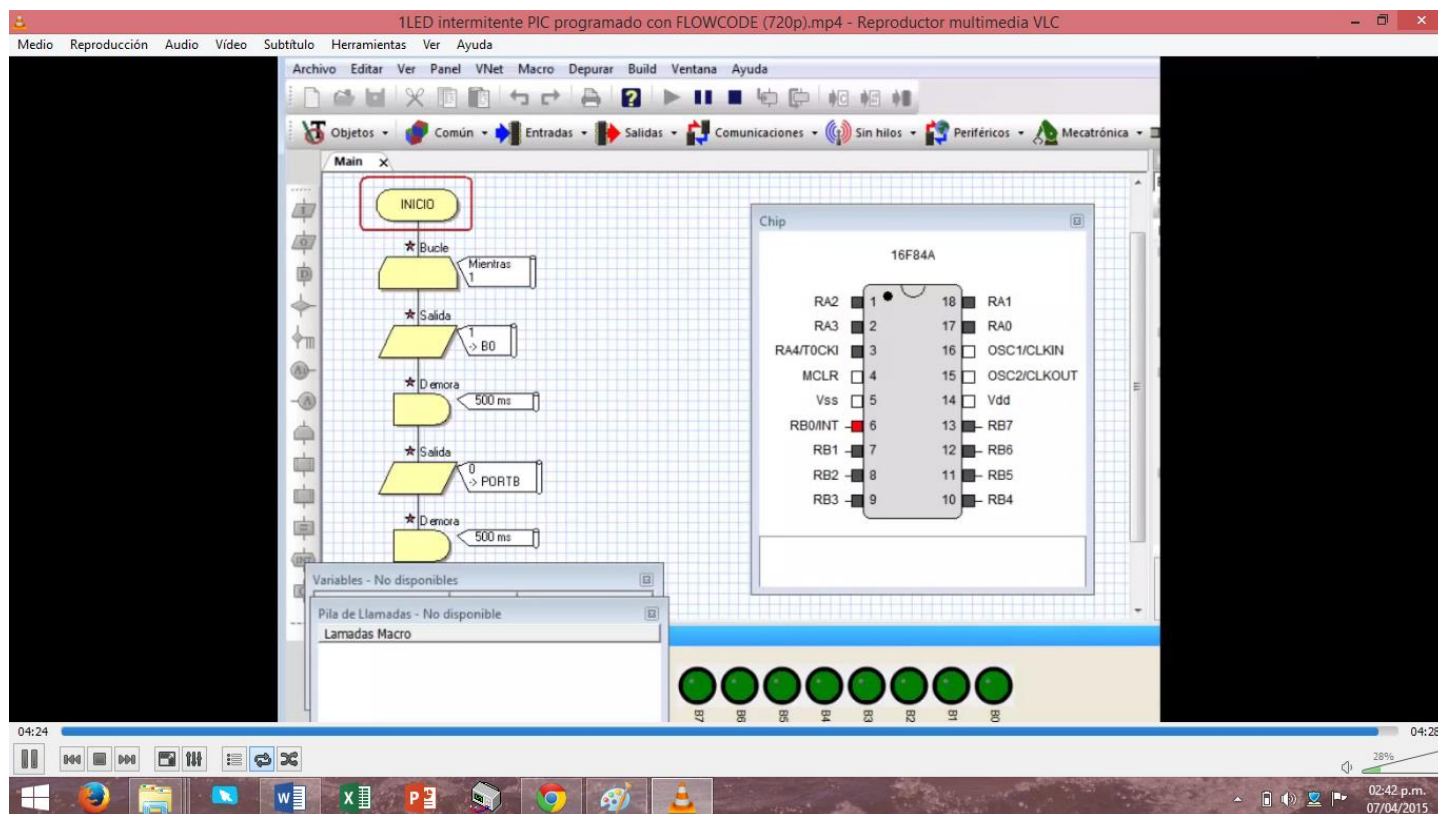


Manual de introducción al uso de FLOWCODE 5 del taller de electrónica 5ºB



Autor: Palacios José Emanuel

Títulos: Técnico Electrónico

Catedra: Taller de electrónica de 5ºB

Fecha: 23/03/2018

Índice

Prologo.....	4
Empezando a programar en FLOWCODE.....	5
El proceso de diseño	5
Diseño de sistemas	6
Flowcode en la educación.....	6
Primera experiencia.....	7
Pantalla principal del programa.....	7
HERRAMIENTAS DE COMANDOS	10
Herramienta Ingreso.....	10
Herramienta de Salida	13
Herramienta de Demora (temporización)	14
Herramienta de Decisión	15
Herramienta de Expresión o decisiones múltiples	16
Herramienta Punto de conexión.....	17
Herramienta Ir a Punto de conexión	17
Herramienta Bucle	18
Herramienta Macro	20
Crear Nueva Macro.....	21
Edición de una Macro	22
El comando Macro del menú principal.....	24
Herramienta Macro de Componentes.....	24
Herramienta de Cálculo	28
Herramienta de Interrupción.....	29
Herramienta Código C.....	30
Código C.....	32
Por ejemplo, para utilizar una variable.....	32
Para llamar una macro.....	32
Para saltar a un punto de conexión	32
Herramienta de Comentarios	33
HERRAMIENTAS DE COMPONENTES	33
Clasificación de los componentes.....	34
Objetos.....	34
Común.....	35
Entradas	35

Salidas	36
Comunicaciones	36
Sin hilos	37
Periféricos	38
Mechatrónica	39
Módulo MIAC	39
Varios	40
Configuración y uso de los componentes	41
Uso del componente	41
Establecer las conexiones del componente	41
Establecer las propiedades externas del componente	42
Simulación	44
Grabación	45
EJEMPLOS DE PROGRAMACIÓN	47
Ejemplo vamos a emplear el componente LCD para visualizar mensajes	47
El montaje eléctrico.	47
El programa	48
Simulación	49
Grabación	49
Ejemplo prender y apagar led con demora	50
Ejemplo de programa armado	51
Bibliografía:	52

Prologo

El objetivo de este documento es proporcionar al alumno información básica necesaria en el Taller de electrónica digital, como, por ejemplo, la forma de conectar y alimentar los circuitos integrados, introduciéndonos a los principios elementales del uso de microcontroladores, etc. Se ha sacrificado la rigurosidad por la sencillez, ya que no se pretende exponer conceptos teóricos excesivos, sino dar una guía rápida que sirva de apoyo al alumno en su primer acercamiento al manejo de los MICROCONTROLADORES. Las explicaciones, cálculos y razones de lo expuesto en este documento se impartirán en las clases de las materias específicas, tanto como las teórico prácticas como el taller de electrónica y están disponibles en la bibliografía de la asignatura.

Todo circuito electrónico o circuito activo necesita energía eléctrica de características precisas para alimentarse. La electrónica digital básica que se presenta sirve como cimiento al microcontrolador como herramienta fundamental, hoy en día, de los diseños de mediano nivel en el ámbito de la electrónica digital. La potencialidad de los microcontroladores está basada no solo en la tecnología actual que permite integrar a costos muy bajos alta capacidad de procesamiento en un solo chip, sino también en la habilidad del programador para incorporar en un circuito integrado funciones de alta potencialidad que permitan al microcontrolador transformarse en un sistema con altísimo valor agregado.

Los pasos a seguir para lograr diseños óptimos con sistemas digitales de bajo costo y alta performance basados en microcontroladores comienzan con el estudio de las características funcionales y estructurales que tiene un dispositivo tal o familia de ellos. Son muchos y muy variados los sistemas microcontroladores que podemos encontrar en el mercado por lo que se debió seleccionar algunos de ellos para realizar un estudio detallado y tratar, con esto, de introducir al estudiante en la materia.

No solo conocer a fondo algunos modelos de microcontroladores resuelve un proyecto. También debe tenerse la suficiente práctica en la confección de software para potenciar lo que sin él sería un circuito integrado inútil. Este texto está dedicado a la descripción detallada de algunos modelos de microcontroladores, con algunos ejemplos de programación desde pequeños retardos de tiempo hasta sistemas algo complejos como contadores u osciladores de precisión que ayudarán a generar una programación más elaborada para aquellos que comienzan a transitar este apasionante tema.

Empezando a programar en FLOWCODE

EL PROGRAMA Flowcode es un potente lenguaje basado en gráficos que emplea macros para facilitar el control de dispositivos complejos como los displays de 7 segmentos, controladores de motor y pantallas LCD.

El uso de macros permite a los estudiantes controlar dispositivos electrónicos altamente complejos sin necesidad de ahogarse en el entendimiento de la programación implicada.

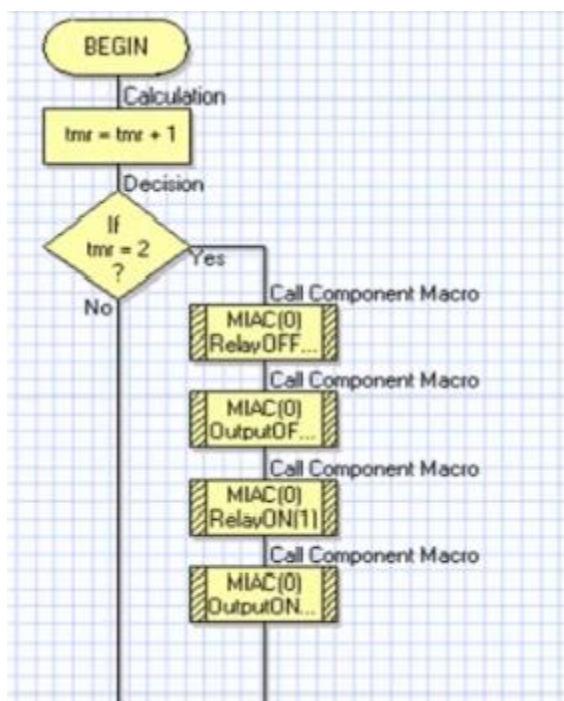
La gran ventaja de Flowcode es que permite a aquellos que tienen poca experiencia crear sistemas electrónicos y robóticos complejos.

El proceso de diseño

Inserta iconos de diagrama de flujo para crear un programa.

Con solo un click en los iconos y componentes puedes cambiar las propiedades.

Cuenta con componentes que van desde un simple led hasta componentes de BLUETOOTH.

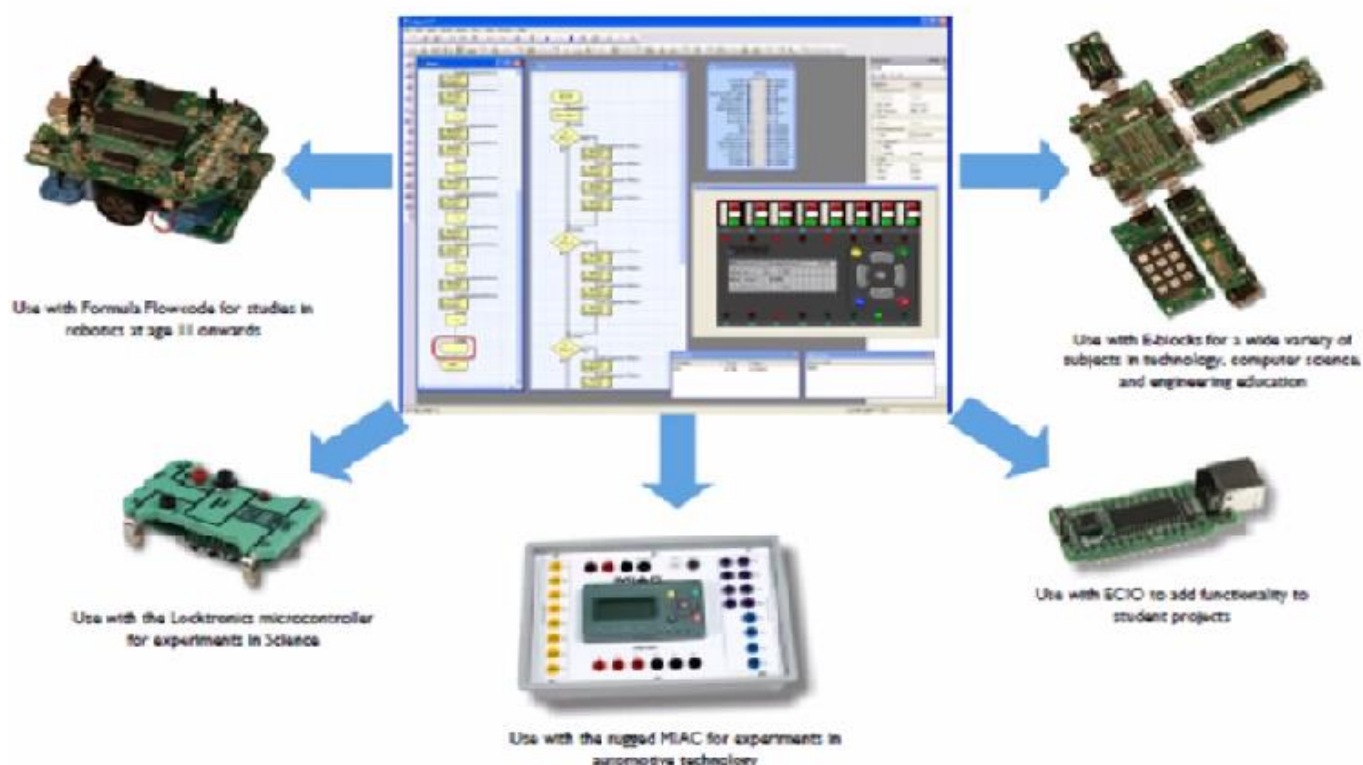


Diseño de sistemas

Cuenta con un panel para colocar componentes y ajustarlos de manera gráfica.

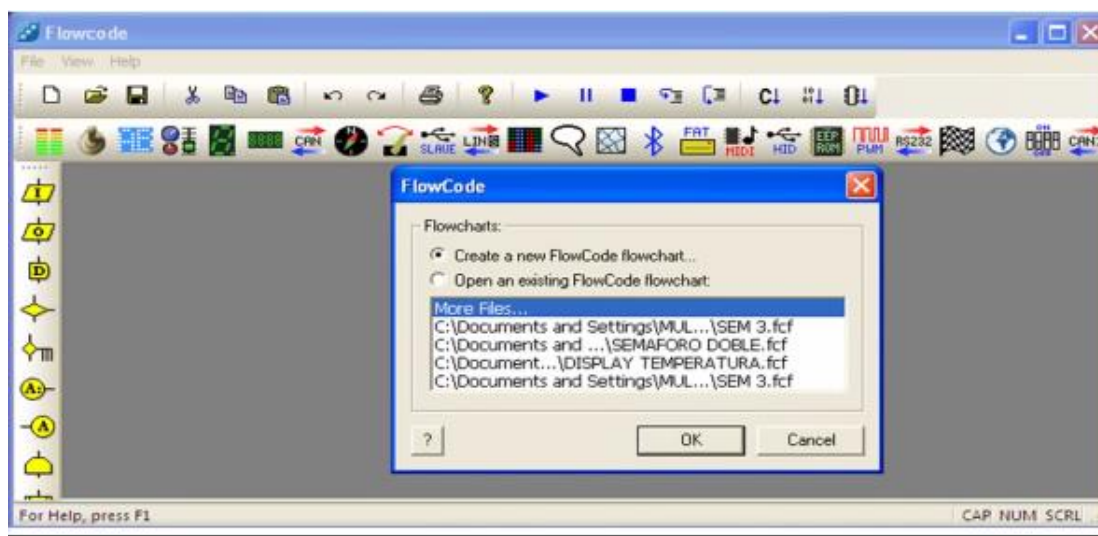
Flowcode en la educación

Flowcode es reconocido internacionalmente como un líder en el desarrollo de microcontroladores para la educación. Es usado en más de 700 escuelas, colegios y universidades a nivel mundial.

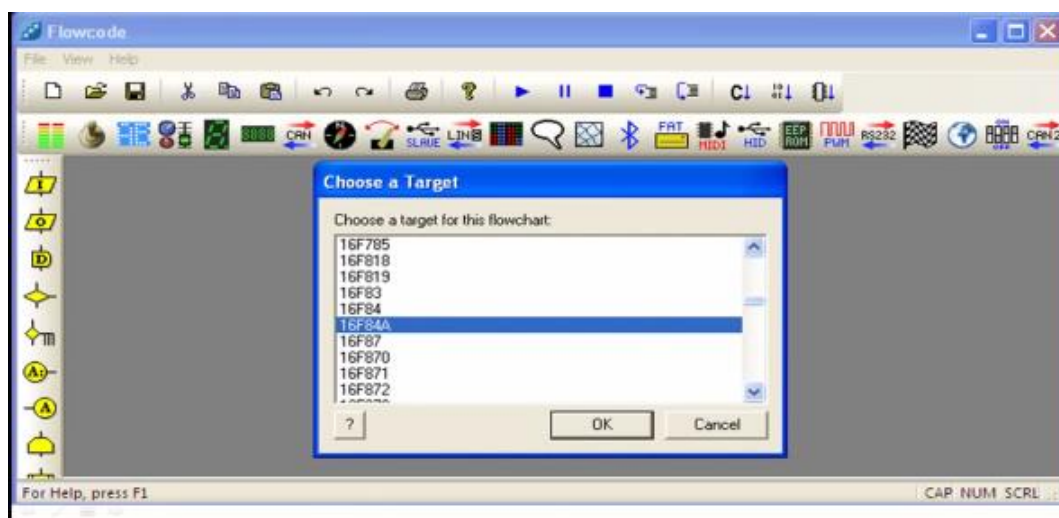


Primera experiencia

Paso 1: abrir FLOWCODE y seleccionar **CREAR UN DIAGRAMA DE FLUJO**



Paso 2: seleccionar el **TIPO DE PIC** de acuerdo a su familia.



Pantalla principal del programa

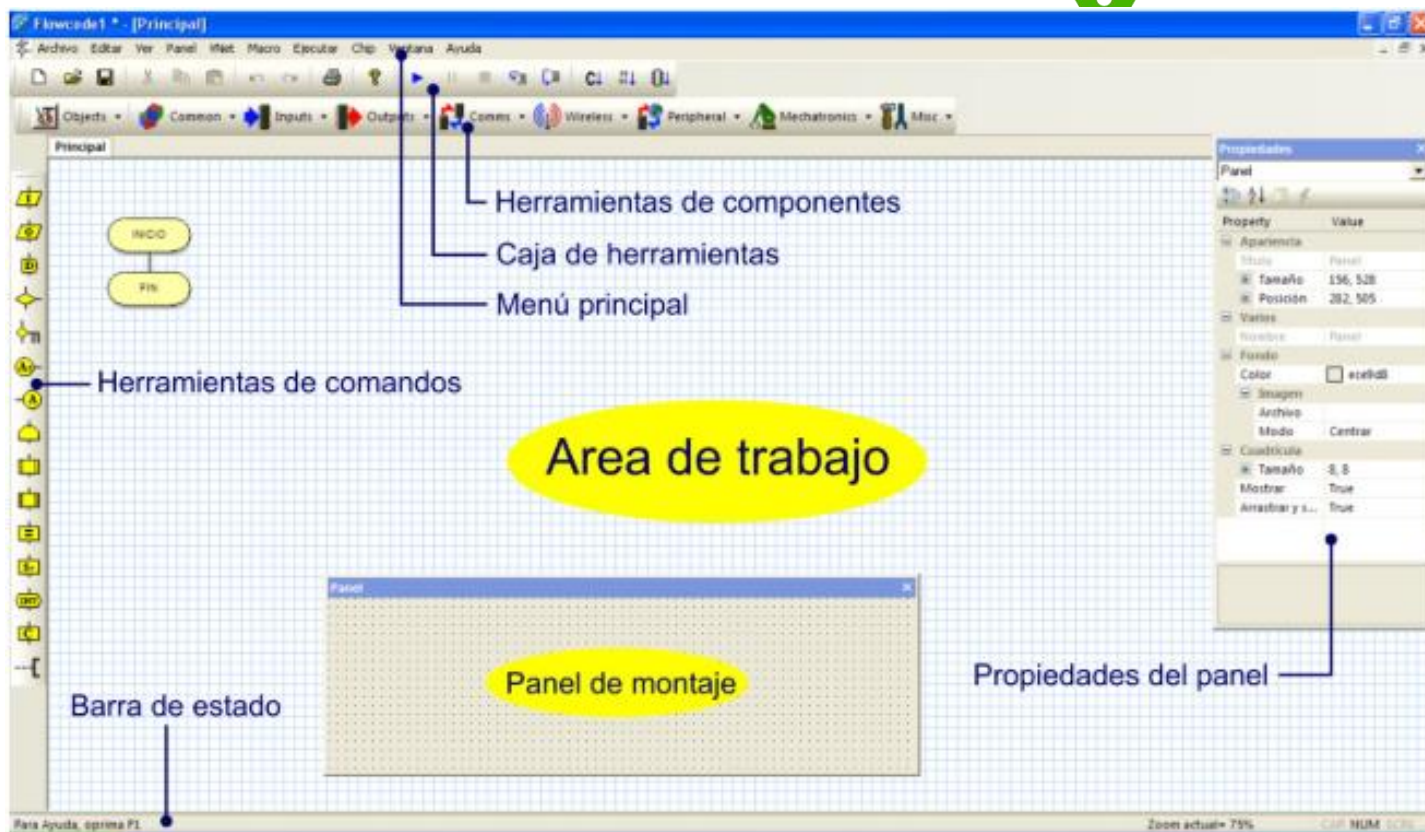
Para ejecutar el FlowCode basta con hacer click en su correspondiente acceso directo del escritorio de FlowCode

El área se divide en diferentes secciones. Cada una de ellas se puede visualizar o no en la pantalla.

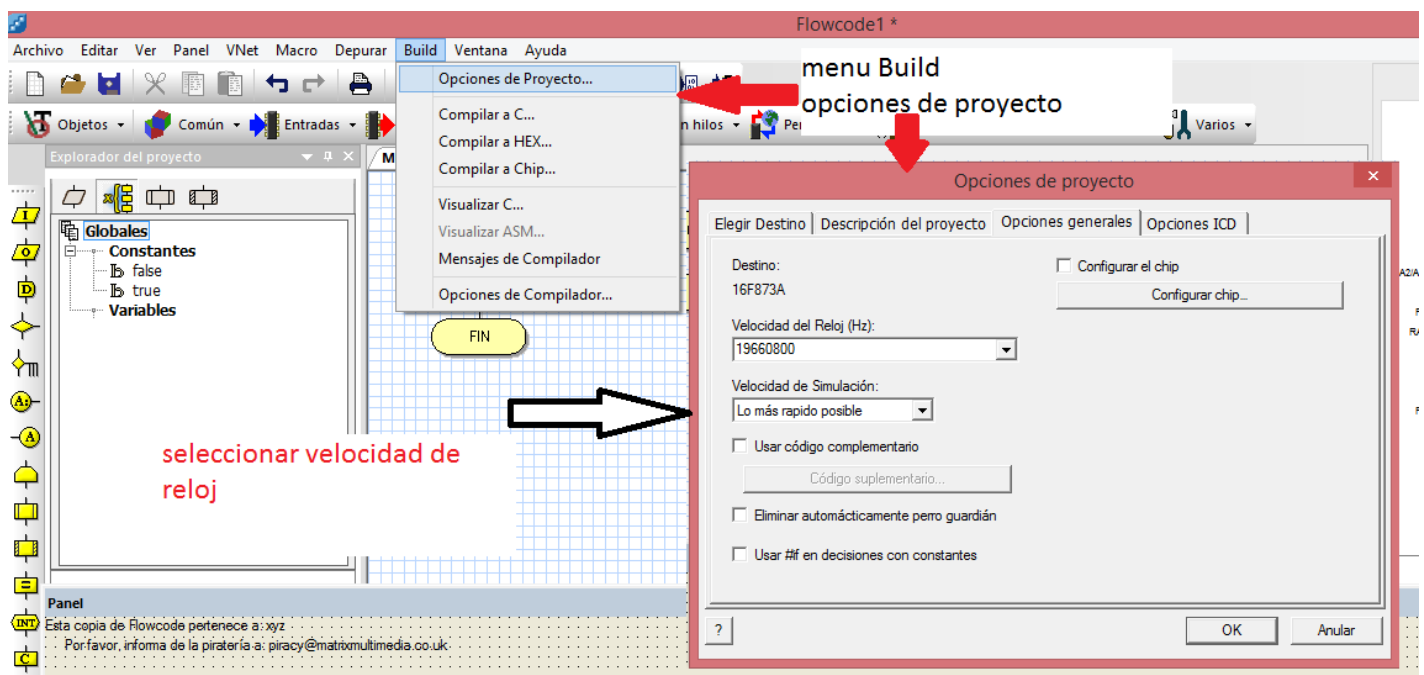
- **Menú principal:** Aparece en la línea superior de la pantalla. Contiene las diferentes opciones de trabajo del programa FowCode. Algunas de estas opciones, como “Archivo”, “Editar” y “Ver” son comunes a otras aplicaciones Windows. El resto son

opciones propias del programa y algunas de ellas las comentaremos en este documento.

- 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. 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.



Paso 3: configurar el **CHIP**



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 click 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).

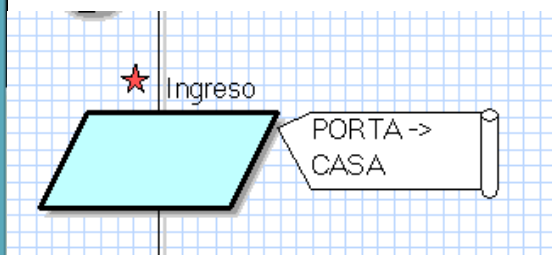
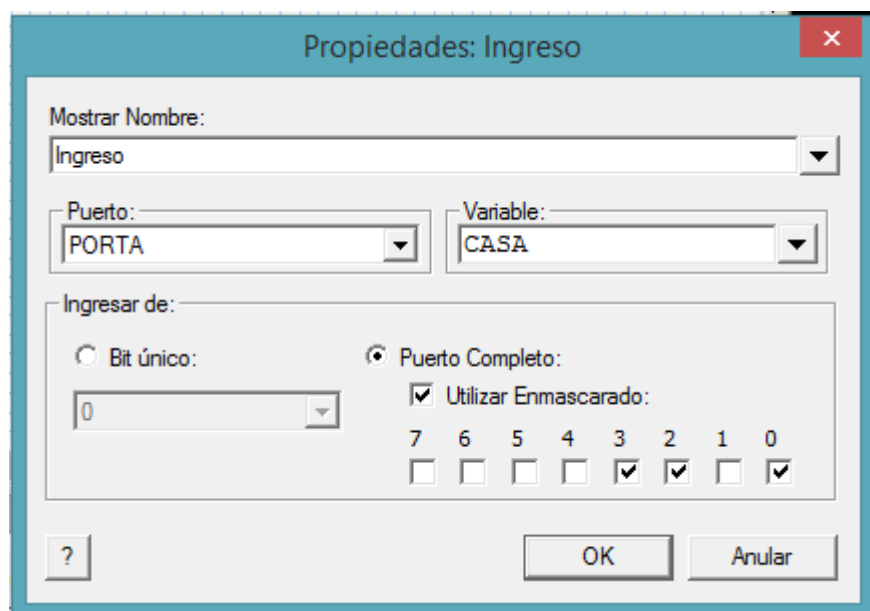
Herramienta Ingreso

Permite ejecutar tareas relacionadas con la adquisición de datos procedentes de los Puertos de entrada del controlador. La figura siguiente muestra su icono así como la ventana de propiedades que se abre, cada vez que hacemos doble click sobre él. Veamos para que sirve cada campo:

- **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.

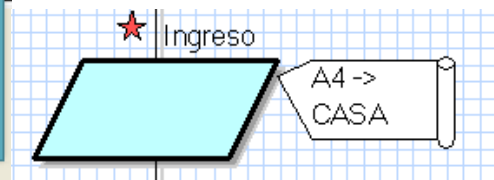
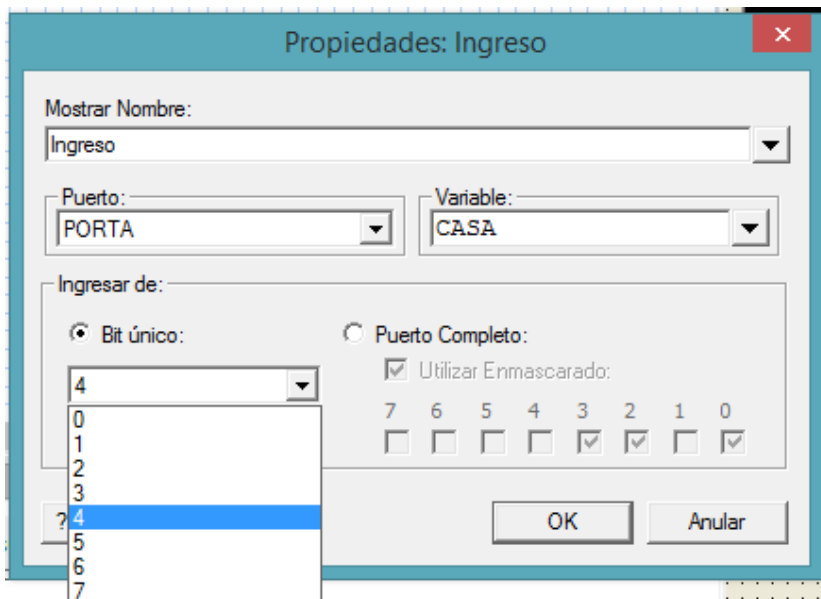
Podemos pulsar el botón “Variables...” que nos abrirá una nueva ventana donde podremos crear, editar y borrar variables de 8 bits, de 16, de cadenas ASCII o strings y variables para almacenar números en coma flotante.

- **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.

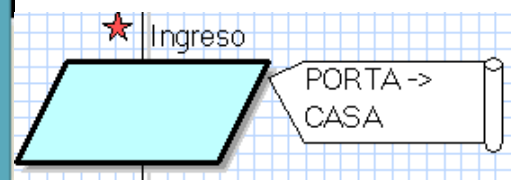


- **En el campo “Ingresar de”:** La lectura de un Puerto de entrada puede hacerse a nivel de un **bit único** o a **nivel del puerto completo**. En este caso podemos hacer además una lectura **enmascarada**. Es decir, podemos indicar qué bits de ése puerto son los que nos interesa leer. En el ejemplo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

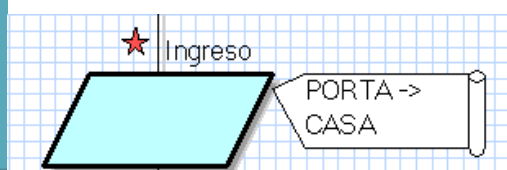
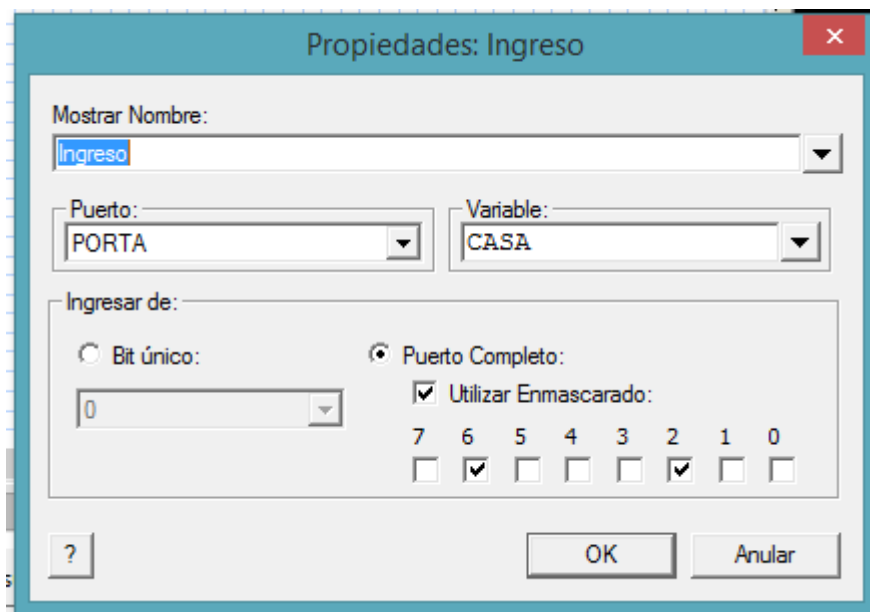
- ✓ **"Bit único":** elijo que bit del puerto le voy a leer, en el ejemplo siguiente se muestra la elección de lectura del **puerto A , solo del bit 4 y guardar en la variable CASA.**



- ✓ **"Puerto completo":** elijo que voy a leer el puerto completo, en el ejemplo siguiente se muestra la elección de lectura del **puerto A , puerto completo y guardar en la variable CASA**

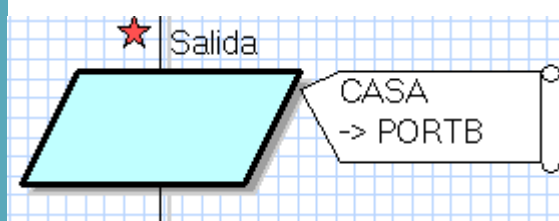
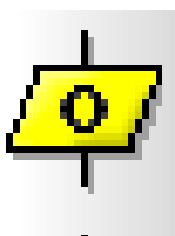


- ✓ **"Utilizar Enmascarado":** elijo que voy a leer el puerto y que bit del puerto voy a leer, en el ejemplo siguiente se muestra la elección de lectura del **puerto A , puerto de los bit 2 y 6 ; guardar en la variable CASA**

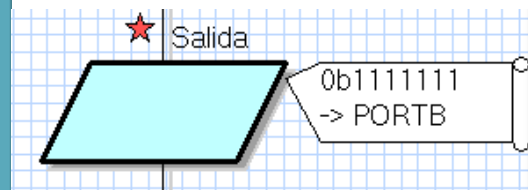
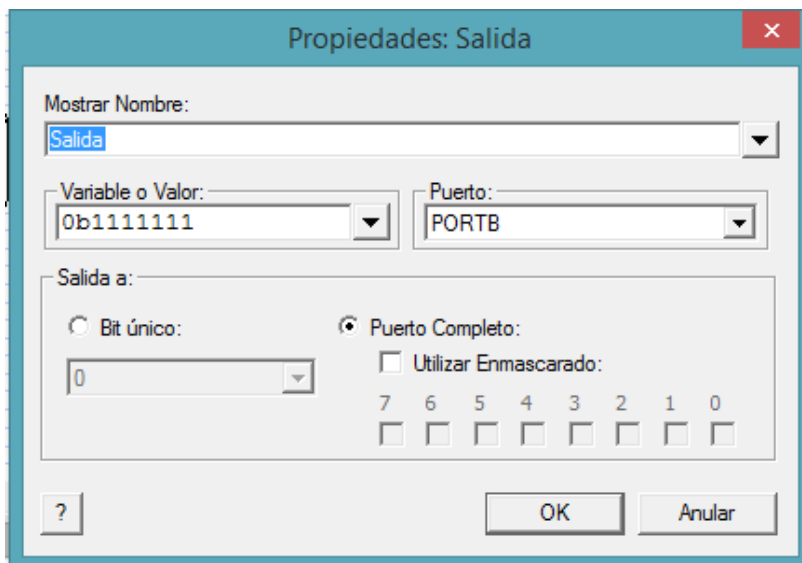


Herramienta de Salida

Permite ejecutar operaciones relacionadas con los Puertos de salida del controlador. En la figura siguiente 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 la imagen siguiente sacamos el valor de la variable "CASA". En el campo "Puerto" seleccionamos el puerto de salida, Puerto B en el ejemplo.



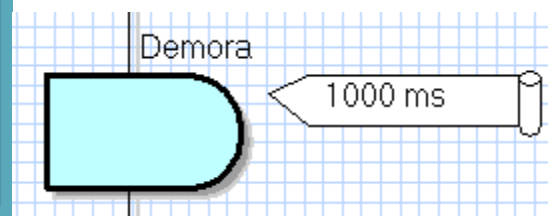
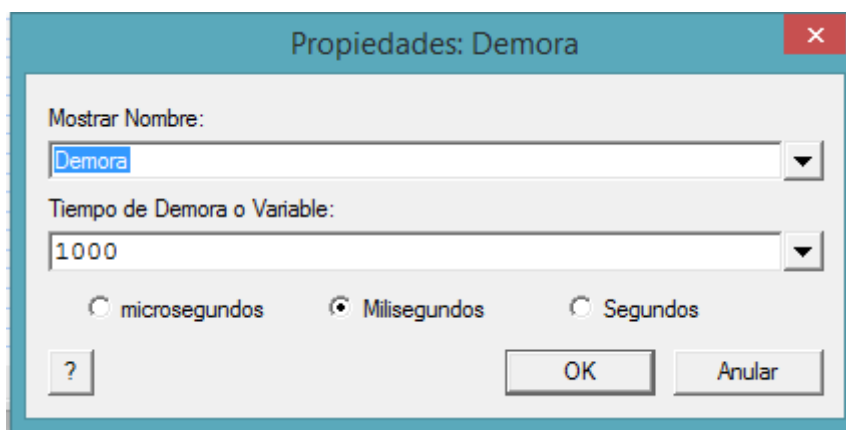
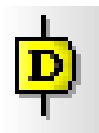
La escritura sobre un Puerto de salida también se puede realizar a nivel de bit único o a nivel de Puerto completo (como en el icono entrada). En este caso la escritura puede ser o no enmascarada.



En la imagen anterior se observa que se envía al puerto B el valor **0b1111111** y se usa el **puerto completo**.

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" Æ "Opciones de Proyecto"). Su ventana de propiedades se muestra en la figura. 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, 1000 en el ejemplo, o bien indicando el nombre de la variable que lo contiene.



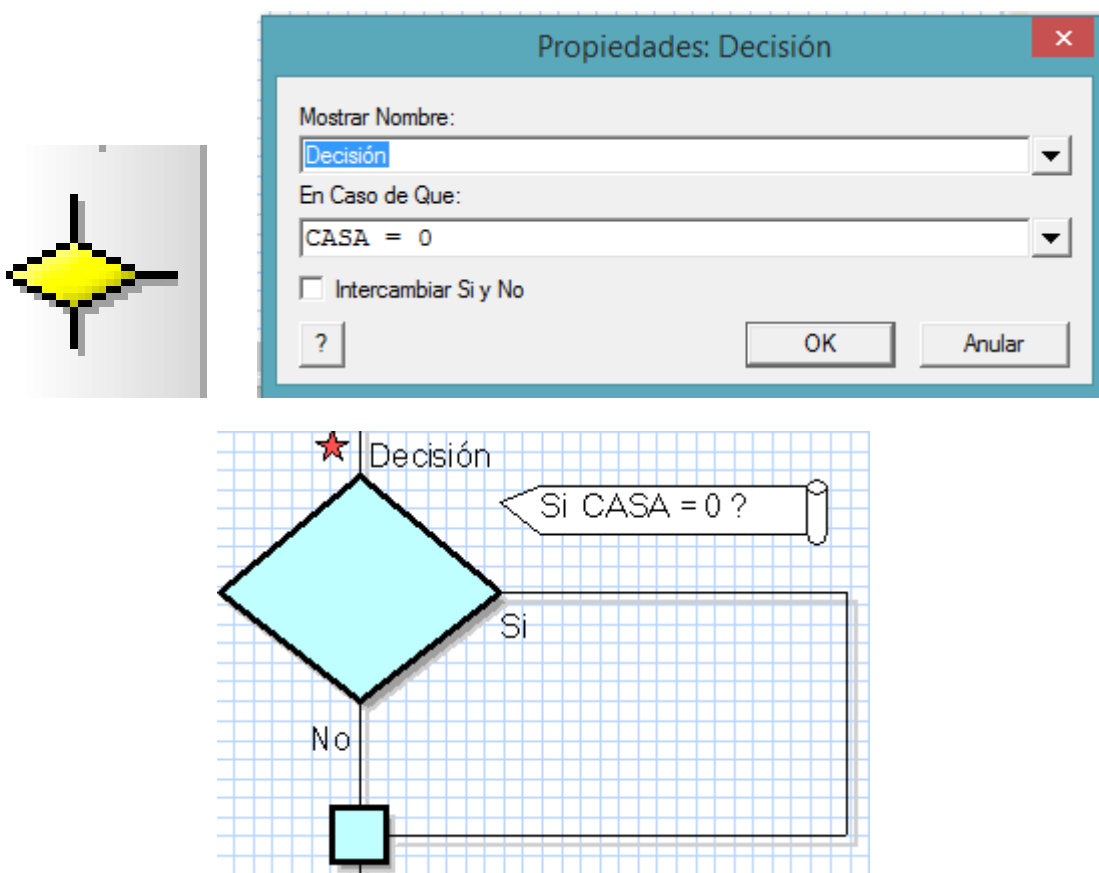
Esta ventana de propiedades también nos permite seleccionar las unidades en que se representa el tiempo: Microsegundos, Milisegundos o Segundos.

En el ejemplo se realiza una temporización de 1000 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 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.



En la figura anterior la condición es si “CASA=0 “ , si se cumple la condición el programa sigue por el lugar donde se indica el “SI” , en caso contrario el programa sigue por el lugar que dice “NO”.

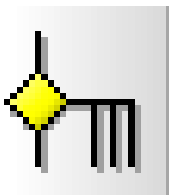
Las condiciones pueden estar formadas por números y/o variables y uno de los siguientes operadores:

()	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

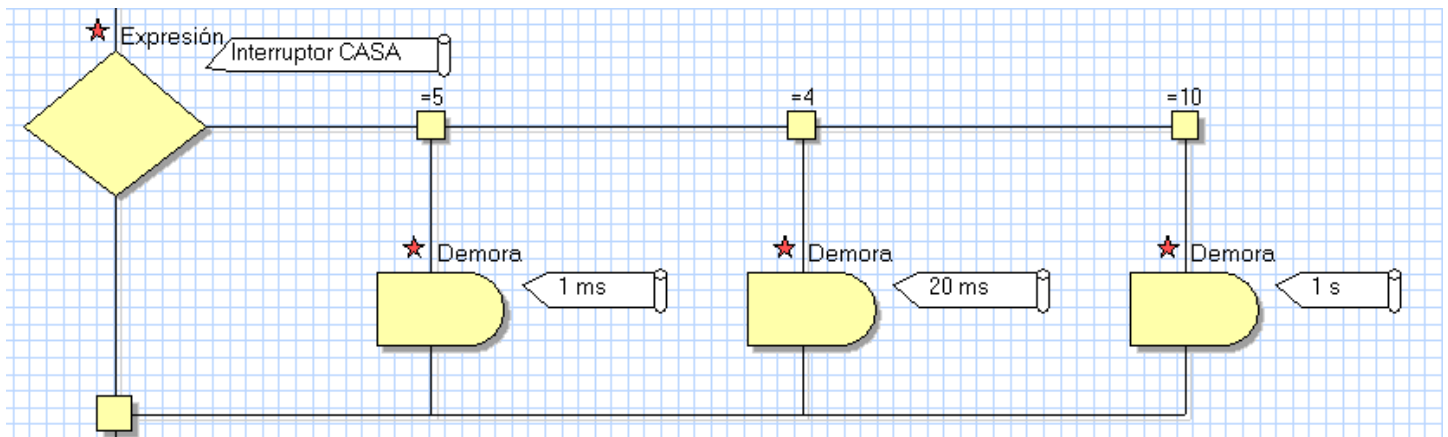
Herramienta de Expresión o decisiones múltiples

La figura 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 (SI 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.



En el campo “Expresión” se indica el nombre de la variable cuyo contenido se desea evaluar, “CASA” en el ejemplo. Los campos “Casos”, cuando se activan, permiten determinar con qué valores hay que comparar el contenido de la variable “CASA”: 5, 4, 10. 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 siguiente

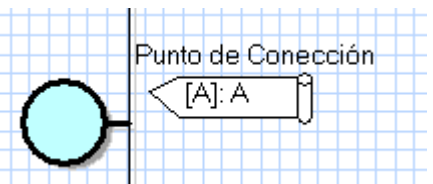
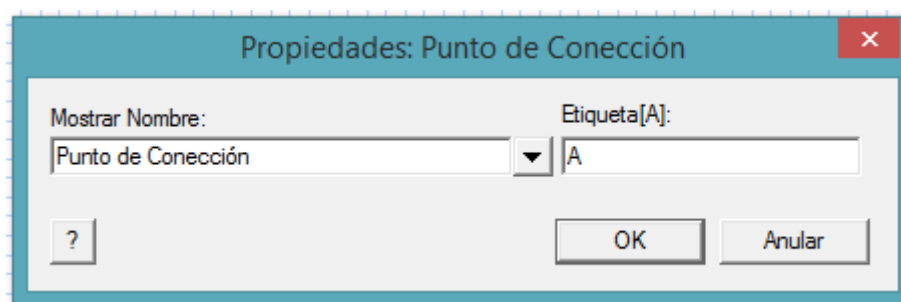
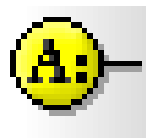


Se evalúa el valor de la variable “Entradas”. Si es igual a 5, se temporiza 1 milisegundo; si es 4 se temporiza 20 milisegundos; si es 10 se temporiza 1 segundos. Si no coincide con ninguno de esos tres valores, el programa sigue su curso sin hacer temporización alguna.

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 siguiente, permite establecer un punto de conexión de destino.

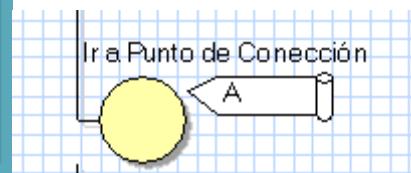
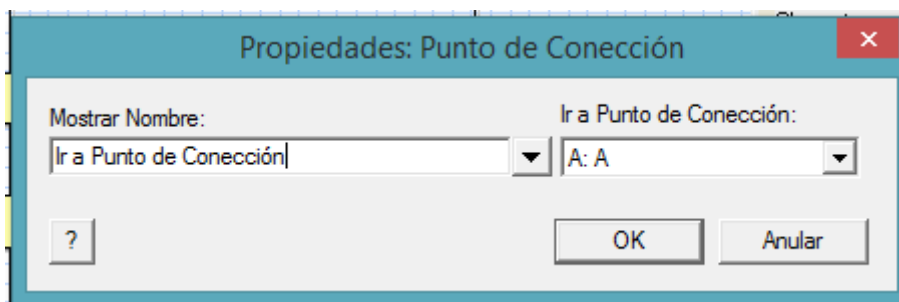
En el campo “Etiqueta” asignamos un nombre o etiqueta al punto de destino. En el ejemplo hemos asignado la etiqueta “A”.



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 siguiente.

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



La figura siguiente 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.

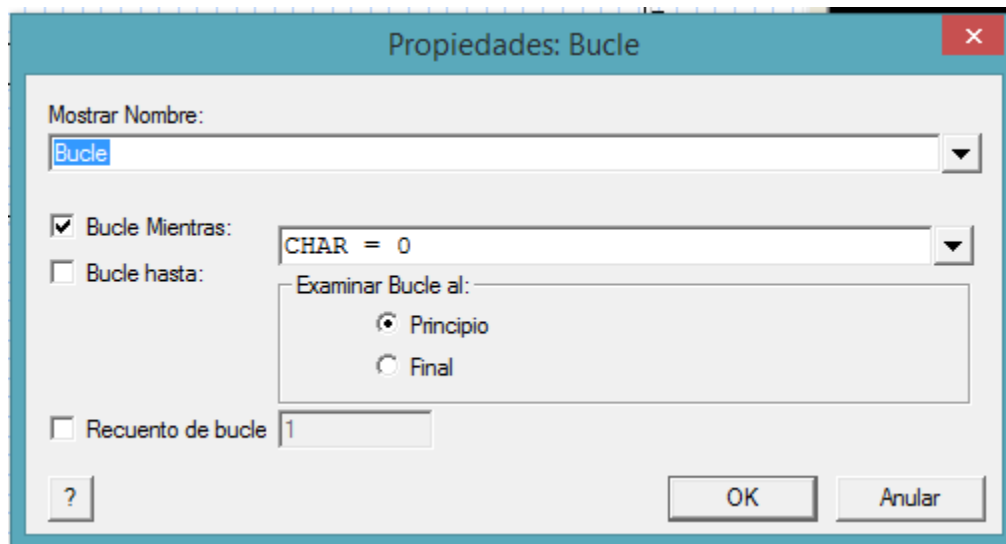
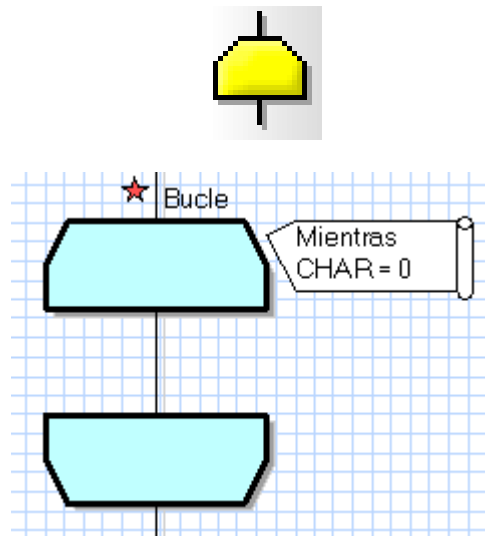


El programa finaliza con la herramienta “Ir a Punto de Conexión” llamado “Bucle”, lo que origina que el flujo se desvíe al punto de conexión superior. Como consecuencia de todo ello, tenemos un programa que lee el valor de las entradas del Puerto A y lo deposita sobre el Puerto B de salida. Esto se repite 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.

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 siguiente 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 “CHAR” 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).

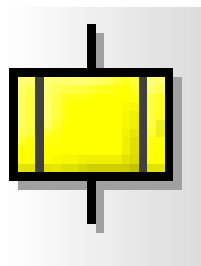
Indicar que también se puede crear un bucle en el que sus comandos se ejecuten de forma indefinida e incondicional. Basta con activar “Bucle Mientras” e introducir el valor 1 en el campo de condiciones. En el ejemplo de la figura siguiente vemos un bucle que consiste en

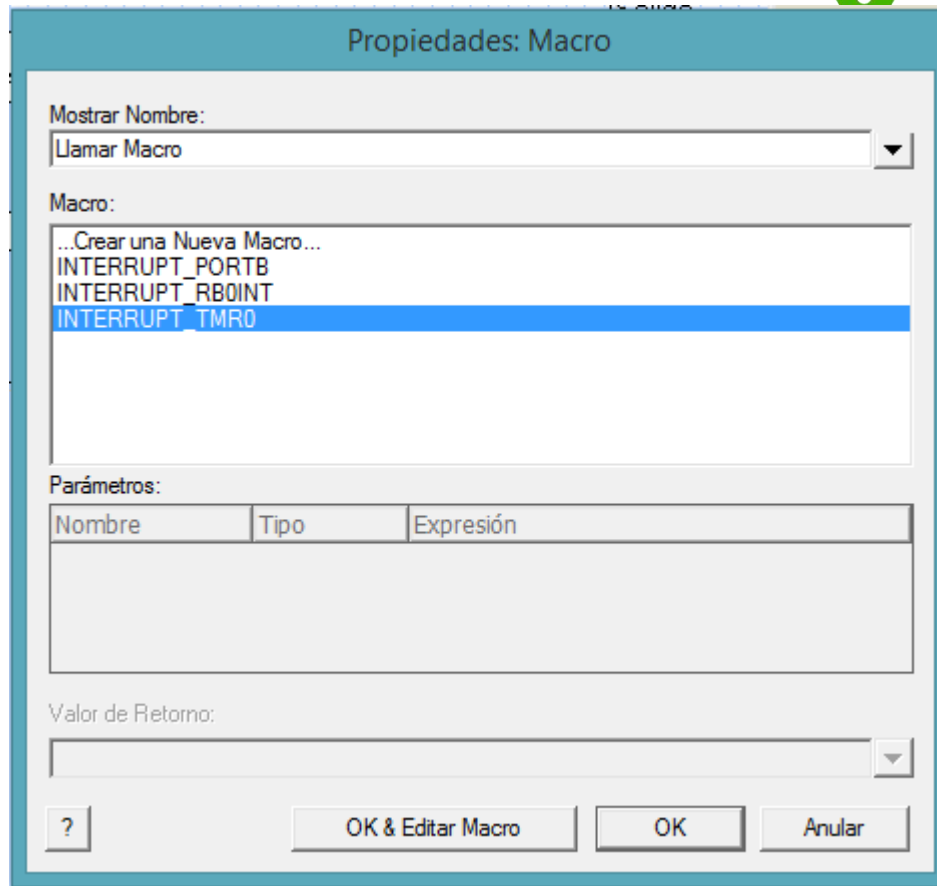
leer las entradas del Puerto A y reflejarlos en las salidas del Puerto B. El bucle se ejecuta hasta que las entradas valgan 0.



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 sea necesario, sin tener que reescribir los comandos que la componen. La figura siguiente muestra la ventana de propiedades del comando Macro.





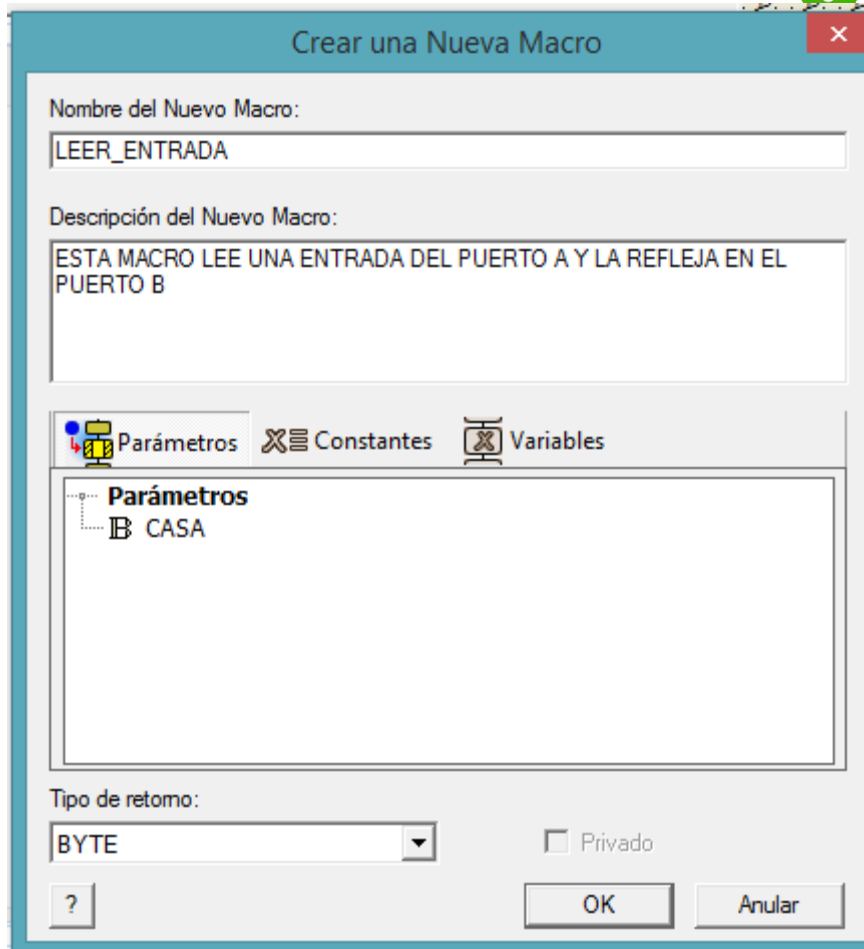
En el campo “Mostrar Nombre” podemos escribir un comentario. En el campo “Macro” podemos ver una lista con todas las macros disponibles. Seleccionamos la que necesitamos en un momento dado.

En el campo Parámetros nos muestra las variables o parámetros que usa o devuelve la macro a usar.

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 siguiente. En el campo “Nombre del Nuevo Macro” indicamos el nombre que le queremos asignar a esa Macro, por ejemplo, “LEER_ENTRADA”.



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

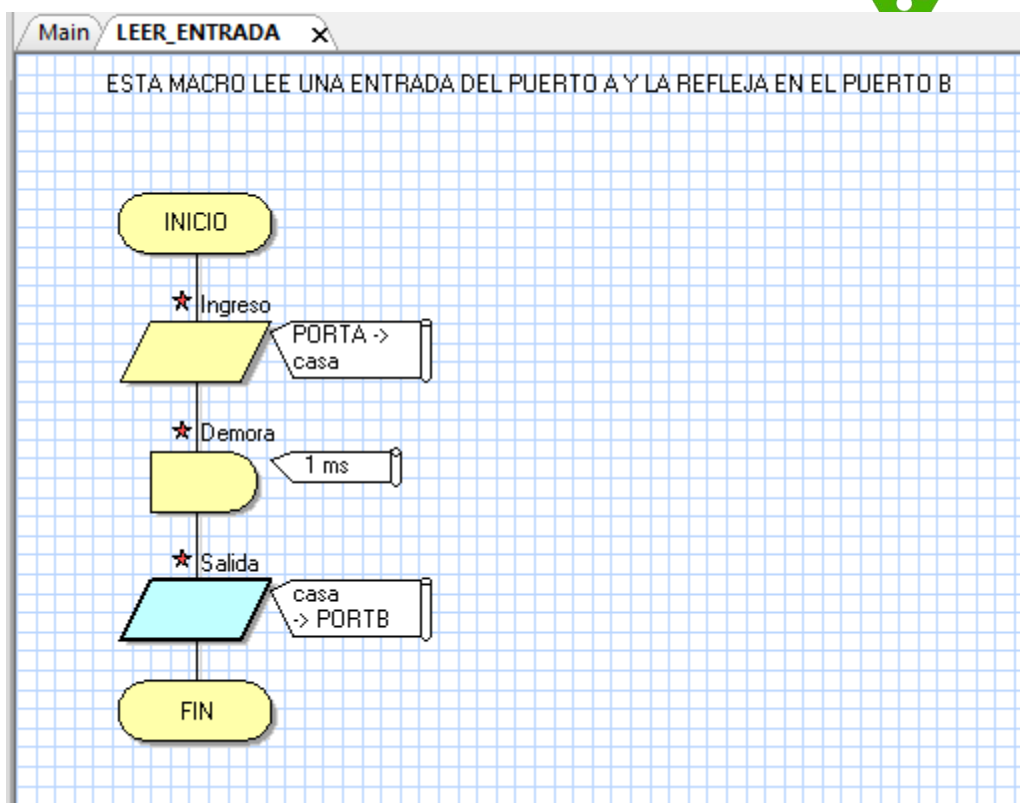
En el campo “Parámetros” podemos definir datos o variables de entrada que nuestra Macro pueda necesitar.

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.

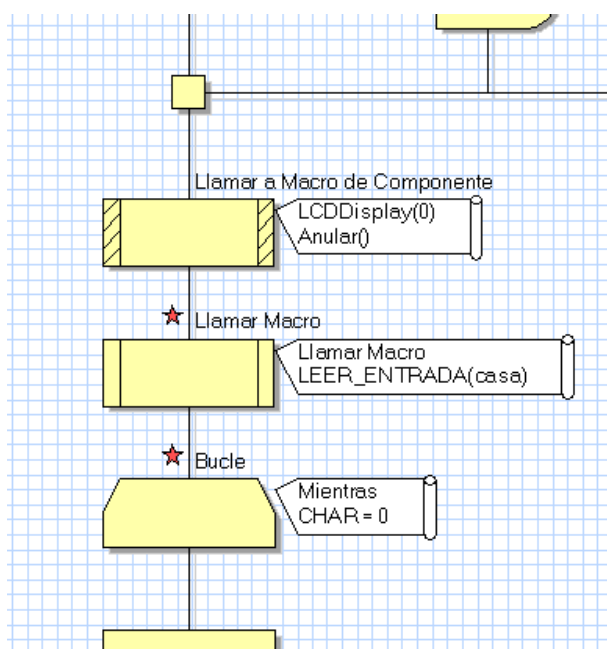
Edición de una Macro

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 siguiente se muestra un ejemplo de la Macro que hemos llamado “LEER_ENTRADA”.



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 “LEER_ENTRADA”. Empieza leyendo el puerto A y lo guarda en la variable “casa”, demora 1ms y luego genera una salida en la que le envía al Puerto B la variable “casa”.

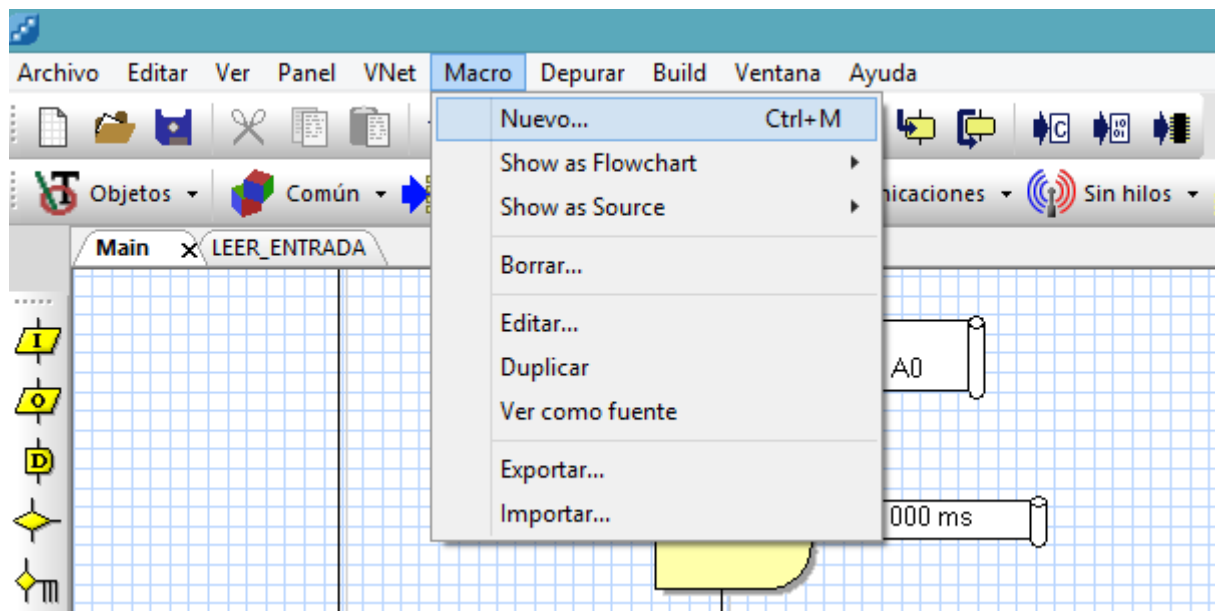
En la figura siguiente se muestra cómo quedo el programa principal.



El comando Macro del menú principal

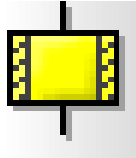
Este comando nos abre una serie de opciones relacionadas con el empleo de las Macros:

- **Nuevo.** Nos permite definir una nueva Macro.
- **Mostrar.** Nos presenta un listado de todas las Macros que hubiera definidas. Haciendo click 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 utilizadas en múltiples proyectos o aplicaciones.
- **Importar.** Esta opción permite leer y recuperar desde un archivo en disco las distintas Macros que hubiera almacenadas.

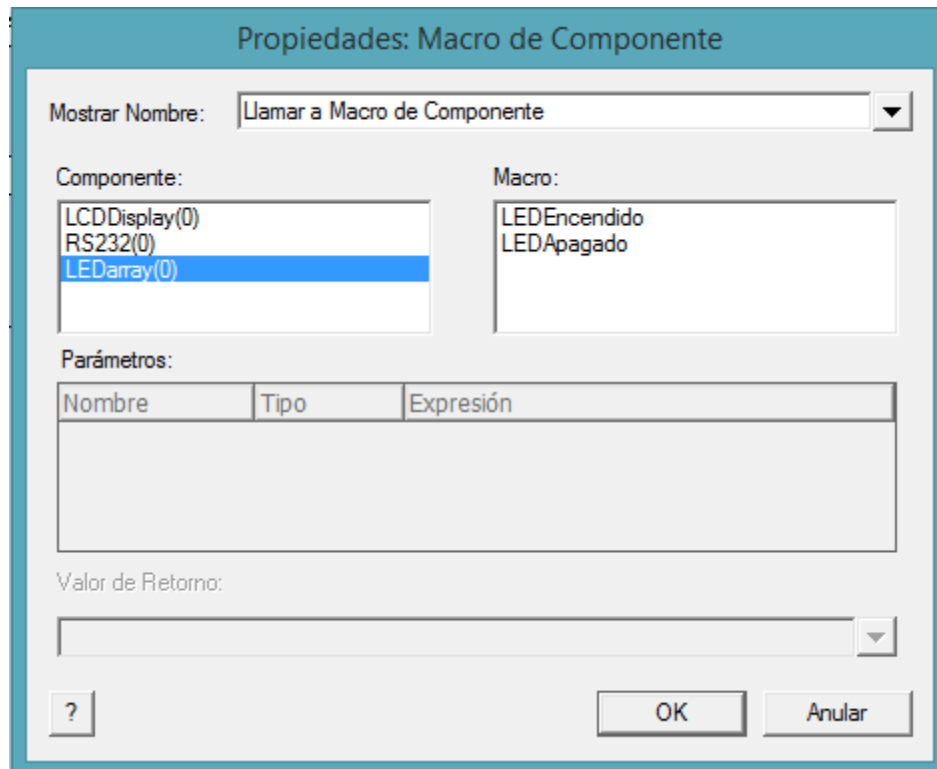


Herramienta Macro de Componentes

FlowCode 5 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.

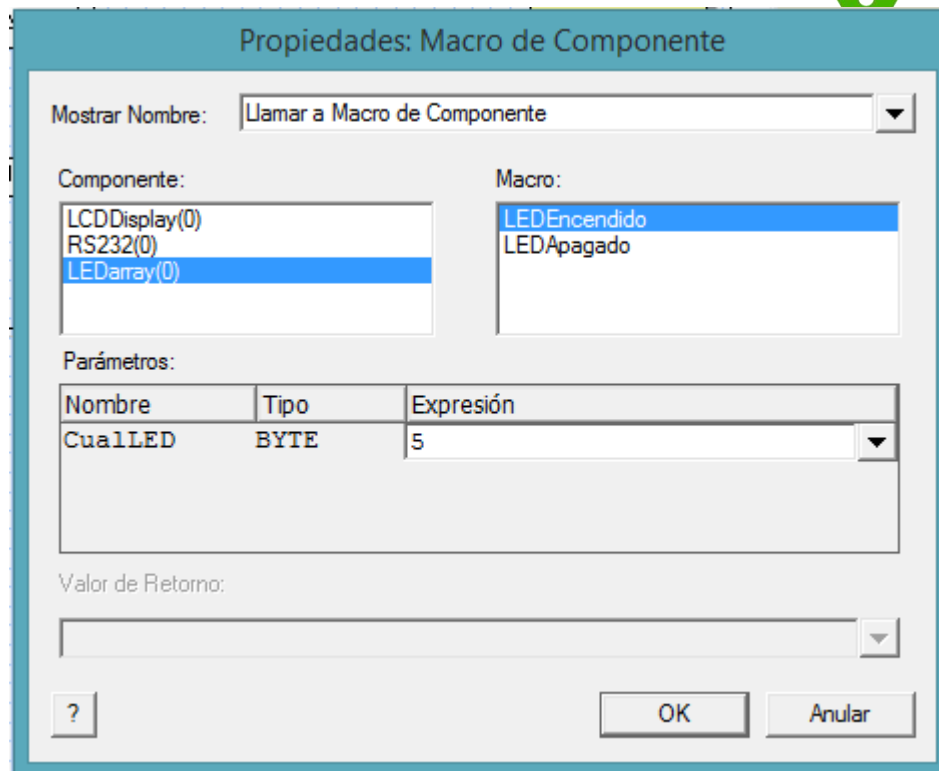


Cuando empleamos la herramienta “Macro de Componentes” y hacemos doble click sobre ella, aparece una ventana de propiedades como la mostrada en la figura siguiente. En ella disponemos de un listado con todos los componentes que hubiera en ese momento en el panel de montaje y accedemos a todas las funciones asociadas a los mismos.



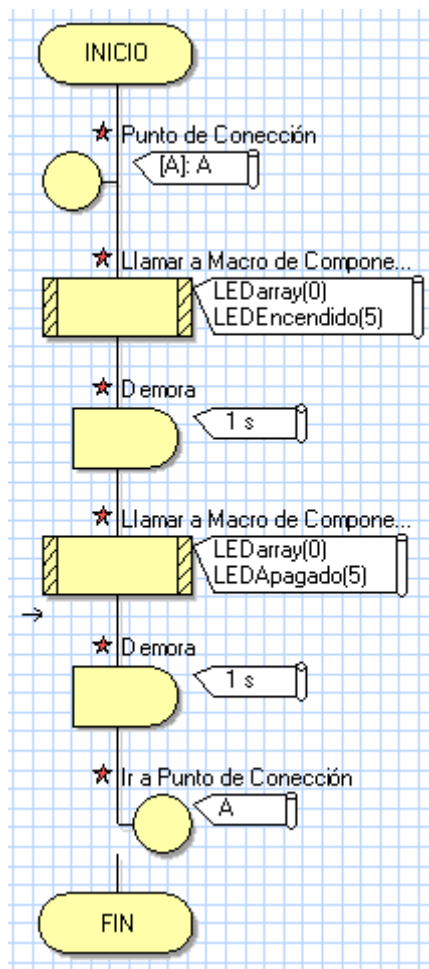
En la figura, vemos que hay tres componentes: un LCDDisplay(0), un RS232(0) y un LEDarray(0). Al seleccionar por ejemplo, el LEDarray(0), a la derecha, en la ventana “Macro”, aparecen las funciones a él asociadas. Hay dos:

1. LEDEncendido: Manda un 1 al led que quiero que se ilumine.
2. LEDApagado: Manda un 0 al led que quiero que se apague.



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

1. LEDEncendido. La línea de salida donde se encuentra conectado que indicamos en el parámetro se activa, en este caso el led "5".
2. LEDApagado. La línea de salida donde se encuentra conectado que indicamos en el parámetro se desactiva.



En la figura anterior 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, una tira led, el LEDarray(0). El componente se supone previamente dibujado y definido en el panel de montaje.

Primero se ejecuta el componente punto de conexión “A”.

Segundo se ejecuta la función “LEDEncendido” para el led 5 contenida en la Macro de componentes LED(0). La línea de salida RA5 se pone a nivel “1” y el led se enciende.

Tercero se ejecuta una demora de 1 segundo.

Cuarto se ejecuta la función “LEDApagado” para el led 5 contenida en la Macro de componentes LED(0). La línea de salida RA5 se pone a nivel “0” y el led se apaga.

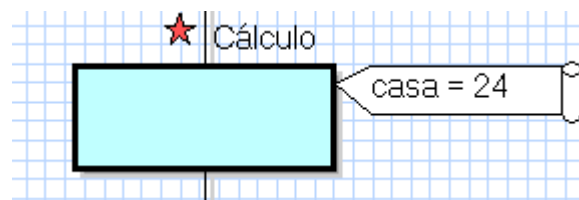
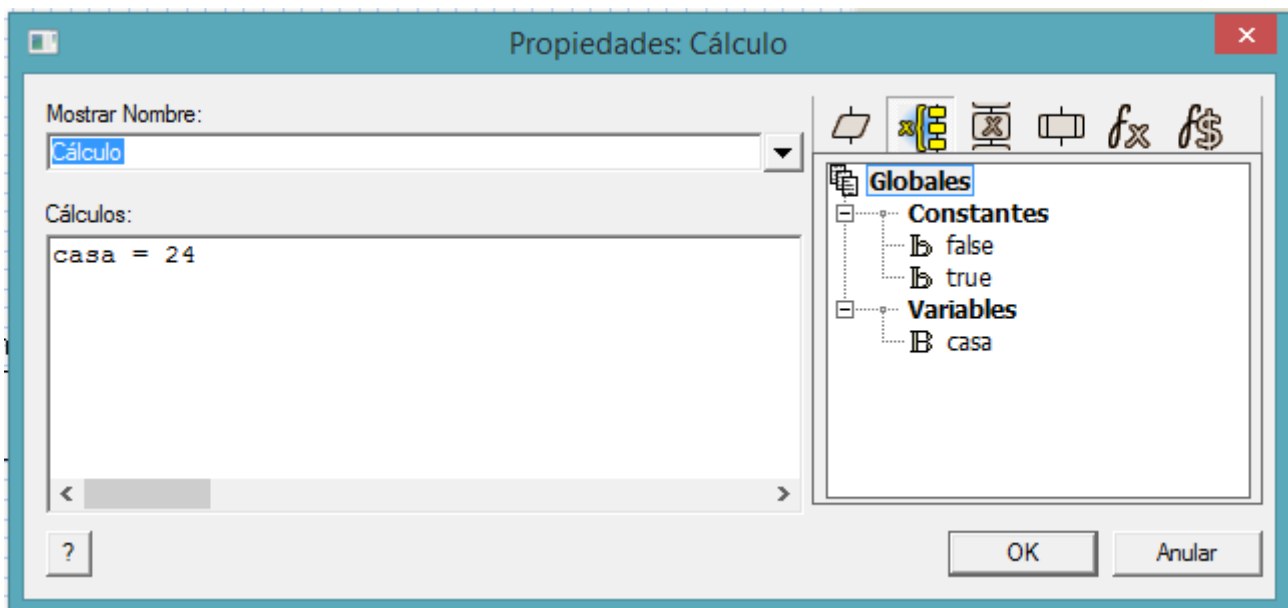
Quinto se ejecuta una demora de 1 segundo.

Sexto se ejecuta el componente ir a punto de conexión “A”.

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

Herramienta de Cálculo

Otra herramienta esencial en FlowCode 5 es la llamada “Herramienta de Cálculo”. Permite insertar operaciones lógico/aritméticas en nuestros programas. La ventana de propiedades se muestra en la figura siguiente.



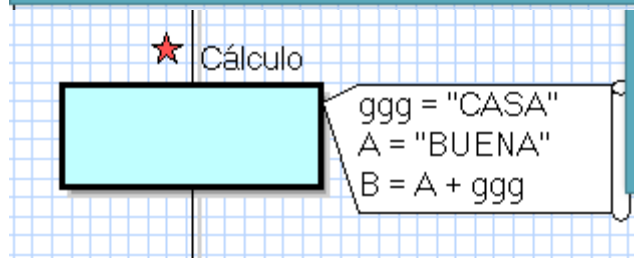
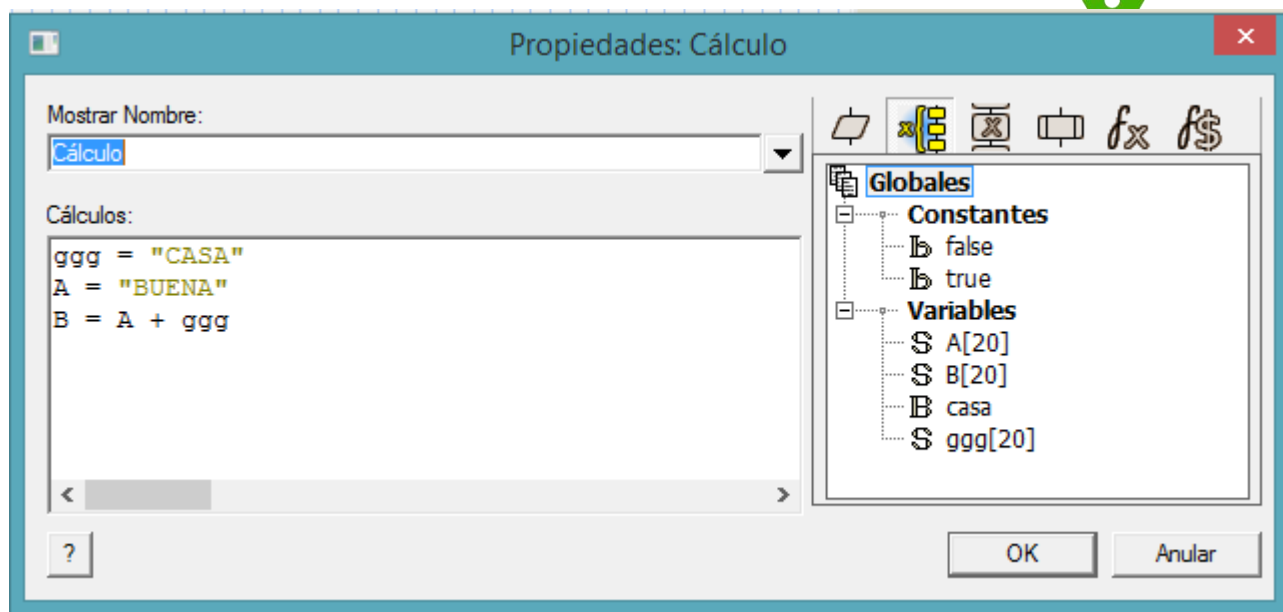
En el campo “Mostrar Nombre” podemos escribir el comentario que deseemos. En el campo “Cálculos” describimos los cálculos a realizar.

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.

En el ejemplo, la variable “casa” se carga con 24.

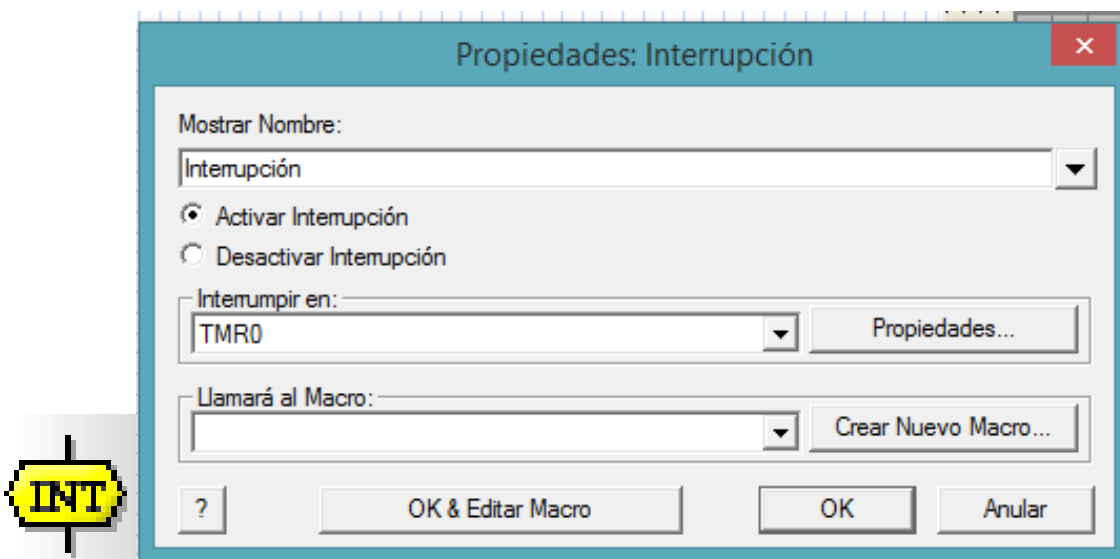
También permite utilizar funciones de manipulación de secuencia, permite manipular y operar con cadenas de caracteres ASCII.

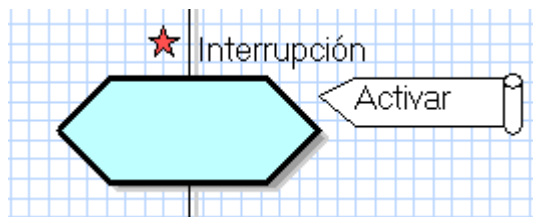


En el ejemplo de la figura, la variable “ggg” se carga con la cadena “CASA” y la variable “A” con la cadena “BUENA”. La variable “B” se carga con el resultado de concatenar las dos cadenas anteriores, por lo que su contenido será “BUENA CASA”.

Herramienta de Interrupción

FlowCode 5 también soporta la activación/desactivación de algunas de las interrupciones más relevantes del microcontrolador seleccionado. En la figura siguiente se muestra la ventana de propiedades de esta herramienta.





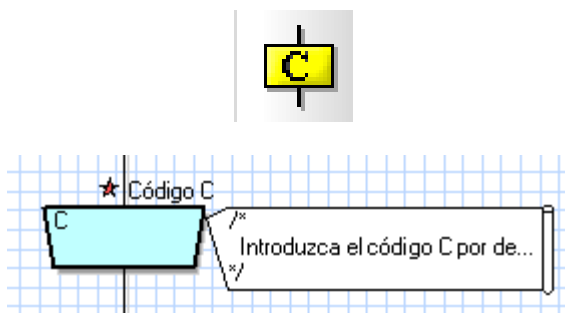
En el campo “Interrumpir en” seleccionamos la fuente de interrupción, por ejemplo la interrupción externa TMR0. Aunque no están disponibles todas las que tiene el controlador, si están las más representativas. Con el botón “Propiedades” se selecciona el evento deseado para esa interrupción. Por ejemplo, en el caso de la interrupción externa INT, podemos seleccionar el flanco ascendente o descendente.

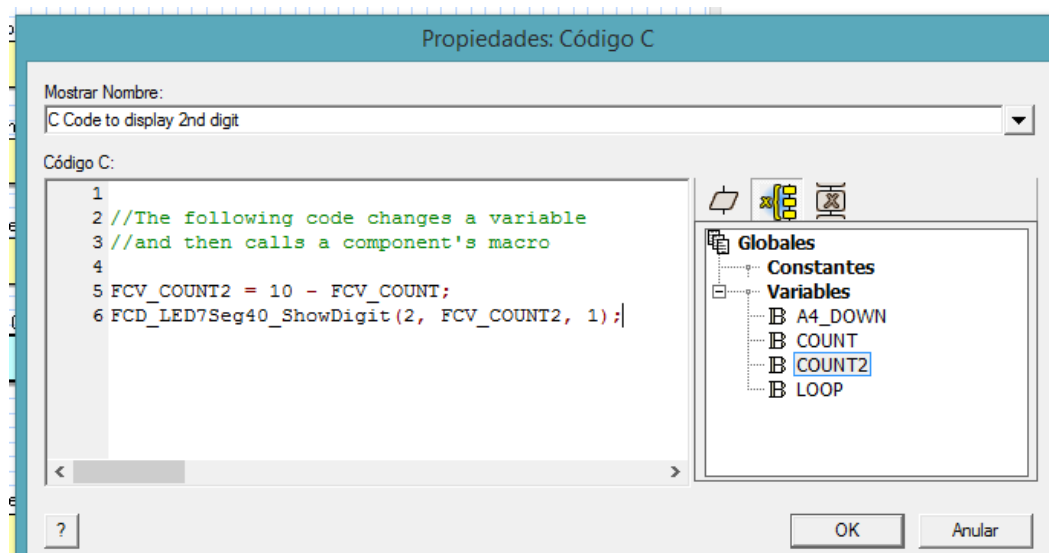
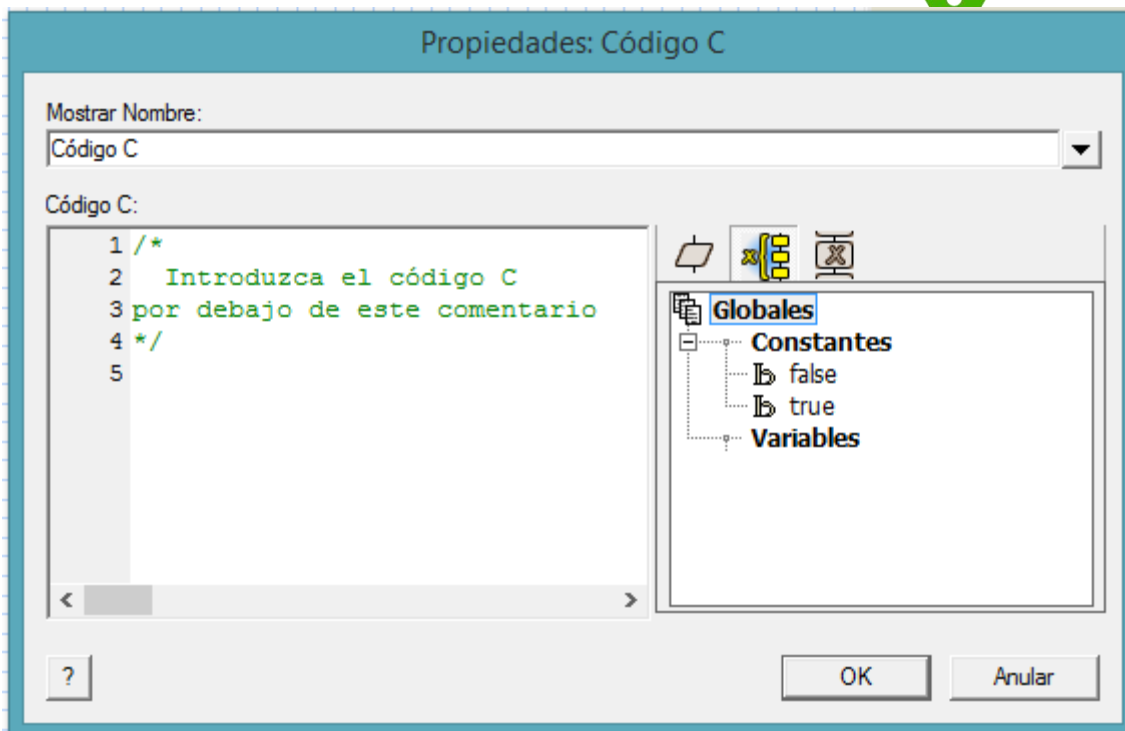
En el campo “Llamará al Macro” se indica qué Macro o función se debe ejecutar cada vez que se produzca la interrupción. Con el botón “Crear Nuevo Macro” creamos una nueva función. Con el botón “OK & Editar Macro” editamos el contenido de la Macro. Es decir, dibujamos el organigrama de tareas que esa 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 siguiente. Simplemente consiste en un editor de textos donde escribimos nuestro programa empleando instrucciones del lenguaje C o del lenguaje ensamblador.





Flowcode se ha diseñado para que personas sin experiencia en programación de microcontroladores chip puedan diseñar chips con pocos conocimientos de lenguajes de alto nivel. Sin embargo, en los programas de Flowcode se puede integrar código escrito en otros lenguajes. Los programas escritos en código C y código ensamblador se pueden integrar en Flowcode utilizando el icono Código. Esto significa que es posible tomar programas complejos (usted encontrará muchos programas en Internet) escritos en C o en lenguaje ensamblador e integrarlos en sus diseños.

Flowcode no puede simular este código, pero se pasa al microcontrolador durante la compilación.

Mostrar Nombre

El nombre del icono que aparece en el diagrama de flujo.

Cuadro de variables

Navegar rápidamente a las variables definidas en el proyecto. Puede hacer referencia a los puertos directamente o seleccionarlo de una lista de las variables globales o locales. Arrastrar y soltar las variables en el cuadro Código C, añadirá automáticamente el prefijo Flowcode específico correcto.

Código C

Introduzca cualquier código C que desee incluir en el diagrama de flujo. Flowcode no comprueba el código C, pero lo pasa directamente al compilador C cuando compila el diagrama de flujo. Es importante verificar que el código C introducido sea correcto, ya que cualquier error de sintaxis hará que falle la compilación de todo el diagrama de flujo.

Para acceder a variables, funciones de macro y puntos de conexión de Flowcode, es necesario añadir a la variable en código C los prefijos FCV_, FCM_ y FCC_NombreMacro_, respectivamente.

Por ejemplo, para utilizar una variable de Flowcode denominada DEMORA en su código C, se debe referir a ella utilizando FCV_DEMORA. Tenga en cuenta que todas las variables definidas en Flowcode se escriben en mayúsculas.

Para llamar una macro de Flowcode denominada PRUEBA en su código C, debe llamar FCM_PRUEBA();. Tenga en cuenta que todos los nombres de macros de Flowcode se escriben en mayúsculas.

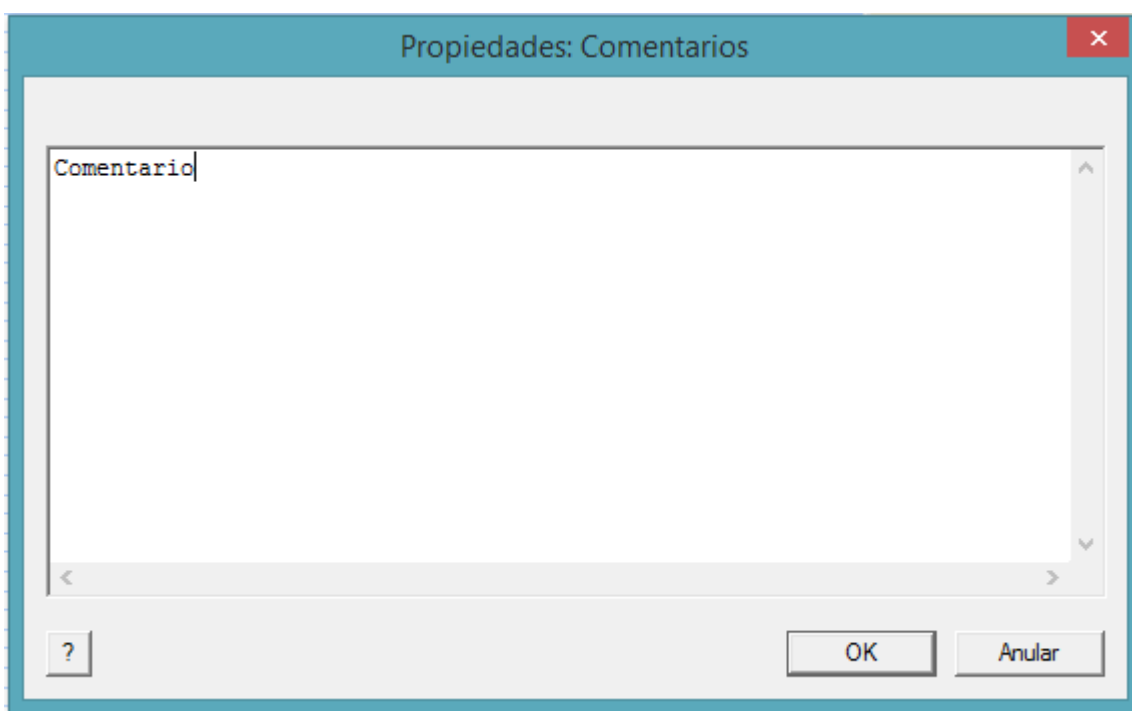
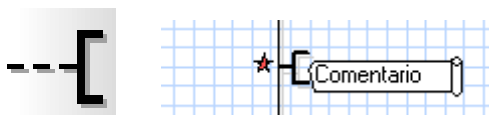
Para saltar a un punto de conexión denominado A, definido en una macro de Flowcode denominada PRUEBA, su código C debe ser goto FCC_PRUEBA_A;. Los puntos de conexión

definidos en el diagrama de flujo principal de un archivo de Flowcode llevan el prefijo FCC_Principal_.

Para introducir un carácter de tabulación en el campo de código C, utilice Ctrl+Tab

Herramienta de Comentarios

Es la última herramienta de comandos que dispone FlowCode 5. Su ventana de propiedades se muestra en la figura siguiente y consiste simplemente en un área donde podemos insertar todo tipo de textos que describan el programa y facilite su comprensión y análisis.



HERRAMIENTAS DE COMPONENTES

El programa FlowCode 5 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 anteriormente.

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”. Esta se presenta en la figura 28.

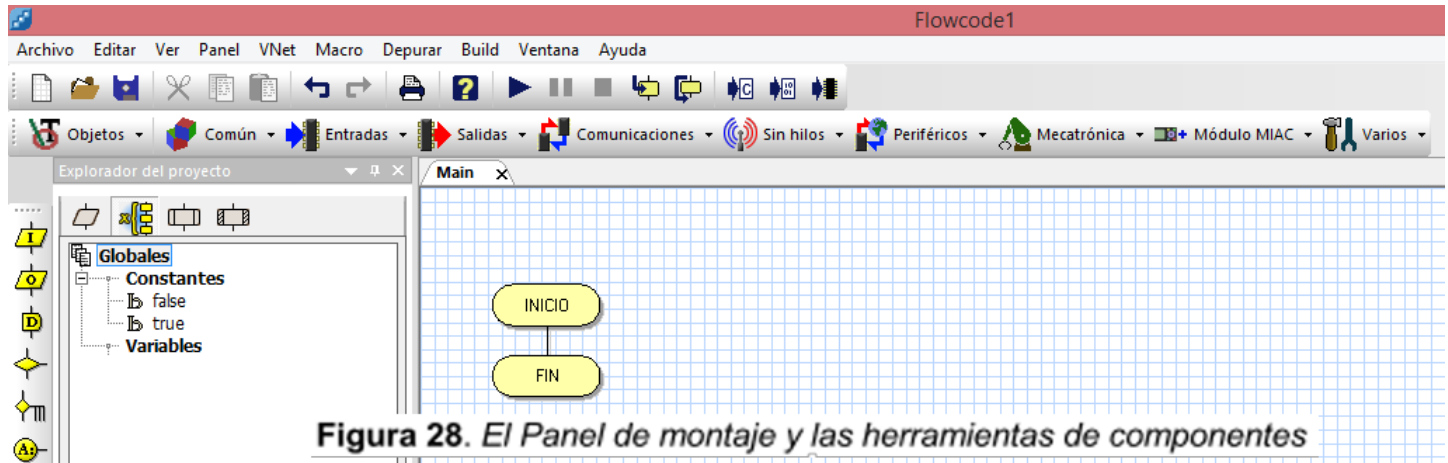


Figura 28. El Panel de montaje y las herramientas de componentes

Los componentes se organizan en nueve grupos que se seleccionan mediante otros tantos botones de la barra de componentes. 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. Esta se activa/desactiva mediante “Ver Panel → Properties”

Objetos

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

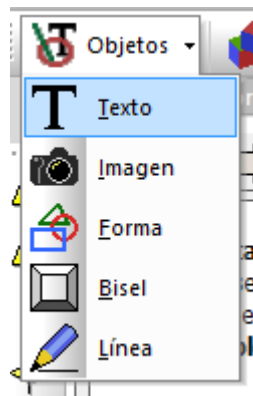
Texto. Permite escribir texto en el panel de montaje para establecer etiquetas, rótulos, signos, etc. en nuestro circuito.

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.



Común

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

Led. Coloca un led en el panel de montaje.

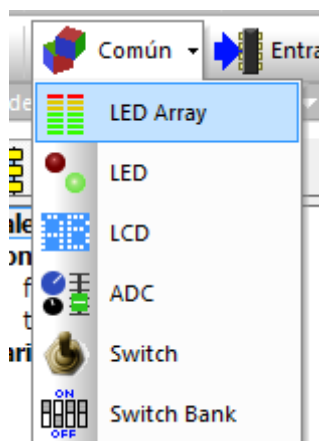
LEDarray. Coloca una barra de hasta ocho diodos leds.

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.



Entradas

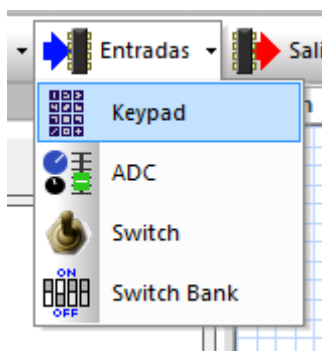
Contiene componentes o periféricos típicos de entrada. Algunos son los mismos que los que había en el grupo "Common".

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.



Salidas

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

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

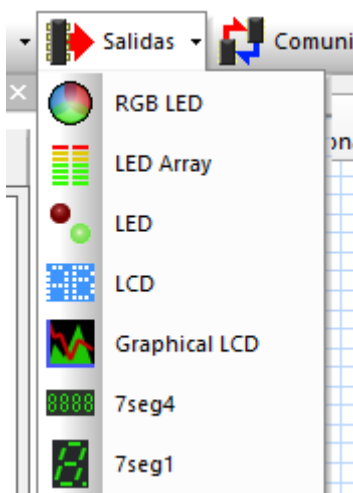
Led. Coloca un led en el panel de montaje

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.



Comunicaciones

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.

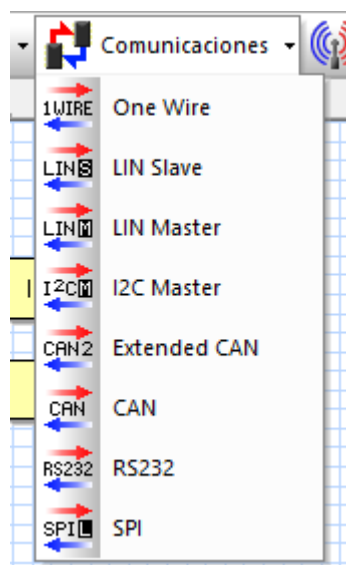
I2C_Master. Este componente permite hacer uso de Macros y funciones que implementan el protocolo I2C en el modo Master para la transferencia de datos.

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.



Sin hilos

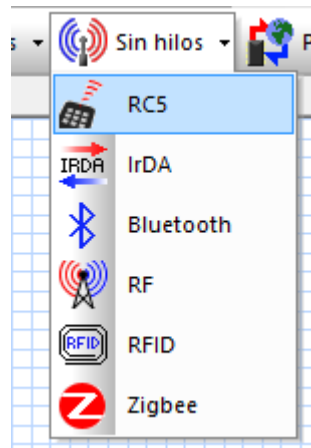
Estos componentes y sus correspondientes funciones o Macros permiten realizar transferencia de datos sin cables.

IrDA Comunicación mediante infra rojos.

RFID. Identificación por radio frecuencia.

Zigbee. Comunicación por radio frecuencia según el protocolo Zigbee

Bluetooth. Comunicación por radio frecuencia según el protocolo Bluetooth.



Periféricos

Sus componentes, junto con las Macros y Funciones que poseen, permiten implementar distinto 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.

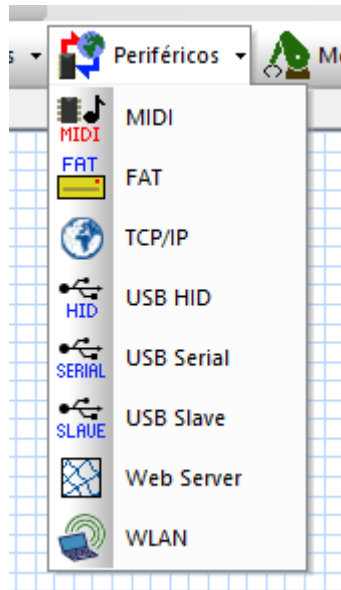
USBHID. Permite implementar un dispositivo USB como dispositivo de interface humano (HID).

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.



Mechatrónica

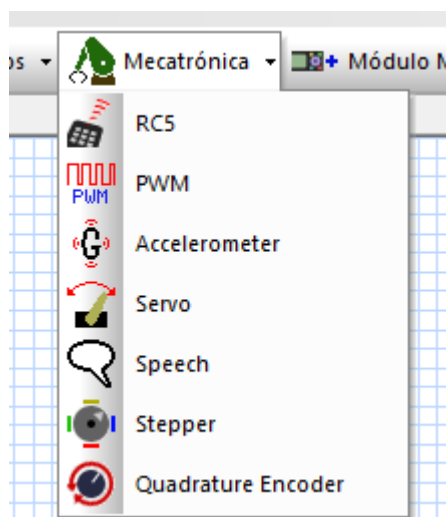
Contiene componentes auxiliares.

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.



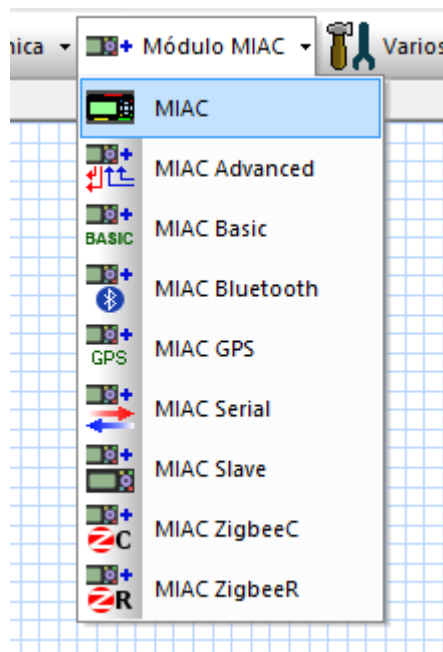
Módulo MIAC

Contiene componentes diseñados por Matrix Multimedia. Representa a un panel de control de propósito general que fabrica y comercializa Matrix Multimedia.

MIAC

MIAC advansed

Entre otros



Varios

En este grupo encontramos una serie de componentes varios

Custom2. Componente nº2 para ser definido por el usuario

Custom. Componente nº1 para ser definido por el usuario

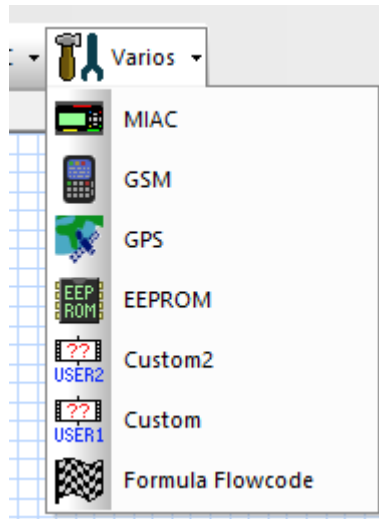
MIAC. Es un componente diseñado por Matrix Multimedia. Representa a un panel de control de propósito general que fabrica y comercializa Matrix Multimedia.

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.



Configuración y uso de los componentes

Ya hemos hecho una breve descripción de todos los componentes que posee FlowCode 5 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.

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.

Establecer las conexiones del componente

Lo siguiente que debemos establecer son las conexiones de éste componente con las patillas de E/S del microcontrolador. Basta con seleccionarlo en el panel de montaje haciendo click 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. Nos aparece una ventana de conexiones como la mostrada en la figura 29.

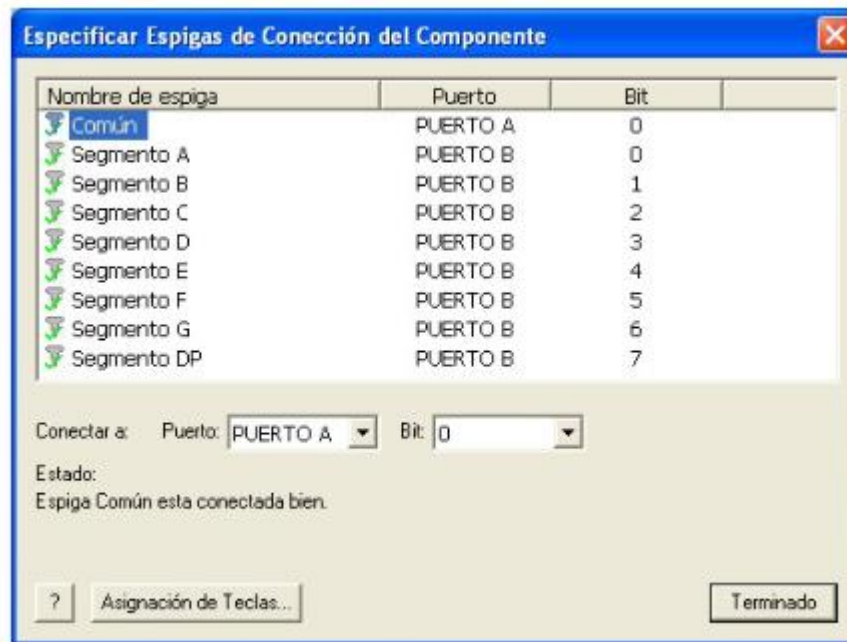


Figura 29. Estableciendo conexiones de un display de 7 segmentos

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.

Establecer las propiedades externas del componente

Esto lo hacemos seleccionando el componente, haciendo click 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 30.

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



Figura 30. *Propiedades del display de 7 segmentos*

A modo de ejemplo, en la figura 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.



Figura 31. *Propiedades de un interruptor.*

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 electro mecánico.

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

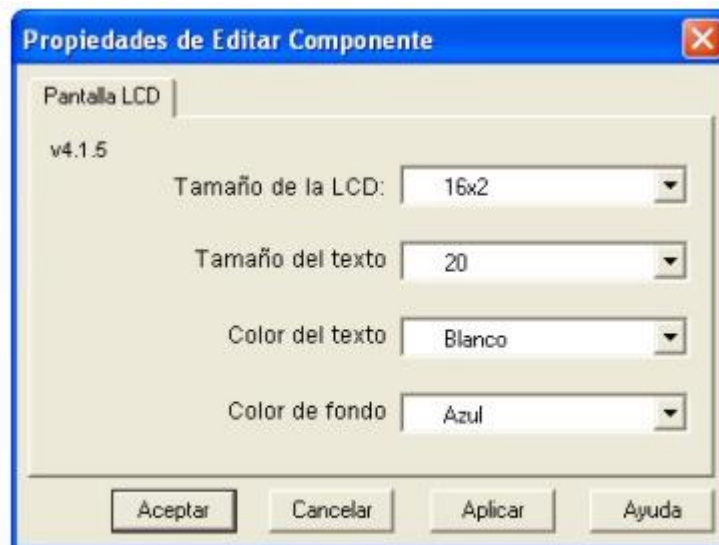


Figura 32. *Propiedades de la pantalla LCD*

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

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.

Simulación

Una de las grandes aportaciones de FlowCode 5 es el poder simular en el PC el comportamiento del programa. Para ello disponemos de los botones mostrados en la figura 35.

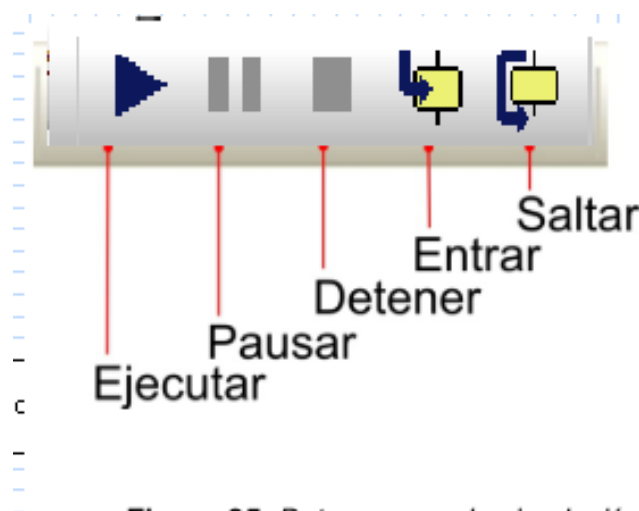


Figura 35. *Botones para la simulación*

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.

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 click sobre el interruptor que hemos puesto en el panel de montaje, los leds deben cambiar de estado.

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, por ejemplo el PIC16F886.

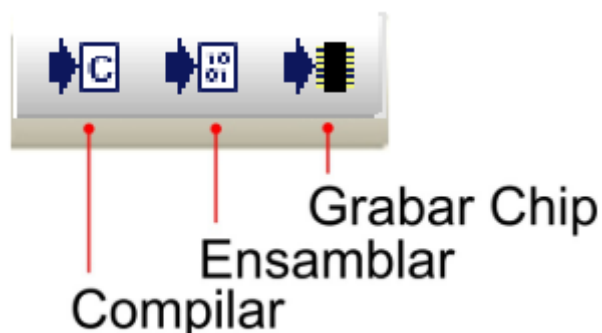


Figura 36. Botones de ensamble y grabación

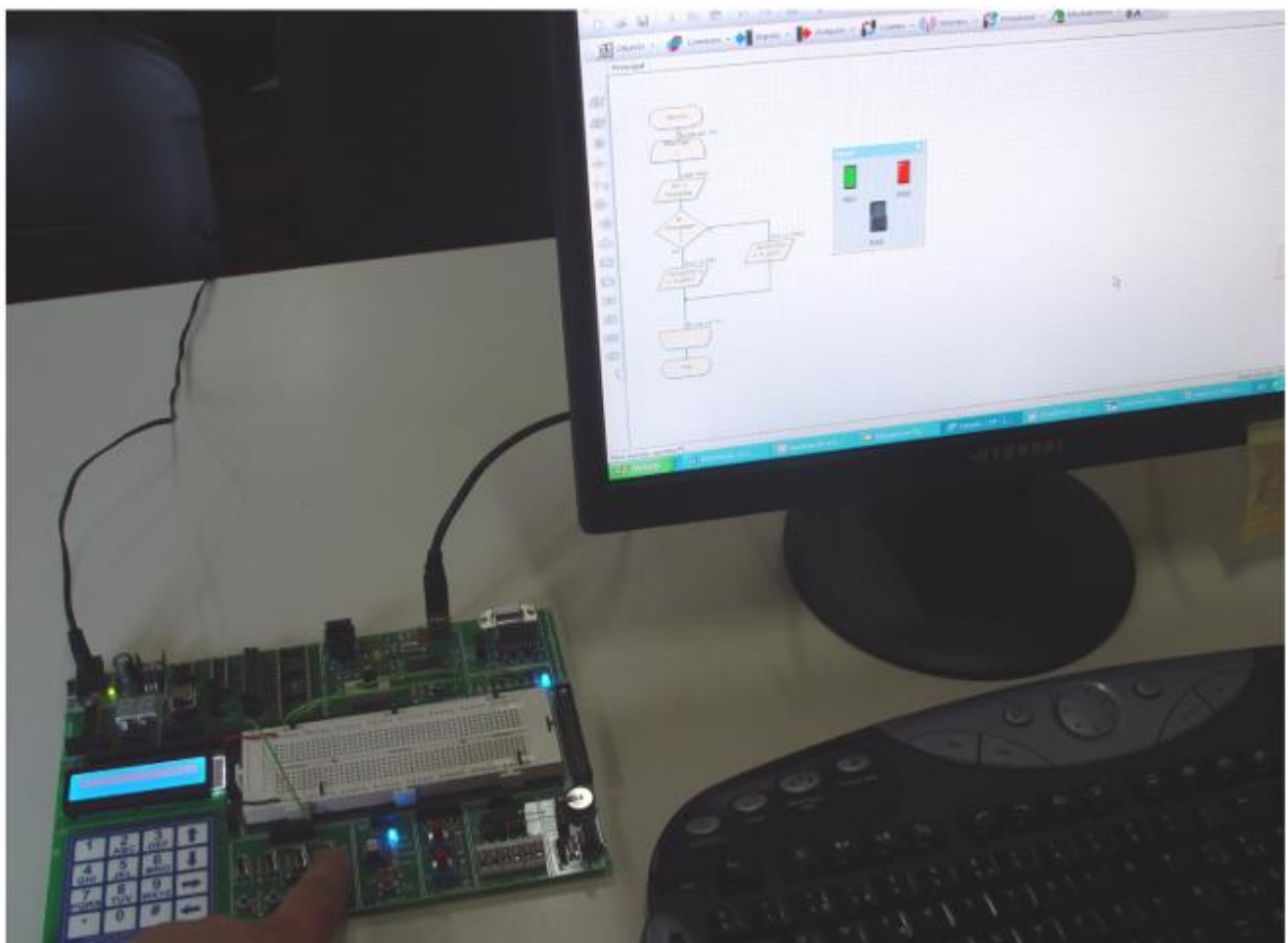
Disponemos de los botones mostrados en la figura 36.

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. Este 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 5. El PICKit-2 es uno de ellos.

En otras palabras, una vez desarrollado el organigrama de nuestra aplicación y, si disponemos de uno de los grabadores admitidos por FlowCode 5, basta con pulsar el botón “Grabar Chip” para realizar todo el proceso automáticamente. La fotografía de la figura 37 muestra la ejecución del ejemplo sobre el laboratorio USB- PIC'School.



EJEMPLOS DE PROGRAMACIÓN

Ejemplo vamos a emplear el componente LCD para visualizar mensajes.

El montaje eléctrico.

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

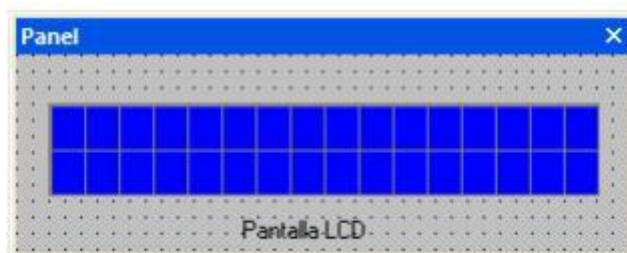


Figura 50. Montaje del ejemplo 4

Como se muestra en la figura 51, definimos las conexiones haciendo click 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 51. Montaje del Ejemplo

También hay que definir las propiedades del componente para establecer su color, forma, tamaño, etc... Hacemos click con el botón derecho y seleccionamos la opción "Propiedades ext". En la figura 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 52. *Propiedades de la pantalla LCD*

El programa

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



Figura 53. *El organigrama del ejemplo*

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

Simulación

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

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.

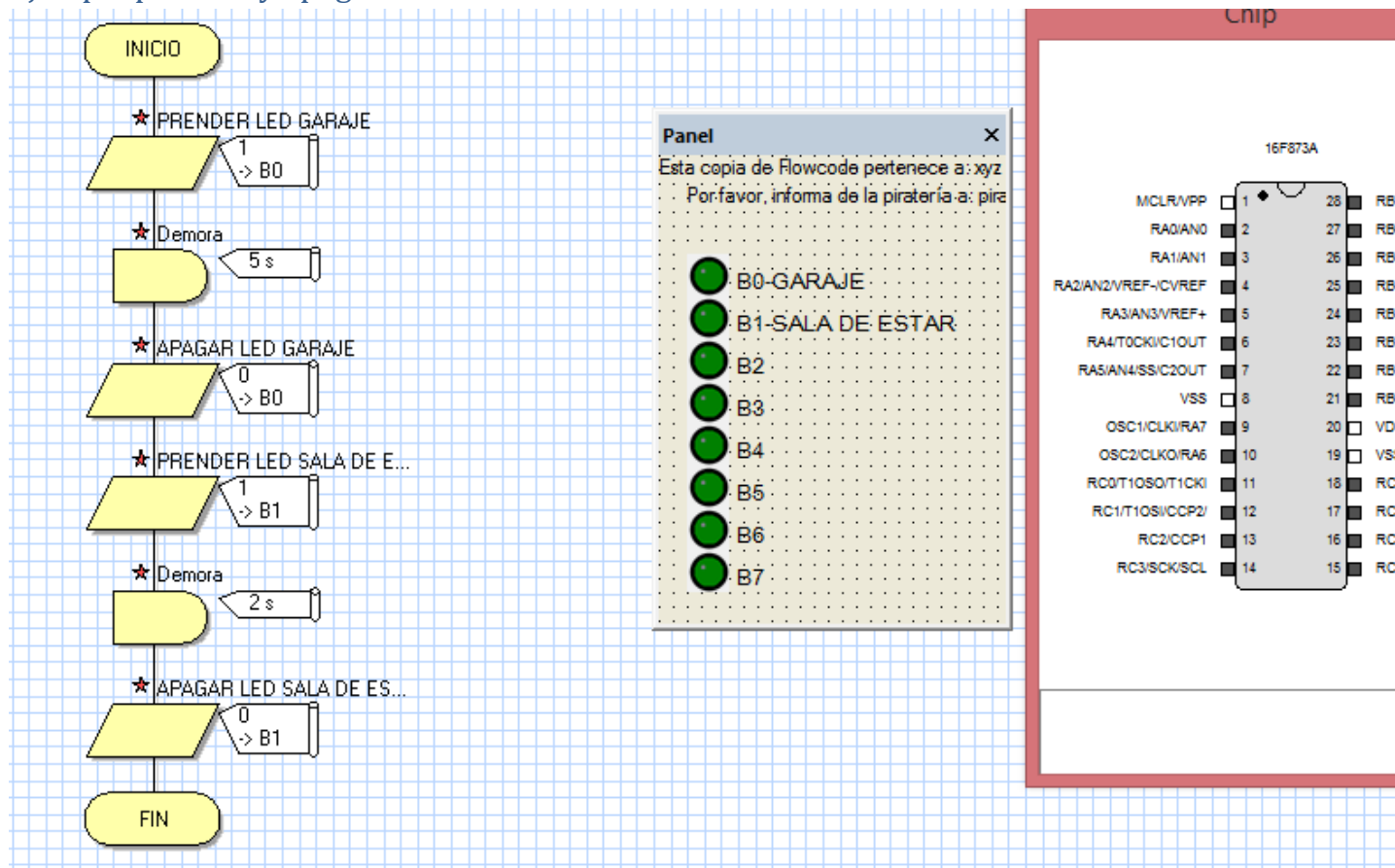
RB0:RB3 se conectan con las patillas L4:L7 de la pantalla. La señal RS de control se conecta con la salida RA3 y la de activación con RA1.

FlowCode 5 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 54 muestra la ejecución del ejemplo sobre el laboratorio.

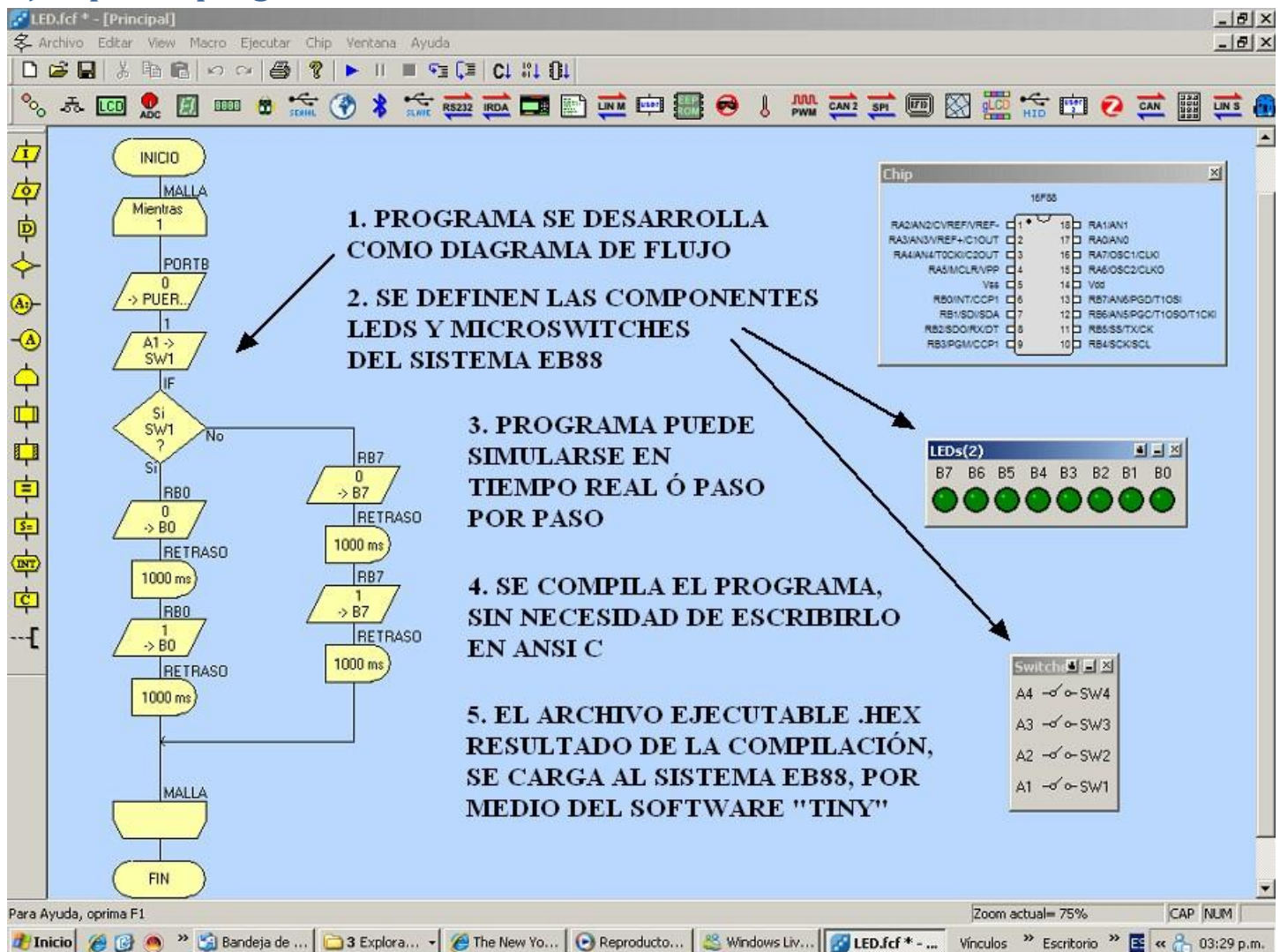


Figura 54. Ejecución del Ejemplo 4 sobre el laboratorio USB-PIC'School

Ejemplo prender y apagar led con demora



Ejemplo de programa armado



Como pueden observar este método de programación es sumamente simple en comparación a los sistemas de programación tradicionales.

Las prácticas de este se dejan a cargo de los alumnos en el taller de electrónica.

También lo descripto aquí puede complementarse con la herramienta ayuda





Bibliografía:

- <https://www.youtube.com/user/elprofegarcia/videos>
- http://www.matrixtsl.com/lc_index.php?p=18
- www.matrixmultimedia.com
- Ventana de ayuda del programa FLOWCODE.
- <https://youtu.be/57kD9NJ7Kwo>