



# Instituto Politécnico Nacional



## Escuela Superior de Cómputo

Interrupciones externas

TAREA 7

Materia:

Introducción a los microcontroladores

Grupo:

3CM16

Profesor:

Pérez Pérez José Juan

Integrantes:

Castro Cruces Jorge Eduardo

Cortes Ramírez Roberto Carlos

Domínguez Acosta José Praxedes

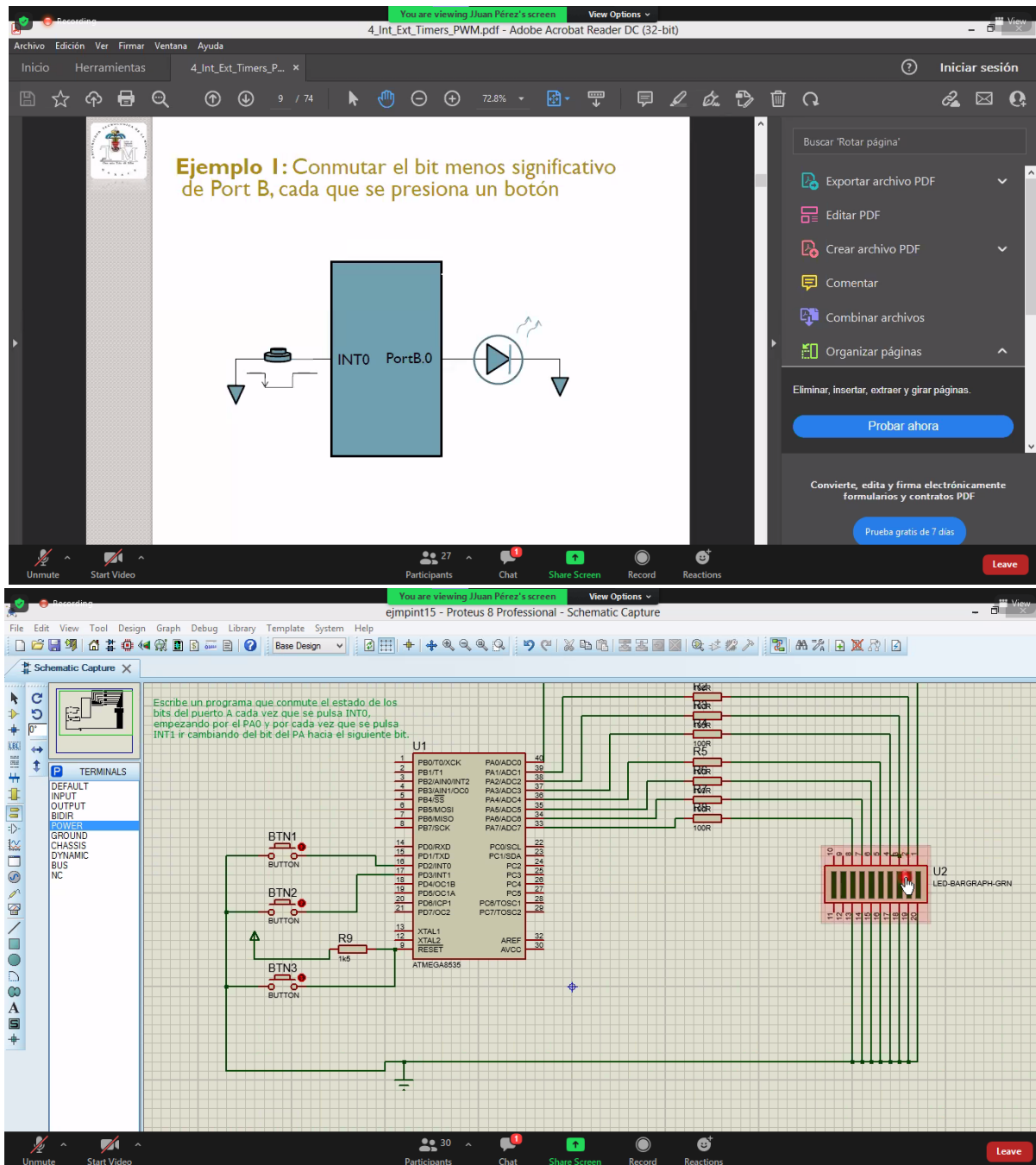
Fecha:

martes, 26 de octubre de 2021

## Descripción del problema

Descripción:

Escribe un programa utilizando las interrupciones externas INT0 e INT1, que realice lo siguiente: Se tendrán conectados interruptores momentáneos (pushboton) en INT0 e INT1, y 8 leds conectados en el puerto A, al inicio por cada vez que se pulsa el botón conectado a INT0 se deberá conmutar el estado del PA0, y por cada vez que se pulsa el botón conectado en INT1 se deberá cambiar al siguiente led en el puerto A a conmutar.



## Código del programa

```
1.      .include "m8535def.inc"
2.      .def aux = r16
3.      .def nbit = r17
4.      rjmp main
5.      rjmp cnta
6.      rjmp cmba
7.
8. main:
9.          ldi aux, low(ramend)
10.         out spl, aux
11.         ldi aux, high(ramend)
12.         out sph, aux
13.         ser aux
14.         out ddra, aux
15.         out portd, aux
16.         ldi aux, $0a
17.         out mcucr, aux
18.         ldi aux, $c0
19.         out gicr, aux
20.         clr aux
21.         ldi nbit, 1
22.         sei
23.
24.     fin:
25.         out porta, aux
26.         rjmp fin
27.
28.     cnta:
29.         in aux, pina
30.         eor aux, nbit
31.         reti
32.
33.     cmba:
34.         lsl nbit
35.         reti
```

## Simulación en AVR Studio 4

AVR Studio - [C:\Users\georg\Desktop\ESCOM\8vo Semestre\Micros\Tareas\7INT\AVRstudio\INT.asm]

File Project Build Edit View Tools Debug Window Help

Trace Disabled

**Processor**

Name	Value
Program Counter	0x000000
Stack Pointer	0x0000
X pointer	0x0000
Y pointer	0x0000
Z pointer	0x0000
Cycle Counter	0
Frequency	4.0000 MHz
Stop Watch	0.00 us
SREG	00000000
Registers	

```
.include "m8535def.inc"
.def aux = r16
.def nbit = r17
rjmp main
rjmp cnta
rjmp cmba

main:
ldi aux, low(ramend)
out spl, aux
ldi aux, high(ramend)
out sph, aux
ser aux
out ddra, aux
out portd, aux
ldi aux, $0a
out mcucr, aux
ldi aux, $c0
out gicr, aux
clr aux
ldi nbit, 1
sei

fin:
out porta, aux
rjmp fin

cnta:
```

aux=0 "Register R16"

**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**

ATmega8535 memory use summary [bytes]:

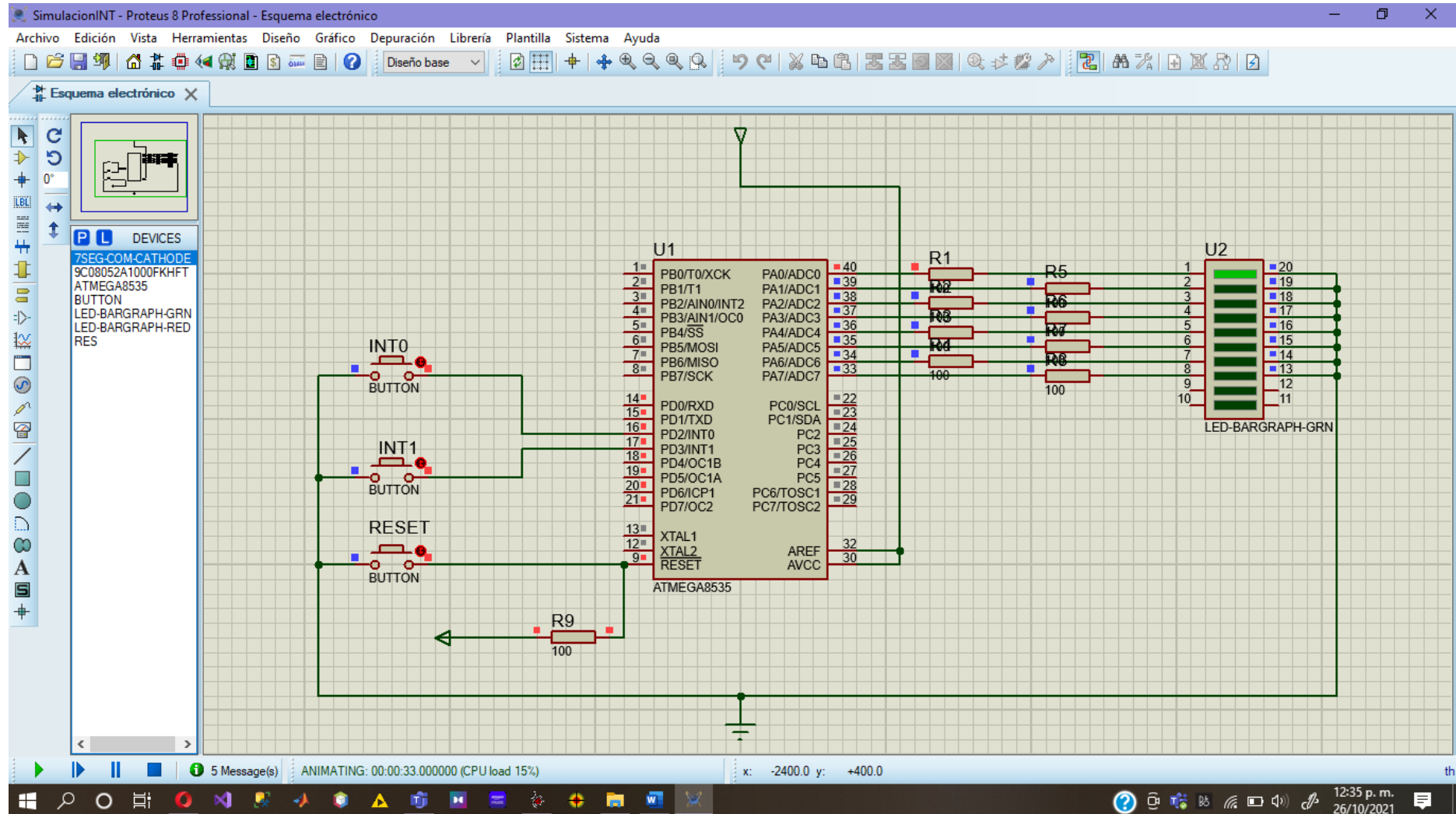
Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x000030	48	0	48	8192	0.6%
[.dseg]	0x000060	0x000060	0	0	0	512	0.0%
[.eseg]	0x000000	0x000000	0	0	0	512	0.0%

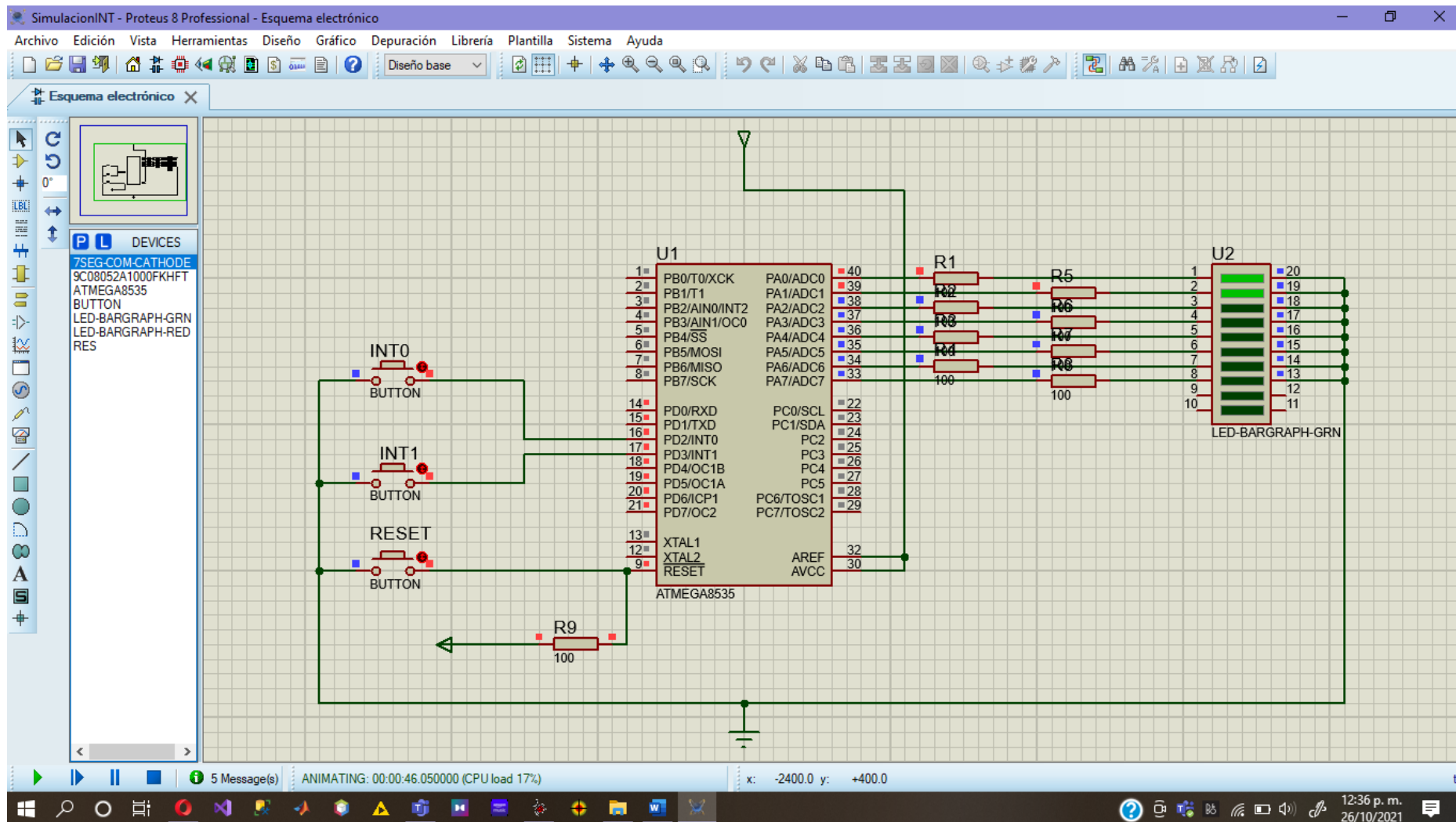
Build Message Find in Files Breakpoints and Tracepoints

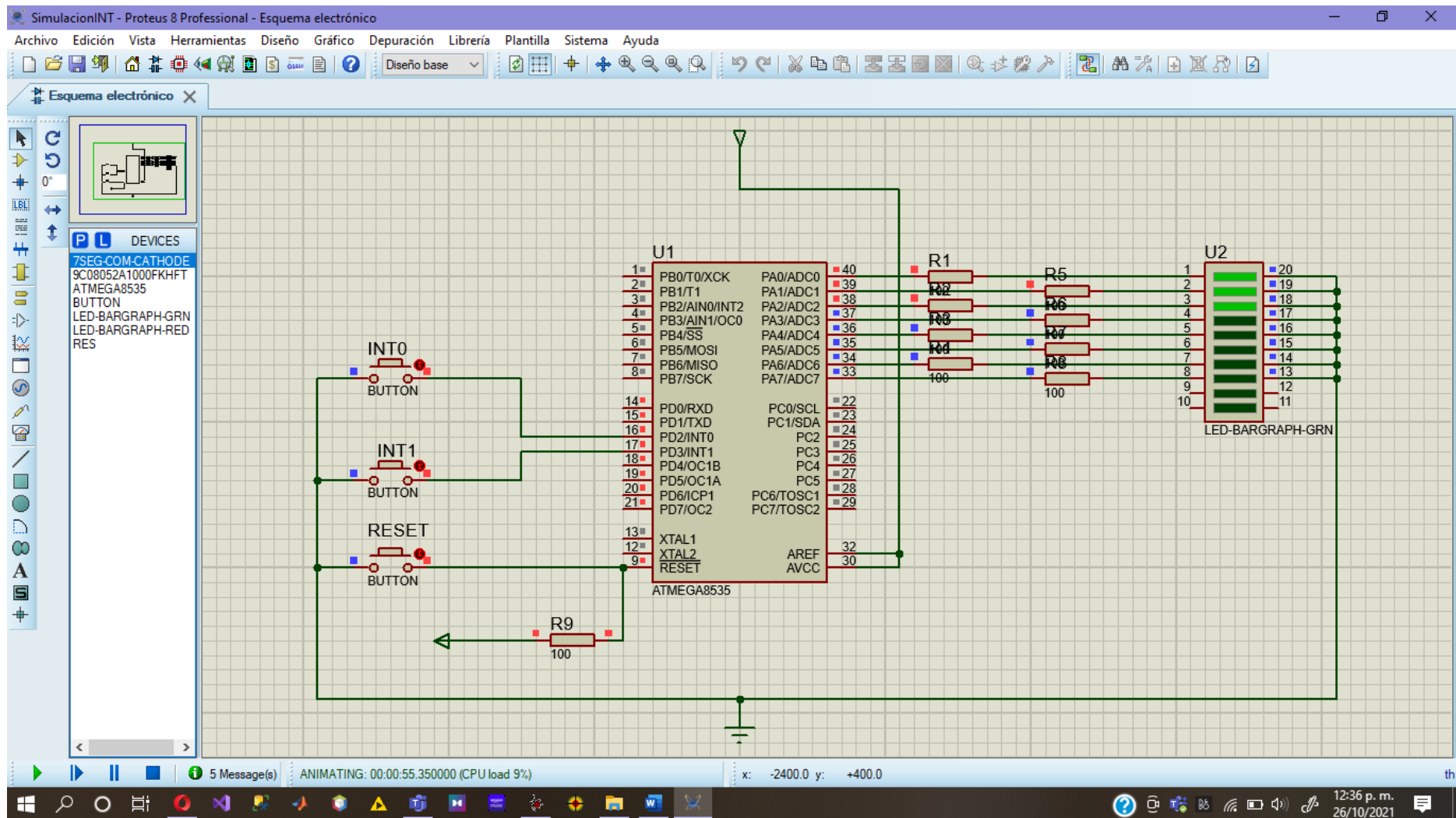
ATmega8535 AVR Simulator Auto Stopped Ln 4, Col 1 CAP NUM OVR

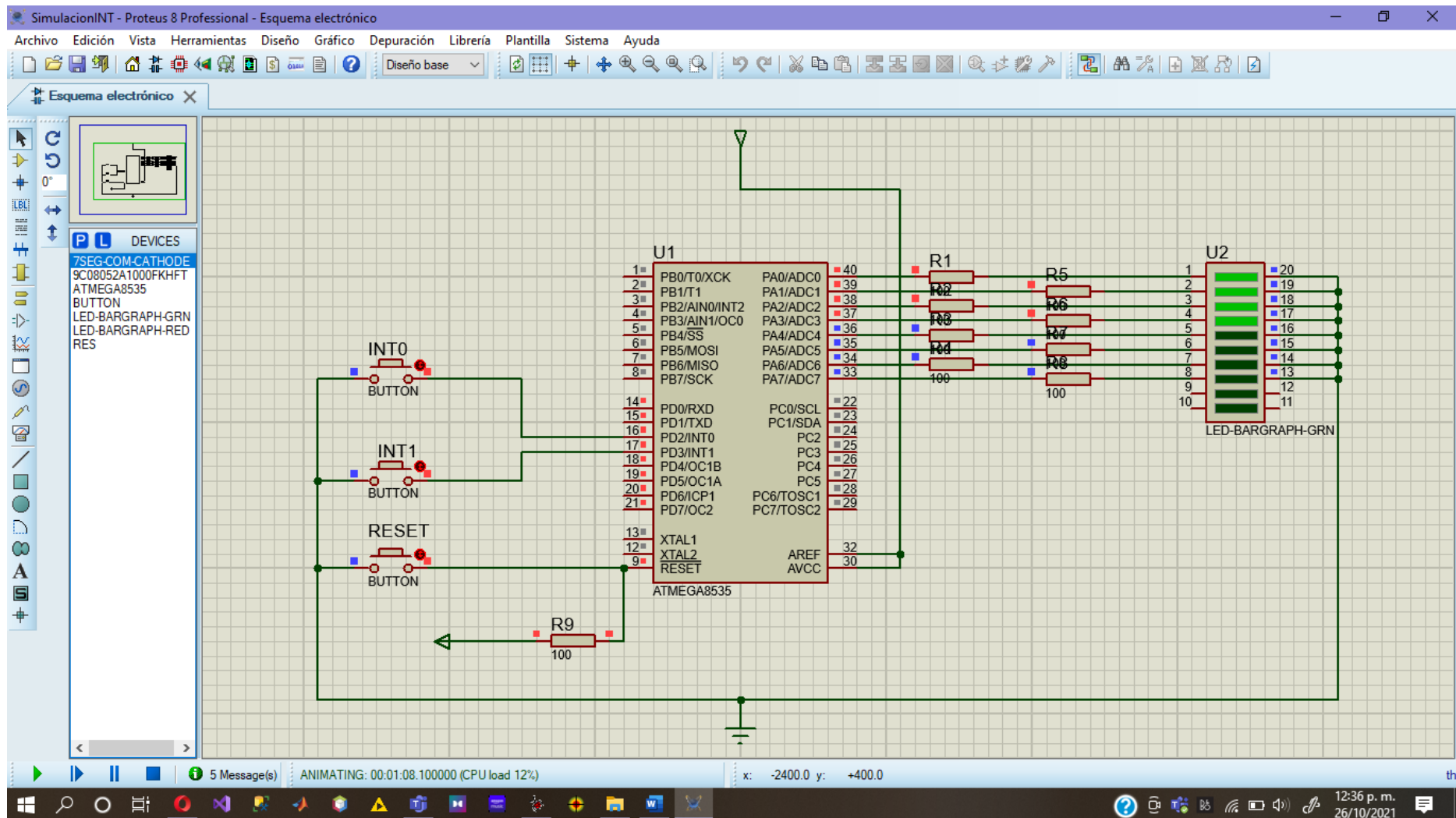
12:29 p. m. 26/10/2021

## Simulación en Proteus 8 Professional

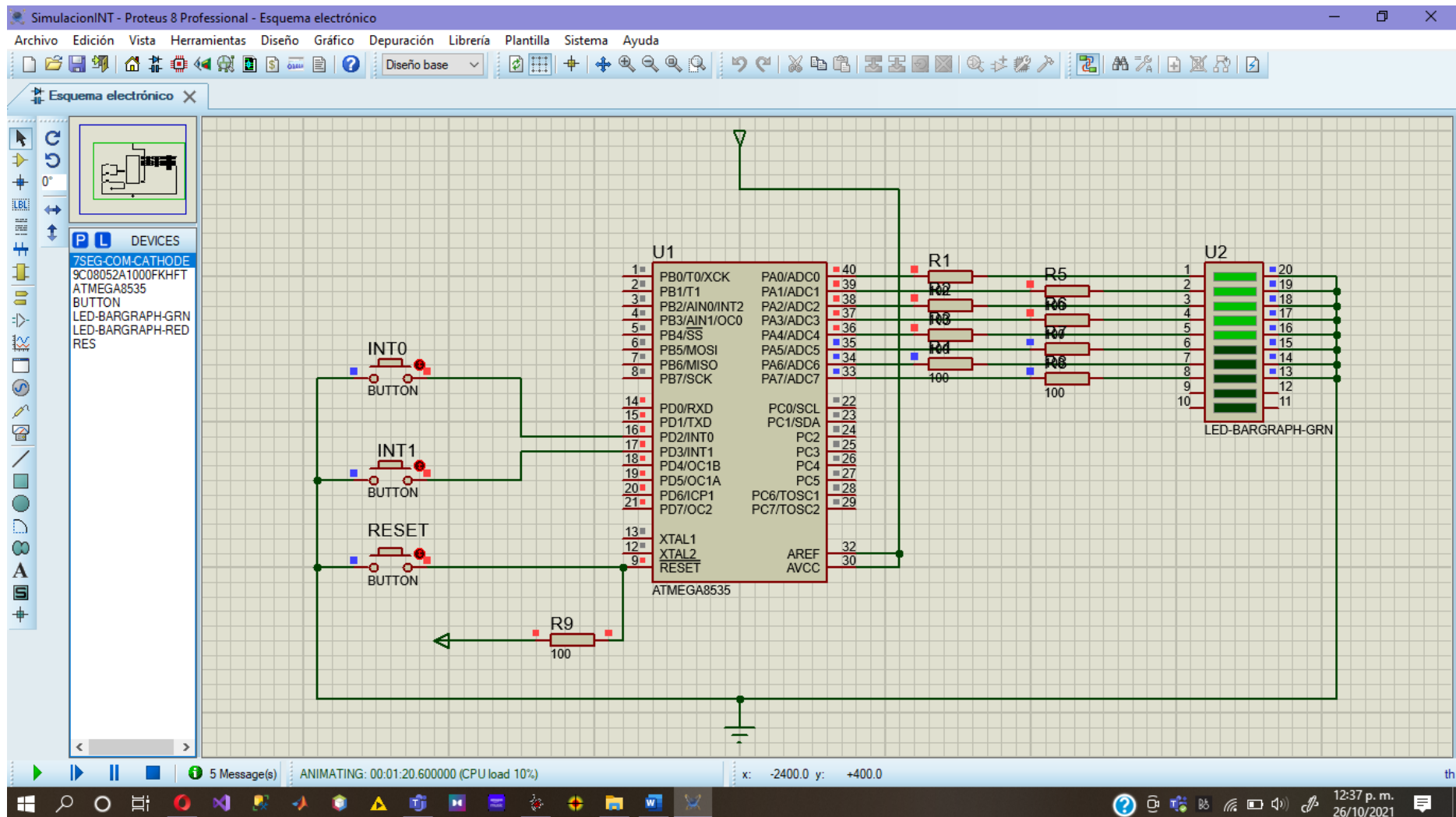


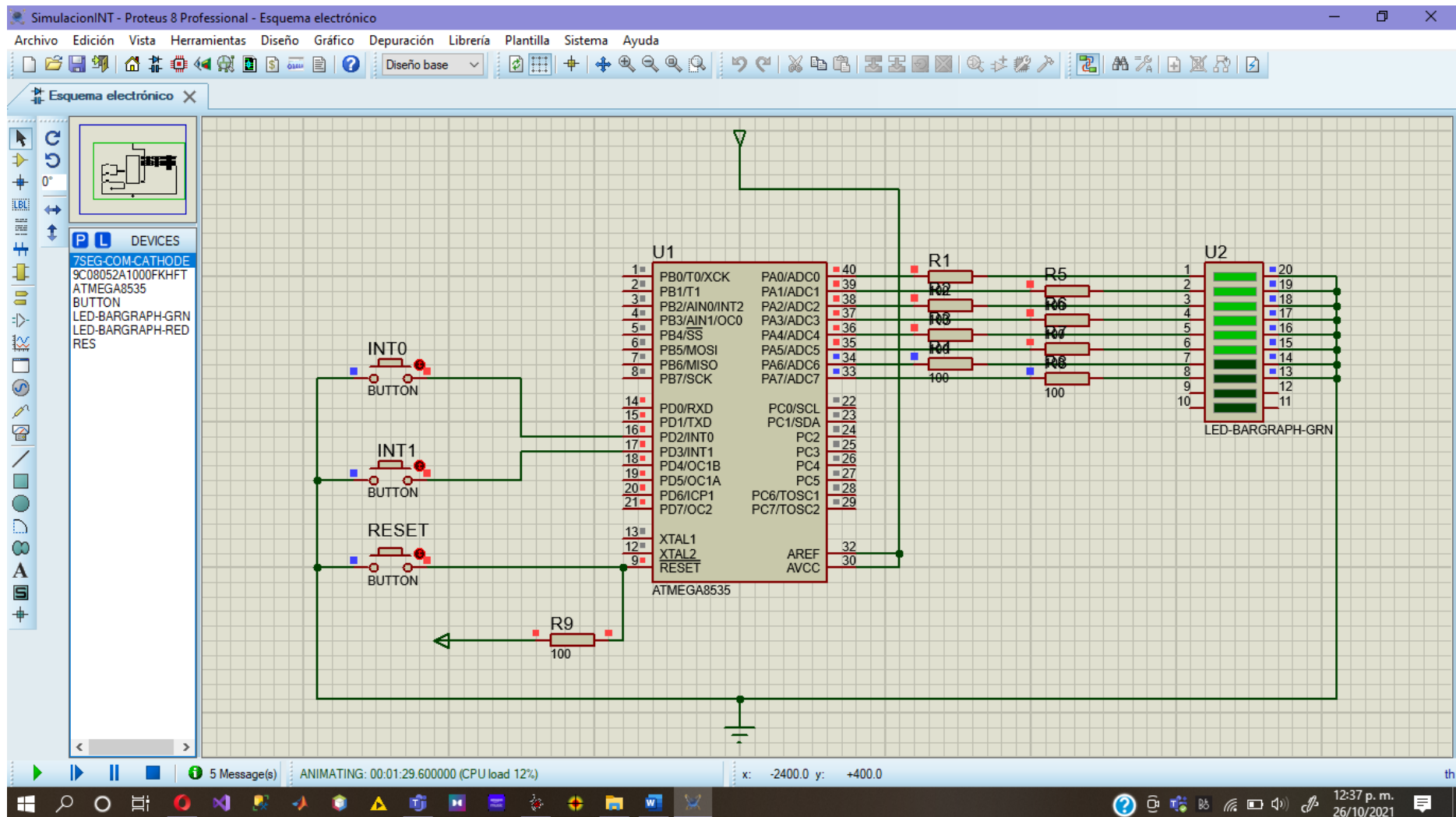


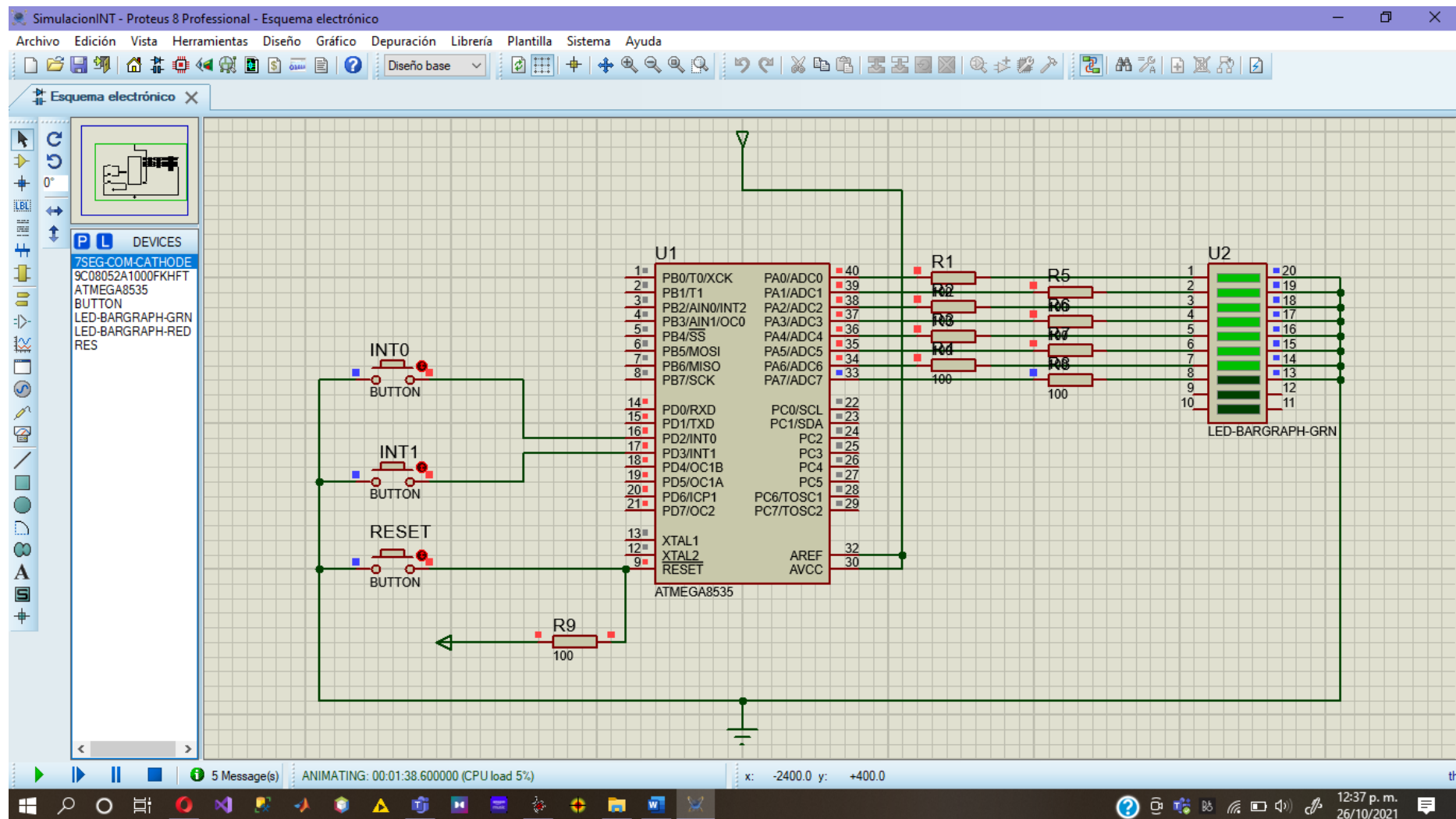


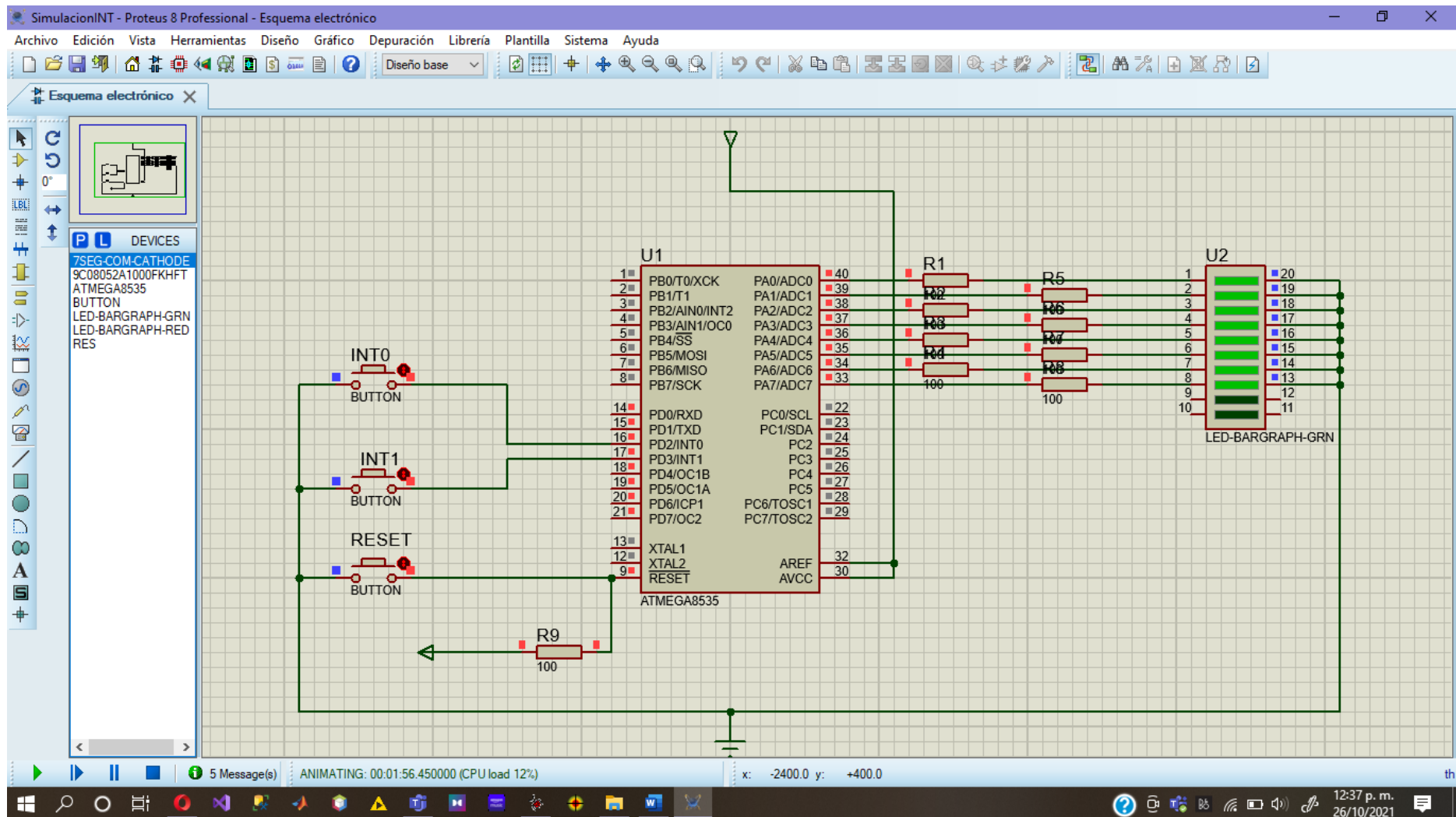












## **Conclusiones**

- **Castro Cruces Jorge Eduardo**

Esta práctica se logró el objetivo principal, que fue desarrollar un programa que, utilizando las interrupciones externas INT0 e INT1, realice lo siguiente:

- Se tendrán conectados interruptores momentáneos (pushboton) en INT0 e INT1
- 8 leds conectados en el puerto A
- Al inicio por cada vez que se pulsa el botón conectado a INT0 se deberá conmutar el estado del PA0, y por cada vez que se pulsa el botón conectado en INT1 se deberá cambiar al siguiente led en el puerto A a conmutar.

- **Cortes Ramírez Roberto Carlos**

En esta práctica desarrollamos un contador en cual por medio del uso de macros repite una secuencia determinada al presionar los botones en el circuito. Los valores de la secuencia fueron almacenados en registros los cuales son llamados dependiendo de lo que se quiera mostrar. Para mostrar el circuito se hizo uso del simulador proteus el cual también se utilizó para simular el código de la presente práctica y el AVR Studio para poder compilar el código.

- **Domínguez Acosta José Práxedes**

En esta práctica pude entender cómo funcionan las interrupciones en el microcontrolador. Con las interrupciones en el microcontrolador mediante eventos de flancos de subida o de baja, Las cuales pude observar en esta práctica que al presionar el botón de las interrupciones, el programa hace otro funcionamiento el cual es dependiendo de las interrupciones que se señaló. Para mostrar el circuito se hizo uso de proteus el cual también se utilizó para simular el código de la presente práctica.