

Programación de Microcontroladores PIC en lenguajes de alto nivel : PicBasic, PicBasic Pro y C.

Dr. Eugenio Martín Cuenca
Facultad de Ciencias. Universidad de Granada.
E-mail:emartin@goliat.ugr.es

Iniciamos este mes la andadura de una nueva sección dedicada principalmente a la programación y empleo de los microcontroladores **PIC**, así como sus aplicaciones en el control, la robótica, bioingeniería, etc.

Los Microcontroladores

Todos consciente o inconscientemente, a diario hacemos uso de los microcontroladores. Estos computadores (en un solo CHIP), los más pequeños del mundo, han invadido nuestros hogares, e incluso nuestros bolsillos, encontrándose en la mayoría de los dispositivos de uso cotidiano, en nuestras tarjetas, en los futuros monederos electrónicos, en los teléfonos móviles, en el mando a distancia, en el microondas, en la encimera de la cocina, en el encendido electrónico de nuestro automóvil y en cualquier sitio por más inverosímil que nos parezca.

Su misión sin embargo no es sólo la de hacernos la vida más fácil a las gentes de a pie, también controlan los procesos industriales, los sistemas de navegación de los aviones, y otros procesos complejos.

A lo largo de esta sección, el lector aprenderá a programarlos en lenguajes de alto nivel como el **BASIC** y el **C**. La programación y empleo de los microcontroladores se expondrá con multitud de ejemplos prácticos. Se construirá desde un pequeño robot hexápodo programable, un brazo articulado, una estación meteorológica como se presentan en las fotografías y multitud de otros pequeños montajes.

Los lectores que deseen profundizar en el tema de los microcontroladores PIC, y su programación en Ensamblador pueden acudir a los libros indicados en la bibliografía.



Foto 1.- Pequeño robot que podrá construir y programar en BASIC.

Nos encontramos en este punto, en un caso paralelo al que ocurre con los hinchas de un equipo de fútbol. Cada diseñador piensa que los microcontroladores que usa son los más idóneos, siendo en muchos casos como una religión, al igual que en el mundo de los ordenadores pasa entre los entusiastas de los MAC y PC.

Mis inicios, allá por 1982, se cimientan en el VIC 20 y otros ordenadores de Commodore basados en el 6502 y los Sinclair QL con un microprocesador 68008 de Motorola. De ahí, de alguna forma el “*aprecio sentimental*” hacia los PIC. Aunque hoy día sea un usuario de la familia x86 y Windows, esta traición a los colores ha sido inevitable dada la evolución de la informática actual.

Programación en BASIC de los Microcontroladores PIC

Es por este motivo, que vamos a tratar de ser imparciales y aunque el enfoque principal de la sección se destina a los microcontroladores **PIC**, no se dejarán de lado otros microcontroladores empleados asiduamente. Entre otros se encuentran los **8051** de Intel, familia de gran solera, y su saga de compatibles, los Motorola **68HC11** y los nuevos **AVR** de Atmel y los potentes **Scenix SX** que son clónicos de los PIC pero con una potencia de **50** y **100 MIPS**. También a título de promesa, al igual que para los PIC, se han desarrollado las herramientas de trabajo y su documentación, trataremos de hacer lo mismo con los demás microcontroladores que se presenten, siempre dentro de la disponibilidad de tiempo.

Por otro lado, para facilitar tanto la programación de los microcontroladores como la realización de los montajes, estos se presentarán en forma de **KITs** que pueden conseguirse con relativa facilidad. Muchos de estos **KITs** y los grabadores han sido desarrollados por el autor y están disponibles, otros se han localizado expresamente para esta sección, y en el caso de que algún montaje propuesto no estuviese comercializado, los lectores podrán solicitar al autor las placas de circuito impreso necesarias para su montaje.



Foto 2.- Anemómetro de la estación meteorológica que se describirá.



Foto 3.- Brazo robot articulado.

¿ Porqué los PICs ?

El autor de esta serie de artículos, trabaja desde 1982 con microprocesadores, realizando diseños de automatización y control. Hace unos años con motivo del diseño y construcción de un sistema de control para Piscifactorías, realizó un exhaustivo estudio de los microcontroladores disponibles en el mercado, eligiendo finalmente los microcontroladores PIC.

Aunque existen otras familias de microcontroladores muy interesantes, los PIC, tienen “algo” que nos fascina. Puede ser su velocidad, su precio, la facilidad de uso, la gran cantidad de información disponible, lo asequible de las herramientas de desarrollo. En la actualidad, el autor ha diseñado una serie de **KITs**, incluyendo un programador para múltiples tipos de PICs y están disponibles a un precio verdaderamente económico. Todo esto produce una imagen global de sencillez y utilidad.

Las Familias de Microcontroladores PIC

Los PIC son una extensa familia de microcontroladores divididos en tres gamas, la gama baja con cuerpo de 12 bits, la gama media y la gama alta con cuerpo de 14 bits. En la **Tabla 1**, se presentan algunos miembros de cada una de estas gamas con un resumen de sus principales características.

Debido al espacio reducido de una publicación periódica, y con el objeto de guiar certeramente al lector, el curso se basará en el **PIC16F84** por sus especiales características, pero todo lo que se indique sobre él, puede ser aplicado con mínimas modificaciones a cualquier miembro de la familia PIC. Así el aprender a manejar uno de ellos, capacita para trabajar con cualquier miembro de la familia con un esfuerzo mínimo.

Los **PIC-16F84** y **PIC-16C84**, son microcontroladores que ofrecen muchas ventajas al diseñador. Están basados en memoria FLASH/EEPROM que al permitir su borrado y grabado eléctricamente de forma casi instantánea, los hace potentes, flexibles y fáciles de emplear. Pueden utilizarse tanto en el desarrollo como en la producción. Su coste se reduce respecto a otros modelos pues no necesitan encapsulado cerámico con ventana de programación como los chips EPROMs.

Disponen además de una EEPROM de 64 bytes para datos, lo que elimina la necesidad de una memoria EEPROM externa, y aun más importante, pueden ser programados “*en circuito*” sin necesidad de ser retirados, lo que permite su programación después del ensamblado y que los programas puedan ser actualizados o modificados para eliminar errores detectados con posterioridad o para introducir nuevas versiones más actualizadas del programa.

	MEMORIA EPROM / EEPROM / FLASH (Palabras de 14 bits)	MEMORIA RAM (Bytes)	PINES DE E/S	TEMPORIZADORES TMR0 - 8 BITS TMR1 - 16 BITS TMR2 - 8 BITS	FUENTES DE INTERRUPCIÓN	COMPARADORES ANALÓGICOS	MÓDULO DE TENSIÓN DE REFERENCIA	DIRECCIÓN DE FALLO DE ALIMENTACIÓN	CONVERTIDOR A/D (CANALES)	MÓDULOS PWM	PUERTOS SERIE SÍNCRONOS SPI - I ² C ASÍNCRONOS SCI - UART	PUERTO PARALELO SECUNDARIO
PIC12C508	512 B	25	6	TMRO	---	---	---	---	---	---	---	---
PIC12C509	1K	41	6	TMRO	---	---	---	---	---	---	---	---
PIC14000	4K	192	22	TMRO, ADTMR	11	2	---	SI	14	---	I ² C / SMBUS	---
PIC16C84	1K	36	13	TMRO	4	---	---	---	---	---	---	---
PIC16F84	1K	68	13	TMRO	4	---	---	---	---	---	---	---
PIC16C71	1K	36	13	TMRO	4	---	---	---	4	---	---	---
PIC16C74	4k	192	33	TMRO, TMR1, TMR2	12	---	---	---	8	2	SPI/I ² C/SCI	SI
PIC16C73	4K	192	22	TMRO, TMR1, TMR2	11	---	---	---	5	2	SPI/I ² C/SCI	---
PIC17C44	8K	454	33	TMRO, TMR1, TMR2	11	---	SI	SI	SI	2	SPI/I ² C/SCI	SI

Tabla 1.- Resumen de las características de algunos de los microcontroladores PIC de la tres gamas.

La rueda de reconocimiento. Los sujetos bajo sospecha.

En la foto 4 se presentan algunos de los sujetos en la rueda de reconocimiento. De derecha a izquierda se encuentra el BS1-IC y el BS2-IC de Parallax. En los artículos se trabajará además con microcontroladores PIC 16F8XX montados sobre nuestros propios módulos.

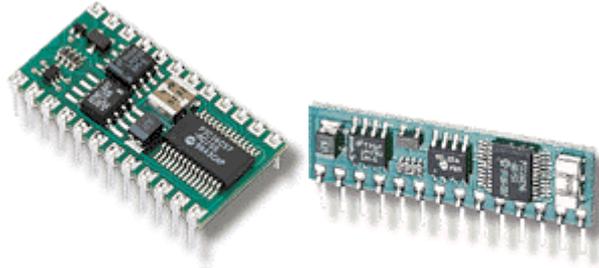


Foto 4 .- Módulos BS2-IC y BS1-IC de la empresa americana *Parallax*.

Se preguntará el lector, ¿ No estábamos hablando solo de PIC ?. Si, pero en este punto hagamos un poco de historia.

PBASIC de Parallax BS1-IC. Compilador PIC BASIC (PBC) de MicroEngineering Labs.

La empresa *Parallax* fue la primera en desarrollar un interprete BASIC para los microcontroladores PIC. Lo incluyó en una pequeña tarjeta para construcción de prototipos denominada **BASIC Stamp 1** (BS-1). El producto se compone de un PIC16C56 funcionando a 4 MHz, una memoria EEPROM serie, la 93LC56 y un circuito de “brownout”. Deja disponibles para el usuario ocho líneas de E/S de propósito general y 256 bytes de memoria de programa, capaces de contener unas 100 instrucciones.

El **interprete PBASIC** integra un editor y un programa que permite el volcado sobre la tarjeta BS1-IC desde el puerto paralelo del PC. Aunque su gran ventaja es la flexibilidad de un interprete, las desventajas que presenta el PBASIC son principalmente dos: es lento, pues solo ejecuta 2.000 instrucciones por segundo y sólo puede ser empleado con la tarjeta Parallax BS1-IC.

Ante este estado de cosas, la compañía *MicroEngineering Labs* diseñó el **compilador PBC** (PIC BASIC Compiler) totalmente compatible con el interprete PBASIC de Parallax, pero con la ventaja del gran aumento de velocidad que proporciona un compilador además de poder escribir programas más largos.

El compilador PBC permite los escribir programas desarrollados para el interprete BASIC Stamp I, compilarlos y ejecutarlos directamente en microcontroladores PIC. Sus principales características son :

- Juego de instrucciones compatible con BASIC Stamp I, además de algunas ampliaciones.
- Verdadero compilador que permite la ejecución de los programas con mayor velocidad que el interprete BASIC. En algunas operaciones es hasta 100 veces más rápido.
- Comandos **I2CIN** e **I2COUT** que permiten el acceso a cualquier periférico **I2C**.
- Mayor numero de variables de usuario, hasta 80 en total.
- Instrucciones PEEK y POKE para acceder a cualquier registro del PIC desde el BASIC.
- Velocidad serie 9600 baudios.
- Soporta segmentos de código en ensamblador y CALL.
- Puede usarse con los microcontroladores PIC12C76x, PIC14C4000, PIC16C55x, PIC16C6xx, PIC16C7xx, PIC 16c92x, PIC18Cxx y PIC18Fxx.

El compilador PBC convierte los programas BS1-IC, en ficheros *.HEX o binarios que pueden ser grabados directamente en los microcontroladores PIC. El fácil uso del lenguaje BASIC coloca la programación de los microcontroladores PIC al alcance de todos.

Por último, indicar, que también nos ha inclinado a su elección, el desarrollo por parte de la empresa *Micromint*, de la serie de módulos PICStic, el PS-1, el PS-2, y el PS-3. A un precio muy económico, estos módulos ofrecen un 16C84, con toda la circuitería necesaria, y incluye además en los casos del PS-2 un reloj en tiempo real y el PS-3 un conversor A/D de 12 bits. Estos módulos pueden programarse en ensamblador, en PBC, o en cualquier lenguaje disponible para los PIC.

PBASIC II de Parallax para el BS2-IC. Compilador PIC BASIC Profesional (PBC PRO) de MicroEngineering Labs.

Con posterioridad *Parallax* desarrolló el **BASIC STAMP II** (BS2-IC), que ampliaba tanto el hardware como el conjunto de instrucciones BASIC del BS1-IC, llamado ahora PBASIC2.

El hardware del BS2-IC es un PIC16C57 a 20 MHz, una memoria serie EEPROM 24LC16 y un circuito de “*brownout*” dejando disponibles para el usuario 16 líneas de E/S de propósito general y otras dos para comunicaciones serie, 2 Kbytes de memoria de programa, capaces de contener unas 600 instrucciones. Ejecuta de 3.000 a 4.000 instrucciones por segundo. A principios de este año Parallax ha presentado un nuevo modelo el BS2SX-IC basado en un microcontrolador Scenix SX28 que puede ejecutar 10.000 instrucciones por segundo, una memoria de programa 16Kb que reemplaza al BS2-IC original.

Aquí se suma a su gran ventaja es la flexibilidad de un interprete, la mayor capacidad de memoria, mayor velocidad y un juego de instrucciones más completo.

Por supuesto como cabría esperar, *MicroEngineering Labs*, ha desarrollado un compilador compatible con **PBASIC2**, llamado **PIC BASIC Pro Compiler**. Sus principales características son :

- Juego de instrucciones compatible con BASIC Stamp I y BASIC Stamp II con algunas ampliaciones.
- Verdadero compilador que permite la ejecución de los programas con mayor velocidad que el interprete BASIC, en algunas operaciones hasta 100 veces más rápido.
- Acceso directo o desde librería a cualquier registro o PIN del microcontrolador.
- Arrays tipo bit, byte y word.
- Control automático del cambio de página de memoria.
- Velocidad serie hasta 19.200 baudios
- Soporta segmentos de código en ensamblador y CALL.
- Verdadera instrucción IF .. THEN .. ELSE .. ENDIF.
- Interrupciones en BASIC y ensamblador.
- Puede usarse con los microcontroladores PIC12C76x, PIC14C4000, PIC16C55x, PIC16C6xx, PIC16C7xx, PIC 16c92x, PIC18Cxx y PIC18Fxx.
- Control directo de pantallas LCD mediante un simple comando.
- Soporta velocidades de reloj desde 3,58 a 20 MHz.

Programación en BASIC de los Microcontroladores PIC

Tabla 2.- Resumen comparativo de las instrucciones de los diferentes BASIC.

PBASIC	PBC	PBASIC2	PBC Pro.	Función
----	----	----	@	Inserta una línea de ensamblador.
----	ASM....	----	ASM....	Las instrucciones que siguen van en lenguaje ensamblador.
----	ENDASM	----	ENDASM	Salida del modo de instrucciones en lenguaje ensamblador.
BRANCH	BRANCH	BRANCH	BRANCH	GOTO calculado. Equivalente a ON GOTO.
----	----	----	BRANCHL	Salto largo fuera de la página.
BUTTON	BUTTON	BUTTON	BUTTON	Elimina rebotes y autorepite un PIN específico.
----	CALL	----	CALL	Llama una subrutina en lenguaje ensamblador.
----	----	----	CLEAR	Pone a cero todas las variables.
----	----	COUNT	COUNT	Cuenta el número de pulsos en un PIN.
----	----	DATA	DATA	Define el contenido inicial de la EEPROM del PIC.
DEBUG	----	DEBUG	DEBUG	Se ignora en PBC.
----	----	----	DISABLE	Desactiva el proceso ON INTERRUPT.
----	----	DTMFOUT	DTMFOUT	Produce un pulso de tono en un PIN.
EEPROM	EEPROM	EEPROM	EEPROM	Define el contenido inicial de la EEPROM.
----	----	----	ENABLE	Activa el proceso ON INTERRUPT.
END				Detiene la ejecución del programa. Pasa a bajo consumo.
FOR.. NEXT	FOR.. NEXT	FOR.. NEXT	FOR.. NEXT	Bucle iterativo.
----	----	FREQOUT	FREQOUT	Produce hasta 2 frecuencias en un PIN.
GOSUB	GOSUB	GOSUB	GOSUB	Llamada a la subrutina especificada en la etiqueta.
GOTO	GOTO	GOTO	GOTO	Continúa la ejecución en una etiqueta especificada.
HIGH		HIGH	HIGH	Coloca un PIN de salida a nivel alto.
----	I2COUT	----	----	Salida comunicación con dispositivo I2C.
----	I2CIN	----	----	Entrada comunicación con dispositivo I2C.
----	----	----	I2CREAD	Lee bytes de un dispositivo I2C.
----	----	----	I2CWRITE	Escribe bytes en un dispositivo I2C.
----	----	----	HSERIN	Entrada serie asíncrona hardware.
----	----	----	HSEROUT	Salida serie asíncrona hardware.
IF..THEN	IF..THEN	IF..THEN	IF..THEN	Salta si la condición es verdadera.
----	----	----	IF..THEN..ELSE ..END IF	Ejecuta instrucciones condicionales.
INPUT	INPUT	INPUT	INPUT	Coloca el PIN como entrada.
----	----	----	LCDOUT	Muestra caracteres en un LCD.
LET	LET	LET	LET	Ejecuta una operación matemática y asigna el resultado a una variable.
LOOKDOWN	LOOKDOWN	LOOKDOWN	LOOKDOWN	Busca un valor en una tabla.
----	----	LOOKDOWN2	LOOKDOWN2	Busca el valor de una variable o una constante en una tabla.
LOOKUP	LOOKUP	LOOKUP	LOOKUP	Toma un valor de la tabla.
----	----	LOOKUP2	LOOKUP2	Toma el valor de una variable o una constante de una tabla.
LOW	LOW	LOW	LOW	Coloca un PIN de salida a nivel bajo.
NAP	NAP	NAP	NAP	Coloca el procesador en bajo consumo durante un corto período de tiempo.
----	----	----	ON INTERRUPT	Ejecuta una subrutina BASIC en una interrupción.
OUTPUT	OUTPUT	OUTPUT	OUTPUT	Coloca un PIN como salida.
PAUSE	PAUSE	PAUSE	PAUSE	Retardo. Resolución un ms.
----	----	----	PAUSEUS	Retardo. Resolución 1µs.
----	PEEK	----	PEEK	Lee un byte de un registro del PIC.
----	POKE	----	POKE	Escribe un byte en un registro del PIC.
POT	POT	POT	POT	Lee un potenciómetro en el PIN especificado.
PULSIN	PULSIN	PULSIN	PULSIN	Mide el ancho de un pulso (10µs de resolución).
PULSOUT	PULSOUT	PULSOUT	PULSOUT	Genera un pulso (10µs de resolución).
PWM	PWM	PWM	PWM	Saca un tren de pulsos PWM por una patita.
RANDOM	RANDOM	RANDOM	RANDOM	Genera un número pseudo-aleatorio.

Tabla 2.- Continuación.

PBASIC	PBC	PBASIC2	PBC Pro.	Función
----	----	RCTIME	RCTIME	Mide el ancho de un pulso en un PIN.
READ	READ	READ	READ	Lee un byte de la EEPROM.
----	----	RESUME	RESUME	Continúa la ejecución después de una interrupción.
RETURN	RETURN	RETURN	RETURN	Reanuda la ejecución de la instrucción que sigue al GOSUB.
REVERSE	REVERSE	REVERSE	REVERSE	Coloca un PIN que estaba como salida como entrada o viceversa.
SERIN	SERIN	SERIN	SERIN	Entrada serie asíncrona (N81).
----	----	SERIN2	SERIN2	Entrada serie asíncrona.
SEROUT	SEROUT	SEROUT	SEROUT	Salida serie asíncrona (N81).
----	----	SEROUT2	SEROUT2	Salida serie asíncrona.
----	----	SHIFTIN	SHIFTIN	Entrada serie síncrona.
----	----	SHIFTOUT	SHIFTOUT	Salida serie síncrona.
SLEEP	SLEEP	SLEEP	SLEEP	Coloca en procesador en modo de bajo consumo SLEEP, con un segundo de resolución.
SOUND	SOUND	SOUND	SOUND	Genera un tono o ruido-blanco en el PIN especificado.
----	----	STOP	STOP	Detiene la ejecución del programa.
			SWAP	Intercambia el valor de dos variables.
----	----	----	WHILE.. WEND	Ejecuta las instrucciones hasta que la condición es verdadera.
TOGGLE	TOGGLE	TOGGLE	TOGGLE	Coloca un PIN como salida y conmuta su estado.
WRITE	WRITE	WRITE	WRITE	Escribe un byte en la EEPROM.
----	----	----	XIN	Entrada X-10.
----	----	XOUT	XOUT	Salida X-10.