



Unidad Académica Multidisciplinaria
Mante



UNIVERSIDAD AUTONOMA DE TAMAULIPAS
UNIDAD ACADEMICA MULTIDISCIPLINARIA MANTE
ARQUITECTURA DE COMPUTADORAS



PRACTICA 2

DOCENTE: DANIEL LOPEZ PIÑA

INTEGRANTES: JUAN DE DIOS ZAPATA ECHEVERRIA

ERICK GABRIEL VILLAFUERTE DE LARA

ANGEL FERNANDO RIOS PEREZ

JEAN BRANDON HERNANDEZ DE LA CRUZ

ANA ISABELA ZÚÑIGA GONZÁLEZ

ARMANDO DE JESÚS AVALOS CASTILLO



Contenido

Introducción	2
Procedimiento	3
1. Conexión básica del microcontrolador:	3
2. Materiales:	4
3. Creación del proyecto en MikroC:	5
3. Compilación del programa:.....	6
4. Grabación del programa en el PIC:.....	7
Conclusión	8

Introducción

Un microcontrolador posee puertos de entrada salida o mejor dicho pines de entrada / salida (I/O) de propósito general, mediante ellos el PIC puede monitorear y controlar otros dispositivos. Por consiguiente, es necesario configurarlos apropiadamente. Para este microcontrolador el puerto A tiene funciones analógicas que en prácticas posteriores se revisara a detalle su configuración.

Procedimiento

1. Conexión básica del microcontrolador:

Se identificaron los pines principales del PIC16F877A consultando su hoja de datos (DATASHEET)

Se realizaron las conexiones mínimas:

Alimentación

- **Pin 11 (VDD)** → Conectar a **+5V**.
- **Pin 32 (VDD)** → Conectar también a **+5V**.
- **Pin 12 (VSS)** → Conectar a **GND**.
- **Pin 31 (VSS)** → Conectar también a **GND**.

Reset (MCLR)

- **Pin 1 (MCLR/VPP)** → Conectar a **+5V** mediante una resistencia de **10 kΩ**.
- También puede añadirse un **push button a GND** para reiniciar el microcontrolador manualmente.

Oscilador externo (reloj)

- **Pin 13 (OSC1/CLKI)** → Conectar un **cristal de 4, 8 o 16 MHz**.
- **Pin 14 (OSC2/CLKO)** → Conectar el otro terminal del cristal.
- Desde cada terminal del cristal, colocar un **capacitor de 22 pF** a tierra (GND).

Entradas/Salidas digitales

- Los pines de los **puertos A, B, C, D y E** se pueden usar como entradas o salidas, configurándolos en el programa con los registros TRIS.
- el **Puerto C (pines 15 a 18 y 23 a 26)** para conectar LEDs con resistencias limitadoras (330–470 Ω).

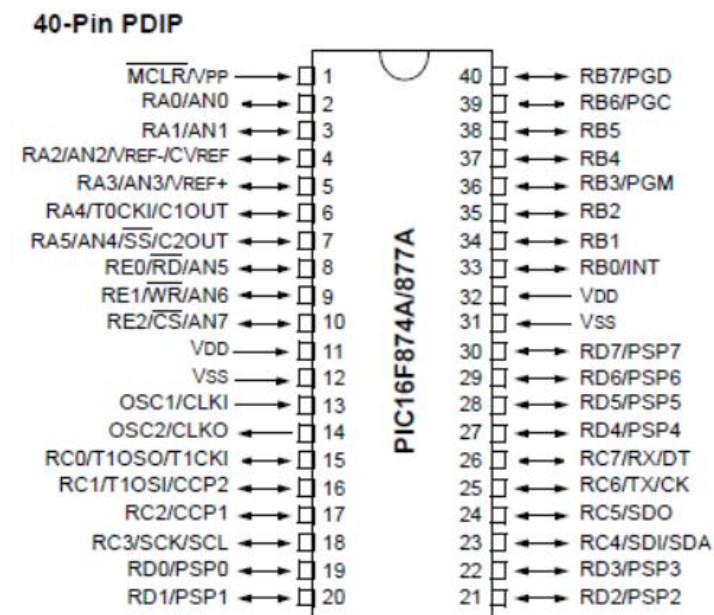
□ Programación

- Los pines **RB6 (PGC)** y **RB7 (PGD)** se utilizan para programar el microcontrolador con un programador externo (ej. PICKit).

2. Materiales:

- Microcontrolador PIC16F877A
- Cristal de 16 MHz con dos capacitores de 22 pF
- Resistencia de 10 k Ω para pin MCLR
- 8 LEDs con resistencias limitadoras de 330–470 Ω
- Protoboard de 400 puntos mínimo
- Cables de conexión
- Programador PIC (ej. PICkit 2/3)
- Computadora con MikroC instalado
- Grabadora de Microcontroladores (Master Prog)

Se uso el DATASHEET siguiente:



3. Creación del proyecto en MikroC:

- Se abrió el entorno de MikroC.
- Se creó un nuevo proyecto especificando nombre, carpeta, dispositivo (PIC16F877A) y frecuencia de reloj (16 MHz).

Se introdujo el siguiente Código:

```
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
```

```

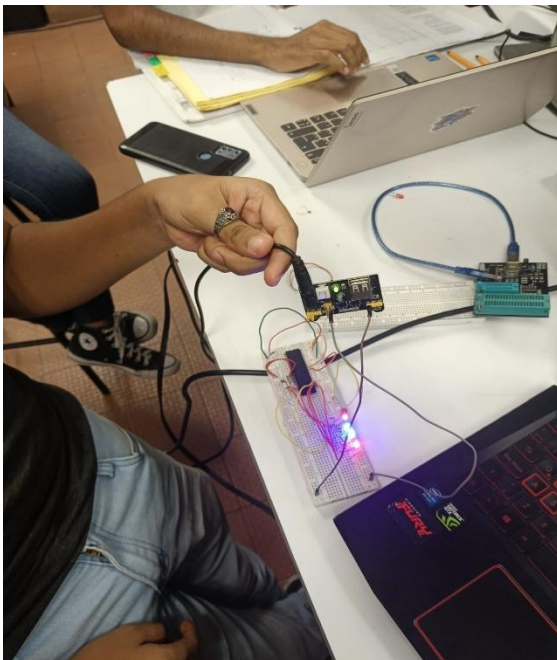
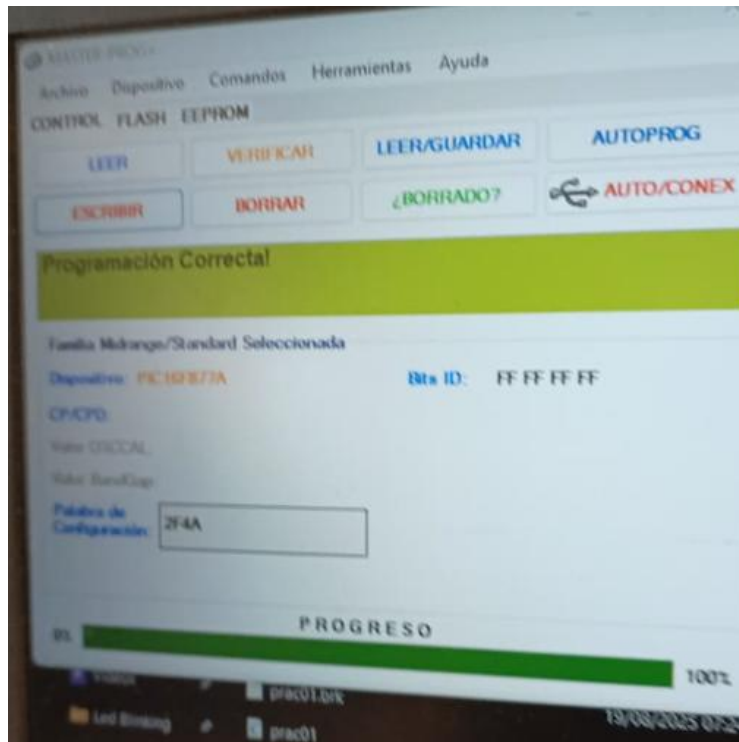
// PORTC=0x55;      // Equivalente en Hexadecimal
// PORTC=85;        // Equivalente en Decimal
Delay_ms(1000);     // Espera 1 Segundo
PORTC.F0=0;
PORTC.F1=1;
PORTC.F2=0;
PORTC.F3=1;
PORTC.F4=0;
PORTC.F5=1;
PORTC.F6=0;
PORTC.F7=1;

// ; // Equivale a las 8 líneas anteriores
// PORTC=0xAA;      // Equivalente en Hexadecimal
// PORTC=170;       // Equivalente en Decimal
Delay_ms(1000);     // Espera un segundo
}}

```

3. Compilación del programa:

Se compiló el proyecto sin errores y se generó un archivo .HEX.



4. Grabación del programa en el PIC:

Se utilizó la grabadora Master-PROG conectada vía USB, Después se colocó el microcontrolador en la base ZIF, verificando la orientación del pin 1 y finalmente se cargó el archivo .HEX desde el software y se grabó en el dispositivo.

Se utilizó la grabadora Master-PROG conectada vía USB, Después se colocó el microcontrolador en la base ZIF, verificando la orientación del pin 1 y finalmente se cargó el archivo .HEX desde el software y se grabó en el dispositivo.

Como ultimo, se instaló el microcontrolador en la protoboard y se conectó el puerto C a los LEDs de la entrenadora digital.

Conclusión

La práctica permitió comprobar el uso de los puertos de entrada/salida del microcontrolador, en específico el Puerto C del PIC16F877A, configurado como salida digital. Mediante la programación se logró controlar el encendido y apagado de LEDs en un patrón alternado, lo que evidenció el correcto manejo de los registros TRIS y PORT. Con ello se reforzó la importancia de configurar adecuadamente los pines según su función, ya que de esta manera el microcontrolador puede interactuar eficazmente con otros dispositivos. Este conocimiento constituye la base para prácticas posteriores en las que se abordará la configuración de puertos con funciones especiales, como el Puerto A en modo analógico.