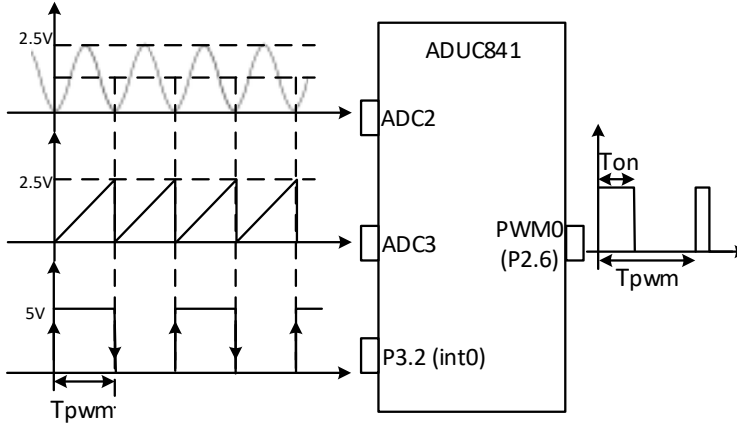


SORU 1



Yukarıda görüldüğü gibi ADUC841'in ADC2 kanalına sinüs işareti ve ADC3 kanalına testere dişi sinyal gelmektedir. Bu iki analog sinyal $T_s=1 \text{ msn}$ ile örneklenerek RAM'in sırasıyla 40h, 41h, 42h ve 43h adreslerine kaydedilecektir (ADC2H-L $\rightarrow$ 40h,41h ADC3H-L $\rightarrow$ 42h,43h). ADC2 ve ADC3 kanalından okunan değerler aşağıdaki gibi karşılaştırılarak PWM işaretinin **Ton** süresi belirlenecektir. **Tpwm** süresi ise P3.2 kanalından gelen işaret ile belirlenecektir.

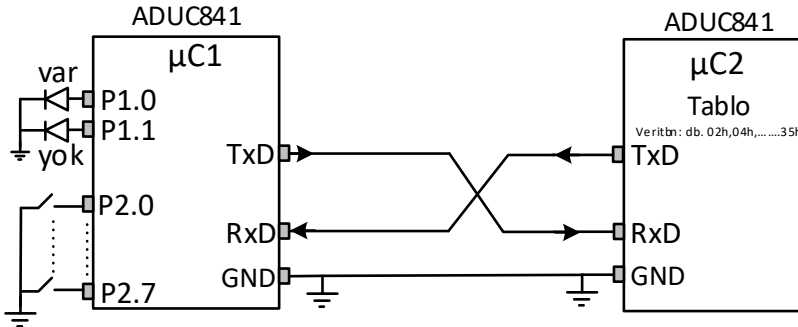
- $ADC3 > ADC2$ ise $Ton = ADC0 - ADC1$ (12 bit)

- $ADC3 < ADC2$ ise $Ton = ADC1 - ADC0$ (12 bit)

- $ADC3 = ADC2$ ise $Ton = \%10$ doluluk oranında olacaktır. (yüzde doluluk oranı $\rightarrow \%d = (Ton/Tpwm) * 100$)

(**Tpwm** süresi için P3.2'den gelen darbelerin "Lojik 1" olarak kaldığı süre; Dış kesme 0 ve Timer 0 kesmesi kullanılarak algılanacaktır, PWM0 (P2.6) Mod1'de kullanılacaktır, Bölücü katsayısı $N=1$, PWM giriş saati f_{osc} , SGB=1, ADC için çalışma ayarları; Dahili referans, ADC kesmesi, T/H için 4 ADC saat darbesi, $f_{ADC} = f_{OSC}/2$, ADC'den örneklenen işaretin örnekleme zamanı için Timer2 ile ayarlanacaktır, $f_{osc} = 11.0592 \text{ MHz}$)

SORU 2



Yukarıda görüldüğü gibi iki mikro kontrolör seri haberleşme kullanılarak haberleştirilmek istenmektedir. uC1'de P2'den okunan değer seri kanaldan uC2'ye gönderilecektir. uC2 ise uC1'den seri kanaldan gelen veriyi kendi veri tabanındaki (aşağıda verilen) tablo ile karşılaştırarak eğer değerler eşit ise seri kanaldan uC1'e 'A' karakteri gönderilecek eğer eşit değil ise 'B' karakteri gönderilecektir. Tablo'da toplam 20 adet veri vardır. uC1 eğer seri kanaldan 'A' karakteri alırsa P1.0'a bağlı 'var' ledini yakacaktır. Eğer 'B' karakterini alırsa 'var' ledini söndürüp P1.1'e bağlı 'yok' ledini yakacaktır. Program sürekli P2'yi okuyacak şekilde bir döngüde olmalıdır. (Seri kanal ayarları: Mod 1, Boundrate: 9600bps, $f_{osc} = 11.0592 \text{ MHz}$, SMOD=0, seri kanaldan veri alışverişi yoklamalı olarak kontrol edilecektir.)

Veritbn: db. 00h, 02h, 07h, 0Ah, ..., 35h
(20 adet) 0 19

Başarılar.. Süre:100dk

Yrd. Doc. Dr. Zekive ERDEM

CEVAPLAR

SORU 1:

```
#include<Aduc841.h>
```

```
ADC_KANAL2 EQU 02H
```

```
ADC_KANAL3 EQU 03H
```

```
ADC2H EQU 40H
```

```
ADC2L EQU 41H
```

```
ADC3H EQU 42H
```

```
ADC3L EQU 43H
```

```
FLAG EQU 00H
```

```
FLAG2 EQU 01H
```

```
TONH EQU 50H
```

```
TONL EQU 51H
```

```
ORG 00H
```

```
SJMP BASLA
```

```
ORG 03H
```

```
SJMP DIS_KESME
```

```
ORG 0BH
```

```
SJMP TIMER_KESME
```

```
ORG 33H
```

```
SJMP ADC_KESME
```

```
BASLA:
```

```
MOV ADCCON1,#10111110B
```

```
MOV ADCCON2,#ADC_KANAL2
```

```
MOV T2CON,#00H
```

```
MOV PWMCON,#10010011B
```

```
MOV T2H,#D4H
MOV T2L,#98H; A3 'DE OLABİLİR
MOV RCAP2L,#D4H
MOV RCAP2H,#98H    ; BASLANGIC AYARLARI YAPILDI
MOV TL0,#00H
MOV TH0,#00H
MOV R0,#00H
SETB EA
SETB EADC
SETB EX0
SETB IT0            ; DUSEN KENAR KESMESI
SETB ET0            ; KESMELER AKTIF
SETB PT0            ;Z/S 0 KESMESI ONCELIKLI

SETB TR0            ; Z/S 0 START
SETB TR2            ;Z/S 2 START
X: SJMP X
```

```
ADC_KESME:
INC R0
CJNE R0,#02H,KANAL_DEGISTIR
MOV R0,#00H
MOV ADCCON2,#ADC_KANAL2
MOV ADC3H,ADCDATAH
MOV ADC3L,ADCDATAL
ACALL KARSILASTIR
RETI
```

```
KANAL_DEGISTIR:
MOV ADCCON2,#ADC_KANAL3
MOV ADC2H,ADCDATAH
MOV ADC2L,ADCDATAL
RETI
```

TIMER_KESME:

SETB FLAG

RETI

DIS_KESME:

JB FLAG, TASMA

MOV A,TH0

MOV PWM1H,A

MOV A,TLO

PWM1L,A

RETI

TASMA:

CLR FLAG

PWM1H,#0FFh

PWM1L,#0FFh

RETI

KARSILASTIR:

CJNE ADC2H,ADC3H,K1

CJNE ADC2L,ADC3L,K3

ACALL YUZDE_ON

RET

K1: JB C,KUCUK

SJMP BUYUK

K3: JB C,KUCUK

SJMP BUYUK

KUCUK:

MOV A,ADC3H

SUBB A,ADC2H

MOV TONH, A

MOV A,ADC3L

```
SUBB A,ADC2L
MOV TONL, A
MOV PWM0L, TONL
MOV PWM0H,TONH
SJMP SON

BUYUK:
MOV A,ADC2H
SUBB A,ADC3H
MOV TONH, A
MOV A,ADC2L
SUBB A,ADC3L
MOV TONL, A
MOV PWM0L, TONL
MOV PWM0H,TONH
SJMP SON
```

```
SON: RET
```

```
YUZDE_ON:
MOV PWM0L, TONL
MOV PWM0H,TONH
RET
END
```

SORU 2:

:UC1 ICIN PROGRAM KODLARI ASAGIDADIR

```
#include<Aduc841.h>
```

```
DEGER EQU 30H
VAR EQU P1.0
YOK EQU P1.1
```

```
ORG 00H
SJMP BASLA
```

BASLA:

```
MOV SCON,#01010000B    ;mod 1, seri kanaldan veri alma yetkilendirildi
MOV TMOD,#00100000B    ; timer1 8 bit yeniden yüklemeli mod secildi
MOV TH1,#0DCH           ;9600 baudrate için deęer yuklendi
MOV TL1,#0DCH
CLR VAR                 ; var ledi sonuk
CLR YOK                 ; yok ledi sonuk
```

```
SETB TR1
```

MAIN:

```
MOV A,P2
ACALL GONDER
ACALL AL
ACALL KARSILASTIR
SJMP MAIN
```

GONDER:

```
MOV SBUF, A
JNB TI, $
CLR TI
```

RET

AL:

```
JNB RI,$
MOV A,SBUF
MOV DEGER, A
CLR RI
```

RET

KARSILASTIR:

```
MOV A,DEGER
MOV 20H,#'A'
```

```
        CJNE A,20H , YOK_YAK
        SETB VAR
        CLR YOK
RET
```

```
YOK_YAK:
        MOV 21H,#'B'
        CJNE A,21H,SON
        SETB YOK
        CLR VAR
SON:    RET
END
```

;UC2 ICIN PROGRAM ASAGIDADIR

#include<Aduc841.h>

```
DEGER EQU 30H
ORG 00H
SJMP BASLA
```

```
BASLA:
        MOV SCON,#01010000B
        MOV DPTR,#VERITBN
        MOV TMOD,#00100000B
        MOV TH1,#0DCH
        MOV TL1,#0DCH
        MOV R0,#13H
        MOV R1,#00H
        SETB TR1
```

```
MAIN:
        ACALL AL
        ACALL KARSILASTIR
        SJMP MAIN
```

AL:

```
JNB RI,$  
MOV A,SBUF  
MOV DEGER, A  
MOV A,#00H  
CLR RI
```

RET

KARSILASTIR:

```
MOVC A,@A+DPTR  
CJNE A,DEGER, DVM  
ACALL A_GONDER
```

RET

DVM:

```
INC R1  
MOV A,R1  
DJNZ R0,KARSILASTIR  
ACALL B_GONDER  
MOV R0,#13H  
MOV R1,#00H
```

RET

A_GONDER:

```
MOV A,#'A'  
MOV SBUF, A  
JNB TI, $  
CLR TI
```

RET

B_GONDER:

```
MOV A,#'B'  
MOV SBUF, A
```


JNB TI, \$

CLR TI

RET

VERITBN:

DB 00H,02H,05H,06H,0AH,0CH,0CDH,00H,02H,05H,06H,0AH,0CH,0CDH, 00H,02H,05H,06H,0AH,0CH

END