Reporte. Semáforo (Modo Normal-Modo Nocturno)

Unidad Académica Multidisciplinaria Mante (UAMM)

Arquitectura de Computadoras

Montoya Garza Luis Ángel

Hernández Ruiz Haydee Michelle

Rueda Martínez Alison Michelle

Silva Sánchez Yamilka Arely

7° “E”

M.C. López Piña Daniel

Ciudad Mante, Tamaulipas. Septiembre del 2025

Introducción

En esta práctica, nos adentramos en el mundo de los sistemas embebidos mediante la implementación de un semáforo doble utilizando el microcontrolador PIC16F877A. Para ello, empleamos el IDE MikroC Pro for PIC para el desarrollo del código en lenguaje C. Una vez escrito y compilado, el programa es transferido al microcontrolador utilizando la grabadora de microcontroladores MasterPIC, lo que nos permite observar el comportamiento del circuito en la vida real.

La práctica se divide en dos fases principales. Primero, en el entorno de desarrollo MikroC Pro for PIC, escribimos el código que define la secuencia lógica de las luces del semáforo (rojo, amarillo y verde). Este código también implementa dos modos de operación, uno normal para el control de tráfico regular, y un modo nocturno que activa una luz amarilla intermitente para indicar precaución. Una vez que el código está listo y sin errores de compilación, pasamos a la fase de hardware. Aquí, utilizamos la grabadora MasterPIC para cargar el firmware en el chip PIC16F877A. Finalmente, conectamos el microcontrolador a un circuito físico con LEDs y resistencias, observando cómo el código cobra vida para simular un semáforo real.

Desarrollo

En esta práctica, se realizó la creación de un semáforo utilizando un microcontrolador. La combinación de luces para dos semáforos en una intersección simple es la siguiente:

- Semáforo 1 en Verde y Semáforo 2 en Rojo.

Durante esta fase, el tráfico de la calle 1 puede avanzar, mientras que el tráfico de la calle 2 está detenido. Esta es la fase principal de "avance" para una de las calles.

- Semáforo 1 en Amarillo y Semáforo 2 en Rojo.

Esta fase es una transición. El Semáforo 1 cambia a amarillo para indicar que pronto se detendrá el tráfico. Es crucial que el Semáforo 2 permanezca en rojo para evitar que el tráfico de la calle 2 pueda avanzar prematuramente y cause algún choque.

- Semáforo 1 en Rojo y Semáforo 2 en Verde.

Después de la transición, el Semáforo 1 se pone en rojo, y el Semáforo 2 se pone en verde. Ahora, el tráfico de la calle 2 puede avanzar y el de la calle1 está detenido.

- Semáforo 1 en Rojo y Semáforo 2 en Amarillo.

Similar a la fase 2, esta es la transición para la segunda calle. El Semáforo 2 cambia a amarillo, advirtiendo que su ciclo de "avance" está por acabar. El Semáforo 1 se queda en rojo para mantener el tráfico de la calle 1 detenido.

Además del ciclo normal, el proyecto integra un **modo nocturno**. Este modo se activa mediante un dipswitch y está diseñado para condiciones de bajo tráfico. Durante este modo, el sistema entra en una secuencia de parpadeo intermitente, en donde un semáforo muestra una luz roja fija mientras que el otro parpadea con una luz amarilla, sirviendo como una señal de precaución para los conductores.

```
/* ARQUITECTURA DE COMPUTADORAS - Ing. Daniel López Piña UAMM - UAT
* Practica 01: Semaforo Doble
* Integrantes:
            Montoya Garza Luis Angel
            Hernandez Ruiz Haydee Michelle
            Rueda Martinez Alison Michelle
            Silva Sanchez Yamilka Arely

* PIC16F877A 16Mhz
*
* Conectar todos los pines del puerto C a los Leds de la Entrenadora Digital
*/

void main() {
    // Configuramos los pines
    TRISA = 0x00;  // PORTA como salida
    TRISB = 0x00;  // PORTB como salida
    TRISD = 0B00000100; // RD2 como entrada, los demás pines son salida

    // Inicializa los pines
    PORTA = 0x00;
    PORTB = 0x00;

    while (1) {
        if(PORTD.F2==0){ //Modo normal
            // Estado 1: Semáforo 1 en ROJO, Semáforo 2 en VERDE
            PORTA = 0x04;  // A2 (Rojo)
            PORTB = 0x01;  // B0 (Verde)
            Delay_ms(5000);  // 5 segundos

            // Estado 2: Semáforo 1 en ROJO, Semáforo 2 en AMARILLO
            PORTA = 0x04;  // A2 (Rojo)
```

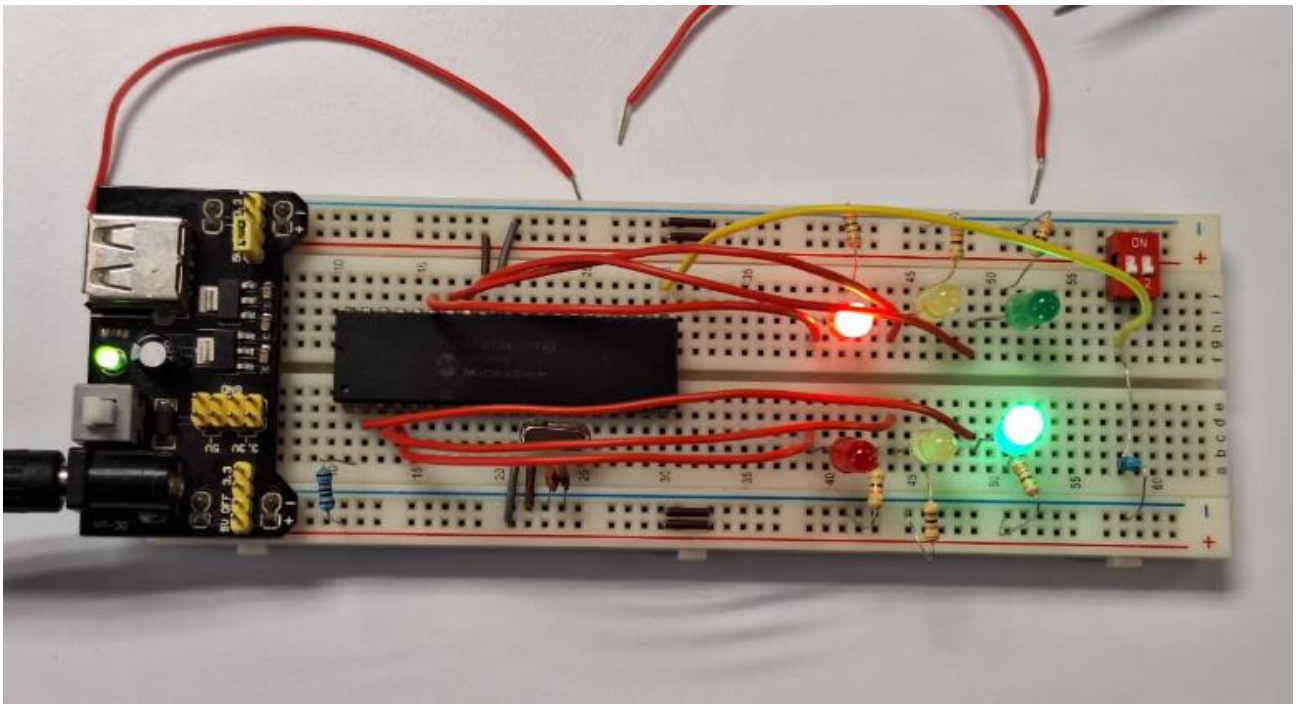
```
            PORTB = 0x02;  // B1 (Amarillo)
            Delay_ms(3000);  // 3 segundos

            // Estado 3: Semáforo 1 en VERDE, Semáforo 2 en ROJO
            PORTA = 0x01;  // A0 (Verde)
            PORTB = 0x04;  // B2 (Rojo)
            Delay_ms(5000);  // 5 segundos

            // Estado 4: Semáforo 1 en AMARILLO, Semáforo 2 en ROJO
            PORTA = 0x02;  // A1 (Amarillo)
            PORTB = 0x04;  // B2 (Rojo)
            Delay_ms(3000);  // 3 segundos
        }else{ //Modo nocturno
            // Semáforo 1 en ROJO, Semáforo 2 en AMARILLO
            PORTA = 0x04;  // A2 (Rojo)
            PORTB = 0x02;  // B1 (Amarillo)
            Delay_ms(500);  // 500 ms
            PORTA = 0x00;  // Apaga el semáforo 1
            PORTB = 0x00;  // Apaga el semáforo 2
            Delay_ms(500);  // 500 ms para parpadeo
        }
    }
}
```

Evidencia

https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.4%20Practica%20Evidencia%20Video.mp4



Conclusión

Esta práctica nos proporcionó una valiosa comprensión de los sistemas embebidos, mostrando cómo la combinación de software y hardware da vida a un proyecto funcional. Se logró la simulación de un semáforo doble, validando la metodología de desarrollo de software para microcontroladores y el uso de herramientas como MikroC Pro for PIC y la grabadora MasterPIC.

El proyecto demostró cómo un microcontrolador puede ser programado para operar en diferentes estados lógicos, como el modo normal con su secuencia completa de luces de tráfico, y un modo nocturno que activa una señal de precaución. Esta flexibilidad subraya la capacidad de los sistemas embebidos para adaptarse a distintas condiciones de uso. La experiencia reforzó la importancia de un enfoque sistemático, desde el diseño lógico del código hasta su implementación física, confirmando que la programación de sistemas embebidos es una disciplina poderosa con aplicaciones prácticas infinitas.

Perfiles GitHub

https://github.com/luisangelmg11
https://github.com/HaydeesitaSJJH
https://github.com/alisonrueda
https://github.com/Yamixjk