Capacitación en Tecnología de Alto Nivel

MANUAL DEL PIC 16F87X v 1.0

MICROCHIP

PRINCIPALES CARACTERÍSTICAS

Se enumeran las prestaciones y dispositivos especiales de los PIC16F87X.

- Procesador de arquitectura RISC avanzada
- Juego de solo 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción, menos las de salto que tardan dos.
- Hasta 8K palabras de 14 bits para la Memoria de Programa, tipo FLASH en los modelos 16F876 y 16F877 y 4KB de memoria para los PIC 16F873 y 16F874.
- Hasta 368 Bytes de memoria de Datos RAM.
- Hasta 256 Bytes de memoria de Datos EEPROM.
- Pines de salida compatibles para el PIC 16C73/74/76/77.
- Hasta 14 fuentes de interrupción internas y externas.
- Pila de 8 niveles.
- Modos de direccionamiento directo e indirecto.
- Power-on Reset (POP).
- Temporizador Power-on (POP) y Oscilador Temporizador Start-Up.
- Perro Guardián (WDT).
- Código de protección programable.
- Modo SLEEP de bajo consumo.
- Programación serie en circuito con dos pines, solo necesita 5V para programarlo en este modo.
- Voltaje de alimentación comprendido entre 2 y 5,5 V.
- Bajo consumo: < 2 mA valor para 5 V y 4 Mhz 20 µA para 3V y 32 M < 1 µA en standby.

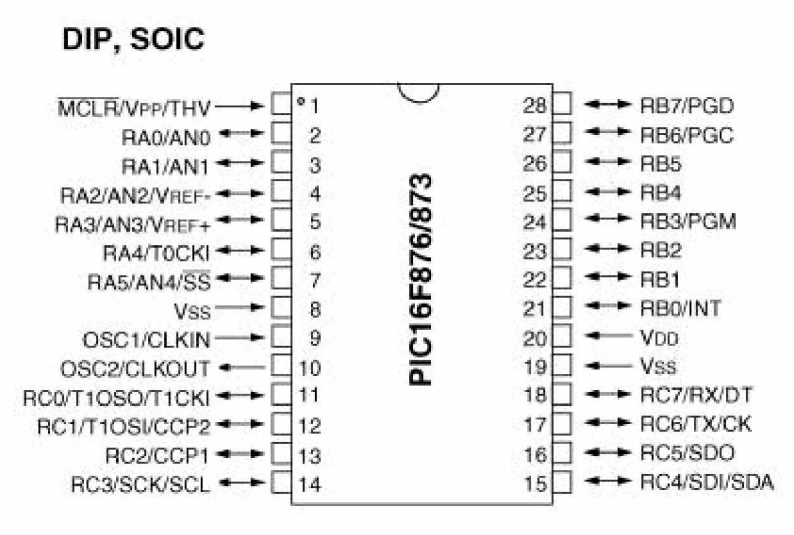


Figura 1.-Encapsulado DIP,SOIC de 28 pines

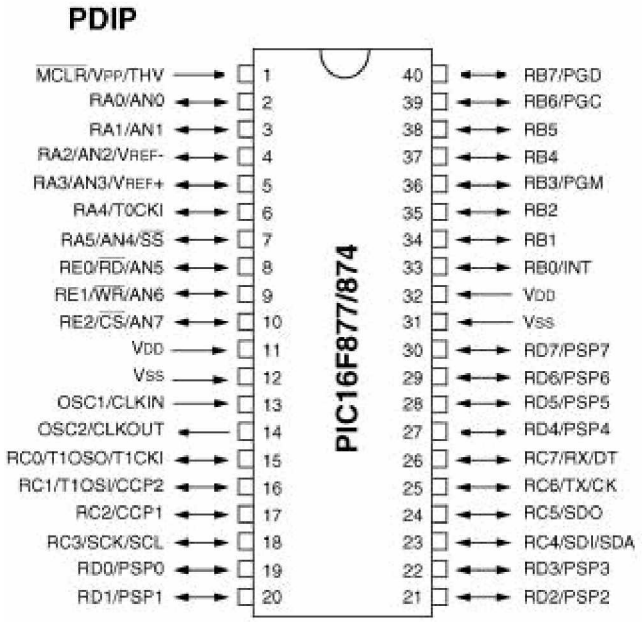


Figura 2.- Encapsulado PDIP de 40 pines

DISPOSITIVOS PERIFÉRICOS

- Timer0: Temporizador-contador de 8 bits con preescaler de 8 bits
- Timer1: Temporizador-contador de 16 bits con preescaler que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con preescaler y postscaler.
- Dos módulos de Captura, Comparación, PWM (Modulación de Ancho de Pulsos).
- Conversor A/D de 1 0 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I^2C (Master/Slave).
- USART/SCI (Universal Syncheronus Asynchronous Receiver Transmitter) con 9 bit.

- Puerto Paralelo Esclavo (PSP) solo en encapsulados con 40 pines

DIFERENCIAS ENTRE, LOS MODELOS DE 28 Y LOS DE 40 PINES

El PIC 16F873 y el 876 tienen 28 pines, mientras que el PIC 1 6F874 y 877 tienen 40.
Nos centraremos en el PIC 16F873 y las diferencias que tiene con sus hermanos son mínimas y se detallan a continuación:
Los modelos de 40 pines disponen de 5 Puertos de E/S: A, B, C, D y E, mientras que los de 28 solo tienen 3 Puertos: A, B y C.
Los modelos de 40 pines tienen 8 canales de entrada al Conversor A/D, mientras que los de 28 solo tienen 5 canales.
Sólo poseen el Puerto Paralelo Esclavo los PIC 16F87X de 40 pines.

EL PIC 16F87X

Bajo el nombre de esta subfamilia de microcontroladores, actualmente encontramos cuatro modelos: EL PIC 16F873/4/6 y 7. Estos microcontroladores disponen de una memoria e programa FLASH de 4 a 8 KBytes de 14 bits, considerablemente superior frente al PIC 16F84 en el que solo disponíamos de l Kbyte de 14 bits.

De los microcontroladores indicados, el 16F873 y el 16F876 son de 28 pines, mientras que 16F874 y el 16F877 tienen 40 pines, lo que les permite disponer de hasta 33 líneas de E/S. En su arquitectura además incorporan:

- Varios Timer
- USART
- Bus I2C

En la Tabla 1 se muestran las características comparativas más relevantes de esta familia de microcontroladores:

Características	16F873	16F874	16F876	16F877
Frecuencia Máxima	DC-20Mhz	DX-20Mhz	DX-20Mhz	DX-20Mhz
Memoria de programa FLASH Palabra de 14 bits	4KB	4KB	8KB	8KB
Posiciones RAM de datos	192	192	368	368
Posiciones EEPROM de datos	128	128	256	256
Ports E/S	A, B y C	A, B, C y D	A, B y C	A, B, C y D
Nº de Pines	28	40	28	40
Interrupciones	13	14	13	14
Timers	3	3	3	3
Módulos CCP	2	2	2	2

Comunicaciones Serie	MSSP, USART	MSSP,USART	MSSP,USART	MSSP, USART
Comunicación Paralelo	-	PSP	-	PSP
Líneas de entrada en Convertidor A/D de 10 bits	5	8	5	8
Juego de Instrucciones	35 instrucciones	35 instrucciones	35 instrucciones	35 instrucciones
Longitud de la instrucción	14 bits	14 bits	14 bits	14 bits

Tabla.- Comparación de entre los PIC de la Familia 16F87X

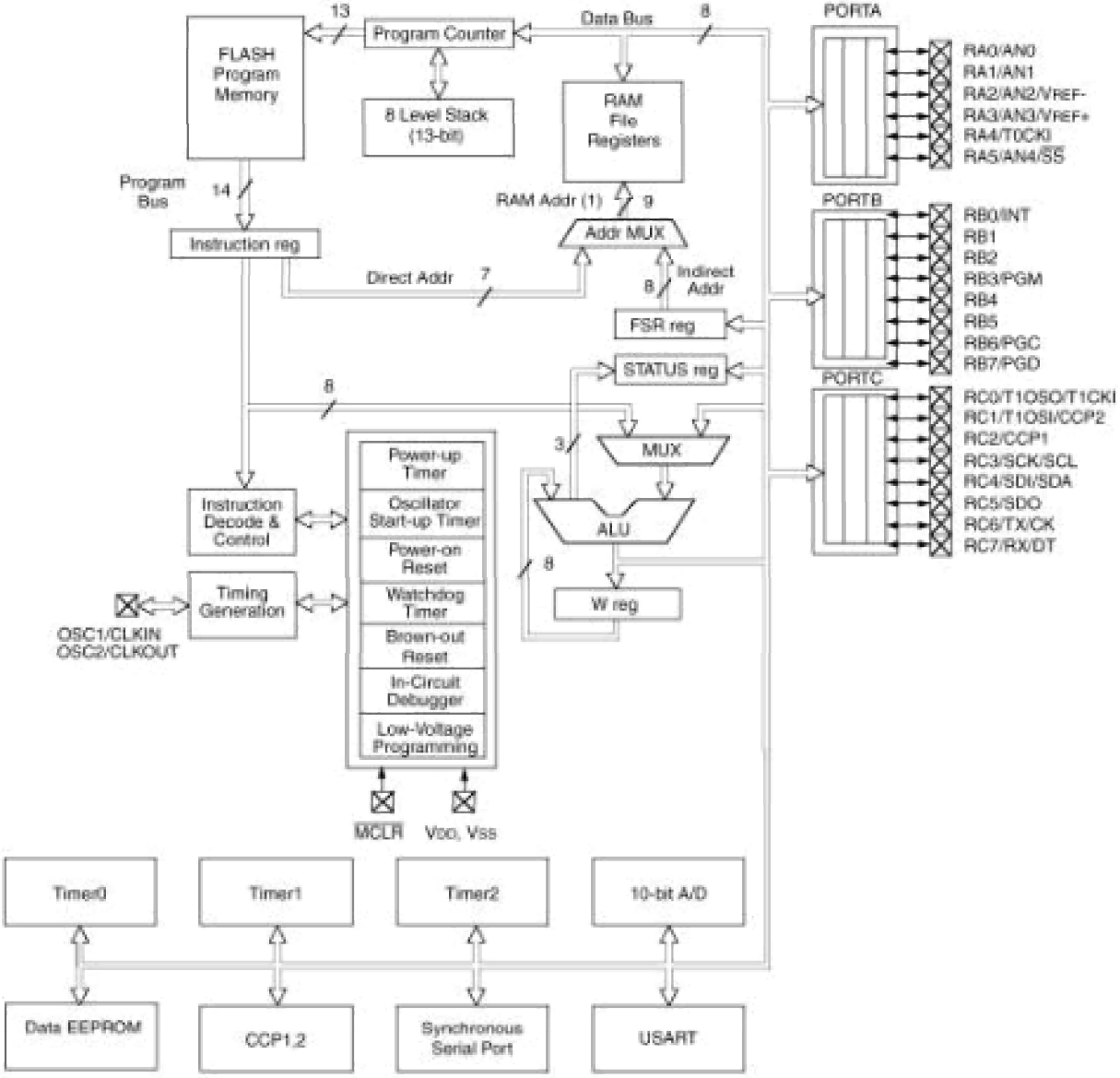
DESCRIPCIÓN GLOBAL DEL DISPOSITIVO

Este documento contiene la información específica del dispositivo. Se puede encontrar información adicional del PICmicro™ en el Manual de Referencia de los PIC de la gama media, (DS33023) que se puede obtener en el *website* de ***Microchip***. El Manual de referencia debe ser considerado un documento complementario a estos datos, y recomendable leerlo para entender mejor la arquitectura del dispositivo y el funcionamiento de los módulos periféricos.

La familia consta de cuatro dispositivos (PIC16FS73, PIC16F874, PIC16F876 y PIC 16F877) en estas hojas de datos. Los PIC 1 6F876/873 entran en el bloque de dispositivos encapsulados en 28-pines y los PIC 1 6F877/874 entran en el bloque de dispositivos encapsulados en 40 pines. Los dispositivos de 28-pines no tienen implementado el puerto paralelo esclavo.

Las siguientes figuras corresponden a los diagramas de bloque de los dispositivos ordenados por el número de pines; El 28-pines en la Figura 3 y los de 40-pines en. la Figura 4. En las Tablas 2 y 3 hay un listado de los mismos.

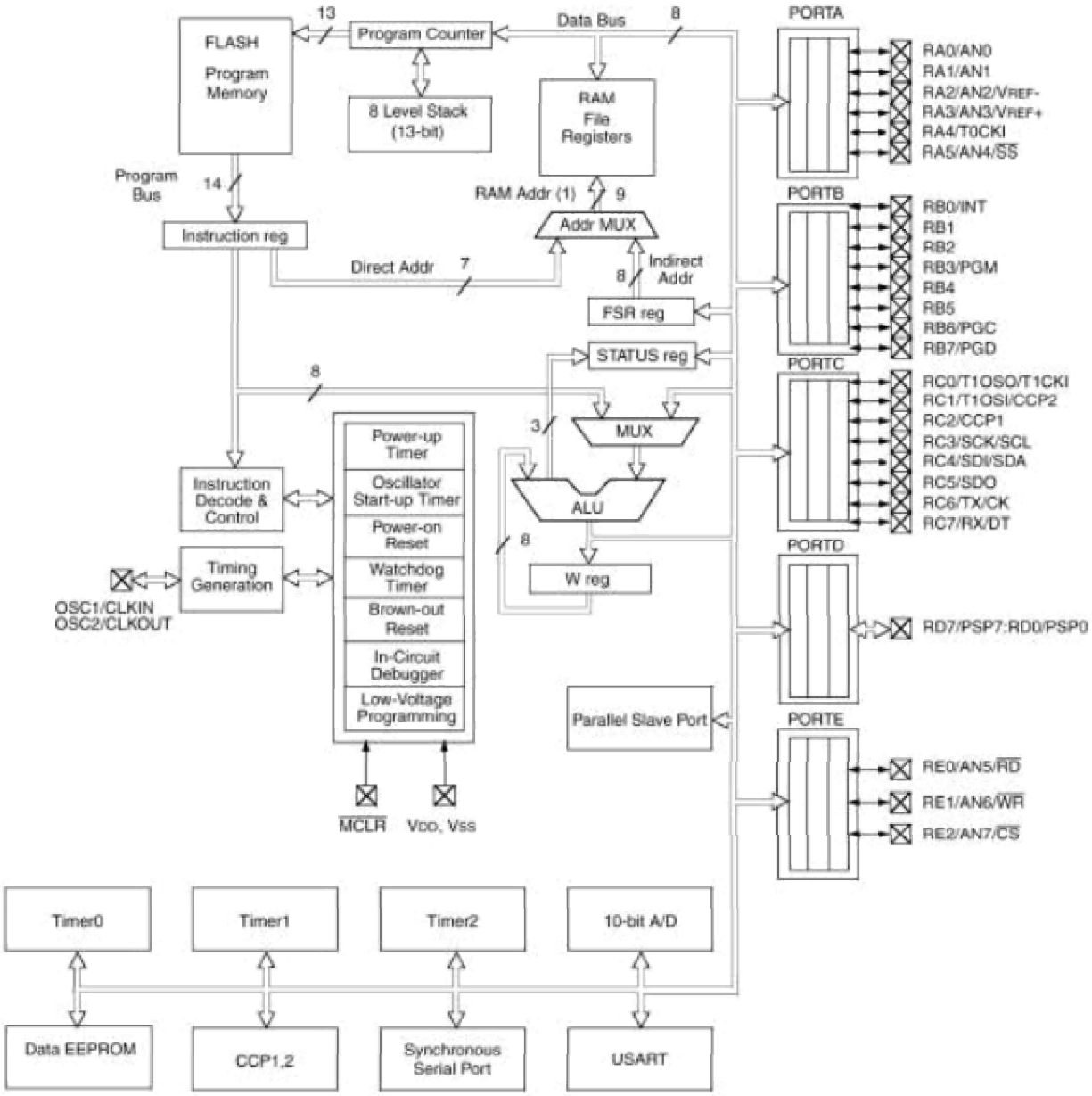
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

Figura .- Diagrama de bloques del PIC16F873 Y DEL pic16F876

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

Figura .- Diagrama de Bloques del PIC16F874 y 16F877

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input
RB4	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	I/O	TTL	Interrupt on change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Tabla.- Descripción de los pines del PIC16F873 y PIC16F876

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS(4)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP/THV	1	2	18	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input
RB4	37	41	14	I/O	TTL	Interrupt on change pin.
RB5	38	42	15	I/O	TTL	Interrupt on change pin.
RB6/PGC	39	43	16	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Tabla.- Descripción de los Pines del PIC 16F874 y PIC16F877

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Descripción de los pines de los PIC 16F874 y 16F877 (Continuación Tabla 3)

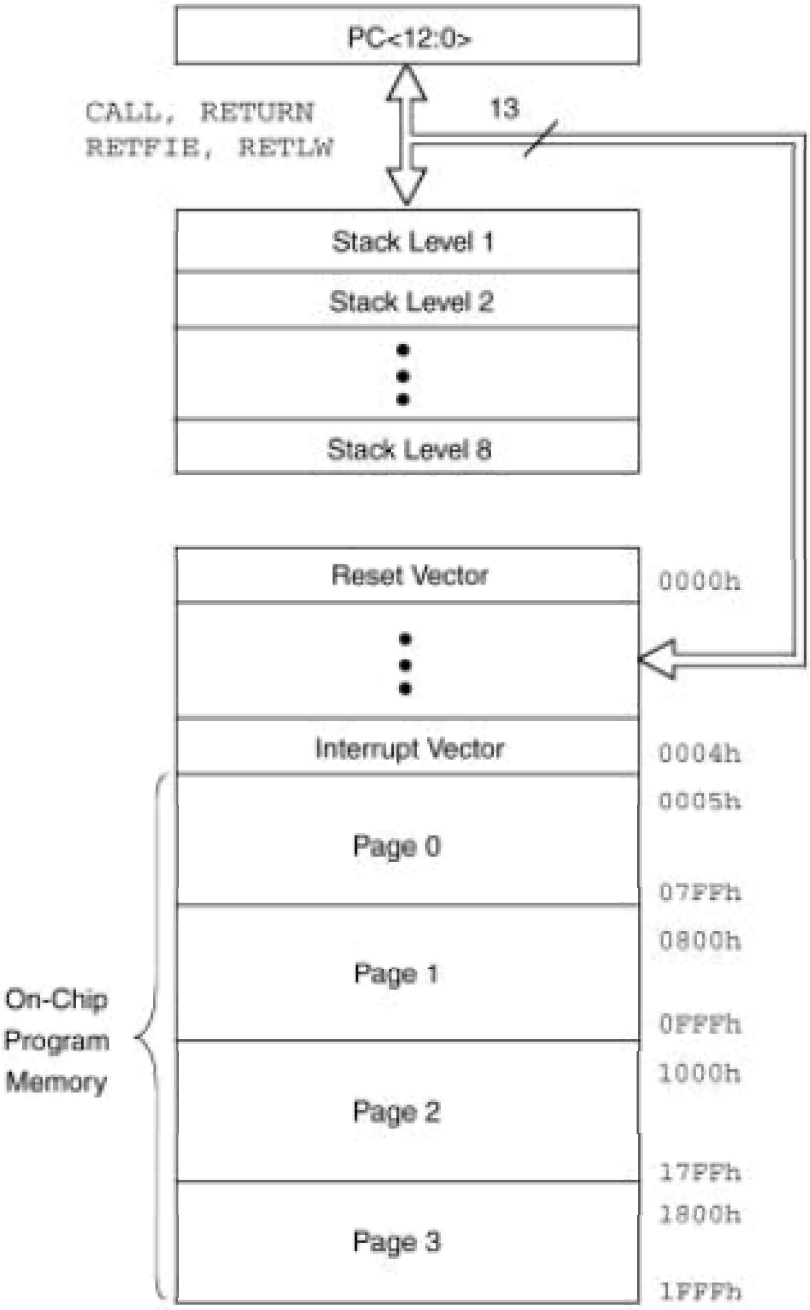
ORGANIZACIÓN DE LA MEMORIA

Hay tres bloques de memoria en cada uno de estos PICmicro. La Memoria de Programa y la Memoria de Datos que tienen los buses separados para poder permitir el acceso simultáneo a estos dos bloques. El tercer bloque que la Memoria de datos EEPROM.

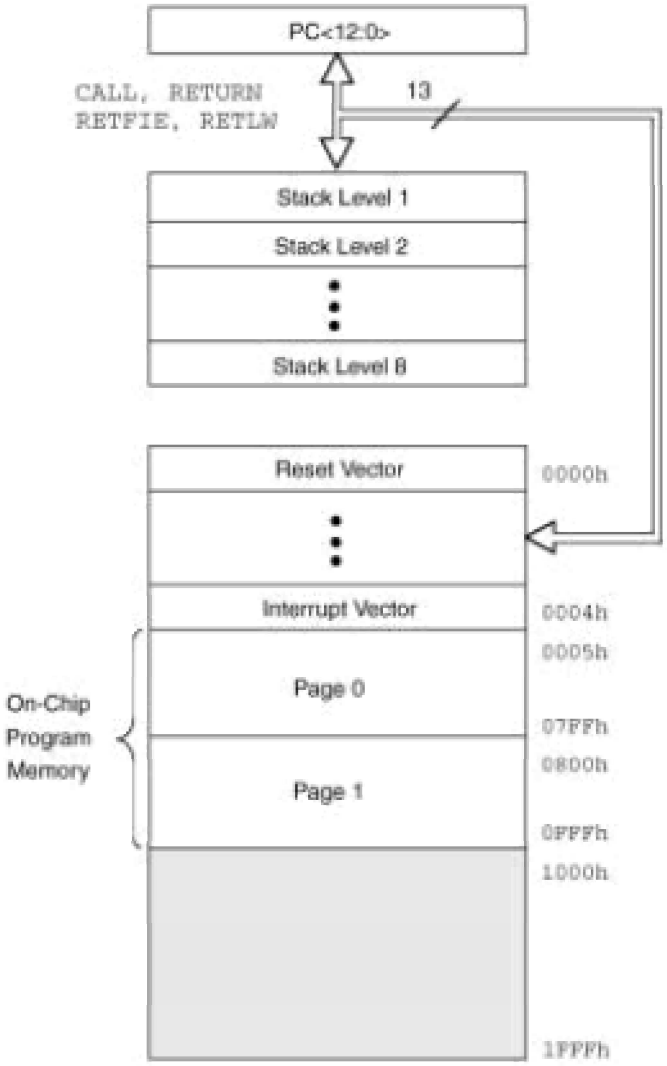
La información adicional sobre la memoria del dispositivo puede encontrarse en el manual de referencia de los PICmicros de gama media (DS33023).

Organización de la Memoria de programa

Los dispositivos de PIC 16F87X tienen un contador de programa de 13-bits capaz de direccionar 8Kxl4 posiciones de memoria. Los dispositivos de PIC16F877/876 tienen 8K x 14 posiciones de memoria de programa tipo FLASH y el PIC 1 6F873/874 tienen 4Kx 14. El vector de Reset está en la posición de memoria 0000h y el vector de interrupción está en la posición de memoria 0004h.



Mapa de Memoria y STACK del PIC16F877/878



Mapa de Memoria y SACK del PIC16F873/878

Organización de la Memoria de datos

La memoria de los datos se divide en los múltiples bancos que contiene los Registros del Propósito Generales y Los Registros de la Funciones especiales Los bit RP1 (STATUS <6> y RP0 (el ESTADO <5> seleccionan cada uno de estos bancos, de acuerdo a la siguiente tabla:

RP1	RP0	Banco
0	0	Banco 0
0	1	Banco 1
1	0	Banco 2
1	1	Banco 3

Tabla.- Selección de los bancos de memoria RAM con RP0 y RP1

En cada banco hay 7Fh posiciones de memoria (128 bytes). Las posiciones más bajas están, reservadas para los Registros de Funciones Especiales. Por

encima de los Registros de Funciones Especiales se encuentran los Registros de Propósito General, que se utilizan como posiciones de memoria RAM estática. Todos están estructurados en bancos. Algunos Registros de Funciones Especiales están reflejados en varios bancos para reducir el código y tener un acceso más rápido.

LECTURA Y ESCRITURA DE LAS MEMORIAS EEPROM Y FLASH

En esta familia de microcontroladores tanto la memoria EEPROM de datos como la memoria de programa FLASH puede ser modificada sin necesidad de utilizar un programador exterior y a la tensión nominal de VDD. Es decir, un programa dinámicamente puede generar información que se puede grabar en la memoria FLASH.

Por lo tanto podemos utilizar esta propiedad para que:

La propia aplicación se puede programar según las condiciones externas

Es posible ampliar el área de la memoria de datos no volátil EEPROM con posiciones libres de la memoria FLASH.

Se dispone de seis registros especiales para leer y escribir sobre la memoria no volátil, estos registros son: EECON1, EECON2, EEDATA, EEDATH, EEADR y EEADRH. Para direccionar los 128 posiciones de memoria EEPROM de 8 bits del PIC16F873 y 16F874 o las 256 posiciones del PIC16FS76 y 16FS77 basta con 8 bit, por ello para escribir o leer en la memoria EEPROM solo hacen faltan el registro EEADR para direccionar la posición y el registro EEDATA para colocar el dato leído o escrito. Sin embargo para poder escribir o leer datos en la memoria FLASH que puede tener hasta 8K palabras de 14 bits hacen falta dos registros para direccionar la posición de memoria, por ello se utiliza el registro EEADR concatenado con el registro EEADRH que contiene la parte alta de la palabra de direccionamiento de memoria. De forma similar se utilizan los registros EEDATA concatenado con el registro EEADRH que contiene los 6 bit de mayor peso de las palabras de 14 bits.

Además para controlar el proceso de lectura y escritura de la memoria EEPROM y FLASH se dispone de dos registros: el EECON1 y el EECON2.

REGISTROS EECON1 (Dirección 18Ch) y EECON2

Seguidamente se describen los bit de control de registro EECON1

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	**---**	**---**	**---**	**WRERR**	**WREN**	**WR**	**RD**
Bit 7							Bit 0

bit 7: **EEPGD**: Programa / bit de selección de datos EEPROM

1= Acceso a la memoria de programa FASH

0= Acceso a la memoria de datos EEPROM

(No se puede cambiar mientras la lectura o escritura está en proceso)

bit 6:4:**No implementados**: Se leen como “0”

bit 3: **WRERR**: Flag de error de escritura en la EEPROM

1 = El proceso de lectura se ha producido prematuramente

(se ha producido un Reset por MCLR o un WDT durante el proceso)

0 = Se ha producido el proceso de escritura con éxito

bit 2: **WREN**: bit de habilitación de escritura

1 = Permite inicializar el ciclo de escritura

0 = Inhibe la escritura

bit 1: **WR**: bit de inicio de escritura

1 = Cuando se le pone a 1 comienza el ciclo de escritura de la memoria no volátil. (El bit se pone de nuevo a cero por hardware cuando la escritura se completa).

0 = Toma este valor cuando completa el ciclo de escritura de la memoria no volátil.

bit 0: **RD**: bit de inicio de lectura

1 = Cuando se le pone a 1 se inicia un ciclo de lectura. El bit **RD** se pone a cero por hardware.

0 = no ha comenzado el ciclo de lectura de la memoria no volátil.

El registro EECON2, no está implementado físicamente y sólo se utiliza para la operación de escritura, de igual forma que se hace con el PIC 16F84, es decir, antes de iniciar la escritura de un dato en la memoria, se debe escribir en el registro EECON2 primero el dato 55h y posteriormente el dato AAh.

Seguidamente se muestran una serie de subrutinas para la escritura y lectura de la memoria no volátil.

SUBRUTINA PARA LA ESCRITURA DE LA MEMORIA EEPROM DE DATOS

```
ESCRIBE_EEPROM
        bsf     STATUS, RPl         ;Selección del banco 2 de memoria
        bcf     STATUS, RP0
        movlw   DATA_EE_ADDR        ;Dirección de la EEPROM a la que
        movwf   EEADR               ;se quiere acceder
        movlw   DATA_EE_DATA        ;Dato que se quiere escribir
        movwf   EEDATA
        bsf     STATUS, RP0         ;Selección del banco 3
        bcf     EECON1, EEPGD       ;Selección de acceso a la EEPROM
        bsf     EECON1, WREN        ;Habilita la escritura en la EEPROM
        bcf     INTCON, GIE         ;Se Prohíben las interrupciones
        movlw   55h
        movwf   EECON2              ;Escribe 55h en EECON2
        movlw   AAh;
        movwf   EECON2              ;Escribe AAh en EECON2
        bsf     EECON1, WR          ;Se da la orden de escritura
        bsf     INTCON, GIE         ;Se habilitan las interrupciones
        sleep   ;Se espera hasta que se produzca la interrupción de fin de escritura
        bcf     EECON1, WREN        ;Prohibir la escritura de nuevos datos
        returm
```

SUBRUTINA PARA LA LECTURA DE LA MEMORIA EEPROM DE DATOS

```
LEE_EEPROM
        bsf         STATUS,RP1
        bcf         STATUS,RP0          ;Selección del banco 2 de memoria
        movf        ADDR_L,W            ;Dirección a leer de EEPROM
        movwf       EEADR
        bsf         STATUS,RP0          ;Selección del banco 3 de memoria
        bcf         EECON1,EEPGD        ;Se selecciona el acceso a EEPROM
        bsf     EECON1,RD           ;Se da la orden de efectuar
        bcf         STATUS,RP0  ; Selecciona el banco 2 de memoria
        movf        EEDATA_L    ;Se recoge el dato leído y se deja en DATA_L
         returm
```

SUBRUTINA PARA LA ESCRITURA DE LA MEMORIA FLASH DE CÓDIGO

```
ESCRIBE_FLASH
        bsf     STATUS,RP1
        bcf         STATUS,RP0          , Selección de Banco 2 de memoria
        movlw       ADDRH               ;Se escribe en EEADRH la parte alta de la
        movwf       EEADRH              ;dirección a escribir
        movlw       ADDRL               ;Se escribe en EEADR la parte baja de la
        movwf       EEADR               ; dirección a escribir
        movlw       DATA_H              ;Se mete en EEI:)ATH la parte alta
        movwf       EEDATH              ;del dato a escribir
        movlw       DATA_L              ;Se mete en EEDAL la parte baja
        movwf       EEDATA              ;del dato a escribir
```

```
        bsf         STATUS,RP0          ;Selección del banco 3 de memoria

        bsf         EECON1, EEPGD       ;Selección de acceso a la memoria FLASH

        bsf         EECON1, WREN        ;Se habilita la escritura en la memoria
FLASH

        bcf         INTCON, GIE         ;Se prohíben las interrupciones

        movlw       55h                 ;Se escriben los datos 55h y AAh en el
registro

        movwf       EECON2      ;EECON2 tal y como recomienda el fabricante

        movlw       AAh

        movwf       EECON2

        bsf         EECON1, WR          ;Se da la orden de escritura

        nop         ;El microcontrolador ignora estas dos instrucciones

        nop

        bsf         INTCON, GIE         ;Se habilitan las interrupciones

        bcf         EECON1, WREN        ;Se prohíbe la escritura de nuevos datos

        returm
```

SUBRUTINA PARA LA LECTURA DE LA MEMORIA FLASH DE CÓDIGO

```
LEE_FLASH

        bsf         STATUS,RP1

        bcf         STATUS,RP0          ;Selección del banco 2 de memoria

        movlw       ADDRH               ;Se mete en EEADRH la parte alta

        movwf       EEADRH              ;de la dirección a leer

        movlw       ADDRL               ;Se mete en EEADRL la parte baja

        movwf       EEADR               ;de la dirección a leer

        bsf         STATUS,RP0          ;Selección del banco 3 de memoria
```

```
bsf     EECON1, EEPGD   ;Selecciona el acceso a memoria FLASH
bsf     EECON1, RD      ;Orden de lectura de la memoria FLASH
                        ;Tarda tres ciclos en tener el dato, por lo
                        que
nop                     ;cualquier instrucción es ignorada
nop
bcf     STATUS,RPO      ;Selección del banco 2 de memoria
movf    EEDATA,W        ;Se recoge la parte baja del dato leído
movf    DATO L
movf    EEDATH, W       ;Se recoge la parte alta del dato leído
movf    DATO_H
returm
```

La operación de escritura de esta memoria dura aproximadamente 2 ms. El fabricante recomienda la verificación de los datos escritos en las memorias EEPROM y FLASH para comprobar que su grabación se ha realizado correctamente.

Para evitar escrituras indeseables en la EEPROM motivadas por espurios en la inicialización del microcontrolador, se controla el bit **WREN** del registro **EECON1**, prohibiendo cualquier operación de escritura mientras duran los 72 ms que temporiza el Timer Power-up. En el caso de la memoria FLASH se debe poner a 0 el bit **WRT** de la Palabra de Configuración, que solo puede escribirse desde un grabador externo.

Dependiendo del valor del bit **WRT** y de los bit de Protección de código CP1 y CP0, de la Palabra de Configuración, se consiguen diversas alternativas de protección contra escritura de la memoria FLASH.

CONFIGURACIÓN DE BITS			POSICIONES DE FLASH	LECTURA INTERNA	ESCRITURA NTERNA	LECTURA ICSP	ESCRITURA ICSP
CP1	CP0	WRT					
0	0	X	Toda la memoria de Programa	SI	NO	NO	NO
0	1	0	Áreas no Protegidas	SI	NO	SI	NO
0	1	0	Área Protegida	SI	NO	NO	NO
0	1	1	Área no Protegida	SI	SI	SI	NO

0	1	1	Área Protegida	SI	NO	NO	NO
1	0	0	Áreas no Protegidas	SI	NO	SI	NO
1	0	0	Área Protegida	SI	NO	NO	NO
1	0	1	Áreas no Protegidas	SI	SI	SI	NO
1	0	1	Áreas Protegidas	SI	NO	NO	NO
1	1	0	Toda la Memoria de Programa	SI	NO	SI	SI

REGISTROS ASOCIADOS CON LA MEMORIA DE DATOS EPROM/PROGRAMA FLASH

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh, 8Bh, 10Bh,18Bh	**INTCON**	**GIE**	**PEIE**	**T0IE**	**INTE**	**RBIE**	**T0IF**	**INTF**	**RBIF**	0000 000x	0000 000u
10D h	**EEADR**	Registro de direccionamiento de EEPROM								xxxx xxxx	uuuu uuuu
10F h	**EEADRH**	--	--	--	Parte alta de la direccionamiento de EEPROM					xxxx xxxx	uuuu uuuu
10F h	**EEADRH**	--	--	--	Parte alta de la direccionamiento de EEPROM					xxxx xxxx	uuuu uuuu
10F h	**EEADRH**	--	--	--	Parte alta de la direccionamiento de EEPROM					xxxx xxxx	uuuu uuuu
10F h	**EEADRH**	--	--	--	Parte alta de la direccionamiento de EEPROM					xxxx xxxx	uuuu uuuu
10C h	**EEDATA**	Registro de dato de la EEPROM								xxxx xxxx	uuuu uuuu
10E h	**EEDATH**	--	--							xxxx xxxx	uuuu uuuu
18C h	**EECON1**	EEPGD	---	---	---	WRERR	WREN	WR	RD	x--- xxxx	x--- u000
18D h	**EECON2**	Registro2 de control EEPROM (registro físico no implementado)									
8D h	**PIE2**	--	(1)	--	EEIE	BCLIE	---	---	CCP2IE	-r-0 0—0	-r-0—0
0D h	**PIR2**	--	(1)	--	EEIF	BCLIF	---	---	CCP21F	-r-0 0—0	-r-0 0--0

Leyenda: x = Indeterminado, u = Permanece Invariable, r = reservado; - = No implementado se lee como 0; No se usan las células sombreadas durante el acceso a FLASH/EEPROM.

Nota 1.- Área de memoria reservada; estos bits se mantienen siempre a cero

REGISTROS DE FUNCIONES ESPECIALES

Los Registros de Funciones Especiales (SFR) son registros usados por la CPU y los módulos periféricos para controlar el funcionamiento deseado del dispositivo. Estos registros están realizados como RAM estática. Un listado de estos registros es el que se muestra en la Tabla 5.

Los Registros de Función Especiales podemos clasificarlos en dos tipos, los correspondientes al la CPU y los que controlan los periféricos. Los registros que afectan directamente a la CPU se describen seguidamente con detalle. Aquellos registros relacionados con el funcionamiento de los periféricos se describen en el apartado correspondiente al periférico.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 0											
00h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h(4)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h(5)	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09h(5)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
0Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(3)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	(6)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	-r-0 0--0
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Registro Especiales de los PIC 16F87X

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 1											
80h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(4)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h(5)	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h(5)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
8Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	PSPIE(3)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	(6)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	BOR	---- --qq	---- --uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Tabla 5.2.- Banco 1 Registros especiales del PIC 16F87X (2)

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
Bank 2											
100h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h(4)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high byte						xxxx xxxx	uuuu uuuu
10Fh	EEADRH	—	—	—	EEPROM address register high byte					xxxx xxxx	uuuu uuuu
Bank 3											
180h(4)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h(4)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h(4)	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h(4)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah(1,4)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh(4)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

6: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

Tabla 5.3.- Banco 2 Registros especiales del PIC 16F87X (2)

REGISTRO DE STATUS (Direcciones 03h, 83h,103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	**RP1**	**RP0**	**#T0**	**#PD**	**Z**	**DC**	**C**
Bit 7							Bit 0

bit 7: **IRP**: Selección de bancos para el direccionamiento indirecto

1 = Bancos 2 y 3 (1 00h- 1 FFh)

0= Bancos 0 y 1 (00h-FFh)

bit 6-5: **RP1:RP0**: Selección del banco para el direccionamiento directo

11 = Banco 3 (1 80h- 1 FFh)

10 =Banco 2(100h-17Fh)

01 =Banco 1 (80h-FFh)

00 = Banco 0 (00h-7Fh)

Cada Banco es de 128 bytes

bit 4:**#TO**: Flag de Timer Out

1= Después de conectar VDD o ejecutar CLRWDT o SLEEP

0= Al desbordarse el temporizador de WDT

bit 3:**#PD**:Flag de Power Down

1= Después de conectar VDD o al ejecutar la instrucción CLRWDT

0 = Al ejecutar la instrucción SLEEP

bit 2: **Z**: Flag de Cero

1= El resultado de la última operación aritmética o lógica es Cero

0= El resultado de la última operación aritmética o lógica es distinto cero

bit 1: **DC**: Flag de acarreo en el 4º bit de menos peso

1 = Acarreo en la suma

0 = No acarreo en la suma

En la resta es al contrario

bit 0: **C**: Flag de acarreo en el octavo bit

1 = Se ha producido un acarreo en la suma y no en la resta

0= Se ha producido un acarreo en la resta y no en la suma

Este bit también se utiliza en las instrucciones de rotación

REGISTRO OPTION u OPTION_REG (Dirección 81h,181h)

El Registro de **OPTION_REG** es un registro que puede ser leído o escrito y que contiene varios bits de control para configurar la asignación del preescaler al **TMR0** o al WDT, la interrupción externa, el **TMR0** y las resistencias de pull-up del **PORTB**.

Nota.- Asignar el preescaler como 1: 1 al registro **TMR0**, asigna el preescaler al **Watchdog**.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	**INTED**	**T0CS**	**T0SE**	**PSA**	**PS2**	**PS1**	**PS0**
Bit 7							Bit 0

Bit 7: **#RBPU**: Resistencia de Pull-up en, el PORTB

1=Resistencia de Pull-up desactivada

0= Resistencia de Pull-up activada

bit6 1:**INTEDG**:, Flanco, de control de interrupciones

1:Interrupción por flanco ascendente en el pin RB0/INT,

0:Interrupción por flanco: descendente en el pin RB0/INT

bit 5: **T0SC**. Selección del tipo de Reloj para TMRO

1 = Los pulsos se introducen a través del, pin RA4/TOCK1

0 = Los Pulsos de reloj internos- Fosc/4

bit 4: **T0SE**: Tipo de flanco para **TMR0**

1 = Incremento de **TMR0** en cada flanco, descendente por el pin RA4/TOCKI

0 = Incremento de **TMR0** en cada flanco ascendente :por el pin RA4/TOCKI

bit 3: **PSA**:, Asignación del Preescaler

1 = El preescaler se le asigna al **WDT**

0 = El preescaler se le asigna al **TMR0**

bit 2-0:**PS2:PS0**: Rango de actuación del preescaler

PS2	PS1	PS0	Divisor de TMR0	Divisor de WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

REGISTRO INTCON (Dirección 0Bh, 8Bh, l0Bh, 18Bh)

El registro INTCON es un registro de lectura y escritura que contiene los bit de habilitación de interrupciones por desbordamiento de **TMR0** por cambio de nivel en el PORTB e interrupciones externas por la línea **RBO/INT**.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	**PEIE**	**T0IE**	**INTE**	**RBIE**	**T0IF**	**INTF**	**RBIF**
Bit 7							Bit 0

bit 7: **GIE**: bit de habilitación global de Interrupciones

1= Habilita el permiso de interrupciones

0= Inhabilita todas las interrupciones

bit 6: **PEIE**: bit de habilitación de interrupciones de los periféricos que no se controla con el registro INTCON

1= Habilita el permiso de interrupciones de los periféricos

0= Inhabilita las interrupciones de los periféricos

bit 5: **T0IE**: bit de habilitación de la interrupción por desbordamiento del TMR0

1= Habilita la interrupción

0= Inhabilita la interrupción

bit 4: **INTE**: bit de habilitación de la interrupción externa por el pin RB0/INT

1= Habilita la interrupción

0= Inhabilita la interrupción

bit 3: **RBIE**: bit de habilitación de interrupción por cambio de nivel en el PORTB

1= Habilita la interrupción

0= Inhabilita la interrupción

bit 2: **T0IF**: flag de indicación de desbordamiento de TMR0

1= El TMR0 se ha desbordado. Se borra por software

0= El TMR0 no se ha desbordado

bit 1: **INTF**: flag de estado de la interrupción externa INT

1= La interrupción externa se ha producido. Se borra por software

0= La interrupción externa no se ha producido

bit 0: **RBIF**: flag de indicación de interrupción por cambio de nivel en PORTB

1= Se ha producido un cambio de nivel en los pines RB7:RB4. Se borra por software.

0= No se ha producido un cambio de nivel en los pines RB7:RB4

REGISTRO PIE1 (Dirección 8Ch)

Este registro contiene los bit individuales de habilitación de interrupciones de los periféricos

Nota.- El bit **PEIE** (INTCON <6>) debe esta a uno para habilitar las interrupciones de cualquiera de los periféricos.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	**ADIE**	**RCIE**	**TXIE**	**SSPIE**	**CCP1IE**	**TMR2IE**	**TMR1IE**

Bit 7 Bit 0

bit 7: **PSPIE**: bit de habilitación de interrupción por lectura / escritura en el Puerto Paralelo Esclavo. Para los modelos de 40 pines.

1 = Habilita la interrupción por lectura/escritura en el **PSP**

0= inhabilita la interrupción por lectura/escritura en el **PSP**

bit 6: **ADIF**: bit de habilitación de interrupción por finalización de la conversión A/D

l = Habilita la interrupción del convertidor A/D

0 = Inhabilita la interrupción del convertidor A/D

bit 5: **RCIE**: bit de habilitación de interrupción en recepción por el USAR, cuando se llena el buffer,

1 = Habilita interrupción por recepción en el USAR

0 = Inhabilita interrupción por recepción en el USAR

bit 4: **TXIE**: bit de interrupción al transmitir por el USAR, cuando se vacía el buffer.

1 = Habilita la interrupción de transmisión por el USAR

0 = Inhabilita la interrupción de transmisión por el USAR

bit 3: **SSPIE**: bit de habilitación de interrupción por el Puerto Serie Síncrono (SSP)

1= Habilita la interrupción del **SSP**

0= Inhabilita la- interrupción del **SSP**

bit 2: **CCP1IE**: bit de habilitación de interrupción del módulo CCPI cuando se produce una captura o una comparación.

1= Habilita la interrupción del **CCPI**

0=. Inhabilita la interrupción del **CCPI**

bit 1: **TMR2IE**: bit de habilitación de interrupción por desbordamiento de **TMR2** que

está emparejado con el registro PR2 (92h)

1= Habilita la interrupción por desbordamiento de **TMR2** emparejado a **PR2**

0= Inhabilita la interrupción de **TMR2** emparejado a PR2

bit 0: **TMR1IE**: bit de habilitación de interrupción por desbordamiento del TMRI

1= Habilita la interrupción por desbordamiento de TMR1

0= Inhabilita la interrupción por desbordamiento de TMR1

Nota 1.- **PSPIE** para los dispositivos de 28 pines, siempre mantiene este bit a cero

REGISTRO PIRI (Dirección 0Ch)

El registro de **PIRI** contiene los flags individuales que indican las interrupciones provocadas por los periféricos.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
PSPIF	**ADIF**	**RCIF**	**TXIF**	**SSPIF**	**CCP1IF**	**TMR21F**	**TMR1IF**
Bit 7							Bit 0

bit 7: **PSPIF**: Flag de Lectura/Escritura por el Puerto Paralelo esclavo

1= Concedido el permiso de interrupción para la puerta paralela esclava al realizar una operación de Lectrura/Escritura. En los modelos de 40 pines

0= No está permitida la interrupción

bit 6: **ADIF**: Flag de final de conversión del convertidor A/D

1= El convertidor A/D ha finalizado la conversión

0= El convertidor A/D no ha finalizado la conversión

bit 5 **RCIF**: Flag de recepción por el USAR

1= El buffer de datos recibidos por el USAR está lleno

0= El buffer de datos recibidos por el USAR no está lleno

bit 4: **TXIF**: Flag de transmisión el USAR

1= El buffer de datos a transmitir no está lleno

0= El buffer de daos a transmitir está lleno

bit 3: **SSPIF**: Flag de interrupción del Puerto Serie Síncrono (SSP)

1= La condición del SSP ha ocurrido, debe ponerse a cero por software antes de volver del programa de atención a la interrupción. Las

condiciones que soporta este bit son

SPI

Ha tenido lugar una Transmisión / Recepción

I2C Slave

Ha tenido lugar una Transmisión / Recepción

I2C Master

Ha tenido lugar una Transmisión / Recepción

La condición de salida iniciada se completó por el módulo de SSP.

La condición de parada inicio se completó por el módulo de SSP.

La condición de reinicialización se completo por el módulo SSP

Una condición de la salida se ha realizado mientras el módulo de SSP estaba en estado de espera (sistema de Multimaster).

0 = no ha ocurrido ninguna condición de interrupción del módulo SSP

bit 2: **CCP1IF**: Flag de interrupción de CCPI

Modo Captura

1= Ocurrió una captura de TMR1 (debe ponerse a cero por software)

0= No ocurrió ninguna captura de TMR1

Modo Comparación

1= Se ha realizado una comparación de TMR1 Con el registro emparejado /debe ponerse a cero por Software)

0= No se ha realizado comparación

Modo PWM

Este modo no se utiliza

bit 1: **TMR2IF**: Flag de interrupción de TMR2 emparejado con PR2

1= **TMR2** emparejado con **PR2** ocurrió (debe ponerse a cero por software)

0= No ha ocurrido el emparejamiento de TMR2 con PR2

bit 0: **TMR1IF**: Flag de desbordamiento de TMRI

1 =el registro se desbordo (debe ponerse a cero por software)

0 = el registre¡ de TMRI no se desbordo

Nota 1.- **PSPIF** está reservado para los dispositivos de 28 pines; se mantiene siempre a cero.

REGISTRO PIE2 (Dirección 8Dh)

El registro de PIE2 contiene los bit individuales que habilita las interrupciones del periférico CCP2,. la interrupción por colisión del SSP y la ininterrupción de escritura en la EEPROM.

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
---	0	---	EEIE	**BCLIE**	--	--	**CCP2IE**
Bit 7							Bit 0

bit 7: **No implementado**: se lee como “0’

bit 6 **Reservado**, Mantiene este bits a cero

bit 5: **No implementado**: se lee como “0”

bit 4: **EEIE**: Habilita la interrupción por escritura en la EEPROM de datos

1= Habilita la interrupción por escritura de la EEPROM de datos

0 =Deshabilita la interrupción por escritura en la EEPROM de datos

bit 3: **BCLIE**: Habilita la interrupción por colisión en el bus SSP cuando dos o más maestros tratan de transferir al mismo tiempo.

1 = Habilita la interrupción por colisión de bus SSP

0 = Deshabilita la interrupción por colisión en el bus SSP.

bit 2-1 **No implementados**, se leen como “0”

bit 0: **CCP2IE**: Habilita la interrupción del modulo CCP2

1 = habilita la interrupción de CCP2

0 = inhabilita la interrupción de CCP2

REGISTRO PIR2 (Dirección 0Dh)

El registro de PIR2 contiene los flags que indican la interrupción del **CCP2**, de la

colisión del bus **SSP** y la interrupción de la escritura en la **EEPROM**

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-x
---	**(1)**	**---**	**EEIF**	**BCLIF**	**---**	**---**	**CCP2IF**
Bit 7							Bit 0

bit 7: **No implementado**: se lee como “0”

bit 6: **Reservado**: Mantiene este bit siempre a cero

bit 5: **No implementado**: se lee como “0”

bit 4: **EEIF**: Flag que indica si se ha producido escritura en al EEPROM

1 = Se ha completado la escritura en la EEPROM (Se pone a cero por software)

0 = No se ha finalizado la escritura o no se ha comenzado

bit 3: **BCLIF**: Flag que indica la colisión en el bus

1 = Se ha producido una colisión en el bus SSP, cuando se configura en el modo I2C Master.

0 = No se ha producido colisión en el bus

bit 2-1:**No implementados**: se leen como “0”

bit 0: **CCP21F**: Flag de ininterrupción de CCP2

<u>Modo Captura</u>

1 = Ha ocurrido una captura del registro TMR1 (debe ponerse a cero por software)

0= No se ha producido captura

<u>Modo Comparación</u>

1 = Se ha producido una comparación de TMR1 con su pareja (debe ponerse a cero por software).

0 = No se ha producido comparación

Modo PWM

No se utiliza

REGISTRO PCON (Dirección 8Eh)

El registro PCON (*Power Control*) Control de Alimentación, contiene los *flags* que permiten diferenciar entre un *Power-on Reset* (POP), un *Brown-out Reset* (BOR), un Reset por *Watchdog* (WDT) y un Reset externo por MCLR.

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
---	0	---	---	---	---	**#POR**	**#BOR**
Bit 7							Bit 0

bit 7-2:**No implementado**: se lee como “0”

bit 1: **#POR**: bit de estado Power-on Reset

1 = No se ha producido un Power-on Reset

0= Se ha producido un Power-on Reset (debe reestablecerse por software)

bit 0: **#BOR**: bit de estado de Borwn-out Reset

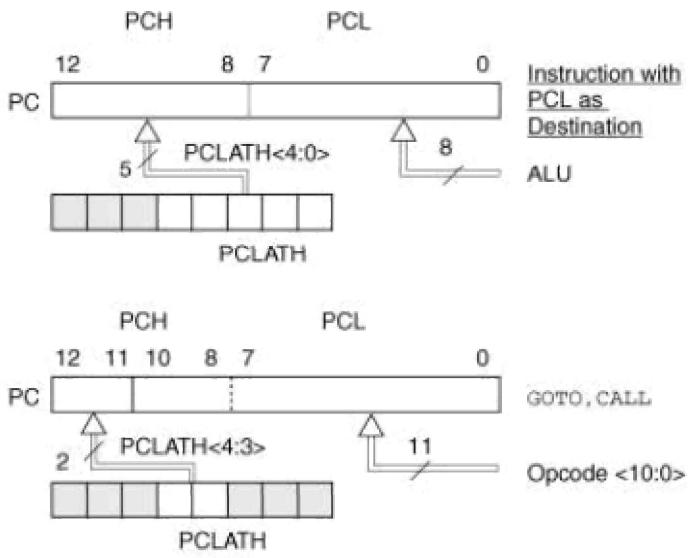
1 = No se ha producido ningún Brown-out Reset

0= Se ha producido un Brown-out Reset (debe reestablecerse por software)

PCL y PCLATH

El contador del programa (PC) está formado por 13 bit que sirven para direccionar la memoria de código, estos bits se encuentran en dos registros específicos. El byte bajo viene del registro de PCL que puede ser leído y escrito. Los bits superiores (PC<12:8>). Están alojados en el registro PCH , sobre el que no se puede leer ni escribir, pero se puede acceder a él indirectamente a través del registro PCLATH.

Las instrucciones de salto CALL y GOTO sólo proporcionan 11 bits de la dirección a saltar. Esto limita el salto dentro de cada banco de 2K. Cuando se desea salir del banco actual hay que programar correctamente los bits PCLATH <4:3> que seleccionan el banco. Es labor del programador modificar el valor de dichos bits en las instrucciones CALL y GOTO.



Estructura del Contador de Programa

EL STACK

La familia de los PIC 16F87X dispone de una pila o stack de 8 niveles de profundidad para un tamaño de PC de 13 bits. Esta pila es transparente al programador, es decir, funciona automáticamente y no dispone de instrucciones para guardar o sacar de ella información.

Con la Instrucción CALL y con las interrupciones el valor de PC se salva en el nivel superior. Con las instrucciones RETURN, RETLW y RETFIE el valor contenido en el valor superior de la pila se carga en el PC.

La pila funciona como un buffer circular. Esto significa que después de que se han guardado 8 valores en ella, el noveno borra el valor que se guardó en primer lugar. El décimo borra el que se guardo en segundo lugar, etc.

Es por tanto responsabilidad del programador cuidar el no provocar un desbordamiento de pila, no hay mecanismo que advierta de esta situación y si sucede se provocará un error en tiempo de ejecución muy difícil de encontrar.

Paginación de la Memoria de Programa

Los dispositivos de la familia PIC 16F87X son capaces de direccionar un bloque de hasta 8K de memoria continua. Las instrucciones CALL y GOTO proporcionan solo 11 bits de la dirección de memoria, lo que permite un salto de bifurcación dentro de una página de 2K de la memoria de programa.

Al hacer un CALL o una instrucción GOTO, dos 2 bit superiores de la dirección son proporcionados por el PCLATH <4:3>. Al hacer un CALL o un GOTO, el usuario debe asegurarse de que la página se han seleccionados con los bit correspondientes a la página que se desea llamar. Hay que tener precaución

en los retornos de subrutinas y retornos de interrupción para no salirse con el valor almacenado en la PILA.

Por consiguiente en la manipulación del PCLATH <4:3> no se requieren estos bits para las instrucciones del retorno. El ejemplo 2-1 muestras la llamada a una subrutina en página 1 de la memoria del programa. Este ejemplo asume que el PCLATH se salva y se restaura por el programa de atención a la interrupción (si se usan las interrupciones).

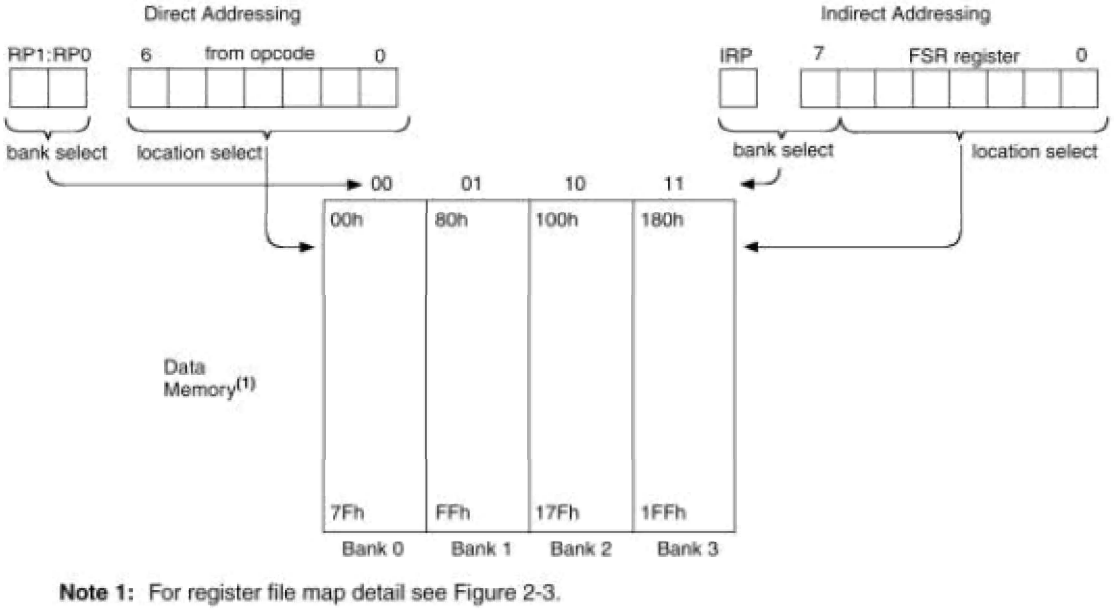
Ejemplo 2.1

```
        ORG     0x500

        bcf     PCLATH,4

        bsf     PCLATH,3        ;Selecciona la pagina 1 (800h-FFFh)

        call    SUB 1-P 1       ;Llamada a subrutina en la Página 1 (800h-FFFh)



        ORG    0x900            ;página 1 (800h-FFFh)

SUB1_Pl

                                ;Llamada a subrutina

                                ;página 1 (800h-FFFh)

        return                  ;Retorno de subrutina en pagina 0 (000h-7FFh)
```

Direccionamiento indirecto, los registros INDF y FSR

El registro INDF no es un registro físico, que ocupa la dirección 0 en todos los bancos, este registro se utiliza cuando queremos realizar un direccionamiento indirecto. Cuando se hace referencia al registro, se accede realmente a la dirección de un banco de registro especificada con los 7 bits de menor peso del registro FSR. El bit de más peso del FSR junto al bit IRP de registro de STATUS se encarga de seleccionar el banco a acceder, mientras que los 7 bit de menos peso apuntan a la dirección

deseada.



-Forma de seleccionar el banco y la dirección de la RAM en los direccionamientos directo e indirecto.

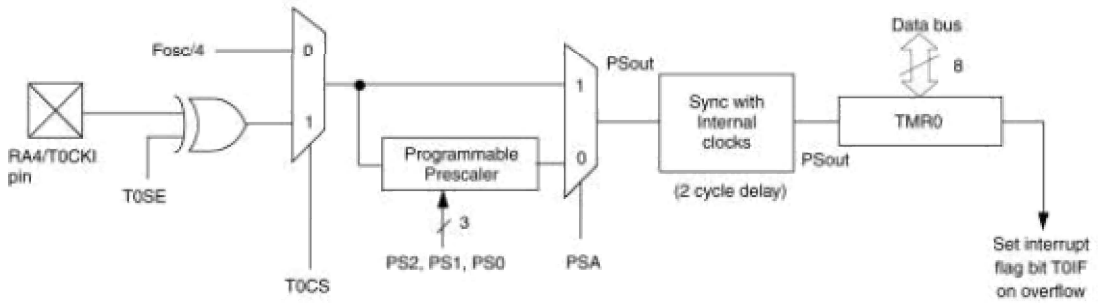
MODULO TIMER0

El módulo Titner0 es temporizador/contador con las siguientes características:

El temporizador/contador dispone de 8 bits, puede escribirse y leerse
Preescaler programable por Software de 8 bits
Puede trabajar con el reloj interno o con una señal de reloj externa
Dispone de una interrupción por desbordamiento al pasar de FFh a 00h
Selección de flanco ascendente o descendente para el flaco del reloj externo

En la Figura 13 se muestra un diagrama de bloques del Timer0 y el preescaler que comparte con el WDT. Puede encontrarse información adicional sobre el módulo Timer0 en el manual de referencia de los microcontroladores de la escala media (DS33023).

Figura 13



Notas.

1 - El preescaler es compartido con el WDT (En la figura 14 se puede ver un diagrama de bloques detallado

2 -TOCS, TOSE, PSA, PS2:PS0 (OPTION_REG<5:0>)

El modo temporizador se selecciona poniendo a cero el bit **T0CS** (registro **OPTION_REG** <5>). En el modo temporizador, el módulo Timer0 se incremento en cada cielo de instrucción (sin el preescaler). Si el registro TMR0 se escribe, el incremento se inhibe durante los siguientes dos ciclos de instrucción. EL usuario puede trabajar teniendo en cuenta esto y ajustando el valor a cargar en el **TMR0**.

El modo contador se selecciona poniendo a uno el bit **T0CS** (registro **OPTION_REG** <5>). El modo contador, Timer0 se incremento en cada flaco de subida o de bajada de la señal que le llega por RA4/TOCK1. El flanco de incremento se determina por el bit **T0SE** (registro **OPTION_REG** <4>). Poniéndose a cero **T0SE** se selecciona el flanco ascendente. El preescaler se comparte exclusivamente entre el Timer0 y el WDT. El preescaler no es de lectura/escritura.

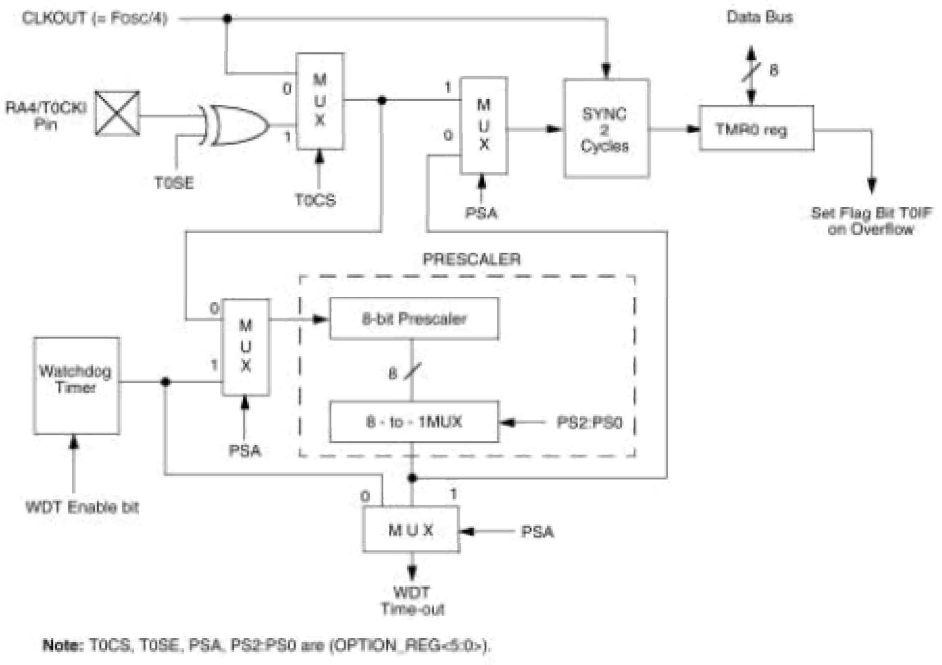


Diagrama de bloques del TMR0

Interrupción del Timer0.

La interrupción de TMR0 se produce cuando el registro **TMR0** se desborda al pasar de FFh a 00h. Este desbordamiento pone a uno el bit **T0IF**

(**INTCON**<2>). La ininterrupción puede enmascararse poniendo a cero el bit **T0IE** (**INTCON** <5>). EL bit **T0IF** debe ponerse a cero por software al finalizar la rutina de atención a la interrupción del desbordamiento de **TMRO**. La ininterrupción de TMRO no saca al microcontrolador del estado de SLEEP, debido a que el temporizador está desactivado durante el modo SLEEP

Utilización del Timer0 con un Reloj Externo

Cuando no se utiliza el preescaler, la entrada de reloj externa es igual a la salida del preescaler. Las sincronización de TOCKI con los relojes de fase interior se acopla, a la salida del preescaler en los ciclos Q2 y Q4 de los relojes de fase internos. Por consiguiente, es necesario que TOCKI está a nivel alto por al menos durante 2Tosc (y un pequeño retardo de 20ns) y a nivel bajo por lo menos 2Tosc (y un retardo RC de 20ns).Ver las características eléctricas del dispositivo deseado.

Preescaler

Hay sólo un preescaler disponible que es está compartido y puede asignarse indistintamente al moduló de Timer0 y el al WDT. La asignación del preescaler al Timer0 hace que no haya ningún preescaler para el WDT, y viceversa. Este preescaler no se puede leer ni escribir.

El bit **PSA** y **PS2**:**PS0** (**OPTION_REG** <3:0>) determinan la asignación del preescaler y el rango del preescaler. Cuando se le asigna al módulo del Tirner0 , todas las instrucciones, que escriben en el registro **TMR0** (por ejemplo CLRF TMR0, MOVWF TMR0, BSF TMR0,x... etc.) ponen a cero el preescaler. Cuando se le asigna al WDT, una instrucción CLRWDT limpia el preescaler junto con el temporizador del WDT. EL preescaler no se puede leer ni escribir.

Nota.- Escribir en TMR0, cuando el preescaler es asignado a Timer0, limpia la cuenta del preescaler, pero no cambia el contenido del preescaler.

TABLA DE REGISTROS ASOCIADOS AL TIMER0

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0bh, 8Bh 10Bh,18Bh	**INTCON**	GIE	PEIE	T0IE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
81h,181h	**OPTION_REG**	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	**TRISA**	---	---	Registro de direccionamiento de datos del PORTA						--11 1111	--11 1111

Leyenda: x = desconocido, u = inalterado; - = no implementado se lee como ‘0’.
Las celdas sombreadas no son usadas por el TIMER0

EL MÓDULO DE TIMER1

El módulo de TIMER1 es un temporizador/contador de 16 bits, formado por dos registros de 8 bits cada uno **TMR1H** y **TMR1L**, que son de lectura y escritura y que son los que guardan el valor de la cuenta en cada momento. El valor de los registros **TMR1H** y **TMR1L** se incrementan desde 0000H hasta FFFFH, en cuyo instante se vuelve a poner a 0000h y se activa el flag **TMRIF** (**PIR1**<0>) La ininterrupción de **TMR1** puede ser habilitada/deshabilitada según este a uno/cero el bit **TMR1IE** (**PIE1** <0>).

EL TIMERl tiene los siguientes modos de trabajo:

1. Como Temporizador
2. Como contador Síncrono
3. Como contador Asíncrono

El modo de trabajo viene determinado la fuente de los impulsos de reloj, es decir, la señal de reloj puede ser externa o interna, se selecciona con el bit **TMRLCS** del registro **TlCON**<l>, cuando este bit está a nivel bajo se selecciona el reloj el interno del micro-controlador (Fosc/4) y cuando está a uno se selecciona el modo contador y cuenta los impulsos que le llegan a través del pin RC0/TlCKl. Además como se ver mas adelante el TIMER1 tiene la posibilidad de reinicializarse, a partir del módulo CCP.

REGISTRO T1CON: Registro de control del TIMER1 (dirección 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
---	---	**T1CKPS1**	**T1CKPS0**	**T1OSCEN**	**T1SYNC**	**TMR1CS**	**TMR1ON**
Bit 7							Bit 0

bit 7-6:**No implementados**: Se lee como “0”

bit 5-4:**TlCKPS1:T1CKPS0**: bit de selección del preescaler de la señal de reloj del TIME1

11 = valor del preescaler 1:8

10 = valor del preescaler 1:4

01 = valor del preescaler 1:2

00 = valor del preescaler 1: 1

bit 3 : **T1OSCEN**: bit de habilitación del oscilador del TIMER1. Cuando se emplea un oscilador externo, hay que poner este bit a 1. El TMR1 puede trabajar a una frecuencia totalmente independiente de la del sistema.

1 = Habilita el oscilador

0 = Deshabilita el oscilador

Nota.- El oscilador y la resistencia de desconectan para reducir el consumo

bit 2: #TlSYNC: bit de control de sincronización de la señal de entrada.

Con TMR1CS = 1

1= No sincroniza la entrada de reloj externa

0 = Sincroniza la entrada de reloj externa

Con TMR1CS = 0

En esta condición se ignora. El TIMER1 utiliza el reloj interno cuando TMRICS=0

bit 1 **TMR1CS**: bit de selección de la fuente de reloj del TIMER1

1 = Reloj externo por el pin RC0/T1OSO/T1CK1 (flanco ascendente)

0 = Reloj interno (FOSC/4)

bit 0: **TMR1ON**: TIMER1. activo. Hace entrar o no en funcionamiento el TIMER1.

1 = Habilita el TIMER1

0 = Deshabilita el TIMER1

Modo de funcionamiento del Timer1 como Temporizador

Este modo se selecciona poniendo a cero el bit **TMR1CS** (**T1CON** <1>. En este modo la señal de reloj es el reloj interno del microcontrolador FOSC/4. En este modo de trabajo el bit **T1SYNC** (**T1CON** <2>) no tiene ningún efecto ya que el reloj interno está siempre sincronizado.

Modo de funcionamiento del Time1 como Contador

En este modo puede trabajar como contador síncrono o asíncrono . Cuando el TIMER1 se está incrementando según le llegan los impulsos externos, los incrementos ocurren en los flancos de subida. Después de que el TIMER1 se ha configurado como contador, debe producirse un flanco de bajada antes de empezar a contar.

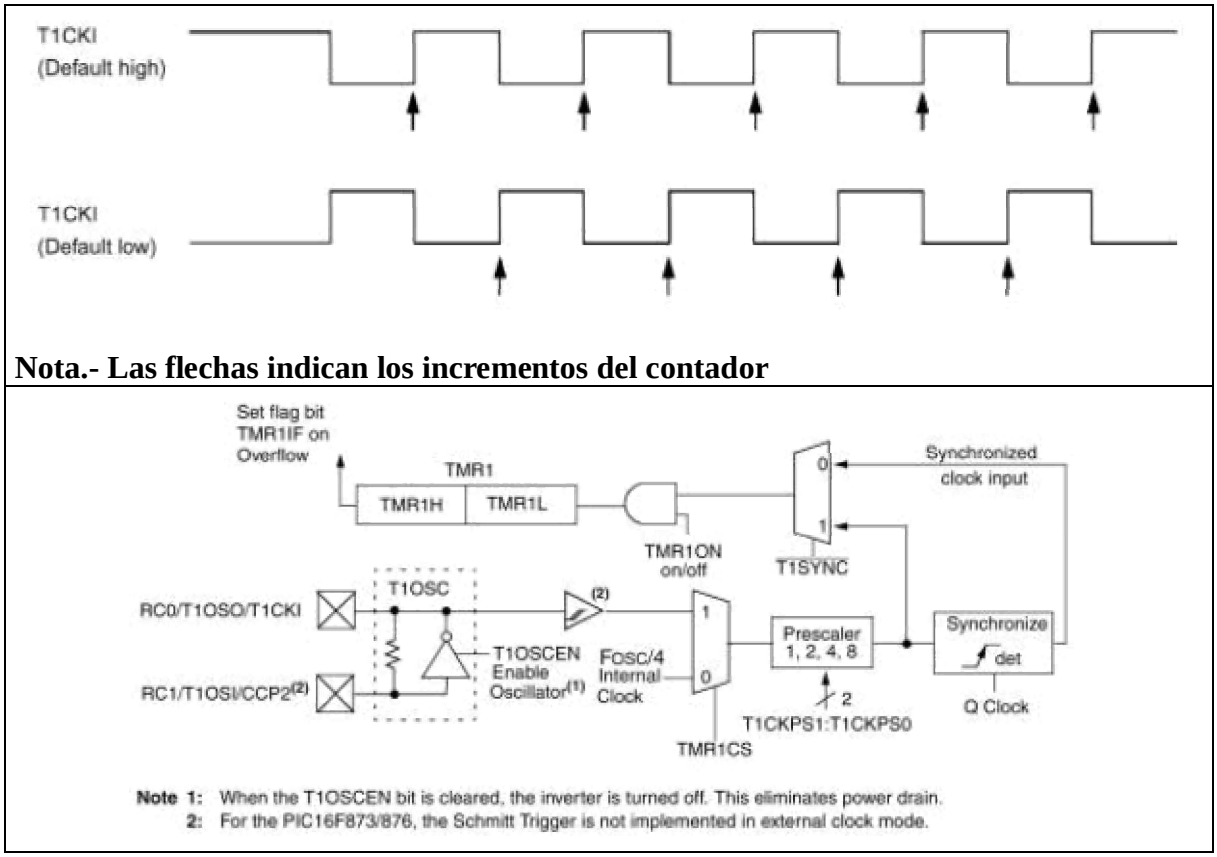


Diagrama de bloques de TMR1

Modo de funcionamiento del Timer1 como Contador Síncrono

Para seleccionar este modo se pone a uno el bit **TMR1CS** (**T1CON** <1>). En este modo el contador se incrementa en cada flanco ascendente de la señal de reloj que se introduce por el pin RC0/T1OSO/TICK1 cuando el bit **T1OSCEN** está a uno, y por el pin RC1/TlOSI/CCP2, cuando el bit **T1OSCEN** está a cero.

Si **T1SYNC** se pone a cero, entonces la entrada de reloj externa se sincroniza con los relojes de fase interiores. La sincronización se hace después de la fase del preescaler. En el preescaler la fase de la señal de reloj es por lo tanto asíncrona.

En este modo de trabajo, durante el modo SLEEP el TIMER1 no se incrementa aún cuando la señal de reloj externa esté' presente. El preescaler sin embargo continúa incrementándose.

Modo de funcionamiento del TIMER1 como Contador Asíncrono

Cuando el bit de control T1SYNC (T1CON <2>) se poner a uno, la señal de reloj externa no se sincroniza. El contador sigue realizando la cuenta de forma asíncrona respecto a la fase de la señal de reloj interna. El contador continúa la cuenta incluso en el modo SLEEP y puede generar una interrupción por desbordamiento que despierta al procesador. Hay que tener especial cuidado con el software al leer o escribir el contador (Sección 6.5.1).

Cuando se trabaja en el modo contador asíncrono, el TIMER1 no puede usare como base de tiempos para el módulo CCP (Captura y comparación-PWM)

Lectura y escritura en el TIMER1 cuando se trabaja en el modo contador asíncrono

Se pueden leer los contadores TMR1H y TMR1L mientras la señal externa del contador se está recibiendo (teniendo cuidado con el hardware). Sin embargo, el usuario debe tener en cuenta que el contador es de 16 bits y se pueden tener ciertos problemas al leer los dos registros de ocho bits, ya que el contador puede desbordarse entre las lecturas. Para escribir en él, se recomienda que el usuario simplemente pare el contador y escriba los valores deseados. Cuando se escribe el registro del contador puede haber conflicto mientras este se está incrementando. Esto puede producir un valor imprevisible en el contador. Lee el registro de 16 bit requiere algún cuidado. Los ejemplos 12-2 y 12-3 de la familia de microcontroladores de la escala media, manual de referencias (DS33023), muestra como leer y escribir el TIMER1, cuando está corriendo en el modo asíncrono.

Oscilador del TIMER1

Se puede conectar un oscilador a cristal entre los pines T1OSI (entrada) y T1OSO (salida del amplificador). Se debe habilitar poniendo a uno el bit de control TLOSCEN (TICON <3>). El oscilador de bajo consumo puede trabajar hasta 200 kHz. En estas condiciones el oscilador sigue funcionando aunque se fuerce el modo SLEEP. Está pensado para trabajar con un cristal de 32 kHz. La Tabla 6 muestra el valor de los condensadores para el TIMER1. El oscilador TIMER1 es idéntico al oscilador LP. El usuario debe proporcionar un tiempo de retardo por software para asegurar la salida apropiada del oscilador.

Tipo de Oscilador	Frecuencia	C1	C2
LP	32 kHz	33pF	33pF
	100kHz	15 pF	15 pF
	200	15 pF	15 pF
Estos valores son para los siguientes cristales de referencia			
32.768	EpsonC-001R21.768K-A		±20 PPM
100 kHz	Epson C-2 100.000 KC-p1		±20 PPM
200 kHz	STD XTL 200.000 kHz		±20 PPM

Nota.-

1. Capacidades mayores aumentan la estabilidad del oscilador, pero también aumentan el start-up.
2. Cada cristal/resonador tiene sus propias características, el usuario debe consultar los valores apropiados de los componentes externos al cristal.

- Selección de los condensadores para el oscilador del TIMER1

RESTABLECIMIENTO DEL TIMER1 UTILIZADO LA SALIDA TRIGGER DEL CCP

Si el módulo **CCPI** o **CCP2** se configuran en modo comparación para generar un "disparo por evento especial" (**CCP1M3**:**CCP1MO** = 1011), esta señal reestablecerá el **TIMIER1**.

Nota.- El evento especial que activa los módulos CCP1 y CCP2 no activará el flag **TMR1IF** (**PIR1**<0>).

El TIMER1 debe configurarse como temporizador o contador síncrono

Restablecimiento TIMER1 que usa un disparador de CCP

TIMER1 debe configurarse para temporizador o contador síncrono para aprovechar esta característica. Si el **TIMER1** está trabajando en modo contador asíncrono, el reset no puede ser activado.

En el caso de escribir sobre el **TIMER1** en el momento del disparo del evento especial de **CCP1** o **CCP2**, se tomará el valor anterior.

En el caso de escribir sobre **TIMER1** si coincide con un disparo por evento especial de **CCP1** o **CCP2**, se carga el valor anterior.

En este modo de funcionamiento, el par de registros **CCPRxH**:**CCPRxL** se reponen para volver a tener el periodo del **TIMER1**.

Restablecimiento del par de registro del TIMER1 (TMR1R, TMR1L)

Los registros TMR1H y TMR1L no se inicializan a 00h después de un Power-on Reset o por cualquier otro reset excepto por un disparo de evento especial en **CCP1** y **CCP2**.

El registro **T1CON** se inicializa a 00h después de un Power-on Reset o Brown-out Reset que deja a **1:1** el preescaler. En los demás reset, el registro no es alterado.

Timer1 Preescaler

El contador del preescaler se pone a cero al escribir en los registros **TMR1H** o **TMR1L**.

RESUMEN DE REGISTROS ASOCIADOS AL TIMER1

En la siguiente tabla se muestran los registros principales que controlan el comportamiento del TIMER1 y la distribución de los bit.

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset

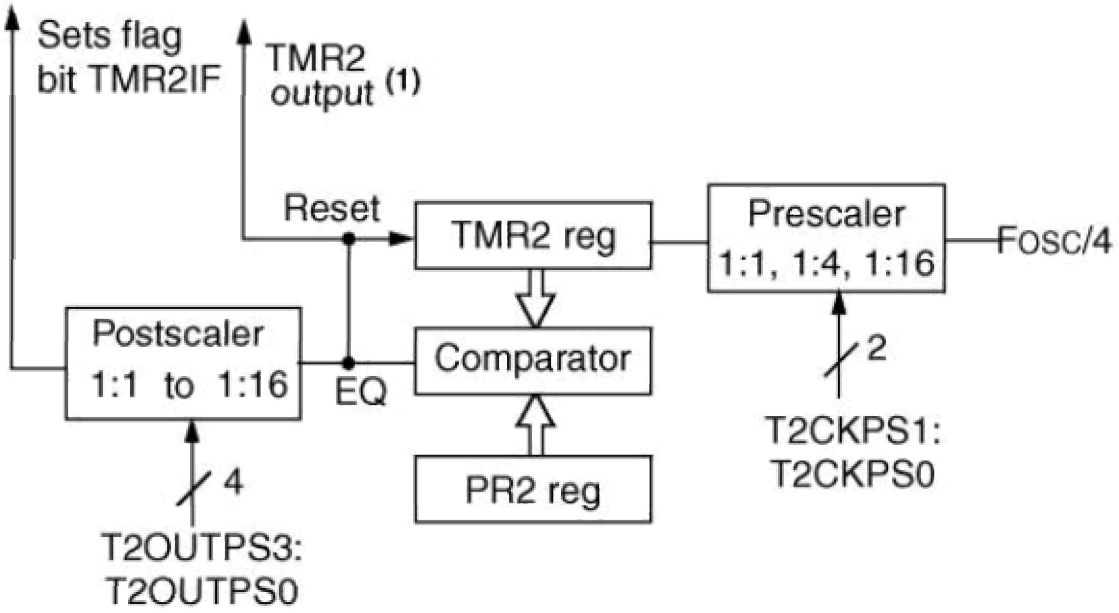
0Bh,8Bh 10Bh,18Bh	**INTCON**	GIE	PEIE	T0IE	INTE	RBIE	TOIF	INTE	RBIF	0000 000x	0000 000u
0Ch	**PIR1**	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Bh	**PIE1**	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	**TMR1L**	Registro de carga del byte de menor peso del registro de 16 bits de TMR1								xxxx xxxx	uuuu uuuu
0Fh	**TMR1H**									xxxx xxxx	uuuu uuuu
10h	**T1CON**	---	---	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--xx xxxx	--uu uuuu

Leyenda x = desconocido, u = inalterado; - = no implementado se lee como '0'. Las celdas sombreadas no son usadas por el TIMER1

Nota 1: Los bits PSPIE y PSPIF están reservados para el PIC16F873/876, mantener estos bit a cero.

EL MÓDULO DE TIMER2

El TIMER2 es un temporizador-contador ascendente de 8 bits que ocupa la posición 11H del banco de registros especiales, con un preescaler y postscaler. Se puede usar como base de tiempo para el módulo **PWM,** el módulo **CCP** (Módulo de Captura y Comparación) y la Puerta Serie Síncrona (SSP). El registro **TMR2** se puede leer y escribir y se pone a cero después de un Reset.



Nota 1.- Se puede seleccionar por software la salida del módulo TMR2 como señal de reloj para la transmisión en baudios del módulo SSP

Diagrama de bloques del módulo TIMER2

La señal de reloj interna (FOSC/4) antes de llegar al **TMR2** pasa por un preescaler (Predivisor) con rangos de 1:1, 1:4, 1:16 que se selecciona a través de los bits **T2CKPS1**:**T2CKPS0** (**T2CON** <1:0>).

Operaciones con el TIMER2

El Timer2 tiene emparejado el registro **PR2** que ocupa la posición de 92H del banco de registros especiales, de manera que al incrementarse **TMR2** y coincidir con el valor del registro **PR2** se produce un impulso de salida por **EQ**, estos impulsos pueden ser divididos por un postescaler antes de activar el flag **TMR2FI** (**PIR1<1>**).

El registro **PR2** es un registro de 8 bits que puede ser escrito y leído, este registro toma el valor FF después de un *Reset*.

El postscaler permite dividir la señal por cualquier valor comprendido entre 1:1 hasta 1:16, para controlar el postescaler se utilizan los bit **TOUTPS3**: **TOUTPS0** (T2CON <6:3>).

El Preescaler y el Postescaler se ponen a cero cuando:

- Se escribe sobre el registro **TMR2**
- Se escribe sobre el registro **T2CON**
- Se produce un reset (POR, MCLR restablecido, WDT reestablecido o BOR)

TMR2 no se pone a cero cuando se escribe en **T2CON**

REGISTRO T2CON: REGISTRO DE CONTROL DEL TIMER2 (dirección 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
---	**TOUTPS3**	**TOUTPS2**	**TOUTPS1**	**TOUTPS0**	**TMR2ON**	**T2CKPS1**	**T2CKPS0**
Bit 7							Bit 0

bit 7**:** **No implementado**: Se lee como 0

bit 6-3:**TOUTPS3:TOUTPS0**: bit de selección del rango del divisor del Postescaler para el **TIMER2**

0000 = Divisor del postescaler 1:1

0001 = Divisor del postescaler 1:2

0010 = Divisor del postescaler 1:3

0011 = Divisor del postescaler 1:4

0100 = Divisor del postescaler 1:5

0101 = Divisor del postescaler 1:6

0110 = Divisor del postescaler 1:7

0111 = Divisor del postescaler 1:8

1000 = Divisor del postescaler 1:9

1001 = Divisor del postescaler 1:10

1010 = Divisor del postescaler 1:11

1011 = Divisor del postescaler 1:12

1100 = Divisor del postescaler 1:13

1101 = Divisor del postescaler 1:14

1110 = Divisor del postescaler 1:15

1111 = Divisor del postescaler 1:16

bit 2: **TMR2ON**: bit de activación del TIMER2

1:= habilita el funcionamiento del TIMER2

0 = Inhibe el funcionamiento del TIMER2

bit 1-2:**T2CKPS1:T2CKPS0** Selección del rango de divisor del Preescaler del TIMER2

00 = Divisor del Preescaler 1:1

01 = Divisor del Preescaler 1:4

Ix = Divisor del Preescaler 1:16

Interrupciones del TIMER2

El temporizador **TMR2** tiene un flag de desbordamiento el **TMR2IF** (<1>**PIR1**).

El **TMR2** tiene asociado un Registro de Periodo PR2, que ocupa la posición 92h. Cuando el valor de cuenta de **TMR2** coincide con el valor cargado en **PR2**

se genera un impulso en la salida **EQ** (ver la Figura 16) y se pone a cero el **TMR2**. Estos impulsos pueden ser divididos por el postdivisor antes de activar el flag TMR21F(<1> PIR1).

El temporizador puede producir una interrupción si se pone a 1 el bit TMR2IE (<1> PIE1)

Salida del TMR2

La salida de **TMR2** (antes del postscaler) alimenta al Módulo de **SSP** que opcional mente puede usarse para generar la señal de reloj de desplazamiento.

REGISTROS ASOCIADOS AL TMR2

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh,8Bh 10Bh, 18Bh	**INTCON**	GIE	PEIE	TOIE	INTE	TOIF	RBIE	INTE	RBIF	0000 000x	0000 000u
0Ch	**PIR1**	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Bh	**PIE1**	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
11h	**TMR2**	Módulo Del registro Timer2								0000 0000	0000 0000
12h	**T2CON**	---	TOUTPS3	TOUPS2	TOUPS1	TOUPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 000	-000 0000
92h	**PR2**	Registro de período del TMR2								1111 1111	1111 1111

Leyenda: x = desconocido, u = inalterado; - = no implementado se lee como ‘0’. Las celdas sombreadas no son usadas por el TIMER2

Nota 1: Los bits PSPIE y PSPIF están reservados para el PIC16F873/876, mantener estos bit a cero.

MODULO CCP CAPTURA / COMPARACIÓN / PWM (Modulación de ancho de Pulsos)

Los microcontroladores de la familia 16F87X disponen de dos módulos de Captura / Comparación y PWM, cada modulo tiene un registro de 16 bits que pueden trabajar como:

- · Registro de captura de 16 bit
- · Registro de comparación de 16 bits
- · Modulación de anchura de pulsos PWM

Los módulos **CCP1** y **CCP2** son idénticos en su modo de funcionamiento, con la excepción del modo de disparo especial. En las tablas 17 y 18se muestran los recursos e interacciones de los módulos. En las secciones siguientes se describe el funcionamiento del módulo **CCP1**. **CCP2** opera de igual modo salvo en lo ya dicho.

Módulo CCP1:

El registro de Captura / Comparación / PWM (CCPR1) de 16 bits esta formado por dos registros de 8 bits: **CCPR1L** (byte, bajo) que ocupa la dirección 15h y el **CCPR1H** (byte alto) que ocupa la posición 16h. El registro de control del **CCP1** es el **CCP1CON** que ocupa la dirección 17h. El modo de disparo especial se genera por la igualdad en la comparación de **CCPR1** con **TMR1** y reestablecerá el **TIMER1** y el **CCPR1,** funciona como un registro de período, capaz de provocar periódicamente interrupciones.

Módulo CCP2:

El registro de Captura / Comparación / PWM (CCPR2) de 16 bits está formado por dos registros de 8 bits: **CCPR2L** (byte bajo) que ocupa la dirección lBh y **CCPR2H** (byte alto) que ocupa la dirección 1Ch. El registro de control del **CCP2** es el **CCP2CON** que se encuentra en la dirección lDh. El modo de disparo especial se genera por igualdad en la comparación con el **TIMER1** lo que provoca que se reestablezca el **TIMER1** y comience una conversión A/D, si el convertidor A/D está habilitado. Con lo cual puede realizar la conversiones A/D de forma periódica sin en control del programa de instrucciones.

Modo CCP	Recurso de Timer
Captura	Timer1
Comparación	Timer1
PWM	Timer2

Modo CCP – Recurso de TIMER. En los módulos de captura y comparación intervienen el TMR1, en PWM el TMR2.

INTERACCIÓN DE LOS DOS MÓDULOS CCP

Modo CCPx	Modo CCPy	Interacción
Captura	Captura	La misma base de tiempos de TMR1
Captura	Comparación	El comparador debe configurarse para el modo de disparo especial que pone a cero el TMR1
Comparación	Comparación	El Comparador(es) debe configurarse para el modo de disparo especial que pone a cero el TMR1
PWM	PWM	El PWM tendrá la misma frecuencia y proporción de actuación (interrupción de TMR2)
PWM	Captura	Ninguna
PWM	Comparación	Ninguna

Registro CCP1CON (dirección 17h)/Registro CCP2CON (dirección 1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
---		CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
Bit 7							Bit 0

bit 7-6:**No implementados**: Se lee como "0"

bit 5-4:**CCPxX: CCPxY**: bit menos significativos de PWM

Modo Captura sin usar

Modo Comparación sin usar

Modo PWM: Estos dos bit son los menos significativos del ciclo de PWM. Los ocho bits más significativos se encuentran en CCPRXL.

bit 3-0:**CCPxM3-.CCPxM0**; bit de selección del modo de trabajo del módulo comparador **CCPX**.

CCPxM3: CCPxM0	MÓDO DE TRABAJO DEL MÓDULO
0000	Módulo Caputa/Comparación/PWM desactivado (reset del módulo CCPx)
0100	Modo de captura por flanco descendente RCy/CCP
0101	Modo de captura por flanco ascendente en RCy/CCPx
0110	Modo de captura, cada 4 flancos ascendentes en RCy/CCPx
0111	Modo Captura. Cada 16 flancos ascendentes en RCy/CCPx
1000	Modo comparación, activa la patilla, se pone a 1 RCy/CCPx al coincidir los valores (el bit CCPxIF se pone a uno)
1001	Modo de comparación se pone a 0 la patilla RC/CCPx al coincidir los valores (el bit CCPxIF se pone a uno)
1010	Modo de comparación, genera una interrupción software (el bit CCPxIF se pone a 1, el pin de CCPx no es afectado)
1011	Modo de comparación, en el que se produce un disparo especial para cada módulo (el bit CCPxIF se pone a uno, el pin CCPx no es afectado); CCP1 resetea TMR1; CCP2 resetea TMR1 y comienza una conversión de A/D (si el módulo de A/D se habilita)
11xx	Modo de PWM

MODULO CONVERTIDOR ANALÓGICO/DIGITAL (A/D)

El módulo de conversión **Analógico/Digital** dispone de cinco entradas para los dispositivos de 28 pines y ocho para los otros dispositivos de la familia.

A través de la entrada analógica se aplica la señal analógica a un condensador de captura y retención (*sample and hold*) que después se introduce en el

convertidor. El convertidor de aproximaciones sucesiva da como resultado una palabra de 10 bits.

El convertidor A/D puede seleccionar como tensión de referencia la interna VDD y masa o bien una externa que se introduzca entre RA3/AN3/V_{REF+} y RA3/AN3/V_{REF}. Cuando se selecciona la tensión de referencia externa, hay que tener en cuenta que existen algunas limitaciones; el máximo voltaje aplicado a la patilla V_{REF+} (RA3/AN3) será V_{DD}+0,3V y el mínimo V_{DD}-2,5V. En cuanto a la tensión V_{REF-} (RA2/AN2) la mínima tensión será V_{SS}-0,3V y la máxima V_{REF+} - 2V, así por ejemplo, si la tensión de alimentación es de 5V, la tensión en RA3/AN3 no podrá exceder de 5V. Por lo que el máximo voltaje en V_{REF-} será de 3V. Siempre se ha de cumplir que $V_{REF+} - V_{REF-} \leq 2V$.

El convertidor A/D tiene como característica especial el ser capaz de seguir trabajando mientras el dispositivo esté en el modo SEEP. Para ello el oscilador interno RC debe conectarse al conversar.

El módulo de A/D tiene cuatro registros. Estos registros son:

- **ARDES** : Parte alta del resultado de la conversión
- **ADREL** : Parte baja del resultado de la conversión
- **ADCON0**: Registro de Control 0
- **ADCON1**, Registro de Control 1

Los registros **ADCON0** de control del funcionamiento del conversar se muestra seguidamente al igual que el registro **ADCON1** de configuración de los pines del puerto. Los pines del PORTA pueden configurarse como entradas analógicas (RA, también puede ser entrada de tensión de referencia) o como E/S digital.

Puede encontrarse información adicional sobre los convertidores A/D de rango medio de la familia PICmicro^{Tm} en el manual de referencia (DS33023).

Registro ADCON0 (dirección lFh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	**ADCS0**	**CHS2**	**CHS1**	**CHS0**	**G0/#DONE**	**---**	**ADON**
Bit 7							Bit 0

bit 7-6:**ADCS1:ADCS0**: bit de selección de reloj para el Convertidor A/D

00 Fosc/2

01 Fosc /8

10 Fosc/32

11 F_{RC} (Procede del oscilador RC interno)

bit 5-3:**CH2:CH0:** bit de selección del canal analógico

000 = Canal 0, (RA0/AN0)

001 = Canal 1, (RA1/AN1)

010 = Canal 2, (RA2/AN2)

011 = Canal 3, (RA3/AN3)

100 = Canal 4, (RA4/AN4)

101 = Canal 5, (RA5/AN5); (28 pines no tienen este canal)

110 = Canal 6, (RA6/AN6); (28 pines no tienen este canal)

111 = Canal 7, (RA7/AN7); (28 pines no tienen este canal)

bit 2: **GO/#DONE**. bit de estado de la conversión A/D

Si ADON=1

1= La conversión A/D está en marcha (mientras está a 1 se está realizando la conversión)

0 = La conversión ha finalizado. (el bit se pone a cero automáticamente por hardware cuando la conversión A/D finaliza) el resultado de la conversión aparece en ADRDES:ADRESL

bit 1: **No implementado**: Se lee como “0”

bit 0: **ADON**: bit de puesta en marcha

1 = El convertidor A/D está operativo

0 = El convertidor A/D está apagado y no consume corriente.

Registro ADCON1 (dirección 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	**---**	**---**	**---**	**PCFG3**	**PCFG2**	**PCFG1**	**PCFG0**
Bit 7							Bit 0

bit 7: **ADFM**: Selecciona el formato del resultado de la conversión A/D

1 = Pone en el registro **ARDESH** los seis bit de mayor peso a "0" (alineación a la derecha).

0 = pone los 6 bit de menor peso del registro **ADRESL** a "0" (alineación a la izquierda).

Por lo tanto los 16 bits que forman el registro **ARDESH-ARDESL** con **ADFM**=1 tiene los 6 bit de mayor peso a cero y con **ADFM**=0 los 6 bit de menor peso están a cero

bit 6-4: No implementados: Se leen como cero

bit 3-0: **PCFG3:PCFG0**: bits de configuración de los canales de entrada del convertidor A/D. Se utilizan para configurar las patillas como E/S digital o como entrada analógica de acuerdo con la siguiente tabla:

PCFG3: PCFG0	AN7(1) **RE2**	**AN6(1) RE1**	**AN5(1) RE0**	**AN4 RA5**	**AN3 RA3**	**AN2 RA2**	**AN1 RA1**	**AN0 RA0V**	**V_{REF+}**	**V_{REF-}**	**CHAN/ REFS**
0000	A	A	A	A	A	A	A	A	V_{DD}	V_{SS}	8/0
0001	A	A	A	A	V_{REF+}	A	A	A	RA3	V_{SS}	7/1
0010	D	D	D	A	A	A	A	A	V_{DD}	V_{SS}	5/0
0011	D	D	D	A	V_{REF+}	A	A	A	RA3	V_{SS}	4/1
0100	D	D	D	D	A	D	A	A	V_{DD}	V_{SS}	3/0
0101	D	D	D	D	V_{REF+}	D	A	A	RA3	V_{SS}	2/1
011X	D	D	D	D	D	D	D	D	V_{DD}	V_{SS}	0/0
1000	A	A	A	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	RA3	V_{SS}	6/0
1010	D	D	A	A	V_{REF+}	A	A	A	RA3	V_{SS}	5/1
1011	D	D	A	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	4/2
1100	D	D	D	A	V_{REF+}	V_{REF-}	A	A	RA3	RA2	3/2
1101	D	D	D	D	V_{REF+}	V_{REF-}	A	A	RA3	RA2	2/2

1110	D	D	D	D	D	D	D	A	V_{DD}	V_{SS}	1/0
1111	D	D	D	D	V_{REF+}	V_{REF-}	D	A	RA3	RA2	½

A= Entrada Analógica; D = E/S Digital; (1) Estos canales no están disponibles para los dispositivos de 28 pines

Los registros **ADRESH:ADRESL** contienen los 10 bits resultado de la conversión A/D. Cuando se completa la conversión A/D, el resultado se guarda en los registros y se pone a cero el bit **GO/DONE** y el flag de fin de conversión **ADIF** (**PIR1**<6>) se pone a 1.El diagrama de bloques del módulo conversar A/D es el de la Figura 24.

Después de que el conversar A/D se ha configurado como se quiere, la selección del canal debe realizarse antes de hacer la adquisición. Los canales de entrada analógica deben tener los correspondientes bits del registro **TRIS** seleccionados como entradas. Para determinar el tiempo de muestreo, ver la sección 9.3.

Para realizar una conversión A/D deben seguirse los siguientes pasos:

1. Configurar el módulo conversor A/D

- Configurar los pines que actúan como entradas analógicas, las señales que harán de tensión de referencia la que trabajarán como E/S digitales (**ADDCON1**).
- Seleccionar el canal de entrada. A/D (**ADCON0**).
- Seleccionar la fuente de la señal de reloj para la conversión A/D (**ADON0**).
- Activar el módulo de conversión A/D (**ADCON0**)

2. Activar si, se desea, la interrupción. del módulo conversor A/D

- Poner a cero el bit **ADIF**
- Poner a uno el bit **ADIE**
- Poner a uno
- Poner a una los bits habilitadores **GIE** y **PEIE**

3. Esperar el tiempo requerido para la adquisición

4. Inicio de la conversión

- Poner a 1 el bit **GO/#DONE** (**ADCON0**)

5. Tiempo de espera para terminar la conversión A/D que puede detectarse por:

- Exploración de bit **GO/#DONE** , que al finalizar la conversión toma el valor “0”
- Esperar que se produzca una interrupción si se ha programado al finalizar la conversión.
- Aunque no se autorice la interrupción el flag. **ADIF** se pone a “ 1 “ al finalizar la interrupción.

6. Leer el par de registros **ARDESH:ADRESL** donde se encuentran los 10 bits que resultan de la conversión y poner a cero el flag **ADIF**.

7. Para una nueva conversión volver a los pasos 1 y 2 , El tiempo de conversión por bit está definido por T_{AD}, se exige un mínimo de $2*T_{AD}$ para realizar una nueva conversión.

El diagrama de flujos que deberá seguir el programa para realizar una conversión A/D puede ser la que se muestra en la Figura 23

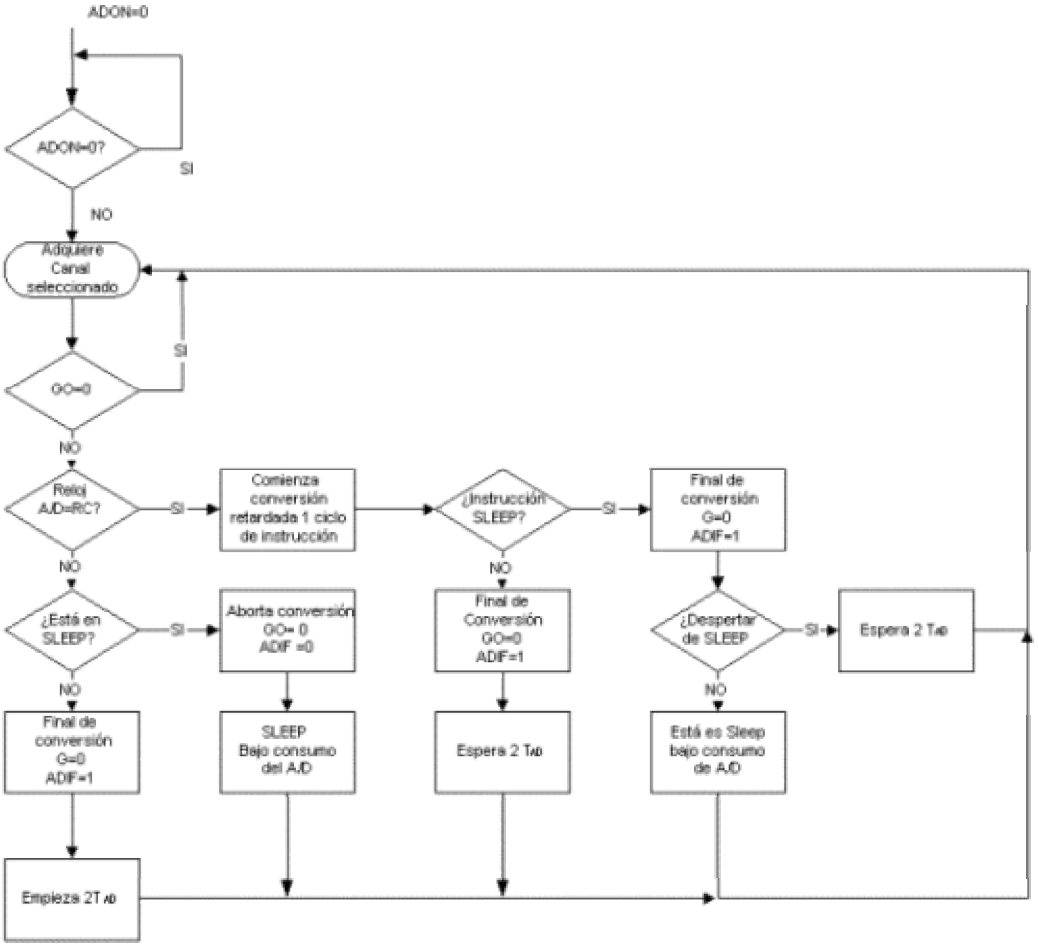


Diagrama de flujos que deberá seguir el programa para realizar una conversión A/D

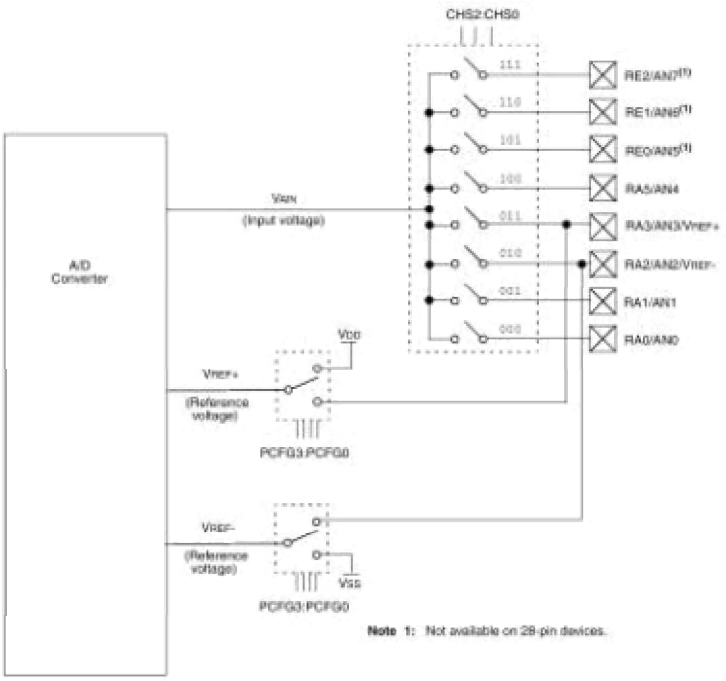
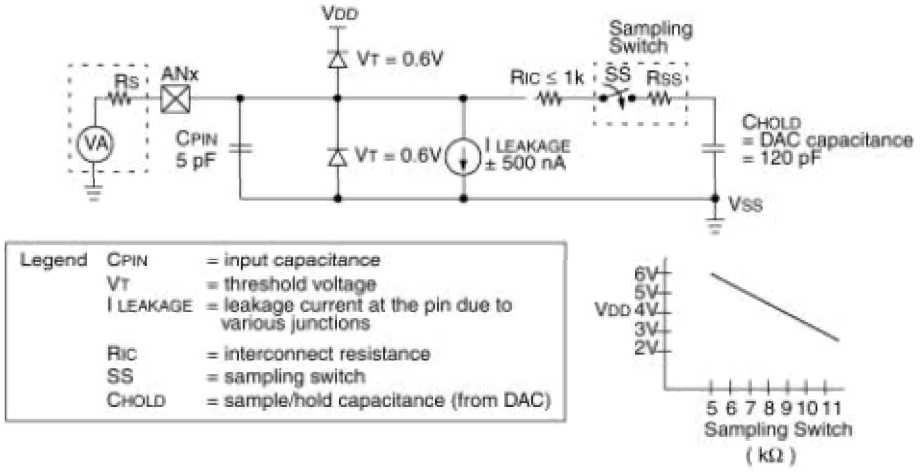


Diagrama de bloques del convertidor A/D

Requisitos para la Adquisición A/D

Para que el convertidor A/D pueda tener la exactitud especificada, debe permitirse la carga del condensador (**C_{HOLD}**) que mantiene la entrada a un nivel de tensión. El modelo de una entrada analógica se muestra en la Figura 25.



Estructura de un pin de un canal Analógico

De la impedancia de la fuente (**R_S**) se toma una muestra a través del interruptor (**R_{SS}**) la impedancia afecta directamente al tiempo de carga del condensador **C_{HOLD}**. La impedancia recomendada para las fuentes analógicas es de 10 kΩ. Cuando la impedancia disminuye también disminuye el tiempo de adquisición. Antes de realizar, la conversión debe seleccionarse el canal de la entrada analógica.

Para calcular el tiempo mínimo de adquisición mínimo se puede utilizar la siguiente ecuación, que asume un error de ½ LSB para 1024 pasos del convertidor A/D.

$$T_{ACQ} = Amplifier\ Settling\ Timer + Holding\ Capacitor\ Charging\ Time + Temperature\ Coefficient =$$
$$= T_{AMP} + T_C + T_{COFF} \approx 19.72\,\mu s$$

Selección del reloj del convertidor A/D

Se define por T_{AD} al tiempo de conversión por bit. El convertidor A/D requiere un mínimo de 12 T_{AD} para la conversión de los 10 bits, La señal de reloj para la conversión A/D se selecciona por software mediante los bits **ADCS1:ADCS0**

- · 2 T_{OSC}
- · 8 T_{OSC}
- · 32 T_{OSC}
- · Oscilador interno RC (2-6 µs)

Para realizar conversiones correctas el reloj del convertidor A/D debe seleccionarse para asegurar un tiempo mínimo de T_{AD} de 1,6 mS.

La tabla siguiente muestra los tiempos de T_{AD} dependiendo de la señal de reloj seleccionada para la conversión A/D

Fuente del Reloj A/D		Frecuencia máxima del dispositivo
Operación	**ADCS1:ADCS0**	**Máx**
$F_{OSC}/2$	0 0	1.25 MHz
$F_{OSC}/8$	0 1	5 MHz
$F_{OSC}/32$	1 0	20 MHz
RC (1,2,3)	1 1	Nota 1

Nota.-

1. Si la fuente es el oscilador RC tiene un T_{AD} típico de 4 µs, pero puede variar entre 2 –6 µs.

2. Cuando la frecuencia del dispositivo es mayor de 1MHz, la fuente del oscilador RC para la conversión A/D se recomienda solo si se trabaja en el modo Sleep.

3. Para dispositivos de tensión mayores (LC), comprobar las características eléctricas

Configuración de los Pines del Puerto Analógico

Los registros **ADCON1** y **TRIS** controlan la configuración de los pines del puerto del convertidor A/D. Cuando se desea configurar una entrada como analógica, debe configurarse como entrada los bits correspondientes del registro **TRIS**. Si el bits del **TRIS** correspondiente se pone a cero (salida), el valor digital de la salida (V_{OH} o V_{OL}) se convertirá.

El funcionamiento del convertidor A/D es independiente de los bits **CHS2:CHS0** y del valor del **TRIS**:

Notas:

1. Al leer el registro del puerto, cualquier pin configurado como un canal analógico se lee como un cero (nivel bajo). Los pines configurados como entradas digitales convertirán la entrada analógica. Los niveles analógicos de una entrada digital no afectarán la exactitud de la conversión.

2. Los niveles analógicos en cualquier pin de entrada digital (**AN7:AN0**) pueden causar que el buffer de entrada consuma una corriente superior a la de las especificaciones del dispositivo.

Conversiones A/D

Si se pone a cero el bit **GO/#DONE** durante la conversión, se aborta la conversión actual. El par de registros no se modificarán parcialmente con los bit que se hayan completado hasta el momento. Es decir, los registros **ADRESH:ADRESL** seguirán conteniendo el valor de la última conversión completa (o el último valor que se haya escrito en **ADRESH:ADRESL**) después de abortar la conversión A/D, es requerido el T_{AD} de espera para realizar la próxima adquisición comience. Después de 2 T_{AD} de espera, la adquisición en cauce se comienza automáticamente. En la Figura 26, después de poner el bit **GO** a uno, la primera vez el segmento tiene un T_{CY} mínimo y un T_{AD} máximo.

Nota:

El bit **GO/#DONE** no debe ponerse a uno en la misma instrucción que se pone en **ON** el convertidor A/D.

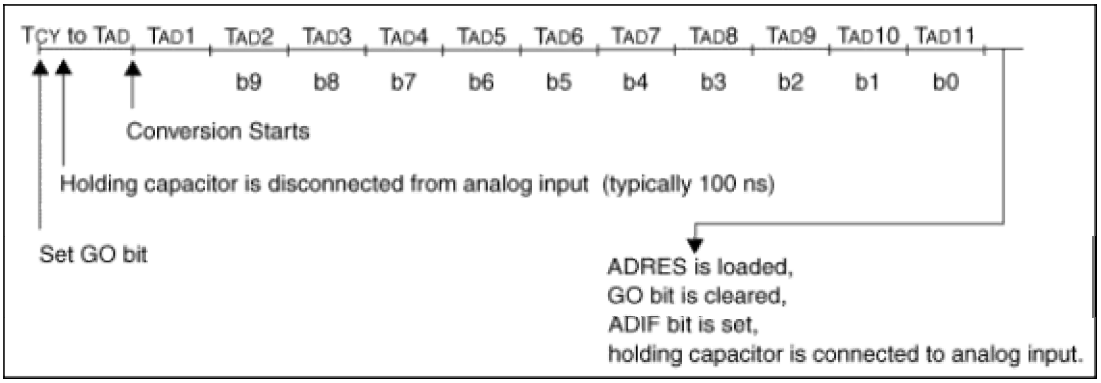
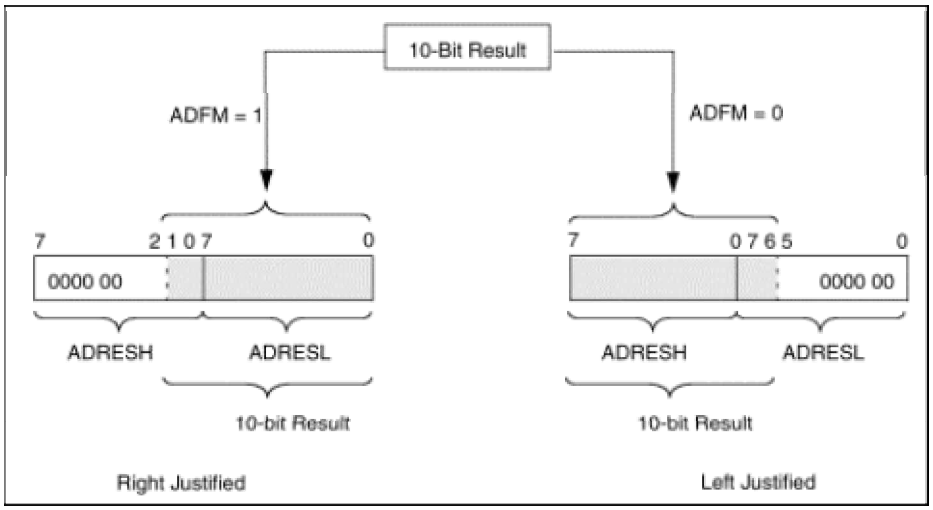


Figura 26 -Ciclos T_{AD} del convertidor A/D

Registro de resultados de la conversión A/D

El par de registros **ADRESH:ADRESL** se carga con el resultado de 10 bits de la conversión A/D. Este par de registros se extienden hasta 16 bits. El módulo A/D tiene la posibilidad de justificar el resultado de 10 bits dentro de los 16 bits de la pareja de registros. La selección del formato de justificación a la izquierda o derecha se realiza con el bit **ADFM** (**ADCON1** <7>) . Los bits restantes (a los 10 de la conversión) se llenan con ceros.

Estos dos registros cuando el convertidor A/D está en OFF y no se utiliza, pueden utilizarse como dos registros de 8 bits de propósito general.



Justificación de los resultados de la conversión A/D

Funcionamiento del Convertidor A/D en el modo Sleep

El módulo del convertidor A/D puede trabajar durante el modo Sleep. Para ello se debe poner como fuente de reloj el modo **RC** (**ADCS1.ADCS0** = 1l). Cuando la fuente de reloj es el RC interno, el módulo A/D espera un ciclo de máquina antes de empezar la conversión. Esto permite eliminar el ruido de la conversión digital, Cuando la conversión finaliza el bit **GO/#DONE** se pone a cero y el resultado se carga en, la pareja de registros **ADRESH:ADRESL**. Si la interrupción por final de conversión A/D está habilitada, el dispositivo se despierta. Si la interrupción no está habilitada, el módulo A/D se apagará aunque el bit **ADON** (**ADCON0** <0>) permanezca a uno.

Cuando el reloj del convertidor **A/D** es otra fuente distinta del RC interno, la instrucción sleep aborta la conversión actual y el módulo **A/D** se apaga, aunque el bit **ADON** permanezca a uno. Apagando el módulo del convertidor **A/D** el consumo es más bajo.

Efectos del Reset

Cuando se fuerza un Reset se reestablecen todos los registros, esto obliga a apagar el convertidor A/D y se aborta cualquier conversión en curso. El valor de los registros **ADRESH:ADRESL** no es modificado por un Power-on Reset.

Registros asociados al Convertidor A/D

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh,8Bh 10Bh,18Bh	**INTCON**	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	**PIR1**	PSPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Bh	**PIE1**	PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1E h	**ADDRESH**	Parte alta de la conversión A/D								xxxx xxxx	uuuu uuuu
9E h	**ADDRESL**	Parte baja de la conversión A/D								xxxx xxxx	uuuu uuuu
1Fh	**ADCON0**	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/#DONE		ADOM	0000 00-0	0000 00-0
9Fh	**ADCON1**	ADFM	---	---	---	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	**TRISA**	---	---	Registro de direccionamiento de datos de PORTA						--11 1111	--11 1111
05h	**PORTA**	---	---	Latch de datos para escritura del PORTA						--0x 0000	--0u 0000
89h [(1)]	**TRISE**	IBF	OBF	IBOV	PSPMODE	---	Bit de direccionamiento del PORT			0000 - 111	0000 - 111
09h [(1)]	**PORTE**	---	---	---	---	---	RE2	RE1	RE0	---- -xxx	---- uuuu

Características del convertidor A/D

Symb.	Características	Mínimo	Típica	Máximo	Unid.	Condiciones
N_R	Resolución	--	--	10-bits	Bit	$V_{REF}=V_{DD}$ = 5.12V $V_{ss} \leq V_{AIN} \leq V_{REF}$
E_{IL}	Error absoluto	--	--	<±1	LSb	$V_{REF}=V_{DD}$ = 5.12V $V_{ss} \leq V_{AIN} \leq V_{REF}$
E_{DL}	Error Direccional	--	--	<±1	LSb	$V_{REF}=V_{DD}$ = 5.12V $V_{ss} \leq V_{AIN} \leq V_{REF}$
E_{GN}	Error de ganancia	--	--	<±1	LSb	$V_{REF}=V_{DD}$ = 5.12V $V_{ss} \leq V_{AIN} \leq V_{REF}$
--	Monotonicidad	--	Garantizada	--	--	$V_{ss} \leq V_{AIN} \leq V_{REF}$
V_{REF}	Voltaje de referencia (V_{REF+}- V_{REF-})	2.0V	--	V_{DD}+0.3	V	Absoluta especificación. Asegura la precisión de 10 bits
V_{REF+}	Voltaje de referencia alto	AV_{DD}-2.5V		AV_{DD}+0.3V	V	
	Voltaje de referencia bajo	AV_{SS} – 0.3	--	V_{REF} + 0.3	V	
V_{AIN}	Voltaje analógico de entrada	V_{SS}-0.3	--	V_{REF} + 0.3	V	
Z_{AIN}	Impedancia recomendada para la fuente de voltaje analógico					

I_{AD}	Corriente de conversión Estándar Extendida	-- --	220 90	— —	µA µA	Promedio de consumo de corriente cuando el A/D está activado (Nota1)
I_{REF}	Corriente de entrada V_{REF} (Nota 2)	10 --	-- --	1000 10	µA µA	Durante la adquisición V_{AIN} basada en la diferencia V_{HOLD} y V_{AIN} a carga Durante el ciclo de conversión.

Nota 1: Cuando el módulo A/D está “apagado” (OFF) no consume corriente

MÓDULO DE COMUNICACIONES SERIE SÍNCRONA: MSSP

El módulo de comunicaciones serie síncrona (MSSP) es un interface serie, útil para comunicar con otro periférico o dispositivos. Estos periféricos los dispositivos pueden ser memorias EEPROMs serie, registros de desplazamiento, controladores de displays, convertidores A/D, convertidores D/A, etc.

El módulo de MSSP pueda operar en uno de dos modos:

- · Serial Peripheral Interface (SPI)
- · Inter-Integrated el Circuito (I2C)

Para el control de este módulo hay dos registros de configuración se trata del registro **SSPCON :** Registro de control del Puerto Serie Síncrono (*Sync Serial Port Control Register*) en la dirección 14h y el **SSPCON2**: Registro de Control2 del Puerto Serie Síncrono *(Sync Serial Port Control Register2*) En la dirección 91h **.** Además se dispone del registro **SSPSTAT** : Registro de estado del puerto Serie (*Sync Serial Port Status Register*) en la dirección 94, que seguidamente se describen.

Registro SSPSTAT: Registro de estado del puerto Serie (Dirección 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/#A	P	S	R/#W	UA	BF
Bit 7							Bit 0

bit 7: SMP: Fase de datos de entrada SPI

<u>Modo Master SPI</u>

1 = Muestra datos de entrada al final del tiempo de salida

0 = Muestra datos de entrada a la mitad del tiempo de salida

<u>Modo esclavo SPI</u>

SMP debe ser borrado cuando SPI se emplea en modo esclavo

Modo I^2C Master o Esclavo

1= Deshabilita control de variaciones para velocidad estándar (100 KHz y 1MHz)

0= Habilita control (400KHz)

bit 6: **CKE**: Selección del flanco de reloj SPI (ver los diagramas de tiempos de las Figuras 10.6, 10.7 y 10.9)

CKP = 0

1= Dato transmitido en flanco ascendente de SCK

0= Dato transmitido en flanco descendente de SDK

CKP = 1

1= Dato transmitido en flanco descendente de SCK

0= Dato transmitido en flanco ascendente de SCK

Modo I^2C Master o Esclavo

1= Niveles de entrada de acuerdo a SMBUS

0= Niveles de entrada conforme a I^2C

bit 5: **D/#A**: Bit Datos/Direcciones (solo en modo I^2C)

1= Indica el último byte recibido o transmitido es un dato

0= Indica que el último byte recibido o transmitido es una dirección

bit 4: **P**: Bit de Stop (Solo en modo I^2C. Ese bit es borrado cuando se deshabilita el módulo SSP, o cuando el último bit Start es detectado, SSPEN es borrado)

1= Indica que se ha detectado un bit de Stop (este bit es “0” en RESET)

0= Bit de Stop no detectado.

bit 3: **S**: Bit de Start (Solo en modo I^2C. Este bit es borrado cuando se deshabilita el módulo SSP, o cuando el bit último bit Start es detectado, SSPEN es borrado)

1= Indica que se ha detectado un bit de Start (este bit es “0” en RESET)

0= Bit de Start no detectado

Bit 2: **R/#W**: Bit de información Lectura/escritura (solo en modo I^2C)

1= Lectura

0= Escritura

Modo I^2C Modo Esclavo

1= Lectura

0= Escritura

Modo I^2C Modo Master

1= Transmisión en progreso

0= Transmisión no en progreso

bit 1 **UA**: Activación de dirección (sólo en modo I^2C 10-bits)

1= Indica que es necesario actualizar la dirección en el registro SSPADD

0= No es necesario actualizar la dirección

bit 0 **BF**: Bit estado de Buffer lleno

Receptor (Modo SPI e I^2C)

1= Recepción completa, SSPBUF está lleno

0= Recepción no completa, SSPBUF está vacío

Transmisor (Modo I^2C)

1= Transmisión en progreso, SSPBUF está lleno (no incluye ACK y bit de Stop)

0= Transmisión completa, SSPBUF está vacío (no incluye ACK y bit de Stop)

Registro SSPCON: Registro de control del puerto Serie (Dirección 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

Bit 7: **WCOL**: Bit detector de colisión

1= El registro SSPBUF se ha escrito cuando hay una transmisión en proceso (Se debe borrar por software)

0= No hay colisión

bit 6: **SSPOV**: Bit detector de desbordamiento en recepción

En modo SPI

1= Se recibe un nuevo byte cuando el registro SSPBUF aún mantiene los datos anteriores. En caso de rebose, los datos del registro SSPSR serán falsos, sólo puede ocurrir en modo esclavo. Se deberá leer el SSPBUF, aún si sólo se envían datos, para anular el rebose. En modo master el bit no sube a uno ya que en cada nueva recepción/transmisión es inicializado por la escritura del registro SSPBU.

0= No hay desbordamiento

En modo I^2C

1= Se recibe un nuevo byte cuando el registro SSPBUF aún mantiene los datos anteriores. En modo de transmisión no tiene importancia, en cualquier caso debe borrarse por software.

0= No hay desbordamiento

bit 5: **SSPEN**: Bit de habilitación del módulo SSP (*Synchronous Serial Port*)

En modo SPI

1= Habilita puerto serie y configura SCK, SDO y SDI como patillas del puerto serie.

0= Deshabilita el puerto serie y configura estas patillas como puerto de Entrada/Salida..

En modo I^2C

1= Habilita el puerto serie y configura SDA y SCL como patillas del puerto serie

0= Deshabilita el puerto serie y configura estas patillas como puerto de Entrada/Salida.

En ambos casos, se deberán configurar como entradas o salidas adecuadamente.

Bit 4: **CKP**: Bit de selección de polaridad del reloj

En modo SPI

1= Reloj en nivel alto. Transmisión en flancos de bajada, en recepción de subida

0= Reloj en nivel bajo. Transmisión en flancos de subida, en recepción por flancos de bajada.

En modo I^2C Esclavo (En este modo no se usa como master)

Control de desbloqueo de SCK

1= Habilita reloj

0= Mantiene reloj en estado bajo.

Bit 3-0:**SSPM3:SSPM0**: Bit de selección del modo del SSP (Syncrhonous Serial Port)

0000	Modo Master del SPI, reloj = $F_{OSC}/4$	0110	Modo Esclavo I^2C, dirección de 7 bits
0001	Modo Master del SPI, reloj $=F_{OSC}/16$	0111	Modo Esclavo I^2C, dirección de 10 bits
0010	Modo Master del SPI, reloj $=F_{OSC}/32$	1000	Modo Master I^2C, reloj = $F_{=OSC}/(4x(SSPADD+1))$
0011	Modo Master del SPI, reloj =TMR2 salida/2	1011	Modelo Esclavo I^2C con soporte
0100	Modo Master del SPI, reloj = patilla SCK. Habilita patilla de control #SS	1110	Modo Esclavo I^2C, dirección de 7 bits de interrupción Start y Stop habilitados
0101	Modo Esclavo del SPI, reloj = patilla SCK. Deshabilitada patilla #SS, se puede emplear como I/O	1111	Modo Esclavo I^2C, dirección de 10 bits con bits de interrupción Start y Stop habilitados
1001, 1010, 1100 y 1101 Están reservados			

SSPCON2: Registro de control 2 del puerto Serie (dirección 91)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ASKSTART	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
Bit 7							Bit 0

Bit 7: **GCEN**: Bit de habilitación general (sólo en modo esclavo I^2C)

1= Habilita Interrupción cuando una llamada general es

recibida en el SSPST (0000h)

0= Deshabilita dirección de llamada general

bit 6: **ACKSTART**: Bit de estado de reconocimiento (solo en modo master I^2C)

Modo de Transmisión

1= Reconocimiento del esclavo no recibido

0= Reconocimiento del esclavo recibido

bit 5: **ACKDT**: Bit de reconocimiento (solo en modo master I^2C)

Modo de Recepción

Valor transmitido cuando el usuario indica una secuencia de reconocimiento y final de una recepción.

1= No reconocimiento

0= Reconocimiento

bit 4: **ACKEN**: Bit de habilitación de secuencia de reconocimiento (sólo en modo master I^2C)

Modo de Recepción

1= Indica una secuencia de reconocimiento en patillas SDA y SCL y transmite bit ACKDT. Automáticamente borrado por hardware.

0= Secuencia de reconocimiento en reposo

bit 3: **RCEN**: Bit de habilitación de recepción (sólo en modo master I^2C)

1= Habilita recepción en modo I^2C

0= Recepción en reposo

bit 2: **PEN**: Bit de habilitación de condición de Stop(sólo en modo master I^2C)

Control SCK desbloqueado

1= Indica una condición de Stop en SDA y SCL. Automáticamente borrado por hardware.

0= Condición de Stop en reposo

bit 1: **RSEN**: Bit de habilitación repetir condición Start (sólo en modo master I^2C)

1= Indica repetición de condición de Start en SDA y SCL. Automáticamente borrado por hardware.

0= Condición de repetición de Start en reposo.

bit 0: **SEN**: Bit de habilitación de condición Start (sólo en modo master I^2C)

1= Indica condición de Start en SDA y SCL. Automáticamente borrado por hardware

0= Condición de Start en reposo.

Modo SPI

Este modo permite la transferencia de datos de 8 bits en serie, que pueden ser transmitidos y recibidos de forma síncrona y simultanea. Para el establecimiento de la comunicación se utilizan tres hilos de conexión :

- *Serial Data Out* (SDO): Salida de datos Serie
- *Serial Data In* (SDI); Entrada de datos Serie
- *Serial Clock* (SCK), Reloj de Sincronización

Adicionalmente puede utilizarse un cuarto pin trabajando en modo de funcionamiento esclavo.

- *Slave Select* (#SS): Selección de Esclavo

Modo de trabajo

Al inicializarse el módulo **SPI**, varias especificaciones necesitan ser configuradas. Esto se en los registros **SSPSTAT** y **SSPCON** (**SSPCON**<5:0> y **SSPSTAT**<7:6>). Pudiéndose seleccionar las distintas modalidades de trabajo:

- Modo Master (donde la señal de reloj SCK será de salida)
- Modo Esclavo (donde la señal de reloj SCK será de entrada)
- Polaridad del Reloj: (Cambio de estado por flanco ascendente o descendente de SCK, y su estado de inactivad)
- Fase de datos de entrada (es decir, debemos de precisar el momento en el que se toma la muestra del dato de entrada, que puede ser en la mitad o al final del impulso).
- Selección del flanco del reloj (Transmisión por flanco de bajada o subida de SCK)

- Velocidad de SCK (Solo en modo Master)
- Selección del modo Esclavo, si trabajamos de esta forma.

En la Figura 28 se muestra el diagrama de bloques del módulo SSP, donde pueden apreciarse los cuatro pines que intervienen en las comunicaciones.

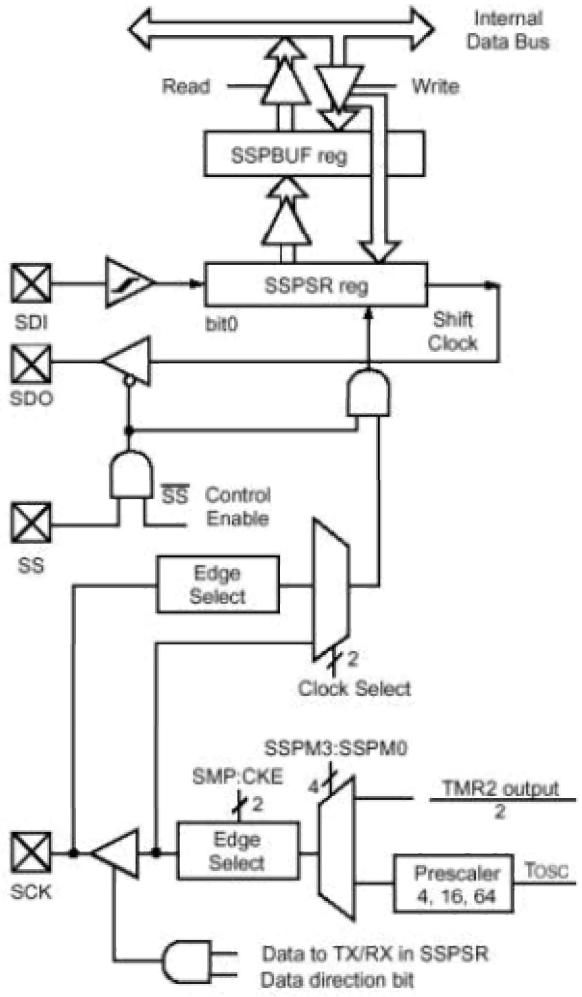


Diagrama de bloques del módulo SSP

Para habilitar el puerto de serie, deben configurarse adecuadamente los bits del registro **SSPCON** <5:0> y los bit **SMP** y **CKE** del registro **SSPSTAT** <7:6>. Cuando se inicializa el modo SPI es necesario especificar en el registro **SSPCON** la diferentes opciones de trabajo:

Modo *Master*, donde la señal **SCK** será salida

Modo *Slave*, donde la línea **SCK** será entrada

Polaridad de la señal de reloj, flanco ascendente o descendente.

Velocidad del SCK, evidentemente solo en el modo master.

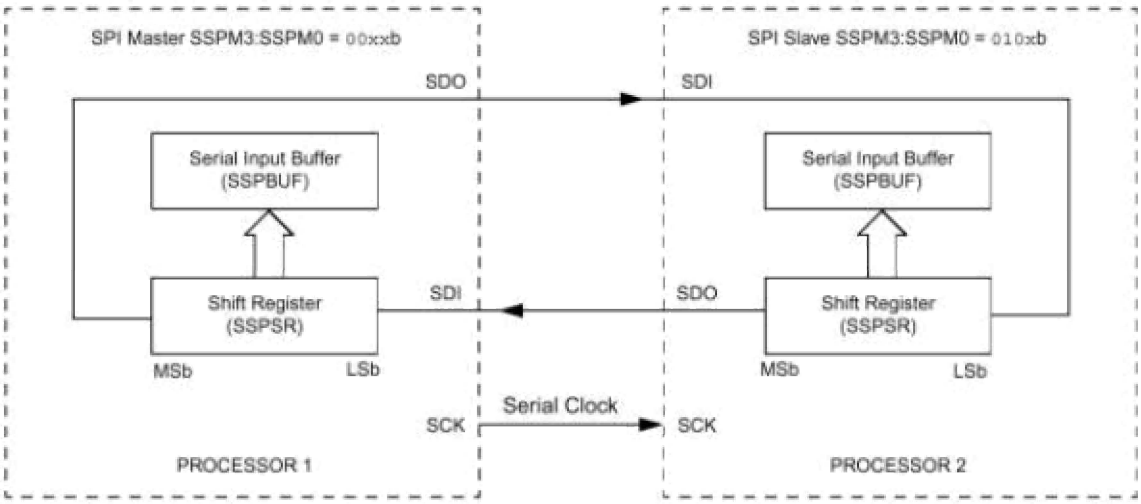
Definir el momento en el que se va a tomar la muestra del dato de entrada, que puede ser en la mitad o al final del impulso.

En el modo de recepción los datos recibidos se introducen en serie en el registro de desplazamiento **SSPSR** que, una vez haya recibido el octavo bit, pasarán al registro SSBUF en paralelo, poniéndose en ese momento los bits **SSPSTAT** <0> **BF** y **PIR1** <3> (**SSPIF**) a uno, indicando el primero de ellos que el buffer está lleno y el segundo para indicar que la recepción/transmisión ha finalizado. Si durante la recepción/transmisión se intenta escribir en el registro **SSPBUF**, la escritura será ignorada y el bit **WCOL** (**SSPCON**<7>) se pone a uno para indicar que ha habido "colisión". El bit **BF** se pone a cero, cuando se proceda a leer el registro **SSPBUF**, lo cual debe realizarse ante de que tenga lugar una nueva recepción.

El registro **SSPSR** no se puede leer o escribir directamente, y solo se puede acceder a él a través del registro **SSPBUF**. Por lo tanto, por el registro **SSPBUF** deben pasar forzosamente todos los datos tanto en transmisión como en recepción. En el siguiente ejemplo se muestra el proceso de carga en modo transmisión.

```
        bsf     STATUS, RP0     ;Selecciona el banco 1
LOOP    btfss   SSPSTAT, BF     ;¿Buffer lleno? (¿BF=1?)
        goto    LOOP            ;No, entonces esperar
        bcf     STATUS, RP0     ;Selecciona el banco 0
        movf    SSPBUF, W       ; SSPBUF à W
        movwf   RXDATA          ;Guardar el dato en una memoria temporal RAM
        movf    TXDATA, W       ; TXDATA à W
        movwf   SSPBUF          ;Nuevo dato a transmitir
```

Conexión Típica



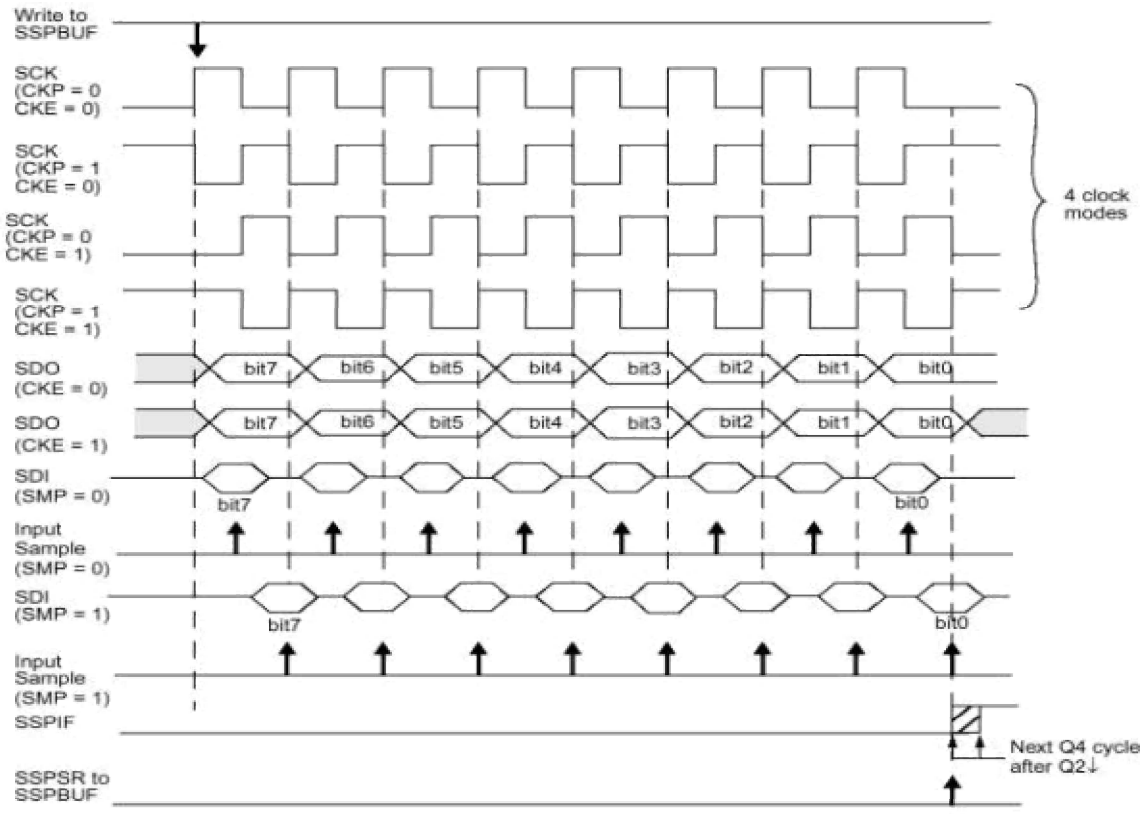
Conexión SPI Master/Slave sólo uno de los procesadores puede actuar cada vez como master.

En la Figura 29 se muestra la conexión entre dos procesadores, donde sólo uno puede actuar como master cada vez . El circuito que trabaja como master (Procesador 1) indica la transferencia de datos con la señal de reloj. Cada bit es desplazado en el registro por cada flanco programado de reloj; en el flanco opuesto es enclavado en el *latch*. Los dos procesadores deberán programarse con idéntica polaridad de reloj. Los circuitos a conectar por medio de este protocolo pueden ser dos o más periféricos o microcontroladores pero, como se ha comentado, solo uno puede trabajar cada vez como master.

Modo Master

Cuando el microcontrolador se configura como *Master*, los datos son transmitidos/recibidos cada vez que se escribe en el registro **SSPBU**. Si el módulo actúa como esclavo, la patilla **CLK** deberá configurarse como entrada para que la señal, presente en la patilla **SDI**, se vaya desplazando en el registro **SSPSR** por cada pulso de reloj recibido. El rango programable para el reloj con los bits **SSPM3:SSPM0** es de $F_{osc}/4$, $F_{osc}/16$, $F_{osc}/64$ y Timer2 (Salida/2).

Esto permite a una frecuencia de reloj máxima (a 20 MHz) una frecuencia de bits de 5MHz. La Figura 30 muestra el cronograma en modo *Master*. Cuando **CKE** =1, el dato de **SDO** es valido si es anterior a un flanco de reloj en **SDO**. El cambio en la entrada en el ejemplo es en el flanco de reloj de **SCK**. El cambio en el ejemplo de entrada se depende del estado del bit **SMP**. El tiempo de la señal corresponde al de carga de ls bit recibidos en el registros **SSPBUF**.



Cronograma del modo SPI (Modo Master). La señal de reloj actúa como motor del proceso.

En una operación de SLEEP, cuando está configurado como master, todos los módulos de reloj son detenidos por lo que la transición recepción permanecerá congelada hasta que el microcontrolador despierte. Momento en el que el módulo reanudará la transmisión o la recepción. En el modo esclavo, los desplazamientos sobre el registro son asíncronos, lo que permite seguir transmitiendo o recibiendo datos estando el procesador en estado Sleep. El microcontrolador despertará del SLEEP cuando los ocho bits se hayan recibido y, como es obvio, si la interrupción correspondiente está habilitada.

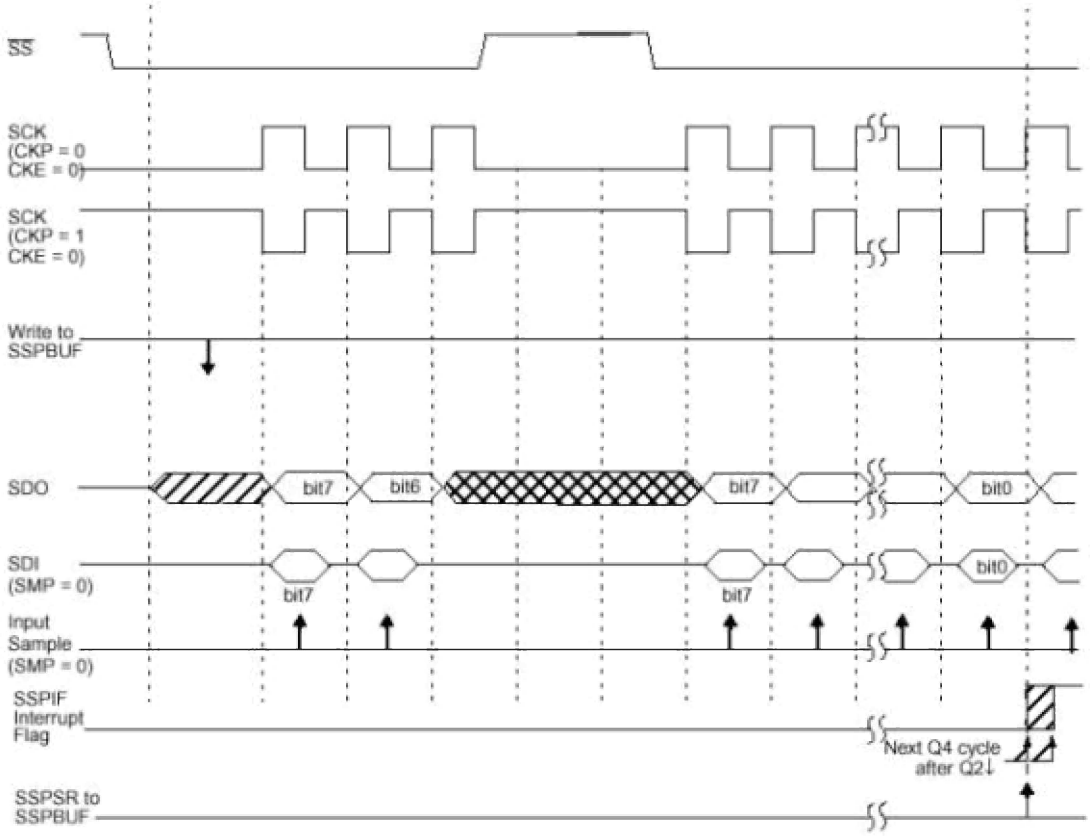
Un Reset, deshabilita el módulo **MSSP**, por lo que si se produce durante una transferencia ésta se dará por finalizada.

Modo Slave

Como ya se ha comentado, en este modo de trabajo los datos son transmitidos o recibidos con el control de la señal externa de reloj. Cuando el último bit ha sido enclavado, el flag de interrupción **SSPIF** (**PIR**<3>) se pondrá a uno. En este caso, como la fuente de reloj es externa, la patilla **SCK** deberá programarse como entrada.

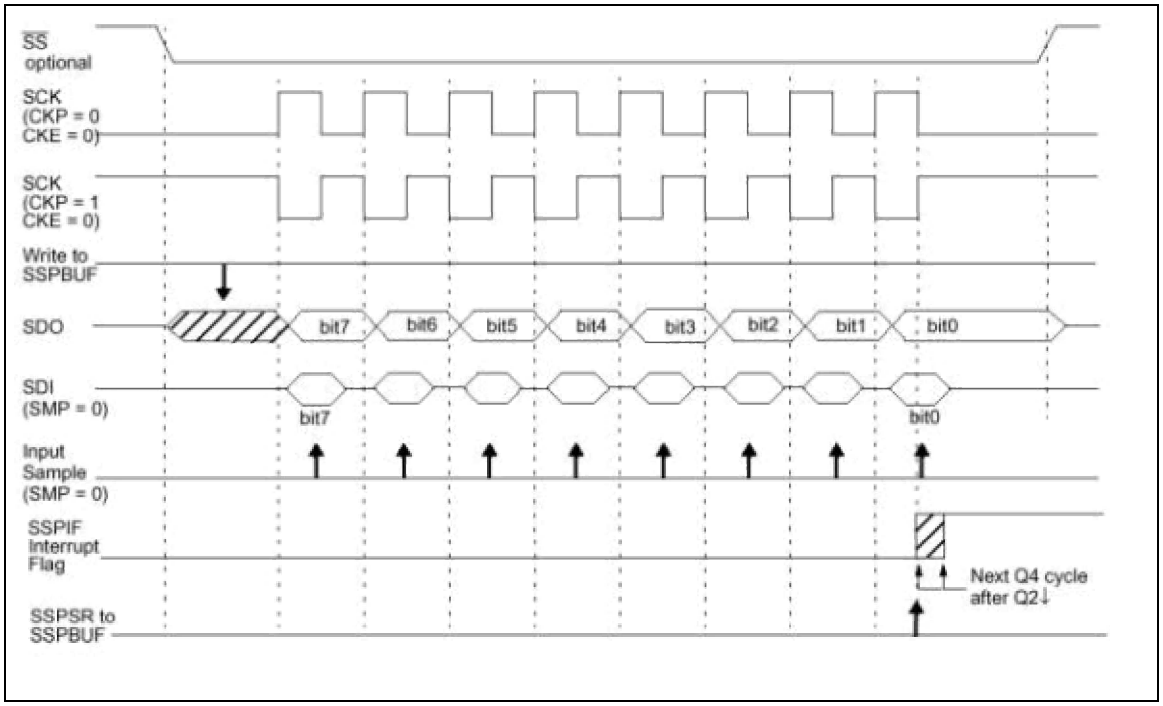
El pin **#SS** permite sincronizar el modo esclavo. Cuando este modo es habilitado (**SSPCON**<3:0> = 0110) el pin **#SSP** se comporta como una entrada, por lo que habrá que programarla como una tal (**TRISA**<5> =1). Forzando esta pastilla a nivel alto, el módulo **SPI** se Resetea y el bit **SSPEN** se borra.

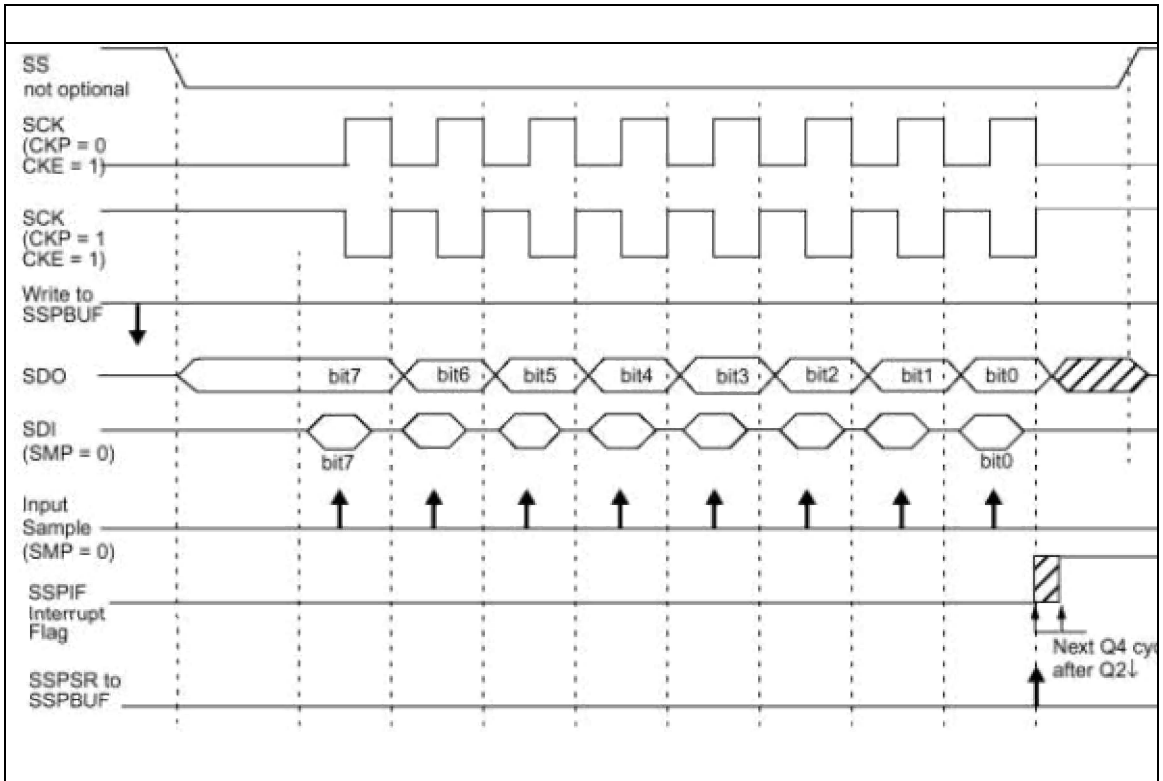
Es posible realizar transmisiones y recepciones con sólo dos hilos de conexión; en este caso las patillas **SDI** y **SDO** pueden unirse eliminando así la tercera conexión. Cuando la pastilla **SDI** necesita operar como receptora, la patilla **SDO** deberá ser configurada como una nueva entrada, de este modo, se deshabilita la transmisión por esta última y no crea conflictos con los datos de la **SDI**.



Cronograma de sincronización del slave

Cronograma del SPI en modo Esclavo (CKE=0)





Cronograma del SPI en Modo Esclavo (CK=1)

Registros Asociados con una operación SPI

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Bh,8Bh 10Bh,18Bh	**INTCON**	GIE	PEIE(1)	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	**PIR1**	PSPIF[1]	ADIF(1)	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	**PIE1**	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
87	**TRISC**	PORTC (Direcciónamiento de los Datos)								1111 1111	1111 1111
13h	**SSPBUF**	Registro Buffer Transmisión/Recepción del módulo SSP								xxxx xxxx	uuuu uuuu
14h	**SSPCON**	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPPM0	0000 0000	0000 0000
85h	**TRISA**	---	---	Registro de direccionamiento de datos de PORTA						--11 1111	--11 1111
94h	**SSPSTAT**	SMP	SKE	D/#A	P	S	R/W	UA	BF	--00 0000	--00 0000

Leyenda: x = desconocido, u = inalterado; - = no implementado se lee como ‘0’.
Las celdas sombreadas no son usadas para la conversión A/D

Nota 1: Estos registros/bits no están disponibles para los dispositivos de 28 pines

EL PUERTO MSSP EN MODO I^2C

El módulo MSP (Módulo de Comunicaciones Serie Síncrono) lleva a cabo todas las funciones para el bus I^2C tanto en modo Master como Slave (incluida la llamada general) y está provisto de interrupciones en los bits de Start y Stop. Tiene dos formatos de direcciones, de 7 y 10 bits de R/W (Ver la nota de aplicación AN578 de MICROCHIP referente al uso del módulo SSP utilizado en modo I^2C multi-Master).

Cuando los pines SCL y SDA se configuran como entradas, disponen de un filtro de interferencias (glitch) que trabaja en los modos de 100 KHz y 400 KHz. Cuando actúan como salidas, la velocidad de transferencia es independiente de la frecuencia de reloj del dispositivo. Estos dos pines se emplean para transferir datos, donde el pin SCL envía o recibe la señal de reloj y el pin SDA transmite o recibe los datos; ambas patillas son configuradas automáticamente cuando se habilita el modo I^2C del módulo SSP al poner a “1” el bit SSPEN (SSPCON<5>).

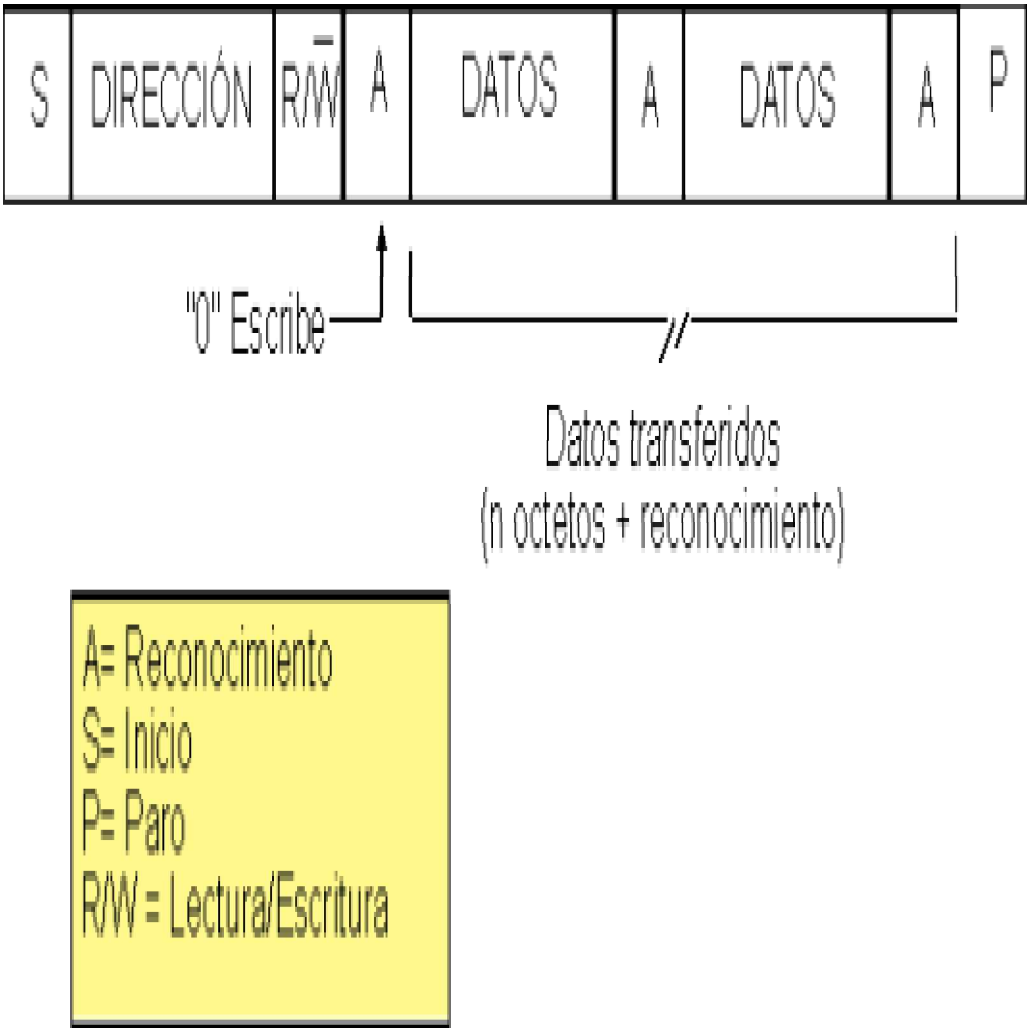


Diagrama de bloques del módulo SSP en modo I2C Slave. En él se puede ver los dos pines del circuito que intervienen en el proceso,

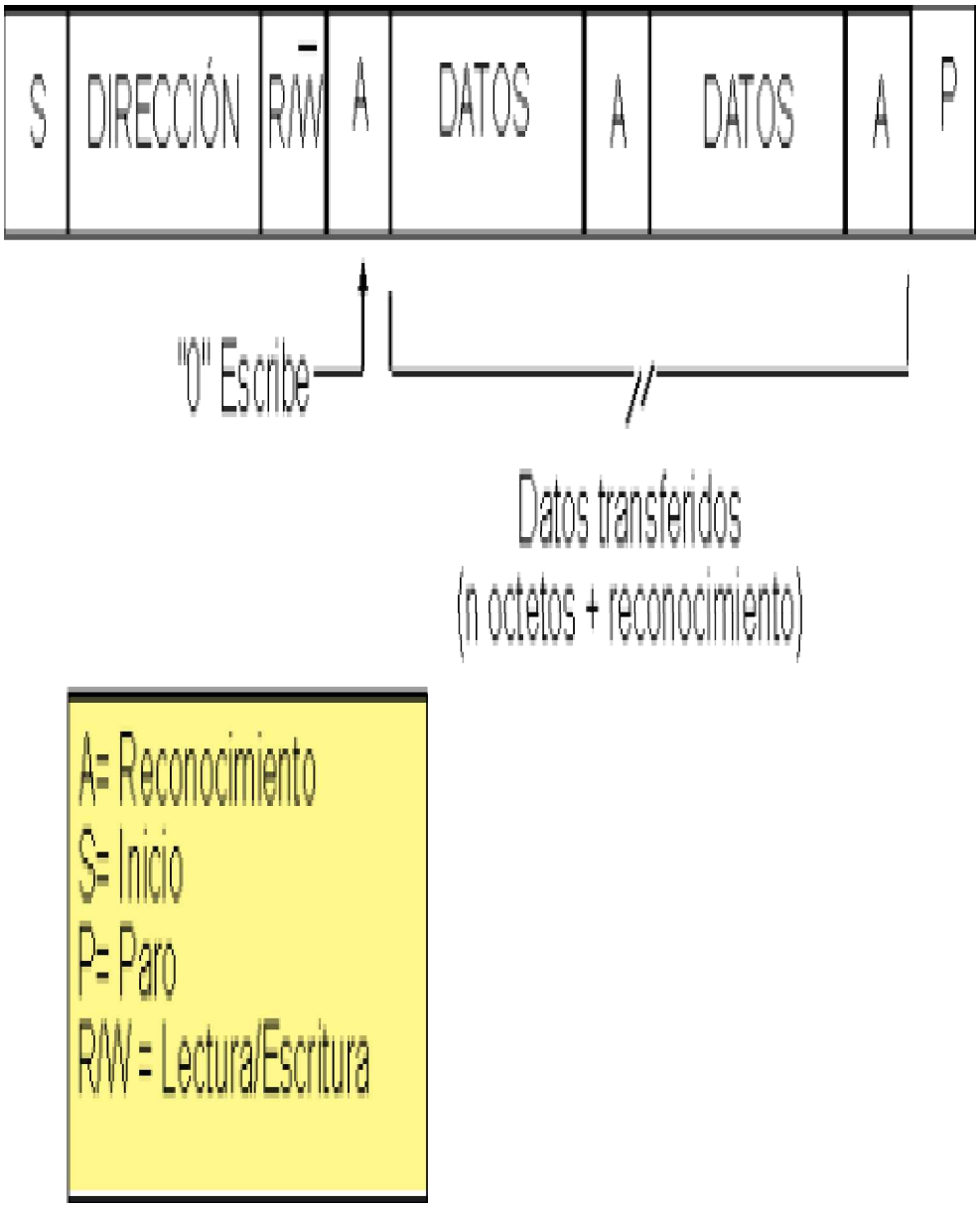
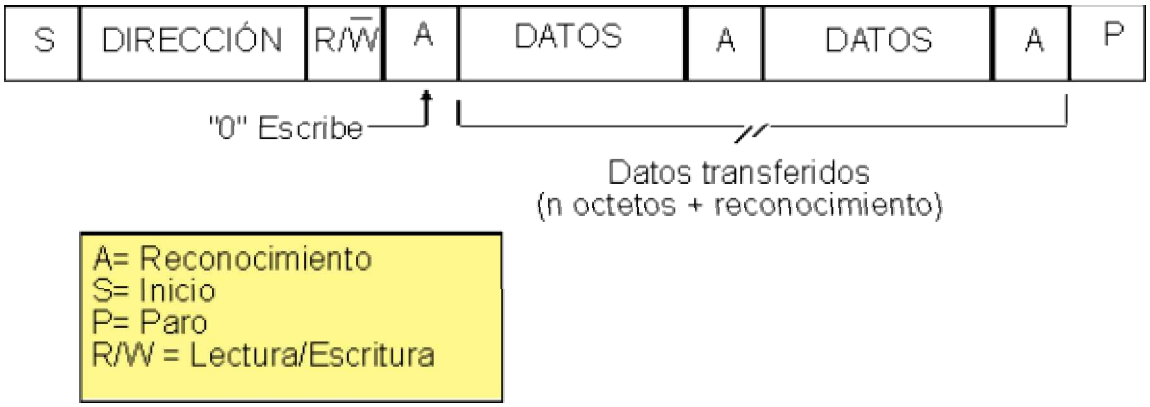


Diagrama de bloques del I²C en modo Master

Para realizar una transmisión en el bus I²C, siempre se inicia con una secuencia Start, seguida de los bits de dirección del circuito que actúa como esclavo y a continuación, el impulso que indica si se procederá a leer o a escribir en el esclavo seleccionado. Después del impulso R/W (primer byte) el circuito que actúa como Master espera recibir un bit (estado bajo) de reconocimiento (ACK) y si no se recibe, se aborta la transferencia. A continuación del ACK correcto, se envía otro byte que puede ser una dirección, un comando, etc. Si la secuencia de transmisión finaliza, el Slave pondrá a “1” la señal SDA (ACK alto) para que el master pueda generar la secuencia de Stop.



El Master Transmite información al Slave

En una operación I^2C con el módulo MSSP intervienen seis registros:

- Registro de control SSPCON
- Registro de control SSPCON2
- Registro de estado SSPSTAT
- Buffer serie de Transmisión / Recepción SSPBUF
- Registro de desplazamiento SSPSR, que no es accesible directamente.
- Registro de dirección SSPADD.

Los cuatro bits del registro de control SSPCON (SSPCON <3:0>) permiten en control del módulo I^2C en una de las siguientes formas:

- Modo Slave del I^2C (7 bits de dirección)
- Módulo Slave del I^2C (10 bits de dirección)
- Modo Master del I^2C, reloj = SC/4 (SSPADD+1)

Antes de seleccionar uno de los modos del I^2C, es necesario programar los pines SCL y SDA con los valores apropiados en los bits del registro TRIS. Con sólo seleccionar uno de los modos y colocando a “1” el bit SSPEN, las patillas SCL y SDA se configuran automáticamente como señal de reloj y datos respectivamente. Con el bit CKE (SSPSTAT<6>) se configuran los niveles de entrada para estas patillas. Cuando CKE es “1”, los niveles están de acuerdo a las especificaciones SMBUS; sin embargo, cuando es “0”, los niveles serán conformes a las especificaciones del bus I^2C.

El registro SSPSTAT, que es sólo de lectura, contiene el estado de los datos a transferir. Esta información incluye la detección de los datos bits de Start o Stop, indica si el byte recibido es un dato o una dirección de 10 bits y si será leído o escrito.

El registro SSBUF es de lectura / escritura y en él se guardan los datos a transferir por el bus o del que se leerán los datos recibidos. En el modo de transmisión los modos almacenados en este registro se transfieren al registro SSPSR desde el cual saldrán en serie por la línea SDA. Los dos registros forman un doble buffer que permiten que en recepción se “invierta” el sentido de los datos; son recibidos en serie por el registros SSPSR y transferidos al SSPBUF cuando la recepción finaliza al completarse un byte. Si se recibe otro byte sin que el anterior se haya leído, el bit SSPOV (SSPCON<6>) se pone a “1” para indicar que el byte recibido en el registros SSPSR se ha perdido.

El registros SSPADD ,mantiene la dirección en el modo de 10 bits es necesario escribir el byte alto de la dirección (1111 0A9 A8 =) según el byte de menor peso (A7:A0).

UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

El módulo transmisor (USART) es otro de los dos puertos serie de los que dispone esta familia de microcontroladores. Los módulos de I/O. (USART también se conocen como *Serial Communications o Interface* o SCI). El USART puede configurarse como asíncrono full dúplex que puede comunique con los dispositivos periféricos como el terminales de CRT y los ordenadores personales, o puede configurarse como un sistema síncrono half duplex que puede comunicar con otros microcontroladores, con dispositivos periféricos como A/D o D/A circuitos integrados, EEPROMs serie etc.

El USART puede configurarse en los modos siguientes:

- · Asíncrono(Full duplex)
- · Síncrono- Master (half duplex)
- · Síncrono- Slave (half duplex)

Poniendo a "1" bit **SPEN** (**RCSTA** <7>), y los bits **TRISC** <7:6>, se configura **RC6/TX/CK** y **RC7 /RX/DT** como USART.

En modo síncrono se utilizan formatos estándar: un bit de *Start* y ocho bits o nueve bits de *Stop*, siendo el más común el de ocho bits de datos.

Registro de Estado y Control TXSTA (Transmisor)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSRC	TX9	TXEN	SYNC	----	BRGH	TRMT	TX9D
Bit 7							Bit 0

bit 7: **CSRC**: Bit de selección de reloj

Modo Asíncrono

Sin importancia

Modo Síncrono

1 = Modo master (reloj generado internamente para BRG (Generador de Baudios))

0 = Modo esclavo (reloj externo)

bit 6: **TX9**: Habilita transmisión de 9-8 bits

1 = Transmisión de 9 bits

0 = Transmisión de 8 bits

bit 5: **TXEN**: Bit de habilitación del transmisor

1 = Habilita transmisión

0 = Deshabilita transmisión

Nota: **SREN/CREN** anula **TXEN** en modo **SYNC**.

bit 4: **SYNC**: Bit de selección del modo USART

1 = Modo Síncrono

0 = Modo Asíncrono

bit 3: **Sin implementar**: En lectura es'0'

bit 2: **BRGH**: Bit de selección del rango de baudios alto

Modo Asíncrono

1 = Alta velocidad

0 = Baja velocidad

Modo Síncrono

No se usa este modo

bit 1: **TRMT**: Bit de estado del registro de desplazamiento del transmisor (TSR)

1 = TSR vacío

0 = TSR lleno

bit 0: **TX9D**: Noveno bit de datos de transmisión. Se puede emplear como bit de paridad.

Registro de estado y Control RCSTA (Receptor)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
SPEN	RX9	SREN	CREN	----	FERR	OERR	RX9D
Bit 7							Bit 0

bit 7: **SPEN**: Bit de habilitación del puerto serie

1 = Habilita puerto serie (configura patillas **RC7/RX/DT** y **RC6/TX/CK** para el puerto serie)

0 = Deshabilita puerto serie

bit 6: **RX9**: Habilita recepción de 9-8 bits

1 = Selecciona 9 bits de recepción

0 = Selecciona 8 bits de recepción

bit 5: **SREN**: Bit de habilitación de recepción única

Modo asíncrono

Este modo no se usa

Modo Síncrono master

1 = Habilita una recepción única

0 = Deshabilita una recepción única

Este bit se borra después de completar la recepción

Modo Síncrono esclavo

En este modo no se usa

bit 4: **CREN**: Bit de habilitación de recepción continua

Modo Asíncrono

1 = Habilita recepción continua

0 = Deshabilita recepción continua

Modo Síncrono

1 = Habilita recepción continua hasta que el bit **CREN** sea borrado (**CREN** anula **SREN**)

0 = Deshabilita la recepción continua

bit 3: Sin implementar: En lectura es '0'

bit 2: **FERR**: Bit de error de empaquetamiento

1 = Error de empaquetamiento (Puede actualizarse al leer el registro **RCREG** y recibir el siguiente byte)

0 = No hay error

bit 1: **OERR**: Bit de error de desbordamiento

1 = Error de desbordamiento (puede ser borrado al borrar **CREN**)

0 = No hay desbordamiento

bit 0: **RXGD**: Noveno bit de datos de recepción. Se puede emplear como bit de paridad

Generador de Baudios Rate (BRG)

Este generador sirve tanto para el modo síncrono como el asíncrono y consiste de un contador/divisor de frecuencia de 8 bits controlado por el registro **SPBRG (99H**). De tal manera que la frecuencia de transmisión se calcula de acuerdo a la siguiente tabla:

SYNC	BRGH=0(baja velocidad)	BRGH=1 (Alta Velocidad)
0 (Modo asíncrono)	Baud rate = $F_{OSC}/(64(X+1))$	Baud rate = $F_{OSC}/16(X+1))$
1 (Modo Síncrono)	Baud rate = $F_{OSC}/(4(X+1))$	--

BAUD RATES en Modo Asíncrono

BAUD RATE (K)	Fosc = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)	16 MHz KBAUD	% ERROR	SPBRG value (decimal)	10 MHz KBAUD	% ERROR	SPBRG value (decimal)	7.15909 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	Fosc = 5.0688 MHz KBAUD	% ERROR	SPBRG value (decimal)	4 MHz KBAUD	% ERROR	SPBRG value (decimal)	3.579545 MHz KBAUD	% ERROR	SPBRG value (decimal)	1 MHz KBAUD	% ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.615	+0.16	103	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.231	+0.16	51	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	76.923	+0.16	12	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	1000	+4.17	9	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	NA	-	-	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	100	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.906	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

BAUD RATES en Modo Asíncrono (BRG=0)

BAUD RATE (K)	Fosc = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)	16 MHz KBAUD	% ERROR	SPBRG value (decimal)	10 MHz KBAUD	% ERROR	SPBRG value (decimal)	7.15909 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	Fosc = 5.0688 MHz KBAUD	% ERROR	SPBRG value (decimal)	4 MHz KBAUD	% ERROR	SPBRG value (decimal)	3.579545 MHz KBAUD	% ERROR	SPBRG value (decimal)	1 MHz KBAUD	% ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.3005	-0.17	207	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.202	+1.67	51	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.404	+1.67	25	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	NA	-	-	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	NA	-	-	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	62.500	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	3.906	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

Debido a que el divisor es de 8 bits, no se puede tener cualquier velocidad de transmisión deseada, ya que X se deberá redondear al entero más cercano. En las dos tablas anteriores se muestran algunos valores de velocidad de transmisión (baud) estándar, el divisor necesario (X=SPBRG) bajo diferentes frecuencias Fosc y el error producido en porcentaje.

Ejemplo:

Cálculo de la proporción del error de velocidad de transmisión para las siguientes condiciones:

F_{OSC} = 16 MHZ

Velocidad de transmisión deseada = 9600 baudios

BRGH = 0

SYNC = 0

$$Velocidad\ de\ transmisión = \frac{F_{osc}}{(64*(X+1))}$$

$$9600 = \frac{16.000.00}{(64*(X+1))}$$

$$X = (25{,}042) = 25$$

$$Velocidad\ de\ transmisión\left(= \frac{16000000}{(64*(25+1))}\right) = 9615$$

$$error = \frac{9615 - 9600}{9600} = 0{,}16\%$$

Como puede apreciarse, el error no tiene ninguna importania para la mayoría de las aplicaciones, no obstante, se puede realizar los cálculos empleando el índice (**BRGH**=1) y sustituyendo la constante 64 por 16 (ver la tabla) y comprobar si el error se reduce.

Los datos de la patilla **RC7/RX/DT** son muestreados en tres tiempos para detectar y determinar mejor si el nivel presente en la patilla RX está a nivel alto o bajo. Si el bit **BRGH** (**TXSTA**<2>) está a cero, el muestreo se realiza en el séptimo, octavo, y noveno flanco de bajada de x1 6 de reloj.

Si por el contrario, el bit **BRGH** está a uno, el muestreo tiene lugar en el tercer flanco de reloj precedido de la segunda subida de reloj y de la primera bajada de x4 de reloj.

Registros asociados al Generador de Baud Rate

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 –010	0000 –010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
99h	SPGBRG	Registro generador de Baud Rate									

Leyenda.- x = desconocido u= sin cambios - = Sin implementar, se lee “0”

Modo Asíncrono USART

Como ya se ha dicho, en este modo de comunicaciones se emplea una configuración estándar de un bit de *Start*, ocho o nueve bits de datos, y un bit de *Stop*. Este modo de trabajo es detenido durante la función *Sleep*.

Aunque la USART tiene funciones independientes para la transmisión y recepción, los formatos de datos y de velocidad de transmisión son iguales para los dos casos. El generador de baudios produce siempre las señales de reloj x16 ó x64 (dependiendo de **BRGH**, **TXSTA**<2>) para el desplazamiento de los bits.

El bit de paridad no está soportado por el hardware, pero se puede implementar por software empleando para ello el noveno bit de datos.

Para realizar una transmisión de datos, el registro de desplazamiento **TSR** obtiene los datos del registro (buffer) de lectura/escritura **TXREG**. El **TSR** no carga el bit de Stop hasta que se hayan transmitido los datos cargados. Una vez que se haya enviado este bit, el registro será cargado con los nuevos datos (si están disponibles), en este caso el bit **TXIF** (**PIR1** <4>) se pone a ,uno si está habilitada la interrupción (**PIE1** <4>).

Los pasos necesarios para realizar una transmisión asíncrona son los siguientes:

- Inicializar el registro **SPBRG** con el valor apropiado según los cálculos realizados. Si se desea el margen alto se pondrá a uno el bit **BRGH**.
- Habilitar el puerto asíncrono borrando el bit **SYN** y poniendo a uno el **SPEN**.
- Habilitar la interrupción (si se desea) poniendo a uno el bit **TXIE**.
- Poner a cero o a uno el bit **TX9** (8 ó 9 bits).
- Habilitar la transmisión poniendo a uno el bit **TXEN**, el cual pondrá a uno el bit **TXIF.**
- Si se han seleccionado 9 bits se deber poner el noveno en el bit **TX9D**.
- Cargar el dato en el registro **TXREG** (comenzar la transmisión).

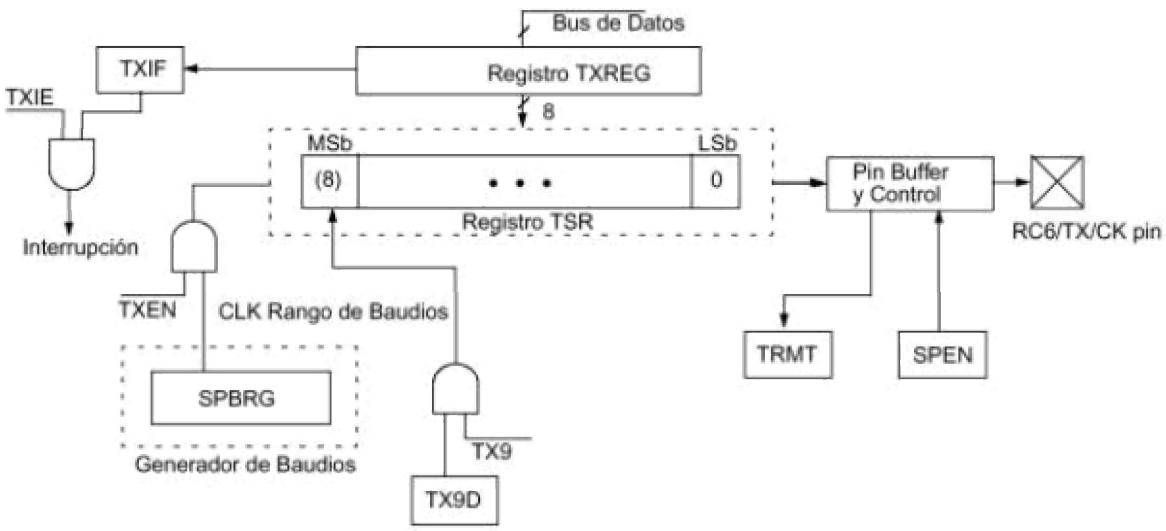
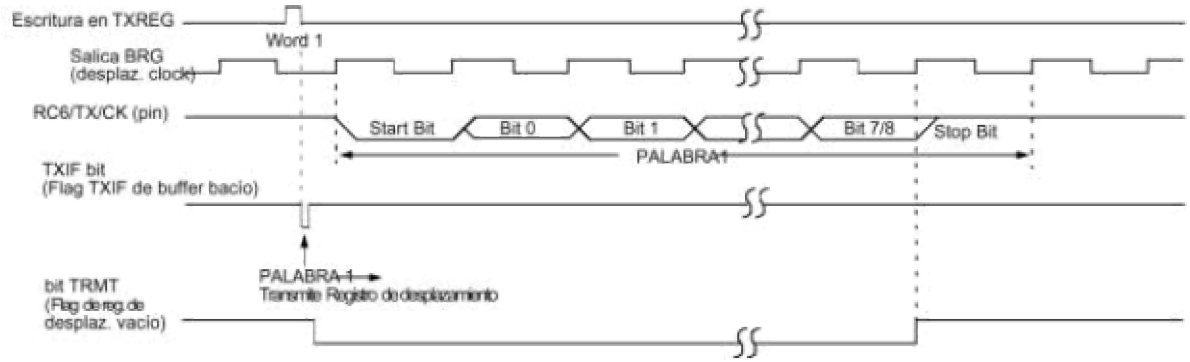
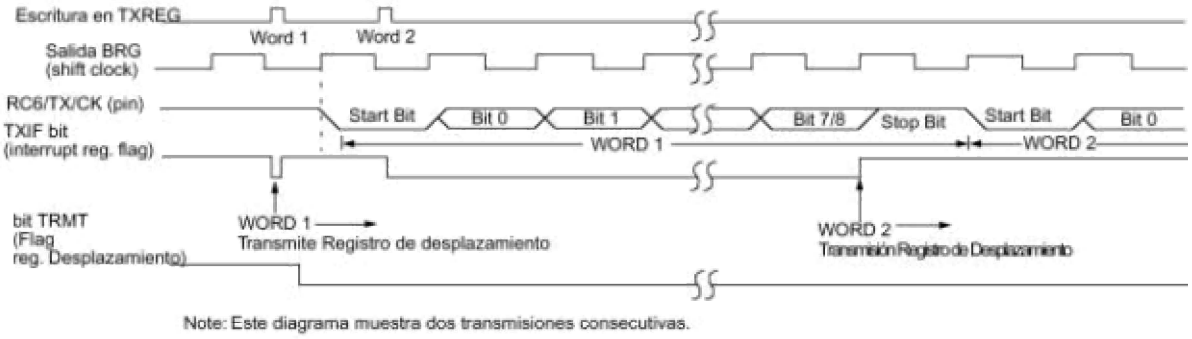


Diagrama de bloques del módulo USART configurado como transmisor



Transmisión Asíncrona



Transmisión Asíncrona Continua

En el modo de recepción, los datos son recibidos por la patilla **RC7/RX/DT** y se introducen en serie en el registro **RSR**; después de recibir el bit de Stop los datos acumulados en este registro se cargan en el de datos **RCREG** (si está vacío). Cuando la transferencia se ha completado, el bit **RCIF** (**PIR1** <5>) sube a uno. La interrupción puede habilitarse o deshabilitarse colocando a

uno o borrando, respectivamente, el bit **RCIE** (**PIE1** <5>). El bit **RCIF** se borra cuando el registro **RCREG** se ha leído (se "vacía" el registro). Para configurar el puerto como receptor hay que seguir los siguientes pasos:

- Inicializar el registro **SPBRG** con el valor apropiado según los cálculos realizados. Si se desea el margen alto se pondrá a uno el bit **BRGH**.
- Habilitar el puerto asíncrono borrando el bit **SYN** y poniendo a uno el **SPEN**.
- Habilitar la interrupción (si se desea) poniendo a uno el bit **RCIE**.
- Poner a uno el bit **RX9** para habilitar 9 bits de recepción.
- Habilitar la recepción poniendo a uno el bit **CREN**.
- El flag **RCIF** subirá a uno cuando se complete la recepción.
- Se ocasionar una interrupción si está habilitado el bit **RCIE**.
- Leer el registro **RCSTA**, cargar el noveno bit (si está habilitado) y comprobar si ha ocurrido un error durante la recepción.
- Leer el octavo bit recibido y dar lectura al registro **RCREG** para conocer si el dispositivo se ha direccionado.
- Si ha ocurrido un error borrarlo poniendo a cero el bit **CREN**.
- Si el dispositivo ha sido direccionado, borrar el bit **ADDEN** para permitir que los datos puedan leerse del buffer receptor

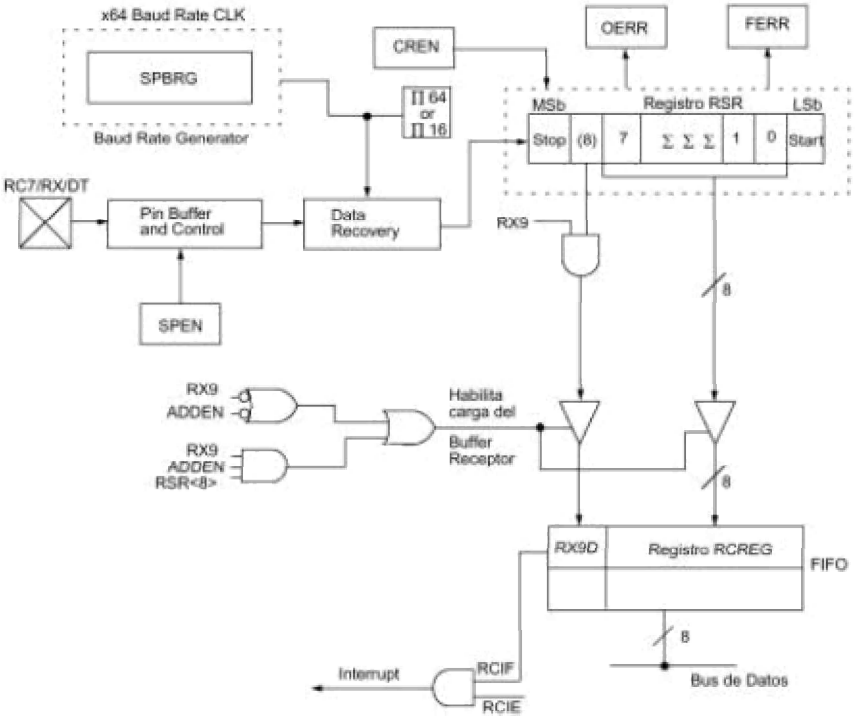
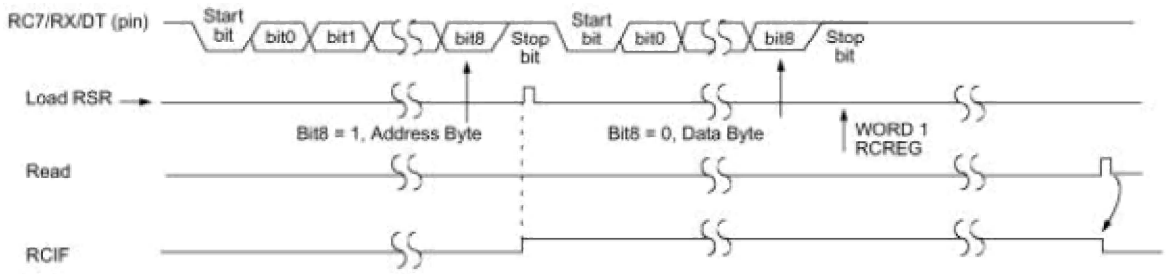


Diagrama de Bloques del Módulo USART configurado en modo Receptor



Nota.- Esto cronometrando el diagrama muestra un byte del datos seguido por un byte de dirección. El byte de los datos no se lee en el RCREG (reciba el pulidor) porque ADDEN = 1.

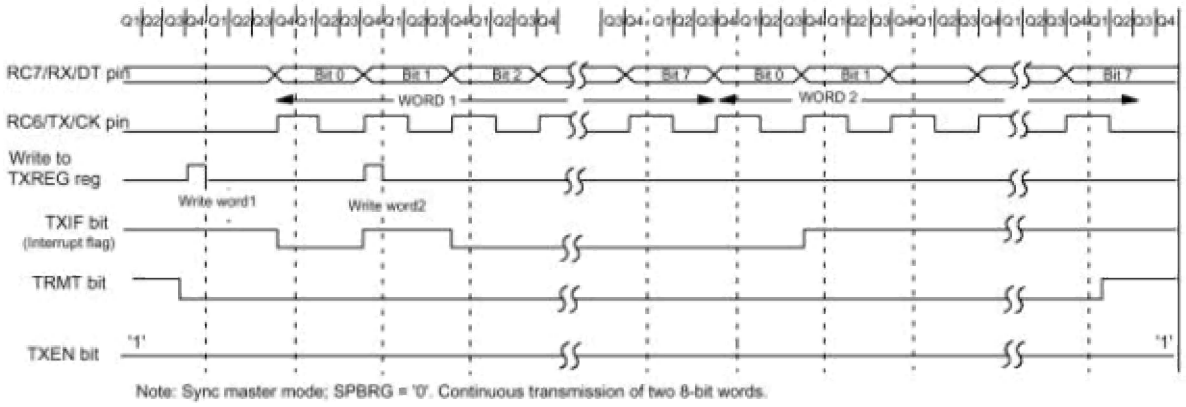
Recepción Asíncrona con Dirección y Primer Byte

Modo síncrono

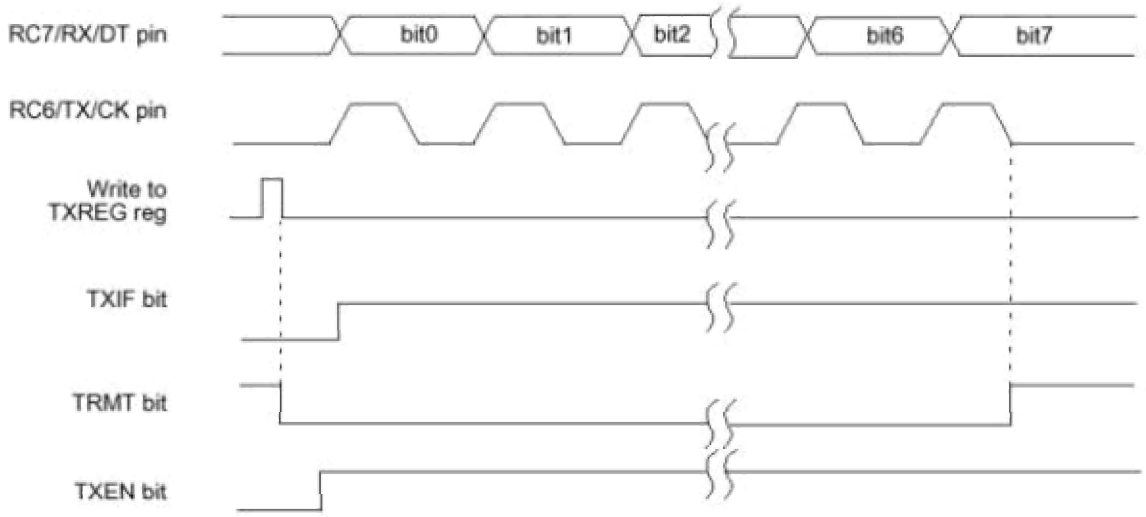
Este modo de trabajo tiene dos posibles formas de configuración, como *master* y como esclavo. En el modo *master* la transmisión y recepción no tienen lugar al mismo tiempo, cuando se transmite un dato, la recepción es inhibida y viceversa. En la transmisión el registro de desplazamiento, **TSR** (*Transmit Shift Register*), obtiene los datos del registro de lectura/escritura **TXREG**, pero éste no se carga hasta que el último bit de la anterior carga no haya sido transmitido; cuando esto suceda el **TSR** se vuelve a cargar con el nuevo dato del registro **TXREG** (si está disponible) éste se vacía y se genera una interrupción (si está habilitada por el bit **TXIE**, **PIE1**<4>. La transmisión se habilita por el bit **TXEN** del registro **TXSTA**<5>, si se borra durante una transmisión, ésta será abortada y reseteado el transmisor.

Para iniciar una transmisión es necesario seguir los siguientes pasos:

- Inicializar el registro **SPBRG** con el valor apropiado calculado.
- Habilitar el puerto master asíncrono colocando a uno los bits **SYN**, **SPEN** y **CSRC**.
- Si se desea emplear la interrupción, habilitarla poniendo a uno el bit **TXIE**.
- Si se desea una transmisión de nueve bits, el noveno se cargar en el bit **TX9**.
- Habilitar la transmisión poniendo a uno el bit **TXEN**.
- Si se ha habilitado la transmisión de 9 bits, el noveno se cargar en el registro **TX9D**.
- La transmisión se iniciar al cargar los datos en el registro TXREG.



Transmisión Síncrona



Sincronismo de transmisión

Para seleccionar el modo receptor master hay que poner a uno el bit SREN (RCSTA<5> o habilitar el CREN (RCSTA<4>). El dato es tomado de la patilla RC7/WDT por cada flanco de caída del reloj , si el bit SREN está a uno, sólo se recibir una palabra, pero si se borra el bit CREN la recepción ser continua. Después del último impulso de reloj, el dato recibido en el registro de desplazamiento RSR será transferido al RCREG (si está vacío). Una vez haya terminado la recepción se originar una interrupción si ésta está habilitada, RCIE (PIE1 <5>).

Los pasos a seguir para iniciar la recepción en modo master son las siguientes:

- Inicializar el registro SPBRG con el valor apropiado de baudios.
- Habilitar el puerto master asíncrono colocando a uno los bits SYN, SPEN y CSRC.
- Asegurarse de que los bits CREN y SREN están borrados.

- Si se desea emplear la interrupción habilitarla poniendo a uno el bit RCIE.
- Si se desean nueve bits poner a uno el bit RX9.
- Si sólo se desea una recepción única, poner a uno el bit SREN; para recepción continua poner a uno el bit CREN.
- El flag de interrupción RCIF se pone a uno cuando la recepción haya finalizado si está habilitada por el bit RCIE.
- Leer el registro RCSTA y cargar el noveno bit (si está habilitado) y comprobar si ha ocurrido un error durante la recepción.
- Leer el octavo bit recibido y dar lectura al registro RCREG.
- Si ha ocurrido un error borrarlo poniendo a cero el bit CREN.

Registros de Sincronización en modo master en transmisión

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMREIF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
1Ah	RCGEG	Registro Receptor USAR								0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 –010	0000 –010
99h	SPGBRG	Registro generador de Baud Rate									

Leyenda.- x = desconocido u= sin cambios - = Sin implementar, se lee “0”

Modo esclavo síncrono

Este modo de trabajo difiere del modo master, pues la señal de reloj de desplazamientos es proporcionada externamente por la patilla **RC6/TX/CK** en lugar de ser generada internamente, esto permite que se puedan transferir o recibir datos cuando el dispositivo está en el modo *Sleep*. Las operaciones de transmisión son idénticas al modo *master*, excepto en el caso del modo *Sleep*.

Si se escriben dos palabras en el registro **TXREG** y después se ejecuta la instrucción *Sleep*, ocurrir lo siguiente:

- La primera palabra será inmediatamente transferida al registro **TSR** y transmitida.
- La segunda palabra permanecer en el registro **TXREG**.
- El bit **TXIF** no se pondrá a uno.
- Cuando la primera palabra haya salido del registro **TSR**, la segunda palabra, que permanece en el registro **TXREG**, se enviar al **TSR** y el bit bandera **TXIF** subir a uno.
- Si está habilitado el bit **TXIF** la interrupción despertar al microcontrolador de *Sleep* y el programa saltar a la dirección del vector de interrupciones (0004 hex.).

Registros de Sincronización en modo master en recepción

Dirección	Nombre	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor en POR,BOR	Valor en el resto de Reset
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMREIF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000X	0000 000X
1Ah	RCGEG	Registro Receptor USAR								0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	---	BRGH	TRMT	TX9D	0000 –010	0000 –010
99h	SPGBRG	Registro generador de Baud Rate									

Leyenda.- x = desconocido u= sin cambios - = Sin implementar, se lee “0”

CARACTERÍSTICAS ESPECIALES DE LA CPU

Estos dispositivos están organizados para aumentar al máximo sus características de fiabilidad del sistema, minimizar el costo de elementos externos, modo de trabajo en bajo consumo así como protección de código. Los modos de trabajo de que dispone son los siguientes

- Selección de oscilador
- Reset:
- Power-on Reset (POP)
- Power-Timer (PWRT)
- El oscilador Salida-a el Cronómetro (OST)
- Interrupciones
- Watchdog
- SLEEP
- Localizador ID
- Programación serie en circuito
- Programación en bajo voltaje
- Degugger en circuito

Estos dispositivos tienen un Timer *Watchdog*, que puede ser habilitado a través de los bits de configuración. Dispone de un oscilador RC interno muy fiable. Dispone de dos temporizadores de *Power-up*. Uno asociado al oscilador *Start-up* (OST), pensado para funcionar desde que se genera un reset hasta que el oscilador de cristal sea estable. El otro es un temporizador asociado al *Power-up* Timer (PWRT) que proporciona un retardo de 72 ms (nominal) solo con un *Power-up.*

Esta diseñado para esperar que el resto de los periféricos del circuito se estabilicen en su funcionamiento.

El modo SLEEP está diseño para tener un consumo muy bajo de la fuente de alimentación. El usuario puede salir del modo SLEEP a través de un reset externo, un desbordamiento del Watchdog o al generarse una interrupción. También dispone de arias opciones del oscilador para diversas aplicaciones. El

oscilador RC ahorra costes al sistema. Mientras que la opción LP ahorra consumo. La configuración se realiza en el Registro de Configuración.

Bits de Configuración

Los bits de la configuración pueden ponerse a 0 o a 1 para seleccionar las distintas configuraciones del dispositivo. Estos bits se graban en la posición de memoria situada en la dirección 2007h. Esta dirección de memoria está más allá de la memoria de programa de usuario. De hecho, pertenece al espacio de memoria de test/configuración (2000h - 3FFFh), a la que solo puede accederse durante la programación.

PALABRA DE CONFIGURACIÓN

CP1	CP0	BKBUG	- - -	WRT	CPD	LVP	BODEN	CP1	CP0	#PWRTE	WDTE	FOSC1	FSC0
bit13													bit0

bit 13-12:

bit 5-4: **CP1:CP0**: Bits de protección del código de la memoria Flash de programa[(2)]

11 = Sin protección de código

10 = 1F00h a 1FFFh código protegido (PIC16F877, 876)

10 = 0F00h a 0FFFh código protegido (PIC16F874, 873)

01 = 1000h a 1FFFh código protegido (PIC16F877, 876)

01 = 0800h a 0FFFh código protegido (PIC16F874, 873)

00 = 0000h a lFFFh código protegido (PIC16F877, 876)

00 = 0000h a 0FFFh código protegido (PIC16F8741873)

bit 11: **DEBUG**: Modo *In-Circuit Debugger*

1 = Deshabilitado modo *In-Circuit Debugger*, RB6 y, RB7 serán patillas de entrda/salida

0 = Habilitado modo *In-Circuit Debugger*, RB6 y RB7 dedicadas al *debugger*

bit 10: Sin implementar: En lectura es "1"

bit 9: **WRT**: Habilita escritura en memoria Flash de programa

1 = Sin protección, se puede escribir con el control **ECON**.

0 = Sin protección, no se puede escribir con el control **EECON**

bit 8: **CPD**: Protección de los datos de la memoria **EE**

1 = Sin protección

0 = Datos de la **EEPRON** protegidos

bit 7: **LVP**: Bit de habilitación de programación a bajo voltaje

1 = Patillas **RB3/PGM** tienen funciones **PGM** , bajo voltaje de programación habilitado

0 = **RB3** es digital I/O, **HV** habilitado, **#MCLR** se emplea para programación

bit 6: **BODEN**: Bit de habilitación **Brown-out Reset** (1)

1 = **BOR** habilitado

0 = **BOR** deshabilitado

bit 3: **PWRTE**: Bit de habilitación deL temporizador *Power-up* (1)

1 = **PWRT** deshabilitado

0 = **PWRT** habilitado

bit 2: **WDTE**: Bit de habilitación *Watchdog*

1 = **WDT** habilitado

0 = **WDT** deshabilitado

bit 1-0: **FOSC1:FOSC0**: Bit de selección del oscilador

11 = Oscilador RC

10 = Oscilador HS

01 = Oscilador XT

00 = Oscilador LP

Nota (1)= Al habilitar el Brown-out Reset se habilita automáticamente el

temporizador *Power-up* (PWRT) .Habilitando el Power-up Timer se habilita al mismo tiempo el *Brown-out* Reset

(2): Los dos pares de bit **CP1:CP0** tienen que tener el mismo valor para habilitar el esquema de la protección del código listado.

Configuración del Oscilador

Estos modelos de microcontroladores no se apartan de la línea seguida por las familias anteriores, también disponen de los cuatro modos de configuración del oscilador:

- LP Oscilador de cristal de cuarzo o resonador cerámico de baja potencia (*Low Power Cristal*). Se trata de un oscilador de bajo consumo con un cristal o resonador para trabajar con frecuencias comprendidas entre 32 y 200 KHz.
- XT (*Cristal/Resonador*) Oscilador o resonador cerámico. Se trata de un oscilador estándar que permite una frecuencia de reloj comprendida entre 100 KHz y 4 MHz.
- HS (*High Spedd Cristal/Resonador*) Oscilador de cristal o resonador de alta velocidad. Es un oscilador de una frecuencia comprendida ente 4MHz y 20 MHz.
- RC (*Resistor/Capacitor*) . Es un oscilador de bajo coste, formado por una resistencia y un condensador

En los modos XT, LP o HS, el cristal o resonador se conecta entre las patillas **OSC1/CLKIN** y **OSC2/CLKOUT**, pero también permite conectar un oscilador externo a la patilla **OSC1/CLKIN**.

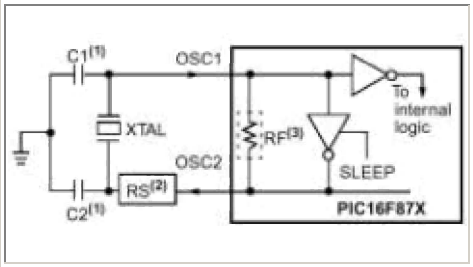
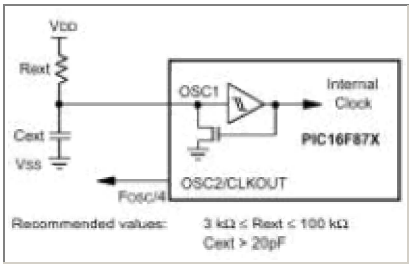
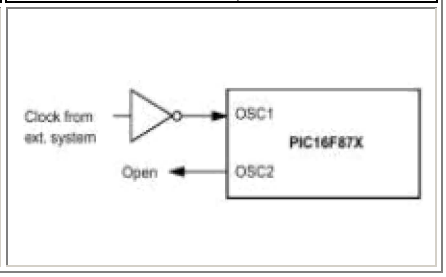
Conexión del Cristal o Resonador en modo (HS, XT o LP)	Oscilador en modo RC	Conexión de Oscilador externo en modo (HS, XT o LP)
		

Figura 56.- Modos de configuración del oscilador principal del sistema

Resonadores Cerámicos	Condensadores para cristales

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	68 - 100 pF 15 - 68 pF 15 - 68 pF	68 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	10 - 68 pF 10 - 22 pF	10 - 68 pF 10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			

Resonators Used:		
455 kHz	Panasonic EFO-A455K04B	± 0.3%
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%
All resonators used did not have built-in capacitors.		

PRELIMINARY

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			

Crystals Used		
32 kHz	Epson C-001R32.768K-A	± 20 PPM
200 kHz	STD XTL 200.000KHz	± 20 PPM
1 MHz	ECS ECS-10-13-1	± 50 PPM
4 MHz	ECS ECS-40-20-1	± 50 PPM
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM

Valores de condensadores necesarios a instalar entre la masa y las patillas de los cristales y/o resonadores

EL RESET

En estos circuitos hay seis formas de que se produzca un reset:

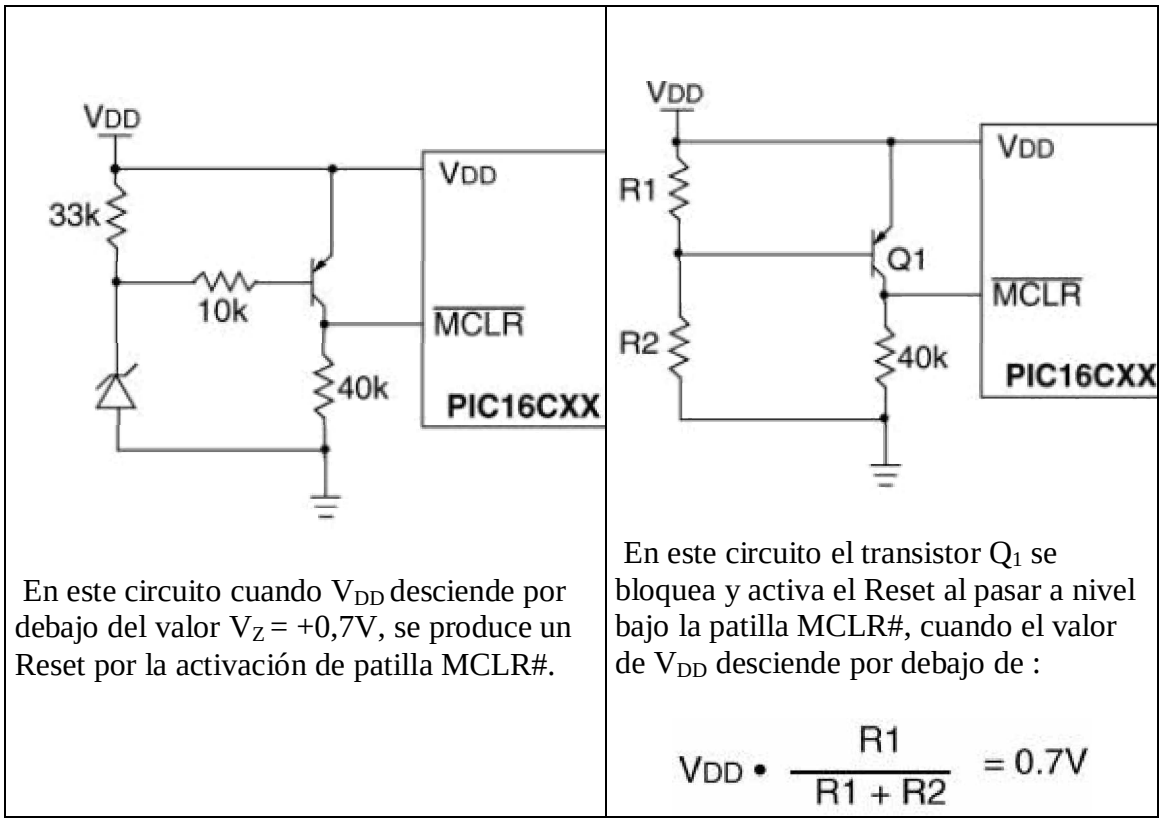
- Power-On Reset (POR)
- MCLR reset durante una operación normal
- MCLR reset durante Sleep
- WDT reset durante una operación normal
- WDT por despertar de Sleep
- Brown-Out Reset (BOR)

Con la lectura de los bits **#TO** y **#PD** del registro de **STATUS** y los bits **POR** y **BOR** del registro **PCON** (83h) sabremos en todo momento quién y cómo se produjo un reset.

Llevando a cero la patilla #**MCLR** se puede generar un reset. Si no se desea ningún sistema de reset externo, esta patilla se puede conectar directamente a la alimentación o a través de una resistencia.

El pulso de inicialización **POR** (*Power-On Reset*) es generado internamente cuando la tensión de alimentación se encuentra entre los limites de 1,2 V y 1,7 V, es un método cómodo de originar un reset al conectar la alimentación, ya que no requiere de ningún componente adicional pero no siempre es el método ideal para realizarlo, en muchas ocasiones se necesita que éste se

produzca a una tensión superior. En estos casos se pueden utilizar los circuitos y cálculos que muestra la Figura 58.



- Dos circuitos externos de reset y formulas para calcular la tensión a la que tendrá lugar

Cuando se conecta la tensión de alimentación a los circuitos, sobre todo cuando la fuente de alimentación tiene grandes condensadores de filtrado, la salida de tensión continua tarda en crecer desde cero hasta alcanzar el máximo valor; una vez se hayan sobrepasado los 1, 7 V y antes de llegar a la máxima tensión, en estos pocos milisegundos el microcontrolador tiene tiempo para ejecutar algunas miles de instrucciones, pero antes de que comiencen a ser ejecutadas hay que estar seguro de que el resto de los componentes que constituyen el diseño se han terminado de inicializar. Es aquí donde se aprovechan las ventajas del *Power-On Reset* y si se ha habilitado, las del temporizador *Power-Up Timer* (**PWRT**), bit configurable en la programación del microcontrolador) que añade 72 milisegundos adicionales al impulso interno de reset. Para la mayoría de los casos con el **POR** y el tiempo añadido por el **PWRT** será más que suficiente para que, tanto la tensión de alimentación como el resto de los circuitos, se hayan estabilizado. No obstante, si se desea que el microcontrolador permanezca inactivo hasta que la tensión de alimentación esté, muy próxima a su máximo valor, no queda más remedio

que, además de habilitar el **PWRT**, se instale un circuito externo de reset con tensión de disparo ajustable, tres resistencias y un transistor así lo garantizan (ver el esquema de la Figura 58).

Los circuitos internos de reset también tienen un circuito de control (OST, Oscillator Start-up Timer) que añade un retardo de 1.024 ciclos del reloj externo después de los 72 ms. proporcionados por el **PWRT**. Este retardo asegura que el oscilador externo ha terminado de estabilizarse antes de que la CPU comience a ejecutar instrucciones.

Otro de los modos de reset es el **Brown- Out Reset** que tiene un comportamiento muy similar al **POR**, sólo que los márgenes de tensión son diferentes, en este caso se sitúan entre 3,8 V y 4,2 V, en realidad lo que trata de detectar, es que la tensión de alimentación no tenga fallos o pequeñas caídas de voltaje durante periodos de tiempo muy cortos.

No sustituye a los circuitos de reset es una forma muy eficaz de controlar los pequeños fallos que se pueden producir en la alimentación, bien por cortes rápidos de la tensión de entrada, o bien porque se ha producido un pico de corriente muy elevado y la fuente de alimentación no ha sido capaz de proporcionarlo y ha caído su tensión.

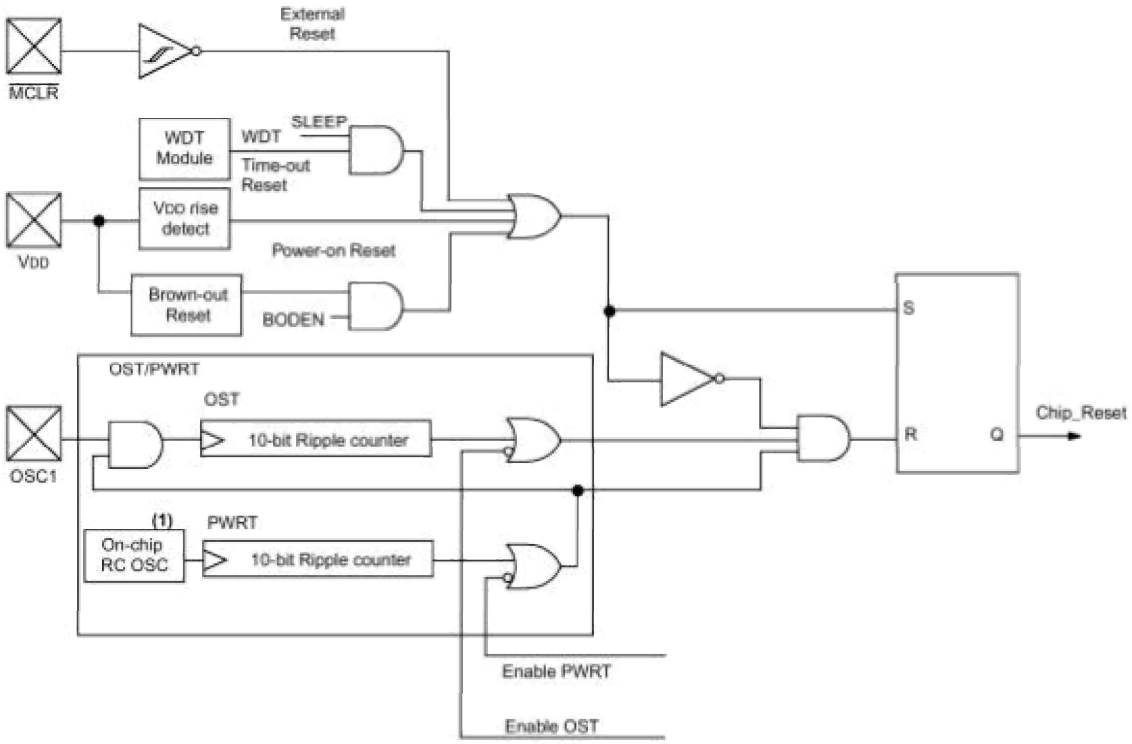


Diagrama de bloques del circuito de reset dentro del chip

En el diagrama de bloques de la Figura 59 se pueden ver, los circuitos PWRT y OST, el primero está formado por un contador de 10 bits que se encarga de contar 1.024 impulsos de reloj interno formado por una red RC (este oscilador no tiene nada que ver con el oscilador principal). Este es el circuito encargado de proporcionar los 72 milisegundos de retardo; una vez hayan transcurrido se habilita un segundo contador (OST) también de 10 bits, que se encarga de contar 1.024 impulsos del oscilador externo; suficientes ciclos para que al finalizar la cuenta el oscilador ya esté totalmente estabilizado.

Las Interrupciones

Esta familia de microcontroladores dispone de 14 fuentes de interrupción, de las cuales algunas de ellas se habilitan por medio del registro de control **INTCON**, otras, sin embargo, como la del convertidor A/D se encuentra en el registro **PIE1**, en este mismo registro está el bit del **TIMER1** (**TMRIE** ,**PIE1**<0>) entre otros. En el registro **INTCON** sólo se encuentran tres controles de interrupciones. Interrupciones ocasionadas en la patilla **RBO/INT**, las ocasionadas por cambios de estado en las patillas **RB7:RB4** y las que tienen lugar por el desbordamiento del temporizador **TMR0**, el resto de los controles de periféricos y funciones especiales se encuentran en los registros **PIR1**, **PIR2**.

El bit **GIE** (**INTCON**<7> habilita las interrupciones no enmascaradas, por el contrario, cuando está a cero, deshabilita todas las interrupciones con independencia de cómo se encuentren los bits individuales de interrupciones.

Cuando tiene lugar una interrupción, el valor del Contador de Programa (**PC**) se envía al *Stack* y se carga con 0004 hex (vector de interrupción). El microcontrolador comenzar a ejecutar las instrucciones de la subrutina de interrupciones y cuando se encuentre una instrucción **RETFILE** dará por finalizada la subrutina volviendo a la siguiente dirección de la última instrucción ejecutada antes de producirse la interrupción.

Hay que tener en cuenta que cuando se produce un **Reset** se borra el bit **GIE**, por lo que se deshabilitan todas las interrupciones, aunque se habiliten los bits individuales.

Al atender una interrupción, el bit **GIE** se borra automáticamente con lo que se deshabilita la posibilidad de futuras interrupciones, la dirección de retorno se envía al *Stack* y el Contador de Programa se carga con la dirección 0004 hex. (vector de interrupción). En esta dirección comenzará la subrutina de servicio de interrupciones. Para determinar qué periférico o función ha ocasionado la interrupción se deberán leer los bits de control de las correspondientes interrupciones

Interrupción de TMR0

Si el temporizador **TMR0** se ha desbordado, el *flag* de *overflow* (**TOIF**) del registro **INTCON** se habrá puesto a uno. Es evidente que previamente se habrán borrado todos los *flags* que pudieran ocasionar una interrupción, así se evitarán interrupciones recursivas.

Interrupción externa por RB0/INT

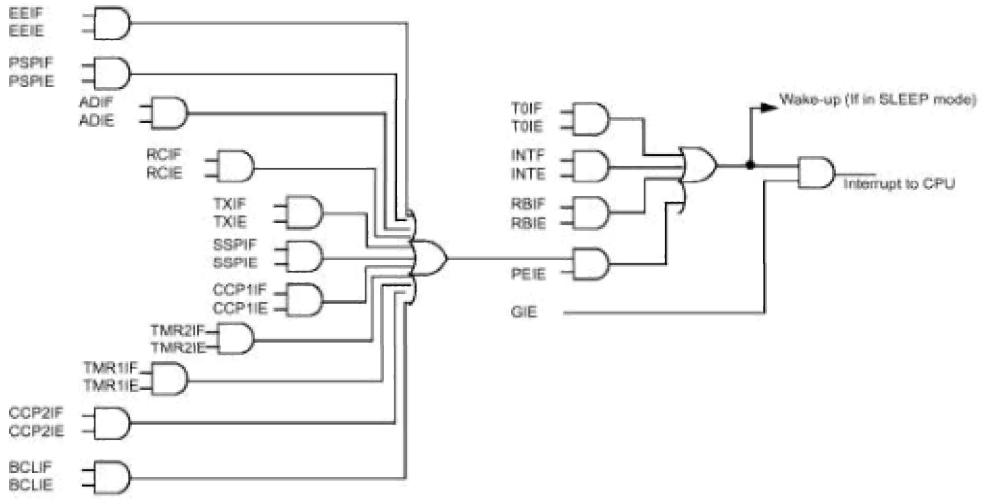
Las transiciones de bajo a alto, o viceversa, de la patilla **RB0/INT** puede ocasionar una interrupción, el flanco que la producirá se elige con el bit **INTEDG** (**OPTION_REG**<6>).

Cuando este bit se pone a uno, la interrupción tendrá lugar en la transición de cero a uno, por el contrario, el flanco de bajada ocasionará la interrupción cuando el bit se coloque a cero. En cuanto tenga lugar en la patilla **RB0/INT** el flanco elegido, el bit **INTF** (**INTCON**<1>) se pondrá a uno; el estado de los bits de interrupciones globales (**GIE**) e interrupción externa (**INTE**) determina si se atenderá o no la interrupción. Si están habilitadas y el microcontrolador se encuentra en reposo (*Sleep*) éste despertará.

Interrupción por cambio de estado en RB4:RB7

Además de la interrupción externa descrita, estos microcontroladores se disponen de otro modo de ocasionar una interrupción externa; cualquier cambio de estado que se produzca en las patillas **RB7:RB4** también la ocasionará , para habilitarla, hay que poner a uno el flag de interrupciones **RBIE** (**INTCON**<3>). Cualquier cambio de estado que haya en estas patillas harán que el bit **RBIF** (**INTCON**<0>) se ponga a uno y se produzca la interrupción.

Para que el sistema pueda ocasionar una interrupción por cambios de estado en las patillas **RB7:RB4** es necesario realizar algunas actuaciones previas que no son necesarias en otro tipo de interrupciones ya que el estado de estas cuatro patillas es comparado continuamente, en la fase **Q1** del reloj, con el estado previo de los *latch* del puerto. Por consiguiente, antes de habilitar esta interrupción habrá que leer o escribir en él para que su contenido se quede almacenado en los *latch*; solamente los bits configurados como entradas se comparan y si son diferentes a los previamente almacenados pueden ocasionar una interrupción.



The following table shows which devices have which interrupts.

Device	T0IF	INTF	RBIF	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	EEIF	BCLIF	CCP2IF
PIC16F876/873	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PIC16F877/874	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

- Diagrama de interrupciones lógicas de los microcontroladores de la familia PIC16F87X. Se muestran las diferencias entre los dos modelos de 28 y 40 pines.

Interrup.	Descripción	Registro	Interrupción	Descripción	Registro
TOIF	Desbordamiento del TMR0	**INTCON**	**TXIF**	Transmisión USAR	**PIR1**
INTF	Externa	**INTCON**	**SSPIF**	Puerto Serie Síncrono	**PIR1**
RBIF	Por cambios en el PortB	**INTCON**	**CCP1IF**	Capturador/Comparador 1	**PIR1**
PSPIF	Lectura/Escritura puerto paralelo	**PIR1**	**TMR2IF**	Temporizador TMR2	**PIR1**
ADIF	Convertidor A/D	**PIR1**	**TMR1IF**	Temporizador TMR1	**PIR1**
RCIF	Recepción USAR	**PIR1**	**CCP2IF**	Capturador/Comparador 2	**PIR2**

Salvar el entorno durante una interrupción

Estos microcontroladores disponen de ocho niveles de Stack y cuando se produce una interrupción sólo se salva en él la dirección de retorno (**PC**).

Cualquier operación que se realice en la subrutina de interrupciones podrá destruir, o modificar, el contenido de los datos almacenados en ciertos registros y que podrían ser necesarios cuando se retorne de la, subrutina de servicio de interrupciones. Si esto fuera así, se deberán guardar los datos de los registros cuyo contenido sea de interés y que vayan a ser modificados a lo largo de la subrutina de servicio de interrupciones.

Una vez finalizada la subrutina, se deberán recuperar para que queden en su estado original. Un ejemplo para salvar los registros de estado, **W** y **PCLATH** en memoria puede ser el siguiente:

```
movwf W_TEMP          ;Copia W en el registro TEMP, puede ser banco 0 o 1

swapf   STATUS,W      ;Intercambia los nibles del registro de STATUS

                      ; y los guarda en  W

clrf    STATUS        ;Selecciona banco0, independientemente del banco actual,

                      ;borra IRP,RP1,RP0

movwf STATUS_TEMP     ;salva el registro de STATUS en  STATUS_TEMP

movf    PCLATH,W      ;Solo es necesario si se emplean las páginas 1,2 y/o 3

movwf PCLATH_TEMP     ;Salva  PCLATH en W
```

```
clrf     PCLATH              ;Página cero, independientemente de la página actual

mov      FSR, W              ;Copia FSR en W

movwf    FSR_TEMP            ;Copia el FSR de W a FSR_TEMPORAL

:

:(Ejecución de instrucciones)

:

movf     PCLATH_TEMP, W      ;Recupera PCLATH en W

movwf PCLATH                 ; W lo lleva  PCLATH

swapf    STATUS_TEMP,W       ;Intercambia los nibles del STATUS_TEMP y los deja en W

                             ; (pone el banco original)

movwf STATUS                 ;Mueve W al registro de STATUS

swapf    W_TEMP,F            ;Intercambia los nibles de W_TEMP

swapf    W_TEMP,W            ;Intercambia los nibbles de W_TEMP y deja el resultado en W
```

El programa está dividido en dos partes, la primera guarda el contenido de los registros y se ejecutan las instrucciones propias de la subrutina de servicio de interrupciones (comprobar quién ocasionó la interrupción, etc.) y la segunda recupera de la memoria los registros previamente almacenados. La instrucción "**Retfie**" llevará el flujo del programa a la dirección siguiente a la última instrucción ejecutada (recupera el **PC** del Stack) con los registros ya actualizados.

EL TEMPORIZADOR TMR0 Y EL WDT

El temporizador del Watchdog tiene su propio reloj de funcionamiento por lo que no depende del oscilador principal del sistema, esto permite que, cuando el WDT está, habilitado y el temporizador se desborde por no borrarlo a tiempo (aún estando en reposo), pueda ocasionar una detención en el flujo de ejecución de instrucciones del programa (Watchdog Timer Reset). Es evidente que esto no puede considerarse una interrupción como las que puede ocasionar el temporizador TMR0, por ejemplo, pero hay que tenerlo en cuenta ya que al igual que otras interrupciones también puede despertar al microcontrolador del estado de reposo (Sleep) y que con una simple lectura de los bits correspondientes sabremos quién lo despertó.

Cuando enviamos al microcontrolador a Sleep existen varias formas de despertarlo. Uno de los eventos lo acabamos de comentar: si al programar el

microcontrolador se habilita el WDT, el despertar tendrá lugar si se excede del tiempo prefijado sin borrarlo. Las actuaciones sobre la patilla de reset externo #MCLR, las interrupciones externas, RB0/INT y los cambios de estado de las patillas RB7:RB0, también ocasionan que el circuito salga de Sleep, pero también algunos periféricos que ocasionan interrupciones pueden despertarlo. Estas son las siguientes:

- Lectura o escritura del PSP
- El TMR1 en modo contador asíncrono.
- Captura del CCP
- Eventos especiales de disparo en el TIMER1 cuando se usa con reloj externo.
- El bit Start/Stop (SSP).
- El SSP transmisión o recepción en modo esclavo (SP1/I^2C).
- Rx o Tx de la USART (en modo esclavo).
- El convertidor A/D cuando la fuente de su reloj es RC interna.
- Cuando una operación de escritura en la EEPROM finalice.

De esto es fácil deducir que los periféricos que emplean la señal del reloj principal del sistema, no pueden despertar a la CPU del estado de reposo.

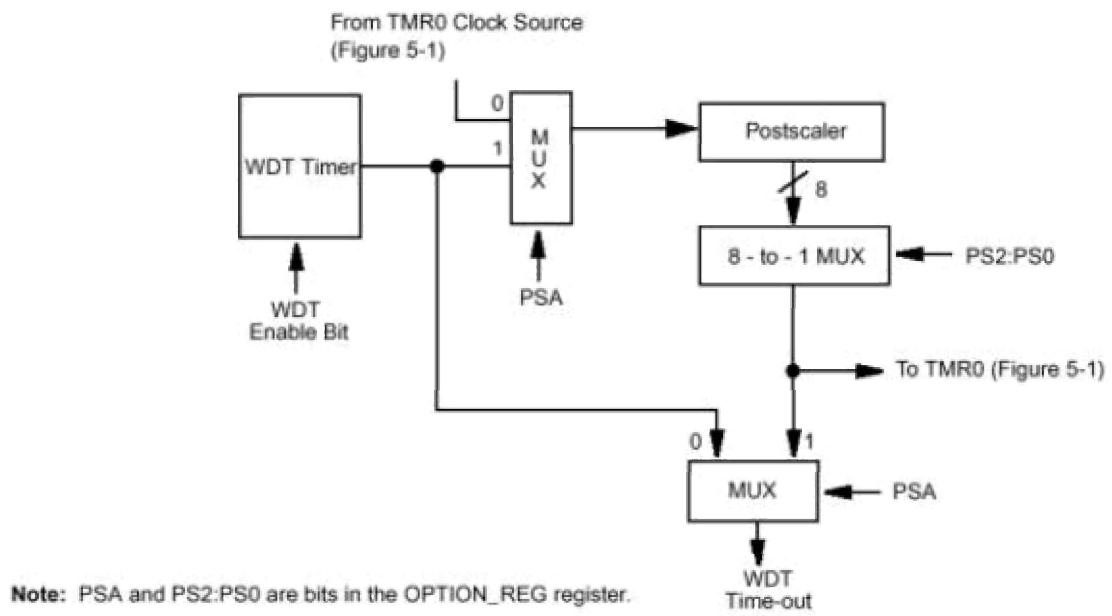


Diagrama de bloques del Timer Watchdog

Registros asociados al Warchdog

Dirección	Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
2007	Bit de conf	(1)	BODEN(1)	CP1	CP0	PWETE(1)	WDTE	FOSC1	FOSC0
81h,181h	OPTION_REG	#RBPU	INTEDG	T0SC	T0SE	PSA	PS2	PS1	PS0

La leyenda: Las células sombreadas no son usadas por el Watchdog

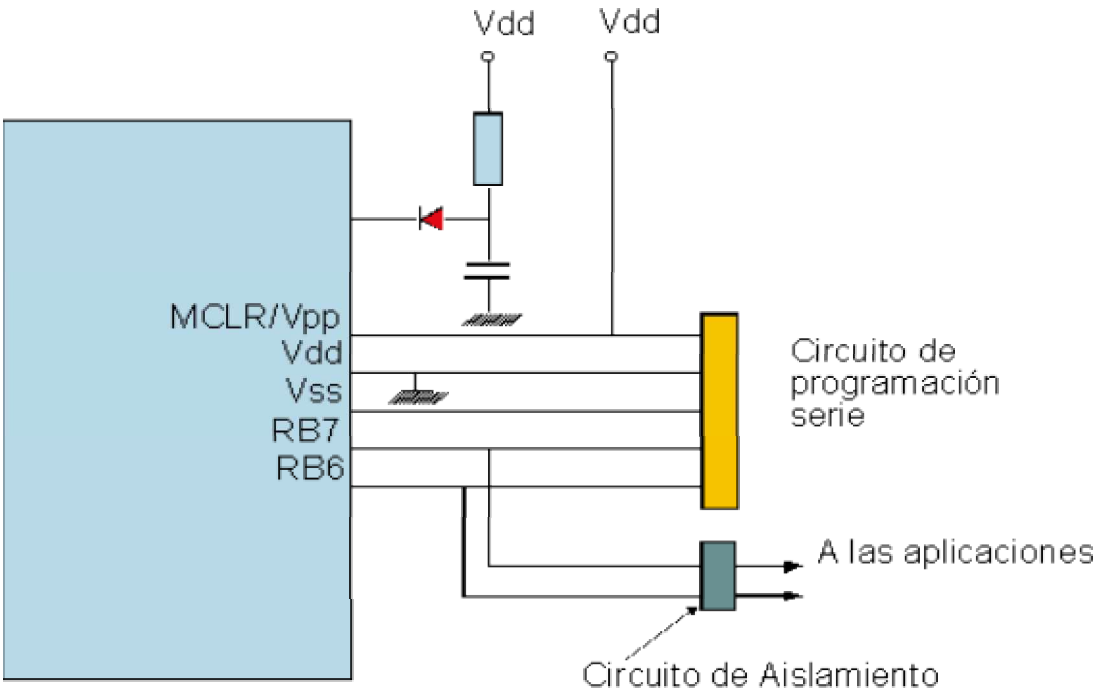
Note 1: Vea Registro para el funcionamiento de estos bits.

Programación Serie de los PIC16F87X

Programar los microcontroladores sin tener que desmontarlos de los circuitos donde están ubicados es una particularidad que tienen casi todos los dispositivos, pero si, además, su memoria de programa es del tipo Flash les confiere aún mayor flexibilidad, pues se pueden realizar cambios en los programas y volver a reprogramar el microcontrolador sin la necesidad de desmontarlo y conectarlo a un programador, eso si, el diseño que lo emplea ha de contemplar esta posibilidad ya que requiere cumplir algunas normas.

Para programar los dispositivos se emplea una comunicación serie que utiliza tres conexiones: el flujo de datos circula por la patilla **RB7** (bidireccional), **RB6** se emplea como entrada de reloj y **MCLR/Vpp** como tensión de programación; por consiguiente, estas tres patillas de los microcontroladores se podrán emplear en las aplicaciones, pero con ciertas limitaciones.

En primer lugar la patilla **MCLR/Vpp** como tiene dos funciones, una el reset del circuito y otra para la programación serie, tiene que estar perfectamente aislada cuando se emplea como programación. Esto no representa ningún problema ya que para aislar las dos funciones basta con un simple diodo; lo podemos comprobar en el esquema de la Figura 62. El circuito de reset está formado por una resistencia y un condensador.



- Circuito para programación serie

En condiciones normales de funcionamiento la resistencia mantiene a la patilla a nivel alto y cuando se aplica la tensión de programación el diodo la bloquea para que no pase hacia el circuito de reset. Algo más difícil es mantener aisladas las patillas RB6 y RB7 y sobre todo, cuando se emplea el *In-Circuit Debugger* es casi imposible, no obstante, todo depender de los circuitos que estén conectados a estas patillas; si **RB6** y **RB7**, en condiciones normales de funcionamiento, van a actuar siempre como salidas, es obvio que los circuitos que controlan no pueden saber que los datos que están recibiendo no son del microcontrolador, sino del circuito de programación, a no ser que se diseñe un circuito específico gobernado por la tensión de programación.

Otra posibilidad que hay es la de emplear dos patillas más en el conector de forma que, cuando se conecte, realice un puente entre ellas para que sirva de control a un circuito especifico de aislamiento, pero todo tiene sus desventajas, y en este caso, también la hay, ya que mientras esta conectado, obviamente el conjunto del circuito no funcionará. Si las patillas van a actuar siempre como entradas, también presenta problemas aislar las salidas de los circuitos cuando se está programando, pues se deben mantener en alta impedancia.

En definitiva, lo más lógico es que si el diseño contempla la posibilidad de programación serie en el circuito, estas patillas se reserven sólo y exclusivamente para este menester y conectarlas directamente al interfaz de programación.

Bajo voltaje de programación

Es posible programar los circuitos con bajo voltaje siempre que se habilite esta posibilidad con el bit **LVP** de la palabra de configuración. Es este caso, la patilla **RB3/PGM** tendrá funciones de programación, mientras que la **MCLR/Vpp** tiene que estar conectada a VDD.

Ya sabemos que estos microcontroladores funcionan con tensiones de alimentación desde 2 a 5,5 V, pero el requisito necesario para la programación a baja tensión es que la alimentación sea de 5 V.

Mapa de memoria y programación verificación

El espacio reservado para memoria de usuario se extiende desde 0x0000 a 0x1FFF hex.(8K), En el modo de programación este espacio es de 0x0000 a 0x3FFF hex. en dos bloques, la primera mitad llega a 1FFF hex., la segunda comienza en 0x2000 y acaba en 0x3FFF hex. El espacio comprendido desde 0x2000 hex. a 0x2003 hex. está reservado para la ID (Identification Information).

Para estos dispositivos la ID es escrita como " 11 1111 1000 xxxx", donde "xxxx" es la información ID.

La entrada de comandos para programación /verificación entran en serie por la patilla RB7, en seis ciclos de reloj (RB6). Los bits son enclavados por el latch interno por cada flanco de baja da de reloj. Después de un retardo aproximado

a 1 microsegundo se envían los comandos en 16 ciclos de reloj, comenzando por el bit Start y Finalizando por el Stop el comando siempre se enviar comenzando por el bit menos significativo.

Se dispone de 10 comandos de 16 bits cada uno, donde el primer bit y el último siempre son 0, en realidad son 14 bits de datos. Para más información ver la nota de aplicación de Microchip (DS30277B).

El presente manual ha sido confeccionado con material extraído de Internet, información proporcionada por la empresa Microchip y aportes del personal técnico de Firtec.

Por consultas o dudas por favor envíe sus comentarios a informes@firtec.com.ar .