

El lenguaje grafico FlowCode y el laboratorio USB-PIC'School. Una revolucion en la formación y el diseño de aplicaciones con μ Cs PIC (2ª parte)

Por Mikel Etxebarria Isuskiza



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

En el mes de Febrero del 2010, en esta misma revista, publicamos un artículo dedicado al laboratorio USB-PIC'School. En él se explicaba sus características, arquitectura y posibilidades. Han transcurrido dos años desde que se comenzó la comercialización de este equipo. Durante este tiempo han sido numerosos los centros de estudios, empresas, alumnos, profesionales y aficionados que han acercado al mundo de los microcontroladores PIC de la mano de esta potente, flexible y económica herramienta, tanto a nivel nacional como internacional. También han sido numerosas las presentaciones, seminarios y cursos de formación sobre PIC's que, desde MSE, hemos realizado utilizando este laboratorio como plataforma de prácticas en todas las demostraciones, charlas y clases impartidas.

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

Herramientas de componentes. Clasificación

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 la primera parte de este artículo.

Para acceder a los componentes es necesario tener habilitado el panel de montaje mediante el comando "Ver → Panel" y la barra de com-

ponentes mediante "Ver → Herramientas de componentes". Esta se presenta en la figura 31.

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"

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 las etiquetas, rótulos, signos, etc. de 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.

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.

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.

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

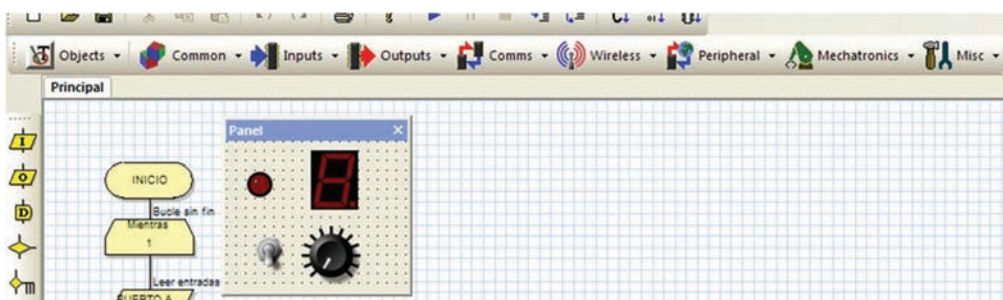
- Led. Coloca un led en el panel de montaje

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

- gLCD. Coloca una pantalla gráfica

- LCDDisplay. Inserta una pantalla LCD alfanumérica.

Figura 31. El Panel de montaje y las herramientas de componentes



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

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.

Wireless

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.

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.

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.

Misc

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.

Herramientas de componentes. Configuración y uso.

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.

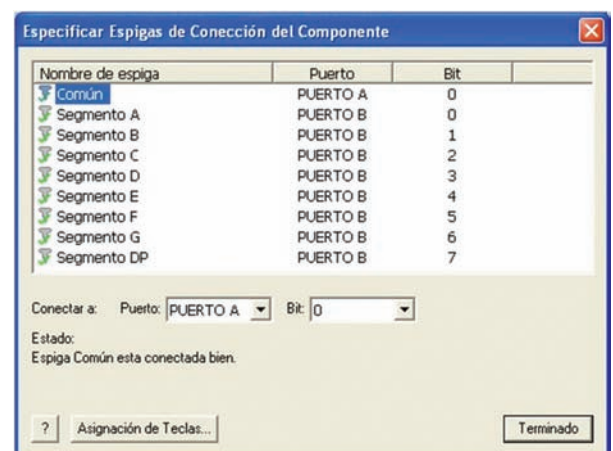
Uso del componente

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 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. Nos aparece una ventana de conexiones como la mostrada en la figura 32.

Figura 32. 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

Figura 34. Propiedades de un interruptor.

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.

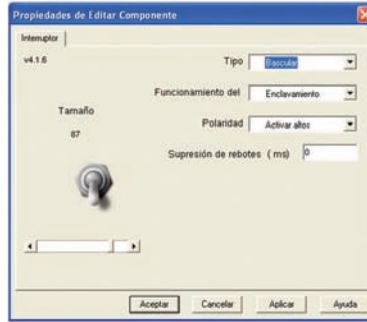
No todos los componentes necesitan siempre las mismas conexiones. Habrá algunos que se conectan mediante una única patilla, otros, mediante dos y otros mediante diez.

Figura 36. Montaje del Ejemplo 1

Establecer las propiedades externas del componente

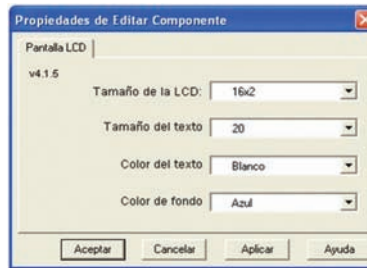
Esto lo hacemos seleccionando el componente, haciendo clic con el botón derecho del ratón, y eligiendo la opción "Propiedades ext". También desde la ventana de propiedades del panel de montaje. Cada componente tiene sus propias propiedades. En el caso de un display de 7 segmentos nos aparece una ventana como la mostrada en la figura 33.

Figura 35. Propiedades de la pantalla LCD



clásicos de cualquier dispositivo electro mecánico. Por supuesto, podemos seleccionar el tamaño con el que se verá sobre el panel de montaje.

En el caso de utilizar una pantalla LCD alfanumérica, sus propiedades serían las mostradas en la figura 35.



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.

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.

Ejemplo 1

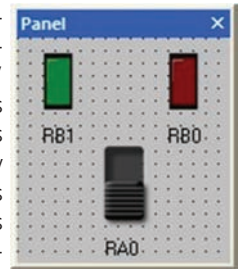
Se trata de leer un interruptor conectado a la entrada RA0. Si está a nivel "0" se activa un led conectado a la línea RB0 de salida. Si está a nivel "1" se activa un led conectado a la línea RB1 de salida.

El montaje eléctrico

En el panel de montaje distribuimos los componentes que vamos a utilizar más o menos como

se muestra en la figura 36. 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.



El programa

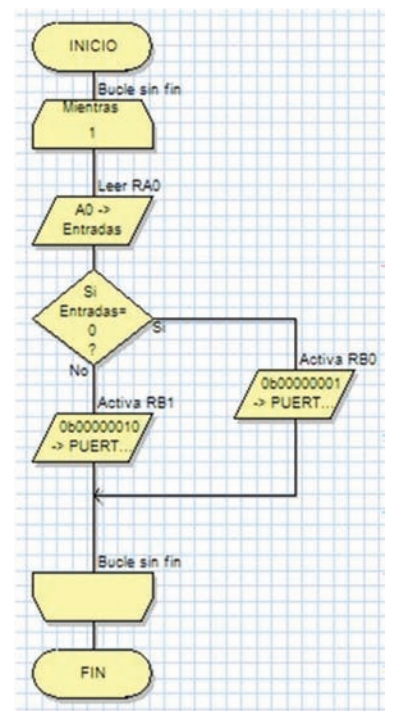
En el área de trabajo hemos confeccionado el organigrama que se muestra en la figura 37. Responde a las expectativas del ejemplo y, por supuesto, se podría haber resuelto de múltiples formas.

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.

Figura 33. Propiedades del display de 7 segmentos



Figura 37. El organigrama del ejemplo 1



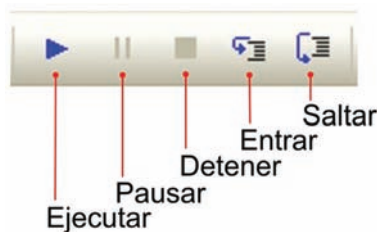
El programa empieza leyendo el estado del Puerto A que va a parar a la variable "Entradas". Si editamos esta herramienta de comando observamos que la lectura se realiza en el modo de bit único, el bit 0 del Puerto A.

A continuación el comando de "Toma de decisión" evalúa si el valor de la variable "Entradas" (el estado de RAO) 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.

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 figura 38.



- 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 o iconos 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.

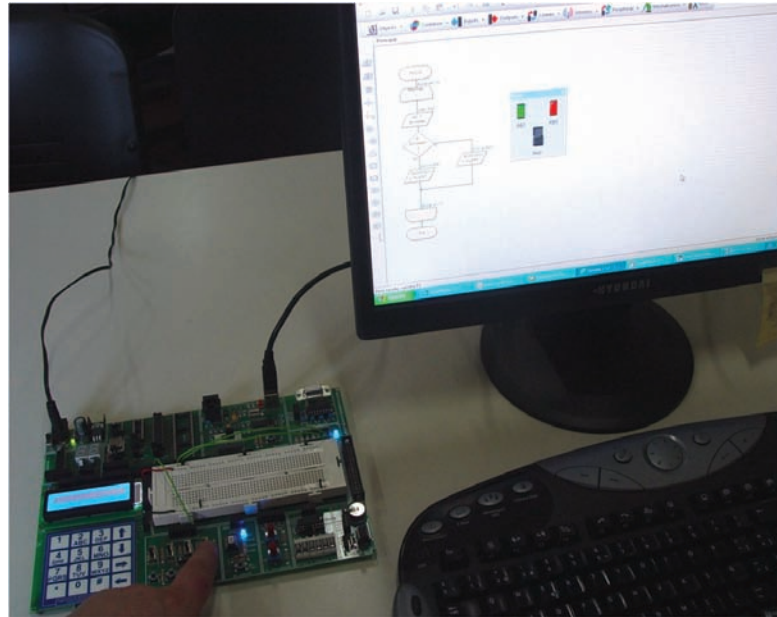


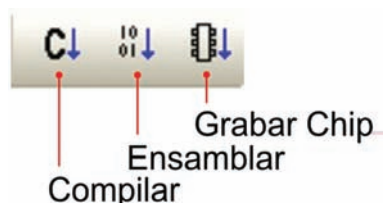
Figura 40. Ejecución sobre USB-PIC'School

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.

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 39.



•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 4. El PICKit-2 es uno de ellos y ... ¡¡ es compatible con el USB-PIC'School !! de Ingeniería de Microsistemas Programados.

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 40 muestra la ejecución del ejemplo sobre el laboratorio USB-PIC'School.

Figura 38. Botones para la simulación

Figura 39. Botones de ensamblado y grabación

Figura 43. Propiedades del componente led

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.

El montaje eléctrico

En el panel de montaje realizamos el circuito mostrado en la figura 41. 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

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 42 vemos las propiedades que hemos establecido para el componente ADC (potenciómetro P1).

Figura 41. Montaje del Ejemplo 2

Figura 45. Macro "Pulso_en_RB0"

Figura 42. Propiedades del ADC (potenciómetro)

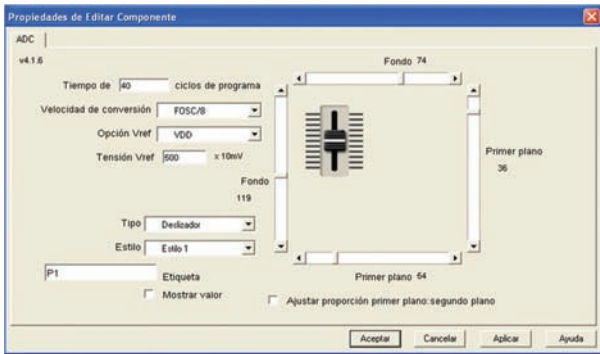
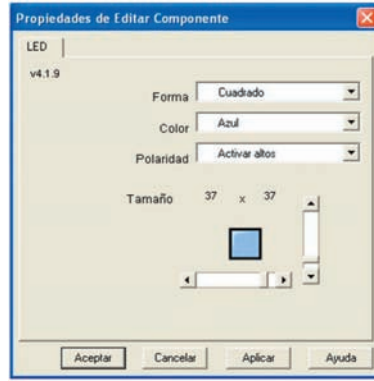


Figura 44. El organigrama del ejemplo 2

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 43 muestra las propiedades seleccionadas para el Led: Forma y Color.

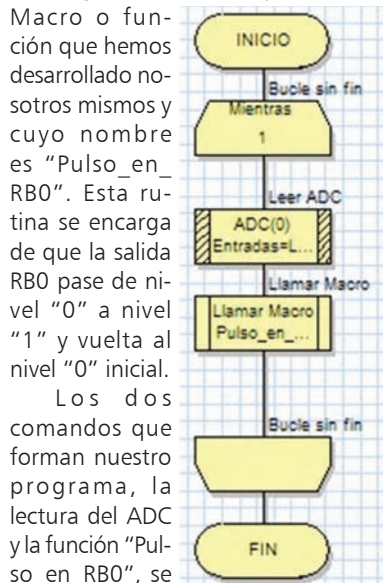
También se establece la polaridad. Esto es, el nivel lógico con el cual se debe activar el led. En el ejemplo se activa mediante nivel "1".

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

El programa

El organigrama de este segundo ejemplo se muestra en la figura 44. 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".

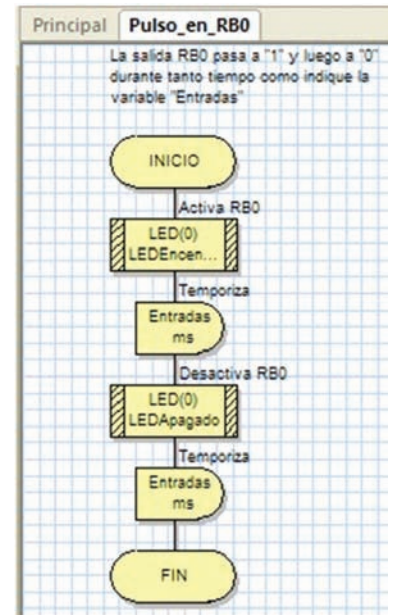


encuentran dentro de un bucle que se ejecuta indefinidamente.

En la figura 45 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". Esta 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.



Este 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.

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"

Este 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". Este 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.

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 de este ejemplo y observamos su ejecución, tal y como se muestra en la fotografía de la figura 46.

Ejemplo 3

Con este ejemplo vamos a realizar un contador de pulsos. Estos 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 mediante la línea de salida RA2 del Puerto A.

El montaje eléctrico

En el panel de montaje realizamos el circuito mostrado en la figura 47. Definimos las conexiones haciendo click 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.

Definimos ahora las propiedades de cada componente para establecer su color, forma, tamaño, etc... Hacemos click con el botón derecho y seleccionamos la opción "Propiedades ext". En la figura 48 vemos las



Figura 48. Propiedades del display

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.

La figura 49 muestra las propiedades exteriores seleccionadas para el pulsador. En "Tipo" elegimos el modelo de mecanismo: "Pulsador", "Bascular", "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.

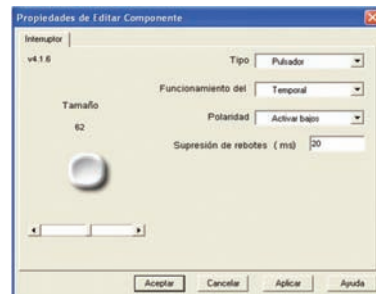


Figura 47. Montaje del Ejemplo 3

Figura 49. Propiedades del pulsador

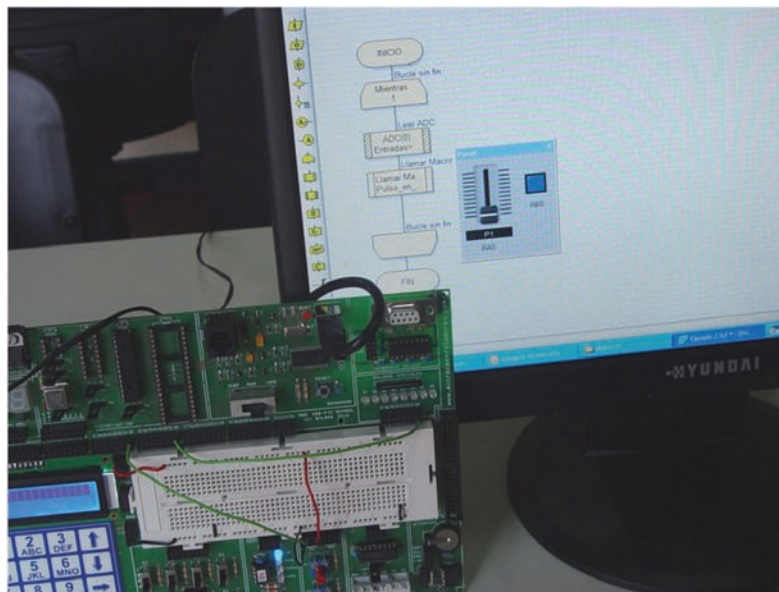


Figura 46. Ejecución del Ejemplo 2 sobre el USB-PIC'School

Con la barra deslizante establecemos el tamaño.

El programa

El organigrama de este ejemplo se muestra en la figura 50. 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.

Figura 50. El organigrama del ejemplo 3

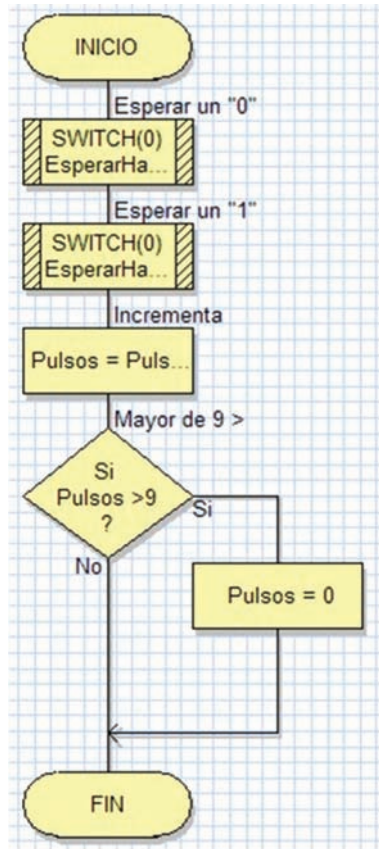
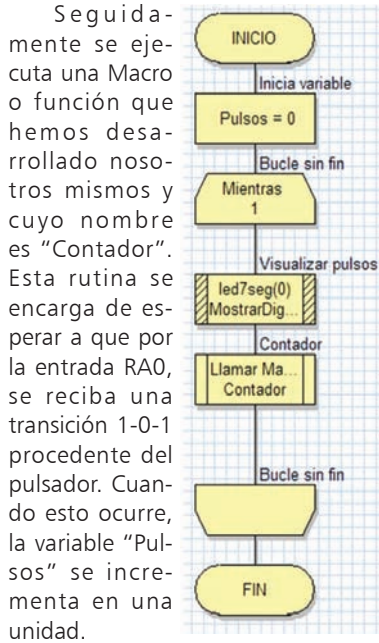


Figura 51. Macro "Contador"

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

En la figura 51 se muestra el organigrama correspondiente a la función "Contador" que hemos desarrollado para este ejemplo.

Utilizamos la herramienta "Marco 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.

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 esta va desde cero a nueve y vuelta a empezar de cero. Es decir, hemos construido una década contadora.

Este 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

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 52.

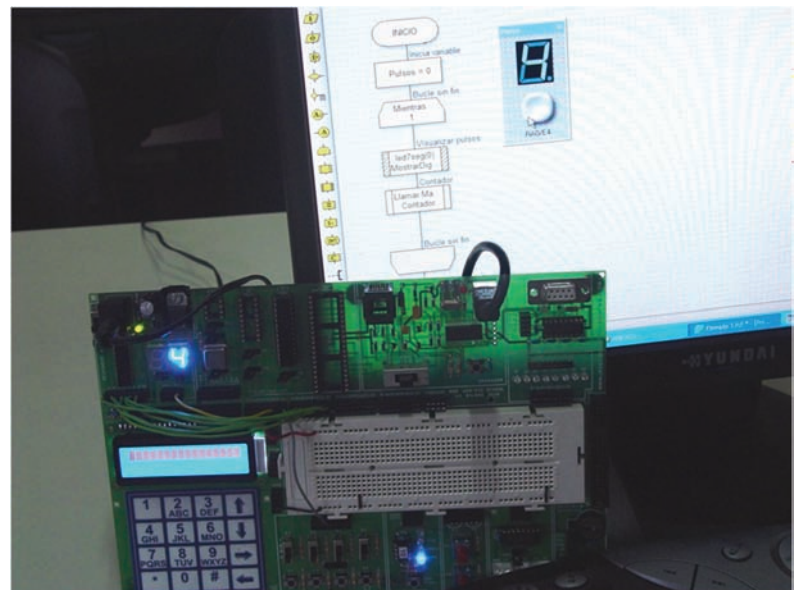


Figura 52. Ejecución del Ejemplo 3 sobre el laboratorio USB-PIC'School

Ejemplo 4

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

El montaje eléctrico

En el panel de montaje realizamos el circuito mostrado en la figura 53. Únicamente consta de una pantalla LCD alfanumérica de 2 líneas y 16 caracteres por línea.

Como se muestra en la figura 54, 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.

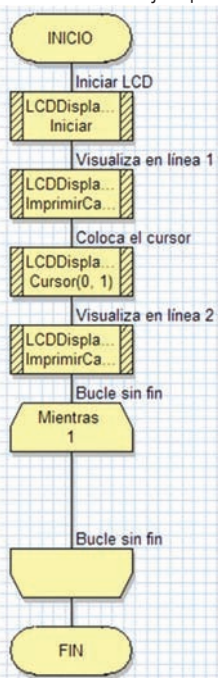
La señal RS se controla desde la salida RA3 del Puerto A y la señal de activación de la pantalla LCD desde la salida RA1.

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 55 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.

El programa

El organigrama de este ejemplo se muestra en la figura 56. 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

Simulación

Mediante el 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 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 57 muestra la ejecución del ejemplo sobre el laboratorio.

Bibliografía

"Manual de Usuario del USB-PIC'School", "Ingeniería de Microsistemas Programados S.L., www.microcontroladores.com

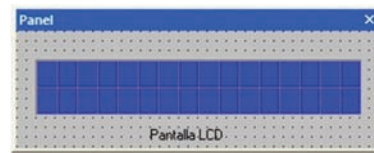


Figura 53. Montaje del ejemplo 4

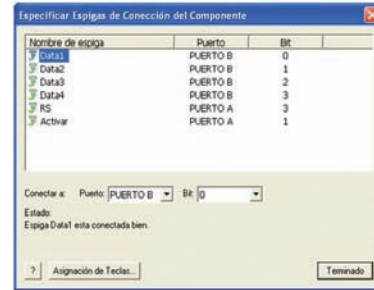


Figura 54. Montaje del Ejemplo 4

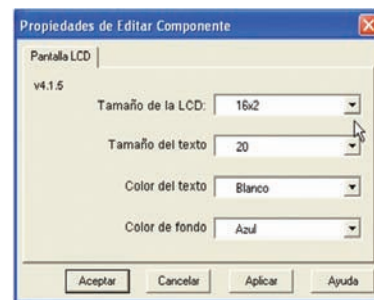


Figura 55. Propiedades de la pantalla LCD



Figura 57. Ejecución del Ejemplo 4 sobre el laboratorio USB-PIC'School

"Microcontroladores PIC. Diseño Práctico de Aplicaciones. PIC12F508 y PIC16F84A" Primera Parte, 4ª edición, Angulo, J.Mª; Etxebarria, A. y Angulo, I., ISBN: 978-84-481-5647-3, Mc Graw-Hill.

"Microcontroladores PIC. Diseño Práctico de Aplicaciones. PIC16F87X y PIC18FXXX" Segunda Parte, 2ª edición, Angulo, J.Mª, Romero, S. y Angulo, I., ISBN: 84-481-4627-1, Mc Graw-Hill.

"Microcontroladores PIC. Teoría y Práctica" Mikel Etxebarria Isuskiza, ISBN 978-84-92779-98-7. Creaciones Copyright S.L.

Figura 56. El organigrama del ejemplo 4