INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Introducción a los Microcontroladores

Reportes de Prácticas | 5 – 9

Chávez Chávez Ángel Alexis

Hernández López Pamela

Flores Alvarado Diego Armando

Profesor | Pérez Pérez José Juan

3CM6

Semestre 18/19-1

Reportes para junio de 2018

Índice de Contenido

Reporte de Práctica 5

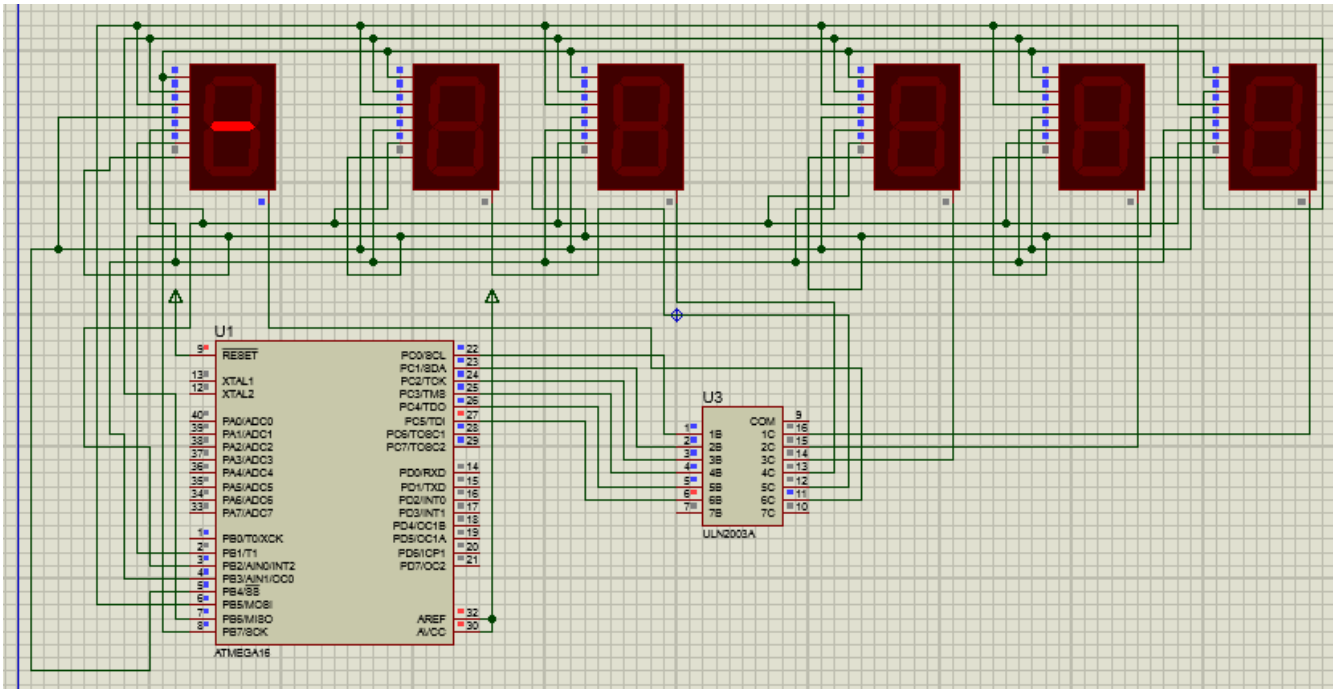
OBJETIVO

Aprender a programar una persistencia de la visión eficaz y sin interrupciones para que la leyenda se lea correctamente y pueda ser entendible.

CODIGO

```
1. .include "m16def.inc"
2.          .def col = R17 ;posicion en el puerto c de multiplexacion
3.          .def dato = R18
4.          .def aux = R19
5.
6. ; ---- Iniciar apuntador de pila ---
7.          ldi aux, low(RAMEND)
8.          out spl, aux
9.          ldi aux, high(RAMEND)
10.              out sph, aux
11.      ; ---------------------------------
12.              ser aux
13.              out DDRB, aux
14.              out DDRC, aux
15.
16.              ldi aux, 0b00000010 ;-
17.              mov r0, aux
18.              ldi aux, 0b11101110 ;A
19.              mov r1, aux
20.              ldi aux, 0b00011100 ;L
21.              mov r2, aux
22.              ldi aux, 0b11111100 ;O
23.              mov r3, aux
24.              ldi aux, 0b01101110 ;H
25.              mov r4, aux
26.              ldi aux, 0b00000010 ;-
27.              mov r5, aux
28.
29.              clr zh
30.
31.      uno:    clr zl
32.              ldi col, 1
33.      dos:    out PORTC, col
34.              ld aux, z+
35.              out PORTB, aux
36.              rcall delay
37.              out PORTB, zh
38.              lsl col
39.              cpi col, 128
40.              breq uno
41.              rjmp dos
42.
43.      delay:  ldi R18, $1F
44.      WGLOOP0:        ldi R19, $2A
45.      WGLOOP1:        dec R19
46.                      brne WGLOOP1
47.                      dec R18
```

```
48.                     brne WGLOOP0
49.                     nop
50.                     ret{
```



CONCLUSIONES

Chávez Chávez Ángel Alexis: Fue interesante de esta práctica el realizar la sincronización de los displays encendidos con la salida, permitiendo pocas conexiones para el mostrar un mensaje aprovechando la capacidad de visión humana.

Flores Alvarado Diego Armando: Otro salto, vamos escalando poco a poco. Esta práctica la considero un híbrido de las cuatro anteriores, creo que eso me agrada, que todo lo que vemos en prácticas anteriores lo vamos aplicando. Me gustó mucho el uso de los displays para poder mostrar un mensaje. Esperemos las siguientes prácticas sigan así.

Hernández López Laura Pamela: En esta práctica trabajamos bastante y aunque no tuvimos los mejores resultados considero que aprendí un mucho, además de recordar conocimientos previos adquiridos en las materias predecesoras a esta ya que para llegar esta materia tuvimos que aprender el uso y aplicación de compuertas básicas, el cual es conocimiento que adquirimos al cursar otras materias.

Reporte de Práctica 6

OBJETIVO

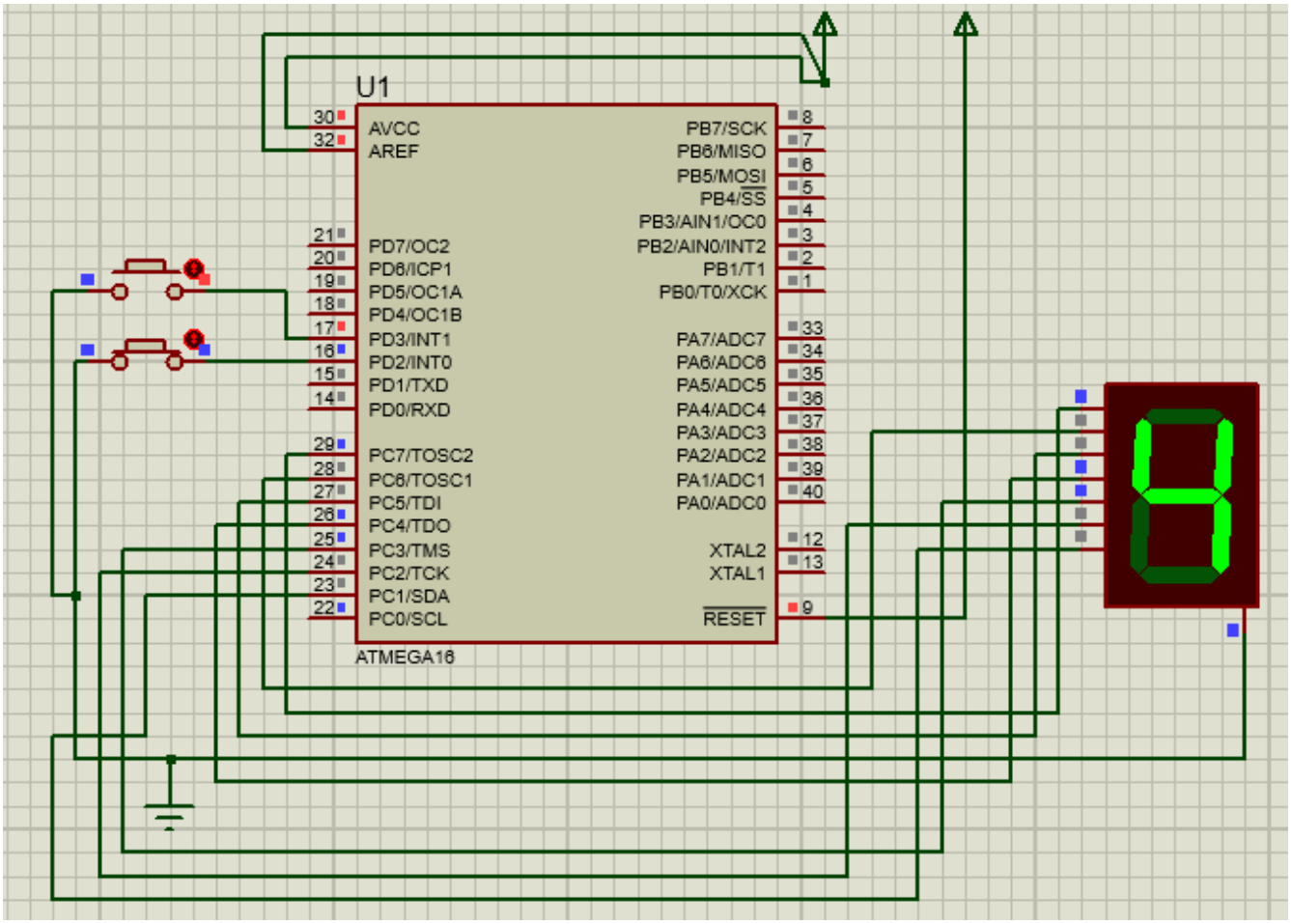
Aprender la forma correcta de hacer un contador y hacer un buen uso del sensado de la interrupción, logrando así un conteo ascendente y descendente que sea acertado en ambos casos.

CODIGO

```
1. .include "m16def.inc"
2. .def aux = r16
3. .def cont = r17
4. .def contdec = r18
5.          rjmp p6
6.          nop
7.          rjmp baja
8.          nop
9.          rjmp sube
10.
11.       p6:     ldi aux, low(RAMEND)
12.               out spl, aux
13.               ldi aux, high(RAMEND)
14.               out sph, aux
15.               ser aux
16.               out DDRC, aux
17.               ldi aux, $0C ;--- 0000 1100
18.               out PORTD, aux
19.               ldi aux, $0A ;--- 0000 1010 Sensado de la interrupcion
20.               out mcucr, aux
21.               ldi aux, $C0 ;--- 1100 0000
22.               out gicr, aux
23.               rcall crgtbl
24.               sei
25.               clr cont
26.               clr zh
27.
28.       loop:   rcall deco
29.               out PORTC, contdec
30.               rjmp loop
31.
32.       sube:   rcall delay
33.               inc cont
34.               cpi cont, 10
35.               brne uno
36.               clr cont
37.
38.       uno:    reti
39.
40.       baja:   rcall delay
41.               dec cont
42.               brpl dos
43.               ldi cont, 9
44.
45.       dos:    reti
46.
47.       crgtbl: ldi r20, 0b11111100     ;0
```

```
48.             ldi r21, 0b01100000     ;1
49.             ldi r22, 0b11011010     ;2
50.             ldi r23, 0b11110010     ;3
51.             ldi r24, 0b01100110     ;4
52.             ldi r25, 0b10110110     ;5
53.             ldi r26, 0b10111110     ;6
54.             ldi r27, 0b11100000     ;7
55.             ldi r28, 0b11111110     ;8
56.             ldi r29, 0b11100110     ;9
57.             ret
58.
59.     deco:   ldi zl, 20
60.             add zl, cont
61.             ld contdec, z
62.             ret
63.
64.     delay: ;---50ms---;
65.             ldi r20, 65
66.             ldi r19, 238
67.     L1:
68.             dec r19
69.             brne L1
70.             dec r20
71.             brne L1
72.             ldi r20, 0b11111100
73.             rjmp PC+1
```

```
74.                 ret
```



CONCLUSIONES

Chávez Chávez Ángel Alexis: Poner en práctica las interrupciones en el atmega16 nos permite implementar eventos que reinicien o bien hagan empezar un proceso bajo ciertas condiciones, tenemos que reiniciar siempre el apuntador a pila.

Flores Alvarado Diego Armando: Las famosas interrupciones. Este tema en clase no se me hizo tan complicado de entender, pero lo sentí diferente en la práctica, aquí se me complicó y a eso hay que agregarle los problemas que tuvimos con el microcontrolador en prácticas pasadas. Al final con la ayuda de mi equipo pudimos terminarla con éxito. Las famosas interrupciones. Este tema en clase no se me hizo tan complicado de entender, pero lo sentí diferente en la práctica, aquí se me complicó y a eso hay que agregarle los problemas que tuvimos con el microcontrolador en prácticas pasadas. Al final con la ayuda de mi equipo pudimos terminarla con éxito.

Hernández López Laura Pamela: En la realización de esta práctica tuvimos retos al igual que en las practicas anteriores, aunque fueron un poco mayores debido a que el microcontrolador que adquirimos mi equipo y yo no funciono al cien. Pero con un poco más de esfuerzo y trabajo en equipo logramos los objetivos de la práctica que fueron adquirir nuevos conocimientos y hacer que el circuito funcione correctamente.

Reporte de Práctica 7

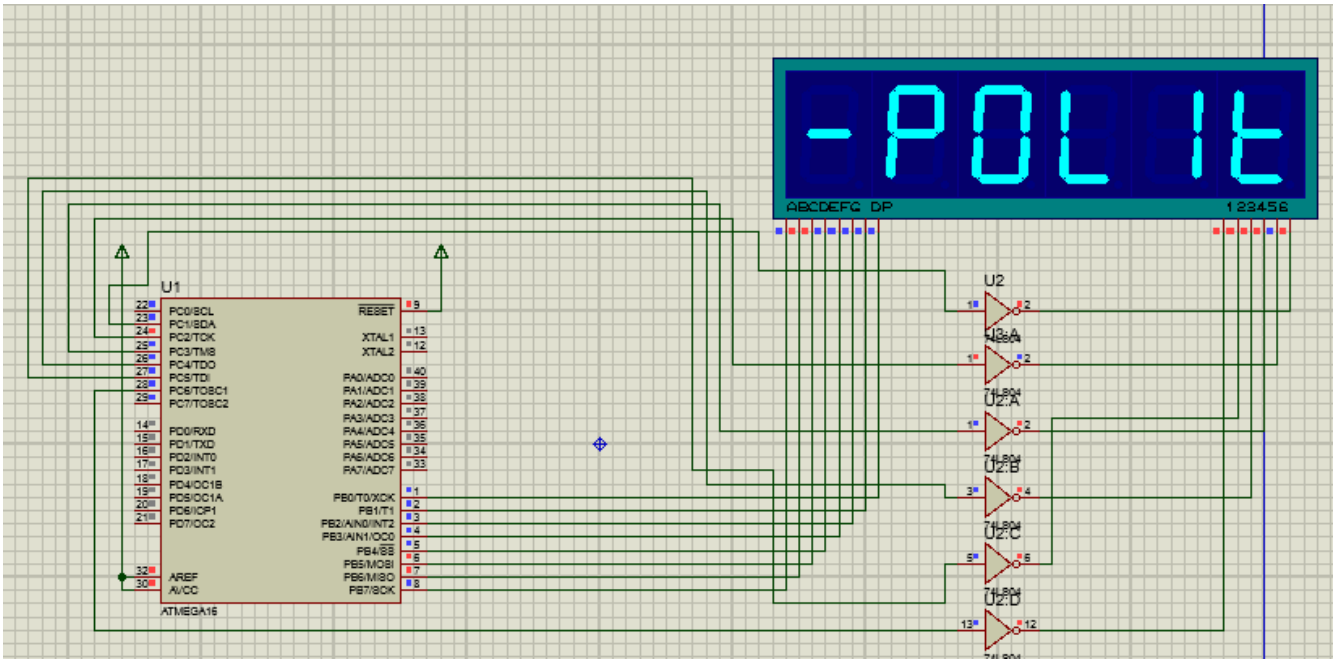
OBJETIVO

Aprender el funcionamiento de un barrido de texto y lograr el desplazamiento exitoso de caracteres en un conjunto de display de siete segmentos.

CODIGO

```
1.      .include "m16def.inc"
2.      .def aux = r16
3.      .def col = r17
4.
5.  reset:
6.              rjmp scrll1
7.              .org $010
8.              rjmp corre
9.              .org $012
10.                  rjmp barre
11.
12.      scrll1:
13.                  ldi aux, low(ramend)
14.                  out spl, aux
15.                  ldi aux, high(ramend)
16.                  out sph, aux
17.                  rcall config_io
18.                  rcall carga_msg
19.                  clr zh
20.                  clr zl
21.                  ldi col, 1
22.                  out portc, col
23.                  ld aux, z
24.                  out portb, aux
25.
26.      uno:
27.                  rjmp uno
28.
29.      config_io:
30.                  ser aux
31.                  out ddrb, aux
32.                  out ddrc, aux
33.                  ldi aux, 3
34.                  out tccr0, aux
35.                  ldi aux, 2
36.                  out tccr1b, aux
37.                  ldi aux, $05
38.                  out timsk, aux
39.                  ldi r18, 193
40.                  sei
41.                  ret
42.
43.      carga_msg:
44.                  ldi aux, 0b00000010 ; -
45.                  mov r0, aux
46.
47.                  ldi aux, 0b11111100 ; o
```

```
48.                 mov r1, aux
49.                 ldi aux, 0b10011100 ; c
50.                 mov r2, aux
51.                 ldi aux, 0b01100000 ; i
52.                 mov r3, aux
53.                 ldi aux, 0b00101010 ; n
54.                 mov r4, aux
55.                 ldi aux, 0b10011100 ; c
56.                 mov r5, aux
57.                 ldi aux, 0b10011110; e
58.                 mov r6, aux
59.                 ldi aux, 0b00011110 ; t
60.                 mov r7, aux
61.                 ldi aux, 0b01100000 ; i
62.                 mov r8, aux
63.                 ldi aux, 0b00011100 ; l
64.                 mov r9, aux
65.                 ldi aux, 0b11111100 ; o
66.                 mov r10, aux
67.                 ldi aux, 0b11001110 ; p
68.                 mov r11, aux
69.                 ldi aux, 0b00000010 ; -
70.                 mov r12, aux
71.                 ret
72.
73.     barre:
74.                 out tcnt0, r18
75.                 ;out portb, zh
76.                 inc zl
77.                 lsl col
78.                 cpi col, 128
79.                 brne dos
80.                 ldi col, 2
81.                 clr zl
82.
83.     dos:
84.                 out portc, col
85.                 ld aux, z
86.                 out portb, aux
87.                 reti
88.
89.     corre:
90.                 mov r13, r12
91.                 mov r12, r11
92.                 mov r11, r10
93.                 mov r10, r9
94.                 mov r9, r8
95.                 mov r8, r7
96.                 mov r7, r6
97.                 mov r6, r5
98.                 mov r5, r4
99.                 mov r4, r3
100.                mov r3, r2
101.                mov r2, r1
102.                mov r1, r0
103.                mov r0, r14
104.                mov r14, r15
105.                mov r15, r13
106.                reti
```



CONCLUSIONES

Chávez Chávez Ángel Alexis: Practicamos la sincronización de la salida sincronizada y un corrimiento en el apuntador de salida. Como comentario en la simulación tuvimos que utilizar un display que ya integraba los 6 display de 7 segmentos dado que en la simulación de la práctica 5 el despliegue de los valores era lento.

Flores Alvarado Diego Armando: La práctica 5 pero en versión mejorada. Me gustó el corrimiento que se le agregó a esta práctica, además de que tuvimos la opción de poner el mensaje que quisiéramos en los displays. Fue menos complicada puesto que supimos lidiar con las prácticas anteriores.

Hernández López Laura Pamela: Esta práctica me gusto bastante ya que en ella pudimos realizar algo que se ve con más frecuencia en las calles con aplicaciones que ayudan a la sociedad, ya que los mensajes que se desplazan en displays son súper útiles y aplicables a muchas cosas en la actualidad.

Reporte de Práctica 8

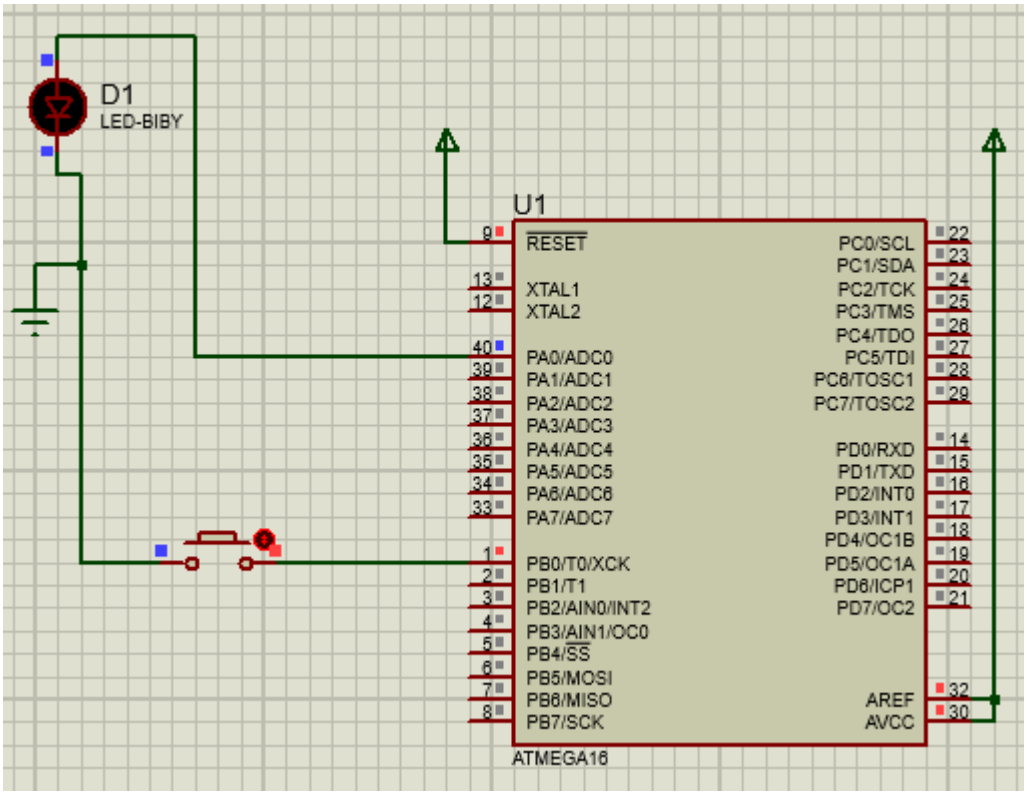
OBJETIVO

Aprender el funcionamiento, usabilidad y aplicación de los timers aplicando los conocimientos adquiridos en la programación de microcontroladores para tener una visualización acertada de un LED.

CODIGO

```
1.    .include "m16def.inc"
2. .def aux = r16
3. .def aux_tono = r21
4.
5. reset:
6.     rjmp main
7.     .org $008 ; timer 2, el de los 440 Hz
8.     rjmp suena
9.     .org $010
10.          rjmp para; timer 1, el de los 5 segundos
11.            .org $012
12.          rjmp empieza_suena ;timer 0, el que cuenta clientes
13.
14.      main:
15.          ldi aux, low(ramend) ; iniciar apundador de pila
16.          out spl, aux
17.          ldi aux, high(ramend)
18.          out sph, aux
19.          rcall config_io ; configurar timers y puertos
20.              clr zh
21.              clr zl
22.          ldi aux, $01
23.          out ddra, aux ; usamos el puerto A, pin 0 como salida de la señal
   de la bocina
24.          ldi aux, $01
25.          out portb, aux ; se activa el pull up del pin 0 del puerto B, que
   es la entrada externa para el timer0
26.
27.      loop:
28.              nop
29.              rjmp loop
30.
31.      config_io:
32.          ldi aux, 6 ; detecta entrada externa en flanco de baja
33.          out tccr0, aux
34.          ldi aux, 0 ; no empezar aún el contador de los 5 segundos
35.          out tccr1b, aux
36.          ldi aux, $0; no empezar aún el sonido de 440 Hz
37.          out tccr2, aux
38.          ldi aux, 0b01000101 ; activar interrupciones de desbordamiento para
   los 3 timers
39.          out timsk, aux
40.              ldi r18, 251 ; Empezar en 251 y llegar al 256 en el timer 0
   para contar 5 clientes
41.              ldi r19, 0b11101100 ; parte alta para que el timer 1 empieze en
   60535 y llegue a $ffff,durando 5 seg, contando cada 1024 ciclos
42.          ldi r20, 0b11101101; parte baja
```

```
43.             out tcnt1l, r20 ; se le asigna el valor inicial al timer1
44.         out tcnt1h, r19
45.             ldi r22, $01 ; usaremos este registro para ayudar a alternar
    entre 0 y 1 la señal a la bocina
46.             out tcnt0, r18; asignar el valor de inicio al contador de
    clientes (timer0)
47.
48.         sei
49.
50.         ret
51.
52.     empieza_suena:
53.             out tcnt0, r18; reiniciar el timer0 a 251 (conteo de clientes)
54.         ldi aux, 5 ; para los 5 segundos, contar cada 1024 ciclos
55.         out tccr1b, aux
56.         ldi aux, 7 ; para los 440 Hz, el timer2 incrementa en 1 cada 1024
    ciclos de reloj
57.         out tccr2, aux
58.             ldi r23, 254
59.             out tcnt2, r23 ; empieza en 254 para llegar a 255
60.         out tcnt1l, r20 ; se le asigna el valor inicial al timer1
61.         out tcnt1h, r19
62.         ldi aux_tono, $01; Valor inicial de 1 para la señal de los 440Hz
63.             reti
64.
65.     suena:
66.             out tcnt2, r23; reiniciar el timer2 a 254
67.         out porta, aux_tono ; señal de la bocina
68.         eor aux_tono, r22 ; alternamos entre 1 y 0
69.             reti
70.
71.     para:
72.             ldi aux, 0 ; detenemos el timer1 de los 5 segundos
73.         out tccr1b, aux
74.         ldi aux, $0 ; detener el timer2 de los 440Hz
75.         out tccr2, aux
76.             reti
```



CONCLUSIONES

Chávez Chávez Ángel Alexis: Integrar un contador nos da oportunidad a desarrollar prácticas más elaboradas como esta, implementamos el manejo de interrupciones y counters.

Flores Alvarado Diego Armando: Una práctica más sencilla que las que se venían haciendo. Trabajamos de nuevo con las interrupciones, creo que ya pude entender mejor el tema, aunque sigo teniendo mis dudas.

Hernández López Laura Pamela: Esta práctica, aunque fue sencilla pienso que es de las que se le puede encontrar mayores aplicaciones ya que para casi todo en la industria son necesarios los contadores y las interrupciones usando los timers para que las cosas funcionen correctamente y me gusto aprender más sobre esto.

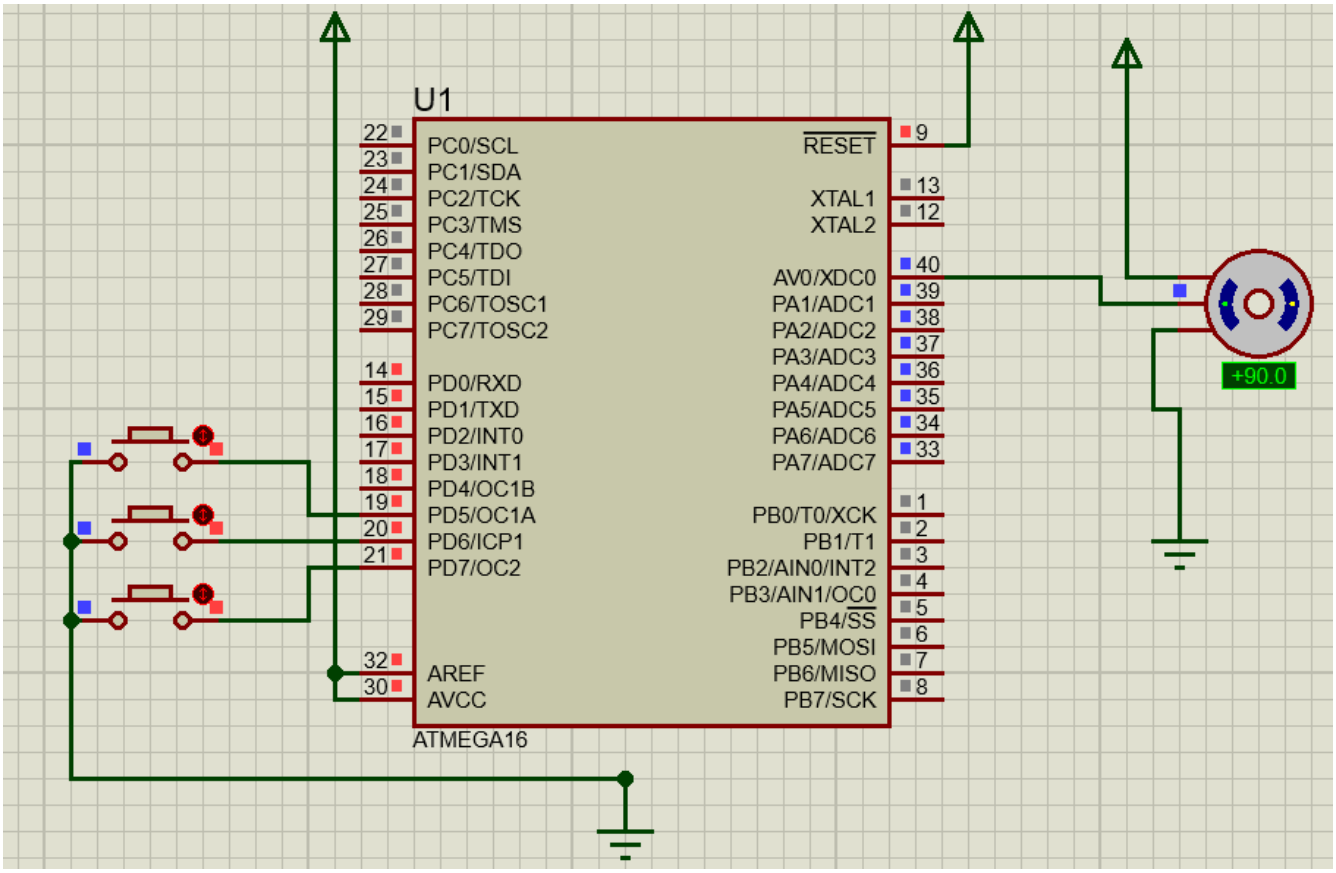
Reporte de Práctica 9

OBJETIVO

Aprender a realizar el control de un servo motor 5g90, todo esto mediante el uso de macros. Así aprender a declarar una macro haciendo uso de la directiva y las instrucciones.

CODIGO

```
1. .def aux = r16
2. .macro pulso
3.            sbi porta, 0
4.            ldi aux, @0
5.    loop1:
6.            rcall delay
7.            dec aux
8.            brne loop1
9.            cbi porta, 0
10.                ldi aux, @1
11.         loop2:
12.                rcall delay
13.                dec aux
14.                brne loop2
15.                rjmp otro
16.     .endm
17.
18.     main:
19.             ldi aux, low(ramend)
20.             out spl, aux
21.             ldi aux, high(ramend)
22.             out sph, aux
23.             ser aux
24.             out ddra, aux
25.             out portd, aux
26.     otro:
27.             sbis pind, 5
28.             rjmp cero
29.             sbis pind, 6
30.             rjmp uno
31.             sbis pind, 7
32.             rjmp dos
33.             rjmp otro
34.     cero:
35.             pulso 2, 38
36.     uno:
37.             pulso 3, 37
38.     dos:
39.             pulso 4, 36
40.
41.     delay:
42.             ldi  r18, 166
43.     L1: dec  r18
44.         brne L1
45.         rjmp PC+1
46.             ret
```



CONCLUSIONES

Chávez Chávez Ángel Alexis: Utilizamos 3 botones para hacer funcionar el servomotor girándolo a 3 diferentes posiciones, observando en la simulación como un voltmetro en el servo nos indicaba una variación en la salida de voltaje.

Flores Alvarado Diego Armando: Esta práctica me recordó a una que hice en la asignatura de Electrónica Analógica sólo que ahora lo hicimos con ayuda de un microcontrolador. Fue interesante ver el uso de las señales PWM para controlar el servomotor y el uso de los macros y cómo pueden ser reutilizados para otras funciones.

Hernández López Laura Pamela: En esta práctica pusimos en práctica más de los conocimientos que adquirimos con anterioridad en otras materias, al usar la bocina pudimos ver otras aplicaciones en las variaciones de las salidas de voltaje y del funcionamiento de los servomotores.