

## Introducción a los Microcontroladores

### 2º Departamental

1.- Con el sobreflujo del TIMERO aproximadamente cada 1 mseg. haga un toggle en la señal de salida por el pin PB0. (2.5 pts)

$$1\mu s = (4\mu s)N$$

Despejando

$$N = 1\mu s / 4\mu s = 4$$

```
.def temp=r16
.cseg
.org 0
rjmp reset
.org $020
rjmp timer_0
reset:ldi temp,$03
out tccr0b,temp
ldi temp,$01
out ddrb,temp
sts timsk0,temp
out portb,temp
sei
main:nop
rjmp main
;;*interrupciones
timer_0:in temp,pinb
com temp
andi temp,$01
out portb,temp
reti
```

2.- Se tiene conectado un push-boton con resistencia de pull-up interna en el pin PB1 de manera que cada que se presiona el botón se toma una muestra de una señal analógica en ADC0, la resolución del ADC es de 8 bits y el resultado de la conversión se muestra en 8 leds conectados en el puerto D. (2.5 pts)

```
.def temp=r16
.def cont1=r17
.def cont2=r18
.cseg
```

**.org 0**

**rjmp reset**

**.org \$015**

**rjmp fin\_conv**

**reset:ldi temp, \$03**

**out portb, temp**

**ldi temp,\$ff**

**out ddrd,temp**

**ldi temp,\$60**

**sts admux,temp**

**ldi temp,\$01**

**sts didr0,temp**

**ldi temp,\$cb**

**sts adcsra,temp**

**sei**

**main:sbic pinb,1**

**rjmp main**

**rcall delay\_15m**

**ldi temp,\$cb**

**sts adcsra,temp**

**rjmp main**

**fin\_conv:lds temp,adch**

**out portd,temp**

**reti**

**delay\_15m: ldi cont1,6**

**lazo2: ldi cont2,250**

**lazo1: nop**

**nop**

**nop**

**nop**

**nop**

**nop**

**nop**

```

dec cont2
brne lazo1
dec cont1
brne lazo2
ret

```

3.- Utilizando el TIMER 0 generar una señal PWM con una frecuencia de 244 Hz aproximadamente y un ciclo de trabajo del 25% aprox. (2.5 pts)

```

.def temp = r16
.def cte = r17
.def cont1 = r18
.def cont2 = r19
.def cont3 = r20
.cseg
.org 0

ldi temp, $40 ; pind6 como salida
out ddrd, temp

; trabajamos a 16M =>  $16 \times 10^6 / (256 * 256)$  ponemos el preescalador a 64

; usamos fast pwm
ldi temp, $83
out tccr0a, temp
; preescalador
ldi temp, $04
out tccr0b, temp
; registro de comparacion
ldi temp, 63 ; funcionando a 1/4 aprox 25%
out ocr0a, temp

ldi cte, $05

main: sbic pind, 5
      jmp main

; incrementamos de 5 en 5
call delay_100m
in temp, ocr0a
add temp, cte
out ocr0a, temp
jmp main

delay_100m: ldi cont1, 4
lazo3: ldi cont2, 200
lazo2: ldi cont3, 200

```

```

lazo1:  nop
        nop
        nop
        nop
        nop
        nop
        nop
        dec cont3
        brne lazo1
        dec cont2
        brne lazo2
        dec cont1
        brne lazo3
        ret

```

4.- Modifique el código del problema anterior de manera que por medio de la USART se pueda modificar el ciclo de trabajo de la señal PWM, es decir los datos que se reciben en la USART modifican el ciclo de trabajo de la señal PWM. (2.5 pts)

5.- Obtenga una señal cuadrada en el pin PB0 de un periodo aproximado de 32 mseg, utilice un delay. (2.5 pts)

```

.def temp=r16
.def cont1=r17
.def cont2=r18

.cseg
.org 0

ldi temp,$01
out portb,temp

main:ldi temp,$00
out portb,temp
sbi portb,0
rcall

delay_16m
rjmp main

;;*subrutinas
delay_16m: ldi cont1,6
lazo2:ldi cont2,250
lazo1:nop
        nop
        nop

```

```
nop  
nop  
nop  
nop  
dec cont2  
brne lazo1  
dec cont1  
brne lazo2  
ret
```

**RESOLVER ÚNICAMENTE 4 PROBLEMAS, CONSIDERAR UNA FRECUENCIA DE TRABAJO DE 16 MHz.**

**otros**