



Módulo 3.7

Lenguaje IEC : FBD

● Generalidades

- Diagrama Bloque Función consiste de un Editor Gráfico Forma Libre que está orientado hacia flujo de datos.
- Secciones FBD tienen una rejilla como fondo. La programación lenguaje FBD no está orientada hacia celdas, sin embargo los objetos aún se alinean con las coordenadas de las rejillas.
- Conforme al estándar IEC 61131-3
- Adecuado particularmente para aplicaciones control continuas o discreto
- Lenguaje consiste de Bloques Función reusables elementales y derivados, Elementos Programa, Macros y Estructuras Datos
- Una Sección FBD contiene una rejilla de 30 renglones y 23 columnas

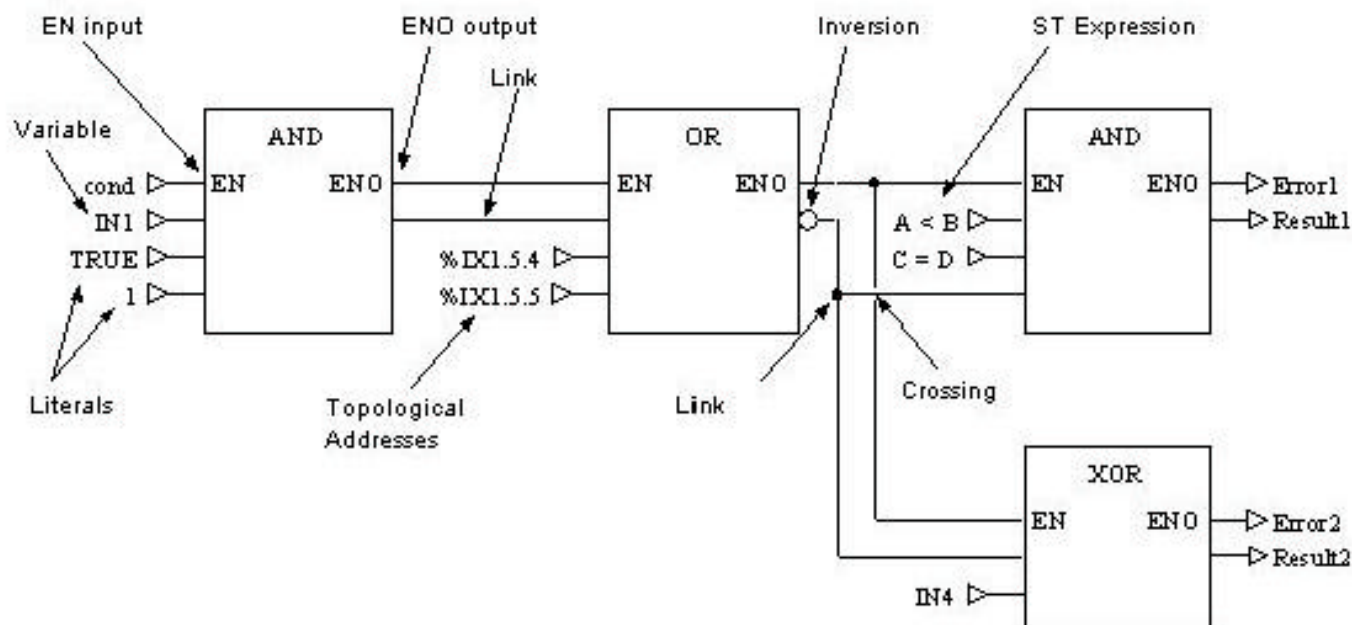
● Generalidades (cont.)

- Los objetos del lenguaje programación FBD (Diagrama Bloque Función) ayuda a dividir una Sección en un número de:
 - ▶ EFs y EFBs (Funciones Elementales y Bloques Funciones Elementales)
 - ▶ DFBs (Bloques Función Derivados)
 - ▶ Procedimientos y
 - ▶ Elementos Control
- Estos objetos, combinados bajo el nombre FFBs, pueden enlazarse con otros por:
 - ▶ Enlaces o
 - ▶ Parámetros Actuales
- Pueden proporcionarse comentarios para la lógica de la Sección con objetos textos

● Ejemplo FBD

Representation of an FBD section

Representation:



● Creando una sección

1 - Botón-derecho y seleccionar

2 - Seleccionar FBD

3 - Validar

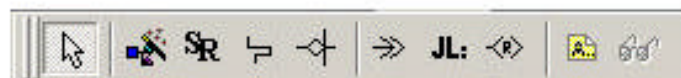
The image shows a screenshot of the Unity Pro software interface. On the left, a tree view shows the project structure with folders like 'Networks', 'Routing Table', 'Program', 'Tasks', 'MAST', 'Events', 'Animation Tables', 'Operator Screens', and 'Documentation'. A right-click context menu is open over the 'Section' folder, with 'New Section ...' highlighted. A 'New' dialog box is open in the foreground, showing the 'General' tab. The 'Name' field contains 'TrainFBD'. The 'Language' dropdown menu is set to 'FBD'. The 'OK' button is highlighted.

● Objeto Programación



- Hay 3 métodos para acceder los Objetos en la programación FBD:
 - ▶ Elegir “Edición”, después “Nuevo”

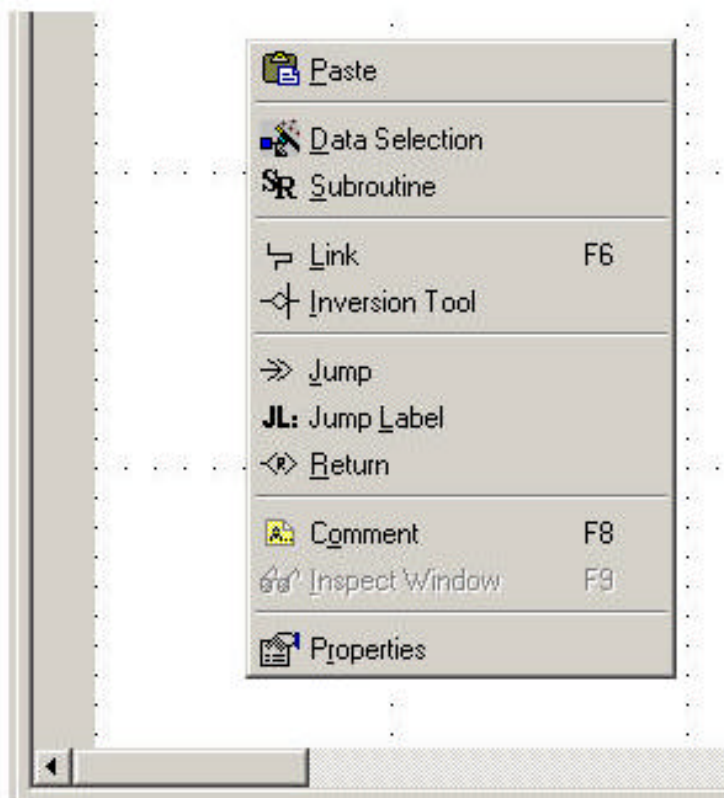
● Objeto Programación - Barra Herramientas



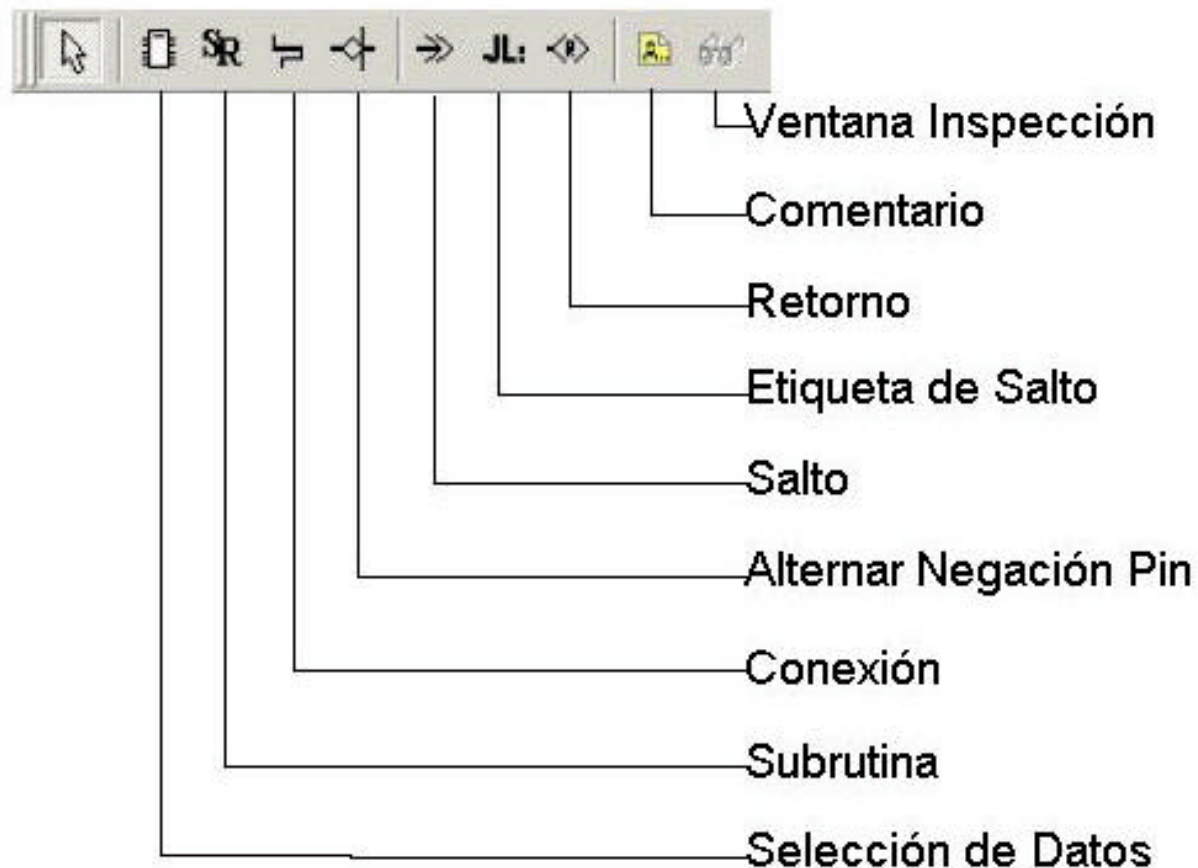
- Hay 3 métodos para acceder los Objetos en la programación FBD:
 - ▶ Elegir “Edición”, después “Nuevo”
 - ▶ Uso de la Barra Objetos FBD

● Objeto Programación - Botón derecho

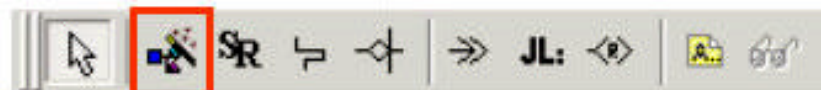
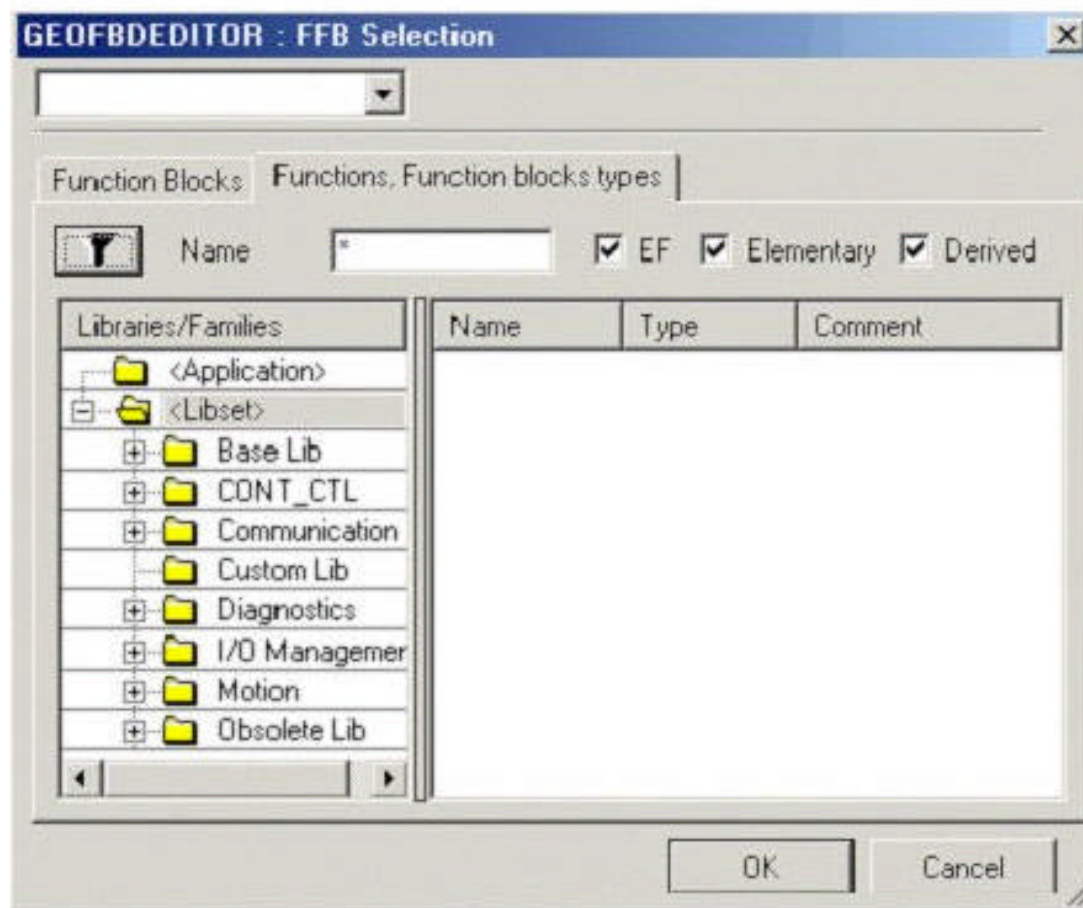
- Hay 3 métodos para acceder los Objetos en la programación FBD:
 - ▶ Elegir "Edición", después "Nuevo"
 - ▶ Uso de la Barra Objetos FBD
 - ▶ Colocación cursor en la Ventana Editor Lógica, después botón derecho para seleccionar Grupos Objeto



● Objeto Programación



● Librería



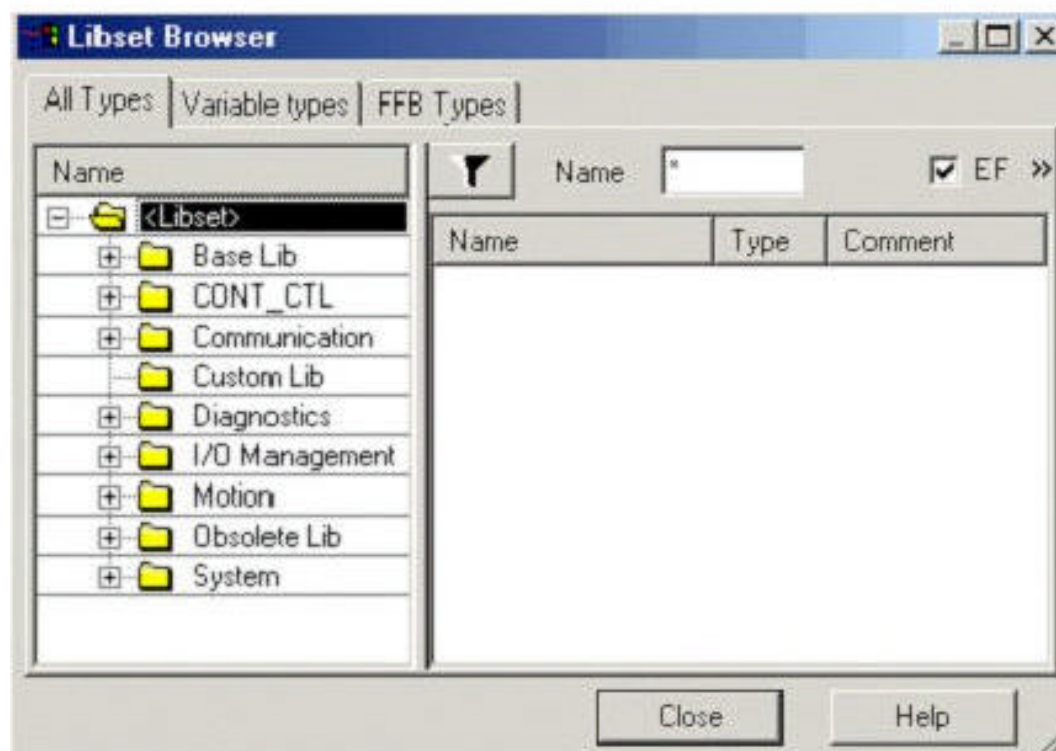
- Funciones (F)
- Bloque Función (FB)
- Bloques Función Elementales (EFB)
- Bloques Función Derivados (DFB)
- Abreviación General = FFB (Función/Bloque Función)
- Macros (FFBs, DFBs)
- FFBs están en Librerías y Grupos
- Interconexión de FFBs con:
 - ▶ Enlaces Gráficos o
 - ▶ Nombre Variable: "MOT_LE"

● Librería

FFBs pueden seleccionarse eligiendo el Icono en la Barra de Herramientas Selección de Datos. Esta acción abrirá la caja de diálogo mostrada anteriormente.

El Conjunto de Librerías Libset está en categorías la cual puede expanderse para localizar el FFB adecuado.

● Explorador Libset



- Eligiendo un Icono Librería muestra al Explorador Librería siendo un modo conveniente para ver el contenido de las diversas librerías
- Expandiendo las Librerías principales genera sub-librerías las cuales contienen una gran variedad de FFBs
- Puede crearse una Librería Particular conteniendo justo aquellos FFBs necesarios para la aplicación propia

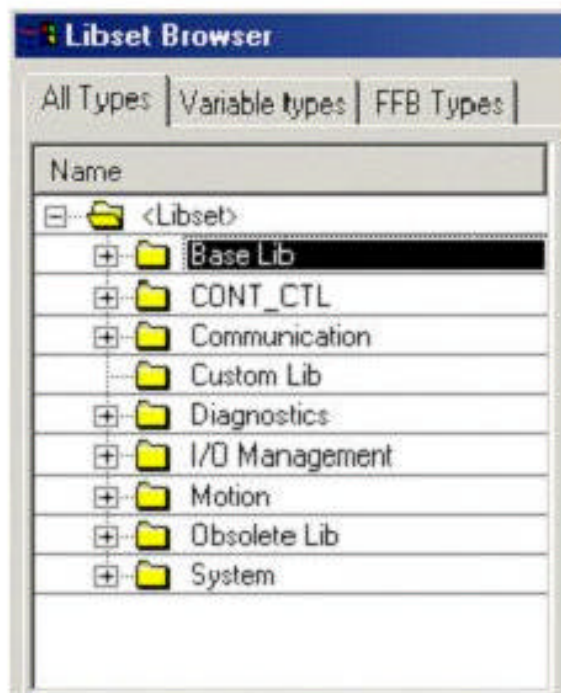
● Librería

FFBs pueden seleccionarse eligiendo el Icono en la Barra de Herramientas Selección de Datos. Esta acción abrirá la caja de diálogo mostrada anteriormente.

El Conjunto de Librerías Libset está en categorías la cual puede expanderse para localizar el FFB adecuado.

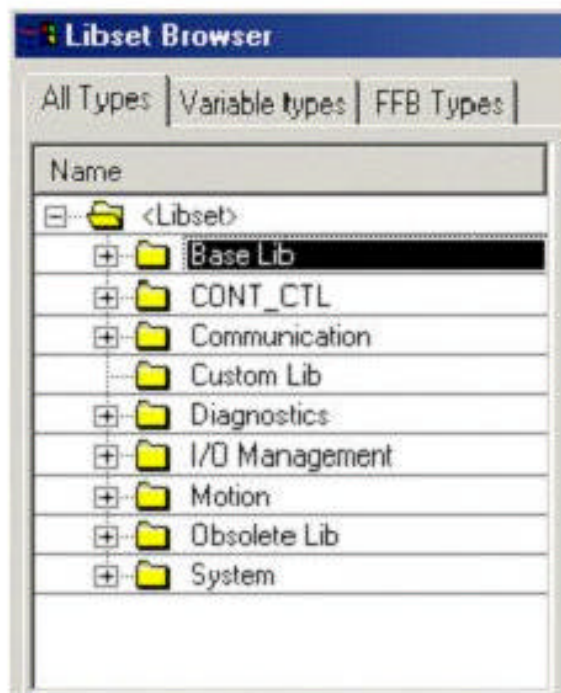


● Librerías Proporcionadas



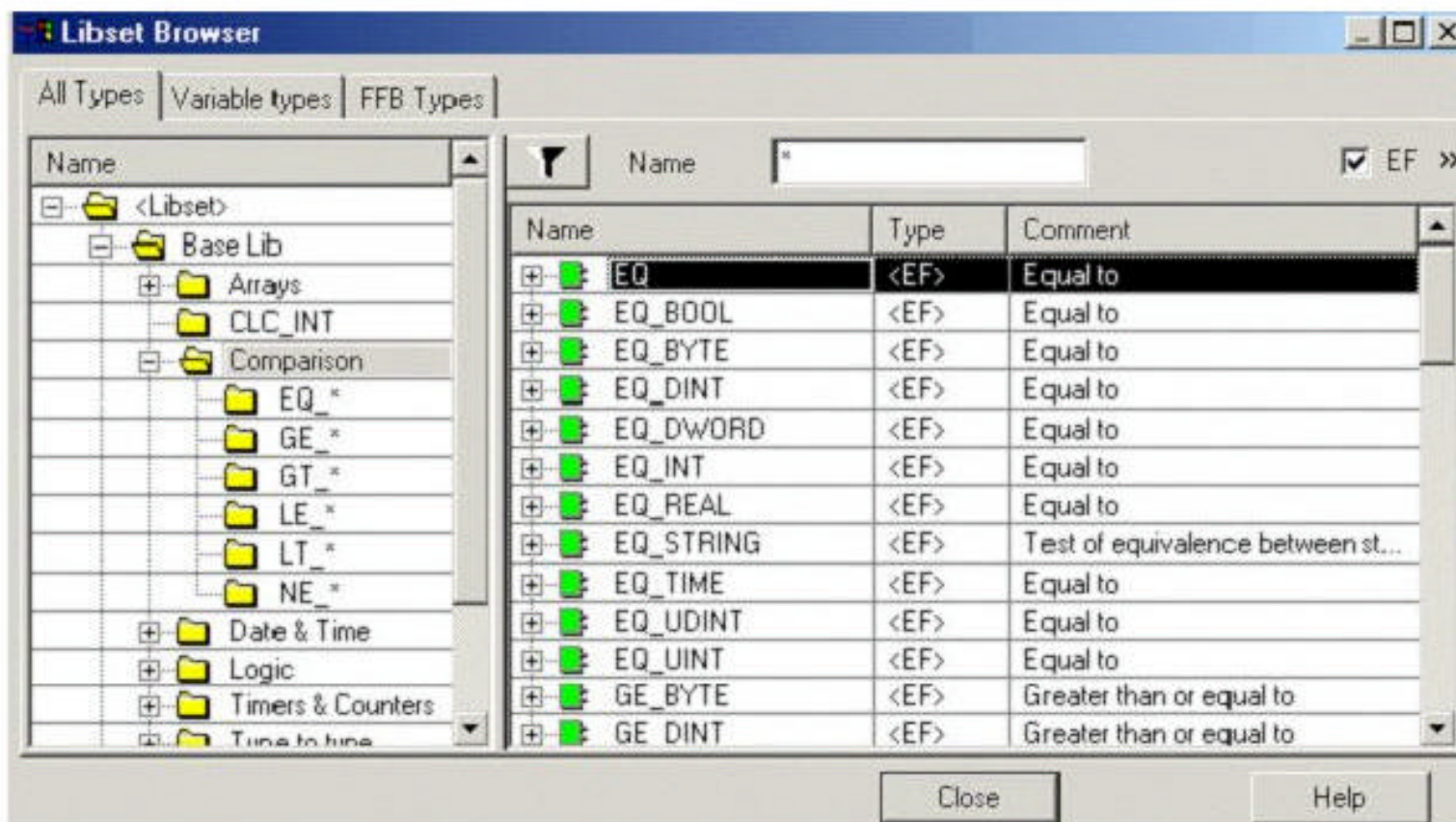
- **BASE LIB**: con los grupos Arreglos, CLC_INT, Comparación, Fecha & Hora, Lógica, Matemáticas, Estadísticas, Caracteres, Temporizadores & Contadores, Tipo a Tipo
- **CONT_CTL**: con los grupos Condicionamiento, Controlador, Matemáticas, Medición, Procesamiento Salida, Gestión Punto Ajuste
- **COMMUNICATION**: con el grupo Extendido conteniendo 28 FFBs Comunicación
- **DIAGNOSTICS**: con el grupo Diagnósticos conteniendo 15 FFBs Diagnósticos

● Librerías Proporcionadas (cont.)



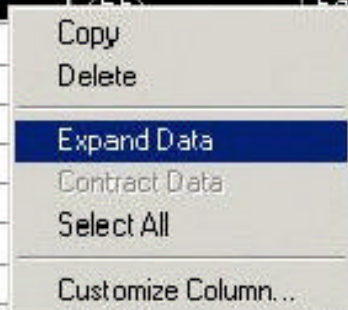
- **I/O MANAGEMENT**: con los grupos Configuración E/S Análogas, Escalación E/S Análogas, Intercambio Explícito, E/S Inmediatas, Interbus_S, Configuración E/S Quantum
- **MOTION**: con los grupos Control Ejes, Control CAM , MMF_Start
- **OBSOLETE LIB**: con los grupos CLC, CLC_PRO, Extensiones/Compatibilidad
- **SYSTEM**: con los grupos Eventos, SFCManagement, SysClock

● Expansión Explorador Libset

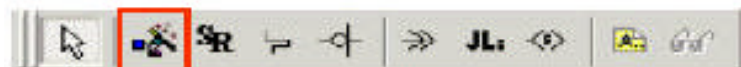


● Expansión Explorador Libset (cont.)

Name	Type	Com
+ EQ	<EF>	Equal to
+ EQ_BOOL	<EF>	Equal to
+ EQ_BYTE	<EF>	Equal to
+ EQ_DINT	<EF>	Equal to
+ EQ_DWORD	<EF>	Equal to
+ EQ_INT	<EF>	Equal to
+ EQ_REAL	<EF>	Equal to
+ EQ_STRING	<EF>	Equal to
+ EQ_TIME	<EF>	Equal to

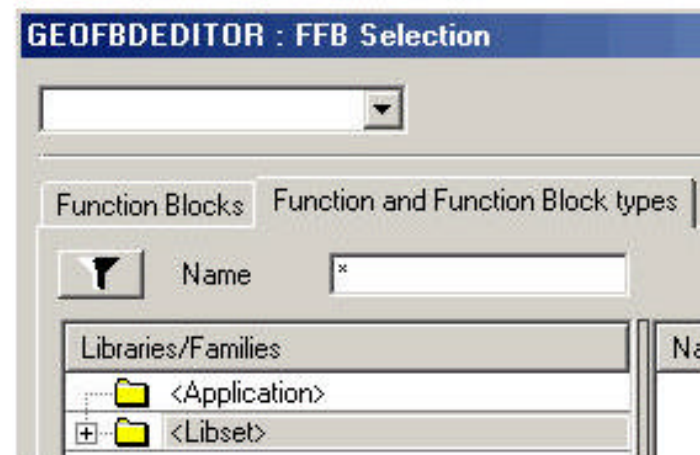


Name	Type	Comment
EQ	<EF>	Equal to
<inputs>		
IN*	ANY_ELEMEN...	Input 1..31
IN1	ANY_ELEMEN...	
IN2	ANY_ELEMEN...	
<outputs>		
OUT	BOOL	1: equal, 0: not equal
<inputs/outputs>		
EQ_BOOL	<EF>	Equal to
EQ_BYTE	<EF>	Equal to



● Inserción de Bloque Función

Para insertar un Bloque Función en una Sección FBD:



1. Crear una nueva Sección FBD
2. Elegir el Icono FFB de la Barra
3. Seleccionar la opción "Function and Function Blocks"
4. Bajo Libset - Base Library - Logic, seleccionar AND_BOOL y confirmar OK.
5. Colocar el Bloque AND_BOOL donde se requiere.
6. Asignar las Variables a los pins (entradas/salidas)

Gráficos continúan en la siguiente página



Módulo 3 : Unity Pro
 Sección 7 : Lenguaje IEC : FBD
 Página 16/23

● Inserción de Bloque Función (cont)

GEOFBEDITOR : FFB Selection

AND_BOOL

Function Blocks | Function and Function Block types

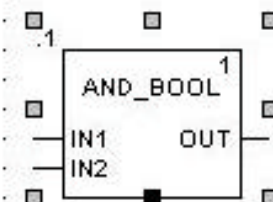
Name *

EF EFB DFB

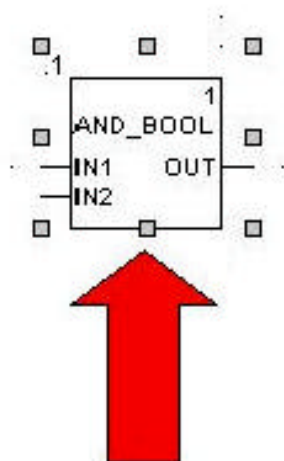
Libraries/Families

- <Application>
- <Libset>
 - Base Lib
 - Arrays
 - CLC_INT
 - Comparison
 - Date & Time
 - Logic
 - AND *
 - OR *

Name	Type	Comment
AND	<EF>	AND function
AND_BOOL	<EF>	AND function
AND_BYTE	<EF>	AND function
AND_DWORD	<EF>	AND function
AND_WORD	<EF>	AND function

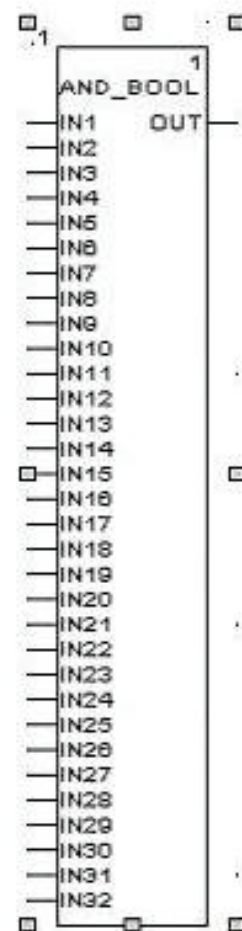


● Extendible

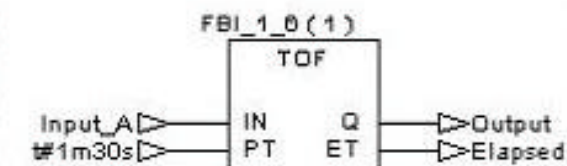
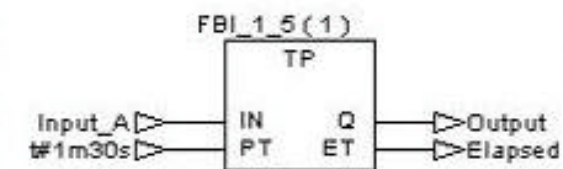
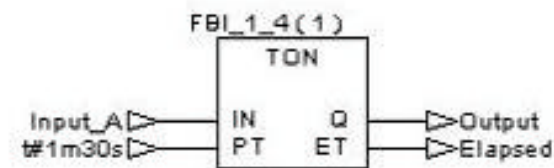
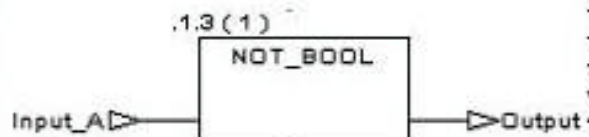
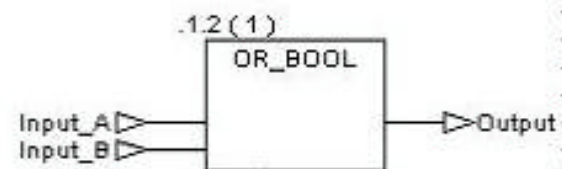
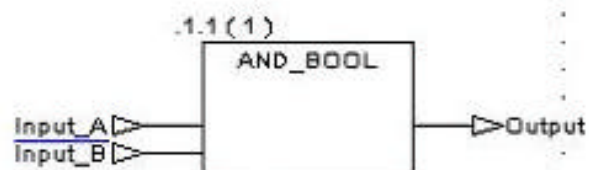


¿cuántos pines necesita?
Solo tomar el fondo del
bloque función y extenderlo

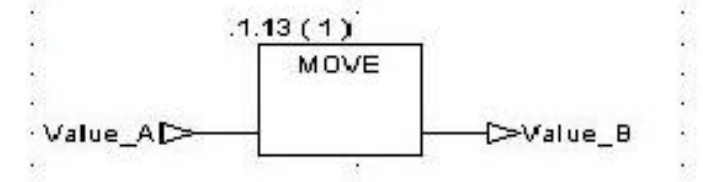
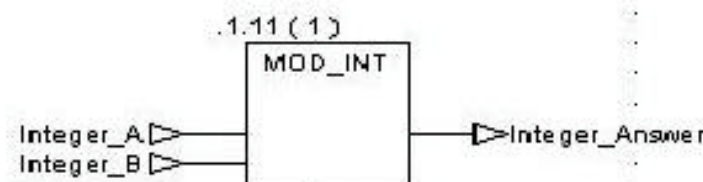
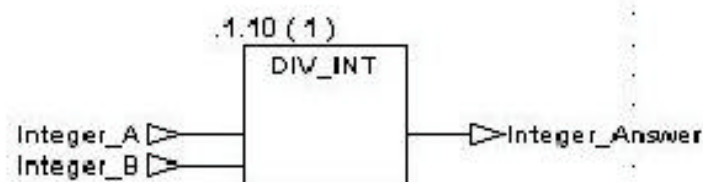
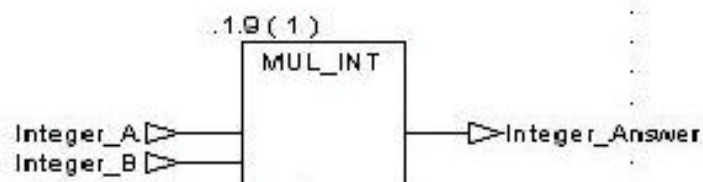
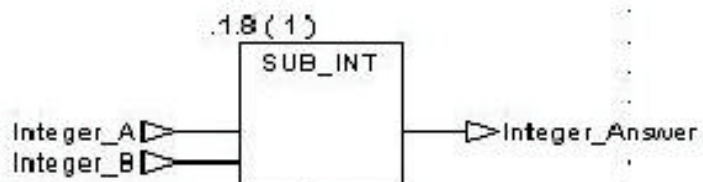
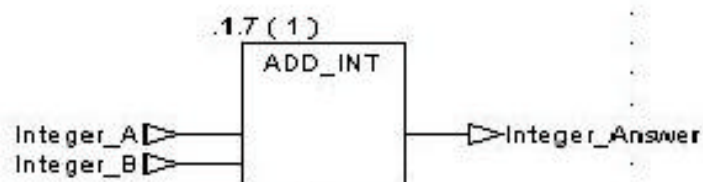
Hasta 32 pines



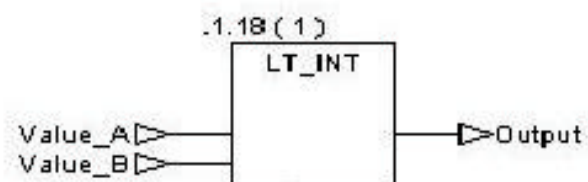
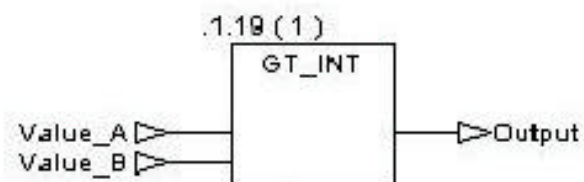
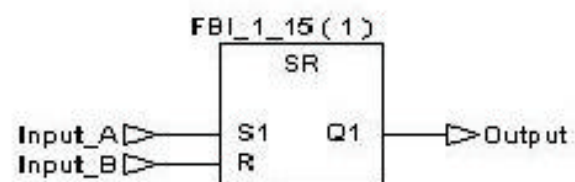
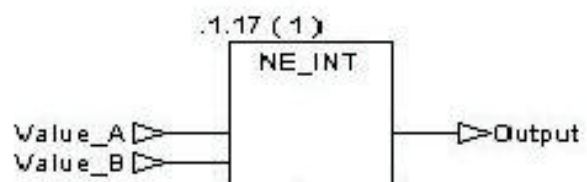
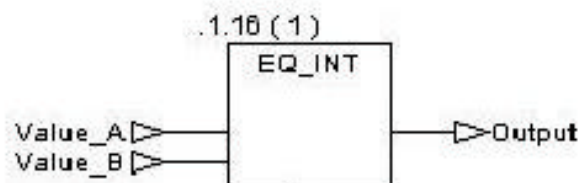
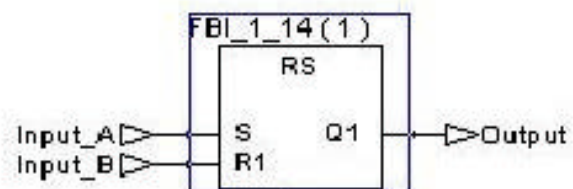
● Funciones Básicas Lógica y Tiempo



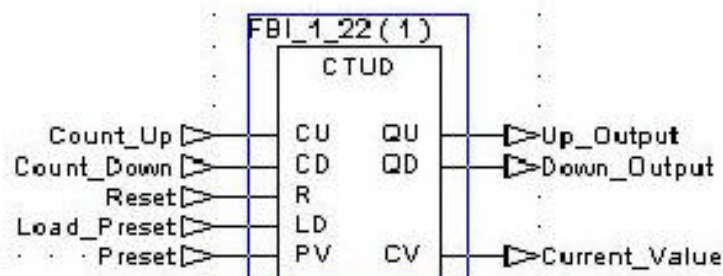
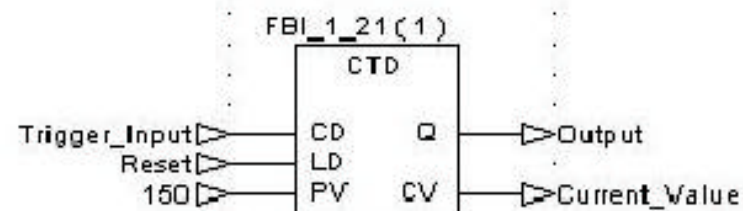
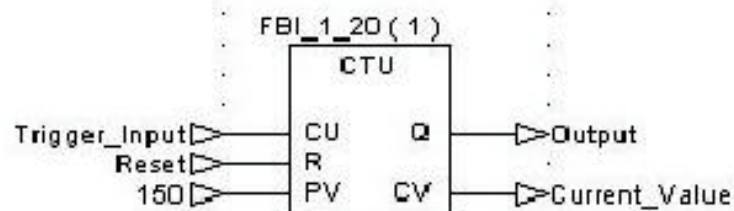
● Funciones Básicas Matem.



● Enclaves Básicos y Comparaciones



● Funciones Contadores Básicos



● Funciones Contadores Básicos

Nota: Contadores IEC no detienen el conteo cuando éstos alcanzan su valor pre-seleccionado, sino que continuarán acumulando cuentas. Este comportamiento es diferente de los contadores 984LL, los cuales se detienen cuando han alcanzado su cuenta pre-seleccionada.

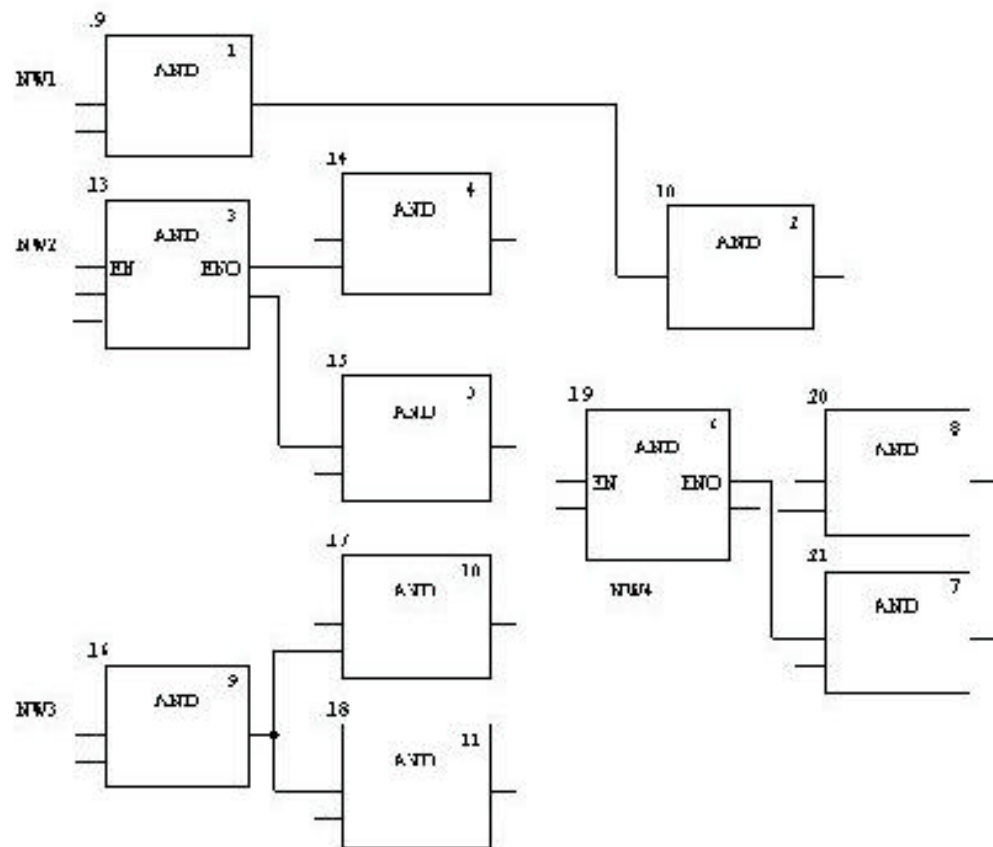


Módulo 3 : Unity Pro

Sección 7 : Lenguaje IEC : FBD

Página 22/23

● Secuencia Ejecución



A la izquierda hay un ejemplo de la Secuencia Ejecución de Objetos en una Sección FBD.

La secuencia ejecución se determina por la posición de los FFBs dentro de la Sección (ejecutada de izquierda a derecha y de arriba a abajo). Si los FFBs se enlazan gráficamente, la secuencia ejecución se determina por el flujo de señal.

La secuencia ejecución se indica por el número ejecución (número en la esquina superior derecha del marco FFB). **Ver comentarios adicionales**

● Notas Secuencia Ejecución

Secuencia Ejecución en Mallas

Para secuencias ejecución malla, se aplican las siguientes reglas:

- Ejecución de una sección se completa malla por malla basado en los enlaces FFB a partir de arriba y abajo.
- Los enlaces no pueden usarse para crear lazos debido a que la secuencia de ejecución en este caso no puede determinarse claramente. Lazos deben crearse usando parámetros actuales.
- La secuencia de ejecución para mallas que no están enlazadas se determina por la secuencia gráfica (desde superior-derecha a inferior-izquierda). Esta secuencia ejecución puede ser influenciada.
- Procesamiento en una malla se termina por completo antes que comience el procesamiento en otra malla cuyas salidas hayan sido usadas en la malla anterior.
- Ningún elemento de una malla se considera para su procesamiento mientras que el estado de todas las entradas de este elemento no hayan sido calculadas.
- Procesamiento de una malla solo se termina si todas las salidas en dicha malla han sido procesadas.



● Cambio Secuencia Ejecución

El orden de ejecución en mallas y el orden de ejecución de objetos dentro de una malla se definen por un número de reglas. En algunos casos, el orden de ejecución sugerido por el sistema deberá ser cambiado.

El procedimiento para definir/cambiar la secuencia ejecución de mallas es como sigue:

- Usando Enlaces en lugar de Parámetros Actuales
- Posiciones Malla
- Definición Explícita Secuencia Ejecución

El procedimiento para definir/cambiar la secuencia ejecución dentro de una malla es como sigue:

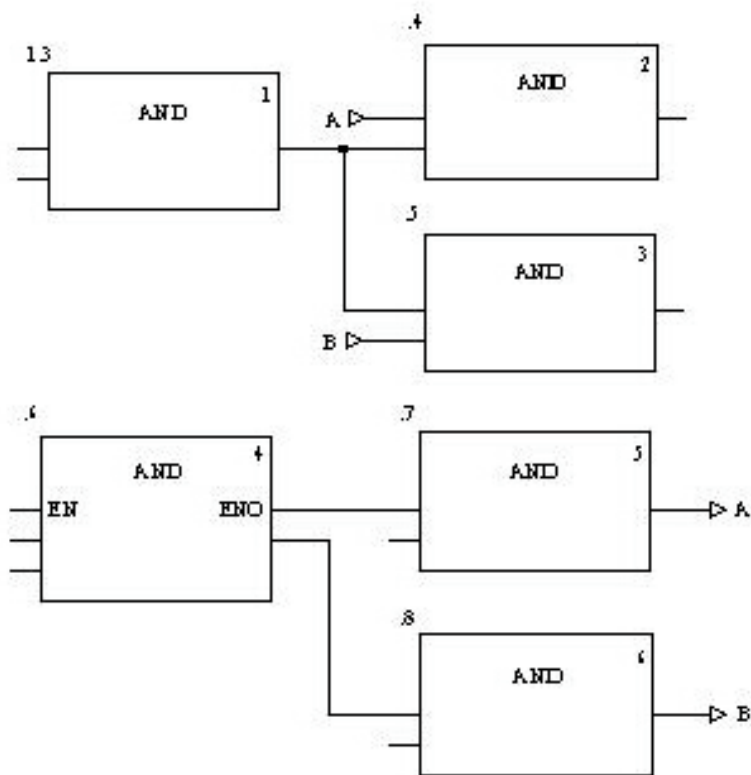
- Posiciones FFB

Ver ejemplos de estas reglas bajo Comentarios

● Ejemplos de Reglas

1. Situación Original

La representación siguiente muestra dos mallas donde las secuencias ejecución se definen por sus posiciones dentro de la sección, sin tomar en consideración que el bloque 0.4/0.5 y 0.7/0.8 requieren otra secuencia ejecución.





Módulo 3 : Unity Pro

Sección 8 : Pruebas a la aplicación

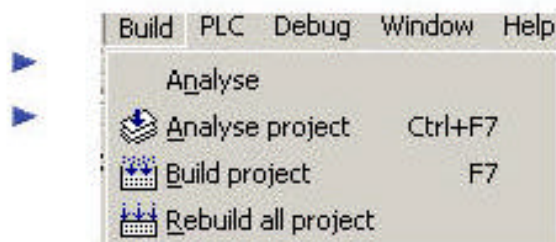
Página 0/8

Módulo 3.8

Haciendo pruebas a la Aplicación

● Generación

El submenú de generación "Build" permite tener acceso a las herramientas del software utilizadas para realizar la generación y análisis del proyecto.



Análisis: Asegura que la sección actual contiene un programa válido.

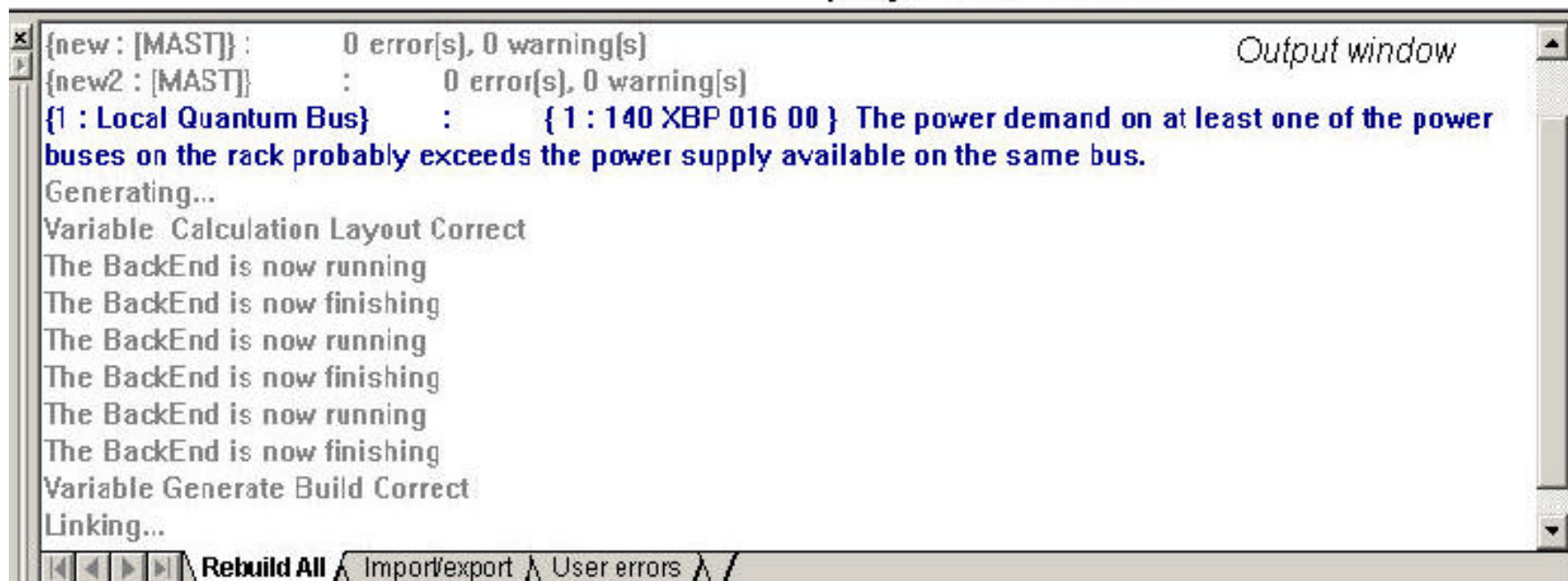
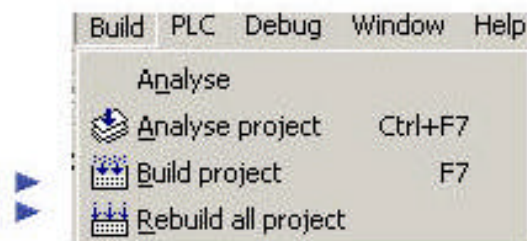
- ▶ detecta errores de Programación como, símbolos indefinidos, elementos no conectados etc.. Estos errores se desplegarán en la ventana asociada la análisis.

Análisis del proyecto: realiza un barrido de todo el programa



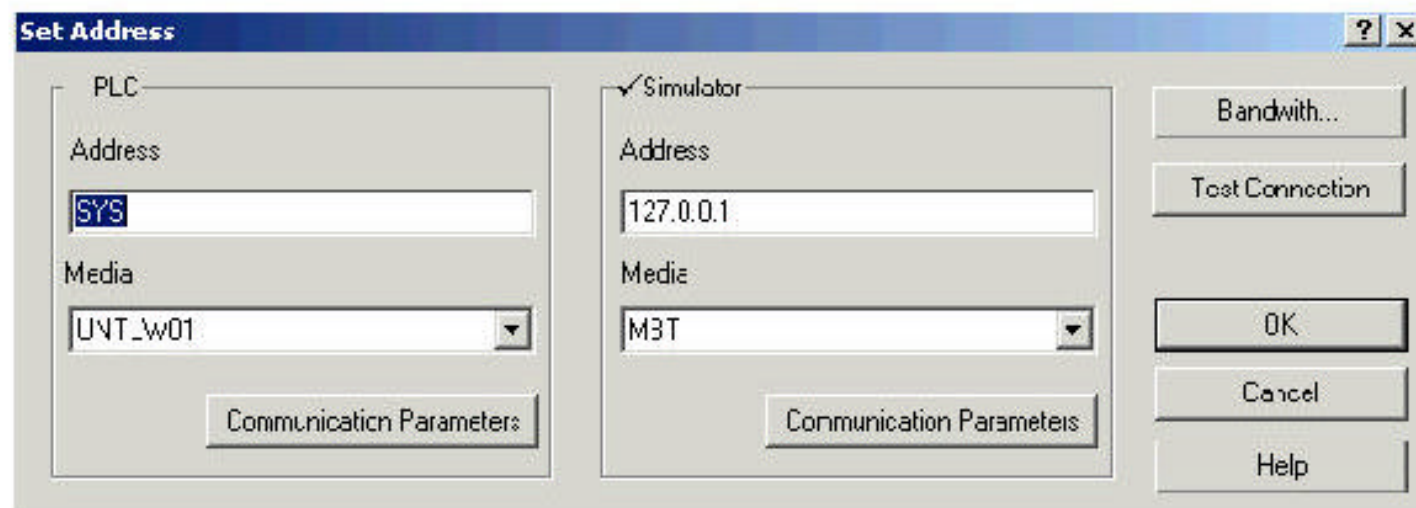
● Generador (cont.)

- Generador de proyectos
 - ▶ Creación de un archivo que puede ser bajado a el PLC o al simulador
- Regeneración de todo el proyecto
 - ▶ Toma en cuenta cualquier cambio hecho al proyecto existente.



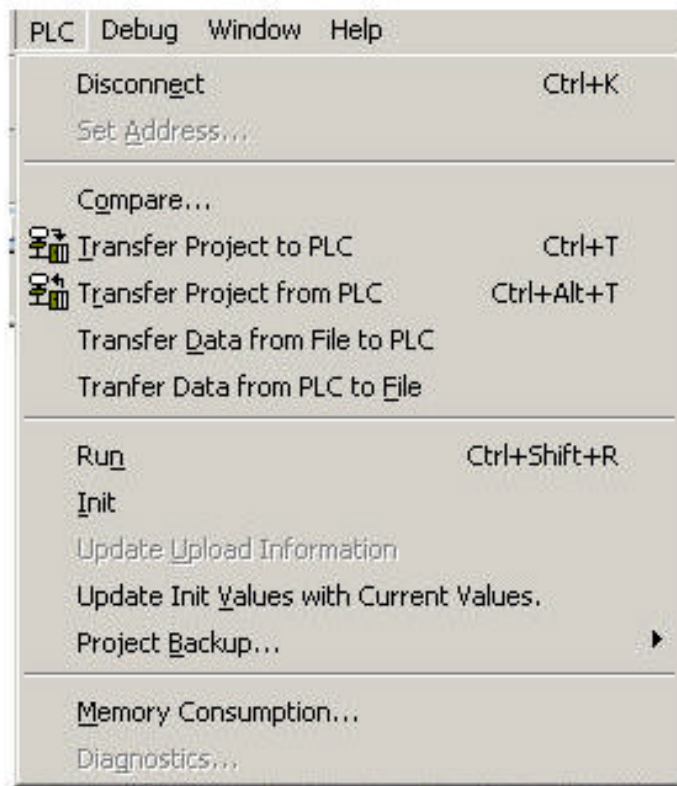
● Conectando a el PLC

El primer paso para conectarse al PLC o al simulador es definir la dirección del PLC.



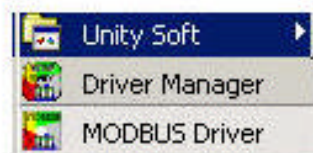
El boton "Comunication Parameters" permite acceder directamente el controlador del driver utilizando la función "Driver Settings"

● Modo conexión



- En el modo Conectado Modo (online)
 - ▶ Se puede realizar una comparación entre el proyecto configurado en la PC y el proyecto del PLC
 - ▶ El proyecto puede ser transferido hacia el PLC o del PLC a la PC .
 - ▶ Los Datos pueden ser transferidos hacia el PLC o desde el PLC.
 - ▶ Funciones de respaldo como comparación del Respaldo, Salvar , Restaurar, o borrado.
 - ▶ El PLC puede ser puesto en modo Run o Stop o inicializado
 - ▶ Se puede ver en cualquier momento la memoria utilizada

● Simulador Unity



- El simulador Unity puede ser utilizado para:
 - ▶ Realizar la Simulación de la operación del PLC Quantum Premium
 - ▶ Permite la Búsqueda de errores en el programa
- Iniciando el Simulador del PLC.
 - ▶ Lanzar el simulador desde el menú de Inicio (Programa-> Schneider Electric-> Unity Soft-> PLC Simulator)
 - ▶ Lanzar el simulador desde el atajo del escritorio
 - ▶ Nota: *El simulador puede ser iniciado con software Unity Pro en modo Run .*

Módulo 3.: Unity Pro

Sección 8 : Probando una aplicación

Página 6/8

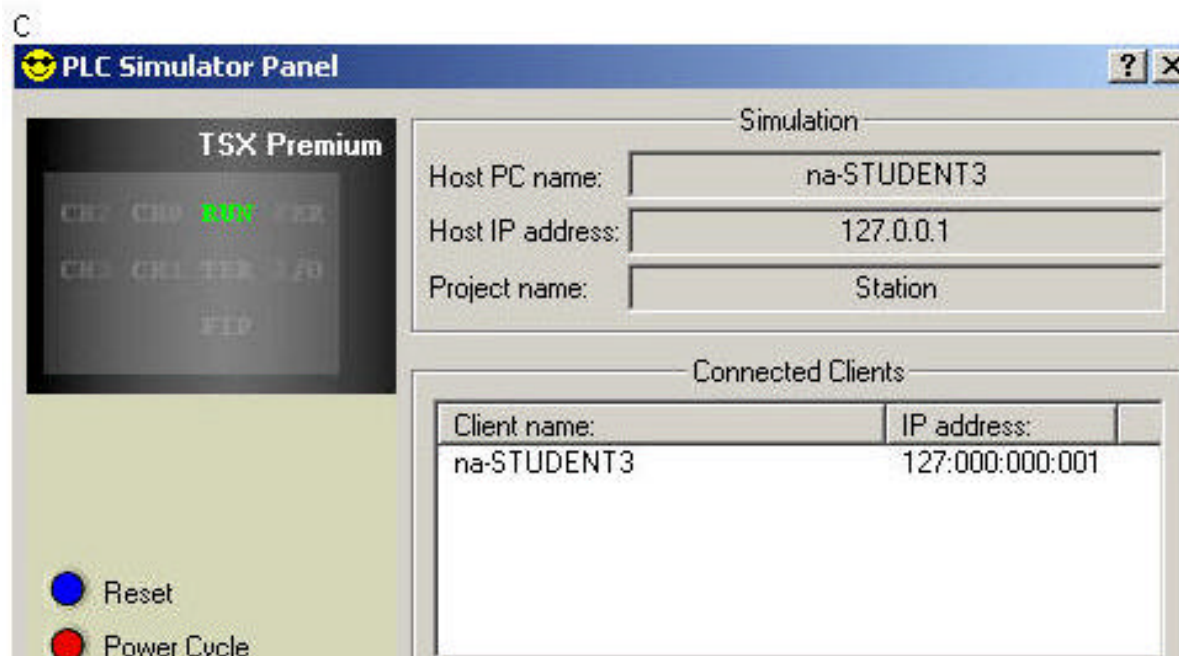
● Simulador Unity

- Cargando una aplicación en el Simulador
 - ▶ Desde la barra del menu de Unity Pro seleccionar PLC-> Connect
 - ▶ Haga un click y arrastre el archivo de la aplicación hacia el icono del simulador si el simulador no ha arrancado
 - ▶ Haga un click y arrastre el archivo de la aplicación para abrir el simulador si el simulador no ha arrancado.

● Panel de control del simulador.

Haga un click en el icono del simulador en la función "System tray" y seleccione el panel del simulador para lanzar el panel del simulador.

- Cuando el proyecto es transferido a el simulador el display muestra una imagen del PLC configurado. Esta imagen muestra:



- ▶ El estatus del PLC, por ejemplo Run, Error, Error de E/S etc..
- ▶ El nombre del Host PC y su dirección IP .
- ▶ El nombre del proyecto que corre en el simulador .
- ▶ El nombre y la dirección IP de los clientes conectados.

● Simulador del Tiempo

Simulator Timing

CPU Usage

Process:	0%	16 h
Overhead:	0%	16 h
<input checked="" type="radio"/> Mast:	0%	16 h
<input type="radio"/> Fast:	0%	16 h
<input type="radio"/> Aux0:		0 s
<input type="radio"/> Aux1:		0 s
<input type="radio"/> Aux2:		0 s
<input type="radio"/> Aux3:		0 s
<input type="radio"/> Evt0:		0 s
<input type="radio"/> Evt113:		0 s
<input type="radio"/> Evt11:		0 s

Display: Absolut Relative

Simulator Priority

Realtime (use with care) Actual:

High

Normal

Low

User Task Data

Scantime for: **MAS-**

Sleeptime for: **MAS-**

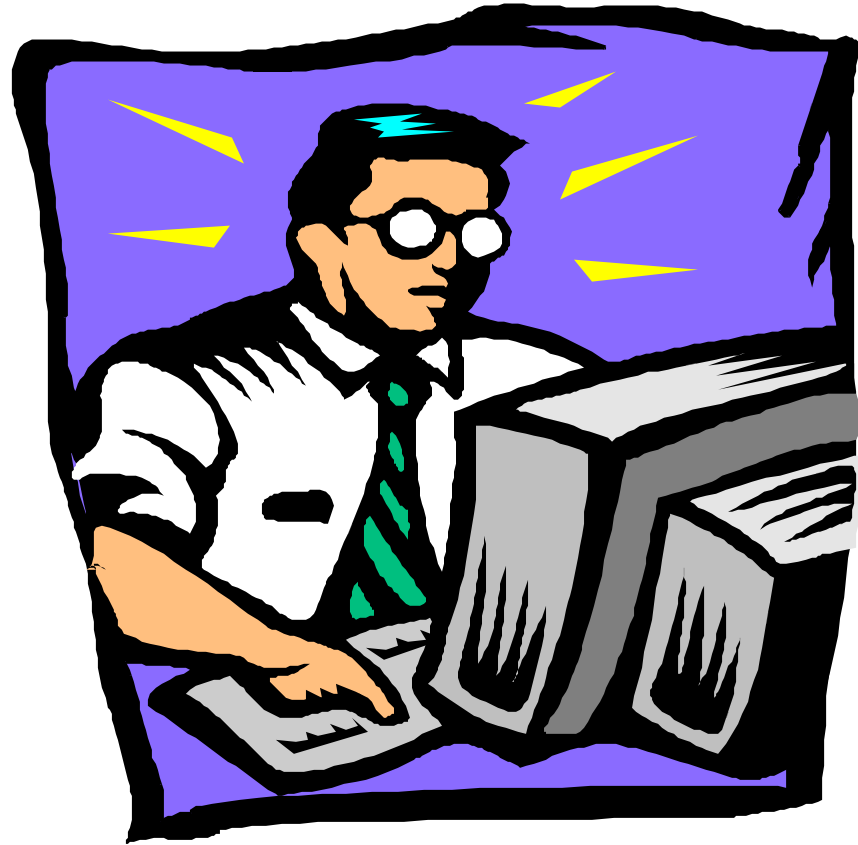
Minimum Sleep Time [ms]

Actual: Set Point:

- Esta ventana muestra las estadísticas del simulador como el tiempo de barrido, la carga del procesador y sobre carga en porcentaje.
- Simulador de prioridad y mínimo tiempo de inactividad puede ser ajustado para optimizar el tiempo del simulador

Nota : para mayor información de estos ajustes vea el menu de ayuda o los manuales del usuarios

Ejercicio 7



Módulo 3.9

Lenguaje IEC : Ladder

● Panorama

- La estructura de una sección de diagrama Ladder corresponde a una línea horizontal con una serie de contactos que activan una bobina o relevador.
- La línea vertical de la izquierda del editor corresponde a la fase L (línea) del diagrama de escalera o ladder. Los objetos que están ligados a esta línea están unidos a la fuente de poder. Unicamente los objetos conectados a la fase son procesados, la línea derecha entonces corresponde al neutro
- Cumple con el estandar IEC 61131-3
- Un grupo de objetos que estan ligados entre si son llamados red o línea.
- La programación del programa LD está orientado a celdas, unicamente un objeto puede ser ubicado en una celda.

● Panorama (cont)

- Una sección LD consiste en una página
- Esta página está dividida en una cuadrícula dividida en columnas y líneas
- Pueden ser definidas de 11-64 columnas y 17 - 2000 líneas para secciones LD.
- La secuencia del proceso de un objeto individual en una sección LD es determinado por el flujo de datos entre la sección. Las líneas conectadas a la izquierda de la pantalla son procesadas de arriba a abajo. Las líneas que son independientes una de otra dentro de una sección son procesadas en el orden de posición (de arriba a abajo)

Ver comentarios para objetos de LD y notas de programación

● Panorama

Objetos del diagrama de escalera o Ladder Diagram:

Los objetos ayudan a dividir las secciones del programa en un número de:

- Contactos
- Bobinas.
- EFs y EFBs (Elementary Functions y Elementary Function Blocks)
- Procedimientos
- Elementos de control
- Bloques de operación y comparación que representan una extensión de IEC 61131-3

Estos objetos pueden ser conectados entre si por medio de:

- Enlaces o
- Parámetros actuales (unicamente FFBs)

Los comentarios pueden ser definidos por la lógica de la sección con objetos texto.

Notas adicionales al programa:

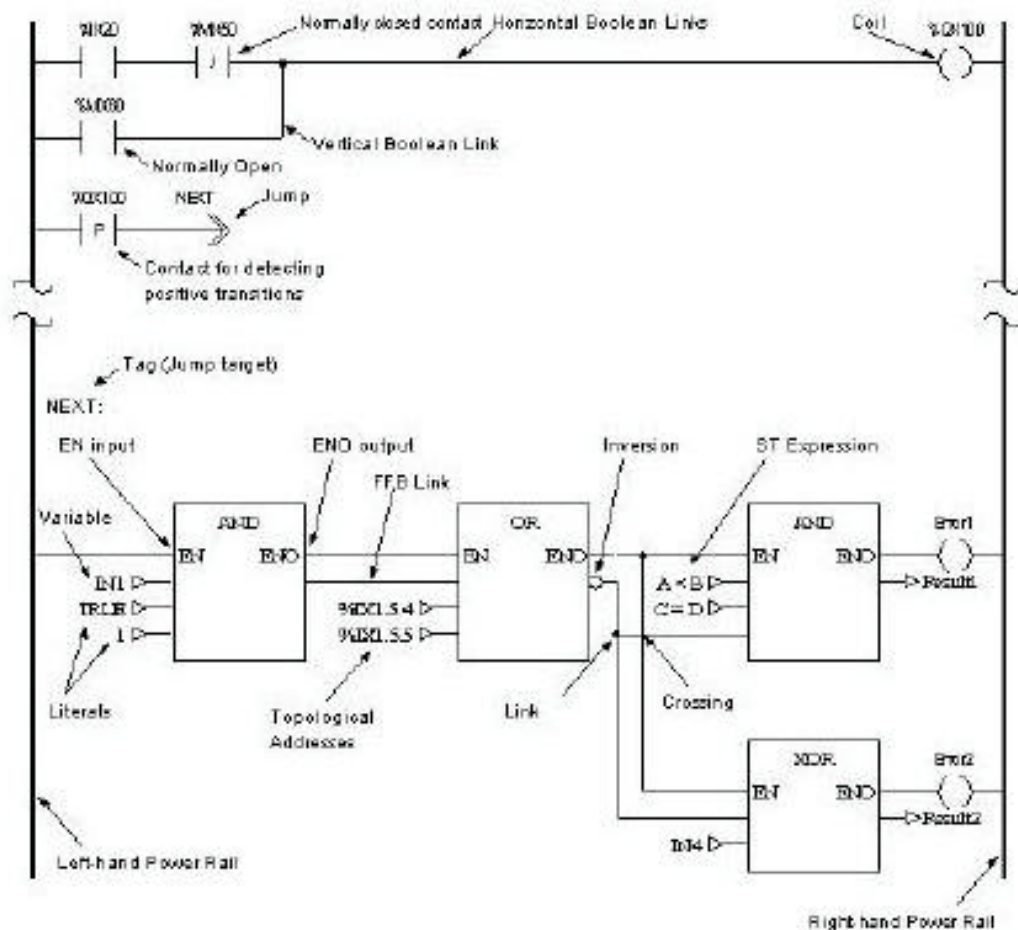
- Los objetos del lenguaje de programación LD permiten estructurar una secuencia con un número de contactos, bobinas y EFBs (elementary functions, elementary function blocks, Derived function blocks)

Módulo 3 : Unity Pro

Sección 9: Lenguaje IEC: Ladder

Página 3/21

● Ejemplo de sección LD



Módulo 3 : Unity Pro

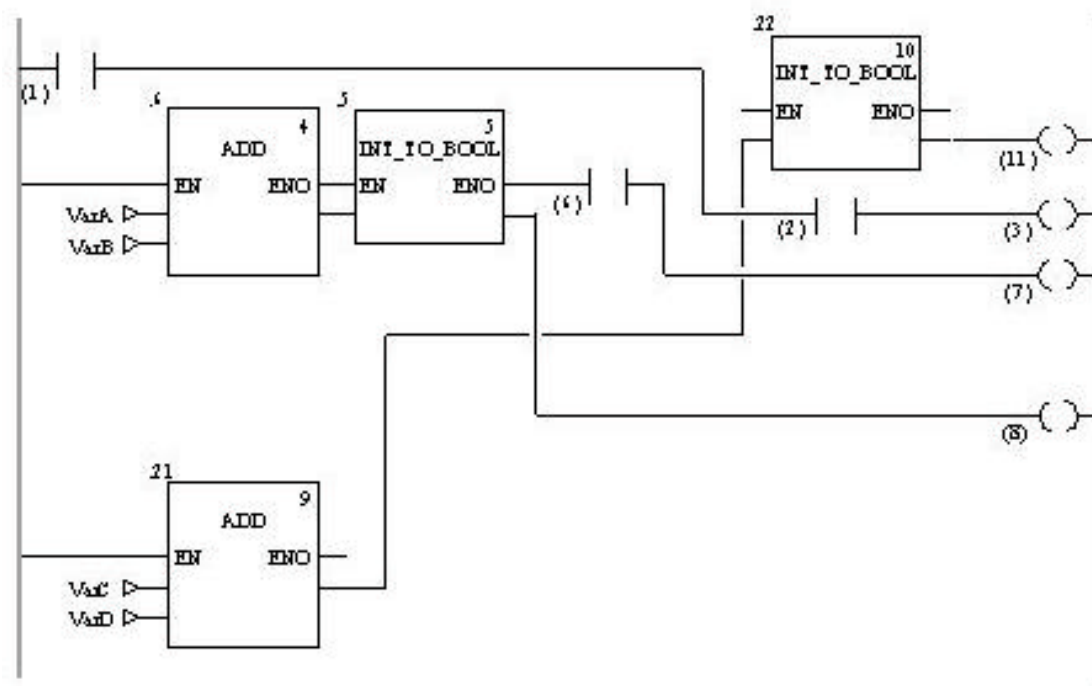
Sección 9: Lenguaje IEC : Ladder

Página 4/21

[Comentarios](#)
[Imprimir](#)

● Ejecución de secuencias

Example of the Execution Sequence of Objects in an LD Section:



El el diagrama se muestra la secuencia de ejecución de los objetos.

La secuencia de ejecución está indicada por los números.

Nota: Los números de ejecución para los contactos y bobinas no se muestran normalmente como aparece en el dibujo.

Vea comentarios para una mejor descripción de la secuencia de ejecución.

Prioridades

Prioridades definiendo el flujo de control dentro de una sección:

Prioridad	Regla	Descripción
1	Enlace	Los enlaces tienen la más alta prioridad definiendo el flujo de control dentro de una sección LD.
2	Red por red	El procesamiento en una red es terminado completamente antes de iniciar el procesamiento de otra red.
3	Secuencia de salida	Las salidas del mismo bloque función o salidas de enlaces verticales son procesadas de arriba a abajo.
4	Línea por línea	Baja prioridad. La ejecución de la secuencia de una red en la cual únicamente está enlazado a la parte izquierda de la pantalla, es determinada por la secuencia gráfica de arriba a hacia abajo. (únicamente aplica si ninguna de las otras reglas es aplicada).



● Cambiar la secuencia de ejecución

El orden de ejecución en una red y el orden de ejecución de objetos en una red está definido por un cierto número de reglas.

En algunos casos, el orden de ejecución sugerido por el sistema debe ser cambiado

El procedimiento para definir o cambiar la secuencia de ejecución de la red es el siguiente:

- Utilizar los enlaces en lugar de los parámetros actuales
- Posición de la red

El procedimiento para cambiar la secuencia de ejecución de un objeto en la red es el siguiente:

- Posicionar los objetos

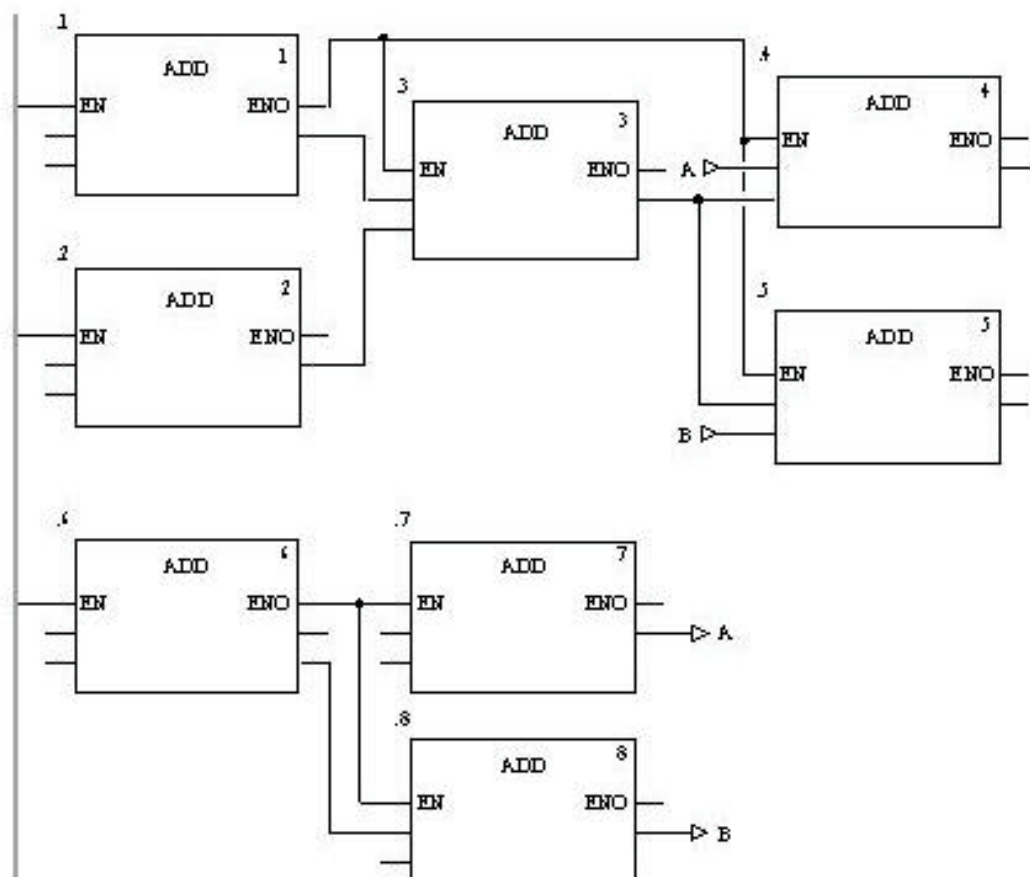
Ver ejemplos de estas reglas en los comentarios

● Ejemplos de reglas

Situación original

La siguiente representación muestra dos redes en las cuales la secuencia de ejecución está definida por su posición en la sección.

Sin tomar en cuenta los bloques 0.4/0.5 y 0.7/0.8 requiere otra secuencia de ejecución .



Módulo 3 : Unity Pro

Sección 9: Lenguaje IEC : Ladder

Página 6/21

Comentarios

Imprimir

● Enlaces diagrama Ladder

Los enlaces son conexiones entre objetos LD (ladder) (contactos, bobinas y FFBs etc.).

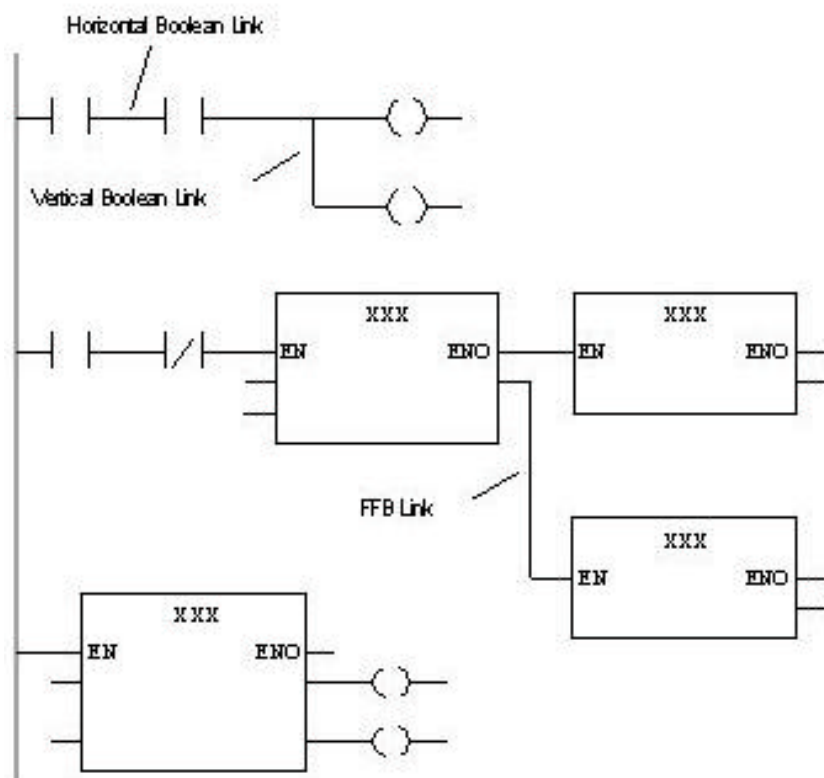
Hay dos tipos de enlace:

- Enlace Boleano - consiste en uno o más segmentos ligados a objetos Boleanos (contactos, bobinas).

Hay dos tipos de enlaces Boleanos:
Enlaces Boleanos Horizontales-permiten hacer una lógica secuencial con contactos y bobinas.

Enlaces Boleanos Verticales - permiten conectar contactos en paralelo.

- Enlace FFB - son enlaces combinados horizontales y verticales que conectan las entradas y salidas de los FFB con otros objetos



● Notas de programación

Notas generales de Programación

Attention should be paid to the following general programming notes:

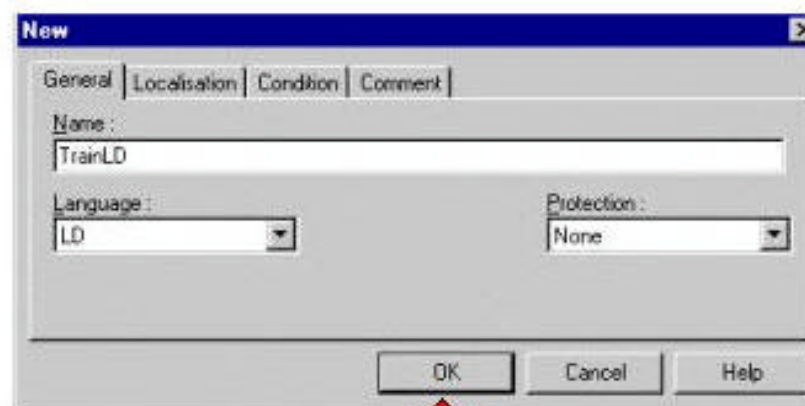
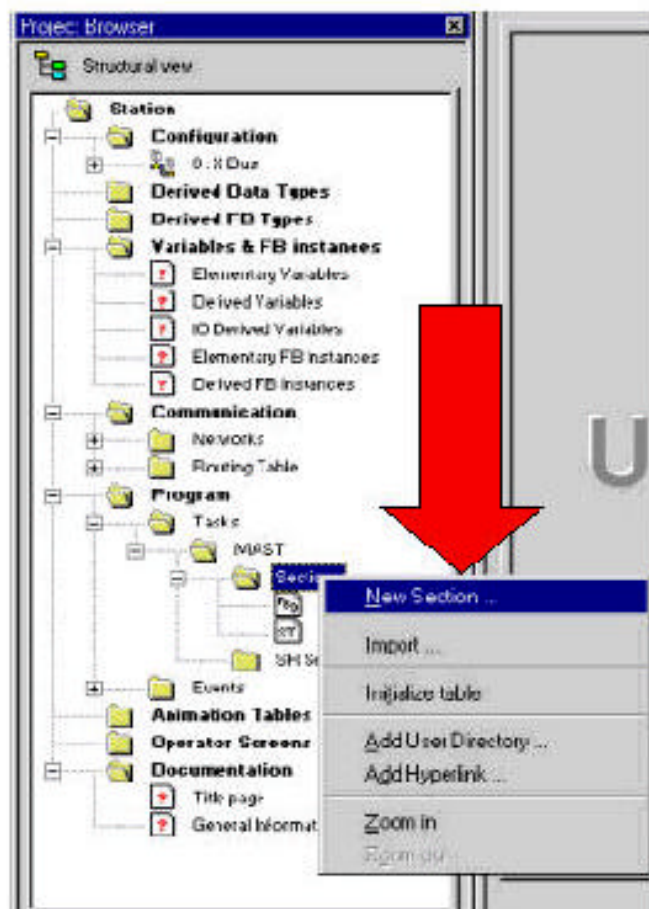
- Los tipos de datos de entrada/salida que deban estar ligados deben ser los mismos.
- Los enlaces entre los parámetros con longitud de variables (ejem. ANY_ARRAY_INT) no están permitidos
- Varios enlaces pueden ser conectados con una salida sin embargo solamente un enlace puede estar conectado al elemento final de salida.
- Elementos no conectados se especifican como "0" por default.

Notas de programación de enlaces Boleanos

- Sobreponer enlaces booleanos no está permitido.
- El flujo de control es de izquierda a derecha
- Si dos enlaces booleanos se cruzan estos se conectarán automáticamente entre si, dado que el cruzamiento de enlaces no está permitido.

● Creación de una seccion en lenguaje de escaleras

Ver comentarios



● **Creando una sección de diagrama en escaleras Ladder.**

Enlaces no unidos, bobinas , y entradas/salidas de FFBs son asignados por default el valor "0".

Una grán cantidad de lógica de Macros pueden ser ubicados en la sección LD, También el texto puede ser ubicado en la sección LD.

Los objetos de Texto no pueden ser sobrepuestos en otros objetos, pero estos pueden ser sobrepuestos en los enlaces.

Los objetos de Texto no ocupan espacio en los PLC Quantum porque en el PLC no se cargan los textos .

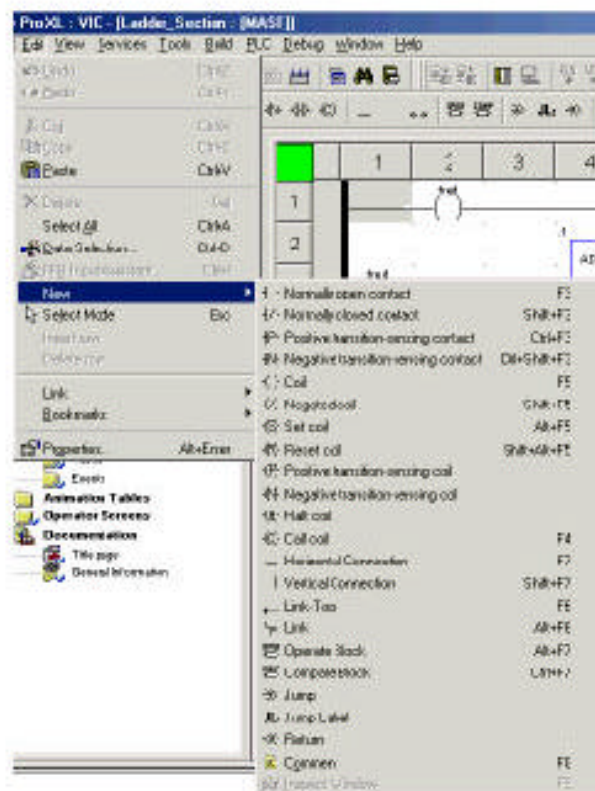
Módulo 3 : Unity Pro

Sección 9 : Lenguaje IEC : Ladder

Página 8/21

● Programación de objetos

- Son 3 metodos de acceso a la edición del diagrama de escaleras:
 - ▶ Hacer un Click en "Edit", y luego **"Nuevo"**

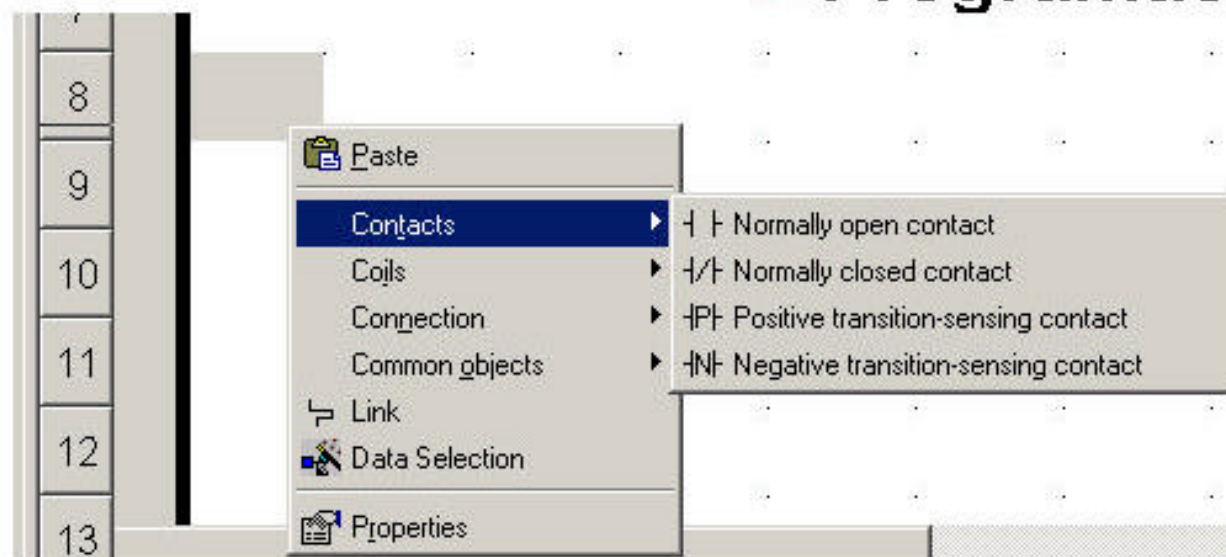


● Programación de objetos - Barra de herramientas



- hay 3 maneras de acceder la programación de objetos en el lenguaje de escaleras:
 - ▶ Haga un Click en “Edición”,y luego en “Nuevo”
.....
 - ▶ Use la herramienta de objetos del diagrama de escaleras

● Programación de objetos



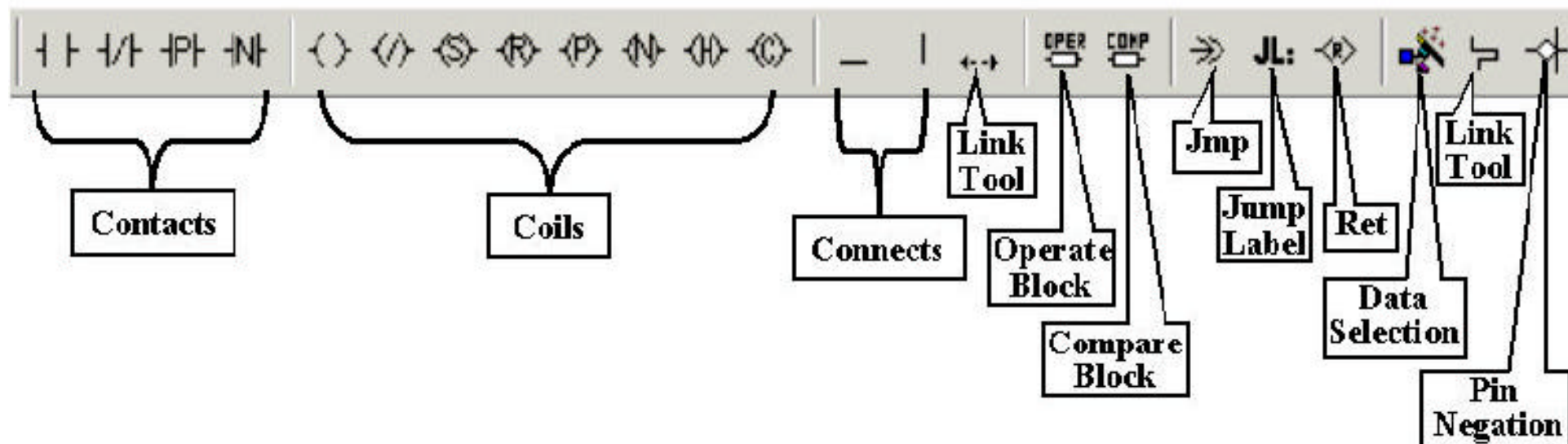
- Hay 3 maneras de acceder los objetos del diagrama de escaleras:
 - ▶ haga un Click en “Edición”, y luego en “Nuevo”
.....
 - ▶ Utilice la barra de herramientas del editor de objetos en lenguaje de escaleras.
 - ▶ Ponga el cursor en cualquier celda vacia, haga un click derecho para seleccionar los objetos.

Module 3 : Unity Pro

Sección 9 : Lenguaje IEC : Ladder

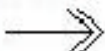

Página 11/21

● Herramienta de programación de objetos



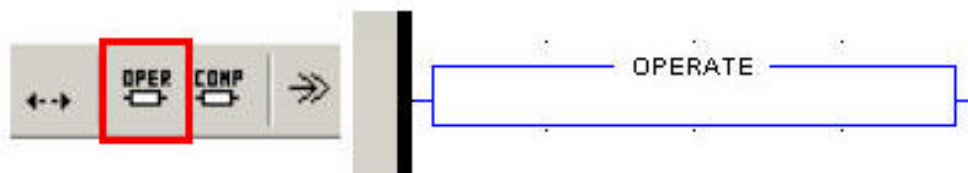
● Elementos de control

Los elementos de control son utilizados para ejecutar saltos entre una sección LD y el regreso de una subrutina (SRx) o un bloque de función derivado (DFB) hacia el programa principal.

Designación	Representación	Descripción
Jump	NEXT 	Cuando el estatus de un enlace izquierdo es 1, el salto se realiza a la etiqueta (en la sección actual). Para generar un salto incondicional, el objeto del salto debe ser ubicado directamente en el lado izquierdo. Para generar un salto condicional, el objeto del salto es ubicado al final de una serie de contactos.
Tag	LABEL:	El Tags (dirección del salto)está indicado como un texto con doble punto al final. El texto está limitado a 32 caracteres y debe ser único en toda la sección entera. El texto debe estar conforme a las convenciones de nombres . Las etiquetas de salto pueden ser unicamente ubicadas en la primer celda directamente a la derecha Nota: Las etiquetas de salto no pueden cortar las redes de diagramas.
Return		Cualquier subrutina y cada DFB (bloque de función derivado) es activo antes de ser procesado y regresa al programa principal del que fué llamado. Si la subrutina /DFB es dejada preauturamente, el regreso de la subrutina o DFB (bloque derivado de función) hacia el programa puede ser forzado por medio del objeto de retorno o regreso. Los objetos de retorno pueden ser utilizados en subrutinas o en DFB (bloques de función derivados). No pueden ser utiizados en el programa principal. Para generar un regreso condicional, el objeto de regreso debe se ubicado al final de una serie de contactos o condiciones.

● Bloques de operación y comparación

- En adición a los objetos definidos en IEC 61131-3 existen otros dos bloques para ejecutar instrucciones ST y expresiones ST para operaciones de comparación simple .Estos bloques están solamente disponibles en el language de programación LD. y son llamados:
- Bloque de operación (OPERATE Block)



- Bloque de comparación (COMPARE Block)



● Bloques de operación

Los bloques de operación están únicamente disponibles en el lenguaje de programación LD (ladder) .

Estos son utilizados para ejecutar una instrucción ST (texto estructurado). Cuando el estatus del enlace izquierdo es 1, la instrucción ST en el bloque es ejecutada.

Todas la instrucciones ST son permitidas excepto las instrucciones de control:

- (RETURN,
- IF,
- FOR
- JUMP,
- CASE,
- etc.)

Para los bloques de operación, el estado del enlace izquierdo es pasado al enlace derecho (cualquiera que sea el resultado dela instrucción ST).

Los bloques de operación pueden ser ubicados en cualquier celda. Un bloque de operación requiere 1 línea y 4 columnas.

● Notas sobre bloques de operación


Notas adicionales

Un bloque operación puede contener hasta 4096 caracteres. Si no se pueden desplegar todos los caracteres, entonces se debe poner al inicio de los caracteres un apóstrofe (...).

Un bloque operación crea sus enlaces automáticamente con los objetos asociados, si son objetos tipo Boleanos no debe haber espacios entre ambos.

Activando un bloque operación

Un bloque operación puede ser definido de las siguientes maneras:

- Utilizar el menú de comando **Edit -> New -> Operate Block**,
- Utilizar el menu de comando **Common Objects -> Operate Block**,
- Utilizar la combinación de teclas **Alt+F7**, o
- Seleccione el símbolo .

Una vez activo el bloque el cursor indica el siguiente símbolo .

Nota

En el menu **Tools -> Options** y en la carpeta **Data and Languages**, si está seleccionada la casilla, **"Automatically assign a variable to a new graphical object"**, entonces el cuadro de diálogo aparecerá automáticamente cuando el objeto es posicionado.

● Bloques de comparación

- La comparación de Bloques está disponible únicamente es el lenguaje LD.
Esta función es utilizada para ejecutar las funciones de comparación (<, >, <=, >=, =, <>) en el lenguaje de programación ST(texto estructurado) la misma funcionalidad es posible también utilizarlas.
- Si el estado del enlace izquierdo es 1 y el resultado de la comparación es 1, el estado del enlace derecho es 1
- El bloque de comparación puede ser posicionado en cualquier celda exepcto al final del lado derecho.
- Un bloque de comparación requiere 1 línea, y 2 columnas.
- *Al posicionar un bloque, automáticamente se crean las conexiones con los objetos que se encuentren en la derecha e izquierda. Ver comentarios adicionales*


● Comparación de Bloque de notas

Notas adicionales

Un Bloque de comparación puede contener hasta 4096 caracteres. Si no todos los caracteres pueden ser visualizados, entonces al inicio de la secuencia del carácter debe llevar apostrofe (...).

Activando un bloque de comparación

Un bloque de comparación puede ser configurado en las siguientes maneras:

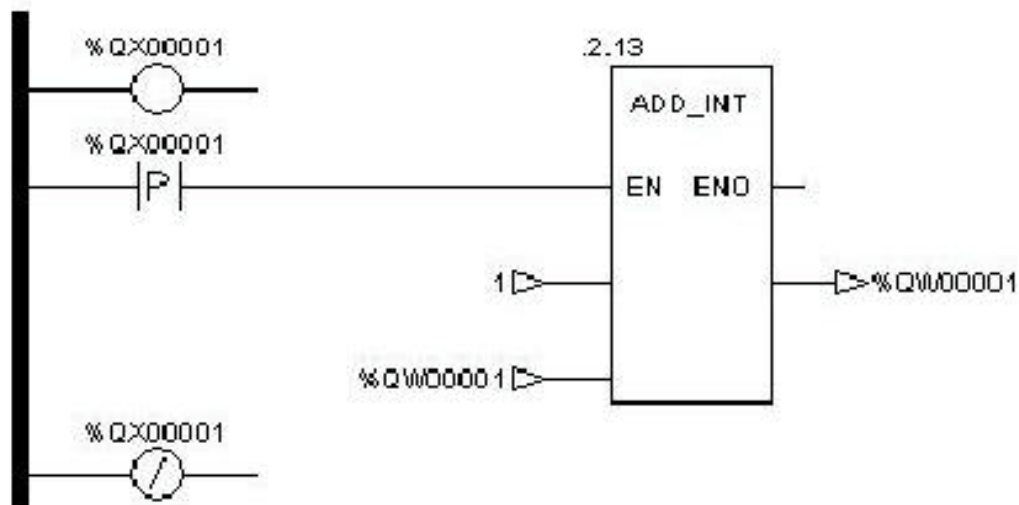
- Utilise el menú de comando **Edit -> New -> Compare Block**,
- Utilise el menú **Common Objects -> Compare Block** en el menú,
- Utilise la combinación de teclas **Ctrl+F7**, o
- Seleccione el símbolo .

Nota

Si en **Tools -> Options** en la carpeta **Data and Languages**, si está activa la casilla, **Automatically assign a variable to a new graphical object**, entonces las propiedades del cuadro se abren automáticamente cuando un objeto está definido.

● Objetos de detección de frentes

Ver comentarios



● **Objetos de detección de frentes:**

Diferentes implementaciones de objetos requieren el uso de variables StateRam (registros 0x/1x). Bajo la condición de que varios accesos de escritura a los registros 0x/1x son posibles en un scan diferentes comportamientos pueden aplicar.

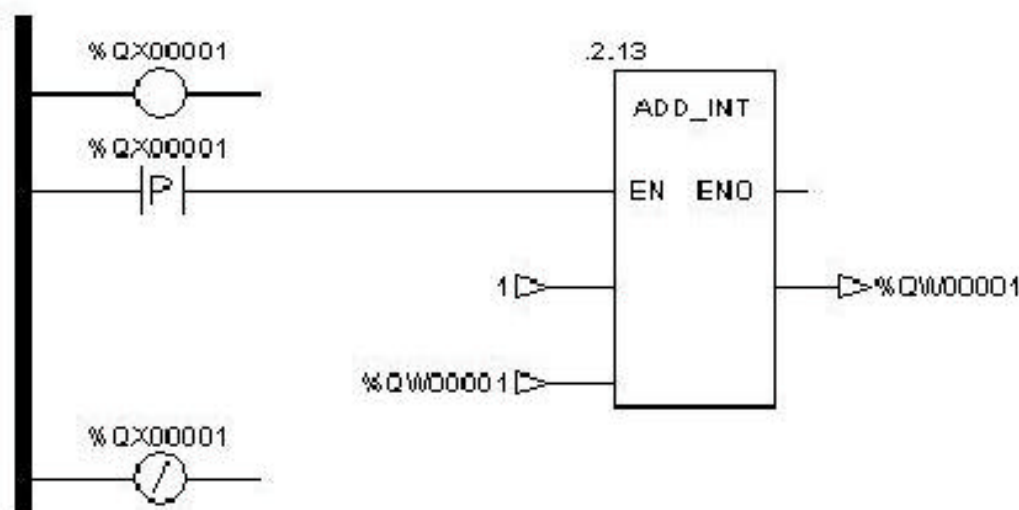
- **Objetos afectados:**
 - contacto con frente ascendente
 - contacto con frente descendente

En Concept la bobina de los objetos de transición son disponibles como EFB llamados RTRIG/FTRIG. La información necesaria para la detección del estado anterior es asociada con el llamado del EFB y evaluada únicamente en su ciclo de ejecución.

En Unity los objetos de detección de frente ascendente o descendente son posibles únicamente utilizando variables EBOOL . El estado anterior es asociado directamente con el tipo de variables booleanas. Los registros %M y %I son siempre del tipo EBOOL. La actualización del estado anterior ocurre en cada acceso de escritura a la variable EBOOL.

● Ejemplos de objetos de detección de frentes

Ver comentarios



Valores iniciales: %QX1 es FALSO. %QW=1

● Ejemplo de detección de objetos de frentes

Concept:

- Primer ciclo: %QW1 es incrementado de 1 de 2.
- Siguiendo ciclo (s): %QW1 mantiene el valor 2 hasta la transición del frente siempre compara el valor anterior (verdadero) con el actual (Verdadero)

Unity:

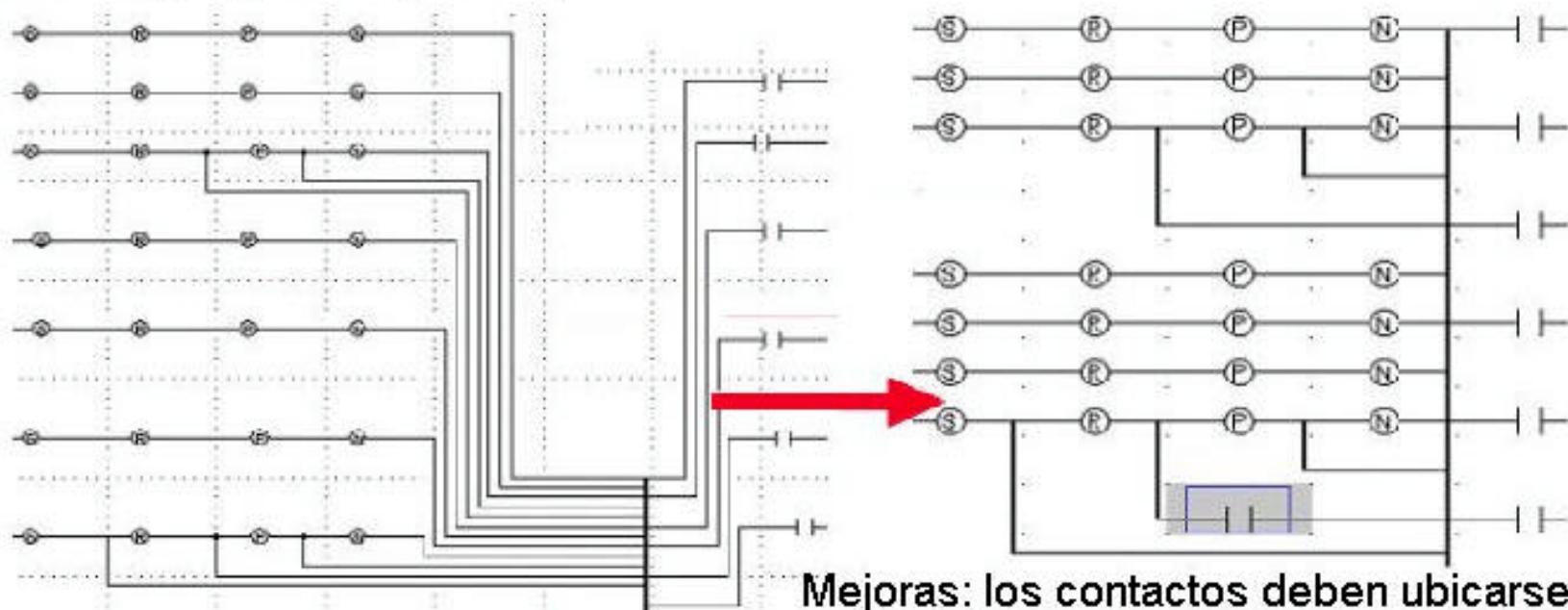
- A cada ciclo(s): %QW1 es incrementado por uno 1. La transición siempre es reconocida en el último acceso a la escritura cuando la transición es positiva

Precaución:

- Utilice la detección de objetos si es posible con las variables 0x/1x . No escriba a los registros 0x más de una vez en un ciclo si el registro está ligado a un objeto LD con detección de frente ascendente descendente (ejem. ---|P|---) Para asegurar el mismo comportamiento en Concept y Unity.
- En Unity utilizando DFBs, puede substituir la funcionalidad de detección de frente con los EFBs, R_TRIG y F_TRIG EFBs .

● Diagrama Ladder

Respetar las guías de ubicación incluyendo los objetos y enlaces.
(Este es un ejemplo exagerado)



Mejoras: los contactos deben ubicarse como la figura

Ver comentarios adicionales

● Respetar la representación de la malla incluyendo los objetos los enlaces y puentes.

Una conversión del gráfico es hecha por el convertidor Unity el cual puede resultar en ajustes gráficos . Abajo encontrará unas reglas del proceso de conversión con la finalidad de minimizar y un reajuste durante el proceso de futuras conversiones

- 1) Representación del Malla en Unity (como en Concept 2.2 o superior). Los enlaces también respetan la representación gráfica de la malla.
- 2) Los elementos más cercanos a la parte derecha del editor gráfico son asociados al riel vertical derecho.
- 3) Enlaces cruzados (sin conexión) no están permitidos en Unity. En este caso de conflictos de cruzamineto, se insertan elementos adicionales en el drigramma Unity
- 4) Optimizaciones como eliminación de espacios en blanco a arreglos de alguna áreas son realizados.
- 5) Las bobinas en la parte derecha unicmante se define una por red.

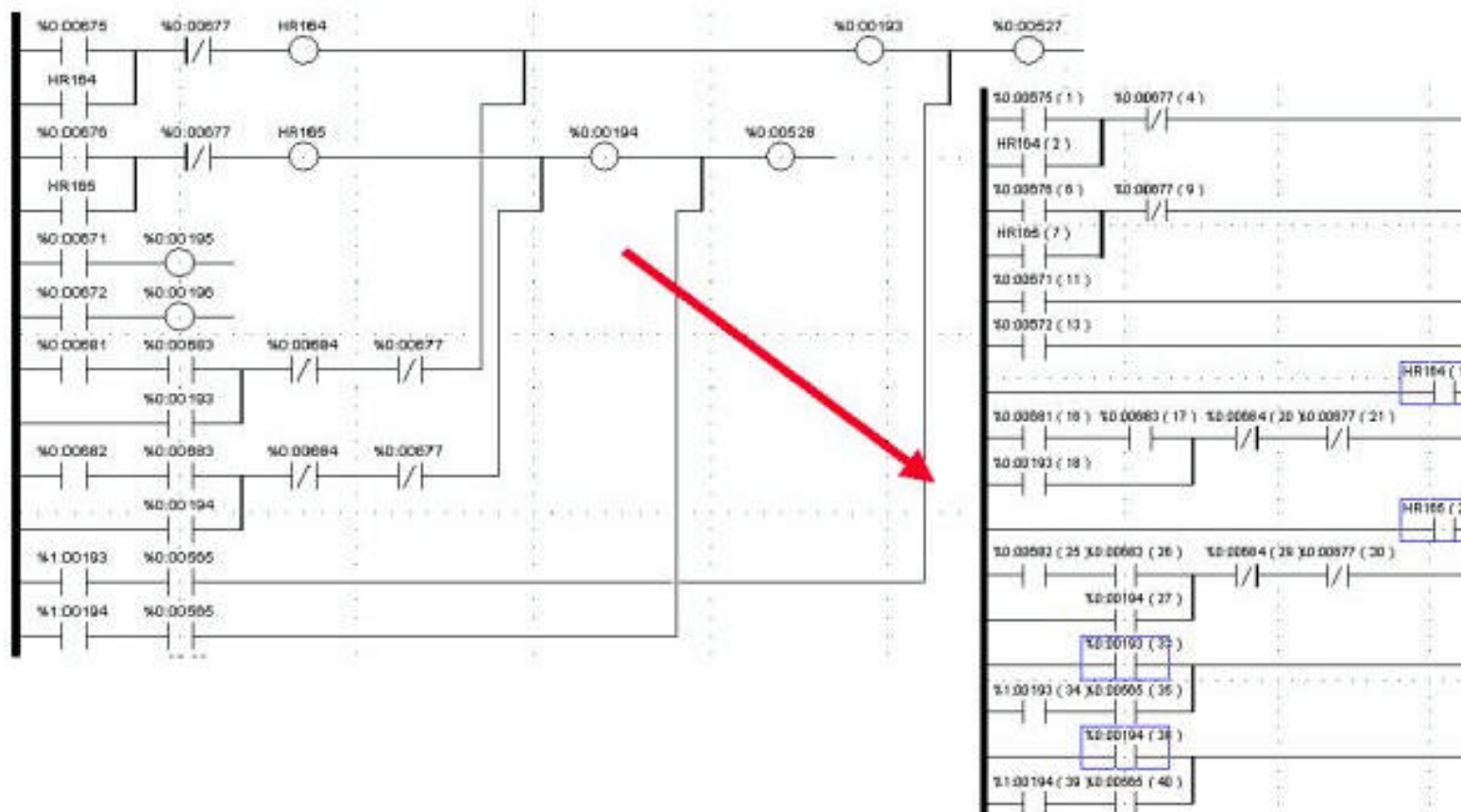


Módulo 3 : Unity Pro

Sección 9 : Lenguaje IEC : Ladder

Página 19/21

● Eliminar cruces entre enlaces



● **Eliminando cruces**

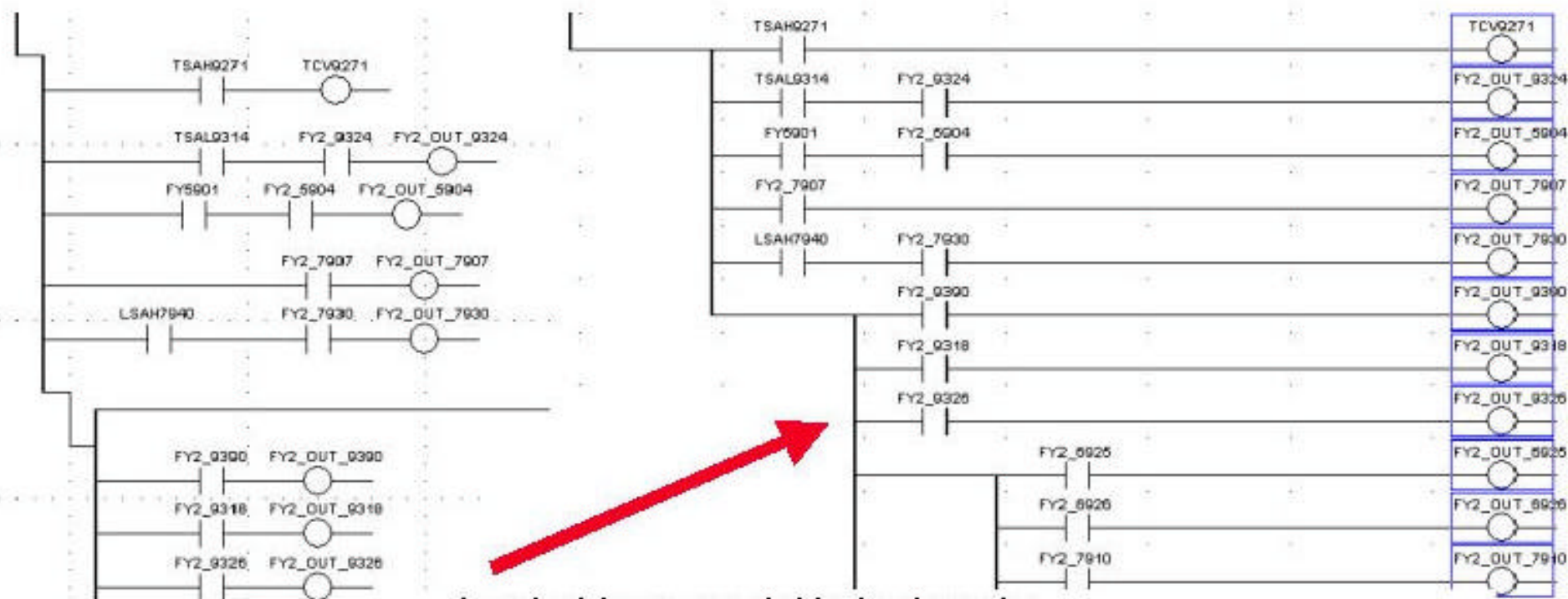
- 1) Para Concept 2.1 o versiones anteriores (free floating graphic): Respeta la malla virtual para las representaciones de los objetos tipo corto circuito o puente . Para Concept 2.2 o posterior (la representación de la malla): Respeta la representación de la malla con los enlaces y puentes o cortos circuitos.
- 2) Ponga al final siempre en la linea, en la misma columna.
- 3) Evite cruzamineot de enlaces y encimado de conexiones
- 4) Mantenga la misma estructura en su aplicación.

Módulo 3 : Unity Pro

Sección 9 : Lenguaje IEC : Ladder

Página 20/21

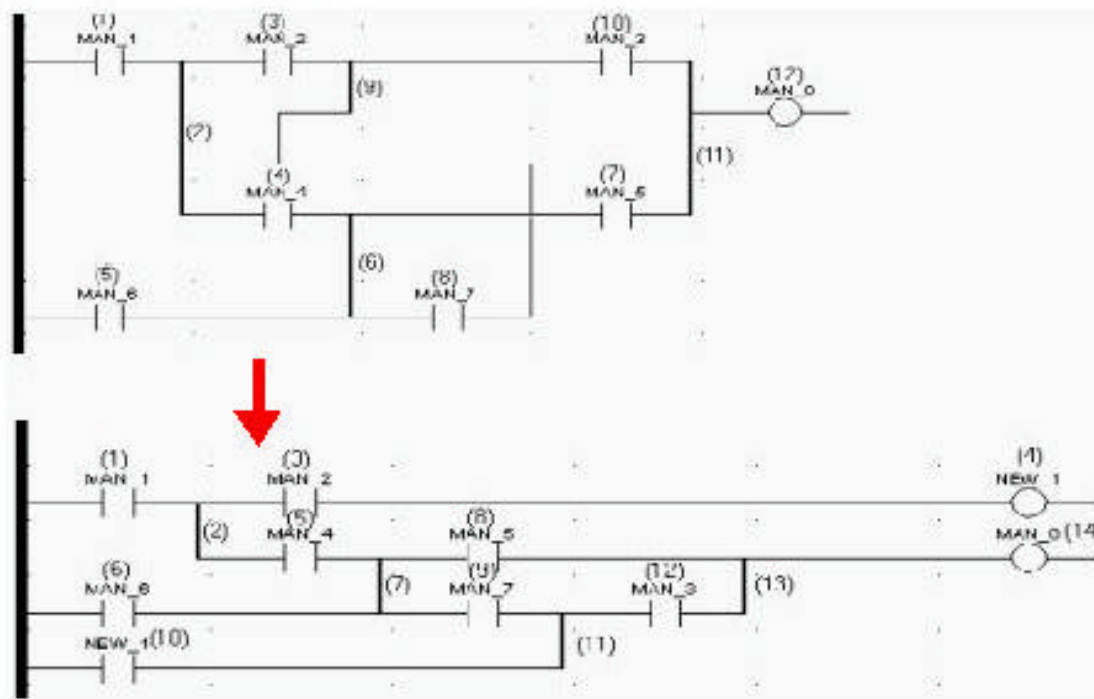
● Evitar islas de enlaces con cortos circuitos



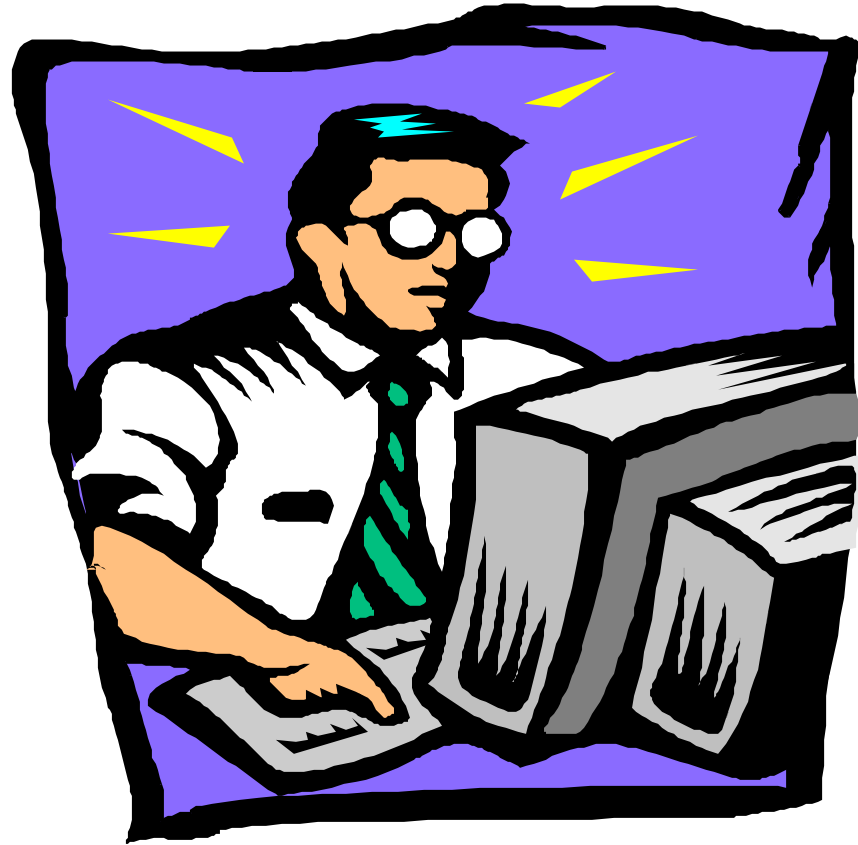
- Las bobinas van del lado derecho

● Evite enlaces cruzados y conexiones encimadas

Ejemplo para evitar conexiones cruzadas



Ejercicio 8



Notas

Notas