



## Práctica 2 – Entrada y salida en C

### Manejo de puertos:

```
1  #include <xc.h>
2
3  int main(void)
4  {
5      // se ponen los LEDs como salida
6      TRISC = ~0x0F;
7
8      // los LEDs son activos a nivel bajo
9      LATC = ~0x0F;
10
11     while(1)
12     {
13         return 0;
14     }
15
```

En este apartado se pide encender los 4 LEDs conectados a los bits menos significativos del puerto C, siendo los bits RC0 – RC3.

En este primer apartado se ha optado por no utilizar máscaras y escribir el puerto C directamente. No obstante, para los siguientes apartados si se va a trabajar con máscaras.

## Diseño 1:

### Encender LED RC2

```
1  #include <xc.h>
2
3  #define BIT_LED0 0
4  #define BIT_LED1 1
5  #define BIT_LED2 2
6  #define BIT_LED3 3
7
8  int main(void)
9  {
10     int puerto = 0x0000;
11
12     TRISC = ~0x0F;
13
14     //encendemos solo el LED 2
15     puerto |= (1 << BIT_LED0);
16     puerto |= (1 << BIT_LED1);
17     puerto &= ~(1 << BIT_LED2);
18     puerto |= (1 << BIT_LED3);
19     LATC = puerto;
20
21     while(1)
22     ;
23     return 0;
24 }
```

### Encender LEDs RC3 y RC2

```
1  #include <xc.h>
2
3  #define BIT_LED0 0
4  #define BIT_LED1 1
5  #define BIT_LED2 2
6  #define BIT_LED3 3
7
8  int main(void)
9  {
10     TRISC = ~0x0F;
11
12     puerto = 0x0000;
13     //encendidos LED 2 y LED 3
14     puerto |= (1 << BIT_LED0);
15     puerto |= (1 << BIT_LED1);
16     puerto &= ~(1 << BIT_LED2);
17     puerto &= ~(1 << BIT_LED3);
18     LATC = puerto;
19
20     while(1)
21     ;
22     return 0;
23 }
```

## Manejo del pulsador:

```
1  #include <xc.h>
2
3  #define PIN_PULSADOR 5
4
5  int main(void)
6  {
7      int pulsador;
8
9      TRISC = ~0x0F;
10     LATC = 0x0F;
11
12     TRISB = 0xFF;
13
14     while(1){
15         //Se lee el estado del pulsador
16         pulsador = ( PORTB >> PIN_PULSADOR ) & 1;
17         //Cuando se pulsa el pulsador se enciende el LED 0, al dejar de pulsar se apagan
18         if(pulsador==0){
19             LATC &= ~1;
20         }else{
21             LATC |= 1;
22         }
23     }
24
25     return 0;
26 }
```

## Detección de flancos:

```
1  #include <xc.h>
2
3  #define BIT_PULSADOR 5
4
5  int main(void)
6  {
7      int puerto = 0x0000;
8      int pulsador_ant, pulsador_act;
9
10     TRISC = ~0x0F;
11     LATC = 0x0F;
12
13     //todos los pines como entradas
14     TRISB |= (1 << BIT_PULSADOR);
15
16     //leemos el estado del pulsador
17     //con PORTB --> leemos el valor del puerto
18     //>> BIT_PULSADOR --> leemos valor en el campo del pulsador
19     pulsador_ant = (PORTB >> BIT_PULSADOR) & 1;
20     int contador = 0;
21
22     while(1){
23         pulsador_act = (PORTB >> BIT_PULSADOR) & 1;
24
25         //si el pulsador == 0 --> está pulsado
26         if((pulsador_act != pulsador_ant) && (pulsador_act == 0)){
27             //ponemos todos los campos a 0 --> se encienden los LEDs
28
29             if(contador > 15) { //PARA QUE VUELVA A EMPEZAR
30                 contador = 0;
31             }
32             contador = contador + 1;
33             //escribimos en el puerto el valor del contador para que se enciendan los LEDs
34             LATC = ~contador;
35         }
36         pulsador_ant = pulsador_act;
37     }
38
39     return 0;
40 }
```

## Diseño 2:

```
1  #include <xc.h>
2
3  #define BIT_PULSADOR 5
4  #define BIT_LED0 0
5  #define BIT_LED1 1
6  #define BIT_LED2 2
7  #define BIT_LED3 3
8  #define NUMERO_LEDS 4
9
10 int main(void)
11 {
12     int pulsador_ant, pulsador_act;
13
14     TRISC = ~0x0F;
15     LATC = 0x0F;
16
17     //todos los pines como entradas
18     TRISB |= (1 << BIT_PULSADOR);
19
20     //leemos el estado del pulsador
21     //con PORTB --> leemos el valor del puerto
22     //>> BIT_PULSADOR --> leemos valor en el campo del pulsador
23     pulsador_ant = (PORTB >> BIT_PULSADOR) & 1;
24     int contador = 0;
25
26     int puerto = 0x0000;
27     //encendemos solo el LED 2
28     puerto &= ~(1 << BIT_LED0);
29     puerto |= (1 << BIT_LED1);
30     puerto |= (1 << BIT_LED2);
31     puerto |= (1 << BIT_LED3);
32     LATC = puerto;
33
34     while(1){
35         pulsador_act = (PORTB >> BIT_PULSADOR) & 1;
36
37         //si el pulsador == 0 --> está pulsado
38         if((pulsador_act != pulsador_ant) && (pulsador_act == 0)){
39             //ponemos todos los campos a 0 --> se encienden los LEDs
40             contador = contador + 1;
41
42             if(contador < 5){
43                 //desplazamos un 0 a la derecha y ponemos un 0 en la posicion anterior
44                 LATC &= ~(1 << contador);
45                 //tenemos que hacer un
46                 asm(" NOP");
47                 LATC |= (1 << (contador - 1));
48             }
49             //si llegamos al final de la secuencia resetear al valor inicial
50             if(contador == 4) {
51                 contador = 0;
52                 LATC = 14;
53             }
54         }
55
56         pulsador_ant = pulsador_act;
57     }
58
59     return 0;
60 }
```