



## **Reporte. Práctica 01**

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. Agosto del 2025

## Introducción

En esta práctica, nos adentraremos en el mundo de los sistemas embebidos mediante la implementación de un semáforo utilizando el microcontrolador PIC 16F877A. Para ello, emplearemos el IDE MikroC Pro for PIC para el desarrollo del código en lenguaje C. Una vez escrito y compilado, el programa será transferido al microcontrolador utilizando la grabadora de microcontroladores MasterPIC, lo que nos permitirá observar el comportamiento del circuito en la vida real.

La práctica se dividirá en dos fases principales. Primero, en el entorno de desarrollo MikroC Pro for PIC, escribiremos el código que definirá la secuencia lógica de las luces del semáforo (rojo, amarillo y verde). Una vez que el código esté listo y sin errores de compilación, pasaremos a la fase de hardware. Aquí, utilizaremos la grabadora MasterPIC para cargar el *firmware* en el chip PIC 16F877A. Finalmente, conectaremos el microcontrolador a un circuito físico con LEDs y resistencias, observando cómo el código funciona para simular un semáforo real.

## Desarrollo

Para la primera parte de esta práctica, se hizo primero la conexión, se conectaron los LEDs a los pines correspondientes del Puerto C del microcontrolador PIC16F877A.

Después de las conexiones físicas, se procedió a la configuración del microcontrolador a través del código. Para ello, se utilizó la instrucción `TRISC=0;` en el programa. La instrucción `TRIS` se encarga de definir si un pin del microcontrolador funcionará como entrada o salida. Al asignar el valor de cero (0) a `TRISC`, se configuró todo el Puerto C (desde el pin `RC0` hasta el `RC7`) como salida. Esto es crucial, ya que permite al microcontrolador enviar una señal eléctrica para encender los LEDs.

Se inicializó el puerto con la instrucción `PORTC=0;`. Esta línea de código se utiliza para enviar un cero lógico a todos los pines del Puerto C, lo que asegura que los LEDs se encuentren apagados al inicio del programa. Se implementó un ciclo infinito utilizando la instrucción `while(1)`. Dentro de este ciclo, se colocó la línea `PORTC=~PORTC;`. Si el pin estaba en alto, lo cambia a bajo, y viceversa. Como esta instrucción se encuentra dentro de un ciclo infinito, el Puerto C se encenderá y apagará constantemente, lo que resulta en un parpadeo continuo de todos los LEDs conectados a ese puerto.

/\* ARQUITECTURA DE COMPUTADORAS Ing. Daniel López Piña UAMM - UAT

\* Practica 01: Salida por grupo

\* Integrantes: Montoya Garza Luis Angel, Hernández Ruíz Haydee Michelle, Rueda Martínez Alison Michelle y Silva Sánchez Yamilka Arely

\* La instrucción `TRIS` (`TRISA`, `TRISB`, `TRISC`, `TRISD`, `TRISE`) configura el puerto como entrada o salida

\* Se asigna un valor de 0 para salida y un valor de 1 para entrada

\* PIC16F877A 16Mhz

\* Conectar todos los pines del puerto C a los Leds de la Entrenadora Digital

\*/

```

void main(){

TRISC=0; //Configura todo el puerto (RC0, RC1, RC2, RC3, RC4, RC5, RC6 y RC7) C como Salida

// TRISC=0x00; //Es el equivalente a la línea anterior, pero asignándole valor Hexadecimales

// TRISC=0b00000000; //Es otro equivalente, pero asignándole valor Binario

PORTC=0; // Manda un 0 lógico (LOW) a todos los pines del puerto C. en otras palabras apaga
todo el puerto.

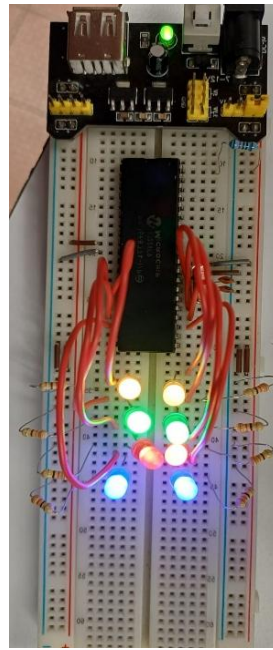
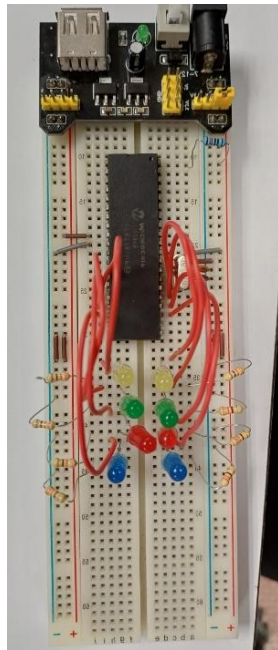
//PORTC=0x00; //Es el equivalente a la línea anterior, pero asignándole valor Hexadecimales

//PORTC=0b00000000; //Es otro equivalente, pero asignándole valor Binario

while(1){ //Inicia un ciclo infinito

PORTC=~PORTC; // "~" Invierte el valor del puerto completo: si está apagado lo enciende; si esta
encendido lo apaga.
}
}

```



Practica 1 parte 1

<https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.1.1%20Practica%20Evidencia%20Img.jpeg>

En la segunda parte de la practica, se configuran los puertos B, C, D y E como salida. Para esta parte, se modificó el código para configurar los puertos B, C, D y E del microcontrolador PIC16F877A como salida. Esto se logró mediante la instrucción TRIS. Por ejemplo, al asignar TRISB = 0; se configuró todo el Puerto B para enviar señales de salida, lo que permitió controlar los LEDs conectados a sus pines. Este mismo procedimiento se repitió para los puertos C, D y E con las líneas de código TRISC = 0;, TRISD = 0; y TRISE = 0;,. Al inicio del programa, se inicializaron todos estos puertos a un estado bajo (apagado) con las instrucciones PORTB = 0;, PORTC = 0;, PORTD = 0; y PORTE = 0; para asegurar que ningún LED estuviera encendido al arrancar.

```
/* ARQUITECTURA DE COMPUTADORAS Ing. Daniel López Piña UAMM - UAT
* Practica 01: Salida por grupo
* Integrantes: Montoya Garza Luis Angel, Hernández Ruíz Haydee Michelle, Rueda
Martínez Alison Michelle y Silva Sánchez Yamilka Arely
* Configura el puerto B, C, D y E como salida
* PIC16F877A 16Mhz
* Conectar todos los pines del puerto C a los Leds de la Entrenadora Digital
*/
```

```
void main() {
    // Configura los puertos B, C, D y E como salida
    TRISB = 0; // Configura todos los pines del puerto B como salida
    TRISC = 0; // Configura todos los pines del puerto C como salida
    TRISD = 0; // Configura todos los pines del puerto D como salida
    TRISE = 0; // Configura todos los pines del puerto E como salida

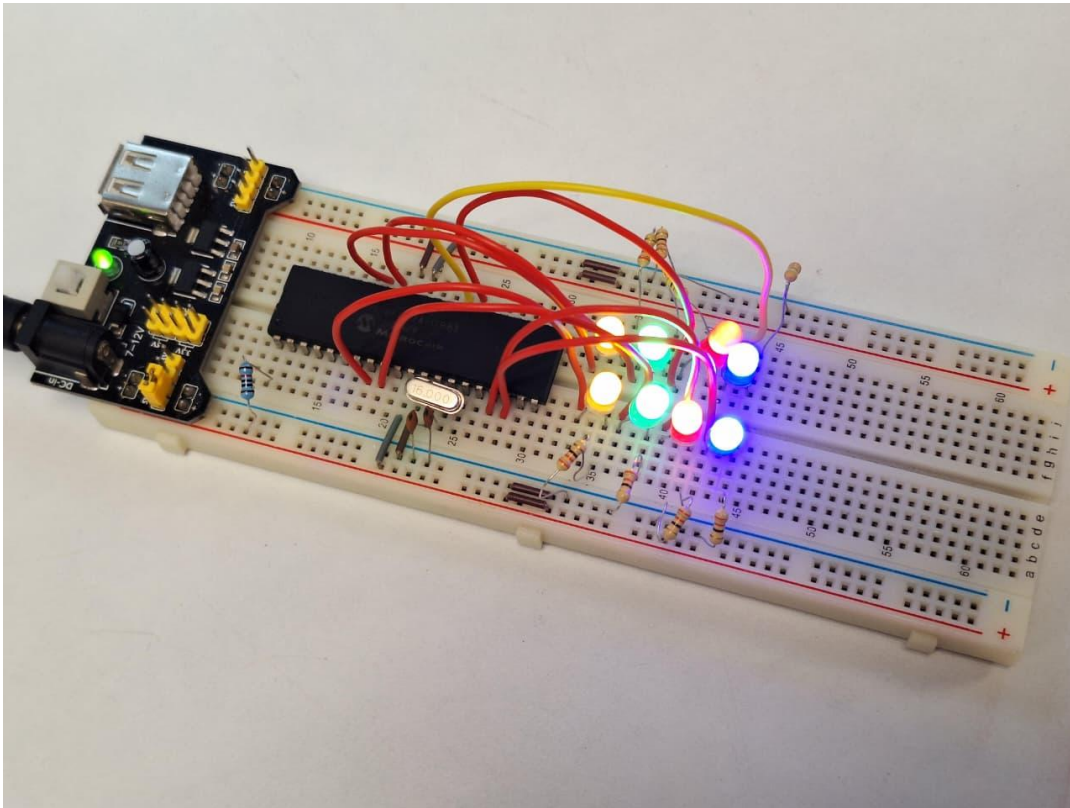
    // Inicializa los puertos B, C, D y E en 0 (apagados)
    PORTB = 0;
    PORTC = 0;
    PORTD = 0;
    PORTE = 0;

    while(1) { // Ciclo infinito
        // Enciende todos los LEDs
        PORTB = 0xFF; // Todos los pines del puerto B en alto
        PORTC = 0xFF; // Todos los pines del puerto C en alto
        PORTD = 0xFF; // Todos los pines del puerto D en alto
        PORTE = 0xFF; // Todos los pines del puerto E en alto
    }
}
```

```
    Delay_ms(500); // Espera 500ms (medio segundo)

    // Apaga todos los LEDs
    PORTB = 0x00; // Todos los pines del puerto B en bajo
    PORTC = 0x00; // Todos los pines del puerto C en bajo
    PORTD = 0x00; // Todos los pines del puerto D en bajo
    PORTE = 0x00; // Todos los pines del puerto E en bajo

    Delay_ms(500); // Espera 500ms (medio segundo)
  }
}
```



Practica 1 parte 2

<https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.1.2%20Practica%20Evidencia%20Video.mp4>

## Conclusión

Esta práctica de laboratorio nos permitió comprender y aplicar los principios básicos de la programación y control de puertos de entrada/salida (I/O) en un microcontrolador PIC16F877A. A lo largo de los ejercicios, se demostró la importancia de las instrucciones TRIS y PORT para la configuración y control de los pines del microcontrolador.

En conclusión, la práctica demostró los conceptos fundamentales de la arquitectura de microcontroladores, incluyendo la función de los registros TRIS y PORT. Se ilustró cómo, a través de la programación, se pueden manipular los puertos como entidades de 8 bits para controlar múltiples salidas de forma simultánea. El ejercicio no solo validó la capacidad del microcontrolador para interactuar con el entorno físico, sino que también subrayó la importancia de la temporización precisa en el diseño de sistemas electrónicos.

### **Perfiles GitHub**

<https://github.com/luisangelmg11>

<https://github.com/HaydeesitaSJJH>

<https://github.com/alisonrueda>

<https://github.com/Yamixjk>