



GSM/GPRS-GPS

Introducción a los Microcontroladores
PIC18 8-bit

Introducción Microcontroladores **PIC18**



MTI Jorge Gutiérrez
Github: @dave-dignal



www.dignal.com | cursos@dignal.com



Organización de la memoria



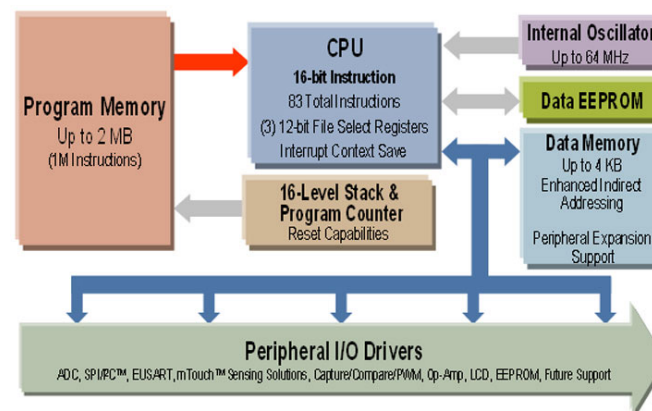
Organización de la Memoria

Organización de la memoria

Memoria de Programa: Almacena las instrucciones que gobiernan la aplicación. Es del tipo **no volátil**, es decir, las instrucciones se mantienen aunque desaparezca la alimentación.

Memoria de datos RAM: Almacena las variables y datos. Es del tipo **volátil**, es decir, los datos almacenados se borran cuando desaparece la alimentación.

Memoria EEPROM: Es un área de memoria de datos de lectura y escritura no volátil. Al perder la alimentación no hay pérdida de la información.

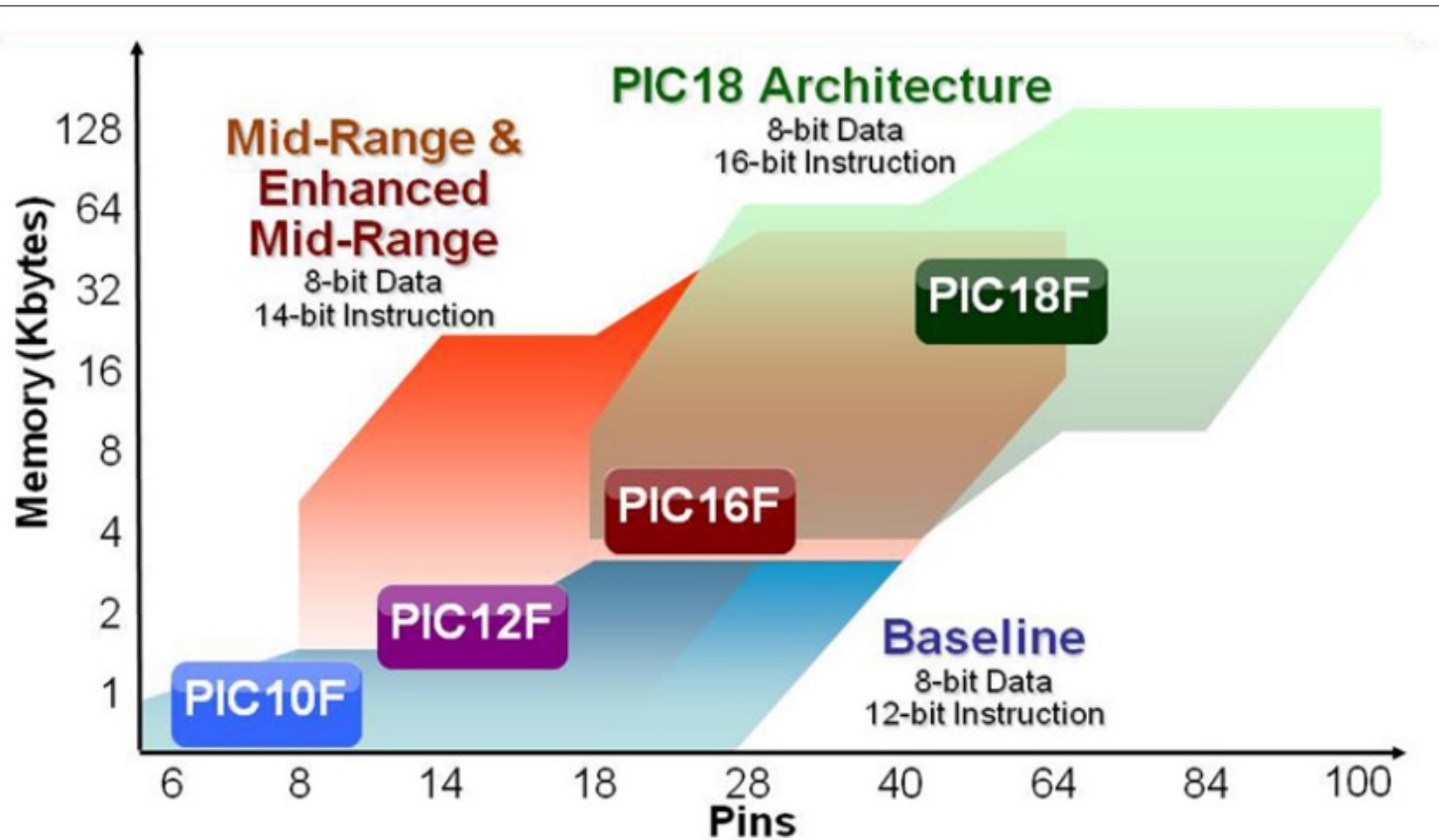


Arquitectura PIC18



Arquitectura PIC18

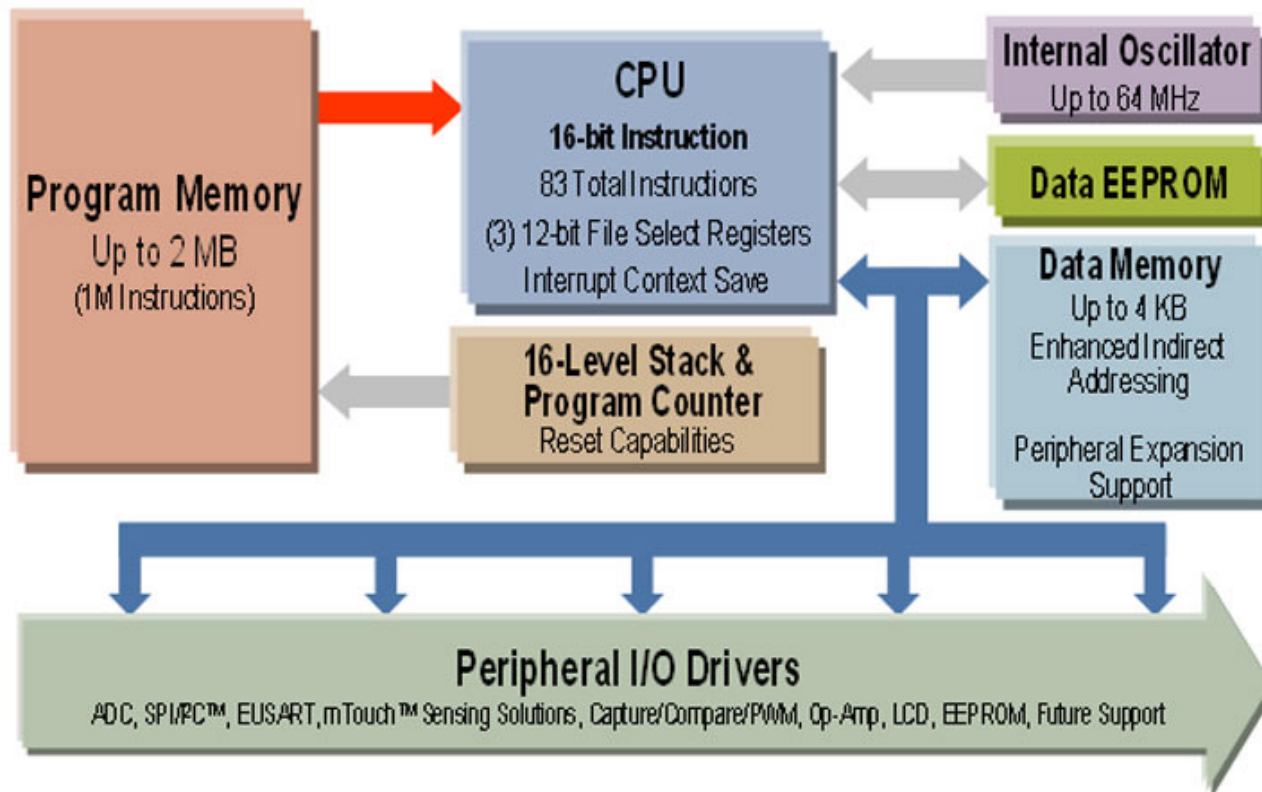
Arquitectura PIC18



Arquitectura PIC18

	Baseline Architecture	Mid-Range Architecture	Enhanced Mid-Range Architecture	PIC18 Architecture
Pin Count	6-40	8-64	8-64	18-100
Interrupts	No	Single interrupt capability	Single interrupt capability with hardware context save	Multiple interrupt capability with hardware context save
Performance	5 MIPS	5 MIPS	8 MIPS	Up to 16 MIPS
Instructions	33, 12-bit	35, 14-bit	49, 14-bit	83, 16-bit
Program Memory	Up to 3 KB	Up to 14 KB	Up to 28 KB	Up to 128 KB
Data Memory	Up to 138 Bytes	Up to 368 Bytes	Up to 1,5 KB	Up to 4 KB
Hardware Stack	2 level	8 level	16 level	32 level
Features	■ Comparator ■ 8-bit ADC ■ Data Memory ■ Internal Oscillator	In addition to Baseline: ■ SPI/I²C™ ■ UART ■ PWMs ■ LCD ■ 10-bit ADC ■ Op Amp	In addition to Mid-Range: ■ Multiple Communication Peripherals ■ Linear Programming Space ■ PWMs with Independent Time Base	In addition to Enhanced Mid-Range: ■ 8x8 Hardware Multiplier ■ CAN ■ CTMU ■ USB ■ Ethernet ■ 12-bit ADC
Highlights	Lowest cost in the smallest form factor	Optimal cost to performance ratio	Cost effective with more performance and memory	High performance, optimized for C programming, advanced peripherals

Arquitectura PIC18

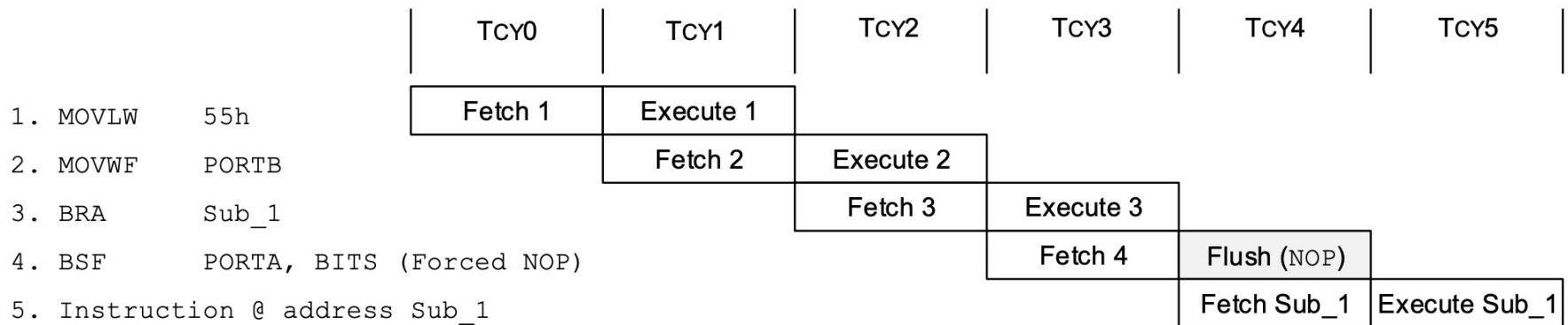


Flow/Pipelining

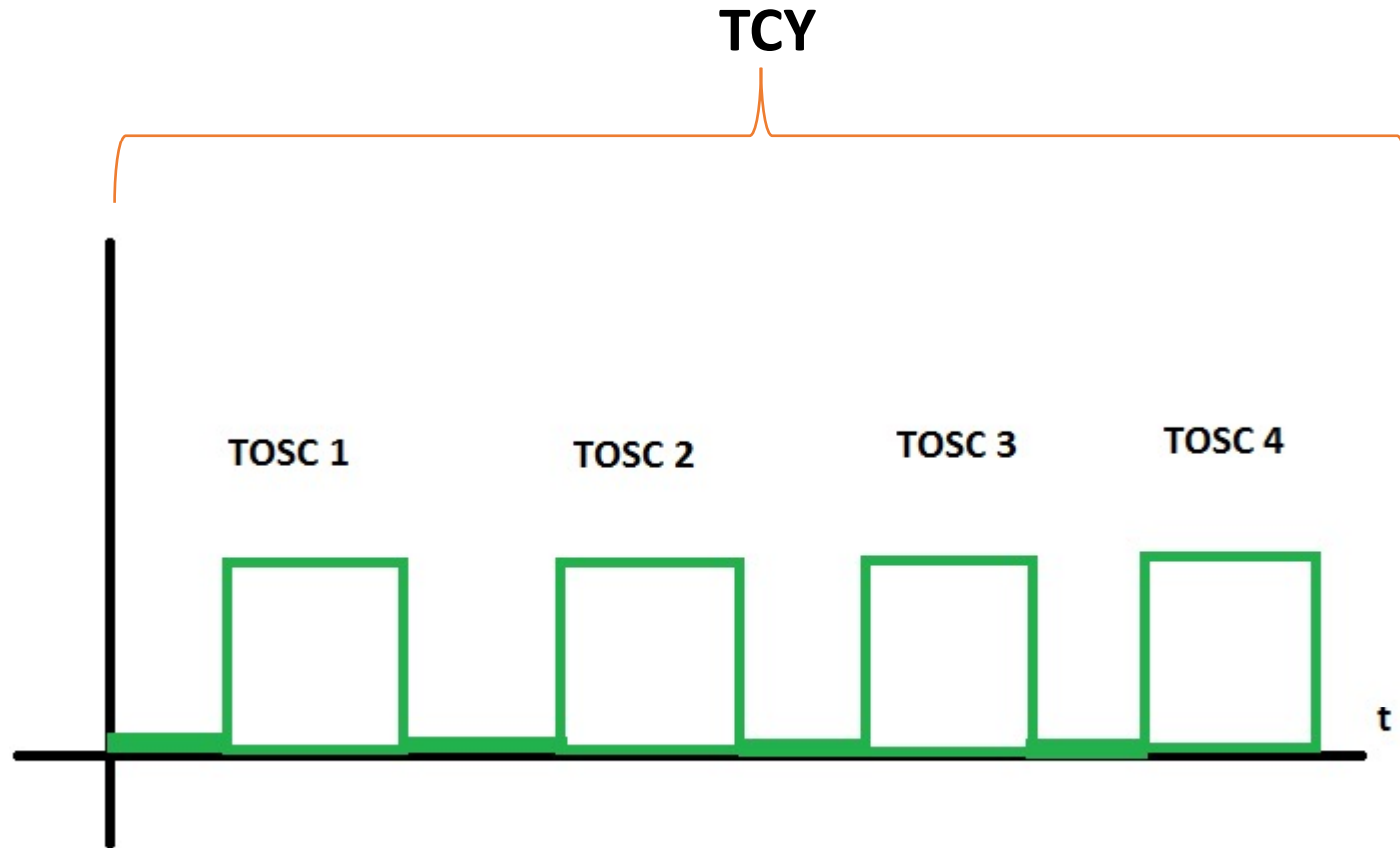


Flow/Pipelining

Un procesador segmentado o Pipeline realiza simultáneamente la ejecución de una instrucción y la búsqueda de código de la siguiente, de esta manera se puede ejecutar una instrucción en un ciclo.



Flow/Pipelining



Flow/Pipelining

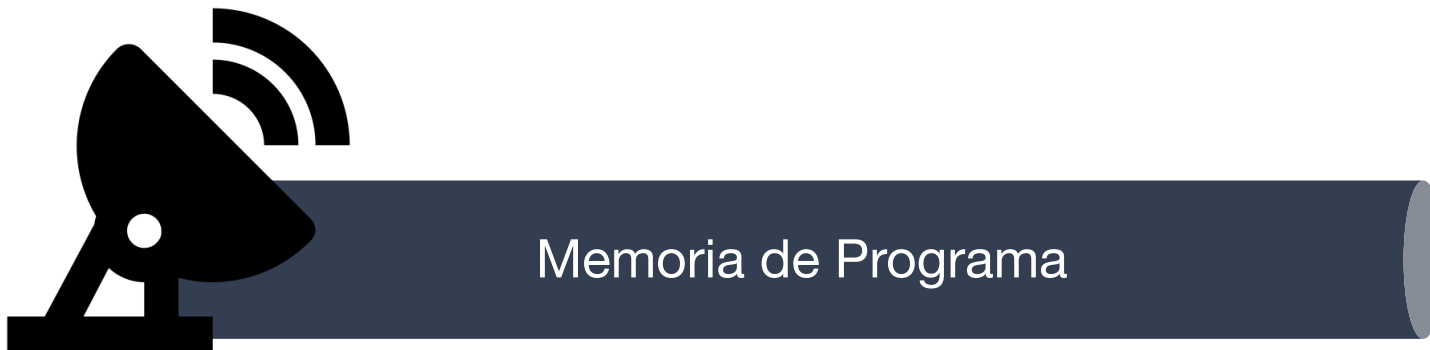
TCY

$$T_{osc} = \frac{1}{F_{osc}}$$
$$T_{osc} = \frac{1}{32Mhz} = 31.25nS$$

$$TCY = T_{osc} \times 4 = 125nS$$

$$TCY = \frac{4}{32Mhz} = 125nS$$

Memoria de Programa



Memoria de Programa

La memoria de programa se direcciona en bytes. Las instrucciones se almacenan como dos bytes o cuatro bytes en la memoria de programa. El byte menos significativo (LSB) de una palabra de instrucción siempre se almacena en una ubicación de memoria de programa con una dirección par (LSb = 0).

Word Address	LSB	MSB
000000h	000000h	000001h
000002h	000002h	000003h
000004h	000004h	000005h
000006h	000006h	000007h
000008h	000008h	000009h
00000Ah	00000Ah	00000Bh
00000Ch	00000Ch	00000Dh
00000Eh	00000Eh	00000Fh
000010h	000010h	000011h
000012h	000012h	000013h

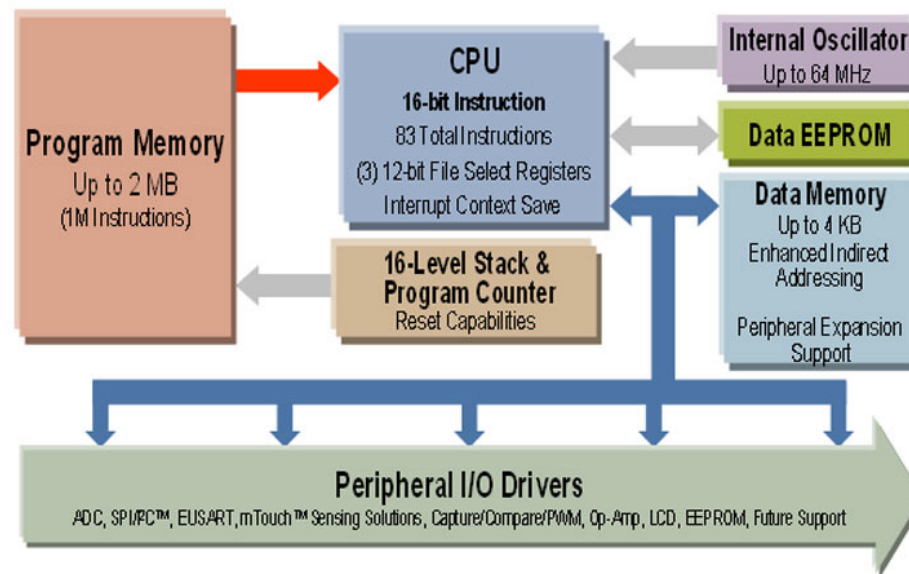
Memoria de Programa



Memoria de Datos RAM

Memoria de datos RAM

Memoria de datos RAM: Almacena las variables y datos. Es del tipo **volátil**, es decir, los datos almacenados se borran cuando desaparece la alimentación.



Memoria de datos RAM

La memoria de datos en los dispositivos PIC18 se implementa como RAM estática. Cada registro de la memoria de datos tiene una dirección de 12 bits, que permite direccionar hasta 4096 bytes de datos en memoria.

La memoria de datos contiene **registros de funciones especiales (SFR)** y **registros de propósito general (GPR)**.

- **SFR:** Se utilizan para el control/estado del microcontrolador y para el control y configuración de periféricos.
- **GPR:** Se utilizan para almacenamiento de los datos/variables de la aplicación del usuario.

SFR

Registros de Funciones
Especiales

GPR

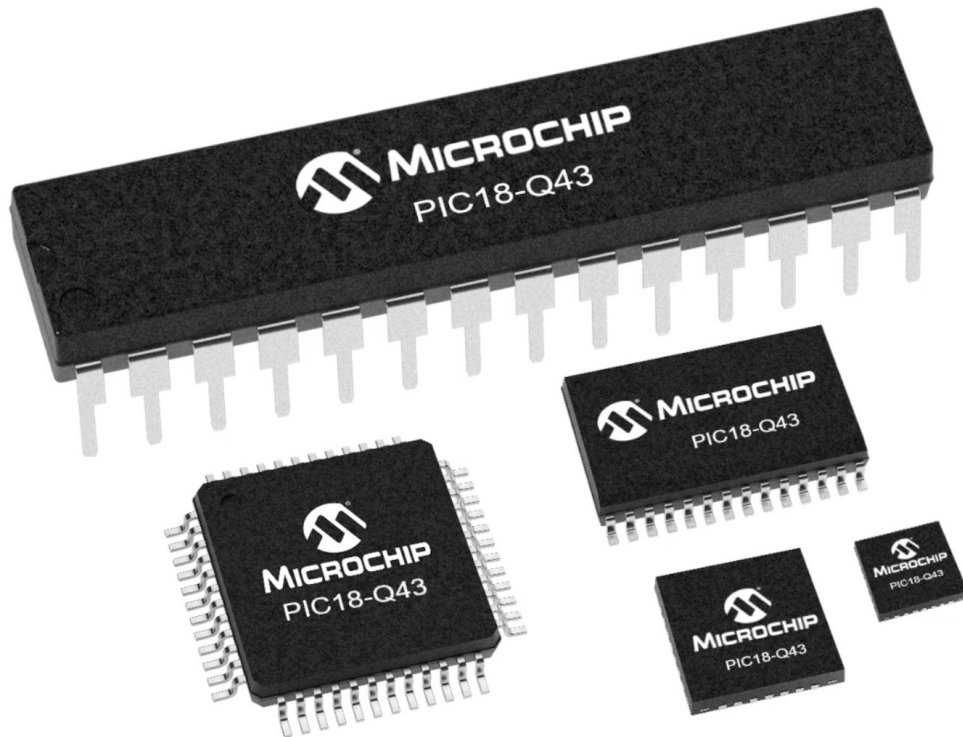
Registros de Propósito
General

PIC18F57Q43



PIC18F57Q43

PIC18F57Q43



Up to 48 pin packages

Up to 44 IO pins

As small as 4x4 mm

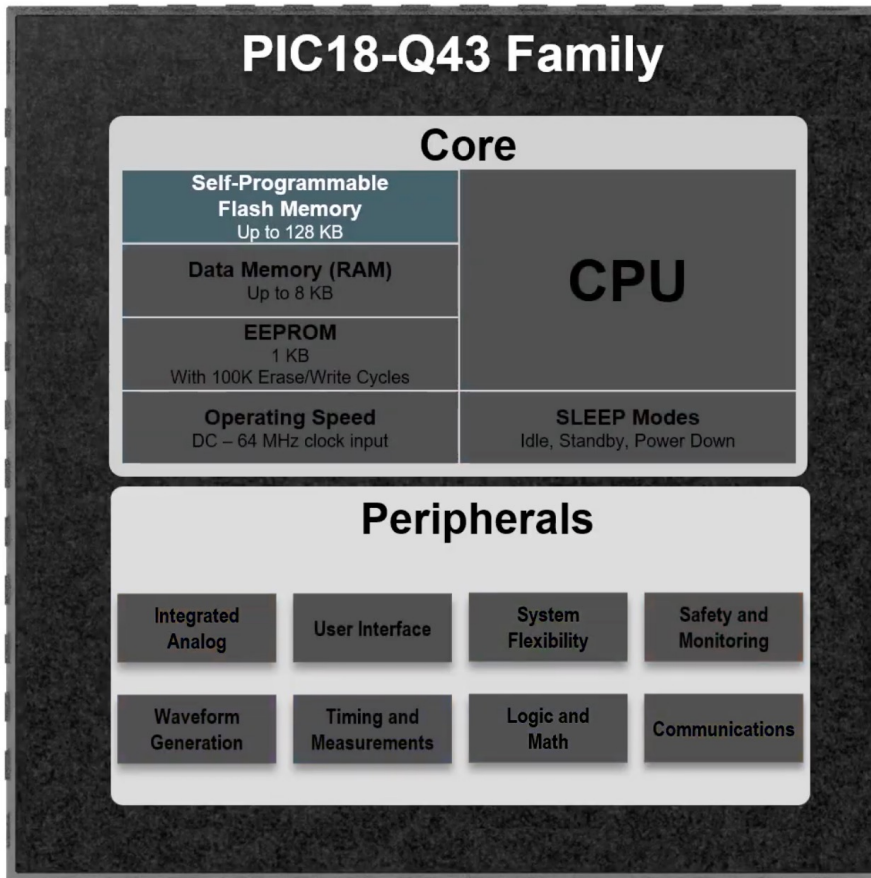
PDIP, SOIC, SSOP, TQFP, VQFN packages available

PIC18F57Q43

Características destacadas del PIC18F57Q43:

- 64 MHz de clock
- 127 niveles de hardware stack
- Vectored-Interrupt

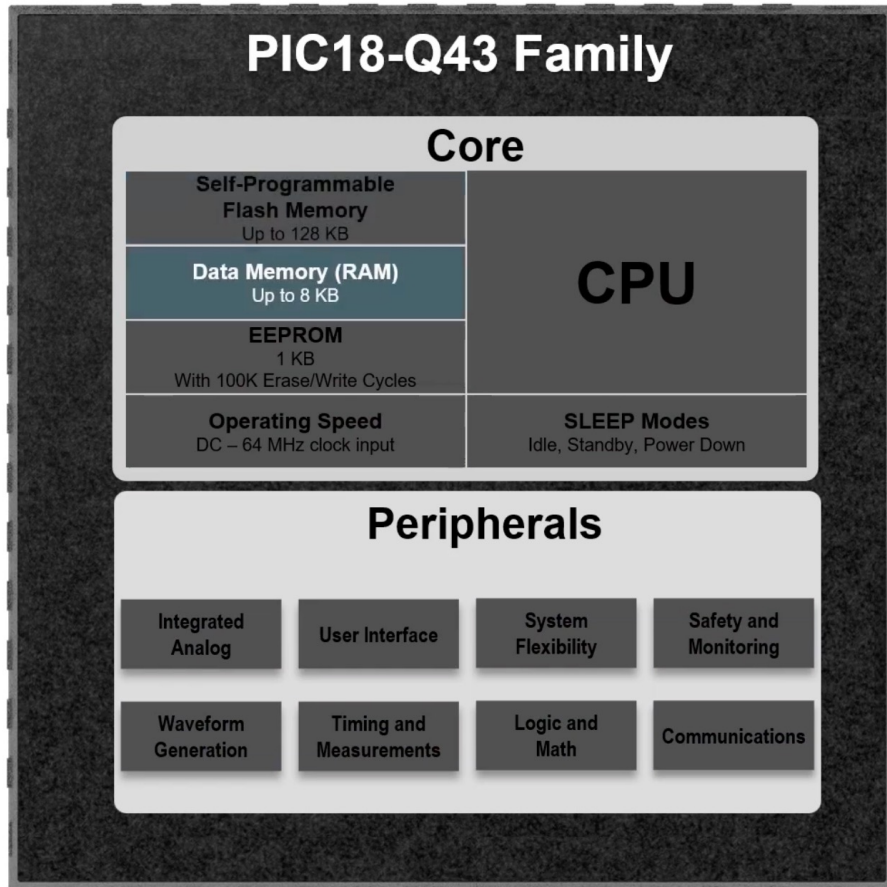
PIC18F57Q43



Memoria de Programa de hasta 128 KB

En la memoria de programa se almacenan las instrucciones del Microcontrolador.

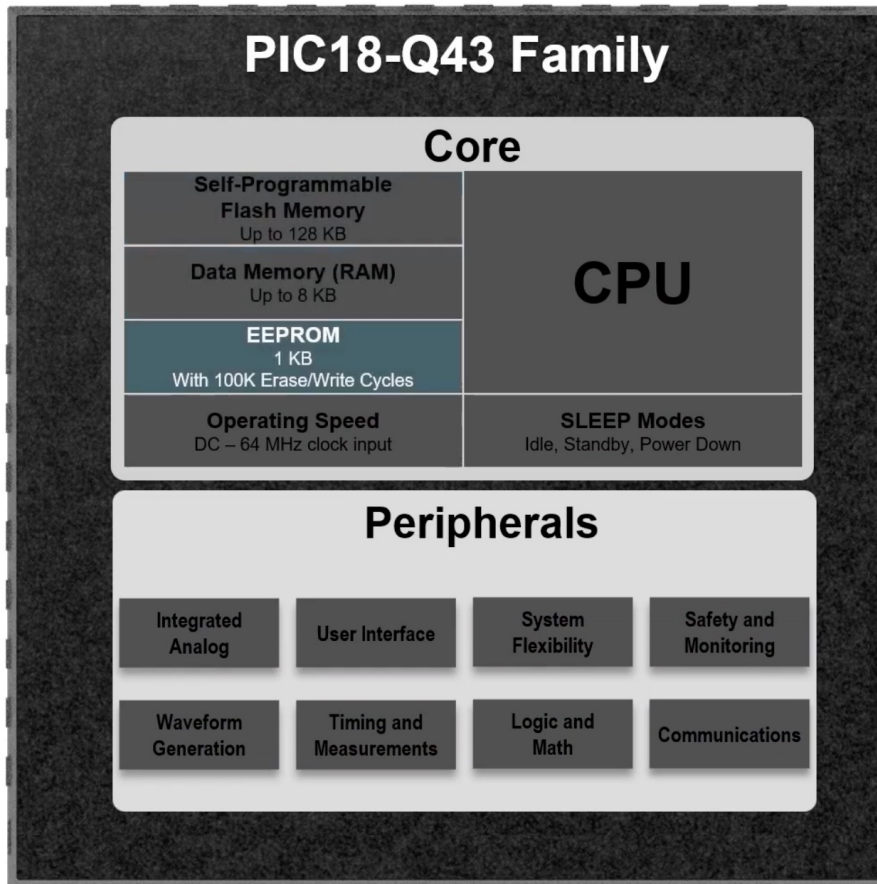
PIC18F57Q43



Memoria de Datos RAM

Memoria de datos de hasta 8 KB de espacio.

PIC18F57Q43



Memoria EEPROM

Memoria de datos EEPROM de hasta 1 KB de espacio con 100K ciclos de Borrado/Escritura.

PIC18F57Q43

PIC18-Q43 Family

Core

**Self-Programmable
Flash Memory**

Up to 128 KB

Data Memory (RAM)

Up to 8 KB

EEPROM

1 KB

With 100K Erase/Write Cycles

Operating Speed

DC – 64 MHz clock input

CPU

SLEEP Modes

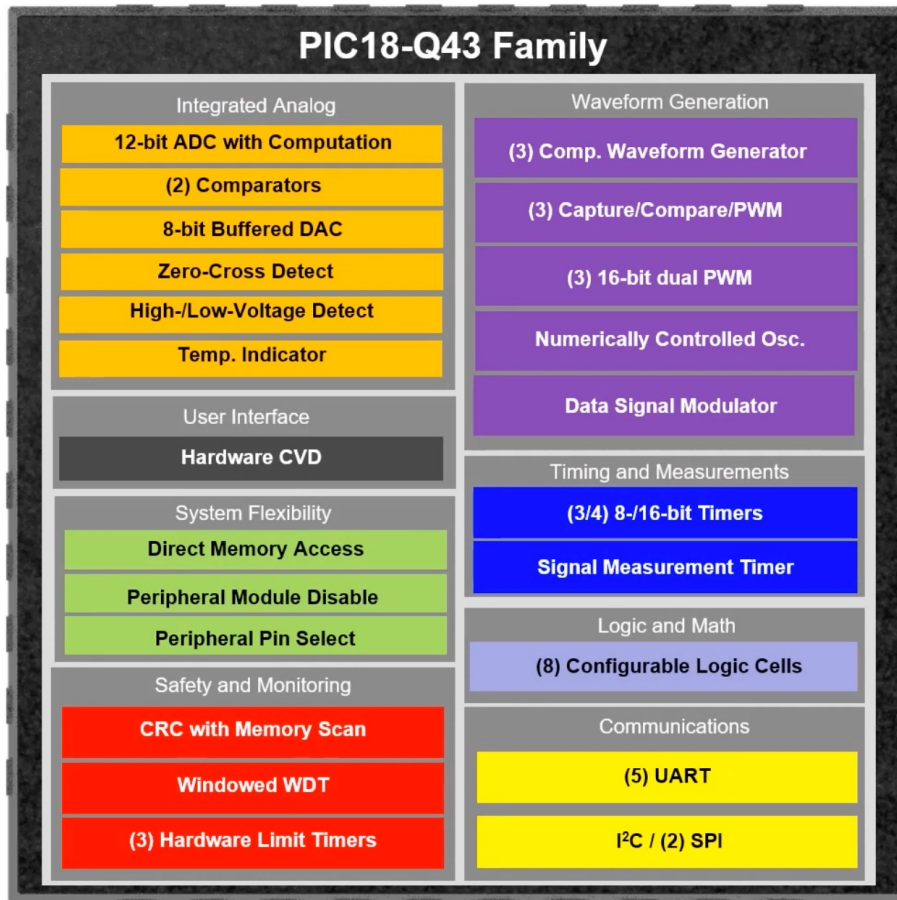
Idle, Standby, Power Down

1.8 to 5.5 V Operation
-40 to 125°C

Voltaje de Operación

Voltaje de operación 1.8 a 5.5 V

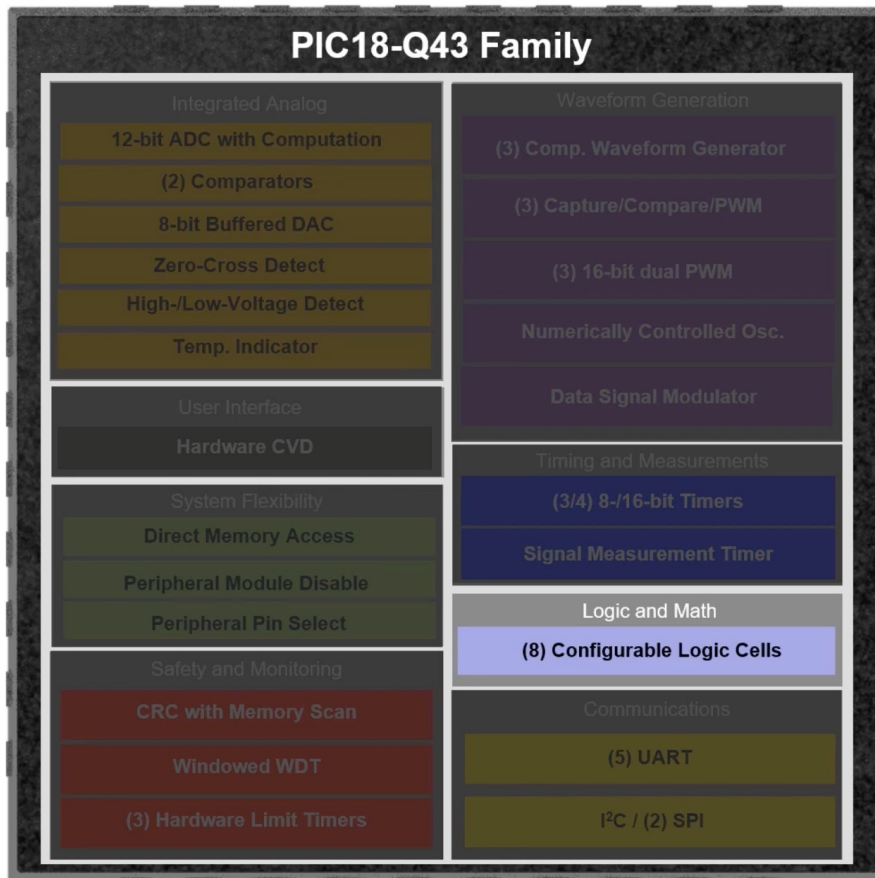
PIC18F57Q43



Periféricos

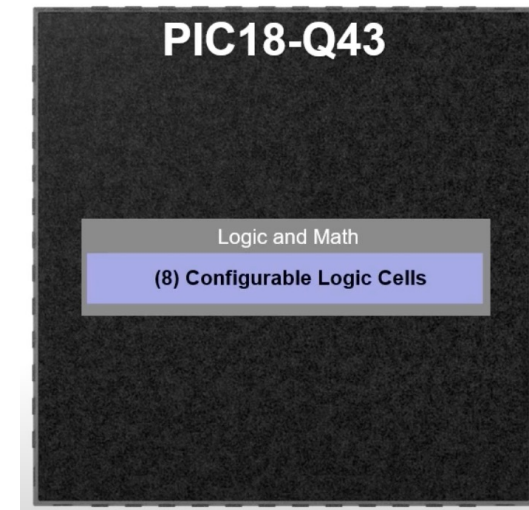
La familia PIC18-Q43 cuenta con una diversidad de periféricos para el desarrollo de aplicaciones.

PIC18F57Q43



Periférico CLC

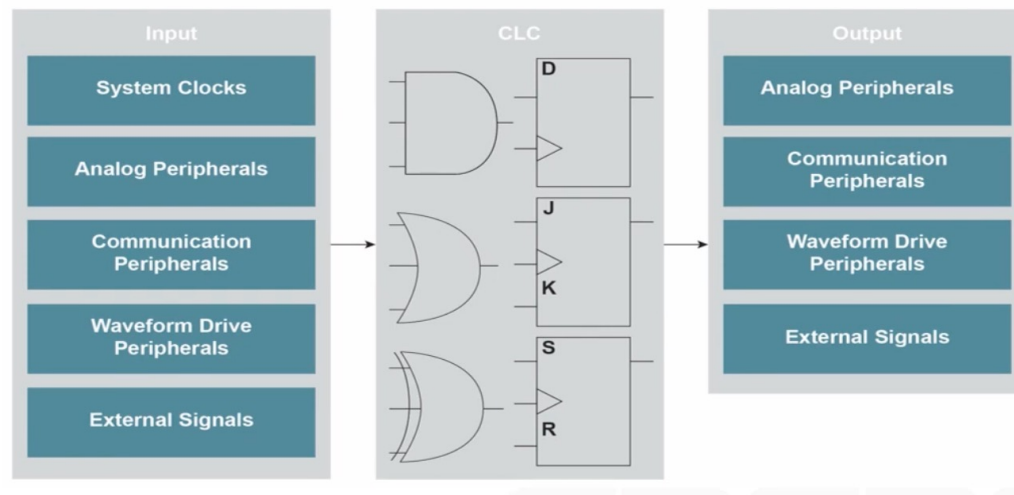
Celdas Lógicas Configurables.



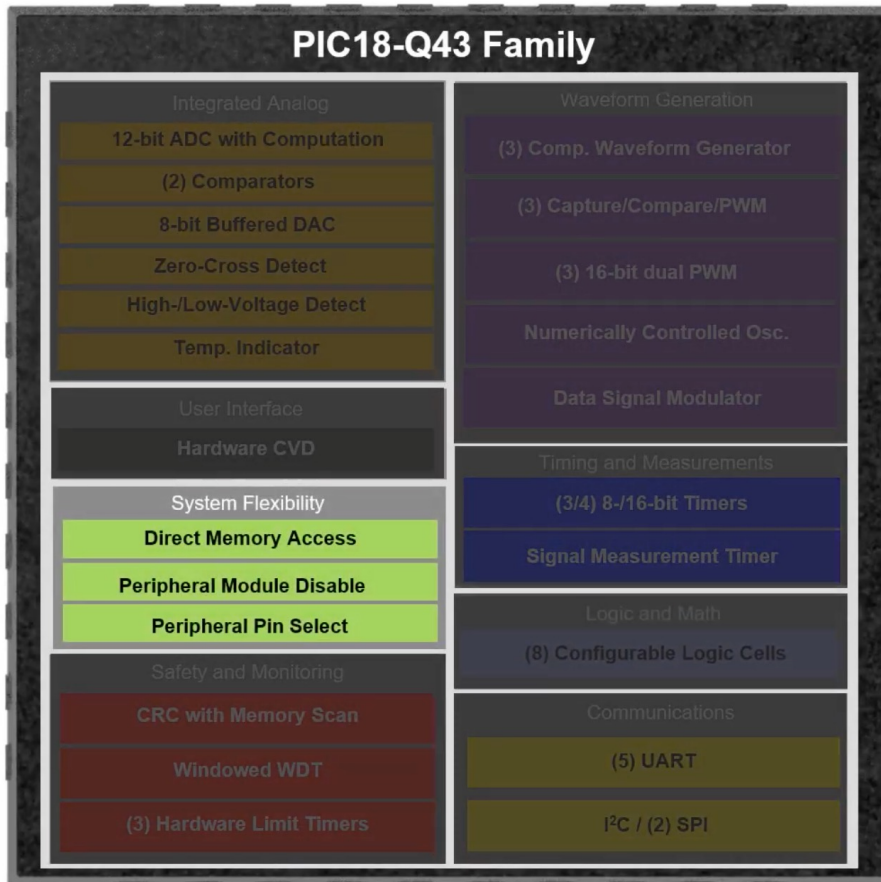
PIC18F57Q43

Periférico CLC

Permite crear una lógica personalizada combinacional o secuencial para señales internas o externas. Se puede utilizar para la interconexión de periféricos para crear una aplicación personalizada. Esto automatiza el manejo de tareas dentro del sistema y **reduce el procesamiento de la CPU** a través de la lógica basada en hardware. La reducción de procesamiento en la CPU impacta en el consumo de energía del microcontrolador.



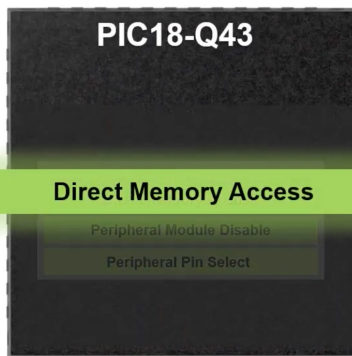
PIC18F57Q43



Flexibilidad

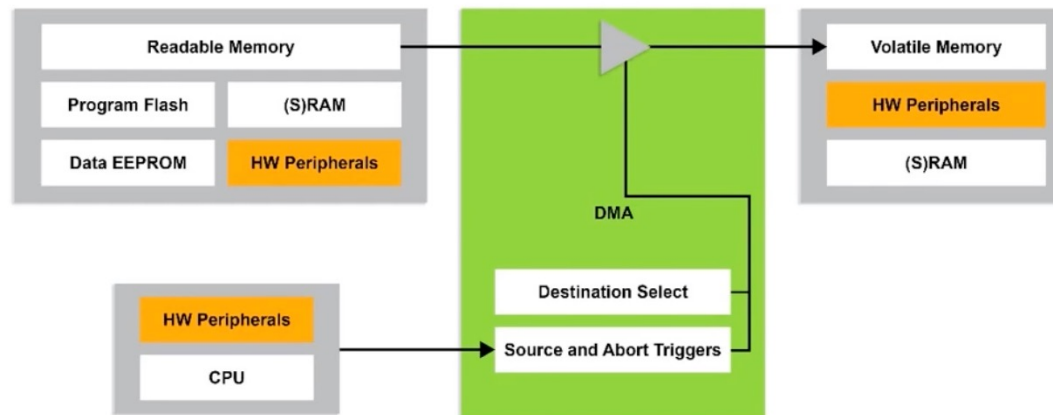
Periféricos como el DMA, PMD y PPS ayudan en el rendimiento y flexibilidad en el desarrollo de aplicaciones.

PIC18F57Q43



Direct Memory Access (DMA)

Elimina la necesidad de que la CPU participe en las transferencias de datos. El DMA puede acceder a todos los espacios de memoria para leer datos y escribir en la RAM y periféricos. Esto reduce la latencia en la transferencia de datos para aplicaciones de control en tiempo real.

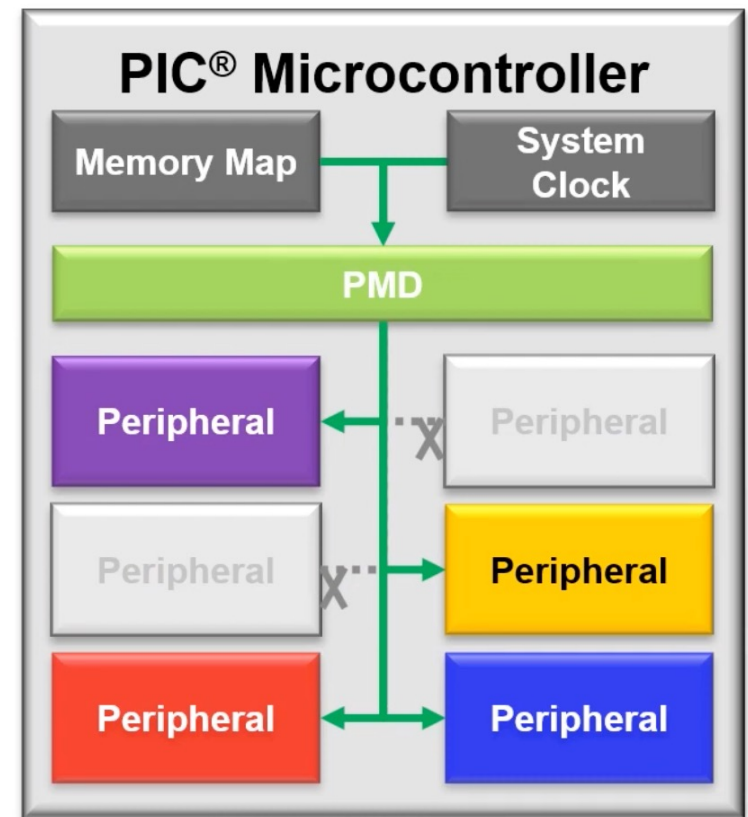


PIC18F57Q43

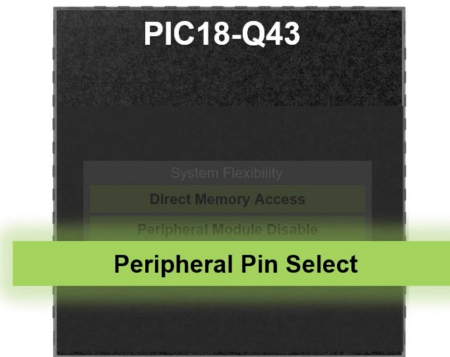


Peripheral Module Disable (PMD)

Permite desactivar los periféricos que no están en uso, logrando un consumo de energía cero para mejorar la vida útil de la batería.

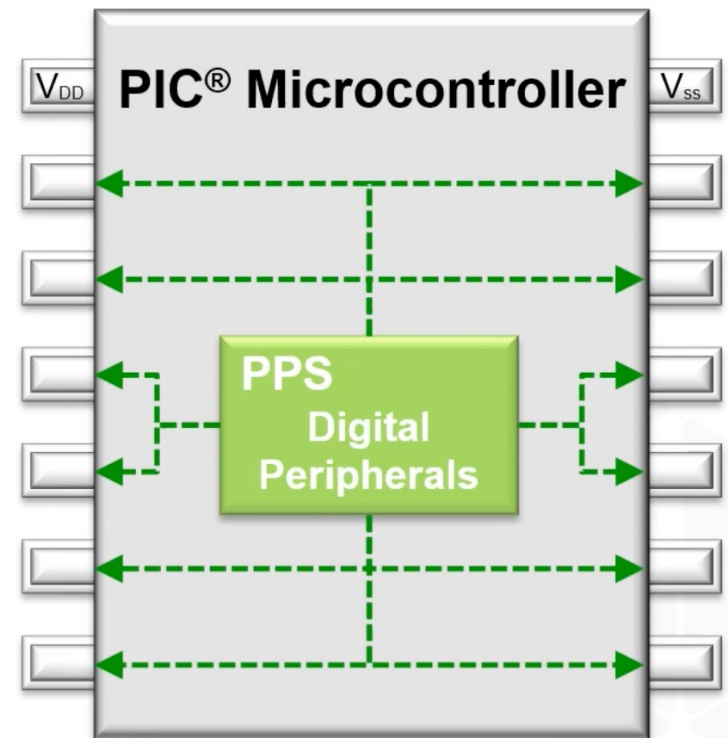


PIC18F57Q43

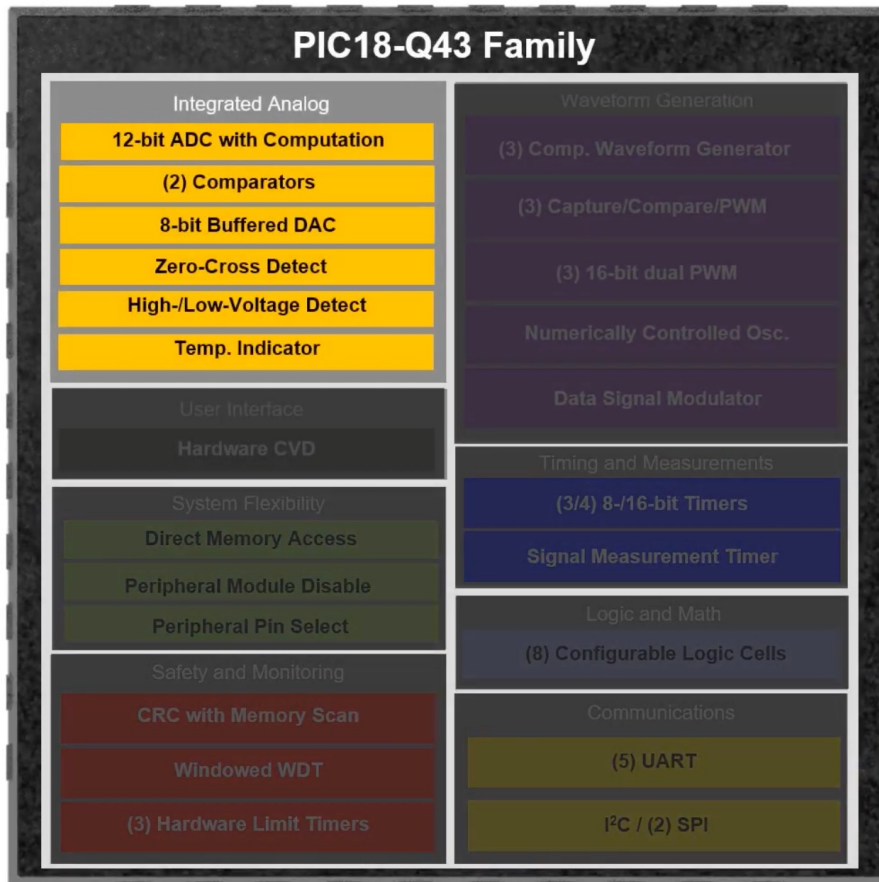


Peripheral Pin Select (PPS)

Ofrece una asignación flexible de pines y elimina la superposición de pines para una optimización total de los recursos de los periféricos. De esta manera la complejidad del diseño se puede reducir drásticamente.



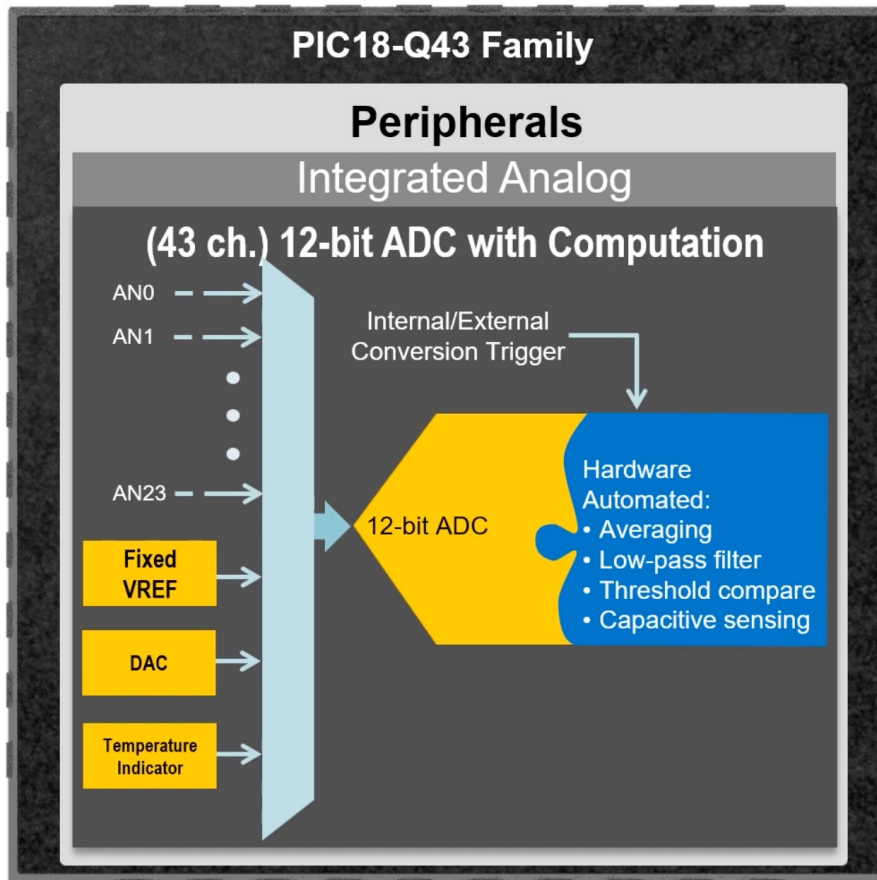
PIC18F57Q43



Flexibilidad

Periféricos como el DMA, PMD y PPS ayudan en el rendimiento y flexibilidad en el desarrollo de aplicaciones.

PIC18F57Q43

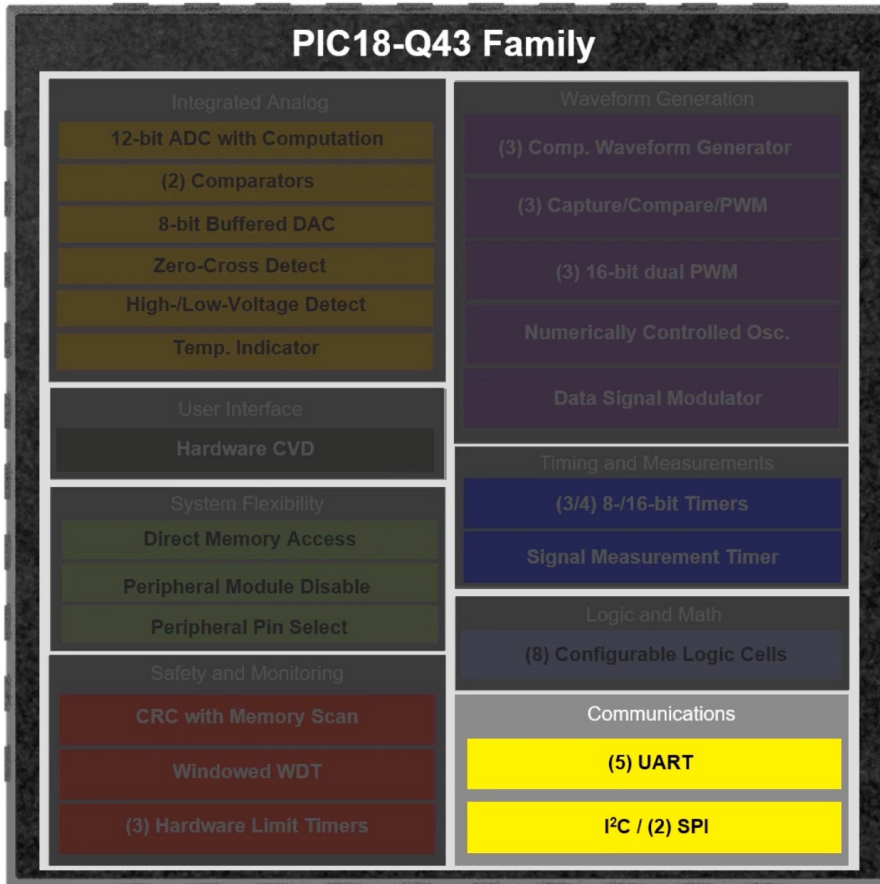


Integración Analógica

Las capacidades analógicas avanzadas reducen aún más la necesidad de algoritmos de software. Agrega hardware que permite realizar automáticamente cálculos avanzados y filtrado.

ADC: 12-bit de resolución con Módulo Computacional.

PIC18F57Q43

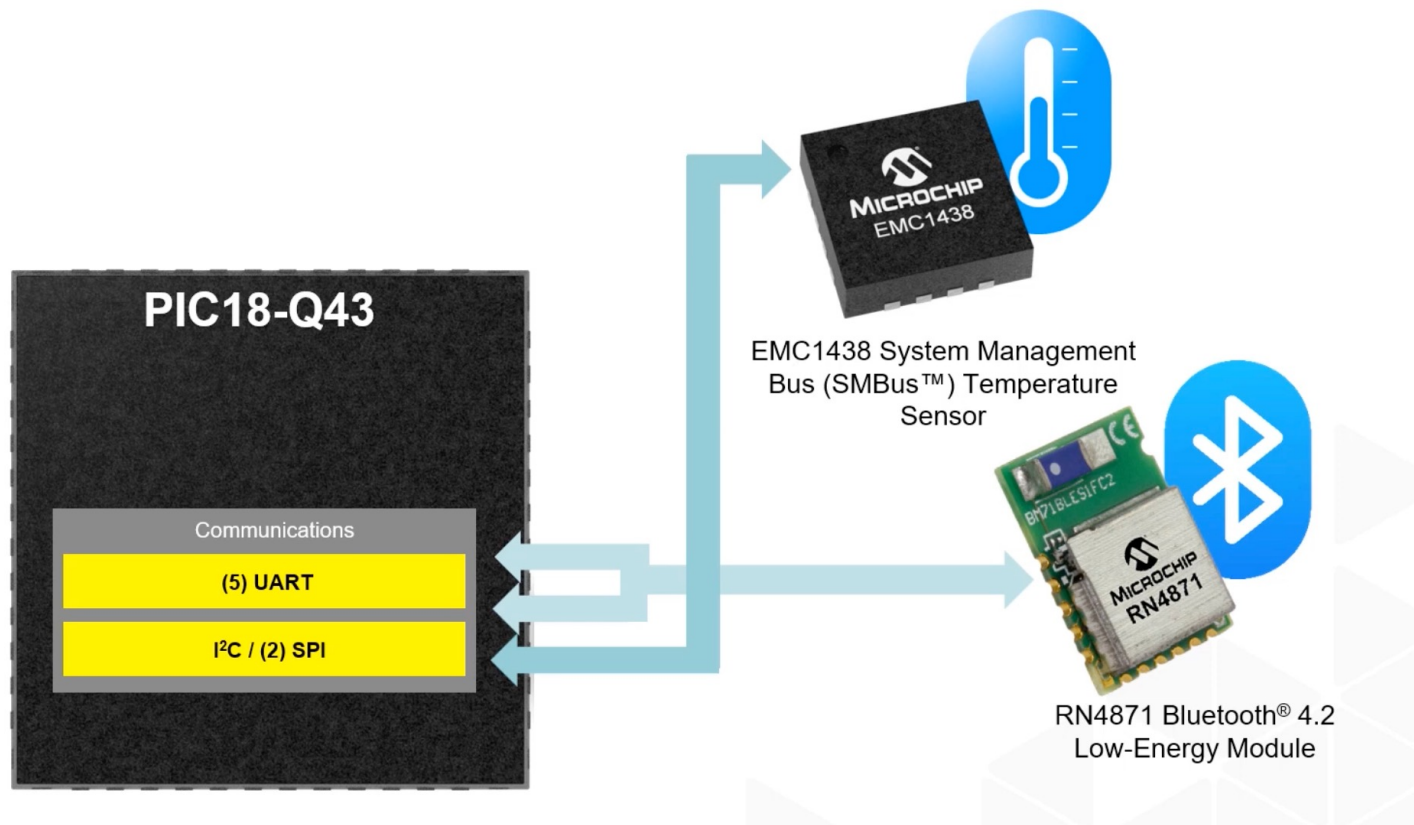


Periféricos de Comunicación

Periféricos de comunicación independientes del núcleo como I2C y UART.

Permite la comunicación con componentes externos como sensores y módulos de comunicación inalámbrica.

PIC18F57Q43

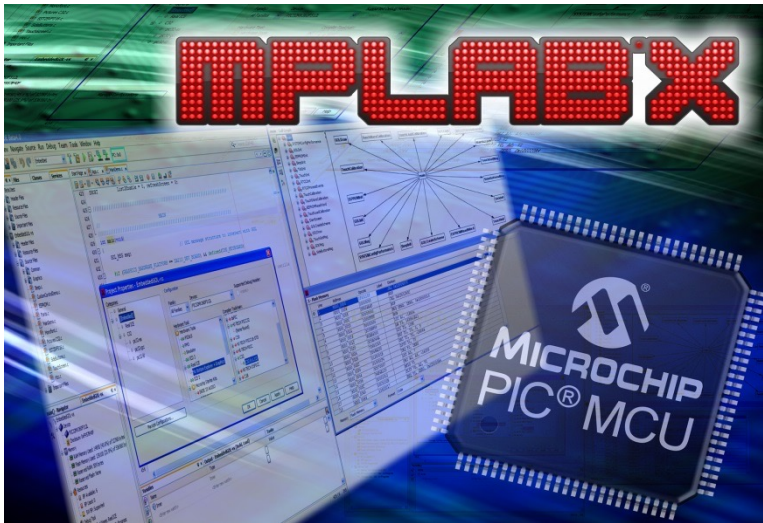


Herramientas de Software



Herramientas de Software

MPLAB X



MPLAB X

Entorno de desarrollo oficial para desarrollar aplicaciones con Microcontroladores PIC y AVR.

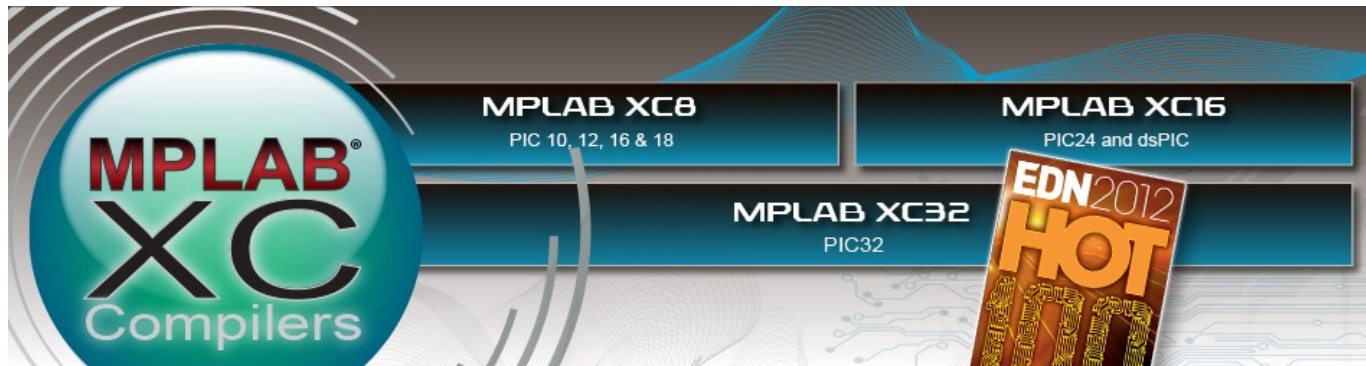
Compatible con Windows, Linux y Mac.

Flexibilidad de integración de diversos compiladores para mcu de 8 bits:

- XC8
- CCS
- C18
- HITECH

Compilador XC8

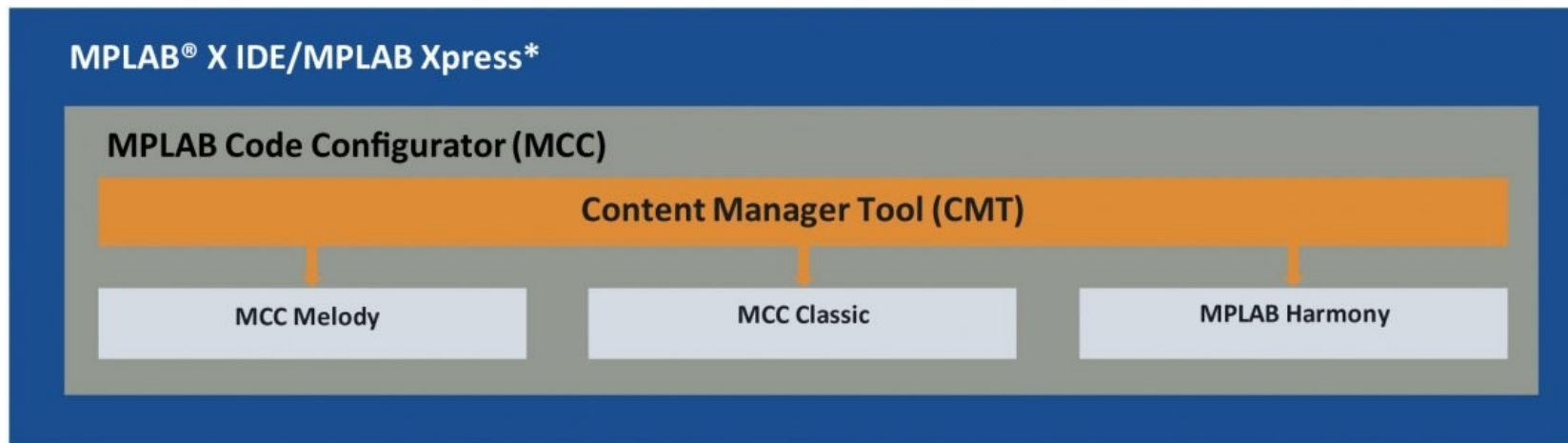
- Compilador ISO C90 ANSI C
- Soporta todos los MCU PIC® de 8 bits: PIC10, PIC12, PIC16 y PIC18.
- Disponible para Windows®, Linux® and Mac OS® X
- Integración con MPLAB X.



MPLAB® Code Configurator (MCC) Melody

MCC Melody

Es la evolución de MCC Classic. Es una herramienta de configuración gráfica que ofrece una arquitectura mejorada y más flexible para configurar de manera más sencilla los dispositivos.



*MPLAB Xpress only supports MCC Melody.

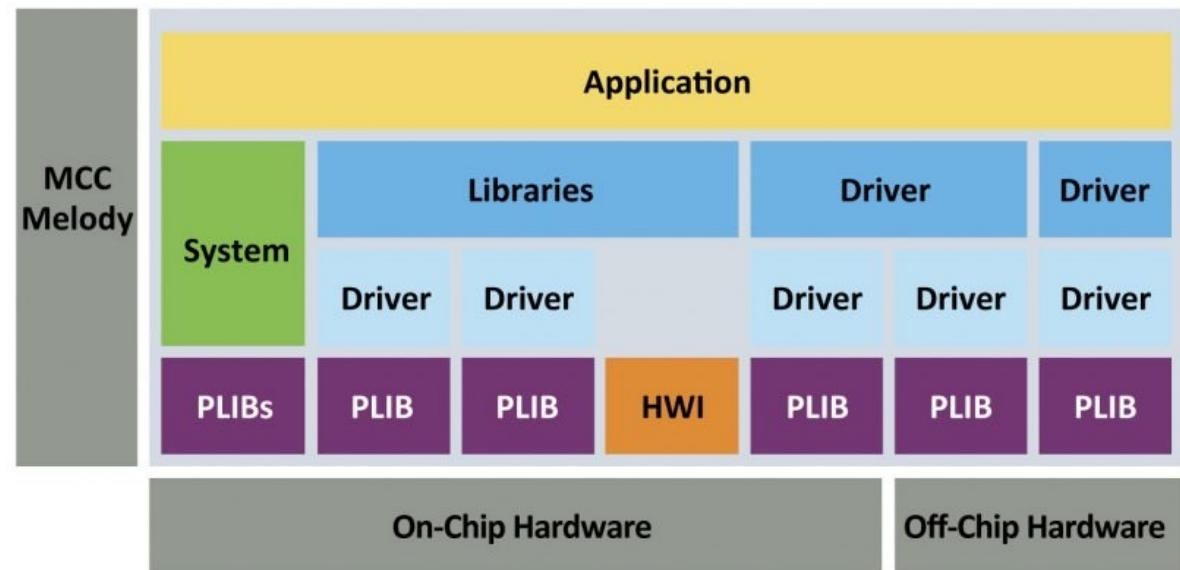
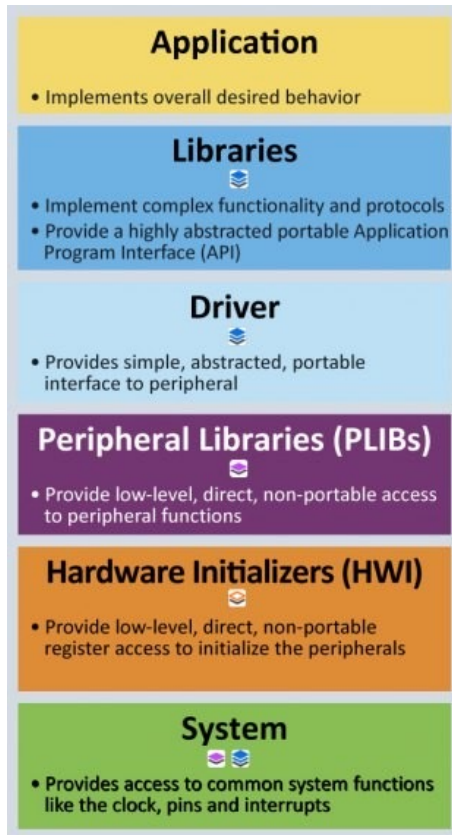
MPLAB® Code Configurator (MCC) Melody

MCC Melody

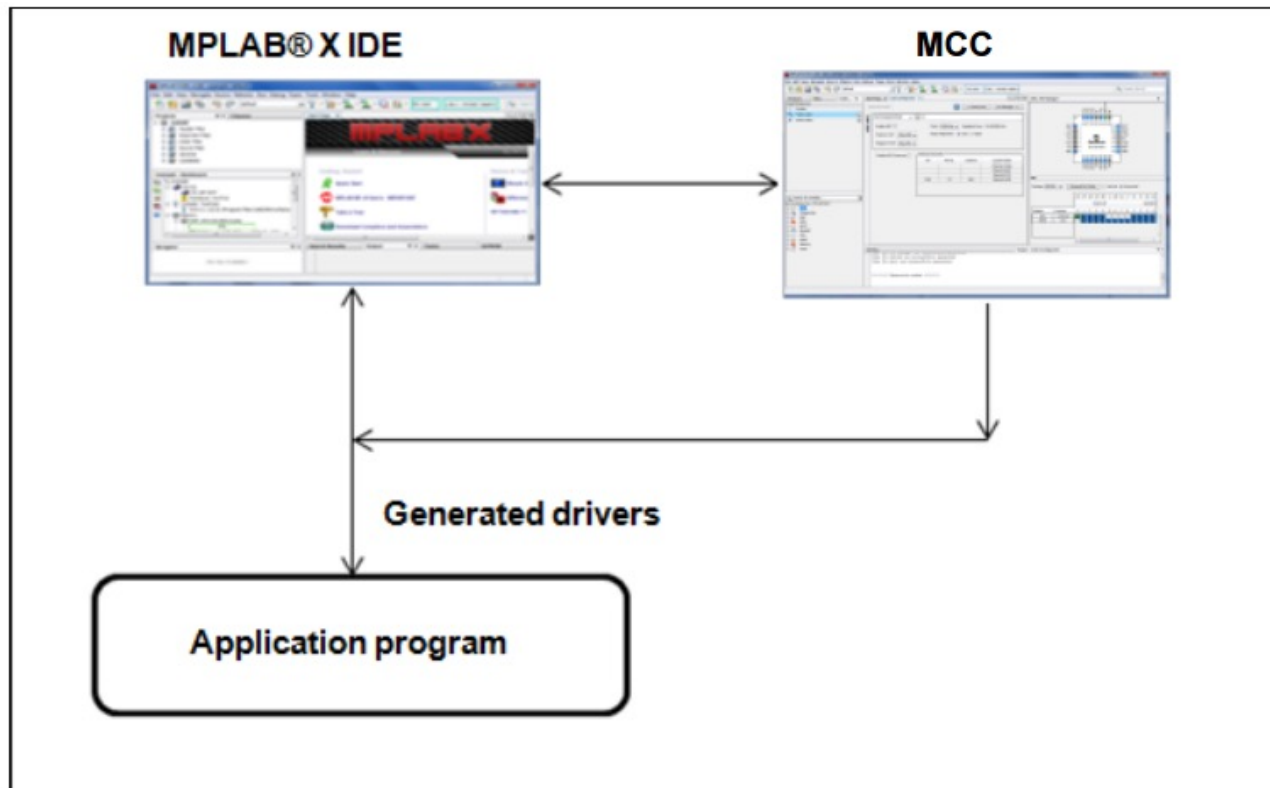
Permite visualizar de manera clara las dependencias de los componentes para simplificar el desarrollo y ofrece un fácil mantenimiento al permitir el control de versiones de contenido a nivel de drivers.

Proporciona librerías, drivers, peripheral libraries (PLIB) e inicializadores de hardware (HWI).

MPLAB® Code Configurator (MCC) Melody



MPLAB® Code Configurator (MCC) Melody



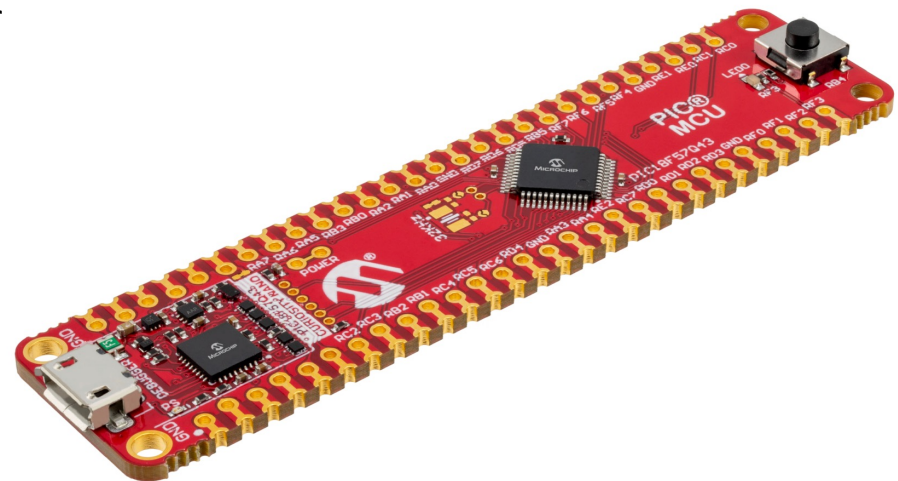
PIC18F57Q43 Curiosity Board



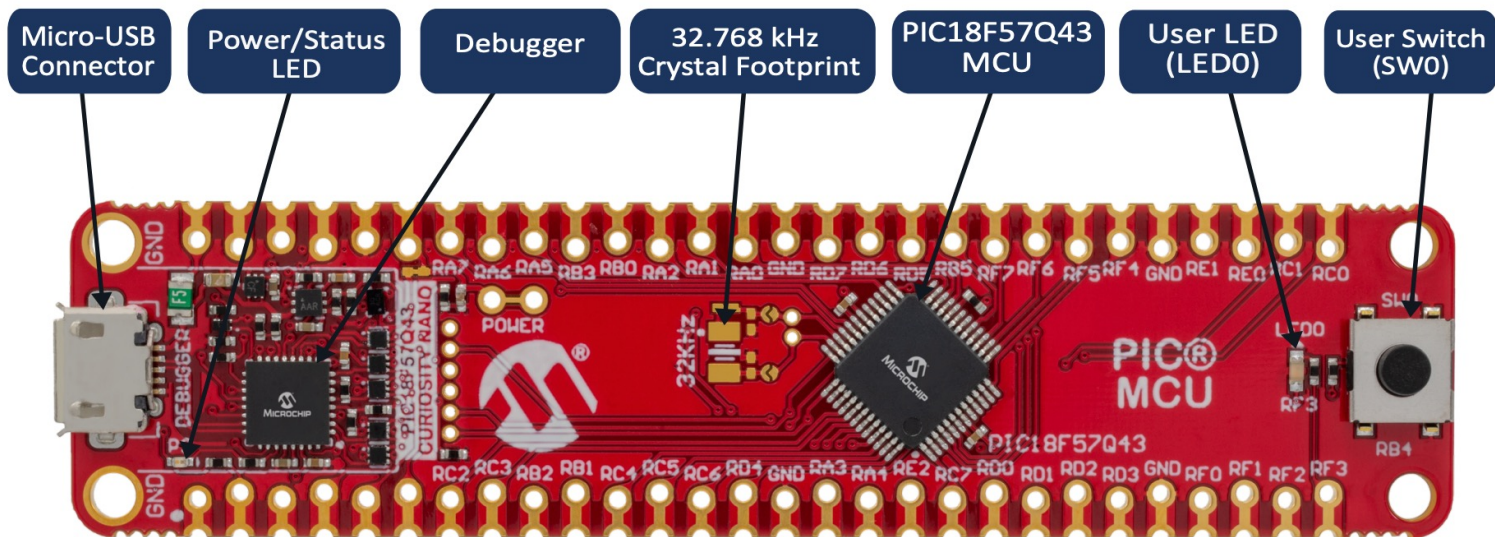
PIC18F57Q43 Curiosity Board

MPLAB® Code Configurator (MCC) Melody

La tarjeta PIC18F57Q43 Curiosity Nano es una plataforma de hardware para desarrollar aplicaciones con el PIC18F57Q43.



MPLAB® Code Configurator (MCC) Melody



MPLAB® Code Configurator (MCC) Melody

On-Board Debugger

La placa contiene un circuito programador y depurador. El circuito es un USB-composite con las siguientes interfaces:

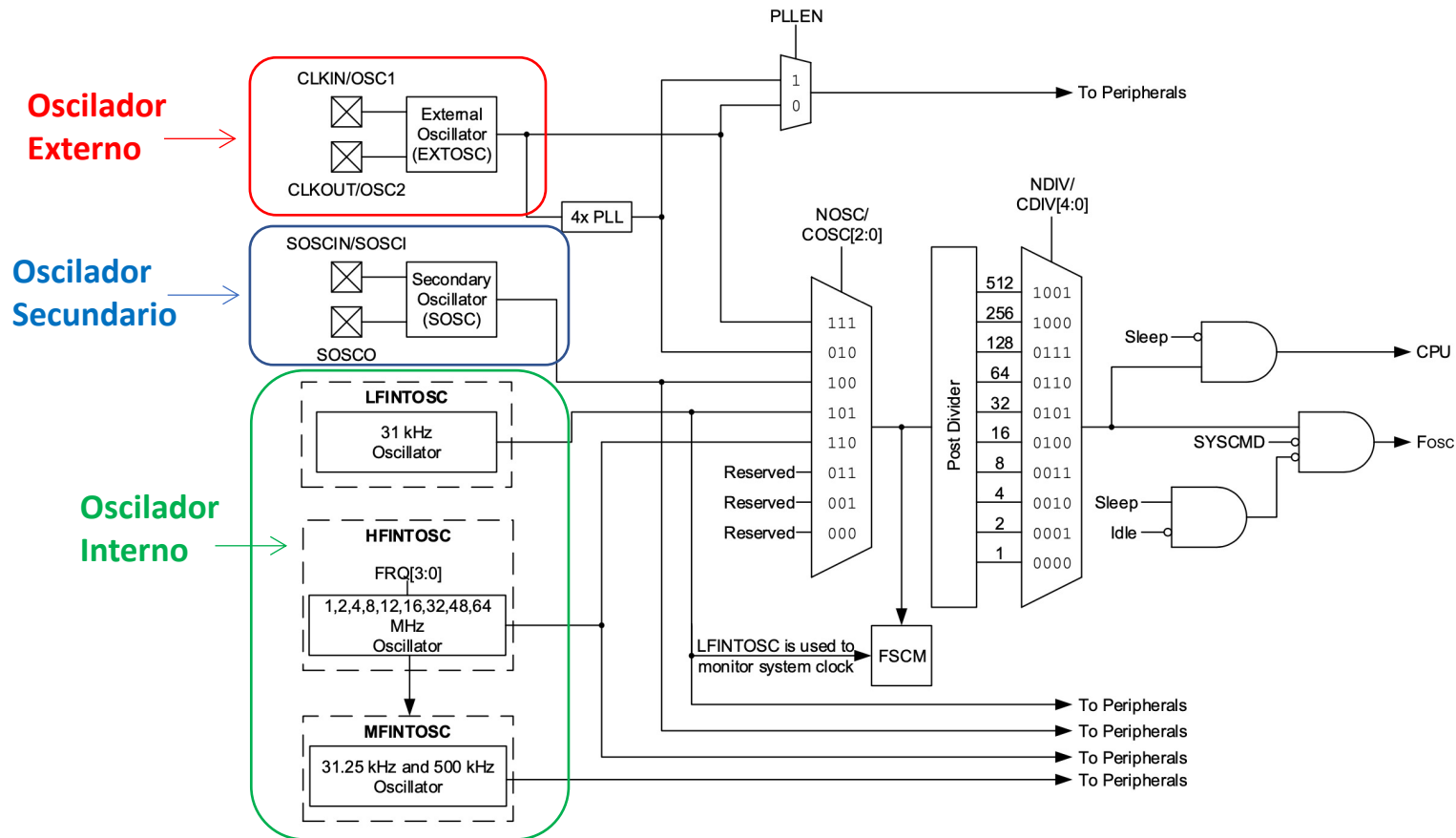
- El debugger el cual puede programar y depurar el PIC18F57Q43.
- La clase MSD que permite la programación del PIC18F57Q43 en modo drag-and-drop.
- Puerto Virtual CDC conectado al periférico UART del PIC18F57Q43, y proporciona una manera fácil de comunicarse con la aplicación de destino a través del software de terminal.

Osilador PIC18-Q43



Oscilador PIC18-Q43

Oscilador



Oscilador

Oscilador Externo: Es posible configurar el oscilador externo predeterminado del sistema al arranque y reinicio del dispositivo utilizando los bits de configuración `RSTOSC` y `FEXTOSC`. Si se requiere configurar el oscilador externo en tiempo de ejecución debe utilizarse los bits `NOSC` y `NDIV`.

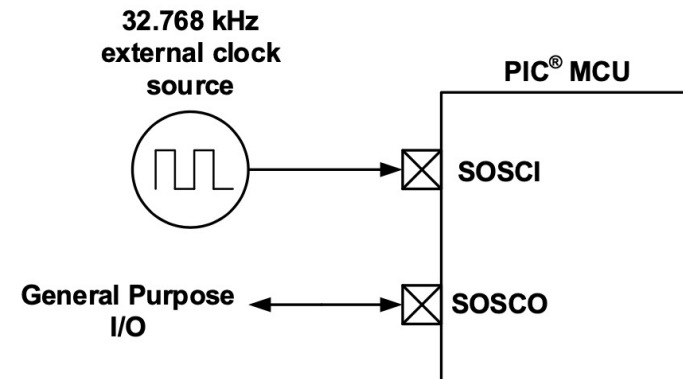
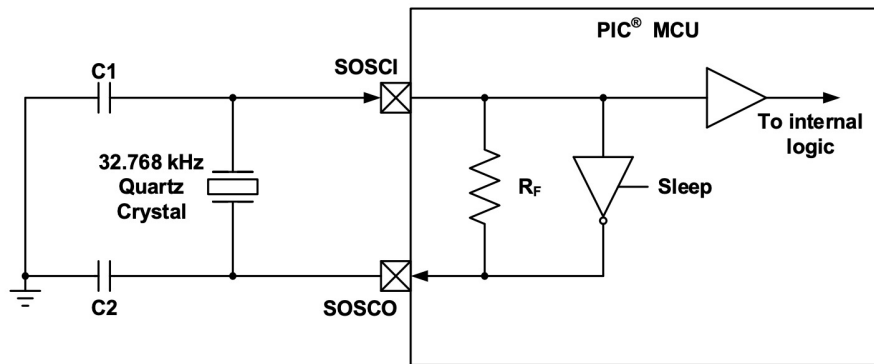
- **ECL:** External Clock Low-Power mode (below 500 kHz)
- **ECM:** External Clock Medium-Power mode (500 kHz to 8 MHz)
- **ECH:** External Clock High-Power mode (above 8 MHz)
- **LP:** 32 kHz Low-Power Crystal mode
- **XT:** Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 500 kHz and 4 MHz)
- **HS:** High Gain Crystal or Ceramic Resonator mode (above 4 MHz, up to 20 MHz)

*LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2

*The ECH, ECM, and ECL clock modes rely on an external logic level signal as the device clock source connected to CLKIN pin

Oscilador

Oscilador Secundario: El oscilador secundario es un bloque oscilador separado que puede usarse como una fuente de reloj del sistema alternativo o como fuente de clock de un Timer. Esta optimizado para 32.768 kHz y puede ser utilizado con un cristal externo conectado a los pines SOSCI y SOSCO o una fuente de clock externa conectada al pin SOSCIN.



Oscilador

Oscilador Interno:

- **HFINTOSC** (High-Frequency Internal Oscillator): Puede trabajar hasta 64 MHz.
- **LFINTOSC** (Low-Frequency Internal Oscillator) : Opera a 31 kHz
- **MFINTOSC** (Medium-Frequency Internal Oscillator): Genera dos frecuencias de clock (500 kHz y 31.25 kHz). Este oscilador es generado a partir del HFINTOSC utilizando logica de división dinámica. El MFINTOSC no se puede utilizar como reloj del sistema, pero se puede utilizar como fuente de reloj para ciertos periféricos, como un temporizador.

Osilador PIC18-Q43



Puertos

Los **puertos** son la comunicación con el exterior del microcontrolador. Dependiendo del modelo y encapsulado del microcontrolador, pueden haber diferentes puertos disponibles.

Algunos pines de los puertos se multiplexan entre diferentes funciones alternativas de los periféricos del dispositivo.

Cada puerto contiene **3 registros** involucrados para su operación:

PORT_x : Registro utilizado para leer o escribir datos.

LAT_x : Registro para escribir datos.

TRIS_x : Registro para configurar un puerto como entrada o salida.

Puertos – Registro TRISx

El registro `TRISx` es utilizado para configuración de **entradas y salidas** digitales.

0	SALIDA
1	ENTRADA

Ejemplo de configuración con el Puerto B, es decir `TRISB`:

TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
0	0	0	0	1	1	1	1

Salidas: TRISB7, TRISB6, TRISB5, TRISB4
Entradas: TRISB3, TRISB2, TRISB1, TRISB0

Puertos – Registro TRISx

TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
0	0	0	0	1	1	1	1
Salidas				Entradas			

Implementación en Código C:

```
TRISB = 0b00001111; //notacion binaria
```

```
TRISB = 0x0F; //notacion hexadecimal
```

Puertos – Registro TRISx

El registro TRIS_x es utilizado para configuración de **entradas** y **salidas** digitales.

0	SALIDA
1	ENTRADA

Ejemplo de configuración con el Puerto D, es decir TRISD:

TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
0	0	0	0	0	0	0	0

Salidas

Puertos – Registro TRISx

TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
0	0	0	0	0	0	0	0

Salidas

Implementación en Código C:

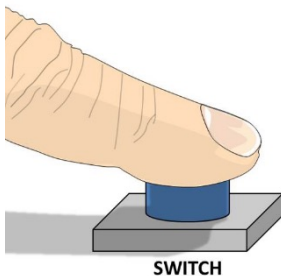
```
TRISD = 0b00000000; //notacion binaria
```

```
TRISD = 0x00; //notacion hexadecimal
```


Puertos – Registro PORTx

El registro PORTx es utilizado para lectura. Ejemplo con PORTB :

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
1	1	1	0	1	1	1	1



Al presionar el botón, el pin RB4 cambia de 1 a 0

```
if(PORTBbits.RB4 == 0) { } //Lectura del estado del pin RA5
```

Puertos – Registro LATx

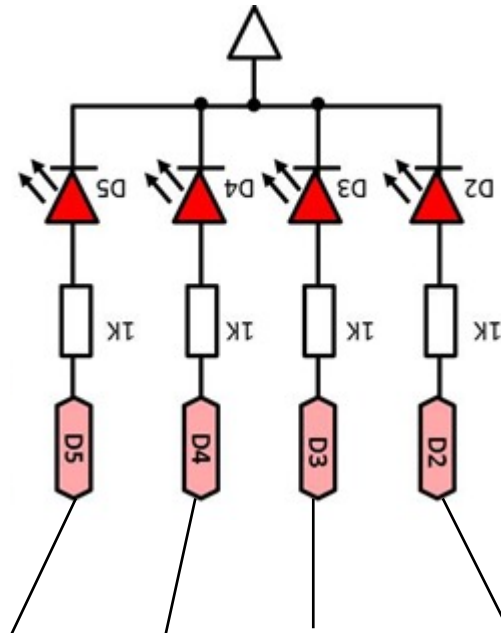
El registro LAT_x es utilizado para escritura.
Ejemplo con LATA.

Notación Binaria:

```
LATA = 0b00001111;
```

Notación Hexadecimal:

```
LATA = 0x0F;
```



LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
0	0	0	0	1	1	1	1

GRACIAS!



Dignal | Sistemas de desarrollo
ventas@dignal.com
www.dignal.com

