



# **Universidad Nacional Autónoma de México**

FACULTAD DE INGENIERÍA

División de ingeniería eléctrica



Microcomputadoras

Tarea 05

Profesor: Moises Melendez

Sem 2022-1

Grupo: 4

Equipo #7

Jaimes Ojeda Erick Simeón

Martínez Ortiz Saúl Axel

Fecha: 15/10/2021

Diseñar, programar e implementar un sistema utilizando un pic16f877a que muestre la cuenta ascendente desde 0 hasta F (desborda a 0) si la entrada en el pin 0 del puerto A es 1, si la entrada en el puerto PORTA.0 es 0 (el cambio debe ser inmediato) la cuenta deberá ser descendente (F deborda a 0), mostrar cada dígito durante 500 ms.

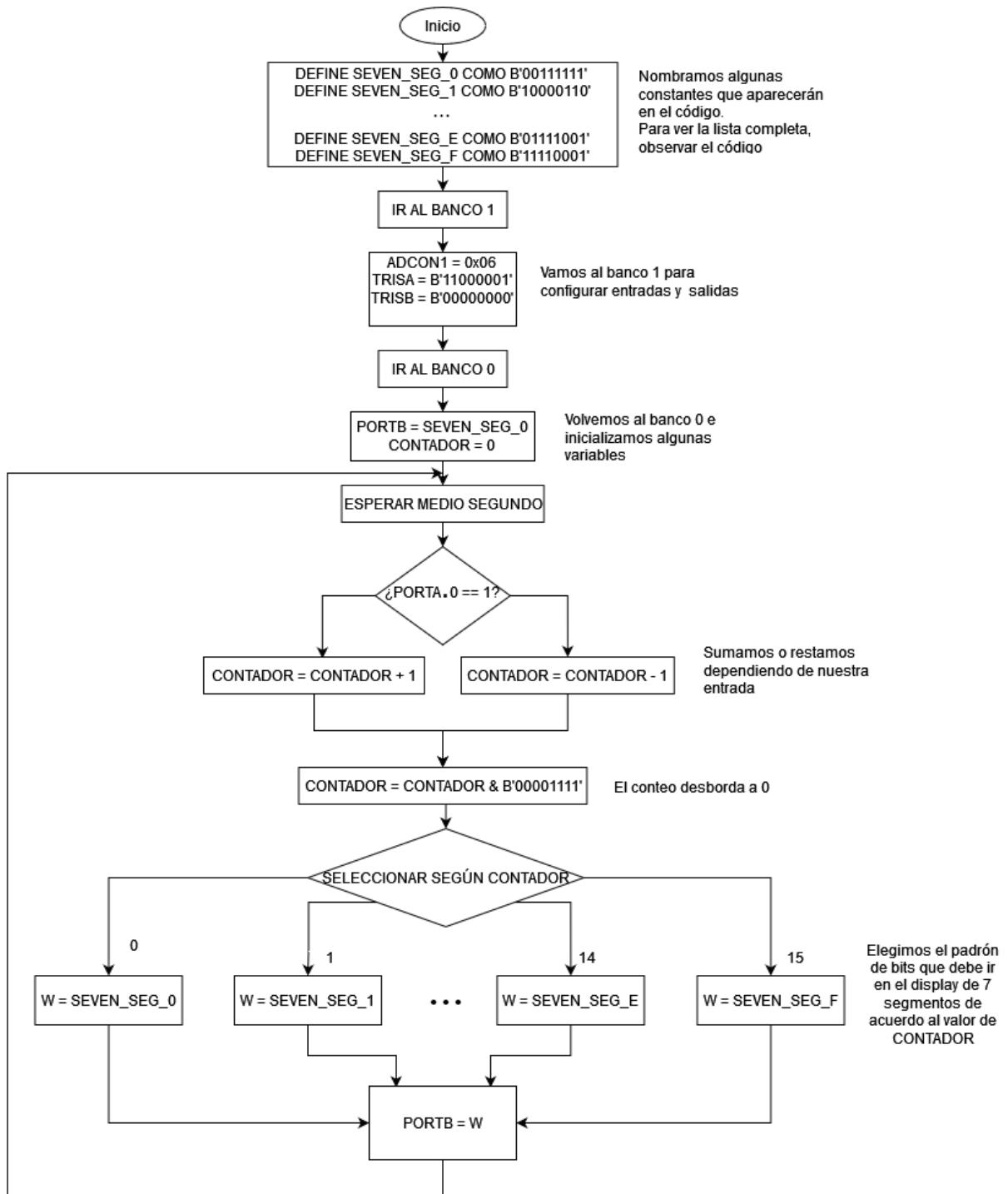
La solución deberá incluir una propuesta de solución con diagramas de flujo o pseudocódigo, código comentado y evidencias del correcto funcionamiento del sistema en Proteus.

### **Solución:**

Se selecciona dos puertos uno para la entrada de datos PUERTO A y uno para la salida PUERTO B, Se empleó una máscara de bits para tomar solo los cuatro bits menos significativos del puerto B y obtener el desbordamiento de F a 0 en la cuenta ascendente y de 0 a F en la cuenta descendente.

Para mostrar la secuencia en hexadecimal se hizo una decodificación BCD a 7 segmentos dejando como constantes su decodificación en lógica positiva, los leds encienden con un uno y se apagan con un cero.

Se hizo uso del direccionamiento indexado para hacer la selección del conteo y poder generar un switch-case en ensamblador.



C:\Users\erick\Downloads\Telegram Desktop\tarea5.asm

```
; Equipo 7 de MICROCOMPUTADORAS grupo 04  
; Tarea 5 Contador ascendente / descendente
```

```
PROCESSOR PIC16F877A  
INCLUDE <P16F877A.INC>
```

```
;;;;;;;;;; DEFINICIONES ;;;;;;;;;;  
; Localidades de memoria  
CONTADOR EQU 0X20
```

```
D1 EQU 0X21  
D2 EQU 0X22  
D3 EQU 0X23
```

```
; Bits para el display de 7 segmentos
```

```
SEVEN_SEG_0 EQU B'00111111'  
SEVEN_SEG_1 EQU B'10000110'  
SEVEN_SEG_2 EQU B'01011011'  
SEVEN_SEG_3 EQU B'11001111'  
SEVEN_SEG_4 EQU B'01100110'  
SEVEN_SEG_5 EQU B'11101101'  
SEVEN_SEG_6 EQU B'01111101'  
SEVEN_SEG_7 EQU B'10000111'  
SEVEN_SEG_8 EQU B'01111111'  
SEVEN_SEG_9 EQU B'11100111'  
SEVEN_SEG_A EQU B'01110111'  
SEVEN_SEG_B EQU B'11111100'  
SEVEN_SEG_C EQU B'00111001'  
SEVEN_SEG_D EQU B'11011110'  
SEVEN_SEG_E EQU B'01111001'
```

C:\Users\erick\Downloads\Telegram Desktop\tarea5.asm

```
SEVEN_SEG_E EQU B'01111001'
SEVEN_SEG_F EQU B'11110001'

;;;;;;;;; PROGRAMA ;;;;;;;;;;

ORG 0
    GOTO INICIO
ORG 5

INICIO:

    ; CAMBIAMOS AL BANCO 1 PARA CONFIGURAR LOS PUERTOS
    BSF STATUS, RP0

    MOVLW 0x06
    MOVWF ADCON1

    MOVLW B'11000001'
    MOVWF TRISA

    CLRF TRISB

    ; REGRESAMOS AL BANCO 0
    BCF STATUS, RP0

    ; Inicialización de variables
    MOVLW SEVEN_SEG_0
    MOVWF PORTB

    CLRF CONTADOR
```

C:\Users\erick\Downloads\Telegram Desktop\tarea5.asm

```
        CLRF CONTADOR

CICLO_PRINCIPAL:
    CALL DELAY_MEDIO_SEG
    ; Sumamos o restamos dependiendo de PORTA.0
    BTFSC PORTA, 0
    INCF CONTADOR, F
    BTFSS PORTA, 0
    DECF CONTADOR, F

    ; Para lograr que F+1 desborde a 0 y 0-1 desborde a F
    ; se aplica una máscara que deja los 4 bits menos
    ; significativos
    MOVLW B'00001111'
    ANDWF CONTADOR, F

    CALL CONTADOR_A_7SEG
    MOVWF PORTB

    GOTO CICLO_PRINCIPAL

; Rutina que nos sirve como menú de selección
; Dependiendo del valor de contador, vamos a elegir
; la secuencia de bits de salida
CONTADOR_A_7SEG:
    MOVF CONTADOR, W
    ADDWF PCL, F
    RETLW SEVEN_SEG_0
    RETLW SEVEN_SEG_1
    RETLW SEVEN_SEG_2
```

C:\Users\erick\Downloads\Telegram Desktop\tarea5.asm

```
ADDWF PCL, F
RETLW SEVEN_SEG_0
RETLW SEVEN_SEG_1
RETLW SEVEN_SEG_2
RETLW SEVEN_SEG_3
RETLW SEVEN_SEG_4
RETLW SEVEN_SEG_5
RETLW SEVEN_SEG_6
RETLW SEVEN_SEG_7
RETLW SEVEN_SEG_8
RETLW SEVEN_SEG_9
RETLW SEVEN_SEG_A
RETLW SEVEN_SEG_B
RETLW SEVEN_SEG_C
RETLW SEVEN_SEG_D
RETLW SEVEN_SEG_E
RETLW SEVEN_SEG_F
```

```
; Rutina de retraso generada proceduralmente
```

```
; DELAY = 0.5 SECONDS
```

```
; CLOCK FREQUENCY = 20 MHZ
```

```
; ACTUAL DELAY = 0.5 SECONDS = 2500000 CYCLES
```

```
; ERROR = 0 %
```

```
DELAY_MEDIO_SEG:
```

```
;2499992 CYCLES
```

```
MOVLW 0X15
```

```
MOVWF D1
```

```
MOVLW 0X01
```

```
C:\Users\erick\Downloads\Telegram Desktop\tarea5.asm

; ERROR = 0 %

DELAY_MEDIO_SEG:
    ;2499992 CYCLES
    MOVLW    0X15
    MOVWF    D1
    MOVLW    0X74
    MOVWF    D2
    MOVLW    0X06
    MOVWF    D3
DELAY_MEDIO_SEG_AUX:
    DECFSZ   D1, F
    GOTO     $+2
    DECFSZ   D2, F
    GOTO     $+2
    DECFSZ   D3, F
    GOTO     DELAY_MEDIO_SEG_AUX

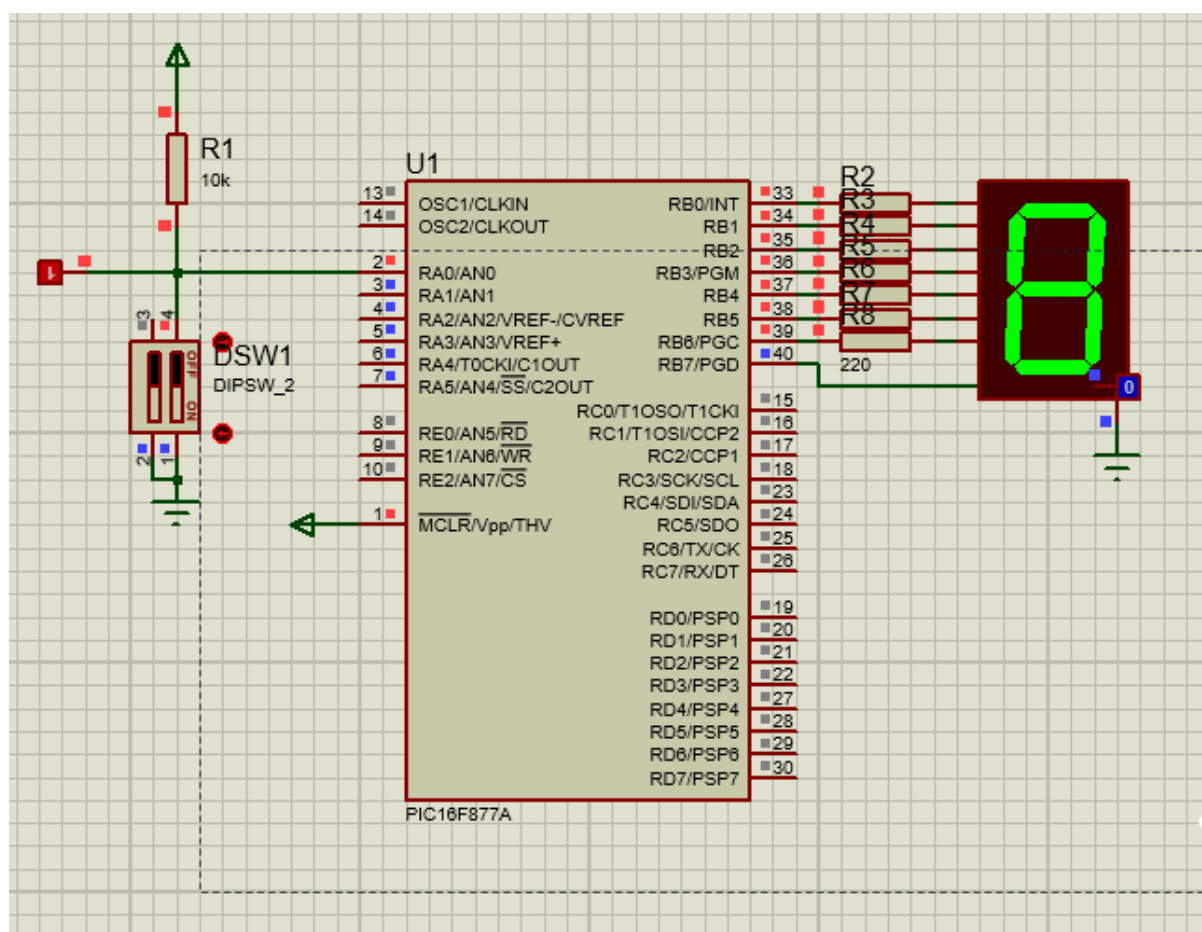
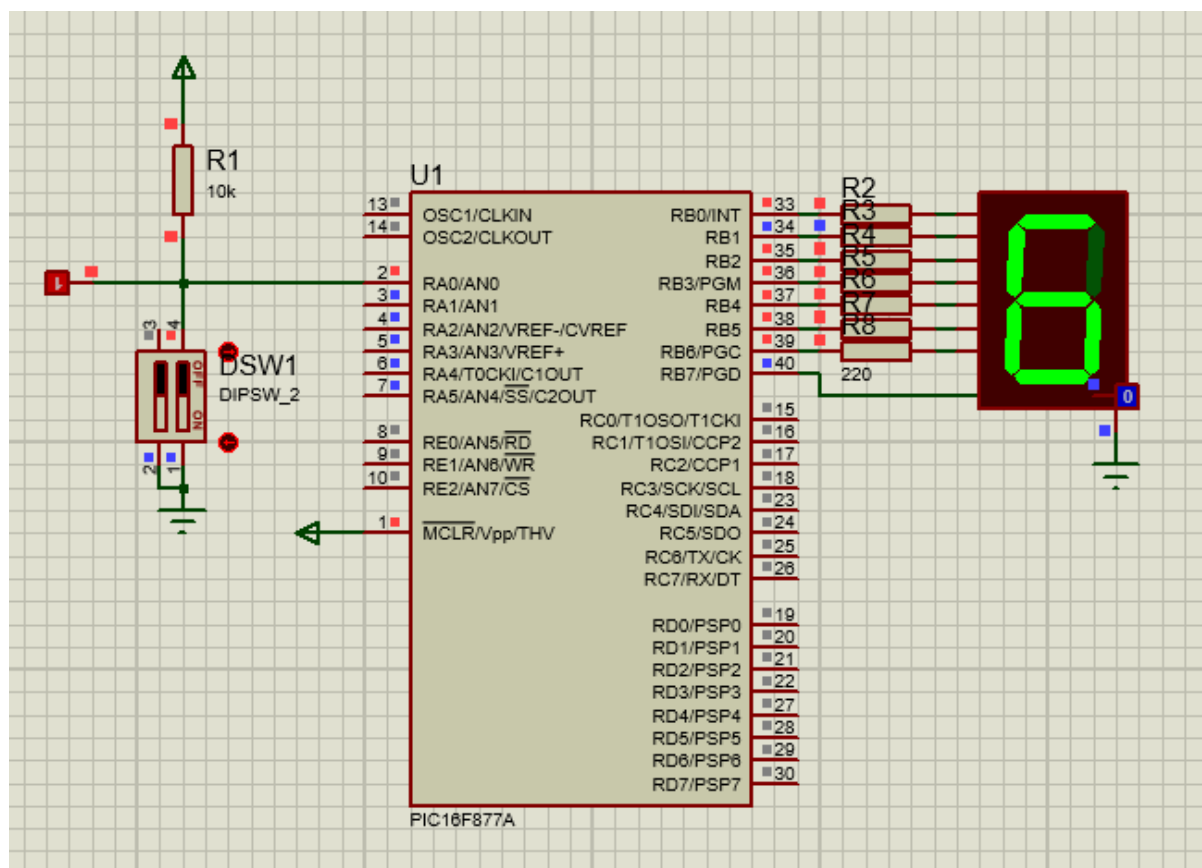
    ;4 CYCLES
    GOTO     $+1
    GOTO     $+1

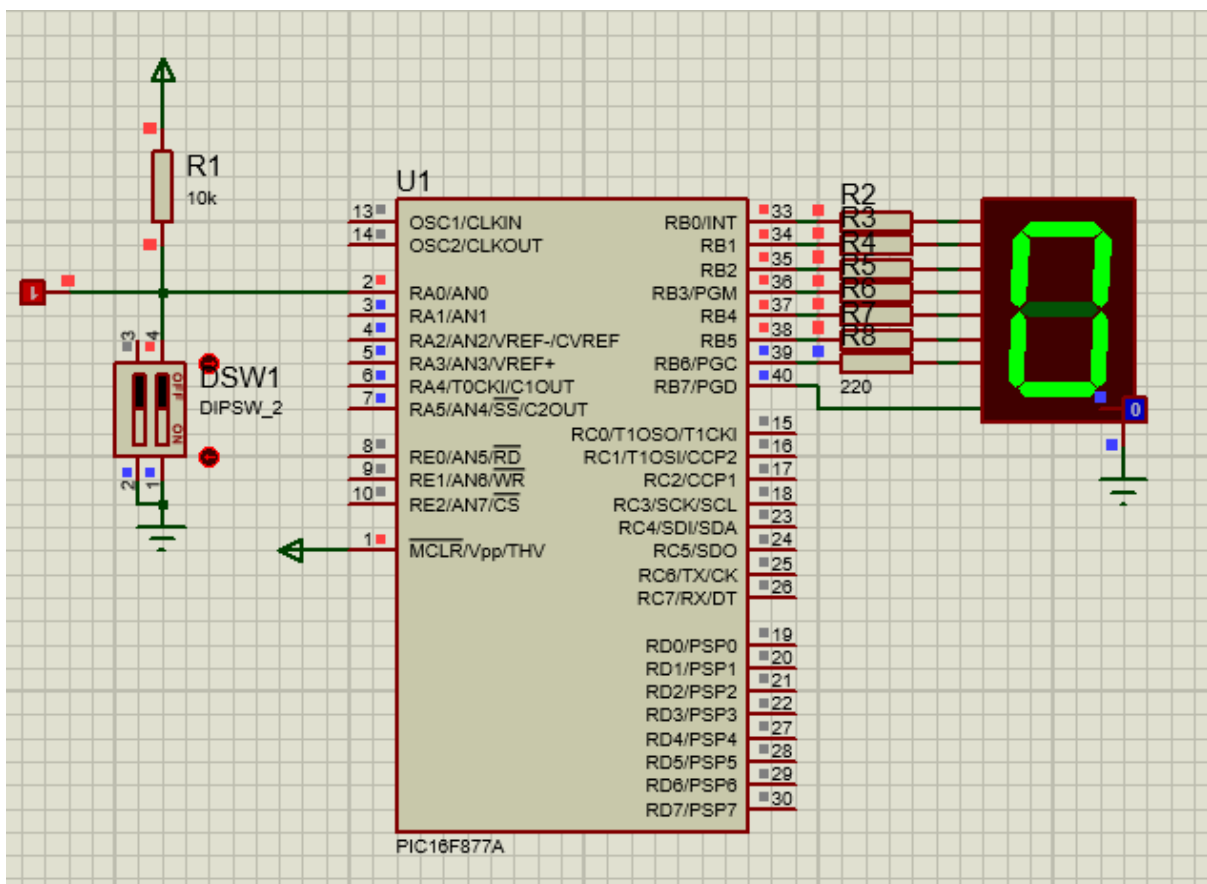
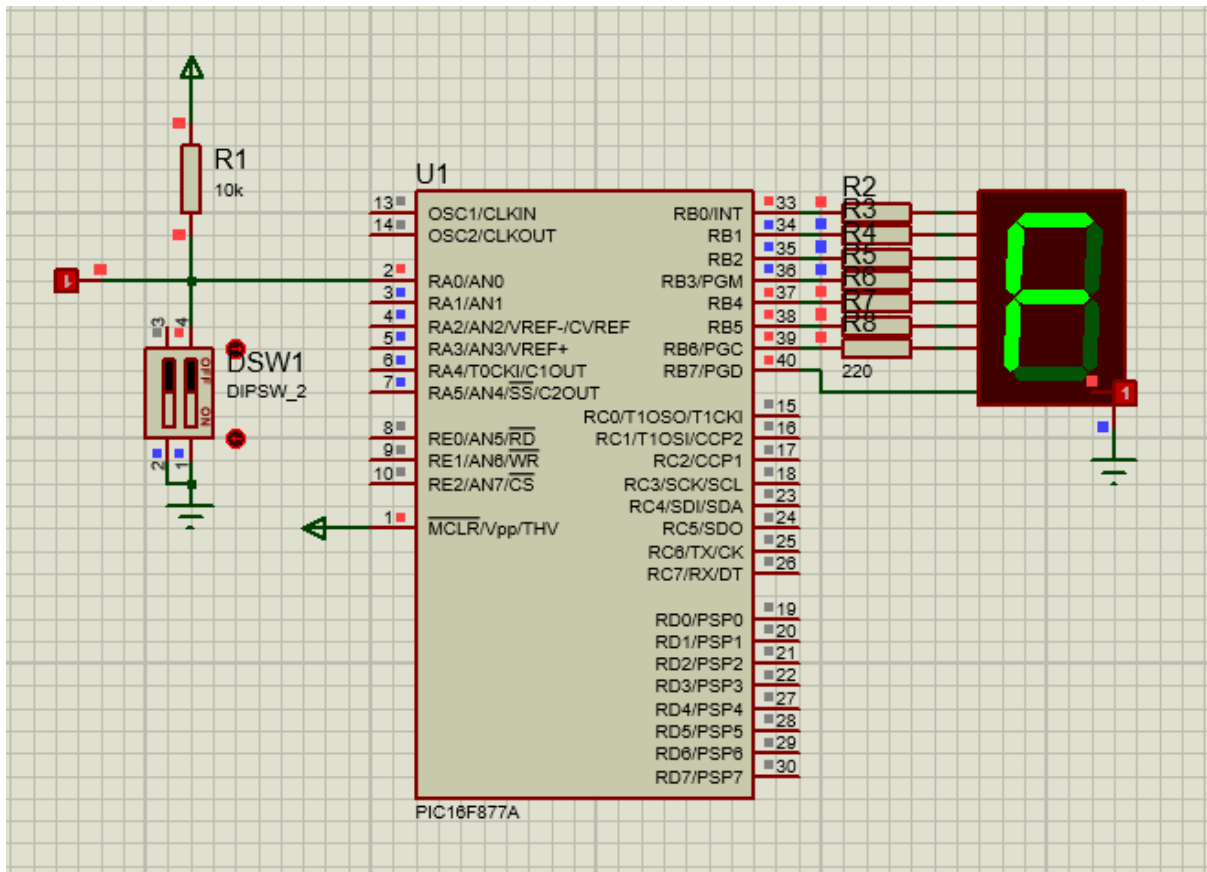
    ;4 CYCLES (INCLUDING CALL)
    RETURN

END
```

Capturas en funcionamiento:







## **Bibliografía:**

1. Microchip PIC Delay Code Generator  
*<http://www.piclist.com/techref/piclist/codegen/delay.htm>*
2. <https://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>