



Universidad Politécnica de Victoria.

**Control de un display de 7 segmentos
con pulsador (contador 0–9).**

Alumno.

Jesús Francisco Rodríguez Urdiales

Maestro.

M.C Carlos Antonio Tovar García.

21/11/2025



Contenido

Introducción	3
Diagrama de conexión.	4
Evidencias de simulación en SimulIDE.	5
Código en MpLab.	7
Evidencias del circuito real.	8
Conclusiones.	9



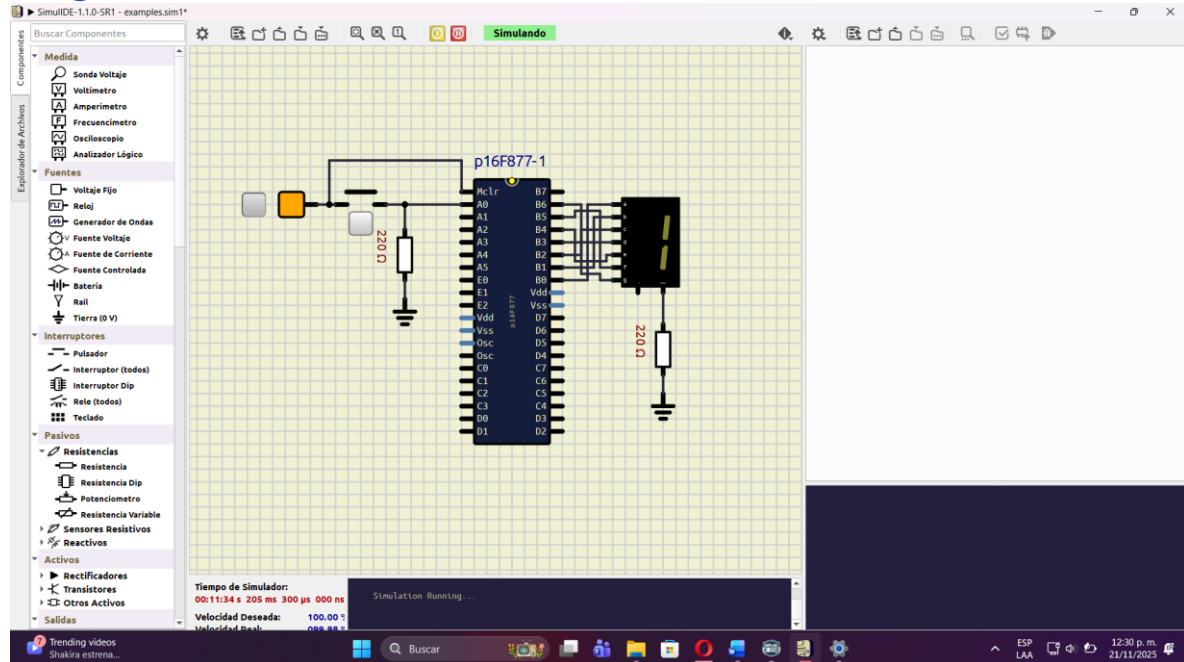
Introducción

Los displays de 7 segmentos son uno de los dispositivos de salida más utilizados en sistemas embebidos, debido a su simplicidad y a la capacidad que ofrecen para mostrar números de forma clara y directa. En esta práctica se emplea un microcontrolador PIC16F877A para controlar un display de 7 segmentos que actúa como un contador visual. Para interactuar con el sistema se utiliza un pulsador conectado como entrada digital, el cual permite avanzar el número mostrado de 0 a 9.

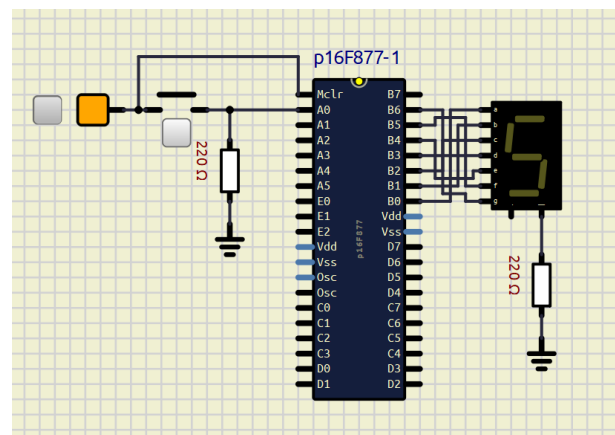
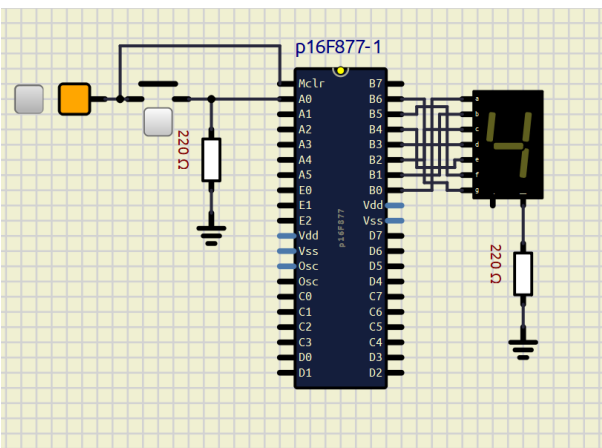
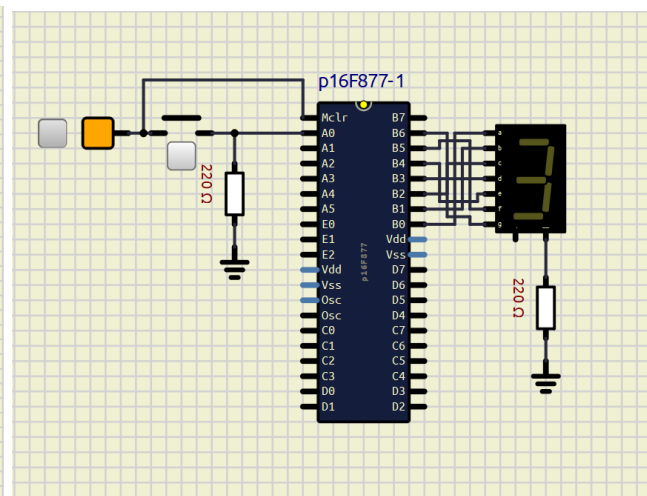
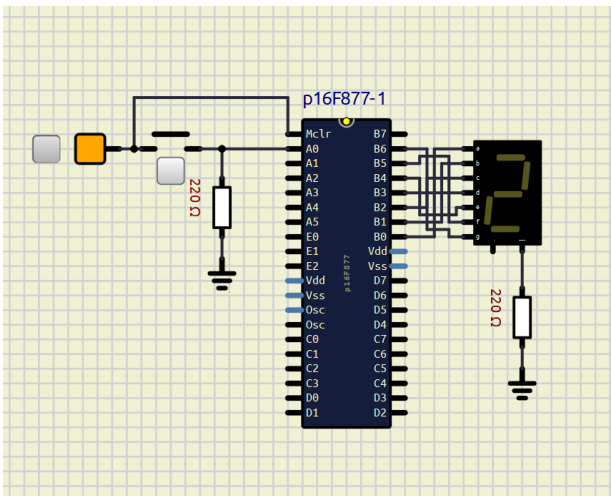
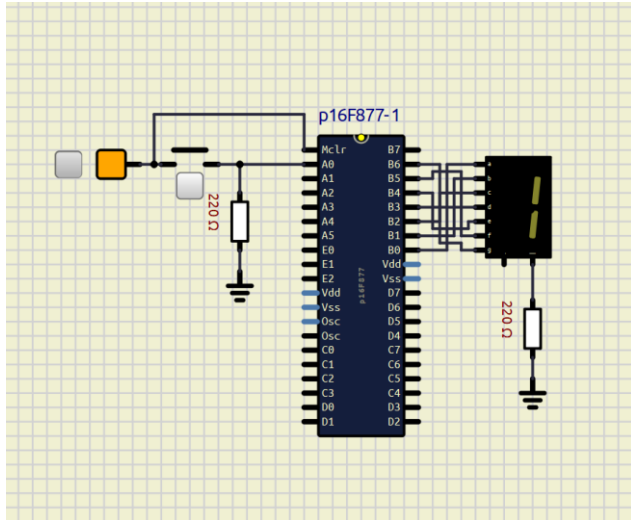
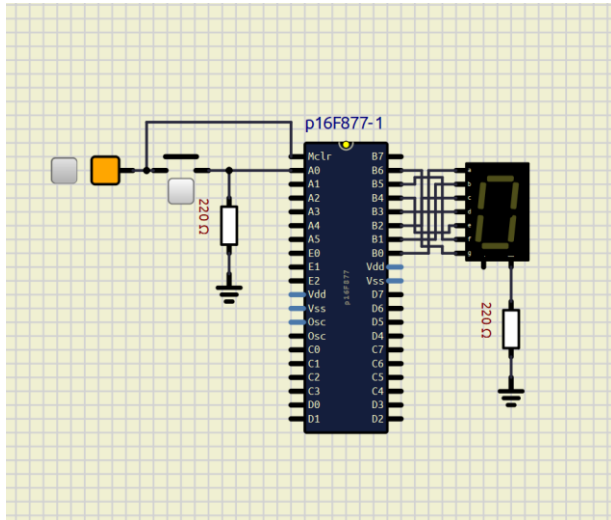
Sin embargo, los pulsadores presentan un fenómeno conocido como rebote mecánico, que provoca que una sola pulsación genere múltiples transiciones eléctricas. Esto puede causar incrementos no deseados en el contador y un funcionamiento inestable. Por esta razón, es necesario implementar una técnica de antirrebote, ya sea mediante hardware o software, que asegure que cada pulsación sea registrada únicamente una vez.

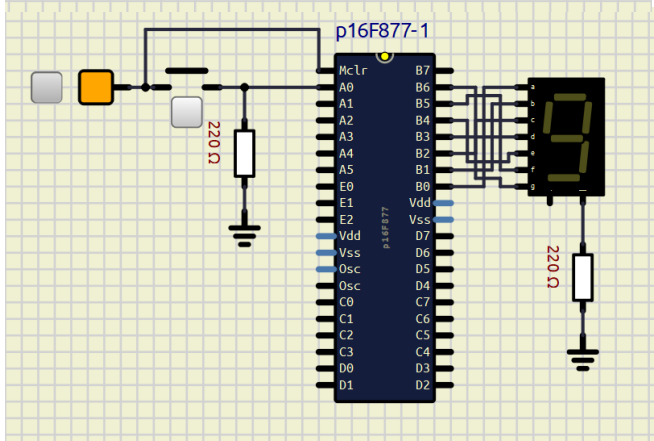
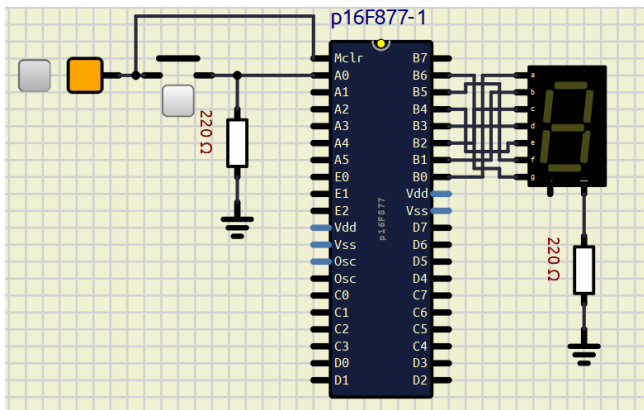
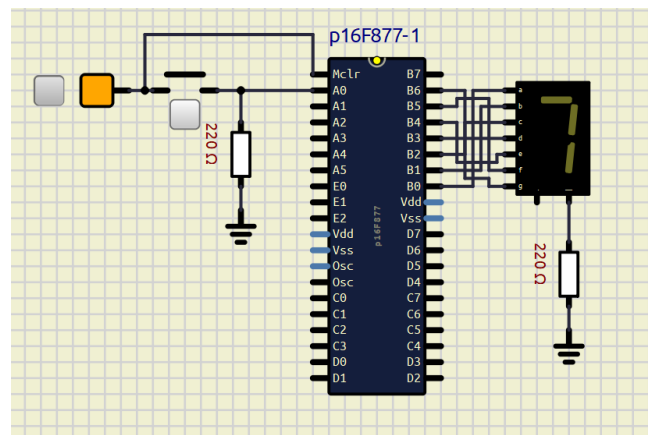
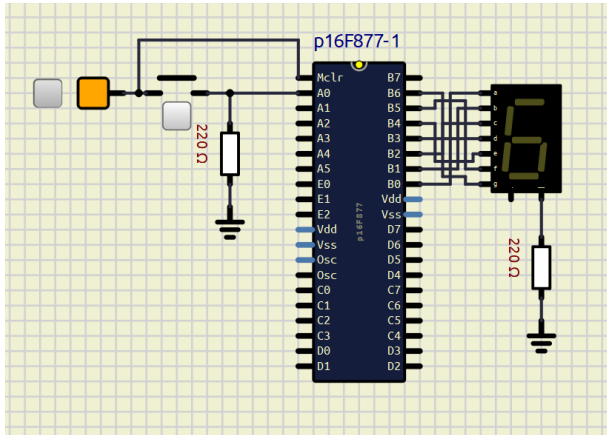
El objetivo de esta práctica es diseñar, simular y comprobar un sistema capaz de mostrar números del 0 al 9 en un display de 7 segmentos, avanzar con cada pulsación y reiniciar al llegar nuevamente a cero, garantizando un funcionamiento confiable mediante la integración del método de antirrebote. Este reporte documenta el diseño del circuito, la simulación en SimulIDE, el código implementado en MPLAB y la validación del sistema en un montaje físico.

Diagrama de conexión.



Evidencias de simulación en SimulIDE.





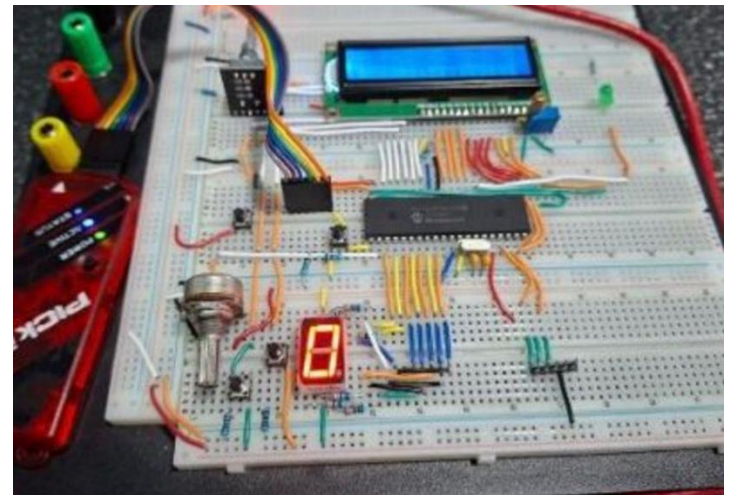
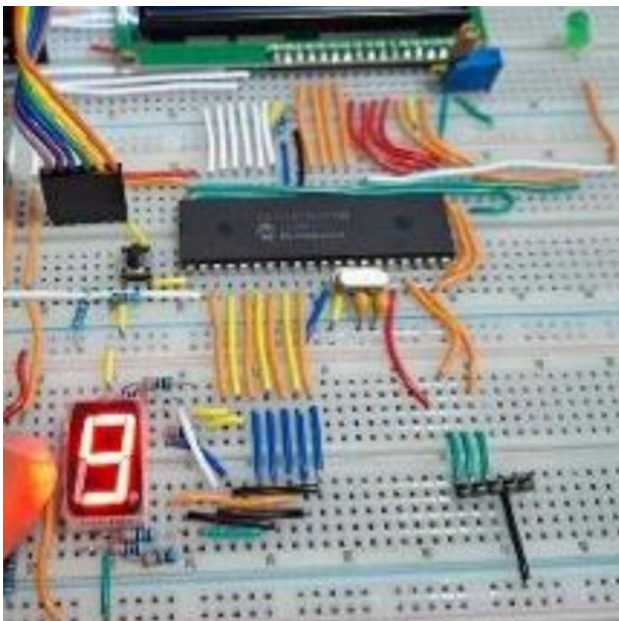
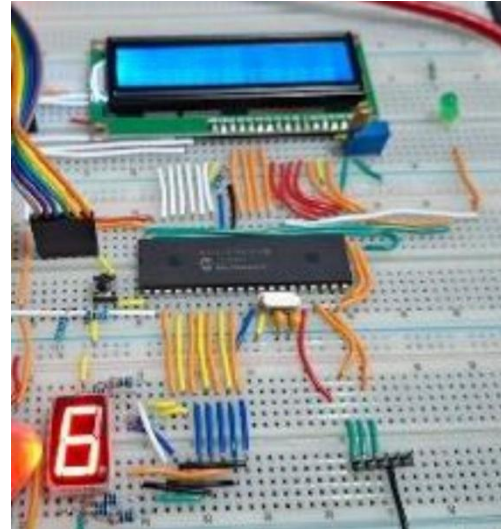
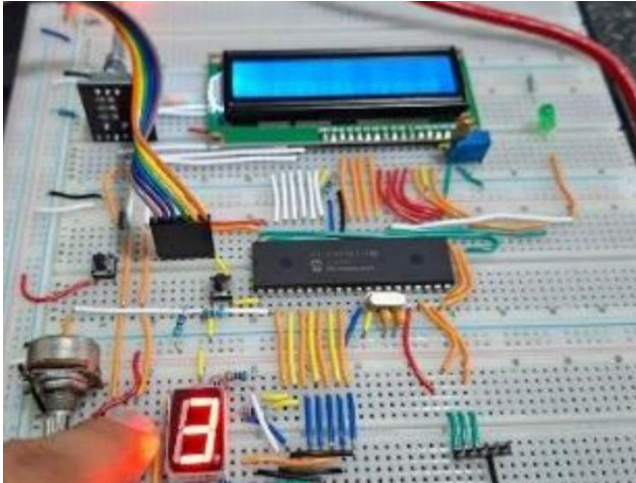
Código en Mplab.

```

Start Page x Counter.c x
Source History
1 #define _XTAL_FREQ 20000000
2 #include <xc.h>
3 #define DEBOUNCE_MS 50
4 // Antirrebote
5 unsigned char leer_boton(void){
6     if (PORTAbits.RA0 == 1){
7         _delay_ms(DEBOUNCE_MS);
8         if (PORTAbits.RA0 == 1){
9             return 1;
10        }
11    }
12    return 0;
13 }
14 void main(void) {
15     TRISB = 0x00; //b salida
16     TRISA = 0xff; //a como entrada
17     // Patrones para 0-9 en display de 7 segmentos (cátodo común)
18     const unsigned char DISPLAY[10] = {
19         0b00111111, // 0
20         0b00000110, // 1
21         0b01011011, // 2
22         0b01001111, // 3
23         0b01100110, // 4
24         0b01101101, // 5
25         0b01111101, // 6
26         0b00000111, // 7
27         0b01111111, // 8
28         0b01101111 // 9
29     };
30     unsigned char contador = 0;
31     unsigned char ultimo_estado = 0;
32     unsigned char estado_actual;
33     // Inicia mostrando 0
34     PORTB = DISPLAY[contador];
35     while(1){
36         estado_actual = leer_boton();
37         // Flanco de subida: antes estaba en 0 y ahora en 1
38         if (estado_actual == 1 && ultimo_estado == 0){
39             contador++;
40             if (contador > 9){
41                 contador = 0; // Reinicia después de 9
42             }
43             PORTB = DISPLAY[contador];
44             ultimo_estado = estado_actual;
45         }
46     }
47     return;
48 }

```


Evidencias del circuito real.





Conclusiones.

- La practica me ayudo a comprender un poco mas las funciones posibles de un microcontrolador, como se comportan sus entradas y salidas y la complejidad que puede llegar a tener. Me dio algunas ideas de las aplicaciones que tendría en una empresa o en la vida cotidiana. Es la primera vez que trabaje con un microcontrolador de tipo PIC, solo había usado el Atmega328p, pero me di cuenta que este tiene algo mas de capacidad en puertos I/O, aun que tenga menos memoria que un ATmega.
- El principal desafío que tuvimos en clase fue el manejo del rebote del contador, ya que al inicio de la práctica en clase no lo implementamos y todo el salón empezamos a tener problemas de ruido en las señales, pero después de aplicar el antirrebote, esos problemas se erradicaron completamente.
- El correcto funcionamiento de la práctica se comprobó tanto en simulación como en nuestro montaje con el PIC proporcionado por el maestro, se estuvo comprobando una y otra vez mediante pulsaciones y no se observó ninguna irregularidad.