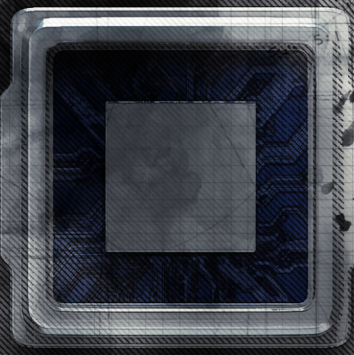


ROBINSON JIMÉNEZ - JAVIER MARTÍNEZ - CÉSAR ROMERO

INTRODUCCIÓN A LA PROGRAMACIÓN DE MICROCONTROLADORES PIC

INTRODUCCIÓN A LA PROGRAMACIÓN DE MICROCONTROLADORES PIC



$$E_{\text{add}} = \frac{h\nu}{e}$$
$$E_{\text{pot}} = \frac{q}{e}$$

*En el arte de la ingeniería, la precisión
es un requisito esencial para el éxito.
No se trata de ser perfecto, sino de ser
útil y eficiente.*

ROBINSON JIMÉNEZ MORENO / JAVIER EDUARDO MARTÍNEZ BAQUERO
CÉSAR AUGUSTO ROMERO MOLANO



Introducción a la programación de microcontroladores PIC

**Editorial
Unillanos** 

Introducción a la programación de microcontroladores PIC

Robinson Jiménez Moreno, D. Eng. Ing.

Docente, Programa Ingeniería en Mecatrónica UMNG

Javier Eduardo Martínez Baquero, MSc. Ing.

Docente, Escuela de Ingeniería Universidad de los Llanos

César Augusto Romero Molano, Esp. Ing.

Docente Escuela de Ingeniería Universidad de los Llanos

Un cura de mura roman, que mura
preservacion mulleres libros non ut
aliqua ex om comitate consequunt.
Dura ultra omes habere in repudiatione
non velintem veli esse ullum dulla
om quoyant.

Introducción a la programación de microcontroladores PIC / Robinson Jiménez Moreno; Javier Eduardo Martínez Baquero; César Augusto Romero Molano - Primera edición. – Villavicencio: Editorial Universidad de los Llanos, 2022.

212 páginas: gráficas, tablas, diagramas; 27 cm.
ISBN: 978-958-8927-51-0
ISBN DIGITAL: 978-958-8927-67-1

Incluye bibliografía.

1. Programación (Computadores digitales analógicos) 2. Circuitos electrónicos 3. Circuitos integrados 4. Controladores (Programas de computador) 5. Microelectrónica

22a
CDD 629.895416 Ed.21

Catalogación en la fuente – Universidad de los Llanos. Sistema de Bibliotecas

Primera edición 2022

Introducción a la programación de microcontroladores PIC

18a
© Robinson Jiménez  <https://orcid.org/0000-0002-4812-3734>

© Javier Eduardo Martínez Baquero  <https://orcid.org/0000-0003-4377-7867>

© César Augusto Romero Molano  <https://orcid.org/0000-0001-9797-0115>

ISBN digital: 978-958-8927-67-1

ISBN impreso: 978-958-8927-51-0

© Universidad de los Llanos

Coordinación editorial: Ana María Lombana Gracia

Diseño de cubierta y diagramación: Mario A. Calderón Collazos

3
Corrección de estilo: Andrés Mantilla Meluk

Editorial Unillanos

Calle 37 # 41-02 Barzal - Sede San Antonio
editorialunillanos@unillanos.edu.co
www.editorial.unillanos.edu.co
Villavicencio, Meta

Impresión

Editorial Unillanos
Villavicencio - Meta

Descargo de responsabilidad: la información contenida en este libro es producto del autor y por consiguiente no compromete la posición de la Universidad de los Llanos. Prohibida la reproducción total o parcial, en cualquier medio, formato o propósito, sin la autorización escrita de la Editorial Unillanos.

Un cura de mura roman, que mura
preservacion mulleres libros non ut
aliqua ex om comitate consequunt.
Dura ultra omes habere in repudiatione
non velintem veli esse ullum dulla
om quoyant.

AGRADECIMIENTOS

Los autores expresan su agradecimiento a las universidades que permitieron el desarrollo de esta obra como sigue:

El autor Robinson Jiménez Moreno agradece a la Universidad Militar Nueva Granada, donde es docente de planta tiempo completo en el Programa de Ingeniería en Mecatrónica, por el tiempo y apoyo brindado para la realización de este documento orientado a fortalecer el material de consulta en las áreas de electrónica digital.

Los autores Javier Eduardo Martínez Baquero y César Augusto Romero Molano agradecen a la Universidad de los Llanos, donde son docentes de planta tiempo completo en la Facultad de Ciencias Básicas e Ingeniería, por el tiempo y apoyo brindado para la realización de este documento orientado a fortalecer el material de consulta en las áreas de electrónica digital.

A su vez, los autores extienden sus agradecimientos al fondo editorial de la Universidad de los Llanos por la gestión asociada a la publicación de este libro.

TABLA DE CONTENIDO

AGRADECIMIENTOS	7
LISTA DE FIGURAS	11
LISTA DE TABLAS	14
INTRODUCCIÓN	15
CAPÍTULO I – LENGUAJE C	23
1.1 Conceptos básicos de programación en C para microcontrolador	24
1.2 Funciones	24
1.3 Tipos de datos	27
1.4 Tipo de datos y estructura de una función	31
1.5 Operadores lógicos aritméticos	32
1.6 Sentencias condicionales	37
1.6.1 Condicional if–else	38
1.6.2 Condicional switch	40
1.6.3 Condicional break	40
1.6.4 Condicional while	41
1.6.5 Condicional for	43
1.7 Arreglos	43
1.8 Librerías	44
1.9 Creación de librerías	45
1.10 Definiciones	47
1.11 Ejercicios del capítulo	48
CAPÍTULO II – PUERTOS DE DATOS Y APLICACIONES	49
2.1 Aplicaciones básicas de manejo de puertos	50
2.2 Entrada y salida de información	50
2.3 Contadores	58
2.4 Retardos	64
2.5 Manejo de teclados	74
2.5.1 Teclado matricial	74
2.6 Manejo de Display	79
2.7 Visualización dinámica	81
2.8 Matriz de led	85
2.9 Manejo de LCD	88
2.9.1 Uso de librería para el módulo LCD	98

2.10 Control de motor DC	103
2.11 Control de servomotores	104
2.12 Control de motores paso a paso	107
2.13 Aplicaciones de entrada y salida de datos	111
2.14 Ejercicios del capítulo	125
CAPÍTULO III – CONFIGURACIÓN DE MÓDULOS BASE	127
3.1 Memoria EEPROM	128
3.1.1 Uso de librería para memoria EEPROM	128
3.2 Conversor A/D	129
3.2.1 Uso de librería para el módulo A/D	136
3.3 Transmisión recepción serial USART	139
3.3.1 Uso de librería para el módulo USART	148
3.4 Ejercicios del capítulo	150
CAPÍTULO IV – MÓDULOS ESPECÍFICOS	151
4.1 Módulo de captura/comparación y PWM	152
4.2 Modo de captura	154
4.2.1 Librería para modo de captura	155
4.3 Modo de comparación	157
4.4 Modo de PWM	159
4.4.1 Librería para PWM	161
4.5 Módulo comparador análogo	164
4.6 Módulo I2C	168
4.6.1 Memoria I2C	170
4.6.2 Reloj en tiempo real (RTC)	176
4.7 Aplicaciones de periféricos especiales	179
4.8 Ejercicios del capítulo	191
ANEXO A – ENTORNO DE DESARROLLO EN MPLAB®	193
ANEXO B – INTERFAZ SERIAL EN C#	201
ANEXO C – DIAGRAMA DE PINES DE LOS MICROS DE REFERENCIA	213
REFERENCIAS	215

LISTA DE FIGURAS

FIGURA 1.	Gamas de microcontroladores de 8 bits de Microchip	17
FIGURA 2.	Herramienta de selección de productos de Microchip	20
FIGURA 3.	Filtros de selección para microcontroladores	20
FIGURA 4.	Almacenamiento con una variable tipo entero que requiere 16 bits	28
FIGURA 5.	Almacenamiento con una variable tipo carácter que requiere 8 bits	28
FIGURA 6.	Simulación para validar el cambio de valor entre las variables	30
FIGURA 7.	Advertencia de error	30
FIGURA 8.	Simulación de llamado iterativo a la función	32
FIGURA 9.	Programa y simulación con operaciones aritméticas	33
FIGURA 10.	Simulación con operaciones lógicas	35
FIGURA 11.	Diagrama de flujo de la condición if-else	38
FIGURA 12.	Diagrama de flujo del ciclo if anidado	39
FIGURA 13.	Diagrama de flujo del condicional while	41
FIGURA 14.	Creación de librerías	46
FIGURA 15.	Uso de librerías	47
FIGURA 16.	Ejemplo de led	55
FIGURA 17.	Primer ejemplo de tres ledes	56
FIGURA 18.	Segundo ejemplo de tres ledes	58
FIGURA 19.	Ejemplo de contador por ledes	60
FIGURA 20.	Ejemplo de contadores anidados	62
FIGURA 21.	Ejemplo de la función delay	66
FIGURA 22.	Simulación delay seis segundos; ejemplo de error	67
FIGURA 23.	Ejemplo de conteo basado en la función delay	72
FIGURA 24.	Voltaje vs. tiempo para estabilización de un pulsador	73
FIGURA 25.	Esquema de teclado matricial	74
FIGURA 26.	Esquema de teclado matricial comercial	75
FIGURA 27.	Diagrama de flujo del barrido matricial para teclado	78
FIGURA 28.	Display de 7 segmentos	79
FIGURA 29.	Ejemplo de visualización dinámica	83
FIGURA 30.	Ejemplo de contador mediante visualización dinámica	84
FIGURA 31.	Matriz de led	85
FIGURA 32.	Activación de matriz de led	87
FIGURA 33.	Visualización dinámica en matriz de led	88
FIGURA 34.	Tabla de configuración LCD	89
FIGURA 35.	Display de cristal líquido	90
FIGURA 36.	Montaje para simulación LCD	92
FIGURA 37.	Manipulación de la LCD a 2 líneas	94

FIGURA 38.	<i>Manipulación LCD a 2 líneas, con visualización de dato ingresado por teclado matricial ..</i>	96
FIGURA 39.	<i>Manejo de la LCD a 4 líneas</i>	98
FIGURA 40.	<i>Simulación del motor</i>	104
FIGURA 41.	<i>Señal de control PWM para un servomotor</i>	104
FIGURA 42.	<i>Manejo de un servomotor</i>	106
FIGURA 43.	<i>Manejo de dos servomotores</i>	107
FIGURA 44.	<i>Arquitectura interna de motor paso a paso</i>	108
FIGURA 45.	<i>Acción del motor paso a paso</i>	108
FIGURA 46.	<i>Control de un motor paso a paso con control de giro y opción de velocidad alta o baja</i>	110
FIGURA 47.	<i>Encendido secuencial de 3 ledes</i>	112
FIGURA 48.	<i>Contador de eventos de dos dígitos</i>	114
FIGURA 49.	<i>Calculadora sencilla con microcontrolador utilizando manejo de teclado y visualización dinámica</i>	117
FIGURA 50.	<i>Indicador de dirección</i>	123
FIGURA 51.	<i>Indicador de dirección con control</i>	124
FIGURA 52.	<i>Funciones básicas del uso de librería para LCD</i>	130
FIGURA 53.	<i>Visualización de temperatura por LCD</i>	139
FIGURA 54.	<i>Transmisión serial de los caracteres "hola"</i>	143
FIGURA 55.	<i>Transmisión del valor de dos sensores (s0 y s1) de temperatura LM35</i>	146
FIGURA 56.	<i>Transmisión del valor de tres sensores (s0, s1 y s2) de temperatura LM35</i>	147
FIGURA 57.	<i>Registro de velocidad de giro</i>	158
FIGURA 58.	<i>Señal PWM de 1 KHz</i>	163
FIGURA 59.	<i>Comparador</i>	165
FIGURA 60.	<i>Memoria I2C</i>	170
FIGURA 61.	<i>Diagramas de la memoria 24XX01</i>	171
FIGURA 62.	<i>Almacenamiento del valor 125 en la posición de memoria 312</i>	172
FIGURA 63.	<i>Almacenamiento en memoria I2C</i>	174
FIGURA 64.	<i>Almacenamiento de cadena de caracteres PRUEBA I2C MEM</i>	176
FIGURA 65.	<i>Reloj en tiempo real DS107</i>	177
FIGURA 66.	<i>Diagrama del RTC</i>	177
FIGURA 67.	<i>Uso de la librería I2C para la manipulación de un RTC</i>	179
FIGURA 68.	<i>Medidor de frecuencia de una señal de entrada</i>	181
FIGURA 69.	<i>Monitoreo de la velocidad en rpm de un motor e ingreso de velocidad límite máxima</i>	185

FIGURA 70.	<i>Desarrollo filtro paso bajo</i>	189
FIGURA 71.	<i>Visualización de la señal del filtro</i>	189
FIGURA 72.	<i>Configuración RTC (10 de mayo de 2015)</i>	190
FIGURA 73.	<i>Descarga del software</i>	194
FIGURA 74.	<i>Link del compilador</i>	194
FIGURA 75.	<i>Ejecutable del software</i>	195
FIGURA 76.	<i>Inicio de instalación del software</i>	195
FIGURA 77.	<i>Selección del compilador</i>	196
FIGURA 78.	<i>Ejecutable del compilador</i>	196
FIGURA 79.	<i>Instalación del compilador</i>	197
FIGURA 80.	<i>Instalación del compilador paso 2</i>	198
FIGURA 81.	<i>Instalación del compilador paso 3</i>	198
FIGURA 82.	<i>Creación del proyecto</i>	199
FIGURA 83.	<i>Archivo fuente en C</i>	199
FIGURA 84.	<i>Asignación nombre</i>	200
FIGURA 85.	<i>Compilación del código</i>	200
FIGURA 86.	<i>Creación del proyecto</i>	202
FIGURA 87.	<i>Forma inicial</i>	203
FIGURA 88.	<i>Herramientas de programa 1</i>	203
FIGURA 89.	<i>Herramientas de programa 2</i>	204
FIGURA 90.	<i>Herramientas de programa 3</i>	205
FIGURA 91.	<i>Herramientas de programa 4</i>	205
FIGURA 92.	<i>Herramientas de programa 5</i>	206
FIGURA 93.	<i>Herramientas de programa 6</i>	207
FIGURA 94.	<i>Código de interacción 1</i>	207
FIGURA 95.	<i>Código de interacción 2</i>	208
FIGURA 96.	<i>Código de interacción 3</i>	209
FIGURA 97.	<i>Código de interacción 4</i>	209
FIGURA 98.	<i>Código de interacción 5</i>	210
FIGURA 99.	<i>Código de interacción 6</i>	210
FIGURA 100.	<i>Almacenamiento del proyecto</i>	211
FIGURA 101.	<i>Simulación interactiva</i>	211
FIGURA 102.	<i>Diagrama de pines de los micros de gama alta para los cuales se ejecutaron los programas realizados</i>	214

LISTA DE TABLAS

TABLA 1.	<i>Detalles de las gamas de microcontroladores de 8 bits de Microchip</i>	18
TABLA 2.	<i>Tipos de datos</i>	27
TABLA 3.	<i>Operadores aritméticos</i>	33
TABLA 4.	<i>Operadores lógicos</i>	34
TABLA 5.	<i>Operaciones lógicas básicas</i>	36
TABLA 6.	<i>Resultados de las operaciones lógicas básicas</i>	36
TABLA 7.	<i>Desplazamiento de los bits</i>	36
TABLA 8.	<i>Sentencias condicionales</i>	37
TABLA 9.	<i>Configuración ADC</i>	134
TABLA 10.	<i>Configuración de pines</i>	142
TABLA 11.	<i>Registro CCPxCON</i>	152
TABLA 12.	<i>Configuración de la preescalización de la entrada del reloj del TMR 3</i>	154
TABLA 13.	<i>Configuración de la entrada del reloj o habilitación para módulo CCP en función del TMR 1 o del TMR 3</i>	154
TABLA 14.	<i>Configuración de bits para los registros</i>	159
TABLA 15.	<i>Bits de selección de preescalización</i>	160
TABLA 16.	<i>Configuración de bits de postescalización</i>	160
TABLA 17.	<i>Configuración registro CMCON</i>	165
TABLA 18.	<i>Voltajes de referencia para registros</i>	167
TABLA 19.	<i>Configuración RTC</i>	177

INTRODUCCIÓN

Microcontroladores PIC de 8 bits

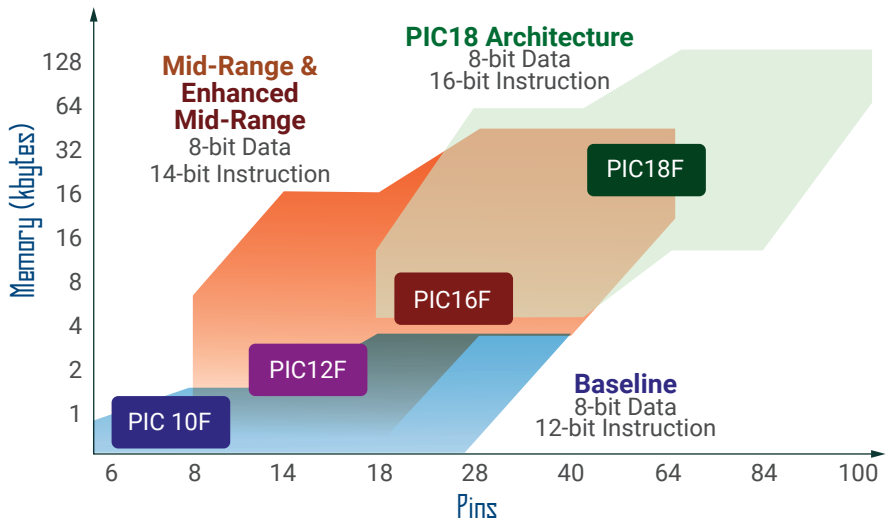
Actualmente, muchas de las actividades cotidianas implican el uso de algún equipo electrónico, de allí que se pueda afirmar que la vida moderna se enmarca en la era digital. La inclusión de la electrónica, en particular de la electrónica digital, es la que materializa este hecho al ser concebida como celulares, tabletas, computadores de escritorio, portátiles, entre otros. La electrónica digital se ve implementada típicamente mediante microcontroladores o microprocesadores, los cuales son circuitos integrados con capacidad de procesamiento básico o avanzado, que requieren ser programados, es decir, *hardware* unido con *software*.

Los conceptos y bases de programación son hoy en día parte fundamental de la formación de cualquier ingeniero. Por ello, a nivel de pregrado universitario, los primeros semestres incluyen materias de esta fundamentación, sin discriminar el tipo de ingeniería, lo que significa que el estudiante de ingeniería industrial, catastral, de sistemas, mecatrónica, de telecomunicaciones y electrónica, entre otros, debe estar en la capacidad de desarrollar algoritmos de programación. Hasta hace pocos años, en los centros de enseñanza de educación superior a nivel técnico y profesional, la electrónica digital, mediante microcontroladores, era enseñada en lenguaje de máquina, es decir, instrucciones denominadas *de bajo nivel*, que estaban orientadas a manipular casi que directamente la estructura del *hardware* del microcontrolador. En la actualidad, esta tarea se realiza con lenguajes denominados *de alto nivel*, como la programación en C, que manejan instrucciones más sintácticas orientadas a la acción más que a la manipulación del *hardware*. Este tipo de programación más simple de conceptualizar es hoy la herramienta para lograr desarrollar aplicaciones de una creciente complejidad.

Los diferentes fabricantes de microcontroladores ofrecen sus propias plataformas de programación en C para desarrollo, muchas de ellas cerradas a su casa desarrolladora, pero con licencias de uso libre. Otras plataformas propietarias permiten abordar diferentes microcontroladores de distintos fabricantes, pero bajo una licencia con costo. Por ello, y dado las regulaciones de *software* para entidades públicas y privadas, es importante elegir un fabricante de microcontrolador y su plataforma de desarrollo de uso libre, con el fin de poder orientar al futuro ingeniero en la programación y solución de problemas de ingeniería mediante este tipo de dispositivos. Microchip, como fabricante de microcontroladores reconocido en el mercado, comercializa una serie de sus dispositivos sobre los que se encuentra bastante literatura en español. La mayoría de los textos se centran en la arquitectura de 8 bits de microcontroladores de Microchip, conocidos como PIC's, los cuales se consiguen fácilmente en cualquier tienda electrónica; sin embargo, lo hacen orientados al lenguaje de máquina, ya que su compilador es de uso libre. Algunos textos de programación de microcontroladores PIC en C, como el de García-Breijo (2008), emplean compiladores con licencia bajo costo, mientras que libros como el de Jiménez-Moreno (2009), que corresponde a la versión previa de esta publicación, emplean un compilador que ya ha sido reemplazado, razón por la cual en este libro se aborda el tema de la programación de microcontroladores PIC en función del compilador (relativamente reciente) XC8 de Microchip, un compilador más general que el MCC18 del mismo fabricante, que cuenta con más de 10 años en el mercado y solo permite programar la línea de microcontroladores de referencia P18 o gama alta de Microchip, expuesto en Jiménez-Moreno (2009). El compilador XC8 de Microchip aborda todas las referencias de procesamiento a 8 bits de este fabricante, que desarrollan su programación en lenguaje C y se clasifican como P10, P12, P16 y P18.

Estas referencias se organizan en cuatro gamas diferentes: baja, media, media mejorada y alta. Se diferencian en aspectos como el número de puertos de entrada/salida de datos con que operan. Así mismo, varía la capacidad de la memoria de programa, la capacidad de la memoria de datos y la cantidad de periféricos internos con que cuentan. Esto va de la mano con el costo por unidad, por lo cual es necesario que el programador o diseñador conozca las características de la arquitectura del microcontrolador con el que ha de diseñar su solución, para que pueda elegir uno que no tenga módulos inutilizados en la aplicación a realizar, lo que, a su vez, implica que no se genere un mayor tamaño del circuito o diseño final y no se incurra en costos superiores a los necesarios. La figura 1 ilustra una relación de algunas de las diferencias entre cada una de las gamas mencionadas.

Figura 1. Gamas de microcontroladores de 8 bits de Microchip



Fuente: <https://bit.ly/3zeDhoA>

En la figura 1 se pueden apreciar las características de memoria y número de pines de las referencias de los microcontroladores de Microchip para las cuatro gamas mencionadas. La arquitectura de cualquier microcontrolador PIC posee una estructura común, basada en puertos bidireccionales que pueden actuar como entrada o salida digital (lógica) y, en varios casos, tener una segunda o incluso tercera función, ya sea una entrada analógica, un pulso de reloj, un puerto de comunicación de tipo serie, entre otras. Todas las gamas presentan una arquitectura común, cuyo procesamiento se basa en la arquitectura Harvard (Zuloaga & Astarloa, 2008), la cual consta de una memoria de datos y una de programa. Entre más capacidad de almacenamiento tenga la memoria de programa y de datos, más tareas podrá realizar el microcontrolador, y a mayor cantidad de pines, mayor interacción del microcontrolador con el exterior, lo que le permite operar elementos de entrada digital como pulsadores, interruptores DIP, teclados y sensores (análogos), así como elementos de salida digital como ledes, matrices de led, displays, pantallas LCD, dispositivos de audio (sonoros), entre otros.

La tabla 1 permite hacer una relación más detallada de las características de cada gama de microcontroladores de 8 bits.

Tabla 1. Detalles de las gamas de microcontroladores de 8 bits de Microchip

Product Family	Pin Count	Program Flash Memory (KB)	RAM (KB)	Data EE (B)	Intelligent Analog		Communications		
					ADC (# of bits)	DAC (# of bits)	USART	I2C/SPI	LIN Capable
PIC10(L)F3XX	6	384–896	0.064	HEF	8				
PIC16F152XX	8–40	3.5–28	0.5–2	–	10		1	✓	✓
PIC12/16 LF155X/6X	14–20	7–14	1.024	HEF	10(2)		✓	✓	✓
PIC16(L)F145X	14–20	14	1.024	HEF	10		✓	✓	✓
PIC1X(L)F157X	8–20	1.75–14	1.024	HEF	10	5	✓		✓
PIC16(L)F153XX	8–48	3.5–28	2.048	HEF	10	5	2	✓	✓
PIC1X(HV)F752/53	8–14	1.75–3.5	0.128	–	10	5/9			
PIC1X(L)F161X	8–14	3.5	0.256	HEF	10	8			
PIC16(L)F161X (3)	14–20	7–14	1.024	HEF	10	8	✓	✓	✓
PIC18-Q40/1	14–20	16–32	1–4	512	12(4)	8	2	✓	✓
PIC16(L)F183XX	8–20	3.5–14	2.048	256	10	5	✓	✓	✓
PIC16(L)F184XX	14–28	7–28	2.048	256	12(4)	5	✓	✓	
PIC16(L)F188XX	28–40	7–56	4.096	256	10(4)	5	✓	✓	✓
PIC18-Q10	28–40	16–128	1–3.6	256–1K	10(4)	5	2	✓	✓
PIC16(L)F191XX	28–64	14–56	4.096	256	12(4)	5	✓	✓	✓
PIC18-K40	28–64	16–128	3.728	256–1K	10(4)	5	5	✓	✓
PIC18-K42	28–48	16–128	8.192	256–1K	12(4)	5	1	✓	✓
PIC18-J94	64–100	32–128	4.096	–	12		4	✓	✓

Fuente: <https://www.microchip.com/content/dam/mchp/documents/OTH/ProductDocuments/Brochures/30010068G.pdf>

Otra referencia útil para escoger un microcontrolador es la capacidad de procesamiento mediada en instrucciones por segundo, la cual suele estar en el orden de millones de instrucciones o millones de instrucciones por segundo (MIPS). Por ejemplo, una instrucción en microcontroladores de Microchip de 8 bits se ejecuta típicamente en cuatro ciclos de reloj (para instrucciones que no impliquen saltos de memoria), tiempo que se denomina *ciclo de instrucción* (TCY), el cual se relaciona en la ecuación 1. Este tiempo

depende de la frecuencia del cristal utilizado (F_{osc}), suponiendo un cristal oscilador de $F_{osc} = 4$ MHz, que genera un pulso de reloj igual al inverso de la frecuencia ($1/F_{osc}$). El TCY se puede calcular como tiempo de instrucción en MIPS, mediante la ecuación 2. Este parámetro no solo lo determina el cristal, dado que los microcontroladores de Microchip permiten operar mediante un múltiplo de este valor, gracias a un circuito PLL interno.

Ecuación 1

$$TCY = \frac{4}{F_{osc}}$$

Ecuación 2

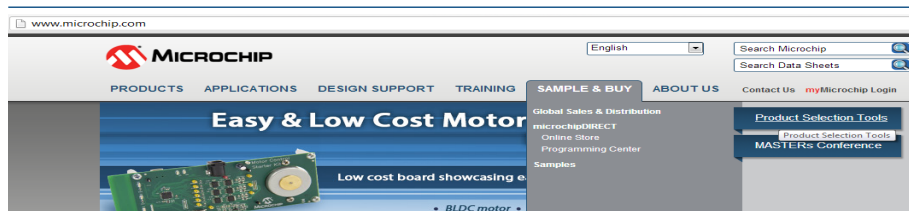
$$MIPS = \frac{1}{4 * \frac{1}{4 Mhz}} = 1$$

Hoy en día se utiliza una medida adicional para el desempeño de este tipo de dispositivos, el Dhrystone, que es un benchmark sintético, es decir, una técnica utilizada para medir el rendimiento de un sistema o componente de este. El Dhrystone está basado en estadísticas publicadas sobre el uso de particularidades de los lenguajes de programación, así como de sistemas operativos, de compiladores, entre otros. Se caracteriza porque involucra muchas instrucciones simples, llamadas a procedimiento y condicionales, y pocas instrucciones de coma flotante y bucles. No realiza llamadas al sistema y utiliza pocas variables globales ejecutando operaciones con punteros. Se compone de 12 procedimientos incluidos en un bucle de medida con 94 sentencias, del cual no se puede variar su tamaño. De forma general, se compone de un 53 % de instrucciones de asignación, un 32 % de instrucciones de control y un 15 % de llamadas a procedimiento. El Dhrystone no ocupa más de 1,5 KB de memoria y, por ser tan pequeño, entra completamente en la caché interna. El Dhrystone compara el rendimiento del procesador usando una máquina de referencia, para este caso, la VAX 11/780, la cual corre a 1 MIPS y logra 1757 Dhrystones por segundo. El lector deberá validar qué medida de desempeño prefiere emplear entre el MIPS y el Dhrystone (Patterson & Hennessy, 2011).

A lo largo del libro se presentan varios ejemplos que ilustran el uso de una u otra gama de microcontrolador, con el fin de orientar al programador hacia el diseño de sistemas microcontrolados y sobre cuándo y cómo usar una u otra

referencia según las aplicaciones a las que se enfrente. Como se mencionó, el entorno de desarrollo se basa en el compilador XC8 en su versión libre y en el *software* MPLAB® X, ambos de Microchip, lo cual no genera problemas de licenciamiento para el programador ni para los centros de enseñanza. De igual forma, en la página web de Microchip se pueden descargar estas herramientas y realizar la búsqueda de la referencia de microcontrolador adecuada, mediante la herramienta de selección de productos, como se ilustra en la figura 2.

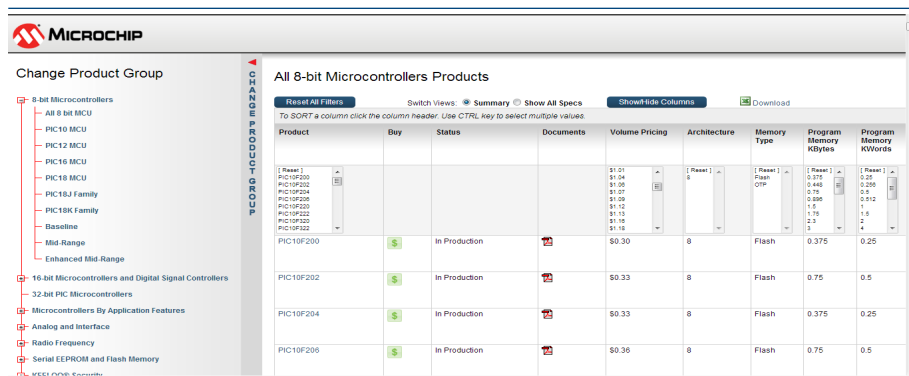
Figura 2. Herramienta de selección de productos de Microchip



Fuente: <http://www.microchip.com/>

La figura 3 ilustra cómo es posible discriminar la búsqueda por referencia de microcontrolador o por gama, relación de precio, memoria de programa y demás características y periféricos que pueda necesitar una aplicación determinada, así como los diferentes filtros que presenta la herramienta para tal fin.

Figura 3. Filtros de selección para microcontroladores



Fuente: <https://bit.ly/3hJrOHF>

Es importante que el lector tenga conocimientos básicos de electrónica digital y de lógica de programación para poder asimilar los módulos planteados tanto del microcontrolador como los externos a este y, también, cada uno de los ejemplos. El material base para el desarrollo de los programas expuestos es derivado de la información del fabricante por medio de los *datasheet* que facilita en su página web y que se relacionan en la infografía al final de este documento.



