M16

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Semestre 2023-2

Laboratorio de Microcomputadoras

Proyecto 2:

Puertos paralelos

Profesor:

M.I. Ruben Anaya García

Alumnos:	Número de cuenta:
Barreiro Valdez Alejandro	317520888
Gil Márquez Arath Emiliano	317083875
Herrera Carrillo Cristhian	317094662
Zepeda Baeza Jessica	317520747

Grupo Teoría: 1

Fecha de realización: 27 de marzo de 2023

Fecha de entrega: 1 de mayo de 2023

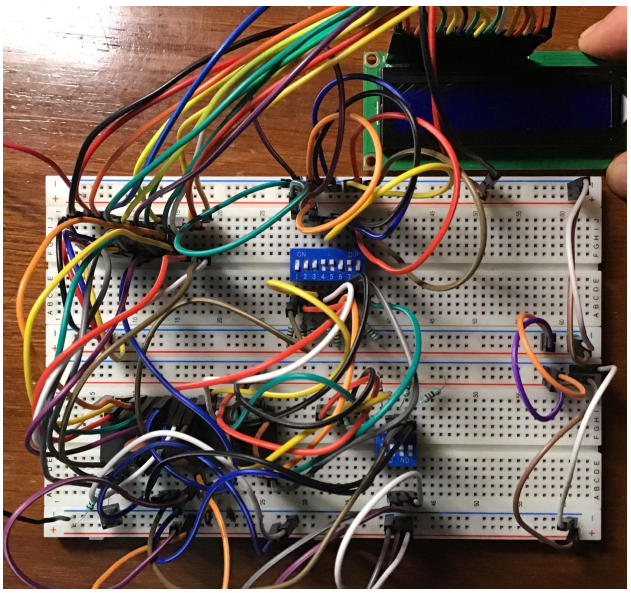
Introducción

Un puerto paralelo es una interfaz de comunicación que permite la transferencia de datos entre un ordenador y un dispositivo periférico. A diferencia de los puertos serie, que transfieren los datos bit por bit, los puertos paralelos transfieren múltiples bits al mismo tiempo, lo que los hace ideales para enviar grandes cantidades de datos en poco tiempo. El puerto paralelo más comúnmente utilizado es el puerto paralelo de la impresora, también conocido como el puerto LPT (Line Printer Terminal). Este puerto se encuentra en la mayoría de las computadoras personales y se utiliza para conectar una impresora a la computadora.

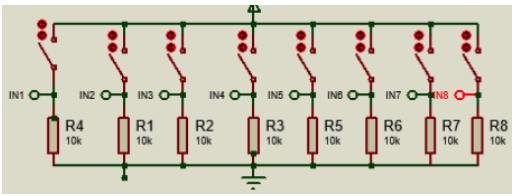
Además de las impresoras, los puertos paralelos también se utilizan para conectar otros dispositivos periféricos, como escáneres, cámaras digitales, unidades de disco duro externas y dispositivos de control de instrumentos. Los puertos paralelos se han utilizado durante décadas en la informática, pero han sido reemplazados en gran medida por puertos USB más rápidos y versátiles. Sin embargo, algunos dispositivos periféricos todavía utilizan puertos paralelos debido a su capacidad de transferir grandes cantidades de datos en poco tiempo y su simplicidad de diseño. En este proyecto se utilizarán los puertos paralelos para recibir datos de entrada a través de dos *switches* y generar una salida a través de un *display* de cristal líquido.

Desarrollo

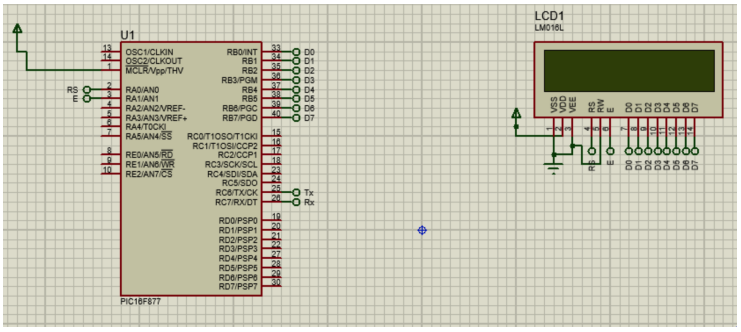
Se agregaron al sistema mínimo realizado en el proyecto anterior dos entradas y una salida. Las dos entradas que se agregaron fueron dos *switches*. El primero, el que se encuentra en la parte posterior de la imagen fue utilizado para cambiar el modo de la salida. El segundo *switch*, el que se encuentra en la parte superior de la imagen, fue utilizado para recibir un número que serviría para mostrarlo como salida dependiendo el modo. Los modos que se programaron fueron: número en decimal, número en binario, número en hexadecimal, símbolo creado por cada uno y nombres del equipo. Estos modos se mostraron en la salida que fue un display de cristal líquido. Este componente también se tuvo que alambrar y se muestra en la parte superior derecha de la imagen. El resto del sistema se mantuvo como se realizó en el proyecto anterior. Lo único que cambió fue que para este proyecto se desconectaron los LEDs que se utilizaban de salida en el proyecto anterior.



Para las entradas de los *switches* se utilizaron resistencias, conexiones al voltaje y conexiones a tierra. Además, se utilizó un *Dip Switch* de ocho posiciones. Se conectó la entrada entre el *switch* y las resistencias. Se utilizó el siguiente diagrama para alambrar esta entrada.



Para la salida LCD se utilizó el siguiente diagrama para alambrar cada uno de los pines del LCD con una salida del microcontrolador. Además, se conectó el LCD a voltaje y tierra. El LCD que se utilizó también tenía un pin de cátodo y de ánodo. El de ánodo se conectó a voltaje y el de cátodo a tierra.

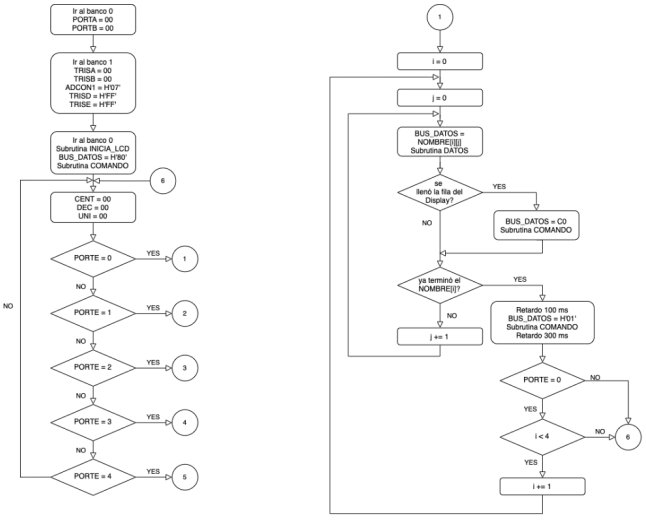


Algoritmos

Para este proyecto se realizó una carta ASM que se divide en 6:

1. El programa principal
2. La opción 0 del Switch de 3 bits: Nombres de los integrantes
3. La opción 1 del Switch de 3 bits: Número decimal
4. La opción 2 del Switch de 3 bits: Número hexadecimal
5. La opción 3 del Switch de 3 bits: Número binario
6. La opción 4 del Switch de 3 bits: Carácteres especiales

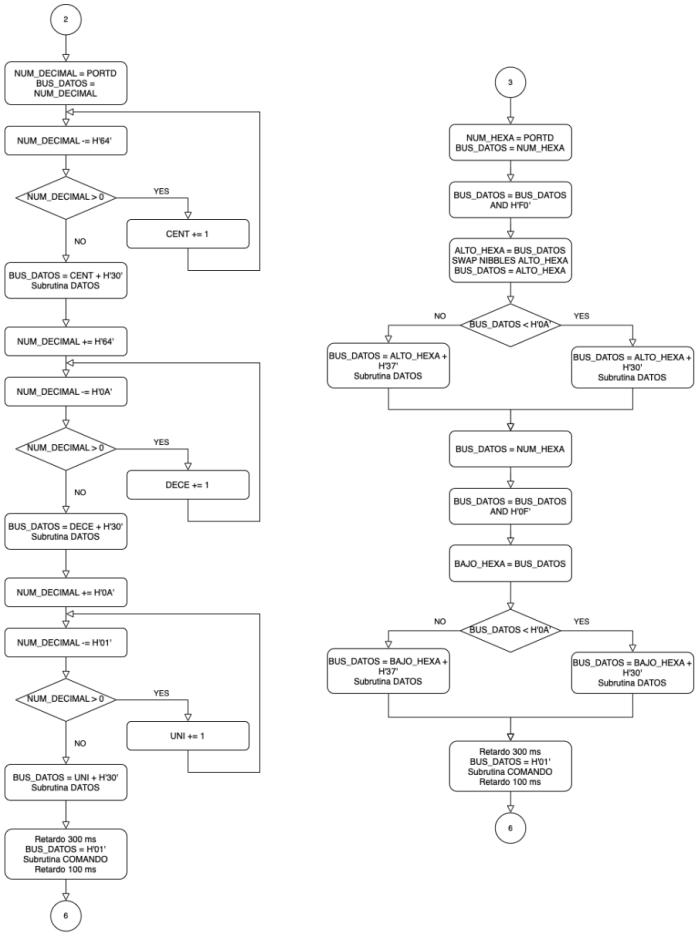
En la carta del programa principal se hacen las configuraciones de los puertos paralelos, definiendo el A y B como salidas y D y E como entrada. También se configura a los puertos A y E como digitales y se realiza la subrutina de inicialización del LCD. Por último, se verifica el valor del Switch de 3 bits, comparándolo con 0, 1, 2, 3, y 4. Cuando se encuentra su valor se hace un salto a otra parte del código que implementa la acción a realizar. Una vez que se realiza o cuando el valor no corresponde a las opciones indicadas, regresa a verificar el valor del Switch.



La opción 0 muestra los nombres de cada integrante del equipo. Para ello guarda en el bus de datos el valor ASCII de cada letra del nombre y llama la subrutina DATOS. Cuando la siguiente palabra ya no cabe en la primera línea del Display, se manda un C0 al bus de datos y se llama la subrutina COMANDO. Una vez que se tiene el nombre completo, se llama un retardo de 100ms, luego se manda al bus de datos 01 para llamar la subrutina COMANDO y por último se hace un retardo de 300ms. Después se verifica que el valor del Switch siga siendo 0, si es así se realiza el mismo proceso para el segundo nombre y luego para el tercero y cuarto. Si el valor del Switch cambia o cuando el proceso del cuarto nombre acaba, se regresa a verificar el valor del Switch en el programa principal.

La opción 1 muestra el número ingresado en el Switch de 8 bits en decimal. Primero guarda el valor del Switch en una variable. Para encontrar las centenas le resta 100 (64 hexadecimal) a la variable hasta que dicha variable sea negativa, incrementando en 1 la variable CENT cada vez que hace la resta y no se ha llegado al valor negativo. Una vez que se obtiene el valor negativo, se le suma H’30’ a la variable CENT para obtener el valor ASCII del número en la variable y se manda al bus de datos para llamar la subrutina DATOS y que se muestre en el Display.

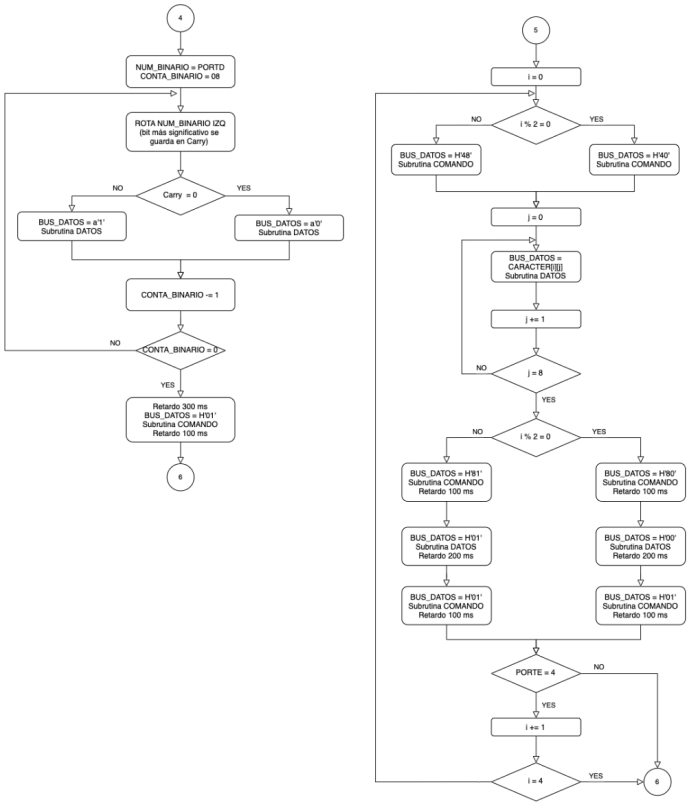
Después se le suma 100 a la variable para regresar al valor positivo y ahora se repite el proceso pero para encontrar las decenas. Para ello, se le resta de 10 en 10 (0A hexadecimal) y el contador de las restas se guarda en DECE. De igual forma, al llegar al valor negativo, se obtiene el ASCII del número en DECE y se manda al Display. Por último se repite el proceso para las unidades, restando de 1 en 1 y guardando el contador en UNI. Cuando se llega al valor negativo se encuentra su ASCII y se manda al Display. Por último se llama un retardo de 300ms, luego se manda al bus de datos 01 para llamar la subrutina COMANDO y por último se hace un retardo de 100ms. Finalmente regresa al programa principal a verificar el valor del Switch.



La opción 2 muestra el número ingresado en el Switch de 8 bits en hexadecimal. Primero guarda el valor del Switch en una variable y en el bus de datos. Se hace un AND entre el bus de datos y F0 para obtener solo los primeros 4 bits del número ingresado. Y se pasan a otra variable ALTO_HEXA y para hacer un swap de nibbles y pasar el valor a su parte baja. Se manda al bus de datos y se verifica si este valor es menor a A hexadecimal para obtener su valor en ASCII correcto. Si es mayor o igual se le suma H'37' al bus de datos (para una letra ASCII) y si es menor se le suma H'30' (para un número ASCII). Después se llama la subrutina DATOS.

Después se vuelve a pasar el valor del número ingresado completo al bus de datos y se hace un AND con 0F para ahora obtener la mitad baja del número. Se verifica si es menor a A y de igual forma se suma H'30' si sí lo es y H'37' si es mayor o igual. Después se llama la subrutina DATOS para mandarlo al Display. Por último, se llama un retardo de 300ms, luego se manda al bus de datos 01 para llamar la subrutina COMANDO y por último se hace un retardo de 100ms. Finalmente regresa al programa principal a verificar el valor del Switch.

La opción 3 muestra el número ingresado en el Switch de 8 bits en binario. Primero guarda el valor del Switch en una variable e inicializa un contador en 08. Se hace un ciclo donde se hace un corrimiento a la izquierda de la variable, lo que va a ocasionar que el bit más significativo se almacene en Carry. Después se analiza si Carry es 0 porque en ese caso se manda al bus de datos el ASCII de 0 y en caso contrario se manda el ASCII de 1. Después se llama la subrutina DATOS y se decrementa en 1 el contador. En caso de que el contador no sea cero aún se repite el ciclo. Cuando llega a 0 se llama un retardo de 300ms, luego se manda al bus de datos 01 para llamar la subrutina COMANDO y por último se hace un retardo de 100ms. Finalmente regresa al programa principal a verificar el valor del Switch.



La opción 4 muestra cuatro caracteres especiales en el Display, uno a la vez. Primero se manda al bus de datos el valor H’40’ (para caracteres 0 y 2) y H’48’ (para caracteres 1 y 3) para llamar la subrutina COMANDO, lo que esto hace es activar la CGRAM. Para cada carácter se mandan 8 valores binarios de 8 bits y después de cada valor se llama la subrutina DATOS. Después se manda al bus de datos el valor de H’80’ (para caracteres 0 y 2) o H’81’ (para caracteres 1 y 3) y se llama a la subrutina COMANDO y a un retardo de 100ms; esto posiciona al LCD en la primera y segunda posición respectivamente. Después se manda al bus de datos H’00’ (para caracteres 0 y 2) o H’01’ (para caracteres 1 y 3) para llamar a la subrutina DATOS y a un retardo de 200ms y ver el caracter en el Display. Lo último que se manda al bus de datos es H’01’ para llamar la subrutina COMANDO y a un retardo de 100ms para limpiar la pantalla. Finalmente se verifica si la opción en el Switch sigue siendo 4, si es así se pasa a mostrar el siguiente caracter repitiendo el proceso anterior. Si no es así o cuando el último caracter se muestre, el flujo regresa al programa principal a verificar el valor del Switch.

Programa comentado

Este programa por medio de la implementación de un sistema mínimo con un microcontrolador PIC16F877A, realiza varias acciones dependiendo de los valores de entrada que reciba de los diferentes Dip Switches conectados a los puertos del microcontrolador, pudiendo realizar conversiones de números decimales, binarios y hexadecimales, que con la ayuda de un LCD configurado por medio de comandos es capaz de mostrar los resultados con los respectivos caracteres existentes, así como personalizados.

```
PROCESSOR 16f877    ;Procesador a utilizar
INCLUDE <p16f877.INC>

VALOR equ h'20'
VALOR1 equ h'21'
VALOR2 equ h'22'

CENT EQU h'23'
DECE EQU h'24'
UNI EQU h'25'
NUM_DECIMAL EQU h'26'

NUM_HEXA EQU h'27'
ALTO_HEXA EQU h'28'
BAJO_HEXA EQU h'29'

NUM_BINARIO EQU h'30'
CONTA_BINARIO EQU h'31'

ORG 0
GOTO INICIO

ORG 5     ;Origen del programa


INICIO:    CLRF PORTA        ;LIMPIA TODO EL PORTA
           CLRF PORTB        ;LIMPIA TODO EL PORTB

           BSF STATUS, 5
           BCF STATUS, 6    ;BANCO 01

           MOVLW 0X00        ;CARGA A W UN 0
           MOVWF TRISB     ;TRISB = 00 - PORTB COMO SALIDA

           MOVLW 0X07                ;CARGA A W UN 7
           MOVWF ADCON1  ;CONFIGURACION DEL ADCON1 - PUERTOS A Y E DIGITALES
           MOVLW 0X00                 ;CARGA A W UN 0
           MOVWF TRISA               ;TRISA = 00 - PORTA COMO SALIDA

           BSF TRISE, 0X00
           BSF TRISE, 0X01
           BSF TRISE, 0X02            ;PUERTOS DEL PORTE COMO ENTRADA

           BSF TRISD, 0X00
           BSF TRISD, 0X01
           BSF TRISD, 0X02
           BSF TRISD, 0X03
```

```
        BSF TRISD, 0X04
        BSF TRISD, 0X05
        BSF TRISD, 0X06
        BSF TRISD, 0X07         ;PUERTOS DEL PORTD COMO ENTRADA


        BCF STATUS, 5           ;BANCO 00
        CALL INICIA_LCD         ;LLAMADO A SUBRUTINA - INICIALIZACIÓN DEL LCD
        MOVLW 0X80              ;CARGA A W UN 80
        CALL COMANDO            ;LLAMADO A SUBRUTINA

;Comparaciones para cada acción a realizar
SWITCH_3    MOVLW 0X00          ;CARGA A W UN 0
            MOVWF CENT
            MOVWF DECE
            MOVWF UNI           ;INICIALIZA CENT, DECE, UNI EN 0

            MOVF TRISE, 0X00        ;W = al valor del PORTE
            SUBLW 0X00              ;RESTA 0 CON EL VALOR DE W
            BTFSC STATUS, Z         ;Revisa la bandera Z, si es 1 sigue, si es 0 salta una instrucción
            GOTO NOMBRES            ;Caso 0 - nombres
            GOTO DHB                ;Caso Decimal, Hexadecimal, Binario o Caracter


DHB     MOVF TRISE, 0X00        ;W = al valor del PORTE
        SUBLW 0X01              ;RESTA 1 CON EL VALOR DE W
        BTFSC STATUS, Z         ;Revisa la bandera Z, si es 1 sigue, si es 0 salta una instrucción
        GOTO SWITCH8_DEC        ;Caso 1 - Decimal
        GOTO HEX_BIN            ;Caso Hexadecimal, Binario o Caracter


HEX_BIN     MOVF TRISE, 0X00        ;W = al valor del PORTE
            SUBLW 0X02              ;RESTA 2 CON EL VALOR DE W
            BTFSC STATUS, Z         ;Revisa la bandera Z, si es 1 sigue, si es 0 salta una instrucción
            GOTO SWITCH8_HEX        ;Caso 2 - Hexadecimal
            GOTO BIN                ;Caso Binario o Caracter


BIN     MOVF TRISE, 0X00        ;W = al valor del PORTE
        SUBLW 0X03              ;RESTA 3 CON EL VALOR DE W
        BTFSC STATUS, Z         ;Revisa la bandera Z, si es 1 sigue, si es 0 salta una instrucción
        GOTO SWITCH8_BIN        ;Caso 3 - Binario
        GOTO CARACTER           ;Caso Caracter o ningun caso


CARACTER    MOVF TRISE, 0X00        ;W = al valor del PORTE
            SUBLW 0X04              ;RESTA 4 CON EL VALOR DE W
            BTFSC STATUS, Z         ;Revisa la bandera Z, si es 1 sigue, si es 0 salta una instrucción
            GOTO SWITCH8_CAR        ;Caso 4 - Caracter personalizado
            GOTO SWITCH_3           ;Regresa


;Se cargan los caracteres para formar los nombres en el LCD
NOMBRES
        MOVLW a'A'
        CALL DATOS
        MOVLW a'R'
            CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'T'
        CALL DATOS
        MOVLW a'H'
        CALL DATOS
```

```
        MOVLW a' '
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'M'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'N'
        CALL DATOS
        MOVLW a'O'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD

        BCF PORTA, 0
        MOVLW 0XC0
        CALL COMANDO    ;Comando para utilizar la segunda línea del LCD

        MOVLW a'G'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a' '
        CALL DATOS
        MOVLW a'M'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'Q'
        CALL DATOS
        MOVLW a'U'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'Z'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD


        CALL RET100MS
        CALL RETARDO    ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO    ;Comando para limpiar pantalla del LCD
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS   ;Llamado a subrutinas para retardos
        GOTO CAMBIO

NOMBRE2
        MOVLW a'C'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'S'
        CALL DATOS
        MOVLW a'T'
```

```
        CALL DATOS
        MOVLW a'H'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'N'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD


        BCF PORTA, 0
        MOVLW 0XC0
        CALL COMANDO    ;Comando para utilizar la segunda línea del LCD

        MOVLW a'H'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a' '
        CALL DATOS
        MOVLW a'C'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a'O'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD


        CALL RET100MS
        CALL RETARDO    ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO    ;Comando para limpiar pantalla del LCD
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS   ;Llamado a subrutinas para retardos
        GOTO CAMBIO2

NOMBRE3
        MOVLW a'A'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'J'
```

```
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'N'
        CALL DATOS
        MOVLW a'D'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'O'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD

        BCF PORTA, 0
        MOVLW 0XC0
        CALL COMANDO    ;Comando para utilizar la segunda línea del LCD

        MOVLW a'B'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'R'
        CALL DATOS
        MOVLW a'O'
        CALL DATOS

        MOVLW a' '
        CALL DATOS


        MOVLW a'V'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'L'
        CALL DATOS
        MOVLW a'D'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'Z'          ;Carga las letras al LCD
        CALL DATOS          ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD

        CALL RET100MS
        CALL RETARDO    ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO    ;Comando para limpiar pantalla del LCD
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS   ;Llamado a subrutinas para retardos
        GOTO CAMBIO3

NOMBRE4
        MOVLW a'J'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'S'
        CALL DATOS
```

```
        MOVLW a'S'
        CALL DATOS
        MOVLW a'I'
        CALL DATOS
        MOVLW a'C'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a' '
        CALL DATOS
        MOVLW a'Z'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'P'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'D'
        CALL DATOS
        MOVLW a'A'      ;Carga las letras al LCD
        CALL DATOS      ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD

        BCF PORTA, 0
        MOVLW 0XC0
        CALL COMANDO    ;Comando para utilizar la segunda línea del LCD

        MOVLW a'B'
        CALL DATOS
        MOVLW a'A'
        CALL DATOS
        MOVLW a'E'
        CALL DATOS
        MOVLW a'Z'
        CALL DATOS
        MOVLW a'A'      ;Carga las letras al LCD
        CALL DATOS      ;Llamado a subrutina para activar la lectura y escritura de caracteres en el LCD

        CALL RET100MS
        CALL RETARDO    ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO    ;Comando para limpiar pantalla del LCD
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS   ;Llamado a subrutinas para retardos
        GOTO SWITCH_3   ;Regresa

;Los cambios se encargan de verificar si los valores del PORTE (switch 3) han cambiado mientras se muestran los nombres en el LCD
CAMBIO      MOVF TRISE, 0X00
            SUBLW 0X00
             BTFSS STATUS, Z
            GOTO SWITCH_3
            GOTO NOMBRE2

CAMBIO2     MOVF TRISE, 0X00
            SUBLW 0X00
            BTFSS STATUS, Z
            GOTO SWITCH_3
            GOTO NOMBRE3

CAMBIO3     MOVF TRISE, 0X00
            SUBLW 0X00
            BTFSS STATUS, Z
            GOTO SWITCH_3
            GOTO NOMBRE4
```

```
;Carga de los valores del PORTD (switch 8) para hacer el decimal
SWITCH8_DEC     MOVF TRISD, 0X00                ;W = al valor del PORTD
                MOVWF NUM_DECIMAL
                MOVF NUM_DECIMAL, 0X00          ;NUM_DECIMAL = W
                GOTO CENTENAS

;Cálculo de las centenas
CENTENAS        MOVLW 0X64                      ;W = 64 = 100
                SUBWF NUM_DECIMAL , 0X01        ;NUM_DECIMAL - 100, se guarda en NUM_DECIMAL
                BTFSC STATUS, C                 ;Revisa la bandera C, si es 1 sigue, si es 0 salta una instrucción
                INCF CENT, 0X01             ;CENT + 1
                BTFSS STATUS, C                 ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                GOTO CARGARCENTENA
                GOTO CENTENAS


CARGARCENTENA       MOVF CENT, 0X00         ;W = CENT
                    ADDLW 0X30
                    CALL DATOS              ;Comando para mostrar el caracter del numero en el LCD
                    MOVLW 0X64
                    ADDWF NUM_DECIMAL       ;Se suma un 64 = 100 a NUM_DECIMAL
                    GOTO DECENAS

;Cálculo de las decenas
DECENAS         MOVLW 0X0A                      ;W = 0A = 10
                SUBWF NUM_DECIMAL, 0X01         ;NUM_DECIMAL - 10, se guarda en NUM_DECIMAL
                BTFSC STATUS, C                 ;Revisa la bandera C, si es 1 sigue, si es 0 salta una instrucción
                INCF DECE, 0X01                 ;DECE + 1
                BTFSS STATUS, C                 ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                GOTO CARGARDECENA
                GOTO DECENAS

CARGARDECENA        MOVF DECE, 0X00         ;W = DECE
                    ADDLW 0X30
                    CALL DATOS              ;Comando para mostrar el caracter del numero en el LCD
                    MOVLW 0X0A
                    ADDWF NUM_DECIMAL       ;Se suma un 0A = 10 a NUM_DECIMAL
                    GOTO UNIDADES

;Cálculo de las unidades
UNIDADES        MOVLW 0X01                      ;W = 01 = 1
                SUBWF NUM_DECIMAL, 0X01         ;NUM_DECIMAL - 1, se guarda en NUM_DECIMAL
                BTFSC STATUS, C                 ;Revisa la bandera C, si es 1 sigue, si es 0 salta una instrucción
                INCF UNI, 0X01                  ;UNI + 1
                BTFSS STATUS, C                 ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                GOTO CARGARUNIDAD
                GOTO UNIDADES

CARGARUNIDAD        MOVF UNI, 0X00          ;W = UNI
                    ADDLW 0X30
                    CALL DATOS              ;Comando para mostrar el caracter del numero en el LCD

                    CALL RET100MS
                    CALL RET100MS
                    CALL RET100MS               ;Llamado a subrutinas para retardos
                    MOVLW H'01'
                    CALL COMANDO                ;Comando para limpiar pantalla del LCD
                    CALL RET100MS               ;Llamado a subrutinas para retardos

                    GOTO SWITCH_3           ;Regresa
```

```
;Carga de los valores del PORTD (switch 8) para hacer el hexadecimal
SWITCH8_HEX     BCF STATUS, C
                BCF STATUS, Z                   ;Pone en 0 a la bandera C y Z
                 MOVF TRISD, 0X00                ;W = al valor del PORTD
                 MOVWF NUM_HEXA
                 MOVF NUM_HEXA, 0X00            ;NUM_HEXA = W
                 ANDLW 0XF0                     ;Operacion AND entre W y F0
                 MOVWF ALTO_HEXA                ;ALTO_HEXA = W
                 SWAPF ALTO_HEXA, 0X01          ;Intercambia los nibbles y se guarda en ALTO_HEXA
                 MOVF ALTO_HEXA, 0X00    ;W = NUM_HEXA
                 SUBLW 0X0A                     ;W - 0X0A
                 BTFSS STATUS, C                ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                 GOTO LETRA_ALTO
                 GOTO BAND_Z

;Cuando W >= 0A
BAND_Z          BTFSS STATUS, Z
                GOTO NUMERO_ALTO
                GOTO LETRA_ALTO

;Calcula la letra de los primeros 4 bits para mostrarlo en el LCD
LETRA_ALTO      MOVF ALTO_HEXA, 0X00
                ADDLW 0X37
                CALL DATOS
                GOTO BAJO

;Calcula el número de los primeros 4 bits para mostrarlo en el LCD
NUMERO_ALTO     MOVF ALTO_HEXA, 0X00
                ADDLW 0X30
                CALL DATOS
                GOTO BAJO

;Últimos 4 bits
BAJO            BCF STATUS, C
                BCF STATUS, Z                   ;Pone en 0 a la bandera C y Z
                MOVF NUM_HEXA, 0X00             ;W = NUM_HEXA
                ANDLW 0X0F                      ;Operacion AND entre W y 0F
                MOVWF BAJO_HEXA                 ;BAJO_HEXA = W
                SUBLW 0X0A                      ;W - 0A
                BTFSS STATUS, C                 ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                GOTO LETRA_BAJO
                GOTO BAND_Z2

;Cuando W >= 0A
BAND_Z2         BTFSS STATUS, Z
                GOTO NUMERO_BAJO
                GOTO LETRA_BAJO

;Calcula la letra de los últimos 4 bits para mostrarlo en el LCD
LETRA_BAJO      MOVF BAJO_HEXA, 0X00
                ADDLW 0X37
                CALL DATOS
                CALL RET100MS
                CALL RET100MS
                CALL RET100MS    ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO   ;Comando para limpiar pantalla del LCD
                CALL RET100MS           ;Llamado a subrutinas para retardos
                GOTO SWITCH_3           ;Regresa

;Calcula el número de los últimos 4 bits para mostrarlo en el LCD
NUMERO_BAJO     MOVF BAJO_HEXA, 0X00
                ADDLW 0X30
                 CALL DATOS
                CALL RET100MS
                CALL RET100MS
```

```
                CALL RET100MS           ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO    ;Comando para limpiar pantalla del LCD
                CALL RET100MS   ;Llamado a subrutinas para retardos
                GOTO SWITCH_3           ;Regresa


;Carga de los valores del PORTD (switch 8) para hacer el binario
SWITCH8_BIN     MOVF TRISD, 0X00                ;W = al valor del PORTD
                MOVWF NUM_BINARIO               ;NUM_BINARIO = W
                MOVLW 0X08
                MOVWF CONTA_BINARIO   ;COMTA_BINARIO = 8

  LOOPBIN       RLF NUM_BINARIO                 ;Rota los bits de NUM_BINARIO hacia la izquierda
                BTFSS STATUS, C                 ;Revisa la bandera C, si es 1 salta una instrucción, si es 0 sigue
                GOTO ZERO
                GOTO ONE

;Muestra un cero en pantalla del LCD
ZERO    MOVLW a'0'
        CALL DATOS
        DECFSZ CONTA_BINARIO    ;CONTA_BINARIO - 1
        GOTO LOOPBIN
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS           ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO            ;Comando para limpiar pantalla del LCD
        CALL RET100MS           ;Llamado a subrutinas para retardos
        GOTO SWITCH_3           ;Regresa

;Muestra un 1 en la pantalla del LCD
ONE     MOVLW a'1'
        CALL DATOS
        DECFSZ CONTA_BINARIO   ;CONTA_BINARIO - 1
        GOTO LOOPBIN
        CALL RET100MS
        CALL RET100MS
        CALL RET100MS           ;Llamado a subrutinas para retardos
        MOVLW H'01'
        CALL COMANDO    ;Comando para limpiar pantalla del LCD
        CALL RET100MS   ;Llamado a subrutinas para retardos
        GOTO SWITCH_3           ;Regresa


;Carga de los valores del PORTD (switch 8) para hacer el caracter personalizado
SWITCH8_CAR     MOVLW 0X40
                CALL COMANDO            ;Activa la CGRAM

                MOVLW B'00011100'
                CALL DATOS
                MOVLW B'00000101'
                CALL DATOS
                MOVLW B'00001111'
                CALL DATOS
                MOVLW B'00011111'
                CALL DATOS
                MOVLW B'00011110'
                CALL DATOS
                MOVLW B'00010100'
                CALL DATOS
                MOVLW B'00000111'
                CALL DATOS
                MOVLW B'00000000'       ;Indican las posiciones que se prenden en 1 y cuáles se apagan en 0
```

```
                CALL DATOS              ;Carga los bits en la dirección correspondiente
                CALL RET100MS

                MOVLW 0X80
                CALL COMANDO            ;Se posiciona en la primera posición del LCD
                CALL RET100MS

                MOVLW 0X00
                CALL DATOS              ;Comando para mostrar el carter en el LCD
                CALL RET100MS
                CALL RET100MS           ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO            ;Comando para limpiar pantalla del LCD
                CALL RET100MS           ;Llamado a subrutinas para retardos

                GOTO CAMBIOA

CARACTER2
                MOVLW 0X47
                CALL COMANDO

                MOVLW B'00010101'
                CALL DATOS
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
                MOVLW B'00000000'       ;Indican las posiciones que se prenden en 1 y cuáles se apagan en 0
                CALL DATOS              ;Carga los bits en la dirección correspondiente
                CALL RET100MS

                MOVLW 0X81
                CALL COMANDO            ;Se posiciona en la segunda posición del LCD
                CALL RET100MS

                MOVLW 0X01
                CALL DATOS              ;Comando para mostrar el caracter en el LCD
                CALL RET100MS
                CALL RET100MS           ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO            ;Comando para limpiar pantalla del LCD
                CALL RET100MS           ;Llamado a subrutinas para retardos

                GOTO CAMBIOB

CARACTER3
                MOVLW 0X40
                CALL COMANDO
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00000100'
                CALL DATOS
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00011111'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
```

```
                MOVLW B'00001010'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
                CALL RET100MS
                MOVLW B'00000000'       ;Indican las posiciones que se prenden en 1 y cuáles se apagan en 0
                CALL DATOS              ;Carga los bits en la dirección correspondiente

                MOVLW 0X80
                CALL COMANDO            ;Se posiciona en la primera posición del LCD
                CALL RET100MS

                MOVLW 0X00
                CALL DATOS              ;Comando para mostrar el caracter en el LCD
                CALL RET100MS
                CALL RET100MS           ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO            ;Comando para limpiar pantalla del LCD
                CALL RET100MS           ;Llamado a subrutinas para retardos

                GOTO CAMBIOC

CARACTER4
                MOVLW 0X48
                CALL COMANDO

                MOVLW B'00011100'
                CALL DATOS
                MOVLW B'00001000'
                CALL DATOS
                MOVLW B'00011111'
                CALL DATOS
                MOVLW B'00010101'
                CALL DATOS
                MOVLW B'00011111'
                CALL DATOS
                MOVLW B'00000010'
                CALL DATOS
                MOVLW B'00000111'
                CALL DATOS
                MOVLW B'00000000'       ;Indican las posiciones que se prenden en 1 y cuáles se apagan en 0
                CALL DATOS              ;Carga los bits en la dirección correspondiente
                CALL RET100MS

                MOVLW 0X81
                CALL COMANDO            ;Se posiciona en la segunda posición del LCD
                CALL RET100MS

                MOVLW 0X01
                CALL DATOS              ;Comando para mostrar el caracter en el LCD
                CALL RET100MS
                CALL RET100MS           ;Llamado a subrutinas para retardos
                MOVLW H'01'
                CALL COMANDO            ;Comando para limpiar pantalla del LCD
                CALL RET100MS           ;Llamado a subrutinas para retardos

                GOTO SWITCH_3           ;Regresa

;Los cambios se encargan de verificar si los valores del PORTE (sitch 3) han cambiado mientras se muestran los caracteres en el LCD
CAMBIOA         MOVF TRISE, 0X00
                SUBLW 0X04
                BTFSS STATUS, Z
                GOTO SWITCH_3
                GOTO CARACTER2
```

```
CAMBIOB         MOVF TRISE, 0X00
                SUBLW 0X04
                BTFSS STATUS, Z
                GOTO SWITCH_3
                GOTO CARACTER3

CAMBIOC         MOVF TRISE, 0X00
                SUBLW 0X04
                BTFSS STATUS, Z
                GOTO SWITCH_3
                GOTO CARACTER4


;Comfiguracion del LCD
INICIA_LCD:     MOVLW 0X30
                CALL COMANDO
                CALL RET100MS
                MOVLW 0X30
                CALL COMANDO
                CALL RET100MS
                MOVLW 0X38
                CALL COMANDO
                MOVLW 0X0C
                CALL COMANDO
                MOVLW 0X01
                CALL COMANDO
                MOVLW 0X06
                CALL COMANDO
                MOVLW 0X02
                CALL COMANDO
                RETURN

COMANDO:        MOVWF PORTB             ;PORTB = W
                CALL RET200             ;LLAMADO A UN RETARDO
                BCF PORTA, 0            ;BIT 0 DEL PORTA = 0  --> RS
                BSF PORTA, 1            ;BIT 1 DEL PORTA = 1  --> E
                CALL RET200             ;LLAMADO A UN RETARDO
                BCF PORTA, 1            ;BIT 1 DEL PORTA = 0  --> E
                RETURN

DATOS:          MOVWF PORTB              ;PORTB = W
                CALL RET200             ;LLAMADO A RETARDO
                BSF PORTA, 0            ;BIT 0 DEL PORTA = 1  --> RS
                BSF PORTA, 1            ;BIT 1 DEL PORTA = 1  --> E
                CALL RET200             ;LLAMADO A RETARDO
                BCF PORTA, 1            ;BIT 1 DEL PORTA = 0  --> E
                CALL RET200             ;LLAMADO A RETARDO
                CALL RET200             ;LLAMADO A RETARDO
                RETURN

RET200          MOVLW 0X02
                MOVWF VALOR1
        LOOP    MOVLW d'164'
                MOVWF VALOR
        LOOP1   DECFSZ VALOR, 1
                GOTO LOOP1
                DECFSZ VALOR1, 1
                GOTO LOOP
                RETURN

RET100MS:       MOVLW 0X03
                MOVWF VALOR
        TRES:   MOVLW 0XFF
                MOVWF VALOR1
        DOS:    MOVLW 0XFF
```

```
                MOVWF VALOR2
        UNO:    DECFSZ VALOR2
                GOTO UNO
                DECFSZ VALOR1
                GOTO DOS
                DECFSZ VALOR
                GOTO TRES
                RETURN


;Retardo más largo para dejar en pantalla más tiempo los caracteres a mostrar
RETARDO         MOVLW 0XFF      ;Carga una literal a w
                MOVWF 0X33      ;Carga el valor de w a 0x22
LOOPRET         CALL RETARDO2   ;Llamado a otra subrutina
                DECFSZ 0X33     ;Decrementa el valor de la dirección 0x22 y salta si es 0
                GOTO LOOPRET    ;Ciclo
                RETURN          ;Retorna a la siguiente línea donde se quedó
RETARDO2        MOVLW 0XFF      ;Carga una literal a w
                MOVWF 0X34      ;Carga el valor de w a 0x23
LOOP2   CALL RETARDO3   ;Llamado a otra subrutina
                DECFSZ 0X34     ;Decrementa el valor de la dirección 0x23 y salta si es 0
                GOTO LOOP2      ;Ciclo
                RETURN          ;Retorna a la siguiente línea donde se quedó
RETARDO3        MOVLW 0X1E      ;Carga una literal a w
                MOVWF 0X35      ;Carga el valor de w a 0x24
LOOP3   DECFSZ 0X35     ;Decrementa el valor de la dirección 0x24 y salta si es 0
                GOTO LOOP3      ;Ciclo
                RETURN


END     ;Fin del programa
```

Conclusiones y/o comentarios

Barreiro Valdez Alejandro:

En este proyecto se agregaron nuevas componentes al sistema mínimo que se realizó anteriormente. Se utilizaron puertos paralelos para poder tener dos entradas y una salida en el mismo microcontrolador. Se utilizaron dos *Dip Switches* y un *display* de cristal líquido para realizar el proyecto. Se programaron varios modos de cosas que se podrían mostrar en el display. De esta manera se pudo entender cómo manipular los caracteres que se muestran en el *display*. Además, se hicieron tres conversiones para poder mostrar un número dado de entrada en una salida en hexadecimal, en binario y en decimal. Por último, se pudo realizar un caracter especial por integrante que se mostró en el *display*. Para este proyecto se pusieron en práctica conceptos vistos en teoría y en laboratorio como la programación en ensamblador, la creación de subrutinas de retardo, asignación de puertos paralelos como entrada o salida y esquemas de conexiones con el microcontrolador. De esta manera, se logró realizar el proyecto y entender cómo funciona el *display*.

Gil Márquez Arath Emiliano:

Con lo visto en clase y con la realización de los proyectos se puede ver que el PIC16F877A es un microcontrolador que se puede utilizar para una amplia variedad de aplicaciones, en este caso para mostrar caracteres en un LCD en función de los valores de unos Dip Switches, el microcontrolador tiene suficientes pines de entrada y salida para manejar tanto los Dip Switches como el LCD, y su capacidad de procesamiento es más que suficiente para manejar las operaciones necesarias para leer el estado de los switches y enviar los caracteres correspondientes al LCD. Para poder llevar a cabo de manera correcta las acciones en el LCD fue necesario entender bien el lenguaje ensamblador, además de la configuración respecto al uso del LCD.

Herrera Carrillo Cristhian:

Los puertos paralelos son una tecnología de comunicación de datos que ha sido ampliamente utilizada en la informática desde la década de 1970 hasta principios de la década de 2000. Estos puertos permitían la transmisión de datos entre un ordenador y dispositivos periféricos como impresoras, escáneres, unidades de cinta y otros dispositivos de almacenamiento externos. Con la evolución de la tecnología, la mayoría de los dispositivos periféricos han adoptado interfaces de conexión más avanzadas como el USB o el Thunderbolt, lo que ha llevado a una disminución en la popularidad y el uso de los puertos paralelos. Aunque todavía hay algunos dispositivos antiguos que utilizan puertos paralelos, la mayoría de los ordenadores modernos ya no incluyen estos puertos.

Los puertos paralelos fueron una tecnología importante en la informática, pero han sido reemplazados por interfaces de conexión más avanzadas y eficientes. Aunque todavía se pueden encontrar algunos dispositivos que utilizan puertos paralelos, la mayoría de los usuarios no necesitan preocuparse por esta tecnología en la actualidad.

Zepeda Baeza Jessica:

En este proyecto se observó una aplicación más real del PIC16F877A debido a que se le conectó un Display LCD y dos Switches a sus puertos paralelos. Además se le cargó un programa mucho más largo y complejo que dependiendo el valor de uno de los Switches, mostraba nombres, números en diferentes formatos y caracteres en el Display LCD. La construcción del circuito no fue tan complicada debido a que se utilizó el sistema mínimo realizado en el proyecto anterior y sólo se conectaron los switches y LCD a los puertos correspondientes. La única dificultad fue que conseguimos un LCD sin pines por lo que hubo que soldarlos. Por otro lado, manejar el LCD fue igual sencillo debido a las subrutinas de COMANDO y DATOS que nos permitieron diferenciar entre info que aplica un control al LCD o info que representa un dato a mostrar. En el código se configuró cada puerto como entrada o salida y como datos digitales. Fue un proyecto largo pero muy completo y alcanzable con la información proporcionada, los esquemas y los conocimientos en lenguaje ensamblador.