

# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

---

### **Materia**

Introducción a los microcontroladores

### **Ejercicio 5**

#### **Integrantes del equipo (Sala 3):**

Azpeitia Hernández Vladimir

Chávez Avila Osvaldo Antonio

Fabela Galván Aarón Jhair

Pérez Fabian Noé

### **Grupo**

3CM17

### **Profesor**

Juan José Pérez Pérez

### **Fecha de entrega**

25 abril del 2021

---

# Índice

<b>Introducción</b>	<b>3</b>
Planteamiento del problema	3
<b>Desarrollo</b>	<b>3</b>
Código de la solución	3
Programa principal	3
Subrutina para convertir un binario a 7 segundos	5
Subrutina para generar retardo	7
Simulación del circuito en Proteus 8 Professional	7
<b>Conclusiones</b>	<b>9</b>

# Ejercicio 5

## Introducción

### Planteamiento del problema

Para el circuito dado en la Figura 1, escribe un programa que muestre una cuenta entre los números 00 y 15 (\$x0 y \$xF) que será el número presente en los bits b3, b2, b1 y b0 del puerto B

El tiempo entre cuenta y cuenta irá desde 1 segundo hasta 15 segundos (\$1y a \$Fy) que será el nibble alto del dato presente en el puerto B, si el nibble alto es cero (\$0y) solo se mostrará la decodificación del nibble bajo presente en el puerto B en los dos displays.

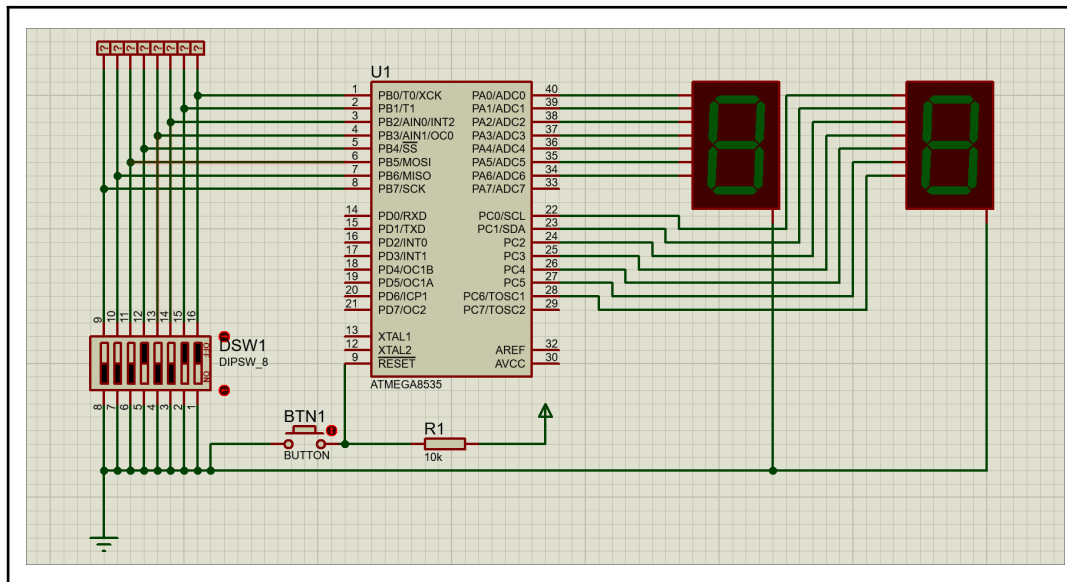


Figura 1. Circuito propuesto en el planteamiento del problema

## Desarrollo

### Código de la solución

#### Programa principal

```
.include "m8535def.inc"
.def input = r16
.def aux = r17
.def nbajo = r18
.def nalto = r19
```

```

.def convertir = r20
.def convertir2 = r21
.def segundos = r25
.def cuenta = r26

ser aux
out ddra,aux
out portb, aux
out ddrc,aux
ldi aux,low(RAMEND)
out spl,aux
ldi aux,high(RAMEND)
out sph,aux
leer:
    in input,pinb
    ldi nbajo,$0f
    ldi nalto,$f0
    clr convertir2
    and nbajo,input
    and nalto,input
    swap nalto
    clr cuenta
    cpi nalto,0
    brne contar
    mov convertir,nbajo
    rcall conversion
    nop
    nop
    rcall mostrar
    nop
    rjmp leer
contar:
    mov convertir,cuenta
    rcall conversion
    rcall mostrar
    mov segundos,nalto
    rcall retardo
    inc cuenta
    cp nbajo,cuenta
    brge contar
    clr cuenta
    rjmp contar

mostrar:
    out portc,convertir
    out porta,convertir2
    ret

```

Figura 2. Código principal de la solución

## Subrutina para convertir un binario a 7 segundos

```
conversion:
    cpi convertir,0
    breq es0
    cpi convertir,1
    breq es1
    cpi convertir,2
    breq es2
    cpi convertir,3
    breq es3
    cpi convertir,4
    breq es4
    cpi convertir,5
    breq es5
    cpi convertir,6
    breq es6
    cpi convertir,7
    breq es7
    cpi convertir,8
    breq es8
    cpi convertir,9
    breq es9
    cpi convertir,10
    breq es10
    cpi convertir,11
    breq es11
    cpi convertir,12
    breq es12
    cpi convertir,13
    breq es13
    cpi convertir,13
    breq es13
    cpi convertir,14
    breq es14
    cpi convertir,15
    breq es15

    ldi convertir,$5b
    ldi convertir2,$06
    ret
es0:
    ldi convertir,$3f
    ret
es1:
    ldi convertir,$06
    ret
es2:
    ldi convertir,$5b
    ret
es3:
```

```

        ldi convertir,$4f
        ret
es4:
        ldi convertir,$66
        ret
es5:
        ldi convertir,$6d
        ret
es6:
        ldi convertir,$7d
        ret
es7:
        ldi convertir,$07
        ret
es8:
        ldi convertir,$7f
        ret
es9:
        ldi convertir,$67
        ret
es10:
        ldi convertir,$3f
        ldi convertir2,$06
        ret
es11:
        ldi convertir,$06
        ldi convertir2,$06
        ret
es12:
        ldi convertir,$5b
        ldi convertir2,$06
        ret
es13:
        ldi convertir,$4f
        ldi convertir2,$06
        ret
es14:
        ldi convertir,$66
        ldi convertir2,$06
        ret
es15:
        ldi convertir,$6d
        ldi convertir2,$06
        ret

```

Figura 3. Subrutina para convertir binario a 7 segundos

## Subrutina para generar retardo

```
retardo:
    ldi r22, 6
    ldi r23, 19
    ldi r24, 174

L1: dec r24
    brne L1
    dec r23
    brne L1
    dec r22
    brne L1
    dec segundos
    brne retardo
    ret
```

Figura 4. Subrutina para generar un retardo

## Simulación del circuito en Proteus 8 Professional

Cuando el nibble alto de B es \$0y, solo se muestra el contenido del nibble bajo de B en los dos displays:

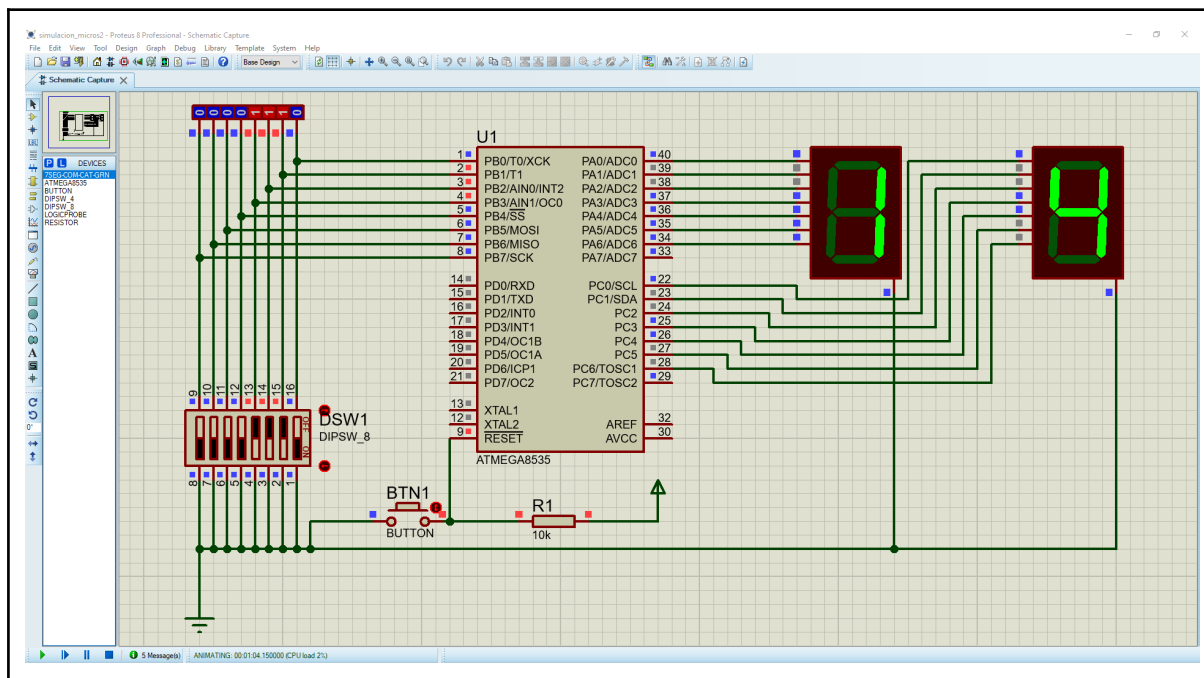


Figura 5. Simulación en Proteus 8 Professional: Captura 1

Si colocamos 1 (\$1y) en el nibble alto de B y 15 (\$0F) en el nibble bajo de B. El contador comenzará en 0 y contará hasta 15, con una duración de 1 segundo de cuenta en cuenta:

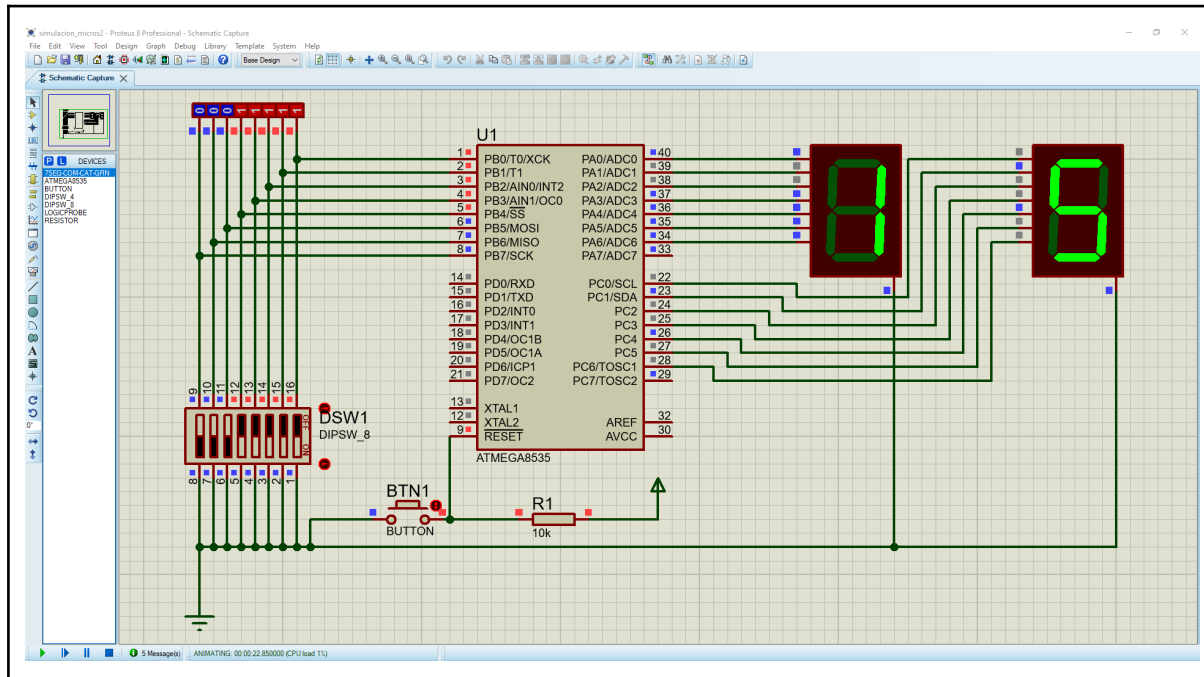


Figura 6. Simulación en Proteus 8 Professional: Captura 2

Si colocamos 15 (\$Fy) en el nibble alto de B y 3 (\$03) en el nibble bajo de B. El contador comenzará en 0 y contará hasta 3, con una duración de 15 segundos de cuenta en cuenta:

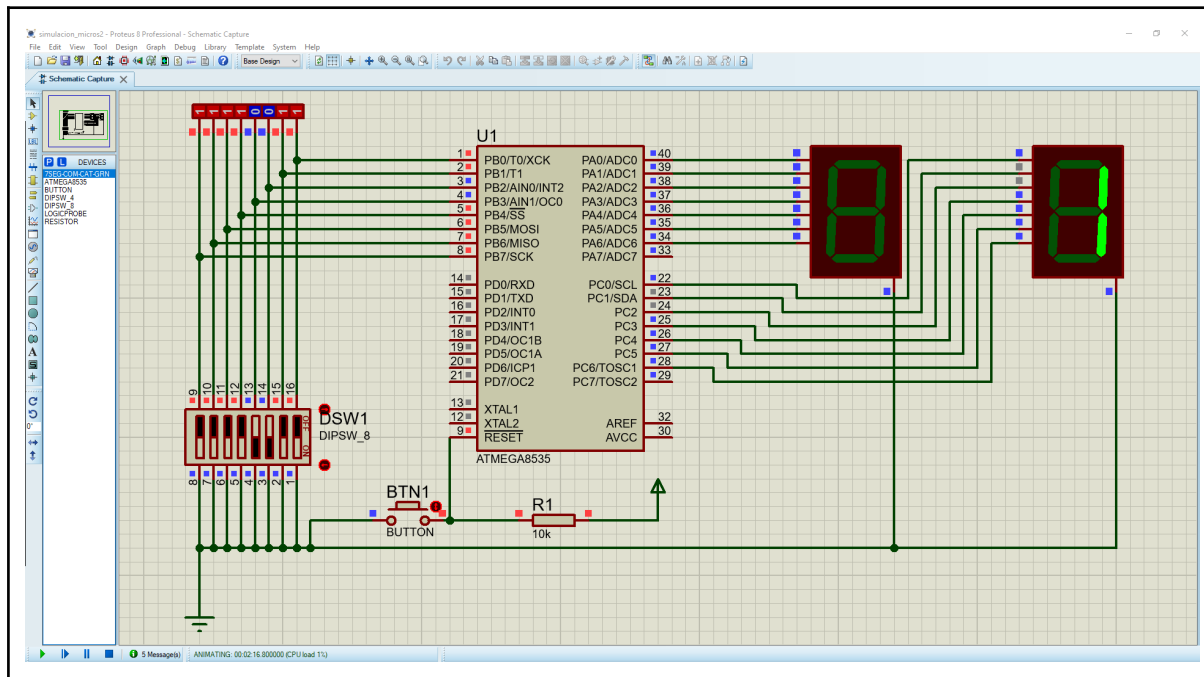


Figura 6. Simulación en Proteus 8 Professional: Captura 3



# Conclusiones

En esta práctica aprendimos a utilizar las subrutinas, las cuales son equivalentes a las funciones que conocemos de los lenguajes de programación de alto nivel. También aprendimos a generar retardos para poder utilizarlos para codificar contadores, por ejemplo. Para poder realizar la práctica también recordamos algunos conceptos que habíamos visto en ejercicios previos, por ejemplo hacer un decodificador para mostrar los números en los displays.