



**INSTITUTO POLITÉCNICO  
NACIONAL**  
ESCUELA SUPERIOR DE CÓMPUTO



**“Convertidor ADC con Matriz Led”**

**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

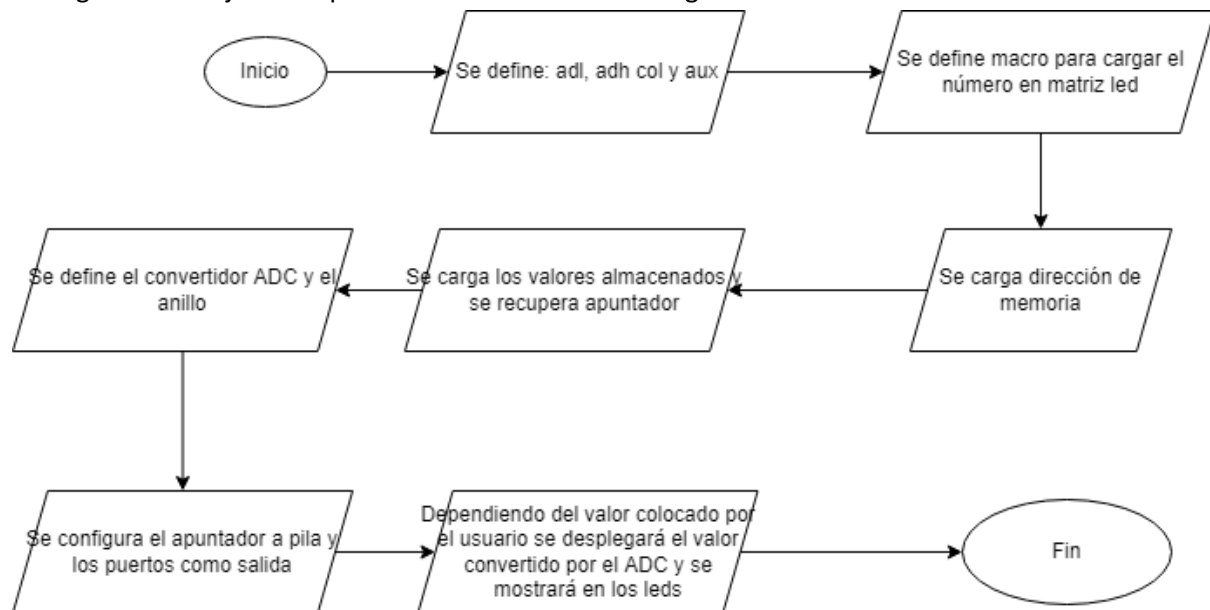
## Planteamiento del Problema

Inventar algo xd el profe no puso el enunciado

## Desarrollo

Para realizar la implementación se utilizó como base el programa visto en la sesión de clase para manejo del convertidor análogo digital, partiendo de allí se añadieron 2 funcionalidades, primero, el uso de divisores de voltaje en el circuito para generar distintos valores analogicos, y segundo, el manejo de una matriz led por cada valor generado por los divisores de voltaje.

El diagrama de flujo correspondiente al resultado es el siguiente:



**Figura 1.** Diagrama de Flujo

El respectivo código en ensamblador es el siguiente:

```

;programa para manejo del convertidor ADC

.include "m8535def.inc"
.def adl = r16
.def adh = r17
.def col = r18
.def aux = r19

;macro para cargar numero en matriz led
.macro num
    ;guardando apuntador
    push zh
    push zl

    ;cargando direccion de memoria
    ldi ZH, high(@0<<1)
    ldi ZL, low(@0<<1)

    ;cargando valores almacenados
    lpm r0, Z+
    lpm r1, Z+
    lpm r2, Z+
    lpm r3, Z+
    lpm r4, Z+
    lpm r5, Z+
    lpm r6, Z+
    lpm r7, Z

    ;recuperando apuntador
    pop zl
    pop zh
.endm

```

**Figura 2.** Primera Sección de Código

```

reset:
    rjmp start

;timer0 para contador de anillo
.org $008
rjmp cargarMatriz
rjmp anillo

;direccion de memoria del convertidor ADC
.org $0E
rjmp conv

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

;configurando puertos como salidas
ser aux
out ddrb,aux
out ddrc,aux
out ddrd,aux

;cargando $ED (1110 1101) en adcsra para activar:
;+ aden = encender convertidor ADC
;+ adsc = iniciar conteo de conversion
;+ adate = modo carrera libre (repetir conversion infinitamente)
;+ adie = activar interrupcion por conversion
;+ adps = factor de division de 32
ldi aux,$ED
out adcsra,aux

```

**Figura 3.** Segunda Sección del Código

```

; cargando $00 en admux para tener:
;+ pin AREF para voltaje de referencia
;+ alineacion a la izquierda (usar $00 para derecha)
;+ ADC0 como entrada al convertidor ADC
ldi aux,$20
out admux,aux

;timer0 y timer1 sin preescalado
ldi aux,2
out tccr0,aux
out tccr1b,aux

;habilitando desborde timer0 y timer1
ldi aux,5
out tmsk,aux

;habilitando interrupciones globalmente
sei

;reiniciando apuntador z
clr zh
clr zl

;reiniciando contador de anillo
ldi col,1

;ciclando el programa
loop:
    rjmp loop

```

**Figura 4.** Tercera Sección del Código

```

conv:
    ;leyendo conversion ADC
    in adl,adcl
    in adh,adch

    ;cargando el resultado
    out portd,adh

    reti

anillo:
    ;cargando valor de las filas
    com zh
    out portc,zh
    com zh

    ;cargando valor de las filas
    ld aux,z+
    com aux
    out portc,aux

    ;cargando valor del contador de anillo
    out portb,col
    lsl col

    ;verificando si es necesario reset
    cpi col,0
    brne saltarReinicio

    ;reset del contador de anillo y z
    ldi col,1
    clr zh
    clr zl

    saltarReinicio:
    reti

```

**Figura 5.** Cuarta Sección del Código

```
cargarMatriz:
    cpi adh,$00
    breq cargaX

    cpi adh,$19
    breq carga0

    cpi adh,$33
    breq carga1

    cpi adh,$4c
    breq carga2

    cpi adh,$66
    breq carga3

    rjmp trampolin

cargaX:
    num carita
    reti

carga0:
    num cero
    reti

carga1:
    num uno
    reti
```

**Figura 6.** Quinta Sección del Código

```
carga2:
    num dos
    reti

carga3:
    num tres
    reti

trampolin:
    cpi adh,$80
    breq carga4

    cpi adh,$99
    breq carga5

    cpi adh,$b3
    breq carga6

    cpi adh,$cc
    breq carga7

    cpi adh,$e6
    breq carga8

    rjmp carga9

carga4:
    num cuatro
    reti
```

**Figura 7.** Sexta Sección del Código

```
carga5:
|  num cinco
|  reti

carga6:
|  num seis
|  reti

carga7:
|  num siete
|  reti

carga8:
|  num ocho
|  reti

carga9:
|  num nueve
|  reti
```

**Figura 8.** Séptima Sección del Código



```

cero:
| .db $00,$18,$24,$24,$24,$24,$18,$00

uno:
| .db $00,$18,$38,$18,$18,$18,$3c,$00

dos:
| .db $00,$18,$24,$08,$10,$20,$3c,$00

tres:
| .db $00,$38,$04,$18,$18,$04,$38,$00

cuatro:
| .db $00,$04,$0c,$14,$3c,$04,$04,$00

cinco:
| .db $00,$3c,$20,$38,$04,$04,$38,$00

seis:
| .db $00,$1c,$20,$38,$24,$24,$18,$00

siete:
| .db $00,$38,$08,$08,$3c,$08,$08,$00

ocho:
| .db $00,$18,$24,$18,$18,$24,$18,$00

nueve:
| .db $00,$18,$24,$24,$1c,$04,$04,$00
|
carita:
| .db $7e,$81,$a5,$81,$a5,$99,$81,$7e

```

**Figura 9.** Octava Sección del Código

Para la simulación se muestra el valor inicial de la matriz led, los divisores de voltaje y algunos casos donde se presionan los pulsadores que están conectados a los diversos divisores de voltaje:

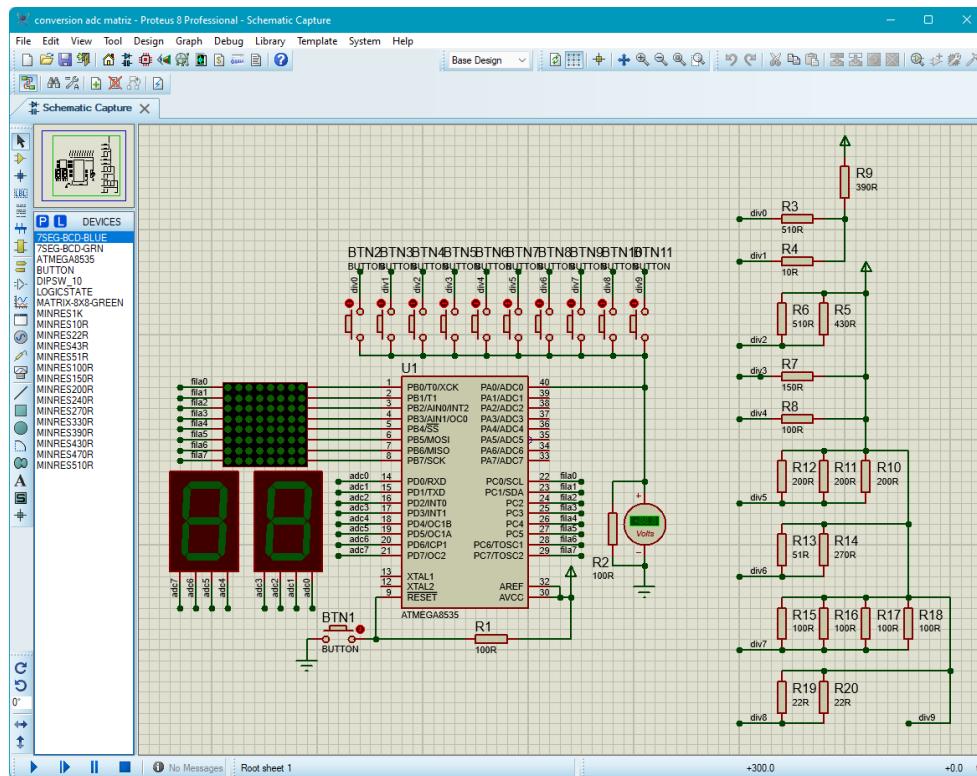


Figura 10. Vista General del Circuito

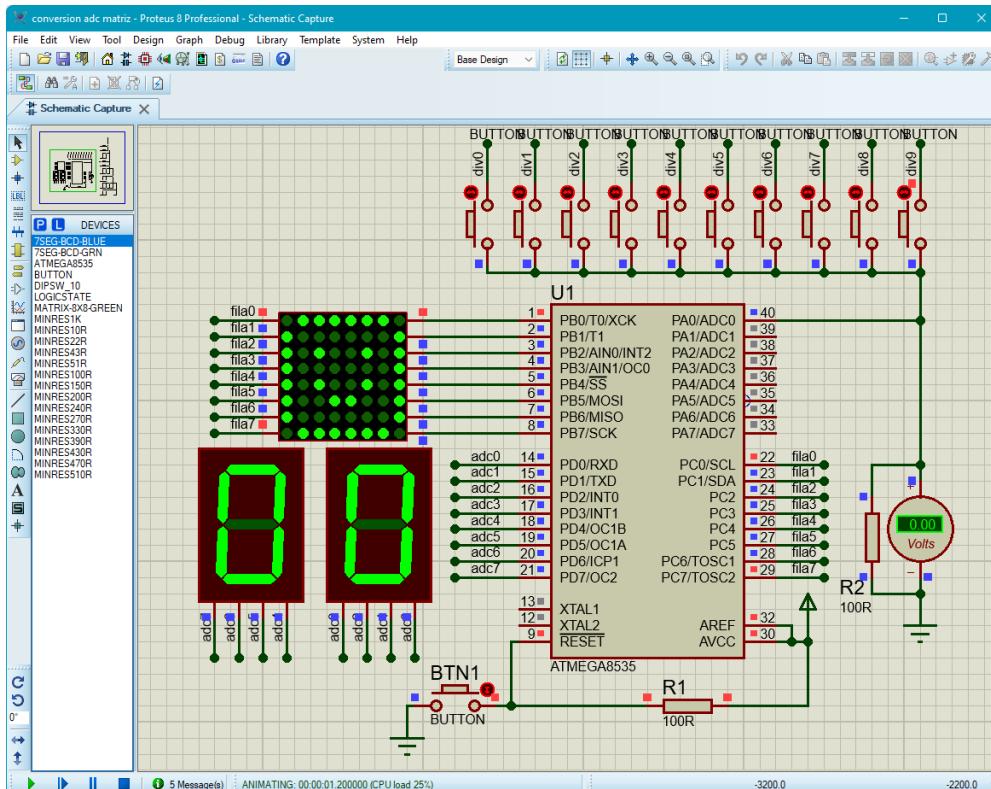


Figura 11. Valor Inicial de la Matriz Led

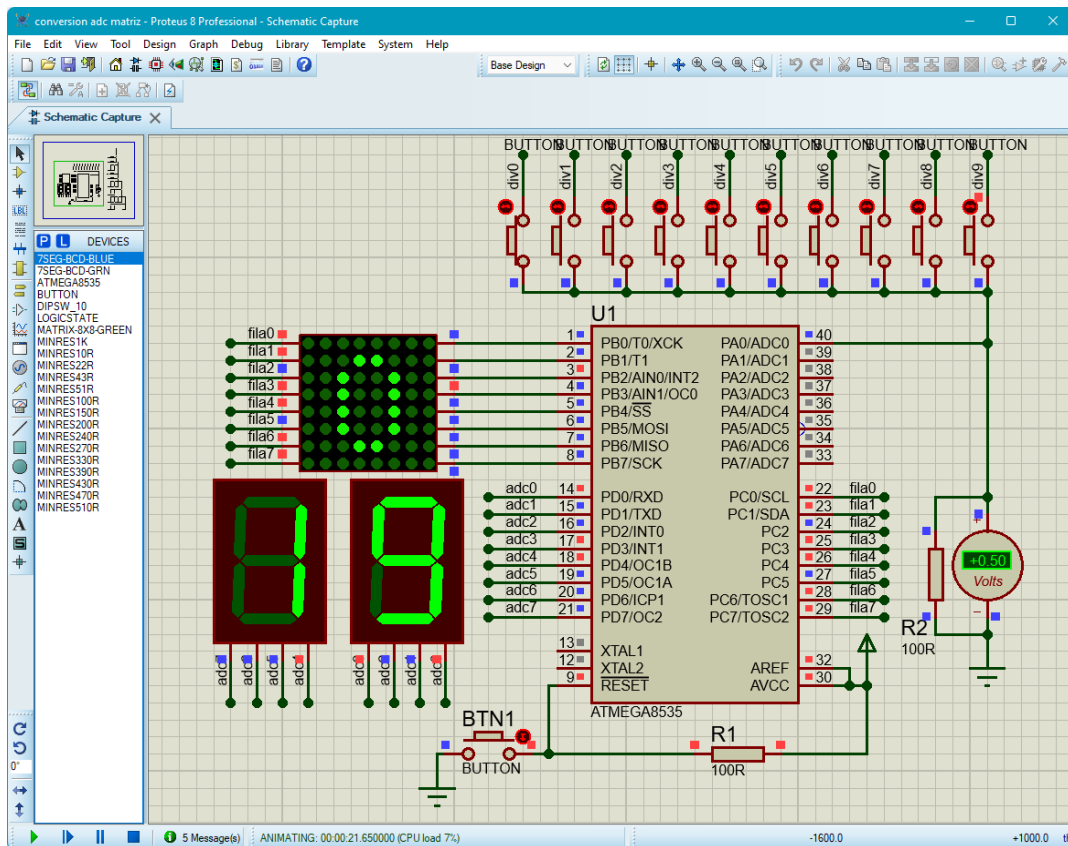


Figura 12. Valor al Presionar el Push Button Número 0

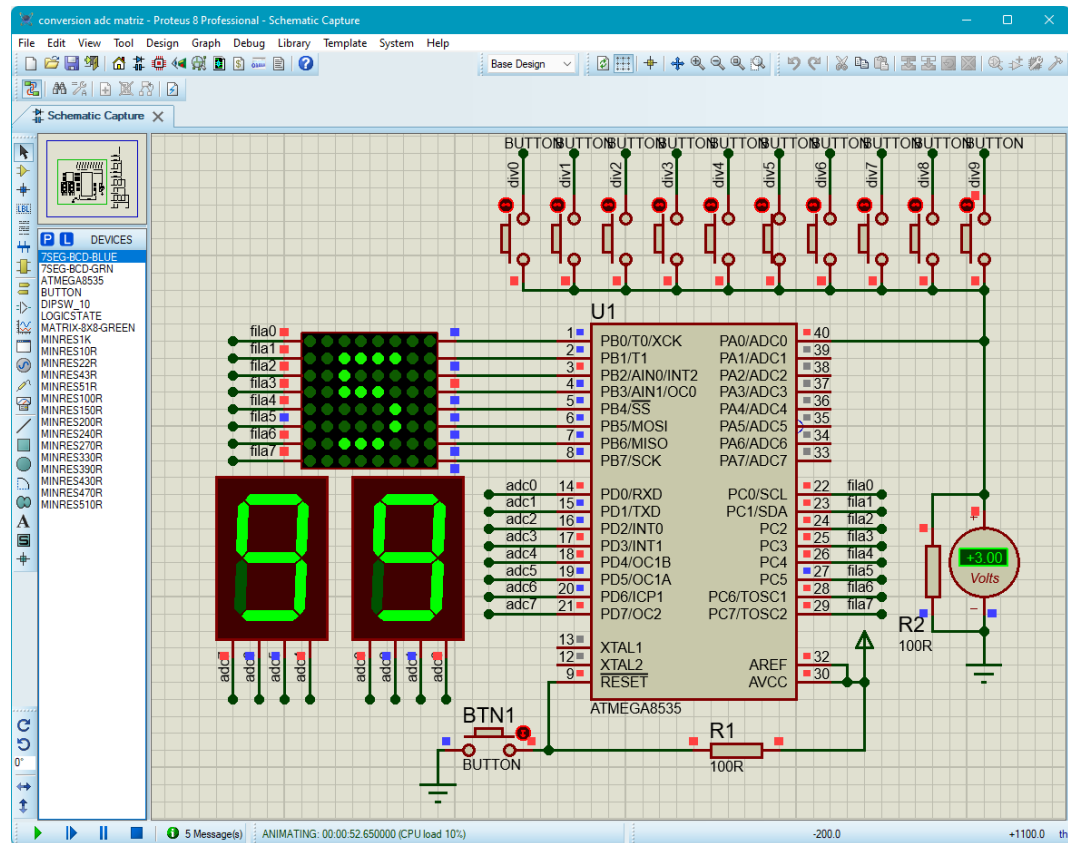


Figura 13. Valor al Presionar el Push Button Número 5

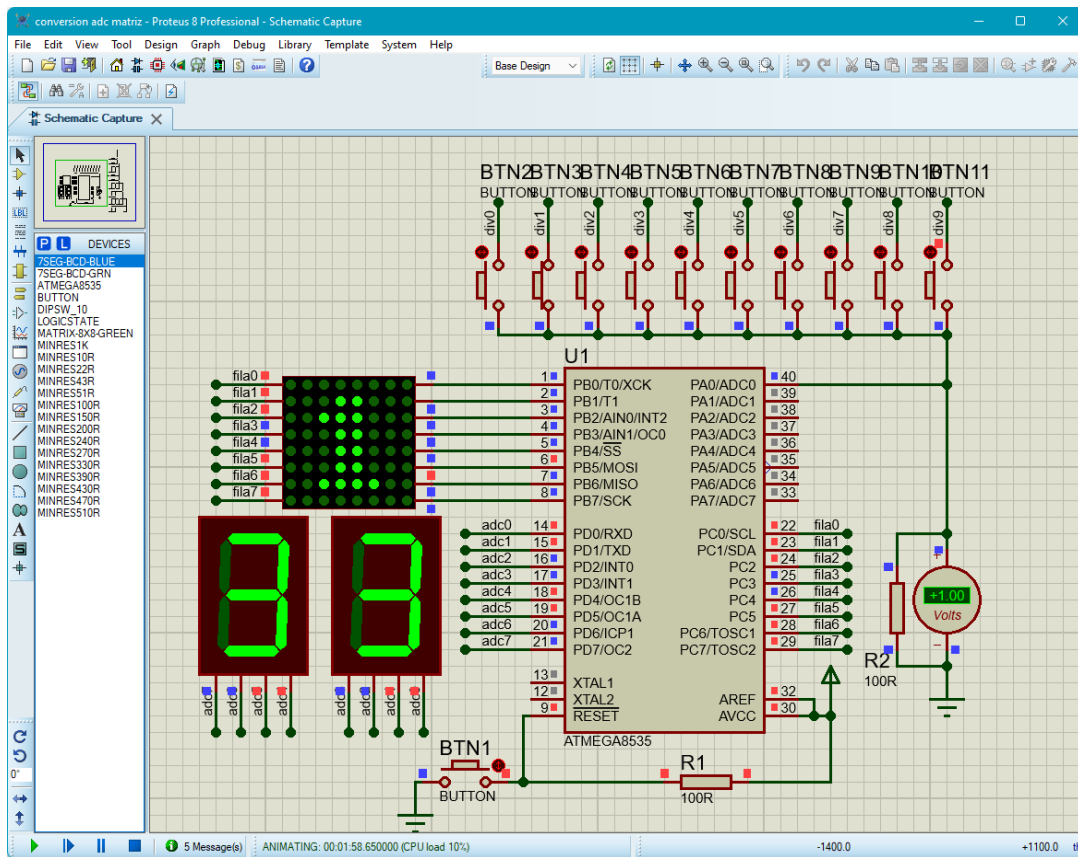


Figura 14. Valor al Presionar el Push Button Número 1

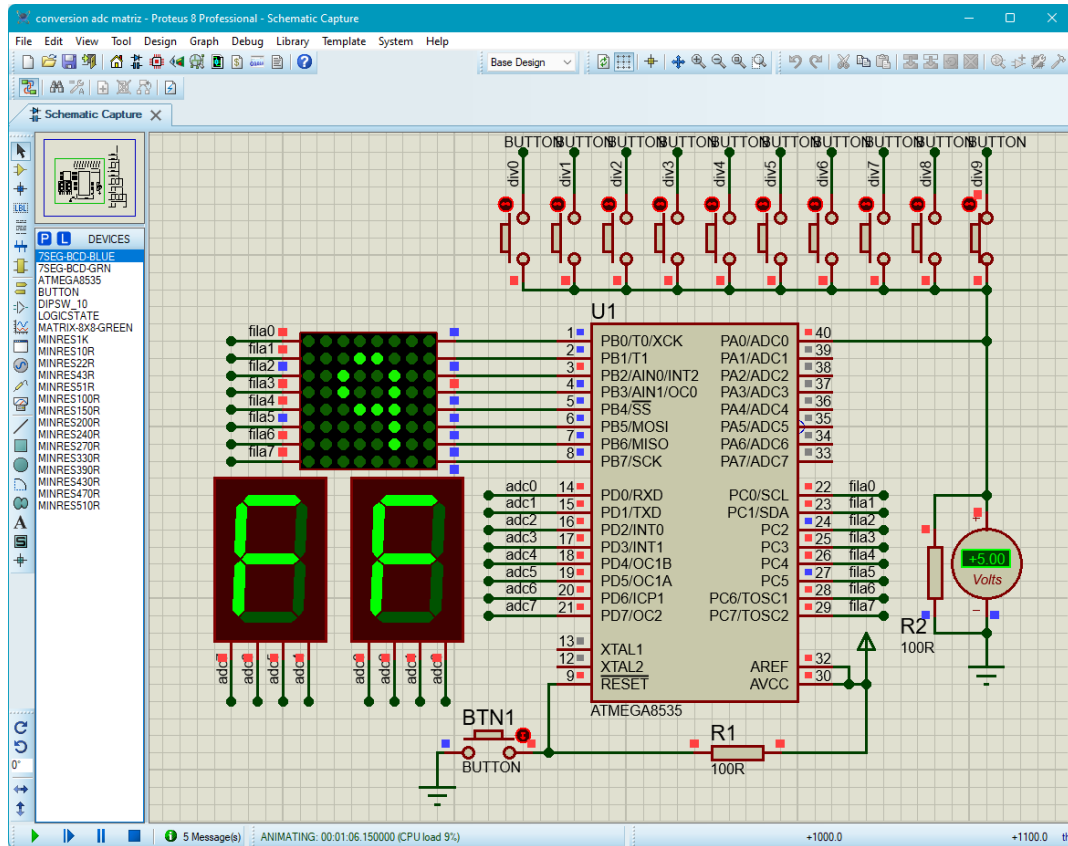


Figura 15. Valor al Presionar el Push Button Número 9

## **Conclusiones Individuales**

### **Contreras Cardoso Adolfo**

Con el apoyo de mis compañeros y la retroalimentación que me han hecho pasar, logré entender cómo fue la implementación de esta práctica, agradezco a mi compañero Santiago por apoyarme en los trabajos, con constancia y esfuerzo se pudo realizar dicho trabajo y así estar preparados para las futuras prácticas que vienen.

### **Martínez Alvarado Bryan Alexis**

Durante la realización de esta práctica logré analizar cómo es que los lenguajes tanto ensamblador como c pueden trabajar en conjunto para generar funciones específicas, aplicando lo visto en la parte teórica de la clase, y de igual manera realizando la simulación de la misma.

### **Maya Martínez Alonso Rubén**

Esta práctica requirió conocimientos de muchas de las prácticas anteriores y resultó complicado, en lo particular me resultó más difícil de entender que otras prácticas ya que los temas usados son más complicados de los que han sido las prácticas pasadas. La ayuda de mis compañeros me ayudó a entender que se hacía en esta práctica.

### **Pérez Gómez Santiago**

La realización de este trabajo resultó ser el mayor desafío hasta ahora, se necesitaron todos los conocimientos adquiridos con cada práctica, de manera que lo más complicado fue configurar la matriz led y ajustar su funcionamiento, teniendo que usar el método del “trampolín” para realizar saltos o brincos condicionales, con todo lo anterior, podemos finalizar con el uso de ensamblador y comenzar con lenguaje C para trabajos futuros.