

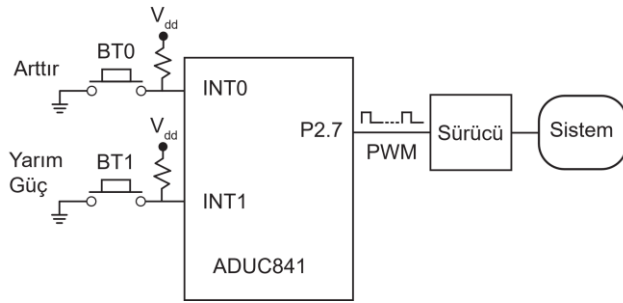
Sıra No: (Listeden bakıp yazınız)	EEM 304 MİKROİŞLEMCİLER 2018 – 2019 Bahar Yarıyılı	
Ad:		
Soyad:	FİNAL	
Öğrenci No:	24.05.2018	
İmza:	Süre: 95dk	

1) Şekilde verilen **10.24 MHz osilatör frekansı** ile çalışan ADUC841 temelli sistemde mikrokontrolöre bağlı bulunan sistemin sürücü aracıyla ayarı ve kontrolü için PWM çıkışı kullanılacaktır. PWM duty cycle değeri (görev süresi) harici kesme 0'a bağlı girişteki BT0 arttırma yönünde ayarlanacaktır. Duty cycle ayarı 1/50000'lik kademelerle olacaktır. Her butona basılıştta 50000'de 1 oranında artım olacaktır. (Artımda saklayıcı olarak DPTR'yi kullanınız. DPTR ile inc komutu çalışmaktadır).

BT1'e basıldığında ise PWM direkt %50 duty cycle ayarına getirilecektir. Butonlar için kesme kullanımı zorunludur. Butonların işlevleri basıp-çekilme işlemi ile aktif olacaktır. Başlangıç işlemlerinden sonra program sonsuz döngüde beklemelidir.

1/50000'lik duty cycle 12ms değerine karşılık gelecektir. (İşaretin periyodu bunun 50000 katıdır yani 600sn'dir).

Aşağıda istenenleri cevaplayınız ve gerekli assembly kodunu verilen kriterlere eksiksiz uyararak oluşturunuz.



Verilen PWM ayarları:

Çıkış P2.7,

PWM giriş saati, CLKPWM=fosc/(15*Bölme faktörü)

Bölücü katsayısı: N=16,

Kullanılacak PWM modu: Mod1

Başka bir modda kullanmak mümkün mü

(Evet/Hayır): Hayır

Neden: Hem 1/50000 ayarı hem de 600s'lik periyot değerinin elde edilmesi aynı anda bir başka modla sağlanamaz.

Belirlenen PWM ayarları:

Bölme Faktörü: **512**

PWM frekansı: $1/600 = 1.6 \times 10^{-3}$ Hz

PWM periyodu: **600s**

Aşağıdaki istenen değerleri belirleyiniz. Doldurulması zorunludur.

Not: Bilgilerin burada istenmesi doğru işlem adımlarına yönlendirme amaçlıdır. Kod içinde kaydedicilere gerektiği gibi değer ataması eksiksiz şekilde olmalıdır.

								Yapılan ayarlar
CFG841								Çıkış P2.7
0	1	0	0					Bölme faktörü: 512

								Yapılan ayarlar
PWMCON								Tek Çıkış PWM, Mod 1
1	0	0	1	1	0	0	0	Bölücü katsayısı: N=16 CLKPWM=fosc/(15*Bölme faktörü)

PWM'in açılımı nedir? DC işaret ile ilişkisini bağıntı ile veriniz. DAC çıkışındaki işaret ile karşılaştırınız. Sistemi sürmeyle ilgili kullanıma mantığını açıklayınız. (Yalnızca sorulara ve birer cümle ile yanıt veriniz)

***PWM (Pulse Width Modulation – Darbe Genlik Modülasyonu**

***0-V_{LH} arasında değişen bir PWM işareti için DC eşdeğeri anlamında ortalaması: $V_{AVG} = (T_{ON}/T_{PWM}) \times V_{LH} = DutyCycle \times V_{LH}$**

***DAC çıkışında sabit DC işaret elde edilirken PWM ile bunun eşdeğeri min ve maks noktaları sabit, doluluk oranı değiştirilen bir kare dalga ile elde edilir.**

***PWM çok temel bir kontrol işareti olup, mikrodenetleyici tarafından çok farklı tür ve seviyedeki sistem için kontrolü için oluşturulur. Sürücü devre bu PWM işaretini kontrol edilecek sistem için uygun seviyedeki elektriksel büyüklüklere dönüştürür.**

Kodun tamamını arka sayfaya yazınız.

Başarılar dileriz...

Dr. Öğr. Üyesi Burhan BARAKLI

Doç. Dr. Şuayb Çağrı YENER

```

#include<ADUC841.H>
    ORG      0000h
    SJMP    BASLA
    ORG      0003h
    SJMP    ARTTIR
    ORG      0013h
    SJMP    YARIM
BASLA:
    mov     acc, cfg841
    clr     acc.6                ;pwm çıkisi p2.7 ayari
    setb   acc.4
    clr     acc.3
    clr     acc.2                ; bolucu 512 oldu
    mov     cfg841, acc
    mov     acc, pwmcon
    setb   acc.7                ;tek cikis.
    clr     acc.6                ;MD2
    clr     acc.5                ;MD1
    setb   acc.4                ;MD0 ( PWM mode-1 seçildi)
    setb   acc.3                ;CDIV1
    clr     acc.2                ;CDIV0 ( Çarpan N=16 seçildi )
    clr     acc.1                ;CSEL1
    clr     acc.0                ;CSEL0 ( CLKpwm=fosc/(15*bölme faktörü) seçildi )
    mov     pwmcon, acc

    mov     dptr, #50000d
    mov     pwm1H, dph           ;z pwm frekans ayari
    mov     pwm1L, dpl           ;Ton süre ayari, baslangicta 0

    mov     DPTR, #00h           ;Sayac
    mov     pwm0H, dph
    mov     pwm0L, dpl

    SETB   EA
    SETB   EX0
    SETB   EX1
DNG:
    SJMP   DNG
ARTTIR:
    JNB    P3.2, ARTTIR         ;Butondan cekme kontrolu
    inc    DPTR                 ;Her basimde 1 arttir
    mov     pwm0H, dph
    mov     pwm0L, dpl
    RETI
YARIM:
    JNB    P3.3, YARIM         ;Butondan cekme kontrolu
    mov     DPTR, #25000d       ;%50 duty
    mov     pwm0H, dph
    mov     pwm0L, dpl
    RETI
END

```

Sıra No: (Listeden bakıp yazınız)	EEM 304 MİKROİŞLEMCİLER 2018 – 2019 Bahar Yarıyılı	
Ad:		
Soyad:	FİNAL	
Öğrenci No:	24.05.2018	
İmza:	Süre: 95dk	

2. 4msn aralıklarla ADC'den 10 değer alınacak ve alınan her değer ADCDATAL kısmı sırasıyla RAM'in [0Ah,09H,....,02H,01H] adreslerine kaydedilecektir. Değerlerin RAM'e kaydedilmesinin ardından, değerlerin toplamı 255'ten büyük ise timer'ı durdurunuz, aksi halde işlemleri en baştan başlatabilecek asm kodunu yazınız. (Timer 0 kesmesi ve ADC kesmesi kullanınız. RAM'e kaydetme işlemleri döngüyle yapılmalıdır)

ADC ayar: ADC çevrim başlangıcı tek çevrim, (T/H)=1, Fadc=Fosc/32, Vref=harici, ADC Kanalı ADC0

```

#include "aduc841.h"
org          0000
sjmp        ayar
org          000bh
sjmp        t0kesmesi
org          0033h
sjmp        adckesmesi

ayar:
mov         tmod,#00000001b
mov         dptr,#21299d
mov         t1,dpl
mov         th1,dph

mov         adcon1,#11001100b
mov         adcon2,#00000000b

setb        ea
setb        et0
setb        eadc

mov         r0,#0ah
setb        tr0
basla:      sjmp        basla
t0kesmesi:
mov         t1,dpl
mov         th1,dph
setb        sconv ; tek çevrim basladi
reti

adckesmesi:
mov         @r0,adcdatal
djnz        r0,son

mov         a,#00h ; toplam degiskeni
mov         r0,#0ah
clr         c

acall       toplam

son:        reti

toplam:
add         a,@r0
djnz        r0,toplam

jc          timerdur
mov         r0,#0ah
sjmp        son1

timerdur:
clr         tr0

son1:      ret
end

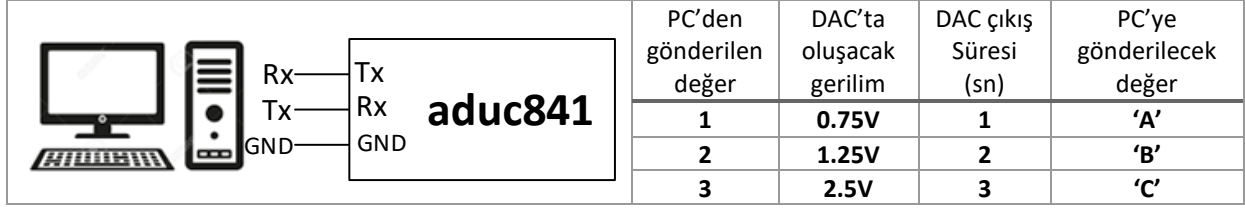
```

Başarılar dileriz...

3. Bir bilgisayar(PC) ile ADUC841 arasında seriport haberleşmesi yapılacaktır. Bilgisayardan gönderilen 3 değişik karaktere karşılık, değişik değerlerde DAC0 çıkışları ve Timer süreleri ayarlanıp bilgisayara bilgi verilecektir.

Örneğin bilgisayardan ADUC841'e 1 değeri gelmesi durumunda DAC0 çıkışında 1sn süre ile 0.75V oluşturunuz ve PC'ye 'A' karakterini gönderiniz ve işlemlerin tekrar baştan devam edebileceği asm kodunu yazınız. Diğer değerler aşağıda tabloda verilmiştir.

DAC0 (12bit) ve dahili ref kullanılacaktır. Seriport 9600 baudrate ayarında (0DCh), timer 0 kullanılacaktır. Kesme kullanımı ve timer modları isteğinize bağlı olarak kullanabilirsiniz!



```

#include "aduc841.h"
org      0000
sjmp    ayar

ayar:
    mov     scon,#01010000b
    mov     tmod,#00100001b
    mov     tl1,#0dch
    mov     th1,#0dch
    setb    tr1
    mov     adcon1,#10000000b
    mov     daccon,#00001101b

basla:
    jnb     ri,$
    clr     ri
    mov     a,sbuf

birmi:  cjne  a,1d,ikimi
    mov     daccon,#00001101b
    mov     dptr,#1228d
    mov     dac0h,dph
    mov     dac0l,dpl
    acall   birsaniye
    mov     daccon,#00000101
    mov     sbuf,'B'
    acall   karaktergonder
    sjmp    x

ikimi:  cjne  a,2d,ucmu
    mov     daccon,#00001101b
    mov     dptr,#2047d
    mov     dac0h,dph
    mov     dac0l,dpl
    acall   birsaniye
    acall   birsaniye
    mov     daccon,#00000101b
    mov     sbuf,'B'
    acall   karaktergonder
    sjmp    x

ucmu:   cjne  a,3d,birmi
    mov     daccon,#00001101b
    mov     dptr,#4095d
    mov     dac0h,dph
    mov     dac0l,dpl
    acall   birsaniye
    acall   birsaniye
    acall   birsaniye
    mov     daccon,#00000101
    mov     sbuf,'C'
    acall   karaktergonder
x:      sjmp  basla

karaktergonder:
    jnb     ti,$
    clr     ti
    ret

birsaniye:
    setb    tr0
    mov     r0,#169d
y:      jnb     tf0,$
    clr     tf0
    mov     tl0,#00h
    mov     th0,#00h
    djnz    r0,y
    ret
end

```