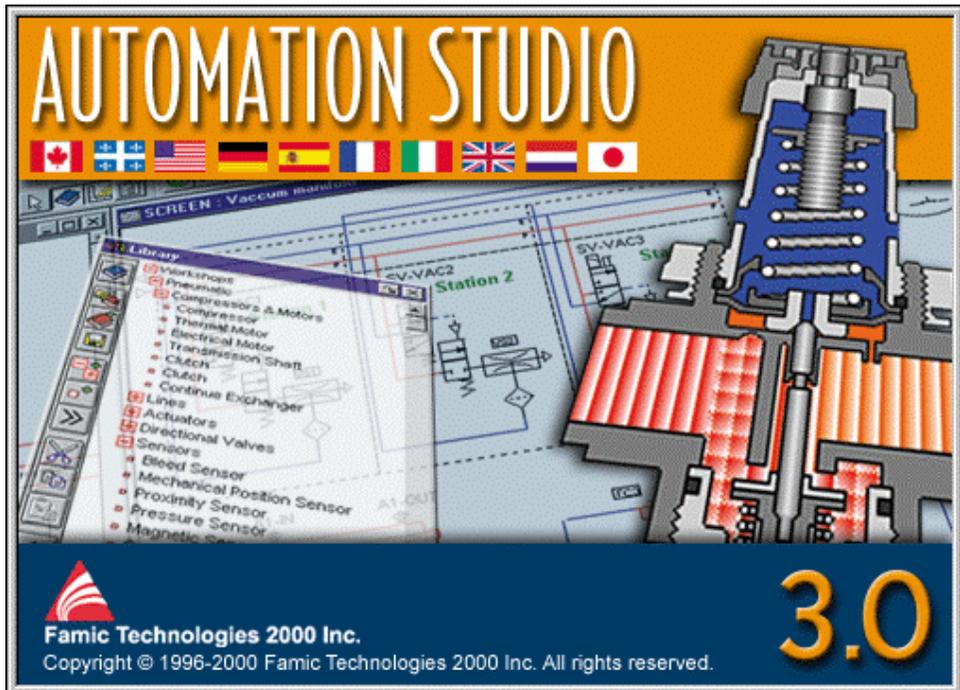


Taller

# GRAF CET

Guía del Usuario



**AUTOMATION STUDIO**

Library

- WorkSpace
- Pneumatics
  - Compressors & Motors
    - Compressor
    - Electrical Motor
    - Transmission Shaft
    - Clutch
    - Continous Exchanger
  - Lines
  - Actuators
  - Directional Valves
  - Sensors
    - Bleed Sensor
    - Mechanical Position Sensor
    - Proximity Sensor
    - Pressure Sensor
    - Magnetic Sensor

SV-VAC2 Station 2

SV-VAC3 Sta

3.0

**Famic Technologies 2000 Inc.**  
Copyright © 1996-2000 Famic Technologies 2000 Inc. All rights reserved.



© **FAMIC® Technologies 2000 Inc.**

Todos los derechos reservados.

Guía del usuario del taller GRAFCET, Automation Studio.

Referencia del documento: FT-DOC-85306, versión 3.0

## **REPRODUCCIÓN**

Toda reproducción parcial o total, de esta guía o del programa, está prohibida sin el consentimiento escrito de **FAMIC® Technologies 2000 inc.**

IBM es una marca registrada de IBM Corporation.

Windows es una marca registrada de MicroSoft Corp.

PNEUSIM es una marca registrada de FAMIC inc.

AUTOMATION STUDIO es una marca registrada de FAMIC Technologies 2000 inc.



---

# Sumario

<b>1</b>	<b>GRAF CET: sintaxis .....</b>	<b>3</b>
1.1	Etiquetas .....	4
1.2	Variables.....	8
1.3	Constantes.....	8
1.4	Operadores de expresiones enteras.....	8
1.5	Operadores de expresiones booleanas .....	9
1.6	Operadores de flancos .....	10
1.7	Operadores relacionales.....	12
1.8	Prioridad de los operadores .....	13
1.9	Verificación de la sintaxis.....	14
1.10	Ejemplos de transiciones .....	16
1.11	Operadores de asignación.....	17
1.12	Acción independiente .....	20
1.13	Temporización.....	21
1.14	Forzados .....	27
1.15	Ejemplos de etapas - acciones .....	28
<b>2</b>	<b>Simulación.....</b>	<b>31</b>
2.1	Disposición de las ventanas .....	31
2.2	Simulación del proyecto .....	32
2.3	Causas de los problemas.....	34
<b>3</b>	<b>Ejercicio .....</b>	<b>37</b>
3.1	Ejercicio – Circuito de control de un taladro.....	37
<b>A.</b>	<b>Glosario .....</b>	<b>41</b>
<b>B.</b>	<b>INDEX.....</b>	<b>49</b>



---

# Introducción

Esta *Guía del usuario del taller GRAFCET* presenta las informaciones necesarias para utilizar el taller con el Sistema de base del programa Automation Studio.

Automation Studio es un programa modular de simulación compuesto por un Sistema de base y por los diferentes módulos de simulación que usted puede agregar.

Los módulos, llamados «talleres», comprenden «librerías» de componentes que permiten realizar circuitos de diferentes tipos – neumáticos, eléctricos, etc – o combinaciones entre ellos.

El Sistema de base comprende las funciones de edición, simulación, gestión de ficheros, impresión y visualización.



---

# 1

## GRAF CET: sintaxis

Un GRAFCET contiene una o varias expresiones compuestas de variables, constantes, de números y de operadores. La sintaxis precisa las reglas que conviene respetar para crear un GRAFCET.

Este capítulo presenta en detalle la sintaxis válida en lo que toca a etiquetas y a la declaración de acciones. Aborda los siguientes temas:

- Etiquetas;
- Variables;
- Constantes;
- Operadores de expresión entera;
- Operadores de expresión booleana;
- Prioridad de los operadores;
- Ejemplos de transiciones;
- Operadores de asignación;
- Acciones independientes;
- Temporizadores;
- Forzados;
- Ejemplos de etapas-acciones.

## 1.1 Etiquetas

Las etiquetas permiten atribuir a todos los símbolos del GRAFCET una referencia material. Esta referencia se lleva a cabo con las etapas, las transiciones y las variables. Cuando estos símbolos son creados (con la excepción de las variables), Automation Studio genera automáticamente un nombre.

De manera general, los nombres son etiquetas y deben respetar las siguientes reglas elementales:

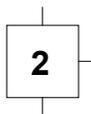
- se lo establece a partir del alfabeto: " a-z, A-Z, 0-9, \_, +, -";
- comienza siempre por una letra;
- debe estar compuesto de 1 hasta 26 caracteres;
- no contiene ningún carácter en blanco (es decir: un espacio);
- no contiene ningún carácter acentuado;
- los caracteres pueden ser en minúscula o en mayúscula. Una vez que Automation Studio lo reconoce, el nombre será presentado siempre en mayúscula.



Si quiere controlar circuitos neumáticos o hidráulicos con un GRAFCET, debe seguir estas reglas para nombrar los componentes de tales circuitos.

### 1.1.1 Etapas

Quando se inserta una nueva etapa, Automation Studio le atribuye un número comprendido entre 1 y 999. Por eso y dentro de un mismo proyecto, cada etapa tiene un número único.

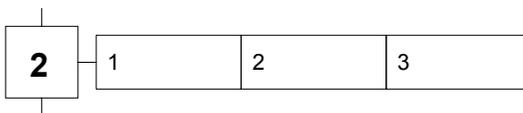


Por convención, si  $n$  representa el número de la etapa, entonces la variable  $Xn$  designa el estado lógico ( $\{\text{activo/inactivo}\}$  o  $\{0,1\}$ ) de la etapa. Por ejemplo,  $X2$  designa el estado lógico de la etapa 2. El estado lógico de una etapa puede ser empleado en una expresión lógica por su nombre  $Xn$ .

## 1.1.2 Acciones

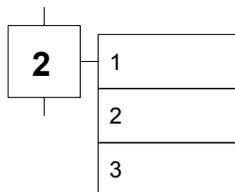
Las acciones deben siempre estar conectadas o bien directamente o bien por medio de otra acción a una etapa.

En el ejemplo siguiente tres acciones están asociadas a la etapa 2.

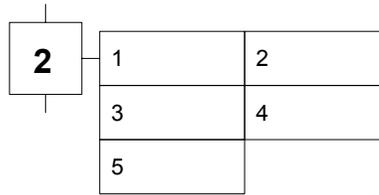


Los números inscritos en las acciones presentadas en los ejemplos de esta sección ilustran el orden de ejecución de las acciones; no representan una sintaxis válida.

Las acciones también pueden estar conectadas verticalmente con una etapa. Así lo muestra el siguiente ejemplo.



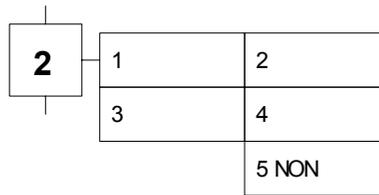
O bien, de manera vertical y horizontal:



Sin embargo, cuando se disponen acciones vertical y horizontalmente, se debe siempre comenzar por la izquierda una nueva línea, como lo muestra el ejemplo precedente.

Si no se respeta esta regla, las acciones de la línea irregular no podrán ser ejecutadas.

El ejemplo siguiente infringe precisamente esta regla y la acción número 5 no será ejecutada.



### 1.1.3 Transición

Cuando se inserta una nueva transición, Automation Studio le atribuye un número comprendido entre 1 y 999. En un proyecto, cada transición tendrá un número único



Por convención, si n representa el número de la transición, entonces la variable  $Y_n$  designa el estado lógico ( $\{\text{verdadero, falso}\}$  o  $\{0,1\}$ ) de la transición. Por ejemplo,  $Y_2$  designa el estado lógico de la transición 2.

Notas:

- La notación «=1» asociada a una transición indica una receptividad siempre verdadera;
- Si la receptividad no es precisada, entonces esto significa que ella es siempre falsa ( notación «=0»)

### 1.1.4 Palabras reservadas

El taller GRAFCET contiene palabras reservadas, es decir que no pueden ser utilizadas para las etiquetas.

- Los nombres que comienzan por X están reservados para las etapas;
- Los nombres que comienzan por Y están reservados para las transiciones;
- Las palabras claves **OR**, **XOR**, **AND** y **NOT** están reservadas para las expresiones booleanas;
- Las palabras claves **F/** y, **T/** están reservadas para los operadores forzadores y de temporización;
- La palabras clave **IF** está reservada para el operador de condición;



Si construye GRAFCET para controlar circuitos neumáticos, eléctricos o digitales, asegúrese de que los nombres de los componentes de dichos circuitos no contengan palabras reservadas.



Cuando se insertan acciones y transiciones en un esquema, las expresiones que contienen serán mostradas con un color diferente si la sintaxis es incorrecta. El color será el mismo para los componentes erróneos. Véase 1.7 Verificación de la sintaxis, en página 39.

## 1.2 Variables

El tipo de una variable depende de la información que contiene. Los tipos de variables son los siguientes:

Tipo de variable	Descripción
Variable booleana	Variable cuyo valor sólo puede ser 0 o 1 (VERDADERO (ON) o FALSO (OFF)).
Variable numérica	Variable cuyo valor es un entero comprendido entre -32768 y +32767.

## 1.3 Constantes

Tipo de variable	Descripción
Constante booleana	Constante cuyo valor sólo puede ser 0 o 1 (VERDADERO (ON) o FALSO (OFF)).
Constante numérica	Constante cuyo valor es un entero comprendido entre -32768 y +32767.

## 1.4 Operadores de expresiones enteras

Los operadores permiten la realización de operaciones enteras sobre variables enteras. Las expresiones enteras enteras dan un resultado de tipo entero. La tabla siguiente describe los operadores.

Operador	Descripción y ejemplo
+	Adición de un valor entero natural (valor positivo o nulo) EJ: Val1:= +25
+	Adición de valores enteros EJ: Val1:= Val2 + Val3
-	Sustracción de un valor entero EJ: Val1:= -25
-	Sustracción de valores enteros EJ: Val1:= Val2 - Val3
/	División de valores enteros (div.euclidiana) EJ: Val1:= Val2 / Val3
*	Multiplicación de valores enteros EJ: Val1:= Val2 * Val3

## 1.5 Operadores de expresiones booleanas

Estos operadores pueden ser utilizados para atribuir el valor de una expresión a una variable o para verificar una condición de transición. Las expresiones booleanas dan resultados de tipo booleano. La tabla siguiente describe estos operadores.

Operador	Descripción
OR	O lógico ej: ls1 OR ls2
XOR	O exclusivo ej: ls1 XOR ls2
AND	Y lógico Ej: ls1 AND ls2
NOT	Negación de un valor booleano Ej: NOT ls1

## 1.6 Operadores de flancos

La expresión formada con un operador de flanco es una expresión booleana que permite aprovechar un cambio de estado.

Un flanco es asimilable a un operador particular que indica si el valor de una variable booleana pasa de 0 a 1 o de 1 a 0.

Hay dos tipos de operadores de flanco: operador de flanco en subida y operador de flanco en bajada. Se los define en la tabla siguiente.

Operador	Descripción
^	<p>Flanco en subida</p> <p>El operador se vuelve VERDADERO cuando el valor de la expresión booleana cambia de FALSO a VERDADERO.</p> <p>Ejemplo: A SI ^B. A toma el valor VERDADERO cuando B pasa de FALSO a VERDADERO</p>
!	<p>Flanco en bajada</p> <p>El operador se vuelve VERDADERO cuando el valor de la expresión booleana cambia de VERDADERO a FALSO.</p> <p>Ejemplo: A SI !B. A toma el valor VERDADERO cuando B pasa de VERDADERO a FALSO.</p>

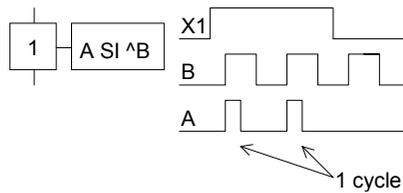


Figura 1-1: Cronograma de una acción con un flanco en subida

## 1.7 Operadores relacionales

Estos operadores permiten comparar variables y/o valores numéricos (números y constantes).

El resultado es booleano: si la comparación se verifica, el resultado es VERDADERO, si no, el resultado es FALSO.

Operador	Descripción
>	La expresión $A > B$ verifica si el valor de A es mayor que el valor de B
>=	La expresión $A \geq B$ verifica si el valor de A es mayor o igual que el valor de B
=?	La expresión $A =? B$ verifica si el valor de A es igual al valor de B
<=	La expresión $A \leq B$ verifica si el valor de A es menor o igual que el valor de B
<	La expresión $A < B$ verifica si el valor de A es menor que el valor de B
<>	La expresión $A \neq B$ verifica si el valor de A es diferente del valor de B

## 1.8 Prioridad de los operadores

El orden de prioridad de los operadores influencia la manera en que las expresiones son evaluadas.

El orden de prioridad es decreciente de arriba a abajo según se muestra en la tabla siguiente. Cuando dos operadores se hallan en la misma línea, se da prioridad de izquierda a derecha.

Operador	Prioridad
()	Paréntesis
+, -, NOT, ^, !	Operadores unitarios
*, /	Operadores multiplicativos.
+, -	Operadores de adición
< <= > >=	Operadores relacionales
=? <>	Operadores relacionales
XOR	Operador de adición
AND	Operador de adición
OR	Operador de adición

Utilizando los paréntesis se puede cambiar este orden. Así, las expresiones encerradas dentro del mayor número de paréntesis reciben la más alta prioridad de evaluación.

## 1.9 Verificación de la sintaxis

Como ayuda para la detección de errores de sintaxis durante la edición de acciones, de transiciones o de acciones independientes, Automation Studio atribuye un color diferente a cada elemento de sintaxis una vez que los reconoce como tales. De esta manera, si usted ingresa un operador y el programa le atribuye el color de una etiqueta, es claro que la sintaxis del operador es incorrecta.

Para asociar colores a los diferentes elementos de la sintaxis:

1. Seleccione «Configuración» en el menú desplegable «Archivo».
2. Pulse en el tabulador «GRAFCET».

La ventana de diálogo siguiente aparece en la pantalla.

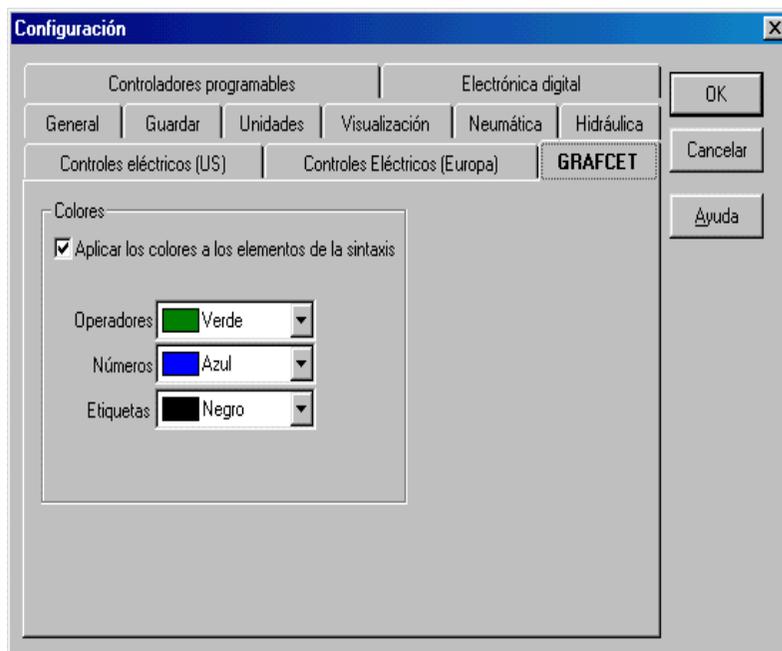


Figura 1-2: Ventana de diálogo de Configuración, tabulador «GRAFCET»

3. Marque la opción «Aplicar los colores a los elementos de la sintaxis».
4. Modifique los colores asociados a los elementos.
5. Pulse «OK» para conservar la modificación.

Si la opción «Aplicar los colores a los elementos de la sintaxis» no es marcada, ninguna modificación de la sintaxis será hecha durante la edición. Sin embargo, si la sintaxis es incorrecta, todos los componentes serán presentados en el esquema con un color particular. Este color es el Color de Incidente del componente que puede hallarse en el tabulador general de la ventana de diálogo de «Configuración» accesible en el menú «Archivo».

Para cambiar el color asociado a los componentes erróneos:

1. Seleccione «Configuración» del menú «Archivo».
2. Pulse en el tabulador «General».

Esta ventana de diálogo aparece en la pantalla:

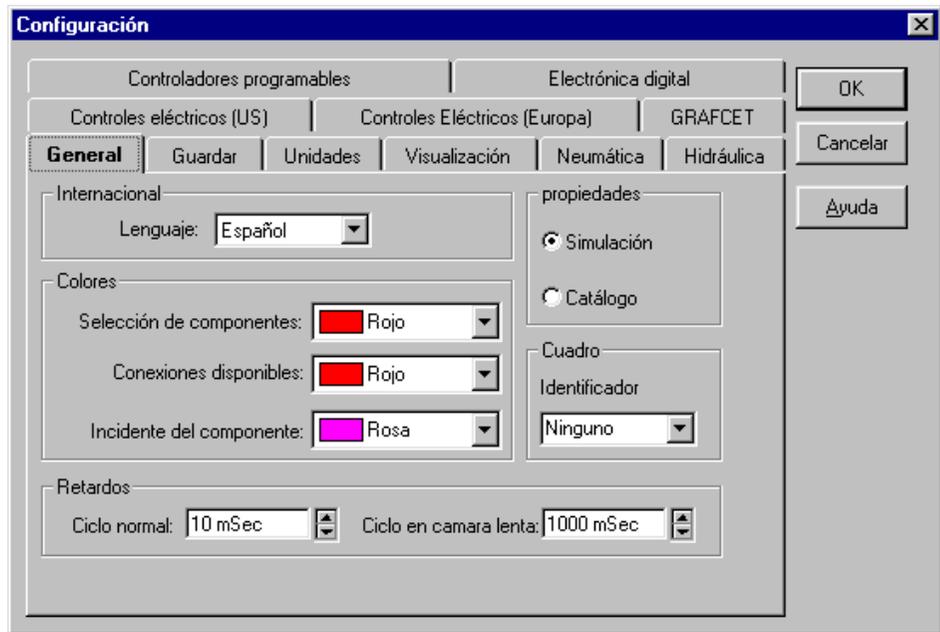


Figura 1-3: Ventana de diálogo de Configuración, tabulador «General»

3. Seleccione el color que usted desea asociar a los componentes erróneos (zona: «Incidente del componente»).
4. Pulse «OK» para conservar los cambios.

## 1.10 Ejemplos de transiciones

Una receptividad (condición de transición) se asocia a cada transición - la ausencia de receptividad significa que ésta tiene siempre el valor FALSO -. Se trata de una condición que determina la posibilidad o no de evolución del sistema por el camino de esta transición. Es una expresión booleana escrita con ayuda de las variables de entrada, de las variables de etapa ( $X_n$ ), de los operadores lógicos, del operador de temporización y de los operadores de flanco.

Las receptividades pueden tener un comentario asociado.

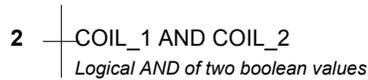


Figura 1-4: Receptividad de una transición

## 1.11 Operadores de asignación

Los operadores de asignación definen el tipo de acción y sus efectos. Se aplican tanto a variables enteras como a variables booleanas. Las secciones siguientes tratan los diferentes tipos de operadores de asignación.

### 1.11.1 Asignación booleana continua

La acción booleana continua (o acción incondicional en una etapa) es una asignación cuya duración es igual al tiempo de activación de la etapa con la que está asociada. No hay operadores de asignación. La variable vale VERDADERO cuando la etapa está activa y vale FALSO cuando la etapa es desactivada.

La sintaxis de una acción booleana continua es la misma que la del nombre o la de la etiqueta de la variable. En tal caso no se utilizan operadores de asignación.

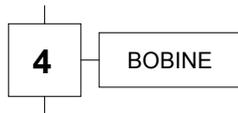


Figura 1-5: Etapa cuya acción contiene una acción booleana continua

En este ejemplo:

- BOBINA es una variable booleana;
- Cuando la etapa 4 es activada, la variable BOBINA toma el valor VERDADERO por la duración del lapso de activación de la etapa;
- Una vez desactivada la etapa 4, la variable BOBINA toma el valor FALSO.

### 1.11.2 Asignación numérica

El operador de asignación numérica asigna un valor (variable, constante o resultante de un cálculo) a una variable numérica. El operador de asignación numérica tiene el signo: «:=».

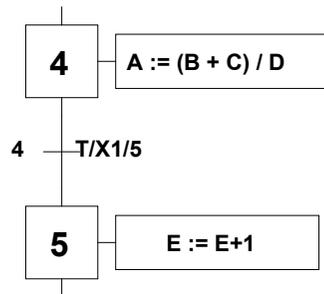


Figura 1-6: Asignación numérica

En este ejemplo:

- Las variables A, B, C, D, E y F son variables numéricas (enteras);
- Cuando es activada la etapa 4, A toma el valor de la expresión  $(B + C) / D$ ;
- A conserva el último valor de  $(B + C) / D$  luego de la desactivación de la etapa 4;
- Cuando se activa la etapa 5, la variable E se incrementa de 1.

### 1.11.3 Asignación booleana condicional

La asignación booleana condicional permite asignar el estado de una variable o de una expresión a una variable booleana, si es satisfecha una condición determinada. En caso contrario, se asigna el valor FALSO a la variable booleana. El signo del operador de asignación booleana condicional es: «IF».



Figura 1-7: Asignación booleana condicional

En este ejemplo:

- Cuando la etapa 4 está activa, el estado de la variable BOBINA es el estado de la expresión ^X1;
- Cuando la etapa 4 es desactivada, BOBINA toma el valor FALSO.

### 1.11.4 Asignación numérica condicional

Este tipo de afectación es evaluado cuando la condición especificada es verificada. Los operadores de asignación numérica condicional son «IF» y «:=».

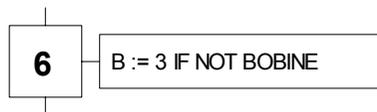


Figura 1-8: Asignación numérica condicional

En este ejemplo:

- Cuando la etapa 6 está activa, la variable B toma el valor 3 si la expresión NOT BOBINA es evaluada con valor VERDADERO;
- Desactivada la etapa 6, B conserva su último valor.

### 1.11.5 Asignación booleana memorizada

La asignación booleana memorizada permite asignar a una variable un estado VERDADERO o FALSO, estado que ella conservará (memorización). Para asignar el estado VERDADERO o FALSO a una variable, los operadores de asignación son respectivamente: «:= 1» o «:= 0».

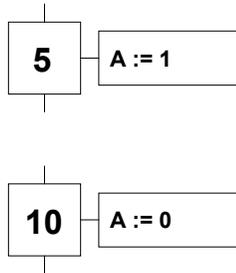


Figure 1-9: Asignaciones memorizadas

En este ejemplo:

- La variable A toma el estado VERDADERO con la activación de la etapa 5 y conserva este valor luego de la desactivación;
- La variable A toma el estado FALSO con la activación de la etapa 10 y conserva este valor luego de la desactivación.

## 1.12 Acción independiente

Las acciones independientes actúan exactamente como acciones ordinarias salvo que no están asociadas con ninguna etapa en particular. Son ejecutadas justo antes de la etapa inicial de un GRAFCET y esto se repite para cada ciclo de simulación.

Se puede insertar tantas acciones independientes como se quiera y se las puede ubicar libremente dentro del esquema. Cuando un esquema tiene varias acciones independientes, la acción independiente ubicada en la posición más alta dentro del esquema será la primera en ser ejecutada. Por ejemplo, los números de las acciones independientes de la figura siguiente ilustran el orden de ejecución.

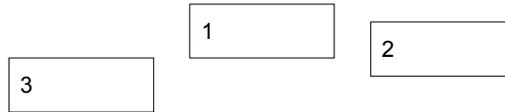


Figura 1-10: Acciones independientes: orden de ejecución

Si las acciones independientes son emplazadas en la misma línea horizontal, entonces el orden de ejecución corresponde con el orden de inserción en el esquema.

## 1.13 Temporización

Las temporizaciones son operadores booleanos que toman en cuenta el tiempo (lapso, espera, retraso,...). Se escriben generalmente con esta forma «T/En/duración base\_de\_tiempo/» donde:

- *T/* es el operador de temporización;
- *En* designa una variable de entrada, el nombre de la etapa o de la variable que gatilla el tiempo de temporización;
- *duración* es una variable numérica que fija el lapso de temporización;
- *unidad\_de\_tiempo* fija la unidad temporal, se la indica con una minúscula y es, por defecto, el segundo. Las distintas posibilidades de unidad de tiempo son:

<b>Tiempo</b>	<b>Simbolo</b>
1 segundo	s/
0,1 segundo	d/
0,01 segundo	c/
10 segundos	z/

Las temporizaciones utilizadas en GRAFCET se refieren a las etapas y a las variables.

### 1.13.1 Temporización de las etapas

Siguiendo la norma internacional IEC 1131-03, la temporización «T/Xn/t1 s/» toma el valor lógico «1» desde el momento en que han transcurrido «t1» segundos desde el último flanco de subida de la etapa «Xn». *Ella no cambia de valor con la desactivación* de la etapa «Xn» pero retoma el valor lógico «0» a cada reactivación de la etapa «Xn».

El ejemplo de la figura siguiente, muestra la evolución de una temporización «T/X1/3 s/». Si durante la simulación la etapa X1 es activada, la temporización es gatillada, la fórmula booleana «T/X1/3 s/» toma el valor FALSO y la fórmula booleana «NOT(T/X1/3 s/») toma el valor VERDADERO.

Tres segundos después de la activación de la etapa X1 la fórmula booleana «T/X1/3 s/» pasa del estado FALSO a VERDADERO y «NOT(T/X1/3 s/») de VERDADERO a FALSO. Valores que conservarán hasta la próxima activación de la etapa X1.

Si en el transcurso de la temporización, la etapa X1 es desactivada, la temporización es interrumpida y las fórmulas booleanas «T/X1/3 s/» y «NOT (T/X1/3 s/»)» conservan sus valores hasta la próxima activación de la etapa X1.

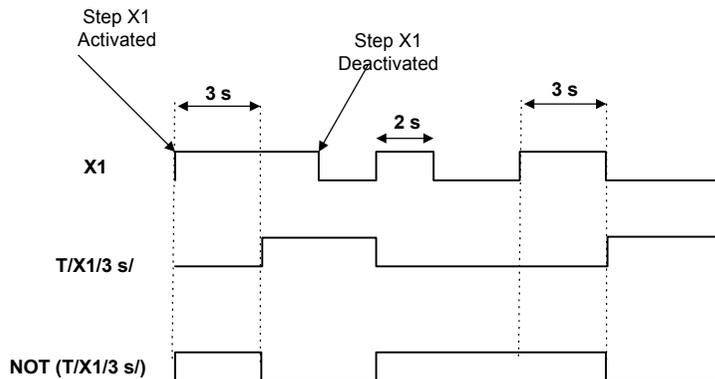


Figura 1-11: Cronograma de una temporización de etapa :  $T/X1/3 s/$

Se usa la temporización para definir acciones retardadas o limitadas temporalmente.

### Acciones retardadas



Figura 1-12: Asignaciones retardadas

La variable STOP toma el valor VERDADERO tres segundos después de la activación de la etapa 1. Una vez desactivada esta etapa, STOP toma el valor FALSO.

## Acciones limitadas en el tiempo

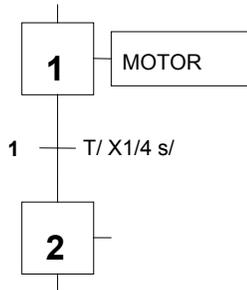


Figure 1-13: Asignación limitada en el tiempo

La variable MOTOR toma el valor VERDADERO con la activación de la etapa 1. Cuatro segundos más tarde, la transición 1 es traspasada y la variable MOTOR toma el valor FALSO.

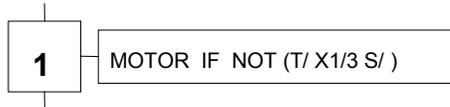


Figure 1-14: Asignación limitada en el tiempo

La variable MOTOR toma el valor VERDADERO con la activación de la etapa 1. Tres segundos después, la condición «NOT (T/ X1 /3 S/ )» toma el valor FALSO y la variable MOTOR toma el valor FALSO.

### 1.13.2 Temporización de las variables

La temporización «T/Var\_a/t1 s/» toma el valor lógico «1» cuando han transcurrido «t1» segundos desde el último flanco de subida de la variable «Var\_a ». La variable retoma el valor lógico «0» a cada flanco de bajada de la variable «Var\_a ».

El ejemplo de la figura siguiente muestra la evolución de las temporizaciones siguientes: «T/Var\_a/3 s/», «T/NOT Var\_a /3 s/» y «NOT(T/Var\_a /3 s/»).

Si en el transcurso de la simulación, la variable  $Var\_a$  es activada, la temporización es gatillada, la fórmula booleana «T/Var\_a/3 s/» toma el valor FALSO, la fórmula booleana «T/NOT Var\_a/3 s/» toma el valor FALSO y la fórmula booleana «NOT(T/Var\_a/3 s/» toma el valor VERDADERO.

Tres segundos luego de la activación de la variable  $Var\_a$ , la fórmula booleana «T/Var\_a/3 s/» pasa del estado FALSO a VERDADERO y «NOT(T/Var\_a /3 s/») de VERDADERO a FALSO. Conservarán sus valores hasta la próxima desactivación de  $Var\_a$ .

Si en el transcurso de la temporización, la variable  $Var\_a$  es desactivada, la temporización es interrumpida y las fórmulas booleanas «T/X1/3 s/» y «NOT (T/X1/3 s/») toman respectivamente los valores FALSO y VERDADERO.

Si la variable  $Var\_a$  queda desactivada durante 3 segundos o más, la fórmula booleana «T/NOT Var\_a/3 s/» pasa del estado FALSO al estado VERDADERO. Conservará este valor hasta la próxima activación de  $Var\_a$ .

Las temporizaciones pueden ser utilizadas para definir acciones retardadas o limitadas en el tiempo.

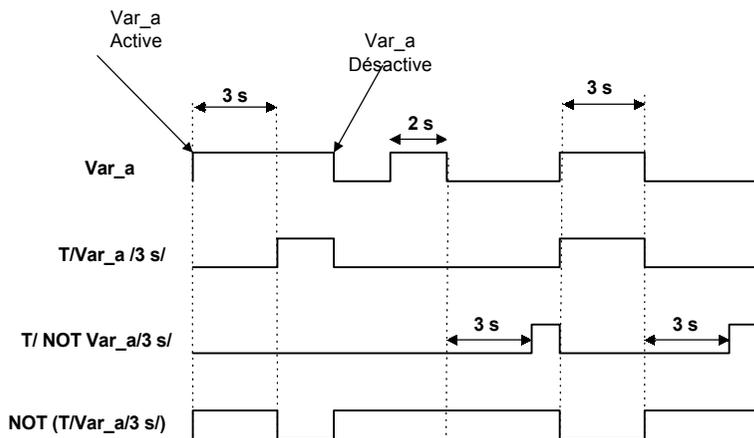


Figure 1-15: Cronograma de una temporización sobre una Variable

## Acciones retardadas

Son acciones efectuadas al cabo de un lapso de tiempo especificado

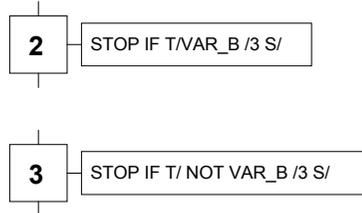


Figura 1-16: Asignaciones retardadas

- En la etapa 2: la variable STOP toma el valor VERDADERO si la variable VAR\_B queda activada durante 3 segundos. Una vez que la etapa 3 es desactivada, la variable STOP toma el valor FALSO;
- En la etapa 3: la variable STOP toma el valor VERDADERO si la variable VAR\_B queda desactivada durante 3 segundos. Una vez que la etapa 3 es desactivada, la variable STOP toma el valor FALSO;

## Acciones limitadas en el tiempo

Son acciones realizadas durante un cierto lapso de tiempo especificado.

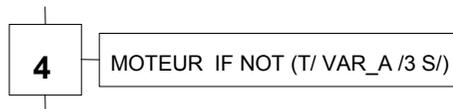


Figura 1-17: Asignación limitada en el tiempo

MOTOR es VERDADERO hasta que la variable VAR\_A sea activada durante 3 segundos. MOTOR toma entonces el valor FALSO. Cuando VAR\_A es desactivada, MOTOR retoma el valor VERDADERO.

## 1.14 Forzados

Los forzados permiten la modificación de la situación de un GRAFCET. Se los emplea para imponer una situación determinada. El forzado se mantiene mientras su condición de activación permanece verdadera.

En simulación es posible forzar una etapa o una transición sin respetar la sintaxis del GRAFCET. Esta funcionalidad aporta una ayuda a la concepción del sistema automatizado PO-PC.

### 1.14.1 Utilización de los forzados

Los forzados pueden ser empleados tanto en la edición como en la simulación.

#### Utilización de los forzados durante la edición

La forma general del forzado es:  $F/(X_i, X_j, \dots)$

- $F/$  es el operador de forzado;
- $X_i$  es el nombre de una etapa de la situación de forzado.

Se puede especificar una o varias etapas del GRAFCET. Si se especifica varias etapas, éstas deben ser separadas por comas.

Ejemplos de uso de forzados:

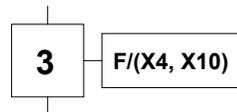


Figura 1-18: Forzado

En este ejemplo, el GRAFCET es forzado en la situación donde solamente la etapas 4 y 10 están activadas.

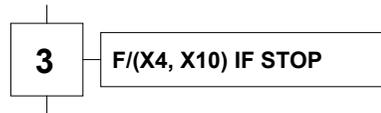


Figura 1-19: Forzado condicional

En este otro ejemplo, el GRAFCET es forzado en la situación donde solamente las etapas 4 y 10 están activadas mientras la variable STOP tenga el valor VERDADERO.

## Utilización de los forzados durante la Simulación

Método para acceder al forzado de una etapa o de una transición durante la simulación:

1. Arranque la simulación del esquema o del proyecto.
2. Ubique el cursor sobre una etapa o sobre una transición.
3. Pulse en el botón derecho del ratón.  
El menú contextual se abre.
4. Elija la función «Forzar la etapa» o la función «Forzar la transición» según corresponda.

La etapa o la transición es entonces forzada independientemente de la evolución normal del GRAFCET.

## 1.15 Ejemplos de etapas - acciones

Una acción de etapa es un proceso que se ejecuta cuando la etapa está activada. Las acciones se componen de expresiones, operadores de asignación, operadores de control.

Cuando es posible asociar varias acciones a una misma etapa, tales acciones emiten órdenes a la Parte operativa cuando la etapa está activada.

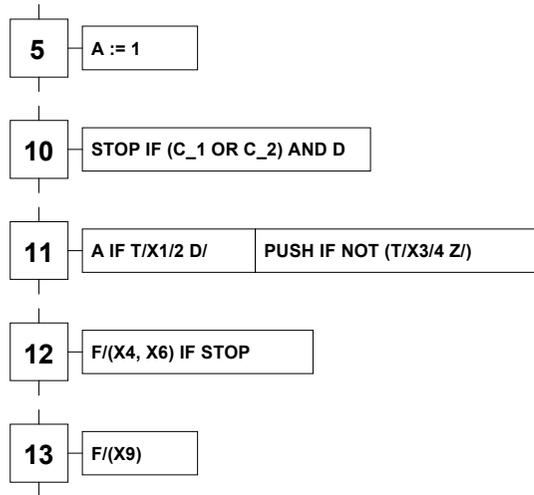


Figura 1-20: Ejemplos de etapas-acciones



## 2 Simulación

Este capítulo trata de particularidades relacionadas con la simulación del GRAFCET. Se propone familiarizarlo con el modo de simulación de este taller.

Puede referirse al capítulo 6 de la Guía del usuario del Sistema de base para mayor información acerca del proceso de simulación en general.

### 2.1 Disposición de las ventanas

Antes de simular su primer proyecto SFC01.PRO usted debe organizar la disposición de las ventanas de modo que sea posible ver los dos esquemas simultáneamente.

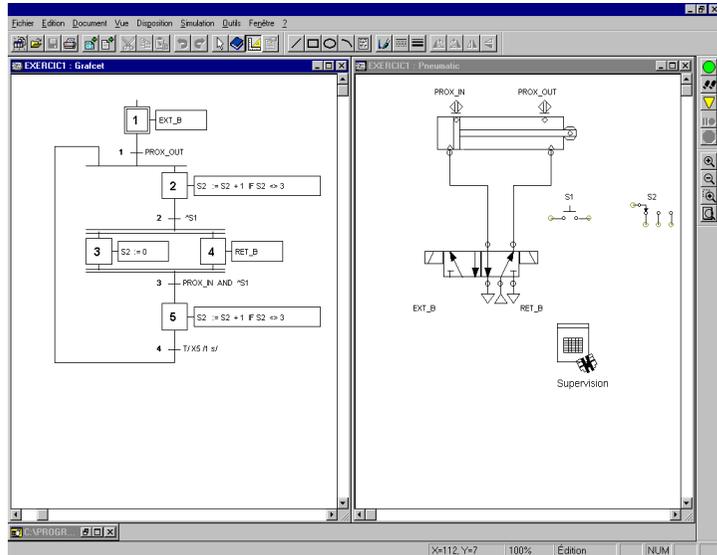


Figura 2-1: Disposición de diferentes ventanas

## 2.2 Simulación del proyecto

Para simular el proyecto SFC01.PRO o SFC03.PRO:

1. Elija «Comenzar el proyecto» en el menú desplegable «Simulación».

o



Pulse en «Simulación del proyecto» en la barra de herramientas.

Comienza el modo simulación.

2. Para observar el desarrollo de cada ciclo de cálculo seleccione «Paso a paso» en el menú «Simulación».

o



Pulse en «Paso a paso» en la barra de herramientas.

La simulación avanza un paso un ciclo a cada pulso del botón de izquierda del ratón. En cada ciclo se efectúa un cálculo para determinar el nuevo estado de los componentes.

Las etapas y las acciones activas aparecen enmarcadas en rojo.

### 2.2.1 Intervenciones del usuario

Durante la simulación es posible cambiar el estado de un componente forzando su activación por medio del ratón.

Por ejemplo, para activar el pulsador S1 o el interruptor multiposicional durante la simulación:

1. Desplace el cursor sobre el componente.

El cursor adquiere la apariencia de una mano. La mano simboliza la posibilidad de intervenir durante la simulación.





Figura 2-2: Ejemplos de forzado de un pulsador NA

2. Pulse en el pulsador del componente. El modo Simulación permite al componente reaccionar a la acción del cursor.
3. Suelte el botón del ratón, el componente vuelve a su estado inicial.

## 2.2.2 Simulación de SFC01.PRO o SFC03.PRO

1. Pulse en el botón de simulación.  
El GRAFCET arranca y la etapa 1 es activada.
2. Pulse en el pulsador BP\_1.  
Se activa entonces la etapa 2.
3. Pulse en COM\_A.  
El cilindro A levanta la carga.
4. Pulse en COM\_B.  
El cilindro B empuja la carga hacia la derecha.
5. Durante las etapas 4 y 5, los dos cilindros efectúan sus carreras de entrada.
6. Pulse en M\_A o sobre M\_B para transportar la carga al contenedor de la izquierda o al de la derecha.  
La etapa 1 es reactivada. Pulsando sobre BP\_!, otra carga puede ser desplazada.

## 2.3 Causas de los problemas

El editor de GRAFCET no analiza la construcción de un GRAFCET durante su edición. Es decir que es posible crear lo que se quiera en la edición sin que por eso el resultado sea un GRAFCET.

El problema se plantea cuando se trata de pilotear o de simular un GRAFCET. Es absolutamente necesario entonces que el GRAFCET creado sea conforme a la norma y que sus acciones y sus receptividades sean correctas.

En la sección siguiente encontrará ciertas particularidades de los GRAFCET.

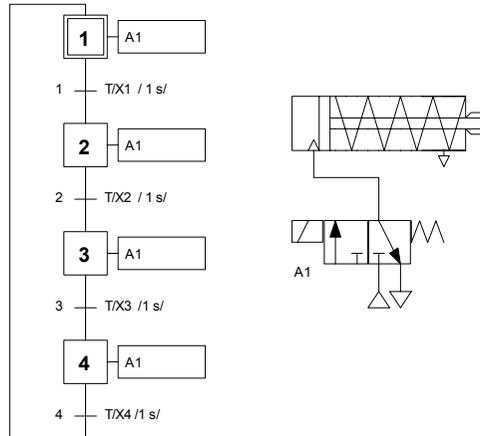
### 2.3.1 Particularidades de los GRAFCET

Aquí son presentadas las características distintivas de los GRAFCET:

1. La etapa inicial es la etapa activa al comienzo del funcionamiento. Un GRAFCET puede tener varias etapas iniciales.
2. Se traspasa una etapa si todas las etapas que la preceden inmediatamente están activas y si su receptividad es verdadera. Varias transiciones pueden ser traspasadas simultáneamente.
3. El atravesar una transición provoca la desactivación de todas las etapas inmediatamente anteriores ligadas a esta transición y la activación de todas las etapas inmediatamente sucesivas también ligadas a esta transición.
4. Si en un GRAFCET una etapa debe encontrarse simultáneamente activada y desactivada, permanece activada.

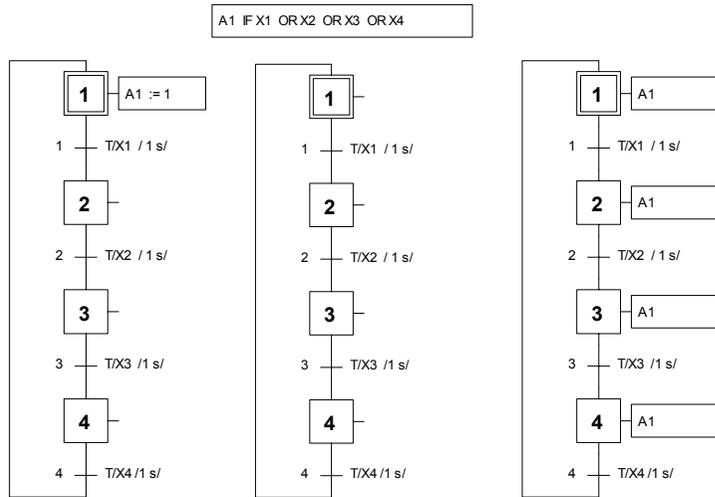
## 2.3.2 Particularidad del taller

La versión actual de este taller no actua exactamente como lo estipulan las reglas del GRAFCET. Por ejemplo, en el siguiente GRAFCET:



Según las reglas del GRAFCET, si A1 es activada en la etapa 1 y en las etapas 2, 3 y 4, deberá permanecer activa incluso en el transcurso de las transiciones entre esas etapas. No obstante, la versión actual de este taller desactiva A1 momentaneamente durante las transiciones.

Para evitar este problema, le proponemos una entre las tres soluciones siguientes:



La primera elimina las acciones para las etapas 2, 3 y 4 y asigna 1 a A1 con la activación de la primera etapa. La segunda solución elimina las etapas y añade una acción independiente que activa A1 cuando al menos una etapa es activada.



La última solución no es recomendable. Las acciones han sido reemplazadas por acciones autónomas. Este método puede funcionar con un GRAFCET sencillo pero no funcionará con un GRAFCET más complejo.

Opte por la solución más conveniente.

# 3 Ejercicio

Este capítulo presenta un ejercicio que le permitirá realizar diferentes circuitos con GRAFCET.

## 3.1 Ejercicio – Circuito de control de un taladro

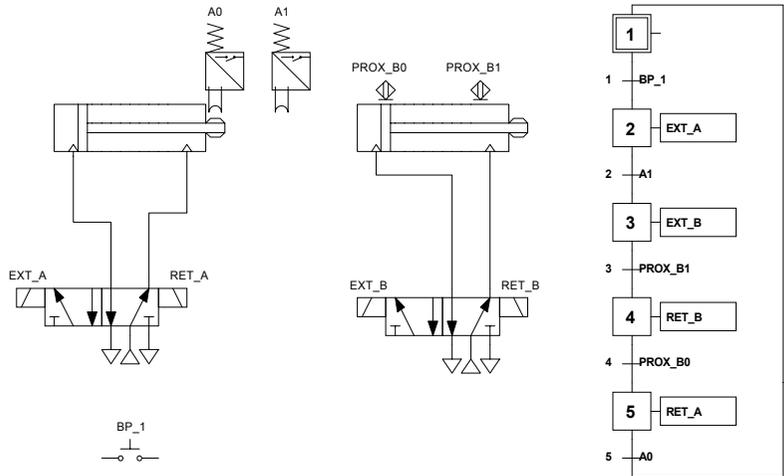


Figura 3-1: Circuito-GRAFSET control de una taladro

## Descripción de la parte operativa

El taladro está compuesto en primer lugar de un cilindro horizontal A. El rol de este cilindro es el de retener la pieza sobre la que se efectúa la perforación. Un segundo cilindro B se encarga de la perforación. Con la orden de puesta en marcha, se inicia un movimiento de salida del vástago del cilindro A, seguido de un movimiento de descenso y de ascenso del vástago del cilindro B. El ciclo termina con el regreso a su posición inicial del vástago del cilindro A. El control de cada cilindro es asegurado por una válvula 5/2 (12) controlada en ambos costados por solenoides. Los movimientos del vástago del cilindro A son detectados por dos detectores de posición mecánica: A0 y A1. Los del vástago del cilindro B son detectados por dos detectores de proximidad: PROX\_B0 y PROX\_B1.

El proyecto SFC02.PRO que contiene este ejemplo está situado en el archivo EXERC de este programa.

## GRAFCET

El ciclo de funcionamiento del taladro es representado por el GRAFCET.

Ce dernier est composé de: Éste se compone de:

1. Cinco etapas (X1, X2, X3, X4 y X5). A cada etapa se le asocia una acción, salvo a la etapa inicial.
2. Cinco transiciones (Y1, Y2, Y3, Y4 et Y5). A cada transición se le asocia una receptividad.

## Descripción del funcionamiento

Cuando el circuito está completado, se puede pasar al modo Simulación para verificar su funcionamiento.

1. Pulse sobre el botón de simulación.  
Esto permite posicionar el GRAFCET en la etapa inicial, etapa X1. El circuito es puesto bajo tensión por primera vez.
2. Pulse en el pulsador BP\_1 para arrancar.

Cuando el pulsador BP\_1 es accionado, la transición Y1 toma el valor VERDADERO y el control pasa a la etapa X2: comienza la salida del vástago del cilindro A.

3. A1 es activada en fin de carrera y permite el paso a la etapa X3.

Dado que la acción asociada a esta etapa es la salida del vástago del cilindro B, el fin de carrera es detectado por el detector de proximidad PROX\_B1 que es activado.

4. En ese momento, la etapa X4 se activa y se procede al reingreso del vástago del cilindro B.
5. La activación de la etapa X5 es posible cuando el detector de proximidad PROX\_B0 se activa indicando que el vástago del cilindro B ha completado su reingreso.

Cuando la etapa X5 es activada, el vástago del cilindro A reingresa y el automatismo vuelve a la etapa inicial X1.

6. La activación del pulsador BP\_1 es necesaria para el reinicio del ciclo. Si el pulsador permanece oprimido, el ciclo se renueva continuamente.



---

# A. Glosario

## **Acción**

Una acción de etapa es una declaración que se ejecuta cuando la etapa está activada. Es una orden emitida en la forma de una salida desde la parte control hacia la parte operativa. Las acciones se componen de expresiones, de operadores de asignación, de operadores de control.

Todas las acciones asociadas a una misma etapa son ejecutadas cuando una etapa está activada.

## **Alfanumérica**

Califica una expresión compuesta de caracteres pertenecientes al juego de caracteres reconocidos por la máquina, de los cuales los principales son las 10 cifras decimales y las 26 letras del alfabeto.

## **Barra de estado**

Barra horizontal situada debajo de todas las ventanas, que contiene varias informaciones (comentarios, porcentaje de zoom o coordenadas del cursor, etc).

## **Barra de herramientas**

Barra situada debajo de la barra de menús, que agrupa botones que permiten ejecutar los comandos más utilizados.

## **Barra de menús**

Barra horizontal situada bajo la barra de título del programa, que presenta los menús de los comandos disponibles para la ventana activa.

### **Barra de título**

Barra horizontal situada en el tope de una ventana y que contiene el título de ésta.

### **Ciclo de simulación**

Corresponde a un ciclo de cálculo determinante del estado de cada uno de los componentes.

### **Componente**

Elemento básico para la concepción de esquemas. Cada componente está asociado a un comportamiento o a una función representados en la simulación. Los componentes integran las librerías de los talleres.

### **Conector**

Componente que representa una conexión en la forma de un punto negro, que puede ser insertado en los enlaces para identificar puntos de conexión. Contrariamente a las conexiones, los conectores pueden ser impresos.

### **Conexión**

Una conexión simboliza con círculos el punto que conecta enlaces o componentes. La conexión es del mismo color que los elementos si los puntos de conexión están en contacto y de colores distintos si no lo están.

Una conexión de un tipo de tecnología no puede ser conectada con una conexión de una tecnología diferente. Por ejemplo, una línea de presión neumática no puede conectarse con un componente GRAFCET y viceversa. Para que un piloto GRAFCET actúe sobre una válvula neumática, deberá ser alimentado por una línea GRAFCET proveniente del taller GRAFCET.

### **Convergencia en Y**

Una convergencia en Y es una relación entre varias etapas precedentes y una transición. Se obtiene de la librería GRAFCET.

### **Convergencia en O**

Una convergencia en O es una relación entre varias transiciones precedentes y una etapa. Se obtiene de la librería GRAFCET.

### **Cuadrícula**

Líneas de puntos horizontales y verticales en el espacio de trabajo del Editor de esquemas sobre los cuales son alineados los elementos del esquema.

### **Divergencia en O**

Una divergencia en O es una relación entre una etapa precedente y varias transiciones. Se obtiene de la biblioteca GRAFCET.

### **Divergencia en Y**

Una divergencia en Y es una relación entre una transición precedente y varias etapas. Se obtiene de la biblioteca GRAFCET.

### **Editor de proyectos**

Función que permite la creación, la modificación y la gestión de los documentos «proyectos». Contiene la lista de documentos que enumera todos los documentos del proyecto.

### **Enlace**

Elemento de la librería que sirve para conectar los componentes de un esquema. En simulación el enlace transmite una señal de un componente a otro.

### **Espacio de trabajo**

Parte de la ventana donde son presentadas las informaciones sobre las que se trabaja.

### **Esquema**

Se trata del esquema que permite representar gráficamente un circuito con la ayuda de elementos y de componentes seleccionados en la librería de los talleres.

### **Etapas**

Una etapa normal es representada con un cuadrado identificado por un número. Se obtiene pulsando en el símbolo «Etapa» de la librería GRAFCET.

### **Etapas activas**

Una etapa activa es identificada por un símbolo rojo ubicado al centro del cuadrado de la etapa.

### **Etapas iniciales**

Una etapa inicial es representada con un cuadrado doble identificado por un número y es activada siempre con el arranque del GRAFCET. Se obtiene pulsando «Etapa inicial» en la ventana de diálogo de las propiedades de una etapa.

### **Etiquetas del proyecto**

Breve descripción del proyecto que figura en su resumen.

### **Forzados**

Los forzados permiten la modificación de la situación en un GRAFCET. Son empleados para imponer una cierta situación a un

GRAFCET. El forzado es mantenido mientras su condición de activación sea verdadera.

La forma general de un forzado es:  $F/(X_i, X_j, \dots)$ .

### **Flanco**

Un flanco es un operador particular que indica el paso de una variable o de una expresión booleana del valor 0 al valor 1 y del valor 1 al valor 0: respectivamente, un flanco de subida y un flanco de bajada.

### **Forma del enlace**

Trazo del enlace entre dos puntos de conexión.

### **GRAFCET**

GRÁfico Funcional de Control por Etapas y Transiciones. El GRAFCET es un modelo de representación gráfica que utiliza sus elementos básicos para representar un sistema automatizado.

### **Herramientas de visualización**

Designa los accesorios de concepción del Editor de esquemas: cuadrícula, reglas, conexiones, números de conexión. Su visualización puede ser seleccionado en el menú «Ver».

### **Hoja de trabajo**

Superficie total disponible en Automation Studio para la realización de un esquema.

### **Interfaz usuario**

Entorno constituido por las ventanas, las ventanas de diálogo, los menús, los comandos, el ratón, los botones, etc. que permite al usuario comunicar con el ordenador.

### **Librería**

Ventana que agrupa los elementos básicos para diseñar un circuito o un modelo de simulación. Dichos elementos pueden ser de tres tipos: componentes, enlaces y objetos gráficos. Tales elementos vienen con los talleres.

### **Lista de documentos**

Contenido de una ventana del *Editor de proyectos*. La lista de documentos es una lista que enumera los documentos del proyecto.

### **Modo Edición**

Modo de funcionamiento en el que los esquemas del proyecto son creados y modificados.

### **Modo Simulación**

Modo de funcionamiento en el que un proyecto o un esquema es simulado. Contrariamente al modo edición, no se puede introducir ninguna modificación al proyecto.

### **Número de conexión**

Número asociado a cada punto de conexión de los componentes.

### **Objeto gráfico**

Elemento de la librería que no puede ser simulado. Un objeto gráfico puede ser insertado en un esquema como elemento sin funcionalidad. Hay cinco tipos de objetos gráficos: rectángulos, elipses, líneas, arcos y textos.

### **Propiedades**

Característica o parámetro de un componente. Es posible visualizar o modificar las propiedades abriendo la ventana de diálogo «Propiedades» del componente.

### **Proyecto**

Conjunto coherente formado por los documentos. El proyecto es administrado por el Editor de proyectos.

### **Receptividad**

Una receptividad o condición de transición es una condición lógica que determina la posibilidad o no de evolución del sistema a través de esta transición. Es una expresión booleana escrita con la ayuda de las variables de entrada, de las variables de etapa  $X_n$ , de los operadores lógicos, del operador de temporización y de los operadores de flanco. Puede ser verdadera o falsa.

### **Reglas**

Las reglas, presentadas en los bordes del esquema, indican las unidades de medida y sirven de referencia para las dimensiones del esquema y la posición respectiva de los elementos.

### **Sistema de base**

Conjunto de funciones generales del programa Automation Studio, que agrupa los comandos de edición y de simulación.

## **Taller**

Módulo complementario del Sistema de base. Cada taller contiene los elementos y las funciones relativas a la tecnología del taller y al tipo de proyectos que permite crear.

## **Taller activo**

Taller instalado cuyos componentes aparecen en la librería del Editor de esquemas. Si el taller no está activado, usted puede hacerlo. Para el procedimiento de activación, véase la Guía del usuario del Sistema de base del programa Automation Studio.

## **Temporizaciones**

Las temporizaciones son expresiones booleanas. Permiten tomar en cuenta el tiempo (lapso, espera, retraso,...).

## **Transición**

La condición de una transición debe ser verdadera para que la etapa siguiente pueda ser ejecutada.

Las temporizaciones usadas en GRAFCET se refieren a las variables y a las etapas.

## **Utilitario**

Término general que designa los diferentes tipos de ventana en Automation Studio. El Sistema de base tiene dos utilitarios: el Editor de proyectos y el Editor de esquemas.

---

## B. INDEX

- Acción independiente, 20
- Asignación
  - booleana memorizada, 20
- Asignación booleana condicional, 19
- Asignación booleana continua, 17
- Asignación numérica, 18
- Asignación numérica condicional, 19
- Forzado
  - sintaxis, 45
- Operador
  - expresiones booleanas, 9
  - Flancos, 10
  - Prioridad, 13
  - relacionales, 12
- Simulación
  - Paso a paso, 32
  - Simula el proyecto, 32
- Sintaxis
  - Acciones, 5
  - Etapas, 4
  - Etiquetas, 4
  - flancos, 10
  - forzado, 45
  - forzado, 27
  - introducción, 3
  - Palabras reservadas, 7
  - Transición, 6
  - variable, 8
  - vérificación, 14
- Temporización, 21
- Temporización de las etapas, 22
- Temporización de las variables, 24
- Variable
  - tipo, 8