

Para demostrar el uso de interrupciones en este tipo de microcontroladores se presentan a continuación dos ejemplos. El primero de ellos simplemente repite el programa hola mundo. La configuración del *timer* se hace de tal manera que se produzcan interrupciones cada 128us.

$$t_{\text{interrupcion}} = (4 \times \text{pre-escala} \times 256 / f_{\text{osc}} = (4 \times 1 \times 256) / 8000000 = 128 \text{ us}$$

Con este valor, el parpadeo del LED se puede calcular del siguiente modo:

$$T = 2 \times t_{\text{interrupcion}} \times \text{MAX\_LED\_CNT} = 2 \times 128 \text{ us} \times 2000 = 512 \text{ ms}$$

Utilizando un registro sombra que actualiza GPIO periódicamente en la rutina principal y recordando que "main.h" es el mismo de las secciones anteriores, el código es el siguiente:

```
puertos.h
1  #ifndef PORT_H
2  #define PORT_H
3
4  #include "main.h"
5
6  #define LED_1 sGPIO.GP0
7  #define MAX_LED_CNT 2000
8
9  volatile union {
10     uint8_t      port;
11     struct {
12         unsigned GP0 : 1;
13         unsigned GP1 : 1;
14         unsigned GP2 : 1;
15         unsigned GP3 : 1;
16         unsigned GP4 : 1;
17         unsigned GP5 : 1;
18     };
19 } sGPIO;
20
21 /* prototipos */
22 void inicializar_puertos(void);
23 void inicializar_timers(void);
24 void inicializar_interrupciones(void);
25
26 #endif /* PORT_H */

puertos.c
1  #include "puertos.h"
2
3  void inicializar_puertos()
4  {
5      OSCCON = 0b01110001; /* IRCF = 111 -> OSC. a 8MHz */
6      CMCON0 |= 0b00000111; /* desactiva comparadores */
7      ANSEL  &= 0b11110000; /* puerto como E/S digitales */
8      GPIO   = 0b0000000; /* clear latches */
9      TRISIO = 0b111000; /* GP0, GP1 y GP2 salidas */
10 }
11
```

Nótese que el registro sombra *sGPIO* debe ser declarado como **volatile** y la variable *cnt* como **static**.

El segundo ejemplo en esta sección realiza el parpadeo de un LED conectado a GP0 mientras un pulsador conectado a GP5 controla el encendido de otro LED conectado a GP1. El control de rebotes se realiza mediante el uso de un contador.

```
main.c
1  #include "puertos.h"
2
3  void main (void)
4  {
5      inicializar_puertos();
6      inicializar_timers();
7      inicializar_interrupciones();
8
9      sGPIO.port = GPIO;
10
11     while (1)
12     {
13         if (PB_cambio && !PB_estado){
14             LED_1 = ~LED_1;
15             PB_cambio = 0;
16         }
17         GPIO = sGPIO.port;
18     }
19 }
20
21 /* rutina de servicio de interrupciones */
22 void interrupt isr(void)
23 {
24     static uint8_t cnt_tm0 = 0;
25     static uint8_t cnt_db = 0;
26     static uint16_t cnt_led2 = 0;
27
28     INTCONbits.T0IF = 0;
29
30     if (++cnt_tm0 == MAX_TM0_CNT){
31         cnt_tm0 = 0;
32         if (PB_1 == PB_estado)
33             cnt_db = 0;
34         else
35             if (++cnt_db == MAX_DB_CNT){
36                 PB_estado = !PB_estado;
37                 cnt_db = 0;
38                 PB_cambio = 1;
39             }
40     }
```