

# Microcontroladores

## Semana 7

Semestre 2025-2

Por Kalun José Lau Gan

### Preguntas previas

- ¿Qué es el “BRA \$-2”?
  - Es un salto (branch) de dos posiciones de memoria (asumiendo cada posición de un byte) hacia atrás.
- ¿Va a dejar debates en el AV?
  - --
- La redacción del informe es de manera impersonal
- En I2C. ¿Cómo hacemos con las resistencias de pullup?
  - El PIC18F57Q43 posee pullups especialmente para dicha comunicación, revisar cap 19 del datasheet, de lo contrario se tendrán que colocar de manera externa en los puertos SDA y SCL.

## Agenda:

- **Interrupciones** en el PIC18F57Q43 (cap 11 del datasheet)
- Cap. 11 – VIC (Vectorized Interrupt Controller)
- En la PC se aneja en base a Interrupt Requests (IRQs)

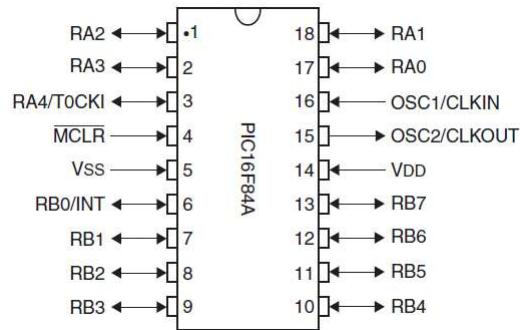
IRQ	Usage
0	system timer (cannot be changed)
1	keyboard controller (cannot be changed)
2	cascaded signals from IRQs 8–15
3	second RS-232 serial port (COM2: in Windows)
4	first RS-232 serial port (COM1: in Windows)
5	parallel port 2 and 3 or sound card
6	floppy disk controller
7	first parallel port
8	real-time clock
9	open interrupt
10	open interrupt
11	open interrupt
12	PS/2 mouse
13	math coprocessor
14	primary ATA channel
15	secondary ATA channel

## ¿Qué son las Interrupciones?

- Es la atención de eventualidades dentro de la ejecución de un programa.
- En general, los microcontroladores poseen diversas fuentes de interrupción pero por defecto esta funcionalidad se encuentra desactivada.
- Es mejor trabajar bajo un esquema de interrupciones que estar con el “polling”
- Se tiene “acción de interrupción” y “motivo de la interrupción”

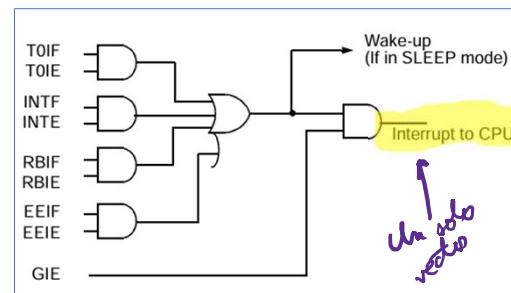
## Antes: El PIC16F84A

- Microcontrolador de la gama media de Microchip
- Posee 1K de memoria de programa. 68bytes de memoria de datos.
- Solo un periférico: Timer0



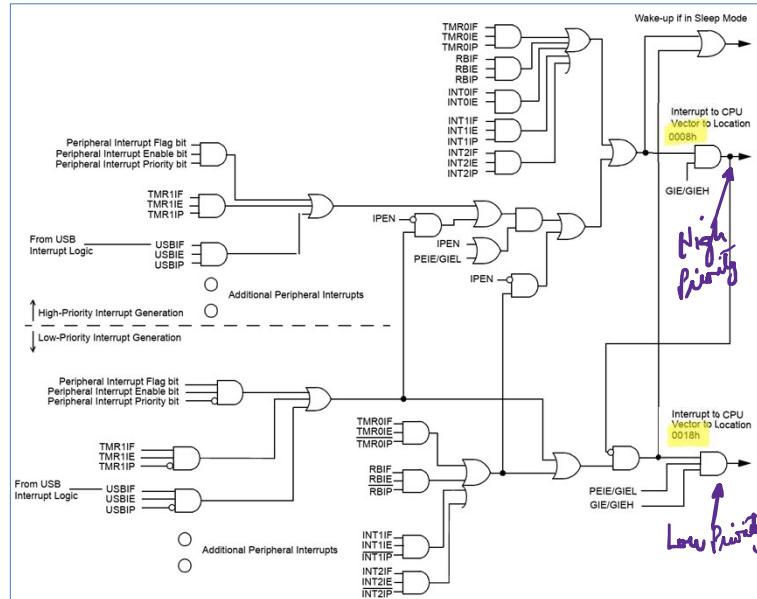
## Antes: El PIC16F84A

- Un solo vector de interrupción (canal de atención de las interrupciones por el CPU)
- Cuatro fuentes de interrupción
- En software se tenía que definir la prioridad, o sea, a quien atender primero si es que se tiene mas de una fuente de interrupción habilitada



## Antes en el PIC18F4550:

- El concepto de prioridades en las interrupciones.
- Se tiene dos prioridades manejadas en hardware: high priority y low priority
- Las políticas de atención:
  - Si llegan ambas a la vez (la alta y la baja), se da prioridad de ejecución a la alta prioridad
  - Si ocurre una interrupción de alta prioridad y luego mientras se está ejecutando ésta y llega una interrupción de baja prioridad, se sigue ejecutando la alta prioridad y se deja en espera la baja prioridad.
  - Si ocurre una interrupción de baja prioridad y mientras se está ejecutando ésta y llega una interrupción de alta prioridad, la interrupción de baja prioridad que se encuentra ejecutando pasará a pausarse y de inmediato se atiende la alta prioridad, luego de ejecutarse la de alta prioridad se resume la ejecución de la interrupción de baja prioridad.



## En el PIC18F57Q43

- El **modo legacy** funciona de manera similar que el microcontrolador PIC18F4550 en la forma de atención de las interrupciones con dos vectores de interrupción (high priority y low priority).
- Para activar el modo legacy, se tiene que colocar el bit de configuración MVECEN = 0 ó OFF

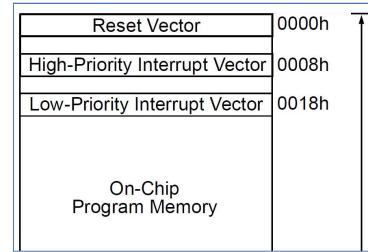
### 11.3.2 Interrupt Vector Table Contents MVECEN = 0

When MVECEN = 0, the address location pointed to by **IVTBASE** has a **GOTO** instruction for a high-priority interrupt. Similarly, the corresponding low-priority vector also has a **GOTO** instruction, which is executed in case of a low-priority interrupt.

Name	Value	Field	Option	Category	Setting
CONFIG1	FF	-	-	-	-
7	PEXTOSC	ECH		External Oscillator Selection	EC (external clock) above 8 MHz
7	RSTOSC	EXTOSC		Reset Oscillator Selection	EXTOSC operating per PEXTOSC bits (device manufacturing default)
CONFIG2	FF	-	-	-	-
1	CLKOUTEN	OFF		Clock out Enable bit	CLKOUT function is disabled
1	PRIWAT	ON		PRILOCKED One-Way Set Enable bit	PRILOCKED bit can be cleared and set only once
1	CSEN	ON		Clock Switch Enable bit	Writing to NOSC and NDIV is allowed
1	FSCMEN	ON		Fail-Safe Clock Monitor Enable bit	Fail-Safe Clock Monitor enabled
CONFIG3	F?	-	-	-	-
1	MCLR	EXTMCLR		MCLR Enable bit	If LVP = 0, MCLR pin is MCLR; If LVP = 1, RE3 pin function is MCLR
3	PWR1	PWR1 OFF		Power-up timer selection bits	PWR1 is disabled
0	MVECEN	OFF		Multi-vector enable bit	Interrupt controller does not use vector table to prioritize interrupts

## Interrupciones:

- Las interrupciones son **eventualidades** que detienen el flujo normal de operación del microcontrolador.
- Dos modos de operación: **Vectorizado** (MVECEN=ON) y el **Legacy** (MVECEN=OFF). MVECEN es un bit de configuración
- Todos los periféricos internos del microcontrolador (Timers, INTs externas, CCP, EUSART, A/D, comparadores analógicos, etc) pueden emitir interrupciones al CPU.
- Las banderas que indican la fuente de interrupción (xxxIF) deberán de bajarse manualmente una vez activados.
- Cuando bit de configuración MVECEN = OFF:
  - En el PIC18F57Q43 tenemos dos vectores de interrupción:
    - Alta Prioridad (0x0008)
    - Baja Prioridad (0x0018)
  - Las interrupciones vienen desactivadas por defecto (GIE=0 y todos los xxxIE=0).
  - Las prioridades están desactivadas por defecto (IPEN=0), si están desactivadas, todas van al 0x0008)
  - Al activarse las prioridades (IPEN=1) todas las fuentes estarán en alta (xxxIP=1)



```

ORG 000000H
bra configuro

ORG 000008H
bra INT_ISR_HP

ORG 000018H
bra INT_ISR_LP

ORG 000020H
configuro:    ;--;
               ;--;
               ;--;
  
```

## Sobre el **modo legacy** en interrupciones

- Hay que tener en cuenta que para que funcione este modo, el bit de configuración MVECEN debe de estar en OFF (por defecto esta en ON)**
- Tener en cuenta que si no se trabaja con prioridades (IPEN=0), todos los eventos se van a la dirección 000008H de la memoria de programa
- Si es que se activa las prioridades, por defecto todas las fuentes de interrupción están en alta prioridad
- Políticas de atención en prioridades:
  - Cuando un evento de alta prioridad y un evento de baja prioridad ocurren al mismo tiempo, el CPU primero atiende al de alta prioridad dejando en cola la atención del de baja prioridad, luego de atender la de alta prioridad se procede a atender al de baja prioridad.
  - Cuando ocurre un evento de alta prioridad, la atiendes y en ese momento recibes una de baja prioridad, el CPU sigue atendiendo al de alta prioridad y luego de terminar atiende la de baja prioridad.
  - Cuando ocurre un evento de baja prioridad, la atiendes y en ese momento ocurre una de alta prioridad, el CPU pone en pausa la atención del de baja prioridad y se va a atender al de alta prioridad, una vez terminado de atender la alta prioridad resume la atención del de baja prioridad.
  - Cuando se reciben dos interrupciones de la misma prioridad. El desarrollador tiene que decidir en el programa a quien atender primero.

## Table 11-2 Sobre el modo vectorizado en interrupciones

Vector Number	Interrupt source	Vector Number (cont.)	Interrupt source (cont.)
0x0	Software Interrupt	0x3E	PWM3RINT
0x1	HLV/D (High/Low-Voltage Detect)	0x3F	PWM3GINT
0x2	OSF (Oscillator Fail)	0x40	U0RX
0x3	CSW (Clock Switching)	0x41	U2TX
0x4	-	0x42	U2E
0x5	CLC1 (Configurable Logic Cell)	0x43	U2
0x6	-	0x44	TMR5
0x7	IOC (Interrupt-On-Change)	0x45	TMR5G
0x8	INT0	0x46	CCP2
0x9	ZCD (Zero-Cross Detection)	0x47	SCAN
0xA	AD (ADC Conversion Complete)	0x48	U3RX
0xB	ACT (Active Clock Tuning)	0x49	U3TX
0xC	CMI (Comparator)	0x4A	U3E
0xD	SMT1 (Signal Measurement Timer)	0x4B	U3
0xE	SMT1PRA	0x4C	-
0xF	SMT1PWA	0x4D	CLC4
0x10	ADT	0x4E - 0x4F	-
0x11 - 0x13	-	0x50	INT2
0x14	DMA15CNT (Direct Memory Access)	0x51	CLC5
0x15	DMA1DCNT	0x52	CWG2 (Complementary Waveform Generator)
0x16	DMA1OR	0x53	NCO2
0x17	DMA1TA	0x54	DMA35CNT
0x18	SPI1RX (Serial Peripheral Interface)	0x55	DMA3DCNT
0x19	SPI1TX	0x56	DMA3OR
0x1A	SPI1	0x57	DMA3A
0x1B	TMR2	0x58	CCP3
0x1C	TMR1	0x59	CLC6
0x1D	TMR1TG	0x5A	CWG3
0x1E	CCP1 (Capture/Compare/PWM)	0x5B	TMR4
0x1F	TMR0	0x5C	DMA45CNT
0x20	U1RX	0x5D	DMA4DCNT
0x21	U1TX	0x5E	DMA4OR
0x22	U1E	0x5F	DMA4AA
0x23	U1	0x60	U4RX
0x24 - 0x25	-	0x61	U4TX
0x26	PWM1RINT	0x62	U4E
0x27	PWM1GINT	0x63	U4
0x28	SPI2RX	0x64	DMA55CNT
0x29	SPI2TX	0x65	DMA5DCNT
0x2A	SPI2	0x66	DMA5OR
0x2B	-	0x67	DMA5A
0x2C	TMR3	0x68	USRX
0x2D	TMR3G	0x69	USTX
0x2E	PWM2RINT	0x6A	U5E

Vector Number	Interrupt source	Vector Number (cont.)	Interrupt source (cont.)
0x2F	PWM2GINT	0x6B	U5
0x30	INT1	0x6C	DMA5CNT
0x31	CLC2	0x6D	DMA6CNT
0x32	CWG1 (Complementary Waveform Generator)	0x6E	DMA6OR
0x33	NCO1 (Numerically Controlled Oscillator)	0x6F	DMA6A
0x34	DMA25CNT	0x70	-
0x35	DMA2DCNT	0x71	CLC7
0x36	DMA2OR	0x72	CM2
0x37	DMA2A	0x73	NCO3
0x38	I2C1RX	0x74 - 0x77	-
0x39	I2C1TX	0x78	NVM
0x3A	I2C1	0x79	CLC8
0x3B	I2C1E	0x7A	CRC (Cyclic Redundancy Check)
0x3C	-	0x7B	TMR6
0x3D	CLC3	0x7C - 0x8F	-

IVTBASE por defecto en MVECEN ON es 000000H

- Tener en cuenta que, para activar este modo, el bit de configuración MVECEN debe de estar en ON.
- En este modo, todas las fuentes de interrupción poseen un vector de interrupción con una dirección en particular establecido por el cálculo siguiente:

Table 11-1. IVT Calculations Summary

IVT Address Calculation	Interrupt Priority INTCON Register, IPEN Bit	
	0	1
Multivector Enable, MVECEN Configuration bit	0	IVTBASE High-priority IVTBASE Low-priority IVTBASE + 8 words
	1	IVTBASE + 2*(Vector Number)

## La interrupción del Timer0

- Dependiendo del modo de trabajo:
  - Modo 8 bit: El evento de interrupción proviene de la comparación en igualdad entre TMR0H (valor de referencia de comparación) y TMROL (registro de cuenta) - tmr0\_match
  - Modo 16 bit: El evento de interrupción proviene del desborde del registro de cuenta (65535 hacia 0) – tmr0\_overflow
- Tener en cuenta el valor que se le coloca en POSTSCALER, este dispositivo contará una cantidad de eventos (tmr0\_match o tmr0\_overflow) antes de activar la bandera TMROIF y que va desde 1:1 hasta 1:16.
  - Si POSTSCALER = 1:1, cada evento (tmr0\_latch ó tmr0\_overflow) activará TMROIF
  - Si POSTSCALER = 1:2, cada dos eventos (tmr0\_match ó tmr0\_overflow) activará TMROIF
- Cuando se levante la bandera (TMROIF), ésta deberá de bajarse manualmente.
- Para habilitar la interrupción del Timer0:
  - Activar TMROIE, ubicado en el bit7 del registro PIE3
  - Activar GIE, ubicado en el bit7 del registro INTCON0
- De usar prioridades en las interrupciones del Timer0 (modo legacy MVECEN=OFF):
  - Revisar el bit 7 (TMROIP) del registro IPR3, por defecto TMROIP=1 (alta prioridad)
  - Revisar bits IPEN, GIEH y GIEL del registro INTCON0

## Plantilla para manejar interrupción del TMR0 en modo legacy (MVECEN=OFF) y sin prioridades (IPEN=0)

```

PROCESSOR 18F57Q43
#include "cabecera.inc"

PSECT upcino, class=CODE, reloc=2, abs
upcino:
    ORG 000000H
    bra configuro
    ORG 000008H
    bra TMR0_ISR

    ORG 000100H
configuro:
    ;-- 
    ;-- 
    bsf PIE3, 7      ;Habilito interrupcion de TMR0
    bsf INTCON0, 7   ;Habilito GIE

    inicio: ;--          ;Rutina principal
    ;-- 
    bra inicio

TMR0_ISR:           ;Rutina de interrupcion de TMR0
    ;Lo que tiene que hacer cuando ocurrió una INT del TMR0
    bcf PIR7, 3       ;Baja bandera TMR0IF
    ;-- 
    ;-- 
    retfie            ;retorno de donde viniste

end upcino

```

← Vector de alta prioridad

## Plantilla para manejar interrupción de TMR0 con MVECEN=ON (modo vectorizado)

```

PROCESSOR 18F57Q43
#include "cabecera.inc"

PSECT upcino, class=CODE, reloc=2
upcino:
    ORG 000000H      ;vector de RESET
    goto configuro

    ORG 00003EH      ;vector de la TMR0 (2*1FH)
    DW TMR0_ISR shr 2

;Cada rutina de interrupción debe de tener su propio PSECT
    PSECT TMR0_ISR, class=CODE, reloc=4
TMR0_ISR:
    bcf PIR3, 7, 1    ;bajamos bandera INT0IF
    ;-- 
    retfie

;Inicio de código de usuario
    PSECT code
    ORG 000100H

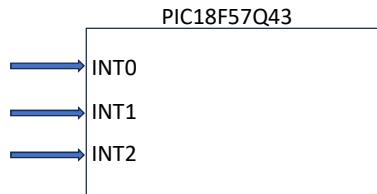
configuro:
    ;-- 
    clrf IVTBASEU, 1
    clrf IVTBASEH, 1
    clrf IVTBASEL, 1
    bsf PIE3, 7, 1    ;activando TMR0 con TMR0IE=1
    bcf PIR3, 7, 1    ;bajo bandera TMR0IF
    bsf INTCON0, 7, 1 ;GIE=1 habilitando el interruptor global

;Rutina principal no hace nada
inicio:
    bra inicio

end upcino

```

## Las interrupciones externas INT0, INT1, INT2



- Interrumpen al CPU del microcontrolador, estas acciones provienen de pines externos (INT0, INT1 e INT2).
- Pines externos (por defecto RB0 para INT0, RB1 para la INT1 y RB2 para INT2). Configurables su asignación por el PPS (**registros INTxPPS**, revisar cap 21).
- Pueden ser activos en alto (flanco ascendente) o activos en bajo (flanco descendente). Revisar bits INTOEDG, INT1EDG y INT2EDG (**registro INTCON0**).
- Los bits de habilitación de las interrupciones externas son INT0IE, INT1IE e INT2IE. Estos se encuentran en el grupo de **registros PIEx**
- Las banderas INTOIF, INT1IF e INT2IF deberán de bajarse (acción manual) cuando se activen. Se encuentran en el grupo de **registros PIRx**.

## Las interrupciones externas INT0, INT1, INT2

- En MVECEN=OFF (modo legacy) por defecto las tres interrupciones saltan al vector de alta prioridad (000008H) al ser activadas. Si se requiere pasar a baja prioridad (000018H) alguno de ellos se deberá activar la prioridad de las interrupciones (IPEN=1) y configurar los bits INTOIP, INT1IP e INT2IP según requerimientos. Se encuentran en el grupo de **registros IPRx**.
- Recordar que el GIE/GIEH (habilitador global, global alta prioridad), GIEL (habilitador global baja prioridad) e IPEN (habilitador de prioridades se encuentran en el **registro INTCON0**).

## Las interrupciones externas INT0, INT1, INT2

- En MVECEN=ON hay que tener en cuenta que las direcciones de los vectores de interrupción se tienen que calcular según tabla 11-1:

IVTBASE + (2 x Número de Vector)

- En MVECEN=ON las interrupciones INT0, INT1 e INT2 tienen sus propios vectores, al iniciar IVTBASE tiene valor 0 por lo que usando la fórmula anterior y según tabla 11-2 se obtienen las siguientes direcciones:
  - INT0 – 000010H
  - INT1 – 000060H
  - INT2 – 0000A0H
- Revisar la plantilla vista anteriormente ya que se requiere crear PSECTs para cada vector de interrupción en este modo modificando el reloc a valor 4.

## Las interrupciones externas INT0, INT1, INT2

- Registros PPS de las interrupciones externas
- Tener en cuenta que estos registros se encuentran en el bank 2

Ejemplo: Asignar a RA3 como entrada de INT0:

INTOPPS = 03H

Peripheral	PPS Input Register	Default Pin Selection at POR	Register Reset Value at POR	Available Input Port							
				28-Pin Devices		40-Pin Devices		48-Pin Devices			
0x023E	Interrupt 0	INT0PPS	'b001 000	A	B	—	A	B	—	A	B
0x023F	Interrupt 1	INT1PPS	'b001 001	A	B	—	A	B	—	B	D
0x0240	Interrupt 2	INT2PPS	'b001 010	A	B	—	A	B	—	B	F

Name: xxxPPS  
Peripheral Input Selection Register

Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	PORT[2:0]	R/W	R/W	R/W	R/W	R/W
Reset	m	m	m	m	m	m	m	m

Bits 5:3 – PORT[2:0] Peripheral Input PORT Selection<sup>(1)</sup>  
See the PPS Input Selection Table for the list of available Ports and default pin locations.

PORT	Selection
101	PORTF
100	PORTE
011	PORTD
010	PORTC
001	PORTB
000	PORTA

Reset States: POR = mmm  
All other Resets = uuu

Bits 2:0 – PIN[2:0] Peripheral Input PORT Pin Selection<sup>(2)</sup>  
Reset States: POR = mmm  
All other Resets = uuu

Value	Description
111	Peripheral input is from PORTx Pin 7 (Rx7)
110	Peripheral input is from PORTx Pin 6 (Rx6)
101	Peripheral input is from PORTx Pin 5 (Rx5)
100	Peripheral input is from PORTx Pin 4 (Rx4)
011	Peripheral input is from PORTx Pin 3 (Rx3)
010	Peripheral input is from PORTx Pin 2 (Rx2)
001	Peripheral input is from PORTx Pin 1 (Rx1)
000	Peripheral input is from PORTx Pin 0 (Rx0)

## Plantilla para manejar interrupciones externas con MVECEN=OFF (modo legacy)

- Se está considerando que INT0 se encuentre en alta prioridad mientras que INT1 se encuentre en baja prioridad.
- Si IPEN=1, todas las fuentes de interrupción se encuentran en alta prioridad por defecto.
- Para cambiar alguna fuente de interrupción a baja prioridad debemos de ubicar dicha fuente en los registros IPRx.

```

PROCESSOR 18F57Q43
#include "cabecera.inc"

PSECT upcino, class=CODE, reloc=2, abs
upcino:
    ORG 000000H
    bra configuro

    ORG 000000H      ;Vector de alta prioridad
    bra INT0_ISR

    ORG 000018H      ;Vector de baja prioridad
    bra INT1_ISR

    ORG 000100H
configuro:
    ;-
    bsf PIE1, 0      ;Habilito interrupcion de INT0
    bsf PIE6, 0      ;Habilito interrupcion de INT1
    bsf INTCON0, 5   ;Habilito IPEN
    bsf INTCON0, 7   ;Habilito GIEH
    bsf INTCON0, 6   ;Habilito GIEL
    bcf IPR6, 0      ;Mando INT1 a baja prioridad

inicio: ;-
        bra inicio          ;Rutina principal

INT0_ISR:                         ;Rutina de interrupcion de INT0
    bcf PIR1, 0          ;Lo que tiene que hacer cuando ocurrió una INT0
    ;-
    retfie               ;Baja bandera INT0IF

INT1_ISR:                         ;Rutina de interrupcion de INT1
    bcf PIR6, 0          ;Lo que tiene que hacer cuando ocurrió una INT1
    ;-
    retfie               ;Baja bandera INT1IF
end upcino

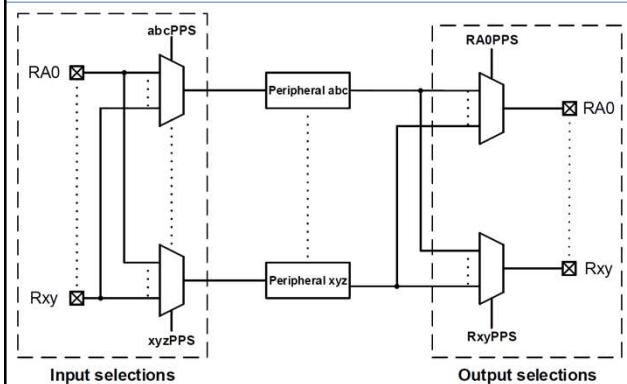
```

## Plantilla para manejar interrupciones externas con MVECEN=ON (modo vectorizado)

- Se está considerando INT0 e INT1 como fuentes de interrupción.
- La prioridad se da según tabla 11-2 de la hoja técnica.

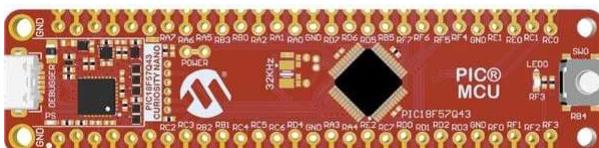
<pre> PROCESSOR 18F57Q43 #include "cabecera.inc"  PSECT upcino, class=CODE, reloc=2 upcino:     ORG 000000H      ;vector de RESET     goto configuro      ORG 000010H      ;vector de la INT0 (2*08H)     DW INT0_ISR shr 2      ORG 000060H      ;vector de la INT1 (2*30H)     DW INT1_ISR shr 2  ;Cada rutina de interrupción debe de tener su propio PSECT PSECT INT0_ISR, class=CODE, reloc=4 INT0_ISR:     bcf PIR1, 0, 1    ;bajamos bandera INT0IF     ;-     retfie  PSECT INT1_ISR, class=CODE, reloc=4 INT1_ISR:     bcf PIR6, 0, 1    ;bajamos bandera INT1IF     ;-     retfie </pre>	<pre> ;Inicio de codigo de usuario PSECT code ORG 000100H  configuro: ;-     movlb 2H           ;Faltó el PPS XD     movlw 0CH     movwf INT0PPS, 1   ;asignación de INT0 a RB4     movlw 1BH     movwf INT1PPS, 1   ;asignación de INT1 a RD3     clrf INTVBASEU, 1     clrf INTVBASEH, 1     clrf INTVTD, 1     bsf PIE1, 0, 1     ;activando INT0 con INT0IE=1     bcf PIR1, 0, 1     ;bajo bandera INT0IF     bsf PIE6, 0, 1     ;activando INT1 con INT1IE=1     bcf PIR6, 0, 1     ;bajo bandera INT1IF     bcf INTCON0, 0, 1   ;INT0EDGE=0 para detección de flanco negativo     bcf INTCON0, 1, 1   ;INT1EDGE=0 para detección de flanco negativo     bsf INTCON0, 7, 1   ;GIE=1 habilitando el interruptor global  ;Rutina principal no hace nada inicio:     bra inicio  end upcino </pre>
--	--

## El módulo PPS (Peripheral Pin Select)

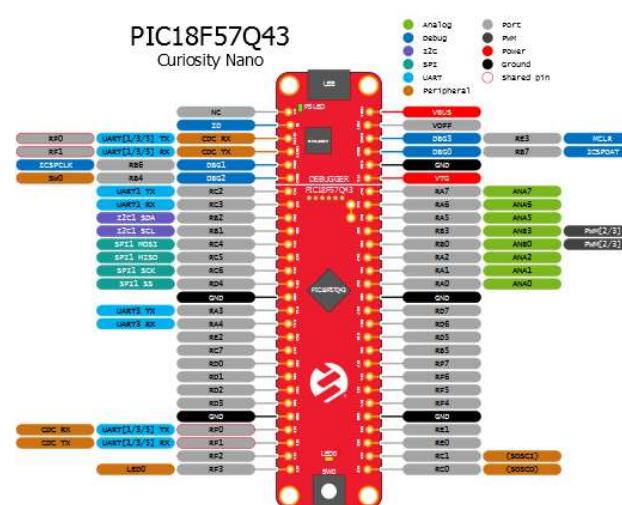


- Referencia cap 21 del datasheet
- Módulo encargado de asignar las señales de los periféricos a los pines del microcontrolador.
- Sólo para señales digitales, tanto de entrada como de salida.
- Con este módulo podrás reasignar los pines de los diferentes periféricos que tiene el microcontrolador, limitado a solo lo que indican las tablas 21-1 (entradas) y 21-2 (salidas).

## Ejemplo de PPS aplicado a la INT0



- La INT0 tiene por defecto el RB0 como puerto de entrada
- Tenemos en el Curiosity Nano un botón en RB4 activo en bajo drenador abierto.
- ¿Cómo haría para que el RB4 sea la INT0 del microcontrolador?



## Ejemplo de PPS aplicado a la INT0

- Respondiendo a la pregunta:
- Usando el PPS podemos cambiar la asignación por defecto (RB0) de la entrada del INT0 hacia el pin RB4 (revisar tabla 21-1 del datasheet)
- El valor obtenido para el registro INTOPPS fue 0CH según 21.8.1 del datasheet
- Recordar que INTOPPS se ubica en bank2 (ver sección 21.9 del datasheet)

```

ORG 000000H           ;Vector de reset
bra configuro

ORG 000008H           ;Vector de interrupcion
bra INT0_ISR

ORG 000100H           ;configuro:
movlb 4H
bsf TRISB, 4, 1
bcf ANSELB, 4, 1
bsf WPUB, 4, 1
movib 2H
movlw 0CH
movwf INTOPPS          ;Importante: cambiando bank2
movib 4H
bsf PIE0, 0, 1
movlw B1H
movwf INTCON0           ;asignando RB4 para INT0
                           ;al BANK4
                           ;habilitando INT0 (INT0IE=1)
                           ;GIE=1, INT0EDG=0
inicio:    movlb 4H           ;Rutina principal
;---
;---
;---
bra inicio

INT0_ISR:   ;---           ;Rutina de interrupcion
;---
;---
retfie

```

## Ejemplo de PPS aplicado a la INT0

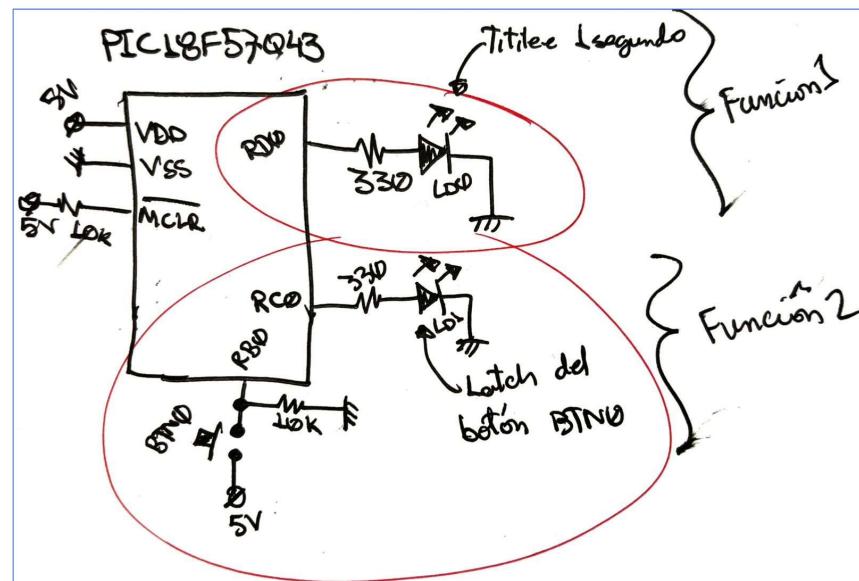
- Detalle del INTOPPS:

$\text{INT0 PPS} = 0\text{CH}$

21.8.1 xxxPPS	
Name:	xxxPPS
Peripheral Input Selection Register	INT0 PPS
Bit	0 0 0 0 0 0 1 3 1 1 1 1 1 1 1 1
Access	m m m m m m m m m m m m m m m m
Reset	m m m m m m m m m m m m m m m m
PORT[2:0]	POR
Bits 5:3 - PORT[2:0] Peripheral Input PORT Selection <sup>(1)</sup>	See the PPS Input Selection Table for the list of available Ports and default pin locations.
PORT	Selection
101	PORTF
100	PORTE
011	PORTD
010	PORTC
001	PORTB
000	PORTA
Reset States: POR = mmm All other Resets = uuu	
Bits 2:0 - PIN[2:0] Peripheral Input PORT Pin Selection <sup>(2)</sup>	
Reset States: POR = mmm All other Resets = uuu	
Value	Description
111	Peripheral input is from PORTx Pin 7 (Rx7)
110	Peripheral input is from PORTx Pin 6 (Rx6)
101	Peripheral input is from PORTx Pin 5 (Rx5)
100	Peripheral input is from PORTx Pin 4 (Rx4)
011	Peripheral input is from PORTx Pin 3 (Rx3)
010	Peripheral input is from PORTx Pin 2 (Rx2)
001	Peripheral input is from PORTx Pin 1 (Rx1)
000	Peripheral input is from PORTx Pin 0 (Rx0)

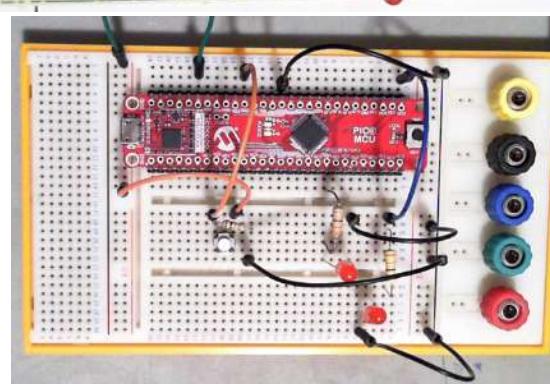
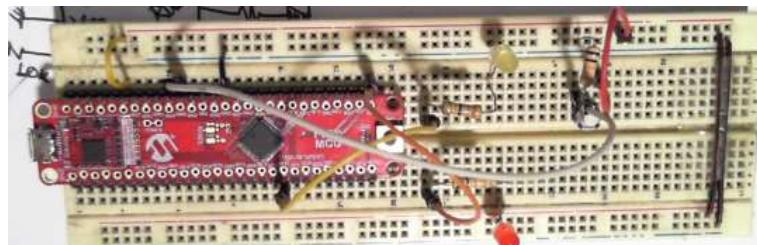
## Ejemplo sobre interrupciones

- Realizar dos funciones en el microcontrolador, uno de parpadeo de un LED con periodo de un segundo y otro de función latch de un LED con un pulsador (activo en alto)



## Ejemplo sobre interrupciones

- Hardware:

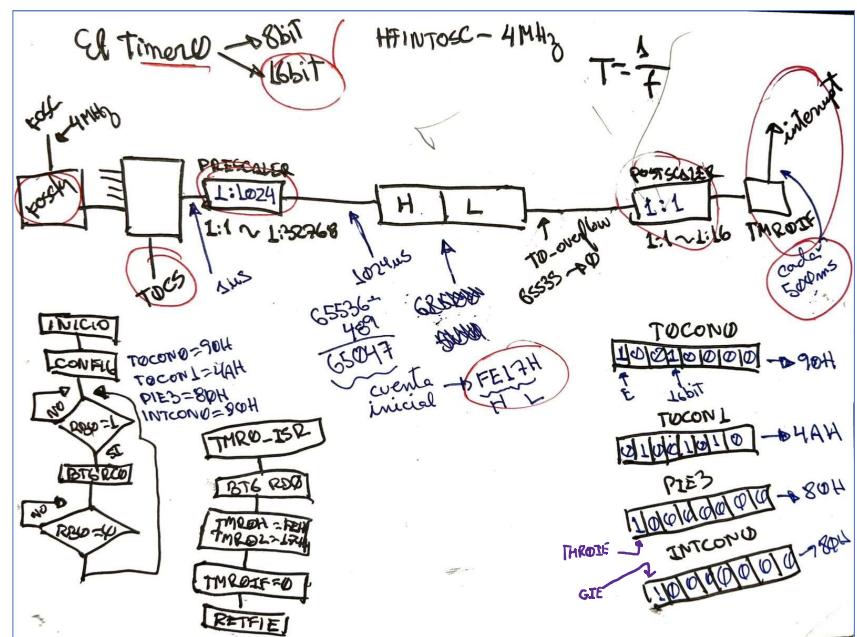


## Ejemplo sobre interrupciones

- El ejemplo es hacer dos cosas aparentemente al mismo tiempo (multi-tarea ó multitasking)
- Estrategias para desarrollar el ejemplo:
  1. No utilizar interrupciones (polling – high load CPU) X
  2. Emplear **interrupción del Timer0** para el parpadeo del LED y en la rutina principal hacer el Latch
  3. Emplear **interrupción externa INT0** para el Latch y en la rutina principal colocar el Timer0
  4. Emplear interrupciones tanto para el Timer0 como para el Latch (INT0)
    - Se puede interrupciones simples (un solo vector de interrupción)
    - Se puede utilizar prioridades legacy (high priority y low priority con MVECEN=OFF)
    - Se puede usar el VIC (interrupciones vectorizadas con MVECEN=ON)

## Estrategia2: Timer0 en interrupción y rutina principal el Latch

- El Timer0 debe de temporizar 1 segundo.
- Empleando el Timer0 en modo 16 bits, prescaler 1:1024, postscaler 1:1, fosc/4, cuenta inicial de 65047 (FE17H)
- Se habilita la interrupción para Timer0 (INT0IE=1)
- No olvidarse del GIE=1



## Estrategia2: Timer0 en interrupción y rutina principal el Latch

- Código propuesto

```

1  PROCESSOR 16F57043
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6    ORG 000000H      ;Vector de reset
7    bra configuro
8
9    ORG 000008H      ;Vector de interrupcion
10   bra TMRO_ISR
11
12   ORG 000100H
13 configuro:
14   movlb 0H          ;bank0
15   movlw 60H
16   movwf OSCCON1, 1  ;hfintosc, 1:1
17   movlw 02H
18   movwf OSCFRCQ, 1 ;hfintosc 4MHz
19   movlw 40H
20   movwf OSCEN, 1   ;hfintosc enabled
21   movlb 3H          ;bank3
22   movlw 90H
23   movwf T0CON0, 1  ;tmr0 enabled, 16 bit, posts 1:1
24   movlw 4AH
25   movwf T0CON1, 1  ;fosc/4 presc 1:1024

```

Nota: No olvidar de colocar el bit de configuración MVECEN=OFF

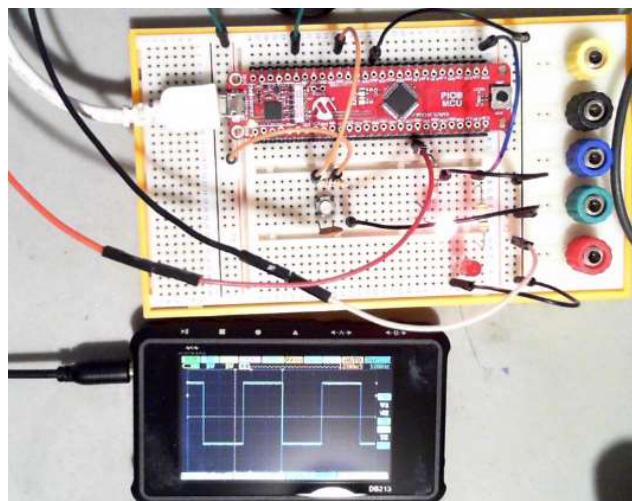
```

26   movlb 4H          ;bank4
27   bcf TRISC, 0, 1
28   bcf ANSEL0, 0, 1
29   bcf TRISD, 0, 1
30   bcf ANSELD, 0, 1
31   bsf TRISB, 0, 1
32   bcf ANSELB, 0, 1
33   movlw 80H
34   movwf PIE3, 1     ;TMROIE=1 (TMRO interrupt enabled)
35   movwf INTCON0, 1  ;GIE=1 (Global interrupt enabled)
36
37 inicio:           ;rutina principal
38   btfs PORTB, 0, 1 ;pregunto si pulse boton
39   bra S-2           ;no pulse
40   btg LATC, 0, 1    ;si pulse y basculo LED
41   btfs PORTB, 0, 1  ;pregunto si solte boton
42   bra S-2           ;no solte
43   bra inicio         ;si solte, retorno a inicio
44
45 TMRO_ISR:        ;rutina de interrupcion para TMRO
46   btg LATD, 0, 1    ;basculo LED
47   movlb 3H          ;bank3
48   movlw 0FEH
49   movwf TMROH, 1
50   movlw 17H
51   movwf TMROL, 1    ;carga de cuenta inicial a TMRO
52   movlb 4H          ;bank4
53   bcf PIR3, 7, 1
54   retfie
55
56 end upcino

```

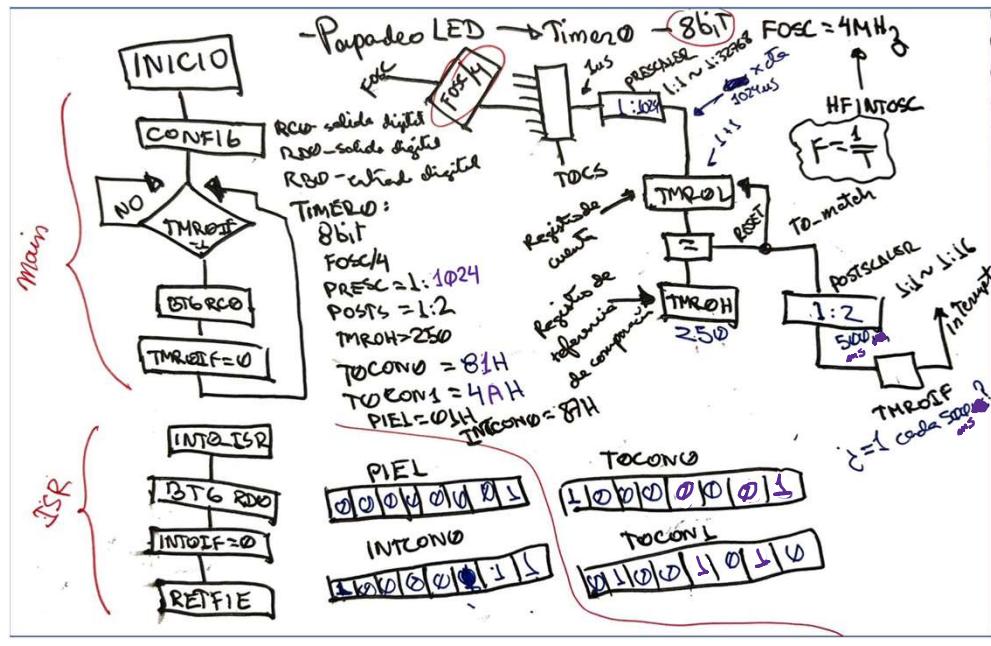
## Estrategia2: Timer0 en interrupción y rutina principal el Latch

- Funcionamiento



## Estrategia3: Timer0 en rutina principal y Latch con INT0

- En esta estrategia se ha cambiado el modo de trabajo del Timer0 a 8bit.
- Recordar que el evento que emite el TMRO en 8bit es debido a la igualdad entre TMROH y TMROL.
- Tener en cuenta que el INTOIE (habilitador de la INT0) se encuentra en el bit0 del registro PIE1 y el INTOIF (bandera de la INT0) se encuentra en el bit del registro PIR1



## Estrategia3: Timer0 en rutina principal y Latch con INT0

### • Código propuesto

```

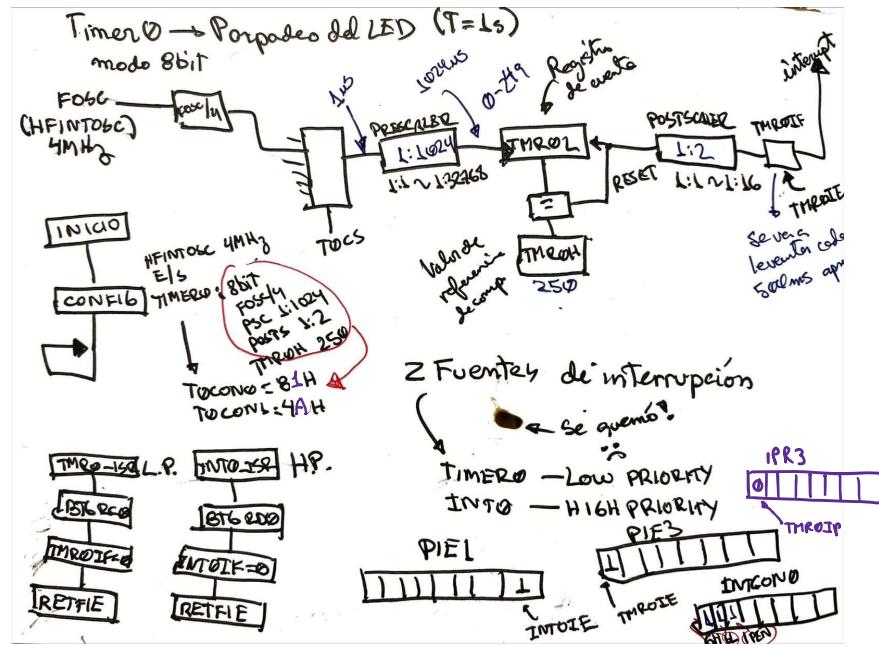
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6      ORG 000000H ;vector de reset
7      bra configuro
8
9      ORG 000008H ;vector de interrupcion
10     bra INT0_ISR
11
12     ORG 000100H
13 configuro:
14     movlb 0H ;bank0
15     movlw 60H
16     movwf OSCCON1, 1 ;hfintosc, 1:1
17     movlw 02H
18     movwf OSCFRQ, 1 ;hfintosc 4MHz
19     movlw 40H
20     movwf OSCEN, 1 ;hfintosc enabled
21     movlb 2H ;bank2
22     movlw 08H
23     movwf INT0PPS, 1 ;rb0 asignado a INT0
24     movlb 3H ;bank3
25     movlw 81H
26     movwf TOCON0, 1 ;tmr0 on, 8bit, posts
27     movlw 4AH
28     movwf TOCON1, 1 ;fosc4, presc 1:1024
29     movlw 250
30     movwf TMROH, 1 ;valor de referencia
31
32     bcf TRISD, 0, 1 ;rd0 salida
33     bcf ANSEL0, 0, 1 ;rd0 digital
34     bcf TRISC, 0, 1 ;rc0 salida
35     bcf ANSELC, 0, 1 ;rc0 digital
36     bsf TRISB, 0, 1 ;rb0 entrada
37     bcf ANSELB, 0, 1 ;rb0 digital
38     movlw 01H
39     movwf PIE1, 1 ;INT0 enabled
40     movlw 87H
41     movwf INTCON0, 1 ;GIE enabled, INT0EDG=1
42     bcf LATD, 0, 1 ;rd0=0
43
44     inicio:
45     btfss PIR3, 7, 1 ;pregunto si hubo evento en TMRO (TMROIF=1)
46     bra $-2 ;no hubo evento en TMRO (TMROIF=0)
47     btg LATC, 0, 1 ;si hubo evento, basculo rc0
48     bcf PIR3, 7, 1 ;bajo bandera TMROIF
49     bra inicio ;repito
50
51     INT0_ISR:
52     btg LATD, 0, 1 ;basculo rd0
53     bcf PIR1, 0, 1 ;bajo bandera INT0IF
54     retfie ;retorno a posicion antes de interrupcion
55
56     end upcino

```

**Nota:** No olvidar de colocar el bit de configuración MVECEN=OFF

## Estrategia4: Timer0 e INT0 en interrupciones

- En estrategia, la configuración del Timer0 es la misma que la de la estrategia 3: modo 8 bits.
- Se está considerando el uso de prioridades en las interrupciones (bits de configuración MVECEN=OFF), donde la interrupción del Timer0 pasará a la baja prioridad (vector 18H) mientras que la interrupción INT0 permanecerá en la alta prioridad (vector 8H).
- Tener en cuenta que el habilitador de interrupciones del Timer0 (TMROIE) se encuentra en el bit 7 del registro PIE3, y su bandera TMROIF se encuentra en el bit 7 del registro IPR3.
- El bit TMROIP que determina la prioridad de la interrupción del Timer0 se encuentra en el bit 7 del regístro IPR3



## Estrategia4: Timer0 e INT0 en interrupciones

- Código propuesto

```

1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  upcino:
6      ORG 000000H ;vector de reset
7      bra configuro
8
9      ORG 000008H ;vector de int hp
10     bra INTO_ISR
11
12     ORG 000018H ;vector de int lp
13     bra TMRO_ISR

15      ORG 000100H
16  configuro:
17      movlb 0H ;bank0
18      movlw 60H
19      movwf OSCCON1, 1
20      movlw 02H
21      movwf OSCFRQ, 1
22      movlw 40H
23      movwf OSCEN, 1
24      movlb 3H ;bank3
25      movlw 81H
26      movwf TOCON0, 1
27      movlw 4AH
28      movwf TOCON1, 1
29      movlw 250
30      movwf TMROH, 1
31      bcf IPR3, 7, 1 ;TMROIP=0 (low priority)
32      movlb 4H ;bank4
33      bsf PIE3, 7, 1 ;TMROIE=1
34      bsf PIE1, 0, 1 ;INTOIE=1
35      movlw 0E7H
36      movwf INTCON0, 1 ;GIEH=1, GIEL=1, IPEN=1
37      bcf TRISC, 0, 1 ;RC0 como salida
38      bcf ANSEL0, 0, 1 ;RC0 como digital
39      bcf TRISD, 0, 1 ;RD0 como salida
40      bcf ANSEL0, 0, 1 ;RD0 como digital
41      bsf TRISB, 0, 1 ;RB0 como entrada
42      bcf ANSEL0, 0, 1 ;RB0 como digital

44  inicio:
45      bra inicio
46
47
48  TMRO_ISR:
49      btg LATD, 0, 1 ;complementamos RC0
50      bcf PIR3, 7, 1 ;bajamos TMROIF
51      retfie
52
53  INTO_ISR:
54      btg LATD, 0, 1 ;complementamos RD0
55      bcf PIR1, 0, 1 ;bajamos INTOIF
56      retfie
57
58  end upcino

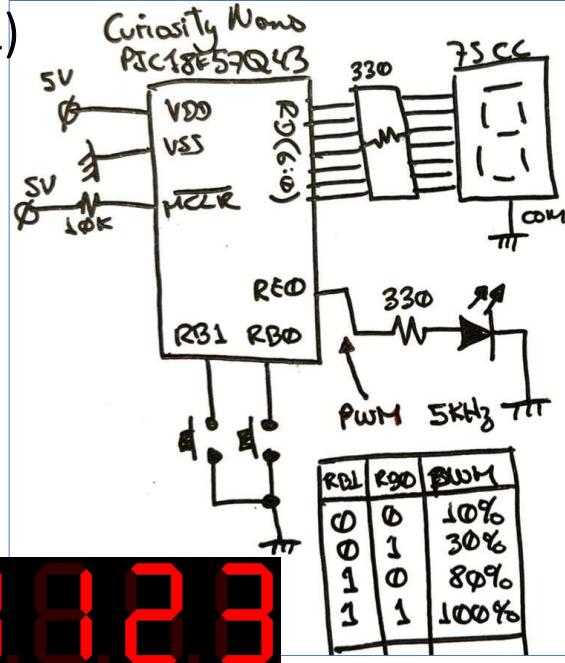
```

## ¿Multitarea?

- Ejecución de varias tareas a la vez en el microcontrolador
- Sistemas RTOS (Real Time Operating System), relacionado con lenguajes de alto nivel generalmente.
- En ensamblador la multitarea está relacionado con el uso de interrupciones.
- Ejecución de una instrucción ≠ ejecución de una tarea

## Asignación del LB2 EL57 (2024-1)

- Desarrollar el siguiente ejercicio de emisión de una señal PWM de 5KHz con Duty Cycle variable y configurado a través de una combinatoria en RB1 y RB0 según tabla.
- No olvidar de activar las pullup de RB1 y RB0.
- En el display de siete segmentos aparecerá la combinación seleccionada de duty cycle (0, 1, 2 ó 3)
- Para verificar la salida de PWM se usará un LED el cual cambiará de intensidad según combinatoria.
- Seguir las indicaciones de la actividad en el AV y activar sus webcams durante la evaluación



## Asignación del LB2 (2024-1)

```

1  PROCESSOR 18F57043
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  temporal EQU 500H ;GPR
6
7  upcino:
8      ORG 000000H ;Vector de reset
9      bra configuro
10
11     ORG 000300H ;tabla de decodificacion
12     ceroaltres: DB 3FH, 06H, 5BH, 4FH
13
14     ORG 000100H
15 configuro:
16     movlb 0H ;bank0
17     movlw 60H
18     movwf OSCCON1, l ;hfintosc, 1:l
19     movlw 02H
20     movwf OSCERQ, l ;hfintosc 4MHz
21     movlw 40H
22     movwf OSCEN, l ;hfintosc enabled
23     movlb 3H ;bank3
24     movlw 90H
25     movwf T0CON0, l ;tmr0 on, 16 bit, posts 1:l
26     movlw 40H
27     movwf T0CON1, l ;fosc/4 presc 1:l
28     movlb 4H ;bank4
29     bcf TRISE, 0, 1 ;REO salida
30     bcf ANSEL0, 0, 1 ;REO digital
31     clrf TRISD, l ;RD salida
32     clrf ANSEL0, l ;RD digital
33     setf TRISB, l ;RB1 y RB0 entradas
34     movlw 0FCH
35     movwf ANSELB, l ;RB1 y RB0 digitales
36     clrf TBLPTRU, l

37     movlw 03H
38     movwf TBLPTRH, l
39     clrf TBLPTRL, l ;TBLPTR apuntando a 300H
40     movlb 5H
41     clrf temporal, l
42
43 inicio: ;rutina principal
44     movlb 4H
45     movf PORTB, 0, 1 ;leo RB1 y RB0
46     andlw 03H ;enmascaramiento
47     movlb 5H
48     movwf temporal, l ;almacenar en GPR temporal
49     call deco7s ;visualizacion en el display
50     movlb 5H
51     movlw 0
52     cpfseq temporal, l ;combinacional 0?
53     bra siguiente ;no, sigue preguntando
54     bra estado_0 ;salta a combinacional 0
55 siguiente:
56     movlw 1
57     cpfseq temporal, l ;combinacional 1?
58     bra siguiente2 ;no, sigue preguntando
59     bra estado_1 ;salta a combinacional 1
60 siguiente2:
61     movlw 2
62     cpfseq temporal, l ;combinacional 2?
63     bra estado_3 ;salta a combinacional 3
64     bra estado_2 ;salta a combinacional 2
65
66 estado_0: ;duty cycle 10%
67     movlb 4H
68     bsf LATE, 0, 1 ;REO a uno
69     movlb 3H
70     movlw 0FFH
71     movwf TMROL, l
72     movlw 0ECH

73     movwf TMROL, l
74     movlb 4H
75     btfss PIR3, 7, 1
76     bra $-2
77     bcf PIR3, 7, 1 ;termino de TON (20us)
78     bcf LATE, 0, 1 ;REO a cero
79     movlb 3H
80     movlw 0FFH
81     movwf TMROH, l
82     movlb 4CH
83     movwf TMROL, l
84     movlb 4H
85     btfss PIR3, 7, 1
86     bra $-2
87     bcf PIR3, 7, 1 ;termino de TOF (180us)
88     bra inicio

89 estado_1: ;duty cycle 30%
90     movlb 4H
91     bsf LATE, 0, 1 ;REO a uno
92     movlb 3H
93     movlw 0FFH
94     movwf TMROH, l
95     movlw 0C4H
96     movwf TMROL, l
97     movlb 4H
98     btfss PIR3, 7, 1
99     bra $-2
100    bcf PIR3, 7, 1 ;termino de TON (60us)
101    bcf LATE, 0, 1 ;REO a cero
102    movlb 3H
103    movlw 0FFH
104    movwf TMROH, l
105    movlw 74H
106    movwf TMROL, l
107    movlb 4H
108

```

## Asignación del LB2 (2024-1)

```

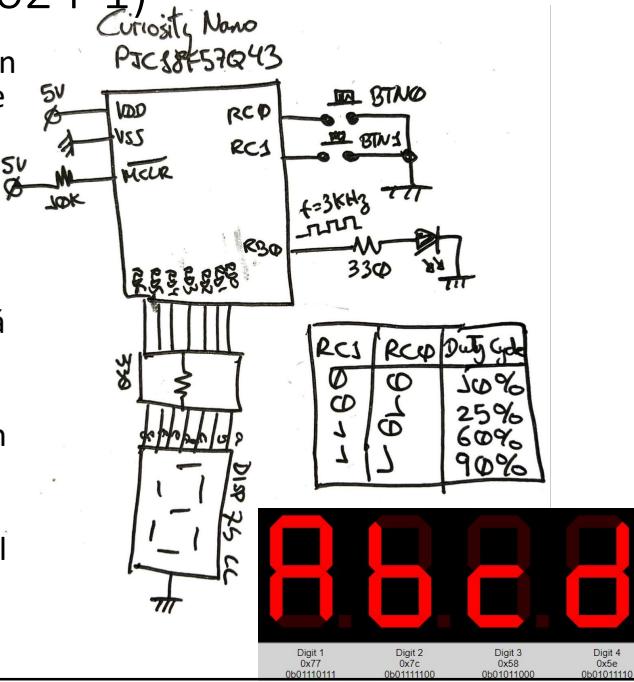
109     btfss PIR3, 7, 1
110     bra $-2
111     bcf PIR3, 7, 1 ;termino de TOF (140us)
112     bra inicio
113
114     estado_2: ;duty cycle 80%
115     movlb 4H
116     bsf LATE, 0, 1 ;REO a uno
117     movlb 3H
118     movlw 0FFH
119     movwf TMROH, l
120     movlw 060H
121     movwf TMROL, l
122     movlb 4H
123     btfss PIR3, 7, 1
124     bra $-2
125     bcf PIR3, 7, 1 ;termino de TON (160us)
126     bcf LATE, 0, 1 ;REO a cero
127     movlb 3H
128     movlw 0FFH
129     movwf TMROH, l
130     movlw 0DBH
131     movwf TMROL, l
132     movlb 4H
133     btfss PIR3, 7, 1
134     bra $-2
135     bcf PIR3, 7, 1 ;termino de TOF (40us)
136     bra inicio
137
138     estado_3: ;duty cycle 100%
139     movlb 4H
140     bsf LATE, 0, 1 ;REO a uno
141     bra inicio
142
143     deco7s: ;decodificacion 7seg
144     movlb 4H

145     movff temporal, TBLPTRL
146     TBLRD*
147     movff TABLAT, LATD
148     return
149
150     end upcino

```

# Asignación del LB2 EL52 (2024-1)

- Desarrollar el siguiente ejercicio de emisión de una señal PWM de 3KHz con Duty Cycle variable empleando como base de tiempo el Timer0 y configurado a través de una combinatoria en RC1 y RC0 según tabla.
  - No olvidar de activar las pullup de RC1 y RC0.
  - En el display de siete segmentos aparecerá la combinación seleccionada de duty cycle (A, B, C ó D)
  - Para verificar la salida de PWM se usará un LED el cuál cambiará de intensidad según combinatoria.
  - Seguir las indicaciones de la actividad en el AV y activar sus webcams durante la evaluación



Asignación del LB2 EL52 (2024-1)

```

1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5  temporal EQU 500H ;GPR
6
7  upcino:
8      ORG 000000H ;Vector de reset
9      bra configuro
10
11     ORG 000300H ;tabla de decodificacion
12     ceroaltres: DB 77H, 7CH, 58H, 5EH
13
14     ORG 000100H
15     configuro:
16         movlb OH ;bank0
17         movlw 60H
18         movwf OSCCON1, 1 ;hfintosc, 1:l
19         movlw 02H
20         movwf OSCFRC, 1 ;hfintosc 4MHz
21         movlw 40H
22         movwf OSCEN, 1 ;hfintosc enabled
23         movlb 3H ;bank3
24         movlw 90H
25         movwf T0CON0, 1 ;tmr0 on, 16 bit, posts 1:l
26         movlw 40H
27         movwf T0CON1, 1 ;fosc/4 presc 1:l
28         movlb 4H ;bank4
29         bcf TRISB, 0, 1 ;RB0 salida
30         bcf ANSELB, 0, 1 ;RB0 digital
31         clrf TRISD, 1 ;RD salida
32         clrf ANSELD, 1 ;RD digital
33         setf TRISC, 1 ;RC1 y RC0 entradas
34         movlw 0FCH
35         movwf ANSELC, 1 ;RC1 y RC0 digitales
36         clrf TBLPTR, 1
37         movlw 03H
38         movwf TBLPTR, 1
39         clrf TBLPTR, 1 ;TBLPTR apuntando a 300H
40         movlb 5H
41         clrf temporal, 1

43     inicio:          ;rutina principal
44         movlb 4H
45         movf PORTC, 0, 1 ;leo RCL y RC0
46         andlw 03H ;enmascaramiento
47         movlb 5H
48         movwf temporal, 1 ;almacenar en GPR temporal
49         call deco7s ;visualizacion en el disp
50         movlb 5H
51         movlw 0
52         cpfseq temporal, 1 ;combinacional 0?
53         bra siguiente ;no, sigue preguntando
54         bra estado_0 ;salta a combinacional 0
55     siguiente:
56         movlw 1
57         cpfseq temporal, 1 ;combinacional 1?
58         bra siguiente2 ;no, sigue preguntando
59         bra estado_1 ;salta a combinacional 1
60     siguiente2:
61         movlw 2
62         cpfseq temporal, 1 ;combinacional 2?
63         bra estado_3 ;salta a combinacional 3
64         bra estado_2 ;salta a combinacional 2
65
66     estado_0:          ;duty cycle 10%
67         movlb 4H
68         bsf LATB, 0, 1 ;RB0 a uno
69         movlb 3H
70         movlw 0FFH
71         movwf THROH, 1
72         movlw 0DFFH
73         movwf THROL, 1
74         movlb 4H
75         btfs PIR3, 7, 1
76         bra 5-2
77         bcf PIR3, 7, 1 ;termino de TON (33us)
78         bcf LATB, 0, 1 ;RB0 a cero
79         movlb 3H
80         movlw 0FEH
81         movwf THROH, 1
82         movlw 0D4H
83         movwf THROL, 1
84         movlb 4H

85         btfs PIR3, 7, 1
86         bra 5-2
87         bcf PIR3, 7, 1 ;termino de TOF (300us)
88         bra inicio
89
90     estado_1:          ;duty cycle 30%
91         movlb 4H
92         bsf LATB, 0, 1 ;RB0 a uno
93         movlb 3H
94         movlw 0FFH
95         movwf THROH, 1
96         movlw 0A0H
97         movwf THROL, 1
98         movlb 4H
99         btfs PIR3, 7, 1
100        bra 5-2
101        bcf PIR3, 7, 1 ;termino de TON (83us)
102        bcf LATB, 0, 1 ;RB0 a cero
103        movlb 3H
104        movlw 0FFH
105        movwf THROH, 1
106        movlw 0E6H
107        movwf THROL, 1
108        movlb 4H
109        btfs PIR3, 7, 1
110        bra 5-2
111        bcf PIR3, 7, 1 ;termino de TOF (250us)
112        bra inicio
113
114        btfs PIR3, 7, 1
115        movlb 4H
116        bsf LATB, 0, 1 ;RB0 a uno
117        movlb 3H
118        movlw 0FFH
119        movwf THROH, 1
120        movlw 22H
121        movlb 3H
122        movlw 0FFH
123        movwf THROH, 1
124        bra 5-2
125        bcf PIR3, 7, 1 ;termino de TON (222us)
126        bcf LATB, 0, 1 ;RB0 a cero

```

## Asignación del LB2 EL52 (2024-1)

```

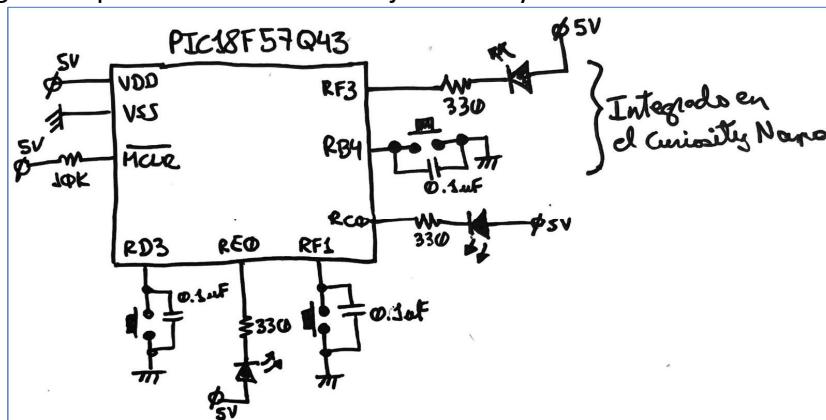
126      bcf LATB, 0, 1 ;RBO a cero
127      movlb 3H
128      movlw 0FFH
129      movwf TMR0H, 1
130      movlw 91H
131      movwf TMR0L, 1
132      movlb 4H
133      btfs PIR3, 7, 1
134      bra $-2
135      bcf PIR3, 7, 1 ;termino de TOF (illus)
136      bra inicio
137
138      estado 3: ;duty cycle 100%
139      movlb 4H
140      bcf LATB, 0, 1 ;RBO a uno
141      movlb 3H
142      movlw 0FEH
143      movwf TMR0H, 1
144      movlw 0D4H
145      movwf TMR0L, 1
146      movlb 4H
147      btfs PIR3, 7, 1
148      bra $-2
149      bcf PIR3, 7, 1 ;termino de TON (300us)
150      bcf LATB, 0, 1 ;RBO a cero
151      movlb 3H
152      movlw 0FFH
153      movwf TMR0H, 1
154      movlw 0DFH
155      movwf TMR0L, 1
156      movlb 4H
157      btfs PIR3, 7, 1
158      bra $-2
159      bcf PIR3, 7, 1 ;termino de TOF (33us)
160      bra inicio
161

```

	deco7s: ;decodificacion 7seg
162	movlb 4H
163	movff temporal, TBLPTRL
164	TBLRD*
165	movff TABLAT, LATD
166	return
167	
168	
169	end upcino

## Ejemplo 2025-1

- Basculación de tres LEDs empleando las tres interrupciones externas INT0, INT1 e INT2, en modo vectorizado (MVECEN ON)
  - INT0 ingresará por botón activo en bajo en RB4 y basculará LED activo en bajo en RF3
  - INT1 ingresará por botón activo en bajo en RD3 y basculará LED activo en bajo en RE0
  - INT2 ingresará por botón activo en bajo en RF1 y basculará LED activo en bajo en RC0



# Ejemplo 2025-1

<pre> 6      PROCESSOR 18F57Q43 7      #include "cabecera.inc" 8 9      PSECT upcino, class=CODE, reloc=2 10     upcino: 11         ORG 000000H ;vector de RESET 12         goto configuro 13 14         ORG 000010H ;vector de la INT0 (2*08H) 15         DW INT0_ISR shr 2 16 17         ORG 000060H ;vector de la INT1 (2*30H) 18         DW INT1_ISR shr 2 19 20         ORG 0000A0H ;vector de la INT2 (2*50H) 21         DW INT2_ISR shr 2 22 23 ;Cada rutina de interrupción debe de tener su propio PSECT 24         PSECT INT0_ISR, class=CODE, reloc=4 25     INT0_ISR: 26         bcf PIR1, 0, 1 ;abajamos bandera INTOIF 27         btg LATF, 3, 1 ;basculamos LED que esta en RF3 28         retfie 29 30         PSECT INT1_ISR, class=CODE, reloc=4 31     INT1_ISR: 32         bcf PIR6, 0, 1 ;abajamos bandera INT0IF 33         btg LATC, 0, 1 ;basculamos LED que esta en RE0 34         retfie 35 36         PSECT INT2_ISR, class=CODE, reloc=4 37     INT2_ISR: 38         bcf PIR10, 0, 1 ;abajamos bandera INTOIF 39         btg LATC, 0, 1 ;basculamos LED que esta en RC0 40         retfie </pre>	<pre> 42     ;Inicio de codigo de usuario 43         PSECT code 44         ORG 000100H 45         configuro: 46             movlw 0H 47             movlw 60H 48             movwf OSCCON1, 1 49             movlw 02H 50             movwf OSCFRC, 1 51             movlw 40H 52             movwf OSCEN, 1 53             movlw 2H 54             movlw 0CH 55             movwf INT0PPS, 1 56             movlw 1BH 57             movwf INT1PPS, 1 58             movlw 29H 59             movwf INT2PPS, 1 60             movlw 4H 61             bcf TRISF, 3, 1 ;RF3 salida 62             bcf ANSEL, 3, 1 ;RF3 digital 63             bcf TRISE, 0, 1 ;RE0 salida 64             bcf ANSELE, 0, 1 ;RE0 digital 65             bcf TRISC, 0, 1 ;RC0 salida 66             bcf ANSELC, 0, 1 ;RC0 digital 67             bcf ANSELB, 4, 1 ;RB4 digital (ya es entrada) 68             bcf WPUB, 4, 1 ;RB4 pullup activo 69             bcf ANSELD, 3, 1 ;RD3 digital (ya es entrada) 70             bcf WPUD, 3, 1 ;RD3 pullup activo 71             bcf ANSELF, 1, 1 ;RF1 digital (ya es entrada) 72             bcf WPUF, 1, 1 ;RF1 pullup activo 73             bcf LATF, 3, 1 ;LED apagado al inicio 74             bsf LATC, 0, 1 ;LED apagado al inicio 75             bsf LATC, 0, 1 ;LED apagado al inicio 76             clrf IVTBASEU, 1 </pre>	<pre> 77             clrf IVTBASEH, 1 ;forzar IVTBASE a cero4 78             clrf IVTBASEL, 1 ;activando INT0 con INTOIE=1 79             bcf PIE1, 0, 1 ;bajo bandera INTOIF 80             bcf PIR1, 0, 1 ;activando INT1 con INT1IE=1 81             bcf PIR6, 0, 1 ;bajo bandera INT1IF 82             bcf PIR10, 0, 1 ;bajando INT2 con INT2IE=1 83             bcf PIE10, 0, 1 ;bajo bandera INT2IF 84             bcf INTCON0, 1, 1 ;INT0EDGE=0 para detección de flanco negativo 85             bcf INTCON0, 2, 1 ;INT2EDGE=0 para detección de flanco negativo 86             bcf INTCON0, 7, 1 ;GIE=1 habilitando el interruptor global 87 88 ;Rutina principal: no hace nada 89         inicio: 90             bra inicio 91 92         end upcino 93 94 </pre>
--	--	--

Fin de la sesión