

El lenguaje gráfico FlowCode y el laboratorio USB-PIC'School. Una revolución en la formación y el diseño de aplicaciones con μ Cs PIC (1ª parte)

Por Mikel Etxebarria Isuskiza

MSE MICROSYSTEMS
ENGINEERING
www.microcontroladores.com

Mikel Etxebarria Isuskiza es Dtor. Técnico de Ingeniería de Microsistemas Programados S.L. (MSE)

En el mes de Febrero del 2010, en esta misma revista, publicamos un artículo dedicado al laboratorio USB-PIC'School. En él se explicaba sus características, arquitectura y posibilidades. Han transcurrido dos años desde que se comenzó la comercialización de este equipo. Durante este tiempo han sido numerosos los centros de estudios, empresas, alumnos, profesionales y aficionados que se han acercado al mundo de los microcontroladores PIC de la mano de esta potente, flexible y económica herramienta, tanto a nivel nacional como internacional.

También han sido numerosas las presentaciones, seminarios y cursos de formación sobre PIC's que, desde MSE, hemos realizado utilizando este laboratorio como plataforma de prácticas en todas las demostraciones, charlas y clases impartidas.

Durante este mismo tiempo también nos hemos encontrado en el mercado con múltiples herramientas, tanto hardware como software, para el desarrollo de aplicaciones basadas en PIC. Una de ellas, a nuestro juicio la más relevante, es el lenguaje gráfico FlowCode para la programación de PIC's, y distribuido por Ingeniería de Microsistemas Programados (MSE). De este lenguaje, y de su relación con el laboratorio USB-PIC'School, hablaremos en el presente artículo.



Figura 1. El laboratorio USB-PIC'School en su maletín de transporte y almacenamiento.

El laboratorio USB-PIC'School

Se muestra en la Figura 1. De entre sus muchas características vamos destacar las siguientes:

- Admite dispositivos PIC encapsulados en 8, 18, 28 y 40 patillas de las conocidas familias PIC12, PIC16 y PIC18. De serie incorpora el potente controlador PIC16F886.
- Integra un buen número de periféricos. Ello no permite estudiar y explotar al máximo los recursos que integra el controlador, así como rea-

lizar gran cantidad de aplicaciones y proyectos.

- Se conecta con el PC mediante un puerto USB, e incluye un sistema para la depuración y grabación de nuestros programas y aplicaciones.
- Compatible a nivel de hardware con el sistema PICKit 2 de Microchip. Esto nos permite utilizar el entorno de desarrollo MPLAB-IDE. Desde el MPLAB-IDE editamos, ensamblamos, simulamos, depuramos, ejecutamos y grabamos sobre el PIC, todos nuestros programas.
- Incluye un CDROM con el manual en castellano, tutorial y una colección de más de 70 ejemplos de programación escritos en ensamblador y en C de alto nivel.
- Se presenta totalmente montado y comprobado, listo para funcionar, sobre un maletín que facilita su transporte y almacenamiento. Únicamente precisa de un alimentador externo de 12VDC.

El usuario conecta mediante cableado sin soldadura, los diversos periféricos disponibles con las líneas de E/S del PIC, según precise. También dispone de un módulo board para el montaje sin soldadura, de cualquier tipo de circuito eléctrico auxiliar que la aplicación pudiera requerir.

Mediante un sencillo conmutador el usuario elige uno de los tres modos de trabajo disponibles en el laboratorio USB-PIC'School:

1) Modo RUN. Sirve para comprobar el funcionamiento real de una aplicación, cuando el programa ya está depurado y grabado sobre la memoria del PIC, al que también le habremos conectado los periféricos con sus líneas de E/S.

2) Modo USB. En este modo el laboratorio conecta su sistema de depuración/grabación con el puerto USB del PC. Se emplea durante la fase de depuración de la aplicación. El usuario puede ejecutar su programa paso a paso, insertar puntos de parada, visualizar y/o modificar los registros internos, variables del programa, etc. En resumidas cuentas, se puede seguir el curso de la ejecución del programa y localizar y corregir los fallos en el mismo. Este modo también permite grabar la memoria del PIC con el programa recién depurado y puesto a punto. Todo ello se realiza mediante el entorno MPLAB-IDE original de Microchip y gratuito.

3) Modo ICSP. En este modo el laboratorio USB-PIC'School se puede conectar con herramientas de depuración/grabación externas proporcionadas por otros fabricantes: ICD2, ICD3, PICKit 3, REAL ICE, etc.

El laboratorio USB-PIC'School incluye un elevado conjunto de periféricos comunes en la mayor parte de las aplicaciones comerciales. Un detalle importante es que dichos periféricos no están conectados a las líneas de E/S del PIC de forma fija y predeterminada. Se pueden unir a las líneas de E/S que se deseen y cuando sea necesario, mediante un cableado rápido sin soldadura. Teniendo en cuenta que una misma línea de E/S de un PIC actual puede tener múltiples configuraciones y aplicaciones, esta posibilidad la consideramos imprescindible para las labores formativas y de diseño.

Se enumeran los diferentes periféricos disponibles:

1. Entradas digitales (8)
2. Salidas digitales (8)
3. Salidas digitales de alta corriente (4)
4. Displays de 7 segmentos (2)
5. Entradas analógicas (2)
6. Generador lógico de diferentes frecuencias.
7. Teclado matricial de membrana de 4x4.
8. Interface RS232.
9. Pantalla LCD de 2x16 con fondo azul
10. Reloj calendario en tiempo real alimentado por pila y controlado mediante interface I2C.
11. Sensor de temperatura controlado mediante interface 1-Wire

El laboratorio USB-PIC'School se comercializa también en la versión "De Luxe" como se muestra en la figura 2. Esta versión incluye la fuente de alimentación y una colección de varios controladores PIC de las diferentes familias y con diferentes patillaje. Se presenta sobre un maletín de aluminio robusto y elegante.



Por último indicar que desde Junio del 2011, en todas las librerías especializadas está disponible la última obra de Mikel Etxebarria, autor también de este artículo. Se trata del título "Micro controladores PIC: Teoría y Prácticas". Editado por Creaciones Copyright se muestra en la figura 3.



En el aspecto teórico de la obra se muestra la arquitectura y programación de los mismos y se desgana todos y cada uno de los recursos in-

ternos que integran, y que son comunes en la mayor parte de dispositivos de la gran familia PIC.

En el aspecto práctico se hace una extensa explicación del entorno de trabajo MPLAB-IDE y sus diferentes herramientas para la edición, ensamblado, simulación, depuración y grabación de programas. También se explica la integración con el laboratorio USB-PIC'School y se complementa con una colección de unos 100 ejemplos prácticos.

Se dedica un anexo completo a la introducción del lenguaje gráfico FlowCode V4 para la programación de PIC's

El lenguaje gráfico FlowCode

Desarrollado por Matrix Multimedia Limited (www.matrixmultimedia.com). Se trata de un lenguaje gráfico, orientado a iconos, con los que el programador diseña su programa de aplicación, describiendo el organigrama o diagrama de flujo del mismo. Aunque aconsejable, no se hace necesario tener grandes conocimientos de electrónica ni de microcontroladores, por lo que es un lenguaje destinado a prácticamente todo tipo de usuarios: aficionados, estudiantes de ESO, Formación Profesional tanto en grado medio como en grado superior, ingenierías, profesionales, investigadores y desarrolladores de aplicaciones. La gran ventaja de FlowCode 4 es que permite, con unos mínimos conocimientos, crear complejos sistemas electrónicos en muy poco tiempo.

Ventajas

Son muchas las ventajas que ofrece el empleo de un lenguaje como el FlowCode 4:

- Ahorro de tiempo y dinero. FlowCode 4 facilita el diseño rápido de sistemas y aplicaciones basadas en microcontroladores (en nuestro caso los PIC's).

- Sencillo interface de usuario. Basta simplemente con coger iconos y arrastrarlos sobre el área de trabajo para confeccionar un diagrama de flujo que responda a las necesidades de la aplicación. No es necesario escribir un programa de la forma tradicional, instrucción a instrucción.

- Rápido y flexible. FlowCode 4 integra un gran número de subrutinas de alto nivel para la gestión de diversos componentes, lo que facilita un rápido desarrollo de cualquier aplicación. El método de programación de FlowCode 4 permite el uso de prácticamente todos los recursos que integra el microcontrolador.

- Resultado libre de errores. Cuando se diseña un programa, su funcionamiento se puede simular totalmente sobre el PC. El ejecutable obtenido se puede grabar directamente sobre la memoria de programa del microcontrolador.

- Arquitectura abierta. FlowCode 4, al igual que la mayor parte de los lenguajes gráficos y/o de alto nivel, ofrece una puerta abierta para que el usuario pueda insertar sus propias funciones o subrutinas escritas en lenguaje ensamblador o en C.

Características

Las más relevantes son:

- Soporta diferentes microcontroladores. Según la versión elegida, FlowCode 4 puede trabajar con dispositivos PIC, AVR, Atmel y ARM. El presente artículo se refiere a la versión para controladores PIC's

- Soporta diferentes sistemas de comunicación. FlowCode 4 integra las funciones necesarias para implementar comunicaciones según diferentes estándares o protocolos: I2C, SPI, RS232, Bluetooth, Zigbee, IrDA, CAN, LIN, TCP/IP, Webserver, USB y RFID.

- Soporta Componentes estándar. Efectivamente, FlowCode 4 integra las funciones necesarias para el control de los componentes más típicos usados en cualquier aplicación: LEDs, interruptores, teclados, pantallas, LCD de texto y gráficas, displays de 7 segmentos, memorias EEPROM, dispositivos de comunicación, etc.

- Panel de montaje. Donde colocamos y conectamos los componentes eléctricos que forman parte del

Figura 2. La versión De Luxe del USB-PIC'School

Figura 3. Libro teórico-práctico

proyecto y que serán controlados desde nuestro programa. Permite la simulación del funcionamiento de los mismos.

- Soporta diferentes herramientas. Una vez compilado y ensamblado el diagrama de flujo, se obtiene el programa ejecutable *.HEX. Este fichero puede ser utilizado por la mayor parte de equipos grabadores disponibles en el mercado, para proceder a grabar la memoria de programa del controlador. También soporta herramientas propias diseñadas y comercializadas por Matrix Multimedia y de otros fabricantes como el PICKit 2 de Microchip o, lo que es lo mismo, ¡¡ Nuestro laboratorio USB-PIC'Shool !!. En este caso basta un simple clic para compilar el diagrama de flujo y volcarlo directamente sobre nuestro PIC.

Versiones de FlowCode 4

FlowCode 4 se puede adquirir en diferentes versiones, independientemente del modelo de microcontrolador que vaya a ser usado:

- La versión libre. Se puede descargar de forma gratuita desde la web del fabricante (www.matrixmultimedia.com). Es ideal para poder evaluar el producto sin coste ni compromiso alguno.
- La versión de estudiante o "Home". Es bastante económica y está especialmente dirigida a estudiantes y aficionados. Es una buena alternativa para todos aquellos usuarios que quieran comenzar a utilizar esta técnica de programación y, sobre todo, para aquellos que desean familiarizarse con el hábito de programar.
- La versión "Pro" profesional. Es la versión completa de FlowCode 4, sin ningún tipo de límite ni restricciones. Admite la mayor parte de dispositivos PIC así como todos los componentes e iconos. No tiene límite en el código generado.

La siguiente tabla resume las características de cada una de estas tres versiones.

CARACTERÍSTICAS	VERSION LIBRE	VERSION HOME	VERSION PRO
Iconos ilimitados	NO	SI	SI
Límite del código generado	2K	4K	NINGUNO
Todos los dispositivos	NO	SI	SI
Todos los componentes	NO	NO	SI
Depuración en circuito (requiere hardware adicional)	NO	NO	SI
Licencia para uso comercial	NO	NO	SI
Derechos sobre el código	NO	NO	SI

Trabajar con FlowCode 4

La mecánica de trabajo con FlowCode 4 es muy similar a la empleada con cualquier otro lenguaje:

- Diseño del programa fuente. Simplemente, cogiendo y arrastrando los diferentes iconos de comandos disponibles, vamos dibujando el organigrama o diagrama de flujo de nuestra aplicación. Haciendo clic sobre esos iconos de comandos podemos ajustar los diferentes parámetros y propiedades asociados a ellos. De este modo, el diseño de programas complejos se realiza rápidamente.
- Diseño del sistema. FlowCode 4 dispone de una gran variedad de periféricos típicos como leds, interruptores, pantallas LCD, etc. En el panel de montaje podemos colocar los que necesitemos en nuestra aplicación y asignarles sus correspondientes propiedades y conexiones. Es decir, podemos realizar el montaje del circuito eléctrico.
- Simulación del programa. FlowCode 4 dispone de herramientas para la simulación y depuración de nuestro programa. Podemos ejecutar el programa paso a paso, icono a icono, analizar los resultados obtenidos sobre los componentes del sistema, editar y/o visualizar las variables del programa o ejecutar la simulación en tiempo real. Podemos comprobar el correcto funcionamiento, interactuando directamente con los componentes que hemos colocado en el panel de montaje.
- Descargar y grabar el microcontrolador. Con un único botón podemos compilar, ensamblar y grabar el programa ejecutable obtenido, sobre la memoria física del microcontrolador. FlowCode 4 se puede conectar directamente con el laboratorio USB-PIC'School (compatible con PICKit 2), cuyo dispositivo PIC recibirá el programa para una ejecución real y definitiva.

Instalación de FlowCode 4

Excepto la versión libre, que se descarga directamente desde la web del fabricante, las versiones Home y Pro se presentan en un CDROM. Basta con ejecutar el programa de instalación y seguir las instrucciones que van apareciendo.

Una vez instalado el programa, se permite su uso durante 30 días como máximo antes de ser activado. De no hacerlo así dejará de funcionar y el usuario deberá contactar con Matrix Multimedia para adquirir un nuevo código. El proceso de activación es sencillo:

- 1) Durante la instalación de FlowCode, se pide introducir el código de licencia que acompaña al CDROM. Se le denomina "Código CD".
- 2) Al ejecutar FlowCode hacemos click en el botón "Actívate licencia". Esto nos abre la página web de activación del fabricante, donde se nos pide el nombre de usuario y una dirección válida de correo electrónico.
- 3) Al cabo de un tiempo se recibe un email de confirmación.
- 4) Transcurridos unos días como máximo, se recibe un nuevo email con un código permanente de licencia al que se le denomina "Código de registro". Este código se introduce a través del menú Help → Register.

Si, por algún motivo, el proceso de instalación fallara, se debe contactar con support@matrixmultimedia.com e indicar el mensaje de error que nos indique.

Ejecución de FlowCode 4

Para ejecutar el FlowCode basta con hacer clic en su correspondiente acceso directo del escritorio de nuestro PC. Transcurrido un cierto tiempo, aparece el área de trabajo tal y como se muestra en la figura 4.

El área se divide en diferentes secciones. Cada una de ellas se puede visualizar o no en la pantalla, seleccionándola mediante la opción "Ver" del menú principal.

- Menú principal. Aparece en la línea superior de la pantalla. Contiene las diferentes opciones de trabajo del programa FlowCode. Algunas de estas opciones, como "Archivo", "Editar" y "Ver" son comunes a otras aplicacio-

nes Windows. El resto son opciones propias del programa y algunas de ellas las comentaremos en este artículo.

- Caja de Herramientas. Se encuentra justo debajo de las opciones del menú principal. Contiene, en forma de botones, algunas de las opciones del menú más utilizadas: Nuevo, Abrir, Salvar, Cortar, Pegar, Imprimir, etc. También hay botones para la ejecución de nuestro programa de aplicación: Ejecutar, Entrar, Saltar, etc.
- Herramientas de comandos. Se visualizan en vertical, a la izquierda de la pantalla. Contiene los diferentes iconos con los cuales vamos a confeccionar los organigramas o diagramas de flujo correspondientes a nuestro programa de aplicación. Estos comandos se arrastran y colocan dentro del área de trabajo.
- Área de trabajo. Es el área en donde se van colocando todos los iconos o comandos que forman el organigrama, se conectan entre sí y se establecen sus parámetros y propiedades.
- Herramientas de componentes. Aparece arriba, justo debajo de la caja de herramientas. Cada uno de esos botones nos permite seleccionar de entre un buen número de dispositivos periféricos que pueden ser utilizados por nuestra aplicación: Dispositivos más comunes, Dispositivos de entrada, de salida, de comunicaciones por radio frecuencia, etc. El componente elegido se coloca sobre el panel de montaje desde donde se definen sus propiedades y conexiones.
- Panel de montaje. Es el área donde se irán colando los componentes y periféricos que se emplean en la aplicación.
- Propiedades del panel. En esta ventana se editan algunas de las propiedades de los objetos que se encuentran en el panel de montaje.
- Barra de estado. Se visualiza a la izquierda, en la parte inferior de la pantalla. Aparecen pequeños mensajes que informan del estado del sistema.

Configuración del entorno de trabajo

Antes de empezar con un nuevo proyecto o aplicación, conviene configurar una serie de parámetros para establecer las condiciones y el entorno de trabajo del FlowCode 4.

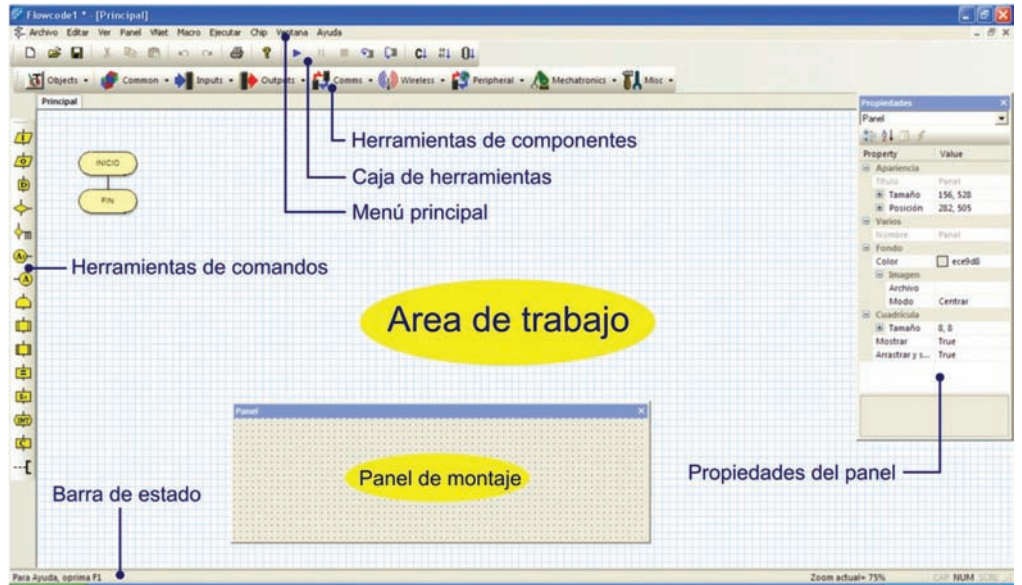


Figura 4. La pantalla de trabajo de FlowCode 4

Opciones del Compilador

Se establece mediante "Chip" → "Opciones del compilador". Se abre una ventana como la mostrada en la figura 5. Los distintos campos de esta ventana permiten establecer diferentes parámetros, y las rutas que debe utilizar FlowCode para acceder al compilador, el enlazador, el programador, etc. Estos campos vienen establecidos por defecto y se aconseja no modificarlos. En cualquier caso, si lo hiciéramos, siempre disponemos del botón "Restaurar Valores Pre-

determinados" para dejarlos en su estado original.

Especial mención se merece el campo "Programador" resaltado en la figura. Con él se establece el grabador al que FlowCode va a dirigir el programa ejecutable, para grabarlo físicamente sobre la memoria del microcontrolador. En el caso que nos ocupa observamos que se ha seleccionado el PICkit-2. Este grabador, original de Microchip,... ¡es compatible con el USB-PIC'School!!, del que ya hemos hablado.

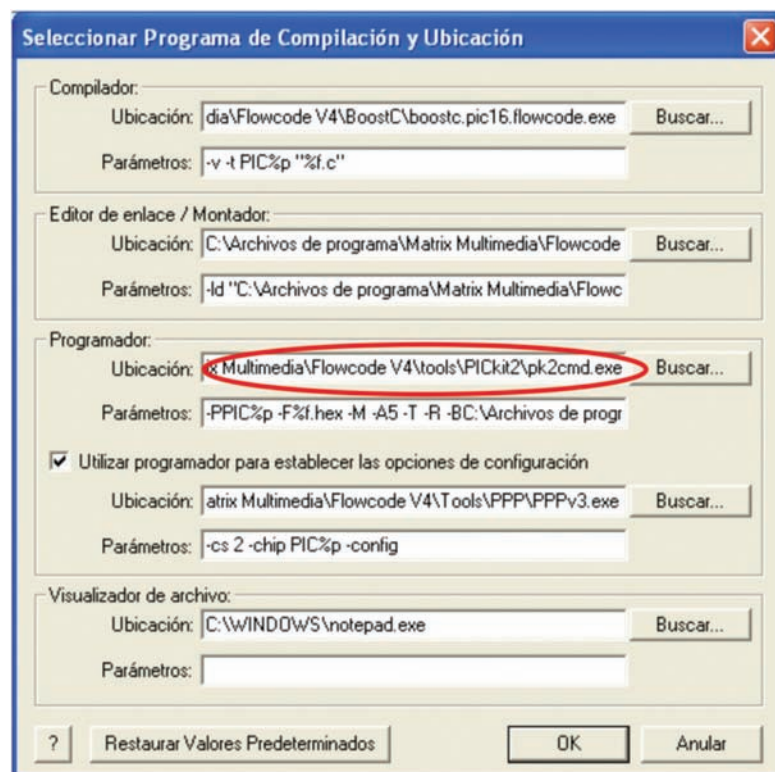
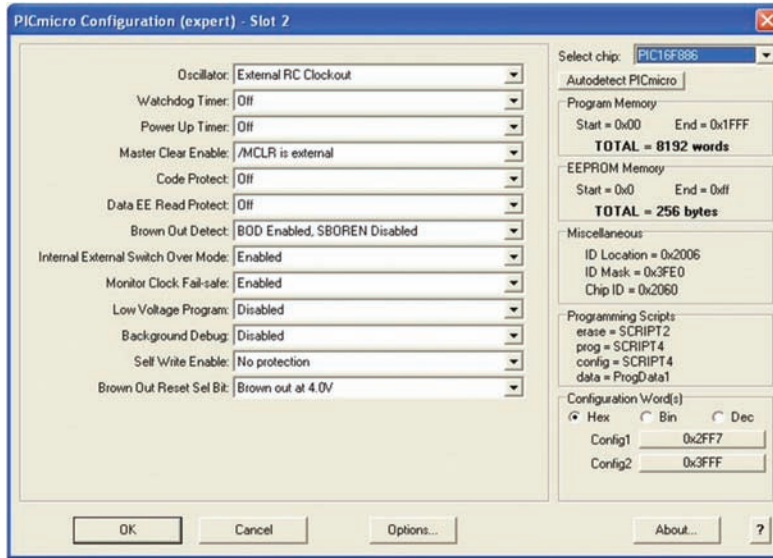


Figura 5. Configuración por defecto del entorno del FlowCode 4

Figura 6. Selección y configuración del microcontrolador



Selección del microcontrolador

Esta acción, fundamental a lo hora de comenzar cualquier proyecto o aplicación, se realiza mediante "Chip" → "Configurar" → "Switch To Expert".

En el caso que nos ocupa hemos seleccionado el PIC16F886 contenido en el laboratorio. Podemos comprobar que su palabra de configuración queda establecida de forma similar a como se hace en los ejemplos escritos en ensamblador y en C.

Opciones del proyecto

Para acabar. La configuración exige definir una serie de opciones propias del proyecto con el que vamos a trabajar. Estas se eligen mediante "Ver" → "Opciones de Proyecto" que nos abre una ventana como la mostrada en la figura 7.

Las opciones generales nos permiten seleccionar el modelo de microcontrolador que se va a emplear durante la simulación. Por defecto, debe ser el mismo que se eligió mediante "Chip" → "Configurar" → "Switch To Expert". De igual manera, podemos seleccionar la velocidad del reloj del sistema así como la velocidad que deseamos utilizar durante la simulación.

También hay opciones para la depuración en circuito o ICD en tiempo real. Estas opciones sólo son útiles si disponemos de la herramienta hardware que propone Matrix Multimedia como es su depurador "FlowKit".

Herramientas de comandos

Son los iconos o símbolos gráficos con los que vamos a confeccionar los organigramas o diagramas de flujo de nuestra aplicación. Es decir, los comandos se corresponden con tareas concretas con las que vamos a con-

feccionar los programas. Todas ellas se encuentran en la columna de la izquierda de la pantalla de trabajo.

Cuando cualquiera de los iconos de comandos se traslada al área de trabajo, queda automáticamente conectado con el resto de iconos existentes. Vamos dibujando así, paso a paso, el organigrama. Al hacer un doble clic con el botón izquierdo del ratón sobre cualquiera de los iconos existentes, se abrirá su correspondiente ventana de propiedades, en donde estableceremos los diferentes parámetros del comando en cuestión. Todas las ventanas de propiedades de todas las herramientas de comandos disponen de un botón ("?") que nos facilita la información de ayuda correspondiente (en castellano).

Herramienta Ingreso

Permite ejecutar tareas relacionadas con la adquisición de datos procedentes de los Puertos de entrada del controlador. La figura 8 muestra su icono así como la ventana de propiedades que se abre, cada vez que hacemos doble clic sobre él. En el campo "Mostrar Nombre" podemos añadir un comentario, por ejemplo "Leer entradas", que aclare la operación que vamos a realizar. Es similar a los comentarios que se ponen en los programas fuente escritos en ensamblador o en lenguaje C.

En el campo "Variable" indicamos en qué variable se va a guardar el valor leído desde el Puerto de entrada. Podemos pulsar el botón "Variables..." que nos abrirá una nueva ventana donde podremos crear, editar y borrar variables de 8 bits, de 16, de cadenas ASCII o strings y variables para almacenar números en coma flotante.

En el campo "Puerto" seleccionamos qué Puerto de entrada del microcontrolador vamos a leer. Según el modelo de PIC seleccionado, podemos leer desde el PUERTO A hasta el PUERTO X en función de los Puertos disponibles. En el ejemplo se lee el Puerto A

La lectura de un Puerto de entrada puede hacerse a nivel de un bit único o a nivel del puerto completo. En este caso podemos hacer además una lectura enmascarada. Es decir, podemos indicar qué bits de éste puerto son los que nos interesa leer. En el ejemplo se leen los bits 0,2 y 3 del Puerto A.

Figura 7. Las opciones de un proyecto basado en FlowCode 4

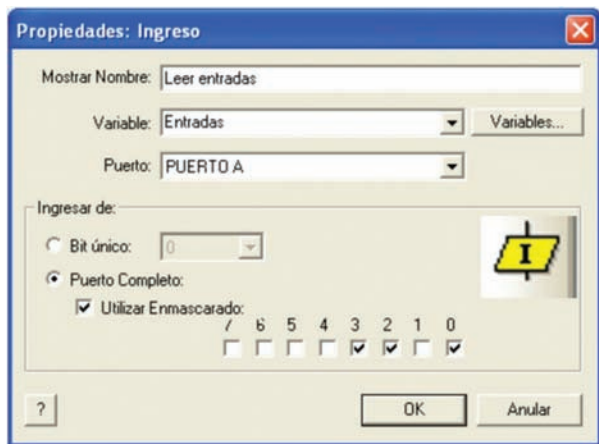
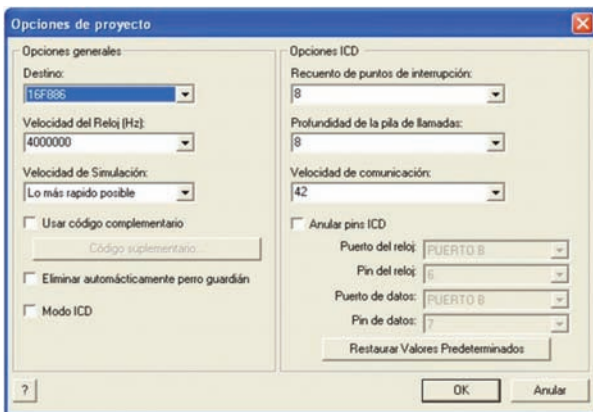


Figura 8. Herramienta de ingreso

Aparece una nueva ventana como la mostrada en la figura 6. En ella se ajustan parámetros propios del controlador que vayamos a utilizar.

Herramienta de Salida

Permite ejecutar operaciones relacionadas con los Puertos de salida del controlador. En la figura 9 se muestra el icono y la correspondiente ventana de propiedades. En "Mostrar Nombre" podemos añadir el comentario deseado. En el campo "Variable o Valor" introducimos directamente el valor de salida o bien el nombre de la variable que contiene el valor que deseamos sacar. En el ejemplo sacamos el valor de la variable "Entradas". En el campo "Puerto" seleccionamos el puerto de salida, Puerto B en el ejemplo.

La escritura sobre un Puerto de salida también se puede realizar a nivel de bit único o a nivel de Puerto completo. En este caso la escritura puede ser o no enmascarada. En el ejemplo se escribe únicamente sobre los bits 0, 2 y 3 del Puerto B.

Herramienta de Demora (temporización)

Permite realizar todo tipo de temporizaciones basadas en la frecuencia de trabajo del sistema que hemos seleccionado mediante las opciones del proyecto ("Ver" → "Opciones de Proyecto"). Su ventana de propiedades se muestra en la figura 10. Como en todas las herramientas, en el campo "Mostrar Nombre", podemos incluir un comentario. En el campo "Tiempo de Demora o Variable" introducimos el valor a temporizar. Este valor se puede indicar directamente (13 en el ejemplo), o bien indicando el nombre de la variable que lo contiene. En este caso pulsamos el botón "Variables..." para seleccionarla de entre todas las que estén definidas y disponibles.

Esta ventana de propiedades también nos permite seleccionar las unidades en que se representa el tiempo: Microsegundos, Milisegundos o Segundos. En el ejemplo se realiza una temporización de 13 mS.

Herramienta de Decisión

Se trata de la clásica toma de decisiones. Se realiza una determinada evaluación y, en función del resultado, el flujo del programa va por un camino o por otro. En la figura 11 se muestra la correspondiente ventana de propiedades. En el campo "En Caso de Que" se establece la condición. En el ejemplo de la figura se evalúa si el

valor de la variable "Entradas" es igual a 1. Esto es equivalente a evaluar si la línea RA0 de la puerta A está a nivel lógico "1" o no. En caso afirmativo, el curso de la ejecución se desvía por la derecha. En caso contrario la ejecución continúa su curso. Podemos activar (o no) el campo "Intercambiar Si y No". En este caso el desvío del curso de la ejecución se produce si NO se da la condición establecida. Las condiciones pueden estar formadas por números y/o variables y uno de los siguientes operadores:

()	Paréntesis.
=, <>	Igual a, diferente a.
+, -, *, /, MOD	Adición, sustracción, multiplicación, división y módulo.
<, <=, >, >=	Menor que, menor que o igual a, mayor que, mayor que o igual a.
>>, <<	Desplazamiento a la derecha, desplazamiento a la izquierda.
NOT, AND, OR, XOR	Funciones lógicas: NOT, AND, OR u OR exclusiva

Herramienta de Expresión o decisiones múltiples

La figura 12 muestra la correspondiente ventana de propiedades. Se trata de una herramienta similar a la anterior herramienta de decisión. La diferencia es que aquella podía generar dos posibles resultados (SI o NO) en función de la evaluación. En este caso se pueden evaluar hasta diez posibles condiciones. Resultan especialmente útiles a la hora de personalizar el programa para que reaccione de formas diferentes basándose en una variable de entrada.

En el campo "Expresión" se indica el nombre de la variable cuyo contenido se desea evaluar, "Entradas" en el ejemplo. Los campos "Caso", cuando se activan, permiten determinar con qué valores hay que comparar el contenido de la variable "Entradas": 1, 4, 2 o 7. Cuando coincide uno de ellos, el flujo de ejecución del programa toma un camino u otro. Si no coincide con ninguno, la ejecución sigue su curso. Esto se aprecia claramente en la figura 13.

Se evalúa el valor de la variable "Entradas". Si es igual a 1, se temporiza 1 segundo; si es 4 se temporiza 4 segundos; si es 2 se temporiza 2 segundos y si es 7 se temporiza 7 segundos. Si no



Figura 9. Herramienta de salida



Figura 10. Herramienta de temporización

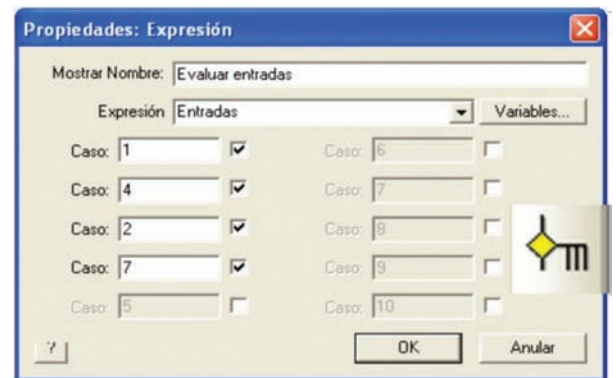
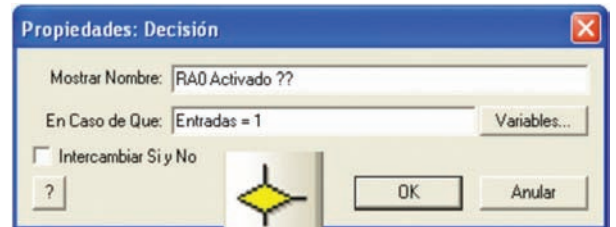


Figura 11. Herramienta de decisión

Figura 12. Herramienta de expresión múltiple

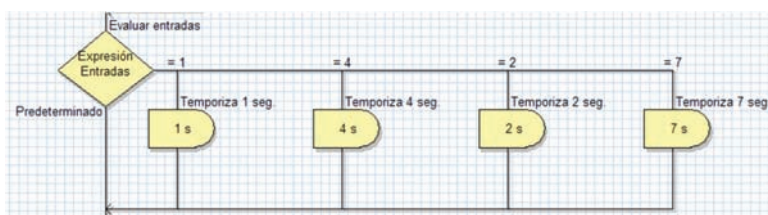


Figura 13. Evaluando cuatro posibilidades

Figura 14. Herramienta Punto de conexión

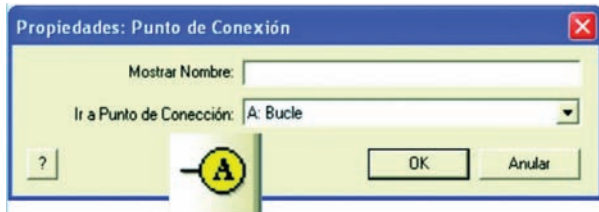
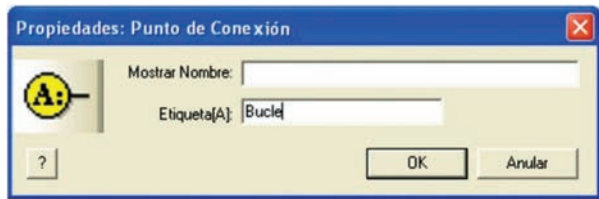


Figura 15. Herramienta Ir a Punto de conexión

Figura 16. Un sencillo bucle sin fin.

Figura 17. Herramienta de Bucles

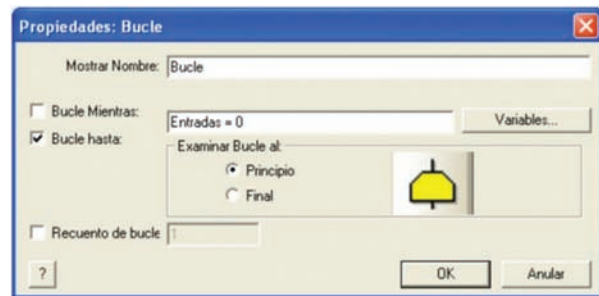
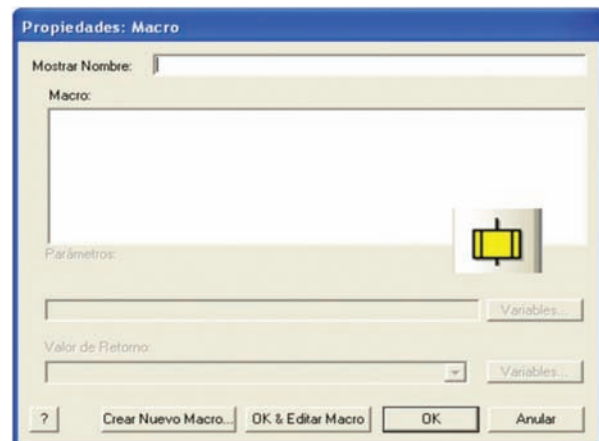


Figura 18. Ejecución condicional de un bucle

Figura 19. Herramienta Macro



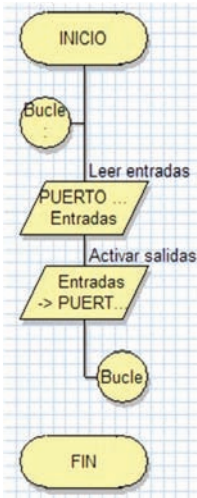
coincide con ninguno de esos cuatro valores, el programa sigue su curso sin hacer temporización alguna.

Herramienta Ir a Punto de conexión

Esta herramienta se emplea junto con la anterior. Permite ir o saltar al punto de conexión de destino al que debe dirigirse el flujo del programa. La ventana de propiedades se muestra en la figura 15.

En el campo "Ir a Punto de Conexión" asignamos un nombre o etiqueta del punto de destino. En el ejemplo hemos asignado la etiqueta "Bucle".

La figura 16 muestra un sencillo ejemplo en el que se emplean ambas herramientas. Comenzamos estableciendo un punto de conexión llamado "Bucle". A continuación se lee el contenido de Puerto A de entrada y se deposita en la variable "Entradas". Seguidamente, el contenido de la variable "Entradas" (el valor del Puerto A) se deposita sobre el Puerto B de salidas.



El programa finaliza con la herramienta "Ir a Punto de Conexión" llamado "Bucle", lo que origina que el flujo se desvíe al punto de conexión superior. Como consecuencia de todo ello, tenemos un programa que lee el valor de las entradas del Puerto A y lo deposita sobre el Puerto B de salida. Esto se repite de forma indefinida en lo que se conoce como un "bucle sin fin". Cualquier cambio en las entradas se verá inmediatamente reflejado en las salidas.

Herramienta Bucle

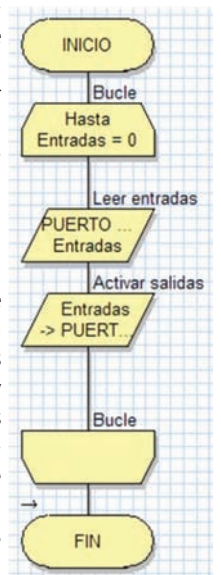
Esta herramienta de Bucle se emplea para repetir una tarea, bien de forma incondicional o indefinida, o bien hasta que se cumplan las condiciones indicadas. La figura 17 muestra la correspondiente ventana de propiedades. Esta herramienta siempre se muestra como dos iconos, uno de inicio y otro de final. Entre ambos, a modo de sándwich, se alojan los comandos de programa que forman el bucle.

Los comandos encerrados dentro de un bucle se pueden ejecutar mientras se produzca una condición ("Bucle Mientras") o hasta que se produzca esa condición ("Bucle Hasta"). En el ejemplo el bucle se ejecuta hasta que el valor de la variable "Entradas" sea igual a 0. En el momento en que se cumpla esta condición, el bucle finaliza y el flujo del programa reanuda su curso.

Las condiciones se pueden evaluar y comprobar si se cumplen, al "Principio" del bucle o al "Final" del mismo. Los comandos o instrucciones que componen un bucle también se pueden ejecutar un número determinado de veces. Para ello, activamos e indicamos en el campo "Recuento de bucle", el número de veces deseado (entre 1 y 255).

Indicar que también se puede crear un bucle en el que sus comandos se ejecuten de forma indefinida e incondicional.

Basta con activar "Bucle Mientras" e introducir el valor 1 en el campo de condiciones. En el ejemplo de la figura 18 vemos un bucle que consiste en leer las entradas del Puerto A y reflejarlos en las salidas del Puerto B. El bucle se ejecuta hasta que las entradas valgan 0.



Herramienta Macro

Una macro no es ni más ni menos que una función o una subrutina. Está compuesta de cualquiera de los comandos o iconos disponibles y/o también de otras macros. Es una opción muy interesante que ahorra mucho tiempo de programación y, sobre todo, mucho código de programa. Una macro realiza una determinada tarea compuesta de varios comandos. Nuestro programa principal puede ejecutar esa tarea simplemente llamado a la macro tantas veces como sea necesario, sin tener que reescribir los comandos que la componen. La figura 19 muestra la ventana de propiedades del comando Macro.

En el campo "Mostrar Nombre" podemos escribir un comentario. En el campo "Macro" podemos ver una lista con todas las macros disponibles. Seleccionamos la que necesitemos en un momento dado.

Mediante el botón "Crear Nuevo Macro" accedemos a una ventana que nos permite definir una nueva Macro.

Con el botón "OK&Editar Macro" podemos editar una Macro. Es decir, describir los componentes que debe ejecutar para realizar una determinada tarea.

Crear Nueva Macro

Cuando se pulsa el botón "Crear Nueva Macro" se nos abre una ventana como la mostrada en la figura 20. En el campo "Nombre del Nuevo Macro" indicamos el nombre que le queremos asignar a esa Macro, por ejemplo, "Test".

En el campo "Descripción del Nuevo Macro" podemos poner un texto que describa la tarea que realiza la Macro.

En el campo "Parámetros" podemos definir datos o variables de entrada que nuestra Macro pueda necesitar.

En "Variables Locales" podemos crear y editar variables que sólo serán utilizadas por la propia Macro. No se usan en ningún otro si lugar del programa de aplicación.

En el campo "Tipo de Retorno" se indica si la Macro devuelve, tras su ejecución, un valor de 8 bits, de 16 bits, una cadena ASCII o un número en coma flotante. También se puede indicar que no se retorna ningún valor.

Edición de una Macro

Cuando se pulsa el botón "OK&Editar Macro", aparece una nueva hoja en blanco en el área de trabajo. En ella vamos colocando los comandos o instrucciones que se deben ejecutar cada vez que se llame a la Macro. En la figura 21 se muestra un ejemplo de la Macro que hemos llamado "Test".

Podemos ver que, en el área de trabajo, aparece una nueva pestaña correspondiente a la hoja donde vamos a dibujar el diagrama de flujo con las funciones que integra la Macro "Test". Empieza evaluando el valor de la variable "Entradas" que se

corresponde con el valor cargado desde el Puerto A. Si el valor leído es 0, el Puerto B de salida saca 0b00001111, si es 1 se saca 0b11110000, si es 2 se saca 0b01010101 y, por último, si se lee 3 desde el Puerto A de entrada, se saca el valor 0b10101010 por el Puerto B de salida. Si el valor leído no es ninguno de esos cuatro, por el Puerto B se saca 0.

En la figura 22 se muestra cómo queda el programa principal. Consiste en un bucle sin fin que ejecuta los siguientes comandos:

1) Lee el Puerto A de entrada y almacena su contenido en la variable "Entradas"

2) Llama a la Macro "Test" donde se ejecutan las comprobaciones descritas anteriormente.

Salta a la vista que el empleo de las Macros, que no dejan de ser funciones o subrutinas, facilita enormemente la comprensión de un programa.

El comando Macro del menú principal

Este comando nos abre una serie de opciones relacionadas con el empleo de las Macros:

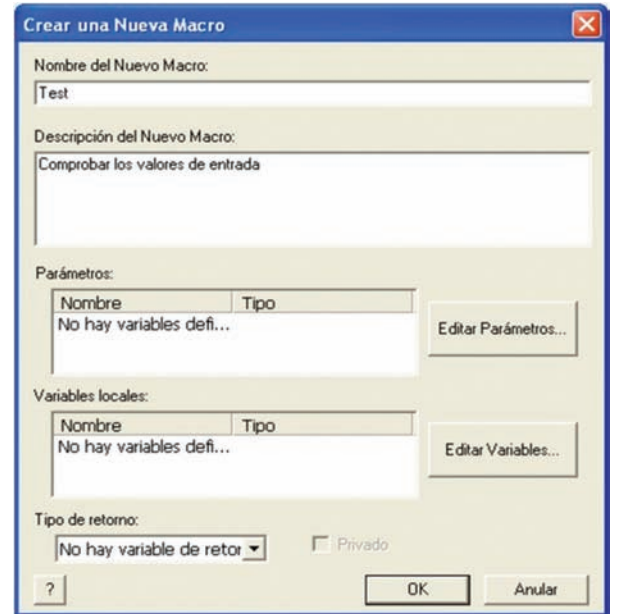


Figura 20. Definición de una Macro

- Nuevo. Nos permite definir una nueva Macro.
- Mostrar. Nos presenta un listado de todas las Macros que hubiera definidas. Haciendo clic sobre cualquiera de ellas nos mostrará su composición.
- Borrar. Visualiza la lista de Macros disponibles para que seleccionamos aquella que queremos borrar.
- Editar. Visualiza el contenido de una Macro con objeto de editarla añadiendo, modificando o quitando componentes de la misma.
- Descripción. Nos permite modificar el texto que describe la función de una Macro.
- Exportar. Con esta opción podemos guardar una Macro sobre un archivo en disco. Esto nos permite crear librerías de Macros que pueden ser

Figura 22. El programa principal del ejemplo

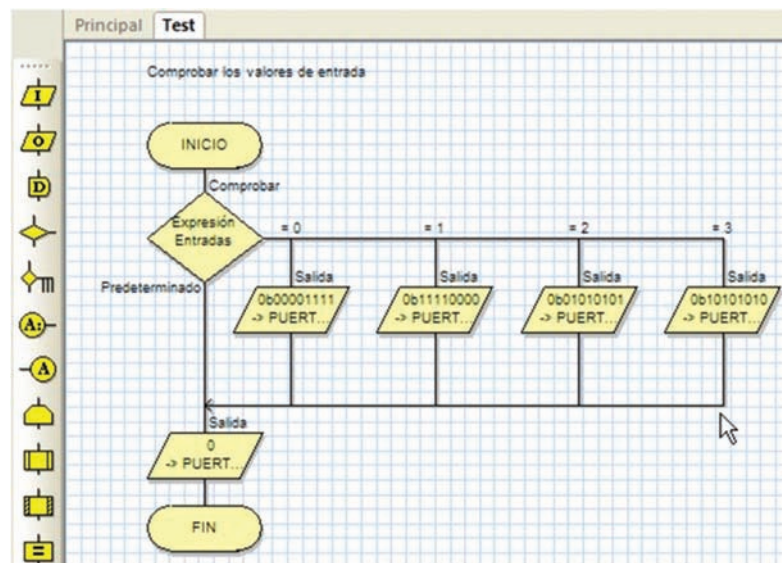
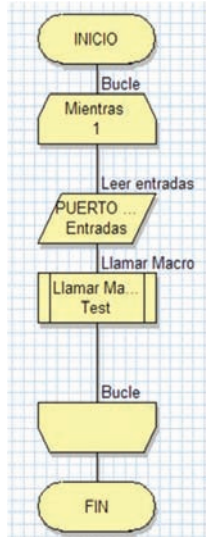


Figura 21. Contenido de la Macro "Test"

utilizadas en múltiples proyectos o aplicaciones.

- Importar. Esta opción permite leer y recuperar desde un archivo en disco las distintas Macros que hubiera almacenadas

colocan en el "Panel de Montaje", en donde se establecen sus propiedades y conexiones con el controlador. Cada componente tiene asociadas, según sus características, un número determinado de funciones o Macros para su control. Ya lo veremos más adelante.

Cuando empleamos la herramienta "Macro de Componentes" y hacemos doble clic sobre ella, aparece una ventana de propiedades como la mostrada en la figura 23. En ella disponemos de un listado con todos los componentes que hubiera en ese momento en el panel de montaje y accedemos a todas las funciones asociadas a los mismos. En la figura vemos que hay dos componentes: un interruptor SWITCH(0) y un led LED(0). Al seleccionar por ejemplo, el SWITCH(0), a la derecha, en la ventana "Macro", aparecen las funciones a él asociadas. Hay tres:

1) LeerEstado. Lee el estado de la línea de entrada a la que está conectado el SWITCH(0) y lo deposita en la variable de retorno que se indica en el campo "Valor de Retorno(BYTE)".

2) EsperarHastaAlto. Espera a que la línea de entrada a la que está conectado el SWITCH(0) se ponga a nivel "1". En este momento finaliza la Macro o función, que no retorna ningún valor.

3) EsperarHastaBajo. Espera a que la línea de entrada a la que está conectado el SWITCH(0) se ponga a nivel "0". En este momento finaliza la Macro o función, que no retorna ningún valor.

La figura 24 muestra la misma ventana de propiedades de la herramienta "Macro de Componentes". En esta ocasión se selecciona el componente LED(0) donde podemos ver las funciones o Macros asociadas a éste:

1) LEDEncendido. La línea de salida donde se encuentra conectado el LED(0) se activa.

2) LEDApagado. La línea de salida donde se encuentra conectado el LED(0) se desactiva.

Estas funciones no necesitan de ningún otro parámetro.

En la figura 25 se muestra un sencillo ejemplo que utiliza la herramienta "Macro de Componentes" para utilizar las funciones o Macros que permiten controlar, de forma muy sencilla, un interruptor, el SWIT-

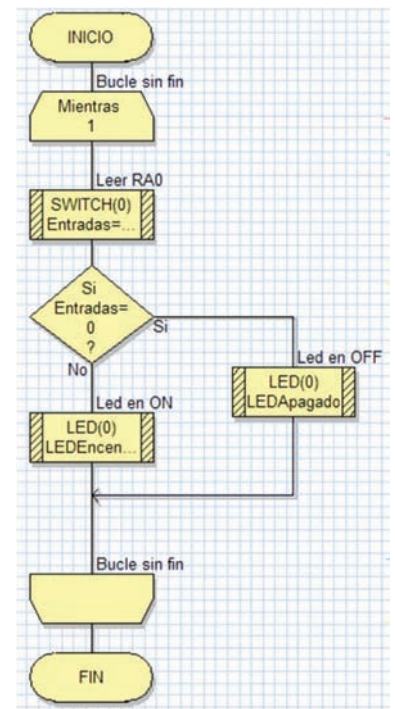
CH(0) y un led, el LED(0). Ambos componentes se suponen previamente dibujados y definidos en el panel de montaje.

La primera Macro de componentes lee el estado del SWITCH(0) conectado con la línea RA0, y lo deposita en la variable "Entradas". Este se evalúa mediante la herramienta de decisión.

Si RA0=0 se ejecuta la función "LEDApagado" contenida en la Macro de componentes LED(0). La línea de salida RBO se pone a nivel "0" y el led se apaga.

Si RA0=1 se ejecuta la función "LEDEncendido" contenida en la Macro de componentes LED(0). La línea de salida RBO se pone a nivel "1" y el led se enciende.

Se trata de un bucle sin fin que se ejecuta constantemente.



Herramienta de Cálculo

Otra herramienta esencial en FlowCode 4 es la llamada "Herramienta de Cálculo". Permite insertar operaciones lógico/aritméticas en nuestros programas. La ventana de propiedades se muestra en la figura 26.

En el campo "Mostrar Nombre" podemos escribir el comentario que deseemos. En el campo "Cálculos" describimos los cálculos a realizar.

Figura 23. Herramienta Macro de Componentes

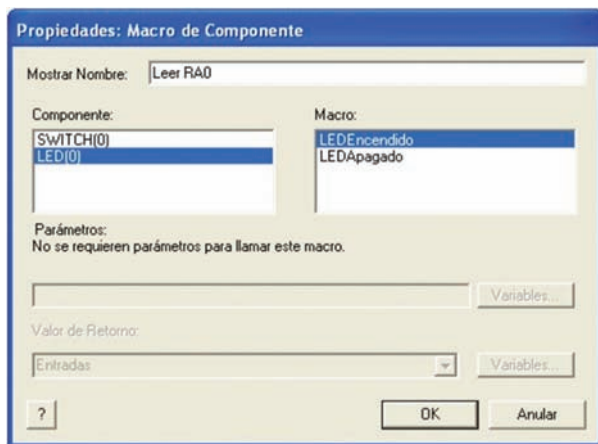
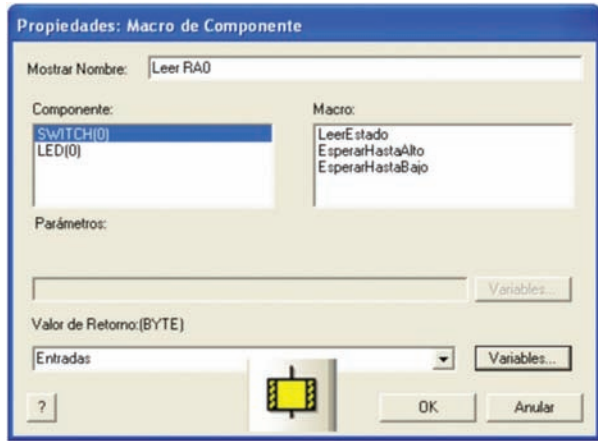


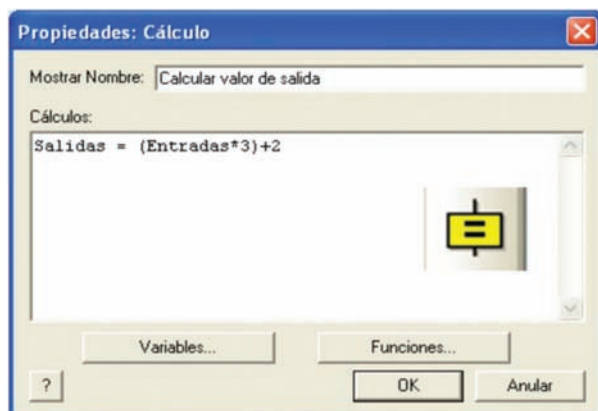
Figura 24. Herramienta Macro de Componentes

Herramienta Macro de Componentes

Figura 25. Ejemplo con Macros de Componentes

FlowCode 4 incluye una serie de componentes o periféricos con los que podemos construir, para su simulación, el circuito eléctrico de nuestra aplicación. Estos componentes se

Figura 26. La Herramienta de Cálculo



El botón "Variables" nos abre una ventana donde podemos crear, editar, borrar y seleccionar distintas variables. Con el botón "Funciones" accedemos a todas las funciones lógicas/aritméticas disponibles.

En el ejemplo, la variable "Salidas" se carga con el resultado de multiplicar por tres el valor de la variable "Entradas" y sumarle dos.

Herramienta Manipulación de Secuencia

La herramienta de Manipulación de Secuencia permite manipular y operar con cadenas de caracteres ASCII. Su ventana de propiedades se muestra en la figura 27.

En el campo "Funciones de" editamos las operaciones a realizar con una o varias cadenas de caracteres.

El botón "Variables" nos permite crear, borrar, editar o usar variables de secuencia o cadenas.

El botón "Funciones" nos presenta una lista con todas las funciones disponibles para el manejo de secuencias o cadenas de caracteres.

En el ejemplo de la figura, la variable "Cadena1" se carga con la cadena "Hola " y la variable "Cadena2" con la cadena "Mundo". La variable "Cadena3" se carga con el resultado de concatenar las dos cadenas anteriores, por lo que su contenido será "Hola Mundo".

Herramienta de Interrupción

FlowCode 4 también soporta la activación/desactivación de algunas de las interrupciones más relevantes del microcontrolador seleccionado. En la figura 28 se muestra la ventana de propiedades de esta herramienta.

En el campo "Interrumpir en" seleccionamos la fuente de interrupción, por ejemplo la interrupción externa INT. Aunque no están disponibles todas las que tiene el controlador, si están las más representativas. Con el botón "Propiedades" se selecciona el evento deseado para esa interrupción. Por ejemplo, en el caso de la interrupción externa INT, podemos seleccionar el flanco ascendente o descendente.

En el campo "Llamará al Macro"

se indica qué Macro o función se debe ejecutar cada vez que se produzca la interrupción. Con el botón "Crear Nuevo Macro" creamos una nueva función. Con el botón "OK & Editar Macro" editamos el contenido de la Macro. Es decir, dibujamos el organigrama de tareas que esa Macro debe ejecutar. El funcionamiento es muy similar al empleado en la herramienta de componente "Macro".

Herramienta Código C

En FlowCode sucede lo mismo que en la mayor parte de lenguajes de alto nivel. Es imposible prever y diseñar las herramientas y Macros o Funciones necesarias para aprovechar TODOS los recursos del microcontrolador. Puede haber casos en que el lenguaje como tal no pueda realizar ciertas tareas. Por ello, este tipo de lenguajes siempre tienen una puerta abierta que nos permita insertar nuestro propio código fuente en C o en ensamblador, para resolver aquellas tareas que FlowCode no pueda realizar directamente. Disponemos así de la herramienta Código C.

La ventana de propiedades de esta herramienta es muy sencilla y se muestra en la figura 29. Simplemente consiste en un editor de textos donde escribimos nuestro programa empleando instrucciones del lenguaje C o del lenguaje ensamblador.

Herramienta de Comentarios

Es la última herramienta de comandos que dispone FlowCode

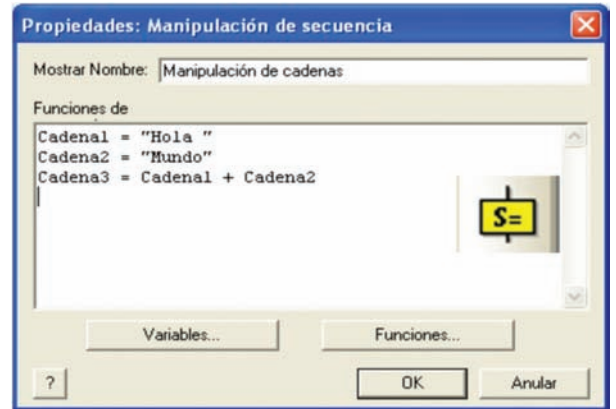


Figura 27. Herramienta de Manipulación de Secuencia

4. Su ventana de propiedades se muestra en la figura 30 y consiste simplemente en un área donde podemos insertar todo tipo de textos que describan el programa y facilite su comprensión y análisis.

Figura 28. Herramienta de interrupciones

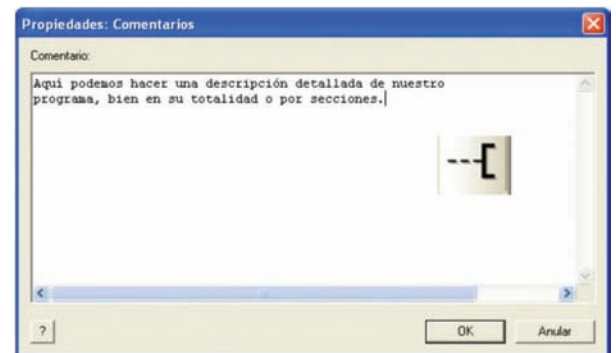
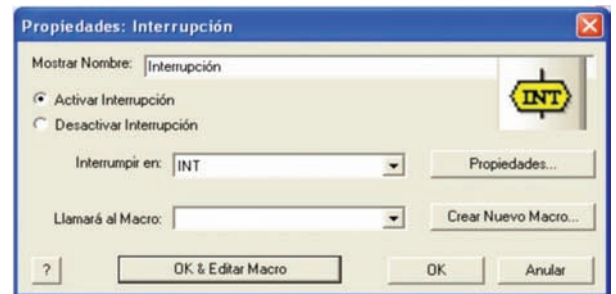


Figura 30. Herramienta de Comentarios

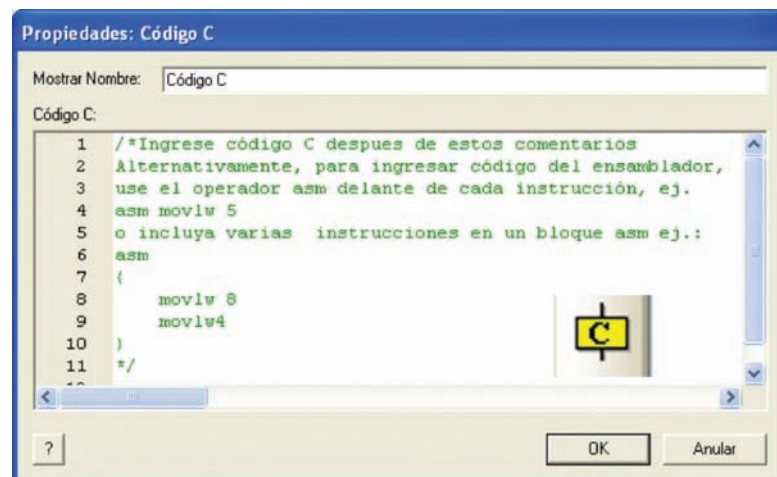


Figura 29. Herramienta para insertar código C o ensamblador