



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Laboratorio de Microcomputadoras

Práctica 8. Programación en C
Puertos paralelos E/S, Puerto Serie Asíncrono

Grupo de Laboratorio: 4

Sanjuan Aldape Diana Paola
Grupo de Teoría: 5

Esparza Fuentes Jorge Luis
Grupo de Teoría: 4

Fecha de realización: 26 de abril del 2022



DESARROLLO

1.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

```
#include <16f877.h>
#fuses HS,NOPROTECT, // fusibles
#use delay(clock=2000000) // reloj
#org 0x1F00, 0x1FFF void loader16F877(void) {} // se manda a llamar el bootloader
// con la finalidad de que no se sobre escriba esta seccion de memoria
void main(){
    // seccion de configuraciones y decalacione de variables
    while(1){ // Loop infinito
        output_b(0x01); // se pasa el valor de 1 al puerto B => 00000001
        delay_ms(1000); // retardo de 1 segundo
        output_b(0x00); // se pasa el valor de 0 al puerto B => 00000000
        delay_ms(1000);
    } //while
} //main
```

2.- Modificar el programa para que active y desactive todos los bits del puerto B.

```
#include <16f877.h>
#fuses HS,NOPROTECT, // fusibles
#use delay(clock=2000000) // reloj
#org 0x1F00, 0x1FFF void loader16F877(void) {} // se manda a llamar el bootloader
// con la finalidad de que no se sobre escriba esta seccion de memoria
void main(){
    // seccion de configuraciones y decalacione de variables
    while(1){ // Loop infinito
        output_b(0x01); // se pasa el valor de 1 al puerto B => 00000001
        delay_ms(1000); // retardo de 1 segundo
        output_b(0x00); // se pasa el valor de 0 al puerto B => 00000000
        delay_ms(1000);
    } //while
} //main
```

3.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

```
#include <16f877.h>
#fuses HS,NOPROTECT, // fusibles
#use delay(clock=2000000) // reloj
#org 0x1F00, 0x1FFF void loader16F877(void) {} // se manda a llamar el bootloader
// con la finalidad de que no se sobre escriba esta seccion de memoria
void main(){
    // seccion de configuraciones y decalacione de variables
    while(1){ // Loop infinito
        var = input_a(); //Se guarda en var lo que se ingresa en el puerto a
        output_b(var); //Se pasa "var" al puerto b
    } //while
} //main
```



4.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

```
#include <16f877.h>
#fuses HS,NOPROTECT,
#use delay(clock=2000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void) {}
void main(){
    while(1){
        output_b(0xff);    //se pasa el valor de 1 al puerto B => 11111111
        printf(" Todos los bits encendidos \n\r"); //Se imprime en la terminal
        delay_ms(1000); //Retardo de 1 segundo
        output_b(0x00); // se pasa el valor de 0 al puerto B => 00000000
        printf(" Todos los leds apagados \n\r");
        delay_ms(1000); //Retardo de 1 seg
    } //while
} //main
```

5.- Escribir, comentar, compilar, el siguiente programa usando el ambiente del PIC C Compiler y comprobar el funcionamiento.

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=2000000)
#define use_portd_lcd true
#include <lcd.c>
void main() {
    lcd_init(); //Se llama a la función lcd_init de la librería lcd.c
    while( TRUE ) {
        lcd_gotoxy(1,1); //Se selecciona la fila y columna 1 del LCD
        printf(lcd_putc, " UNAM \n "); //Se imprime en la posición seleccionada
        lcd_gotoxy(1,2); //Se selecciona la columna y fila, 1 y 2 respectivamente.
        printf(lcd_putc, " FI \n "); //Se imprime en la posición seleccionada
        delay_ms(300); // Retardo de 0.3 s
    }
}
```

6.- Realizar un programa empleando el compilador de C, para ejecutar las acciones mostradas en la siguiente tabla, estas son controladas a través del puerto serie; usar retardos de ½ segundos.



Práctica 5. Puertos paralelos E/S, Puerto Serie Asíncrono
Fecha de realización: 26/04/2022

DATO	ACCION Puerto B	Ejecución
0	Todos los bits apagados	00000000
1	Todos los bits encendidos	11111111
2	Corrimiento del bit más significativo hacia la derecha	10000000 00000001
3	Corrimiento del bit menos significativo hacia la izquierda	00000001 10000000
4	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 00000001 10000000
5	Apagar y encender todos los bits.	00000000 11111111

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=2000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void) {}
void main() {
    char dato;
    int i;
    int aux;
    while( TRUE ) {
        dato = getchar();
        putchar(dato);
        switch(dato){
            case '0':
                output_b(0);
                break;
            case '1':
                output_b(255);
                break;
            case '2':
                aux = 0x01;
                for(i=0; i<8 ; i++){
                    delay_ms(500);
                    output_b(aux);
                    aux <<= 0b1;
                }
                break;
            case '3':
                aux = 0b10000000;
                for(i=0; i<8 ; i++){
                    delay_ms(500);
                    output_b(aux);
                    aux >>= 0b1;
                }
                break;
            case '4':
                aux = 0x01;
                for(i=0; i<7 ; i++){
                    delay_ms(500);
                    output_b(aux);
                    aux <<= 0b1;
                }
                for(i=0; i<8 ; i++){
                    delay_ms(500);
                    output_b(aux);
                    aux >>= 0b1;
                }
                break;
            case '5':
                output_b(0);
                delay_ms(500);
                output_b(255);
                delay_ms(500);
                output_b(0);
                delay_ms(500);
                output_b(255);
                break;
        }
    }
}
```



7.- Realizar un programa que muestre en un Display de Cristal Líquido, la cantidad de veces que se ha presionado un interruptor, el cual está conectado a la terminal A0.

El despliegue a mostrar es:

- Primer línea y 5 columna; la cuenta en decimal
- Segunda línea y 5 columna; la cuenta en hexadecimal

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=2000000)
#define use_portd_lcd true
#include <lcd.c>
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void) {}
void main() {
    int cont = 0;
    lcd_gotoxy(5,1);
    printf(lcd_putc," %d\n ",cont);
    lcd_gotoxy(5,2);
    printf(lcd_putc," %x \n ",cont);
    while( TRUE ) {
        if(input(PIN_A0)){
            cont ++;
        }
        lcd_gotoxy(5,1);
        printf(lcd_putc," %d\n ",cont);
        lcd_gotoxy(5,2);
        printf(lcd_putc," %x \n ",cont);
        delay_ms(300);
    }
}
```

CONCLUSIÓN

- Sanjuan Aldape Diana Paola

En esta práctica empleamos el PIC C Compiler, un IDE que nos permitió realizar programas controlados a través del puerto serie. El código en comparación con ensamblador fue mucho más corto y más sencillo, a pesar de que el objetivo fue el mismo. C también nos permitió acceder a los registros del sistema, trabajar con direcciones de memoria, utilizar los puertos, así como con el lenguaje ensamblador.

- Esparza Fuentes Jorge Luis

Se realizó por primera vez en el curso la codificación por medio del lenguaje C para poder programar el microcontrolador PIC, y de esta forma se logró una vez más entablar comunicación serial e incluso mantener un monitor serial, además se logró mostrar información (cadenas de caracteres) controlando una pantalla LSD 16x2. De esta forma se logró obtener los resultados esperados de esta práctica, ahora utilizando el lenguaje C para programar el microcontrolador.