

# INTERRUPCIONES EN LOS MICROCONTROLADORES

CHRISTIAN FONSECA  
christianfonseca864@gmail.com

## RESUMEN

*En el presente documento se expresa la información concerniente al manejo de interrupciones en los microcontroladores PIC como eventos programables para satisfacer requerimientos emergentes o que pueden ser influenciados mediante dispositivos ajenos al PIC y así sincronizar la ejecución interna con dichos acontecimientos externos; se detallan las definiciones y comportamientos de los sistemas utilizados mediante interrupciones, junto con los recursos aplicables, su funcionalidad ante procesos electrónicos y orientados en mayor magnitud al PIC16F628A.*

**PALABRAS CLAVE:** Interrupción, microcontroladores, programación, PIC, sincronización, sistemas, electrónico, PIC16F628A.

## ABSTRACT:

*This document expresses the information concerning the handling of interruptions PIC microcontrollers and programmable events to meet emerging requirements or that can be influenced by devices outside the PIC and so synchronize the internal execution with these external events; definitions and behaviors of the systems used by interruptions, along with resources applicable to electronic functionality and magnitude directed more detailed processes to PIC16F628A.*

**KEYWORDS:** Interruption, microcontrollers, programming, PIC, synchronization systems, electronic, PIC16F628A.

## 1. INTRODUCCIÓN

Los seres humanos no estamos conscientes de las interrupciones, en nuestro organismo existen mecanismos que nos interrumpen constantemente, para ello tenemos a nuestro sistema sensorial, pero no somos conscientes del proceso de interrupción, aunque sí de la atención a las interrupciones.

Eso es porque incorporamos mecanismos que nos sacan rápidamente de la tarea que estemos haciendo para atender una situación que no puede o no debe esperar mucho tiempo. Bien, esa misma es la idea que se incorpora en los microprocesadores para atender procesos que no pueden esperar o que no sabemos

cuándo deben ser atendidos porque ello depende de determinadas condiciones.

El manejo de interrupciones es una técnica de programación, basada en un mecanismo automático en el hardware del microcontrolador, que permite dar atención a algún periférico interno o externo, únicamente en el momento en que ésta se requiera. Una interrupción es en realidad una llamada a una subrutina, pero iniciada por el hardware del propio periférico y no por la instrucción "GOSUB". La interrupción puede ocurrir en cualquier instante.

## 2. DESARROLLO

### INTERRUPCIÓN

Una interrupción es un evento que hace que el microcontrolador deje de ejecutar la tarea que está realizando para atender dicho acontecimiento y luego regrese y continúe la tarea que estaba realizando antes de que se presentara la interrupción.

Las interrupciones permiten a cualquier suceso interior o exterior la ejecución del programa principal en cualquier momento. En el momento de producirse la interrupción, el PIC ejecuta un salto a la rutina de atención a la interrupción, previamente definida por el programador, donde se atenderá a la demanda de la interrupción. Cuando se termina de ejecutar dicha rutina, el PIC retorna a la ejecución del programa principal en la misma posición de la memoria de programa donde se produjo la interrupción.

Al producirse una interrupción, el PIC salta automáticamente a la dirección del vector de interrupción de la memoria de programa y ejecuta la porción de programa correspondiente a la atención de la interrupción, hasta encontrar la instrucción Retfie. Al encontrar dicha instrucción, abandona la interrupción y retorna a la posición de memoria del programa principal desde la que saltó al producirse la interrupción.

### FUENTES DE INTERRUPCIÓN

Las fuentes de interrupción dependen del PIC utilizado. El pic 16F628 (y el 16F628A) tiene 10 fuentes de interrupción, si las interrupciones están habilitadas cada vez que una de estos acontecimientos se presente el pic dejará de ejecutar el programa para ir a atender la interrupción y al término de la misma continuará ejecutando el programa donde lo había dejado. Las fuentes de interrupción son:

- Interrupción externa RB0/INT.
- Interrupción por cambio lógico en el puerto B (pines RB7 a RB4).
- Interrupción por desborde del timer 0 (TMR0).
- Interrupción por desborde del timer 1 (TMR1).
- Interrupción por comparación exitosa en TMR2.
- Interrupción del comparador.
- Interrupción del transmisor del USART.
- Interrupción del receptor del USART.
- Interrupción del módulo CCP.
- Interrupción del EEPROM.



Fig. 1.- Manejo del sistema de interrupción

## TIPOS DE INTERRUPTACIONES

1.- Mediante una acción interna. El desbordamiento de la Pila (Stack) por una operación indebida, por ejemplo:

- ✓ Al completarse la escritura de datos en una EEPROM.
- ✓ Por desbordamiento del registro TMR0 al rebasar el valor 255 (FFh) a 0.

2.- Mediante una acción externa, la más útil. Al producirse un cambio del nivel en uno de sus pines por una acción externa.

- ✓ Estando en el modo de reposo (SLEEP), un cambio de nivel en el pin RB0/INT.
- ✓ Un cambio de nivel en uno de los pines RB4 a RB7 estando configurados como entrada

## PROCESO DE INTERRUPCIÓN

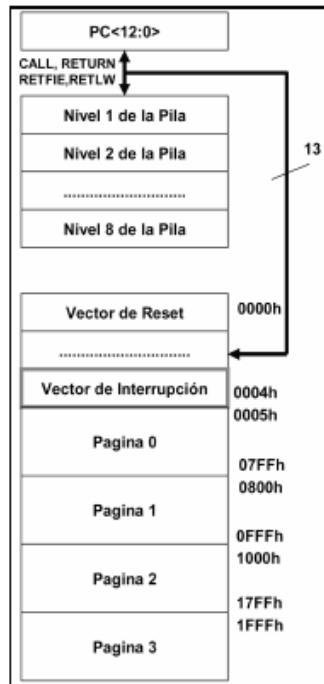
Cuando ocurre un evento de los descritos anteriormente, se produce una petición de interrupción, guardando el valor actual del PC (contador de programa) en la Pila, sea cual sea la fuente de la interrupción, se pone a cero el bit7 GIE (Global Interrupt Enable), con lo cual inhibe cualquier otra petición de interrupción, el registro PC se carga con el valor 0004h que, es la posición del vector de interrupción. Aquí, empieza la ejecución del programa

de atención a la interrupción ISR (Rutina de Servicio de Interrupción).

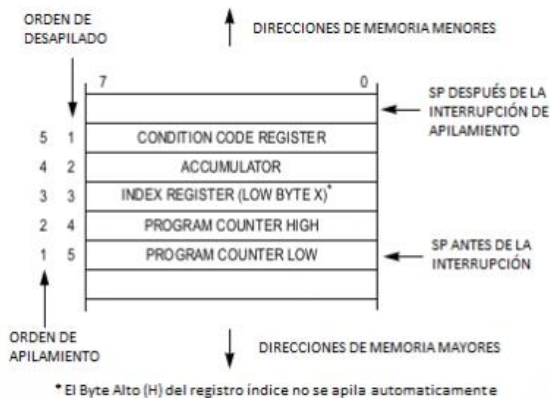
El tiempo de procesamiento de la ISR debe ser lo más breve posible, para que se ejecuten las otras interrupciones ya que, pueden habilitarse más de una de ellas. Además, cualquier tipo de interrupción también puede sacar al micro del modo de reposo (SLEEP).

1. En el mundo real se produce el evento para el cual queremos que el procesador ejecute un programa especial, este proceso tiene la característica de que no puede esperar mucho tiempo antes de ser atendido o no sabemos en qué momento debe ser atendido.
2. El circuito encargado de detectar la ocurrencia del evento se activa, y como consecuencia, activa la entrada de interrupción del procesador.
3. La unidad de control detecta que se ha producido una interrupción y "levanta" una bandera para registrar esta situación; de esta forma si las condiciones que provocaron el evento desaparecen y el circuito encargado de detectarlo desactiva la entrada de interrupción del procesador, ésta se producirá de cualquier modo, porque ha sido registrada.
4. La unidad de ejecución termina con la instrucción en curso y justo antes de comenzar a ejecutar la siguiente comprueba que se ha registrado una interrupción
5. Se desencadena un proceso que permite guardar el estado actual del programa en ejecución y saltar a una dirección especial de memoria de programas, donde está la primera instrucción de la subrutina de atención a interrupción.
6. Se ejecuta el código de atención a interrupción, esta es la parte "consciente" de todo el proceso porque es donde se realizan las acciones propias de la atención a la interrupción y el programador juega su papel.
7. Cuando en la subrutina de atención a interrupción se ejecuta la instrucción de retorno, se desencadena el proceso de restauración del procesador al estado en que estaba antes de la atención a la interrupción.

Los PIC de gama baja y media tienen un único vector de interrupción situado en la dirección 04h de programa, como se puede observar en la figura, mientras que los de gama alta tienen dos vectores de interrupción de distinta prioridad, alta y baja, situados en la posición 08h y 18h de la memoria de programa.



**Fig. 2.-** Memoria de programa (Posición del vector interrupción).



**Fig. 3.-** Apilamiento de registros al entrar y salir de la interrupción.

### Registro de control de interrupciones (INTCON)

Se trata de un registro leíble y escribible que, para facilitar su acceso, se halla duplicado en los cuatro bancos, ocupando las direcciones 0Bh, 8Bh, 10Bh y 18Bh, respectivamente. Tiene la misión de controlar las interrupciones provocadas por el TMR0, cambio de estado a las 4 líneas de más peso de la puerta B y activación de la pata RB0/INT.

### Registro de permiso de interrupciones 1 (PIE 1)

Contiene los bits que permiten (1) o prohíben (0) las interrupciones provocadas por los periféricos internos del microcontrolador y que no estaban contemplados en INTCON.

REGISTRO INTCON

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
7							0

<b>GIE</b>	Bit de permiso global de interrupciones. 1 = Permitido 0 = Prohibido
<b>PEIE</b>	Bit de permiso de los periféricos que no se controlan con INTCON.
<b>TOIE</b>	Bit de permiso de interrupción del TMR0.
<b>INTE</b>	Bit de permiso de la interrupción externa por RB0/INT.
<b>RBIE</b>	Bit de permiso de la interrupción por cambio en RB4-RB7.
<b>TOIF</b>	Señalizador de desbordamiento en TMR0.
<b>INTF</b>	Señalizador de activación de la pata RB0/INT.
<b>RBIF</b>	Señalizador de cambio en RB4-RB7.

**Fig. 4.-** Designación y misión de cada uno de los bits del registro INTCON.

### Registro de permiso de interrupciones 2 (PIE2)

Contiene los bits de permiso de interrupciones de las causas que no figuraban en PIE1. La de fin de escritura en la EEPROM, la precedente del comparador, colisión del bus en el modo SSP y producción de una captura o una comparación en el módulo CCP2.

### ESTABLECIMIENTO DE LAS DECLARACIONES DE EJECUCIÓN PARA EL PIC16F628A EN INTERRUPCIONES.

Todas éstas declaraciones sirven para ejecutar un handler (Subrutina con RESUME) de interrupción.

ON INTERRUPT GOTO prog2; quiere decir en caso de darse una interrupción suspender el programa actual e ir a prog2.

DISABLE; sirve para deshabilitar la interrupción, en caso de que no deseemos que atienda la interrupción, como por ejemplo cuando ya está atendiendo una interrupción.

RESUME; equivale al RETURN de un GOSUB, en donde retorna a la línea del programa en donde ocurrió la interrupción.

ENABLE; quiere decir habilitar nuevamente la interrupción, después de esto todas las interrupciones, son atendidas.

### Utilizando la interrupción del puerto b.4 al b.7.

Para trabajar con la interrupción por cambio de estado del puerto b.4 al puerto b.7, podemos hacerlo solamente cambiando las 2 líneas que dicen INTCON=%10010000 por INTCON=%10001000, que quiere decir habilitar la interrupción de los puertos B.4 al B.7.

### Reloj digital utilizando la interrupción del tmr0.

El TMR0, es una valiosa herramienta que disponen los PIC'S, para el caso del PIC16F628A, dispone de 3 TIMERS, 1 de 16 bits (TMR1) y 2 de 8 bits, los TMR0 y TMR2, la calibración para estos contadores, se dan en el registro OPTION, aquí se puede seleccionar si el incremento es con flanco de subida o de bajada y si la fuente es externa (pin A4/TOCKI) o interna (Oscilador).

Cuando el conteo del temporizador TMR0, llega a 256 y pasa a 0, se genera una interrupción, para que esto suceda se debe habilitar el registro INTCON bit 7 (GIE = 1), y también el bit 5 (TOIE =1), quedando así:

INTCON=%10100000 ; habilita interrupción por TMR0

En el registro OPTION se debe definir la pata del pre escalador, se debe poner:

OPTION\_REG=%01010110 ; pre escalador 1:128, asignado al TMR0 y ciclo de reloj interno.

### 3.- SOFTWARE

; programa para manejar una interrupción en el p B.0

LED VAR PORTB.1

LED2 VAR PORTB.2

**ON INTERRUPT GOTO** verde ; en caso de existir una interrupción ir a verde

INTCON = %10010000 ;habilita la interrupción B.0

PROG: ;programa principal

**HIGH LED** ;encender el led rojo

**PAUSE 200**

**LOW LED**

**PAUSE 200**

**GOTO PROG** ;mantener en este lazo

**DISABLE** ;deshabilita las interrupciones en el handler verde: ;handler de la interrupción

**HIGH LED2**

**PAUSE 2000**

**LOW LED2**

INTCON = %10010000 ;habilita la interrupción B.0

**RESUME** ;retorna a programa principal donde ocurrió la inte.

**ENABLE** ;Habilita las interrupciones después del handler

**END**

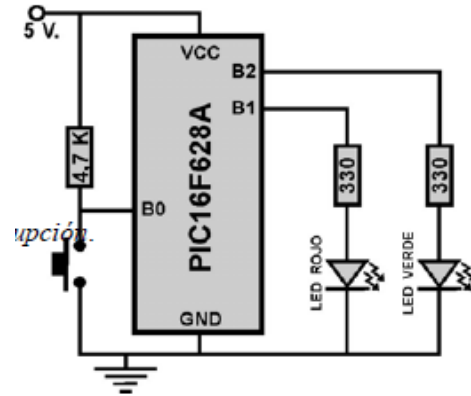


Fig. 5.- Conexión de dos leds y un pulsador para la práctica de interrupción.

### 4.- CONCLUSIONES

- La interrupción en los microcontroladores es un evento programable o inducido externamente por dispositivos electrónicos, en el cual el PIC deja de realizar lo que estaba haciendo y procede a realizar otra tarea momentáneamente, hasta que se cumple con la misma y regresa a hacer lo que estaba cumpliendo previamente.
- El manejo de interrupciones es una alternativa al método conocido como "polling", en el cual dentro del programa principal, se está revisando el status de cada uno de los periféricos, una y otra vez en un loop permanente, para conocer si alguno de ellos necesita de atención.
- El manejo de interrupciones permite operar el manejo del "multitasking" o multitareas, es decir, una técnica de programación en la cual es posible dar atención a multitud de periféricos, reduciendo al mínimo el tiempo de procesamiento.
- El 16F628A cuenta las mismas fuentes de interrupción que el 16F84, más otras provenientes del USART, de los 3 temporizadores y de los módulos de CAPTURE/COMPARE. El 16F628A puede manejar hasta 10 fuentes de interrupción.
- En el PIC16F628A se utilizan declaraciones como On interrupt, disable, resume y enable. todas estas declaraciones sirven para ejecutar un handler (subrutina) de interrupción.

## 4.- RECOMENDACIONES

- Para el proceso de las interrupciones en los microcontroladores es conveniente conocer los diferentes tipos de configuraciones y el establecimiento de declaraciones para que el funcionamiento del circuito sea el adecuado
- Se recomienda analizar el funcionamiento de los pines de nuestro PIC en función de las fuentes de interrupción y así mismo darles un buen uso para que al momento de editar interrupciones, éstas no queden deshabilitadas.

## 5.- BIBLIOGRAFÍA

ANGULO José, ROMERO Susana, ANGULO Ignacio. Microcontroladores PIC: diseño práctico de aplicaciones. Segunda parte. Editorial McGraw-Hill España. 2006, pág 392, 393 y 394. Extraído del repositorio digital de la Universidad de las Fuerzas Armadas ESPE el 22 de diciembre del 2014.

GARCÍA, Eduardo. Compilador C CCS y Simulador PROTEUS para Microcontroladores PIC. 1ra ED. Editorial Marcombo España, 2008, páginas 271, 272, 273 y 274. Extraído del repositorio digital de la Universidad de las Fuerzas Armadas ESPE el 22 de diciembre del 2014.

REYES, Carlos. Microcontroladores PIC, Programación en BASIC. 3ra Ed. Editorial Rispergraf, 2008, pág 160, 161 y 162.

Enlaces relacionados recuperados el 22 de diciembre del 2014:

- <http://www.puntoflotante.net/INTERRUPCIONES.htm>
- <http://avecomputointe.blogspot.com/2012/05/interrupciones-en-microcontroladores.html>