

TRABAJO FIN DE GRADO INGENIERÍA ELÉCTRICA

DISEÑO DEL CONTROL DE POSICIONANDO APLICADO A UN ALMACÉN INTELIGENTE

Autor

Alejandro Ballesteros García

Tutor

Ángel Sapena Bañó

Cotutor

Manuel Pineda Sánchez

Valencia, junio del 2021

AGRADECIMIENTOS

Antes de comenzar, me gustaría agradecer su labor a los profesores que, desde hace ya 18 años, han participado en mi formación, desde enseñarme a leer hasta llegar a ser ingeniero. Todo esto se debe a la labor docente que todos ellos han desempeñado a lo largo de los años.

Especialmente me gustaría mencionar a Manuel Zafrilla, José Bermúdez y Pedro Alfaro, que me motivaron a decidirme a estudiar ingeniería, a los responsables de la mención Ángel Sapena, Juan Pérez y Rubén Puche por la dedicación a su trabajo y a Jordi Burriel por ayudarme tanto como amigo como profesor.

También me gustaría agradecer a mi familia la oportunidad de haber podido estudiar fuera y por apoyarme siempre en todas mis decisiones. A mi pareja y amigos por su apoyo incondicional y por ayudarme siempre que lo he necesitado.

RESUMEN

Los procesos industriales siempre se han caracterizado por la repetición de tareas sencillas para poder fabricar en masa productos de una calidad satisfactoria. La automatización busca sustituir la mano de obra humana para liberar a las personas de realizar tareas simples que solo requieren repetitividad o de trabajos que por su naturaleza puedan ser peligrosos para el trabajador. En el caso de un almacén, se consigue que el operario pase de realizar esfuerzos físicos que pueden resultar lesivos a tener un trabajo de supervisión mucho más cómodo. Por otra parte, la gestión de los procesos mejora debido a la informatización de los datos.

Con el presente proyecto se busca el diseño e implementación de una función genérica de control de posición aplicada que actúa sobre dos motores accionados de forma independiente y del SCADA correspondiente a la supervisión y planificación del proceso.

Durante este documento se estudiará la viabilidad de la transformación de la tarea del almacenaje en un proceso controlado por un automatismo y se presentarán alternativas a la solución elegida. Así mismo, se justificará la elección tomada. Se llevará a cabo el diseño de esta función teniendo en cuenta la normativa vigente y asegurando que su funcionamiento se producirá de forma segura sin incurrir en riesgos para los trabajadores. A continuación, se parametrizarán los distintos elementos necesarios para su funcionamiento y se integrarán en una red de comunicaciones industriales. Una vez comprobado que las comunicaciones funcionan de forma correcta se procederá a programar el código en el entorno TIA PORTAL de Siemens, seguidamente se diseñará el SCADA que acompañará al proceso y en el cual se podrán supervisar los valores del proceso y actuar sobre el mismo. Una vez el código funcione adecuadamente, se descargará en un PLC S7-1215 el cual controlará los variadores de frecuencia.

Palabras clave: control de posición, SCADA, función genérica, IC 4.0, almacén inteligente, matriz.

ABSTRACT

Industrial processes has been always typified by the cyclic realization of simple task in order to craft quality products in mass numbers. Automatization seeks to remplace workers for machines able to solve the same tasks in a more efficient way. This document will explain the automatization of a storage process.

This project will address the desing and implementation of a generic control function applied to an electric axis and the desing of the correspondent SCADA system.

During this document, the viability of automatizing the storage tasks will be studied. Also alternative solutions will be presented along the justification for the election taken. Moreover, the desing of the function will be acomplished assuring the compliance of the pertinent laws. Next to this, the elements of the control process will be parametrized in order to prepare them to work properly and will be loaded into a profinet network. After testing the project fiability the programming of the function will begin in the specialized software TIA PORTAL. Afterward, it will proceed to the desing of the SCADA system which will work with the control. In the final steps, the programm will be loaded on a real equipement and will be tested until the results are satisfactory.

In the end, the project will be prepared to be included as a part of a bigger system.

Key words: Position control, SCADA, generic function, CI 4.0, Smart wahrehouse, matrix

RESUM

Els processos industrials sempre s'han caracteritzat per la repetició de tasques senzilles per a poder fabricar en massa productes d'una qualitat satisfactòria. L'automatització busca substituir als treballadors per màquines capaces de solucionar les mateixes tasques de forma més eficient. Aquest treball se centra en l'automatització d'un procés d'emmagatzemat.

Amb el present projecte es busca el disseny i implementació d'una funció genèrica de control de posició aplicada que actua sobre dos motors accionats de manera independent i del SCADA corresponent a la supervisió i planificació del procés.

Durant aquest document s'estudiarà la viabilitat de la transformació de la tasca del magatzematge en un procés controlat per un automatisme i es presentaran alternatives a la solució triada. Així mateix, es justificarà l'elecció presa. S'emportarà acabe el disseny d'aquesta funció tenint en compte la normativa vigent i assegurant que el seu funcionament es produirà de manera segura sense incórrer en riscos per als treballadors. A continuació, es parametrizaran els diferents elements necessaris per al seu funcionament i s'integraran en una xarxa de comunicacions industrials. Una vegada comprovat que les comunicacions funcionen de manera correcta es procedirà a programar el codi en l'entorn TIA PORTAL de Siemens, seguidament es dissenyarà el SCADA que acompanyarà al procés i en el qual es podran supervisar els valors del procés i actuar sobre el mateix. Una vegada el codi funcione adequadament, es descarregarà en un PLC S7-1215 el qual controlarà els variadors de freqüència. Com a punt final es prepararà la integració del TFG en un sistema de majors proporcions tal com queda reflectit en la definició de la indústria 4.0.

Paraules clau: control de posició, SCADA, funció genèrica, IC 4.0, magatzem intel·ligent, matriu

DOCUMENTOS CONTENIDOS EN EL TFG

- I. MEMORIA
- II. PLANOS
- III. PLIEGO DE CONDICIONES
- IV. PRESUPUESTO
- V. BIBLIOGRAFÍA
- VI. ANEXOS

ÍNDICE GENERAL

MEMORIA

1. Objeto del proyecto.....	17
2. Alcance.....	19
3. Introducción	20
3.1 Sistemas de almacenamiento inteligente.....	20
3.2 Automatización industrial.....	21
3.3 INDUSTRIA CONECTADA 4.0 (I C 4.0)	23
4. Justificación	26
4.1 Académica	26
4.2 Técnica	26
5. Estudio de viabilidad.....	27
5.1 Normativa.....	27
5.2 Requisitos	27
6. Criterios de selección y justificación de la solución adoptada	30
6.1 Control del proceso	30
6.2 Lenguaje de programación	35
6.3 Elección del motor.....	38
6.4 Control del motor.....	41
6.5 Selección de los elementos de visualización	45
7. Justificación de la solución adoptada.....	47
8. Funcionamiento del programa	51
8.1 Descripción del proceso	51
8.2 Descripción del programa.....	52
9. Programación	59
9.1 Creación de la función.....	59
9.2 Declaración de variables y justificación	64
10. Funcionamiento del SCADA.....	66
11. Conclusiones	76

PLANOS

Plano I. Conexionado eléctrico	81
Plano II. Red Profinet.....	82
Plano III. Conexiones PLC.....	83
Plano IV. Conexiones variador.....	84

PLIEGO DE CONDICIONES

1. Introducción	88
2. Objetivo del pliego de condiciones.....	89
3. Normativa vigente.....	90
4. Condiciones técnicas.....	91
4.1.Cableado eléctrico	91
4.2.Interruptores y pulsadores.....	91
4.3.Pulsadores de emergencia y demás elementos de seguridad.....	92
4.4.Fuente de alimentación	92
4.5.Ordenador personal	92
4.6.Software.....	93
4.7.PLC.....	93
4.8.Variador de frecuencia	93
4.9.Motor.....	94
5. Condiciones constructivas	95
5.1.Montaje eléctrico.....	95
5.2.Pruebas de funcionamiento.....	95
5.3.Comprobación visual del conexionado	95
5.4.Pruebas en tensión	96
5.5.Prueba final.....	96
5.6.Control de calidad	96
6. Condiciones de ejecución	98
6.1.Compra de material.....	98
7. Condiciones económicas	99
7.1.Pagos de los trabajos.....	99
8. Condiciones legales.....	100
8.1.Contrato	100
8.2.Arbitraje y jurisdicción	100
8.3.Impuestos	100
8.4.Rescisión del contrato	101
8.5.Condiciones facultativas.....	101
8.6.Derechos y deberes del contratista	101

PRESUPUESTO

1. Objeto y alcance del presupuesto.....	106
2. Material.....	107
3. Software	108
4. Mano de obra	109
5. Coste total	110

BIBLIOGRAFÍA.....	113
--------------------------	------------

ANEXOS

Anexo I Manual de usuario	118
1. Introducción	118
2. Menú	118
3. Modo automático	120
4. Modo manual.....	122
5. Acusar errores	124
6. Configuración	125
7. Datos.....	126
Anexo II Manual TIA PORTAL	127
1. Introducción	127
2. Generar archivo de programa.....	127
3. Configurar equipos	129
4. Añadir bloques.....	130
5. Herramientas de edición	131
6. Cargar y monitorizar programa	132
Anexo III Manual WinCC.....	133
1. Introducción	133
2. Generar pantallas y plantillas.....	134
3. Herramientas de edición	134
4. Recetas	135
5. Usuarios	136
6. Alarmas	137
7. Idiomas.....	137
Anexo IV Listado de programa.....	135
Anexo V Hoja de datos S7-1200.....	174
Anexo VI Hoja de datos SINAMICS S210.....	181
Anexo VII Hoja de datos motor.....	183

ABREVIATURAS

PLC	Programmable Logic Controller
IP	Internet Protocol
PMSM	Permanent Magnets Synchronous Motor
SCADA	Supervisory Control And Data Adquisition
VSD	Variable Speed Drive
HMI	Human Machine Interface
MC	Motion Control
MES	Manufacturing execution systems
EMI	ElectroMagnetical Interference
ERP	Enterprise Resource Planning
IC 4.0	Industria Conectada 4.0
AC	Alternating Current
DC	Direct Current
PC	Personal Computer
ST	Structured Text
HSC	High Speed Counter

ÍNDICE DE FIGURAS

Figura 1 Almacén inteligente.....	20
Figura 2 Transelevador industrial. Fuente Web Principal MECALUX	21
Figura 3 Pirámide de la automatización	22
Figura 4 Etapas de las revoluciones industriales. Fuente: Página IC 4.0.gob	24
Figura 5 PLC SIEMENS. Fuente: Web Principal Siemens.....	30
Figura 6 Lógica cableada (izquierda) / Esquemas de potencia y control (derecha)	33
Figura 7 Ejemplo de sistema embebido	34
Figura 8 Ejemplo texto estructurado.....	35
Figura 9 Ejemplo listado de instrucciones	36
Figura 10 Ejemplo diagrama Ladder	37
Figura 11 Ejemplo Bloques de función.....	37
Figura 12 Motor síncrono de imanes permanentes Fuente: Web Principal Siemens ..	38
Figura 13 Servomotor DC	40
Figura 14 Variador de Frecuencia ABB Fuente: Web Principal ABB	42
Figura 15 Ejemplo pantalla SCADA	45
Figura 16 Ejemplo Panel sinóptico	46
Figura 17 PLC elegido Fuente: Web Principal Siemens	48
Figura 18 Variador y motor elegido Fuente: Web Principal Siemens.....	49
Figura 19 Solución adoptada	50
Figura 20 Ejemplo transelevador	51
Figura 21 Flujograma carga	53
Figura 22 Flujograma función carga.....	54
Figura 23 Flujograma descarga	55
Figura 24 Flujograma función descarga.....	56
Figura 25 Flujograma JOG.....	57
Figura 26 Flujograma retorno.....	57
Figura 27 Flujograma manual	58
Figura 28 Ejemplo FBs.....	59
Figura 29 Funciones Motion Control	60
Figura 30 Estructura función del eje.....	60
Figura 31 Bloque Almacén.....	61
Figura 32 Fragmento código del bloque.....	61
Figura 33 Bloque Control Manual.....	62
Figura 34 Bloque Modo Carga	62
Figura 35 Bloque Modo Descarga.....	63
Figura 36 Datos almacén	64
Figura 37 Datos recetas.....	65
Figura 38 Estructuras de la función.....	65
Figura 39 Plantilla SCADA	67
Figura 40 Ejemplo Animación	67
Figura 41 Menú Principal	68
Figura 42 Flujograma Menú	68
Figura 43 Recetas HMI	69
Figura 44 Pantalla de Configuración	69
Figura 45 Modo Manual	70
Figura 46 Flujograma Manual	71

Figura 47 Pantalla Automático	72
Figura 48 Flujograma Automático	73
Figura 49 Interfaz Corrección de Errores	73
Figura 50 Flujograma Corrección de Errores.....	74
Figura 51 Interfaz de Datos.....	74
Figura 52 Pantalla Global de Errores	75
Figura 53 Menú SCADA.....	118
Figura 54 Diálogo de Login	119
Figura 55 Barra de pantallas	119
Figura 56 Pantalla automático.....	120
Figura 57 Pantalla Manual	122
Figura 58 Pantalla Acusar Errores	124
Figura 59 Pantalla Configuración	125
Figura 60 Pantalla Monitorización	126
Figura 61 Insertar equipos en programa	127
Figura 62 Pantalla equipos accesibles	128
Figura 63 Equipos del proyecto.....	128
Figura 64 Ejes tecnológicos	129
Figura 65 Configuración Eje tecnológico	129
Figura 66 Bloques del programa	130
Figura 67 Añadir bloques	130
Figura 68 Tabla de variables.....	131
Figura 69 Herramientas programa	131
Figura 70 Bloques insertables.....	131
Figura 71 Tecnología Motion Control	132
Figura 72 Funciones barra herramientas.....	132
Figura 73 Insertar simulación	133
Figura 74 Configuración Runtime.....	133
Figura 75 Pantallas proyecto.....	134
Figura 76 Elementos SCADA.....	135
Figura 77 Interfaz recetas	135
Figura 78 Visualizador de recetas.....	136
Figura 79 Interfaz de usuarios.....	136
Figura 80 Definición de alarmas.....	137

ÍNDICE DE TABLAS

Tabla 1 Desglose del coste por material	107
Tabla 2 Desglose del coste por Software.....	108
Tabla 3 Tabla salario Ingeniero Técnico Industrial	109
Tabla 4 Desglose del coste del diseño del proceso	109
Tabla 5 Tabla resumen del coste total.....	110

MEMORIA

1. Objeto del proyecto

En este punto se van a definir las metas del proyecto y se subdividirán en puntos independientes con la intención de facilitar la gestión de estos subapartados.

La finalidad del presente proyecto es el diseño de una función de control de posición y de una estructura de almacenamiento de datos para automatizar un almacén logístico.

Por tanto, el **objetivo principal** del proyecto será la programación de dicha funcionalidad atendiendo a los criterios de seguridad y fiabilidad necesarios para obtener niveles satisfactorios de operabilidad.

Se ha subdividido el proyecto en distintas tareas más sencillas las cuales se irán realizando individualmente. Al seguir esta metodología será más sencillo aislar los errores según se vayan produciendo ya que se reduce el área en la que se pueden haber dado.

Además, se asegurará el correcto funcionamiento de un objetivo para pasar al siguiente, en caso de no tomar ninguna medida es posible que se encuentren errores difíciles de rastrear y corregir.

Así, los **objetivos parciales** del presente proyecto son:

- Identificación de todos los elementos

El primer paso será diferenciar todo aquello que vaya a ser utilizado en el proyecto, realizar las conexiones eléctricas y comprobar que todos los dispositivos se encuentran en buen estado.

- Parametrización de los dispositivos de control

Seguidamente se parametrizarán los equipos a utilizar, configurando sus parámetros y haciendo su primera puesta en marcha. Seguidamente, se asignarán las direcciones de las comunicaciones.

- Selección y establecimiento de la comunicación entre equipos

Una vez se disponga de los equipos, se estudiarán los protocolos de comunicación disponibles y se elegirá el que se considere más adecuado. Una

vez se tome la decisión se creará la red de comunicaciones mediante la cual se va a realizar el control del proceso y se comprobará el funcionamiento del mismo. Esto se llevará a cabo enviando al accionamiento una orden sencilla desde el equipo de control.

- Diseño de la función

Se analizará el proceso en cuestión para determinar los requerimientos que debe tener el código. Seguidamente, se planificará la estrategia a seguir a la hora de escribirlo.

- Diseño del programa:

Se implementará un código en texto estructurado el cual incluirá funciones que permitan gestionar la información del almacén para determinar las acciones del sistema, dirigir a los accionamientos hasta el posicionado correcto u obtener la información que se debe monitorizar

- Implementación a la red de un sistema HMI en un PC:

Seguidamente se diseñará una de las comunicaciones humano-máquina. El HMI cumplirá la función de MES en el proceso, ya que en esta pantalla se introducirán los planes de producción que se traducen en el control del proceso.

- Introducción de un SCADA para supervisar el proceso:

El SCADA actúa como el tercer nivel de la pirámide de la automatización y permite controlar directamente el proceso, dándole orden de marcha, paro o pudiendo acceder a un control manual en el que se dan las órdenes de forma manual, siendo este modo útil para corregir algún error en la máquina o moverla para realizar mantenimiento

- Seguridad:

Se incluirán niveles de seguridad para el acceso del personal y para la introducción de datos. Se van a diseñar de medidas de seguridad que protejan la máquina de un uso inadecuado mediante límites de posición por software o revisión automática de los parámetros introducidos.

Se implementarán de medidas físicas que aseguren el correcto funcionamiento del sistema como inclusión de entradas en lógica inversa, conexión de una tensión de seguridad al variador o parametrización del sistema de frenado auxiliar del motor. Además, se dispondrá de cables apantallados para evitar EMIs.

2. Alcance

Con el presente trabajo de fin de grado se pretende acometer el diseño de una función genérica de control de posición de un eje motor. Se comenzará acotando las necesidades del proyecto, seguidamente se diseñará la función a implementar y se comprobará su funcionamiento utilizando el equipo correspondiente.

Se buscará que la función sea flexible y genérica, haciéndola por tanto adecuada para su reutilización. Esto es porque al referir el programa a variables que pueden cambiar sus valores es posible modificar el sistema sin tener que alterar la función. También se diseñará una estructura de almacenamiento de datos que permita dirigir los ejes hasta las posiciones libres o recoger los productos de sus posiciones de forma efectiva.

Por otra parte, se esbozarán los niveles de seguridad y de acceso de usuario ya que esto ha cobrado gran importancia al contar con sistemas interconectados.

3. Introducción

En este punto se van a explicar los conceptos más importantes para el correcto entendimiento del presente proyecto, se explicará en qué consiste la implementación de la automatización industrial al almacenamiento y las mejoras y cambios que aporta. A continuación, se hablará acerca del concepto de automatización industrial y de sus sectores, diferenciando los campos de cada uno de ellos y centrándose en los niveles que se van a aplicar en el proyecto. Finalmente, se expondrá el significado de la industria conectada 4.0, explicando sus objetivos y que tiene que ver con el tema presentado.

3.1 Sistemas de almacenamiento inteligente

Todos los procesos requieren de material para ser realizados y normalmente, estos materiales son almacenados en un lugar cercano a la fábrica. Resulta necesario entonces catalogar y guardar de forma ordenada y segura tanto la materia prima como el producto terminado para asegurar una producción constante que sea capaz de cubrir la demanda.



Figura 1 Almacén inteligente

El almacenado de materiales ha necesitado siempre de grandes extensiones de terreno y de unas instalaciones adecuadas. La utilización de la automatización industrial ha propiciado la aparición de instalaciones mucho más eficientes. Según algunas referencias, (Cisneros, Datadec, 2021) en un almacén tradicional

se pierde entre el 65% y el 80% del tiempo en recorrer las distancias. Además, solo es posible recorrer entre 35 y 45 líneas por hora. En el caso de los almacenes automatizados se trabaja a un ritmo de unas 120 líneas por hora mientras que se mantienen abiertos unos 75 pedidos de forma simultánea e incluso puede haber varios automatismos trabajando de forma paralela. Además, el operario no pierde tiempo en desplazamientos y el robot puede realizar distintas tareas en el mismo recorrido. Finalmente, otra de las ventajas más llamativas del sistema es el ahorro de espacio ya que frente a los 140m² que puede ocupar un almacén al uso se puede lograr la misma capacidad de almacenamiento en 8m² (su altura ronda los 16m) Existen sistemas capaces de procesar 400 líneas por hora e incluso ya se han implementado sistemas conocidos como almacenes ciegos en los que no hay interacción humana y por tanto están completamente oscurecidos. Como añadido, algunas de las tareas realizadas por humanos acaban resultando en lesiones para el operario por lo que se puede decir que la calidad de vida del trabajador se ve incrementada.

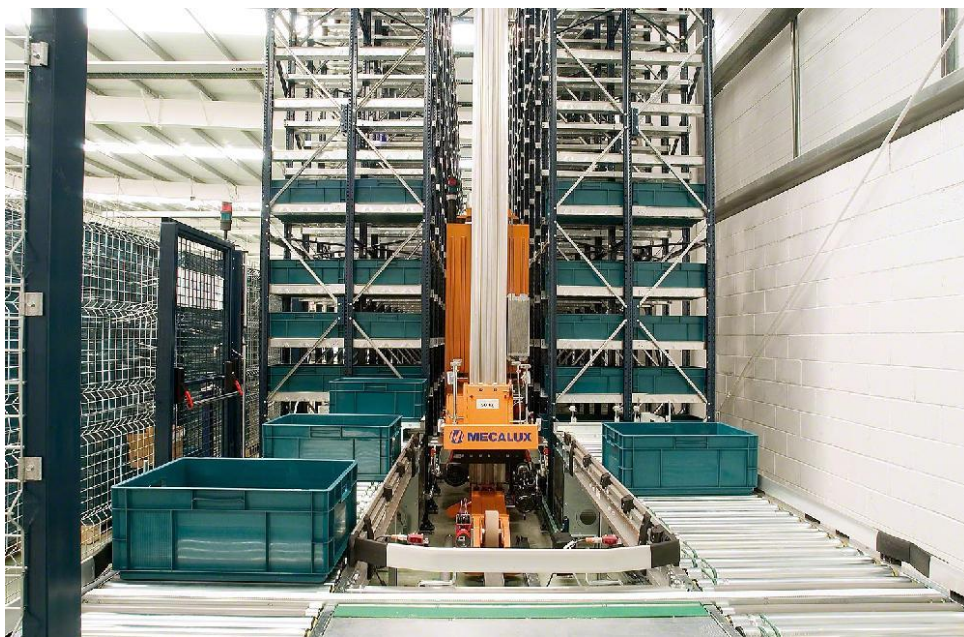


Figura 2 Transelevador industrial. Fuente Web Principal MECALUX

3.2 Automatización industrial

La automatización industrial es la rama de la ingeniería que se encarga de controlar y monitorizar procesos industriales mediante la utilización de elementos informáticos, electromecánicos y de comunicaciones. (Panadero, 2020)

Es una disciplina que, de forma general, suele ser dividida en 5 módulos tal y como representa la pirámide de la automatización.

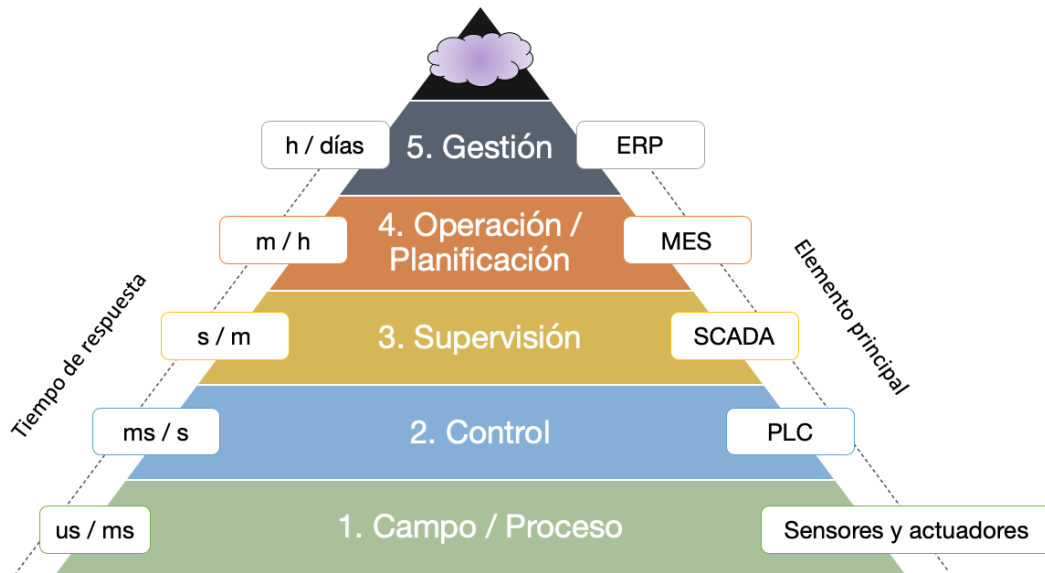


Figura 3 Pirámide de la automatización

- Nivel de campo:

El nivel de campo está compuesto por todos los actuadores físicos de monitorización y acción. Esto incluye todos los sensores digitales y analógicos que se emplean para supervisar el estado del proceso, salidas de visualización para establecer comunicación con los trabajadores, como pueden ser pilotos o sirenas o accionamientos que actúan directamente sobre la instalación para llevar a cabo la tarea del automatismo, por ejemplo, motores o robots y elementos que modifican el valor de algún proceso, como electroválvulas.

- Nivel de control

En este apartado se agrupan todos los dispositivos que controlan y ejecutan el programa. Incluye a los PLC y PCs industriales, aparatos diseñados específicamente para actuar en entornos agresivos. Los PLCs se emplean para realizar tareas repetitivas que requieren de un alto nivel de supervisión y respuesta ya que se diseñan específicamente para esta tarea. En cuanto a los PCs industriales se trata de computadores similares a los comerciales pero adaptados al medio y configurados con elementos suficientemente potentes como para ejecutar los software que se necesiten.

- Nivel de supervisión:

Este bloque el cual también es conocido como SCADA (Supervisory Control And Data Acquisition) es el encargado de establecer la comunicación humano-máquina permitiendo al usuario interpretar los datos de monitorización y ejercer control sobre el proceso. Su elemento principal suelen ser pantallas de PC, desde donde se puede llevar una supervisión completa de los procesos, incluyendo funciones complejas o códigos programados para tratar datos. A pie de fábrica se utilizan HMIs desde los cuales se controlan máquinas concretas con una supervisión mucho más básica.

- Operación:

También conocido como MES (Manufacturing execution systems) actúa como puente entre los niveles de planificación y diseño con la operación industrial. Se presenta como un software orientado a la planificación y ejecución del proceso teniendo como requisitos las especificaciones de niveles superiores y como objetivo traducir estas órdenes a la ejecución de tareas óptima para los recursos disponibles. En este caso, el nivel MES será interpretado por el SCADA, ya que en este se podrá cambiar la planificación de fabricación y monitorizar y extraer información para su tratado (Stockage Data).

- Gestión:

El ERP (Enterprise Resource Planning) es el sistema de gestión y procesado de datos en el cual converge toda la información. Su función es la planificación y toma de decisiones que afecta a todos los apartados inferiores. Con esto se busca seguir el estado de todos los procesos presentes en la empresa con el fin de optimizar los recursos disponibles. Usualmente manejan la producción, distribución, inventario, logística y contabilidad entre alguna otra funcionalidad.

3.3 INDUSTRIA CONECTADA 4.0 (I C 4.0)

La industria 4.0 es el concepto mediante el cual hace referencia a la cuarta revolución industrial, cuyos cambios están relacionados con la integración de los sistemas físicos de producción y las nuevas tecnologías digitales. Con esta tendencia se busca mejorar el comportamiento de las instalaciones industriales

mediante el empleo de softwares que mejoren la gestión y utilización de la información existente en un medio industrial. Destaca el empleo del BIG DATA, de la realidad aumentada o de la integración de protocolos de comunicación. (Nanclares, 2021)

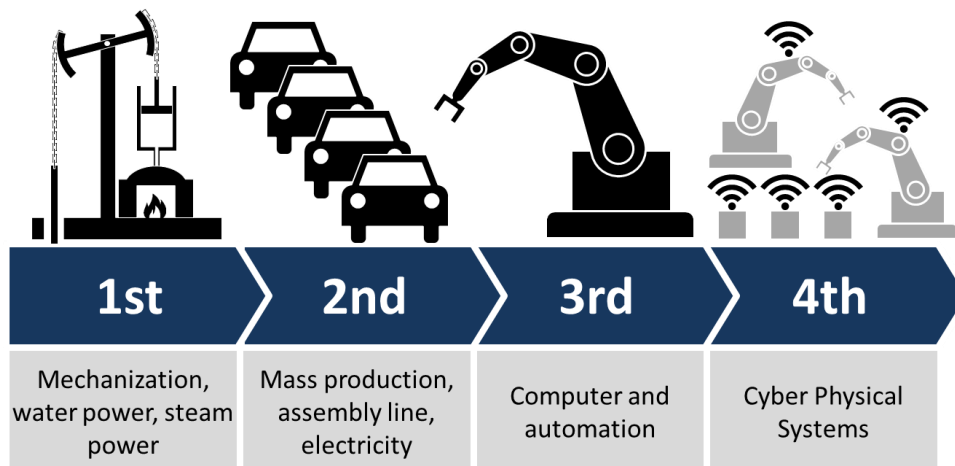


Figura 4 Etapas de las revoluciones industriales. Fuente: Página IC 4.0.gob

Las características principales de esta tendencia son:

- Automatización y adaptación del sistema

Mediante la automatización de los procesos industriales se busca una mejora del rendimiento y calidad de los productos. Los sistemas que actúan en estos entornos están específicamente diseñados para satisfacer las necesidades de la industria. Para esto se emplean equipos PLC y programas informáticos utilizados en estos dispositivos o en computadores al uso.

- Intercambio de información mediante sistemas de comunicación

Con esto se busca mejorar la realimentación de los distintos estados de la automatización industrial, permitiendo así que cualquier variación en el sistema tenga la repercusión adecuada. Mediante la inclusión de buses de comunicación y conexiones inalámbricas se consigue un flujo de información lo suficientemente rápido como para actuar el tiempo real.

- Procesado masivo de datos (Big Data)

La cantidad de información que cualquier sistema produce hoy en día puede ser abrumadora, es por esto que es necesario implementar métodos que permitan gestionar y utilizar esta información ya que esta puede ser utilizada para hacer previsiones a futuro y mejora la eficiencia de la empresa.

- Interacción hombre máquina

Muchos de los procesos industriales actuales no requieren de la actuación directa de las personas ya que los dispositivos de control utilizados son más eficientes a la hora de realizar tareas repetitivas. Sin embargo, la actuación del ser humano sigue siendo necesaria. Por esto, se busca implementar interfaces que permitan que esta comunicación se produzca de forma eficaz. Además, la supervisión de los procesos debe de ser llevada a cabo por humanos, por lo que es necesario que el trabajador pueda visualizar de forma gráfica los valores de las magnitudes del proceso que convenga monitorizar.

- Digitalización y optimización de procesos industriales:

Como punto final, también se ha considerado la importancia de transformar tantos sistemas y procesos a un entorno digital como sea posible ya que se puede incrementar la eficacia y el rendimiento de estos procesos utilizando técnicas digitales, como programar un modelo software para prever las evoluciones de la actividad o establecer criterios de control informáticos, esto es increíblemente útil a la hora de evitar introducir sensores en zonas muy engorrosas, como el estator de un motor eléctrico o para gestionar grandes volúmenes de datos utilizando redes neuronales para monitorizar tan solo una pequeña fracción del mismo

Sus objetivos son por tanto aumentar el nivel de automatización de la industria para así incrementar la eficiencia y la productividad, optimizando los consumos y permitiendo llegar a niveles de precisión solo posibles mediante la simbiosis de distintas tecnologías.

4. Justificación

En este apartado se explicarán los motivos de realizar este proyecto en concreto. Se abordará desde un punto de vista académico, relacionando el trabajo con lo aprendido en la carrera y desde un punto de vista técnico, se explicará la relevancia de automatizar el proceso del almacenado.

4.1 Académica

La finalidad de realizar este proyecto desde un punto de vista educativo es la de aplicar los conocimientos obtenidos durante la realización del grado para demostrar que se han adquirido las competencias y aptitudes propias de un graduado en ingeniería eléctrica. Por otra parte, se desea profundizar en el campo de la automatización industrial con el objetivo de mejorar las habilidades propias del ingeniero eléctrico en el campo de la ingeniería de control. Como añadido, se busca obtener nuevas competencias como el montaje y puesta en marcha del sistema desde cero, sobre el funcionamiento de las comunicaciones industriales y aprender acerca de las nuevas tendencias en la industria. Finalmente se ha deseado contrastar el comportamiento de la comunicación de sistemas de la misma marca comercial.

4.2 Técnica

Desde el punto de vista técnico, la implementación de un sistema de control de posición automatizado es completamente viable. Esto se debe a que este tipo de procesos se beneficia ampliamente de las capacidades que brindan los autómatas programables.

Mediante la implementación de una función genérica, se puede mejorar la flexibilidad del proceso, incrementando la calidad de su ejecución y reduciendo el tiempo y esfuerzo de su instalación. Por otra parte, al disponer de sistemas capaces de monitorizar el proceso en tiempos muy cortos, se dispondrá de una mejor respuesta y una mayor seguridad.

5. Estudio de viabilidad

En este punto se presentarán las normativas de obligado cumplimiento para la realización del proyecto y se enumerarán, junto a una rápida explicación, los requisitos que debe cumplir el programa para poder garantizar su funcionamiento de forma segura.

5.1 Normativa

Para la realización del proyecto se ha tenido en cuenta esta normativa.

- Real decreto 842/2002 Reglamento electrotécnico de baja tensión 2002.

El REBT es el conjunto de normativas que regulan el correcto uso de la baja tensión eléctrica. Entre sus apartados, se explica cómo dimensionar los cables, el código de colores que debe usarse para diferenciar el uso de cada conexión y las consideraciones que deben ser tenidos en cuenta en las instalaciones de motores eléctricos.

- IEC 61131

Este documento es un conjunto de normas elaborado por la Comisión Electrotécnica Internacional cuyo objetivo es la estandarización de los autómatas programables, en este documento se especifican los lenguajes de programación normalizados, las directrices de seguridad y control o los ensayos que deben realizarse entre otros.

5.2 Requisitos

El programa cuenta con determinados puntos obligatorios, la inclusión de estos elementos tiene como función asegurar que el proyecto es seguro y fiable.

- Funciones que garanticen la seguridad del trabajador

Se diseñarán interrupciones que eviten en todo momento que un comportamiento inadecuado pueda resultar en un accidente. Esto se realizará mediante el acotado de un perímetro de seguridad cuyo acceso no estará permitido. En caso de que se violase el espacio restringido, la máquina se detendría.

- Funciones que garanticen la seguridad del automatismo

Se programarán distintos límites que garantizar la integridad de los accionamientos tales como limitaciones de velocidad y par para que no sean sobrepasados los valores recomendados de los motores o líneas de código dedicadas a asegurarse que los valores introducidos en los parámetros editables no son erróneos.

- Entradas de seguridad configuradas en lógica inversa

Las entradas que monitoricen el estado de variables comprometidas se cablearan en lógica inversa para asegurar que en caso de fallo este no pasa desapercibido. Esto haría referencia a pulsadores de emergencia, relés térmicos, finales de carrera hardware o sensores de proximidad.

- Disposición de un modo JOG

Se insertará una función que genere un tren de pulsos que mueva los accionamientos de forma lenta. Este modo tiene como fin habilitar un movimiento controlado a la hora de comprobar que el mecanismo funciona, ya sea durante la instalación o al realizar algún tipo de mantenimiento. (Fort, 2020)

- Diseño del SCADA

En primer lugar, el SCADA debe de ser capaz de ofrecer una monitorización clara del proceso y ofrecer al usuario un control intuitivo del mismo.

La interfaz del SCADA será simple y garantizará que pueda ser empleada sin necesidad de formación específica. Los estados de los procesos se presentarán de forma clara y los valores de las variables podrán ser consultadas con un simple vistazo. Por otra parte, se permitirá incluir nuevos usuarios desde la interfaz y se habilitará el cambio de idiomas entre los idiomas más comunes de la Unión Europea.

- Maniobra de retorno al origen

Contando con la posibilidad de que se produzca un fallo y el automatismo quede bloqueado en una posición distinta del origen, se dispondrá de una funcionalidad que ordene a la máquina retornar al punto x 0 y 0.

- Uso de un lenguaje estandarizado

El proyecto debe de ser lo más genérico posible para que pueda ser implementado en distintos casos sin necesidad de hacer grandes cambios. Esto se conseguirá utilizando lo que se conoce como programación orientada a objetos. La programación orientada a objetos es un método en el cual se diseña el código pensando en cada proceso como un módulo independiente. Por ejemplo, se diseñará una función que controle la carga del almacén y otra que controle su descarga. Sin embargo, estos bloques podrán ser utilizados para cargar otros almacenes ya que no están restringidos a los datos de uno en concreto.

- Enclavamientos

Se programará las activaciones de los distintos modos usando enclavamientos, es decir, mediante funciones SR (Set Reset) se impedirá activar de forma simultánea procesos que no pueden realizarse en el mismo momento. Además, se introducirán medidas de seguridad para evitar cierres intempestivos del programa.

- Control de posición preciso

Se va a llevar a cabo una función que controle la posición exacta a la que se mueven unos ejes por lo que se necesitará un equipo capaz de tener esta precisión. Por tanto, los motores que se utilicen deben de disponer de una precisión muy elevada.

6. Criterios de selección y justificación de la solución adoptada

A continuación, se estudiarán las alternativas posibles a la ejecución del proyecto ya que un análisis profundo podría mostrar alguna vía alternativa más interesante. Seguidamente se presentará el estudio de distintas posibilidades para cada uno de los elementos de la instalación.

6.1 Control del proceso

Es necesario seleccionar un dispositivo que actúe como cerebro del proceso. En este, mediante lógica digital o cableada, se introducirá el diseño del control del proceso. A continuación, se presentarán algunas de las alternativas posibles y se justificará la elección más adecuada al proceso a controlar.

PLC

Un PLC o autómatas programables es un dispositivo electrónico diseñado específicamente para controlar procesos industriales secuenciales en tiempo real. Su manejo puede ser llevado a cabo por personal electrónico o eléctrico sin necesidad de amplios conocimientos de informática. Su interfaz permite la utilización de funciones lógicas, temporizadores, contado, realización de cálculos matemáticos, etc. La función de un PLC es reducir el trabajo del usuario mediante la automatización del proceso ya que es capaz de realizar ciertas acciones previamente programadas ante determinados valores de entrada.



Figura 5 PLC SIEMENS. Fuente: Web Principal Siemens

Este aparato es el centro de cualquier automatismo ya que es el encargado de la supervisión y control del proceso. En el mercado existen gran cantidad de tipos de autómatas programables y es importante seleccionar uno adecuado a la tarea. En el caso actual, un autómata modular, de reducido tamaño, precio asequible y capacidad media debería ser suficiente. Además, es posible mejorar sus prestaciones mediante la inclusión de nuevos módulos.

Entre sus principales ventajas se encuentra:

- Altas prestaciones de control

El diseño de un autómata programable está enfocado de forma que mediante una alta velocidad de procesamiento el dispositivo es capaz de responder de forma rápida y de gestionar varios procesos de forma simultánea. Aun así, la capacidad de respuesta del sistema dependerá de la habilidad del programador a la hora de utilizar los recursos del aparato y de la potencia del modelo que se esté utilizando.

- Diseño específico para utilización industrial

Los PLCs están pensados específicamente para ser utilizados en procesos industriales, es por eso por lo que cuentan con buses de comunicaciones, entradas y salidas tanto analógicas como digitales o fuentes de tensión de 24 VDC. Además, los modelos que son modulares admiten el acoplamiento de mejoras que pueden ir desde pantallas hasta un mayor número de puertos.

- Fiabilidad y facilidad

Sus sistemas son de alta fiabilidad y además resulta sencillo realizar el mantenimiento de los equipos. Incluso es posible reemplazar los modelos o lenguajes entre sí con pocos cambios. Además, se puede descargar el mismo código en distintos autómatas, lo que ahorra mucho tiempo.

Actualmente los autómatas programables son empleados en multitud de tareas distintas. El aumento en su demanda ha crecido sustancialmente estos últimos años debido a los resultados que ha supuesto.

Al controlar el proceso mediante un código escrito tenemos como ventaja también que cualquier cambio solo debe ser realizado sobre este. Lo cual reduce los tiempos en el mantenimiento y las reparaciones.

LÓGICA CABLEADA

Un sistema basado en el empleo de contactores consiste en controlar los accionamientos del proceso mediante la apertura y cierre de contactos eléctricos. Estos contactos se abren o cierran, dependiendo su estado en reposo (normalmente abiertos y normalmente cerrados), mediante la alimentación de la bobina del contactor. Este planteamiento hace posible la disposición de un control simple basado en la acción de estos elementos dirigidos por la actuación de sensores, temporizadores u otro tipo de entrada. De todas formas, este método presenta ciertas carencias:

- La complejidad de implementar determinadas funciones

La realización de algunos procesos se vuelve compleja al utilizar contactores complicando en gran medida sus esquemas y exigiendo la utilización de elementos adicionales que aumentaría el coste de la instalación.

- La dificultad y costes de cualquier intervención sobre el sistema

El empleo de métodos cableados supone que cualquier modificación o implementación a realizar se traduce en una desinstalación e instalación de todo el sistema, ya que se tendrían que desconectar cables ya sea para acceder al elemento a reparar o bien reconfigurar el sistema con el fin de realizar la nueva funcionalidad. Esto produce que cualquier acción sobre el control supondría tener que detener el proceso durante un periodo de tiempo, el cual no será precisamente corto.

- La cantidad de cable necesario

Todos los contactos de la instalación van por conexiones hilo a hilo, es decir, mandando 2 cables por sensor y por actuador (entrada y salida). Por lo que el volumen de cable será muy elevado y el espacio ocupado también lo será.

El control mediante contactores se realiza mediante la llamada lógica cableada, cuya utilización es conocida por los problemas que acaba generado. De hecho, es conocida comúnmente como la “pesadilla lógica”.

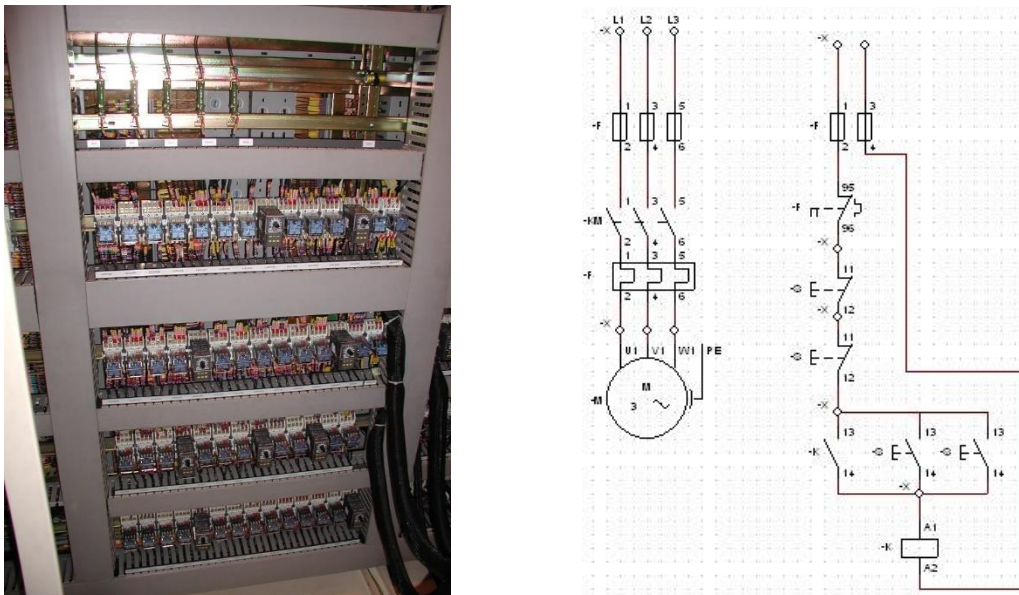


Figura 6 Lógica cableada (izquierda) / Esquemas de potencia y control (derecha)

SISTEMA EMBEBIDO

El microprocesador es un circuito programable integrado, el cual puede ejecutar las órdenes previamente programadas en un código almacenado en su memoria interna. Su sistema está compuesto por bloques funcionales, que son los responsables de realizar una tarea específica. En estos podemos encontrar la unidad central de procesamiento, un cierto número de periféricos y la memoria del dispositivo. Entre sus ventajas, destaca:

- Características de forma

El coste de un sistema embebido es reducido en respecto a sus alternativas ya que sus componentes se fabrican en masa y su utilización está muy extendida. Las pequeñas dimensiones alcanzadas por el desarrollo de los sistemas informáticos también permiten instalar los circuitos de forma cómoda y sin requerir de espacios muy grandes. Finalmente, es posible adquirir muchos de sus componentes de forma separada si fuese necesario.

- Versatilidad

Al utilizar lenguajes de programación muy extendidos, es posible reutilizar los proyectos y también disponer de una comunidad de programadores mundial.

Por otra parte, tiene varias limitaciones:

- Diseño genérico

Los sistemas embebidos no están diseñados para ser utilizados en un ambiente industrial, es por esto que sería necesario preparar el circuito con las entradas, salidas y puertos de comunicaciones deseados si se desea utilizarlo. Además, al no estar preparado para soportar las altas temperaturas, la humedad o cualquier otra característica especial de estos entornos seguramente no sería del todo fiable.

- Ejecución del programa

Estos equipos funcionan mediante interrupciones, es decir, su código se activa cuando una determinada entrada da señal. Esto es un hándicap importante para su aplicación ya que sería más interesante que el controlador realizase un ciclo continuo en el que monitorizase el proceso y ejecutase el código en intervalos de tiempo reducidos.



Figura 7 Ejemplo de sistema embebido

Finalmente, se ha decidido utilizar un autómata programable debido a las superiores características que proporciona. Esto es debido a que, al estar

diseñado específicamente para su uso industrial, contamos con servicios que serán de gran ayuda.

6.2 Lenguaje de programación

Una vez seleccionado el PLC, se deberá elegir el lenguaje de programación entre todos los lenguajes de programación normalizados.

Se dispone de distintos lenguajes de programación normalizados, los cuales están compuestos de distintas funciones que, al traducirse al lenguaje máquina, actúan sobre el controlador. Diferenciamos entre lenguajes literales y gráficos.

LENGUAJES LITERALES

La norma tipifica dos lenguajes literales como los adecuados.

- Texto estructurado

```
1
2 □ (* infoPLC.net TEMPORIZADOR *)
3
4 □ (*Instruccion TIMX ( Bit de activacion, Nombre del temporizador, Valor de temporizacion) *)
5 TIMX(Marcha,Temporizador1,Valor_Tim);
6
7 □ (* infoPLC.net ACTIVACION DE TEMPORIZADOR y PUESTA A ON DE MOTOR *)
8 □ IF Temporizador1.CF = TRUE THEN
9   Motor:= TRUE;
10 □ END_IF;
11
12
```

Figura 8 Ejemplo texto estructurado

Lenguaje de programación basado en instrucciones concretas escritas dentro de un entorno de programación. Este lenguaje está condicionado por los conocimientos de la persona encargada de la programación por lo que es un lenguaje que permite alcanzar un nivel bastante alto siempre y cuando el programador disponga de amplios conocimientos. Es un lenguaje similar a los códigos escritos de otros lenguajes como C y permite una abstracción más alta que la de los lenguajes gráficos. Como contrapartida, no es usual que un técnico tenga conocimientos de programación tan avanzados por lo que cualquier

cambio o mantenimiento deberá ser llevado a cabo por un ingeniero, lo que incrementará el coste.

- Listado de instrucciones

```

000 LD    %I0.1  Bp. inicio ciclo
      AND   %I0.0  Dp. presencia vehículo
      AND   %M3    Bit autorización reloj calendario
      AND   %I0.5  Fc. alto rodillo
      AND   %I0.4  Fc. detrás pórtico
005 S    %M0    Memo inicio ciclo
      LD    %M2
      AND   %I0.5
      OR    %I0.2  Bp. parada ciclo
      R     %M0
010 LD    %M0
      ST   %Q0.0  Piloto ciclo

```

Figura 9 Ejemplo listado de instrucciones

Este código está formado por una serie de instrucciones en formato lista que va siendo ejecutado de forma cíclica. Similar al texto estructurado.

LENGUAJES GRÁFICOS

La legislación contempla la utilización de ciertos lenguajes en concreto

- Ladder

Este lenguaje es uno de los primeros en ser implementado, por lo que es bastante fácil de encontrar en la industria. La programación es de izquierda a derecha y de arriba abajo. Los valores de las entradas se sitúan a la izquierda de la interfaz, seguido de los procesos a seguir para activar las salidas, las cuales se encuentran a la derecha. Su programación es sencilla y fácil de interpretar en tiempo real por lo que es un lenguaje adecuado para profesionales con poca experiencia o para sistemas que vayan a monitorizar o reparar técnicos cuyo conocimiento de programación no se suponga muy avanzado.

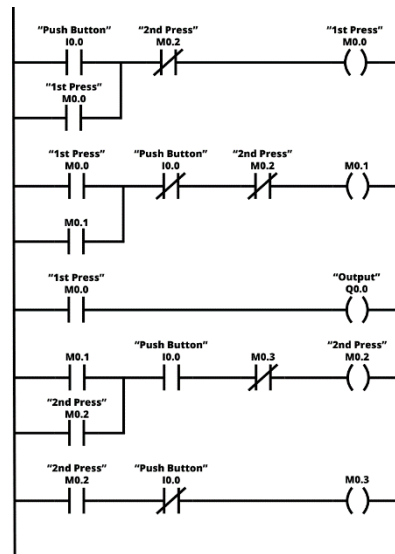


Figura 10 Ejemplo diagrama Ladder

- Bloques de función

Este sistema se basa en la utilización de cajas lógicas para la representación del bloque de la función. Las salidas son representadas por una variable concreta asignada a la salida del bloque. Los outputs de los bloques no están conectados entre sí por lo que la evaluación de cada red es independiente.

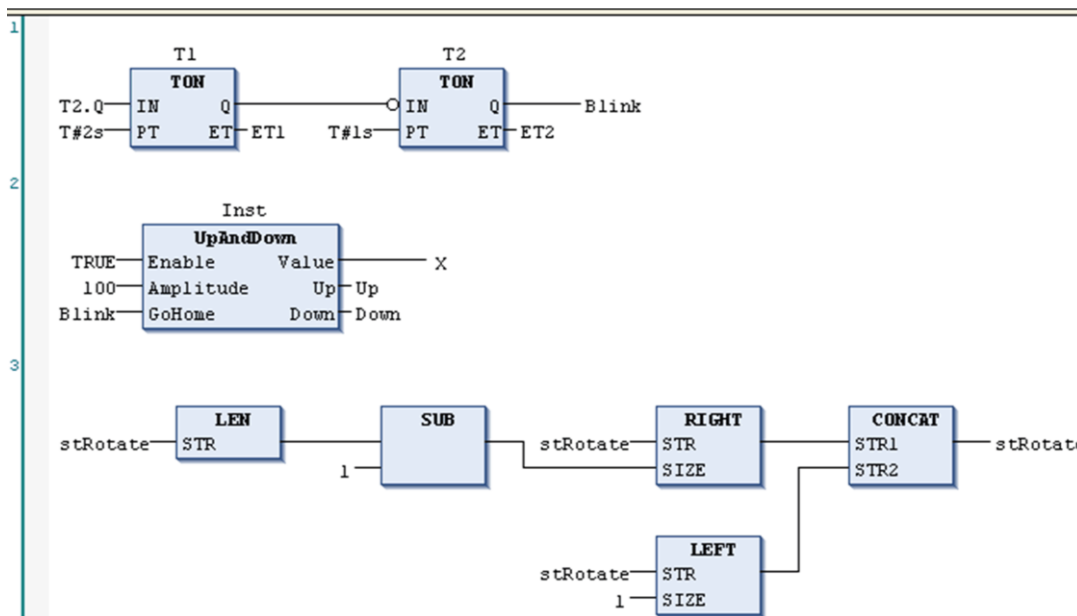


Figura 11 Ejemplo Bloques de función

Desde un punto de vista académico, resulta más interesante realizar en código en un lenguaje literal, ya que son los más avanzados y de los que mayor valor se puede obtener.

6.3 Elección del motor

Seguidamente será necesario seleccionar el accionamiento eléctrico que mejor se adecue al proceso. Desde un inicio se descartan los motores asíncronos ya que al buscarse un posicionado preciso, las características de este motor lo hacen poco adecuado para su uso. A continuación, se presentarán algunas de las alternativas posibles y se justificará su elección.

PMSM

El motor síncrono de imanes permanentes es uno de los motores eléctricos que mejores características presenta. Este accionamiento dispone de un campo magnético en el rotor generado por imanes, por lo que la parte móvil del motor actúa de forma autónoma. Como el resto de los motores eléctricos, el estator se encuentra bobinado y mediante el paso de corrientes trifásicas desfasadas se consigue un campo magnético rotatorio que gira a la velocidad de sincronismo de la máquina. El movimiento del rotor se producirá por la interacción entre ambos campos y al ser completamente independientes el uno del otro, la posición de este seguirá exactamente a la posición del campo del estator.



Figura 12 Motor síncrono de imanes permanentes Fuente: Web Principal Siemens

Entre sus ventajas se encuentra:

- Altas prestaciones

Como el resto de los motores síncronos, el motor de imanes permanentes presenta unas prestaciones muy elevadas. Esto es debido a que los campos de rotor y estator son independientes y a que la densidad de flujo producida por los imanes es muy alta. Además, al controlar únicamente las magnitudes del estator, el control de la máquina se vuelve más preciso y seguro.

- Ventajas resultado del cierre hermético

Al no necesitar acceder al interior de la máquina para ninguna actividad, el PMSM se encuentra completamente cerrado, reduciendo por tanto el mantenimiento requerido ya que nada penetrará en el interior del motor y el ruido producido por el mismo. Además, las pérdidas en el rotor son nulas y por tanto la refrigeración del motor será nula y su eficiencia mejorará.

Sin embargo, este dispositivo también cuenta con distintas desventajas.

- Precios elevados

Los materiales magnéticos de los PMSM están fabricados con tierras raras, materiales escasos y que requieren de procesos costosos para ser utilizados y no es lo único que encarece estas máquinas ya que requieren de sistemas de arranque como variadores de frecuencia o arrancadores suaves para poder funcionar. También necesitan sensores de posición para conocer la localización del rotor en todo momento, esto puede realizarse mediante un encoder o mediante métodos matemáticos implementando un modelo en el control.

- Necesidad de control

Como ya se ha mencionado en el punto anterior, estos motores requieren de un control dedicado para funcionar de forma correcta. Esto será necesario tanto para que el motor pueda arrancar como para corregir el campo del estator en caso de que el rotor se salga de la posición de sincronismo.

SERVOMOTOR DC



Figura 13 Servomotor DC

La otra opción posible sería utilizar un motor de corriente continua. Se trata de un accionamiento el cual lleva un microcontrolador integrado y que mediante la generación de una señal PWM consigue llevar al motor hasta la posición deseada. El ancho de pulso de esta señal determinará el ángulo que debe recorrer el motor. Cuenta con las siguientes ventajas:

- Precio

Los motores DC son más económicos debido a la simpleza de su diseño y además no requieren de un dispositivo de control más allá de un microcontrolador el cual es mucho más fácil de adquirir que un variador de frecuencia.

- Espacio ocupado

Este motor ocupará menor espacio ya que integra el control en su interior, en el caso del PMSM, necesitamos disponer de un armario para el variador, el cual además producirá interferencias electromagnéticas debido a la alta frecuencia de su etapa rectificadora. Además, se requiere de comunicación constante entre el PLC y el VSD

- Precisión

El posicionado que puede lograr el control es de alta calidad y por tanto resultará suficientemente fiable para la tarea.

Pero también sufre de ciertos puntos flacos:

- Par dependiente de inducido

En el momento que la señal PWM desaparezca el motor dejara de ejercer par, esta desventaja generaría muchos problemas en la aplicación, ya que pondría en riesgo la seguridad del accionamiento. El motor PMSM si que es capaz de ejercer par aún detenido.

- Elementos rozantes

El servomotor de DC dispone de elementos rozantes y por tanto no puede estar encapsulado. Esto producirá que su mantenimiento y ruido sea mayor.

Finalmente se ha tomado la decisión de emplear un PMSM debido a las superiores características técnicas del mismo, el mejor control de par, el freno magnético en parado y el reducido mantenimiento que supone. Al utilizar un PMSM, será necesario seleccionar un elemento de control ya que este tipo de motores requiere de un dispositivo como un variador de frecuencia para poder funcionar de forma efectiva

6.4 Control del motor

Será necesario disponer de un elemento que traduzca las órdenes dadas por el control a magnitudes eléctricas para que el comportamiento del motor sea el deseado en la instalación. A continuación, se expondrán los elementos capaces de seguir esta premisa y se justificará la elección.

VARIADOR DE FRECUENCIA

Una de las soluciones posibles sería utilizar un convertidor de frecuencia, el cual es un dispositivo de electrónica de potencia diseñado para alimentar y controlar un motor eléctrico. Su uso permite, mediante el empleo de controles como el vectorial o el directo de par, seguir consignas de posición, velocidad y par mediante la variación de la tensión de alimentación y de la frecuencia. En este caso, será utilizado para generar el tren de pulsos necesario para que el motor

alcance la posición demandada. Además, es capaz de eliminar las ineficiencias del sistema y permite alimentar al motor sin armónicos.



Figura 14 Variador de Frecuencia ABB Fuente: Web Principal ABB

Un variador de frecuencia está compuesto por 3 etapas:

- Etapa rectificadora: Mediante el empleo de diodos o tiristores se convierte la tensión alterna a tensión continua.
- Bus de continua: En esta etapa intermedia se reduce el rizado de la tensión con un filtro (condensador en paralelo).
- Etapa inversora: En la última etapa de un variador de frecuencia se produce la conversión de continua a alterna de nuevo. Mediante el empleo de interruptores de potencia cuyo disparo se ordena desde el control del variador se genera una onda de alimentación que produzca que el motor siga las referencias estipuladas por el usuario.

Entre las ventajas del variador de frecuencia se encuentran

- Eliminación de sistemas mecánico

Al realizar la regulación de las magnitudes físicas del motor de forma digital, permite eliminar cualquier sistema de transmisión mecánica utilizado, ya sean transmisiones por cinta, hidráulicas o por engranajes. Esto elimina las posibles limitaciones que estos mecanismos pueden producir, como por ejemplo en el par. Además, no será necesario disponer de espacio extra para

instalar estos sistemas y el rendimiento mejorará debido a la eliminación de pérdidas mecánicas.

- Control regulable

Los métodos de control internos de los que dispone un variador permiten configurar las consignas de par, velocidad y posición, consiguiendo así ajustar estas magnitudes a los valores requeridos por la función que estén desempeñando. Al disponer de estas facilidades, los pares e intensidades en el arranque pueden reducirse suavizando la alimentación en los primeros momentos de este, por lo que se reducirán los picos de demanda y no será necesario dimensionar el cableado para soportar las grandes corrientes que requeriría un motor en otro caso. Además, el arranque de ciertos motores como el de los motores síncronos no sería posible sin emplear controles como el proporcionado por un variador.

- Mejoras en el comportamiento del motor

Permite que la respuesta dinámica del accionamiento sea lo más elevada posible. Supone también una mejora de la eficiencia y permite utilizar motores avanzados como el síncrono de imanes permanentes o el de reluctancia. También mejora el comportamiento de los motores asíncronos, consiguiendo que se acerquen a la velocidad de referencia a pesar del hándicap que presenta el deslizamiento. Por último, permite configurar parámetros como las rampas de aceleración, el tipo de frenado o evitar bandas de frecuencia que puedan producir problemas en la instalación.

En cuanto a sus desventajas se puede mencionar

- Precio

El coste de la electrónica de potencia es elevado y es necesario asegurar que el desembolso va a ser amortizable. Las prestaciones que ofrecen estos equipos son elevadas y por tanto su fabricación es muy especializada.

- Delicadeza

Los variadores de frecuencia son equipos delicados que deben ser empleados de forma correcta. Una instalación o protección incorrecta pueden dejar el equipo fuera de funcionamiento con relativa facilidad. Por tanto, son equipos que requieren de mucha planificación tanto a la hora de usarse como a la de instalarlos.

ARRANCADO DIRECTO A RED

Otra posibilidad sería utilizar un algún método alternativo para arrancar el motor, como por ejemplo que este dispusiese de una jaula de ardilla en el rotor que le hiciese funcionar como un motor asíncrono hasta alcanzar la velocidad de sincronismo. Para esto convendría incluir un arrancador suave, que es un dispositivo que permite arrancar motores asíncronos sin producir el pico de intensidad que caracteriza la activación de estos motores. De esta forma, se prescindiría de un elemento tan caro como el variador de frecuencia y se controlaría la posición del carril mediante sensores.

Si se desea ir a una posición concreta se alimentará el motor hasta que el sensor de esa posición de la estantería se active, una vez esto pase, se frena el accionamiento. Este método presenta como principal beneficio su coste ya que los elementos utilizados son asequibles y el funcionamiento es sencillo. Sin embargo, la precisión de la maniobra sería dudosa y el motor actuaría a la velocidad de sincronismo sin posibilidad de regular ni configurar nada.

Se ha decidido implementar un variador de frecuencia debido a las elevadas prestaciones que ofrece. Sus hándicaps pueden ser solucionados mediante un trabajo de calidad y las posibilidades que ofrece son muy recomendables para la aplicación en cuestión.

6.5 Selección de los elementos de visualización

Se necesitará disponer de uno o varios soportes para ofrecer información visual al trabajador. Además, se prescindirá de botoneras físicas, dejando libres las entradas del PLC para elementos de supervisión del proceso por lo que también se emplearán estos elementos para controlar el mismo. Como ventaja adicional, estas interfaces ofrecen más posibilidades ya que entre otras permiten introducir valores numéricos mediante teclados flotantes.

HMI-SCADA

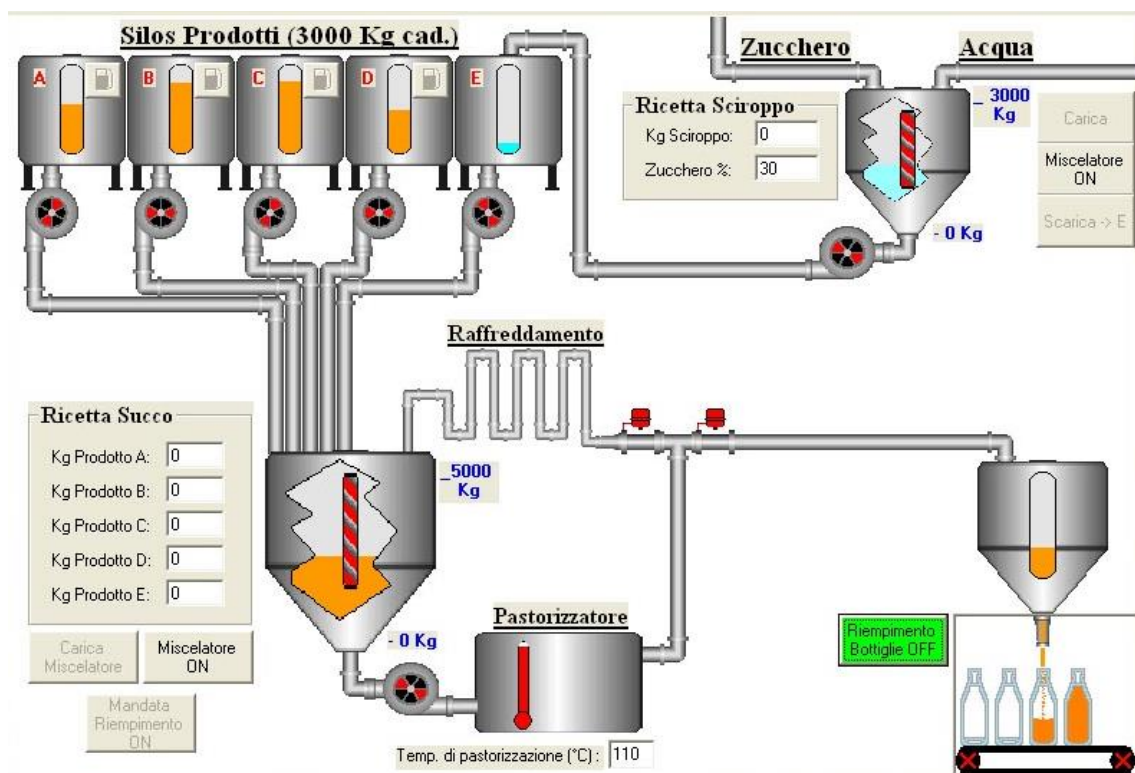


Figura 15 Ejemplo pantalla SCADA

Responde a las siglas de Supervisory Control and Data Acquisition. Consiste en una pantalla que permite al usuario obtener información del proceso y actuar sobre el mismo. Este tipo de sistema cuenta con software dedicado en el cual se ofrecen muchas opciones y facilidades para su diseño. Algunos cuentan con programas independientes como Vijeo designer para Scheneider o están integrados en una interfaz como TIA portal en Siemens.

El SCADA permite el control y/o comunicación con varios elementos. Los cuales pueden ir desde información de las magnitudes de los motores, medidas de los sensores o datos obtenidos desde los estados de gestión de la automatización (MES o ERP). También permite gestionar las alarmas configuradas o generar un sistema de acceso con usuarios.

PANEL SINÓPTICO

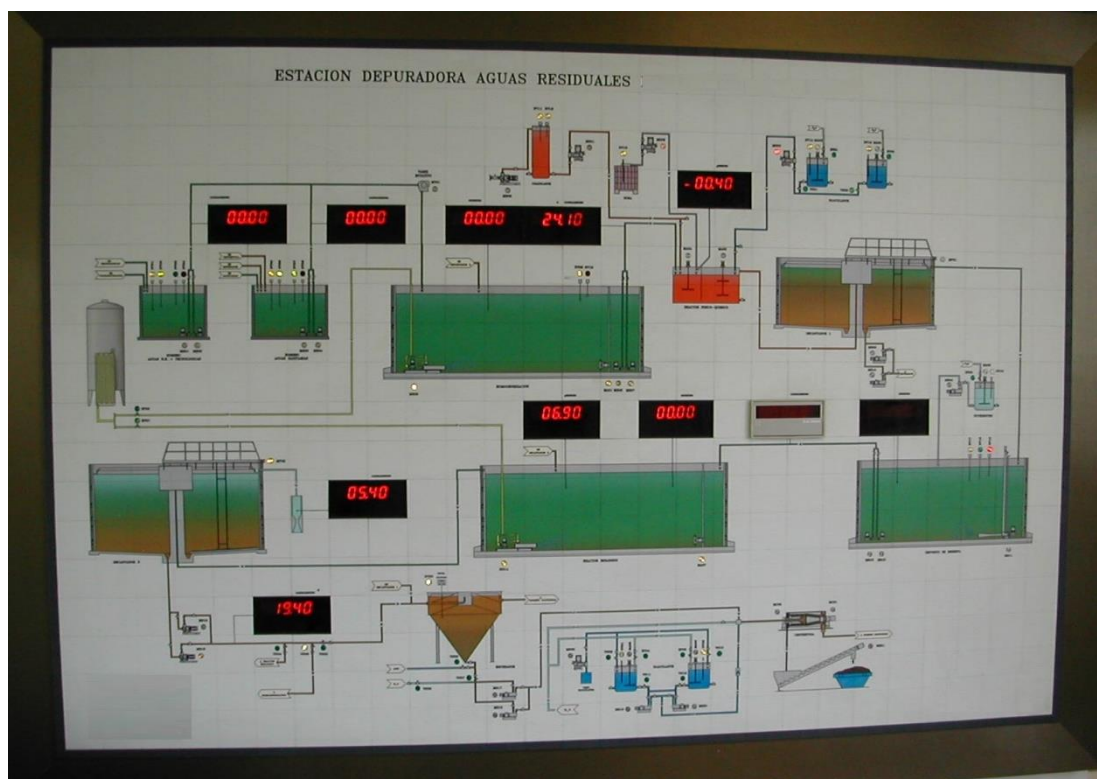


Figura 16 Ejemplo Panel sinóptico

Otra posibilidad a estudiar son los paneles sinópticos, que consisten en imágenes realizadas sobre el propio cuadro y que, mediante displays y pilotos, puede mostrar los valores que resulten interesantes o indicar ciertas condiciones de marcha

Como punto positivo, su visualización es clara y su implementación es sencilla ya que no requiere de conocimientos informáticos. Por otro lado, presenta desventajas como que cualquier modificación requiere volver a realizar el panel o que su espacio es limitado y no se podrá mostrar toda la información. Finalmente, son muy voluminosos y por tanto su transporte es difícil.

Finalmente se optará por utilizar un SCADA, este estará subdividido en dos partes, una para la supervisión del proceso, utilizando una pantalla HMI, y otra para el apartado de planificación, para la cual se empleará un PC industrial.

7. Justificación de la solución adoptada

Como resumen de los apartados anteriores, se ha decidido emplear un PMSM para el control del proceso, el cual deberá contar con un VSD para asegurar su correcto funcionamiento, mientras que el nivel de supervisión y planificación será llevado a cabo por un SCADA dividido entre un PC industrial y un HMI. El proceso será controlado de forma total por un PLC.

Se ha decidido utilizar un PLC ya que es un controlador diseñado específicamente para el entorno industrial que es el tipo de instalación en la que se va a encontrar el automatismo. Además, sus prestaciones elevadas permiten supervisar el programa de forma efectiva y segura. Además, los autómatas disponen de herramientas como entradas y salidas tanto digitales como analógicas, fuente de tensión de 24 VDC o switch de comunicaciones. Por otra parte, disponen de interfaces de programación propias donde se ofrecen funcionalidades preprogramadas que facilitan la escritura del código. En algunos casos se dispone de software para diseñar SCADAS de forma paralela como puede ser Codesys de ABB o TIA PORTAL de Siemens.

El modelo elegido es un S7-1215AC de la marca Siemens ya que cuenta con dos puertos de PROFINET, 16 entradas y salidas digitales y dos entradas analógicas. El PLC seleccionado se muestra en la imagen 17



Figura 17 PLC elegido Fuente: Web Principal Siemens

El motor utilizado será de síncrono de imanes permanentes ya que sus prestaciones son muy recomendables para el caso particular que se está tratando. Su principal ventaja es que permiten un posicionado muy preciso y disponen de par total a velocidad 0. Como se van a elevar los productos desde parado, es interesante contar con esta posibilidad y desde el punto de vista de la seguridad, los ejes no se desplomarán si se detiene el motor.

Estos accionamientos disponen de una alta eficiencia y están encapsulados por lo que su tamaño es reducido, apenas hacen ruido y necesitan menos mantenimiento ya que emiten poco calor y al ser herméticos no tienen problemas de suciedad.

Se ha optado por utilizar variadores de frecuencia debido a que es necesario disponer de este dispositivo para poder controlar un PMSM. Además, pueden contar con realimentación por encoder para conocer la posición exacta del rotor y disponen de la capacidad de controlar las magnitudes del motor de forma muy precisa.

Se ha seleccionado un variador SINAMICS S210 de la marca Siemens, este variador ya incorpora el motor síncrono de imanes permanentes y dispone de acceso a ciertas funcionalidades de control para autómatas de siemens. Por esto se ha seleccionado un equipo de la misma marca.

Además, se han empleado dos fuentes de alimentación de 24Vdc para alimentar la entrada de control de los variadores. Esto es una medida de seguridad bastante importante, ya que es posible extinguir la alimentación del motor sin apagar el variador, por lo que podrá mantener el par a velocidad 0, impidiendo que el eje caiga si pierde tensión en altura. El equipo elegido se muestra en la imagen 18.



Figura 18 Variador y motor elegido Fuente: Web Principal Siemens

Se ha simulado el SCADA de planificación con la aplicación RT Advanced de TIA portal en la cual se han representado todas las pantallas que se utilizarán en el proyecto. Este software de simulación es equivalente a la utilización de una pantalla HMI y se ha utilizado por la posibilidad de probar distintos modelos de pantalla sin tenerlos disponibles.

El lenguaje utilizado será ST, debido a la versatilidad y profundidad que permite. Además, se realizará el programa de forma modular mediante bloques de funciones. Al hacer esto se van a programar bloques de forma independiente los cuales se irán relacionando unos con otros lo que permitirá, como se ha comentado anteriormente, reutilizar la función en otros equipos.

En la imagen número 19 es posible apreciar los equipos seleccionados ya interconectados entre sí.

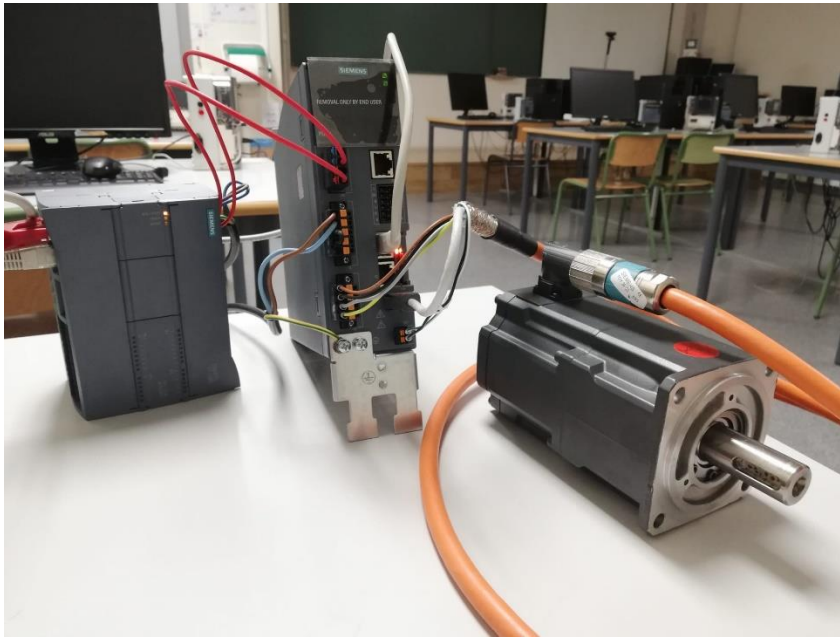


Figura 19 Solución adoptada

Cabe destacar que el PLC seleccionado no trabaja en modo isócrono por lo que habrá que desactivar esta opción en el variador el cual si que dispone de esta tecnología. Se ha decidido utilizar estos equipos ya que su conexión permite llegar a las funciones que interesan en este proyecto de una forma económica. En el futuro podría cambiarse el PLC por uno más potente y que nos permita explotar todas las funcionalidades del variador.

Utilizar la plataforma de Siemens es muy atractivo ya que su software dispone de muchas funciones preprogramadas que resultan muy útiles, al igual que integra un editor de pantallas HMI para realizar el SCADA.

Siemens es una de las empresas más extendidas en el sector de la automatización industrial, por lo que es muy fácil encontrar personal familiarizado con sus equipos. Todo esto ha propiciado que se elijan productos de entre su gama

8. Funcionamiento del programa

Tras elegir los equipos, se describirá tanto el proceso a automatizar como el funcionamiento que el programa debe seguir. Es importante entender la tarea que se busca mejorar, ya que un análisis completo puede facilitar la programación. Además, sería factible que al estudiar el proceso se decidiese no automatizarlo o abordarlo desde un frente distinto al planeado.

8.1 Descripción del proceso

El automatismo deberá de sustituir a un sistema tradicional de almacenamiento, el cual consiste en una nave industrial equipada con estantes en los cuales los operarios van catalogando los productos que entran por el muelle de carga y los colocan en la posición que se estipule. Una vez se requiere alguno de los productos almacenados, los trabajadores buscan en los archivos del almacén la localización de estos y realizan la entrega en la salida del almacén. La solución que se propone es la siguiente

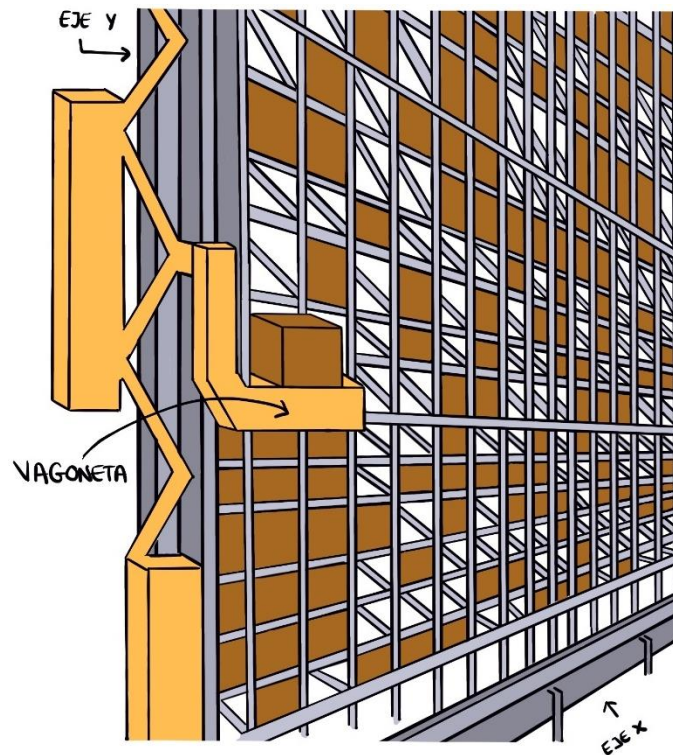


Figura 20 Ejemplo transelevador

Se dispondrá de un almacén vertical el cual cuenta con una cinta transportadora a la entrada y otra a la salida. El automatismo constará de dos railes accionados por motores independientes controlados por el mismo PLC. La primera función del automatismo será la de colocar los paquetes en el hueco más cercano y almacenar las posiciones en la memoria del PLC. El sistema también deberá de ser capaz de suministrar de forma eficaz estos paquetes a la cinta transportadora de salida siguiendo la jerarquía indicada. El programa cargado en el autómatas calculará la distancia que debe recorrer en ambos ejes para alcanzar la posición del almacén deseada.

8.2 Descripción del programa

A continuación, se explicará detalladamente el concepto que se ha seguido a la hora de programar la función. Se explicarán las acciones que deberá cumplir el código, como se programará, que elementos se necesitarán y como se abordará su diseño, es decir, que métodos proporcionados por el programa se utilizarán para conseguir que el motor se mueva según se espera.

Primeramente, se describirá la función y se mostrará un flujograma explicativo

El programa dispondrá de distintos modos y controles, buscando abarcar todas las necesidades posibles del cliente. Se explican a continuación dichos apartados:

MODOS

CONTROL AUTOMÁTICO

El modo principal del programa es el automático, en este, el PLC ordenará una serie de movimientos con el objeto de alcanzar posiciones determinadas del almacén. Estas posiciones serán calculadas de forma autónoma por el programa dependiendo del control seleccionado. El programa determinará la posición a la que debe acudir mediante una función que recorre la matriz en la que se encuentran representados todos los estantes del almacén.

El programa determinará el lugar más cercano y se dirigirá a esa posición para dejar o recoger el paquete.

Además, se dispondrá de un selector de modos que permitirá al operario decidir la operación que se va a llevar a cabo.

- Modo Carga

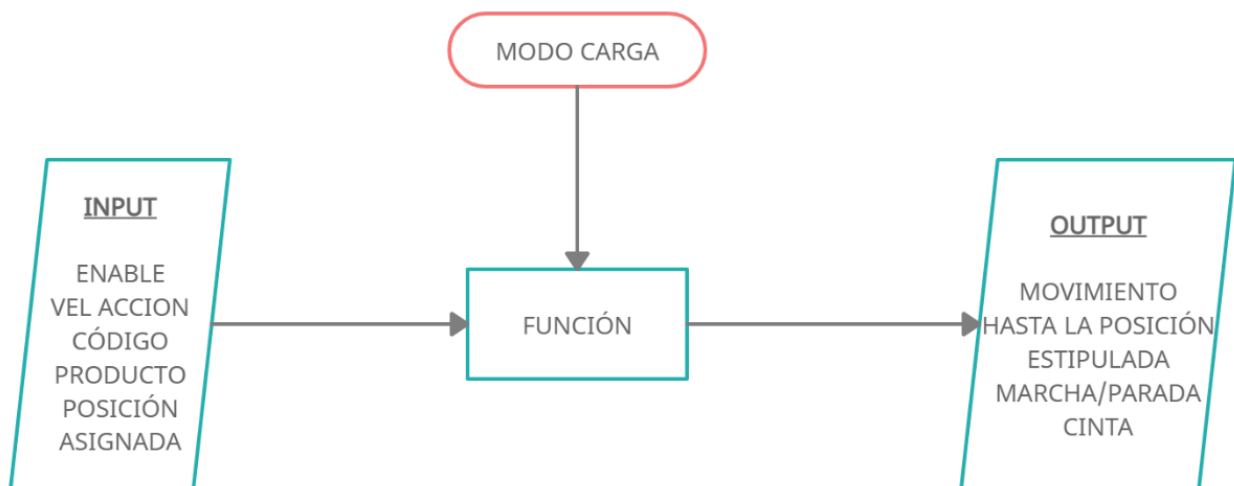


Figura 21 Flujograma carga

Cuando se seleccione este modo, el automatismo determinará la naturaleza del paquete de entrada mediante la combinación en las posiciones de 3 entradas digitales. Seguidamente, y dependiendo del modo de colocación elegido, se llevará cada paquete al lugar correspondiente. Este movimiento se irá ejecutando de forma repetitiva mientras el modo esté activo. Una vez se desactive, el sistema regresará a la posición inicial. El funcionamiento del código se aprecia en la imagen 21.

La función actúa de la siguiente manera tal y como puede verse en el flujograma de la imagen 22.

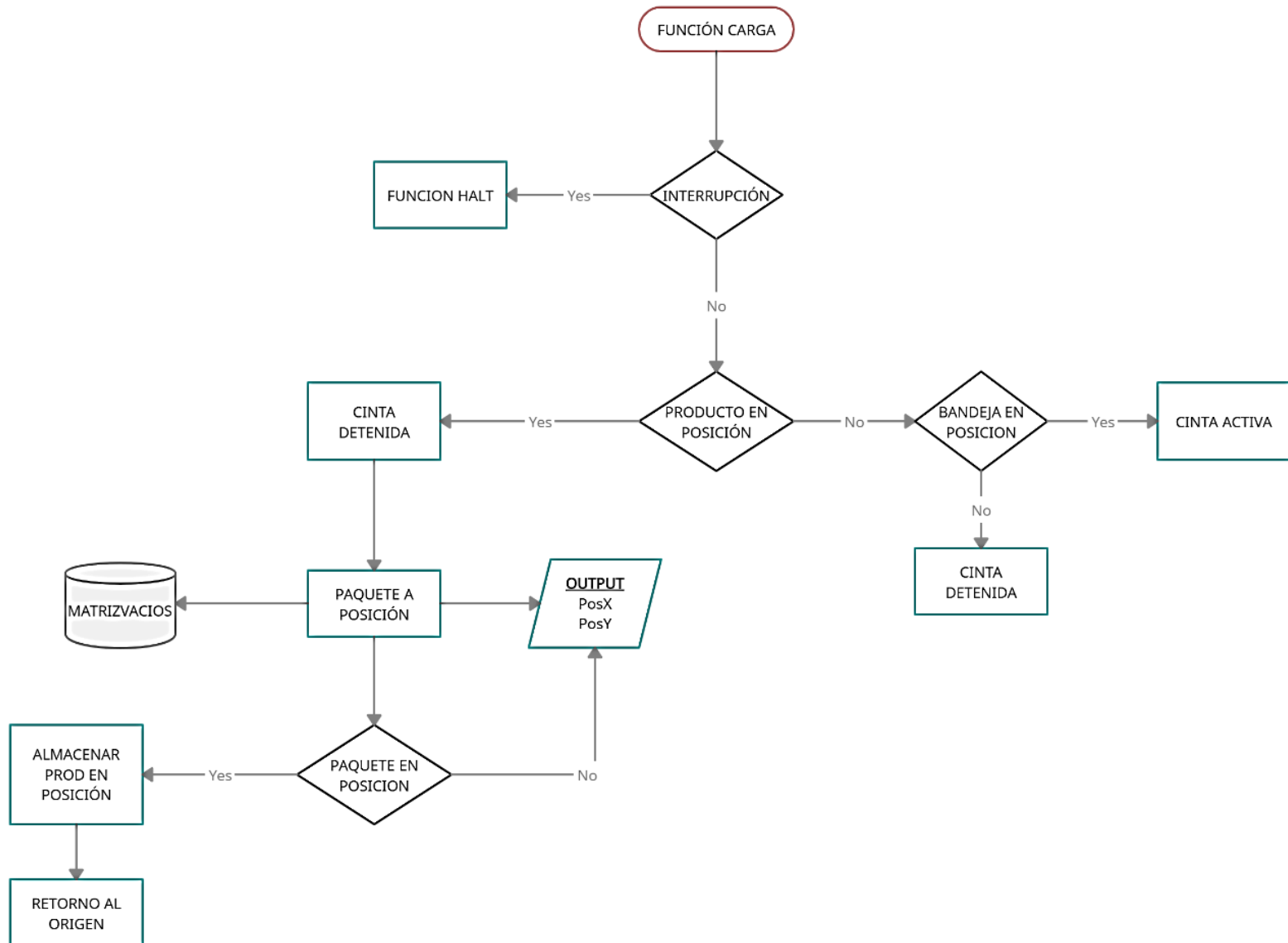


Figura 22 Flujograma función carga

- Modo Descarga

En cuanto a esta posibilidad, se programará la función para que, siguiendo un orden establecido, entregue el número y tipo exacto de paquetes demandados. Al igual que en el modo anterior, repetirá la maniobra de forma cíclica hasta cumplir con el pedido. Una vez se dé por terminada la actividad, este regresará al origen. El funcionamiento de este modo puede consultarse en la figura 23

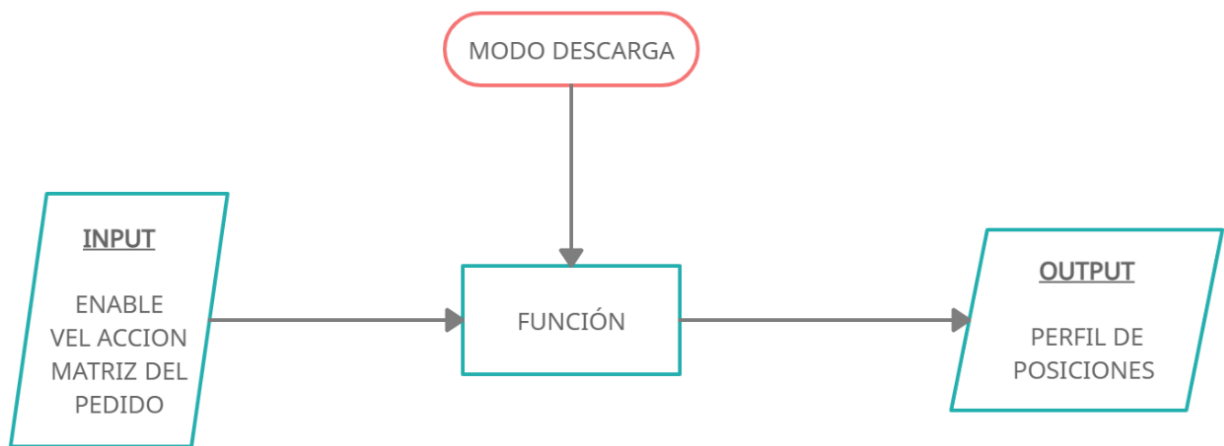


Figura 23 Flujograma descarga

Su funcionamiento se ejemplifica a continuación utilizando el flujograma de la imagen 24

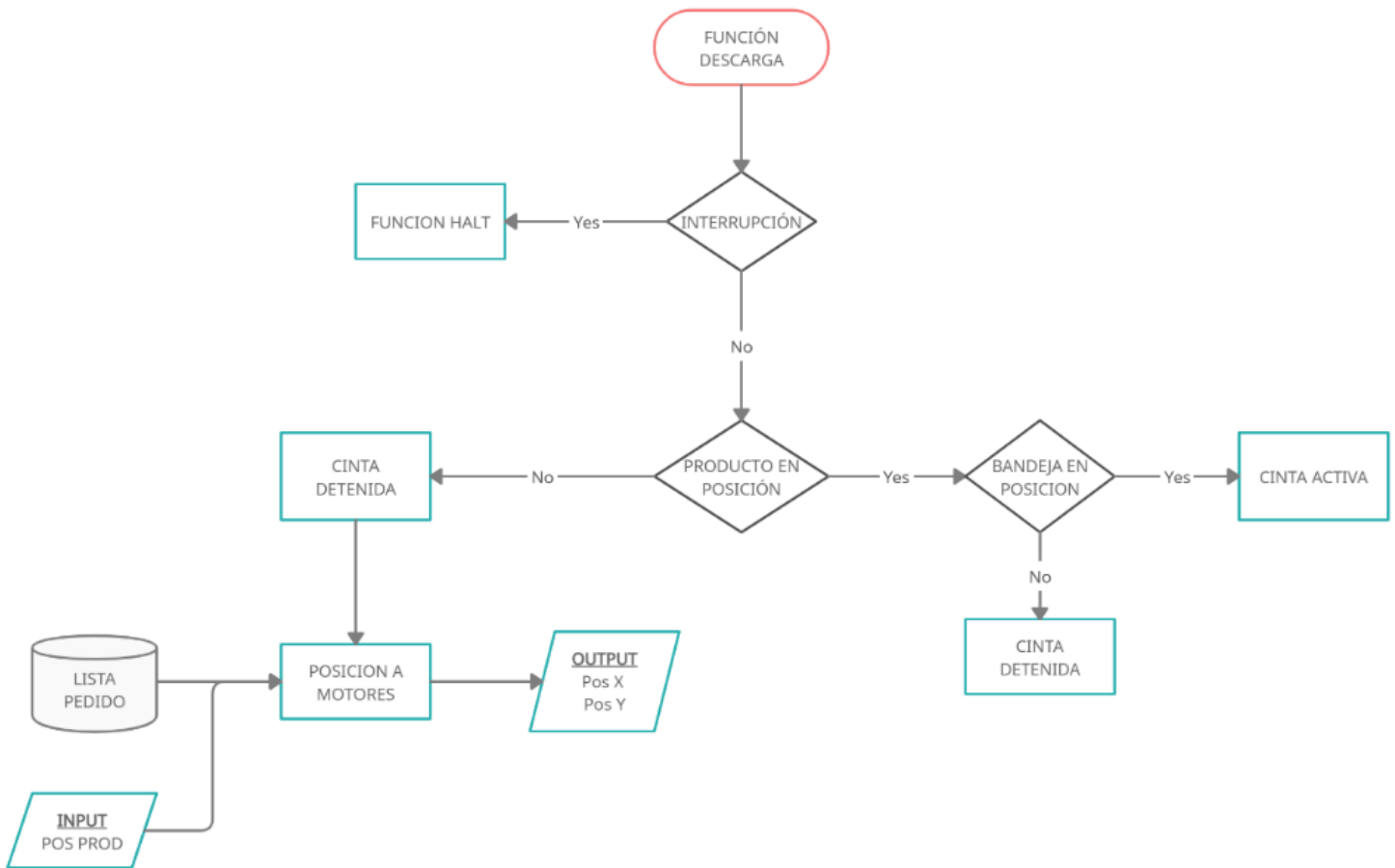


Figura 24 Flujograma función descarga

CONTROL MANUAL

Como complemento, se dispondrá de un modo manual el cual no tiene como objetivo el control del proceso si no permitir al operario mover a puntos determinados el automatismo si fuese necesario. Esto ocurriría cuando sea necesario accionar la máquina para su mantenimiento, cuando se haya detenido en una posición que no es adecuada, por ejemplo, tras un corte de tensión o cuando se quiere comprobar el correcto funcionamiento del accionamiento sin probarlo en un caso real. Se dispondrá de las siguientes opciones:

- JOG

El modo JOG habilitará un movimiento lento en una de las dos direcciones. Utilizando el bloque JOG de MC, se habilitará un movimiento a derechas o a izquierdas mientras se mantenga el botón presionado tal y como se observa en la figura 25

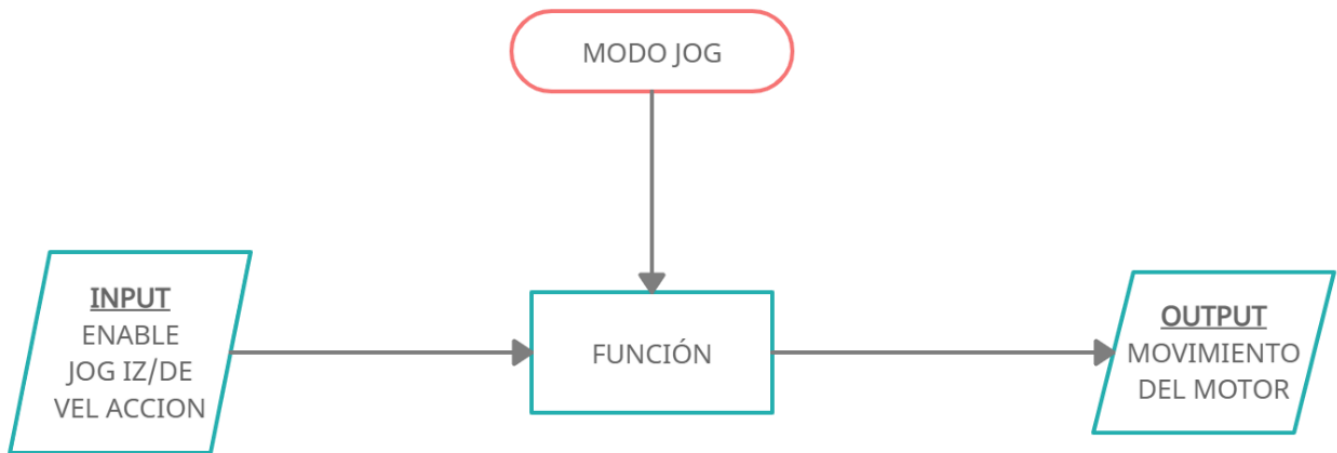


Figura 25 Flujograma JOG

- Retorno al origen

Esta función servirá para devolver los ejes a la posición x0 y0. Una vez se habilita, el motor girará utilizando la función MOVE_ABSOLUTE hasta el punto 0 que se haya referenciado como puede verse en la figura 26.

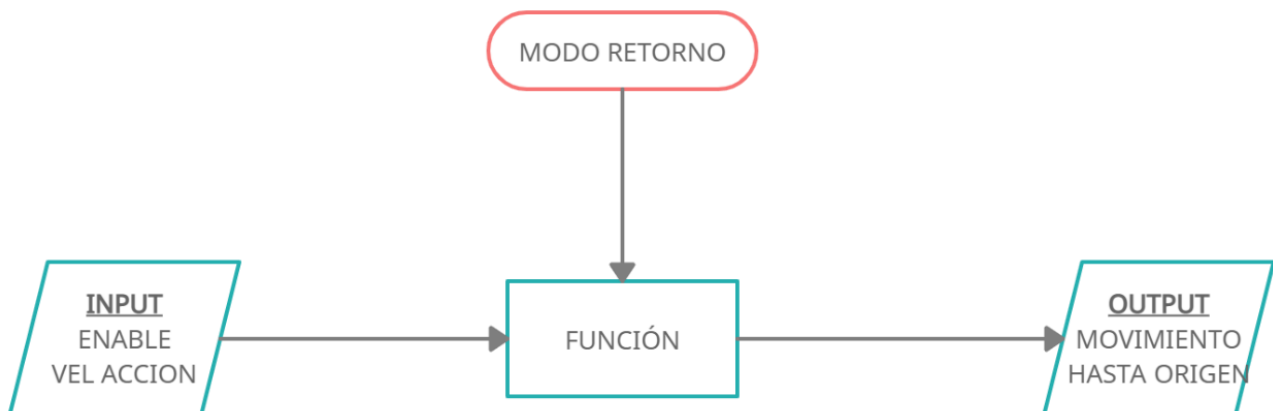


Figura 26 Flujograma retorno

- Pedido único

Permite seleccionar una posición exacta de la matriz y pedir el objeto que se encuentra en el mismo. Conociendo el paso del motor, se ejecuta la función MOVE_ABSOLUTE para mover los ejes hasta la posición deseada. Una vez se ha llegado hasta la posición, los ejes retornan automáticamente al origen. Esto se ejemplifica en la figura 27

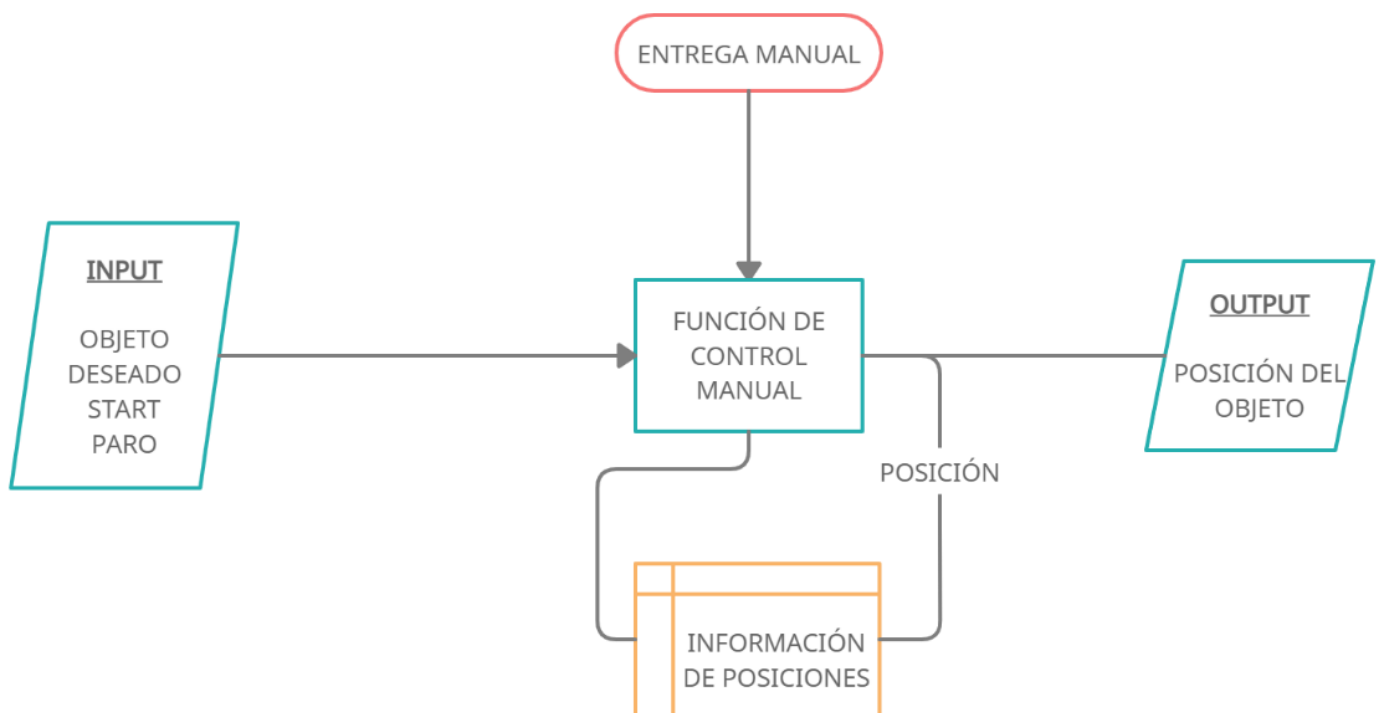


Figura 27 Flujograma manual

Por último, este modo permitirá referenciar los ejes a una posición determinada, esto se ha implementado ya que si el equipo se apaga de forma intempestiva puede perderse esta configuración y en este caso no podría moverse de forma absoluta.

Se necesitará disponer de bucles para recorrer las matrices, contadores y funciones de traspaso de datos y tipos de variables. Para transmitir las órdenes finales al VSD será necesario emplear objetos tecnológicos para cada motor y emplear bloques del ámbito Motion Control para acceder a la comunicación con el VSD y conseguir que el motor se dirija a la posición deseada, que despeje los errores internos o que se detenga.

9. Programación

En este punto se explicará cómo se ha creado el programa y cuál es la estructura que se decidió darle, todo dentro del entorno del programa TIA PORTAL. Por último, se comentará el orden que se ha seguido para declarar las variables y la forma en la que se ha decidido almacenar la información. El uso del programa TIA PORTAL ha sido explicado en el Anexo II “Manual TIA PORTAL”

9.1 Creación de la función

En este subapartado se mostrará el razonamiento que se ha seguido para determinar los distintos bloques del proceso atendiendo a los principios de la programación orientada a objetos.

En primer lugar, se ha subdividido el proyecto en distintas tareas, de esta manera, se programarán estas acciones de forma independiente. La unión de estos fragmentos actuará como el control completo del proceso.

Se han acotado como bloques la función comunicación con el variador, gestión de almacén, manual, carga y descarga como se muestra en la figura 34.

```
(* LLAMADA A LA GESTION DEL ALMACÉN *)  
E "GESTORPOSICIONALMACEN" (ENABLE:="AUXAUTO", ...);  
  
(* LLAMADA A LA FUNCION COMUNICACION *)  
E "CONFIGURACION_DB" (ENABLE := "AUXCONFIG", ...);  
  
(*LLAMADA A LA FUNCIÓN MANUAL*)  
E "MODOMANUAL_DB" (HABILITAR:="AUXMANUAL", ...);  
  
(* LLAMADA A LA FUNCIÓN "CARGA" *)  
E "CARGA_DB" (ENABLE:="AUXLLENADO", ...);
```

Figura 28 Ejemplo FBs

COMUNICACIÓN CON EL VARIADOR

En primer lugar, se dispondrá de una función encargada de enviar las órdenes al variador. Esto se ha realizado con las herramientas Motion Control las cuales permiten encender o apagar el motor, establecer control de posición, velocidad o par y monitorizar el estado de dichos comandos

```

1 #MC_Power_Instance_1(Axis:= "Axis_1", ...);
10 #MC_Reset_DB_1(Axis:="Axis_1",
11     Execute:=#I_AX1_MC.EX_RES,
12     Restart:=#I_AX1_MC.REST_RES,
13     Done=>#Q_AX1_MC.DON_RES,
14     Busy=>#Q_AX1_MC.BUSY_RES,
15     Error=>#Q_AX1_MC.ERR_RES,
16     ErrorID=>#Q_AX1_MC.ID_RES,
17     ErrorInfo=>#Q_AX1_MC.INFO_RES);
18 #MC_Home_DB(Axis:="Axis_1", ...);
29 #MC_Halt_Instance(Axis:="Axis_1", ...);
37 #MC_MoveAbsolute_Instance(Axis:="Axis_1", ...);
48 #MC_MoveRelative_Instance(Axis:="Axis_1", ...);
58 #MC_MoveVelocity_Instance(Axis:="Axis_1", ...);
70 #MC_MoveJog_Instance_1(Axis:="Axis_1", ...);
--

```

Figura 29 Funciones Motion Control

. Se gestionarán las entradas y salidas mediante una estructura de datos para facilitar el uso de las variables. Con este sencillo paso será posible utilizar tantos motores como se quiera ya que la función actuará por igual para todos los ejes. En la imagen 31 se muestra la estructura de datos mientras que en la número 32 aparece el código encargado de esta tarea.

1	Static		
2	INPUTAXIS1	Struct	
3	OUTPUTAXIS1	Struct	
4	Stat_POW	Bool	false
5	Busy_POW	Bool	false
6	ERR_POW	Bool	false
7	ID_POW	Word	16#0
8	INFO_POW	Word	16#0

Figura 30 Estructura función del eje

GESTIÓN DEL ALMACÉN

Cuando se utiliza el control automático, el PLC decide la posición del almacén en la que se deposita la caja cuando se encuentra seleccionado el modo descarga. Sin embargo, en el modo descarga la posición a la que se manda el motor es a aquella donde se encuentra el último paquete depositado del producto que se esté demandando.

```
(* LLAMADA A LA GESTION DEL ALMACÉN *)
"ALMACEN_DB" (ENABLE := "AUXAUTO",
              CARGA := "AUXLLENADO",
              DESCARGA := "AUXVACIADO",
              PRODUCTO := "PRODUCTO",
              MENOS := "AUXMENOS",
              MAS := "AUXMAS",
              POSX => "AUXPOSX",
              POSY => "AUXPOSY");
```

Figura 31 Bloque Almacén

Este bloque gestiona las memorias que almacenan dichas posiciones y se encarga de editarlas para que siempre estén en sincronismo con la realidad. El mayor beneficio de hacer esto es que no es necesario leer las matrices al completo si no que la orden exacta se ofrece a través de este bloque.

```
IF #CARGA THEN
  #POSX := "DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].X;
  #POSY := "DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].Y;
END_IF;
IF #DESCARGA THEN
  #POSX := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].X;
  #POSY := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].Y;
END_IF;
```

Figura 32 Fragmento código del bloque

El funcionamiento de las estructuras de datos que almacenan la situación del almacén se explica en el punto 9.2 “Declaración de variables y justificación”.

CONTROL MANUAL

La actuación del equipo cuando se ha seleccionado el control manual o de mantenimiento se realiza desde este bloque. Dependiendo del modo interno que se haya seleccionado las órdenes que se transmiten al variador son distintas. Este bloque es el más sencillo de todos y no interactúa con las memorias ya que no recoge ningún producto.

```
"MODOMANUAL_DB" (HABILITAR := "AUXMANUAL",
                  "JOG" := "JOG",
                  "MANUAL" := "PEDIDO",
                  "RETORNO" := "RETORNO",
                  SALIDAX => "INPUTS/OUTPUTS".INPUTSX_MAN,
                  SALIDAY => "INPUTS/OUTPUTS".INPUTSY_MAN);
```

Figura 33 Bloque Control Manual

CARGA

El modo carga y el modo descarga son las dos mitades que componen al control automático. Cuando está seleccionado, identifica el paquete que se está cargando y se encarga de activar y desactivar la cinta dependiendo si el transelevador se encuentra actuando sobre la estantería o está detenido esperando recibir el siguiente producto.

```
IF "AUXLLENADO" THEN
  (* LLAMADA A LA FUNCIÓN "CARGA" *)
  "CARGA_DB" (ENABLE:="AUXLLENADO",
              PRODUCTO:="PRODUCTO",
              LISTO:="LISTO",
              POSX:="AUXPOSX",
              POSY:="AUXPOSY",
              MARCHACINTA=>"Tag_1",
              SALIDASX=>"INPUTS/OUTPUTS".INPUTSX_CARGA,
              SALIDASY=>"INPUTS/OUTPUTS".INPUTSY_CARGA,
              MENOS=>"AUXMENOS");
END_IF;
```

Figura 34 Bloque Modo Carga

También transmite al variador la posición objetivo calculada por el bloque de gestión del almacén y le da la orden de eliminar la posición ahora ocupada del vector de vacíos, editar el correspondiente vector de productos almacenados y guardar el producto exacto en la posición adecuada del almacén. Por último, realiza el conteo de los productos totales que hay en la instalación.

DESCARGA

En cuanto al modo descarga, actúa de forma similar a la carga. Este modo también acciona la cinta transportadora pero en sentido inverso ya que está suministrando paquetes hacia la bahía de carga y transmite al variador la posición del producto que se busca extraer. Esta función se encarga de transmitir al gestor que posición acaba de quedar vacía para transferirla a la matriz de productos vacíos.

```
IF "AUXVACIADO" THEN
  "DESCARGA_DB" (ENABLE_DESCARGA:="AUXVACIADO",
    SALIDA:="Tag_2",
    PEDIDOS:="PEDIDOS".PEDIDOS,
    MARCHACINTA=>"Tag_1",
    MAS=>"AUXMAS",
    SALIDASX=>"INPUTS/OUTPUTS".INPUTSX_DESCARGA,
    SALIDASY=>"INPUTS/OUTPUTS".INPUTSY_DESCARGA,
    VACIO=>"VACIO");
END_IF;
```

Figura 35 Bloque Modo Descarga

Como se puede observar, estas secciones actúan sobre actividades muy genéricas que pueden extrapolarse a instalaciones completamente distintas y es ahí donde se encuentra uno de los valores de la programación orientada a objetos. Además, permite disponer de distintos trabajadores que se encarguen de partes distintas del código ya que solo habría que unirlos y coordinarlos una vez haya que ensamblar el código final.

El código del programa podrá consultarse en el Anexo IV "Listado del Programa".

9.2 Declaración de variables y justificación

Para disponer de los datos necesarios de forma ordenada, se dispondrá de distintas variables, como puede verse en la imagen, las posiciones del almacén serán consultadas mediante matrices y estructuras de datos, se declararán las siguientes:

- Gestión del almacén

Se declarará un bloque de datos en el cual se almacenan distintas estructuras. Con estas se controlará la posición de la matriz que debe ser consultada para depositar un producto, la posición a la que se debe ir a recoger cada uno de estos, las posiciones en las que se ha almacenado cada uno de los productos y cuál es el producto de cada una de las posiciones del almacén lo cual se representa en la imagen 35.

2	▶	VALOR ACCESIBLE	Array[0..9] o...
3	▼	Matrizproductos	Array[0..7] of Struct
4	▶	Matrizproductos[0]	Struct
5	▶	Matrizproductos[1]	Struct
6	▶	Matrizproductos[2]	Struct
7	▶	Matrizproductos[3]	Struct
8	▶	Matrizproductos[4]	Struct
9	▶	Matrizproductos[5]	Struct
10	▶	Matrizproductos[6]	Struct
11	▶	Matrizproductos[7]	Struct
12	▶	MATRIZALMACEN	Array[0..5, 0..5] of l...
13	▶	MATRIZVACIOS	Array[0..24] of Struct

Figura 36 Datos almacén

- Gestión de los pedidos

Mediante este bloque se recibirá la información de los pedidos gestionados a través del HMI, los cuales se utilizan para determinar a las posiciones que hay que ir en el modo descarga del automático. Mediante las variables de estas estructuras se le comunica al PLC la cantidad de producto que debe ser entregada. La estructura del vector se muestra en la figura 36.

Element	Type
RECETAS	Array[0..2] o...
RECETAS[0]	Struct
RECETAS[1]	Struct
Pasta	Int
Carnes	Int
Lacteos	Int
Verduras	Int
Cerveza	Int
Bebidas	Int
Snacks	Int
Conservas	Int
RECETAS[2]	Struct

Figura 37 Datos recetas

- Gestión de la función del eje

Finalmente, se han declarado varias estructuras de datos para enviar las órdenes al variador y recibir realimentación de este y actuar sobre el proceso dependiendo de los modos. Mediante las variables individuales de estas estructuras se puede habilitar la alimentación del motor, ordenar un movimiento referenciado, la ejecución de un JOG o una parada de emergencia. También es posible recibir información, como por ejemplo si se ha ejecutado la orden o si ha habido alguna clase de error. Puede verse la llamada del bloque en la figura 37

```
"POSAXIS_DB"(I_AX1_MC := "AXIS1INPUT/OUTPUT".INPUTAXIS1,
              Q_AX1_MC => "AXIS1INPUT/OUTPUT".OUTPUTAXIS1);
```

Figura 38 Estructuras de la función

10. Funcionamiento del SCADA

El SCADA es una parte fundamental del proyecto ya que será la interfaz mediante la que el operario actuará sobre la máquina. Las pantallas se simularán en la herramienta WinCC de TIA PORTAL. El funcionamiento de este software se ha explicado en el Anexo III "Manual WinCC RT ADVANCED.

Sin embargo, aquellas relacionadas con el control directo del programa se encontrarán en un HMI a pie de fábrica mientras que las que permiten acceso a la configuración y gestión del proceso se encontrarán en un PC industrial situado en una ubicación más cómoda. La diferencia se encuentra en que una pantalla HMI es un dispositivo que permite interactuar con la máquina, habilitando su control el tiempo real.

Un SCADA permite supervisar el proceso, pudiendo monitorizar valores del mismo y actuar sobre este de forma mas compleja y completa. Normalmente se utilizan HMIs a pie de fábrica como herramienta del operario y SCADAs situados en ordenadores que se encuentren en entornos de trabajo más cómodos, como despachos u oficinas. El funcionamiento del SCADA queda recogido en el Anexo I "Manual de funcionamiento del programa"

A continuación, se mostrarán las distintas pantallas de la interfaz de planificación y en estas se mostrará el funcionamiento mediante el uso de flujogramas.

En todas las pantallas se mostrará la fecha y la hora, el usuario activo y el estado de la máquina mediante un semáforo virtual. En la parte inferior de estas se podrá navegar entre las opciones sin necesidad de regresar al inicio. Para dotar a todas las pantallas de un formato y funcionalidades idénticas se han empleado fondos de plantilla tal y como se muestra en la imagen 38.



Figura 39 Plantilla SCADA

En aquellos modos que se considere necesario se incluirán representaciones gráficas del proceso, así como se utilizará un código de colores para diferenciar los distintos productos almacenados en las estanterías o conocer la posición exacta a la que se está yendo. En la imagen 39 puede verse un ejemplo.

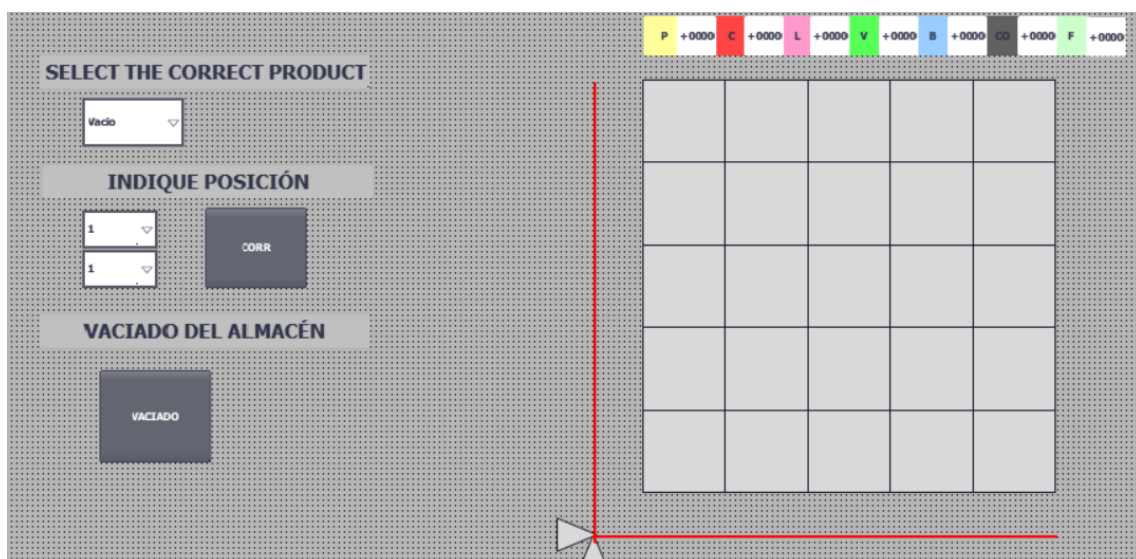


Figura 40 Ejemplo Animación

El SCADA dispondrá de una página principal en la que se realizará el Login del usuario y, dependiendo de los permisos concedidos al mismo, se permitirá el acceso a unas ventanas u otras. Se considerarán tres niveles de acceso, el más alto permitirá gestionar el proceso, el nivel intermedio habilitará al usuario a realizar movimientos manuales de la máquina para su mantenimiento y el más

limitado solo consentirá accionar el proceso. Los usuarios se darán de alta en la base de datos del programa, quedando ahí registradas las contraseñas y autorizaciones. La pantalla del menú se encuentra en la imagen 40.

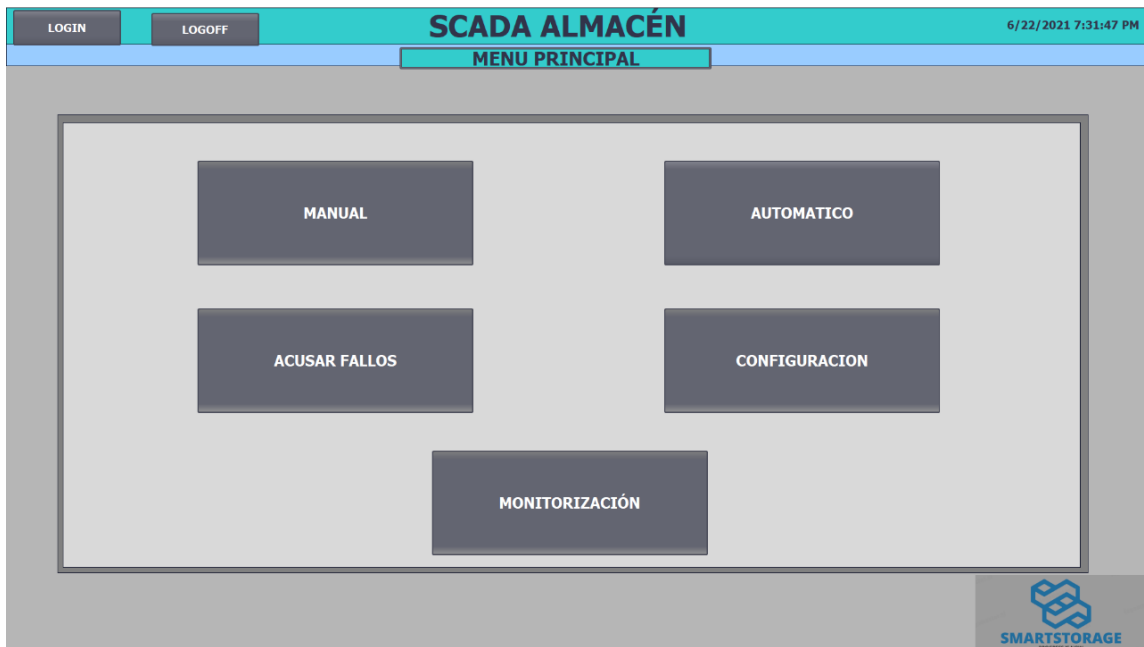


Figura 41 Menú Principal

Como puede verse en el flujograma de la imagen 41, desde el menú es posible acceder a todas las pantallas del SCADA.

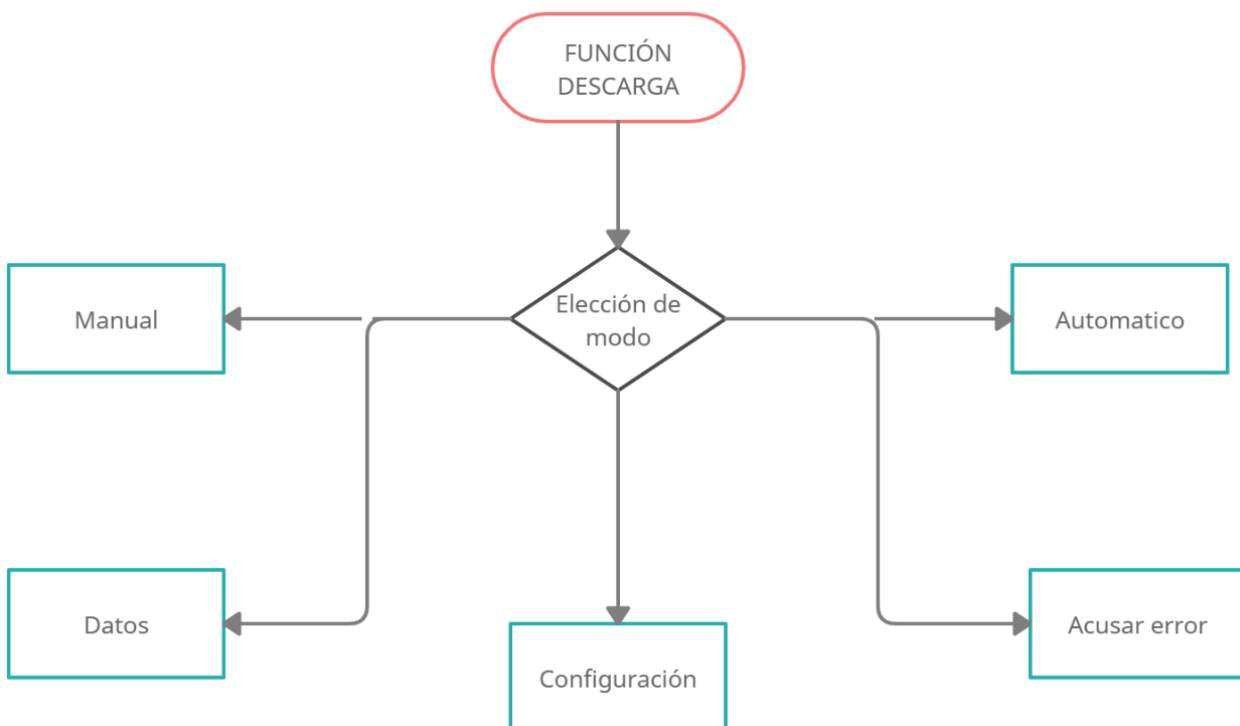


Figura 42 Flujograma Menú

A continuación, se hablará de cada una de estas pantallas

- La pantalla de configuración

Permitirá gestionar las opciones de pedidos especificadas. Esto se realizará mediante la inclusión de recetas en el HMI y su unión a un bloque de datos en el PLC. Según el manual de RT Advanced de SIEMENS, la función recetas permite visualizar, editar y crear estructuras de datos. (SIEMENS, 2018). La interfaz de esta función aparece en la imagen 42.

The screenshot shows the 'Recetas HMI' interface. It contains two main tables:

Name	Display name	Number	Version	Path	Type	Maximum number of d..	Communication type	Tooltip
BLASCO	BLASCO	1	6/15/2021 12:0...	C:\Recipes	Limited	5000	Tags	
CEDRO	CEDRO	2	6/15/2021 12:1...	C:\Recipes	Limited	5000	Tags	
ARAGÓN	ARAGÓN	3	6/15/2021 12:1...	C:\Recipes	Limited	5000	Tags	

Name	Display name	Tag	Data type	Data length	Default value	Minimum value	Maximum value	Decimal places	Tooltip
PASTA	PASTA	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
CARNES	CARNES	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
LACTEOS	LACTEOS	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
VERDURAS	VERDURAS	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
BEBIDAS	BEBIDAS	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
CONSERVAS	CONSERVAS	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	
FRUTAS	FRUTAS	PEDIDOS_PEDIDO...	Int	2	0	-32768	32767	0	

Figura 43 Recetas HMI

Además, se dispondrá de la posibilidad de exportar e importar la información en un archivo .csv, lo cual permitirá acceder a niveles más altos de gestión. Como función adicional, habilitará introducir nuevos usuarios, asignar contraseñas y nivel de seguridad. Finalmente, será posible cambiar entre idiomas, alemán, francés, inglés, castellano y valenciano como se puede apreciar en la figura 43

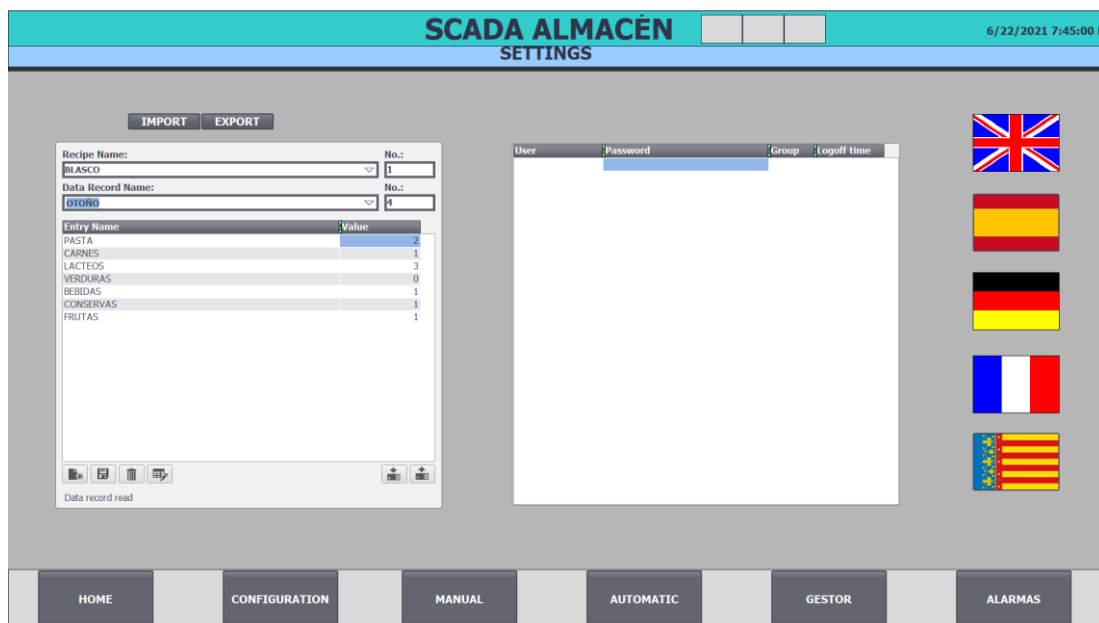


Figura 44 Pantalla de Configuración

- El control manual

El cual quedará habilitado para el encargado del mantenimiento de la máquina y permitirá mover los ejes manualmente, pedir a la máquina que retorne a 0, solicitar el movimiento a una posición determinada o referenciar los ejes en caso de que un apagón produzca que el proceso se detenga. Además, se incluirá una animación que muestra la posición de los ejes. La pantalla aparece en la imagen 44.

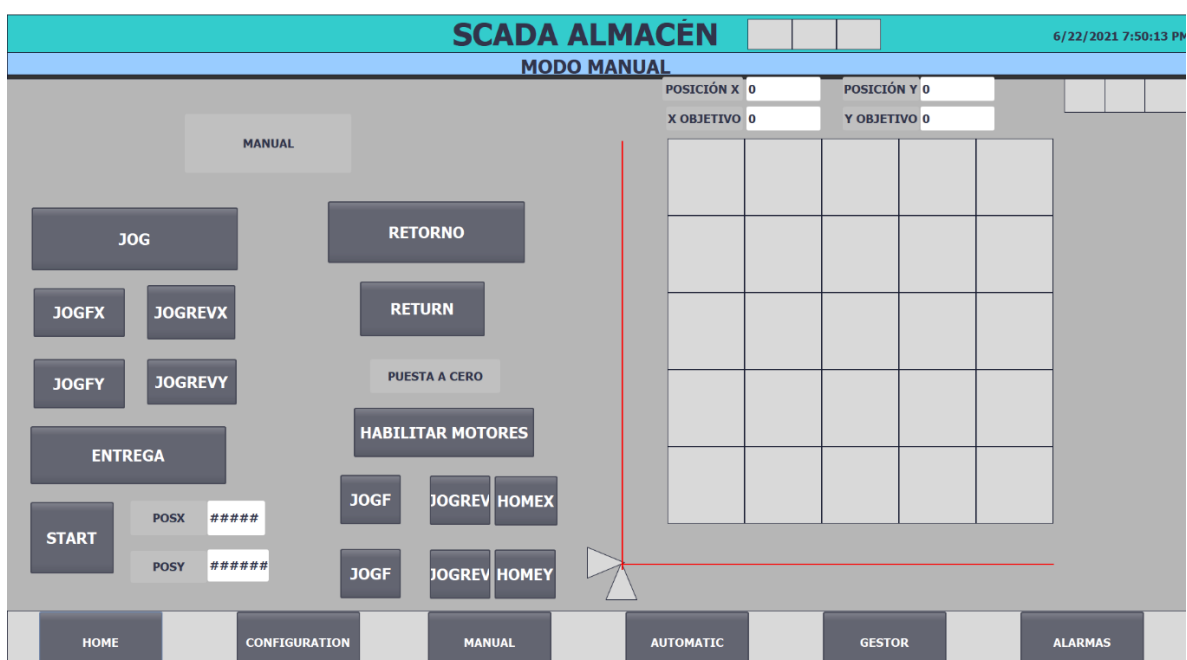


Figura 45 Modo Manual

En el flujograma 45 se puede ver como se elige entre el modo manual y la puesta a 0, dentro del modo manual, se puede seleccionar el modo JOG, para comprobar que los ejes se activan correctamente, el retorno a 0 y solicitar una entrega de una posición estipulada en el display. En la puesta a 0, se podrá mover el eje de forma manual y referenciarlo a 0 en el lugar que queramos, esto se ha diseñado con la intención de poder dar un 0 al sistema en caso de que se desconfigure ya que los movimientos que se mandan al eje son absolutos.

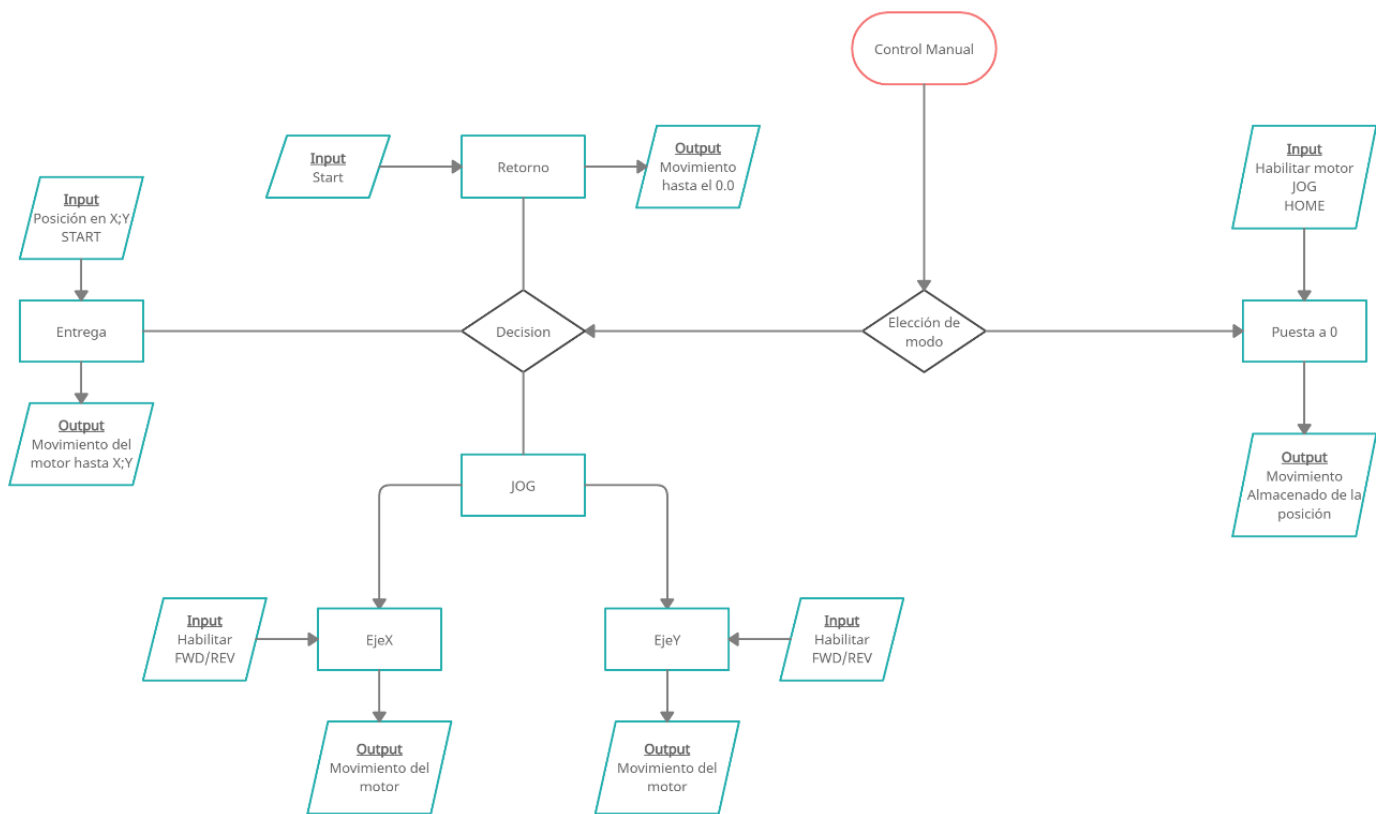


Figura 46 Flujograma Manual

- El control automático

Es el modo principal de ejecución y se espera que sea la pantalla que más tiempo este activa, permitirá seleccionar modo carga y modo descarga, iniciar la alimentación de los motores y arrancar el proceso. En el modo descarga se podrá elegir el pedido mediante un desplegable y cuenta con un botón de marcha. Es posible consultar la interfaz en la imagen 47.

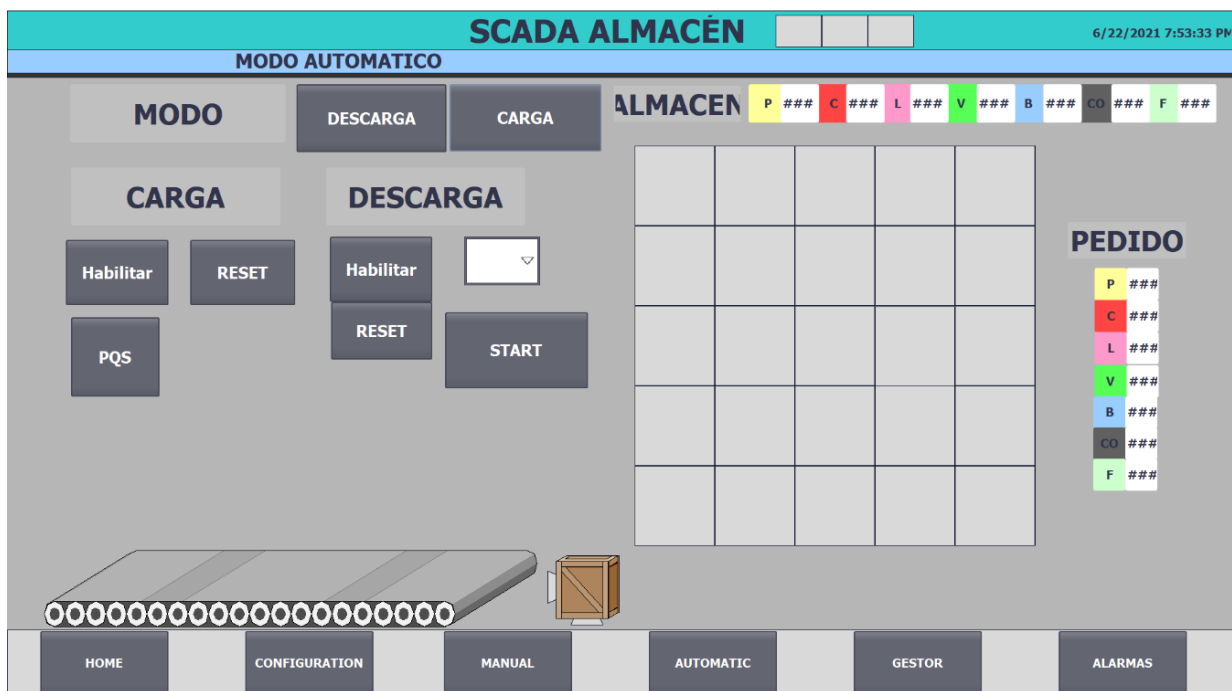


Figura 47 Pantalla Automático

Se emplearán bloques de datos para almacenar las posiciones a las que debe dirigirse el automatismo en cada movimiento.

Finalmente, se dispondrá de elementos gráficos para conocer el estado del proceso, tales como una animación que monitoriza en tiempo real el estado del almacén, una tabla que registra el número de productos almacenados y pilotos del estado de la cinta y los sensores. Se utilizarán imágenes bitmap para diferenciar los productos que se van almacenando.

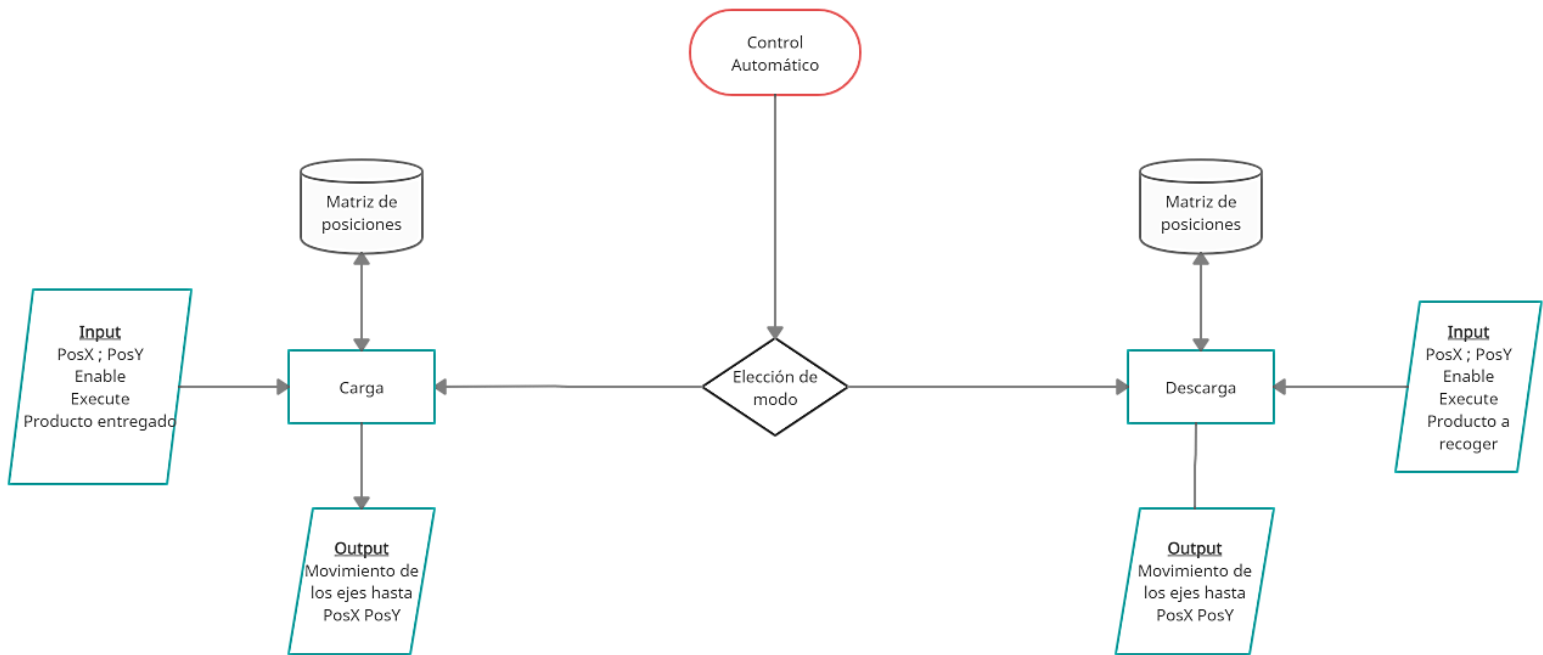


Figura 48 Flujoograma Automático

- Acusado de errores

La pantalla de acusado de errores permitirá modificar las posiciones del almacén en caso de que se haya producido un error del programa, también consentirá vaciar completamente el almacén. El objeto y la posición se seleccionará en los desplegables y al pulsar el botón de corrección actualizará la posición de la memoria. Además, el botón vaciado hará una limpieza total de la memoria. La pantalla en cuestión se aprecia en la imagen 49.

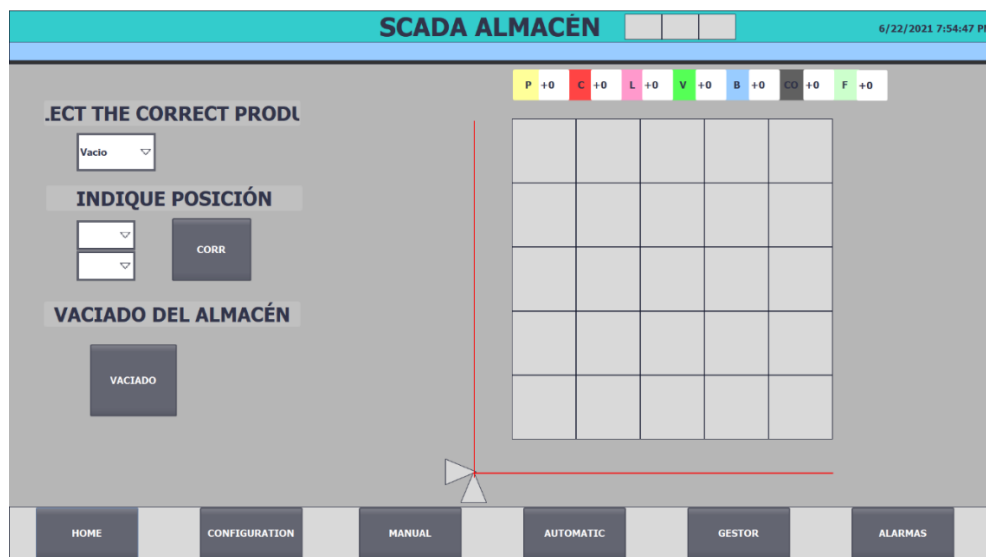


Figura 49 Interfaz Corrección de Errores

Siendo su flujograma el que se muestra a continuación en la imagen 50.

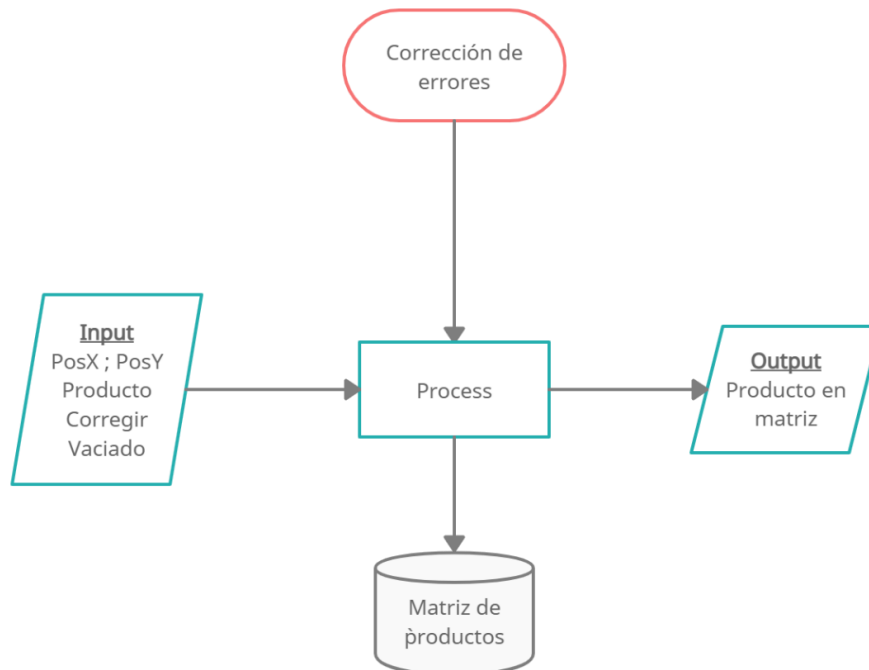


Figura 50 Flujograma Corrección de Errores

- Monitorización

Como añadido, se dispondrá de una pantalla de datos en la que se monitorizarán variables que se consideren importantes para comprobar que todo funciona correctamente, tales como velocidades, numero de productos almacenados, historial de errores, etc. Se muestra la pantalla de monitorización en la siguiente imagen, número 51.



Figura 51 Interfaz de Datos

- Pantalla global de errores

Por último, se añadirá una pantalla global que se activará en cualquier modo cuando un error salte. Se configurarán errores de protecciones eléctricas o de salidas de recorrido. Se editarán avisos que notifiquen que el almacén está vaciándose o a punto de completar sus posiciones. La pantalla emergente de errores aparece en la figura 51.

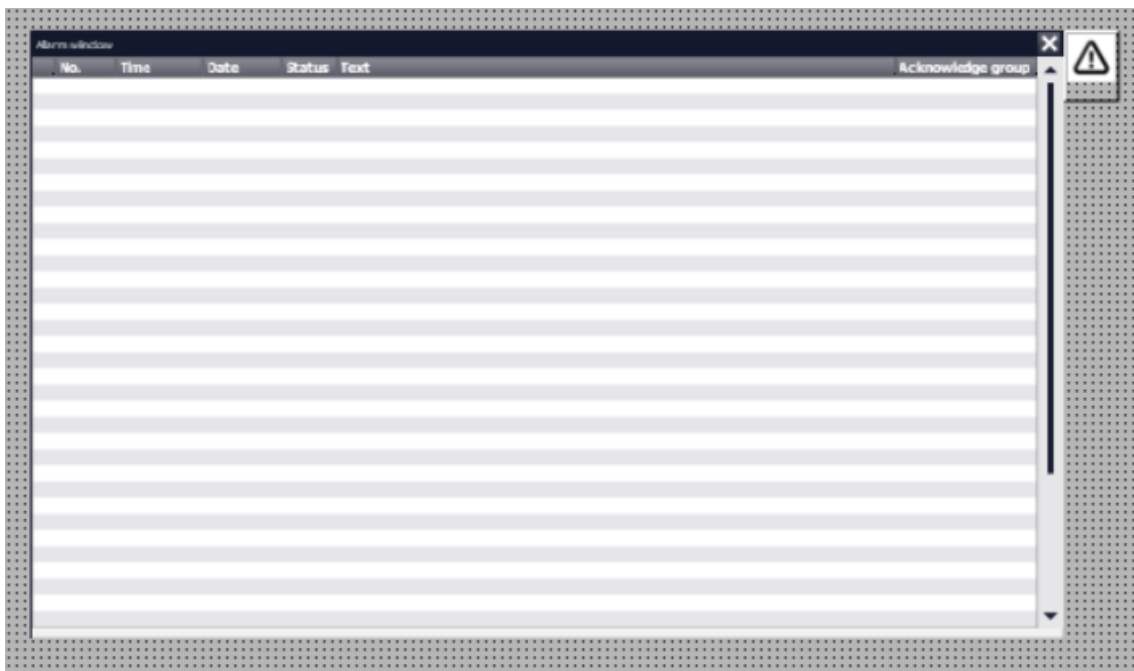


Figura 52 Pantalla Global de Errores

El Anexo I “Manual SCADA” contiene información acerca de cómo operar todos los paneles tal y como han sido diseñados.

11. Conclusiones

Para concluir se puede decir que a lo largo del proyecto se ha realizado el análisis, diseño, ejecución e implantación de una función de control de posicionado de carácter genérico y orientada a objetos. Además, se ha llevado a cabo la selección, parametrización e instalación del equipo físico correspondiente al cual se le han realizado pruebas de funcionamiento con resultado satisfactorio.

Desde un punto de vista de gestión, se puede decir que la implementación de automatismos en el proceso de almacenaje y suministro que realiza un almacén es eficiente. En la maniobra de carga, no es necesario comprobar los paquetes ni mantener informes de almacenamiento ya que se encarga el PLC en ambos casos mientras que en la maniobra de descarga se utiliza esta información para descargar los paquetes ya que se conoce su posición.

Además, también ocupan menores espacios ya que se puede conseguir la misma superficie con almacenes más altos en vez de anchos ya que no hay ningún problema en elevar los motores a alturas superiores de las que se hace con los trabajadores y puede estar trabajando durante más tiempo y a horas intempestivas.

Finalmente, permite ofrecer prestaciones más competitivas como menores tiempos de carga y descarga, ya que como se ha mencionado son más rápidos y una posibilidad de error muy reducida.

En cuanto a la función, se ha podido comprobar que la programación orientada a objetos es una estrategia utilizable en este tipo de proyectos ya que se ha ganado tiempo utilizando las mismas funciones en distintos modos. Esto ahorra mucho tiempo ya que es posible utilizar la misma función para distintas tareas y además en caso de surgir errores basta con encontrarlo en un bloque para poder corregirlos todos.

PLANOS

ÍNDICE PLANOS

Plano I. Conexionado eléctrico.....	81
Plano II. Red Profinet.....	82
Plano III. Conexiones PLC.....	83
Plano IV. Conexiones variador.....	84

Plano I. Conexionado eléctrico

Plano II. Red Profinet

Plano III. Conexiones PLC

Plano IV. Conexiones variador

PLIEGO DE CONDICIONES

ÍNDICE PLIEGO DE CONDICIONES

1. Introducción.....	88
2. Objetivo del pliego de condiciones	89
3. Normativa vigente.....	90
4. Condiciones técnicas.....	91
4.1. Cableado eléctrico.....	91
4.2. Interruptores y pulsadores.....	91
4.3. Pulsadores de emergencia y demás elementos de seguridad.....	92
4.4. Fuente de alimentación.....	92
4.5. Ordenador personal.....	92
4.6. Software.....	93
4.7. PLC.....	93
4.8. Variador de frecuencia.....	93
4.9. Motor.....	94
5. Condiciones constructivas.....	95
5.1. Montaje eléctrico.....	95
5.2. Pruebas de funcionamiento.....	95
5.3. Comprobación visual del conexionado.....	93
5.4. Pruebas en tensión.....	96
5.5. Prueba final.....	96
5.6. Control de calidad.....	96
6. Condiciones de ejecución.....	98
6.1. Compra de material.....	98
7. Condiciones económicas.....	99
7.1. Pago de los trabajos.....	99
8. Condiciones legales.....	100
8.1. Contrato.....	100
8.2. Arbitraje y jurisdicción.....	100
8.3. Impuestos.....	100
8.4. Recisión del contrato.....	101
8.5. Condiciones facultativas.....	101
8.6. Derechos y deberes del contratista.....	101

1. Introducción

El pliego de condiciones es un documento de carácter contractual en el cual se señalan las relaciones entre el contratista y el ejecutor del proyecto. En el mismo se fijan las exigencias legales, económicas y de ejecución que han de tenerse en cuenta para la correcta puesta en marcha y utilización del proyecto. Se dividirá el presente pliego en descripción del proyecto, legislación, normativa y dispositivos utilizados; condiciones requeridas de las instalaciones, prescripciones de ejecución, dirección y recepción de obra.

2. Objetivo del pliego de condiciones

El objetivo de este documento es regular la ejecución del proyecto fijando los niveles técnicos y de calidad exigibles.

En este apartado se procederá a definir los requerimientos legales, económicos técnicos y facultativos del proyecto. Se definirán los aspectos técnicos que presenta el proyecto y se realizará una descripción de los productos empleados. Se estudiarán cuatro tipos de condiciones:

- **Condiciones técnicas:** se hará referencia a las tareas a realizar, a las características necesarias de los materiales y se hace referencia a los trabajos que hay que realizar. También se definirán las condiciones adecuadas de ejecución y control del proceso diseñado, así como los ensayos de calidad.
- **Condiciones facultativas:** se dejará constancia de los derechos y obligaciones contraídos por ambas partes en el momento de la ejecución del proyecto.
- **Condiciones económicas:** se determinará la forma de pago y las indemnizaciones por el incumplimiento de este, se justificará la formación de precios y por último se hará referencia a las garantías.
- **Condiciones legales:** En este apartado se dejará constancia del perfil de contratista, el tipo de contrato utilizado, el proceso de adjudicación y se incidirá en la obligatoriedad a la hora de cumplir con las normativas vigentes.

Mediante la presentación de estos apartados se busca realizar una descripción detallada de la normativa a la cual deben adherirse ambas partes y de los controles de seguridad y calidad que deben darse tanto durante el diseño y montaje como durante la ejecución y explotación. En el supuesto de que en alguna de las etapas del proyecto surgiera algún percance no determinado en este documento se deberá contactar con el profesional al cargo para obtener la solución adecuada al mismo.

3. Normativa vigente

Los dispositivos utilizando en el presente proyecto se alimentan directamente de la red eléctrica de 220 V 50 Hz, por lo que todos ellos deben regirse por las normas presentes en el Reglamento Electrotécnico de Baja Tensión (REBT). Por otra parte, será necesaria la aplicación de las normativas UNE y DIN.

Las normas a las que se debe plegar el proyecto son las siguientes

- **ITC-BT-18:** instalaciones de puesta a tierra.
- **ITC-BT-19:** instalaciones interiores o receptoras. Prescripciones generales.
- **ITC-BT-20:** instalaciones interiores o receptoras. Sistemas de instalación.
- **ITC-BT-22:** instalaciones interiores o receptoras. Protección contra sobreintensidades
- **ITC-BT-23:** instalaciones interiores o receptoras. Protección contra sobretensiones.
- **ITC-BT-24:** instalaciones interiores o receptoras. Protección contra contactos directos e indirectos.
- **ITC-BT-51:** instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad.
- **ITC-BT-43:** instalación de receptores. Prescripciones generales.
- **ITC-BT-47:** instalación de receptores. Motores.
- **UNE 20 514 1M:** reglas de seguridad para aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uno general análogo conectado a una red de energía.

4. Condiciones técnicas

Este apartado de Condiciones Técnicas comprende el conjunto de características que deberán cumplir los materiales empleados en el dispositivo, las condiciones de los materiales, las características de cada uno de los productos suministrados.

4.1. Cableado eléctrico

Se limitará la posibilidad de accidentes eléctricos, ya sean sobrecargas, cortocircuitos o contactos directos/ indirectos que puedan resultar en un deterioro del sistema o en lesiones humanas mediante el uso de cableado envuelto en material aislante como PVC. Se han empleado conductores de cobre con 1,5mm² de sección y aislamiento PVC y del color correspondiente para cada uso. Para la transmisión de datos entre dispositivos se emplearán cables de Ethernet correctamente protegidos para esfuerzos mecánicos e interferencias y para la alimentación y comunicación con el variador se emplearán las mangueras suministradas por el distribuidor, las cuales se encuentran apantalladas, aisladas y protegidas de esfuerzos mecánicos. En la entrada a la alimentación se dispondrá de un apantallamiento adicional para disminuir al mínimo las interferencias. Además, todos estos elementos deben cumplir el RD 2267/2004: Reglamento de seguridad contra incendios en los establecimientos industriales.

4.2. Interruptores y pulsadores

Todos los interruptores y pulsadores que se utilicen deberán cumplir con las normas UNE 20 353 y UNE 20 378. Los pulsadores e interruptores se instalarán de forma que no se permita su ejecución accidental, ya sea por medios humanos o ambientales

4.3. Pulsadores de emergencia y demás elementos de seguridad

Para los pulsadores de emergencia se empleará tipo seta de color rojo y se colocarán en un lugar visible y de fácil acceso desde el lugar de trabajo del operario. Tanto los pulsadores como los sensores de seguridad se utilizarán contactos normalmente cerrados. Además, todos los pulsadores de emergencia deberán cumplir con la norma EN 408 para la seguridad en maquinaria.

4.4. Fuente de alimentación

Se empleará una fuente de alimentación de 24Vdc para cada variador de frecuencia. Esta entrada actúa como elemento de seguridad y debe suministrar 2 A como mínimo. Este dispositivo deberá ser adecuado a un ambiente industrial, disponer de protecciones de sobrecarga, sobre corriente y temperatura. Además, dispondrá de indicadores que notifiquen su estado.

4.5. Ordenador personal

Se requerirá de una computadora desde la que ejecutar el software de programación para el PLC. Este debe contar de monitor, teclado, unidad central, teclado y puerto de conexión de ethernet. Se deberán tomar precauciones frente a la situación ambiental de temperatura y humedad que puedan producirse en el entorno de trabajo, respondiendo a la norma DIN 40050IP20. La temperatura ambiental de servicio oscilará entre los 0°C y los 60°C. Para ejecutar los programas requeridos será necesario contar con las siguientes características.

- Procesador: Core TM i5-6440EQ o similar.
- Velocidad mínima del procesador 3.4 GHz.
- RAM: 16 GB como mínimo, siendo recomendable 32 GB.
- Disco Duro: 50GB de SSD como mínimo.
- Sistema operativo, Windows 7 o 10(recomendable).
- Resolución de pantalla de 1920x1080.

4.6. Software

Para la realización, compilación y transferencia del programa de control al PLC se utilizará el software TIA PORTAL V16.0 de SIEMENS. El SCADA se ha realizado en la extensión WinCC Advanced del mismo distribuidor y el variador se ha parametrizado con la función extendida MOTION CONTROL. En caso de ser necesario transferir el programa a un PLC de otra marca será necesario emplear el software correspondiente. Será obligatorio realizar los cambios necesarios al mismo ya que no se garantiza su correcto funcionamiento en otro entorno de programación.

4.7. PLC

El autómata programable estará alimentado directamente a la red de 220 V, por lo que deberá de estar correctamente protegido y aislado. Todas las entradas y salidas del autómata deberán estar correctamente aisladas entre sí. Será obligatorio seguir las condiciones de montaje y protección especificadas en el manual suministrado con el dispositivo.

4.8. Variador de frecuencia

Las conexiones al variador se harán a través de las borneras suministradas y se apantallará y conectará a tierra todos los puntos metálicos. Se interconectarán también las masas de todos los dispositivos. Al igual que en el PLC, se seguirán las condiciones de instalación y protección estipuladas en el manual de usuario. Se dispondrán protecciones teniendo en cuenta que el motor será alimentado a través del variador, por lo que debe ser correctamente protegido. Finalmente, se protegerán las entradas y salidas del variador que queden al aire.

4.9. Motor

Los accionamientos deben ser instalados de forma que no sea posible que sean causa de un accidente laboral. Deben de ser colocados de forma que no puedan penetrar líquidos en el interior del motor o de los cables de alimentación y deben estar alejados de cualquier sustancia peligrosa. El variador de frecuencia que alimenta al motor debe de estar protegido frente a sobreintensidades y sobrecargas y es necesario contar con que el motor es capaz de suministrar tres veces su par, lo cual afectará a la intensidad admisible por el cable. Se debe evitar en todo lo posible los arranques intempestivos del motor de acuerdo con la norma UNE 20460-4-45. Será obligada también la correcta parametrización de variador y motor de forma que su control sea adecuado y eficiente, así como la inclusión del enconder integrado.

Los motores deben incorporar una placa de características situada en un lugar visible en la que aparecerán de forma indeleble la velocidad de rotación nominal del eje, la potencia nominal junto a la intensidad y tensión nominal y la intensidad máxima en el arranque y el modelo de motor junto al fabricante.

5. Condiciones constructivas

Este capítulo del presente pliego tendrá como finalidad fijar los requisitos de las pruebas, ensayos y montaje necesarios para asegurar el funcionamiento del autómeta

5.1. Montaje eléctrico

Se protegerán los dispositivos mediante el empleo de interruptores automáticos y se unirán tanto las masas como las tierras de todos los dispositivos. El cableado de los actuadores, sensores y alimentación se introducirá en canaletas para impedir su manipulación. Los cables de comunicaciones se instalarán de forma independiente y se aislarán de posibles interferencias. Finalmente, se seguirán las recomendaciones del fabricante para la colocación de los motores y variadores. Se protegerá el elemento de control introduciéndolo en un armario cerrado al cual solo dispondrá de acceso personal autorizado.

5.2. Pruebas de funcionamiento

Antes de la puesta en marcha se realizarán los ensayos expuestos en la ITC-BT-05 del REBT: verificaciones e inspecciones para asegurar que las conexiones y protecciones actúan de forma correcta. Seguidamente, se explicarán estos ensayos.

5.3. Comprobación visual del conexionado

Se verificará que todas las conexiones se han realizado de forma correcta y que los cables están bien sujetos y en su lugar correspondiente. Por último, se comprobará que no hay contactos entre estos

5.4. Pruebas en tensión

Una vez se está seguro de que el conexionado es correcto, se dará tensión para asegurarse que todos los componentes de seguridad y protección actúan de forma adecuada. Seguidamente, será necesario asegurarse de que todos los dispositivos están en funcionamiento y que no se encuentra ningún comportamiento anómalo.

5.5. Prueba final

Finalmente, se parametrizarán los motores y se configurarán los variadores para que se puedan controlar desde el PLC. Desde el autómatas se ejecutarán órdenes de mando manuales para comprobar que la comunicación es correcta. Primero se realizará un Tuning de los motores y seguidamente un JOG. Seguidamente, se introducirá el programa de control y se irá comprobando el funcionamiento de las entradas y salidas. Por último, se probarán los modos de control programados y se asegurará que su funcionamiento es el esperado. En el caso de que todo se ejecute de la forma esperada, las pruebas de ensayo se darán por concluidas y con ello el proceso de montaje y puesta a punto. Al utilizar la tensión de red todas las partes metálicas deben de estar correctamente aisladas y protegidas para evitar contactos indirectos.

5.6. Control de calidad

Los elementos empleados en el proyecto han sido adquiridos de forma legal a un distribuidor oficial por lo que estos deben de haber superado los controles de calidad reglamentarios en el marco de la UE. Por esto, la verificación de la calidad de los mismos queda fuera de las obligaciones del contratista y del ingeniero al cargo del proyecto. Se seguirán las indicaciones del fabricante para disponer de la garantía que este ofrece. Sin embargo, se realizará una comprobación de todos los componentes para asegurarse de que no presenta defectos visibles. Se verificarán y ensayaran los elementos que pueden ocasionar accidentes en caso de estar defectuosos. Una vez se haya realizado

la instalación, se comprobará el cableado y se probará que todos los componentes funcionan de forma correcta.

En caso de disponer los elementos sobre superficie metálica, como puede ser un armario de seguridad, se verificará que no hay contacto en el cableado y la superficie metálica. Además, se realizará un circuito de ventilación natural o forzada si procede.

6. Condiciones de ejecución

Para el correcto montaje del equipo se deberá disponer de las herramientas y materiales necesarios para asegurar la calidad y seguridad de la instalación y es necesario que estas herramientas y los materiales y dispositivos utilizados hayan cumplido con las verificaciones y pruebas realizadas. Al tratarse de un prototipo, se aconseja su revisión completa antes de aplicarse a un proyecto real.

6.1. Compra de material

Para adquirir el material necesario para la realización del proyecto se analizarán los precios, oferta y tiempo de suministro de los proveedores disponibles, siempre buscando ofrecer un precio y tiempo de ejecución competitivo. También se estudiará la disponibilidad de los dispositivos a emplear y de la vida útil de la que dispondrán los mismos en el mercado ya que no es interesante trabajar sobre un equipo obsoleto. También se deberá estudiar la procedencia de estos y será interesante informarse acerca de la calidad de estos. Por último, al ser más económico comprar en grandes lotes se valorará la posibilidad de adquirir más de un equipo con la intención de replicarlo en otras instalaciones.

7. Condiciones económicas

Al tratarse de un Trabajo Fin de Grado no está previsto el pago de ninguna cantidad económica en concepto de honorarios ya que no está prevista su explotación en el mercado. En caso de emplear este trabajo con fines lucrativos se seguirán los siguientes puntos.

7.1. Pagos de los trabajos

A continuación, se fijarán las condiciones a seguir por el projectista, el contratista y el contratante.

- El importe no sufrirá de ninguna clase de sobrecargo si el pago se produce antes de que transcurran cinco días naturales desde el fin definitivo del proyecto.
- El proyecto podrá sufrir un sobrecargo extra del 2% en caso de que el importe se abone entre seis y treinta días tras la recepción final del dicho proyecto.
- En caso de que el abonamiento se produzca entre el día treinta y uno y el sesenta, se aplicará un sobrecargo del 4% del total.
- Si el pago se realiza tras sesenta y un días y hasta los noventa y un días se aplicará un recargo del 6,5% sobre el precio pactado.
- En caso de que la demora estuviese entre los noventa y un y los trescientos días, se producirá un sobrecoste del 30% del total previamente pactado.
- Se admitirá el pago por plazos, siempre y cuando este haya sido pactado con anterioridad, habiendo quedado reflejado el número de plazos y el tiempo que transcurriría entre estos. En caso de producirse demoras se aplicaría el sobrecoste correspondiente siguiendo lo antes dispuesto. En caso de que el contratante optara por pagar a plazos, debe comunicarlo
- En caso de que se produzca el impago dentro del tiempo estipulado el projectista se guarda el derecho de presentar una querrela ante los Tribunales.

8. Condiciones legales

En este apartado se definirán las obligaciones para con la ley que son contraídas por todas las partes del presente proyecto. El incumplimiento de estas puede devenir en consecuencias legales.

8.1. Contrato

Es obligatorio que en el documento contractual quede expresado el precio y costes del proyecto de forma clara. También es necesario que quede constancia de los recargos que pueden darse tal y como se ha establecido en el apartado anterior del presente pliego de condiciones.

El contrato debe de realizarse por escrito y debe también cumplir con todos los requerimientos que exige la ley.

8.2. Arbitraje y jurisdicción

La responsabilidad del arbitraje de los intereses de ambas partes recaerá sobre la dirección facultativa encargada al proyecto. En caso de que fuese imposible llegar a una solución satisfactoria, se elegirán técnicos que, representando a cada una de las partes, negociarán un acuerdo. Si esto también fracasa se deberá recurrir a los tribunales.

8.3. Impuestos

Se exigirá al cliente que se encuentra informado acerca del pago de impuestos, contribuciones y demás obligaciones legales para llevar a cabo sus actividades empresariales. Se aplicará el impuesto sobre el valor añadido (I.VA) al precio resultante para comercializar este diseño. Actualmente este se encuentra en un 21%. Sin embargo, en caso de que este valor se modificase sería necesario aplicar el nuevo valor.

8.4. Rescisión del contrato

El contratista podrá rescindir el contrato si se produce un retraso considerable en la entrega del producto. Pasado el tiempo acordado, el contratista dispondrá del derecho a decidir debido a la tardanza en recibir el encargo.

8.5. Condiciones facultativas

La finalidad de este proyecto es automatizar el almacenamiento de productos de una instalación logística genérica, habiéndose presentado de forma ejemplificativa el caso de un almacén de productos alimenticios.

La función programa ha sido probada sin ser instalada en ejes reales. En caso de que estas funciones se implementasen en un caso real, sería necesario revisar tanto el código como el par entregable por el motor y sería necesario redimensionar los cableados y las conexiones de datos. También sería obligatorio rediseñar todas las protecciones para adecuar su instalación a un entorno industrial. En caso contrario, no se garantiza la correcta ejecución de este.

8.6. Derechos y deberes del contratista

El contratista debe de disponer de conocimientos suficientes como para explotar el proyecto de forma segura y correcta. Debe de conocer las limitaciones y especificaciones de los equipos que ha adquirido y entender que su margen de maniobra está limitado a un rango. Quedará bajo la responsabilidad del estado de las máquinas, siendo encargado de las verificaciones y mantenimientos necesarios. También es necesario que disponga de técnicos habilitados legalmente para la realización de estas tareas. Serán de obligatorio cumplimiento todas las legislaciones que afecten a la instalación, como puede ser el REBT. Se aconseja por tanto disponer de un ejemplar actualizado o de conocimientos de este. También deberá de obtener los permisos necesarios que le permitan explotar de forma legal la instalación y debe cumplir con los requisitos y

condiciones de funcionamiento especificadas en este proyecto. En caso contrario, el proyectista no podrá garantizar que la actuación del proyecto sea correcta. Finalmente, cualquier causa debidamente justificada, ya sea un fallo, el retraso de la entrega o cualquier otro impedimento ajeno al contratista, deberá ser aceptado por el contratante. Cualquier retraso no justificado será abonado según se ha especificado en este documento.

Ambas partes se comprometen a facilitar el proyecto de forma que la programación, instalación y puesta en marcha de este se realice de forma precisa y rápida. El cumplimiento de los objetivos mínimos será de obligatorio cumplimiento.

Será obligatorio consultar con el diseñador al cargo del proyecto cualquier mejora o variación que altere el funcionamiento de la instalación. El estado base del proyecto, así como sus componentes y especificaciones serán reflejados en la memoria de este proyecto. Estos deben de ser tenidos en cuenta.

Se dará luz verde a la realización del proyecto cuando las partes implicadas se hayan comprometido, mediante firma válida, a cumplir las cláusulas del contrato. El trabajo del contratista se considera acabado tras verificar la primera puesta en marcha de este.

Cualquier negligencia que no provenga directamente del proyectista carecerá de responsabilidad para el mismo ya que se considera que los requerimientos del proyecto se han especificado de forma clara.

El contratista se encuentra en derecho de disponer por escrito los documentos del proyecto que le sean necesarios, ya sea la memoria de este, un juego de planos o cualquier otro documento relevante

PRESUPUESTO

ÍNDICE PRESUPUESTO

1. Objeto y alcance del presupuesto.....	106
2. Material.....	107
3. Software.....	108
4. Mano de obra.....	109
5. Coste total.....	110

1. Objeto y alcance del presupuesto

La finalidad de este documento es hacer un resumen detallado de los costes y gastos de la materia prima, maquinaria y mano de obra que se va a utilizar.

Se tendrá también en cuenta los costos de las piezas adquiridas de otras empresas y, por último, se añadirán costes como IVA y otros impuestos a los que el producto está sujeto, para llegar así a un presupuesto lo más aproximado posible.

Antes de comenzar con el desarrollo del presupuesto, hay que destacar la amortización de la maquinaria, de tal forma que a lo largo del tiempo se compensará la gran inversión inicial que esta conlleva.

Para el desarrollo de este documento, se han tenido en cuenta presupuestos de otras empresas para la compra de materiales y componentes, buscando siempre la mejor relación calidad/precio.

A continuación, se desglosará el presupuesto del proyecto en costes del material, uso del software y horas trabajadas. Se incluirán también los gastos generales, impuestos y porcentaje de beneficios pactado por el proyectista. El importe total resultante correspondería al precio final del proyecto.

2. Material

Los costes por material constan de los precios por la compra del hardware del producto, PLC, variador de frecuencia, motor, ordenador y los precios relacionados con el material eléctrico y cableado. Estos se refieren a los cables de alimentación y comunicación, herramientas utilizadas y demás material eléctrico.

Como ordenador, se necesitará utilizar una computadora con unas características mínimas que pueda soportar los programas de edición de código necesarios. Para este caso, se aplica un porcentaje de su valor proporcional al número de horas útiles que se han empleado en el proyecto

El desglose del coste por parte del material queda reflejado en la siguiente tabla:

Tabla 1 Desglose del coste por material

Tabla 1	Elemento	Descripción	Cantidad	Precio Unitario	Precio Total
M A T E R I A L	PLC	Automata Siemens S7-1215AC	1	497,60 €	497,60 €
	Variador de frecuencia	Variador de frecuencia SIMATIC S210 con motor PMSM	2	564,00 €	1.128,00 €
	Fuente 24V	Fuentes DC de 24V para la conexión X124 del variador	2	30,00 €	60,00 €
	Material eléctrico	Cables, regletas, interruptores, contactores y herramientas	1	300,00 €	300,00 €
	Cables ethernet	Cables Profinet SIEMENS para uso industrial	1	60,00€	60,00 €
	Pc	Ordenador 8GB RAM 1TB SSD i7-7700 con monitor y teclado	1	1.200,00 €	49,22 €
				Total	2.094,82 €

3. Software

Los costes por software consisten en la suma de todas las aplicaciones informáticas usadas. Para ello, será necesario contar con distintos programas que puedan editar código, diseñar pantallas como el ejemplo del SCADA, acceder al variador de frecuencia y la redacción del presente documento.

A los precios totales de las licencias se aplicará el porcentaje de uso que se les ha dado contando con que se querrá actualizar el software siempre a la versión más avanzada. Se ha considerado que se renovarían las licencias de SIEMENS cada dos años en el momento de la actualización, mientras que el pack de Office tiene un precio anual y el sistema operativo se adquiere una sola vez.

De esta forma, el desglose del coste según el software es el siguiente:

Tabla 2 Desglose del coste por Software

Tabla 2	Elemento	Descripción	Cantidad	Precio Unitario	Precio Total
S O F T W A R E	Licencia TIA PORTAL Proffesional	Licencia básica del software de programación para PLC, SCADA y motor	1	1.512,00 €	1.512,00 €
	WinCC RT Advanced	Extensión de diseño para SCADAs	1	1.645,00 €	1.645,00 €
	startdrive	Extensión de parametrización de Motion Control	1	443,00 €	443,00 €
	Oficce 365	Pack de programas ofimáticos de Windows	1	70,00 €	70,00 €
	Windows 10	Sistema operativo PC	1	279,00 €	279,00 €
				Total	3.949,00 €

4. Mano de obra

En este caso, se remunerarán las horas de trabajo del proyectista según el salario regulado de un ingeniero eléctrico en calidad de Ingeniero Junior, el cual se puede consultar en la Tabla Salarial emitida por el BOE. El precio por hora trabajada queda reflejado en la siguiente tabla:

Tabla 3 Tabla salario graduado Ingeniería eléctrica

Salario Ingeniero Técnico Industrial	
Año	26.197,10 €
Mensual	2.181,59 €
Diario	109,00 €
Por hora	13,63 €

Según estos datos, se han aplicado las horas correspondientes que se han realizado durante las distintas labores de este proyecto. Sobre estas horas trabajadas, se ha añadido un porcentaje del 7% referido a los gastos asociados que hayan surgido a lo largo de la realización (facturas, gestiones, transportes). Finalmente, podrían pactarse trabajos adicionales tales como labores de mantenimiento, de formación o distintos tipos de seguros. Estos datos quedan reflejados en la siguiente tabla:

Tabla 4 Desglose del coste del diseño del proceso

Tabla 3	Elemento	Descripción	Horas	Precio Unitario	Precio Total
D I S E Ñ O D E L P R O C E S O	Estudio del proyecto	Planteamiento de la solución y estudio de equipos y posibilidades	20	13,63 €	272,50 €
	Parametrización y montaje	Configuración de los equipos, cableado y comprobaciones para la puesta en marcha	35	13,63 €	476,88 €
	Programación de la función	Desarrollo del código ST encargado del control del proceso	120	13,63 €	1.635,00 €
	Diseño del SCADA	Estructuración de la interfaz Humano-Maquina integrada en el proyecto	65	13,63 €	885,63 €
	Pruebas y correcciones	Realización de ensayos al producto para la certificación de su correcto funcionamiento	20	13,63 €	272,50 €
	Redacción de la memoria	Escritura del documento que acompaña al proyecto	55	13,63 €	749,38 €
Coste	Gasto general		315	13,63 €	4.291,88 €
Gastos	7% del total				300,43 €
				Total	4.592,31 €

5. Coste total

Finalmente, y según los datos de las tablas anteriores, se ha calculado el importe final del proyecto, añadiendo al mismo los porcentajes de beneficio y aplicando el impuesto sobre el valor añadido (I.V.A)

De esta forma, el importa final y desglose quedan reflejados en la tabla siguiente:

Tabla 5 Tabla resumen del coste total

Tabla 4	Grupo	Elemento	Descripción	Precio
R E S U M E N	Compra de material	Material eléctrico	Herramientas, cableado, contactores, etc	300
		Hardware	CPU, PLC, Variador	1794,82
	Software	Programación y ofimática	Programas utilizados en la realización del proyecto	323,94 €
	Servicios	Trabajo del ingeniero	Tareas realizadas por el proyectista	4.291,88 €
	Gastos	Gastos generales	Desplazamientos, gestiones, alquileres	300,43 €
Coste				7.011,07 €
Beneficio	Beneficio Industrial		20%	1.402,21 €
Impuestos	I.V.A		21%	1.472,32 €
			Total	9.885,61 €

Asciende el presupuesto por contrata a la expresada cantidad de: NUEVE MIL OCHOCIENTOS OCHENTA Y CINCO EUROS CON SESENTA Y UN CÉNTIMO.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Junta de Andalucía. (25 de 28 de 2021). *IEDA*. Obtenido de Pilas: http://agrega.juntadeandalucia.es/repositorio/02122016/a5/es-an_2016120212_9131705/32_pilas.html
- Angel Sapena Baño, R. P. (2020). *Apuntes asignatura Protección y Regulación de Máquinas Eléctricas*. Valencia: Universidad Politécnica de Valencia.
- Autracen. (2 de 06 de 2021). *Autracen*. Obtenido de Walking through industry 4.0: <https://www.autracen.com/descubre-la-estructura-interna-plc>
- Bañó, Á. S. (2018). *User Management*. Valencia: UPV.
- Castillo, J. C. (20 de 06 de 2021). *Youtube*. Obtenido de Juan Carlos Martín Castillo: <https://www.youtube.com/watch?v=wejEuHudw-8>
- Cisneros, J. (5 de 06 de 2021). *Datadec*. Obtenido de <https://www.datadec.es/blog/que-son-almacenes-inteligentes>
- Cisneros, J. (05 de 06 de 2021). *Datadec*. Obtenido de Qué son los almacenes inteligentes: <https://www.datadec.es/blog/que-son-almacenes-inteligentes>
- EDS/ROBOTICS. (15 de 06 de 2021). *EDS/ROBOTICS*. Obtenido de ¿Qué es la automatización industrial?: <https://www.edsrobotics.com/blog/que-es-la-automatizacion-industrial/>
- Fort, V. (2020). *Apuntes de prácticas de la asignatura Control de máquinas*. Valencia: Universidad Politécnica de Valencia.
- JUNGHEINRICH. (27 de 05 de 2021). *JUNGHEINRICH*. Obtenido de Transelevadores piezas: <https://www.jungheinrich.es/>
- Nanclares, E. B. (20 de 06 de 2021). *industriaconectada40.gob*. Obtenido de industriaconectada40.gob: <https://www.industriaconectada40.gob.es/SiteCollectionDocuments/Grup>

- os-Trabajo/GT-Estandarizacion/Estandarizacion-Manuf4-0MOND-Industria-Conectada.pdf
- Panadero, R. P. (2020). *Apuntes asignatura Control de Máquinas*. Valencia: UPV.
 - Pérez, E. J. (2018). *Máquinas eléctricas dinámicas*. Valencia: Escritos técnicos.
 - *plc-hmi-scadas*. (13 de 06 de 2021). Obtenido de Recetas en TIA WinCC Runtime Professional: <https://plc-hmi-scadas.com/039.php>
 - Ricard0. (13 de 06 de 2021). *infoPLC*. Obtenido de Configuración Básica de Animaciones en WINCC BASIC: <https://www.infopl.net/foro/forum/hmi/scada/16030-configuración-básica-de-animaciones-en-wincc-basic>
 - SIEMENS. (2018). *Manual WinCC Advanced*. Múnich: SIEMENS.
 - SIEMENS. (2019). *Configuration of technology objects with SIMATIC S7-1500 and SINAMICS S210 in the TIA Portal*. Munich: Siemens.
 - SIEMENS. (2019). *Manual S7-1200*. Múnich: SIEMENS.
 - SIEMENS. (2019). *Manual SINAMICS S210*. Múnich: SIEMENS.
 - SIEMENS. (2019). *STEP 7 S7-1200 Motion Control V6.0...V7.0 en TIA Portal V16* . Múnich: SIEMENS.
 - SIEMENS. (15 de 05 de 2021). *Industry Online Support Internacional*. Obtenido de ¿Cómo se puede realizar un cambio de idioma para un panel de operador SIMATIC HMI utilizando el WinCC (TIA Portal)?: <https://support.industry.siemens.com/cs/document/109478713/>

ANEXOS

ÍNDICE ANEXOS

1. Anexo 1 Manual de programa	
1.1. Introducción.....	118
1.2. Menú.....	118
1.3. Modo automático.....	120
1.4. Modo manual.....	122
1.5. Acusar errores.....	124
1.6. Configuración.....	125
1.7. Datos.....	126
2. Anexo 2 Manual uso TIA PORTAL	
2.1. Introducción.....	127
2.2. Generar archivos de programa.....	127
2.3. Configurar equipos.....	129
2.4. Añadir bloques.....	130
2.5. Herramientas de edición.....	131
2.6. Cargar y monitorizar programa.....	132
3. Anexo 3 Manual uso WinCC RT ADVANCED	
3.1. Introducción.....	133
3.2. Generar pantallas y plantillas.....	134
3.3. Recetas.....	134
3.4. Usuarios.....	135
3.5. Alarmas.....	136
3.6. Idiomas.....	137
4. Anexo4 Listado del programa.....	138
5. Anexo 5 Hoja de datos PLC.....	178
6. Anexo 6 Hoja de datos S210.....	185
7. Anexo 7 Hoja de datos motor.....	187

Anexo I Manual de usuario

1. Introducción

En este anexo se mostrarán como accionar y controlar el programa desde el SCADA, se explicarán los controles de cada uno de los modos para clarificar su modo de funcionamiento. Se analizarán las funcionalidades de cada pantalla por separado y con indicadores numéricos se describirá el funcionamiento de cada botón o display presente.

2. Menú

Desde la pantalla del Menú se realiza el login y se podrá acceder al resto de interfaces. Una vez registrado es importante comprobar que el grupo de permisos aparece en la esquina superior derecha, en caso contrario, no se ha introducido correctamente al usuario.

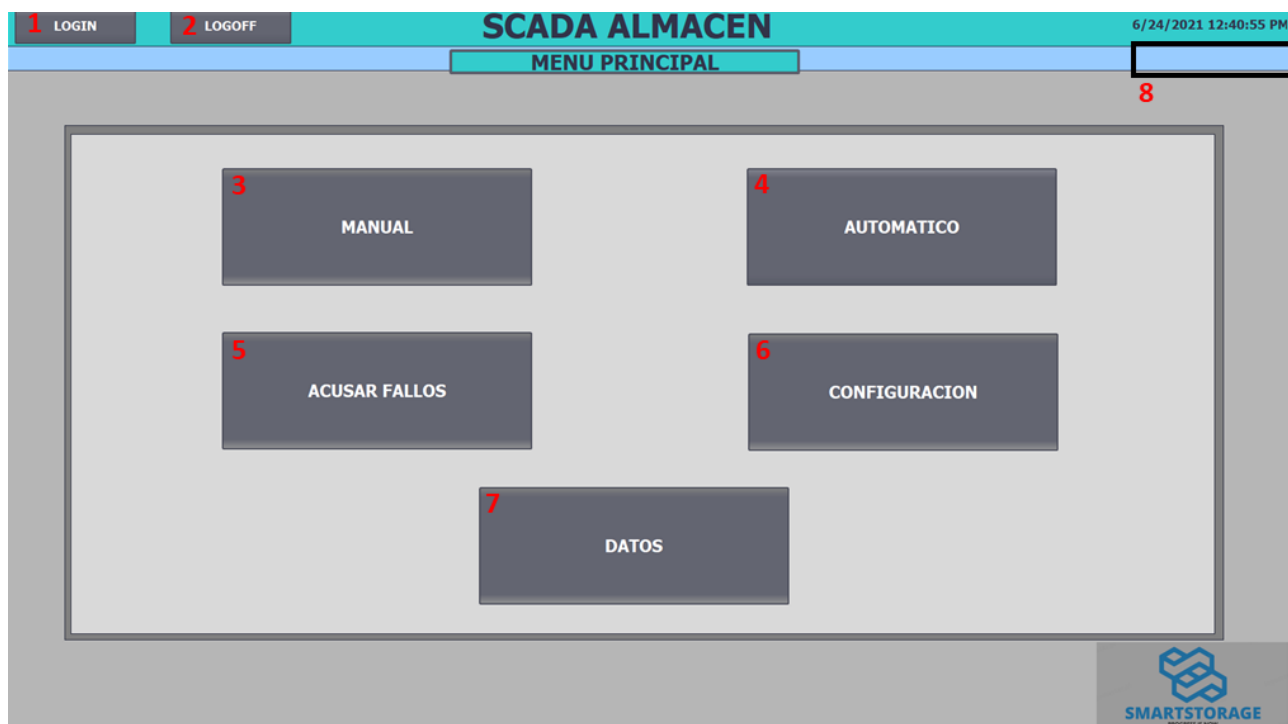


Figura 53 Menú SCADA

1. **Login** Aparece el desplegable para introducir nombre y contraseña

2. **Logoff** El programa deja de recordar el usuario actual y desactiva
3. los permisos introducidos
4. **Manual** Cambia la pantalla al modo manual
5. **Automático** Cambia la pantalla al modo automático
6. **Acusar Fallos** Cambia la pantalla al modo acusar fallos
7. **Configuración** Cambia la pantalla a la interfaz de comunicación
8. **Datos** Cambia la pantalla al panel de información
9. **Display** donde aparece el usuario dado de alta

Tan solo se puede acceder a las pantallas incluidas en el permiso del usuario

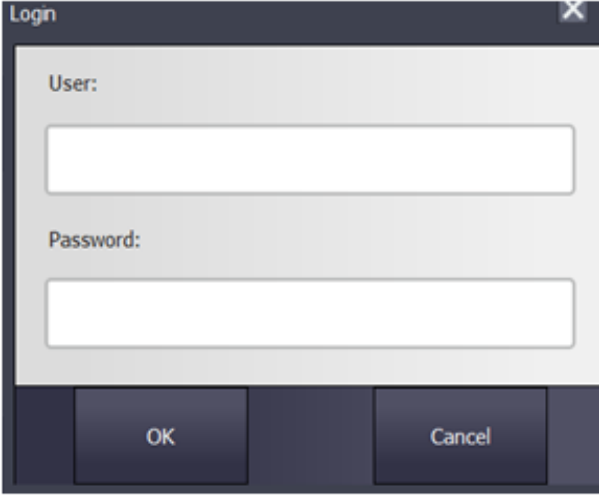


Figura 54 Diálogo de Login

NAVEGACIÓN

En la barra de navegación es posible intercambiar la pantalla con cualquier otra de las disponibles. El proceso debe de estar detenido sin posibilidad de arranque para poder cambiar entre pantallas. En caso opuesto no se realiza el cambio.

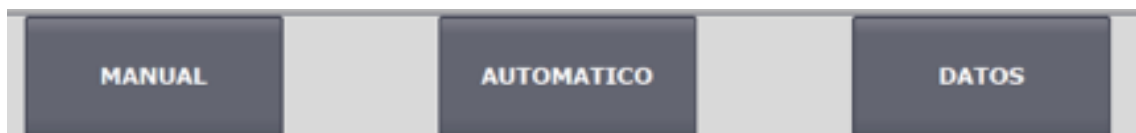


Figura 55 Barra de pantallas

3. Modo automático

Dentro del modo automático se puede accionar el proceso principal. Es posible cambiar entre modo carga y descarga, pero siempre se ha de desactivar un modo para poder actuar en el otro. Dentro de ambos modos se dispone de botones para activar los motores, iniciar el proceso y resetear las memorias, también es posible seleccionar el pedido a realizar y representa el almacén de forma gráfica

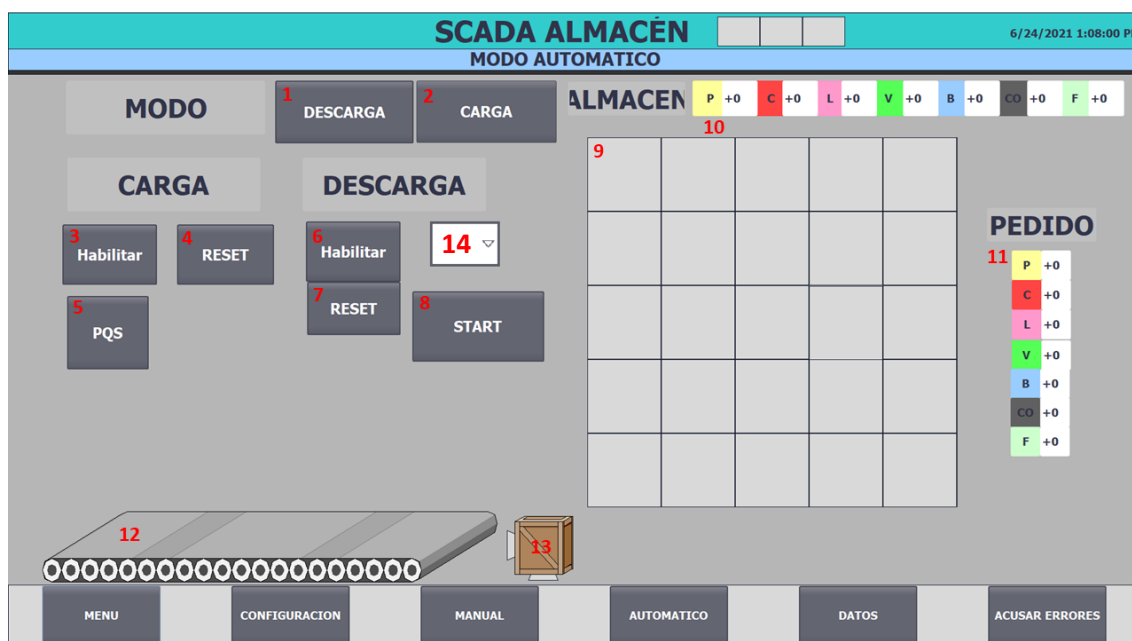


Figura 56 Pantalla automático

1. Selecciona el **modo descarga** e ilumina el letrero
2. Selecciona el **modo carga** e ilumina el letrero
3. **Enciende los motores** en el modo carga
4. **Resetea las posiciones ocupadas** en la memoria
5. **Resetea el contaje de productos guardados**
6. **Enciende los motores** en el modo descarga
7. **Resetea las posiciones de descarga**
8. **Inicializa** el proceso
9. Representación gráfica de las posiciones del almacén
10. Lista de productos almacenados

11. Lista de pedido actual de descarga
12. Representación de la cinta, si gira hacia la izquierda significa que las cajas salen del almacén, si gira hacia la derecha significa que las cajas entran al almacén
13. Representación gráfica de la posición actual del carril del almacén, una vez llega a su destino, vierte el producto en la posición del almacén.
14. El desplegable permite elegir la receta a utilizar

4. Modo manual

Una vez se accede al modo manual se puede seleccionar entre el modo Jog, Entrega y Retorno o la puesta a cero. El modo Jog permite un giro lento a izquierdas o derechas para ambos motores. La entrega lleva el accionamiento hasta la posición indicada en los display al activar Start mientras que el modo retorno lleva los ejes hasta la posición 0.0 absoluta.

En el caso de la puesta a 0, se pueden activar y desactivar los motores y llevar los motores manualmente a la posición que se desee. Con el botón Home se definirá la posición actual del eje como el punto de referencia.

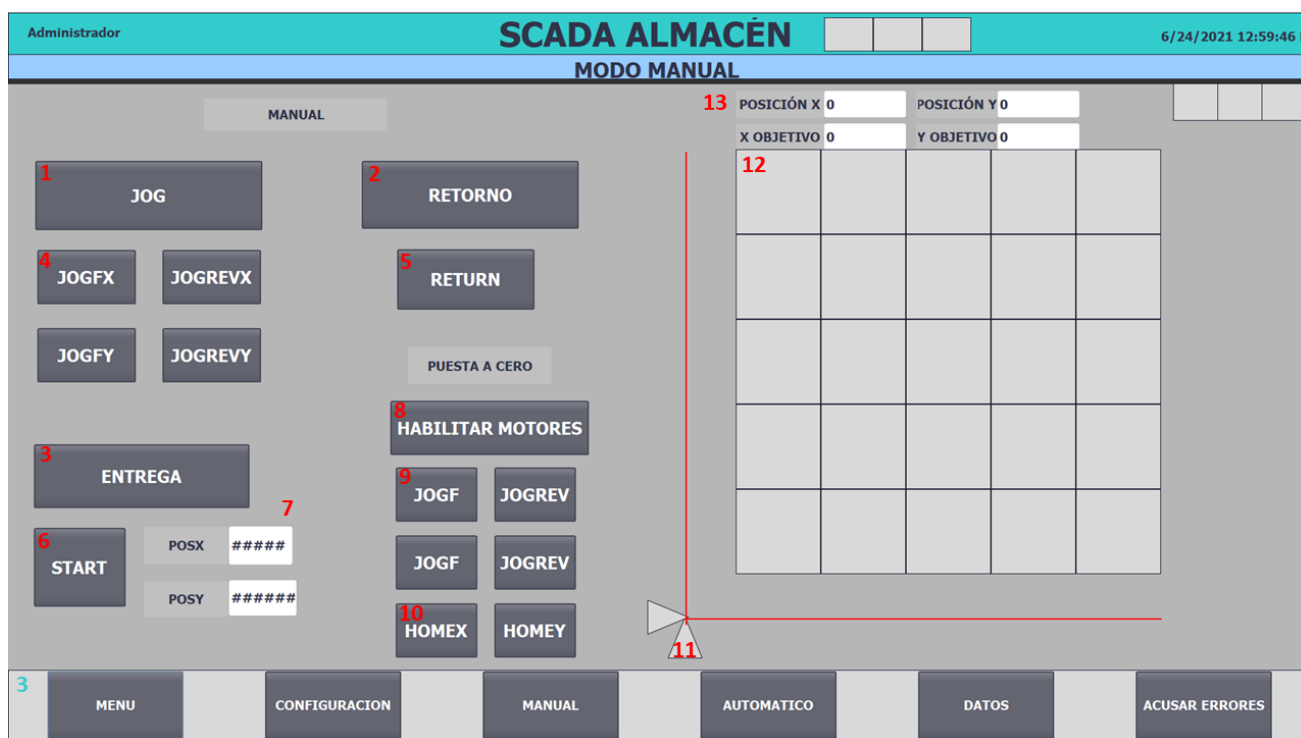


Figura 57 Pantalla Manual

1. Activa el modo **JOG**
2. Activa el modo **Retorno**
3. Activa el modo **Entrega**
4. **Habilita el JOG** en forward/reverse para ambos ejes tal y como indica cada uno de los botones
5. **Activa el movimiento** de los ejes hasta el 0

6. **Activa el movimiento** hasta la posición indicada en 7
7. Introducir aquí la posición a la que se quiere ir
8. **Arrancar motores** en el modo puesta a 0
9. **JOG** en forward/reverse en ambos ejes
10. Definir como punto de origen la localización en X y en Y
11. Representación gráfica de la posición de los ejes
12. Representación gráfica del almacén
13. Datos numéricos de posición actual y posición objetivo

5. Acusar errores

En la pantalla acusar errores es posible corregir la información almacenada en la memoria del PLC. En caso de encontrarse un error en el almacenado, sería posible corregir esos datos posición a posición. También se permite vaciar toda la memoria. Este proceso carece de realimentación real por lo que su manipulación queda bajo la responsabilidad del usuario



Figura 58 Pantalla Acusar Errores

1. **Selector** del producto
2. **Posición X** que se desea corregir
3. **Posición Y** que se desea corregir
4. Hace efectiva la corrección
5. **Vacía el almacén** completamente
6. Representación gráfica del almacén
7. Cursores que señalan la posición seleccionada.

6. Configuración

El modo configuración solo es accesible para el administrador. Desde este es posible editar, cargar, descargar y visualizar los pedidos que pueden realizarse desde la pantalla de recetas, también es posible introducir nuevos usuarios con contraseñas y equipos editables e intercambiar idioma entre los 5 disponibles.

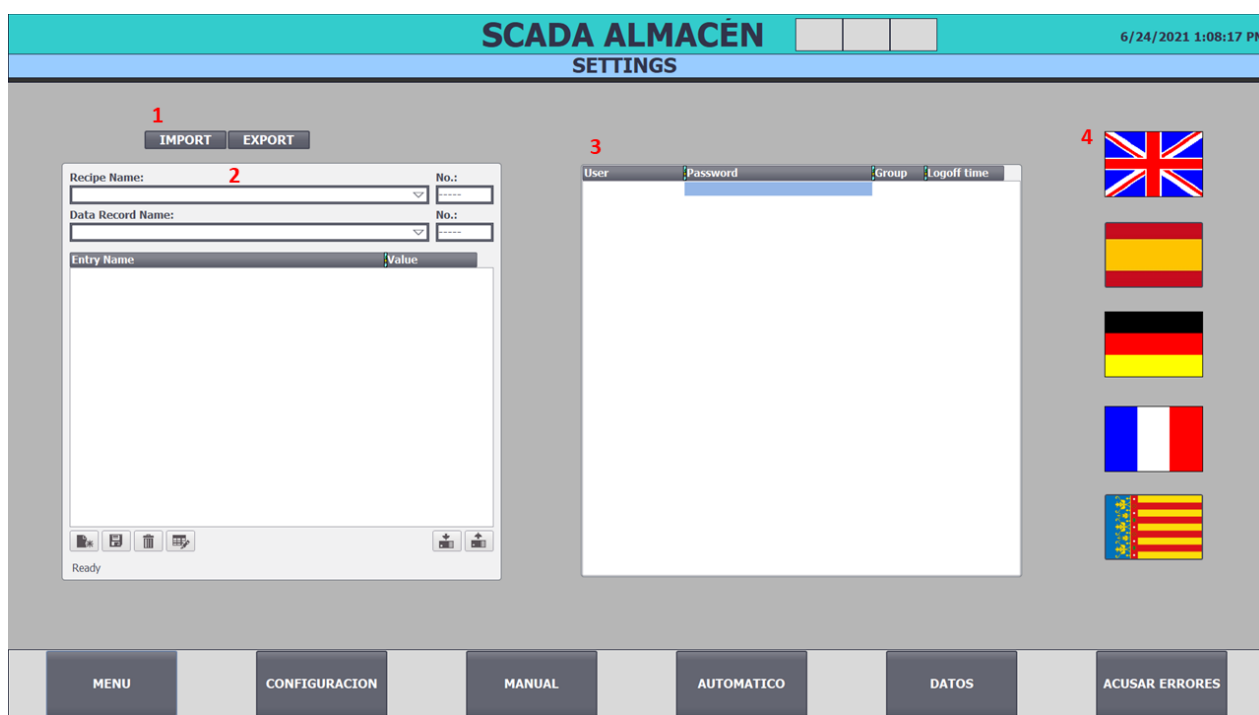


Figura 59 Pantalla Configuración

1. Pulsando el botón importar se carga los datos de un Excel previamente generado sobre la memoria del PLC. El botón exportar genera un Excel con los datos de todas las recetas, las preprogramadas y las creadas durante el runtime
2. El menú de recetas permite seleccionar el pedido exacto entre las categorías y subcategorías. En este display pueden visualizarse las cantidades de todos los productos, también es posible editar datos, guardar los cambios, crear nuevas recetas y cargar desde el PLC o descargar al PLC los datos
3. El desplegable de usuarios permite crear nuevos miembros, asignándoles contraseña, permisos y tiempo que podrán estar activos
4. Clicando sobre cada bandera se cambiará el idioma del SCADA, esto afectará a todos los cuadros de texto disponibles.

7. Datos

Sobre la pantalla de datos se puede visualizar un gran número de variables en formato numérico. Sobre la misma no se permite realizar ninguna acción, solo permite la visualización.

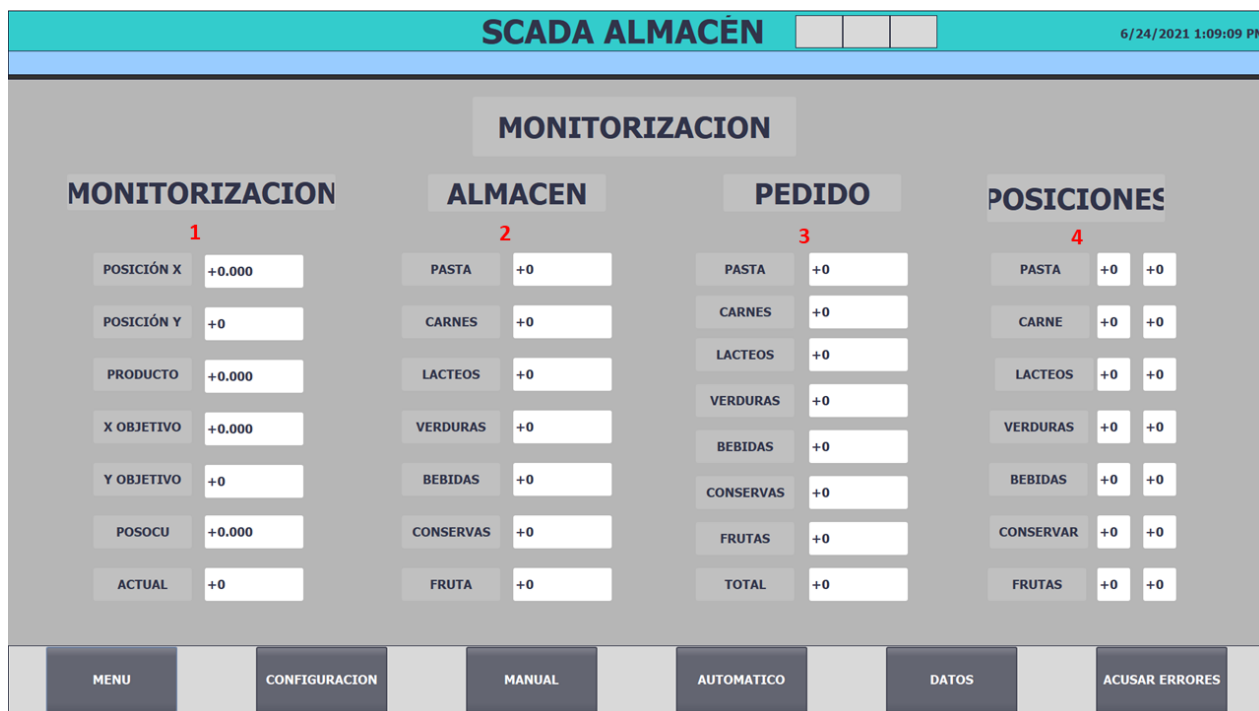


Figura 60 Pantalla Monitorización

1. Presenta datos acerca de la monitorización del proceso, las posiciones de los ejes, el producto que se está transportando, la posición a la que se dirige el automatismo, el número de posiciones ocupadas y la posición a la que se va.
2. En esta columna se visualizan todos los productos almacenados ordenados por tipos
3. En el apartado de PEDIDO se puede consultar la demanda del modo seleccionado
4. En POSICIONES se puede ver a que posición debería ir el control para traer cada uno de los productos almacenados. En caso de no aparecer un valor, significa que no hay disponibilidad del producto.

Anexo II Manual TIA PORTAL

1. Introducción

El software utilizado para la programación del autómata es TIA PORTAL V16, de la marca Siemens, ya que los equipos utilizados pertenecen a dicha casa. El programa permite seleccionar entre una amplia gama de equipos de Siemens y también elegir distintos lenguajes, además, incluye herramientas para diseñar el SCADA. Para poder crear el programa se debe partir de una base. Los pasos son los que se muestran en los siguientes apartados:

2. Generar archivo de programa

En primer lugar, se selecciona “Crear un nuevo proyecto” y se da nombre y dirección de guardado al mismo

Seguidamente, el diálogo nos permite la posibilidad de agregar dispositivo. En este apartado se elige el PLC correspondiente. Es posible consultar el nombre del equipo en una pestaña situada debajo de los LED a la derecha del producto.

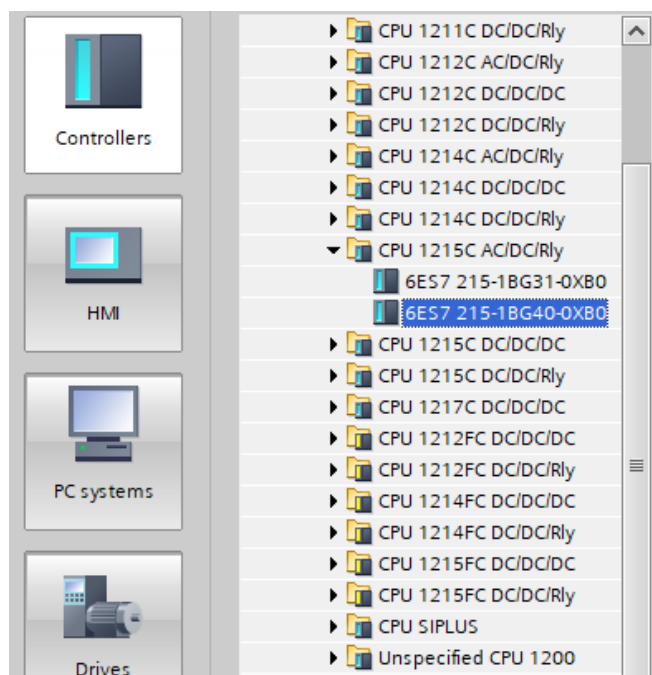


Figura 61 Insertar equipos en programa

A continuación, se utiliza la función “Encontrar equipos”, en la que hay que seleccionar la tarjeta de red del equipo, para localizar los variadores de frecuencia y, al aparecer en dispositivos disponibles de la tarjeta, se cargan en el proyecto.

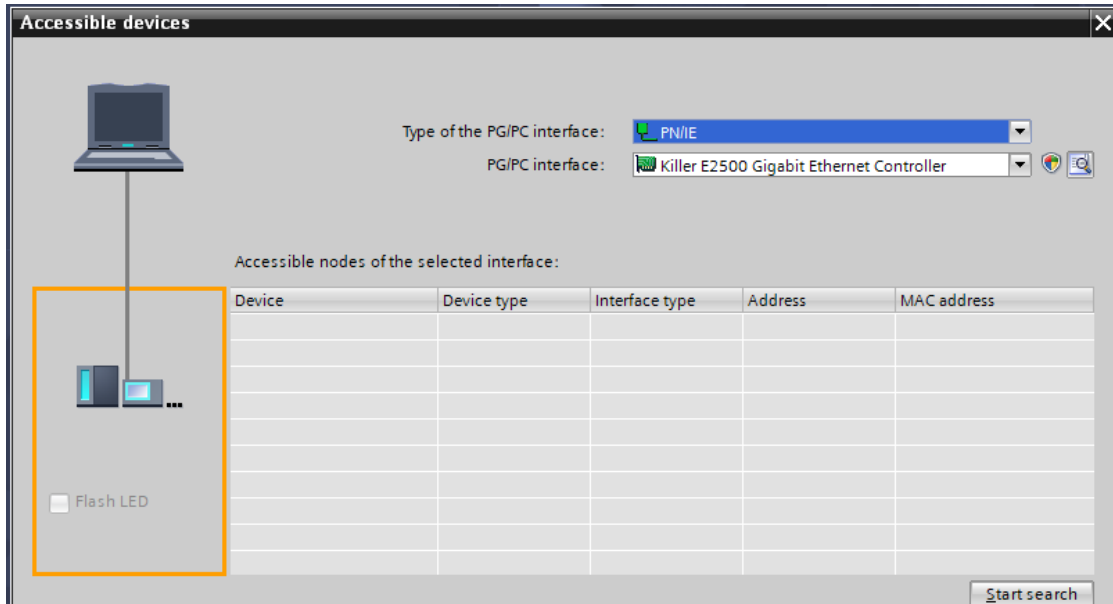


Figura 62 Pantalla equipos accesibles

Cuando se disponga de los equipos que conforman el sistema, se tendrá una interfaz tal que así

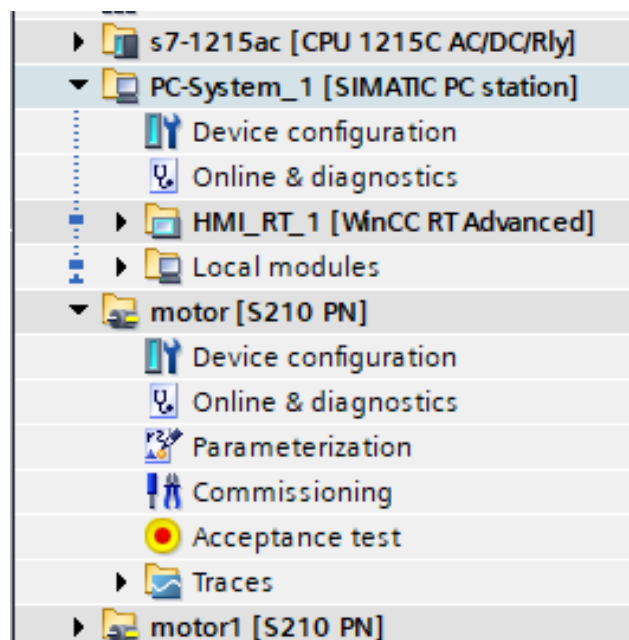


Figura 63 Equipos del proyecto

3. Configurar equipos

Una vez cargados los variadores, se asignan los nombres y direcciones IP de los puertos conectados via PROFINET para poder generar la red que unirá los dispositivos. En el caso de los variadores, es necesario crear nuevos objetos tecnológicos para dirigir las ordenes al equipo real.

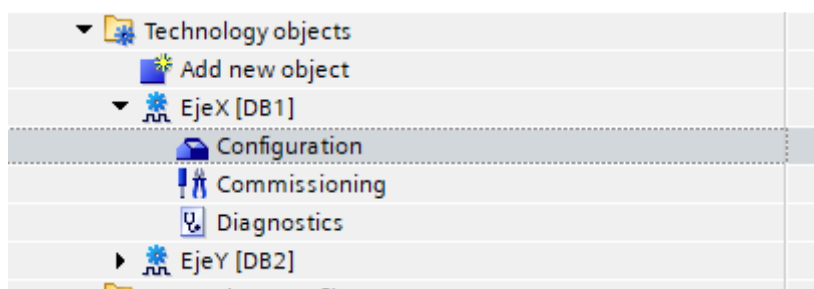


Figura 64 Ejes tecnológicos

En el objeto tecnológico se necesita seleccionar el motor conectado, asignar el encoder para conocer la posición del rotor y el protocolo de comunicación a emplear.

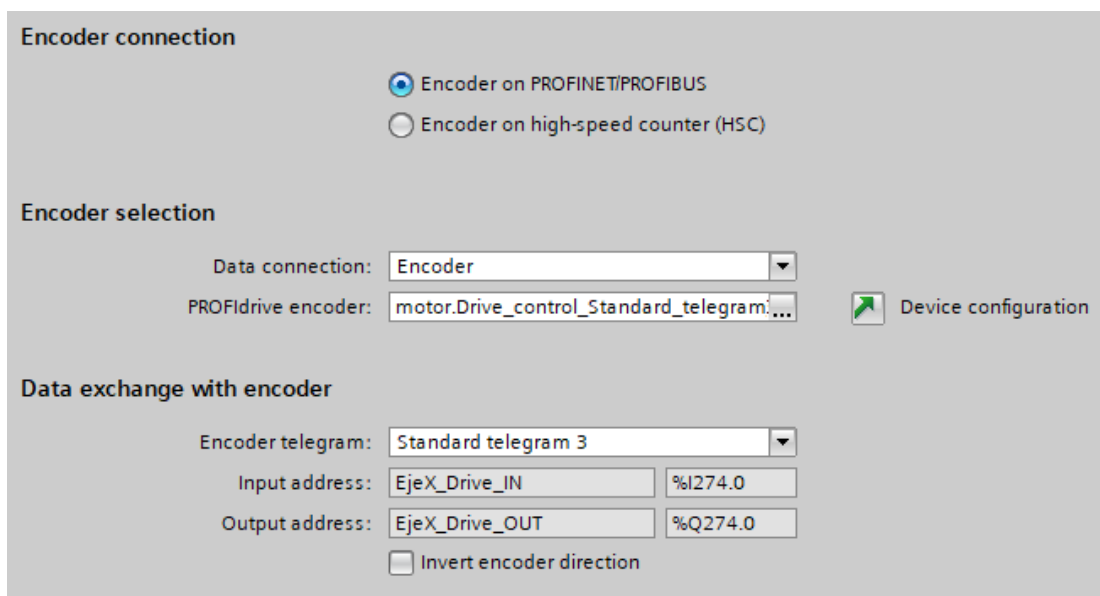


Figura 65 Configuración Eje tecnológico

4. Añadir bloques

Para escribir el programa, se dispone de un MAIN, el cual lee el autómata y puede utilizar cualquier lenguaje normalizado, en el caso actual se ha empleado STL, que es la interpretación que SIEMENS le da al lenguaje ST.

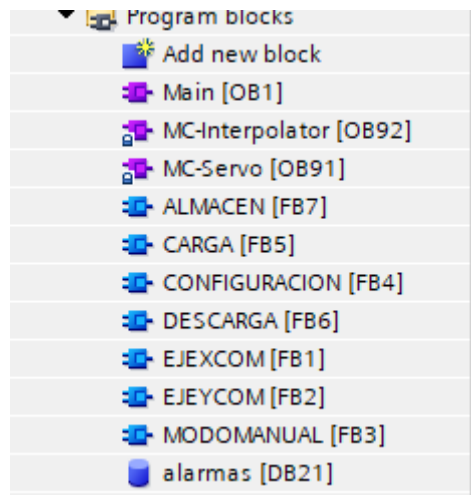


Figura 66 Bloques del programa

Para añadir los bloques de función, funciones y bloques de datos se emplea la función añadir nuevo bloque dentro de la cual se puede elegir el tipo y el lenguaje que se usa.

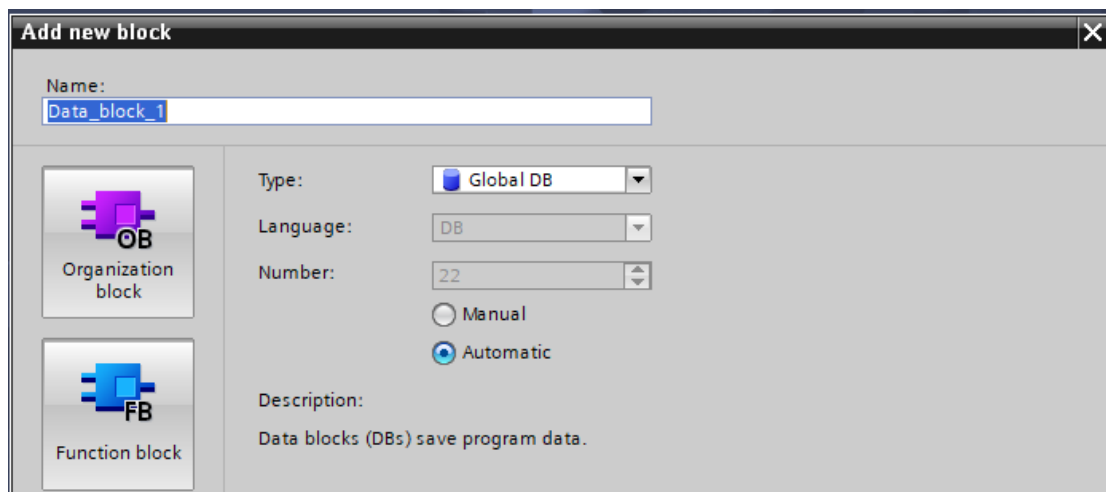


Figura 67 Añadir bloques

Las variables deben de ser declaradas en tablas dentro de las cuales se asignan nombres, dirección, tipo de variable y valor inicial. Estas pueden ser llamadas al programa para ser editadas.

	Name	Tag table	Data type	Address
1	EjeX_Drive_IN	Default tag table	"PD_TEL3_IN"	%I274.0
2	EjeX_Drive_OUT	Default tag table	"PD_TEL3_OUT"	%Q274.0
3	EjeY_Drive_IN	Default tag table	"PD_TEL3_IN"	%I256.0
4	EjeY_Drive_OUT	Default tag table	"PD_TEL3_OUT"	%Q256.0
5	HABILITAREJEX	Default tag table	Bool	%M10.0
6	HABILITAREJEY	Default tag table	Bool	%M39.0
7	MANUAL	VARIABLESMODOS	Bool	%M200.0
8	AUTOMATICO	VARIABLESMODOS	Bool	%M200.2
9	CONFIG	VARIABLESMODOS	Bool	%M200.4
10	AUXMANUAL	VARIABLESMODOS	Bool	%M200.6
11	AUXAUTO	VARIABLESMODOS	Bool	%M201.0
12	AUXCONFIG	VARIABLESMODOS	Bool	%M201.2
13	JOG	VARIABLESMODOS	Bool	%M201.4

Figura 68 Tabla de variables

5. Herramientas de edición

La plataforma contiene gran cantidad de estructuras precargadas, como IFs, Case of, Do While

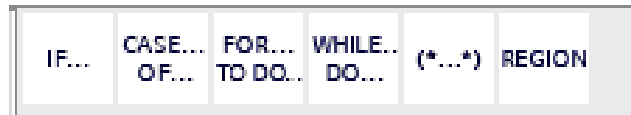


Figura 69 Herramientas programa

y bloques que resultan indispensables para cualquier código como temporizadores, contadores, conversores y herramientas para trabajar sobre memorias.

Name	Description
Bit logic operations	
R_TRIG	Detect positive signal e..
F_TRIG	Detect negative signal ...
Timer operations	
TP	Generate pulse
TON	Generate on-delay
TOF	Generate off-delay
TONR	Time accumulator

Figura 70 Bloques insertables

Además, existen funciones específicas para comunicarse con el variador como MOVE_ABSOLUTE, que permite mover el eje a una posición absoluta siempre que el mismo este referenciado o POWER que habilita la alimentación al motor.

Technology		
Name	Description	Version
▶ Counting		V1.1
▶ PID Control		
▼ Motion Control		V7.0
MC_Power	Enable/disable axis	V7.0
MC_Reset	Acknowledge errors, re...	V7.0
MC_Home	Home axis, set home p...	V7.0
MC_Halt	Pause axis	V7.0
MC_MoveAbsolute	Position axis absolutely	V7.0
MC_MoveRelative	Position axis relatively	V7.0
MC_MoveVelocity	Move axis at predefine...	V7.0
MC_MoveJog	Move axis in Jog mode	V7.0
MC_CommandTable	Run axis commands as ..	V7.0
MC_ChangeDynamic	Change Dynamics setti...	V7.0
MC_WriteParam	Write parameters of a t...	V7.0
MC_ReadParam	Read parameters of a te...	V7.0

Figura 71 Tecnología Motion Control

Finalmente, los bloques de función, objetos tecnológicos, bloques de datos y demás elementos pueden ser vertidos al main desde donde el autómata leerá y escribirá información en los mismos. Mediante la síntesis de todas estas herramientas se ha diseñado el programa.

6. Cargar y monitorizar programa

Una vez se termina el programa, se compila el código y en caso de ser correcto se procede a cargar el código al PLC. Una vez está cargado, se activa el modo online y se arranca la CPU.

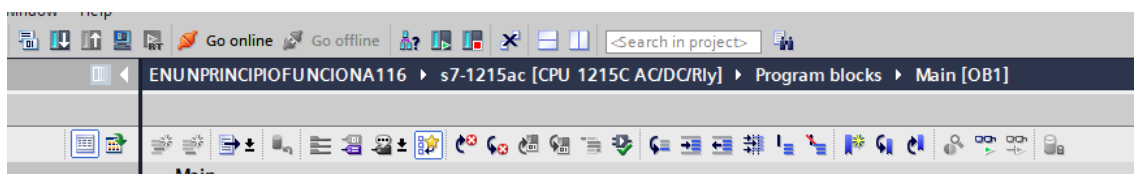


Figura 72 Funciones barra herramientas

En caso de que el LED de error parpadease en rojo, es posible comprobar el error en la pantalla diagnóstico del PLC. Los errores del variador pueden verse en la información de los bloques MOTION CONTROL.

Anexo III Manual WinCC

1. Introducción

Finalmente, se diseñará el SCADA con la herramienta WinCC. Este complemento de TIA PORTAL ofrece soporte a las pantallas HMI de la marca SIEMENS. Además, permite utilizar un PC industrial y simular su funcionamiento en un ordenador personal. En este caso se va a simular en el ordenador del proyecto.

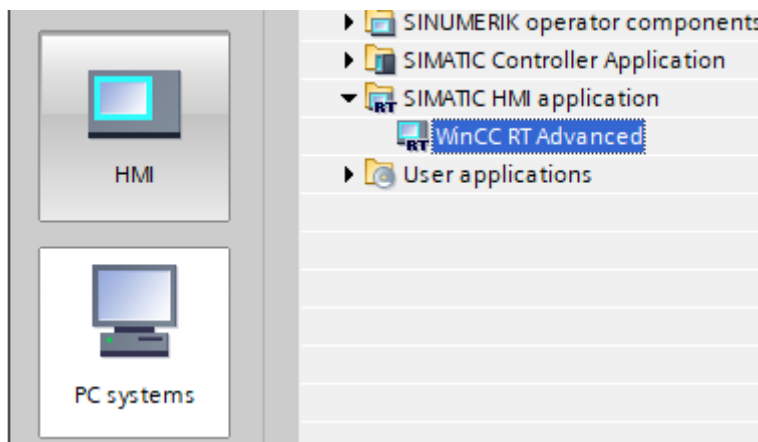


Figura 73 Insertar simulación

Los parámetros de la pantalla simulada se pueden consultar y editar en la pestaña configuración del runtime.

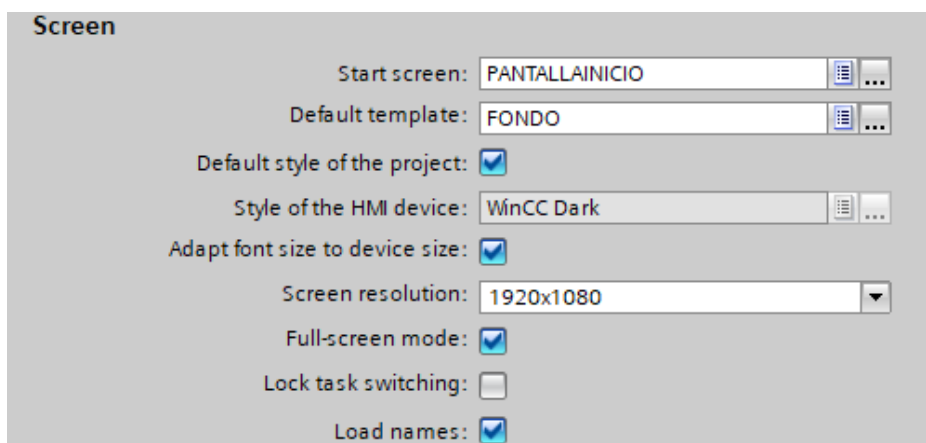


Figura 74 Configuración Runtime

2. Generar pantallas y plantillas

Primeramente, se introduce una nueva pantalla utilizando “Crear nueva pantalla”. Se puede seleccionar cual es la pantalla que aparecerá por defecto como principal o aplicar plantillas entre otras funcionalidades. Es posible dotar de un patrón a las pantallas generando una plantilla y aplicándola a todas estas.

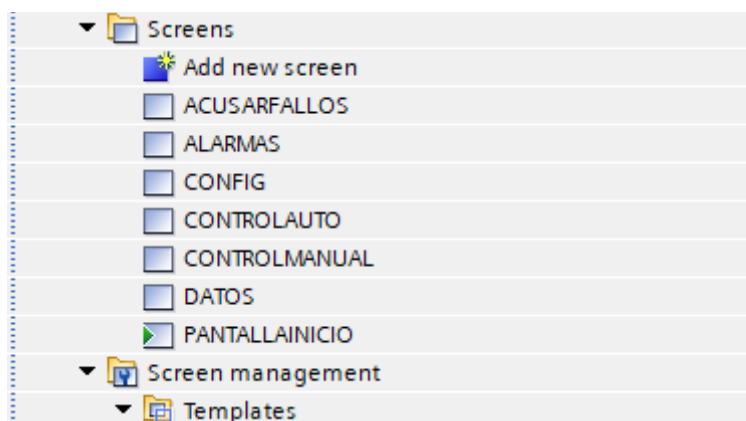


Figura 75 Pantallas proyecto

3. Herramientas de edición

Dentro del menú de edición de las pantallas, es posible emplear distintas funcionalidades. Se destacan

- **Botones** Los botones activan una de las muchas funciones programadas. Por ejemplo, pueden utilizarse para cambiar pantallas, modificar valores de variables, cambiar de idiomas, activar la pantalla de login entre otras funcionalidades.
- **Visualizadores** Es posible visualizar el valor real de una variable mediante un display. Se utilizan marcadores para monitorizar variables numéricas, escritas o lógicas. También pueden representarse de forma gráfica editando la apariencia del objeto y uniéndolo al valor concreto de una variable.

- **Animaciones** Otra posibilidad interesante es la de animar un objeto. Esto hará que, dependiendo del valor de una variable de tipo entero, la posición de la imagen irá variando de forma vertical, horizontal, diagonal o en las dos dimensiones de forma simultánea.



Figura 76 Elementos SCADA

4. Recetas

El menú de recetas permite crear una estructura de datos en el HMI que se encuentre relacionada con otra en el PLC. Es útil para generar perfiles editables o para transmitir una cantidad de datos razonable de una sola vez.

...	Name	Display name	Number	Version	Path
	BLASCO	BLASCO	1	6/15/2021 12:0...	C:\Recipe
	CEDRO	CEDRO	2	6/15/2021 12:1...	C:\Recipe
	ARAGÓN	ARAGÓN	3	6/15/2021 12:1...	C:\Recipe
	<Add new>				

...	Name	Display name	Tag	Data type
	PASTA	PASTA	PEDIDOS_PEDIDO...	Int
	CARNES	CARNES	PEDIDOS_PEDIDO...	Int
	LACTEOS	LACTEOS	PEDIDOS_PEDIDO...	Int

Figura 77 Interfaz recetas

Además, se puede añadir un visualizador de estas al SCADA, desde el cual es posible editar y comprobar los valores de las mismas.

Recipe Name: No.:

Data Record Name: No.:

Entry Name	Value

Figura 78 Visualizador de recetas

5. Usuarios

Para mejorar la seguridad del SCADA es interesante gestionar permisos de usuario. Con esto se puede restringir el acceso de las distintas pantallas tan solo a las personas que se considere adecuado ya que será necesario conocer el nombre de usuario y la contraseña para acceder a la misma.

Users				
	Name	Password	Automatic logoff	Logoff time
	Administrador	*****	<input checked="" type="checkbox"/>	5
	Operario	*****	<input checked="" type="checkbox"/>	5
	Mantenimiento	*****	<input checked="" type="checkbox"/>	5
	<Add new>			

Groups				
	Member of	Name	Number	Display name
	<input checked="" type="radio"/>	Administrator group	1	Administrator group
	<input type="radio"/>	Users	2	Users
	<input type="radio"/>	Mantenimiento	3	Group_1
	<Add new>			

Figura 79 Interfaz de usuarios

6. Alarmas

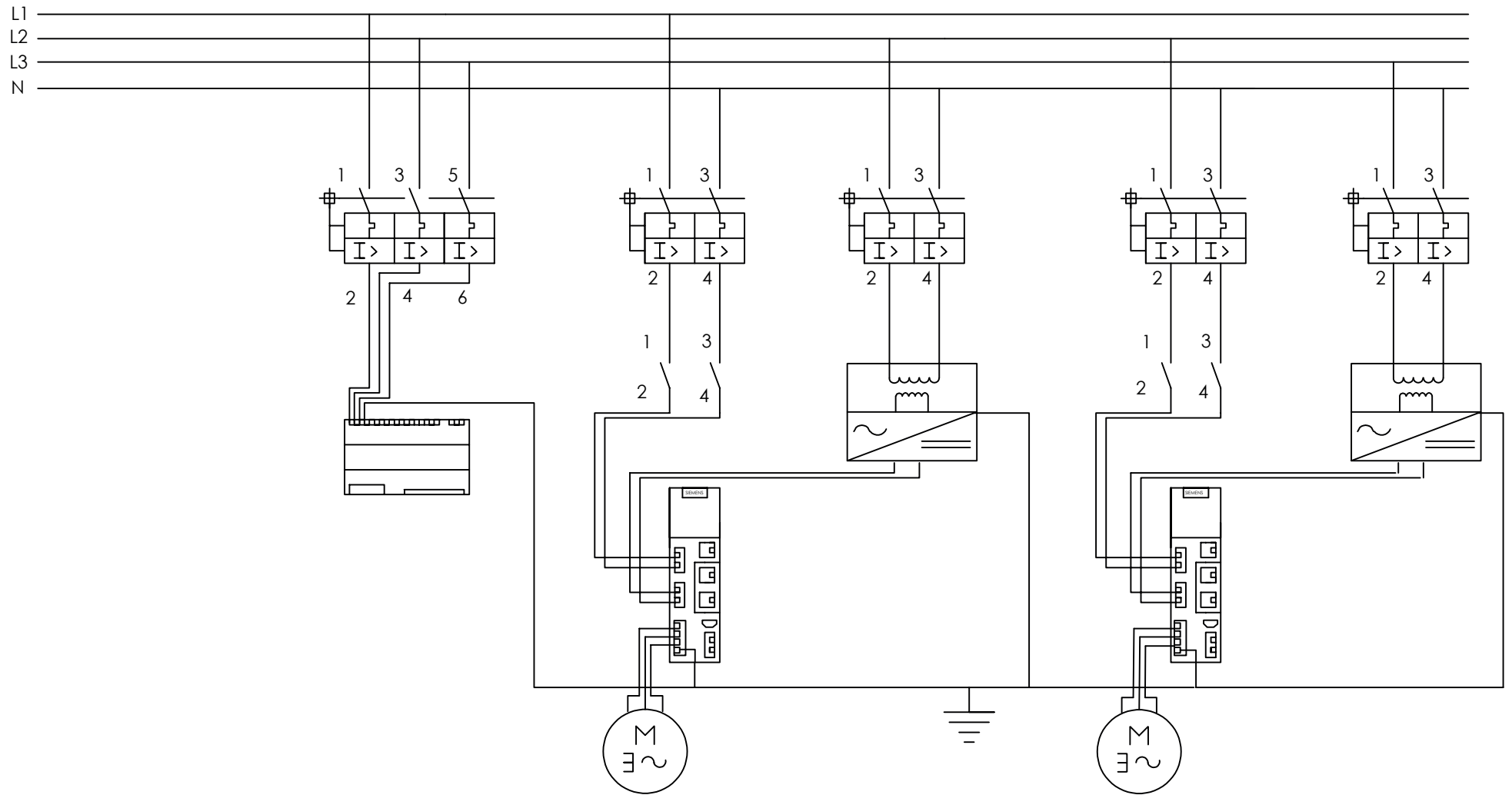
Es posible también generar una tabla de acciones que tienen lugar cuando se activan determinadas variables lógicas o cuando otras numéricas superan cierto umbral. Estos avisos pueden configurarse de forma que aparezca una pantalla flotante en el momento en el que se active cualquiera de ellos en la cual se pueda leer una descripción del error surgido y la hora exacta de su aparición.


ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Trigger address
1	Termico	Disparo del Térmico	Errors	Alarmas	8	%DB21.DBX0.0
2	Emergencia	Paro pulsado	Errors	Alarmas	9	%DB21.DBX0.1
3	Perimetro	Perimetro violado	Errors	Alarmas	10	%DB21.DBX0.2
4	Lleno	Almacén lleno	Warnings	Alarmas	11	%DB21.DBX0.3
5	Vacio	Almacén vacío	Warnings	Alarmas	12	%DB21.DBX0.4

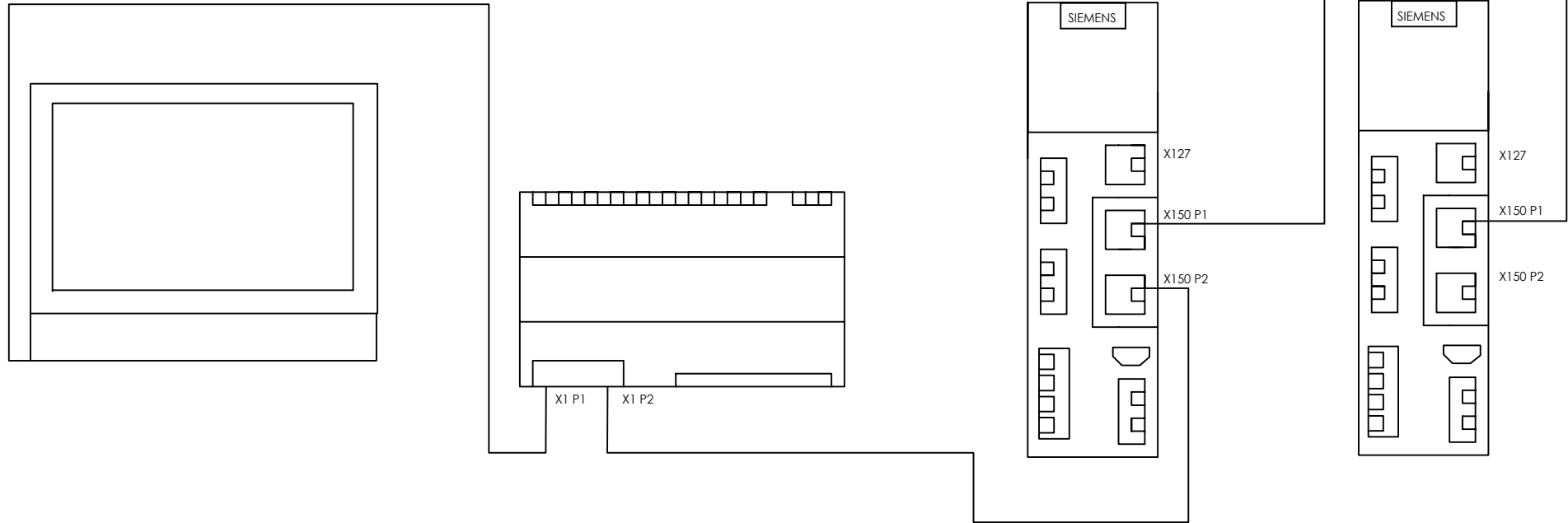
Figura 80 Definición de alarmas


7. Idiomas

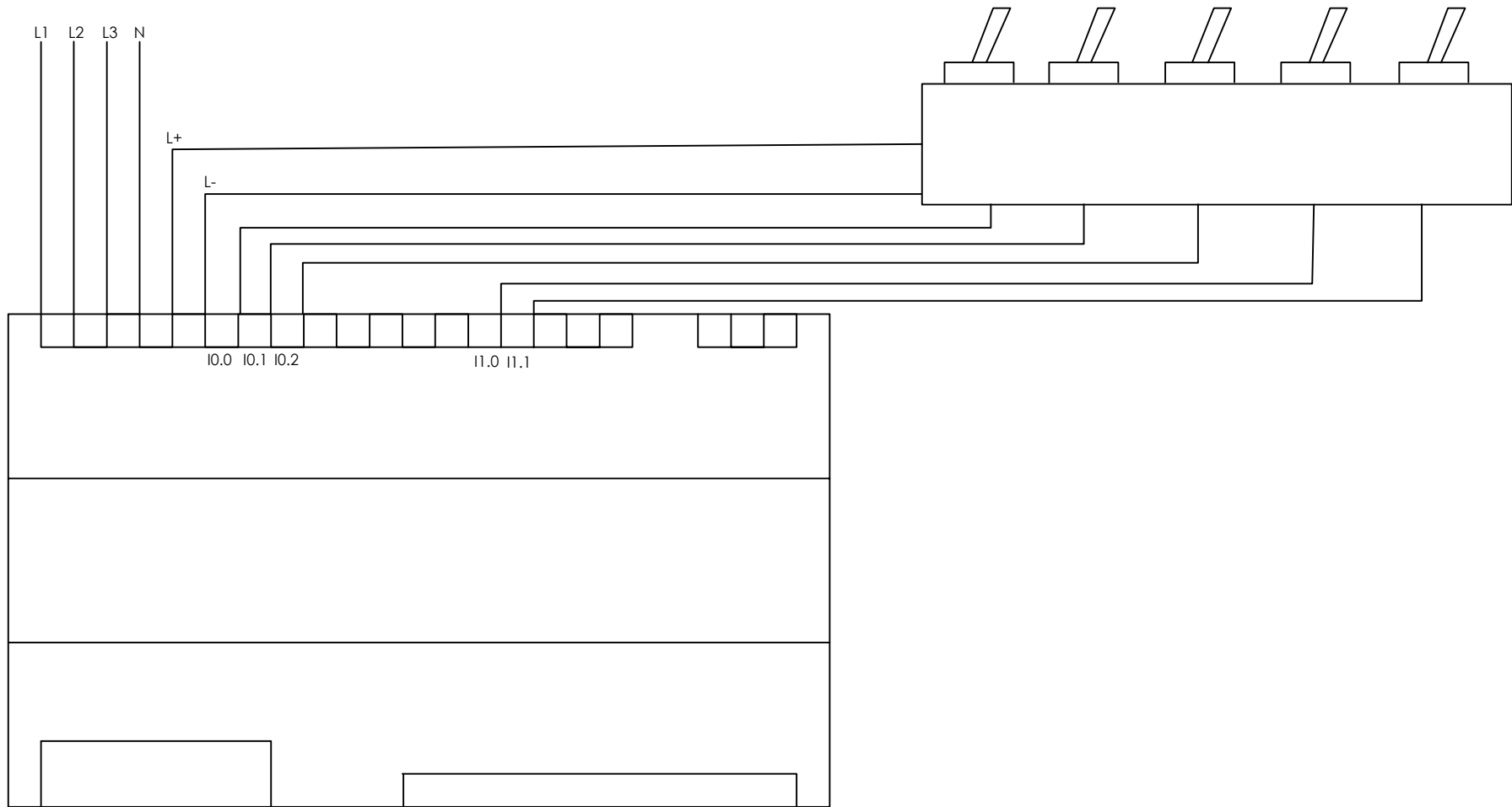
Por último, es posible activar idiomas de una biblioteca interna del programa, esto permitirá intercambiar el idioma utilizado al pulsarse un botón con la función de cambio de idioma activada.




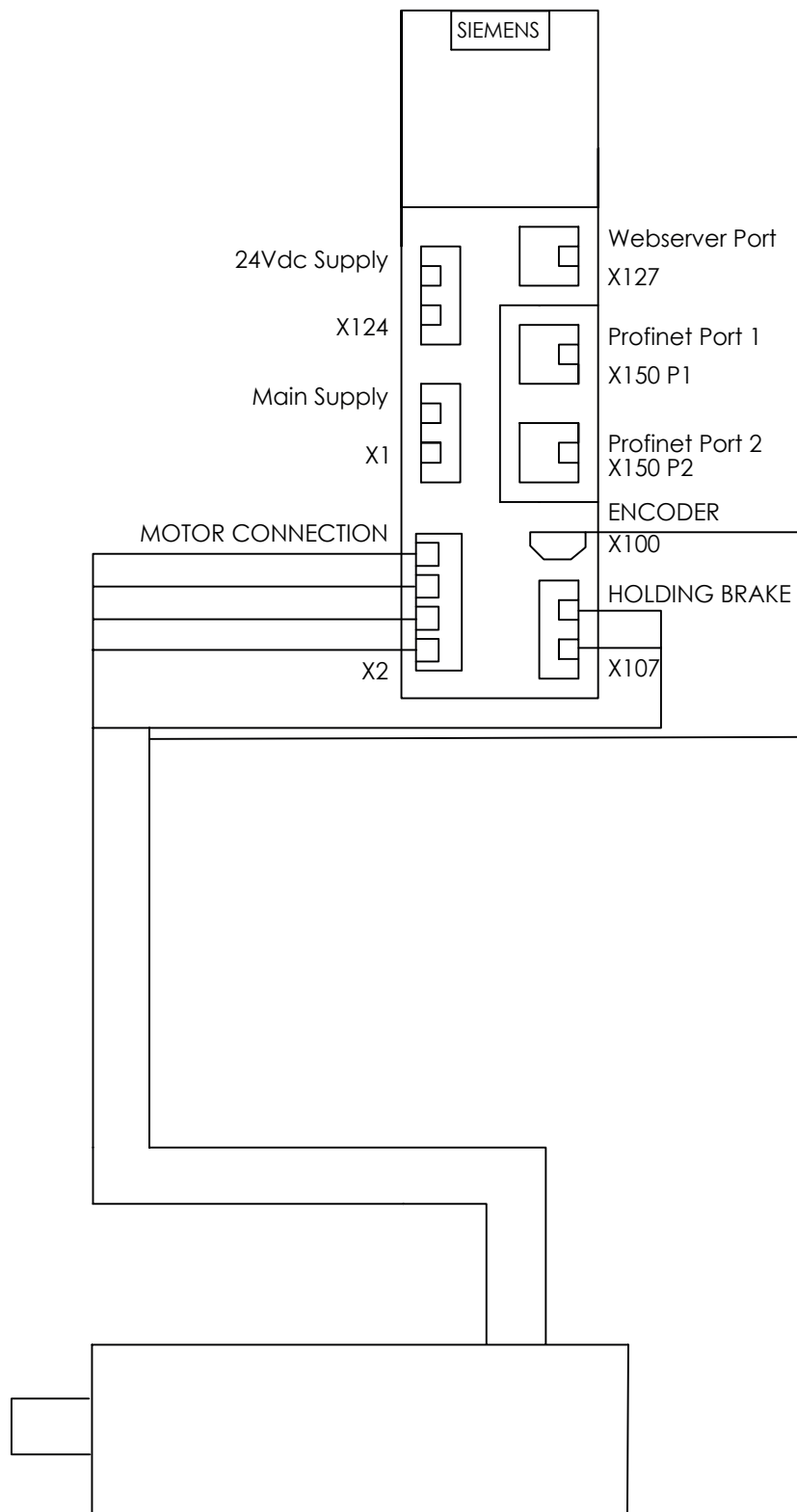
	NOMBRE	FECHA	NOMBRE DEL PLANO	
DIBUJADO	ALEJANDRO BALLESTEROS GARCÍA	20/06/2021	CONEXIONADO ELÉCTRICO	
ENTIDAD	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Valencia, Junio 2021	TÍTULO	
		A3	ESCALA: S/N	HOJA 1 DE 4
			DISEÑO DE CONTROL DE POSICIÓN	




	NOMBRE	FECHA	NOMBRE DEL PLANO	
DIBUJADO	ALEJANDRO BALLESTEROS GARCÍA	20/06/2021	RED PROFINET	
ENTIDAD	 UNIVERSITAT POLITÀCNICA DE VALÈNCIA	Valencia, Junio 2021	TÍTULO	
		A3	ESCALA: S/N	HOJA 2 DE 4
			DISEÑO DE CONTROL DE POSICIÓN	



	NOMBRE	FECHA	NOMBRE DEL PLANO	
DIBUJADO	ALEJANDRO BALLESTEROS GARCÍA	20/06/2021	CONEXIONES PLC	
ENTIDAD	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Valencia, Junio 2021	TÍTULO	
		A3	ESCALA: S/N	HOJA 3 DE 4
			DISEÑO DE CONTROL DE POSICIÓN	



	NOMBRE	FECHA	NOMBRE DEL PLANO	
DIBUJADO	ALEJANDRO BALLESTEROS GARCÍA	20/06/2021	CONEXIONES VARIADOR	
ENTIDAD	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Valencia, Junio 2021	TÍTULO	
		A4	DISEÑO DE CONTROL DE POSICIÓN	
			ESCALA: S/N	HOJA 4 DE 4

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	SCL
Numbering	Automatic						

Information

Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

```

0001 (*LLAMADA A LAS FUNCIONES DE DATOS*)
0002 "EJEXCOM_DB"(INPUTAXIS1 := "INPUTS/OUTPUTS".INPUTSX,
0003     OUTPUTAXIS1 => "INPUTS/OUTPUTS".OUTPUTSX);
0004 "EJEYCOM_DB"(INPUTAXIS2:="INPUTS/OUTPUTS".INPUTSY,
0005     OUTPUTAXIS2=>"INPUTS/OUTPUTS".OUTPUTSY);
0006 IF "Tag_6" THEN
0007     "alarmas".Termico := TRUE;
0008 ELSE
0009     "alarmas".Termico := FALSE;
0010 END_IF;
0011 IF ("INPUTS/OUTPUTS".INPUTSX.EN_POW OR "INPUTS/OUTPUTS".INPUTSY.EN_POW) AND "EjeX".ActualVelocity < 1 AND "EjeY".ActualVelocity < 1 THEN
0012     "amarillo" := true;
0013     "rojo" := false;
0014     "verde" := false;
0015 END_IF;
0016 IF ("INPUTS/OUTPUTS".INPUTSX.EN_POW OR "INPUTS/OUTPUTS".INPUTSY.EN_POW) AND (("EjeX".ActualVelocity > 10.0 OR "EjeY".ActualVelocity>10.0) OR ("EjeX".ActualVelocity <- 10.0 AND "EjeY".ActualVelocity<-10.0)) THEN
0017     "amarillo" := FALSE;
0018     "rojo" := TRUE;
0019     "verde" := false;
0020 END_IF;
0021 IF NOT "INPUTS/OUTPUTS".INPUTSX.EN_POW AND NOT "INPUTS/OUTPUTS".INPUTSY.EN_POW THEN
0022     "amarillo" := FALSE;
0023     "rojo" := false;
0024     "verde" := true;
0025 END_IF;
0026
0027 IF "AUXLLENADO" AND "INPUTS/OUTPUTS".INPUTSX.EN_POW AND "INPUTS/OUTPUTS".INPUTSY.EN_POW THEN
0028     "movimiento" := "movimiento" + 1;
0029     IF "movimiento" >= 1000 THEN
0030         "movimiento" := 0;
0031     END_IF;
0032
0033     "VERBLOQUE" := TRUE;
0034
0035 END_IF;
0036 IF "AUXVACIADO" AND "INPUTS/OUTPUTS".INPUTSX.EN_POW AND "INPUTS/OUTPUTS".INPUTSY.EN_POW THEN
0037     "movimiento" := "movimiento" - 1;
0038     IF "movimiento" <= 0 THEN
0039         "movimiento" := 1000;
0040     END_IF;
0041     "VERBLOQUE" := TRUE;
0042
0043 END_IF;
0044 IF NOT "AUXVACIADO" AND NOT "AUXLLENADO" OR ("EjeX".ActualPosition>=1.0 AND "EjeY".ActualPosition>=1.0) THEN
0045     "VERBLOQUE" := FALSE;
0046 END_IF;
0047 "VISORX" := "EjeX".ActualPosition;
0048 "VISORY" := "EjeY".ActualPosition;
0049 "PRODUCTO" := BYTE_TO_INT("PROD");
0050 "POSAUX" := REAL_TO_INT("EjeX".ActualPosition);
0051 "POSAUXY" := REAL_TO_INT("EjeY".ActualPosition);
0052 "AUXPAQUETEX" := - "POSAUX" * 0.63;
0053 "AUXPAQUETEXY" := "POSAUXY" * 0.64;
0054 "RecipeExport_DB"(REQ := "GENERAREXCEL",
0055     RECIPE_DB := "DATOSMATRICES".MATRIZALMACEN);
0056 IF "ACUSARFALLO" THEN
0057     IF "CORREGIR" THEN
0058         "DATOSMATRICES".MATRIZALMACEN["POSIX", "POSIY"] := "PRODSELEC";
0059     END_IF;
0060     IF "VACIAR" THEN
0061         "DATOSMATRICES".MATRIZALMACEN := "DATOSMATRICES".VACIAAUXILIAR;
0062     END_IF;
0063 END_IF;
0064 (*ENCALVAMIENTO MODOS*)
0065 IF "MANUAL" AND NOT "AUXAUTO" AND NOT "AUXCONFIG" THEN
0066     "AUXMANUAL" := TRUE;

```

```
0067 ELSIF NOT "MANUAL" THEN
0068     "AUXMANUAL" := FALSE;
0069 END_IF;
0070 IF "AUTOMATICO" AND NOT "AUXMANUAL" AND NOT "AUXCONFIG" THEN
0071     "AUXAUTO" := TRUE;
0072 ELSIF NOT "AUTOMATICO" THEN
0073     "AUXAUTO" := FALSE;
0074 END_IF;
0075 IF "CONFIG" AND NOT "AUXAUTO" AND NOT "AUXMANUAL" THEN
0076     "AUXCONFIG" := TRUE;
0077 ELSIF NOT "CONFIG" THEN
0078     "AUXCONFIG" := FALSE;
0079 END_IF;
0080 IF "AUXAUTO" THEN
0081     IF "LLENADO" AND NOT "AUXVACIADO" THEN
0082         "AUXLLENADO" := TRUE;
0083     ELSIF NOT "LLENADO" THEN
0084         "AUXLLENADO" := FALSE;
0085     END_IF;
0086     IF "VACIADO" AND NOT "AUXLLENADO" THEN
0087         "AUXVACIADO" := TRUE;
0088     ELSIF NOT "VACIADO" THEN
0089         "AUXVACIADO" := FALSE;
0090     END_IF;
0091 ELSE
0092     "LLENADO" := FALSE;
0093     "VACIADO" := FALSE;
0094 END_IF;
0095
0096
0097 (* LLAMADA A LA GESTION DEL ALMACÉN *)
0098 "ALMACEN_DB" (ENABLE := "AUXAUTO",
0099     CARGA := "AUXLLENADO",
0100     DESCARGA := "AUXVACIADO",
0101     PRODUCTO := "PRODUCTO",
0102     MENOS := "AUXMENOS",
0103     MAS := "AUXMAS",
0104     POSX => "AUXPOSX",
0105     POSY => "AUXPOSY");
0106
0107 IF "AUXCONFIG" THEN
0108 (* LLAMADA A LA FUNCION COMUNICACION *)
0109     "CONFIGURACION_DB" (ENABLE := "AUXCONFIG",
0110         SALIDAX => "INPUTS/OUTPUTS".INPUTSX,
0111         SALIDAY => "INPUTS/OUTPUTS".INPUTSY);
0112 END_IF;
0113
0114 IF "AUXMANUAL" THEN
0115 (*LLAMADA A LA FUNCIÓN MANUAL*)
0116     "MODOMANUAL_DB" (HABILITAR := "AUXMANUAL",
0117         "JOG" := "JOG",
0118         "MANUAL" := "PEDIDO",
0119         "RETORNO" := "RETORNO",
0120         SALIDAX => "INPUTS/OUTPUTS".INPUTSX_MAN,
0121         SALIDAY => "INPUTS/OUTPUTS".INPUTSY_MAN);
0122 END_IF;
0123
0124 IF "AUXLLENADO" THEN
0125 (* LLAMADA A LA FUNCIÓN "CARGA" *)
0126 "CARGA_DB" (ENABLE:="AUXLLENADO",
0127     PRODUCTO:="PRODUCTO",
0128     LISTO:="LISTO",
0129     POSX:="AUXPOSX",
0130     POSY:="AUXPOSY",
0131     MARCHACINTA=>"Tag_1",
0132     SALIDASX=>"INPUTS/OUTPUTS".INPUTSX_CARGA,
0133     SALIDASY=>"INPUTS/OUTPUTS".INPUTSY_CARGA,
0134     MENOS=>"AUXMENOS");
0135 END_IF;
0136
0137 IF "AUXVACIADO" THEN
0138     "DESCARGA_DB" (ENABLE_DESCARGA:="AUXVACIADO",
0139         SALIDA:="Tag_2",
0140         PEDIDOS:="PEDIDOS".PEDIDOS,
0141         MARCHACINTA=>"Tag_1",
0142         MAS=>"AUXMAS",
0143         SALIDASX=>"INPUTS/OUTPUTS".INPUTSX_DESCARGA,
0144         SALIDASY=>"INPUTS/OUTPUTS".INPUTSY_DESCARGA,
0145         VACIO=>"VACIO");
0146
0147     END_IF;
0148
0149 IF "AUXMANUAL" AND NOT "AUXLLENADO" AND NOT "AUXVACIADO" THEN
0150     "INPUTS/OUTPUTS".INPUTSX := "INPUTS/OUTPUTS".INPUTSX_MAN;
0151     "INPUTS/OUTPUTS".INPUTSY := "INPUTS/OUTPUTS".INPUTSY_MAN;
0152 ELSIF "AUXLLENADO" AND NOT "AUXVACIADO" AND NOT "AUXMANUAL" THEN
0153     "INPUTS/OUTPUTS".INPUTSX := "INPUTS/OUTPUTS".INPUTSX_CARGA;
0154     "INPUTS/OUTPUTS".INPUTSY := "INPUTS/OUTPUTS".INPUTSY_CARGA;
```

```

0155 ELSIF "AUXVACIADO" AND NOT "AUXLLENADO" AND NOT "AUXMANUAL" THEN
0156     "INPUTS/OUTPUTS".INPUTSX := "INPUTS/OUTPUTS".INPUTSX_DESCARGA;
0157     "INPUTS/OUTPUTS".INPUTSY := "INPUTS/OUTPUTS".INPUTSY_DESCARGA;
0158 END_IF;

```

Symbol	Address	Type	Comment
"ACUSARFALLO"	%M205.5	Bool	
"alarmas".Termico	%DB21.DBX0.0	Bool	
"amarillo"	%M206.3	Bool	
"AUTOMATICO"	%M200.2	Bool	
"AUXAUTO"	%M201.0	Bool	
"AUXCONFIG"	%M201.2	Bool	
"AUXLLENADO"	%M202.7	Bool	
"AUXMANUAL"	%M200.6	Bool	
"AUXMAS"	%M203.4	Bool	
"AUXMENOS"	%M203.3	Bool	
"AUXPAQUETEX"	%MW706	Int	
"AUXPAQUETEY"	%MW708	Int	
"AUXPOX"	%MD720	Real	
"AUXPOSY"	%MD740	Real	
"AUXVACIADO"	%M203.1	Bool	
"CONFIG"	%M200.4	Bool	
"CORREGIR"	%M205.7	Bool	
"DATOSMATRICES".MATRIZALMACEN		Array	EN CADA POSICION SE GUARDA EL PRODUCTO ALMACENADO
"DATOSMATRICES".MATRIZALMA- CEN[*,*]		Int	EN CADA POSICION SE GUARDA EL PRODUCTO ALMACENADO
"DATOSMATRICES".VACIAAUXILIAR		Array	
"EjeX".ActualPosition		Real	Actual position of the axis
"EjeX".ActualVelocity		Real	Actual velocity of the axis
"EjeY".ActualPosition		Real	Actual position of the axis
"EjeY".ActualVelocity		Real	Actual velocity of the axis
"GENERAREXCEL"	%M204.2	Bool	
"INPUTS/OUTPUTS".INPUTSX		Struct	
"INPUTS/OUTPUTS".INPUTSX.EN_POW		Bool	
"INPUTS/OUTPUTS".INPUTSX_CARGA		Struct	
"INPUTS/ OUTPUTS".INPUTSX_DESCARGA		Struct	
"INPUTS/OUTPUTS".INPUTSX_MAN		Struct	
"INPUTS/OUTPUTS".INPUTSY		Struct	
"INPUTS/OUTPUTS".INPUTSY.EN_POW		Bool	
"INPUTS/OUTPUTS".INPUTSY_CARGA		Struct	
"INPUTS/OUTPUTS".INPUTSY_DESCAR- GA		Struct	
"INPUTS/OUTPUTS".INPUTSY_MAN		Struct	
"INPUTS/OUTPUTS".OUTPUTSX		Struct	
"INPUTS/OUTPUTS".OUTPUTSY		Struct	
"JOG"	%M201.4	Bool	
"LISTO"	%I1.0	Bool	
"LLENADO"	%M202.3	Bool	
"MANUAL"	%M200.0	Bool	
"movimiento"	%MW2000	Int	
"PEDIDO"	%M201.6	Bool	
"PEDIDOS".PEDIDOS		Array	
"POSAUX"	%MW700	Int	
"POSAUXY"	%MW704	Int	
"POSIX"	%MW714	Int	
"POSIY"	%MW716	Int	
"PROD"	%IB0	Byte	
"PRODSELEC"	%MW718	Int	
"PRODUCTO"	%MW600	Int	
"RETORNO"	%M202.1	Bool	
"rojo"	%M206.2	Bool	
"Tag_1"	%Q0.3	Bool	
"Tag_2"	%I1.1	Bool	
"Tag_6"	%M1.7	Bool	
"VACIADO"	%M202.5	Bool	
"VACIAR"	%M206.0	Bool	
"VACIO"	%M203.6	Bool	
"VERBLOQUE"	%M3000.0	Bool	
"verde"	%M206.1	Bool	
"VISORX"	%MD700	Real	
"VISORY"	%MD710	Real	

ALMACEN [FB7]

ALMACEN Properties

General

Name	ALMACEN	Number	7	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
ENABLE	Bool	false	Non-retain	True	True	True	False		
CARGA	Bool	false	Non-retain	True	True	True	False		
DESCARGA	Bool	false	Non-retain	True	True	True	False		
PRODUCTO	Int	1	Non-retain	True	True	True	False		
MENOS	Bool	false	Non-retain	True	True	True	False		
MAS	Bool	false	Non-retain	True	True	True	False		
▼ Output									
POSX	Real	0.0	Non-retain	True	True	True	False		
POSY	Real	0.0	Non-retain	True	True	True	False		
InOut									
▼ Static									
RESET	Bool	false	Non-retain	True	True	True	False		
Temp									
Constant									

```

0001 IF #ENABLE THEN
0002     "IEC_Counter_0_DB".CTUD(CU := #MENOS,
0003         CD := #MAS,
0004         R := #RESET,
0005         PV := 24,
0006         CV => "DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO) ;
0007 IF #CARGA THEN
0008     #POSX := "DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].X;
0009     #POSY := "DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].Y;
0010
0011 END_IF;
0012 IF #DESCARGA THEN
0013     #POSX := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[#PRODUCTO].HUECO].X;
0014     #POSY := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[#PRODUCTO].HUECO].Y;
0015 END_IF;
0016 END_IF;
    
```

Symbol	Address	Type	Comment
"DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[*].HUECO		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POSI[*].X		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POSI[*].Y		Int	
"DATOSMATRICES".MATRIZVACIOS[*].X		Int	UTILIZANDO UNA ESTRUCTURA DE PILA, SE ALMACENAN LAS POSICIONES DISPONIBLES
"DATOSMATRICES".MATRIZVACIOS[*].Y		Int	UTILIZANDO UNA ESTRUCTURA DE PILA, SE ALMACENAN LAS POSICIONES DISPONIBLES
#CARGA		Bool	
#DESCARGA		Bool	
#ENABLE		Bool	
#MAS		Bool	
#MENOS		Bool	
#POSX		Real	
#POSY		Real	
#PRODUCTO		Int	
#RESET		Bool	

CARGA [FB5]

CARGA Properties

General

Name	CARGA	Number	5	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
ENABLE	Bool	false	Non-retain	True	True	True	False		
PRODUCTO	Int	0	Non-retain	True	True	True	False		INDICA EL NUMERO DE PRODUCTO
LISTO	Bool	false	Non-retain	True	True	True	False		INDICA SI EL PRODUCTO ESTÁ EN LA BANDEJA
POSX	Real	0.0	Non-retain	True	True	True	False		
POSY	Real	0.0	Non-retain	True	True	True	False		
▼ Output									
MARCHACINTA	Bool	false	Non-retain	True	True	True	False		
▼ SALIDASX									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
▼ SALIDASY									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
MENOS	Bool	false	Non-retain	True	True	True	False		
InOut									
▼ Static									
VUELTA	Bool	false	Non-retain	True	True	True	False		
CONTADOR	Int	0	Non-retain	True	True	True	False		
IDA	Bool	false	Non-retain	True	True	True	False		
PASO	Real	200.0	Non-retain	True	True	True	False		
LLEGADA	Bool	false	Non-retain	True	True	True	False		
AUXLLEGADA	Bool	false	Non-retain	True	True	True	False		
START	Bool	false	Non-retain	True	True	True	False		
HABILITAR	Bool	false	Non-retain	True	True	True	False		
▼ PRODUCTOS									
PRODUCTOS[0]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[1]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[2]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[3]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[4]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[5]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[6]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[7]	Int	0	Non-retain	True	True	True	False		
PRODUCTOS[8]	Int	0	Non-retain	True	True	True	False		
PORQUESI	Bool	false	Non-retain	True	True	True	False		
Temp									
Constant									

```

0001 IF #ENABLE THEN
0002     IF #HABILITAR THEN
0003         #SALIDASX.EN_POW := TRUE;
0004         #SALIDASY.EN_POW := TRUE;
    
```

Totally Integrated Automation Portal		
<pre> 0005 ELSIF NOT #HABILITAR THEN 0006 #SALIDASX.EN_POW := FALSE; 0007 #SALIDASY.EN_POW := FALSE; 0008 END_IF; 0009 (* VELOCIDAD DEL ACCIONAMIENTO *) 0010 #SALIDASX.VEL_ABSM := 100.0; 0011 #SALIDASY.VEL_ABSM:= 100.0; 0012 (* LLEGADA EN EJEY O EJEX *) 0013 IF #POSX > #POSY THEN 0014 #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM; 0015 ELSIF #POSX < #POSY THEN 0016 #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM; 0017 ELSE 0018 #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM; 0019 END_IF; 0020 0021 IF #PORQUESI THEN 0022 "PROD1".CV := 0; 0023 "PROD2".CV := 0; 0024 "PROD3".CV := 0; 0025 "PROD4".CV := 0; 0026 "PROD5".CV := 0; 0027 "PROD6".CV := 0; 0028 "PROD7".CV := 0; 0029 END_IF; 0030 IF #PRODUCTO = 1 THEN 0031 "PROD1".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0032 PV := 24, 0033 CV => "DATOSMATRICES".MATRIZDISPONIBLE[1].HUECO); 0034 ELSIF #PRODUCTO = 2 THEN 0035 "PROD2".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0036 PV := 24, 0037 CV => "DATOSMATRICES".MATRIZDISPONIBLE[2].HUECO); 0038 ELSIF #PRODUCTO = 3 THEN 0039 "PROD3".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0040 PV := 24, 0041 CV => "DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO); 0042 ELSIF #PRODUCTO = 4 THEN 0043 "PROD4".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0044 PV := 24, 0045 CV => "DATOSMATRICES".MATRIZDISPONIBLE[4].HUECO); 0046 ELSIF #PRODUCTO = 5 THEN 0047 "PROD5".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0048 PV := 24, 0049 CV => "DATOSMATRICES".MATRIZDISPONIBLE[5].HUECO); 0050 ELSIF #PRODUCTO = 6 THEN 0051 "PROD6".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0052 PV := 24, 0053 CV => "DATOSMATRICES".MATRIZDISPONIBLE[6].HUECO); 0054 ELSIF #PRODUCTO = 7 THEN 0055 "PROD7".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100, 0056 PV := 24, 0057 CV => "DATOSMATRICES".MATRIZDISPONIBLE[7].HUECO); 0058 END_IF; 0059 (* SE GUARDA LA SEÑAL DE DONE *) 0060 IF #LLEGADA AND "EjeX".ActualPosition>=#POSX*#PASO-100 AND "EjeY".ActualPosition>=#POSY*#PASO-100 THEN 0061 #AUXLLEGADA := TRUE; 0062 "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[#PRODUCTO].HUECO].X := #POSX; 0063 "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI["DATOSMATRICES".MATRIZDISPONIBLE[#PRODUCTO].HUECO].Y := #POSY; 0064 "DATOSMATRICES".MATRIZALMACEN[REAL_TO_INT(#POSX), REAL_TO_INT(#POSY)] := #PRODUCTO; 0065 END_IF; 0066 IF "EjeX".ActualPosition<=0.0+10 AND "EjeY".ActualPosition<=0.0+10 THEN 0067 #MARCHACINTA := TRUE; 0068 ELSE 0069 #MARCHACINTA := FALSE; 0070 END_IF; 0071 (* ESPERA PARA CARGAR EL #PRODUCTO *) 0072 "LLEGADABIS".TON(IN := #AUXLLEGADA, 0073 PT := T#2s); 0074 0075 IF "LLEGADABIS".Q THEN 0076 0077 #SALIDASX.POS_ABSM := 0.0; 0078 #SALIDASY.POS_ABSM := 0.0; 0079 #AUXLLEGADA := FALSE; 0080 #MENOS := TRUE; 0081 END_IF; 0082 IF #LISTO THEN 0083 #MENOS := FALSE; 0084 END_IF; 0085 IF "EjeX".ActualPosition < 1 AND "EjeY".ActualPosition < 1 THEN </pre>		


```

0086     #SALIDASX.POS_ABSM := #PASO * #POSX;
0087     #SALIDASY.POS_ABSM := #PASO * #POSY;
0088     END_IF;
0089     IF #HABILITAR AND (#LISTO OR "LLEGADABIS".Q) THEN
0090         #START := TRUE;
0091     ELSE
0092         #START := FALSE;
0093     END_IF;
0094     IF #START AND NOT #AUXLLEGADA THEN
0095         #SALIDASX.EX_ABSM := TRUE;
0096         #SALIDASY.EX_ABSM := TRUE;
0097     ELSE
0098         #SALIDASX.EX_ABSM := FALSE;
0099         #SALIDASY.EX_ABSM := FALSE;
0100     END_IF;
0101
0102 END_IF;

```

Symbol	Address	Type	Comment
"DATOSMATRICES".MATRIZALMACEN[*,*]		Int	EN CADA POSICION SE GUARDA EL PRODUCTO ALMACENADO
"DATOSMATRICES".MATRIZDISPONIBLE[1].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[2].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[4].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[5].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[6].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[7].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[*].HUECO		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POS[*].X		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POS[*].Y		Int	
"EjeX".ActualPosition		Real	Actual position of the axis
"EjeY".ActualPosition		Real	Actual position of the axis
"INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM		Bool	
"INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM		Bool	
"LLEGADABIS".Q		Bool	
"PROD1".CV		Int	
"PROD2".CV		Int	
"PROD3".CV		Int	
"PROD4".CV		Int	
"PROD5".CV		Int	
"PROD6".CV		Int	
"PROD7".CV		Int	
#AUXLLEGADA		Bool	
#ENABLE		Bool	
#HABILITAR		Bool	
#LISTO		Bool	INDICA SI EL PRODUCTO ESTÁ EN LA BANDEJA
#LLEGADA		Bool	
#MARCHACINTA		Bool	
#MENOS		Bool	
#PASO		Real	
#PORQUESI		Bool	
#POSX		Real	
#POSY		Real	
#PRODUCTO		Int	INDICA EL NUMERO DE PRODUCTO
#SALIDASX.EN_POW		Bool	
#SALIDASX.EX_ABSM		Bool	
#SALIDASX.POS_ABSM		Real	
#SALIDASX.VEL_ABSM		Real	
#SALIDASY.EN_POW		Bool	
#SALIDASY.EX_ABSM		Bool	
#SALIDASY.POS_ABSM		Real	
#SALIDASY.VEL_ABSM		Real	
#START		Bool	

CONFIGURACION [FB4]

CONFIGURACION Properties

General

Name	CONFIGURACION	Number	4	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
ENABLE	Bool	false	Non-retain	True	True	True	False		
▼ Output									
▼ SALIDAX									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
▼ SALIDAY									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
InOut									
▼ Static									
HOMINGX	Bool	false	Non-retain	True	True	True	False		
HOMINGY	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGR	Bool	false	Non-retain	True	True	True	False		
JOGF1	Bool	false	Non-retain	True	True	True	False		
JOGR1	Bool	false	Non-retain	True	True	True	False		
CONTROL	Bool	false	Non-retain	True	True	True	False		
Temp									
Constant									

```

0001 IF #ENABLE THEN
0002     #SALIDAX.EX_HOME := #HOMINGX;
0003     #SALIDAY.EX_HOME := #HOMINGY;
0004     IF #CONTROL THEN
0005         #SALIDAX.EN_POW := TRUE;
0006         #SALIDAY.EN_POW := TRUE;
0007         #SALIDAX.JOGF := #JOGF;
0008         #SALIDAX.JOGREV := #JOGR;
0009         #SALIDAY.JOGF := #JOGF;
0010         #SALIDAY.JOGREV := #JOGR1;
0011     ELSE
0012         #SALIDAX.EN_POW := FALSE;
0013         #SALIDAY.EN_POW := FALSE;
0014     END_IF;
0015 END_IF;
    
```

Symbol	Address	Type	Comment
#CONTROL		Bool	
#ENABLE		Bool	
#HOMINGX		Bool	
#HOMINGY		Bool	
#JOGF		Bool	
#JOGR		Bool	
#JOGR1		Bool	
#SALIDAX.EN_POW		Bool	
#SALIDAX.EX_HOME		Bool	
#SALIDAX.JOGF		Bool	

Symbol	Address	Type	Comment
#SALIDAX.JOGREV		Bool	
#SALIDAY.EN_POW		Bool	
#SALIDAY.EX_HOME		Bool	
#SALIDAY.JOGF		Bool	
#SALIDAY.JOGREV		Bool	

DESCARGA [FB6]

DESCARGA Properties

General

Name	DESCARGA	Number	6	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
ENABLE_DESCARGA	Bool	false	Non-retain	True	True	True	False		
SALIDA	Bool	false	Non-retain	True	True	True	False		
▼ PEDIDOS	Array[1..3] of Struct		Non-retain	True	True	True	False		
▼ PEDIDOS[1]	Struct		Non-retain	True	True	True	False		
PASTA	Int	0	Non-retain	True	True	True	False		
CARNES	Int	0	Non-retain	True	True	True	False		
LACTEOS	Int	0	Non-retain	True	True	True	False		
VERDURAS	Int	0	Non-retain	True	True	True	False		
BEBIDAS	Int	0	Non-retain	True	True	True	False		
CONSERVAS	Int	0	Non-retain	True	True	True	False		
FRUTAS	Int	0	Non-retain	True	True	True	False		
▼ PEDIDOS[2]	Struct		Non-retain	True	True	True	False		
PASTA	Int	0	Non-retain	True	True	True	False		
CARNES	Int	0	Non-retain	True	True	True	False		
LACTEOS	Int	0	Non-retain	True	True	True	False		
VERDURAS	Int	0	Non-retain	True	True	True	False		
BEBIDAS	Int	0	Non-retain	True	True	True	False		
CONSERVAS	Int	0	Non-retain	True	True	True	False		
FRUTAS	Int	0	Non-retain	True	True	True	False		
▼ PEDIDOS[3]	Struct		Non-retain	True	True	True	False		
PASTA	Int	0	Non-retain	True	True	True	False		
CARNES	Int	0	Non-retain	True	True	True	False		
LACTEOS	Int	0	Non-retain	True	True	True	False		
VERDURAS	Int	0	Non-retain	True	True	True	False		
BEBIDAS	Int	0	Non-retain	True	True	True	False		
CONSERVAS	Int	0	Non-retain	True	True	True	False		
FRUTAS	Int	0	Non-retain	True	True	True	False		
▼ Output									
MARCHACINTA	Bool	false	Non-retain	True	True	True	False		
MAS	Bool	false	Non-retain	True	True	True	False		
▼ SALIDASX	Struct		Non-retain	True	True	True	False		
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
▼ SALIDASY	Struct		Non-retain	True	True	True	False		
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
VACIO	Bool	false	Non-retain	True	True	True	False		
InOut									
▼ Static									
VUELTA	Bool	false	Non-retain	True	True	True	False		
IDA	Bool	false	Non-retain	True	True	True	False		
PASO	Real	200.0	Non-retain	True	True	True	False		

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
LLEGADA	Bool	false	Non-retain	True	True	True	False		
AUXLLEGADA	Bool	false	Non-retain	True	True	True	False		
START	Bool	false	Non-retain	True	True	True	False		
HABILITAR	Bool	false	Non-retain	True	True	True	False		
PEDIDO	Int	1	Non-retain	True	True	True	False		
PASTA	Int	0	Non-retain	True	True	True	False		
CARNES	Int	0	Non-retain	True	True	True	False		
LACTEOS	Int	0	Non-retain	True	True	True	False		
VERDURAS	Int	0	Non-retain	True	True	True	False		
BEBIDAS	Int	0	Non-retain	True	True	True	False		
CONSERVAS	Int	0	Non-retain	True	True	True	False		
FRUTAS	Int	0	Non-retain	True	True	True	False		
CONTADOR	Int	0	Non-retain	True	True	True	False		
TOTAL	Int	0	Non-retain	True	True	True	False		
PRODUCTO	Int	0	Non-retain	True	True	True	False		
ACTUAL	Int	0	Non-retain	True	True	True	False		
POSX	Real	0.0	Non-retain	True	True	True	False		
POSY	Real	0.0	Non-retain	True	True	True	False		
RESET	Bool	false	Non-retain	True	True	True	False		
RESETBIS	Bool	false	Non-retain	True	True	True	False		
COMIENZA	Bool	false	Non-retain	True	True	True	False		
CONTADORPROD	Int	0	Non-retain	True	True	True	False		
Temp									
Constant									

```

0001 IF #ENABLE_DESCARGA THEN
0002   (* AUX *)
0003   #POSX := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].X;
0004   #POSY := "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].Y;
0005
0006   (* #HABILITAR EJES *)
0007   IF #HABILITAR THEN
0008     #SALIDASX.EN_POW := TRUE;
0009     #SALIDASY.EN_POW := TRUE;
0010   ELSE
0011     #SALIDASX.EN_POW := FALSE;
0012     #SALIDASY.EN_POW := FALSE;
0013   END_IF;
0014
0015   (* VELOCIDADES *)
0016   #SALIDASX.VEL_ABSM := 100.0;
0017   #SALIDASY.VEL_ABSM := 100.0;
0018
0019   (* LLEGADA EN EJEY O EJEX *)
0020   IF "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].X
0021     >"DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].Y THEN
0022     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM;
0023   ELSIF "DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].X
0024     <"DATOSMATRICES".MATRIZPRODUCTOS[#PRODUCTO].POSI[#CONTADORPROD].Y THEN
0025     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM;
0026   ELSE
0027     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM;
0028   END_IF;
0029
0030   (* DETERMINAR #PRODUCTO A EXTRAER *)
0031   #PASTA:=#PEDIDOS[#PEDIDO].PASTA;
0032   #CARNES := #PEDIDOS[#PEDIDO].CARNES;
0033   #LACTEOS := #PEDIDOS[#PEDIDO].LACTEOS;
0034   #VERDURAS := #PEDIDOS[#PEDIDO].VERDURAS;
0035   #BEBIDAS := #PEDIDOS[#PEDIDO].BEBIDAS;
0036   #CONSERVAS := #PEDIDOS[#PEDIDO].CONSERVAS;
0037   #FRUTAS := #PEDIDOS[#PEDIDO].FRUTAS;
0038   #TOTAL := #PASTA + #CARNES + #LACTEOS + #VERDURAS + #BEBIDAS + #CONSERVAS + #FRUTAS;
0039
0040   IF #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100 THEN
0041     "DATOSMATRICES".MATRIZALMACEN[REAL_TO_INT("EjeX".ActualPosition/#PASO), REAL_TO_INT("EjeY".ActualPosition/#PASO)] := 0;
0042   END_IF;
0043   IF #CONTADOR = #TOTAL - #FRUTAS - #CONSERVAS - #BEBIDAS - #VERDURAS - #LACTEOS - #CARNES THEN
0044     #PRODUCTO := 1;
0045   ELSIF #CONTADOR = #TOTAL - #FRUTAS - #CONSERVAS - #BEBIDAS - #VERDURAS - #LACTEOS THEN
0046     #PRODUCTO := 2;
0047   ELSIF #CONTADOR = #TOTAL - #FRUTAS - #CONSERVAS - #BEBIDAS - #VERDURAS THEN
0048     #PRODUCTO := 3;
0049   ELSIF #CONTADOR = #TOTAL - #FRUTAS - #CONSERVAS - #BEBIDAS THEN
0050     #PRODUCTO := 4;
0051   ELSIF #CONTADOR = #TOTAL - #FRUTAS - #CONSERVAS THEN
0052     #PRODUCTO := 5;
0053   ELSIF #CONTADOR = #TOTAL - #FRUTAS THEN

```

```
0054     #PRODUCTO := 6;
0055     ELSIF #CONTADOR = #TOTAL THEN
0056     #PRODUCTO := 7;
0057     END_IF;
0058
0059     #ACTUAL := #TOTAL - #CONTADOR;
0060     IF #CONTADOR = #TOTAL THEN
0061     #VACIO := TRUE;
0062     ELSE
0063     #VACIO := FALSE;
0064     END_IF;
0065     (* CONTADORES PARA PILAS INDIVIDUALES *)
0066     IF #PRODUCTO = 1 THEN
0067     "PROD1".CTUD(CU:=FALSE,
0068     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0069     PV := 0,
0070     CV => "DATOSMATRICES".MATRIZDISPONIBLE[1].HUECO);
0071     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[1].HUECO;
0072     ELSIF #PRODUCTO = 2 THEN
0073     "PROD2".CTUD(CU:=FALSE,
0074     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0075     PV := 0,
0076     CV => "DATOSMATRICES".MATRIZDISPONIBLE[2].HUECO);
0077     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[2].HUECO;
0078     ELSIF #PRODUCTO = 3 THEN
0079     "PROD3".CTUD(CU:=FALSE,
0080     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0081     PV := 0,
0082     CV => "DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO);
0083     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO;
0084     ELSIF #PRODUCTO = 4 THEN
0085     "PROD4".CTUD(CU:=FALSE,
0086     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0087     PV := 0,
0088     CV => "DATOSMATRICES".MATRIZDISPONIBLE[4].HUECO);
0089     #CONTADOR := "DATOSMATRICES".MATRIZDISPONIBLE[4].HUECO;
0090     ELSIF #PRODUCTO = 5 THEN
0091     "PROD5".CTUD(CU:=FALSE,
0092     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0093     PV := 0,
0094     CV => "DATOSMATRICES".MATRIZDISPONIBLE[5].HUECO);
0095     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO;
0096     ELSIF #PRODUCTO = 6 THEN
0097     "PROD6".CTUD(CU:=FALSE,
0098     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0099     PV := 0,
0100     CV => "DATOSMATRICES".MATRIZDISPONIBLE[6].HUECO);
0101     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[6].HUECO;
0102     ELSIF #PRODUCTO = 7 THEN
0103     "PROD7".CTUD(CU:=FALSE,
0104     CD := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO -
100,
0105     PV := 0,
0106     CV => "DATOSMATRICES".MATRIZDISPONIBLE[7].HUECO);
0107     #CONTADORPROD := "DATOSMATRICES".MATRIZDISPONIBLE[7].HUECO;
0108     END_IF;
0109
0110     (* SE RESTA DE TOTALES ELPRODUCTO RECIEN GUARDADO *)
0111     "IEC_Counter_0_DB_1".CTUD(CU := #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition
>= #POSY * #PASO - 100 AND NOT #VACIO,
0112     R := #RESET,
0113     PV := 24,
0114     CV => #CONTADOR);
0115
0116     (*SE GUARDA LA LLEGADA*)
0117     IF #LLEGADA AND "EjeX".ActualPosition >= #POSX * #PASO - 100 AND "EjeY".ActualPosition >= #POSY * #PASO - 100 THEN
0118     #AUXLLEGADA := TRUE;
0119     #MAS := TRUE;
0120     (*"DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].X := #POSX;
0121     "DATOSMATRICES".MATRIZVACIOS["DATOSMATRICES".MATRIZDISPONIBLE[0].HUECO].X := #POSX;*)
0122     END_IF;
0123
0124     (* TIEMPO DE ESPERA *)
0125     "LLEGADAPRODUCTO".TON(IN := #AUXLLEGADA,
0126     PT := t#2s);
0127
0128     (* POSICION OBJETIVO DEPENDIENDO DEL MOMENTO *)
0129     IF "LLEGADAPRODUCTO".Q THEN
0130     "DATOSMATRICES".MATRIZALMACEN[REAL_TO_INT(#POSX),REAL_TO_INT(#POSY)] := 0;
0131     #SALIDASX.POS_ABSM := 0.0;
0132     #SALIDASY.POS_ABSM := 0.0;
0133     #AUXLLEGADA := FALSE;
```

```

0134     #MAS := TRUE;
0135     END_IF;
0136     IF "EjeX".ActualPosition < 1 AND "EjeY".ActualPosition < 1 THEN
0137         #SALIDASX.POS_ABSM := #POSX * #PASO;
0138         #SALIDASY.POS_ABSM := #POSY * #PASO;
0139     END_IF;
0140
0141     (* ACCIÓN AL SALIR EL #PRODUCTO DEL TRANSELEVADOR *)
0142     IF #SALIDA THEN
0143         #MAS := FALSE;
0144     END_IF;
0145     IF #HABILITAR AND (#SALIDA OR "LLEGADAPRODUCTO".Q OR #COMIENZA) THEN
0146         #START := TRUE;
0147     ELSE
0148         #START := FALSE;
0149     END_IF;
0150
0151     (* PULSOS PARA #HABILITAR EL EJECUTAR *)
0152     IF #START AND NOT #AUXLLEGADA THEN
0153         #SALIDASX.EX_ABSM := TRUE;
0154         #SALIDASY.EX_ABSM := TRUE;
0155     ELSE
0156         #SALIDASX.EX_ABSM := FALSE;
0157         #SALIDASY.EX_ABSM := FALSE;
0158     END_IF;
0159 END_IF;

```

Symbol	Address	Type	Comment
"DATOSMATRICES".MATRIZALMACEN[*,*]		Int	EN CADA POSICION SE GUARDA EL PRODUCTO ALMACENADO
"DATOSMATRICES".MATRIZDISPONIBLE[1].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[2].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[3].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[4].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[5].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[6].HUECO		Int	
"DATOSMATRICES".MATRIZDISPONIBLE[7].HUECO		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POSI[*].X		Int	
"DATOSMATRICES".MATRIZPRODUCTOS[*].POSI[*].Y		Int	
"EjeX".ActualPosition		Real	Actual position of the axis
"EjeY".ActualPosition		Real	Actual position of the axis
"INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM		Bool	
"INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM		Bool	
"LLEGADAPRODUCTO".Q		Bool	
#ACTUAL		Int	
#AUXLLEGADA		Bool	
#BEBIDAS		Int	
#CARNES		Int	
#COMIENZA		Bool	
#CONSERVAS		Int	
#CONTADOR		Int	
#CONTADORPROD		Int	
#ENABLE_DESCARGA		Bool	
#FRUTAS		Int	
#HABILITAR		Bool	
#LACTEOS		Int	
#LLEGADA		Bool	
#MAS		Bool	
#PASO		Real	
#PASTA		Int	
#PEDIDO		Int	
#PEDIDOS[*].BEBIDAS		Int	
#PEDIDOS[*].CARNES		Int	
#PEDIDOS[*].CONSERVAS		Int	
#PEDIDOS[*].FRUTAS		Int	
#PEDIDOS[*].LACTEOS		Int	
#PEDIDOS[*].PASTA		Int	
#PEDIDOS[*].VERDURAS		Int	
#POSX		Real	
#POSY		Real	
#PRODUCTO		Int	
#RESET		Bool	
#SALIDA		Bool	
#SALIDASX.EN_POW		Bool	
#SALIDASX.EX_ABSM		Bool	
#SALIDASX.POS_ABSM		Real	
#SALIDASX.VEL_ABSM		Real	
#SALIDASY.EN_POW		Bool	

Symbol	Address	Type	Comment
#SALIDASY.EX_ABSM		Bool	
#SALIDASY.POS_ABSM		Real	
#SALIDASY.VEL_ABSM		Real	
#START		Bool	
#TOTAL		Int	
#VACIO		Bool	
#VERDURAS		Int	

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
InOut									
Static									
Enable	Bool	false	Non-retain	True	True	True	False		Enable axis
StartMode	Int	1	Non-retain	True	True	True	False		Start mode on enabling of axis
StopMode	Int	0	Non-retain	True	True	True	False		Stop mode when disabling axis
▼ Output									
Status	Bool	false	Non-retain	True	True	True	False		Axis is enabled
Busy	Bool	false	Non-retain	True	True	True	False		MC_Power is active
Error	Bool	false	Non-retain	True	True	True	False		Error at MC_Power or associated technology object
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_MoveJog_Instance	MC_MoveJog			True	True	True	True		
▼ Input									
▼ Axis	TO_SpeedAxis			False	False	False	False		Used technology object "Axis"
▼ Base	TO_Axis			False	False	False	False		
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of technology object
Version-Maj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Version-Min	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
Static									
JogForward	Bool	false	Non-retain	True	True	True	False		Jogging in positive direction
JogBackward	Bool	false	Non-retain	True	True	True	False		Jogging in negative direction
Velocity	Real	10.0	Non-retain	True	True	True	False		Target velocity of axis
PositionControlled	Bool	true	Non-retain	True	True	True	False		Position-controlled mode
▼ Output									
InVelocity	Bool	false	Non-retain	True	True	True	False		Target velocity of axis was reached
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
CommandAborted	Bool	false	Non-retain	True	True	True	False		Job was cancelled
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_MoveAbsolute_Instance	MC_MoveAbsolute			True	True	True	True		
▼ Input									
▼ Axis	TO_PositioningAxis			False	False	False	False		Used technology object "Axis"

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
▼ Base	TO_SpeedAxis			False	False	False	False		
▼ Base	TO_Axis			False	False	False	False		
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of tech-nology object
Version-Maj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Version-Min	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Re-served 1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Re-served 2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
▼ Static									
Position	Real		Non-retain	False	False	False	False		Current axis position
Velocity	Real		Non-retain	False	False	False	False		Current axis velocity
ActualPosition	Real		Non-retain	False	False	False	False		Actual position of the axis
ActualVelocity	Real		Non-retain	False	False	False	False		Actual velocity of the axis
▼ Actor	TO_Struct_Ac-tor		Non-retain	False	False	False	False		Configuration data of drive connection
Type	DInt		Non-retain	False	False	False	False		Type of drive connection
InverseDirec-tion	Bool		Non-retain	False	False	False	False		Invert direction signal
DirectionMode	Int		Non-retain	False	False	False	False		Permitted direction of rota-tion
DataAdaption	DInt		Non-retain	False	False	False	False		Automatic transfer of drive parameters to the CPU
PTOSliceTime	Int		Non-retain	False	False	False	False		Segment time for PTO
▼ Interface	TO_Struct_Ac-torInterface		Non-retain	False	False	False	False		Configuration data of drive interface
▼ AddressIn	VREF		Non-retain	False	False	False	False		Drive telegram (internal pa-rameter)
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ AddressOut	VREF		Non-retain	False	False	False	False		Drive telegram (internal pa-rameter)
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ EnableDri-veOutput	VREF		Non-retain	False	False	False	False		Enable output
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ DriveRea-dyInput	VREF		Non-retain	False	False	False	False		Ready input
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
DB_NUM-BER	UInt		Non-retain	False	False	False	False			
OFFSET	UDInt		Non-retain	False	False	False	False			
PTO	DWord		Non-retain	False	False	False	False		Pulse output	
▼ DriveParameter	TO_Struct_Ac-torDriveParam-eter		Non-retain	False	False	False	False		Configuration data of drive parameters	
Reference-Speed	Real		Non-retain	False	False	False	False		Enter reference speed of drive	
MaxSpeed	Real		Non-retain	False	False	False	False		Enter maximum drive speed	
PulsesPerDriveRevolution	DInt		Non-retain	False	False	False	False		Pulses per motor revolution	
▼ Sensor	Array[1..1] of TO_Struct_Sensor		Non-retain	False	False	False	False		Configuration data of encoder	
▼ Sensor[1]	TO_Struct_Sensor		Non-retain	False	False	False	False		Configuration data of encoder	
Type	DInt		Non-retain	False	False	False	False		Encoder type (internal parameter)	
InverseDirec-tion	Bool		Non-retain	False	False	False	False		Invert rotation direction of encoder signals	
System	DInt		Non-retain	False	False	False	False		Encoder system	
Mounting-Mode	DInt		Non-retain	False	False	False	False		Select encoder mounting type	
DataAdap-tion	DInt		Non-retain	False	False	False	False		Automatic transfer of encoder parameters to the CPU	
▼ Interface	TO_Struct_SensorInterface		Non-retain	False	False	False	False		Configuration data of encoder interface	
▼ AddressIn	VREF		Non-retain	False	False	False	False		Encoder telegram (internal parameter)	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
▼ Address-Out	VREF		Non-retain	False	False	False	False		Encoder telegram (internal parameter)	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
Type	DInt		Non-retain	False	False	False	False		Encoder connection (internal parameter)	
HSC	DWord		Non-retain	False	False	False	False		High speed counter	
Number	UDInt		Non-retain	False	False	False	False		Encoder number	
▼ Parameter	TO_Struct_SensorParameter		Non-retain	False	False	False	False		Configuration data of encoder	
Resolu-tion	Real		Non-retain	False	False	False	False		Distance between two increments	
StepsPer-Revolution	UDInt		Non-retain	False	False	False	False		Number of encoder steps per revolution	
FineReso-lutionX-ist1	UDInt		Non-retain	False	False	False	False		Number of bits for fine resolution in incremental actual value (Gn_XIST1)	
FineReso-lutionX-ist2	UDInt		Non-retain	False	False	False	False		Bits in abs. actual value (GN_XIST2)	
Determi-nableRevolutions	UDInt		Non-retain	False	False	False	False		Number of encoder steps per revolution	
Distance-PerRevo-lution	Real		Non-retain	False	False	False	False		Load revolutions per number of motor revolutions	
Behav-iourGx_XIST1	DInt		Non-retain	False	False	False	False		Evaluation of the bits Gx_XIST1	
▼ ActiveHoming	TO_Struct_SensorActiveHoming		Non-retain	False	False	False	False		Configuration data for active homing	
Mode	DInt		Non-retain	False	False	False	False		Active homing mode	
SideInput	Bool		Non-retain	False	False	False	False		Side of reference point switch to which homing is executed with "active homing"	
▼ DigitalIn-putAd-dress	VREF		Non-retain	False	False	False	False		Input address of homing switch	

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
HomePositionOffset	Real		Non-retain	False	False	False	False		Home position offset	
SwitchLevel	Bool		Non-retain	False	False	False	False		Selection of signal level pending with approached homing switch at the CPU input	
▼ PassiveHoming	TO_Struct_SensorPassiveHoming		Non-retain	False	False	False	False		Configuration data for pas-sive homing	
Mode	DInt		Non-retain	False	False	False	False		Passive homing mode	
SideInput	Bool		Non-retain	False	False	False	False		Side of reference point switch to which homing is executed with "passive homing"	
▼ DigitalInputAddress	VREF		Non-retain	False	False	False	False		Input address of homing switch	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
SwitchLevel	Bool		Non-retain	False	False	False	False		Selection of signal level pending with approached homing switch at the CPU input	
▼ Units	TO_Struct_Units		Non-retain	False	False	False	False		Unit of measurement con-figuration data	
LengthUnit	Int		Non-retain	False	False	False	False		Unit of measurement of pa-rameters	
▼ Mechanics	TO_Struct_Mechanics		Non-retain	False	False	False	False		Configuration data of me-chanics	
LeadScrew	Real		Non-retain	False	False	False	False		Distance per motor revolu-tion	
▼ DynamicLimits	TO_Struct_Dy-namicLimits		Non-retain	False	False	False	False		Configuration data of veloci-ty limits	
MaxVelocity	Real		Non-retain	False	False	False	False		Maximum velocity of axis	
MinVelocity	Real		Non-retain	False	False	False	False		Start / stop velocity of axis	
▼ DynamicDefaults	TO_Struct_Dy-namicDefaults		Non-retain	False	False	False	False		Configuration data of dy-namic settings	
Acceleration	Real		Non-retain	False	False	False	False		Acceleration of axis	
Deceleration	Real		Non-retain	False	False	False	False		Deceleration of axis	
EmergencyDe-celeration	Real		Non-retain	False	False	False	False		Emergency stop decelera-tion of axis	
Jerk	Real		Non-retain	False	False	False	False		Jerk of axis	
▼ Modulo	TO_Struct_Mod-ulo		Non-retain	False	False	False	False		Configuration data	
Enable	Bool		Non-retain	False	False	False	False		Enable modulo property of axis	
StartValue	Real		Non-retain	False	False	False	False		Define start value of modulo area	
Length	Real		Non-retain	False	False	False	False		Define length of modulo area	
▼ PositionLimits_SW	TO_Struct_Pos-itionLimitsSW		Non-retain	False	False	False	False		Configuration data of soft-ware limit switches	
Active	Bool		Non-retain	False	False	False	False		Enable software limit switches	
MinPosition	Real		Non-retain	False	False	False	False		Position of lower software limit switch	
MaxPosition	Real		Non-retain	False	False	False	False		Position of upper software limit switch	
▼ PositionLimits_HW	TO_Struct_Pos-itionLimitsHW		Non-retain	False	False	False	False		Configuration data of hard-ware limit switches	
Active	Bool		Non-retain	False	False	False	False		Enable hardware limit switches	
MinSwitchLevel	Bool		Non-retain	False	False	False	False		Voltage level at which the lower hardware limit switch reports "approach"	
▼ MinSwitchAd-dress	VREF		Non-retain	False	False	False	False		Input address of lower hard-ware limit switch	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFFSET	UDInt		Non-retain	False	False	False	False			

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
MaxSwitchLevel	Bool		Non-retain	False	False	False	False		Voltage level at which the upper hardware limit switch reports "approach"	
▼ MaxSwitchAd-dress	VREF		Non-retain	False	False	False	False		Input address of upper hard-ware limit switch	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFFSET	UDInt		Non-retain	False	False	False	False			
▼ Homing	TO_Struct_Hom-ing		Non-retain	False	False	False	False		Configuration data for hom-ing	
AutoReversal	Bool		Non-retain	False	False	False	False		Enable direction reversal on the hardware limit switch	
ApproachDirec-tion	Bool		Non-retain	False	False	False	False		Direction of approach and homing direction of axis	
ApproachVeloc-ity	Real		Non-retain	False	False	False	False		Approach velocity of axis	
ReferencingVe-locity	Real		Non-retain	False	False	False	False		Homing velocity of axis	
▼ PositionControl	TO_Struct_Pos-itionControl		Non-retain	False	False	False	False		Configuration data	
Kv	Real		Non-retain	False	False	False	False		Gain Kv of position control loop	
Kpc	Real		Non-retain	False	False	False	False		Precontrol of position con-trol loop	
▼ FollowingError	TO_Struct_Fol-lowingError		Non-retain	False	False	False	False		Configuration data	
EnableMonitor-ing	Bool		Non-retain	False	False	False	False		Enable following error moni-toring	
MinValue	Real		Non-retain	False	False	False	False		Permitted following error for small velocities	
MaxValue	Real		Non-retain	False	False	False	False		Following error for maxi-mum velocity	
MinVelocity	Real		Non-retain	False	False	False	False		Velocity as of which the fol-lowing error is to be adapted dynamically	
▼ PositioningMoni-toring	TO_Struct_Pos-itioningMonitor-ing		Non-retain	False	False	False	False		Configuration data	
ToleranceTime	Real		Non-retain	False	False	False	False		Tolerance time in which the current position value must reach the positioning win-dow	
MinDwellTime	Real		Non-retain	False	False	False	False		Minimum dwell time in posi-tioning window	
Window	Real		Non-retain	False	False	False	False		Size of the window in which the current value must be located	
▼ StandstillSignal	TO_Struct_Stan-dstillSignal		Non-retain	False	False	False	False		Configuration data	
VelocityThres-hold	Real		Non-retain	False	False	False	False		Size of the standstill window	
MinDwellTime	Real		Non-retain	False	False	False	False		Minimum dwell time in standstill window	
▼ Simulation	TO_Struct_Sim-ulation		Non-retain	False	False	False	False		Configuration data	
Mode	UDInt		Non-retain	False	False	False	False		Simulation mode	
▼ StatusPositioning	TO_Struct_Sta-tusPositioning		Non-retain	False	False	False	False		Current positioning status	
Distance	Real		Non-retain	False	False	False	False		Current distance of axis to target position	
TargetPosition	Real		Non-retain	False	False	False	False		Target position of axis	
FollowingError	Real		Non-retain	False	False	False	False		Current axis following error	
▼ StatusDrive	TO_Struct_Sta-tusDrive		Non-retain	False	False	False	False		Status of drive	
InOperation	Bool		Non-retain	False	False	False	False		Operational status of the drive	
Communicatio-nOK	Bool		Non-retain	False	False	False	False		Cyclic BUS communication between controller and drive	
AdaptionState	DInt		Non-retain	False	False	False	False		Transfer status of the drive	
▼ StatusSensor	Array[1..1] of TO_Struct_Sta-tusSensor		Non-retain	False	False	False	False		Status of encoder	
▼ StatusSensor[1]	TO_Struct_Sta-tusSensor		Non-retain	False	False	False	False		Status of encoder	
State	DInt		Non-retain	False	False	False	False		Status of the encoder value	
Communica-tionOK	Bool		Non-retain	False	False	False	False		Cyclic BUS communication between controller and en-coder	
AbsEncoder-Offset	Real		Non-retain	False	False	False	False		Home position offset for val-ue of an absolute value en-coder	

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
Adaption-State	DInt		Non-retain	False	False	False	False		Transfer status of the encoder
▼ StatusBits	TO_Struct_StatusBits		Non-retain	False	False	False	False		Status information of axis
Activated	Bool		Non-retain	False	False	False	False		Axis is activated
Enable	Bool		Non-retain	False	False	False	False		Axis is enabled
AxisSimulation	Bool		Non-retain	False	False	False	False		Simulation activated
NonPosition-Controlled	Bool		Non-retain	False	False	False	False		Position control deactivated
HomingDone	Bool		Non-retain	False	False	False	False		Axis is homed
Done	Bool		Non-retain	False	False	False	False		No Motion Control job is active on the axis
Error	Bool		Non-retain	False	False	False	False		An error has occurred on the axis
Standstill	Bool		Non-retain	False	False	False	False		Axis is at a standstill
Positioning-Command	Bool		Non-retain	False	False	False	False		Axis executes a positioning job
VelocityCom-mand	Bool		Non-retain	False	False	False	False		Axis executes a job with velocity specification
HomingCom-mand	Bool		Non-retain	False	False	False	False		Axis executes a homing job
CommandTa-bleActive	Bool		Non-retain	False	False	False	False		Command table is being processed
ConstantVeloc-ity	Bool		Non-retain	False	False	False	False		Axis traverses with a constant velocity
Accelerating	Bool		Non-retain	False	False	False	False		Axis is accelerating
Decelerating	Bool		Non-retain	False	False	False	False		Axis is decelerating
ControlPanelAc-tive	Bool		Non-retain	False	False	False	False		"Manual control" mode was enabled in the axis control panel
DriveReady	Bool		Non-retain	False	False	False	False		Drive is ready
RestartRequired	Bool		Non-retain	False	False	False	False		Restart required
SWLimitMinAc-tive	Bool		Non-retain	False	False	False	False		Status of lower software limit switch
SWLimitMaxAc-tive	Bool		Non-retain	False	False	False	False		Status of upper software limit switch
HWLimitMinAc-tive	Bool		Non-retain	False	False	False	False		Status of lower hardware limit switch
HWLimitMaxAc-tive	Bool		Non-retain	False	False	False	False		Status of upper hardware limit switch
▼ ErrorBits	TO_Struct_ErrorBits		Non-retain	False	False	False	False		Error information of axis
SystemFault	Bool		Non-retain	False	False	False	False		Internal system error
ConfigFault	Bool		Non-retain	False	False	False	False		Faulty configuration of axis
DriveFault	Bool		Non-retain	False	False	False	False		Drive has displayed an error due to failure of drive-ready signal
SWLimit	Bool		Non-retain	False	False	False	False		Software limit switch approached or overtraveled
HWLimit	Bool		Non-retain	False	False	False	False		Hardware limit switch approached or overtraveled
DirectionFault	Bool		Non-retain	False	False	False	False		Invalid movement direction
HWUsed	Bool		Non-retain	False	False	False	False		Another axis is using the same PTO (Pulse Train Output) and is enabled
SensorFault	Bool		Non-retain	False	False	False	False		Error in encoder system
Communica-tionFault	Bool		Non-retain	False	False	False	False		Communication with a connected device is faulty
FollowingError	Bool		Non-retain	False	False	False	False		The maximum permitted following error has been exceeded
PositioningFault	Bool		Non-retain	False	False	False	False		The positioning axis was not positioned correctly at the end of a positioning motion
AdaptionError	Bool		Non-retain	False	False	False	False		The transfer of the drive or encoder parameters failed
▼ ControlPanel	TO_Struct_ControlPanel		Non-retain	False	False	False	False		Parameters of axis control table
▼ Input	TO_Struct_ControlPanelInput		Non-retain	False	False	False	False		Input parameters of axis control panel
TimeOut	DInt		Non-retain	False	False	False	False		Internal parameter; please do not change!
EsLifeSign	DInt		Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ Command	Array[1..1] of TO_Struct_ControlPanelInputCmd		Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ Com-mand[1]	TO_Struct_ControlPanelInputCmd		Non-retain	False	False	False	False		Internal parameter; please do not change!

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
InOut									
Static									
Execute	Bool	false	Non-retain	True	True	True	False		Start command
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		Job is completed
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
CommandAborted	Bool	false	Non-retain	True	True	True	False		Job was cancelled
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_Reset_Instance	MC_Reset			True	True	True	True		
▼ Input									
▼ Axis	TO_Axis			False	False	False	False		Used technology object "Axis"
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of technology object
VersionMaj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
VersionMin	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Execute	Bool	false	Non-retain	True	True	True	False		Start command
Restart	Bool	false	Non-retain	True	True	True	False		Execute axis restart
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		Job is completed
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
Temp									
Constant									

```

0001
0002 #MC_Power_Instance(Axis:="EjeX",
0003     Enable:=#INPUTAXIS1.EN_POW,
0004     StartMode:=1,
0005     StopMode:=0,
0006     Error=>#OUTPUTAXIS1.ERROR_POW,
0007     ErrorID=>#OUTPUTAXIS1.ID_POW,
0008     ErrorInfo=>#OUTPUTAXIS1.INFO_POW);
0009
0010 #MC_MoveJog_Instance(Axis := "EjeX",
0011     JogForward := #INPUTAXIS1.JOGF,
0012     JogBackward := #INPUTAXIS1.JOGR,
0013     Velocity := 100.0,
0014     PositionControlled := TRUE,
0015     Error => #OUTPUTAXIS1.ERROR_JOG,
0016     ErrorID => #OUTPUTAXIS1.ID_JOG,

```

```

0017         ErrorInfo => #OUTPUTAXIS1.INFO_JOG);
0018
0019 #MC_MoveAbsolute_Instance(Axis := "EjeX",
0020         Execute := #INPUTAXIS1.EX_ABSM,
0021         Position := #INPUTAXIS1.POS_ABSM,
0022         Velocity := #INPUTAXIS1.VEL_ABSM,
0023         Direction := 0,
0024         Done => #OUTPUTAXIS1.DONE_ABSM,
0025         Error => #OUTPUTAXIS1.ERROR_ABSM,
0026         ErrorID => #OUTPUTAXIS1.ID_ABSM,
0027         ErrorInfo => #OUTPUTAXIS1.INFO_ABSM);
0028
0029 #MC_Home_Instance(Axis := "EjeX",
0030         Execute := #INPUTAXIS1.EX_HOME,
0031         Position := 0.0,
0032         Mode := 7,
0033         Done => #OUTPUTAXIS1.DONE_HOME,
0034         Error => #OUTPUTAXIS1.ERROR_HOME,
0035         ErrorID => #OUTPUTAXIS1.ID_HOME,
0036         ErrorInfo => #OUTPUTAXIS1.INFO_HOME);
0037
0038 #MC_Halt_Instance(Axis := "EjeX",
0039         Execute := #INPUTAXIS1.EX_HALT,
0040         Done => #OUTPUTAXIS1.DONE_HALT,
0041         Error => #OUTPUTAXIS1.ERROR_HALT,
0042         ErrorID => #OUTPUTAXIS1.ID_HALT,
0043         ErrorInfo => #OUTPUTAXIS1.INFO_HALT);
0044
0045 #MC_Reset_Instance(Axis := "EjeX",
0046         Execute := #INPUTAXIS1.EX_RESET,
0047         Restart := #INPUTAXIS1.RESTART_RESET,
0048         Done => #OUTPUTAXIS1.DONE_RESET,
0049         Error => #OUTPUTAXIS1.ERROR_RESET,
0050         ErrorID => #OUTPUTAXIS1.ID_RESET,
0051         ErrorInfo => #OUTPUTAXIS1.INFO_RESET);
0052
0053

```

Symbol	Address	Type	Comment
#INPUTAXIS1.EN_POW		Bool	
#INPUTAXIS1.EX_ABSM		Bool	
#INPUTAXIS1.EX_HALT		Bool	
#INPUTAXIS1.EX_HOME		Bool	
#INPUTAXIS1.EX_RESET		Bool	
#INPUTAXIS1.JOGF		Bool	
#INPUTAXIS1.JOGREV		Bool	
#INPUTAXIS1.POS_ABSM		Real	
#INPUTAXIS1.RESTART_RESET		Bool	
#INPUTAXIS1.VEL_ABSM		Real	
#MC_Halt_Instance		Multi_FB	
#MC_Home_Instance		Multi_FB	
#MC_MoveAbsolute_Instance		Multi_FB	
#MC_MoveJog_Instance		Multi_FB	
#MC_Power_Instance		Multi_FB	
#MC_Reset_Instance		Multi_FB	
#OUTPUTAXIS1.DONE_ABSM		Bool	
#OUTPUTAXIS1.DONE_HALT		Bool	
#OUTPUTAXIS1.DONE_HOME		Bool	
#OUTPUTAXIS1.DONE_RESET		Bool	
#OUTPUTAXIS1.ERROR_ABSM		Bool	
#OUTPUTAXIS1.ERROR_HALT		Bool	
#OUTPUTAXIS1.ERROR_HOME		Bool	
#OUTPUTAXIS1.ERROR_JOG		Bool	
#OUTPUTAXIS1.ERROR_POW		Bool	
#OUTPUTAXIS1.ERROR_RESET		Bool	
#OUTPUTAXIS1.ID_ABSM		Word	
#OUTPUTAXIS1.ID_HALT		Word	
#OUTPUTAXIS1.ID_HOME		Word	
#OUTPUTAXIS1.ID_JOG		Word	
#OUTPUTAXIS1.ID_POW		Word	
#OUTPUTAXIS1.ID_RESET		Word	
#OUTPUTAXIS1.INFO_ABSM		Word	
#OUTPUTAXIS1.INFO_HALT		Word	
#OUTPUTAXIS1.INFO_HOME		Word	
#OUTPUTAXIS1.INFO_JOG		Word	
#OUTPUTAXIS1.INFO_POW		Word	
#OUTPUTAXIS1.INFO_RESET		Word	

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
InOut									
Static									
Enable	Bool	false	Non-retain	True	True	True	False		Enable axis
StartMode	Int	1	Non-retain	True	True	True	False		Start mode on enabling of axis
StopMode	Int	0	Non-retain	True	True	True	False		Stop mode when disabling axis
▼ Output									
Status	Bool	false	Non-retain	True	True	True	False		Axis is enabled
Busy	Bool	false	Non-retain	True	True	True	False		MC_Power is active
Error	Bool	false	Non-retain	True	True	True	False		Error at MC_Power or associated technology object
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_MoveJog_Instance	MC_MoveJog			True	True	True	True		
▼ Input									
▼ Axis	TO_SpeedAxis			False	False	False	False		Used technology object "Axis"
▼ Base	TO_Axis			False	False	False	False		
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of technology object
Version-Maj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Version-Min	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
Static									
JogForward	Bool	false	Non-retain	True	True	True	False		Jogging in positive direction
JogBackward	Bool	false	Non-retain	True	True	True	False		Jogging in negative direction
Velocity	Real	10.0	Non-retain	True	True	True	False		Target velocity of axis
PositionControlled	Bool	true	Non-retain	True	True	True	False		Position-controlled mode
▼ Output									
InVelocity	Bool	false	Non-retain	True	True	True	False		Target velocity of axis was reached
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
CommandAborted	Bool	false	Non-retain	True	True	True	False		Job was cancelled
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_MoveAbsolute_Instance	MC_MoveAbsolute			True	True	True	True		
▼ Input									
▼ Axis	TO_PositioningAxis			False	False	False	False		Used technology object "Axis"

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
▼ Base	TO_SpeedAxis			False	False	False	False		
▼ Base	TO_Axis			False	False	False	False		
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of tech-nology object
Version-Maj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Version-Min	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Re-served 1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Re-served 2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
Static									
Input									
Output									
InOut									
▼ Static									
Position	Real		Non-retain	False	False	False	False		Current axis position
Velocity	Real		Non-retain	False	False	False	False		Current axis velocity
ActualPosition	Real		Non-retain	False	False	False	False		Actual position of the axis
ActualVelocity	Real		Non-retain	False	False	False	False		Actual velocity of the axis
▼ Actor	TO_Struct_Ac-tor		Non-retain	False	False	False	False		Configuration data of drive connection
Type	DInt		Non-retain	False	False	False	False		Type of drive connection
InverseDirec-tion	Bool		Non-retain	False	False	False	False		Invert direction signal
DirectionMode	Int		Non-retain	False	False	False	False		Permitted direction of rota-tion
DataAdaption	DInt		Non-retain	False	False	False	False		Automatic transfer of drive parameters to the CPU
PTOSliceTime	Int		Non-retain	False	False	False	False		Segment time for PTO
▼ Interface	TO_Struct_Ac-torInterface		Non-retain	False	False	False	False		Configuration data of drive interface
▼ AddressIn	VREF		Non-retain	False	False	False	False		Drive telegram (internal pa-rameter)
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ AddressOut	VREF		Non-retain	False	False	False	False		Drive telegram (internal pa-rameter)
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ EnableDri-veOutput	VREF		Non-retain	False	False	False	False		Enable output
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUM-BER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ DriveRea-dyInput	VREF		Non-retain	False	False	False	False		Ready input
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
DB_NUM-BER	UInt		Non-retain	False	False	False	False			
OFFSET	UDInt		Non-retain	False	False	False	False			
PTO	DWord		Non-retain	False	False	False	False		Pulse output	
▼ DriveParameter	TO_Struct_Ac-torDriveParame-ter		Non-retain	False	False	False	False		Configuration data of drive parameters	
Reference-Speed	Real		Non-retain	False	False	False	False		Enter reference speed of drive	
MaxSpeed	Real		Non-retain	False	False	False	False		Enter maximum drive speed	
PulsesPerDriveRevolution	DInt		Non-retain	False	False	False	False		Pulses per motor revolution	
▼ Sensor	Array[1..1] of TO_Struct_Sensor		Non-retain	False	False	False	False		Configuration data of encoder	
▼ Sensor[1]	TO_Struct_Sensor		Non-retain	False	False	False	False		Configuration data of encoder	
Type	DInt		Non-retain	False	False	False	False		Encoder type (internal parameter)	
InverseDirec-tion	Bool		Non-retain	False	False	False	False		Invert rotation direction of encoder signals	
System	DInt		Non-retain	False	False	False	False		Encoder system	
Mounting-Mode	DInt		Non-retain	False	False	False	False		Select encoder mounting type	
DataAdap-tion	DInt		Non-retain	False	False	False	False		Automatic transfer of encoder parameters to the CPU	
▼ Interface	TO_Struct_SensorInterface		Non-retain	False	False	False	False		Configuration data of encoder interface	
▼ AddressIn	VREF		Non-retain	False	False	False	False		Encoder telegram (internal parameter)	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
▼ Address-Out	VREF		Non-retain	False	False	False	False		Encoder telegram (internal parameter)	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
Type	DInt		Non-retain	False	False	False	False		Encoder connection (internal parameter)	
HSC	DWord		Non-retain	False	False	False	False		High speed counter	
Number	UDInt		Non-retain	False	False	False	False		Encoder number	
▼ Parameter	TO_Struct_SensorParameter		Non-retain	False	False	False	False		Configuration data of encoder	
Resolu-tion	Real		Non-retain	False	False	False	False		Distance between two increments	
StepsPer-Revolution	UDInt		Non-retain	False	False	False	False		Number of encoder steps per revolution	
FineReso-lutionX-ist1	UDInt		Non-retain	False	False	False	False		Number of bits for fine resolution in incremental actual value (Gn_XIST1)	
FineReso-lutionX-ist2	UDInt		Non-retain	False	False	False	False		Bits in abs. actual value (GN_XIST2)	
Determi-nableRevolutions	UDInt		Non-retain	False	False	False	False		Number of encoder steps per revolution	
Distance-PerRevo-lution	Real		Non-retain	False	False	False	False		Load revolutions per number of motor revolutions	
Behav-iourGx_XIST1	DInt		Non-retain	False	False	False	False		Evaluation of the bits Gx_XIST1	
▼ ActiveHoming	TO_Struct_SensorActiveHoming		Non-retain	False	False	False	False		Configuration data for active homing	
Mode	DInt		Non-retain	False	False	False	False		Active homing mode	
SideInput	Bool		Non-retain	False	False	False	False		Side of reference point switch to which homing is executed with "active homing"	
▼ DigitalIn-putAd-dress	VREF		Non-retain	False	False	False	False		Input address of homing switch	

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
HomePositionOffset	Real		Non-retain	False	False	False	False		Home position offset	
SwitchLevel	Bool		Non-retain	False	False	False	False		Selection of signal level pending with approached homing switch at the CPU input	
▼ PassiveHoming	TO_Struct_SensorPassiveHoming		Non-retain	False	False	False	False		Configuration data for pas-sive homing	
Mode	DInt		Non-retain	False	False	False	False		Passive homing mode	
SideInput	Bool		Non-retain	False	False	False	False		Side of reference point switch to which homing is executed with "passive homing"	
▼ DigitalInputAddress	VREF		Non-retain	False	False	False	False		Input address of homing switch	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFF-SET	UDInt		Non-retain	False	False	False	False			
SwitchLevel	Bool		Non-retain	False	False	False	False		Selection of signal level pending with approached homing switch at the CPU input	
▼ Units	TO_Struct_Units		Non-retain	False	False	False	False		Unit of measurement con-figuration data	
LengthUnit	Int		Non-retain	False	False	False	False		Unit of measurement of pa-rameters	
▼ Mechanics	TO_Struct_Mechanics		Non-retain	False	False	False	False		Configuration data of me-chanics	
LeadScrew	Real		Non-retain	False	False	False	False		Distance per motor revolu-tion	
▼ DynamicLimits	TO_Struct_Dy-namicLimits		Non-retain	False	False	False	False		Configuration data of veloci-ty limits	
MaxVelocity	Real		Non-retain	False	False	False	False		Maximum velocity of axis	
MinVelocity	Real		Non-retain	False	False	False	False		Start / stop velocity of axis	
▼ DynamicDefaults	TO_Struct_Dy-namicDefaults		Non-retain	False	False	False	False		Configuration data of dy-namic settings	
Acceleration	Real		Non-retain	False	False	False	False		Acceleration of axis	
Deceleration	Real		Non-retain	False	False	False	False		Deceleration of axis	
EmergencyDe-celeration	Real		Non-retain	False	False	False	False		Emergency stop decelera-tion of axis	
Jerk	Real		Non-retain	False	False	False	False		Jerk of axis	
▼ Modulo	TO_Struct_Mod-ulo		Non-retain	False	False	False	False		Configuration data	
Enable	Bool		Non-retain	False	False	False	False		Enable modulo property of axis	
StartValue	Real		Non-retain	False	False	False	False		Define start value of modulo area	
Length	Real		Non-retain	False	False	False	False		Define length of modulo area	
▼ PositionLimits_SW	TO_Struct_Pos-itionLimitsSW		Non-retain	False	False	False	False		Configuration data of soft-ware limit switches	
Active	Bool		Non-retain	False	False	False	False		Enable software limit switches	
MinPosition	Real		Non-retain	False	False	False	False		Position of lower software limit switch	
MaxPosition	Real		Non-retain	False	False	False	False		Position of upper software limit switch	
▼ PositionLimits_HW	TO_Struct_Pos-itionLimitsHW		Non-retain	False	False	False	False		Configuration data of hard-ware limit switches	
Active	Bool		Non-retain	False	False	False	False		Enable hardware limit switches	
MinSwitchLevel	Bool		Non-retain	False	False	False	False		Voltage level at which the lower hardware limit switch reports "approach"	
▼ MinSwitchAd-dress	VREF		Non-retain	False	False	False	False		Input address of lower hard-ware limit switch	
RID	DWord		Non-retain	False	False	False	False			
AREA	Byte		Non-retain	False	False	False	False			
DB_NUMBER	UInt		Non-retain	False	False	False	False			
OFFSET	UDInt		Non-retain	False	False	False	False			

Totally Integrated Automation Portal									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
MaxSwitchLevel	Bool		Non-retain	False	False	False	False		Voltage level at which the upper hardware limit switch reports "approach"
▼ MaxSwitchAd-dress	VREF		Non-retain	False	False	False	False		Input address of upper hard-ware limit switch
RID	DWord		Non-retain	False	False	False	False		
AREA	Byte		Non-retain	False	False	False	False		
DB_NUMBER	UInt		Non-retain	False	False	False	False		
OFFSET	UDInt		Non-retain	False	False	False	False		
▼ Homing	TO_Struct_Hom-ing		Non-retain	False	False	False	False		Configuration data for hom-ing
AutoReversal	Bool		Non-retain	False	False	False	False		Enable direction reversal on the hardware limit switch
ApproachDirec-tion	Bool		Non-retain	False	False	False	False		Direction of approach and homing direction of axis
ApproachVeloc-ity	Real		Non-retain	False	False	False	False		Approach velocity of axis
ReferencingVe-locity	Real		Non-retain	False	False	False	False		Homing velocity of axis
▼ PositionControl	TO_Struct_Positi-onControl		Non-retain	False	False	False	False		Configuration data
Kv	Real		Non-retain	False	False	False	False		Gain Kv of position control loop
Kpc	Real		Non-retain	False	False	False	False		Precontrol of position control loop
▼ FollowingError	TO_Struct_Fol-lowingError		Non-retain	False	False	False	False		Configuration data
EnableMonitor-ing	Bool		Non-retain	False	False	False	False		Enable following error moni-toring
MinValue	Real		Non-retain	False	False	False	False		Permitted following error for small velocities
MaxValue	Real		Non-retain	False	False	False	False		Following error for maxi-mum velocity
MinVelocity	Real		Non-retain	False	False	False	False		Velocity as of which the fol-lowing error is to be adapted dynamically
▼ PositioningMoni-toring	TO_Struct_Positi-oningMonitor-ing		Non-retain	False	False	False	False		Configuration data
ToleranceTime	Real		Non-retain	False	False	False	False		Tolerance time in which the current position value must reach the positioning win-dow
MinDwellTime	Real		Non-retain	False	False	False	False		Minimum dwell time in posi-tioning window
Window	Real		Non-retain	False	False	False	False		Size of the window in which the current value must be located
▼ StandstillSignal	TO_Struct_Stan-dstillSignal		Non-retain	False	False	False	False		Configuration data
VelocityThres-hold	Real		Non-retain	False	False	False	False		Size of the standstill window
MinDwellTime	Real		Non-retain	False	False	False	False		Minimum dwell time in standstill window
▼ Simulation	TO_Struct_Sim-ulation		Non-retain	False	False	False	False		Configuration data
Mode	UDInt		Non-retain	False	False	False	False		Simulation mode
▼ StatusPositioning	TO_Struct_Sta-tusPositioning		Non-retain	False	False	False	False		Current positioning status
Distance	Real		Non-retain	False	False	False	False		Current distance of axis to target position
TargetPosition	Real		Non-retain	False	False	False	False		Target position of axis
FollowingError	Real		Non-retain	False	False	False	False		Current axis following error
▼ StatusDrive	TO_Struct_Sta-tusDrive		Non-retain	False	False	False	False		Status of drive
InOperation	Bool		Non-retain	False	False	False	False		Operational status of the drive
Communicatio-nOK	Bool		Non-retain	False	False	False	False		Cyclic BUS communication between controller and drive
AdaptionState	DInt		Non-retain	False	False	False	False		Transfer status of the drive
▼ StatusSensor	Array[1..1] of TO_Struct_Sta-tusSensor		Non-retain	False	False	False	False		Status of encoder
▼ StatusSensor[1]	TO_Struct_Sta-tusSensor		Non-retain	False	False	False	False		Status of encoder
State	DInt		Non-retain	False	False	False	False		Status of the encoder value
Communica-tionOK	Bool		Non-retain	False	False	False	False		Cyclic BUS communication between controller and en-coder
AbsEncoder-Offset	Real		Non-retain	False	False	False	False		Home position offset for val-ue of an absolute value en-coder

Totally Integrated Automation Portal										
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment	
Adaption-State	DInt		Non-retain	False	False	False	False		Transfer status of the encoder	
▼ StatusBits	TO_Struct_StatusBits		Non-retain	False	False	False	False		Status information of axis	
Activated	Bool		Non-retain	False	False	False	False		Axis is activated	
Enable	Bool		Non-retain	False	False	False	False		Axis is enabled	
AxisSimulation	Bool		Non-retain	False	False	False	False		Simulation activated	
NonPosition-Controlled	Bool		Non-retain	False	False	False	False		Position control deactivated	
HomingDone	Bool		Non-retain	False	False	False	False		Axis is homed	
Done	Bool		Non-retain	False	False	False	False		No Motion Control job is active on the axis	
Error	Bool		Non-retain	False	False	False	False		An error has occurred on the axis	
Standstill	Bool		Non-retain	False	False	False	False		Axis is at a standstill	
Positioning-Command	Bool		Non-retain	False	False	False	False		Axis executes a positioning job	
VelocityCommand	Bool		Non-retain	False	False	False	False		Axis executes a job with velocity specification	
HomingCommand	Bool		Non-retain	False	False	False	False		Axis executes a homing job	
CommandTableActive	Bool		Non-retain	False	False	False	False		Command table is being processed	
ConstantVelocity	Bool		Non-retain	False	False	False	False		Axis traverses with a constant velocity	
Accelerating	Bool		Non-retain	False	False	False	False		Axis is accelerating	
Decelerating	Bool		Non-retain	False	False	False	False		Axis is decelerating	
ControlPanelActive	Bool		Non-retain	False	False	False	False		"Manual control" mode was enabled in the axis control panel	
DriveReady	Bool		Non-retain	False	False	False	False		Drive is ready	
RestartRequired	Bool		Non-retain	False	False	False	False		Restart required	
SWLimitMinActive	Bool		Non-retain	False	False	False	False		Status of lower software limit switch	
SWLimitMaxActive	Bool		Non-retain	False	False	False	False		Status of upper software limit switch	
HWLimitMinActive	Bool		Non-retain	False	False	False	False		Status of lower hardware limit switch	
HWLimitMaxActive	Bool		Non-retain	False	False	False	False		Status of upper hardware limit switch	
▼ ErrorBits	TO_Struct_ErrorBits		Non-retain	False	False	False	False		Error information of axis	
SystemFault	Bool		Non-retain	False	False	False	False		Internal system error	
ConfigFault	Bool		Non-retain	False	False	False	False		Faulty configuration of axis	
DriveFault	Bool		Non-retain	False	False	False	False		Drive has displayed an error due to failure of drive-ready signal	
SWLimit	Bool		Non-retain	False	False	False	False		Software limit switch approached or overtraveled	
HWLimit	Bool		Non-retain	False	False	False	False		Hardware limit switch approached or overtraveled	
DirectionFault	Bool		Non-retain	False	False	False	False		Invalid movement direction	
HWUsed	Bool		Non-retain	False	False	False	False		Another axis is using the same PTO (Pulse Train Output) and is enabled	
SensorFault	Bool		Non-retain	False	False	False	False		Error in encoder system	
CommunicationFault	Bool		Non-retain	False	False	False	False		Communication with a connected device is faulty	
FollowingError	Bool		Non-retain	False	False	False	False		The maximum permitted following error has been exceeded	
PositioningFault	Bool		Non-retain	False	False	False	False		The positioning axis was not positioned correctly at the end of a positioning motion	
AdaptionError	Bool		Non-retain	False	False	False	False		The transfer of the drive or encoder parameters failed	
▼ ControlPanel	TO_Struct_ControlPanel		Non-retain	False	False	False	False		Parameters of axis control table	
▼ Input	TO_Struct_ControlPanelInput		Non-retain	False	False	False	False		Input parameters of axis control panel	
TimeOut	DInt		Non-retain	False	False	False	False		Internal parameter; please do not change!	
EsLifeSign	DInt		Non-retain	False	False	False	False		Internal parameter; please do not change!	
▼ Command	Array[1..1] of TO_Struct_ControlPanelInputCmd		Non-retain	False	False	False	False		Internal parameter; please do not change!	
▼ Command[1]	TO_Struct_ControlPanelInputCmd		Non-retain	False	False	False	False		Internal parameter; please do not change!	

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writ-able from HMI/OPC UA/ Web API	Visible in HMI engi-neering	Setpoint	Supervi-sion	Comment
InOut									
Static									
Execute	Bool	false	Non-retain	True	True	True	False		Start command
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		Job is completed
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
CommandAborted	Bool	false	Non-retain	True	True	True	False		Job was cancelled
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
▼ MC_Reset_Instance	MC_Reset			True	True	True	True		
▼ Input									
▼ Axis	TO_Axis			False	False	False	False		Used technology object "Axis"
▼ Base	TO_AnyMotionObject			False	False	False	False		
Input									
Output									
InOut									
▼ Static									
▼ Header	TO_Struct_Head-er		Non-retain	False	False	False	False		Version information of technology object
VersionMaj	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
VersionMin	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Type	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved1	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Reserved2	Int		Non-retain	False	False	False	False		Internal parameter; please do not change!
Input									
Output									
InOut									
Static									
Execute	Bool	false	Non-retain	True	True	True	False		Start command
Restart	Bool	false	Non-retain	True	True	True	False		Execute axis restart
▼ Output									
Done	Bool	false	Non-retain	True	True	True	False		Job is completed
Busy	Bool	false	Non-retain	True	True	True	False		Job is being executed
Error	Bool	false	Non-retain	True	True	True	False		Error during execution of the job
ErrorID	Word	16#0	Non-retain	True	True	True	False		Error ID for parameter "Error"
ErrorInfo	Word	16#0	Non-retain	True	True	True	False		Error info ID for parameter "ErrorID"
InOut									
▼ Static									
FB_ID	DInt	0	Non-retain	False	False	False	False		Internal parameter; please do not change!
Temp									
Constant									

```

0001
0002 #MC_Power_Instance(Axis := "EjeY",
0003     Enable := #INPUTAXIS2.EN_POW,
0004     StartMode := 1,
0005     StopMode := 0,
0006     Error => #OUTPUTAXIS2.ERROR_POW,
0007     ErrorID => #OUTPUTAXIS2.ID_POW,
0008     ErrorInfo => #OUTPUTAXIS2.INFO_POW);
0009
0010 #MC_MoveJog_Instance(Axis := "EjeY",
0011     JogForward := #INPUTAXIS2.JOGF,
0012     JogBackward := #INPUTAXIS2.JOGR,
0013     Velocity := 100.0,
0014     PositionControlled := TRUE,
0015     Error => #OUTPUTAXIS2.ERROR_JOG,
0016     ErrorID => #OUTPUTAXIS2.ID_JOG,

```

```

0017         ErrorInfo => #OUTPUTAXIS2.INFO_JOG);
0018
0019 #MC_MoveAbsolute_Instance(Axis := "EjeY",
0020         Execute := #INPUTAXIS2.EX_ABSM,
0021         Position := #INPUTAXIS2.POS_ABSM,
0022         Velocity := #INPUTAXIS2.VEL_ABSM,
0023         Direction := 0,
0024         Done => #OUTPUTAXIS2.DONE_ABSM,
0025         Error => #OUTPUTAXIS2.ERROR_ABSM,
0026         ErrorID => #OUTPUTAXIS2.ID_ABSM,
0027         ErrorInfo => #OUTPUTAXIS2.INFO_ABSM);
0028
0029 #MC_Home_Instance(Axis := "EjeY",
0030         Execute := #INPUTAXIS2.EX_HOME,
0031         Position := 0.0,
0032         Mode := 7,
0033         Done => #OUTPUTAXIS2.DONE_HOME,
0034         Error => #OUTPUTAXIS2.ERROR_HOME,
0035         ErrorID => #OUTPUTAXIS2.ID_HOME,
0036         ErrorInfo => #OUTPUTAXIS2.INFO_HOME);
0037
0038 #MC_Halt_Instance(Axis := "EjeY",
0039         Execute := #INPUTAXIS2.EX_HALT,
0040         Done => #OUTPUTAXIS2.DONE_HALT,
0041         Error => #OUTPUTAXIS2.ERROR_HALT,
0042         ErrorID => #OUTPUTAXIS2.ID_HALT,
0043         ErrorInfo => #OUTPUTAXIS2.INFO_HALT);
0044
0045 #MC_Reset_Instance(Axis := "EjeY",
0046         Execute := #INPUTAXIS2.EX_RESET,
0047         Restart := #INPUTAXIS2.RESTART_RESET,
0048         Done => #OUTPUTAXIS2.DONE_RESET,
0049         Error => #OUTPUTAXIS2.ERROR_RESET,
0050         ErrorID => #OUTPUTAXIS2.ID_RESET,
0051         ErrorInfo => #OUTPUTAXIS2.INFO_RESET);
0052

```

Symbol	Address	Type	Comment
#INPUTAXIS2.EN_POW		Bool	
#INPUTAXIS2.EX_ABSM		Bool	
#INPUTAXIS2.EX_HALT		Bool	
#INPUTAXIS2.EX_HOME		Bool	
#INPUTAXIS2.EX_RESET		Bool	
#INPUTAXIS2.JOGF		Bool	
#INPUTAXIS2.JOGREV		Bool	
#INPUTAXIS2.POS_ABSM		Real	
#INPUTAXIS2.RESTART_RESET		Bool	
#INPUTAXIS2.VEL_ABSM		Real	
#MC_Halt_Instance		Multi_FB	
#MC_Home_Instance		Multi_FB	
#MC_MoveAbsolute_Instance		Multi_FB	
#MC_MoveJog_Instance		Multi_FB	
#MC_Power_Instance		Multi_FB	
#MC_Reset_Instance		Multi_FB	
#OUTPUTAXIS2.DONE_ABSM		Bool	
#OUTPUTAXIS2.DONE_HALT		Bool	
#OUTPUTAXIS2.DONE_HOME		Bool	
#OUTPUTAXIS2.DONE_RESET		Bool	
#OUTPUTAXIS2.ERROR_ABSM		Bool	
#OUTPUTAXIS2.ERROR_HALT		Bool	
#OUTPUTAXIS2.ERROR_HOME		Bool	
#OUTPUTAXIS2.ERROR_JOG		Bool	
#OUTPUTAXIS2.ERROR_POW		Bool	
#OUTPUTAXIS2.ERROR_RESET		Bool	
#OUTPUTAXIS2.ID_ABSM		Word	
#OUTPUTAXIS2.ID_HALT		Word	
#OUTPUTAXIS2.ID_HOME		Word	
#OUTPUTAXIS2.ID_JOG		Word	
#OUTPUTAXIS2.ID_POW		Word	
#OUTPUTAXIS2.ID_RESET		Word	
#OUTPUTAXIS2.INFO_ABSM		Word	
#OUTPUTAXIS2.INFO_HALT		Word	
#OUTPUTAXIS2.INFO_HOME		Word	
#OUTPUTAXIS2.INFO_JOG		Word	
#OUTPUTAXIS2.INFO_POW		Word	
#OUTPUTAXIS2.INFO_RESET		Word	

MODOMANUAL [FB3]

MODOMANUAL Properties

General

Name	MODOMANUAL	Number	3	Type	FB	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
HABILITAR	Bool	false	Non-retain	True	True	True	False		
JOG	Bool	false	Non-retain	True	True	True	False		
MANUAL	Bool	false	Non-retain	True	True	True	False		
PASO	Int	40	Non-retain	True	True	True	False		
RETORNO	Bool	false	Non-retain	True	True	True	False		
▼ Output									
▼ SALIDAX									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
▼ SALIDAY									
EN_POW	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGREV	Bool	false	Non-retain	True	True	True	False		
EX_ABSM	Bool	false	Non-retain	True	True	True	False		
POS_ABSM	Real	0.0	Non-retain	True	True	True	False		
VEL_ABSM	Real	0.0	Non-retain	True	True	True	False		
EX_HOME	Bool	false	Non-retain	True	True	True	False		
EX_HALT	Bool	false	Non-retain	True	True	True	False		
EX_RESET	Bool	false	Non-retain	True	True	True	False		
RESTART_RESET	Bool	false	Non-retain	True	True	True	False		
InOut									
▼ Static									
AUXJOG	Bool	false	Non-retain	True	True	True	False		
AUXMANUAL	Bool	false	Non-retain	True	True	True	False		
AUXRETORNO	Bool	false	Non-retain	True	True	True	False		
JOGF	Bool	false	Non-retain	True	True	True	False		
JOGR	Bool	false	Non-retain	True	True	True	False		
JOGF1	Bool	false	Non-retain	True	True	True	False		
JOGR1	Bool	false	Non-retain	True	True	True	False		
RETORNO_1	Bool	false	Non-retain	True	True	True	False		
POSX	Int	0	Non-retain	True	True	True	False		
POSY	Int	0	Non-retain	True	True	True	False		
START	Bool	false	Non-retain	True	True	True	False		
IDA	Bool	false	Non-retain	True	True	True	False		
VUELTA	Bool	false	Non-retain	True	True	True	False		
AUXDONEX	Bool	false	Non-retain	True	True	True	False		
AUXDONEY	Bool	false	Non-retain	True	True	True	False		
LLEGADA	Bool	false	Non-retain	True	True	True	False		
AUXLLEGADA	Bool	false	Non-retain	True	True	True	False		
Temp									
Constant									

```

0001 IF #HABILITAR THEN
0002     #SALIDAX.EN_POW := TRUE;
0003     #SALIDAY.EN_POW := TRUE;
0004
0005     (*ENCLAVAMIENTOS*)
0006     IF #JOG AND NOT #AUXMANUAL AND NOT #AUXRETORNO THEN
0007         #AUXJOG := TRUE;
0008     ELSIF NOT #JOG THEN
0009         #AUXJOG := FALSE;
0010     END_IF;
0011     IF #MANUAL AND NOT #AUXJOG AND NOT #AUXRETORNO THEN

```

```

0012     #AUXMANUAL := TRUE;
0013     ELSIF NOT #MANUAL THEN
0014     #AUXMANUAL := FALSE;
0015     END_IF;
0016     IF #RETORNO AND NOT #AUXJOG AND NOT #AUXMANUAL THEN
0017     #AUXRETORNO := TRUE;
0018     ELSIF NOT #RETORNO THEN
0019     #AUXRETORNO := FALSE;
0020     END_IF;
0021
0022     IF #AUXJOG THEN
0023     #SALIDAX.JOGF := #JOGF;
0024     #SALIDAX.JOGREV := #JOGR;
0025     #SALIDAY.JOGF := #JOGF1;
0026     #SALIDAY.JOGREV := #JOGR1;
0027     ELSIF #AUXMANUAL THEN
0028     #SALIDAX.VEL_ABSM := 100.0;
0029     #SALIDAY.VEL_ABSM := 100.0;
0030     IF #POSX > #POSY THEN
0031     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM;
0032     ELSIF #POSX < #POSY THEN
0033     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM;
0034     ELSE
0035     #LLEGADA := "INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM;
0036     END_IF;
0037     IF #LLEGADA THEN
0038     #AUXLLEGADA := TRUE;
0039     END_IF;
0040     "LLEGADA".TON(IN := #AUXLLEGADA,
0041     PT := t#2s);
0042
0043     IF "LLEGADA".Q THEN
0044     #SALIDAX.POS_ABSM := 0.0;
0045     #SALIDAY.POS_ABSM := 0.0;
0046     #AUXLLEGADA := FALSE;
0047     END_IF;
0048
0049     IF "EjeX".ActualPosition<1 AND "EjeY".ActualPosition<1 THEN
0050     #SALIDAX.POS_ABSM := #PASO*#POSX;
0051     #SALIDAY.POS_ABSM := #PASO*#POSY;
0052     END_IF;
0053     IF #START AND NOT #AUXLLEGADA THEN
0054     #SALIDAX.EX_ABSM := TRUE;
0055     #SALIDAY.EX_ABSM := TRUE;
0056     ELSE
0057     #SALIDAX.EX_ABSM := FALSE;
0058     #SALIDAY.EX_ABSM := FALSE;
0059     END_IF;
0060
0061     ELSIF #AUXRETORNO THEN
0062     #SALIDAX.POS_ABSM := 0.0;
0063     #SALIDAX.VEL_ABSM := 100.0;
0064     #SALIDAY.POS_ABSM := 0.0;
0065     #SALIDAY.VEL_ABSM := 100.0;
0066     #SALIDAX.EX_ABSM := #RETORNO_1;
0067     #SALIDAY.EX_ABSM := #RETORNO_1;
0068     ELSE
0069     #SALIDAX.EN_POW := FALSE;
0070     #SALIDAY.EN_POW := FALSE;
0071     END_IF;
0072 END_IF;
0073
0074
0075

```

Symbol	Address	Type	Comment
"EjeX".ActualPosition		Real	Actual position of the axis
"EjeY".ActualPosition		Real	Actual position of the axis
"INPUTS/OUTPUTS".OUTPUTSX.DONE_ABSM		Bool	
"INPUTS/OUTPUTS".OUTPUTSY.DONE_ABSM		Bool	
"LLEGADA".Q		Bool	
#AUXJOG		Bool	
#AUXLLEGADA		Bool	
#AUXMANUAL		Bool	
#AUXRETORNO		Bool	
#HABILITAR		Bool	
#JOG		Bool	
#JOGF		Bool	
#JOGF1		Bool	
#JOGR		Bool	
#JOGR1		Bool	
#LLEGADA		Bool	
#MANUAL		Bool	
#PASO		Int	
#POSX		Int	
#POSY		Int	

Symbol	Address	Type	Comment
#RETORNO		Bool	
#RETORNO_1		Bool	
#SALIDAX.EN_POW		Bool	
#SALIDAX.EX_ABSM		Bool	
#SALIDAX.JOGF		Bool	
#SALIDAX.JOGREV		Bool	
#SALIDAX.POS_ABSM		Real	
#SALIDAX.VEL_ABSM		Real	
#SALIDAY.EN_POW		Bool	
#SALIDAY.EX_ABSM		Bool	
#SALIDAY.JOGF		Bool	
#SALIDAY.JOGREV		Bool	
#SALIDAY.POS_ABSM		Real	
#SALIDAY.VEL_ABSM		Real	
#START		Bool	



SIMATIC S7-1200, CPU 1215C, CPU compacta DC/DC/DC, 2 puertos PROFINET, E/S INTEGRADAS: 14 DI 24 V DC; 10 DO 24 V DC; 0,5A; 2 AI 0-10V DC, 2 AO 0-20 mA DC, alimentación: DC 20,4-28,8V DC, Memoria de programas/datos 125 KB

Información general	
Designación del tipo de producto	CPU 1215C DC/DC/DC
Versión de firmware	V4.4
Ingeniería con	
<ul style="list-style-type: none"> Paquete de programación 	STEP 7 V16 o superior
Tensión de alimentación	
Valor nominal (DC)	
<ul style="list-style-type: none"> 24 V DC 	Sí
Rango admisible, límite inferior (DC)	20,4 V
Rango admisible, límite superior (DC)	28,8 V
Protección contra inversión de polaridad	Sí
Tensión de carga L+	
<ul style="list-style-type: none"> Valor nominal (DC) Rango admisible, límite inferior (DC) Rango admisible, límite superior (DC) 	24 V 20,4 V 28,8 V
Intensidad de entrada	
Consumo (valor nominal)	500 mA
Consumo, máx.	1 500 mA; CPU con todos los módulos de ampliación
Intensidad de cierre, máx.	12 A; con 28,8 V DC
I^2t	0,5 A ² ·s
Intensidad de salida	
Para bus de fondo (5 V DC), máx.	1 600 mA
Pérdidas	
Pérdidas, típ.	12 W
Memoria	
Memoria de trabajo	
<ul style="list-style-type: none"> integrada ampliable 	125 kbyte No
Memoria de carga	
<ul style="list-style-type: none"> integrada 	4 Mbyte
Respaldo	
<ul style="list-style-type: none"> existente libre de mantenimiento sin pila 	Sí Sí Sí
Tiempos de ejecución de la CPU	
para operaciones de bits, típ.	0,08 μ s
para operaciones a palabras, típ.	1,7 μ s

para aritmética de coma flotante, típ.	2,3 µs
CPU-bloques	
Nº de bloques (total)	DBs, FCs, FBs, contadores y temporizadores. El número máximo de bloques direccionables es de 1 a 65535. No hay ninguna restricción, uso de toda la memoria de trabajo
Áreas de datos y su remanencia	
Área de datos remanentes (incl. temporizadores, contadores, marcas), máx.	10 kbyte
Marcas	
• Número, máx.	8 kbyte; Tamaño del área de marcas
Datos locales	
• por cada prioridad, máx.	16 kbyte
Área de direcciones	
Imagen del proceso	
• Entradas, configurables	1 kbyte
• Salidas, configurables	1 kbyte
Configuración del hardware	
Nº de módulos por sistema, máx.	3 Communication Module, 1 Signal Board, 8 Signal Module
Hora	
Reloj	
• Reloj de hardware (en tiempo real)	Sí
• Duración del respaldo	480 h; típicamente
Entradas digitales	
Nº de entradas digitales	14; integrado
• De ellas, entradas usable para funciones tecnológicas	6; HSC (High Speed Counting)
Fuente/sumidero (M/P)	Sí
Número de entradas atacables simultáneamente	
Todas las posiciones de montaje	
— hasta 40 °C, máx.	14
Tensión de entrada	
• Valor nominal (DC)	24 V
• para señal "0"	5 V DC, con 1 mA
• para señal "1"	15 V DC at 2,5 mA
Retardo a la entrada (a tensión nominal de entrada)	
para entradas estándar	
— en transición "0" a "1", máx.	0,2 ms
— en transición "0" a "1", máx.	12,8 ms
para entradas de alarmas	
— parametrizable	Sí
Longitud del cable	
• apantallado, máx.	500 m; 50 m para funciones tecnológicas
• no apantallado, máx.	300 m; para funciones tecnológicas: No
Salidas digitales	
Número de salidas	10
• de ellas, salidas rápidas	4
Limitación de la sobretensión inductiva de corte a	L+ (-48 V)
Poder de corte de las salidas	
• con carga resistiva, máx.	0,5 A
• con carga tipo lámpara, máx.	5 W
Tensión de salida	
• para señal "0", máx.	0,1 V; con carga de 10 kOhm
• para señal "1", mín.	20 V
Intensidad de salida	
• para señal "1" valor nominal	0,5 A
• para señal "0" intensidad residual, máx.	0,1 mA
Retardo a la salida con carga resistiva	
• "0" a "1", máx.	1 µs

• "1" a "0", máx.	5 µs
Frecuencia de conmutación	
• de las salidas de impulsos, con carga óhmica, máx.	100 kHz
Salidas de relé	
• N° de salidas relé	0
Longitud del cable	
• apantallado, máx.	500 m
• no apantallado, máx.	150 m
Entradas analógicas	
N° de entradas analógicas	2
Rangos de entrada	
• Tensión	Sí
Rangos de entrada (valores nominales), tensiones	
• 0 a +10 V	Sí
Longitud del cable	
• apantallado, máx.	100 m; trenzado y apantallado
Salidas analógicas	
N° de salidas analógicas	2
Rangos de salida, intensidad	
• 0 a 20 mA	Sí
Formación de valor analógico para entradas	
Tiempo de integración y conversión/resolución por canal	
• Resolución con rango de rebase (bits incl. signo), máx.	10 bit
• Tiempo de integración parametrizable	Sí
• Tiempo de conversión (por canal)	625 µs
Formación de valor analógico para salidas	
Tiempo de integración y conversión/resolución por canal	
• Resolución con rango de rebase (bits incl. signo), máx.	10 bit
Sensor	
Sensores compatibles	
• Sensor a 2 hilos	Sí
1. Interfaz	
Tipo de interfaz	PROFINET
con aislamiento galvánico	Sí
Detección automática de la velocidad de transferencia	Sí
Autonegociación	Sí
Autocrossing	Sí
Física de la interfaz	
• RJ 45 (Ethernet)	Sí
• Número de puertos	2
• Switch integrado	Sí
Protocolos	
• PROFINET IO-Controller	Sí
• PROFINET IO-Device	Sí
• Comunicación SIMATIC	Sí
• Comunicación IE abierta	Sí; También disponible cifrada
• Servidores web	Sí
• Redundancia del medio	Sí; como cliente MRP
PROFINET IO-Controller	
• Velocidad de transferencia, máx.	100 Mbit/s
Servicios	
— Comunicación PG/OP	Sí
— Modo isócrono	No
— IRT	No
— PROFIenergy	No
— Arranque priorizado	Sí

— Número de dispositivos IO con arranque preferente, máx.	16
— N° de IO Devices que se pueden conectar en total, máx.	16
— N° de IO-Devices conectables para RT, máx.	16
— de ellos, en línea, máx.	16
— Activar/desactivar IO Devices	Sí
— N° de IO-Devices activables/desactivables simultáneamente, máx.	8
PROFINET IO-Device	
Servicios	
— Comunicación PG/OP	Sí
— Modo isócrono	No
— IRT	No
— PROFIenergy	Sí
— Shared Device	Sí
— N° de IO Controller con Shared Device, máx.	2
Protocolos	
Soporta protocolo para PROFINET IO	Sí
PROFIBUS	Sí; Requiere CM 1243-5 (maestro) o CM 1242-5 (esclavo)
AS-Interface	Sí; Se requiere un CM 1243-2
Protocolos (Ethernet)	
• TCP/IP	Sí
• DHCP	No
• SNMP	Sí
• DCP	Sí
• LLDP	Sí
Funcionamiento redundante	
Redundancia del medio	
— MRP	Sí
— MRPD	No
Comunicación SIMATIC	
• S7-Routing	Sí
Comunicación IE abierta	
• TCP/IP	Sí
— Tamaño de datos, máx.	8 kbyte
• ISO-on-TCP (RFC1006)	Sí
— Tamaño de datos, máx.	8 kbyte
• UDP	Sí
— Tamaño de datos, máx.	1 472 byte
Servidores web	
• Soporta	Sí
• Páginas web definidas por el usuario	Sí
OPC UA	
• Requiere licencia runtime	Sí
• OPC UA Server	Sí; Acceso a datos (Read, Write, Subscribe), requiere licencia runtime
— Número de sesiones, máx.	5
— Número de variables accesibles, máx.	1 000
— Número de suscripciones por sesión, máx.	5
— Intervalo de muestreo, mín.	100 ms
— Intervalo de emisión, mín.	200 ms
— Número de elementos vigilados (monitored items), máx.	500
— Número de interfaces del servidor, máx.	2
— Número de nodos en interfaces del servidor definidas por el usuario, máx.	1 000
Otros protocolos	
• MODBUS	Sí
Funciones de comunicación	

Comunicación S7	
<ul style="list-style-type: none"> • Soporta 	Sí
<ul style="list-style-type: none"> • como servidor 	Sí
<ul style="list-style-type: none"> • Como cliente 	Sí
<ul style="list-style-type: none"> • Datos útiles por petición, máx. 	ver la Ayuda online (S7 communication, User data size)
Nº de conexiones	
<ul style="list-style-type: none"> • total 	8 conexiones para la comunicación de usuarios abierta (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV, 8 conexiones CPU/CPU (cliente o servidor) para datos GET/PUT, 6 conexiones para asignación dinámica a GET/PUT o a comunicación de usuarios abierta
Funciones de test y puesta en marcha	
Estado/forzado	
<ul style="list-style-type: none"> • Estado/forzado de variables 	Sí
<ul style="list-style-type: none"> • Variables 	Entradas/salidas, marcas, DB, E/S de periferia, tiempos, contadores
Forzado permanente	
<ul style="list-style-type: none"> • Forzado permanente 	Sí
Búfer de diagnóstico	
<ul style="list-style-type: none"> • existente 	Sí
Traces	
<ul style="list-style-type: none"> • Número de Traces configurables 	2
<ul style="list-style-type: none"> • Tamaño de memoria por Trace, máx. 	512 kbyte
Alarmas/diagnósticos/información de estado	
LED señalizador de diagnóstico	
<ul style="list-style-type: none"> • LED RUN/STOP 	Sí
<ul style="list-style-type: none"> • LED ERROR 	Sí
<ul style="list-style-type: none"> • LED MAINT 	Sí
Funciones integradas	
Nº de contadores	6
Frecuencia de contaje (contadores), máx.	100 kHz
Medida de frecuencia	Sí
Posicionamiento en lazo abierto	Sí
Número de ejes de posicionamiento con regulación de posición, máx.	8
Número de ejes de posicionamiento mediante interfaz impulsos/sentido	4
Regulador PID	Sí
Nº de entradas de alarma	4
Nº de salidas de impulsos	4
Frecuencia límite (impulsos)	100 kHz
Aislamiento galvánico	
Aislamiento galvánico módulos de E digitales	
<ul style="list-style-type: none"> • Aislamiento galvánico módulos de E digitales 	No
<ul style="list-style-type: none"> • entre los canales, en grupos de 	1
Aislamiento galvánico módulos de S digitales	
<ul style="list-style-type: none"> • Aislamiento galvánico módulos de S digitales 	Sí
<ul style="list-style-type: none"> • entre los canales 	No
<ul style="list-style-type: none"> • entre los canales, en grupos de 	1
CEM	
Inmunidad a perturbaciones por descargas de electricidad estática	
<ul style="list-style-type: none"> • Inmunidad a perturbaciones por descargas de electricidad estática IEC 61000-4-2 	Sí
<ul style="list-style-type: none"> — Tensión de ensayo con descarga en aire 	8 kV
<ul style="list-style-type: none"> — Tensión de ensayo para descarga por contacto 	6 kV
Inmunidad a perturbaciones conducidas	
<ul style="list-style-type: none"> • Inmunidad a perturbaciones en cables de alimentación según IEC 61000-4-4 	Sí
<ul style="list-style-type: none"> • Inmunidad a perturbaciones por cables de señales IEC 61000-4-4 	Sí
Inmunidad a perturbaciones por tensiones de choque (sobretensión transitoria)	

<ul style="list-style-type: none"> ● Inmunidad a perturbaciones en cables de alimentación según IEC 61000-4-5 	Sí
Inmunidad a perturbaciones conducidas, inducidas mediante campos de alta frecuencia	
<ul style="list-style-type: none"> ● Inmunidad a campos electromagnéticos radiados a frecuencias radioeléctricas según IEC 61000-4-6 	Sí
Emisión de radiointerferencias según EN 55 011	
<ul style="list-style-type: none"> ● Clase de límite A, para aplicación en la industria 	Sí
<ul style="list-style-type: none"> ● Clase de límite B, para aplicación en el ámbito residencial 	Sí
Grado de protección y clase de protección	
Grado de protección IP	IP20
Normas, homologaciones, certificados	
Marcado CE	Sí
Homologación UL	Sí
cULus	Sí
Homologación FM	Sí
RCM (anteriormente C-TICK)	Sí
Homologación KC	Sí
Homologaciones navales	Sí
Condiciones ambientales	
Caída libre	
<ul style="list-style-type: none"> ● Altura de caída, máx. 	0,3 m
Temperatura ambiente en servicio	
<ul style="list-style-type: none"> ● mín. 	-20 °C
<ul style="list-style-type: none"> ● máx. 	60 °C; N.º de entradas o salidas conectadas al mismo tiempo: 7 o 5 (sin puntos contiguos) con 60 °C en horizontal o 50 °C en vertical, 14 o 10 con 55 °C en horizontal o 45 °C en vertical
<ul style="list-style-type: none"> ● Posición de montaje horizontal, mín. 	-20 °C
<ul style="list-style-type: none"> ● Posición de montaje horizontal, máx. 	60 °C
<ul style="list-style-type: none"> ● Posición de montaje vertical, mín. 	-20 °C
<ul style="list-style-type: none"> ● Posición de montaje vertical, máx. 	50 °C
Temperatura ambiente en almacenaje/transporte	
<ul style="list-style-type: none"> ● mín. 	-40 °C
<ul style="list-style-type: none"> ● máx. 	70 °C
Presión atmosférica según IEC 60068-2-13	
<ul style="list-style-type: none"> ● En servicio mín. 	795 hPa
<ul style="list-style-type: none"> ● En servicio máx. 	1 080 hPa
<ul style="list-style-type: none"> ● Almacenamiento/transporte, mín. 	660 hPa
<ul style="list-style-type: none"> ● Almacenamiento/transporte, máx. 	1 080 hPa
Altitud en servicio referida al nivel del mar	
<ul style="list-style-type: none"> ● Altitud de instalación, mín. 	-1 000 m
<ul style="list-style-type: none"> ● Altitud de instalación, máx. 	2 000 m
Humedad relativa del aire	
<ul style="list-style-type: none"> ● En servicio máx. 	95 %; sin condensación
Vibraciones	
<ul style="list-style-type: none"> ● Resistencia a vibraciones durante el funcionamiento según IEC 60068-2-6 	Montaje en pared 2 g (m/s ²); perfil DIN 1 g (m/s ²)
<ul style="list-style-type: none"> ● En servicio, según DIN IEC 60068-2-6 	Sí
Ensayo de resistencia a choques	
<ul style="list-style-type: none"> ● ensayado según DIN IEC 60068-2-27 	Sí
Configuración	
programación	
Lenguaje de programación	
— KOP	Sí
— FUP	Sí
— SCL	Sí
Protección de know-how	
<ul style="list-style-type: none"> ● Protección de programas de usuario/Protección por contraseña 	Sí


• Protección contra copia	Sí
• Protección de bloques	Sí
Protección de acceso	
• Nivel de protección: Protección contra escritura	Sí
• Nivel de protección: Protección contra escritura/lectura	Sí
• Nivel de protección: Protección completa	Sí
Vigilancia de tiempo de ciclo	
• Configurable	Sí
Dimensiones	
Ancho	130 mm
Altura	100 mm
Profundidad	75 mm
Pesos	
Peso, aprox.	500 g
Última modificación:	16/01/2021 



Figure similar

MLFB-Ordering data

6SL3210-5HB10-2UF0

Client order no. :

Order no. :

Offer no. :

Remarks :

Item no. :

Consignment no. :

Project :

Rated data

Input

Number of phases	1 AC
Line voltage	200 ... 240 V ±10 %
Line frequency	45 ... 66 Hz
Rated current	2.7 A
Inrush current	8.0 A

Output

Number of phases	3 AC
Rated power	0.20 kW (0.27 hp)
Rated current I_N	1.4 A
Max. output current	4.8 A
Pulse frequency	8 kHz
Output frequency for servo control	0 ... 550 Hz

Electronics power supply

Voltage	24 V -15 % ... +20 %
Current demand, max.	0.8 A

Communication

Communication	PROFINET
---------------	----------

Ambient conditions

Installation altitude	1000 m (3280.84 ft)
Cooling	convection cooling

Ambient temperature during

Operation	0 ... 50 °C (32 ... 122 °F) Better than class 3K3, in acc. with EN 60721-3-3, without derating
Transport	-40 ... 70 °C (-40 ... 158 °F) Class 2K4, in acc. with EN 60721-3-2, in transport packaging
Storage	-25 ... 55 °C (-13 ... 131 °F) Class 1K4, in acc. with EN 60721-3-1, in product packaging

Relative humidity during

Max. operation	95 % RH, condensation not permitted
Transport, max.	95 % at 40 °C (104 °F)
Bearing, max.	95 %



Figure similar

Inputs / outputs

Standard digital inputs

Number	5
	of which 2 for F-DI

Fail-safe digital inputs

Number	1
	Can only be used for STO/SS1

Mechanical data

Dimensions

Width	45.0 mm (1.77 in)
Height without shield plate	170.0 mm (6.69 in)
Depth	172.4 mm (6.69 in)
Mounting position	vertical wall mounting
Degree of protection	IP20 / UL open type
Size	FSA
Net weight	1.10 kg (2.43 lb)

Connections

Line side

Version	push-in spring-type terminals
Conductor cross-section	0.2 ... 2.5 mm ² / 24 ... 14 AWG

Motor end

Version	push-in spring-type terminals
Conductor cross-section	0.2 ... 2.5 mm ² / 24 ... 14 AWG

DC link (for braking resistor)

Version	push-in spring-type terminals
Conductor cross-section	0.2 ... 2.5 mm ² / 24 ... 14 AWG
Cable length	3 m (9.84 ft)

PE connection

Version	M4 screw studs
---------	----------------

Max. motor cable length

Shielded	50 m (164.04 ft)
----------	------------------

Standards

Compliance with standards	CE, cULus, RCM, KC, EAC
---------------------------	-------------------------

CE marking	Low-voltage directive 2006/95/EC
------------	----------------------------------

Verification of suitability for fail-safety

SIL 2 in acc. with IEC 61508 parts 1-3 (2010) and IEC 61800-5-2 (2016), PL d in acc. with ISO 13849 part 1 (2015), Category 3 in acc. with IEC 60204 (2007)

Data sheet for SIMOTICS S-1FK2



Figure similar

MLFB-Ordering data

1FK2104-4AK10-1MA0

Client order no. :

Order no. :

Offer no. :

Remarks :

Item no. :

Consignment no. :

Project :

Basic motor data		Mechanical data	
Motor type	Permanent-magnet synchronous motor, Natural cooling, IP64	Design acc. to Code I	IM B5 (IM V1, IM V3)
Motor type	High Dynamic	Vibration severity grade	Grade A
Static torque	1.27 Nm	Shaft height	40
Static current	2.4 A	Flange size (AB)	80 mm
Maximum torque	3.85 Nm	Centering ring (N)	70 mm
Maximum current	8.7 A	Hole circle (M)	90 mm
Maximum speed	8000 rpm	Screw-on hole (S)	6.5 mm
Rotor moment of inertia	0.00004300 kgm ²	Overall length (LB)	142 mm
Weight	2.9 kg	Diameter of shaft (D)	19 mm
		Length of shaft (E)	40 mm
		Length of flange diagonal (P)	105 mm
		Shaft extension	Fitted key
		Color of the housing	Standard (Anthracite, similar to RAL 7016)

Rated data	
SINAMICS S210, 3AC 400V	
Rated speed	6000 rpm
Rated torque	0.95 Nm
Rated current	1.9 A
Rated power	0.60 kW

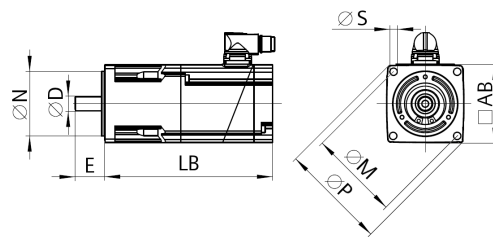


Figure similar

Encoder system	
Encoder system	Encoder AM22DQC: Absolute encoder 22 bit + 12 bit multiturn

Motor connection	
Connection type	OCC for S210
Connector size	M17



Figure similar

MLFB-Ordering data 1FK2104-4AK10-1MA0

Holding brake

Holding torque 3.30 Nm

Opening time 50 ms

Closing time 15 ms

Maximum single switching energy ¹⁾ 270.0 J

Service life, operating energy 120000 J

Holding current 0.2 A

Break-induced current for 500 ms ²⁾ 1.2 A

¹⁾ Up to three consecutive emergency stops and up to 25% of all emergency stops as a W_{max} high energy stop possible.

²⁾ Typical value for 20°C ambient temperature. At -15°C the break-induced currents can be increased by up to 30%.