

Curso **Básico** de PICS y Picaxe

Estimado Alumno :

Vamos a introducirnos con este apunte en el apasionante mundo de la programación de los microcontroladores. Primero vamos a realizar una breve descripción de que son los PICS, su arquitectura, descripción de sus pines y su memoria a efectos de tener una idea de cómo funciona internamente. Luego instalaremos un programa gratuito de la empresa MicroChips para programar este microcontrolador con un lenguaje nativo (lenguaje propio del microcontrolador) lo cual como verán mas adelante no es muy cómodo para usuarios principiantes así que no lo vamos a usar pero es importante para que sepan de las herramientas que brinda esta empresa especialmente a la gente de la orientación Informática.

Un microcontrolador es un tipo de computadora en miniatura que se encuentra en muchos aparatos. Algunos ejemplos de productos comunes de uso cotidiano que contienen microcontroladores se muestran en la figura 1-1. Si estos tienen botones y pantallas digitales lo más seguro es que tengan como “cerebro” un microcontrolador programable.



Figura 1-1
Ejemplos cotidianos de
aparatos que tienen
microcontroladores.

Estos son algunos ejemplos, si su radio reloj se apaga, y presiona varias veces en la mañana la perilla de su aparato, la primera acción de ese día es interactuar con un microcontrolador. Cuando calienta alimentos en el horno de microondas y hace una llamada con su teléfono celular también está interactuando con un microcontrolador. Esto solo es el comienzo. Veamos otros ejemplos. Cuando enciende el TV con el control remoto, cuando juega con su Nintendo y cuando usa una calculadora, estos aparatos tienen internamente un microcontrolador que interactúa con usted.

Esta etapa es para poder brindarles una comparación entre lo que son los **pics** y los **picaxe** que es lo que finalmente usaremos. Los **picaxe** son muy sencillos de programar con diagramas de flujo y se necesita elementos muy sencillos para cargarlos e implementarlos en algún proyecto que Ud elija.

ESTRUCTURA Y CARACTERISTICAS DE UN PIC

¿QUÉ ES UN PIC?

Los circuitos integrados programables (Programmable Integrated Circuits = PIC) son componentes sumamente útiles en la Electrónica de Consumo. Aun cuando son conocidos desde hace más de veinte años, existen en la actualidad nuevos tipos que cumplen con una serie de requisitos y características sumamente útiles. Como una primera aproximación podemos definir a un PIC como “un chip que me permite obtener un circuito integrado a mi medida”, es decir puedo hacer que el PIC se comporte como un procesador de luminancia o un temporizador o cualquier otro sistema mediante un programa que le grabo en una memoria ROM interna.

Los microcontroladores PIC son, en el fondo, procesadores similares a otros tipos, como por ejemplo la familia de los microprocesadores X86, 80486, Pentium y muchos otros que usan una arquitectura interna del tipo Von Neumann. En este tipo de arquitectura los datos y la memoria del programa se encuentran en el mismo espacio de direcciones.

En realidad, un microprocesador y un microcontrolador no son la misma cosa. Los PICs son microcontroladores, es decir, una unidad que posee en su interior al microprocesador y a los elementos indispensables para que pueda funcionar como una minicomputadora en un solo chip.

Un microprocesador es solamente la unidad central de procesos o CPU, la memoria, los puertos y todos los demás periféricos son exteriores. La programación de un microprocesador es, por lo tanto, una tarea compleja porque deben controlarse todos estos dispositivos externos.

Un microcontrolador integra la CPU y todos los periféricos en un mismo chip. El programador se desentiende de una gran cantidad de dispositivos y se concentra en el programa de trabajo. Esta circunstancia da lugar a una gran pérdida de tiempo porque los datos tienen que ser retirados de la memoria y llevados a la CPU (Central Processor Unit) y viceversa. Esto significa que la computadora dedica la mayor parte del tiempo al transporte de datos de ida o de vuelta, en lugar de usar este tiempo para trabajar sobre los datos.

Los PICs emplean un conjunto de instrucciones del tipo RISC (Reduced Instruction Set Computer). Con el RISC se suele ejecutar la mayoría de las instrucciones con un solo pulso del clock. Con las instrucciones que se usan en otros equipos del tipo CISC (Complex Instruction Set Computer), se logran instrucciones más poderosas, pero a costa de varios ciclos del clock. En el bien conocido procesador 68HC11 de Motorola se requieren a veces hasta 5 ciclos del clock para ejecutar una instrucción.

A los fines prácticos, nos vamos a referir a los microcontroladores como bloques que poseen una memoria de programa, que es el lugar donde deben alojarse los datos que le indiquen al chip qué es lo que debe hacer; una memoria de datos donde ingresen las señales que debe procesar el programa, una unidad aritmética y lógica donde se desarrollen todas las tareas, una unidad de control que se encargue de supervisar todos los procesos y puertos de entrada y salida para que el PIC tenga contacto con el exterior.

Diferencia entre **microprocesador** y **microcontrolador**.

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada **procesador**, de un computador. La UCP está formada por la Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la **Memoria** y los Módulos de E/S y configurar un computador implementado por varios **circuitos** integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine. (Figura 1.1.)

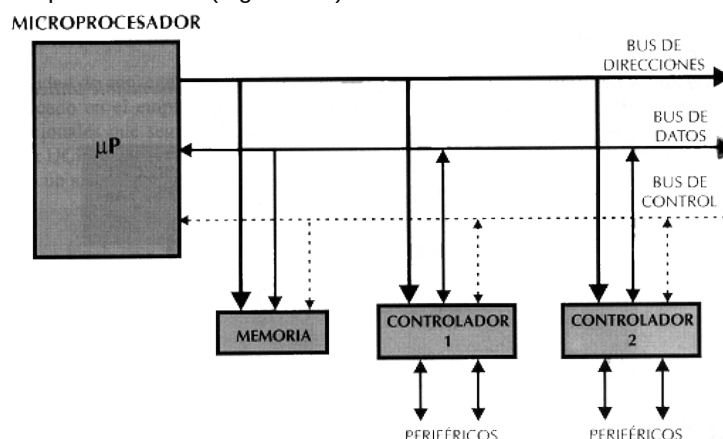


Figura 1.1. **Estructura** de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación.

Si sólo se dispusiese de un **modelo** de microcontrolador, éste debería tener muy potenciados todos sus **recursos** para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores **oferta** un elevado número de **modelos** diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las **memorias**, el número de líneas de E/S, la cantidad y **potencia** de los elementos auxiliares, la **velocidad** de funcionamiento, etc. Por todo ello, un aspecto muy destacado del **diseño** es la **selección** del microcontrolador a utilizar.

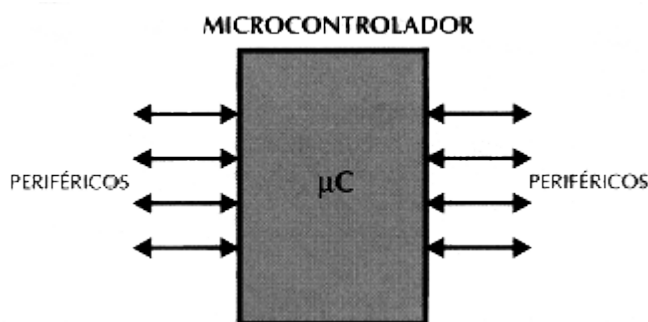


Figura 1.2. El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

1 Introducción a los pics

Con este tutorial se pretende introducir al lector en el mundo de la programación de [microcontroladores](#) PIC de forma práctica y sencilla. Doy por supuestos unos conocimientos muy básicos sobre electrónica digital. Hablaremos de Instrucciones, registros de [memoria RAM](#), [memoria EEPROM](#) (un tipo de [ROM](#)), de niveles lógicos "0" o "1" y cosas por el estilo.

El PIC16F84 es un [microcontrolador](#), una especie de "ordenador en miniatura" (con muchas comillas) que podremos programar. En su interior posee un [microprocesador](#), una [memoria RAM](#) (volátil) donde guardaremos las variables, una [memoria EEPROM](#) (no volátil) donde guardaremos nuestro programa, un Timer o contador que nos facilitará algunas tareas, y alguna cosilla mas...

Algunas características mas representativas son:

	Opera a una frecuencia máxima de 10 MHz
	1Kbyte de memoria EEPROM para nuestro programa
	68 bytes (de 8 bits) de memoria RAM
	64 bytes de memoria EEPROM para datos (no volátiles)
	Solo 35 instrucciones
	13 pines de entrada/salida (un puerto de 8 bits + otro de 5 bits)
	Timer/contador de 8 bits
	..

2. Descripción de sus Pines

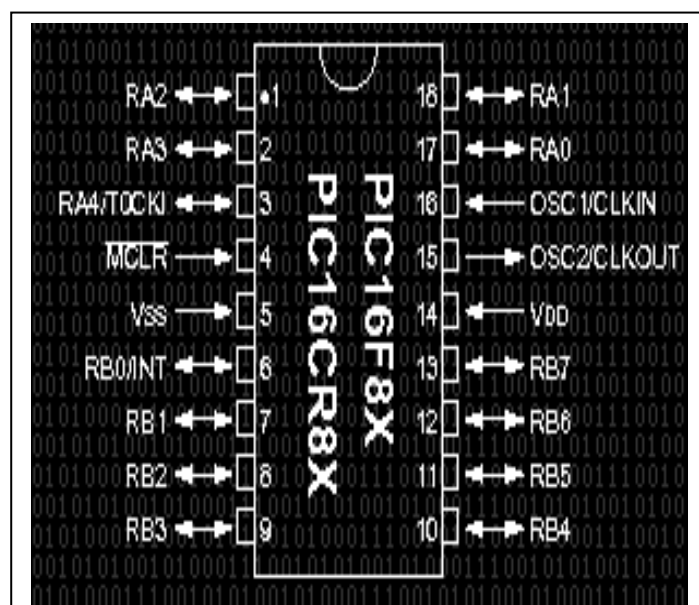
RA0, RA1, RA2, RA3 y RA4: son los pines del puerto A

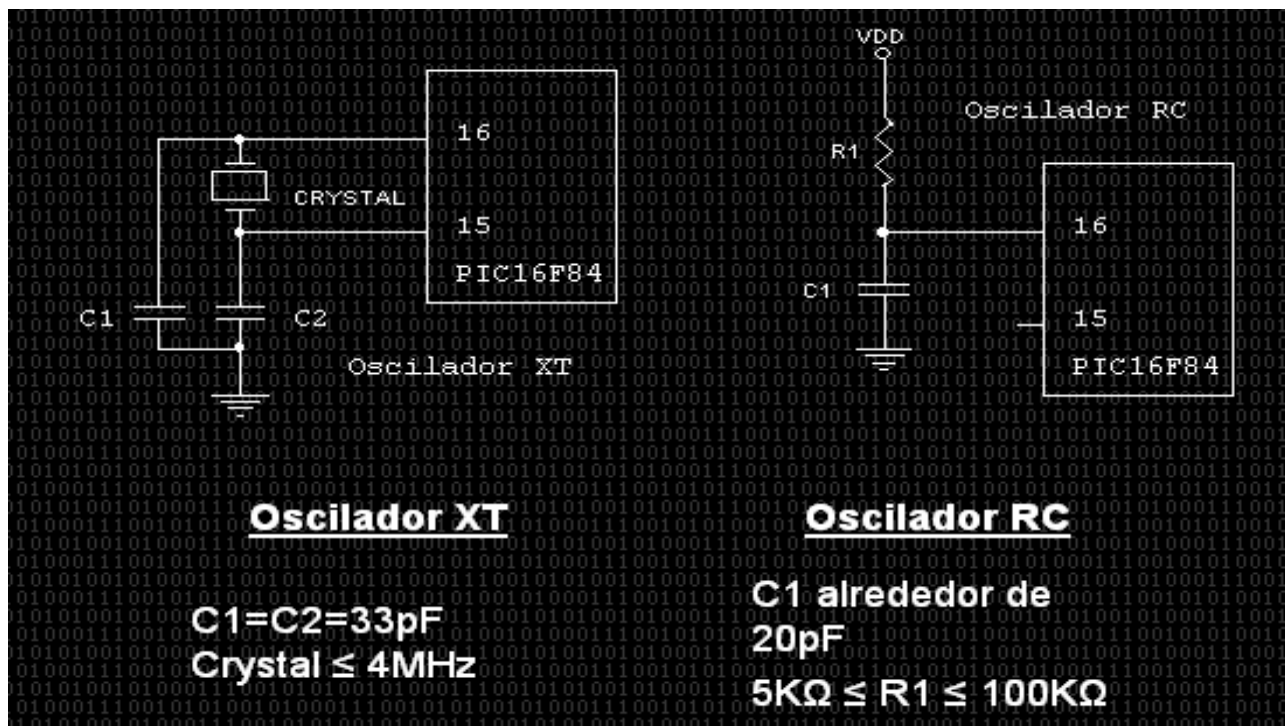
RB0, RB1, RB2, RB3, RB4, RB5, RB6 y RB7: son los pines del puerto B.

MCLR: Pin de reseteo del pic, cuando se pone a "0" el pic se resetea.

Vdd y Vss: pines de alimentación (Vdd 5V y Vss a masa)

OSC1/CLKIN y OSC2/CLKOUT: son para el [oscilador](#) Los tipos de [osciladores](#) mas usados son el XT (cristal de cuarzo) y el RC ([resistencia](#) y [condensador](#)) el modo de conexionado es el siguiente:





3. Organización de la memoria

En primer lugar tenemos que distinguir claramente entre tres tipos de memoria:

1	Una: la memoria EEPROM donde almacenaremos el programa que haremos, esta memoria solo podrá ser leída por el pic (el pic va leyendo las instrucciones del programa almacenado en esta memoria y las va ejecutando). Al apagar el pic esta memoria no se borra.
2	Dos: la memoria RAM en cuyos registros se irán almacenando los valores de las variables que nosotros queramos y cuando nosotros queramos (por programa), al apagar el pic esta memoria se borra.
3	Tres: la memoria EEPROM para datos, es un espacio de memoria EEPROM en la que se pueden guardar variables que queremos conservar aunque se apague el pic. No se tratará aquí por ser una memoria mas difícil de emplear.

Ejemplo de programa bien ordenado:

```

;*****
;* EJEMPLO 1: PROGRAMA BIEN ORDENADO*
;*****
;* El siguiente programa configura  *
;* RA1 como entrada y RA0 como    *
;* salida y hace que la salida (RA0)*
;* sea la inversa de la entrada    *
;* (RA1)                            *
;*****

;(Conviene poner título y una
;pequeña explicación de lo que
;hace el programa)

;(Primero los ECU y los #DEFINE)

STATUS EQU    03H
TRISA EQU    05H
PORTA EQU    05H

#DEFINE BANCO0 BCF    STATUS,5
#DEFINE BANCO1 BSF    STATUS,5

;(Después empezamos con el programa)

        ORG      00H      ;Empezamos siempre a escribir en esta dirección
BANCO1   ;Pasamos al banco 1 para hacer algunas
        ;configuraciones
        BCF      TRISA,0 ;Configuramos RA0 como salida
        BSF      TRISA,1 ;Configuramos RA1 como entrada
BANCO0   ;Volvemos al banco 0

INICIO  BTFSC    PORTA,1 ;Comprueba la entrada (RA1), si es "0" se salta la
        ;siguiente instrucción
        GOTO     ESUNO   ;si la entrada (RA1) es "1" va a ESUNO

        BSF      PORTA,0 ;Pone a "1" la salida RA0. Ejecuta esta instrucción
        ;porque la entrada RA1 era "0"
        GOTO     INICIO  ;Vuelve otra vez a comprobar el estado de la
        ;entrada RA1

ESUNO   BCF      PORTA,0 ;Pone a "0" la salida RA0. Ejecuta esta instrucción
        ;porque la entrada RA1 era "1"
        GOTO     INICIO  ;Vuelve otra vez a comprobar el estado de la
        ;entrada RA1

        END          ;Indica final del programa

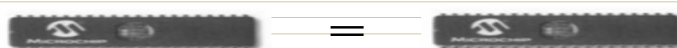
```

LOS MICROCONTROLADORES PICAXE:

UN SISTEMA MUY FÁCIL DE PROGRAMAR QUE NO PRECISA DE COSTOSOS CARGADORES Y EN LOS QUE EL PROGRAMA PUEDE HACERSE EN BASIC O EN DIAGRAMA DE FLUJO.

¿Que diferencia existe entre un PICAXE y un PIC?

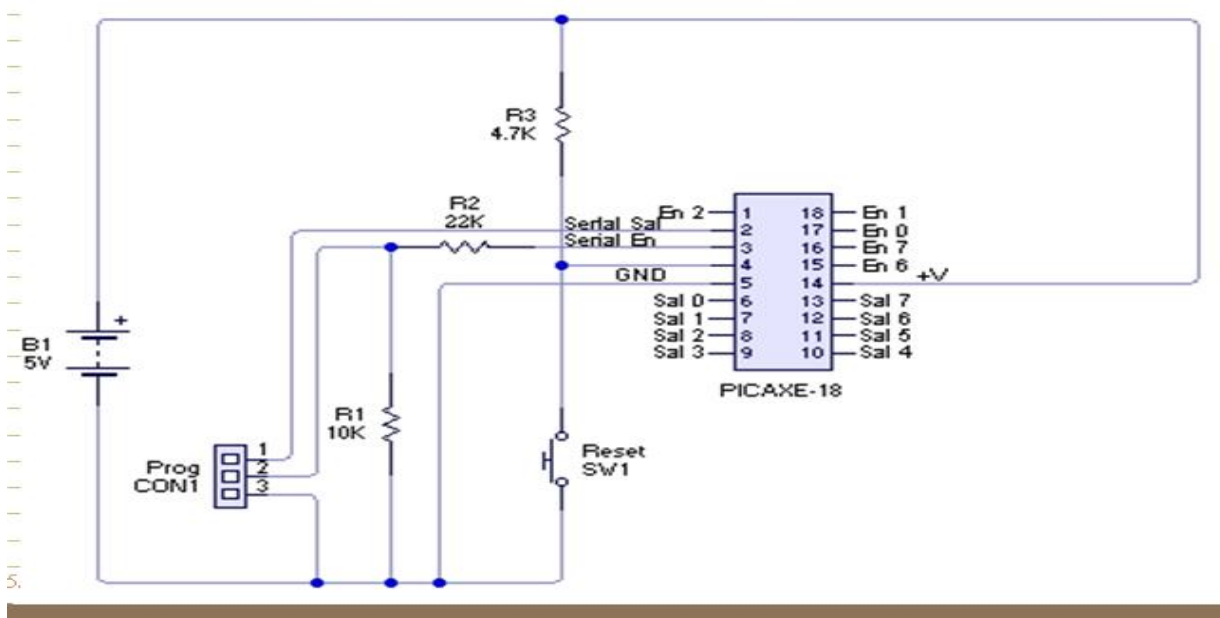
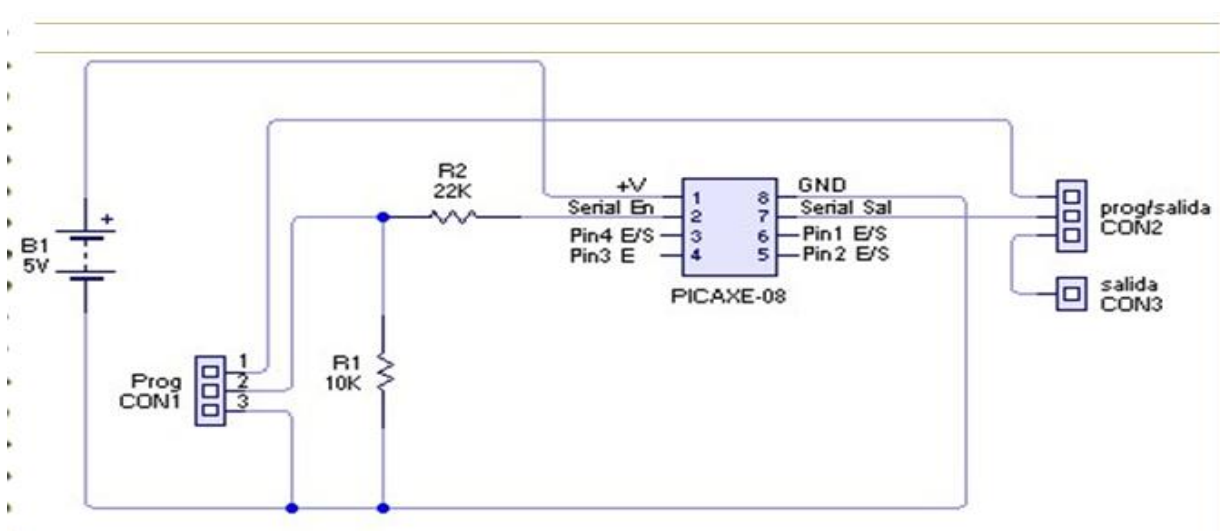
Física: Ninguna



Operativa : Algunas



5/10/2013

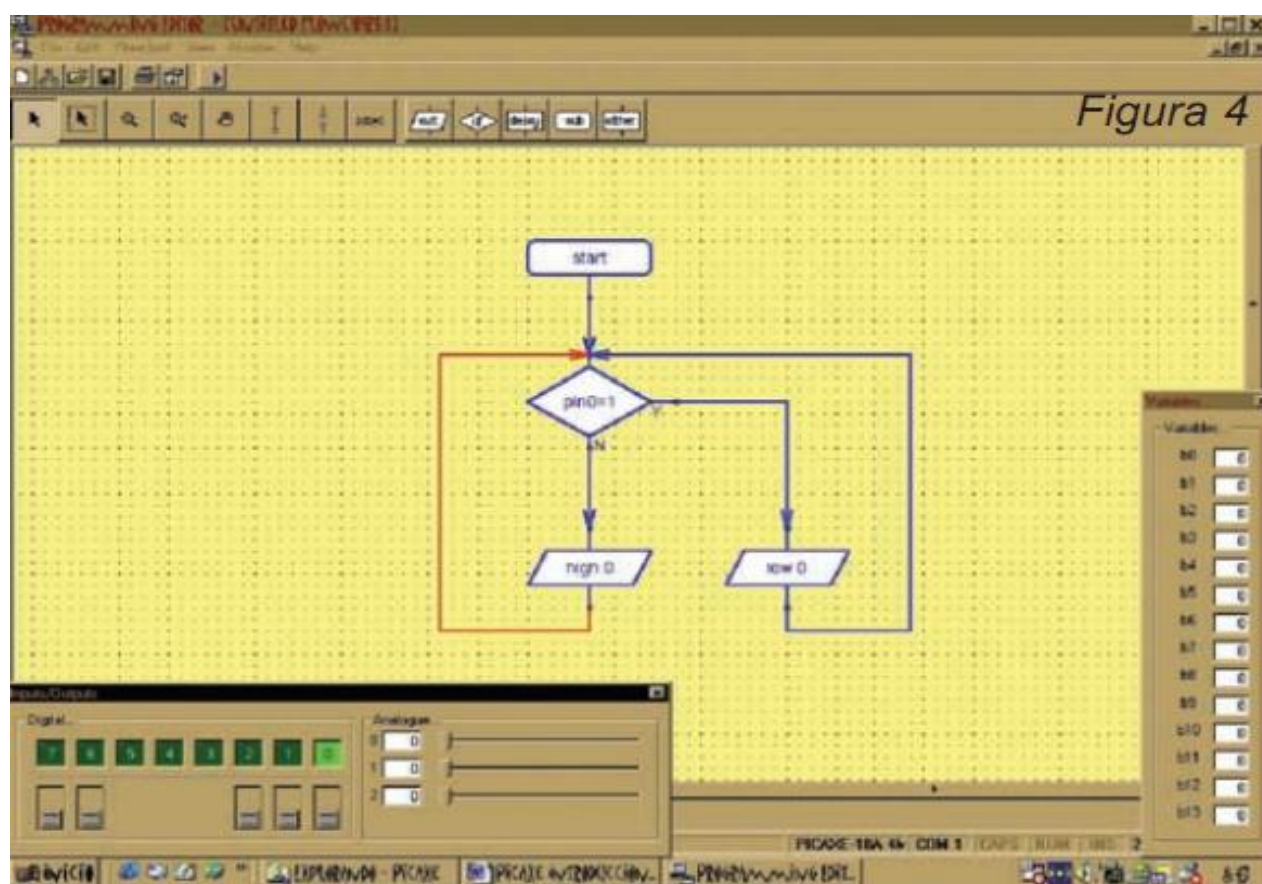


Un PIC es un microcontrolador que precisa un entorno de desarrollo (el MPLAB) para editar programas, simularlos, convertir el programa en archivo hexadecimal y realizar la simulación que verifique que está todo bien. Luego, se precisa un cargador para “descargar” el programa en la memoria del PIC, y por último se debe quitar el PIC del cargador y colocarlo en el circuito donde va a funcionar.

Un PICAXE “no precisa nada de todo eso...” es un PIC al que se le ha grabado un programa interno (firmware) para que pueda ser “cargado” en la mencionada misma placa donde va a funcionar por medio de un software gratuito y en el que los programas pueden escribirse en BASIC o en diagrama de flujo.

Tenga en cuenta que no es preciso que compre programa alguno para empezar a trabajar, dado que lo puede bajar gratis de Internet, además, Ud. puede armar el cable de conexión a la PC y la placa de circuito impreso del dispositivo que desee.

Hace apenas un tiempo, se ha lanzado al mercado el sistema de desarrollo para programar microcontroladores PIC llamado PICAXE, que de por sí, quien ha utilizado estos microcontroladores, puede constatar lo sencillo que resulta su programación, el sistema de desarrollo PICAXE hace las cosas toda vía más sencillas para el programador ya que cuenta con dos opciones de diseñar una aplicación: una por medio de diagramas de flujo y otra por medio de “BASIC”, y aunque esto no es ninguna novedad, (ya que estas herramientas existían con anterioridad), lo ventajoso del PICAXE radica en el hecho de que se trata de un microcontrolador PIC que, en un segmento de memoria ROM interna le ha sido grabado desde su fabricación, un firmware a manera de BIOS que simplifica la forma de programarlo.



Ya se ha mencionado que el sistema PICAXE no requiere de programador o borrador, ya que utiliza únicamente tres alambres conectados al puerto serie de una computadora, tal como se describe en la siguiente figura 9.

Una vez que han sido identificadas las terminales a utilizar en el conector del puerto serie de la PC, ahora lo que sigue es preparar la conexión hacia el PIC tomando en cuenta las terminales, tal como se aprecia en la figura 10.

Como se puede observar en la figura 10, se puede emplear (es recomendable) un plug de los utilizados para conectar los audífonos a la salida de audio de un walkman o discman, y tener un cable con un conector DB9 en un extremo y un plug de audio en el otro, tal como se ilustra en la figura 11.

