

# Bits de configuración PIC16F1xxx de rango medio mejorado de 8 bits

---

Los bits de configuración son una colección de datos binarios ubicados en la memoria flash de un microcontrolador PIC® (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	CLKOUTEN	BOREN1	BOREN0	CPD	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	PWRTÉ	WDTE1	WDTE0	FOSC2	FOSC1	FOSC0
bit 6						bit 0

(/local--files/8bit:emr-configuration-bits/config1.png)

#### 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	DEBUG	—	BORV	STVREN	PLLEN	—
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

(/local--files/8bit:emr-configuration-bits/config2.png)

Ajustes de configuración de la hoja de datos de PIC16F1937

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



```

1  #include <xc.h>
2
3  #config pragma FOSC = ECH
4  #config pragma WDTE = ON
5  #config pragma PWRTE = OFF
6  #config pragma MCLRE = ON
7  #config pragma CP = OFF
8  #config pragma BOREN = ON
9  #config pragma CLKOUTEN = OFF

```



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 (/8bit:emr-summary-configbit-in-c) .

**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 (/mplabx:view-and-set-configuration-bits) 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



```
1 #include <xc.h>
2 #pragma config FOSC = HS
```

### 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



```
1 #include <xc.h> ?
2 #pragma config FCMEN = ON // E
```

### Conmutación interna/externa (IES0)

El IES0 es un único bit de configuración que reside en CONFIG1.

IES0 establece el modo de cambio de reloj y arranque de dos velocidades. Con IES0 habilitado, la fuente del reloj puede controlarse mediante el programa de aplicación.

Las opciones para IES0 son:

IES0	Ajuste	Función de puesta en marcha de dos velocidades
0	APAGADO	Desactivado
1	EN	Activado

### Ejemplo: Habilitación del arranque de dos velocidades



```
1 #include <xc.h> ?
2 #pragma config IES0 = ON
```

### Habilitación de salida de reloj ( $\overline{\text{CLKOUTEN}}$ )

CLKOUTEN es un único bit de configuración que reside en  $\overline{\text{en}}$  CONFIG1.

$\overline{\text{CLKOUTEN}}$  permite que el pin OSCx/CLKOUT emita el reloj del sistema interno. Esto permite que el reloj del sistema PIC16F1xx controle otros componentes.

Las opciones para  $\overline{\text{CLKOUTEN}}$  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



```
1 #include <xc.h> ?
2 #pragma config CLKOUT = ON
```

## 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 OSCCON.

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



```
1 #include <xc.h> ?
2 #pragma config PLEN = ON
```

# 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



```
1 #include <xc.h>
2 #pragma config BOREN = OFF
```

### 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.



```
1 #include <xc.h>
2 #pragma config BOREN = ON
3 #pragma config BORV = HI
```

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



```
1 #include <xc.h>
2 #pragma config BOREN = NSLEEP
3 #pragma config BORV = L0
```

## 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  $V_{cap}$  en V.

Los ajustes para VCAPEN son:

VCAPEN	Ajuste	Función
0 0	RA6	RA6 se asigna como $V_{cap}$
0 1	RA5	RA5 se asigna como $V_{cap}$
1 0	RA0	RA0 se asigna como $V_{cap}$
1 1	APAGADO	La $V_{cap}$ 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



```
1 #include <xc.h>
2 #pragma config VCAPEN = RA6
```

## Deshabilitar la funcionalidad de $V_{cap}$ en V



```
1 #include <xc.h>
2 #pragma config VCAPEN = OFF
```

## 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 /Vpp`.

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



```
1 #include <xc.h> ?
2 #pragma config LVP = OFF
```

## 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



```
1 #include <xc.h> ?
2 #pragma config CPD = ON
```

### 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  $\overline{\text{CP}}$ .

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  $\overline{\text{CP}}$  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



```
1 #include <xc.h> ?
2 #pragma config CP = ON
```

### Activación de autoescritura flash (WRT)

$\text{WRT}$  consta de dos bits de configuración individuales:  $\text{WRT1}$  y  $\text{WRT0}$ . El campo de bits  $\text{WRT}$  se encuentra en  $\text{CONFIG2}$ .

$\text{WRT}$  establece el rango de direcciones en el que el programa de usuario puede escribir la memoria del programa.

Los ajustes para  $\text{WRT}$  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 $\overline{\text{MCLR}}$ /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  $\overline{\text{MCLR}}$  /V<sub>pp</sub> como pin RESET.**



```
1 #include <xc.h> ?
2 #pragma config MCLRE = ON
```

**Haciendo  $\overline{\text{MCLR}}$  /V<sub>pp</sub> un pin de entrada digital.**



```
1 #include <xc.h> ?
2 #pragma config MCLRE = OFF
```

## Temporizador de encendido (PWRTS)

PWRTS 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<sub>dd</sub> se estabilice. La activación o desactivación de este retraso está controlada por PWRTS.

Las opciones para PWRTS son:

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

## Ejemplo de activación del temporizador de encendido



```
1 #include <xc.h> ?
2 #pragma config PWRT = ON
```

## 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 CLRWD.

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.



```
1  #include <xc.h>           ?
2  #pragma config STVREN = ON
```

## 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!



Más información:

- Resumen de bits de configuración del lenguaje ensamblador (/8bit-i:emr-summary-configbit-in-assembly)
- Oscilador de 8 bits (/8bit:osc)
- Conmutación de reloj a prueba de fallas de 8 bits (/8bit:fscm)
- Cambio de reloj de 8 bits (/8bit:cswtch)
- PLL de 8 bits (/8bit:pll)
- Control de reinicio maestro de 8 bits (/8bit:mclr)
- REINICIO de oscurecimiento de 8 bits (/8bit:bor)
- REINICIO de oscurecimiento de 8 bits (/8bit:bor)
- Temporizador de encendido de 8 bits (/8bit:pwrst)
- Temporizador de vigilancia MCU de 8 bits (/8bit:wdt)
- Desbordamiento/subdesbordamiento de pila de 8 bits (/8bit:sof)