



## Módulo 3.C

## Lenguaje IEC : IL

## ● Panorama

- Lenguaje máquina booleano para procesamiento lógico o numérico
- Confome a la norma IEC 61131-3
- Muy sencillo y fácil de utilizar
- Código rápido y optimizado para secciones críticas
- Ideal para resolver ciertos problemas complejos
- Dificultad para escribirlo, leerlo y entender el programa
- Dificultad para seguir el flujo del programa

## ● Instrucciones

Un programa en listado de instrucciones se compone de una serie de instrucciones.

Starting_motor :	LD	Starting_motor	(*To start the motor*)
	ANDN	Motor_fault	(*Motor in fault*)
	ST	Motor	(*Motor starting*)
Convert_speed :	LD	Motor_speed	(*Motor speed*)
	MUL	thread	(*Thread*)
	ST	Speed	(*Speed in m/s*)

Starting\_motor

LD

ANDN

Motor\_fault

(\*Thread\*)

Cada instrucción inicia con una nueva línea y consiste en :

- Etiqueta (opcional) : localiza una secuencia en un programa
- Operador : determina la operación a efectuar
- Modificador : influencia la ejecución del operador
- Operando : es un objeto sobre el cual el operando actúa
- Comentario (opcional) : proporciona alguna información referente a la línea

Módulo 3 : Unity Pro

Sección C : Lenguaje IEC : IL

Página 3/12

## ● Etiquetas

- Siempre deben ser el primer elemento de la línea
- Debe ser única en la sección
- puede ser mayúsculas o minúsculas
- puede ser de 32 caracteres de longitud (max.)
- Son separadas por dos puntos ":"
- Únicamente se permiten al inicio de la expresión "Expressions"



## ● Operadores

Las siguientes instrucciones puede ser ejecutadas :

- asignación (LD, ST, S, R)
- logicas (AND, OR, XOR, NOT)
- aritmeticas (ADD, SUB, MUL, DIV, MOD)
- comparación (GT, GE, EQ, NE, LE, LT)
- estructura (JMP, RET, ")")
- call (CAL, FUNCTIONSNAME, PROCEDURENAME)

## ● Operators

### Load and save

Operator	Modifier	Meaning / Operand
LD	N	Loads the operands value into the battery Value, variable, direct address of any data type
ST	N	Saves the battery value in the operand Variable, direct address of any data type

### Set and reset

Operator	Modifier	Meaning / Operand
S	-	Sets the operand to 1, when the battery content is 1 Variable, direct address of BOOL data type
R	-	Sets the operand to 0, when the battery content is 1 Variable, direct address of BOOL data type

### Logic

Operator	Modifier	Meaning / Operand
AND	N, N(, (	Logic AND Value, variable, direct address of BOOL, BYTE, WORD or DWORD data types

## ● Modificadores

- N : utilizado para invertir el valor de los operandos bit por bit
- C : utilizado para definir la asociación de instrucciones, el valor del resultado debe ser 1 o verdadera.
- CN : los modificadores C y N son combinados, la instrucción asociada en entonces realizada, el valor del resultado debe ser 0 boleano o FALSO.
- "(" : utilizado para efectuar la evaluación del operando hasta el siguiente operador ")"

## ● Operandos

Un operando puede ser :

- una dirección directa
- Un valor
- Un simbolo de una variable
- una variable multi elemento
- un elemento de una variable multielemento
- una salida EFB/DFB
- un llamado EFB/DFB call



Module 3 : Unity Pro

Section C : IEC Language : IL

Page 7/12

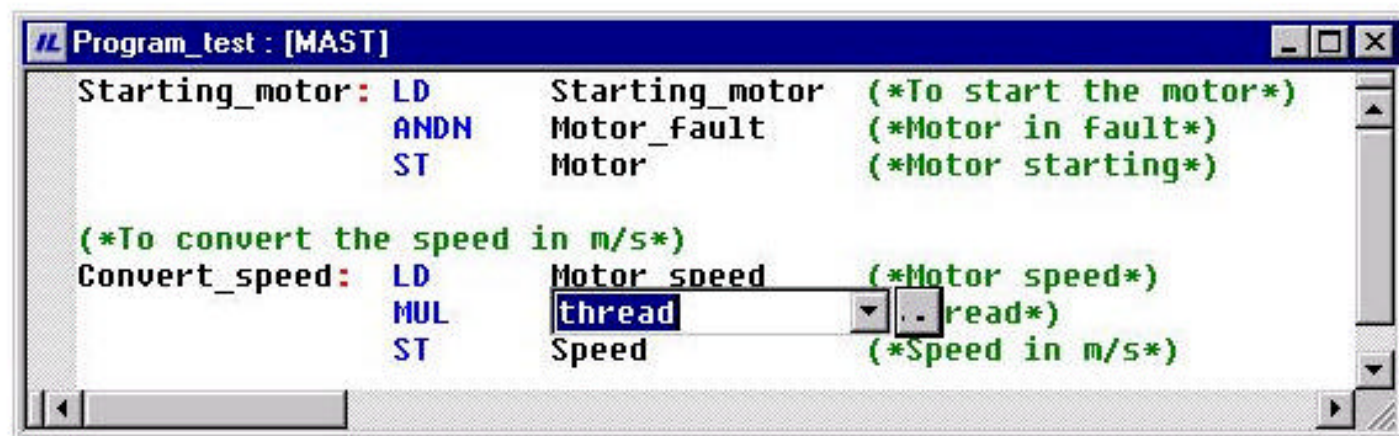
## ● Comentarios

En el editor IL, los comentarios siempre inician con la cadena "(" y terminan con "\*").

Entre esta secuencia de caracteres cualquier comentario se puede incluir

Los comentarios anidados o incluidos no están permitidos de acuerdo a la norma IEC 61131-3, pero es posible configurarlos en la opción de proyecto.

## ● Editor del Lenguaje



```

// Program_test : [MAST]
Starting_motor: LD      Starting_motor  (*To start the motor*)
                ANDN    Motor_fault    (*Motor in fault*)
                ST      Motor          (*Motor starting*)

(*To convert the speed in m/s*)
Convert_speed: LD      Motor_speed     (*Motor speed*)
                MUL     thread         (*read*)
                ST      Speed          (*Speed in m/s*)
    
```

- tiene el mismo aspecto que un editor tradicional (Wordpad) y funciones completas (pegar/copiar, tab,...)
- una instrucción máximo por línea
- utilización de colores para definir diferentes objetos
- Análisis de los datos al momento de teclearlos (revisión de la sintaxis y semántica durante la creación del programa)
- Ayuda al momento de definir los datos , funciones EF o FB via ventana de diálogo

# ● Editor de lenguaje

## Características

- Una instrucción utiliza una línea
- Una línea contiene como máximo 300 caracteres (instrucción, comentarios, etiquetas)
- La longitud de la sección no está limitada
- La posición de las etiquetas, símbolos y comentarios no son fijos
- La línea y columna están desplegadas en la barra de estatus
- Copiar , Pegar, Cortar...
- Secciones con errores pueden ser guardadas
- 
- Colores y Marcaje

Azul	Operadores como LD, AND, ...)
Rojo	Operadores como "(", ":", ":", "=", "=>", .....
Verde	Comentarios
Negro	Texto normal como variables, etiquetas, FB, ...
Rojo subrayado	Texto inválido (detección de un error sintaxis o semántica )
video inverso	Selección de un texto

## "Quick info"

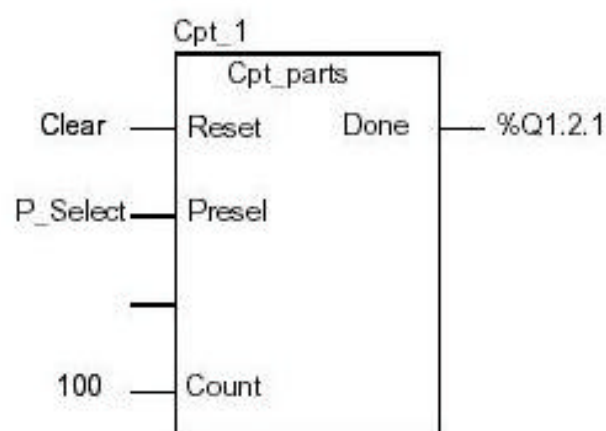
Si el cursor es puesto sobre un texto de falla, aparecerá automáticamente un texto con una breve descripción de los que causa el error.



Modulo 3 : Unity Pro

Sección C :Lenguaje IEC : IL

Página 9/12



```
CAL Cpt_1 (Reset:=Clear, Presel:=%
MD10, Count:=100)
```

```
...
LD Cpt_1.Done
ST %Q1.2.1
```

```
CAL Cpt_1 (Clear, %MD10,, 100)
```

```
...
LD Cpt_1.Done
ST %Q1.2.1
```

Comentario

## ● Reglas de Programación

- Variable
  - ▶ configuración deirecta con la función "Wizard"
- Funciones Elementales
  - ▶ no requieren declaración
  - ▶ EF utilizado directamente (sin el operador CAL)
  - ▶ definido directamente en el "FFB Wizard"
- Bloques de función Elemental o Bloques de función derivado
  - ▶ no requieren declaración (automáticamente definido en el "FFB Wizard")
  - ▶ EFB-definido utilizando el operador CAL
  - ▶ llamado Formal o informal
  - ▶ Definido directamente o por medio del "FFB Wizard"



# ● Editor de lenguaje

## Características

- Una instrucción utiliza una línea
- Una línea contiene como máximo 300 caracteres (instrucción, comentarios, etiquetas)
- La longitud de la sección no está limitada
- La posición de las etiquetas, símbolos y comentarios no son fijos
- La línea y columna están desplegadas en la barra de estatus
- Copiar, Pegar, Cortar...
- Secciones con errores pueden ser guardadas
- 
- Colores y Marcaje

Azul	Operadores como LD, AND, ...)
Rojo	Operadores como "(", ":", ":", "=", "=>", .....
Verde	Comentarios
Negro	Texto normal como variables, etiquetas, FB, ...
Rojo subrayado	Texto inválido (detección de un error sintaxis o semántica )
video inverso	Selección de un texto

## "Quick info"

Si el cursor es puesto sobre un texto de falla, aparecerá automáticamente un texto con una breve descripción de los que causa el error.

## ● Desarrollo del programa

- Crear o abrir una sección de listado de instrucciones IL
- Escribir el programa y editar las variable utilizadas
  - ▶ Definir el operador
  - ▶ DEfinir o seleccionar el operador (variable)
  - ▶ Definir o seleccionar los EFB/DFB
- Analizar el programa editar frecuentemente para verificar la sintaxis
  - ▶ Cualquier error será desplegado en la ventana de mensajes
  - ▶ Doble click en la línea para encontrar dierctamente el error en IL
- Guardar el proyecto
- Generar el proyecto y descargarlo
- Transfiera y conecte el proyecto al PLC
- Pruebe el programa en modo conectado

## ● Ejemplo de programación

```
(* Simple example *)
LD t#5s
ST delay

(* fault motor *)
LD overspeed
AND hot_temp
ST fault_motor

(* Start the motor after delay (in seconds) *)
CAL MOTOR_TIMER (PT := Delay, IN := start_motor)
LD motor_timer.q
ST motor_run

(* Stop the motor after ten seconds in running *)
cal stop_timer(motor_run,t#10s)
LDN stop_timer.q
ST start_motor
|
```

▶ Asignación

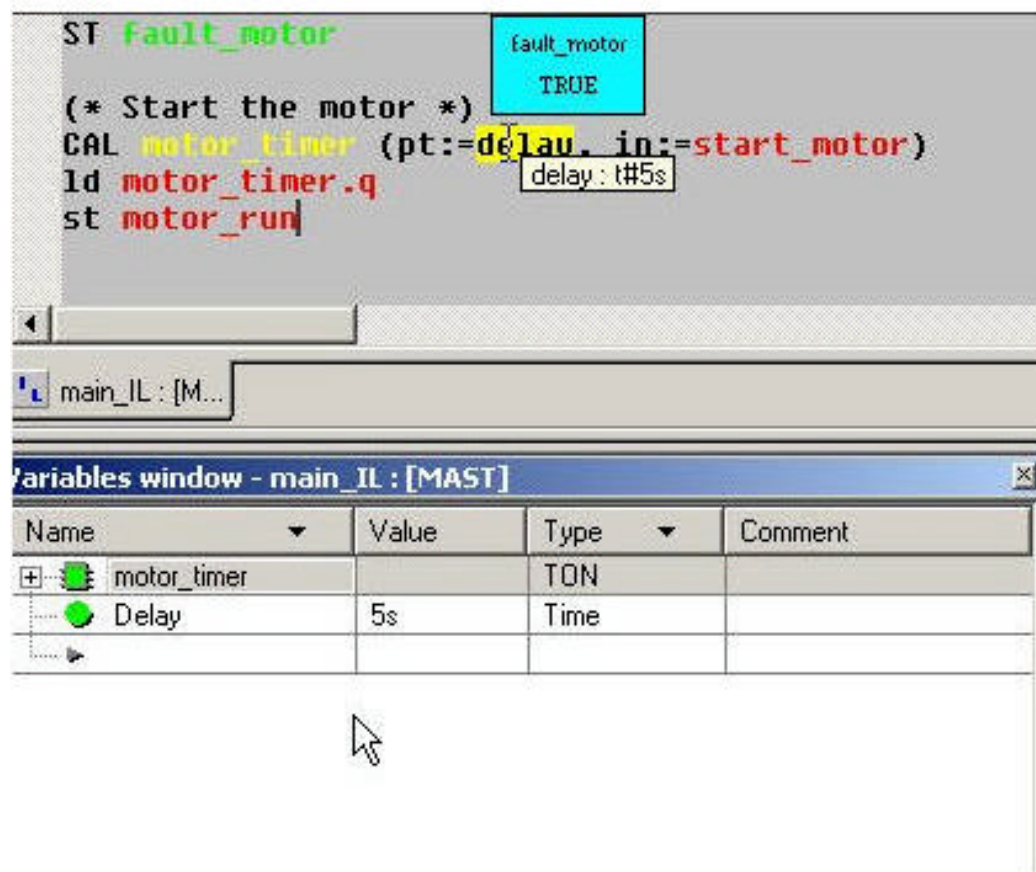
▶ lógica AND

▶ definición Formal de EFB

▶ definición informal de EFB



## ● Pruebas del programa



The screenshot shows the IL editor with the following code:

```

ST fault_motor
(* Start the motor *)
CAL motor_timer (pt:=delay, in:=start_motor)
ld motor_timer.q
st motor_run
  
```

A callout box for 'fault\_motor' shows its value as 'TRUE'. A callout box for 'delay' shows its value as 't#5s'. Below the editor is a 'Variables window - main\_IL : [MAST]' with the following table:

Name	Value	Type	Comment
motor_timer		TON	
Delay	5s	Time	

Animación de la sección IL:

- Variables Booleanas verdaderas. Verde
- Variables Booleana Falsas : Rojo
- Otros tipos de variables: Ambar

Las variables no booleanas no son animadas en el editor pero:

- En tips de pantalla
- Ventanas de inspección
- Ventana de despliegue de variable
- Tablas de animación





## Módulo 3.D

# DFB & Diagnóstico DFB

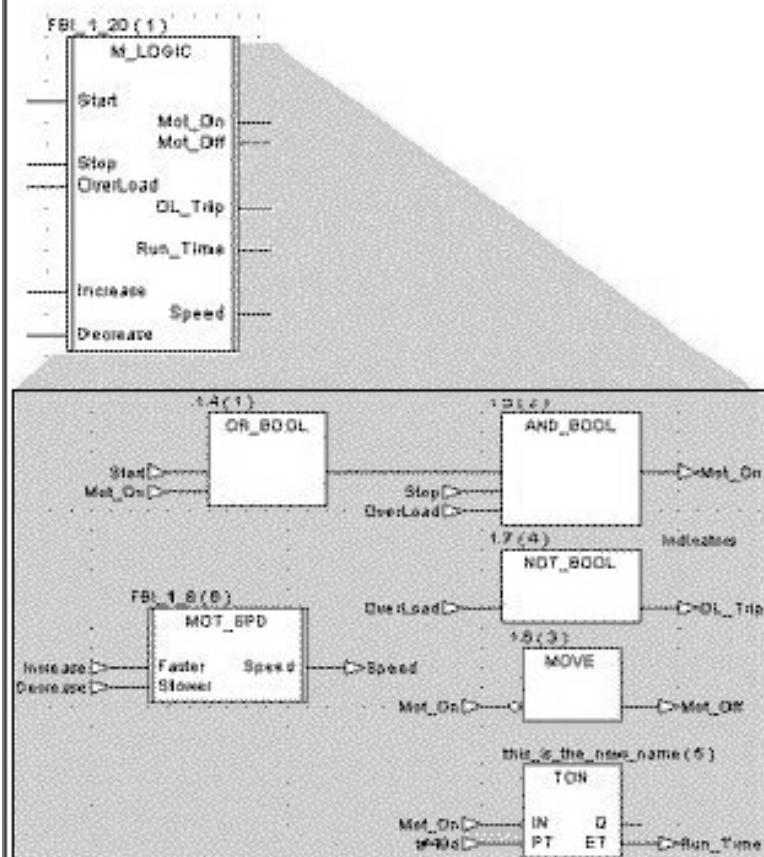
Módulo 3 : Unity Pro

Sección D : DFB &amp; Diagnosticos DFB

Página 1/18

## ● Bloque de función derivado

- DFB : es un bloque encapsulado realizado por el usuario, siendo un bloque reutilizable (escrito para cubrir los requerimientos específicos de una aplicación)
- Utilizando
  - ▶ El diseñador de programas del bloque función DFB llamado 'DFB type'
  - ▶ El usuario final crea una imagen de este bloque llamado "Instance", designando los parámetros utilizados en la aplicación.
  - ▶ Desarrolladores y usuarios pueden emplear las funciones de DFB Exportación / Importación
  - ▶ El código DFB es cargado una sola vez en el CPU
  - ▶ los DFBs pueden ser protegidos
  - ▶ los DFBs son locales a una aplicación ó pueden ser globales en una librería.





## ● Principales definiciones

- DFBs incluye :
  - ▶ parámetros de entrada/salida
  - ▶ variables internas públicas o privadas (declaradas)
  - ▶ una o más secciones escritas en LD, IL, ST, FBD

*Nota : un DFB puede tener anidados DFBs en sus secciones.*

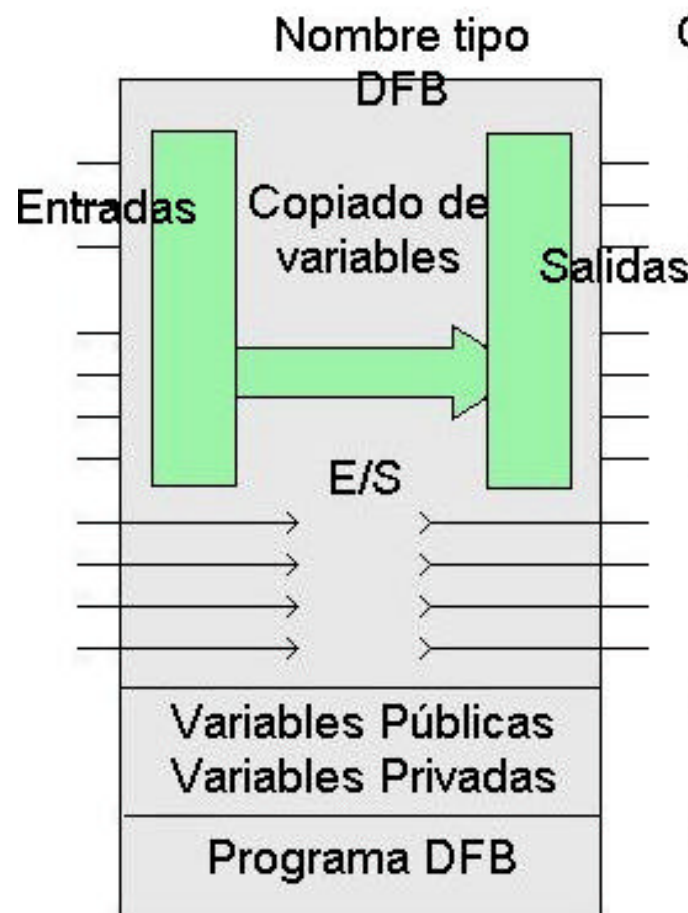
- DFBs también son utilizados para :
  - ▶ estructurar una aplicación
  - ▶ simplificar el diseño y la entrada de datos
  - ▶ mejorar la lectura del programa
  - ▶ Hacer pruebas más facilmente
  - ▶ ptoreger el saber hacer ( know-how)

## ● Principios

- mejorar la disponibilidad del programa
  - Proporcionan, en el lenguaje de escaleras, una vista funcional del bloque haciendo sencilla la programación y las pruebas
- Haciendo sencilla la localización de fallas
  - Las variables del PLC controladas por la función DFB son identificadas en su interface.
- reduce el volumen del código generado
  - El código en el DFB se carga una sola vez, si embargo la función DFB puede utilizarse varias veces en el programa.



## ● Estructura

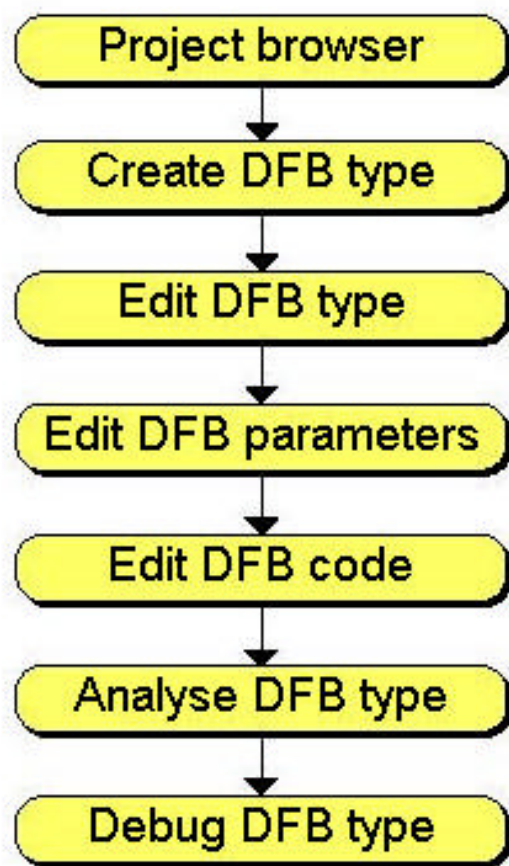


### Creación de un DFB bajo Unity Pro L & XL

- Nombre: 32 caracteres máximo.
- Parametros : entradas, salidas, entradas/salidas
- Variables internas:
  - ▶ publicas : accesible por la aplicación
  - ▶ privadas : no accesibles por la aplicación
- Secciones de Código : escritas en LD, IL, ST o FBD
  - ▶ conforme a IEC, una sola sección
  - ▶ no conforme a IEC , varias secciones (recomendado)
- Comentario : 1024 caracteres máximo sin formato de caracteres
- Archivo descriptivo y nivel de protección

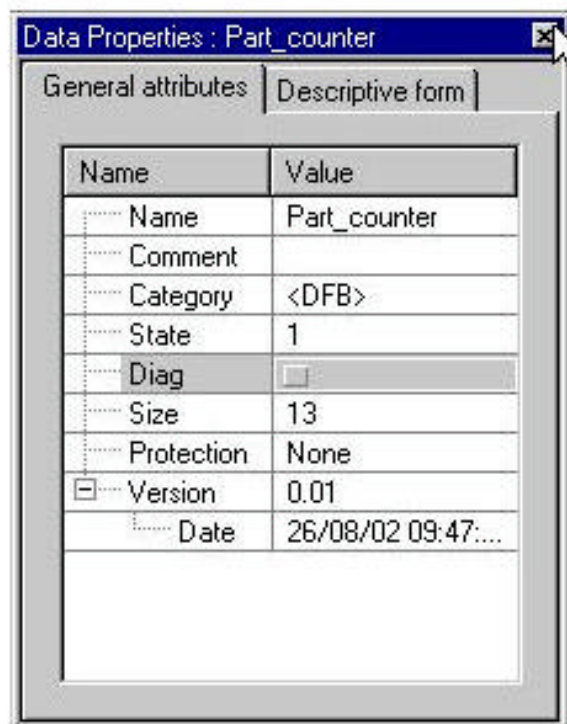


## ● Principios de desarrollo



- Abrir una aplicación Unity Pro
- [Crear una nueva sección tipo DFB](#)
- [Editar las propiedades](#) del nuevo tipo de DFB type que ha sido creado
- [definir los parámetros](#) del DFB
- Introducir el código del DFB en una o más secciones
- [Analizar el tipo de DFB](#) para validarlo
- Pruebe el tipo de DFB
- Exporte el DFB (si es necesario) a otr aaplicación
- Utilise el DFB

## ● Propiedades



- "Tipos DFB " En el editor de datos, seleccionar el DBF y hacer in click derecho => propiedades
- Dar alguna información acerca del tipo de DFB
- Algunos atributos en modo escritura:
  - ▶ Nombre
  - ▶ Comentario
  - ▶ Protección: para elegir el nivel de protección y el password (pregunta automáticamente despues de la selección)  
*Nota : es independiente de la protección de la aplicación*
  - ▶ Diag : Para definir si el DFB será utilizado en diagnóstico

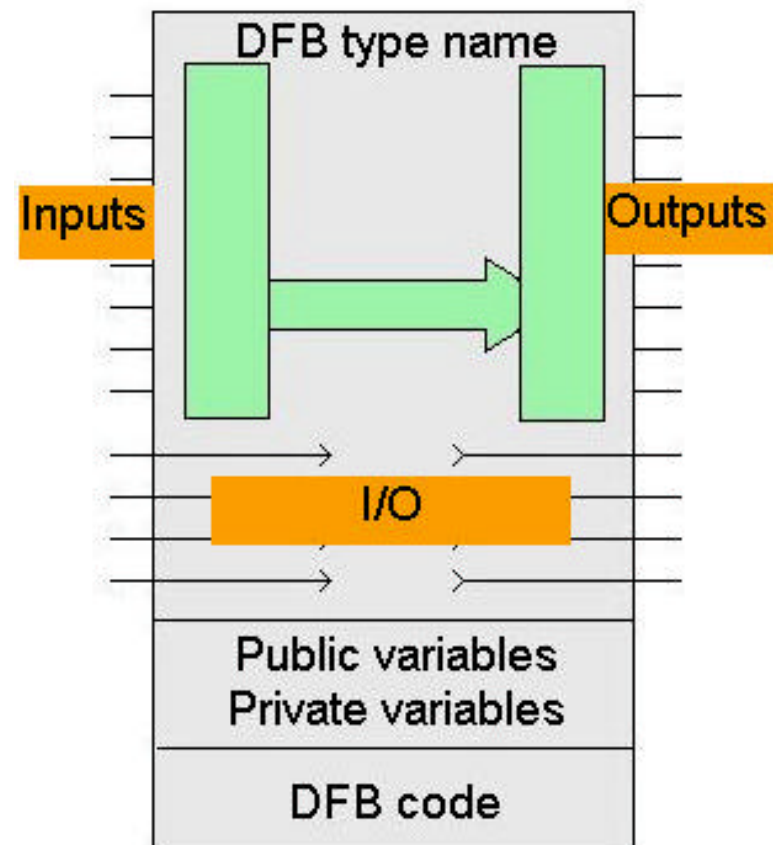


## ● Propiedades

Protección DFB :

Cuando un DFB es protegido en Lectura "Read Only" o en modo NO lectura ni escritura "No read& Write" es posible exportar el DFB. Pero el archivo ".XDB" permanece codificado.





## ● Interface E/S

- **entradas : 32 max**
  - ▶ En lectura unicamente por el DFB
  - ▶ para transferir el valor de la aplicación DFB
  - ▶ tipo y número (32 caracteres máximo)
- **Salidas : 32 max**
  - ▶ En escritura solamente por el DFB
  - ▶ Para transferir el valor desde el DBF a la aplicación
  - ▶ tipo y número (32 characters max)
- **Entradas/Salidas : 32 max**
  - ▶ en lectura/escritura por el DFB
  - ▶ Transferencia del valos del DBF modificandolo y regresandolo a la aplicación
  - ▶ Tipo y nombre (32 caracteres máximo)

## ● Interface E/S

Para cada uno de los parámetros del DFB, las siguientes familias pueden ser utilizadas :

<b>Familias de objetos</b>	<b>EDT</b>	<b>DDT (1)</b>	<b>IODDT</b>	<b>ANY_...(2)</b>	<b>ANY_ARRAY</b>	<b>FB</b>
Inputs	SI	SI	NO	SI (4)	SI	NO
Inputs/outputs	SI (3)	SI	SI	SI (4)	SI	NO
Outputs	SI	SI	NO	SI (4)(5)	NO	NO

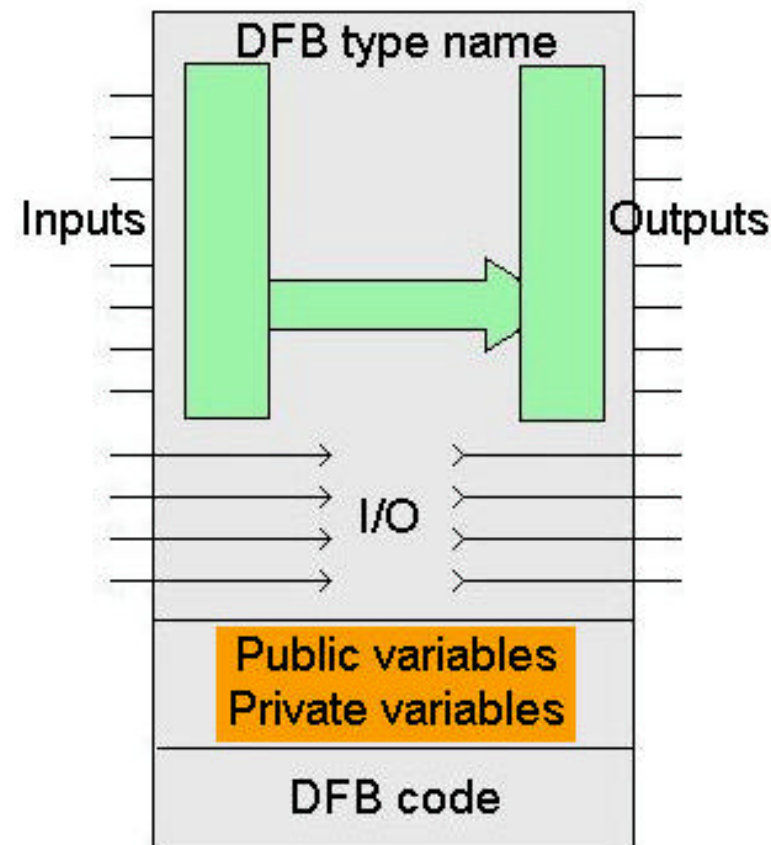
(1) Datos Derivados, excepto datos derivados tipo (IODDT)

(2) Todas las familias de datos genéricos, excepto la familia ANY\_ARRAY ,

(3) Excepto para el tipo EBOOL variables estáticas , con PLC's Quantum,

(4) Excepto para variables tipo BOOL y EBOOL,

(5) Estos parámetros no pueden ser utilizados fuera del bloque función llamado.



## ● Variables

- Privada : ilimitada
  - ▶ Variable interna del DFB
  - ▶ únicamente utilizada para programar el DFB (ejemplo :cálculo inetrmedio)
  - ▶ nombre (8 caracteres máximo)
- Publica : ilimitada
  - ▶ variable interna del DFB
  - ▶ puede ser accesada por la aplicaicón o el usuario en el modo ajuste fuera del bloque (derechos de escritura y lectura)
  - ▶ nombre (8 caracteres máximo)

los valores de las variables públicas, son modificadas por el programa o por ajustes del usuario y pueden ser guardadas poniendo a 1 el registro %S94 to 1, en lugar de los valores iniciales.



## ● Variables

Para cada parámetro del DFB las siguientes familias pueden ser utilizadas :

<b>Familia de objetos</b>	<b>EDT</b>	<b>DDT (1)</b>	<b>IODDT</b>	<b>ANY_...(2)</b>	<b>ANY_ARRAY</b>	<b>FB</b>
variables Publicas	Si	Si	No	No	No	No
variables Privadas	Si	Si	No	No	No	Si

(1) Familia de datos derivados, excepto datos tipo derivados de entrada/salida (IODDT)

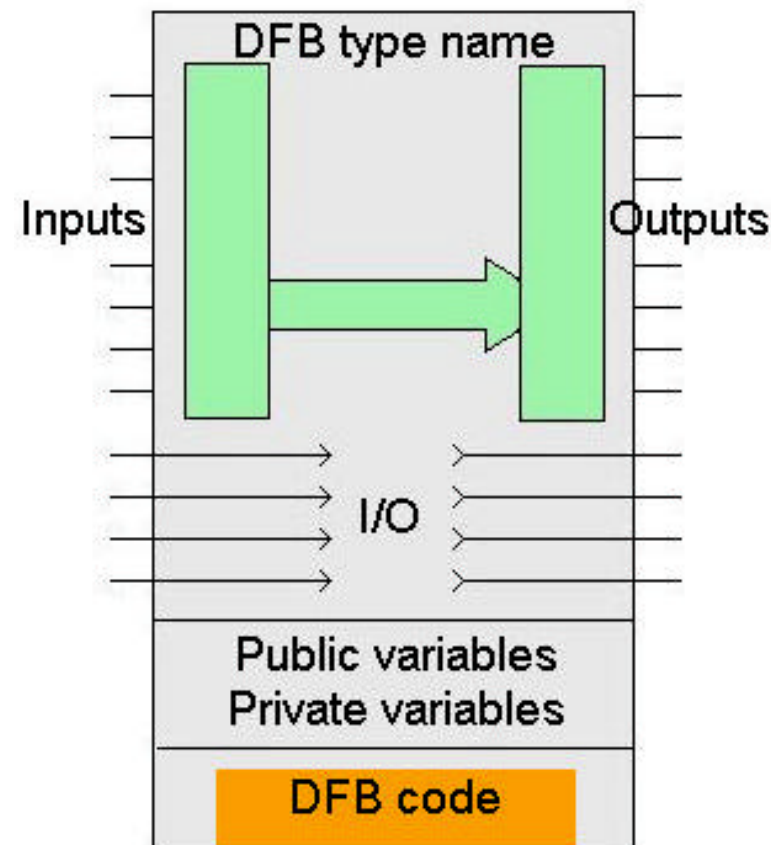
(2) Todas las familias genéricas, excepto la familia ANY\_ARRAY,



Módulo 3 : Unity Pro

Sección D : DFB & Diagnosticos DFB

Página 8/18



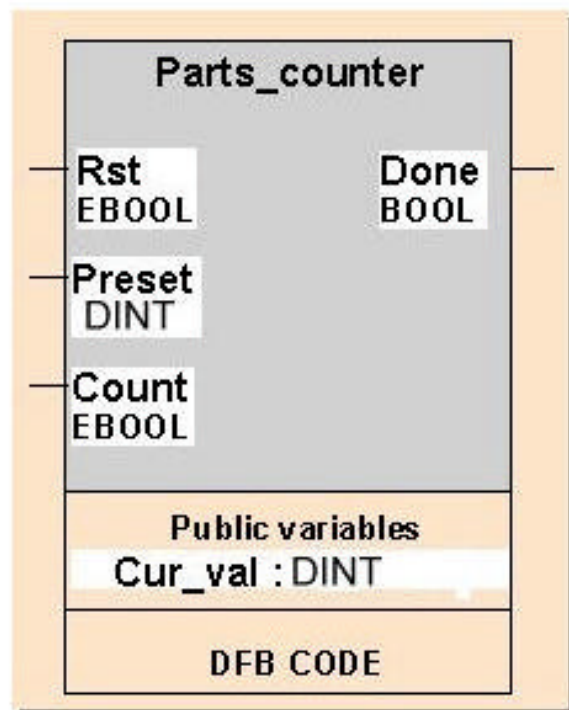
## ● Código

- Estructurado en secciones :
  - ▶ Conforme a IEC, una sola sección
  - ▶ No cumple con IEC, varias secciones
- Sección
  - ▶ escrita en instrucciones listado, texto estructurado ,lenguaje ladder,diagrama de bloque funcional
  - ▶ un nombre simbólico (32 caracteres máximo)
  - ▶ una condición de validación
  - ▶ atributos de protección (nsin protección, portegido en escritura , ptotección lectura/escritura)
  - ▶ comentario (máximo 256 caracteres)
- Utilizar solo el tipo de parámetros definidos para el bloque función o el sistema palabras y bits

Módulo 3 : Unity Pro

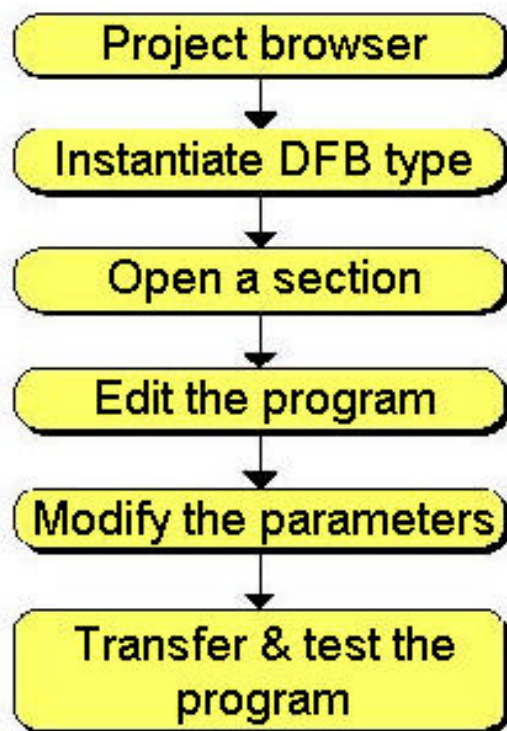
Sección D : DFB & Diagnosticos DFB

Página 9/18



## ● Ejemplo

- Nombre : Parts\_counter
- Entradas :
  - ▶ Rst : Reset del contador
  - ▶ Preset : Valor de Preset del contador
  - ▶ Count : Entrada del contador
- Salida :
  - ▶ Done : Valor de preset alcanzado
- Variable Pública :
  - ▶ Cur\_val : Valor incrementado por la entrada de conteo
- Código
- Forma descriptiva



## ● Utilizando principios

- Abrir una aplicación Unity Pro
- Importar un tipo de DFB type (si es necesario con control de la versión )
- Inicializar el tipo de DFB
- Abrir una sección de programa
- Editar el programa con DFB
- Modificar los parámetros (valores iniciales)
- Transferencia y pruebas del programa

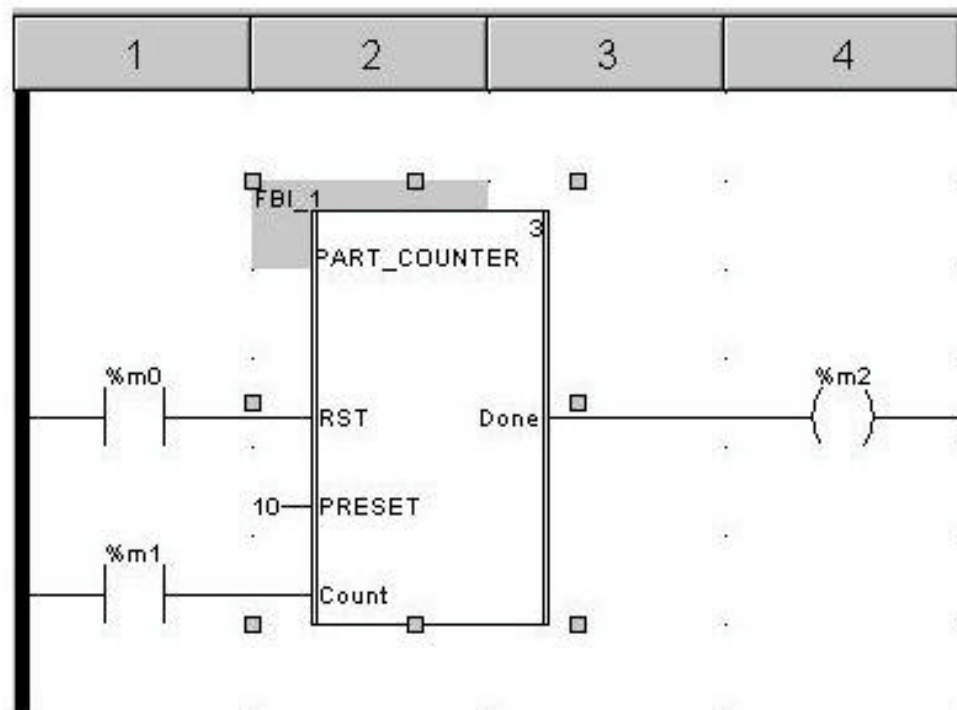


Módulo 3 : Unity Pro

Sección D : DFB & Diagnosticos DFB

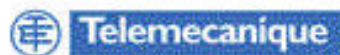
Página 11/18

## ● Utilizando un ejemplo



- programación del DFB:
  - ▶ DFB programado bajo Unity Pro M, L or XL
  - ▶ en todas las tareas de aplicaciones in all application excepto en las tareas de evento y transiciones SFC.
- instancia utilizada DFB como:
  - ▶ un bloque estandar en lenguaje LD o FDB
  - ▶ una función elemental en lenguaje ST o IL





Módulo 3 : Unity Pro

Sección D : DFB & Diagnosticos DFB

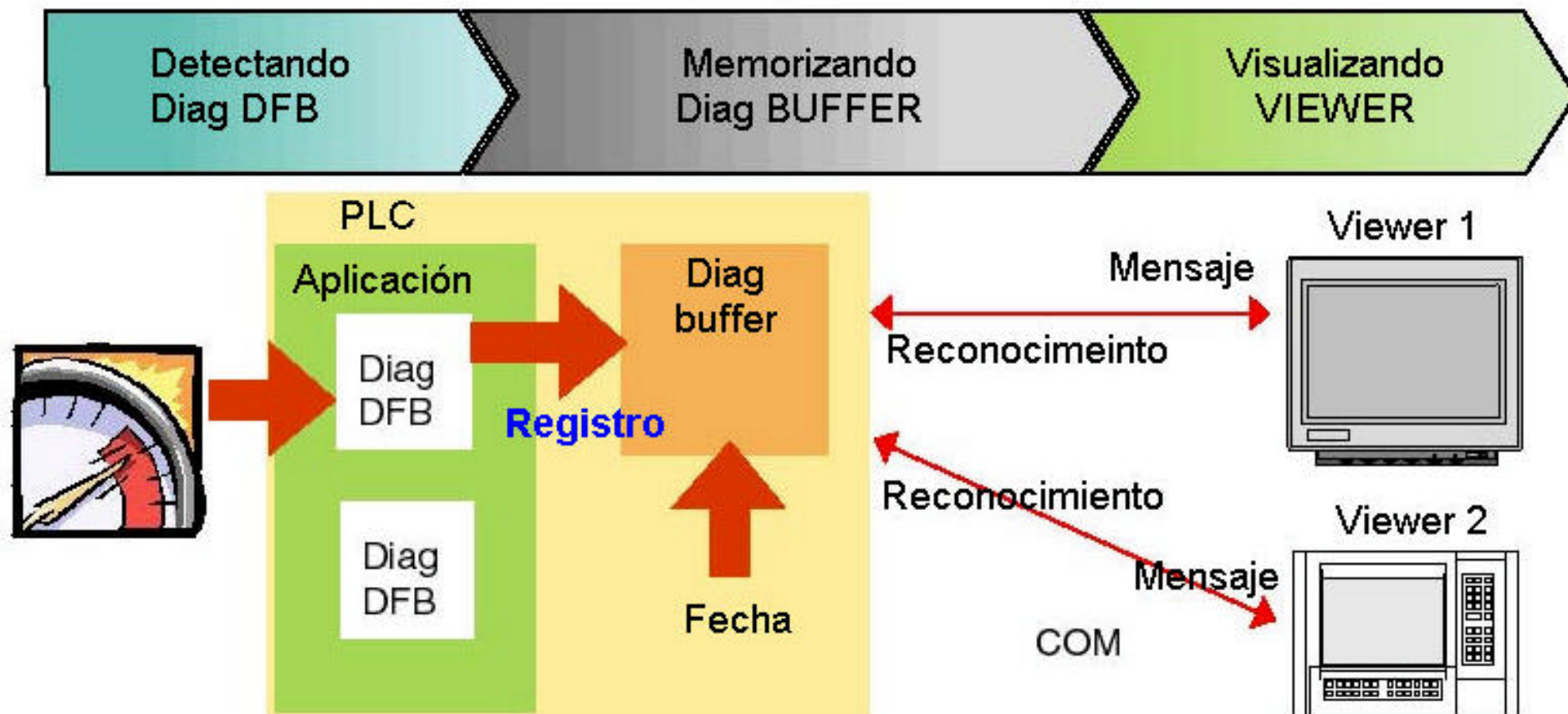
Página 12/18

# Diagnostico DFBs

## ● Panorama

- La herramienta adecuada para el diagnóstico de la aplicación
  - ▶ Aplicación controlada en el programa del PLC
- **Diagnóstico integrado**
  - ▶ Hacer un diagnóstico a la fuente (directamente al nivel del PLC)
- Es posible crear un propio diagnóstico con DFBs (Quantum y Premium)
- Lista de DFBs predefinidos por Schneider
  - ▶ ALRM\_DIA : monitoero de eventos simplificados (Quantum & Premium)
  - ▶ EV\_DIA : monitoreo de eventos (Quantum & Premium)
  - ▶ MV\_DIA : monitoreo de movimientos (Quantum & Premium)
  - ▶ ASI\_DIA & A2SI\_DIA : monitoreo de dispositivos en un Bus ASI (solamente para Premium)

## ● Principio



## ● Registro

- Registro de alarmas
  - ▶ La instrucción de registro de error n **REGDFB ( )** escribe un tiempo almacenandolo en el Diag buffer cuando el error ocurre.
  - ▶ Muchos registros pueden ser controlados dentro de un solo DFB (ejemplo : hasta 4 registros)
  - ▶ **Uso unicamente dentro de un DFBs de diagnóstico**
- Sintaxis

**REGDFB (AREA, 16#0062, SLEN, CTRL, PIN, VALPIN, ESTS, ERID, STAT);**



## ● Registro

parámetros de entrada

Parámetros	Tipo	Comentarios
<i>AREA</i>	INT	Permite asociarlo a una área específica en el monitoreo de la aplicación por el DFB : 0 a 15 Cada Zona puede controlar varios visualizadores, cada visualizador puede controlar zonas separadas independientemente.
<i>CLAS</i>	INT	Clase de error : <ul style="list-style-type: none"><li>• 16#0062 : Erros de utilización DFB</li></ul>
<i>SLEN</i>	INT	Longitud del estatus: 0, 2 o 4 bytes: <ul style="list-style-type: none"><li>• 0 = sin estatus de control,</li><li>• 2 = estatus de control en una palabra</li><li>• 4 = estatus de control en una doble palabra</li></ul>
<i>CTRL</i>	BOOL	1 = reconocimiento del operador requerido 0 = reconocimiento no requerido
<i>PIN</i>	INT	número de entrada incorrecto
<i>VALPIN</i>	BOOL	Valor alcanzado en la entrada de error

## ● Eliminación del registro

- Alarma Eliminación de registro
  - ▶ La instrucción 'eliminación de registro' **DEREG** ( )actualiza la memoria con la fecha y tiempo en que el error ha desaparecido
  - ▶ El mensaje será borrado dentro de la memoria unicamente cuando en todos los "visualizadores" ( terminales de visualización ) han reconocido la información
  - ▶ **Utilizado unicamente dentro de un DFBs de diagnóstico**
- Sintaxis

**OUT := DEREG(ERID);**

*Nota: La alarma de mantiene almacenada en la memoria de diagnóstico hasta que el error sea reconocido ( Para errores que requieran reconocimiento ) y es visto por todos los visualizadores .*

## ● "de-registration"

### Parámetros de entrada

Parámetros	Tipo	Comentario
<i>ERID</i>	INT	Identificador de error

### Parámetros de salida

Parámetros	Tipo	Comentario
<i>OUT</i>	INT	<p>Registro de reporte de error</p> <ul style="list-style-type: none"><li>• Si la función "de-registration" es exitosa, OUT = 0,</li><li>• Si la función "de-registration" falla :<ul style="list-style-type: none"><li>○ OUT = 1: bufer de diagnóstico no configurado,</li><li>○ OUT = 21: error de indentificador incorrecto,</li><li>○ OUT = 22: ningún error se ha registrado con este indentificador</li></ul></li></ul> <p>La palabra de sistema %SW77 es reservada para recepción del resultado del diagnóstico DFB "de-registration"</p>





## ● Creación

- Activar la opción de diagnóstico dentro de la aplicación (Tool => Project settings)
- crear un nuevo DFB en el editor de variables o Importando el código fuente del "modelo" Diag DFB si es necesario
- En la ventana de este DFB, activar la opción **Diag**
- Declarar la Entrada/Salida del DFB y variables internas

*Nota : todas las entradas monitoreadas deben tener su propia propiedad **Diag** seleccionada*

Diagnosics

Application diagnostics

System diagnostics

Language: English

Application level: No cause of fault

Data Properties : Test\_diag

General attributes | Descriptive form

Name	Value
Name	Test_diag
Comment	
Category	<DFB>
State	1
<b>Diag</b>	<input checked="" type="checkbox"/>
Size	0
Protection	None
Version	0.0



## ● Creación (cont)

- Teclar la descripción
- Teclar el código
  - ▶ utilizar la sininstrucciones **REGDFB()** y **DEREG()**

*Nota : El parámetro PIN indica el orden del número de entradas monitoreadas*

- configure el DFB
  - ▶ Personalise el mensaje de error o falla
- Pruebe el DFB
- Exporte el DFB

```

IF NOT ERROR THEN
  (* We haven't yet reported a
  REGDFB ( AREA := AREA_NR,
           CLAS := 16#0062,
           SLEN := 0,
           CTRL := OP_CTRL,
           PIN := PIN_NB,
           VALPIN := PIN_VAL ,
           ESTS => INTSTAT ,
           ERID => ERROR_ID,
           STAT => %SW76);

IF (%SW76 <> 0) THEN
  SET( OUT => BUFPLEIN

```

## ● Creación (cont)

Ejemplo :

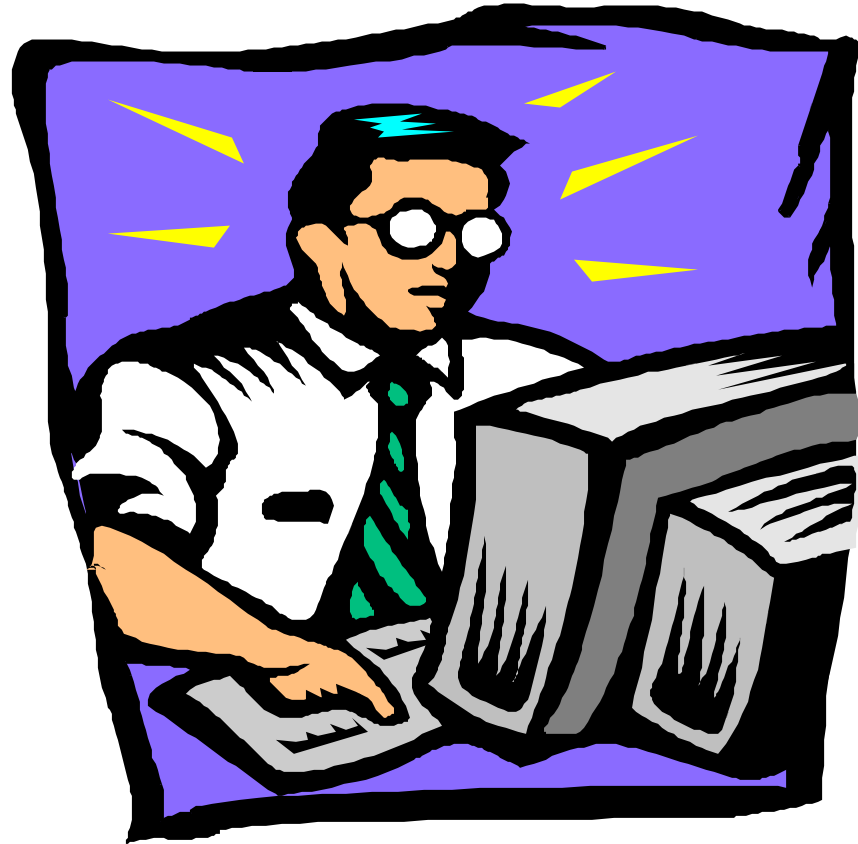
Un diagnóstico DFB con cinco entradas y 3 salidas es monitoreado.

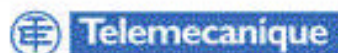
<b>Entrada</b>	<b>número físico</b>	<b>Monitoreo (Diag checked)</b>	<b>número de monitoreo</b>
Input_0	1	Yes	1
Input_1	2	Yes	2
Input_2	3	No	-
Input_3	4	Yes	3
Input_4	5	No	-

El Parámetro PIN del registro de función tiene el siguiente valor:

- PIN = 1
- PIN = 2
- PIN = 3

# *Ejercicio 11*





Módulo 3 : Unity Pro  
Sección E : Diagnóstico y Pruebas  
Página 0/20

Contenido

Siguiente Página

Salir



Ayuda

## Módulo 3.E

# Diagnóstico y funciones de prueba



Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 1/20

## ● Probando el programa

Para disminuir el tiempo de puesta en servicio y pruebas, Unity Pro provee los siguientes servicios:

- Animación dinámica del programa
- Inserción de un bandera de paro en un elemento del programa
- Ejecución del programa en modo paso a paso incluyendo adicionales como paso: dentro, fuera, durante
- Llamado de Pila de datos para verificar el programa entero
- Manejo de valores de Entrada/Salida utilizando la tabla de animación integrada a las pantallas del operador

Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Página 2/20

## ● Animación dinámica

Name	Value	Type	C..
● Level_1	500	Int	
● Level_2	0	Int	
● Level_3	270	Int	
● Box	0	Int	

```

Security : [MAST]
IF Level_1 >= 600 THEN Pump_1 := False;
END_IF;
IF Level_2 >= 200 THEN Pump_2 := False;
END_IF;
IF Level_3 >= 400 THEN Valve_1 := False;
Valve_2 := False;
END_IF;
IF Box = 10 THEN Valve_3 := False;
END_IF;
IF Level_3 < 80 THEN Heater_on := False;
Mixing_2 := False;

```

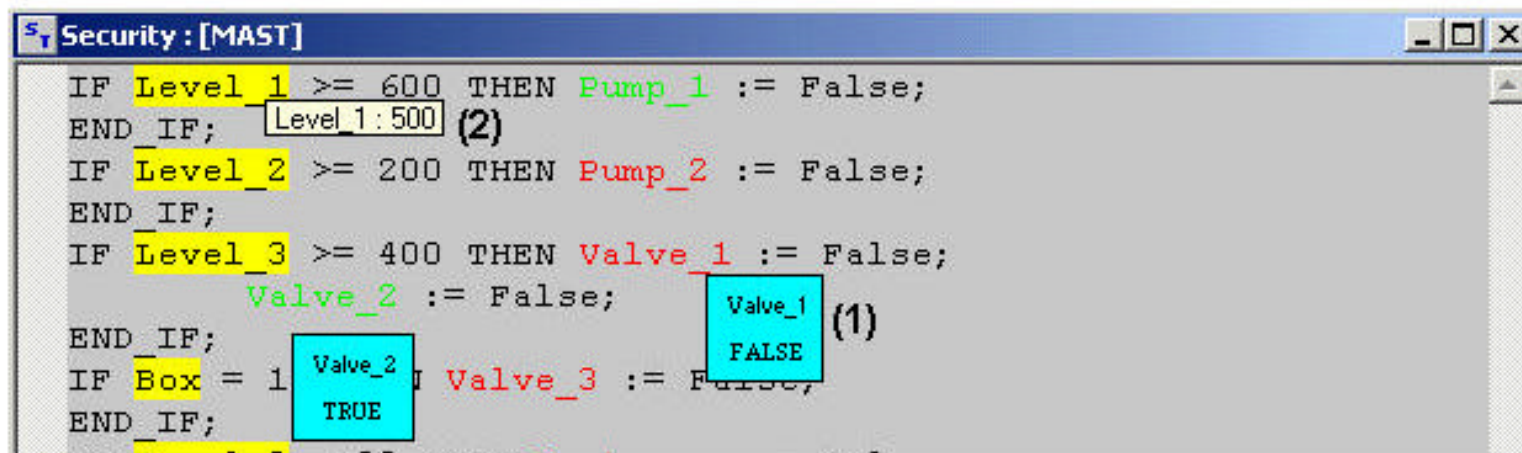
- Las variables Boleanas y analógicas son animadas directamente por el editor del programa (verde, rojo o amarillo)
- **Ventana de Variables** muestra las variables (excepto las variables booleanas) utilizadas en la ventana active del editor

Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Página 3/20

## ● Ventanas de inspección



```

Security : [MAST]
IF Level_1 >= 600 THEN Pump_1 := False;
END_IF;
IF Level_2 >= 200 THEN Pump_2 := False;
END_IF;
IF Level_3 >= 400 THEN Valve_1 := False;
Valve_2 := False;
END_IF;
IF Box = 1 THEN Valve_3 := False;
END_IF;

```

- **Ventana de inspección (1) :**

ligada a la variable despliega el valor.El color de la ventana dependerá del valor comparado con el valor máximo y mínimo (amarillo, azul o magenta)

- **Tool tip (2)** despliega el valor cuando el cursor está sobre la variable



Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Página 4/20

## ● Tablas de animación

```

Security : [MAST]
IF Level_1 >= 600 THEN Pump_1 := False;
END_IF; Level_1: 500 (2)
IF Level_2 >= 200 THEN Pump_2 := False;
END_IF;
IF Level_3 >= 400 THEN Valve_1 := False;
Valve_2 := False; Valve_1 (1)
END_IF;
IF Box = 1 Valve_2 TRUE Valve_3 := False;
END_IF;

```

- Modificar el valor de las variables
- Modificar el formato de monitoreo de los registros (binario, hexadecimal,...)
- Forzamineto de variables booleanas.



Módulo 3 : Unity Pro

Sección E : Diagnóstico y Prueba

Página 5/20



Inserte / borre  
breakpoint

Breakpoint

## ● Bandera de paro "breakpoint"

```
Security : [MAST]
IF Level_1 >= 600 THEN Pump_1 := False;
END_IF;
● IF Level_2 >= 200 THEN Pump_2 := False;
END_IF;
IF Level_3 >= 400 THEN Valve_1 := False;
Valve_2 := False;
END_IF;
```

- Detiene la ejecución de tareas para inspeccionar el código, comportamiento y valor de las variables
- La bandera de paro no está disponible en la tarea de eventos
- Una bandera de paro se puede definir en todo el programa

Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Página 6/20

## ● Bandera de visualización "Watch point"



Insertar /borra  
Watchpoint

Watchpoint

Security : [MAST]

```
IF Level_1 >= 600 THEN Pump_1 := False;
END IF;
IF Level_2 >= 200 THEN Pump_2 := False;
END IF;
IF Level_3 >= 400 THEN Valve_1 := False;
Valve_2 := False;
END IF;
IF Level_4 >= 100 THEN Valve_3 := False;
```

- La bandera de visualización permite inspeccionar variables, exactamente cuando la línea de programa es ejecutada y no al final del barrido o escan
- La bandera de visualización no está disponible en las tareas de eventos
- Solamente una ventana de visualización puede ser definida en todo el proyecto

## ● bandera de visualización Watch point (cont)



sincroniza la  
tabla de  
animación

curity : [MAST]]

Name	Value	Type	Comment
● Level_1	500	Int	
● Pump_1	1	Bool	
● Level_2	0	Int	
● Pump_2	0	Bool	
● Level_3	270	Int	
● Valve_1	0	Bool	

Indica la  
sincronización  
con la tabla

- La tabla de datos puede ser sincronizada con la bandera de visualización
- Un contador da el ciclo de actualización



## ● Paso a Paso (LD/ST/IL/FBD)



comandos paso a paso

Bandera

```

END_IF;
IF Niveau_2 >= 200 Then Pompe
END_IF;
IF Niveau 3 >= 400 Then Vanne
    Vanne_2 := False;
END_IF;

```



```

END_IF;
IF Niveau 3 >= 400 Then Vanne
    Vanne_2 := False;
END_IF;
IF Boite = 10 Then Vanne_3 :=
End IF;

```

- El paso a paso consiste en la ejecución del programa elemento por elemento
- El modo Paso a paso es lanzado con una bandera o breakpoint
- 3 comandos pueden ser utilizados en el modo paso a paso : **Step Over**, **Step Into** y **Step Out**
- El modo paso a paso se detiene en la tarea correspondiente



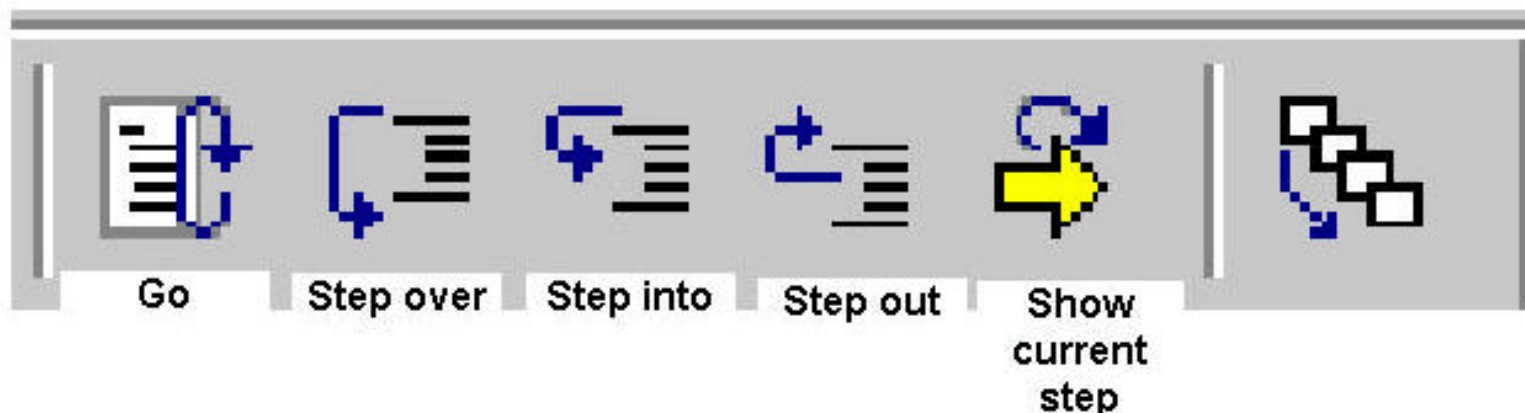
Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 9/20

Comentario

## ● Comandos Paso a paso

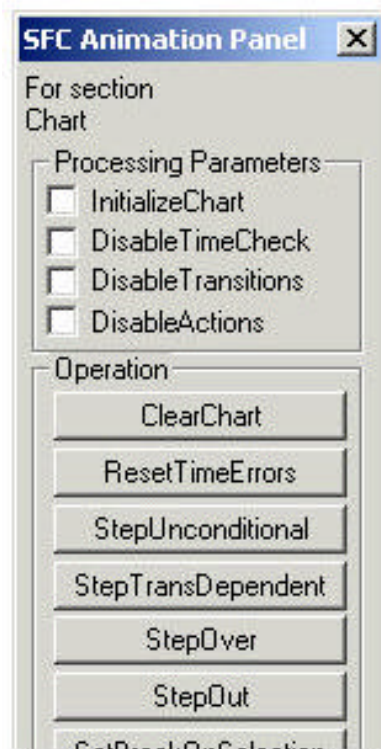


- Ir **"Go"** ejecuta el código hasta la bandera de Paso (ciclo por ciclo)
- Paso sobre **"Step over"** ejecuta el elemento actual
- Paso adentro **"Step into"** del código en el elemento actual
- Paso fuera **"Step out"** del código del elemento actual
- muestra paso actual **"Show current step"**

## ● Comandos Paso a Paso

Tres comando pueden ser utilizados en el modo paso a paso estos son:

- El comando Paso dentro **"Step Into"**:
  - Si el elemento actual es una línea que no contiene una llamada a subrutina o un DFB  
este comando se ejecuta la línea y se mueve a la siguiente línea
  - Si el elemento en curso es una instrucción, este comando se ejecuta y se mueve a la siguiente instrucción.
  - Si el elemento actual es bloque función, se ejecuta este comando y se dirige al siguiente bloque función,
  - Si el elemento actual es una llamada a una instancia SR ó DFB, puede ser usado para entrar al código e ir al primer elemento del SR ó DFB.



comando  
 paso a paso

## ● Paso a Paso (SFC)

- La función paso a paso consiste en la ejecución de una sección SFC paso a paso
- El modo Paso a Paso es lanzado desde una bandera
- Utilice en el panel de animación **SFC** la **herramienta de pruebas de una sección SFC** y active el modo paso a paso
- **comando "Step Into" no está disponible**
- el modo Paso a paso no detiene la tarea ,pero congela la gráfica correspondiente
- La animación de la gráfica SFC no está disponible en el modo de pruebas de macro etapas

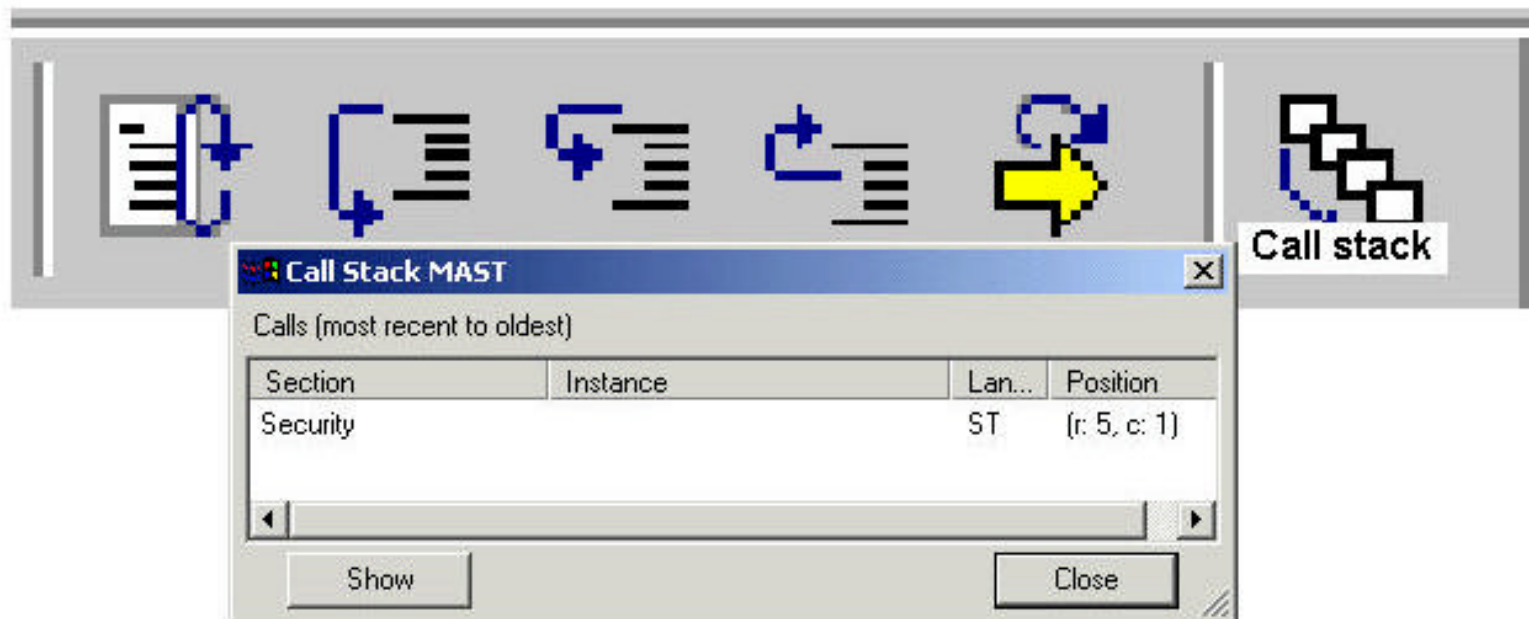


Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 11/20

## ● Llamado de Pila "Call stack"



- Muestra la pila de datos en el paso actual. En el caso de múltiples comandos **Step Into**. Esta opción permite que usted vea el camino completo desde la primer bandera "breakpoint"

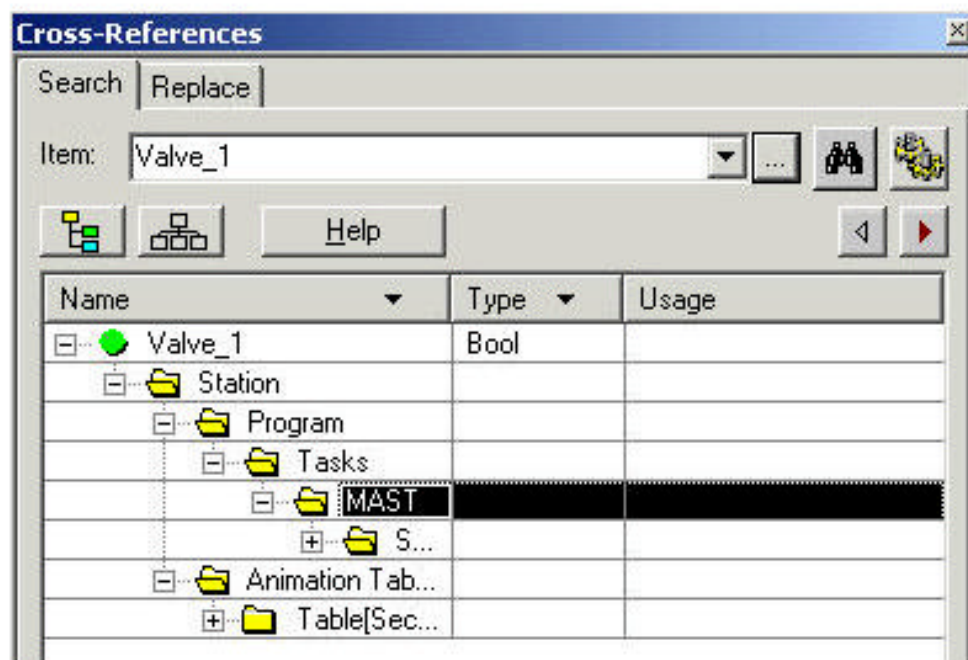


Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Página 12/20

## ● Referencias Cruzadas



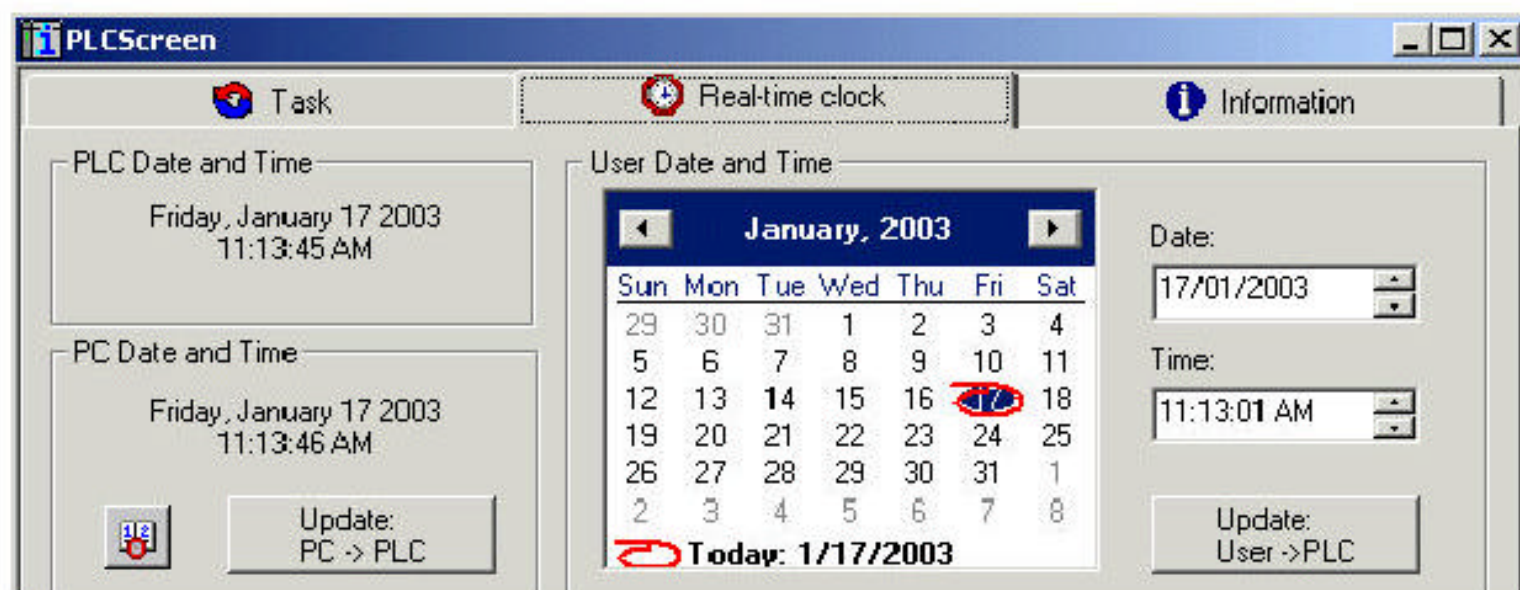
- La herramienta de referencia cruzada le permite ver el uso de las variables en cualquier parte de la aplicación y como es utilizada

Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 13/20

## ● Pantalla de prueba y ajuste PLC



The screenshot shows the 'PLCScreen' application window. It features three main tabs: 'Task', 'Real-time clock', and 'Information'. The 'Real-time clock' tab is active, displaying two date and time synchronization sections: 'PLC Date and Time' and 'PC Date and Time'. Both sections show the date 'Friday, January 17 2003' and the time '11:13:45 AM' and '11:13:46 AM' respectively. A central calendar for 'January, 2003' is displayed, with the 17th of January highlighted in red. To the right of the calendar are input fields for 'Date:' (17/01/2003) and 'Time:' (11:13:01 AM), each with a dropdown arrow. Below these fields are two 'Update:' buttons: 'Update: PC -> PLC' and 'Update: User -> PLC'. At the bottom of the calendar, it says 'Today: 1/17/2003'.

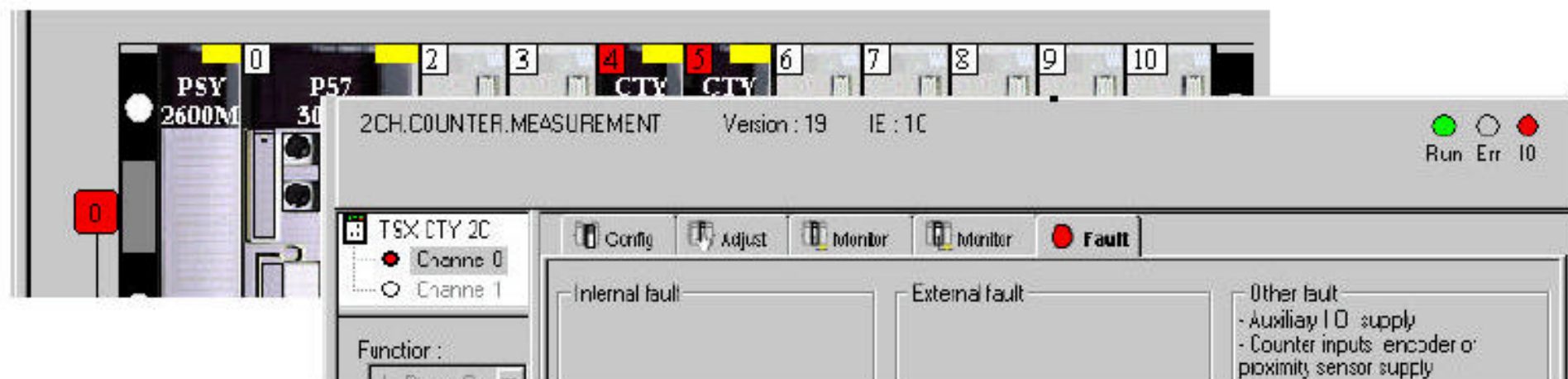
- Estatus y ejecución de las tareas del programa
- Ajuste de reloj en tiempo real
- Información del proyecto y el procesador PLC

Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 14/20

## ● Modulo de reporte da fallas



Una falla des enmascarada es reportada:

- En el módulo en la ventana de vista centralizada
- En la ventana de configuración en el rack (Led rojo)
- En todas las pantallas del módulo en falla o canal (led rojo y listado de fallas)
- Con el lenguaje dedicado a los objetos (bits de error y palabras de estatus)



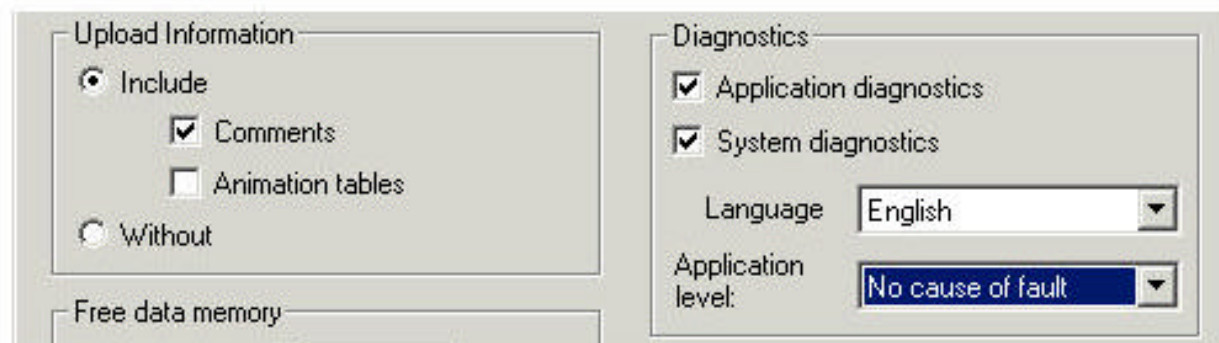
Módulo 3 : Unity Pro

Sección E : Diagnóstico y pruebas

Page 15/20

## ● Sección de diagnóstico integrado

Utilizando la ventana de ajuste de proyectos, seleccione **Aplicacion /Diagnóstico del sistema para desplegar** los mensajes de errores el visualizador de diagnóstico



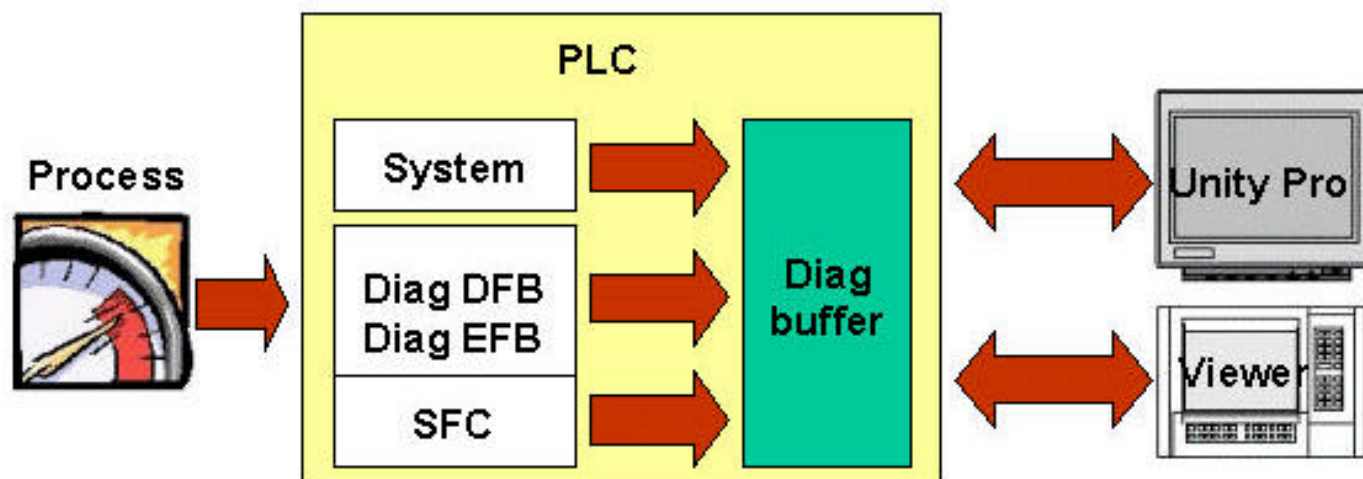
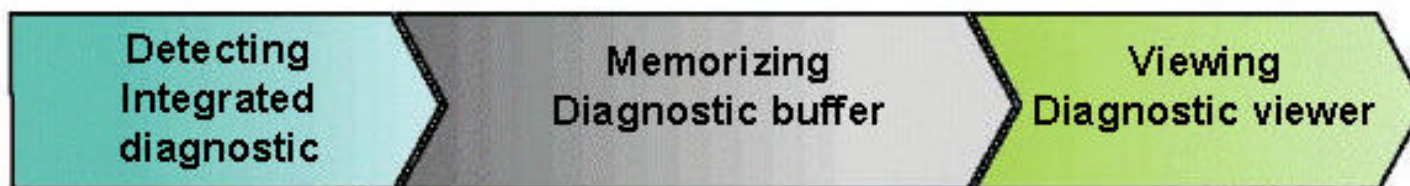
The screenshot shows a dialog box with two main sections: "Upload Information" and "Diagnostics".

- Upload Information:**
  - Radio button "Include" is selected.
  - Checkboxes: "Comments" (checked), "Animation tables" (unchecked).
  - Radio button "Without" is unselected.
  - Section "Free data memory" is partially visible at the bottom.
- Diagnostics:**
  - Checkboxes: "Application diagnostics" (checked), "System diagnostics" (checked).
  - Language: "English" (dropdown menu).
  - Application level: "No cause of fault" (dropdown menu).

- Elija el idioma en el cual se visualizarán los errores
- Elija desplegar o no la causa de la falla (Nivel de aplicación)



## ● Diagnóstico y arquitectura



## ● Diagnóstico del sistema

- El diagnóstico del sistema es automático (trabaja sin programación )
- Todos los errores del sistema son reconocidos y guardados en el PLC
- Mensajes de error por medio de Unity Pro
- Diagnóstico del sistema utiliza el reporte de fallas de los bits sistemas y palabras (ejemplo: %S68 indica el estado de la batería de respaldo de un procesador Premium, ...)

## ● Diagnóstico del proceso

El proceso de diagnóstico es implementado utilizando los EFB / DFB

- en el SFC el diagnóstico es integrado directamente
  - ▶ Tiempo de errores de etapas
  - ▶ Errores de Transición y errores
- Diagnóstico EFB (plataformas Premium y Quantum)
  - ▶ D\_ACT : diagnóstico de acciones y combinación extendida
  - ▶ D\_DYN : diagnóstico dinámico extendido
  - ▶ D\_GRP : monitoreo extendido de un grupo
  - ▶ D\_LOCK : diagnóstico extendido "locking" diagnostics with reacción de entrada
  - ▶ D\_PRE : monitoreo extendido de los requerimientos del proceso
  - ▶ D\_REA : diagnóstico extendido (reacción)
  - ▶ DEREG : Alarma de registro "de-registration"
  - ▶ REGDFB : alarma guardado y fechado



Módulo 3 : Unity Pro

Sección E : Diagnóstico y Prueba

Página 19/20

## ● Diagnóstico del proceso (cont)

- Diagnóstico DFB (solamente en plataforma Premium)
  - ▶ ALRM\_DIA : interface con el diagnóstico de la memoria "diagnostic buffer "
  - ▶ A2SI\_DIA : monitoreo del bus AS-i V2
  - ▶ ASI\_DIA : monitoreo del bus AS-i
  - ▶ EV\_DIA : monitoreo de eventos
  - ▶ MV\_DIA : monitoreo de movimiento "motion"
  - ▶ NEPO\_DIA : comando y diagnóstico de la parte operativa
  - ▶ TEPO\_DIA : comando y diagnóstico de la parte operativa
- Diagnóstico del usuario DFB
  - ▶ Posibilidad de crear su propio diagnóstico DFB

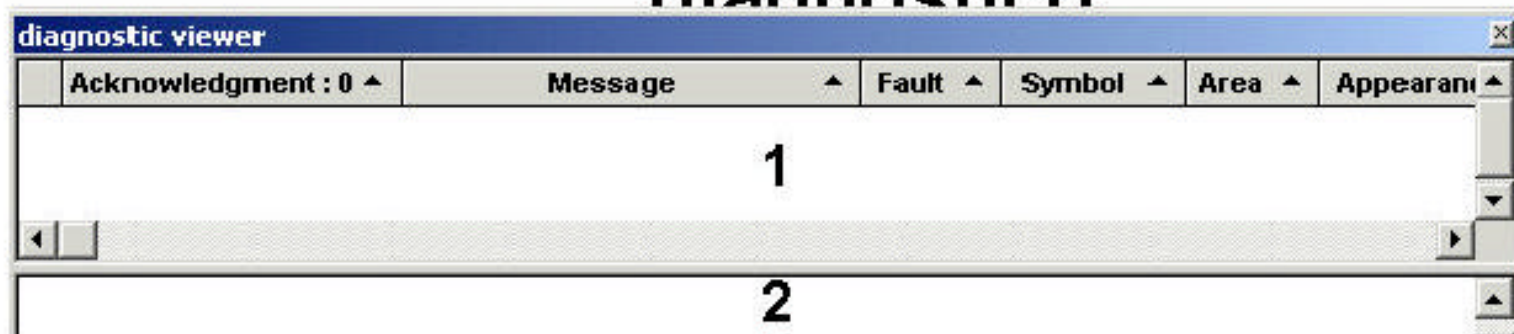


Módulo 3 : Unity Pro

Sección E : Diagnóstico y Pruebas

Página 20/20

## ● Visualizador del diagnóstico



Acknowledgment : 0 ▲	Message ▲	Fault ▲	Symbol ▲	Area ▲	Appearance ▲
		1			
		2			

- (1) Lista de mensajes de erros (columnas son configurables)
  - ▶ Bit de sistema o diagnóstico tipo FB detecta la falla
  - ▶ Mensaje de error
  - ▶ Simbolo asociado a la falla
  - ▶ Fecha y hora de la aparición de desaparición de la falla
  - ▶ Fecha y hora de reconocimiento de la falla
- (2) Información suplementaria en el mensaje seleccionado
- Doble click en el mensaje para desplegar la falla en el programa

# ***Notas***