# ARQUITECTURA DE COMPUTADORAS

## Práctica: Semáforo con Microcontrolador PIC16F877A

OBJETIVO:

Diseñar e implementar un sistema de control de semáforo utilizando un microcontrolador PIC16F877A, con el fin de comprender el uso de puertos digitales como entradas y salidas, así como la aplicación de rutinas condicionales en lenguaje C mediante MikroC.

|  |  |
| --- | --- |
| Microcontrolador 16F877A con cristal (16 MHz) y capacitores (22 pF) |  |
| Protoboard (mínimo de 400 puntos) |  |
| Resistencias de 330 Ω para los LEDs |  |
| Cables de conexión (UTP o Dupont) |  |
| Entrenadora Digital (IDL-800) |  |
| Grabadora de Microcontroladores (Master PROG) con su cable USB y el software de grabación. | Resultado de imagen de master prog mercadolibre |
| capacitor ceramico |  |
| Cristal osilador 20.000Hz |  |
| Leds |  |

INTRODUCCIÓN:

Los microcontroladores son circuitos integrados que contienen una CPU, memoria y periféricos en un solo chip, lo que los convierte en sistemas embebidos muy versátiles. Entre sus aplicaciones más comunes está el control de procesos secuenciales, como los semáforos, que requieren encender y apagar LEDs en un orden específico para simular el control de tráfico.  
  
En esta práctica se utilizó el microcontrolador PIC16F877A, configurado en MikroC, para implementar la lógica de un semáforo. Se emplearon LEDs de distintos colores, resistencias, un cristal oscilador y capacitores para el correcto funcionamiento del microcontrolador. La programación se basó en el uso de estructuras condicionales if y bucles while, que permiten cambiar entre un modo normal y un modo intermitente dependiendo de la entrada digital en el puerto RB0.

**PROCEDIMIENTO:**

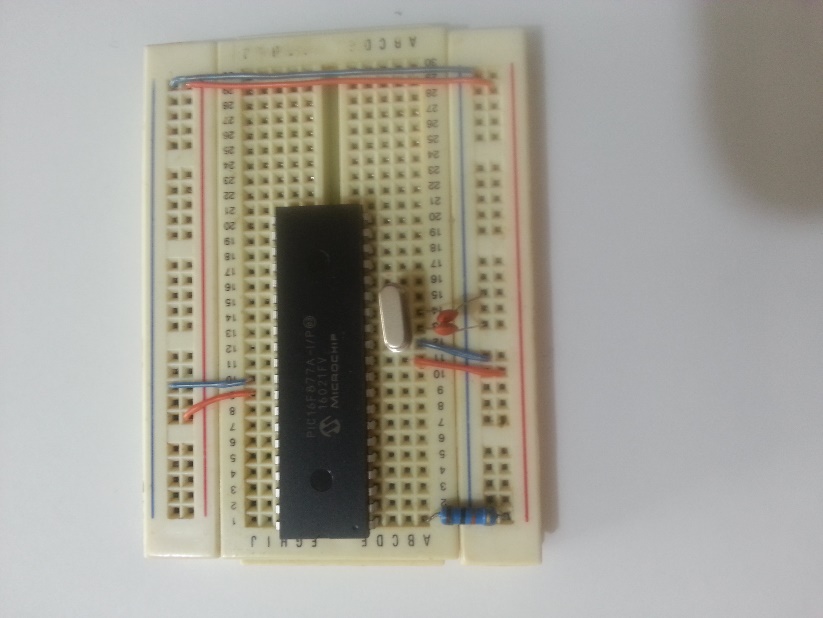
Materiales y equipo:

- Microcontrolador PIC16F877A con cristal de 16 MHz y capacitores de 22 pF.  
- Protoboard.  
- LEDs de colores (rojo, amarillo, verde).  
- Resistencias de 330 Ω para los LEDs.  
- Cables Dupont de conexión.

- Cristal osilador 20.000Hz

-Capacitor ceramic.  
- Grabadora de PIC y software MikroC.

1. Se realizó la conexión básica del PIC16F877A en protoboard: alimentación en VDD (5V), tierra en VSS, oscilador con cristal de 16 MHz y capacitores a tierra.



2. Se conectaron LEDs a los pines del Puerto B (RB1, RB2, RB3, RB5, RB6 y RB7) con sus resistencias limitadoras.  
3. Se utilizó el pin RB0 como entrada, que funciona como selector de modo:  
 - RB0 = 1 → Modo normal: Semáforo 1 en verde y Semáforo 2 en rojo.  
 - RB0 = 0 → Modo intermitente: Los LEDs cambian a parpadeo con retardos de 500 ms.  
4. Se programó el PIC en MikroC con el siguiente código:

void main() {  
 TRISB = 0b00000001; // RB0 como entrada, el resto como salidas  
 while(1){  
 if( PORTB.F0 == 1){  
 // Modo normal  
 PORTB.F7 = 1; // Semáforo 1 verde  
 PORTB.F6 = 0;  
 PORTB.F5 = 0;  
  
 PORTB.F3 = 0;  
 PORTB.F2 = 0;  
 PORTB.F1 = 1; // Semáforo 2 rojo  
 }  
 else{  
 // Modo intermitente  
 PORTB.F7 = 1;  
 PORTB.F6 = 0;  
 PORTB.F5 = 0;  
  
 PORTB.F3 = 0;  
 PORTB.F2 = 1;  
 PORTB.F1 = 0;  
 Delay\_ms(500);  
  
 PORTB.F7 = 0;  
 PORTB.F6 = 0;  
 PORTB.F5 = 0;  
  
 PORTB.F3 = 0;  
 PORTB.F2 = 0;  
 PORTB.F1 = 0;  
 Delay\_ms(500);  
 }  
 }  
}

### CONCLUSIÓN:

La práctica permitió comprender cómo utilizar los puertos de un microcontrolador PIC para el control de salidas digitales, en este caso LEDs que simulan un sistema de semáforo. Se evidenció la importancia de configurar correctamente los registros TRISB y PORTB, así como el uso de retardos con Delay\_ms() para generar el parpadeo.  
  
Este tipo de proyectos son fundamentales para entender la lógica secuencial y el control de procesos, además de ser una base para aplicaciones más avanzadas en el control de tráfico vehicular, sistemas embebidos y automatización.

### EVIDENCIA FOTOGRÁFICA:

A continuación se muestran las imágenes del montaje en protoboard realizado durante la práctica.

