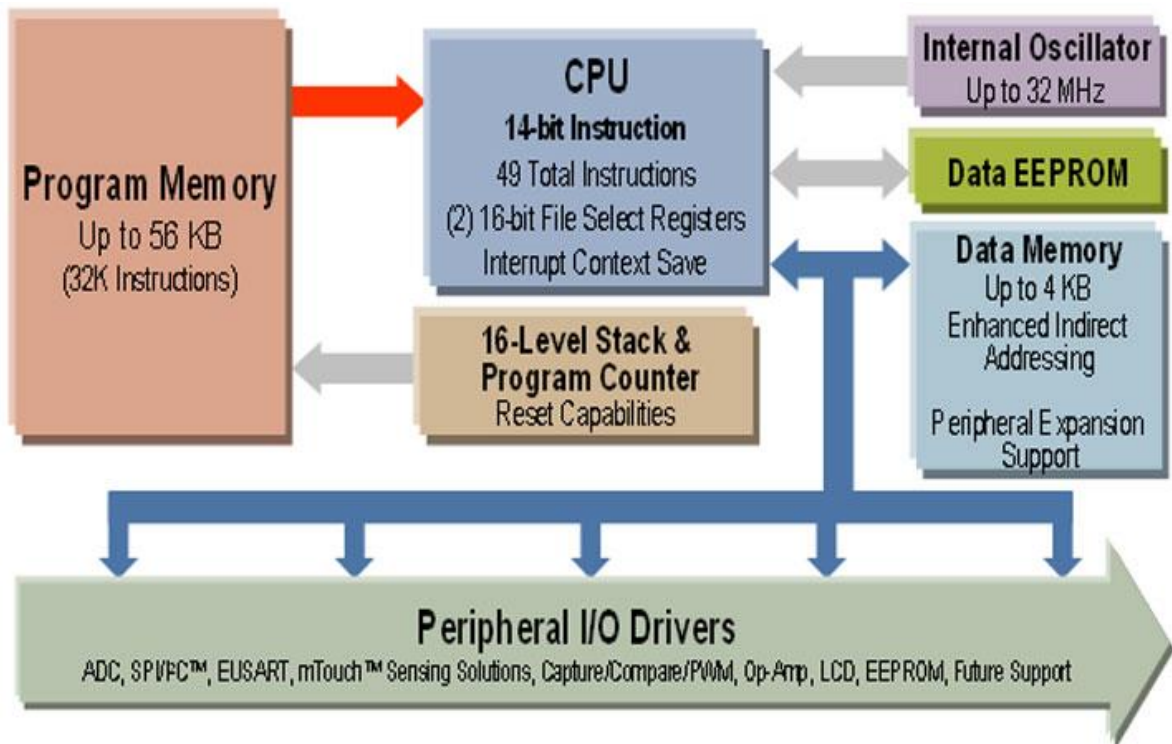


## Familia de gama media mejorada

### MCU PIC® de 8 bits de rango medio mejorado

PIC12F 1 xxx, PIC16F 1 xxx

- 49 (14 bits de ancho) instrucciones fáciles de aprender
- Memoria de programa direccionable de 32 K palabras (56 KB)
- 4 KB de RAM (máx.)
- Pila de hardware de 16 niveles
- 2 registros de selección de archivos (16 bits)
- Manejo de interrupciones de hardware con guardado de contenido
- Conjunto de funciones avanzadas, múltiples comunicaciones en serie y capacidad de control de motores

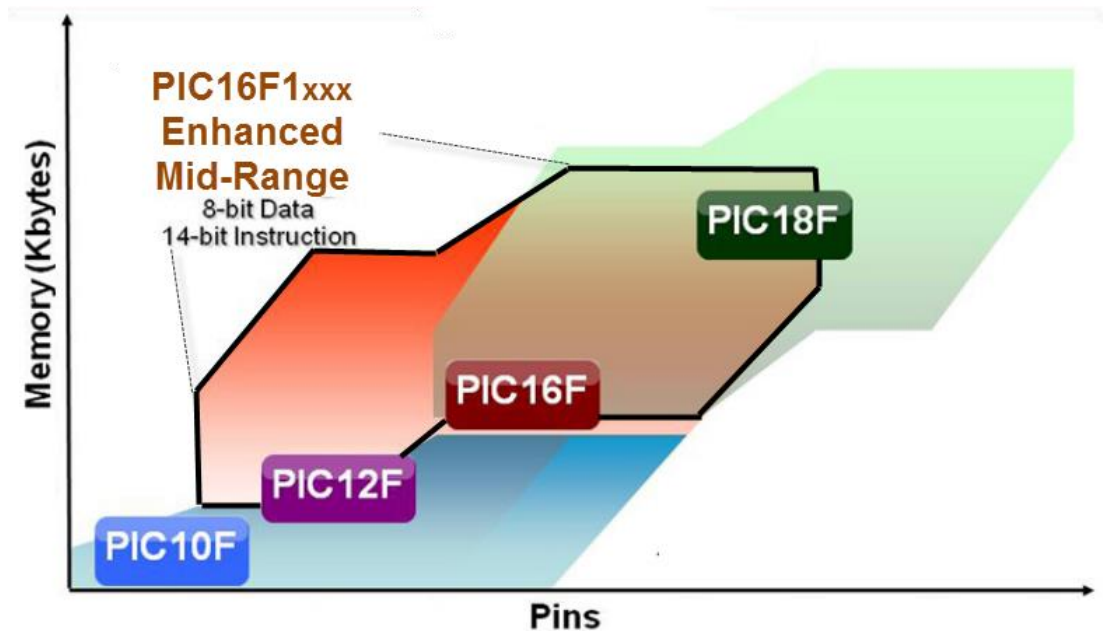


Microchip sigue invirtiendo en su línea de microcontroladores PIC de 8 bits para ofrecer una amplia cartera de productos que satisfaga las necesidades de los clientes actuales y futuros. El nuevo núcleo de rango medio mejorado se basa en los mejores elementos del núcleo de rango medio y proporciona un rendimiento adicional, al tiempo que mantiene la compatibilidad con MCU PIC de rango medio para una verdadera migración de productos.

## Descripción general de la arquitectura MCU PIC de gama media mejorada

### Resumen

### Familias de MCU de 8 bits de Microchip

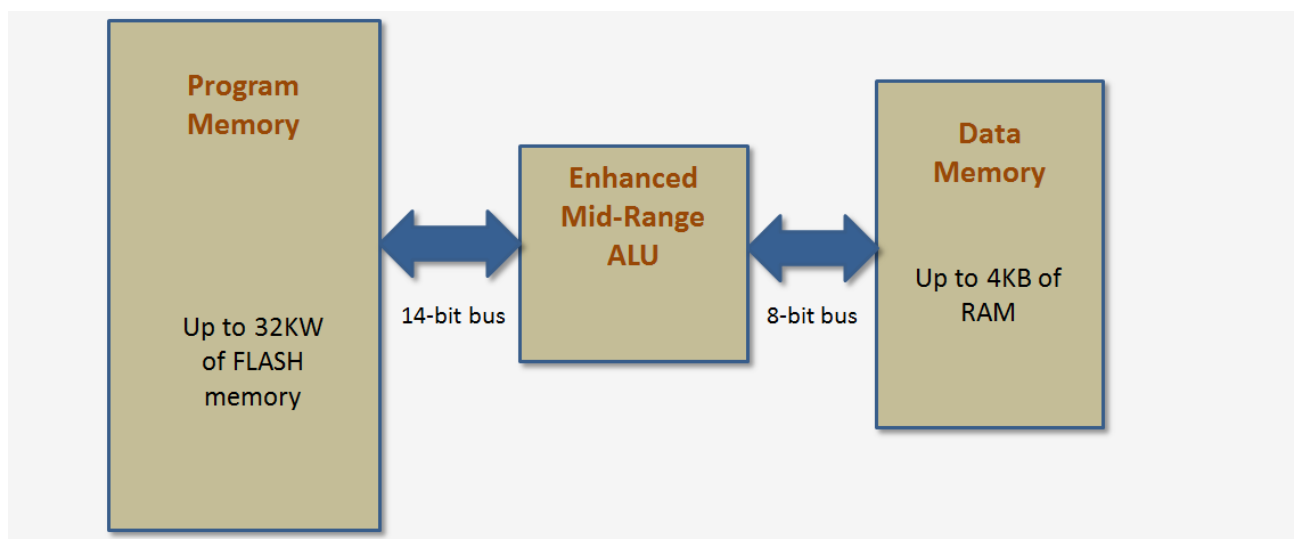


La familia de MCU PIC® de 8 bits de rango medio mejorado PIC16F1xxx <sup>abarca</sup> una amplia variedad de tamaños de memoria y pines de E/S.

Esta página presenta las características arquitectónicas clave de la familia de MCU PIC16F1xxx. En esta página se proporcionan enlaces a los detalles técnicos necesarios para implementar aplicaciones en la familia de microcontroladores PIC de gama media mejorada.

### Arquitectura de Harvard

Los microcontroladores PIC® de gama media mejorados utilizan una arquitectura Harvard de doble bus.



autobús de instrucción

Las instrucciones del programa se introducen en la ALU desde la memoria del programa FLASH a través del bus de instrucciones de 14 bits. En cada ciclo de reloj de instrucción, se lee una palabra de programa de 14 bits en la ALU.

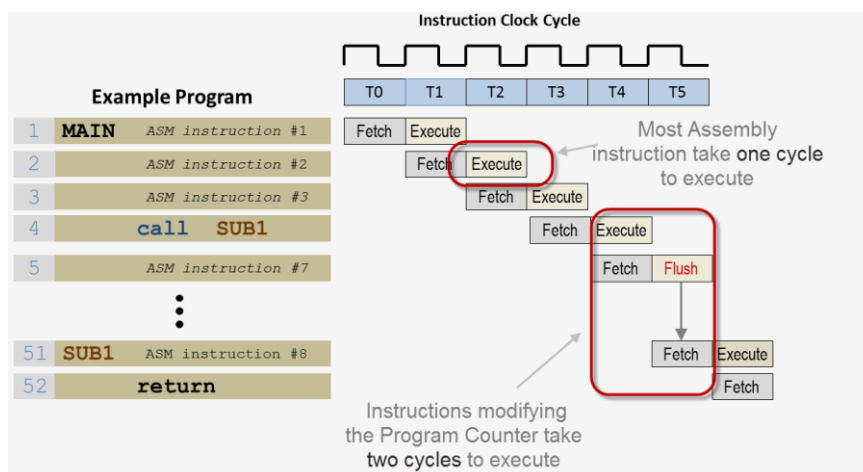
### Bus de datos

Un bus de datos de 8 bits conecta la ALU al espacio de memoria de datos. Durante cada instrucción, la ALU puede leer datos desde la ubicación de la memoria de datos, modificar los datos y luego volver a escribir los datos en la memoria.

### Canalización de instrucciones

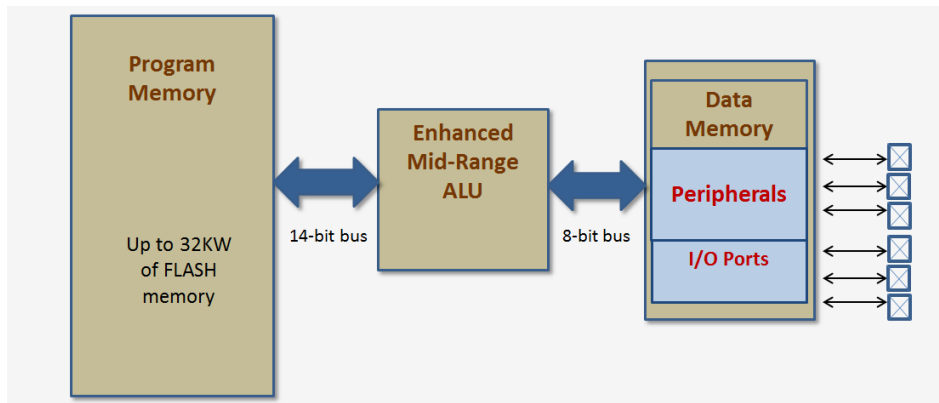
La arquitectura de doble bus del PIC de rango medio mejorado proporciona una línea de instrucción de dos etapas. Una cada ciclo de reloj ejecuta dos fases de instrucción:

1. La siguiente instrucción se "**obtiene**" de la memoria del programa
2. La instrucción actual se "**ejecuta**" y lee/modifica/escribe la memoria de datos (si es necesario)



### Periféricos asignados a la memoria

Una mirada más cercana a la sección de memoria de datos de la MCU PIC® de gama media mejorada muestra que se accede a los registros que controlan los periféricos y a los puertos de E/S leyendo o escribiendo en direcciones de memoria de datos específicas. Esta asignación de periféricos a la dirección de la memoria simplifica enormemente el aprendizaje de cómo programar el PIC mejorado de rango medio.



La [página de memoria](#) de datos del tutorial mejorado de gama media ofrece una descripción completa junto con ejemplos de programas que acceden a los periféricos asignados a la memoria.

### Conjunto de instrucciones ortogonales

Cada MCU PIC<sup>®</sup> de gama media mejorada tiene 49 instrucciones. Las instrucciones que acceden directamente a las direcciones de la memoria de datos se ejecutan en un ciclo de instrucción. Las instrucciones que provocan un cambio en el contador del programa (BRA, GOTO, RETURN, CALL, ..etc) tardan dos ciclos de instrucción en ejecutarse.

Al mapear los registros de E/S y periféricos a direcciones de memoria, las MCU PIC no necesitan instrucciones especiales para las operaciones de E/S o para establecer registros periféricos. Escribir en un puerto de E/S o configurar un periférico es una simple escritura en una ubicación de memoria. Leer el valor de un pin de entrada, registro de resultado ADC o temporizador es una simple lectura de una ubicación de memoria. Mediante el uso de una pequeña cantidad de instrucciones ortogonales, los MCU PIC de rango medio mejorado son fáciles de programar, usan menos silicio para construir y consumen menos energía.

### Ejemplos de implementación de instrucciones

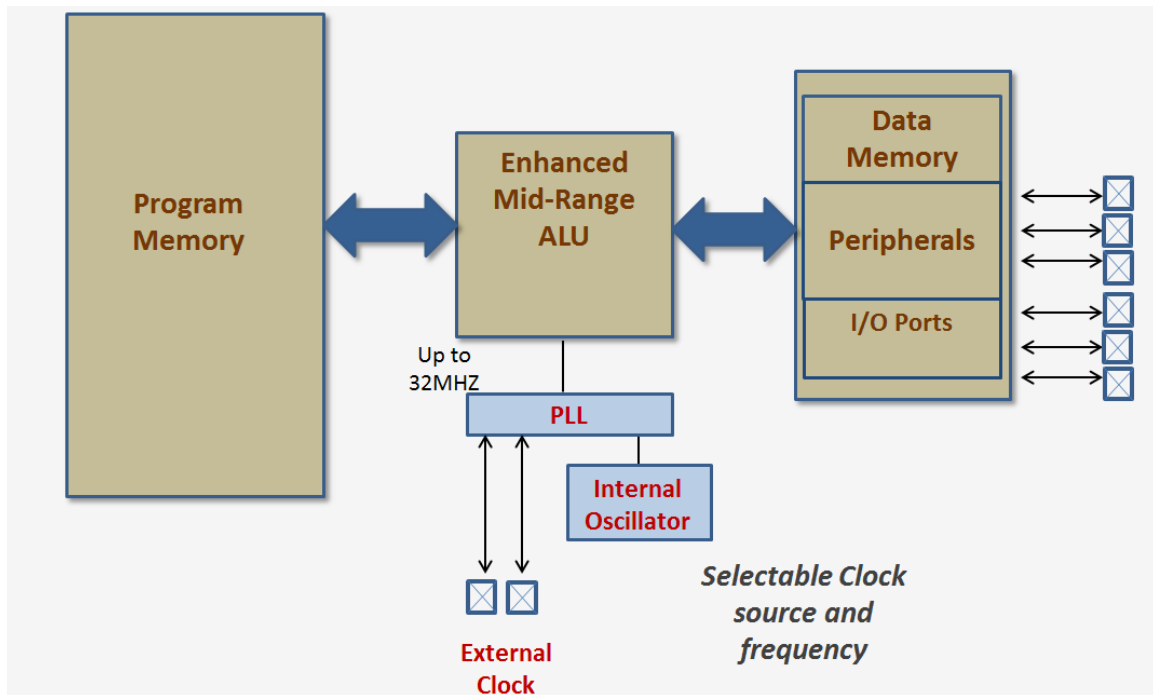
Task	C Language	Assembly
<b>WRITE</b> a value to a variable	var1 = 7 ;	<b>movlw</b> 7 <b>movf</b> var1
<b>WRITE</b> a value to a Register	T1CON = 0x72;	<b>movlw</b> 0x72 <b>movf</b> T1CON
<b>WRITE</b> to an Output Latch	LATD = 0xFF ;	<b>movlw</b> 0xFF <b>movf</b> LATD
<b>READ</b> an input PORT	var1 = PORTB;	<b>movf</b> PORTB <b>movwf</b> var1
<b>READ</b> a variable's value	var1 = var2;	<b>movf</b> var2 <b>movwf</b> var1
<b>READ</b> an SFR into another SFR	CCPR1L = ADRESH;	<b>movf</b> ADRESH <b>movwf</b> CCPR1L

Para obtener una lista detallada del conjunto de instrucciones y una discusión completa del tiempo de instrucción, consulte la página [Conjunto](#) de instrucciones del tutorial de rango medio mejorado.

### Opciones de reloj flexibles (hasta 32 MHz)

Seleccionado por los [bits de configuración](#) del PIC<sup>®</sup> MCU, el reloj del sistema tiene las siguientes propiedades:

- Fuente opcional ( **oscilador interno** o **circuito externo** )
- Opciones de velocidad flexibles **hasta 32MHz**
- **Arranque de dos velocidades** : permite que el sistema ejecute el software de inicialización mientras el oscilador externo se estabiliza
- **Cambio** de reloj: la fuente de reloj del sistema se puede cambiar entre fuentes de reloj externas e internas a través del software.
- **Monitor de reloj** a prueba de fallas: cambia al oscilador interno en caso de falla del reloj externo



Para obtener una descripción detallada de las opciones de configuración del oscilador, consulte la [página del oscilador de 8 bits](#) en el Tutorial mejorado de rango medio.

### E/S digitales

Casi todos los pines de la MCU PIC Enhance de rango medio se pueden usar como pines de entrada o salida digital. Los pines digitales comparten estos atributos:

- Supervisión de entradas digitales
- Controlar dispositivos digitales
- Dominadas débiles internas
- Multiplexado con Periféricos
- Alta capacidad de accionamiento (hasta 25 mA sumidero/fuente en muchos pines de E/S)
- Manipulación directa de bit de ciclo único
- Diodos de protección ESD de 4kV

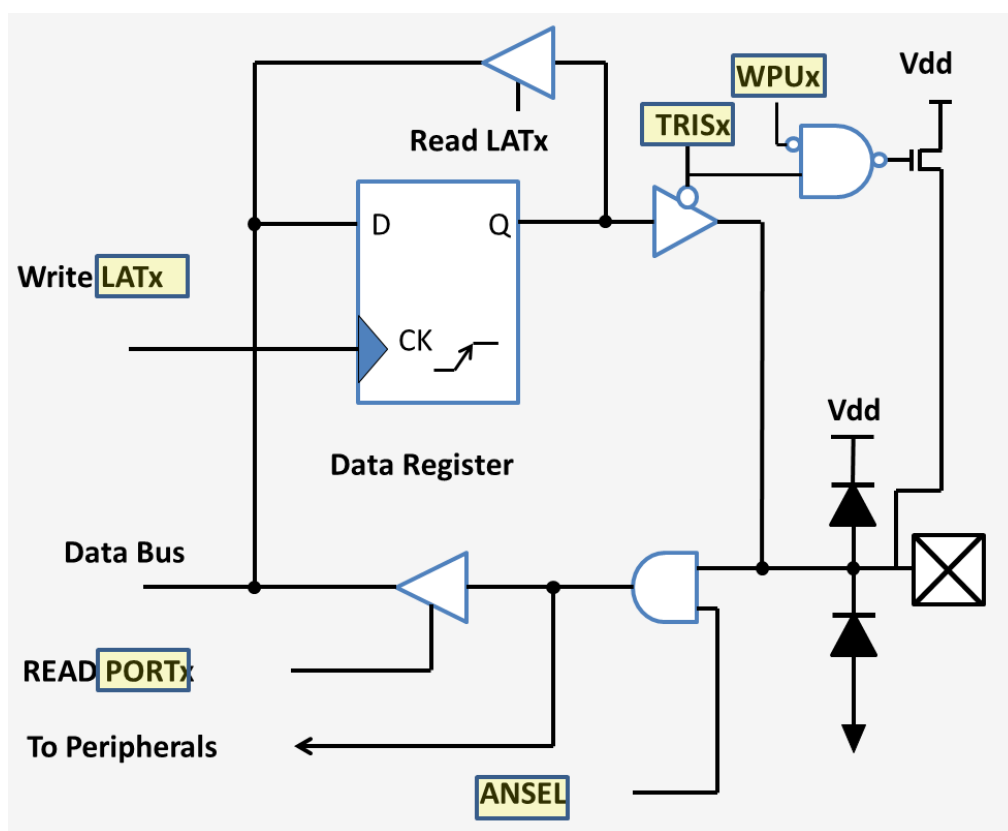
Al reiniciar:

- Los pines digitales vuelven a la entrada (Hi-Z)
- Los pines con capacidad analógica vuelven a ser analógicos

### Estructura típica de pines digitales

Cinco registros controlan el funcionamiento de un pin digital. Estos registros de 8 bits controlan 8 pines de un PUERTO. Usando los registros **TRISx**, **PORTx**, **LATx**, **WPUx** y **ANSEL**, el programa puede:








- Configure el pin a como entrada o salida **TRISx**
- Leer un pin de entrada (o los 8 pines PORT) **PORTx**
- Envía un 1 o 0 a un pin **LATx**
- Habilitar o deshabilitar la resistencia pull up interna **WPUx**
- Determine si los pines con capacidad analógica funcionan en modo analógico o digital **ANSEL**



Para una discusión completa de la E/S digital PIC mejorada de rango medio, incluidos los detalles sobre la programación de las operaciones de entrada y salida digital, consulte la sección de [E/S digital](#) de este tutorial mejorado de rango medio.

### Pines multiplexados

Además de configurarse como E/S digitales, los pines de los MCU PIC mejorados de rango medio pueden tener varias funciones posibles. El diagrama de pines en la hoja de datos muestra las opciones para cada pin. Al inicio, el programa tiene la opción de configurar los pines.

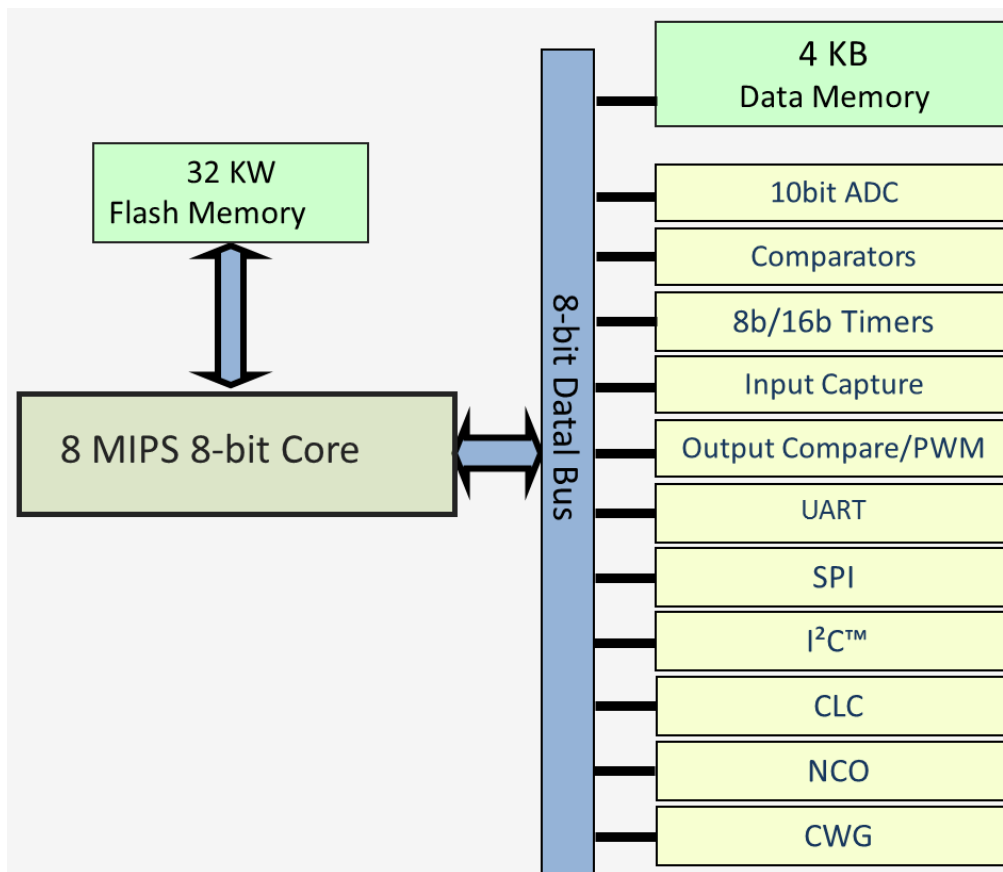
40		↔	RB7/ICSPDAT/ICDDAT/SEG13
39		↔	RB6/ICSPCLK/ICDCLK/SEG14
38		↔	RB5/AN13/CPS5/CCP3/P#A/T1G/COM1
37		↔	RB4/AN11/CPS4/COM0
36		↔	RB3/AN9/C12IN2-/CPS3/CCP2/P2A/VLCD3
35		↔	RB2/AN8/CPS2/VLCD2
34		↔	RB1/AN10/C12IN3/CPS1/VLCD1
33		↔	RB0/AN12/CPS0/SRI/INT/SEG0

Para una discusión completa de la E/S digital PIC mejorada de rango medio, incluidos los detalles sobre la configuración de los pines, consulte la [sección Periféricos](#) del tutorial de rango medio mejorado.

#### **Periféricos avanzados**

Además de las E/S digitales mejoradas, los miembros de la familia de MCU PIC de rango medio tienen una variedad de periféricos avanzados. Estos periféricos incluyen periféricos para conversión de datos, comunicación y acondicionamiento de señales.





Para obtener una lista completa de los periféricos disponibles, consulte la [sección Periféricos](#) del tutorial .

### Interrupciones

Los microcontroladores PIC de rango medio mejorados utilizan una estructura de interrupción preventiva de un solo vector.

Cada periférico en el PIC es capaz de generar una solicitud de interrupción. Cuando ocurre una solicitud de interrupción Y las interrupciones para el dispositivo solicitante están habilitadas, se producirá una interrupción.

El PIC de rango medio mejorado utiliza una pila de hardware de 16 niveles para almacenar el contenido actual de la PC cuando ocurre una interrupción. El contexto del programa se guarda en registros sombra y el control se pasa a la dirección de memoria del programa 0x04.

### Rutina de servicio de interrupción (ISR)

El usuario es responsable de escribir el código para dar servicio a la interrupción y colocar el código en la dirección 0x04. Esta Rutina de Servicio de Interrupción (ISR) determina la fuente de la interrupción, luego realiza la tarea necesaria para dar servicio al periférico interrumpido. La instrucción final de un ISR es la instrucción Return From Interrupt (RETFIE).

### Guardado automático de contexto

Los siguientes registros se guardan en un registro de sombra de un solo nivel establecido en caso de una interrupción

- registro W
- BSR
- ESTADO
- FSR
- PCLATH

Cuando el ISR ejecuta la instrucción RETFIE, estos registros se restauran al valor anterior a la interrupción.

### **Prioridad de interrupción de un solo nivel**

Cuando ocurre una interrupción, el bit de habilitación de interrupción global (GIE) en el registro de estado está deshabilitado. Esto evitará que una interrupción sea reemplazada por otra interrupción.

Al ejecutar un RETFIE, el estado del bit de control GIE se restaura a su valor previo a la interrupción.

Para obtener una descripción completa del proceso de interrupción y ejemplos de programación, consulte la [sección](#) de interrupción del tutorial de rango medio mejorado.

### **Qué sucede en el inicio del sistema (RESET)**

Hay varias fuentes de un REINICIO en la MCU PIC mejorada de rango medio. Las fuentes de RESET comunes a casi todas las aplicaciones son Power On Reset (POR) y Brown Out Reset (BOR) debido a una caída del voltaje de la fuente de alimentación (es decir, caída de tensión). Hay varios otros métodos para restablecer la MCU, incluido el tiempo de espera de Watchdog y el acceso directo al pin MCLR.

### **El contador de programa se establece en 0x00.**

Después de un RESET, la instrucción ubicada en la dirección 0 es la primera instrucción ejecutada. El desarrollador de la aplicación es responsable de colocar el código en esta dirección para "iniciar" el PIC. El compilador MPLAB® XC8 de Microchip insertará las instrucciones apropiadas para poner en marcha el PIC y transferir el control a **main**. Los programadores de nivel de ensamblaje tendrán que escribir el código para inicializar el PIC y pasar el vector de interrupción ubicado en la dirección 0x04.

### **Todos los registros de funciones especiales se establecen en un valor predeterminado**

La hoja de datos para cada MCU PIC de rango medio mejorado muestra los valores que contendrán los registros en RESET.

### Sample Register

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>	
bit 7							bit 0

<b>Legend:</b>		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

### Conjunto de instrucciones de rango medio mejorado

Esto se aplica a las familias de microcontroladores PIC<sup>®</sup> PIC16F1xxx y PIC16LF1xxx .

#### Operaciones orientadas a bytes

Mnemónicos, Operandos		Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
ADDWF	f, d	Añadir W y f	1	00 0111 ffff ffff	C, CC, Z	2
AÑADIRWFC	f, d	Suma con Carry W y f	1	11 1101 ffff ffff	C, CC, Z	2
ANDWF	f, d	Y W con f	1	00 0101 ffff ffff	Z	2
ASRF	f, d	Desplazamiento aritmético a la derecha	1	11 0111 ffff ffff	C, Z	2
LSLF	f, d	Desplazamiento lógico a la izquierda	1	11 0101 ffff ffff	C, Z	2
LSRF	f, d	Desplazamiento lógico a la derecha	1	11 0110 ffff ffff	C, Z	2
CLRF	F	Borrar f	1	00 0001 1fffffff	Z	2

CLRW		Borrar W	1	00 0001 0000 00xx	Z	
COMF	f, d	Complemento f	1	00 1001 ffff ffff	Z	2
DECF	f, d	Decremento f	1	00 0011 ffff ffff	Z	2
FIN	f, d	Incremento f	1	00 1010 ffff ffff	Z	2
IORWF	f, d	Inclusivo OR W con f	1	00 0100 dfff ffff	Z	2
MOVF	f, d	Mover f	1	00 1000 ffff ffff	Z	2
MOVWF	F	Mover W a f	1	00 0000 1ffffff	Ninguna	2
RLF	f, d	Girar a la izquierda f a través de Carry	1	00 1101 ffff ffff	C	2
FRR	f, d	Girar a la derecha f a través de Carry	1	00 1100 dfff ffff	C	2
SUBWF	f, d	Reste con Préstamo W de f	1	11 1011 ffff ffff	C, CC, Z	2
SUBWFB	f, d	Restar W de f	1	00 0010 ffff ffff	C, CC, Z	2
SWAPF	f, d	Intercambiar mordiscos en f	1	00 1110 ffff ffff	Ninguna	
XORWF	f, d	Exclusivo OR W con f	1	00 0110 ffff ffff	Z	2
Instrucciones de omisión orientadas a bytes						

<b>Mnemónicos, Operandos</b>	<b>Descripción</b>	<b>Ciclos</b>	<b>Código de operación de 14 bits MSb.....LSb</b>	<b>Estado Afectado</b>	<b>notas</b>
----------------------------------	--------------------	---------------	-----------------------------------------------------------	----------------------------	--------------

DECFSZ	f, d	Decrementar f, Saltar si 0	1(2)	00 1011 ffff ffff	Ninguna	1,2
--------	------	----------------------------	------	-------------------	---------	-----

INCFSZ	f, d	Incrementar f, Saltar si 0	1(2)	00 1111 ffff ffff	Ninguna	1,2
--------	------	----------------------------	------	-------------------	---------	-----

### Operaciones de registro de archivos orientados a bits

Mnemónicos, Operandos		Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
FCB	pensión completa	poco claro f	1	01 00bb bff ffff	Ninguna	2
BSF	pensión completa	Juego de bits f	1	01 01bb bfff ffff	Ninguna	2

### Operaciones de salto orientadas a bits

Mnemónicos, Operandos		Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
BTFSC	pensión completa	Bit Test f, Saltar si Borrar	1(2)	01 10bb bfff ffff	Ninguna	1,2
BTFSS	pensión completa	Prueba de bits f, omitir si se establece	1(2)	01 11bb bfff ffff	Ninguna	1,2

### operaciones literales

Mnemónicos, Operandos		Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
ADDLW	k	Agregar literal y W	1	11 1110 kkkk kkkk	C, CC, Z	
ANDLW	k	AND literal con W	1	11 1001 kkkk kkkk	Z	

IORLW	k	literal OR inclusivo con W	1	11 1000 kkkk kkkk	Z
MOVLB	k	Mover literal a BSR	1	00 0000 001k kkkk	Ninguna
MOVLP	k	Mover literal a PCLATH	1	11 0001 1kkk kkkk	Ninguna
MOVLW	k	Mover literal a W	1	11 0000 kkkk kkkk	Ninguna
SUBLW	k	Restar W de literal	1	11 1100 kkkk kkkk	C, CC, Z
XORLW	k	Exclusivo OR literal con W	1	11 1010 kkkk kkkk	Z

**operaciones de control**

<b>Mnemónicos, Operandos</b>		<b>Descripción</b>	<b>Ciclos</b>	<b>Código de operación de 14 bits MSb.....LSb</b>	<b>Estado Afectado</b>	<b>notas</b>
SOSTÉN	k	Rama relativa	2	11 001k kkkk kkkk	Ninguna	
BRW		Rama relativa con W	2	00 0000 0000 1011	Ninguna	
LLAMAR	k	Subrutina de llamada	2	10 0kkk kkkk kkkk	Ninguna	
LLAMAR		Subrutina de llamada con W	2	00 0000 0000 1010	Ninguna	
IR	k	Ir a dirección	2	10 1kkk kkkk kkkk	Ninguna	
RECUPERAR	k	Regreso de interrupción	2	00 0000 0000 1001	Ninguna	
RETLW	k	Volver, colocar literal en W	2	11 0100 kkkk kkkk	Ninguna	
DEVOLVER	k	Regreso de subrutina	2	00 0000 0000 1000	Ninguna	

## Operaciones inherentes

Mnemónicos, Operandos	Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
CLRWDT	Borrar temporizador de vigilancia	1	00 0000 0110 0100	A , PD	
NOP	No operacion	1	00 0000 0000 0000	Ninguna	
OPCIÓN	Cargue el registro de OPCIÓN con W	1	00 0000 0110 0010	Ninguna	
REINICIAR	Restablecer dispositivo de software	1	00 0000 0000 0001	Ninguna	
DORMIR	Entrar en modo de espera	1	00 0000 0110 0011	A , PD	
TRIS            F	Cargar registro TRIS	1	00 0000 0110 0fff	Ninguna	

## Compilador C optimizado

Mnemónicos, Operandos	Descripción	Ciclos	Código de operación de 14 bits MSb.....LSb	Estado Afectado	notas
ADDFSR	Agregar literal a FSRn	1	11 0001 0nkk kkkk	Ninguna	
PELÍCULA	Mover FSRn indirecto a W	1	00 0000 0001 0nnn	Z	2
MOVWI	Mover W a FSRn indirecto	1	00 0000 0001 1nnnn	Z	2

### notas

1. Si se modifica el contador de programa (PC), o si una prueba condicional es verdadera, la instrucción requiere dos ciclos. El segundo ciclo se ejecuta como NOP .
2. Si esta instrucción direcciona un registro INDF y se establece el MSb del FSR correspondiente, la instrucción requiere un ciclo de instrucción adicional.

Los bits de configuración son una colección de datos binarios ubicados en la memoria flash de un microcontrolador <sup>PIC®</sup> (MCU). Los bits de configuración se programan en la MCU PIC con el código de la aplicación. No son código ejecutable ya que el Contador de programa (PC) no puede acceder a su dirección. Cuando se programan en una MCU PIC, los bits de configuración completan el circuito que activa o desactiva las funciones de hardware de la MCU.

Los bits de configuración se leen al salir de un restablecimiento y no se pueden modificar durante el tiempo de ejecución.

Las características especiales de la operación de MCU controladas por los bits de configuración incluyen:

1. Reloj del sistema
2. Administración de energía
3. Seguridad del dispositivo
4. Características de funcionamiento

Los bits de configuración se generan a partir de directivas de compilador/ensamblador incluidas en los archivos de código fuente.

Esta página describe qué funciones están controladas por bits de configuración y cómo generarlas en el código fuente.

Los bits de configuración y las configuraciones para dispositivos PIC16F1xxx individuales pueden variar. Consulte su hoja de datos para conocer los detalles de los bits de configuración de PIC MCU que está utilizando.

### **Ubicación y formato**

Los bits de configuración para la familia de MCU PIC16F1 se combinan en dos palabras de 14 bits denominadas CONFIG1 y CONFIG2. Las palabras de configuración están ubicadas fuera del alcance de la PC en las direcciones 0x8007 y 0x8008 en la memoria flash de la MCU.



## REGISTER 10-1: CONFIGURATION WORD 1

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN1	BOREN0	$\overline{\text{CPD}}$	$\overline{\text{CP}}$
bit 13						bit 7

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
MCLRE	$\overline{\text{PWRTE}}$	WDTE1	WDTE0	FOSC2	FOSC1	FOSC0
bit 6						bit 0

## REGISTER 10-2: CONFIGURATION WORD 2

R/P-1/1	R/P-1/1	U-1	R/P-1/1	R/P-1/1	R/P-1/1	U-1
LVP	$\overline{\text{DEBUG}}$	—	BORV	STVREN	PLEN	—
bit 13						bit 7

U-1	R/P-1/1	R/P-1/1	U-1	U-1	R/P-1/1	R/P-1/1
—	VCAPEN1	VCAPEN0	—	—	WRT1	WRT0
bit 6						bit 0

Los bits de configuración se insertan en el código fuente de la aplicación. Cuando se construye un proyecto PIC MCU, los ajustes de bits de configuración se cargan en el archivo de salida HEX. Los bits de configuración se programan en el PIC con el programa de aplicación.

### **Generación de bits de configuración en código C**

El compilador MPLAB<sup>®</sup> XC8 C de Microchip acepta directivas `#pragma` para establecer los bits de configuración.

La sintaxis para generar bits de configuración:

**#pragma config** *CONFIG\_BIT\_NAME* = *CONFIG\_VALUE*

### **Ejemplo de establecimiento de bits de configuración en C**

Cuando se configuran los bits de configuración usando C, no es necesario conocer la palabra que contiene el bit que se está configurando. Todo lo que se necesita es el nombre del bit de configuración y el valor deseado.

Los archivos de encabezado para cada dispositivo PIC16F1xx contienen `CONFIG_BIT_NAME` y `CONFIG_VALUE`. Puede encontrar una lista completa de los ajustes en el [resumen de bits de configuración](#).

**Una vez escritas, las directivas del compilador deben agregarse al proyecto PIC MCU de una de estas tres maneras:**

1. En un archivo fuente C independiente que se agrega al proyecto
2. En un archivo de encabezado ( `.h` ) colocado en el proyecto con una instrucción `#include`
3. Colocado directamente dentro de uno de los archivos de código fuente que ya están en el proyecto

Los desarrolladores que utilizan el compilador XC8 pueden consultar la sección de [bits de configuración](#) del tutorial MPLAB X para ver accesos directos en la generación del código necesario para establecer los bits de configuración.

### **Generación de bits de configuración en ensamblaje**

Cuando se trabaja con lenguaje ensamblador, se requiere que el programador genere los valores específicos de 14 bits para cada una de las dos direcciones de configuración. Una vez que se generan los patrones, el programa realiza una llamada a la directiva CONFIG para establecer cada una de las palabras de configuración.

En el archivo .INC de cada MCU de PIC se incluye un patrón de 14 bits para cada configuración de bit de configuración individual . El valor de 14 bits que se pasa a la directiva CONFIG se genera mediante la combinación lógica AND de la configuración de bits en el archivo .INC .

El código para generar y cargar los bits de configuración se puede colocar en cualquier archivo de origen de ensamblado del proyecto.

## Reloj del sistema

### Fuente de reloj (FOSC)

FOSC consta de tres bits de configuración individuales: FOSC2 , FOSC1 y FOSC0 . El campo de bits FOSC se encuentra en CONFIG1 .

FOSC selecciona la fuente de reloj para MCU.

Las opciones para FOSC son:

FOSC<2:0>	Ajuste	reloj agrio
0 0 0	INTOSC	oscilador interno
0 0 1	EXTRC	Oscilador RC externo
0 1 0	SA	Oscilador de cristal externo de alta velocidad
0 1 1	XT	Oscilador de cristal externo
1 0 0	LP	Oscilador de cristal externo de baja potencia
1 0 1	ECH	Reloj externo con rango de frecuencia 4 - 32 MHz
1 1 0	ECM	Reloj externo con rango de frecuencia 0,5 - 4 MHz
1 1 1	ECL	Reloj externo con rango de frecuencia 0 - 0,5 MHz

### Ejemplo: Selección del reloj del sistema

### Monitor de reloj a prueba de fallas (FCMEN)

El FCMEN es un único bit de configuración que reside en CONFIG1 .

FCMEN controla el funcionamiento del monitor de reloj a prueba de fallas, lo que permite que el reloj cambie de externo a interno en caso de falla del reloj externo.

Las opciones para FCMEN son:

FCMEN	Ajuste	Función de monitor de reloj a prueba de fallas
0	APAGADO	Desactivado
1	EN	Activado

**Ejemplo: Habilitación del monitor de reloj a prueba de fallas**

### Conmutación interna/externa (IESO)

El IESO es un único bit de configuración que reside en CONFIG1 .

IESO establece el modo de cambio de reloj y arranque de dos velocidades. Con IESO habilitado, la fuente del reloj puede controlarse mediante el programa de aplicación.

Las opciones para IESO son:

IESO	Ajuste	Función de puesta en marcha de dos velocidades
0	APAGADO	Desactivado
1	EN	Activado

**Ejemplo: Habilitación del arranque de dos velocidades**

### Habilitación de salida de reloj ( CLKOUTEN )

CLKOUTEN es un único bit de configuración que reside en CONFIG1 .

CLKOUTEN permite que el pin OSCx/CLKOUT emita el reloj del sistema interno. Esto permite que el reloj del sistema PIC16F1xxx controle otros componentes.

Las opciones para CLKOUTEN son:

CLKOUTEN	Ajuste	Función CLKOUT
0	EN	Fosc se enviará a OSCx/CLKOUT
1	APAGADO	OSCx/CLKOUT será el oscilador o una función periférica

**Ejemplo: salida del reloj del sistema interno**

#### Habilitación de bucle de bloqueo de fase (PLEN)

PLEN es un único bit de configuración que reside en CONFIG2 .

El bucle de bloqueo de fase (PLL) de 4 X del oscilador interno se controla mediante una combinación del bit de configuración PLEN y el bit SPLLEN en el registro OSCON.

Las opciones para PLEN son:

PLEN	Ajuste	Función PLL interna
0	EN	4 X PLL siempre está habilitado
1	APAGADO	4 X PLL está controlado por el bit SPLLEN en OSCCON

No todos los MCU PIC16F1xxx tienen las mismas opciones para el PLL. Consulte la hoja de datos de la MCU PIC que está utilizando para determinar los ajustes de configuración de PLL específicos.

**Ejemplo: Habilitación del bucle de bloqueo de fase interno**

#### Administración de energía

### Habilitación de restablecimiento de caída de tensión (BOREN)

El ajuste de configuración de BOREN consta de dos bits individuales: BOREN1 y BOREN0 . El campo de bits BOREN reside en CONFIG1 .

BOREN permite que ocurra un REINICIO DE MCU si  $V_{dd}$  cae por debajo de un valor preestablecido. El nivel de voltaje que precipita el RESET está determinado por el bit de configuración BORV .

Hay cuatro opciones para los dos bits BOREN :

1. Brown-out Reset siempre está habilitado
2. Brown-out Reset siempre está deshabilitado
3. Brown-out Reset está habilitado cuando se está ejecutando pero deshabilitado cuando MCU ingresa al modo SLEEP
4. Brown-out Reset es controlado en tiempo de ejecución por el bit SBOREN del registro PCON.

ABURRIDO	Ajuste	Función de reinicio de oscurecimiento
1 1	EN	Siempre habilitado
0 0	APAGADO	Siempre deshabilitado
1 0	NSLEEP	Habilitado mientras está ACTIVO, deshabilitado en modo SLEEP
0 1	SBODEN	controlado por el bit SBOREN del registro PCON

**Ejemplo: Desactivación de RESET de Brown-out**

### Nivel de voltaje de caída de voltaje (BORV)

BORV es un único bit de configuración que reside en CONFIG2 .

BORV solo es aplicable cuando el reinicio de Brown-out (controlado por el bit de configuración BOREN ) está activo.

BORV selecciona uno de los dos niveles de voltaje preestablecidos como el voltaje de restablecimiento de Brown-out.

Los niveles de voltaje establecidos por BORV son:

BORV	Ajuste	Voltaje de caída de tensión
0	L0	1,9 voltios
1	HOLA	2,5 voltios

El nivel de voltaje de caída de voltaje establecido por BORV puede variar según la MCU de PIC en particular. Consulte la hoja de datos para conocer las opciones de voltaje de reducción de voltaje.

**Ejemplo: Permita que Brown-out RESET esté siempre activo y activado a 2,5 V.**

**Ejemplo: habilite el REINICIO de Brown-out a 1,9 V mientras la MCU está ACTIVA y deshabilite Brown-out en el modo SLEEP.**

### Regulador de Voltaje (VCAPEN - LDO)

El ajuste de configuración de VCAPEN consta de dos bits individuales: VCAPEN1 y VCAPEN0 . El campo de bits VCAPEN reside en CONFIG2 .

Para dispositivos con un regulador LDO interno, VCAPEN determina qué pin se asigna como pin de tapa en V.

Los ajustes para VCAPEN son:

VCAPEN	Ajuste	Función
0 0	RA6	RA6 se asigna como <small>tapa V</small>
0 1	RA5	RA5 se asigna como <small>tapa V</small>
1 0	RA0	RA0 se asigna como <small>tapa V</small>
1 1	APAGADO	La <small>tapa</small> en V está desconectada de todos los pines

No todas las MCU PIC16F1xxx tienen un regulador LDO interno. Consulte la hoja de datos de la MCU PIC que está utilizando para determinar si hay un LDO y qué pines están disponibles como  $V_{cap}$ .

**Ejemplo: Asignación de  $V_{cap}$  a RA6**

**Deshabilitar la funcionalidad de tapa en  $V$**

**Programación de bajo voltaje (LVP)**

LVP es un único bit de configuración que reside en CONFIG2.

El modo de entrada de programación de bajo voltaje permite que las MCU PIC16F1xxx se programen con solo  $V_{dd}$ . El uso de LVP elimina la necesidad de suministrar un voltaje superior a  $V_{dd}$  en MCLR /  $V_{pp}$ .

Los ajustes para LVP son:

LVP	Ajuste	Función de programación
1	EN	Activado
0	APAGADO	Desactivado

**Ejemplo: deshabilitar las lecturas de EEPROM desde fuentes externas**

**Seguridad del dispositivo**

**CPD - Protección de lectura de EEPROM de datos**

CPD es un único bit de configuración que reside en CONFIG1.

La memoria EEPROM de datos internos se puede proteger de lecturas externas con CPD. Los programadores externos tienen prohibido leer la EEPROM protegida. El contenido de EEPROM todavía está disponible para lecturas desde fuentes internas.

Los ajustes para CPD son:



DPC	Ajuste	Función de protección de lectura de EEPROM
0	EN	Habilitado - EEPROM no es legible
1	APAGADO	Deshabilitado - EEPROM es legible

**Ejemplo: deshabilitar las lecturas de EEPROM desde fuentes externas**

### Protección de lectura de memoria de programa ( CP )

CP es un único bit de configuración que reside en CONFIG1 .

Todo el espacio de la memoria del programa se puede proteger de lecturas externas con CP .

Verá todos los 0 cuando lea la memoria del programa protegido. El contenido de la memoria del programa todavía está disponible para lecturas desde fuentes internas.

Los ajustes para CP son:

PC	Ajuste	Función de protección de lectura
0	EN	Habilitado - la memoria no es legible
1	APAGADO	Deshabilitado - la memoria es legible

**Ejemplo: deshabilitar las lecturas de la memoria del programa externo**

### Activación de autoescritura flash (WRT)

WRT consta de dos bits de configuración individuales: WRT1 y WRT0 . El campo de bits WRT se encuentra en CONFIG2 .

WRT establece el rango de direcciones en el que el programa de usuario puede escribir la memoria del programa.

Los ajustes para WRT son:

WRT	Ajuste	Funcionalidad Flash Self Right
-----	--------	--------------------------------

1 1	TODOS	El programa de usuario no puede escribir en la memoria
1 0	BOTA	Rango de direcciones 0 - 0x1FF protegido contra escritura 200 h - FFFn grabable
0 1	MITAD	Rango de direcciones 0 - 0x7FF protegido contra escritura 800 h - FFFn grabable
0 0	APAGADO	El programa de usuario puede escribir en todas las direcciones de memoria

No todas las MCU PIC16F1xxx tienen las mismas opciones de WRT que se muestran. Consulte la hoja de datos de la MCU PIC que está utilizando para determinar las opciones de WRT para el dispositivo.

## Características de funcionamiento

### Control de clavijas para MCLR (MCLRE)

MCLRE es un único bit de configuración que reside en CONFIG1 .

MCLRE controla la función del pin MCLR /V<sub>pp</sub> .

MCLRE se ignora si la programación de bajo voltaje (establecida por el bit de configuración LVP ) no está habilitada.

Los ajustes para MCLRE son:

MCLRE	Ajuste	Función
1	EN	La función de pin es MCLR /V <sub>pp</sub> con pull-up débil interno habilitado
0	APAGADO	La función de pin es una entrada digital con pull-up interno controlado por WPUx

**Ejemplo: Dejar MCLR /V<sub>pp</sub> como pin RESET.**

**Haciendo MCLR /V<sub>pp</sub> un pin de entrada digital.**

### Temporizador de encendido (PWRTS)

PWRTE es un único bit de configuración que reside en CONFIG2 .

El tiempo de encendido proporciona un retraso nominal de 72 ms después de un reinicio de encendido o un reinicio de apagón para permitir que  $V_{dd}$  se estabilice. La activación o desactivación de este retraso está controlada por PWRTE .

Las opciones para PWRTE son:

PWRTE	Ajuste	Función de temporizador de encendido
0	EN	Desactivado
1	APAGADO	Activado

### Ejemplo de activación del temporizador de encendido

### Habilitación del temporizador de vigilancia (WDTE)

El ajuste de configuración de WDTE consta de dos bits individuales: WDTE1 y WDTE0 . El campo de bits WDTE reside en CONFIG1 .

WDTE permite que se produzca un REINICIO de MCU si el temporizador de vigilancia interno pasa de 0xFF a 0x00 antes de que la MCU pueda ejecutar una instrucción CLRWDT .

Hay cuatro opciones para los dos bits WDTE :

1. El restablecimiento del temporizador de vigilancia siempre está habilitado
2. El restablecimiento del temporizador de vigilancia siempre está deshabilitado
3. El restablecimiento del temporizador de vigilancia está habilitado cuando se está ejecutando pero deshabilitado cuando la MCU ingresa al modo SUSPENSIÓN
4. El restablecimiento del temporizador de vigilancia se controla en tiempo de ejecución mediante el bit SWDTEN del registro WDTCON

WDTE	Ajuste	Función de reinicio del perro guardián
1 1	EN	Siempre habilitado

0 0	APAGADO	Siempre deshabilitado
1 0	NSLEEP	Habilitado mientras está ACTIVO, deshabilitado cuando está en modo SLEEP
0 1	SBODEN	Controlado por bit SBOREN de registro PCON

### Desbordamiento de pila de hardware (STVREN)

STVREN es un único bit de configuración que reside en CONFIG2 .

El bit de configuración STVREN habilita o deshabilita un REINICIO en un desbordamiento o subdesbordamiento de la pila.

Un desbordamiento o subdesbordamiento de pila siempre establece el bit STKOVF o STKUNF en el registro PCON independientemente del valor de STVREN .

Las opciones para STVREN son:

STVREN	Ajuste	Función
1	EN	El REINICIO del flujo excesivo/insuficiente de la pila está HABILITADO
0	APAGADO	El REINICIO del flujo excesivo/insuficiente de la pila está DESHABILITADO

Los bits de configuración y los ajustes para dispositivos PIC16F1xxx individuales pueden ser diferentes. Consulte su hoja de datos para conocer los detalles de los bits de configuración de PIC MCU que está utilizando.

### DEPURACIÓN - Modo de depuración

El bit DEBUG en la palabra de configuración 2 es administrado automáticamente por MPLAB X IDE.

¡Para garantizar el correcto funcionamiento del dispositivo, este bit no debe modificarse!

**Resumen de los ajustes de bits de configuración para el MCU PIC® de rango medio mejorado mediante el compilador XC8**

A continuación se muestra un resumen de las directivas de bits de configuración PIC16F1xxx aceptadas por el compilador MPLAB® XC8 C de Microchip.

En la página "[Bits de configuración](#)" del tutorial mejorado de rango medio se presenta una descripción completa de estos ajustes de bits de configuración .

Algunos microcontroladores PIC® de rango medio mejorados <sup>pueden</sup> tener un conjunto diferente de bits de configuración.  
Consulte la hoja de datos de la MCU que está utilizando para obtener una lista completa.

**Sintaxis**

#pragma config CONFIG\_BIT\_NAME = CONFIG\_VALUE

CONFIG_BIT_NOMBRE	CONFIG_VALUE	Función de los bits de configuración
FOSC	INTOSC EXTRC HS XT LP ECH ECM ECL	<p>El oscilador interno es la fuente del reloj.</p> <p>El oscilador RC externo es la fuente del reloj .</p> <p>Cristal/ oscilador externo de semilla alta.</p> <p>Cristal/oscilador externo. Oscilador externo de baja potencia . / rango de frecuencia 0 - 0,5 MHz</p>
WDTE	ON OFF NSLEEP SWDTEN	<p>El temporizador de vigilancia (WDT) está deshabilitado.</p> <p>WDT está habilitado .</p> <p>WDT está habilitado cuando está en funcionamiento y</p>

		<p>deshabilitado cuando está en modo SLEEP.</p> <p>WDT controlado por el bit SWDTEN en el registro WDTCON.</p>
PWRTE	<b>ENCENDIDO APAGADO</b>	<p>Temporizador de encendido activado</p> <p>Temporizador de encendido desactivado</p>
MCLRE	<b>ENCENDIDO APAGADO</b>	<p>La función pin es MCLR La función pin es entrada digital</p>
PC	<b>ENCENDIDO APAGADO</b>	<p>La protección de código está habilitada</p> <p>La protección de código está deshabilitada</p>
DPC	<b>ENCENDIDO APAGADO</b>	<p>La protección del código de memoria de datos está activada</p> <p>La protección del código de memoria de datos está desactivada</p>
ABURRIDO	<b>ENCENDIDO APAGADO</b>	<p>Brown-Out Reset está habilitado</p> <p>Brown-Out Reset está deshabilitado</p>
CLKOUTEN		

	<b>ENCENDIDO APAGADO</b>	<p>La función CLKOUT está habilitada en el pin CLKOUT La función CLKOUT está deshabilitada. Función I/O u osc en el pin CLKOUT</p>
IESO	<b>ENCENDIDO APAGADO</b>	<p>El modo de cambio de reloj interno/externo está habilitado El modo de cambio de reloj interno/externo está deshabilitado</p>
FCMEN	<b>ENCENDIDO APAGADO</b>	<p>El monitor de reloj a prueba de fallas está habilitado El monitor de reloj a prueba de fallas está deshabilitado</p>
WRT	<b>ENCENDIDO APAGADO</b>	<p>La protección contra escritura automática de la memoria flash está habilitada La protección contra escritura automática de la memoria flash está desactivada</p>
PLÉN	<b>ENCENDIDO APAGADO</b>	<p>PLL interno 4X está habilitado PLL interno 4X está deshabilitado</p>

STVREN

**ENCENDIDO APAGADO**

El desbordamiento o subdesbordamiento de pila provocará un reinicio  
El desbordamiento o subdesbordamiento de pila NO provocará un reinicio

BORV

**LO HOLA**

Restablecimiento de voltaje Brown-Out - Punto de disparo bajo seleccionado  
Restablecimiento de voltaje Brown-Out - Punto de disparo alto seleccionado

LVP

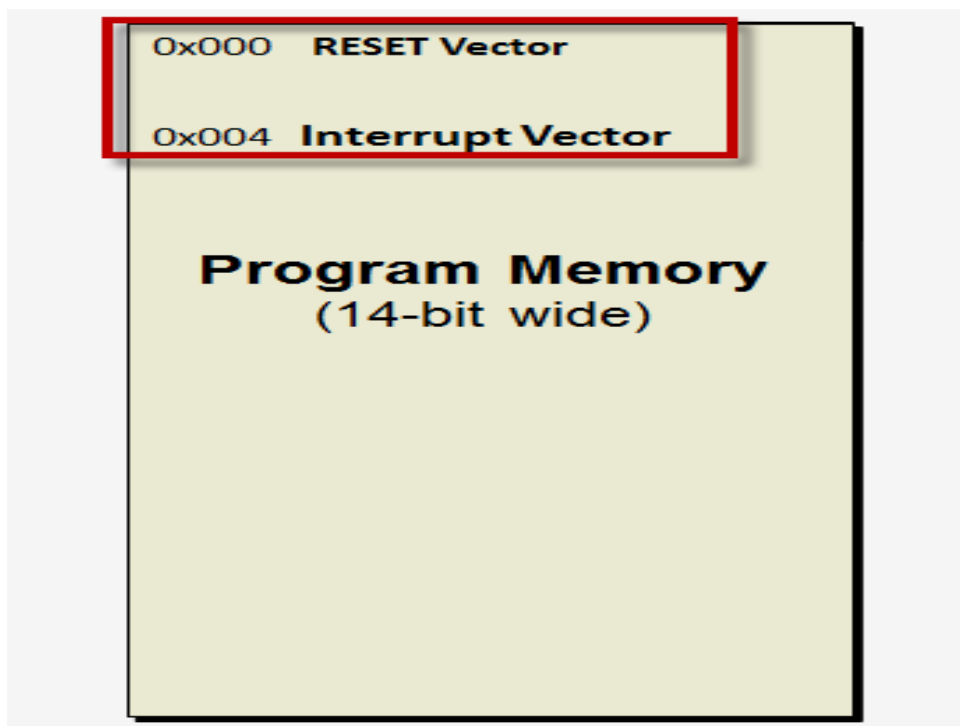
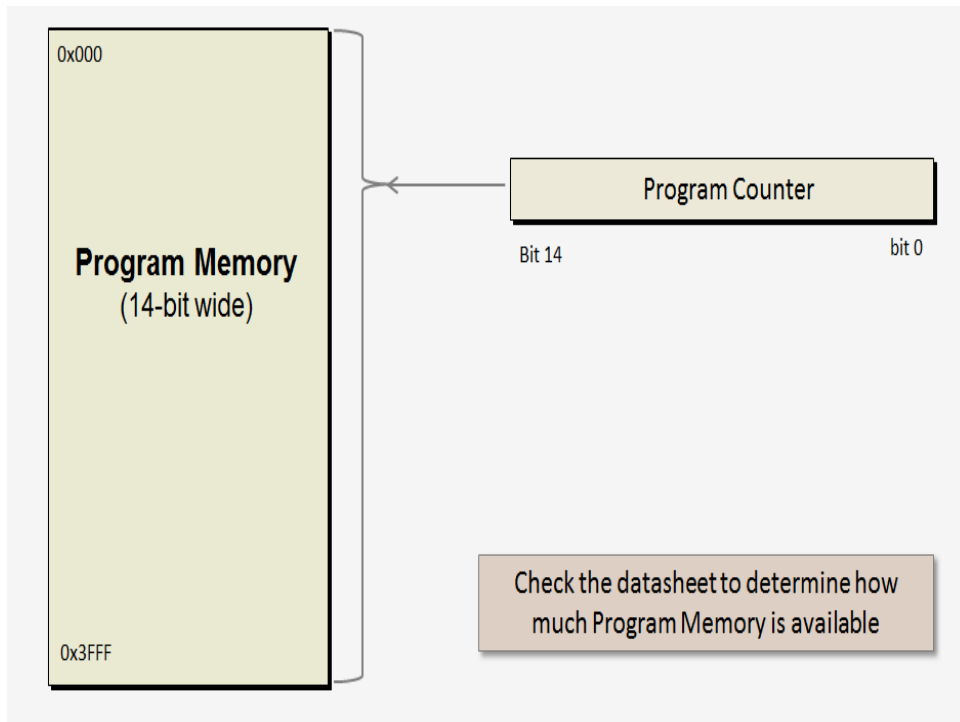
**ENCENDIDO APAGADO**

Programación de bajo voltaje habilitada  
Se debe usar alto voltaje en MCLR/VPP para la programación

### Memoria de programa MCU PIC® de gama media mejorada

La memoria de programa en la MCU PIC® de gama media mejorada consta de hasta 32 MB de memoria Flash de 14 bits de ancho. Después de programar la MCU, la memoria del programa contiene el código de aplicación del usuario. Se accede a la memoria de programa mediante un registro de contador de programa (PC) de 15 bits.





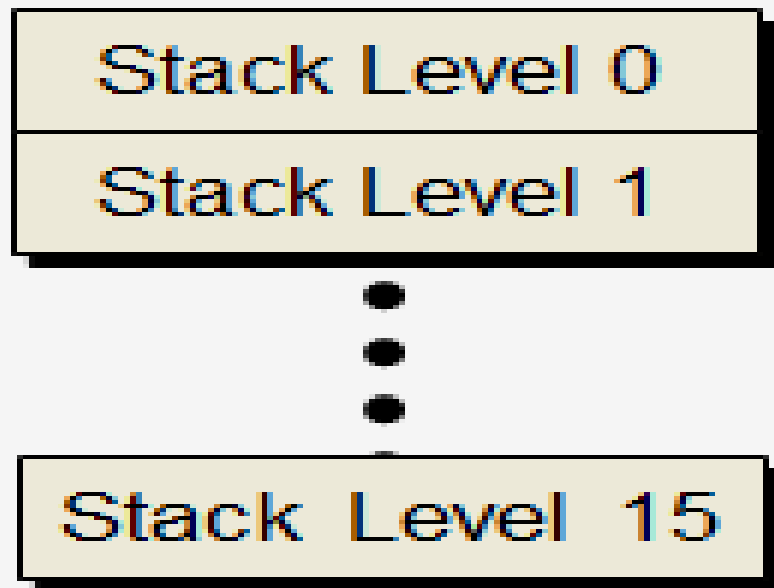
### **REINICIO vectorial**

En RESET, el contador del programa se borra, dando como resultado todo ceros. Esto permite que la dirección de memoria de programa 0h sea la ubicación de la primera instrucción ejecutada después de una condición de REINICIO.

### **Vector de interrupción**

Cuando ocurre una interrupción, el control del programa se transfiere a la dirección 04h. La sección ["Interrupciones"](#) proporciona una descripción completa del proceso de interrupción

# Return Stack



## Pila de retorno

Una pila de retorno de hardware de 16 entradas y 15 bits de ancho almacena la PC en caso de una interrupción o llamada a una subrutina. La pila de retorno funciona sobre la base de que el último en entrar es el primero en salir.

Al ejecutar una instrucción RETURN, ( RETFIE o RETURN ), el elemento superior de la pila se elimina de la pila y se coloca en el contador del programa.

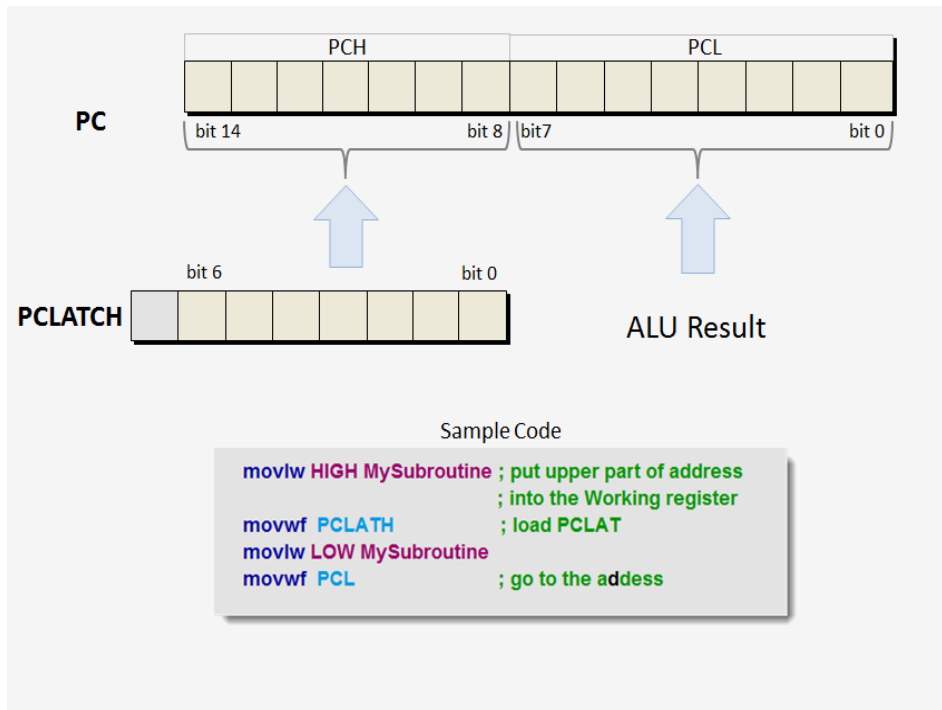
Para leer o modificar la PC de 14 bits con una MCU de 8 bits, se utilizan dos registros de funciones especiales (SFR):

- PCL : contiene los 8 bits inferiores del contador de programa <PC7:0>
- PCLATH : el contenido depende de la operación de MCU que se esté realizando

PCL y PCLATH se utilizan cuando el programa escribe en la PC, lee el contador del programa o ejecuta una instrucción GOTO o CALL .

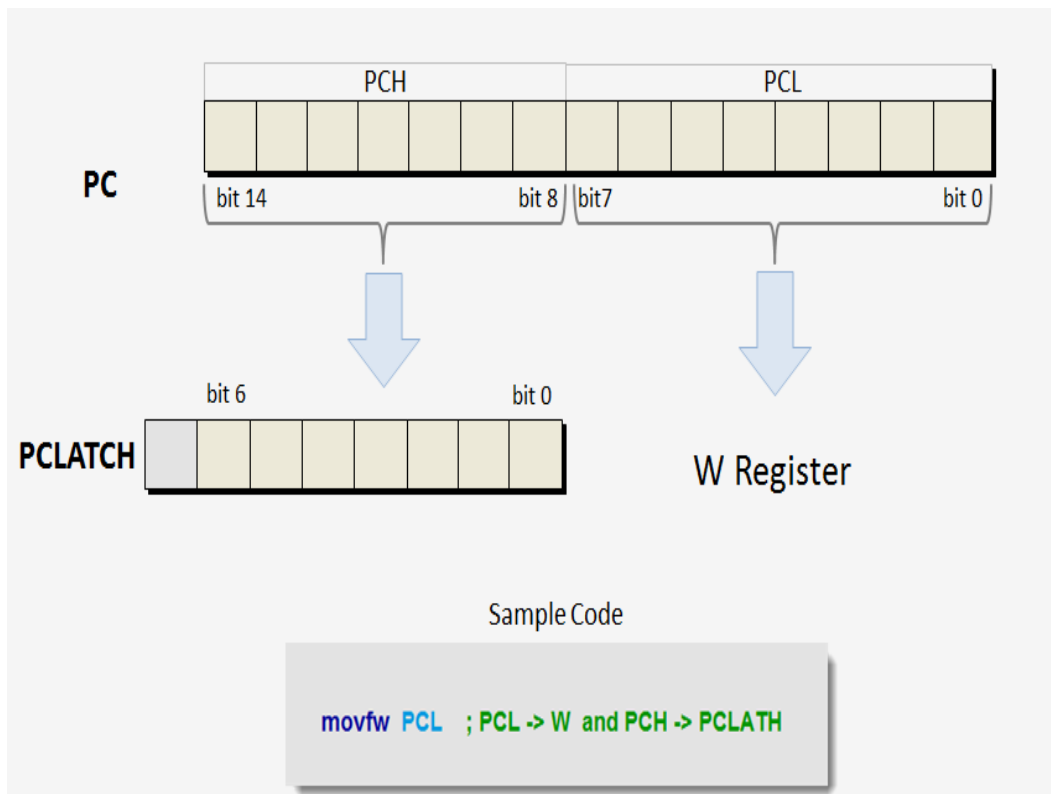
## Escribir en la PC

Cuando la aplicación escribe en PCL , el contenido actual de PCLATH<5:0> se escribirá en PC<14:8> . Debido a esto, el contenido de PCLATH<5:0> siempre DEBE ser correcto ANTES de escribir en PCL .



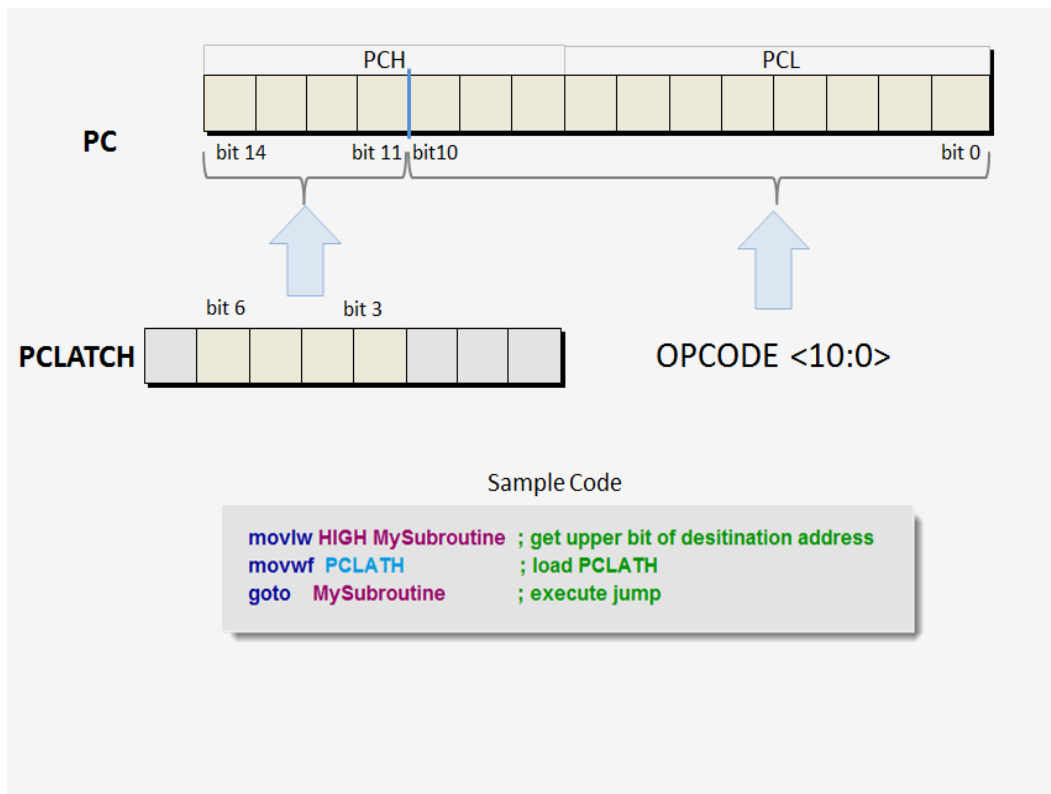
## leyendo la computadora

Cuando una aplicación lee PCL , PC<14:8> se captura en PCLATH .



### Ejecución de una instrucción CALL o GOTO

Las instrucciones CALL y GOTO solo tienen 11 bits disponibles para especificar la dirección de destino. PCLATH se usa para extender el operando para acceder a todas las direcciones de memoria del programa. Cuando se ejecuta CALL o GOTO, la dirección de 11 bits del operando se carga en PC<10:0> y PCLATH<6:3> se carga en PC<14:11>.



## Periféricos de la familia de microcontroladores <sup>PIC®</sup> de gama media mejorada de 8 bits

Módulo	Nombre
<a href="#">ADC</a>	Convertidor analógico a digital de 10 bits
<a href="#">CVX</a>	Celda lógica configurable
<a href="#">CMP</a>	Comparadores analógicos
<a href="#">CPS</a>	Módulo de detección capacitiva
<a href="#">CAD</a>	Convertidor digital a analógico de 5 bits
<a href="#">DSM</a>	Modulador de señal de datos

<a href="#"><u>ECCP</u></a>	Captura, comparación y PWM mejorados
<a href="#"><u>EUSART</u></a>	Transmisor receptor asíncrono universal mejorado
<a href="#"><u>RVF</u></a>	Referencia de voltaje fijo
<a href="#"><u>MI2C</u></a>	Circuito Interintegrado Maestro (I <sup>2</sup> C)
<a href="#"><u>MSSP</u></a>	Puerto serie síncrono maestro (SPI/I <sup>2</sup> C)
<a href="#"><u>suboficial</u></a>	Oscilador controlado numéricamente
<a href="#"><u>SPI</u></a>	Interfaz de periféricos en serie (SPI)
<a href="#"><u>TMR0</u></a>	Temporizador 0 (8 bits)
<a href="#"><u>TMR1</u></a>	Temporizador 1 (16 bits)
<a href="#"><u>TMR2</u></a>	Temporizador 2 (8 bits)
<a href="#"><u>UART</u></a>	Transmisor receptor asíncrono universal

## E/S digital de gama media mejorada de 8 bits

### Estructura básica de E/S

Casi todos los pines del microcontrolador PIC<sup>®</sup> de rango medio mejorado (MCU) se pueden usar como pines de entrada o salida digital. Los pines digitales comparten estos atributos:

- Capacidad para monitorear entradas digitales
- Señales de salida digital de control
- Dominadas débiles internas
- Multiplexado con periféricos

- Alta capacidad de accionamiento (hasta 25 mA disipador/fuente en muchos pines de E/S)
- Manipulación directa de bit de ciclo único
- Diodos de protección ESD de 4 kV

En REINICIO:

- Los pines digitales vuelven a la entrada (Hi-Z)
- Los pines con capacidad analógica vuelven a ser analógicos

Las E/S digitales están controladas por software en la MCU. El programa MCU configura, lee y envía los valores a los pines digitales.

## **Puertos de E/S**

### **Puertos de E/S**

Figura 1. Puerto de E/S.

Los pines de E/S digitales individuales se combinan en grupos llamados puertos. Los puertos de E/S contienen hasta ocho pines digitales. Se puede acceder a los pines de E/S digital individualmente pin por pin. También se puede acceder a todos los miembros de un puerto de E/S particular en un ciclo de instrucción mediante una de las instrucciones de acceso a bytes de la MCU.

Los puertos de E/S se denominan por letras (p. ej., PORTA, PORTB, PORTC). El número de puertos de E/S variará según el MCU de PIC que se utilice. Consulte la hoja de datos individual para determinar las asignaciones de PUERTO para una MCU PIC.

## Pin típico de E/S digital

Figura 2. E/S de pin digital.

Hay cinco registros disponibles para configurar y controlar los pines de E/S digitales.

1. TRISx: establece la dirección como entrada o salida.
2. ANSELx: determina si un pin con capacidad analógica funciona como entrada analógica o E/S digital.
3. LATx: se utiliza para generar valores para un pin digital.
4. PORTx- Lee el valor de entrada de un pin digital.
5. WPUx- Habilita el pull up débil interno.

Hay cinco registros de control de E/S para cada puerto.

- Para el puerto A los registros de control son: TRISA, ANSELA, LATA, PORTA y WPUA.
- Para el puerto B los registros de control son: TRISB, ANSELB, LATB, PORTB y WPUB.

A lo largo de esta página se proporcionan ejemplos de programas que muestran cómo utilizar los registros de control de E/S.

### Identificación de los pines de E/S en la hoja de datos

Las E/S digitales pueden compartir pines con otros periféricos y líneas de control de MCU. Algunos pines de E/S digitales tienen capacidad analógica y se pueden configurar para



operar como pines de entrada analógica. Consulte el diagrama de pines de la hoja de datos del dispositivo para determinar qué pines están disponibles como E/S digital.

Los pines digitales se identifican mediante tres identificadores secuenciales:

- El primer identificador de un pin digital es la letra R.
- El segundo identificador es una letra del PUERTO en el que está asociado el pin (como A para PORTA , B para PORTB , etc.).
- El identificador final es un número del 0 al 7 que indica la posición en el PUERTO que ocupa el PIN.

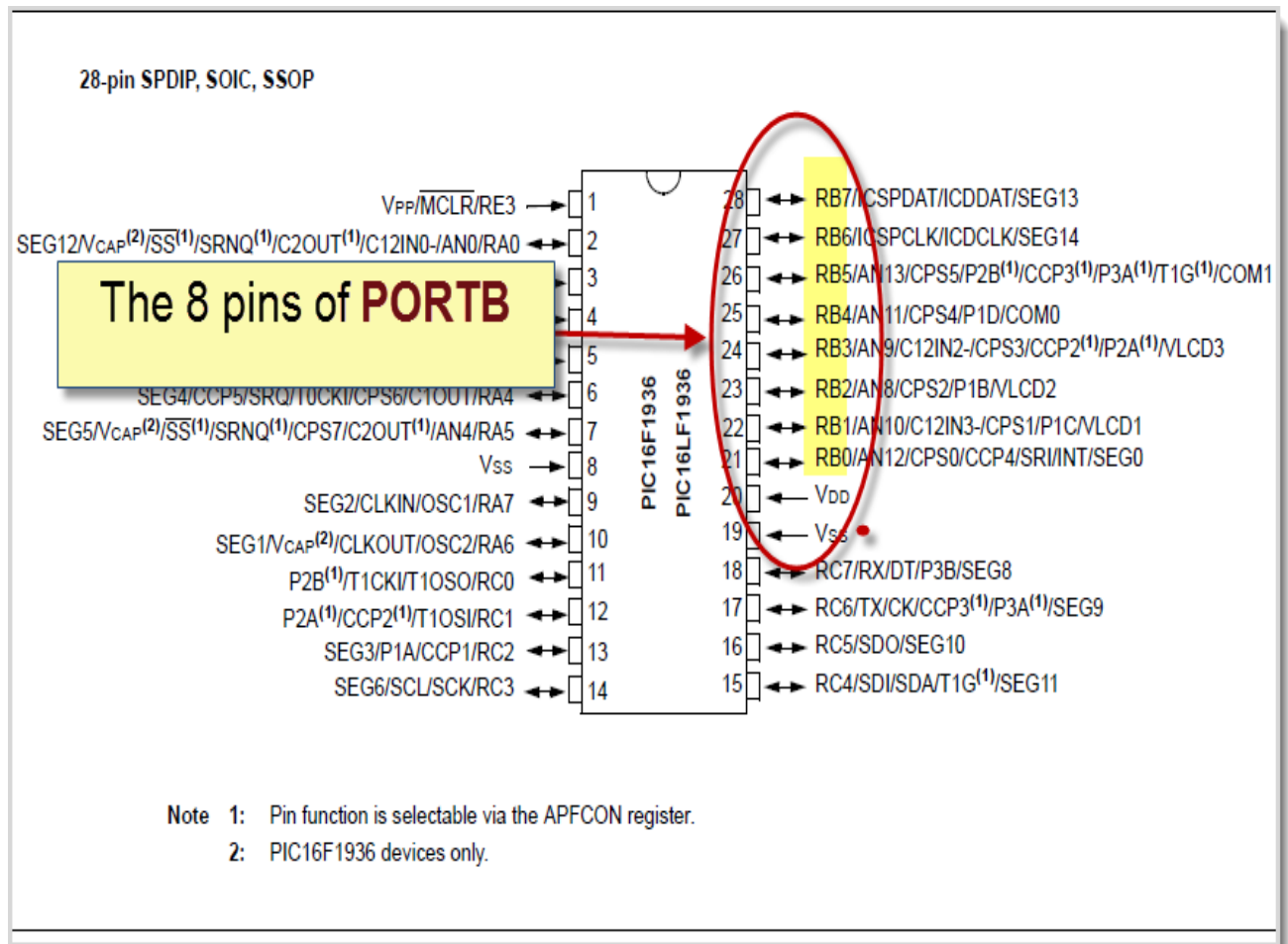


Figura 3. Todos los pines digitales asociados con PORTB están resaltados en amarillo.

Los miembros de PORTB van desde RB0 - RB7.

Para los otros pines en la MCU:

- Los pines con capacidad analógica se designan con las dos letras AN seguidas de un número.
- Los pines periféricos y de control de MCU están designados por nombre en la hoja de datos.

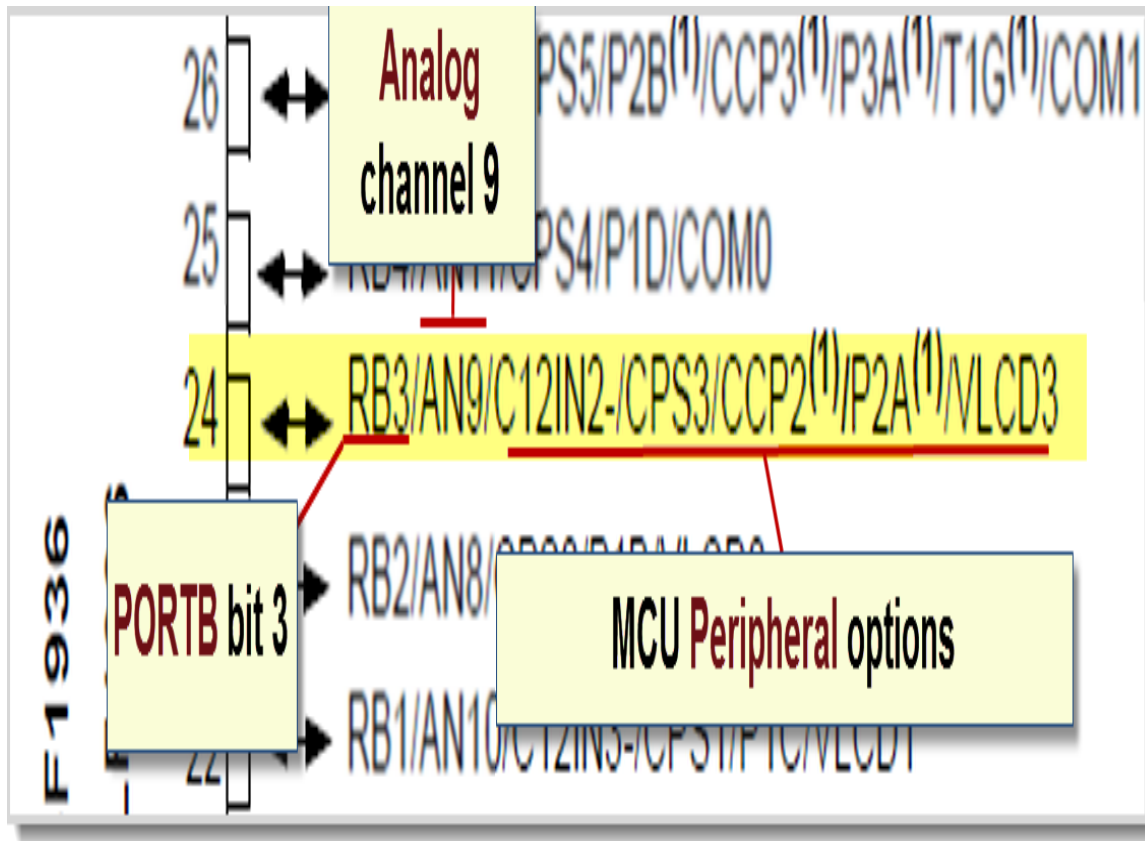


Figura 4. En el primer plano del RB3/pin 24, se muestran las opciones para el PIC16L/F1936.

El pin 24 se puede configurar como pin digital PORTB bit 3, canal analógico 9 o uno de varios periféricos

Para que un pin funcione como un pin digital, todos los periféricos asociados con el pin NO deben estar habilitados.

Si se va a utilizar un pin con capacidad analógica como pin digital, además de no habilitar el periférico, el pin *debe* configurarse específicamente como un pin digital.

### Configuración analógica vs digital

Dependiendo de qué MCU PIC de rango medio mejorado se utilice, se pueden configurar hasta 30 pines digitales para que sean pines analógicos. Los registros ANSELx se utilizan para configurar el modo de los pines con capacidad analógica. (ANSELA controla el modo de todos los pines con capacidad analógica en PORTA , ANSELB controla el modo para PORTB , ANSELD para PORTD , etc.).

Los pines con capacidad analógica en el puerto se asignan a bits individuales en un registro ANSELx. Un valor de 1 en un bit de un registro ANSELx habilitará el modo analógico del pin del puerto correspondiente. Un valor de 0 configura el pin para que sea digital.

En RESET, todos los pines con capacidad analógica vuelven al modo analógico. La MCU establece todos los bits ANSELx relevantes en 1.

#### **Ejemplo:**

El pin 24, en la Figura 4, puede configurarse como canal analógico 9 o como pin digital RB3. Para usar este pin como pin digital, se debe borrar el bit 3 de ANSELB.

Cuando trabaje con pines digitales con capacidad analógica, recuerde:

- Si se intenta una conversión de analógico a digital (ADC) en un pin con capacidad analógica configurado como digital, la conversión devolverá un valor invariable que no refleja el voltaje en el pin.
- Si un pin está configurado como un pin analógico, cualquier valor digital leído del pin siempre será 1, independientemente del voltaje en el panel de entrada.
- Si se escribe un pin analógico configurado como si fuera un pin digital, el nivel de salida del pin no cambiará.

#### **Salidas digitales**

El registro TRISx controla la dirección de los datos en cada bit de un puerto. Cada pin en un puerto se asigna a un bit en un registro TRIS. La dirección de datos para cada pin se puede configurar escribiendo un valor de 8 bits en el registro TRIS individualmente, configurando/borrando un bit de registro TRIS o la dirección de todos los bits en un puerto.

En RESET, todos los bits asociados con los pines en los registros TRIS se establecen en 1, lo que hace que todos los pines sean entradas High-Z.

#### **Convertir un pin en un pin de salida digital**

Para configurar un pin como salida digital, se deben colocar 0 en los bits de registro TRISx correspondientes.

#### **Escribir en un pin digital**

El valor de salida para cada puerto se puede cargar escribiendo en el registro LAT del puerto. (Al igual que todos los demás registros de control de puertos, los nombres de los registros LATx están escritos con letras. Los registros LATx comienzan con LATA y continúan con LATB, LATC...)

Escribir un 1 a un bit en el registro LAT conducirá el pin a Vdd. Un 0 en un bit LAT llevará el pin a Vss.

Escribir en el registro POTRTx también impulsará la señal de salida al igual que escribir en el registro LATx.

Sin embargo, bajo cargas altas o alta frecuencia, si se escriben secuencialmente varios comandos de modificación de bits (BSF, BCF) en un registro de PUERTO de salida, es posible que la última instrucción de manipulación de bits sobrescriba una instrucción anterior, dando como resultado un valor incorrecto en el puerto de salida. Para evitar la posibilidad de que esto ocurra, se recomienda encarecidamente que la salida siempre se realice en el registro LATx.

### **Código de ejemplo:**

El código que se muestra aquí es un ejemplo de configuración de todos los pines en PORTB como salidas digitales. Una vez configurados como pines de salida, los cuatro bits inferiores del puerto se elevan mientras que los cuatro bits superiores se establecen en 0.

¿Por qué se borró el registro LAT antes de configurar el registro TRIS?

En RESET se desconoce el contenido de los registros LATx. Se recomienda que el valor de todos los bits de registro LAT de salida se establezca en un valor conocido (y seguro) antes de habilitar los pines de salida. Esto evitará pulsos de salida espurios e involuntarios.

En este ejemplo, solo la configuración de RB3 se cambia a una salida digital. Una vez configurado, RB3 se eleva. Todos los demás pines (y sus respectivos bits de registro de control) se dejan sin cambios.

## **Entradas digitales**

### **Convertir un pin en un pin de entrada digital:**

Para configurar un pin como entrada digital, se deben colocar 1 en los bits de registro TRISx correspondientes.

### **Lectura de un pin digital**

El valor de cada pin de entrada se puede observar leyendo el registro PORTx correspondiente. Si el nivel de voltaje en un pin en particular está por encima del umbral de entrada, el bit en el registro PORTx asociado con el pin contendrá un 1. Los voltajes por debajo del umbral contendrán un 0.

Escribir en un bit de registro LATx de un pin configurado como entrada no afectará el valor del pin.

#### **Código de ejemplo:**

Este programa de muestra configura RA3 ( PORTA bit 3) como un pin de entrada digital. Luego, el programa monitorea continuamente el valor de RA3. Cuando RA3 sube, el control del programa se transfiere a una rutina de excepción.

#### **Dominadas débiles internas**

Las resistencias pull-up débiles internas están habilitadas para cada pad de E/S mediante el registro WPUx.

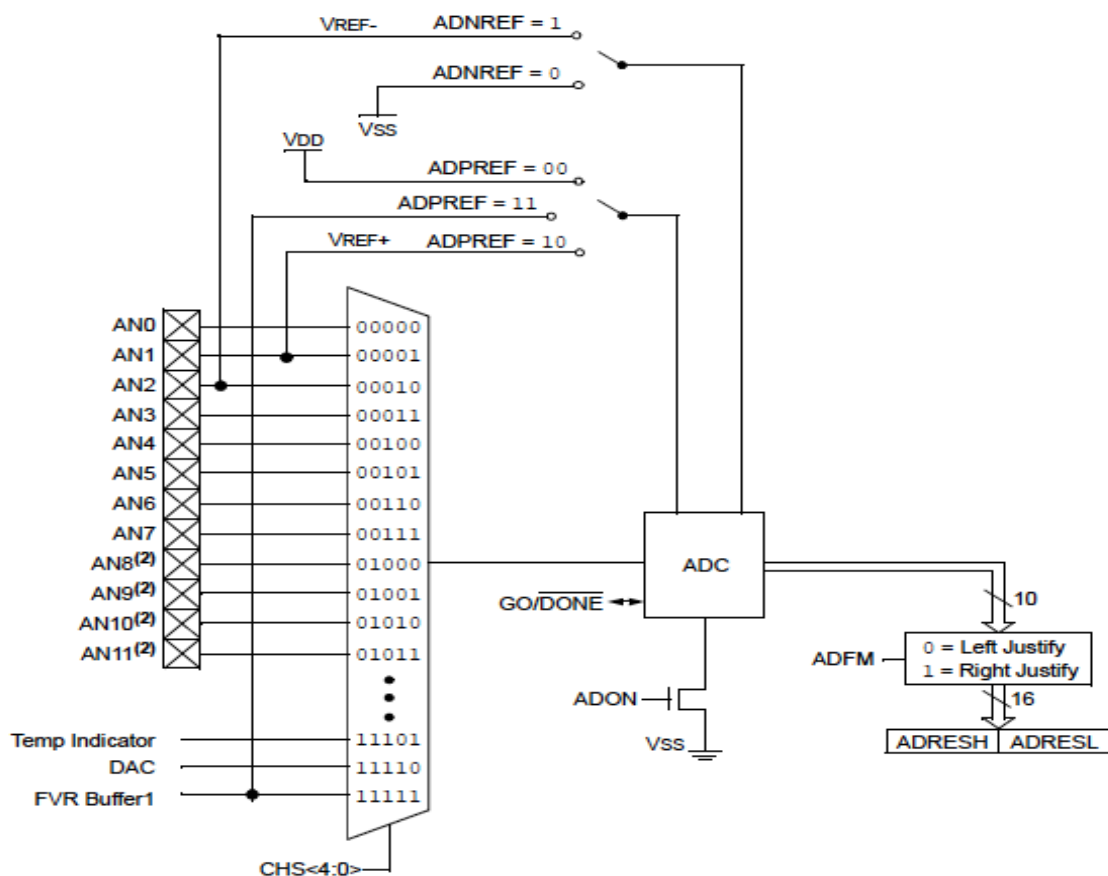
El pull-up se desactivará automáticamente cuando TRIS se configure como una salida o el pin se configure como una entrada analógica. Estos cambios en TRIS y ANSEL anularán la configuración de WPU

#### **Habilitación de pull-ups**

El pull-up interno para un pin de entrada digital se habilita escribiendo un 1 en el registro WPUx apropiado. Escribir un 0 en el registro WPUx deshabilita el pull-up.

### Convertidor analógico a digital (ADC)

El convertidor de analógico a digital (ADC) puede convertir una señal de entrada analógica en una representación digital binaria de 10 bits de esa señal. Las entradas analógicas de los microcontroladores de Microchip, que se multiplexan en un solo circuito de muestra y retención. La salida de la muestra y retención está conectada a la entrada del ADC. El ADC genera el resultado binario de 10 bits a través de aproximaciones sucesivas y almacena el resultado de la conversión en los registros de resultados del ADC.



El ADC utiliza una referencia de voltaje que es seleccionable por software para generarse internamente o suministrarse externamente.

El ADC también puede generar una interrupción al finalizar una conversión. Esta interrupción se puede utilizar para despertar el dispositivo de SLEEP.

### Configuración ADC

Cuando el ADC se configura por primera vez, debe tener varios ajustes de configuración habilitados. Éstos incluyen:

- Configuración del puerto ADC
- Selección de canal ADC
- Selección de referencia de voltaje ADC
- Fuente de reloj de conversión ADC
- control de interrupción ADC
- formato de resultado ADC

Veremos paso a paso cada uno de estos a continuación.

---

### Configuración del puerto ADC

El primer ajuste de configuración es la configuración de pines de E/S. La mayoría de los pines de E/S del ADC se pueden usar como entrada analógica o entrada digital. Al convertir señales analógicas usando el ADC, el pin de E/S debe configurarse para entrada analógica configurando los bits asociados en el registro TRIS y el registro ANSEL.

El registro TRIS para el pin de E/S debe tener su bit asociado establecido en 1 para convertirlo en una entrada. Si el pin de E/S es parte del bloque PORTA, el registro TRISA contiene el bit.

### REGISTER 6-4: TRISA: PORTA TRI-STATE REGISTER

RW-1/1	RW-1/1	RW-1/1	RW-1/1	RW-1/1	RW-1/1	RW-1/1	RW-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

El siguiente paso es establecer el bit en el registro ANSEL para el pin de E/S y establecer el bit en 1 para habilitar el ADC en ese pin. Si el pin de E/S es parte del bloque PORTA, el registro ANSELA contiene el bit.

## REGISTER 6-5: ANELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

### Selección de canal ADC

El multiplexor ADC debe conectarse al pin de E/S antes de iniciar el proceso de muestreo y retención. Esto se hace con un conjunto de bits en el registro ADCON0. Antes de que se solicite una muestra de ADC, estos bits de selección de canal se configuran para conectarse al pin de E/S deseado. Solo se puede conectar un pin al ADC a la vez. Una vez que se completa el proceso, los bits de selección se pueden cambiar para conectarse al siguiente pin y el proceso ADC comienza de nuevo.

## REGISTER 11-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0



bit 6-2      **CHS<4:0>**: Analog Channel Select bits

00000 = AN0  
 00001 = AN1  
 00010 = AN2  
 00011 = AN3  
 00100 = AN4  
 00101 = AN5  
 00110 = AN6  
 00111 = AN7  
 01000 = AN8<sup>(1)</sup>  
 01001 = AN9<sup>(1)</sup>  
 01010 = AN10<sup>(1)</sup>  
 01011 = AN11<sup>(1)</sup>  
 01100 = Reserved. No channel connected.  
 .  
 .  
 .  
 11100 = Reserved. No channel connected.  
 11101 = Temperature Indicator<sup>(4)</sup>  
 11110 = DAC output<sup>(2)</sup>  
 11111 = FVR (Fixed Voltage Reference) Buffer 1 Output<sup>(3)</sup>

Algunos dispositivos pueden tener menos canales

- Consulte la página ["Módulo convertidor de digital a analógico \(DAC\) \(MCU de 8 bits\)"](#) para obtener más información.
- Consulte la página ["Referencia de voltaje fijo \(FVR\)"](#) para obtener más información.
- Consulte la página ["Indicador de temperatura"](#) para obtener más información.

## Selección de referencia de voltaje ADC

El ADC puede usar varias fuentes de referencia de voltaje como base para las mediciones de voltaje analógico.

**Valor digital = [Tensión analógica / ( $V_{REF+} - V_{REF-}$ )] \* 1024**

Los bits ADPREF del registro ADCON1 proporcionan control de la referencia de voltaje positivo. La referencia de tensión positiva puede ser:

- $V_{REF+}$
- $VDD_{-}$
- Referencia de voltaje fijo (FVR)

Los bits ADNREF del registro ADCON1 proporcionan control de la referencia de voltaje negativo. La referencia de tensión negativa puede ser:

- $REF - V-$
- $V_{SS}$

$V_{DD}$  y  $V_{SS}$  son las conexiones al bus de voltaje que alimenta el dispositivo.

$V_{REF+}$  y  $V_{REF-}$  son pines de E/S específicos en el dispositivo. Una referencia de voltaje externo está conectada a estos pines.

FVR es una función en muchos dispositivos <sup>PIC®</sup>, aunque no en todos. Puede incluir un solo voltaje o, a veces, más de un nivel de voltaje está disponible.

Los bits de selección de referencia de voltaje están en el registro ADCON1 y las opciones de selección se muestran a continuación.

REGISTER 16-2: ADCON1: AD CONTROL REGISTER 1

RW-0/0	RW-0/0	RW-0/0	RW-0/0	U-0	RW-0/0	RW-0/0	RW-0/0
ADFM	ADCS<2:0>			-	ADNREF	ADPREF<1:0>	
bit 7							bit 0

bit 2	ADNREF: A/D Negative Voltage Reference Configuration bit
0	VREF- is connected to Vss
1	VREF- is connected to external VREF- pin <sup>(1)</sup>
bit 1-0	ADPREF<1:0>: A/D Positive Voltage Reference Configuration bits
00	VREF+ is connected to VDD
01	Reserved
10	VREF+ is connected to external VREF+ pin <sup>(1)</sup>
11	VREF+ is connected to internal Fixed Voltage Reference (FVR) module <sup>(1)</sup>

---

#### Fuente de reloj de conversión ADC

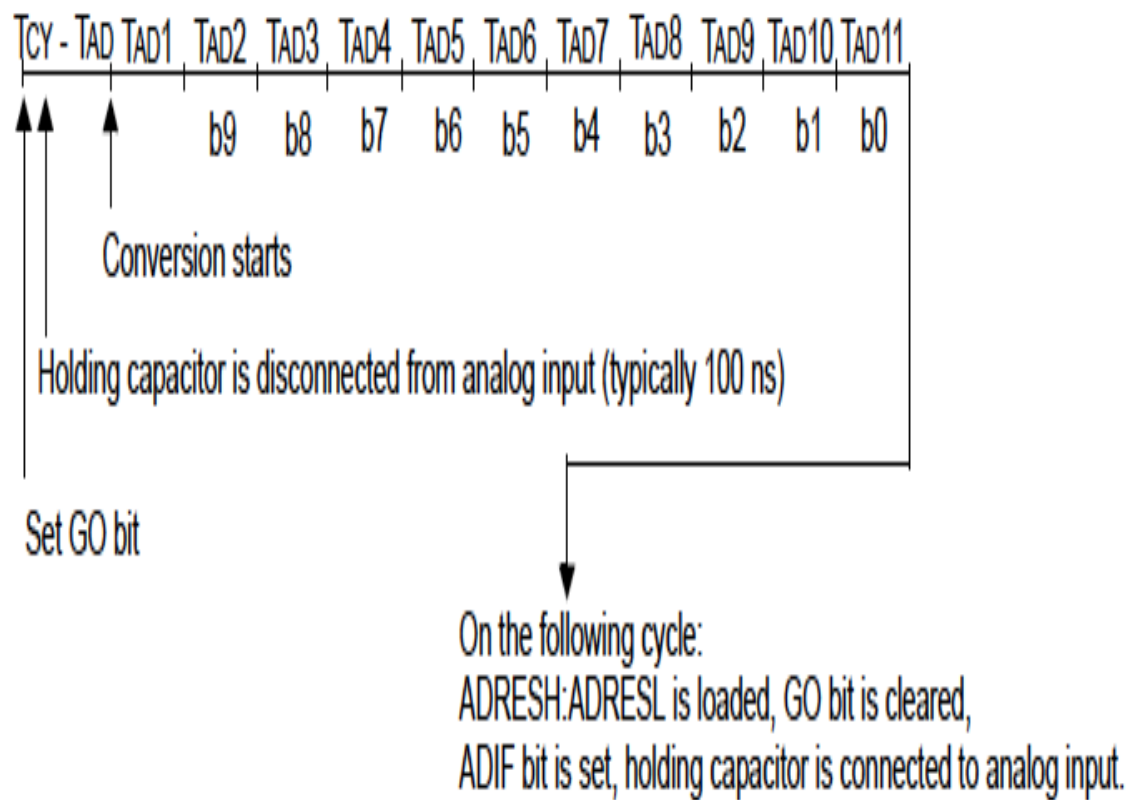
La fuente del reloj de conversión es seleccionable por software a través de los bits ADCS del registro ADCON1. Hay hasta siete opciones de reloj posibles según el dispositivo que se utilice:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (oscilador interno dedicado)

$F_{OSC}$  es el oscilador del sistema que ejecuta el reloj de instrucciones del dispositivo.

El reloj es fundamental para producir la conversión analógica a digital más rápida pero también precisa.

El tiempo para completar la conversión de un bit se define como  $T_{AD}$ . Una conversión completa de 10 bits requiere períodos de  $11,5 T_{AD}$ , como se muestra aquí:



Para una conversión correcta, se debe cumplir con la especificación  $T_{AD}$  adecuada. Un reloj ADC se puede seleccionar fácilmente de la tabla a continuación. Aparece un gráfico similar en la hoja de datos del dispositivo. Los mejores valores se muestran en el medio del gráfico con un fondo blanco.

ADC Clock Period (T <sub>AD</sub> )		Device Frequency (F <sub>osc</sub> )					
ADC Clock Source	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
F <sub>osc</sub> /2	000	62.5 ns <sup>(2)</sup>	100 ns <sup>(2)</sup>	125 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs
F <sub>osc</sub> /4	100	125 ns <sup>(2)</sup>	200 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 μs	4.0 μs
F <sub>osc</sub> /8	001	0.5 μs <sup>(2)</sup>	400 ns <sup>(2)</sup>	0.5 μs <sup>(2)</sup>	1.0 μs	2.0 μs	8.0 μs <sup>(3)</sup>
F <sub>osc</sub> /16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs <sup>(3)</sup>
F <sub>osc</sub> /32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>
F <sub>osc</sub> /64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>
F <sub>RC</sub>	x11	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>

Legend: Shaded cells are outside of recommended range.

Note 1: The F<sub>RC</sub> source has a typical T<sub>AD</sub> time of 1.6 μs for V<sub>DD</sub>.

2: These values violate the minimum required T<sub>AD</sub> time.

3: For faster conversion times, the selection of another clock source is recommended.

4: The ADC clock period (T<sub>AD</sub>) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock F<sub>osc</sub>. However, the F<sub>RC</sub> clock source must be used when conversions are to be performed with the device in Sleep mode.

La selección de FRC del oscilador interno será una conversión más lenta pero garantizará que se cumplan los requisitos de T<sub>AD</sub>. El FRC también se puede usar en modo de suspensión para ejecutar mediciones de ADC.

## Control de interrupciones

El módulo ADC tiene la capacidad de generar una interrupción al finalizar una conversión de analógico a digital. Esta interrupción también se puede generar mientras el dispositivo está funcionando o mientras está en SUSPENSIÓN. Si el dispositivo está en SUSPENSIÓN, la interrupción activará el dispositivo y luego procesará la Rutina de servicio de interrupción (ISR) siempre que los bits de interrupción estén habilitados.

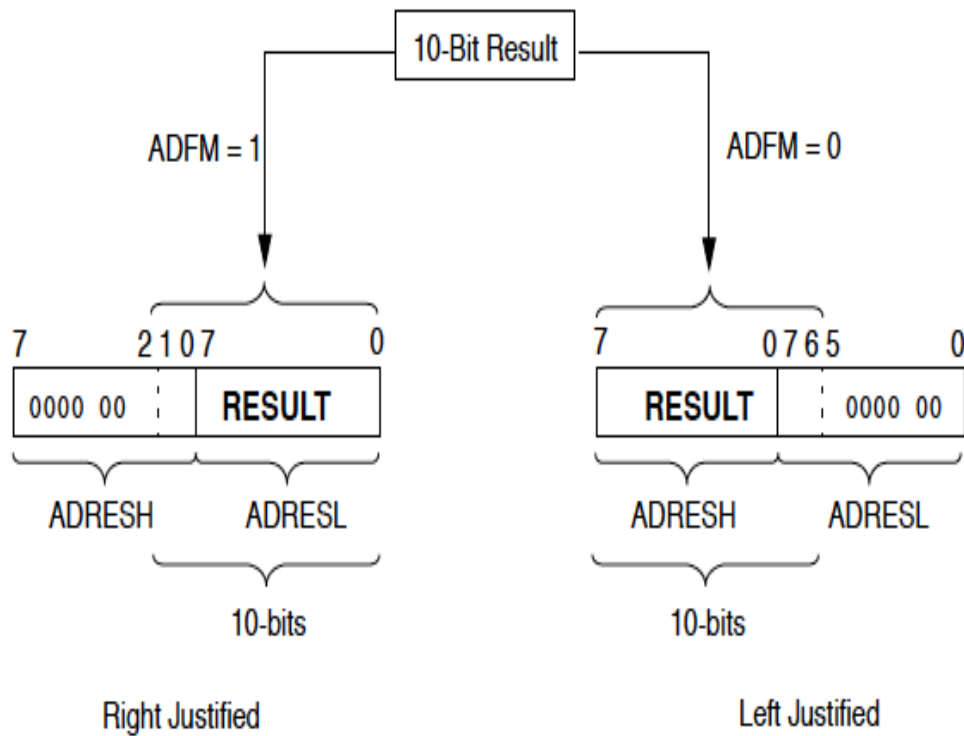
Esos bits de interrupción incluyen:

- El indicador de interrupción ADC es el bit ADIF en el registro 1 de interrupción periférica (PIR1).
- La habilitación de interrupción ADC es el bit ADIE en el registro de habilitación de interrupción periférica (PIE1).
- El bit de habilitación de interrupción global (GIE) y los bits de habilitación de interrupción periférica (PEIE) en el registro INTCON también deben estar habilitados.

Después de ejecutar una interrupción del modo SLEEP y completar ISR y ADC, el bit ADIF debe borrarse en el software.

### Formato de resultado ADC

El resultado de la conversión ADC se almacena en dos registros de 8 bits de ancho; ADRESH y ADRESL. Este par de registros tiene 16 bits de ancho, por lo que el módulo ADC tiene la flexibilidad de justificar a la izquierda o a la derecha el resultado de 10 bits en el registro de resultados de 16 bits. El bit de selección de formato ADC (ADFM) en el registro ADCON1 controla esta justificación. Los bits adicionales en los registros ADRESH y ADRESL se cargan con '0'.



bit 7      ADFM: A/D Result Format Select bit

1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.

0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.

Luego, el resultado puede copiarse en una variable o usarse en una ecuación para implementar una función basada en el resultado del ADC.

---

#### **Configurador de código MPLAB<sup>®</sup> - Configuración ADC**

[MPLAB<sup>®</sup> Code Configurator \(MCC\)](#) facilita la configuración del código ADC. Todos los ajustes de configuración descritos anteriormente se pueden configurar en una pantalla GUI simple dentro de MPLAB X IDE. La pantalla ADC se muestra aquí:

Generate Code (4)

<< Resources

ADC::ADC

InitializerDefault
+
x

Enable ADC ☒

Positive Vref: chip\_VDD

Negative Vref: chip\_VSS

Clock: Frc Sampling Freq. : -

Result Alignment: ☒ Left ☐ Right

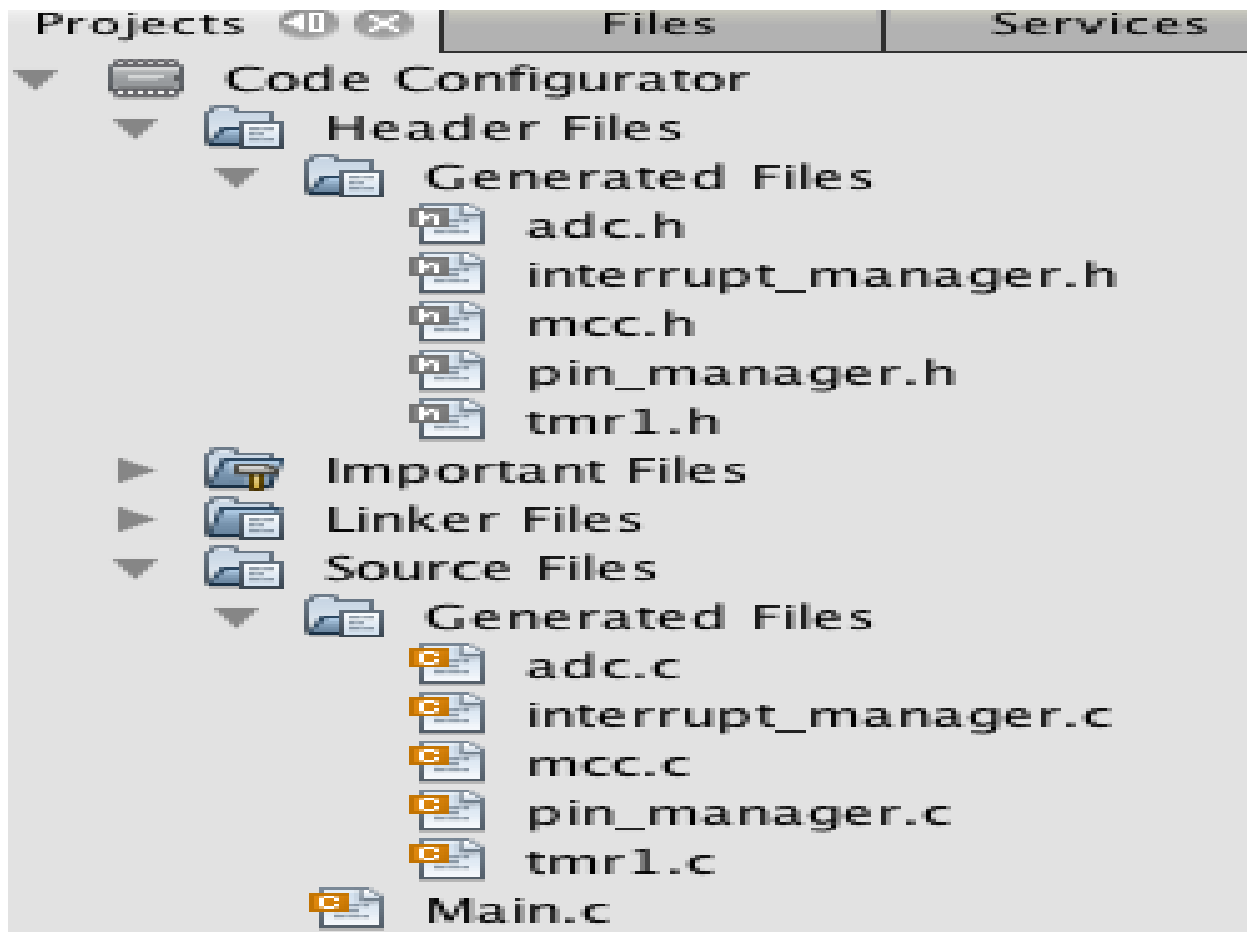
Enable ADC Interrupt ☒

Selected channels

Pin	Pin No.	Channel	Custom Name
			channel_Temp
			channel_DAC
			channel_FVR
RA2	11	AN2	channel_AN2

Cada opción de configuración se selecciona como una casilla de verificación o desde un menú desplegable. Una vez realizadas las selecciones, el código se genera y se coloca en el proyecto. Se crean dos archivos llamados `adc.h` y `adc.c`. Contienen código de configuración de ADC y también funciones personalizadas para usar el ADC dentro de los archivos de su proyecto principal.





### Ejemplo de proyecto ADC

Aquí hay un proyecto de ejemplo paso a paso para configurar el [ADC usando MCC](#).

