'er arttırır. BASILIP BIRAKILINCA kısmını inceleyiniz. PULL VE DOWN konumunda inceleyiniz. En son sayı FFh artırılınca kaç oluyor.

```
LIST
             P=16F877A
                                        ; LIST P=16F84A
      INCLUDE
                    "P16F877A.INC"
                                        ;INCLUDE "P16F84A.INC"
;******* AYARLAR **********
      BSF
                    STATUS,5
      MOVLW
                    H'00'
                                        ;PORTb'nin 8 pinini de çıkış yapıyoruz
      MOVWF
                    TRISB
      MOVLW
                    H'06'
                                        ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
      MOVWF
                    ADCON1
                                        ;PORTA nın 0. pinini giriş yapıyoruz
      MOVLW
                    H'01'
      MOVWF
                    TRISA
      BCF
                    STATUS,5
;****** ANA PROGRAM ******
      CLRF PORTB
TEST
      BTFSS PORTA,0
      GOTO TEST
OKU
      BTFSC PORTA,0
      GOTO OKU
      INCF PORTB
      GOTO TEST
      END
```

UYGULAMA-4 Uygulama-3 'deki buton yerine gecikme programı kullanarak portb'ye bağlı 8 ledi 1 saniye aralıklarla arttıran program. Yukardakiler gibi derleyiniz, yükleyiniz. Gecikme programının ne kadar zaman tükettiğini hesaplamayı araştırınız. Gecikmeyi alt program olarak düzenlemeyi CALL ve RETURN komutlarını kullanmayı öğreniniz.

LIST P=16F877A INCLUDE "P16F877A.INC" ;******* AYARLAR ********** SAYAC1 EQU H'20' SAYAC2 EQU H'21' SAYAC3 EQU H'22' BSF STATUS,5 H'00' ;PORTb nın 8 pininide çıkış yapıyoruz MOVLW MOVWF TRISB BCF STATUS,5 ;****** ANA PROGRAM ****** CLRF PORTB TEKRAR INCF PORTB H'45' MOVLW MOVWF SAYAC1 DONGU1 H'45' MOVLW MOVWF SAYAC2 MOVWF SATT DONGU2 SAYAC3 DONGU3 DECFSZ SAYAC3, F GOTO DONGU3 DECFSZ SAYAC2, F GOTO DONGU2 DECFSZ SAYAC1, F GOTO DONGU1 GOTO TEKRAR END

Ödev2: PORTA'yı okuyup PORTB'ye yazan programı yazınız. Hemen yapamazsanız araştırıp haftaya deneyiniz.

	LIST	P=16F8	77A	
	INCLUI	DE	"P16F877A.INC"	
;*****	** AYAI	RLAR **	******	
	BSF	STATU	S,5	
	MOVLV	V	H'00'	;PORTb nın 8 pininide çıkış yapıyoruz
	MOVW	F	TRISB	
	MOVLV	V	H'06'	
	MOVW	F	ADCON1	;PORTA nın digital olması için adcon1 e 06h yazıyoruz
	MOVLV	V	H'3F'	;PORTA nın 6 pininide giriş yapıyoruz
	MOVW	F	TRISA	
	BCF	STATU	S,5	
;*****	** ANA	PROGR.	AM *****	
DONGU	J	MOVF	PORTA, W	;PORTA nın değeri working registere alındı
	MOVW	F	PORTB	;w nin değeri portb ye yazıldı
	GOTO	DONGU	J	
	END			

UYGULAMA-5 Aşağıdaki program 7 parçalı göstergeye 8 yazar. Pull-down anahtarı pull konumunda olmalı. Başka sayılar yazmak için tablodaki bilgileri kullanarak gerekli değişikleri yapınız, deneyiniz.

İleri geri sayıcı yapmayı düşününüz.

;7 PARÇALI GÖSTERGEYE 8 YAZAR PULL KONUMUNDA OLMALI

LIST P=16F877A INCLUDE "P16F877A.INC" ;******* AYARLAR ********** CLRF PORTB ;PORTB ' ye bağlı LED ' leri söndür BSF STATUS,5 ;BANK1 ' e geç CLRF TRISB ;PORTB 'nin uçlarını çıkış yap H'FF' MOVLW BCF STATUS,5 ;****** ANA PROGRAM ****** H'7F' ;W YE 7F YAZ BU SAYI 7PARÇALI GÖSTERGEDE 8 DİR MOVLW MOVWF PORTB ;PORTB SEÇME ANAHTARINI 7SEGLERDEN BİRİNE GETİRİNİZ DON GOTO DON END

(Çevrilecek	Çevrilen 7 segment	7 segment	7 segment'te
kod. Hex sayı)	kodu (PORTB'ye)	uçlarındaki veri	görülecek sayı
h'00'	h'3F'	00111111	0
h'01'	h'06'	00000110	1
h'02'	h'5B'	01011011	2
h'03'	h'4F'	01001111	3
h'04'	h'66'	01100110	4
h'05'	h'6D'	01101101	5
h'06'	h'7D'	01111101	6
h'07'	h'07'	00000111	7
h'08'	h'7F'	01111111	8
h'09'	h'6F'	01101111	9
h'0A'	h'77'	01110111	А
h'0B'	h'7C'	01111100	В
h'0C'	h'39'	00111001	С
h'0D'	h'5E'	01011110	, D
h'0E'	h'79'	01111001	E
h'0F'	h'71'	01110001	F
Nokta	h'80'	1000000	•

DENEY-3:MİKRODENETLEYİCİ UYGULAMALARI

Konular: program yazma, derleme, pic'e yükleme, 2 tane7 parçalı göstergeye yazma, tablodan okuma, step motor sürme

UYGULAMA-6 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız.

Yazdığınız programı uyg6.asm adıyla kaydediniz. MPASMWIN programını kullanarak uyg6.asm'yi derleyiniz. İşlemci türünü 16f877a olarak seçmeyi unutmayınız. Hatasız derlendi ise micropro26 programını kullanarak uyg6.hex dosyasını deney setindeki mikrodenetleyiciye yükleyiniz. 7 parçalı göstergenin altındaki dip switch'in üstten 2 ve 3. Anahtarlarını sağa çekerek her iki 7 parçalı göstergeninde devrede olmasını sağlayınız. Aşağıdaki bağlantı şemasını inceleyiniz. Deney setimizdeki bu bağlantıya göre

PIC16F877A'nın a portunun son iki ucu ra1 ve ra0, 7SEG'in sırasıyla 2 ve 1 uçlarına bağlıdır. ra0=1,ra1=0 ise onlar basamağı, ra0=0,ra1=1 ise birler basamağı seçiliyor.

Yüklenmesi bittiğinde 2 tane7 parçalı göstergede 81 yazacaktır. Aşağıdaki incelemeleri yapınız.

+25 yazmak için gerekli değişikliği yapınız. Gecik alt programındaki sayaçların her birinin değerini 01 yapınız, yeniden derleyiniz ve çalıştırınız. Ne görüyorsunuz. Açıklayınız.

+Gecik alt programındaki sayaçların her birinin değerini 45 yapınız, yeniden derleyiniz ve çalıştırınız. Ne görüyorsunuz. Açıklayınız.

+Bu durumu düzeltmek için uygun bir gecikmeden başka bir yol önerebilir misiniz? 7447 entegresini inceleyiniz.

+İkiden fazla sayıda 7 parçalı gösterge olsaydı ne yapılacaktı. Düşününüz.

+Pull-down anahtarını pull konumuna getiriniz. Ra0 ve ra1 butonlarına sırayla basınız. Ne gözlemlediniz, açıklayınız.



ÖDEV 3: UYG6.ASM 'yi kullanarak bu programı 00-99 arasında ileri geri sayan program haline getiriniz. RA4' e bağlı butona her basıldığında sayıyı bir arttıran,RA5'e bağlı butona basıldığında sayıyı bir azaltsın. Alt limiti 00 da üst limiti de 99 da kalsın.

Dönem içinde yapacağınız ödevlerden biri olarak ödev3 ü ve uyg6 hakkında sorulan soruların cevaplarını yazıp getirebilirsiniz.

LIST P=16F877A INCLUDE "P16F877A.INC" ;******** AYARLAR ******** SAYAC1 EQU H'20' ;gecik alt programmin sayaçları tanımlandı SAYAC2 EQU H'21' SAYAC3 EQU H'20' SAYAC3 EQU H'20' BCF STATUS,RP0 BCF STATUS,RP1 (BANK0 a geçildi CLRF PORTA ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı MOVLW H'00' ;PORTb mn 8 pininide çıkış yapıyoruz MOVWF TRISB MOVLW H'06' MOVWF ADCONI ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;********* ANA PROGRAM ******* MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;********* ANA PROGRAM ******* MOVLW B'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTA MOVLW H'06' MOVWF PORTA MOVLW H'06' MOVWF PORTA MOVLW H'06' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'07' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVLW H'07' MOVLW H'07' MOVWF SAYAC1 DONGUI	;uyg6 2 tane 7 parç	alı göstergeye 81 yazm;	ak
INCLUDE "P16F877A.INC" ;******* AYARLAR ******** SAYAC1 EQU H21' SAYAC2 EQU H21' SAYAC3 EQU H22' BCF STATUS.RP0 BCF STATUS.RP1 ;BANK0 a geçildi ;cLRF PORTA ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;pORTA nın digital olması için adconl e 06h yazıyoruz MOVLW H'06' MOVLW B'00111100' ;PORTA nın d pini giriş,ra0 ve ral çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVLW B'00000010' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'06C' MOVLW H'07F' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'17F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVLW H'07F' MOVWF SAYACI DONGU	LIST P=1	6F877A	
;****** AYARLAR ******** SAYAC1 EQU H'20' SAYAC2 EQU H'21' SAYAC3 EQU H'22' BCF STATUS,RP0 BCF STATUS,RP1 :BANK0 a geçildi CLRF PORTA :çıkış veri latchları silinerek porta hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı MOVLW H'00' ;PORTB nın 8 pininide çıkış yapıyoruz MOVWF TRISB MOVLW H'06' MOVUF ADCON1 :PORTA nın digital olması için adeon1 e 06h yazıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;******** ra0a l.ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW B'0000010' MOVLW B'00RTA MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW B'00RTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW H'07F MOVVF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' M	INCLUDE	"P16F877A.INC"	
SAYAC1 EQU H'20' ;gccik alt programmn sayaçları tanımlandı SAYAC2 EQU H'21' SAYAC3 EQU H'21' BCF STATUS.RP0 BCF STATUS.RP1 BCF STATUS.RP1 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BSF STATUS.RP0 BSF STATUS.RP0 BSF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BSF STATUS.RP0 BCF STATUS.RP0 MOVLW H00' MOVWF TRISB MOVLW B00111100' PORTA nın digital olması için adcon l e 06h yazıyoruz MOVWF SATATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 BCF STATUS.RP0 MOVLW BORTA MOVLW	;******* AYARLA	R *****	
SAYAC2 EQU H'21' SAYAC3 EQU H'22' BCF STATUS,RP0 BCF STATUS,RP1 :BANK0 a geçildi CLRF PORTA :çıkış veri latchları silinerek porta hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı CLRF PORTB :gıkış veri latchları silinerek portB hazırlandı DOVLW H'00' :PORTb nın digital olması için adcon1 e 06h yazıyoruz MOVUF TRISB MOVLW H'06' MOVUF TRISA BCF STATUS,RP0 :BANK0 a geçildi ;******** ANA PROGRAM ******* DONGU MOVUF PORTA MOVUF PORTA MOVUF PORTA MOVUF PORTA MOVUF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVUF H'17F' MOVUF PORTB ;w nin değeri portb ye DİRLER BASAMAĞI yazıldı CALL GECİK MOVUF H'17F' MOVUF DORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVUF H'17F' MOVUF H'17F' MOVUF BORTA MOVUF H'17F' MOVUF BORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVUF VATB BORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVUF YATA MOVUF H'17F' MOVUF YATA MOVUF H'17F' MOVUF YATA MOVUF H'17F' MOVUF YATA MOVUF H'17F' MOVUF YATA MO	SAYAC1	EQU H'20'	;gecik alt programının sayaçları tanımlandı
SAYAC3 EQU H'22' BCF STATUS,RP0 BCF STATUS,RP1 ;BANK0 a geçildi CLRF PORTA ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı BSF STATUS,RP0 ;BANK1 seçildi MOVLW H'00' ;PORTb nın 8 pininide çıkış yapıyoruz MOVWF ADCONI ;PORTA nın digital olması için adconl e 06h yazıyoruz MOVWF ADCONI ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;******** ANA PROGRAM ******* mailer in adconl e 06h yazıyoruz MOVWF D'0111100' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVWF B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVWF B'0RTA MOVLW B'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW B'00000001' MOVWF PORTA MOVWF PORTA MOVWF PORTB ;w nin değeri portb ye DİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVLW H'07' MOVWF SAYACI DONGU GECİK MOVLW H'07' MOVWF SAYACI DONGU H'05'	SAYAC2	EQU H'21'	
BCF STATUS,RP0 BCF BCF STATUS,RP1 BANK0 a geçildi CLRF PORTA ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı BSF STATUS,RP0 ;BANK1 seçildi MOVLW H'00' ;PORTb nın 8 pininide çıkış yapıyoruz MOVLW H'00' ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 BCF STATUS,RP0 ;BANK0 a geçildi ;******* WOVF PORTA ;ma0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW H'06' ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı ALL GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVLW H'0F' ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı ;AIL GECİK GOTO DONGU ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı ;AIL GECİK <t< td=""><td>SAYAC3</td><td>EQU H'22'</td><td></td></t<>	SAYAC3	EQU H'22'	
BCF STATUS,RP1 :BANK0 a geçildi CLRF PORTA :çıkış veri latchları silinerek porta hazırlandı CLRF PORTB :çıkış veri latchları silinerek portB hazırlandı BSF STATUS,RP0 :BANK0 a geçildi MOVLW H00' :PORTb nın 8 pininide çıkış yapıyoruz MOVLW H00' :PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVLW B'OOTI :PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'OOTI11100' :PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'OOTI11100' :PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'OOTOR :ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'OOTA MOVLW b'OOTA :w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK :w nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVLW H'OF' :w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK :w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK GOTO DONGU :w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK GONUWF SAYACI DONGU	BCF STA	TUS,RP0	
CLRF PORTA :çıkış veri latchları silinerek porta hazırlandı CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı BSF STATUS, RPO ;BANK1 seçildi MOVLW H'00' ;PORTb nın 8 pininide çıkış yapıyoruz MOVLW H'00' ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVLW B'0011110' ;PORTA nın 4 pini giriş,ra0 ve ral çıkış yapıyoruz MOVWF TRISA ;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW B'00000010' MOVLW B'00000001' ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVLW H'0F' ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı GECİK ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVLW H'0F' MOVLW H'0F' </td <td>BCF STA</td> <td>TUS,RP1</td> <td>;BANK0 a geçildi</td>	BCF STA	TUS,RP1	;BANK0 a geçildi
CLRF PORTB ; ;;kiş veri latchları silinerek portB hazırlandı BSF STATUS,RPO ;BANK1 seçildi MOVLW H'00' ;PORTb nın 8 pininide çıkış yapıyoruz MOVWF TRISB MOVLW B'00111100' ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVLW B'00000010' ;******* ANA PROGRAM ******* DONGU MOVLW b'00000010' MOVLW B'00F MOVLW B'00F MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'0000001' MOVLW B'00000001' MOVLW B'0000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'0000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'0000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' MOVLW B'00000001' B'''''''''''''''''''''''''''	CLRF POR	RTA	çıkış veri latchları silinerek porta hazırlandı
BSF STATUS,RP0 ;BANK1 seçildi MOVLW H'00' ;PORTb nn 8 pininide çıkış yapıyoruz MOVWF TRISB MOVLW H'06' MOVWF ADCON1 ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVWF ADCON1 ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;******* ANA PROGRAM ******* ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVWF PORTB CALL GECİK GOTO DONGU GECİK MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVLW H'07' MOVLW H'06'	CLRF POR	RTB	çıkış veri latchları silinerek portB hazırlandı
MOVLW H'00' :PORTb nn 8 pininide çıkış yapıyoruz MOVWF TRISB	BSF STA	TUS,RP0	;BANK1 seçildi
MOVWF TRISB MOVLW H'06' MOVWF ADCON1 ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVWF ADCON1 ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;******* ANA PROGRAM ******* ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'00000010' MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVLW H'06' MOVLW b'0000001' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'7F' MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVLW H'0F' MOVLW H'0F' MOVLW H'0F' MOVLW H'0F' MOVLW H'0F' MOVLW H'0F' MOVLW H'0F'	MOVLW	H'00'	;PORTb nın 8 pininide çıkış yapıyoruz
MOVLWH'06'MOVWFADCON1;PORTA nın digital olması için adcon1 e 06h yazıyoruzMOVLWB'00111100';PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruzMOVWFTRISA;BANK0 a geçildi;*******stantus,RP0;BANK0 a geçildi;*******stantus,RP0;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyorDONGUMOVLWb'00000010'MOVLWb'00000010';w nin değeri portb ye BİRLER BASAMAĞI yazıldıCALL GECİKMOVLWb'0000001'MOVLWb'0000001';w nin değeri portb ye DİRLER BASAMAĞI yazıldıCALL GECİK;w nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;w nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;w nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;w nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W nin değeri portb ye ONLAR BASAMAĞI yazıldıCALL GECİK;W NIN H'0F'MOVWF;YACIDONGUI;W NIN H'0F'MOVWF;YACIDONGUI;W NIN H'0F'MOVUN H'0F';W YA'N </td <td>MOVWF</td> <td>TRISB</td> <td></td>	MOVWF	TRISB	
MOVWF ADCON1 ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS,RPO ;BANK0 a geçildi ;******* ANA PROGRAM ****** Ta0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'00000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGUI	MOVLW	H'06'	
MOVLW B'00111100' ;PORTA nn 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS,RP0 ;BANK0 a geçildi ;******** ANA PROGRAM ******* ;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK MOVLW H'7F' MOVWF SAYAC1 DONGUI	MOVWF	ADCON1	;PORTA nın digital olması için adcon1 e 06h yazıyoruz
MOVWF TRISA BCF STATUS,RP0 ;******* ANA PROGRAM ****** DONGU MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB CALL GECIK MOVLW b'00000001' MOVWF PORTA MOVLW b'00000001' MOVWF PORTA MOVLW b'00000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB CALL GECIK GOTO DONGU GECIK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVWF NOF	MOVLW	B'00111100'	PORTA nin 4 pini giris,ra0 ve ra1 cikis yapıyoruz
BCF STATUS,RP0 ;BANK0 a geçildi ;******* ;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b00000010' MOVLW PORTA ,wu nin değeri portb ye BİRLER BASAMAĞI yazıldı MOVLW H'06' ,wu nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK ,wu nin değeri portb ye DİRLER BASAMAĞI yazıldı MOVLW H'07F' ,wu nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVWF PORTB ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı GOTO DONGU ;wu nin değeri portb ye ONLAR BASAMAĞI yazıldı MOVLW H'0F' MOVQUY YOY	MOVWF	TRISA	
;******* ANA PROGRAM ****** ;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor DONGU MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'00000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1	BCF	STATUS.RP0	:BANK0 a gecildi
JONGU ;ra0=1,ra1=0 ise onlar,ra0=0,ra1=1 ise birler basamağı seçiliyor MOVLW b'00000010' MOVLW H'06' MOVLW H'06' MOVLW H'06' MOVLW B'0000001' CALL GECİK MOVLW b'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'0000001' MOVLW B'7F' MOVLW H'7F' MOVUF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW MOVLW H'0F' MOVWF SAYAC1 DONGUI MOVLW	:****** ANA PRO	GRAM ******	
DONGU MOVLW b'00000010' MOVWF PORTA MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'00000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVLW H'0F' MOVWF SAYAC1 DONGU1	,	-	:ra0=1.ra1=0 ise onlar.ra0=0.ra1=1 ise birler basamağı seciliyor
MOVLW b'00000010' MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'0000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGUI	DONGU		, , , , , , , , , , , , . , . , . , . , . , . , . , . , . , . , . ,
MOVWF PORTA MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'0000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVLW H'0F' MOVWF SAYAC1 DONGU1	MOVLW	b'00000010'	
MOVLW H'06' MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'0000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1	MOVWF	PORTA	
MOVWF PORTB ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı CALL GECİK MOVLW b'0000001' MOVWF PORTA MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGUI	MOVLW	H'06'	
CALL GECİK MOVLW b'0000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1	MOVWF	PORTB	w nin değeri portb ve BİRLER BASAMAĞI vazıldı:
MOVLW b'0000001' MOVWF PORTA MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVLW H'0F'	CALL GE	CİK	,
MOVUN PORTA MOVUN H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVUN HU0F'	MOVLW	b'00000001'	
MOVLW H'7F' MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVLW H'0F'	MOVWF	PORTA	
MOVUS HYT MOVWF PORTB ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVUW H'0F'	MOVLW	H'7F'	
CALL GECİK GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1	MOVWF	PORTB	w nin değeri north ve ONLAR BASAMAĞI yazıldı
GOTO DONGU GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVLW H'0F'	CALL GEO	TİK	, , min auguri porto ye or (Linte Brior mini for yazılar
GECİK MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVLW H'0F'	GOTO DON	NGU	
MOVLW H'0F' MOVWF SAYAC1 DONGU1 MOVLW HI0F'	GECİK		
MOVER INTER MOVER SAYACI DONGUI	MOVLW	H'0F'	
DONGU1	MOVWF	SAYAC1	
	DONGUI	0111101	
	MOVIW	H'0F'	
MOVWE SAYAC2	MOVWF	SAYAC2	
DONGU2	DONGU2	SHTRE2	
MOVI W H'0F'	MOVI W	H'OF'	
MOVER SAVAC3	MOVWE	SAVAC3	
DONGU3	DONGUS	5/11/105	
DECESZ SAVAC3 E	DECEGZ	SAVAC2 E	
GOTO DONGU3	COTO DO	JATACJ, P NGU3	
	DECEST	SAVAC2 E	
$DUCISL \qquad SATAC2, F$	DECLOT	JATAU2, F	
	DECERT	NUUZ SAVACI E	
$DUCISL SATACI, \Gamma$ $COTO DONGUI$	DECLOT	DONCUI	
RETURN	RETURN	DOMUUI	

RETU END

UYGULAMA-7 Aşağıdaki program STEP motoru ra0 a basılınca(basılıp bırakılınca 1 adım basılı tutunca sürekli) sağa ra1 e basılınca sola döndürür. Setimizde step motor olmadığı için portb ye bağlı ledlerde çıkışı izleyeceğiz. dip switchin en üst anahtarı sağa doğru olmalı. Programı adım adım inceleyiniz. Gerekli açıklamalar program satırlarına yazılmıştır

;DENEY SETİNDE PULL-DOWN ANAHTARI DOWN KONUMUNDA OLACAK ;PULL KONUMUNDADA ÇALIŞIR BU ÇALIŞMA DURUMUNUN FARKINI BULUNUZ.

;step motoru sağa ve sola döndüren program

	LIST	P=16F8	77A	
	INCLUI	DE	"P16F877A.IN	
·******	*** AYA	RLAR **	******	******
	SAYAC1 EQU H'20'			
	SAYAC	2	EQU H'21'	
	SOL	EQU H'	22'	
	SAG	EOU H	23'	
	ADIM	EOU H	24'	
	BCF	STATU	S.RP0	
	BCF	STATU	S.RP1	:BANK0 a gecildi
	CLRF	PORTA		cıkıs veri latchları silinerek porta hazırlandı
	CLRF	PORTB		cıkıs veri latchları silinerek portB hazırlandı
	BSF	STATU	S.RP0	BANK1 secildi
	MOVLV	N	H'00'	PORTb nin 8 pininide cıkıs yapıyoruz
	MOVW	F	TRISB	
	MOVLV	N	H'06'	
	MOVW	F	ADCON1	PORTA nın digital olması için adçonl e 06h yazıyoruz
	MOVLV	N	B'00111111'	PORTA nin 6 pini giris
	MOVW	F	TRISA	
	BCF	STATU	S.RP0	;BANK0 a gecildi
·*****	*** ANA	PROGR	AM *********	***********
,	MOVLV	N	h'07'	
	MOVW	F	ADIM	
	MOVLV	N	B'00000010'	
	MOVW	F	SAG	
	MOVLV	N	B'00000001'	
	MOVW	F	SOL	
;DENE	Y SETİN	DE PUL	L-DOWN ANAI	HTARI DOWN KONUMUNDA OLACAK
,	OKU			
	MOVF		PORTA, W	;PORTAnin değeri w registera alındı. Yani porta okundu. Butona basılınca
				0 oluyor.
				;porta1=1 ve porta0=0, ise sağa; porta1=0 ve porta0=1 ise sola; porta1=1 ve
				porta0=1 ise duracak
	ANDLW	V	B'00000011'	; üst 6 bit maskelendi.
	XORWI	Ę	SAG,W	w reg yani porta dan okunan değer SAG ile aynıysa sonuç 0, zero bayrağı;
				1 olur
	BTFSC		STATUS,Z	; z bayrağı 0 ise alt satırı atla diğer seçeneğe bak
	GOTO		SAGADON	
	MOVF		PORTA, W	; W deki değer bozuldu porta tekrar okunur
	ANDLW	V	B'00000011'	; üst 6 bit maskelendi.
	XORWF	7	SOL,W	;w reg yani porta dan okunan değer SOL ile aynıysa sonuç 0, zero bayrağı 1
				olur
	BTFSC		STATUS,Z	; z bayrağı 0 ise alt satırı atla başa git
	GOTO		SOLADON	
	GOTO		OKU	;01 veya 10 dışında ne gelirse gelsin(00 ve 11) porta yı okumaya devam
				eder

SAGADON		
INCF	ADIM	
GOTO	DEVAM	
SOLADON		
DECF	ADIM	
DEVAM		
MOVF	ADIM,W	; adım değeri w ye alındı
ANDLW	B'00000111'	; adım değeri 7 den büyük olamaz(0 dahil 8 adım). üst 5 bit maskelendi
CALL	ADIMTBL	
MOVWF	PORTB	
CALL	GECIKME	
GOTO OKU		;butonların durumu değişti mi kontrol edilecek
ADIMTBL		
ADDWF	PCL,F	;program sayacının bu andaki değerine w deki değer eklenecek
RETLW	B'00000001'	;w=0 ile gelirse w=01 ile döner
RETLW	B'00001001'	
RETLW	B'00001000'	
RETLW	B'00001010'	
RETLW	B'00000010'	
RETLW	B'00000110'	
RETLW	B'00000100'	
RETLW	B'00000101'	;7.satır,w=07 ile gelirse w=05 ile döner
GECIKME		
MOVLW	H'FF'	
MOVWF	SAYAC1	
DONGU1		
MOVLW	H'FF'	
MOVWF	SAYAC2	
DONGU2		
DECFSZ	SAYAC2,F	
GOTO DONGU	J2	
DECFSZ	SAYAC1,F	
GOTO DONGU	J1	
RETURN		
END		

ÖDEV4. UYGULAMA 7 yi kullanarak bir step motor sürücü ve step motor ekleyerek projeyi tamamlayınız. Dönem içinde yapacağınız ödevlerden biri olarak ödev4 ü ve uyg7 hakkında sorulan soruların cevaplarını yazıp getirebilirsiniz.

DENEY-4:MİKRODENETLEYİCİ UYGULAMALARI

Konular: ödev3 olarak verilen 2 tane7 parçalı gösterge ile 00-99 ileri-geri sayıcı, yeni bir ödev, eksik tamamlamak için serbest çalışma.

ÖDEV 3: UYG6.ASM 'yi kullanarak bu programı 00-99 arasında ileri geri sayan program haline getiriniz. RA4' e bağlı butona her basıldığında sayıyı bir arttıran,RA5'e bağlı butona basıldığında sayıyı bir azaltsın. Alt limiti 00 da üst limiti de 99 da kalsın.

;0-99 ileri GERİ sayıcı PORTA4 BASILIP BIRAKILINCA 1 ARTAR.

;PORTA5 E BASILIP BIRAKILINCA 1 AZALIR

;ARTIRMAYA DEVAM EDİLİRSE 99 DA KALIR AZALTMAYA DEVAM EDİLİR 00 DA KALIR ;7447 LİSİ S99ILGER.ASM DİR. O PROGRAM 16F84A İÇİNDİ. SİZE VERİLEN CD DE BULABİLİRSİNİZ

;DENEY SETİNDE 7 PARÇALI GÖST. ALTINDAKİ DİP SWİTCH 2 VE 3. ANAHTARLAR SAĞA ÇEKİLECEK.

;ALTTAKİ PULL-DOWN PULL KONUMUNDA OLACAK

UYGULAMA 8

LIST P=16F877A INCLUDE "P16F877A.INC" ;****** AYARLAR ********* BIRLER EQU H'20' ONLAR EQU H'21' SAYAC1 EOU H'22' SAYAC2 EQU H'23' SAYAC3 EOU H'24' BCF STATUS, RP0 STATUS, RP1 ;BANK0 a geçildi BCF ;çıkış veri latchları silinerek porta hazırlandı CLRF PORTA CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı STATUS, RP0 ;BANK1 secildi BSF ;PORTb nin 8 pininide çıkış yapıyoruz MOVLW H'00' MOVWF TRISB MOVLW H'06' ;PORTA nın digital olması için adcon1 e 06h yazıyoruz MOVWF ADCON1 MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz MOVWF TRISA BCF STATUS, RP0 ;BANK0 a geçildi :****** ANA PROGRAM ****** CLRF BIRLER CLRF ONLAR CLRF PORTB TEST4 ;BASILI İSE 0 GELMELİ PULL KONUMU **BTFSS PORTA,4** GOTO BASIL4 GOTO TEST5 BASIL4 BTFSS PORTA.4 GOTO BASIL4 GOTO ARTTIR TEST5 **BTFSS PORTA,5** GOTO BASIL5 GOTO YY BASIL5 BTFSS PORTA.5 GOTO BASIL5 GOTO AZALT YY CALL YAZ

GOTO TEST4 ARTTIR INCF BIRLER D'10' MOVLW SUBWF BIRLER,W BTFSC STATUS,2 GOTO DUZEN1 CALL YAZ GOTO TEST4 DUZEN1 CLRF BIRLER INCF **ONLAR** MOVLW D'10' ;ONLAR BASAMAĞI 10 OLDUYSA TEKRAR 9 YAPIYORUZ SUBWF ONLAR,W ;0 DAN 9 A KADARSA YAZDIRIYORUZ BTFSC STATUS,2 ;BİRLER BASAMAĞI 10 OLDUYSA TEKRAR 9 YAPIYORUZ GOTO DUZEN2 CALL YAZ GOTO TEST4 DUZEN2 MOVLW D'9' MOVWF BİRLER VE ONLAR 99 DA KALIYOR BIRLER MOVWF **ONLAR** CALL YAZ TEST4 GOTO ;azaltma kısmı arttırma kısmına benzer olarak azalt DECF BIRLER MOVLW H'FF' **SUBWF** BIRLER,W BTFSC STATUS,2 GOTO DUZE1 YAZ CALL GOTO TEST4 DUZE1 MOVLW D'9' MOVWF BIRLER DECF ONLAR ;00 BİR AZALIRSA FF OLUR MOVLW H'FF' ;ONLAR BASAMAĞI FF OLDUYSA TEKRAR 0 YAPIYORUZ SUBWFONLAR,W **BTFSC STATUS,2** :0 DAN 9 A KADARSA YAZDIRIYORUZ GOTO DUZE2 BİRLER BASAMAĞI FF OLDUYSA TEKRAR 0 YAPIYORUZ CALL YAZ GOTO TEST4 DUZE2 MOVLW D'0' MOVWF ;BİRLER VE ONLAR 00 DA KALIYOR BIRLER MOVWF **ONLAR** CALL YAZ GOTO TEST4 **CEVTBL** ADDWF PCL,F ;program sayacının bu andaki değerine w deki değer eklenecek RETLW H'3F' ;w=0 ile gelirse w=3F ile döner RETLW H'06' RETLW H'5B' RETLW H'4F' RETLW H'66' RETLW H'6D' RETLW H'7D' RETLW H'07'

RETLW H'7F' RETLW H'6F' ;w=9 ile gelirse w=6F ile döner YAZ MOVLW b'00000010' MOVWF PORTA MOVF BIRLER,W CALL CEVTBL MOVWF ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı PORTB CALL GECİK MOVLW b'00000001' MOVWF PORTA MOVF ONLAR,W CALL CEVTBL MOVWF ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı PORTB CALL GECİK RETURN ;*** GECİK ALT PROGRAMI ***************** GECİK MOVLW H'15' MOVWF SAYAC1 DONGU1 MOVLW H'15' MOVWF SAYAC2 DONGU2 MOVLW H'05' MOVWF SAYAC3 DONGU3 DECFSZ SAYAC3, F GOTO DONGU3 DECFSZ SAYAC2, F GOTO DONGU2 DECFSZ SAYAC1, F GOTO DONGU1 RETURN END **UYGULAMA 9** LIST P=16F877A INCLUDE "P16F877A.INC" MOVLW B'10000001' MOVWF ADCON0 STATUS, RP0 BSF MOVLW H'00' MOVWF TRISB MOVLW B'000001110' MOVWF ADCON1 BCF STATUS, RP0 BASLA BSF ADCON0,GO TEST BTFSC ADCON0,GO GOTO TEST MOVF ADRESL,W MOVWF PORTB GOTO BASLA **END**

Ödev 5 Yukarıdaki programın ne iş yaptığını16f877a nın ADCON0 ve ADCON1 adlı özel amaçlı kaydedicilerini inceleyerek anlayınız. ADRESL nedir?

DENEY-5:MİKRODENETLEYİCİ UYGULAMALARI

Konular: ödev5 olarak verilen UYGULAMA 8 - analog-dijital dönüştürücü, ADC(Analog Digital Converter), RB0 Harici kesmesi,TIMER0 ödevi

UYGULAMA 10

UYGULAMA 9 HATALARI DÜZELTTİM. BU SEÇİMDE ADRESH EN ÖNEMLİ 8 BİTİ PORTB YE YAZIYORUZ.rb0 a bir potansiyometreden analog giriş uygulayınız. pot yoksa rb0 a iki tane en az 1k lık direnç kullanarak 5v ve 2.5v uygulayınız. dirençte yoksa 5v(vcc) uygulayınız

LIST P=16F877A INCLUDE "P16F877A.INC" BCF STATUS, RP1 BCF STATUS, RP0 ; adcon1 deki ads2, 6.bit 0 ve adcon0 daki adcs1=1 ve adsc0 = 0MOVLW B'10000001' seçildiğinden ; clock fosc/32 seçilmiş oldu. 5.4.ve 3. Bitler 0 seçildiğinden kanal0=an0=ra0 seçildi.bit0=0 a/d dön.kapalı MOVWF ;BANK0 DA ADCON0 BCF STATUS, RP1 BSF STATUS, RP0 ;BANK1 MOVLW H'00' MOVWF TRISB :BANK1 DE ;ADRESH=000000XX ADRESL=XXXXXXXX ;Alttakini Kullandık MOVLW B'10001110' MOVLW B'00001110' ;ADRESH=XXXXXXX ADRESL=XX000000 MOVWF ADCON1 ;BANK1 DE BCF STATUS, RP1 BCF STATUS, RP0 ;BANK0 DAYIZ BASLA ADCON0.GO BSF ;BANK0 DA,,,analog digital dönüşüm başlasın TEST BTFSC ADCON0,GO ;BANK0 DA,,, bu bit sıfırsa dönüşüm tamamlanmış demektir GOTO TEST MOVF ADRESH,W :BANK0 DA..EN ÖNEMLİ 8 BİTİ GÖSTERİR MOVWF PORTB :BANK0 DA GOTO BASLA END **ÇOK BENZİYOR KARIŞMASIN DİYE TEKRAR YAZDIM. Farkın ne olduğuna dikkat ediniz.**

ÇOK BENZIYOR KARIŞMASIN DIYE TEKRAR YAZDIM. Farkın ne olduğuna dıkkat ediniz. ;UYGULAMA 8 HATALARI DÜZELTTİM BU SEÇİMDE ADRESL EN ÖNEMSİZ 2 BİTİ PORTB

YE YAZIYORUZ

	LIST P=16F877A		
	INCLUE	DE "P16F877A.INC"	
	BCF STA	ATUS,RP1	
	BCF STA	ATUS,RP0	
	MOVLW B'10000001'		; adcon1 deki ads2, 6.bit 0 ve adcon0 daki adcs1=1 ve adsc0 = 0 seçildiğinden ;clock fosc/32 seçilmiş oldu. 5.4.ve 3. Bitler 0 seçildiğinden kanal0=an0=ra0 seçildi.bit0=0 a/d dön.kapalı
	MOVWF ADCON0		;BANK0 DA
	BCF STA	ATUS,RP1	
	BSF STA	ATUS,RP0	;BANK1
	MOVLW H'00'		
	MOVWF TRISB		;BANK1 DE
	MOVLW B'00001110'		;ADRESH=XXXXXXXX ADRESL=XX000000
	MOVWF ADCON1		;BANK1 DE
	BCF STA	ATUS,RP1	
	BCF STATUS,RP0		;BANK0 DAYIZ
BASLA			
	BSF ADCON0,GO		;BANK0 DA,,,analog digital dönüşüm başlasın

TEST

BTFSC ADCON0,GO	;BANK0 DA,,,,bu bit sıfırsa dönüşüm tamamlanmış demektir
GOTO TEST	
BCF STATUS,RP1	
BSF STATUS,RP0	;BANK1 DEYIZ
MOVF ADRESL,W	;BANK1 DE. EN ÖNEMSİZ 2 BİTİ GÖSTERİR
BCF STATUS,RP1	
BCF STATUS,RP0	;BANK0 DAYIZ
MOVWF PORTB	;BANK0 DA
GOTO BASLA	
END	

UYGULAMA 11 RB0 HARİCİ KESMESİ. Programın üstüne yazdığım açıklamaları uygulayınız

;Deney setinde rb0 ucu bir tel ile vcc ye bağlanır. ;rb0 ucu 5 v dan ayrılıp gnd a bağlandığında kesme gelir ve rb7 ye bağlı led yanar ve öyle kalır tekrar çalıştırmak için rb0 ucu vcc ye bağlanır deney seti resetlenir; tekrar rb0 ucu sıfıra çekildiğinde kesme oluşur...

```
LIST
            P=16F877A
      INCLUDE
                  "P16F877A.INC"
;******* AYARLAR **********
      ORG H'000'
      GOTO ANAPROG
      ORG H'004'
      GOTO RB0KES
ANAPROG
      CLRF PORTB
      MOVLW
                  B'10010000'
                               ;GIE VE RB0 KESMESİNİ AÇ(ENABLE)
      MOVWF
                  INTCON
                        ;BANK1 E GEÇ
      BSF
            STATUS, RP0
      CLRF OPTION REG :KESME DÜSEN KENARDA 6.BİT 0
                               ;PORTB NİN 0.PİNİ GİRİŞ RB0 KESMESİ İÇİN, DİĞERLERİ ÇIKIŞ
      MOVLW
                  b'00000001'
      MOVWF
                  TRISB
      BCF
            STATUS, RP0
DÖNGÜ
      GOTO DÖNGÜ
                               ;SONSUZ DÖNGÜDE KESME BEKLER
RB0KES
            INTCON,INTF ;RB0 HARİCİ KESME BAYRAĞINI İNDİR SIFIR YAP
      BCF
                               ;RB7 Yİ BİR YAP LED YANAR
      BSF
            PORTB,7
                         ;KESME ALTPROGRAMINDAN DÖN
      RETFIE
      END
```

ÖDEV 6 UYGULAMA 4 de ki iç içe 3 çevrimli gecikme programının kaç milisaniye(veya kaç peryot) gecikme yaptığını hesaplayınız. Aynı gecikmeyi yapan bir altprogramı TIMER0'ı sayaç olarak kullanarak yazınız.

DENEY-6:MİKRODENETLEYİCİ UYGULAMALARI

Konula	ar: TIM	ER0'ın	sayaç olarak k	ullanılması, TIN	/IER0 Kesmesi
UYGU	LAMA	12			
;TIME	R0 in sag	yaç olar	ak kullanılması		
	LIST	P=16F8	77A		
	INCLU	DE "P16	5F877A.INC"		
	CON	FIG h'3F	'31' ;CONFIG t	5'11111100110001	';CONFIG _CP_OFF &_WDT_OFF & _PWRT_ON &
_XT_O	SC &				
					;_BODEN_OFF &_LVP_OFF & _CPD_OFF
					;configürasyonu yukarıda verdik micropro26 sormayacak
	BCF	STATU	S,RP1		
	BSF	STATU	S,RP0		;BANK1 E GEÇTİK
	MOVL	W	B'11010111'	; Prescaler(ön ölç	ekleme) timer0 a ait. 1/256, B'xxxx0111' x ler önemsiz
	MOVW	F	OPTION_REG		
	CLRF	TRISB			;BANK1 DE
	BCF	STATU	JS,RP1		
	BCF	STATU	S,RP0		;BANK0 DAYIZ
	CLRF	PORTB	5		
YAK					
	BSF	PORTB	5,0		;portb0 daki led yanar
	CALL	GECİK	ME		
SONDU	JR				
	BCF	PORTB	5,0		;portb0 daki led söner
	CALL	GECİK	ME		
	GOTO	YAK			
·*****	*gecikm	e alt prog	gramı timer0 ı say	aç olarak kullanıy	0r*****
GECİK	ME				
	CLRF	TMR0		;sayıcı 00h d	en itibaren saymaya başladı. 256 peryodu 1 diye sayıyor
TESTB	İT				
	BTFSS	TMR0,	7	;timer0 in 7.1	piti 1 olduysa döngüden çıkıyor, değilse devam ediyor
	GOTO	TESTB	İT		
	RETUR	N			
	END				
BTFSS	5 TMR0	.7 satırıı	11 6 ve 5 vaparal	k tekrar derlevini	z ve programı denev setinde calıstırınız.

BTFSS TMR0,7 iken CLRF TMR0 satırı yerine MOVLW H'xx' ve MOVWF TMR0 satırlarını yazıp xx yerine 22, 33 ve 44 yazarak deneyiniz. Ne gözlüyorsunuz?

UYGULAMA 13

;TIMER0 kesmesi LIST P=16F877A INCLUDE "P16F877A.INC" __CONFIG h'3F31' ;__CONFIG b'11111100110001' ;__CONFIG _CP_OFF &_WDT_OFF & _PWRT_ON & _XT_OSC & ;_BODEN_OFF &_LVP_OFF & _CPD_OFF ;configürasyonu yukarıda verdik micropro26 sormayacak

;******* AYARLAR ********* ORG H'000'

GOTO ANAPROG

ORG H'004'

GOTO TMR0KES

ANAPROG

CLRWDT

BSF STATUS, RP0 ; BANK1 E GEÇ

	MOVLW		B'11010111'	;tmr0 siny kayn dahili, Prescaler(ön ölçekleme) timer0 a ait. 1/256,
	MOWWE		OPTION REG	B XX0X0111 X let ohenisiz
	MOVLV	V	B'10100000'	:GIE VE TMR0IE KESMESİNİ AC(ENABLE)
	MOVWF		INTCON	,,,, (,)
	CLRF	TRISB		
	BCF	STATU	S,RP0	
	CLRF	PORTB		
	•			
DONGL	J			
	GOTO	DÖNGÜ	j	;SONSUZ DÖNGÜDE KESME BEKLER
TMR0K	ES			
	BCF	INTCO	N,TOIF ;TMR(IF kesme bayrağını indir sıfır yap. türkçe tabloda TMR0IF yazıyor yanlış
	MOVLW		H'01'	;1 bu üç satırda ne yapıldığını inceleyiniz
	ANDWF		PORTB,F	;2
	XORWF		PORTB,F	;3
	CLRF TMR0			;timer0 yeniden 00h den başlatılarak 256 peryotta 1 saydırılıyor
	RETFIE			;KESME ALTPROGRAMINDAN DÖN
	END			

Bu programda uygulama 11 de verdiğim programla benzer işi yapar. Ancak bu defa timer0 kesmesi kullanılmıştır.

MOVLW B'11010111' satırını MOVLW B'11010110' olarak değiştirerek deneyiniz. Benzer şekilde TMR0KES içindeki **CLRF TMR0** yerine uygulama 10 daki gibi MOVLW H'xx' ve MOVWF TMR0 satırlarını yazıp xx yerine 22, 33 ve 44 yazarak deneyebilirsiniz.

Bu bilgileri ve sonuçları kullanarak deney4 de verilen ödev6 yı tekrar sorgulayınız.

DENEY-7:MİKRODENETLEYİCİ UYGULAMALARI

```
Konular: yazılımla pwm üretilmesi
UYGULAMA 14
;o.altınbaşak kitaptaki program 34 ün 877 ye uyarlanmış hali
      LIST
            P=16F877A
                   "P16F877A.INC"
      INCLUDE
       _CONFIG h'3F31' ;__CONFIG b'11111100110001' ;__CONFIG _CP_OFF &_WDT_OFF & _PWRT_ON &
_XT_OSC &
;_BODEN_OFF &_LVP_OFF & _CPD_OFF
;****** AYARLAR *********
IS
            H'20'
      EQU
      EQU
            H'21'
PER
ISYED EQU
            H'22'
SAYAC1
             EQU
                   H'23'
SAYAC2
             EQU
                   H'24'
      CLRF
            PORTB
            STATUS,5
      BSF
      CLRF TRISB
                   H'06'
      MOVLW
      MOVWF
                   ADCON1
                                ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
      MOVLW
                   H'FF'
      MOVWF
                   TRISA
      BCF
            STATUS,5
;****** ANA PROGRAM ******
BASLA
      MOVLW
                   D'25'
      MOVWF
                   IS
      MOVWF
                   ISYED
DONGU
                   ISYED,W
      MOVF
      MOVWF
                   IS
      MOVLW
                   D'255'
      MOVWF
                   PER
      BSF
                   PORTB,0
                   PORTA,1
                                :PULL UP-DOWN ANAHTARI PULL OLACAK
      BTFSS
      GOTO
                   M2
      BTFSS
                   PORTA,2
      GOTO
                   M3
PWM0
      DECFSZ
                   IS,F
      GOTO PWM1
      BCF
            PORTB,0
PWM1
      DECFSZ
                   PER,F
      GOTO
                   PWM0
      GOTO
                   DONGU
M2
      CALL
                   GECİKME
      MOVF
                   ISYED,W
      MOVWF
                   IS
      INCF
                   IS,F
      MOVF
                   IS,W
      MOVWF
                   ISYED
      GOTO
                   DONGU
M3
                   GECİKME
      CALL
      MOVF
                   ISYED,W
      MOVWF
                   IS
      DECF
                   IS,F
```

VF	IS,W
VWF	ISYED
O DONGL	J
VLW	H'0F'
VWF	SAYAC1
VLW	H'FF'
VWF	SAYAC2
EFSZ	SAYAC2,F
O D2	
EFSZ	SAYAC1,F
O D1	
URN	
)	
	VF VWF O DONGU VLW VWF VSSZ O D2 SFSZ O D1 URN

Ödev:

Programın çalışması tatmin edici değildir. İyileştirmek için uğraşınız. Varsa hataları belirleyiniz. Kendiniz görev peryodu %25 ve %35 olan pwm programı yazınız. 877 nin pwm modülünü inceleyiniz.

DENEY-8:MİKRODENETLEYİCİ UYGULAMALARI

Konular: yazılımla pwm üretilmesi yeni bir program **UYGULAMA 15** LIST P=16F877A "P16F877A.INC" INCLUDE _CONFIG h'3F31' ; __CONFIG b'11111100110001' ; __CONFIG _CP_OFF & _WDT_OFF & _PWRT_ON & _XT_OSC & ;_BODEN_OFF &_LVP_OFF & _CPD_OFF ;****** AYARLAR ********** EQU SAYAC1 H'22' SAYAC2 EQU H'23' DUTY EQU H'24' KALANEQU H'25' :DUTY(1 DE KALMA.GÖREV) SÜRESİ İCİN ÜST SINIR USTSINEOU H'26' ;DUTY SÜRESİ İÇİN ALT SINIR ALTSIN EQU H'27' BSF STATUS,5 MOVLW H'00' ;PORTb nin 8 pininide çıkış yapıyoruz MOVWF TRISB H'06' MOVLW MOVWF ;PORTA nın digital olması için adcon1 e 06h yazıyoruz ADCON1 ;PORTA nin 6 pininide giris yapıyoruz MOVLW H'3F' MOVWF TRISA BCF STATUS,5 ;BANK0 a geri dön ;****** ANA PROGRAM ****** CLRF PORTB MOVLW H'01' MOVWF ALTSIN SUBLWH'FE' ;W=L-W,YANİ W=255-ALTSIN = ÜSTSINIR MOVWF USTSIN MOVLW H'FE' MOVWF DUTY ;DUTY BAŞLANGIÇ DEĞERİ ;**BASBIRAK BASLANGIC TEST **BTFSS PORTA,1** ;PUL-UP ANAHTARI UP BASILINCA 0 GELİYOR BASILMAYINCA 1 GOTO BIRAK1 **BTFSS PORTA,2** GOTO BIRAK2 GOTO YAZ **BIRAK1** BTFSC PORTA,1 ;BUTON BIRAKILINCA 1 GELECEK ALT SATIRA GEÇECEK GOTO ART GOTO BIRAK1 ; BUTON BIRAKILINCAYA KADAR BIRAK1 E GİDECEK ART MOVF USTSIN.W ;SONUÇ W DE DUTY Yİ BOZMAMAK İÇİN XORWF DUTY,W BTFSC STATUS,Z ;DUTY FEH İSE Z=1 OLUR GOTO SINIRÜ ;ÜST SINIR DEĞERİNE GİT INCF DUTY,F ;DUTY USTSINIR OLMAMIŞ ARTSIN GOTO YAZ SINIRÜ MOVF USTSIN.W ;DUTY ÜSTSINIRA SABİTLENDİ MOVWF DUTY GOTO YAZ BIRAK2 **BTFSC PORTA,2** ;BUTON BIRAKILDIMI GOTO AZAL

GOTO BIRAK2 AZAL MOVF ALTSIN,W ;SONUÇ W DE DUTY Yİ BOZMAMAK İÇİN XORWF DUTY,W ;DUTY ALSINIRA ULAȘTI İSE İSE Z=1 OLUR BTFSC STATUS,Z ;ALT SINIR DEĞERİNE GİT SINIRA GOTO DECF DUTY,F ;DUTY ALT SINIR OLMAMIŞ AZALSIN GOTO YAZ **SINIRA** MOVF ALTSIN,W ;DUTY ALTSINIRA SABİTLENDİ MOVWF DUTY GOTO YAZ YAZ MOVF DUTY,W MOVWF SAYAC1 MOVLW H'01' MOVWF PORTB CALL GECİK MOVF DUTY,W ;W=DUTY SUBLWH'FF' ;W=L-W,YANİ W=255-DUTY KALAN SIFIR SÜRESİ MOVWF SAYAC1 MOVLW H'00' MOVWF PORTB CALL GECİK GOTO TEST GECİK DONGU1 MOVLW H'01' MOVWF SAYAC2 DONGU2 DECFSZ SAYAC2,F GOTO DONGU2 DECFSZ SAYAC1,F GOTO DONGU1 RETURN **END**

Programi derleyip çalıştırınız. Bir voltmetre ile portb0 ve GND arasını DC olarak ölçünüz. Porta1 e bağlı buton arttırma butonudur. Bu butona bastıkça portb0 a bağlı ledin parlaklığı artacaktır. Ancak programın baş kısmında DUTY başlangıç değeri FE olarak verildiğinden led en parlak durumundadır. Yine de arttırmak için porta1 e bağlı butona basıp bırakırsanız voltmetrede ve led deki durumun değişmediğini izleyebilirsiniz. Porta2 azaltma butonuna basıldığında voltmetreden 0.017v bir azalma olduğunu ölçebilirsiniz. 5v/255=0.019v olduğuna göre ölçtüğünüz 0.017v u hesaplamaya çalışınız.

INCFDUTY,Fbulunduğu satıra 4 dört kere alt alta yazınız ve benzer şekildeDECFDUTY,Fbulunduğu satıra 4 dört kere alt alta yazınız. Derleyip çalıştırınız. 0.07V luk artma veazalmayıbelirleyiniz.0.07 v u hesaplayınız. Programda uygun değişikliği yaparak 0.17 voltluk adımlarlaarttırma ve azalma yapar hale getiriniz. Burada oluşabilecek aksaklıkları ve çözümlerini bulmaya çalışınız

Tekrar orijinal hale dönünüz. Bu defa GECİK alt programında sayac2 ye yazdırılan 01 sayısı yerine 55h yazınız. Programı derleyip çalıştırınız. Ne gözlemlediniz.

DENEY-9- 8085 SİMÜLATÖR KULLANIMI-1

UYGULAMA-1a: Aşağıdaki programı Sim8085 simülatöründe yazın. Dosyayı uyg1 ismiyle kaydedin ve derleyin. Program hafizasına yerleşen op- code'ları yandaki tabloya yazın. Bu hex değerleri komut tablosu kâğıdından okuduğunuz değerlerle karşılaştırın.		Adres	op-code	•
MVI B,30H MVI C,39H MOV A,B SUB C OUT 20H HLT				

UYGULAMA-1b:

uyg1 isimli programı adım adım çalıştırarak B, C ve A kaydedicilerindeki değerlerin değişimini gözleyin. MOV (taşı) ve SUB (çıkar) komutunun yürütülmesinin ardından bayrakların nasıl değiştiğini inceleyin. S ve CY bayraklarının neden 1 olduğunu düşünün. 20H nolu IO portuna hangi değerin yazıldığına bakın.

UYGULAMA-2a:

Aşağıdaki programı Sim8085 simülatöründe yazın. Dosyayı uyg2 ismiyle kaydedin ve derleyin. Program hafizasına yerleşen opcode'ları yandaki tabloya yazın. Bu hex değerleri komut tablosu kâğıdından okuduğunuz değerlerle karşılaştırın. Son satırdaki JMP komutuna karşılık gelen opcode'u dikkatlice inceleyin.

LDA 0040H CMA STA 0041H DUR: JMP DUR

Adres	op-code
	-
_	
<u> </u>	
	1

UYGULAMA-2b:

uyg2 isimli programı adım adım çalıştırın ve sonucu gözleyin. Hafizanın 0040H adresine 12H sayısını yükleyerek programı tekrar çalıştırın. LDA ve CMA komutlarının yürütülmesinin ardından akümülatöre hangi değerin yüklendiğini görün. STA komutunun çalışma şeklini inceleyin.

UYGULAMA-3:

Aşağıdaki program 0200H adresinden itibaren 20 adet hafiza hücresine 00H yazar, yani hafiza içeriğini sıfırlar. Programı yazıp derleyin. Hafizanın ilgili bölgesine rasgele 20 sayı yazıp programı adım adım çalıştırın ve sonucunu gözleyin. HL kaydedicisindeki değerin nasıl değiştiğini inceleyin.

MVI A.00H			
LXI H.0200H			
MVI B.14H	···	·	
GIT:MOV M.A			
INX H			
DCR B			
JNZ GIT			
HLT			•

UYGULAMA-4:

Aşağıdaki program hafizanın 0400H adresindeki sayı ile 0401H adresindeki sayıyı toplar ve sonucu 0402H adresine yazar. Programı yazıp derleyin. Hafizaya iki sayı yükleyip programı çalıştırın ve sonucu gözleyin.

LXI H,0400H MOV A,M INR L ADD M INR L MOV M,A HLT

Aynı programda INR L yerine INX H komutunu yazın ve ilk satırı LXI H,04FEH olarak değiştirin. Birinci sayıyı 04FEH adresine, ikinci sayıyı 04FFH adresine yükleyip programı çalıştırın ve toplam sonucunun hangi adrese yüklendiğini gözleyin.

UYGULAMA-5:

Hafizanın 2050H adresinde bulunan sayıyı akümülatöre yüklemek için aşağıda 4 farklı 8085 kodu yazılmıştır. Simülatörü kullanarak her bir yöntemin kaç saat periyodunda (clock cycle) işlendiğini tespit edin. Komut tablosu kâğıdındaki T-durumları ile kıyaslayın.

1. yöntem	2. yöntem	3. yöntem	4. yöntem		
LDA 2050H	LXI B,2050H LDAX B	LXI H,2050H MOV A,M	MVI H,20H MVI L,50H MOV A.M		

DENEY-10: 8085 SİMÜLATÖR KULLANIMI-2

UYGULAMA-1a: Zaman gecikmesi oluşturma

Aşağıdaki programı Sim8085 simülatöründe yazın. Dosyayı zaman1 ismiyle kaydedin ve derleyin. Programı RUN komutu ile çalıştırıp toplam saat periyodu (clock cycle) değerini ekrandaki T-states adlı kısımdan okuyun. Bu sonucu kendi hesapladığınız periyot değeri ile kıyaslayın. (Not: Simülatörde süre hesabı için $T_{esc}=0.325\mu s$ olarak alınmalıdır.)

MVI C,FFH TEKRAR: ĎCR C JNZ TEKRAR

UYGULAMA-1b: 1ms zaman gecikmesi oluşturma

1ms'lik gecikme süresi elde etmek için C kaydedicisine hangi değerin yüklenmesi gerektiğini düşünün. Gerekli hesaplamaları aşağıdaki formüle göre yapın ve bu değere göre programı tekrar çalıştırarak sonucun tanı 1ms (1000µs) olup olmadığını görün.

C'deki değer n ise toplam saat periyodu=7+(4+10).(n-1)+4+7=14n+41ms gecikme için periyot sayısı 1ms/0.325µs=3076.92 olmalıdır. 14n+4=3076.92 ise n=219.49 olur.

C'ye 219 (DBH) yüklenirse periyot sayısı 3070 ve süre=997.75µs elde edilir. C'ye 220 (DCH) yüklenirse periyot sayısı 3084 ve süre=1002.3µs elde edilir.

UYGULAMA-2a: 16 bitlik sayı ile elde edilecek en büyük zaman gecikmesi.

Aşağıdaki programı Sim8085 simülatöründe yazın. Dosyayı zaman2 ismiyle kaydedin ve derleyin. Programı RUN komutu ile çalıştırıp toplam saat periyodu (clock cycle) değerini ekrandaki T-states adlı kısımdan okuyun. Bu sonucu kendi hesapladığınız periyot değeri ile kıyaslayın. (Not: Simülatörde süre hesabı için $T_{osc}=0.325\mu s$ olarak alınmalıdır.)

LXI D,FFFFH TEKRAR: DCX D MOV A,D ORA E JNZ TEKRAR

Sonuc: 1572847 periyot. Süre=511.175ms

UYGULAMA-2b: 16 bitlik sayının sıfır olup olmadığını anlama.

zaman2 programındaki MOV A,D ve ORA E satırlarını anlamaya çalışın. D'ye 0002H yükleyerek programı adım adım çalıştırın ve kaydedici içeriklerini gözleyin. Son aşamada Z bayrağının 1 olduğuna dikkat edin.

UYGULAMA-3: Iç içe döngü ile zaman gecikmesi

Aşağıdaki programı Sim8085 simülatöründe yazın. Dosyayı **zaman3** ismiyle kaydedin ve derleyin. Programı RUN komutu ile çalıştırıp toplam saat periyodu (clock cycle) değerini ekrandaki T-states adlı kısımdan okuyun. Bu sonucu kendi hesapladığınız periyot değeri ile kıyaslayın. (Not: Simülatörde süre hesabı için $T_{osc}=0.325\mu s$ olarak alınmalıdır.)

. . .

MVI B,0AH DONGU: MVI C,8FH TEKRAR: DCR C JNZ TEKRAR DCR B JNZ DONGU

Sonuç: 20204 periyot. Süre=6.566ms

UYGULAMA-4:	UYGULAMA-5:
Port 00'dan kare dalga oluşturma.	8 bitlik iki sayının çarpımı.
Aşağıdaki programı simülatörde yazın ve SLOW RUN ile çalıştırın. IO Port 00 adresindeki değerin belirli aralıklarla değiştiğini gözleyin.	Aşağıdaki programı simülatörde yazarak çalışmasını inceleyin. Program 2BH ile 17H sayısını çarpar ve sonucu 1000H ve 1001H adreslerine yükler.
TEKRAR:	MVI D,00H
MVI A,01H	MVI A,00H
OUT OOH	MVI C,2BH
CALL BEKLE	MVI B,17H
MVI A,00H	YY: ADD C
OUT 00H	JNC XX
CALL BEKLE	INR D
JMP TEKRAR	XX: DCR B
	JNZ YY
BEKLE:	STA 1001H
MVI C,10H	MOV A,D
XX: DCR C	STA 1000H
JNZ XX	HLT
RET	

UYGULAMA-6: Sağa kaydırma ve toplama yöntemiyle 8 bitlik çarpma.

Yazdığınız ödev programını simülatörde çalıştırıp sonucunu gözleyin.

UYGULAMA-7: Hafızadaki 20 sayıdan en büyüğünü ve en küçüğünü bulan program.

Yazdığınız ödev programını simülatörde çalıştırıp sonucunu gözleyin.

.

ADCONO Kaydedicisi (Adres Kontrol Kaydedicisi 0 - Adres 1Fh)

R/W	-0	R/W-0	R/W-0	R/W-0 R/W-0 R		R/W-0	U-0	R/W-0		
ADC	S1	ADCS0	CHS2	CHS1	CHS0	GO/DONE'		ADON		
Bit 7						1		Bit 0		
Bit 7 Bit 6	→	ADCS1 ADCS0	: A V fi k	/D çevirici içir erilecek değer rekans değeri aydedicisindel	n clock (saat) fr Ier ile A/D çevi bize sunulan d ki ADCS2 bit'i il	ekansı seçim bi irme işlemi esna eğerler içerisind e birlikte kullan	t'leridir. Bu Isında kullar Ien seçilir. (<i>i</i> Ilırlar.)	bit'lere IIIacak ADCON1		
				ADCON1 <adcs2></adcs2>	ADCON0) [S0>	Clock (saat)			
			1	0	00	Fosc/2	-			
				0	01	Fosc/8				
				0	10	Fosc/3	2			
				0	11	Frc (Da kaynač	ahili RC osila Iindan gelen	tör clock)		
•			t	1	00	Fosc/4				
				1	01	Fosc/1	6	the second second		
				1	10	Fosc/6	4			
				1	11	Frc (Da kaynač	ahili RC osila jindan geler	itör clock)		
Bit 5 Bit 4 Bit 3	→	CHS2 CHS1 CHS0	: /	VD çevirici için VD çevirme iç pelirtildiği gibi f 000 001 010 011 100 101 110 111	analog kanal se gin hangi kanal PIC 16F877A'da = kanal 0, (AN = kanal 1, (AN = kanal 2, (AN = kanal 3, (AN = kanal 4, (AN = kanal 6, (AN = kanal 7, (AN	eçim bit'leri. Bu t Iın seçileceği be A/D çevirici için i I0) I1) I2) I3) I3) I4) I5) I6) I7)	oit1ere verileo lirlenir. Daha 8 kanal mevo	:ek değerlerle, i önceden de :uttur.		
Bit 2	→	GO/DONI	E :	A/D dönüştürücü durum bit'i. <u>ADON = 1 iken:</u> 1 = A/D çevirici işlem yapıyor demektir. 0 = A/D cevirici herhangi bir işlem yapmıyordur.						
Bit 1	→	U	:	Kullanılmaz.						
Bit O	→	ADON	:	A/D çeviriciyi 1 = A/D çevir 0 = A/D çevir	yetkilendirme t ici açıktır ve işle ici kapalıdır.	bit'idir. em yapılabilir.				
R W U	: 1 : 1 : K	gili bit'in yalnı: gili bit'in yazıla sullanılmayan t	zca okur abilir bit bit'i göst	nabileceğini gö olduğunu gös erir.	österir. terir.					

Tablo 7.1 ADCONO özel amaçlı kaydedicisinin bit açıklamaları.

294 PIC Programlama Teknikleri ve PIC16F877A

		R/W-0	U-0)	U-0	R,	/W-0		R/	′W-0		R/V	V-0	R/W-0
ADFM		ADCS2			-	PC	CFG3		PC	FG	2	PCF	-G1	PCFG
Bit 7														Bit 0
Bit 7	→	ADFM	f	A/D çev yarayar bit'i 0 k bit'lik 1 kısmınd ADRESI	virme is bit'tir abul ec SB kıs laki 6 l L'nin so	şlemi e . ADFM dilir ve mına v pit'i 0 l n iki bit	snasında I=1 ise A/D çev ve ADRE kabul ec t'ine ve a	a ma AD IIIII IIIII ADR	eydar DRESH e soni ye ya ve A ESH'a	ha ge I kay ucun izilir. /D ç a yaz	elen ydedi da el ADf evirn ilir.	verinin cisinin de edil M=0 i ne sont	biçimi MSB I en ver ise AD ucu ele	ni belirlem kısmındaki i ADRESH'ı DRESL'nin I de edilen y
Bit 6	→	ADCS2	:	A/D çe kayded	virici iç licisind	in clocl eki AD(k (saat) CS1 ve .	fre ADC	kansı CSO b	seçi it'ler	im bi i ile l	ťidir. (pirlikte	ADCO kullar	NO nihr.)
				Ą	DCON1				505				Clock	
-30					0		0	0	302	1	Fosc/2	<u>- 2010 - 2011 -</u>	1 1 1 1 1 1 1 1	
					0		0	1			Fosc/8			
					0 .		1	.0		!	Fosc/3	2		
					0	1	1	1			rRC (L	anili RC	osilator	*
					1		0	0			Fosc/4	Indan ge	Jen cloc	.n/
					1		0	1		1	Fosc/1	6	<u>3</u> -	
x				-	1		1	0		H	Fosc/6	4		
				80 I	1		1	1			FRC (E	ahili RC	osilatör	· L)
											aynag	indan ge		<u>. N</u>
Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	→→	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind P(maz, " irici port e kullar CFG3:P	0" oku c'unun b iilacak p CFG0 E	nur. Diçimini d Din'lerin r Bit'lerini	lüzer nasıl n Al	nleme davra Idığı [yi sağ naca Değe	ğlayaı ğını b ere G	n bit'lere elirleme öre Ya	dir. Yar emize c ptığı İ	ni A/D çevir blanak sağlı şlemler:
新15 新14 新13 新12 新11 新10	→	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG3:	maz, " irici port e kullar CFG3:P	0″ okur c'unun b iilacak p CFG0 E v6 ANS	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI	lüzer nasıl n Al	nleme davra dığı [AN2	yi sağ naca Değe AN1	ğlayar ğını b ere G	n bit'lere elirleme öre Ya VREF+	dir. Yar emize c ptığı İ VREF-	ni A/D çevir blanak sağla şlemler: KANAL/RE
it 5 it 4 it 3 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000	maz, " irici port e kullar CFG3:P AN7 AI	0" okur c'unun b nilacak p CFG0 E N6 AN5	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI	lüzer nasıl n Al N3	nleme davra dığı [AN2 A	yi sağ naca Değe AN1 A	ğlayaı ğını b ere G ANO A	n bit'lere elirleme öre Ya VREF+	dir. Yar emize c ptığı İ VREF- VSS	ni A/D çevir olanak sağlı şlemler: KANAL/RE 8/0
it 5 it 4 it 3 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG3 0000 0001	maz, " irici port e kullar CFG3:P AN7 AI A	0" okur c'unun b iilacak p CFG0 E N6 AN5	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI A VRI	lüzer nasıl n Al N3 A EF+	nleme davra dığı [AN2 A A	yi sad naca Değe AN1 A	ğlayar ğını b ere G ANO A	n bit'lere elirleme öre Ya VREF+ VDD AN3	dir. Yar emize c ptığı İ VREF- VSS VSS	ni A/D çevi blanak sağl şlemler: KANAL/RE 8/0 7/1
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PC PCFG3: PCFG0 0000 0001 0010	maz, " irici port e kullari CFG3:P ANZ AI A A D U	0" okun c'unun b iilacak p CFG0 E N6 AN5 A A A A A A	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI A VRI A VRI	lüzer nasıl n Al N3 A EF+	nleme davra diği [AN2 A A A A	yi sağ naca Değe AN1 A A A	ğlayar ğını t ere G ANO A A	n bit'lere elirleme öre Ya VREF+ VDD AN3 VDD	dir. Yar emize c ptığı İ VREF- VSS VSS VSS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0
it 5 it 4 it 2 it 2 it 1 it 0	→	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0010	maz, " irici port e kullar CFG3:P ANZ AI A A D C D C	0" okun c'unun b iilacak p CFG0 E N6 AN5 A A A A A A D D D	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI A VRI A VRI	lüzer nasıl n Al N3 A EF+ A EF+	aleme davra diği [AN2 A A A A A	yi sağ naca Değe AN1 A A A	ğlayar ğını b ere G ANO A A A A	n bit'lere elirleme öre Ya VREF+ VDD AN3 VDD AN3	dir. Yar emize c ptığı İ VREF- VSS VSS VSS VSS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1
it 5 it 4 it 3 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101	maz, " irici port e kullar CFG3:P ANZ AI A Z D I D I D I	0" okun c'unun b iilacak p CFG0 E N6 AN5 A A A A A A D D D D	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI A VRI A VRI A VRI D VRI	lüzer nasıl n Al N3 A EF+ A EF+ A EF+	aleme davra dığı [AN2 A A A A A D	yi saq naca Değe AN1 A A A A	ğlayaı ğını t re G ANO A A A A	n bit'lere elirleme öre Ya VREF+ VDD AN3 VDD AN3 VDD AN3	dir. Yar emize c ptığı İ VREF- VSS VSS VSS VSS VSS	ni A/D çevil blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0
it 5 it 4 it 3 it 2 it 1 it 0	→	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x	maz, " irici port e kullar FG3:P	0" okur c'unun b iilacak p CFG0 E N6 AN5 A A A A D D D D D D D D D D	nur. Diçimini d Din'lerin r Bit'lerini AN4 AI A VRI A VRI A VRI D VRI D VRI	lüzer nasıl n Al A EF+ A EF+ A EF+ A EF+ D	AN2 A A A A A A A A A D D D	yi sad naca Değe AN1 A A A A A A A A D	ğlayaı ğını b re G A A A A A A A A A A A A A	n bit'lera elirlema öre Ya VREF+ VDD AN3 VDD AN3	dir. Yar emize c ptığı İ VREF- VSS VSS VSS VSS VSS	ni A/D çevi olanak sağı şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0
it 5 it 4 it 2 it 1 it 0	→	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x 1000	maz, " irici port e kullar FG3:P AN7 AI A 7 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1	0" okur c'unun b iilacak p CFG0 E NG ANS A A A A D D D D D D D D D D D D D D	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI A VRI D VRI D VRI D VRI A VRI	lüzer nasıl n Al N3 A EF+ A EF+ A EF+ D EF+	AN2 A A A A A A A D D D VREF-	yi sad naca Değe AN1 A A A A A A A A A A A A A A A A	jlayai ğini b re G A A A A A A A A A A A A A A A A A A	n bit'lera elirlema öre Ya VREF+ VDD AN3 VDD AN3 - AN3	dir. Yar emize c ptığı İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2
it 5 it 4 it 2 it 2 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x 1000 1001	maz, " irici port e kullar CFG3:P AN7 AI A 7 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1 D 1	0" okur c'unun b iilacak p CFG0 E N6 AN5 A A A A A A D D D D D D D D D D D D A A A A	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI D VRI D VRI D VRI A VRI A VRI	lüzer nasıl n Al EF+ A EF+ A EF+ D EF+ A	AN2 A A A A A A A D D VREF- A	yi sad naca Değe AN1 A A A A A A A A A A A A	jlayai ğini b ere G A A A A A A A A A A A A A A A A A A	n bit'lera elirlema öre Ya VREF+ VDD AN3 VDD AN3 - AN3 VDD	dir. Yar emize c ptīği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x 1000 1001 1010	maz, " irici port e kullar CFG3:P AN7 AI A 7 A 7 D 10 D 10 D 10 D 10 D 10 D 10 D 10 D 10	0" okur c'unun b iilacak p CFG0 E N6 AN5 A A A A A A A A A A A A A A A A A A A	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI D VRI D VRI A VRI A VRI A VRI	lüzer nasıl n Al EF+ A EF+ A EF+ D EF+ A EF+ A EF+	AN2 A A A A A A A D D VREF- A A	yi sad naca Değe AN1 A A A A A A A A A A A A A A	ğlayaı ğını b ere G ANO A A A A A A A A A A A A A A A A A A	n bit'lere elirleme öre Ya VDD AN3 VDD AN3 VDD AN3 VDD AN3 VDD AN3	dir. Yar emize c ptiği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi olanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0 5/1
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x 1000 1001 1010 1011	maz, " irici port e kullar CFG3:P AN7 AI A 7 A 7 D 10 D 10 D 10 D 10 D 10 D 10 D 10 D 10	0" okun c'unun b iilacak p CFG0 E N6 AN5 A A A A A A A A A A A A A A A A A A A	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI A VRI A VRI A VRI A VRI A VRI A VRI	lüzer nasıl n Al EF+ A EF+ A EF+ D EF+ A EF+ EF+	AN2 A A A A A A A D D VREF- A A VREF- A X	yi sad naca Değe AN1 A A A A A A A A A A A A	ğlayaı ğını b ere G ANO A A A A A A A A A A A A A A A A A A	n bit'lere elirleme öre Ya VREF+ VDD AN3 VDD AN3 VDD AN3 VDD AN3 AN3 AN3	dir. Yar emize c ptiği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0 5/1 4/2
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PCFG3: PCFG0 0000 0001 0010 0011 0100 0101 011x 1000 1001 1011 1010	maz, " irici port e kullar CFG3:P AN7 AI A 7 A 7 D 10 D 10 D 10 D 10 D 10 D 10 D 10 D 10	CFGO E	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI D VRI D VRI D VRI A VRI	lüzer nasıl n Al EF+ A EF+ A EF+ D EF+ A EF+ EF+ EF+ EF+	AN2 A A A A A A A A D D VREF- A A VREF- VREF- VREF- VREF-	yi sad naca Değe AN1 A A A A A A A A A A A A A A A A A A	ğlayaı ğını b ere G ANO A A A A A A A A A A A A A A A A A A	n bit'lere elirleme öre Ya VDD AN3 VDD AN3 VDD AN3 VDD AN3 AN3 AN3 AN3 AN3 AN3	dir. Yar emize c ptiği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0 5/1 4/2 3/2
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PC PCFG3: PCFG0 0000 0001 0010 0011 0101 011x 1000 1001 1011 1010 1011 1100	maz, " irici port e kullar CFG3:P AN7 AI A 7 A 7 D 10 D 10 D 10 D 10 D 10 D 10 D 10 D 10	o" okur c'unun b iilacak p CFGO E N6 AN5 A A A A A A A A D D D D D D D D A A A A	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI D VRI D VRI D VRI A VRI	lüzer nasıl n Al EF+ A EF+ A EF+ A EF+ EF+ EF+ EF+ EF+	AN2 A A A A A A A D D VREF- A A VREF- VREF- VREF- VREF- VREF- D	yi sað naca Değe AN1 A A A A A A A A A A A A A A A A A A	ğlayal ğini b ere G A A A A A A A A A A A A A A A A A A A	n bit'lere elirleme öre Ya VDD AN3 VDD AN3 VDD AN3 - AN3 AN3 AN3 AN3 AN3 AN3	dir. Yar emize c ptiği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi blanak sağl şlemler: KANAL/RI 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0 5/1 4/2 3/2 2/2
it 5 it 4 it 2 it 1 it 0	 → → 	U PCFG3 PCFG2 PCFG1 PCFG0	:	Kullanıl A/D çev işlemind PC PCFG3: PCFG0 0000 0001 0010 0011 0101 1000 1001 1001 1010 1001 1010 1001 1010 1101 1110	maz, " irici port e kullan CFG3:P ANZ AI A Z D C D C D C D C D C D C D C D C D C D C	CFG0 E	nur. biçimini d bin'lerin r Bit'lerini AN4 AI A VRI A VRI D VRI D VRI A VRI A VRI A VRI A VRI D VRI	lüzer nasıl n Al EF+ A EF+ A EF+ EF+ EF+ EF+ EF+ EF+ D EF+ D EF+	AN2 A A A A A A A D D D VREF- A A VREF- VREF- D VREF- D VREF-	yi saq naca Değe AN1 A A A A A A A A A A A A A D D D D D	ğlayal ğını b ere G A A A A A A A A A A A A A A A A A A A	n bit'lero elirleme öre Ya VDD AN3 VDD AN3 VDD AN3 - AN3 AN3 AN3 AN3 AN3 AN3 AN3 AN3 AN3 AN3	dir. Yar emize o ptiği İ VREF- VSS VSS VSS VSS VSS VSS VSS VSS VSS VS	ni A/D çevi blanak sağı şlemler: KANAL/R 8/0 7/1 5/0 4/1 3/0 2/1 0/0 6/2 6/0 5/1 4/2 3/2 2/2 1/0 1/2

Tablo 7.2 ADCON1 özel amaçlı kaydedicisinin bit açıklamaları.

INTCON Kaydedicisi (Kesme Kontrol Kaydedicisi – Adres OBh, 8Bh, 10Bh, 18Bh)

R/W-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE		PEIE	TMROIE	INTE	RBIE	TMROIF	INTF	RBIF
Bit 7								Bit O
Bit 7	\rightarrow	GIE	- 1	ům kesme işi 0 = Tüm	emierine izin kesmelere iz	in verilmez.		
				1 = Aktif	yapılmış olar	n tüm kesmele	ere izin verilir	-
Bit 6	\rightarrow	PETE	: (Cevre ara birin	nlerden geler	n kesmelere iz	in verme biť	i.
DICO				0 = Tüm	çevresel kes	melere izin ve	rilmez.	ain vorilir
				1 = Aktit	yapılmış olar	n tum çevrese	i kesmelere i	zm venin.
Bit 5	\rightarrow	TMROI	E :7	MR0 sayici ta	aşma kesmes	sine izin verm	e bit'i.	
				0 = TMF 1 = TMF	lo kesmesine 10 kesmesine	izin verilir.		
					dan gelen ke	emeve izin ve	rme bit'i	
Bit 4	->	INTE		0 = RB0	/INT kesmesi	ine izin verilm	ez.	
				1 = RB0	/INT kesmesi	ine izin verilir.		
Bit 3	• ->	RBIE		PORTB (4, 5,	6, 7. bit'leri) (değişiklik kesn	nesine izin v	erme bit'i.
				0 = POF	RTB'deki deği	şiklik kesmesi	ne izin veriln	nez.
				1 = POF	T B deki degi	şıkık kesinesi		
Bit 2	\rightarrow	TMROI	F:	TMR0 sayicis	i taşma kesm	iesî bayrağı bi	ťi.	
	1		-	0 = Zan 1 = Zan	an aşımı var	(h'FF' den h'	00' a geçiş.)	
Bit 1	\rightarrow	INTF	:	RB0/INT haric	i kesme bayr	ağı bit'i.		
				0 = HBC 1 = BBC	VINT harici ke	esme oluşmac esme oluştuğu	inginoa. Inda.	
Bit O	\rightarrow	RBIF	:	PortB değişik	ik kesmesi ba	ayrağı bit'i.		
				0 = RB4	-RB7 uçlarır	nda değişiklik	yok. an birisinda	doğisiklik var
D	• İlail	i bit'in valn	zca okuna	i = RB2 bileceăini aöst	erir.	iuan en azinu	anomsnide	acgiginin var.
Ŵ	: İlail	i bit'in yazıl	abilir bit ol	duğunu göste	rir.			
υ	: Kull	anılmayan	bit'i göster	ir.				

OPTION_REG Kaydedicisi (Seçme Kaydedicisi – Adres 81h, 181h)

R/W-1	F	R/W-1	R/W-1	R,	/W-1	R/	W-1 R/	W-1 R/V	W-1	R/W-1	
RBPU'	IN	TEDG	TOCS	Т	OSE	P	SA P	S2 P	S1	PS0	
Bit 7										Bit 0	
Bit 7	→	RBPU'	:	PortB'nii 1 = Pull 0 = Pull	n pull-u -up din -up din	ıp direr ençleri ençleri	nçlerini aktif e aktif. pasif.	etme bit'i.			
Bit 6	→	INTEDG	:	 Harici kesme (RB0/INT) sinyali seçme bit'i. 1 = RB0/INT pininden gelen sinyalin yükselen kenarında kesme. 0 = RB0/INT pininden gelen sinyalin düşen kenarında kesme. 							
Bit 5	Bit 5 → TOCS : Timer0 sinyal kaynağı seçme bit'î. 1 = Harici sinyal kaynağı (RA4/T0CKI). 0 = Dahili komut çevrimi seçilir.										
Bit 4	→	TOSE	:	Timer0 sinyal kaynağı kenar seçme bit'i. 1 = RA4/T0CKI pininden yükselen kenar tetiklemesi. 0 = RA4/T0CKI pininden düşen kenar tetiklemesi.							
Bit 3	→	PSA		Frekans 1 = Fre 0 = Fre	s bölüc kans b kans b	ü seçm ölme s ölme s	e bit'i. ayısı WDT içir ayısı Timer0 i	n geçerli. çin geçerli.			
Bit 2-0	\rightarrow	PS2:PS	o :	Prescal	er değ	eri seçr	ne bit'leri.				
				B	it Değe	ri	TMR0 Oran	WDT Oran			
				PSZ	PSI	PSO	1.2	1:1			
				U O	0	1	1:4	1:2			
				0	1	ō	1:8	1:4			
				Ŏ	1	1	1:16	1:8			
				1	Ō	Ō	1:32	1:16			
				1	0	1	1:64	1:32			
				1	1	0	1:128	1:64			
				1	1	1	1:256	1:128	l		

: İlgili bit'in yalnızca okunabileceğini gösterir. : İlgili bit'in yazılabilir bit olduğunu gösterir. : Kullanılmayan bit'i gösterir. R W U