

$$T = (2^{16} - 10931) \cdot \frac{4}{8 \times 10^6} \cdot 8 \cdot 8 = 1965780 \mu s / (2^{16} - 53070) \cdot \frac{4}{28 \times 10^6} \cdot 12 \cdot 8 = 0.17096228$$



Escuela de Ingeniería Industrial

Grado en Ingeniería en Electrónica Industrial y Automática

Prácticas de Electrónica Digital y Microcontroladores

$$T = (12 + 1) \cdot \frac{1}{8 \times 10^6} \cdot 8 \cdot 8 = 904 \mu s$$

$$T = (238 + 1) \cdot \frac{1}{8 \times 10^6} \cdot 16 \cdot 16$$

PRÁCTICA 11

GESTIÓN DE INTERRUPCIONES

1. INTRODUCCIÓN

En la estructura de un microcontrolador existen un conjunto de recursos que permiten el acoplamiento de periféricos por Interrupción. Para que un diseñador de aplicaciones pueda utilizar el acoplamiento de periféricos por interrupción debe conocer y manejar los conceptos ligados a este tipo de acoplamiento, como son: **Petición de interrupción, Permiso de petición, Prioridad de atención, Borrado de petición.** Además, debe entender la **secuencia de operaciones ligadas a este tipo de acoplamiento.** En esta práctica se analizan y utilizan los recursos del PIC18F47Q10 para la gestión de interrupciones sobre la "Curiosity HPC".

2. OBJETIVOS DE APRENDIZAJE

Para el microcontrolador PIC18F47Q10:

- Identificar los periféricos con capacidad de pedir atención por interrupción.
- Comprender los conceptos ligados a la gestión de interrupciones.
- Identificar los recursos destinados a la gestión de interrupciones.
- Comprender la estructura de un programa informático que realice el acoplamiento de periféricos por interrupción.

3. TAREAS PREVIAS A LA REALIZACIÓN DE LA PRÁCTICA

Para una preparación adecuada de la práctica, el alumno debe realizar las siguientes tareas previas:

- a) Leer y entender el **tema correspondiente** a los recursos del PIC18F47Q10 destinados a **la gestión de interrupciones.**
- b) **Identificar los periféricos con posibilidad de acoplamiento por interrupción en el PIC18F47Q10.**
- c) Realizar el cuestionario de evaluación.
- d) Preparar una propuesta de listados de programa que respondan a los diagramas de flujo de cada una de las tareas de la práctica y traerlos a la sesión de prácticas en un formato que permita incorporarlos al ejercicio Moodle.

4. DESARROLLO DE LA PRÁCTICA

Tarea 11_1: Escribir y depurar un programa que apague y encienda el LED conectado al terminal 4 (LED D2) del PORTA cada vez que se active (PULSADOR_ON) el pulsador S1 conectado al terminal 4 del PORTB. **El acoplamiento del pulsador debe realizarse por interrupción.**

RA 4 → Salida

PB 4 → Entr

El programa debe responder al siguiente diagrama de flujo:

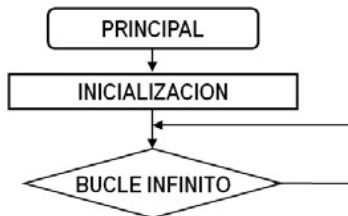


Figura 1. Programa principal tarea 11_1

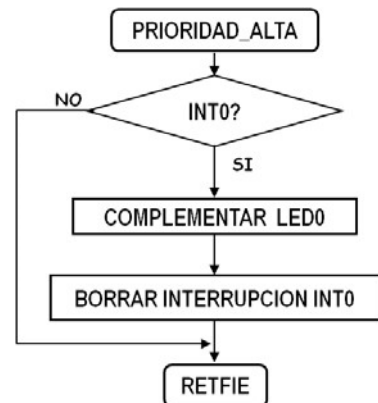


Figura 2. Diagrama de flujo de la rutina de atención interrupción en la tarea 11_1

En el bloque de inicialización se debe programar el modo de funcionamiento de los terminales de E/S (PORTB y PORTA) tal y como se realizó en la práctica 9 y **se deben inicializar los bits relacionados con la gestión de la interrupción de INT0 (terminal RB4 al que está conectado el pulsador)**. Tener en cuenta que en las inicializaciones se debe cambiar la INT0 a dicho terminal RB4, que por defecto se encuentra asociada al terminal 0:

- INT0EDG (INTCON.0): Selecciona flanco activo (de bajada) de petición de interrupción.
- INT0IE (PIE0.0): Permiso de interrupción externa (INT0)
- GIEH (INTCON.7): Permiso global de interrupciones de prioridad alta.

INT0PPS
 A|0
 B|1 + Puerto } RB4
 0000 1100 B

Tarea 11_2: Escribir un programa que utilice el temporizador TMR0 para eliminar el efecto de los rebotes del interruptor y que responda al siguiente diagrama de flujo:

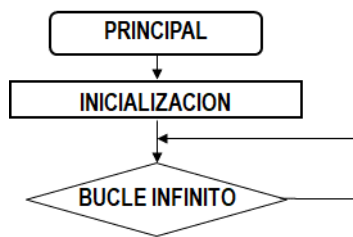


Figura 3. Diagrama de flujo del programa principal de la tarea 11_2

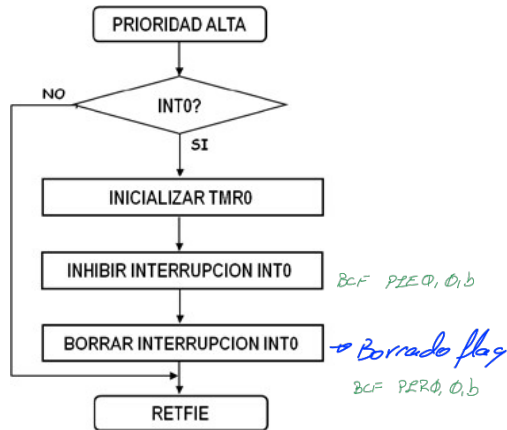


Figura 4. Diagrama de flujo de la rutina de interrupción del pulsador en la tarea 11_2

$$5ms = 5 \times 10^{-3}s = (2^{16} - TMR0) \cdot 4 \cdot 1\mu s.$$

$$TMR0 = 64286 \rightarrow TMR0H = 0x FB$$

$$PS = 1 \quad TMR0L = 0x 1E$$

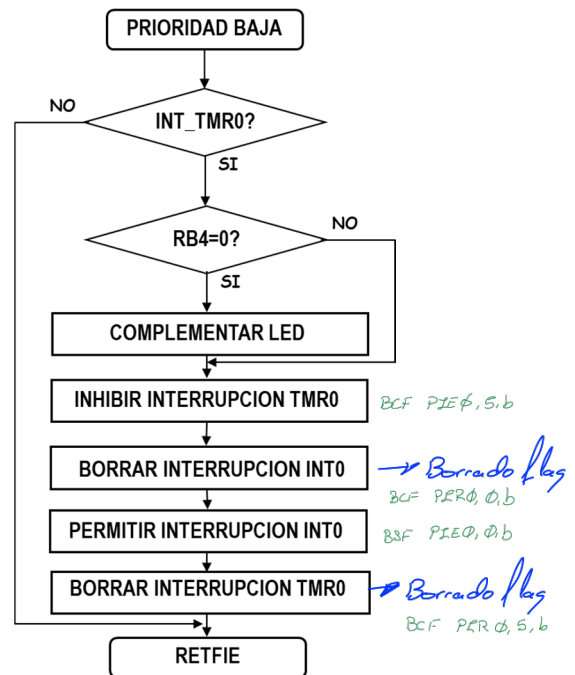


Figura 5. Diagrama de flujo de la rutina de atención de interrupción del Temporizador TMR0 en la tarea 11_2

En este caso debe programarse TMR0 para una temporización de 5ms y atender por interrupción el evento de rebosamiento del temporizador. Además, debe programarse prioridad baja vector 18h y la int0 a la prioridad alta vector 08h.

Tarea 11_3: Una vez comprobado el funcionamiento del programa, realizar los cambios necesarios para que ambas interrupciones tengan el mismo nivel de prioridad hardware y la prioridad se establezca por programa por el orden que se atienden en la subrutina de atención a la interrupción. Indicar el diagrama de flujo de la rutina de atención de interrupciones

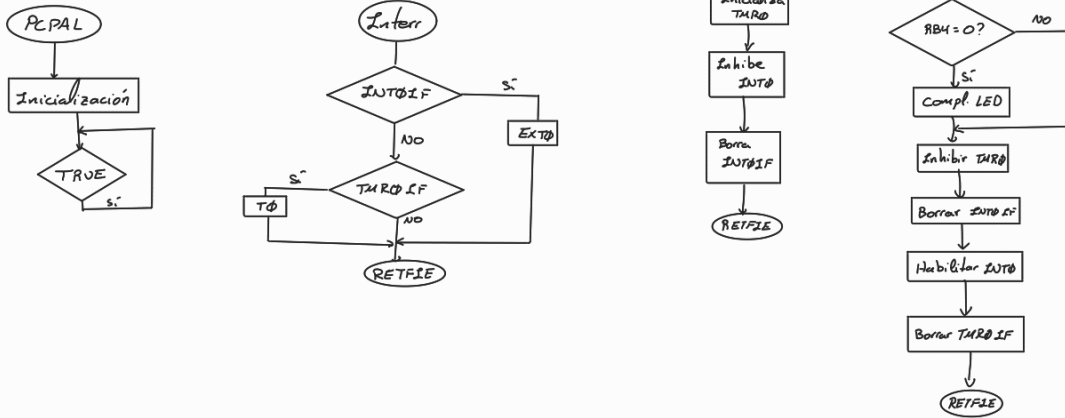
Tarea 11_4: Escribir y depurar un programa de acuerdo con las siguientes características. La asignación de cada trabajo será según el número de grupo.

- 1) Mostrar los LEDs incrementándose con una periodicidad de 0,5 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 0 por interrupción y el pulsador por consulta en el programa principal. No filtrar rebotes.
- 2) Mostrar los LEDs incrementándose con una periodicidad de 1 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 1 por interrupción y el pulsador por consulta en el programa principal. No filtrar rebotes.
- 3) Mostrar los LEDs incrementándose con una periodicidad de 2 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 2 por interrupción y el pulsador por consulta en el programa principal. No filtrar rebotes.

- 4) Mostrar los LEDs incrementándose con una periodicidad de 2 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 0 por consulta en el programa principal y el pulsador por interrupción. No filtrar rebotes.
- 5) Mostrar los LEDs incrementándose con una periodicidad de 0,5 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 1 por consulta en el programa principal y el pulsador por interrupción. No filtrar rebotes.
- 6) Mostrar los LEDs incrementándose con una periodicidad de 1 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 2 por consulta en el programa principal y el pulsador por interrupción. No filtrar rebotes.
- 7) Incrementar los LEDs después de 1 s de que se active el pulsador. Utilizar el Temporizador 0 por interrupción y los pulsadores por consulta en el programa principal.
- 8) Decrementar los LEDs después de 2 s de que se active el pulsador. Utilizar el Temporizador 1 por interrupción y los pulsadores por consulta en el programa principal.
- 9) Incrementar los LEDs después de 0,5 s de que se active el pulsador. Utilizar el Temporizador 2 por interrupción y los pulsadores por consulta en el programa principal.
- 10) Decrementar los LEDs después de 3 s de que se active el pulsador. Utilizar el Temporizador 0 por consulta en el programa principal y los pulsadores por interrupción en el programa principal.
- 11) Incrementar los LEDs después de 3 s de que se active el pulsador. Utilizar el Temporizador 1 por consulta en el programa principal y los pulsadores por interrupción.
- 12) Decrementar los LEDs después de 3 s de que se active el pulsador. Utilizar el Temporizador 2 por consulta en el programa principal y los pulsadores por interrupción.

Introducir en el ejercicio Moodle en la Hoja de Respuestas el diagrama de flujo del programa diseñado.

Tarea 11.3



Tarea 11.4

- 6) Mostrar los LEDs incrementándose con una periodicidad de 1 s. El pulsador S1 cambia de incrementar a decrementar y viceversa cada vez que se pulsa. Utilizar el Temporizador 2 por consulta en el programa principal y el pulsador por interrupción. No filtrar rebotes.

$$1s = 10^6 \mu s = (PR2 + 1) \cdot 1 \mu s \cdot 4 \cdot 128 \cdot 8$$

$$PR2 = 243$$

$$PS = 128$$

$$PD = 8$$

$$10^6 \mu s = (PR2 + 1) \cdot 1 \mu s \cdot 4 \cdot 128 \cdot 16$$

$$PR2 = 121$$

$$PS = 128$$

$$PD = 16$$

Tarea 11.4

