



Instituto Politécnico Nacional



Escuela Superior de Cómputo

Supermercado EEPROM

TAREA 11

Materia:

Introducción a los microcontroladores

Grupo:

3CM16

Profesor:

Pérez Pérez José Juan

Integrantes:

Castro Cruces Jorge Eduardo

Cortes Ramírez Roberto Carlos

Domínguez Acosta José Práxedes

Fecha:

Viernes, 26 de noviembre de 2021

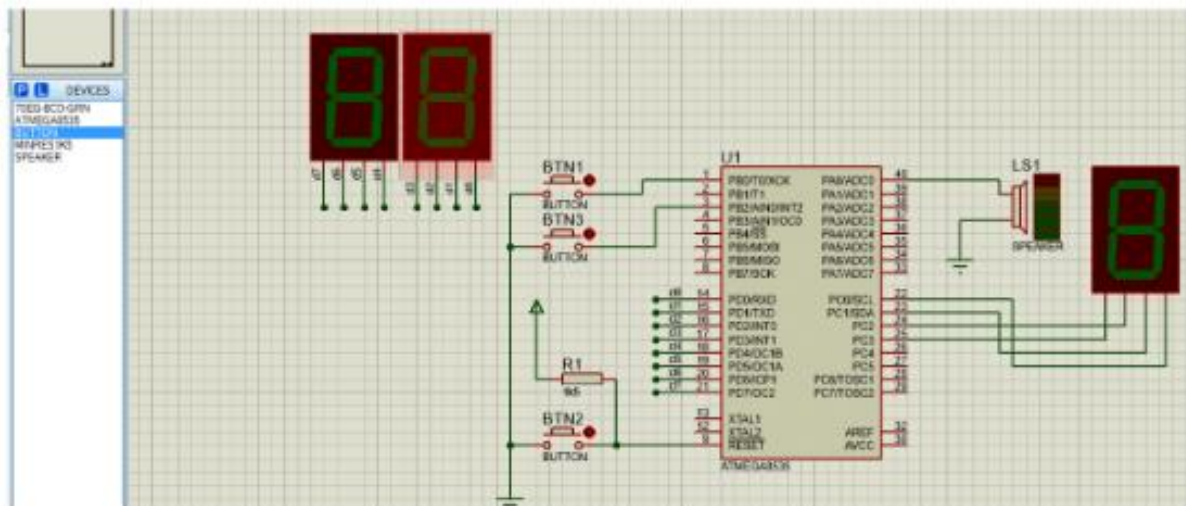
Descripción del problema

Descripción:

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 mostrara "00" y por cada cliente ganador se ira 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".

Subir a más tardar el viernes 26 de nov.:

<https://www.dropbox.com/request/SWIOSwt07kMMKHMSgFAZ>



Código del programa

```
1.      .include "m8535def.inc"
2.      .def aux = r16
3.      .def msk = r17
4.      .def ini = r18
5.      .def tc1h = r19
6.      .def tc1l = r20
7.      .def auc = r21
8.      .def conteo_cli = r21
9.      .def const5 = r22
10.
11.     reset:
12.         rjmp main
13.         .org $004
14.         rjmp onda
15.         .org $008
16.         rjmp tmpo
17.         rjmp cliente
18.     main:
19.         ldi aux, low(ramend)
20.         out spl, aux
21.         ldi aux, high(ramend)
22.         out sph, aux
23.         ser aux
24.         clr conteo_cli
25.         ldi const5, 200
26.         out ddra, aux
27.         out ddrc, aux
28.         out portb, aux
29.         ldi aux, 6
30.         out tccr0, aux
31.         ldi aux, 0b01000101
32.         out tmsk, aux
33.         sei
34.         ldi aux, 250
35.         out tcnt0, aux
36.         ldi msk, 1
37.         ldi ini, 256-141
38.         ldi tc1h, $B3
39.         ldi tc1l, $B5
40.         out tcnt1h, tc1h
41.         out tcnt1l, tc1l
42.     loop:
43.         out portd, conteo_cli
44.         rjmp loop
45.     nada:
46.         in auc, tcnt0
```

```

47.         clr aux
48.         sub aux,auc
49.         out portc,aux
50.         rjmp nada
51. onda:
52.         out tcnt2,ini
53.         in aux,pina
54.         eor aux,msk
55.         out porta,aux
56.         reti
57. tmpo:
58.         ldi aux,0
59.         out tccr2,aux
60.         out tccr1b,aux
61.         ldi aux,250
62.         out tcnt0,aux
63.         add conteo_cli, const5
64.         reti
65. cliente:
66.         ldi aux,2
67.         out tccr2,aux
68.         ldi aux,4
69.         out tccr1b,aux
70.         out tcnt1h,tc1h
71.         out tcnt1l,tc1l
72.         reti
73. ;
74.
75. ; para 5 segundos se necesitan 5,000,000 ciclos
76. ;          19531.25x256=5000000
77. ;          65536-19531=46005=$b3b5
78. ;          tcnt1h <= $B3
79. ;          tcnt1l <= $B5
80.

```

Simulación en AVR Studio 4

AVR Studio - [C:\Users\georg\Desktop\ESCOM\8vo Semestre\Micros\Tareas\11SupermercadoEEPROM\AVR\Supermercado.asm]

File Project Build Edit View Tools Debug Window Help

Trace Disabled

Project

- Supermercado
 - Source Files
 - Supermercado.asm
 - Included Files
 - Labels
 - Output
 - Object File

```
.include "m8535def"
.def aux = r16
.def msk = r17
.def ini = r18
.def tc1h = r19
.def tc1l = r20
.def auc = r21
.def conteo_cli = r21
.def const5 = r22

reset:
    rjmp main
    .org $004
    rjmp onda
    .org $008
    rjmp tmpo
    rjmp cliente

main:
    ldi aux, low(ramend)
    out spl, aux
    ldi aux, high(ramend)
    out sph, aux
    ser aux
    clr conteo_cli
    ldi const5, 200
    out ddra, aux
    out ddrc, aux
    out portb, aux
    ldi aux, 6
    out tccr0, aux
```

Assemble and run (Ctrl+F7)

I/O View

ANALOG_COMPARATOR

Name	Value
AD_CONVERTER	
ANALOG_COMPARATOR	
CPU	
EEPROM	
EXTERNAL_INTERRUPT	
PORTA	
PORTB	
PORTC	
PORTD	
SPI	
TIMER_COUNTER_0	
TIMER_COUNTER_1	
TIMER_COUNTER_2	
TWI	
USART	

Name	Address	Value	Bits
------	---------	-------	------

Build

[.cseg]	0x000000	0x000076	106	0	106	8192	1.3%
[.dseg]	0x000060	0x000060	0	0	0	512	0.0%
[.eseg]	0x000000	0x000000	0	0	0	512	0.0%

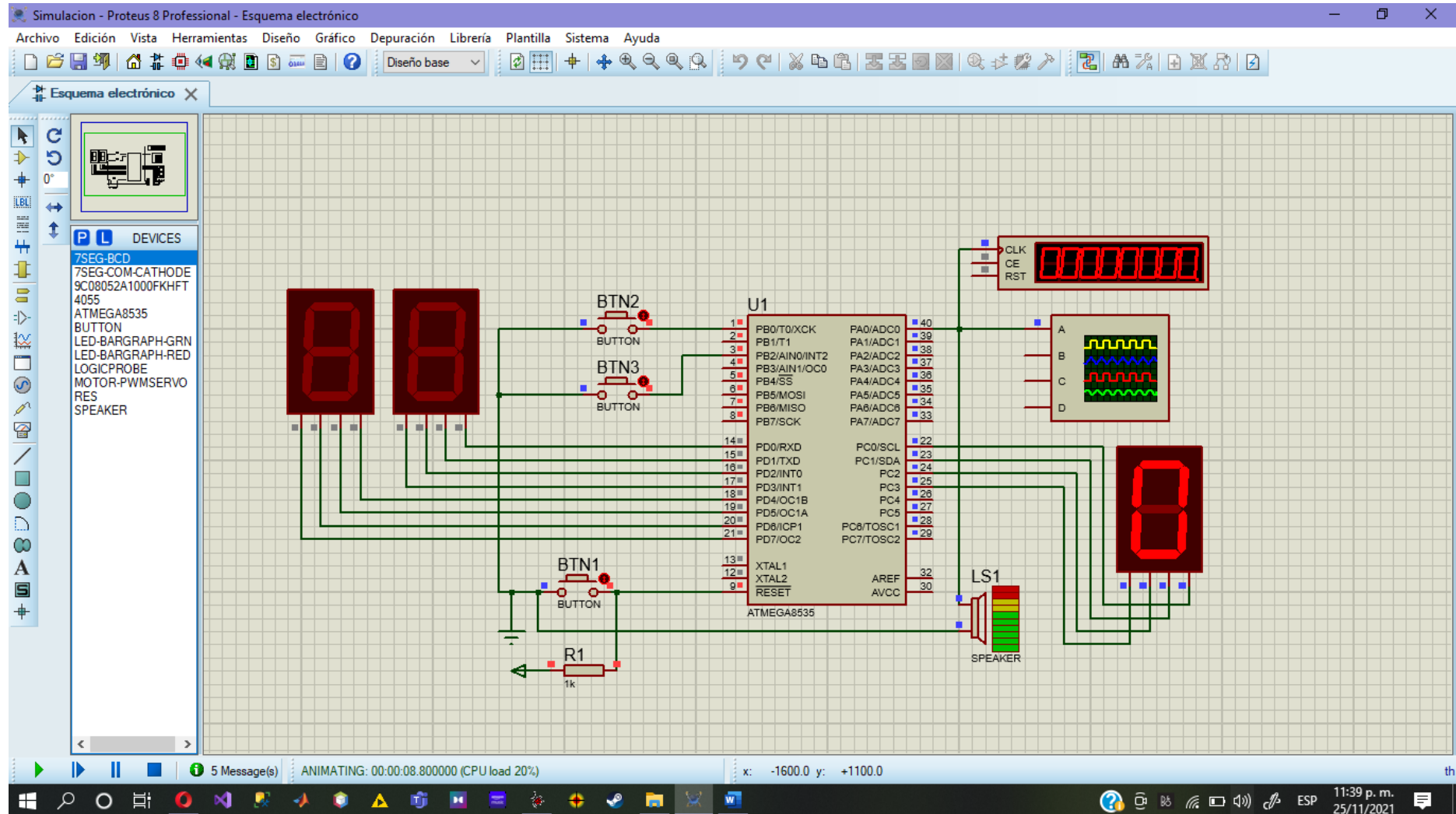
Assembly complete, 0 errors. 1 warnings

Build Message Find in Files Breakpoints and Tracepoints

Loading event memory... ATmega8535 AVR Simulator Auto Ln 1, Col 1 CAP NUM OVR

11:40 p. m. 25/11/2021

Simulación en Proteus 8 Professional



Conclusiones

- **Castro Cruces Jorge Eduardo**

Para esta práctica tuvimos que realizar los cálculos para poder encontrar el valor inicial del contador 2 para aproximar el tiempo de desborde al deseado y el del contador 1 para los 5s, además de que vimos cómo hacer el conteo usando un timer con entrada externa activada por flanco de bajada, para resolver este ejercicio nos basamos mucho en las prácticas anteriores para realizar este ejercicio.

- **Cortes Ramírez Roberto Carlos**

En esta práctica hicimos un código el cual podemos hacer que en un display bcd se imprima una cuenta descendente del 5 al 0. Para esto tuvimos que generar un tono a 440 hz y un temporizador de 5 segundos para que pueda funcionar sin ningún problema. Para poder generar este código debimos tener varias variables para así poder asociarlo con el simulador y que pueda correr sin problema aparte de que tuvimos que usar 1MHz para que no le pase nada al microcontrolador.

Para poder ver cómo funciona usamos proteus en este fue un circuito un poco más grande, con un botón para reiniciar la secuencia, una bocina, y un display BCD, y con el atmega8535 para poder usar el programa.

- **Domínguez Acosta José Práxedes**

En esta práctica realizamos los cálculos para poder encontrar el valor inicial del contador 2, vimos cómo hacer el conteo usando un timer con entrada externa activada por flanco de bajada para resolver este ejercicio, el ejercicio me pareció bastante bueno