

# Interrupciones de rango medio mejoradas

---

## Resumen

Las interrupciones son eventos detectados por la MCU que hacen que se anule el flujo normal del programa. Las interrupciones pausan el programa actual y transfieren el control a una rutina de firmware específica escrita por el usuario llamada Rutina de servicio de interrupción (ISR). El ISR procesa el evento de interrupción y luego reanuda el flujo normal del programa. Este artículo muestra cómo habilitar y procesar interrupciones en la familia PIC16F1xxx de PIC® de rango medio <sup>mejorado</sup>.

---

## Descripción general del proceso de interrupción

### 1 Programa MCU para reaccionar a las interrupciones

La MCU debe programarse para permitir que ocurran interrupciones. La configuración de Global Interrupt Enable (GIE) y, en muchos casos, Peripheral Interrupt Enable (PEIE), permite que la MCU reciba interrupciones. GIE y PEIE se encuentran en el registro de funciones especiales de control de interrupciones (INTCON).

### 2 Habilitar interrupciones de periféricos seleccionados

Cada periférico en la MCU tiene un bit de habilitación individual. Se debe establecer el bit de activación de interrupción individual de un periférico, además de GIE/PEIE, antes de que el periférico pueda generar una interrupción. Los bits de habilitación de interrupción individuales se encuentran en INTCON, PIE1, PIE2 y PIE3.

### 3 Periférico afirma una solicitud de interrupción

Cuando un periférico alcanza un estado en el que se necesita la intervención del programa, el periférico establece un indicador de solicitud de interrupción (xxIF). Estos indicadores de interrupción se establecen independientemente del estado de GIE, PEIE y los bits de activación de interrupción individuales. Los indicadores de interrupción se encuentran en INTCON, PIR1, PIR2 y PIR3.

Los indicadores de solicitud de interrupción se bloquean en alto cuando se establecen y deben ser borrados por el ISR escrito por el usuario.

#### **4 Ocurre una interrupción**

Cuando se establece un indicador de solicitud de interrupción y la interrupción está habilitada correctamente, comienza el proceso de interrupción:

- Las interrupciones globales se desactivan borrando GIE a 0.
- El contexto del programa actual se guarda en los registros de sombra.
- El valor del contador de programa se almacena en la pila de retorno.
- El control del programa se transfiere al vector de interrupción en la dirección 04h.

#### **5 Ejecuciones de ISR**

El ISR es una función escrita por el usuario y ubicada en la dirección 04h. El ISR hace lo siguiente:

1. Comprueba los periféricos habilitados para interrupción en busca del origen de la solicitud de interrupción.
2. Realiza las tareas periféricas necesarias.
3. Borra el indicador de solicitud de interrupción correspondiente.
4. Ejecuta la instrucción Return From Interrupt (RETFIE) como instrucción ISR final.

#### **6 El control vuelve al programa principal**

Cuando se ejecuta RETFIE:

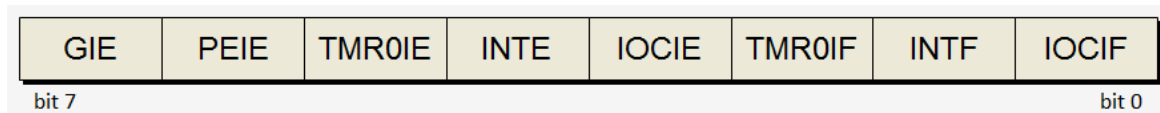
1. Las interrupciones globales están habilitadas (GIE=1).
2. El contexto del programa se restaura desde los registros de sombra.
3. La dirección de retorno de la pila se carga en el contador de programa.
4. La ejecución se reanuda desde el punto en que se interrumpió.

---

## **Registros utilizados para procesar interrupciones**

## Registro de control de interrupciones

registro INTCON



(/local--files/8bit:emr-interrupts/intcon-reg.png)

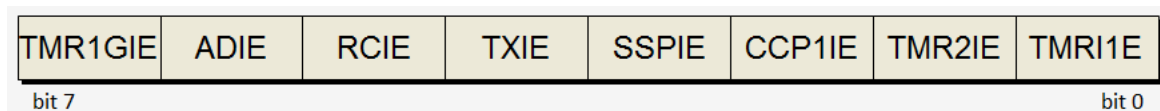
**GIE** - Habilitación de interrupción global **PEIE** - Habilitación de interrupción periférica **TMR0IE** - Habilitación de interrupción del temporizador 0 **INTE** - Habilitación de interrupción externa **IOCF** - Habilitación de interrupción por cambio **TMR0IF** - Indicador de interrupción del temporizador 0 **INTF** - Indicador de interrupción externa **IOCF** - Indicador de interrupción por cambio

☐  
☐  
☐  
☐  
☐  
☐  
☐

**INTCON** contiene banderas de habilitación de interrupción global y periférica, así como banderas de solicitud de interrupción individuales y banderas de habilitación de interrupción para tres de las interrupciones PIC16F1xxx.

## Registros de habilitación de interrupción

registro PIE1



(/local--files/8bit:emr-interrupts/pie1-reg.png)

**TMR1GIE** - Habilitación de interrupción de puerta del temporizador 1 **ADIE** - Habilitación de interrupción de convertidor analógico a digital (ADC) **RCIE** - Habilitación de interrupción de recepción de transmisor receptor asíncrono síncrono universal (USART) **TXIE** - Habilitación de interrupción de transmisión USART **SSPIE** - Habilitación de interrupción de puerto serie síncrono (MSSP) **CCP1IE** - Habilitación de interrupción **CCP1** **TMR2IE** - Habilitación de interrupción del temporizador 2 **TMR1IE** - Habilitación de interrupción

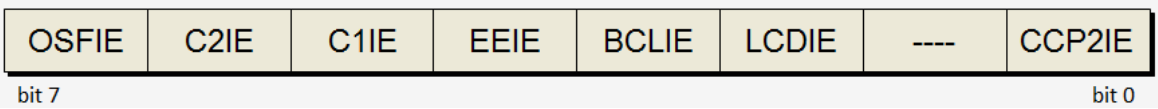
☐  
☐  
☐  
☐

☐

☐

☐

registro PIE2



(/local--files/8bit:emr-interrupts/pie2-reg.png)

☐ **OSFIE** - Habilitación de interrupción por falla del oscilador ☐ **C2IE** - Habilitación de interrupción del comparador C2 ☐ **C1IE** - Habilitación de interrupción del comparador C1 ☐ **EEIE** - Habilitación de interrupción de finalización de escritura de EEPROM ☐ **BCLIE** - Habilitación de interrupción de colisión de bus MSSP ☐ **LCDIE** - Habilitación de interrupción del módulo LCD ☐ --- ☐ - Sin implementar, se lee como 0 ☐ **CCP2IE** - CCP2 Habilitar interrupción

☐

☐

☐

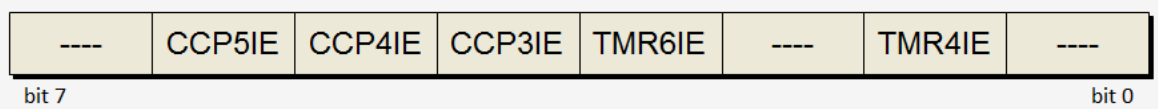
☐

☐

☐

☐

registro PIE3



(/local--files/8bit:emr-interrupts/pie3-reg.png)

☐ --- ☐ - No implementado leído como 0 ☐ **CCP5IE** - CCP5 Interrupt Enable ☐ **CCP4IE** - CCP4 Interrupt Enable ☐ **CCP3IE** - CCP3 Interrupt Enable ☐ **TMR6IE** - Timer6 Interrupt Enable ☐ --- ☐ - No implementado, leído como 0 ☐ **TMR4IE** - Timer4 Interrupt Enable ☐ --- ☐ - No implementado, leído como 0

☐

☐

☐

☐

☐

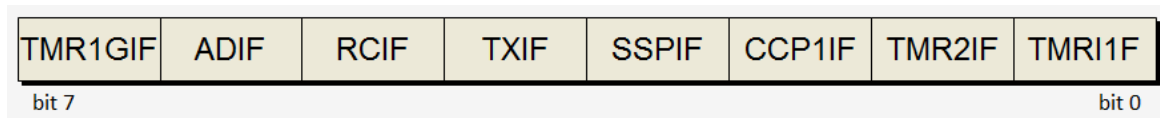
☐

☐

☐ **PIE1** , ☐ **PIE2** y ☐ **PIE3** contienen los indicadores de activación de interrupción individuales para los periféricos de la MCU.

## Registros de solicitud de interrupción

### registro PIR1

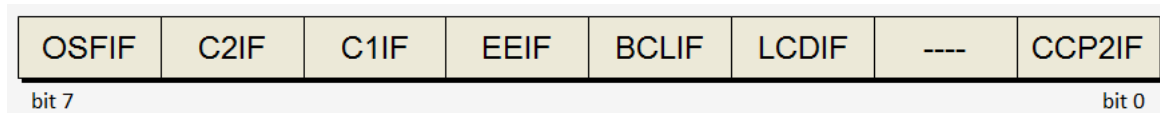


(/local--files/8bit:emr-interrupts/pir1-reg.png)

☐ **TMR1GIF** - Indicador de interrupción de puerta del temporizador 1
 ☐ **ADIF** - Indicador de interrupción de ADC
 ☐ **RCIF** - Indicador de interrupción de recepción de USART
 ☐ **TXIF** - Indicador de interrupción de transmisión de USART
 ☐ **SSPIF** - Indicador de interrupción de MSSP
 ☐ **CCP1IF** - Indicador de interrupción de CCP1
 ☐ **TMR2IF** - Indicador de interrupción de temporizador 2
 ☐ **TMR1IF** - Indicador de interrupción

☐  
☐  
☐  
☐  
☐  
☐  
☐

### registro PIR2

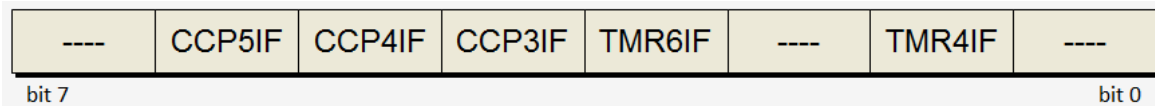


(/local--files/8bit:emr-interrupts/pir2-reg.png)

☐ **OSFIF** - Indicador de interrupción por falla del oscilador
 ☐ **C2IF** - Indicador de interrupción C2 del comparador
 ☐ **C1IF** - Indicador de interrupción C1 del comparador
 ☐ **EEIF** - Indicador de interrupción de finalización de escritura de EEPROM
 ☐ **BCLIF** - Indicador de interrupción de colisión del bus MSSP
 ☐ **LCDIF** - Indicador de interrupción del módulo LCD
 ☐ --- - Sin implementar, se lee como 0
 ☐ **CCP2IF** - CCP2 Indicador de interrupción

☐  
☐  
☐  
☐  
☐  
☐  
☐

### registro PIR3



(/local--files/8bit:emr-interrupts/pir3-reg.png)

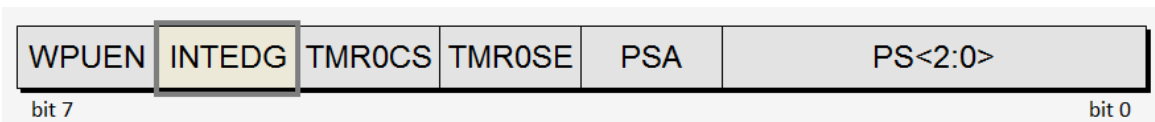
--- - Sin implementar, leer como 0 **CCP5IF** - Bandera de interrupción de CCP5  
**CCP4IF** - Bandera de interrupción de CCP4 **CCP3IF** - Bandera de interrupción de CCP3  
**TMR6IF** - Bandera de interrupción de Timer6  
 --- - No implementada, leer como 0 **TMR4IF** - Bandera de interrupción de Timer4  
 --- - No implementada, leer como 0



**PIR1**, **PIR2** y **PIR3** contienen los indicadores de solicitud de interrupción individuales para los periféricos de la MCU.

## OPCIÓN\_REG

### OPCIÓN\_REG



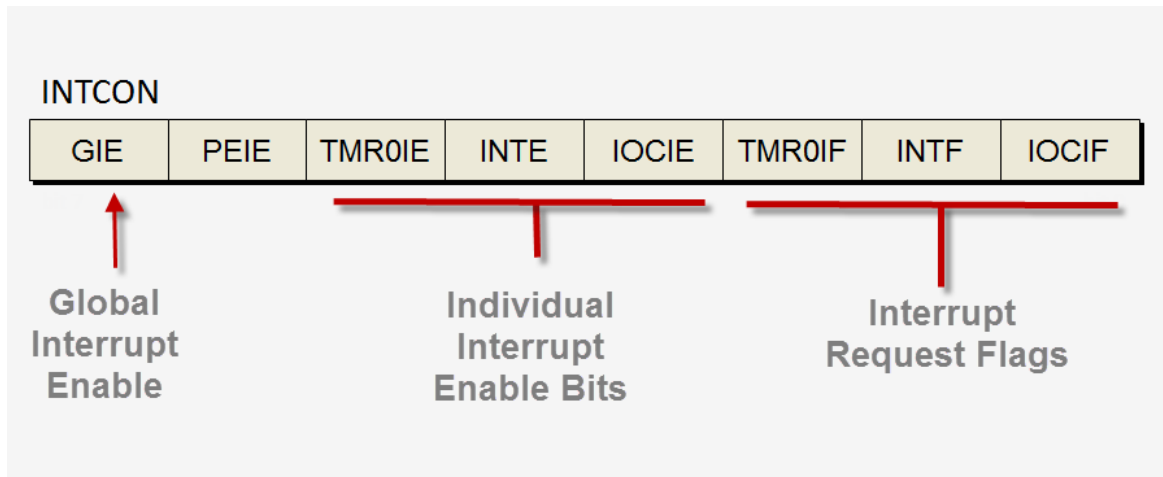
(/local--files/8bit:emr-interrupts/option-reg.png)

El indicador **INTEDG** en **OPTION\_REG** se usa para establecer un flanco ascendente o descendente en el pin **INT** como desencadenante de una interrupción **INTE**.

## Habilitación de interrupciones

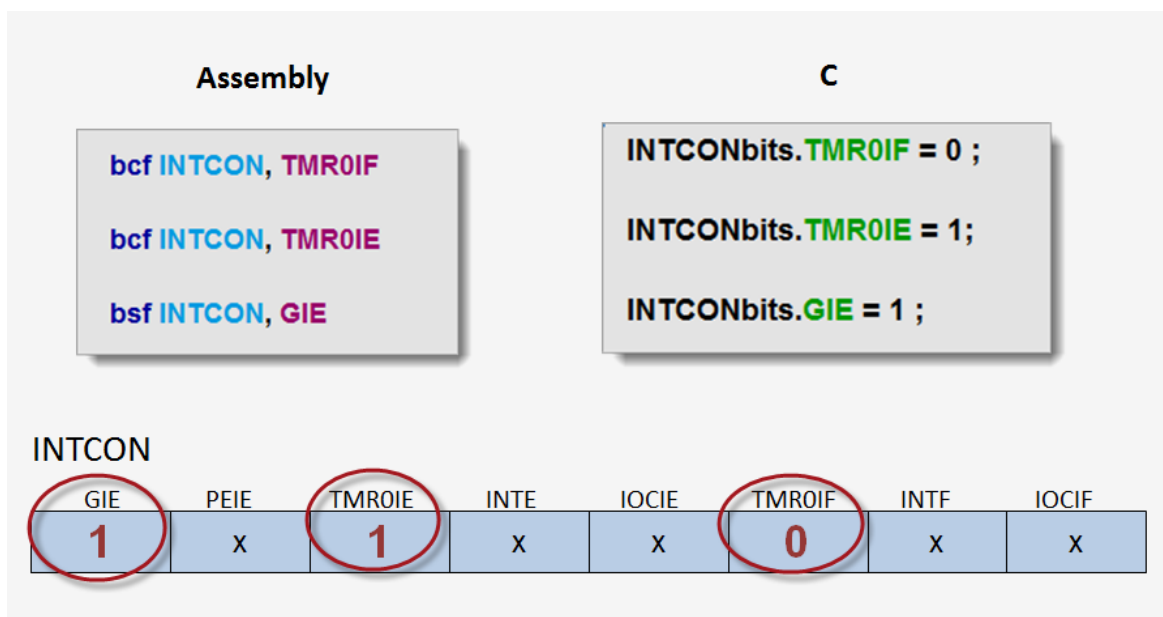
### Interrupciones del núcleo

Tres fuentes de interrupción (Timer0, External Interrupt e Interrupt on Change) tienen bits de habilitación de interrupción ubicados en **INTCON**. Estas interrupciones se denominan interrupciones centrales.



(/local--files/8bit:emr-interrupts/core-enable-reg.png)

Para habilitar una de las interrupciones principales, solo es necesario establecer el bit de habilitación de interrupción individual y el GIE.



(/local--files/8bit:emr-interrupts/core-enable-code.png)



Borrar un indicador de solicitud de interrupción antes de establecer el indicador de activación de interrupción evita que cualquier solicitud de interrupción pendiente active una interrupción inmediata.

## Interrupciones Periféricas

Los periféricos PIC16F1xxx, capaces de generar solicitudes de interrupción cada uno, tienen sus indicadores de activación de interrupción en uno de los tres registros PIE. Para habilitar una interrupción periférica, el indicador de interrupción individual, GIE y PEIE deben estar todos configurados.

Assembly

```
banksel PIR1
bsf PIR1, SSPIF
banksel PIE1
bsf PIE1, SSPIE

bsf INTCON, PEIE
bsf INTCON, GIE
```

C

```
PIR1bits.SSP1IF = 0 ;

PIE1bits.SSP1IE = 1;

INTCONbits.PEIE = 1;

INTCONbits.GIE = 1 ;
```

INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF
	1	1	X	X	X	X	X	X
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCPIE	TMR2IE	TMR1IE
	X	X	X	X	1	X	X	X
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCPIF	TMR2IF	TMR1IF
	X	X	X	X	0	X	X	X

(/local--files/8bit:emr-interrupts/peie-code.png)

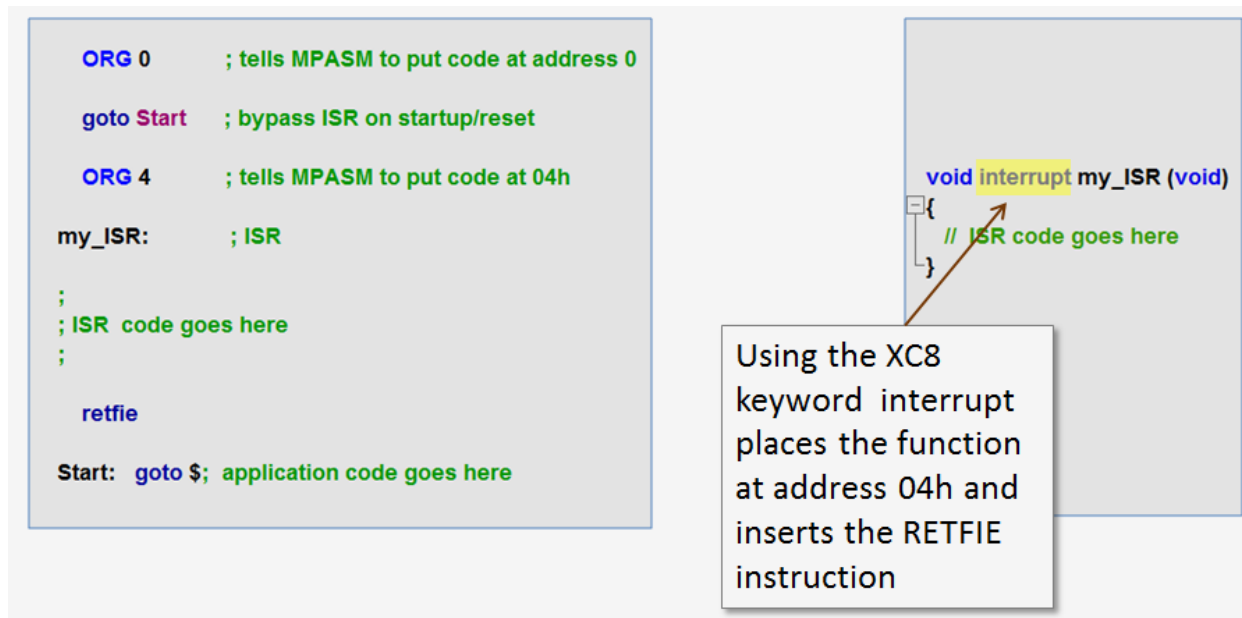
## Dar servicio a una interrupción

### ISR

El ISR es un programa escrito por el usuario que realiza las tareas necesarias cuando ocurre una interrupción. El usuario es responsable de escribir el ISR y colocarlo en la dirección 04h. La última instrucción ejecutada por el ISR debe ser la instrucción RETFIE. Los ISR se pueden escribir en lenguaje C o ensamblador.

#### ISR en ensamblaje (izquierda) e ISR en C (derecha)

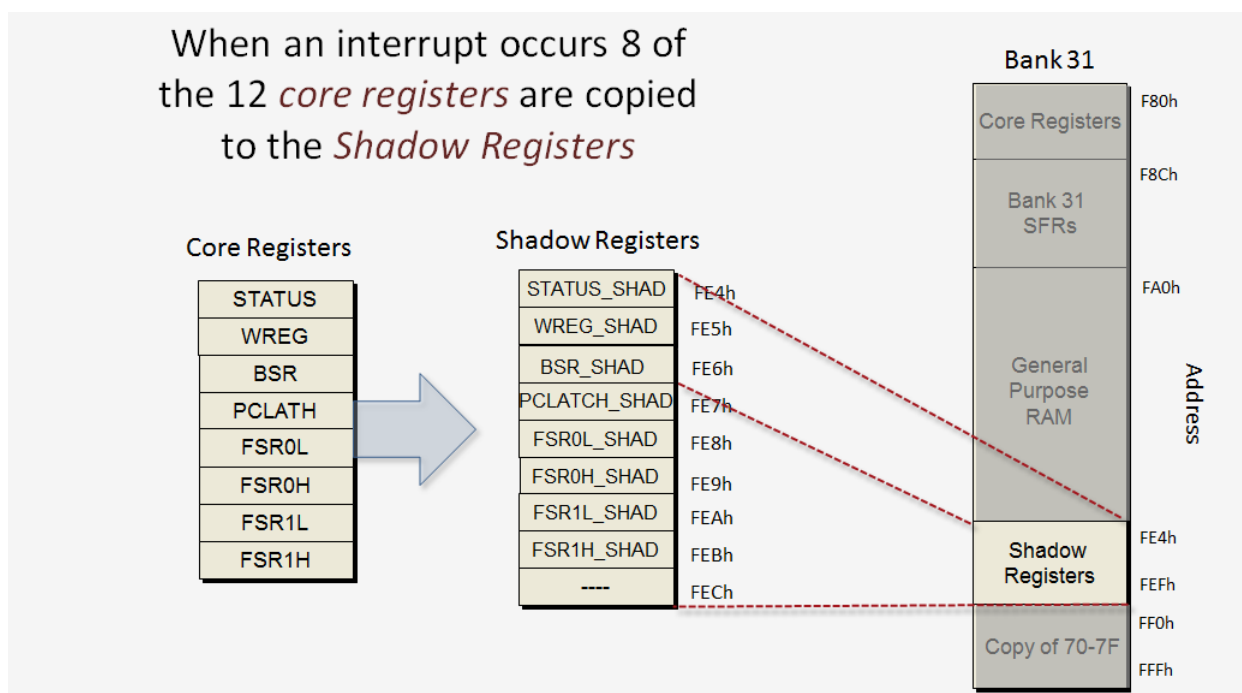




(/local--files/8bit:emr-interrupts/isr-location.png)

## Guardar contexto

El hardware de la MCU invocará el mecanismo de ahorro de contexto cuando ocurra una interrupción.



(/local--files/8bit:emr-interrupts/context-saving.png)

## Verificar fuente de la interrupción

Hay un ISR que da servicio a todas las interrupciones de la aplicación. El ISR necesita verificar secuencialmente los indicadores de solicitud de interrupción individuales para determinar la fuente de la interrupción.

## Borrado del indicador de solicitud de interrupción

Los indicadores de solicitud de interrupción se bloquean cuando los establece el periférico. Deben ser autorizados por ISR. Si el ISR no restablece un indicador de solicitud, se producirá otra interrupción inmediatamente después de que el ISR devuelva el control al programa principal.

```
void interrupt sample_isr (void)
{
    if (TMR2IF == 1)
    {
        // perform TMR2 Interrupt task
        TMR2IF = 0 ;
    }

    if (INTE == 1)
    {
        // External Interrupt tasks
        INTF = 0 ;
    }

    if (CCP1IF == 1)
        // CCP tasks
        CCP1IF = 0;
}
```

(/local--files/8bit:emr-interrupts/sample-isr.png)

---

## Ejemplo de código de interrupción

### Procesando una interrupción del Timer2

La siguiente animación muestra el código para configurar y procesar una interrupción del Timer2 en el MCU de rango medio mejorado PIC16F1xxx. Los íconos de control en la parte inferior de la animación permiten pausar la visualización y realizar un solo paso.

 [Imagen de presentación de diapositivas](#)

⏮ ⏪ ⏩ ⏭ 1 / 0

Se puede encontrar una explicación más detallada de las interrupciones PIC16F1xxx en la página Interrupciones (/mcu1102:interrupts) .

La página Timer2/4/6 (/8bit:timer2) proporciona los detalles sobre la configuración y el funcionamiento de Timer2.

---