TECNICATURA SUPERIOR EN TELECOMUNICACIONES

ELECTRÓNICA MICROCONTROLADA

Docentes: Ing. Jorge E. Morales, Téc. Gonzalo Vera.

**Título:** Referencia de la gama media mejorada PIC.

**Grupo 1:**

❖ Birge, Adolfo Federico.

❖ Carunchio, Carlos Javier.

❖ Ferreyra, María Luciana.

❖ Gutiérrez, Emma Vilma.

❖ Merlo, Emmanuel.

❖ Romero, Gisela de Lourdes.

**Familia de gama media mejorada**

[**MCU PIC® de 8 bits de rango medio mejorado**](http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=1025&mid=10&lang=en&pageId=74)  
**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

[Diagrama

Descripción generada automáticamente](https://microchipdeveloper.com/local--files/8bit:emr/EnhancedMidRange.jpg)

*Haga clic en la imagen para ampliar.*

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.

**Entrenamiento a su propio ritmo**

El material de estos módulos de capacitación existe en otras partes de este sitio en un formato de referencia general. Sin embargo, los módulos de capacitación lo presentan en una secuencia organizada paso a paso para ayudarlo a aprender el tema desde cero.

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

# Resumen

### Familias de MCU de 8 bits de Microchip

Diagrama

Descripción generada automáticamente

La familia de MCU PIC® de 8 bits de rango medio mejorado PIC16F1xxx abarca 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.

Captura de pantalla de un celular con letras

Descripción generada automáticamente

### 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)

Diagrama

Descripción generada automáticamente

# 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.

Diagrama

Descripción generada automáticamente

​

La [**página de memoria**](https://microchipdeveloper.com/mcu1102:data-memory) 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 ® 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

Tabla

Descripción generada automáticamente

​

Para obtener una lista detallada del conjunto de instrucciones y una discusión completa del tiempo de instrucción, consulte la página [**Conjunto**](https://microchipdeveloper.com/mcu1102:instruction-set) de instrucciones del tutorial de rango medio mejorado.

# Opciones de reloj flexibles (hasta 32 MHz)

Seleccionado por los [**bits de configuración**](https://microchipdeveloper.com/mcu1102:configuration-bits)del PIC ® 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

Diagrama

Descripción generada automáticamente

​

Para obtener una descripción detallada de las opciones de configuración del oscilador, consulte la [**página del oscilador de 8 bits**](https://microchipdeveloper.com/8bit:osc) 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**

Diagrama, Esquemático

Descripción generada automáticamente

​

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**](https://microchipdeveloper.com/mcu1102:digital-io) /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.

Texto

Descripción generada automáticamente

​

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**](https://microchipdeveloper.com/mcu1102:peripherals) 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.

Diagrama

Descripción generada automáticamente

​

Para obtener una lista completa de los periféricos disponibles, consulte la [**sección Periféricos**](https://microchipdeveloper.com/mcu1102:peripherals) 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**](https://microchipdeveloper.com/mcu1102:interrupts) 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.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Conjunto de instrucciones de rango medio mejorado**

Esto se aplica a las familias de microcontroladores PIC ® 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 1fffffff | 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** | | | | | | |
| **Mnemónicos, Operandos** | | **Descripción** | **Ciclos** | **Código de operación de 14 bits MSb……LSb** | **Estado Afectado** | **notas** |
| 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** | | | | | | |
| **Mnemónicos, Operandos** | | **Descripción** | **Ciclos** | **Código de operación de 14 bits MSb……LSb** | **Estado Afectado** | **notas** |
| 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.

**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.

[Tabla

Descripción generada automáticamente](https://microchipdeveloper.com/local--files/8bit:emr-configuration-bits/config1.png)

[Tabla

Descripción generada automáticamente](https://microchipdeveloper.com/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 ® 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](https://microchipdeveloper.com/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](https://microchipdeveloper.com/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

​

### 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/CLCKOUT 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 (PLLEN)

PLLEN 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 PLLEN y el bit SPLLEN en el registro OSCON.

Las opciones para PLLEN son:

|  |  |  |
| --- | --- | --- |
| **PLÉN** | **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 tapa V |
| 0  1 | RA5 | RA5 se asigna como tapa V |
| 1  0 | RA0 | RA0 se asigna como tapa V |
| 1  1 | APAGADO | La tapa 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 pp .

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

Los ajustes para MCRLE son:

|  |  |  |
| --- | --- | --- |
| **MCLRE** | **Ajuste** | **Función** |
| 1 | EN | La función de pin es MCLR /V pp 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 MCRL /V pp como pin RESET.

​

### Haciendo MCRL /V pp un pin de entrada digital.

​

### Temporizador de encendido (PWRTE)

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!

​

Más información:

**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*](https://microchipdeveloper.com/mcu1102:configuration-bits) " 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 pueden 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** | El oscilador interno es la fuente del reloj. El oscilador RC externo es la fuente del reloj . Cristal/ oscilador externo de semilla alta. Cristal/oscilador externo. Oscilador externo de baja potencia . / rango de frecuencia 0 - 0,5 MHz |
| WDTE | **ON OFF NSLEEP SWDTEN** | El temporizador de vigilancia (WDT) está deshabilitado. WDT está habilitado . WDT está habilitado cuando está en funcionamiento y deshabilitado cuando está en modo SLEEP. WDT controlado por el bit SWDTEN en el registro WDTCON. |
| PWRTE | **ENCENDIDO APAGADO** | Temporizador de encendido activado Temporizador de encendido desactivado |
| MCLRE | **ENCENDIDO APAGADO** | La función pin es MCLR La función pin es entrada digital |
| PC | **ENCENDIDO APAGADO** | La protección de código está habilitada La protección de código está deshabilitada |
| DPC | **ENCENDIDO APAGADO** | La protección del código de memoria de datos está activada La protección del código de memoria de datos está desactivada |
| ABURRIDO | **ENCENDIDO APAGADO** | Brown-Out Reset está habilitado Brown-Out Reset está deshabilitado |
| CLKOUTEN | **ENCENDIDO APAGADO** | La función CLKOUT está habilitada en el pin CLKOUT La función CLKOUT está deshabilitada. Función I/O u osc en el pin CLKOUT |
| IESO | **ENCENDIDO APAGADO** | El modo de cambio de reloj interno/externo está habilitado El modo de cambio de reloj interno/externo está deshabilitado |
| FCMEN | **ENCENDIDO APAGADO** | El monitor de reloj a prueba de fallas está habilitado El monitor de reloj a prueba de fallas está deshabilitado |
| WRT | **ENCENDIDO APAGADO** | La protección contra escritura automática de la memoria flash está habilitada La protección contra escritura automática de la memoria flash está desactivada |
| PLÉN | **ENCENDIDO APAGADO** | PLL interno 4X está habilitado PLL interno 4X está deshabilitado |
| 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 |