



Instituto Politécnico Nacional



Escuela Superior de Cómputo

ADC

TAREA 12

Materia:

Introducción a los microcontroladores

Grupo:

3CM16

Profesor:

Pérez Pérez José Juan

Integrantes:

Castro Cruces Jorge Eduardo

Cortes Ramírez Roberto Carlos

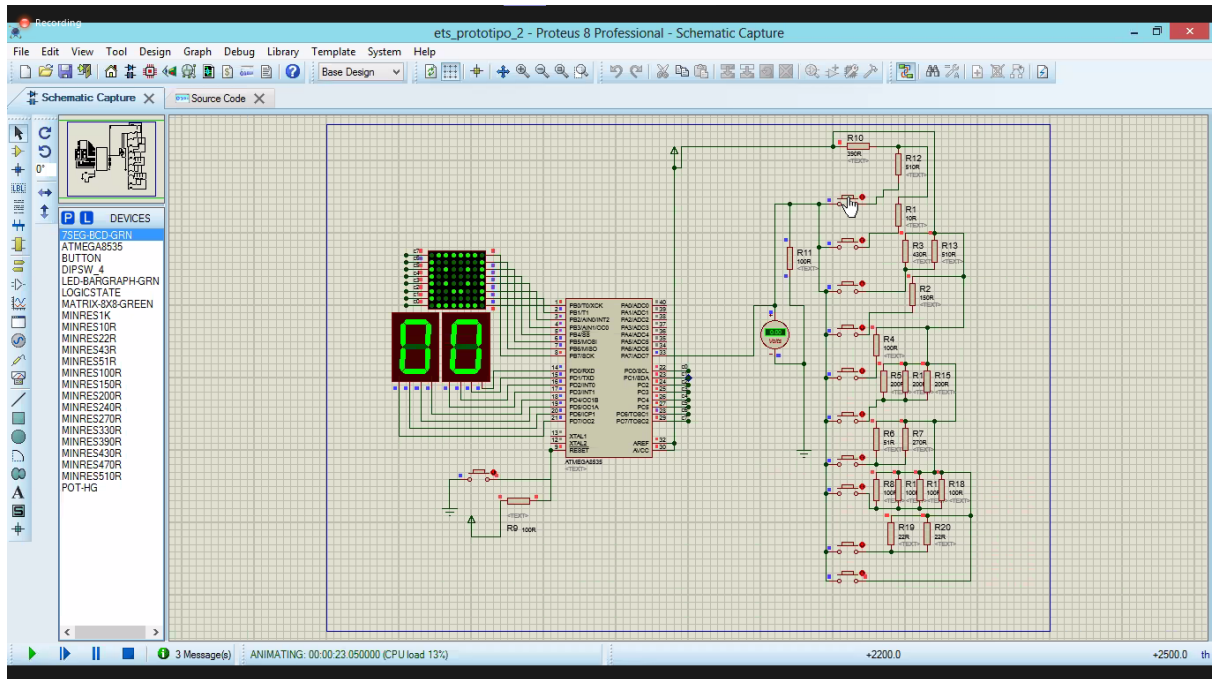
Domínguez Acosta José Práxedes

Fecha:

lunes, 6 de diciembre de 2021

Descripción del problema

Descripción:



Código del programa

```
1.      .include "m8535def.inc"
2.      .def adl = r17
3.      .def adh = r16
4.      .def col = r18
5.      .def aux = r19
6.      .def cta = r20
7.      ;-----
      -----
8.      .macro num
9.          push zh
10.         push zl
11.         ldi ZH, high(@0<<1)           ; Initialize Z
      pointer
12.         ldi ZL, low(@0<<1)
13.         lpm r0, Z+
14.         lpm r1, Z+
15.         lpm r2, Z+
16.         lpm r3, Z+
17.         lpm r4, Z+
18.         lpm r5, Z+
19.         lpm r6, Z+
20.         lpm r7, Z
21.         pop zl
22.         pop zh
23.     .endm
24.     ;-----
      -----
25.         rjmp Start
26.         .org $008
27.         rjmp cuenta
28.         rjmp barre
29.         .ORG $0E
30.         RJMP CONV
31.     ;-----
      -----
32.     Start:
33.         LDI R16, LOW(RAMEND)
34.         OUT SPL, R16
35.         LDI R16, HIGH(RAMEND)
36.         OUT SPH, R16
37.         SER R16
38.         OUT DDRD, R16
39.         OUT DDRB, R16
40.         OUT DDRC, R16
41.         LDI R16, $ED
42.         OUT ADCSRA, R16
```

```

43.         ldi r16, $27
44.         out ADMUX, r16
45.         ldi aux, 2
46.         out tccr0, aux
47.         ldi aux, 2
48.         out tccr1b, aux
49.         ldi aux, 5
50.         out timsk, aux
51.         SEI
52.         ldi cta, -1
53.         ldi col, 1
54.         clr zh
55.         ldi zl, 0
56.         ;-----
57.         Loop:
58.             OUT PORTD, adh
59.             rjmp Loop
60.         ;-----
61.         CONV:
62.             IN adl, ADCL
63.             IN adh, ADCH
64.             RETI
65.         ;-----
66.         barre:
67.             out portb, zh
68.             ld aux, z+
69.             lsl col
70.             brcs nbarre
71.         sss:
72.             com col
73.             out portc, col
74.             com col
75.             out portb, aux
76.             reti
77.         nbarre:
78.             ldi col, 1
79.             ldi zl, 0
80.             ld aux, z+
81.             rjmp sss
82.         ;-----
83.         cuenta:
84.             ;inc cta
85.             mov cta, adh
86.             cpi cta, $19
87.             breq cta0
88.             cpi cta, $33
89.             breq cta1
90.             cpi cta, $4C

```

```

91.          breq cta2
92.          cpi cta, $66
93.          breq cta3
94.          cpi cta, $80
95.          breq cta4
96.          cpi cta, $99
97.          breq cta5
98.          cpi cta, $B3
99.          breq cta6
100.         cpi cta, $CC
101.         breq cta7
102.         cpi cta, $E6
103.         breq cta8
104.         cpi cta, $FF
105.         breq cta9
106.         cpi cta, $00
107.         breq ctax
108.         ;brne ncta
109.         ;ldi cta, -1
110.         ncta:
111.         reti
112.         cta3:
113.         rjmp cta31
114.         cta4:
115.         rjmp cta41
116.         cta5:
117.         rjmp cta51
118.         cta6:
119.         rjmp cta61
120.         cta7:
121.         rjmp cta71
122.         cta8:
123.         rjmp cta81
124.         cta9:
125.         rjmp cta91
126.         ctax:
127.         rjmp ctax1
128.         cta0:
129.         num cero
130.         rjmp ncta
131.         cta1:
132.         num uno
133.         rjmp ncta
134.         cta2:
135.         num dos
136.         rjmp ncta
137.         cta31:
138.         num tres

```

```

139.         rjmp ncta
140. cta41:
141.         num cuatro
142.         rjmp ncta
143. cta51:
144.         num cinco
145.         rjmp ncta
146. cta61:
147.         num seis
148.         rjmp ncta
149. cta71:
150.         num siete
151.         rjmp ncta
152. cta81:
153.         num ocho
154.         rjmp ncta
155. cta91:
156.         num nueve
157.         rjmp ncta
158. ctax1:
159.         num equis
160.         rjmp ncta
161. ;-----
162. cero:
163.         .db $00,$7C,$82,$82,$82,$7C,$00,$00
164. uno:
165.         .db $00,$22,$42,$FE,$02,$02,$00,$00
166. dos:
167.         .db $00,$42,$86,$8A,$92,$62,$00,$00
168. tres:
169.         .db $00,$44,$82,$92,$92,$6C,$00,$00
170. cuatro:
171.         .db $00,$08,$18,$28,$48,$FE,$00,$00
172. cinco:
173.         .db $00,$F4,$92,$92,$92,$8C,$00,$00
174. seis:
175.         .db $00,$7C,$92,$92,$92,$4C,$00,$00
176. siete:
177.         .db $00,$80,$80,$80,$80,$FE,$00,$00
178. ocho:
179.         .db $00,$6C,$92,$92,$92,$6C,$00,$00
180. nueve:
181.         .db $00,$64,$92,$92,$92,$7C,$00,$00
182. equis:
183.         .db $7E,$81,$A9,$85,$85,$A9,$81,$7E

```

Simulación en AVR Studio 4

AVR Studio - [C:\Users\georg\Desktop\ESCOM\8vo Semestre\Micros\Tareas\12ADC\AVR\ADC.asm]

File Project Build Edit View Tools Debug Window Help

Trace Disabled

Project

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

Assembly and run (Ctrl+F7)

```
.include "m8535def"
.def adl = r17
.def adh = r16
.def col = r18
.def aux = r19
.def cta = r20

;-----
.macro num
    push zh
    push zl
    ldi ZH, high(@0<<1)      ; Initialize Z pointer
    ldi ZL, low(@0<<1)
    lpm r0, Z+
    lpm r1, Z+
    lpm r2, Z+
    lpm r3, Z+
    lpm r4, Z+
    lpm r5, Z+
    lpm r6, Z+
    lpm r7, Z
    pop zl
    pop zh
.endm

;-----
rjmp Start
.org $008
rjmp cuenta
rjmp barre
.org $0E
RJMP CONV
```

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 0x000252 484 88 572 8192 7.0%
[.dseg] 0x000060 0x000060 0 0 0 512 0.0%
[.eseg] 0x000000 0x000000 0 0 0 512 0.0%
```

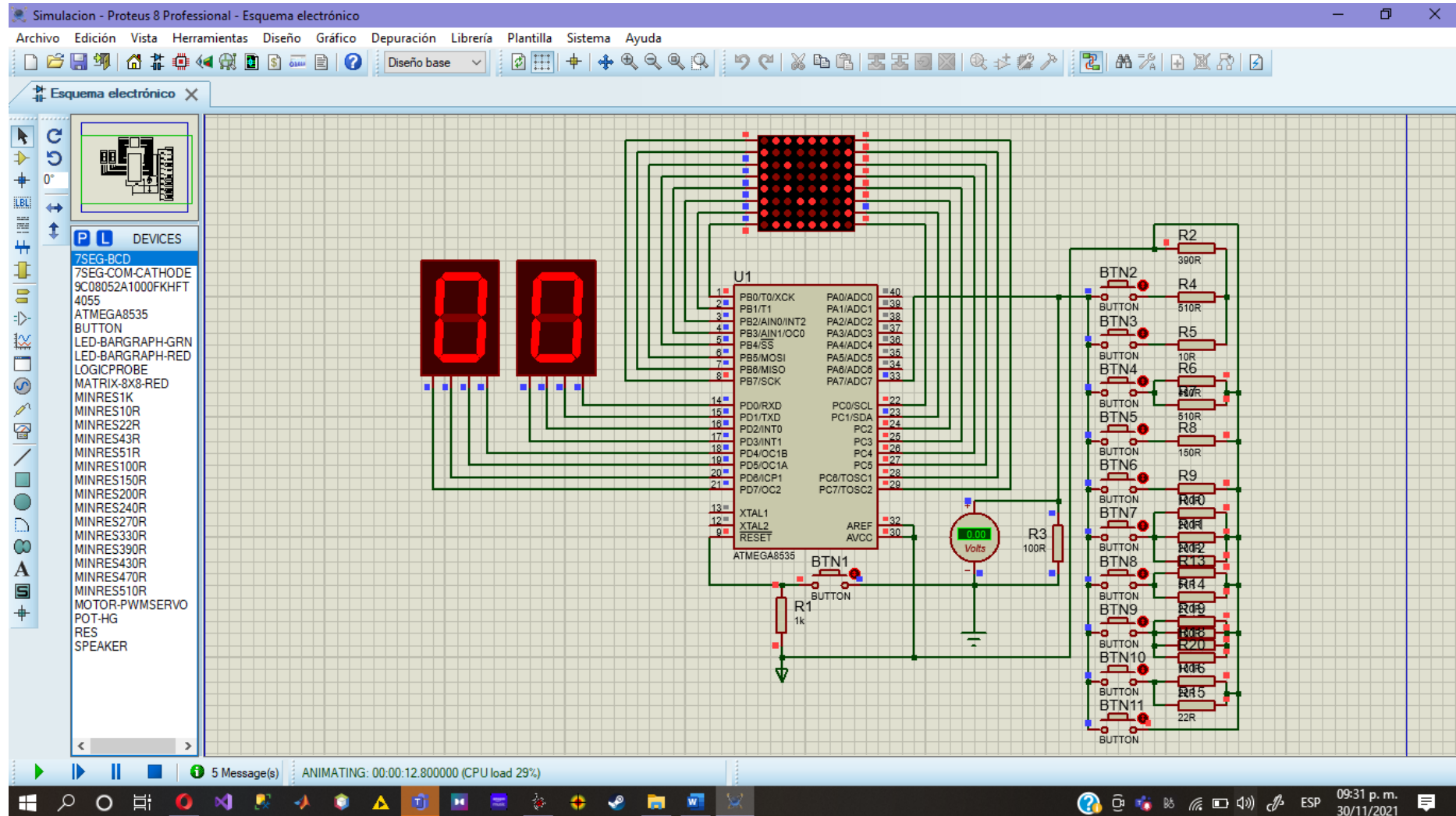
Assembly complete, 0 errors. 0 warnings

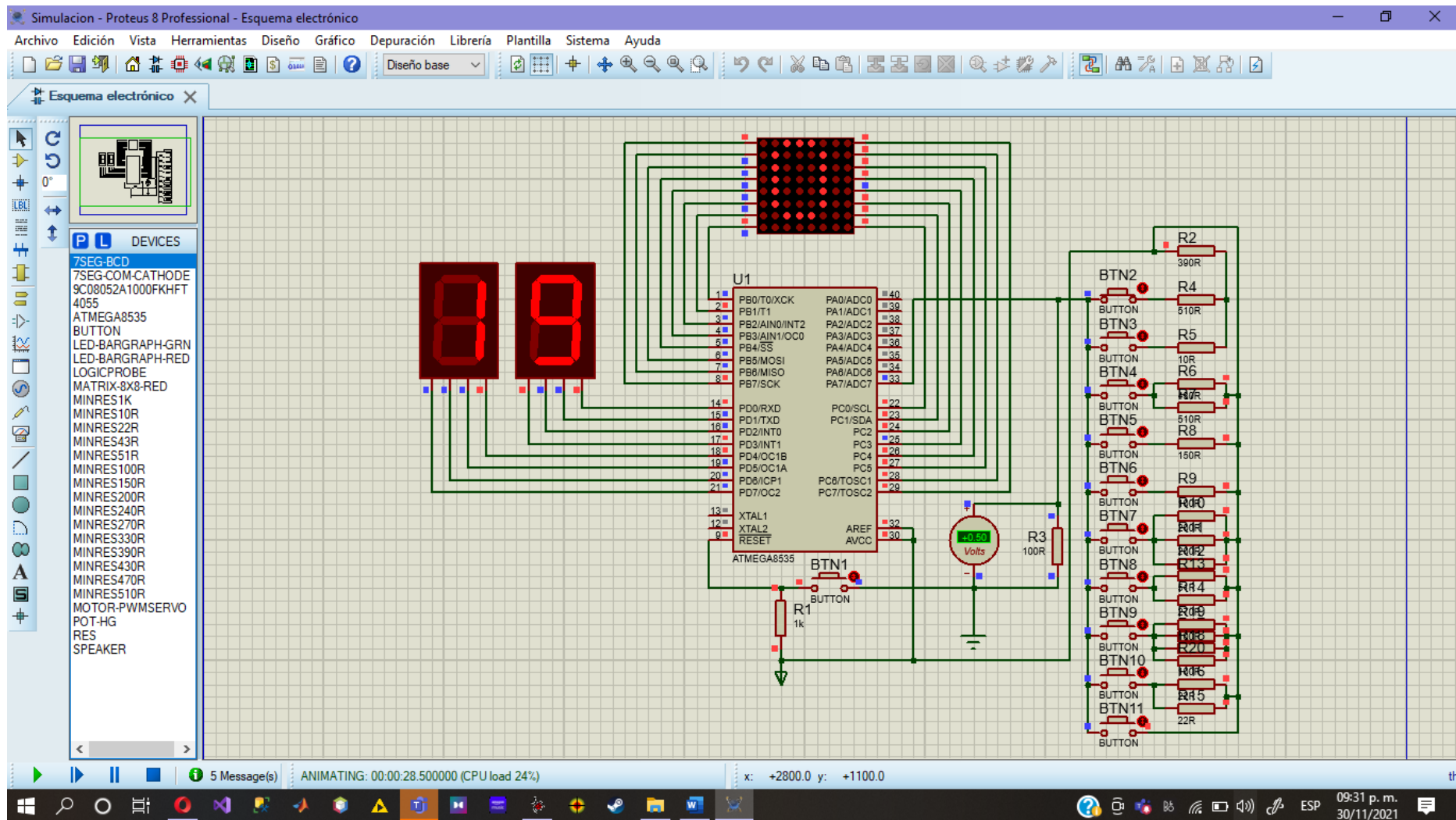
Build Message Find in Files Breakpoints and Tracepoints

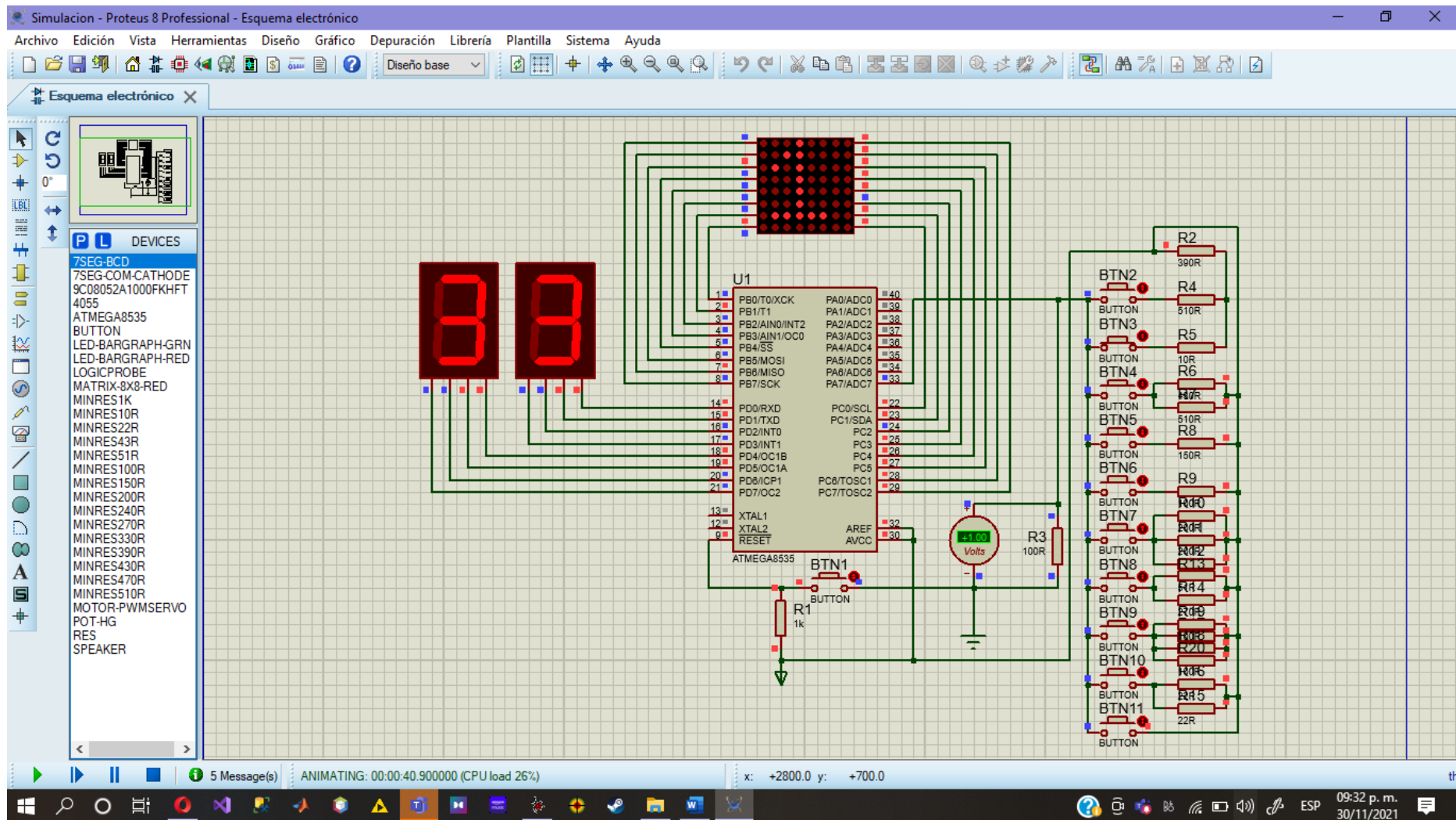
Loading event memory... ATmega8535 AVR Simulator Auto Ln 1, Col 1 CAP NUM OVR

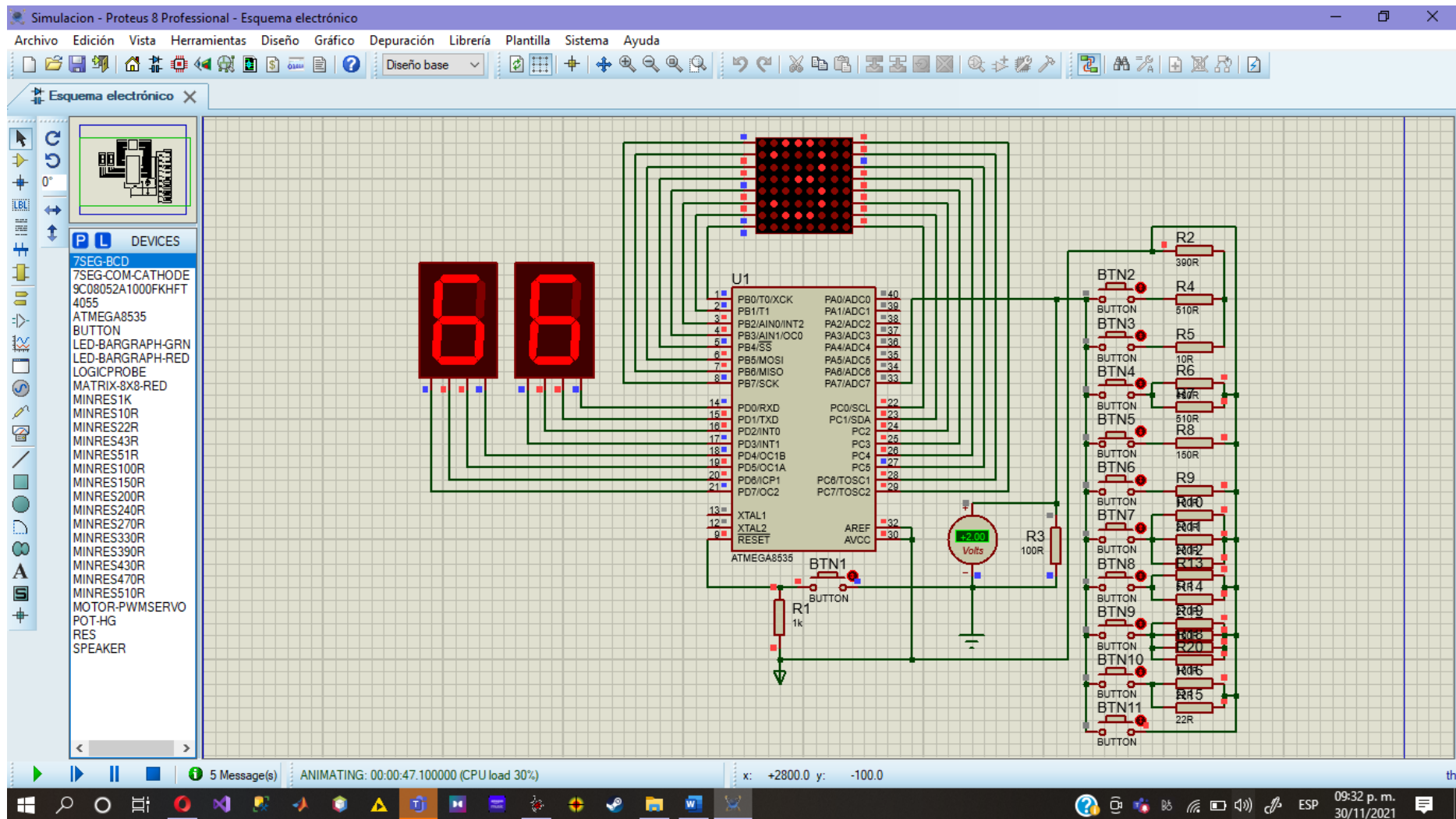
09:30 p. m. 30/11/2021

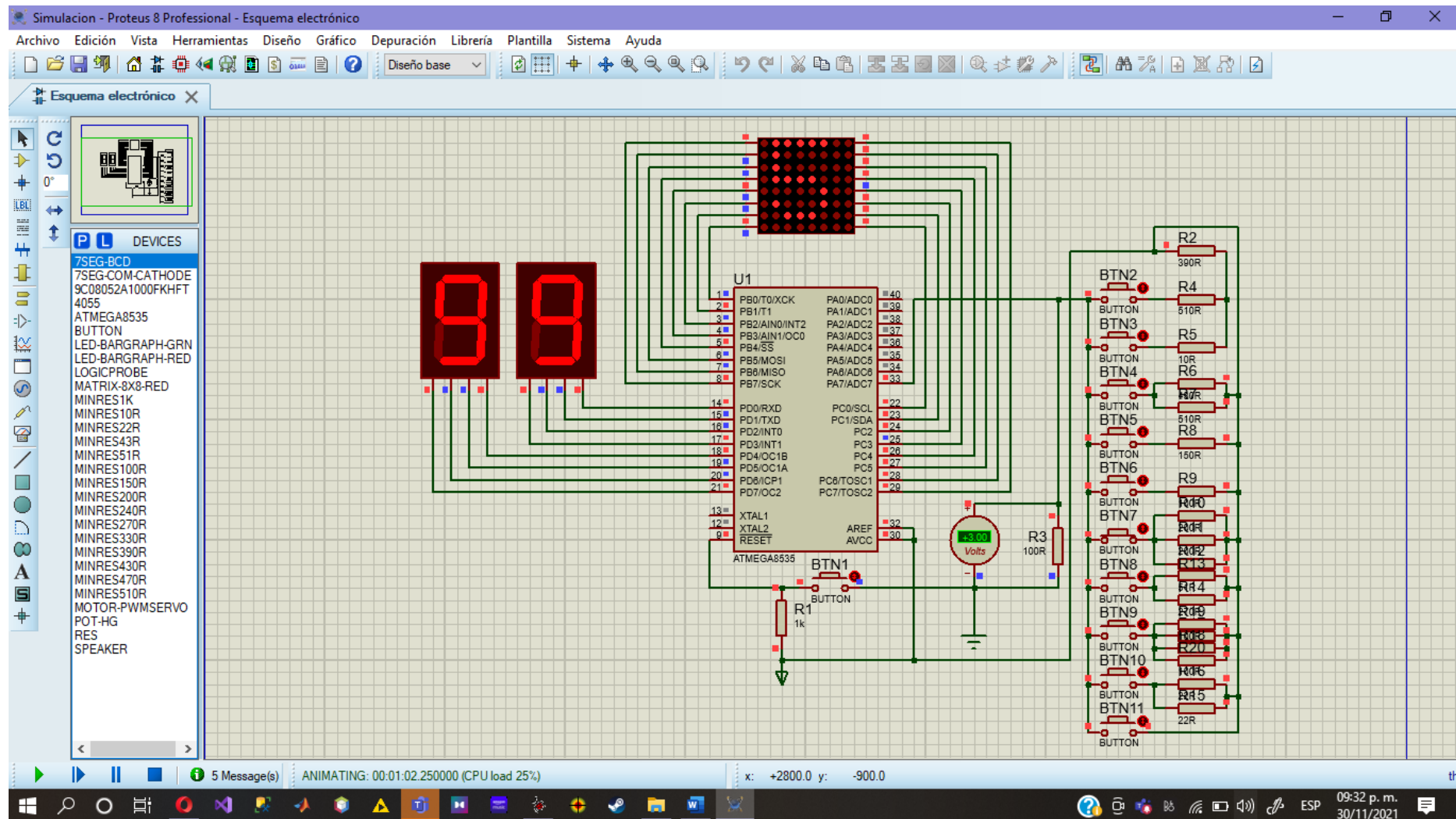
Simulación en Proteus 8 Professional

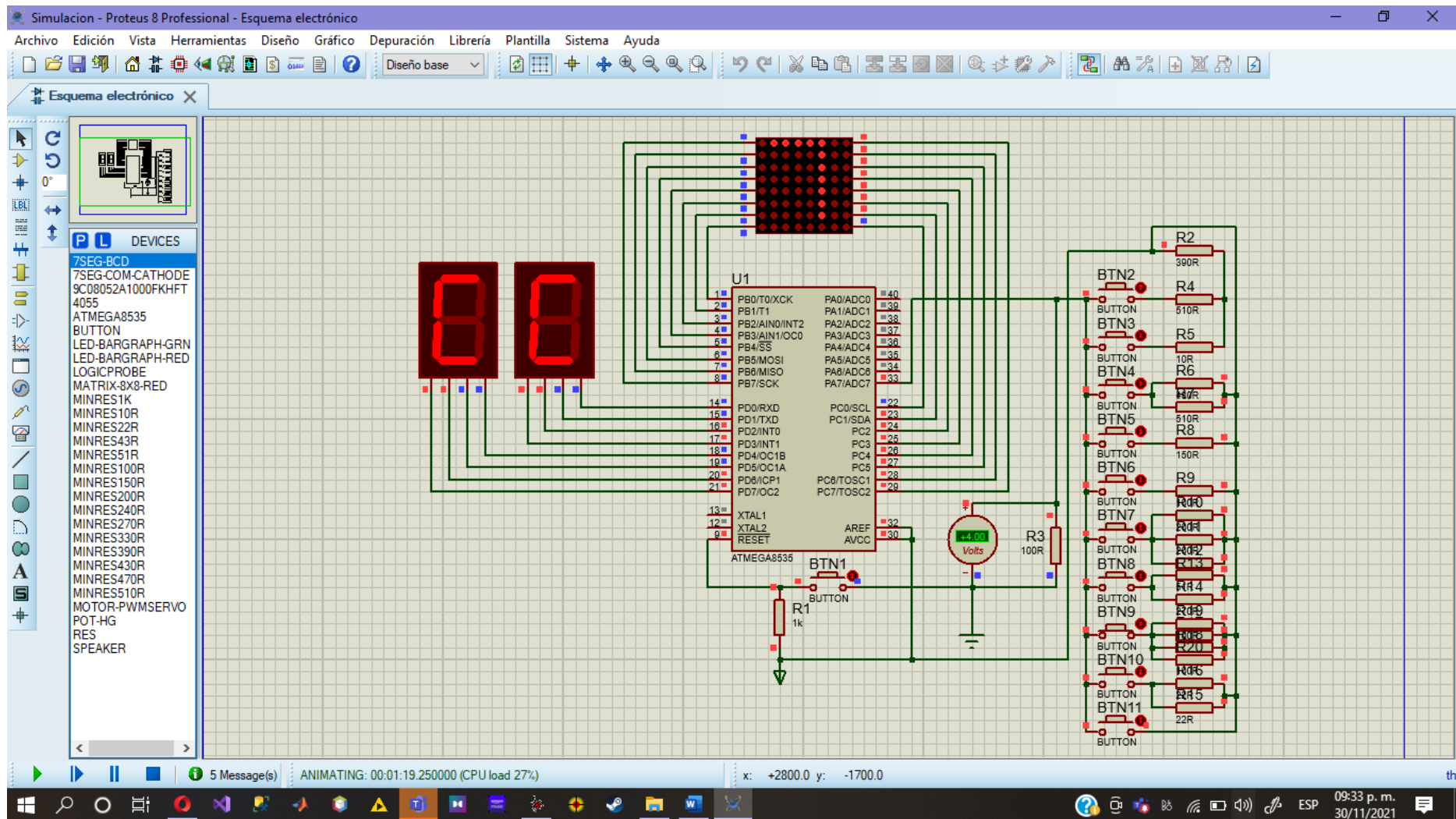


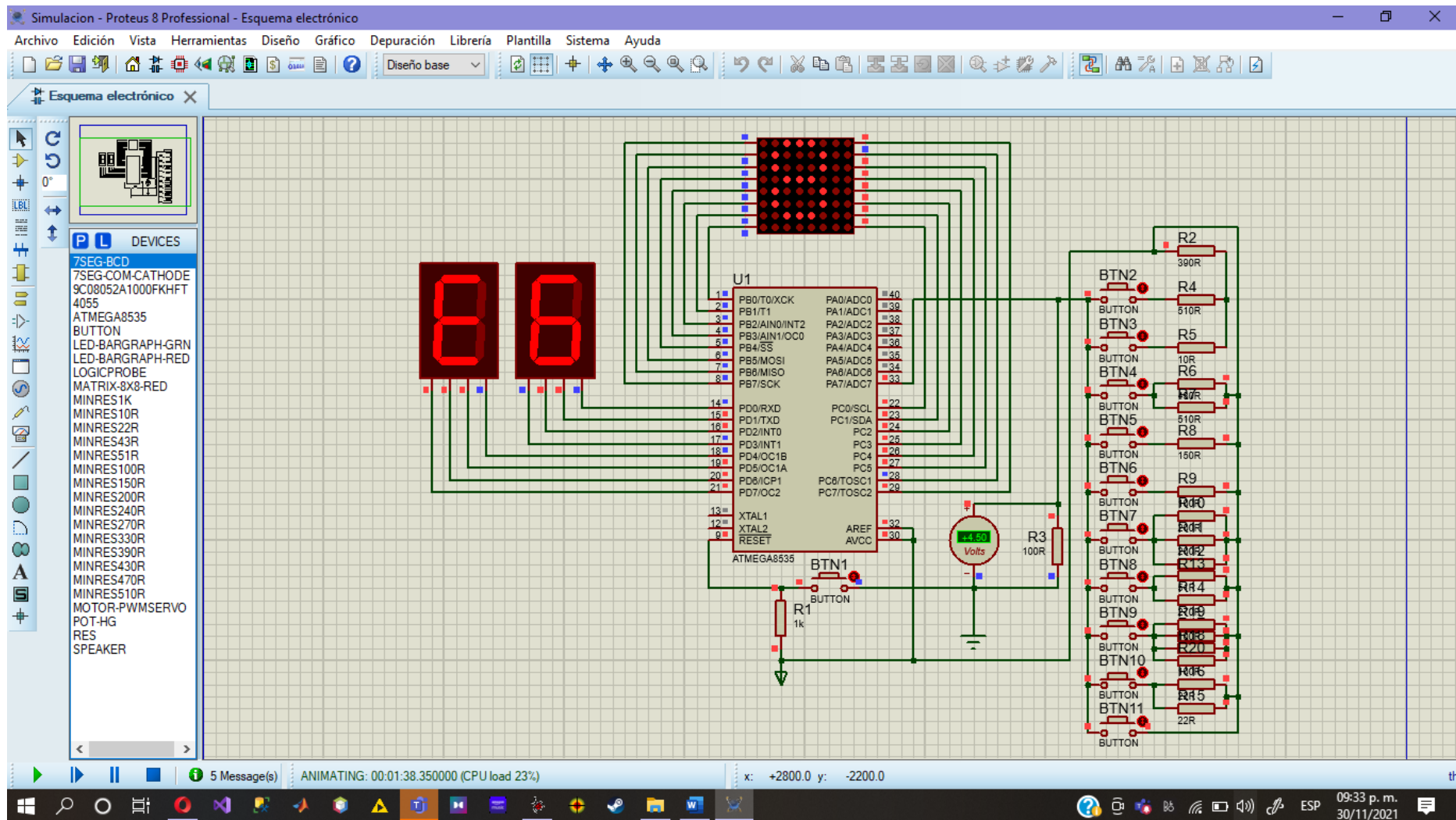


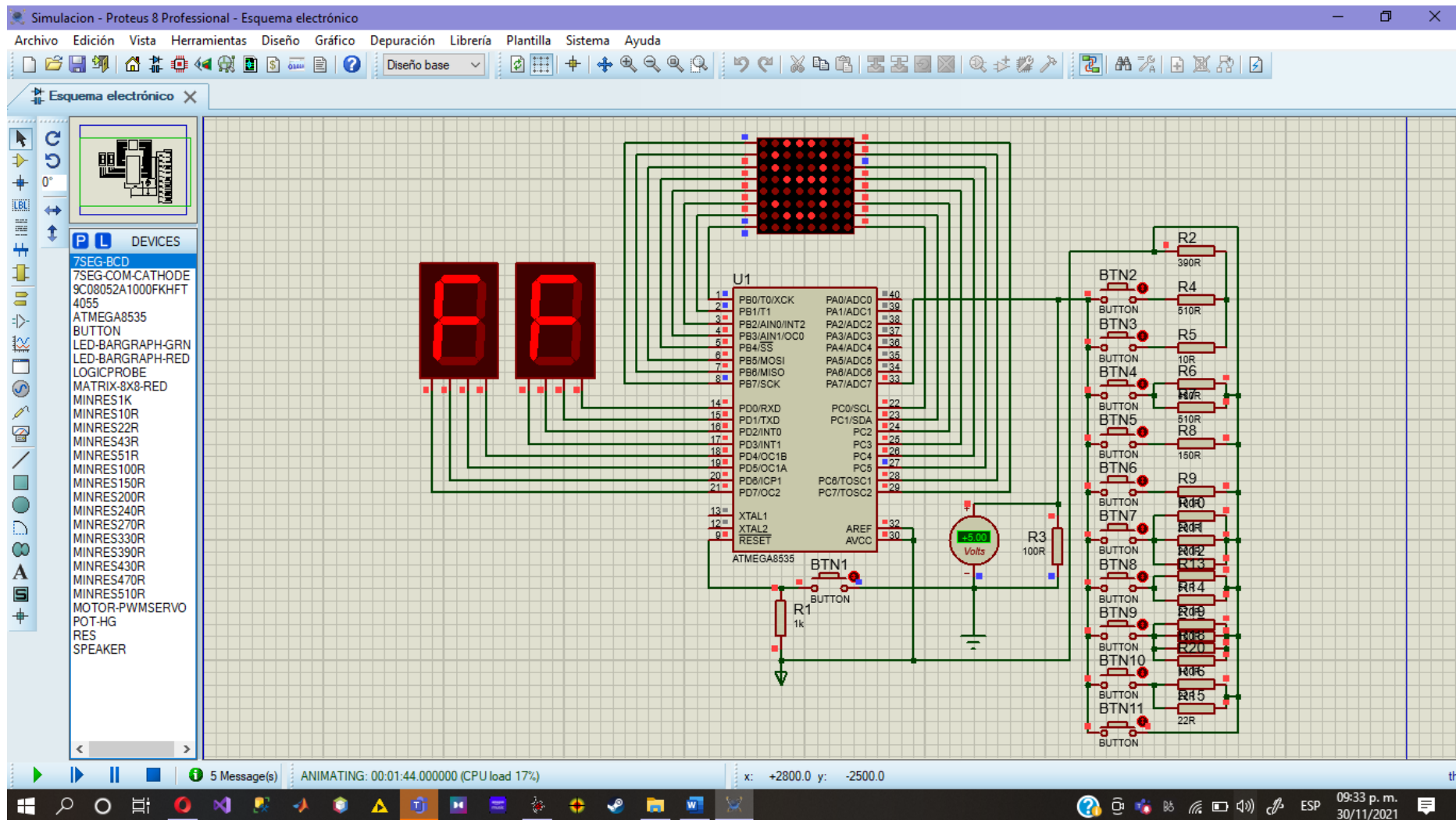












Conclusiones

- **Castro Cruces Jorge Eduardo**

Esta practica fue bastante sencilla, porque prácticamente ya estaba hecho gran parte del código, solo fue cosa de armar al circuito y probarlo para verificar que funcionase bien.

- **Cortes Ramírez Roberto Carlos**

En esta práctica hicimos un circuito y un Código el cual podemos hacer que en un display de 64 segmentos se pueda imprimir diferentes números o caracteres y al lado de este mismo hay 2 display de 8 segmentos donde se imprime la combinación que se está ejecutando en el código, también tenemos unos botones que cuenta como entradas de datos, que con este vamos a definir el valor de dato que queremos. Para poder generar este código debimos tener varias variables para así poder asociarlo con el simulador y que pueda correr sin problema aparte una secuencia de datos apra que pueda imprimirlos en los display.

Para poder ver cómo funciona usamos proteus en este fue un circuito un grande, con varios botones para reiniciar la secuencia, 2 display de 8 segmentos, y un display de 64 segmentos, y con el atmega8535 para poder usar el programa.

- **Domínguez Acosta José Práxedes**

En esta práctica Se generó un código en el que debimos tener varias variables para así poder asociarlo con el simulador y que pueda correr sin problema aparte una secuencia de datos para que pueda imprimirlos en los display. Además, se realizó un circuito que usa un display de 64 segmentos y lograr imprimir diferentes números o caracteres, también hay 2 display de 8 segmentos en los cuales se muestra la combinación que se realiza en el código, Hacemos uso de botones para que cuenten como entradas de datos y definir el valor de dato que queremos. Se generó un código en el que debimos tener varias variables para así poder asociarlo con el simulador y que pueda correr sin problema aparte una secuencia de datos apra que pueda imprimirlos en los display.