MITSUBISHI ELECTRIC
Changes for the Better

GX-Developer 8.12- Manual Práctico

Versión 1.1

Antes de empezar...

- Este manual pretende introducir al usuario en la programación de los autómatas Mitsubishi Electric en entorno GX-Developer. Mucha y más completa información se encuentra en los manuales de Hardware / Software de las series FX/A/QnA y Q.

- Este manual hace referencia en concreto a la versión 8.12. Algunos controles y menús pueden diferir un poco en versiones inferiores o superiores.

- Si tiene alguna duda sobre la instalación o uso de los autómatas, así como su programación, póngase en contacto con el distribuidor más próximo .

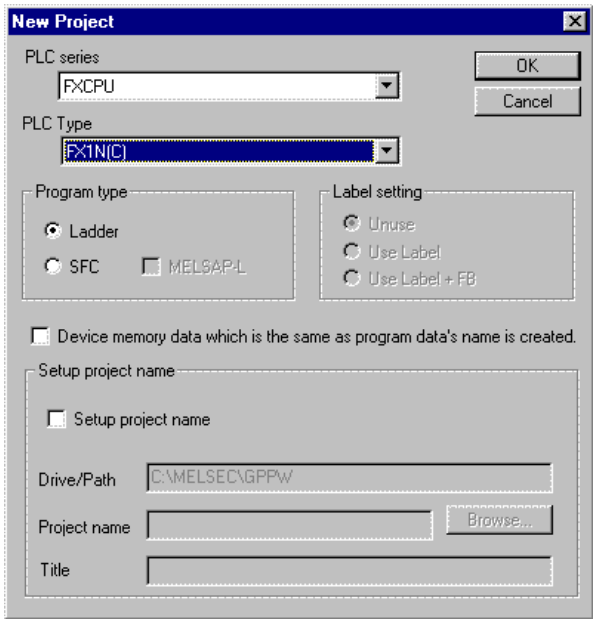
- Las explicaciones de este manual están sujetas a mejoras y revisiones sin previo aviso.

ÍNDICE:

Programación con GX Developer

CREACIÓN DE UN PROYECTO NUEVO:

Para empezar un proyecto nuevo hay que clicar sobre la opción del menú principal "Project-New project…". Con esto se abrirá el siguiente formulario:



A través de la opción **PLC Series**(1) se puede seleccionar la serie de PLC's con la que se va a trabajar.

Con la opción **PLC Type** (2) se escogerá el modelo de CPU apropiado.

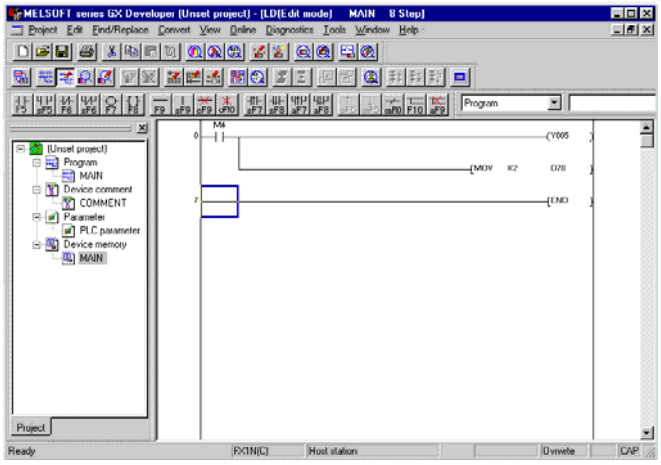
Dentro de **Program type** (3) debe escogerse entre trabajar con lenguaje de contactos (LADDER) o a través de diagrama de flujo SFC. Para la serie Q, hay la posibilidad de seleccionar el Label setting, que permite trabajar utilizando la programación por Etiquetas (labels) o con etiquetas y Bloques de función (FB).

La opción 4 sirve para crear directamente un archivo de dispositivos, con el que iniciar los valores de las variables (datos D y marcas de bit M). Si no se selecciona, tenemos la posibilidad de activarlo durante la ejecución del proyecto.

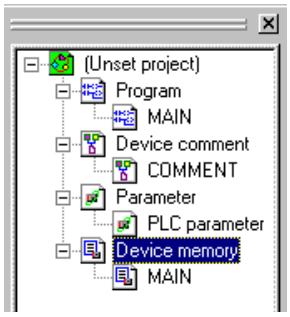
Si se activa el **Setup project name** (5) tenemos la posibilidad de darle un nombre al proyecto, así cuando se quiera guardar el proyecto no será necesario teclear el nombre del archivo.

ENTORNO DE PROGRAMACIÓN:

Cuando se abre un nuevo proyecto aparece en la pantalla el entorno de programación. Es desde esta pantalla desde donde deberemos programar y configurar el autómata.



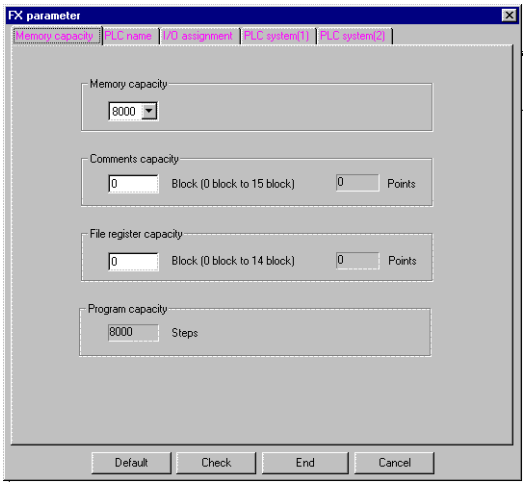
EL PROJECT DATA LIST:



En la parte izquierda aparece el **Project data list**, éste sirve para separar las diferentes secciones de que consta el proyecto. Program, Device comment, Device memory, Device init y Parameter (PLC Parameter, Network Parameter). Clicando en este árbol sobre la selección adecuada se puede acceder a cada una de estas secciones.

PROGRAM: Desde Program tenemos el programa principal (en este caso es MAIN), si se trabaja con un autómata de la serie Q/QnA aquí puede aparecer una lista de programas ya que las CPU's de esta serie pueden trabajar en un entorno multiprograma (trabajando con varios módulos programados de manera independiente y juntados por la CPU a través de varios modos de funcionamiento). Con las series FX/A se trabaja con un programa que por defecto se llama MAIN (aunque puede ser modificado el nombre).

PARAMETER: Con el formulario que aparece si se selecciona Parameter-PLC Parameter obtenemos el entorno de configuración del autómata. En este formulario aparecen varias solapas desde las cuales accedemos a cualquier tipo de información que pueda necesitar el PLC para saber como debe funcionar.



Estos parámetros deben descargarse al autómata conjuntamente con todo el programa en el momento en que pasamos el proyecto al PLC.

El puerto de comunicaciones, la organización de la memoria de programa y datos, el tipo de ejecución de cada programa que ejecuta el PLC, rango de dispositivos retentivos, organización de las tarjetas de PLC conectada, direccionamiento, etc… son algunas de las cosas que pueden ser configuradas desde esta sección.

Tenemos también la posibilidad de acceder (en series superiores) a los parámetros de configuración de redes como Ethernet, CC-Link o MELSECNET/10 y MELSECNET/H a través del Network parameter.

Las opciones disponibles para configurar el autómata difieren en cada serie utilizada, por lo que se omite su explicación en esta guía (ver manual correspondiente de cada serie de autómatas).

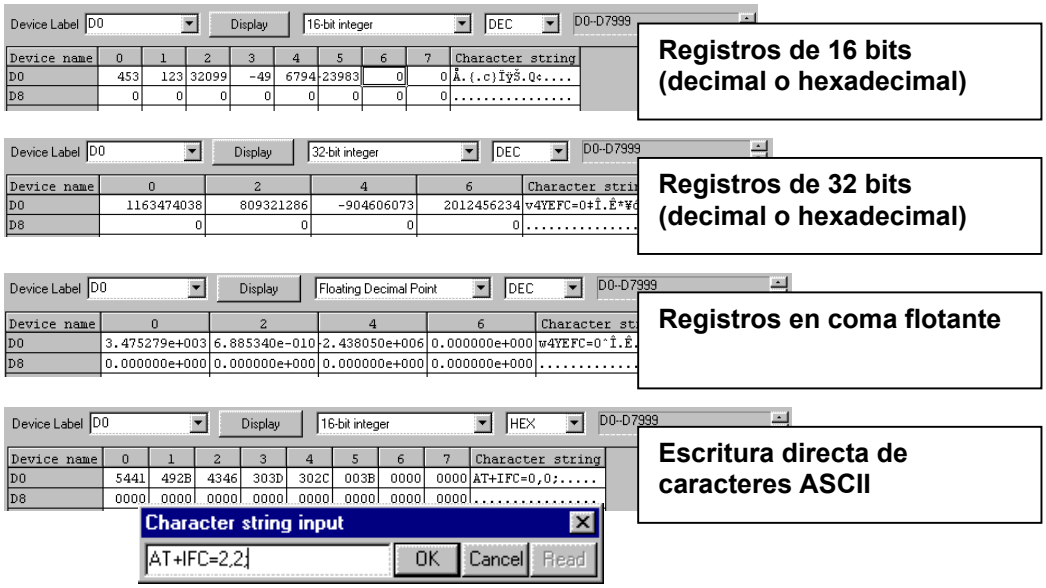
DEVICE COMMENT: Utilizando la sección Device comment pueden ser comentadas las variables utilizadas en el proyecto. Aparece un formulario desde el cual podemos escribir un texto asociado a cada dispositivo. Se explicará en páginas posteriores de este manual.

NOTA: Es en los parámetros desde donde se deberá configurar si se quieren guardar los comentarios al autómata junto al programa que se transfiere.

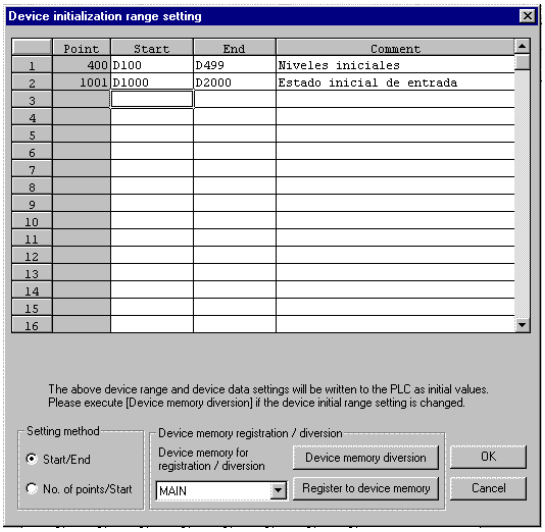
DEVICE MEMORY: Si desplegamos el menú Device memory podemos ver todos los dispositivos del autómata (D, M…) para poder configurar su estado inicial. Posteriormente, en el momento de la descarga del programa al PLC, pueden ser también transferidos todos estos datos iniciales para los dispositivos indicados.

NOTA: Si se quieren mantener en la memoria del PLC los datos transferidos, deben ser mapeados en posiciones retentivas. Los autómatas Mitsubishi tienen zonas de memoria de dispositivos D y M **volátiles**, por lo tanto cuando el autómata pasa a estado STOP, éstos se pierden y pasan a tener un valor de cero, mientras que los D y M **retentivos** se mantienen con el mismo valor que tenían en el momento de pasar a STOP o quitar tensión en el PLC. Para más información sobre el rango de dispositivos D y M retentivos que tiene cada serie de autómatas o como configurarlo desde la configuración de parámetros, referirse al manual correspondiente.

Desde esta pantalla se pueden configurar los registros en diferentes formatos: a 16 bits, 32 bits (en decimal o hexadecimal), en coma flotante (para modelos que tienen esta posibilidad), o incluso escribir directamente un texto y almacenar el código ASCII correspondiente a cada carácter en registros consecutivos (D)



DEVICE INIT: Esta sección sólo está disponible en autómatas de las series Q/QnA. Sirve para asignar valores creados con el Device memory, de manera que sean considerados por el PLC como valores iniciales de variables. En esta ocasión, los valores descargados al autómata pasan a ser los valores por defecto de las variables asignadas.



EDITANDO PROGRAMA:

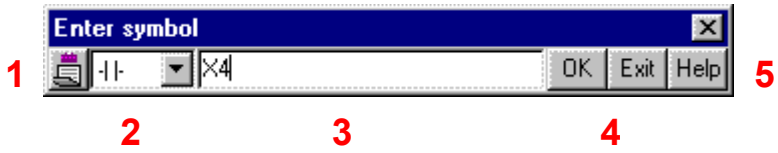
EDICIÓN DE PROGRAMA EN LENGUAJE DE CONTACTOS (LADDER)

En la parte superior hay una colección de botones dedicados a la inserción de contactos y bobinas en la pantalla de edición. Se tiene la posibilidad de insertar componentes a través de estos botones, desde el menú general (Edit – Ladder symbol…) o con la utilización de teclas abreviadas.

En los botones se puede ver, de forma abreviada, cual es la tecla o combinación de teclas que deberemos pulsar para insertar en la pantalla de edición un componente de programación. Las abreviaturas tienen el siguiente significado:

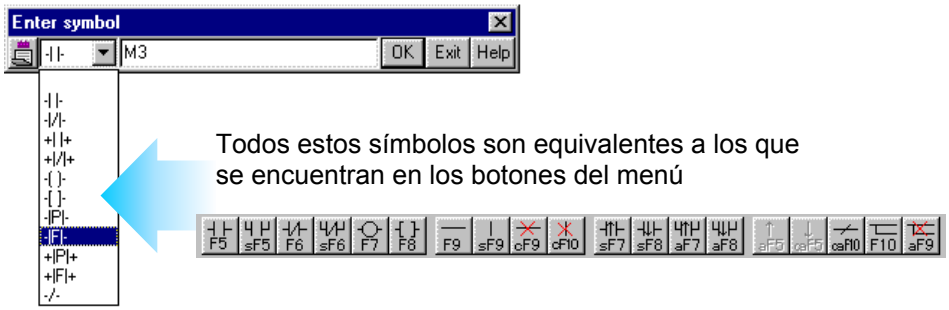
(**s**__SHIFT, **c**__CTRL, **a**__ALT, **ca**__CTRL+ALT)

Cuando se inserta un componente aparece en pantalla el formulario de inserción **Enter symbol** desde el cual escribiremos el nombre del contacto/bobina o la instrucción a pegar.



1. Clicando sobre este icono podemos hacer que la edición se repita una vez detrás de otra hasta que se desactive.

2. A través de este cuadro puede ser cambiado el componente a insertar en la pantalla de edición

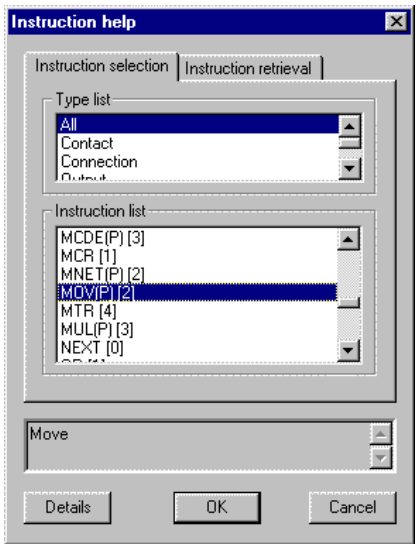


3. Se escribe en este cuadro de texto el contacto/bobina o instrucción a ejecutar. Si es una instrucción se tecleará el nombre de ésta seguido de todos sus parámetros separados por espacios en blanco. (por ejemplo MOV K3 D10)

4. Botones de confirmación (OK) y salida sin edición (Exit). Se puede confirmar también a través de la tecla RETURN.

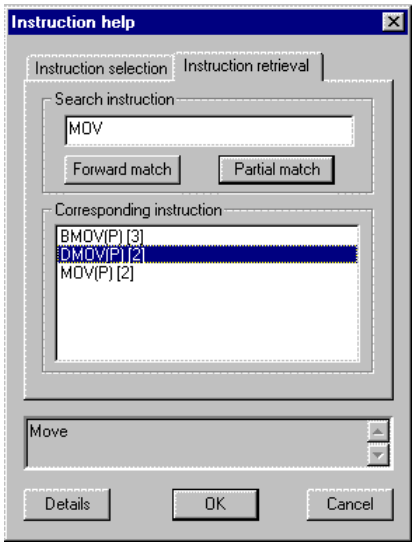
5. Si se pulsa el botón Help podremos acceder al menú de ayuda. En este menú encontraremos un formulario que nos puede ser muy útil para saber como eran los parámetros de una función, o simplemente buscar una función para insertarla en el editor.

Si entramos en la opción Help, tenemos dos posibles caminos para encontrar la instrucción que queremos ejecutar: **Instruction selection** e **Instruction retrieval.**



A. Con **Instruction selection** se selecciona el tipo de instrucción que queremos buscar (o sea, a que grupo pertenece: instrucción de comparación, de control de flujo del programa, operación lógica, de rotación…etc).

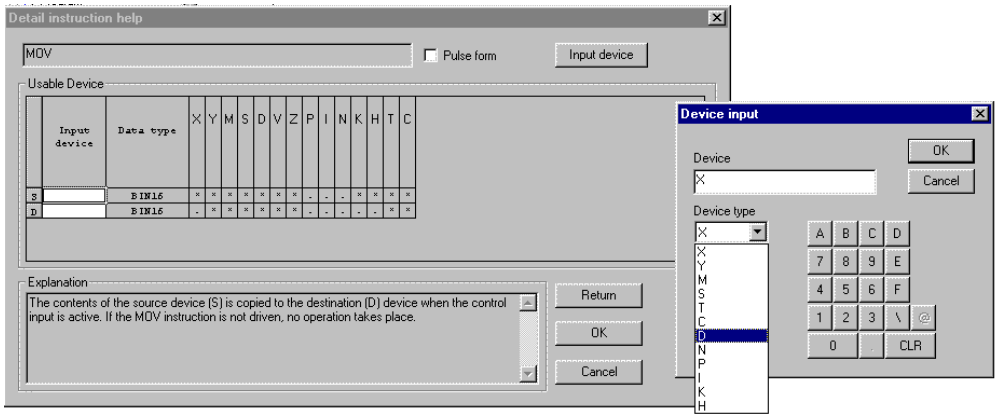
Seguidamente, una vez seleccionada la instrucción, podemos acceder a una explicación del funcionamiento y de los parámetros que deberían ser insertados. (figura C)



B. Con **instruction retrieval** se debe escribir el nombre de la instrucción a buscar, o parte de esta, para que se muestren por pantalla una lista de instrucciones que tienen un fragmento de caracteres iguales al texto entrado en el cuadro Search instruction.

También, una vez se ha encontrado la instrucción, se puede acceder a un formulario detallado de instrucciones.

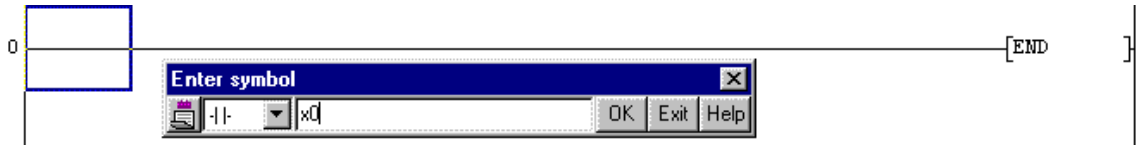
C. Con la opción **Details** se visualiza el siguiente formulario con el cual podemos insertar los parámetros de la instrucción y podemos ver, además, una explicación detallada de su funcionamiento.



INSERTANTO CÓDIGO:

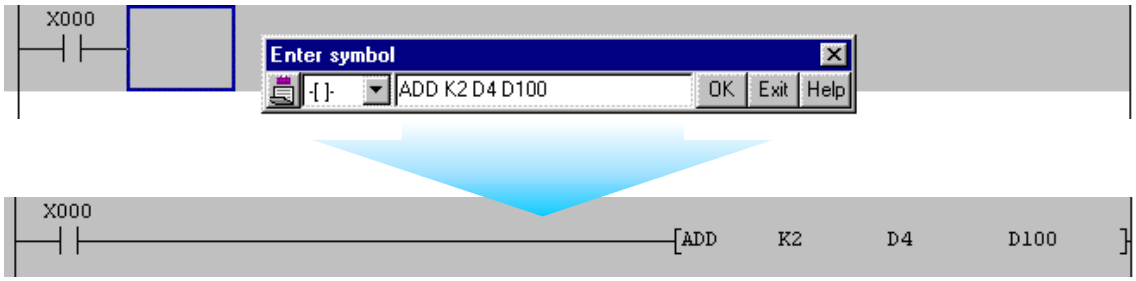
Se va ha seguir un ejemplo para ver que pasos hay que seguir para editar circuito en el editor de programa:

1. Situar el cursor en la posición donde se quiere editar y clicar sobre el botón del objeto a insertar (también puede ser pulsada la tecla abreviada correspondiente). Aparecerá el formulario **Enter symbol**. Introducimos la referencia del dispositivo y pulsamos sobre **OK** (también la tecla RETURN puede ser pulsada para este fin).

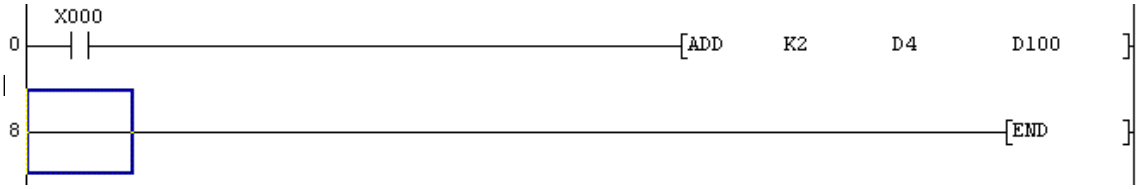


2. Una vez introducido el objeto, en nuestro caso un contacto de entrada (X0), aparece el cursor en la siguiente posición editable. Podemos ver que el fondo del editor donde se han introducido las nuevas líneas aparece de color gris. Esto quiere decir que el código introducido aún no ha sido convertido a lenguaje de instrucciones para que pueda ser descargado al autómata. Por lo que después de introducir varias líneas tenemos que ir convirtiendo el texto (se explica como hacer la conversión en el punto 4 de este ejemplo)

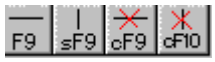
3. Si queremos seguir editando, solo tenemos que volver a insertar un nuevo objeto en la posición adecuada. En el siguiente ejemplo se explicará como insertar líneas de conexión entre contactos, bobinas e instrucciones. De momento en la posición actual del cursor, si introducimos una bobina de salida o una instrucción se dibujará la línea de conexión entre el contacto y la bobina/instrucción automáticamente.



4. Es en este punto donde podríamos ejecutar el comando **Convert** para convertir el programa creado hasta este momento. Esto puede ser conseguido pulsando sobre la opción del menú principal **Convert – convert**, clicando directamente sobre el botón con el símbolo  o también pulsando la tecla F4.



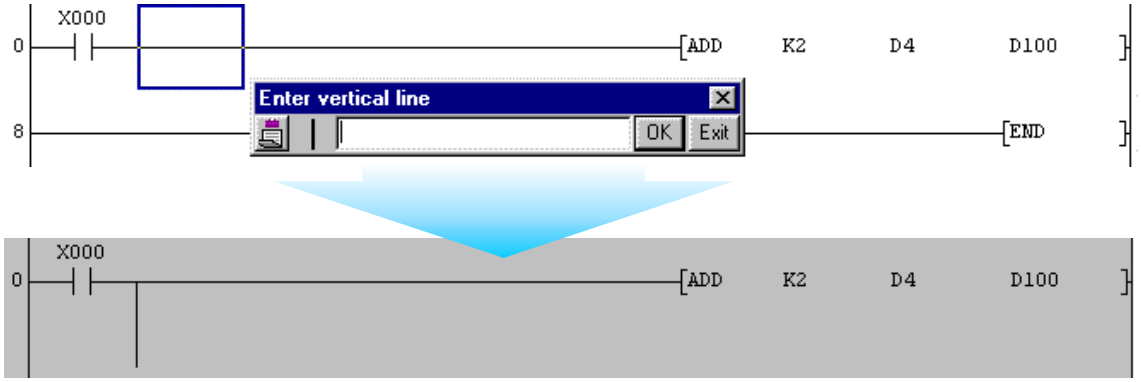
Se va ha seguir un ejemplo para ver que pasos hay que seguir para editar líneas de conexión en el editor de programa. Tenemos dos modos de inserción de líneas. Vamos a ver las dos:

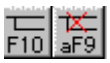
PRIMER MÉTODO: Utilización de las teclas 

Situando el cursor desde donde se quiera empezar el trazado de la línea de conexión y pulsando, por ejemplo, sobre el botón 

podremos pegar una línea vertical.

Aparece el formulario **Enter vertical line** que nos preguntará por la dimensión (en posiciones de cursor) de la línea a insertar. Si queremos insertar una línea de longitud uno, no hace falta insertar ningún valor en el formulario.

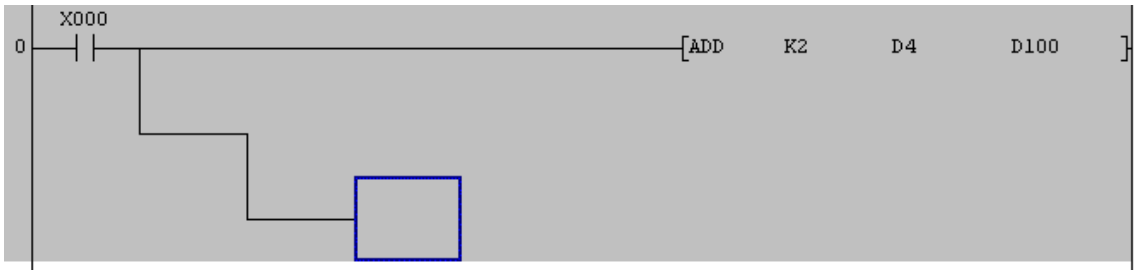


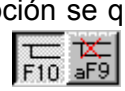
SEGUNDO MÉTODO: Utilización de las teclas 

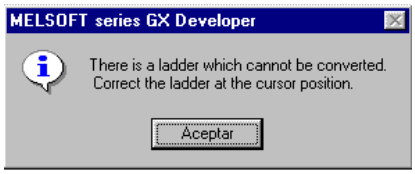
Si pulsamos sobre 

podemos editar líneas de conexión con solo pulsar el botón derecho del ratón y arrastrar el cursor. Pulsando la tecla SHIFT del teclado y, al mismo tiempo, utilizando las teclas del cursor podemos trazar líneas de conexión. Si por el contrario utilizamos la tecla 

borraremos las líneas creadas.



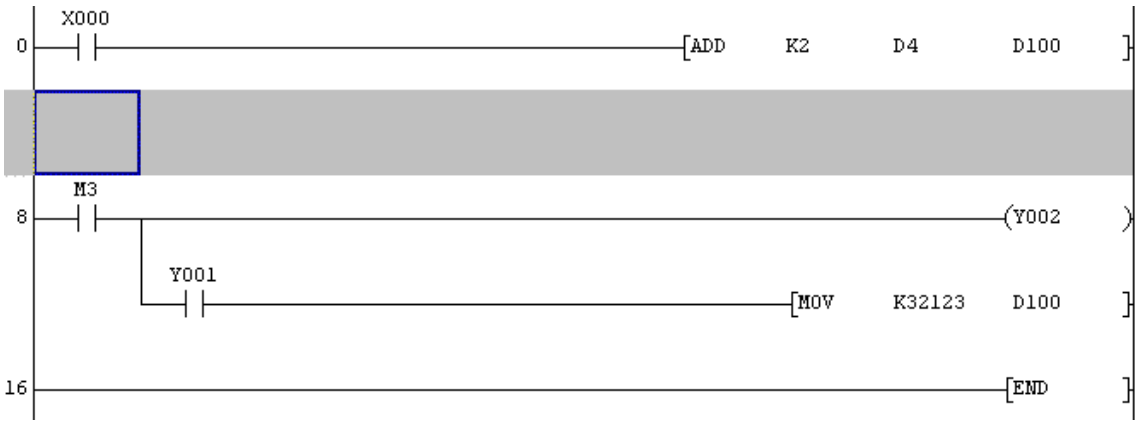
Cuidado porque esta opción se queda seleccionada hasta que se desactiva pulsando otra vez sobre el botón. 



ATENCIÓN: Si se crea una línea de conexión con contactos/bobinas o instrucciones que no pueden ser convertidas correctamente se mostrará el siguiente mensaje de error y no convertirá nada.

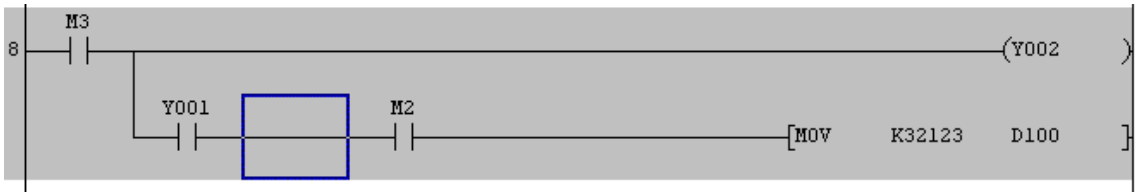
INSERTANTO / BORRANDO LÍNEAS DE CÓDIGO:

Si se quiere insertar una línea de código entre otras dos ya existentes, se debe situar el cursor sobre la línea que debe desplazarse hacia abajo y activar en el menú principal la opción **Edit – insert line** (teclas abreviadas SHIFT + Insertar).



Si por el contrario, se quiere borrar una línea de código, solo hay que situar el cursor encima de la línea y activar en el menú principal la opción **Edit – delete line** (teclas abreviadas SHIFT + Suprimir).

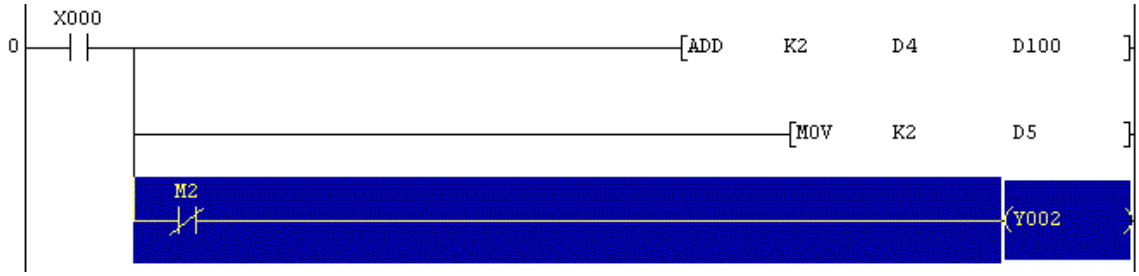
A través del mismo procedimiento que se ha utilizado para insertar líneas de código, pueden ser insertadas columnas. En este caso en **Edit – insert row**.



Si se quieren editar más líneas de código desde la instrucción END, no es necesario desplazarla hacia abajo, automáticamente se insertará una línea.

BORRANDO PARTES DE CÓDIGO:

También se tiene la posibilidad de borrar trozos de código, contactos, bobinas e instrucciones por separado. Solo hay que seleccionar el fragmento a borrar y pulsar la tecla Suprimir del teclado.

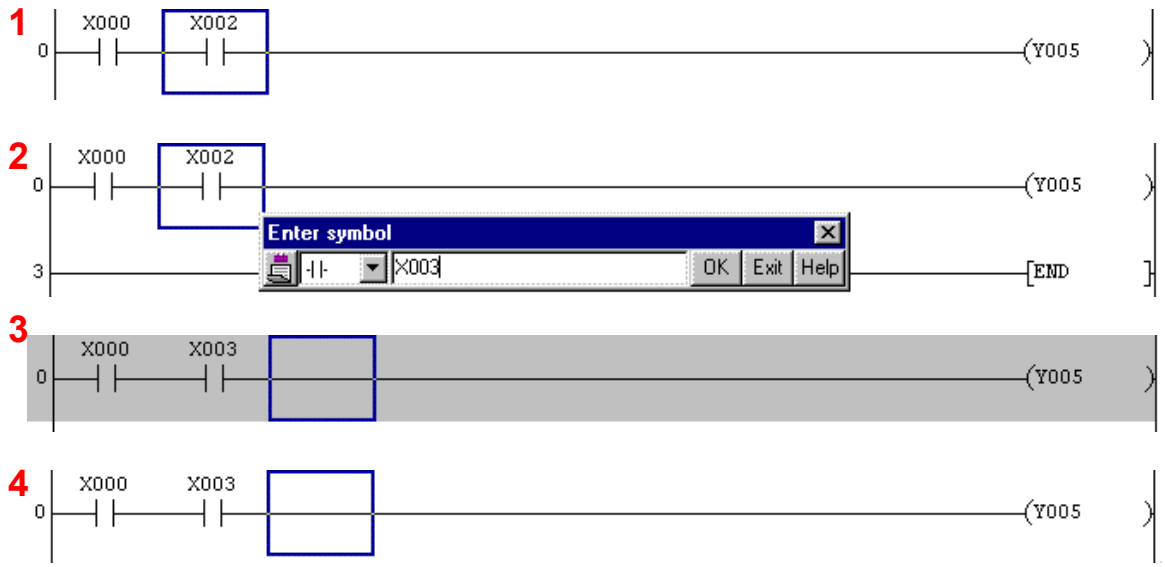


INSERTANDO / SOBREESCRIBIENDO OBJETOS:

Cuando en una línea de código hay, por ejemplo, dos contactos seguidos y queremos escribir otro nuevo en medio de los dos, tenemos que poner el modo edición en Insertar Insert (pulsando la tecla Insertar del teclado). El cursor cambia su color actual a violeta. El estado normal es Sobre-escritura Ovrwrte (ha este modo se puede volver pulsando otra vez la tecla Insertar). El cursor es de color azul en este modo de edición.

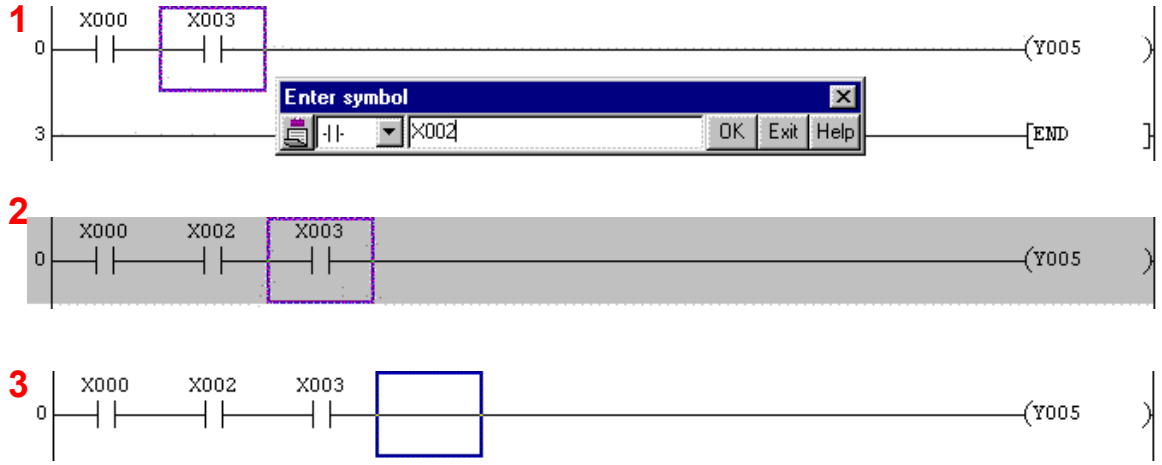
EJEMPLO CON MODO SOBREESCRIBIR ACTIVADO:

Se pretende cambiar el nombre del dispositivo de un contacto. También se podría cambiar por ejemplo un contacto normalmente abierto por otro normalmente cerrado.



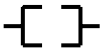
EJEMPLO CON MODO INSERCIÓN ACTIVADO:

Se pretende insertar un nuevo contacto en medio de otros dos.

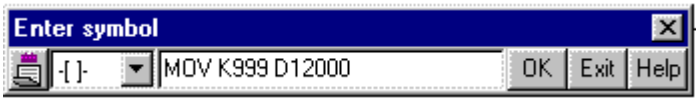


SÍMBOLOS UTILIZADOS POR GX DEVELOPER:

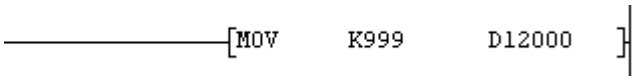
Los dispositivos (contactos, bobinas…) que pueden ser insertados en la pantalla de edición de código del GX Developer son los siguientes: (Se especifica también su función y las teclas de inserción rápida)

	CONTACTO NORMALMENTE ABIERTO	F5
	CONTACTO NORMALMENTE ABIERTO EN PARALELO	SHIFT + F5
	CONTACTO NORMALMENTE CERRADO	F6
	CONTACTO NORMALMENTE CERRADO EN PARALELO	SHIFT + F6
	CONTACTO EN FLANCO ASCENDENTE	SHIFT + F7
	CONTACTO PARALELO EN FLANCO ASCENDENTE	ALT + F7
	CONTACTO EN FLANCO DESCENDENTE	SHIFT + F8
	CONTACTO PARALELO EN FLANCO DESCENDENTE	ALT + F8
	BOBINA DE SALIDA	F7
	BLOQUE PARA INSTRUCCIONES	F8
	TRAZAR RAMA HORIZONTAL	F9
	BORRAR RAMA HORIZONTAL	CTRL + F9
	TRAZAR RAMA VERTICAL	SHIFT + F9
	BORRAR RAMA VERTICAL	CTRL + F10
	NEGACIÓN DE LA LÓGICA PREVIA	CTRL + ALT + F10
	TRAZAR RAMAS CON CURSORES	F10
	BORRAR TRAZADO DE RAMAS CON CURSORES	ALT + F9
	ACTIVACIÓN POR FLANCO DESCENDENTE	CTRL + ALT + F5
	ACTIVACIÓN POR FLANCO ASCENDENTE	ALT + F5

Cuando hay que insertar una instrucción en el editor, hay que activar el símbolo correspondiente a la tecla F8 (**-[ ]-**). Después hay que teclear la instrucción seguida de todos los parámetros uno detrás de otro separándolos utilizando espacios en blanco.

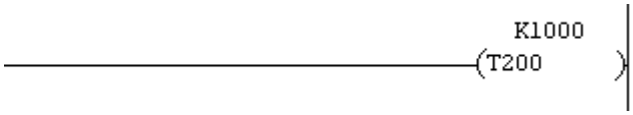


Si en la instrucción se aplican valores constantes, es obligatoria la inclusión de la letra **K** delante si es un valor decimal (K999 en el ejemplo) o **H** si es un valor hexadecimal. En todos los parámetros es necesaria la inclusión de una letra delante de cualquier número. No existen parámetros sin letra de referencia delante, como puede ser ADD D10 8 D11 (en este caso tendríamos ADD D10 K8 D11).



Cuando se utiliza un contador o temporizador, hay que especificar la cuenta máxima del dispositivo y se utiliza como si fuera una bobina (**-( )**), símbolo que corresponde con la tecla F7. En el siguiente ejemplo se puede ver la programación de un temporizador:

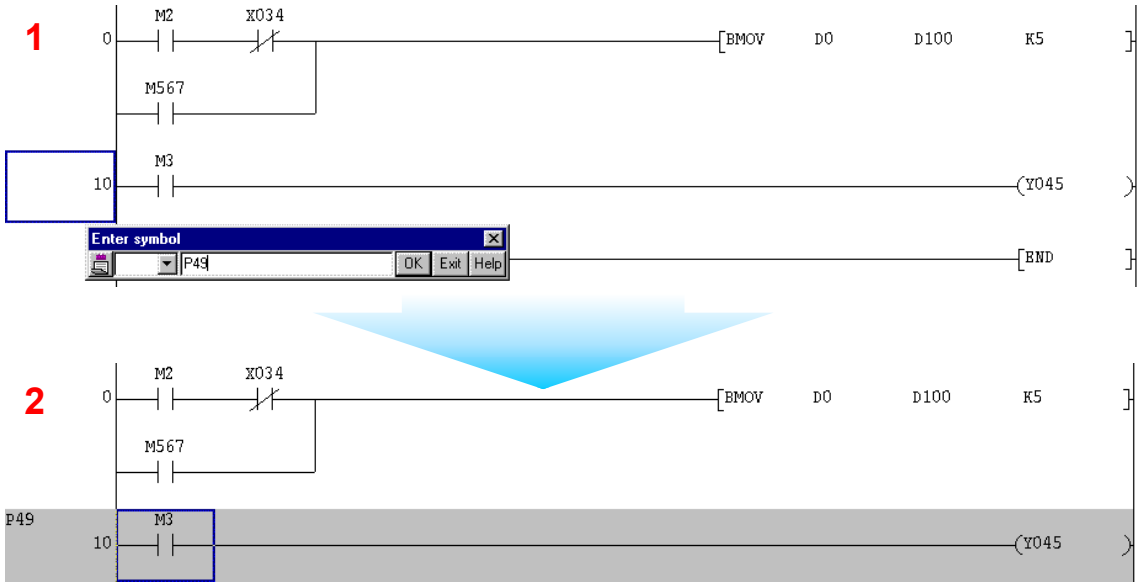
Una vez confirmado pulsando la tecla OK, el resultado es el siguiente:



EDICIÓN DE PUNTEROS P/I:

Si es necesaria la inserción de punteros (etiquetas de código) en el programa para poder efectuar saltos condicionales o subrutinas, así como punteros de interrupción, para crear subrutinas de interrupción, se identificarán con las letras **P** e **I** seguido del número de etiqueta (siguiendo el estándar de programación de Mitsubishi Electric).

Estas etiquetas se insertan en la parte izquierda de la pantalla de edición de código. Si se quiere insertar el puntero P49 en nuestro programa, debemos situar el cursor en la línea adecuada, siempre a la izquierda, he inmediatamente tecleamos una letra P o una letra I según sea el caso:



DOCUMENTANDO EL PROGRAMA:

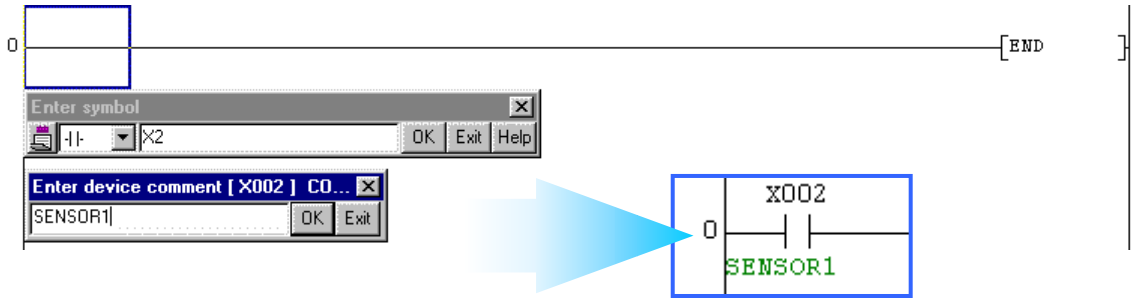
Para documentar el programa tenemos varias posibilidades. La inclusión de comentarios y alias asignados a dispositivos, frases (statements) y notas referidas a instrucciones son las posibilidades que nos ofrece GX Developer para documentar el programa.

EDICIÓN DE COMENTARIOS:

Los comentarios son, como su nombre indica, una breve descripción de cada dispositivo. Pueden ser mostrados por pantalla, para ayudar a interpretar el programa. El procedimiento para editar los comentarios es a través del **Project data list** generando un archivo **Device comment**, o insertándolo directamente cuando se añade cada objeto (contactos, bobinas…) desde el formulario **Enter symbol** que aparece cuando se edita un nuevo objeto (en este caso se actualiza automáticamente el archivo Device comment).

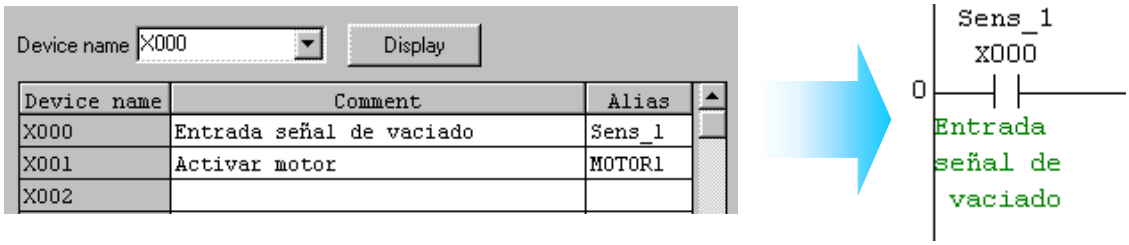
Para realizar la edición de comentarios directamente cada vez que se edita el dispositivo hay que tener activada la opción **Tools-Options…**, después en la solapa **program common** en el apartado **comment input**, activar **continues during command write**.

Así cada vez que introducimos un dispositivo en la pantalla de edición tendremos la posibilidad de teclear el comentario:



Desde el **Project Data List** podemos acceder al editor de comentarios y alias de los dispositivos de memoria del autómata.

Device comment se utilizará para insertar comentarios a los dispositivos (D, M, X, Y…), así como un alias para asociar a cada dispositivo. (Un alias es un nombre de 8 letras que se puede asignar a un dispositivo para que, durante la programación se muestre conjuntamente con el nombre del dispositivo).



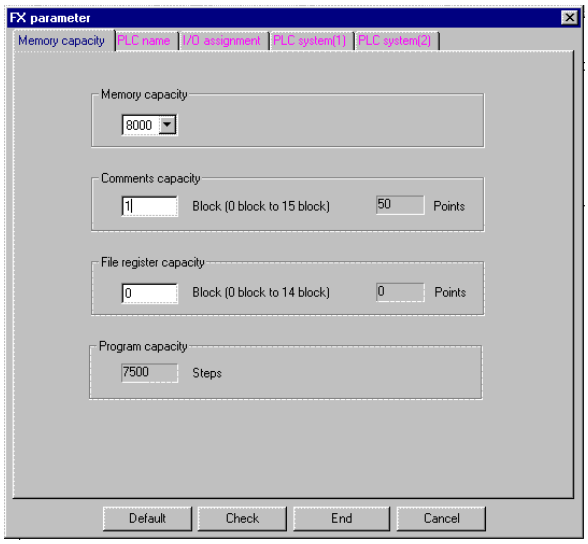
Cuando se transfiere el programa al PLC debe activarse en el menú Online – Write to PLC… la opción Device comment (COMMENT). De esta manera podemos descargar al autómata el archivo de comentarios. Con la serie Q estos archivos se crean automáticamente en la memoria del PLC, pero en la serie FX debe ser configurada la memoria del autómata mediante los parámetros:

Utilizando la sección **Memory capacity** del **PLC Parameter** (desde el **Data project list**) podemos configurar la memoria de un autómata de la serie FX. Por ejemplo, en un FX1N tenemos inicialmente 8000 pasos de programa, los cuales pueden ser recortados para añadir comentarios al proyecto y que éstos queden almacenados en el PLC.

EJEMPLO:

En este ejemplo, del total de memoria del PLC (8000 pasos iniciales), hemos configurado un bloque para comentarios, con esto conseguimos poder comentar 50 contactos/bobinas. La memoria para programa se reduce 500 pasos por cada bloque de 50 comentarios que es asignado.

Se pueden configurar un máximo de 15 bloques de comentario (por lo tanto 750 comentarios para dispositivos pueden ser insertados al PLC, disponiendo en este caso de 500 pasos de programa)

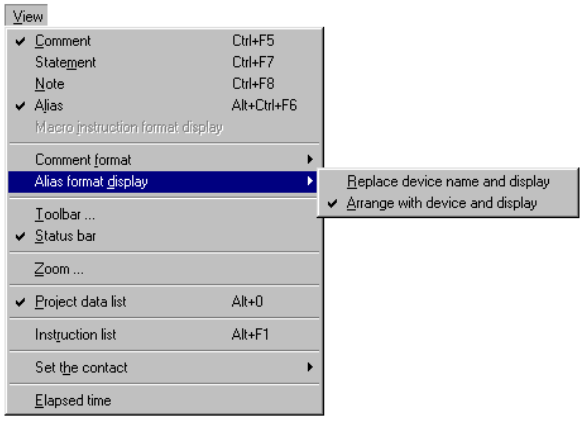


La visualización de los comentarios y los alias en la pantalla de edición del programa no será posible a no ser que estén activadas las siguientes opciones:

View–Comment: el editor se ensancha y aparecen los comentarios.

View–Alias: Aparecen los alias en pantalla.

Hay dos modos de visualización de los alias: **Replace device name and display** (el nombre del alias sustituye el nombre del dispositivo) y **Arrange with device and display** (se muestran los dos en pantalla).

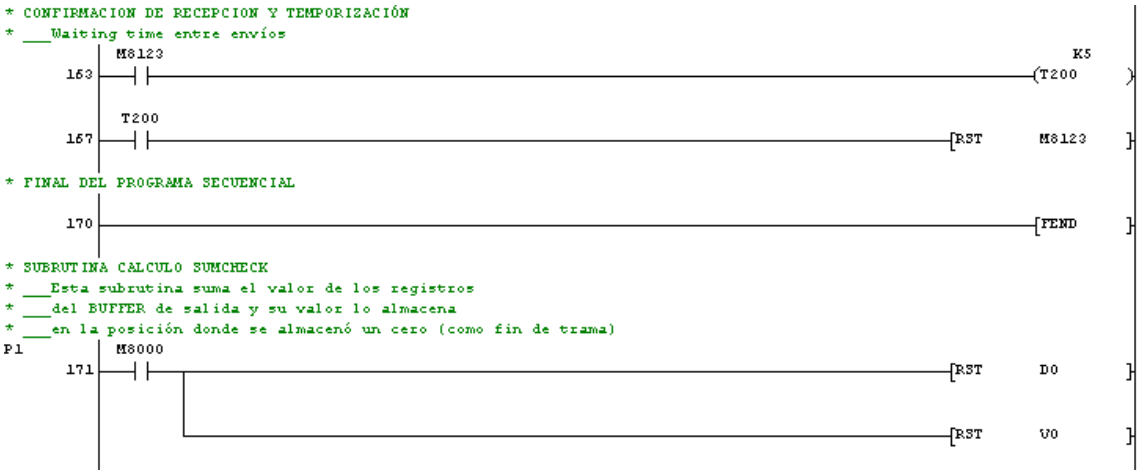


Cuando se acaba de diseñar y editar un proyecto, pueden haber varios comentarios que pertenecían a variables que ya no se van ha utilizar, por lo tanto puede ser interesante borrar todos los comentarios que no se van ha utilizar. Esto se puede hacer clicando sobre **Tools – Delete unused comments**.

EDICIÓN DE ETIQUETAS (statements) Y NOTAS (notes):

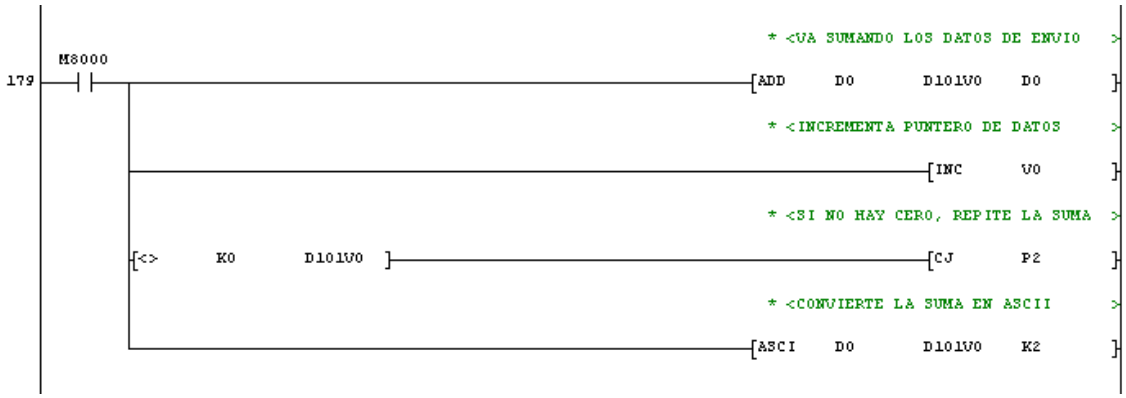
Una Etiqueta o **statement** servirá para anotar en cada bloque de programa algunos comentarios para asegurar una comprensión rápida de todo el programa, permitiendo separar las diferentes partes de que consta el proyecto. (Máximo 64 caracteres por etiqueta, pueden ser insertadas varias etiquetas consecutivas)

Ejemplo:



Las Notas o **notes** incluyen comentarios específicos para periféricos. Las notas pueden ser creadas para las bobinas de salida o instrucciones aplicadas a la derecha del ladder editado. (Máximo 32 caracteres por nota)

Ejemplo:



Para visualizar los statements y notes deben ser activadas estas opciones en el menú **View**:

Las etiquetas y notas pueden ser de tipo **Separate** o **Embedded**.

SEPARATE: Estas etiquetas/notas se añaden al GX Developer durante la edición, pero no se transfieren al PLC en el momento de la descarga de programa. Por lo que si llevamos a cabo cualquier edición sobre el programa con consola de programación o recuperación del programa desde el PLC para realizar cambios, sin tener el proyecto creado con el software en un archivo guardado perderemos los comentarios que hacen referencia a etiquetas y notas. Solo se recuperará el código de programa.

Cuando se escribe el texto se selecciona la opción **Separate** en el formulario **Enter line statements**. En el editor aparecerá el statement/note con un asterisco delante.

EMBEDDED: Este tipo de statements/notes será almacenado al autómata juntamente con el código de programa, por lo tanto, todo el proyecto se almacena realmente en el PLC, de manera que puede ser recuperado todo el conjunto de etiquetas/notas junto el programa que contiene un PLC.

Cuando se escribe el texto se selecciona la opción **Embedded** en el formulario **Enter line statements**. En el editor aparecerá el statement/note sin ningún símbolo adicional.

Este tipo de almacenamiento de etiquetas/notas en la memoria del autómata tiene como consecuencia un mayor consumo de memoria, por lo tanto se aconseja utilizar esta característica (**embedded**) en los PLC's cuya memoria es elevada. Los autómatas de las series **FX** y **A** trabajan solamente con el modo **Separate**. Las series **Q**/**QnA** pueden trabajar con los dos modos.

Cuando se utiliza texto en modo embedded, los pasos de programa utilizados se pueden calcular a través de la siguiente fórmula:

$$\text{STEPS} = 2 + \frac{\text{Num. caracteres}}{2}$$

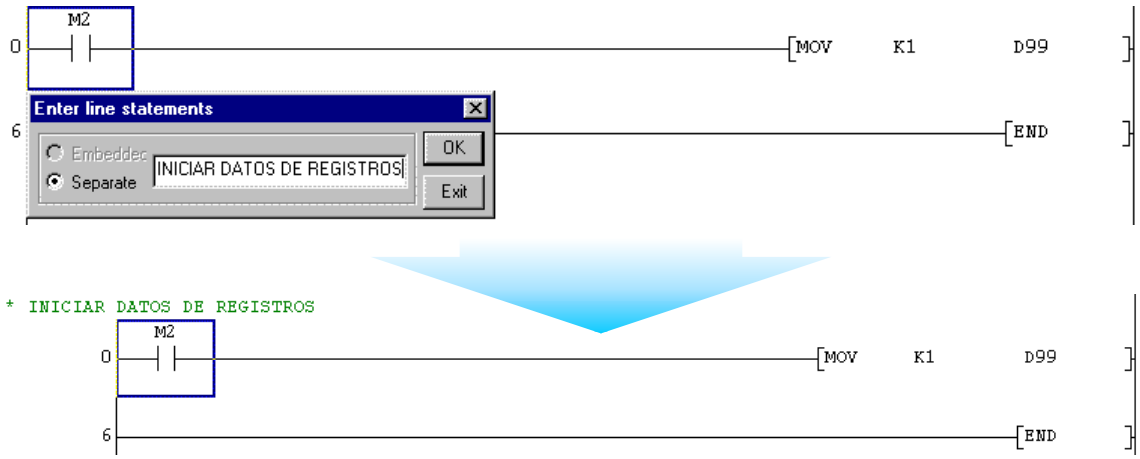
Donde el número de caracteres es el total de letras que forman parte del texto de la etiqueta/nota (los espacios en blanco también cuentan como caracteres). El número resultante (STEPS) se redondea hacia arriba.

INSERTANDO ETIQUETAS Y NOTAS:

Para insertar una etiqueta al programa deben utilizarse los botones de la barra de herramientas o en el menú principal activar **Edit – Documentation – Statement**, o **Edit – Documentation – Note**.

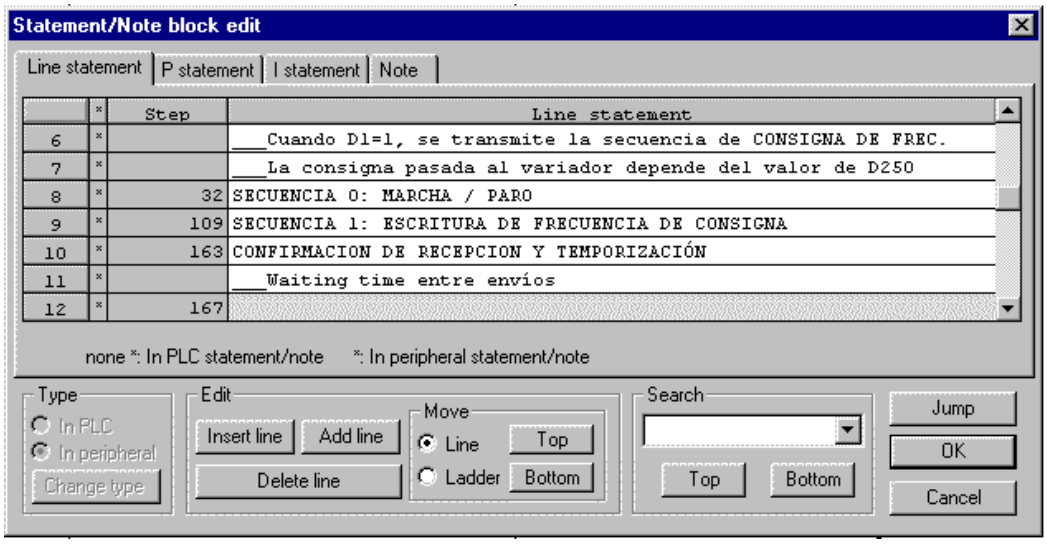
STATEMENTS **NOTES**

Cuando está activado el botón de inserción de etiquetas al clicar 2 veces sobre un objeto de la pantalla de edición, aparece el formulario **Enter line statements** comentado en la página anterior (**Enter note** en el caso de las notas).



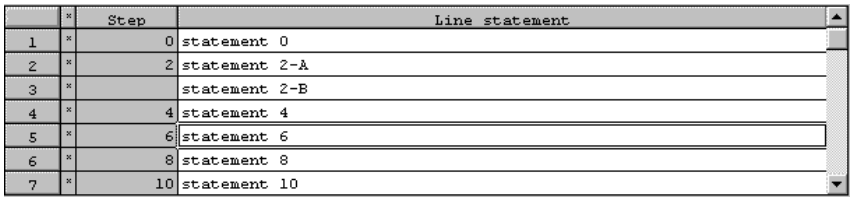
BATCH-EDITING:

Si entramos en el formulario **Statement / Note block edit** utilizando desde el menú principal la opción **Edit – Documentation – Statement/Note block edit….** (Acceso directo mediante la combinación de teclas Alt+E y seguidamente 0 y luego A). Pueden ser editadas las etiquetas y notas directamente desde un formulario dedicado. Además desde esta pantalla se puede saltar a cualquier parte del programa señalando la etiqueta correspondiente y pulsando sobre el botón **Jump**.



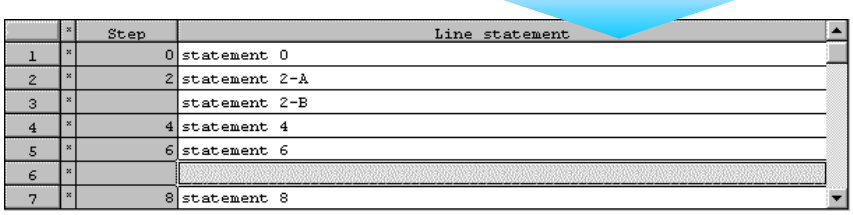
Si utilizamos los botones de la sección **Edit**, podemos insertar líneas nuevas entre la statement/note actual y la posición inmediatamente superior (**Insert line**) o añadir una nueva línea entre la posición del cursor y la línea inferior (**Add line**), también tenemos la posibilidad de borrar líneas (**Delete line**).

Ejemplo para añadir un statement nuevo: Situar el cursor en la posición a la cual debemos añadir una nueva etiqueta. Después pulsar sobre Add line.



		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2-A
3	*		statement 2-B
4	*	4	statement 4
5	*	6	statement 6
6	*	8	statement 8
7	*	10	statement 10

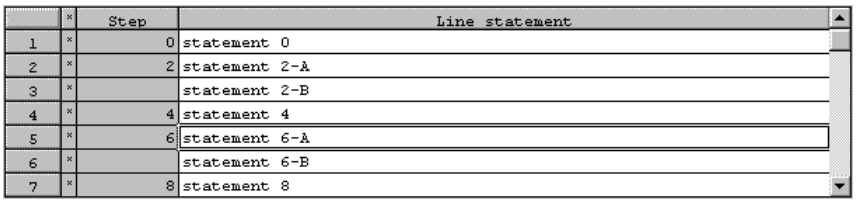
El resultado será el siguiente:



		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2-A
3	*		statement 2-B
4	*	4	statement 4
5	*	6	statement 6
6	*		
7	*	8	statement 8

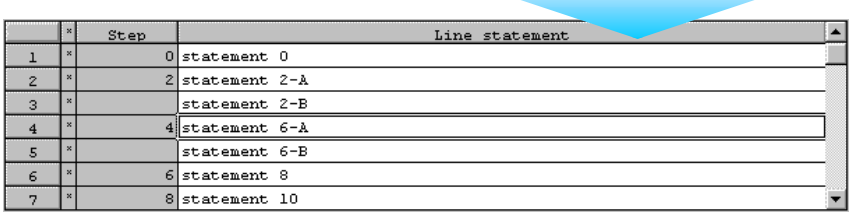
Dentro de la sección **Edit**, encontramos también la sub-sección **Move** que nos ayudará a reorganizar las etiquetas y las notas en el caso de haber llevado a cabo un **merge** (se explicará en líneas posteriores). Un merge, a modo de introducción, sirve para mezclar programas de diferentes archivos o combinar por ejemplo los comentarios/statements/notes de un programa guardado en disco con el programa que se está editando en pantalla (Esto puede ser útil cuando se utilizan etiquetas y notas de tipo **separated**, que no se almacenan en el PLC, y sin embargo se han realizado cambios online en el autómata que está trabajando en modo RUN).

Si las etiquetas/notas están desplazadas hacia abajo, por ejemplo, y queremos desplazarlas todas hacia arriba. (es decir, las etiquetas que corresponden con el step de programa 4 están almacenadas en el step 6 y necesitamos eliminar las etiquetas del step 4 para ser substituidas por las del step 6). Situar el cursor en la posición del step 6 y se desplaza hacia arriba pulsando sobre el botón **Top** de la sección **Move** con la opción **Ladder** activada.



		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2-A
3	*		statement 2-B
4	*	4	statement 4
5	*	6	statement 6-A
6	*		statement 6-B
7	*	8	statement 8

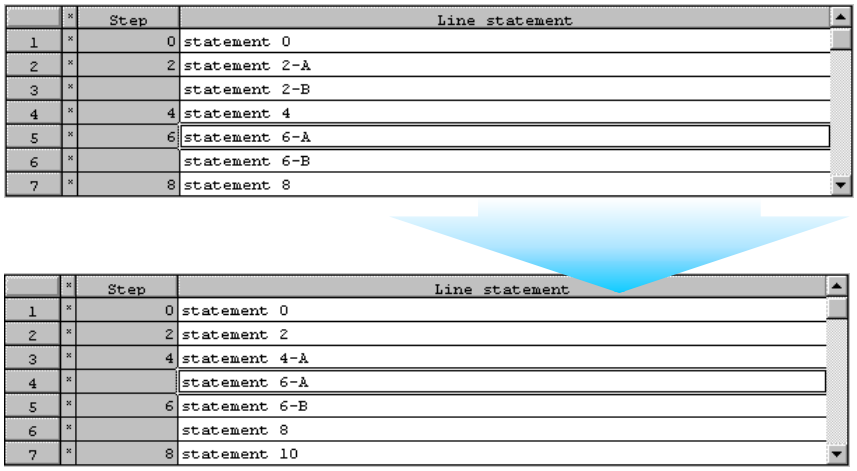
El resultado será el siguiente:



		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2-A
3	*		statement 2-B
4	*	4	statement 6-A
5	*		statement 6-B
6	*	6	statement 8
7	*	8	statement 10

Si por el contrario queremos desplazar las etiquetas/notas sin tener en cuenta los steps de programa, podemos hacerlo utilizando la opción **Line** dentro de la sección **Move**, a través de los botones **Top** y **Bottom**. Con esto borramos la línea anterior o posterior a la que está situado el cursor (según botón utilizado) y se desplazan todas las etiquetas/notas.

En el siguiente ejemplo se sitúa el cursor sobre el comentario “statement 6-A” y se activa el botón **Top** con la opción **Line** activada. Se desplazarán todos las etiquetas una posición hacia arriba, borrando la etiqueta “statement 4-B”



		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2-A
3	*		statement 2-B
4	*	4	statement 4
5	*	6	statement 6-A
6	*		statement 6-B
7	*	8	statement 8

		Step	Line statement
1	*	0	statement 0
2	*	2	statement 2
3	*	4	statement 4-A
4	*		statement 6-A
5	*	6	statement 6-B
6	*		statement 8
7	*	8	statement 10

MODOS DE FUNCIONAMIENTO: 

Estos 4 botones marcan el funcionamiento del entorno de programación. Siempre debe estar uno de estos botones activado.

 **Modo lectura**

 **Modo edición**

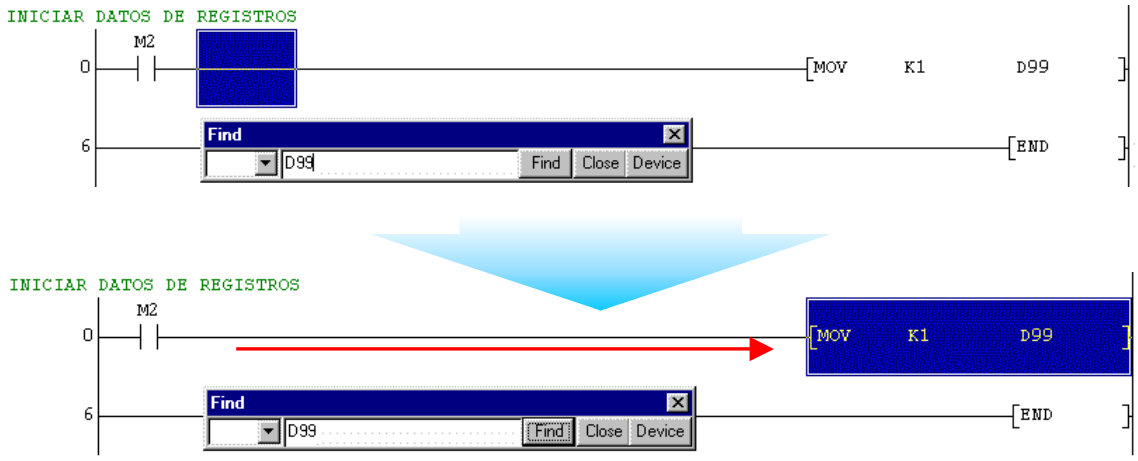
 **Modo Monitor**

 **Modo edición Online**

Dependiendo del estado de estos botones, el GX Developer necesitará estar conectado al autómata para poder intercambiar información con la CPU del PLC.

A continuación se detalla la función de cada uno de estos modos de funcionamiento:

MODO LECTURA: Trabajando en este modo activado, el programa en pantalla no puede ser editado. Cuando hacemos la acción de editar, en lugar de aparecer el formulario **Enter symbol**, aparece el formulario **Find**. Esto puede ser utilizado para buscar un dispositivo concreto. Por ejemplo, en el siguiente caso si tecleamos D99 en el cuadro de texto **Find**, nos buscará en que partes del programa se encuentra el dispositivo especificado.

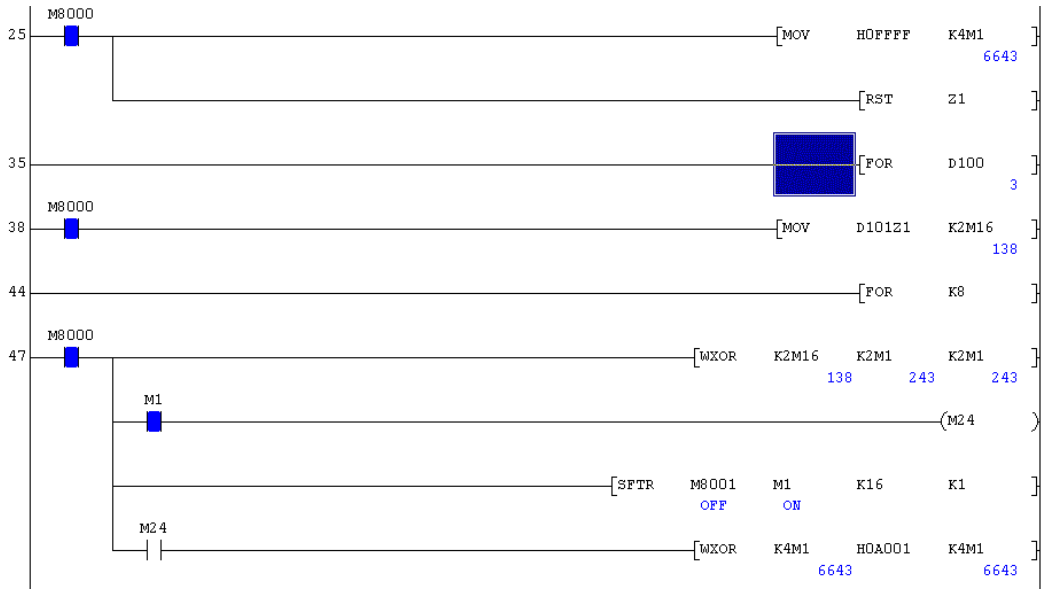


Si no queremos seguir buscando tenemos que salir del formulario activando el botón **Close**.

MODO EDICIÓN: Este es el modo de funcionamiento con el que se ha trabajado hasta ahora en todas las explicaciones que se han dado. Es el modo de edición de programa, a través del cual se crean los comentarios e instrucciones del programa.

MODO MONITOR: Este modo es como el modo de lectura, pero necesita tener el autómata conectado para monitorizar el programa, ya que esta es la función principal de este modo de trabajo, el monitorizado de los datos del PLC. El autómata debe estar en modo RUN.

Cuando se activa el modo monitor la pantalla tiene el siguiente aspecto:

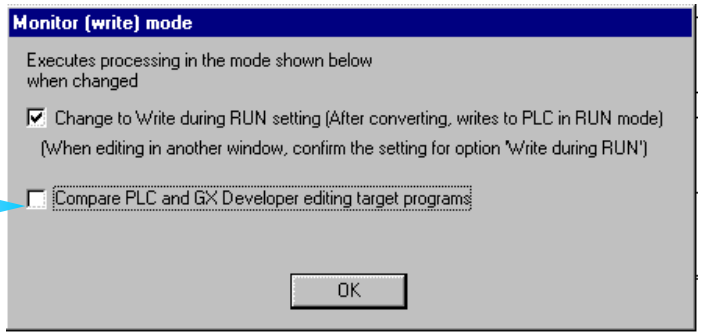


En la pantalla aparece información del estado del autómata, y del tiempo medio actual del ciclo de scan del programa ejecutandose en el PLC.

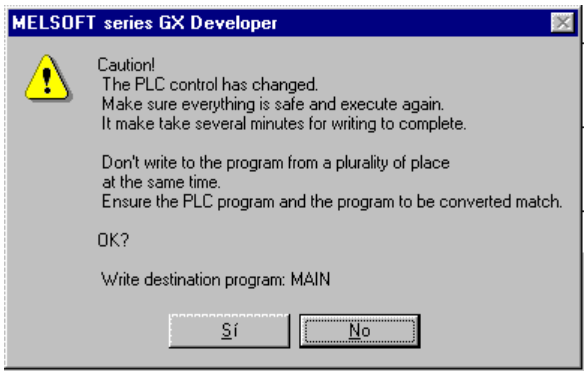
MODO EDICIÓN ONLINE: Este modo de funcionamiento permite editar programa, de igual manera que lo hacemos con el modo edición. La diferencia es que los cambios realizados sobre el programa en el momento de la conversión (recordar el procedimiento para convertir programa) se descargan al autómata instantáneamente. El autómata puede estar en modo RUN, por lo que los cambios se llevan a cabo "on line".

Cuando activamos el modo Edición Online, se nos da la posibilidad de verificar el programa (comparar el programa, que se está ejecutando en el PLC, con el que tenemos actualmente en pantalla), esto es aconsejable si no estamos seguros de que los programas sean iguales, ya que sino el cambio "on line" puede escribir código de programa en posiciones de memoria del autómata que no corresponden con el programa en pantalla!!. La opción verificar está activada por defecto.

DESACTIVAR ESTA OPCIÓN SI NO SE QUIERE VERIFICAR



Cuando convertimos un fragmento de programa para realizar un cambio "on line" aparece un formulario advirtiéndonos de lo peligroso que esto puede ser.



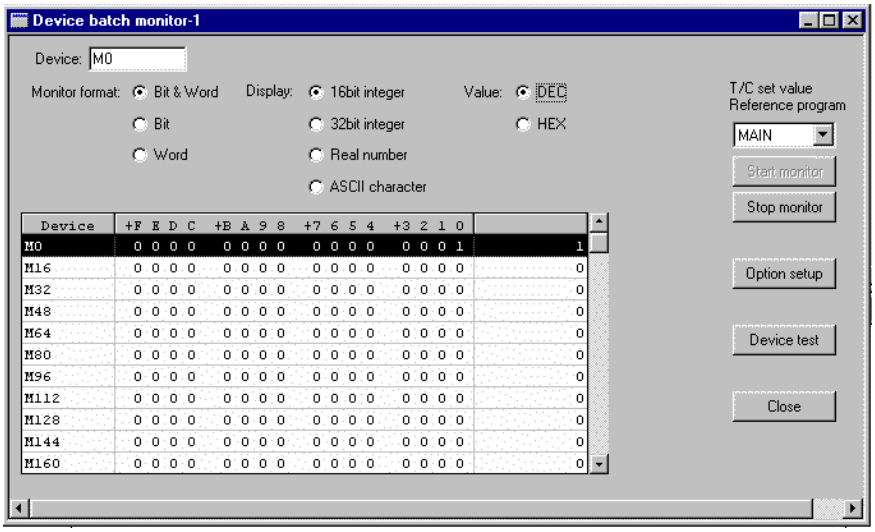
Cuando se finaliza el proceso de escritura al autómata se muestra un mensaje que nos informa de que ya se ha finalizado. Solo hay que pulsar Aceptar.

MODO MONITOR (ventanas):

Se explica a continuación que posibilidades tenemos disponibles para el monitorizado de variables, cuando el autómata está en modo RUN. Básicamente disponemos de cuatro pantallas con las que podemos acceder a todo el rango de memoria del PLC. Estas pantallas sólo están disponibles desde el modo MONITOR y el modo EDICIÓN ONLINE, explicados anteriormente. Son las siguientes:

Device batch monitor
Entry data monitor
Buffer memory batch
Device test

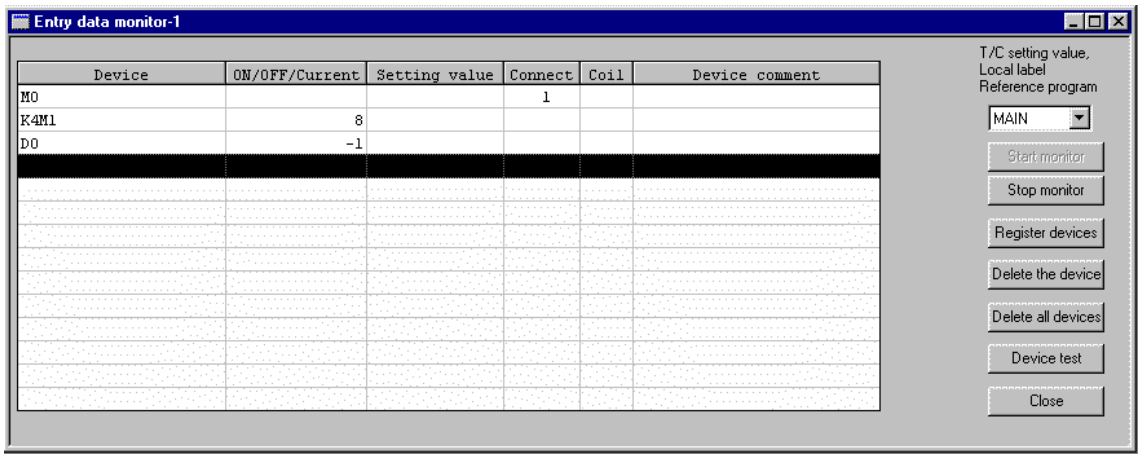
DEVICE BATCH MONITOR: Desde esta pantalla, que se muestra en la figura siguiente, hay que introducir el valor de la primera dirección de los dispositivos que se quieren mostrar. Por ejemplo, en la figura se ha introducido en el cuadro de texto **Device** el valor M0. Con esto, cuando pulsamos sobre **Start monitor** serán visualizadas las variables de manera consecutiva. Se puede configurar, a través de las opciones disponibles en **Monitor format**, el formato de visualización de los datos que queremos visualizar. Por ejemplo, si visualizamos datos D (16 bits) podemos verlos en formato hexadecimal, decimal, binario o incluso los caracteres ASCII correspondientes.



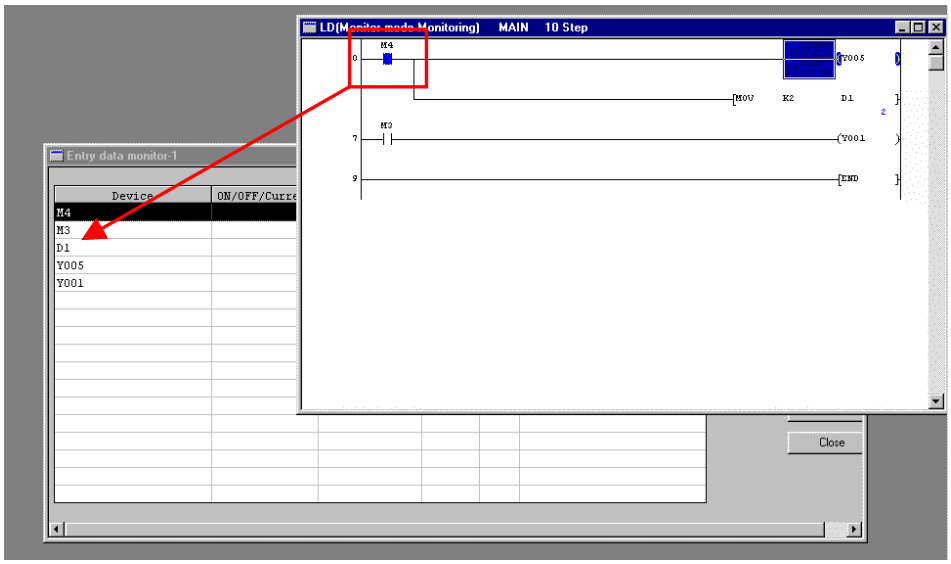
Desde este formulario tenemos acceso a la pantalla **Device test** pulsando sobre el botón correspondiente (**Device test**). Este formulario de test es útil para cambiar valores en las variables del autómata, para poder ver los efectos que estos cambios puedan tener sobre la ejecución del programa. Este nuevo formulario se explicará después de explicar la pantalla **Entry device monitor**.

ENTRY DEVICE MONITOR: Este formulario es similar al explicado anteriormente, pero en este caso tenemos que ir introduciendo el nombre de todas las variables que queremos visualizar. Mientras que en **Device batch monitor** se visualiza todo el rango disponible dentro del formulario desde la variable introducida en **Device**.

Desde este formulario diponemos de unos botones de edición como **Register devices**, **Delete the device** y **Delete all devices**. El primer botón servirá para introducir una nueva variable a leer (se consigue el mismo efecto haciendo doble click sobre la tabla donde se visualizan las variables). El segundo botón, **Delete the device**, borrará la variable que esté seleccionada en la tabla (de color negro). El último de estos botones, **Delete all devices**, borrará todo el contenido de la tabla, es decir, todas las variables.

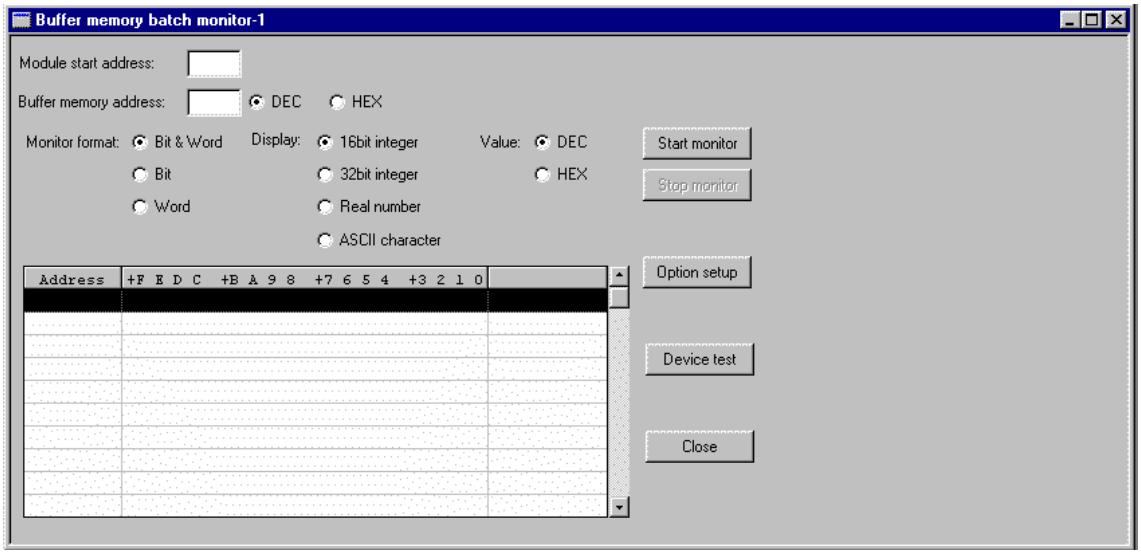


Para registrar variables pueden ser arrastrados y pegados los objetos desde la pantalla de edición hacia la pantalla **Entry data monitor**. Esto se puede hacer cuando se está en modo lectura o modo monitor. Se pulsa el botón izquierdo del ratón con el cursor situado encima del objeto y, sin soltar el botón, arrastrar el objeto hasta una posición de la tabla de **Entry data monitor**.



Este formulario también dispone del botón de acceso a **Device test**. Se explica a continuación.

BUFFER MEMORY BATCH: Desde este formulario se pueden hacer las mismas tareas que en el caso del **Device batch monitor**, pero esta vez podemos ver las variables BFM de los módulos especiales de función. Se tienen las mismas opciones, pero para especificar una variable tenemos dos campos de texto para rellenar. En el cuadro **Module start address** hay que especificar la posición del módulo y en **Buffer memory address** hay que indicar la dirección del primer registro BFM en el que queremos comenzar la visualización en la tabla.

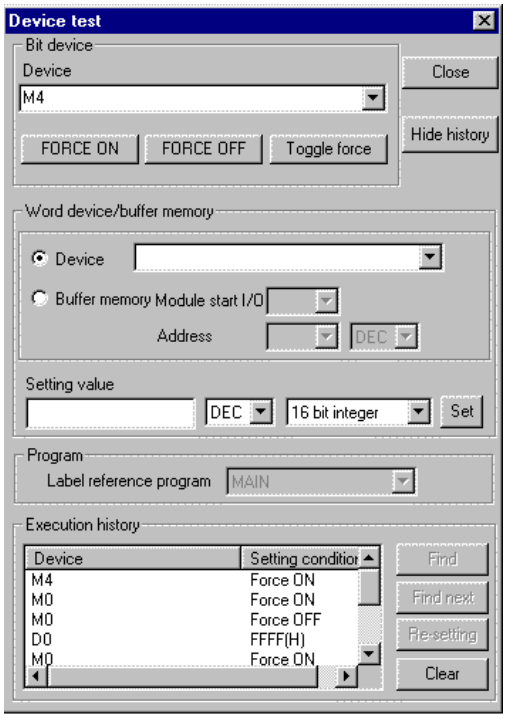


También, desde **Buffer memory batch**, tenemos acceso al formulario **Device test**. Se explica a continuación.

DEVICE TEST: A este formulario se puede acceder desde el menú principal directamente o a través de **Device batch monitor**, **Buffer memory batch** y **Device Entry monitor**. Desde aquí tenemos la posibilidad de cambiar valores de variables (M, D, X, Y…)

Tenemos la posibilidad de cambiar valores numéricos o de contactos/bobinas. En la sección **Bit device**, disponemos del cuadro de texto **Device** donde insertamos el nombre de la variable binaria a modificar. Luego, a través de los botones **FORCE ON**, **FORCE OFF** y **Toggle force**, podemos cambiar su estado a 1 (activo), 0 (inactivo) o conmutar su valor actual respectivamente.

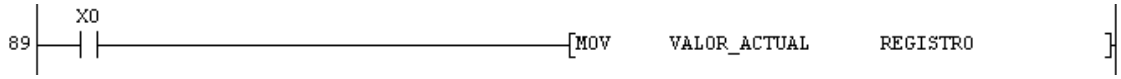
En la sección **Word device/buffer memory** podemos cambiar el valor de una variable D (16 o 32 bits) seleccionando en el cuadro **Device** que dispositivo queremos modificar. En **Setting value** introducimos el valor (en formato decimal o hexadecimal según selección en las opciones disponibles a la derecha del cuadro de texto **Setting value**). Cuando se introduce el valor deseado hay que pulsar sobre el botón **Set**.



La sección **Word device/buffer memory** también nos permite modificar el valor de un BFM (Buffer memory) de un módulo que tengamos instalado en el autómata. Si activamos **Buffer memory Module start I/O** en lugar de **Device**, debemos seleccionar la posición del módulo al que accedemos y seguidamente la dirección BFM dentro de ese módulo (en formato decimal o hexadecimal). El proceso para introducir el valor a cambiar es el mismo que el explicado para cambiar un valor de variable normal.

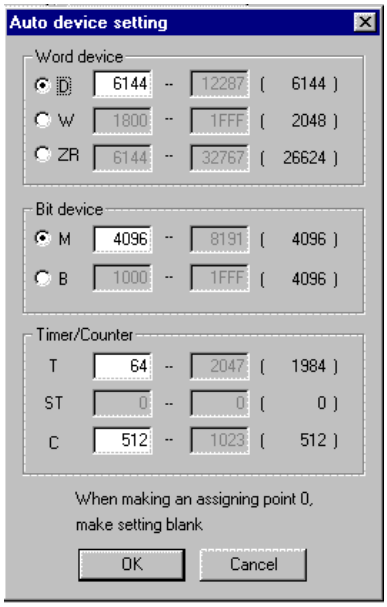
EDITANDO PROGRAMA CON LABELS:

Con CPU's de las series Q/QnA (excepto Q00CPU y Q01CPU) puede ser editado el programa utilizando etiquetas de variable (labels). Siguiendo este tipo de edición se trabaja insertando en el programa nombres editados para las variables (no se utiliza normalmente el nombre habitual de las variables D100, M25…). Así pueden verse líneas de código con el siguiente aspecto:



Utilizando tablas como la siguiente pueden ser editados los nombres de las variables. Utilizando labels, no se asocia el nombre de la variable (label) a una variable concreta (como podría ser D100), simplemente se le indica al compilador del GX Developer que el valor de esta etiqueta corresponde, por ejemplo, a un número entero (INT). A través de este tipo de declaraciones se estandariza el programa y se deja al margen la necesidad de tener en mente que dispositivo (D, M…) se está utilizando. Se crean por lo tanto variables con nombre y tipo (INT, BOOL, ARRAYS…).

	Au	Label	Constant	Device type
1		VALOR_ACTUAL		INT
2		REGISTRO		INT

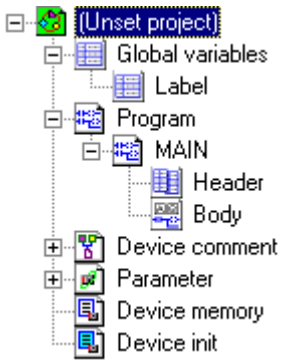


Las variables "reales" asignadas a las labels creadas en cada programa las asigna el GX Developer libremente. Hay un formulario desde el cual podemos especificar que rango de variables queremos que utilice el sistema como asignación automática (es el mostrado a la izquierda de este texto). Este formulario aparece cuando activamos la opción **Auto device Setting (L)…** desde el menú principal **Edit** (tiene que estar seleccionada anteriormente la opción **Global variables** del **Project data list**, sino esta opción no aparece en el menú **Edit**…).

En esta pantalla se nos da la oportunidad de modificar el rango de las variables utilizadas libremente por el software GX. Por ejemplo, en la figura hay configurado un rango de D6144 a D12287 para este tipo de datos, esto quiere decir que si creamos una variable de tipo entero (INT) se le asignará un dispositivo libre del rango especificado en **Auto device setting** (por ejemplo D6144).

De todas maneras pueden ser utilizados los dispositivos sin label, es decir, podemos escribir D100, M24, etc… (en este caso se consideran variables globales, se explicará en líneas posteriores).

El **Project data list** pasa a tener el aspecto que aparece a la derecha de estas líneas. Cada programa que forma parte del proyecto tiene dos secciones el **HEADER** (cabecera) y **BODY** (cuerpo). En el **Header** se definen las variables locales del programa y en el **Body** se crea el código de programa utilizando las labels creadas en la cabecera.

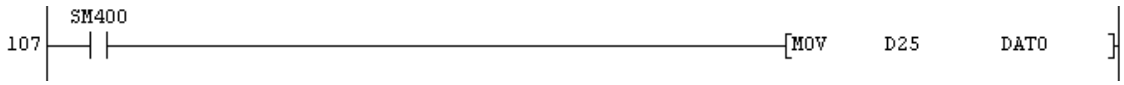


En la sección del **Project data list** está el formulario **Label** dentro de **Global variables**. Esta pantalla servirá para editar las variables globales (el formulario es muy similar al header que se utiliza para cada programa). Pero desde aquí se edita el nombre de la label, el tipo de variable y también se le asigna una variable concreta (D49, X2…).

	Au	Label	Device/Constant	Device type
1	*	VARIABLE1	D10	INT

VARIABLES LOCALES Y GLOBALES: Las variables **LOCALES** son las que se asignan a un programa (o bloque de función) y sólo pueden ser tratadas desde dentro del programa o función. Mientras que las variables definidas como **GLOBALES** pueden ser vistas y modificadas por cualquier programa o función de un proyecto. Por lo tanto, hay que definir como globales todas aquellas variables de uso común en el proyecto y como locales las que no deben ser utilizadas por otros programas exteriores al programa o función donde se han declarado.

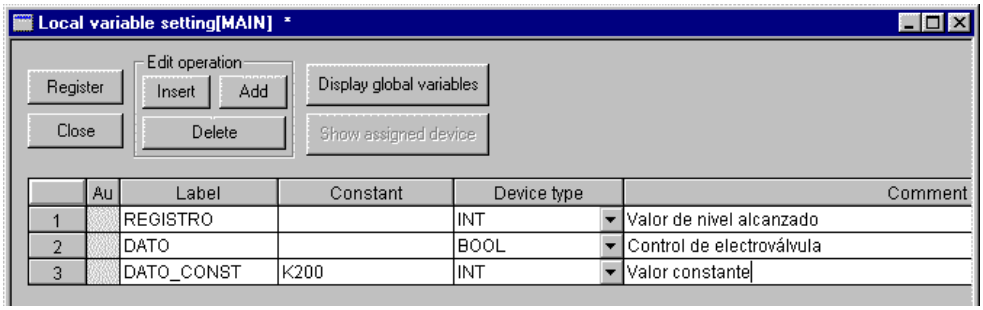
Hay que tener en cuenta que cuando una variable se define como global, se pide la identificación del dispositivo a tratar (D12, D78, M3…). Estos dispositivos tienen que estar dentro del rango especificado como de uso general en el formulario **Auto device setting** (D0 a D6143 en el formulario mostrado en la página anterior para datos de tipo D). Si se utilizan variables directamente sin label, por ejemplo:



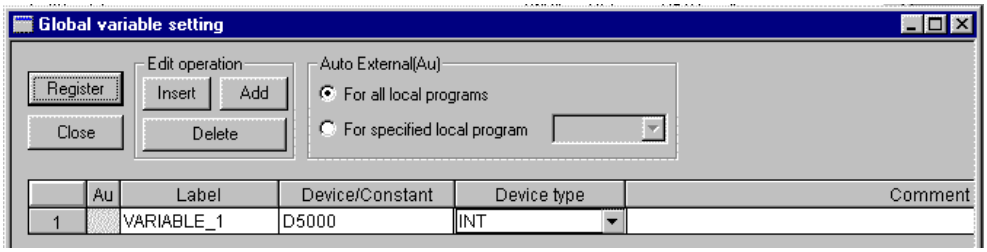
En este caso D25 es una variable global, por lo tanto si se utiliza D25 desde otro programa del proyecto será modificado su valor (es accesible en todo el proyecto).

HEADER O CABECERA DE PROGRAMA:

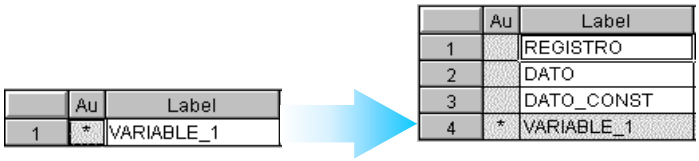
Desde esta pantalla se configuran las etiquetas de cada programa, el formulario que aparece es el que se muestra a continuación.



Los botones **Insert**, **Add** y **Delete** de la sección **Edit operation** sirven para insertar, añadir y borrar labels respectivamente. El botón **Register** servirá para que el GX Developer compile el header y asigne dispositivos a las variables creadas. El asterisco que aparece en el título de la ventana indica que no está compilado el header (por lo tanto, las variables no pueden ser utilizadas en el body del programa hasta que no sea compilado). Si es asignado un valor en el campo Constant, cada vez que se utilice la label correspondiente en el body del programa, el GX Developer no asignará una variable, utilizará en su lugar la constante.



La lista de variables globales (**Global variable setting**) es igual que la utilizada en el header de cada programa o función. La diferencia es la comentada en líneas anteriores (la inclusión de un campo donde debe especificarse el dispositivo concreto a utilizar, **Device/Constant**). También se dispone de un campo **Au,** desde el cual podemos decir al software que incluya esta label en todos los headers de los programas que se utilicen en el proyecto (se activa clicando sobre este campo para que aparezca un asterisco). Si después se visualizan los headers, las variables globales que tienen el campo **Au** activado aparecen como se muestra a continuación:

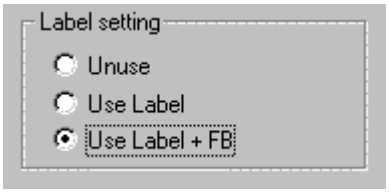


Evidentemnte, el header debe ser programado (y compilado) antes de la generación de código en el body de programa, para que puedan ser interpretadas las labels creadas. Se pueden insertar en la edición del body etiquetas (labels) no creadas con un header, pero en el momento de convertir (convert) recibiremos un mensaje de error.

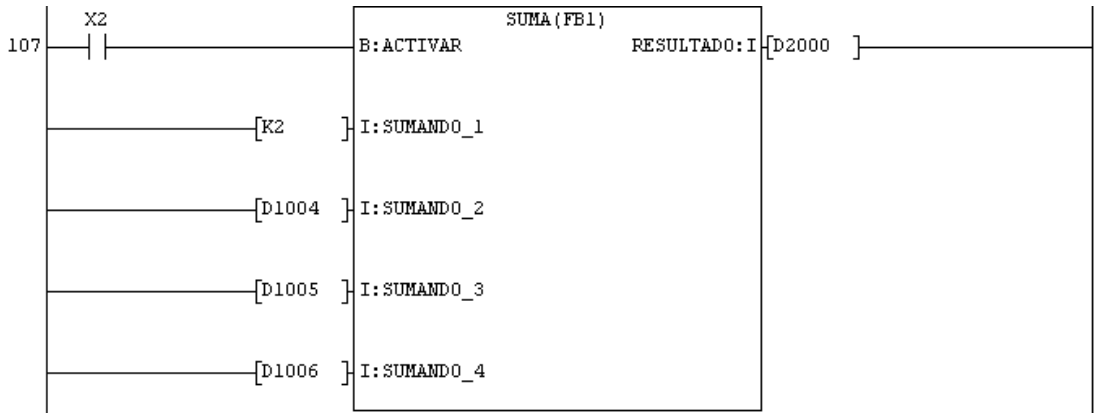
EDITANDO BLOQUES DE FUNCIÓN:

Los bloques de función trabajan con CPU's de las series Q/QnA (excepto Q00CPU y Q01CPU) ya que éstas son las CPU's que pueden trabajar utilizando labels (explicado en el apartado anterior), y el uso de labels para programar utilizando bloques de función es obligatorio.

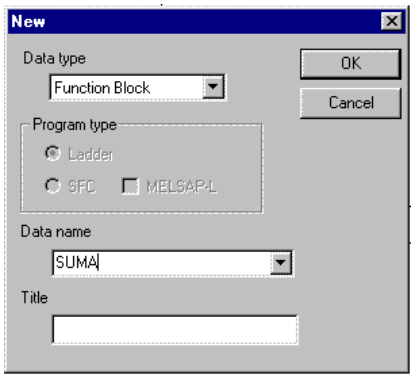
En el formulario **New project** cuando se crea un nuevo proyecto, se nos da la posibilidad de escoger **Use Label + FB** en la sección **Label setting**. Esto hace que en el **Project data list** aparezca una nueva pestaña para los bloques de función.



Cuando se crea una función se inserta en el programa editado y tiene el siguiente aspecto: (en este ejemplo se ha creado un FB que se llama SUMA)

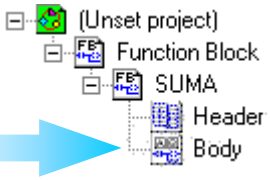


CREACIÓN DE UN BLOQUE DE FUNCIÓN (FB):



Situando el ratón en el **Project data list**, pulsar el botón derecho del mouse, luego escoger la opción **New…** (o a través del menú principal **Project**, activar la opción **Edit Data-New…**). Aparece el siguiente formulario desde el cual hay que seleccionar **Function Block** en la sección **Data type**. Una vez hecho esto, se da un nombre a la función a crear en **Data name** (en nuestro caso es SUMA). Pulsar **OK** al finalizar. Se nos preguntará si realmente queremos crear la función, respondemos que sí y ya podemos empezar a editar.

Si activamos ahora la lengüeta **FB** en el **Project data list** podremos ver lo siguiente. Podemos editar la función igual que se ha explicado en el apartado de programación por labels.



HEADER DE UN BLOQUE DE FUNCIÓN:

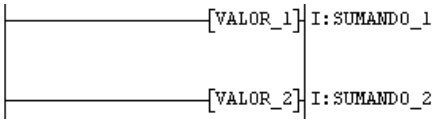
La cabecera de un bloque de función es igual que en el caso de la programación por etiquetas, pero esta vez tenemos un campo adicional **Input/Output** desde el cual especificamos si la variable actual es de entrada a la función (VAR_INPUT), de salida (VAR_OUTPUT), o es una variable que entra a la función y modificará su valor dentro de ésta (VAR_IN_OUT). También puede dejarse este campo en blanco para indicar al compilador del GX Developer que esa variable es local.

	Input/Output	Label	Constant	Device type
1	VAR_INPUT	ACTIVAR		BOOL
2	VAR_INPUT	SUMANDO_1		INT
3	VAR_INPUT	SUMANDO_2		INT
4	VAR_INPUT	SUMANDO_3		INT
5	VAR_INPUT	SUMANDO_4		INT
6	VAR_OUTPUT	RESULTADO		INT

Una variable de tipo VAR_IN_OUT puede modificar su valor, esto quiere decir que si introducimos a la función, en tiempo de ejecución, una variable puede ser utilizada y modificada por la función (variable de entrada y salida). Mientras que si introducimos una variable a través de una entrada de tipo VAR_INPUT, aunque se modifique su valor dentro de la función, la variable que ha sido entrada no será modificada.

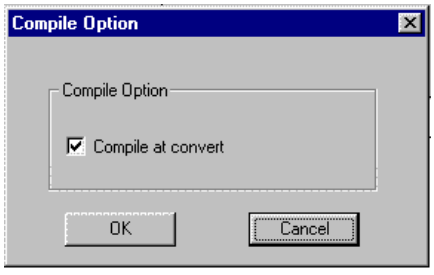
EJEMPLO:

El valor de la label VALOR_1 es copiado a la función como SUMANDO_1 (variable de tipo VAR_INPUT). Por lo tanto si se modifica SUMANDO_1, no se modifica el valor de VALOR_1.



Pero si SUMANDO_1 es de tipo VAR_IN_OUT, se tiene la posibilidad de recuperar el valor entrado porque la función nos permite extraer este valor nuevamente fuera de la función.

BODY DE UN BLOQUE DE FUNCIÓN:

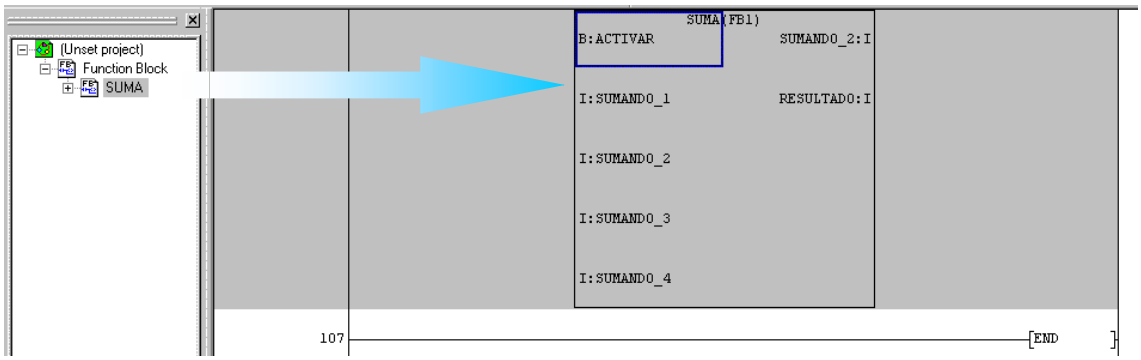


Cuando se genera código para un bloque de función, tiene que ser compilado para poder insertar la función a un programa. Esto se lleva a cabo desde el menú principal **Convert–Compile**. Tenemos la posibilidad de compilar el código al mismo tiempo que convertimos (**Convert**). Para hacer esto tiene que estar activada la opción **Compile at convert** en **Convert-Compile option…**

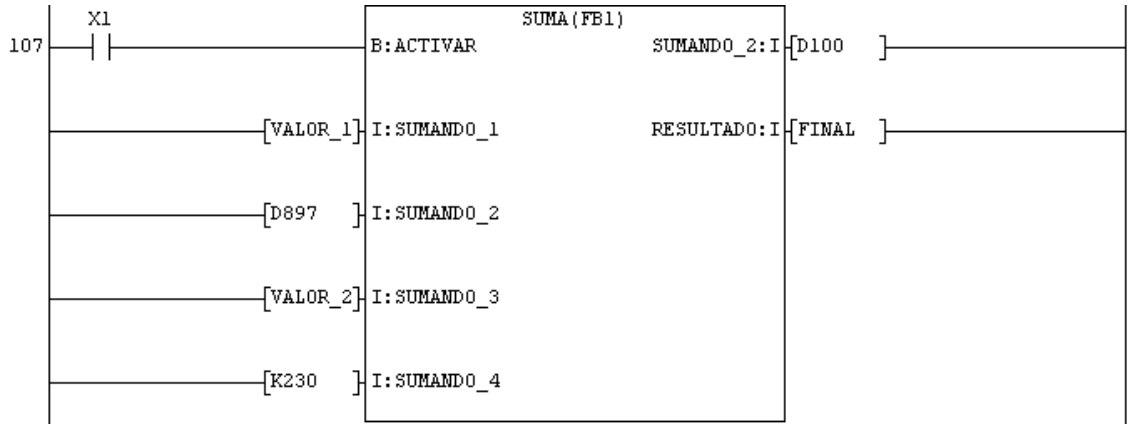
Por lo demás, la generación de código es idéntica a lo explicado en el apartado de programación con labels. Teniendo en cuenta que el programa debe ser compilado al finalizar.

INSERCIÓN DE UN BLOQUE DE FUNCIÓN (FB) EN EL PROGRAMA:

Cuando se está editando un programa y se quiere insertar una función creada, hay que seleccionar en el **Project data list** la lengüeta **FB** para visualizar los bloques de función. Después hay que arrastrar hacia el editor de programa la función que se quiere utilizar (pulsando con el botón izquierdo del mouse y soltando al llegar a la pantalla de edición de programa).



Seguidamente, se añaden las variables de entrada y salida al bloque de función. El código queda como se muestra en la siguiente figura:



Dentro del bloque de función mostrado en pantalla, se puede leer el nombre de las variables con su tipo, por ejemplo en B:ACTIVAR, se está dando la información de que la variable ACTIVAR es de tipo BOOL (lo mismo para el caso de I:SUMANDO_1, que nos dice que esta variable es de tipo INT).

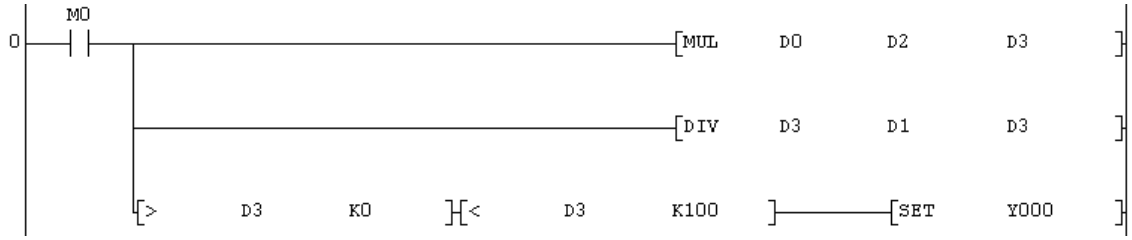
UTILIZACIÓN DE MACROS:

Con GX Developer tenemos la posibilidad de crear macros. A través de una macro, podemos guardar, dándole un nombre, un trozo de programa que se puede repetir en diferentes proyectos. De esta forma pueden ser creadas librerías de mini-programas creadas por el usuario. El "programa" almacenado en formato macro tiene la particularidad de poder ser salvado con dispositivos generales, esto permitirá, en el momento de la inserción de una macro en el proyecto actual, renombrar las variables para adaptarlas a la necesidad de nuestro nuevo programa.

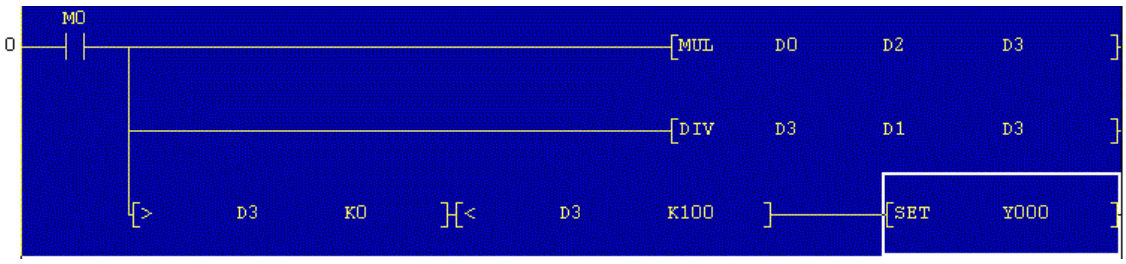
El siguiente ejemplo es un trozo de programa que hace un escalado de un dato y luego compara si el resultado está entre 0 y 100, si está dentro del rango se activará la salida Y0. Todo esto se ejecuta si está activada la marca M0.

$$y = \frac{a}{b}\,x \quad \Rightarrow \quad D3 = \frac{D0}{D1}\;D2$$

1. CREAR PROGRAMA: Se escribe el programa en el editor, con unas variables aleatorias, por ejemplo M0, Y0, D0, D1, D2 y D3.



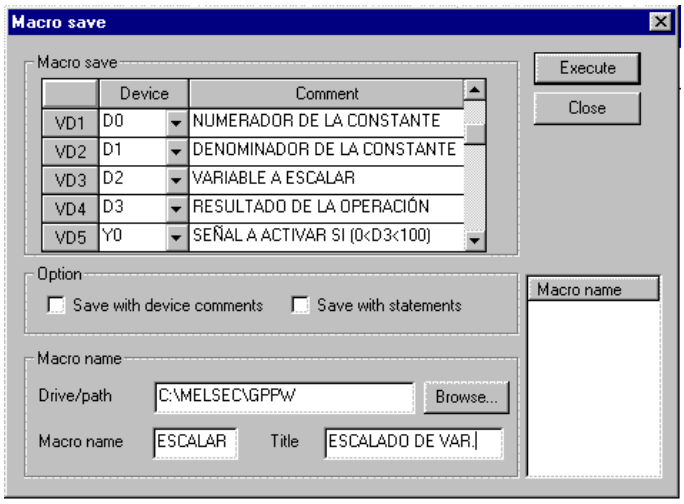
2. SELECCIONAR AREA: Se escoge la parte de programa con la cual se quiere crear la macro. Esto se consigue colocando el cursor el la esquina superior izquierda del programa y, pulsando el botón izquierdo del ratón, se arrastra el cursor hasta la esquina opuesta, hasta que todo quede sombreado.



3. REGISTRAR MACRO: En el menú **Project – Macro – Registration macros…** podemos registrar la macro.

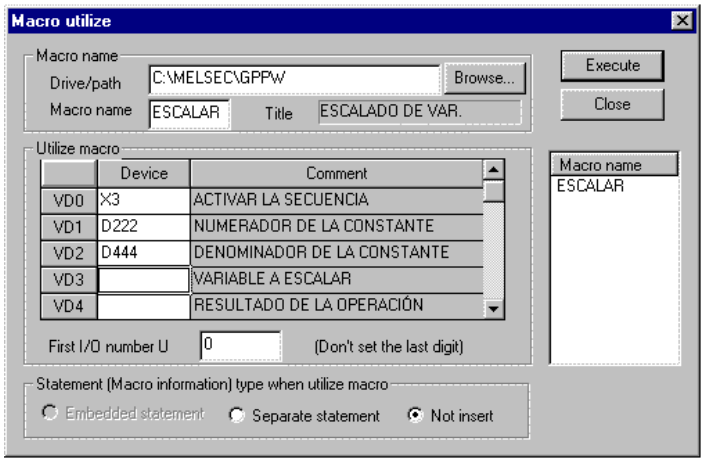
Aparecerá un formulario como el que se muestra en la páguina siguiente, en él pueden ser insertadas las variables que queremos que sean generales, o sea , que deben ser reutilizadas con otro nombre. En nuestro ejemplo asignaremos M0, Y0, D0, D1, D2 y D3 como variables respectivamente VD0, VD1, VD2, VD3, VD4 y VD5.

Podemos asignar a cada variable VD un dispositivo de los cuales queremos que en el momento de insertar la macro podamos cambiar su nombre. Por ejemplo para el denominador D1 le asignamos VD2, esto quiere decir que en el momento de la utilización de la macro se nos pedirá que introduzcamos un dispositivo para suplir a VD2, con lo que si introducimos en ese momento D9, el denominador pasará a ser el registro D9.



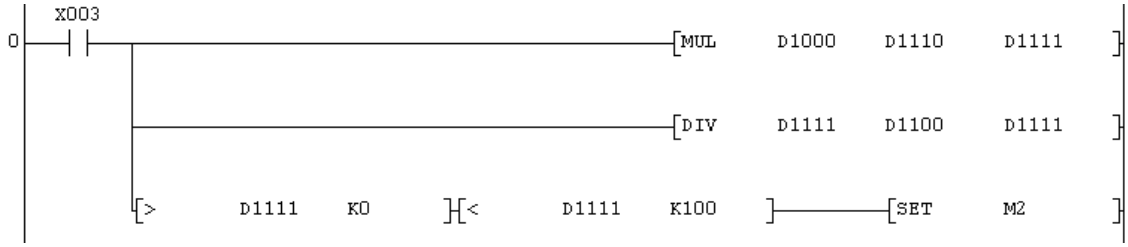
Una vez asignadas las variables, se procede a designar el nombre de la macro en **Macro name**, así como activar/desactivar las opciones de grabar junto con la información de la macro **statements** o comentarios de dispositivo (**device comments**). Después ejecutar **Execute** y la macro estará creada.

4. UTILIZAR MACRO: Para copiar la macro al proyecto actual se utilizará la opción del menú **Project – Macro- Macro utilize…**



Se desplegará un formulario en pantalla con el que podremos escoger la variables a utilizar y si queremos insertar los **statements**. La selección **First I/O number U** sirve para añadir un offset a las direcciones de **X** y de **Y**, en los PLC's modulares de las series A/Q.

Finalmente se insertará en nuestro nuevo proyecto el código generado por la macro, pero utilizando las nuevas variables:



OTRAS FUNCIONES:

GX SIMULATOR:

Si se instala el software de simulación de autómatas Mitsubishi Electric, podrá ser activada la opción **Start ladder logic test** del menú principal **Tools**. (También es posible activar el simulador pulsando sobre el botón 

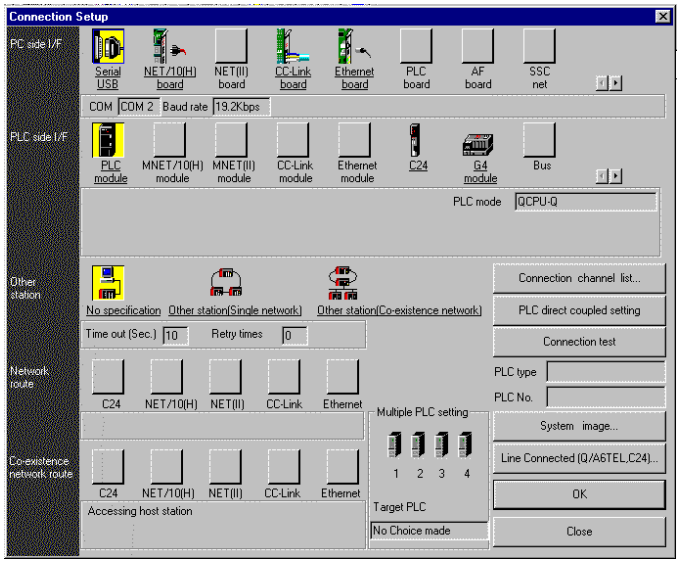
Desde el momento que se activa el software GX Simulator, la situación del GX Developer pasa a ser idéntica a cuando se está en modo monitor. Esto quiere decir que se pueden utilizar todas las herramientas disponibles en el software GX Developer para leer el estado de las variables, así como para modificarlas.

Cuando se activa la simulación no se pueden hacer cambios online. Si se quiere hacer algún cambio en el programa, hay que parar la simulación.

El paquete de software GX Simulator permite trabajar ejecutando el programa paso a paso.

CONFIGURACIÓN DE LA COMUNICACIÓN CON EL PLC:

Si activamos desde el menú principal **Online-Transfer setup…** aparecerá el formulario dedicado a la configuración de la comunicación entre el autómata y el ordenador desde donde se está ejecutando el GX Developer.



En este formulario aparecen distintas secciones:

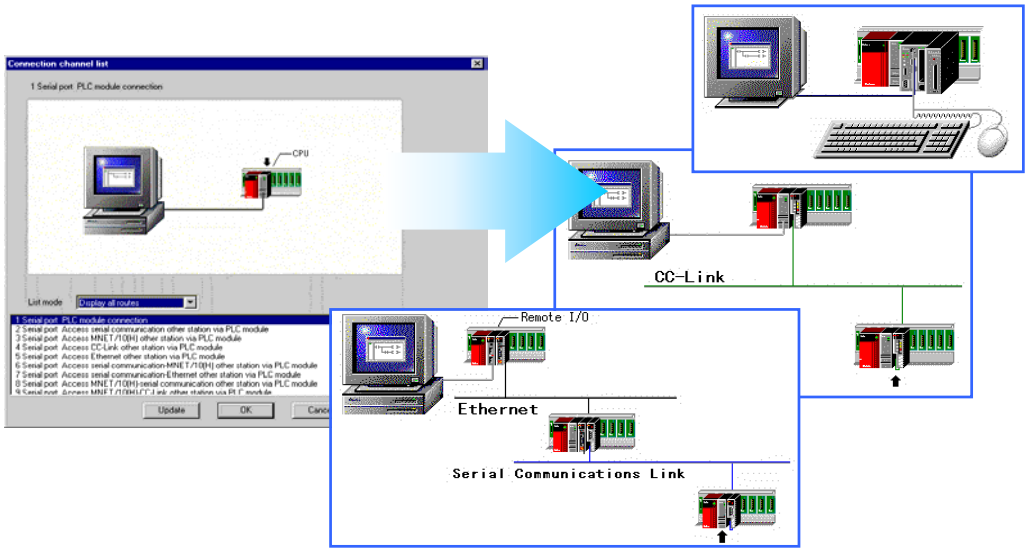
PC side I/F: configuración desde el punto de vista del PC.

PLC side I/F: configuración desde el punto de vista del PLC.

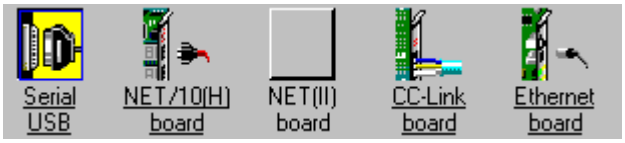
Other station: Sirve para configurar la comunicación con estaciones remotas conectadas a la estación con la que comunicamos directamente con el GX Developer.

Estas configuraciones que se llevan a cabo desde estas secciones, pueden hacerse a través del botón **Connection channel list**. En este formulario aparecen, de una forma totalmente gráfica, las configuraciones posibles desde la serie de autómatas con la que se está trabajando.

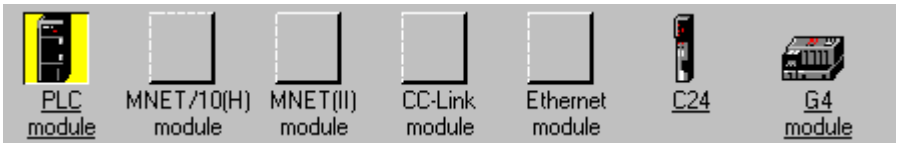
La página siguiente muestra el contenido de la pantalla **connection channel list**, donde pueden verse a través de :



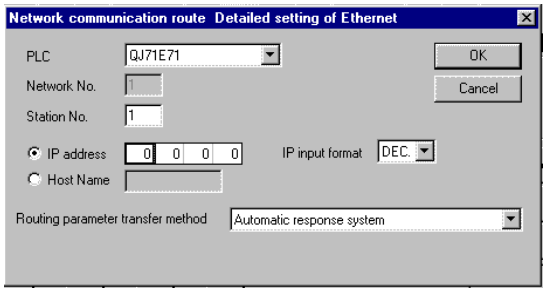
PC Side I/F: Aparecen todas las posibilidades de comunicación del software GX Developer según la serie de autómatas con la que se está trabajando. Por ejemplo, en la serie Q, podemos utilizar comunicación serie con RS-232 o puerto USB, red CC-Link o a través de Ethernet son algunas de las posibilidades que se nos ofrecen.



PLC Side I/F: En esta sección aparecen las opciones disponibles para el autómata que se utiliza en el proyecto. Hay que especificar si la conexión es directa al puerto de la CPU, o si se utiliza un módulo de comunicación serie (RS-232 o RS485/422), Un módulo para red Ethernet, CC-Link, etc…

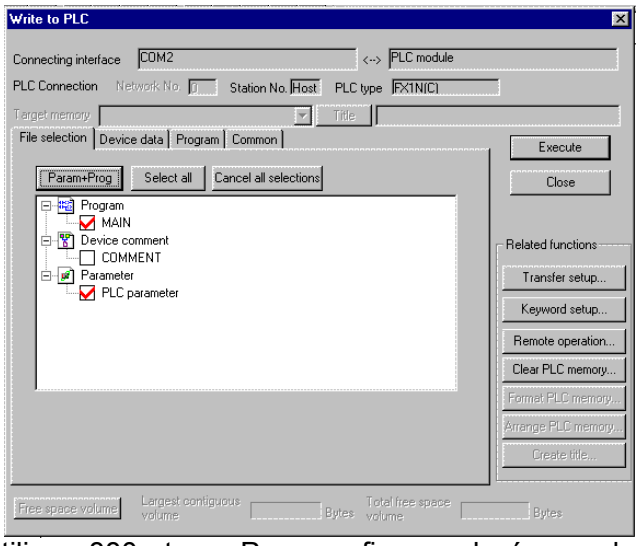


En PC/PLC Side I/F podemos configurar con más detalle el sistema de comunicaciones entre el ordenador y el autómata pulsando con el botón izquierdo del ratón sobre el icono con fondo de color amarillo (la opción seleccionada). Aparecen formularios similares al mostrado en la figura siguiente. La ventana que se muestra depende del tipo de comunicación a configurar:

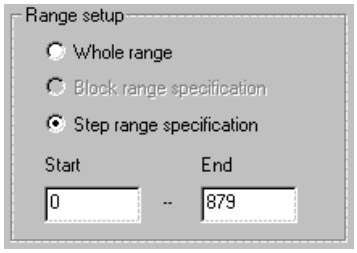


DESCARGANDO UN PROGRAMA AL PLC:

Utilizando la opción del menú principal **Online-Write to PLC…** aparecerá el formulario dedicado a la descarga de archivos al autómata. Desde esta ventana se nos da la posibilidad de escoger que archivos del proyecto queremos descargar al PLC. Si sólo queremos descargar los parámetros porque es lo único que hemos modificado, se selecciona solo la opción PLC Parameter y solo se descargarán los parámetros.



Cuando se selecciona el programa para ser descargado, éste se descarga desde el step cero de programa hasta la última posición disponible correspondiente a la serie del PLC. Por ejemplo, en un FX1N se descargarán 8000 steps de programa aunque solamente se utilicen 880 steps. Para configurar el número de steps a transmitir hacia el PLC (para conseguir un tiempo de transmisión más óptimo), podemos seleccionar la lengüeta Program en el formulario Write to PLC… y escoger Step range specification en lugar de Whole range (seleccionado por defecto). Después escribimos el número total de steps que tiene nuestro programa (este número aparece en el título de la ventana principal del GX Developer). Si el título de la ventana principal nos da la información de que en el programa se han utilizado 880 steps de programa, hay que teclear en la sección Range setup 879 steps (uno menos) ya que el último step siempre es la intrucción END, y el GX Developer la insertará automáticamente cuando se descarga un programa a la CPU.



RECUPERANDO UN PROGRAMA DESDE EL PLC:

Para recuperar un programa desde el autómata hacia el GX Developer, hay que seleccionar la opción del menú principal **Online-Read from PLC…** desde la cual podemos seleccionar los archivos que queremos cargar al ordenador. El formulario que aparece es idéntico al mostrado en el apartado anterior (Descarga de programa).

CREANDO UN CÓDIGO (KEYWORD):

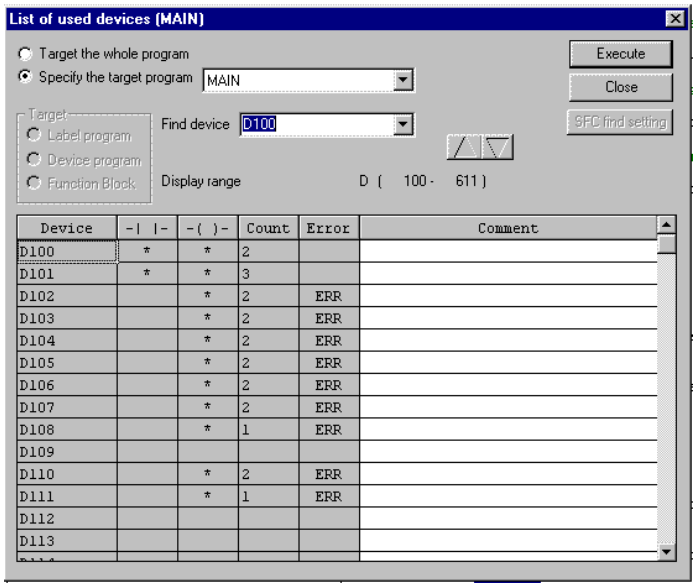
Si necesitamos crear una contraseña de protección para que no pueda ser leído el programa almacenado en el autómata, debe activarse la opción del menú principal **Online-Keyword setup-Register…** para registrar el código de acceso. Este registro se llevará a cabo con el PLC conectado.

Este código debe ser de 8 caracteres (no puede tener ni más ni menos caracteres). Es obligatorio que tenga 8 letras. Los símbolos utilizados pueden ser los números del **0 al 9** y las letras de la **A a la F**.

En la opción **Online-Keyword setup-Delete…** podremos borrar la contraseña, pero hay que conocer el código para que ésta pueda ser borrada. Por lo tanto, es importantísimo NO OLVIDAR EL CÓDIGO!!.

LISTA DE VARIABLES UTILIZADAS:

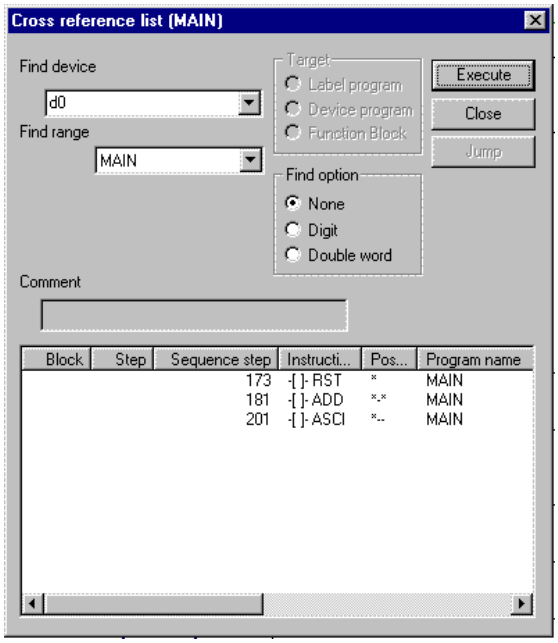
Seleccionando en el menú principal **Find/Replace-List of used devices…**se abre una ventana desde la cual podemos visualizar todas las variables utilizadas desde el proyecto. Hay que seleccionar desde que dispositivo hay que empezar la lista y pulsar en **Execute**. Aparecerá una lista donde especifica si se han utilizado como entrada o salida cada uno de los dispositivos.



Con esto podemos saber que variables están libres para ser utilizadas.

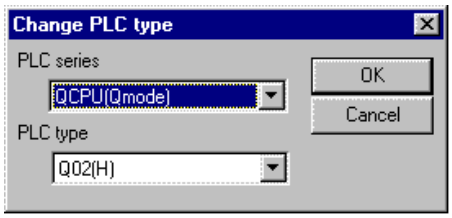
LISTA DE REFERENCIAS CRUZADAS:

Desde esta ventana de trabajo (seleccionada a través del menú principal **Find/Replace-Cross reference list…**) podemos tener una lista similar a la del apartado anterior. Esta vez damos una variable de referencia y se nos devuelve un listado de todos los steps de programa desde donde aparece, así como el nombre de las instrucciones en las que ha sido utilizada.



CAMBIANDO LA CPU DE UN PROYECTO:

A través del menú principal **Project-Change PLC type…** se accede al formulario dedicado al cambio de CPU. Se nos pide, al igual que cuando se crea un nuevo proyecto, la serie de autómatas y la CPU correspondiente a la serie con la que queremos trabajar en el proyecto actual. La ventana es como se muestra a continuación. Hay que escoger la CPU y pulsar **OK**.



PROGRAMACIÓN EN LENGUAJE ESTRUCTURADO (ST)

Que es la programación en lenguaje estructurado (ST = Structured Text)?
El lenguaje ST viene definido en el estándar internacional IEC61131-3 que estipula el sistema de descripción lógica de los controladotes abiertos. El lenguaje ST soporta operadores, sintaxis de control y funciones que permiten las siguientes prestaciones:

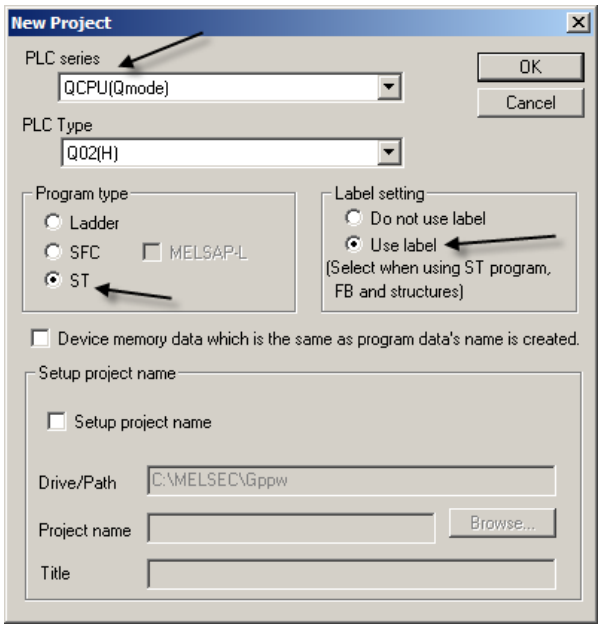
- sintaxis de control y sentencias condicionales de selección de rama. Sentencias de control repetitivo.
- Expresiones con operadores: *, / , + , - , >, <, =, etc...
- Llamadas a bloques de función del usuario (FB)
- Llamadas a instrucciones Melsec y IEC
- Descripción de comentarios

Que prestaciones aporta ST?

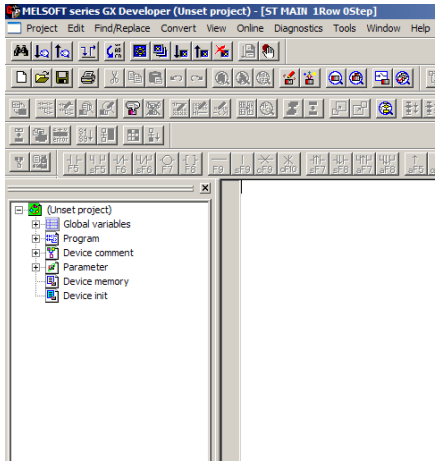
- Mejora de la eficiencia de programación al definir el procesamiento como partes. El uso de código como bloques de función (FB) reutilizables a lo largo del programa reduce errores y mejora la calidad del programa.
- Conexión con otros lenguajes de programación. Al disponer de varios lenguajes de programación, es posible usar lo mejor de cada uno de ellos allá donde se requiere. Por ejemplo, usar ladder para el control de secuencia y usar ST para el procesamiento de señales.
- Gran cantidad de funciones disponibles: las funciones Melsec propias de la serie MELSEC-Q y las funciones IEC61131-3 están disponibles para facilitar el trabajo del desarrollador.

CREACIÓN DE UN PROYECTO EN ST

La programación en lenguaje estructurado ST solamente es válida para la familia de PLCs MELSEC-Q.
Para crear un nuevo proyecto [Project]->[New Project..]

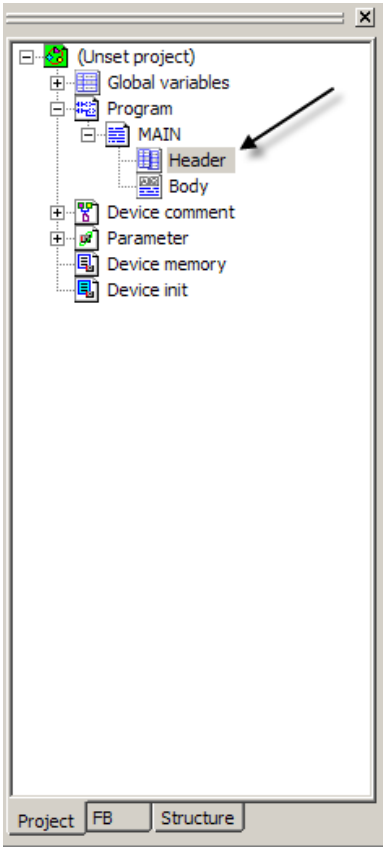


Asegúrese de seleccionar una CPU de la familia MELSEC-Q.
Seleccione **Use Label** si va a trabajar en ST.
Seleccione **ST** como lenguaje de programación.

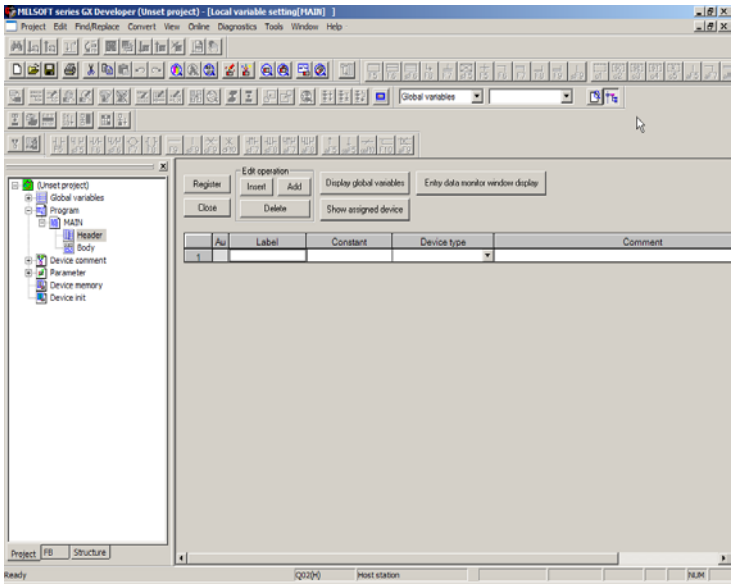


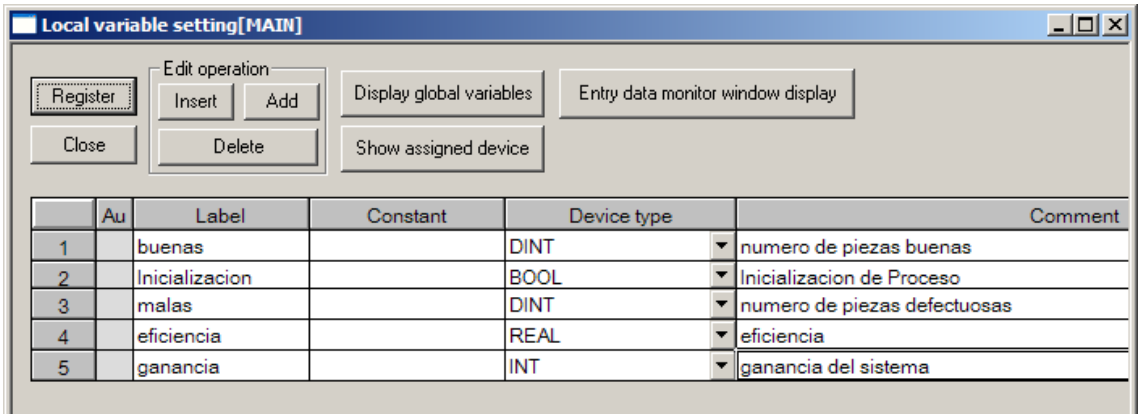
DEFINICIÓN DE LABELS

El trabajar en ST nos obliga a declarar de antemano las variables con las que vamos a trabajar. El concepto es muy similar a trabajar en el lenguaje de programación Pascal. Se dispone de dos tipos de labels: las variables globales y las variables locales. Las variables globales se pueden usar en todo el proyecto mientras que las locales se usan en el programa donde se han definido.



Clicar en Header para llegar a la ubicación de declaración de variables locales a Main.



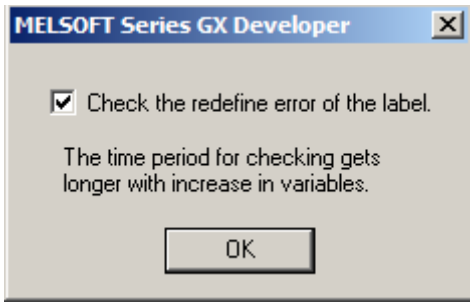


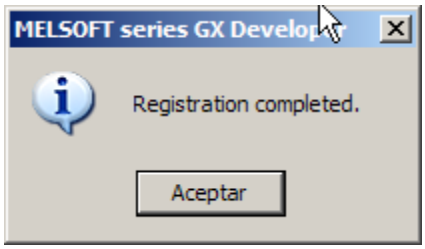
	Au	Label	Constant	Device type	Comment
1		buenas		DINT	numero de piezas buenas
2		Inicializacion		BOOL	Inicializacion de Proceso
3		malas		DINT	numero de piezas defectuosas
4		eficiencia		REAL	eficiencia
5		ganancia		INT	ganancia del sistema

Hay que definir el nombre de la variable en **Label**, del tipo de dispositivo en **Device Type** (Booleana, entera, doble entera, real, cadena, matriz, etc) y en **Comment** se puede escribir un comentario de hasta 64 caracteres.

Los botones **Insert**, insertan una fila para definir una variable más, **Add** añade una fila al final, y **delete** borra la fila seleccionada.

Finalizamos la introducción de variables pulsando el boton **Register** que verifica las variables introducidas hasta el momento.

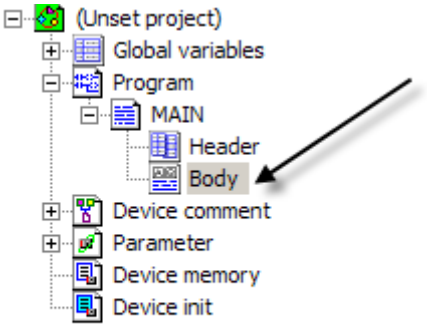




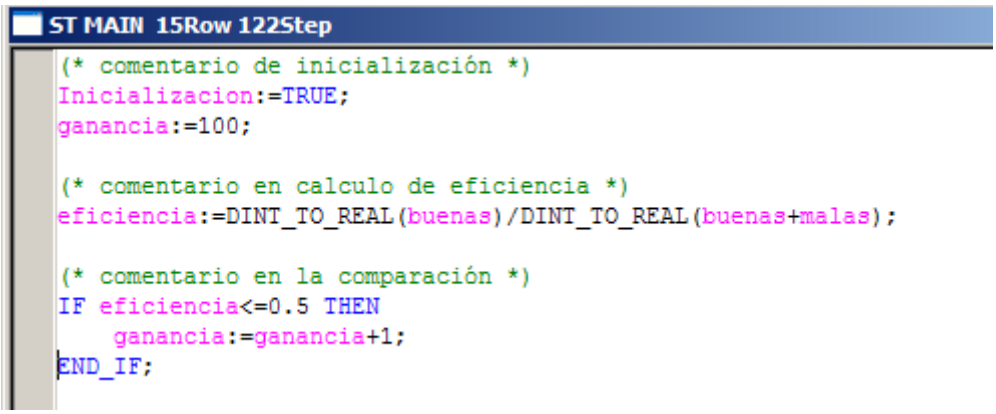
Tras Register pulsar OK y Aceptar.

EDICIÓN DE PROGRAMA ST

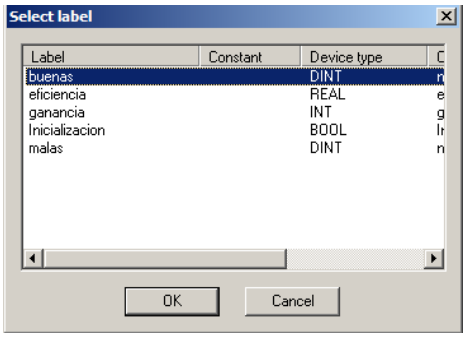
Seleccionar con el ratón el Body (cuerpo) del procedimiento.



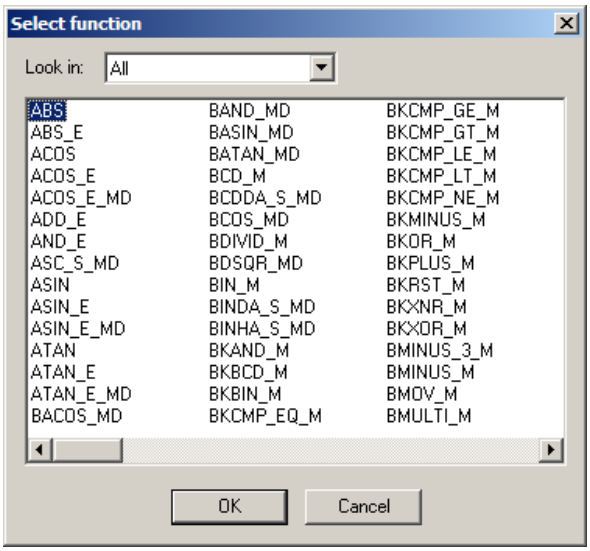
La introducción de las diferentes sentencias del programa se realiza mediante una pantalla de edición de texto.
Durante la introducción de una variable o una sintaxis de control, el color del texto varia. Si no cambia, puede ser debido a un error de sintaxis o bien a una variable no definida.



Nótese que en modo edición, pulsando la tecla ***F11*** o bien [Edit]->[Select Label...] se puede seleccionar la variable de una lista de las que hemos definido previamente.



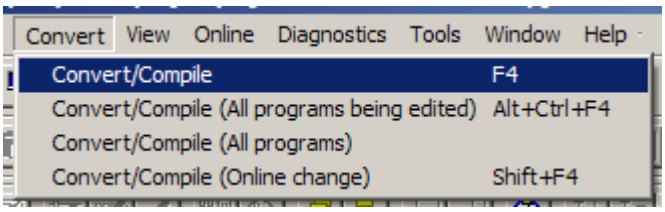
De igual manera pulsando la ***Shift+F11***, o bien seleccionando [Edit]->[Select function...] aparece un menú con un listado de todas las funciones disponibles para usar en nuestro proyecto.



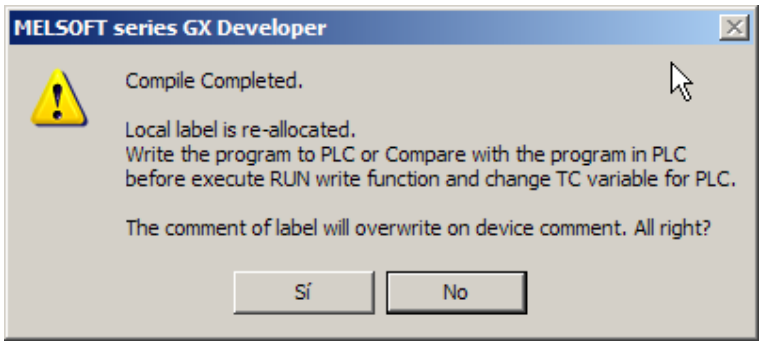
COMPILANDO (CONVERT) EL PROGRAMA ST

Conversión (Compilación en términos informáticos) del programa creado con lenguaje de texto estructurado ST para convertirlo en un programa de secuencia que pueda ser ejecutado por la CPU del PLC.

Ejecución de Convert: ***F4*** o [Convert]->[Convert/Compile]

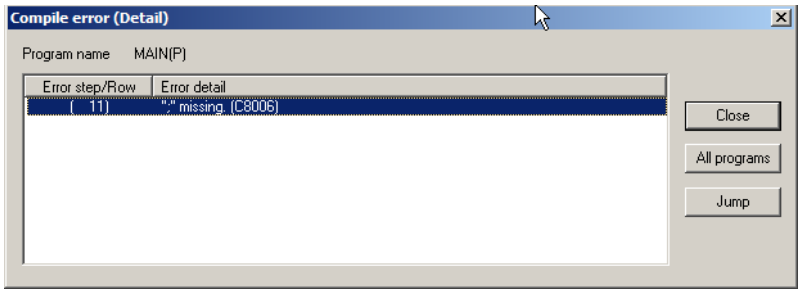


si la compilación ha tenido éxito aparecerá la ventana. Pulse la opción [No].

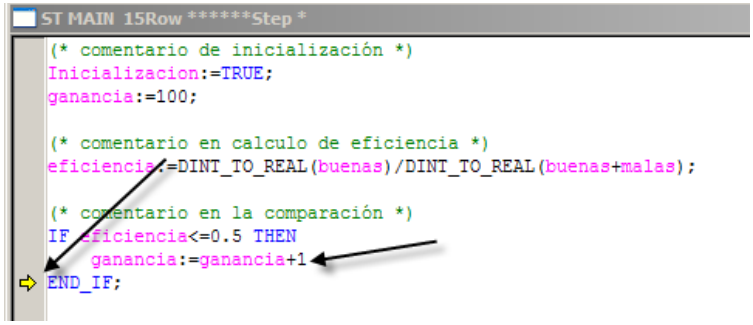


el número de pasos de programa ocupados por nuestro código aparecerá en el marco de la ventana del editor de textos.

Si aparecen errores durante la compilación el programa indica en que paso/línea esta el error y una pequeña explicación del error. En el ejemplo, falta el punto y coma final.



el programa nos indica con una flecha amarilla la ubicación del error dentro de la ventana del editor de textos.



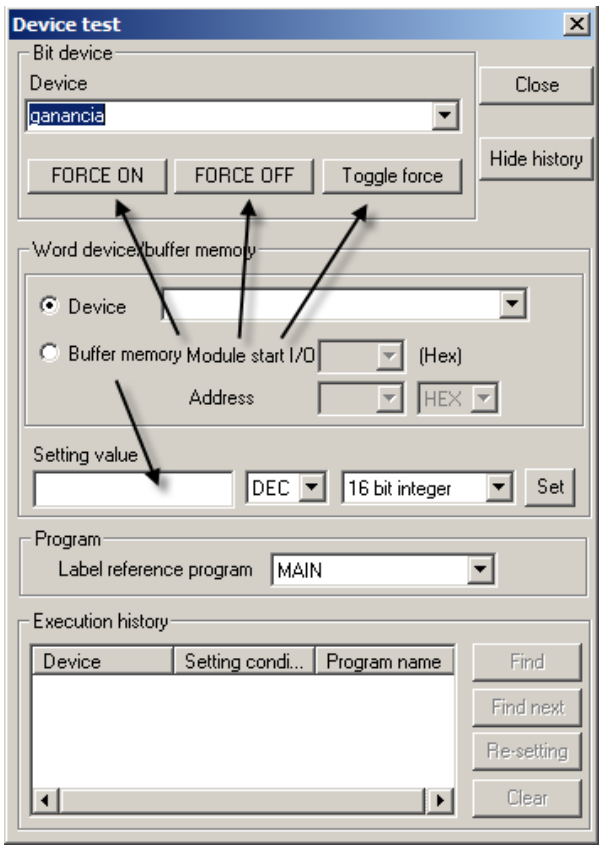
MONITORIZACIÓN DEL PROGRAMA ST

GX Developer permite monitorizar online el valor de las variables que se procesan en el programa ST. Para ello estando en la ventana del editor del programa pulsamos ***F3*** o vamos a [Online]->[Monitor]->[Stara Monitor]

La ventana del editor de textos ST se divide en dos, una de ellas mostrando el código ST y la otra mostrando solamente las variables y su valor en cada momento.

MODIFICACIÓN DE VARIABLES (DEVICE TEST)

GX Developer permite así mismo, modificar online el valor de las variables para poder simular el correcto funcionamiento de la aplicación. Pulsando ***Alt+1*** o [Online]->[Debut]->[Device Test...]



Desde esta ventana es posible forzar a ON, forzar a OFF o alternar el valor de un dispositivo (variable).

También es posible dar valor a una variable.

Adicionalmente nos visualiza un histórico de la ejecución de forzados.

CAMBIOS ONLINE

GX Developer admite también cambios del programa ST online, es decir mientras la CPU esta en estado de RUN.

Por ejemplo si nuestro programa es:

Area := base + altura;

Podemos acceder a la ventana de editor ST i cambiarlo por:

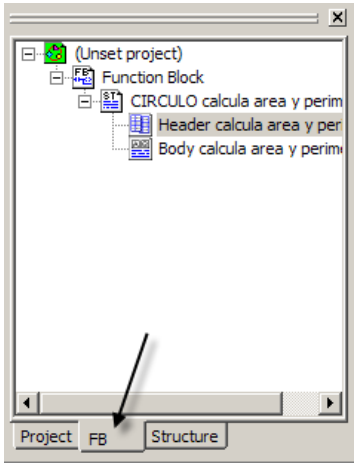
***** Area := (base * altura) / 2; aparece un asterisco para indicarnos que se trata de una línea modificada respecto a la original.

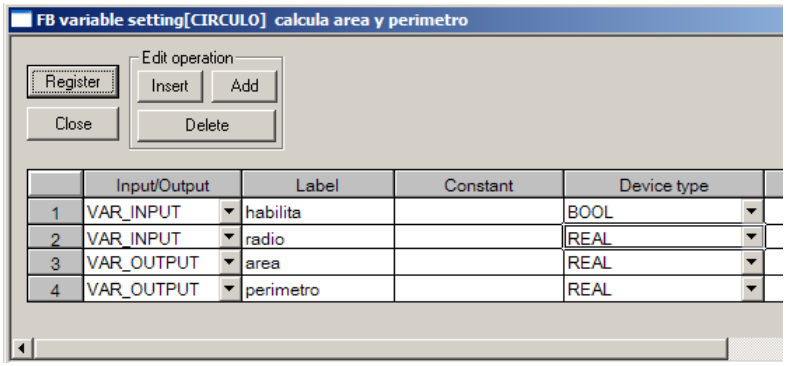
Seleccionando ***Shift+F4*** o [Convert]->[Convert/Compile (online change)] se recompila y se transfiere el programa modificado al PLC.

CREACIÓN DE BLOQUES DE FUNCIÓN EN ST

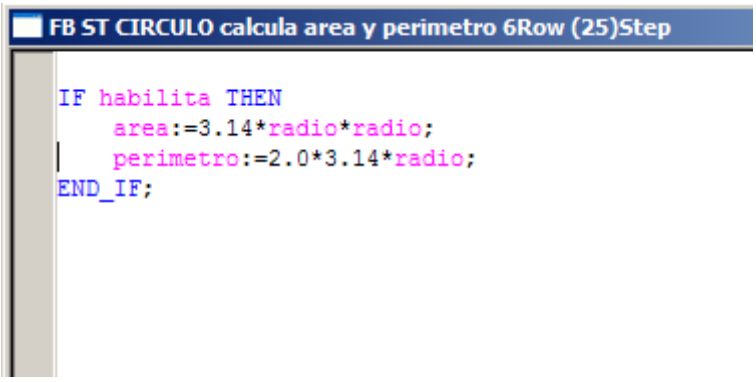
Para mostrar la creación de bloques de función (FB) empleando texto estructurado (ST) emplearemos el ejemplo siguiente: Se quiere realizar una función que nos calcule el perímetro y el área de un circulo de radio “r”.

Para ello creamos un programa como se describe en la página 32. En este caso definimos el bloque de función CIRCULO en formato texto estructurado ST.

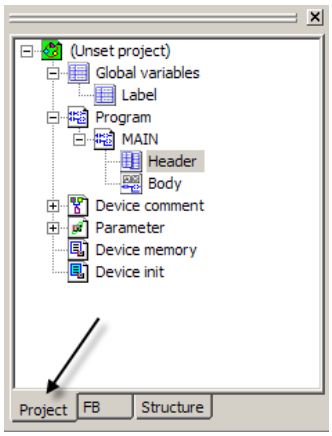


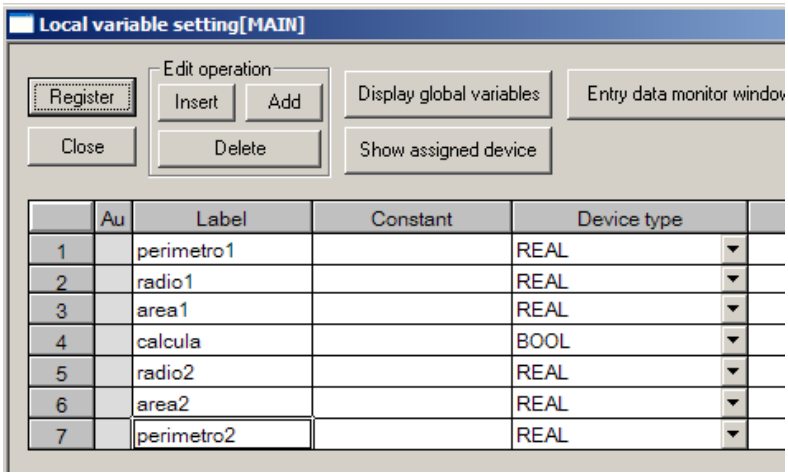


En Header (la cabecera) definiremos las variables (labels) de entrada / salida con sus nombres y sus tipos. Seguidamente en body (cuerpo de programa) hacemos uso de las variables para definir el programa que hará el cálculo del área y el perímetro. El proceso es igual al descrito en la página 32 pero esta ves empleando ST.

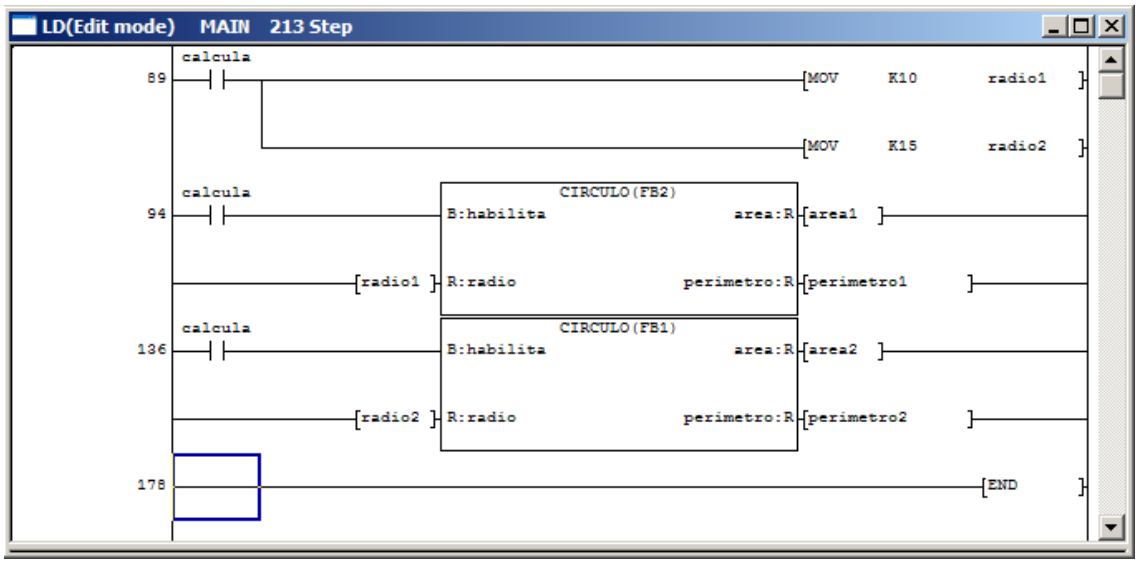


Tras la definición del bloque de función circulo, solamente tenemos que definir las variables (labels) del programa MAIN con las cuales trabajaremos para hacer los cálculos. En nuestro ejemplo calcularemos los las áreas y perímetros de 2 círculos:





Finalmente, el programa en ladder con las dos llamadas al FB CIRCULO queda de la siguiente manera:



MITSUBISHI ELECTRIC

Su Socio de Futuro Presente en Todo el Mundo

mitsubishielectric.es

En nuestra página WEB, usted encontrará la última información a cerca de cada uno de nuestros productos y las más recientes novedades. Continuamente actualizamos nuestra página para que usted esté completamente informado.

industrial@sp.mee.com

Puede contactarnos vía email, donde responderemos inmediatamente a sus consultas.

FAX-RESPUESTA 93 589 15 79

Nuestro sistema FAX RESPUESTA le permitirá realizar cualquier consulta y nosotros le enviaremos la información que usted solicite.

LINEA DE ATENCIÓN AL CLIENTE
902 131121

En caso de tener cualquier tipo de problema, puede contactar con nuestra LINEA DE ATENCIÓN AL CLIENTE donde puede contar con ayuda profesional.

Especificaciones sujetas a cambio sin previo aviso
Art.190066 Impreso en España 06/04

MITSUBISHI ELECTRIC EUROPE, B.V. Sucursal en España
División: FACTORY AUTOMATION