

Book: [Microcontroladores PIC – Programacion en C con ejemplos](#)

#### 1.4 Microcontroladores pic

razones por lo que esto es verdadero... El nombre verdadero de este microcontrolador es PICmicro (Peripheral Interface Controller), conocido bajo el nombre PIC. Su primer antecesor fue creado en 1975 por la compañía General Instruments. Este chip denominado PIC1650 fue diseñado para propósitos completamente diferentes. Diez años más tarde, al añadir una memoria EEPROM, este circuito se convirtió en un verdadero microcontrolador PIC. Hace unos pocos años la compañía Microchip Technology fabricó la 5 billonésima muestra. Si está interesado en aprender más sobre eso, siga leyendo. If you are interested in learning more about it, just keep on reading. La idea principal de este libro es proporcionar la información necesaria al usuario para que sea capaz de utilizar los microcontroladores en la práctica después de leerlo. Para evitar explicaciones pesadas y las historias infinitas sobre las características útiles de los microcontroladores diferentes, este libro describe el funcionamiento de un modelo particular que pertenece a la "clase media alta". Es PIC16F887 - bastante poderoso para ser digno de atención y bastante simple para poder ser utilizado por cualquiera. Así, los siguientes capítulos describen este microcontrolador en detalle y también se refieren a la familia PIC entera.

Familia	RAM [Kbytes]	RAM [bytes]	Pines	Reloj de reloj. [MHz]	Convertidor A/D	Convertidor D/A	Comparadores	Convertidores de 8/16 bits	Comunicación serial	Perif. de E/S	PWM
<b>Arquitectura de la gama baja de 8 bits, palabra de instrucción de 12 bits</b>											
PIC10FXXX	0.375 - 0.75	16 - 24	6 - 8	4 - 8	0 - 2	8	0 - 1	1 x 8	-	-	-
PIC12FXXX	0.75 - 1.5	25 - 38	8	4 - 8	0 - 3	8	0 - 1	1 x 8	-	-	-
PIC16FXXX	0.75 - 3	25 - 134	14 - 44	20	0 - 3	8	0 - 2	1 x 8	-	-	-
PIC16HVXXX	1.5	25	18 - 20	20	-	-	-	1 x 8	-	-	-
<b>Arquitectura de la gama media de 8 bits, palabra de instrucción de 14 bits</b>											
PIC12FXXX	1.75 - 3.5	64 - 128	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	-	0 - 1
PIC12HVXXX	1.75	64	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	-	0 - 1
PIC16FXXX	1.75 - 14	64 - 368	14 - 64	20	0 - 13	8 or 10	0 - 2	1 - 2 x 8 1 x 16	USART I2C SPI	-	0 - 3
PIC16HVXXX	1.75 - 3.5	64 - 128	14 - 20	20	0 - 12	10	2	2 x 8 1 x 16	USART I2C SPI	-	-
<b>Arquitectura de la gama alta de 8 bits, palabra de instrucción de 16 bits</b>											
PIC18FXXX	4 - 128	256 - 3936	18 - 80	32 - 48	4 - 16	10 or 12	0 - 3	0 - 2 x 8 2 - 3 x 16	USB2.0 CAN2.0 USART I2C SPI	-	0 - 5
PIC18FXXJXX	8 - 128	1024 - 3936	28 - 100	40 - 48	10 - 16	10	2	0 - 2 x 8 2 - 3 x 16	USB2.0 USART Ethernet I2C SPI	-	2 - 5
PIC18FXXKXX	8 - 64	768 - 3936	28 - 44	64	10 - 13	10	2	1 x 8 3 x 16	USART I2C SPI	-	2

CPU por más de 8 líneas. Hay microcontroladores de 12, 14 y 16 bits, dependiendo de la anchura del bus. La tabla anterior muestra las características principales de estas tres categorías. Como se puede ver en la tabla de la página anterior, salvo "los monstruos de 16 bits" PIC 24FXXX y PIC 24HXXX - todos los microcontroladores tienen la arquitectura Harvard de 8 bits y pertenecen a una de las tres grandes grupos. Por eso, dependiendo del tamaño de palabra de programa existen la primera, la segunda y la tercera categoría de microcontroladores, es decir microcontroladores de 12, 14 o 16 bits. Puesto que disponen del núcleo similar de 8 bits, todos utilizan el mismo juego de instrucciones y el "esqueleto" básico de hardware conectado a más o menos unidades periféricas. Los microcontroladores PIC con palabras de programa de 14 bits parecen ser la mejor opción para los principiantes. Aquí está el porqué...

El juego de instrucciones para los microcont

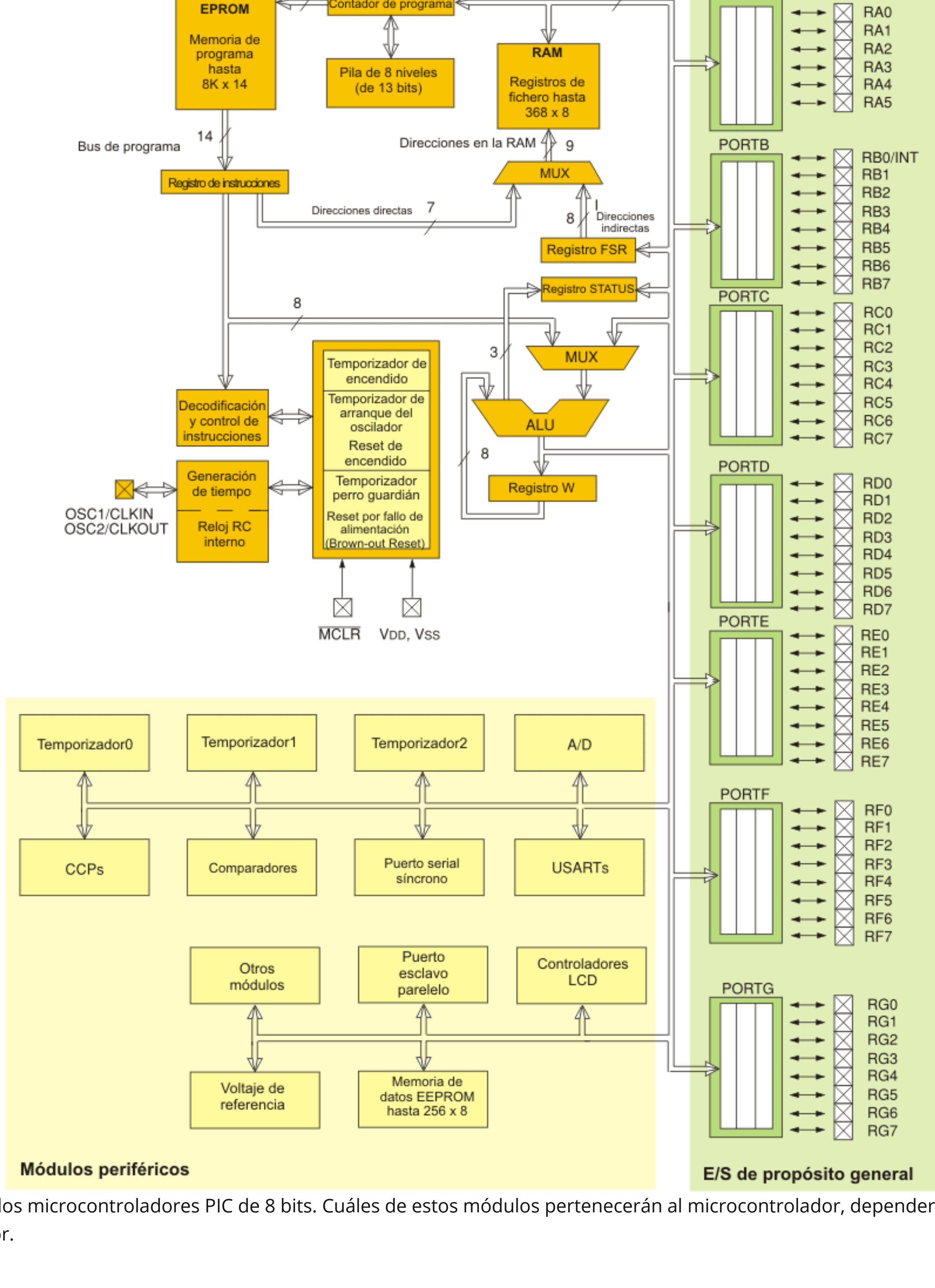
instrucciones yace en la arquitectura RISC. Esto quiere decir que las instrucciones son bien optimizadas desde el aspecto de la velocidad operativa, la sencillez de la arquitectura y la compacidad del código. Lo malo de la arquitectura RISC es que se espera del programador que haga frente a estas instrucciones. Por supuesto, esto es relevante sólo si se utiliza el lenguaje ensamblador para la programación. Este libro se refiere a la programación en el lenguaje de alto nivel C, lo que significa que la mayor parte del trabajo ya fue hecho por alguien más. Así, sólo se tienen que utilizar instrucciones relativamente simples.

Todas las instrucciones se ejecutan en un ciclo. La ú

instrucciones que cambian el contenido del contador de programa. En ambos casos, dos ciclos de reloj son necesarios para la ejecución de la instrucción, mientras que el segundo ciclo se ejecuta como un NOP (No operation). Las instrucciones de un ciclo consisten en cuatro ciclos de reloj. Si se utiliza un oscilador de 4 MHz, el tiempo nominal para la ejecución de la instrucción es 1  $\mu$ s. En cuanto a las instrucciones de ramificación, el tiempo de ejecución de la instrucción es 2  $\mu$ s. Juego de instrucciones de los microcontroladores PIC de 14 bits:

Instrucciones para la transmisión de datos					
MOVLW k	Mover literal a W	k -> w		1	
MOVWF f	Mover el contenido de W a f	W -> f		1	
MOVF f,d	Mover el contenido de f a d	f -> d	Z	1	1, 2
CLRW	Borrar el contenido de W	0 -> W	Z	1	
CLRF f	Borrar el contenido de f	0 -> f	Z	1	2
SWAPF f,d	Intercambiar de nibbles en f	f(7:4),(3:0) -> f(3:0),(7:4)		1	1, 2
Instrucciones aritmético - lógicas					
ADDLW k	Sumar literal a W	W+k -> W	C, DC, Z	1	
ADDWF f,d	Sumar el contenido de W y f	W+f -> d	C, DC, Z	1	1, 2
SUBLW k	Restar W de literal	k-W -> W	C, DC, Z	1	
SUBWF f,d	Restar W de f	f-W -> d	C, DC, Z	1	1, 2
ANDLW k	AND W con literal	W AND k -> W	Z	1	
ANDWF f,d	AND W con f	W AND f -> d	Z	1	1, 2
IORLW k	OR inclusivo de W con literal	W OR k -> W	Z	1	
IORWF f,d	OR inclusivo de W con f	W OR f -> d	Z	1	1, 2
XORWF f,d	OR exclusivo de W con literal	W XOR k -> W	Z	1	1, 2
XORLW k	OR exclusivo de W con f	W XOR f -> d	Z	1	
INCF f,d	Sumar 1 a f	f+1 -> f	Z	1	1, 2
DECF f,d	Restar 1 a f	f-1 -> f	Z	1	1, 2
RLF f,d	Rotar F a la izquierda a través del bit de Acarreo		C	1	1, 2
RRF f,d	Rotar F a la derecha a través del bit de Acarreo		C	1	1, 2
COMF f,d	Complementar f	f -> d	Z	1	1, 2
Instrucciones orientadas a bit					
BCF f,b	Poner a 0 el bit b del registro f	0 -> f(b)		1	1, 2
BSF f,b	Poner a 1 el bit b del registro f	1 -> f(b)		1	1, 2
Instrucciones de control de programa					
BTFSZ f,b	Saltar si bit b de registro f es 0	Skip if f(b) = 0		1 (2)	3
BTFS f,b	Saltar si bit b de reg. f es 1	Skip if f(b) = 1		1 (2)	3
DECFSZ f,d	Disminuir f en 1. Saltar si el resultado es 0	f-1 -> d skip if Z = 1		1 (2)	1, 2, 3
INCFSZ f,d	Incrementar f en 1. Saltar si el resultado es 1	f+1 -> d skip if Z = 0		1 (2)	1, 2, 3
GOTO k	Saltar a una dirección	k -> PC		2	
CALL k	Llamar a una subrutina	PC -> TOS, k -> PC		2	
RETURN	Retornar de una subrutina	TOS -> PC		2	
RETLW k	Retornar con literal en W	k -> W, TOS -> PC		2	
RETFIE	Retornar de una interrupción	TOS -> PC, 1 -> GIE		2	
Otras instrucciones					
NOP	No operación	TOS -> PC, 1 -> GIE		1	
CLRWDT	Reiniciar el temporizador perro guardián	0 -> WDT, 1 -> TO, 1 -> PD	TO, PD	1	
SLEEP	Poner en estado de reposo	0 -> WDT, 1 -> TO, 0 -> PD	TO, PD	1	

ejecuta en el registro TMR y si  $d=1$ , el pre-escalador será borrado. \*3 Si la instrucción se ejecuta en el registro TMR y si  $d=1$ , el pre-escalador será borrado.



SUBSCRIBE TO