

Práctica 2. Entrada y salida en C

(Duración: 1 sesión)

Introducción

En esta práctica se usará la tarjeta montada en la práctica 1 con el objetivo de familiarizarse con los puertos de entrada y salida digital para interactuar con los LEDs y el pulsador de la placa.

Objetivos

Al concluir la práctica, el alumno deberá ser capaz de:


- Manejar los puertos de entrada y salida digital del PIC32.
- Realizar programas sencillos que interactúen con el *hardware*.
- Depurar programas para encontrar sus fallos.

Trabajo previo

Antes de ir al laboratorio ha de:

- Leer detenidamente la práctica, anotando las dudas que le surjan para preguntárselas al profesor.
- Complete los programas de las secciones 2, 4 y 5.
- Escriba los programas solicitados en las secciones 3 y 6.

1. Crear un proyecto

En primer lugar ha de crear un proyecto para esta práctica, siguiendo los pasos indicados en el apartado 2 de la práctica anterior. No olvide trabajar en la carpeta  D: ►Micros►Grupo_XX, creada en la práctica 1.

Por otro lado asegúrese de que los *jumpers* JP1 y JP3 están conectados, ya que en caso contrario no funcionará ni el pulsador ni los LEDs.

2. Manejo de puertos

La tarjeta de desarrollo dispone de 4 LEDs conectados a los 4 bits menos significativos del puerto C (RC3 a RC0). Cree el archivo `main1.c` y copie el siguiente código, modificando las líneas 5 y 6 para definir los pines RC3 a RC0 como salidas, el resto de pines del puerto como entradas y encender los 4 LEDs de la placa. Tenga en cuenta que los LEDs son activos a nivel bajo.

```
1  #include <xc.h>
2
3  int main(void)
4  {
5      TRISC = XX; // A completar por el alumno (trabajo previo)
6      LATC  = XX; // A completar por el alumno (trabajo previo)
7
8      while(1)
9          ;
10     return 0;
11 }
```

3. Diseño 1

Modifique el programa de la sección anterior para:

1. Encender sólo el LED conectado a RC2.
2. Encender sólo los LEDS RC3 y RC2.

4. Manejo del pulsador

El pulsador está conectado al pin 5 del puerto B (RB5). Cree ahora un nuevo fichero, denominado main2.c y escriba el siguiente programa:

```
1  #include <xc.h>
2
3  #define PIN_PULSADOR 5
4
5  int main(void)
6  {
7      int pulsador;
8
9      TRISC = XX; // A completar por el alumno (trabajo previo)
10     LATC  = XX; // A completar por el alumno (trabajo previo)
11
12     TRISB = XX; // A completar por el alumno (trabajo previo)
13
14     while(1){
15         // Se lee el estado del pulsador
16         pulsador = (PORTB >> PIN_PULSADOR) & 1;
17         // A completar por el alumno (trabajo previo)
18         if(pulsador == 0){
19             LATC &= ~1;
20         }else{
21             LATC |= 1;
22         }
23     }
24 }
```

En el programa anterior ha de modificar:

- La línea 9 para definir los pines RC3 a RC0 como salidas, el resto de pines del puerto como entradas.
- La línea 10 para apagar todos los LEDs de la placa.
- La línea 12 para definir todas las patillas del puerto B como entradas.
- La línea 17 para indicar en el comentario qué hace el programa en las líneas 18 a 22.

Antes de compilar el programa ha de eliminar el archivo `main1.c` del proyecto, pues si no el compilador se negará amablemente a crear el ejecutable diciéndole que tiene dos `main` y que él no es lo suficientemente inteligente para saber cuál de los dos hay que ejecutar. Para ello pulse con el botón derecho del ratón sobre el archivo `main1.c` y seleccione “*Remove From Project*”. Una vez hecho esto descargue el programa en la tarjeta y compruebe que el funcionamiento es correcto.

5. Detección de flancos

Es muy frecuente tener que detectar la pulsación de un pulsador, por ejemplo para incrementar un contador o ejecutar alguna tarea. Aunque hay varias maneras de hacerlo, el código siguiente es apropiado:

```
1  #include <xc.h>
2
3  #define PIN_PULSADOR 5
4
5  int main(void)
6  {
7      int pulsador_ant, pulsador_act;
8
9      TRISC = XX; // A completar por el alumno (trabajo previo)
10     LATC  = XX; // A completar por el alumno (trabajo previo)
11
12     TRISB = XX; // A completar por el alumno (trabajo previo)
13
14     pulsador_ant = (PORTB >> PIN_PULSADOR) & 1;
15     while(1){
16         // Se lee el estado del pulsador
17         pulsador_act = (PORTB >> PIN_PULSADOR) & 1;
18         if( (pulsador_act != pulsador_ant) && (pulsador_act
19             == 0) ){
20             // Flanco de bajada en la patilla del pulsador
21         }
22         pulsador_ant = pulsador_act;
23     }
```

Basándose en el código anterior, escriba un programa (`main3.c`) que cuente el número de veces que se ha pulsado el pulsador y muestre dicho número, en binario, en los LEDs.

6. Diseño 2

Escriba un programa, basándose en los anteriores, para encender inicialmente el LED RC0 y en cada pulsación del pulsador, ir desplazando el LED encendido hacia la izquierda. La secuencia a generar será la siguiente (recuerde que los LEDs son activos a nivel bajo):

```
1110
1101
1011
0111
1110
. . . .
```