

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

“Sistema unifila”

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

Desarrollo

Para el circuito mostrado, escribir un programa en "C" o ensamblador para que funcione como parte de un sistema "Unifila"

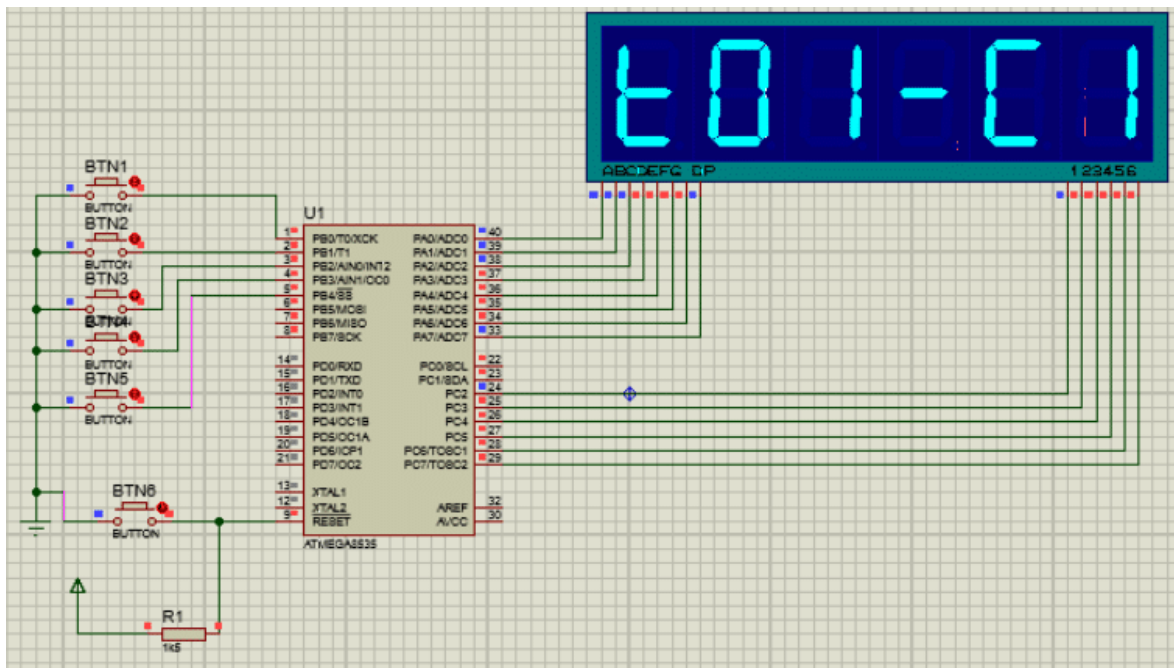


Figura 1. Circuito de ejemplo

El diagrama de flujo correspondiente al resultado es el siguiente:

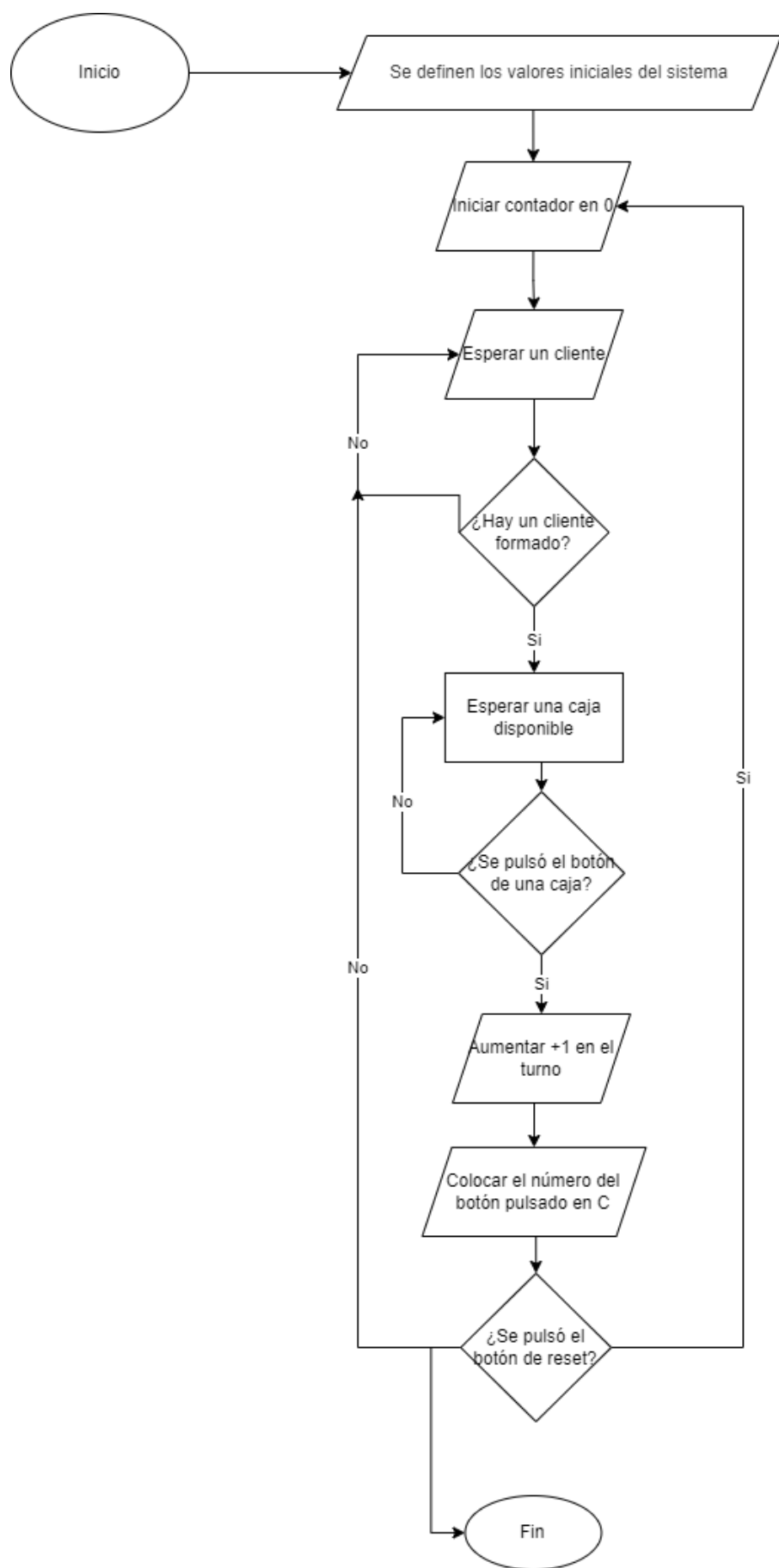


Figura 2. Diagrama de flujo

El respectivo código en ensamblador es el siguiente:

```
.include "m8535def.inc"

.def aux = r16
.def anillo = r17
.def caja = r18
.def unidades = r19
.def decenas = r20
.def flag = r21

;puertos a y c como salidas
;puerto b como entrada
ser aux
out ddra,aux
out ddrc,aux
out portb,aux

reset:
    rjmp main

;timer1 para contador de anillo
.org $008
rjmp contadorAnillo
```

Figura 3. Primer sección de código

```

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

    ;timer1 sin preescalado
    ldi aux,2
    out tccr1b,aux

    ;habilitando desborde timer1
    ldi aux,4
    out tmsk,aux

    ;habilitando interrupciones globalmente
    sei

    ;reiniciando variables
    ldi anillo,1
    ldi caja,1
    clr unidades
    clr decenas
    clr flag

    ;reiniciando timer1
    ldi aux,255
    out tcnt1h,aux

    ldi aux,25
    out tcnt1l,aux

```

Figura 4. Segunda sección de código

```

loop:
    ;cargando una letra distinta
    ;dependiendo el contador de anillo
    sbis portc,0
    ldi aux,$78

    sbis portc,1
    mov aux,decenas

    sbis portc,1
    rcall conversion

    sbis portc,2
    mov aux,unidades

    sbis portc,2
    rcall conversion

    sbis portc,3
    ldi aux,$40

    sbis portc,4
    ldi aux,$39

    sbis portc,5
    rcall actualizarCaja

    ;mostrando valor en displays
    out porta,aux

    rjmp loop

```

Figura 5. Tercera sección de código

```
actualizarCaja:
    ;si se presiona un push button
    ;se muestra un numero de caja distinta
    sbis pinb,0
    ldi caja,1

    sbis pinb,1
    ldi caja,2

    sbis pinb,2
    ldi caja,3

    sbis pinb,3
    ldi caja,4

    sbis pinb,4
    ldi caja,5

    ;validaciones para que al presionar un
    ;push button no se aumente el numero de
    ;turno mas de una vez
    in aux,pinb
    cpi aux,$ff
    breq condicion1
```

Figura 6. Cuarta sección de código

```
    cpi flag,0
    brne condicion1

    ldi flag,1

condicion1:
    cpi flag,1
    brne condicion2

    in aux,pinb
    cpi aux,$ff
    brne condicion2

    ldi flag,2

condicion2:

    cpi flag,2
    brne saltarTurno

    ldi flag,0
    rcall cambioTurno

    ;convirtiendo valores a 7 segmentos
saltarTurno:
    mov aux,caja
    rcall conversion

ret
```

Figura 7. Quinta sección de código


```

cambioTurno:
    ;aumenta el numero de turno
    ;en las unidades
    inc unidades

    cpi unidades,10
    brne regreso

    ;aumenta el numero de turno
    ;en las decenas y se
    ;reinicia en las unidades
    clr unidades
    inc decenas

    cpi decenas,10
    brne regreso

    clr decenas

    regreso: ret

```

Figura 8. Sexta sección de código

```

contadorAnillo:
    ;reiniciando timer1
    ldi aux,255
    out tcnt1h,aux

    ldi aux,25
    out tcnt1l,aux

    ;cargando valor del contador de anillo
    mov aux,anillo
    com aux
    out portc,aux
    lsl anillo

    ;verificando si es necesario reset
    cpi anillo,$40
    brne saltarReinicio

    ;reset del contador de anillo
    ldi anillo,1

    saltarReinicio:
    reti

```

Figura 9. Séptima sección de código

```

;funcion para convertir un numero binario
;a su equivalente en display 7 segmentos
conversion:
    cpi aux,0
    brne saltarCero

    ldi aux,$3f
    rjmp fin

    saltarCero:

    cpi aux,1
    brne saltarUno

    ldi aux,6
    rjmp fin

    saltarUno:

    cpi aux,2
    brne saltarDos

    ldi aux,$5b
    rjmp fin

```

Figura 10. Octava sección de código

```
saltarDos:

    cpi aux,3
    brne saltarTres

    ldi aux,$4f
    rjmp fin

saltarTres:

    cpi aux,4
    brne saltarCuatro

    ldi aux,$66
    rjmp fin

saltarCuatro:

    cpi aux,5
    brne saltarCinco

    ldi aux,$6d
    rjmp fin

saltarCinco:

    cpi aux,6
    brne saltarSeis

    ldi aux,$7d
    rjmp fin
```

Figura 11. Novena sección de código

```
saltarSeis:

cpi aux,7
brne saltarSiete

ldi aux,7
rjmp fin

saltarSiete:

cpi aux,8
brne saltarOcho

ldi aux,$7f
rjmp fin

saltarOcho:

cpi aux,9
brne fin

ldi aux,$6f
rjmp fin

fin: nop

ret
```

Figura 12. Décima sección de código

Simulación:

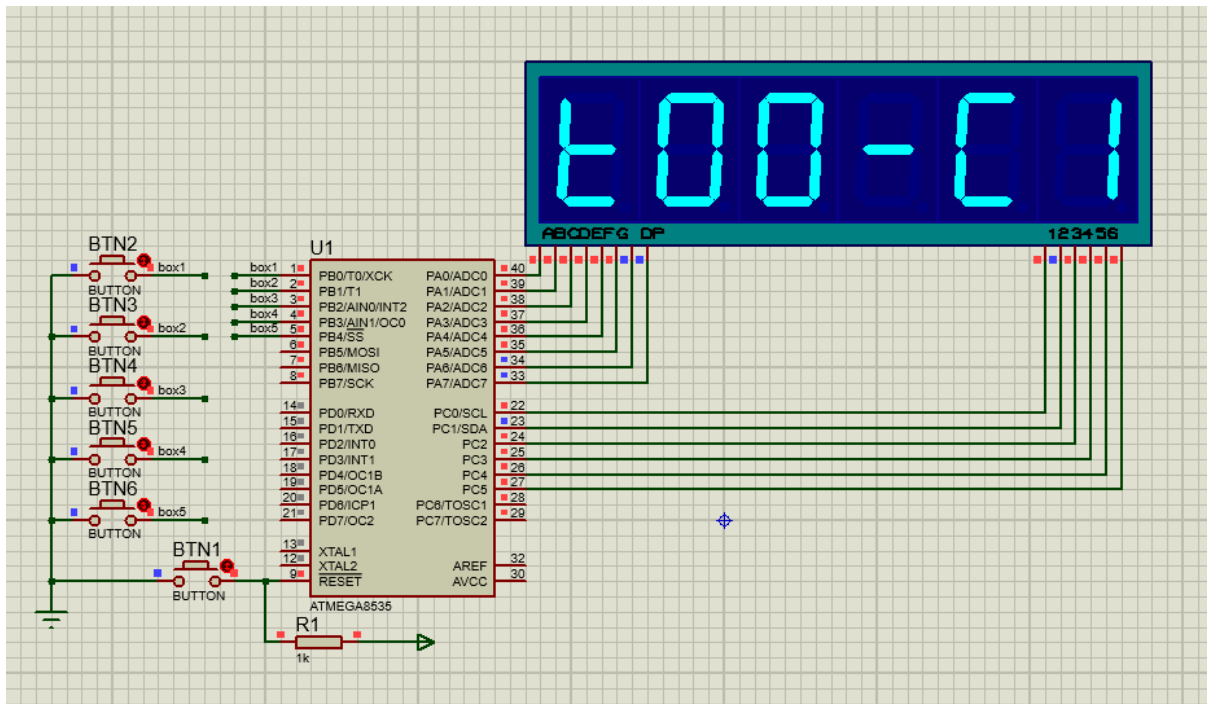


Figura 13. Inicio con el contador de turno en 0

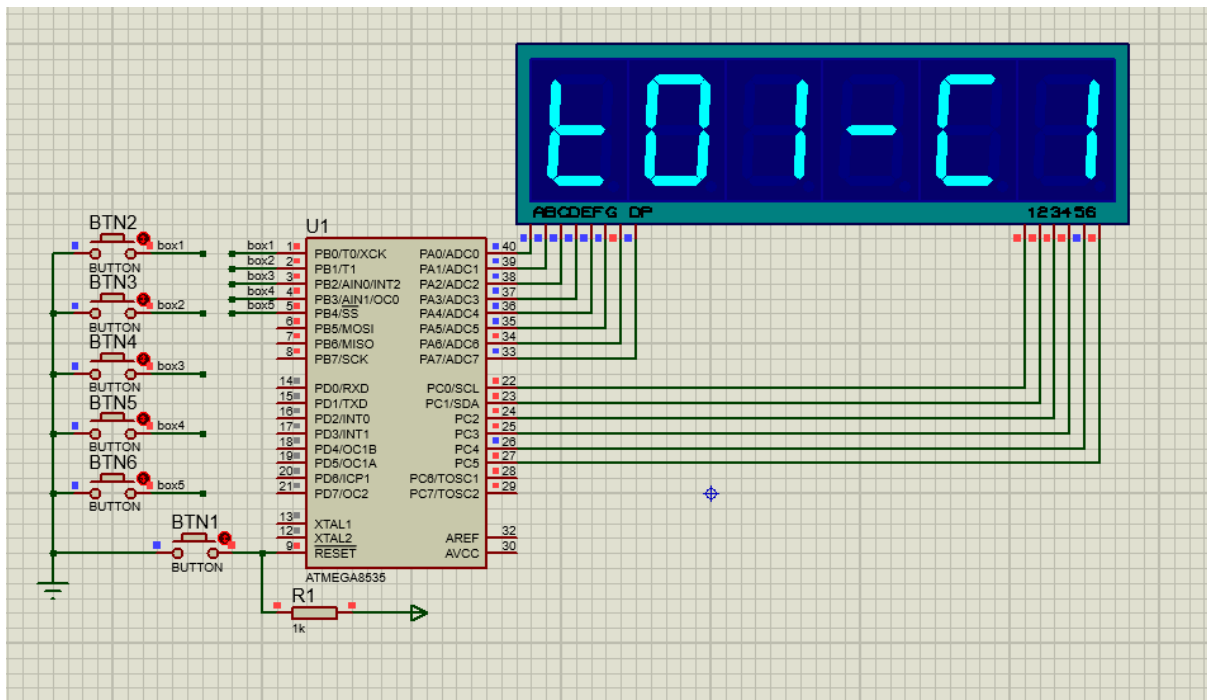


Figura 14. Incremento del contador de turno al pulsar el botón de la fila 1

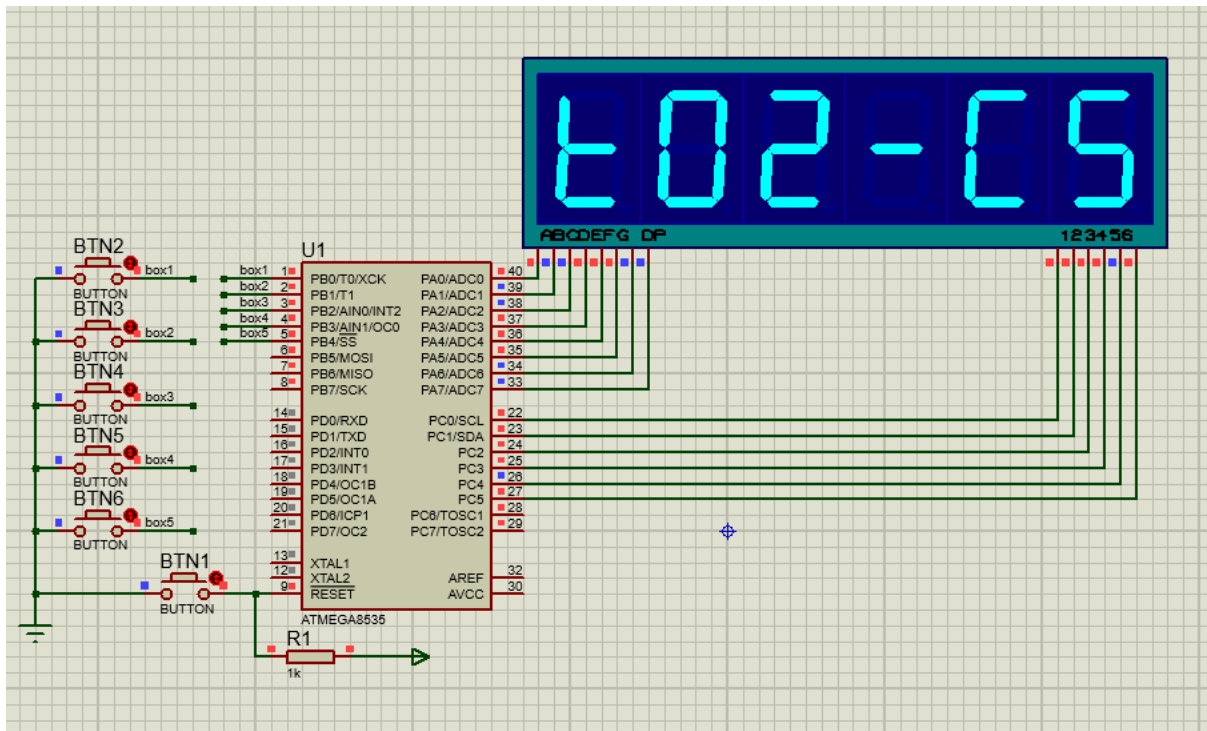


Figura 15. Incremento en el contador y cambio de fila al pulsar el botón 5

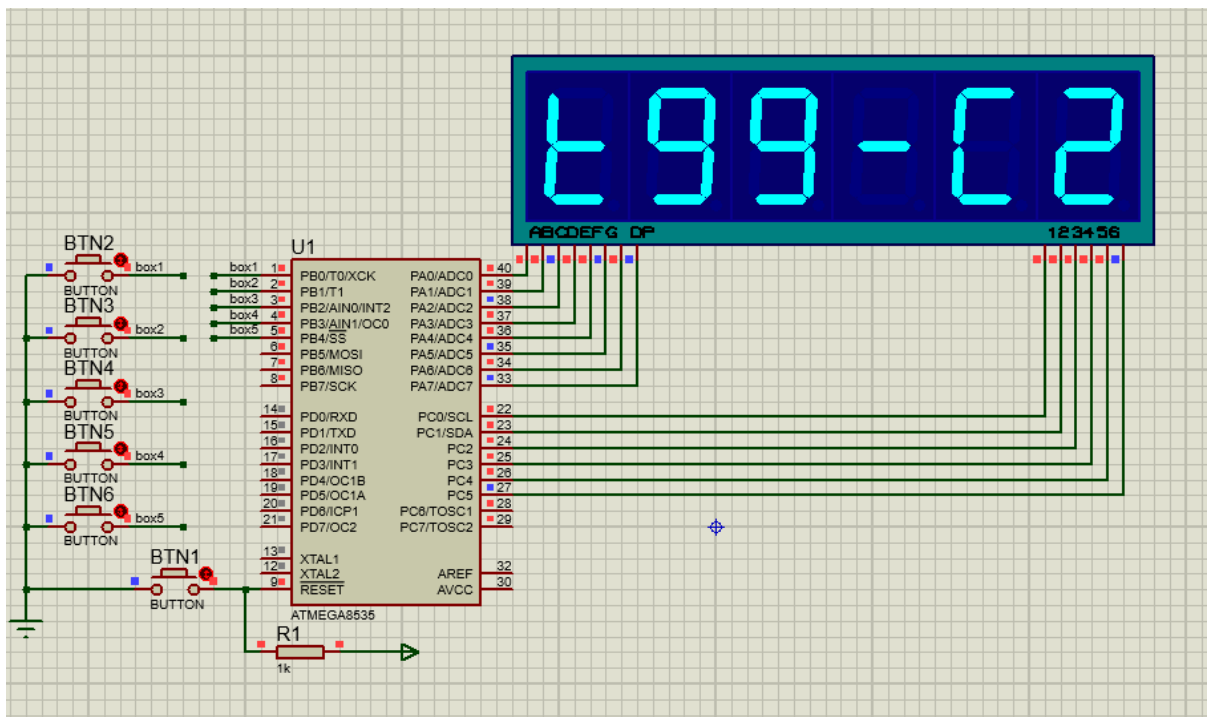


Figura 16. Incremento del contador de turno hasta 99

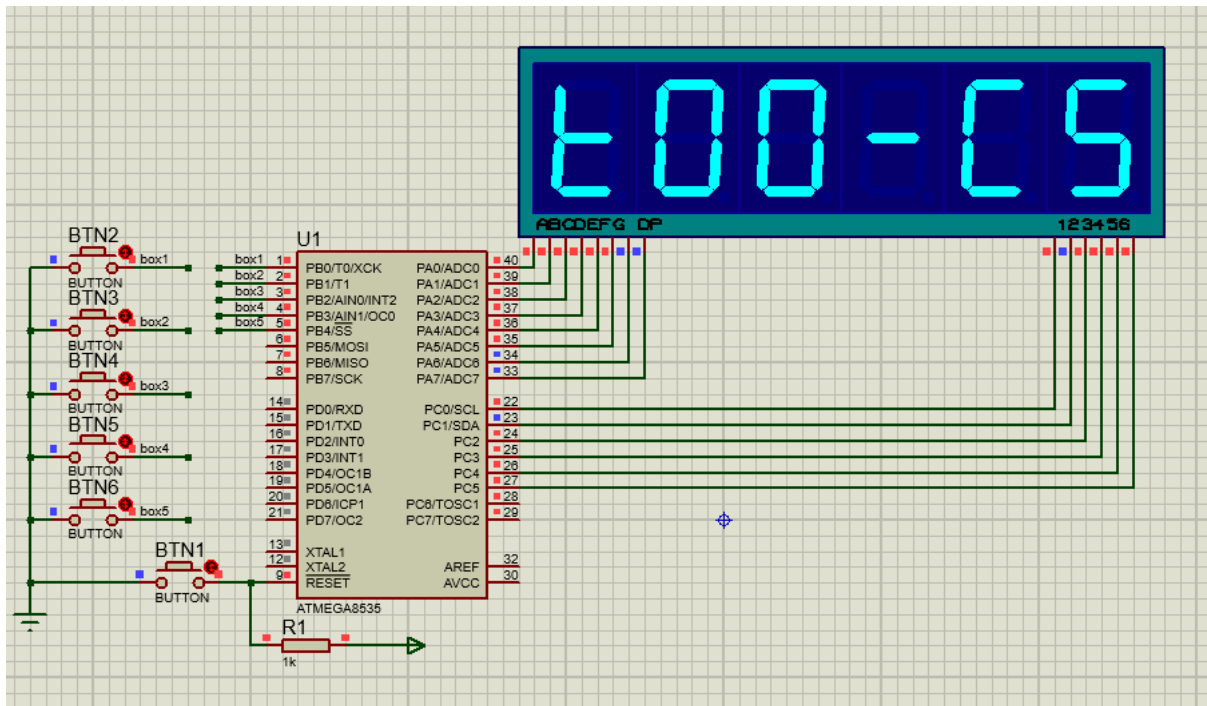
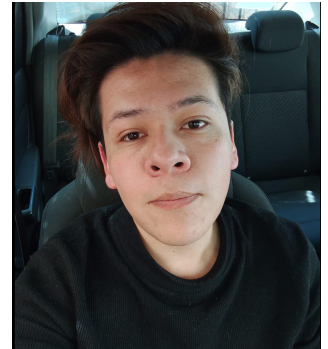


Figura 17. Reseteo del contador después del turno 99 al pulsar el botón de la caja 5

Conclusiones Individuales

Contreras Cardoso Adolfo

El desarrollo de la práctica implicó un gran desafío ya que se puso en marcha bastantes temas que se vieron durante el curso de microcontroladores, en lo personal éste tipo de materias me resultan algo molestas pues no es mi fuerte en toda la carrera pero con ayuda excepcional de un buen equipo se pudo entender parte del código y del cómo fue estructurado para que alcanzará el objetivo que se nos pedía al inicio.



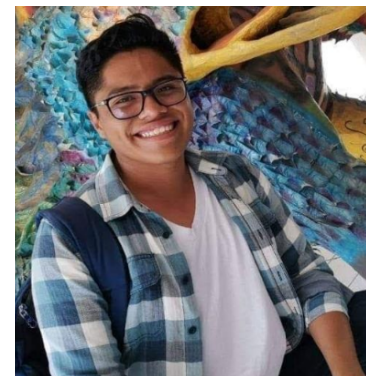
Martínez Alvarado Bryan Alexis

En la práctica de hoy se tomó en cuenta el funcionamiento de varias entradas al microcontrolador, las cuales dependiendo de cuál señal entraba nos arrojaba una salida diferente en nuestro display. Fue muy interesante realizar dicha práctica, ya que, hemos podido observar el funcionamiento de varias instrucciones aprendidas en clase y aplicarlas



Maya Martínez Alonso Rubén

La realización de esta práctica fue complicada ya que se vieron temas avanzados y juntamos muchas prácticas que hemos hecho a lo largo del semestre, la instrucción de mis compañeros me ayudó a comprender el problema mejor y entender cómo se solucionaba. Es interesante como con el paso del curso vamos haciendo programas más complejos.



Pérez Gómez Santiago

Con cada trabajo realizado es evidente el avance en dificultad, así como en los conocimientos adquiridos, lo cual lleva siempre a los resultados esperados, con esto se adquiere experiencia y práctica, lo que permite solucionar errores conocidos de manera sencilla pero al mismo tiempo el surgimiento de nuevos problemas. Personalmente resulta gratificante resolver con éxito el problema planteado de manera inicial.

