Práctica Número 3:

"Manejo de la Interrupción Externa IRQ"

Objetivo

Comprender y aplicar el funcionamiento de la interrupción externa IRQ, usando ProcessorExpert.

Material y equipo para la práctica

1 Multímetro.
1 PC.
1 Tarjeta DEMOJM60.
1 Microcontrolador MC9S08JM60.
1 sensor infrarrojo CNY70.
1 resistencia de 330 Ω.
1 resistencia de 15 KΩ.
1 Protoboard.
Cables.

Cuestionario Preliminar

1. ¿Qué es una interrupción por software?
2. ¿Qué es una interrupción por hardware?
3. ¿Qué es una bandera?
4. ¿En qué situaciones son requeridas las interrupciones externas?
5. ¿Qué es y para que sirve el Processor Expert de CodeWarrior?
6. ¿Cómo se atiende una interrupción en lenguaje C?
7. ¿Qué sucede si no borramos la bandera que originó la atención de la interrupción?

Introducción

Interrupción también conocida como interrupción de hardware o petición de interrupción es una señal recibida por el procesador de un ordenador, indicando que debe "interrumpir" el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

Las interrupciones, atendiendo a su duración, pueden dividirse en dos tipos claramente diferentes: excepciones que se producen a lo largo de un intervalo de tiempo, y excepciones puntuales. La forma en que se notifican estas excepciones hace que las denomine excepciones por flanco y por nivel, respectivamente:

- Excepciones ***por flanco***: estas son puntuales, que se producen en un instante concreto y determinado. La línea asociada a estas indicará el instante en que se producen con un flanco activo (de bajada o de subida). Es importante tener en cuenta que el nivel de la línea de excepción no es relevante, solo un flanco activo marca la ocurrencia de una por flanco.

- Excepciones ***por nivel***: estas no son puntuales, sino que se prolongan a lo largo de un intervalo de tiempo, dicho de otro modo, permanecen activas durante un cierto tiempo. A la hora de comprobar si la excepción se produce o no, se ha de considerar el nivel de la línea asociada ha dicho evento. Mientras esta línea esté activa, la excepción se estará produciendo.

El principal inconveniente de las excepciones por flanco es que, si se producen en un instante en que no pueden ser atendidas, por ejemplo, porque se esté procesando otra excepción de más prioridad, requieren el uso de un hardware adicional que permita registrar dicha excepción, para procesarla posteriormente. Las excepciones por nivel, como se producen a lo largo del intervalo de tiempo, son menos sensibles a este problema.

Funcionamiento

El registro IRQSC es el encargado de manipular la señal externa IRQ, que ocupa un lugar privilegiado dentro de los niveles y prioridades en la máquina. El procesador monitorea permanentemente la lógica del pin IRQ y puede validar tanto el flanco, y nivel de la señal presente en el pin.

La interrupción generada por el pin IRQ puede sacar a la máquina en el modo STOP, aún con el reloj suspendido. Un "0" lógico aplicado al pin de la interrupción externa produce un evento de interrupción. El pin IRQ es configurable por software, activado por flanco descendente o ascendente y nivel bajo.

Es necesario tener en cuenta que se debe atender la interrupción y salir de ella antes de que el pin de interrupción retorne a "1" lógico, siempre y cuando se tenga configurado como flanco y nivel lógico bajo.

- ***Registro IRQSC***

	Bit 7	6	5	4	3	2	1	Bit 0
Lectura	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
Escritura						IRQACK		
Reset	0	0	0	0	0	0	0	0

= No implementado o reservado

IRQPDD (IRQ Pull Device Disable)
Deshabilita la resistencia de *pull-up/pull-down* asociada al pin IRQ.

IRQPDD = 0: La resistencia está habilitada.
IRQPDD = 1: La resistencia esta deshabilitada.

IRQEDG (IRQ Edge Select)
Configura el flanco que activará la interrupción externa IRQ.

IRQEDG = 0: El pin IRQ será sensible al flanco de bajada o flanco de baja y nivel bajo (de existir una resistencia asociada al pin, deberá ser de *pull-up*).
IRQEDG = 1: El pin IRQ será sensible al flanco de subida o flanco de subida y nivel alto de existir una resistencia asociada al pin, deberá ser de *pull-down*.

IRQPE (IRQ Pin Enable)
Habilita el funcionamiento del pin IRQ.

IRQPE = 0: El pin IRQ ha sido deshabilitado.
IRQPE = 1: El pin IRQ ha sido habilitado.

IRQF (IRQ bandera)
Bandera que indica que ha ocurrido un evento de interrupción externa IRQ.

IRQF = 0: No hay evento de IRQ.
IRQF = 1: Ha ocurrido un evento de IRQ.

IRQACK (IRQ Acknowledge)
Bit para el reconocimiento de un evento de IRQ.

Escribiendo un “1” en este bit, se pone a “0” la bandera IRQF. El usuario deberá poner a cero este bit en la atención al evento, siempre y cuando el modo de operación no se encuentre en flanco y nivel (IRQMOD=1) y esté activo.

IRQACK = 0: No tiene efecto.
IRQACK = 1: bandera de IRQF con IRQMOD=0.

IRQIE (IRQ Interrupt Enable)

Habilita un evento de interrupción, ante la aparición de un evento de IRQ.

IRQIE = 0: No se habilita evento de interrupción por IRQ. El usuario puede usar la lectura iterativa (*polling*) sobre la bandera IRQF, siempre y cuando el pin esté habilitado.
IRQIE = 1: Habilita el evento de interrupción por IRQ.

IRQMOD (IRQ Mode)
Elige el modo de operación eléctrico, con el pin de IRQ.

IRQMOD = 0: El pin de IRQ sólo actúa con el flanco de la señal eléctrica presente.
IRQMOD = 1: El pin de IRQ actúa tanto en el flanco como en el nivel de la señal eléctrica.

Desarrollo

La figura 3.21 detalla el circuito a implementar de la práctica número 3.

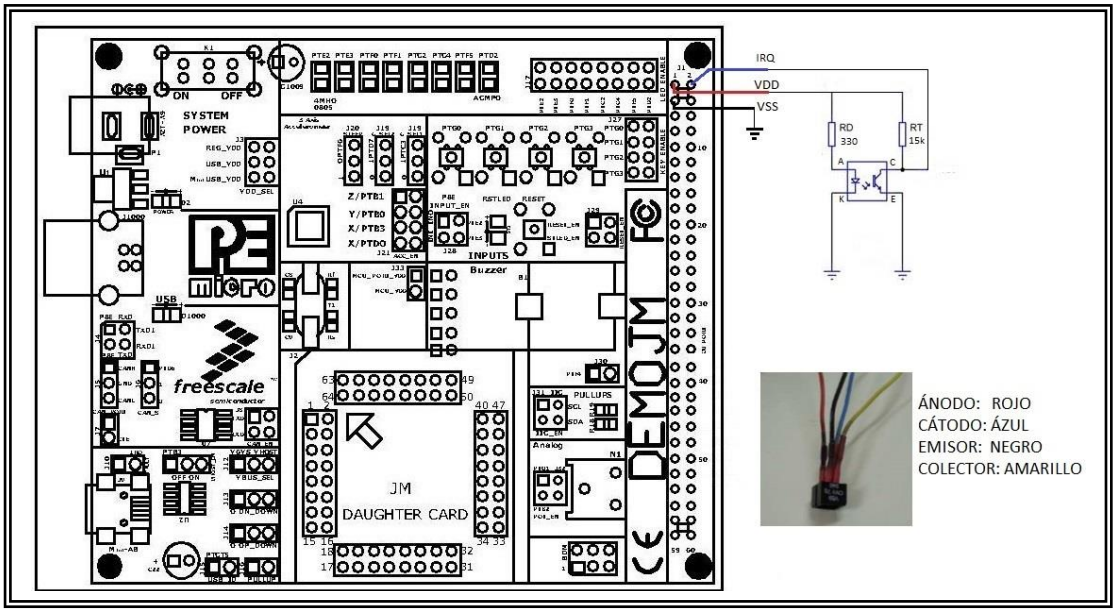
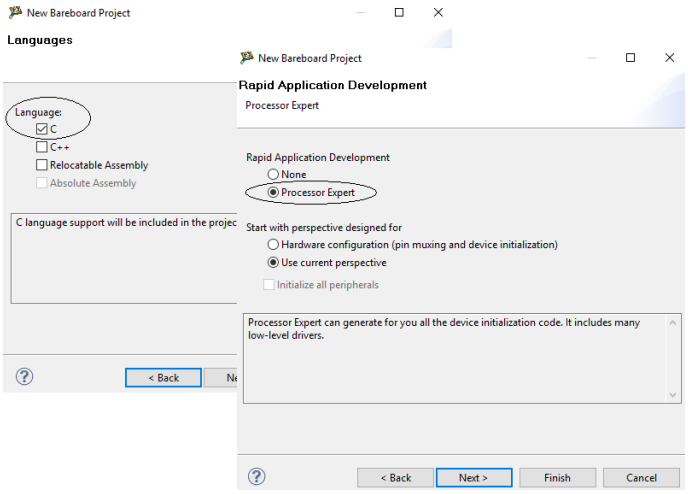
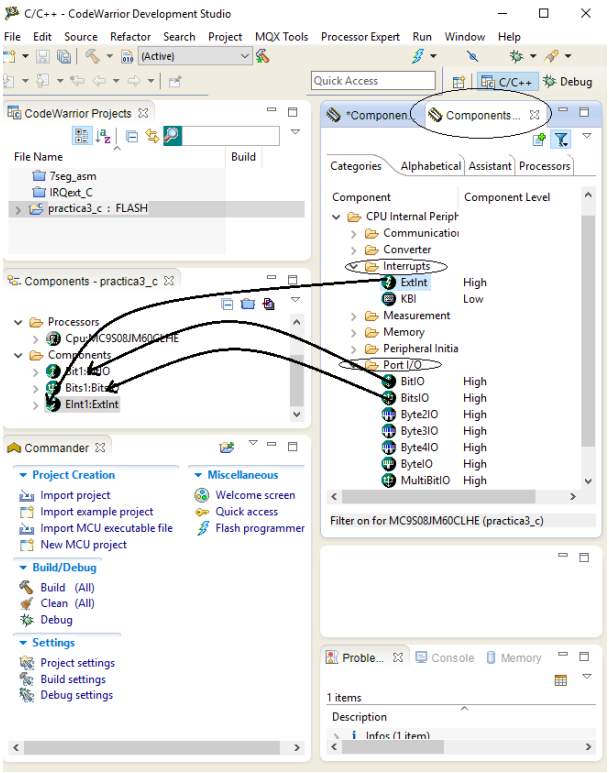


Figura 3.21 Circuito como ejercicio del manejo de la interrupción externa IRQ.

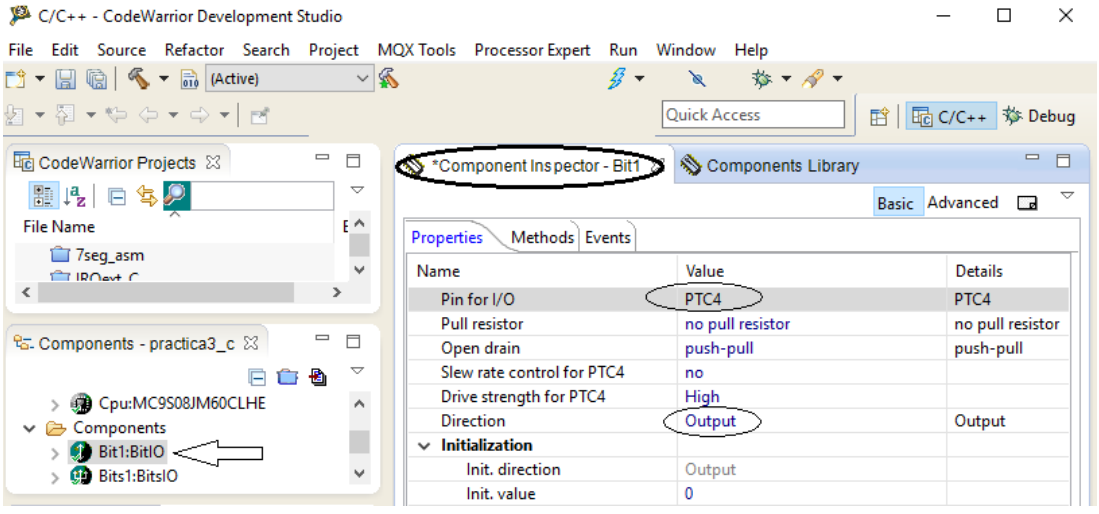
1. Crear un nuevo proyecto en CodeWarrior (Eclipse) seleccionando el dispositivo MC9S08JM60 para ser conectado con P&E USB Multilink y seleccionar **Lenguaje C**, y seleccionar **Processor Expert**.



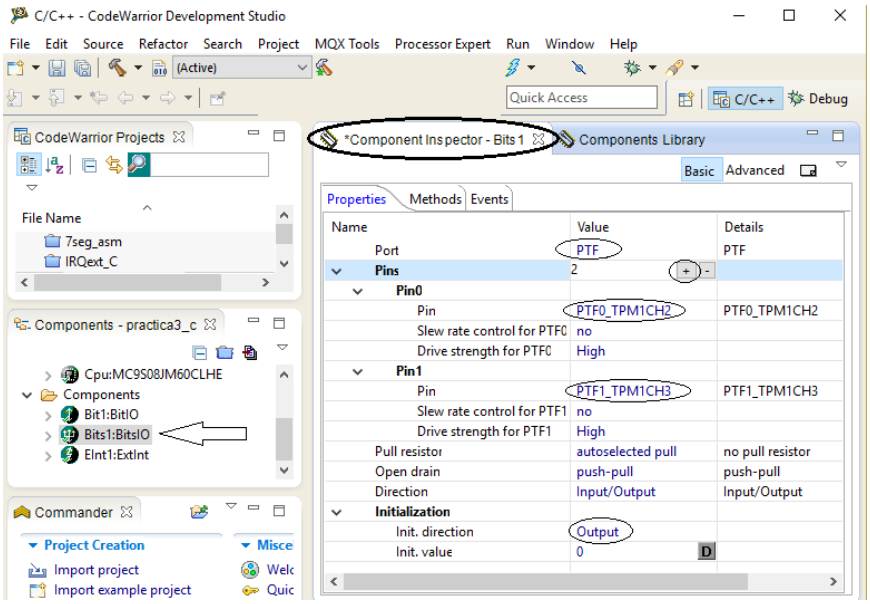
2. Agregar los siguientes componentes: *BitIO, BitsIO y ExtInt*. Haciendo clic en el folder ***Components Library*** y abriendo los folders ***Interrupts*** y ***Port I/O***, haciendo doble clic en los componentes indicados, o arrestándolos a la carpeta de componentes, como se muestra en la siguiente figura.



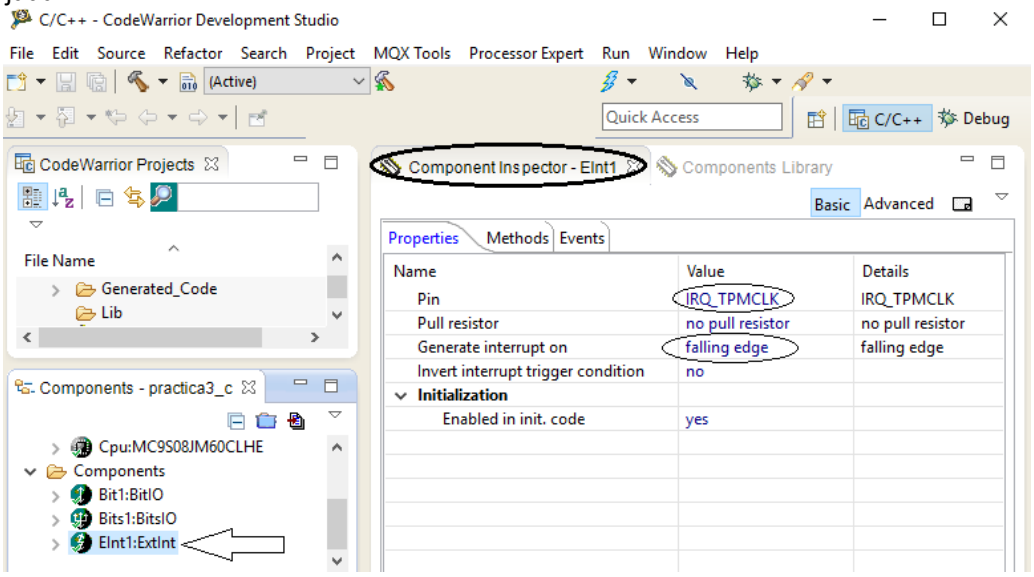
3. Asignar las propiedades de *Bit1:BitIO* para activar el Pin PTC4 como salida:



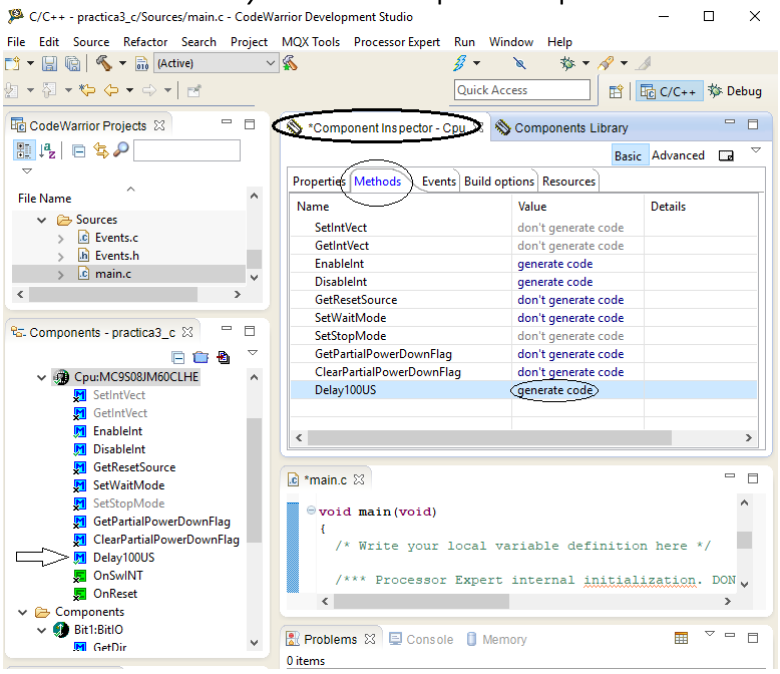
4. Asignar las propiedades de *Bits1:BitsIO* para activar los pines PTF0 y PTF1 como salida:



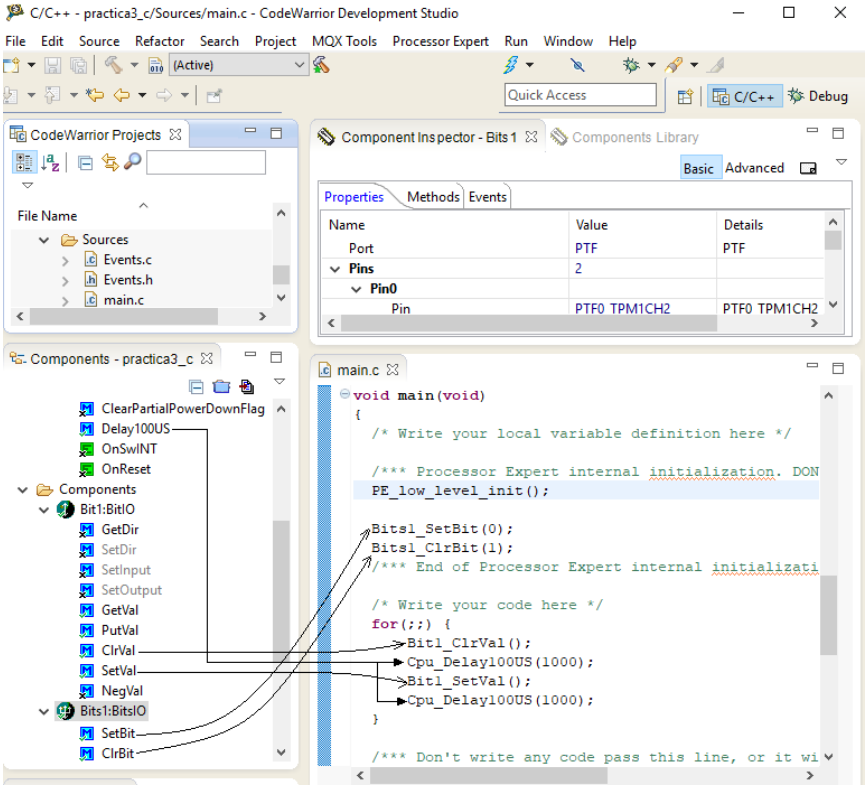
5. Asignar las propiedades de *EInt1:ExtInt* para activar la Interrupción Externa, con flanco de bajada.



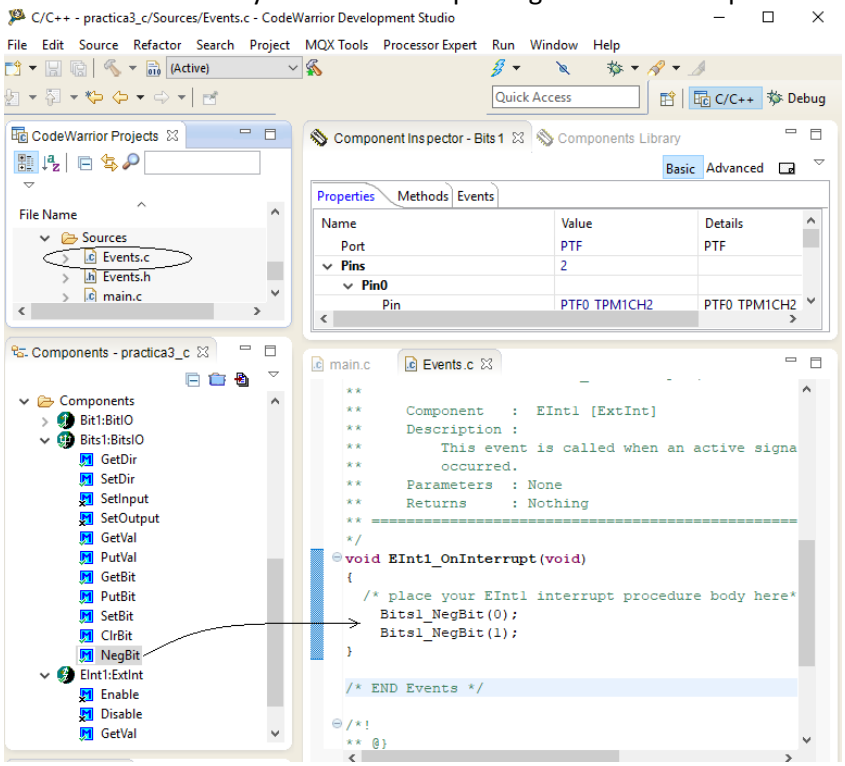
6. Activar la función de retardo *Delay100US* del componente Cpu:MC9S08JM60CLHE



7. Programar en ***main.c*** la inicialización de PTF0=0 y PTF1=1 y hacer un ciclo infinito que encienda y apague el led PTC4:



8. Programar en ***Events.c*** la interrupción dentro de la función `EInt1_OnInterrupt` que invierta el valor de PTF0 y PTF1 cada vez que se genera la interrupción externa.



Se muestra el diagrama de flujo para el manejo la interrupción externa IRQ:

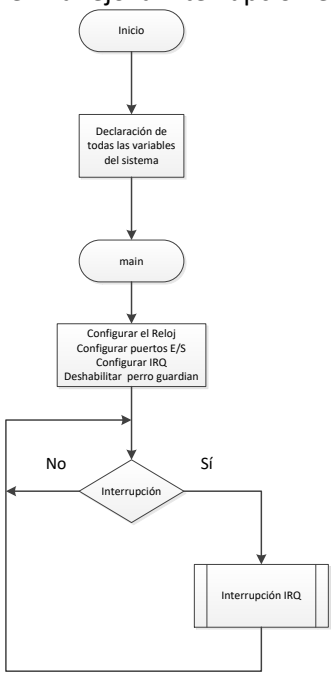


Figura 3.22 Configuración de módulos y periféricos.