

### AN12-1 INTRODUCCIÓN

Efectivamente, tal y como reza el título del presente Anexo, vamos a ver otra forma de programar microcontroladores PIC empleando un nuevo lenguaje, el “FlowCode”, de Matrix Multimedia Limited ([www.matrixmultimedia.com](http://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.

#### AN12-1.1 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, como lo hemos venido haciendo en los ejemplos que se han propuesto en la obra.
- **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 ensamblador o en C.

#### AN12-1.2 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 Anexo se refiere a la versión para 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, sensores, displays de 7 segmentos, memorias EEPROM, dispositivos de comunicación, etc.

- **Panel de montaje.** Donde colocamos y conectamos los componentes que forman parte de nuestra aplicación 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 de nuestra aplicación, el programa ejecutable obtenido se puede descargar directamente sobre nuestro microcontrolador mediante herramientas hardware como pueden ser el PICKit 2 (**¡compatible con el laboratorio USB-PIC-School!**) o el PICKit 3 de Microchip.

### **AN12-1.3 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](http://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.

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

### **AN12-1.4 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. 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. Comprobamos 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.

### **AN12-1.5 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 clic en el botón "*Activate license*". 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 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](mailto:support@matrixmultimedia.com) e indicar el mensaje de error que nos indique.

### **AN12-2 EJECUCIÓN DE FlowCode 4**

Para ejecutar el FlowCode basta con hacer clic en su correspondiente acceso directo del escritorio de nuestro PC. Transcurrido cierto tiempo, nos aparece el área de trabajo tal y como se muestra en la figura AN12-1.

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 aplicaciones Windows. El resto son opciones propias del programa y algunas de ellas las comentaremos en este Anexo.
- **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 visualiza 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 desde donde se conectan entre sí y se establecen los diferentes parámetros y propiedades.
- **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, permite seleccionar de entre un buen número de dispositivos periféricos que pueden ser utilizados por una aplicación: dispositivos más comunes, dispositivos de entrada, de salida, de comunicaciones por radiofrecuencia, 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 colocando 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.

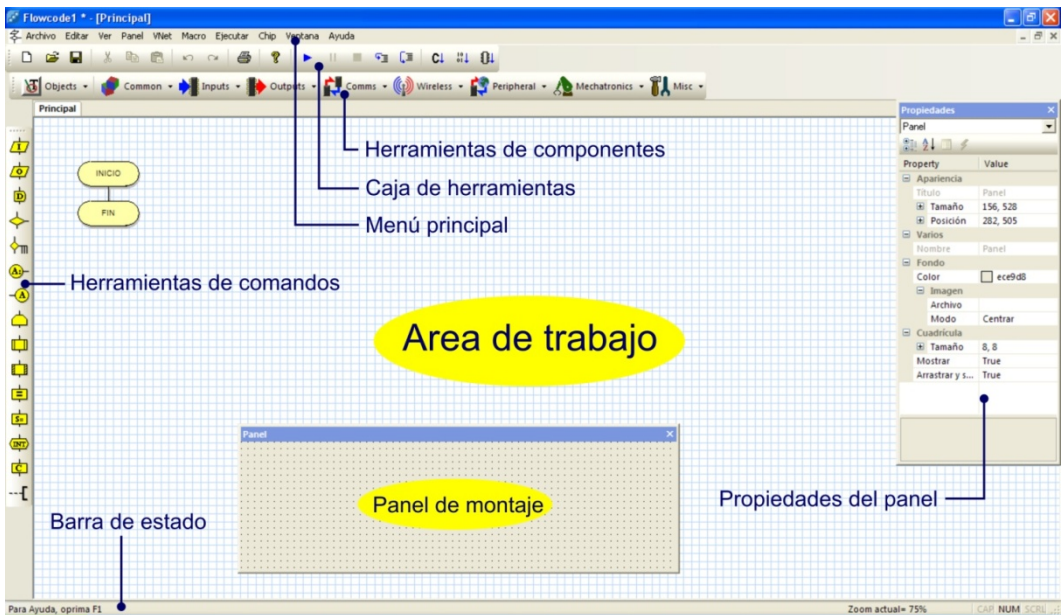


Figura AN12-1. La pantalla de trabajo de FlowCode 4

### AN12-3 CONFIGURACIÓN

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

### AN12-3.1 Opciones del Compilador

Se establece mediante “Chip” → “Opciones del compilador”. Se abre una ventana como la mostrada en la figura A12-2. Los diferentes 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 Predeterminados” 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!**, que hemos utilizado a lo largo de obra.

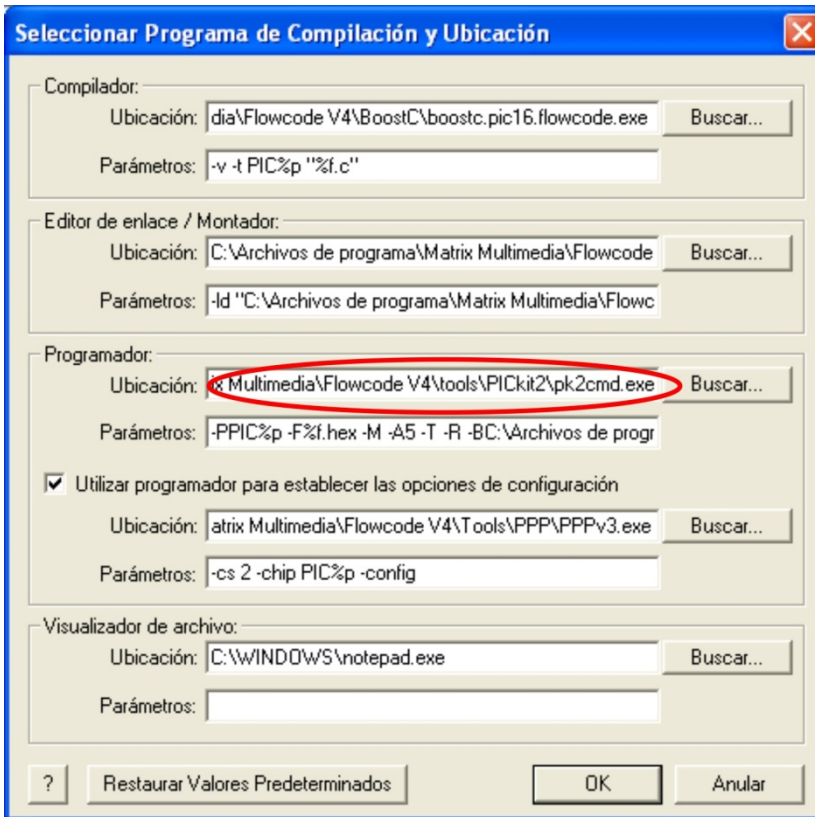
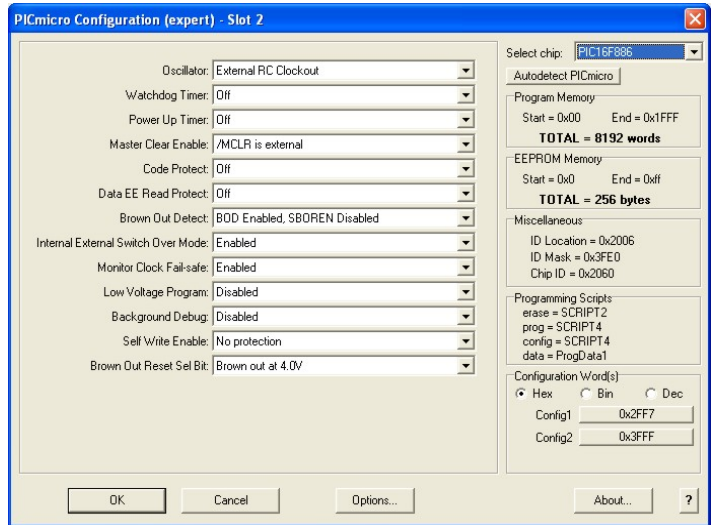


Figura AN12-2. Configuración por defecto del entorno del FlowCode 4

### AN12-3.2 Selección del microcontrolador

Esta acción, fundamental a lo hora de comenzar cualquier proyecto o aplicación, se realiza mediante “Chip” → “Configurar” → “Switch To Expert”. Aparece una nueva ventana como la mostrada en la figura AN12-3. En ella se ajustan parámetros propios del controlador que vayamos a utilizar. En el caso que nos ocupa hemos seleccionado el entrañable PIC16F886.

Podemos comprobar que su palabra de configuración queda establecida de forma similar a como lo hicimos en los ejemplos escritos en ensamblador y en C.



F  
Figura AN12-3. Selección y configuración del microcontrolador

### AN12-3.3 Opciones del proyecto

Para acabar. La configuración exige definir una serie de opciones propias del proyecto con el que vamos a trabajar. Éstas se eligen mediante “Ver” → “Opciones de Proyecto” que nos abre una ventana como la mostrada en la figura AN12-4.

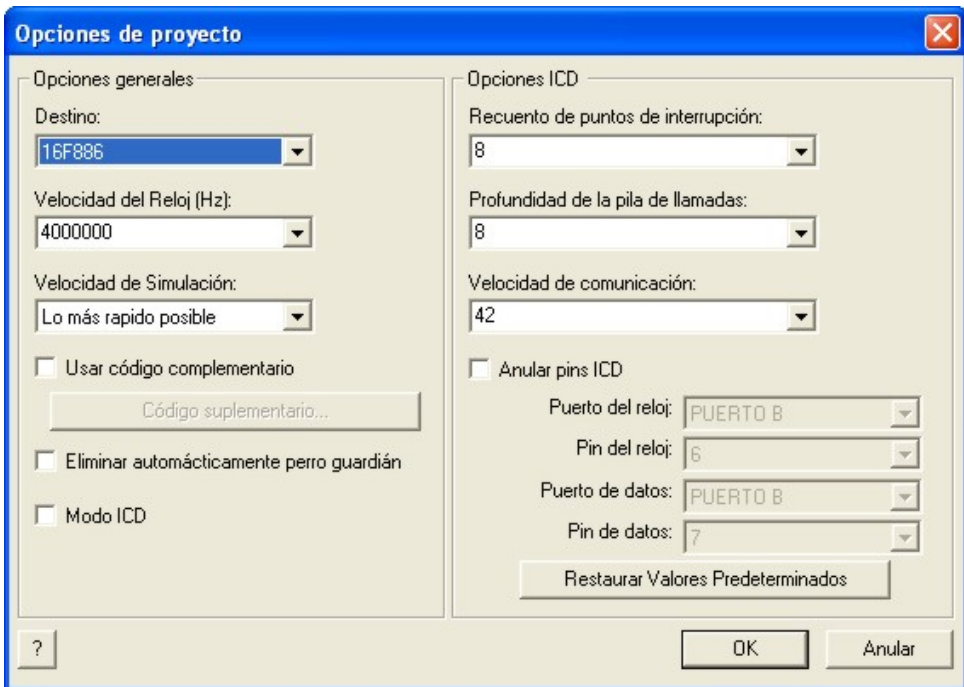


Figura AN12-4. Las opciones de un proyecto basado en FlowCode 4



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”.

## **AN12-4 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 confeccionar los programas. Todas ellas se encuentran en la columna de la izquierda de la pantalla de trabajo.

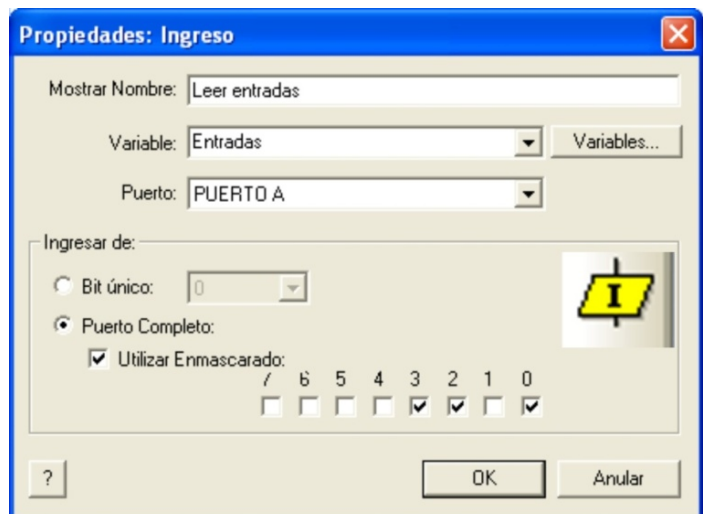
Cuando cualquiera de los iconos de comandos se traslada al área de trabajo, queda automáticamente conectado con el resto de iconos existentes. Vamos dibujando así, paso a paso, el organigrama. Al hacer un doble clic con el botón izquierdo del ratón sobre cualquiera de los iconos existentes, se abrirá 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).

### **AN12-4.1 Herramienta Ingreso**

Permite ejecutar tareas relacionadas con la adquisición de datos procedentes de los Puertos de entrada del controlador. La figura AN12-5 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



**Figura AN12-5. Herramienta de ingreso**

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 ese puerto son los que nos interesa leer. En el ejemplo se leen los bits 0,2 y 3 del Puerto A.

#### **AN12-4.2 Herramienta de Salida**

Permite ejecutar operaciones relacionadas con los Puertos de salida del controlador. En la figura AN12-6 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”. Mediante el campo “Puerto” se selecciona el puerto de salida, Puerto B en el ejemplo.

**Figura AN12-6. Herramienta de salida**

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.

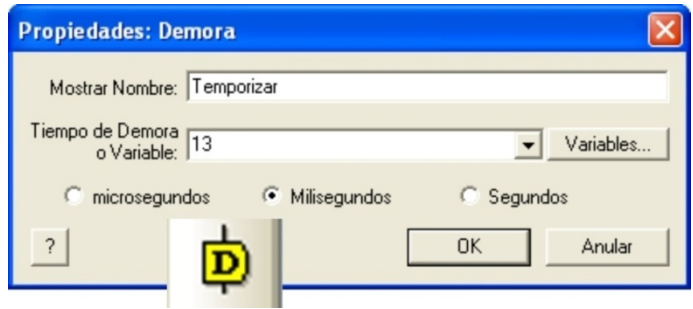
#### **AN12-4.3 Herramienta de Demora (temporización)**

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 AN12-7. 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.



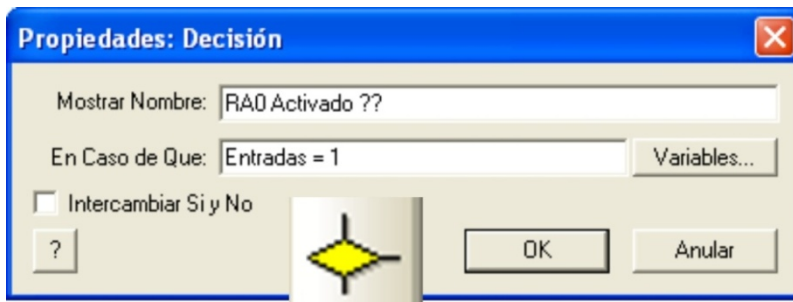
**Figura AN12-7. Herramienta de temporización**



#### **AN12-4.4 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 AN12-8 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.



**Figura AN12-8. Herramienta de decisión**

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                  |

#### **AN12-4.5 Herramienta de Expresión o decisiones múltiples**

La figura AN12-9 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 (Sí o No) en función de la evaluación. En este caso se pueden evaluar hasta once posibles condiciones. Resultan especialmente útiles a la hora de personalizar el programa para que reaccione de formas diferentes basándose en una variable de entrada.

Figura AN12-9. Herramienta de expresión



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 AN12-10.

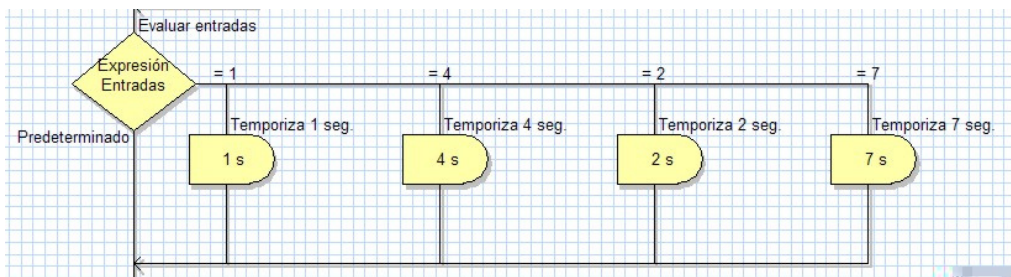


Figura AN12-10. Evaluando cuatro posibilidades

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

#### **AN12-4.6 Herramienta Punto de conexión**

Una de las tareas más frecuentes en el desarrollo de un diagrama de flujo o programa, es desviar la ejecución a un punto de destino de forma incondicional. Esta herramienta, cuya ventana de propiedades se muestra en la figura AN12-11, permite establecer un punto de conexión de destino.

En el campo “Etiqueta” asignamos un nombre o etiqueta al punto de destino. En el ejemplo hemos asignado la etiqueta “Bucle”.

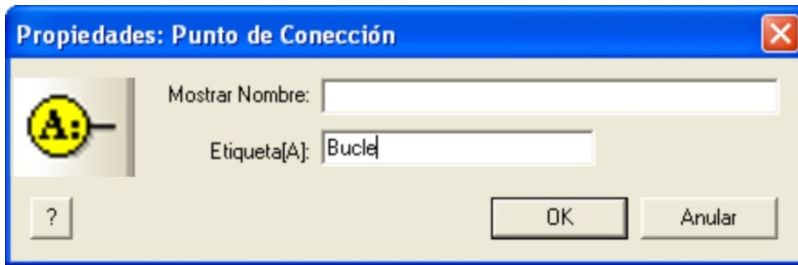


Figura AN12-11. Herramienta Punto de conexión

#### **AN12-4.7 Herramienta Ir a Punto de conexión**

Esta herramienta se emplea junto con la anterior. Permite establecer el punto de conexión de destino al que debe dirigirse el flujo del programa. La ventana de propiedades se muestra en la figura AN12-12.

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".

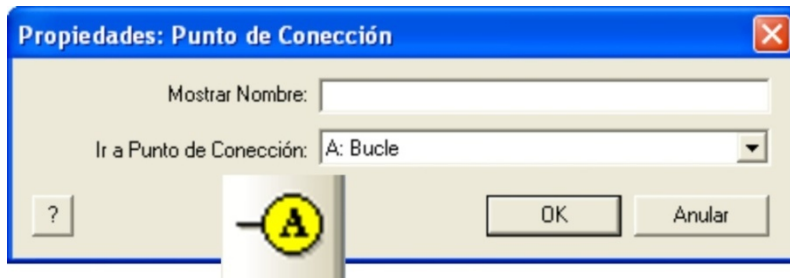
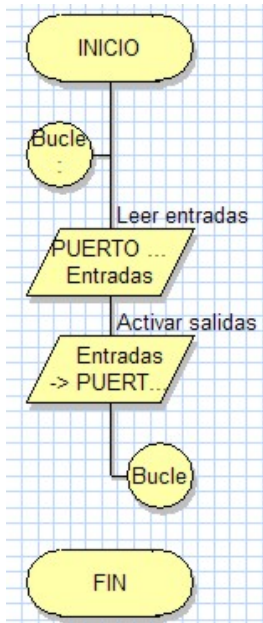


Figura AN12-12. Herramienta Ir a Punto de conexión



La figura AN12-13 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 del 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.

Figura AN12-13. Un sencillo bucle sin fin

## AN12-4.8 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 14 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.

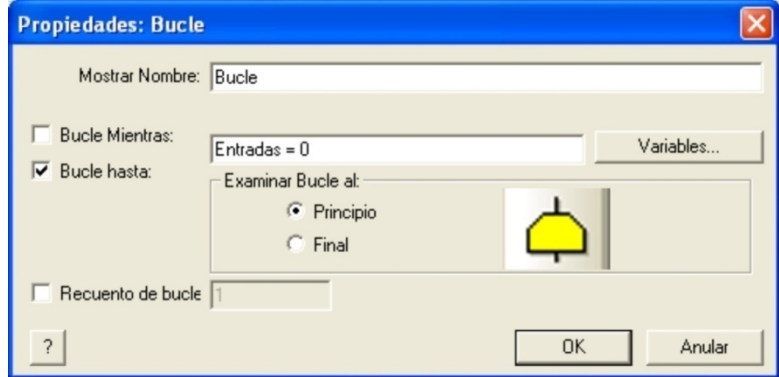
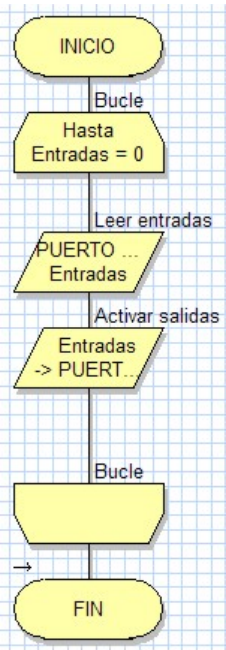


Figura AN12-14. Herramienta de Bucle



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 AN12-15 vemos un bucle que consiste en leer las entradas del Puerto A y reflejarlas en las salidas del Puerto B. El bucle se ejecuta hasta que las entradas valgan 0.

Figura AN12-15. Ejecución condicional de un bucle

## AN12-4.9 Herramienta Macro

Una macro no es ni más ni menos que una función o una subrutina, similar a las que hemos empleado en los distintos ejemplos de la obra. Una macro 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 llamando a la macro tantas veces como sea necesario, sin tener

que reescribir los comandos que la componen. La figura AN12-16 muestra la ventana de propiedades del comando Macro.

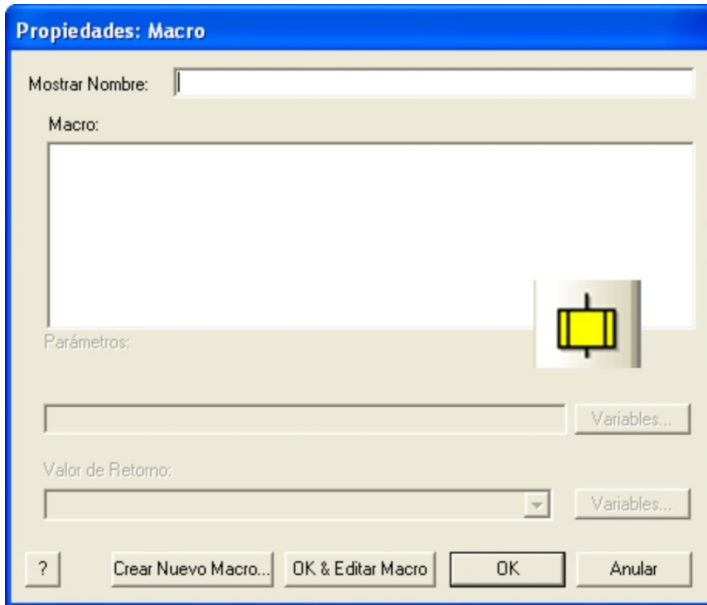


Figura AN12-16. Herramienta Macro

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.

#### AN12-4.9.1 Crear Nueva Macro

Cuando se pulsa el botón “Crear Nueva Macro” se nos abre una ventana como la mostrada en la figura AN12-17. 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 lugar del programa de aplicación.

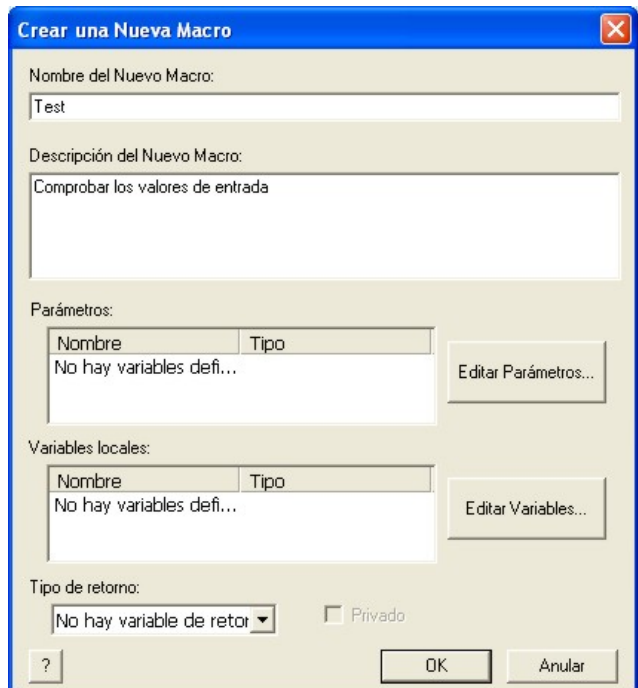


Figura AN12-17. Definición de una Macro

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.

### AN12-4.9.2 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 AN12-18 se muestra un ejemplo de la Macro que hemos llamado “*Test*”.

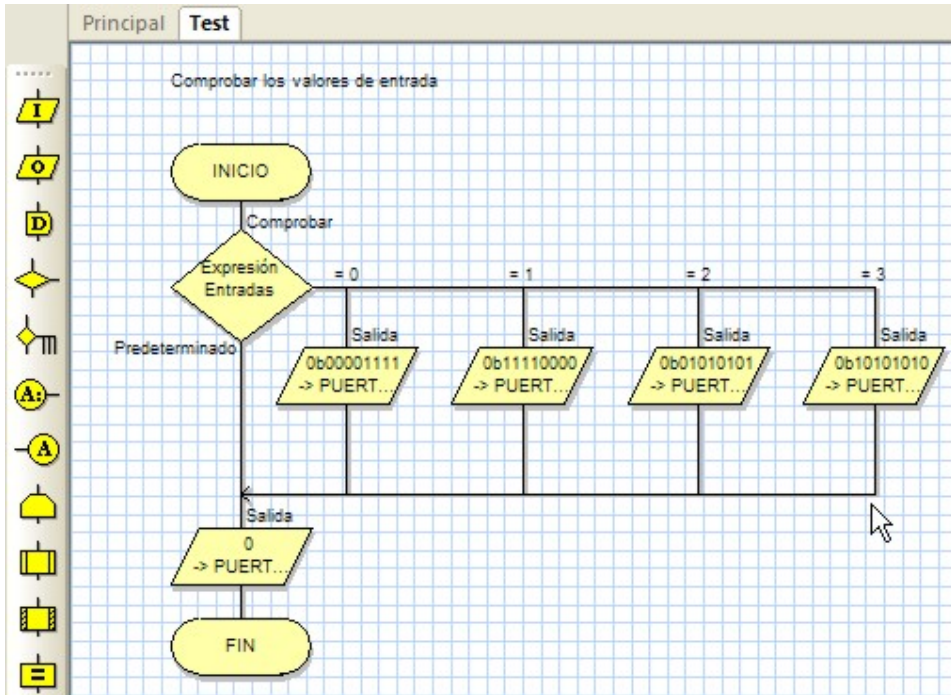


Figura AN12-18. Contenido de la Macro “*Test*”

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 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.

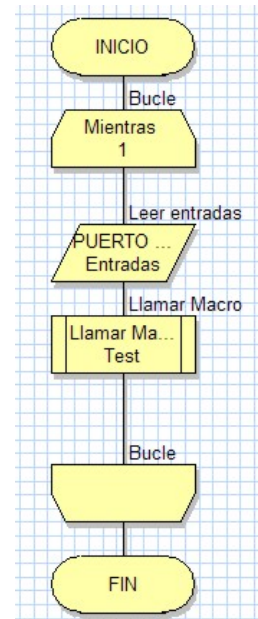


En la figura AN12-19 se muestra cómo queda el programa principal. Consiste en un bucle sin fin que ejecuta los siguientes comandos:

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 Macros, que no dejan de ser funciones o subrutinas, facilita enormemente la comprensión de un programa.

**Figura AN12-19.** El programa principal del ejemplo



### AN12-4.9.3 El comando Macro del menú principal

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

- **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 seleccionemos aquélla 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 utilizadas en múltiples proyectos o aplicaciones.
- **Importar.** Esta opción permite leer y recuperar desde un archivo en disco las distintas Macros que hubiera almacenadas.

### AN12-4.10 Herramienta Macro 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 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 AN12-20. 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:



**Figura AN12-20.** Herramienta de Macro de Componentes

1. **LeerEstado.** Lee el estado de la línea de entrada a la que está conectado el SWITCH(0) y los 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 AN12-21 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.

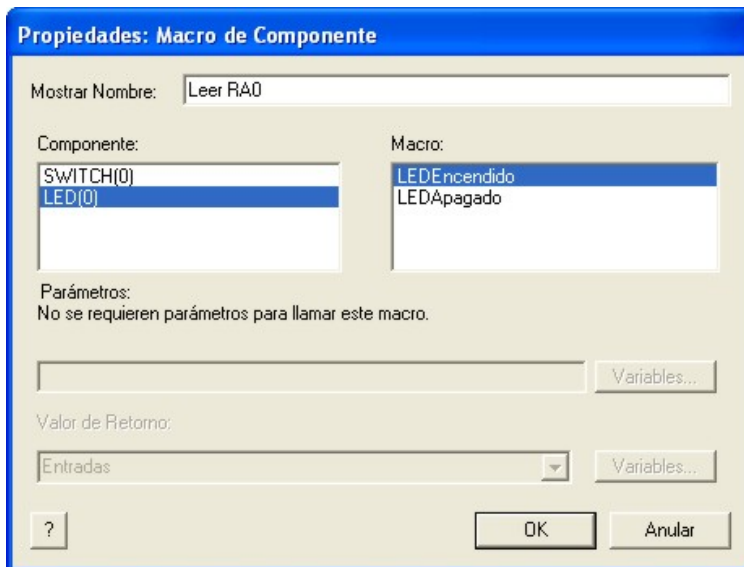


Figura AN12-21. Herramienta Macro de Componentes

En la figura AN12-22 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 SWITCH(0) y un led, el LED(0). Ambos componentes se suponen previamente dibujados y definidos en el panel de montaje.

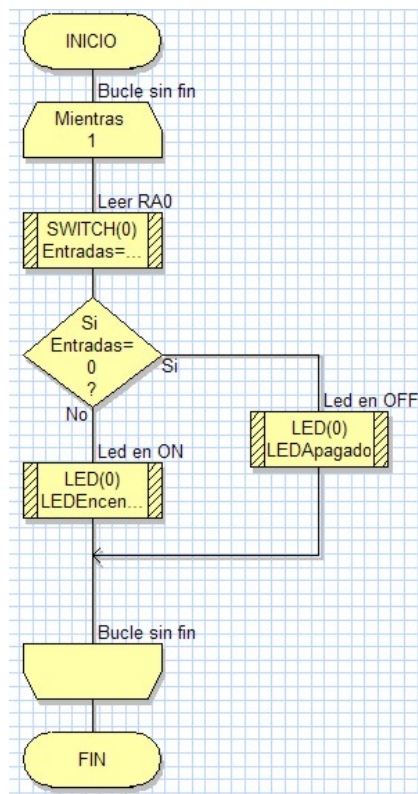
La primera Macro de componentes lee el estado del SWITCH(0) conectado con la línea RA0, y lo deposita en la variable “Entradas”. Éste se evalúa mediante la herramienta de decisión.

Si RA0=0 se ejecuta la función “LEDApagado” contenida en la Macro de componentes LED(0). La línea de salida RB0 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 RB0 se pone a nivel “1” y el led se enciende.

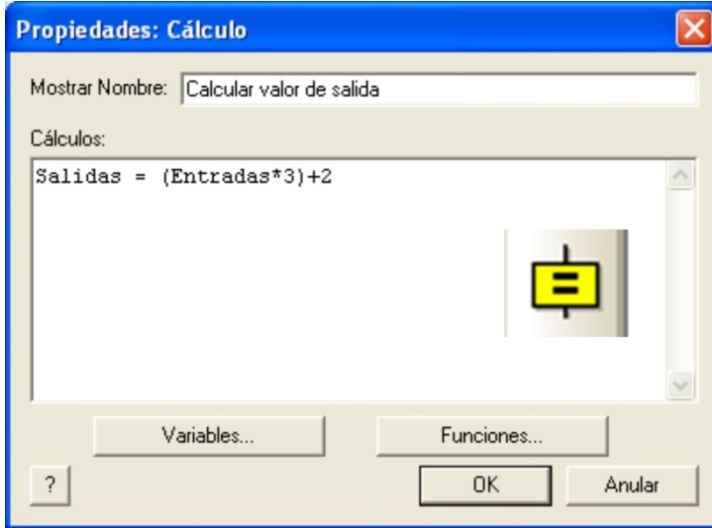
Se trata de un bucle sin fin que se ejecuta constantemente.

Figura AN12-22. Ejemplo con Macros de Componentes



### AN12-4.11 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 AN12-23.



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

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 lógico/aritméticas disponibles.

**Figura AN12-23.** La Herramienta de Cálculo

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

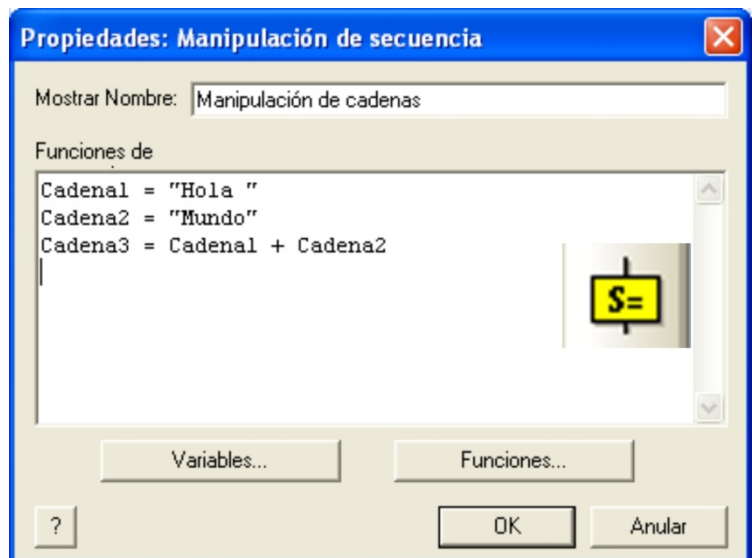
### AN12-4.12 Herramienta Manipulación de Secuencia

La herramienta de Manipulación de Secuencia permite manipular y operar con cadenas de caracteres ASCII. Su ventana de propiedades se muestra en la figura AN12-24.

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.

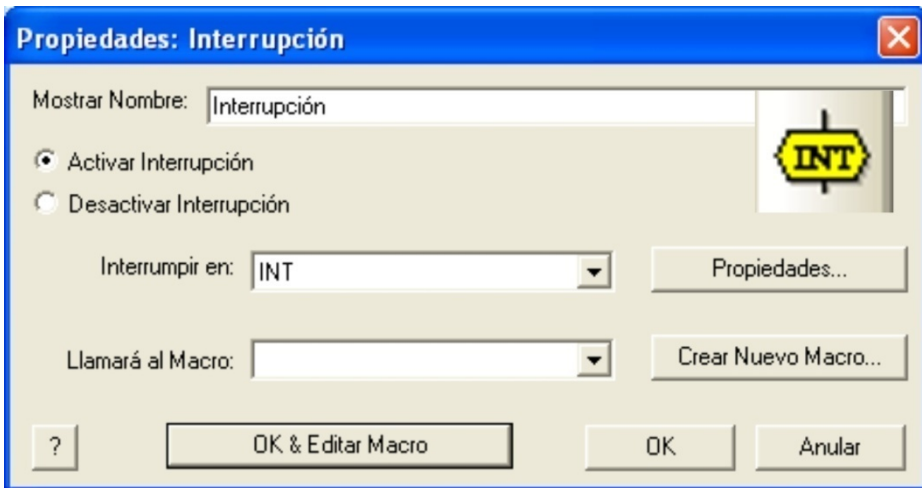


**Figura AN12-24.** Herramienta de Manipulación de Secuencia

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”.

#### **AN12-4.13 Herramienta de Interrupción**

FlowCode 4 también soporta la activación/desactivación de algunas de las interrupciones más relevantes del microcontrolador seleccionado. En la figura AN12-25 se muestra la ventana de propiedades de esta herramienta.



**Figura AN12-25. Herramienta de Interrupciones**

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*”.

#### **AN12-4.14 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 AN12-26. Simplemente consiste en un editor de textos donde escribimos nuestro programa empleando instrucciones del lenguaje C o del lenguaje ensamblador.

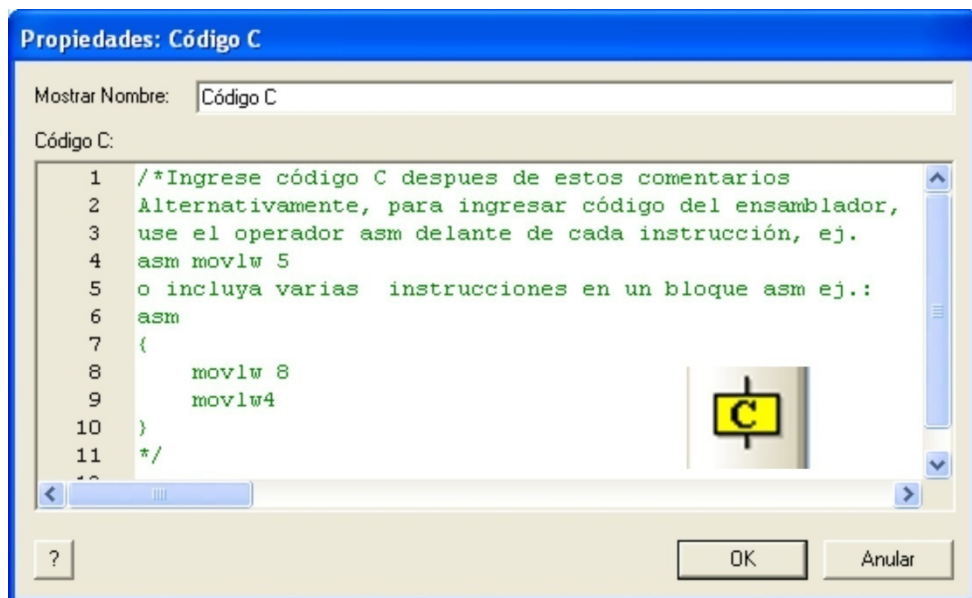


Figura AN12-26. Herramienta para insertar código C o ensamblador

#### **AN12-4.15 Herramienta de Comentarios**

Es la última herramienta de comandos que dispone FlowCode 4. Su ventana de propiedades se muestra en la figura AN12-27 y consiste simplemente en un área donde podemos insertar todo tipo de textos que describan el programa y facilite su comprensión y análisis.

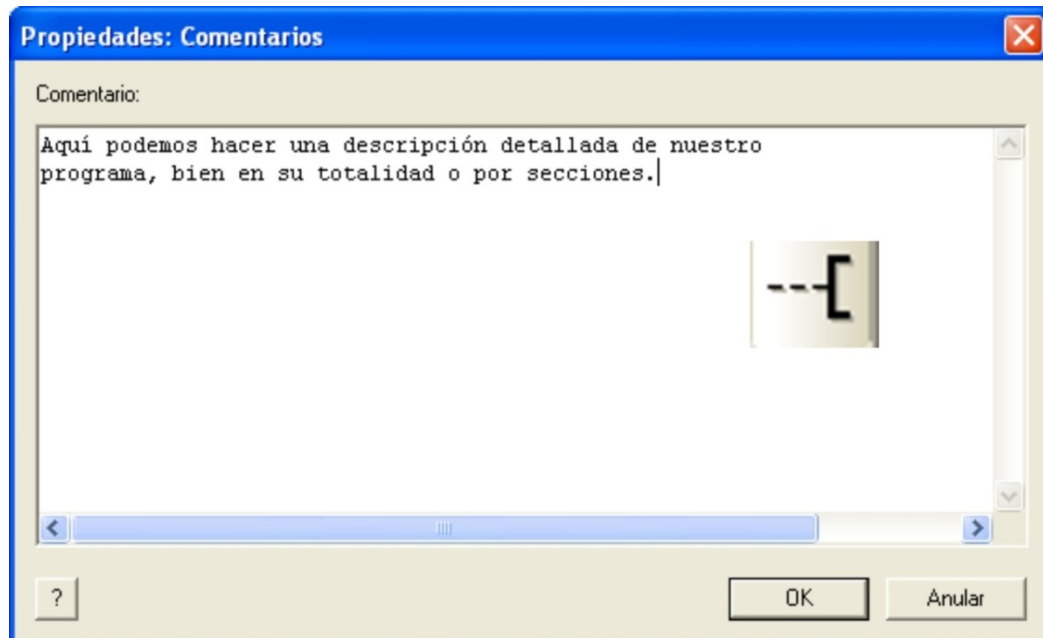


Figura AN12-27. Herramienta de Comentarios

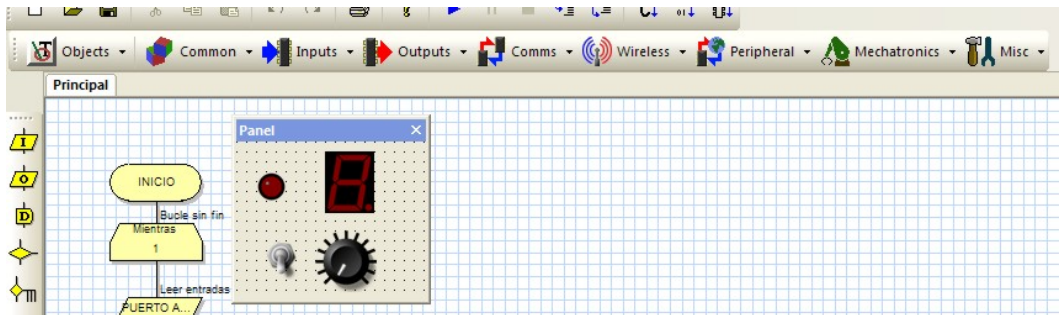


## AN12-5 HERRAMIENTAS DE COMPONENTES

El programa FlowCode 4 integra un buen número de componentes y periféricos que nos permiten diseñar el esquema eléctrico de nuestra aplicación, y simular el funcionamiento de la misma. Todos y cada uno de los componentes se dibujan sobre el panel de montaje y disponen de una serie de Macros o funciones que permiten hacer uso de los mismos de forma rápida y sencilla. Nuestro programa de aplicación accede y hace uso de las diferentes funciones relacionadas con los componentes, mediante la herramienta de comandos “*Macro de Componente*”, tal y como se explicó en el apartado AN12-4.10.

### AN12-5.1 Clasificación de los componentes

Para acceder a los componentes es necesario tener habilitado el panel de montaje mediante el comando “*Ver → Panel*” y la barra de componentes mediante “*Ver → Herramientas de componentes*”. Ésta se presenta en la figura AN12-28.



**Figura AN12-28.** El Panel de montaje y las herramientas de componentes

Los componentes se organizan en nueve grupos que se seleccionan mediante otros tantos botones de la barra de componentes. Al pulsar uno de ellos, se despliegan los componentes disponibles. Cada componente colocado en el panel de montaje, así como el propio panel, tiene una serie de propiedades y características que se pueden ajustar en la ventana de Propiedades de los componentes. Ésta se activa/desactiva mediante “*Ver → Panel Properties*”.

#### **AN12-5.1.1 Objets**

Contiene componentes con los que diseñar el aspecto y la estética del panel de montaje.

**Texto.** Permite escribir texto en el panel de montaje para establecer etiquetas, rótulos, signos, etc., en nuestro circuito.

**Imagen.** Podemos importar una imagen desde el disco, para colocarla en el panel de montaje.

**Forma.** Dibuja figuras geométricas como cuadrados, círculos, elipses, etc.

**Bisel.** Dibuja formas biseladas sobre el panel.

**Línea.** Dibuja líneas rectas en el panel de montaje.

#### **AN12-5.1.2 Common**

Aquí se agrupan los componentes o periféricos utilizados con mayor frecuencia.

**Led.** Coloca un led en el panel de montaje.

**LEDarray.** Coloca una barra de hasta ocho diodos leds.

**SWITCHbank.** Coloca un banco de hasta ocho interruptores.

**SWITCH.** Interruptor sencillo.

**ADC.** Coloca un potenciómetro para emular una entrada analógica.

**LCDDisplay.** Inserta una pantalla LCD alfanumérica.

### **AN12-5.1.3 Inputs**

Contiene componentes o periféricos típicos de entrada. Algunos son los mismos que los que había en el grupo “*Common*”.

**KeyPad.** Teclado matricial de hasta 4 x 4.

**SWITCHbank.** Coloca un banco de hasta ocho interruptores.

**SWITCH.** Interruptor sencillo

**ADC.** Coloca un potenciómetro para emular una entrada analógica.

### **AN12-5.1.4 Outputs**

Contiene componentes o periféricos típicos de salida. Algunos son los mismos que los que había en el grupo “*Common*”.

**Led7seg.** Coloca el clásico display de 7 segmentos con punto decimal.

**Led.** Coloca un led en el panel de montaje.

**LEDarray.** Coloca una barra de hasta ocho diodos leds.

**gLCD.** Pantalla gráfica.

**LCDDisplay.** Inserta una pantalla LCD alfanumérica.

**Led7seg4.** Inserta un conjunto de 4 displays de 7 segmentos conectados en paralelo.

### **AN12-5.1.5 Comms**

Este grupo contiene componentes destinados a los diferentes modos y protocolos de comunicación.

**LinMaster.** Este componente permite hacer uso de Macros y funciones que implementan el protocolo Lin en modo Master para la transferencia de datos.

**SPI\_Legacy.** Este componente permite hacer uso de Macros y funciones que implementan el protocolo SPI para la transferencia de datos.

**One\_Wire.** Este componente permite hacer uso de Macros y funciones que implementan el protocolo One\_Wire para la transferencia de datos.

**I2C\_Master.** Este componente permite hacer uso de Macros y funciones que implementan el protocolo I2C en el modo Master para la transferencia de datos.

**CAN2.** Permite hacer uso de Macros y funciones para realizar transferencia de datos según el protocolo CAN2.

**RS232.** Este componente permite hacer uso de Macros y funciones que implementan el estándar RS232 para la transferencia de datos.

**Lin\_Slave.** Este componente permite hacer uso de Macros y funciones que implementan el protocolo Lin en modo Slave para la transferencia de datos.

**CAN.** Permite hacer uso de Macros y funciones para realizar transferencia de datos según el protocolo CAN.

### **AN12-5.1.6 Wireless**

Estos componentes y sus correspondientes funciones o Macros permiten realizar transferencia de datos sin cables.

**IrDA** Comunicación mediante infrarrojos.

**RFID.** Identificación por radiofrecuencia.

**Zigbee.** Comunicación por radiofrecuencia según el protocolo Zigbee

**Bluetooth.** Comunicación por radiofrecuencia según el protocolo Bluetooth.

### AN12-5.1.7 Peripheral

Sus componentes, junto con las Macros y Funciones que poseen, permiten implementar distintos tipos de periféricos y dispositivos.

**USBSerial.** Permite implementar un dispositivo USB para la comunicación serie.

**TCP\_IP.** Permite la transferencia de datos según el estándar TCP\_IP.

**USBHID.** Permite implementar un dispositivo USB como dispositivo de interface humano (HID).

**MIDI.** Usado para la comunicación en tiempo real entre diferentes dispositivos como controladores musicales o instrumentos.

**FAT16.** Este componente está diseñado para usarse junto con tarjetas de memoria MMC/SD para configurar el sistema de localización de archivos (FAT) y empleo de los mismos.

**WebServer.** Con este componente podemos diseñar aplicaciones basadas en servicios de páginas web.

**USBSlave.** Permite implementar un dispositivo USB esclavo.

### AN12-5.1.8 Mechatronics

Contiene componentes auxiliares

**Stepper.** Para la simulación de aplicaciones que controlan un motor paso a paso

**PWM.** Componente cuyas Macros o funciones podemos emplear para generar y visualizar señales PWM moduladas en anchura.

**Speech.** Permite el diseño de aplicaciones con síntesis de voz.

**Servo.** Para la simulación de aplicaciones que controlan un servo motor.

### AN12-5.1.9 Misc

En este grupo encontramos una serie de componentes varios

**Custom2.** Componente nº 2 para ser definido por el usuario

**Custom.** Componente nº 1 para ser definido por el usuario

**MIAC.** Es un componente diseñado por Matrix Multimedia. Representa a un panel de control de propósito general que fabrica y comercializa Matrix Multimedia.

**FormulaFlowcode.** Este componente emula al robot “*FormulaFlowcode*” que fabrica y comercializa Matrix Multimedia.

**EEPROM.** Con este componente podemos realizar aplicaciones basadas en la EEPROM de datos del controlador.

**GPS.** Este componente emula al GPS que fabrica y comercializa Matrix Multimedia.

## AN12-5.2 Configuración y uso de los componentes

Ya hemos hecho una breve descripción de todos los componentes que posee FlowCode 4 para confeccionar, sobre el panel de montaje, el esquema eléctrico de nuestra aplicación. Ahora vamos a explicar cómo configurarlos para que puedan ser usados por nuestros programas. Debemos seguir tres pasos elementales.

### AN12-5.2.1 Uso del componente

Cualquier componente que hayamos colocado en el panel de montaje puede ser usado por nuestro programa mediante el icono “*Herramientas de componentes*”. Con él podemos seleccionar cualquiera de ellos así como la Macro o función que deseamos ejecutar de entre todas las disponibles.

### AN12-5.2.2 Establecer las conexiones del componente

Lo siguiente que debemos establecer son las conexiones de ese componente con las patillas de E/S del microcontrolador. Lo seleccionamos en el panel de montaje haciendo clic con el botón derecho del ratón y elegir la opción "Conexiones". Esto mismo también se puede hacer desde la ventana de "Propiedades" del panel de montaje. Supongamos que se ha seleccionado un display de 7 segmentos. Aparece una ventana de conexiones como la mostrada en la figura AN12-29.

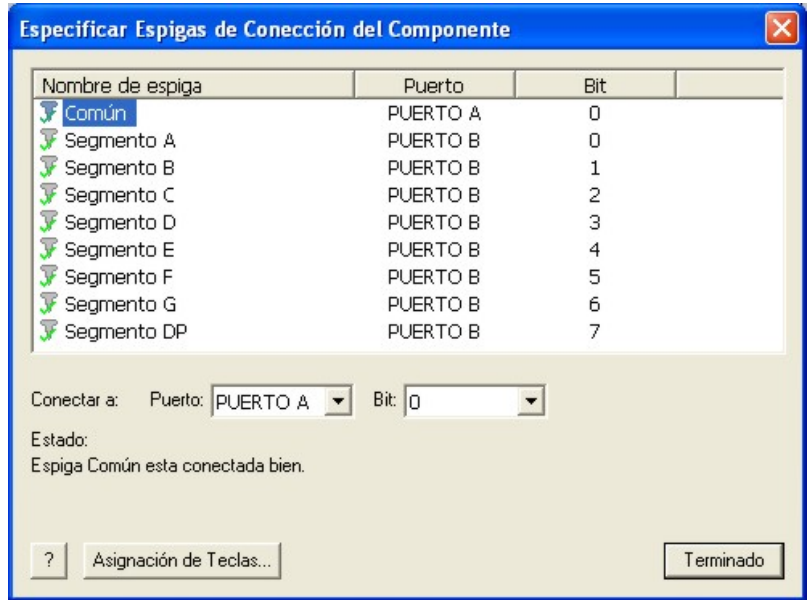


Figura AN12-29. Estableciendo conexiones de un display de 7 segmentos

En el ejemplo, la patilla común del display (el ánodo o el cátodo) se conecta con la línea 0 del Puerto A y los segmentos a:dp con las líneas 0 a 7 del Puerto B. Mediante el botón "Asignación de Teclas" podemos establecer qué teclas del teclado del PC podemos utilizar para estimular al componente cuando el programa se encuentra en ejecución de simulación.

No todos los componentes necesitan siempre las mismas conexiones. Habrá algunos que se conectan mediante una única patilla, otros, mediante dos y otros mediante diez. En cualquier caso la ventana de conexiones será similar a todos ellos.



### AN12-5.2.3 Establecer las propiedades externas del componente

Esto lo hacemos seleccionando el componente, haciendo clic con el botón derecho del ratón, y eligiendo la opción "Propiedades ext". También desde la ventana de propiedades del panel de montaje. Cada componente tiene sus propias propiedades. En el caso de un display de 7 segmentos nos aparece una ventana como la mostrada en la figura AN12-30.

Podemos seleccionar el color de los segmentos, el tamaño del display y si es de ánodo o cátodo común.

Figura AN12-30. Propiedades del display de 7 segmentos

A modo de ejemplo, en la figura AN12-31, se muestra la ventana de propiedades de un clásico periférico o componente de entrada como es un sencillo interruptor. Podemos seleccionar de entre varios tipos: oscilante, basculante, rotatorio, pulsante, etc. También debemos establecer si el mecanismo tiene o no enclavamiento.

En el campo "Polaridad" establecemos si el interruptor es activo por nivel "1" o por nivel "0" cada vez que se acciona. Incluso podemos determinar, en el campo "Supresión de rebotes", una temporización en mili segundos para evitar los rebotes clásicos de cualquier dispositivo electromecánico.

Por supuesto, podemos seleccionar el tamaño con el que se verá sobre el panel de montaje.

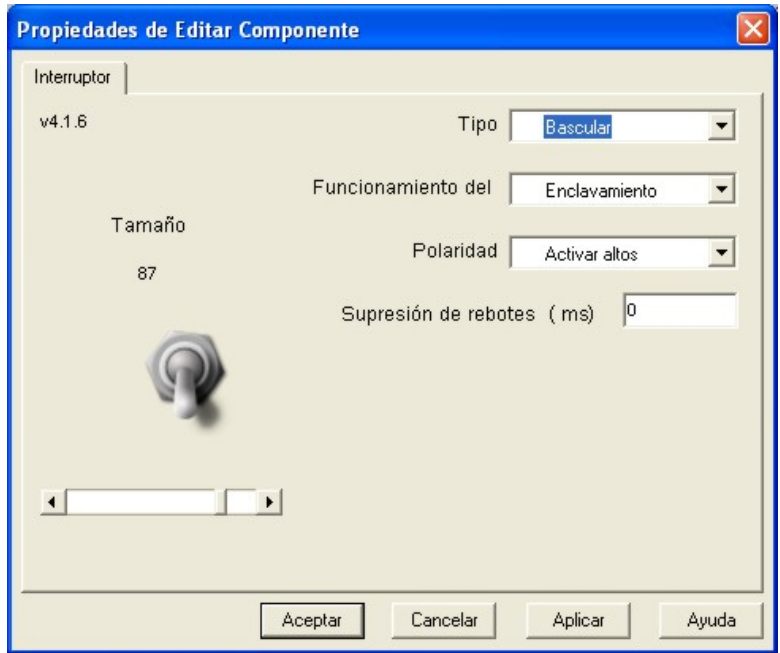


Figura AN12-31. Propiedades de un interruptor

En el caso de utilizar una pantalla LCD alfanumérica, sus propiedades serían las mostradas en la figura AN12-32.



Figura AN12-32. Propiedades de la pantalla LCD

Establecemos el tamaño o configuración de la pantalla. Por ejemplo, de 16 caracteres por fila y 2 filas (16x2). También determinamos el tamaño del texto o carácter, por ejemplo, 20 pixels.

Y por supuesto, también podemos establecer el color de fondo así como el color del carácter. En el ejemplo se configura una pantalla de fondo azul con letras blancas, igual que la del laboratorio USB-PIC'School.

## AN12-6 EJEMPLOS

Con objeto de familiarizarnos con el uso y las ventajas del lenguaje de programación FlowCode, vamos a presentar unos sencillos ejemplos que iremos desarrollando paso a paso.  
¡Manos a la obra!

### AN12-6.1 EJEMPLO 1

Se trata de leer un interruptor conectado a la entrada RA0. Si está a nivel "0" se activa un led conectado a la línea RB0 de salida. Si está a nivel "1" se activa un led conectado a la línea RB1 de salida.

#### AN12-6.1.1 El montaje eléctrico

En el panel de montaje distribuimos los componentes que vamos a utilizar más o menos como se muestra en la figura AN12-33. Empleamos dos leds y un interruptor. Recordar que para cada componente debemos definir sus propiedades (color, tamaño, forma, etc.). Para ello hacemos clic con el botón derecho y seleccionamos la opción "Propiedades ext". También hay que establecer las conexiones. Hacemos clic con el botón derecho y seleccionamos la opción "Conexiones". Mediante el componente "Texto" colocamos las etiquetas RB1, RB0 y RA0. Con las propiedades "Texto" podemos definir su color, tamaño, estilo, etc.

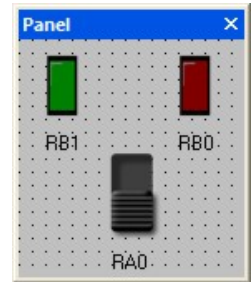
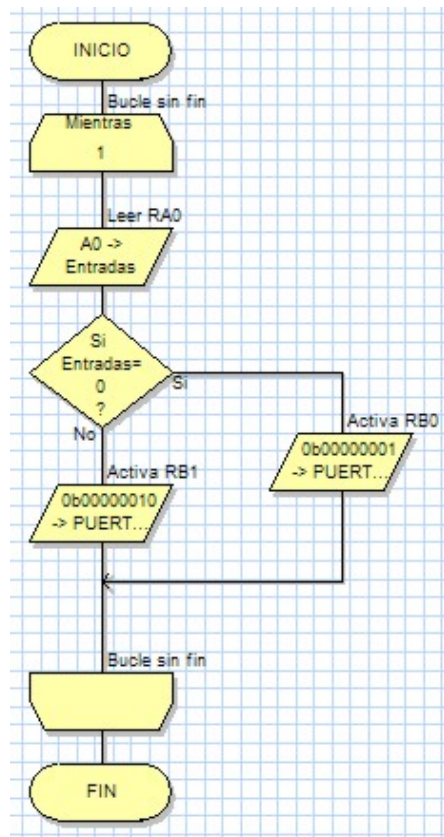


Figura AN12-33. Montaje del Ejemplo 1

#### AN12-6.1.2 El programa



En el área de trabajo hemos confeccionado el organigrama que se muestra en la figura AN12-34. Responde a las expectativas del ejemplo y, por supuesto, se podría haber resuelto de múltiples formas.

Todos los iconos o comandos que forman el programa se encuentran dentro de un bucle sin fin. Haciendo doble clic sobre cada uno de ellos podemos ver y/o editar sus correspondientes propiedades.

El programa empieza leyendo el estado del Puerto A que va a parar a la variable "Entradas". Si editamos esta herramienta de comando observamos que la lectura se realiza en el modo de bit único, el bit 0 del Puerto A. A continuación el comando de "Toma de decisión" evalúa si el valor de la variable "Entradas" (el estado de RA0) es igual o no a 0. En caso afirmativo el flujo de ejecución se desvía por la derecha. En caso negativo sigue su curso.

En cada caso se ejecuta el comando correspondiente "Salida". En un caso por el Puerto B se activa el bit RB0 y desactiva el bit RB1, y en el otro caso se desactiva RB0 y se activa RB1. La salida por el Puerto B se realiza enmascarando los bits 0 y 1. Podemos hacer doble clic en cada herramienta "Salida" para ver y/o editar sus propiedades.

Figura AN12-34. El organigrama del ejemplo 1



### AN12-6.1.3 Simulación

Una de las grandes aportaciones de FlowCode 4 es el poder simular en el PC el comportamiento del programa. Para ello disponemos de los botones mostrados en la siguiente figura AN12-35.

**Ejecutar.** Simula la ejecución del programa en su totalidad. Podemos interactuar con los componentes del panel de montaje para observar el desarrollo de dicho programa.

**Pausar.** Detiene la simulación de la ejecución del programa. Mediante el botón “Ejecutar” se reanuda la ejecución desde donde se detuvo.

**Detener.** Detiene la simulación de la ejecución del programa. Mediante el botón “Ejecutar” la ejecución comienza desde el principio.

**Entrar.** Simula la ejecución de los comandos paso a paso, uno tras otro.

**Saltar.** Igual que “Entrar” excepto que todos los comandos contenidos en una Macro se simulan ejecutar en tiempo real.

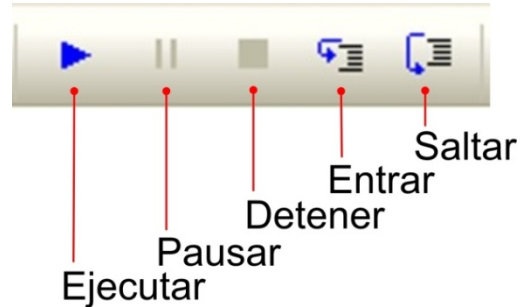
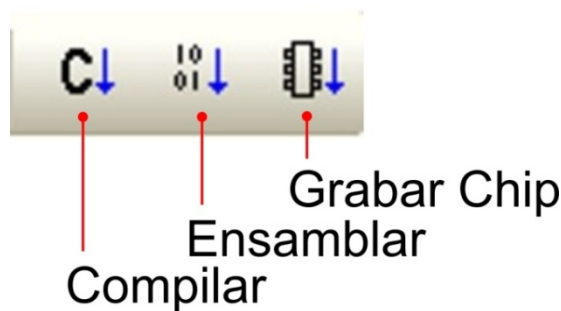


Figura AN12-35. Botones para la simulación

Cuando se entra en el modo de simulación se abren dos nuevas ventanas. La ventana “Variables” nos permite añadir las variables de nuestro programa, cuyo contenido queremos visualizar y/o modificar, según se va ejecutando y desarrollando el mismo. La ventana “Pila de llamadas” simplemente visualiza las distintas Macros o funciones según se van llamando desde el programa principal.

Llegado a este punto, basta con pulsar el botón “Ejecutar” para ver cómo se desarrolla nuestro programa. Haciendo clic sobre el interruptor que hemos puesto en el panel de montaje, los leds deben cambiar de estado.

### AN12-6.1.4 Grabación



No debemos olvidar que lo realizado hasta ahora no es ni más ni menos que una simulación. Es el momento de comprobar el funcionamiento real del programa. Para ello, procederemos a compilarlo, ensamblarlo y grabarlo físicamente sobre la memoria de programa del microcontrolador, el PIC16F886. Disponemos de los botones mostrados en la figura AN12-36.

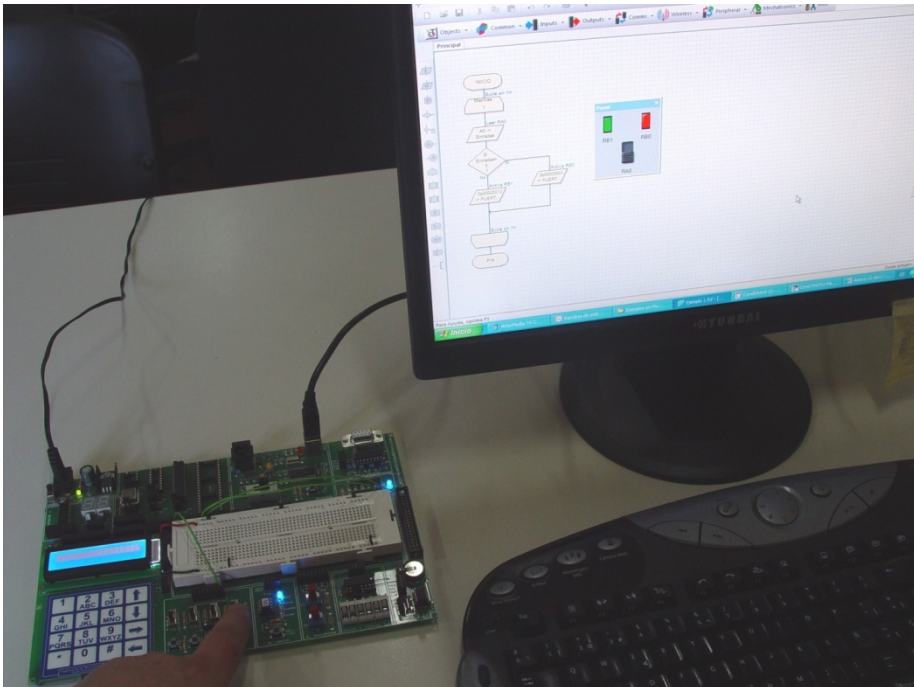
Figura AN12-36. Botones de ensamblado y grabación

**Compilar.** Consiste en traducir cada icono o comando del organigrama de nuestro ejemplo en el equivalente programa fuente en C. Nos crea un fichero con el mismo nombre pero con la extensión \*.C.

**Ensamblar.** A partir del programa fuente en C (\*.C) se obtiene el equivalente programa fuente en ensamblador con extensión \*.ASM. A partir del \*.ASM se obtiene el programa ejecutable \*.HEX. Éste es el fichero ejecutable que debemos grabar en la memoria del controlador. Para ello podemos utilizar cualquiera de los innumerables grabadores comerciales que existen en el mercado.

**Grabar Chip.** Este botón realiza todos los pasos anteriores de forma consecutiva y automática. A partir del organigrama se crea el \*.C. De aquí se obtiene el \*.ASM y de éste se obtiene el ejecutable \*.HEX. No sólo eso. Con este botón se procede también a grabar directamente el controlador. Para ello es necesario tener conectado uno de los que admite el propio FlowCode 4. El PICkit-2 es uno de ellos y... **¡es compatible con nuestro USB-PIC'School!**

En otras palabras, una vez desarrollado el organigrama de nuestra aplicación y, si disponemos de uno de los grabadores admitidos por FlowCode 4, basta con pulsar el botón "Grabar Chip" para realizar todo el proceso automáticamente. La fotografía de la figura AN12-37 muestra la ejecución del ejemplo sobre el laboratorio USB-PIC'School.



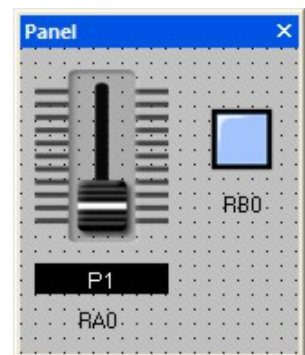
**Figura AN12-37.** Ejecución sobre USB-PIC'School

## **AN12-6.2 EJEMPLO 2**

En este ejemplo vamos a producir una intermitencia sobre un led conectado a la línea de salida RB0 del Puerto B. La frecuencia se establece mediante un potenciómetro conectado con el canal analógico de entrada RA0/AN0.

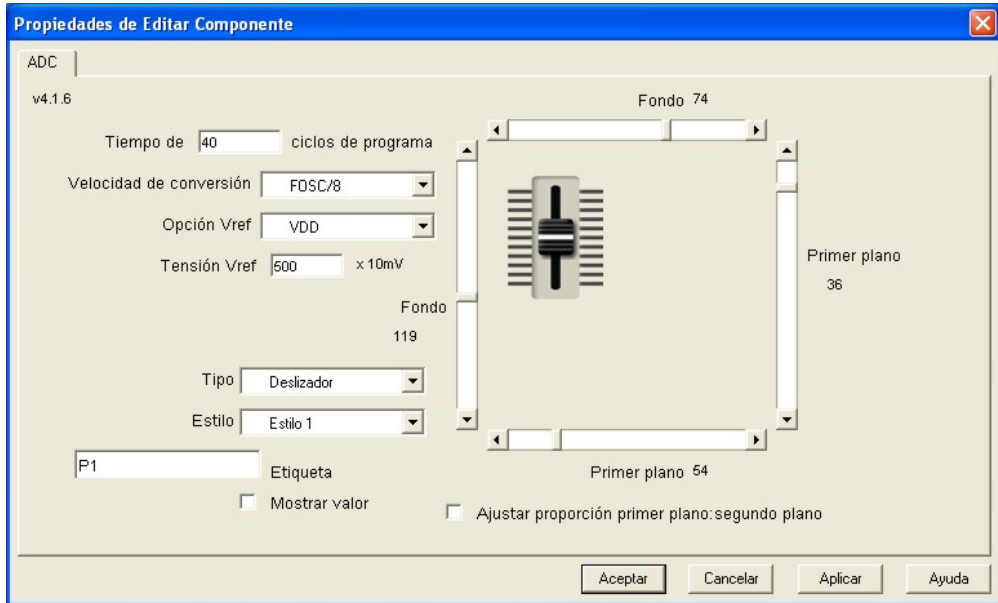
### **AN12-6.2.1 El montaje eléctrico**

En el panel de montaje realizamos el circuito mostrado en la figura AN12-38. Definimos las conexiones haciendo clic con el botón derecho y seleccionando la opción "Conexiones". El componente ADC (potenciómetro P1) se conecta con RA0/AN0 y el led en RB0



**Figura AN12-38.** Montaje del Ejemplo 2

También hay que definir las propiedades de cada componente para establecer su color, forma, tamaño, etc. Hacemos clic con el botón derecho y seleccionamos la opción “*Propiedades ext*”. En la figura AN12-39 vemos las propiedades que hemos establecido para el componente ADC (potenciómetro P1).



**Figura AN12-39.** *Propiedades del ADC (potenciómetro)*

En el campo “*Tiempo*” se establece el número de ciclos de programa que se deben esperar para el muestreo y retención (Sample&Hold) antes de iniciar la conversión. En “*Velocidad de conversión*” se establece la velocidad a la que trabajará el convertidor (FOSC/8). En “*Opción Vref*” seleccionamos el tipo de tensión de referencia y en “*Tensión Vref*” el valor de la misma.



También seleccionamos el tipo de potenciómetro (Deslizador), su estilo, y su tamaño. En el campo “*Etiqueta*” podemos poner la referencia del componente (P1).

La figura AN12-40 muestra las propiedades seleccionadas para el led: forma y color. También se establece la polaridad. Esto es, el nivel lógico con el cual se debe activar el led. En el ejemplo se activa mediante nivel “1”.

Con las barras deslizantes horizontal y vertical ajustamos el tamaño deseado.

**Figura AN12-40.** *Propiedades del componente led*

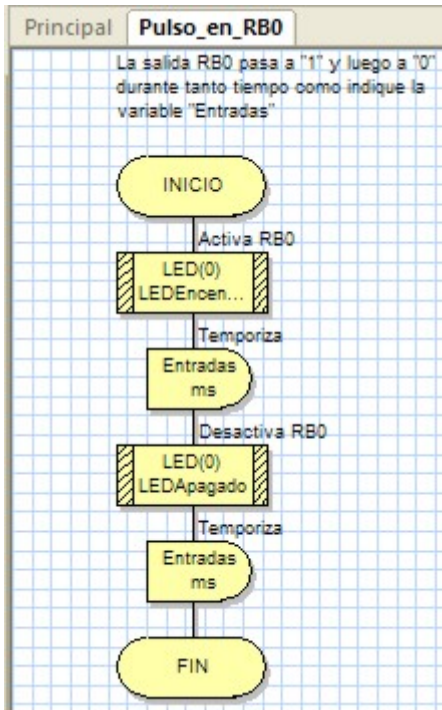
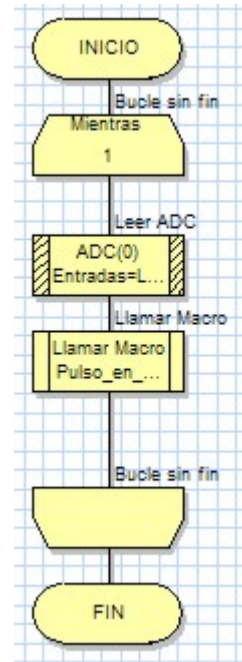
## AN12-6.2.2 El programa

El organigrama de este segundo ejemplo se muestra en la figura AN12-41. En primer lugar se emplea la herramienta “*Macro de componente*”. Nos muestra un listado con todos los componentes que haya en el panel de montaje (el potenciómetro ADC y el led). Seleccionamos el ADC y nos muestra todas las funciones disponibles para él. En el ejemplo se ha seleccionado la función “*leercomoint()*”. Lee el resultado de la conversión como un número entero de 16 bits y lo almacena en la variable “*Entradas*”.

Seguidamente se ejecuta una Macro o función que hemos desarrollado nosotros mismos y cuyo nombre es “*Pulso\_en\_RB0*”. Esta rutina se encarga de que la salida RB0 pase de nivel “0” a nivel “1” y vuelta al nivel “0” inicial.

Los dos comandos que forman nuestro programa, la lectura del ADC y la función “*Pulso\_en\_RB0*”, se encuentran dentro de un bucle que se ejecuta indefinidamente.

Figura AN12-41. El organigrama del ejemplo 2



En la figura AN12-42 se muestra el organigrama correspondiente a la función “*Pulso\_en\_RB0*” que hemos desarrollado para la ocasión.

Utilizamos la herramienta “*Macro de componentes*” en dos ocasiones. En ambas se selecciona el componente led. En la primera de ellas se ejecuta la función “*LEDEncendido()*” para activar el led y en la segunda “*LEDApagado()*” para desactivarlo.

Tras ellas se ejecutan sendos comandos de temporización. Consiste en mantener encendido y apagado el led tanto tiempo como establezca la variable “*Entradas*”. Ésta se cargó con el valor procedente del componente ADC (potenciómetro) en el programa principal. En otras palabras, la frecuencia de la intermitencia depende del valor analógico introducido con ese componente.

Éste es un buen ejemplo que ilustra el uso de las Macros, funciones o subrutinas definidas por el usuario, haciendo que el programa principal sea más sencillo y legible.

Figura AN12-42. Macro “*Pulso\_en\_RB0*”



### AN12-6.2.3 Simulación

Mediante el botón “Ejecutar” procedemos a simular el funcionamiento del programa. El led en el panel de montaje debe parpadear a una frecuencia variable. Movemos el potenciómetro P1 para variar dicha frecuencia. La ejecución se detiene al pulsar el botón “Detener”.

Éste también es un buen ejemplo para familiarizarnos con la ejecución paso a paso que nos ofrece FlowCode 4. Mediante el botón “Entrar” vemos cómo se van ejecutando todas las instrucciones, una tras otra. Incluso cuando toca ejecutar la Macro o función “Pulso\_en\_RB0”, observamos cómo se van ejecutando las instrucciones que contiene. Podemos experimentar también con el botón “Saltar”. Éste también ejecuta paso a paso todas las instrucciones excepto las Macros, que se ejecutan en tiempo real sin detenerse en cada una de las instrucciones que las componen.

En la ventana “Variables” visualizamos el contenido de la variable “Entradas”. Dicho contenido se verá modificado cada vez que se ejecute la Macro de Componentes “Leer ADC(0)” y se mueva el cursor del potenciómetro.

### AN12-6.2.4 Grabación

Si disponemos del laboratorio USB-PIC'School alimentado y conectado a un puerto USB podemos grabar directamente el microcontrolador con el programa del ejemplo. Basta pulsar el botón “Grabar Chip”. Con esta sencilla acción el FlowCode traduce nuestro programa escrito en lenguaje gráfico, a lenguaje fuente en C (\*.C). De aquí lo pasa a lenguaje ensamblador (\*.ASM), lo convierte en ejecutable (\*.HEX) y lo transfiere a la memoria del PIC.

Sobre el laboratorio realizamos las conexiones eléctricas correspondientes al montaje del ejemplo y observamos su ejecución, tal y como se muestra en la fotografía de la figura AN12-43.

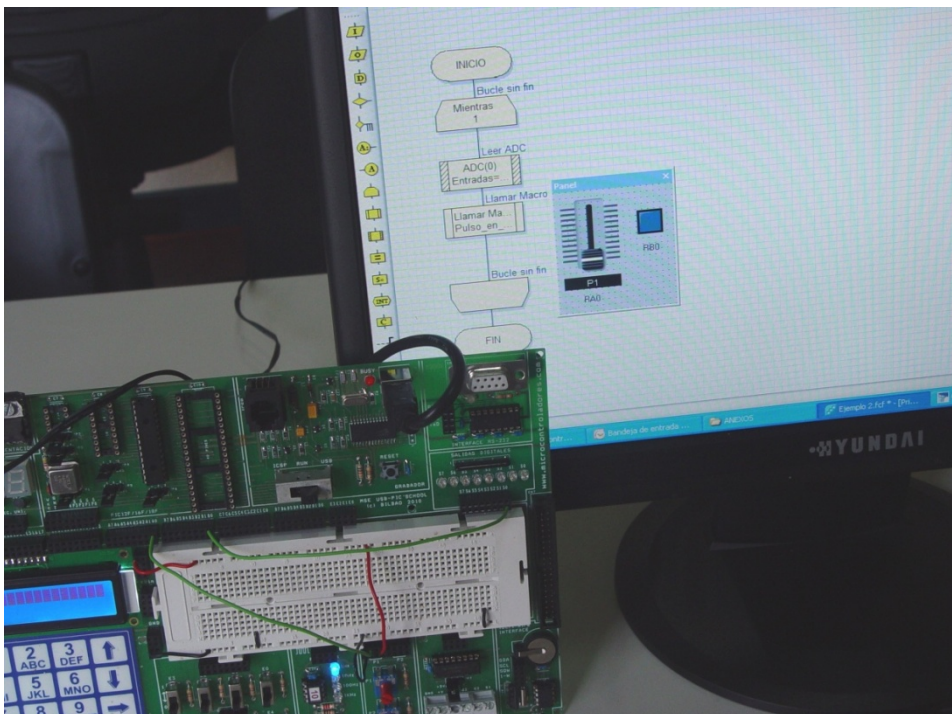


Figura AN12-43. Ejecución del Ejemplo 2 sobre el USB-PIC'School

### AN12-6.3 EJEMPLO 3

Con este ejemplo vamos a realizar un contador de pulsos. Éstos se introducen mediante un pulsador conectado en la línea de entrada RA0 del Puerto A y se visualizan sobre un display de 7 segmentos conectado a las líneas de salida RB7:RB0 del Puerto B. Se supone un display de ánodo común que se conecta y controla con la línea de salida RA2 del Puerto A.

#### AN12-6.3.1 El montaje eléctrico

En el panel de montaje realizamos el circuito mostrado en la figura AN12-44. Definimos las conexiones haciendo clic con el botón derecho y seleccionando la opción “Conexiones”. El componente pulsador se conecta con RA0. Por otra parte, los siete segmentos del display, a:dp, se conectan con las líneas RB0:RB7. El ánodo común se controla mediante la línea RA2.

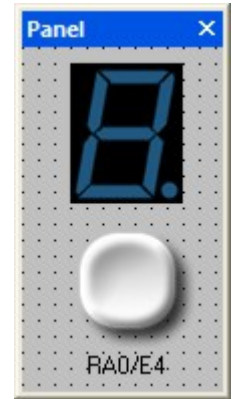


Figura AN12-44. Montaje del Ejemplo 3



Definimos ahora las propiedades de cada componente para establecer su color, forma, tamaño, etc. Hacemos clic con el botón derecho y seleccionamos la opción “Propiedades ext”. En la figura AN12-45 vemos las propiedades que hemos establecido para el display de siete segmentos.

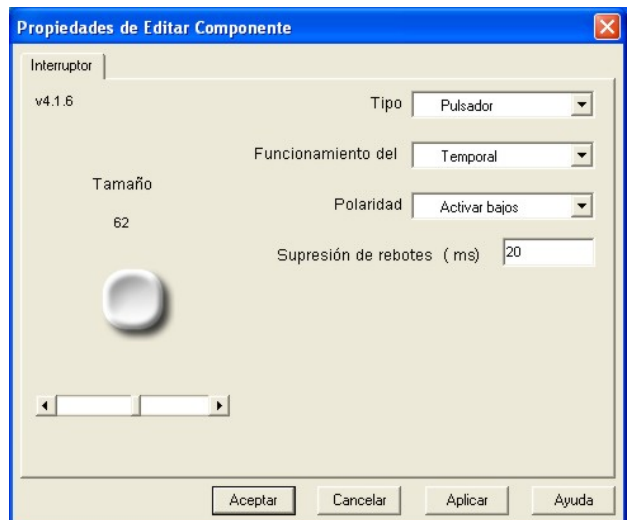
Los segmentos se visualizan en color azul y se selecciona un tamaño de 70 unidades. Trabajamos con un display del tipo de ánodo común.

Figura AN12-45. Propiedades del display

La figura AN12-46 muestra las propiedades exteriores seleccionadas para el pulsador. En “Tipo” elegimos el modelo de mecanismo: “Pulsador”, “Basculador”, “Giratorio”, etc.

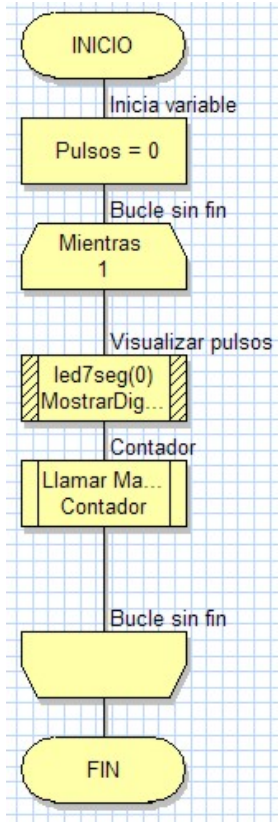
En “Funcionamiento del” indicamos si el mecanismo tiene enclavamiento o no (“Temporal”). También elegimos, en el campo “Polaridad”, el nivel activo del pulsador: por nivel bajo o nivel alto. Con la barra deslizante establecemos el tamaño.

Figura AN12-46. Propiedades del pulsador





### AN12-6.3.2 El programa



El organigrama de este ejemplo se muestra en la figura AN12-47. En primer lugar se emplea la herramienta “Cálculo” para poner inicialmente a 0 la variable “Pulsos”. A continuación empleamos la herramienta “Macro de componente”. Esta herramienta nos muestra un listado con todos los componentes que haya en el panel de montaje (el Display y el pulsador). Seleccionamos el Display y nos muestra todas las funciones disponibles para él. En el ejemplo se ha seleccionado la función “MostrarDigito()”. Lee el contenido actual de la variable “Pulsos” y lo visualiza sobre el Display.

Seguidamente se ejecuta una Macro o función que hemos desarrollado nosotros mismos y cuyo nombre es “Contador”. Esta rutina se encarga de esperar a que por la entrada RA0, se reciba una transición 1-0-1 procedente del pulsador. Cuando esto ocurre, la variable “Pulsos” se incrementa en una unidad.

Los dos comandos que forman nuestro programa, la visualización y la función “Contador”, se encuentran dentro de un bucle que se ejecuta indefinidamente.

Figura AN12-47. El organigrama del ejemplo 3

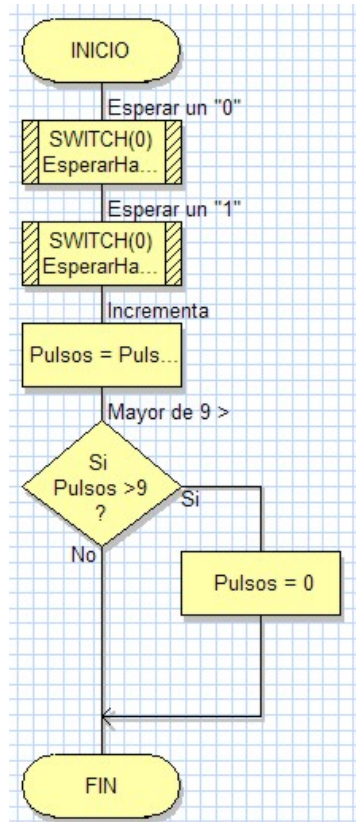
En la figura AN12-48 se muestra el organigrama correspondiente a la función “Contador” que hemos desarrollado para este ejemplo.

Utilizamos la herramienta “Macro de componentes” en dos ocasiones. En ambas se selecciona el componente pulsador SWITCH(). En la primera de ellas se ejecuta la función “EsperarHastaAlto()” y en la segunda “EsperarHastaBajo()”. Conseguimos así detectar el momento en que la línea a RA0 recibe una transición o pulso completo procedente del pulsador en ella conectada.

Recibido un pulso completo, empleamos el componente “Cálculo” para incrementar la variable “Pulsos” en una unidad.

Como únicamente disponemos de un display para visualizar el estado de la cuenta, necesitamos poner a 0 la variable “Pulsos” cuando el número de éstos sea superior a nueve. La cuenta por tanto evolucionará de cero a nueve y vuelta a cero.

Figura AN12-48. Macro “Contador”



### AN12-6.3.3 Simulación

Mediante botón “Ejecutar” procedemos a simular el funcionamiento del programa. En el display la cuenta empieza de cero. Cada accionamiento del pulsador origina un incremento de la cuenta. Observamos que ésta va desde cero a nueve y vuelta a empezar de cero. Es decir, hemos construido una década contadora.

Éste también es un buen ejemplo para familiarizarnos con la ejecución paso a paso que nos ofrece FlowCode 4. Mediante el botón “Entrar” vemos cómo se van ejecutando todas las instrucciones, una tras otra. Incluso cuando toca ejecutar la Macro o función “Contador”, observamos cómo se van ejecutando las instrucciones que contiene. Podemos experimentar también con el botón “Saltar”. Esta opción también ejecuta paso a paso todas las instrucciones excepto las que hay en las Macros o funciones, que se ejecutan en tiempo real sin detenerse en cada una de ellas.

En la ventana “Variables” visualizamos el contenido de la variable “Pulsos”. Dicho contenido se verá modificado cada vez que se detecte un accionamiento en el pulsador de entrada

### AN12-6.3.4 Grabación

Si disponemos del laboratorio USB-PIC'School alimentado y conectado a un puerto USB podemos grabar directamente el microcontrolador con el programa del ejemplo. Basta pulsar el botón “Grabar Chip”. Con esta sencilla acción el FlowCode 4 traduce nuestro programa escrito en lenguaje gráfico, a lenguaje fuente en C (\*.C). De aquí lo pasa a lenguaje ensamblador (\*.ASM), lo convierte en ejecutable (\*.HEX) y lo transfiere a la memoria del PIC.

Sobre el laboratorio realizamos las conexiones eléctricas correspondientes al montaje de este ejemplo. El pulsador E4 se conecta con la entrada RA0 por donde aplicamos los pulsos a contar. Las líneas RB0:RB7 del Puerto B se conectan con los segmentos a:dp del display. El ánodo común de éste se controla desde la salida RA2. Observamos su ejecución, tal y como se muestra en la fotografía de la figura AN12-49.

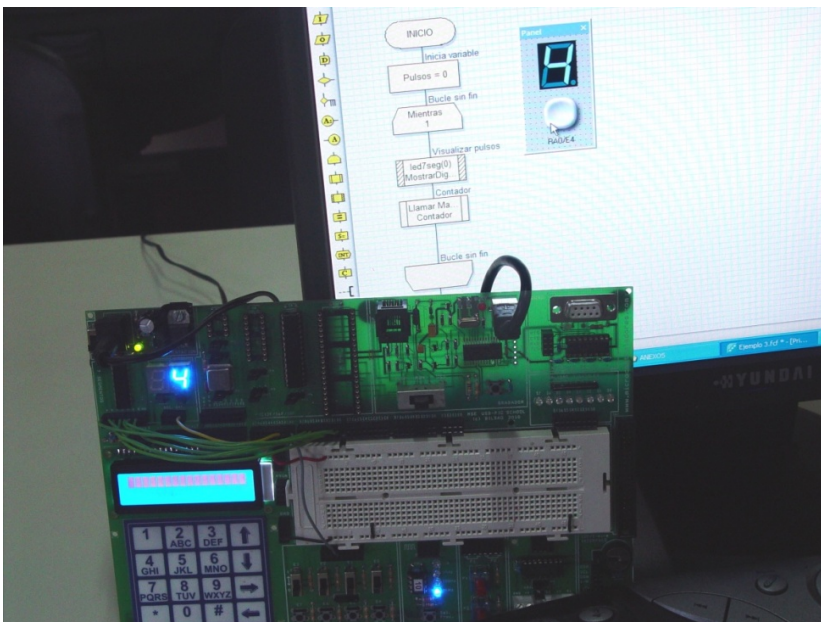


Figura AN12-49. Ejecución del Ejemplo 3 sobre el laboratorio USB-PIC'School

## AN12-6.4 EJEMPLO 4

Un último ejemplo en el que vamos a emplear el componente LCD para visualizar mensajes.

### AN12-6.4.1 El montaje eléctrico

En el panel de montaje realizamos el circuito mostrado en la figura AN12-50. Únicamente consta de una pantalla LCD alfanumérica de 2 líneas y 16 caracteres por línea.

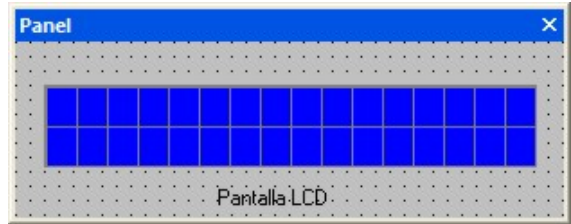
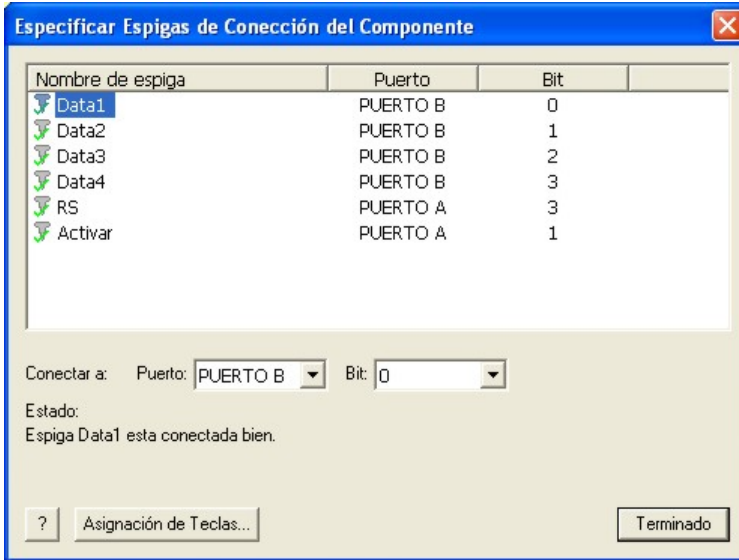


Figura AN12-50. Montaje del ejemplo 4



Como se muestra en la figura AN12-51, definimos las conexiones haciendo clic con el botón derecho y seleccionando la opción "Conexiones". Las líneas de datos Data1:Data4 del componente LCD se conectan con las líneas RB0:RB3 del Puerto B.

La señal RS se controla desde la salida RA3 del Puerto A y la señal de activación de la pantalla LCD desde la salida RA1.

Figura AN12-51. Montaje del Ejemplo 4

También hay que definir las propiedades del componente para establecer su color, forma, tamaño, etc. Hacemos clic con el botón derecho y seleccionamos la opción "Propiedades ext". En la figura AN12-52 vemos las propiedades que hemos establecido para la pantalla LCD. El tamaño de la pantalla es de 16x2 (2 líneas de 16 caracteres). El tamaño de cada carácter se ajusta a 20 pixels. El color de los caracteres se fija en blanco y el color de fondo de la pantalla se establece en azul.

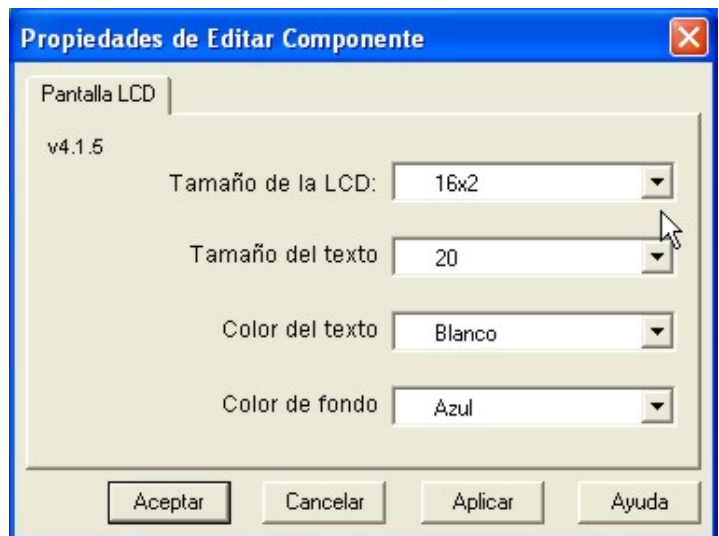
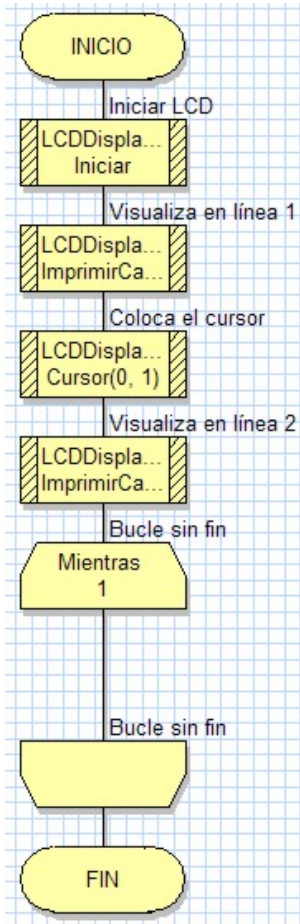


Figura AN12-52. Propiedades de la pantalla LCD

## AN12-6.4.2 El programa



El organigrama de este ejemplo se muestra en la figura AN12-53. Básicamente se emplean las funciones contenidas en la herramienta “Macro de Componentes” relativas a la gestión de la pantalla LCD (LCDDisplay(0)).

La primera función que se ejecuta es “Iniciar()”. Se encarga de configurar los puertos de E/S a los que está conectada la pantalla, según las propiedades de conexiones establecidas. También ejecuta la secuencia de inicialización establecida por su fabricante.

La segunda función es “ImprimirCadena()”. Es capaz de escribir, en la posición actual del cursor, la cadena ASCII indicada (cerrada entre comillas) o el contenido de una variable de cadena. En este caso visualiza la cadena “FlowCode 4: Otra”.

La tercera función, “Cursor()”, coloca el cursor en la posición deseada. En el ejemplo la posición 0 de la línea 1 (1ª posición de la 2ª línea).

La cuarta función es también “ImprimirCadena()”. En esta ocasión imprime, en la posición actual del cursor, la cadena “forma de trabajo”.

El programa entra finalmente en un bucle infinito

Figura AN12-53. El organigrama del ejemplo 4

## AN12-6.4.3 Simulación

Mediante el botón “Ejecutar” procedemos a simular el funcionamiento del programa. En la pantalla LCD situada en el panel de montaje aparecen los dos mensajes indicados anteriormente.

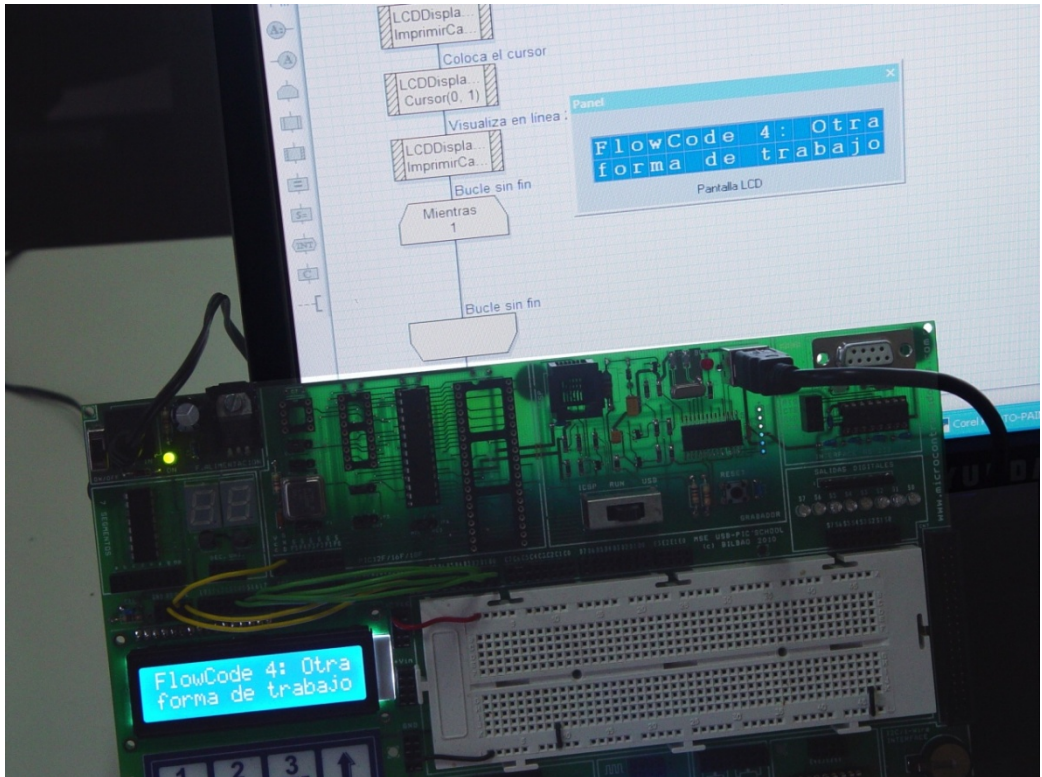
## AN12-6.4.4 Grabación

Si disponemos del laboratorio USB-PIC’School alimentado y conectado a un puerto USB, podemos grabar directamente el microcontrolador con el programa del ejemplo. Basta pulsar el botón “Grabar Chip”. Con esta sencilla acción el FlowCode 4 traduce nuestro programa escrito en lenguaje gráfico, a lenguaje fuente en C (\*.C). De aquí lo pasa a lenguaje ensamblador (\*.ASM), lo convierte en ejecutable (\*.HEX) y lo transfiere a la memoria del PIC.

Sobre el laboratorio realizamos las conexiones eléctricas correspondientes al montaje de este ejemplo. RB0:RB3 se conectan con las patillas L4:L7 de la pantalla. La señal RS de control se conecta con la salida RA3 y la de activación con RA1.



FlowCode 4 no contempla la gestión de la señal R/W que tiene toda pantalla LCD. Es necesario conectar esta patilla a GND. La fotografía de la figura AN12-54 muestra la ejecución del ejemplo sobre el laboratorio.



**Figura AN12-54.** Ejecución del Ejemplo 4 sobre el laboratorio USB-PIC'School

