

ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA INDUSTRIAL DE ZARAGOZA



AUTOMATIZACIÓN DE CÉLULA FLEXIBLE DE FABRICACIÓN

Memoria

<u>Curso:</u>	2011/2012
<u>Dpto.:</u>	Informática e Ingeniería de Sistemas
<u>Titulación:</u>	Ingeniero Técnico Industrial
<u>Especialidad:</u>	Electrónica Industrial
<u>Integrantes:</u>	Rubén Teller Sanz
<u>Tutor:</u>	Ramón Piedrafita Moreno
<u>Fecha:</u>	Diciembre 2011



1- OBJETIVO DEL PROYECTO	6
2- INTRODUCCIÓN A LA PLATAFORMA DE AUTOMATIZACIÓN	7
2.1- DESCRIPCIÓN GENERAL DEL CONTROLADOR LÓGICO MODULAR MODICON M258.....	7
2.2- PRINCIPALES FUNCIONES DEL CONTROLADOR LÓGICO MODULAR MODICON M258.....	8
2.2.1- Funciones Analógicas:.....	8
2.2.2- Función de contaje rápido:	8
2.2.3- Función de control de posición:.....	9
2.2.4- Funciones de comunicación:	9
2.3- DESCRIPCIÓN DE LA BASE COMPACTA ‘TM258LF42DT4L’	10
2.4- COMUNICACIONES. PUERTO DE BUS CANOPEN.....	12
2.4.1- Introducción:.....	12
2.4.2- Protocolo CANopen:.....	12
2.4.2.1- Características Capa física:	12
2.4.2.2- Características perfiles CANopen:	13
2.4.2.3- Perfiles de comunicaciones:	13
2.4.3- Características puerto CANopen del controlador M258:	14
2.5- CONFIGURACIÓN DEL SOFTWARE: SOMACHINE.....	15
2.5.1- Introducción:.....	15
2.5.2- Características de SoMachine:	15
2.5.3- Interfaz de SoMachine:	16
2.5.3.1- Pantalla de selección principal:.....	16
2.5.3.2- Introducción a la barra de tareas:	18
2.5.3.3- Pantalla de Propiedades:	18
2.5.3.4- Pantalla de Configuración:.....	19
2.5.3.5- Pantalla de Programa:.....	20
2.5.3.6- Pantalla de Puesta en marcha:	23
2.5.3.7- Pantalla de Informe:	24
2.6- CONFIGURACIÓN ISLAS ADVANTYS	25
2.6.1- Introducción:.....	25
2.6.2- Características y Composición:	25
2.6.3- Módulo NIM STB NCO 2212:	27
2.6.4- Interface de bus de campo CANopen:	29
3- CURSO DE PROGRAMACIÓN Y CREACIÓN DE APLICACIONES.....	30
3.1- CREACIÓN DE APLICACIONES MEDIANTE SOMACHINE.....	30
4- CREACIÓN ISLAS ADVANTYS.....	60
4.1- INTRODUCCIÓN A LA CONFIGURACIÓN DE ISLAS ADVANTYS.....	60
4.2- CÓMO AGREGAR CONFIGURACIÓN ISLA ADVANTYS A SOMACHINE	64
5- PROGRAMACIÓN EN SFC	66
5.1- INTRODUCCIÓN. NORMA IEC 1131.....	66
5.2- INTRODUCCIÓN AL LENGUAJE SFC.....	67
5.2.1- Definición:	67
5.2.2- Estructura:.....	67

5.2.3- Elementos del lenguaje SFC:.....	68
5.2.3.1- <i>Etapas</i> :.....	68
5.2.3.2- <i>Transiciones</i> :.....	68
5.2.3.3- <i>Acciones</i> :.....	68
5.2.3.4- <i>Salto</i> s:.....	69
5.2.3.5- <i>Otros Elementos: Divergencias y Convergencias</i>	69
5.2.4- Reglas de evolución:.....	70
5.3- PROGRAMACIÓN EN SFC EN SOMACHINE.....	71
5.3.1- Creación POU:.....	71
5.3.2- Diseño y funcionalidades:.....	72
6- INTRODUCCIÓN A LA CÉLULA DE FABRICACIÓN.....	82
6.1- DESCRIPCIÓN GENERAL DE LA CÉLULA.....	82
6.1.1- Características del producto a fabricar:.....	83
6.1.2- Composición de las piezas a fabricar:.....	84
6.1.2.1 <i>Cilindros Neumáticos</i> :.....	84
6.1.2.2 <i>Tapa</i> :.....	84
6.1.2.3 <i>Piezas sólo con Tapa</i> :.....	84
6.1.3- Características de los elementos de las piezas:.....	85
6.2- CARACTERÍSTICAS GENERALES DE LAS ESTACIONES.....	85
6.2.1- Zona de Fabricación:.....	87
6.2.2- Zona de Almacén Intermedio:.....	89
6.2.3- Zona de Expedición:.....	90
6.3- DESCRIPCIÓN ESTACIÓN 1.....	92
6.3.1- Introducción a la estación 1:.....	92
6.3.2- Descripción general de la estación 1:.....	95
6.3.3- Descripción hardware de la estación 1:.....	95
6.3.3.1 <i>Autómata</i> :.....	95
6.3.3.2 <i>Isla Advantys</i> :.....	96
6.3.3.3 <i>Módulos de precableado de entradas y salidas</i> :.....	97
6.3.3.4 <i>Captadores</i> :.....	98
6.3.3.5 <i>Actuadores</i> :.....	101
6.4- DESCRIPCIÓN ESTACIÓN 3.....	103
6.4.1- Introducción a la estación 3:.....	103
6.4.2- Descripción general de la estación 3:.....	105
6.4.3- Descripción hardware de la estación 3:.....	105
6.4.3.1 <i>Autómata</i> :.....	105
6.4.3.2 <i>Isla Advantys</i> :.....	105
6.4.3.3 <i>Módulos de precableado de entradas y salidas</i> :.....	105
6.4.3.4 <i>Captadores</i> :.....	107
6.4.3.5 <i>Actuadores</i> :.....	108
6.5- DESCRIPCIÓN ESTACIÓN 4.....	110
6.5.1- Introducción a la estación 4:.....	110

6.5.2- Descripción general de la estación 4:	112
6.5.3- Descripción hardware de la estación 4:	113
6.5.3.1 <i>Autómata</i> :	113
6.5.3.2 <i>Isla Advantys</i> :	113
6.5.3.3 <i>Equipos de vacío</i> :.....	115
6.5.3.4 <i>Módulos de precableado de entradas y salidas</i> :	115
6.5.3.5 <i>Captadores</i> :	116
6.5.3.6 <i>Actuadores</i> :	118
6.6- DESCRIPCIÓN ESTACIÓN 6	120
6.6.1- Introducción a la estación 6:	120
6.6.2- Descripción general de la estación 6:	123
6.6.3- Descripción hardware de la estación 6:	123
6.6.3.1 <i>Autómata</i> :	123
6.6.3.2 <i>Isla Advantys</i> :	123
6.6.3.3 <i>Módulos de precableado de entradas y salidas</i> :	123
6.6.3.4 <i>Captadores</i> :	125
6.6.3.5 <i>Actuadores</i> :	126
7- GUIONES DE PRÁCTICAS	130
7.1- PRÁCTICA 0. INTRODUCCIÓN A LA PROGRAMACIÓN.....	130
7.2- PRÁCTICA 1. PROGRAMACIÓN ESTACIÓN 3	136
7.3- PRÁCTICA 2. PROGRAMACIÓN ESTACIÓN 6	141
7.4- PRÁCTICA 3. PROGRAMACIÓN ESTACIÓN 4	145
7.5- PRÁCTICA 4. PROGRAMACIÓN CONJUNTA DE ESTACIONES.....	150

1- OBJETIVO DEL PROYECTO

El objetivo principal de este proyecto consiste en la automatización de una célula de fabricación flexible de cilindros neumáticos perteneciente a la Universidad de Zaragoza, ubicada en el edificio Ada Byron del Centro Politécnico Superior.

Para ello, la compañía Schneider ha cedido a la Universidad uno de sus nuevos autómatas de la gama T258, en concreto el controlador TM258LF42DT4L. Por tanto, otro de los objetivos del proyecto es la familiarización con dicho autómata para la posterior programación de las estaciones que componen la célula de fabricación.

Con el fin de cumplir con el objetivo principal del proyecto de automatización de una máquina, se desarrollará la programación de las estaciones 1, 3, 4 y 6 que forman parte de la célula de fabricación.

El lenguaje elegido entre los seis lenguajes normalizados por la norma IEC 1131 y posteriormente utilizado en la programación de dichas estaciones ha sido el lenguaje SFC por lo que el aprendizaje de dicho lenguaje de fabricación ha sido otro de los objetivos dentro de este proyecto.

Los autómatas de Schneider disponen de un software propio, denominado SoMachine, para la programación y configuración de los controladores. El aprendizaje de la configuración de autómatas y su posterior programación con dicho software ha sido otro de los objetivos recogidos en este proyecto.

El proyecto se estructurará en siete capítulos dedicados a:

1. Objetivos del proyecto.
2. Introducción a la plataforma de automatización, SoMachine.
3. Curso de programación en los distintos lenguajes recogidos por la norma IEC 1131.
4. Creación de Islas Advantys
5. Programación en SFC.
6. Introducción a la célula de fabricación.
7. Guiones de prácticas.

Como puede observarse en la estructura del proyecto, otros de los objetivos son la creación de un curso de programación en los diferentes lenguajes de programación que están normalizados por la Comisión Electrotécnica Internacional así como la creación de unos guiones de prácticas relativos a las estaciones programadas con anterioridad para su posterior uso en prácticas de laboratorio de cursos posteriores del Grado en Ingeniería Electrónica y Automática.

En definitiva, la realización de este proyecto ha tenido como fin el aprendizaje de nuevos lenguajes de programación así como la familiarización con una familia de autómatas y el software propio de control de dichos controladores.

2- INTRODUCCIÓN A LA PLATAFORMA DE AUTOMATIZACIÓN

2.1- DESCRIPCIÓN GENERAL DEL CONTROLADOR LÓGICO MODULAR MODICON M258

El controlador lógico Modicon M258 es un autómatas compacto de alto rendimiento y totalmente ampliable. Se trata de un sistema modular flexible.

Este autómatas está diseñado para fabricantes de máquinas o equipos originales (OEM) centrados especialmente en aplicaciones como pueden ser el embalaje, el almacenamiento o la manutención, y ofrece soluciones de alto rendimiento para funciones de comunicación, control de velocidad, contaje y control de ejes.

Atendiendo a términos de rendimiento, el controlador M258 presenta un procesador de doble núcleo donde:

- El Núcleo 1 está dedicado exclusivamente a la gestión de las tareas de los programas ofreciendo los máximos recursos para la ejecución en tiempo real del código de la aplicación.
- El Núcleo 2 está dedicado a la ejecución de las tareas de comunicación por lo que no afectarán en absoluto al rendimiento para la ejecución de aplicaciones.

El controlador M258 tiene capacidad para gestionar 2400 E/S, una memoria RAM de 64 Mbytes para almacenar datos y programas, y una memoria Flash de 128 Mbytes para copias de seguridad de datos y aplicaciones. Además posee una velocidad de ejecución para una instrucción booleana de 22 ns es decir, puede ejecutar por 'ms' más de 45.000 instrucciones booleanas.

En el desarrollo de dicho controlador, se han tenido muy en cuenta los costes incluyendo las CPU integrados de serie 42 ó 66 E/S digitales, un puerto serie y un puerto Ethernet integrado, y 4 entradas analógicas.

Así mismo, el controlador M258 ha sido diseñado para minimizar los costes de embalaje, cableado, puesta en servicio y mantenimiento. Para ello, en su desarrollo:

- Todos los módulos poseen terminales extraíbles
- Todas las conexiones eléctricas se realizan con terminales de resorte agilizando el proceso de cableado
- Cada terminal cuenta con un punto de prueba para voltímetros
- El puerto serie y el puerto Ethernet integrados tienen una conexión RJ45 a 45° con gran visibilidad con el fin de una rápida conexión de los canales de comunicación
- Ha sido optimizada la modularidad de las diversas bases y de los distintos módulos de extensión, los cuales, tienen una capacidad de 2 a 42 canales por módulo

2.2- PRINCIPALES FUNCIONES DEL CONTROLADOR LÓGICO MODULAR MODICON M258

2.2.1- Funciones Analógicas:

El controlador lógico M258 incluye de serie cuatro entradas analógicas con una resolución de 12 bits. Además, existe una gran gama de módulos de extensión, tanto modulares como compactos, así como de funciones de programación avanzada para aquellas máquinas que requieren funciones para el procesamiento de datos emitidos por sensores o accionadores de tensión o intensidad, sensores de temperatura o sensores de control PID.

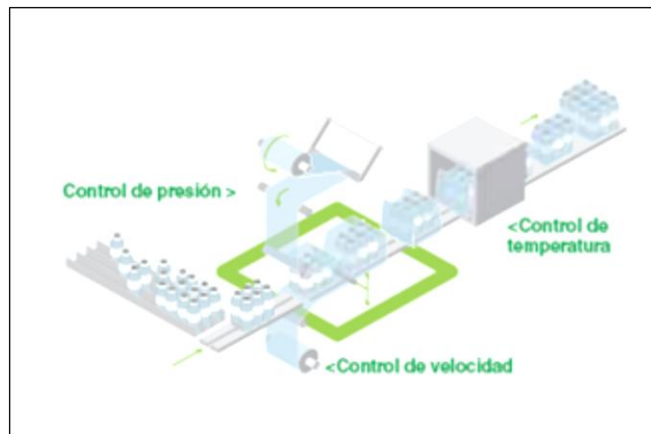


Fig. 1: Funciones Analógicas

2.2.2- Función de contaje rápido:

El controlador lógico M258 tiene ocho contadores incorporados de alta velocidad con una frecuencia de contaje de 200 kHz por canal así como cuatro salidas reflejas. La disponibilidad de dichos contadores junto con el puerto maestro CANopen facilita y agiliza la creación de funciones multieje.

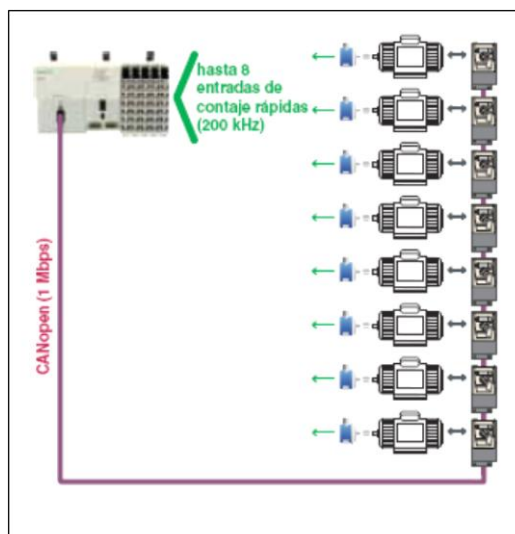


Fig. 2: Función Contaje Rápido

2.2.3- Función de control de posición:

Pueden crearse aplicaciones para el control de posición y ser controladas mediante el puerto maestro CANopen integrado o bien crear secuencias y utilizar las E/S para la comunicación con el controlador.



Fig. 3: Control de posición

2.2.4- Funciones de comunicación:

Todos los controladores lógicos M258 poseen un puerto Ethernet, un puerto maestro CANopen y un puerto serie donde:

- El puerto 'Ethernet' RJ45 (10/100 Mbps, MDI/MDIX) integrado dispone de las siguientes características, como pueden ser, Modbus Ethernet TCP, Dispositivo Ethernet IP, SoMachine por Ethernet, así como los protocolos UDP, TCP y SNMP. Además, dispone de un servidor Web y de un servidor FTP integrados.
- El puerto serie se puede configurar como RS232/RS485 y además incorpora los dos protocolos más utilizados como son el Modbus ASCII/RTU maestro o esclavo y la Cadena de caracteres (ASCII).
- El puerto maestro CANopen admite hasta 32 esclavos y puede configurarse entre 125 Kbps y 1 Mbps. La arquitectura basada en CANopen puede utilizarse para la comunicación con diferentes dispositivos tales como variadores de velocidad o servodrivives, así como para distribuir módulos de E/S lo más próximos posibles a los sensores y accionadores.

El configurador de CANopen está integrado en el software SoMachine y éste también puede usarse para importar archivos de descripción estándar en formato EDS.

2.3- DESCRIPCIÓN DE LA BASE COMPACTA 'TM258LF42DT4L'

La empresa Schneider-Electric dispone de seis modelos de controladores lógicos Modicon M258 para abarcar los diferentes requisitos de control de las aplicaciones. En nuestro caso, disponemos del controlador '**TM258LF42DT4L**' cuyos módulos de E/S se programan mediante el software SoMachine.

Las principales funciones integradas de las que dispone dicho controlador son:

- 26 entradas digitales de 24V, incluidas 8 entradas de conteo rápido (200 kHz)
- 4 entradas analógicas de $\pm 10V$, 4-20 mA/0-20 mA y resolución 12 bits
- 16 salidas digitales de transistor (0.5 A) incluidas 4 salidas reflejas

Así mismo, al igual que el resto de controladores de la gama, también dispone de:

- Cuatro entradas rápidas PNP, 2 entradas estándar y 2 salidas rápidas PNP dedicadas a funciones de HSC o PWM
- Una entrada rápida utilizable como entrada de captura de un encoder
- Dos comunes para las entradas y un común para las salidas
- Una fuente de alimentación de 24V que consta de tres unidades: una para la CPU, otra para los módulos de E/S rápidas y la última para otros módulos

Todos los componentes mecánicos de este sistema han sido diseñados para ser ensamblados mediante enclavamiento mecánico. El cableado y mantenimiento del dispositivo es muy simple ya que posee terminales de resorte extraíbles los cuales, se sueltan simplemente presionando una pequeña pestaña de bloqueo. Una gran ventaja es su accesibilidad ya que todos los conectores se encuentran en la parte frontal del controlador.

El controlador lógico incluye, como puede observarse en la [Fig. 4]:

1. Un bloque de visualización con:
 - 4 LEDs de estado del controlador: RUN/MS, BATTERY, APP0 y APP1
 - 6 LEDs de estado de los puertos de comunicaciones integrados: Eth LA, Eth ST, Eth NS, USB Host, MBS COM y CAN 0 STS
2. Pestaña de bloqueo para montaje o desmontaje
3. Dos ranuras PCI libres para los módulos de comunicación adicionales
4. Un módulo de fuente de alimentación de 24V
5. Módulos de E/S cada uno de ellos con un bornero extraíble con pestaña de bloqueo, un bloque de visualización que muestra el estado de las E/S y una ranura para un portaetiquetas
6. Bornero extraíble con pestaña de bloqueo
7. En el lateral, una conexión de extensión de bus para la conexión con el siguiente módulo
8. Una ranura para la batería del reloj de tiempo real

9. Un conector USB-A, marcado como Host, para conexión de un dispositivo de memoria USB para transferencia de programas, datos o actualizaciones de firmware
10. Un miniconector USB-B, marcado como PGR-Port, para la conexión al PC de programación
11. Un conector RJ45, marcado como Ethernet, para la conexión a la red Ethernet
12. Un conector RJ45, marcado como MBS, para el puerto serie
13. Un conector SUB-D macho de 9 contactos, marcado como CAN0, para conexión al bus CANopen

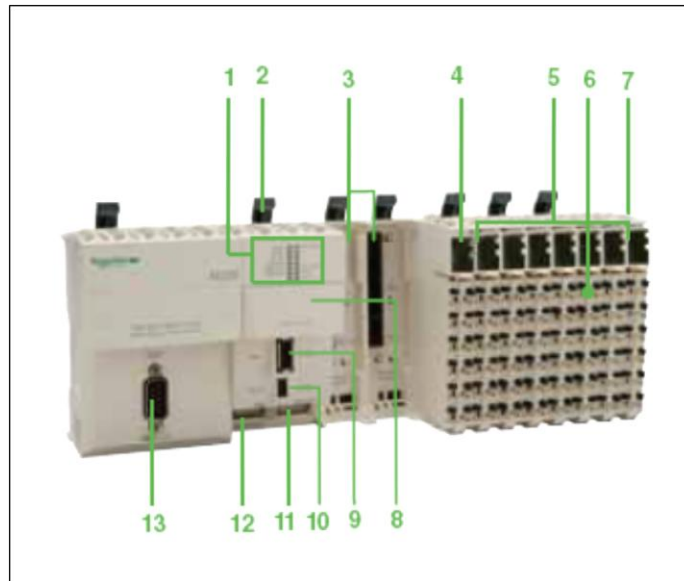


Fig. 4: Base compacta TM258LF42DT4L

2.4- COMUNICACIONES. PUERTO DE BUS CANOPEN

2.4.1- Introducción:

El controlador lógico M258 de Schneider-Electric dispone de un puerto maestro de bus CANopen el cual presenta grandes ventajas en el mundo de la automatización debido a su gran riqueza de funciones. CANopen es un protocolo abierto que cumple las normas EN 50325-4 e ISO 15745-2.

El bus CANopen es un bus multimaestro que garantiza un acceso fiable a los datos de tiempo real de los equipos de sistemas de control. El protocolo CSMA/CA (*Carrier Sense, Multiple Access, Collision Avoidance*) permite que múltiples estaciones utilicen un mismo medio de transmisión a fin de garantizar un uso óptimo del ancho de banda. Este proceso consiste básicamente en que el equipo emisor antes de transmitir datos tiene que asegurarse que la red esté libre para poder realizarlo.

La construcción de sistemas basados en CAN requiere una capa de aplicación y unos perfiles que estandaricen la comunicación en el sistema, la funcionalidad de los dispositivos y la administración del sistema:

- Capa de aplicación (*application layer*): proporciona un conjunto de servicios y protocolos para los dispositivos de la red
- Perfil de comunicación (*communication profile*): define cómo configurar los dispositivos y datos, y la forma de intercambiar entre ellos
- Perfiles de dispositivos (*device profiles*): añade funcionalidad específica a los dispositivos

2.4.2- Protocolo CANopen:

A base de introducción, el protocolo que define el bus CAN se ajusta a la especificación OSI (*open system interconnection*).

El protocolo CANopen se basa en la capa de aplicación CAL de las redes CAN, anteriormente mencionada, la cual, fue una de las primeras especificaciones producidas por CiA (CAN in Automation). Ésta ofrece un ambiente orientado a objetos para el desarrollo de aplicaciones distribuidas de cualquier tipo basadas en CAN.

2.4.2.1- *Características Capa física:*

El medio físico es una línea de bus de dos hilos con un retorno común que es terminada en ambos extremos por resistores que representan la impedancia característica de la línea. Se constituyen así los niveles de tensión de los conductores *CAN_High* y *CAN_Low*. [Fig. 5]

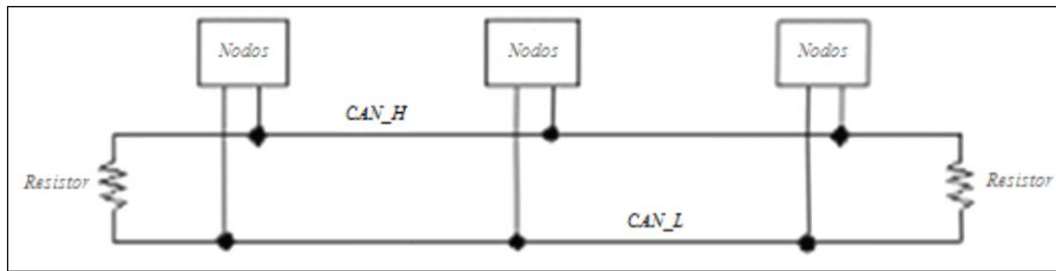


Fig. 5: Línea CAN

Los cables de bus pueden ser conductores en paralelo, trenzados y/o blindados según los requerimientos de compatibilidad electromagnética. Una estructura con una sola línea reduce las reflexiones.

Los segmentos de cable para la conexión de los nodos del bus deben ser tan cortos como sea posible, especialmente en tasas de alta velocidad de transmisión. Además, cada extremo de dicho bus debe incorporar un final de línea.

Debido a su limitación de longitud máxima, pueden utilizarse dispositivos puente o repetidores para aumentar el número de los nodos del bus que pueden ser conectados.

2.4.2.2- Características perfiles CANopen:

El bus CANopen es un conjunto de perfiles en sistemas CAN que presenta las siguientes características:

- Es un sistema de bus abierto
- Permite realizar intercambios de datos en tiempo real sin sobrecargar el protocolo
- Posee un diseño modular que permite modificar el tamaño
- Permite la interconexión y la posibilidad de cambiar los dispositivos.
- Se trata de una configuración normalizada de redes
- Permite el acceso a todos los parámetros de los dispositivos
- Permite la sincronización y circulación de los datos a partir de procesos cíclicos o controlados por eventos lo que se traduce en un breve tiempo de respuesta del sistema

2.4.2.3- Perfiles de comunicaciones:

El modelo de comunicación de CANopen define cuatro tipos de objetos de comunicación que son los siguientes:

- *Objetos administrativos*: se trata de mensajes administrativos que permiten la configuración de las distintas capas de la red así como la inicialización, configuración y supervisión de la misma. Se basa en los servicios NMT, LMS y DBT de la capa CAL.

- Mensajes predefinidos de emergencia y sincronización y time stamp: permiten la sincronización de los dispositivos y generar notificaciones opcionales de emergencia
- Service Data Objects (SDO): son objetos o mensajes de servicio utilizados para leer y escribir cualquiera de las entradas del diccionario de objetos de un dispositivo. De este modo, corresponden a mensajes CAN de baja prioridad.
- Process Data Objects (PDO): son objetos o mensajes de proceso utilizados para el intercambio de datos de proceso es decir, datos en tiempo real. Corresponden por esto a mensajes CAN de alta prioridad

CANopen soporta los modelos de comunicación punto-a-punto, maestro-esclavo y productor-consumidor.

2.4.3- Características puerto CANopen del controlador M258:

El controlador lógico M258 incluye un puerto CANopen macho SUB-D de 9 contactos y actúa como CANopen maestro. El bus consta de un maestro, del controlador lógico M258 y de dispositivos esclavos. El maestro se encarga de la configuración, el intercambio y el diagnóstico de dichos esclavos.

Dicho bus utiliza un par trenzado apantallado doble al que se conecta, con el controlador lógico M258, un máximo de 32 dispositivos esclavos por conexión en cadena o mediante derivación de conexiones.

La velocidad de datos, para el controlador lógico M258, es variable entre 10 Kbps y 1 Mbps y depende de la longitud del bus como puede observarse en la siguiente tabla [Fig. 6]:

Longitud máxima(m)	20	40	100	250	500	1000	2500	5000
Velocidad de datos (Kbps)	1000	800	500	250	125	50	20	10

Fig. 6: Tabla Relación de Longitud y Velocidad de CANopen

2.5- CONFIGURACIÓN DEL SOFTWARE: SOMACHINE

2.5.1- Introducción:

Es el software utilizado para la configuración y programación de todos los controladores y equipos M258. Es un software de programación que permite configurar, desarrollar y depurar una máquina entera desde un solo software incluyendo la lógica, el control de motores, la interfaz de usuario HMI y las funciones de automatización de red relacionadas.

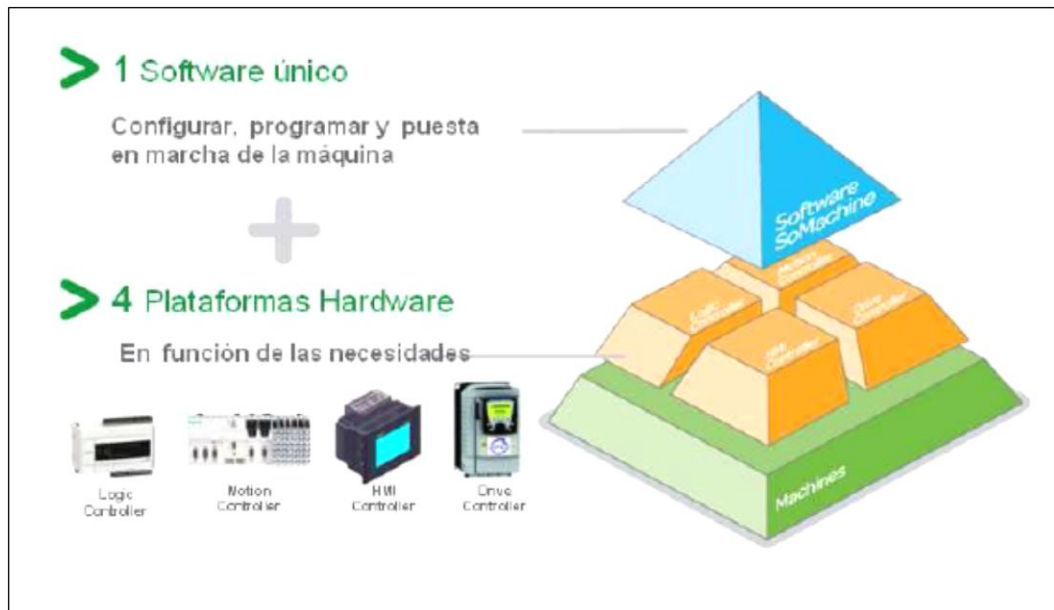


Fig. 7: Pirámide Explicativa Software Somachine

2.5.2- Características de SoMachine:

El software SoMachine incluye los seis lenguajes de la Comisión Electrotécnica Internacional (IEC) y en función de los requisitos, la aplicación puede utilizar una combinación de estos lenguajes. Dichos lenguajes son:

- Lenguaje Lista de Instrucciones (IL)
- Lenguaje de Contactos (LD)
- Lenguaje Grafset (SFC)
- Gráfico de Función Continua (CFC)
- Lenguaje de Texto Estructurado (ST)
- Diagrama de Bloques de Funciones (FBD)

En lo referente a servicios de programación, el software ya mencionado dispone de una gran variedad de posibilidades entre las que pueden destacarse:

- La posibilidad de creación por parte del usuario de estructuras de datos, funciones y bloques de función
- La monitorización o seguimiento gráfico de las variables así como la disponibilidad de ventanas de visualización

- La posibilidad de realizar cambios online, simular las aplicaciones, crear puntos de interrupción y la posible ejecución de paso a paso de dichas aplicaciones.

Si atendemos a la interfaz usuario-máquina (HMI), caben destacar los siguientes servicios disponibles en SoMachine:

- Librerías gráficas que poseen más de 4.000 objetos en 2D y 3D.
- Objetos preconfigurados como pueden ser botones, conmutadores o gráficos de barras entre otros.
- Objetos simples de dibujo tales que líneas, rectángulos, elipses...
- Alarmas y tablas de acción
- Javascripts y compatibilidad con archivos multimedia

2.5.3- Interfaz de SoMachine:

El software SoMachine dispone de una interfaz gráfica visual muy intuitiva y visual. La interfaz de usuario propone las tareas que deben ser realizadas durante todo el ciclo de desarrollo del proyecto a ya que habilita aquellas opciones o herramientas que pueden utilizarse en cada paso del proceso.

2.5.3.1- *Pantalla de selección principal:*

La pantalla de selección principal es aquella que aparece una vez iniciado el software SoMachine y presenta el siguiente aspecto [Fig. 8]:

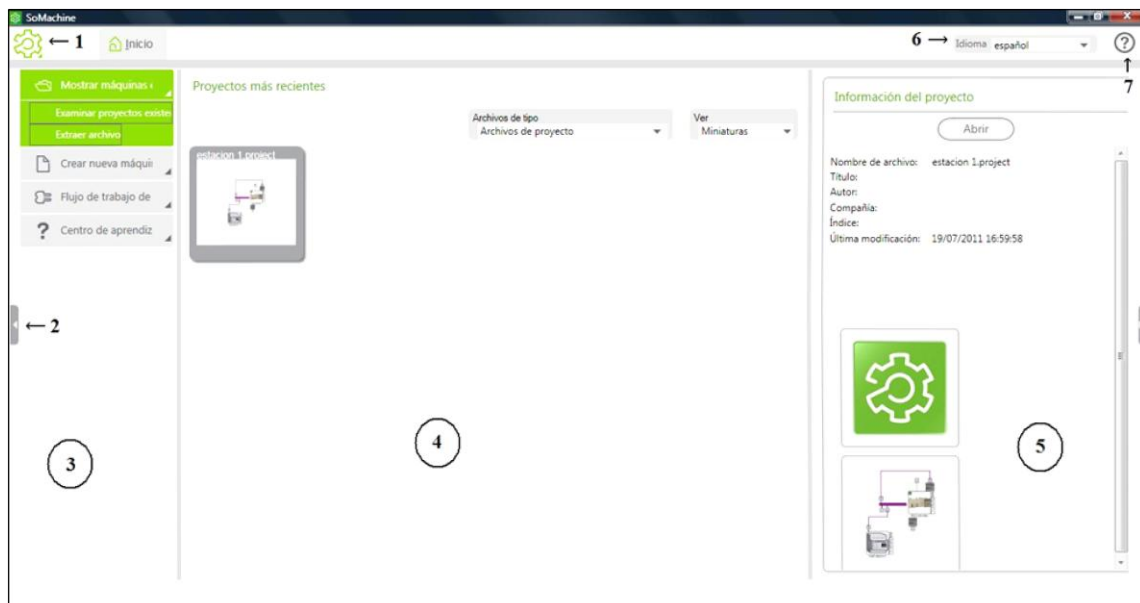


Fig. 8: Pantalla de Selección Principal

En dicha pantalla, pueden distinguirse las siguientes partes:

1. Menú de opciones generales
2. Pestaña para visualizar/ocultar ventana
3. Panel de selección de tareas
4. Área de trabajo

5. Panel de información
6. Menú de selección de idioma
7. Ayuda Online

En el panel de selección de tareas encontramos las diferentes tareas agrupadas en cuatro grupos:

- Mostrar máquinas existentes: esta carpeta muestra las opciones para abrir un proyecto ya existente. [Fig. 9]

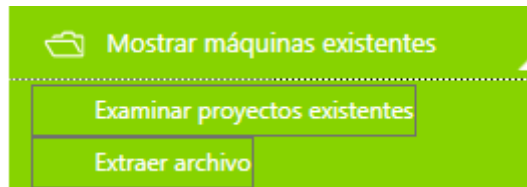


Fig. 9: Menú Mostrar Máquinas Existentes

- Crear nueva máquina: esta carpeta muestra las diferentes opciones para crear un proyecto nuevo, ya sea empezando de cero o a partir de una plantilla. [Fig. 10]

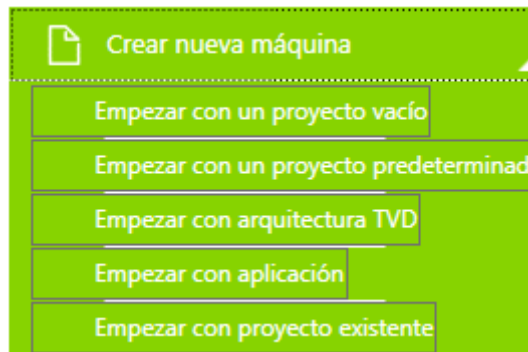


Fig. 10: Menú Crear Nueva Máquina

- Flujo de trabajo de la máquina: esta carpeta proporciona comandos dedicados a la puesta en funcionamiento de una máquina. [Fig. 11]

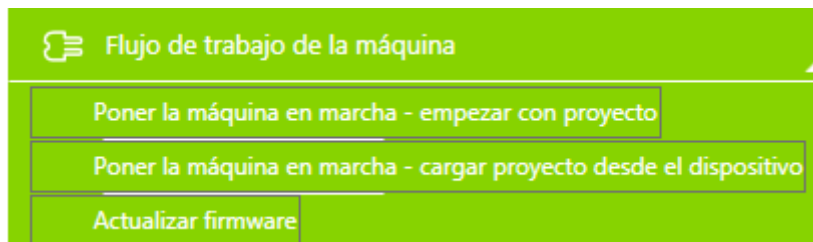


Fig. 11: Menú Flujo de Trabajo

- Centro de aprendizaje: en esta carpeta se proporciona más información sobre el software. [Fig. 12]

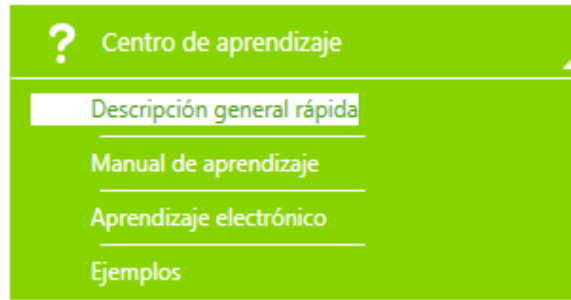


Fig. 12: Menú Centro de Aprendizaje

2.5.3.2- Introducción a la barra de tareas:

La barra de tareas consta de una sola pestaña para cada etapa del ciclo de un proyecto de ahí que la interfaz sea más visual e intuitiva. La barra de tareas no aparece hasta que no hemos creado un proyecto nuevo.

La barra de tareas consta de las siguientes pestañas:

- Propiedades: en esta pestaña aparecen la información del archivo y del autor del proyecto.
- Configuración: aquí es donde se configura la arquitectura de la máquina, como mínimo el controlador a utilizar. Es un proceso muy gráfico por lo que se hace muy fácil incluir módulos de ampliación, buses de comunicaciones o HMI.
- Programa: ventana dedicada a la programación del proyecto.
- Puesta en marcha: permite el acceso online al controlador y la monitorización del estado de todos los equipos conectados. Así mismo, permite descargar la aplicación o el firmware.
- Informe: muestra una vista preliminar del informe del proyecto así como la posibilidad de impresión del mismo.

En la siguiente imagen se puede observar la barra de tareas y como puede verse es bastante visual e intuitiva [Fig. 13]:

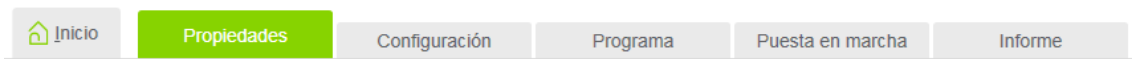


Fig. 13: Barra de Tareas

2.5.3.3- Pantalla de Propiedades:

Permite al usuario escribir información adicional del proyecto, la cual es opcional. Esta información siempre se mostrará en el “*área de información del proyecto*” que aparece en la parte derecha de la pantalla y permite identificar los proyectos sin necesidad de abrirlos.

En el área de selección que aparece a la izquierda de la pantalla, se tienen las siguientes opciones:

- **General:** permite cambiar el nombre del archivo del proyecto y guardarlo en otra ubicación. Además, nos muestra la información del archivo y del autor.
- **Descripción:** permite al usuario añadir una imagen al proyecto que lo hará más visual.
- **Información personalizada:** permite añadir información que el usuario pueda considerar relevante además de la posibilidad de agregar archivos adjuntos.

En la siguiente imagen puede observarse la ya mencionada pantalla de propiedades en su pestaña 'General' [Fig. 14]:

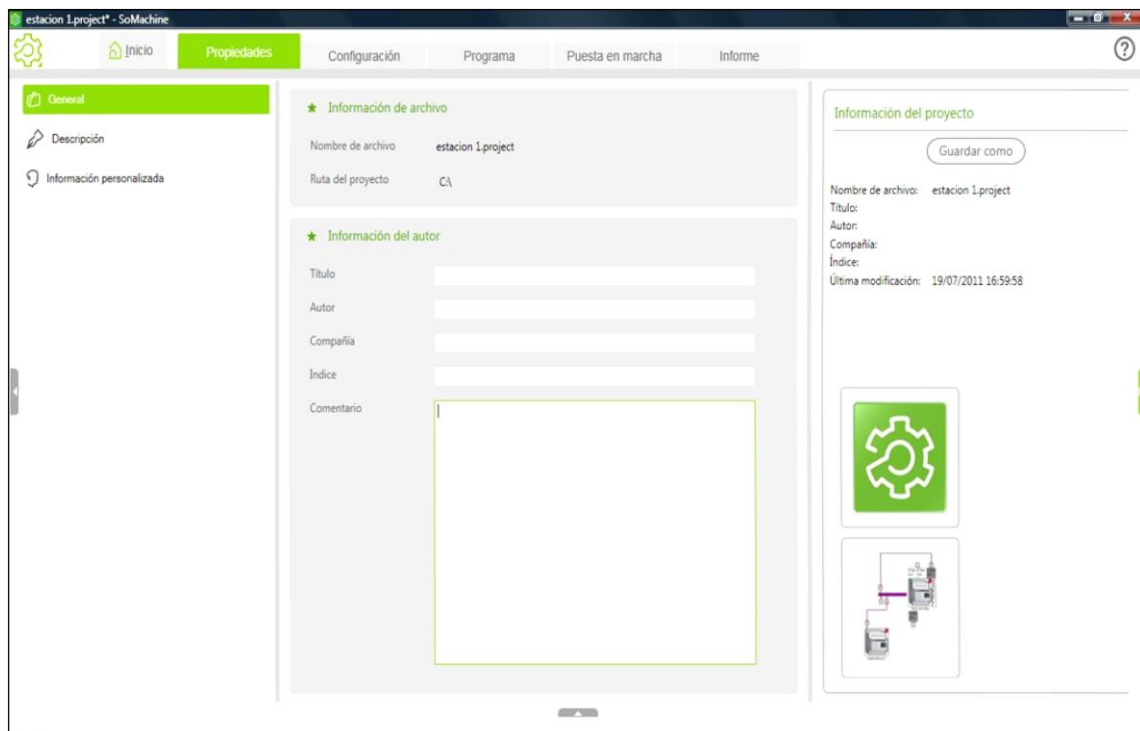


Fig. 14: Pantalla de Propiedades

2.5.3.4- Pantalla de Configuración:

Esta ventana le permite al usuario poder realizar, de una manera sencilla y gráfica, la configuración hardware de la máquina. Ésta sólo aparece una vez que el proyecto haya sido creado.

En la barra de tareas de dicha ventana, se encuentran todos los equipos que pueden añadirse a la configuración agrupados en diferentes carpetas. Además, dispone de un buscador para encontrar más rápidamente el equipo de configuración deseado introduciendo el nombre o la referencia de éste.

En la parte de debajo del área de trabajo el programa dispone de una barra de visualización para facilitar el uso de la configuración de hardware mediante el editor gráfico.

Para añadir un equipo al proyecto basta con, una vez buscado, seleccionarlo, arrastrarlo hasta el área de trabajo y soltarlo en éste. Una vez añadido, aparecerá su representación gráfica en dicho área.

Para añadir un módulo de expansión a un equipo basta con hacer clic en el campo 'Agregar ampliación' que hay en la imagen del equipo en el área de trabajo. Aparecerá una ventana flotante donde aparecerán todos los módulos de expansión que pueden añadirse a ese equipo. Habrá que seleccionar el deseado y pulsar el botón 'Agregar y cerrar'.

En la imagen siguiente [Fig. 15], se puede observar la ventana de configuración y en ella podemos ver desde donde pueden añadirse los equipos y sus ampliaciones:

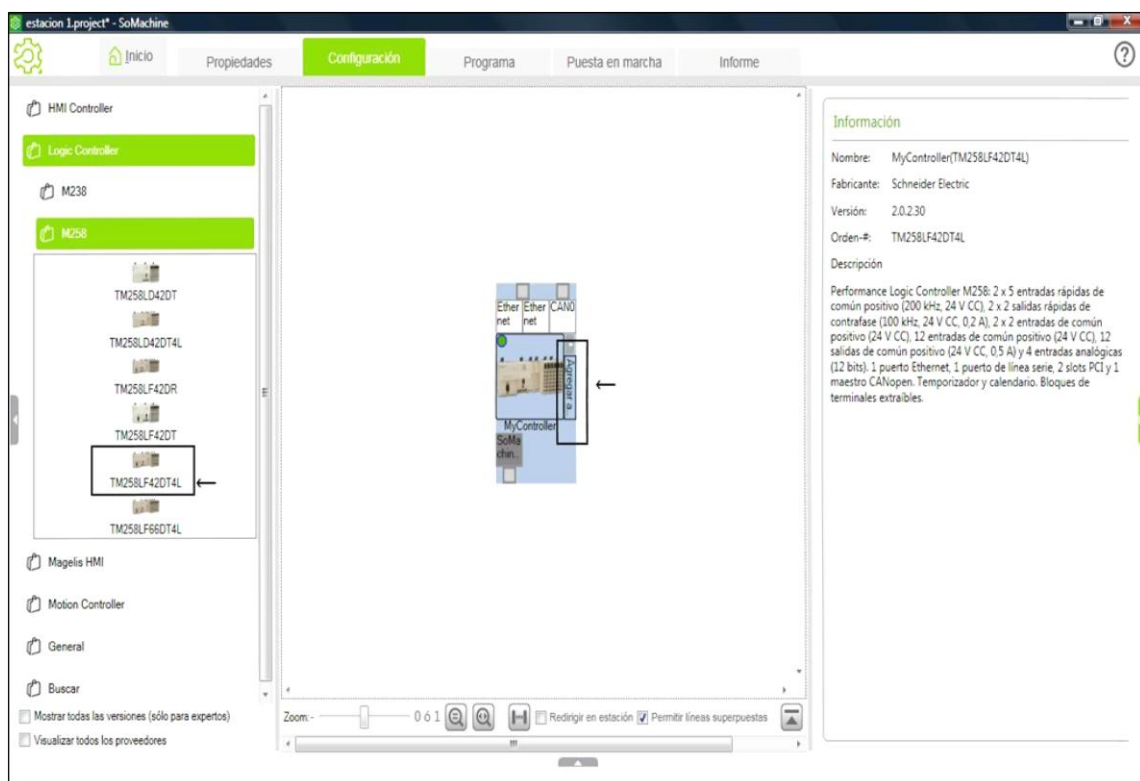


Fig. 15: Pantalla de Configuración

2.5.3.5- Pantalla de Programa:

La pantalla de programa nos permite acceder a un entorno de programación basado en CoDeSys, un sistema de desarrollo de controladores. Está formada por diferentes menús y ventanas [Fig. 16]:

1. Menú principal donde se encuentran las principales opciones del programa
2. Ventana Dispositivos
3. Ventana Mensajes
4. Área de trabajo
5. Ventana Propiedades

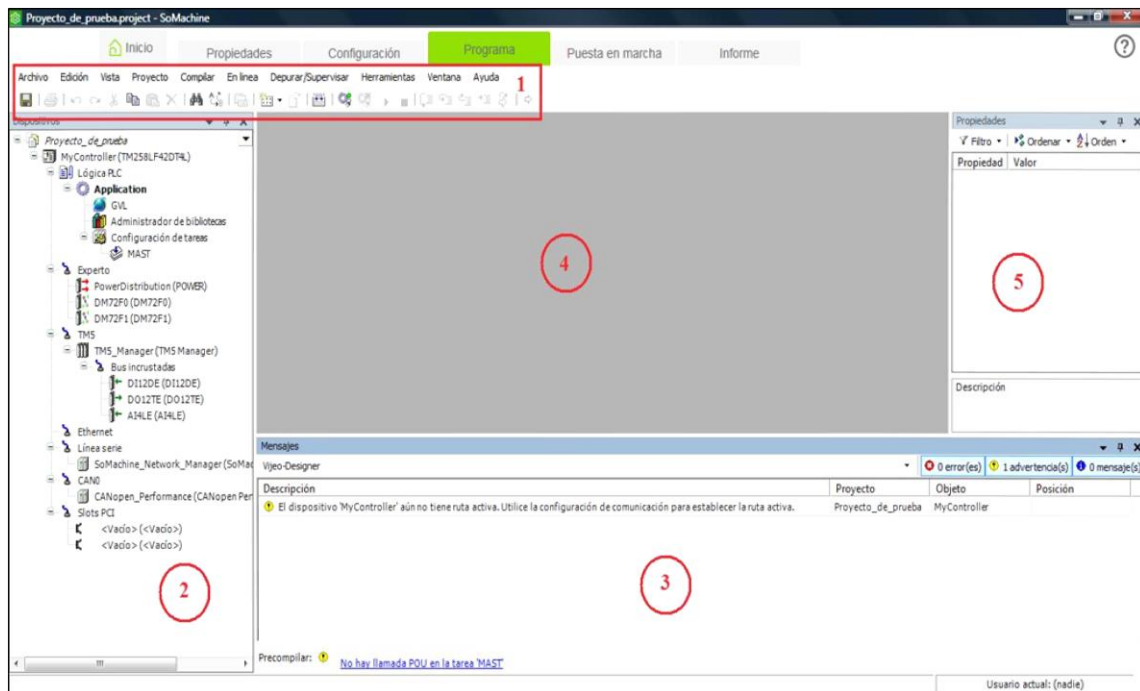


Fig. 16: Pantalla de Programa

La ventana principal de la pantalla de programa es la de 'Dispositivos'. En ella aparecen todos los parámetros de configuración que han sido anteriormente definidos y se pueden modificar, eliminar o incluso añadir nuevos parámetros en esta ventana de ahí su importancia. Todos los cambios o modificaciones realizadas en esta ventana, serán modificados automáticamente en la ventana de 'Configuración'.

La ventana de 'Dispositivos' está formada básicamente por cuatro nodos aunque dependiendo del controlador utilizado, esta disposición puede cambiar. Esos cuatro nodos principales son:

- **Controlador:** está compuesto por el tipo y la configuración del PLC.
- **El área de programa:** donde se muestran todas las partes del programa.
- **Funciones integradas,** propias del controlador, ya sean E/S, entradas HSC de conteo rápido o salidas de tren de pulsos PTO y PWM.
- **Comunicación:** compuesto por la configuración de los puertos de comunicación con dispositivos externos.

Para realizar cualquier modificación, basta con hacer doble click en cualquiera de las opciones de la ventana dispositivos apareciendo una ventana emergente en el área central de programación. A continuación se muestra un ejemplo de ventana emergente al hacer doble click sobre la tarea MAST [Fig. 17]:

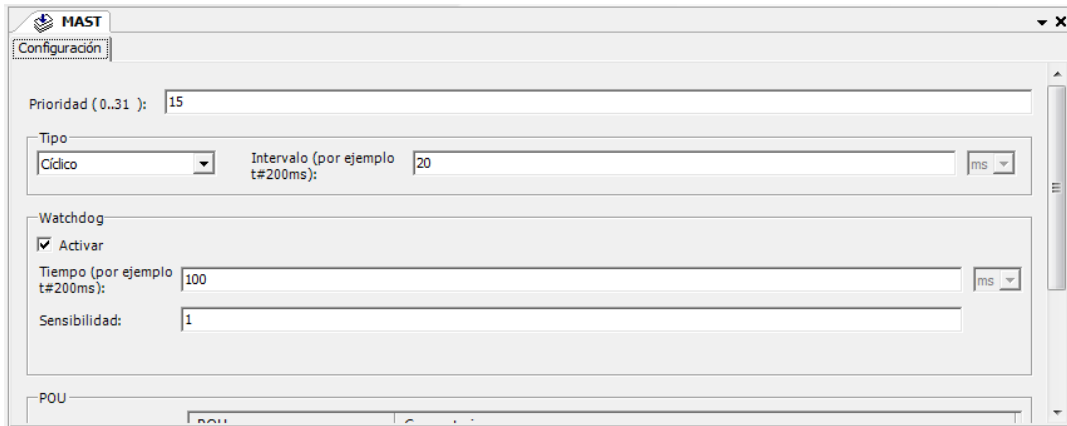


Fig. 17: Ventana Tarea MAST

En el caso del controlador lógico M258, la ventana de dispositivos tiene el siguiente aspecto [Fig. 18]:

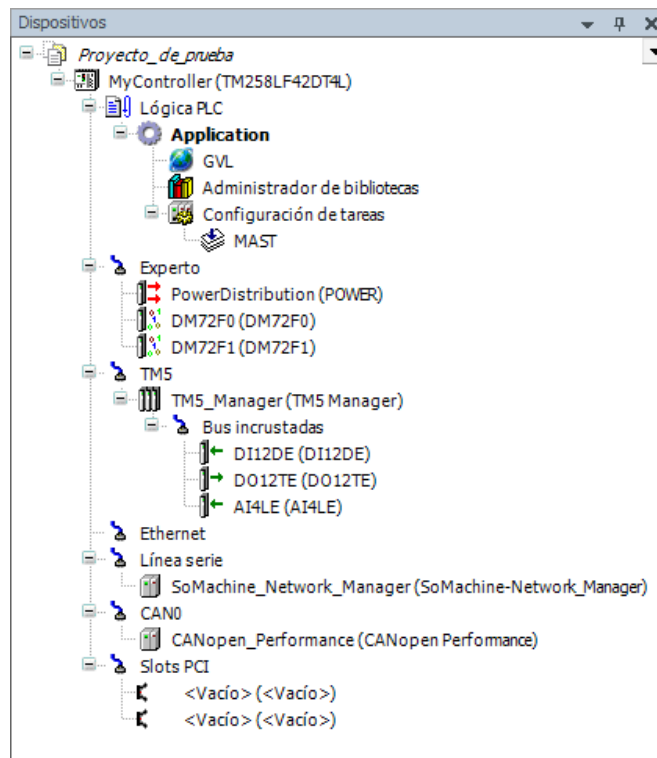


Fig. 18: Ventana Dispositivos

La ventana 'Mensajes' también tiene una gran importancia ya que en ella se muestran mensajes sobre actualizaciones, errores al compilar o advertencias de la programación dándonos una pequeña descripción del posible problema. En el ejemplo siguiente [Fig. 19], puede observarse una advertencia así como la descripción de la misma y en qué parte del proyecto tiene lugar:

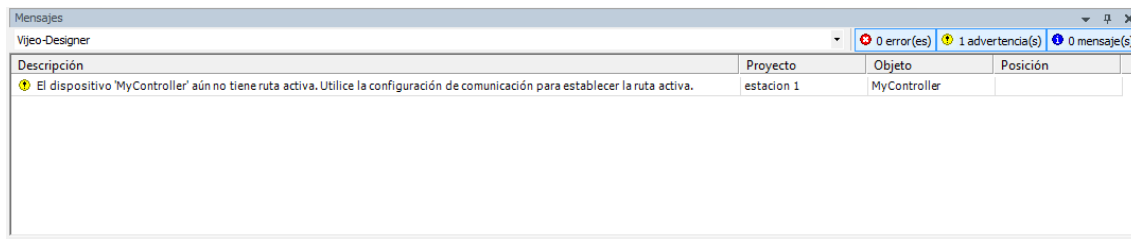


Fig. 19: Ventana Mensaje con Advertencia

La ventana 'Propiedades' es una ventana de menor importancia que las anteriores pero que también nos aporta información al proyecto en la programación de estados, transiciones y variables. [Fig. 20]

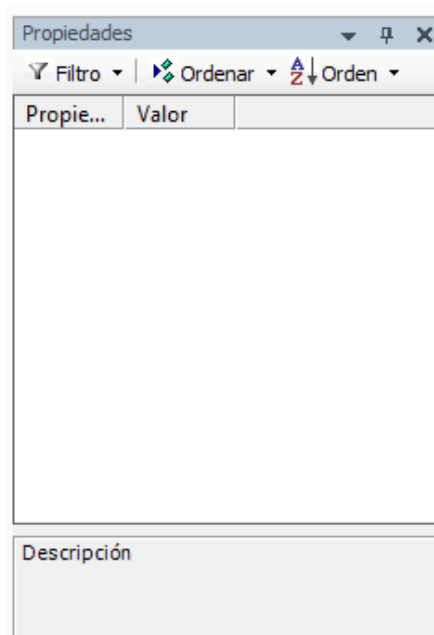


Fig. 20: Ventana Propiedades

En definitiva, la ventana de Programa es la más importante dentro del software a la hora de programar ya que la mayor parte de carga de trabajo se realiza desde aquí pudiéndose incluso modificar la configuración de los equipos conectados al controlador utilizado.

2.5.3.6- Pantalla de Puesta en marcha:

En la ventana de 'Puesta en marcha' se puede tener acceso online al controlador elegido así como monitorizar el estado de todos los equipos conectados a dicho controlador. Además, permite descargar la aplicación o el firmware.

La pantalla se divide en cuatro zonas diferenciadas [Fig. 21]:

1. Zona de funciones de configuración en línea compuesta por:
 - Iniciar/Cerrar sesión
 - Descarga múltiple
 - Descarga de código de origen

2. Indica gráficamente el estado del controlador y los equipos
3. Presenta información del controlador y los equipos conectados
4. Descripción, donde aparecen posibles fallos de la aplicación

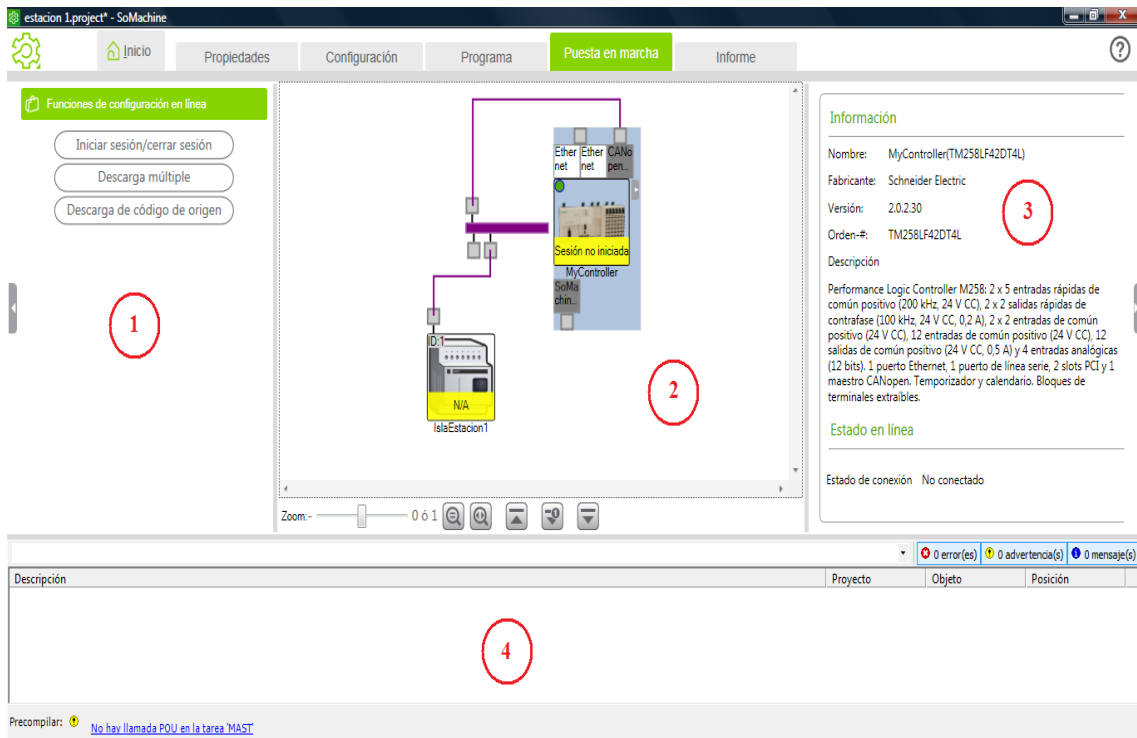


Fig. 21: Pantalla de Puesta en Marcha

2.5.3.7- Pantalla de Informe:

La pantalla de informe es una ventana simple donde aparecen diferentes opciones para la previa configuración de las páginas del informe, su vista previa y su posterior impresión.

2.6- CONFIGURACIÓN ISLAS ADVANTYS

2.6.1- Introducción:

Advantys STB es un sistema de E/S abierto, modular y distribuido, diseñado para la industria de la maquinaria con una ruta de migración hacia la industria de procesos. Se trata de un montaje de módulos de E/S distribuidas y alimentación, entre otros, que funcionan juntos como un nodo de isla en una red de bus de campo abierta.

Lo anteriormente denominado como isla es una estructura donde residen los módulos de alimentación (PDM) y un módulo de interface de red (NIM).

Este sistema, permite diseñar islas de automatismo gestionadas por un controlador maestro a través de un bus o una red de comunicaciones.

Las redes de buses de campo a las que se puede conectar una isla ADVANTYS son muy numerosas, demostrando la versatilidad de la que disfruta esta estructura:

- Profibus DP
- DeviceNet
- Ethernet
- CANopen
- Fipio
- Modbus Plus
- INTERBUS
- Segmento primario

Un NIM se encuentra en la primera posición del bus de la isla, el situado más a la izquierda en el montaje físico. Actúa como camino entre la isla y el bus de campo facilitando el intercambio de datos entre el maestro del bus de campo y los módulos de E/S de la isla. Es el único módulo de la isla que depende del bus de campo. Hay un tipo de módulo NIM diferente disponible para cada bus de campo.

El resto de los módulos de distribución de alimentación y de E/S del bus de la isla funciona exactamente igual, sin importar el bus de campo en el que se encuentre la isla. Tiene la ventaja de poder seleccionar los módulos de E/S para construir una isla independiente del bus de campo en el que funcionará.

2.6.2- Características y Composición:

El rendimiento de una isla está determinado por el tipo de NIM que utilice. Hay diferentes modelos de NIM de bus de campo con capacidades operativas dimensionables.

El conjunto básico de los módulos Advantys STB comprende:

- un conjunto de módulos de E/S analógicos, digitales y especiales
- NIM de bus de campo abiertos

-
- módulos de distribución de alimentación (PDM)
 - módulos de extensión del bus de isla
 - módulos especiales

Estos módulos básicos están diseñados para factores de formato Advantys STB específicos y encajan en unidades base del bus de isla. Aprovechan totalmente las capacidades de distribución de alimentación y de comunicación de isla y son autodireccionables.

En resumidas cuentas, un sistema Advantys STB comienza con un grupo de dispositivos interconectados denominados '*segmento primario*'. Este primer segmento es una pieza obligatoria de una isla.

Dependiendo de sus necesidades y del tipo de NIM que está utilizando, la isla puede ampliarse opcionalmente a segmentos adicionales de módulos Advantys STB, denominados '*segmentos de extensión*', y a dispositivos que no sean STB, tales como módulos preferidos o dispositivos CANopen estándar.

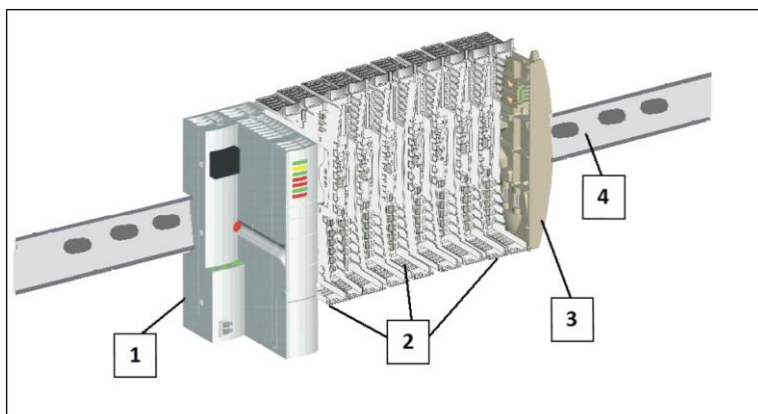
Una isla Advantys STB puede admitir dispositivos no integrados CANopen estándar. Estos dispositivos no son autodireccionables en el bus de isla. Por lo tanto, deben direccionarse manualmente, generalmente por medio de conmutadores físicos integrados en los dispositivos. Se configuran utilizando el software de configuración Advantys. No puede utilizarse un dispositivo CANopen estándar con un NIM básico y cuando se utilizan dichos dispositivos, deben instalarse al final de la isla.

Cada bus de la isla comienza con un segmento primario. Dicho segmento se compone del NIM de isla y de un conjunto de bases de módulos interconectadas unidas a un raíl DIN. Los PDM y los módulos de E/S Advantys STB se montan en estas bases sobre el raíl DIN. El NIM es siempre el primer módulo del segmento primario es decir, el que se encuentra más a la izquierda como ya habíamos visto anteriormente.

Las bases interconectadas sobre el raíl DIN forman la estructura del bus de la isla. El bus de la isla alberga y admite los buses de comunicaciones de isla. Un conjunto de contactos en los laterales de las unidades base proporcionan la estructura de bus para:

- alimentación lógica
- la alimentación de campo del sensor hacia los módulos de entrada
- la alimentación del impulsor hacia los módulos de salida
- la señal de autodireccionamiento
- las comunicaciones del bus de la isla entre la E/S y el NIM

Una de las principales características del NIM es que a diferencia de los PDM y de los módulos de E/S, se une directamente a un raíl DIN. [Fig. 22]



1. NIM
2. Bases de módulos
3. Bases de conexiones
4. Raíl DIN

Fig. 22: Estructura del Bus de Isla

2.6.3- **Módulo NIM STB NCO 2212:**

El módulo de isla NIM STB NCO 2212 es el disponible en el laboratorio de prácticas. [Fig. 24]

Las principales características de dicho módulo son:

Características		Función
1	Interface de bus de campo	Un conector SUB-D de 9 pines utilizado para conectar el NIM y el bus de isla a un bus de campo CANopen.
2	Conmutador rotativo superior	Los dos conmutadores rotativos se utilizan juntos para especificar el ID de nodo del NIM en un bus de campo CANopen y para establecer el valor de baudios del bus de campo en el NIM.
3	Conmutador rotativo inferior	
4	Interface de fuente de alimentación	Un conector de dos receptáculos para conectar una fuente de alimentación externa de 24 VCC al NIM.
5	Matriz de LED	Indicadores LED de color que utilizan varias secuencias para indicar visualmente el estado de funcionamiento del bus de la isla.
6	Tornillo de liberación	El mecanismo utilizado para extraer el NIM del segmento DIN.
7	Cajón de la tarjeta de memoria extraíble	Cajón de plástico en el que puede colocarse la tarjeta de memoria extraíble y luego insertarse en el NIM.
8	Tapa del puerto CFG	Una cubierta abatible del panel frontal del NIM que cubre la interface CFG.

Fig. 23: Tabla Módulo NIM STB NCO 2212

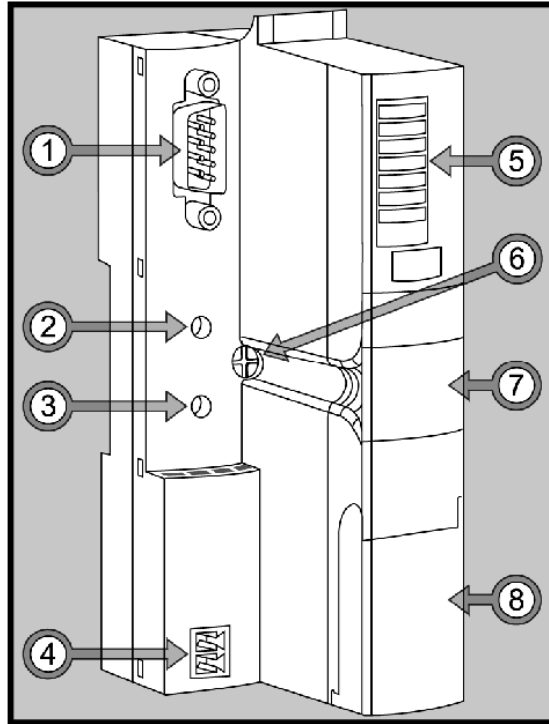


Fig. 24: Esquema Módulo NIM STB NCO 2212

Los conmutadores rotativos del NIM CANopen STB NCO 2212 se utilizan para establecer la dirección de nodo y los baudios de la isla Advantys STB.

Los dos conmutadores rotativos están ubicados en la parte frontal del NIM CANopen [Fig. 25], debajo del puerto de conexión del bus de campo. Cada conmutador tiene dieciséis posiciones.

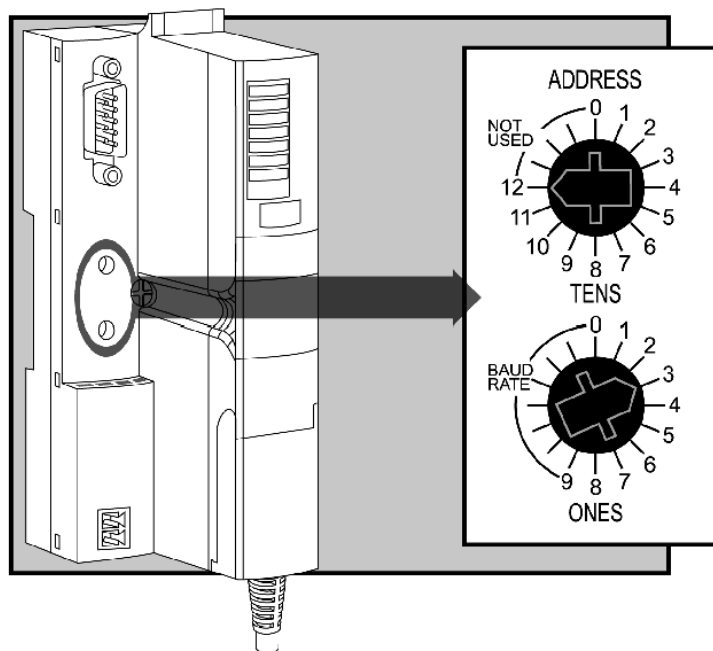


Fig. 25: Conmutadores Rotativos del NIM

2.6.4- Interface de bus de campo CANopen:

La interface de bus de campo situada en la parte frontal del módulo es el punto de conexión entre los módulos de E/S Advantys STB y la red CANopen. La interface es un conector SUB-D (DB-9P) de nueve pines. [Fig. 26]

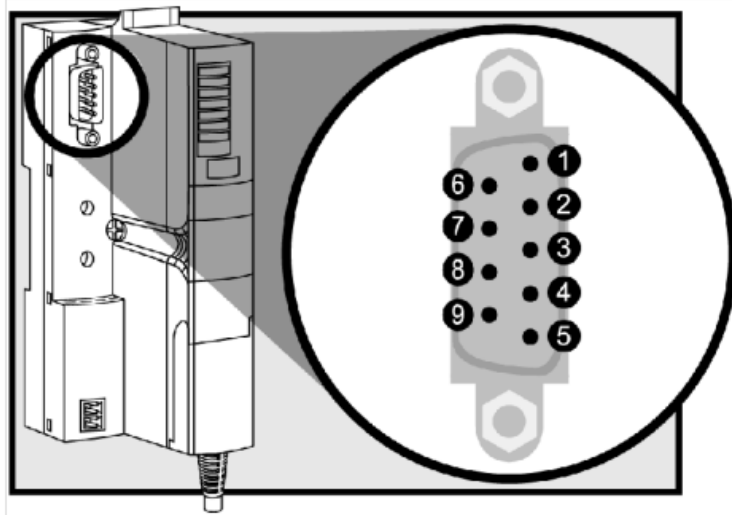


Fig. 26: Conector SUB-D de 9 Pines

La asignación de los pines de la figura anterior [Fig. 26] debe ser la siguiente [Fig. 27]:

Pin	Señal	Descripción
1	No utilizado	Reservado
2	CAN_L	Línea de bus bajo CAN
3	CAN_GND	Tierra CAN
4	No utilizado	Reservado
5	CAN_SHLD	Blindaje CAN opcional
6	GND	Tierra opcional
7	CAN_H	Línea de bus alto CAN
8	No utilizado	Reservado
9	No utilizado	Reservado

Fig. 27: Tabla Asignación Pines Conector SUB-D de 9 Pines

3- CURSO DE PROGRAMACIÓN Y CREACIÓN DE APLICACIONES

3.1- CREACIÓN DE APLICACIONES MEDIANTE SOMACHINE

Para la programación de cualquier aplicación mediante el software SoMachine, en primer lugar iremos al menú Inicio → Programas → Schneider Electric → SoMachine y lanzaremos el programa SoMachine haciendo doble click sobre el mismo.

Nos aparecerá la ventana de Inicio y en ella podremos seleccionar las diferentes opciones para la creación de un nuevo proyecto o la apertura de uno ya existente.

Para crear un nuevo proyecto pulsaremos sobre 'Crear nueva máquina' y nos aparecerá una lista con diferentes formas para iniciar nuestro proyecto. Elegiremos la opción 'Empezar con un proyecto vacío'. [Fig. 28]

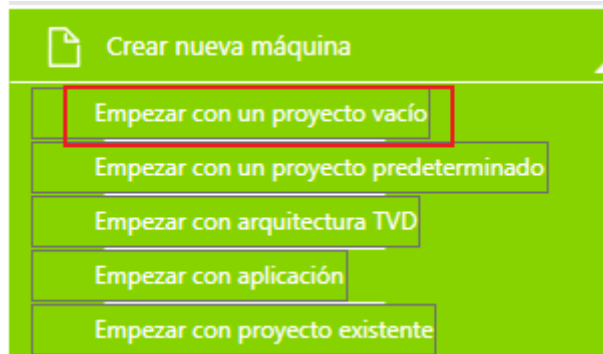


Fig. 28: Menú Creación Nuevo Proyecto

El siguiente paso será guardar el proyecto. Lo guardaremos con el nombre de 'Proyecto_de_prueba' y en tipo de archivo nos aseguraremos de que está seleccionado el tipo Project File (*.project). [Fig. 29]

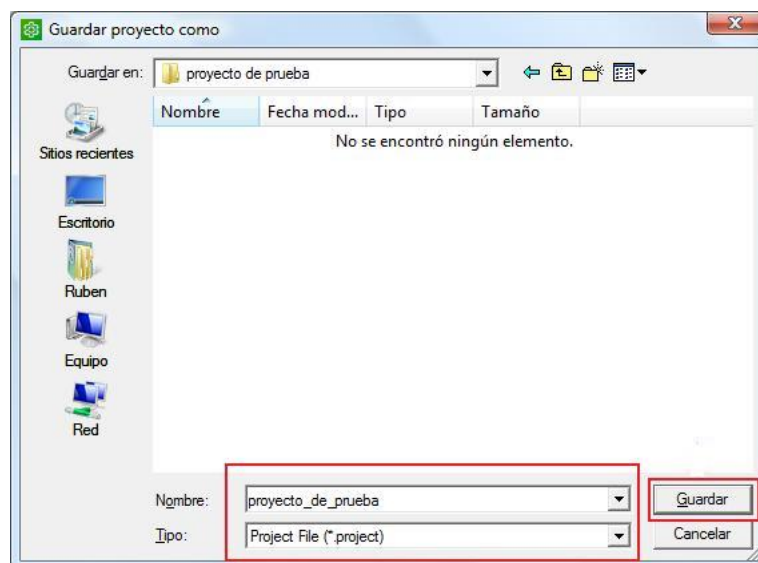


Fig. 29: Ventana Guardar Proyecto

Una vez guardado el nuevo proyecto, nos aparecerá la siguiente ventana [Fig. 30] en la cual, se encuentran los diferentes menús del programa.

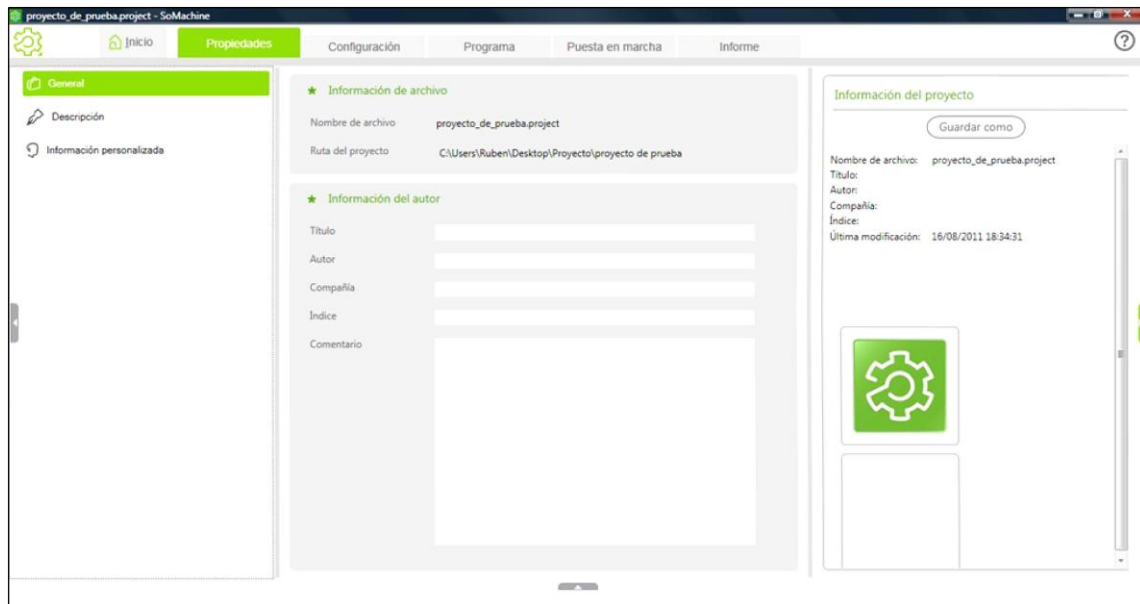


Fig. 30: Menús Proyecto

En primer lugar, antes de adentrarnos en la configuración propiamente dicha de la nueva aplicación, rellenaremos los campos de Información de autor [Fig. 31] que aparecen en la pestaña propiedades. Como Título del proyecto pondremos: 'Mi Primer Proyecto'. Rellenaremos el campo de autor con nuestro nombre y apellidos. Así mismo, escribiremos un pequeño comentario sobre la descripción de la aplicación que realizaremos posteriormente, en este caso: 'Proyecto de pruebas: Curso de Programación'.

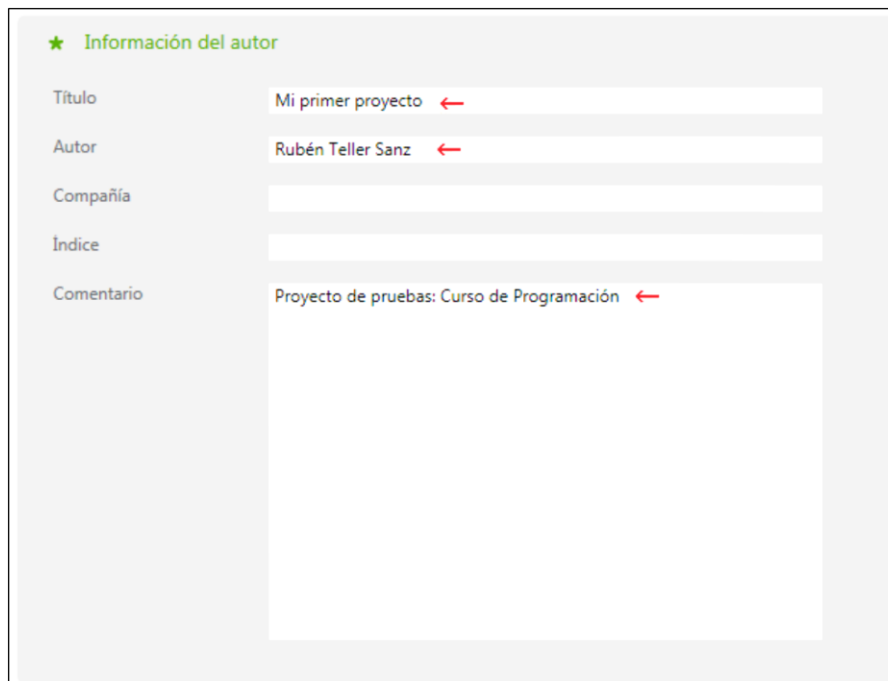


Fig. 31: Información del autor

Dicha información [Fig. 31], aunque no es relevante para la realización de las aplicaciones, nos ayudará cuando tengamos varias aplicaciones creadas para poder distinguirlas sin necesidad de una búsqueda profunda.

Una vez realizado esto, iremos a la pestaña de 'Configuración' [Fig. 32] donde crearemos la configuración de la arquitectura que va a tener la máquina de nuestra aplicación.

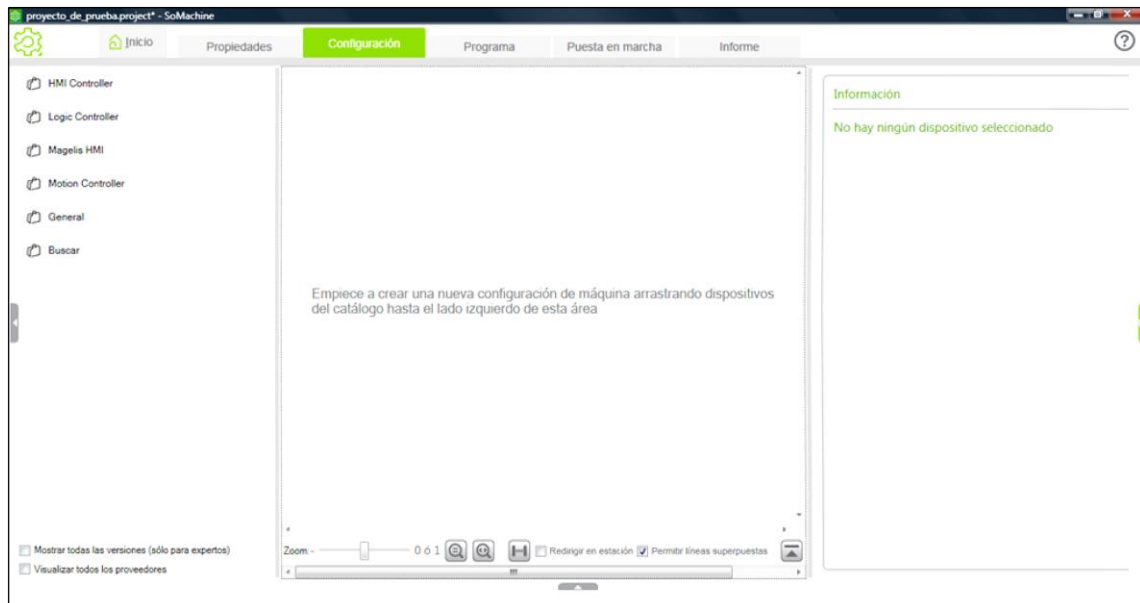


Fig. 32: Pantalla Configuración

Como puede observarse en la imagen [Fig. 32], la zona central se encuentra en blanco y aparece el siguiente mensaje: *'Empiece a crear una nueva configuración de máquina arrastrando dispositivos del catálogo hasta el lado izquierdo de esta área'*.

Para ello, bastará con buscar el controlador requerido para la aplicación en el catálogo que aparece a la izquierda de la pantalla y una vez encontrado, seleccionarlo y arrastrarlo hasta dicha área.

➤ Ejercicio 1: elección del controlador.

- Como primer ejercicio, empezaremos por buscar en la carpeta 'Logic Controller', el controlador TM258LF42DT4L que es del que disponemos en el laboratorio de prácticas. Una vez añadido dicho controlador [Fig. 33], haremos doble click sobre él para familiarizarnos con las propiedades de dicho autómeta.

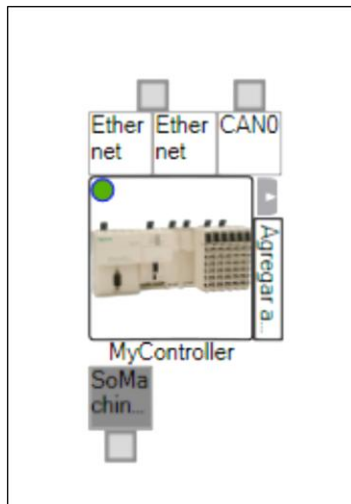


Fig. 33: Controlador TM258LF42DT4L

➤ Ejercicio 2: cómo añadir dispositivos al controlador elegido.

- Para añadir un dispositivo o ampliación al controlador elegido, basta con hacer click en la parte de la imagen en la que pone 'Agregar ampliación'. Nos aparecerá una nueva ventana [Fig. 34] con los dispositivos disponibles para agregar al autómata. A base de ejemplo, buscaremos el módulo de entradas analógicas TM5SAI4H, lo seleccionaremos y le daremos a 'Agregar y cerrar'.

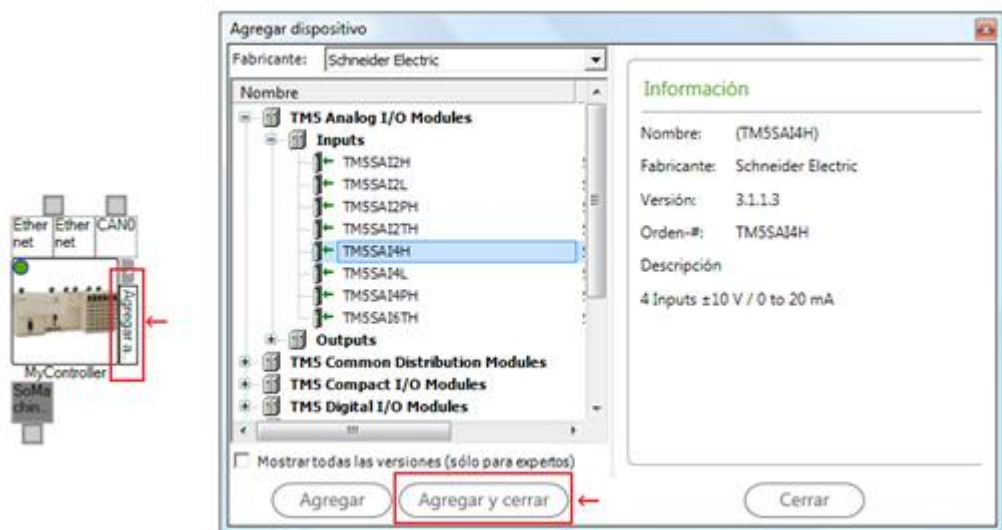


Fig. 34: Agregar Dispositivo al controlador

- Una vez añadido el módulo de entradas analógicas, podremos configurarlo haciendo doble click sobre él. La configuración del controlador y el módulo añadido presentará el siguiente aspecto [Fig. 35]:

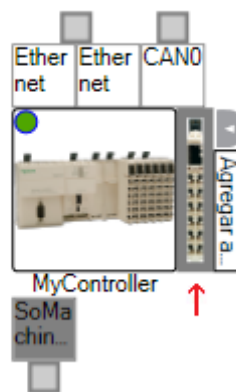


Fig. 35: Controlador con Módulo Añadido

- Para borrar dicho módulo basta con seleccionarlo, hacer click con el botón derecho y darle a eliminar dispositivo.

➤ Ejercicio 3: cómo añadir la configuración para conexión CanOpen

- Para incluir el maestro CanOpen a la configuración, basta con hacer un click sobre el recuadro de la configuración bajo el nombre CAN0. Aparecerá una nueva ventana para agregar el dispositivo. Buscaremos en buses de campo el *CanOpen Performance* de Schneider, lo seleccionaremos y le daremos a 'Agregar y cerrar' quedándonos algo de la siguiente forma:

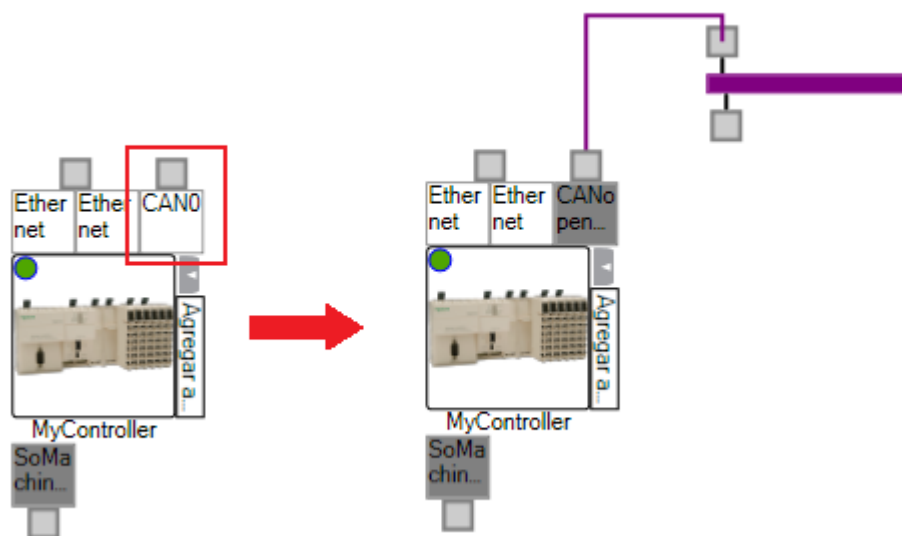


Fig. 36: Añadir Configuración CANopen

- Una vez realizado esto, ya tendremos el maestro CanOpen agregado a nuestra configuración y podremos comprobarlo en la ventana Dispositivos de la pestaña Programa de la barra de menú. [Fig. 18]

➤ **Ejercicio 4:** cómo añadir un dispositivo CanOpen

- Para añadir un dispositivo mediante conexión CanOpen, basta con hacer click sobre el puerto libre es decir, el recuadro que aparece marcado en la siguiente imagen. [Fig. 37]

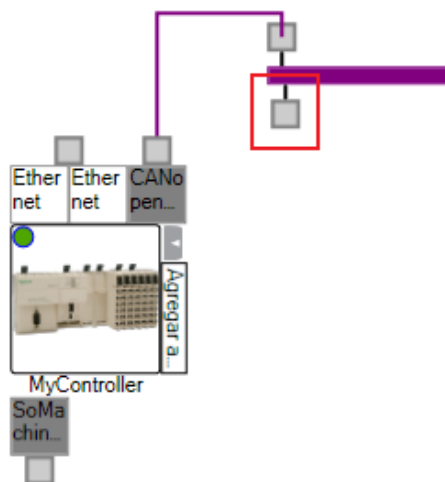


Fig. 37: Puerto Libre Conexión CANopen

- Nos aparecerá una ventana con un listado de dispositivos con conexión CanOpen para agregar. Por ejemplo, para ver cómo sería el proceso, seleccionaremos el variador de velocidad Altivar 312 y le daremos a 'Agregar y cerrar' quedándonos la configuración de la siguiente manera [Fig. 38]:

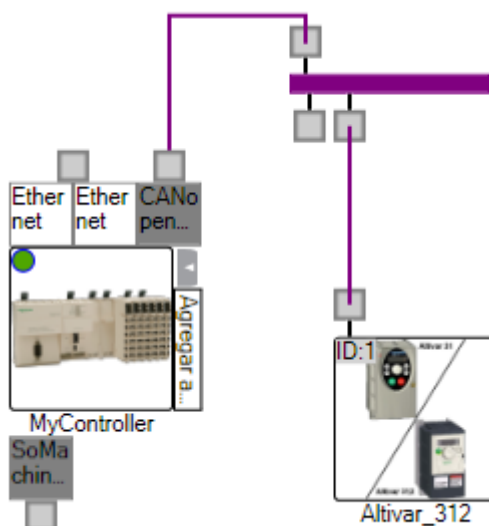


Fig. 38: Conexión Variador de Velocidad Altivar 312

Una vez realizados estos pequeños ejercicios sobre la elección del controlador y de los dispositivos adicionales, pasaremos a la pestaña 'Programa' del menú para comenzar con la creación de una aplicación en diferentes lenguajes de programación. [Fig. 39]

En primer lugar, hay que configurar el tipo de comunicación con el autómata por lo que, se tiene que establecer una ruta activa en el 'Gateway' para poder conectarse a éste. Para ello, haremos doble click sobre 'Mycontroller' en la ventana de dispositivos.

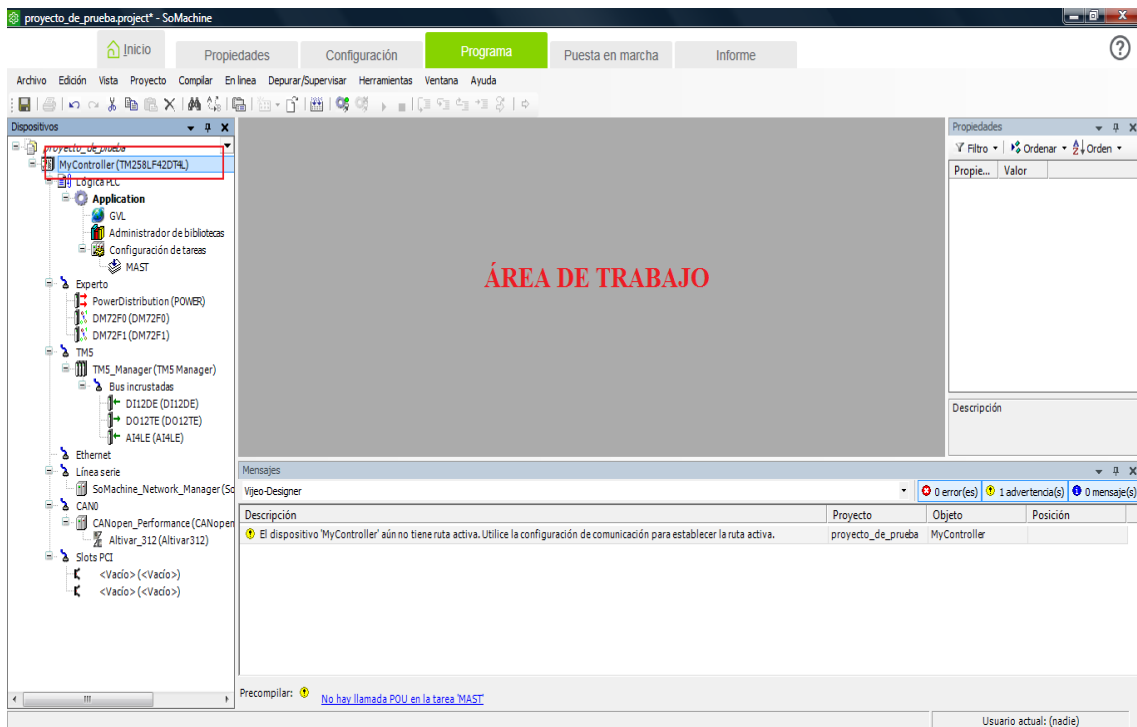


Fig. 39: Ventana Programa

En el área de trabajo nos aparecerá la siguiente ventana [Fig. 40] en la cual podremos establecer el Gateway. Para ello, en la pestaña de 'Configuración de comunicación', haremos click en 'Agregar puerta de enlace'.

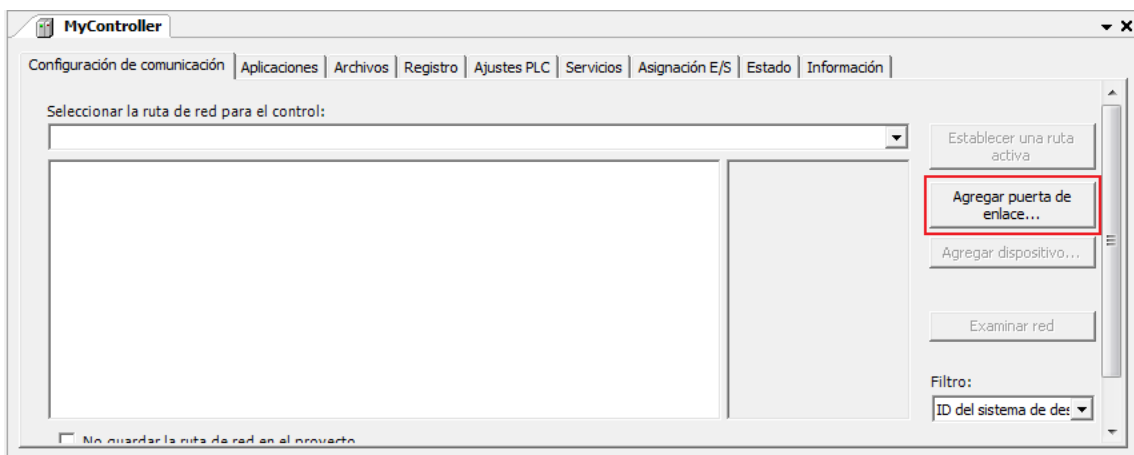


Fig. 40: Ventana Establecimiento del Gateway

Una vez realizado esto, nos aparecerá la ventana de la puerta de enlace en la que bastará simplemente con darle a aceptar. [Fig. 41]

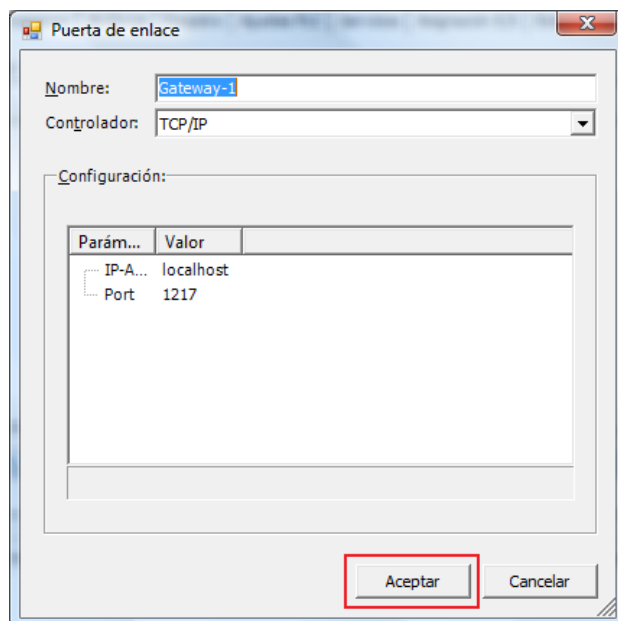


Fig. 41: Puerta de enlace

Al aceptar, nos aparecerá el Gateway ya creado en la pestaña de 'Configuración de comunicación'. Lo seleccionaremos y pulsaremos sobre la opción 'Examinar red'. [Fig. 42]

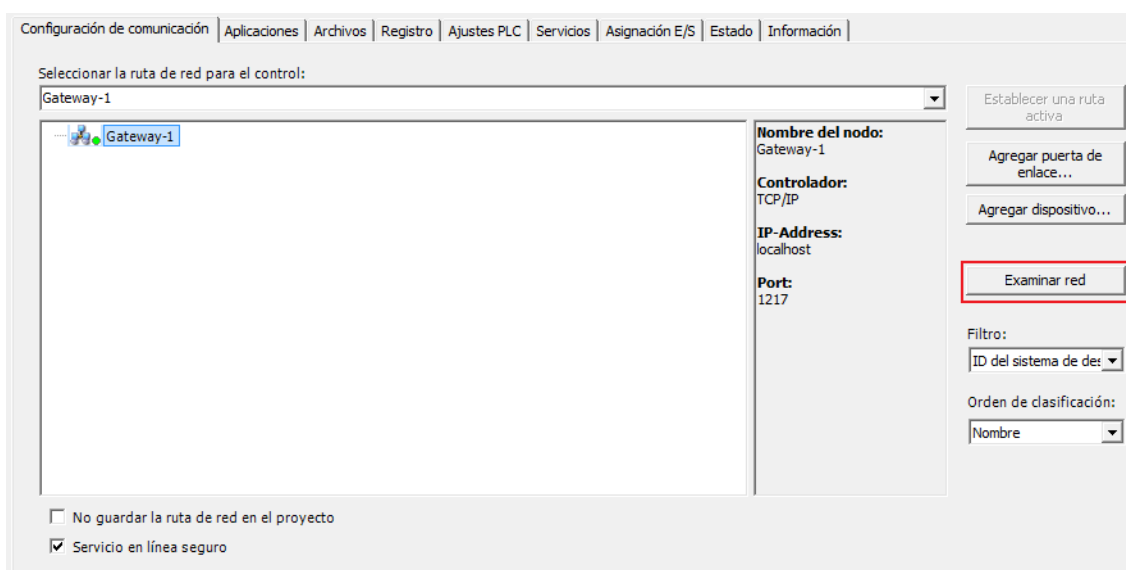


Fig. 42: Configuración de Comunicación. Examinar Red

El programa comenzará a buscar la red para posteriormente establecer una ruta activa de la misma. [Fig. 43]

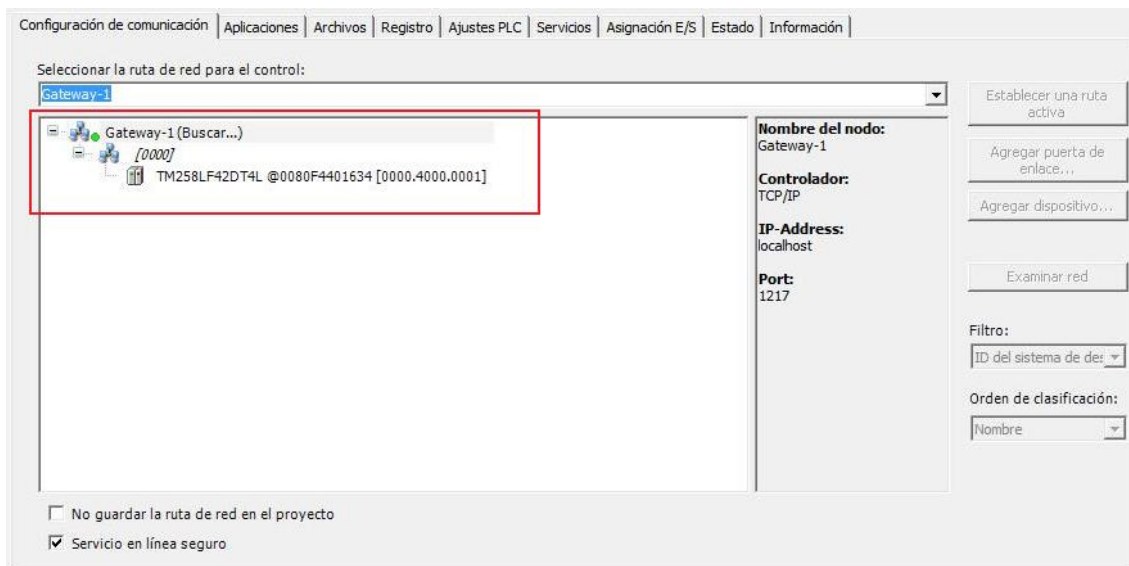


Fig. 43: Configuración de Comunicación. Buscando Red

Para finalizar, una vez encontrada la red, pulsaremos sobre 'Establecer una ruta activa' para la posterior conexión con el controlador. [Fig. 44]

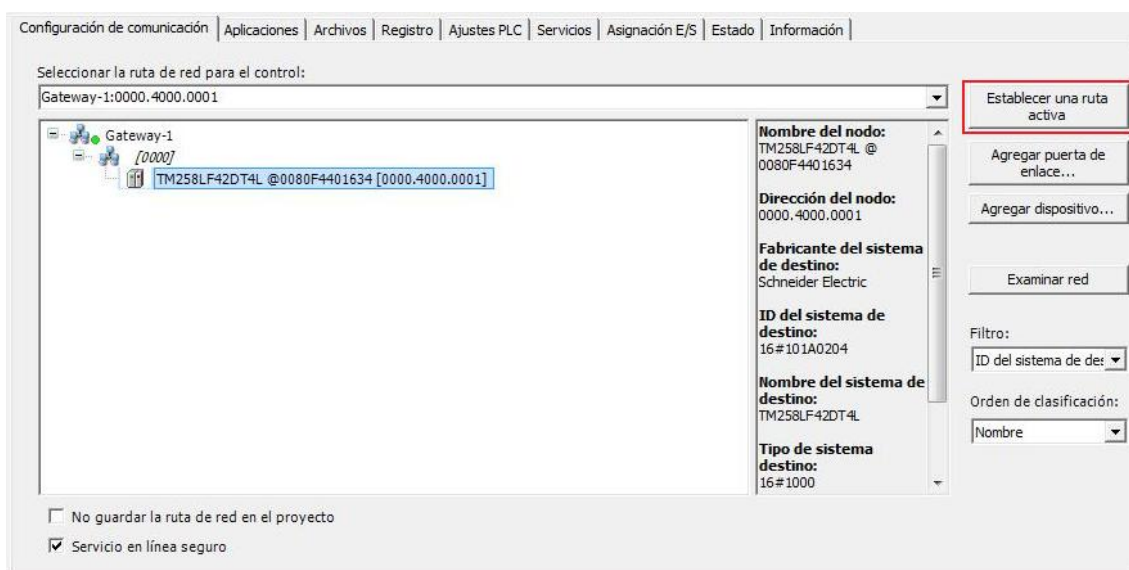


Fig. 44: Configuración de Comunicación. Establecer Ruta Activa

Nos aparecerá la siguiente advertencia [Fig. 45] que aceptaremos pulsando la combinación 'Alt+F' con el teclado.

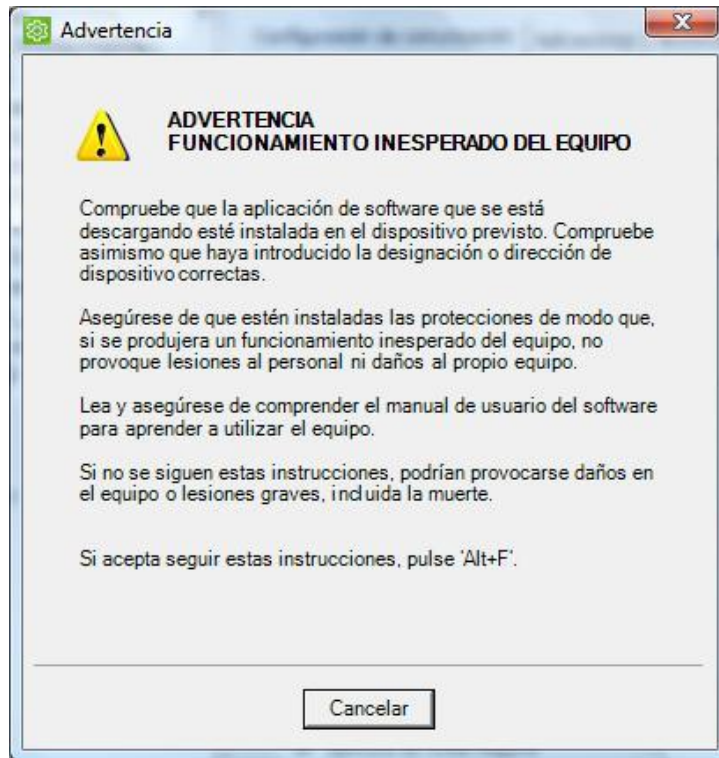


Fig. 45: Advertencia Establecimiento Ruta Activa

Una vez establecida la ruta activa, el autómatas ya estará disponible para descargar las aplicaciones. [Fig. 46]

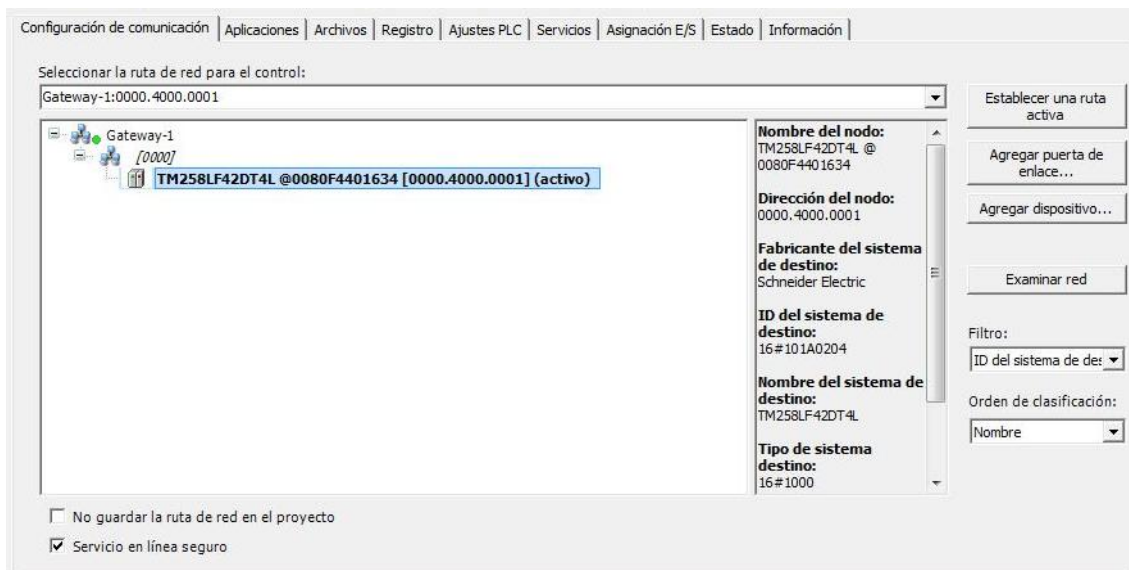


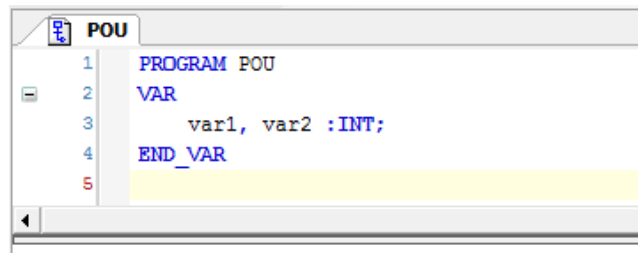
Fig. 46: Configuración de Comunicación. Ruta Activa

Para crear una nueva aplicación, habrá que crear una nueva sección de programa donde se escribirá el código del programa. Estas secciones son las denominadas POU's.

Existen tres tipos de POU's bien diferenciados:

- Programa [Fig. 47]: devuelve diferentes valores durante el funcionamiento. Se conservan todos los valores desde la última vez que se ejecutó el

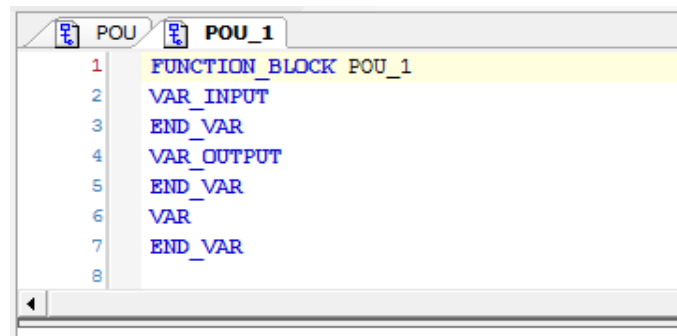
programa hasta la siguiente ejecución. Además, puede ser llamado por otra POU.



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

Fig. 47: POU. Programa

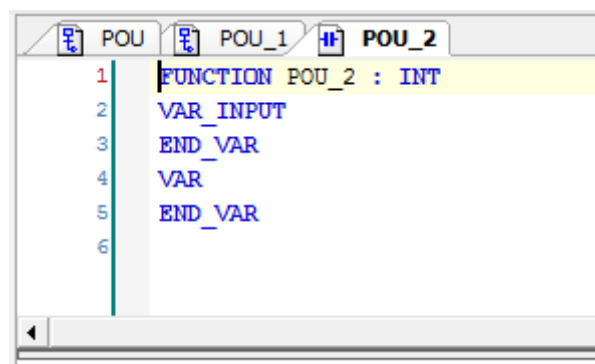
- **Bloque de funciones** [Fig. 48]: proporciona uno o varios valores durante el procesamiento de un programa. A diferencia de las funciones, los valores de las variables de salida y las variables internas necesarias persisten desde una ejecución del bloque a la siguiente ejecución. Además, tiene que ser llamado desde un mismo POU programa.



```
1 FUNCTION_BLOCK POU_1
2 VAR_INPUT
3 END_VAR
4 VAR_OUTPUT
5 END_VAR
6 VAR
7 END_VAR
8
```

Fig. 48: POU. Bloque de funciones

- **Función** [Fig. 49]: produce exactamente un elemento de datos cuando se procesa, es decir, devuelve una salida y no tiene variables internas persistentes. Hay que especificar el tipo de retorno de la función, en el caso de la imagen, INT.



```
1 FUNCTION POU_2 : INT
2 VAR_INPUT
3 END_VAR
4 VAR
5 END_VAR
6
```

Fig. 49: POU. Función

Para crear una sección nueva [Fig. 50], en la ventana de dispositivos vamos a 'Application' y hacemos click con el botón derecho seleccionando la opción POU en Agregar Objeto.

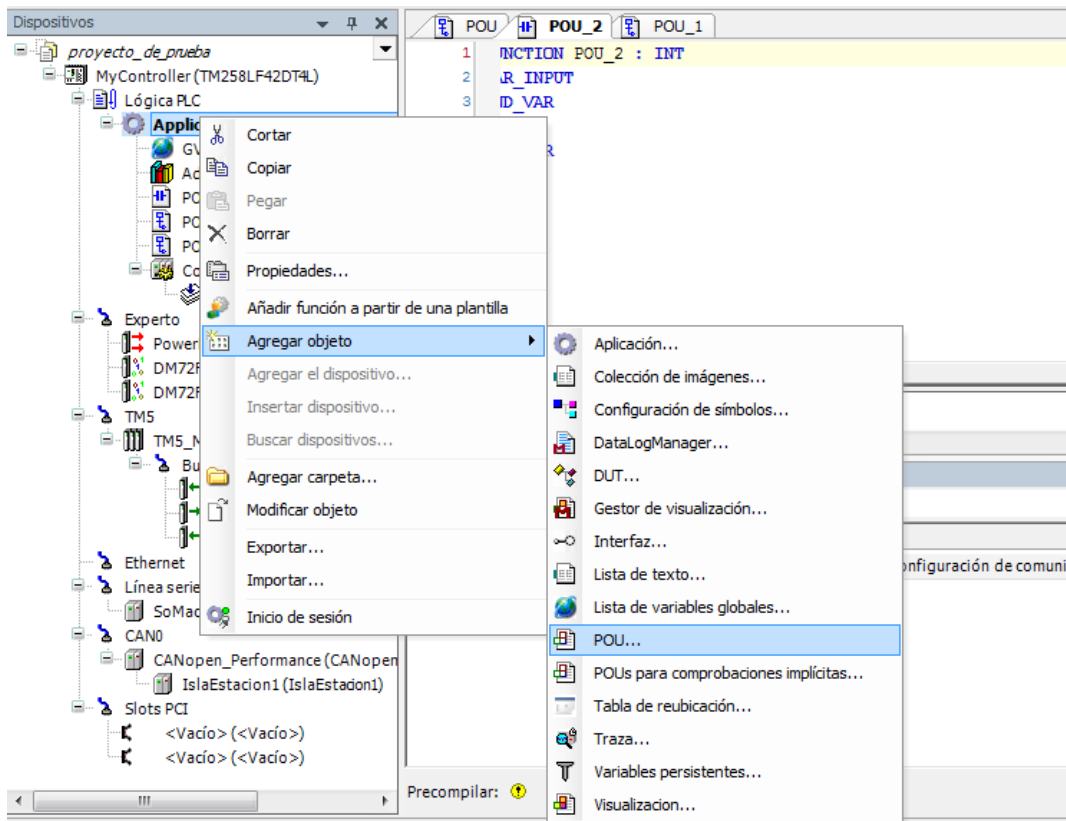


Fig. 50: Creación Nueva Sección de Trabajo

Nos aparecerá la siguiente ventana en la cual seleccionaremos el tipo de POU con el que vamos a trabajar además del lenguaje que vamos a utilizar. [Fig. 51]

En nuestro caso señalaremos la opción Programa y como lenguaje de implementación empezaremos con el lenguaje ST o también llamado Texto Estructurado.

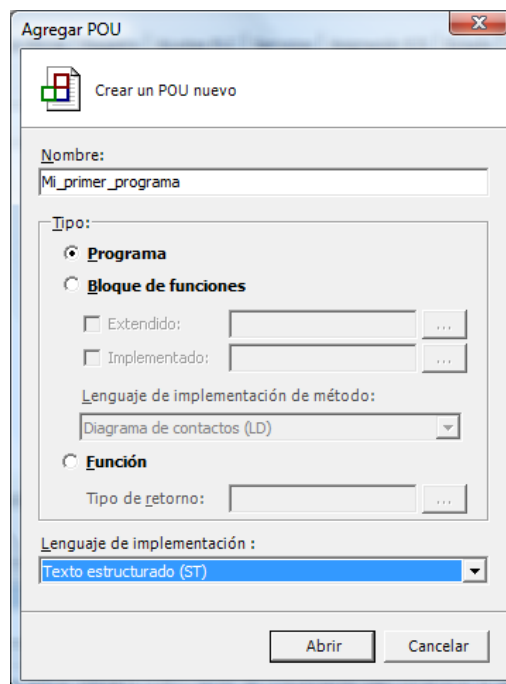


Fig. 51: Ventana Agregar POU

En el área central de trabajo de la pantalla nos aparece una nueva ventana dividida en dos partes, una para la declaración de variables y otra para la programación. [Fig. 52]

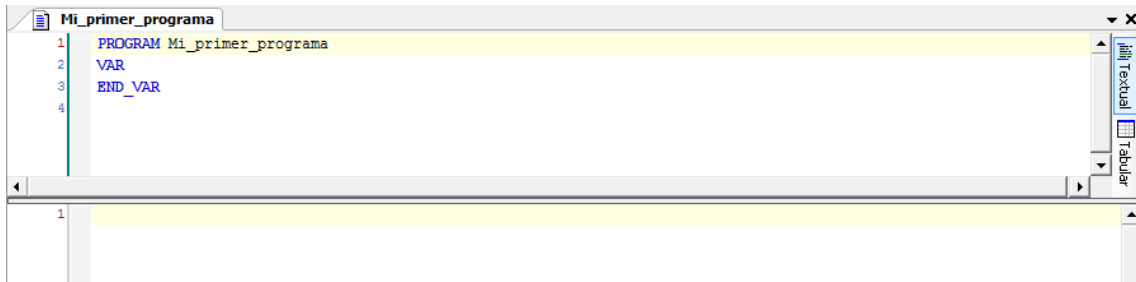


Fig. 52: Ventana de Programación

Una vez creada dicha POU, nos aparecerá en la ventana de dispositivos dentro del menú 'Application'. [Fig. 53]

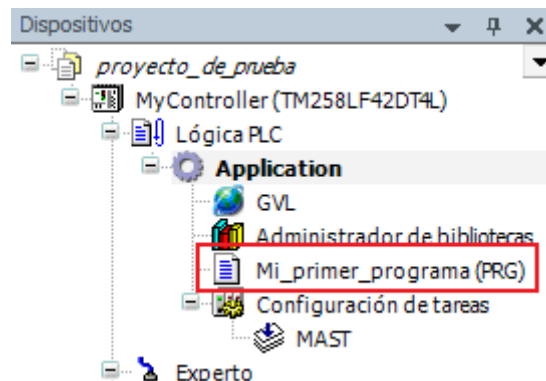


Fig. 53: Aplicación creada

El siguiente paso será la configuración de las tareas. Para configurar las tareas habrá que hacer doble click sobre las tareas que se encuentran en el menú configuración de tareas dentro de la ventana de dispositivos. En nuestro caso solo contamos con la tarea MAST por lo que haremos doble click sobre ella apareciéndonos la siguiente ventana en el área de trabajo. [Fig. 54]

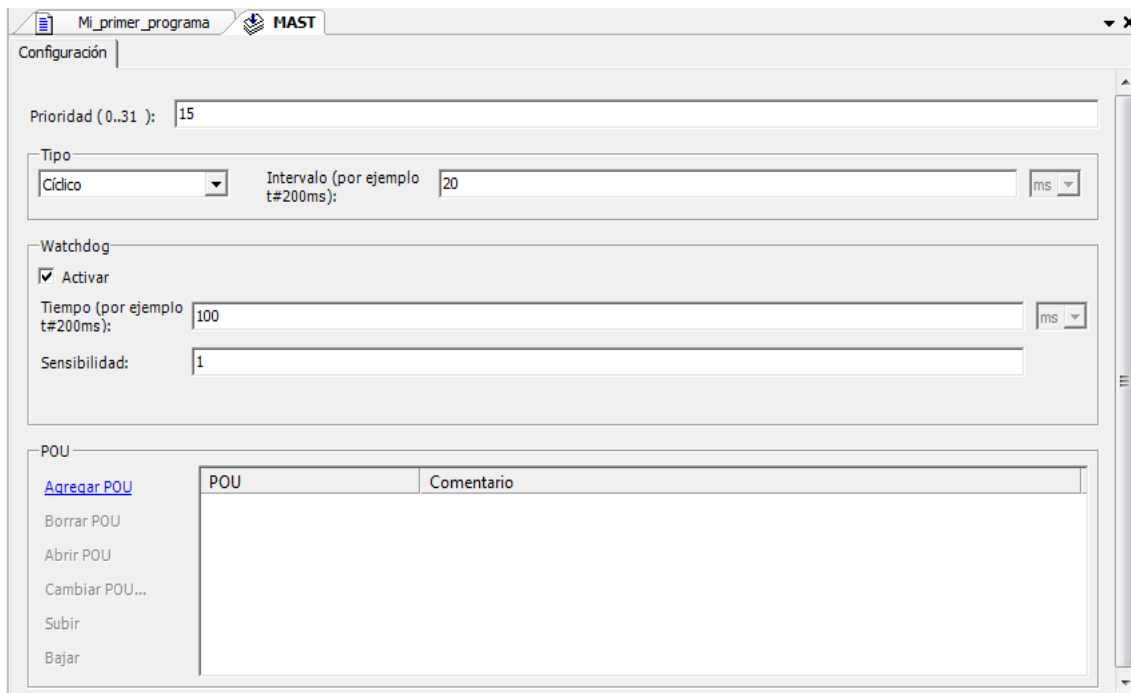


Fig. 54: Ventana de Configuración Tarea MAST

En dicha ventana aparecen todas las opciones para configurar la tarea las cuales se dividen en:

- **Prioridad:** permite definir la prioridad de la tarea actual representando el '0' la máxima prioridad y el '31' la mínima prioridad.
- **Tipo:** permite definir el tipo de tarea que puede ser:
 - ✓ cíclico
 - ✓ controlado por eventos
 - ✓ controlado externamente por eventos
 - ✓ ejecución libre
- **Watchdog:** permite habilitar el Watchdog así como su sensibilidad.
- **POU:** permite añadir programas de la POU que se hayan creado previamente en la tarea y fijar el orden de ejecución de las POU en la modalidad online.

Una vez configurada la tarea, para añadir una POU a ésta, se realiza en la misma ventana donde hemos configurado la tarea. Para ello haremos click sobre 'Agregar POU'. [Fig. 55]

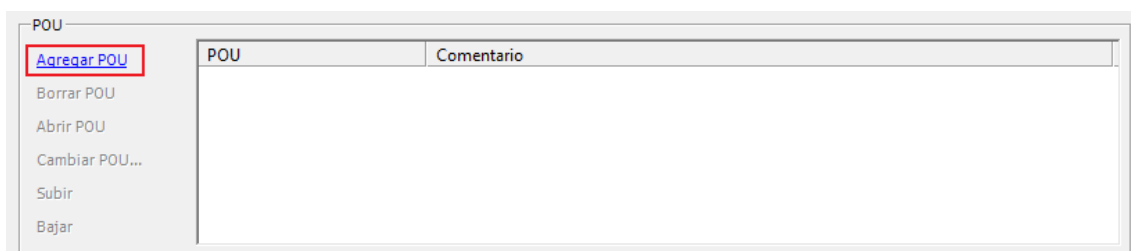


Fig. 55: Agregar POU

Seleccionaremos la POU deseada, en nuestro caso, 'Mi_primer_programa'. [Fig. 56]

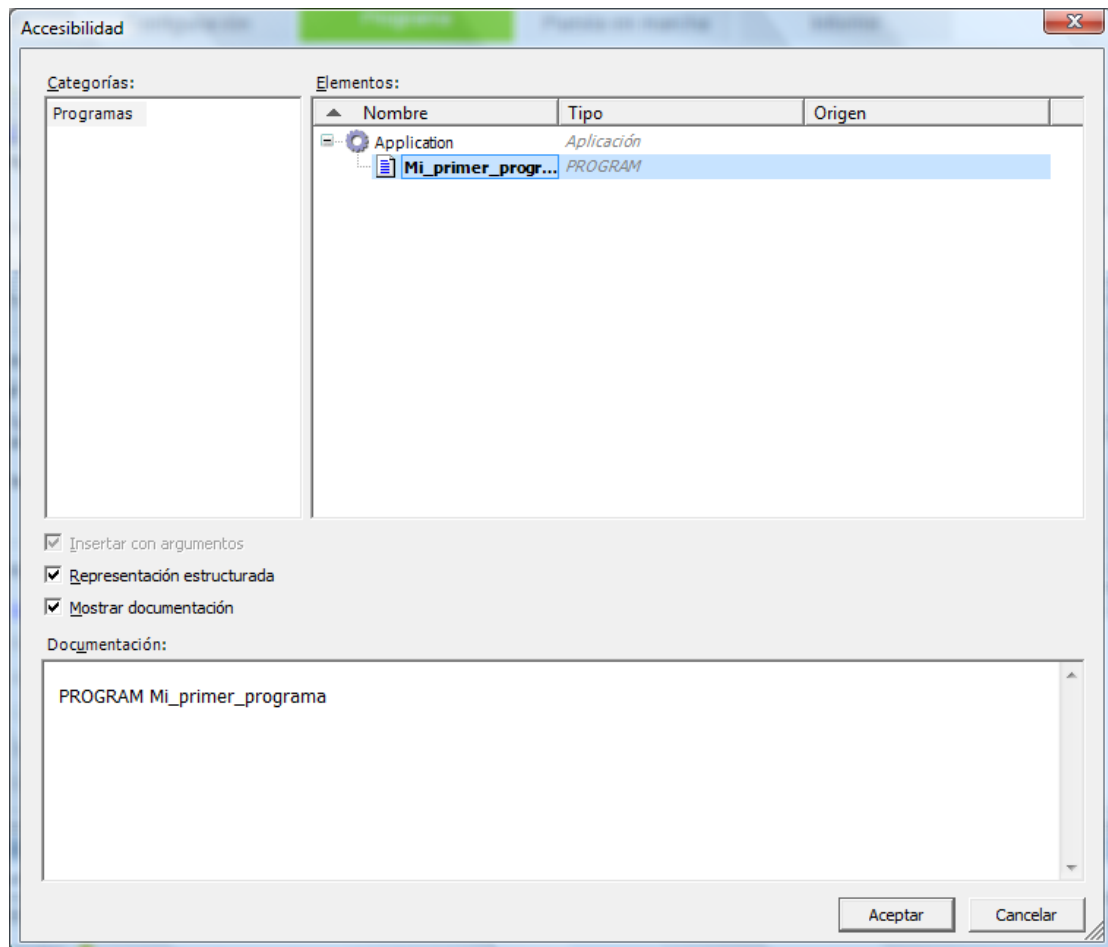


Fig. 56: Selección POU

Una vez seleccionada la POU y aceptada, ésta aparecerá en el listado de POU's agregadas a la tarea MAST. [Fig. 57]

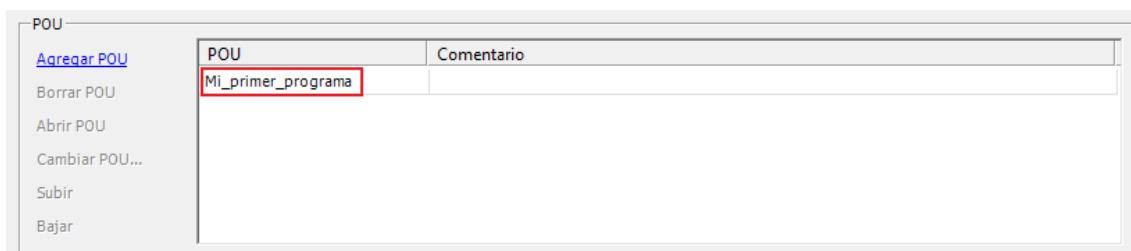


Fig. 57: Listado de POU's Agregadas

Ahora, hay que fijarse en la esquina inferior de la pantalla debajo de la ventana de 'Mensajes' para comprobar que la precompilación es correcta. [Fig. 58]

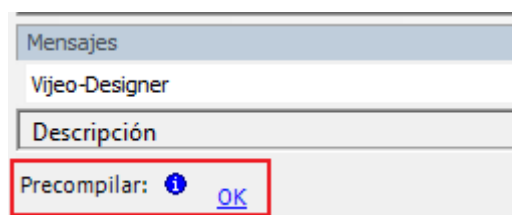


Fig. 58: Precompilación Correcta

Puede ser que nos aparezca un mensaje como el siguiente [Fig. 59] el cual nos indica que la POU no está precompilada todavía.

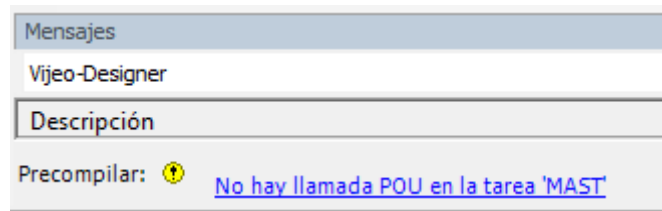


Fig. 59: POU no precompilada

En ese caso, iremos al menú general y en la pestaña 'compilar' elegiremos la opción compilar. Otra opción es presionar la tecla F11. [Fig. 60]

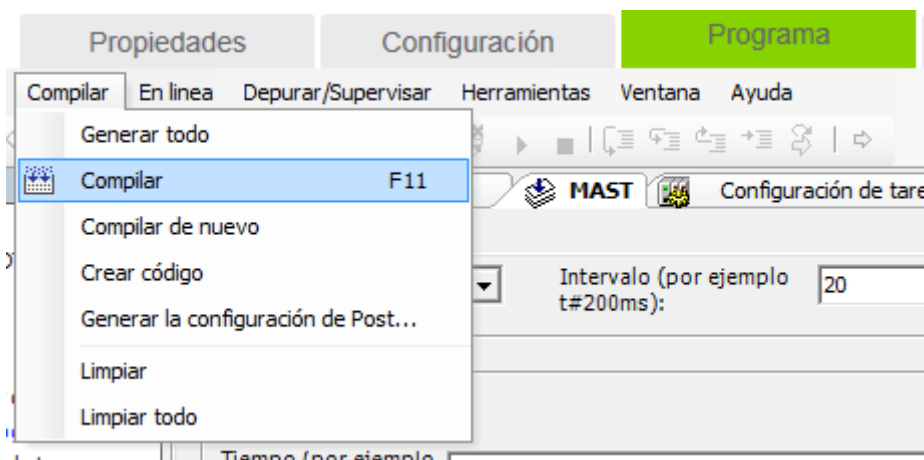


Fig. 60: Opción Compilar Menú General

Una vez realizado esto, ya habremos precompilado la POU y nos aparecerá un mensaje [Fig. 61] indicando que todo está correcto.

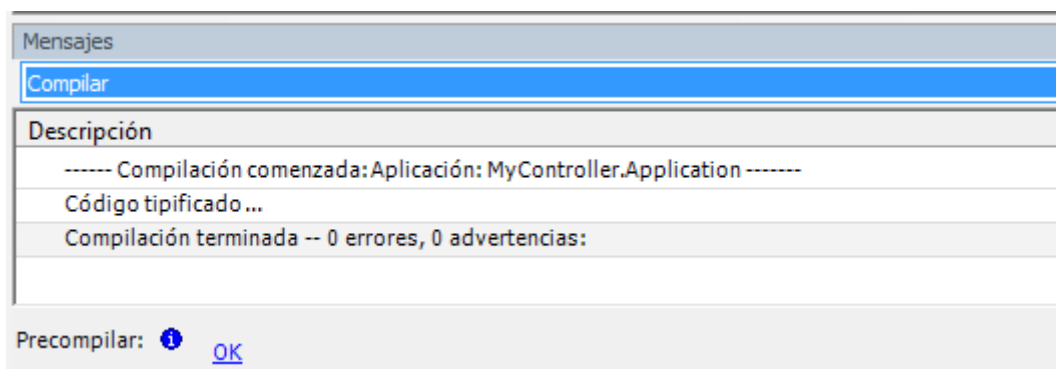


Fig. 61: POU Precompilada

➤ Ejercicio 5: programación en ST. Creación de un programa que calcule la media de cinco variables.

- En primer lugar, siguiendo los pasos anteriormente descritos, crearemos una nueva sección de trabajo en lenguaje ST a la que llamaremos 'Media_ST' y después de esto la agregaremos a la tarea MAST. [Fig. 62]

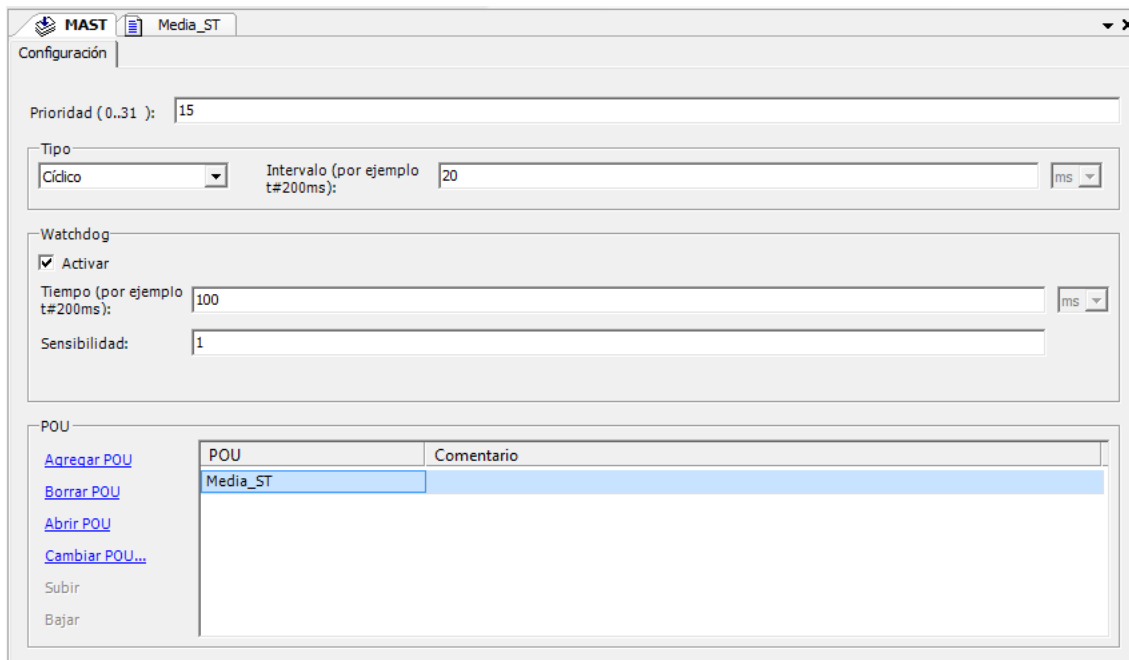


Fig. 62: POU ya Agregada

- Una vez realizado esto, comenzaremos con la programación. En primer lugar definiremos seis variables, cinco con las que haremos la media y una para guardar dicho resultado. Se tratan de variables locales por lo que sólo podrán utilizarse en esta POU. Las variables serán de tipo 'INT' (entero).

- Para declarar variables hay dos formas de poder realizarlo:

- La primera forma es más gráfica. Consiste en declararlas manualmente en la ventana de programa que nos aparece cuando creamos la POU siendo la sintaxis para ello la siguiente [Fig. 63]:

```

1  PROGRAM Media_ST
2  VAR
3  var1, var2, var3, var4, var5, media :INT;
4  END_VAR
5

```

Fig. 63: Declaración de variables

- La segunda forma consiste en ir al menú principal y en 'Edición' seleccionar la opción Declarar variable. [Fig. 64]

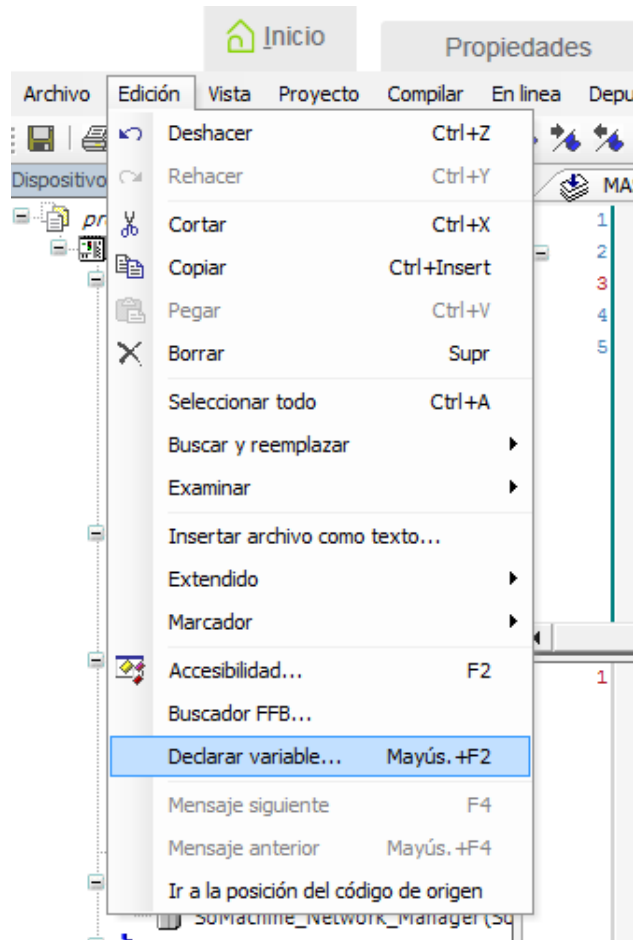


Fig. 64: Edición. Declarar Variables

Una vez realizado esto, nos aparecerá la siguiente ventana para declarar las variables. [Fig. 65]

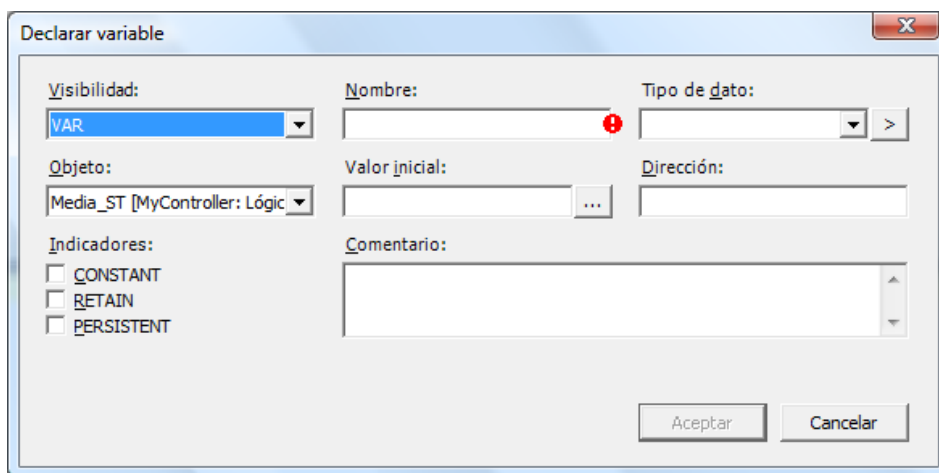


Fig. 65: Declarar Variable

En esta ventana definiremos el tipo de variable, el nombre, el tipo de dato, la POU a la que pertenece y si queremos, darle un valor inicial y el direccionamiento en memoria.

En nuestro caso, trabajaremos con variables locales por lo que seleccionaremos en el campo visibilidad la opción 'VAR'. En tipo de dato seleccionaremos la opción 'INT' y en el campo Objeto la sección de trabajo recientemente creada, Media_ST. Les asignaremos nombres sencillos a las variables como por ejemplo Var1,..., Var5 y media. Los campos Valor inicial, Dirección y Comentario los dejaremos libres y le daremos a aceptar. [Fig. 66]

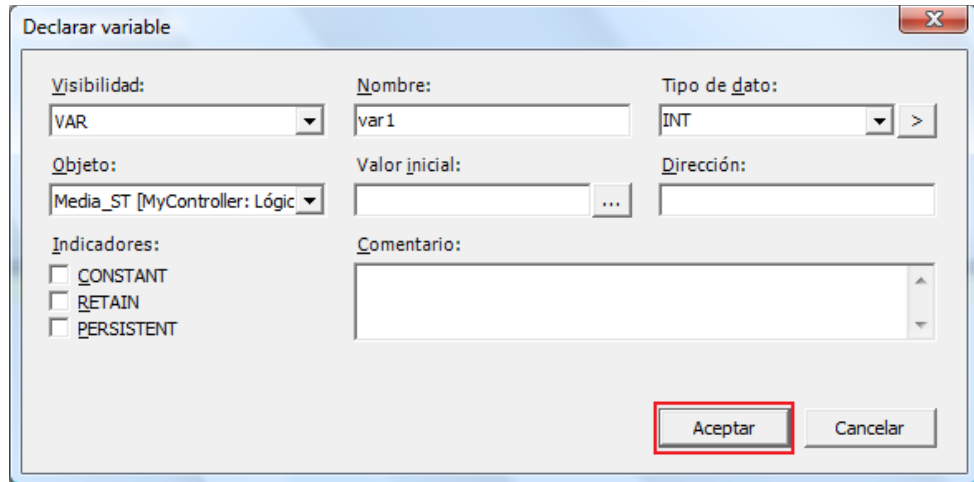


Fig. 66: Declaración de Variable

Realizaremos este mismo paso con todas las variables y una vez realizado, estas aparecerán en la ventana de programación de la POU al igual que cuando las creamos anteriormente manualmente.

- El código de programación de este ejercicio es muy simple y se muestra a continuación. [Fig. 67]

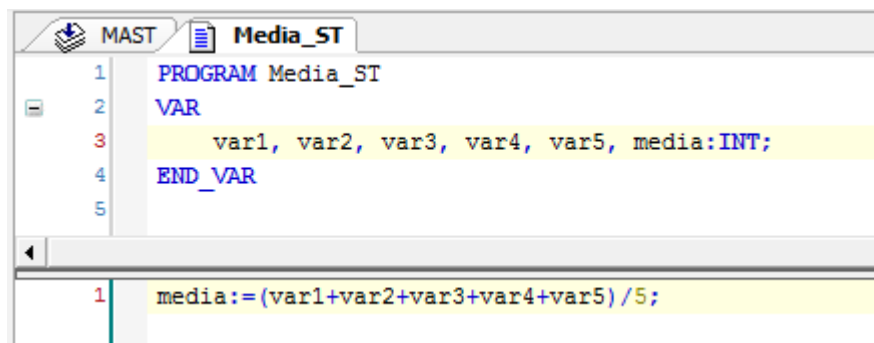


Fig. 67: Programación Media Variables

- Por último y para acabar el ejercicio, compilaremos el programa como anteriormente se ha visto. Si todo está correcto, en la ventana de mensajes no saldrán errores ni advertencias. [Fig. 68] Entonces, lo guardaremos.

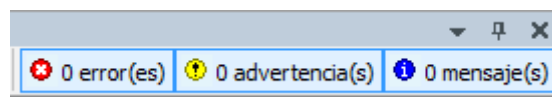


Fig. 68: Mensaje Errores

- Para ver qué sucedería si hubiéramos cometido algún error a lo largo de la programación, borraremos el símbolo ‘:’ que sigue a media en la programación y volveremos a compilar el programa. Nos aparecerá el siguiente error. [Fig. 69]

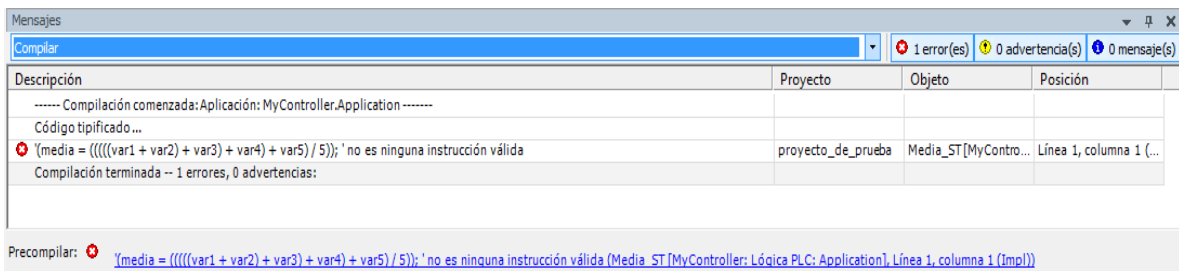


Fig. 69: Mensaje de Error de Compilación

- El programa al compilar detecta dónde se encuentra el error y nos informa de cuál es dicho error y en qué parte del programa se encuentra para poder corregirlo. Por esto, la ventana de ‘Mensajes’ es de gran importancia a la hora de depurar los programas.

Siguiendo con el curso de creación de aplicaciones, el software de SoMachine dispone además de un simulador PLC offline. Este permite descargar las aplicaciones y ejecutarlas sin necesidad de tener el PLC físicamente. Esto nos ayuda en el desarrollo y depuración de los programas creados. Para iniciar el simulador, tendremos que ir a la barra de menú general de la ventana de programación y seleccionar ‘Simulación’ en la pestaña ‘En línea’ del menú. [Fig. 70]

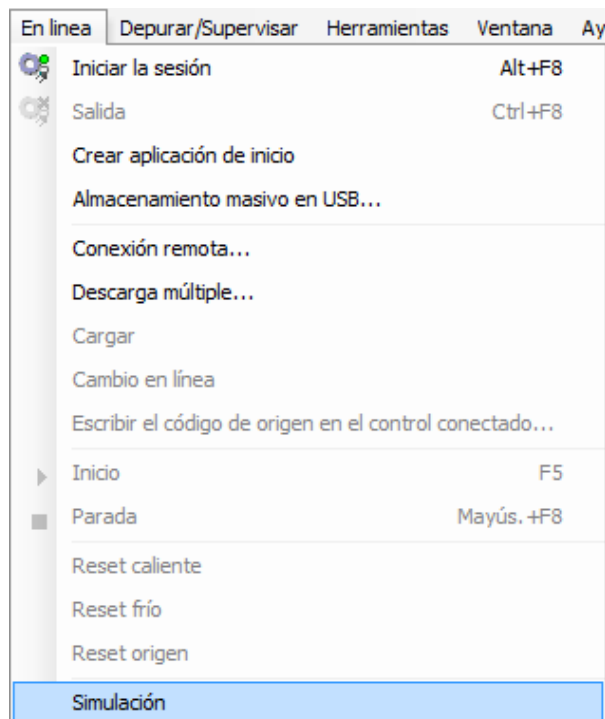


Fig. 70: Simulación PLC Offline

En la parte inferior de la pantalla podremos ver cómo estamos en dicho modo de simulación. [Fig. 71]

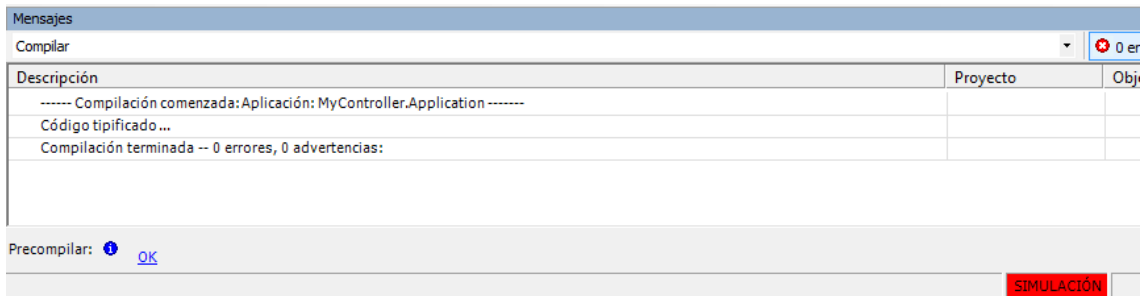


Fig. 71: Modo Simulación en Rojo

El siguiente paso será iniciar la sesión. Para ello iremos al menú general de la ventana de programación y en la pestaña 'En Línea' seleccionaremos la opción 'Iniciar la sesión'. Este paso es similar al anteriormente realizado para conectar el simulador. Veremos que la opción 'Simulación' aparece marcada y el simulador estará conectado y en línea.

- **Ejercicio 6:** Simulación del programa de la media de variables.
 - Siguiendo los pasos anteriormente indicados, conectaremos el programa de la media de cinco variables al simulador e iniciaremos la sesión.
 - Ya que todavía no existe ningún programa en el simulador, nos aparecerá en pantalla el siguiente mensaje el cual habrá que aceptar. [Fig. 72]

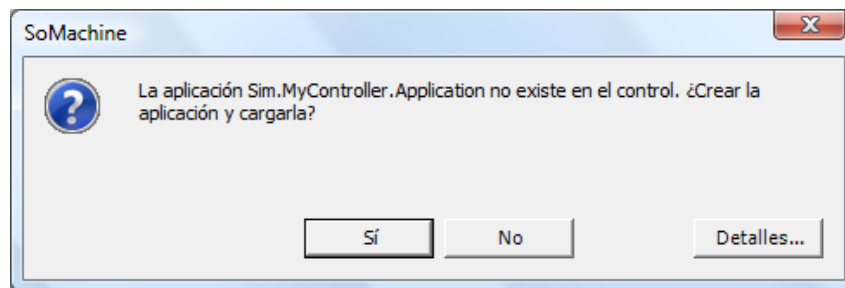


Fig. 72: Creación y Carga de la Aplicación

- Una vez creada la aplicación en el control, la barra de estado nos indica que el programa ha sido cargado y si ha habido modificaciones en él. A su vez, nos indica con la palabra 'Stop' sobre fondo rojo el estado del autómata. [Fig. 73]

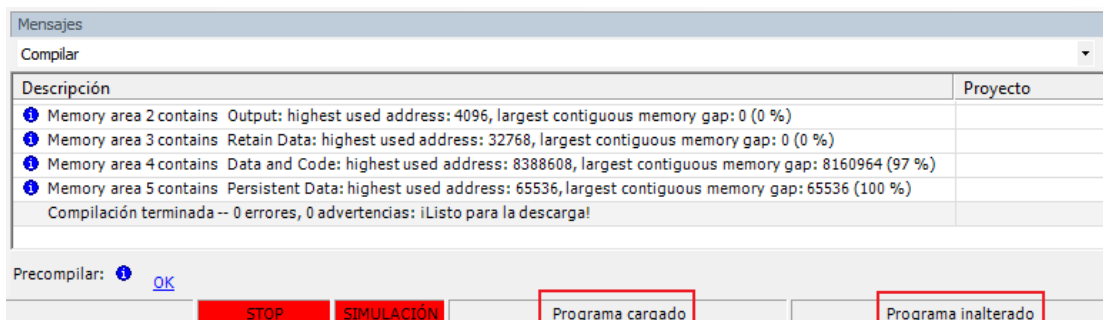


Fig. 73: Programa Cargado

- Además, la ventana 'Dispositivos' quedará de la siguiente manera. [Fig. 74]

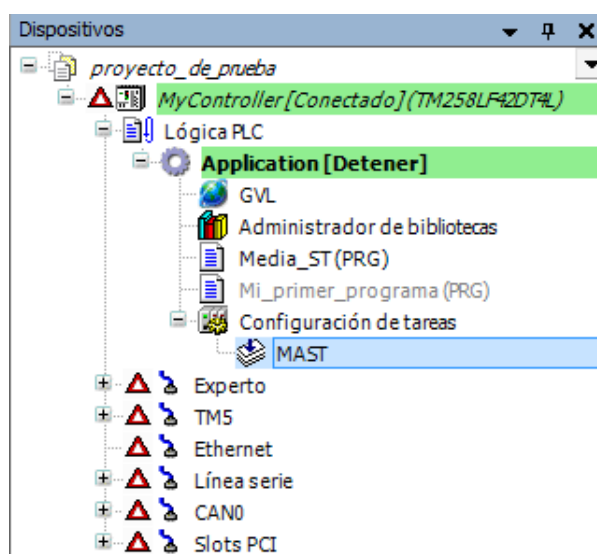


Fig. 74: Ventana Dispositivos

- Si no aparece en el centro del área de trabajo abierta la ventana de la POU, iremos a la ventana dispositivos y haremos doble click sobre ella, en este caso Media_ST apareciendo la siguiente ventana. [Fig. 75]

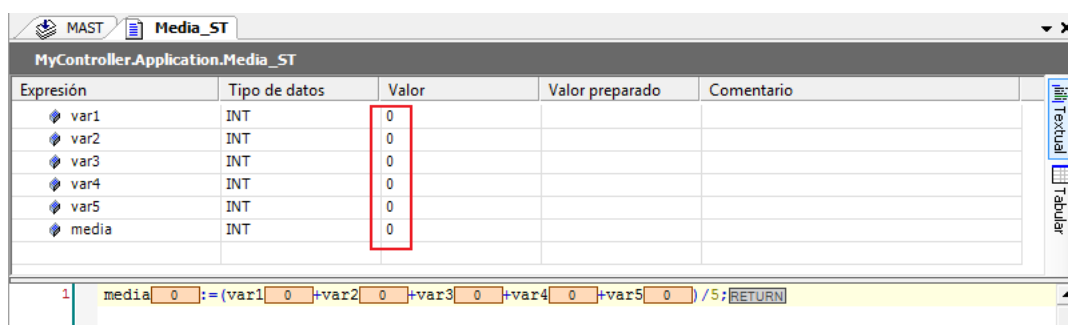


Fig. 75: POU Media Variables

- Como puede observarse en la imagen anterior [Fig. 75], todas las variables tienen un valor inicial nulo.

- Introduciremos en el campo 'Valor preparado' diferentes valores para las variables var1, var2, var3, var4 y var5. Los valores no se aplican en ese mismo momento a las variables si no que quedan preparados. [Fig. 76]

Expresión	Tipo de datos	Valor	Valor preparado
var1	INT	0	5
var2	INT	0	9
var3	INT	0	13
var4	INT	0	6
var5	INT	0	12
media	INT	0	


```

1  media 0 := (var1 0 <5> +var2 0 <9> +var3 0 <13> +var4

```

Fig. 76: Media. Valores Preparados

- En el menú general, seleccionaremos la opción 'Escribir valores para todas las aplicaciones en línea' dentro de la pestaña Depurar/Supervisar como se muestra a continuación. [Fig. 77]

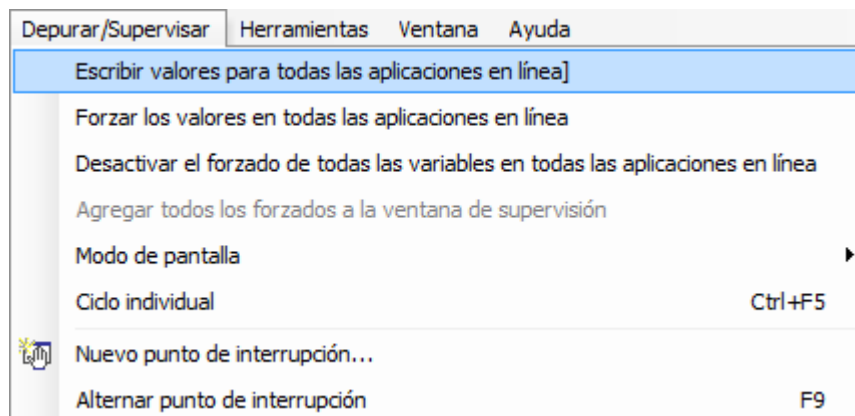


Fig. 77: Escribir los Valores Anteriores en las Variables

- Nos aparecerá una nueva ventana para confirmar la acción de escribir los nuevos valores la cual aceptaremos. [Fig. 78]

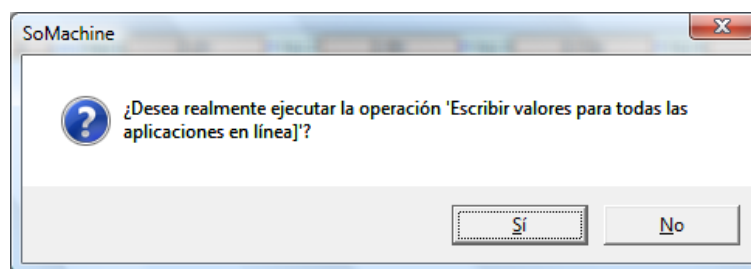


Fig. 78: Confirmación Escribir Valores

- Una vez aceptado, se escribirán en las variables los valores introducidos anteriormente en el campo 'Valor preparado'. [Fig. 79]

Expresión	Tipo de datos	Valor	Valor preparado	Com
var1	INT	5		
var2	INT	9		
var3	INT	13		
var4	INT	6		
var5	INT	12		
media	INT	0		

```

1 media 0 := (var1 5 + var2 9 + var3 13 + var4 6 + var5 12) / 5; RETURN

```

Fig. 79: Valores Escritos en Variables

- El siguiente paso será ejecutar el programa. Para ello, pulsaremos sobre el botón de 'Inicio' en la barra de herramientas [Fig. 80] o también sobre la opción 'Inicio' en la pestaña 'En línea' del menú general de la ventana de programación. [Fig. 81]



Fig. 80: Barra de Herramientas. Inicio

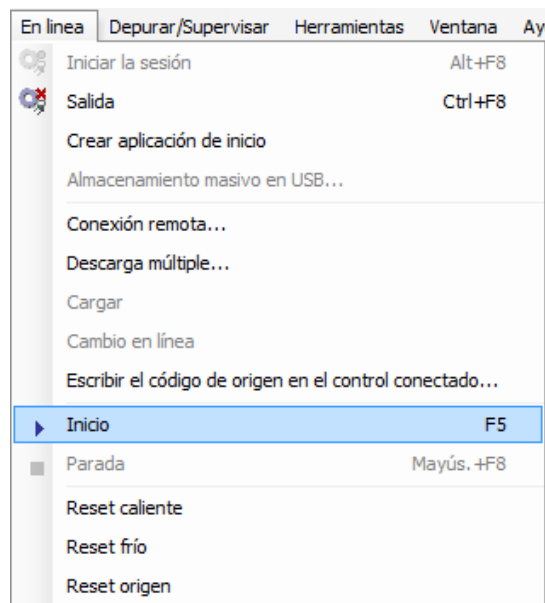


Fig. 81: Pestaña En línea. Inicio

- Podremos observar en la barra de estado como el programa está 'En ejecución' apareciendo dicha frase sobre fondo verde. [Fig. 82]

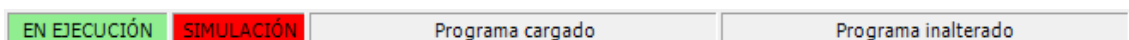


Fig. 82: Barra de Estado en Ejecución

- Además, la media de los cinco valores habrá sido calculada mostrándose el valor de dicha media en el campo 'Valor' Correspondiente a dicha variable.
[Fig. 83]

Expresión	Tipo de datos	Valor	Valor prepara
var1	INT	5	
var2	INT	9	
var3	INT	13	
var4	INT	6	
var5	INT	12	
media	INT	9	

```

1  media 9 := (var1 5 + var2 9 + var3 13 + var4 6 + var5 12) / 5; RETURN
  
```

Fig. 83: Media Calculada

- Cerraremos la sesión pulsando sobre el botón correspondiente en la barra de herramientas [Fig. 84] o pulsando sobre la opción 'Salida' en la pestaña 'En línea' del menú general de la ventana [Fig. 85] y guardaremos el proyecto.



Fig. 84: Barra de Herramientas. Salida Sesión

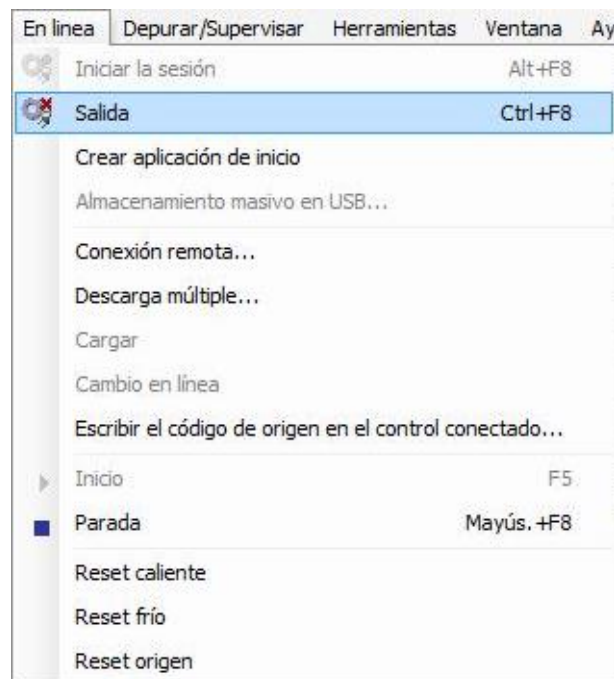


Fig. 85: Pestaña En Línea. Salida

Una vez realizado el ejercicio anterior, crearemos una nueva sección de programa en un nuevo lenguaje de implementación como hemos estado haciendo hasta este momento. En esta ocasión, seleccionaremos el lenguaje LD o también llamado diagrama de contactos.

Una vez creada la nueva sección de programa, a la derecha del área de trabajo nos aparece una nueva ventana de Herramientas donde aparecen los distintos elementos para dibujar el diagrama. [Fig. 86] Para introducirlos en el dibujo utilizando dicha ventana, basta con arrastrarlos hasta la posición deseada dentro de la red en la zona de trabajo.

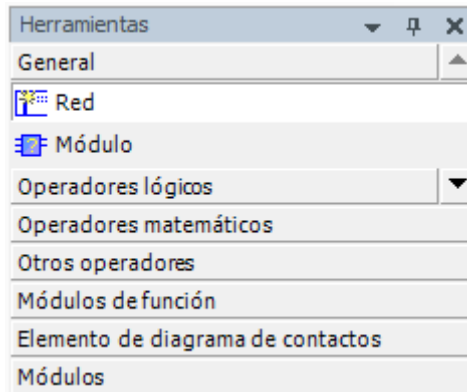


Fig. 86: Ventana de Herramientas

Para dibujar el diagrama también puede utilizarse la barra de herramientas [Fig. 87] o el botón derecho del ratón sobre la red donde también se encuentran los símbolos del lenguaje LD. [Fig. 88]



Fig. 87: Barra de Herramientas

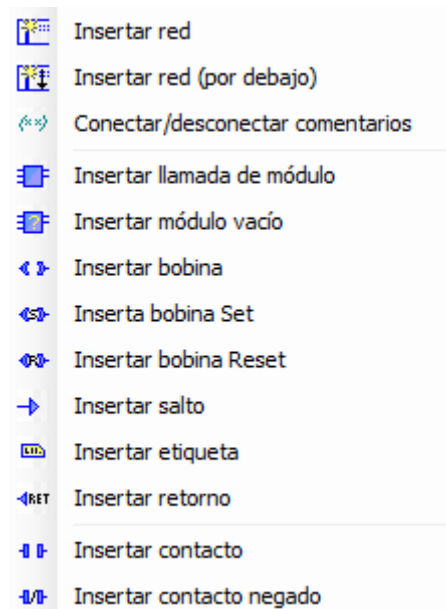


Fig. 88: Símbolos Lenguaje LD. Botón Derecho

➤ **Ejercicio 7:** programación en LD.

- Introducir el siguiente esquema [Fig. 89] y analizar su funcionamiento indicando en qué casos las salidas permanecen o no activas. Para ello haremos uso de la simulación del programa. En esta ocasión, las variables serán de tipo BOOL.

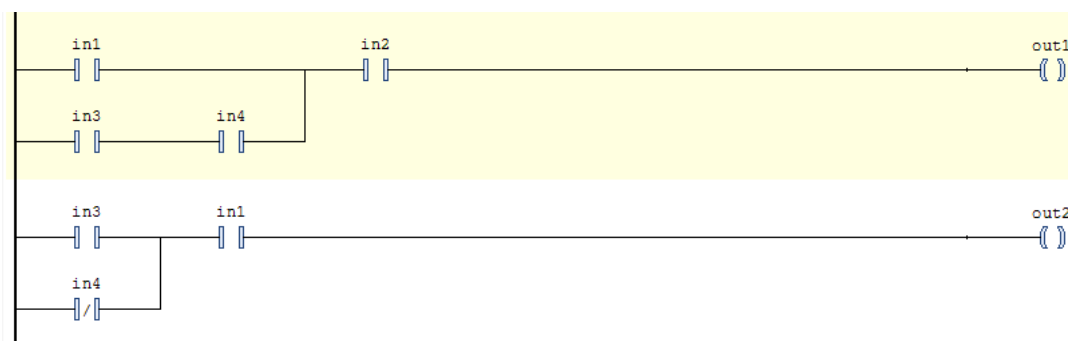


Fig. 89: Esquema Programación LD

Para trabajar otros lenguajes de programación, realizaremos unos pequeños y simples ejercicios para el mínimo aprendizaje de los mismos.

➤ **Ejercicio 8:** programación en IL, lista de instrucciones.

- Crear una nueva sección de programa en lenguaje IL e introducir las siguientes instrucciones. [Fig. 90] Analizar el funcionamiento y comparar los resultados con los obtenidos en el ejercicio anterior. Las variables serán de tipo BOOL.

1	LD	in1
	OR (in3
	AND	in4
)	
	AND	in2
	ST	out1
2	LD	in3
	ORN	in4
	AND	in1
	ST	out2

Fig. 90: Instrucciones en IL

➤ **Ejercicio 9:** programación en FBD, diagrama de bloques de funciones.

- Crear una nueva sección de trabajo en lenguaje FBD e introducir el siguiente esquema. [Fig. 91] El método para introducir los bloques de funciones es similar al seguido con el diagrama de lenguaje de contactos. Basta con seleccionarlos a través de la barra de herramientas o arrastrarlos desde la ventana de herramientas disponible a la derecha de la zona de trabajo al área de programación.

- Analizar de nuevo su funcionamiento y comparar los resultados con los ejercicios anteriores.

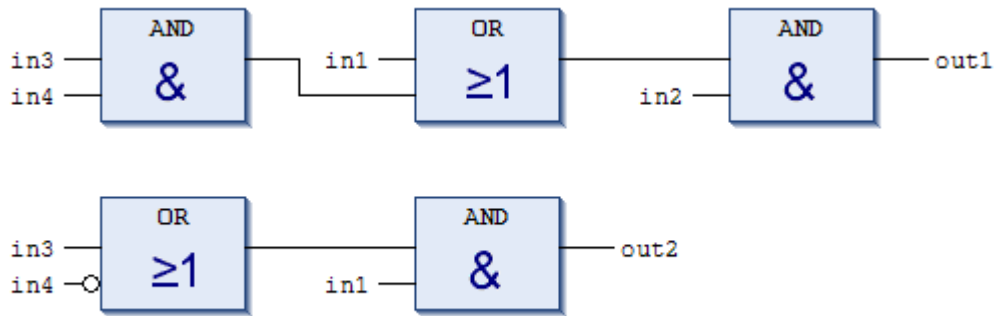


Fig. 91: Esquema Programación En FBD

Como puede observarse, los programas de los ejercicios 7, 8 y 9 son idénticos cambiando únicamente el lenguaje de programación. El software SoMachine dispone de una opción muy útil para convertir secciones de programa entre los lenguajes FBD, IL y LD. Para ello, basta con ir al menú general de programación y en la pestaña 'FBD/LD/IL' seleccionamos 'Vista' y ahí podremos convertir programas entre dichos lenguajes. [Fig. 92]

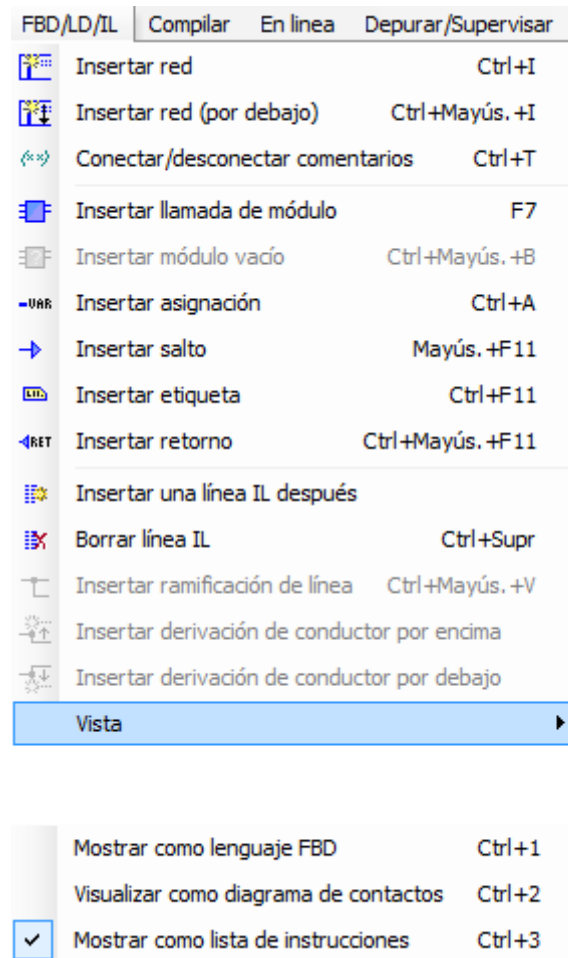


Fig. 92: Conversión entre lenguajes FBD/LD/IL

➤ **Ejercicio 10:** programación en LD, temporizadores.

- Crear una nueva sección de trabajo en lenguaje LD e introducir el siguiente esquema. [Fig. 93]

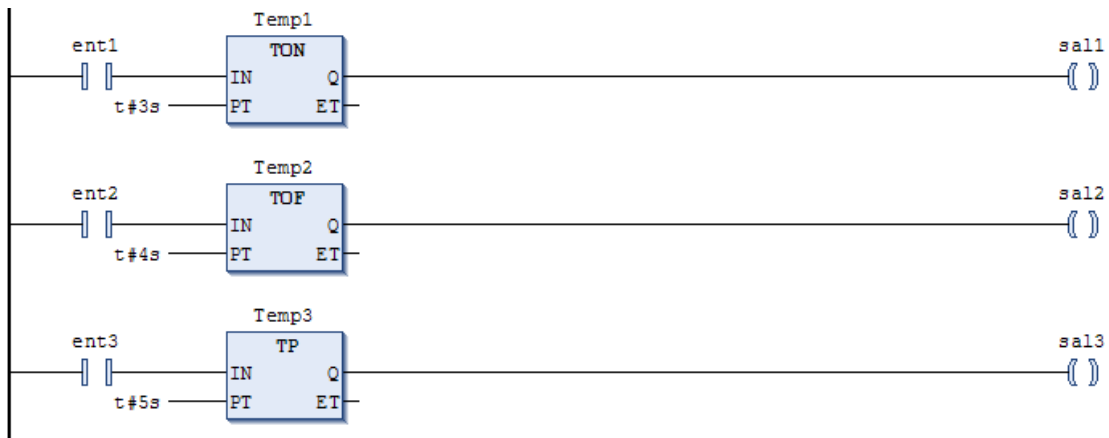


Fig. 93: Esquema Programación LD. Temporizadores

- Para introducir un bloque temporizador, pulsaremos sobre el icono que aparece marcado en la imagen siguiente [Fig. 94] y nos aparecerá el siguiente menú. [Fig. 95] Bastará con seleccionar el deseado y pulsar sobre aceptar.



Fig. 94: Barra Herramientas. Insertar Llamada de Módulo

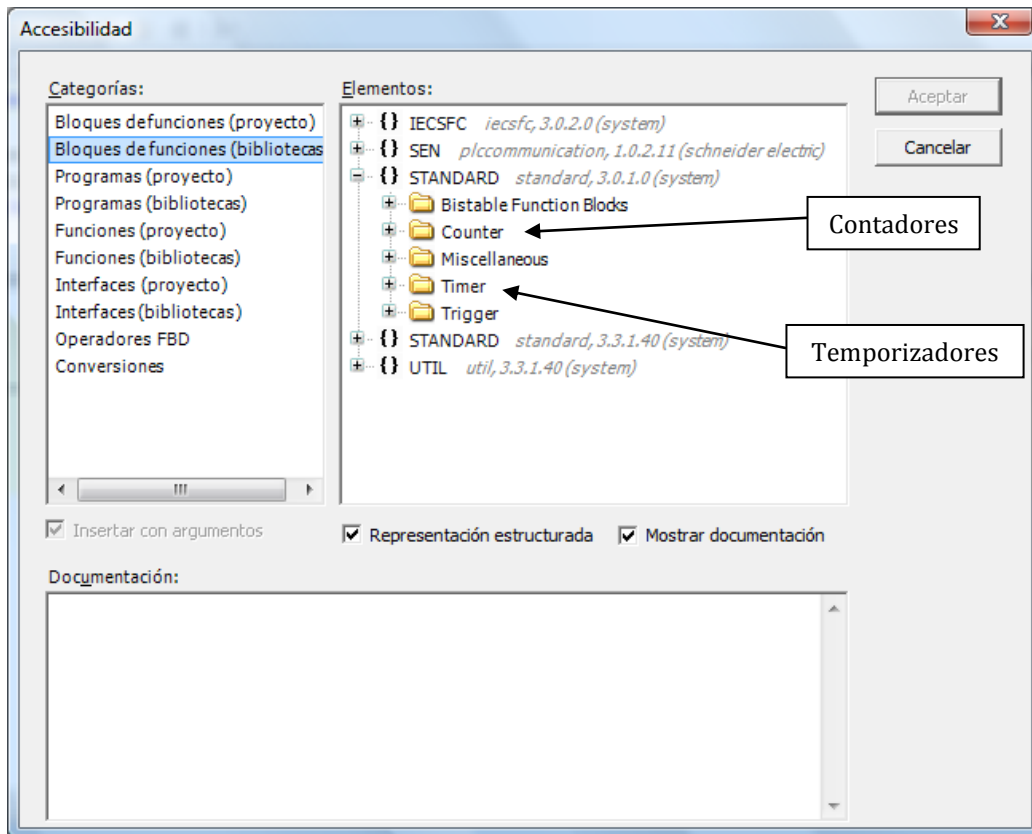


Fig. 95: Ventana Insertar Bloques de Funciones

➤ Ejercicio 11: programación en ST, temporizadores.

- Crear una nueva sección de trabajo en lenguaje ST e introducir el siguiente esquema. [Fig. 96]

```
1 Timer_TON(  
2     IN:= ent1,  
3     PT:= T#3S,  
4     Q=> sal1,  
5 );  
6 Timer_TOF(  
7     IN:= ent2,  
8     PT:= T#4S,  
9     Q=> sal2,  
10 );  
11 Timer_IP(  
12     IN:= ent3,  
13     PT:= T#5S,  
14     Q=> sal3,  
15 );
```

Fig. 96: Esquema Programación ST. Temporizadores

➤ Ejercicio 12: programación en LD, contadores.

- Crear una nueva sección de trabajo en lenguaje LD e introducir el siguiente esquema. [Fig. 97]



Fig. 97: Esquema Programación LD. Contadores

➤ Ejercicio 13: programación en ST, contadores.

- Crear una nueva sección de trabajo en lenguaje ST e introducir el siguiente esquema. [Fig. 98]

```
1 Contador(  
2     CU:= ent4,  
3     RESET:= ent5,  
4     PV:= 10,  
5     Q=> sal4,  
6 );
```

Fig. 98: Esquema Programación ST. Contadores

4- CREACIÓN ISLAS ADVANTYS

4.1- INTRODUCCIÓN A LA CONFIGURACIÓN DE ISLAS ADVANTYS

Para crear la configuración de cualquier isla como las disponibles en el laboratorio, utilizaremos el software de Advantys. En primer lugar iremos al menú Inicio → Programas → Schneider Electric → Advantys y lanzaremos dicho programa haciendo doble click sobre el mismo.

Al lanzar dicho programa nos aparecerá la siguiente ventana. [Fig. 99]

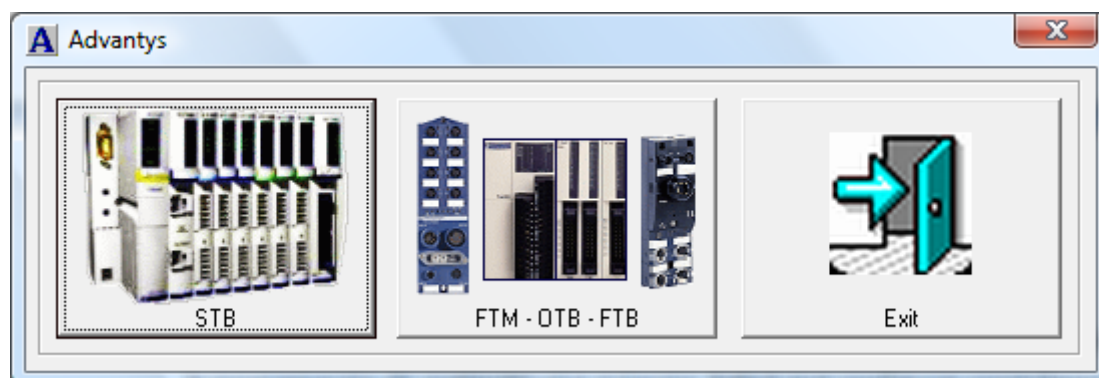


Fig. 99: Ventana Inicio Advantys

Pulsamos en STB y el programa se iniciará. Para crear una nueva isla vamos a Archivo/Área nueva de trabajo, y nos aparecerá la siguiente pantalla [Fig. 100]:

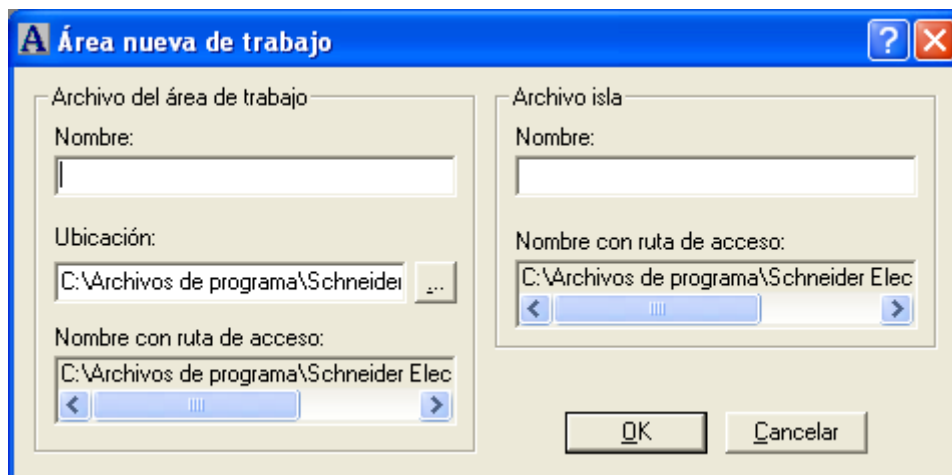


Fig. 100: Agregar Área Nueva de Trabajo

El Archivo del área de trabajo será la carpeta que alojará todos los archivos creados y el Archivo isla será el que contendrá la configuración de la isla.

Una vez abierta la isla, nos encontraremos con la siguiente pantalla [Fig. 101] en la cual, aparece un raíl vacío. En la ventana de 'Navegador de Catálogos' es dónde se encuentran los componentes. En ella habrá que buscar los correspondientes a nuestra isla y una vez encontrados, arrastrarlos hasta el raíl de la derecha.

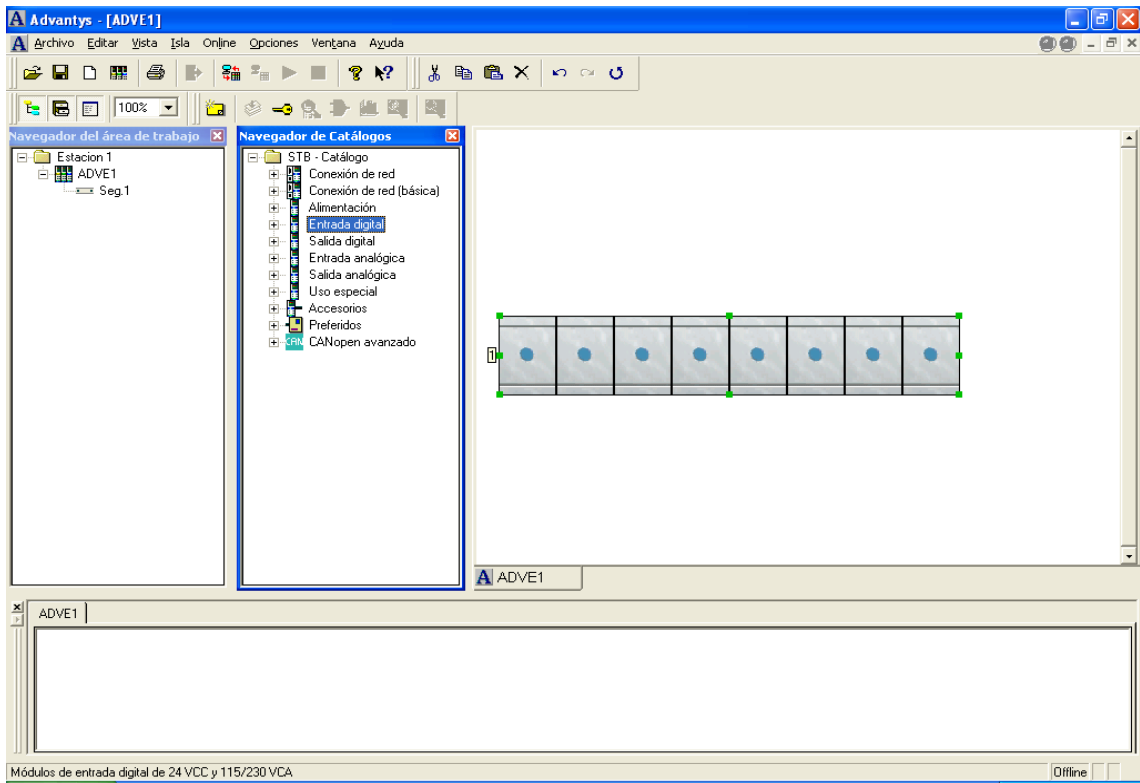


Fig. 101: Raíl Vacío de Isla

Una vez arrastrados todos los componentes hasta el raíl, la isla deberá tener un aspecto similar al de la siguiente imagen [Fig. 102]:

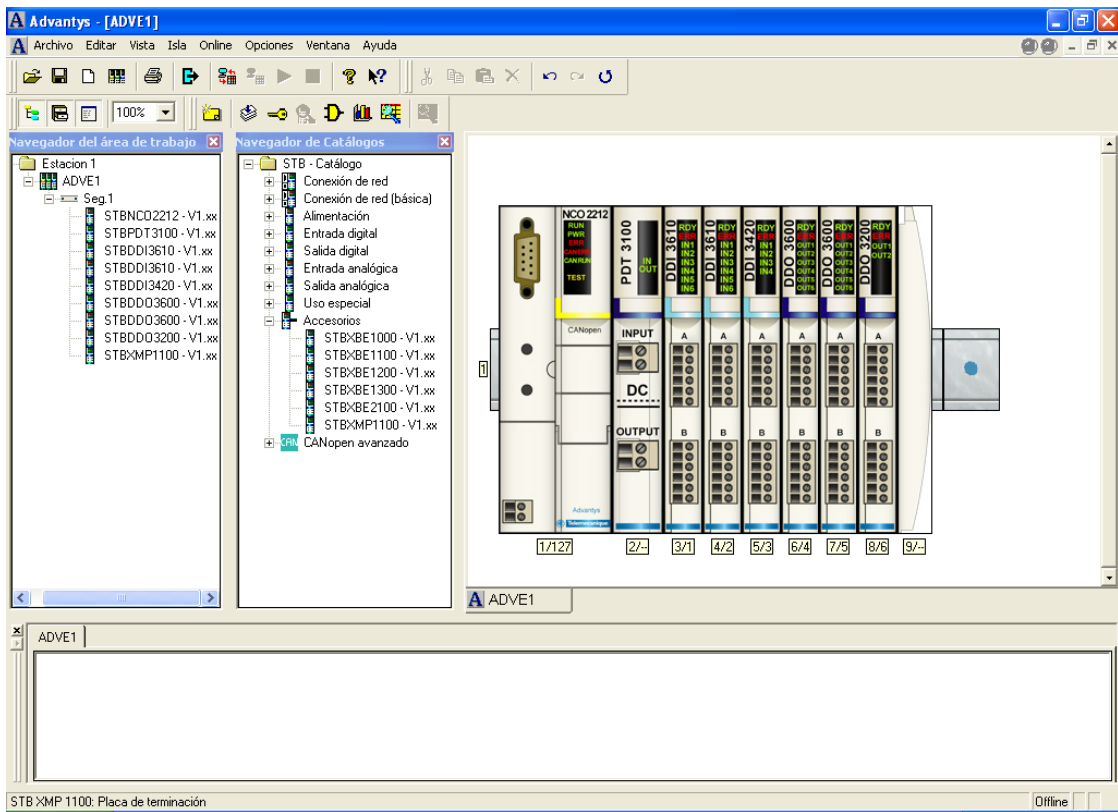


Fig. 102: Isla Advantys

- **Ejercicio:** diseñar una isla de trabajo Advantys.
 - Para familiarizarnos con el programa, simplemente construiremos la siguiente isla con los siguientes componentes y la guardaremos como 'Isla_Prueba'.

- Los componentes son los siguientes:

- Conexión de red: NCO2212-V1
- Alimentación: PDT3100-V1
- Entrada digital: DDI3610-V1
- Salidas digitales: STBDDO3600-V1
- Accesorios: STBXMP1100-V1

El siguiente paso, una vez añadidos los componentes, será asignar nombres a cada una de las variables que estén presentes en los módulos de entradas y salidas. Para ello, haremos doble click en cada uno de ellos y nos aparecerá el 'Editor de módulos'.

Nos aparecerá una ventana [Fig. 103] con información sobre el módulo distribuida en diversas pestañas. Iremos a la pestaña de 'Imagen de E/S' y en 'Etiqueta de usuario', iremos dando una etiqueta a cada dato y estado como se muestra en la siguiente imagen.

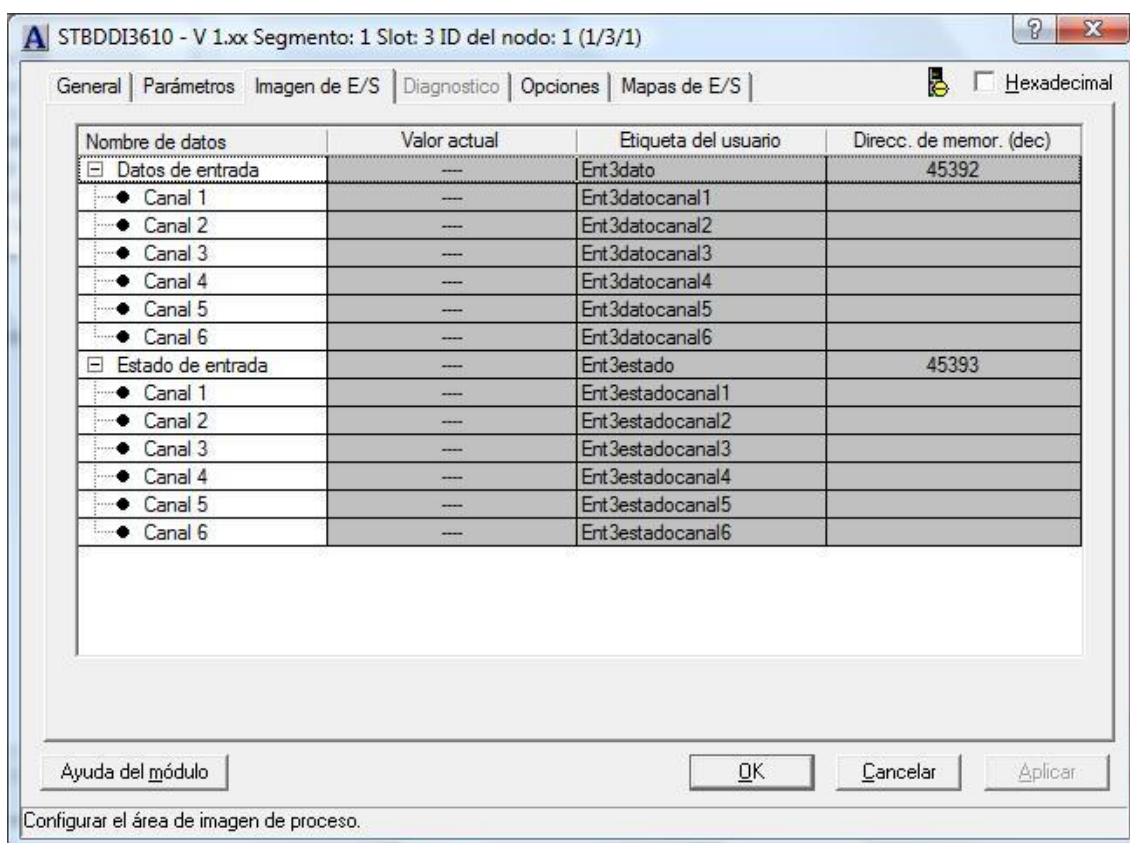


Fig. 103: Información del Módulo Seleccionado

Habrá que realizar esta misma operación con el resto de módulos de entrada/salida y una vez realizado esto guardaremos el área de trabajo.

La configuración de la isla ya estará realizada completamente por lo que solamente nos quedara exportarla. Para ello, iremos a la pestaña 'Archivo' y seleccionaremos la opción de 'Exportar'. [Fig. 104]

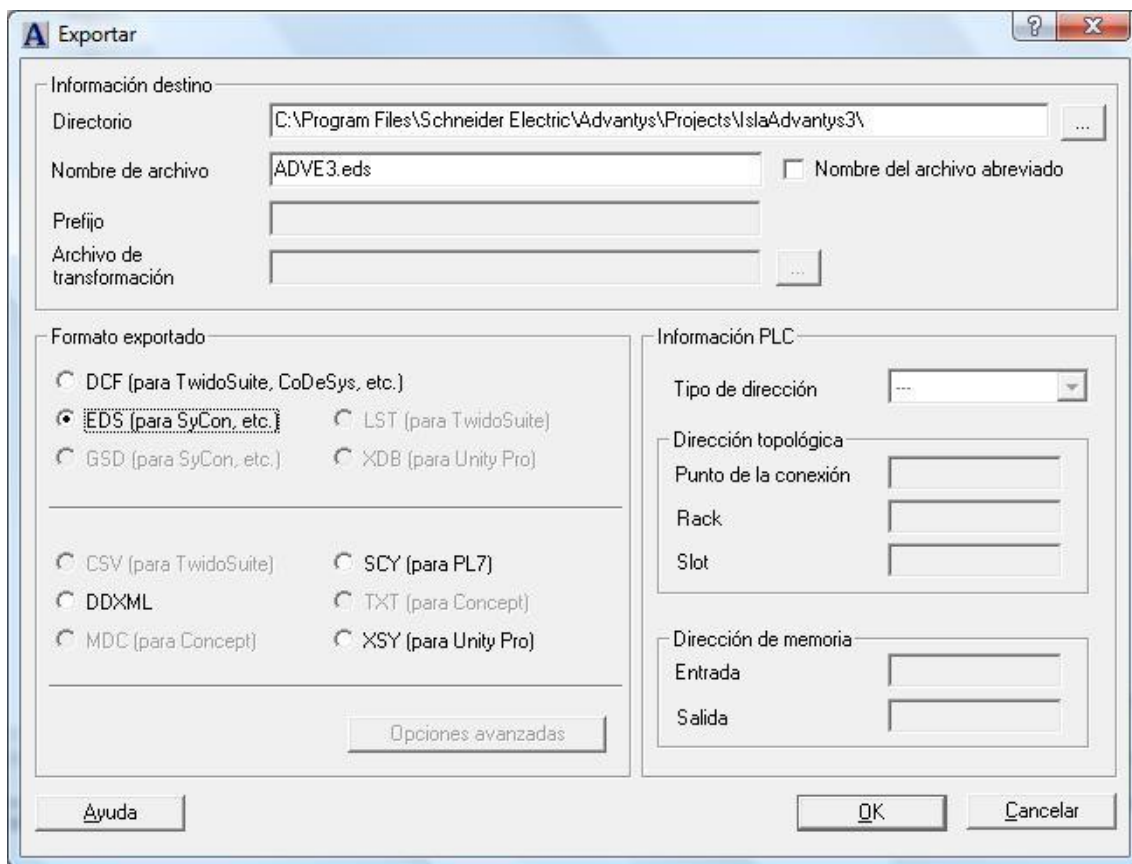


Fig. 104: Exportación de la Isla

Como se puede observar en la imagen anterior, seleccionaremos el Formato deseado para exportar el archivo. En nuestro caso, el formato que posteriormente utilizaremos para la creación de aplicaciones en SoMachine será EDS, por lo que lo seleccionamos y confirmamos la operación para terminar la exportación.

Una vez realizados todos estos pasos, ya tendremos el archivo EDS de configuración de la isla necesario para la programación y el posterior funcionamiento. Éste archivo es el necesario para la configuración de aplicaciones CanOpen en SoMachine.

4.2- CÓMO AGREGAR CONFIGURACIÓN ISLA ADVANTYS A SOMACHINE

Inicialmente, tendremos que instalar en la librería de SoMachine el archivo .EDS creado anteriormente para poder realizar las aplicaciones.

Para ello, lanzaremos el programa SoMachine y crearemos un nuevo proyecto como antes ha sido ya explicado. Una vez creado el nuevo proyecto, iremos a la sección del proyecto de programa y seleccionaremos la opción 'Repositorio de dispositivos' disponible en la pestaña de 'Herramientas' del menú general. [Fig. 105]

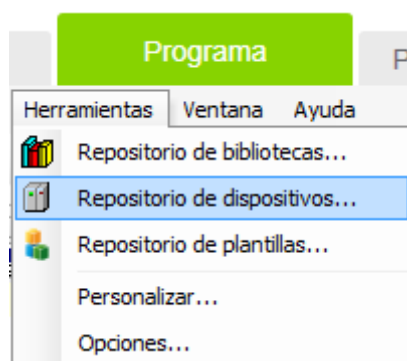


Fig. 105: Programa. Repositorio de dispositivos

En este momento, nos aparecerá una nueva ventana donde aparecen los dispositivos instalados en el programa. [Fig. 106]

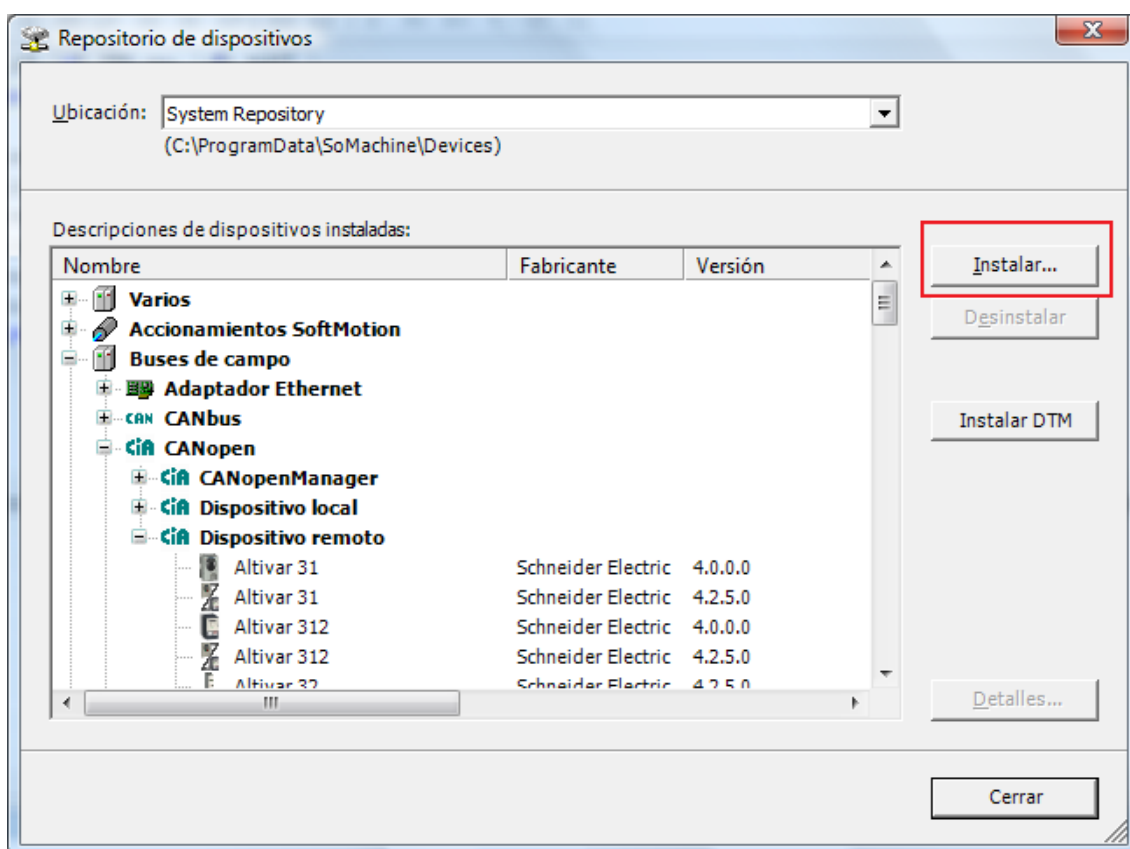


Fig. 106: Repositorio de Dispositivos

Haremos click sobre la opción 'Instalar' señalada en la imagen anterior y buscaremos el archivo 'Eds' que hemos creado anteriormente con Advantys. Una vez encontrado, confirmaremos la operación y el dispositivo habrá sido instalado apareciendo en la lista de dispositivos como se observa ahora en la imagen. [Fig. 107]

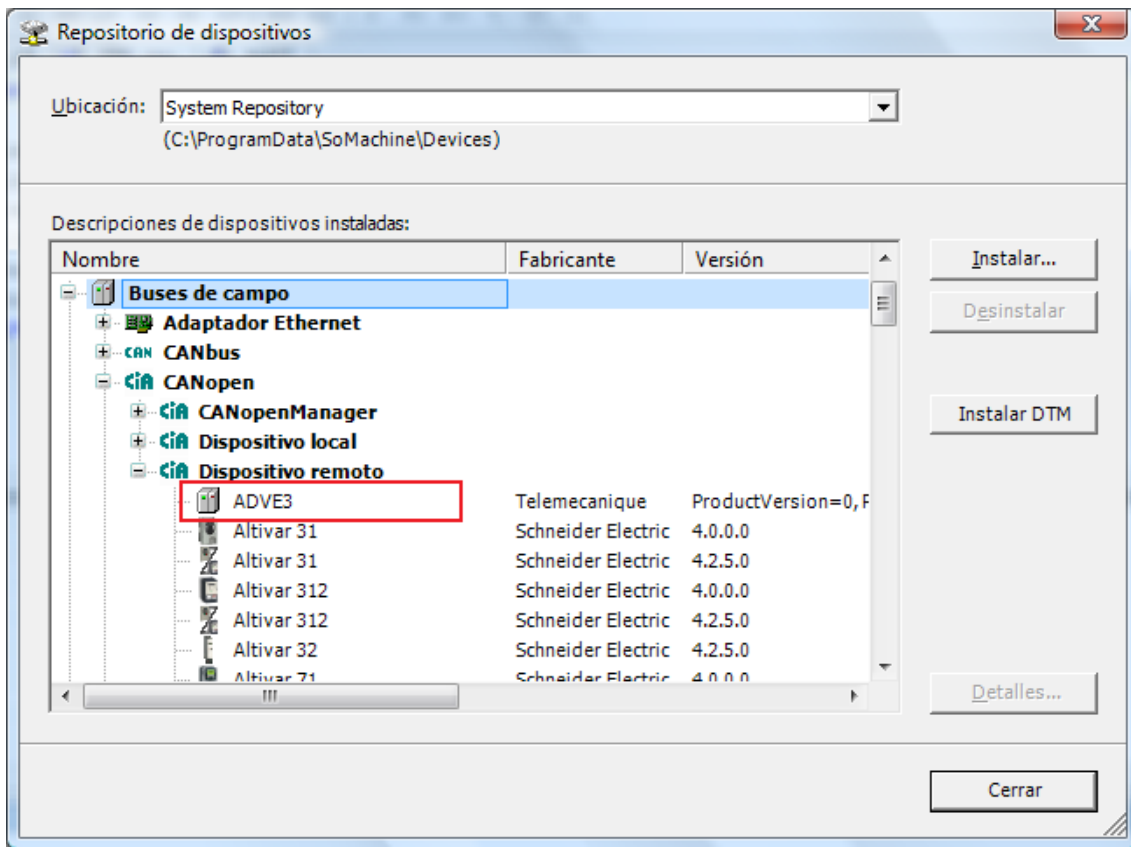


Fig. 107: Repositorio de Dispositivos. EDS instalado

Cerraremos dicha ventana y ya estará disponible la configuración de la isla en el catálogo de dispositivos CanOpen para ser añadido a la configuración hardware del proyecto. [Fig. 107]

5- PROGRAMACIÓN EN SFC

5.1- INTRODUCCIÓN. NORMA IEC 1131

El lenguaje SFC es uno de los seis lenguajes normalizados por la Comisión Electrotécnica Internacional (IEC).

Dicha Comisión, desarrolló el estándar IEC 1131 con el objeto de estandarizar los controladores programables. Uno de los principales objetivos de la Comisión fue crear un conjunto común de instrucciones para poder ser utilizado en todos los PLCs.

El estándar IEC 1131 consta de cinco partes:

1. Parte 1: Vista General
2. Parte 2: Hardware
3. Parte 3: Lenguajes de programación
4. Parte 4: Directrices de usuario
5. Parte 5: Comunicación

Las principales características de la norma IEC 1131-3 referida a los lenguajes de programación son:

- El software está estructurado a través de proyectos, tareas, programas y bloques.
- Presenta un fuerte tipado de datos a través de lenguajes que poseen operaciones a las cuales sólo se les puede aplicar un tipo de datos apropiado.
- El control de la ejecución se realiza a través de tareas.
- La descripción del comportamiento secuencial complejo de un proceso se realiza a través de SFC.
- El software está encapsulado a través de POUs (Unidades de Organización de Programas), estructuras y tipos complejos de datos.

Dicha norma puede dividirse en dos partes, elementos comunes y lenguajes de programación. La norma permite dos caminos en el desarrollo de un programa:

- I. De arriba a abajo: en primer lugar, configuración de los datos pensando en el proceso para pasar a elegir después el software de programación.
- II. De abajo a arriba: en primer lugar, se selecciona el software de programación más adecuado al proceso y posteriormente se definen los tipos de datos.

5.2- INTRODUCCIÓN AL LENGUAJE SFC.

5.2.1- Definición:

El lenguaje SFC (Gráfico Secuencial de Funciones) o también llamado Grafcet, es un lenguaje gráfico que proporciona una representación en forma de diagrama de las secuencias del programa.

Es un lenguaje basado en las redes de Petri que ofrece una metodología de programación con una estructura top-down es decir, de arriba a abajo.

Presenta diversas ventajas entre las que destacan:

- La minimización de los tiempos de diseño y depuración del programa.
- Una ejecución más rápida de las diferentes etapas.
- Permite el acceso directo al punto en donde la lógica de una máquina falla siendo más rápida y sencilla la solución de problemas.

5.2.2- Estructura:

Una secuencia de programa en SFC está compuesta de una serie de etapas representadas por cajas rectangulares conectadas entre sí por líneas verticales. [Fig. 108]

Cada una de estas etapas, representa un estado particular del sistema y a su vez, cada una de las líneas verticales que las conectan entre sí representa una transición de paso entre estados. Las transiciones están asociadas a una condición booleana de "True or False" mediante las cuales, se desactiva la etapa que las precede activando la etapa que viene a continuación.

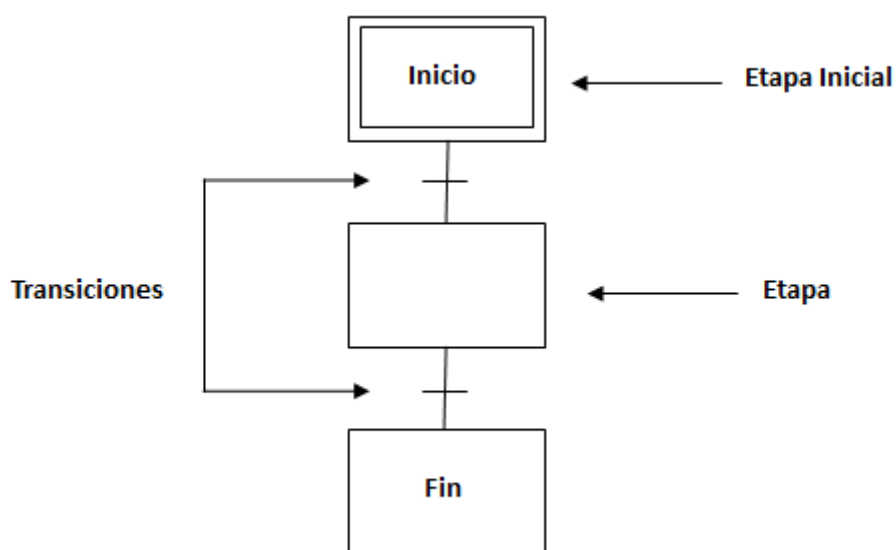


Fig. 108: Estructura Básica SFC

5.2.3- Elementos del lenguaje SFC:

El lenguaje SFC está compuesto por tres elementos básicos: las etapas, las acciones y las transiciones.

5.2.3.1- Etapas:

Representan estados del sistema y sólo pueden estar en dos estados, activas o no activas. En este último caso pueden encontrarse inactivas o ser activables. Hay dos tipos de etapas:

- ✓ Etapa normal
- ✓ Etapa inicial: aquella que queda activa al inicio del algoritmo de control.

5.2.3.2-Transiciones:

Representan la condición que da paso del control de una o varias etapas a la siguiente o siguientes.

Se representan por una línea horizontal que cruza la unión de las etapas y el resultado de dicha condición da como resultado una expresión booleana.

Las transiciones pueden estar formadas por una variable booleana, una dirección booleana o una constante booleana. Así mismo, puede estar formada por una serie de instrucciones con resultado booleano en lenguaje ST.

Por otro lado, no puede contener ningún programa, bloque de funciones o asignaciones.

5.2.3.3- Acciones:

Las acciones se indican con etiquetas conectadas a la etapa correspondiente y cada una de ellas sólo puede tener un único nombre.

Existen tres tipos de acciones:

- ✓ Expresiones booleanas
- ✓ Diagramas de relés
- ✓ Texto estructurado

Los posibles calificadores de asociación de las acciones son [Fig. 109]:

Calificador	Significado	Descripción
N	No almacenado	La acción permanece activa mientras la etapa esté activa.
R	Reset	Desactiva una acción que ha sido almacenada.
S	Set	La acción se activa y permanece en ese estado hasta un Reset.

Calificador	Significado	Descripción
L	Tiempo limitado	La acción se activa durante un tiempo determinado.
D	Tiempo de retardo	La acción se activa al cabo de un determinado tiempo o retardo si el paso sigue estando activo.
P	Pulso	La acción se ejecuta sólo una vez cuando se activa el paso.
SD	Almacenado y tiempo de retardo	La acción se activa al cabo de un determinado tiempo y permanece en ese estado hasta un Reset.
DS	Tiempo de retardo y almacenado	La acción se activa al cabo de un determinado tiempo si el paso sigue activo y permanece en ese estado hasta un Reset.
SL	Almacenado y tiempo limitado	La acción se desactiva durante un determinado tiempo.
P1	Pulso en flanco ascendente	La acción se ejecuta una sola vez cuando el paso se activa.
P0	Pulso en flanco descendente	La acción se ejecuta una sola vez cuando el paso se desactiva.

Fig. 109: Tabla Calificadores de Acción

5.2.3.4- Saltos:

Otro elemento importante del lenguaje SFC son los saltos. Son conexiones hacia el paso cuyo nombre se indica debajo del símbolo de salto. Los saltos son necesarios ya que no está permitido crear conexiones hacia arriba o entrecruzadas.

5.2.3.5- Otros Elementos: Divergencias y Convergencias

Disponemos de convergencias y divergencias cuando tenemos una o varias ramas en paralelo. Ambas pueden ser alternativas o simultáneas.

Cuando las divergencias son alternativas, se evalúa de izquierda a derecha la primera transición de cada rama paralela activándose la primera rama cuya condición de transición tenga un valor de TRUE. Cada rama en paralelo alternativa tiene que comenzar y terminar con una transición. [Fig. 110]

En cambio, si las divergencias son simultáneas, cuando se verifica la condición de la transición común, se activa la primera etapa de todas las ramas en paralelo. Cada una de estas ramas tiene que comenzar y terminar con un paso o etapa. [Fig. 111]

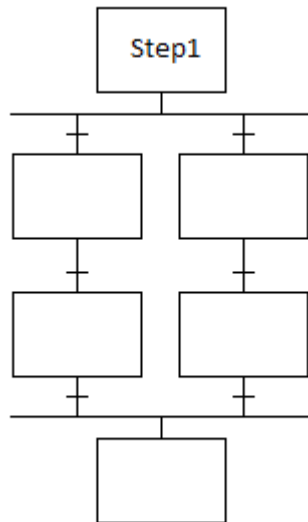


Fig. 110: Alternativa

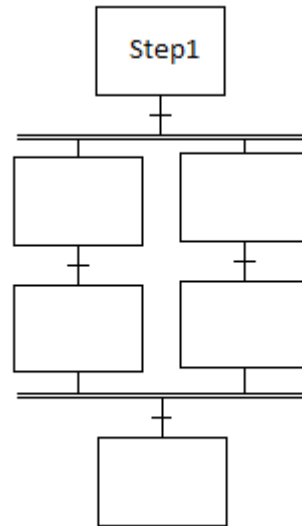


Fig. 111: Paralela

5.2.4- Reglas de evolución:

Además de la posibilidad de secuencias en paralelo y de macroetapas, existen tres reglas de evolución para los diagramas de estado:

- Regla 1: las etapas de inicialización se activan en forma incondicional al poner en marcha el sistema.
- Regla 2: una transición está validada sólo si todas las etapas inmediatas precedentes están activadas.
- Regla 3: cuando se verifica la condición de una transición, se produce la activación de todas las etapas siguientes inmediatas así como la desactivación de todas las etapas inmediatas precedentes.

5.3- PROGRAMACIÓN EN SFC EN SOMACHINE.

5.3.1- Creación POU:

Para crear una aplicación, habrá que crear una sección de trabajo dónde se escribirá el código del programa. Es la denominada POU. Para ello, habrá que hacer clic con el botón derecho sobre 'Application' en la ventana de dispositivos y seleccionar la opción POU que se encuentra en Agregar Objeto. [Fig. 50]

Como ya hemos visto anteriormente, SoMachine dispone de la posibilidad de programación en los seis lenguajes normados por el IEC. Además, existen tres tipos de POU: de programa, de bloque de funciones y de función.

El lenguaje de programación SFC sólo nos permite la creación de secciones de trabajo de tipo programa y de tipo bloque de funciones. No es posible crear una sección de tipo función con este lenguaje.

Por lo tanto, el siguiente paso será elegir el tipo de POU y el lenguaje de programación deseado, en nuestro caso Diagrama Funcional Secuencial (SFC). [Fig. 112]

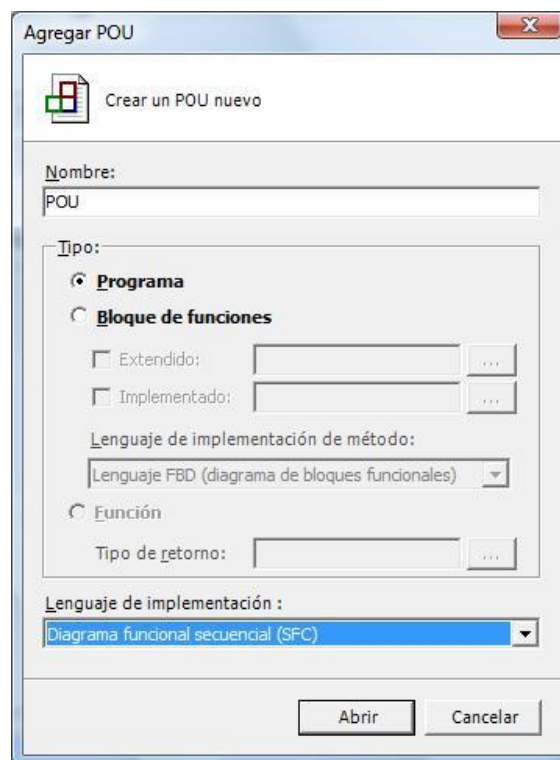


Fig. 112: Agregar POU en SFC

Una vez elegidos el tipo de POU de programa y el lenguaje de programación SFC, deberá aparecernos una nueva pantalla en la zona central dividida en dos partes, una de ellas con el nombre de la POU y la zona de declaración de variables, y otra con la etapa y transición iniciales. [Fig. 113]

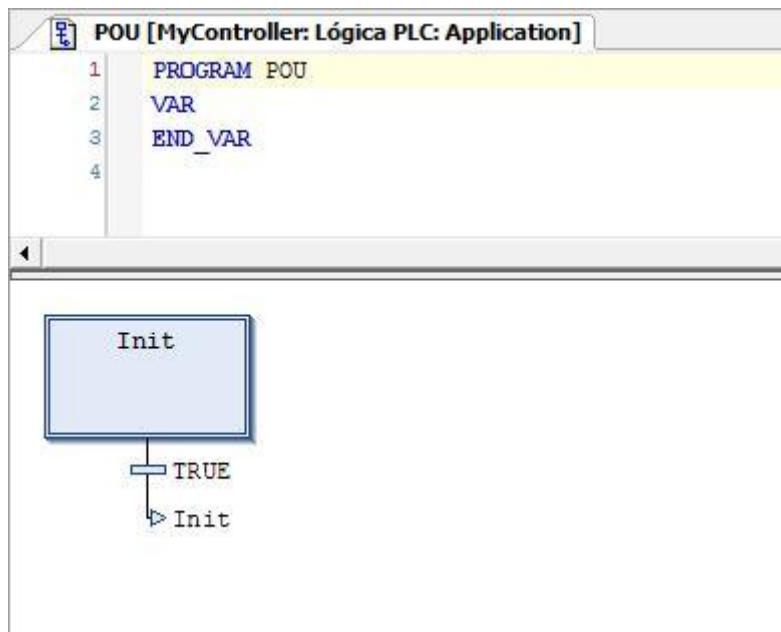


Fig. 113: Sección de Programa SFC

La etapa inicial es aquélla que se ejecuta una vez descargado el programa al controlador. Se diferencia del resto de etapas por su aspecto ya que se representa con un rectángulo doble.

Por defecto, esta primera etapa que nos aparece al crear la POU es de tipo paso inicial pero para establecer como paso inicial cualquier otra etapa basta con hacer clic con el botón derecho sobre ella y hacer clic en la opción paso inicial.

La etapa que hasta ese momento tuviéramos como inicial pasará a ser de tipo normal inmediatamente ya que no puede haber dos pasos iniciales. El mismo programa automáticamente la establece de tipo normal.

5.3.2- Diseño y funcionalidades:

Para la programación en SFC, SoMachine dispone de una completa barra de herramientas con los diferentes elementos disponibles. [Fig. 114]

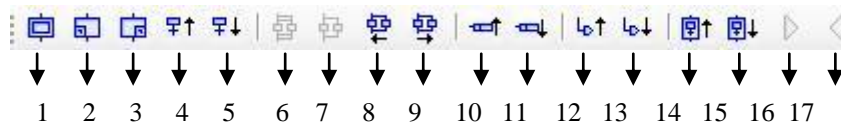


Fig. 114: Barra de Herramientas SFC

Los diferentes elementos de la barra de herramientas son:

1. **Paso inicial:** si seleccionamos una etapa y pulsamos sobre paso inicial, dicha etapa pasará a ser la inicial. [Fig. 115]

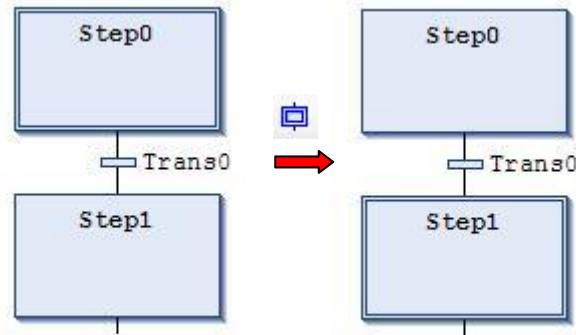
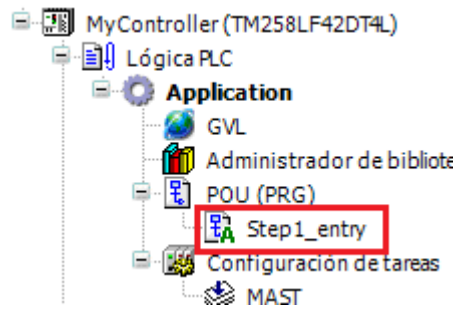


Fig. 115: Establecer Paso Inicial

2. **Agregar acción de entrada:** cuando pulsemos sobre dicho botón, nos aparecerá una nueva ventana para seleccionar el nombre y el lenguaje de programación de dicha acción. Una vez elegidos, nos aparecerá una nueva pantalla para programar dicha acción.



Una vez programada la nueva acción, podremos verla en la ventana de dispositivos.

[Fig. 116]

Fig. 116: Acción de Entrada

Si seleccionamos la etapa correspondiente, veremos la acción en 'Paso activado' en la ventana de Propiedades de la derecha de la pantalla. [Fig. 117]

Propiedad	Valor
Generalidades	
Nombre	Step1
Comentario	
Símbolo	
Específico	
Paso inicial	<input type="checkbox"/>
Tiempos	
Mínimo activo	
Máximo activo	
Acciones	
Paso activo	
Paso activado	Step1_entry
Paso desactivado	

Fig. 117: Propiedades. Acción de Entrada

3. **Agregar acción de salida:** al igual que cuando agregamos una acción de entrada, nos aparecerá una nueva ventana para elegir el nombre y el lenguaje de implementación de la misma. Dicha acción aparecerá en la ventana de propiedades como acción de 'Paso desactivado'. [Fig. 118]

Propiedad	Valor
Generalidades	
Nombre	Step1
Comentario	
Símbolo	
Específico	
Paso inicial	<input type="checkbox"/>
Tiempos	
Mínimo activo	
Máximo activo	
Acciones	
Paso activo	
Paso activado	
Paso desactivado	Step1_exit

Fig. 118: Propiedades. Acción de Salida

Así mismo, cualquier tipo de acción puede agregarse a nuestra sección de programa principal haciendo clic con el botón derecho sobre el nombre la POU en la ventana de dispositivos y en agregar objeto seleccionar la opción de 'Acción'. Además, aunque no aparece un botón explícito en la barra de herramientas, pueden añadirse a las etapas acciones de 'Paso activo' es decir, acciones que se ejecuten mientras la etapa está activa. [Fig. 122]

Para añadir una acción creada ya anteriormente a una etapa y que ésta sea de 'Paso activo', basta con seleccionar dicha etapa y en el menú de propiedades de la derecha añadirla en Acciones → Paso activo. [Fig. 119]

Propiedad	Valor
Generalidades	
Nombre	Step1
Comentario	
Símbolo	
Específico	
Paso inicial	<input type="checkbox"/>
Tiempos	
Mínimo activo	
Máximo activo	
Acciones	
Paso activo	...
Paso activado	
Paso desactivado	

Acciones	
Paso activo	Action_1
Paso activado	
Paso desactivado	

Hay que pulsar sobre el botón para buscar la acción deseada

Fig. 119: Cómo Añadir una Acción

Las etapas con acciones de paso activado (entrada) [Fig. 120], paso desactivado (salida) [Fig. 121] y paso activo [Fig. 122] presentarán el siguiente aspecto:

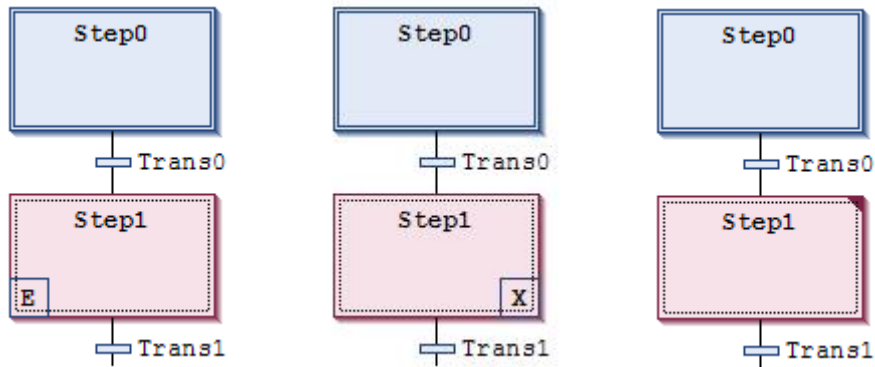


Fig. 120: Paso Activado

Fig. 121: Paso Desactivado

Fig. 122: Paso Activo

4. **Insertar la transición de paso antes:** seleccionando una transición, esta opción introduce una transición y una etapa antes de dicha transición. [Fig. 123]
5. **Insertar la transición de paso después:** con esta opción insertamos la etapa y la transición después de la transición seleccionada. [Fig. 124]

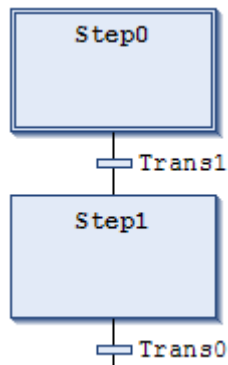


Fig. 123: Transición Antes

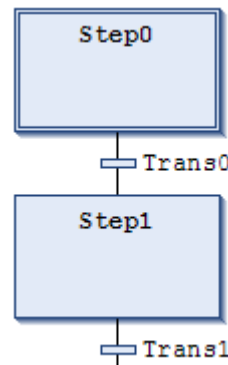


Fig. 124: Transición Después

A las transiciones se les puede asociar operaciones en otros lenguajes de programación como son LD, IL y ST. Así mismo, pueden utilizarse etiquetas específicas para incluir condiciones temporales en el programa o controlar el tiempo que una etapa está activa. [Fig. 125]

**Ejemplo: step0.t >= t#2s → la etapa permanecerá activa durante dos segundos.*



Fig. 125: Ejemplo Temporal

-
6. **Rama paralela:** esta opción sólo está activa cuando seleccionamos una rama. La primera etapa de cada rama conectada a esta se activará simultáneamente cuando se cumpla la condición de la transición general que las precede anteriormente. Se representa con dos líneas horizontales. Puede dársele una etiqueta. [Fig. 126]

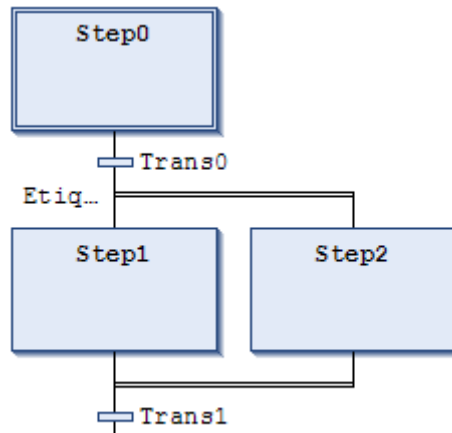


Fig. 126: Rama Paralela

7. **Rama alternativa:** esta opción sólo se encuentra activa cuando seleccionamos una rama. Se activará la etapa de la rama que su transición se cumpla. A diferencia de la rama paralela, se representa con una sola línea horizontal. [Fig. 127]

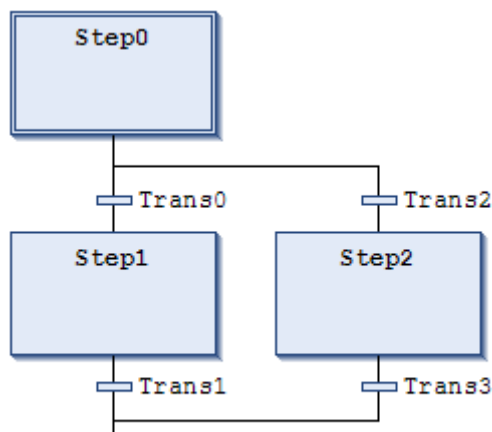


Fig. 127: Rama Alternativa

8. **Insertar rama a la izquierda:** cuando pulsamos sobre esta opción, insertamos una rama alternativa a la izquierda de la etapa seleccionada. [Fig. 128]

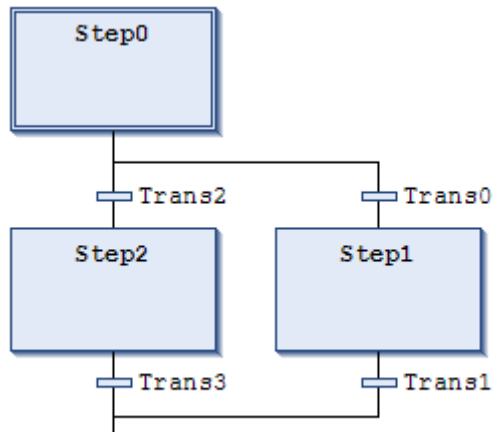


Fig. 128: Rama a Izquierda

9. **Insertar rama a la derecha:** cuando pulsamos sobre esta opción, insertamos una rama alternativa a la derecha de la etapa seleccionada. [Fig. 129]

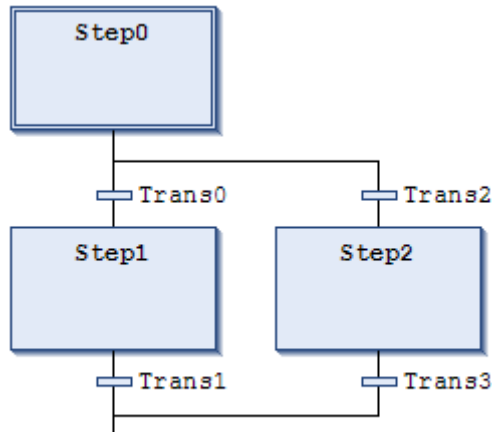


Fig. 129: Rama a Derecha

10. **Insertar asociación de acción antes:** al seleccionar una etapa, insertamos una asociación de acción encima de la que ya teníamos insertada. [Fig. 130]

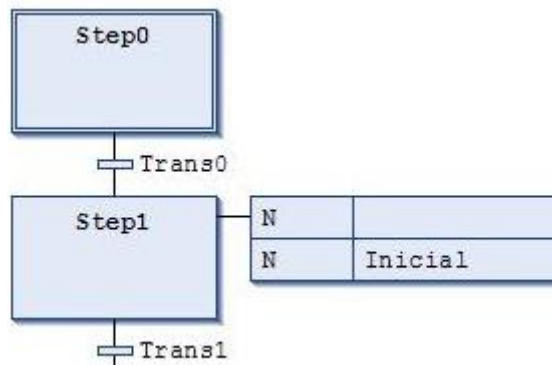


Fig. 130: Asociación Acción Antes

11. Insertar asociación de acción después: al seleccionar una etapa, insertamos una asociación de acción debajo de la que ya teníamos insertada. [Fig. 131]

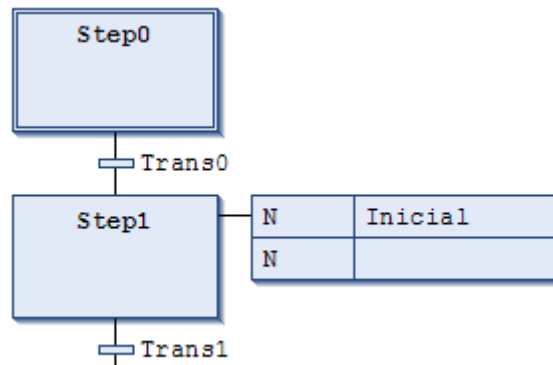


Fig. 131: Asociación Acción Después

Las acciones son uno de los elementos más importantes del lenguaje SFC. Su diseño, consta de dos partes [Fig. 132]:

- ✓ el calificador (recuadro de la izquierda)
- ✓ etiqueta de la acción a activar (recuadro de la derecha)

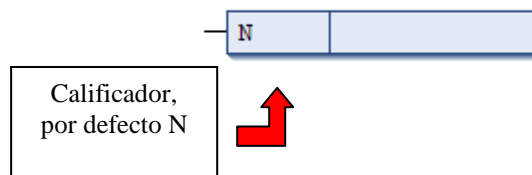


Fig. 132: Diseño Acción

Al pulsar sobre el recuadro de la izquierda, correspondiente al calificador, se nos abrirá una nueva ventana [Fig. 133] con los diferentes calificadores disponibles para dicha acción.

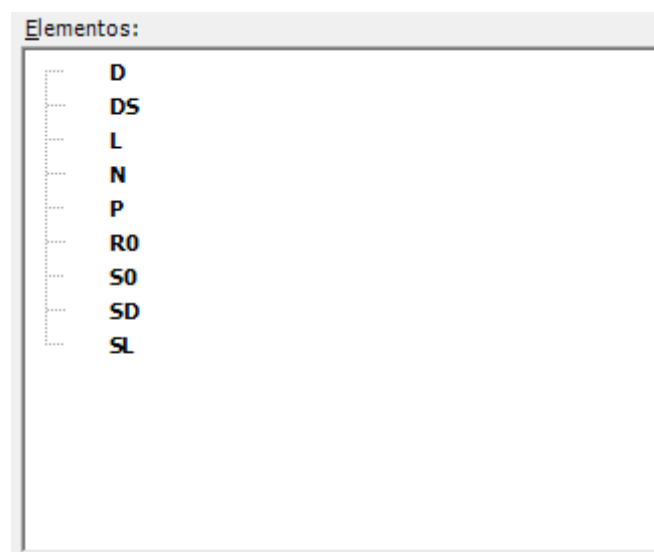


Fig. 133: Calificadores de Acción

Si sobre el recuadro correspondiente a la acción introducimos una variable no definida, saltará una ventana para definirla correctamente. [Fig. 65]

12. Insertar salto antes: al elegir esta opción, insertamos un salto antes de la etapa o transición seleccionada.

13. Insertar salto después: al elegir esta opción, insertamos un salto después de la etapa o transición seleccionada.

Los saltos son elementos fundamentales que nos permiten pasar a otras partes del programa sin tener que pasar por todas las etapas del programa que se encuentran por encima o por debajo de ésta.

La etiqueta nos indica a que etapa salta y su representación es la siguiente. [Fig. 134]

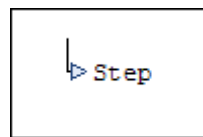


Fig. 134: Salto

Para ver con mayor claridad la función de un salto en un programa, realizaremos el siguiente ejemplo [Fig. 135]:

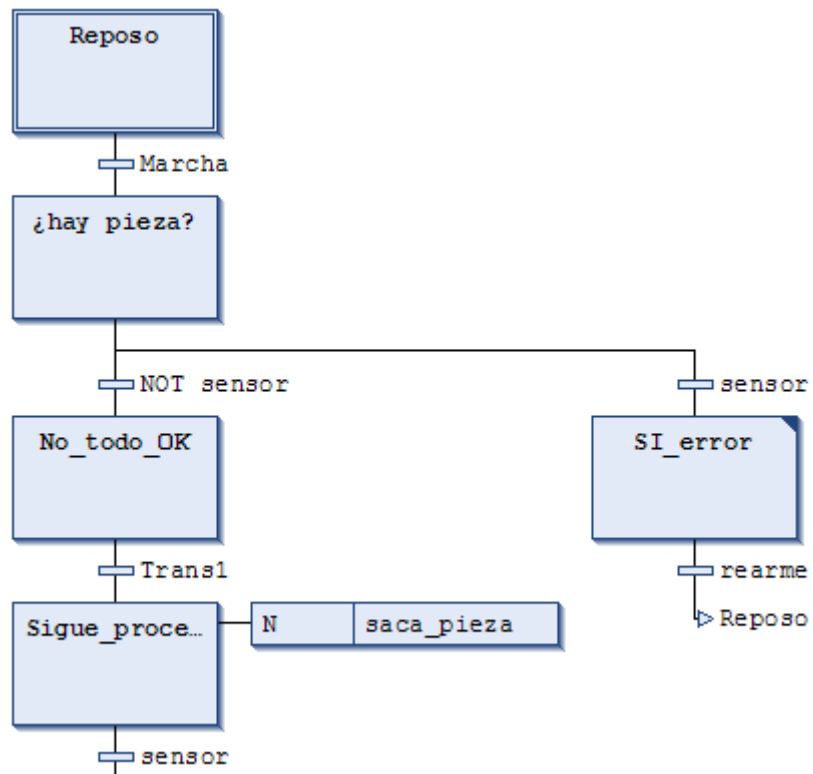


Fig. 135: Ejemplo de Programación con Salto

La máquina se encuentra en estado de reposo. Una vez pulsado el pulsador de marcha comprueba si hay una pieza sacada o no mediante el sensor correspondiente. Si no hay una pieza sacada, todo está correcto y sigue el proceso de fabricación.

Si por el contrario había una pieza sacada, la máquina detecta un error y llamará a un estado de emergencia del que no saldrá hasta que se solucione el problema y el operario pulse el botón de rearme. Una vez realizado esto, la máquina volverá a su estado de reposo marcado por la condición de salto.

***Nota:** destacar que NO pueden utilizarse símbolos tales que exclamaciones o signos de interrogación al nombrar las etapas ya que al intentar compilar el programa nos aparecerá un mensaje de error. Este simplemente es un ejemplo guía para explicar la función de los saltos gráficamente.

14. Insertar macro antes: con esta opción insertamos una macro antes de la transición o etapa seleccionada. [Fig. 136]

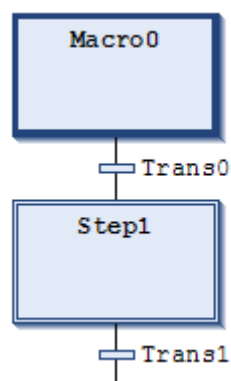


Fig. 136: Macro Antes

15. Insertar macro después: con esta opción insertamos una macro después de la transición o etapa seleccionada. [Fig. 137]

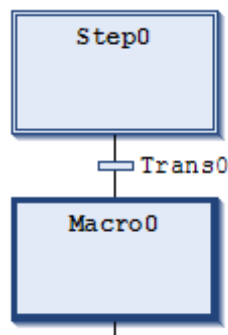


Fig. 137: Macro Después

Cuando hablamos de macros estamos hablando de macro etapas, es decir etapas que pueden contener dentro un programa completo de una estación de trabajo. Estas macro etapas son útiles para cadenas de producción en las que hay varias estaciones de trabajo distintas. Con las macros se pueden dividir los programas en distintas partes y dentro de cada macro programar una estación de trabajo distinta.

-
- 16. *Mostrar macro:*** nos muestra la macro. Sólo está activa dicha opción cuando seleccionamos una macro. [Fig. 138]
- 17. *Salir de la macro:*** sale de una macro cuando estamos dentro de ella. Sólo está activa dicha opción cuando seleccionamos una macro

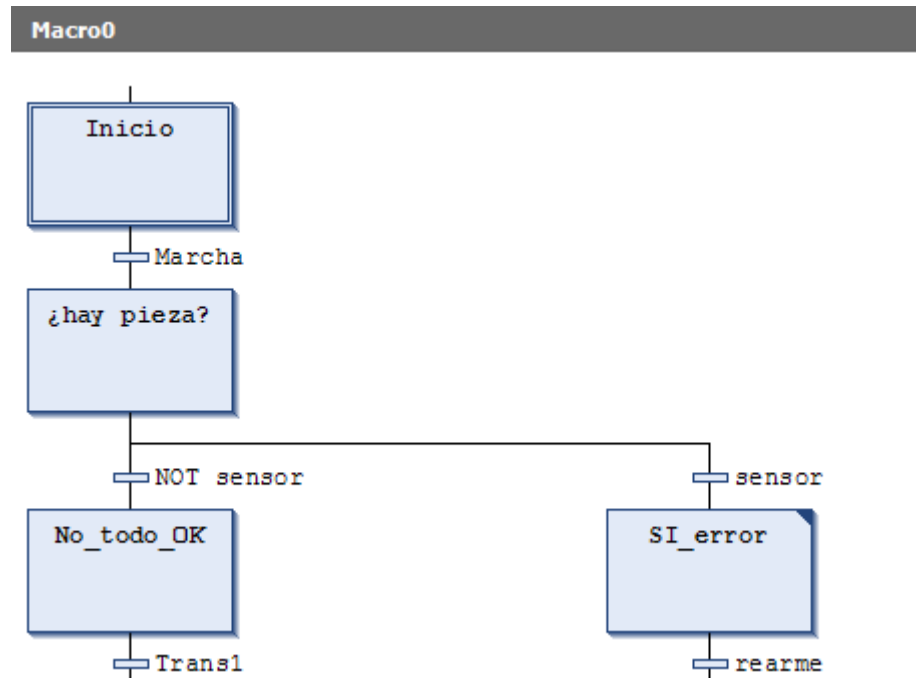


Fig. 138: Macro

Para finalizar, decir que todos los elementos anteriormente citados así como todas las funcionalidades que nos permite el software SoMachine son los que componen la programación básica en SFC y una vez conocidos, la programación en dicho lenguaje resulta sencilla además de muy visual.

6- INTRODUCCIÓN A LA CÉLULA DE FABRICACIÓN

6.1- DESCRIPCIÓN GENERAL DE LA CÉLULA

En primer lugar, antes de analizar las distintas estaciones que componen la célula de fabricación, realizaremos una descripción del funcionamiento de ésta.

Se trata de una célula de producción de cilindros en serie, situada en el edificio Ada Byron del Campus Río Ebro la cual, está compuesta por dos módulos principales y uno de unión.

- ✓ El módulo I es el encargado de la fabricación y verificación de los cilindros.
- ✓ El módulo II es el encargado del montaje de los pedidos y de su almacenamiento final.
- ✓ El módulo de unión se encarga de la identificación y clasificación así como del almacenamiento y suministro de pedidos hacia el modulo II.

La célula de fabricación está formada por un conjunto de siete estaciones cuya finalidad de todas ellas en conjunto es la fabricación y expedición de una determinada mercancía, constituida en este caso por un conjunto de tres cilindros neumáticos sobre una determinada base.

El proceso de fabricación de la célula comienza mediante la solicitud de un determinado pedido de fabricación. Un pedido, estará compuesto por un máximo de 3 piezas depositadas sobre una base concreta. En lo referente al tipo de base donde depositar las piezas a expender, existen 2 posibilidades mientras que con el tipo de pieza, disponemos de 6 posibilidades diferentes.

En la siguiente imagen [Fig. 139] puede observarse un ejemplo de fabricación ya completado.

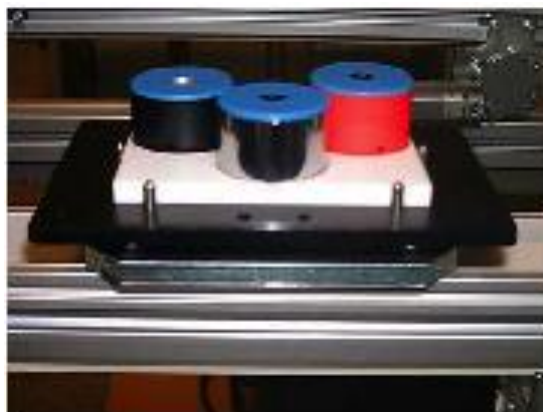


Fig. 139: Pedido Completo

6.1.1- Características del producto a fabricar:

La misión final de la célula de fabricación consiste en expender un pedido que haya sido solicitado a fabricar.

Como ya hemos dicho anteriormente, el producto final está compuesto por una base y unos cilindros.

Si hacemos referencia a los tipos de bases donde depositar los cilindros, existen dos cuya única característica que los diferencia es el color. Por tanto, tenemos la posibilidad de depositar las piezas sobre una base de color blanco o sobre una base de color negro.

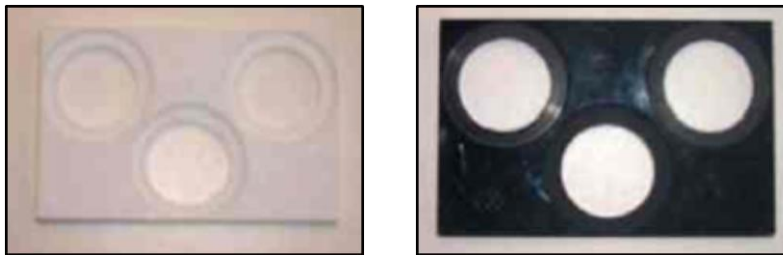


Fig. 140: Bases para los pedidos

Si atendemos ahora a los diferentes tipos de piezas que son posibles de expender, podemos dividirlos en dos grupos diferentes. Por un lado tenemos los cilindros neumáticos y por otro los cilindros cerrados o con tapa. Existen seis posibilidades de producción. [Fig. 141]







	<u>Negra</u>	<u>Roja</u>	<u>Metálica</u>
Piezas con Tapa			
Piezas sin Tapa (cilindros neumáticos)			

Fig. 141: Tabla de Posibilidades de Producción

Los dos tipos de piezas se diferencian en su ensamblaje, es decir, en las partes que las componen. Por lo tanto, dependiendo del tipo de pieza que se desee fabricar, el proceso productivo será uno u otro.

6.1.2- Composición de las piezas a fabricar:

6.1.2.1 Cilindros Neumáticos:

Como ya hemos visto anteriormente, las piezas a fabricar simulan cilindros neumáticos de simple efecto. Cada una de las piezas estará formada por los siguientes elementos:

- **Camisa:** esta pieza constituye la parte exterior o carcasa del cilindro neumático a fabricar y será el soporte del resto de las piezas a colocar en el proceso de producción.
- **Émbolo:** será el encargado de provocar el desplazamiento del eje al inyectar aire comprimido sobre el orificio destinado a tal efecto en la camisa. El diseño de los mismos evitará que el aire se escape a la atmósfera gracias a una junta que posee en su interior.
- **Muelle:** dado que los cilindros neumáticos a fabricar son de simple efecto, deberemos de provocar el retorno del émbolo a su posición de origen una vez cortado el flujo de aire comprimido. Por lo tanto, la misión del muelle es proporcionar dicha fuerza de retorno del émbolo para que éste se recoja.

6.1.2.2 Tapa:

Se debe colocar una tapa a las piezas a fabricar para que las piezas internas del cilindro neumático no abandonen su situación al inyectar el aire comprimido. Por lo tanto, la camisa del cilindro neumático deberá de estar cerrada. [Fig. 142]

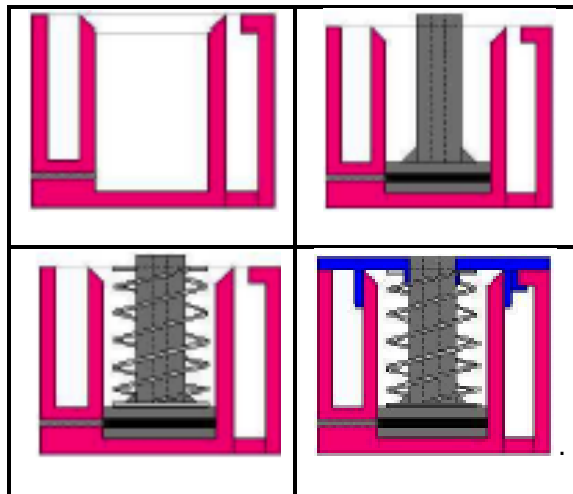


Fig. 142: Proceso Colocación Tapa

6.1.2.3 Piezas sólo con Tapa

Las piezas con tapa son el otro tipo de piezas que la célula permite fabricar. Estas piezas están formadas únicamente por la camisa y una tapa que no posee orificio para la extensión del émbolo.

En estas piezas no será necesaria la colocación del émbolo y el muelle ya que tenemos una pieza compacta en la que camisa y tapa se encuentran sólidamente colocadas.

6.1.3- Características de los elementos de las piezas:

Cada uno de los tres tipos de piezas a fabricar tiene unas características propias que los diferencian entre sí. Estas piezas se diferencian en su color y tamaño.

Si atendemos al tamaño de las piezas, todas disponen del mismo diámetro pero no así de la misma altura. Las camisas de color negro tienen una altura menor que las rojas y las metálicas por lo que los émbolos de dichas camisas deberán tener una longitud más corta. Para diferenciar los dos tipos de émbolos, unos serán de color metálico y otros negro. Los émbolos de menor longitud son los de color metálico.

Como puede comprobarse, ésta es una de las características que hacen que la célula sea flexible, es decir, es capaz de fabricar diferentes tipos de piezas de forma aleatoria y sin incorporar elementos adicionales.

Por otro lado, los muelles serán todos iguales difiriendo únicamente en la compresión del muelle dentro de cada una de las camisas.


Tipo de Pieza			
Camisa	Negra	Roja	Metálica
Émbolo	Metálico(corto)	Negro(largo)	Negro(largo)
Muelle	Estándar		

Fig. 143: Tabla Composición Piezas

6.2- **CARACTERÍSTICAS GENERALES DE LAS ESTACIONES**

La fabricación de las piezas anteriormente descritas es realizada por medio de la acción conjunta de todas las estaciones que componen la célula de fabricación flexible. [Fig. 144]

Cada una de las estaciones que conforman la célula de fabricación flexible posee una función determinada y concreta.

La fabricación de las piezas se realiza teniendo en cuenta el mantenimiento de un pequeño stock dentro de la fábrica para poder atender los posibles pedidos. Así en el almacén de piezas intermedio tendremos acumuladas un pequeño número de piezas para

poder servir pedidos, ordenando la fabricación cuando el número de piezas de un determinado tipo sea inferior a un margen de seguridad prefijado.

Por el contrario la formación de palets con mercancía se lleva a cabo por medio del lanzamiento de pedidos de fabricación a la célula de fabricación. Cuando se solicite un determinado pedido la célula de fabricación realizará las operaciones apropiadas para componerlo y ser capaz de servirlo.

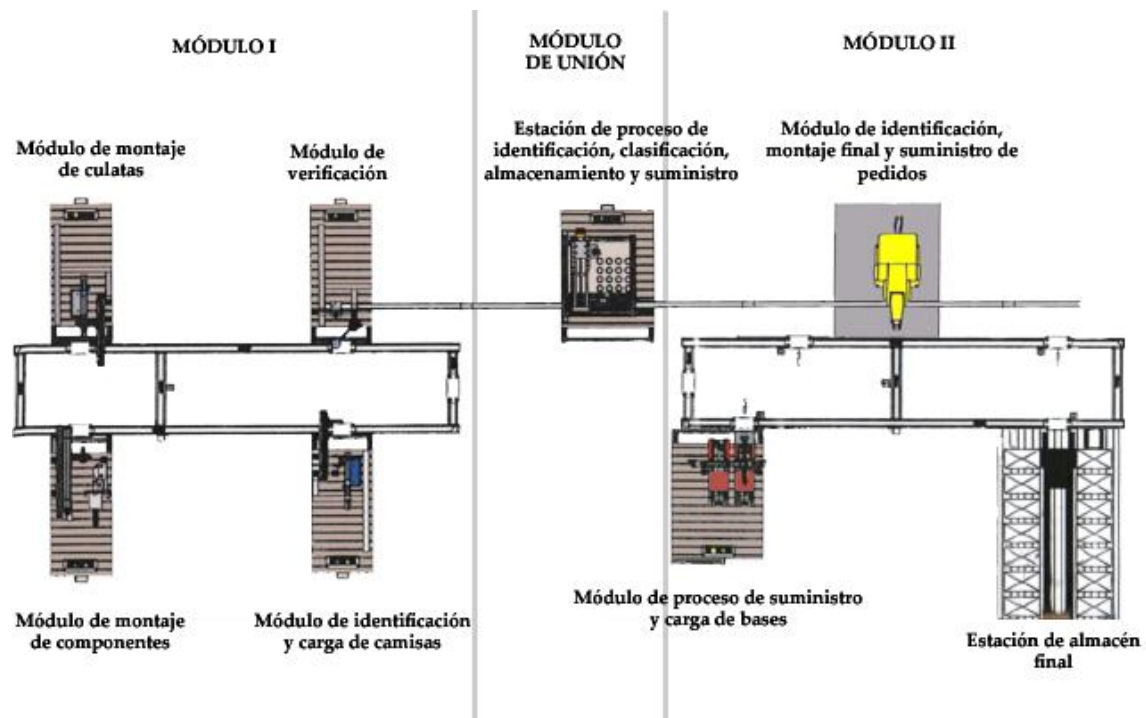


Fig. 144: Célula de Fabricación Completa

En la célula de fabricación podemos distinguir tres zonas diferentes.

- ✓ **Zona de fabricación:** en dicha zona se realizará el montaje de cada uno de los tipos de pieza solicitados por el almacén intermedio.
- ✓ **Zona del almacén intermedio:** en esta zona se depositarán las piezas fabricadas hasta que se solicite la expedición de las mismas en un determinado pedido.
- ✓ **Zona de expedición:** en dicha zona se realizarán las operaciones necesarias para la composición del pedido solicitado.

Todas estas zonas se encuentran coordinadas por medio de un sistema que podríamos denominar como gestor o coordinador. Dicho elemento es el encargado de decirle a cada una de las estaciones la operación que debe realizar en cada momento para llevar a cabo la fabricación completa y efectiva de cada uno de los tipos de piezas.

6.2.1- Zona de Fabricación:

En esta zona tenemos situadas las estaciones necesarias para llevar a cabo el ensamblado de los diferentes elementos que componen la pieza a fabricar en cada momento.

La zona de fabricación está formada por cuatro estaciones y el transporte.

- **Transporte:** es el encargado del traslado de las piezas de una estación a otra por medio de un trasbordador.



Fig. 145: Transporte Zona I

- **Estación 1:** es la estación encargada de la colocación de la camisa deseada en el palet del trasbordador.



Fig. 146: Estación 1

-
- **Estación 2:** es la estación encargada de la colocación del émbolo y el muelle en las piezas que así lo requieran. Por lo tanto, se encargará de colocar el émbolo adecuado para el tipo de pieza fabricada en cada momento así como el muelle que llevan dichos elementos.

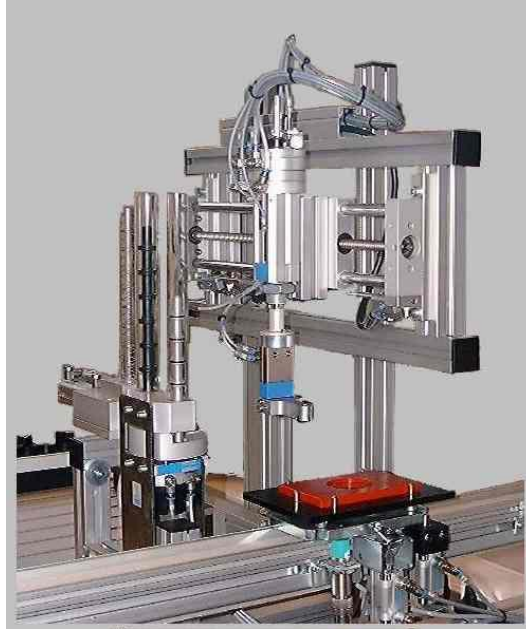


Fig. 147: Estación 2

- **Estación 3:** en esta estación se coloca y rosca la tapa a las piezas que así lo requieran. Por lo tanto, todas las piezas que lleguen a esta estación con el émbolo y el muelle debidamente colocados deberán de ser tapadas mediante la culata mencionada

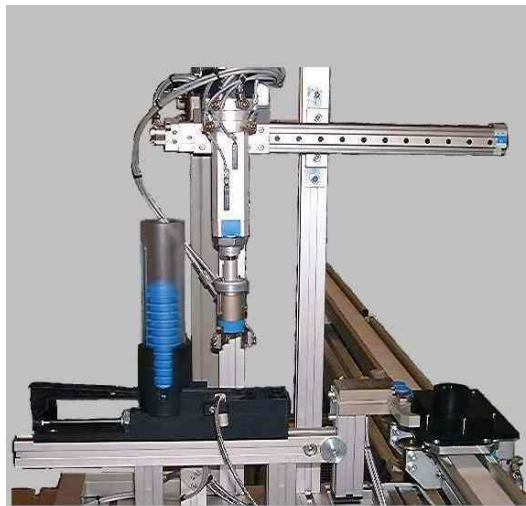


Fig. 148: Estación 3

- **Estación 4:** esta estación es la encargada de la verificación de las piezas para comprobar que han sido fabricadas correctamente desechando las defectuosas antes de dar por finalizada la fabricación de una determinada dicha pieza.

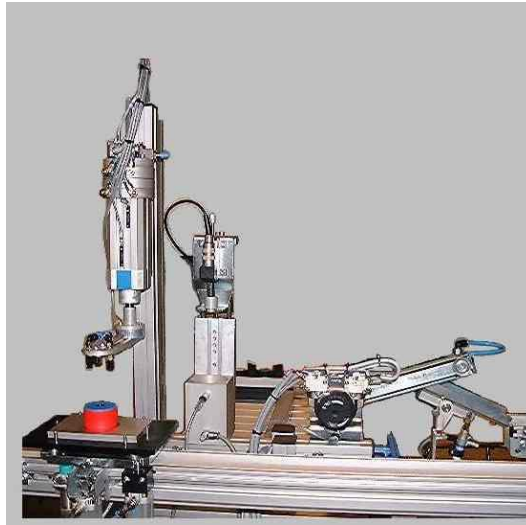


Fig. 149: Estación 4

6.2.2- Zona de Almacén Intermedio:

Entre la zona de transporte y la zona de expedición tenemos un almacén en el cual se guardarán las piezas ya fabricadas. Éstas serán mantenidas en él hasta el momento en el cual se desee cargar dichas piezas en un determinado palet para así poder servir un determinado pedido dentro de la zona de expedición.

La fabricación de las piezas se realiza de una en una, por lo que la finalización de las mismas no se realizará consecutivamente sino que de manera secuencial.

Así mismo, podemos tener activos varios pedidos en la zona de expedición. Las piezas que componen dichos pedidos no tienen por qué llegar al mismo tiempo que el transbordador hasta el punto de carga de piezas por lo que es necesaria la existencia de una zona de almacenaje donde mantener las piezas almacenadas hasta el momento de colocarlas en un palet.

De este modo cuando debamos colocar una pieza sobre el palet del pedido podremos realizar la operación si dicha pieza se encuentra en la zona del almacén intermedio.

Esta función será realizada por una única estación:

- **Estación 5:** esta estación será la encargada de realizar el almacenaje de las piezas sueltas ya fabricadas hasta el momento de expedición de las mismas. Recibirá las piezas desde la cinta transportadora de salida de la estación de verificación y las entregará en otro tramo que la conducirá hasta la zona de expedición de pedidos.

El almacén intermedio posee 16 posiciones sobre las cuales colocará las piezas para mantenerlas en él durante el tiempo necesario. Dichas posiciones se encuentran distribuidas en una matriz plana de 4 filas y 4 columnas.

Para el desplazamiento de las piezas desde la entrada hasta la matriz de posiciones y desde ésta última a la salida de la estación se dispone de dos motores paso a paso. De este modo, y por medio de 2 ejes perpendiculares de desplazamiento, podemos acceder hasta cualquiera de los puntos de dicho almacén intermedio.

Las piezas podremos cogerlas y depositarlas por medio de una ventosa de succión, que hace vacío sobre las mismas y las sujeta, mientras que por medio de un cilindro neumático podremos alzar y bajar las piezas para así poder realizar de forma segura las traslaciones de las mismas.



Fig. 150: Estación 5

Además de dicha función de almacenaje intermedio de las piezas para servir los pedidos activos en cada momento, el autómatas de esta estación junto con todos los periféricos controlados por él es el encargado de la labor de gestión y coordinación de todos los procesos de fabricación.

6.2.3- Zona de Expedición:

Por último, en esta zona tenemos situadas las estaciones necesarias para llevar a cabo las funciones de expedición de los pedidos solicitados.

La zona de expedición está formada por dos estaciones, dos robots y el transporte.

Las estaciones o elementos que componen esta sección son las siguientes:

- **Transporte:** de la misma manera que ocurría en la zona de fabricación, será el encargado de trasladar los palets de una estación a otra.



Fig. 151: Transporte Zona II

- **Estación 6:** esta estación será la encargada de realizar la colocación de las bases sobre las cuales se servirán los pedidos de piezas solicitadas sobre el palet de transporte interno.

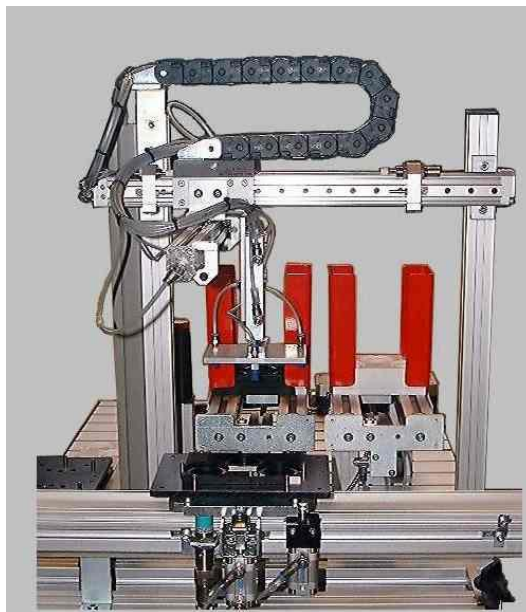


Fig. 152: Estación 6

- **Robot 1:** por medio de dicho robot se cogerán las piezas entregadas por el almacén intermedio para poder colocarlas sobre los orificios de la base del palet de transporte interno situado en la zona de influencia de la misma.
- **Robot 2:** Este segundo robot será el encargado de tomar el pedido ya completado es decir, el conjunto de base más las piezas correspondientes, para ser expedido de forma que abandone la célula de fabricación flexible.
- **Estación 7:** en esta estación se almacenarán los pedidos ya completados sobre los palets utilizados para el transporte interno de la célula de fabricación flexible hasta el momento en el cual deban de ser expedidos.

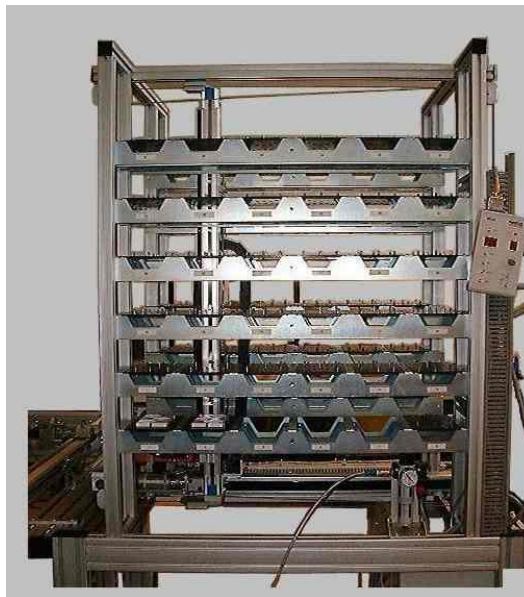


Fig. 153: Estación 7

6.3- DESCRIPCIÓN ESTACIÓN 1

6.3.1- Introducción a la estación 1:

La estación 1, es la denominada estación de identificación y carga de camisas y tiene como objetivo suministrar camisas. Tiene dos formas de funcionamiento si se encuentra dentro del modo de producción automático integrado:

- **Producción bajo pedido:** la estación coloca en el palet de salida la pieza que le ha sido requerida por la estación gestora número 5.
- **Producción en función del tipo de pieza que se encuentra en el cargador:** la estación 5 nos indicará en todo momento si la pieza que tenemos en el cargador puede ser producida si tenemos espacio en el almacén o bien si debe ser desechada.

Las camisas salen desde el cilindro donde se almacenan hasta ser depositadas en un palet que llega por la cinta transportadora y dicho palet, esperará frente a la estación hasta que se realice la operación.

Dicha estación puede dividirse en tres bloques bien diferenciados:

- Zona almacén de camisas.
- Brazo con desplazamiento horizontal, vertical y lateral.
- Zona de verificación de tapas.

El proceso llevado a cabo por la estación 1 es el siguiente:

1. Las camisas se encuentran apiladas de forma aleatoria dentro de un depósito, y un desapilador, formado por un cilindro y un empujador, se encarga de colocar una camisa en una plataforma con tres sensores que se encargaran de su identificación. [Fig. 154]

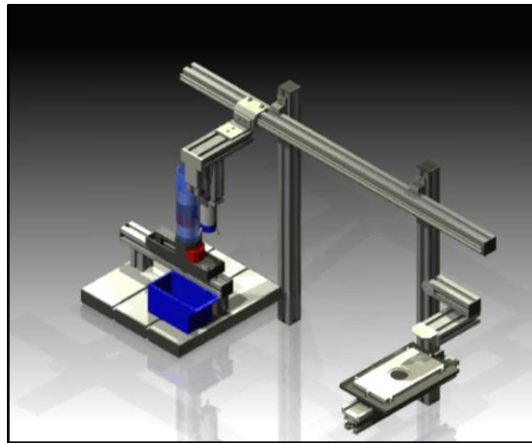


Fig. 154: Estación 1. Paso 1

2. Una vez detectada la pieza por los sensores, el brazo bajará y con una pinza que tiene en su extremo agarrará la pieza. [Fig. 155]

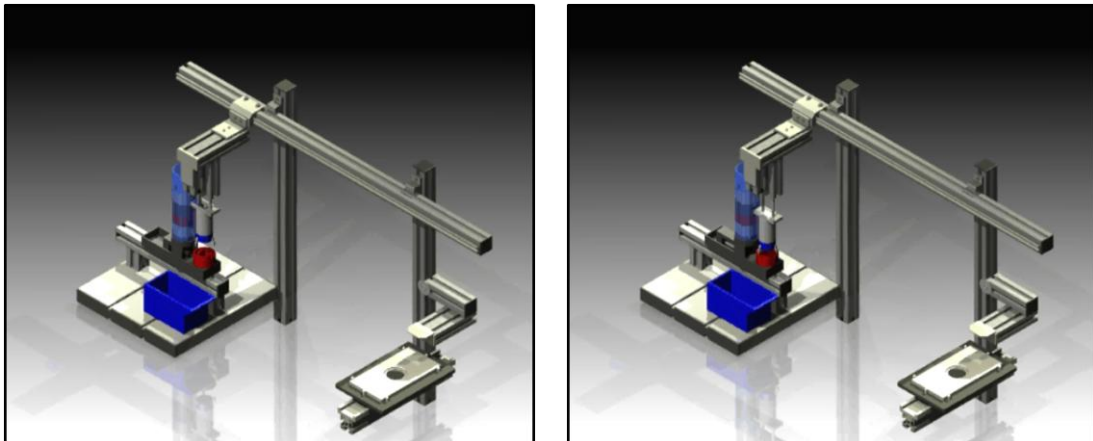


Fig. 155: Estación 1. Paso 2

3. Una vez bien agarrada la camisa, el brazo subirá la pieza. Si quisiéramos desechar la pieza, el brazo realizaría un desplazamiento lateral depositando la pieza en un depósito destinado para ello. Si el funcionamiento es el normal, la pinza se desplazaría hasta colocarse sobre el palet. [Fig. 156]

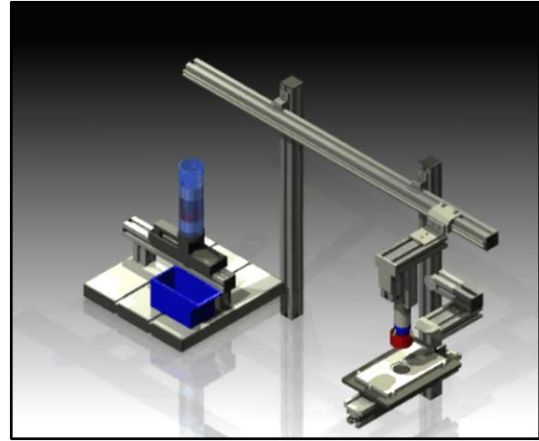
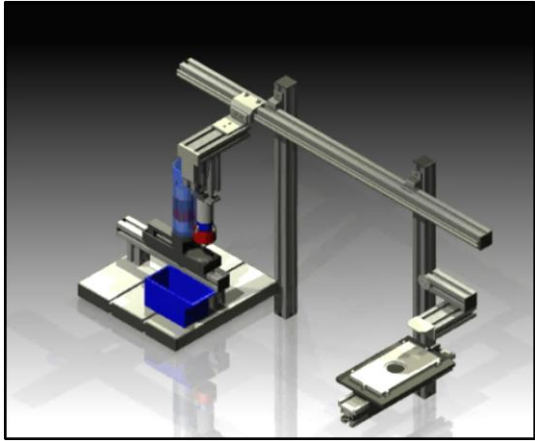


Fig. 156: Estación 1. Paso 3

4. Tras realizar dicho desplazamiento, el brazo bajara la camisa depositándola sobre el palet. [Fig. 157]

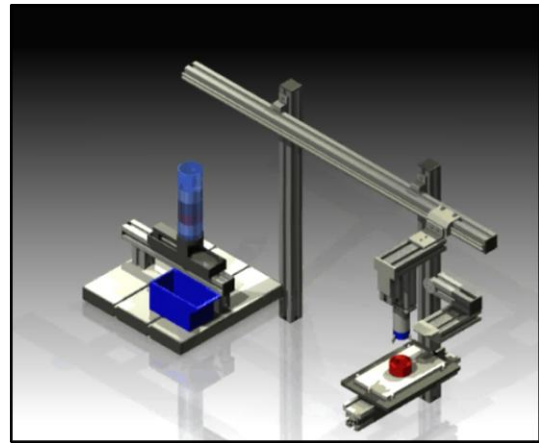
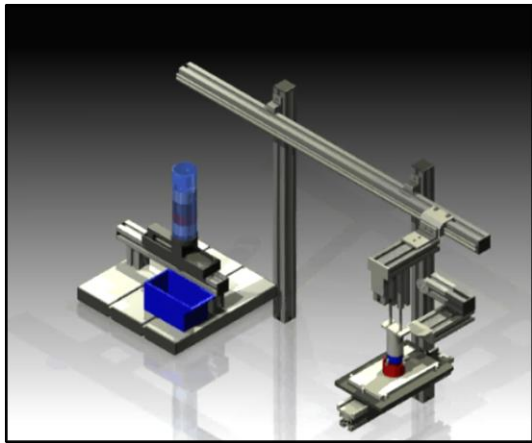


Fig. 157: Estación 1. Paso 4

5. Una vez ahí, se procederá a la lectura de la tapa. Si es positiva, la pieza se deja en el palet y permaneceremos en la posición de reposo. Si no, se cogerá la camisa y se llevara en un desplazamiento lateral sobre el depósito de piezas desechadas. [Fig. 158]

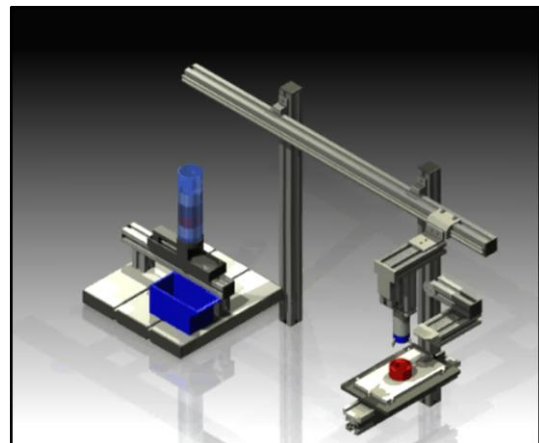
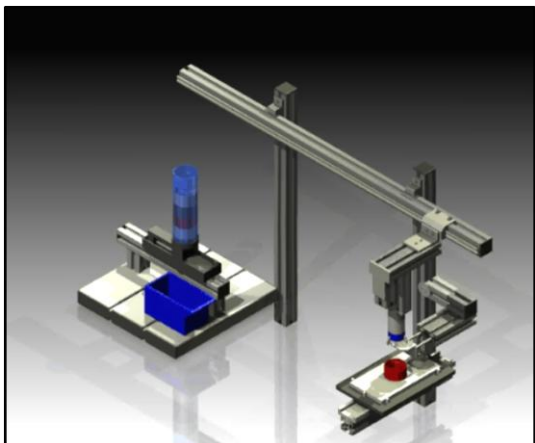


Fig. 158: Estación 1. Paso 5

6.3.2- Descripción general de la estación 1:

El autómata, es el elemento principal encargado de captar las entradas digitales de los sensores, activar las salidas y gestionar todos los procesos de producción, así como de comunicación, mediante los programas internos que disponga.

La información exterior es recibida por el autómata mediante los sensores. Dichos sensores, son uno de los elementos más importantes dentro del proceso productivo automatizado ya que es la única vía por la cual, el programa puede recibir datos realimentados de la acción desarrollada. Estos sensores son todos digitales y se componen de 19 entradas cableadas directamente a los módulos de entrada de la isla Advantys.

Los actuadores de la estación son sus salidas y dicha estación dispone de 8 salidas cableadas directamente sobre el módulo de salidas. Los accionamientos controlados por las salidas del autómata son, en su mayoría, neumáticos lo cual implica que, cada uno de ellos debe tener una electroválvula.

Para controlar las salidas neumáticas también se dispone de un regulador de presión del aire. Las salidas están formadas por cilindros neumáticos de simple y doble efecto que se activan mediante electroválvulas.

6.3.3- Descripción hardware de la estación 1:

6.3.3.1 Autómata:

El autómata 'TM258LF42DT4L' de Schneider- Electric es con el que programaremos las distintas estaciones que componen la célula. Sus principales características ya han sido anteriormente vistas en el punto número dos de la memoria. [2.3- DESCRIPCIÓN DE LA BASE COMPACTA 'TM258LF42DT4L']



Fig. 159: Autómata TM258LF42DTL4

El controlador lógico M258 dispone de un puerto maestro de bus CANopen que será el que utilizemos para la comunicación con la estación.

6.3.3.2 Isla Advantys:

Como ya hemos visto anteriormente, el sistema Advantys STB de E/S consiste en un montaje de módulos de E/S distribuidas y alimentación, entre otros, que funcionan juntos como un nodo de isla en una red de bus de campo abierta. Los módulos instalados en la isla de la estación 1 son:

- **STB PDT 3100:** es un módulo de distribución de energía CC de campo para 24 V. El módulo mide 18.4 mm de ancho y se monta en una base STB XBA 2200 PDM.
- **STB DDI 3610:** es un módulo de entrada digital de seis puntos que toma 24 VCC del bus de sensores de la isla y los transfiere a seis sensores de campo de 24 VCC. Los sensores deben ser dispositivos de dos conductores. Las conexiones de campo se efectúan mediante dos conectores de cableado de campo de seis terminales, STB XTS 2100 de presilla en los seis conectores necesarios para los tres módulos instalado de este tipo. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000. Se debe montar en un grupo de tensión de módulos de E/S de 24 VCC soportados por un módulo de distribución de energía, STB PDT 3100 en nuestro caso.
- **STB DDI 3420:** es un módulo de entrada digital de cuatro puntos que toma 24 VCC del bus de sensores de la isla y transfiere energía a dos dispositivos sensores de campo de 24 VCC. Los sensores pueden ser dispositivos de tres conductores. Las conexiones de campo se efectúan mediante un conector de cableado de campo de seis terminales STB XTS 2100 de presilla, pues no es necesario más. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000. Se debe montar en un grupo de tensión de módulos de E/S de 24 VCC soportados por un módulo de distribución de energía STB PDT 310x.
- **STB DDO 3600:** es un módulo de salida digital de seis puntos que toma 24 VCC del bus de accionadores de la isla y transfiere energía a seis dispositivos accionadores de campo de 24 VCC. Soporta corrientes de 0.5 A por punto. Los accionadores deben ser dispositivos de dos conductores. Las conexiones de campo se efectúan mediante un par de conectores de cableado de campo de seis terminales, STB XTS 1100 atornillables en el módulos instalado. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000.
- **STB DDO 3200:** es un módulo de salida digital de dos puntos que toma 24 VCC del bus de accionadores de la isla y transfiere energía a dos dispositivos accionadores de campo de 24 VCC. Soporta corrientes de 0.5 A por punto. Los accionadores deben ser dispositivos de tres conductores. Las conexiones de campo se efectúan mediante un par de conectores de cableado de campo de seis terminales, STB XTS 1100 atornillables. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000.

En la siguiente imagen puede comprobarse el resultado final de la isla:

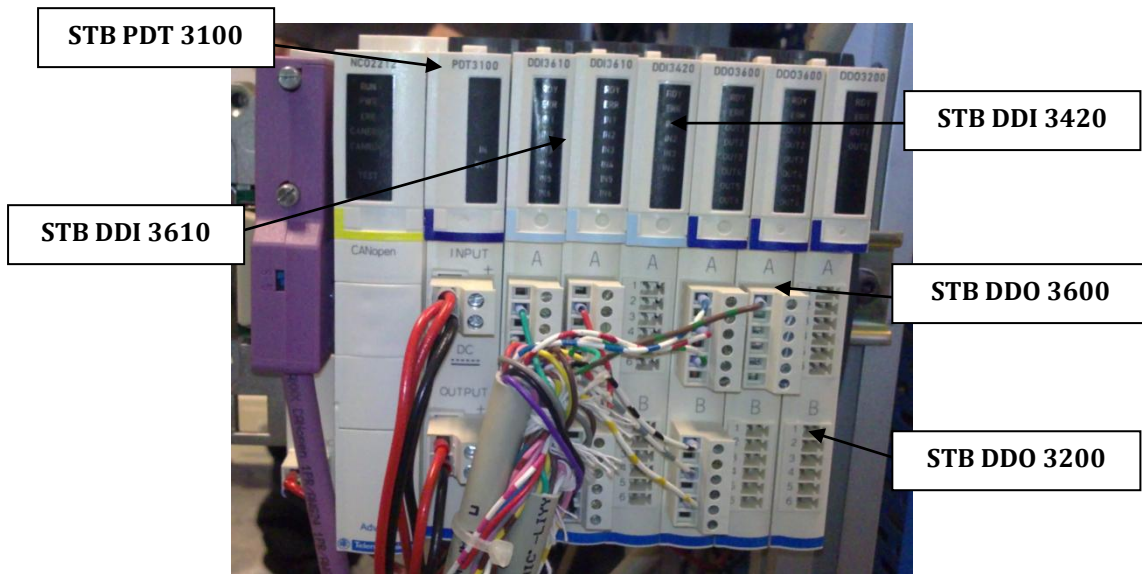


Fig. 160: Isla Advantys Estación 1

6.3.3.3 Módulos de precableado de entradas y salidas:

En la estación 1 disponemos de tres módulos de FESTO integrados con 8 entradas y 8 salidas cada uno de ellos aunque no se utilizan por completo.



Fig. 161: Módulos Precableado Estación 1

Las luces de color verde que se encuentran debajo de cada entrada y salida, muestran cuando éstas se encuentran activas en cada momento.

A continuación se muestra la numeración y denominación de las entradas y salidas que da FESTO.

	Entrada	Símbolo	Salida	Símbolo
Módulo 0	I 0.0	Cinta_atras	O 0.0	Cinta_avanza
	I 0.1	Cinta_adelante	O 0.1	Cinta_retrocede
	I 0.2	Pinza_izda	O 0.2	Pinza_fuera

Módulo 0	I 0.3	Pinza_drcha	O 0.3	Pinza_dentro
	I 0.4	Pinza_arriba	O 0.4	Pinza_baja
	I 0.5	Pinza_abajo	O 0.5	Cargador
	I 0.6	Cargador_adelante	O 0.6	Pinza
	I 0.7	Cargador_atras	O 0.7	Lector
Módulo 1	I 1.0	Emergencia	-	-
	I 1.1	Marcha	-	-
	I 1.2	Manual_automatico	-	-
	I 1.3	Ind_int	-	-
	I 1.4	Rearme	-	-
	I 1.5	Capacitivo_camisa	-	-
	I 1.6	Optico_camisa	-	-
Módulo 2	I 1.7	Inductivo_camisa	-	-
	I 2.0	Lector adelante	-	-
	I 2.1	Lector atras	-	-
	I 2.2	Optico lector	-	-

Fig. 162: Tabla Módulos Entradas y Salidas Estación 1

6.3.3.4 Captadores:

Existen diferentes tipos de sensores dentro de la estación 1. Los principales tipos de sensores son interruptores de proximidad inductivos, sensores de proximidad y sensores ópticos. Además, también existen entradas que se activan de forma manual, pulsadores e interruptores.

- **Interruptores de proximidad inductivos:** producen una señal eléctrica al aproximarse un campo magnético. Se emite a través de un circuito integrando la correspondiente señal eléctrica. El estado de conmutación se indica mediante un diodo luminoso de color amarillo que se ilumina al accionarse. [Fig. 163]

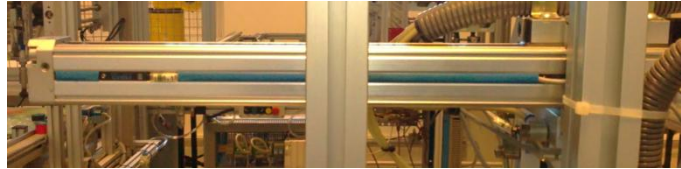


Fig. 163: Interruptor de Proximidad Inductivo

Por lo tanto, estos captadores se utilizan para diferenciar las distintas posiciones que ocupan los diferentes elementos móviles que tiene la estación. Su lectura para cada una de esas posiciones dará como resultado diferentes valores.

- **Sensores de proximidad:** son aquellos dispositivos eléctricos, electromecánicos o electrónicos capaces de reaccionar ante un objeto situado en un entorno definido del mismo. De este modo, no es necesario un contacto físico entre el sensor y el elemento a detectar. En la estación 1 utilizaremos tres tipos de sensores los cuales [Fig. 164], se utilizan para la identificación de los diferentes tipos de camisas disponibles a la salida del cargador:

- **Inductivos:** su principio se basa en el funcionamiento de un circuito oscilante LC de alta frecuencia que deja de oscilar frente a la proximidad de un objeto metálico. Al dejar de oscilar este circuito, se cerrará un contacto de forma que tenemos una continuidad entre los bornes del detector.

- **Capacitivos:** su principio de acción se basa en el funcionamiento de un circuito oscilante RC ajustado a un punto crítico al de oscilación. Al aproximar a este circuito que esta oscilando un material cualquiera con una constante dieléctrica superior a la del aire, se provoca una variación de la capacidad del condensador y el circuito comienza a oscilar.

- **Ópticos:** este detector consta de un emisor y un receptor de luz. Cuando la luz emitida es reflejada, indicará que detectamos la presencia de un objeto. Para poder discriminar entre objetos de diferentes colores, la luz reflejada tiene que llegar a unos niveles de umbral los cuales han sido ajustados de tal modo que con la camisa negra, no se llegue a esos valores y por lo tanto, no se detecte presencia de un objeto.

A continuación, se muestra una imagen donde aparecen agrupados dichos tres sensores.

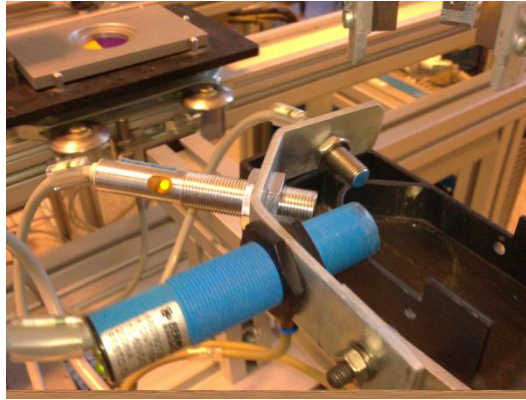


Fig. 164: Sensores de Proximidad

- **Sensores ópticos:** este sensor se utiliza para saber si la camisa dispone ya de tapa o no para continuar el proceso o en cambio ser desechada.

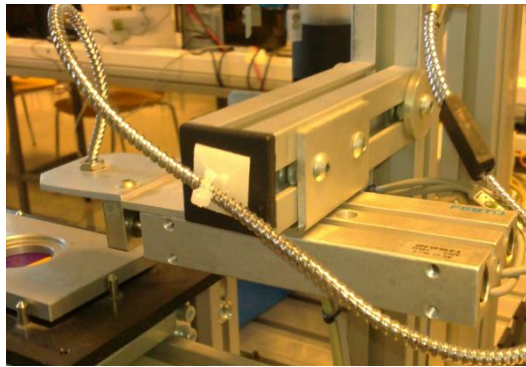


Fig. 165: Sensor Óptico

- **Interruptores y pulsadores:** conforman la llamada botonera. Está formada por dos pulsadores y tres interruptores que sirven para seleccionar los diferentes modos de funcionamiento.



Fig. 166: Botonera Estaciones

Cabe destacar el interruptor con forma de seta roja. Se trata de la seta de emergencia. Es una entrada del autómatas por lo que el control de la emergencia y la interrupción del funcionamiento de las salidas deben hacerse por software. Esto es muy importante ya que si la gestión de emergencias no funciona correctamente los equipos pueden resultar dañados.

6.3.3.5 Actuadores:

En esta estación, todos los actuadores son cilindros neumáticos es decir, son dispositivos mecánicos que permiten, a partir de una presión de aire, obtener un movimiento lineal alternativo limitado.

- a) **Cinta_avanza / Cinta_retrocede:** salidas encargadas de realizar el movimiento horizontal del brazo desde el cargador hasta el palet. Este desplazamiento se realiza mediante un cilindro sin vástago con tope mecánico y amortiguación hidráulica en el extremo y está controlado por una electroválvula biestable. Se corresponden con las salidas %QX4.0 y %QX4.1.

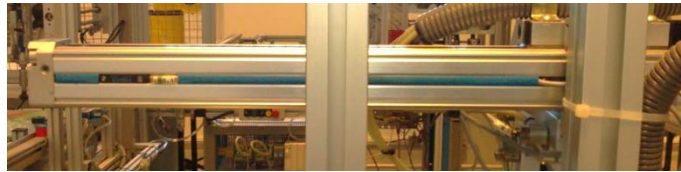


Fig. 167: Actuadores Cinta Avanza / Retrocede

- b) **Pinza_fuera / Pinza_dentro:** salidas encargadas de realizar el movimiento de traslación desde el cargador hasta la cubeta de piezas desechadas. El movimiento se realiza por un cilindro con vástago de doble efecto controlado por una electroválvula biestable. Se corresponden con las salidas %QX4.2 y %QX4.3.



Fig. 168: Actuadores Pinza Fuera / Dentro

- c) **Cargador:** salida destinada a sacar las camisas apiladas en un almacén de forma cilíndrica, empujando la camisa y retrocediendo posteriormente. Es un cilindro de simple efecto controlado por una válvula monoestable. Se corresponde con la salida %QX4.5.

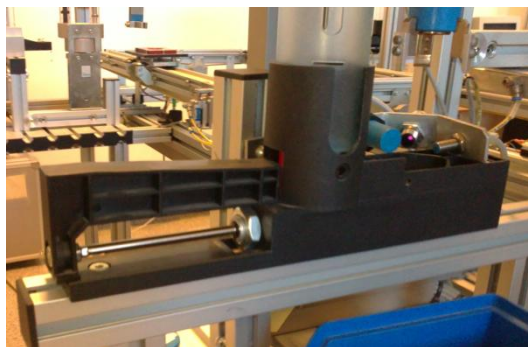


Fig. 169: Cargador

- d) **Pinza_baja:** salida encargada de realizar el movimiento de subir y bajar el brazo para coger la camisa. Es un cilindro con vástago de simple efecto montado verticalmente y está controlado por una válvula monoestable. Se corresponde con la salida %QX4.4.



Fig. 170: Actuador Pinza Baja

- e) **Pinza:** salida encargada de agarrar la camisa. Es un cilindro de simple efecto y válvula monoestable. Para que agarre la camisa, debe estar activa durante todo el momento deseado. Se corresponde con la salida %QX5.0.

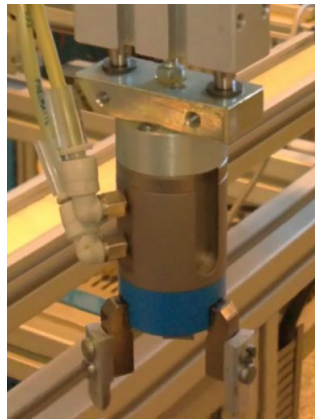


Fig. 171: Pinza

- f) **Lector:** salida encargada del movimiento de sacar el lector óptico encargado de verificar si las camisas tienen tapa. Es realizado por un cilindro con vástago de simple efecto y válvula monoestable. Se corresponde con la salida %QX5.1.



Fig. 172: Lector

6.4- DESCRIPCIÓN ESTACIÓN 3

6.4.1- Introducción a la estación 3:

La estación 3, es la denominada estación de montaje de culatas y tiene como objetivo montar la culata sobre la camisa es decir, roscar la tapa. Las culatas salen desde el cilindro donde se almacenan hasta ser colocadas y roscadas sobre una camisa que se encuentra en la cinta transportadora.

Dicha estación puede dividirse en tres bloques bien diferenciados:

- Zona almacén de culatas, dónde las culatas son almacenadas.
- Brazo con desplazamiento horizontal y vertical para el transporte de las culatas hasta el palet donde se encuentra la camisa.
- Mordaza, también llamada abrazadera que se encarga de sujetar la camisa cuando se está roscando la culata.

El proceso llevado a cabo por la estación 3 es el siguiente:

1. Las culatas se encuentran apiladas dentro de un depósito y un desapilador, formado por un cilindro empujador, se encarga de colocar una culata en una plataforma con un sensor óptico que detecta su presencia. [Fig. 173]

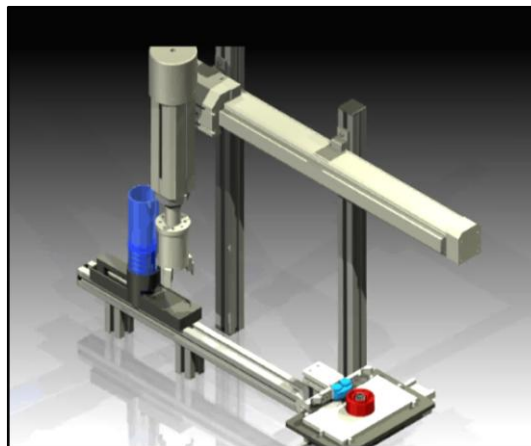


Fig. 173: Estación 3. Paso 1

2. Una vez detectada la culata por el sensor óptico, el empujador se desactivará. El brazo bajará y con una pinza que tiene en su extremo agarrará la pieza. [Fig. 174]

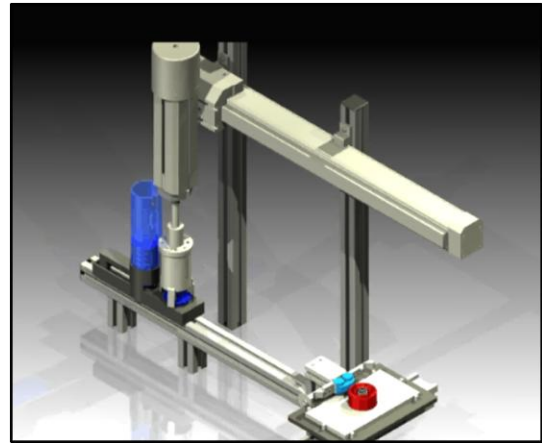
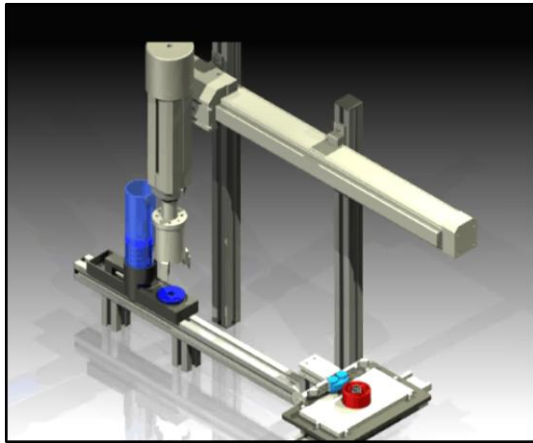


Fig. 174: Estación 3. Paso 2

3. El brazo subirá la culata y la desplazará lateralmente hacia el palet donde se encuentra la camisa. [Fig. 175]

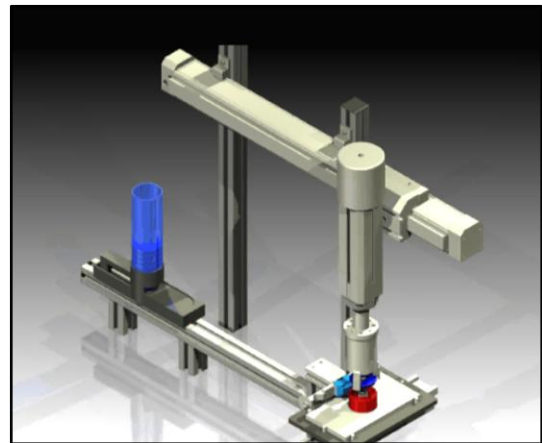
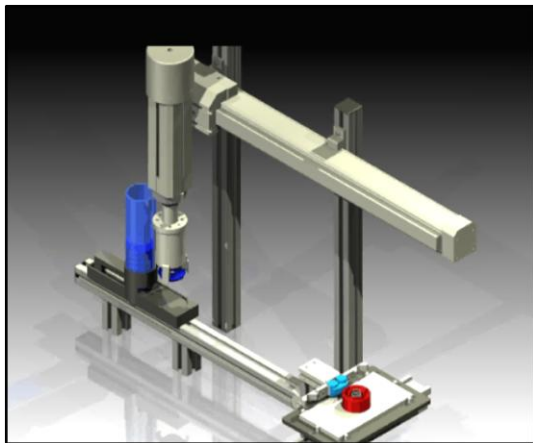


Fig. 175: Estación 3. Paso 3

4. La mordaza se cerrará para sujetar la camisa y el brazo bajará la culata. Una vez cerrada la mordaza y con el brazo abajo, se procederá al roscado de la culata sobre la camisa. [Fig. 176]

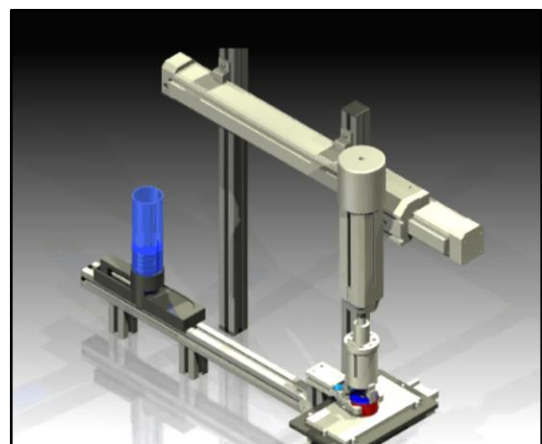
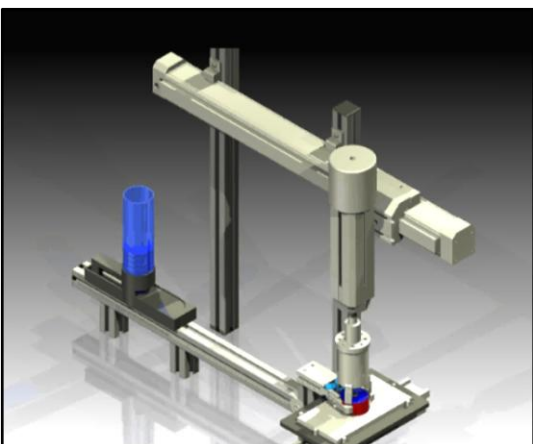


Fig. 176: Estación 3. Paso 4

5. Una vez roscada, ya tendremos la pieza por lo que la mordaza se soltará y el brazo volverá su posición inicial de reposo deshaciendo además el movimiento de roscado.

6.4.2- Descripción general de la estación 3:

Al igual que en la estación anterior, el autómatas es el elemento principal encargado de captar las entradas digitales de los sensores, activar las salidas y gestionar todos los procesos de producción, así como de comunicación, mediante los programas internos que disponga.

La información exterior es recibida por el autómatas mediante los sensores. Dichos sensores, son uno de los elementos más importantes dentro del proceso productivo automatizado ya que es la única vía por la cual, el programa puede recibir datos realimentados de la acción desarrollada. Estos sensores son todos digitales y se componen de 12 entradas cableadas directamente a los módulos de entrada de la isla Advantys.

Los actuadores de la estación son sus salidas y dicha estación dispone de 7 salidas cableadas directamente sobre el módulo de salidas. Los accionamientos controlados por las salidas del autómatas son, en su mayoría, neumáticos lo cual implica que, cada uno de ellos debe tener una electroválvula.

Para controlar las salidas neumáticas también se dispone de un regulador de presión del aire. Las salidas están formadas por cilindros neumáticos de simple y doble efecto que se activan mediante electroválvulas.

6.4.3- Descripción hardware de la estación 3:

6.4.3.1 Autómatas:

Para la programación de la estación 3, utilizaremos el mismo autómatas que para la estación anterior, el autómatas 'TM258LF42DT4L' de Schneider-Electric. [6.3.3.1 Autómatas:]

Utilizaremos nuevamente el maestro de bus CANopen para la comunicación del autómatas con la isla.

6.4.3.2 Isla Advantys:

Como ya hemos visto anteriormente, el sistema Advantys STB de E/S consiste en un montaje de módulos de E/S distribuidas y alimentación, entre otros, que funcionan juntos como un nodo de isla en una red de bus de campo abierta. Los módulos instalados en la isla de la estación 3 son idénticos a los de la estación 1 anteriormente descritos. [6.3.3.2 Isla Advantys:]

6.4.3.3 Módulos de precableado de entradas y salidas:

En la estación 3 disponemos de dos módulos de FESTO integrados con 8 entradas y 8 salidas cada uno de ellos aunque no se utilizan por completo.

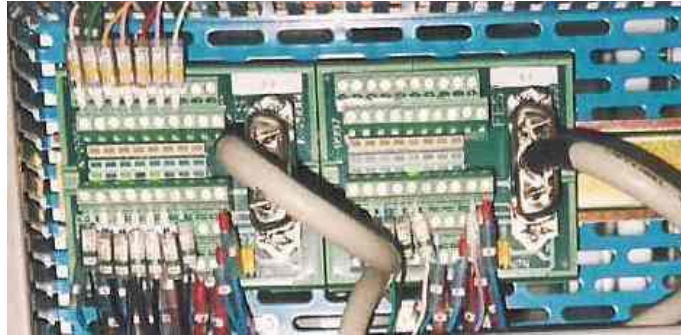


Fig. 177: Módulos Precableado Estación 3

A continuación, se muestra una tabla con la denominación y numeración de las entradas y salidas de la estación 3 que da FESTO.

	Entrada	Símbolo	Salida	Símbolo
Módulo 0	I 0.0	Cinta_atras	O 0.0	-
	I 0.1	Cinta_adelante	O 0.1	-
	I 0.2	Gira_izda	O 0.2	-
	I 0.3	Gira_drcha	O 0.3	-
	I 0.4	Pinza_arriba	O 0.4	-
	I 0.5	Pinza_abajo	O 0.5	-
	I 0.6	Cargador	O 0.6	-
Módulo 1	I 1.0	Emergencia	O 1.0	Cinta_avanza
	I 1.1	Marcha	O 1.1	Cinta_retrocede
	I 1.2	Ind_int	O 1.2	Roscar
	I 1.3	Rearme	O 1.3	Pinza_baja
	I 1.4	Manual_automatico	O 1.4	Culata
	I 1.5	-	O 1.5	Fijar
	I 1.6	-	O 1.6	Pinza

Fig. 178: Tabla Módulos Entradas y Salidas Estación 3

6.4.3.4 Captadores:

Existen diferentes tipos de sensores dentro de la estación 3. Los principales tipos de sensores son interruptores de proximidad inductivos y un sensor óptico. Además, también existen entradas que se activan de forma manual, pulsadores e interruptores.

- **Interruptores de proximidad inductivos:** producen una señal eléctrica al aproximarse un campo magnético. Se emite a través de un circuito integrando la correspondiente señal eléctrica.

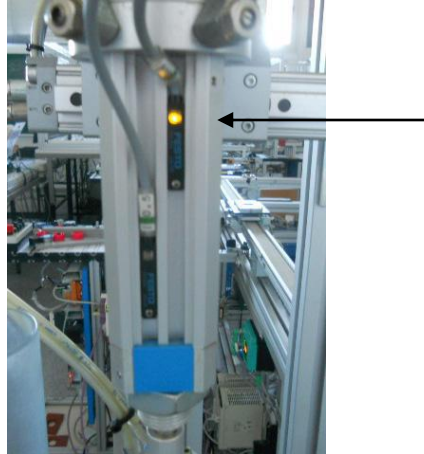


Fig. 179: Interruptor de Proximidad Inductivo

Por lo tanto, estos captadores se utilizan para diferenciar las distintas posiciones que ocupan los diferentes elementos móviles que tiene la estación. Su lectura para cada una de esas posiciones dará como resultado diferentes valores.

- **Sensores ópticos:** este sensor se utiliza para saber si la culata ha salido correctamente del almacén de culatas. En el momento que la luz que emite dicho sensor es reflejada, indica la presencia de una culata.

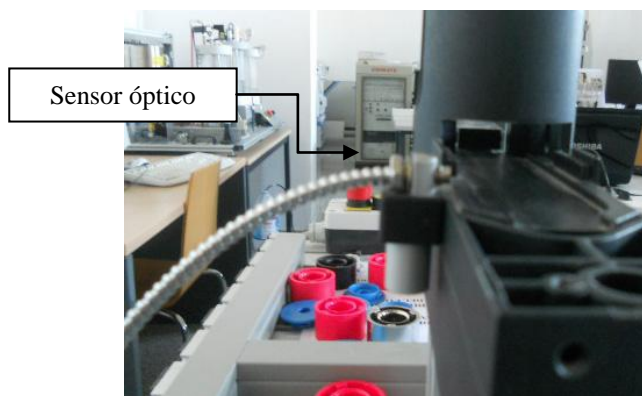


Fig. 180: Sensor Óptico

- **Interruptores y pulsadores:** conforman la llamada botonera. Está formada por dos pulsadores y tres interruptores que sirven para seleccionar los diferentes modos de funcionamiento. [Fig. 166]

Cabe destacar el interruptor con forma de seta roja. Se trata de la seta de emergencia. Es una entrada del autómatas por lo que el control de la emergencia y la interrupción del funcionamiento de las salidas deben hacerse por software. Esto es muy importante ya que si la gestión de emergencias no funciona correctamente los equipos pueden resultar dañados.

6.4.3.5 Actuadores:

En esta estación, todos los actuadores son cilindros neumáticos al igual que en la estación anterior.

- a) **Cinta_avanza / Cinta_retrocede:** salidas encargadas de realizar el movimiento horizontal del brazo desde el desapilador de culatas hasta el palet. Este desplazamiento se realiza mediante un cilindro sin vástago con tope mecánico y amortiguación hidráulica en el extremo y está controlado por una electroválvula biestable. Se corresponden con las salidas %QX4.0 y %QX4.1.



Fig. 181: Actuadores Cinta Avanza / Retrocede

- b) **Roscar:** salida encargadas de realizar el movimiento de roscado de la culata sobre la camisa. Para ello, utiliza un actuador giratorio compuesto por un cilindro de simple efecto controlado por una válvula monoestable. Se corresponde con la salida %QX4.2.



Fig. 182: Actuador Roscar

- c) **Pinza_baja:** salida encargada de realizar el movimiento de subir y bajar el brazo para coger la camisa. Es un cilindro con vástago de simple efecto montado verticalmente y está controlado por una válvula monoestable. Se corresponde con la salida %QX4.3.



Fig. 183: Pinza Baja

- d) **Culata:** salida destinada a sacar las culatas apiladas en un almacén de forma cilíndrica, empujando la culata y retrocediendo posteriormente. Es un cilindro de simple efecto controlado por una válvula monoestable. Se corresponde con la salida %QX4.4.

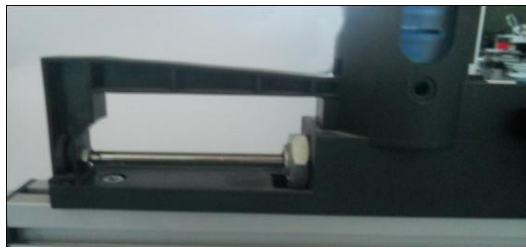


Fig. 184: Desapilador de Culatas

- e) **Fijar:** se trata de una mordaza que agarra la camisa durante el proceso de roscado. Es un cilindro de simple efecto y válvula monoestable. Para que agarre la camisa, debe estar activa durante todo el momento deseado. Se corresponde con la salida %QX4.5.

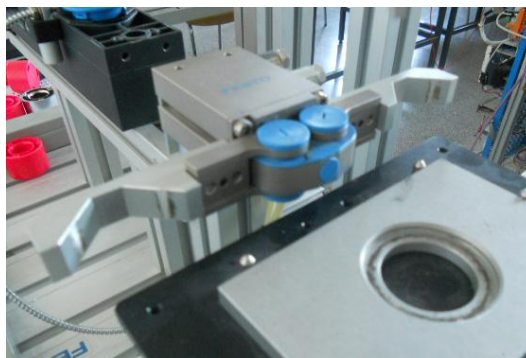


Fig. 185: Mordaza

- f) **Pinza:** salida encargada de agarrar la culata. Es un cilindro de simple efecto y válvula monoestable. Para que agarre la culata, debe estar activa durante todo el momento deseado. Se corresponde con la salida %QX5.0.

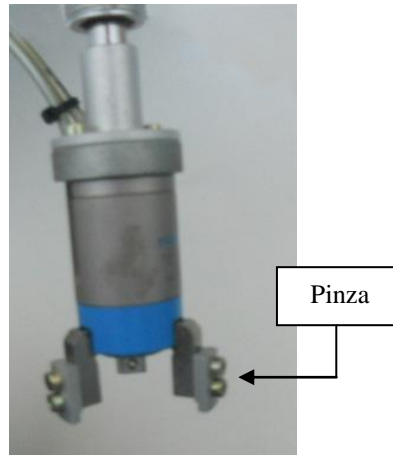


Fig. 186: Pinza

6.5- DESCRIPCIÓN ESTACIÓN 4

6.5.1- Introducción a la estación 4:

La estación 4, es la denominada estación de verificación de cilindros y tiene como objetivo determinar si el cilindro que ha sido producido es válido y puede ser almacenado o si bien es un componente defectuoso y debe ser desechado.

Las camisas que ya tienen tapa y no necesitan el montaje de los componentes, no pasarán esta verificación ya que en caso de hacerlo, siempre sería negativa al no tener émbolo. Por lo tanto, estas piezas pasaran directamente como piezas válidas.

Para dicha verificación, en primer lugar hay que introducir aire a presión en la camisa para provocar la salida del émbolo y poder medir así su carrera. Dicho valor se comparará con uno de referencia y si el resultado es positivo, podrá ser almacenado. En cambio, si el resultado de dicha comparación es negativo, la pieza será desechada.

Dicha estación puede dividirse en tres bloques bien diferenciados:

- Brazo con ventosas, para coger la pieza y transportarla al verificador.
- Verificador, donde se comprueba si la pieza es óptima o si debe ser desechada mediante la inyección de aire a presión.
- Brazo basculante, para recoger la pieza del verificador y encaminarla a la cinta transportadora o en su defecto al recipiente de piezas desechadas.

El proceso llevado a cabo por la estación 4 es el siguiente:

1. Una vez disponemos de una pieza en la estación, el brazo con dos ventosas bajará y hará vacío para recoger la pieza. [Fig. 187]

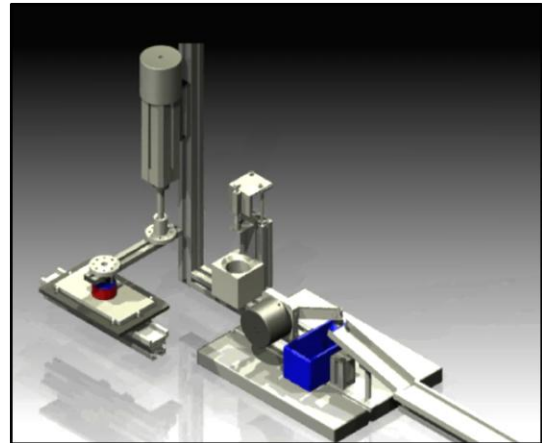
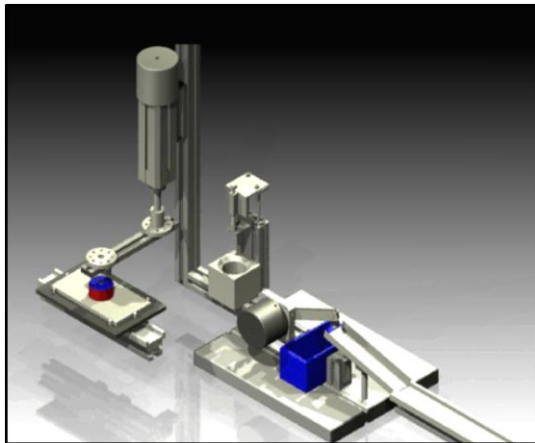


Fig. 187: Estación 4. Paso 1

2. Una vez hayamos hecho vacío, subiremos el brazo y lo giraremos hasta llegar a la zona del verificador. [Fig. 188]

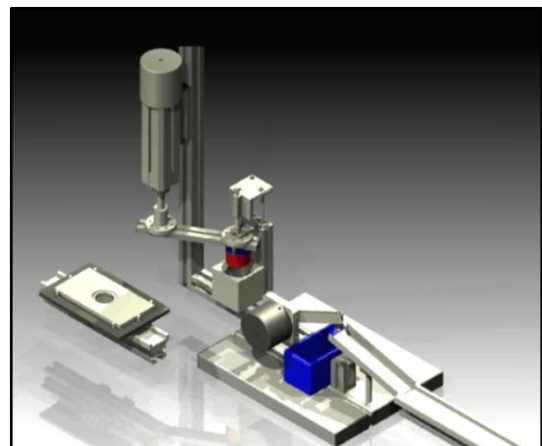
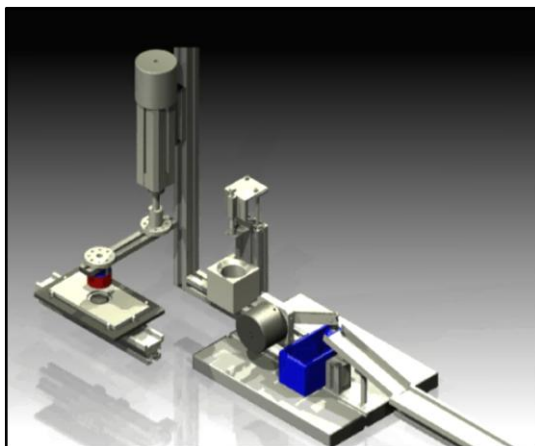


Fig. 188: Estación 4. Paso 2

3. Depositaremos la pieza en el verificador bajando el brazo y dejando de hacer vacío y devolveremos el brazo a su posición inicial. [Fig. 189]

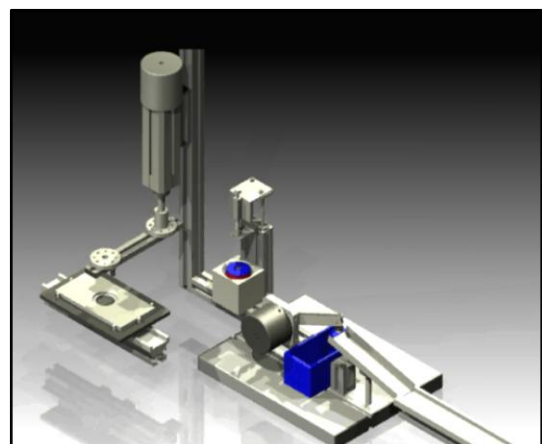
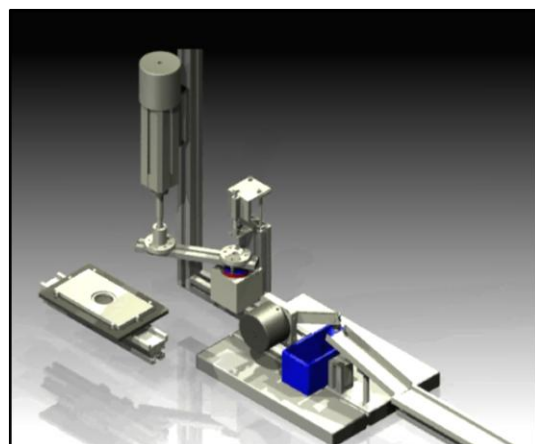


Fig. 189: Estación 4. Paso 3

4. El siguiente paso consiste en la verificación de la pieza. Para ello, bajaremos el verificador e inyectaremos aire durante un cierto intervalo de tiempo para realizar dicha comprobación. Pasado este tiempo, dejaremos de inyectar aire y subiremos el verificador de nuevo. [Fig. 190]

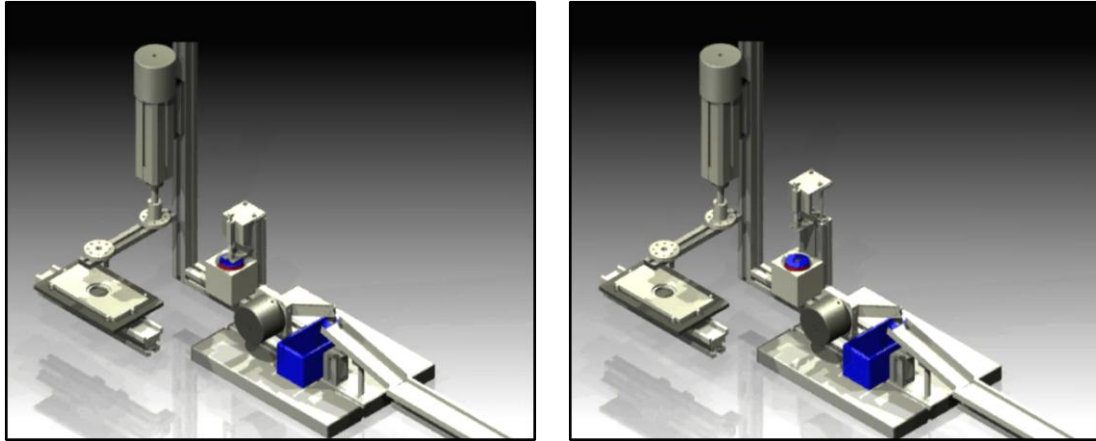


Fig. 190: Estación 4. Paso 4

5. Tras comprobar la verificación, ya sea positiva o negativa, habrá que colocar el brazo basculante sobre la pieza y hacer vacío mediante la ventosa para recogerla. Además, un pequeño expulsador situado en el verificador ayudará a la expulsión de la pieza. El brazo basculante colocará la pieza sobre una rampa basculante. Si la comprobación ha resultado positiva, esta rampa dirigirá la pieza hacia la cinta transportadora. En caso contrario, la pieza irá a parar a un depósito de piezas desechadas. [Fig. 191]

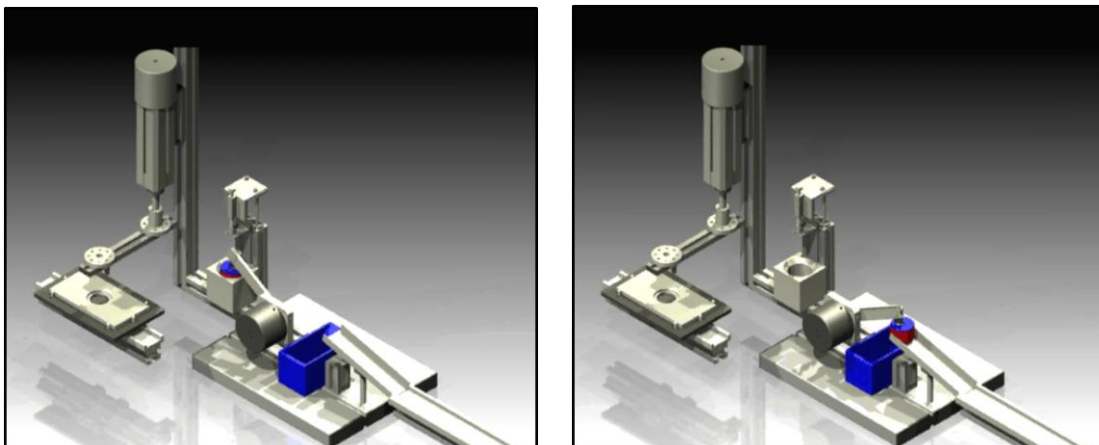


Fig. 191: Estación 4. Paso 5

6.5.2- Descripción general de la estación 4:

Como ya hemos visto en las estaciones, el autómatas es el elemento principal encargado de captar las entradas digitales de los sensores, activar las salidas y gestionar todos los procesos de producción, así como de comunicación, mediante los programas internos que dispone.

La estación 4 se caracteriza por ser una de las estaciones que más elementos incorporan entre los que cabe destacar el equipo de vacío, un vacuostato o micro interruptores entre otros

La información exterior es recibida por el autómata mediante los sensores.. La peculiaridad de esta estación es que además de los sensores digitales, compuestos por 16 entradas cableadas directamente a los módulos de entrada de la isla Advantys, dispone de un sensor analógico utilizado para la verificación de las piezas.

Los actuadores de la estación son sus salidas y dicha estación dispone de 11 salidas cableadas directamente sobre el módulo de salidas. Los accionamientos controlados por las salidas del autómata son neumáticos lo cual implica que, cada uno de ellos debe tener una electroválvula.

6.5.3- Descripción hardware de la estación 4:

6.5.3.1 Autómata:

Para la programación de la estación 4, utilizaremos el mismo autómata que para la estación anterior, el autómata ‘**TM258LF42DT4L**’ de Schneider-Electric. [6.3.3.1 Autómata:]

Utilizaremos nuevamente el maestro de bus CANopen para la comunicación del autómata con la isla.

6.5.3.2 Isla Advantys:

Como ya hemos visto anteriormente, el sistema Advantys STB de E/S consiste en un montaje de módulos de E/S distribuidas y alimentación, entre otros, que funcionan juntos como un nodo de isla en una red de bus de campo abierta.

Los módulos instalados en la estación 4 son:

- **STB DDI 3610:** es un módulo de entrada digital de seis puntos que toma 24 VCC del bus de sensores de la isla y los transfiere a seis sensores de campo de 24 VCC. Los sensores deben ser dispositivos de dos conductores. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000. Se montan en un grupo de tensión de módulos de E/S de 24 VCC soportados por el módulo de distribución de energía STB PDT 3100.
- **STB DDO 3600:** es un módulo de salida digital de seis puntos que toma 24 VCC del bus de accionadores de la isla y transfiere energía a seis dispositivos accionadores de campo de 24 VCC. Soporta corrientes de 0.5 A por punto. Los accionadores deben ser dispositivos de dos conductores. Las conexiones de campo se efectúan mediante un par de conectores de cableado de campo de seis terminales, STB XTS 1100 atornillables o STB XTS 2100 de presilla. El módulo mide 13.9 mm de ancho y se monta en una base

de E/S STB XBA 1000. Se montan en un grupo de tensión de módulos de E/S de 24 VCC soportados por el módulo de distribución de energía STB PDT 3100.

- **STB AVI 1270:** es un módulo de entrada de tensión analógica de dos canales que transfiere datos en el intervalo de -10 a +10 V, con una resolución de 11 bits más signo. Toma 24 VCC del bus del sensor de la isla y los envía a dos sensores de campo analógicos de 24 VCC. Las conexiones de campo se efectúan mediante un par de conectores de cableado de campo de seis terminales, STB XTS 1100 atornillables o STB XTS 2100 de presilla. El módulo mide 13.9 mm de ancho y se monta en una base de E/S STB XBA 1000. Se montan en un grupo de tensión de módulos de E/S de 24 VCC soportados por el módulo de distribución de energía STB PDT 3100.

En la siguiente imagen puede comprobarse el resultado final de la isla:



Fig. 192: Isla Advantys Estación 4

La alimentación de la isla Advantys es independiente de la del autómeta. Es una alimentación ABL7RE2402 con entrada de red y salida a 24v ajustables.



Fig. 193: Alimentación Isla Advantys

6.5.3.3 Equipos de vacío:

Uno de los posibles mecanismos de sujeción que se dan en el ámbito industrial y que es utilizado en la estación 4 es el de la sujeción mediante ventosas por vacío.

En la estación 4 disponemos de dos elementos con dicho tipo de sujeción, el brazo giratorio y el brazo basculante. El brazo giratorio dispone de dos ventosas mientras que el brazo basculante solo posee una para realizar la sujeción de las piezas. Además, la estación dispone de dos equipos de vacío. Las ventosas de los brazos se comunican con los equipos de vacío mediante unos pequeños tubos.

Existen diferentes formas de generar vacío pero en el caso que nos acontece, utilizaremos un generador de vacío de efecto Venturi. Se utiliza dicho generador al tratarse del más sencillo y del que menos mantenimiento necesita.

Dicho dispositivo se basa en una tobera inyectora por la que sale el aire a presión el cual, es utilizado para generar el vacío entre las ventosas y las piezas.

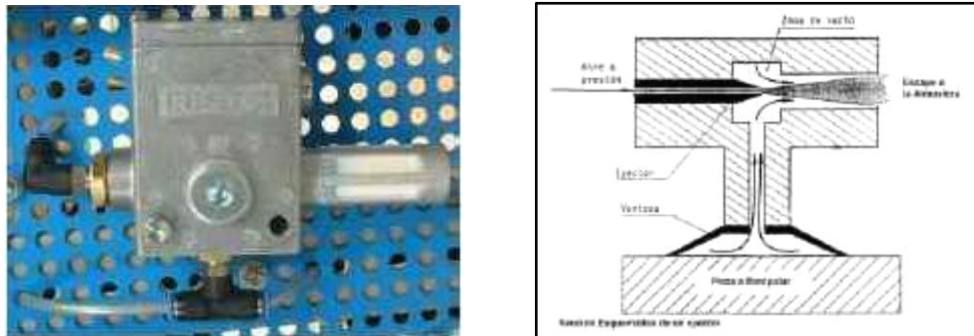


Fig. 194: Generador de Vacío Efecto Venturi

6.5.3.4 Módulos de precableado de entradas y salidas:

En la estación 4 disponemos de dos módulos de FESTO integrados con 8 entradas y 8 salidas cada uno de ellos aunque no se utilizan por completo.

Como puede apreciarse, debajo de cada entrada y salida del módulo de FESTO hay unas luces de color verde que indican cuando una entrada o salida está activa.

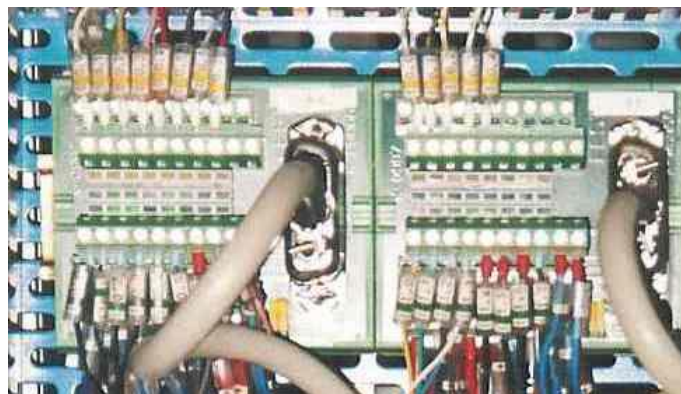


Fig. 195: Módulos Precableado Estación 4

A continuación, se muestra una tabla con la denominación y numeración de las entradas y salidas de la estación 4 que da FESTO.

	Entrada	Símbolo	Salida	Símbolo
Módulo 0	I 0.0	Verificador_abajo	O 0.0	Girar_izqda
	I 0.1	Verificador_arriba	O 0.1	Girar_drcha
	I 0.2	Gira_drcha	O 0.2	Cilindro_baja
	I 0.3	Gira_izqda	O 0.3	Verificador_baja
	I 0.4	Cilindro_arriba	O 0.4	Inyecta
	I 0.5	Cilindro_abajo	O 0.5	Expulsa
	I 0.6	Vacio_pinza	O 0.6	Vacio_en_pinza
	I 0.7	Emergencia	O 0.7	Vacio_en_pieza
Módulo 1	I 1.0	Marcha	O 1.0	Saca_pieza
	I 1.1	Ind_int	O 1.1	Expulsar_pieza
	I 1.2	Rearme	O 1.2	Bascular
	I 1.3	Manual_automatico	O 1.3	-
	I 1.4	Pieza_fuera	O 1.4	-
	I 1.5	Sacar_pieza	O 1.5	-
	I 1.6	Vacio_pieza	O 1.6	-
	I 1.7	Bascula_pieza	O 1.7	-

Fig. 196: Tabla Módulos Entradas y Salidas Estación 4

6.5.3.5 Captadores:

Existen diferentes tipos de sensores dentro de la estación 4. Dicha estación está compuesta de interruptores de proximidad inductivos, vacuostatos, microinterruptores y sensores analógicos. Además, también existen entradas que se activan de forma manual, pulsadores e interruptores.

- **Interruptores de proximidad inductivos:** producen una señal eléctrica al aproximarse un campo magnético. Se emite a través de un circuito integrando la correspondiente señal eléctrica.

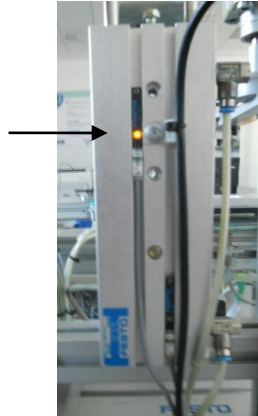


Fig. 197: Interruptor de Proximidad Inductivo

Por lo tanto, estos captadores se utilizan para diferenciar las distintas posiciones que ocupan los diferentes elementos móviles que tiene la estación.

- **Microinterruptores eléctricos:** el brazo basculante utiliza estos elementos para transmitir su posición. Se utilizan como finales de carrera eléctricos.



Fig. 198: Microinterruptor eléctrico

- **Vacuostato:** es el elemento utilizado para la sujeción de piezas mediante vacío. El funcionamiento de éstos se basa en la diferencia de presión que se origina en la ventosa.



Fig. 199: Vacuostato

- **Medidor analógico:** la verificación de las piezas es llevado a cabo gracias a la información que suministra este elemento. El medidor, se encarga de medir el recorrido del émbolo del cilindro en su salida. Según la posición del sensor, obtenemos un valor el cual es comparado con un valor de referencia que determina la validez o no del cilindro.

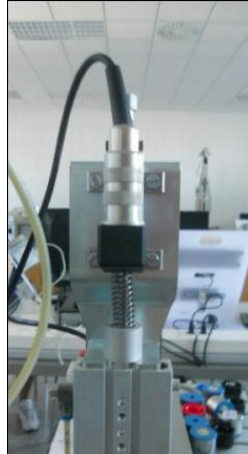


Fig. 200: Medidor Analógico

- **Interruptores y pulsadores:** conforman la llamada botonera. Está formada por dos pulsadores y tres interruptores que sirven para seleccionar los diferentes modos de funcionamiento. [Fig. 166]

Cabe destacar el interruptor con forma de seta roja. Se trata de la seta de emergencia. Es una entrada del autómatas por lo que el control de la emergencia y la interrupción del funcionamiento de las salidas deben hacerse por software. Esto es muy importante ya que si la gestión de emergencias no funciona correctamente los equipos pueden resultar dañados.

6.5.3.6 Actuadores:

En esta estación, todos los actuadores son cilindros neumáticos al igual que en las estaciones anteriores.

- a) **Girar_izqda/ Girar_drcha:** salidas encargadas de realizar el movimiento giratorio del brazo desde el palet al verificador y viceversa. Es controlado por una válvula biestable y su principal característica es que no es necesaria su alimentación eléctrica hasta que termina el movimiento sino que, con la activación de la bobina durante un corto periodo de tiempo, el cilindro se desplaza hasta el final de su recorrido. Se corresponden con las salidas %QX4.0 y %QX4.1.

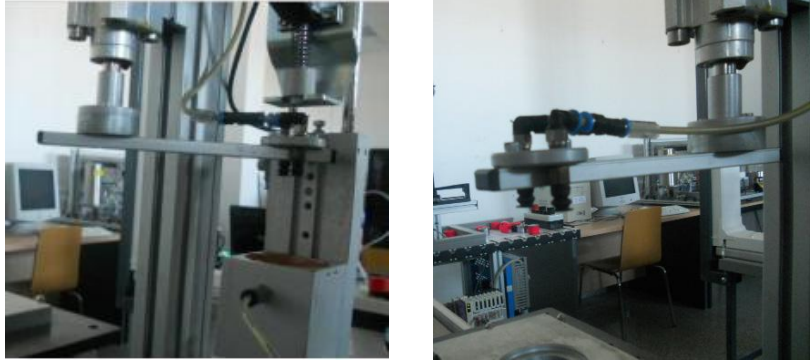


Fig. 201: Brazo a la Derecha o Izquierda

- b) **Cilindro_baja:** salida encargada de realizar el movimiento de subir y bajar el brazo para coger la camisa controlada por una válvula monoestable. Al activarse la bobina, el vástago inicia su recorrido hasta llegar al tope mecánico. Si dejamos de activar esta salida, el brazo vuelve a su posición de reposo. Se corresponde con la salida %QX4.2.



Fig. 202: Cilindro Baja

- c) **Verificador_baja:** salida encargada de realizar el movimiento de subir y bajar el verificador para realizar la comprobación. Es un cilindro de doble efecto construido con dos vástagos, para mayor potencia y seguridad y está controlado por una válvula monoestable. Al igual que el brazo giratorio, cuando se desactiva esta salida, el verificador vuelve a su posición de reposo. Se corresponde con la salida %QX4.3.

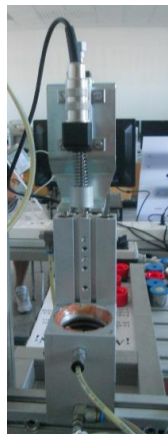


Fig. 203: Verificador

- d) **Inyecta:** esta salida no es un accionador como tal. Es utilizada para inyectar aire en el verificador y es controlada por una válvula monoestable. Se corresponde con la salida %QX4.4.
- e) **Expulsa:** se trata de un cilindro de simple efecto instalado en la base del alojamiento del verificador y se encarga de expulsar el cilindro de dicho alojamiento. Es controlado por una válvula monoestable. Se corresponde con la salida %QX4.5.
- f) **Vacío_en_pinza:** salida encargada de recoger el cilindro mediante vacío para trasladarlo al verificador. Es un cilindro de simple efecto y válvula monoestable. Para que agarre la culata, debe estar activa durante todo el momento deseado. Se corresponde con la salida %QX5.0.
- g) **Vacío_en_pieza:** salida encargada de recoger el cilindro mediante vacío para trasladarlo del verificador a la salida mediante el brazo basculante. Sólo dispone de una ventosa. Se corresponde con la salida %QX5.1.
- h) **Saca_pieza / Expulsar_pieza:** salidas encargadas del movimiento del brazo basculante. Es un cilindro de doble efecto basculante controlado por una válvula biestable. Se corresponden con las salidas %QX5.2 y %QX5.3.
- i) **Bascular:** salida encargada de posicionar la rampa basculante encaminada hacia la cinta transportadora o hacia el depósito de piezas desechadas. Es un cilindro de simple efecto controlado por una válvula monoestable. En reposo, la rampa está dirigida hacia la cinta transportadora. Al activarla, cambia su posición sobre el depósito. Se corresponde con la salida %QX5.4.



Fig. 204: Rampa Basculante

6.6- DESCRIPCIÓN ESTACIÓN 6

6.6.1- Introducción a la estación 6:

La estación 6, es la denominada estación de carga de bases y tiene como objetivo suministrar las bases sobre las que se montan los pedidos que luego son almacenados en la estación 7.

Existen dos tipos de bases diferenciadas únicamente por su color, las bases blancas y las bases negras. Cada tipo de base se encuentra almacenada en su depósito correspondiente. Las bases se utilizarán en función del pedido realizado con anterioridad.

Dicha estación puede dividirse en dos bloques bien diferenciados:

- Zona de almacén y expedición de las bases.
- Brazo con ventosas, para coger las bases y transportarlas al palet de salida.

El proceso llevado a cabo por la estación 6 es el siguiente:

1. Una vez llegado el palet a la estación y recibida la orden de sacar pieza, la estación saca la placa correspondiente. Para ello, un cilindro empuja la pieza del depósito sacándola al exterior y colocándola sobre la plataforma de salida. Un sensor óptico detecta dicha placa y en ese momento se desactiva el cilindro. [Fig. 205]

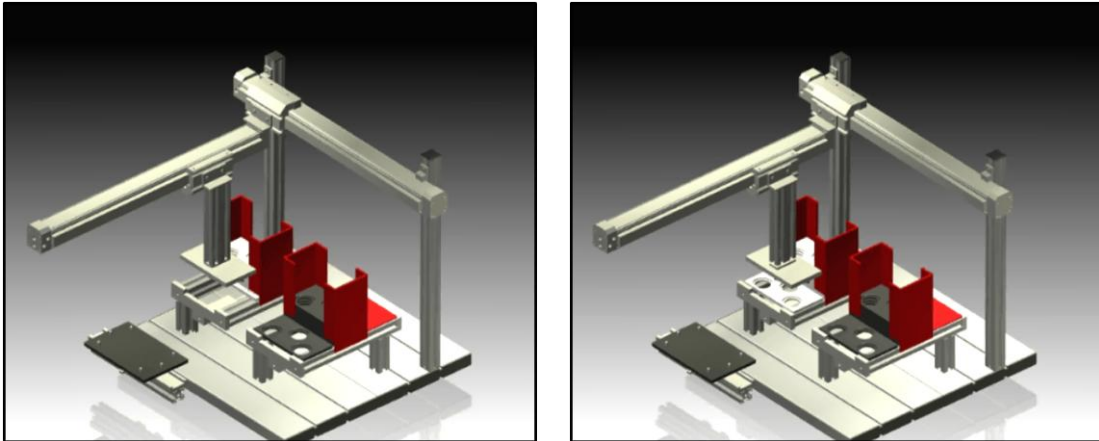


Fig. 205: Estación 6. Paso 1

2. En este momento, habrá que posicionar el brazo sobre la placa dependiendo del tipo de placa desplazándose a la izquierda si la placa requerida es negra. En este momento pasaremos a recoger la placa mediante succión hasta que haga vacío. La succión permanecerá hasta que la soltemos sobre el palet. [Fig. 206]

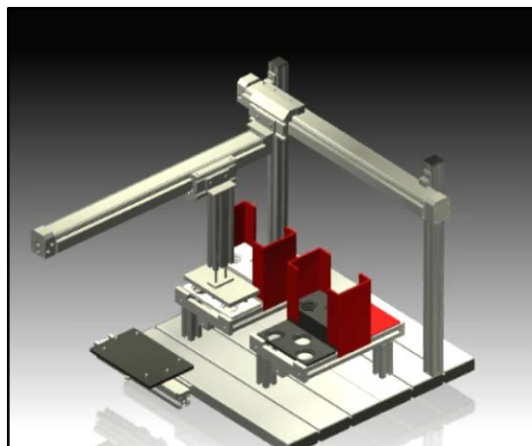


Fig. 206: Estación 6. Paso 2

3. Cuando se haya producido el vacío, subiremos el brazo y lo posicionaremos frente al transbordador del palet si no se encuentra ya bien colocado. [Fig. 207]

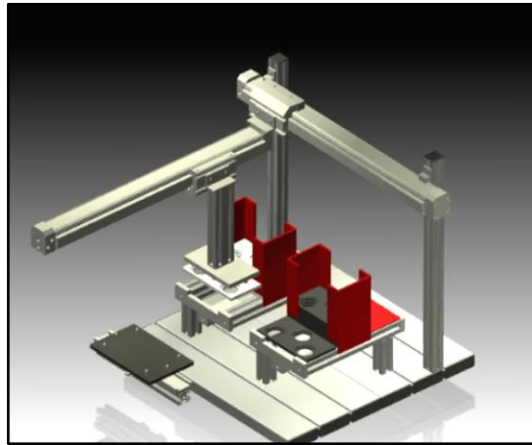


Fig. 207: Estación 6. Paso 3

4. Una vez colocado el brazo en dicha posición, lo avanzamos hasta colocarlo sobre el palet y bajamos la placa. [Fig. 208]

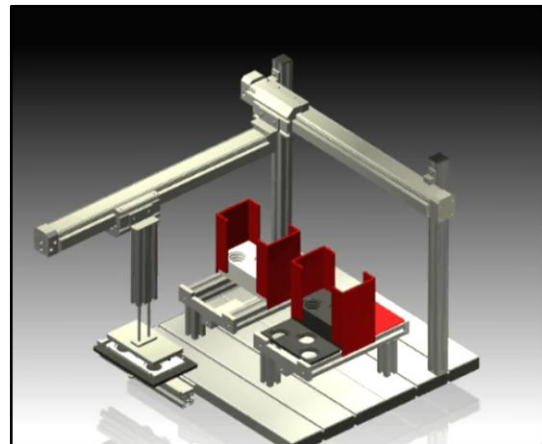
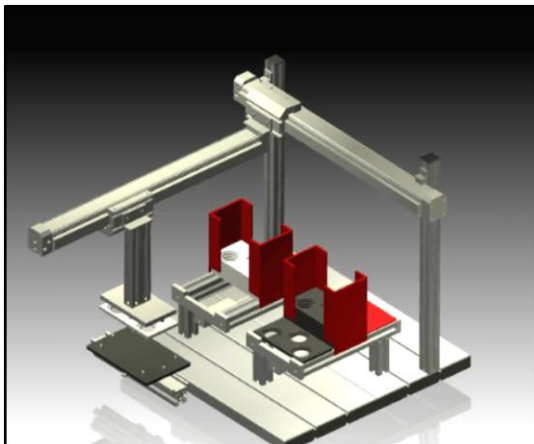


Fig. 208: Estación 6. Paso 4

5. Para finalizar, dejaremos de hacer vacío en la placa para soltarla sobre el palet y subiremos el brazo devolviéndolo a su posición inicial de reposo. [Fig. 209]

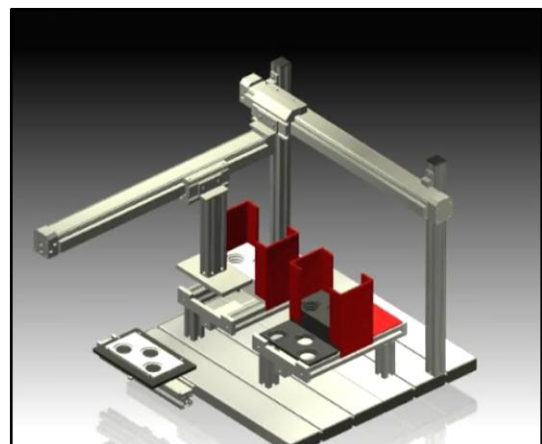
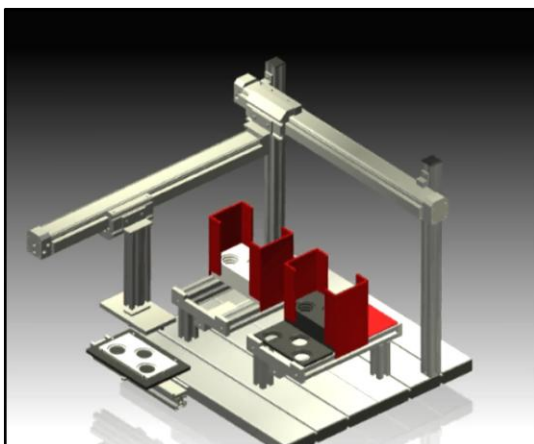


Fig. 209: Estación 6. Paso 5

6.6.2- Descripción general de la estación 6:

Como ya hemos visto en las estaciones anteriores, el autómatas es el elemento principal encargado de captar las entradas digitales de los sensores, activar las salidas y gestionar todos los procesos de producción, así como de comunicación, mediante los programas internos que disponga.

La información exterior es recibida por el autómatas mediante los sensores. Dichos sensores, son uno de los elementos más importantes dentro del proceso productivo automatizado ya que es la única vía por la cual, el programa puede recibir datos realimentados de la acción desarrollada. Estos sensores son todos digitales y se componen de 16 entradas cableadas directamente a los módulos de entrada de la isla Advantys.

Los actuadores de la estación son sus salidas y dicha estación dispone de 9 salidas cableadas directamente sobre el módulo de salidas. Los accionamientos controlados por las salidas del autómatas son, en su mayoría, neumáticos lo cual implica que, cada uno de ellos debe tener una electroválvula.

Para controlar las salidas neumáticas también se dispone de un regulador de presión del aire. Las salidas están formadas por cilindros neumáticos de simple y doble efecto que se activan mediante electroválvulas.

6.6.3- Descripción hardware de la estación 6:

6.6.3.1 Autómatas:

Para la programación de la estación 6, utilizaremos el mismo autómatas que para todas las estaciones anteriores, el autómatas 'TM258LF42DT4L' de Schneider-Electric. [6.3.3.1 Autómatas:]

Utilizaremos nuevamente el maestro de bus CANopen para la comunicación del autómatas con la isla.

6.6.3.2 Isla Advantys:

Como ya hemos visto anteriormente, el sistema Advantys STB de E/S consiste en un montaje de módulos de E/S distribuidas y alimentación, entre otros, que funcionan juntos como un nodo de isla en una red de bus de campo abierta. Los módulos instalados en la isla de la estación 6 son idénticos a los de las estaciones 1 y 3 anteriormente descritos [6.3.3.2 Isla Advantys:].

6.6.3.3 Módulos de precableado de entradas y salidas:

En la estación 6 disponemos de dos módulos de FESTO integrados con 8 entradas y 8 salidas cada uno de ellos como ya hemos visto anteriormente en otras estaciones aunque no se utilizan por completo. Además, debajo de cada entrada y salida del módulo de FESTO hay unas luces de color verde que indican cuando una entrada o salida está activa.

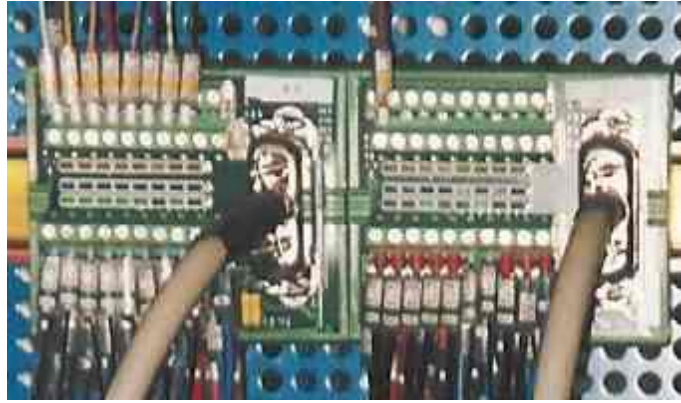


Fig. 210: Módulos Precableado Estación 6

A continuación, se muestra una tabla con la denominación y numeración de las entradas y salidas de la estación 6 que da FESTO.

	Entrada	Símbolo	Salida	Símbolo
Módulo 0	I 0.0	Alimentador_izqdo_atras	O 0.0	Alimentador_izqda
	I 0.1	Optico_alimentador_izqdo	O 0.1	Alimentador_drcha
	I 0.2	Alimentador_drcho_atras	O 0.2	Coger_placa
	I 0.3	Optico_alimentador_drcho	O 0.3	Bajar
	I 0.4	Placa_arriba	O 0.4	Izqda
	I 0.5	Placa_abajo	O 0.5	Drcha
	I 0.6	Placa_drcha	O 0.6	Adelante
	I 0.7	Placa_izqda	O 0.7	Atras
Módulo 1	I 1.0	Emergencia	O 1.0	Alarma
	I 1.1	Marcha	O 1.1	-
	I 1.2	Manual_automatico	O 1.2	-
	I 1.3	Rearme	O 1.3	-
	I 1.4	Ind_int	O 1.4	-
	I 1.5	Placa_atras	O 1.5	-

Módulo 1	I 1.6	Placa_adelante	O 1.6	-
	I 1.7	Vacio	O 1.7	-

Fig. 211: Tabla Módulos Entradas y Salidas Estación 6

6.6.3.4 Captadores:

Existen diferentes tipos de sensores dentro de la estación 6. Dicha estación está compuesta de interruptores de proximidad inductivos, vacuostatos y sensores ópticos. Además, también existen entradas que se activan de forma manual, pulsadores e interruptores.

- **Interruptores de proximidad inductivos:** producen una señal eléctrica al aproximarse un campo magnético. Se emite a través de un circuito integrando la correspondiente señal eléctrica.



Fig. 212: Interruptor de Proximidad Inductivo

Por lo tanto, estos captadores se utilizan para diferenciar las distintas posiciones que ocupan los diferentes elementos móviles que tiene la estación.

- **Sensores ópticos:** estos elementos poseen un emisor y un receptor de luz. Cuando la luz es reflejada, indica la presencia de una pieza sobre ellos. En la estación 6, sirven para detectar la salida de una placa.

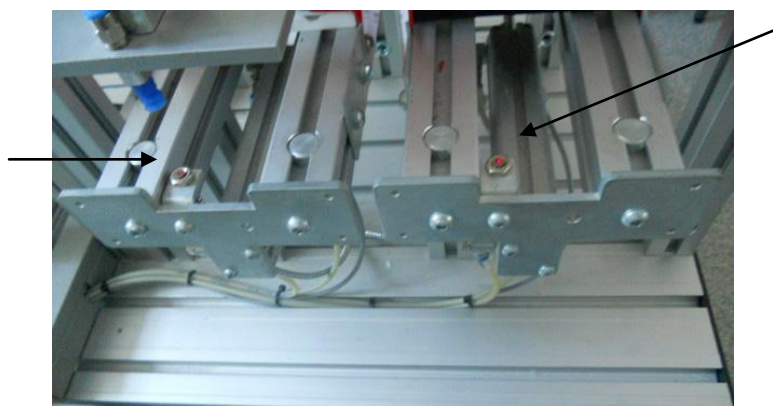


Fig. 213: Sensores Ópticos

- **Vacuostato:** es el elemento utilizado para la sujeción de piezas mediante vacío. El funcionamiento de éstos se basa en la diferencia de presión que se origina en la ventosa.



Fig. 214: Vacuostato

- **Interruptores y pulsadores:** conforman la llamada botonera. Está formada por dos pulsadores y tres interruptores que sirven para seleccionar los diferentes modos de funcionamiento. [Fig. 166]

Cabe destacar el interruptor con forma de seta roja. Se trata de la seta de emergencia. Es una entrada del autómatas por lo que el control de la emergencia y la interrupción del funcionamiento de las salidas deben hacerse por software. Esto es muy importante ya que si la gestión de emergencias no funciona correctamente los equipos pueden resultar dañados.

6.6.3.5 Actuadores:

En esta estación, todos los actuadores son cilindros neumáticos al igual que en las estaciones anteriores.

- a) **Alimentador_izqdo:** esta salida está formada por un empujador encargado de desapilar las placas almacenadas en la izquierda. El movimiento es realizado por un cilindro de doble efecto controlado por una válvula monoestable. Dicha salida debe estar activada hasta que la placa esté completamente fuera ya que si lo desactivamos antes, el cilindro vuelve a su posición de reposo. Se corresponde con la salida %QX4.0. [Fig. 215]
- b) **Alimentador_drcho:** esta salida está formada por un empujador encargado de desapilar las placas almacenadas en la derecha. El movimiento es realizado por un cilindro de doble efecto controlado por una válvula monoestable. Al igual que con el alimentador anterior, la salida debe estar activa hasta que la placa esté completamente sacada. Se corresponde con la salida %QX4.1. [Fig. 215]

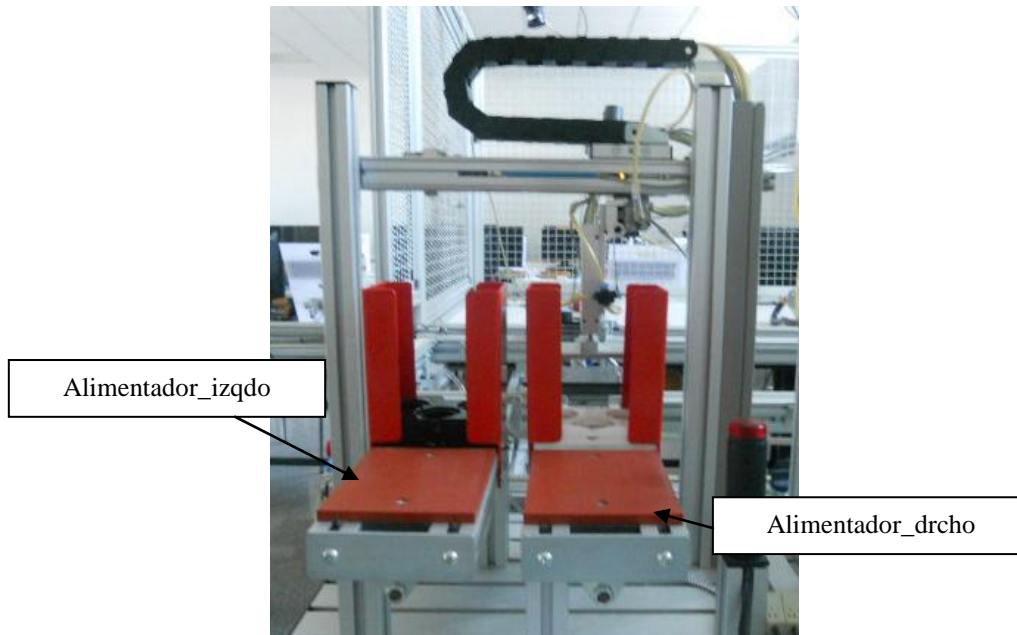


Fig. 215: Alimentadores Estación 6

- c) **Coger_placa:** salida encargada de coger las placas mediante succión por vacío. En este caso no utilizamos ningún cilindro. Utilizamos para el control una válvula monoestable por lo que deberá mantenerse activa durante el periodo de succión de la pieza. Se corresponde con la salida %QX4.2.

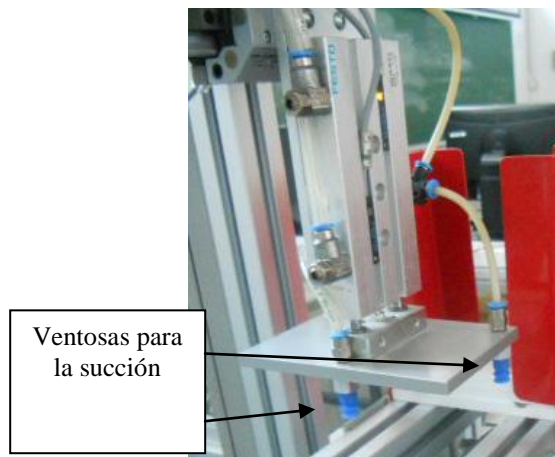


Fig. 216: Brazo con Ventosas

- d) **Bajar:** esta salida es la encargada del movimiento vertical de subir y bajar del brazo succionador. El movimiento es realizado por un cilindro de doble efecto con topes mecánicos controlado por una válvula monoestable por lo que debe mantenerse activa dicha salida para mantener el brazo abajo. Se corresponde con la salida %QX4.3.



Fig. 217: Brazo Abajo

- e) **Izqda / drcha:** salidas encargadas del movimiento lateral del brazo desde el depósito de placas blancas al depósito de placas negras y viceversa. Se utiliza un cilindro sin vástago, con tope mecánico y amortiguación hidráulica en el extremo controlado por una válvula biestable. Por ello, no es necesario mantener la salida activa para realizar dichos movimientos. Sólo debe activarse una de las dos salidas cada vez que se quiera realizar un movimiento. Se corresponden con las salidas %QX4.4 y %QX4.5.



Fig. 218: Actuadores Brazo Izqda / Dcha

- f) **Adelante / Atras:** salidas encargadas del movimiento horizontal del brazo desde el depósito de placas hasta el palet transbordador y viceversa. Se utiliza un cilindro sin vástago, con tope mecánico y amortiguación hidráulica en el extremo controlado por una válvula biestable. Por ello, no es necesario mantener la salida activa para realizar dichos movimientos. Sólo debe activarse una de las dos salidas cada vez que se quiera realizar un movimiento. Se corresponden con las salidas %QX5.0 y %QX5.1.



Fig. 219: Brazo Adelante / Atras

-
- g) **Alarma:** salida encargada de avisar al operario cuando se ha producido un error en la producción en dicha estación. Está formada por un indicador luminoso rojo. Se corresponde con la salida %QX5.2.



Fig. 220: Alarma Luminosa

7- GUIONES DE PRÁCTICAS

7.1- PRÁCTICA 0. INTRODUCCIÓN A LA PROGRAMACIÓN

La programación de las estaciones de la célula de fabricación las realizaremos con el software SoMachine. Para ello, antes de empezar con la programación de las estaciones, realizaremos una pequeña práctica de introducción a la programación y creación de aplicaciones con dicho software que habrá que realizar en todas las prácticas para crear las nuevas aplicaciones.

En primer lugar, iremos al menú de Inicio → Programas → Schneider Electric → SoMachine y lanzamos el programa SoMachine. Nos aparecerá la ventana de Inicio y en ella podremos seleccionar las diferentes opciones para la creación de un nuevo proyecto o la apertura de uno ya existente.

Para crear un nuevo proyecto pulsaremos sobre la opción 'Crear nueva máquina' y nos aparecerá una lista con las diferentes formas posibles de iniciar un nuevo proyecto. Elegiremos la opción 'Empezar con un proyecto vacío'.

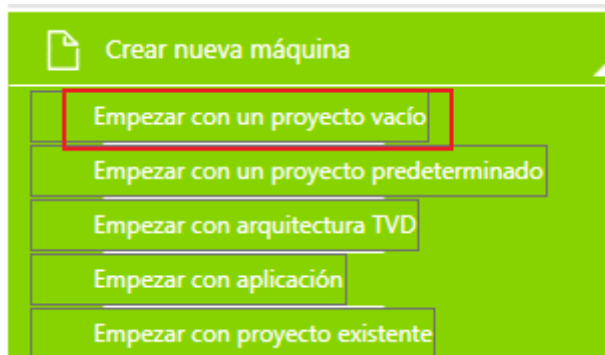


Fig. 221: Menú Nuevo Proyecto

Una vez elegida dicha opción, lo primero que nos pide el programa es guardar el proyecto. Hay que asegurarse que en tipo de archivo está seleccionado el tipo Project File (*.project).

Cuando hayamos guardado el proyecto, nos aparecerán las diferentes pestañas y menús del programa.

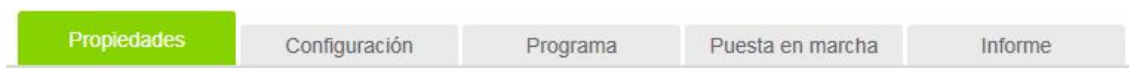


Fig. 222: Menús Programa

La primera pestaña que nos aparece es la de 'Propiedades'. En ella nos aparece la información sobre el fichero la cual puede completarse con información sobre el autor del proyecto aunque para esta práctica no es relevante

En primer lugar iremos a la pestaña de 'Configuración' donde crearemos la configuración de la arquitectura que va a tener la máquina de nuestra aplicación.

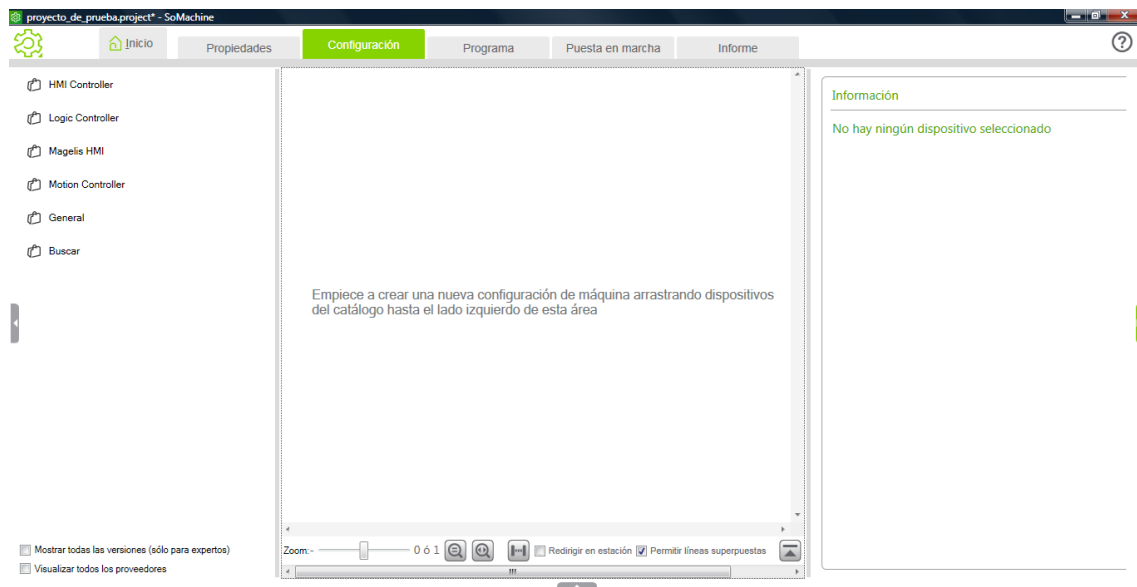


Fig. 223: Pestaña Configuración

Como puede observarse en la imagen [Fig. 223], la zona central se encuentra en blanco y aparece el siguiente mensaje: *'Empiece a crear una nueva configuración de máquina arrastrando dispositivos del catálogo hasta el lado izquierdo de esta área'*.

Para ello, bastará con buscar el controlador requerido para la aplicación en el catálogo que aparece a la izquierda de la pantalla y una vez encontrado, seleccionarlo y arrastrarlo hasta dicha área. En nuestro caso, buscaremos en la carpeta 'Logic Controller' el controlador TM258LF42DT4L que es del que disponemos en el laboratorio de prácticas.

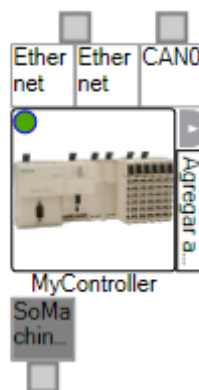


Fig. 224: Controlador TM258LF42DT4L

Vamos a trabajar mediante conexión CanOpen por lo que tenemos que añadir a la configuración el CanOpen Manager. Para incluir el maestro CanOpen a la configuración, basta con hacer un click sobre el recuadro de la configuración bajo el nombre CAN0. Aparecerá una nueva ventana para agregar el dispositivo. Buscaremos en buses de campo el 'CanOpen Performance de Schneider', lo seleccionaremos y le daremos a 'Agregar y cerrar' quedándonos algo de la siguiente forma [Fig. 225]:

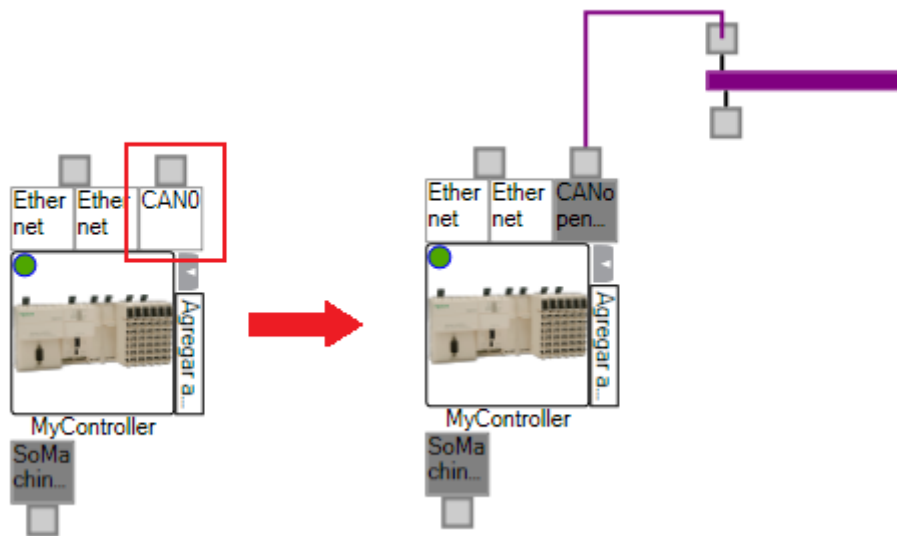


Fig. 225: Agregar Maestro CANopen

Para añadir un dispositivo mediante conexión CanOpen, basta con hacer click sobre el puerto libre es decir, el recuadro que aparece marcado en la siguiente imagen. [Fig. 226]

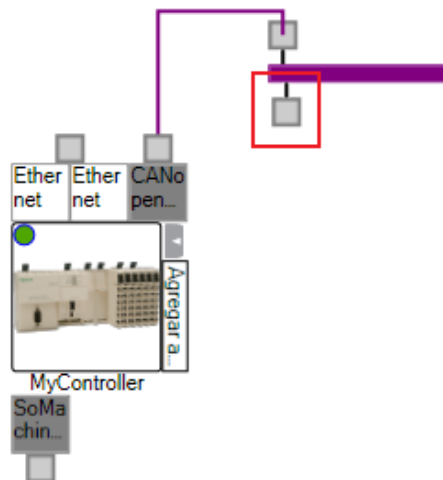


Fig. 226: Añadir Dispositivo CAN

Nos aparecerá una ventana con un listado de dispositivos con conexión CanOpen para agregar. Buscaremos en cada práctica el correspondiente a cada isla y lo agregaremos. Si no lo encontramos en la lista es porque no está instalado en las librerías de SoMachine por lo que habrá que instalarlo para posteriormente agregarlo al maestro CAN.

Para instalar los archivos correspondientes a las islas en la librería, iremos a la pestaña de 'Programa' y seleccionaremos la opción 'Repositorio de dispositivos' disponible en la pestaña de 'Herramientas' del menú general.

Nos aparecerá una nueva ventana con todos los dispositivos instalados en las librerías del programa. Para instalar un dispositivo nuevo pulsaremos sobre 'Instalar' y buscaremos en la carpeta de prácticas disponible en 'C: /temp/practicas célula' los cuatro archivos 'ADVE.eds' correspondientes a las cuatro estaciones sobre las que se realizarán las prácticas posteriores instalándolos uno a uno.

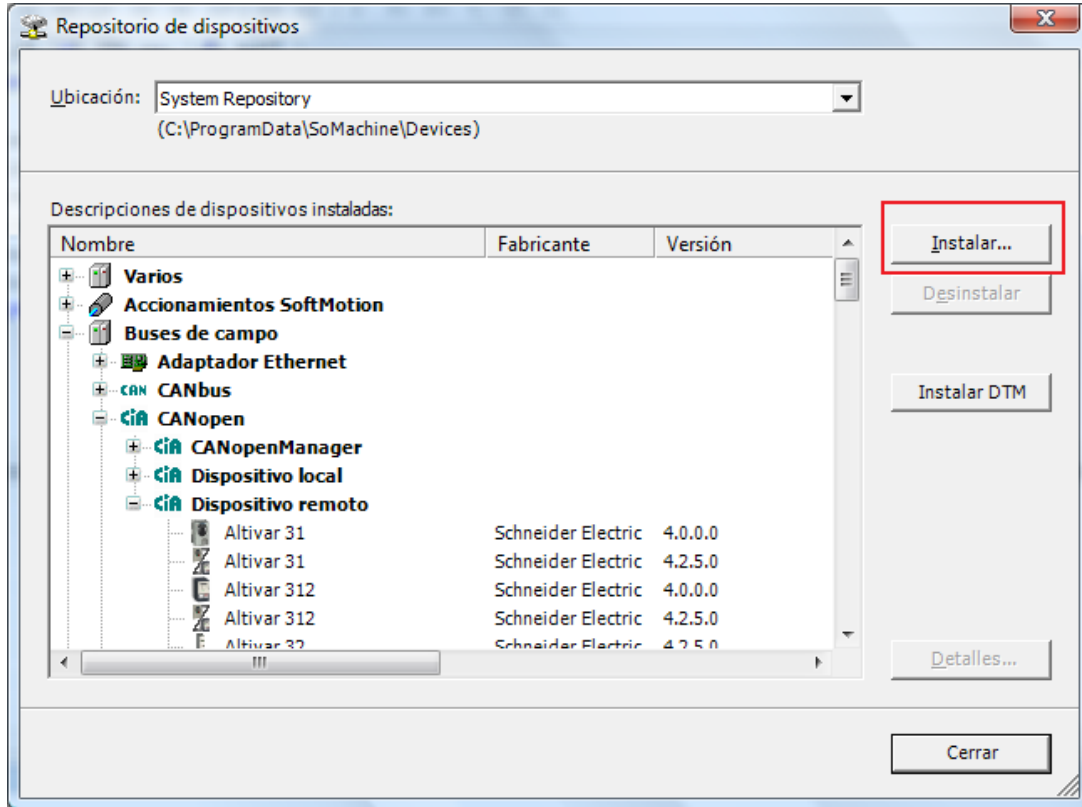


Fig. 227: Instalar Nuevos Dispositivos

Una vez realizado esto, el siguiente paso será establecer la velocidad de transmisión del bus CanOpen en el maestro. Para ello, habrá que hacer doble click sobre el bus Can0 de la ventana de dispositivos en la pestaña de 'Programa'. En el área de trabajo (parte central de la pantalla), nos aparecerá la ventana de configuración. En el campo 'Velocidad de transmisión' estableceremos la velocidad que en nuestro caso son 500 kilobaudios. Una vez realizado esto ya tendremos la configuración del CANopen terminada.



Fig. 228: Establecimiento Velocidad Bus CANopen

Para comenzar a programar habrá que crear una nueva sección de programa y agregarla a la tarea MAST. Para crear una sección nueva, en la ventana de dispositivos vamos a 'Application' y hacemos click con el botón derecho seleccionando la opción POU en Agregar Objeto.

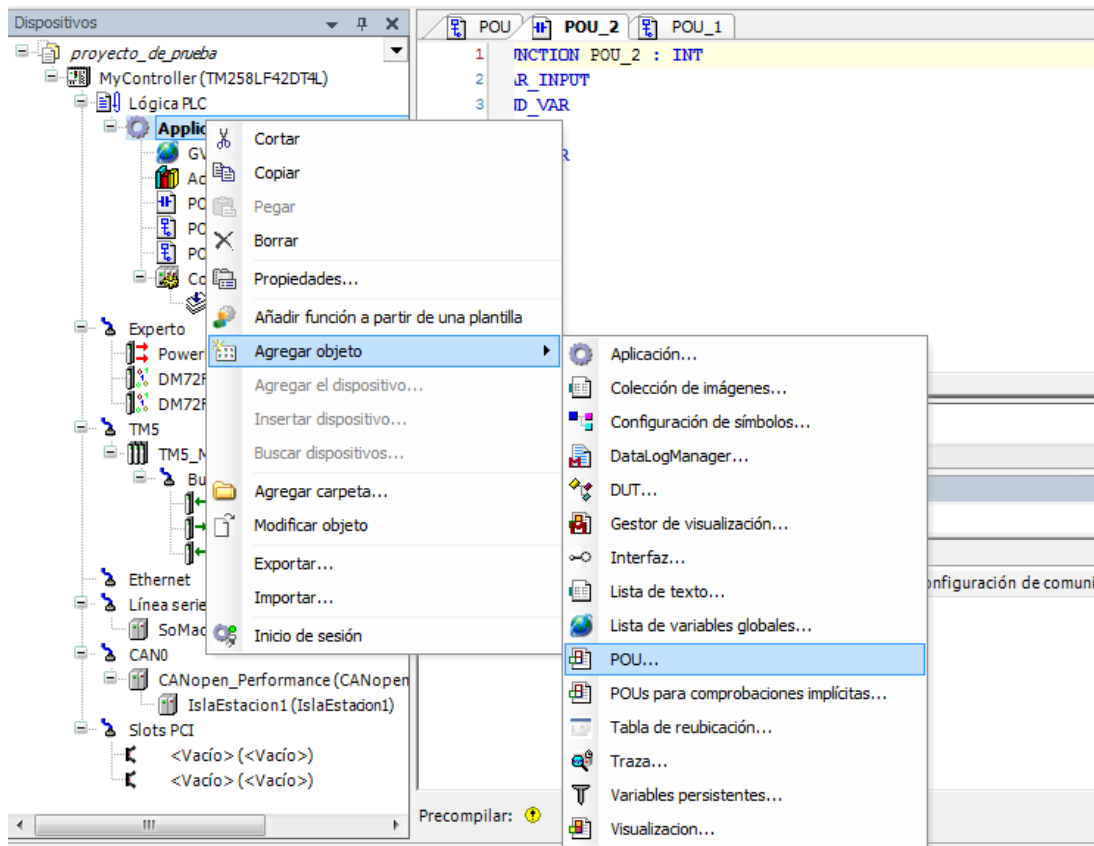


Fig. 229: Agregar Nueva POU

Nos aparecerá la siguiente ventana en la que elegiremos el tipo de sección de trabajo así como el lenguaje de implementación. En nuestro caso, en todas prácticas utilizaremos el lenguaje SFC y 'Programa' como tipo de POU.

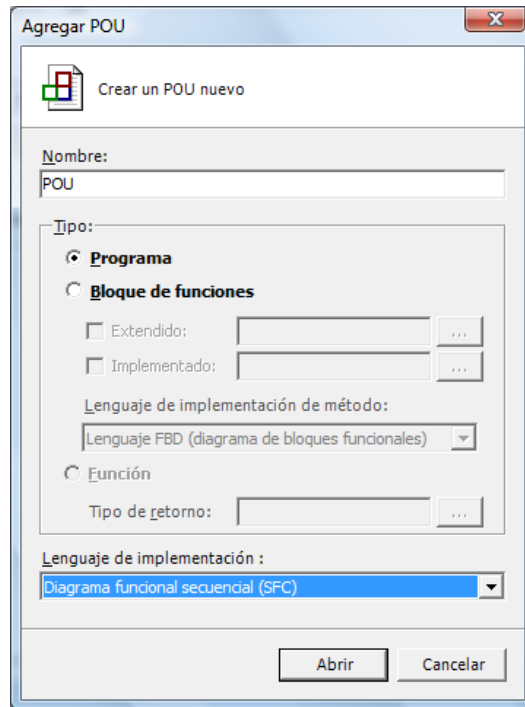


Fig. 230: Nueva POU en SFC

Para agregar la nueva sección de trabajo a la tarea MAST basta con hacer doble click sobre la tarea MAST que se encuentra en la ventana de dispositivos a la izquierda de la pantalla en la pestaña de programa. Nos aparecerá la siguiente pestaña para la configuración de la tarea y a su vez para agregar la nueva sección de trabajo.

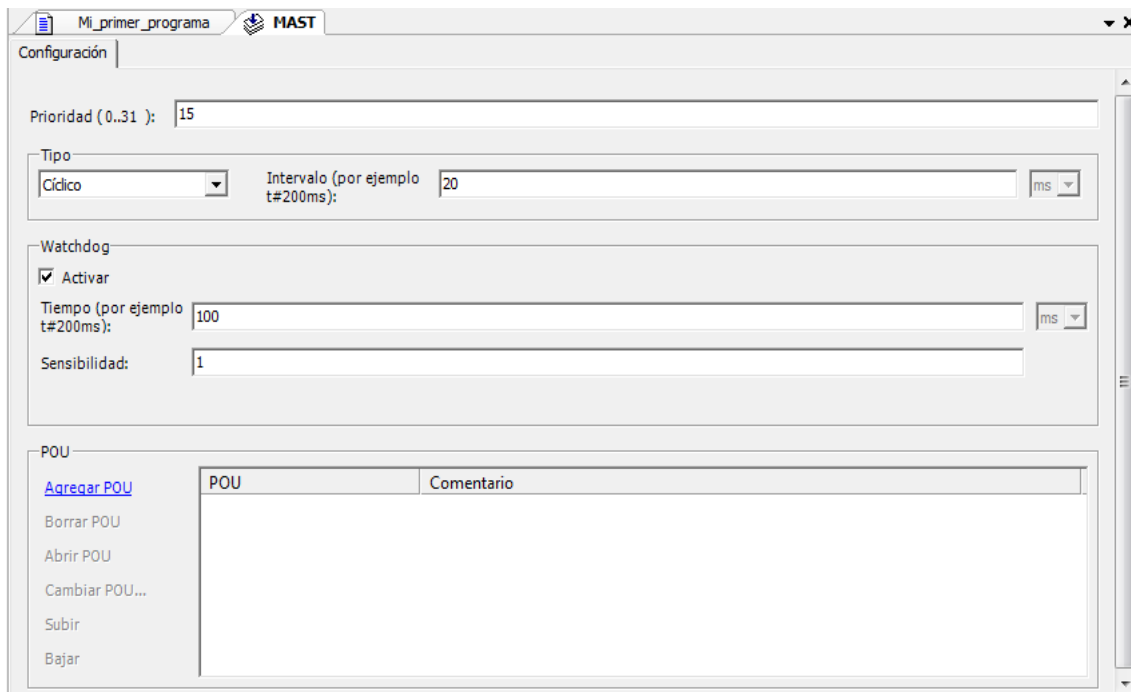


Fig. 231: Agregar POU a Tarea MAST

Haremos click sobre 'Agregar POU', buscaremos la POU deseada y la agregaremos quedándonos algo del siguiente estilo. [Fig. 232]

POU	
POU	Comentario
POU	

Fig. 232: Ejemplo de POU Agregada

Con esto, ya tendremos toda la configuración realizada y la sección de programa ya creada por lo que el siguiente paso ya será comenzar a programar las estaciones.

Esta práctica es de vital importancia ya que habrá que seguir estos pasos en prácticas posteriores para la configuración de las diferentes estaciones.

7.2- PRÁCTICA 1. PROGRAMACIÓN ESTACIÓN 3

Esta primera práctica consiste en la programación de la estación 3 de la célula de fabricación disponible en el laboratorio. Esta estación es una de las cuatro que conforman la zona de fabricación de la célula y su cometido es el montaje de las culatas. En primer lugar habrá que crear la nueva aplicación como se explicó en la práctica de iniciación.

A continuación se detallan las condiciones iniciales y el funcionamiento que debe tener dicha estación:

- **Posicionamiento previo:** en primer lugar hay que realizar un posicionamiento previo de la estación ya que puede encontrarse en cualquier punto de la misma. El brazo debe encontrarse en la posición 'atrás' listo para coger una culata con la pinza arriba abierta. Además, el empujador de culatas deberá encontrarse sin extender.

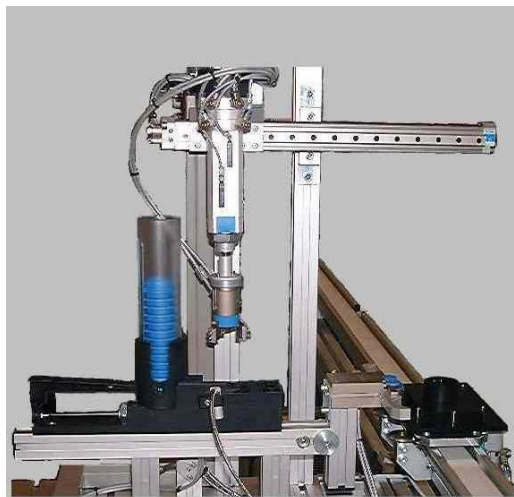


Fig. 233: Estación 3

-
- **Funcionamiento deseado** [Fig. 235]: el inicio del proceso tiene lugar al pulsar el botón MARCHA de la botonera.



Fig. 234: Botonera Estación 3

- I. Sacar una tapa del alimentador hasta que el sensor óptico la detecte. Habrá que mantener la acción durante un segundo más para que la pieza ocupe su posición correcta.
- II. Bajar el brazo con la pinza abierta
- III. Cerrar la pinza y esperar durante un segundo para asegurar que cogemos bien la tapa.
- IV. Subir el brazo hasta arriba y llevarlo hacia delante hasta la posición de las piezas.
- V. Bajar el brazo con la tapa.
- VI. Sujetar la pieza con la mordaza y roscar la tapa sobre esta misma. Se trata de un giro de derechas, en el sentido de las agujas del reloj.
- VII. Abrir la pinza y la mordaza.
- VIII. Subir el brazo y desenroscarlo devolviéndolo a su posición de reposo.
- IX. Retroceder el brazo hasta su posición de reposo.

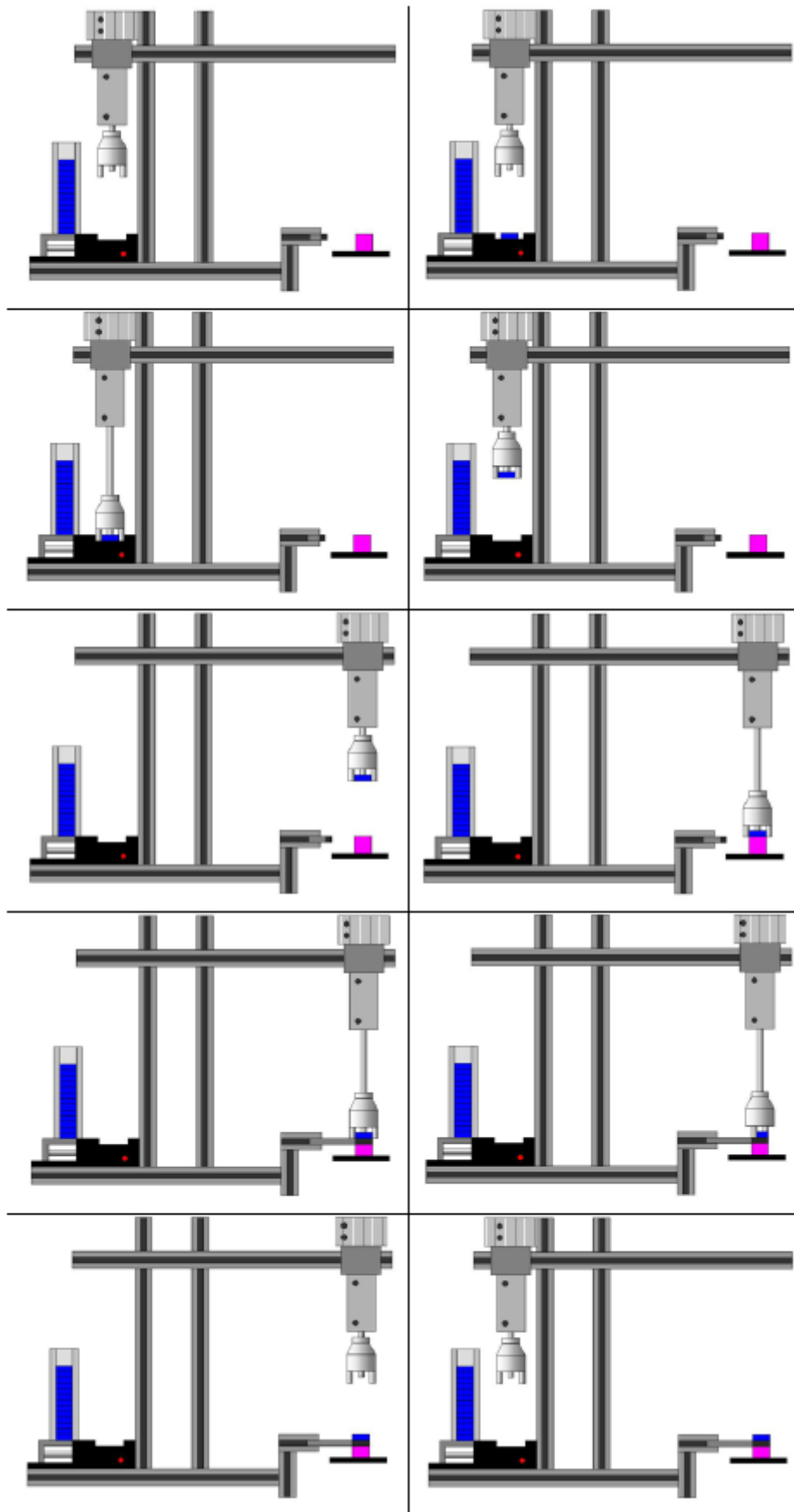


Fig. 235: Funcionamiento Estación 3

- **Situaciones de emergencia:**

- La primera situación de emergencia se corresponde con que ya haya una tapa sacada al iniciar el proceso, es decir el óptico que la detecta está activo.

- La segunda situación de emergencia se corresponde con que no haya tapas en el depósito por lo que el sensor óptico no se activa en un tiempo razonable.

- En ambos casos, habrá que señalar dichas emergencias con la apertura y cierre intermitentes de la pinza y el rearme de la máquina se hará una vez hayan sido solucionados los problemas posicionando la máquina en el estado de reposo. Para ello, habrá que pulsar el botón de 'Rearme' de la botonera.

- Otra situación de emergencia es la pulsación de la seta de emergencia. Cuando esto ocurra el proceso deberá detenerse inmediatamente.

Una vez detallado el funcionamiento de la estación, pasaremos a la definición y descripción de las entradas y salidas que utilizaremos en la programación de la célula.

ENTRADAS		
Dirección	Símbolo	Descripción
%IX19.0	Cinta_atras	Se activa al llegar o estar el brazo atrás.
%IX19.1	Cinta_adelante	Se activa al llegar o estar el brazo adelante.
%IX19.2	Girar_izqda	Se activa al llegar o estar la pinza a la izquierda (Roscar)
%IX19.3	Girar_dcha	Se activa al llegar o estar la pinza a la derecha (Desenroscar)
%IX19.4	Pinza_arriba	Se activa al llegar o estar la pinza arriba.
%IX19.5	Pinza_abajo	Se activa al llegar o estar la pinza abajo.
%IX21.0	Cargador	Se activa cuando detecta culata fuera.
%IX21.1	Emergencia	Seta de emergencia. Se activa al pulsarla.
%IX21.2	Marcha	Pulsador de marcha.
%IX21.3	Ind_int	Interruptor para seleccionar modo de funcionamiento. Activo cuando está en posición INT.
%IX21.4	Cinta_avanza	Pulsador de rearme (Reset)
%IX21.5	Manual_automatico	Interruptor para seleccionar modo de funcionamiento. Activo cuando está en posición de automático.

Fig. 236: Tabla Entradas Estación 3

SALIDAS		
Dirección	Etiqueta	Descripción
%QX4.0	Cinta_avanza	Al activarla, la pinza se mueve hacia adelante. Al desactivarla, permanece en el estado que se encuentra.
%QX4.1	Cinta_retrocede	Al activarla, la pinza se mueve hacia atrás. Al desactivarla, permanece en el estado que se encuentra.
%QX4.2	Roscar	Al activarla, la pinza gira roscando la culata sobre la camisa. Al desactivarla, se produce el movimiento contrario.
%QX4.3	Pinza_baja	Al activarla baja la pinza. Al desactivarla, sube.
%QX4.4	Culata	Al activarla sacamos una culata. Al desactivarla, el empujador vuelve hacia atrás.
%QX4.5	Fijar	Al activarla, la mordaza se cierra sobre la camisa. Al desactivarla, se abre la mordaza.
%QX5.0	Pinza	Al activarla, cerramos la pinza. Al desactivarla, se abre.

Fig. 237: Tabla Salidas Estación 3

Para declarar tanto las entradas como las salidas, basta con ir al menú general de la pestaña de 'Programa' y seleccionar la opción 'Declarar variable' en la pestaña 'Edición'. Este es el procedimiento que se seguirá también en prácticas posteriores.

Fig. 238: Declaración de Variables

7.3- PRÁCTICA 2. PROGRAMACIÓN ESTACIÓN 6

La segunda práctica del curso consiste en la programación de la estación 6 de la célula de fabricación disponible en el laboratorio. Esta estación forma parte de la zona de expedición de la célula de fabricación y su cometido es la carga de bases. En primer lugar habrá que crear la nueva aplicación como se explicó en prácticas anteriores.

A continuación se detallan las condiciones iniciales y el funcionamiento que debe tener dicha estación:

- **Posicionamiento previo:** la posición inicial de reposo será la siguiente. El alimentador de placas deberá encontrarse atrás. Además, el brazo deberá estar encima del alimentador de placas blancas, visto desde la botonera arriba, atrás y a la derecha.

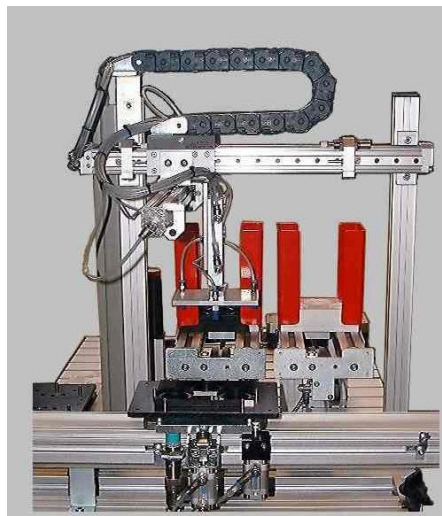


Fig. 239: Estación 6

- **Funcionamiento deseado** [Fig. 241]: el inicio del proceso tiene lugar al pulsar el botón MARCHA de la botonera y el conmutador IND_INT selecciona si la base es blanca o negra.



Fig. 240: Botonera Estación 6

- I. Sacar una placa mediante el alimentador.
- II. Posicionar el brazo sobre la placa dependiendo del tipo.
- III. Bajar el brazo y coger la placa mediante succión hasta que haga vacío.
- IV. Subir el brazo y posicionar enfrente del transbordador del palet.
- V. Colocar el brazo sobre el palet y bajar la placa.
- VI. Soltar la placa sobre el palet dejando de hacer vacío.
- VII. Devolver el brazo a su posición de reposo.

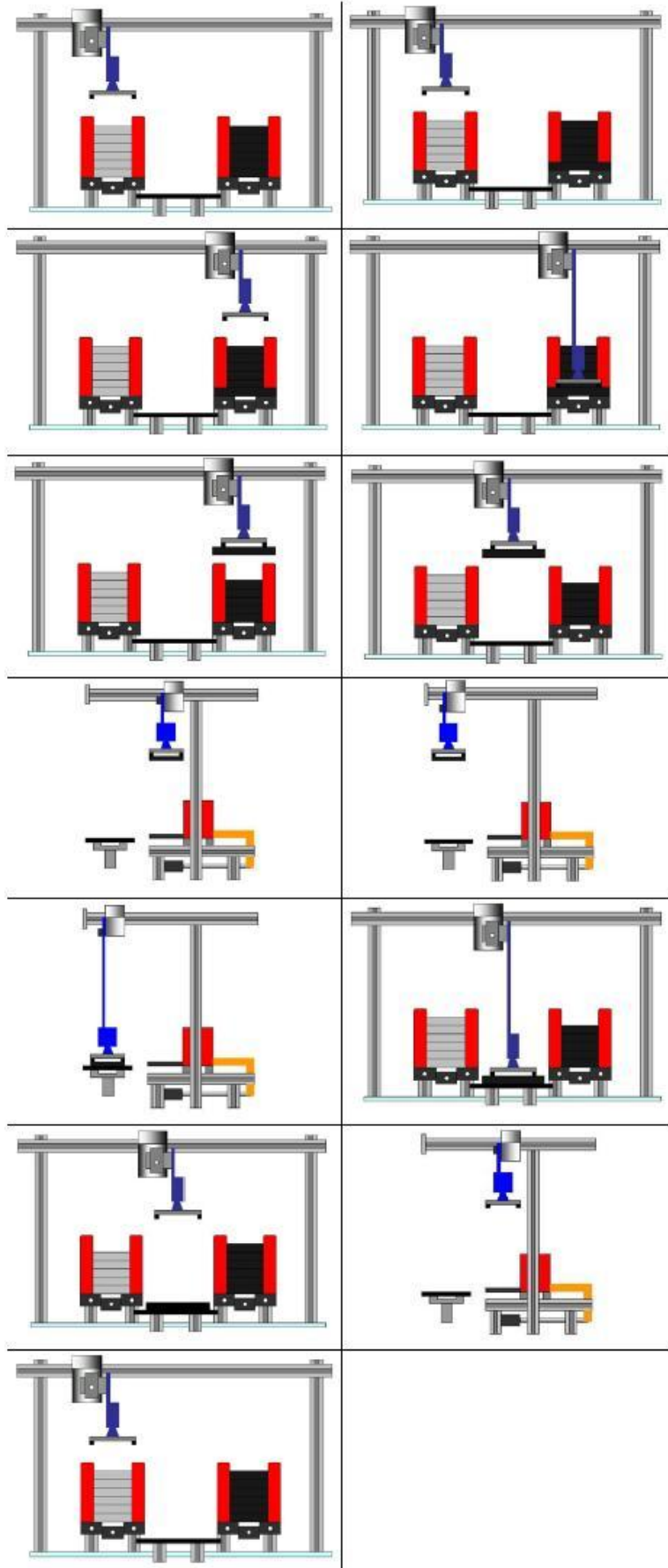


Fig. 241: Funcionamiento Estación 6

- **Situaciones de emergencia:**
 - El óptico detecta que ya hay una base sacada al iniciar el proceso.
 - No quedan bases en el depósito por lo que no se activa el óptico en un tiempo razonable.
 - La pieza está mal colocada y no se hace el vacío.
 - Hay una pieza en el palet de salida y no se activa 'brazo_abajo' en un tiempo razonable.
 - En todos los casos, habrá que señalar dichas emergencias mediante la activación y desactivación de la luz de alarma de forma intermitente y el rearme de la máquina se hará mediante el botón de 'Rearme' de la botonera posicionando la máquina en el estado de reposo.
 - Otra situación de emergencia es la pulsación de la seta de emergencia. Cuando esto ocurra el proceso deberá detenerse inmediatamente.

Una vez detallado el funcionamiento de la estación, pasaremos a la definición y descripción de las entradas y salidas que utilizaremos en la programación de la célula.

ENTRADAS (I)		
Dirección	Símbolo	Descripción
%IX19.0	Alimentador_izqdo_atras	Se activa al llegar o estar el desapilador izquierdo atrás.
%IX19.1	Optico_alimentador_izqdo	Se activa al llegar o estar la placa izquierda sobre el sensor óptico.
%IX19.2	Alimentador_dcho_atras	Se activa al llegar o estar el desapilador derecho atrás.
%IX19.3	Optico_alimentador_dcho	Se activa al llegar o estar la placa derecha sobre el sensor óptico.
%IX19.4	Placa_arriba	Se activa al llegar o estar el brazo arriba.
%IX19.5	Placa_abajo	Se activa al llegar o estar el brazo abajo.
%IX21.0	Placa_dcha	Se activa al llegar o estar el brazo a la derecha.
%IX21.1	Placa_izqda	Se activa al llegar o estar el brazo a la izquierda.
%IX21.2	Emergencia	Seta de emergencia. Se activa al pulsarla.
%IX21.3	Marcha	Pulsador de marcha.
%IX21.4	Manual_automatico	Interruptor para seleccionar el modo de funcionamiento. Activa cuando está en posición automático.
%IX21.5	Rearme	Pulsador de rearme. (Reset)
%IX23.0	Ind_int	Interruptor para seleccionar el modo de funcionamiento. Activa cuando está en posición INT.
%IX23.1	Placa_atras	Se activa al llegar o estar el brazo atrás.

Fig. 242: Tabla Entradas Estación 6 (I)

ENTRADAS (II)		
Dirección	Etiqueta	Descripción
%IX23.2	Placa_adelante	Se activa al llegar o estar el brazo delante
%IX23.3	Vacio	Se activa al detectarse el vacío en las ventosas que succionan la placa.

Fig. 243: Tabla Entradas Estación 6 (II)

SALIDAS		
Dirección	Etiqueta	Descripción
%QX4.0	Alimentador_izqdo	Al activarla, el desapilador de la izquierda avanza empujando una placa. Al desactivarla, retrocede.
%QX4.1	Alimentador_drcho	Al activarla, el desapilador de la derecha avanza empujando una placa. Al desactivarla, retrocede.
%QX4.2	Coge_placa	Al activarla, se pone en marcha el proceso de succión de la placa. Al desactivarla, deja de hacer vacío.
%QX4.3	Bajar	Al activarla, baja la pinza. Al desactivarla, sube.
%QX4.4	Izqda	Al activarla, el brazo se mueve hacia la izquierda. Al desactivarla, permanece en el estado que se encuentra.
%QX4.5	Dcha	Al activarla, el brazo se mueve hacia la derecha. Al desactivarla, permanece en el estado que se encuentra.
%QX5.0	Adelante	Al activarla, el brazo se mueve hacia delante. Al desactivarla, permanece en el estado que se encuentra.
%QX5.1	Atras	Al activarla, el brazo se mueve hacia detrás. Al desactivarla, permanece en el estado que se encuentra.
%QX5.2	Alarma	Al activarla, se enciende el indicador luminoso.

Fig. 244: Tabla Salidas Estación 6

7.4- PRÁCTICA 3. PROGRAMACIÓN ESTACIÓN 4

En esta tercera práctica, nos vamos a ocupar de la programación de la estación 4 de la célula de fabricación. Dicha estación forma parte de la zona de fabricación de la célula y su cometido es la verificación de los cilindros.

A continuación se detallan las condiciones iniciales y el funcionamiento que debe tener dicha estación:

- **Posicionamiento previo:**
 - La pinza debe encontrarse arriba, a la derecha y sobre el palet.
 - El verificador debe encontrarse arriba.
 - El brazo basculante debe estar sobre la rampa.
 - Todas las salidas tienen que estar desactivadas.
 - Además, existe la posibilidad de que haya una pieza en el verificador. Si es así, habrá que desecharla antes de poder atender pedidos. Para ello, la detectaremos con el brazo basculante creando vacío.



Fig. 245: Estación 4

- **Funcionamiento deseado** [Fig. 247, Fig. 248]:
 - El inicio del proceso tiene lugar al pulsar el botón MARCHA de la botonera.



Fig. 246: Botonera Estación 4

- El conmutador IND_INT de la botonera selecciona si:
 - Verificar pieza: se realiza el proceso por completo.
 - Pasar sin verificar: no se verifica la pieza. Simplemente se utiliza el verificador como punto intermedio para dejar la pieza.

- Características del proceso de verificación de las piezas:

- Hay que inyectar aire en el verificador durante un tiempo determinado (aproximadamente unos 10 seg.)
- Medida analógica:
 - si es mayor de 10.000, la pieza es buena.
 - si es menor de 10.000, la pieza es desechable.
- Si la pieza es mala, habrá que bascular la rampa y desplazar el brazo para evitar cualquier tipo de atasco.

- Ciclo de verificación:

- I. Recoger la pieza mediante el brazo y sus ventosas haciendo vacío.
- II. Subir y después girar el brazo depositando la pieza sobre el verificador.
- III. Devolver el brazo a su posición inicial.
- IV. Bajar el verificador.
- V. Inyectar aire durante unos segundos para tomar la medida.
- VI. Dejar de inyectar aire y subir el verificador.
- VII. Colocar el brazo giratorio de una ventosa sobre la pieza y hacer vacío.
- VIII. Accionar el expulsador y simultáneamente girar el brazo hasta la rampa.
- VIII. Si el resultado de la verificación anterior ha sido positivo, la rampa conducirá la pieza sobre la cinta. En caso contrario, la rampa dirige la pieza hacia la cubeta de piezas defectuosas.

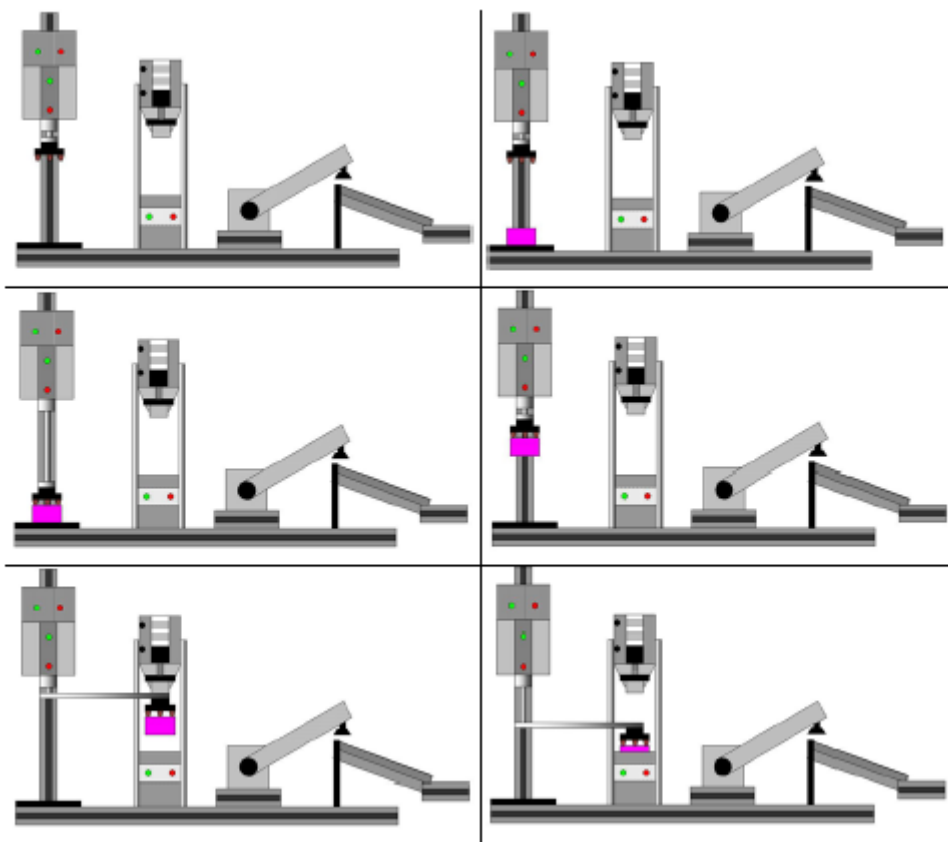


Fig. 247: Funcionamiento Estación 4. Parte I

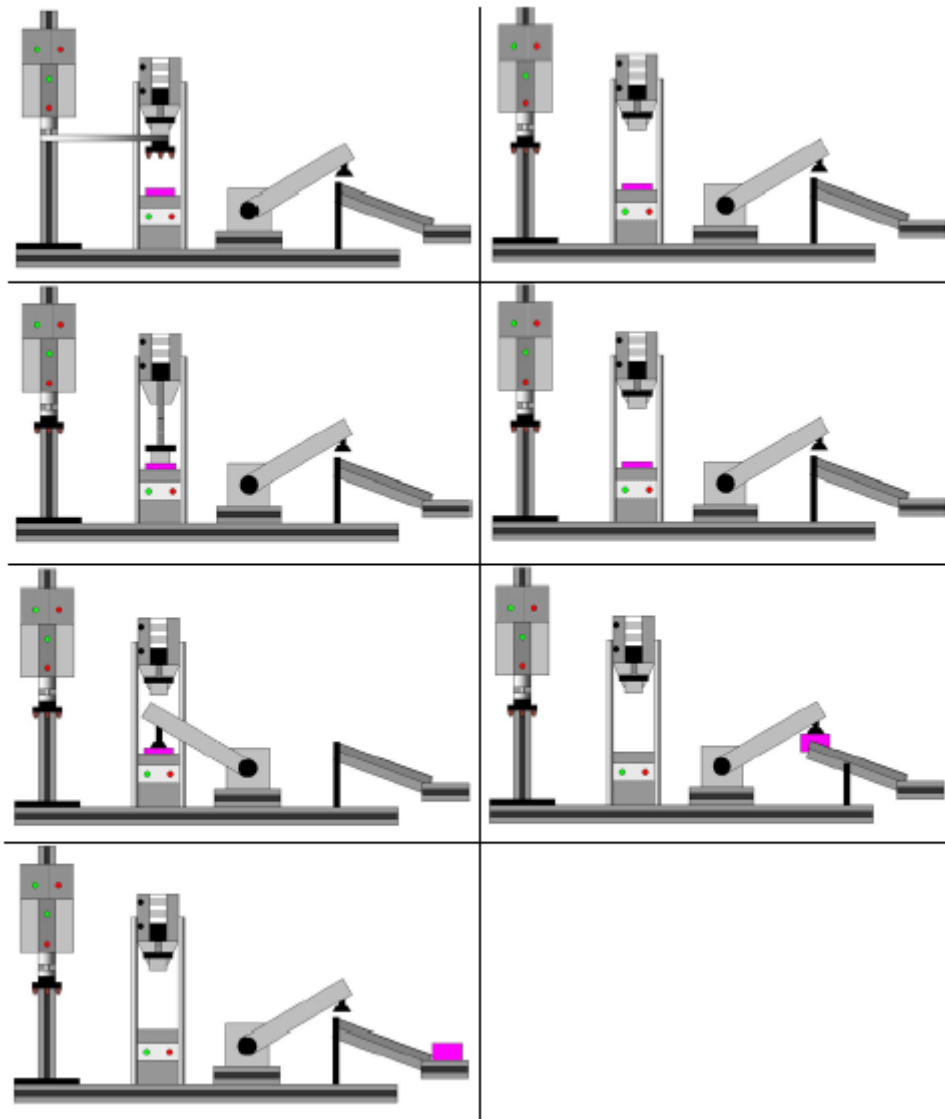


Fig. 248: Funcionamiento Estación 4. Parte II

- **Situaciones de emergencia:**
 - No hay una pieza a verificar por lo que no hace vacío la ventosa de la pinza en un tiempo razonable.
 - Ha ocurrido alguna incidencia en el verificador al no ser capaz de coger la pieza en un tiempo razonable es decir, no hace vacío en la ventosa del brazo basculante.
 - La pieza del verificador se queda enganchada ya que hacemos vacío pero el brazo no llega al extremo de la rampa en un tiempo razonable.
 - Perdemos la pieza en el brazo o la pinza durante el transporte dejándose de hacer vacío.
 - En todos los casos, habrá que señalar dichas emergencias mediante la activación y desactivación del vacío en la pinza de forma intermitente y el rearme de la máquina se hará mediante el botón de 'Rearme' de la botonera posicionando la máquina en el estado de reposo.

- Otra situación de emergencia es la pulsación de la seta de emergencia. Cuando esto ocurra el proceso deberá detenerse inmediatamente.

Una vez detallado el funcionamiento de la estación, pasaremos a la definición y descripción de las entradas y salidas que utilizaremos en la programación de la célula.

ENTRADAS		
Dirección	Símbolo	Descripción
%IX20.0	Verificador_abajo	Se activa al llegar o estar el verificador abajo.
%IX20.1	Verificador_arriba	Se activa al llegar o estar el verificador arriba.
%IX20.2	Gira_dcha	Se activa al llegar o estar la pinza a la derecha, sobre el palet.
%IX20.3	Gira_izqda	Se activa al llegar o estar la pinza a la izquierda, sobre el verificador.
%IX20.4	Cilindro_arriba	Se activa al llegar o estar el brazo arriba.
%IX20.5	Cilindro_abajo	Se activa al llegar o estar el brazo abajo.
%IX22.0	Vacio_pinza	Se activa al crearse o mantenerse el vacío en las dos ventosas del brazo.
%IX22.1	Emergencia	Seta de emergencia. Se activa al pulsarla.
%IX22.2	Marcha	Pulsador de marcha.
%IX22.3	Ind_int	Interruptor para seleccionar el modo de funcionamiento. Activa cuando está en posición INT.
%IX22.4	Rearme	Pulsador de rearme. (Reset)
%IX24.0	Pieza_fuera	Se activa al llegar o estar el brazo basculante sobre la rampa.
%IX24.1	Sacar_pieza	Se activa al llegar o estar el brazo basculante sobre el verificador.
%IX24.2	Vacio_pieza	Se activa al crearse o mantenerse el vacío en el brazo basculante.
%IX24.3	Bascula_pieza	Se activa al llegar o estar la rampa inclinada para tirar la pieza hacia el depósito de piezas defectuosas.
%IW17	Medidor_analogico	Da una medida proporcional a la altura alcanzada por el émbolo al inyectar aire en el cilindro.

Fig. 249: Tabla Entradas Estación 4

SALIDAS(I)		
Dirección	Símbolo	Descripción
%QX4.0	Gira_izqda	Al activarla, el brazo gira hacia el verificador. Al desactivarla, no cambia.
%QX4.1	Gira_dcha	Al activarla, el brazo gira hacia el palet de la cinta. Al desactivarla, no cambia.
%QX4.2	Cilindro_baja	Al activarla, el brazo baja. Al desactivarla, el brazo sube.
%QX4.3	Verificador_baja	Al activarla, el verificador baja. Al desactivarla, el verificador sube.

SALIDAS(II)		
Dirección	Símbolo	Descripción
%QX4.4	Inyecta	Al activarla, inyectamos aire en el depósito de verificación. Al desactivarla, se para la inyección.
%QX4.5	Expulsa	Al activarla, se activa un cilindro que empuja la pieza que se encuentra en el verificador. Al desactivarla, se para el empuje.
%QX5.0	Vacio_en_pinza	Al activarla, hacemos vacío en el brazo que lleva la pieza del palet al verificador. Al desactivarla, dejamos de hacer vacío.
%QX5.1	Vacio_en_pieza	Al activarla, hacemos vacío en el brazo que lleva la pieza del verificador a la rampa. Al desactivarla, dejamos de hacer vacío.
%QX5.2	Saca_pieza	Al activarla, el brazo basculante se coloca sobre el verificador. Al desactivarla, no cambia la posición.
%QX5.3	Expulsa_pieza	Al activarla, el brazo basculante se coloca sobre la rampa. Al desactivarla, no cambia la posición.
%QX5.4	Bascular	Al activarla, cambia la inclinación de la rampa para que la pieza caiga sobre el depósito de piezas defectuosas. Al desactivarlas, la rampa vuelve a su posición de reposo sobre la cinta.

Fig. 250: Tabla Salidas Estación 4

7.5- PRÁCTICA 4. PROGRAMACIÓN CONJUNTA DE ESTACIONES.

La cuarta y última práctica del curso consiste en la programación conjunta de las estaciones 1 y 6 de la célula de fabricación.

Para ello, deberemos realizar los siguientes pasos antes de comenzar con la programación de las estaciones:

- Una vez creados el proyecto y agregados el autómatas y el maestro CANopen, la única diferencia con el resto de prácticas es que ahora tendremos que agregar dos dispositivos al maestro CAN, en este caso los correspondientes a las islas 1 y 6.

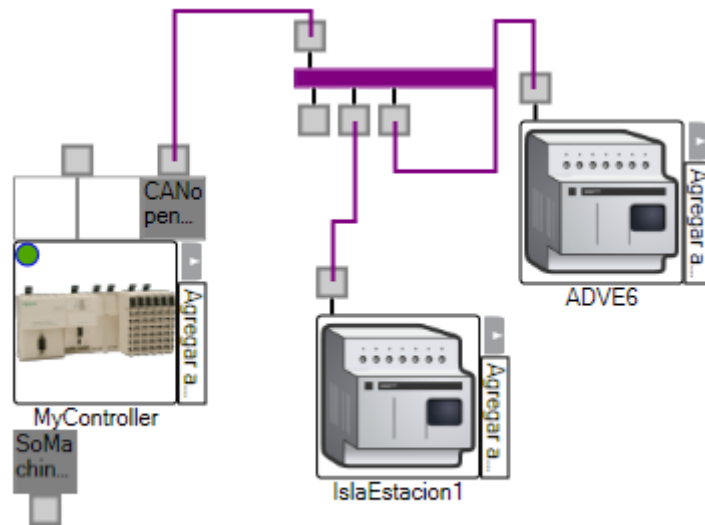


Fig. 251: Conexión Islas Advantys

- Haremos doble click sobre cada uno de los dispositivos agregados y nos aseguraremos que cada uno tiene un Node ID distinto. Por ejemplo la isla 1, con un Node=1 y la isla 6 con un Node=2.

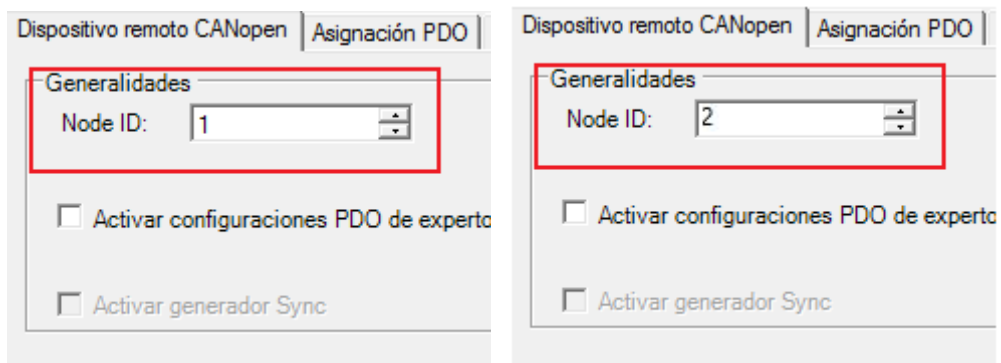


Fig. 252: Configuración Esclavos Islas

Además, tendremos que situar manualmente la pestaña correspondiente a cada isla en el nodo que le hayamos marcado por programa.

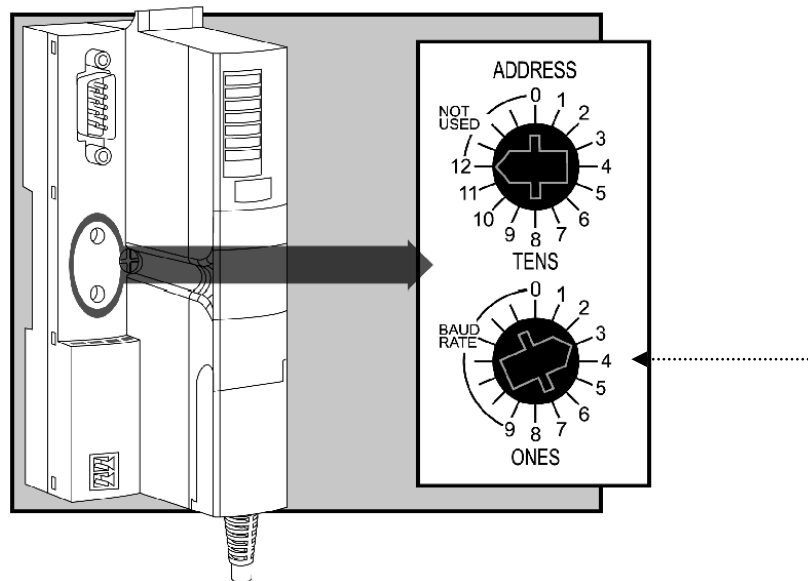


Fig. 253: Esclavos Islas

- Debemos crear dos secciones de trabajo, una para cada estación. Habrá que añadir ambas a la tarea MAST como habíamos visto en prácticas anteriores.

Una vez realizados los pasos anteriores, la ventana de dispositivos deberá tener un aspecto similar al siguiente con ambas secciones de programa creadas:

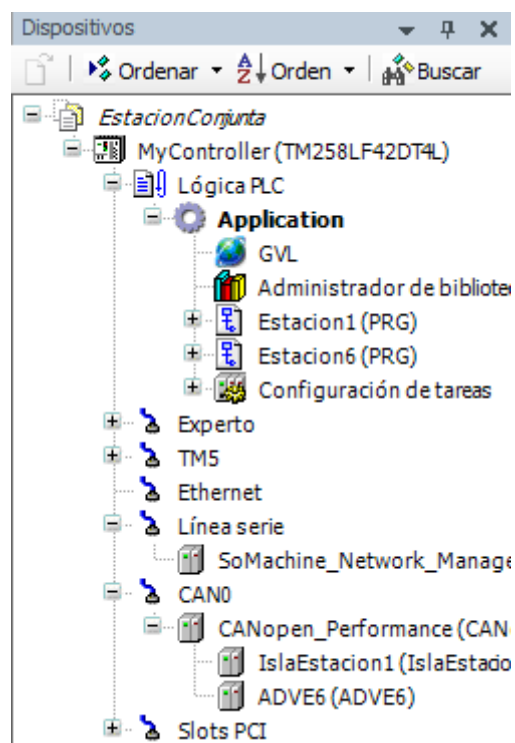


Fig. 254: Ventana Dispositivos. POU's

A continuación se detallarán las condiciones iniciales y el funcionamiento que deben tener dichas estaciones.

❖ **ESTACIÓN 1:**

• **Posicionamiento previo:**

- El brazo debe encontrarse en la posición 'atrás' y a la izquierda listo para coger una culata con la pinza arriba abierta.
- El empujador de camisas deberá encontrarse sin extender.
- El lector de camisas que se encuentra sobre el palet deberá estar atrás.



Fig. 255: Estación 1

• **Funcionamiento deseado** [Fig. 257]:

- El inicio del proceso tiene lugar al pulsar el botón MARCHA de la botonera.



Fig. 256: Botonera Estación 1

- El conmutador IND_INT de la botonera selecciona si:
 - Identificar camisa metálica.
 - Identificar camisa roja.
- Ciclo de identificación y carga de camisas:
 - I. Sacar camisa del depósito. Se encuentran distribuidas aleatoriamente. La camisa estará fuera del depósito cuando el sensor la detecte.
 - II. Identificar la camisa con los sensores.
 - III. Bajar el brazo y coger la camisa.
Si el resultado de la comprobación ha sido *negativo*, subir el brazo con la camisa y desplazarlo hacia la derecha sobre el depósito de piezas desechadas. Soltar la pieza y volver a sacar una nueva camisa.

Si el resultado de la comprobación ha sido *positivo*, subir el brazo con la camisa cogida y continuar con los pasos siguientes.

- IV. Mover el brazo hacia delante sobre el palet.
- V. Bajar la pinza, soltar la camisa.
- VI. Subir el brazo y llevarlo a la posición de reposo.
- VII. Lectura de la tapa mediante el lector.

Si el resultado es *positivo*, la camisa se quedará en el palet y el brazo volverá al estado de repodo.

Si el resultado es *negativo*, habrá que volver a recoger la camisa depositada sobre el palet y llevarla al depósito de piezas desechadas volviendo el brazo a la posición de reposo.

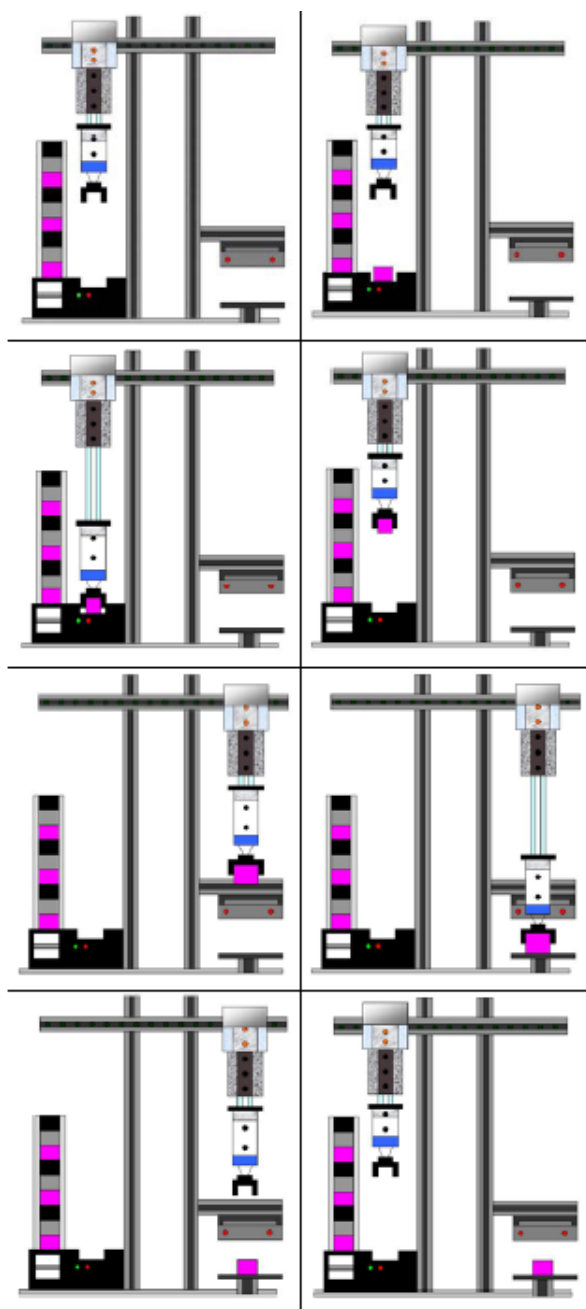


Fig. 257: Funcionamiento Estación 1

- **Situaciones de emergencia:**

- La primera situación de emergencia se corresponde con que ya haya una camisa sacada al iniciar el proceso, es decir el óptico que la detecta está activo.

- La segunda situación de emergencia se corresponde con que no haya camisas en el depósito por lo que el sensor óptico no se activa en un tiempo razonable.

- En ambos casos, habrá que señalar dichas emergencias con la apertura y cierre intermitentes de la pinza y el rearme de la máquina se hará una vez hayan sido solucionados los problemas posicionando la máquina en el estado de reposo. Para ello, habrá que pulsar el botón de 'Rearme' de la botonera.

- Otra situación de emergencia es la pulsación de la seta de emergencia. Cuando esto ocurra el proceso deberá detenerse inmediatamente.

Una vez detallado el funcionamiento de la estación, pasaremos a la definición y descripción de las entradas y salidas que utilizaremos en la programación de la célula.

ENTRADAS		
Dirección	Símbolo	Descripción
%IX19.0	Cinta_atras	Se activa al llegar o estar la cinta atrás.
%IX19.1	Cinta_adelante	Se activa al llegar o estar la cinta adelante.
%IX19.2	Pinza_izqda	Se activa al llegar o estar la pinza a la izquierda.
%IX19.3	Pinza_dcha	Se activa al llegar o estar la pinza a la derecha, sobre el depósito.
%IX19.4	Pinza_arriba	Se activa al llegar o estar el brazo arriba.
%IX19.5	Pinza_abajo	Se activa al llegar o estar el brazo abajo.
%IX21.0	Cargador_adelante	Se activa al llegar o estar el cargador adelante.
%IX21.1	Cargador_atras	Se activa al llegar o estar el cargador atrás.
%IX21.2	Emergencia	Seta de emergencia. Se activa al pulsarla.
%IX21.3	Marcha	Pulsador de marcha.
%IX21.4	Manual_automatiko	Interruptor para seleccionar modo de funcionamiento. Activo cuando está en posición de automático.
%IX21.5	Ind_int	Interruptor para seleccionar el modo de funcionamiento. Activa cuando está en posición INT.
%IX23.0	Rearme	Pulsador de rearme. (Reset)
%IX23.1	Capacitivo_camisa	Se activa con cualquier pieza ya sea negra roja o metálica.
%IX23.2	Optico_camisa	Se activa cuando la pieza es roja o metálica.
%IX23.3	Inductivo_camisa	Se activa cuando la pieza sólo es metálica.
%IX23.4	Lector_adelante	Se activa al llegar o estar el lector adelante.
%IX23.5	Lector_atras	Se activa al llegar o estar el lector atrás.
%IX25.0	Optico_Lector	Se activa al llegar o estar el lector sobre la camisa.

Fig. 258: Tabla Entradas Estación 1

SALIDAS		
Dirección	Símbolo	Descripción
%QX4.0	Cinta_avanza	Al activarla, la cinta avanza. Al desactivarla, no cambia.
%QX4.1	Cinta_retrocede	Al activarla, la cinta retrocede. Al desactivarla, no cambia.
%QX4.2	Pinza_fuera	Al activarla, el brazo se mueve hacia la derecha. Al desactivarla, no cambia.
%QX4.3	Pinza_dentro	Al activarla, el brazo se mueve hacia la izquierda. Al desactivarla, no cambia.
%QX4.4	Pinza_baja	Al activarla, el brazo baja. Al desactivarla, el brazo sube.
%QX4.5	Cargador	Al activarla sacamos una camisa. Al desactivarla, el empujador vuelve hacia atrás.
%QX5.0	Pinza	Al activarla, cerramos la pinza. Al desactivarla, se abre.
%QX5.1	Lector	Al activarla, el lector se mueve hacia la derecha. Al desactivarla, el brazo vuelve a la izquierda.

Fig. 259: Tabla Salidas Estación 1

❖ **ESTACIÓN 6:**

Ya que la estación 6 fue programada en la práctica número 2, no es necesario volver a programarla. Basta con abrir el proyecto de dicha práctica y hacer clic con el botón derecho sobre el nombre de la POU en la ventana de dispositivos. Le daremos a la opción copiar e iremos al proyecto de la práctica número 4 y pegaremos dicha sección en el mismo lugar. Debe quedarnos algo con el siguiente aspecto:

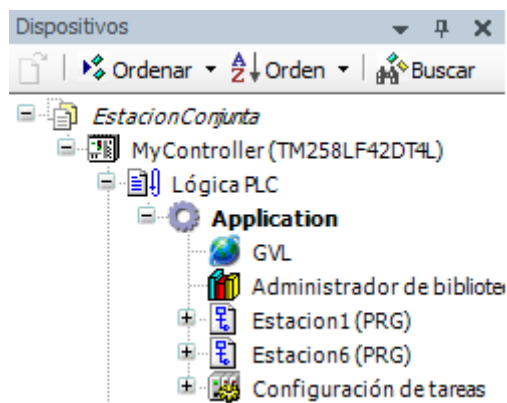


Fig. 260: Dispositivos. POUs

La única modificación que habrá que realizar con respecto a la práctica número dos será el direccionamiento de las entradas y salidas ya que en este caso serán distintos.

En la Práctica 2 teníamos...	...y ahora tendremos en la Práctica 4
%IX19	%IX29
%IX21	%IX31
%IX23	%IX33
%QX4	%QX6
%QX5	%QX7

Fig. 261: Tabla Modificaciones Entradas y Salidas