



Reporte. Práctica 02

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 centraremos en comprender y aplicar la configuración de los puertos de salida del microcontrolador **PIC16F877A**, con el propósito de enviar señales lógicas a pines individuales y controlar dispositivos externos, en este caso, leds. Para lograrlo, trabajaremos con los registros **TRISx** y **PORTx**, los cuales nos permitirán definir la dirección de los pines y establecer su estado lógico respectivamente.

Durante el desarrollo de la actividad, configuraremos diferentes puertos del microcontrolador y verificaremos su funcionamiento al encender y apagar leds en secuencias específicas. Asimismo, exploraremos las distintas formas de asignar valores a los puertos mediante representaciones binarias, hexadecimales y decimales, lo que nos ayudará a comprender la equivalencia y flexibilidad de estas notaciones en la programación.

De esta manera, buscamos afianzar nuestros conocimientos sobre la manipulación de puertos de salida en un microcontrolador, entendiendo su importancia en el diseño de sistemas digitales y estableciendo una base sólida para futuras prácticas en las que se requiera el control de periféricos más complejos.

Desarrollo

En esta práctica, configuramos los puertos de salida del microcontrolador PIC16F877A para controlar leds mediante señales digitales.

- Primero, realizamos la conexión básica del microcontrolador y asignamos todos los pines del puerto C como salidas. De esta forma, pudimos enviar valores lógicos individuales y comprobar el encendido de los leds conectados.

- Después, implementamos un programa que alterna el estado de cada pin del puerto C, utilizando diferentes notaciones (binaria, hexadecimal y decimal) para comprender las equivalencias en la programación.

- Finalmente, configuramos también los puertos B, C, D y E como salidas, estableciendo una secuencia de encendido y apagado en intervalos de tiempo definidos, con lo que verificamos el funcionamiento conjunto de varios pines en distintos puertos.

Con ello reforzamos la comprensión sobre la configuración de los registros TRISx y PORTx, así como la importancia de los puertos de salida en el control de dispositivos externos.

Materiales

- Microcontrolador 16F877A con cristal (16 MHz)
- Capacitores (22 pF)
- Protoboard
- 1 Resistencia de 10kohm
- Resistencias de 330Ohms
- Leds
- Cables de conexión
- Alimentador fuente de protoboard
- Grabadora de Microcontroladores (Master PROG)

Código

(Parte 1)

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

* Practica 02: Salida individual

* Integrantes:

Montoya Garza Luis Angel

Hernández Ruíz Haydee Michelle

Rueda Martínez Alison Michelle

Silva Sánchez Yamilka Arely

* Una manera de acceder a los bit individuales es a través del selector "." seguido del identificador F0, F1, ... F7

* Siendo el F7 el bit mas significativo

* Se asigna un valor de 0 para LOW y un valor de 1 para HIGH

* PIC16F877A 16Mhz

*

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

*/

```
void main(){
```

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

```
PORTC.F0=1; // Manda un 1 lógico (HIGH) únicamente al pin RC0
```

```
//PORTC=0x01; //Es equivalente a la línea anterior, pero asignándole valor  
Hexadecimales
```

```
//PORTC=0b00000001; //Es otro equivalente, pero asignándole valor Binario
```

```
Delay_ms(5000); //Espera 5 segundos
```

```
while(1){ //Inicia un ciclo infinito
```

```
PORTC.F0=1;
```

```
PORTC.F1=0;
```

```
PORTC.F2=1;
```

```
PORTC.F3=0;
```

```
PORTC.F4=1;

PORTC.F5=0;

PORTC.F6=1;

PORTC.F7=0;

// PORTC=0b01010101; // Equivale a las 8 líneas anteriores

// PORTC=0x55; // Equivalente en Hexadecimal

// PORTC=85; // Equivalente en Decimal

Delay_ms(1000); // Espera 1 Segundo

PORTC.F0=0;

PORTC.F1=1;

PORTC.F2=0;

PORTC.F3=1;

PORTC.F4=0;

PORTC.F5=1;

PORTC.F6=0;

PORTC.F7=1;

// PORTC=0b10101010; // Equivale a las 8 líneas anteriores

// PORTC=0xAA; // Equivalente en Hexadecimal

// PORTC=170; // Equivalente en Decimal

Delay_ms(1000); // Espera un segundo

}

}
```

(Parte 2)

/* ARQUITECTURA DE COMPUTADORAS
UAMM-UAT

Ing. Daniel López Piña

Integrantes:

Montoya Garza Luis Angel

Hernández Ruíz Haydee Michelle

Rueda Martínez Alison Michelle

Silva Sánchez Yamilka Arely

* Practica 03: Secuencia de 8 salidas en varios puertos

* PIC16F877A 16Mhz

*/

```
void main() {
```

```
    ADCON1 = 0x07; // Deshabilita entradas analógicas, todos los pines como digitales
```

```
    // 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 en 0
```

```
    PORTB = 0;
```

```
    PORTC = 0;
```

```
PORTD = 0;
```

```
PORTE = 0;
```

```
while(1) { // Ciclo infinito
```

```
    // Secuencia 1: RB0=1, RB1=0, RC2=1, RC3=0, RD4=1, RD5=0, RE0=1,  
    RE1=0
```

```
    PORTB.RB0 = 1;
```

```
    PORTB.RB1 = 0;
```

```
    PORTC.RC2 = 1;
```

```
    PORTC.RC3 = 0;
```

```
    PORTD.RD4 = 1;
```

```
    PORTD.RD5 = 0;
```

```
    PORTE.RE0 = 1;
```

```
    PORTE.RE1 = 0;
```

```
    Delay_ms(2000);
```

```
    // Secuencia 2: RB0=1, RB1=1, RC2=0, RC3=0, RD4=1, RD5=1, RE0=0,  
    RE1=0
```

```
    PORTB.RB0 = 1;
```

```
    PORTB.RB1 = 1;
```

```
    PORTC.RC2 = 0;
```

```
    PORTC.RC3 = 0;
```

```
    PORTD.RD4 = 1;
```



```
PORTD.RD5 = 1;
```

```
PORTE.RE0 = 0;
```

```
PORTE.RE1 = 0;
```

```
Delay_ms(2000);
```

```
// Secuencia 3: RB0=0, RB1=1, RC2=1, RC3=1, RD4=0, RD5=1, RE0=1,  
RE1=1
```

```
PORTB.RB0 = 0;
```

```
PORTB.RB1 = 1;
```

```
PORTC.RC2 = 1;
```

```
PORTC.RC3 = 1;
```

```
PORTD.RD4 = 0;
```

```
PORTD.RD5 = 1;
```

```
PORTE.RE0 = 1;
```

```
PORTE.RE1 = 1;
```

```
Delay_ms(2000);
```

```
// Secuencia 4: RB0=1, RB1=1, RC2=1, RC3=1, RD4=0, RD5=0, RE0=0,  
RE1=0
```

```
PORTB.RB0 = 1;
```

```
PORTB.RB1 = 1;
```

```
PORTC.RC2 = 1;
```

```
PORTC.RC3 = 1;
```

```
PORTD.RD4 = 0;
```

```
PORTD.RD5 = 0;
```

```
PORTE.RE0 = 0;
```

```
PORTE.RE1 = 0;
```

```
Delay_ms(2000);
```

```
// Secuencia 5: RB0=0, RB1=1, RC2=0, RC3=1, RD4=0, RD5=1, RE0=0,  
RE1=1
```

```
PORTB.RB0 = 0;
```

```
PORTB.RB1 = 1;
```

```
PORTC.RC2 = 0;
```

```
PORTC.RC3 = 1;
```

```
PORTD.RD4 = 0;
```

```
PORTD.RD5 = 1;
```

```
PORTE.RE0 = 0;
```

```
PORTE.RE1 = 1;
```

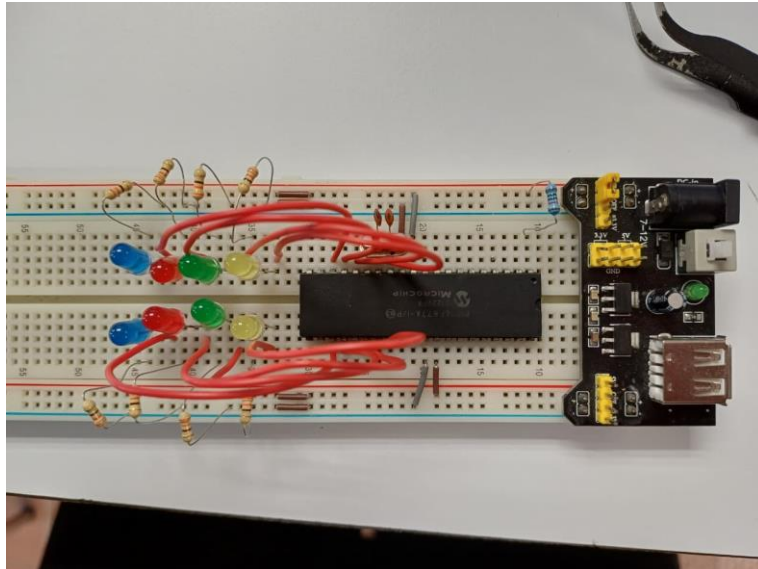
```
Delay_ms(2000);
```

```
}
```

```
}
```

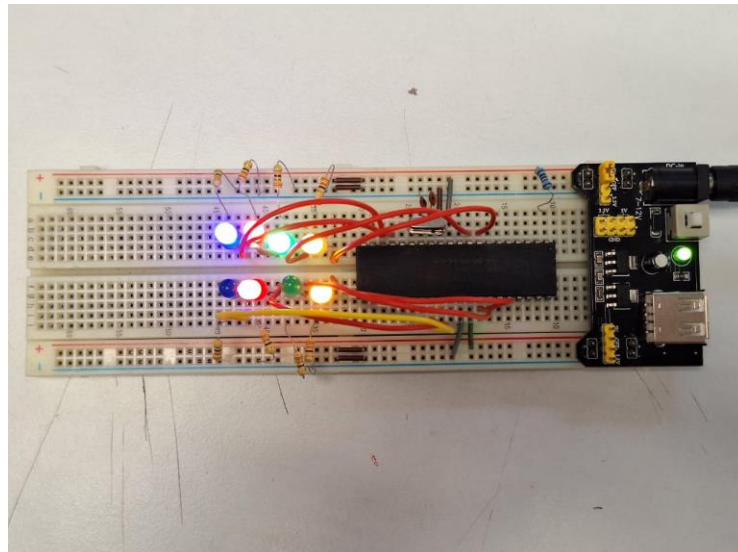
Evidencia

(Parte 1)



<https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.2.1%20Practica%20Evidencia%20Video.mp4>

(Parte 2)



<https://github.com/luisangelmg11/Arquitectura-de-Computadoras/blob/main/1.2.2%20Practica%20Evidencia%20Video.mp4>

Conclusión

En esta práctica logramos configurar y utilizar los puertos de salida del microcontrolador PIC16F877A, comprobando su funcionamiento a través del encendido y apagado de leds en diferentes secuencias. A lo largo del trabajo comprendimos cómo, mediante la correcta configuración de los registros TRISx y PORTx, es posible definir la dirección y el estado lógico de cada pin, lo que nos permitió controlar los leds de forma individual y en conjunto. Asimismo, reforzamos nuestro entendimiento sobre el uso de distintas formas de asignación binaria, decimal y hexadecimal y comprobamos que todas conducen al mismo resultado, aunque brindan distintas ventajas en la programación. Finalmente, confirmamos que la configuración simultánea de varios puertos facilita la creación de secuencias más complejas y nos prepara para futuros proyectos en los que será necesario aplicar estos conocimientos en el control de periféricos y sistemas embebidos más avanzados.

Perfiles GitHub

<https://github.com/luisangelmg11>

<https://github.com/HaydeesitaSJJH>

<https://github.com/alisonrueda>

<https://github.com/Yamixjk>