



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**Departamento de Eléctrica y Electrónica**

**Carrera de Electrónica y Automatización**

**SISTEMAS BASADOS EN MCU**

**Práctica 3.5**

**MANEJO DE UN LCD, TECLADO MATRICIAL E  
INTERRUPCIONES CON EL PIC16F877**

**Autor:**

**Iza Tipanluisa Alex Paul**

**Docente:**

**Ing. Amparo Meythaler**

**NRC: 4891**

## **1) OBJETIVOS**

- Identificar la forma de manejar un LCD.
- Realizar una aplicación con LCD, teclado matricial e interrupción en lenguaje C para el PIC16F877

## **2) MARCO TEORICO**

### **LCD**

La sigla LCD alude a Liquid Crystal Display, una expresión de la lengua inglesa que puede traducirse como Pantalla de Cristal Líquido. Una LCD, por lo tanto, es un tipo de pantalla, que se caracteriza por ser plana y por estar formada por píxeles que contienen moléculas de cristal líquido. Estos píxeles se ubican entre dos capas de cristales polarizados y son iluminados por una fuente de luz fija. Las moléculas de cristal líquido permiten el paso de mayor o menor cantidad de luz según el estímulo eléctrico que reciben.

Puede decirse, por lo tanto, que una pantalla de LCD muestra las imágenes a partir de esa retroiluminación que permite la combinación de los colores. Una de las desventajas de este sistema es que no puede brindar un negro absoluto, una característica que incide en el contraste y que atenta contra la calidad de la imagen.

Aprender a utilizar pantallas LCD es muy importante cuando se utilizan microcontroladores, las pantallas lcd son útiles para obtener información en forma visual sobre qué es lo que está ocurriendo en el microcontrolador, como puede ser los valores de los registros, o de alguna variable que se esté utilizando, la observación por ejemplo de algún dato como puede ser una medida analógica entre muchas otras informaciones de interés.

## **3) EQUIPOS Y MATERIALES**

- PC con el paquete PIC C
- PC con el paquete PROTEUS.

## **4) ACTIVIDADES**

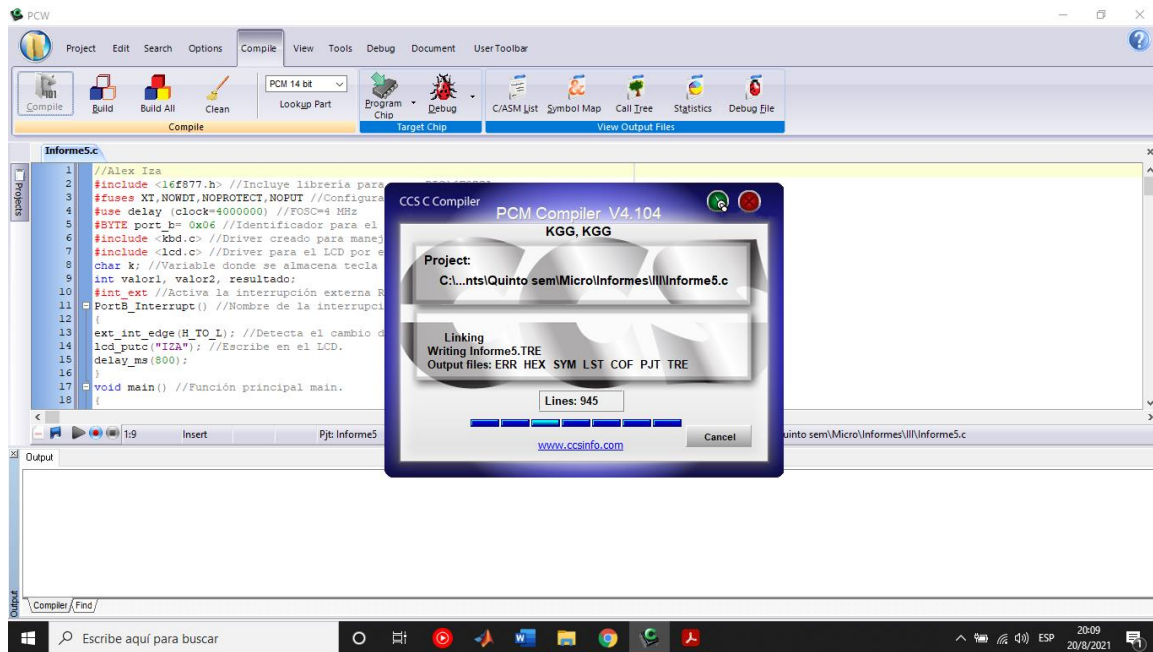
### **1) Trabajo Preparatorio:**

a) Realice el diagrama de flujo, la codificación y la implementación correspondientes de un programa que reciba dos datos de un dígito cada uno y los multiplique. Los datos ingresan mediante un teclado matricial (NO realizado con pulsantes individuales); los datos ingresados y la respuesta se visualiza en un LCD 16x2.

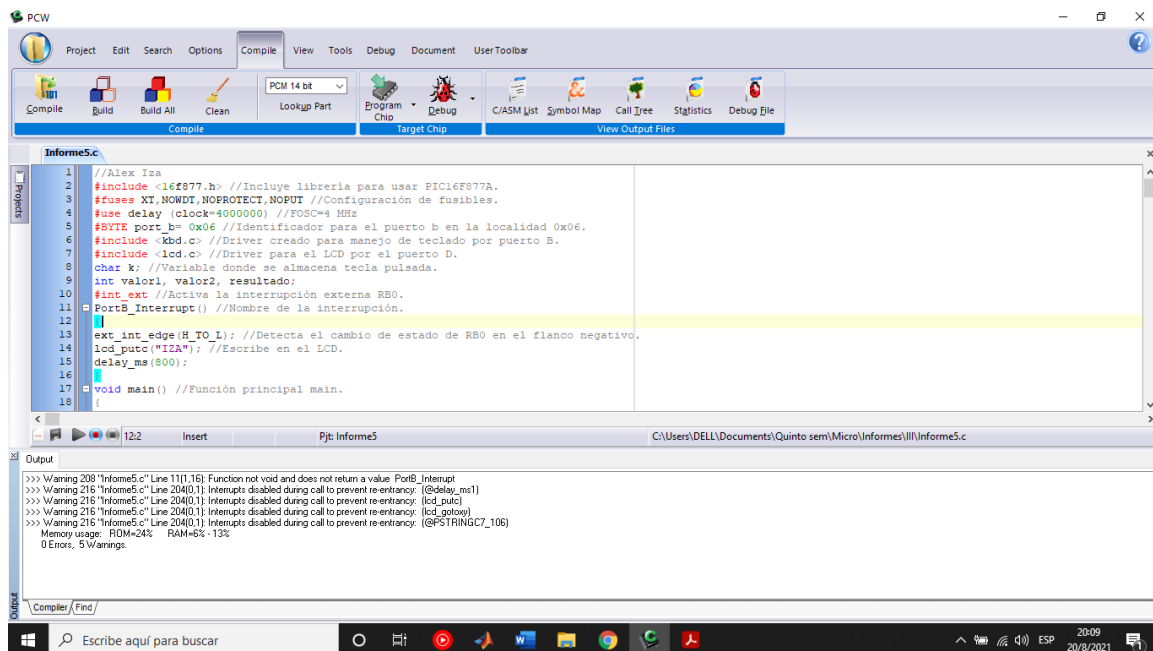
Si aparece la interrupción externa 0, por un tiempo en el LCD, se verá su Primer Apellido.

Si lo requiere coloque un retardo entre operaciones.

b) Verifique el funcionamiento del MPLAB IDE y del PROTEUS.



*Ilustración 1, Compilación del programa*



*Ilustración 2, Compilación exitosa, 0 errores*

## 5) RESULTADOS

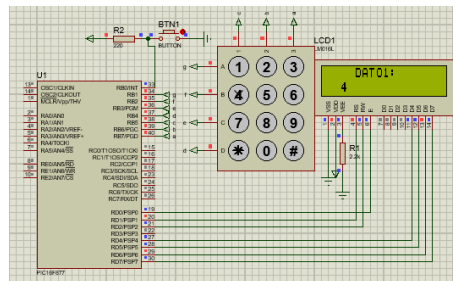
- Explique los errores cometidos en el ejercicio realizado (si los tuvo) y la forma de corregirlos.

Tuve el error de la interrupción externa; no logré solucionarlo.

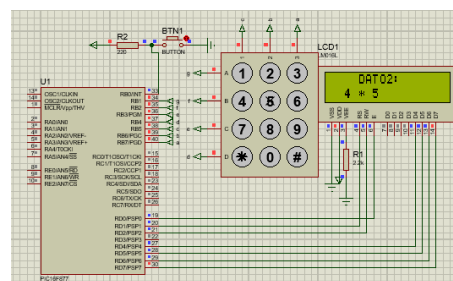
- **Cuáles sentencias se pueden cambiar la conexión de un LCD a otro puerto que no sea el que está por defecto en la librería lcd.c.**

Se debe cambiar las sentencias que se encuentran en el archivo del directorio de instalación del PICC, carpeta Drivers y se debe retirar los // que se encuentran en la línea de código `#define use_portb_kbd TRUE`; y se debe renombrar de nuevo por KBDB.C.

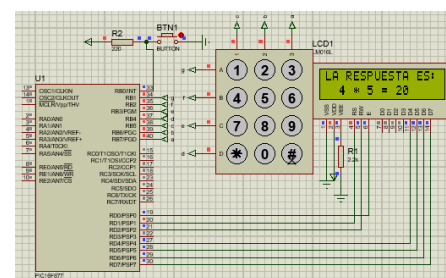
## 6) DISEÑO



*Ilustración 3, Visualización del ingreso del primer dato*

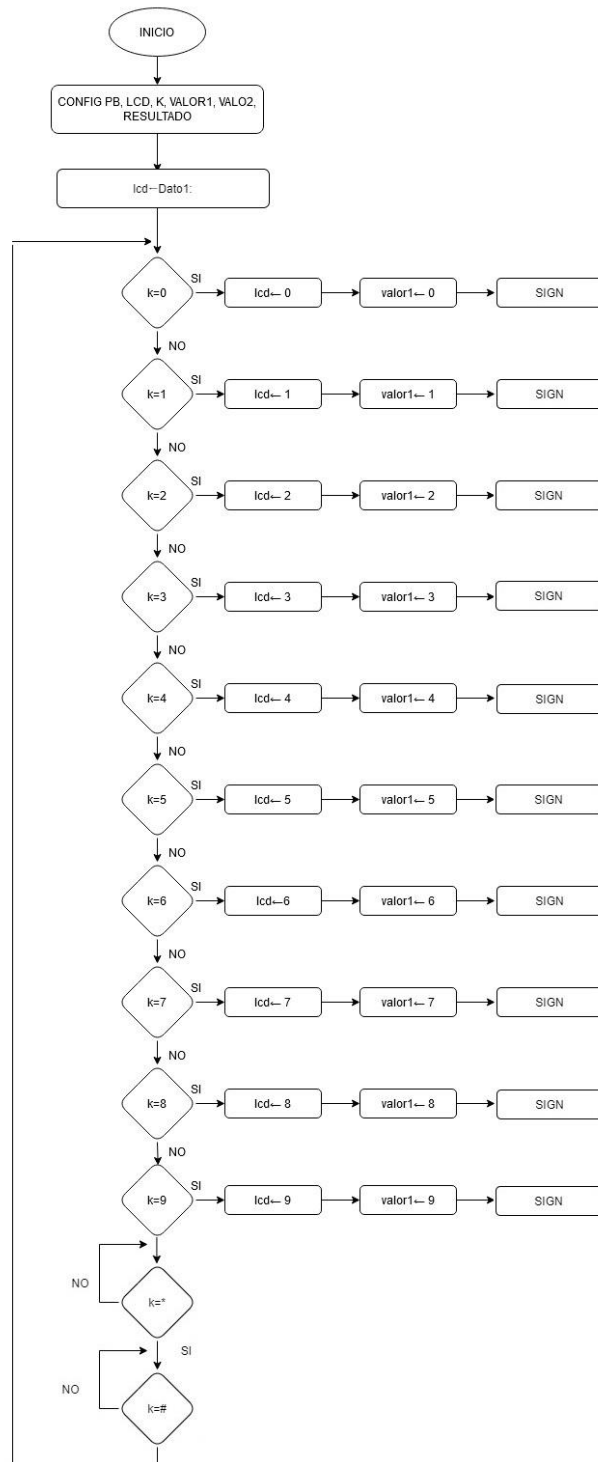


*Ilustración 4, Visualización del ingreso del segundo dato*



*Ilustración 5, Visualización del resultado de la multiplicación*

## 7) DIAGRAMA DE FLUJO



*Diagrama 1*

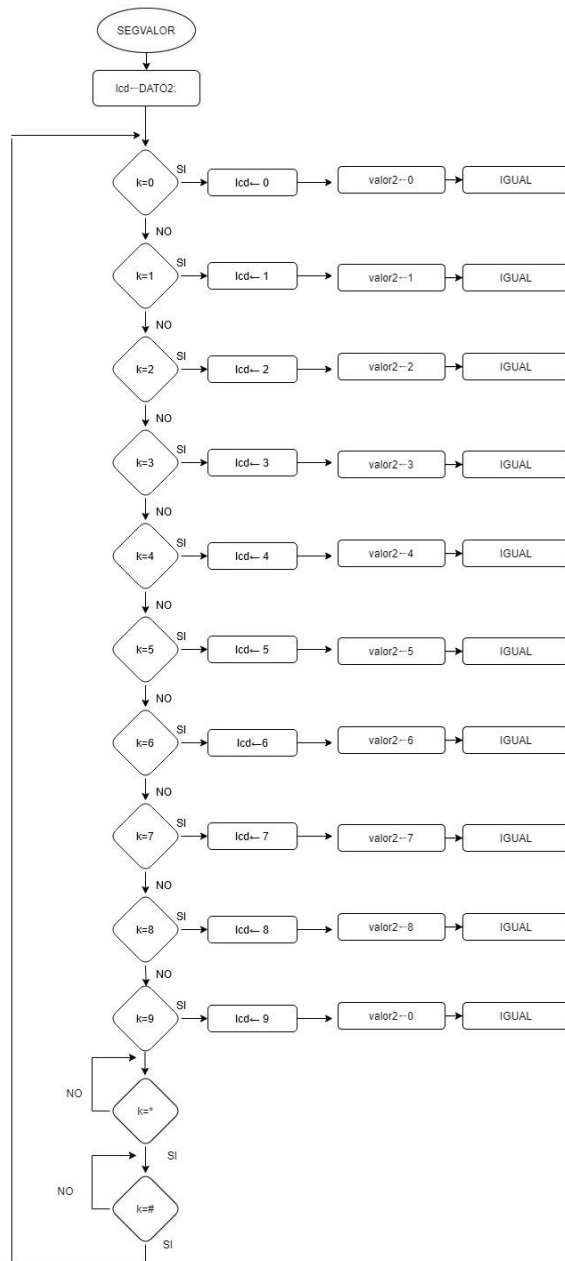


Diagrama 2

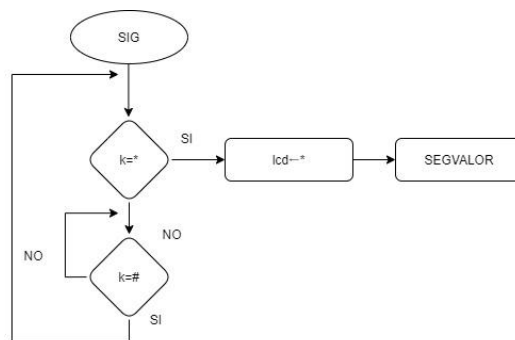


Diagrama 3

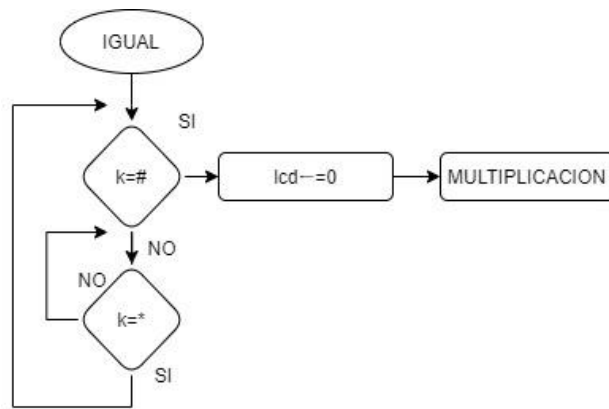


Diagrama 4

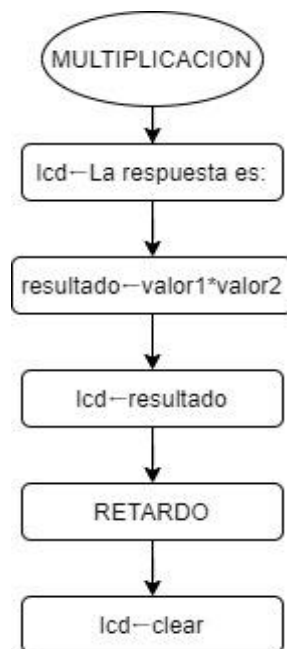


Diagrama 5

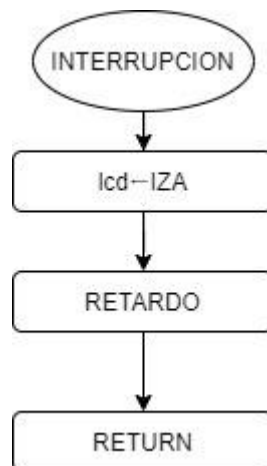


Diagrama 6

## 8) PROGRAMA

```
//Alex Iza

#include <16f877.h> //Incluye librería para usar PIC16F877A.

#fuses XT,NOWDT,NOPROTECT,NOPUT //Configuración de fusibles.

#use delay (clock=4000000) //FOSC=4 MHz

#BYTE port_b= 0x06 //Identificador para el puerto b en la localidad 0x06.

#include <kbd.c> //Driver creado para manejo de teclado por puerto B.

#include <lcd.c> //Driver para el LCD por el puerto D.

char k; //Variable donde se almacena tecla pulsada.

int valor1, valor2, resultado;

#int_ext //Activa la interrupción externa RB0.

PortB_Interrupt() //Nombre de la interrupción.

{

    ext_int_edge(H_TO_L); //Detecta el cambio de estado de RB0 en el flanco
    negativo.

    lcd_putc("IZA"); //Escribe en el LCD.

    delay_ms(800);

}

void main() //Función principal main.

{

    set_tris_b(0b00000001); //Fija el puerto RB1 como entrada y el resto como salida.

    inicio:

    lcd_init(); //Inicializa el LCD.

    kbd_init(); //Inicializa el teclado.

    enable_interrupts(GLOBAL); //Habilitación de todas(GLOBAL) las
    interrupciones.

    enable_interrupts(INT_EXT);
```



```
port_b_pullups(TRUE); //Activa las resistencias PULL UP del puerto B.
```

```
lcd_gotoxy(5,1);
```

```
printf(lcd_putc,"DATO1:"); //Mensaje para la tecla pulsada.
```

```
while(TRUE) { //Inicio del bucle...
```

```
k = kbd_getc(); //Detecta la tecla pulsada.
```

```
lcd_gotoxy(3,2);
```

```
switch (k)
```

```
{ //Muestra en el LCD el valor de la tecla pulsada.
```

```
case '0': lcd_putc("0");
```

```
delay_ms(800);
```

```
valor1=0;
```

```
goto SIG;
```

```
break;
```

```
case '1': lcd_putc("1");
```

```
delay_ms(800);
```

```
valor1=1;
```

```
goto SIG;
```

```
break;
```

```
case '2': lcd_putc("2");
```

```
delay_ms(800);
```

```
valor1=2;
```

```
goto SIG;
```

```
break;
```

```
case '3': lcd_putc("3");
```

```
delay_ms(800);
```

```
valor1=3;
```

```
goto SIG;

break;

case '4': lcd_putc("4");

delay_ms(800);

valor1=4;

goto SIG;

break;

case '5': lcd_putc("5");

delay_ms(800);

valor1=5;

goto SIG;

break;

case '6': lcd_putc("6");

delay_ms(800);

valor1=6;

goto SIG;

break;

case '7': lcd_putc("7");

delay_ms(800);

valor1=7;

goto SIG;

break;

case '8': lcd_putc("8");

delay_ms(800);

valor1=8;

goto SIG;

break;
```

```
case '9': lcd_putc("9");  
delay_ms(800);  
valor1=9;  
goto SIG;  
break;  
case '#': break;  
case '*': break;  
default: break;  
}
```

SIG:

```
lcd_gotoxy(5,2);  
switch (k)  
{ //Muestra en el LCD el valor de la tecla pulsada.  
case '#':  
break;  
case '*': lcd_putc("*");  
GOTO SEGVALOR;  
break;  
default: break;  
}
```

}

SEGVALOR:

```
lcd_gotoxy(5,1);  
printf(lcd_putc,"DATO2:");
```

```
while(TRUE) { //Inicio del bucle...

lcd_gotoxy(7,2);

k = kbd_getc(); //Detecta la tecla pulsada.

switch (k)

{ //Muestra en el LCD el valor de la tecla pulsada.

case '0': lcd_putc("0");

delay_ms(800);

valor2=0;

goto IGUAL;

break;

case '1': lcd_putc("1");

delay_ms(800);

valor2=1;

goto IGUAL;

break;

case '2': lcd_putc("2");

delay_ms(800);

valor2=2;

goto IGUAL;

break;

case '3': lcd_putc("3");

delay_ms(800);

valor2=3;

goto IGUAL;

break;

case '4': lcd_putc("4");
```

```
delay_ms(800);  
  
valor2=4;  
  
goto IGUAL;  
  
break;  
  
case '5': lcd_putc("5");  
  
delay_ms(800);  
  
valor2=5;  
  
goto IGUAL;  
  
break;  
  
case '6': lcd_putc("6");  
  
delay_ms(800);  
  
valor2=6;  
  
goto IGUAL;  
  
break;  
  
case '7': lcd_putc("7");  
  
delay_ms(800);  
  
valor2=7;  
  
goto IGUAL;  
  
break;  
  
case '8': lcd_putc("8");  
  
delay_ms(800);  
  
valor2=8;  
  
goto IGUAL;  
  
break;  
  
case '9': lcd_putc("9");  
  
delay_ms(800);  
  
valor2=9;
```

```
goto IGUAL;

break;

case '#': break;

case '*': break;

default: break;

}
```

IGUAL:

```
lcd_gotoxy(9,2);

switch (k)

{ //Muestra en el LCD el valor de la tecla pulsada.
```

```
case '#':
```

```
    lcd_putc("=");
```

GOTO MULTIPLICACION;

```
break;
```

```
case '*':
```

```
break;
```

```
default: break;
```

```
}
```

```
//Ubica el cursor en la 1 columna. fila2
```

```
}
```

MULTIPLICACION:

```
lcd_gotoxy(1,1);
```

```
printf(lcd_putc,"LA RESPUESTA ES:"); //Mensaje para la tecla pulsada.
```

```
resultado=valor1*valor2;

lcd_gotoxy(11,2);

printf(lcd_putc,"%u",resultado);

delay_ms(2200);


printf(lcd_putc,"\f");

delay_ms(500);

goto inicio;

}
```

## **9) CONCLUSIONES**

- Se ha concluido que si no se sabe el tipo del display; al momento de conectarle en un ánodo común si se necesita el número cero aparecerá una línea si es cátodo común, por lo cual se deberá cambiar.
- Se ha concluido que la programación en C es muy diferente de la programación en ensamblador; pero el diagrama de flujo es general, por lo cual se puede usar con cualquier lenguaje de programación.
- Se ha concluido que en el lenguaje de programación C no es necesario cambiarnos de banco para usar los TRIS

## **10) RECOMENDACIONES**

- Se recomienda revisar la conexión del LCD ya que este no funcionara si uno de sus pines no están correctamente conectados
- Se recomienda verificar el funcionamiento del programa con todos los datos que podrían generar problemas.
- Se recomienda usar variables con el nombre de la función que va a realizar para que no se generen complicaciones y confusiones en su uso

## **11) BIBLIOGRAFIA**

Meythaler, A. (2021). Sistemas basados en MCU. Ecuador: UFA ESPE