



**INSTITUTO POLITÉCNICO  
NACIONAL**  
ESCUELA SUPERIOR DE CÓMPUTO



**“Manejo de EEPROM”**

**Integrantes del Equipo:**

Contreras Cardoso Adolfo

Martínez Alvarado Bryan Alexis

Maya Martínez Alonso Rubén

Pérez Gómez Santiago

**Grupo:**

3CM15

**Profesor:**

Ing. José Juan Pérez Pérez

**Asignatura:**

# Introducción a los Microcontroladores

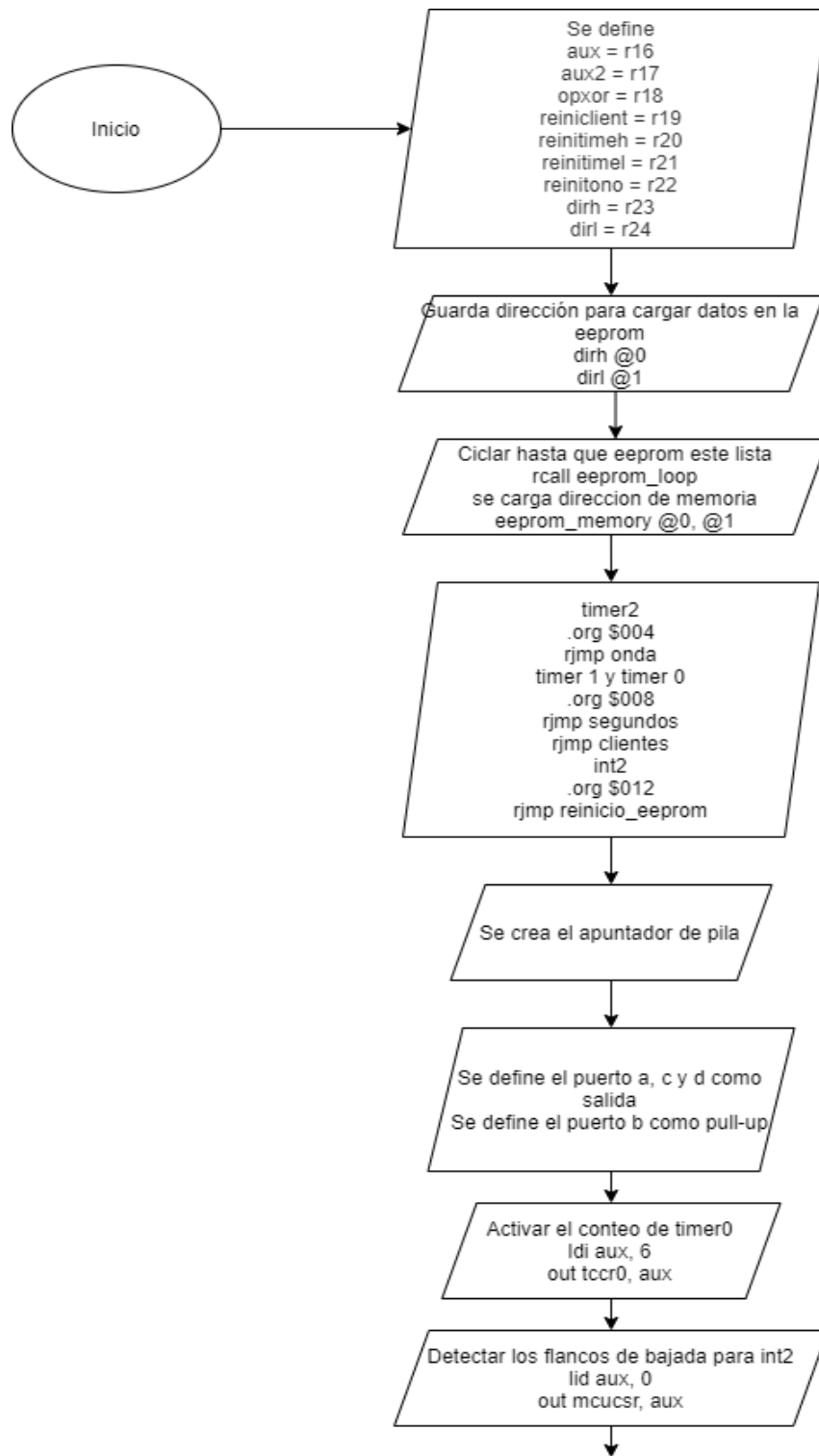
## Planteamiento del Problema

Para el circuito usado en el ejercicio de contador de clientes, agregar 2 displays bcd 7 seg en el puerto D, en estos displays se mostrará la cuenta de clientes ganadores. La primera vez que funcione el circuito mostrará "00" y por cada cliente ganador se irá incrementado en forma decimal hasta "99". Cada vez que se incrementa esta cuenta se debe guardar en la EEPROM este valor, si se apaga el sistema o se pulsa "RESET" debe iniciar la cuenta con el último valor guardado, si se pulsa INT2 la cuenta deberá iniciarse a "00"

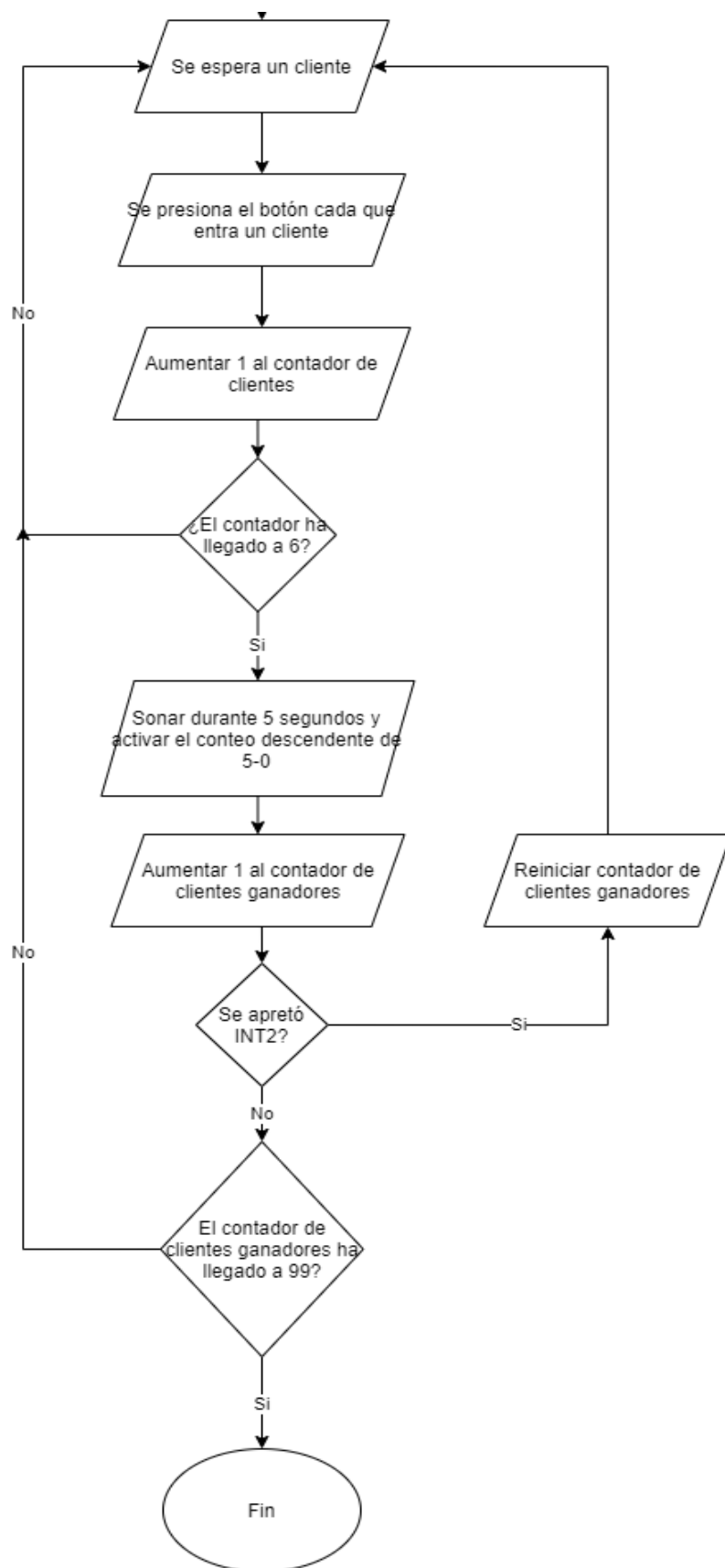
## Desarrollo

Para realizar la implementación se utilizó como base el programa de la práctica anterior añadiendo el uso de la interrupción externa 2, además, para conseguir lo planteado en el problema, se utilizó la lectura y escritura de la EEPROM por medio de macros.

El diagrama de flujo correspondiente al resultado es el siguiente:



**Figura 1.** Diagrama de Flujo Parte 1



**Figura 2.** Diagrama de Flujo Parte 2

El respectivo código en ensamblador es el siguiente:

```
;programa para conteo de clientes con timers
;usando eeprom e interrupcion externa 2

;timer0 = clientes = conteo de clientes
;timer1 = onda = generacion de onda 440hz (2.27ms)
;timer2 = segundos = duracion de 5 segundos

;valor a cargar en timer1
;19531 x 256prescala = 4,999,936ciclos = 4.99ms
;65536 - 19531 = 46005 = $B3B5

;valor a cargar en timer2 1.1363 (mitad de 2.27ms)
;142 x 8prescala = 1136ciclos = 1.136ms
;256 - 142 = 114

.include "m8535def.inc"

.def aux = r16
.def aux2 = r17
.def opxor = r18

;para timer0
.def reiniclient = r19

;para timer1
.def reinitimeh = r20
.def reinitimel = r21
```

**Figura 3.** Primera Sección de Código

```
;para timer2
.def reinitono = r22

;para eeprom
.def dirh = r23
.def dirl = r24

;para reducir codigo de las macros lectura y escritura
;se espera a que eeprom este lista y despues se carga
;la direccion de memoria
.macro eeprom_memory

    ;guardando direccion para cargar datos en la eeprom
    ldi dirh,@0
    ldi dirl,@1

    ;ciclando hasta que eeprom este lista
    rcall eeprom_loop

    ;EEAR = registro de direccion
    ;direccion de eeprom a leer o escribir
    out EEARH,dirh
    out EEARL,dirl

.endm
```

**Figura 4.** Segunda Sección del Código

```

;para escritura en la eeprom
.macro eeprom_write

    ;cargando direccion de memoria
    eeprom_memory @0,@1

    ;EEDR = registro de datos
    ;para guardar o recibir datos de lectura o escritura
    out EEDR,@2

    ;EECR = registro de control para lectura o escritura
    ;EEMWE = indica que se habilite la escritura
    sbi EECR,EEMWE

    ;EWE = indica que se inicie la escritura
    sbi EECR,EWE

.endm

```

**Figura 5.** Tercera Sección del Código

```

;para lectura en la eeprom
.macro eeprom_read

    eeprom_memory @0,@1

    ;EECR = registro de control para lectura o escritura
    ;EEMWE = indica que se inicie la lectura
    sbi EECR,EERE

    ;guardando valor leído de la eeprom
    in @2,EEDR

.endm

inicio:
    rjmp main

;timer2
.org $004
rjmp onda

;timer1 y timer0
.org $008
rjmp segundos
rjmp clientes

```

**Figura 6.** Cuarta Sección del Código

```

        ;para int2
        .org $012
        rjmp reinicio_eeeprom

main:
        ;apuntador de pila
        ldi aux,low(ramend)
        out spl,aux
        ldi aux,high(ramend)
        out sph,aux

        ;puerto a,c y d como salida
        ser aux
        out ddra,aux
        out ddrc,aux
        out ddrd,aux

        ;puerto b como pull-up
        out portb,aux

        ;habilitar interrupciones por
        ;desborde de timer counter
        ;0,1 y 2, pero estan detenidos
        ldi aux,$45
        out tmsk,aux

```

**Figura 7.** Quinta Sección del Código

```

;activando el conteo de timer0
ldi aux,6
out tccr0,aux

;detectando los flancos de bajada para int2
ldi aux,0
out mcucsr,aux

;habilitando int2 con gicr
;(se puede habilitar int2 con otro uno)
ldi aux,0b00100000
out gicr,aux

;habilitando globalmente
;las interrupciones
sei

;reinicio para conteo del display
ldi aux2,255

;reiniciando conteo de clientes
ldi reiniclient,250
out tcnt0,reiniclient

;para generar señal cuadrada
ldi opxor,1

```

**Figura 8.** Sexta Sección del Código

```

;valor inicial timer1
ldi reinitimeh,$B3
ldi reinitime1,$B5

;valor inicial timer2
ldi reinitono,114

;leyendo eeprom
eeprom_read $00,$00,aux
out portd,aux

;ciclando el main
ciclar:
    ;se cuenta cada cliente
    sbic pinb,0
    rcall display
    rjmp ciclar

```

**Figura 9.** Séptima Sección del Código

```

;codigo para actualizar el valor del display
;con cada pulsacion del push button
display:
    ;realizando conteo del 5 al 0
    in aux,tcnt0
    sub aux2,aux
    out portc,aux2

    ;reinicio para conteo del display
    ldi aux2,255

    ret

;codigo para activar speaker al contar 6 clientes
clientes:
    ;cargando eeprom
    in aux,portd
    inc aux
    eeprom_write $00,$00,aux
    out portd,aux

    ;reiniciando timer 0
    out tcnt0,reiniciant

    ;activando el conteo de timer1
    ;(256 veces 65.54ms = 16.77seg)
    ldi aux,4
    out tccr1b,aux

```

**Figura 10.** Octava Sección del Código



```

;activando el conteo de timer2
;(8 veces 0.26ms = 2.08ms)
ldi aux,2
out tccr2,aux

;reiniciando timer1
out tcnt1h,reinitimeh
out tcnt1l,reinitimel

;reiniciando timer2
out tcnt2,reinitono

reti

;codigo para generar onda de 440hz
onda:
;reiniciando timer2
out tcnt2,reinitono

;or exclusiva para generar señal
in aux,pina
eor aux,opxor
out porta,aux

reti

```

**Figura 11.** Novena Sección del Código

```

;codigo para mantener onda por 5 segundos
segundos:
;desactivando el timer1 y timer2
ldi aux,0
out tccr1b,aux
out tccr2,aux
reti

;reiniciando eeprom al usar int2
reinicio_eeprom:
;cargando eeprom
ldi aux,0
eeprom_write $00,$00,aux
out portd,aux
reti

;para macros de lectura y escritura de eeprom
eeprom_loop:
;EECR = registro de control
;EWE = bit que indica si eeprom esta lista para escritura
;esperar a que el bit EWE tenga un cero
sbic EECR,EWE
rjmp eeprom_loop
ret

```

**Figura 12.** Décima Sección del Código

Para la simulación, se muestran algunos de los valores en el display, también la generación de sonido en el speaker y el resultado en el osciloscopio:

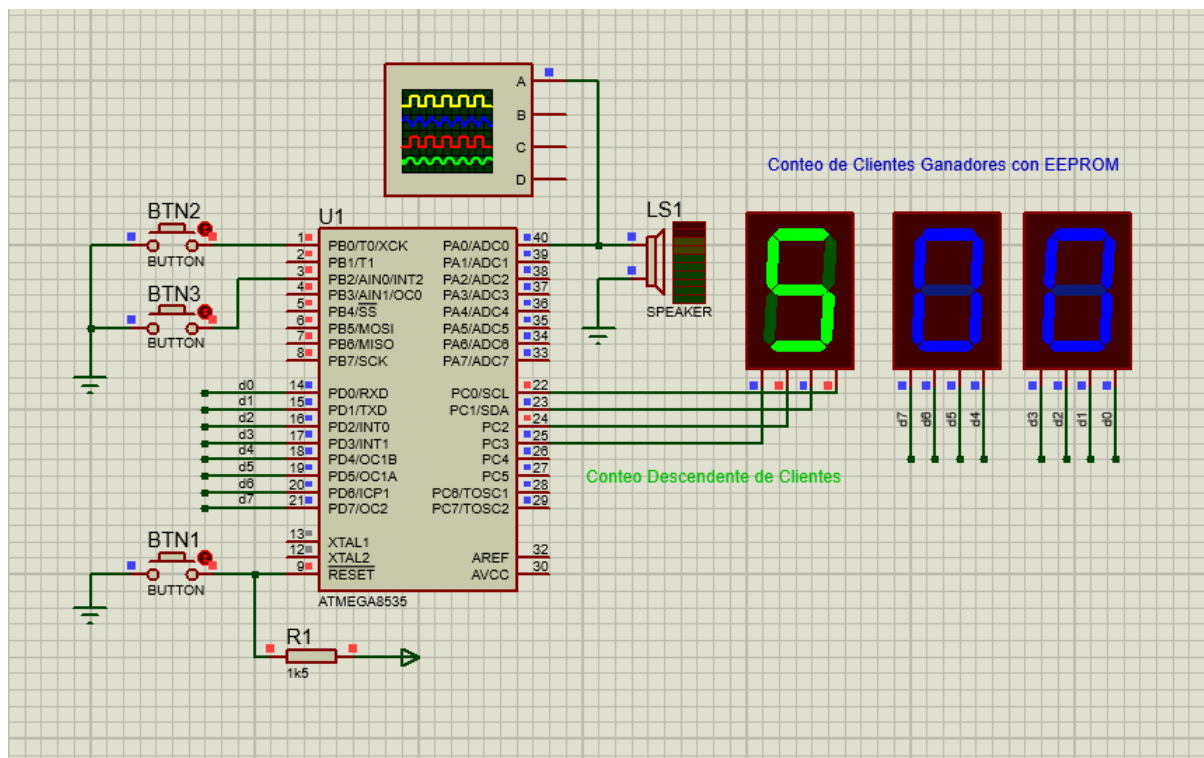


Figura 13. Valor 00 en los display de la EEPROM

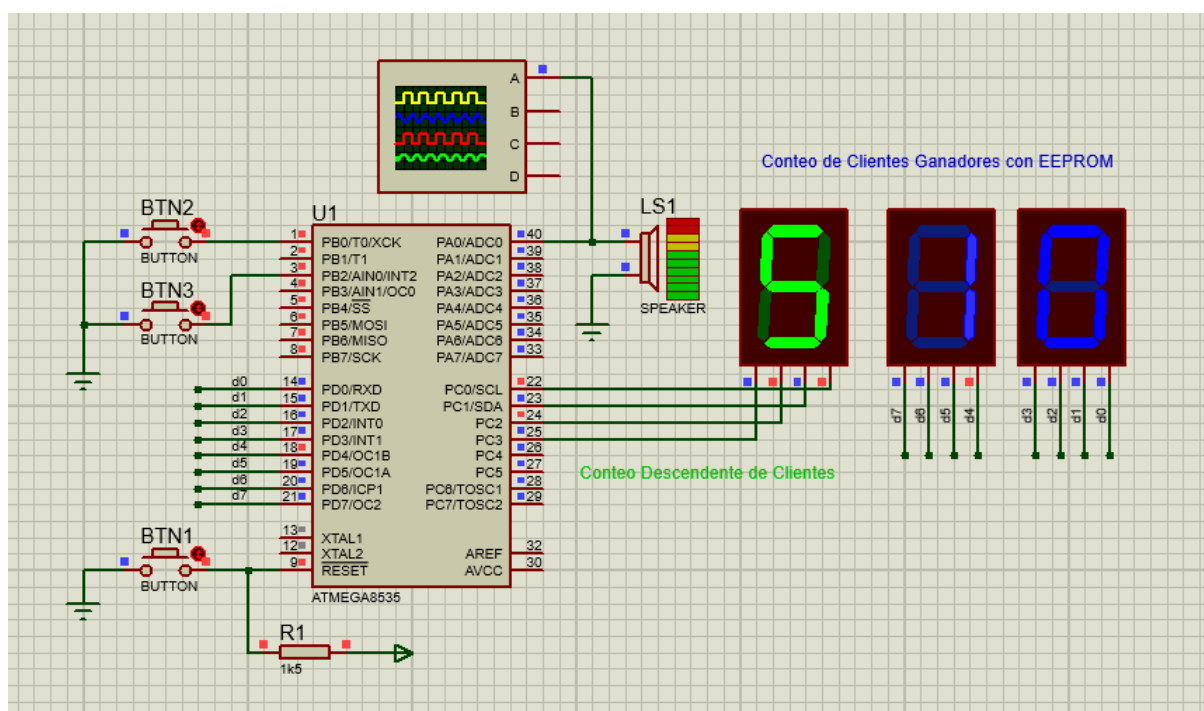


Figura 14. Valor 10 en los display de la EEPROM

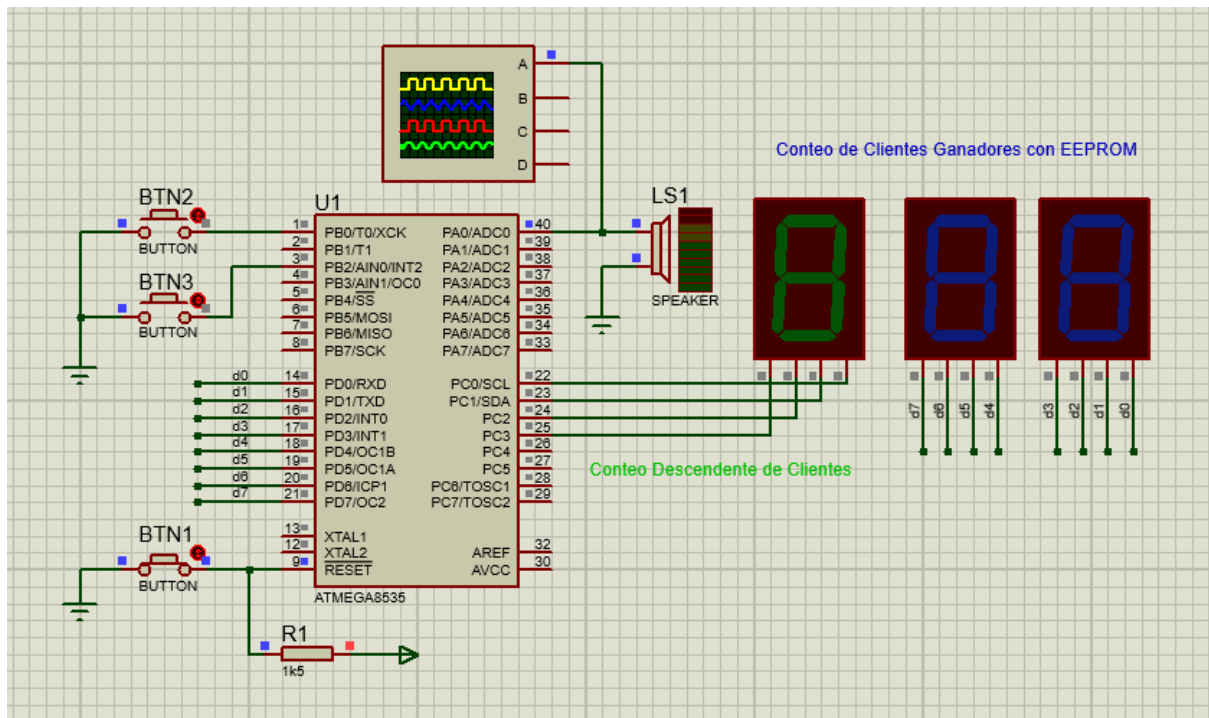


Figura 15. Se presiona reset para probar la EEPROM

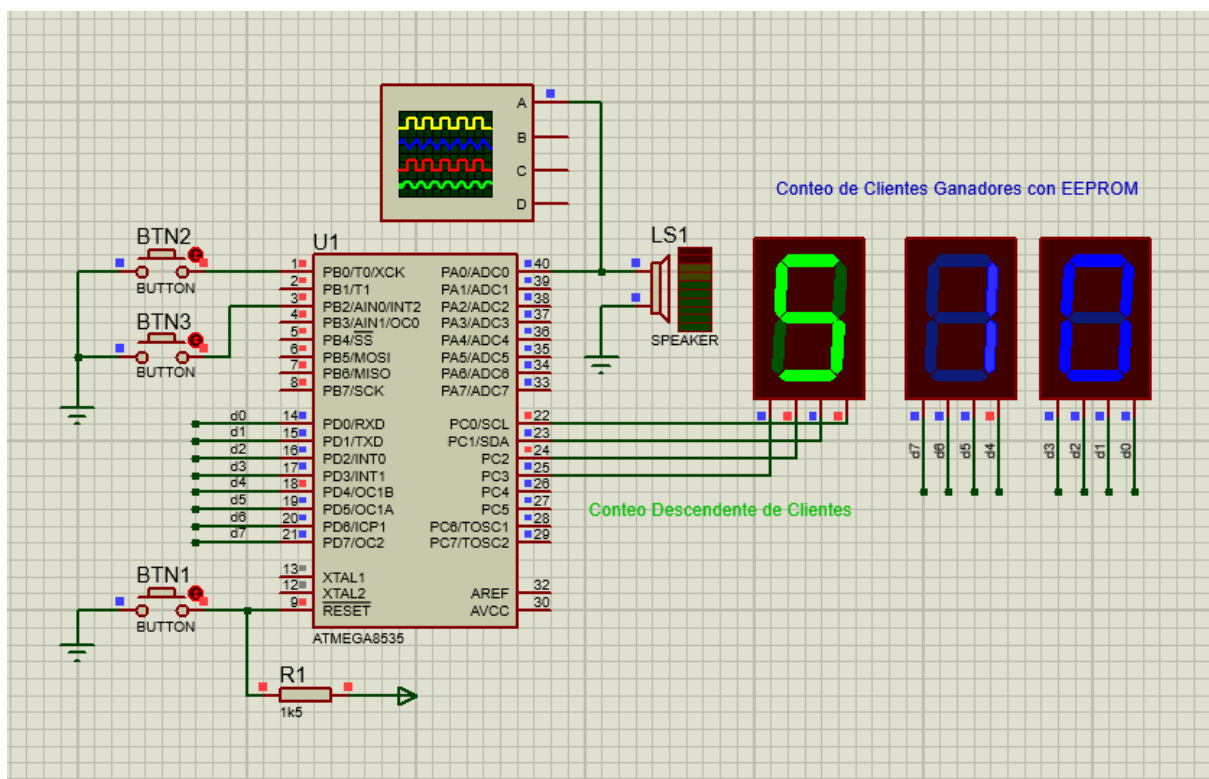
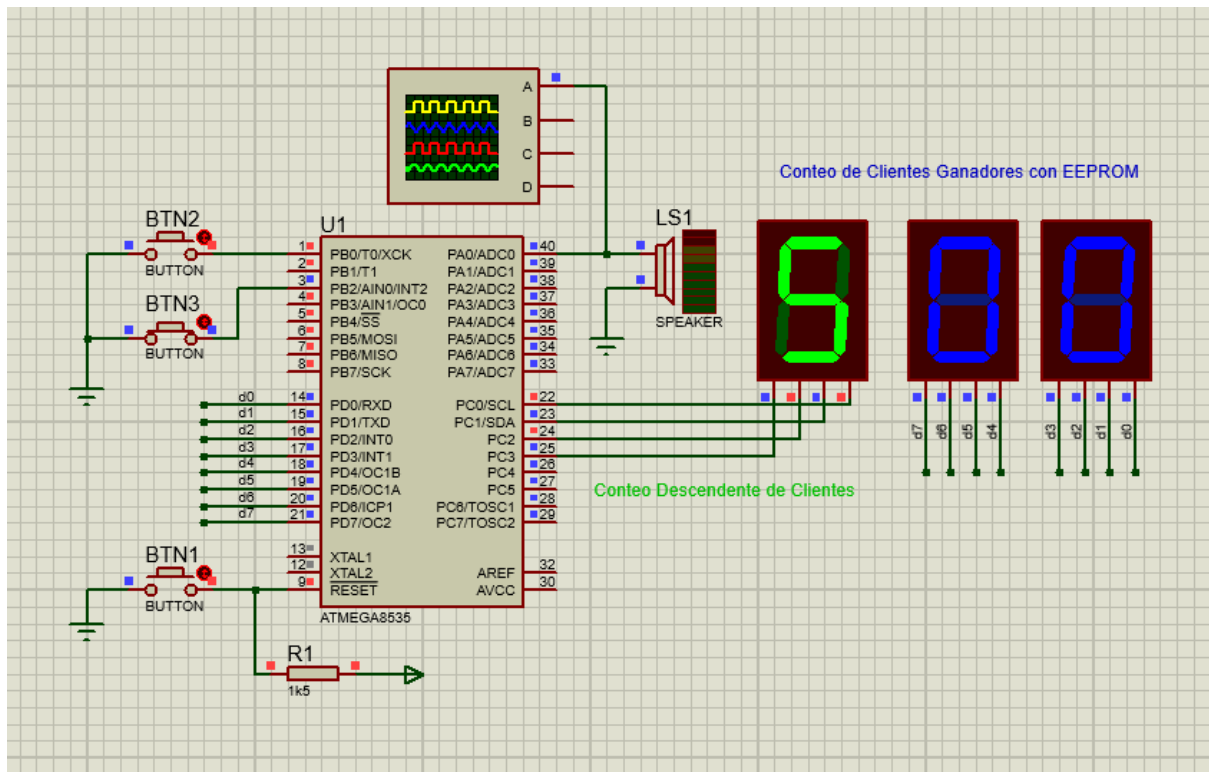


Figura 16. Después de presionar el reset sin cambio alguno en los display de la EEPROM



**Figura 17.** Reinicio al presionar interrupción externa 2

## Conclusiones Individuales

### Contreras Cardoso Adolfo

La realización de la práctica fue algo nuevo en lo personal ya que utilizamos funciones del microcontrolador que hemos usado antes para implementar unas nuevas, con el apoyo de mi equipo se pudo implementar y lograr los objetivos que la práctica nos pedía.

### Martínez Alvarado Bryan Alexis

Durante el desarrollo de esta práctica se puso a prueba el funcionamiento de la memoria eeprom, macros, displays, timer, entradas físicas y salidas, aplicando lo visto durante la clase, se logró poner en práctica los conocimientos adquiridos en la parte teórica y se logró observar el funcionamiento del microcontrolador AtMega8535 junto con otros componentes electrónicos, pudiendo analizar cómo es que funcionan.

### Maya Martínez Alonso Rubén

El uso de la EEPROM nos ayuda a pensar en soluciones en las cuales se necesita guardar algún valor o algo que se vaya a necesitar en un futuro, sea si se hace reset o no. Este es el fundamento para la solución de varios problemas e implementaciones usando el microcontrolador AtMega8535, es interesante pensar en cómo funcionará en otros micros para así poder llevar soluciones.

### Pérez Gómez Santiago

Dentro del microcontrolador AtMega8535 existen una gran cantidad de funcionalidades, tal como el uso de timers, interrupciones, macros y la memoria EEPROM, características sin las cuales no hubiera sido posible realizar este trabajo, esto en conjunto con lo realizado en prácticas anteriores, es así como se obtuvieron resultados satisfactorios. De manera personal, siempre resulta útil e interesante aprender más acerca de las herramientas que nos brinda este microcontrolador.