

# SoMachine

Manual de formación





# Contenido

---

## Capítulo -1 Antes de empezar

Antes de empezar .....	1-2
Descripción del curso .....	1-3

## Capítulo - 2 - SoMachine

Introducción al SoMachine .....	2-2
Necesidades del sistema .....	2-3
Características del SoMachine.....	2-4
Abir SoMachine .....	2-6
Pantalla de inicio .....	2-7
SoMachine Central .....	2-9
Flujo de desarrollo de un proyecto .....	2-102

## Capítulo - 3 - Gestión del Proyecto

Nuevo Proyecto.....	3-2
Crear un proyecto vacío .....	3-3
Crear un proyecto con el asistente.....	3-95
Crear un proyecto a partir de una plantilla .....	3-11
Abrir proyecto .....	3-11
Conectarse al autómeta .....	3-11

## Capítulo - 4 - Configuración Hardware

Ventana de Configuración.....	4-2
Añadir un equipo al proyecto.....	4-3
Añadir un módulo de expansión a un equipo .....	4-4
Configurar módulos de expansión.....	4-5

## Capítulo - 5 - Programación

Logic Builder - Ventana de Programa .....	5-8
Objetos de la pestaña de aplicación.....	5-14
Objetos de la pestaña de dispositivos .....	5-14
Ejercicio – Crear proyecto de Irrigación .....	5-14
Creación de un Programa POU .....	5-17
Tareas.....	5-19
Ejecución del programa.....	5-20
Configurador de tareas.....	5-211
Ejercicio - Configurar una tarea.....	5-244
Simulación PLC .....	5-2525
Lenguajes de programación de CoDeSys .....	5-277
Ejercicio – Programar un POU en FBD .....	5-30
Ejercicio – Convertir un programa FBD a IL ó LD .....	5-304
Watchdog .....	5-355
Estructura de una aplicación .....	5-366
POU tipo función .....	5-377
POU tipo bloque de función .....	5-38

## Capítulo - 6 - Variables

Tipos de datos estándar.....	6-2
Reglas para nombrar variables .....	6-3
Variables Locales .....	6-4
Declaración de variables .....	6-5
Variables globales.....	6-6
Declaración de variables .....	6-7
Configuración de símbolos.....	6-8
Tipos de datos del usuario (DUT) .....	6-8
Matrices.....	6-11
Variables con dirección física.....	6-12
Editor I/O .....	6-13

<b>Capítulo - 7 - Visualizaciones</b>	
Crear una Visualización .....	7-2
Variable de entrada.....	7-5
Reutilizar una Visualización .....	7-6
Ejercicio – Reutilizar una Visualización .....	7-68
<b>Capítulo - 8 - Librerías</b>	
Introducción a las Librerías .....	8-2
Agregar librería .....	8-2
Repositorio de bibliotecas .....	8-24
Creación de una librería de usuario .....	8-6
<b>Capítulo - 9 – Comunicación PLC</b>	
Establecer una comunicación con el PLC.....	9-2
Descarga múltiple.....	9-2
Source download .....	9-37
<b>Capítulo - 10 - Gestión del proyecto</b>	
Compilar programa .....	10-2
Aplicación de inicio.....	10-3
Cambios Online.....	10-4
Importar .....	11-25
Exportar .....	11-36
Gestion de ficheros de datos - Dataloging .....	11-47
<b>Capítulo - 11 – Herramientas de Depuración</b>	
Herramientas de depuración .....	10-2
Monitorización de variables.....	10-3
Editor de trazas .....	10-4
Puntos de interrupción y ejecución paso a paso .....	11-26
<b>Capítulo – 12 –Comunicación Modbus RTU</b>	
Ejercicio - Modbus I/O Scanner.....	12-2
Ejercicio – Mensajería Modbus .....	12-5
<b>Capítulo – 13 –CANOpen</b>	
CANopen Basics – Network & Object Directory .....	13-2
PDO's y SDO's .....	13-2
Añadir el maestro CANopen al controlador .....	13-2
Baudrate del bus CANopen .....	13-2
Añadir un esclavo bus CANopen.....	13-2
Configuración un esclavo en CANopen .....	13-2
Asignación de PDO's .....	13-28
Mapping de las variables de CANopen .....	13-2
Service Data Object (SDO) .....	13-2
Ejercicio – Lectura/escritura SDO's por programa .....	13-2
Ejercicio – Control Lexium32 en CanOpen con MFB's (Motion Funtion Blocks – PLCOpen) .....	13-2
Ejercicio – Inserción de Lexium32 en CanOpen con la plantilla de dispositivos.....	13-2
DTM's – FDT .....	13-2
Ejercicio – Inserción de Advantys OTB en CANOpen con DTM .....	13-2
<b>Capítulo - 14 - Ethernet</b>	
Introducción .....	14-2
Configuración del puerto Ethernet .....	14-3
Ejercicio – Mensajería Modbus TCP/IP .....	14-4
Ejercicio –Modbus TCP/IP – I/O Scanning .....	14-9
Gloval Network Variables .....	14-13
Servicios de Ethernet .....	14-17
Ejercicio – Webvisualization .....	14-19

## Capítulo - 14 – Añadir una HMI a la aplicación

Introducción .....	15-2
Añadir una HMI a la arquitectura.....	15-2
Configurador de símbolos .....	15-3
Añadir variables en el Vijeo Designer .....	15-4
Cambiar nombre del controlador en el SoMachine.....	15-5
Añadir nombre del controlador en el Vijeo Designer.....	15-7



# Capítulo 1: Antes de Empezar

## Descripción

**Este capítulo engloba estos temas:**

- Antes de empezar .....1-2
- Descripción del curso .....1-3

## Antes de Empezar

---

### Consideraciones previas

Este manual de aprendizaje es un suplemento para realizar la formación del software SoMachine. Dirigido al aprendizaje del uso del software.

Los gráficos y capturas de pantalla que se muestran en este manual se han realizado con el sistema operativo Windows XP usando el tema clásico de este. Si el usuario está usando el software con otro sistema operativo las ventanas pueden tener un aspecto diferente al que se muestra en este manual.

## Descripción del curso

### Objetivos

Después de realizar éste curso el alumno tendrá que:

- Ser capaz de crear un programa de PLC usando el software SoMachine.
  - Crear POU programa en todos los lenguajes de programación disponibles en el SoMachine.
  - Crear diferentes tareas y configurar la ejecución de los diferentes POU's en estas.
  - Programar un sistema simple de posicionado motion basado en Lexium 32 A ó M
  - Crear Librerías de usuario.
  - Crear estructuras de datos propias.
  - Aprender a utilizar las herramientas de depuración que dispone el SoMachine.
  - Configurar una comunicación Modbus por I/O scanning.
  - Configurar una comunicación Canopen.
  - Entender las diferentes opciones de conteo de las entradas de contador rápido.
  - Configurar y usar el generador de frecuencia en las salidas habilitadas para tal efecto.
  - Crear un programa de Vijeo Designer para una Magelis XBTGT conectado a un PLC M238/M258/M241/M251.
-



## Capítulo 2: SoMachine de un vistazo

### Descripción

#### Este capítulo engloba estos temas:

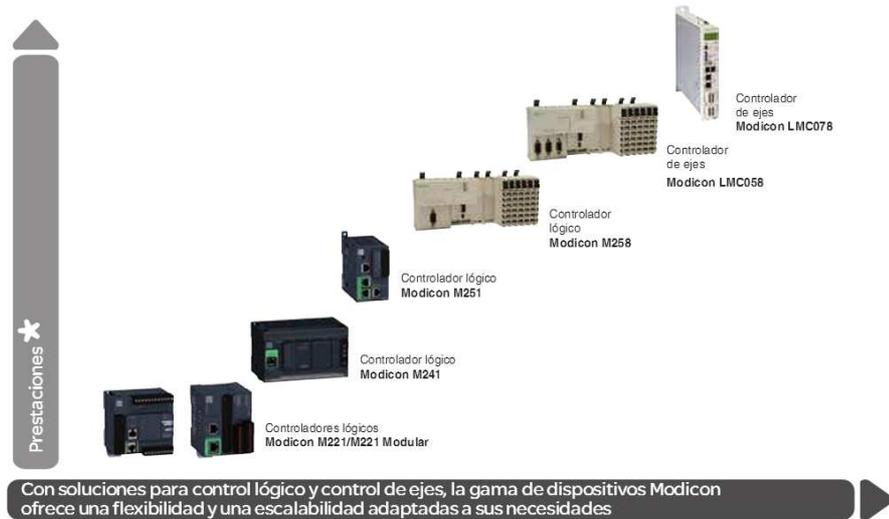
- Introducción al SoMachine .....2-2
- Necesidades del sistema .....2-3
- Características del SoMachine.....2-4
- Abir SoMachine .....2-6
- Pantalla de inicio .....2-7
- SoMachine Central .....2-9
- Flujo de desarrollo de un proyecto.....2-102
- Ventana de opciones del sistema .....2-15

# Introducción al SoMachine

## Solución Machinestruxure

La plataforma MachineStruxure es una solución completa para la automatización de maquinaria, que proporciona controladores escalables y flexibles, arquitecturas listas para usar, soluciones de ingeniería eficientes, así como servicios de personalización y asistencia técnica.

Gracias a la nueva gama 'NEXT Generation' de controladores lógicos Modicon TM, MachineStruxure proporciona un control de máquina escalable y flexible. Conectividad Ethernet, un puerto USB para la programación y servidor web incorporado.

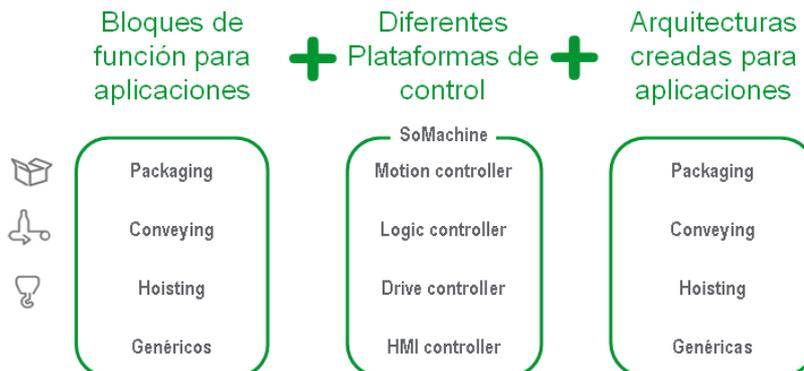


## Un único interface

- **Un sólo software**
  - Para los cuatro tipos de controladores (Lógicos, Motion, HMI y Driver)
- **Un download**
  - Transfiere toda la aplicación de la máquina con una sola descarga.
- **Una conexión**
  - Conecta el cable a un equipo y descarga toda la aplicación
- **Transparente**
  - Accede a todos los equipos comunicados de manera transparente.



## SoMachine para control flexible de una máquina.



## Necesidades del sistema.

### Antes de empezar

Antes de iniciar la instalación de SoMachine V4.1 en su PC se tienen que tener en cuenta las siguientes especificaciones técnicas:

### Necesidades de Hardware

Descripción	Especificaciones Mínimas	Recomendado
Procesador	Intel® Core™ 2 Duo* o equivalente	Intel® Core™ i7 o equivalente
Memoria RAM	3 GB	8 GB
Espacio libre en el Disco Duro	8 GB	15 GB
Drive	Lector de DVD	Lector de DVD
Monitor	Resolución: 1024 x 1024 pixel	Resolución:1680 x 1050 pixel
Periféricos	USB interface	USB interface
Web	Para registrarse vía Web se necesita acceso a Internet	Para registrarse vía Web se necesita acceso a Internet



**Nota:** Se podría ejecutar con CPU de menos rendimiento, pero con restricciones, así como el uso de máquinas virtuales.

### Necesidades Software

Este software se puede instalar en los sistemas operativos siguientes:

SoMachine Componente	Versiones de Sistemas operativos
Sistema operativo	SoMachine software soporta los siguientes sistemas operativos: <ul style="list-style-type: none"><li>➤ Microsoft Windows XP Professional</li><li>➤ Microsoft Windows 7 Professional Edition de 32 bits/64 bits</li></ul>



**Nota:** SoMachine está optimizado para un sistema operativo de 64 bits

# Características del SoMachine

## Lenguajes de programación

SoMachine incluye, como estándar, los 6 lenguajes IEC (International Electrotechnical Commission) que indican la normativa IEC 61131-3. Dependiendo de las necesidades de programación, la aplicación puede estar programada las diferentes POU con cualquiera de estos 6 lenguajes.

- Diagrama de funciones (Function Block Diagram, FBD)
- Grafcet (Sequential Functional Chart, SFC)
- Texto estructurado (Structured Text, ST)
- Listado de instrucciones (Instruction List, IL)
- Diagrama de contactos (Ladder, LD)
- Continuous Function Chart (CFC)

---

## Programación

- Creación de Bloques de función (FBs) por el usuario.
- Creación de Funciones por el usuario.
- Creación de estructura de datos (DUT's).
- Cambios On-line.
- Ventanas de supervisión de variables.
- Monitorización Graficas de variables (trace).
- Puntos de interrupción, ejecución instrucción por instrucción.
- Simulación.
- Ventanas de visualización.

---

## HMI

- Librerías gráficas que contienen más de 4000 objetos 2D y 3D.
- Objetos de dibujo (puntos, líneas, rectángulos, elipses, etc ...)
- Objetos preconfigurados (botón, interruptor, barra gráfica, etc ...)
- Recetas (32 grupos de 256 recetas con un máx. 1024 ingredientes)
- Tablas de Acciones
- Alarmas
- Imprimir
- Java scripts
- Archivos Multimedia: wav, png, jpg, emf, bmp
- Gráficas.

---

## Motion

- Equipos incluidos preconfigurados.
- Editor de CAM
- Trazas
- Visualizaciones asociadas a los MFB.

## **Servicios Globales**

- Seguridad
  - Documentación de proyecto.
  - Comparación de proyectos.
  - Compartir variables (Global Data)
  - Gestión de librerías.
- 

## **Buses Comunicación**

- Maestro:
    - CANopen
    - CANmotion
    - Modbus Serial Line
  - Conectividad:
    - Profibus-DP
    - Ethernet IP
    - Modbus TCP (Mensajería y en el M251 I/O Scanning)
- 

## **Librerías de Aplicación**

- General:
  - Bloques de función Motion de la librería PLCopen.
- Soluciones en función del tipo de máquina:
  - Bloques de función para Empaquetado.
  - Bloques de función para Transporte.
  - Bloques de función para Elevación.

# Abrir SoMachine

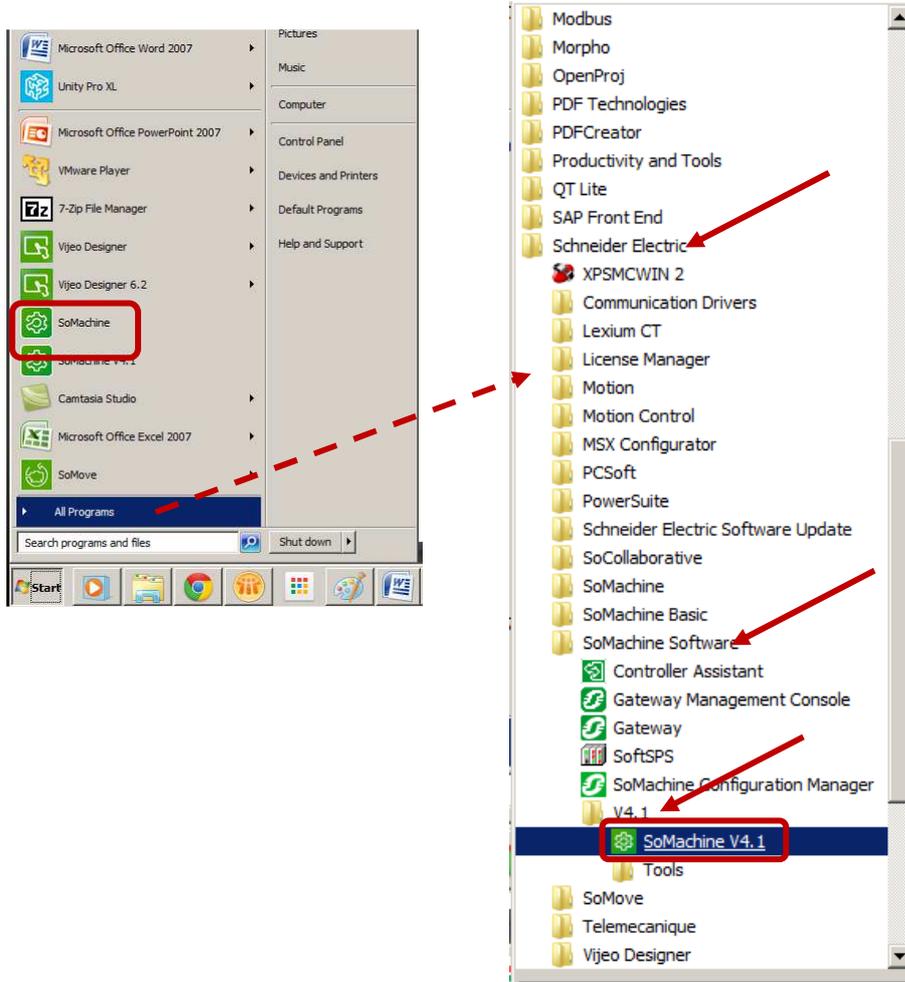
## Como abrir SoMachine

### ➤ Para abrir SoMachine:

Seleccionar el icono SoMachine en el menú de inicio de Windows:

**Inicio » Programas » Schneider Electric » SoMachine » SoMachine**

Hacer doble clic en el icono de SoMachine en el escritorio.



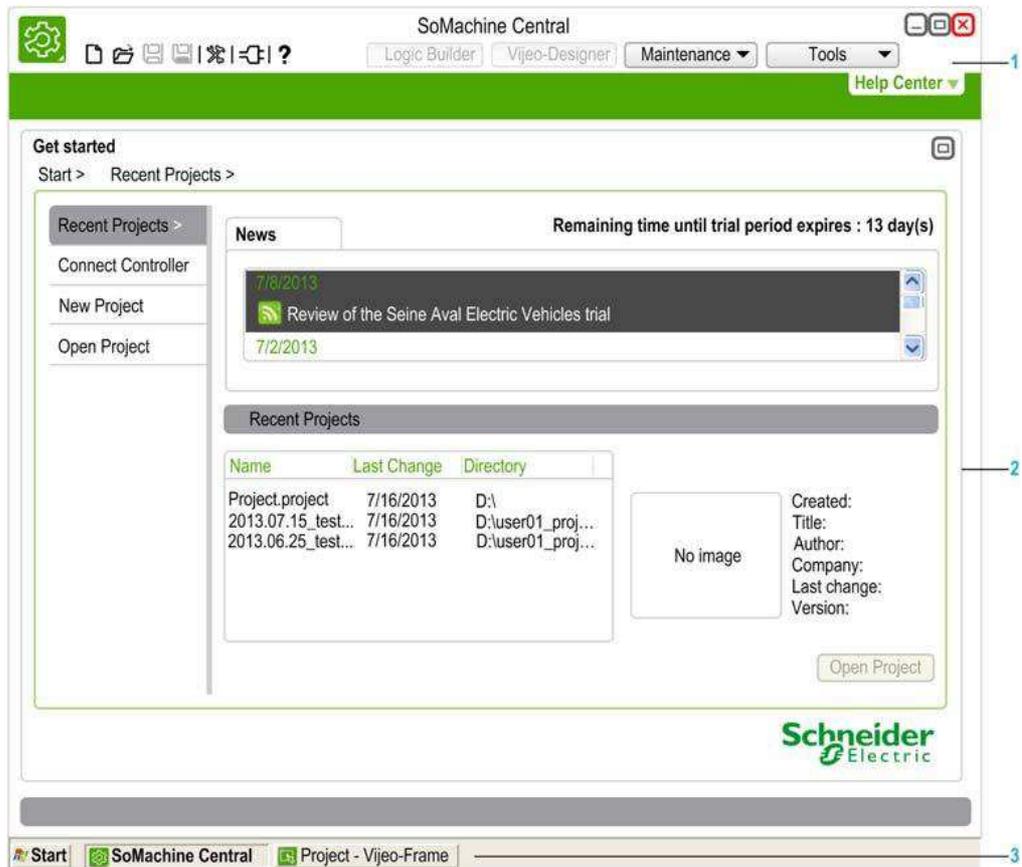
## Pantalla de inicio

### Interface de Usuario

La navegación dentro del SoMachine es intuitiva y muy visual. El interface de usuario está optimizado para que en los diferentes pasos del proyecto, se habiliten las herramientas necesarias para ese paso. El interface de usuario habilitará las opciones que se puedan realizar en cada paso. El espacio de trabajo ha sido optimizado, para sólo mostrar lo necesario en cada momento.

### Pantalla de inicio

La ventana de inicio, es la ventana que aparece una vez iniciado el SoMachine y en ella podremos seleccionar las siguientes opciones para abrir un proyecto existente o crear uno nuevo proyecto, por ejemplo.

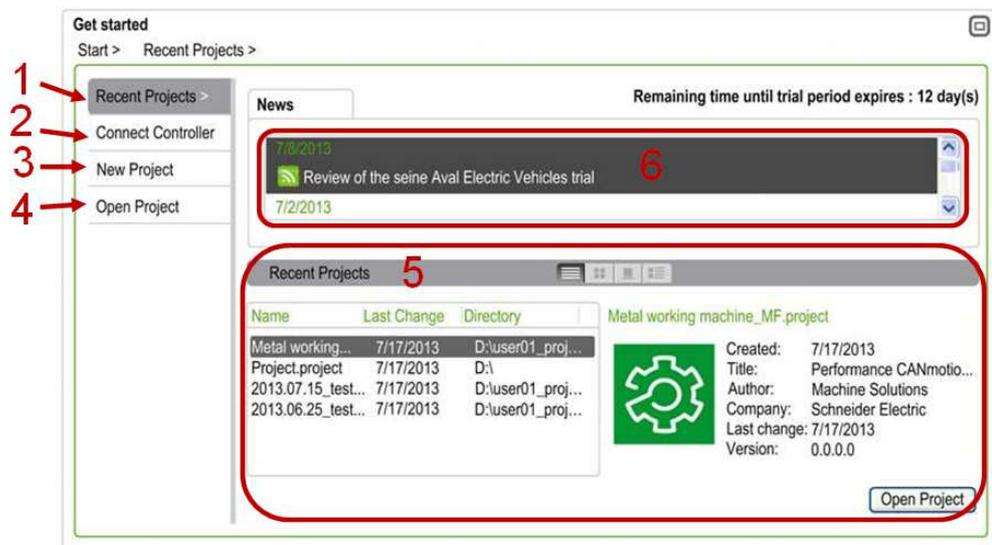


- 1.- Menú general
- 2.- Pantalla principal
- 3.- Barra de tareas del SO

## Opciones en la pantalla de inicio

Cuando se inicia SoMachine, se muestra la pantalla de inicio de SoMachine Central. Esta pantalla ofrece las siguientes funciones de proyecto:

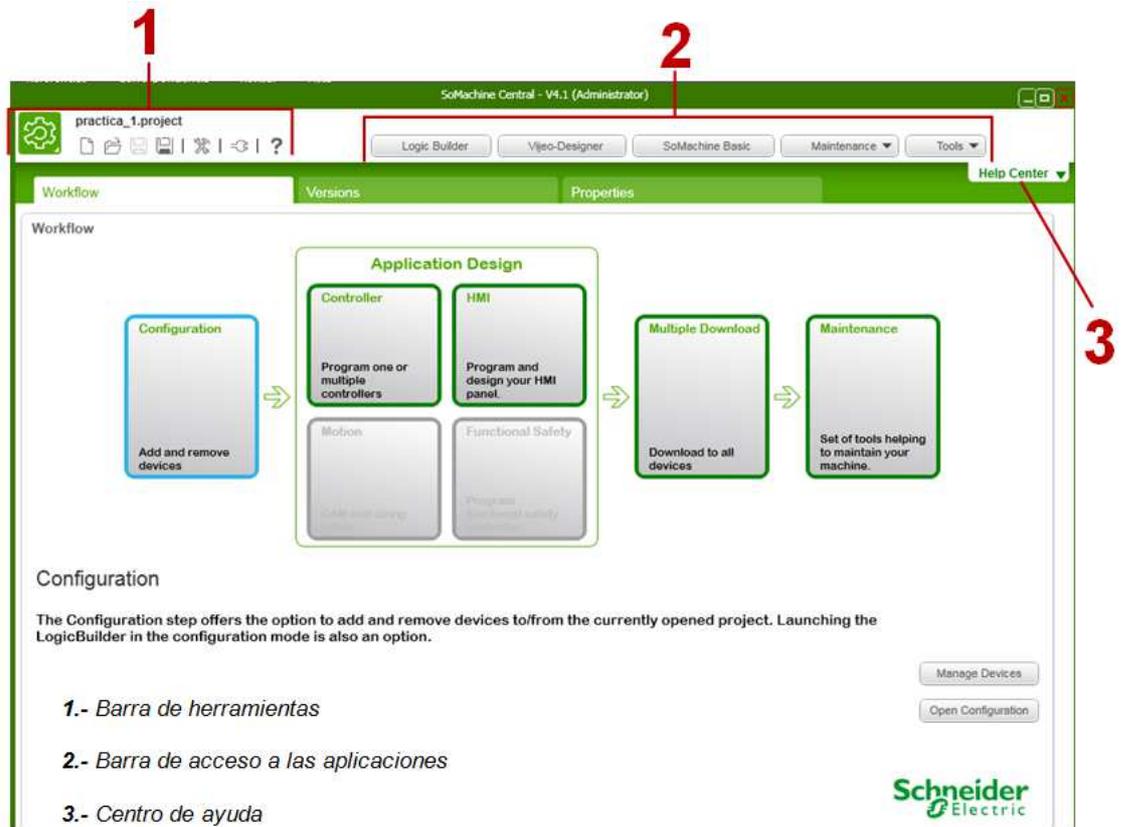
1. Abrir Proyectos recientes
2. Conectar con el controlador
3. Nuevo proyecto
4. Abrir proyecto
5. Lista de proyectos abiertos recientemente
6. Además, proporciona las últimas noticias de Schneider Electric.



# SoMachine Central

## Elementos del Somachine Central

Una vez creado el creado ó abierto cualquier tipo de proyecto el SoMachine nos muestra la ventana de SoMachine Central, que proporciona interface general con los elementos que figuran a continuación.



- 1.- Barra de herramientas
- 2.- Barra de acceso a las aplicaciones
- 3.- Centro de ayuda

## Barra de herramientas

La barra de herramientas es parte del menú general de la ventana del SoMachine Central.

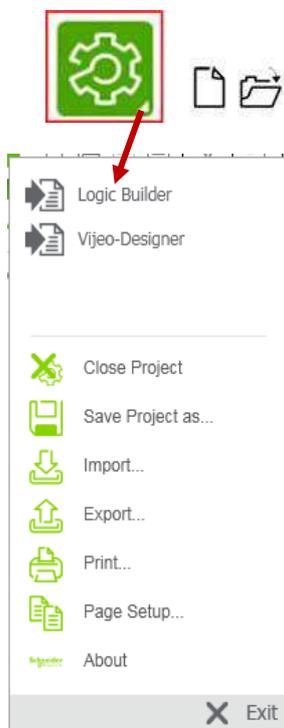
Cada icono de la barra de herramientas muestra la información sobre la acción que realiza dicha herramienta cuando el puntero del ratón se mueve sobre el icono correspondiente sin hacer clic.



Icono	Descripción
1	Abra el menú principal
2	Crear un nuevo proyecto.
3	Abra un proyecto existente.
4	Guarda el proyecto, que actualmente tiene el foco.
5	Guarda el proyecto, que actualmente tiene el foco, con un nuevo nombre.
6	Abra el cuadro de diálogo de opciones del sistema
7	Conectar con el controlador
8	Abra el Centro de ayuda

## Icono del menú general de funciones

El menú general ofrece funciones que se necesitan frecuentemente, para abrir el menú general, haga clic en el icono de la barra de herramientas SoMachine central.



1. Abrir el Logic Builder.
2. Abrir el Vijeo-Designer.
3. Cerrar proyecto
4. Guardar proyecto como ...
  - a. Guardar proyecto como..
  - b. Guardar proyecto como Biblioteca compilada
  - c. Guardar proyecto como plantilla ...
  - d. Guarda la librería y la coloca en el repositorio de bibliotecas
  - e. Guardar archivo ...
5. Importar ...
  - a. Importación Vijeo Designer Proyecto ...
  - b. Importar proyecto SoMachine básico ...
6. Exportar ...
  - a. Exportación de proyecto Vijeo Designer ...
  - b. Exportación de proyecto de SoMachine Basic ...
7. Imprimir ... (Define el contenido de la documentación)
8. Página de configuración
9. Acerca de
10. Salida

## Barra de acceso a herramientas

La barra de herramientas de acceso se muestra en la parte superior de SoMachine Central. Le permite cambiar a otras herramientas integradas para SoMachine.

Cada icono de la barra de herramientas de acceso muestra una punta de la herramienta cuando el puntero del ratón se mueve sobre el icono correspondiente sin hacer clic.

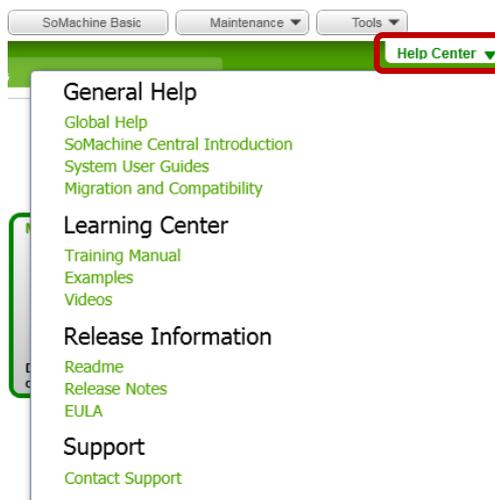


Botón	Descripción
Logic Builder	Abrir el interface de programación del SoMachine
Vijeo-Designer	Abrir el Vijeo Designer.
SoMachine Basic	Abrir el SoMachine Basic.
Mantenimiento	Seleccione una herramienta de mantenimiento (por ejemplo: Controller Assistant ó Configuración del OPC)
Herramientas	Abrir la herramienta seleccionada.

## Centro de Ayuda

Puede abrir el Centro de ayuda con el botón Centro de ayuda en la parte superior derecha de la SoMachine central o con el botón de la barra de herramientas o de la barra de información.

El Centro de Ayuda ofrece los siguientes temas:



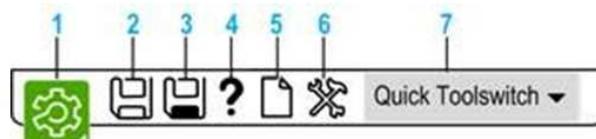
1. Ayuda General  
Ayuda Global (Guías de programación, y así sucesivamente)  
SoMachine Introducción Sistema Central  
Guías del usuario  
Migración y compatibilidad
2. Centro de Aprendizaje  
Manual de Entrenamiento E-Learning  
Videos  
Ejemplos (Ejemplos de proyectos con la documentación correspondiente)
3. Información de autorización
4. Léame
5. Notas de la EULA
6. Apoyo
7. Contacte con el soporte

## Barra flotante de funciones

Se proporciona una barra flotante de funciones para todos los softwares relacionadas con el proyecto (Logic Builder, Vijeo Designer ...etc). Esta ofrece las funciones básicas del SoMachine Central y proporciona una interacción rápida entre las diferentes herramientas relacionadas en el proyecto.

La barra flotante de funciones está siempre accesible a modo de pop-up en la parte superior izquierda de la pantalla en la barra de título de la herramienta, aunque se puede mover a lo largo de la barra de título con el ratón.

Cada icono de la barra flotante de funciones muestra la información sobre la función que realiza cuando el puntero del ratón se mueve sobre el icono correspondiente sin hacer clic.

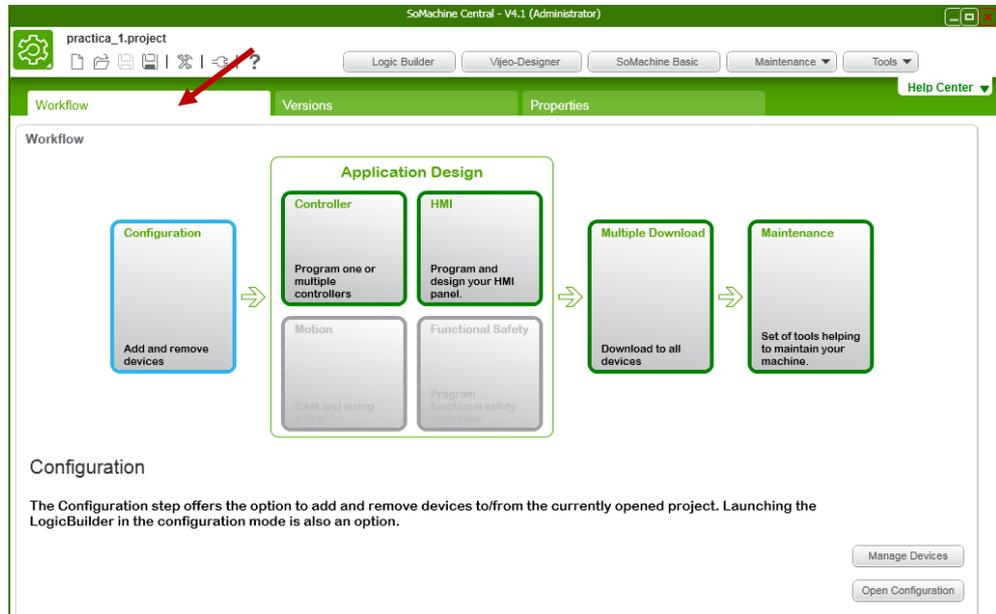


Elemento	Descripción
1	Volver al SoMachine central
2	Guardar el proyecto actual.
3	Guardar el proyecto actual, con un nuevo nombre.
4	Abrir la ayuda
5	Crear un nuevo proyecto.
6	Abra el cuadro de diálogo de opciones del sistema.
7	<b>Quick Toolswitch</b> Haga clic en la flecha a la derecha para cambiar a otra herramienta que esté activa (por ejemplo Vijeo- Designer).

# Flujo de desarrollo de un proyecto

## Pestaña de flujo de trabajo

Después de generar un nuevo proyecto o abrir un proyecto existente, se muestra la pantalla de flujo de trabajo. La pantalla muestra una representación gráfica de la gestión del flujo de trabajo del proyecto.



Para interactuar con el flujo de trabajo, haga clic en los diferentes pasos del flujo de trabajo, que están representados como botones. Si selecciona un paso del flujo de trabajo, la información detallada de este paso se muestra en la parte inferior de la pantalla.

Elemento	Descripción	
<b>Configuración</b>	En esta etapa, se puede añadir, eliminar y configurar los dispositivos y la comunicación.	
<b>Diseño de aplicaciones</b>	<b>Controlador</b>	En este paso, se puede programar el controlador (o varios controladores) de su proyecto. Se abrirá la herramienta de programación 'Logic Builder'.
	<b>HMI</b>	En este paso, se puede programar y diseñar la aplicación HMI. Haga clic en el botón, para iniciar el Vijeo-Designer.
	<b>Movimiento</b>	<i>*No disponible con esta versión de SoMachine.</i>
	<b>Seguridad</b>	<i>*No disponible con esta versión de SoMachine.</i>
<b>Descarga Múltiple</b>	En este paso, usted puede descargar el proyecto a los dispositivos.	
<b>Mantenimiento</b>	En este paso, se pueden realizar ciertas acciones de mantenimiento de los equipos de nuestro proyecto. Haga clic en este botón, para abrir la respectiva herramienta de mantenimiento (por ejemplo Controller Assistant, la configuración OPC).	
<b>Area de información</b>	Muestra información detallada del paso seleccionado en el flujo de trabajo.	

## Pestaña de flujo de trabajo

Esta pantalla muestra una lista de las versiones de su proyecto y ofrece las siguientes funciones de control de las versiones:

- *Bloquear / desbloquear una versión*
- *Eliminar una versión*
- *Restaurar una versión*
- *Guardar una nueva versión manualmente*
- *Parametrizar la creación de versiones de proyectos*

Para la versión de proyecto seleccionado en la lista, las siguientes propiedades se muestran en el lado derecho:

1. Título
2. Autor
3. Empresa Version
4. Imagen (Imagen definida por el usuario)
5. Estadísticas

practica\_1.project

SoMachine Central - V4.1 (Administrator)

Workflow Versions Properties Help Center

Logic Builder Vjeo-Designer SoMachine Basic Maintenance Tools

**Versions** Overview of the Central Versioning Service. The service can be setup to automatically or manually generated project versions for later restoral. Settings Autosave ON, 0h:30min:0sec Interval Settings

**Version Files**

Timestamp	Filesize	Version Comment	Lock/Unlock	Delete
02.05.2013 14:08:32	0,535		🔒	🗑️
02.05.2013 14:08:32	0,767		🔒	🗑️
02.05.2013 14:08:32	1,57	Version Comment <->	🔒	🗑️

**My Project 2013.04.09\_01 Version**  
Version: 0.0.4  
Created: 02.05.2013 13:56:43  
by: user01

- 1 Project Information
- 6 Library Manager
- 6 Task Configuration
- 6 Task
- 6 Application
- 29 Connector
- 53 Device
- 6 PLC Logic
- 6 Global Variable List

Version Comment  
Version Comment <-> Project Comment

Manual Version Restore

Schneider Electric

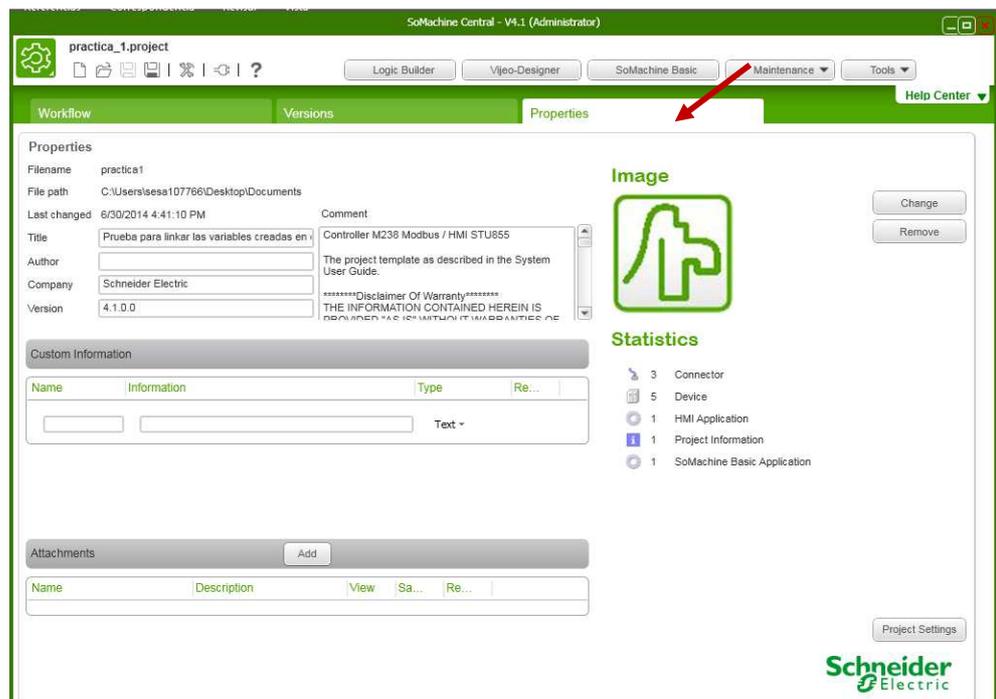
## Pestaña de Propiedades

Esta pantalla muestra las siguientes propiedades de su proyecto (de sólo lectura):

- Ruta de archivo
- Nombre del archivo
- Fecha/Hora de la última modificación

Además le ayuda a editar y guardar información adicional para sus proyectos (lectura / escritura):

- Título
- Autor
- Empresa Versión comentario
- Imagen (Imagen definida por el usuario)
- Información personalizada Adjuntos



## Ventana de opciones del sistema

### Acceder a las opciones del sistema

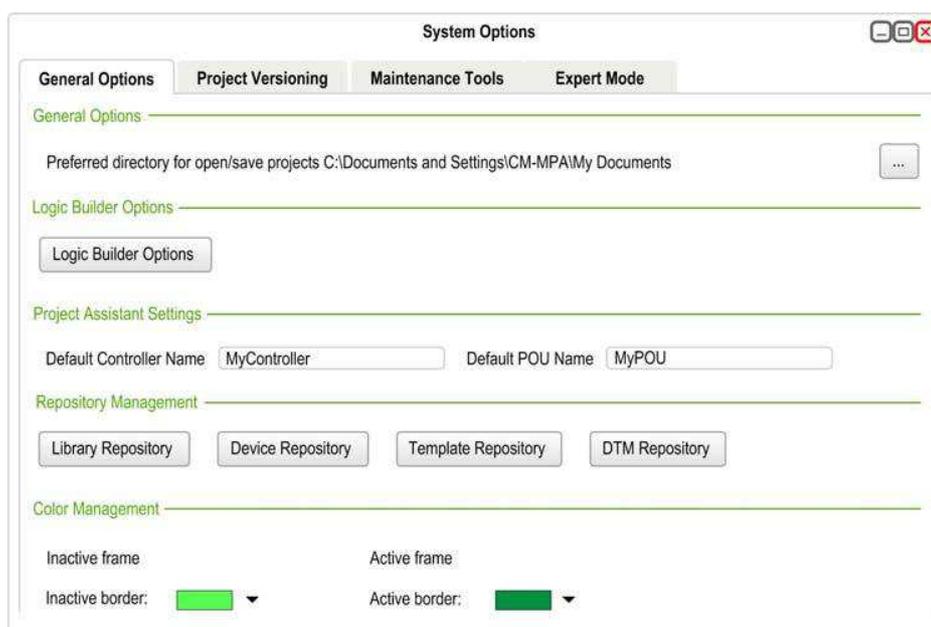
Dentro de la barra flotante del menú general del SoMachine central, podemos abrir las opciones de configuración del software, para adecuar el interface a nuestro gusto o necesidades.



En la venta de opciones del sistema se proporcionados 4 pestañas:

1. Opciones generales
2. Versiones del proyecto
3. Herramientas Mantenimiento
4. Modo Experto

#### • Pestaña de opciones generales:

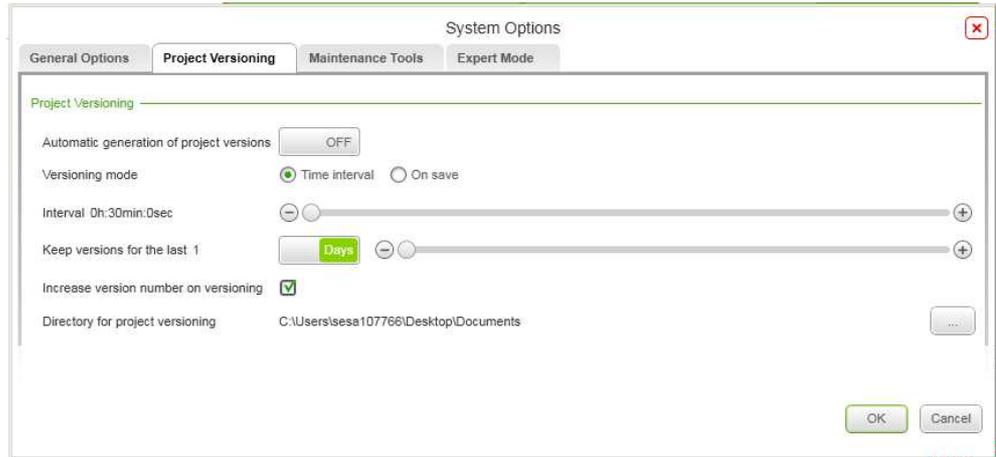


En la pestaña de opciones general de propiedades del software, se puede especificar la ruta donde se guardarán ó buscarán los proyectos existentes, configurar el nombre por defecto que tendrán el controlador y el POU, los colores de las ventanas así como a la ventna de opciones de la herramienta de programación 'Logic Builder' y la gestión tanto del repositorio de bibliotecas, de dispositivos de plantillas y de DTM.



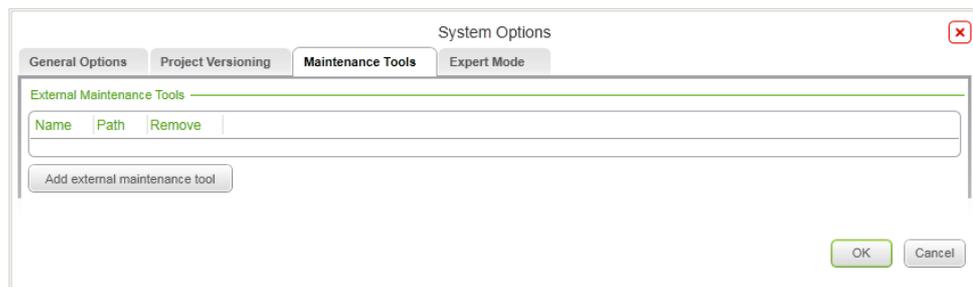
**Nota:** A la configuración de las opciones del Logic Builder como a los diferentes Repositorios, también se pueden acceder desde la misma herramienta 'Logic Builder'.

- Pestaña de versiones del proyecto:



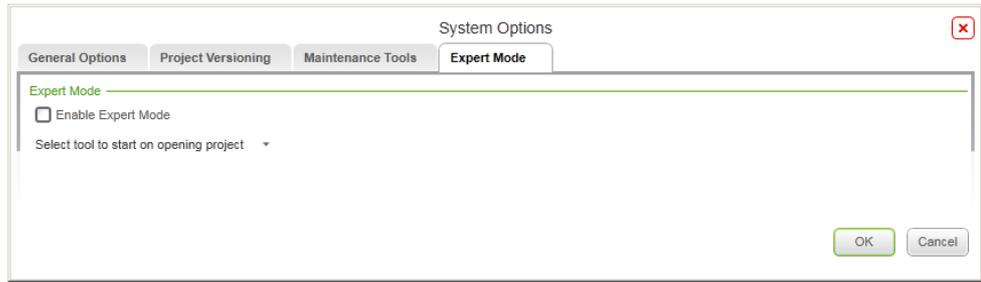
Elemento	Descripción
<b>Generación automática de versiones del proyecto</b>	Utilice el botón ON / OFF para activar la creación automática de versiones del proyecto (Esta función también está disponible en la pantalla de Versiones del Somachine Central).
<b>Método de creación de versiones</b>	Seleccionar uno de los dos modos: <ul style="list-style-type: none"> <li>Intervalo de tiempo</li> <li>Al guardar</li> </ul>
<b>Intervalo de tiempo</b>	Define el patrón de tiempo de guardado automático proyecto (30 minutos a 24 horas). Cada vez que el proyecto se guarda de forma automática, una nueva versión del proyecto es creado.
<b>Mantenimiento de versiones para los últimos (Días/Meses)</b>	Define el periodo durante cuánto tiempo se almacenan las versiones del proyecto (1 ... 31 días / 1 ... 12 meses).
<b>Aumento del número de la versión</b>	Active esta casilla para aumentar el número de versión del proyecto de cada vez que se crea una nueva versión.
<b>Directorio para guardar las versiones del proyecto</b>	Haga clic en el botón '...' para abrir un explorador de Windows y para buscar un directorio para almacenar las versiones del proyecto.

- Pestaña de Herramientas de mantenimiento:



Elemento	Descripción
<b>Botón de Añadir Herramienta de mantenimiento externo</b>	Abre un explorador de Windows, que te permite navegar para añadir una herramienta de mantenimiento externa (*.exe).
Columna <b>Nombre</b>	Nombre de las herramientas de mantenimiento externas añadidas.
Columna <b>Ruta</b>	Ruta de las herramientas de mantenimiento externas añadidas.
Columna <b>Eliminar</b>	Haga clic en el icono de la papelera para que la herramienta añadida sea eliminada.

- **Pestaña de Modo experto:**



En esta pestaña habilitamos el modo experto, Active esta casilla de verificación para iniciar la herramienta seleccionada de la lista de abajo tan pronto cuando se abre un proyecto existente.



## Capítulo 3: Gestión del proyecto

### Descripción

**Introducción** El software SoMachine permite al usuario realizar las principales acciones de gestión como: crear, exportar e importar proyectos. El SoMachine también dispone de herramientas para archivar y comprimir el archivo del proyecto.

El usuario puede seleccionar diferentes opciones a la hora de crear un proyecto Nuevo: Usando un proyecto en blanco, Usando una arquitectura TVDA (Tested Validated and Documented Architectures), Usando un proyecto ejemplo, usando un proyecto existente y usando una aplicación.

#### Este capítulo engloba estos temas:

- Nuevo Proyecto.....3-2
- Crear un proyecto vacío .....3-3
- Crear un proyecto con el asistente.....3-95
- Crear un proyecto a partir de una plantilla .....3-11
- Abrir proyecto .....3-11
- Conectarse al autómata .....3-11

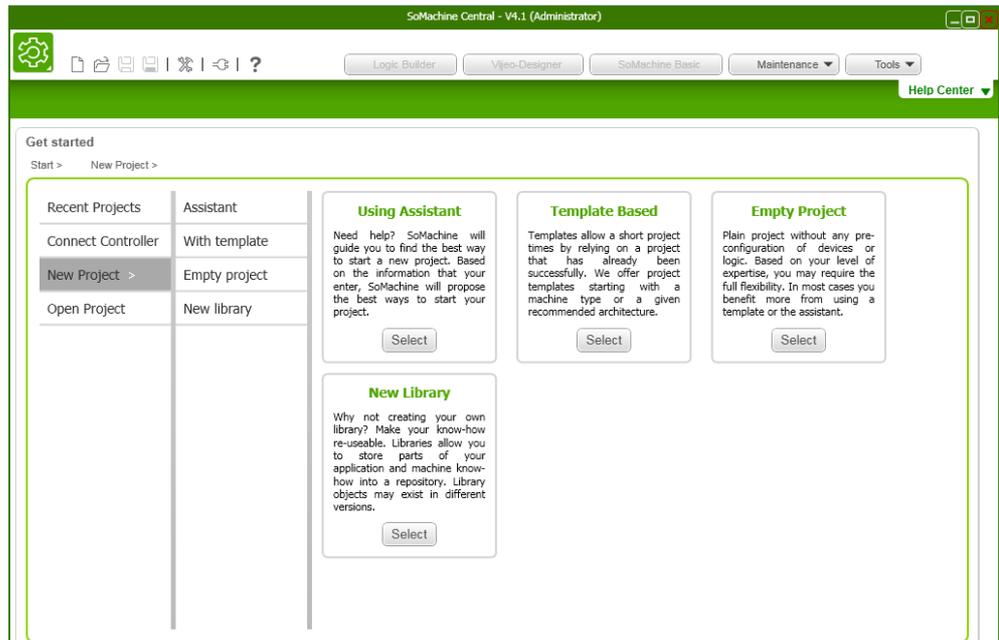
# Nuevo Proyecto

## Primer paso configuración

El primer paso que hay que realizar en el SoMachine es la de crear un nuevo proyecto. El proyecto es el archivo donde toda la información relacionada con este quedará guardada en formato .pro.

## Crear un Proyecto Nuevo

SoMachine tiene herramientas que ayudan al usuario a crear un Nuevo proyecto de manera rápida y sencilla.



Este tiene para iniciar un proyecto las siguientes opciones:

- **Usando el asistente:** Usando esta opción crearemos el proyecto nuevo a través de las opciones que aparece una ventana de asistente que nos guía a realizar todos los pasos a la hora de crear el proyecto.
- **A partir de una plantilla:** Seleccionando una plantilla, entre una variedad de arquitecturas implementadas. Un buscador de arquitecturas TVD (Testeadas Validadas y Documentadas) asesora al usuario a seleccionar una implementación más adecuada a su proyecto y que se adapte a su proyecto.
- **Un proyecto vacío:** Es la opción más básica, donde tienes que implementar el proyecto desde cero.
- **Crear una nueva librería:** Cuando el usuario en vez de crear un proyecto, crea un librería de usuario donde realiza FB's o POU que más tarde reutilizará en varios proyectos a la vez.

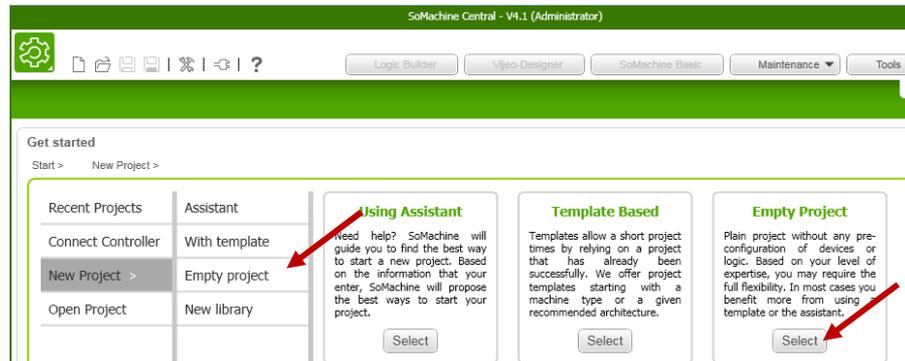
# Crear un proyecto vacío

## Como crear un proyecto vacío

### ➤ Para crear un proyecto vacío:

Cuando se crea un **‘Nuevo Proyecto vacío’** se crea un proyecto simple, sin preconfiguración de dispositivos, ni lógica. Sobre la base de su nivel de experiencia, esta opción le ofrece total flexibilidad.

En la ventana de inicio, hacer clic en **‘New Project > Empty project’** ó pulsar el botón de **‘Select’** que hay en el marco **‘Empty Project’**.

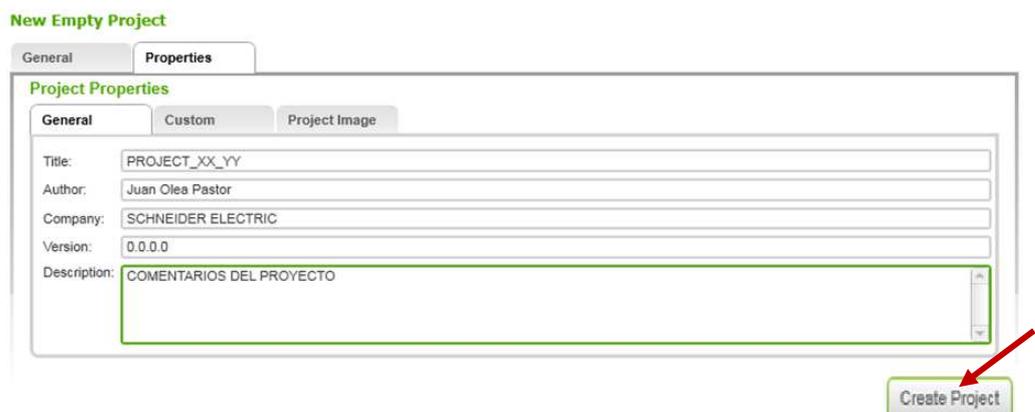


En el área de trabajo, aparecen dos pestañas, en la pestaña **‘General’** escribiremos el Nombre de nuestro proyecto en el campo **‘Project Name’**.

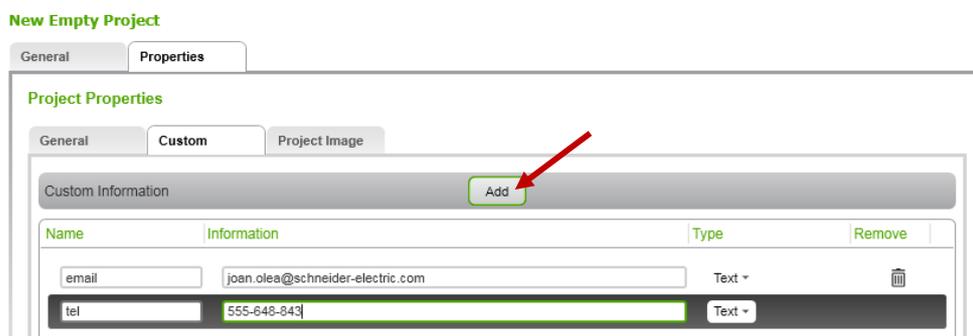


En la pestaña **‘Properties’** especificaremos información relacionada con el **proyecto**. Cuando abrimos esta pestaña nos aparecen tres subpestañas que donde que agrupa diferente tipo de información.

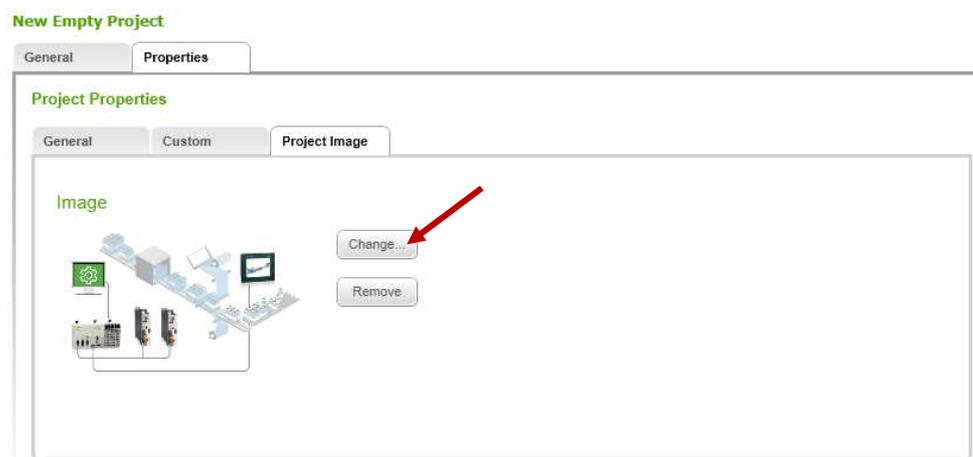
En la subpestaña **‘General’** añadiremos los datos generales del proyecto.



En la subpestaña '**Custom**' añadiremos otros datos de interés para nuestro proyecto (Ej. Persona de contacto, teléfonos), apretando el botón '**Add**' se añadirá un campo, seleccionado el tipo de campo en la columna '**Type**'.



En la subpestaña '**Project Image**' podemos asociar una imagen a nuestro proyecto para ayudarnos a identificarlo mejor.



Una vez insertada toda la información, finalizaremos pulsando el botón '**Create Project**' que se encuentra en la parte inferior derecha.



**Nota:** Las información del proyecto es en todo momento accesible y modifiable desde la pestaña '**Propiedades**' de la pantalla principal del SoMachine Central.

# Crear un proyecto con el asistente

## Como crear un proyecto con el asistente

### ➤ Para crear un proyecto con la ayuda del asistente:

El asistente le ayuda a crear un proyecto que ya contiene un controlador y una POU ya asociado a la tarea principal MAST.

En la pantalla de inicio de SoMachine Central, seleccionar 'New Project' y luego 'Assistant'.

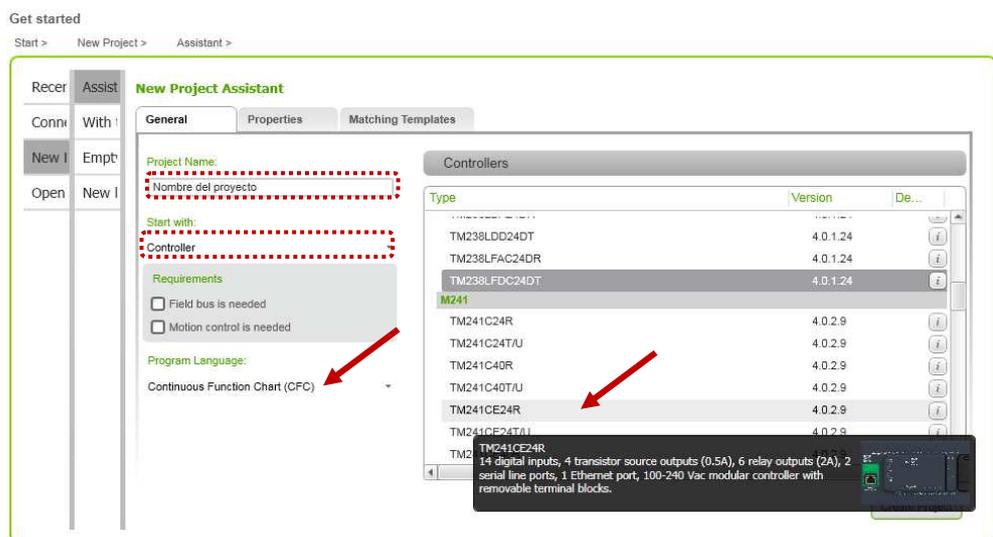


En el área de trabajo, aparecen tres pestañas, en las que nos ayudará a seleccionar lo necesario para crear un proyecto de una manera sencilla.

En la pestaña 'General' especificaremos en nombre del proyecto, el controlador que vamos a utilizar y el lenguaje de programación del primer POU (que ya aparecerá asociado directamente a la tarea MAST).

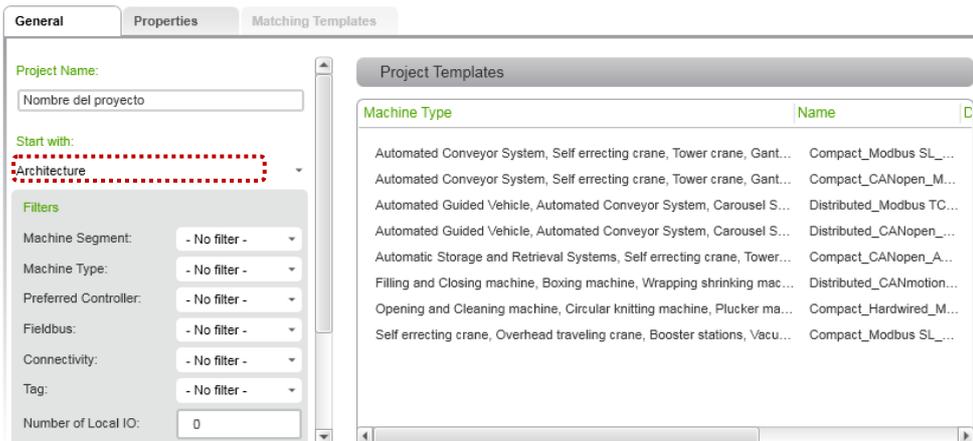
Según lo que se seleccione en el campo 'Start with' aparecerán unos filtros 'Requirements' u otros para reducir la selección.

- **Controlador:** El área de información en la parte derecha muestra la disposición controladores ordenados y agrupados por tipo.



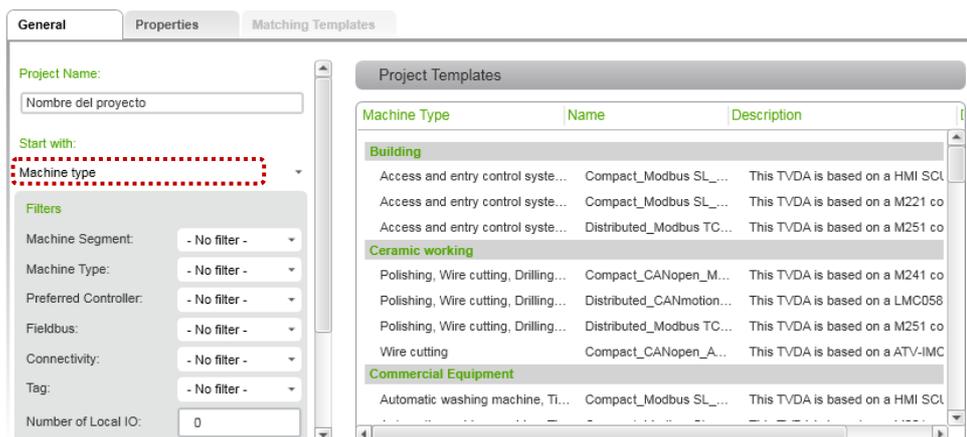
- **Arquitectura:** El área de información en la parte derecha muestra la disposición plantillas de proyectos ordenados por nombre.

#### New Project Assistant



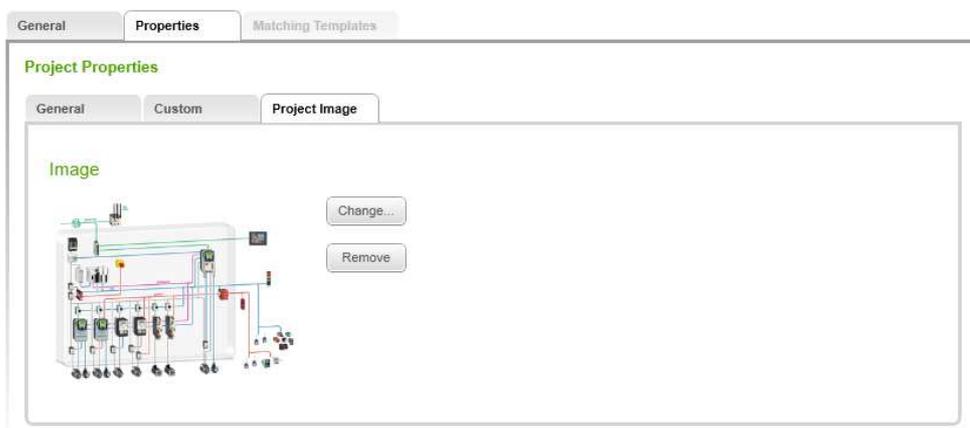
- **Tipo de Máquina:** El área de información en la parte derecha muestra la disposición de las plantillas de proyectos agrupados por tipos de máquinas (transporte, de elevación, y así sucesivamente)

#### New Project Assistant



En la pestaña **'Properties'** añadiremos la información general del proyecto como, el autor, la empresa, imagen del proyecto ...etc.

#### New Project Assistant



En la pestaña **'Matching templates'** se abre una lista previamente filtrada de las plantillas de proyecto que están usando el controlador seleccionado, por si queremos asociar dicha plantilla al proyecto.

#### New Project Assistant

Machine Type	Name	Description	Details
<b>Ceramic working</b>			
Polishing, Wire cutting, Drilling...	Compact_CANopen_M...	This TVDA is based on a M241 co...	
<b>Hoisting</b>			
Overhead crane	Hoisting_Industrial Cra...	Industrial crane with three axes co...	
Overhead crane	Hoisting_Industrial Cra...	Industrial crane with three axes co...	
Self erecting crane, Tower cra...	Compact_CANopen_M...	This TVDA is based on a M241 co...	
<b>HVAC and R</b>			
Ceiling Heating unit	Compact_CANopen_M...	This TVDA is based on a M241 co...	
<b>Lift</b>			
Elevator, Escalator	Compact_CANopen_M...	This TVDA is based on a M241 co...	
<b>Material Handling</b>			
Automated Conveyor System	Compact_CANopen_M...	This TVDA is based on a M241 co...	
<b>Metal working</b>			

## Crear un proyecto a partir de una plantilla

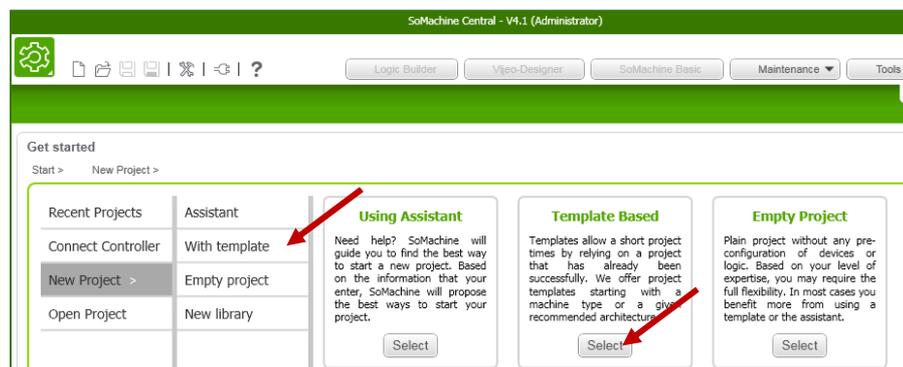
¿Qué es una arquitectura TVDA?

Las arquitecturas **TVDA**, permite al usuario crear un nuevo proyecto basándose en una plantilla TVDA (*Tested Validated Documented Architecture*). SoMachine incluye diferentes proyectos TVDA, listos para ser usados, sólo adaptando las configuraciones y programación ya creados previamente.

Como crear un proyecto a partir de una plantilla

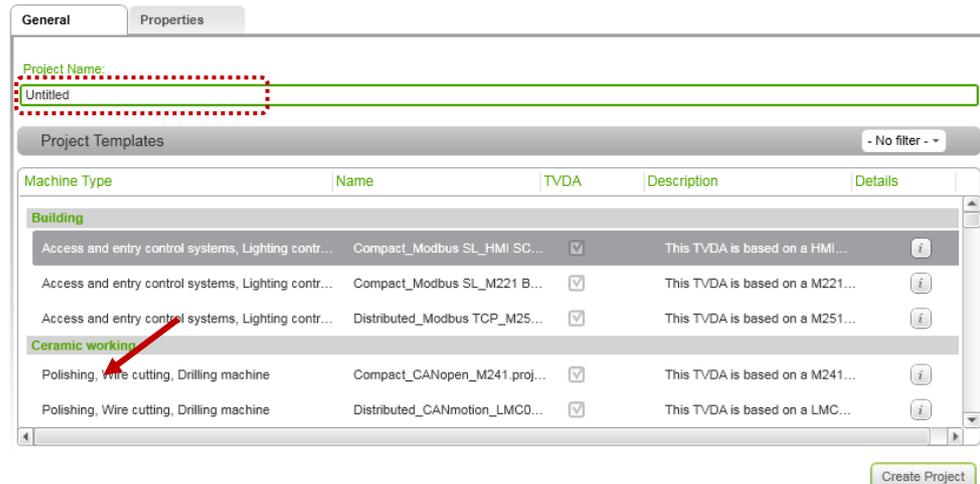
➤ **Para crear un proyecto a partir de una plantilla:**

En la ventana de inicio, hacer clic en **'New Project > With template'** ó pulsar el botón de **'Select'** que hay en el marco **'Templated Based'**.



En el área de trabajo, aparecen dos pestañas, en la pestaña **'General'** escribiremos el Nombre de nuestro proyecto y elegiremos la plantilla que más se ajusta a nuestro proyecto. También tenemos la posibilidad de filtrar los resultados por arquitecturas, aplicaciones o definida previamente por el usuario.

### New Project Assistant - Templates



En la pestaña **'Properties'** especificaremos información relacionada con el **proyecto**. Cuando abrimos esta pestaña nos aparecen tres subpestañas que nos agrupa diferente tipo de información.

## Abrir proyecto

### Como abrir un proyecto existente

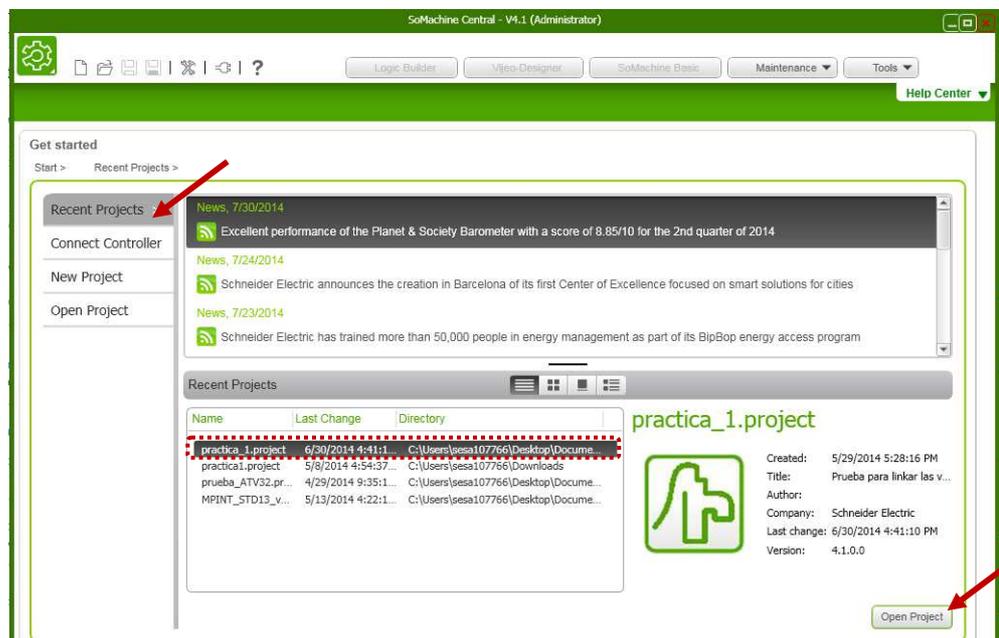
Al abrir el SoMachine, aparece la página de inicio que por defecto muestra en el área de trabajo la opción **'Recent Projects'** donde se muestran los últimos proyectos que han sido abiertos para su acceso rápido.

#### ➤ Para abrir un proyecto existente:

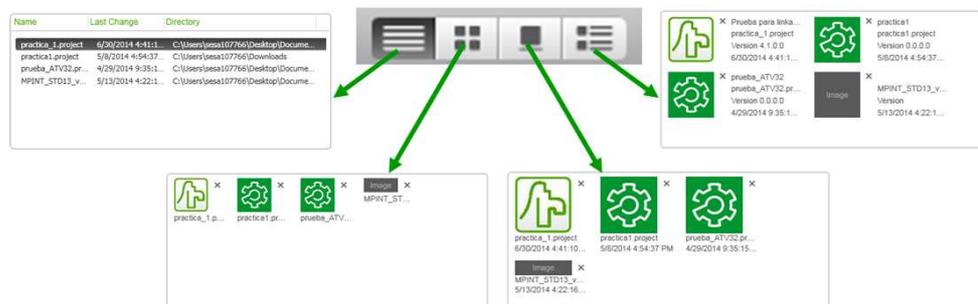
Hacer doble-clic en el icono del proyecto que aparece en el área de trabajo.

Ó

Seleccionar el proyecto en el área de trabajo y hacer clic en el botón de **'Open project'** que se encuentra en el área de **'Información del proyecto'**.



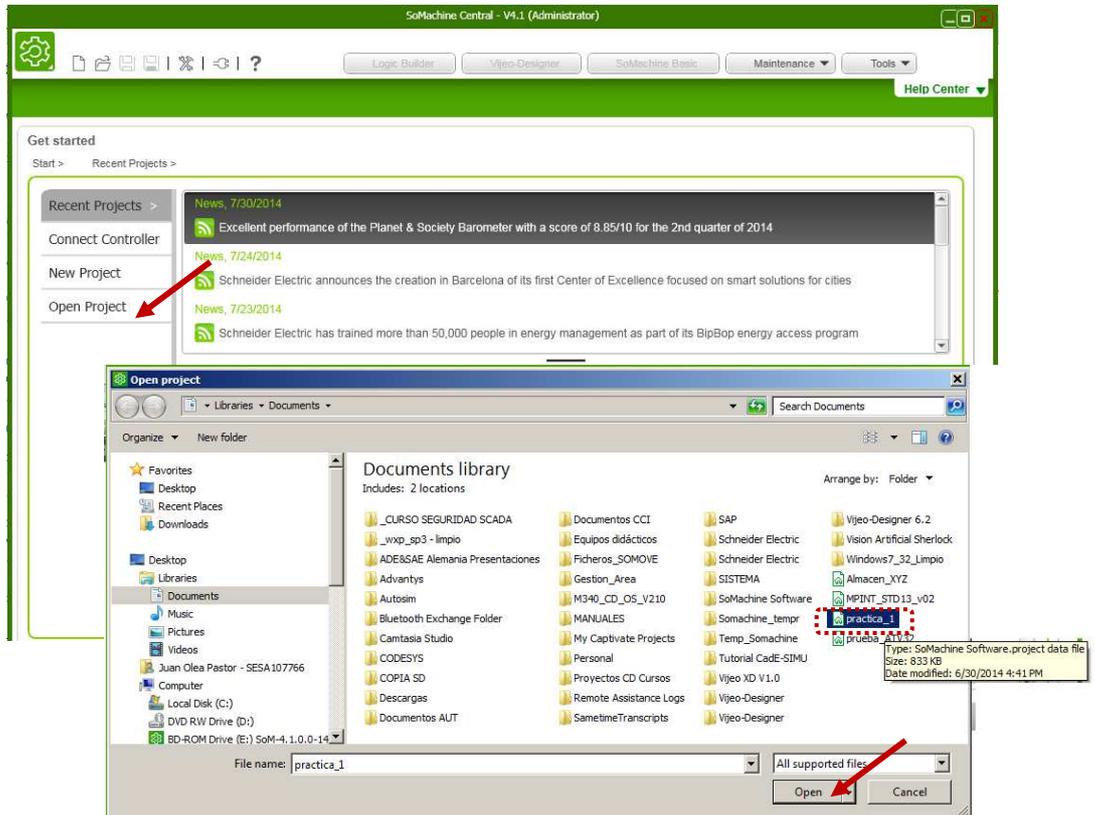
Existen diferentes formatos de visualización de los archivos:



Como buscar un proyecto ya creado.

➤ Para buscar un proyecto ya creado:

Hacer Clic en la opción 'Open Project' y luego en el explorador de Windows buscar la ruta donde se encuentra el archivo de proyecto que se desea abrir.



Buscar en la carpeta donde se encuentra el proyecto guardado utilizando el buscador 'Unidades Locales' que aparece en la parte izquierda de la ventana. Si la carpeta seleccionada contiene algún proyecto de SoMachine los iconos de estos aparecerán en el área de trabajo, donde para abrirlo se hará doble-clic sobre el icono.

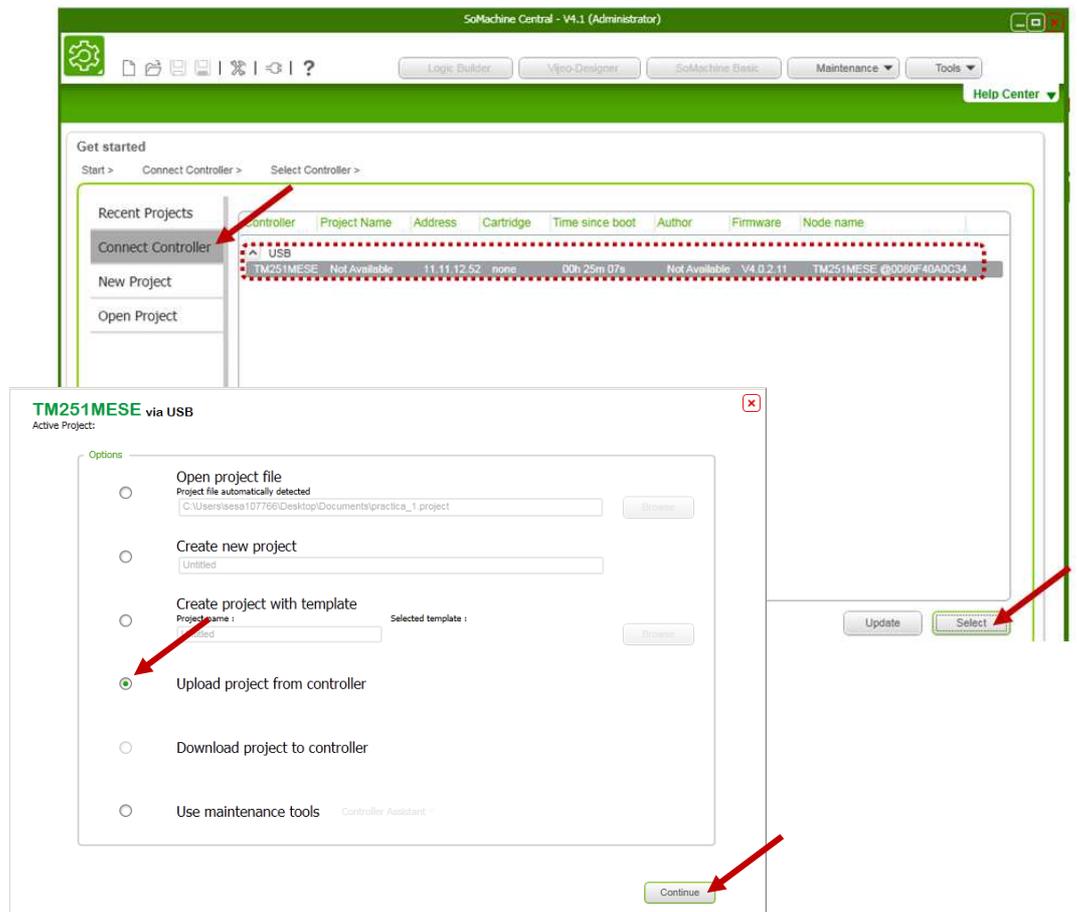
## Conectarse al autómata

### Connect Controller

La opción 'Connect Controller' nos permite escanear por Ethernet y Modbus los controladores conectados y poder conectarnos directamente a ellos y realizar tareas de mantenimiento como (Uploads del proyecto, Descarga de proyecto, firmware ..etc).

Cuando se inicia la página de inicio de SoMachine Central seleccionaremos la función '**Connect Controller**'.

Paso	Acción	Comentario
1	Haga clic en el botón Connect Controller.	Aparecerá el cuadro de diálogo Seleccionar controlador.
2	La red Ethernet y los puertos USB están escaneando los controladores disponibles.	La lista visualizada de controladores contiene los controladores de la red que han enviado una respuesta a la solicitud de SoMachine.
3	Seleccione un controlador de la lista y haga clic en el <b>Seleccione</b> botón.	Aparecerá el cuadro de diálogo Opciones.
4	Seleccione una de las opciones: <b>Abrir Archivo de Proyecto</b> (Examinar) <b>Crear un proyecto nuevo</b> <b>Crear un proyecto con la plantilla</b> <b>Upload del Proyecto del Controlador</b> <b>Descarga proyecto actual al Controlador</b> <b>Utilizar herramientas de mantenimiento</b>	-
5	Haga clic en el botón Continuar.	La opción seleccionada se ejecuta.





# Capítulo 4: Configuración Hardware

## Descripción

**Introducción** La ventana de configuración sólo aparece una vez que el proyecto ha sido creado.

El editor gráfico de configuración hardware tiene las funciones necesarias para realizar de una manera rápida y sencilla la configuración hardware de la arquitectura de control.

Los parámetros de configuración también se pueden realizar desde la ventana de programación o desde el Vijeo-Designer.

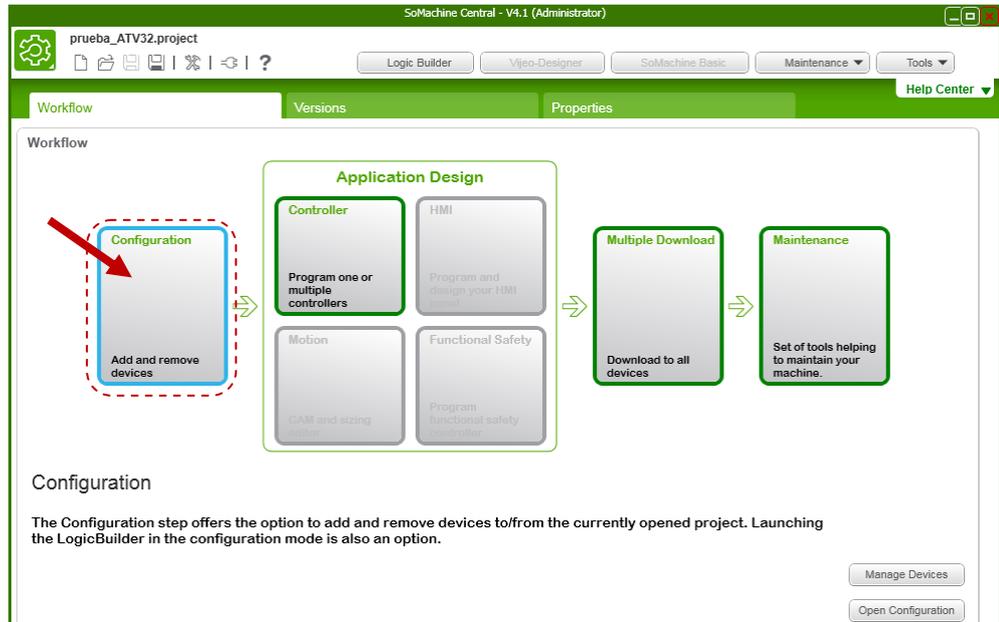
### **Este capítulo engloba estos temas:**

- Ventana de Configuración.....4-2
- Añadir un equipo al proyecto .....4-3
- Añadir un módulo de expansión a un equipo .....4-4
- Configurar módulos de expansión.....4-5

## Ventana de Configuración

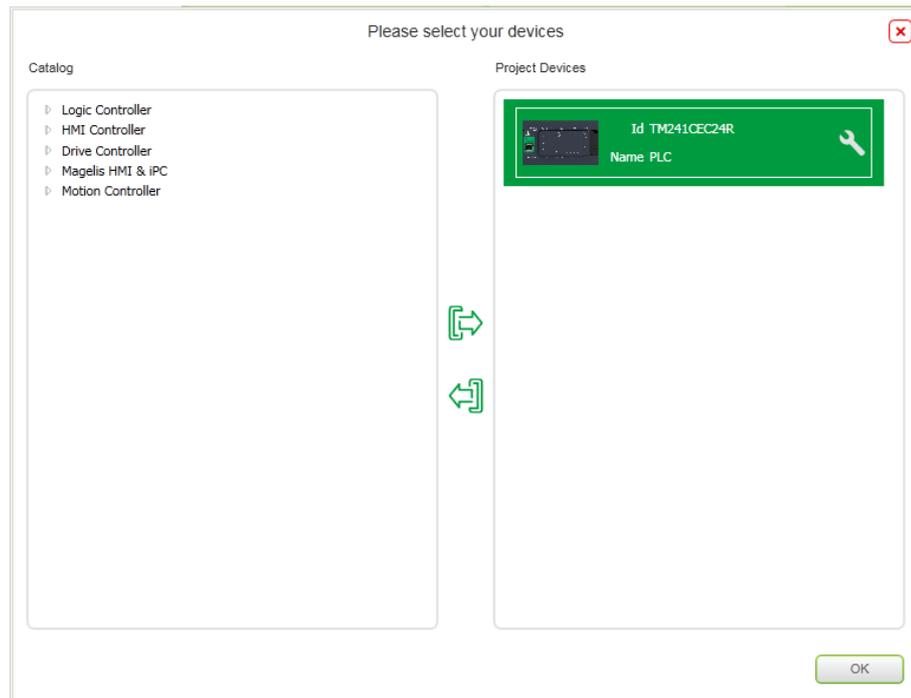
### Abrir ventana de configuración

Para abrir la ventana de configuración hardware, desde la ventana general del SoMachine Central, seleccionaremos la pestaña de **'Workflow'** (flujo de trabajo). Dentro de los bloques del flujo de trabajo de un proyecto seleccionamos el primer bloque **'Configuration'**.



### Ventana de configuración

En la ventana de configuración de hardware aparece dos áreas, el área de la izquierda aparece el catálogo de dispositivos que se puede añadir al proyecto, en el área de la derecha aparecen todos los dispositivos que ya han sido añadidos al proyecto.



Dentro del área de catálogo, los diferentes dispositivos se agrupan en las siguientes familias:

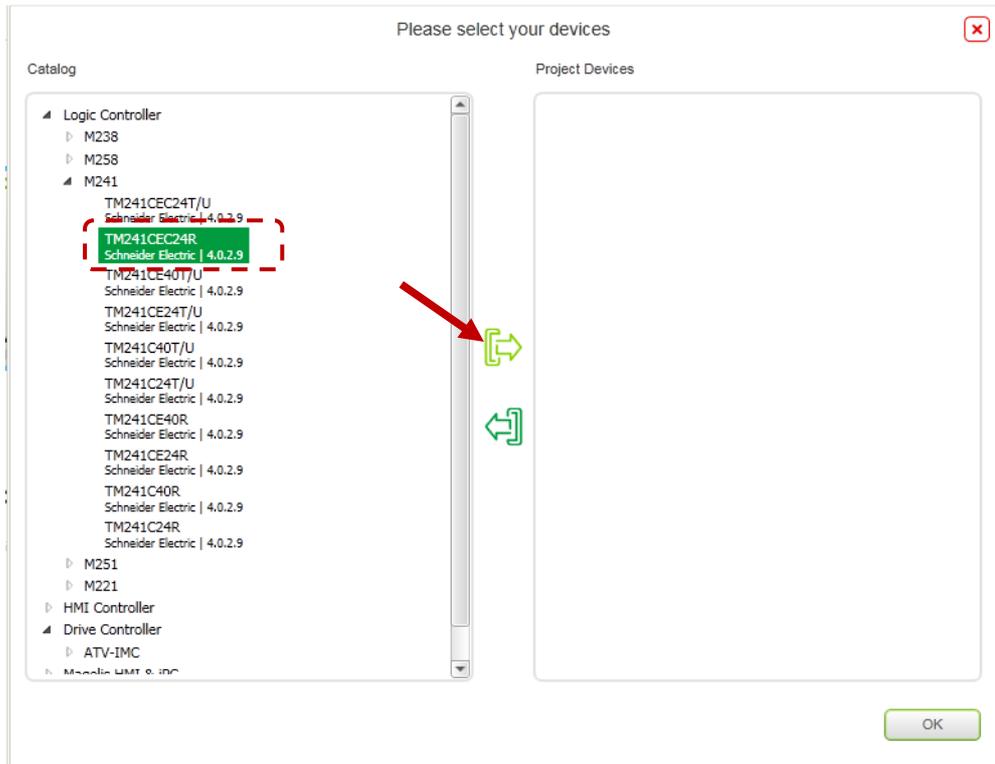
- *Logic Controller*: Controladores lógicos estándar (**M238, M258, M241, M251, M221**)
- *HMI Controller*: Esta familia engloba a las pantallas HMI que disponen de control para ser programado también (**XBTGC, HMISCU, XBTGK y todos los modelos de XBTGT con control**)
- *Driver Controller*: Controlador que se incorpora a un variador de frecuencia como elemento principal de la arquitectura de control, por ejemplo para las aplicaciones de Hoisting (**ATV-IMC**)
- *Magelis HMI & iPC*: Dispositivos HMI e IPC, todas aquellas pantallas HMI de la familia Magelis sin control.
- *Motion Controller*: Controladores que disponen de Canopen sincronizado para poder interpolar ejes (**LMC058**)

## Añadir un equipo al proyecto

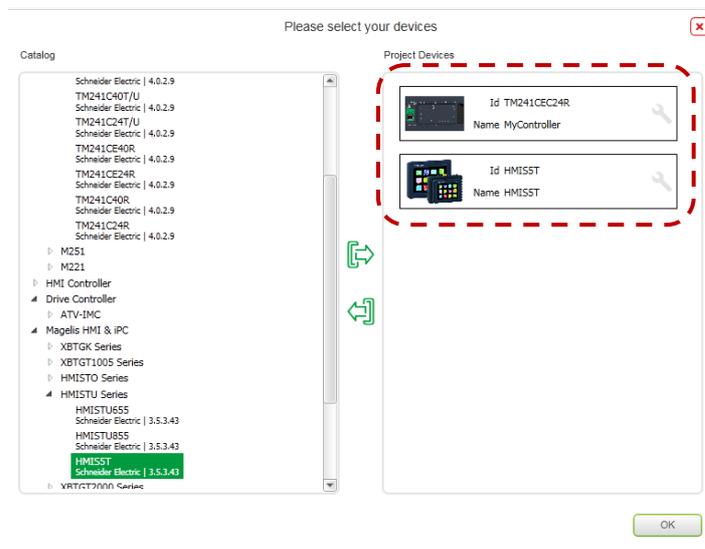
### Como añadir un equipo

#### ➤ Para añadir un equipo:

En el área de la izquierda '**Catalog**', seleccionar el equipo que se desea añadir al proyecto, una vez seleccionado aparecen en el centro los iconos de una flechas para mover los equipos de un área a otra. Pulsar la flecha que indica hacia la derecha y el equipo seleccionado aparecerá en el área de la derecha y ya estará incluido en el proyecto.



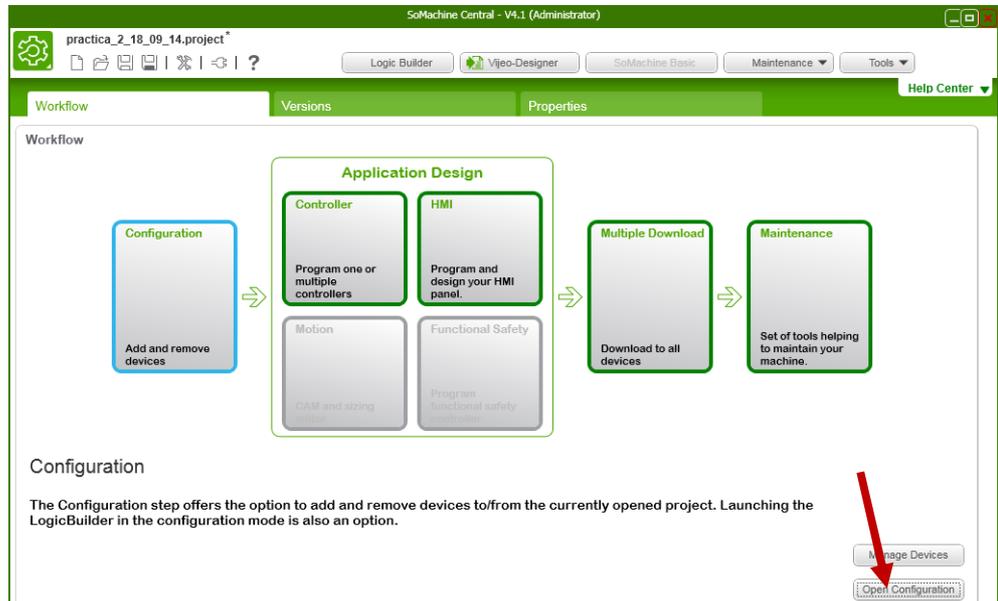
Cuando el equipo se haya añadido, aparecerá su representación gráfica en el área de la derecha '**Project Devices**'.



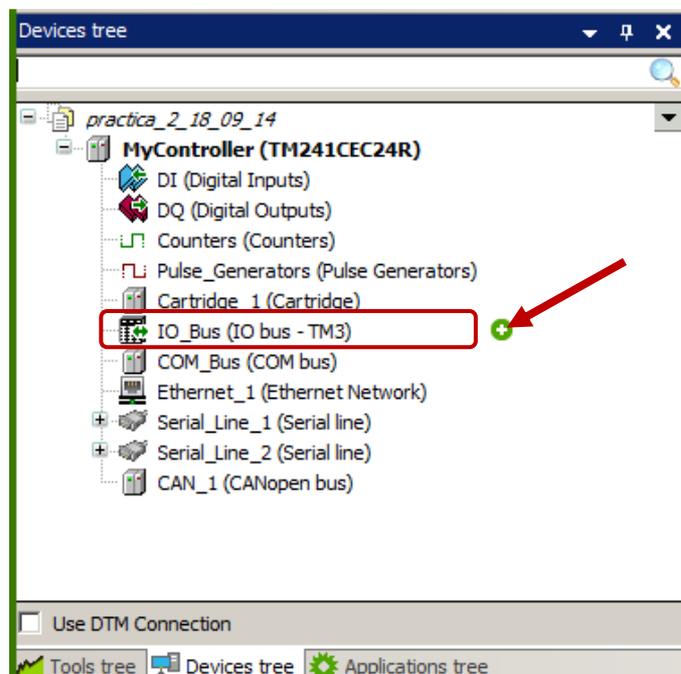
## Añadir un módulo de expansión a un equipo

Como añadir un módulo de expansión a un equipo

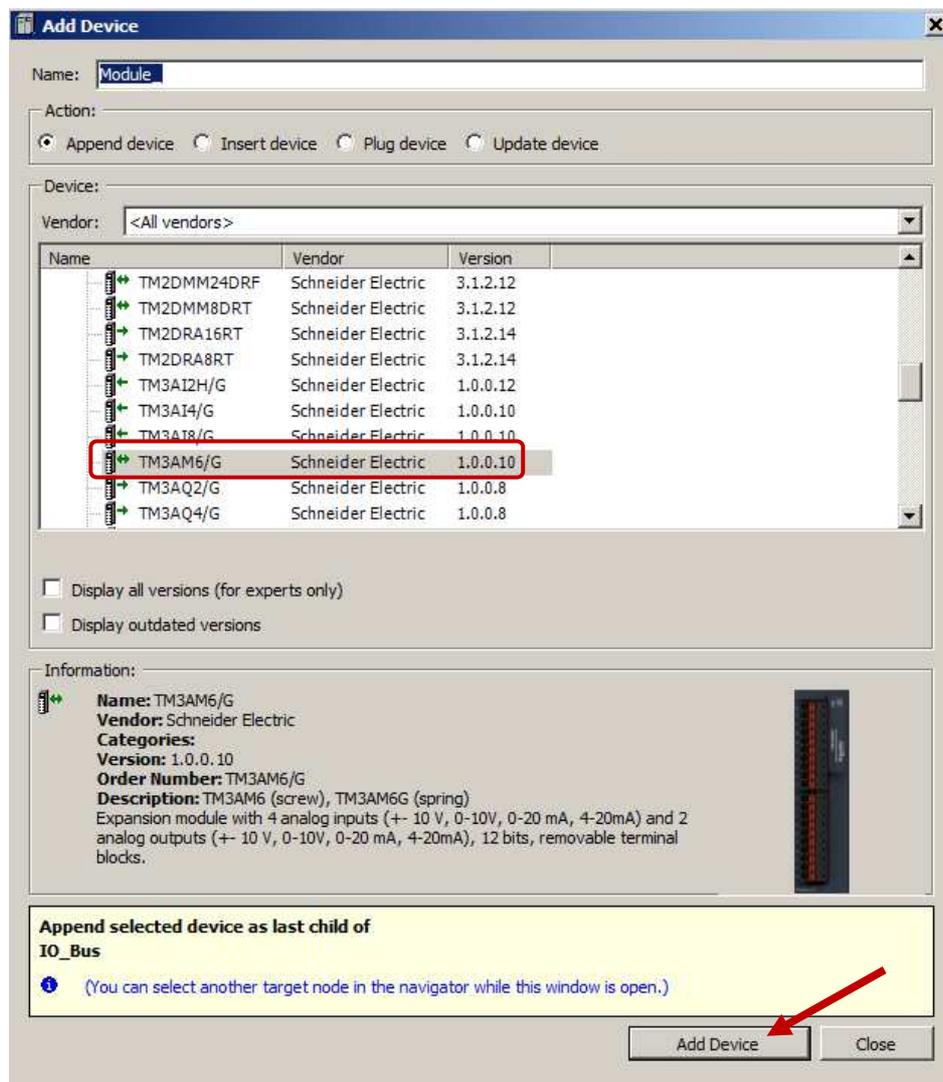
- Para añadir un módulo de expansión a un equipo:



Se abre la ventana de programación 'Logic Builder' y aparece en el 'Navegador' ya aparece seleccionada la pestaña de 'Device tree' expándimos el controlador y al seleccionar 'IO\_BUS (IO bus - TM3)' aparece en lado de la derecha un simbolo de '+' para añadir módulos de expansión.



Aparecerá la ventana flotante de 'Add Device', en ella aparecen todos los módulos de expansión que se pueden añadir a ese equipo (tanto TM2, TM3 y TM4). Para agregarlo bastará con seleccionarlo en la lista y pulsar el botón de la parte de abajo 'Add Device'. Cuando se selecciona un módulo en el campo 'Information' aparece una descripción del módulo seleccionado.

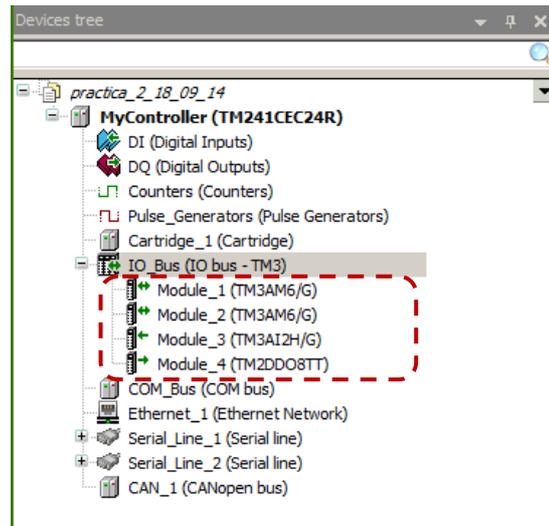


Cuando hayamos incluido los módulos de expansión deseados pulsaremos 'Close' para cerrar la ventana.



**Nota:** Hay tener en cuenta la limitaciones hardware de cada uno de los controladores a la hora de añadir módulos de expansión. Por ejemplo, el M241 tiene una limitación de 7 módulos de ampliación siempre que no se ponga módulos de expansión, también hay que tener en cuenta, que en este controlador, los módulos TM2 no se pueden intercalar con modulos TM3, sinó hay que ponerlos al final del bus del PLC.

Debajo del 'IO\_Bus (IO bus – TM3)' en el 'Navegador', aparecerán representados todos los módulos de expansión añadidos.

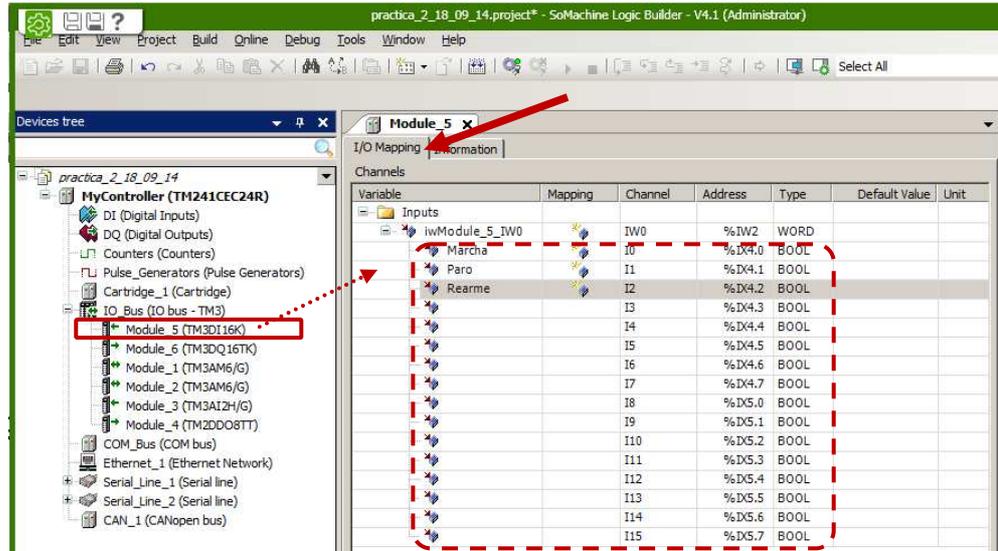


## Configurar módulos de expansión

### Configuración de los módulos de I/O discretas

Si dentro del 'Navegador' hacemos doble clic en uno de los módulos de ampliación añadidos, en el área de trabajo aparecen la ventana de configuración del módulo.

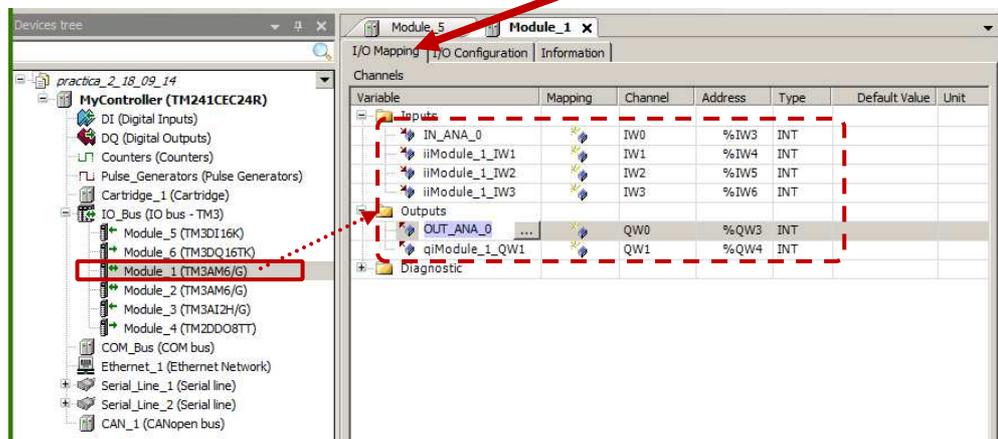
En la pestaña 'I/O Mapping' podemos asignarle un símbolo a cada una de las entradas ó salidas físicas del módulo, escribiéndolo en el campo 'Variable' de cada una de ellas.



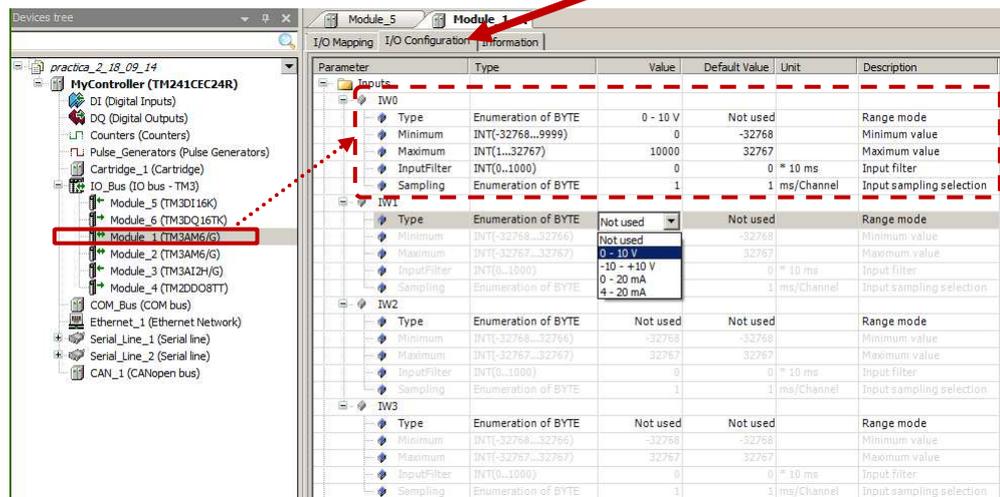
### Configuración de los módulos de I/O analógicas

Para la configuración de los módulos de ampliación de entradas ó salidas analógicas, dentro del 'Navegador' hacemos doble clic en uno de los módulos analógicos añadidos, en el área de trabajo aparece la ventana de configuración del módulo.

En la pestaña 'I/O Mapping' podemos asignarle un símbolo a cada una de las entradas ó salidas analógicas del módulo



En la pestaña 'I/O Configuration' activaremos aquellas entradas/ salidas analógicas que vamos a utilizar y la configuraremos.



Para configurar una entrada análogica, primero tenemos que determinar que tipo de señal análogica nos llega (eso dependerá del tipo de sensor), en el campo 'Type' en la columna 'Value', seleccionaremos entre (Not Used, 0 - 10V, -10 - +10 V, 0 - 20 mA, 4 - 20mA). Cuando seleccionamos un tipo de señal se habilita el resto de campos de la configuración.

- **Minimum:** Valor mínimo de la señal (quiere decir que si hemos configurado una señal 4 - 20 mA cuando a la señal física tenga los 4 mA que valor numérico queremos ver en el programa, por defecto 0).
- **Maximun:** Valor máximo de la señal (quiere decir que si hemos configurado una señal 4 - 20 mA cuando a la señal física tenga los 20 mA que valor numérico queremos ver en el programa, por defecto 10000).
- **Input Filter:** Valor de la ventana de tiempo de filtrado (quiere decir que para señales que lleguen con ruido y que pueden dar valores no reales debido a las interferencias, se utiliza este filtro para minimizarlo).
- **Sampling:** Valor de muestreo (Es el tiempo de muestreo de cada una de las señales analógicas).



# Capítulo 5: PLC Programming

## Descripción

### Este CAPÍTULO trata los siguientes temas:

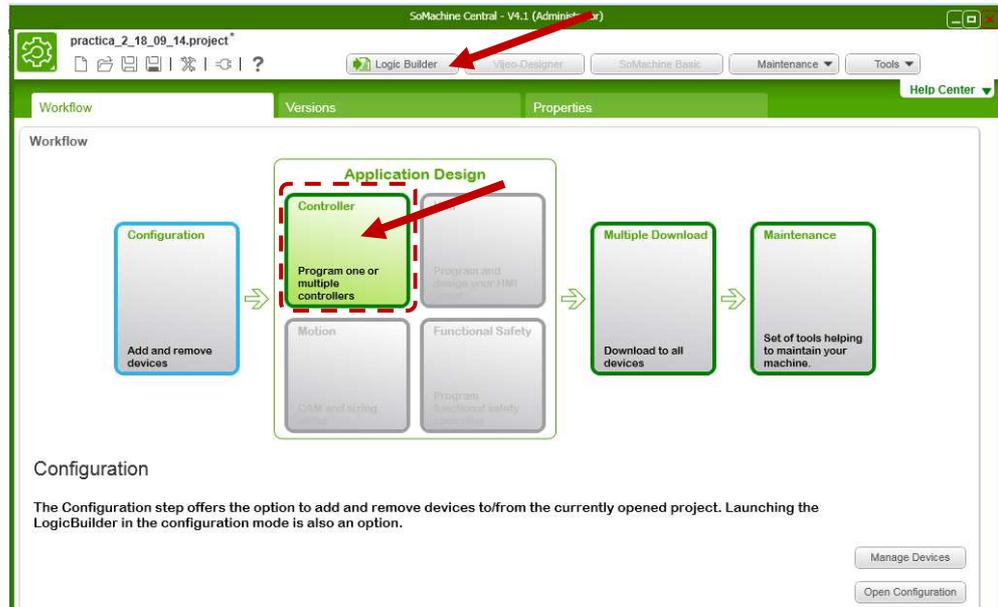
- Logic Builder - Ventana de Programa .....5-8
- Objetos de la pestaña de aplicación .....5-14
- Objetos de la pestaña de dispositivos.....5-14
- Ejercicio – Crear proyecto de Irrigación .....5-14
- Creación de un Programa POU .....5-17
- Tareas .....5-19
- Ejecución del programa .....5-20
- Configurador de tareas .....5-211
- Ejercicio - Configurar una tarea.....5-244
- Simulación PLC .....5-2525
- Lenguajes de programación de CoDeSys .....5-277
- Ejercicio – Programar un POU en FBD .....5-30
- Ejercicio – Convertir un programa FBD a IL ó LD ...5-304
- Watchdog .....5-355
- Estructura de una aplicación .....5-366
- POU tipo función .....5-37
- POU tipo bloque de función .....5-38
- Ejercicio – Crear un bloque de función .....5-40

## Logic Builder – Ventana de programación

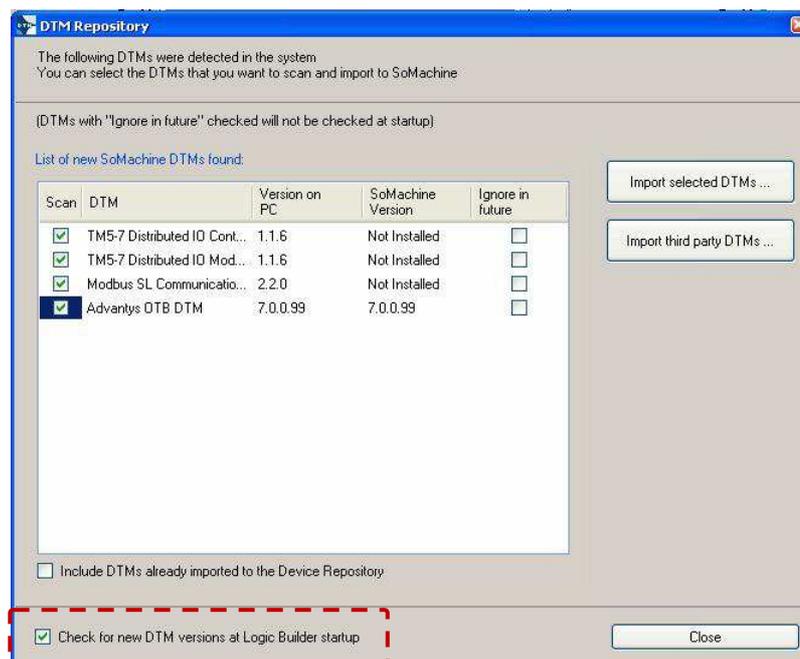
### Acceder al Logic Builder

Para abrir la ventana de programación 'Logic Builder', desde la ventana general del SoMachine Central, seleccionaremos la pestaña de 'Workflow' (flujo de trabajo). Dentro de los bloques del flujo de trabajo de un proyecto seleccionamos el bloque 'Controller' dentro de 'Application Design'.

O en la barra de herramientas de acceso, que se muestra en la parte superior de SoMachine Central, seleccionar 'Logic Builder'.

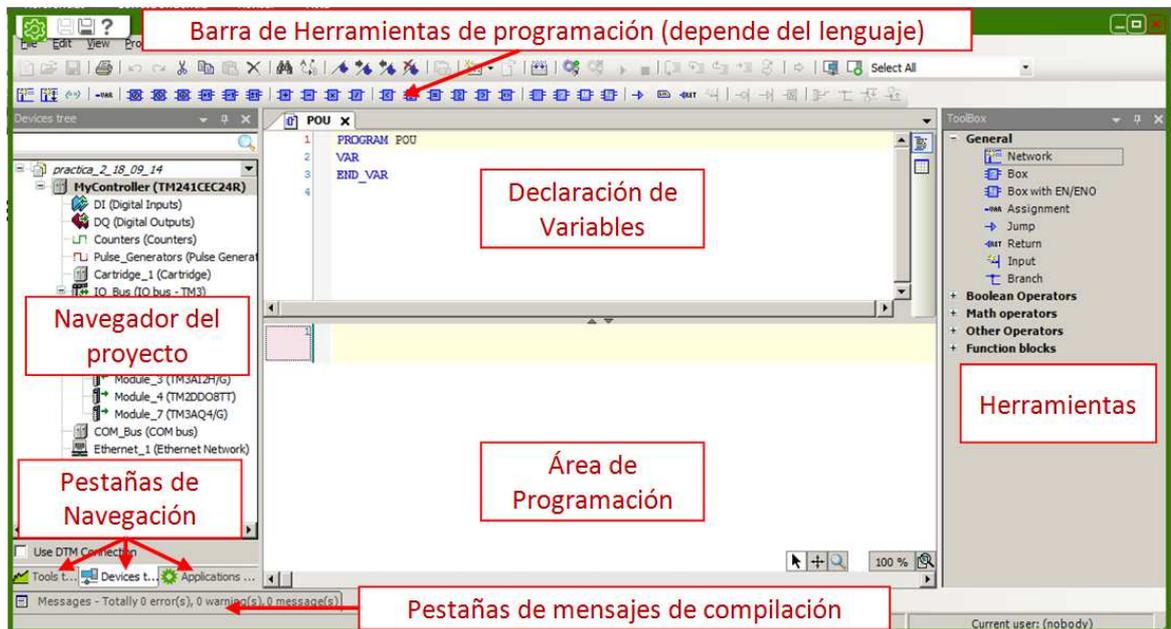


Cuando abrimos el Logic Builder. Aparece una ventana flotante, que nos solicita agregar nuevos DTM para este proyecto. Simplemente cierre esta ventana si aparece. Esta ventana se puede desactivar de forma permanente el 'check option' ubicado en la parte inferior izquierda.



## Interface Logic Builder

En esta figura se muestra las diferentes áreas del interface de programación.



- **Barra de herramientas de programación:** barra de acceso rápido a los elementos principales de programación, los iconos de la barra de programación, variarán en función del lenguaje de programación elegido para ese POU.

### Barra de herramientas del diagrama de contactos (LC)



### Barra de herramientas de diagrama de funciones (FBD)

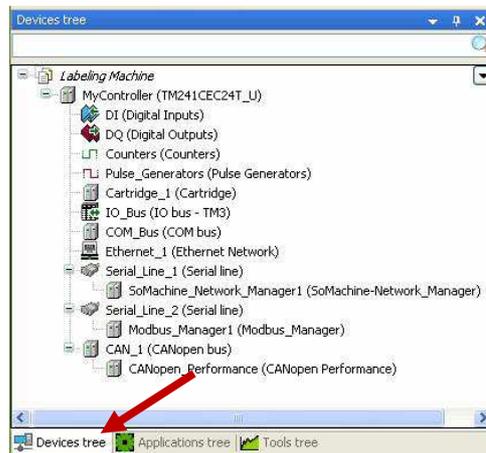


### Barra de herramientas de diagrama de funciones (CFC)

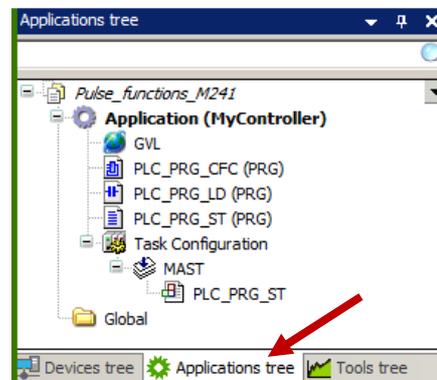
- **Navegador de proyecto:** En esta área se muestra los diferentes elementos que componen el programa, estructurado para una mejor comprensión de que se está haciendo en el proyecto.

El navegador consta de tres pestañas

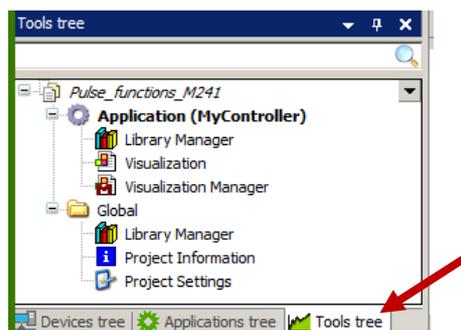
- **Pestaña de Dispositivos:** se utiliza para configurar el hardware del controlador elegido.



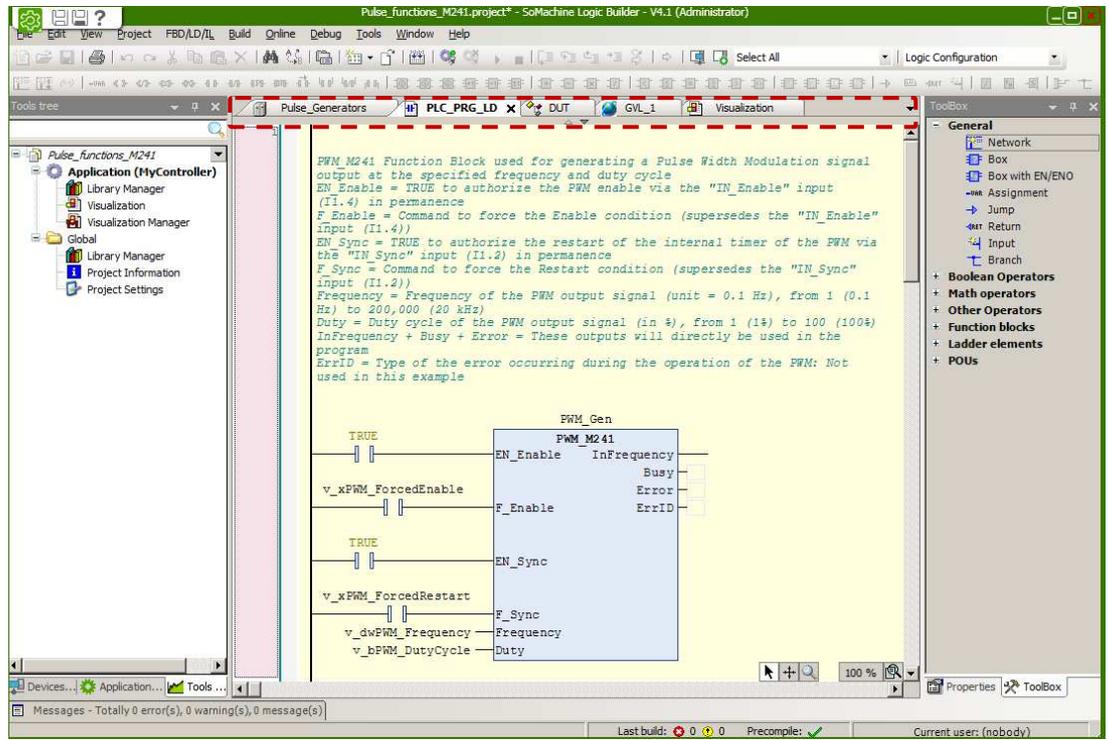
- **Pestaña de Aplicación:** se utiliza para realizar la programación de la aplicación y todos los elementos relacionados (lista de variables globales, tareas, estructura de datos, visualizaciones, POU's).



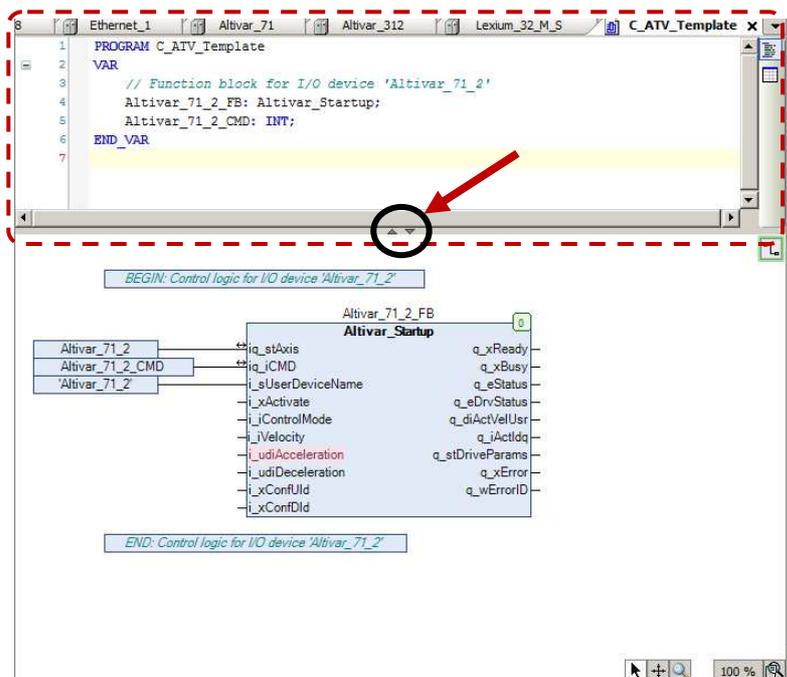
- **Pestaña de Herramientas:** en esta pestaña se encuentra el administrador de bibliotecas y las visualizaciones que se creen en el proyecto.



- **Area de trabajo:** En la zona de trabajo es donde se realizarán todas las parametrizaciones, programación y llamadas de cada uno de los elementos seleccionados en las diferentes pestañas de navegación. Las llamadas realizadas a los diferentes objetos en el navegador quedarán como pestañas de acceso rápido en la parte superior del área de trabajo.



- **Declaración de variable:** Cuando se abre un POU, en el área de trabajo aparece un área privada de declaración de variables de este POU (estas variables serán privadas de este POU).



**Nota:** A veces el área de declaración de variable aparece cerrada la podemos desplegar y plegar pulsando en los pequeños triángulos que se encuentran en el área de programación.

Además podemos seleccionar en que formato queremos declarar las variables, pudiendo utilizar uno u otro en función de lo que más nos convenga, para intercambiar el formato de la declaración, están los iconos en la parte derecha superior del área de declaración.

**Formato Texto:**

```

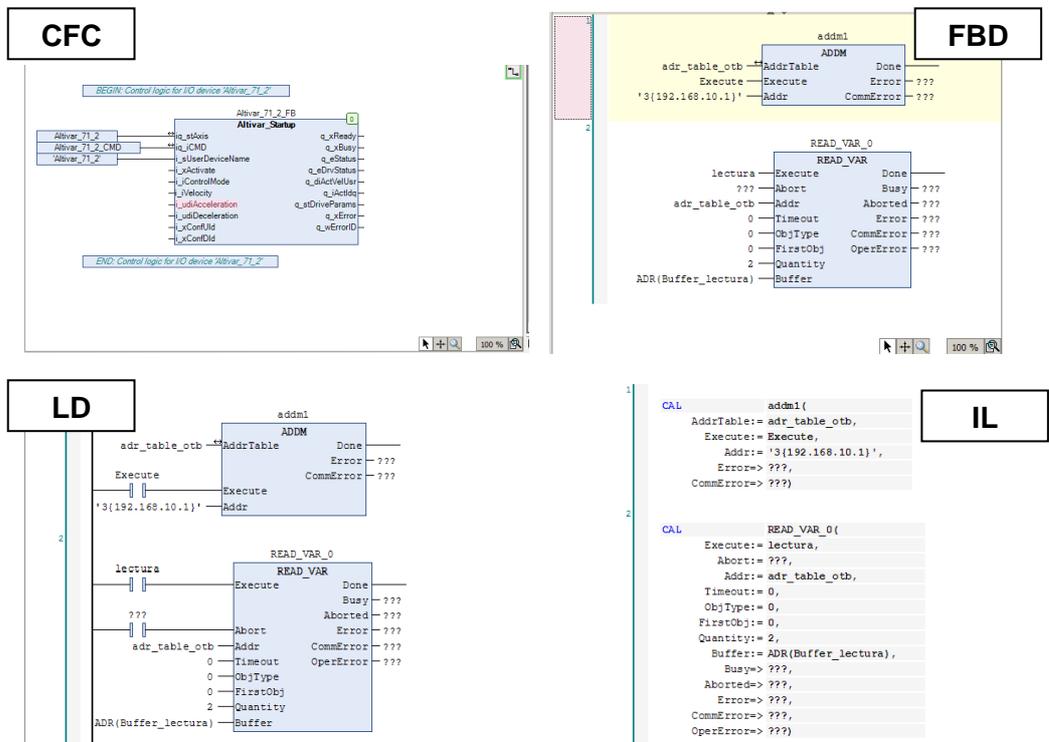
1 PROGRAM C_ATV_Template
2 VAR
3 // Function block for I/O device 'Altivar_71_2'
4 Altivar_71_2_FB: Altivar_Startup;
5 Altivar_71_2_CMD: INT;
6 END_VAR
7

```

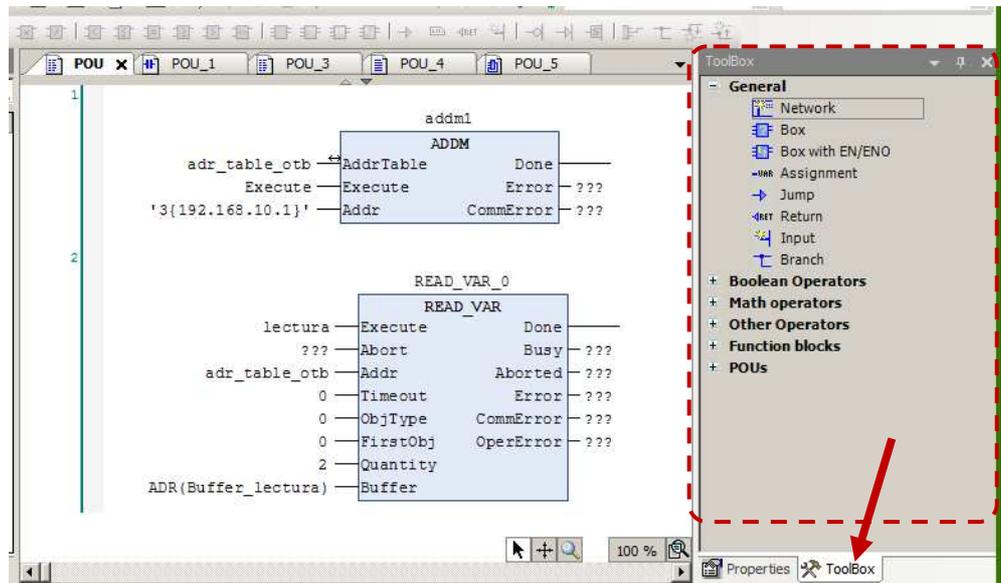
**Formato Tabla:**

	Scope	Name	Address	Data type	Initialization	Comment
1	VAR	Altivar_71_2_FB		Altivar_Startup		Function block for I/O device 'Altivar_71_2'
2	VAR	Altivar_71_2_CMD		INT		

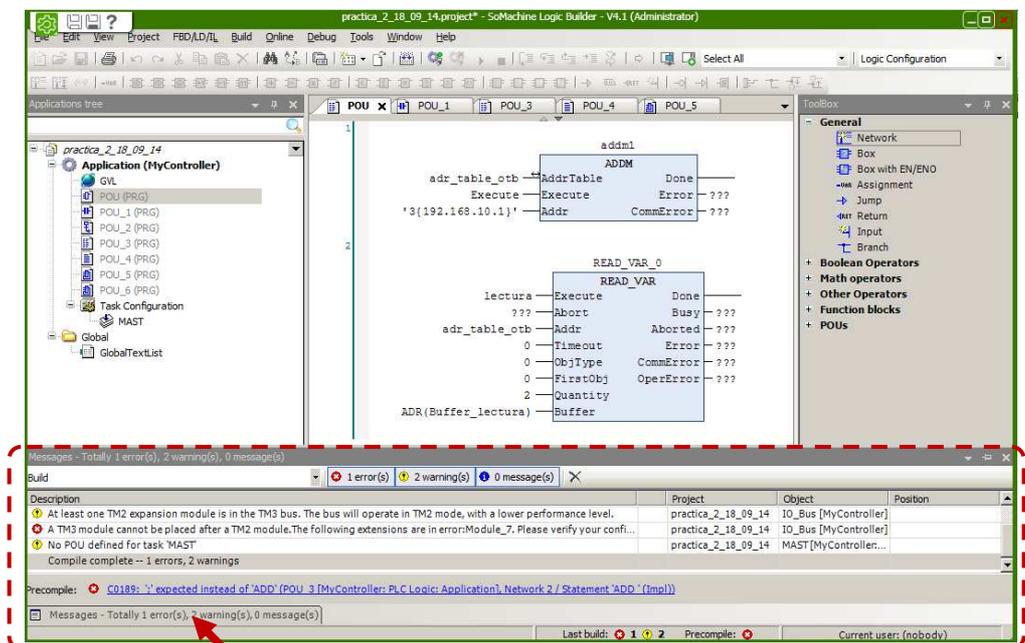
- **Área de programación:** es el área donde se realizarán la programación del POU seleccionado, el formato de visualización dependerá del lenguaje de programación seleccionado.



- **Herramientas:** Área que muestra las funciones herramientas que se pueden utilizar, para la programación del POU, o del objeto que esté en el área de trabajo en ese momento (Por ejemplo una visualización).

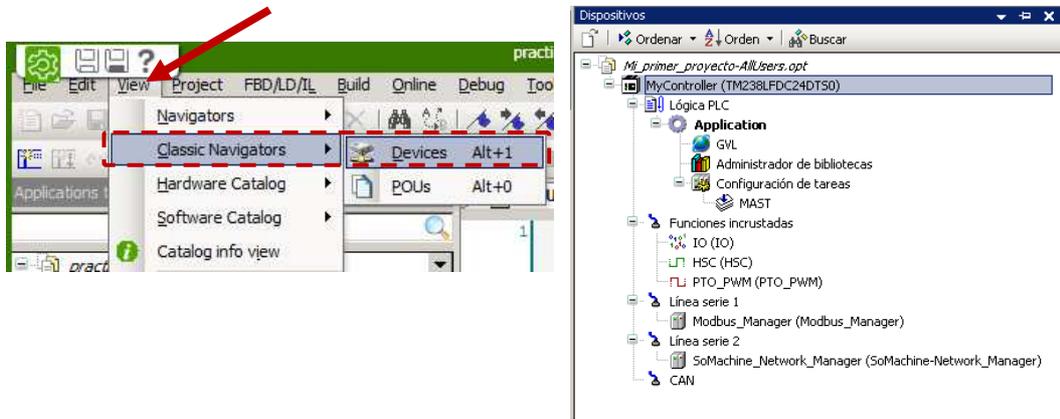


- **Pestaña de mensajes de compilación:** Esta pestaña que normalmente se encuentra en la parte inferior de la ventana de Logic Builder, nos muestra los mensajes de compilación (errores, advertencias y mensajes).



**Ventana clásica de programación**

A nivel del navegador del proyecto, es posible visualizar el proyecto de la manera clásica, de los anteriores versiones de SoMachine, donde todos los objetos del proyecto estaban juntos, sin estar separados en pestañas. Si se ha trabajado con versiones anteriores y se quiere visualizar el navegador del proyecto de la manera clásica, iremos al menú contextual 'View' y seleccionaremos 'Classic Navigator'.



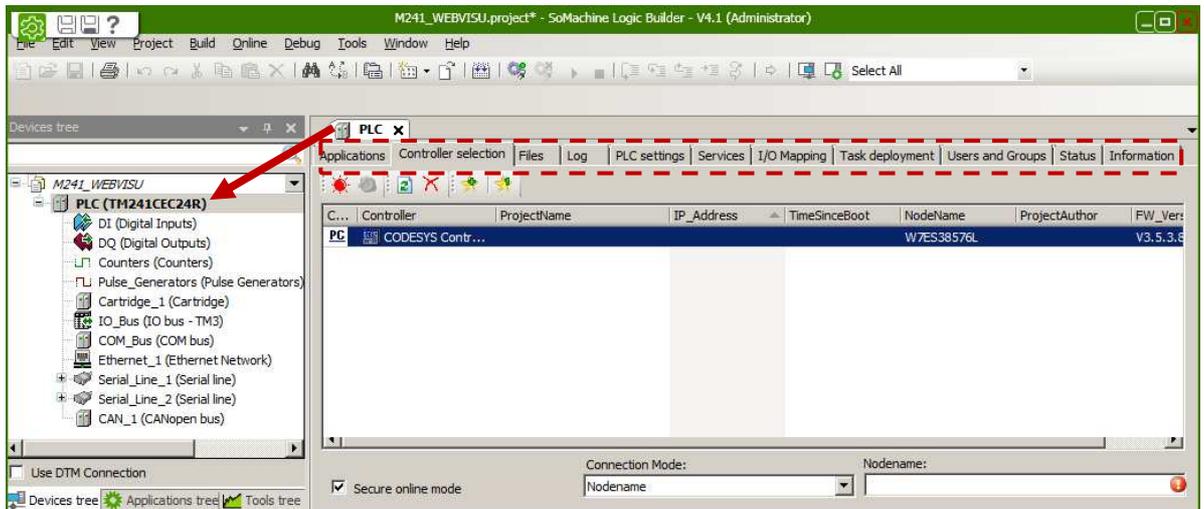
**Objetos principales del proyecto**

La ventana de dispositivos en cuatro nodos principales. Los cuatro nodos principales son:

Nodo	Descripción
<b>Controlador</b>	Tipo de PLC y configuración de los parámetros.
<b>Aplicación</b>	Todas las partes del programa POU que habrá en el PLC's
<b>Funciones Incrustadas</b>	Funciones del M238/M241/M258: <ul style="list-style-type: none"> <li>➤ I/O (fast &amp; normal I/O),</li> <li>➤ HSC entradas de conteo rápido,</li> <li>➤ PTO y PWM (salidas de tren de pulsos, Ancho de pulsos modulado)</li> </ul>
<b>Comunicación</b>	Configuración de los puertos de comunicación. <ul style="list-style-type: none"> <li>➤ Lineas Serie</li> <li>➤ Equipos CAN</li> <li>➤ Ethernet</li> <li>➤ etc.</li> </ul>

**Pestañas del nodo controlador**

Para poder conectarse al autómatas, se tiene que establecer una ruta activa en el **Gateway**. El Gateway hay que crearlo en el controlador en la pestaña de **'Configuración de comunicación'**.

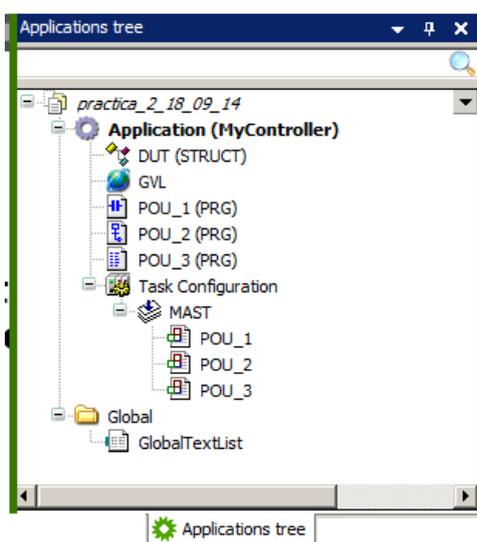


Las pestañas de configuración incluidas en el nodo del controlador, son:

Pestañas	Descripción
<b>Aplicaciones</b>	Muestra las aplicaciones que hay en el PLC.
<b>Ajustes PLC</b>	Nombre de la aplicación que está corriendo en el PLC, comportamiento de las salidas cuando el PLC está en estado de STOP.
<b>Servicios</b>	Poner el PLC en hora.  Firmware rev, Boot version
<b>Estatus</b>	Estado del PLC y del bus de expansión
<b>Información</b>	Información sobre el modelo del PLC conectado

## Objetos de la pestaña de aplicación

La sección de Programa (Lógica PLC) consta de varios ítems:

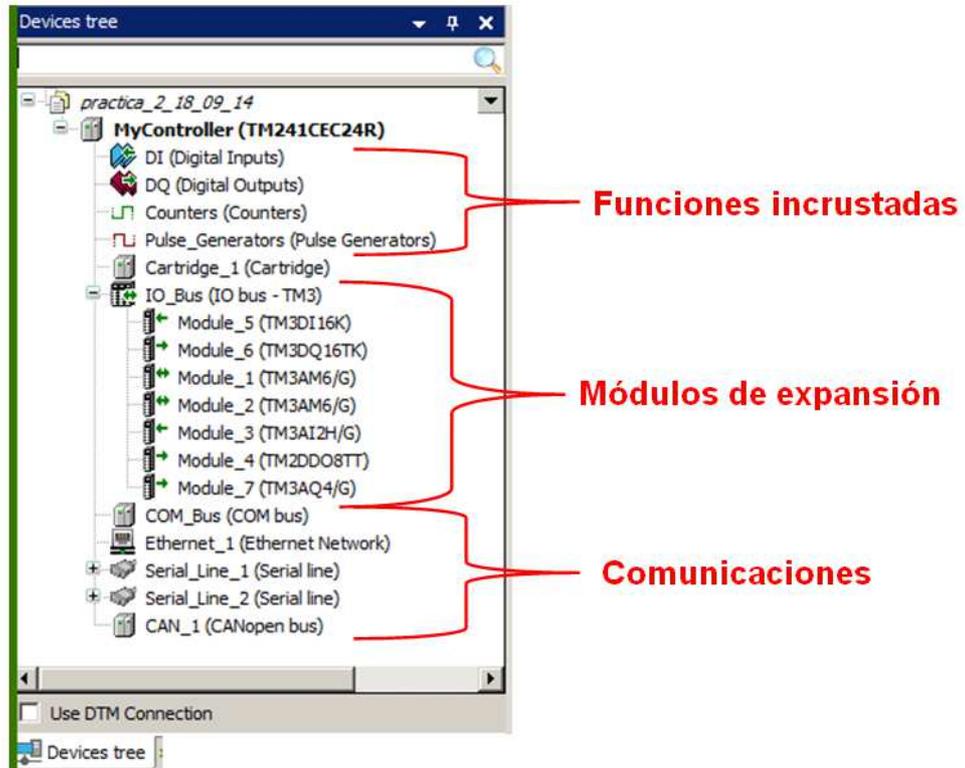


- **Application**  
Engloba todas las partes del programa del PLC's.
- **GVL** (Variables Globales)  
En este ítem se declaran todas las variables globales del programa, es decir, las variables que serán visibles para todos los POU's del proyecto.
- **DUT** (Estructuras de datos)  
En este objeto se pueden crear estructuras de datos generadas por el usuario.
- **POU** (Programas (PRG), Bloques de Función (FB) y Funciones (FUN))  
Llamados **POU (Program Organizational Unit)**. El usuario puede crear secciones o funciones en el proyecto programada en cualquiera de los 6 lenguajes de programación IEC.
- **Configuración de tareas**  
Controla la ejecución de los programas en el proyecto. La tarea MAST – se crea por defecto por el sistema en el momento que creamos un proyecto nuevo.

## Objetos de la pestaña dispositivos

### Pestaña Devices

En la pestaña 'Devices' se encuentra todos los objetos relacionados con el hardware configurado. Los objetos dependen del controlador seleccionado.



### Funciones Incrustadas

#### ➤ IO

En esta opción se puede realizar la configuración de las entradas y salidas que hay embebidas en el controlador. En función del tipo y modelo de controlador el número de entradas y salidas variará. *Ejemplo: el M241 tiene 6 entradas digitales, 8 entradas rápidas, 6 salidas digitales y 4 salidas rápidas.*

E/S de configuración		Asignación E/S			
Parámetro	Tipo	Valor	Valor predeterminado	Unidad	
Entradas					
IO					
Filtrado	Enumeration of BYTE	No	No	ms	
Retención	Enumeration of BYTE	No	No		
Evento	Enumeration of BYTE	No	No		
Filtro de rebote	Enumeration of BYTE	0,004	0,004	ms	
Run/Stop	Enumeration of BYTE	No	No		

➤ **HSC**

Configuración de las entradas de conteo rápido. La función HSC puede contar pulsos provenientes de sensores, encoders, etc. Las HSC son independientes del tiempo de scan del controlador.

Parámetro	Tipo	Valor	Valor predeterminado	Unidad	Descripción
HSC					
HSC000					
Tipo	Enumeration of BYTE	No utilizado	No utilizado		Tipo de contador
Parámetros					
Entradas del reloj					
Entradas auxiliares					
Umbral					
Salidas reflejas					

➤ **PTO\_PWM**

La función **PTO** nos genera una señal de pulsos con un número específico de pulsos y un periodo de tiempo determinado.

Las funciones PTO se pueden configurar en dos modos diferentes.

- Tren de pulsos.
- Pulso de ancho modulado PWM.

Configuración de las salidas como tren de pulsos PTO ó pulsos de ancho modulado PWM.

Hay dos canales PTO disponibles, cada canal PTO están asociadas a 2 salidas rápidas y una entrada estándar.

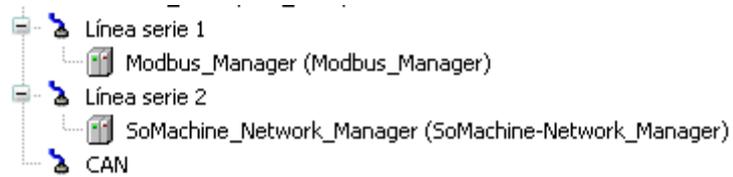
El **PWM** usa una modulación de ancho de pulsos, esta es una señal cuadrada, donde el periodo de la señal varía.

Parámetro	Tipo	Valor	Valor predeterminado	Unidad	Descripción
PTO/PWM					
PTO000					
Modalidad	Enumeration of BYTE	No utilizado	No utilizado		Modalidad de uso
Modalidad de salida	Enumeration of BYTE	Pulso/Dirección	Pulso/Dirección		Modalidad de generación de salidas
Aceleración/Deceleración					
Frecuencia					
Entradas auxiliares					
Punto de referencia					

Como máximo se pueden utilizar dos canales PWM.

## Comunicación

Dependiendo del tipo y modelo de controlador, se disponen de más o menos puertos y protocolos de comunicación, Para el modelo M238 TM238LFDC24DT, por ejemplo, sería:



- Puerto serie 1: Modbus manager (**configuración por defecto**).
- Puerto serie 2: SoMachine - Network manager (**configuración por defecto**).
- CANbus
  - Para conectar remotamente con otros equipos CANopen
  - 16 equipos máximo.

## Ejercicio – Crear proyecto de irrigación.

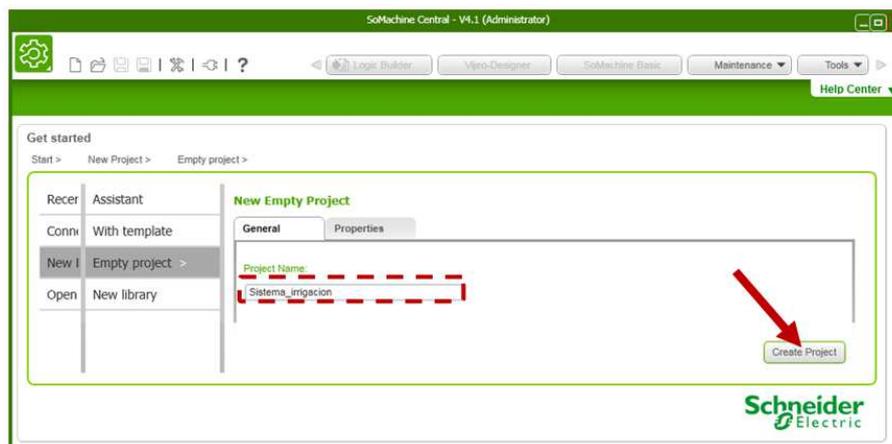
### 1 Crear un proyecto nuevo para la programación de un sistema de irrigación.

Abrir el SoMachine o ir a la ventana de inicio.

Hacer clic en '**New Project**' en barra de tareas situada en la parte izquierda de la ventana.

Hacer clic en la opción '**Empty Project**' del menú abierto.

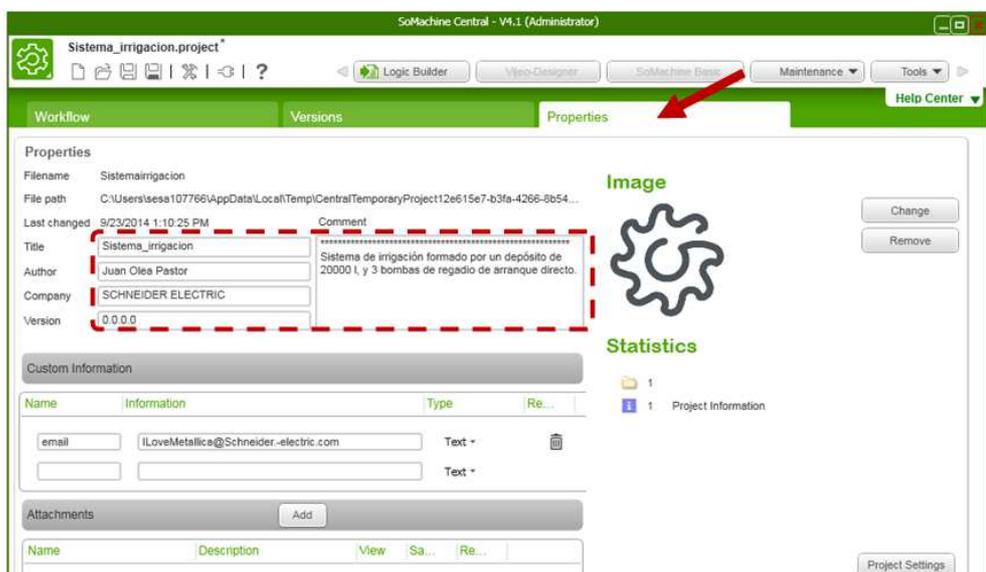
En la pestaña de '**General**' ponemos el nombre que deseamos que tenga el proyecto '**Sistema\_irrigacion**'. Y lo creamos pulsando en el botón '**Create Project**'.



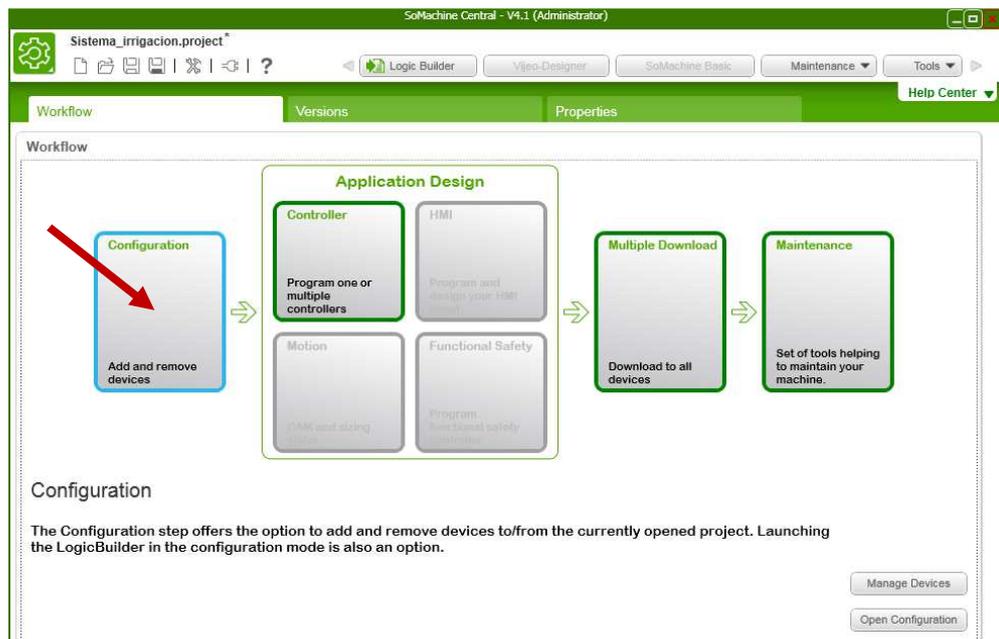
Ahora en el SoMachine Central guardamos el archivo con el nombre '**Sistema \_ irrigación**' en la carpeta C:\Class.



Cuando el proyecto esté guardado en SoMachine, añadir en la pestaña de 'Propiedades' del SoMachine Central, los campos de: Título, Autor, Compañía y Comentarios. (Veremos que el área de 'Información del proyecto' se irá actualizando con la información que se vaya introduciendo).



Una vez introducida la información necesaria para identificar el proyecto, Hacer clic en pestaña **'Workflow'** para ir a la ventana general SoMachine Central y pulsar **'Configuration'**.



En la ventana de configuración, en el área de la izquierda **'Catalog'** desplegamos **'Logic Controller'**, desplegamos la familia **'M241'** y seleccionamos el modelo de controlador **'TM241CEC24R'** y pulsamos la flecha hacia la derecha, que ha aparecido en la parte central.



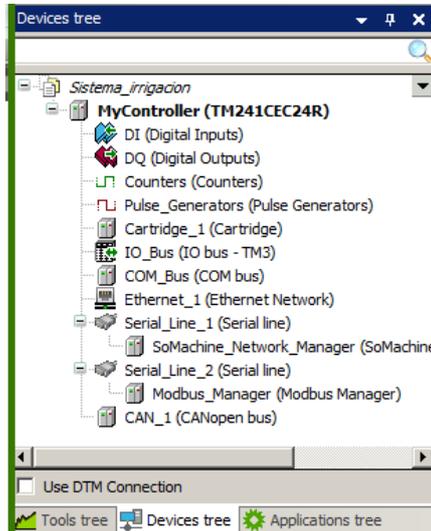
Al cabo de unos segundos aparece el controlador seleccionado en el área derecha de la ventana **'Project Devices'**.

Cuando hemos añadido el controlador, pulsamos el botón **'OK'** para salir de la ventana de configuración.

Abrir la ventana de programación '**Logic Builder**' a través de la barra de acceso rápido del SoMachine Central.

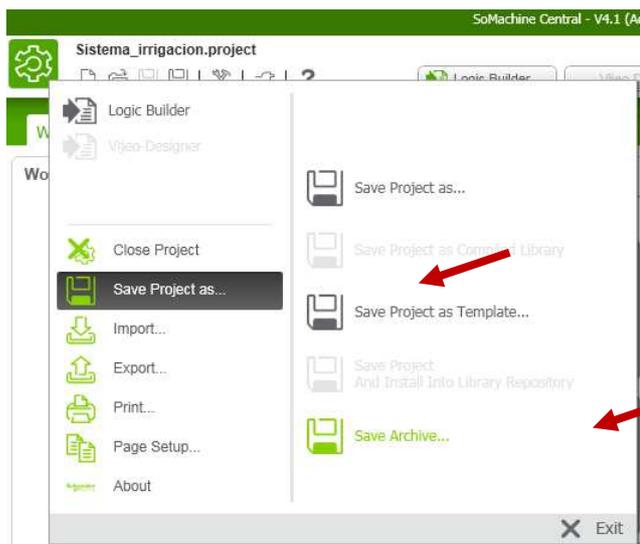


En el Logic Builder, ver la ventana de '**Navegación**' la pestaña de '**Dispositivos**', comprobar si el hardware del controlador seleccionado.



Sin crear ningún programa, dentro de la ventana de programa '**Logic Builder**', pulsar el icono  para volver a la ventana central del SoMachine Central.

Desplegar el icono de menú general, seleccionar 'Save Project as...' y elegir la opción 'Save Archive' guardarlo como '**sistema\_irrigacion.projectarchive**'.



Guardar y cerrar el proyecto. He ir a la carpeta donde están los dos archivos y comparar su tamaño. (Este proyecto se irá utilizando a lo largo de este curso).

## Creación de Programas 'POU'

### Program Organization Units (POU)

La ventana de dispositivos permite al usuario añadir **POU's** (*Program Organization Units*) a la aplicación. Un **POU** es una sección de programa donde se escribirá el código del programa.

```

1  PROGRAM POU_2
2  VAR
3      var1,var2,result1:INT;
4  END_VAR
5

```



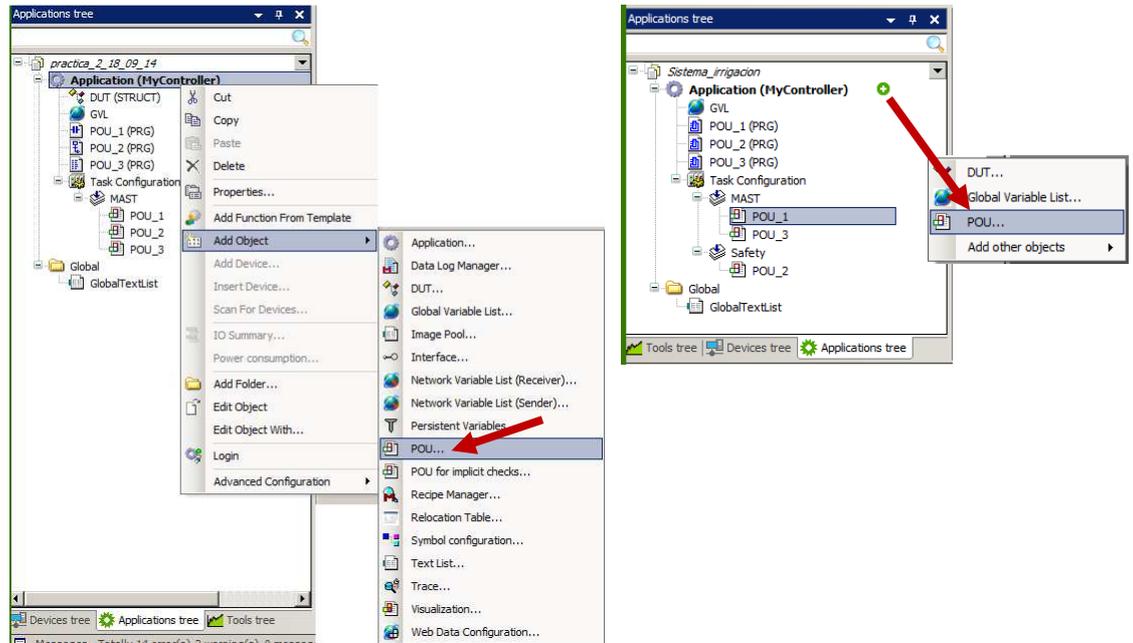
Hay tres tipos de POU's diferentes:

Tipos de POU	Descripción
<b>Programa</b>	Devuelve diferentes valores durante su procesado. Todos los valores se mantienen de un ciclo de máquina al anterior. Este puede ser llamado por otro POU.
<b>Bloque de funciones</b>	Devuelve diferentes valores durante su procesado de un programa. Es opuesto a la función, porque puede tener más de un valor de salida y necesita variables internas que tiene que ser persistente entre un ciclo de ejecución y el siguiente. Este tiene que ser llamado (una o más veces) desde un mismo POU programa.
<b>Función</b>	Devuelve una salida y no tiene variables internas persistentes, solo variables temporales entre un ciclo de ejecución y el siguiente no persisten los datos).

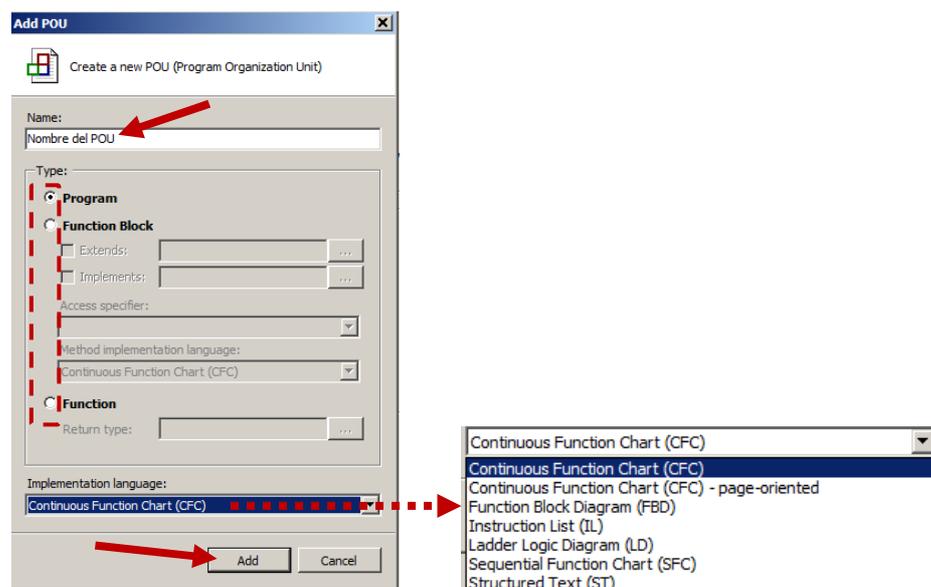
## Como crear un POU

### ➤ Para crear un POU:

Si se selecciona la vista clásic del navegador, hay que ir a la ventana de dispositivos, desplegar el árbol del 'My Controller', 'Lógica PLC' y seleccionar 'Application' hacer clic con el botón derecho del ratón y seleccionar 'Agregar Objeto'. En el caso que tengamos la vista normal hay que ir a la pestaña 'Application tree' desplegar el arbol de la aplicación y pulsamos el icono de '+' que aparece al lado de la aplicación luego seleccionamos 'POU'.



En la ventana flotante de 'Agregar Objeto', seleccionar el Objeto 'POU' en la barra de la izquierda de la ventana y después escribir en el campo 'Nombre' el nombre que desea para esa sección de programa, seleccionar el 'Tipo' de POU y el tipo de lenguaje de programación en la opción 'Lenguaje de implementación'.



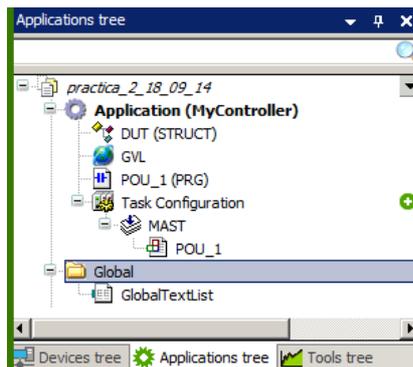
Todos los POU's se crearán de esta manera. Una vez creados, tiene que ser asignados a una **tarea**.

## Tareas

### Tareas Básicas

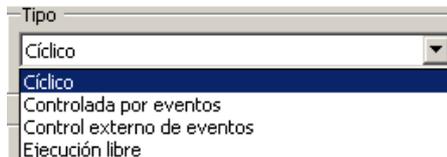
Una **tarea** es un grupo de secciones de programa y subrutinas ejecutadas de una manera cíclica o periódica por la tarea principal **MAST**, y periódica por la tarea **FAST** (tarea cíclica con un tiempo de ejecución menor que la tarea **MAST**).

El '**Configurador de Tareas**' permite definir una o varias tareas que se ejecutarán en el controlador. Las tareas controlan la ejecución de la aplicación.



Es posible configurar hasta siete tareas con las restricciones siguientes:

Hay **cinco** tipos de tareas:



- **Cíclico (3 máx.)** – ejecutada en un periodo de tiempo....cada 50ms
- **Controlada por eventos (2 máx.)** – se ejecuta cuando hay transición 0-1 de la variable de evento seleccionada.
- **Ejecución libre (1 máx.)** – la tarea se ejecuta cíclicamente pero sin un tiempo de ciclo específico.
- **Control externo de eventos (4 máx.)** – la tarea se ejecuta cuando un 'evento de sistema' es verdadero. Por ejemplo – una entrada digital se pone a **ON**.

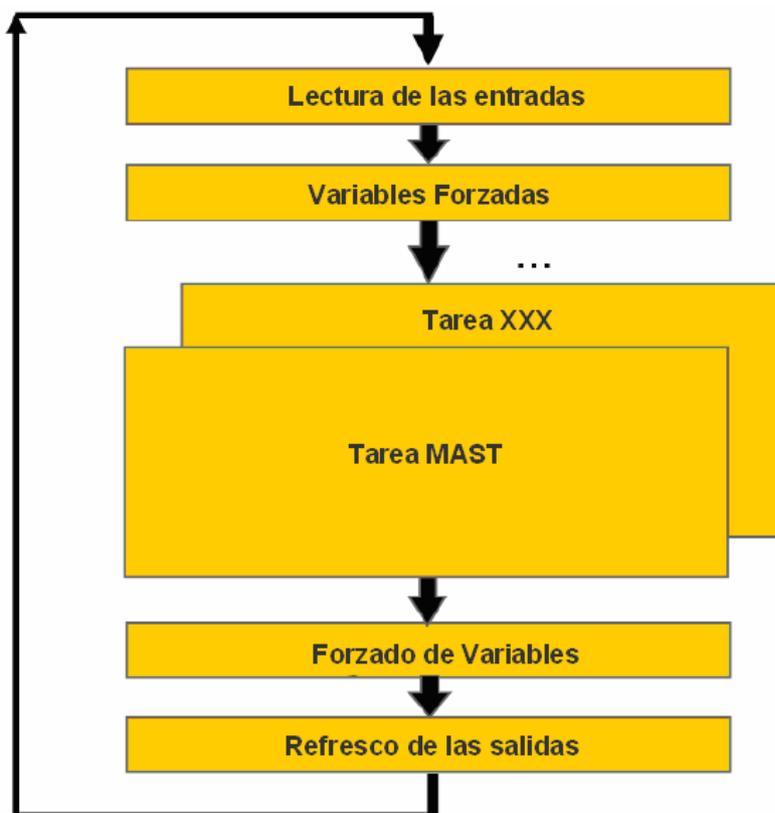
### Elementos que activan una tarea

Las **Tareas** pueden activarse por diferentes elementos:

- Por tiempo (cíclico, ejecución libre).
- Por un evento interno o externo.
  - Por cambio de estado de una variable global del proyecto.
  - Por un interrupción de un evento del controlador.
- La combinación de la prioridad de la tarea y la condición determina el orden de ejecución de estas.

## Ejecución de programa del PLC

Ciclo de programa del PLC



El diagrama de arriba describe el ciclo de ejecución del programa del PLC.

- **Entradas y salidas locales** - se procesan por las tareas que las utilizan (se pueden asignar a la tarea MAST, o a otra tarea creada Task Name-xxx,...).
- 
- **Módulos de expansión I/O** - solo se pueden utilizar en la tarea MAST, no se pueden utilizar en otras tareas existentes.

# Configurador de tareas

## Configurar una tarea

### ➤ Para configurar una tarea:

Lo primero será decidir que POU o POU's se desean añadir a la tarea.

Configuration

Priority ( 0..31 ): 15

Type  
Cyclic Interval (e.g. t#200ms): 20 ms

Watchdog  
 Enable  
Time (e.g. t#200ms): 100 ms  
Sensitivity: 1

- **Prioridad (0...31):** Define la prioridad que tendrá esta tarea. (0 es a prioridad más alta y 31 la más baja).

Sólo puede haber en ejecución una tarea cada vez. La prioridad determina cuándo se ejecutará la tarea:

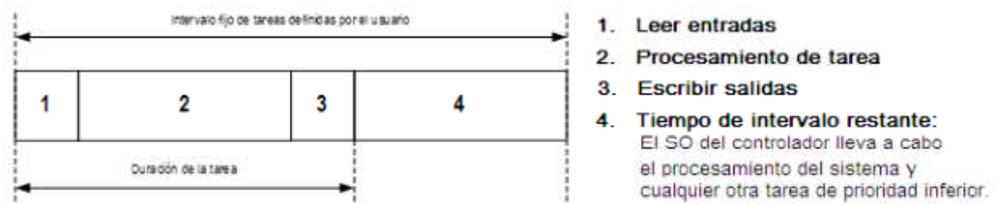
- Una tarea de prioridad alta se adelantará a otra de prioridad inferior
- Las tareas con la misma prioridad se ejecutarán en turnos (secciones de 2 ms)



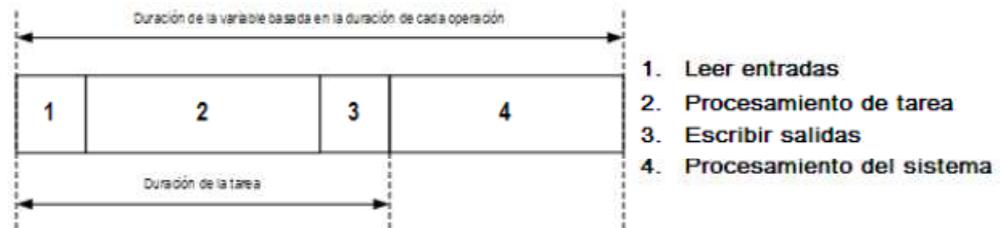
**Nota: No asigne tareas con la misma prioridad. Si aún hay otras tareas que intentan priorizar tareas con la misma prioridad, el resultado podría ser indeterminado e impredecible.**

- **Tipo:** Define el tipo de tarea.

- **Cíclico:** ejecuta cíclicamente la tarea en 'Intervalo' de tiempo definido.



- **Controlada por eventos:** ejecuta cuando hay un flanco positivo de la variable interna se ha configurado para activar la tarea.
- **Control externo de eventos:** se ejecuta cuando hay un flanco positivo de la entrada física que se ha configurado para activar la tarea.
- **Ejecución libre:** Esta tarea cíclica se ejecuta al ponerse el PLC en el estado de Run. No tiene un intervalo de ejecución definido por el usuario, pero internamente tendrá que estar entre T#: 1...1000 ms.



**Nota:** En el 'Procesamiento del sistema': El SO del controlador lleva a cabo el procesamiento del sistema y cualquier otra tarea de prioridad inferior. La duración del período de procesamiento del sistema se establece en el 30% de la duración total de las 3 operaciones anteriores ( $4 = 30\% \times (1 + 2 + 3)$ ). En todo caso, el período de procesamiento del sistema nunca será inferior a 3 ms.

- **Watchdog:** Habilita la opción de watchdog, se introduce el 'Tiempo' y la 'Sensibilidad'. La Sensibilidad configura el número de veces (continuo) que el watchdog se tiene que producirse antes de que salte el evento.

El Watchdog es una protección del sistema que se utiliza para proteger el equipo de errores de hardware y software que pueden provocar que el sistema pueda pararse.

Hay dos tipos de Watchdog en el SoMachine:

- **Watchdog de la Aplicación (configurable)** – Cada tiempo de ciclo se puede monitorizar y comparar con el tiempo de watchdog configurado (que es el tiempo máximo de ciclo que puede tardar la tarea en ejecutarse). Este ayuda a depurar la aplicación para que no se hayan programados bucles infinitos, etc.

El watchdog puede ser definido para cada tarea.

- **Watchdog de sistema** – El sistema se puede sobrecargar cuando las áreas creadas por el usuario usan más del 80% de los recursos del sistema.  
Este watchdog de sistema no se puede desactivar.

Estando en modo Online, en la ventana de dispositivos hacer doble clic en el 'Configurador de tareas' y seleccionar la pestaña 'Supervisión'.

Properties		Monitor				
Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)
MAST	Valid	21051	24716	14	15	457

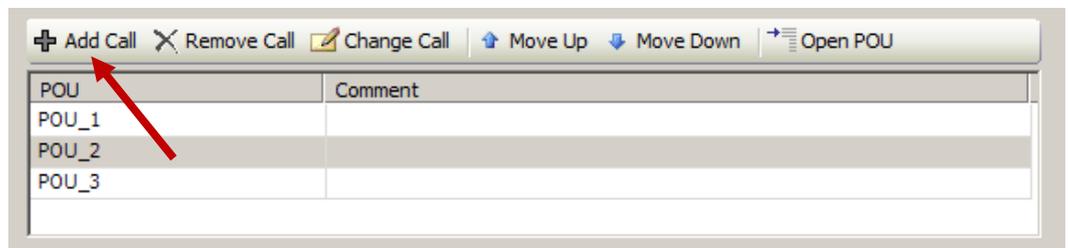
Monitorizar los tiempos de ejecución de las diferentes tareas que están en ejecución.

Ver el tiempo de ejecución medio de una tarea, ayuda

### Asignar un POU a una tarea

#### ➤ Asignar de un POU a una tarea

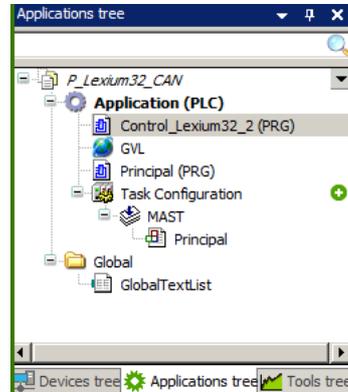
En este campo se añaden los POU's previamente creado en la tarea. Esto fija la ejecución de estas secciones de programa en la tarea, así como, el orden de ejecución.



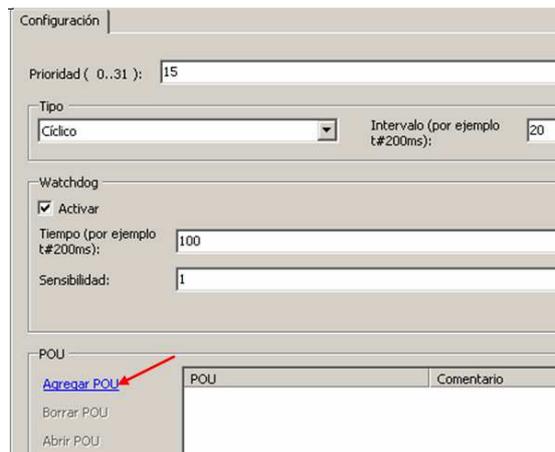
## Ejercicio – Configurar una tarea

### 2 Añade un POU a la tarea MAST.

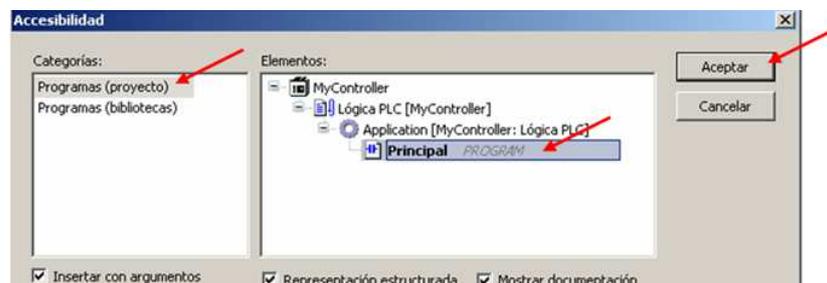
Hacer doble clic en el ítem MAST que hay en la ventana de dispositivos para abrir en la ventana de trabajo la configuración de esta tarea.



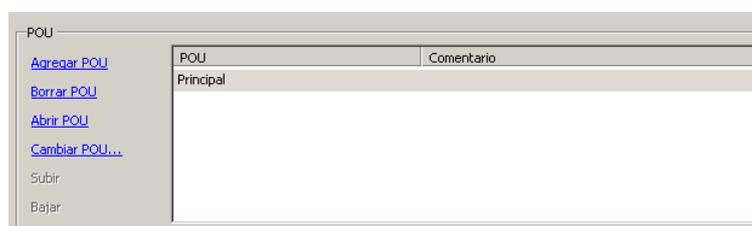
En la ventana de configuración seleccionar 'Añadir POU' dentro del campo 'POU' del configurador de tareas. Se abrir el asistente para seleccionar el POU que se desea añadir.



Expandir el árbol que sale de 'MyPLC' y selecciona el POU 'Principal' y pulsa el botón 'Aceptar'.



El POU añadido en la tarea MAST aparecerá en el configurador de tarea en el campo POU.



## Simulación del PLC

### Simulador del PLC

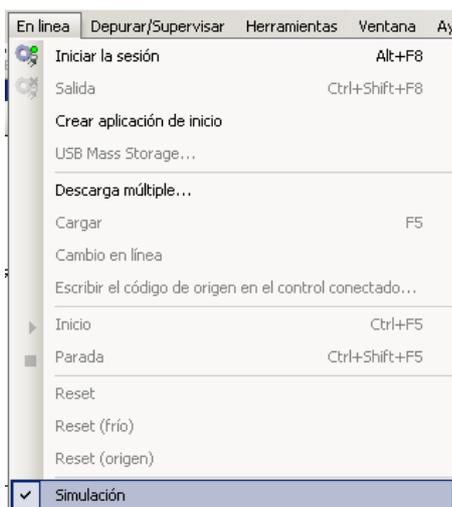
SoMachine contiene un simulador de PLC que permite al usuario descargar y ejecutar la aplicación programada en offline sin necesidad de tener el PLC físicamente. Esto puede ayudar al usuario a:

- Desarrollo del programa.
- Depurar el programa

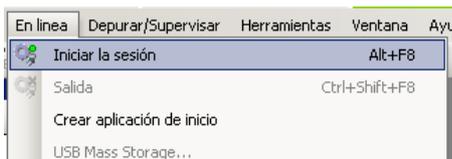
### Como ejecutar el Simulador

#### ➤ Para ejecutar el simulador:

En la barra de menú general de la ventana de programación, seleccionar **En línea » Simulación**, una vez seleccionado, se esta en modo simulación hasta que se deshabilite esta opción, deseleccionándola.

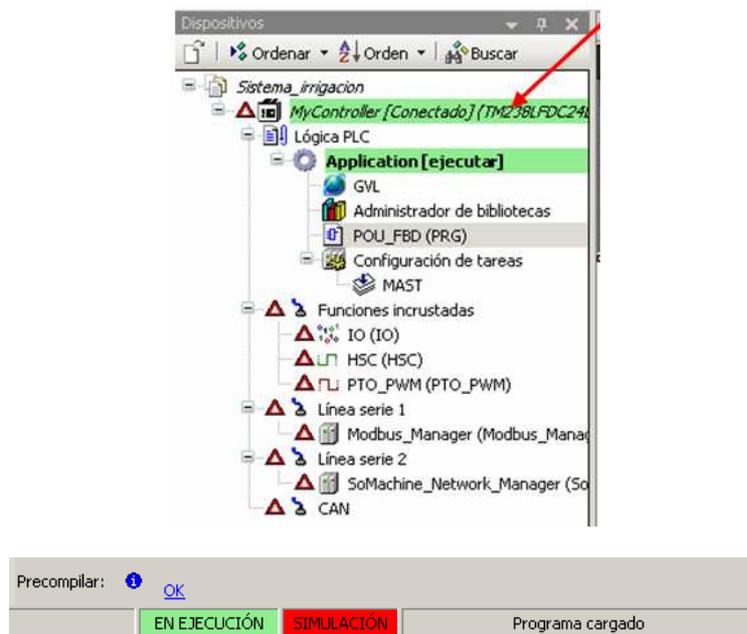


Seleccionar 'En línea » Iniciar la sesión' para conectarse al simulador y poder transferir la aplicación.

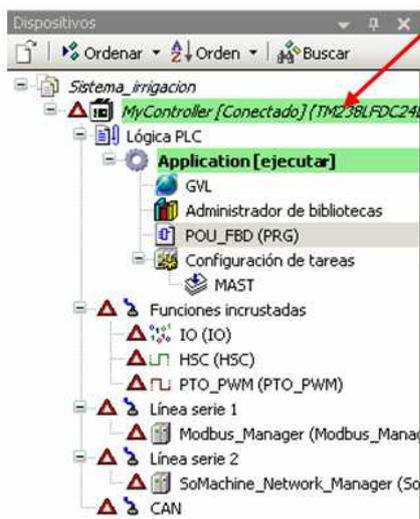


## Como ejecutar el Simulador

El modo de simulación se indica en el software con un rectángulo rojo con la palabra '**SIMULACIÓN**' situada en la línea de información situada en la parte inferior de la ventana. Además en la ventana de dispositivos el nombre de controlador se pone en letras cursivas.



En la pestaña de dispositivos saldrán todos los equipos con el triángulo de peligro, ya que indica que no se encuentran de manera real.



# Lenguajes de programación CoDeSys

## Introducción

Durante un periodo de entre diez, quince años cada fabricante desarrollaba para sus PLC's su propios lenguajes de programación. Los desarrolladores de aplicaciones de control creaban sus lenguajes propietarios basados en Ladder, C o Basic y muchos otros. Debido a las diferencias entre ellos el usuario que quería utilizarlos tenía que gastar tiempo para aprender a utilizar cada uno de ellos.

Durante el principio de los años 90's la **International Electrotechnical Commission (IEC)** desarrollo un estándar para todo los PLC's llamado **IEC 61131**. **IEC 61131-3 (Part 3)** fue creado en 1993. Esta parte del estándar normaliza los lenguajes de programación usados para los PLC's.

Con SoMachine es posible usar los seis lenguajes de programación estandarizados por la IEC. Estos son GRAFCET (**Sequential Function Chart (SFC)**), Diagrama de contactos - (**Ladder Diagram (LD)**), Diagrama de bloques de función - (**Function Block Diagram (FBD)**) **Continuous Function Chart (CFC)**, Texto estructurado - (**Structured Text (ST)**) y listado de instrucciones - (**Instruction List (IL)**).

## ¿Qué es un POU?

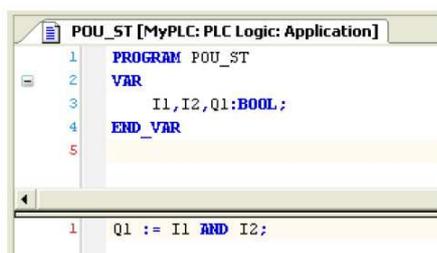
La unidad organizativa de Programa llamada por el acrónimo ( POU ), se utiliza para todos los objetos de programación (programas , bloques de función , funciones, etc.) que se utilizan para crear una aplicación en el controlador.

Un POU ue se gestionan en el nodo Global del árbol Las aplicaciones no son específicos del dispositivo, sino que se pueden crear instancias para el uso en un dispositivo (aplicación). El programa POU debe ser llamado por una tarea de la aplicación respectiva.

Por lo tanto, un objeto de POU en general es una unidad de programación.

## POU ST

**ST (Texto estructurado)** es similar a la programación en PASCAL o C.



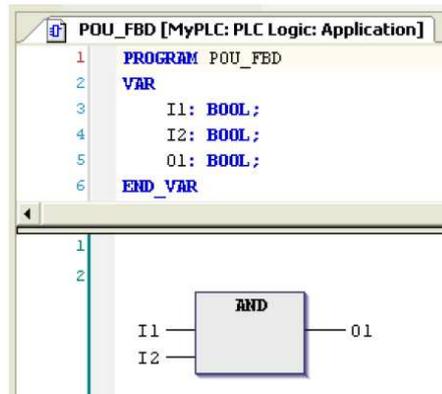
```
1 PROGRAM POU_ST
2 VAR
3 I1, I2, Q1: BOOL;
4 END_VAR
5
1 Q1 := I1 AND I2;
```



**Nota:** *Se pueden hacer una declaración múltiple de diferentes variables del mismo tipo de datos en la misma línea.*

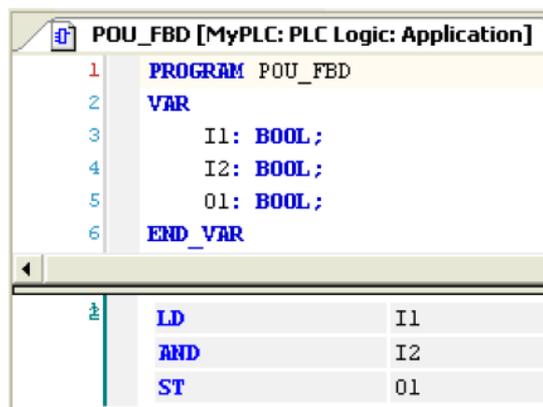
## POU FBD

FBD (Diagrama de bloques de función) es un lenguaje de programación de segmentos muy gráfico. Las instrucciones están colocadas en segmentos.



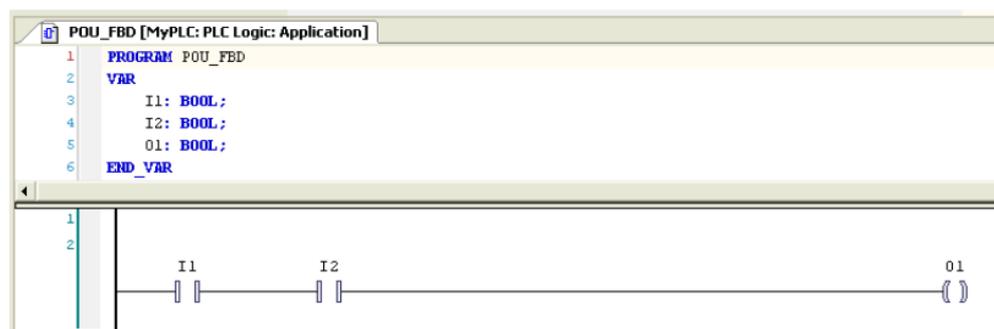
## POU IL

IL (Listado de instrucciones) es muy parecido al código máquina o lenguaje Assembler.



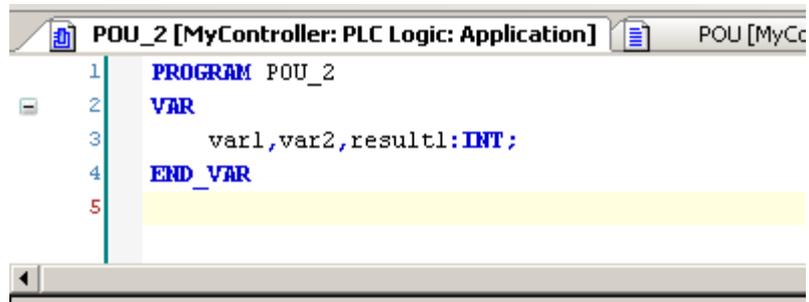
## POU LD

LD (Diagrama de contactos) permite al programador utilizar una combinación de contactos de relés y bobinas.



## POU CFC

El lenguaje de programación CFC (Continuous Function Chart) es similar al FBD. La única diferencia es que los elementos se pueden colocar dentro del área en el sitio que el usuario desee.



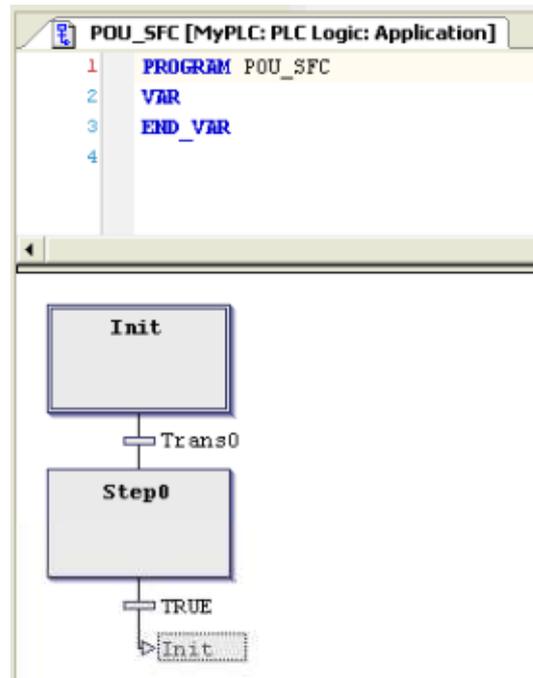
```
1 PROGRAM POU_2
2 VAR
3   var1,var2,result1:INT;
4 END_VAR
5
```



*Nota: Las variables locales se tienen que declara en área de declaración de variables en la sección 'VAR'.*

## POU SFC

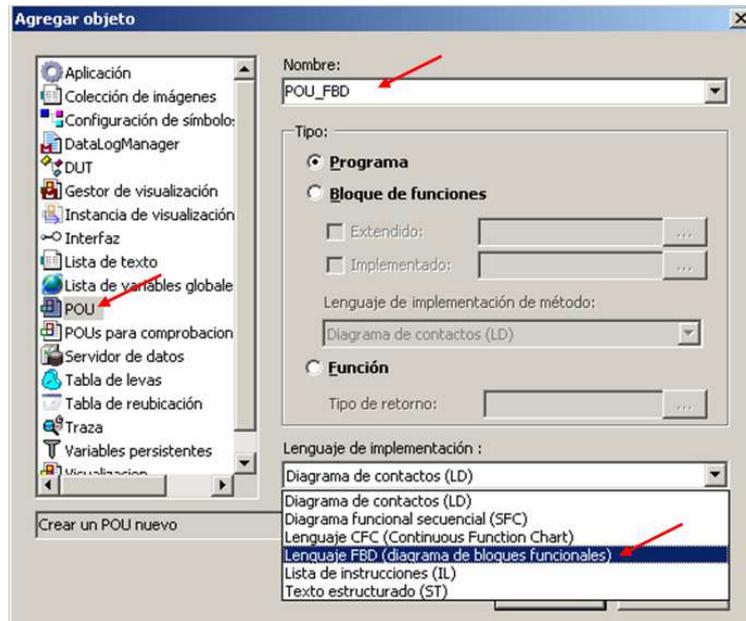
El lenguaje de programación **SFC (Sequential función chart)** se usa para programas procesos secuenciales.



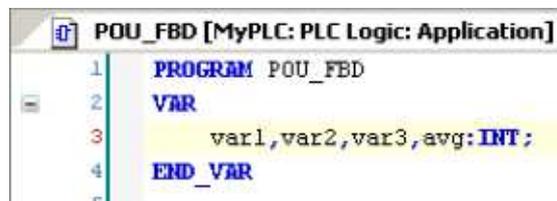
## Ejercicio – Programar un POU en FBD

### 3 Crear un programa utilizando el lenguaje FBD para calcular la media de tres variables.

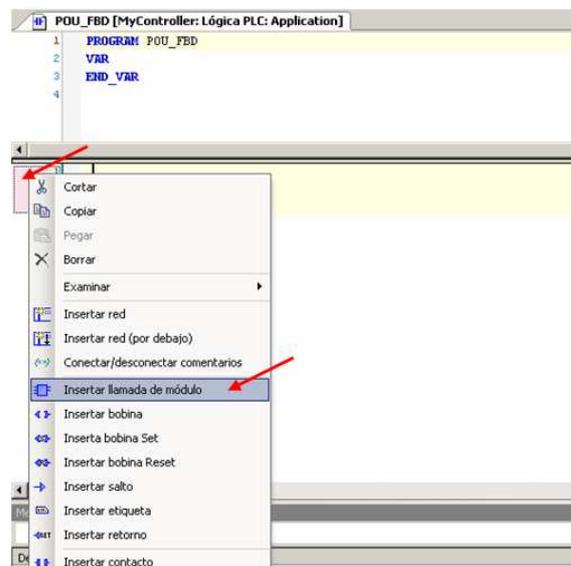
En la ventana de dispositivos añadir un nuevo POU, eligiendo el 'Lenguaje de Implementación' FBD. Poner como nombre **POU\_FBD**.



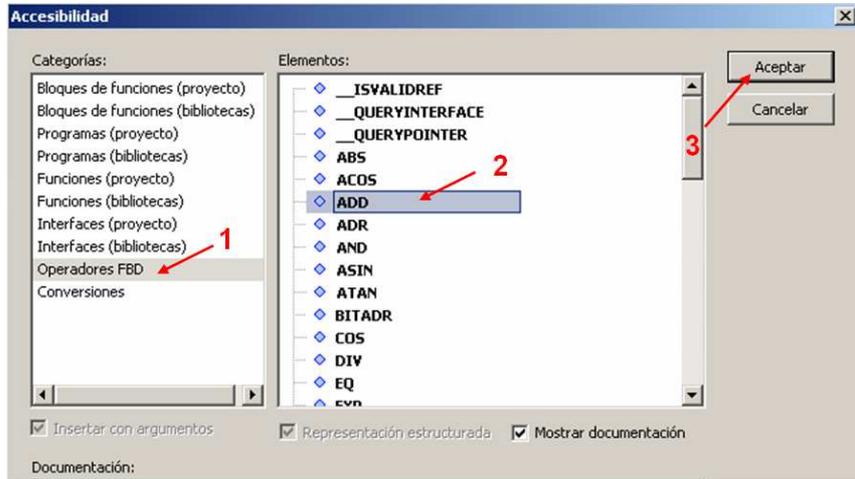
Dentro del área de trabajo en la parte de arriba 'Área de declaración de variables' crear las cuatro variables del tipo entero (INT) que se utilizarán para este ejercicio.



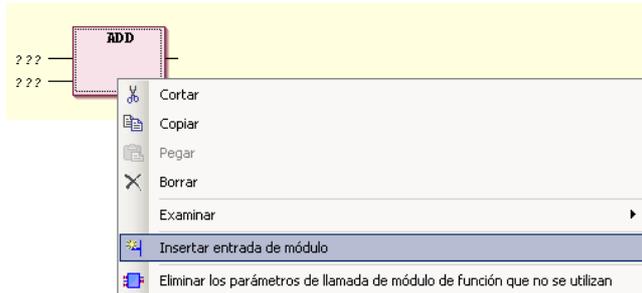
Seleccionar en la parte de la izquierda la primera red 1 del área de programación y hacer clic del botón derecho del ratón y seleccionar 'Insertar llamada a módulo', para añadir una instrucción al programa.



Cuando aparezca la ventana de Asistente, seleccionar **'Operadores FDB'** del campo de la izquierda **'Categorías'**, en la parte de la derecha **'Elementos'** aparecen las diferentes instrucciones, seleccionar la instrucción **'ADD'** pulsar el botón **'Aceptar'** para introducirlo.



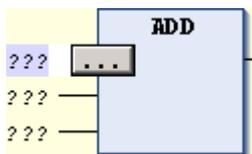
La instrucción suma **'ADD'** aparece en el área de programación. Seleccionar el bloque **ADD** y con hacer clic en el botón derecho y seleccionar **'Insertar entrada a módulo'** para añadir a la instrucción un pin de entrada para que realice la suma de tres elementos.



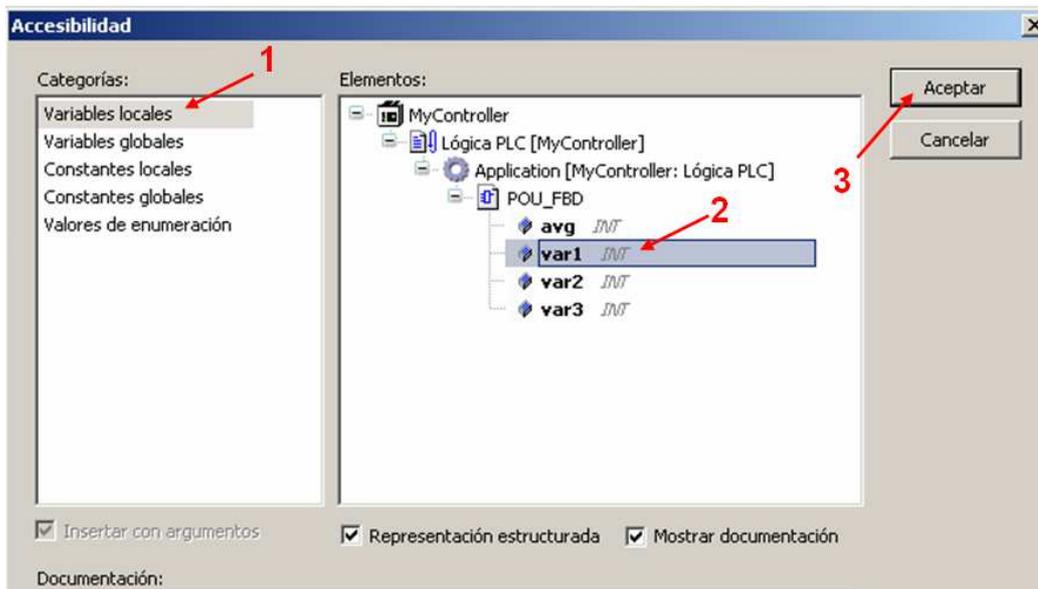
Una tercer pin de entrada aparecerá en la instrucción **ADD**.



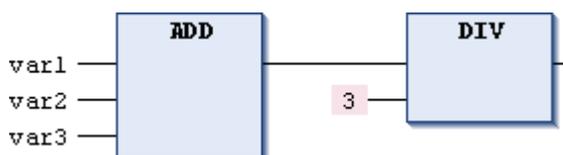
Hacer clic en el primer pin de entrada y hacer clic en el icono de los tres puntos que aparece en el campo **...**.



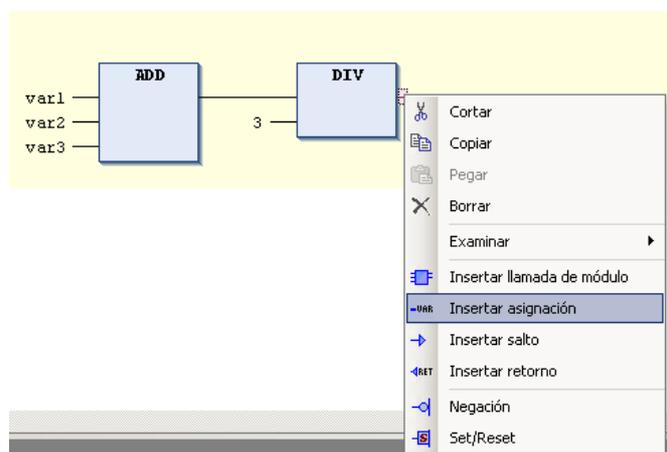
Al pulsar [...] aparece la ventana de asistente de entradas. En el campo 'Categorías' seleccionar 'Variables Locales'. En el campo de la derecha 'Elementos' desplegar el árbol que sale de 'My Controller' y seleccionar la 'Var1' y pulsar el botón de 'Aceptar'. Hacer lo mismo con el resto de pines.



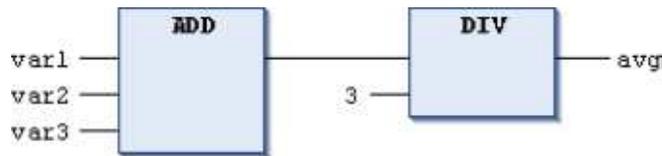
Seleccionando el pin de salida de la instrucción 'ADD', con el botón derecho selecciona del menú desplegable la opción 'Insertar entrada a módulo'. Seleccionar **FDB Operators** del campo 'Categorías' y en el campo 'Elementos' seleccionar la instrucción 'DIV' y confirmar. Añadir un 3 en el pin de abajo de la instrucción.



Seleccionar el pin de salida de la instrucción DIV y con el botón derecho seleccionar la opción 'Insertar Asignación'.



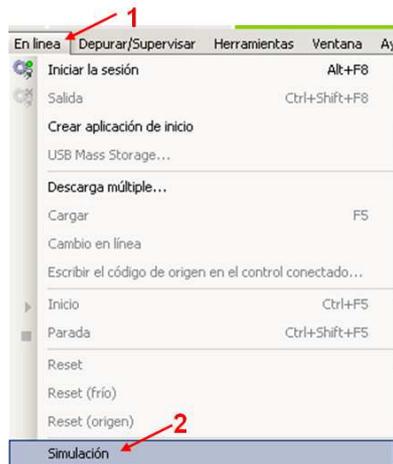
Usar el 'Asistente de entrada' , para añadir la variable 'avg' como resultado de la división. El programa completo tiene que tener este aspecto:



#### 4 Comprobar el programa.

Añadir el nuevo POU a la tarea MAST.

En el Menú principal, seleccionar 'En línea' y activar la 'Simulación'.



Compilar la aplicación pulsando el icono  de la barra superior.

Conectarse pulsando el icono .

Poner en RUN la aplicación pulsando el icono .

Introducir los valores de las variables y comprobar que se realiza la media correctamente. Cambiando el valor de las entradas en el campo 'Valor preparado' y escribiéndolo con la opción 'Depurar/Supervisor / Escribir Valores' o pulsando la combinación de teclas 'CTRL + W'.

Expresión	Comentario	Tipo de datos	Valor	Valor preparado
var1		INT	12	34
var2		INT	37	45
var3		INT	56	67
avg		INT	35	

```

    graph LR
      var1[12] --- ADD[ADD]
      var2[37] --- ADD
      var3[56] --- ADD
      ADD --- DIV[DIV]
      3 --- DIV
      DIV --- avg[35]
  
```

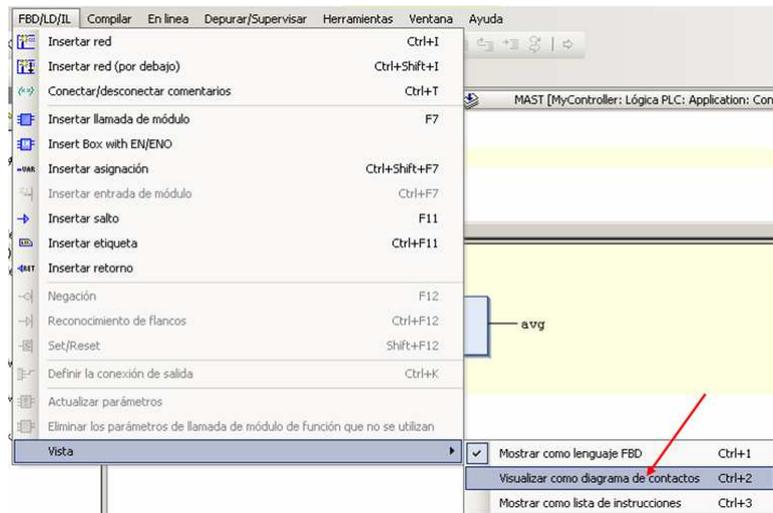
Seleccionar en el barra de menú **En línea » Salida** para desconectarse.

## Ejercicio – Convertir un programa FBD a IL ó LD

### 5 Convertir el programa FBD del ejercicio anterior en lenguaje LD.

Carga el ejercicio anterior y seleccionar el POU\_FBD que habíamos creado previamente.

En el menú principal, seleccionar FBD/LD/IL » Vista » Visualizar como diagrama de contactos para cambiar el POU creado en lenguaje FBD en lenguaje LD.



Volver a seleccionar FBD/LD/IL » Vista » Mostrar como lista de instrucciones para cambiar el POU a listado de instrucciones IL.

```
1 PROGRAM POU_FBD
2 VAR
3   var1, var2, var3, avg: int;
4 END_VAR
5
```

LD	var1
ADD	var2
	var3
DIV	3
ST	avg



**Nota: Los lenguajes FBD, LD e IL se pueden convertir entre ellos. Los lenguajes SFC y CFC no se pueden convertir.**

# Watchdog

## Operación Watchdog

El **Watchdog** es una protección del sistema que se utiliza para proteger el equipo de errores de hardware y software que pueden provocar que el sistema pueda pararse.

Hay dos tipos de **Watchdog** en el SoMachine:

- **Watchdog de la Aplicación (configurable)** – Cada tiempo de ciclo se puede monitorizar y comparar con el tiempo de watchdog configurado (que es el tiempo máximo de ciclo que puede tardar la tarea en ejecutarse). Este ayuda a depurar la aplicación para que no se hayan programados bucles infinitos, etc.

El watchdog puede ser definido para cada tarea.

- **Watchdog de sistema** – El sistema se puede sobrecargar cuando las áreas creadas por el usuario usan más del 80% de los recursos del sistema.

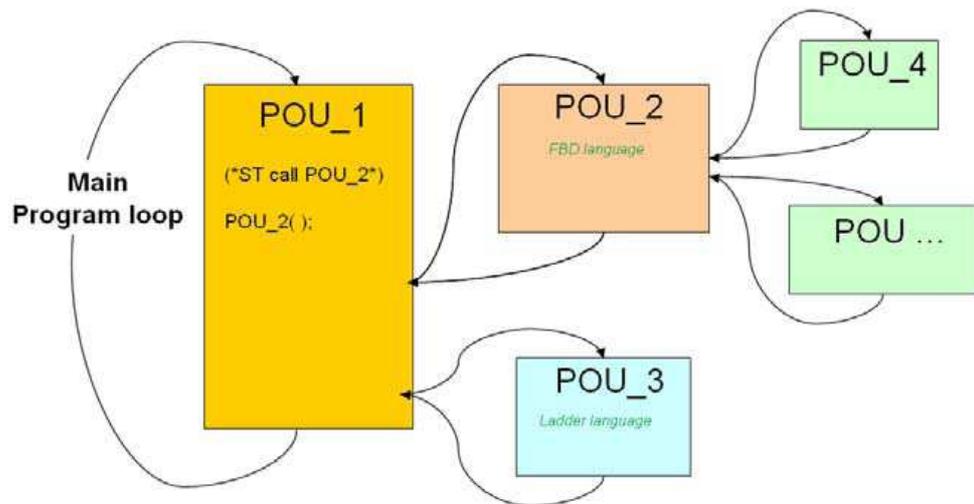
Este watchdog de sistema no se puede desactivar.

Properties		Monitor				
Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)
 MAST	Valid	21051	24716	14	15	457

- Estando en modo **Online**, en la ventana de dispositivos hacer doble clic en el '**Configurador de tareas**' y seleccionar la pestaña '**Supervisión**'.
  - Monitorizar los tiempos de ejecución de las diferentes tareas que están en ejecución.
  - Ver el tiempo de ejecución medio de una tarea, ayuda a parametrizar correctamente el tiempo de watchdog de la misma.

## Estructura de programa

### Estructura de programa



Un POU puede llamar a otros POU's (anidamiento)

- No hay limitaciones del número de llamadas
- Pero puede afectar al Watchdog.

La estructuración de programa es posible.

- Se puede añadir una condición para la ejecución de un POU.
- Se puede llamar a POU's programados en cualquier lenguaje.

Los tipos de llamadas a los POU en la aplicación:

	POU Function	POU Function Block	POU Program
Example	Function Fun1:INT 3 Inputs (INT): A, B, C	Function_Block FunBlck1 3 Inputs (INT): A, B, C 2 Outputs (INT): D, E Instance1: FunBlck1	Program Prgr1 3 Inputs (INT): A, B, C 2 Outputs (INT): D, E
List	LD 5 Fun1 3,2 ST Result	CAL Instance1(A:=5, B:=3, C:=2) ... LD Instance1.D ST Result1 LD Instance1.E ST Result2	CAL Prgr1(a := 5, b := 3, c := 2) ... LD Prgr1.D ST Result1 LD Prgr1.E ST Result2
Structured text	Result:=Fun1(5,3,2); or Result:=Fun1(A:=5,B:=3,C:=2);	Instance1(A:=5, B:=3, C:=2, D=> Result1, E=> Result2); or Result1:=Instance1.D; ...	Prgr1(A := 5, B := 3, C := 2, D=> Result1, E=> Result2); or Result1:= Prgr1.D; ...
Ladder or FBD or CFC			

## POU tipo Función

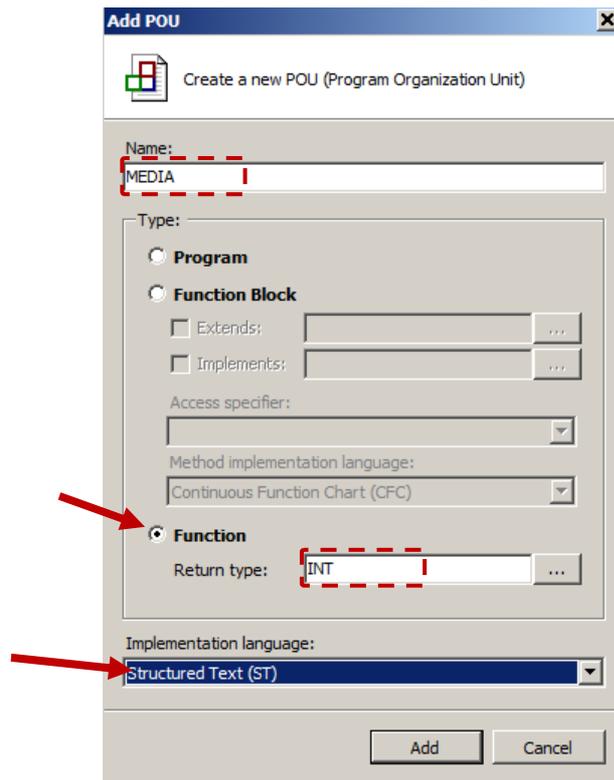
Un POU que devuelve un resultado

El POU tipo función – es un POU que devuelve un solo resultado.

- Normalmente usado cuando se necesita repetir el mismo cálculo con diferentes variables de entrada.
- El resultado se devuelve en la misma llamada de la función.
- La función solo puede devolver un valor de salida.

Creación de una función

Ejemplo – Creación de una función que hace la media aritmética



La función de media aritmética se ha creado en lenguaje ST.

```
1 FUNCTION MEDIA : INT (* El propio nombre de la función devuelve el resultado *)
2 VAR_INPUT (*Declaración de la variables de entrada a la función*)
3   var1:INT; (*Nombre de la variable y tipo de dato*)
4   var2:INT;
5   var3:INT;
6 END_VAR
7 VAR (*Declaración de las variables internas de la función*)
8 END_VAR
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Llamada de una función

La llamada de una función desde un POU programa que esté llamado en una tarea (La función no se puede asignar a una tarea, sólo los POU del tipo programa).

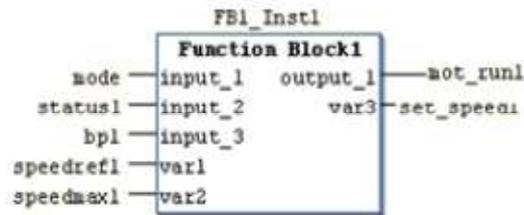
## POU tipo bloque de funciones

### Bloque de funciones

Bloque de funciones –

- Usado cuando el usuario quiere crear sus bloques de función.
- Muy útil cuando se tiene que programar el mismo código en diferentes partes del programa.
- Puede tener una o más entradas y salidas.
- La llamada a la instancia del bloque de función crea una estructura de datos asociado a este FB.

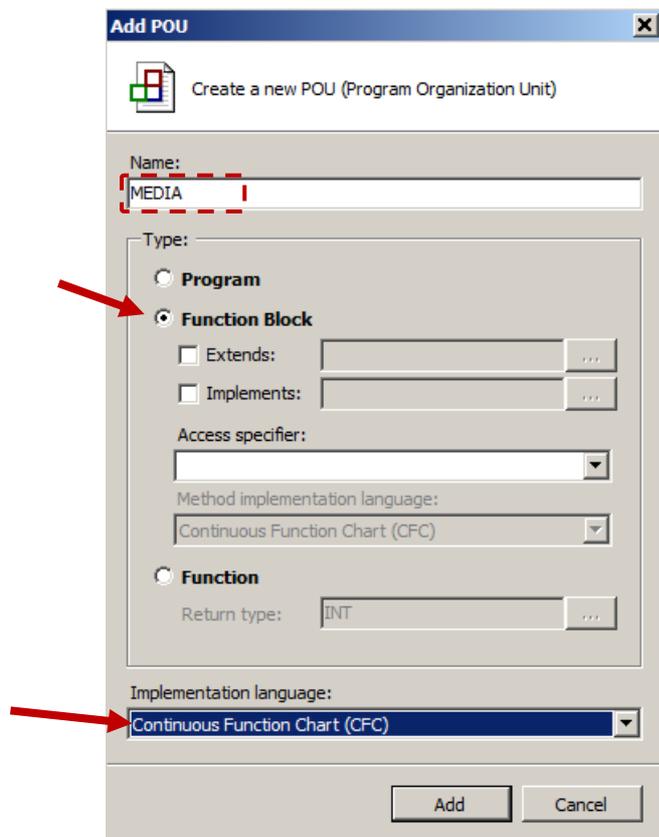
Bloque de función típico en lenguaje FBD.



### Como crear un bloque de funciones

#### ➤ Para crear un bloque de función:

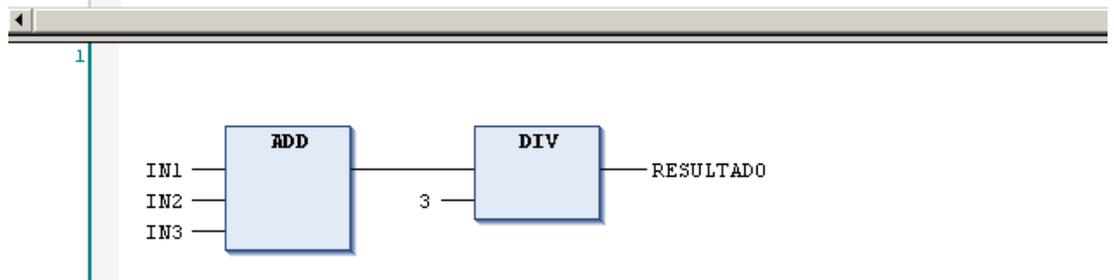
Al añadir el objeto POU, seleccionar el tipo bloque de funciones, ponerle un nombre y elegir el tipo de lenguaje de programación con el que se programará dicho bloque.



## Declaración del FB

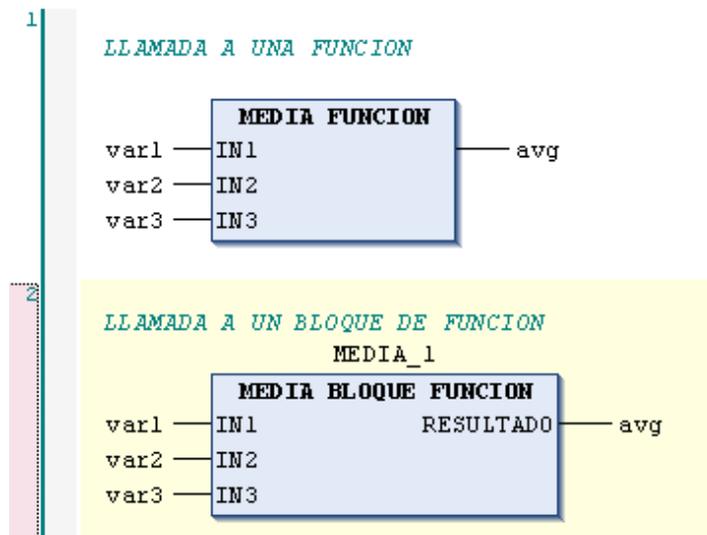
La declaración de las variables en el bloque de funciones, es diferente al de la función, tal como se muestra a continuación.

```
1  FUNCTION_BLOCK MEDIA (*Nombre del bloque de función*)
2  VAR_INPUT  (*Declaración de las variables de entrada al bloque de función*)
3      IN1:INT; (*Nombre y tipo de datos entrada al bloque de función*)
4      IN2:INT;
5      IN3:INT;
6  END_VAR
7  VAR_OUTPUT (*Declaración de las variables de salida del bloque de función*)
8      RESULTADO:INT; (*Nombre y tipo de datos de la salida del bloque de función*)
9  END_VAR
10 VAR
11 END_VAR
12
```



La diferencia con la función, se ve en la declaración de variables el FB tiene un campo de declaración de variables de salida (Se puede tener una o más variables de salida, y la función solo una).

La diferencias entre la llamada de una función en un POU y un bloque de funciones.

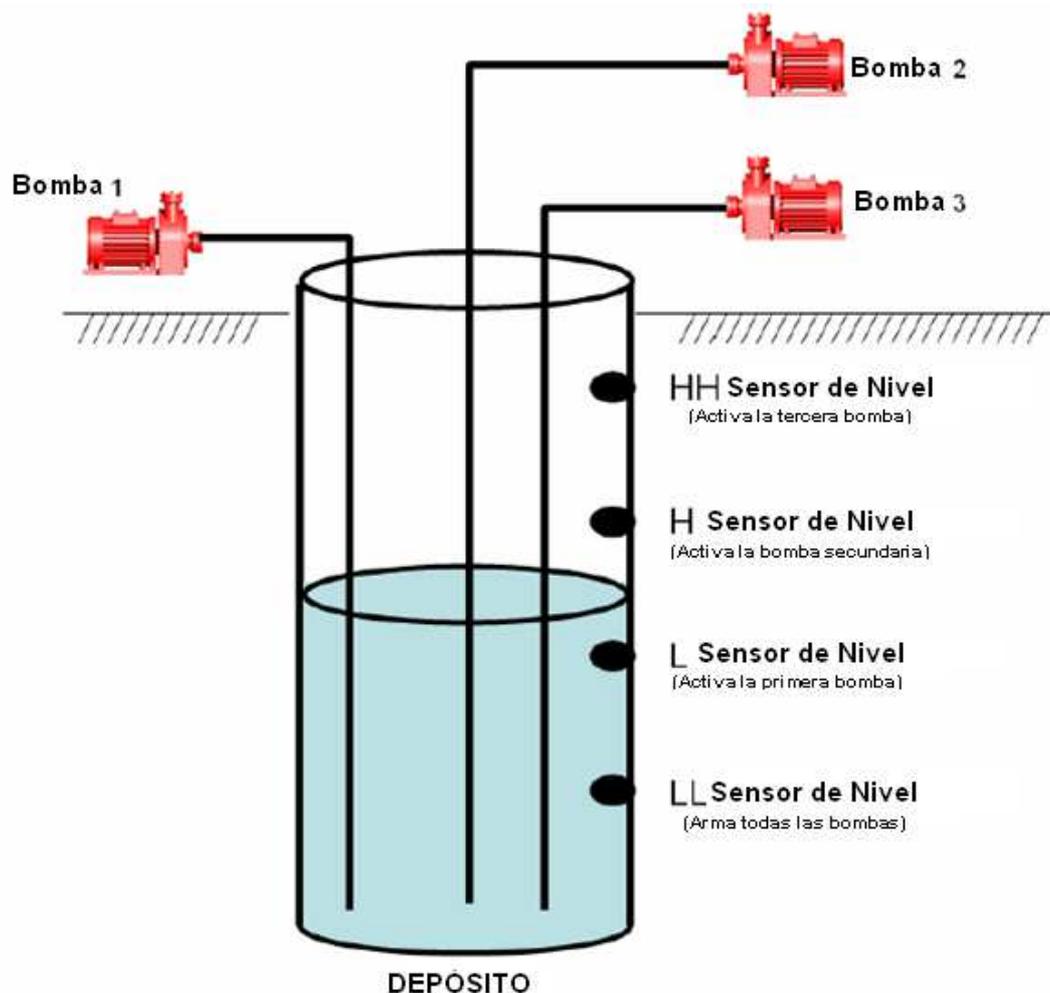


## Ejercicio – Crear un bloque de funciones

### Control de un Bombeo

Las bombas son un objeto muy común en la mayoría de las aplicaciones industriales. En esta primera práctica se tratará de realizar el programa de control de una bomba utilizando todo lo visto hasta ahora en este manual.

El sistema de riego se basa en control de tres bombas de riego.



### Operación

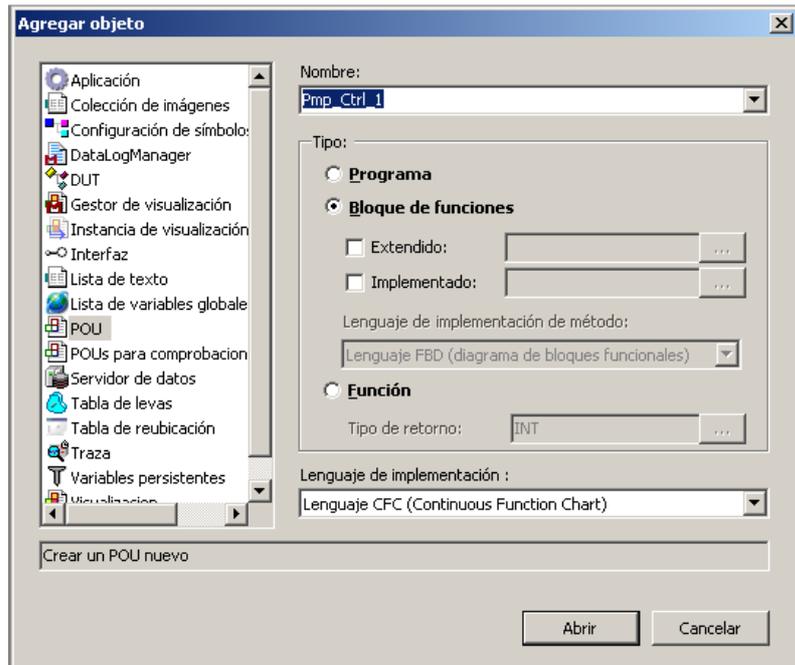
Cuando el nivel de agua active el sensor de nivel LL, quedan listas para ser activadas. Cuando el nivel de agua es mayor que el nivel L, se encenderá la bomba primaria. Si se activa el sensor de nivel H la bomba secundaria también tiene que ponerse a bombear. Si el nivel de agua continúa creciendo y llega al sensor HH, la tercera bomba también se activará.

Si el nivel del agua decrece y el sensor de nivel HH deja de dar señal, ninguna bomba tiene que parar. Cuando el nivel del agua baja del sensor H se para la tercera bomba. Si el nivel sigue bajando y se pierde la señal de nivel L se para la bomba secundaria y finalmente la bomba primaria se parará cuando la señal del sensor LL dejé de dar señal.

**6 Crear a nuevo proyecto, crear un bloque de función y añadir este al programa.**

Crear un **Proyecto nuevo** y con el nombre **Water Proyecto**. En la ventana de 'Inicio' hacer clic en 'Empezar con un proyecto vacío'. Añadir el modelo de PLC que se está usando en la ventana de configuración.

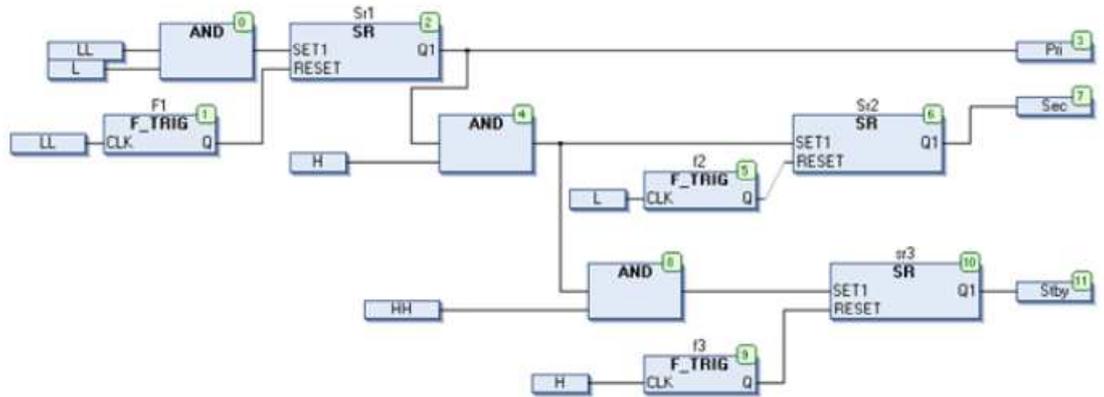
En la ventana de 'Programa' dentro de la aplicación. Añadir un nuevo **POU Function Block** y llamarlo **Pmp\_Ctrl\_1** y elegir el lenguaje de programación **CFC**.



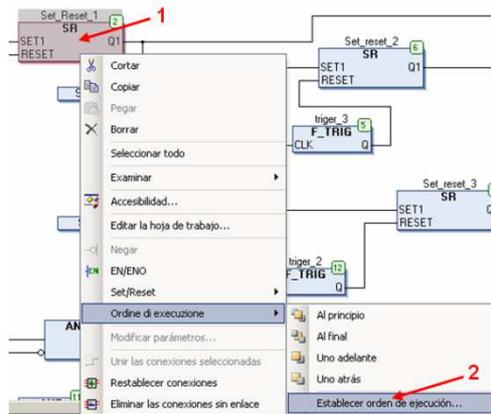
Declarar las siguientes variables dentro del bloque de función.

```
1  FUNCTION_BLOCK Pump_Ctrl_1
2  VAR_INPUT
3      LL, L, H, HH:BOOL;
4  END_VAR
5  VAR_OUTPUT
6      Pri, Sec, Stby:BOOL;
7  END_VAR
```

Programar el bloque de función con el lenguaje CFC siguiendo la secuencia de operación del control del bombeo.



Observar el orden de ejecución de los bloques. El orden de ejecución de los bloques es el número que se encuentra en un recuadro en la parte superior derecha de cada bloque. Esta opción permite modificar el orden de ejecución.

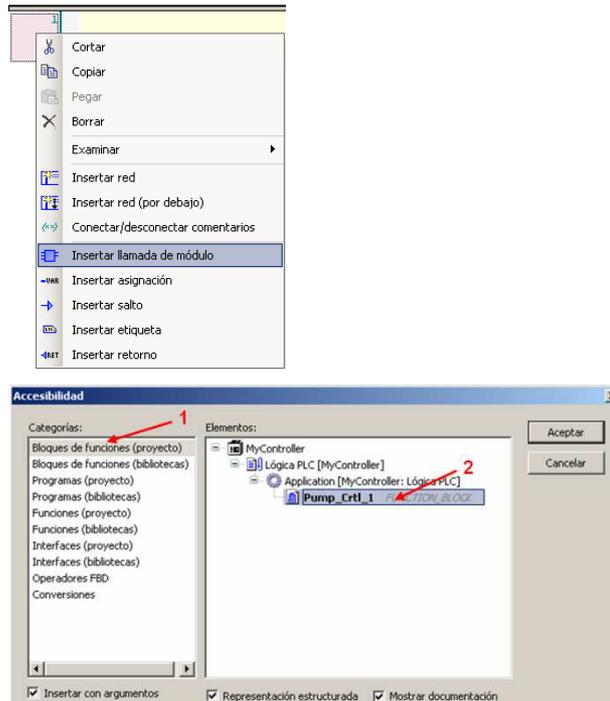


Asigna las variables a las entradas y salidas digitales físicas.

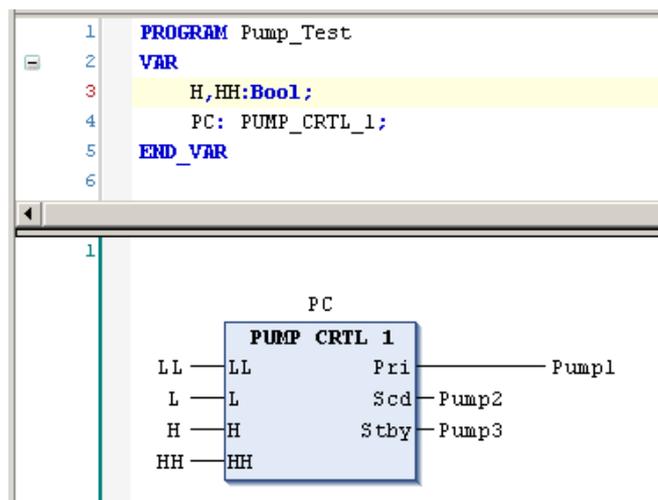
Variable	Asignación	Canal	Dirección	Tipo	Valor actual	Valor predeterminado	Unidad	Descripción
<b>Entradas</b>								
		IW0	%IW0	WORD				
		I0	%IX0.0	BOOL				Entrada ráp...
		I1	%IX0.1	BOOL				Entrada ráp...
LL		I2	%IX0.2	BOOL				Entrada ráp...
L		I3	%IX0.3	BOOL				Entrada ráp...
		I4	%IX0.4	BOOL				Entrada ráp...
		I5	%IX0.5	BOOL				Entrada ráp...
		I6	%IX0.6	BOOL				Entrada ráp...
		I7	%IX0.7	BOOL				Entrada ráp...
		I8	%IX1.0	BOOL				
		I9	%IX1.1	BOOL				
		I10	%IX1.2	BOOL				
		I11	%IX1.3	BOOL				
		I12	%IX1.4	BOOL				
		I13	%IX1.5	BOOL				
<b>Salidas</b>								
		QW0	%QW0	WORD				
Pump1		Q0	%QX0.0	BOOL				Salida rápida
Pump2		Q1	%QX0.1	BOOL				Salida rápida
Pump3		Q2	%QX0.2	BOOL				Salida rápida
		Q3	%QX0.3	BOOL				Salida rápida
		Q4	%QX0.4	BOOL				
		Q5	%QX0.5	BOOL				
		Q6	%QX0.6	BOOL				

Crear un **nuevo POU** tipo programa y llamarlo **Pump\_Test** en lenguaje de programación **FBD**.

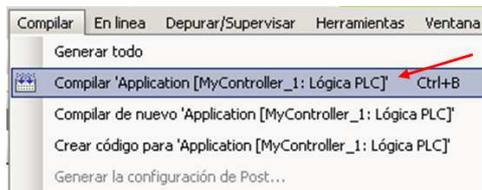
Añadir el bloque de funciones **Pmp\_Ctrl\_1** al POU de programa Pump\_test. Seleccionar la red y con el botón derecho elegir 'Añadir módulo' y en el asistente buscar el bloque de función en 'Bloques de función (Proyecto)'



Tras linkar las variables el programa de control del bombeo quedará como a continuación.



Añadir el Pump\_Test a la tarea MAST que está en el 'Configurador de Tareas'. Cuando esté añadido 'Compilar' el programa y descargarlo.



**Testear**, utilizando el simulador si la aplicación funciona correctamente, según el modo de operación descrito.

# Capítulo 6: Tipos de datos

## Descripción

**Introducción** Este CAPÍTULO describe los tipos de datos que disponen el software SoMachine para crear las variables.

### Este CAPÍTULO trata los siguientes temas:

- Tipos de datos estándar.....6-2
- Reglas para nombrar variables .....6-3
- Variables Locales.....6-4
- Declaración de variables.....6-5
- Variables globales.....6-6
- Declaración de variables.....6-7
- Configuración de símbolos.....6-8
- Tipos de datos del usuario (DUT) .....6-8
- Matrices .....6-11
- Variables con dirección física.....6-12
- Editor I/O.....6-13

## Tipos de datos estándar

### Tipos de datos de CoDeSys

CoDeSys contiene los siguientes tipos de datos:

- Tipos de datos del estándar IEC 61131-3
- Tipos de datos definidos por el usuario
- Extensiones de la IEC 61131-3

### IEC 61131-3

Los Tipos de datos del estándar IEC 61131-3 son:

Keyword	Prefix ***	Size (Bit)	Range	Example
<b>BOOL</b>	x	8	0, 1	FALSE, TRUE, 0, 1
<b>SINT</b>	si	8	-128 .. 127	0
<b>INT</b>	i	16	-32.768 .. 32.767	24453
<b>DINT</b>	di	32	-2.147.483.648 .. 2.147.483.647	-38099887
<b>LINT</b>	li	64	$-2^{63} .. 2^{63}-1$	
<b>USINT</b>	usi	8	0 .. 255	200
<b>UINT</b>	ui	16	0 .. 65.535	47453
<b>UDINT</b>	udi	32	0 .. 4.294.967.295	138099887
<b>ULINT</b>	uli	64	$0 .. 2^{64}$	
<b>BYTE</b>	by	8	0 .. 255	8450, 16#EA3F,
<b>WORD</b>	w	16	0 .. 65.535	2#11_0011_1010
<b>DWORD</b>	dw	32	0 .. 4294967295	
<b>LWORD</b>	lw	64	$0 .. 2^{64}-1$	
<b>REAL</b>	r	32	$-1.2 \times 10^{-38} .. 3.4 \times 10^{38}$	1.34996
<b>TIME</b>	tim	32	0 ms .. 1193h2m47s295ms	T#1d8h12m8s125ms
<b>LTIME*</b>	ltim	64	0	LTIME#10d12h13ns
<b>TOD</b>	tod	32	ns..213503d23h34m33s709ms551us	TOD#12:34:17
<b>DATE</b>	date	32	615ns	D#2001-03-15
<b>DT</b>	dt	64	00:00:00 .. 23:59:59 01.01.1970 bis ca. 06.02.2106	DT#2001-03-15-12:17:03
<b>STRING(xx)**</b>	s		0 .. 255 Characters	'Hello world'
<b>WSTRING(xx)**</b>	ws		0 .. 32767 Characters	

## Reglas para nombrar variables

### Reglas para la declaración de variables

Las reglas principales para ponerle un nombre a una variable, son las siguientes:

- No tienen que contener espacios en blanco, ni caracteres especiales.
- No diferencia mayúsculas de minúsculas. Lo cual significa que “**VAR1**”, “**Var1**” y “**var1**” será la misma variable.
- El carácter de subguión “\_” si que es diferenciador (ejemplo, “**A\_BCD**” y “**AB\_CD**” se consideran dos variables diferentes), pero un nombre no tiene que tener más de un subguión en su raíz.
- La longitud del nombre no tiene limitaciones.
- Un nombre de variable no puede ser idéntico a ningún comando. (Ejemplo: No se puede tener ninguna variable con el nombre “**TRUE**”).
- Una variable global puede se puede declarar con el mismo nombre que una variable local. En éste, caso la variable local tiene prioridad.



#### Ver También:

Para más información sobre las reglas de declaración de variables, ver la *Ayuda de CoDeSys – Declaración de Variables*.

### Notación Húngara

A la hora de ponerle nombre a las variables en aplicaciones y librerías es muy posible que se haya utilizado la **Notación Húngara** que es unas recomendaciones a la hora de ponerle un nombre a una variable.

Para cada variable lo más significativo, es una pequeña descripción de la variable, para la base del nombre. La primera letra de esta base tiene que ir en mayúscula. (Ejemplo: **FileSize**).

Antes del la base, se pondrá un prefijo en letras minúsculas que corresponde a al tipo de dato que es la variable.

#### Ejemplos:

- bySubIndex: BYTE;
- sFileName: STRING;
- udiCounter: UDINT;



#### Ver También:

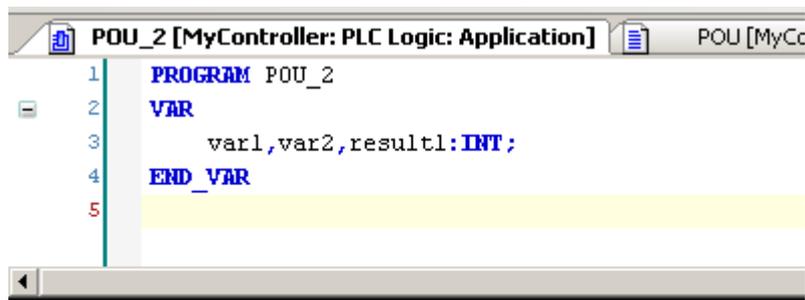
Para más información ver *Ayuda de CoDeSys - Recommendations on the naming of identifiers*.

## Variables locales

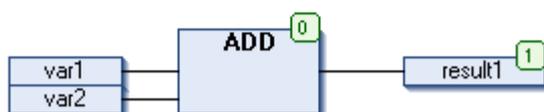
### Zona de declaración

Las **variables locales** se declaran en la cabeceras de los POU's en las área de declaración de variables en la parte superior del área de trabajo.

En este ejemplo, hay la declaración de una entrada y una salida para utilizarla en la función.



```
1 PROGRAM POU_2
2 VAR
3   var1,var2,result1:INT;
4 END_VAR
```



La declaración de variables tiene que respetar esta sintaxis:

**< Identificador > : < Tipo de dato > { :=valor de inicialización (opcional)>;**

La declaración múltiple es posible siempre que las variables sean del mismo tipo de datos:

**Ejemplo: abc, def, efg: INT;**

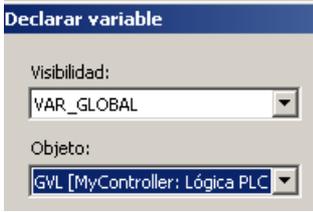
Las variables se pueden declarar e inicializar al mismo tiempo.

**Ejemplo: abcd: INT:= 45;**

## Ventana de Declarar variable

### Ventana de declaración de variables

Cuando en el programa se pone un nombre a una variable que no ha sido declarada con anterioridad, el software automáticamente abre la ventana flotante para definir esa variable.

Campo	Descripción
<b>Visibilidad</b>	Selecciona el tipo de variable (ejemplo: VAR_GLOBAL, VAR significa variable local)
<b>Objeto</b>	Aparece el nombre del POU donde está siendo editada la variable. Si se quiere declarar una variable global (Visibilidad: VAR_GLOBAL), tienes que seleccionar en visibilidad: Var_GLOBAL y en ' <b>Objeto</b> ' elegir GVL del área de dispositivos. 
<b>Indicadores</b>	Activar el indicador deseado si se quiere definir la variable como una constante o un variable remanente.
<b>Nombre</b>	El nombre que va a tener la variable.
<b>Valor inicial</b>	Si se desea que la variable se inicialice con un valor determinado. Si no se pone nada la variable se inicializará con su valor por defecto.
<b>Tipo de dato</b>	Tipo de dato de la variable.
<b>Dirección</b>	Si se quiere direccionar la variable hay que indicar la dirección de memoria física (Ejemplo: %MW200)
<b>Comentario</b>	Opcional poner un comentario descriptivo de la variable.

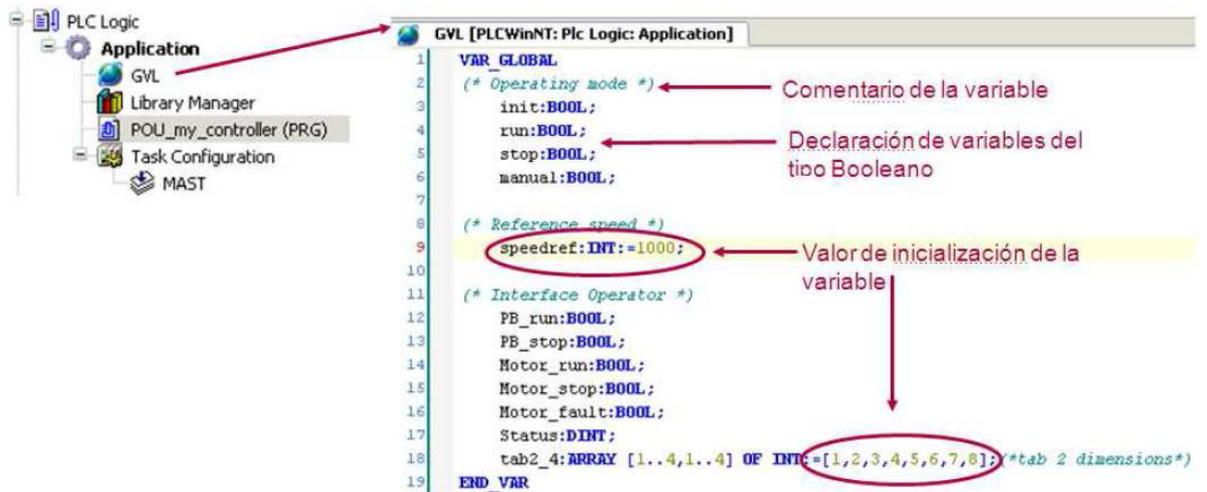
## Variables globales

### Lista de variables globales (GVL)

La lista de variables globales (**GVL**) no se abre en el área de trabajo una zona donde se pueden declarar las variables globales que son aquellas que se pueden utilizar en toda la aplicación. Se pueden crear un máximo de tres listas de variables globales por aplicación.



**Nota:** Las Variables Globales se declaran de la misma manera que las variables locales pero dentro del objeto listado de variables globales 'GVL'.



## Declaración Variables

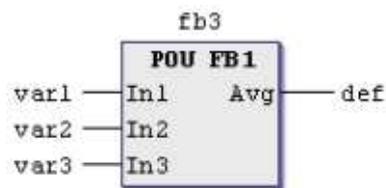
### Bloques de funciones

Cuando un usuario crea un bloque de función, es necesario definir las variables de entrada y salida del bloque de funciones FB.

En el bloque de funciones de abajo, hay tres entradas (**In1-In3**) y una salida (**Avg**).

```
1 FUNCTION_BLOCK POU_FB1
2 VAR_INPUT
3     In1, In2, In3: INT;
4 END_VAR
5 VAR_OUTPUT
6     Avg: INT;
7
8 Avg := (In1+In2+In3)/3;
```

Cuando el FB es llamado el programa, las variables aparecen en el lenguaje FBD como se muestra a continuación.



En este ejemplo las variables locales (var1, var2, var3) se asignan a los pines de entrada del FB y la variable (def) se asigna al pin de salida del FB. Hay que tener en cuenta que las variables que se asignen a las entradas y salidas del FB tienen que tener el mismo tipo de datos con el que se han declarado este tipo de entrada.

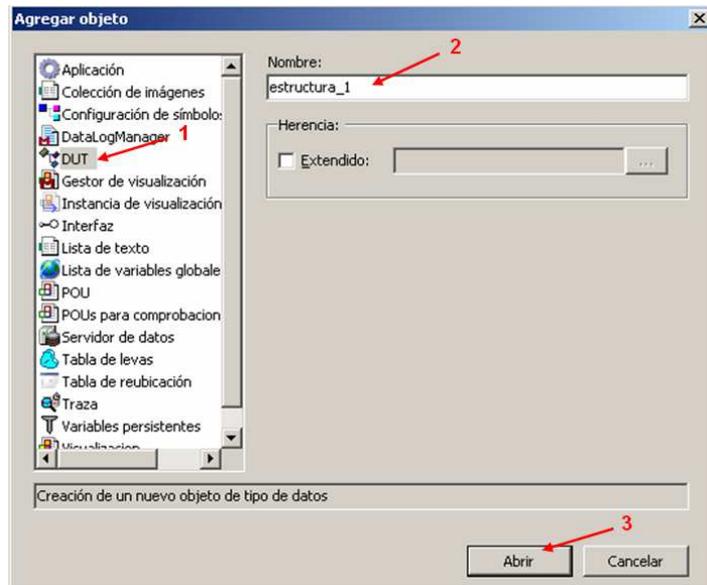
También hay que crear una variable que sea la llamada a la instancia de la FB en este caso **Fb3**.

## Tipos de datos definidos por el usuario (DUT)

### Estructura de datos

Una estructura de datos definida por el usuario, puede ser una agrupación de variables de tipos de datos diferentes o una agrupación de variables de tipos de datos iguales.

Para crear una estructura, hacer botón derecho en la aplicación y del menú desplegable seleccionar **'Añadir Objeto'**, aparecerá la ventana flotante **'Agregar Objeto'** en la barra de objetos seleccionar el ítem **'DUT'** (Data User Type) y en la parte izquierda poner el nombre a la estructura.



En la ventana de dispositivos aparecerá el icono con el nombre de la estructura para abrirla para configurarla hacer doble clic sobre el icono.



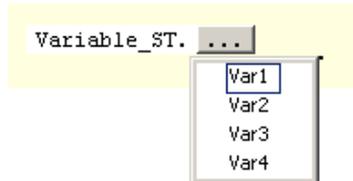
En el área de trabajo aparece la ventana para la declaración de variables de la estructura.

```
1  TYPE Estructura_1 :    (*Nombre de la estructura de tipos de datos*)
2  STRUCT                (*Inicio de la declaración de las variables de la estructura*)
3      Var1:STRING;
4      Var2:INT;         (*Declaración del nombre de la variable y del tipo de dato *)
5      Var3:BOOL;
6      Var4:REAL;
7  END_STRUCT
8  END_TYPE
9
```

Una vez creada la estructura de datos en el programa se puede crear variables con ese tipo de dato.

```
1 PROGRAM Pump_Test
2 VAR
3   Variable_ST: Estructura_1; (*Declaración de variable del tipo de datos Estructura_1*)
4 END_VAR
5
```

Para hacer la llamada a una de las variables internas de la estructura se tiene que poner el nombre de la variable y al poner un punto saldrá una ventana desplegable con las diferentes variables internas de la estructura.



**Nota:** La ventana desplegable con las variables internas de la estructura aparece al escribir “.”.

## Enumeraciones

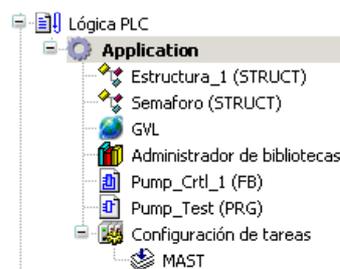
Una **Enumeración** es tipo de dato de usuario que asocia un número a constante de string. Estas constantes se definen como el valor de la enumeración. Los **valores de la Enumeración** son datos globales, incluido cuando se ha declarado en un POU determinado.

Este tipo de dato se crea como una “**DUT**” (Data Type Unit) añadiendo este objeto en la aplicación.

**Ejemplo:** La enumeración ‘**Semáforo**’ tiene los valores de enumeración ‘Rojo, Ámbar y Amarillo’ que tiene los valores 0,1 y 10 correspondientemente. Eliminar de la llamada de la estructura el ‘**STRUCT**’ y ‘**END\_STRUCT**’, para que la llamada sea correcta.

```
1 TYPE
2   Semaforo : (Rojo, Ambar, Verde:=10); (*El tipo INT de las variables está implícito*)
3 END_TYPE
4
```

Una vez creada la enumeración, el icono que se muestra en la ventana de dispositivos pone entre paréntesis (**ENUM**) para diferenciarlo de una estructura de datos.



Para utilizar la enumeración en el programa, se tiene que declarar una variable con este tipo de dato. Cuidado porque se tiene que declarar fuera del **VAR** ' y **VAR\_END**.

```

1  PROGRAM Pump_Test
2  Semaforo_1:=semaforo;
3  VAR
4  END_VAR
5

```

Una Enumeración es una herramienta sencilla para asociar a palabra de string a un valor numérico.

MyController.Application.Pump_Test				
Expresión	Comentario	Tipo de datos	Valor	Valor preparado
Semaforo_1		SEMAFORO	Semaforo.Ambar	
valor		INT	1	

valor 1 — Semaforo\_1 Semaforo.A

# Matrices

## Dimensiones de una matriz

Una-, Dos-, hasta tres dimensiones puedes dimensionar una matriz en el SoMachine como tipo de dato de usuario. Las matrices se tienen que definir en el área de declaración de variables del POU o en la lista global de variables (**GVL**).

## Declaración de una Matriz

Declaración de una matriz de una dimensión:

```
VAR  
  a: ARRAY[0..7] OF BOOL;  
END_VAR
```

Declaración de una matriz de dos dimensiones con 3 columnas y 9 filas.

1,1	2,1	3,1
1,2	2,2	3,2
1,3	2,3	3,3
1,4	2,4	3,4
1,5	2,5	3,5
1,6	2,6	3,6
1,7	2,7	3,7
1,8	2,8	3,8
1,9	2,9	3,9

### Declaración de la Matriz en el POU

```
Prg_2 [PLCWinNT: Plc Logic: Application]  
1 PROGRAM Prg_2  
2 VAR  
3   array1:ARRAY[1..3,1..9] OF WORD;  
4   result:WORD;  
5 END_VAR  
6
```

### Programa que carga el valor de una posición específica de la matriz, posición (columna 2, fila 8)

```
1 (* Access to the component column 2 and line 8 *)  
  result:=array1 [2,8];  
3
```

El valor de la variable 'result' será el valor guardado en la posición de la matriz 2,8.

## Variables con dirección física

### IEC Direccionamiento

El direccionamiento de una variable conforme con la IEC, es:

- **Sintaxis de una dirección:**

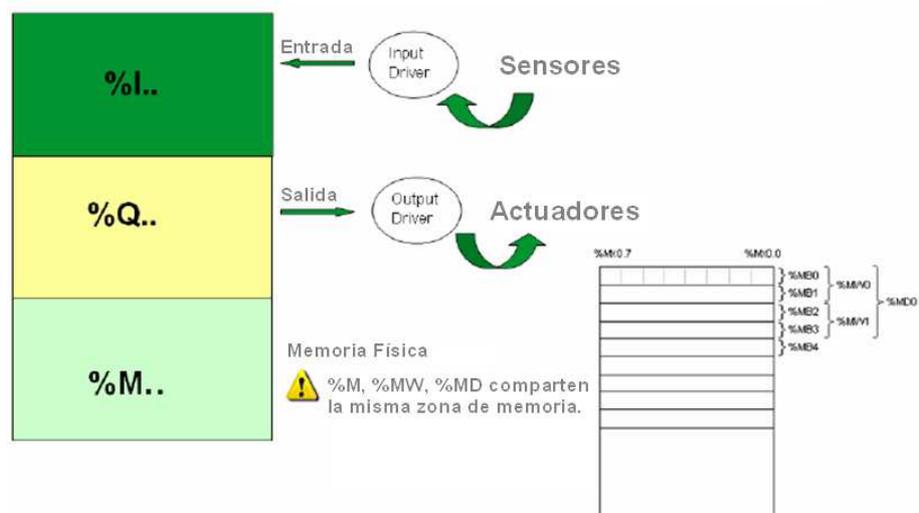
- Signo ‘%’
- Prefijo del área de memoria
  - I      Entrada
  - Q      Salida
  - M      Memoria
- Tamaño
  - X o Nada      Bit
  - B      Byte (8 Bits)
  - W      Word (16 Bits)
  - D      Double word (32 Bits)

#### Ejemplos

**%I W215** = Input word 215  
**%QX3.1** = Output bit 1 in byte 3  
**%MD48** = Double word memory 48

- Las direcciones %I y %Q se asocian a entradas y salidas físicas.
- Las marcas %M están direccionadas a la memoria física.
- Aunque se direccionen diferentes longitudes de datos en la memoria %M comparten la misma zona de memoria. Se tienen que tener en cuenta posibles solapamiento de datos.

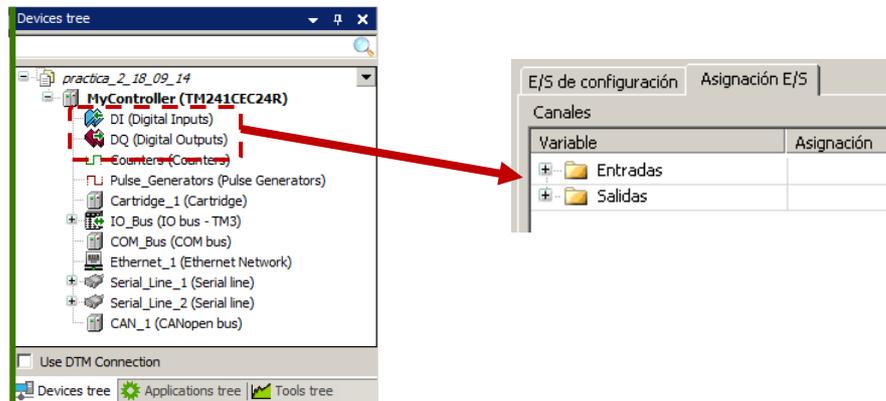
#### Tipos de direcciones – Entradas, Salidas y Memoria.



## Editor I/O

Abrir el editor I/O

Para abrir el editor I/O hacer doble clic en el icono IO (IO) en las Funciones incrustadas de la ventana de dispositivos.

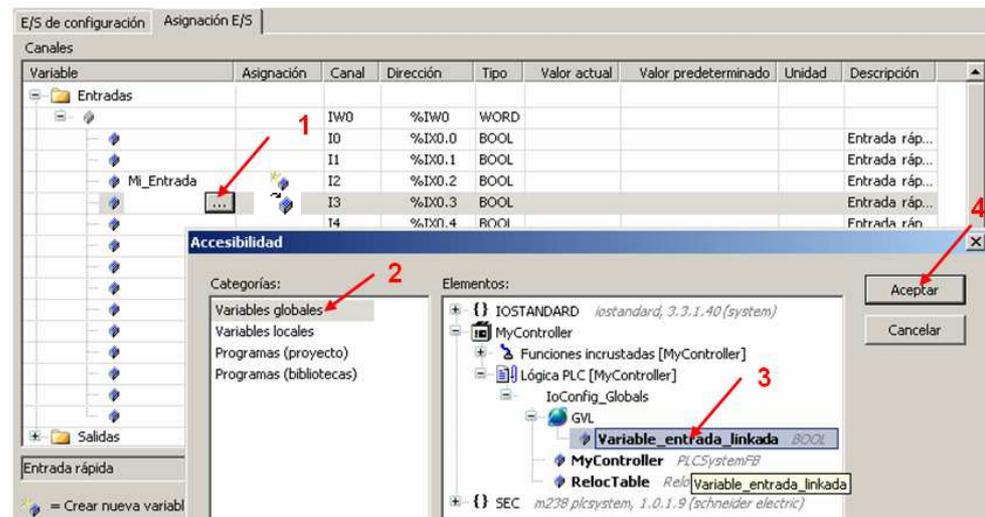


Si una variable se crea nueva, escribirla en el campo 'Asignación' de la entrada que le corresponde (Ejemplo. My\_entrada1).

Variable	Asignación	Canal	Dirección	Tipo	Valor actual	Valor predeterminado	Unidad	Descripción
		IW0	%IW0	WORD				
		I0	%IX0.0	BOOL				Entrada ráp...
		I1	%IX0.1	BOOL				Entrada ráp...
		I2	%IX0.2	BOOL				Entrada ráp...
		I3	%IX0.3	BOOL				Entrada ráp...
		I4	%IX0.4	BOOL				Entrada ráp...



Si se quiere linkar una entrada o salida física a una variable existente se usará el asistente para buscar la variable y seleccionarla.



**Nota:**

A las entradas y salidas físicas, solo se pueden linkar variables que estén declaradas como variables globales



# Capítulo 7: Visualizaciones

## Descripción

**Introducción** Este CAPÍTULO trata la creación de pantallas de visualización del SoMachine.

SoMachine incluye un editor gráfico que permite pantallas gráficas donde se puede mostrar las variables del proceso directamente. Estas visualizaciones son un estándar dentro del paquete CoDeSys.

Las Visualizaciones crear pantallas interactivas y dinámicas que permiten al usuario monitorizar, modificar o simplemente ver los datos en función de sus necesidades.

Además SoMachine tiene plantillas de visualizaciones que han sido desarrolladas para utilizar conjuntamente con FB's. Para ayudar a la puesta en marcha de diferentes equipos.

---

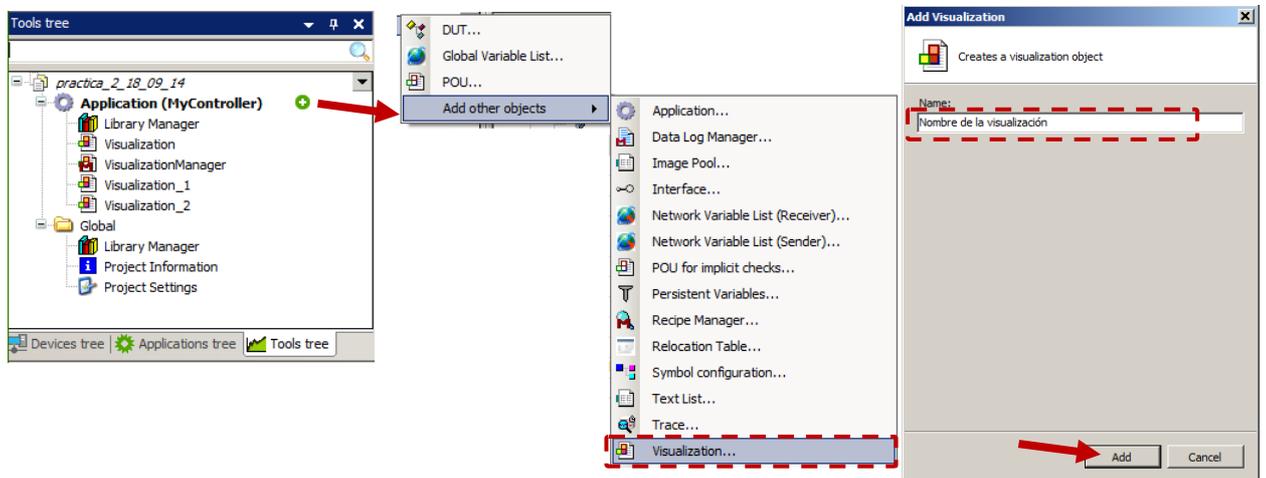
### Este CAPÍTULO trata los siguientes temas:

- Crear una Visualización .....7-2
- Variable de entrada.....7-5
- Reutilizar una Visualización .....7-6
- Ejercicio – Reutilizar una Visualización.....7-6

## Crear una Visualización

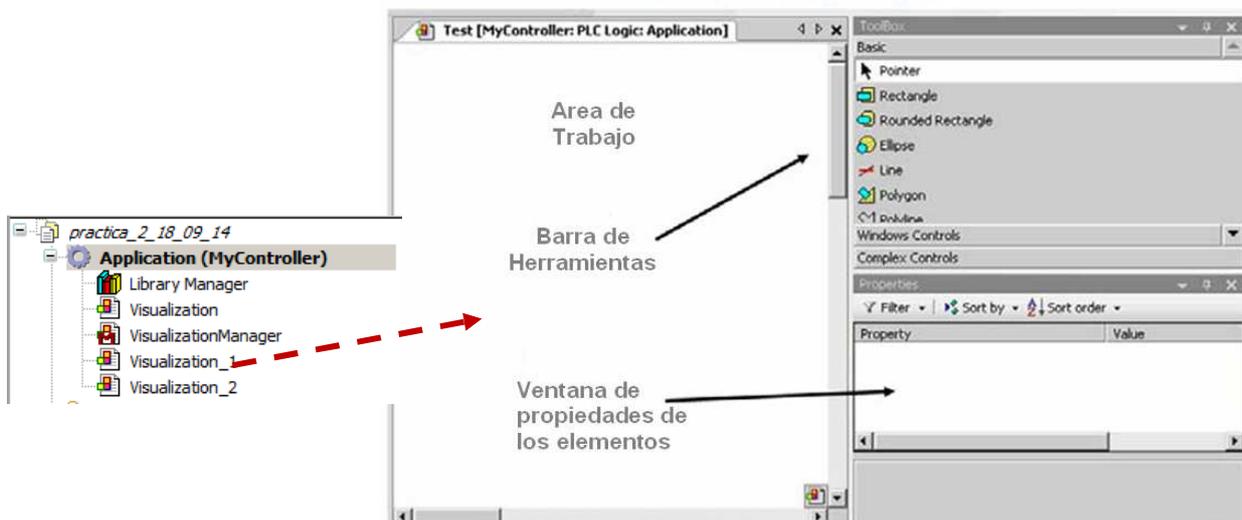
### El objeto Visualización

Para crear una visualización hay primero que ir a la pestaña de 'Herramientas' del navegador y 'Añadir Objeto' Visualización en la aplicación.



El ítem **Gestor de visualización** se añade automáticamente en la aplicación al crear tu primera visualización. Para entrar en la pantalla de visualización hacer doble clic sobre el icono de la pantalla que se desea abrir.

### Editor de Visualizaciones

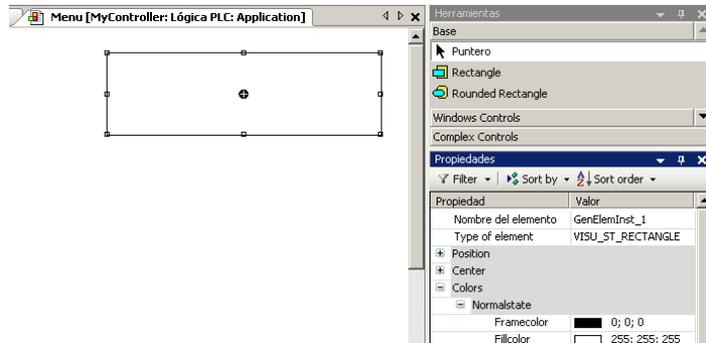


El **Editor de visualizaciones** aparece automáticamente cuando se abre la pantalla y permite crear la visualización. Los elementos principales del editor de visualización son:

<b>Área de trabajo</b>	Área donde se creará la visualización
<b>Barra de Herramientas</b>	Contiene los diferentes objetos de visualización que se pueden usar para crear la pantalla de visualización.
<b>Ventana de propiedades de los elementos</b>	En esta ventana aparecen las propiedades del elemento seleccionado. Aquí se pueden cambiar o configurar estas propiedades.

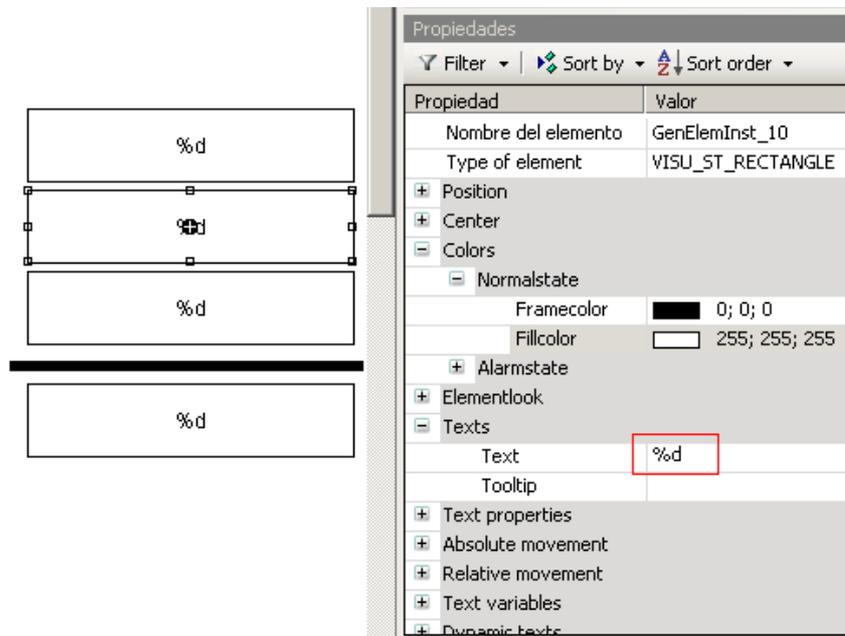
## Crear elementos de visualización

Una visualización contiene elementos de visualización que se utilizan para monitorizar, depurar o cambiar variables de la aplicación, por lo que dichos elementos se tienen que linkar con las variables que desean mostrar.



En la figura de arriba se ha creado un rectángulo que ha sido seleccionado de la barra de 'Herramientas' y después dibujado en el área de trabajo.

Los rectángulos se pueden configurar para visualizar/modificar variables. En la ventana de 'Propiedades', en el campo de 'Propiedades' está la propiedad 'Texts » Text', escribir en el campo de la izquierda el argumento '%d', que indica que en el rectángulo se va a visualizar un número en formato decimal.



Dependiendo del 'Argumento' que se escriba en el campo 'Text' se visualizará el valor de la variable en un formato u otro.

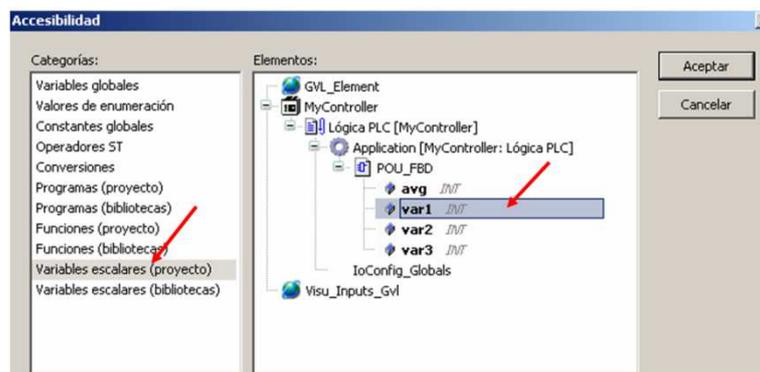
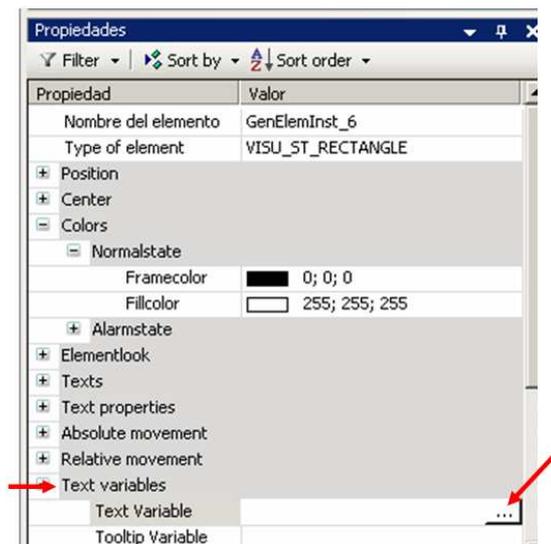
## Argumentos

Otro argumento para la visualización de valores es:

Caracter	Argumento o formato
d,i	Número en decimal
b	Número en binario
o	Número en octal
x	Número en Hexadecimal
u	Número en decimal sin signo
c	Un caracter
s	Se verá el valor en formato string (agrupación de carácter) que se haya especificado en la variable
f	Valor Real; con la siguiente sintaxis '%<alineación> <ancho mínimo>.<decimales> f'. La alineación se define con un signo '-' en el caso que se quiera alineado a la izquierda, con un '+' si se quiere a la derecha.

## Linkar Variables

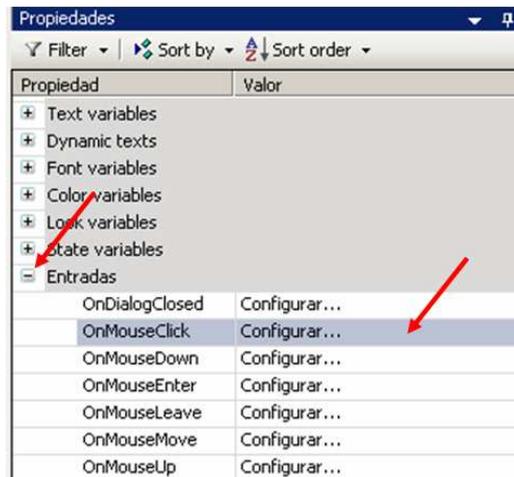
Para linkar la variable para que se vea en el rectángulo, ir a la propiedad 'Text variables » Text Variable' y en el campo escribir la variable del controlador que se desea linkar o pulsar el botón del asistente para seleccionarla en la ventana flotante.



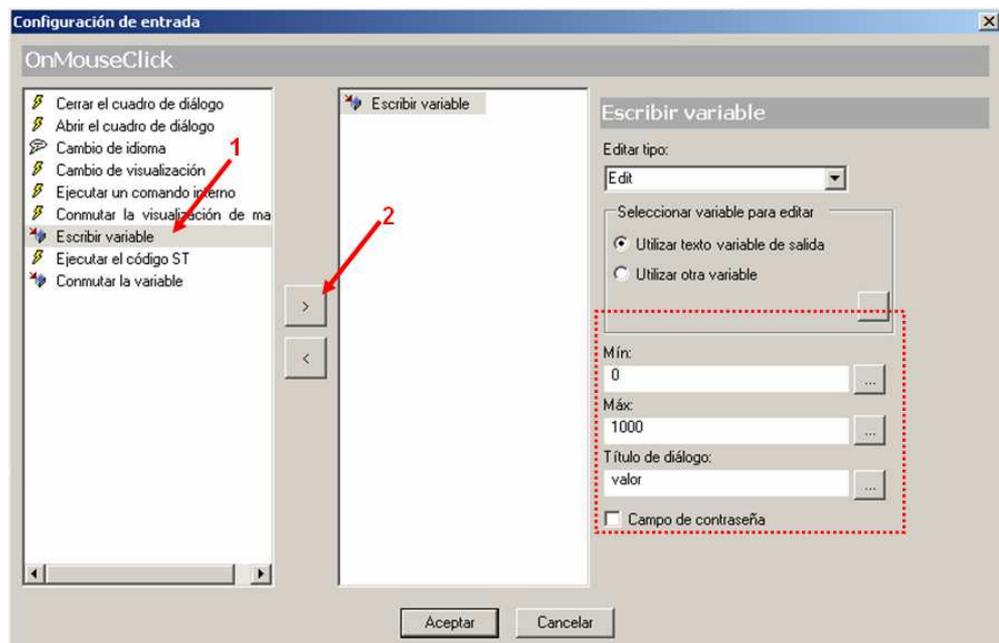
## Variable de entradas

### Campos de entrada

Los **campos de entrada** – son acciones que se configuran para poder introducir un valor en la variable. Para configurarlo hay que seleccionar el rectángulo y en la ventana '**Propiedades**' del elemento seleccionar el parámetro '**Entradas**' y elegir la acción que permitirá variar el valor de la variable (ejemplo, hacer clic con el ratón sobre el rectángulo).



Cuando seleccionamos la acción '**On Mouse clic**'. Aparece la ventana flotante '**Configuración de la entrada**', donde se selecciona el evento que se ejecutará con dicha acción (ejemplo, escribir una variable).

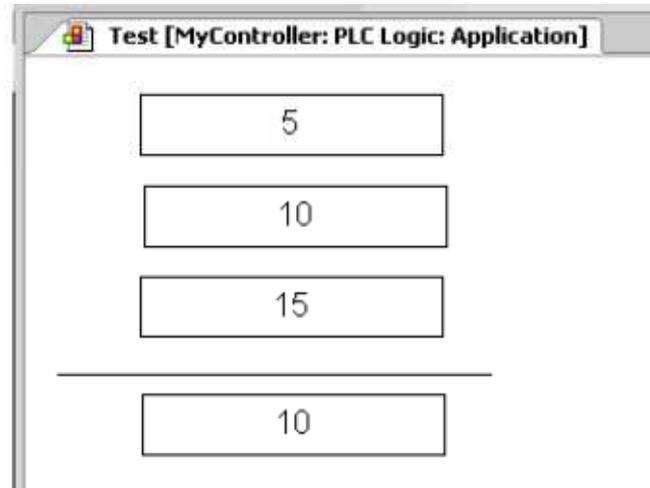


Según el tipo de evento que se desea se habilitará los campos para configurar (ejemplo: valor mínimo, valor máximo).

## Campo de entrada

Si la variable no necesita ser modificada en la visualización, que lo único que se quiere es monitorizarlo, no es necesario configurar ninguna acción de entrada.

Cuando el programa está en modo On-line la pantalla de visualización aparece mostrando los valores.



## Capítulo 8: Librerías de SoMachine

### Descripción

**Introducción** En este capítulo enumera las librerías principales de Codesys y como crear una librería de usuario.

#### **Este CAPÍTULO trata los siguientes temas:**

- Introducción a las Librerías .....8-2
- Agregar librería .....8-2
- Repositorio de bibliotecas .....8-24
- Creación de una librería de usuario .....8-6

## Introducción a las Librerías

### Tres tipos

#### ➤ Librerías de CoDeSys:

Son objetos (Programas, FB, funciones, visualizaciones, estructura de datos) reutilizables que proporcionados por CoDeSys, o una tercera compañía o uno mismo.

Colección de librerías guardadas en el **‘Repositorio de bibliotecas’**.

Las librerías que no son estándar se tienen que añadir en la aplicación del PLC's con el objeto **‘Administrador de bibliotecas’** cuando se tengan que utilizar en la aplicación. (No añadir todas las bibliotecas al proyecto si no van a ser utilizadas en el mismo).

Hay tres tipos de librerías:

#### ➤ Librerías de sistema: Guardadas en (Standard Library)

Proporcionan ciertas funcionalidades del sistema, se agregan automáticamente a la aplicación en función del controlador seleccionado.

- IEC5FC Library, CAN, PLCSystem, ...

#### ➤ Librerías de Aplicación: Librerías que utiliza el usuario par la aplicación. Agregadas por defecto con cada nueva aplicación.

- Librerías Standard :

- Util : Mathematical functions, bit/byte manipulation, ...
- Standard :Timer, Counter,...

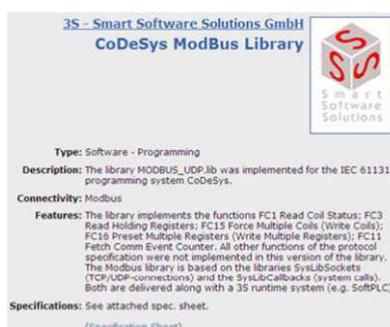
Agregadas por el usuario en función de la necesidad de ser utilizada.

- System Library, PLCopen, Basic Application Library

#### ➤ Librerías Target : depende del vendedor del producto Librerías de los productos de Schneider Electric.

- M238/M258 : Controlador con funciones incrustadas(HSC, PTO\_PWM)

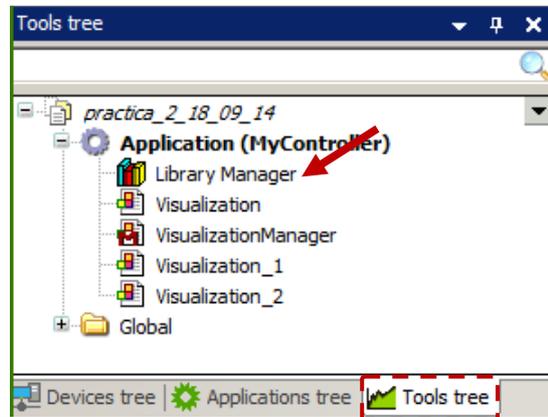
- XBT GC : Controlador con terminal gráfico  
Librerías de productos de terceros.



## Agregar una librería

### Administrador de bibliotecas

El 'Administrador de bibliotecas' es un objeto que ya aparece creado en el árbol de la aplicación, dentro de la pestaña de 'Herramientas' en el navegador.



El administrador de bibliotecas indica las librerías que se han incluido en la aplicación. Las que se agregan automáticamente y las que agrega el usuario.

Para abrir el administrador de bibliotecas hacer doble clic sobre el objeto de 'Administrador de bibliotecas'.

### Agregar librería al proyecto

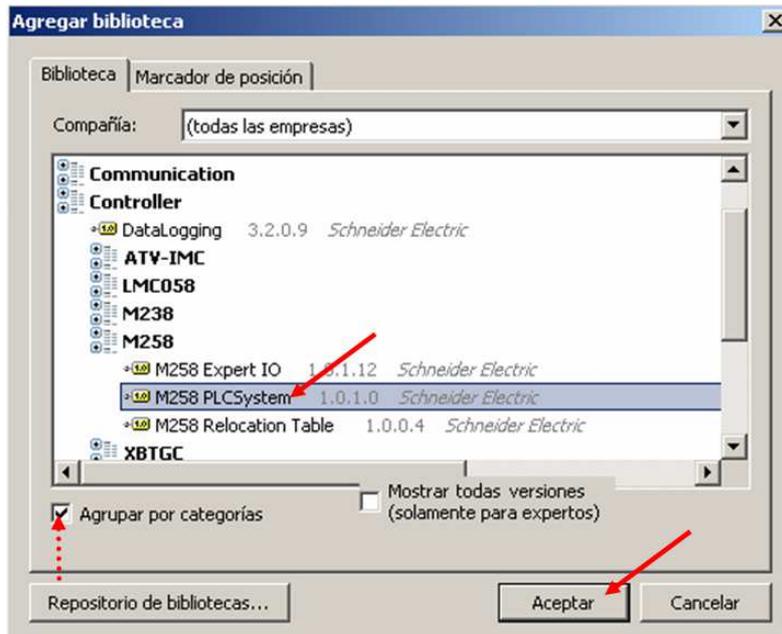
➤ Para agregar una biblioteca al proyecto:

Hacer doble clic sobre el icono de 'Administrador de bibliotecas' que hay en la aplicación.

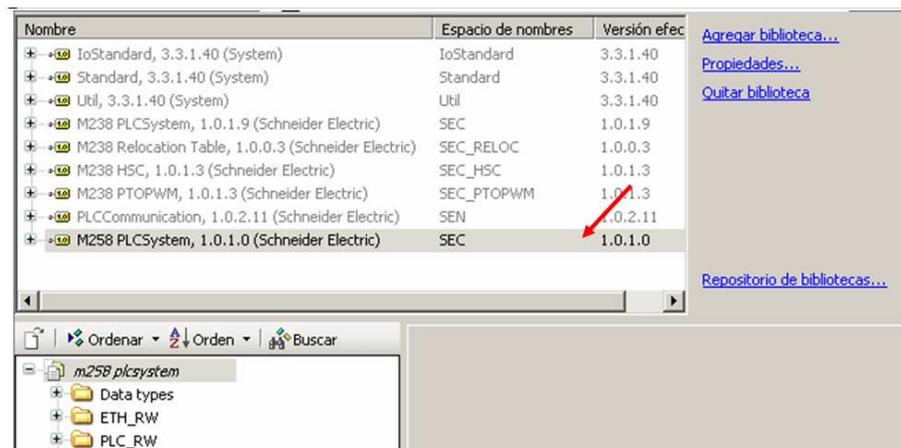
Cuando aparezca la ventana de administrador de bibliotecas, comprobar que la librería no está en el listado de las que ya están agregadas. Sino pulsar sobre el link que hay en la parte superior derecha de la ventana 'Agregar biblioteca...'



Esto abrirá la ventana flotante ‘Agregar biblioteca’ en la pestaña ‘Biblioteca’ elegir la librería que se quiere agregar al proyecto. (Si se sabe el nombre de la librería, deseleccionar la opción de ‘Agregar bibliotecas por categorías’ para que no aparezcan agrupadas en árbol). Pulsar el botón ‘Aceptar’ para agregar al proyecto la librería seleccionada.



Una vez agregada la librería aparecerá en la lista dentro del ‘Administrador de bibliotecas’.



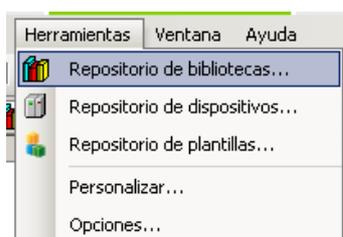
Si seleccionamos la librería en la parte de abajo de la ventana del ‘Administrador de bibliotecas’ aparecen los objetos que contiene esa librería.

## Repositorio de bibliotecas

### Repositorio de bibliotecas

El repositorio de bibliotecas muestra las librerías que hay instaladas en el software SoMachine (que no quiere decir que estén incluidas en el proyecto), y nos permite instalar nuevas bibliotecas de terceros.

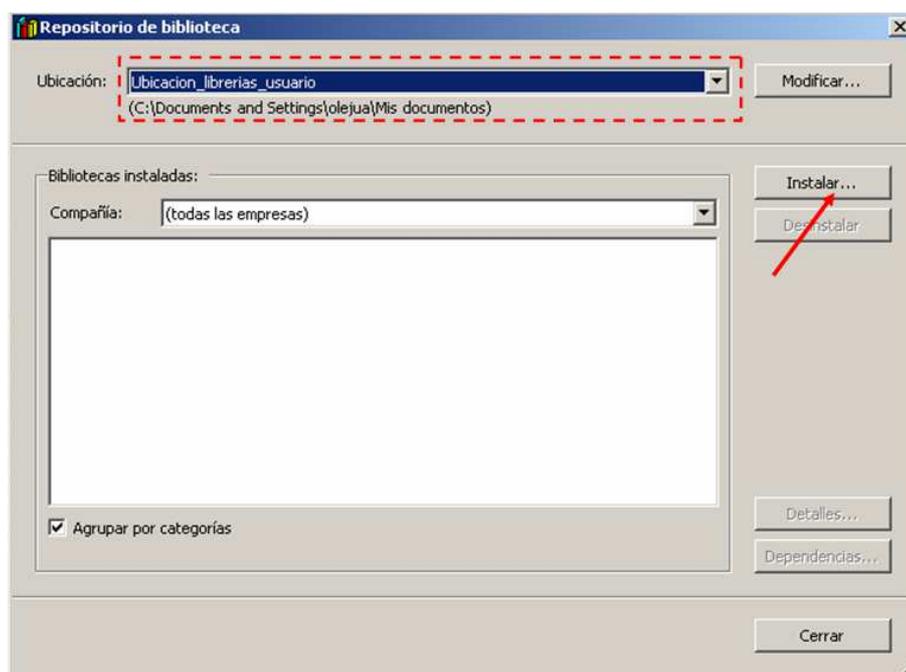
Para abrir el repositorio de bibliotecas ir al menú de contextual y seleccionar '**Herramientas » Repositorio de Bibliotecas**'.



### Instalar una librería

➤ **Para instalar una librería en el Repositorio de Bibliotecas:**

Abrir el repositorio de bibliotecas y aparecerá la ventana flotante.



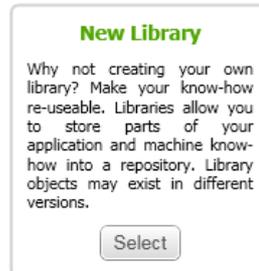
En el campo '**Ubicación**' se especifica la ruta de la carpeta donde están las bibliotecas instaladas (La opción '**System**' es la ruta por defecto, pero pulsando el botón '**Modificar**' se pueden agregar otras rutas).

Una vez especificada la ubicación de donde se instalará la biblioteca, pulsar el botón '**Instalar**' que hay en la parte derecha de la ventana, aparecerá el navegador y seleccionar el archivo de librería \*.lib que se desea instalar.

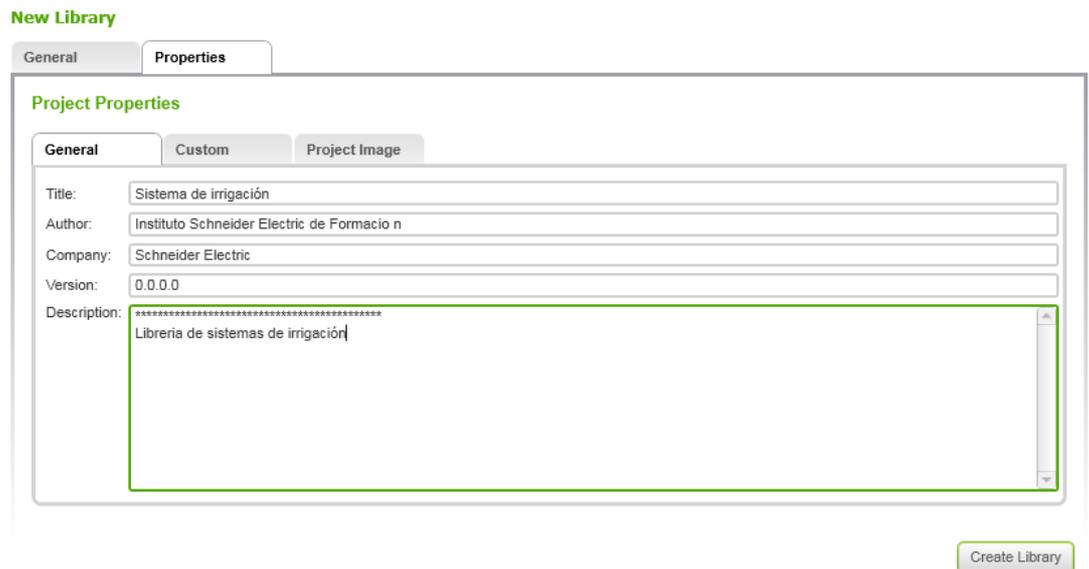
## Crear una librería de usuario

Crear un proyecto vacío tipo librería.

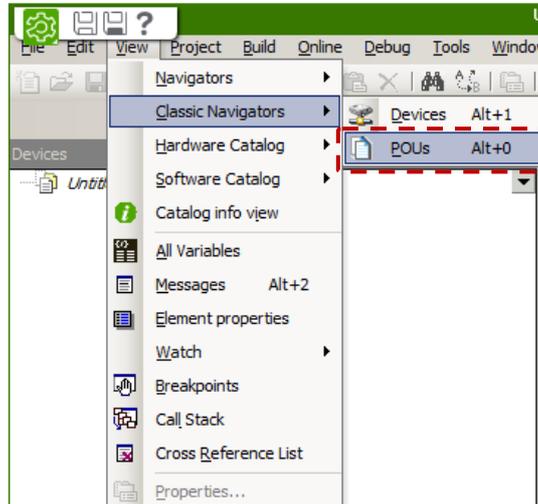
Para crear una librería de usuario, en la ventana de 'Inicio' del SoMachine hay que crear un proyecto **nuevo y pulsar en 'New Library'** y a la hora de guardar el archivo, se creará con el tipo librería.



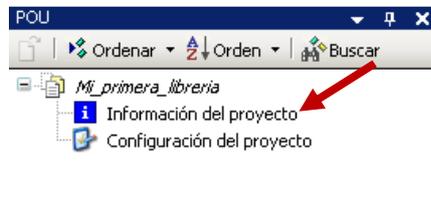
Una vez creado el proyecto, en las pestaña 'General', escribiremos el nombre queremos que tenga nuestra librería. En la pestaña 'Properties', se introducirá la información que va a tener la librería: Compañía, Título (nombre de la librería), Versión – El resto de la información es opcional.



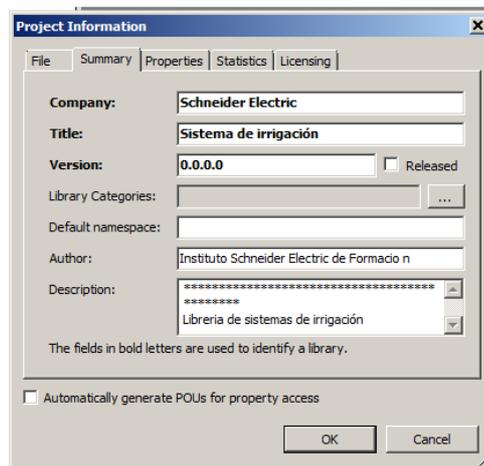
En el menú contextual seleccionar '**Vista » Classic Navigator » POU**' aparecerá en el navegador la vista '**POU's**' que es donde se añaden los programas, funciones, bloques de funciones y estructura de datos que se quieren añadir a la librería.



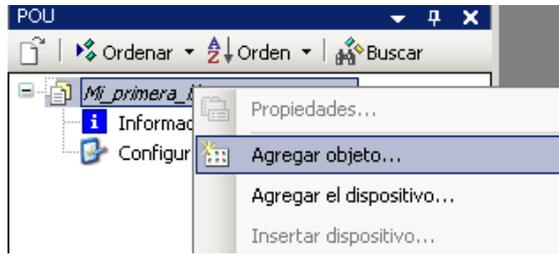
Para añadir los datos que luego se verán en la librería, dentro de la vista '**POU**' hacemos doble clic en '**Información del proyecto**'



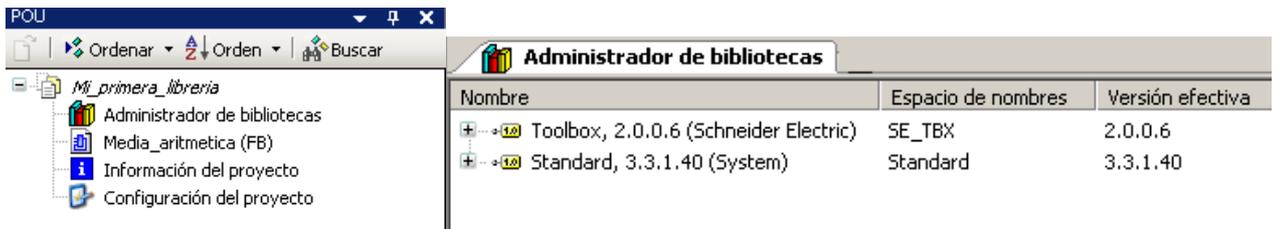
En el campo '**Library Categories**' podemos añadir a que categoría deseamos que se añada esta librería, sino ponemos nada se añadirá a la categoría '**Varios**'.



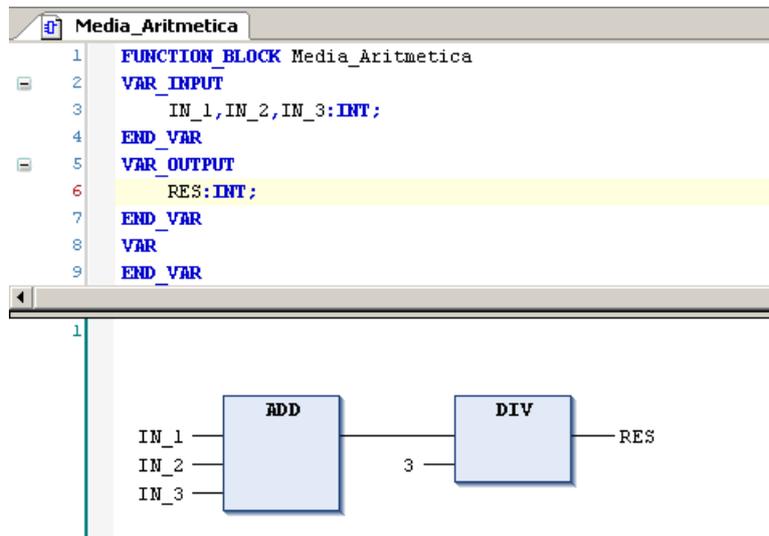
Para añadir los objetos en este caso seleccionar el directorio del proyecto (ejemplo. Mi\_primera\_libreria) pulsar botón derecho y seleccionar '**Añadir Objeto**'.



Agregar un POU del tipo bloque de funciones llamado '**Media aritmetica**' y agregar el objeto '**Administrador de bibliotecas**' y agregar las librerías '**Standard**' y '**Toolbox**' que se utilizarán para realizar la programación del bloque de funciones.



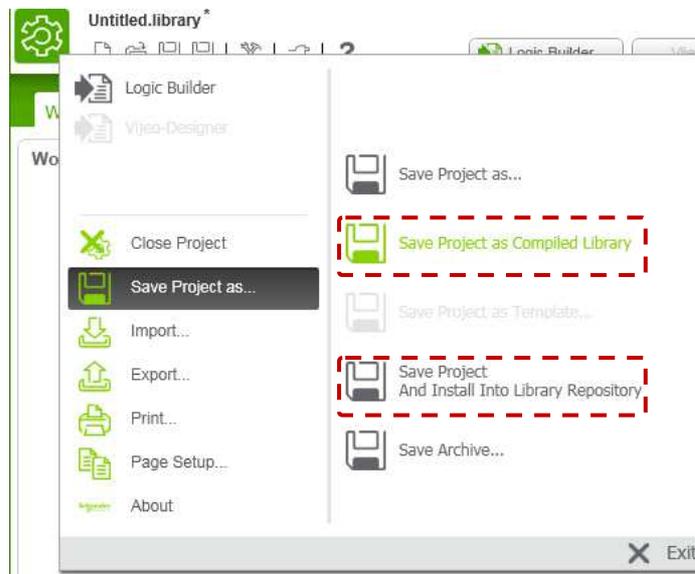
Programar el bloque de funciones como se muestra en el ejemplo.



Cuando se ha finalizado la programación del **FB**, hay que compilarlo sin errores. En el menú contextual seleccionar **'Compilar » Verificar todos los objetos del grupo'**.



Si se tiene la librería verificada y sin errores para guardarla, hay que ir a la ventana general de SoMachine Central y se puede guardar de dos maneras.



- **'Guardar el proyecto como librería compilada'** que se puede seleccionar en el menú contextual de **'Archivo'**. Esta opción guarda la librería compilada, es decir, que el usuario no podrá ver el interior de los elementos de la librería (Librería encriptada).
- **'Guardar el proyecto e instalarlo en el repositorio de bibliotecas'** que se puede seleccionar en el menú **'Archivo'** igual que el anterior. En este caso la librería ya se queda guardada en el repositorio de bibliotecas lista para ser agregada al proyecto, pero es una librería abierta, es decir, que el usuario podrá ver internamente los elementos de la librería.

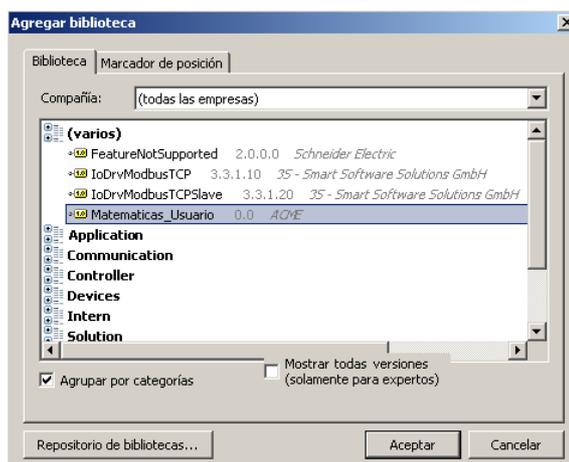
A la hora de utilizar la librería de usuario generada en un proyecto se seguirá una acción u otra en función de cómo se ha grabado la librería.

## Añadir Librería de Usuario al Proyecto

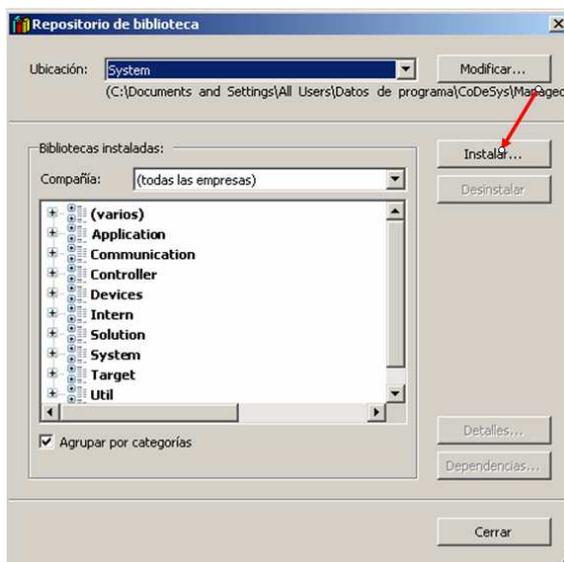
### Como añadir una Librería

#### ➤ Para añadir una librería de usuario:

Si la librería de usuario se ha guardado con la opción **‘Guardar el proyecto e instalarlo en el repositorio de bibliotecas’** simplemente hay que abrir el **‘Administrador de bibliotecas’** del proyecto pulsar **‘agregar biblioteca...’** y seleccionar la biblioteca en la carpeta **‘Varios’** del **‘Repositorio de bibliotecas’**.



Si la librería se ha guardado con la opción **‘Guardar el proyecto como librería compilada’**, hay que instalarla previamente en el **‘Repositorio de bibliotecas’** antes de ser agregada en el **‘Administrador de bibliotecas’** del proyecto del usuario.

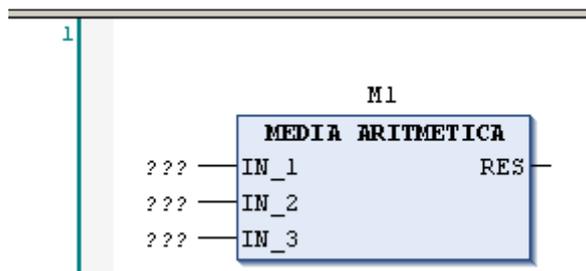
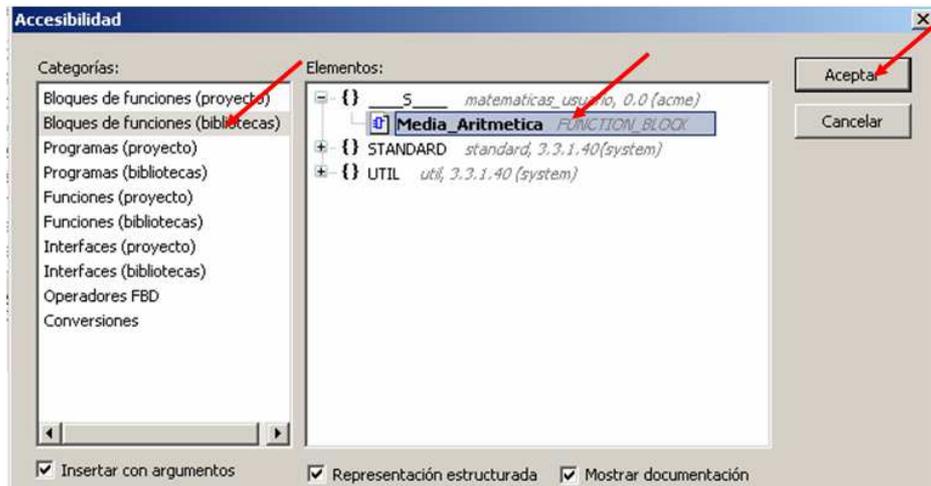


Para instalarla, en el campo **‘Ubicación’** se selecciona **‘System’** que será la ubicación donde la quedará la librería de usuario una vez instalada.

Para instalar la librería pulsar el botón de 'Instalar' en el 'Repositorio de bibliotecas' y en la ventana flotante que aparece **'Seleccionar biblioteca'**, en el campo **'Tipo'** elegir el tipo **'Biblioteca'** que busca la bibliotecas con el formato de archivo **\*.library**. Buscar en el navegador la ruta donde se había guardado previamente la librería y pulsar el botón **'Abrir'**.



Ahora ya se puede encontrar la biblioteca en el repositorio, para agregarla al proyecto. Una vez agregada al proyecto ya se pueden utilizar los elementos de creados en la librería (FB's, DUT, Programas) como una instrucción más para la programación del proyecto.





# Capítulo 9: Comunicación con el PLC

## Descripción

**Introducción** Este capítulo trata de cómo crear el Gateway para poder conectarse al PLC y las opciones de configuración online.

### **Este CAPÍTULO trata los siguientes temas:**

- Establecer una comunicación con el PLC .....9-2
- Descarga múltiple .....9-2
- Source download.....9-37

## Establecer una comunicación con el PLC

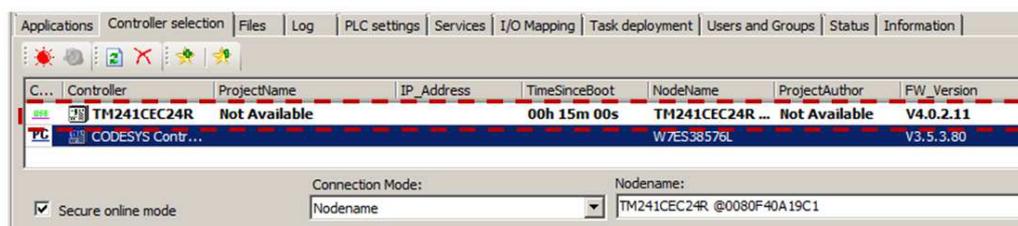
### ➤ Como conectarse al PLC



Inicialmente se tiene que tener conectado con el cable de programación el PC al controlador (cable de programación **USB - Mini USB** para los controladores M238, M241, M251, M258, LMC058 y ATVIMC para las pantallas XBTGC el cable será el XBTZG935). Para acceder añadir un Gateway, hacer doble clic sobre el controlador en la ventana de dispositivos y en el área de trabajo aparece las diferentes pestañas de configuración del controlador, en esta seleccionar la pestaña '**Controller selection**'.

Si está conectado el cable y ha reconocido el driver automáticamente aparecerá un controlador en negrita donde nos especifica una serie de información.

- **Conexión Mode:** especifica el tipo de conexión (USB, ETH, PC).
- **Controller:** Indica el modelo (referencia) del controlador.
- **Project Name:** nos informa del nombre del proyecto que hay cargado en el controlador (si no tiene cargado ninguno porque el controlador es nuevo pondrá '**Not Available**')
- **IP Address:** Dirección IP guardada en el equipo (si dispone de ella).
- **Time Since Boot:** El tiempo que lleva en marcha el controlador desde el último inicio.
- **Node Name:** es el nombre lógico que tiene este dispositivo en concreto, normalmente por defecto está compuesto por la referencia del controlador, @, y la MAC del equipo o el número de serie, dependiendo si tiene Ethernet o no.
- **Project Author:** Especifica el autor del proyecto que hay guardado si se ha especificado en el campo '**Autor**' de propiedades del proyecto.
- **FW\_Version:** Indica la versión de firmware del controlador.

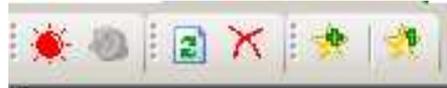


Si queremos cambiar de controlador al que conectarse ó de método de conexión, podremos cambiarlo haciendo doble clic sobre esa línea, que aparecerá ahora resaltado en negrita.

Una vez tenemos elegido el controlador al que nos queremos conectar (resaltado en negrita), para conectarnos solo tenemos que ir al menú textual 'Online >> Login' ó pulsar en el icono de Login  de la barra de herramientas.



En la pestaña 'Controller Selection' dispone de una barra de iconos propia donde se pueden realizar las siguientes acciones.



Botón que activará la intermitencia de los LED's del controlador seleccionado para su identificación física, la intermitencia acabará automáticamente a los 30 segundos ó inmediatamente si volvemos a pulsar el botón.



Botón que activará la intermitencia de los LED's y una señal acústica del controlador seleccionado para su identificación física, esta opción solo estará disponible para aquellos controladores que la soporten.



Botón que refresco, realiza un nuevo escaneo de las conexiones para ver si encuentra nuevos controladores.



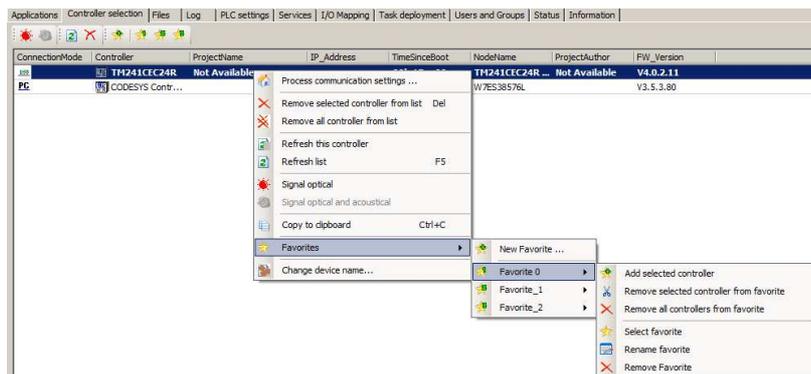
Botón que elimina la conexión seleccionada de la lista.



Botón que nos permite crear una lista de Favoritos, lista de controladores específicos (muy útil cuando los controladores están conectados en red).

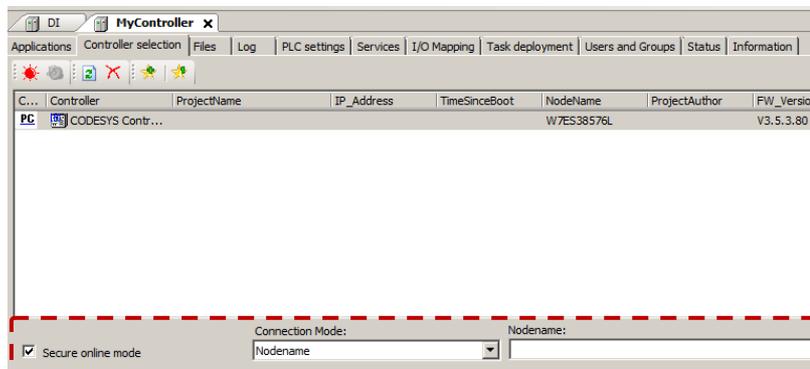


Visualizar favoritos.



## Modos de Conexión

Los modos de conexión, que se encuentran en la parte inferior de la ventana de selección del controlador, nos permite seleccionar un formato para la dirección de conexión. Los siguientes formatos son compatibles:



Si selecciona la opción **'automática'** de la lista **'Modo de conexión'**, puede introducir el nombre del nodo, la dirección IP o la URL de conexión (localizador uniforme de recursos) para especificar la dirección. **NOTA : No utilice espacios al principio o al final de la dirección .**

Si selecciona el **'Nombre de nodo'** de la lista **'Modo de conexión'** , puede introducir el nombre de nodo de un controlador para especificar la dirección. El cuadro de texto se rellena automáticamente si hace doble clic en un controlador en la lista de controladores. Si el controlador que ha seleccionado no proporciona un nombre de nodo, el modo de conexión cambia automáticamente a la dirección IP.

Si selecciona la opción **'dirección IP'** de la lista **'Modo de conexión'**, puede introducir la dirección IP de un controlador para especificar la dirección.

Si selecciona la opción **'Nodename través de NAT (TCP remoto)'** de la lista **'Modo de conexión'**, puede especificar la dirección de un controlador que se encuentra detrás de un router NAT en la red . Introduzca el nombre de nodo del controlador, y la dirección IP o nombre de host y el puerto del router NAT.

Si selecciona la opción de **'dirección IP a través de NAT (TCP remoto)'** de la lista **'Modo de conexión'**, puede especificar la dirección de un controlador que se encuentra detrás de un router NAT en la red. Introduzca la dirección IP del controlador, y la dirección IP o nombre de host y el puerto del router NAT.

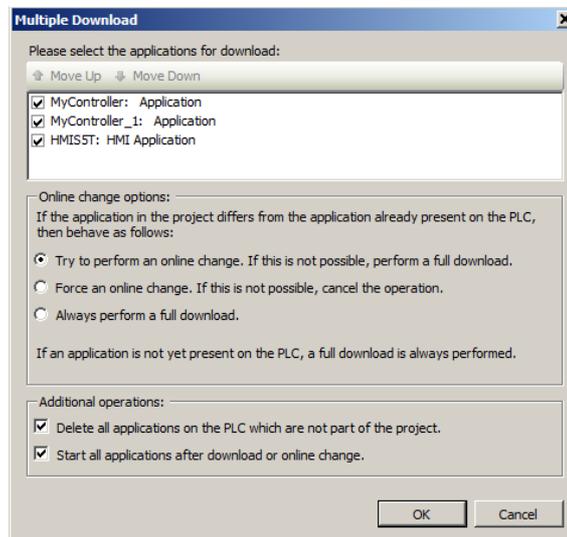
La opción **'Secure online mode'** muestra un mensaje de advertencia que requiere de confirmación, cuando se va a establecer la conexión con un controlador.

## Descarga Múltiple

Para descargar varios programas a diferentes equipos de la arquitectura de control, una vez esté en modo online ir al menú contextual 'En línea » Descarga Múltiple'.



Una vez seleccionado la 'Descarga Multiple', aparece la ventana con las opciones de descarga.



Se pueden seleccionar una o varias aplicaciones, incluso si no se descargaran en el mismo dispositivo (NOTA: Las aplicaciones seleccionadas se descargarán en el mismo orden en que se enumeran en el cuadro de diálogo).

Seleccionar una opción de 'Changue online Options':

- Trate de realizar un cambio en línea. Si esto no es posible, realizar una descarga completa.
- Forzar un cambio en línea. Si esto no es posible, cancele la operación.
- Realice siempre una descarga completa. Independientemente de las versiones ya existentes en el controlador, todas las partes de las aplicaciones seleccionadas se vuelven a cargar en el controlador.

También se pueden habilitar opciones adicionales:

- Las aplicaciones existentes, que ya no forman parte del proyecto, se suprimen del controlador.
- Las aplicaciones seleccionadas, se deben iniciar después de la descarga ó el cambio en línea.

Después de haberse ejecutado la descarga múltiple, aparece una ventana que nos dice la lista de todas las aplicaciones cargadas del proyecto y si la descarga se ha realizado correctamente.

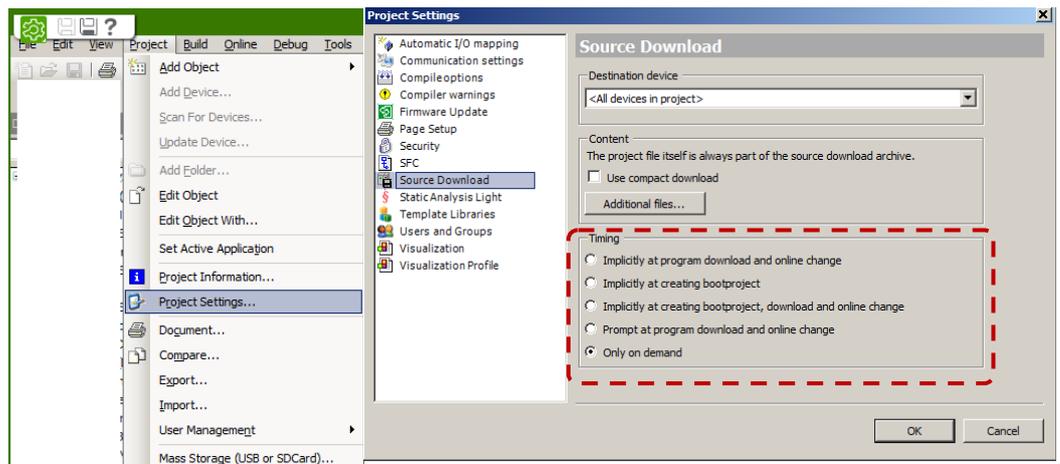


## Source download

En el menú contextual '**Online** → **Source Download to connected device**' esta opción se utiliza para la creación, transferencia y almacenamiento del archivo del proyecto real al controlador.

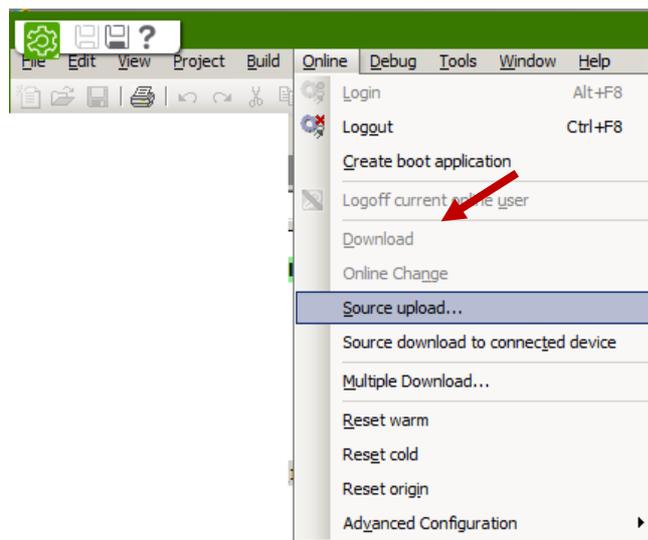
El archivo de proyecto almacenado es un archivo \*.prj.

Para los ajustes del archivo source, cuando se tiene que cargar, se tendrá que ir a el menú contextual '**Project** → **Project Settings..**' y en la ventana flotante de 'Project settings' que aparece elegiremos '**Source Download**'.



## Upload Source

En el menú contextual '**Online** → **Upload Source**' esta opción se utiliza para recuperar el archivo del proyecto que tiene guardado el controlador.





# Capítulo 10: Gestión del proyecto

## Descripción

**Introducción** Este capítulo trata de los diferentes servicios online y de las herramientas para la depuración de la aplicación.

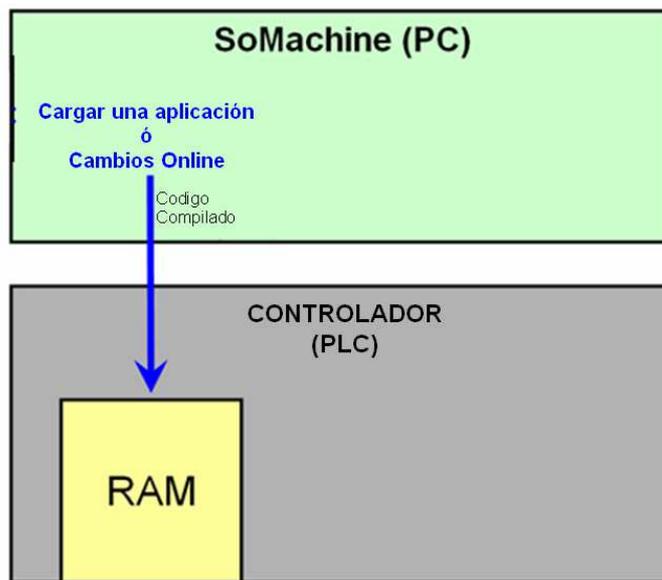
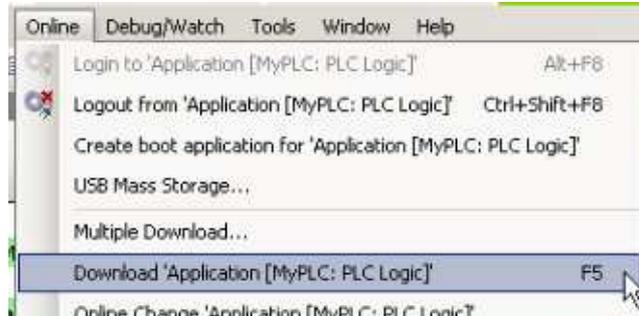
### Este CAPÍTULO trata los siguientes temas:

- Compilar programa ..... 10-2
- Aplicación de inicio..... 10-3
- Cambios Online..... 10-4
- Importar ..... 11-25
- Exportar ..... 11-36
- Gestion de ficheros de datos - Dataloging ..... 11-47

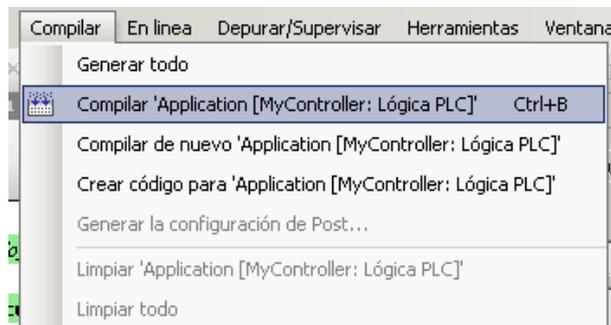
## Compilar programa

### Compilar el programa

Cuando se descarga el programa al PLC, este se compila y luego se carga en la memoria RAM del PLC. Que no es una memoria volátil.



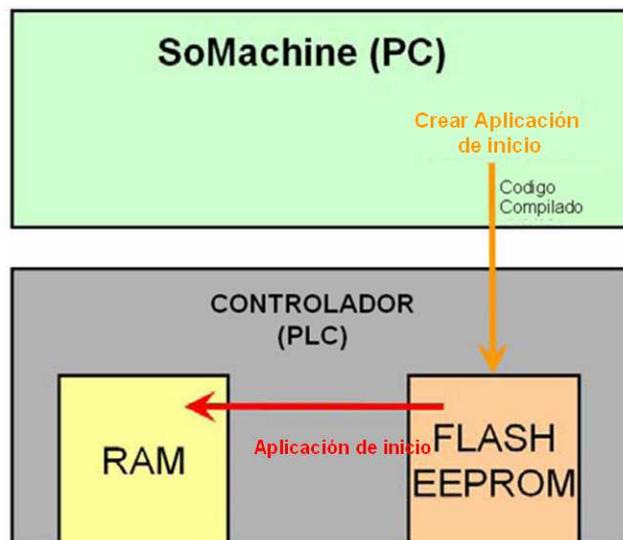
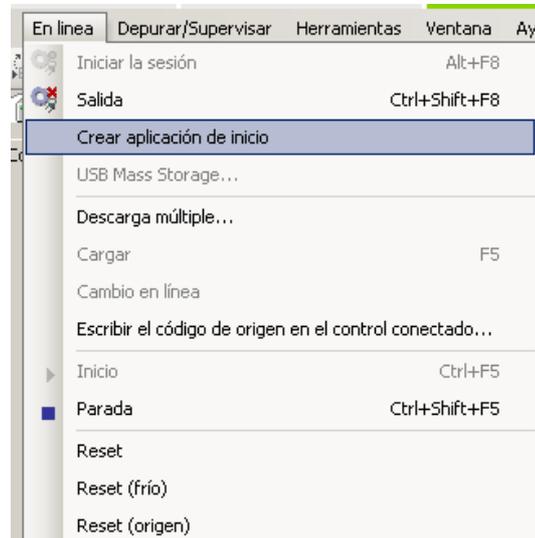
La opción de '**Compilar**' no genera el código de programa. La opción compilar sólo verifica que la aplicación no contiene errores. Generar código de máquina se crea cuando se realiza la descarga de la aplicación.



## Aplicación de Inicio

### Como crear una aplicación de inicio

- Para crear una aplicación de inicio.



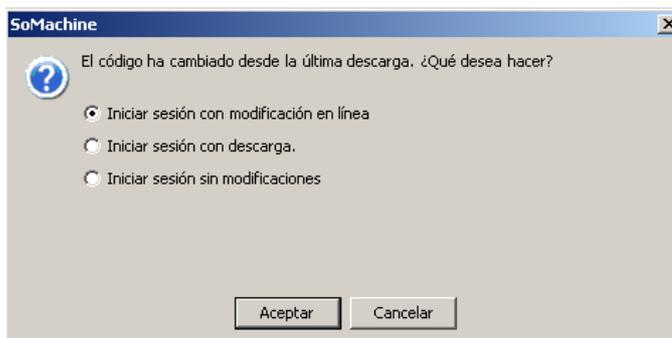
- **Crear aplicación de inicio** – carga el código compilado en la memoria **flash**.
- En el ciclo de inicialización del PLC, el código del programa se carga desde la memoria **flash** a la memoria **RAM**, después el PLC se pone en **Run**.

SoMachine advierte cuando la **aplicación de inicio** es diferente al programa que hay en la RAM o al inerves.

## Cambios Online

### Principios de Operación

#### ➤ Modo de Operación:



**Compilador Incremental** – solo compila los cambios del código.

- Cuando se hace una descarga, la información del número de compilación se queda guardado en el PC, los cambios son compilados y cargados.
- Borrar el contador de compilación con la opción Limpiar que se encuentra en '**Compilar » Limpiar o Limpiar todo**'. Limpiar el contador de compilaciones no es posible si se está online.

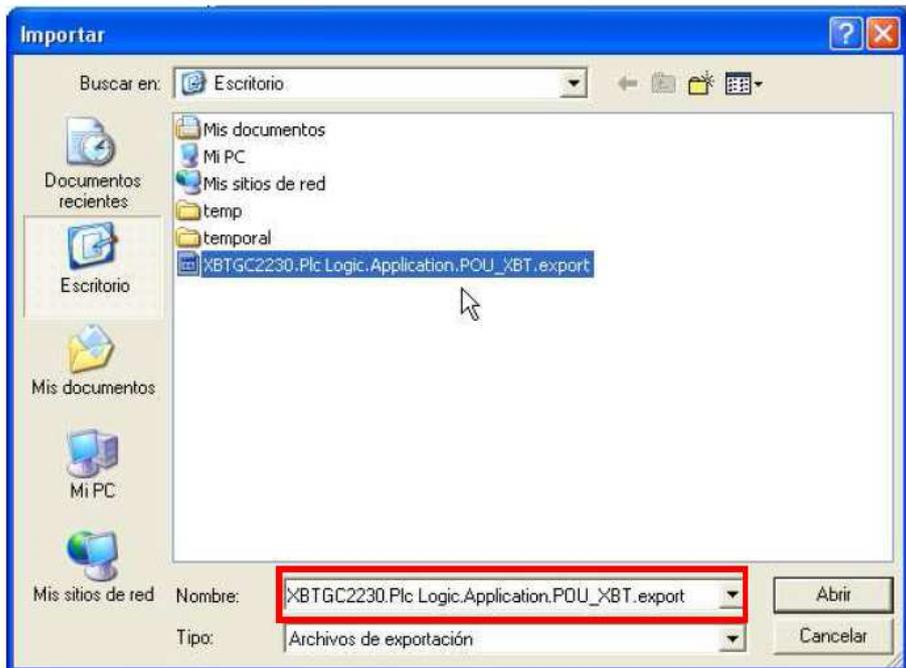
Los cambios online permiten al usuario hacer los cambios de programa pertinentes, sin tener que poner el PLC en Stop. Los siguientes cambios no son posibles hacerlos online:

- Un cambio en el configurador de tareas.
- Modificar la configuración de I/O físicas.
- Después de ejecutar la acción '**Limpiar**' ó '**Limpiar todo**'. Ya que el contador de compilación y la información ha sido borrada.

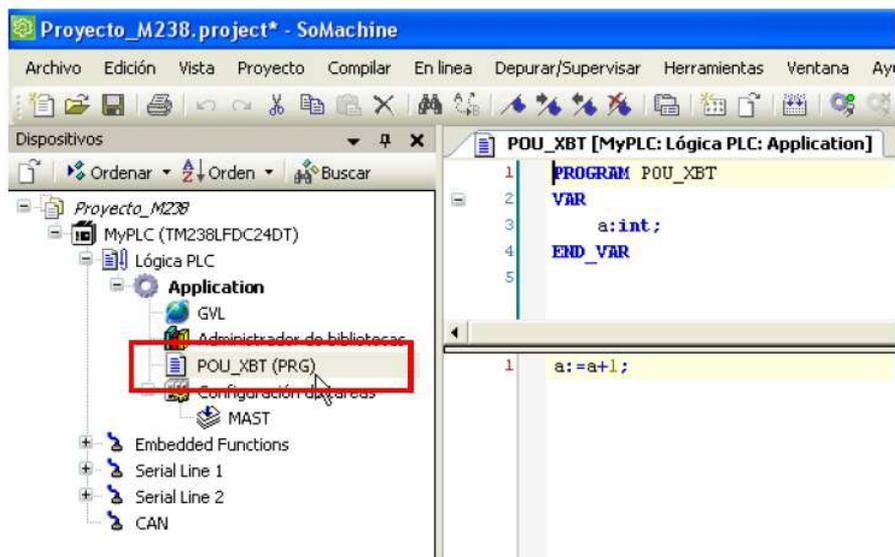
# Importar

## Importar POU

Una vez exportado un POU, es posible importar, solo ese objeto realizando la operación contraria, para ello se tendrá que ir a la barra de menú textual (**Proyecto / Importar**). Se abre un cuadro de diálogo para búsqueda de archivos de exportación. De forma predeterminada el filtro está establecido en archivos con la extensión **".export"**.



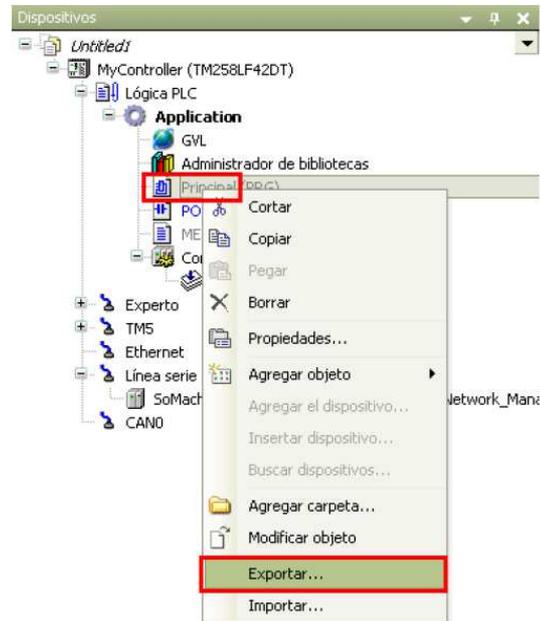
Aparecerá una ventana indicando si la importación se ha realizado con éxito. Y dentro del proyecto se podrá ver el POU importado.



Los objetos importados se agregan al proyecto abierto. Cuando ya existen objetos con nombres similares, se emiten diálogos de mensaje apropiados y en caso dado se puede seleccionar, si un objeto y cuál debe ser sobrescrito o no.

## Exportar

Para exportar cualquier objeto del proyecto con el fin de poderlo añadir a otro proyecto anteriormente se tiene que seleccionar el objeto en cuestión y con el botón derecho seleccionar de la ventana de opciones **'Exportar'**.



Aparece una ventana flotante, donde hay que indicarle en que carpeta se desea guardar los archivos de exportación. Cuando se ha seleccionado la carpeta deseada, confirmar la exportación pulsando el botón de **'OK'**.

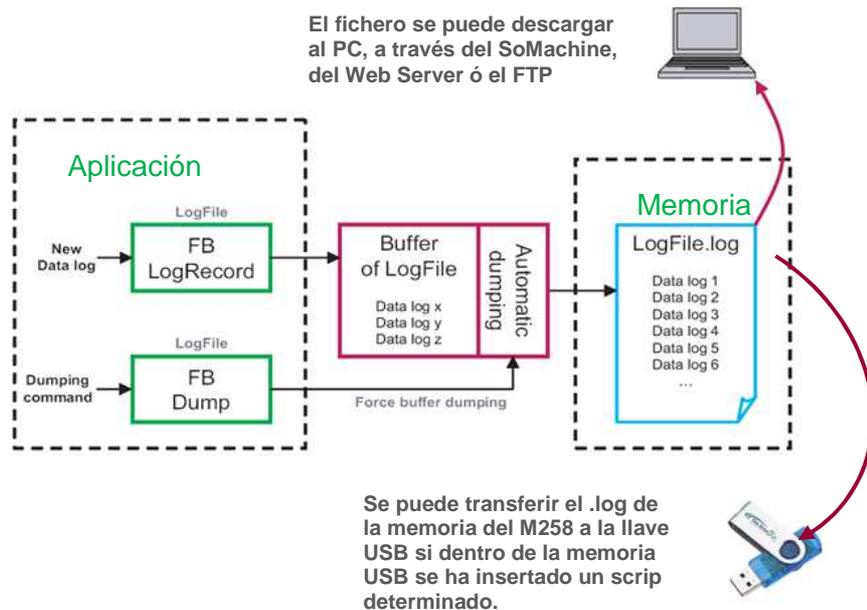


# Gestión de ficheros de datos - DataLogging

## Introducción.

Puede ser necesario para la aplicación archivar ciertas variables, para poder ser descargadas y tratadas por el operario. Para ello necesitamos poder guardar los datos en ficheros.

Con el bloque de función FB LogRecord se crean los ficheros .log en el M258. Inicialmente se escriben los datos en un buffer y son traspasados a la memoria del M258 cuando el buffer se encuentra al 80% o cuando se ejecuta la función Dump.

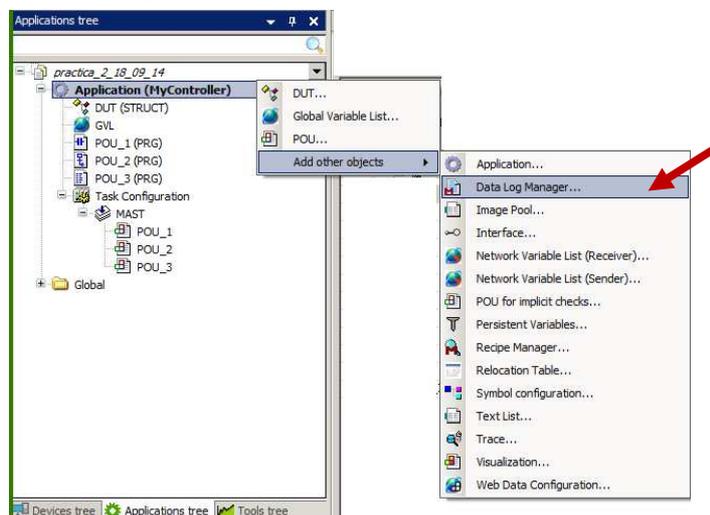


El fichero '.log' que se encuentra en la memoria del M258 puede ser traspasado a una memoria USB, para ello dicha memoria deberá tener un scrip en su interior, y cuando se inserte en el M258 automáticamente obtendrá el fichero .log

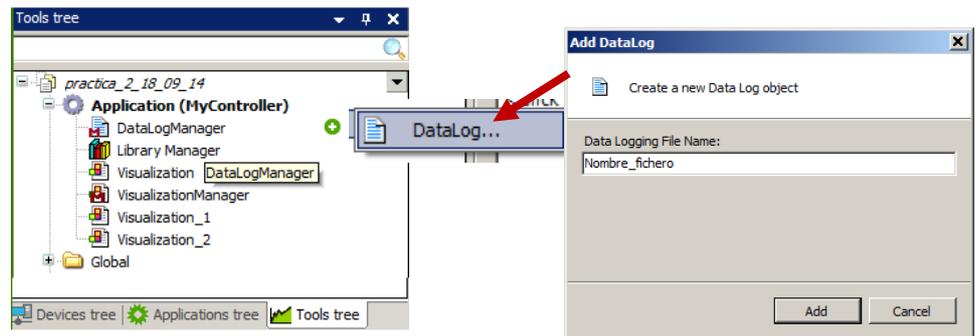
## Creación de un fichero .log

A continuación se muestran los pasos a seguir para realizar el proceso de creación de fichero.log y traspaso de datos a memoria USB.

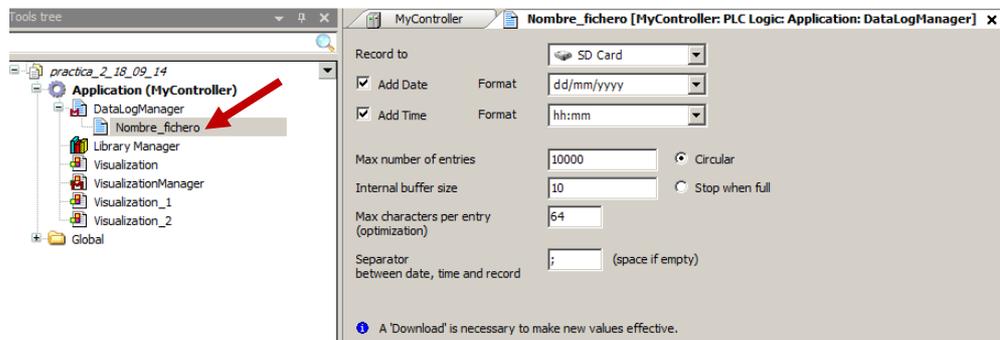
1. Añadir un DataLog Manager al proyecto. Sobre 'Application', botón derecho 'Agregar Objeto' -> DataLogManager



- Ahora en la pestaña de 'Herramientas', insertar el Datalog. Sobre DataLogManager, botón derecho 'Agrega Objeto' -> 'DataLog'. En el campo 'Nombre' escribir el nombre que se desea que tenga el fichero.



- En la pestaña Logfile que se abre configurar el datalog (formato de fecha, de hora...etc).



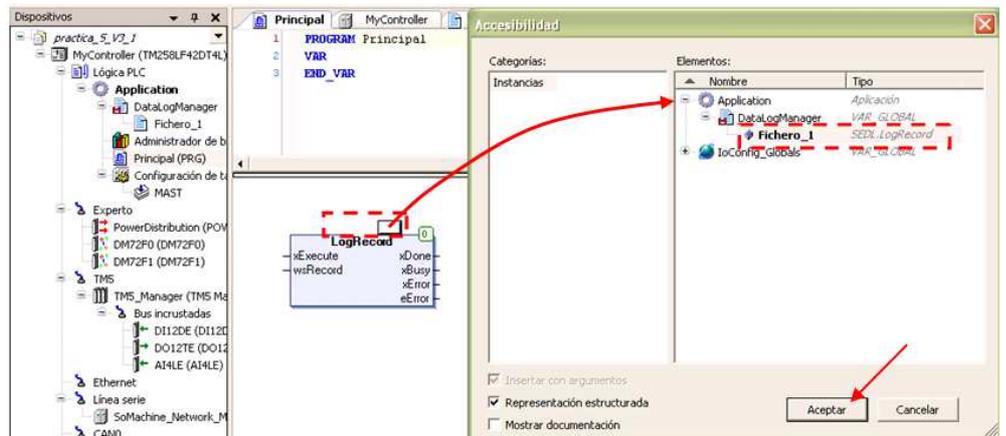
### Insertar datos en el fichero

Una vez creado el fichero .log, se tendrá que utilizar los bloque de función FB LogRecord para abrir el fichero y guardar los datos que se deseen. Inicialmente se escriben los datos en un buffer y son traspasados a la memoria del M258 cuando el buffer se encuentra al 80% o cuando se ejecuta la función LogRecordDump que se guardan en ese preciso momento.

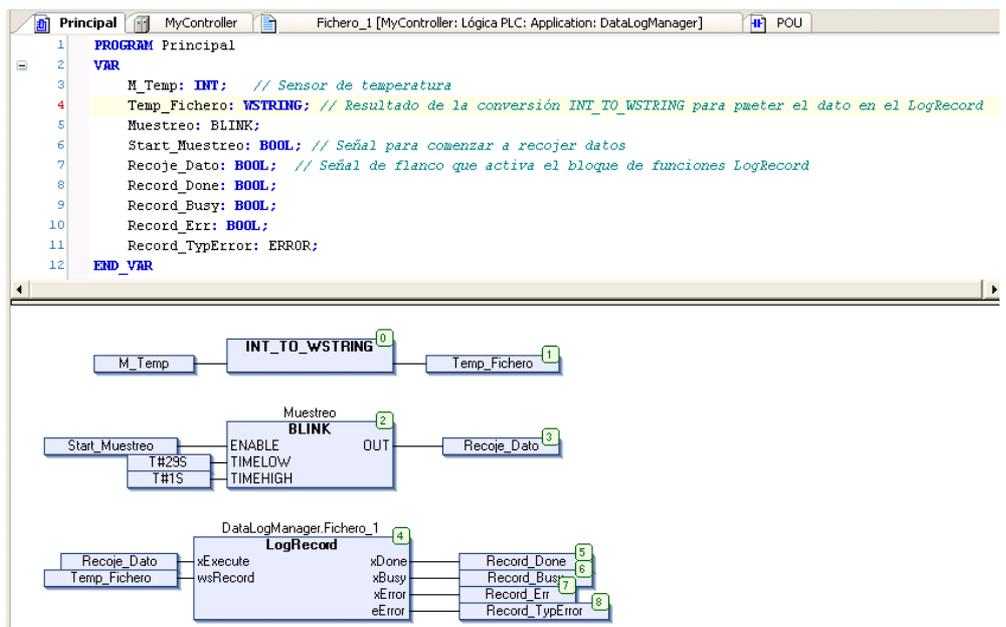
Los pasos para la inserción del bloque de función LogRecord.

1. Abrir el POU previamente creado para la inserción del bloque de función.
2. Después introducir el bloque función 'Insertar llamada a módulo' y elegir el bloque de funciones 'LogRecord' que se encuentra en la librería 'SEDL'.

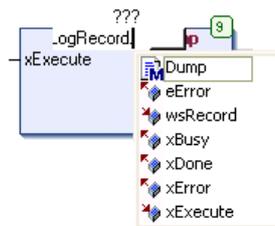
- Una vez insertado el bloque de función en la parte de arriba hay que linkar en la llamada a la instancia al fichero\_1 de la siguiente manera.



- Rellenar las entradas y salidas tal y como se muestra en la figura (*Ejemplo para guardar Medida de Temperatura cada minuto*), teniendo en cuenta que la entrada del bloque 'wsRecord' tiene que tener formato 'wstring'.



- Para llamar al bloque **LogRecord.Dump** que guarda los datos del buffer en memoria cuando se desea. Escribir 'LogRecord' en el módulo, poner un punto '.' y seleccionar la función 'Dump'.



6. La programación del bloque de la siguiente manera.



# Capítulo 11: Herramientas de depuración

## Descripción

**Introducción** Este capítulo trata de las diferentes herramientas para la depuración de la aplicación.

### **Este CAPÍTULO trata los siguientes temas:**

- Herramientas de depuración ..... 10-2
- Monitorización de variables..... 10-3
- Editor de trazas ..... 10-4
- Puntos de interrupción y ejecución paso a paso ..... 11-26

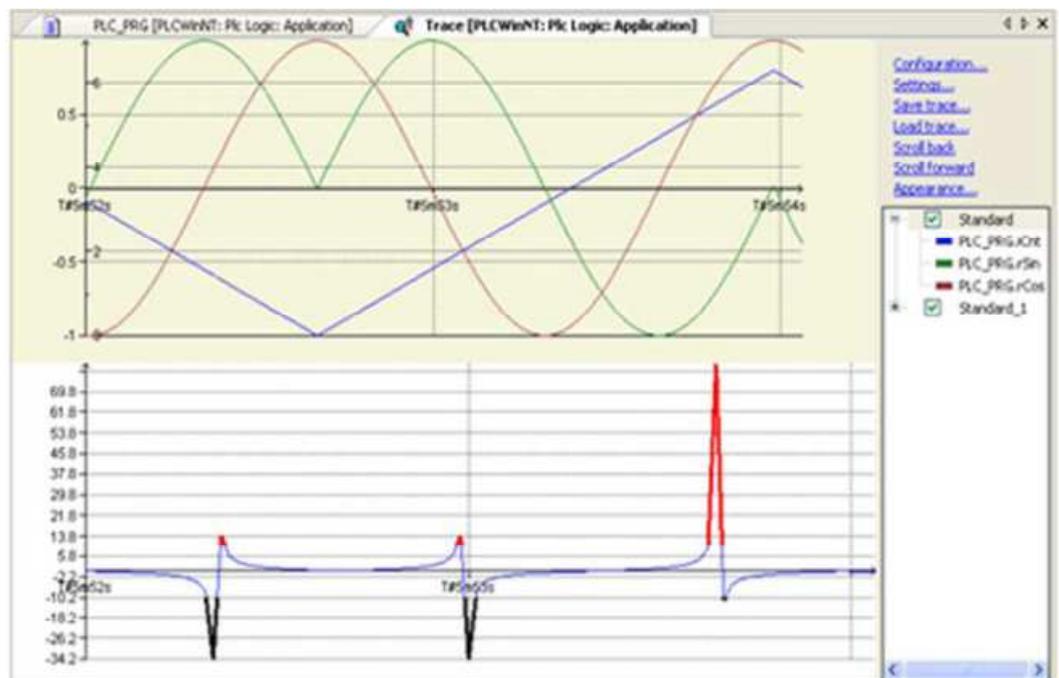
## Herramientas de Depuración

### ➤ Las Herramientas

- **Simulación del programa** sin necesidad de tener el hardware.
  - **Monitorización de variables y forzado.**
  - **Herramientas avanzadas** para el control de la ejecución del programa.
- Puntos de interrupción.
  - Modo de ejecución paso a paso.

```
1 IF switchTRUE = 0 THEN
2   aaa_410 := aaa_410 + 1;
3   IF aaa_410 > 500 THEN
4     SWITCHTRUE := 1;
5   END_IF
6   IF aaa_410 > 10 AND aaa_410 < 20 THEN
7     POU_SUB();
8   END_IF
```

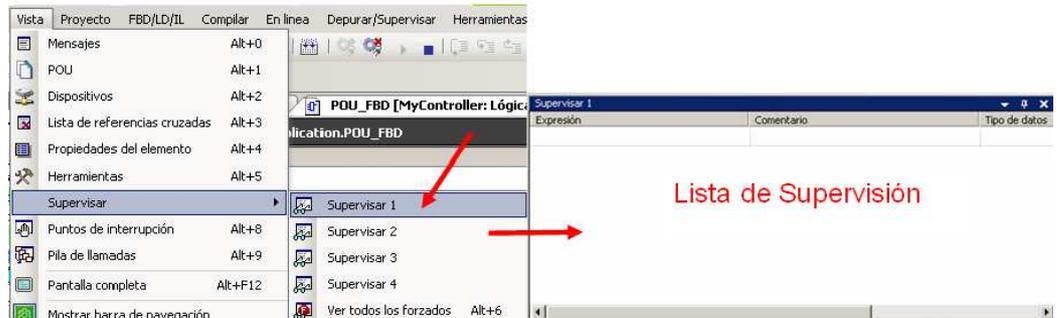
### ➤ Herramientas de Trazas:



## Monitorización de variables

### Lista de supervisión

Una **lista de supervisión** es una ventana donde se pueden monitorizar cualquier variable que hay en la aplicación, una variable local de cualquier POU, variables globales o físicas.



#### ➤ Lista de supervisión.

- Se pueden llamar cuatro listas de supervisión como máximo.
- Se añaden las variables a la lista de supervisión para monitorizarlas.
- Modificar o Forzar las variables que se han introducido.

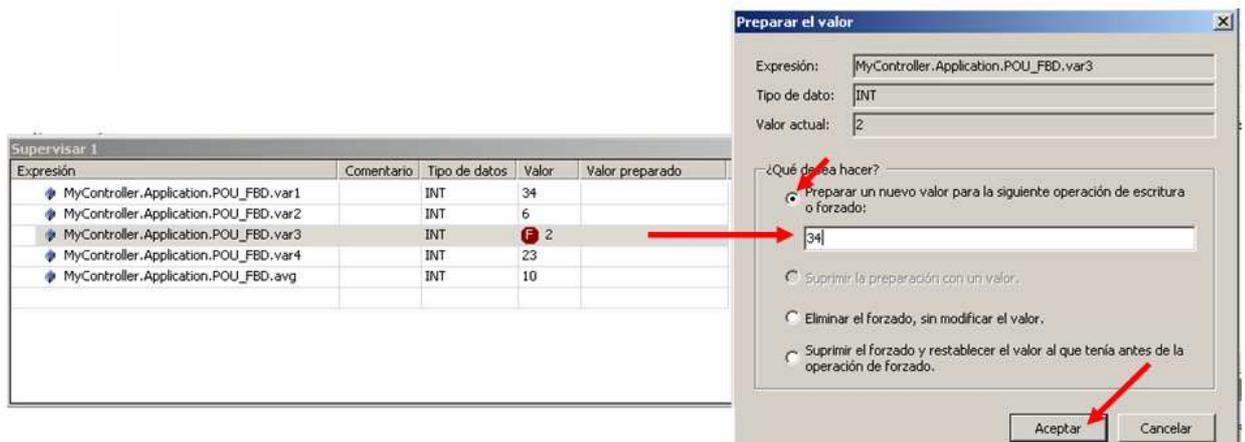


#### ▪ Las variables Booleanas se pueden forzar.

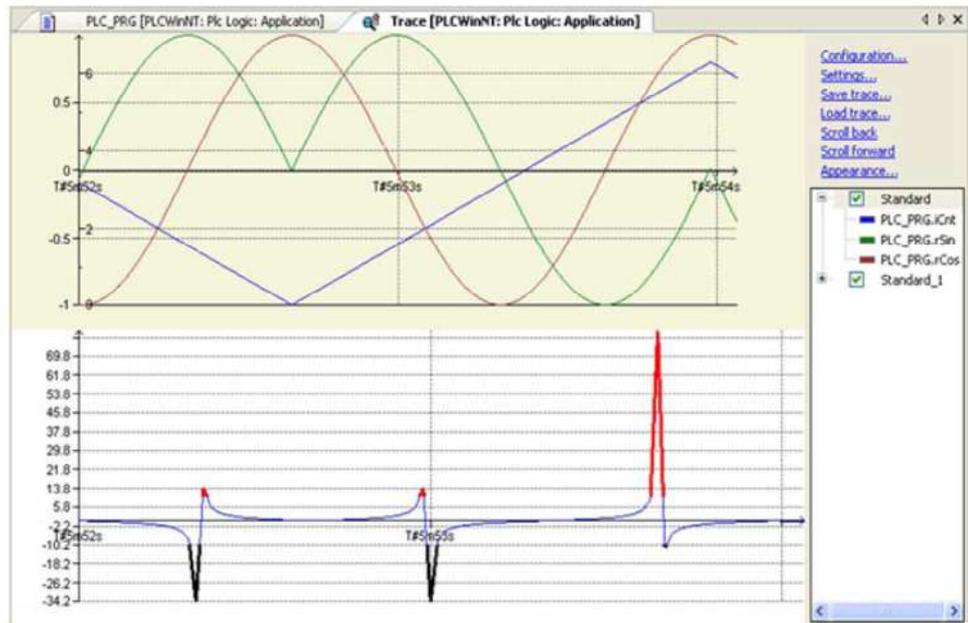
- También hay una lista que se pueden ver todas las variables que han sido forzadas en el sistema.

### Quitar forzado

Para quitar el forzado de las variables seleccionar el valor y hacer doble clic, aparece la ventana para seleccionar como se quiere quitar el forzado de la variable.



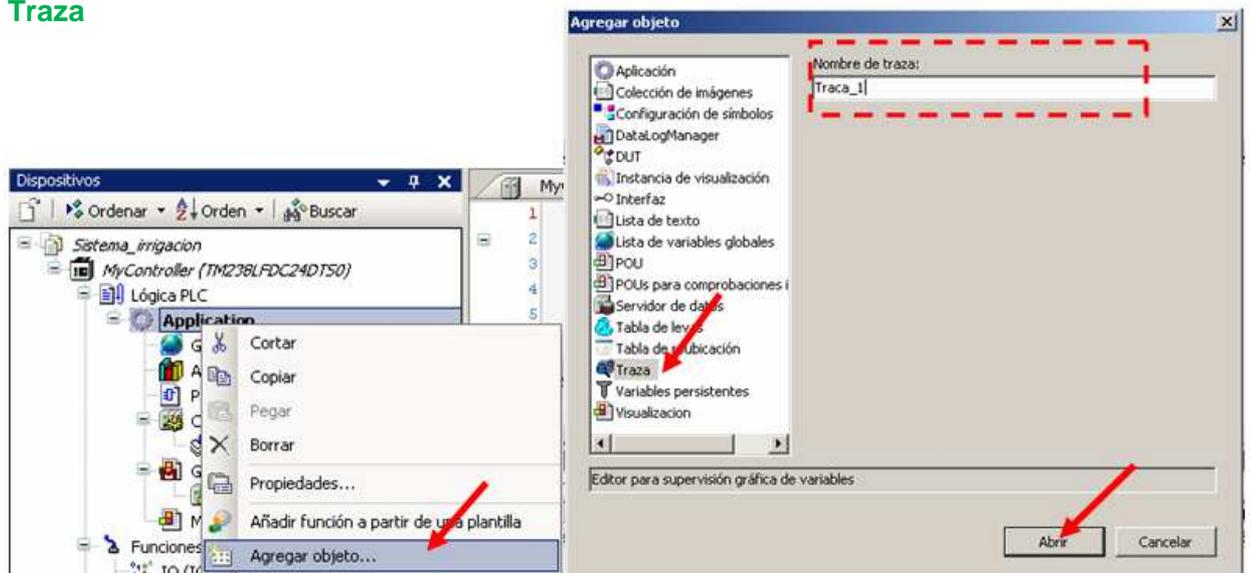
## Editor de Trazas



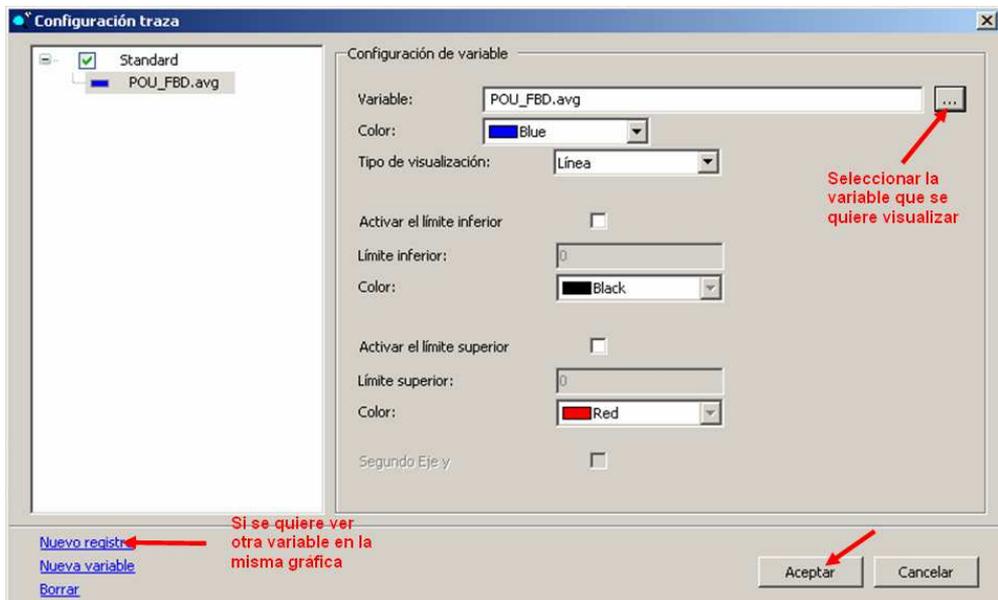
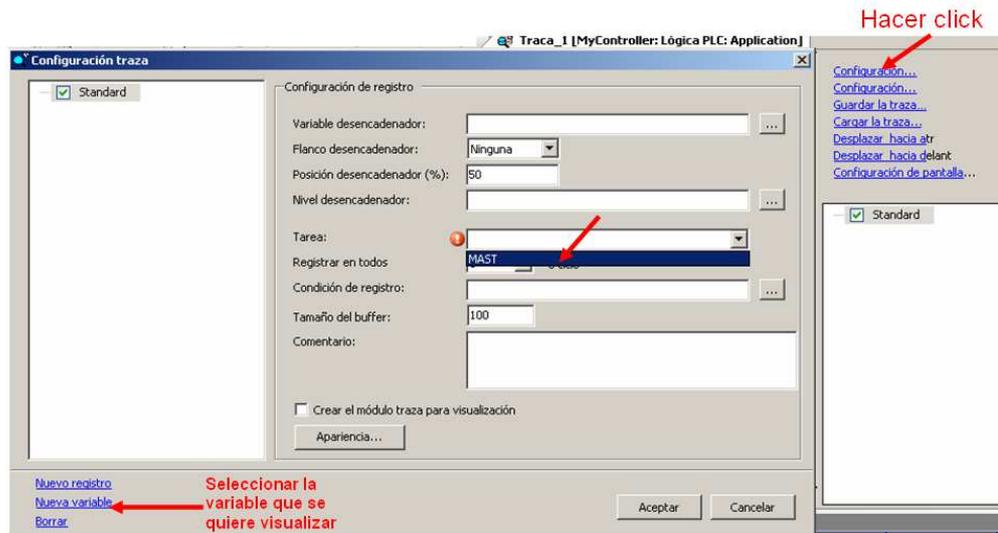
### ➤ El editor de trazas permite:

- Configurar y visualizar de manera gráfica los valores de las variables.
- Definir una o más trazas de variables y visualizarlas en el tiempo.
- Guardar estas trazas en un fichero para poder ser revisualizadas.

## Crear una Taza



## Configuración del editor de Trazas

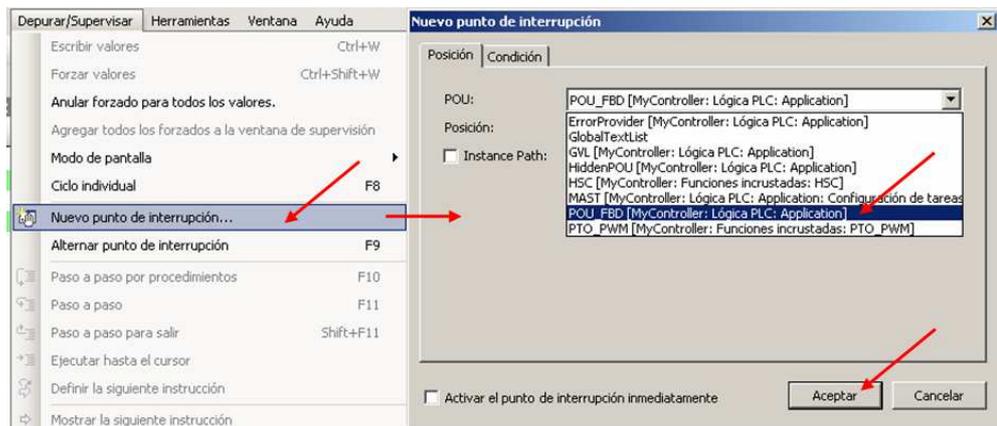


## Puntos de interrupción y ejecución Paso a Paso

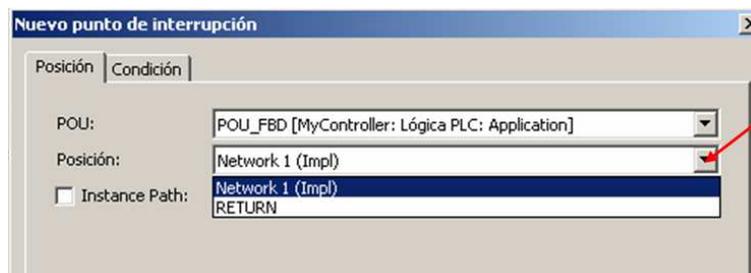
### Punto de Interrupción

Hay muchas opciones a la hora de utilizar los puntos de interrupción.

- Durante la ejecución del programa hace que este se quede parado en el punto de interrupción, esto permite ver como están las variables y la secuencia en ese punto de la ejecución.
- Para agregar un punto de interrupción, ir al menú contextual 'Depurar/Supervisor » Nuevo punto de interrupción'.
- En el campo 'POU' seleccionar en la parte de la aplicación que se desea crear el punto de interrupción.



Una vez seleccionado el POU o parte de la aplicación donde se desea insertar el punto de interrupción, en el campo 'Posición' seleccionaremos la línea específica del POU donde se quiere poner el punto de interrupción.



El punto de interrupción se ha insertado en esa posición.

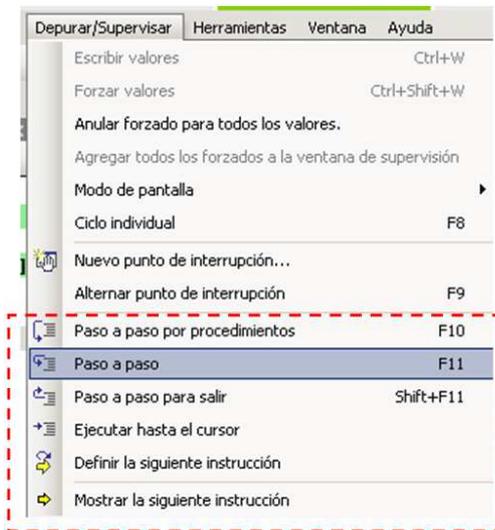
#### Símbolos de los puntos de interrupción

- Punto de interrupción habilitado
- ⏸ Ejecución parada en el punto de interrupción
- Punto de interrupción desactivado.

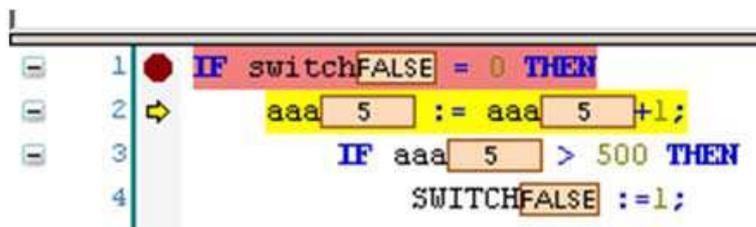
```

1  IF ConveyorCMDFALSE = TRUE THEN
2      IF B_MemoryFALSE = FALSE THEN
3          Bottle_EjectionTRUE:= 1; B_memoryFALSE:=1;
4      ELSE
5          bottle_EjectionTRUE:=0; B_memoryFALSE:=0;
6      END_IF
7  END_IF
8
9  RETURN
    
```

Cuando el punto de interrupción se ha alcanzado, se habilitan diferentes opciones en el menú contextual ‘**Depurar/Supervisar**’.

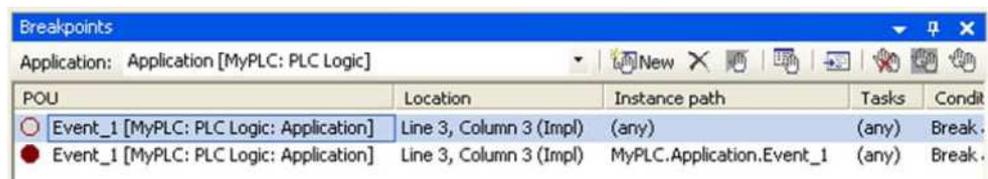


Estas opciones permiten elegir como se va a ejecutar la aplicación después de que se alcance el punto de interrupción: **Paso a Paso por procedimientos, Paso a paso, paso a paso para salir y ejecutar hasta el cursor**. La flecha amarilla indica que línea de programa se está ejecutando en ese momento.



También se puede abrir una ventana donde se pueden ver un listado de los diferentes puntos de interrupción introducidos y su estado. Para abrir la ventana hay que ir al menú ‘**Vista » Puntos de interrupción**’.

Si se quiere borrar uno de los puntos de interrupción que aparece en la lista. Seleccionar los puntos de interrupción en la lista y pulsar el **botón X** para borrarlo.





# Capítulo 12: Comunicación Modbus

## Descripción

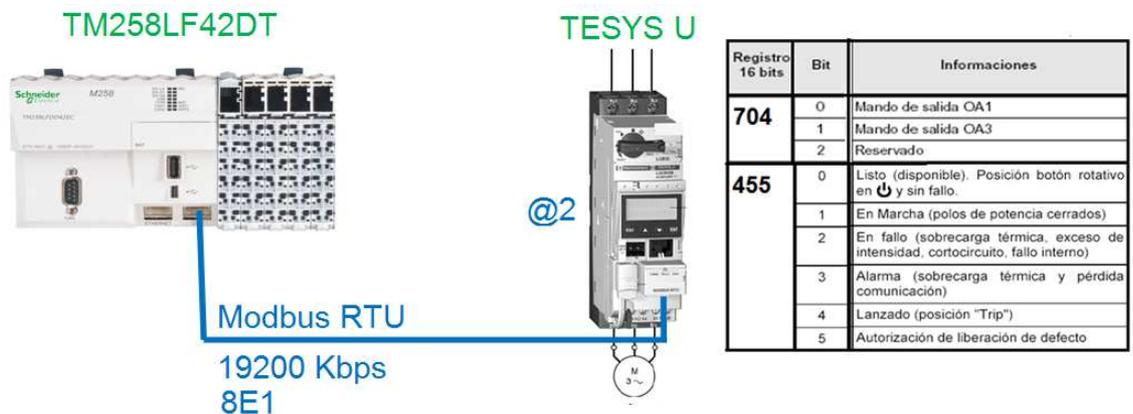
**Introducción** Este capítulo desglosa los dos caminos con los que se puede realizar una comunicación Modbus RTU, con otros equipos. Se presentan estas dos opciones a través de dos ejercicios prácticos.

**Este CAPÍTULO trata los siguientes temas:**

- Ejercicio - Modbus I/O Scanner..... 12-2
- Ejercicio – Mensajería Modbus ..... 12-5

## Ejercicio - Modbus I/O Scanner

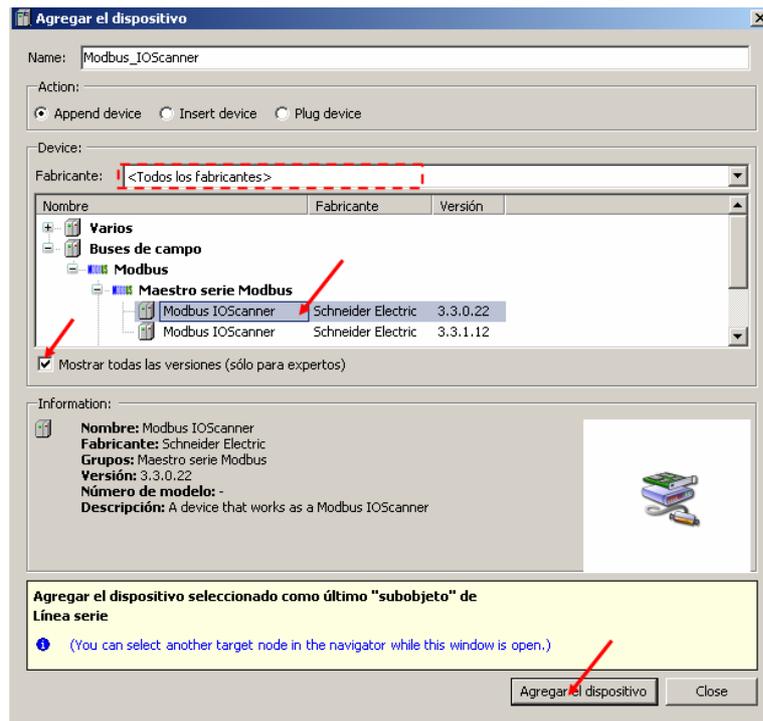
### 1 Crear una comunicación Modbus IO Scanner en la Línea serie 1.



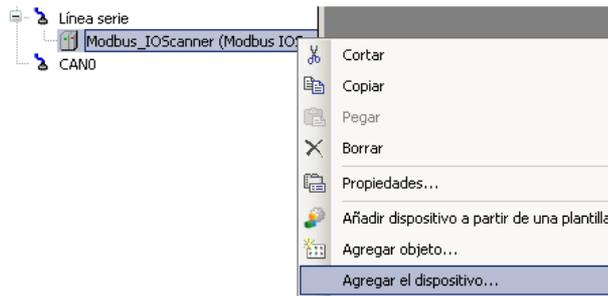
Borrar la configuración presente en la 'Línea serie 1'. Pulsar botón derecho del ratón sobre el ítem 'Línea serie 1' y seleccionar 'Añadir equipo'.



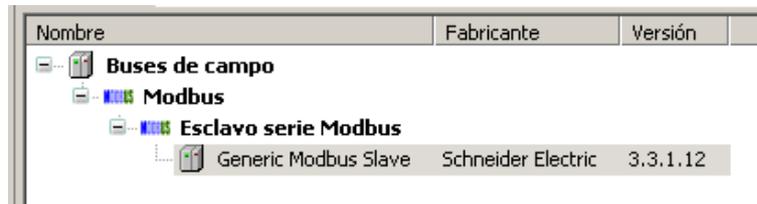
Añadir el elemento 'Modbus IO Scanner' de la lista de equipos a la 'Línea Serie 1'.



En la **Línea serie 1** se ha añadido el '**Modbus IO Scanner**' pulsar botón derecho y seleccionar '**Añadir Equipo**' para añadir el esclavo.



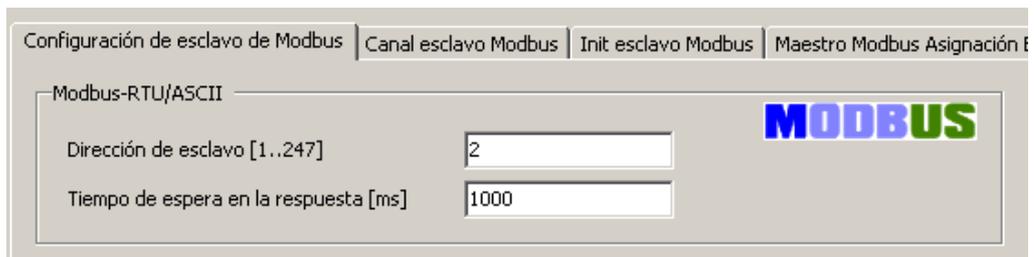
Añadir como esclavo modbus '**Generic Modbus Slave**' al IO Scanner.



Hacer doble clic en la **Línea Serie 1** para abrir la ventana de parámetros de comunicación del puerto donde se parametrizar la comunicación Modbus.

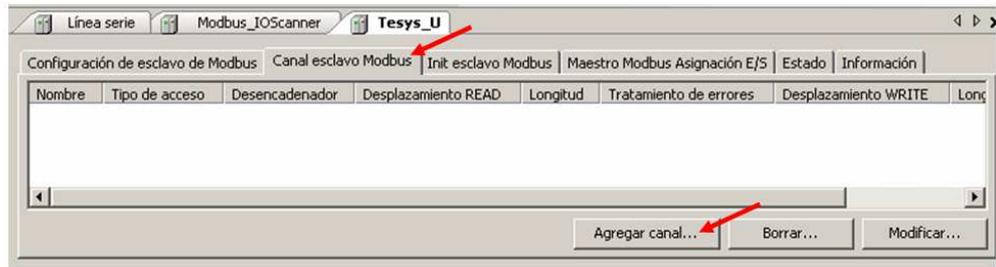


Hacer doble sobre el esclavo para abrir la configuración y asegurarse que en el campo '**Dirección de esclavo**' tiene el número de esclavo "2".

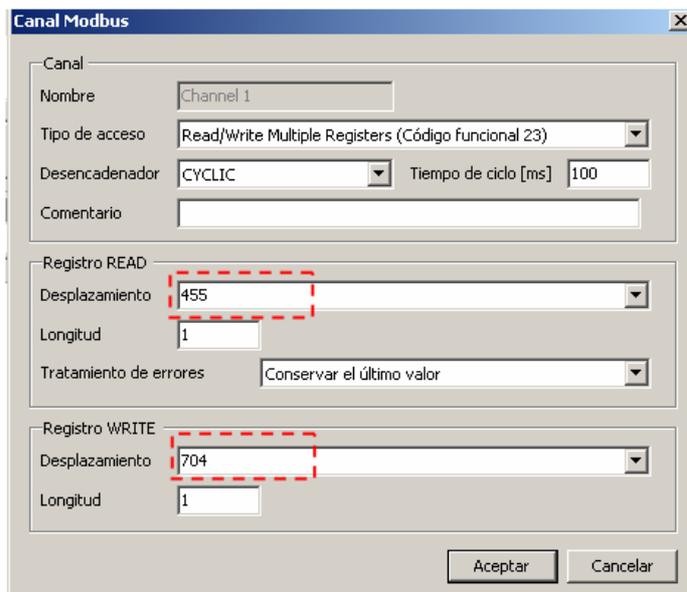


## 2 Continuar con la configuración la Línea serie 1

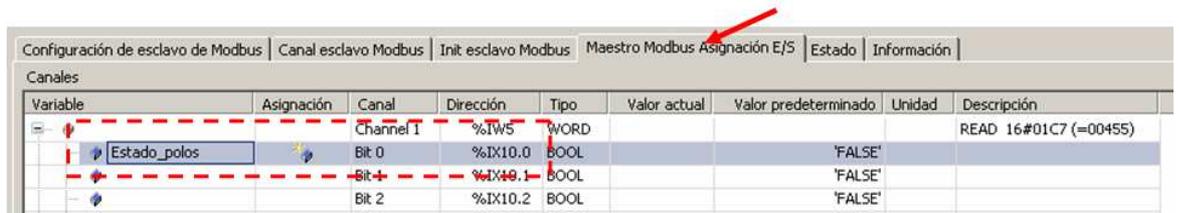
En el esclavo modbus seleccionar la pestaña 'Canal esclavo Modbus' y pulsar el botón 'Añadir Canal'.



En el canal en el campo 'Tipo de acceso' seleccionar **Read/Write Multiple Registers**, modificar el tiempo de ciclo **1000 ms**. Rellenar los campo de 'Desplazamiento' con la dirección que se quiere leer y escribir y la longitud en words (*en el caso del arrancador del Tesys-U el registro 455 no dice el estado del arrancador y en el registro 704 se realiza la activación del arrancador en un sentido y otro*).



Cuando se ha creado el canal de I/O scanning hay que crear las variables asociadas a ese canal para utilizarlas en el programa para esta comunicación Modbus. En este caso (*el bit 0 de la palabra 455 es el estado de los polos y en el bit 0 de la palabra 704 es la activación del arranque del contactor en directo*), seleccionar la pestaña 'Maestro Modbus Asignación E/S' y declarar las variables.



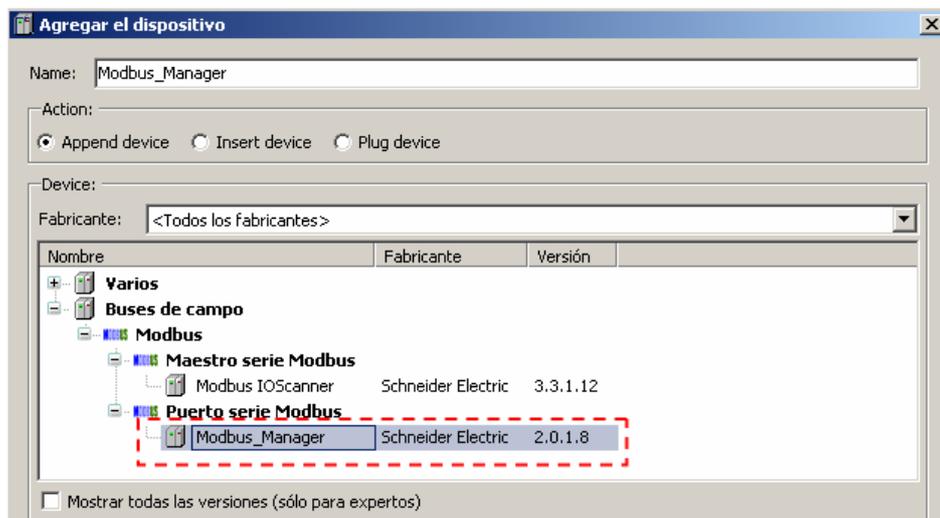
## Ejercicio – Mensajería Modbus RTU

### 1 Configurar los parámetros Modbus para utilización del Modbus por mensajería.

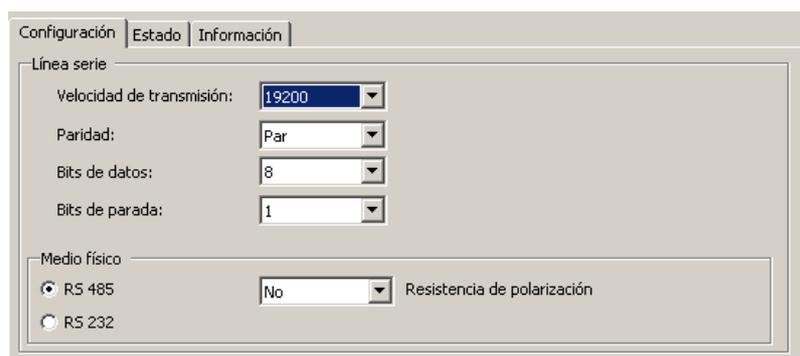
En este ejercicio ejemplo realizaremos la misma función que en el ejemplo del I/O scanning para ver las diferencias. Borrar la configuración presente en la Línea serie 1. Pulsar botón derecho del ratón sobre el ítem Línea serie 1 y seleccionar ‘Añadir equipo’.



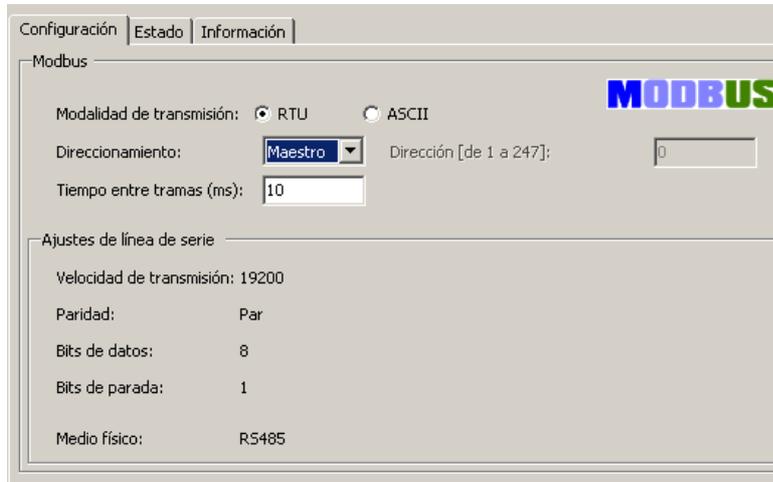
Añadir el elemento **Modbus Manager** de la lista de equipos a la ‘Línea Serie 1’.



Hacer doble clic en la Línea Serie 1 para abrir la ventana de parámetros de comunicación del puerto donde se parametrizar la comunicación Modbus.



Hacer doble clic en la Modbus Manager para abrir la ventana de configuración del gestor. Y en el campo 'Direccionamiento' seleccionar que este controlador será 'Maestro'.



Abrir el 'Administrador de bibliotecas' y agregar la librería 'toolbox'.



## 2 Crear un nuevo POU de programa en lenguaje FBD llamado Read\_Modbus.

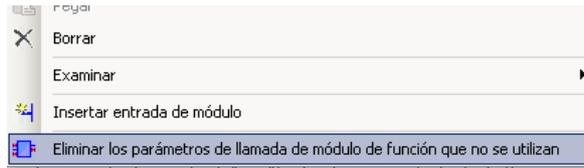
Insertar los FB's de la librería 'Toolbox' insertar 'ADDM' y 'READ VAR' para poder realizar la lectura. Y declarar las siguientes variables.

```

1 PROGRAM Modbus_Read
2 VAR
3     DIR_1: ADDM;
4     Ejecuta_lectura: BOOL; (*Variable que cuando es TRUE ejecuta la lectura por MODBUS*)
5     AD_Table: ADDRESS; (*Variable con formato dirección*)
6     READ_1: READ_VAR;
7     Buffer_lectura: ARRAY[0..1]OF WORD; (*En esta matriz se guardan los datos leídos *)
8     Lectura_OK: BOOL;
9 END_VAR

```

Eliminar los pines de la función que no se utilizan. Botón derecho 'Eliminar los parámetros de la llamada del módulo función que no se utiliza'



Crear la aplicación y si no hay errores conectarse al equipo y 'Cargar' el programa en el PLC. Activar la lectura petición de lectura poniendo a TRUE la variable 'Ejecutar lectura'.

Desplegar la variable 'Buffer\_lectura' y observar si cambiando de posición de la maneta del Tesis-U, se observa algún cambio. Añadir el resto de variables y modificar el programa como se muestra a continuación.

MyController.Application.Modbus_Read				
Expresión	Comentario	Tipo de datos	Valor	Valor preparado
DIR_1		ADDM		
AD_Table	Variable con formato dirección	ADDRESS		
Buffer_lectura	En esta matriz se guardan los datos que se han l...	ARRAY [0..1] OF WO...		
Buffer_lectura[0]		WORD	0	
Buffer_lectura[1]		WORD	0	
Lectura_OK		BOOL	FALSE	
RD_1		READ_VAR		
Ejecuta_lectura	Variable que cuando es TRUE ejecuta la lectura...	BOOL	TRUE	

Hacer las mismas pruebas con la petición de escritura. (Escribiendo un '1' en el Buffer de escritura el Tesis-U se tiene que poner en marcha)

```

1 PROGRAM Write_Modbus
2 VAR
3   Ejecuta_escritura: BOOL; (*Variable que cuando es TRUE ejecuta la escritura por MODBUS*)
4   ad_table: ADDRESS; (*Variable con formato dirección*)
5   WR_1: WRITE_VAR;
6   Buffer_escritura: ARRAY[0..1]OF WORD; (*En esta matriz se escriben los datos que se quieren escribir leído *)
7   Escritura_OK: BOOL;
8   DIR_2: ADDM;
9 END_VAR
10

```

---

1 \* PETICIÓN DE ESCRITURA VIA COMUNICACION MODBUS POR '1.2' EL PUERTO SERIE 1, AL ESCLAVO 2 \*

```

      DIR_2
      WR_1
      ADDM                                WRITE VAR
      ad_table ← AddrTable Done           Execute Done → Escritura_OK
      Ejecuta_escritura → Execute         ad_table → Addr
      '1.2' → Addr                        10 → Timeout
                                           0 → ObjType
                                           704 → FirstObj
                                           1 → Quantity
      ADR(Buffer_escritura) → Buffer

```



# Capítulo 13: CANopen & Remote I/O

## Descripción

**Introducción** Este capítulo trata de temas como: Introducción al CANopen, Configuración del una red CANopen en SoMachine, Comunicación CANopen a través de PDO y Comunicación CANopen a través de SDO.

### Este CAPÍTULO trata los siguientes temas:

- CANopen Basics – Network & Object Directory ..... 13-2
- PDO's y SDO's ..... 13-2
- Añadir el maestro CANopen al controlador ..... 13-2
- Baudrate del bus CANopen ..... 13-2
- Añadir un esclavo bus CANOpen..... 13-2
- Configuración un esclavo en CANOpen ..... 13-2
- Asignación de PDO's ..... 13-28
- Mapping de las variables de CANOpen ..... 13-2
- Service Data Object (SDO) ..... 13-2
- Ejercicio – Lectura/escritura SDO's por programa .... 13-2
- Ejercicio – Control Lexium32 en CanOpen con MFB's (Motion Funtion Blocks – PLCOpen) ..... 13-2
- Ejercicio – Inserción de Lexium32 en CanOpen con la plantilla de dispositivos..... 13-2
- DTM's – FDT ..... 13-2
- Ejercicio – Inserción de Advantys OTB en bus CANOpen con DTM ..... 13-2

## CANopen Básico – Red & Objetos

### Red CAN



**CANOpen** – es un estándar internacional que define un protocolo de comunicación industrial para conectarse con equipos industriales vía red CAN.

**CANbus** – Cable de par trenzado apantallado, con una resistencia de terminal de **120 Ω**, señales **CAN\_H**, **CAN\_L**, Longitud máxima del bus de **1000m**, Velocidad de transmisión depende de la longitud del bus, como máximo se pueden tener **128**.

- No puede haber dos equipos con el mismo número de esclavo.
- Todos los esclavos tienen que estar configurados con la misma velocidad.

### Directorio de objetos

**Directorio de objetos (OD)** – describe todas las funcionalidades del producto.

Indice	Objeto	Descripción
0x0000	Reservado	
0x0001 – 0x009F	Área de tipos de datos	Define los tipos de datos que se utilizan en el equipo: Bytes, Words, Enteros.
0x00A0 – 0x0FFF	Reservado	
0x1000 – 0x1FFF	Área del perfil de comunicación	Describe los objetos asociados a la comunicación CANopen.
0x2000 – 0x5FFF	Área específica del perfil de la empresa	Describe los objetos para la aplicación que usa la compañía creadora del equipo.
0x6000 – 0x9FFF	Área estandarizada del perfil del equipo	Describe los objetos estandarizados por la CiA para la aplicación.
0xA000 – 0xFFFF	Reservado	

El diccionario de objetos (OD) es un grupo secuencial de objetos compuesto por:

- Un índice de **16 bits** y en algunos casos un subíndice de **8 bits**.
- Un archivo formato **ASCII**, llamado **EDS** (Electronic Data Sheet).

## PDO's y SDO's

### Partes integradas de CANopen

CANopen define una capa de aplicación y un perfil de comunicación basado en CAN. CANopen define los siguientes objetos de comunicación (mensajes): los **PDO** (*process data object*) y **SDO** (*service data object*).

### PDO's

---

Los objetos de datos de proceso u objetos de comunicación implícita se utilizan para leer/escribir rápidamente datos de proceso, para aplicaciones de tiempo real.

- Un PDO puede transportar una carga útil de 8 bytes, que es la carga máxima de una trama CAN. Por defecto, cada nodo tiene acceso a sólo 4 PDO's
- La transmisión de un PDO utiliza el modelo de productor y consumidor de CAN ampliado mediante transferencias síncronas.
- Los objetos mapeables permitidos por el equipo se encuentran contenidos en el archivo EDS que se proporciona con cada equipo de CANopen.
- Se seleccionan los registros que se desean leer/escribir desde el diccionario de objetos y se añaden a los PDO's de emisión y PDO's de recepción.

### SDO's

---

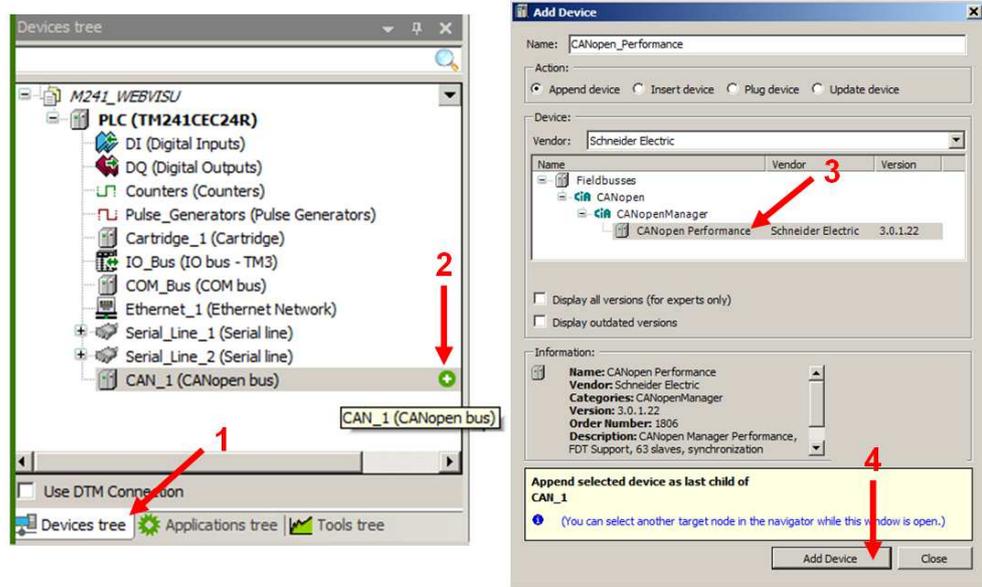
Los objetos de datos de servicio u objetos de comunicación explícita tienen con fin la transmisión de parámetros y permiten petición de lectura y petición de escritura.

- La comunicación de SDO emplea una comunicación de homólogo a homólogo, con uno de los homólogos que actúa como servidor y el otro como cliente. Se acusa el recibo de cada SDO.
- Un SDO no tiene límites de longitud. Si la carga útil no cabe en la trama CAN, se dividirá en varias tramas CAN.
- Es posible acceder a los datos de los nodos mediante SDO's en cualquier momento y sin haber configurado nada previamente mediante una petición en el programa.

## Añadir el maestro CANopen al controlador

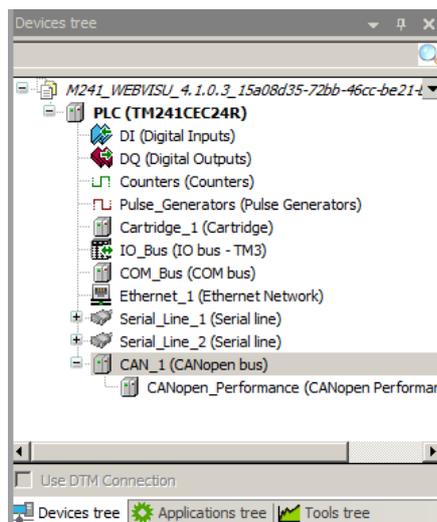
### CANopen Master

Si el equipo configurado tiene bus CanOpen, si se quiere utilizar el CANopen hay que añadir a la configuración el 'CANopen Performance', indicando que el puerto está habilitado y que el controlador va a hacer de maestro del bus. Para incluir el maestro CanOpen, en la ventana de dispositivo seleccionar el puerto CanOpen 'CAN\_1', botón derecho seleccionar 'Añadir Dispositivo' ó pulsar al icono de '+' que sale a la derecha.



En la ventana flotante que aparece, Seleccionar el dispositivo de CAN 'CANopen Performance', y pulsar el botón 'Add Device'.

Cuando se haya agregado aparecerá colgado del bus CANOPEN en la ventana de dispositivos.



## Baudrate del bus CANopen

### Establecer la velocidad

Para establecer la velocidad de transmisión del bus Canopen en el maestro (Todos los equipos se tendrán que parametrizar con la misma velocidad de transmisión baudrate que el maestro).

### Maestro

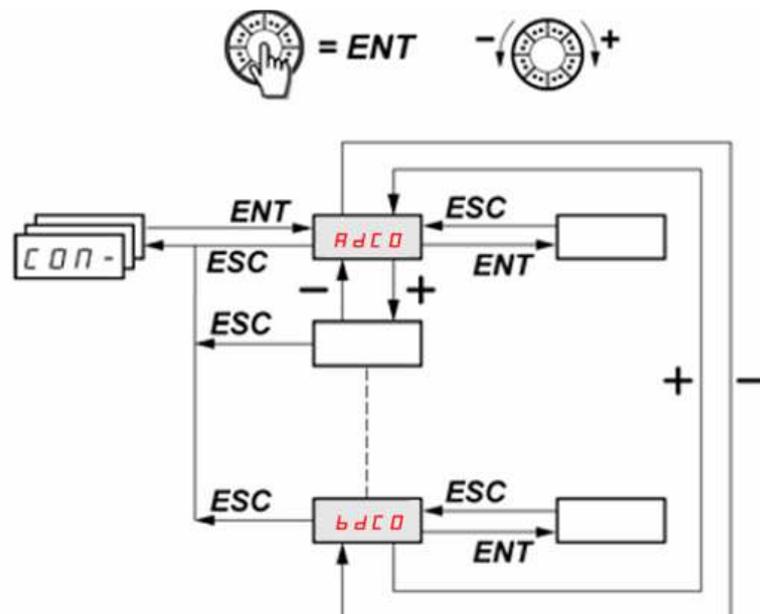
Para establecer la velocidad en el maestro hacer doble clic sobre el bus CANOPEN de la ventana de dispositivos y en el área de trabajo aparecerá la ventana de configuración en el campo '**Velocidad de transmisión**' elegir la velocidad.



### Equipos Remotos

En los equipos remotos (esclavos) además de establecer la dirección de esclavo, será necesario configurar la velocidad de transmisión igual que el maestro. El procedimiento de configuración depende del equipo.

**Ejemplo** con el variador de velocidad con el ATV312. Seleccionar el menú del variador **COM-**, en el parámetro '**AdCO**' se configura la dirección Canopen del equipo y en '**bdCO**' se establece la velocidad.



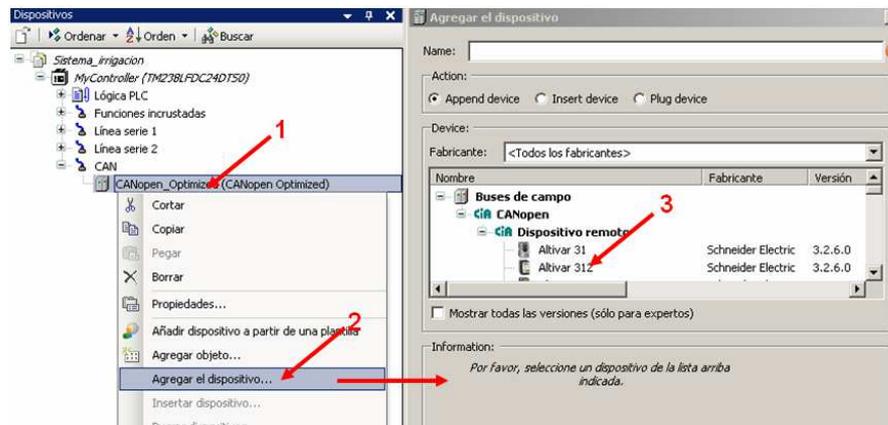
# Añadir un esclavo al bus CANopen

## Añadir Equipos

SoMachine ya viene con una librería de EDS de equipos ya instalada. Si el equipo no está dentro del catálogo de equipos ya instalado es posible añadirlo instalando el archivo .EDS suministrado por el fabricante del equipo.

Un **Maestro** puede comunicarse con múltiples equipos de campo (127 máx.). Cada equipo tiene que estar añadido dentro de la configuración Canopen del maestro en el SoMachine.

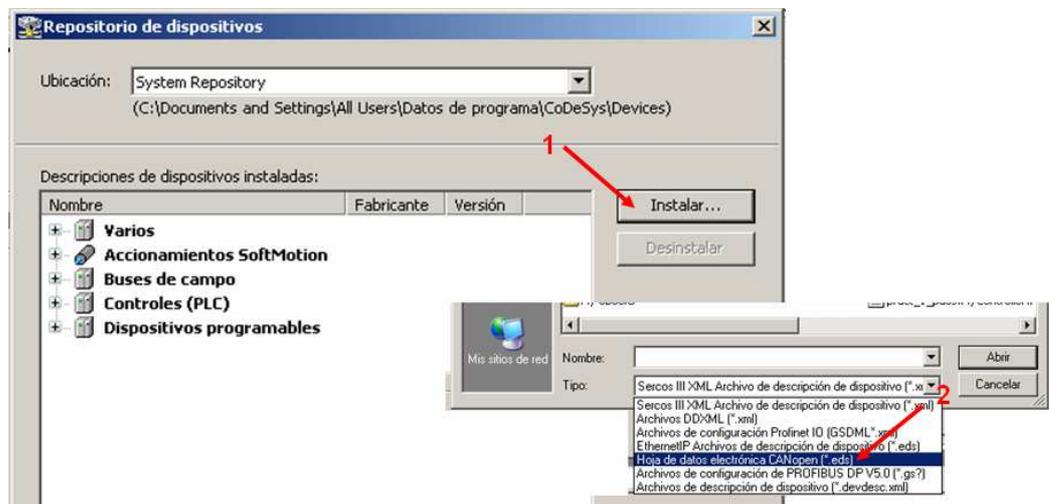
Para añadir un equipo seleccionar el **CANopen Optimized** y con el botón derecho seleccionar 'Añadir Equipo'. Aparecerá una lista de equipos agrupados por diferentes tipos de equipos donde se seleccionará el equipo.



Cuando el equipo esté añadido, este aparecerá colgado del **Maestro**.



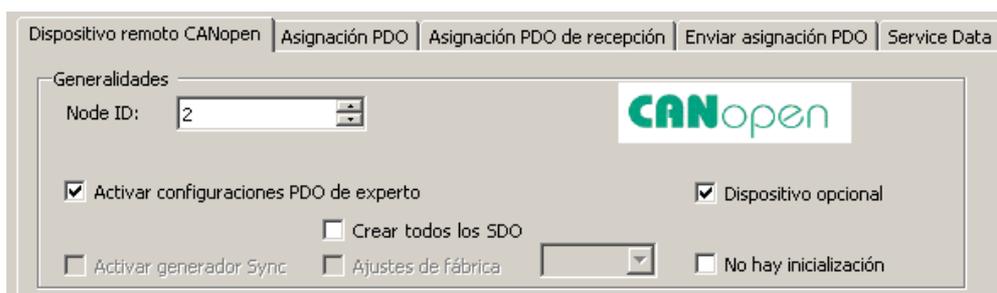
Adicionalmente se podrá añadir nuevos equipos a la lista, utilizando el 'Repositorio de Dispositivos' que se encuentra dentro del menú 'Herramientas' del menú contextual.



## Configuración de un esclavo CANopen

### Dirección del esclavo

Dentro de la configuración, dentro de la ventana de dispositivos, hacer doble clic en el equipo remoto añadido y para asignarle la dirección de esclavo Canopen en '**Node ID**'. También habilitar la opción '**Activar configuraciones PDO de experto**'.



Dispositivo remoto CANopen | Asignación PDO | Asignación PDO de recepción | Enviar asignación PDO | Service Data

Generalidades

Node ID: 2

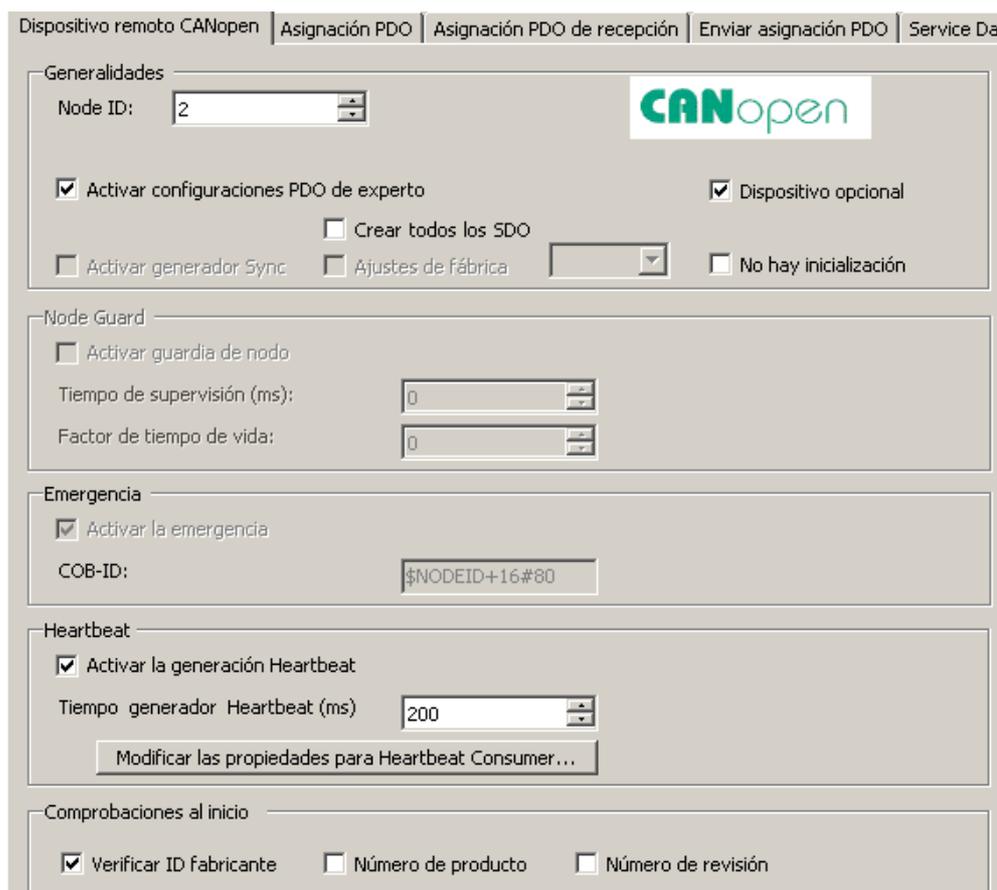
**CANopen**

Activar configuraciones PDO de experto  Dispositivo opcional

Crear todos los SDO

Activar generador Sync  Ajustes de fábrica  No hay inicialización

### Configuración PDO experto



Dispositivo remoto CANopen | Asignación PDO | Asignación PDO de recepción | Enviar asignación PDO | Service Data

Generalidades

Node ID: 2

**CANopen**

Activar configuraciones PDO de experto  Dispositivo opcional

Crear todos los SDO

Activar generador Sync  Ajustes de fábrica  No hay inicialización

Node Guard

Activar guardia de nodo

Tiempo de supervisión (ms): 0

Factor de tiempo de vida: 0

Emergencia

Activar la emergencia

COB-ID: \$NODEID+16#80

Heartbeat

Activar la generación Heartbeat

Tiempo generador Heartbeat (ms): 200

Modificar las propiedades para Heartbeat Consumer...

Comprobaciones al inicio

Verificar ID fabricante  Número de producto  Número de revisión

El modo experto nos permite monitorizar el modo del bus, añadir PDO's, modificar los objetos de los PDO's, mapearlos y crear peticiones de SDO's.

# Asignación de los PDO

## Asignación de PDO

La pestaña de Asignación de PDO permite ver y seleccionar los PDO preconfigurados que están en uso. Esto nos permite añadir y modificar los objetos que hay dentro de cada uno de los PDO's.

## Lista de asignación PDO

Selección PDO (RPDO) de recepción				Selección PDO (TPDO) de emisión			
Nombre	Índice	Subíndice	Longitud...	Nombre	Índice	Subíndice	Longitud...
<input type="checkbox"/> 1st Receive PDO	16#1400	16#00	16	<input type="checkbox"/> 1st Transmit PDO	16#1800		
Drivecom command r	16#6040	16#00	16	Drivecom status regist	16#6041	16#00	16
<input checked="" type="checkbox"/> 6th Receive PDO	16#1405			<input checked="" type="checkbox"/> 6th Transmit PDO	16#1805		
Drivecom command r	16#6040	16#00	16	Drivecom status regist	16#6041	16#00	16
Frequency Reference	16#2037	16#03	16	Extended status regist	16#2002	16#07	16
				Control effort	16#6044	16#00	16
				Logic Input/Output im	16#2016	16#29	16

- La pestaña de 'Asignación PDO' presenta los PDO's preconfigurados con ese equipo. Con los objetos que han sido insertados por defecto.
- Seleccionar los números de PDO's que se desean transmitir y recibir.

## Asignar PDO de recepción ó envío.

En la pestaña 'Asignación PDO de Recepción' permite añadir PDO's y modificar o ampliar los objetos que hay dentro de los PDO's ya existentes. En esta secuencia se ha añadido al PDO número 6 el objeto con el índice 2016 y subíndice 29, que nos da la información de la entradas y salidas del ATV312.

Después de cambiar los PDO's se tiene que declarar las variables. En la pestaña 'Canopen asignación E/S' se crean las variables globales que se utilizarán para la comunicación.

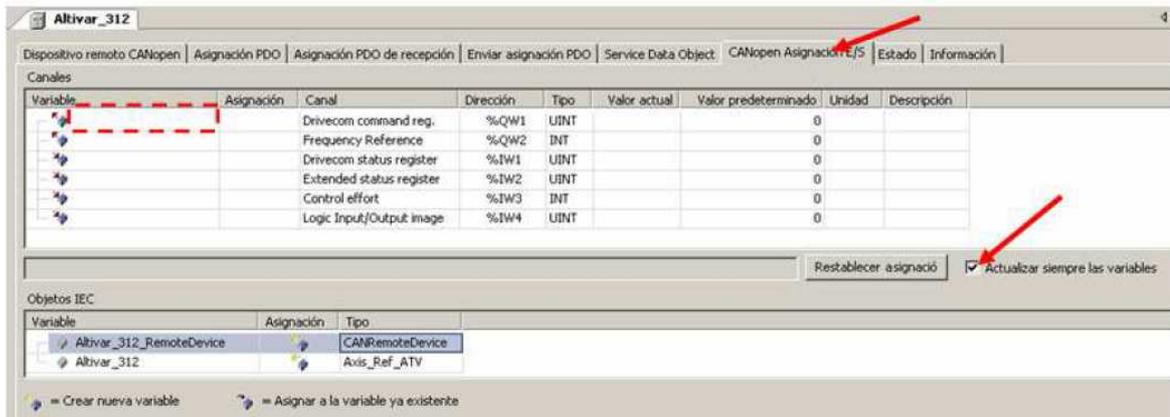
Variable	Asignación	Canal	Dirección	Tipo	Valor actual	Valor predeterminado	Unidad	Descripción
ATV312_ControlWord		Drivecom command reg.	%QW1	UINT		0		
ATV312_Consigna		Frequency Reference	%QW2	INT		0		
ATV312_Salidas		Logic Input/Output image	%QW3	UINT				
ATV312_StatusSord		Drivecom status register	%IW1	UINT		0		
ATV312_Velocidad_Actual		Extended status register	%IW2	UINT		0		
ATV312_Entradas		Control effort	%IW3	INT		0		
		Logic Input/Output image	%IW4	UINT		0		

Restablecer asignación  Actualizar siempre las variables

## Mapping de las variables en CanOpen

### Asignación de las variables de comunicación

En la pestaña '**Canopen asignación E/S**' se crean las variables globales que se utilizarán para la comunicación. De esta manera se asigna a una variable a cada uno de los objetos que se han incluido en el PDO.



Una vez conectado en esta pestaña se puede ver el estado de las variables de comunicación.

# Service Data Object (SDO)

## Secuencia de control

Seleccionar los SDO's que se configuraran en la secuencia de control que se enviarán desde el **Maestro** al **Esclavo**. También se puede configurar el procedimiento de control que tiene que pasar en caso de fallo durante la transmisión.

Dispositivo remoto CANopen	Asignación PDO	Asignación PDO de recepción	Enviar asignación PDO	Service Data Object	CANopen Asignación E/S	Estado	Información
Línea	Índice:subíndice	Nombre	Valor	Longitud en bits	Interrupción por error	Salta a línea en presencia de error	Línea siguiente
1	16#100C:16#00	Set Guardtime	16#00000000	16	<input type="checkbox"/>	<input type="checkbox"/>	0
2	16#100D:16#00	Set Lifetime	16#00000000	8	<input type="checkbox"/>	<input type="checkbox"/>	0
3	16#1016:16#01	Set Heartbeat Consumer	16#007F012C	32	<input type="checkbox"/>	<input type="checkbox"/>	0
4	16#1017:16#00	Set Heartbeat Producer	16#000000C8	16	<input type="checkbox"/>	<input type="checkbox"/>	0
5	16#1400:16#01	Disable PDO	16#80000202	32	<input type="checkbox"/>	<input type="checkbox"/>	0
6	16#1405:16#01	Disable PDO	16#80000302	32	<input type="checkbox"/>	<input type="checkbox"/>	0
7	16#1405:16#02	Set transmission type	16#FF	8	<input type="checkbox"/>	<input type="checkbox"/>	0
8	16#1605:16#00	Clear pdo mapping	16#0	8	<input type="checkbox"/>	<input type="checkbox"/>	0
9	16#1605:16#01	Set Mapping	16#60400010	32	<input type="checkbox"/>	<input type="checkbox"/>	0
10	16#1605:16#02	Set Mapping	16#20370310	32	<input type="checkbox"/>	<input type="checkbox"/>	0
11	16#1605:16#00	Set number of pdos	16#02	8	<input type="checkbox"/>	<input type="checkbox"/>	0
12	16#1405:16#01	Set and enable COB-ID	16#00000302	32	<input type="checkbox"/>	<input type="checkbox"/>	0
13	16#1800:16#01	Disable PDO	16#80000182	32	<input type="checkbox"/>	<input type="checkbox"/>	0
14	16#1805:16#01	Disable PDO	16#80000282	32	<input type="checkbox"/>	<input type="checkbox"/>	0
15	16#1805:16#02	Set transmission type	16#FF	8	<input type="checkbox"/>	<input type="checkbox"/>	0
16	16#1805:16#03	Set inhibit time	16#001E	16	<input type="checkbox"/>	<input type="checkbox"/>	0
17	16#1805:16#05	Set event time	16#0064	16	<input type="checkbox"/>	<input type="checkbox"/>	0
18	16#1A05:16#00	Clear pdo mapping	16#0	8	<input type="checkbox"/>	<input type="checkbox"/>	0
19	16#1A05:16#01	Set Mapping	16#60410010	32	<input type="checkbox"/>	<input type="checkbox"/>	0
20	16#1A05:16#02	Set Mapping	16#20020710	32	<input type="checkbox"/>	<input type="checkbox"/>	0
21	16#1A05:16#03	Set Mapping	16#60440010	32	<input type="checkbox"/>	<input type="checkbox"/>	0
22	16#1A05:16#04	Set Mapping	16#20162910	32	<input type="checkbox"/>	<input type="checkbox"/>	0

## Lectura y escritura del SDO

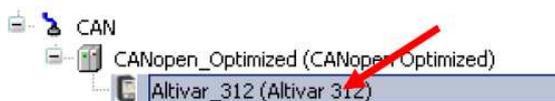
Los objetos de datos de servicio u objetos de comunicación explícita tienen con fin la transmisión de parámetros y permiten petición de lectura y petición de escritura.

- o La comunicación de SDO emplea una comunicación de homólogo a homólogo, con uno de los homólogos que actúa como servidor y el otro como cliente. Se acusa el recibo de cada SDO.
- o Un SDO no tiene límites de longitud. Si la carga útil no cabe en la trama CAN, se dividirá en varias tramas CAN.
- o Es posible acceder a los datos de los nodos mediante SDO's en cualquier momento (online) ó previamente mediante una petición en el programa,

## Lectura y escritura del SDO online

Para acceder a la lectura/ escritura online, es posible realizar estas peticiones desde el SoMachine, cuando se está monitorizando el programa de manera online.

Para ello se tendrá que seleccionar el esclavo Canopen al cual se desea realizar la petición y hacer doble clic para entrar en la ventana de configuración.

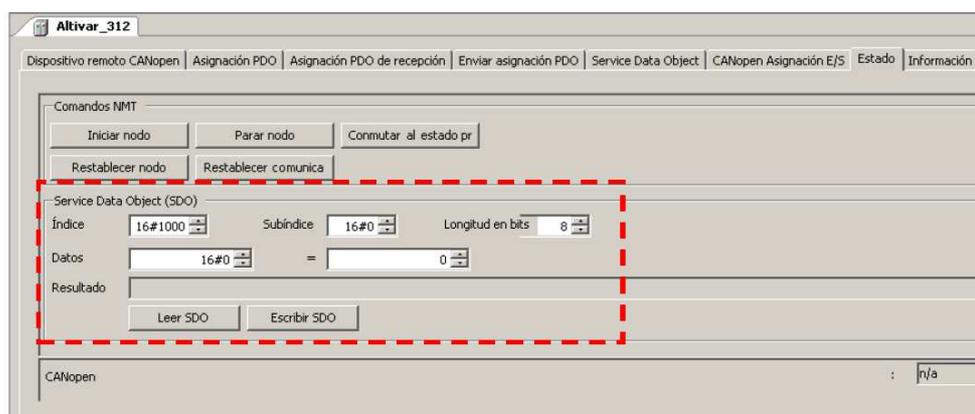


Dentro de la ventana del equipo, seleccionar la pestaña 'Estado' en esta aparece un área con todo lo necesario para realizar la lectura y escritura de comando SDO.

### ➤ Ejemplo de lectura/escritura de un SDO

Para realizar la lectura de algunos parámetros vía SDO, se tiene que saber el objeto de Canopen que se desea leer o escribir. Existen manuales donde se muestran estos datos.

CANopen address	Code	Read/Write	Name/Description/Possible values
2002 / 5	LCr	R	Current in the motor Unit: 0.1 A



En este caso se desea hacer una lectura de la corriente del motor, el manual nos indica la dirección de canopen que está compuesta por el **índice = 2002** y **subíndice = 5**. También nos indica si este objeto de canopen es de lectura ó lectura/escritura, en este caso solo es de lectura **R=Read**.

- **Índice y Subíndice:** Se pondrá el índice del parámetro que se desea leer (*normalmente el valor de índice y el subíndice en los manuales se ponen en Hexadecimal por lo que se tendrá que escribir en los campos con el prefijo 16# para indicar al software que está en formato hexadecimal*).

**Índice:= 16#2002 y subíndice:= 16#05**

- **Longitud en bits:** Hay que indicar la longitud en bits que tiene este parámetro (en este caso *16 bits = 1 word*).

- **Datos:** En el caso que se una petición de escritura, en este campo hay que especificar los datos que se van a escribir en el parámetro.

- **Resultado:** Se mostrarán los datos de lectura del parámetro, si se ha realizado una petición de lectura. Si la petición es de escritura se mostrarán los datos que se han escrito.

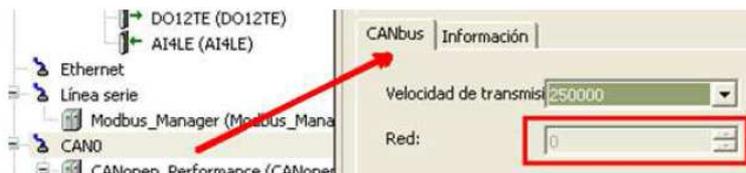
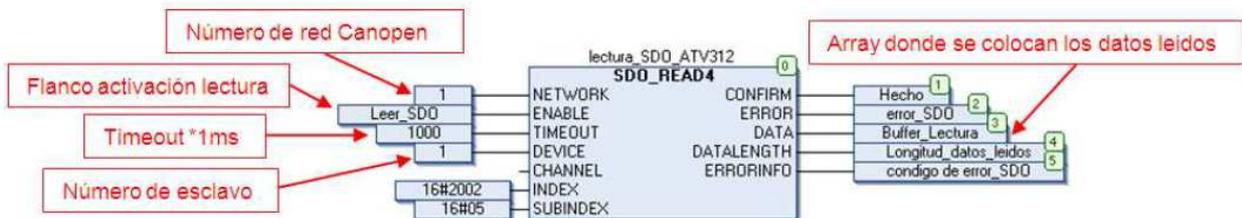
**Si el valor que se muestra en el campo 'Resultado' es 16#0, significa que se ha producido un error y que no se ha realizado la petición.**

Por último hay que pulsar el botón '**Leer SDO**' para realizar una petición de lectura y '**Escribir SDO**' para la petición de escritura.

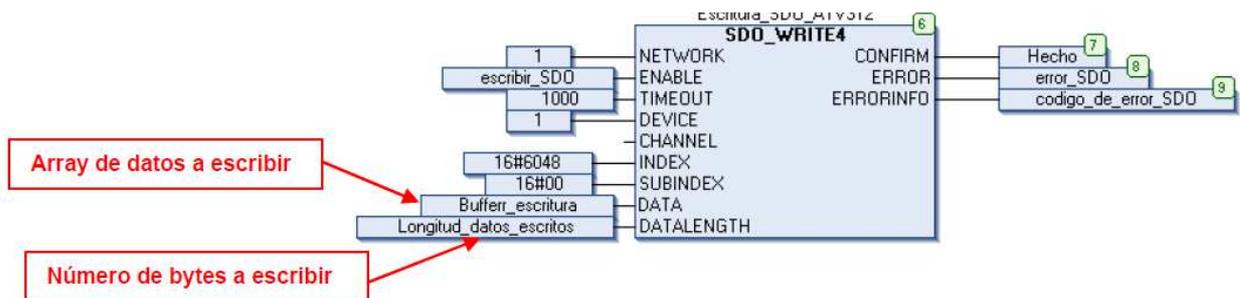
## Ejercicio – Lectura/ Escritura de SDO's por programa

Para la lectura y escritura de parámetros vía Canopen se disponen de unas instrucciones que se pueden incluir en el programa que ejecutan dicha acción SDO's.

Una vez configurado el esclavo, hay que realizar los pasos:

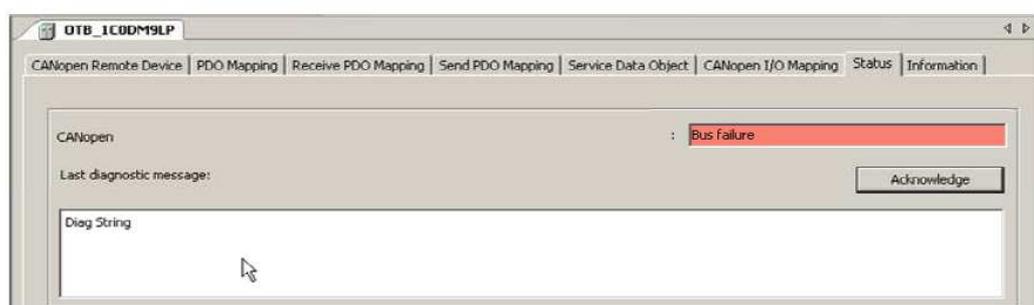


**Nota:** El número de **Network** CAN no es el mismo que el número de **Red** especificado en el configurador de CAN. Para calcular el número de **Network**, este es igual al número de RED + 1.



### Estado del esclavo CANopen

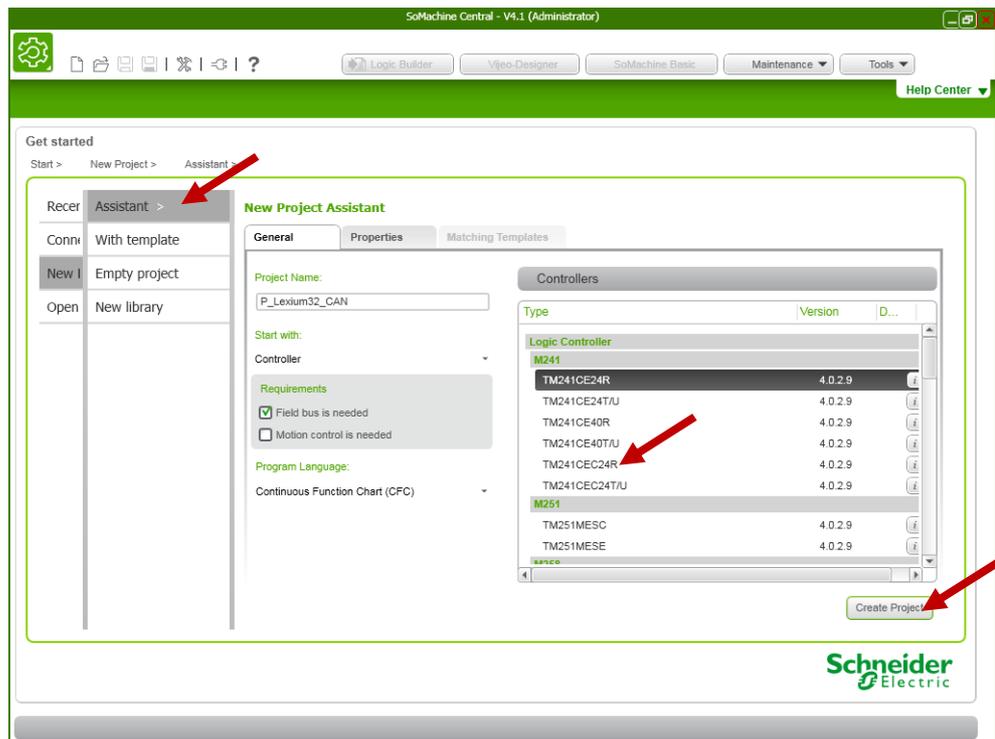
En la pestaña '**Estado**', también se muestra el estado de la operación del bus y si el bus está en error.



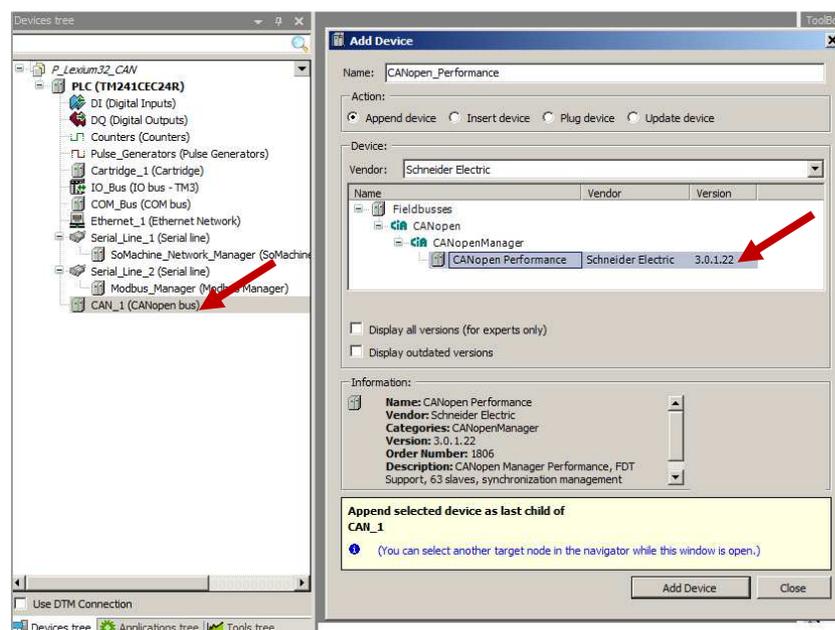
## Ejercicio – Control Lexium32 en CanOpen con MFB's (Motion Function Blocks – PLCOpen).

A continuación se detallan los pasos a seguir para configurar, en SoMachine 4.1, con un M241 y un LXM32 en CanOpen.

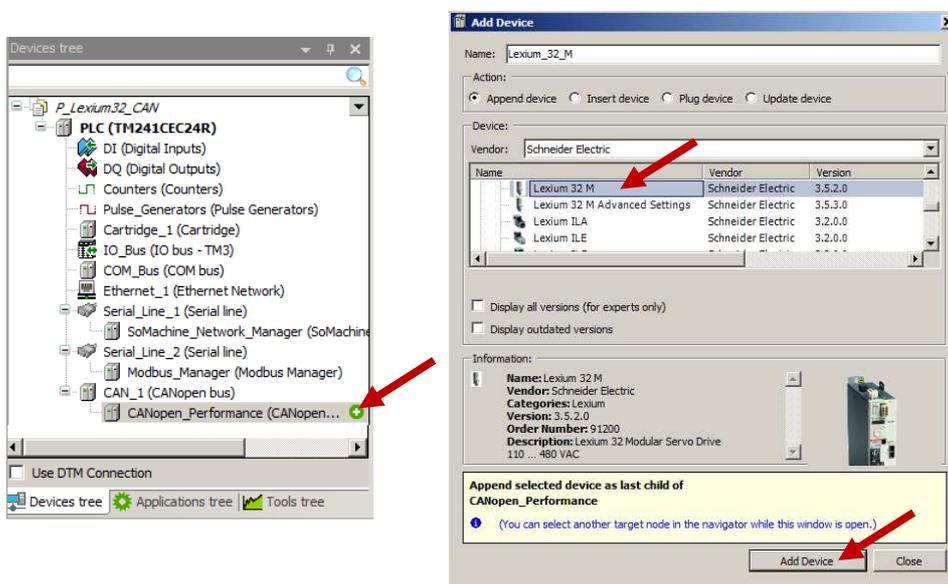
1. Desde la pantalla de inicio, utilizando el asistente de creación de proyecto, se inserta el **TM241CEC24R**.



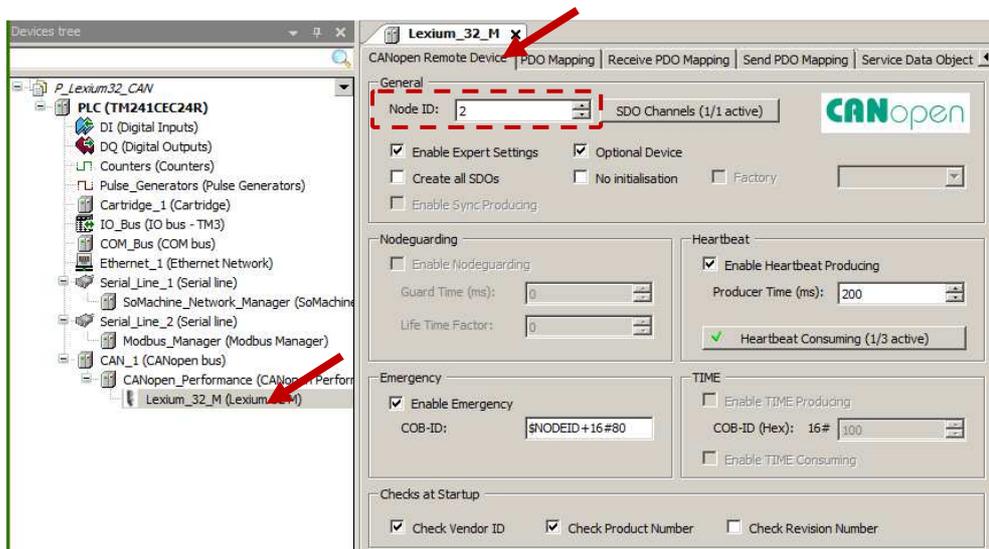
2. Ahora en el 'Logic Builder' dentro de la pestaña de 'Dispositivos' insertamos el maestro de CAN dentro del puerto de CAN del M241.



- Ahora añadimos un esclavo de Canopen, sobre el **'Canopen\_Performance'** añadido pulsamos el icono de **'+'** y buscamos el esclavo que queremos insertar, en este caso **'Lexium 32 M'**.

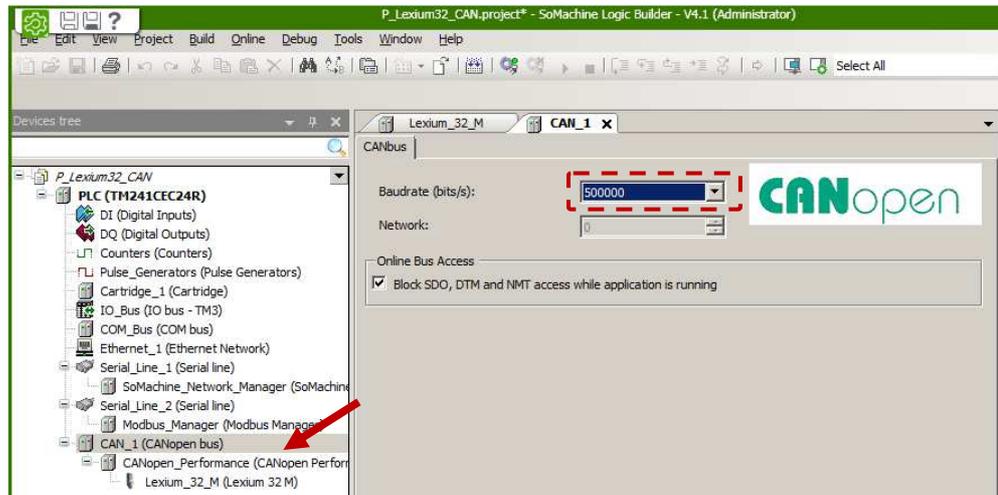


- Para configurar el Lexium 32M que se ha añadido a la red CAN se pulsa dos veces sobre él. Se le introducirá el número de nodo que ocupa en la red can, en este caso el **2**. Si se activa la opción de **"Enable Expert PDO Settings"** se podrán configurar más opciones, como heartbeat, PDOs...

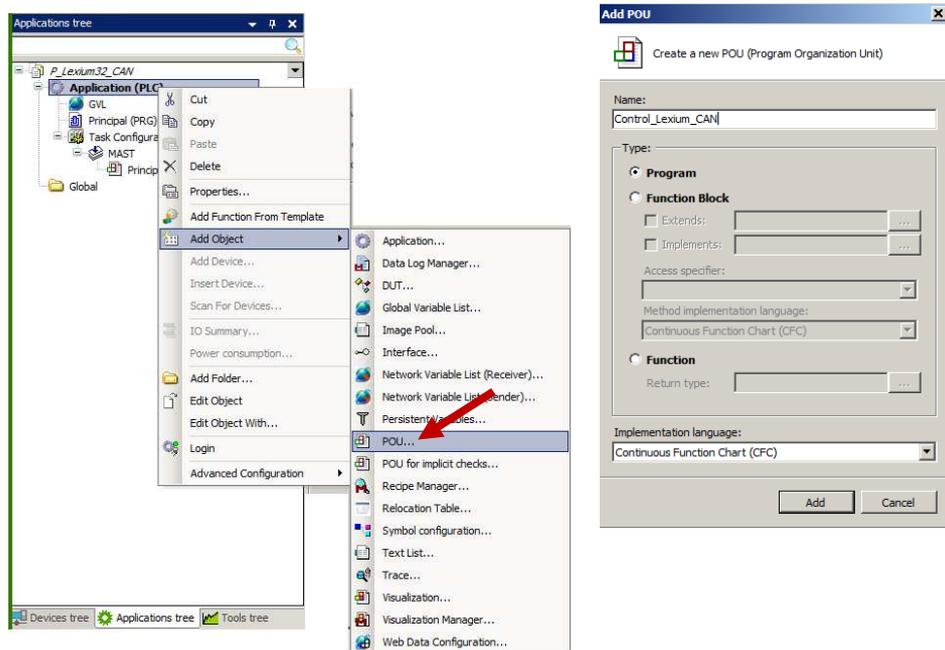


- En este caso solo ha activado la opción de heart beat, no hace falta tocar la configuración de los PDOs para que funcionen los bloques de PLCOpen que se usaran a continuación.

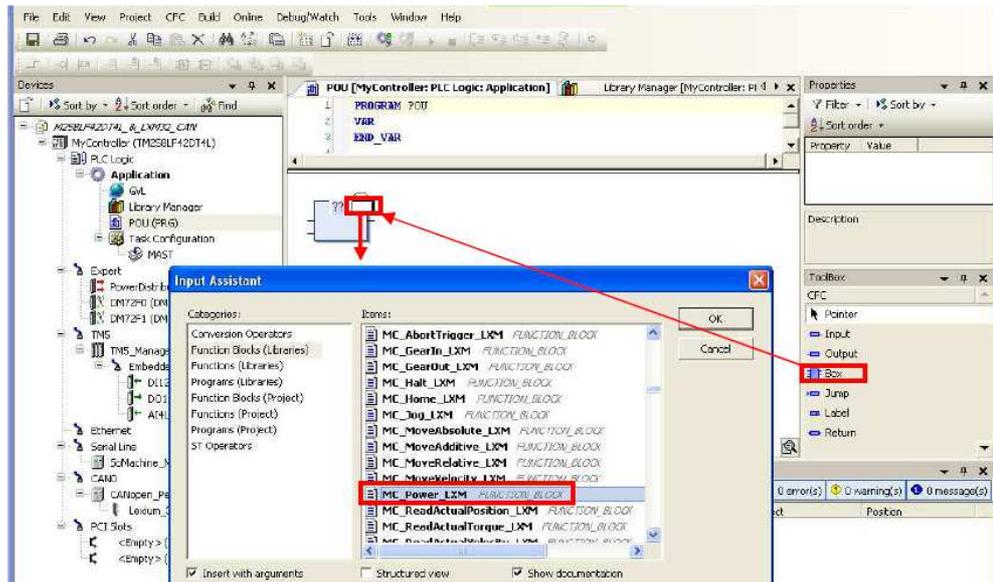
6. El baudrate de la red CAN se ajusta pulsando sobre el puerto de CAN\_1, y lo ponemos 500000 bit/s.



7. Se añade un POU al programa en la pestaña de aplicación, sobre 'Application \_ Botón derecho \_ Add Object \_ POU'.

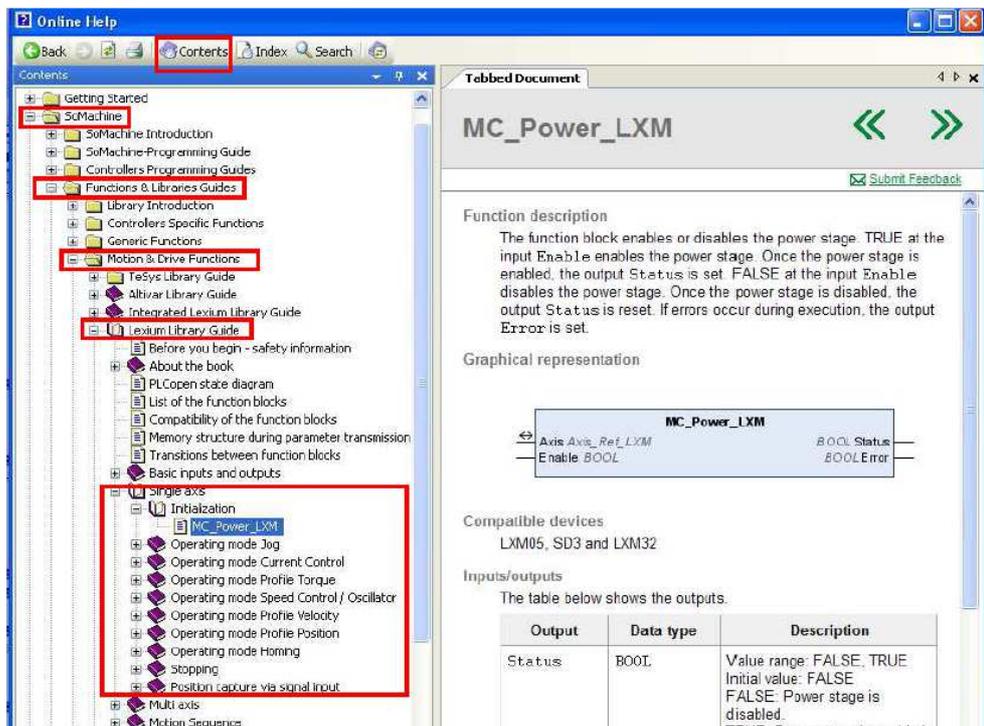


8. En el POU que se inserta una BOX y dentro del asistente, buscamos dentro de la librería '**SEM\_LXM / Funtion Bloks / Single Axis**' añado los bloques de función de movimiento que empiezan por **MC\_**xxx que son bloques de función PLCOpen.

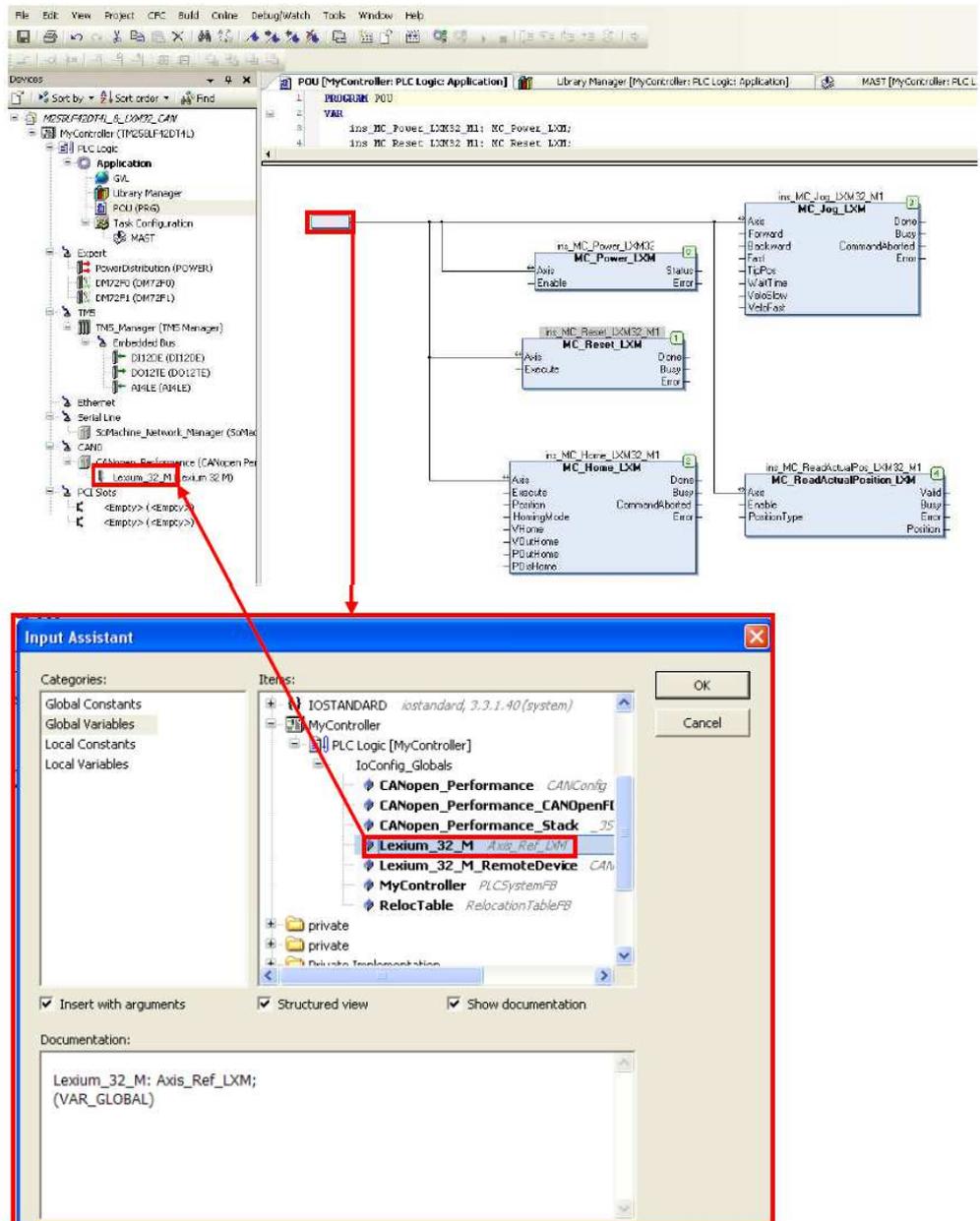


En este caso se han añadido al POU, los siguientes bloques de función: '**MC\_Power\_LXM**', '**MC\_Reset\_LXM**', '**MC\_Home\_LXM**', '**MC\_Jog\_LXM**' y '**MC\_ReadActualPosition\_LXM**'.

En el onlinehelp de SoMachine se puede encontrar información detallada sobre dichos bloques.



9. En el pin de entrada “Axis” de los bloques de función se inserta el nombre del eje, en este caso ‘Lexium\_32\_M’, que es una variable en forma de estructura de datos ‘Axis\_Ref\_LXM’.



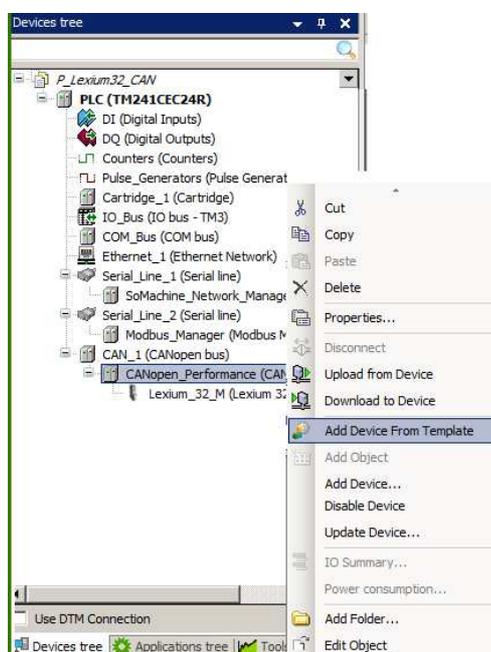
## Ejercicio – Inserción de Lexium32 en CanOpen con la plantilla de dispositivos.

Existe otra manera de insertar el control de un Lexium 32 desde “**template**”. Añadiendo dispositivo desde una plantilla, “**template**”. Se selecciona Lexium 32M, dirección del dispositivo dentro de la red CAN y el lugar dónde se insertará el bloque de función encargado de controlar al LXM32.

Previamente a la inserción de una plantilla de dispositivo, se tiene que haber creado un programa POU, e insertado un maestro CanOpen.

Para insertar una plantilla hay que seguir realizando los siguientes pasos.

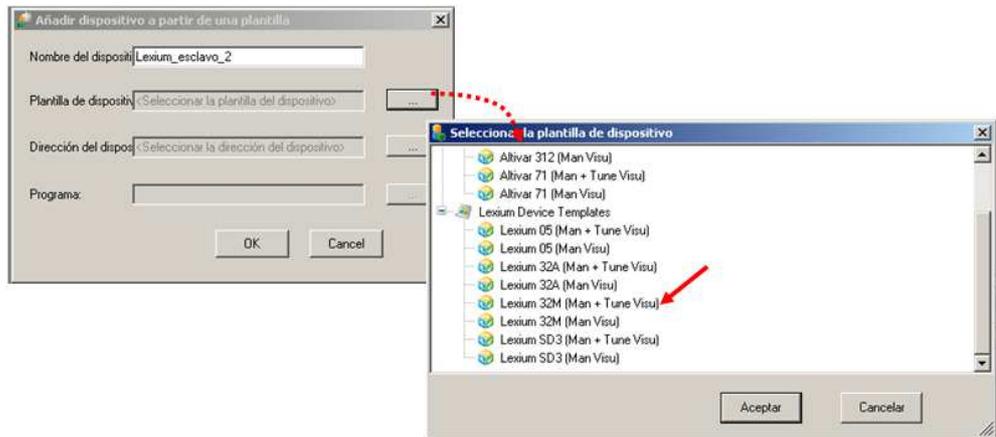
Seleccionar el maestro CanOpen y con el botón derecho de ratón elegir la opción ‘**Añadir dispositivo a partir de una plantilla**’.



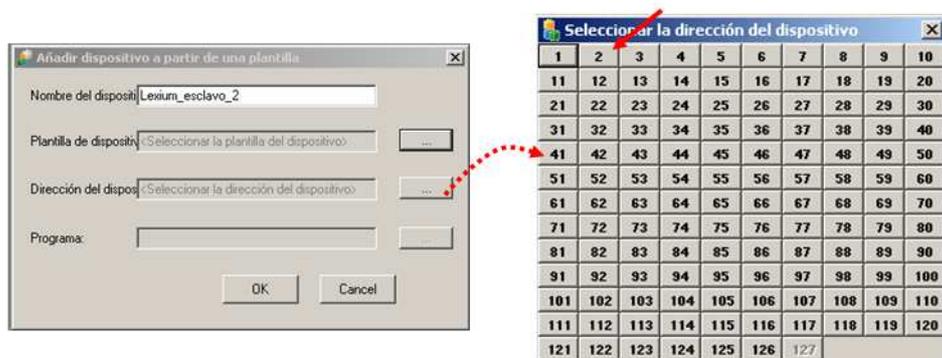
Aparecerá una ventana flotante donde introduciremos el nombre de esclavo que se quiere que tenga el Lexium32, en este caso *Lexium\_esclavo\_2*.



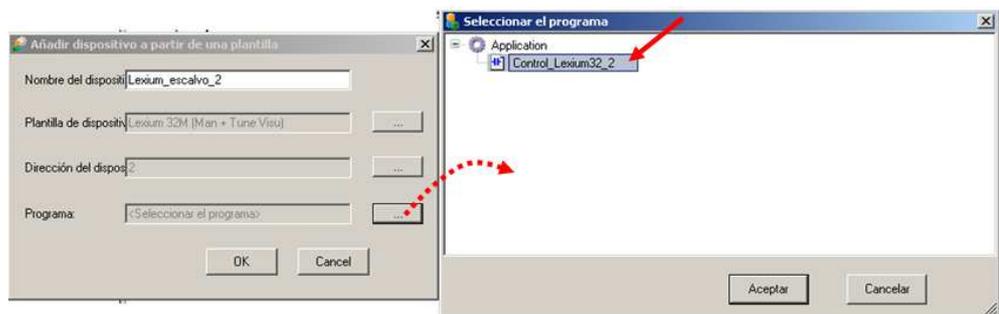
Ahora en el campo plantilla de dispositivo, se seleccionará entre una lista de plantillas, aquella que nos interesa, para acceder al listado apretar el botón . En este caso se seleccionará la plantilla de '**Lexium 32M (Man + Tune Visu)**', que nos generará unas pantallas de visualización para el movimiento manual del lexium y calibración de este.



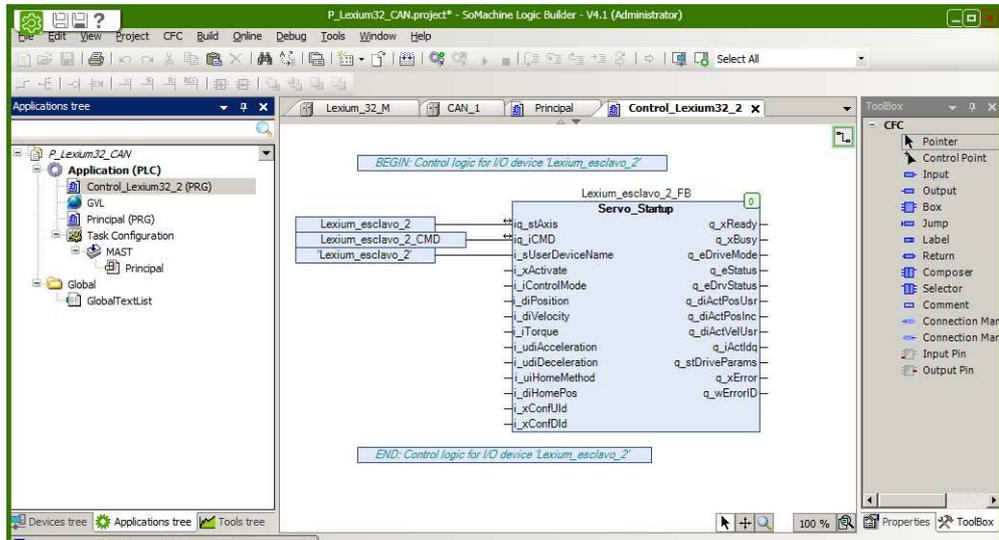
Después de seleccionar el perfil, en el campo '**Dirección de dispositivo**' se tiene que decir que número de esclavo de Canopen se va a configurar el Lexium 32M. (Las direcciones de esclavo ya utilizadas en el programa aparecerán deshabilitadas).



Finalmente en el campo 'Programa' hay que indicarle el POU donde se desea que se inserte el FB para controlar el equipo. En este caso el POU que se había creado previamente.

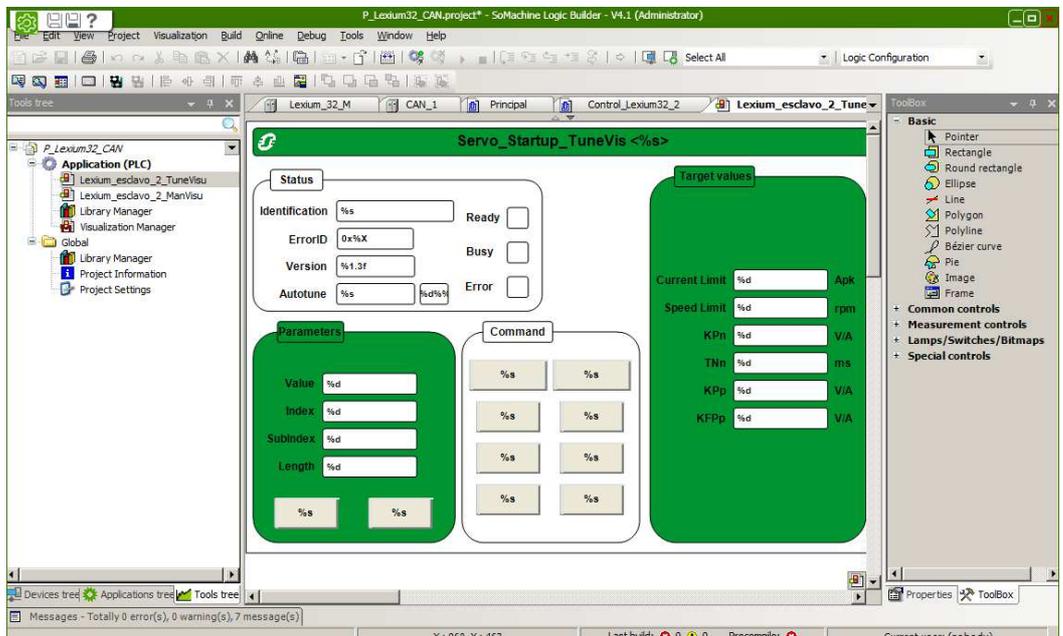


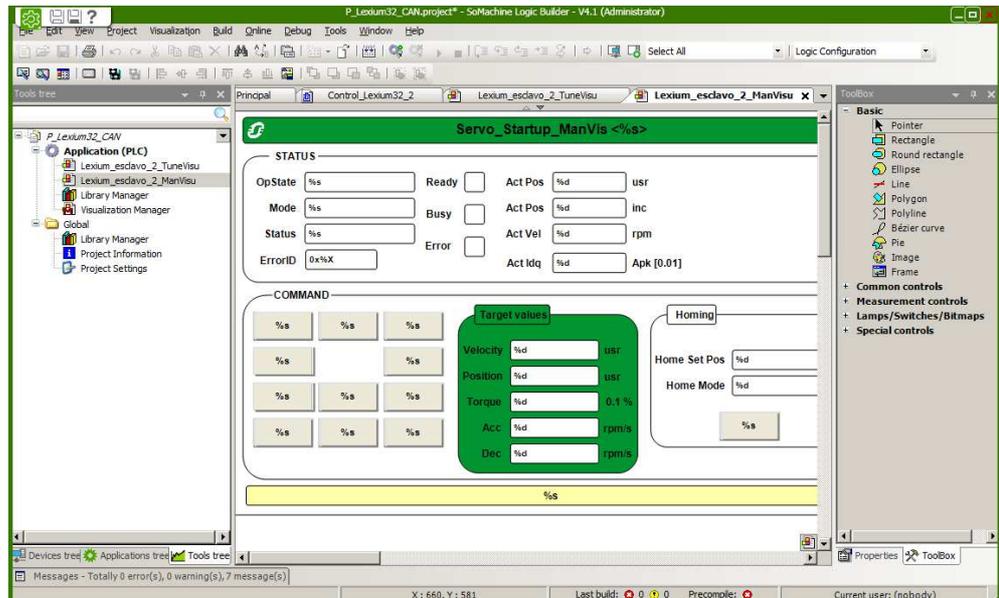
De modo automático se ha añadido un nuevo esclavo al proyecto y el bloque de función en el POU seleccionado.



**Nota:** La opción de control de engranaje no está incluida en este bloque de función, si se desea utilizar se tendrán que hacer llamadas a los bloques de PLC Open MC\_GearIn\_LXM y MC\_Gear\_Out\_LXM.

También se han creado las visualizaciones que cuando que en modo Online se puede utilizar para probar el equipo de modo manual, directamente sin hacer programa.





Aparecerá una ventana indicando si la importación se ha realizado con éxito.

## DTM's - FDT

### Introducción

FDT / DTM es un concepto independiente del fabricante, que nos permite la parametrización de los dispositivos de campo de los diferentes fabricantes que utilizan en una arquitectura de control.

El DTM (Device Type Manager) – es un archivo de software entregado por el fabricante del dispositivo, para acceder a los parámetros del dispositivo, proporcionando unas interfaces gráficas al usuario.

DTM se clasifican en dos categorías :

- DTM de equipos: Para realizar la configuración de dispositivos de campo.
- CommDTM – para conectar con los componentes de comunicación del software.

SoMachine dispone de un marco FDT para aquellos buses de campo más utilizados como son CanOpen y Modbus RTU.

La norma FDT se utiliza para definir las interfaces entre los componentes específicos del dispositivo de software (DTM) y el software de programación SoMachine.

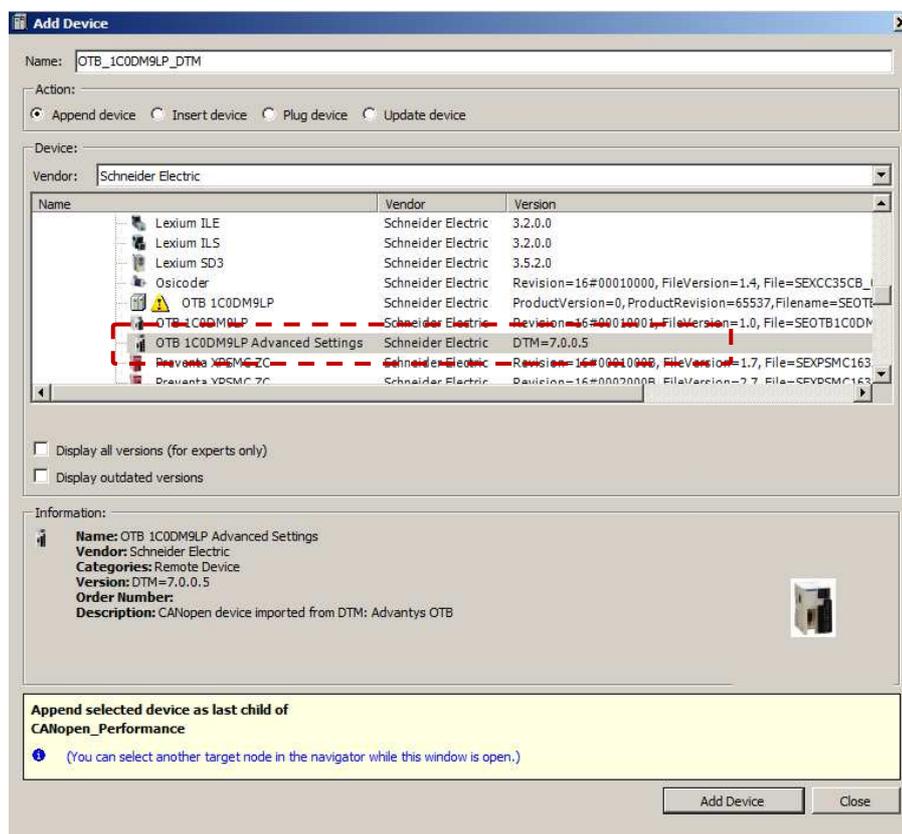
Equipos de Schneider Electric soportados por DTM en SoMachine:

DTM Name	Supported Function	CANopen Supported	Modbus SL Supported	Modbus TCP Supported
Advantys OTB DTM	Configuration of Advantys OTB islands	Yes	No	No
TM5-TM7 DTM	Configuration of TM5 and TM7 devices	Yes	No	No
ATS22 DTM	Configuration of Altistart 22	No	Yes	No
ATS48 DTM	Configuration of Altistart 48	No	Yes	No
ATV12 DTM	Configuration of Altivar 12	No	Yes	No
ATV31-312 DTM	Configuration of Altivar 31	No	Yes	No
	Configuration of Altivar 312	Yes	Yes	No
ATV32 DTM	Configuration of Altivar 32	Yes	Yes	Yes
ATV61 DTM	Configuration of Altivar 61	No	Yes	No
ATV71 DTM	Configuration of Altivar 71	Yes	Yes	Yes
Lexium 32 A DTM	Configuration of Lexium 32 A	Yes	Yes	No
Lexium 32 C DTM	Configuration of Lexium 32 C	No	Yes	No
Lexium 32 i DTM	Configuration of Lexium 32 i	Yes	Yes	No
Lexium 32 M - S DTM	Configuration of Lexium 32 M	Yes	Yes	Yes
	Configuration of Lexium 32 S	No	Yes	No
TeSysT DTM	Configuration of TeSys T	No	Yes	No
TeSysU DTM	Configuration of TeSys U	No	Yes	No
Harmony XB5R DTM	Configuration of ZBRN2 access point	No	Yes	No
	Configuration of ZBRN1 access point	No	No	Yes

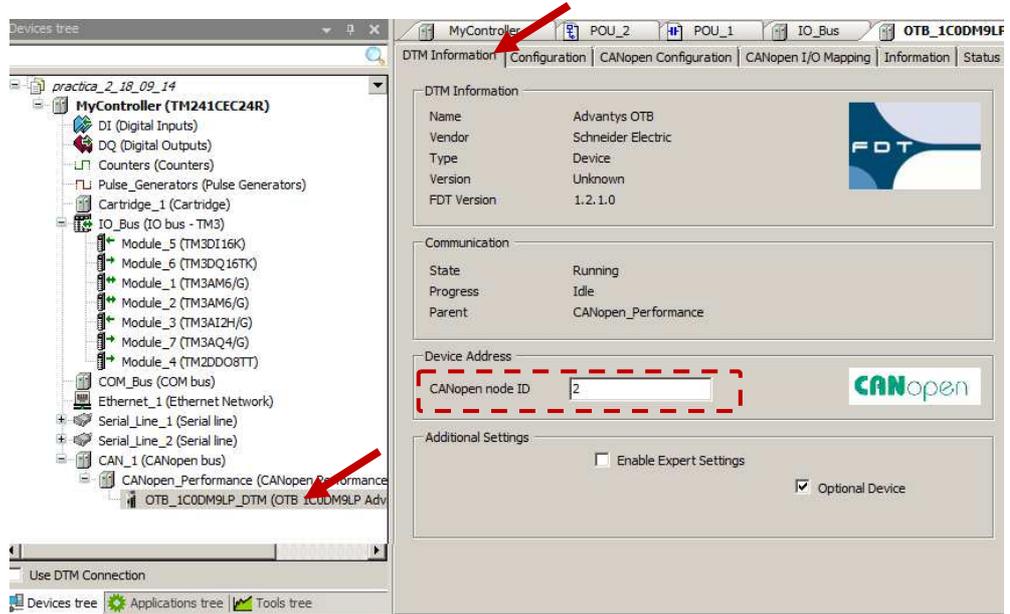
## Ejercicio – Inserción de Advantys OTB en bus CANOPEN con DTM

A continuación se detallan los pasos a seguir para configurar una isla advantys OTB que comunica con bus CanOpen, en SoMachine 4.1, con un M241.

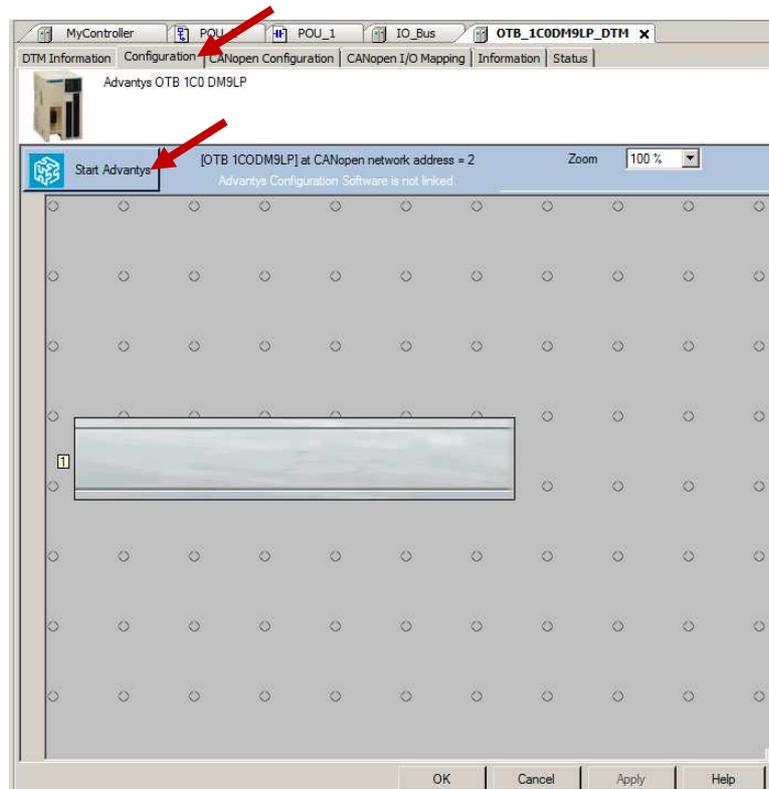
1. Desde la pantalla de inicio, utilizando el asistente de creación de proyecto, se inserta el TM241CEC24R.
2. Ahora en el **'Logic Builder'** dentro de la pestaña de **'Dispositivos'** insertamos el maestro de CAN **'Canopen\_Performance'** dentro del puerto de CAN del M241.
3. Ahora añadimos un esclavo de Canopen, sobre el **'Canopen\_Performance'** añadido pulsamos el icono de **'+'** y buscamos el esclavo que queremos insertar, en este caso **'OTB1C0DM9LP Advance Settings'**, al seleccionarlo cercionarse que en la columna de **'Versión'** nos indica la versión DTM.



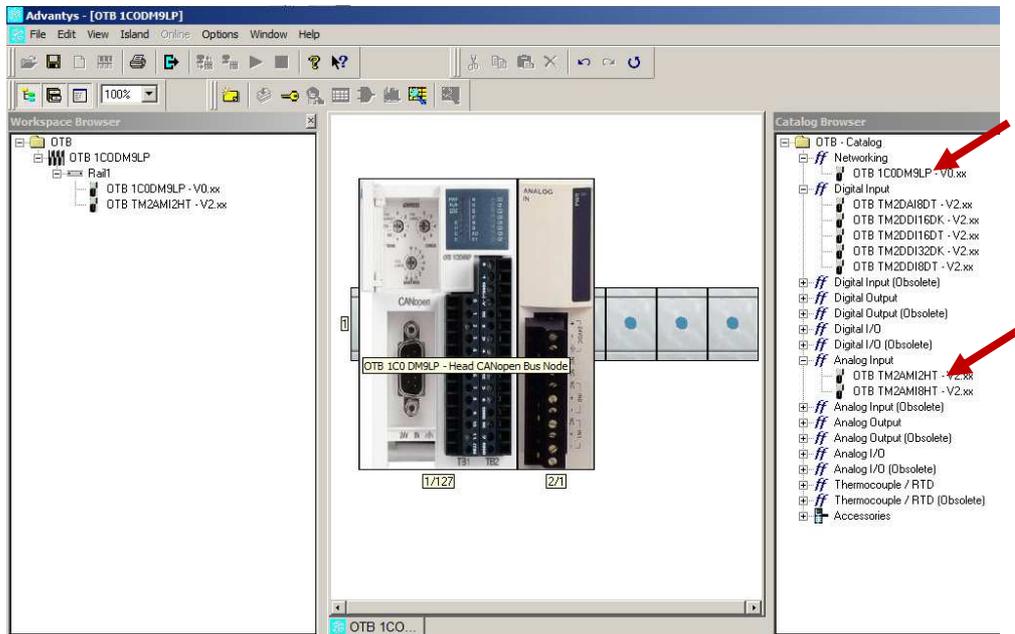
4. Ahora en la pestaña dispositivos del navegador, seleccionamos el esclavo OTB, en la ventana de configuración del esclavo, en la pestaña '**DTM Información**' en el campo '**CANopen node ID**' le diremos el número de esclavo de la isla OTB.



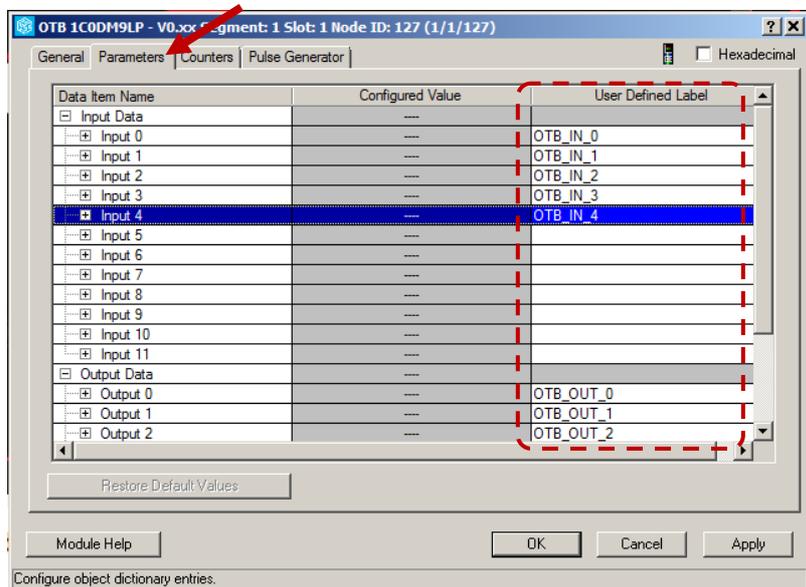
5. Ahora en la pestaña '**Configuration**', abrimos el FDT de la isla Advantys OTB y abrimos FDT, pulsando en el botón de '**Start Advantys**'.



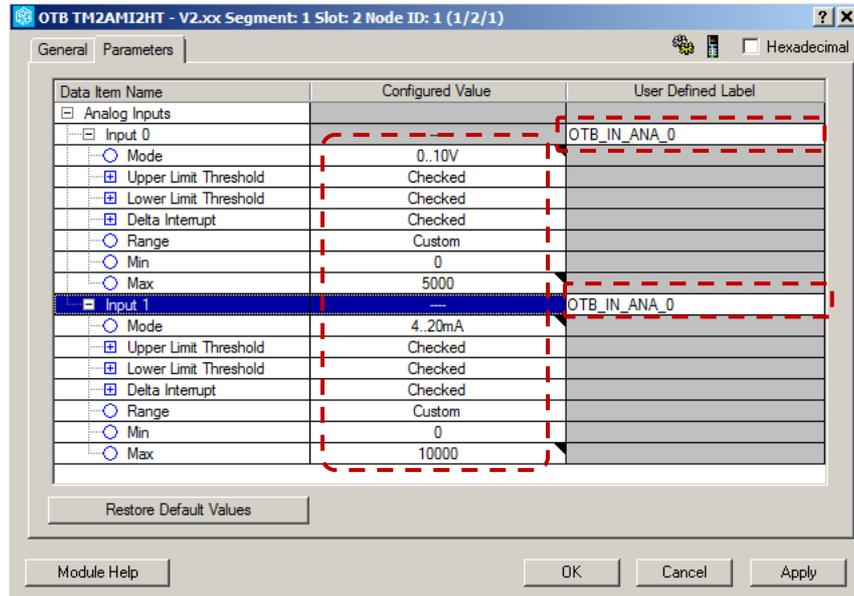
- Ahora realizamos la configuración de la isla Advantys. Añadimos la cabecera Canopen 'OTB 1C0DM9LP' + Tarjeta de 2 entradas analógicas de V/I 'OTB TM22AMI2HT'.



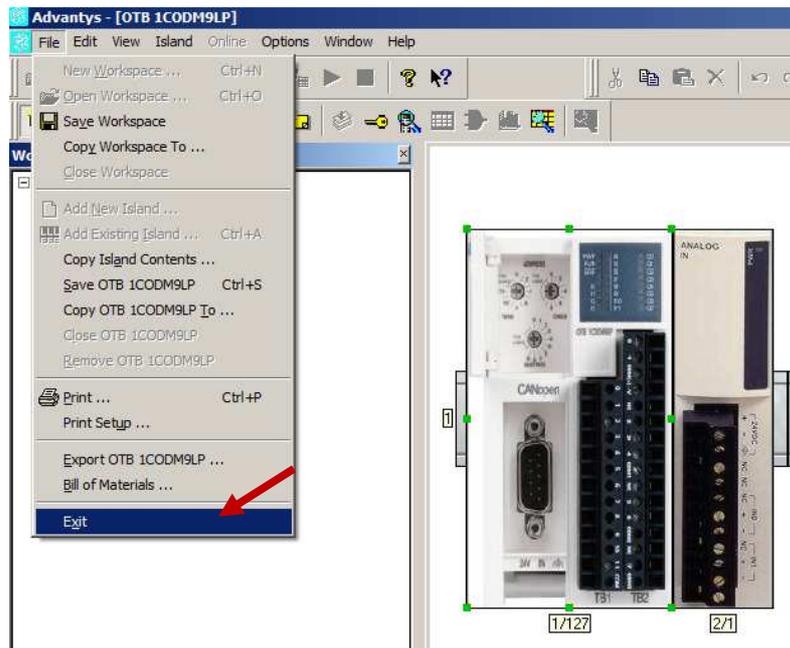
- Ahora añadimos los símbolos de las variables dentro del configurador de Advantys, hacemos doble clic sobre la cabecera de la OTB y escribimos los símbolos de las entradas y salidas digitales.



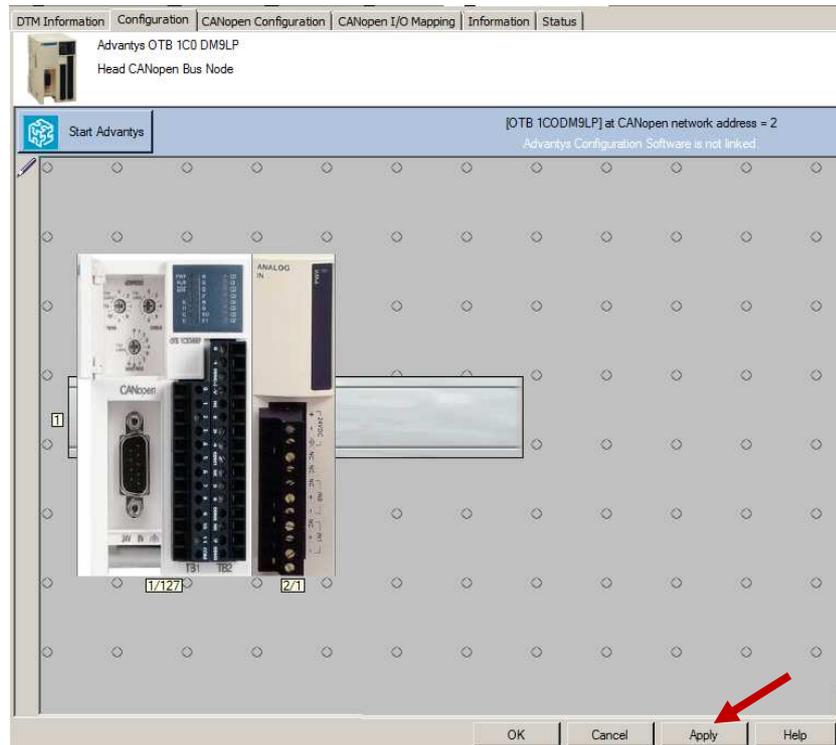
8. Hacemos lo mismo con la tarjeta de entradas analógicas.



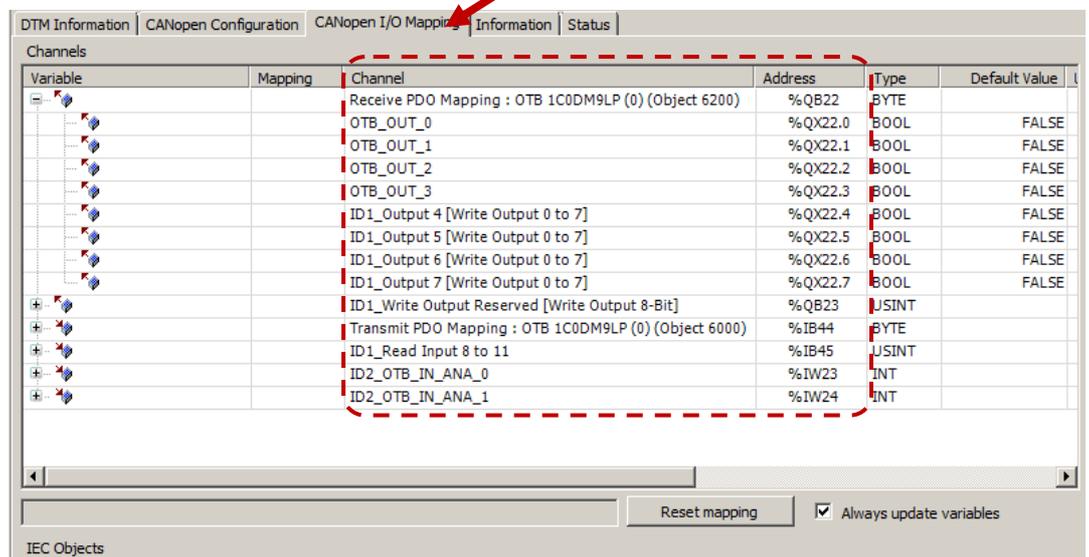
9. Una vez configurada la isla, le damos a salir.



10. Volvemos a la pestaña de configuración del SoMachine, y vemos que aparece la isla ya configurada y pulsamos el botón de aplicar.



11. En la pestaña 'CANopen I/O Mapping' veremos que los símbolos de las variables ya están creados y la configuración queda guardada también.





# Capítulo 14: Ethernet

## Descripción

**Introducción** Este capítulo trata de cómo realizar la configuración del puerto Ethernet y las diferentes posibilidades de comunicación que el software SoMachine ofrece bajo este bus de comunicación.

**Este CAPÍTULO trata los siguientes temas:**

- Introducción ..... 14-2
- Configuración del puerto Ethernet ..... 14-3
- Ejercicio – Mensajería Modbus TCP/IP.....14-4
- Ejercicio –Modbus TCP/IP – I/O Scanning.....14-9
- Global Network Variables ..... 14-13
- Servicios de Ethernet .....14-17
- Ejercicio – Webvisualization .....14-19

# Introducción

## Ethernet

El intercambio de variables en redes Ethernet, se realizarán con dos tipos de intercambio:

**Cíclico:** Configurando la lista de variables globales en el emisor y creando una lista de variables de red en el receptor entre dos equipos controladores; un M258 y una LMC058.

**Explícito:** Con las funciones *READ\_VAR* y *WRITE\_VAR* donde se leen y se escriben variables cuando el cliente realiza la petición sobre el servidor.

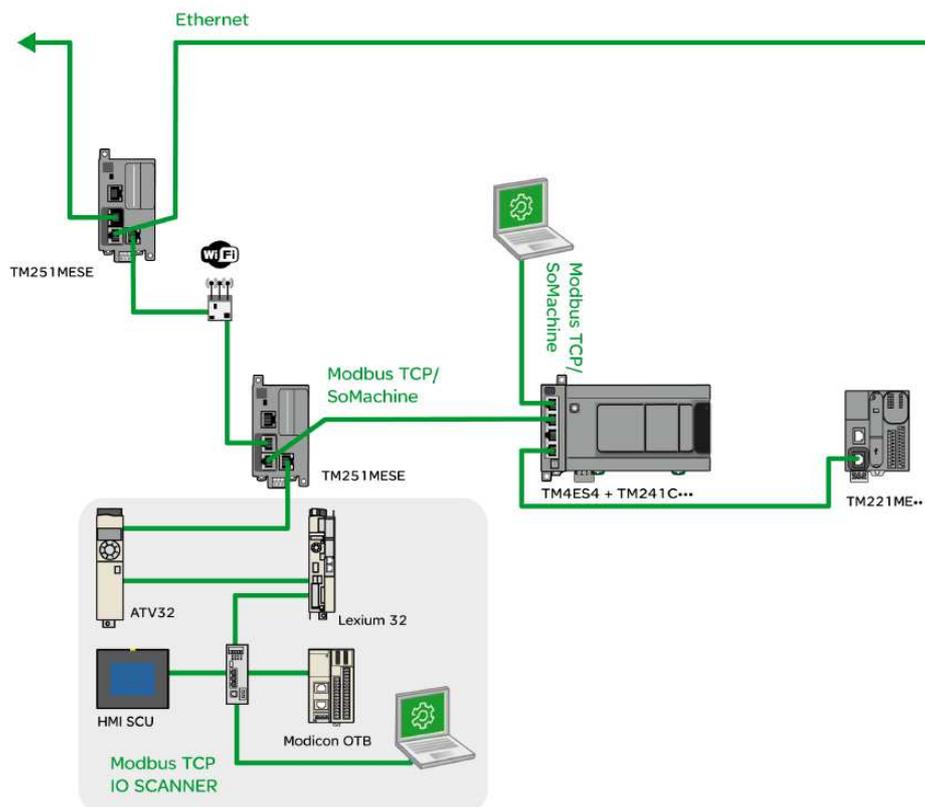
**Modbus TCP I/O Scanner:** es un servicio Ethernet que sondea continuamente módulos de E / S (esclavos) para recoger datos, estado e información de diagnóstico. Este proceso supervisa las entradas y controla las salidas.

El servicio de Modbus TCP I/O Scanner se basa en un modelo **maestro/esclavo**. El maestro es el único Controlador **TM251MESE**. El maestro y los esclavos deben estar en la misma **red IP**. El Modbus TCP I/O Scanner nos permite comunicar con un máximo de **64 esclavos**, y hasta **64 canales** (cada esclavo puede tener más de **1 canal**).



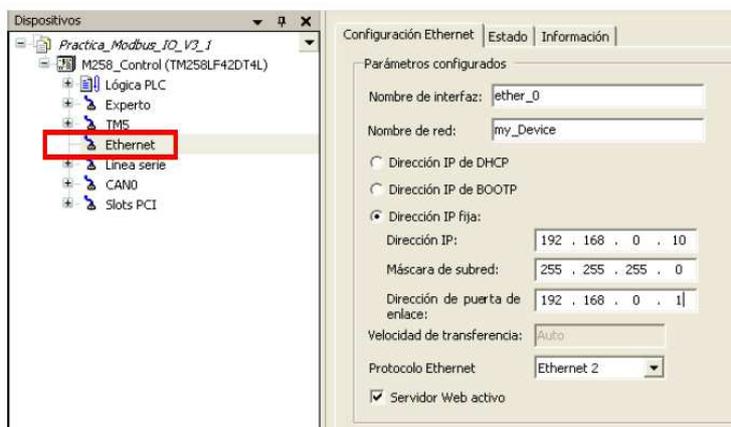
**Nota:** El Modbus TCP I/O Scanner sólo está disponible en el puerto Ethernet 2 del controlador lógico TM251MESE.

Además los controladores que incluyen Ethernet (M241, M251, M258 y LMC058), también disponen de servicios de Ethernet como el FTP y el WebServer.



## Configuración del puerto Ethernet

Para configurar una red Ethernet, hay que hacer doble clic sobre el puerto de Ethernet del controlador y aparece la ventana de configuración del puerto Ethernet.



**Nombre de interfaz:** Nombre de la conexión de red

**Nombre de red:** Se utiliza como nombre de dispositivo para recuperar la dirección IP mediante DHCP, 16 caracteres como máximo.

Existen tres maneras distintas de asignar la dirección IP del controlador:

- Servidor DHCP,
- Servidor BOOTP,
- Dirección IP fija, el usuario define.
  - Dirección IP.
  - Máscara de subred.
  - Dirección de pasarela.. Si no hay ninguna pasarela, la dirección de la puerta de enlace es 0.0.0.0.



**Nota:** Si el método de direccionamiento probado no da resultado, el controlador comenzará a utilizar una dirección IP predeterminada derivada de la dirección MAC.

La dirección IP predeterminada se basa en la dirección MAC del dispositivo. Los primeros dos bytes son **10** y **10**. Los últimos dos bytes son los últimos dos bytes de la dirección MAC del dispositivo, convertidos al formato decimal. La **máscara de subred** predeterminada es **255.0.0.0**.

**Ejemplo:** Si la dirección MAC es **00.80.F4.01.80.F2**, la dirección IP predeterminada es **10.10.128.242**

**Velocidad de transmisión:** La velocidad de transferencia y la dirección en el bus se configuran automáticamente.

**Protocolo Ethernet:** Tipo de protocolo utilizado (Ethernet2 o IEEE 802.3)

**Servidor web activo:** Habilitar/deshabilitar servidor Web

## Ejercicio – Mensajería en Modbus TCP/IP

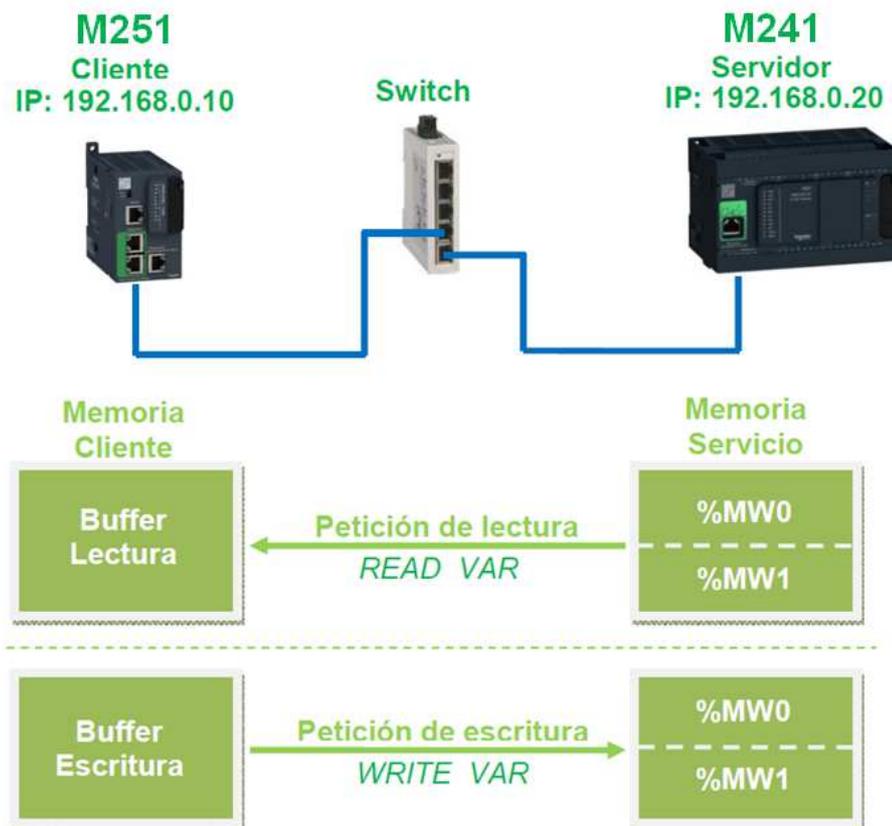
### Mensajería

La transferencia de información entre un cliente y un servidor Modbus TCP/IP **se inicia cuando el cliente envía una petición al servidor para transferir información, ejecutar un comando o efectuar una de las muchas funciones posibles.**

Después de que el servidor reciba la petición, ejecuta el comando o recupera los datos requeridos de la memoria correspondiente. A continuación, el servidor responde al cliente haciéndole saber que el comando se ha completado o proporcionándole los datos solicitados.

Tanto el M241 (cuando dispone de Ethernet), M251, M258 como la LMC058 implementan servicios de cliente de servidor, que pueden iniciar comunicaciones con otros controladores y dispositivos de E/S, así como responder a las peticiones de otros controladores, SCADA, HMI y demás dispositivos.

En este ejemplo se ha utilizado el M251 como Cliente Modbus TCP y el M241 como servidor.



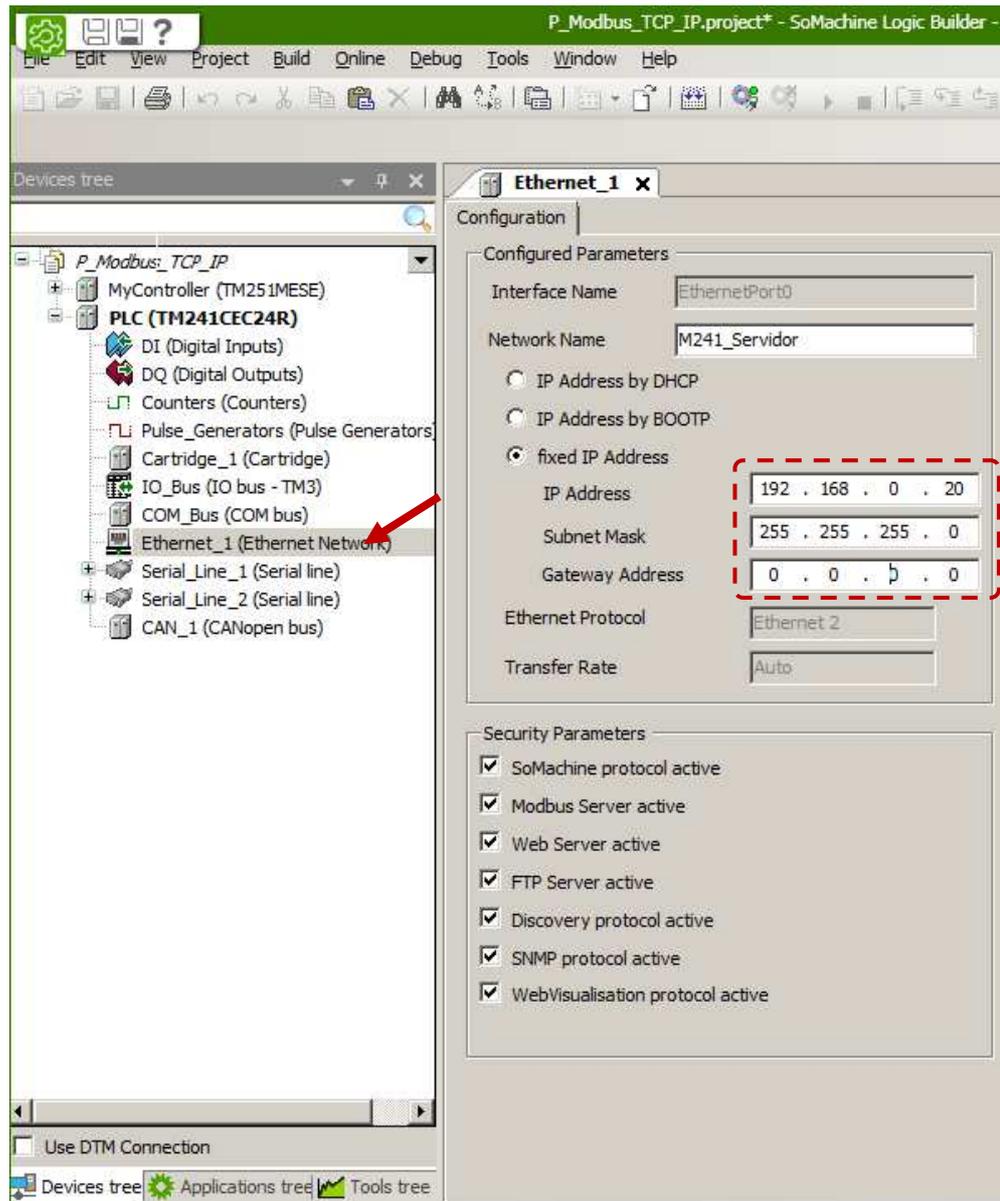
Las peticiones las realiza el M251 y la M241 lo responderá.

## Proyecto servidor

Sin ninguna configuración, el puerto Ethernet incorporado del controlador admite el servidor Modbus.

El servidor/cliente Modbus se incluye en el firmware y no requiere ninguna acción de programación por parte del usuario. Debido a esta característica es accesible en los estados de ejecución, detención y vacío.

Lo único que hay que hacer es configurar la IP en el proyecto del Servidor (M241).



En el proyecto del cliente del Modbus TCP/IP es donde se realizarán las diferentes peticiones. Para ello se disponen de los siguientes bloques de funciones:

**ADDM:** convierte una dirección de destino representado en un STRING en una estructura de ADDRESS que se puede utilizar como una entrada en un bloque de funciones de comunicación.



**Nota:** En el pin 'Addr' del bloque de funciones ADDM, hay que introducir la dirección de destino (en este caso la IP del M241) en modo STRING. El formato de una dirección de Ethernet será '**<número de puerto de comunicaciones> { <dirección IP del destino A.B.C.D> }**', donde el número de puerto Ethernet de los controladores es **3**. (Ejemplo: '**3{192.168.0.20}**').

**READ\_VAR:** lee datos de un dispositivo externo en el protocolo Modbus.

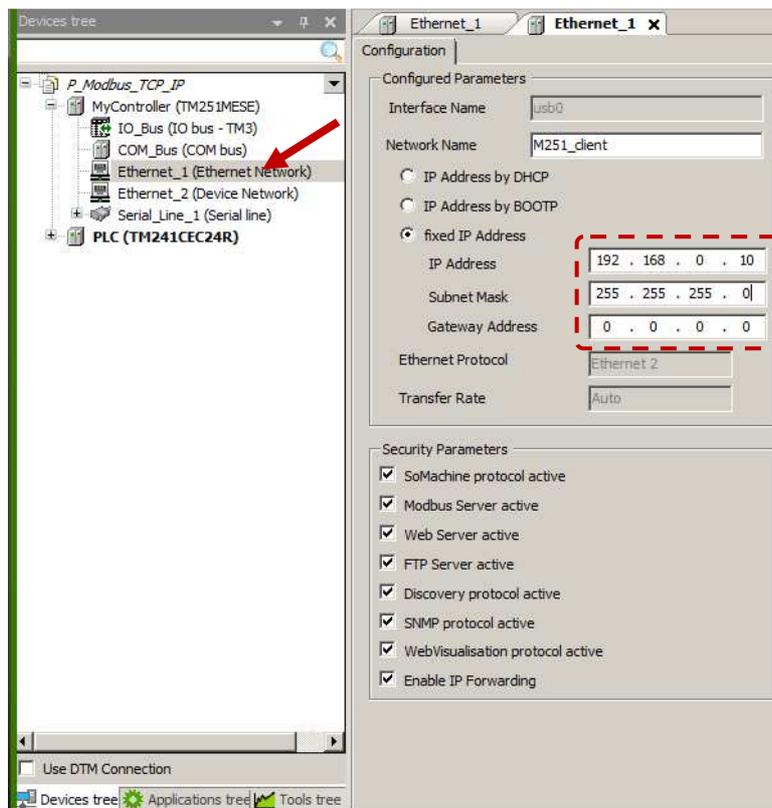
**SEND\_RECV\_MSG:** envía y recibe mensajes definidos por el usuario.

**SINGLE\_WRITE:** un sólo registro en un dispositivo Modbus externo.

**WRITE\_READ\_VAR:** lee y escribe registros internos (sólo el tipo MW) en un dispositivo externo en el protocolo Modbus.

**WRITE\_VAR:** escribe objetos en un dispositivo externo en el protocolo Modbus.

Se configura la IP del cliente de la misma manera que la IP del servidor.



En el caso del controlador **M251** que dispone de 2 redes de Ethernet, configuraremos el puerto **Ethernet 1**, el puerto **Ethernet 2** se utiliza para realizar la comunicación vía **I/O Scanning** con dispositivos (Variadores, I/O remotas ...etc).

Ahora en un POU creado, se realiza la siguiente programación:

```

1 PROGRAM POU
2 VAR
3   FB_ADDM: ADDM;
4   FB_READ_VAR: READ_VAR;
5   FB_WRITE_VAR: WRITE_VAR;
6   DataRead: ARRAY [0..1] OF WORD;
7   DataWrite: ARRAY [0..1] OF WORD;
8   Direccion: ADDRESS;
9   lectura: BOOL;
10  escritura: BOOL;
11  addm_exe: BOOL;
12 END_VAR

```

Se utilizan los bloques de función READ\_VAR y WRITE\_VAR acompañados de ADDM, este último encargado de gestionar la dirección del dispositivo sobre el que se leerán y escribirán las variables.

Dirección IP del M241

3[192.168.0.20]

Se leerán/escribirán dos MW a partir de la dirección 0, o sea, la MW0 y la MW1

lectura

10

Object type.MW

0

2

ADR(DataRead)

escritura

10

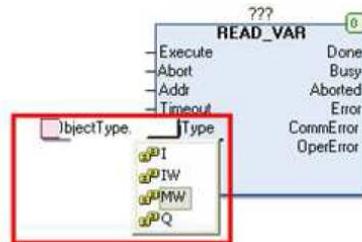
Object type.MW

0

2

ADR(DataWrite)

A la hora de parametrizar tanto el bloque de funciones **READ\_VAR** ó **WRITE\_VAR**, el pin 'ObjectType' es el tipo de objeto a leer o escribir (%I, %IW, %Q, %MW) para llamar al tipo de objeto hay que escribir en el pin 'objecttype.' al escribir el punto aparecerán los diferentes objetos de la estructura de datos.

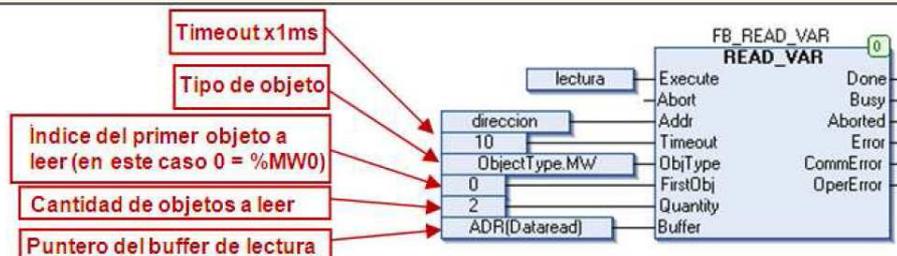


En el pin buffer se tiene que realizar una llamada al puntero, por lo que previamente se tiene que crear la variable 'Dataread' que será una matriz de **WORD** de la dimensión de los datos que se pretenden leer. Una vez creada la variable, en el pin se escribir ' **ADR(Dataread)** ' para crear un puntero de la matriz **POINTER TO BYTE**.

```

1 PROGRAM Modbus_TCP_IP
2 VAR
3   FB_READ_VAR: READ_VAR;
4   lectura: BOOL;
5   direccion: ADDRESS;
6   Dataread: ARRAY [0..1] OF WORD; //La dimensión de la matriz es igual al Quantity
7 END_VAR

```



## Ejercicio - Modbus TCP/IP - I/O Scanning

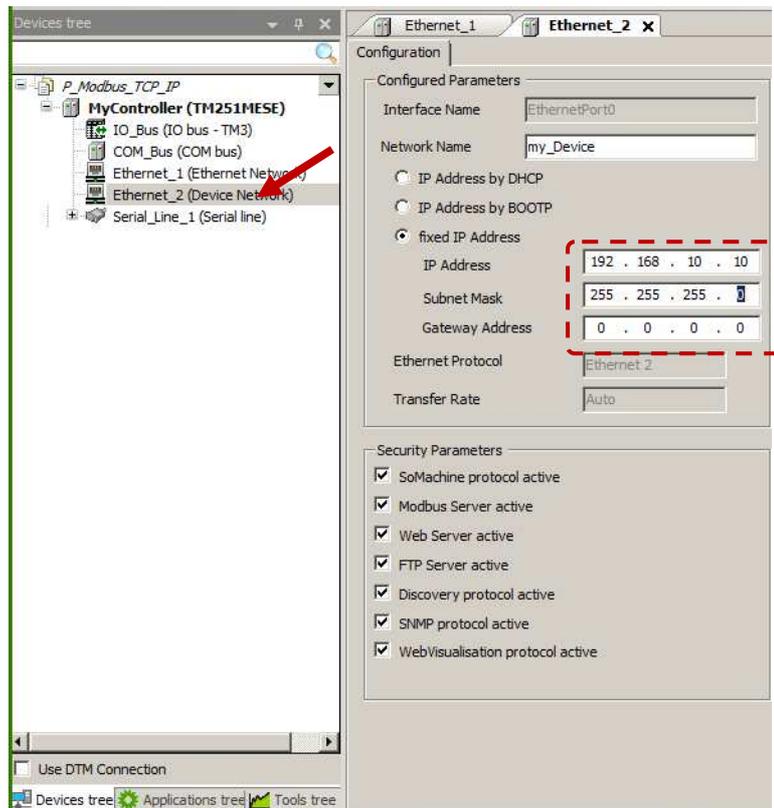
### 1 Crear una comunicación ethernet I/O Scanning con el puerto Ethernet 2 del controlador M251.

Vamos a realizar la comunicación I/O Scanning entre un controlador M251 y una isla de periferia descentralizada OTB.

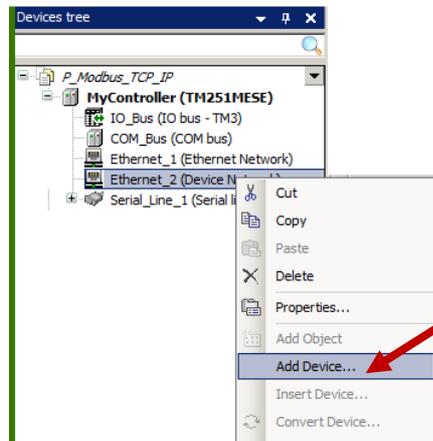


La isla advantys OTB, tiene integradas **12 entradas y 8 salidas** digitales, las cuales están direccionadas en los registros **0000 (entradas) y 0100 (salidas)**.

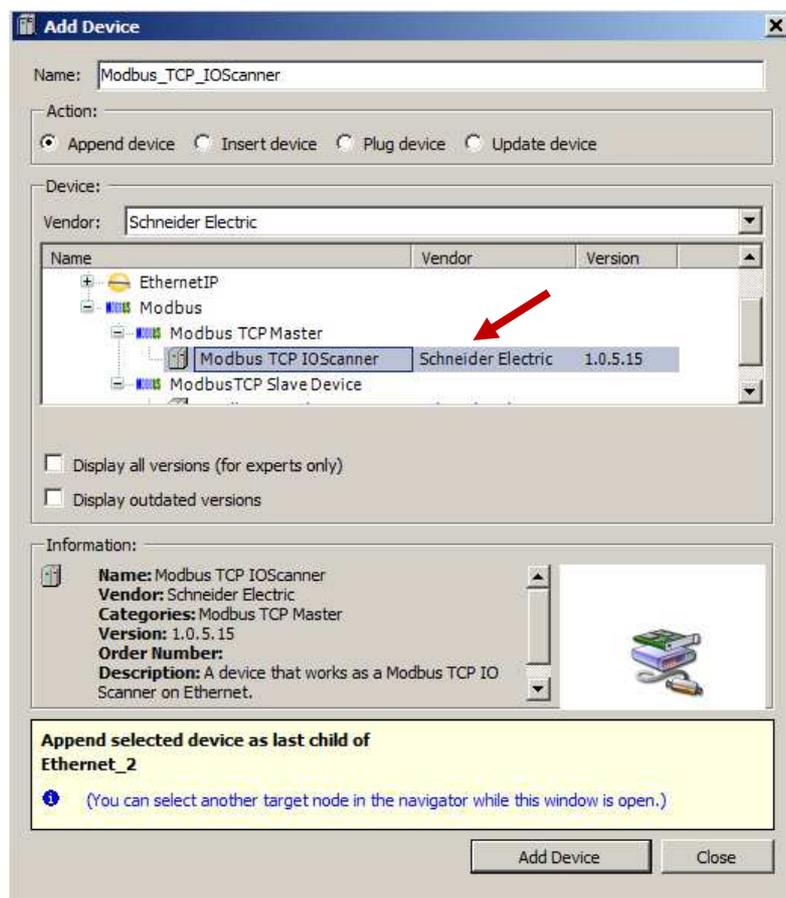
Configurar la IP del puerto **Ethernet 2** del M251, seleccionaremos la IP fija (*IP: 192.168.10.10*).



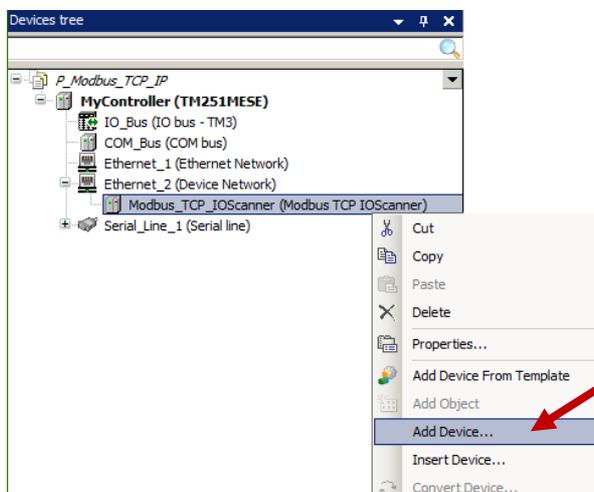
Seleccionar el puerto **'Ethernet 2'**, en la pestaña dispositivos del navegador y pulsar botón derecho del ratón sobre él. En el menú de opciones que aparece seleccionar **'Añadir equipo'**.



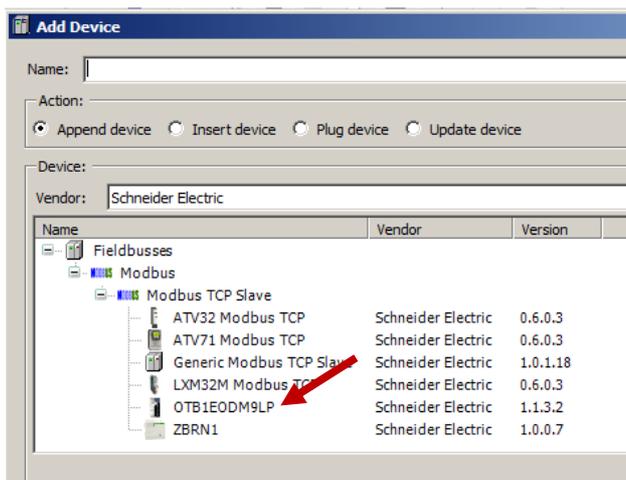
Añadir el elemento **'Modbus TCP IO Scanner'**, que está dentro de **'Modbus/ Modbus TCP Master'** de la lista de equipos.



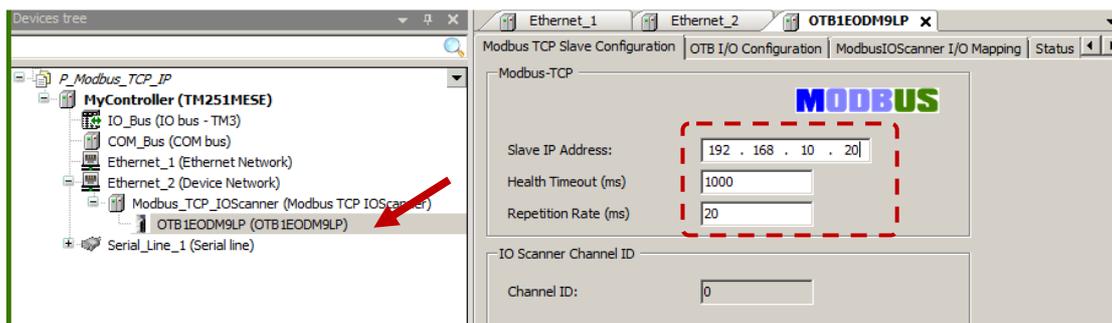
En la **Ethernet 2**, donde se ha añadido el '**Modbus TCP IO Scanner**', seleccionarlo y pulsar botón derecho y seleccionar '**Añadir Equipo**' para añadir el esclavo OTB.



Añadir como esclavo modbus TCP/IP '**OTB1EODM9LP**' al maestro (en el caso que el dispositivo no se encuentre en la lista se elegirá '**Generic Modbus TCP Slave**').



Hacer doble clic sobre el esclavo '**OTB1EODM9LP**' que aparece asociado al maestro de I/O Scanning del puerto Ethernet 2 en el navegador, y configurar la IP de la OTB.



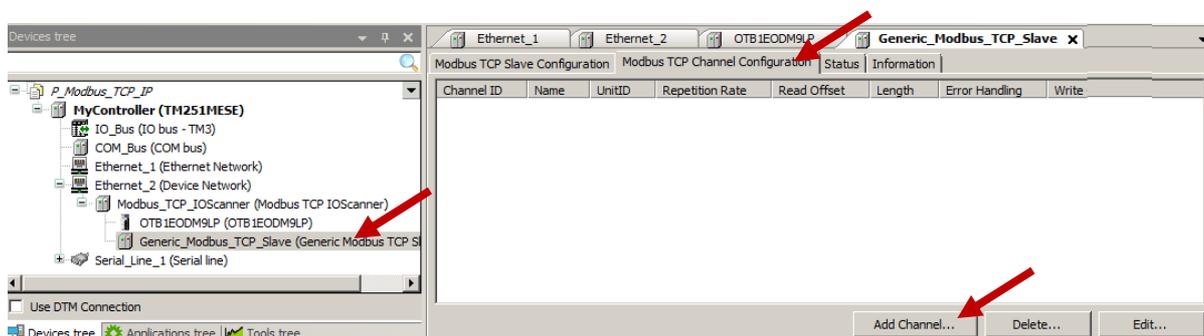
Ahora en el esclavo OTB, seleccionar la pestaña '**ModbusIOScanner I/O Mapping**' donde crearemos el mapeado de las variables del esclavo que vamos a utilizar.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
<b>Inputs</b>							
iwOTB1EODM9LP...		Read Inputs	%IW5	WORD			
OTB_IN_0		Bit 0	%DX1...	BOOL	FALSE		
OTB_IN_1		Bit 1	%DX1...	BOOL	FALSE		
OTB_IN_2		Bit 2	%DX1...	BOOL	FALSE		
OTB_IN_3		Bit 3	%DX1...	BOOL	FALSE		
		Bit 4	%DX1...	BOOL	FALSE		
		Bit 5	%DX1...	BOOL	FALSE		
		Bit 6	%DX1...	BOOL	FALSE		
		Bit 7	%DX1...	BOOL	FALSE		
		Bit 8	%DX1...	BOOL	FALSE		
		Bit 9	%DX1...	BOOL	FALSE		
		Bit 10	%DX1...	BOOL	FALSE		
		Bit 11	%DX1...	BOOL	FALSE		
<b>Outputs</b>							
		Output co...	%QW0	WORD			
OTB_OUT_0		Bit 0	%QX0.0	BOOL	FALSE		
OTB_OUT_1		Bit 1	%QX0.1	BOOL	FALSE		
OTB_OUT_2		Bit 2	%QX0.2	BOOL	FALSE		
		Bit 3	%QX0.3	BOOL	FALSE		
		Bit 4	%QX0.4	BOOL	FALSE		
		Bit 5	%QX0.5	BOOL	FALSE		
		Bit 6	%QX0.6	BOOL	FALSE		
		Bit 7	%QX0.7	BOOL	FALSE		

En este caso al ser un equipo que se encontraba en la lista de esclavos Modbus TCP/IP, las señales ya aparecen preconfiguradas.

Configuración de los canales de lectura y escritura con un esclavo genérico:

En el caso que hubiéramos seleccionado un esclavo Modbus TCP/IP genérico, tendríamos que configurar las llamadas a los registros modbus de los esclavos que queremos leer y escribir. Tendríamos que ir a la pestaña '**Modbus TCP Channel Configuration**' e ir añadiendo canales '**Add Channel...**'.



En el canal en el campo 'Tipo de acceso' seleccionar **Read/Write Multiple Registers**, modificar el tiempo de ciclo **1000** ms. Rellenar los campo de 'Desplazamiento' con la dirección que se quiere leer y escribir y la longitud en words (en el caso del OTB el registro 0000, nos dice las entradas digitales y en el registro 100 realiza la activación de las salidas digitales).

The image shows a 'Modbus Channel' configuration dialog box. It is divided into three main sections: 'Channel', 'READ Register', and 'WRITE Register'.  
- The 'Channel' section includes: Name (Channel\_OTB), Unit-ID [0..255] (255), Repetition Rate (20 ms), Comment (empty), and Function Code (Read/Write Multiple Registers (Function code 23)).  
- The 'READ Register' section includes: Offset (0), Length (1), and Error Handling (KeepLastValue).  
- The 'WRITE Register' section includes: Offset (100) and Length (1).  
At the bottom right, there are 'OK' and 'Cancel' buttons.

Cuando se ha creado el canal de I/O scanning hay que crear las variables asociadas a ese canal para utilizarlas en el programa para esta comunicación Modbus. En este caso (del bit 0 al bit11 de la palabra 0 que son las entradas digitales de la OTB y del bit 0 al bit 7de la palabra 100, que son las salidas digitales de la OTB), seleccionar la pestaña '**Maestro Modbus Asignación E/S**' y declarar las variables, como en el caso anterior.

## Global Network Variables

### Descripción

La característica de lista de variables de red (**NVL**) consiste en una lista fija de variables que se pueden enviar o recibir a través de una red Ethernet.

La lista se debe definir en los controladores de emisor y de receptor (y puede gestionarse en un único proyecto o en varios). El valor de las variables se transmite mediante difusión (**UDP**). El UDP es un protocolo de comunicaciones de Internet que facilita la transmisión directa de datos en redes de protocolo de Internet (**Internet Protocol, IP**). Los mensajes de UDP/IP no necesitan una respuesta y, por lo tanto, son perfectos para aplicaciones en las que los paquetes descartados no requieren retransmisión (como redes y vídeos de transmisión por secuencias que necesitan rendimiento en tiempo real).

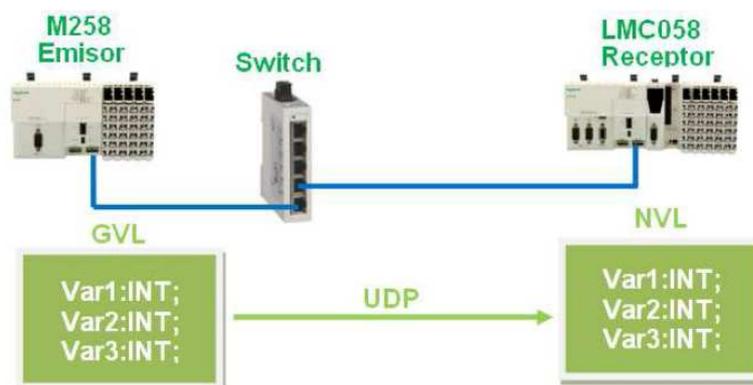
### Configuración de NVL

Las variables de red que se intercambiarán se definen en los 2 tipos de listas siguientes:

- Listas de variables globales (GVL) en un controlador de transmisión (emisor).
- Listas de variables globales de red (NVL) en un controlador de recepción (receptor).

Las variables de red se transmiten de la GVL (emisor) a una o más NVL (receptores). Para cada controlador, puede definir GVL, así como GNVL.

Por tanto, cada uno de los controladores puede actuar tanto de emisor como de receptor.



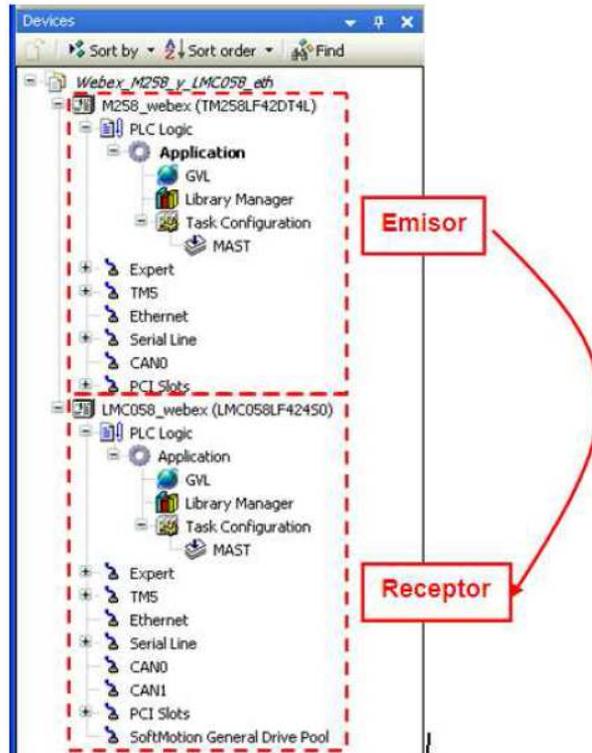
El mismo u otro proyecto pueden proporcionar una GVL de emisor. Así que, al crear una GNVL, puede seleccionarse la GVL de emisor en una lista de selección en la que aparezcan todas las GVL disponibles dentro de la red, o puede leerse de un archivo de exportación generado previamente (por ejemplo, utilizando el cuadro de diálogo Vínculo con archivo) en la GVL.



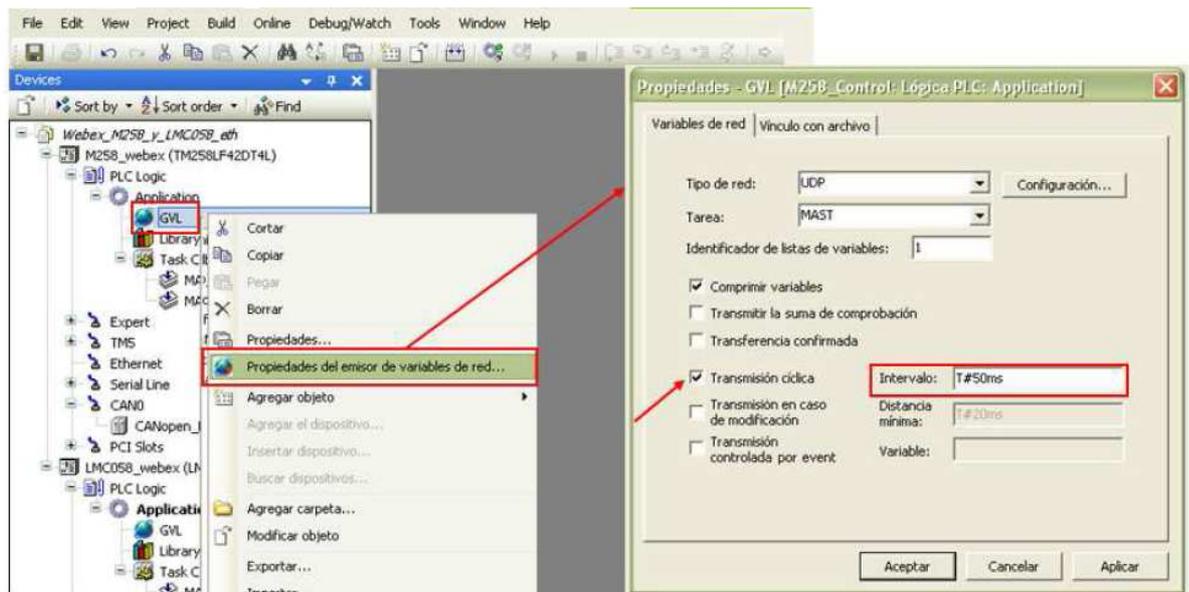
**Nota:** Si la GVL de emisor que se utilizará se define en otro proyecto, se necesitará un archivo de exportación.

## Configuración de emisor

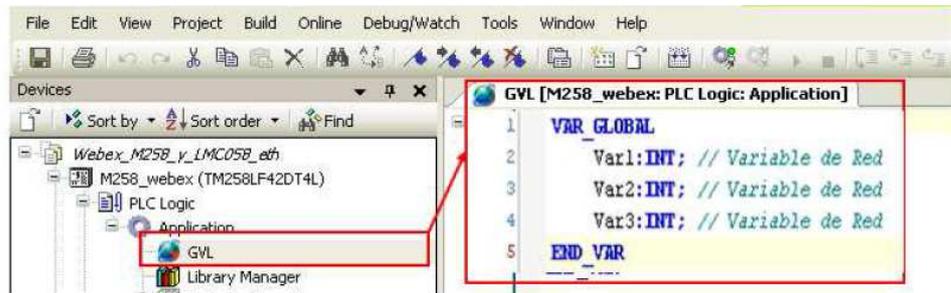
La transferencia de información entre un cliente y un servidor Modbus se inicia cuando el cliente envía una petición al servidor para transferir información, ejecutar un comando o efectuar una de las muchas funciones posibles.



Para compartir de la lista de variables globales, GVL, a través de la red de Ethernet, se debe de modificar las **'Propiedades del emisor de variables de red...'** de la GVL.

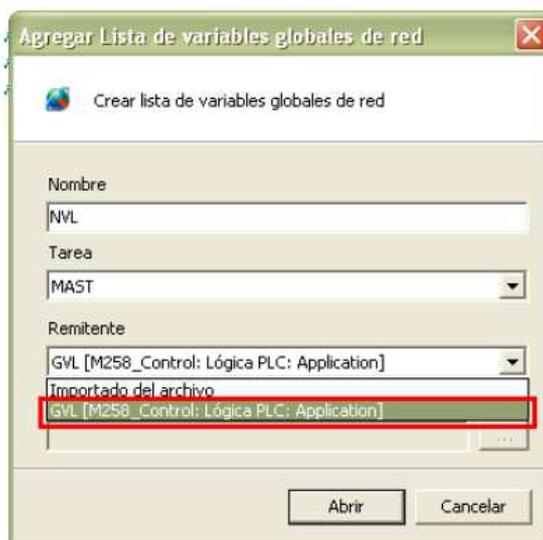
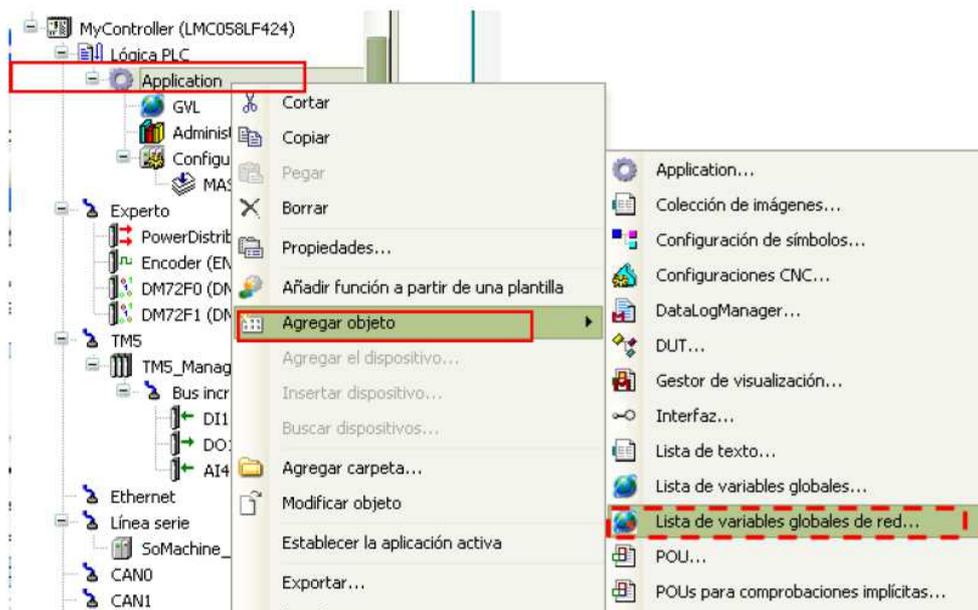


En la lista de variables globales, se insertan las variables a compartir.

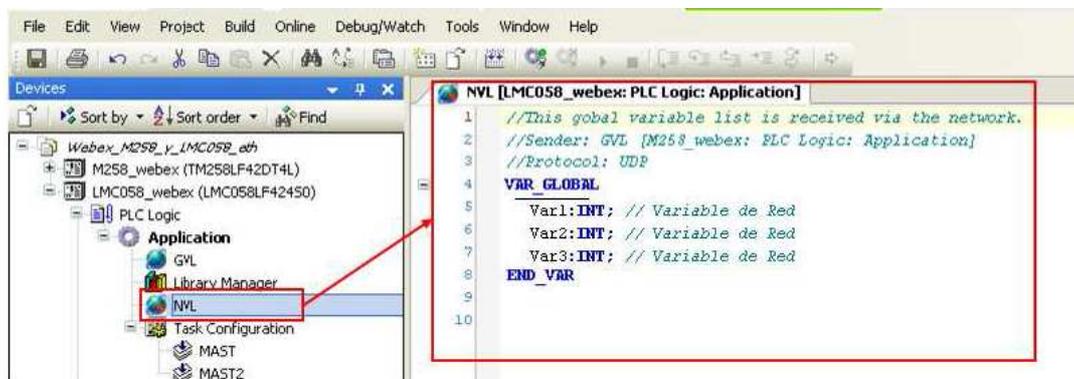


### Configuración de receptor

En la aplicación de la LMC058 se añade la Lista de Variables Globales de Red, NVL, y se asocia a la GVL del M258.



En la NVL de la LMC058 aparecen, de modo automático, las variables declaradas en la GVL del M258. Estas variables se podrán leer desde la LMC058, a través de la red.



## Servicios de Ethernet

### FTP

Cualquier cliente FTP instalado en un equipo que esté conectado al controlador (Ethernet o mediante puerto USB), sin SoMachine instalado, se puede utilizar para transferir archivos al área de almacenamiento de datos del controlador y desde ésta.



**Nota:** El servidor FTP está disponible incluso si el controlador está vacío (ninguna aplicación de usuario).

### Acceso a FTP

Quando el controlador se conecta mediante el puerto USB, el servidor FTP es accesible en la dirección 90.0.0.1.

LOGIN:       **USER**  
Password:   **USER**

La contraseña sólo puede modificarse con la función de seguridad del servidor Web

Si ha perdido u olvidado la contraseña, necesitará conectarse directamente al controlador con SoMachine y realizar un restablecimiento de origen o actualizar el firmware para restablecer la contraseña predeterminada. Después de hacerlo, debe configurar una nueva contraseña segura.



**Nota:** Para los usuarios que tienen un controlador con una versión del firmware anterior o igual a 2.0.2.0, el registro en el servidor FTP es anónimo y sin contraseña.

La memoria Flash esta organizada de la siguiente manera:

Disk	Directory	File	Content	Access	Up/Downloaded data type
/sys	OS	M258FW1v_XX.YY <sup>(1)</sup>	Firmware of core 1	Read/Write	Firmware
		M258FW2v_XX.YY <sup>(1)</sup>	Firmware of core 2		
		M258_top_Vxx.bit	Firmware		
		Version.ini	Control file for firmware version		
	Cmd	Cmd.log	Result of last script executed by the <a href="#">USB memory Key</a>	Read/Write	log file
		Script.cmd	Script executed by the USB memory Key		
	Web	Index.htm	HTML pages served by the Webserver for the website embedded in the controller.	Read/Write	Website
		Conf.htm			
		...			
	/usr	App	Application.app	Boot application	Read/Write
Application.crc					
Application.map					
Archive.prj <sup>(2)</sup>			Application source		
App/MFW		DeviceD_X_fw <sup>(2)</sup>	Expansion modules Firmware	Read/Write	Firmware
Cfg		Machine.cfg	<a href="#">Host configuration file</a>	Read/Write	Configuration
		CodesysLateConf.cfg	Name of application to launch Routing table (main/sub net) FTP login and password Machine variables		
Log	LogFile.log <sup>(2)</sup>	User log file created by <a href="#">data logging function</a> .	Read/Write	log file	

## WebServer

El servidor Web es una herramienta para leer y escribir datos, y controlar el estado del controlador, con acceso completo a todos los datos de la aplicación. El controlador proporciona un servidor web integrado con un sitio web predefinido incorporado de fábrica. Puede utilizar las páginas del sitio web para la configuración y el control del módulo así como para el diagnóstico y la supervisión de la aplicación. Están listos para su uso con un navegador web. No es necesaria ninguna configuración ni programación.

Para acceder al servidor web se puede hacer de dos maneras: conectado al puerto USB y poniendo en el explorador web la IP (**90.0.0.1**) ó a través del puerto Ethernet, colocando en el explorador web la IP configurada.



**Nota:** Por razones de seguridad de su instalación, inmediatamente después del primer inicio de sesión, debe cambiar la contraseña predeterminada.



Los diferentes menús a los que se puede acceder desde la página de inicio son los siguientes:

Menú	Submenú	Descripción
Monitorización	<a href="#">Visor del controlador</a>	<ul style="list-style-type: none"> <li>Número de serie</li> <li>Versión (firmware, arranque...)</li> <li>Estado de la configuración</li> </ul>
	<a href="#">Visualizador de ampliaciones</a>	Ver estado de módulos de ampliación.
	<a href="#">Visualizador E/S</a>	Ver valores E/S módulo por módulo.
	<a href="#">Osciloscopio</a>	Visualización de dos variables en forma de cronograma de tipo de registro.
	<a href="#">Parámetros de datos</a>	Mostrar y modificar las variables del controlador.
Diagnóstico	<a href="#">PLC</a>	Estado del controlador
	<a href="#">Ethernet</a>	Estado de Ethernet
	<a href="#">Serie</a>	Estado de la línea serie
Mantenimiento	<a href="#">FTP</a>	Enlace al servidor del sistema de archivos (carpetas /User, /bd0y /Sys)
Configuración	<a href="#">Configuración de Post</a>	Establecer los parámetros de Ethernet y de línea serie.
	<a href="#">Archivos de configuración EthernetIP</a>	Establece los archivos de configuración EthernetIP.
	Seguridad	Modifique la contraseña de inicio de sesión del usuario (la contraseña predeterminada es USER).

## Ejemplo de Webvisualización

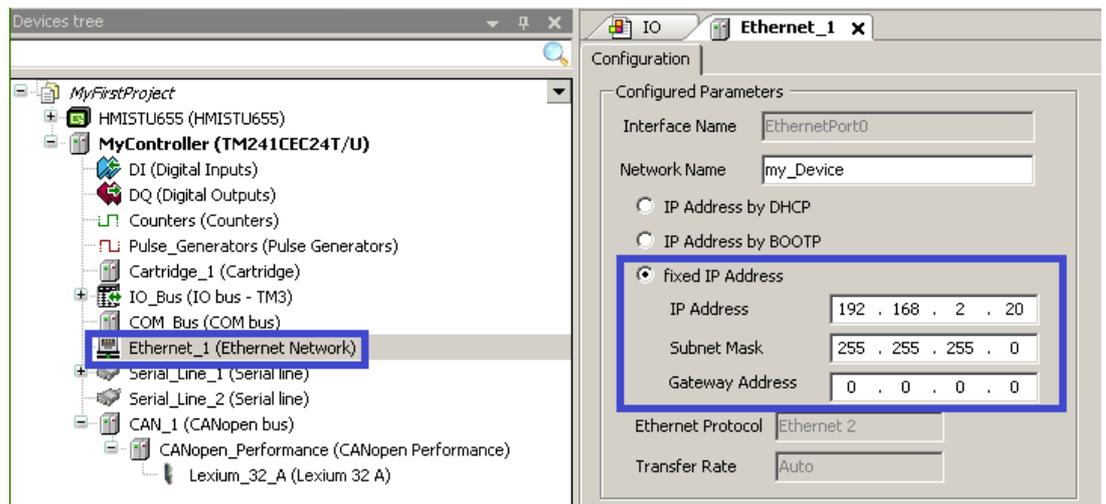
### Introducción

En el caso de los controladores M241 y M251 si disponen de Ethernet dispondrán de la webvisualización. Es una manera de publicar en un webserver en html5 de las visualizaciones previamente creadas para ello se ha de proceder de la siguiente manera.

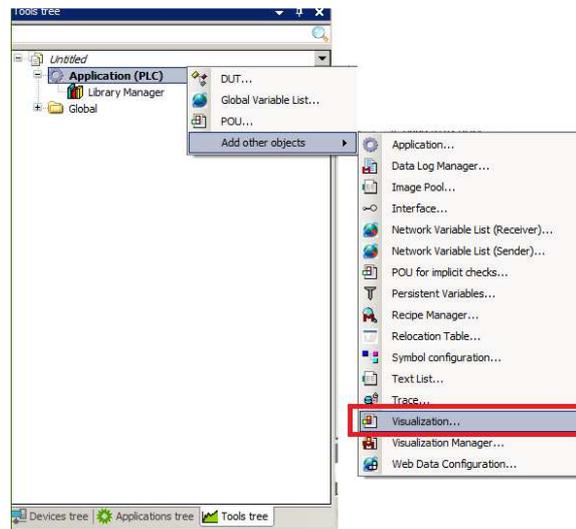
Para el ejemplo utilizaremos un controlador M241CEC24R con el puerto Ethernet integrado, este irá conectado a un router wifi TP LINK.



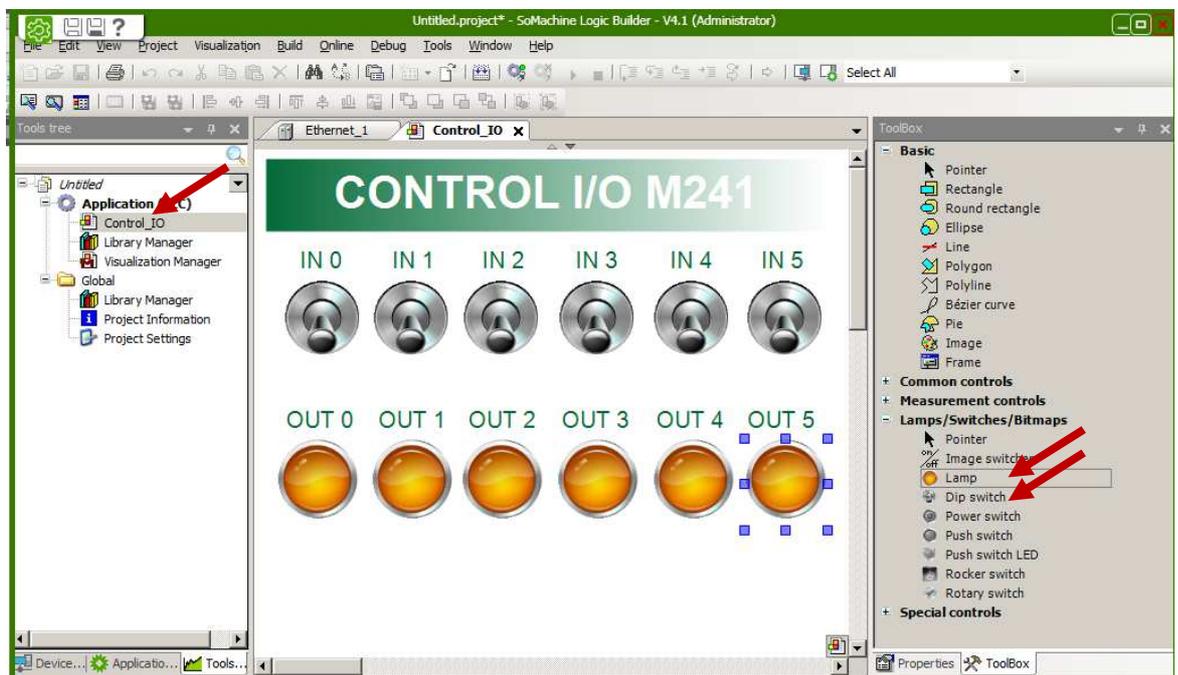
1. Crearemos un proyecto nuevo con el controlador **M241CEC24R**.
2. Asignaremos símbolos, tanto a las entradas digitales **'IN\_M241\_x'** como a las salidas digitales **'OUT\_M241\_x'** que tienen embebidas el controlador M241.
3. Luego asignaremos la IP: **192.168.2.20** al puerto de **'ethernet\_1'** que se encuentra en la pestaña de **'Dispositivos'** del navegador.



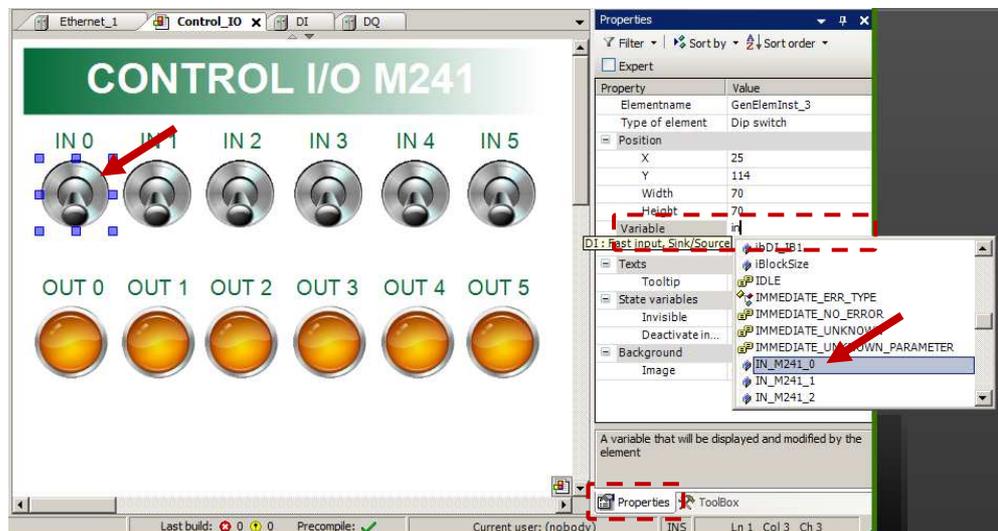
4. Ahora crearemos una visualización para ello nos iremos a la pestaña de **'Herramientas'** del navegador y añadiremos el objeto visualización.



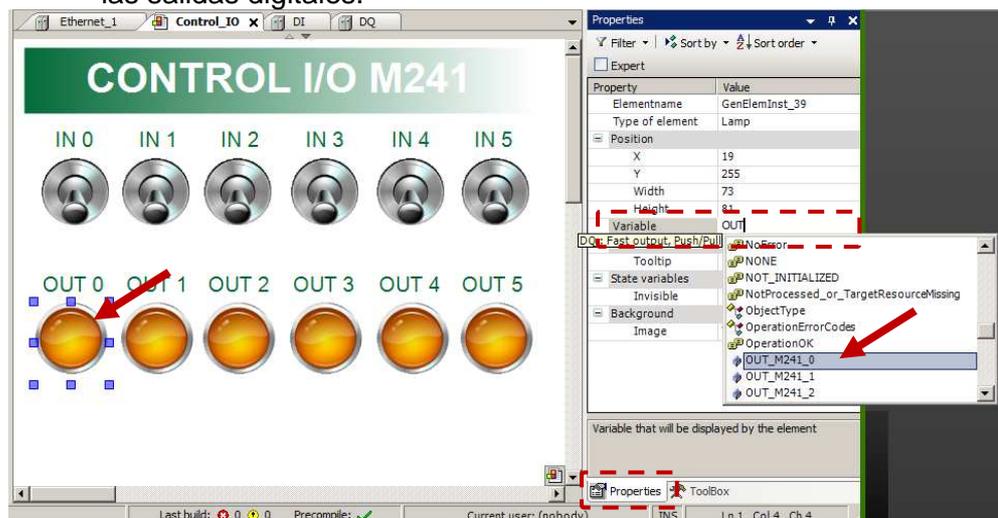
5. En la visualización, que le llamaremos **'Control\_IO'**, añadiremos 6 **'Dip switch'** que se encuentran en la ventana de **'ToolBox'** desplegamos **'Lamps/Switches/Bitmaps'**. También añadimos 6 **'Lamps'** que se encuentran el mismo sitio.



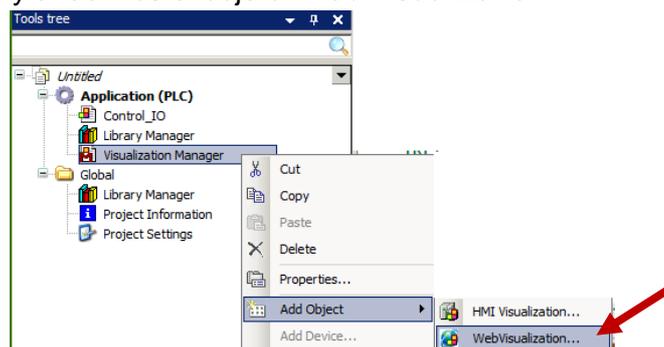
- Ahora tenemos que asociar los símbolos de las entradas digitales del controlador M241 a cada uno de los **'Dip switch'** que hemos creado. Seleccionamos uno, abrimos la pestaña de **'Properties'**, desplegamos **'Position'** y seleccionamos el campo **'Variable'** aquí seleccionamos el símbolo de las entrada digital correspondiente que habíamos creado previamente.



- Hacemos lo mismo con las **'Lamps'** que hemos creado, pero para las salidas digitales.



- Una vez hemos asignado las variables a los elementos ya estará terminada la visualización, tendremos que decirle al programa que la vamos a publicar vía web añadiendo el **'web visualization'**, para ello seleccionaremos **'Visualization Manager'** botón derechos y añadimos el objeto **'Web visualization...'**.



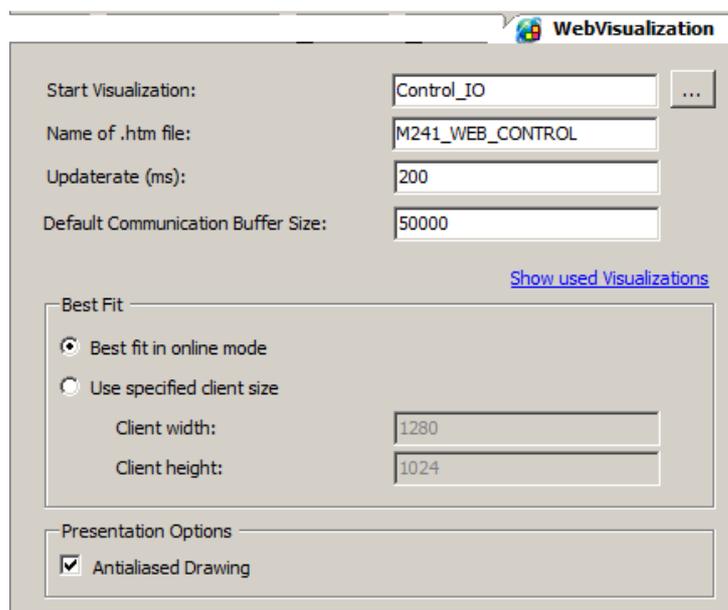
9. Configuraremos el **'Webvisualization'**.

En el indicaremos la **'visualización de inicio'** (en este caso como solo tenemos una, ya aparece el nombre de la visualización **'Control\_IO'**).

Le tendremos que especificar un nombre al archivo \*.htm (en este caso será **'M241\_WEB\_CONTROL'**).

Dejaremos el refresco de la página en **20 ms** y el buffer de comunicación en **50000**.

Por último dejaremos que la pantalla se adapte al tamaño del dispositivo que se conecte en modo online. (tablet, móviles ...etc) **'Best fit in online mode'**.



10. Una vez configurado el **'Webvisualization'**, ya podríamos compilar y cargar el programa en el equipo.

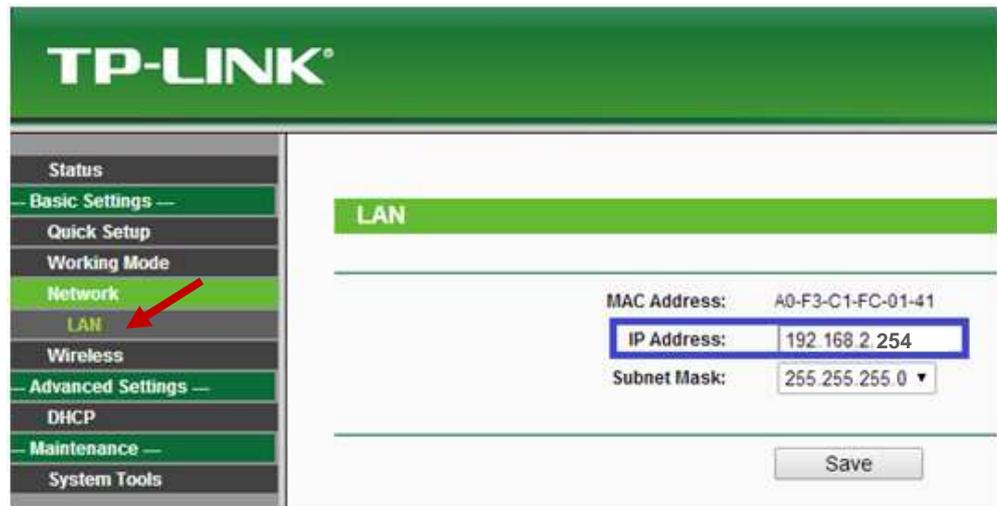
11. En el caso del ejemplo, ahora tenemos que configurar el router\_wifi TP Link que puede diferir de otros equipos. Para configurar conectamos punto a punto con un cable de Ethernet, el equipo TP Link al PC. Abrimos el navegador (internet explore, chrome ...etc) y escribimos **192.168.0.254**. (Si el router ya tiene otra red wifi que la de fábrica , pulsamos el botón de reset)

12. Te pedirá el Usuario y contraseña.

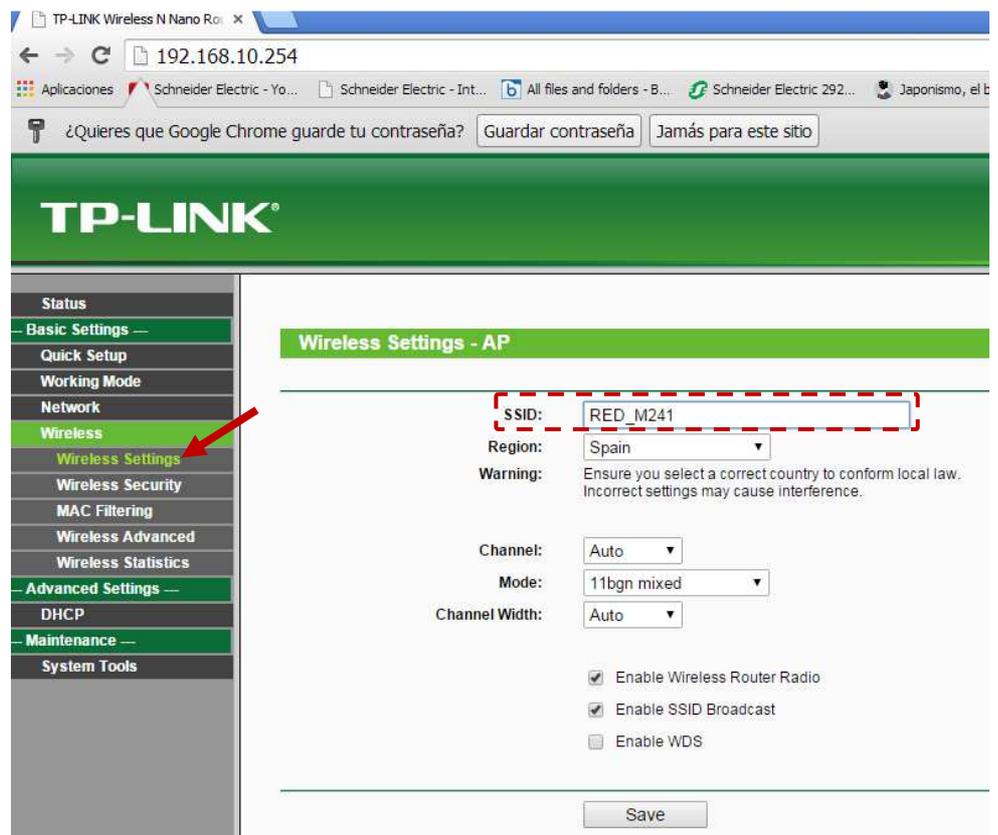
**Login: admin**

**Password: admin**

13. Ahora en la página web de configuración del TP-LINK en 'Basic Setting' seleccionamos 'Network' y 'LAN' en campo 'IP Address' escribimos '192.168.20.254'.



14. Ahora en 'Basic Setting' seleccionamos 'Network' y 'Wireless / Wireless settings' y en campo 'SSID' escribimos 'RED\_M241'.



15. Después para activar la seguridad de la red wifi, hay que ir a 'Basic Setting' seleccionamos 'Network' y 'Wireless / Wireless security' seleccionamos la seguridad 'WPA-PSK/WPA2-PSK' y en campo 'PSK Password' escribimos 'PLC\_M241'.

TP-LINK

Wireless Security

WPA/WPA2

Version: Automatic

Encryption: Automatic

Radius Server IP:

Radius Port: 1812 (1-65535, 0 stands for default port 1812)

Radius Password:

Group Key Update Period: 86400 (in second, minimum is 30, 0 means no update)

WPA-PSK/WPA2-PSK

Version: Automatic

Encryption: Automatic

PSK Password: PLC\_M241  
(You can enter ASCII characters between 8 and 63 or Hexadecimal characters between 8 and 64.)

Group Key Update Period: 86400 (in second, minimum is 30, 0 means no update)

Save

16. Por último, antes de hacer un reboot del TP-LINK, nos aseguramos que en 'Advance Settings / DHCP / DHCP settings' el campo 'DHCP Server' está en 'Enable' y seleccionamos el rango de IP que se le asignarán a los equipos que se conecten a la red vía wifi (asegurarse que ningún equipo de la red esté dentro de ese rango de IP's).

TP-LINK

DHCP Settings

DHCP Server:  Disable  Enable

Start IP Address: 192.168.10.100

End IP Address: 192.168.10.199

Address Lease Time: 120 minutes (1~2880 minutes, the default value is 120)

Default Gateway: 0.0.0.0 (optional)

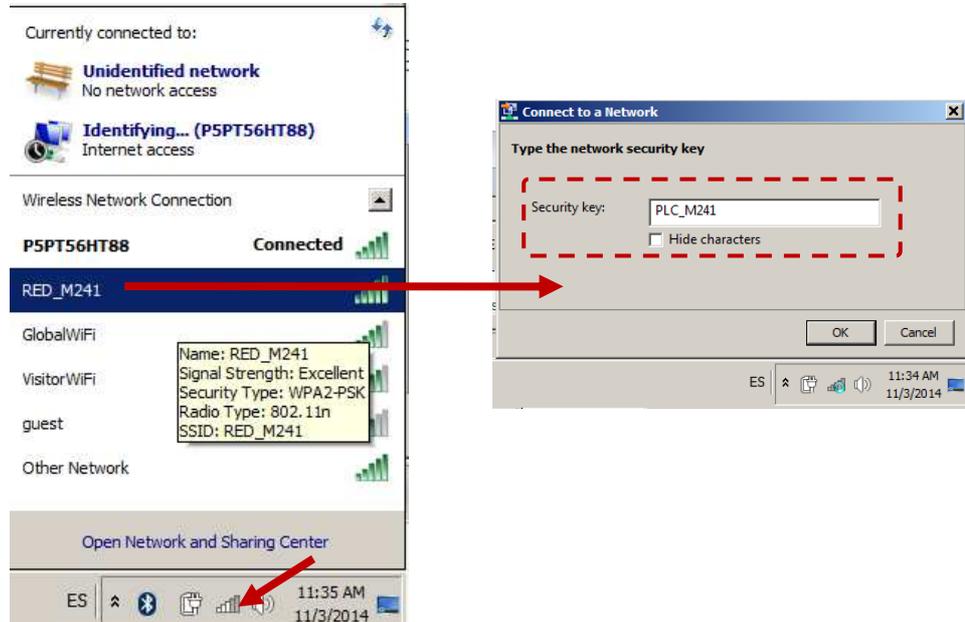
Default Domain: (optional)

Primary DNS: 0.0.0.0 (optional)

Secondary DNS: 0.0.0.0 (optional)

Save

17. Ahora para conectarse para ver la web visualization, desde el dispositivo donde queremos visualizarlo, vamos a la conexiones wifi y seleccionamos la red '**RED\_M241**', al seleccionarla nos pedirá el password de la red para conectarse, escribimos '**PLC\_M241**'. De esta manera ya estaremos conectados vía wifi a nuestra red.



18. Una vez conectados a la red wifi, solo tendremos que abrir nuestro explorador de internet (IE, Chrome, Safari ...etc, una versión que tenga html\_5) y escribir la siguiente dirección en el explorador '<dirección IP del controlador> :8080/<nombre del archivo htm>.htm', en este caso '**http://192.168.2.20:8080/M241\_WEB\_CONTROL.htm**'.



19. Cuando se conecte veremos y podremos actuar sobre las visualizaciones publicadas sobre el web visualization.



# Capítulo 15: Vinculación variables con HMI

## Descripción

**Introducción** Este capítulo trata de cómo realizar la configuración del puerto Ethernet y las diferentes posibilidades de comunicación que el software SoMachine ofrece bajo este bus de comunicación.

### Este CAPÍTULO trata los siguientes temas:

- Introducción ..... 15-2
- Añadir una HMI a la arquitectura ..... 15-2
- Configurador de símbolos ..... 15-3
- Añadir variables en el Vijeo Designer.....15-4
- Cambiar nombre del controlador en el SoMachine .....15-5
- Añadir nombre del controlador en el Vijeo Designer... 15-7

## Añadir una HMI a la aplicación

### Introducción

En la mayoría de las arquitecturas de una máquina, actualmente disponen de una HMI para la supervisión, monitorización y diagnóstico de la misma por parte del operario.

En el SoMachine estos equipos y el software necesario para crear la aplicación de supervisión de la pantalla están enlazados con los del proyecto del SoMachine, el Vijeo Designer está asociado al SoMachine, así que cuando añadimos un terminal HMI, este se abrirá automáticamente.

### Añadir una HMI a la arquitectura

Para añadir una HMI a la configuración de la arquitectura, debemos realizar los siguientes pasos:

1. En la ventana principal del **SoMachine Central**, dentro de la pestaña '**Flujo de Trabajo**', pulsamos '**Configuración**'.
2. En la ventana de configuración, en la parte izquierda 'Catalogo', desplegamos 'Magelis HMI & IPC', buscamos la Serie del modelo HMI que deseamos añadir al proyecto, desplegamos la Serie y seleccionamos el modelo, una vez seleccionado pulsamos el botón de 'añadir dispositivo al proyecto'.



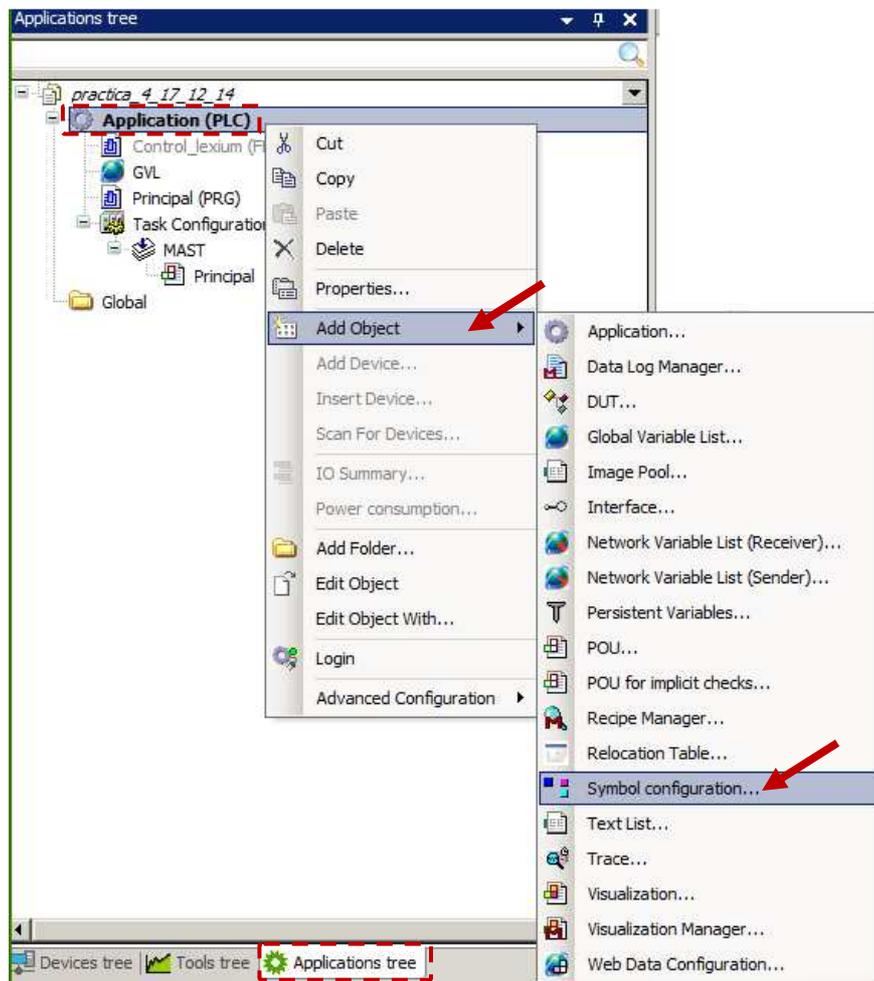
3. Una vez añadido y cuando hayamos confirmado los cambios pulsando el botón OK, al salir de la ventana de configuración, se abrirá automáticamente el Vijeo Designer.

## Configurador de símbolos

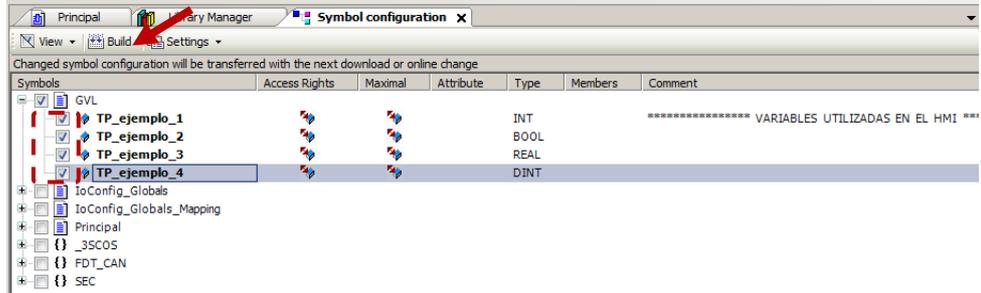
Primero de todo, crearemos un programa, donde se han creado diferentes variables, que son las que se desean utilizar en la HMI. Hay que tener en cuenta que el programa tendrá que estar Generado y sin errores. (En este ejemplo las variables que vamos a utilizar en el HMI, las hemos agrupado en una lista de variables globales 'GVL'.

```
1  VAR GLOBAL
2  //***** VARIABLES UTILIZADAS EN EL HMI *****/
3  TP_ejemplo_1:INT;
4  TP_ejemplo_2:BOOL;
5  TP_ejemplo_3:REAL;
6  TP_ejemplo_4:DINT;
7  END_VAR
```

Ahora seleccionar la pestaña 'Aplicación' y, seleccionamos 'Application' y haciendo clic con el botón derecho, seleccionamos 'Agregar Objeto / Configurador de símbolos'.



Pulsar **'Build'** si no veis la variables, ahora desplegar **'GVL'** y seleccionar todas aquellas variables que queremos publicar en el HMI. Una vez realizada esta acción las variables están publicadas (disponibles para el HMI).

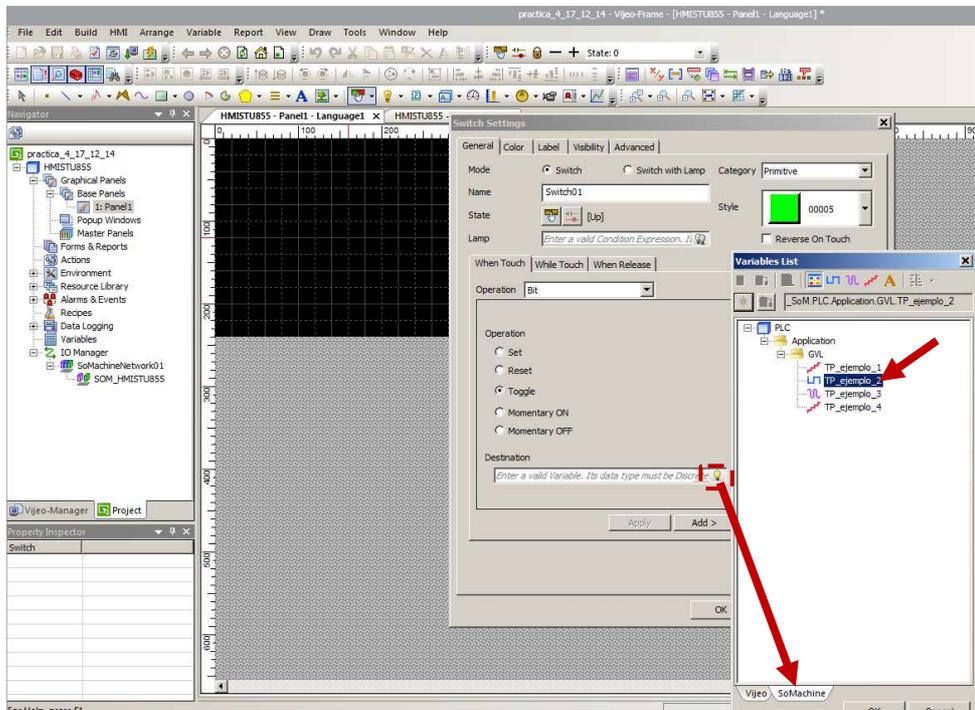


**Nota:** Las variables globales GVL solo se verán en el configurador de símbolos si estas están utilizadas en alguna parte del programa.

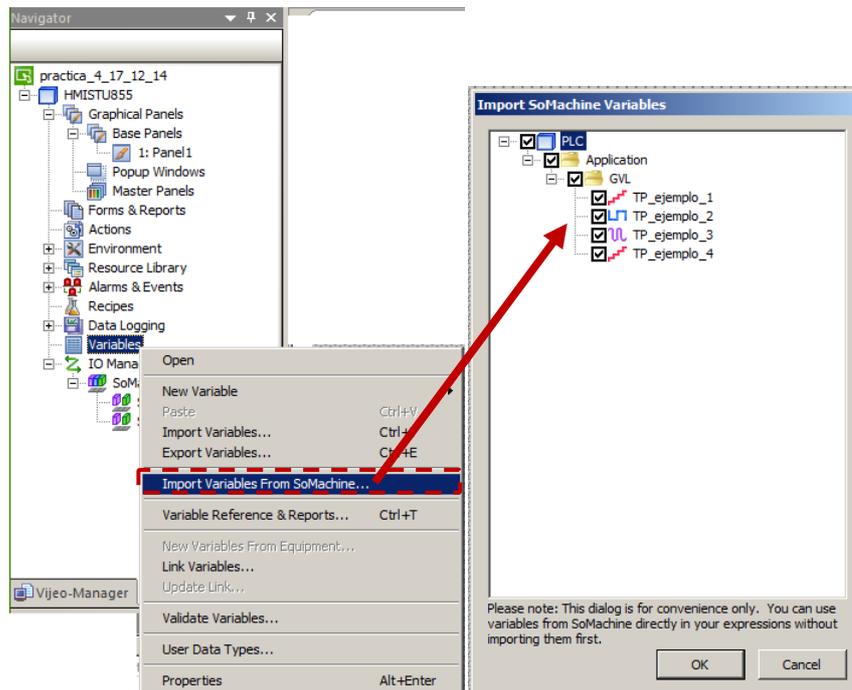
### Añadir variables en el Vijeo Designer

Una vez hecho el Build en el configurador de símbolos del SoMachine, podemos ir al software de la HMI **'Vijeo Designer'** y las variables ya estarán disponibles directamente para ser utilizadas.

Por ejemplo si queremos asociar la variable TP\_ejemplo\_2, que era booleana a un botón de la pantalla, al querer seleccionar la variable aparece una pestaña **'SoMachine'** donde se encontrarán las variables seleccionadas en el **'Configurador de símbolos'**.



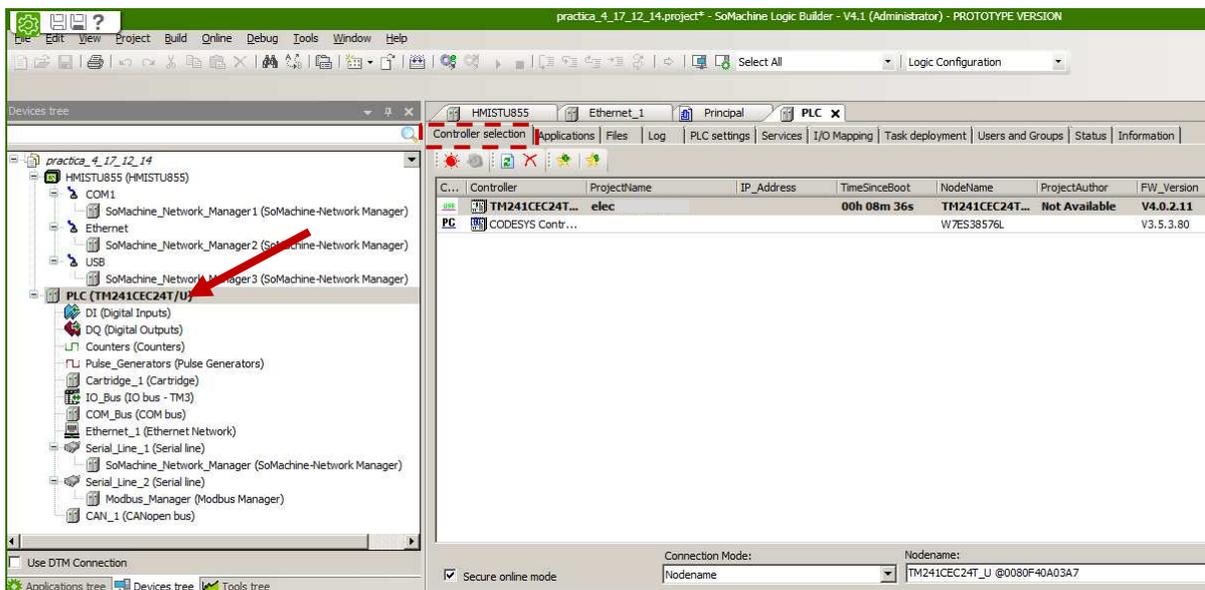
Para verlas en el Editor de Variables del Vijeo Designer lo único que hay que hacer, es seleccionar **'Variables'**, botón derecho **'Importar variables desde SoMachine'**. Aparecerá una ventana donde podemos seleccionar las variables que queremos importar.



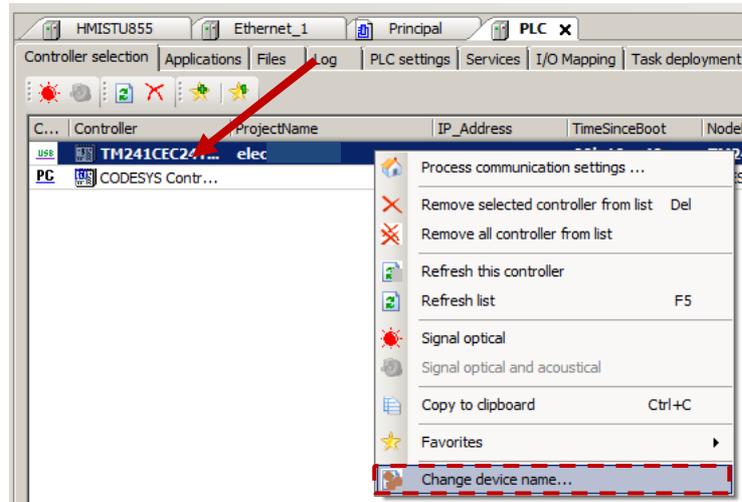
## Cambiar nombre del controlador en el SoMachine

Para poder vincular correctamente las variables entre el controlador y la HMI, es necesario conocer el **'nombre'** específico del controlador. Al conectarnos por primera vez al controlador, sabremos el nombre por defecto, el nombre está compuesto por el **'modelo de controlador @ dirección\_MAC'** (Ejemplo: *TM241CEC24T\_U@0080F40A03A7*).

1. En el SoMachine dentro de la ventana de **'Logic Builder'**, en la pestaña de **'Dispositivos'** del navegador haremos doble clic sobre el controlador, en el área de trabajo, seleccionaremos la pestaña **'Controller selection'**. Si tenemos el cable USB conectado al controlador, este escaneará automáticamente y nos mostrará los controladores conectados.



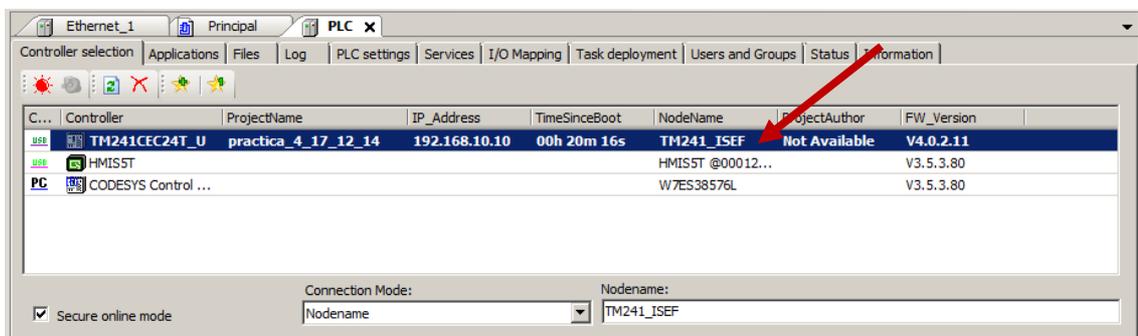
Ahora seleccionamos el controlador al que deseamos cambiar el nombre, primero hacemos doble clic sobre él, este se pondrá en negrita. Luego con el controlador seleccionado, con el botón derecho desplegamos el menú de opciones y seleccionamos **‘Cambiar el nombre del dispositivo...’**



Aparecerá una ventana donde nos indica el nombre actual y en el campo **‘Nuevo’** escribiremos el nombre que deseamos para este controlador (Ejemplo: *TM241\_ISEF*).

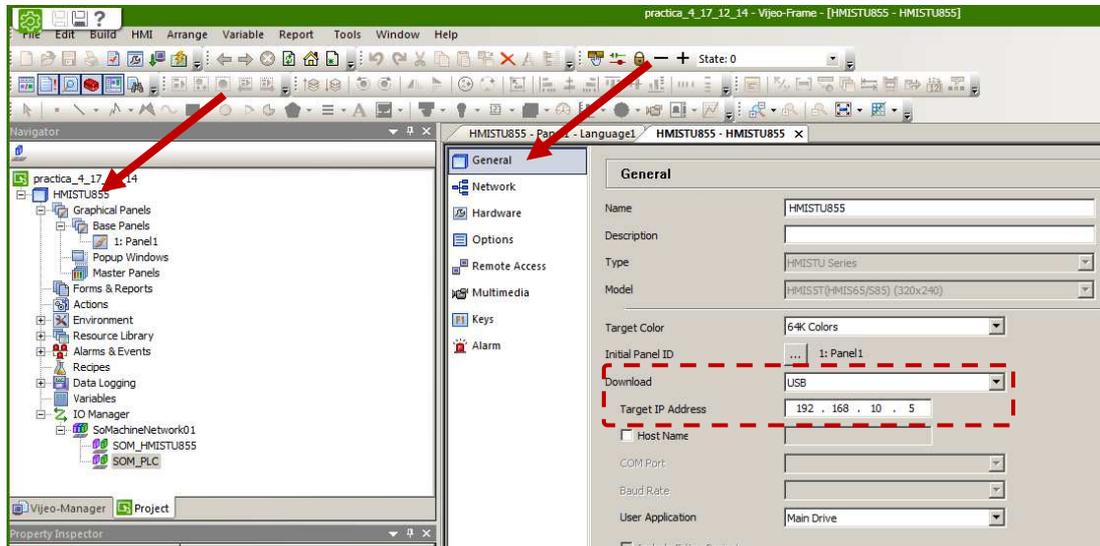


Ahora en la columna **‘Node name’** habrá cambiado el nombre por el nuevo.

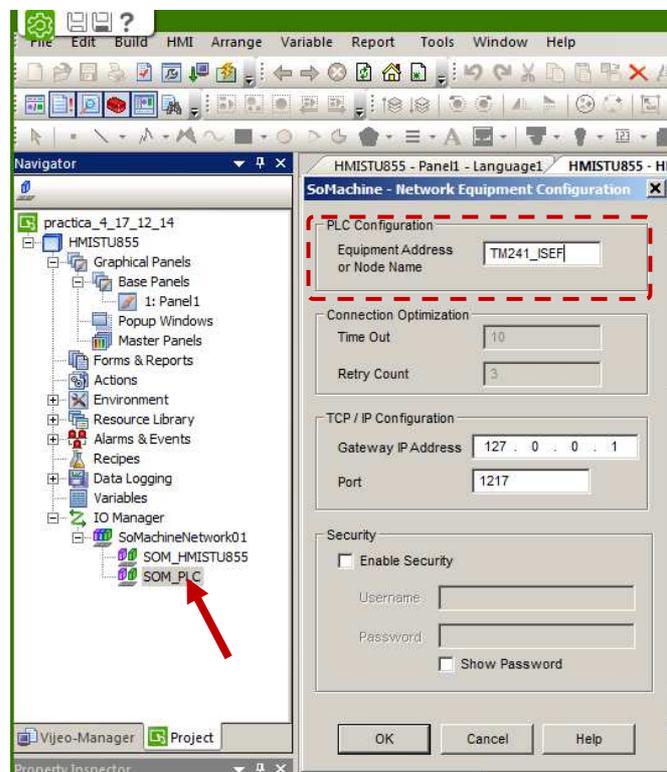


## Añadir nombre del controlador en el Vijeo Designer

Ahora, una vez sabemos el nombre del controlador, volvemos abrir el Vijeo Designer y en el Navegador seleccionamos la pantalla y en la pantalla 'General' colocaremos la IP de la pantalla en el campo 'IP Adress'. También en el campo 'Download' pondremos 'USB' ya que la primera descarga del programa se tendrá que realizar por USB (no se podrá hacer una descarga múltiple la primera vez).



Una vez hemos colocado la IP en la HMI, en el navegador desplegaremos el campo 'IO Manager / SoMachine Network 01' y haremos doble clic en el 'SOM\_PLC' y en la ventana flotante de configuración, en el campo 'Name Equipement or Node name' escribiremos en nombre del controlador que hemos escrito en el SoMachine (Ejemplo: TM241\_ISEF).



Una vez descargado el programa en el HMI y en el controlador, las variables quedarán linkadas y listas para ser utilizadas.