



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Grado en Ing. Electrónica Industrial y Automática

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

*Comunicación entre PLC S7-1200 y driver
C5-E-2-09 mediante protocolo CANopen para
el control de un motor de corriente continua.*

Autor: Rodrigo Barrera San José

Tutor: Alberto Herreros López
Departamento de Ingeniería
de Sistemas y Automática

Valladolid, junio de 2023



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Agradecimientos

Para comenzar este proyecto, me gustaría agradecer a la empresa ICARE Automation el préstamo de los componentes integrados en la maqueta expuesta en este documento, así como darme la oportunidad de trabajar y aprender de ellos y aportar la documentación y medios necesarios para el correcto desarrollo de este trabajo.

También agradecer a mi tutor D. Alberto Herreros López por su implicación, que se ha prestado a dirigir el desarrollo del mismo, aportando su elevada experiencia a la hora de estructurar y presentar el trabajo.

A mis colegas, junto a los que llevo desde prácticamente el colegio, y con los que he compartido horas de estudio, exámenes, clases y laboratorios, que han ayudado a que existan muy buenas experiencias que recordar a lo largo de estos años de aprendizaje, y que los momentos más duros hayan sido mucho más llevaderos.

A mis amigos y familia, por ser el pilar fundamental de mi día a día fuera de la rutina de la universidad, por darme el cariño y la confianza necesarios para afrontar cualquier situación a lo largo de mi vida, además de la oportunidad de dedicarme a lo que verdaderamente me haga feliz.

Y en especial a mis padres, a quienes seguro les hará mucha ilusión leer esta cita. Gracias.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Resumen

El presente Trabajo de Fin de Grado pretende plasmar los conocimientos adquiridos en el grado de Ingeniería Electrónica Industrial y Automática, así como en las prácticas en empresa desarrolladas durante el proyecto. Se trata del análisis previo, realización del presupuesto, prototipado, montaje, programación y comunicación entre un PLC Simatic S7-1200 y el driver C5-E-2-09 de Nanotec, con el objetivo final del control del movimiento de una rueda a través del motor de corriente continua DB59, también de Nanotec.

El citado PLC está programado desde TIA Portal, y la comunicación entre éste y el driver se realiza a través de la pasarela PN/CAN Link de Siemens, que permite pasar de ProfiNet a CANopen, protocolo utilizado por el driver.

El objetivo de este trabajo, así como sus líneas futuras de investigación, es el desarrollo de un vehículo de guiado automático (AGV) con una rueda motriz y varias ruedas “locas”, que funcione mediante unos pulsadores y pueda ser integrado en un entorno real, tanto en el sector automovilístico como en otros sectores.

Palabras clave

PLC, Driver, PN/CAN Link, Motor, CANopen, Siemens, Nanotec, TIA Portal, AGV



Abstract

The aim of the actual Final Degree Project is to put into practice the knowledge acquired throughout the Industrial Electronic and Automatic Engineering degree, but also on the internship during the project. It involves the previous analysis, budgeting development, prototyping, assembly, programming and communication between a S7-1200 Simatic PLC and the C5-E-2-09 driver from Nanotec, with the final objective of controlling the wheel movement through a DB59 direct current motor from Nanotec too.

The aforementioned PLC is programmed on TIA Portal, and communications between PLC and driver are carried out through Siemens' PN/CAN Link gateway which allows passing from ProfiNet to CANopen, the protocol used by the driver.

The final objective of this work, as well as its future lines of research, is the development of an automatic guided vehicle (AGV) with one drive and several "idle" wheels which can be integrated into a real environment, both in the automobile sector and in other ones.

Key words

PLC, Driver, PN/CAN Link, Motor, CANopen, Siemens, Nanotec, TIA Portal, AGV



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Índice de contenidos

Capítulo 1. INTRODUCCIÓN.....	17
1.1 Motivación, origen y justificación del proyecto.....	17
1.2 Objetivos.....	18
1.3 Fundamentos previos.....	18
1.4 Alcance del proyecto.....	19
1.5 Estructura del trabajo.....	19
1.6 Convenciones.....	20
Capítulo 2. MARCO TEÓRICO.....	22
2.1 Robótica.....	22
2.1.1 Historia de la robótica.....	22
2.1.2 Robótica industrial.....	22
2.1.3 Definición y clasificación de robots manipuladores industriales.....	24
2.2 Automatización.....	27
2.2.1 Historia de la automatización.....	27
2.2.2 Automatización industrial.....	29
2.2.3 Controlador lógico programable (PLC).....	30
2.3 Comunicaciones industriales.....	32
2.3.1 Niveles de comunicación en una red industrial.....	33
2.3.2 Protocolos de comunicación industrial.....	34
2.3.3 ProfiNet (PN).....	35
2.3.3.1 Funciones de seguridad PROFIsafe.....	37
2.3.4 CANopen.....	38
2.3.4.1 Introducción a CAN.....	38
2.3.4.2 Protocolo CANopen.....	38
Capítulo 3. ESTADO DEL ARTE.....	43
3.1 Antecedentes.....	43
3.2 Análisis del problema.....	45
3.3 Nanotec.....	45
3.3.1 Plug & Drive Studio (PnDS).....	46
3.3.1.1 Setup.....	49
3.3.1.2 Operation.....	50
3.3.1.3 Object dictionary.....	52
3.4 Siemens.....	54
3.4.1 TIA Portal.....	55
3.5 PN/CANopen Link.....	60



3.5.1 CANopen Transparent.....	60
3.5.2 CANopen Master.....	60
3.5.3 CANopen Slave.....	61
3.6 Schneider.....	61
3.6.1 Elementos de potencia.....	62
3.6.2 Elementos de control.....	62
Capítulo 4. OBJETO DE ESTUDIO.....	66
4.1 Introducción.....	66
4.2 Componentes.....	66
4.2.1 Rueda motriz.....	66
4.2.2 Motor DC. Frenos y encoder.....	68
4.2.3 Driver.....	71
4.2.4 Autómata programable (PLC).....	73
4.2.5 Módulos adicionales de entradas y salidas.....	74
4.2.6 Pasarela (PN/CAN Link).....	75
4.2.7 Fuente de alimentación.....	75
4.2.8 Contactores.....	76
4.2.9 Pulsadores. Selectores. Setas de emergencia.....	77
4.2.10 Caja estanca.....	77
4.2.11 Cables RJ45 y D-SUB9.....	78
4.2.12 Carril DIN.....	79
4.2.13 Borneros para carril DIN.....	79
4.3 Proceso de montaje.....	80
4.4 Conexionado.....	81
4.4.1 Componentes de Siemens.....	82
4.4.2 Componentes de Nanotec.....	84
4.4.3 Botonería.....	86
4.4.4 Esquema eléctrico completo.....	88
4.5 Desarrollo en TIA Portal v17.....	89
4.5.1 Autómata programable (PLC).....	89
4.5.1.1 Main [OB1].....	94
4.5.1.2 Main_Safety_RTG1 [FBO].....	94
4.5.1.3 FBmanual [FB1].....	95
4.5.2 Pasarela (PN/CAN Link). Protocolo CAN.....	95
Capítulo 5. ESTUDIO ECONÓMICO.....	104
5.1 Costes directos.....	104



5.1.1 Costes de personal.....	104
5.1.2 Costes de equipo y software.....	105
5.1.3 Costes de material.....	106
5.1.4 Costes totales.....	108
5.2 Costes indirectos.....	108
5.3 Costes totales.....	108
Capítulo 6. PRUEBAS Y RESULTADOS.....	111
Capítulo 7. CONCLUSIONES Y LÍNEAS FUTURAS.....	115
7.1 Conclusiones.....	115
7.2 Líneas futuras.....	116
BIBLIOGRAFÍA/REFERENCIAS.....	119
ANEXOS.....	124
ANEXO I. VARIABLES DE PROGRAMA DE TIA PORTAL.....	124
ANEXO II. BLOQUES DE PROGRAMACIÓN DE TIA PORTAL.....	125
ANEXO III. ESQUEMA ELÉCTRICO.....	133
ANEXO IV. FOTOGRAFÍAS.....	134
ANEXO V. HOJAS DE DATOS DE LOS COMPONENTES.....	135



Índice de figuras

Figura 2.1: Robot UNIMATE.....	23
Figura 2.2: Resolución, precisión y repetibilidad.....	24
Figura 2.3: Vehículos de guiado automático (AGVs).....	25
Figura 2.4: Robots antropomórficos.....	26
Figura 2.5: Robot Delta.....	26
Figura 2.6: Robot cartesiano.....	26
Figura 2.7: Robot SCARA.....	27
Figura 2.8: Gallo de Estrasburgo [5].....	28
Figura 2.9: Arquitectura interna del PLC [7].....	30
Figura 2.10: Esquema de un PLC tipo modular.....	31
Figura 2.11: Esquema del ciclo de ejecución típico de un PLC.....	32
Figura 2.12: Pirámide CIM [9].....	33
Figura 2.13: Perfiles FMS, DP y PA [10].....	36
Figura 2.14: Método de acceso al medio de bus de campo de ProfiBus [10]...	37
Figura 2.15: Acceso al bus simultáneo de tres nodos CAN [10].....	38
Figura 2.16: Protocolos CAN y CANopen en el modelo OSI.....	39
Figura 2.17: Descripción de los diferentes COB-ID en CANopen.....	40
Figura 2.18: Estructura de la trama de datos en CAN [COB-ID].....	40
Figura 2.19: Espacio de identificación de CANopen.....	41
Figura 3.1: Aula de formación correspondiente al TFG de José Delgado [13].	44
Figura 3.2: Instalación robotizada correspondiente al TFG de Jorge Villa González [14].....	44
Figura 3.3: Menú principal del PnDS.....	46
Figura 3.4: Ventana de Home del PnDS.....	47
Figura 3.5: Menú del controlador en PnDS (I).....	47
Figura 3.6: Menú del controlador en PnDS (II).....	48
Figura 3.7: Menú Setup de PnDS (I).....	49
Figura 3.8: Menú Setup de PnDS (II).....	49
Figura 3.9: Opción de “Profile Position” de PnDS.....	50
Figura 3.10: Opción de “Profile Position” de PnDS.....	50
Figura 3.11: Opción de “Profile Velocity” de PnDS.....	51
Figura 3.12: Opción de “Profile Torque” de PnDS.....	51
Figura 3.13: Opción de “Homing” de PnDS.....	52
Figura 3.14: Menú Object Dictionary de PnDS (I).....	52
Figura 3.15: Menú Object Dictionary de PnDS (II).....	53



Figura 3.16: Menú Object Dictionary de PnDS (III).....	53
Figura 3.17: Menú Object Dictionary de PnDS (IV).....	54
Figura 3.18: Menú Object Dictionary de PnDS (V).....	54
Figura 3.19: TIA Portal V17. Vista del portal.....	55
Figura 3.20: TIA Portal V17.Vista del proyecto.....	56
Figura 3.21: TIA Portal V17. Añadiendo el PLC y la CPU utilizados.....	56
Figura 3.22: TIA Portal V17. Barra de herramientas.....	57
Figura 3.23: TIA Portal V17. Árbol del proyecto.....	57
Figura 3.24: TIA Portal V17. Área de trabajo (I).....	58
Figura 3.25: TIA Portal V17. Área de trabajo (II).....	58
Figura 3.26: TIA Portal V17. Herramientas de configuración.....	59
Figura 3.27: TIA Portal V17. Ventana de propiedades, información y diagnóstico.....	60
Figura 3.28: Ejemplo de una red CANopen similar a la desarrollada en el proyecto.....	61
Figura 3.29: Despiece de un contactor.....	62
Figura 3.30: Contactos NA y NC.....	63
Figura 3.31: Despiece de pulsadores (izquierda) y selectores (derecha).....	63
Figura 3.32: Despiece de seta de emergencia con doble conexión.....	64
Figura 4.1: Despiece de la rueda WD14050-5616-23X de Nanotec.....	67
Figura 4.2: Motores DB59L024035-A (derecha) y DB59L024035R-A (izquierda).....	69
Figura 4.3: Freno BWA de Nanotec.....	69
Figura 4.4: Encoder NME1 y conectores ZK-NME1-13-500-S de Nanotec....	70
Figura 4.5: Driver C5-E de Nanotec.....	72
Figura 4.6: PLC SIMATIC S7-1200 con CPU 1215FC DC/DC/DC de Siemens...	73
Figura 4.7: Módulos adicionales de entrada y salida de Siemens.....	74
Figura 4.8: Pasarela PN/CAN Link de Siemens.....	75
Figura 4.9: Fuente de alimentación a 24V de Schneider.....	76
Figura 4.10: Contactor de Schneider.....	76
Figura 4.11: Elementos de control de Schneider.....	77
Figura 4.12: Caja estanca de Schneider.....	78
Figura 4.13: Cables RJ45 y D-SUB9.....	78
Figura 4.14: Ejemplo de carril DIN [20].....	79
Figura 4.15: Bornero.....	79
Figura 4.16: Montaje del proyecto, dividido por zonas.....	81
Figura 4.17: Asignación de pines del PLC.....	82



Figura 4.18: Asignación de pines del módulo de entrada (izquierda) y salida (derecha)..... 83

Figura 4.19: Asignación de pines de la pasarela..... 84

Figura 4.20: Asignación de pines del conector (izquierda) y del encoder (derecha)..... 85

Figura 4.21: Asignación de pines del driver..... 85

Figura 4.22: Asignación de pines del motor..... 86

Figura 4.23: Símbolo eléctrico de pulsador NA (izquierda) y selector NA (derecha)..... 86

Figura 4.24: Símbolo eléctrico de una bombilla..... 87

Figura 4.25: Símbolo eléctrico de seta de emergencia..... 87

Figura 4.26: Símbolo eléctrico del contactor activo por relé..... 88

Figura 4.27: Esquema eléctrico del proyecto..... 89

Figura 4.28: TIA Portal. Ejemplo de programación en ladder..... 90

Figura 4.29: TIA Portal. Símbolo de contacto general..... 90

Figura 4.30: TIA Portal. Bloques de programación ladder usados en el proyecto..... 93

Figura 4.31: TIA Portal. Segmentos del Main OB1..... 94

Figura 4.32: TIA Portal. Segmento del Main_Safety_RTG1..... 94

Figura 4.33: TIA Portal. Segmentos del FBmanual..... 95

Figura 4.34: TIA Portal. Vista de red en TIA Portal al añadir la pasarela PN/CAN Link..... 95

Figura 4.35: TIA Portal. Modo de configuración de la pasarela (CANopen manager)..... 96

Figura 4.36: TIA Portal. Importar archivo .eds al nodo universal..... 96

Figura 4.37: TIA Portal. Creando las PDOs en el lado del maestro..... 97

Figura 4.38: TIA Portal. Vista general de las Transmit PDOs del nodo..... 100

Figura 4.39: TIA Portal. Vista general de las Receive PDOs del manager..... 100

Figura 4.40: TIA Portal. Vista general de las Receive PDOs del nodo..... 100

Figura 4.41: TIA Portal. Vista general de las Transmit PDOs del manager... 100

Figura 4.42: Tabla de los posibles valores de Node ID y velocidad de transmisión..... 101

Figura 4.43: Valores de Node ID y velocidad de transmisión desde PnDS..... 101

Figura 4.44: TIA Portal. Valor de Node ID del esclavo..... 102

Figura 4.45: TIA Portal. Valor de Node ID del maestro..... 102

Figura 4.46: TIA Portal. Activar o desactivar bus de comunicación..... 102

Figura 6.1: Modo de transmisión del protocolo CAN..... 111



Figura 6.2: Asignación de pines en el lado del driver (RJ45).....	112
Figura 6.3: Asignación de pines en el lado de la pasarela (D-SUB9).....	112
Figura 6.4: Cabezales hembra-macho de la terminación D-SUB9.....	113
Figura 6.5: Conexión manual del cabezal hembra D-SUB9.....	113
Figura 6.6: TIA Portal. Error con la PDO 16#1020:16#1.....	113



Índice de tablas

Tabla 2.1: Autómatas famosos.....	33
Tabla 4.1: Datos técnicos de la rueda WD14050-5616-23X de Nanotec.....	72
Tabla 4.2: Datos técnicos del motor DB59L024035-A de Nanotec.....	73
Tabla 4.3: Datos técnicos del freno BWA de Nanotec.....	75
Tabla 4.4: Datos técnicos del encoder NME1 de Nanotec.....	76
Tabla 4.5: Datos técnicos del driver C5-E-2-09 de Nanotec.....	78
Tabla 4.6: Entradas y salidas del PLC y de los módulos adicionales.....	97
Tabla 4.7: Transmit PDOs del nodo.....	103
Tabla 4.8: Receive PDOs del nodo.....	104
Tabla 5.1: Días laborables en Castilla y León para el año 2023.....	109
Tabla 5.2: Desglose de tiempo dedicado al proyecto.....	110
Tabla 5.3: Costes de equipo y software.....	111
Tabla 5.4: Costes de materiales.....	112
Tabla 5.5: Costes directos totales.....	113
Tabla 5.6: Costes indirectos totales.....	113
Tabla 5.7: Costes totales.....	114



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Capítulo 1. INTRODUCCIÓN

1.1 Motivación, origen y justificación del proyecto

El campo de trabajo del grado de Ingeniería Electrónica Industrial y Automática abarca numerosas áreas, como son la automatización, la robótica o el control de procesos. Fueron estos sectores de aplicación los que me motivaron a tomar la decisión de ingresar en el grado, pues siempre he tenido una inclinación hacia el ámbito de la robótica industrial.

Durante el transcurso del mismo he tenido la oportunidad de cursar diferentes asignaturas que han contribuido en mi curiosidad y formación en estos ámbitos, como pueden ser Automatización Industrial, Sistemas Robotizados, Visión Artificial o Mecatrónica, desde un enfoque más mecánico.

Tanto la **automatización** como la **robótica** son dos ramas que están estrechamente unidas en varios sectores de la industria, pues para conseguir el control de los robots se necesita, en muchos casos, el empleo de autómatas programables (entre otros dispositivos); esta es una de las razones por las que desarrollar un proyecto relacionado con las mismas me parece de gran interés.

Además, estas preferencias me han llevado a realizar las prácticas en **ICARE Automation**, empresa que me ha brindado la posibilidad de realizar el presente Trabajo de Fin de Grado, con el objetivo de plasmar los conocimientos y las habilidades adquiridas durante las mismas. Esta empresa está centrada en aspectos como:

- Los estudios y la realización de equipos eléctricos y neumáticos
- La puesta en servicio de sistemas automatizados
- La robótica industrial
- La concepción y la realización de automatismos industriales

El proyecto tiene su origen en las tareas realizadas durante este período, con la intención de desarrollar un proyecto similar a los que realiza la empresa, o que pueda ser útil para la misma. Por ello la idea es conseguir el prototipo de un futuro **vehículo de guiado autónomo (AGV)** a partir de los componentes eléctricos y electrónicos de los que disponen en sus instalaciones, como son: un driver, motor y rueda de Nanotec; un PLC Simatic S7-1200 y la pasarela Simatic PN/CAN Link para conectarlos a través del protocolo de comunicaciones CANopen. También contará con diferentes elementos eléctricos de potencia y control para su manejo.



Por lo tanto, a lo largo de los capítulos de esta memoria se van a exponer y analizar los pasos seguidos para llevar a cabo el desarrollo del proyecto señalado anteriormente, así como los costes derivados del mismo, y las referencias bibliográficas que se han consultado. También se extraerán conclusiones del resultado y se indicarán las posibles líneas futuras de investigación dedicadas al mismo.

1.2 Objetivos

- Diseño y desarrollo de un proyecto real que incorpore distintas ramas de la ingeniería.
- Montaje y conexionado de diferentes elementos eléctricos y electrónicos.
- Realización de los correspondientes esquemas eléctricos asociados.
- Configuración y programación de un PLC.
- Puesta en marcha del driver de un motor de corriente continua para el movimiento de una rueda integrada.
- Configuración del protocolo de comunicaciones CANopen para el intercambio de información entre el PLC y el driver.

1.3 Fundamentos previos

Para la ejecución de este proyecto no se parte de cero; son necesarios conocimientos previos que se han ido adquiriendo a lo largo del grado cursado, y puestos en práctica en las horas compartidas con la empresa del sector mencionada. Estos fundamentos teóricos están relacionados con la robótica, el control de procesos, la automatización o las comunicaciones, todas ellas en un ámbito industrial. En la asignatura de **Automatización Industrial** se aprendieron las bases del software TIA Portal de Siemens empleado en el desarrollo de este trabajo. Además, esta asignatura es un primer contacto con el mundo de la automatización industrial a través de PLCs mediante su programación en TIA Portal.

Otra de las asignaturas del grado cuyos fundamentos teóricos han sido de utilidad en el desarrollo del presente trabajo es **Mecatrónica**. En este caso se trata más bien de los conocimientos prácticos adquiridos durante la misma, necesarios para realizar un correcto montaje y conexionado de los diferentes componentes eléctricos y electrónicos que se verán más adelante. El objetivo final de esta asignatura era el mismo que en este caso: realizar un programa que, a fin de cuentas, controle el driver de un motor.

Finalmente, para la realización de la parte de comunicación entre el PLC y el driver quiero destacar los conocimientos adquiridos en la asignatura de **Control y**



Comunicaciones Industriales. Esta asignatura está dividida en dos grandes bloques: por un lado la parte de comunicaciones industriales, donde se estudia la historia y los aspectos más destacados de los protocolos de comunicación empleados en el sector. Y por otro lado, el bloque de control de estos protocolos desde un enfoque más computacional.

Algunos de los protocolos que se ven en el bloque de comunicaciones son **ProfiNet y CANopen**, que son los empleados en el PLC, pasarela y driver del motor de corriente continua del proyecto realizado. El hecho de tener una base sobre los mismos antes de empezar a desarrollar mis estudios han sido de gran utilidad para saber sus características y por dónde empezar.

1.4 Alcance del proyecto

Como se ha comentado previamente, el presente trabajo va a documentar el proyecto realizado, el cual consiste en un estudio previo, montaje, conexión, programación y puesta en marcha de diferentes dispositivos electrónicos, como son el driver de un motor y un PLC de Siemens, unidos por el protocolo de comunicación CANopen mediante una pasarela PN/CAN Link.

Para ello se debe hacer uso de diferentes softwares como Plug and Drive Studio, para la puesta en marcha del driver del motor; TIA Portal, para la configuración y programación del PLC y de la pasarela; y Cade Simu, para realizar los esquemas eléctricos.

1.5 Estructura del trabajo

La estructura relativa al informe del proyecto está dividida en 7 capítulos de la siguiente manera:

- En el **capítulo 1** se hace una breve presentación del mismo, comentando aspectos introductorios como son el origen del proyecto, la motivación para realizarlo o los objetivos planteados.
- En el **capítulo 2** se analiza el marco teórico del tema que se va a tratar. Se comentan los fundamentos de la robótica y la automatización industrial, así como su fuerte correlación. También se mencionan los diferentes tipos de protocolos de comunicaciones industriales existentes, haciendo hincapié en ProfiNet y CANopen.
- En el **capítulo 3** se introduce el problema y se hace un análisis de los posibles precedentes. También se introducen las marcas de componentes de automatización industrial empleadas en el proyecto, como son Siemens, Nanotec y Schneider; y se comentan los aspectos más importantes de sus



respectivos software, así como los diferentes modos de configuración de los componentes que serán usados más adelante.

- En el **capítulo 4** se concretan los apartados anteriores con el objeto de estudio en cuestión. Se hace referencia a los componentes utilizados, al proceso de montaje, de conexionado y a la configuración y programación de la pasarela y del PLC llevada a cabo en TIA Portal.
- El **capítulo 5** consiste en un estudio económico del proyecto. Este no es un apartado técnico, pero es considerado de interés analizar el impacto económico de cualquier proyecto real. El desglose realizado permite diferenciar los costes directos e indirectos.
- En el **capítulo 6** se incluyen las pruebas y resultados del proyecto.
- En el **capítulo 7** se establecen las conclusiones en torno a los objetivos previamente asignados, y las posibles futuras líneas de investigación del prototipo desarrollado.
- Finalmente se incorporan las referencias bibliográficas que se han consultado para el desarrollo del trabajo, y los anexos.

1.6 Convenciones

A continuación se presentan una serie de formalidades y normas tanto de redacción como de estructuración que se han seguido a lo largo del proyecto.

- Se ha desglosado el índice en tres partes: índice de contenidos, índice de figuras e índice de tablas, en ese orden.
- Los apartados principales de la memoria se han presentado como capítulos, con sus correspondientes subapartados.
- Se ha respetado una página en blanco entre capítulos.
- Las palabras en otros idiomas, como por ejemplo el inglés, usadas en la memoria (excepto en títulos) se escriben en letra cursiva.
- Se resaltan en negrita las palabras clave de una sección, por comodidad a la hora de la lectura del mismo.
- Las citas incluidas se representan entre corchetes y se listan en la bibliografía final del proyecto, según el estilo IEEE.
- El nombre de los autores mencionados, al igual que las referencias a otros apartados de la memoria, aparecen subrayados.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Capítulo 2. MARCO TEÓRICO

2.1 Robótica

La **robótica** es una disciplina que se enfoca en el diseño, construcción, programación y operación de robots, es decir, sistemas automatizados capaces de realizar tareas de manera autónoma o semiautónoma. Los robots pueden ser controlados por un ser humano o por otros sistemas de control, y se utilizan en una amplia variedad de aplicaciones en sectores como la industria.

2.1.1 Historia de la robótica

El término 'robot' tiene sus orígenes en la palabra 'robota', procedente del checo, cuyo significado es trabajo forzado o servidumbre. Sin embargo, el origen etimológico de la palabra se remonta a 1920, cuando el dramaturgo de origen Checo Karel Čapek (1890-1938) presenta su obra *Rossum's Universal Robots (R.U.R)*. En esta obra aparece la figura de un robot como una herramienta diseñada por una empresa para aligerar la carga de trabajo que recae sobre los humanos [1].

No obstante, fue el escritor Isaac Asimov (1920-1992) el máximo impulsor de la palabra robot. En 1945 publicó una historia en la que por primera vez enunció sus *tres leyes de la robótica* [1].

1. Un robot no hará daño a un ser humano ni, por inacción, permitirá que un ser humano sufra daño.
2. Un robot debe cumplir las órdenes dadas por los seres humanos, a excepción de aquellas que entren en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la primera o con la segunda ley.

Estas leyes, a pesar de su origen ficticio, en la actualidad se tienen en cuenta por toda la comunidad científica dedicada a la robótica. Es por ello que se le atribuye a Asimov y a su obra literaria la creación de este término, que ha contribuido decisivamente a la divulgación y difusión del sector.

2.1.2 Robótica industrial

Una vez establecido un precedente histórico que nos lleva al origen de la palabra robot, conviene centrarse en su importancia a nivel de la industria.

En 1954 Goertz hizo uso de la tecnología electrónica y el servocontrol para conseguir sustituir la transmisión mecánica por una eléctrica, creando el primer telemanipulador con servocontrol bilateral [1].

Ralph Mosher, ingeniero de la empresa americana General Electric, desarrolló en 1958 un dispositivo formado por dos brazos mecánicos teleoperados por un maestro denominado exoesqueleto. Este dispositivo recibió el nombre de 'Handy-Man' [2].

Junto a la industria nuclear, a lo largo de los años sesenta la industria submarina comenzó a interesarse por el uso de telemanipuladores. A este interés se sumó la industria espacial en los años setenta. No obstante, la aplicación y evolución de los telemanipuladores quedó fijada a estos sectores sin poder expandirse debido a que era necesario el mando continuo de un operador. Esto provocó la necesidad de sustituir el operador por un programa de ordenador que controlase los movimientos del manipulador, lo que dio paso al concepto de robot [1].

En el año 1954 el inventor George Charles Devol desarrolló un primitivo brazo programable que podía ser utilizado para realizar tareas específicas. Este brazo artificial denominado 'Unimate' es considerado el primer robot industrial [1]. Devol fundaría Unimation en 1960, junto a Josep Engelber, empresa dedicada al desarrollo de máquinas con enfoque puramente industrial. Estos dos acontecimientos suponen el origen de la robótica industrial.



Figura 2.1: Robot UNIMATE

A partir de este momento surge un número interminable de proyectos que mejoran a los anteriores, llegando hasta la actualidad, momento en el que la robótica es un campo en constante evolución y crecimiento, con un gran potencial para mejorar la eficiencia y la precisión en una amplia gama de aplicaciones.

Pese al desarrollo y expansión de la robótica en la industria, no existe una definición universal como tal, siendo por lo tanto las diferentes organizaciones nacionales e internacionales quienes imponen sus propias definiciones del concepto de robot industrial.

No obstante, es común la aceptación de robot industrial como un brazo mecánico que tiene capacidad de manipular objetos y que incorpora un control de mayor o menor complejidad, incluyendo el principio de secuenciación y retroalimentación que permita proporcionar movimientos rápidos y precisos.

2.1.3 Definición y clasificación de robots manipuladores industriales

Dentro de los robots industriales, los más comunes son los manipuladores, entendiéndose estos como máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que puedan posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento [1].

Estos mecanismos poseen las siguientes características que determinan su funcionalidad:

- **Grados de libertad (GDL):** cada uno de los movimientos independientes que puede realizar una articulación de un robot, respecto de la anterior. Determina su capacidad para orientar herramientas [3].
- **Área de trabajo:** Volumen espacial al que puede acceder el extremo del robot, sin tener en cuenta la herramienta [3].
- **Resolución, precisión y repetibilidad.**

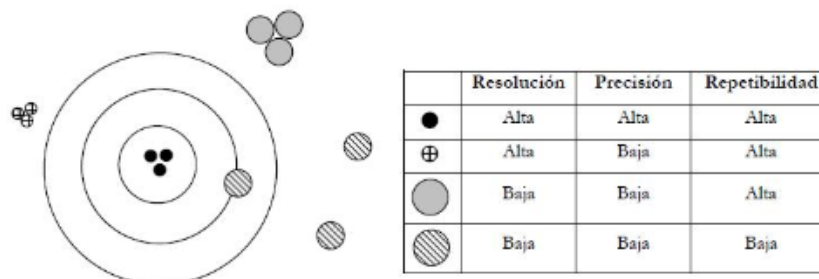


Figura 2.2: Resolución, precisión y repetibilidad

- **Capacidad de carga:** Máximo peso en carga que un robot es capaz de manipular a una velocidad determinada y considerando la configuración menos favorable [3].
- **Velocidad:** Cómo de rápido puede moverse un robot. Es un dato importante para los tiempos de ciclo, e influye en otros parámetros como la carga o la precisión (más velocidad, menos capacidad de carga o precisión) [3].
- **TCP:** tool center point. Punto central de la herramienta. Es el punto cuyas coordenadas se almacenan en el programa. Se pueden definir varios, uno por cada herramienta [3].
- **Puntos singulares:** Puntos del espacio de trabajo del robot sobre los que no es posible realizar una trayectoria rectilínea, ya sea porque implica una velocidad infinita en el movimiento o porque el valor de los ejes en dichos puntos es indeterminado. Se suele dar cuando dos o más ejes se alinean [3].

Atendiendo a estas características, podemos realizar una clasificación de los diferentes tipos de robots manipuladores.

- **Vehículo de guiado automático (AGV).** Son robots que se comportan como vehículos autónomos, principalmente destinados al transporte de materiales dentro de un entorno industrial [4].



Figura 2.3: Vehículos de guiado automático (AGVs)

- **Robot antropomórfico.** También denominados robots articulados, son los más utilizados en los procesos de fabricación. Se utilizan para realizar operaciones más complejas como la soldadura, el montaje de productos y el mecanizado. Las herramientas de fin de brazo montadas en robots articulados están diseñadas para tener 6GDL [4].



Figura 2.4: Robots antropomórficos

- **Robot Delta.** Es un tipo de robot paralelo de 3GDL conformado por dos bases unidas por tres cadenas cinemáticas basadas en el uso de paralelogramos. Se suelen utilizar en tareas de empaquetado [4].



Figura 2.5: Robot Delta

- **Robot cartesiano.** Está compuesto por tres articulaciones prismáticas, en los ejes X Y Z. Por tanto, la herramienta se limita al movimiento lineal. Los robots cartesianos son el sistema robótico más sencillo, ya que su funcionamiento sólo implica movimientos de traslación. Esto es adecuado para aplicaciones que sólo requieren un movimiento en ángulo recto sin necesidad de orientar la carga [4].

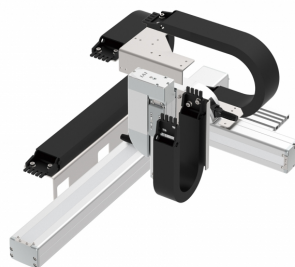


Figura 2.6: Robot cartesiano

- **Robot SCARA.** ‘*Selective Compliant Assembly Robot Arm o Selective Compliant Articulated Robot Arm*’. Es un robot de 4GDL que trabaja en ciclos de trabajo rápidos, es decir, posee una alta velocidad de desplazamiento. Son robots de posicionamiento horizontal, muy precisos, generalmente utilizados en tareas de ensamblaje o de ‘*Pick&Place*’ [4].



Figura 2.7: Robot SCARA

2.2 Automatización

La **automatización** es el proceso mediante el cual se utiliza la tecnología para realizar tareas sin intervención humana directa, es decir, es la sustitución de la mano de obra humana por máquinas o sistemas que realizan tareas de manera autónoma. Actualmente se utiliza en una amplia variedad de industrias y aplicaciones.

2.2.1 Historia de la automatización

La historia de la automatización siempre ha estado muy ligada a la robótica. A lo largo del tiempo, el hombre se ha sentido fascinado por máquinas y dispositivos capaces de imitar las funciones y los movimientos de los seres vivos. Los griegos tenían una palabra específica para denominar a estas máquinas: **autómatas** [1].

La cultura árabe (siglos VIII a XV) heredó y difundió los conocimientos griegos, utilizándolos para aplicaciones prácticas e introduciéndolos gradualmente en la vida cotidiana, como podían ser dispensadores para lavarse la cara o beber agua. El ejemplo de autómatas más antiguo que se conserva en la actualidad es el del Gallo de Estrasburgo (1352) [1].

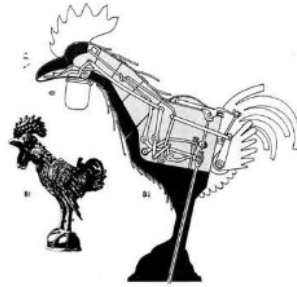


Figura 2.8: Gallo de Estrasburgo [5]

A continuación, se muestra en la siguiente tabla una recopilación de los autómatas más importantes de los siglos XV al XVII.

Año	Autor	Autómata
1352	Desconocido	Gallo de la catedral de Estrasburgo
1499	L. Da Vinci	León mecánico
1525	J. Turriano	Hombre de palo
1738	J de Vaucanson	Flautista, tamborilero, pato, muñecas mecánicas de tamaño humano
1769	W. Von Kempelen	Jugador de ajedrez
1770	Familia Droz	Escriba, organista, dibujante
1805	H. Maillardet	Muñeca mecánica capaz de dibujar

Tabla 2.1: Autómatas famosos [1]

Durante los siglos XVII y XVIII se crearon ingenios mecánicos que tenían algunas de las características de los robots actuales, pero fue a finales del siglo XVIII y principios del siglo XIX cuando se empezó a utilizar dispositivos automáticos en la producción, dando paso a la **automatización industrial**.

2.2.2 Automatización industrial

La automatización industrial se podría definir como el uso de diferentes dispositivos y maquinaria, reduciendo así la intervención humana, con el fin de mejorar la eficiencia, la calidad, la seguridad y la rentabilidad de los procesos industriales [6].

A continuación, se muestra una tabla con las ventajas e inconvenientes de la automatización industrial:

Ventajas	Inconvenientes
Mayor eficiencia y productividad	Dependencia de la tecnología
Creación de mano de obra capacitada	Reducción de mano de obra no capacitada
Reducción de costes a largo plazo	Fuerte inversión inicial
Mejora en la calidad del producto	Riesgo de interrupción de la producción en caso de fallos técnicos
Reducción de errores	
Menor riesgo laboral de los trabajadores	
Flexibilidad y capacidad de adaptación a cambios en la demanda del mercado	

Tabla 2.2: Ventajas e inconvenientes de la Automatización Industrial

Los dispositivos más comunes en la automatización industrial son [6]:

- **Controladores Lógicos Programables (PLC).** Son dispositivos electrónicos programables que controlan una variedad de procesos industriales.
- **Interfaces hombre-máquina (HMI).** Son dispositivos que proporcionan una interfaz gráfica de usuario para controlar y monitorear los procesos industriales.
- **Sistemas de control distribuido (DCS).** Son sistemas que controlan procesos industriales distribuidos en una amplia área geográfica.
- **Sistemas de supervisión, control y adquisición de datos (SCADA).** Son sistemas que proporcionan una plataforma para monitorear y controlar los procesos industriales de forma remota.

- **Robots industriales.** Son dispositivos programables que realizan tareas repetitivas y peligrosas en las líneas de producción.
- **Sensores y actuadores.** Dispositivos que recopilan información y producen una respuesta, como puede ser el movimiento de un servomotor.

2.2.3 Controlador lógico programable (PLC)

La definición de un Controlador Lógico Programable es un dispositivo electrónico utilizado para controlar procesos automatizados en una gran variedad de industrias. Los PLCs son programados para controlar y monitorear equipos y procesos, con el fin de mejorar la eficiencia y precisión del proceso automatizado. Estos dispositivos tienen entradas y salidas que se conectan a sensores, actuadores y otros equipos de control para llevar a cabo las funciones necesarias.

Atendiendo a la arquitectura interna de un PLC industrial típico, se puede desarrollar esquemáticamente como se detalla en la siguiente figura:

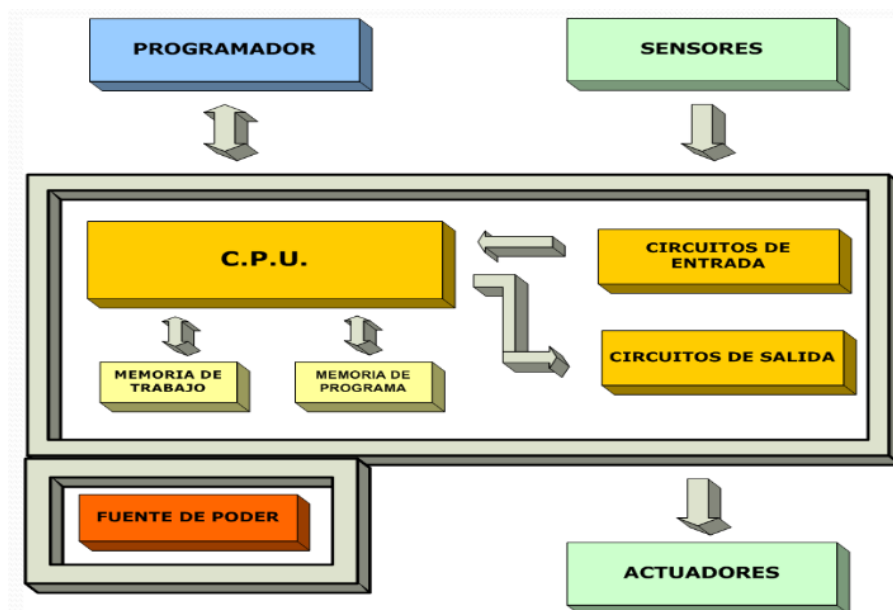


Figura 2.9: Arquitectura interna del PLC [7]

- **CPU.** También llamada procesador, es la parte que se encarga de procesar y ejecutar las instrucciones del programa y monitorizar el estado de las entradas y salidas del sistema por medio de sensores. Mediante un bus de datos transmite y envía la información al resto de partes del sistema [8].

- **Módulos de entrada/salida.** Son dispositivos que permiten la conexión del PLC con los elementos del proceso que se desean controlar. Los módulos de entrada convierten las señales analógicas o digitales provenientes del mismo en señales eléctricas que pueden ser procesadas por la CPU del PLC, mientras que los módulos de salida envían señales eléctricas a los elementos del proceso para que realicen las acciones deseadas.
- **Bloques de memoria.** El PLC cuenta con tres tipos de memoria: la memoria de programa, la memoria de datos y la variable interna del autómatas.
 - **Memoria de programa.** Aquí es donde se almacena el programa con las instrucciones que controlan el funcionamiento del PLC. Este programa es creado en un lenguaje de programación de PLC como puede ser *Ladder*. Tipo de memoria no volátil, es decir, no se pierde cuando el PLC se apaga.
 - **Memoria de datos.** Esta memoria se utiliza para almacenar variables y otros datos que el programa del PLC utiliza para llevar a cabo el control de los procesos. Puede ser volátil o no volátil, dependiendo del tipo de memoria que se utilice.
 - **Memoria interna.** En estas direcciones de memoria se almacenan los valores de los parámetros del propio PLC, como pueden ser fecha, hora, dirección IP, máscara de subred, periféricos conectados...
- **Fuente de alimentación.** Proporciona el voltaje necesario. La alimentación en general suele ser de 24V.

Respecto a la construcción física de los PLC, la mayoría predominante en el mercado son de **tipo modular**, es decir, estos están formados por una cantidad de módulos que pueden interconectarse entre sí. No obstante, existen otros tipos como pueden ser los PLC Nano o los PLC Compactos que incorporan todos los elementos anteriores en un mismo dispositivo.

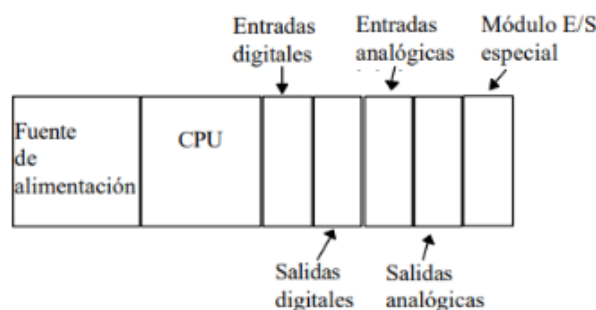


Figura 2.10: Esquema de un PLC tipo modular

Por otro lado, los PLC tienen dos modos de funcionamiento claramente destacados: modo RUN y modo STOP.

- **Modo RUN.** También llamado Modo de Ejecución. En este modo se ejecuta el programa almacenado en la memoria de programa citada y el PLC monitorea constantemente las entradas para detectar cambios. Si se detecta algún cambio, el PLC realiza las acciones necesarias en función del programa.

La ejecución en modo RUN se realiza de forma cíclica, ejecutando ciclos de 'scan' (entendiendo por 'scan', o ciclo de trabajo, al grupo de tareas que un autómatas ejecuta en un ciclo para llevar a cabo el control del proceso) de forma indefinida. El ciclo de trabajo más común tiene la siguiente estructura:

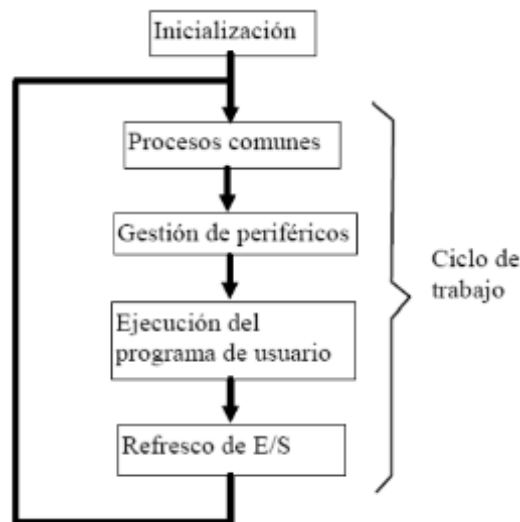


Figura 2.11: Esquema del ciclo de ejecución típico de un PLC

- **Modo STOP.** También llamado Modo de Programación. De este modo se genera el programa que se quiere ejecutar, y se traspa desde la herramienta software de edición al PLC para su ejecución.

2.3 Comunicaciones industriales

El hecho de incorporar elementos robóticos con el fin de automatizar los procesos productivos de las industrias, implica la incorporación de una gran variedad de dispositivos mecánicos, eléctricos y electrónicos encargados de su control y supervisión, que tienen que comunicarse para conseguir un buen funcionamiento de la planta.

Entendemos por **comunicaciones industriales** al conjunto de herramientas, métodos y sistemas que nos permiten que dichos elementos transmitan el mismo tipo de información, y de forma fiable, para conseguir controlar el desarrollo de un proceso industrial en tiempo real.

2.3.1 Niveles de comunicación en una red industrial

Una factoría se suele jerarquizar, desde el punto de vista de las redes de comunicaciones, en diferentes niveles en función de la **velocidad de respuesta y el volumen de datos**. El objetivo final es que la información fluya entre ellos.



Figura 2.12: Pirámide CIM [9]

- **Nivel de entradas y salidas.** Nivel en el que se sustenta la pirámide CIM; está formado por los elementos que están en “contacto” con las piezas del proceso industrial, como son los sensores para recoger información, y los actuadores para realizar las acciones pertinentes.
- **Nivel de campo y proceso.** Este nivel está constituido por los sistemas que adquieren la información de los sensores y actuadores y gestionan sus acciones, además de facilitar la transmisión de información al nivel superior a través del bus de campo. Si combinamos este nivel y el inferior tendremos un nivel de supervisión suficiente para un proceso industrial de una pequeña factoría.
- **Nivel de control.** Nivel formado por los elementos informáticos que controlan el proceso de automatización y que permiten la visualización de las diferentes tareas que conforman el mismo, mediante paneles virtuales como puede ser un SCADA. Para ello se emplean redes de área local (LAN) mediante cable *Ethernet*, mayoritariamente.



- **Nivel de gestión.** Nivel más alejado del proceso productivo que recopila información, normalmente en ordenadores, de todos los niveles inferiores con el objetivo de realizar estadísticas para tomar decisiones empresariales para la optimización del proceso. El tipo de comunicaciones que se emplean aquí no son tan robustas y con tiempos de respuesta tan cortos, como puede ser *WAN Ethernet*.

A medida que descendemos en la pirámide, disminuye la cantidad de información intercambiada pero aumenta la velocidad de transmisión de la misma.

2.3.2 Protocolos de comunicación industrial

Históricamente, uno de los mayores problemas con los que se ha encontrado el mundo de la industria automatizada es la integración de los diferentes componentes que son necesarios para realizar el proceso productivo.

Los dispositivos son puestos a la venta por diferentes fabricantes que emplean protocolos que no son iguales en muchos casos, lo cual complica el diseño de una red de comunicaciones que sea capaz de compatibilizar la información existente entre los diferentes sistemas.

Hoy en día se hace uso de un gran número de protocolos de redes de comunicación industrial. Vamos a hablar de los más relevantes.

- **Industrial Ethernet.** No es un protocolo como tal; se trata de un término genérico que se aplica a cualquier enfoque que trata de adoptar *Ethernet* a su uso en un entorno industrial. Los usos más destacados son [10]:

- **ProfiNet y ProfiBus.** Hablaremos de ellos más adelante en el apartado 2.3.3. ProfiNet (PN), por ser uno de los protocolos empleados en el proyecto.
- **ModBus.** Protocolo basado en el acceso al medio maestro-esclavo para la comunicación entre dispositivos dentro de una misma red. Existen diferentes “versiones” del protocolo, como son *ModBus* serial RS-232 o RS-485, o *ModBus TCP/IP* que incorpora la capa física *Ethernet*.
- **Ethernet IP.** Plantea la utilización de un protocolo de aplicación sobre la pila TCP/IP, por encima de *Ethernet*, denominada CIP (*Common Industrial Protocol*) empleada también en *DeviceNet*. No aporta mejoras desde el punto de vista de las restricciones temporales [10].



- **EtherCAT.** Protocolo creado originalmente para dotar a *Ethernet* de características en “tiempo real”, de tal manera que pueda ser empleado en aplicaciones de automatización en plantas industriales. El método de acceso al medio es mediante maestro-esclavo y no incorpora CIP [10].

- **Comunicación serial.** No es un protocolo, sino un sistema de comunicación bus muy sencillo que consiste en transmitir los datos de forma que los bits se envían uno a uno a través del mismo canal, en lugar de enviarlos todos juntos a la vez. Se implementa a través de protocolos como RS-232, RS-422 y RS-485 y la comunicación es bidireccional por el mismo cable. Algunos de los protocolos que usan este sistema son:
 - **AS-i.** Bus muy simple para automatización que combina potencia y datos sobre los mismos cables, orientado a la transmisión de datos asociados a señales binarias.
 - **DeviceNet (CAN + CIP).** Protocolo de comunicación basado en el bus de campo CAN, del que hablaremos más adelante, cuya principal característica es que añade sobre este el nivel de aplicación CIP, lo que facilita su integración con otras redes como *Ethernet/IP*. Está orientado a la industria de procesos [10].
 - **CANopen (CAN + nivel de aplicación).** También dedicaremos un apartado específicamente para este protocolo. Ver [apartado 2.3.4.CANopen](#).
 - **ModBus** también emplea comunicación serial.

2.3.3 ProfiNet (PN)

Se trata de un protocolo de comunicación con grandes ventajas para la automatización industrial, ofreciendo alta velocidad y rendimiento. Esto lo hace adecuado para aplicaciones que requieren una comunicación precisa en tiempo real. Está dividido en 3 clases o perfiles diferentes [10]:

- PROFINET Clase-C: este perfil destaca por soportar tráfico isócrono (IRT), que es la capacidad para asegurar que la información se va entregando de forma regular y sin retardos [10].
- PROFINET Clase-B. También se denomina PROFINET Real Time (PROFINET RT) pues incluye procedimientos para la gestión de tráfico en tiempo real [10].
- PROFINET Clase-A: permite la integración con *ProfiBus* mediante proxys y puentes *Ethernet*. No aborda restricciones temporales [10].

Como hemos comentado, PROFINET a su vez está pensado para implementar **ProfiBus**.

ProfiBus. Es un bus de campo que se emplea para la comunicación de los sensores y actuadores de la pirámide CIM, basada en la tecnología RS-485 y dividido en 3 perfiles: FMS, DP y PA.

- FMS (*FieldBus Message Specification*): Orientado al intercambio de información estructurada entre PLC's, controladores... Actualmente está obsoleto [10].
- DP (*Decentralized Peripheral*): Orientado a la conexión con dispositivos de campo. Está orientado a industrias genéricas discretas [10].
- PA (*Process Automation*): Es la versión intrínsecamente segura del DP, orientado a su uso en el control de procesos. Un mismo cable se emplea para alimentación y para transmisión de datos [10].

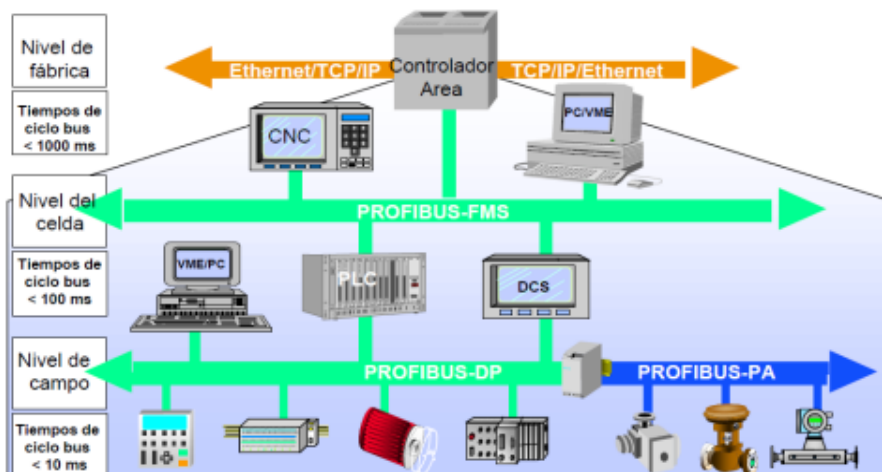


Figura 2.13: Perfiles FMS, DP y PA [10]

El bus *ProfiBus* está dividido en estaciones activas y estaciones pasivas. El método de acceso al medio consiste en un “bus con paso de testigo” (*Token Bus*) entre las estaciones activas, y maestro-esclavo (*Master-Slave*) entre las estaciones pasivas.

Es decir, el testigo se va pasando entre las estaciones activas de tal manera que cuando un usuario (una estación activa) del bus lo recibe, queda autorizado para actuar como maestro, que por sondeo pregunta a sus esclavos si desean transmitir información. El tiempo durante el cual puede hacer esto se denomina “tiempo de retención de testigo”.

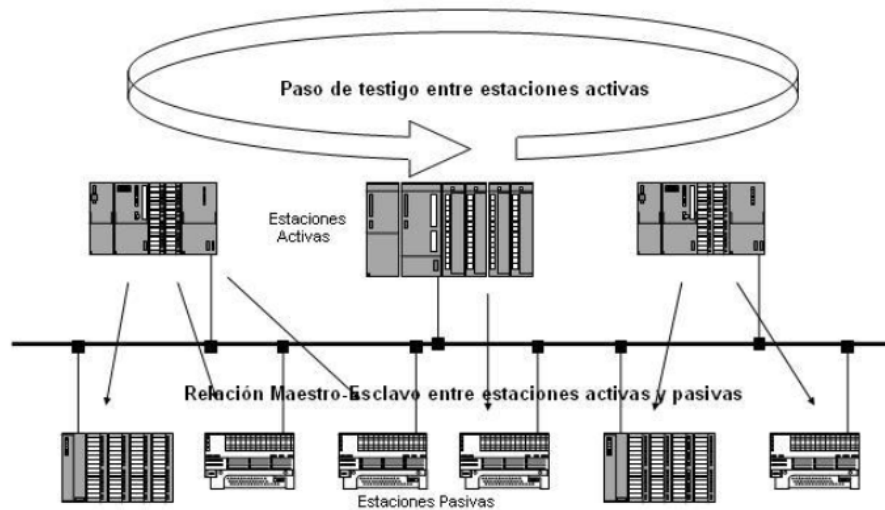


Figura 2.14: Método de acceso al medio de bus de campo de ProfiBus [10]

2.3.3.1 Funciones de seguridad PROFIsafe

PROFIsafe (PROFINET + safety) es una extensión del protocolo de comunicación industrial PROFINET empleada para aplicaciones de seguridad en funciones como la automatización industrial, que permite la transferencia segura de datos.

El papel de PROFIsafe es por tanto reducir al mínimo la posibilidad de un funcionamiento incorrecto de los sistemas de control, entendiéndose este como un error no detectado que pueda desembocar en una situación no segura para la maquinaria y para el personal [11].

PROFIsafe tiene una probabilidad de error no detectado **menor de 1 error por cada 10⁹ horas de operación**, es decir, cumple con el nivel de integridad de seguridad 3 (SIL3 - *Safety Integrity Level 3*), la cual exige menos de 1 error no detectado por cada 10⁷ horas de operación. Existen 4 niveles SIL.

Se menciona esta extensión del protocolo pues el PLC empleado en el proyecto es un autómata de seguridad. Los PLC de seguridad incorporan muchas funciones que no están disponibles en un autómata de propósito general, como pueden ser:

- Autodiagnóstico del *hardware* y *software* para detectar cualquier fallo interno, y en caso de que ocurra, actuar llevando la máquina a una situación segura.
- Monitoreo constante de los dispositivos conectados, lo que les permite detectar de manera casi inmediata cualquier evento peligroso.
- Protección contra acceso no autorizado, mediante por ejemplo contraseñas.

2.3.4 CANopen

CANopen es un protocolo de comunicaciones basado a su vez en el protocolo CAN (*Controller Area Network*). Es por ello que inicialmente se procede a hacer una breve introducción a este, con el objetivo de entender las bases sobre las que se sustenta, y posteriormente se detallará cómo se implementa la pila CANopen.

2.3.4.1 Introducción a CAN

Fue desarrollado por Bosch a mediados de los años 80 en el sector automovilístico con el objetivo de reducir el cableado de los vehículos, y desde los años 90 es estándar internacional (ISO 11898) [10]. Desde entonces es ampliamente usado en otro tipo de aplicaciones como equipos electrónicos o sistemas de automatización. Algunas de sus ventajas son:

- Es un sistema barato y ligero.
- Trabaja con el broadcasting: cuando un nodo conectado a la red manda un mensaje, este es recibido por todos los nodos a la vez.
- Cuenta con una buena gestión de errores en la comunicación, que permite comprobar si el mensaje enviado fue transmitido correctamente a su nodo.
- Tiene un sistema de prioridad basado en un identificador único en la red, de tal manera que cuanto menor es su valor, más prioridad tiene [10]

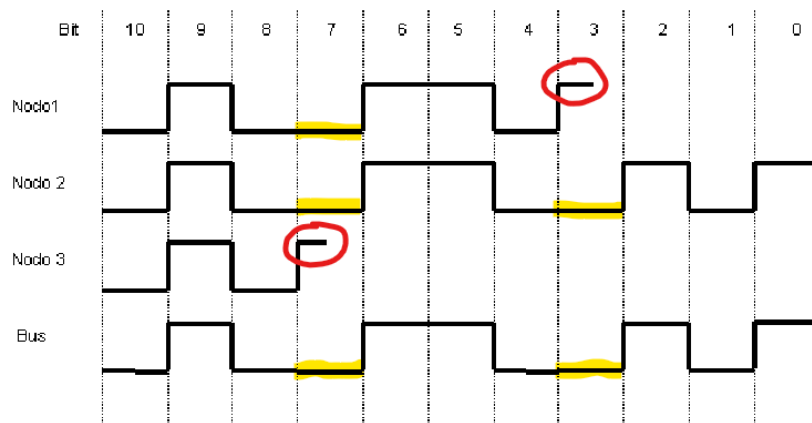


Figura 2.15: Acceso al bus simultáneo de tres nodos CAN [10]

2.3.4.2 Protocolo CANopen

Como se ha introducido previamente, CANopen es un estándar de comunicaciones basado en el protocolo CAN. Originalmente se desarrolló para el control de máquinas motrices, aunque rápidamente se extendió su uso en múltiples aplicaciones. Fue en 1994 cuando se publicó la primera especificación CANopen, la CiA 301 [12].

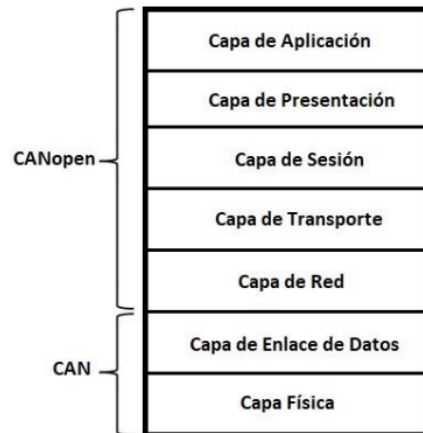


Figura 2.16: Protocolos CAN y CANopen en el modelo OSI

Su método de acceso al medio consiste en un maestro-esclavo en el cual existe un único maestro y uno o varios esclavos. El maestro siempre inicia las comunicaciones con los esclavos, además de supervisar el funcionamiento de estos y proporcionar información acerca de su estado.

A continuación se comentan algunos de los aspectos más importantes de este protocolo [12].

- **Object Dictionary.** Todo dispositivo CANopen debe incorporar un “Diccionario de objetos” que actúa como interfaz entre el bus CAN y la aplicación. Este diccionario es único y básicamente describe el funcionamiento del dispositivo que lo incorpora, mediante la información necesaria sobre las variables o tipos de datos que soporta el dispositivo.
- **Node ID.** Todo dispositivo conectado a una red CANopen debe tener un identificador numérico (*Node Identification*) que es único en la red, de tal manera que cuanto menor es su valor, más prioridad tiene.
- **Service Data Objects (SDO).** Objetos usados para configuración del dispositivo, así como leer y escribir datos poco relevantes del diccionario de objetos de un dispositivo conectado a la red. Es por ello que son mensajes de baja prioridad.
 - La comunicación para las SDOs es de cliente-servidor, de tal forma que si un cliente requiere información, enviará una solicitud a través de una SDO de COB-ID de 600h+NodeID (ver figura 2.17), mientras que la respuesta se enviará a través de un COB-ID de 580h+NodeID.

• **Process Data Objects (PDO).** Objetos usados para transmitir información en tiempo real sobre el proceso en cuestión. La comunicación para las PDOs es de un único productor (maestro) y uno o varios consumidores (esclavos). Su principal ventaja respecto a las SDOs es una mayor velocidad y prioridad.

- Existen dos tipos de PDOs: los TPDOs, que es información del productor a los consumidores; y las RPDOs, que son mensajes acerca del estado del proceso de los consumidores al productor.
- Así mismo, existen dos métodos de transmisión de las PDOs: síncrona mediante una señal SYNC; o asíncrona, mediante eventos o temporizadores.

Objeto de comunicación	COB-ID	Nodos esclavos
Control de nodo NMT	000h	Recepción solo
Sync	080h	Recepción solo
Emergencia	080h + NodeID	Transmisión
TimeStamp	100h	Recepción solo
PDO	180h + NodeID	1.Transmisión PDO
PDO	200h + NodeID	1.Recepción PDO
PDO	280h + NodeID	2.Transmisión PDO
PDO	300h + NodeID	2.Recepción PDO
PDO	380h + NodeID	3.Transmisión PDO
PDO	400h + NodeID	3.Recepción PDO
PDO	480h + NodeID	4.Transmisión PDO
PDO	500h + NodeID	4.Recepción PDO
SDO	580h + NodeID	Transmisión
SDO	600h + NodeID	Recepción
Monitorización de nodo NMT (node guarding/heartbeat)	700h + NodeID	Transmisión
LSS	7E4h	Transmisión
LSS	7E5h	Recepción

Figura 2.17: Descripción de los diferentes COB-ID en CANopen

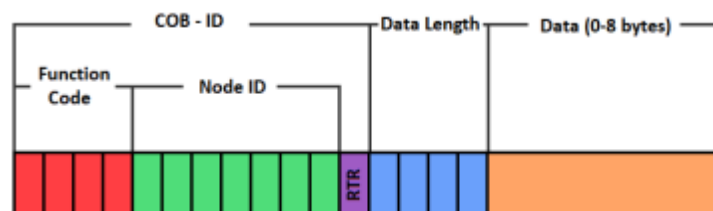


Figura 2.18: Estructura de la trama de datos en CAN [COB-ID]

• **Network Management (NMT).** Parte del protocolo encargada de garantizar el correcto funcionamiento de la red, que consiste en una máquina de estados que define el comportamiento del dispositivo en cuestión. Cuenta con tres estados principales:

- Inicialización (*Initializing*): Estado de inicio tras encender un dispositivo, durante el cual se transmite un mensaje de encendido (*boot-up*).
- Pre-Operacional (*Pre-Operational*): Durante este estado el dispositivo puede comunicarse con cualquier tipo de mensajes excepto con PDOs.
- Operacional (*Operational*): Cuando el dispositivo se encuentra en este estado ya está debidamente configurado y en funcionamiento, listo para enviar y recibir PDOs.
- Detenido (*Stopped*): La comunicación está inactiva y no se pueden transmitir SDOs ni PDOs, tan solo NMT.

Además, para implementar el protocolo CANopen es estrictamente necesario elegir una forma de monitorización de los nodos, es decir, de realizar un seguimiento de los dispositivos conectados para comprobar periódicamente si estos están respondiendo. Las dos formas posibles son:

- *Heartbeat*: Consiste en que los nodos envían un mensaje o pulso (*Heartbeat*) cada cierto período, de tal manera que si el maestro detecta que uno de los pulsos no ha sido recibido, interpretará que ha existido un fallo con el nodo en cuestión.
- *Node Guarding*: En este caso es el máster el que solicita información periódicamente de cada uno de los nodos vía RTR. De la misma manera anterior, si uno de los nodos no respondiera a la solicitud sería interpretado como que existe un fallo en el protocolo.

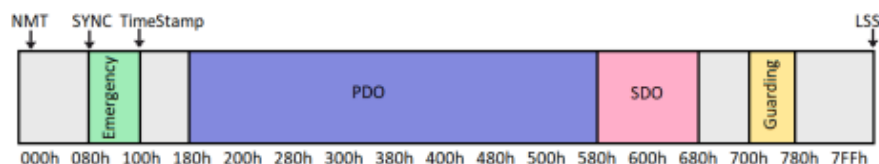


Figura 2.19: Espacio de identificación de CANopen



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática

Capítulo 3. ESTADO DEL ARTE

3.1 Antecedentes

Por la parte del trabajo asociada con la automatización industrial y la robótica, en la **Universidad de Valladolid** se han realizado una gran variedad de proyectos en este campo. Se citan, a continuación, algunos de ellos cuya estructura ha servido de inspiración y referencia a la hora de la redacción.

En el año 2020, **José Delgado García** realizó un una isla de formación para el manejo de un robot M-10ia de Fanuc mediante un autómeta S7-1200 de Siemens a través de programación en Step7, y basada en los buses de campo Profinet y Profisafe. La creación de esta aula de formación abarca desde el cableado eléctrico de los distintos elementos de la instalación, como son los armarios eléctricos, hasta el diseño mecánico y construcción de los diversos útiles de trabajo. Destacar que José Delgado García realizó su Trabajo de Fin de Grado en la misma empresa, ICARE Automation [13].



Figura 3.1: Aula de formación correspondiente al TFG de José Delgado [13]

En el año 2023, **Jorge Villa González** presentó su desarrollo de una célula robotizada controlada por un autómata programable (PLC) y gestionada a nivel de usuario con una interfaz hombre-máquina. La parte correspondiente a la comunicación fue realizada mediante el estándar industrial OPC UA [14].

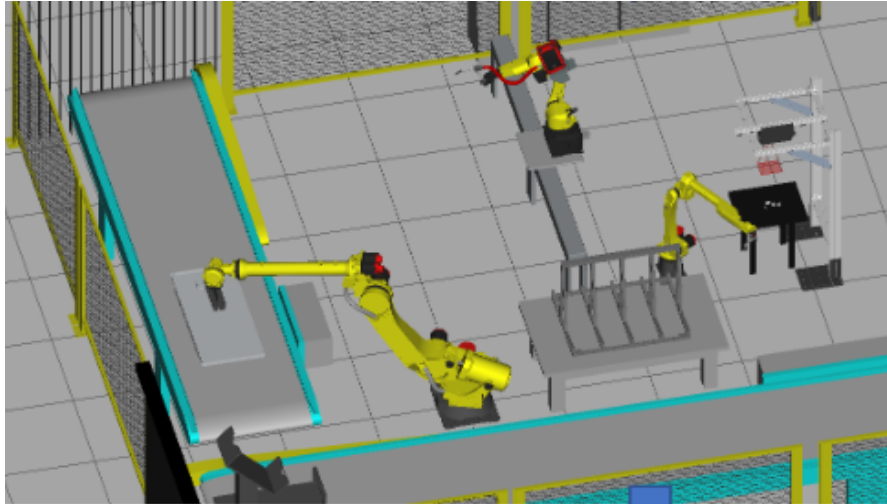


Figura 3.2: Instalación robotizada correspondiente al TFG de Jorge Villa González [14]

No obstante, por la parte de las comunicaciones de los diferentes componentes mediante CANopen se ha recurrido a trabajos realizados en **diferentes universidades del país**, debido a la escasez de proyectos que incorporan este protocolo en la Universidad de Valladolid.

En el año 2012 **Roberto Gámez Pérez**, entonces estudiante de la Universidad Carlos III de Madrid, realizó su Trabajo de Fin de Grado acerca del diseño e implementación de un conjunto de funciones basadas en CAN bus. Dichas funciones se encargan de controlar un servomotor que va montado en el interior de la base del robot *ASIBOT*, el cual fue desarrollado por el grupo de *Robotics Lab* del Departamento de Ingeniería de Sistemas y Automática de dicha Universidad [15].

En el año 2017 **Juan Moliner Malaxechevarría**, entonces estudiante de la Universidad Politécnica de Madrid, realizó su Trabajo de Fin de Grado sobre el robot *DORIS*. Su proyecto permitía detectar y separar errores en el funcionamiento del mismo, mediante la comunicación con los drivers de potencia de los servomotores del robot, a través de una biblioteca de funciones CANopen desarrollada por completo dentro del mismo TFG [16].



En el año 2018 **Natalia Puente Carreño**, entonces estudiante de la Universidad Carlos III de Madrid, realizó su Trabajo de Fin de Grado sobre un prototipo de cuello blando para un robot. Uno de los objetivos principales del proyecto fue el desarrollo de una librería de funciones que permitiese el envío de los mensajes pertinentes para que el manejo por el usuario se efectúe de forma sencilla y eficiente. Para el desarrollo de las mismas se utilizó CANopen, más concretamente los estándares CiA402 y CiA301, de forma que la librería fuera compatible con cualquier dispositivo que emplee este protocolo [17].

Finalmente en el año 2022 **Ricardo Martínez Martín**, entonces estudiante de la Universidad Internacional de la Rioja, realizó su Trabajo de Fin de Máster en su totalidad sobre el protocolo CANopen, desarrollando una herramienta que utilice las vulnerabilidades descubiertas y que siguen estando presentes hoy en día en el bus CAN, así como analizar diferentes tipos de ataques que se realizaron sobre los propios servicios y protocolos ofrecidos y descritos por el estándar CiA. Este TFM permite recopilar información general de interés acerca del protocolo [18].

3.2 Análisis del problema

Como se ha comentado previamente, el presente Trabajo de Fin de Grado se ha realizado durante el período de prácticas en la empresa ICARE Automation. El objetivo de esta era conseguir un vehículo de guiado automático (AGV) similar a los que se incorporan en grandes factorías como Renault España.

La tarea que se pretende desarrollar consiste en un estudio previo, montaje, conexionado, programación y puesta en marcha de diferentes dispositivos eléctricos y electrónicos, con el objetivo de producir un prototipo de un futuro AGV.

También se abordarán los problemas de comunicación existentes entre el SIMATIC S7-1200 1215 FC y el driver C5-E-2-09 de Nanotec, a través de la pasarela SIMATIC PN/CAN Link y mediante el protocolo CANopen.

3.3 Nanotec

Nanotec Electronic GmbH & Co. KG es una empresa alemana fundada en 1991 que se especializa en la fabricación de productos y sistemas de automatización de alta calidad, incluyendo motores, controladores, sensores y accionamientos.

En 2010, Nanotec lanzó su línea de productos **Plug and Drive**, que ofrece una solución de automatización integral, desde la mecánica y la electrónica hasta el

software de control. Esta línea de productos proporciona una solución fácil de usar y altamente integrada para una amplia variedad de aplicaciones.

Desde su lanzamiento, la línea *Plug and Drive* de Nanotec ha ganado una gran aceptación en la industria gracias a su alta calidad y confiabilidad. La compañía continúa innovando y expandiendo su línea de productos para satisfacer las necesidades cambiantes de sus clientes en el mercado.

3.3.1 Plug & Drive Studio (PnDS)

Plug and Drive Studio (PnDS) es un *software* gratuito desarrollado por Nanotec para el control de sus drivers de una manera sencilla e intuitiva. El usuario puede seleccionar y configurar diferentes modos de funcionamiento a través de un menú de navegación muy intuitivo que reduce el número de entradas del programa a unos pocos parámetros, resultando en unos tiempos de set-up muy cortos.

Vamos a ir analizando el *software* en su versión 1.1.0. Una vez instalado e iniciado, la primera pantalla que vemos es la de la *figura 3.3*.

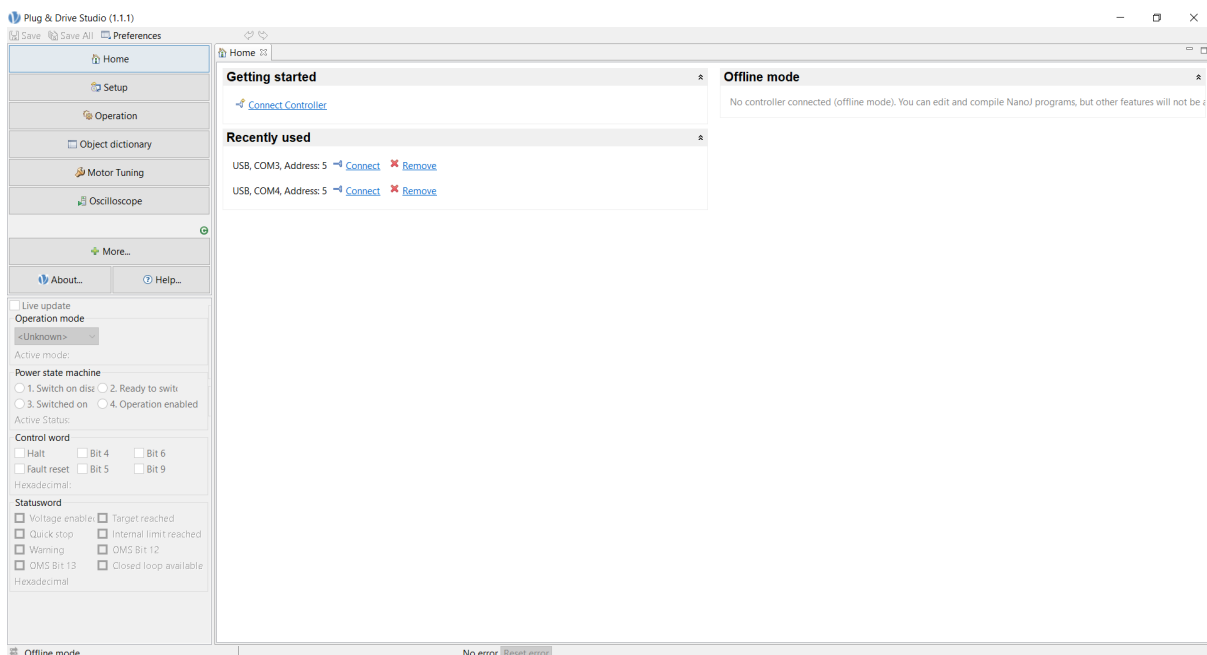


Figura 3.3: Menú principal del PnDS

Si hacemos zoom en la pantalla, la primera opción que nos aparece es la de la *figura 3.4*.

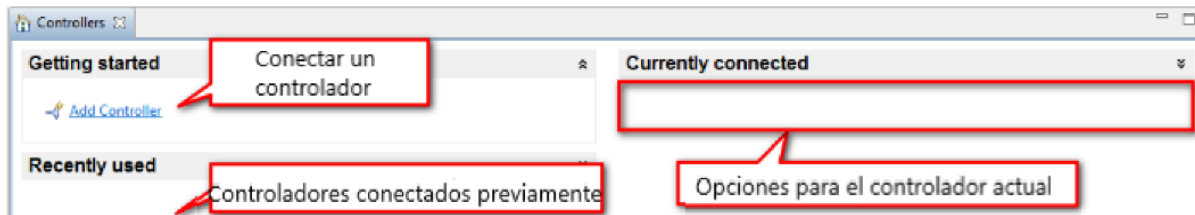


Figura 3.4: Ventana de Home del PnDS

Todos los menús del controlador son accesibles desde las pestañas de la figura 3.5. Vamos a irlos comentando.

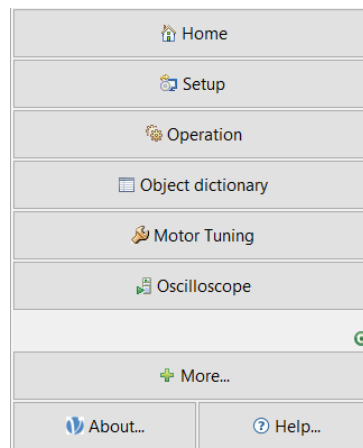


Figura 3.5: Menú del controlador en PnDS (I)

- **Home (conectar y desconectar un controlador)**
Abre la ventana “Home” para conectar o desconectar un controlador. Por el momento un único controlador puede ser conectado a PnDS a la vez. Si se quiere usar otro controlador, se debe hacer click en “Disconnect” y la configuración de los parámetros del mismo se guardarán en el disco. El controlador desconectado aparecerá ahora en la página de bienvenida de la figura 3.3, en la sección de “Recently used” [19].
- **Setup**
Permite ajustar los parámetros generales del controlador y del motor. Ver el apartado 3.3.1.1 Setup para más información [19].
- **Operation**
Permite operar el motor en los diferentes modos y ajustar los correspondientes parámetros. Ver el apartado 3.3.1.2 Operation para más información [19].

- Object dictionary

Muestra todos los objetos del diccionario con su valor correspondiente. Ver el apartado 3.3.1.3 *Object dictionary* para más información [19].

- Motor turning

Permite ajustar los parámetros del controlador del motor para que se ajuste de manera óptima a las necesidades de la aplicación [19].

- Oscilloscope

Muestra los valores de los objetos del diccionario a lo largo del tiempo [19].

- NanoJ (esta opción, debido al tamaño de la pantalla, no aparece directamente en *la figura 3.5*)

Permite escribir programas que son ejecutados en el controlador [19].

- More+

El resto de herramientas están listadas en este menú. Por ejemplo, una línea de comando para leer y escribir objetos en el objeto de diccionario (*Object Dictionary script console*) [19].

Además existe otro cuadro de opciones que nos muestra el modo de operación en el que nos encontramos, el estado actual del controlador, los bits de la palabra de control y de estado, y los posibles errores.

The screenshot displays a control interface for a motor driver. It includes several sections:

- Live update:** A checked checkbox.
- Operation mode:** A dropdown menu set to 'Profile velocity'. Below it, 'Active mode: Profile Velocity' is displayed.
- Power state machine:** Four radio buttons representing states: '1. Switch on dis' (selected), '2. Ready to swit', '3. Switched on', and '4. Operation enabled'. Below this, 'Active Status: Switch on disabled' is shown.
- Control word:** Six checkboxes for 'Halt', 'Bit 4', 'Bit 6', 'Fault reset', 'Bit 5', and 'Bit 9'. The 'Hexadecimal' value is '0x0000'.
- Statusword:** Eight checkboxes for 'Voltage enable', 'Target reached', 'Quick stop', 'Internal limit reached', 'Warning', 'OMS Bit 12', 'OMS Bit 13', and 'Closed loop available'. The 'Hexadecimal' value is '0x8340'.
- Errors:** A list of checkboxes for 'General error', 'Communication', 'Current', 'Mode specific', 'Voltage', 'Wrong direction', and 'Temperature'.
- NanoJ:** A dropdown menu currently showing 'STOPPED'.

Figura 3.6: Menú del controlador en PnDS (II)

3.3.1.1 Setup

Todos los parámetros que pueden ser modificados en este apartado se encuentran por categorías. Cada categoría está representada por una etiqueta. Cuando se realiza un cambio en la interfaz del programa, el nuevo valor no se aplica directamente al controlador. En su lugar, un icono aparece indicando en qué etiqueta se han producido cambios. El mismo icono aparece también en el botón de “Apply”, que es el botón que se debería de pulsar a continuación a menos que se quieran aplicar diferentes cambios a la vez [19].

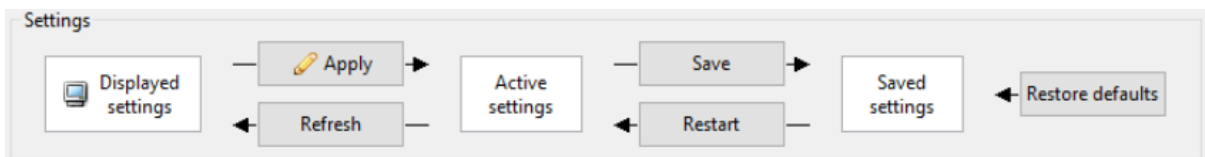


Figura 3.7: Menú Setup de PnDS (I)

Aplicar cambios implica mandar los nuevos valores al controlador, convirtiéndose en “Active settings”. En el momento en el que queramos salvar estos valores, se pueden hacer permanentes pulsando el botón de “Save” [19].

Cada campo en las etiquetas del modo “Setup” representa un objeto o parte de uno en el diccionario de objetos. Algunos de los campos no se pueden modificar debido a que son de solo lectura, es decir, sus valores están derivados del valor de otros objetos, o simplemente no son relevantes. Si colocamos el cursor por encima de las etiquetas se nos muestra el índice y subíndice del objeto al que representa.

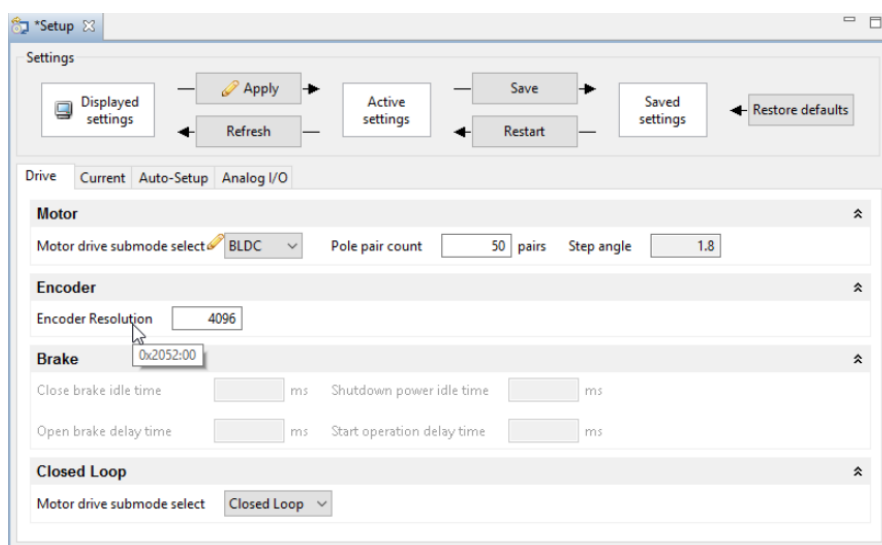


Figura 3.8: Menú Setup de PnDS (II)

3.3.1.2 Operation

Configurar el modo de operación del controlador es similar a cambiar su “Setup”. Existen 4 modos de operación: Posición, velocidad, torque y *homing*.

El primero de ellos, “**Profile Position**”, nos permite detener el movimiento del motor cuando este alcanza una determinada posición. Este modo de funcionamiento incorpora otras funcionalidades, bastante intuitivas y cuya explicación se resumen en su traducción del inglés, como son las de la *figura 3.9*.

Figura 3.9: Opción de “Profile Position” de PnDS

Destacar las siguientes:

- Posición actual, gracias al encoder incorporado.
- Posibilidad de añadir un “Set Point”, con diferentes modos de alcanzarlo (ver *figura 3.9*).
- Contar pulsos del encoder de manera relativa o de manera absoluta.
- Sentido de giro del motor.

Figura 3.10: Opción de “Profile Position” de PnDS

Las funcionalidades de control de velocidad y aceleración de “*Profile Position*” también están disponibles en el modo de funcionamiento de “*Profile Velocity*” de la *figura 3.11*. El programa nos permite jugar con parámetros como la velocidad que se quiere alcanzar, la aceleración y deceleración y los valores límite de estas.

The screenshot shows the 'Profile Velocity' control interface. It includes the following sections and parameters:

- Current Motor Status:** Velocity demand value (0), Velocity actual value (0).
- Acceleration and Velocity Limits:** Motion profile type (trapezoidal ramp), Target velocity (1000), Profile acceleration (750), Profile deceleration (500), Quick stop deceleration (5000), Max acceleration (5000), Max deceleration (5000).
- Velocity Windows:** Velocity window (30), Velocity window time (0 ms), Polarity (Inverted).
- Ramp:** Motor drive submode select (Velocity based ramp).

Figura 3.11: Opción de “*Profile Velocity*” de PnDS

Al igual que en los casos anteriores, el menú de “*Profile Torque*” permite establecer un valor de torque a alcanzar por el motor, entre otras opciones.

The screenshot shows the 'Profile Torque' control interface. It includes the following sections and parameters:

- Current Motor Status:** Torque demand (0 %*Nominal current), Torque actual value (-1 %*Nominal current).
- Torque Limits:** Target torque (0 %*Nominal current), Max torque (0 %*Nominal current).
- Torque Extras:** Torque slope (0 %*Nominal current), Motor drive submode select (Velocity mode).
- Setup:** Nominal current (6666 mA).

Figura 3.12: Opción de “*Profile Torque*” de PnDS

Y finalmente, la opción de “Homing” del programa. Este modo de funcionamiento mueve el motor en una dirección específica hasta que alcanza un límite físico o un sensor, lo que indica que ha alcanzado el punto de origen o “home”. Una vez se ha detectado este punto de origen, el sistema puede utilizar esta información para controlar la posición y el movimiento futuro del motor con mayor precisión.

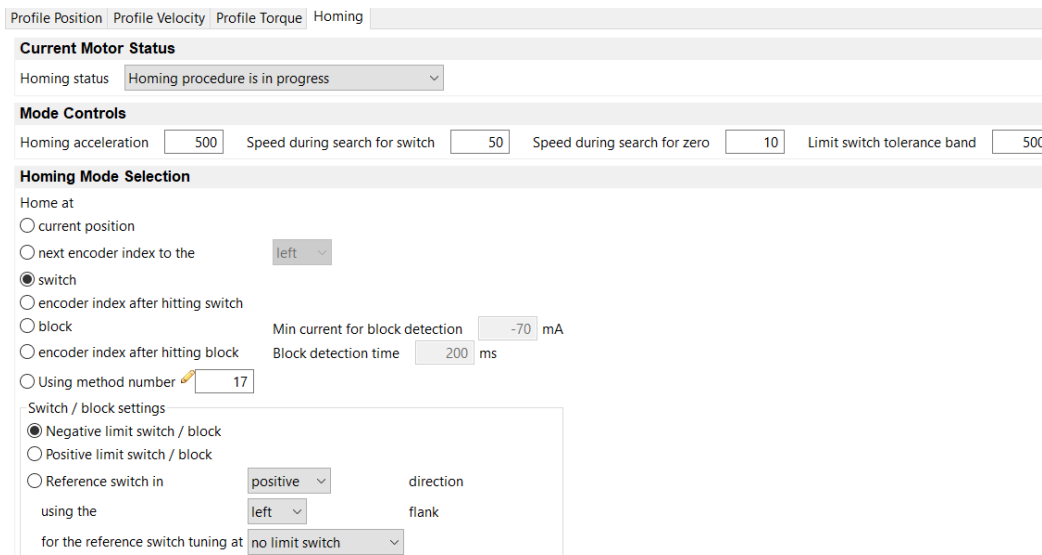


Figura 3.13: Opción de “Homing” de PnDS

3.3.1.3 Object dictionary

La tabla del diccionario de objetos hace una lista de todas las entradas de objetos con sus respectivos nombres, valores, tamaño (tipo), número de índice y subíndice y tipo de acceso. Solo se pueden editar los valores en binario, decimal o hexadecimal.

Solo las entradas con tipo de acceso de escritura o lectura/escritura son editables, y los valores serán escritos en el momento de pulsar la tecla *Enter* del teclado. Si se quieren refrescar los valores de la tabla por completo, se debe hacer *click* en el botón con dos flechas en forma de círculo [19].

Description	Index	Sub-Index	Access	Type	Value	Binary Value	Decimal Value
1 Error Register	1001	00	read only	UNSIGNED8	00h	0000	0
2 Pre-defined Error Field	1003	00	read/write	UNSIGNED8	00h	0000 0000	0
3 COB-ID Sync	1005	00	read/write	UNSIGNED32	00000080h	0000 0000 0000 0000 0000 0000 1000 0000	128
4 Synchronous Window Length	1007	00	read/write	UNSIGNED32	00000000h	0000 0000 0000 0000 0000 0000 0000 0000	0
5 Manufacturer Device Name	1008	00	read only	VISIBLE_STRING	PD4-C5918M4204-E-08		
6 Manufacturer Hardware Version	1009	00	read only	VISIBLE_STRING	W005b		
7 Manufacturer Software Version	100A	00	read only	VISIBLE_STRING	FIR-v1544-B219678		
8 Guard Time	100C	00	read/write	UNSIGNED16	0000h	0000 0000 0000 0000	0
9 Live Time Factor	100D	00	read/write	UNSIGNED8	00h	0000 0000	0
10 Store Parameters	1010	00	read only	UNSIGNED8	06h	0000 0110	6
11 Store Parameters - Save All Parameters To Non-volatile Memory	1010	01	read/write	UNSIGNED32	00000001h	0000 0000 0000 0000 0000 0000 0000 0001	1

Figura 3.14: Menú Object Dictionary de PnDS (I)

Se puede realizar un filtrado de los objetos mostrados en la tabla haciendo *click* con el botón derecho del ratón en cualquier sitio de la tabla y seleccionando la opción “*Select Predefined Filter*”.

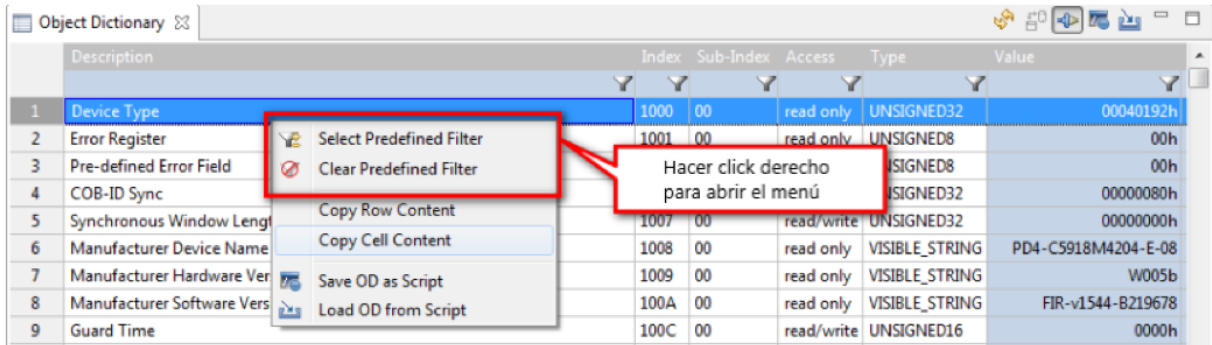


Figura 3.15: Menú Object Dictionary de PnDS (II)

En la parte superior de la ventana emergente se puede seleccionar el filtro. En la parte inferior se muestra una lista de entradas recogidas por este.

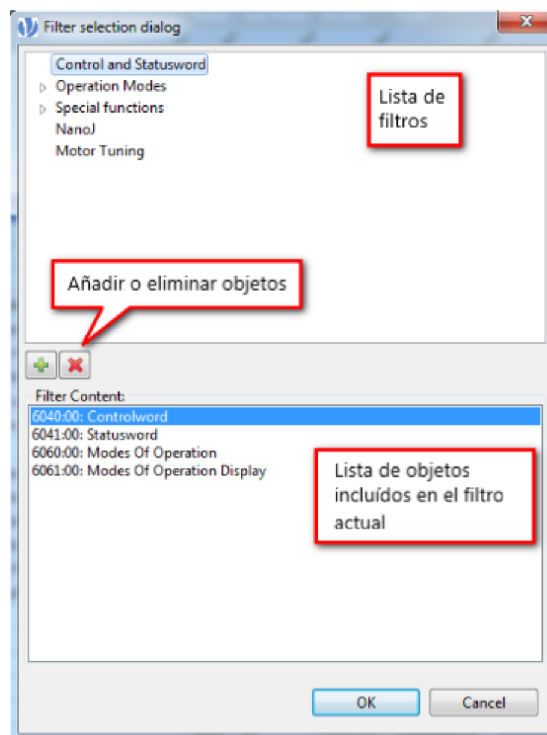


Figura 3.16: Menú Object Dictionary de PnDS (III)

Para eliminar el filtro se debe hacer *click* derecho en cualquier sitio de la tabla y seleccionar la opción “*Clear Filter*”.

También se pueden hacer búsquedas concretas mediante la barra de buscador de la parte superior, como por ejemplo los objetos relacionados con parámetros CAN.

	Description	Index	Sub-Index	Access	Type	Value
	CAN					
1	CANopen Baudrate	2005	00	read/write	UNSIGNED8	88
2	CANopen Config - Highest sub-index supported	2007	00	read only	UNSIGNED8	01
3	CANopen Config - BL Config	2007	01	read/write	UNSIGNED32	00000000
4	CANopen NodeID	2009	00	read/write	UNSIGNED8	01

Figura 3.17: Menú Object Dictionary de PnDS (IV)

La tabla entera puede ser salvada en el PC y cargada otra vez. Los botones para llevar a cabo estas acciones son los de la figura 3.18.

	Description	Index	Sub-Index	Access	Type	Value
1	Error Register	1001	00	read only	UNSIGNED8	0h
2	Pre-defined Error Field	1003	00	read/write	UNSIGNED8	0h
3	COB-ID Sync	1005	00	read/write	UNSIGNED32	00000080h
4	Synchronous Window Length	1007	00	read/write	UNSIGNED32	00000000h
5	Manufacturer Device Name	1008	00	read only	VISIBLE_STRING	PD4-C5918M4204-E-08
6	Manufacturer Hardware Version	1009	00	read only	VISIBLE_STRING	W005b
7	Manufacturer Software Version	100A	00	read only	VISIBLE_STRING	FIR-v1544-8219678
8	Guard Time	100C	00	read/write	UNSIGNED16	0000h
9	Live Time Factor	100D	00	read/write	UNSIGNED8	00h
10	Store Parameters	1010	00	read only	UNSIGNED8	06h
11	Store Parameters - Save All Parameters To Non-volatile Memory	1010	01	read/write	UNSIGNED32	00000001h
12	Store Parameters - Save Communication Parameters To Non-volatile Memory	1010	02	read/write	UNSIGNED32	00000001h
13	Store Parameters - Save Application Parameters To Non-volatile Memory	1010	03	read/write	UNSIGNED32	00000001h
14	Store Parameters - Save Customer Parameters To Non-volatile Memory	1010	04	read/write	UNSIGNED32	00000001h

Figura 3.18: Menú Object Dictionary de PnDS (V)

Existe un menú mencionado previamente que es el “Object Dictionary script console”, que consiste en una línea de comandos donde leer y escribir los valores de los objetos. El archivo a guardar de la figura 3.18 emplea el mismo formato que la consola, por lo que una alternativa para cargar el archivo es copiar el contenido o parte de él en la misma [19].

3.4 Siemens

Siemens AG es una empresa alemana fundada en 1847 por *Werner von Siemens* y *Johann Georg Halske* en Berlín. Desde entonces, la compañía ha crecido y se ha expandido a nivel mundial, convirtiéndose en la mayor empresa de fabricación industrial de Europa y una de las empresas líderes en tecnología y automatización industrial. Su área de aplicación se extiende a 4 sectores: industrial, energético, de

infraestructuras y ciudades pero es la sección de **automatización industrial** la más destacada.

En las décadas de 1960 y 1970, Siemens desarrolló una serie de innovaciones en automatización industrial, incluyendo el primer controlador programable de lógica (PLC) y el primer sistema de control distribuido (DCS). Hoy en día, la sección de automatización industrial ofrece una amplia gama de productos y soluciones para el sector, incluyendo controladores, sistemas de visualización, accionamientos y motores, y *software* de automatización. La compañía también ofrece servicios de consultoría y soporte para ayudar a sus clientes a implementar y mantener sistemas eficientes y efectivos.

3.4.1 TIA Portal

TIA Portal (*Totally Integrated Automation Portal*) es una plataforma de *software* de ingeniería de automatización desarrollada por Siemens. Es una herramienta integral que permite diseñar, programar, configurar y mantener sistemas de automatización de manera eficiente, como PLCs, interfaces hombre-máquina, accionamientos y actuadores... TIA Portal permite además conectar dichos dispositivos en una misma vista de proyecto y configurarlos de manera adecuada según las necesidades de esta.

Para toda versión de TIA Portal existen dos tipos de vistas: vista del portal y vista del proyecto (*Portal View* y *Project View* en inglés).

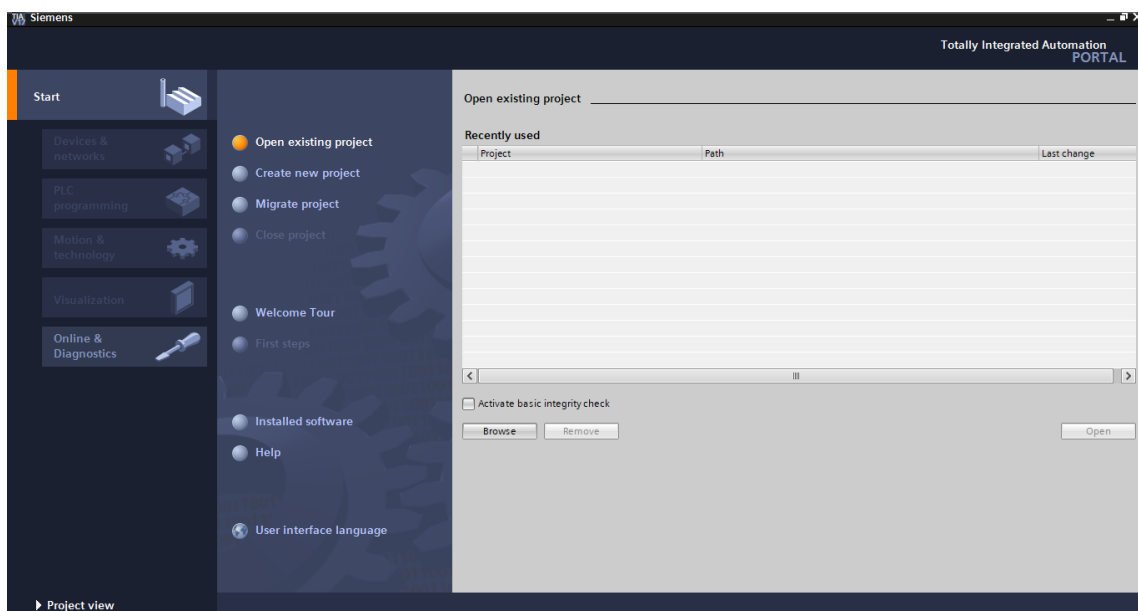


Figura 3.19: TIA Portal V17. Vista del portal

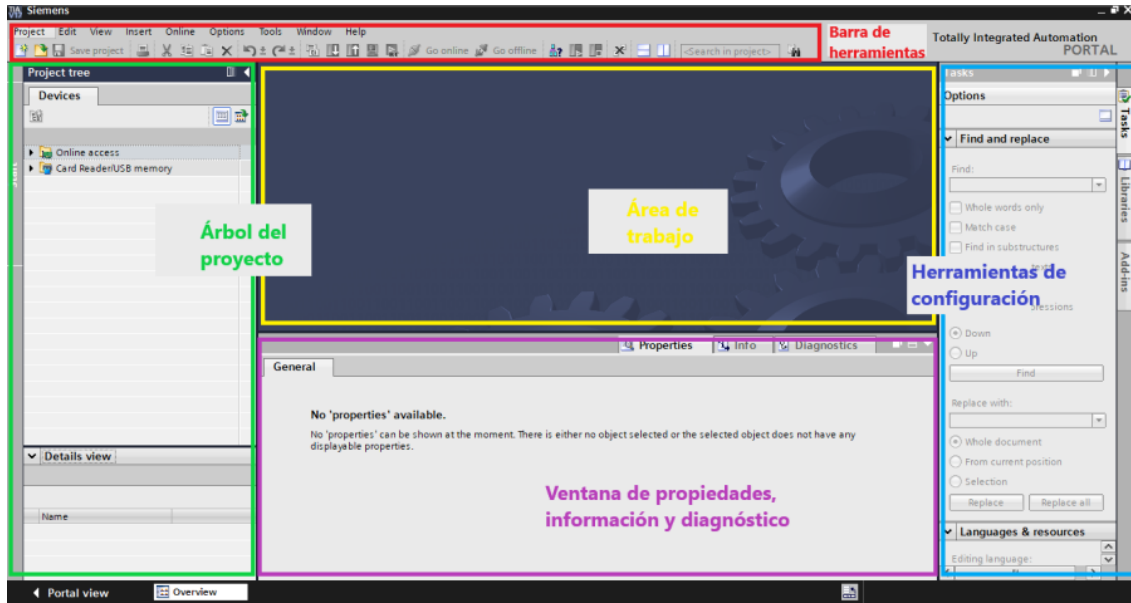


Figura 3.20: TIA Portal V17. Vista del proyecto

La vista del portal (figura 3.19) se suele emplear para crear nuevos proyectos y añadir los componentes usados o bien abrir proyectos existentes. Si creamos un proyecto nuevo debemos darle un nombre y una ruta para almacenarlo. La plataforma dispone de una gran base de datos con todos los dispositivos de Siemens disponibles, de entre los cuales deberán agregarse los que se quieran utilizar, especificando su versión de *firmware*. Cuando se hayan agregado, se verán en el propio árbol del proyecto (figura 3.20) y estarán listos para su configuración/programación.

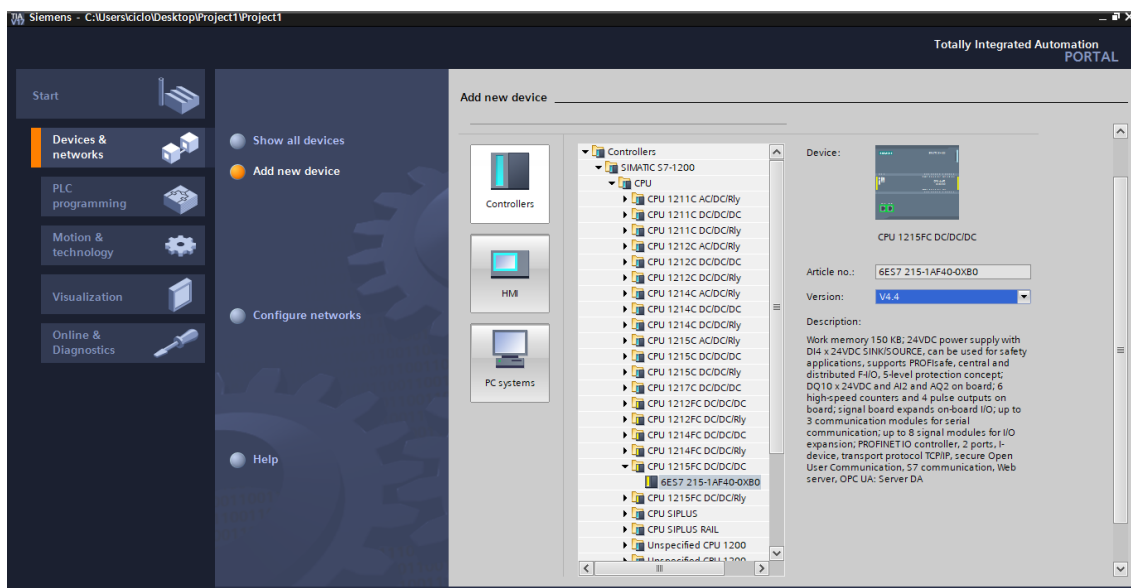


Figura 3.21: TIA Portal V17. Añadiendo el PLC y la CPU utilizados

La vista del proyecto (*figura 3.20*), desde la cual se trabaja, está dividida en diferentes secciones cuyas funcionalidades más importantes son las siguientes:

• Barra de Herramientas.

En **verde** tenemos las funcionalidades relacionadas con el proyecto actual; de izquierda a derecha: crear proyecto, abrir proyecto, guardar proyecto, cortar, copiar, pegar, eliminar.

En **rojo** tenemos las funcionalidades relacionadas con con la simulación o funcionamiento real del dispositivo; de izquierda a derecha: compilar, cargar en dispositivo, subir de dispositivo, iniciar simulación, establecer/deshacer conexión *online* (para poder visualizar el valor de las variables del dispositivo en tiempo real).

En **azul** tenemos las funciones que tienen que ver con la puesta en marcha/pausa del dispositivo; de izquierda a derecha: ver dispositivos accesibles, poner CPU en modo *RUN*, poner CPU en modo *STOP*.



Figura 3.22: TIA Portal V17. Barra de herramientas

• Árbol del Proyecto.

Muestra la estructura jerárquica de un proyecto. Es decir, muestra todos los elementos de este, como dispositivos, redes, programas, bloques de función, entre otros. Se utiliza para organizar y gestionar todos los elementos del proyecto en una estructura clara y accesible.

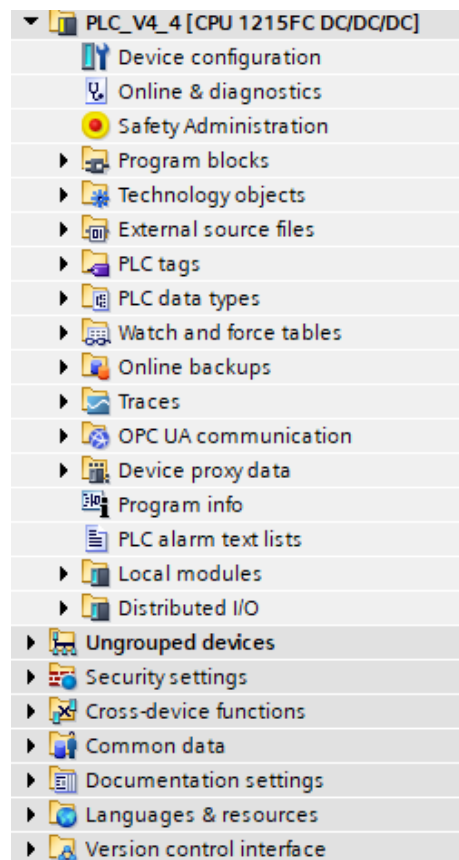
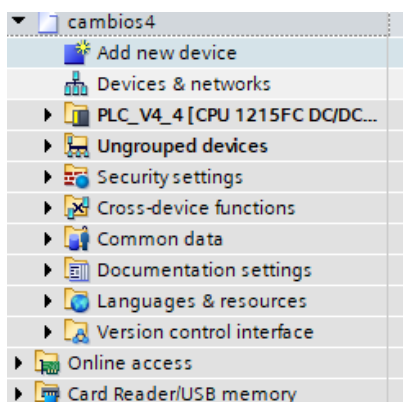


Figura 3.23: TIA Portal V17. Árbol del proyecto

• **Área de Trabajo.**

Es el espacio principal de trabajo en TIA Portal donde se pueden crear, programar y configurar los diferentes elementos del proyecto. En esta área se encuentran los editores de programación, las pantallas de configuración de dispositivos y redes, y las pantallas de visualización y monitoreo. Es el espacio donde se realizan la mayoría de las tareas y donde se puede interactuar con los diferentes elementos del proyecto.

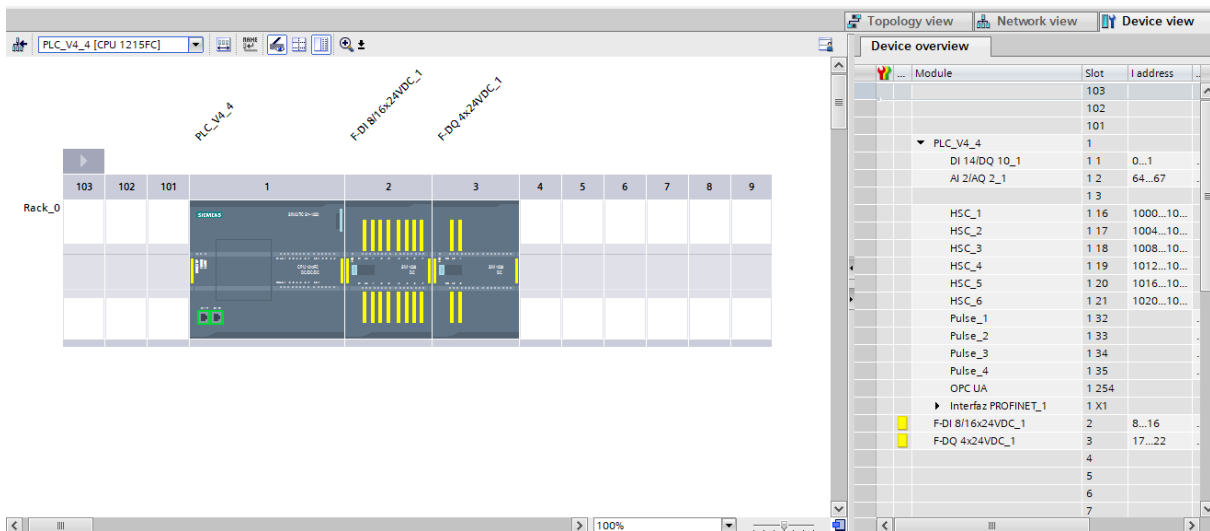


Figura 3.24: TIA Portal V17. Área de trabajo (I)

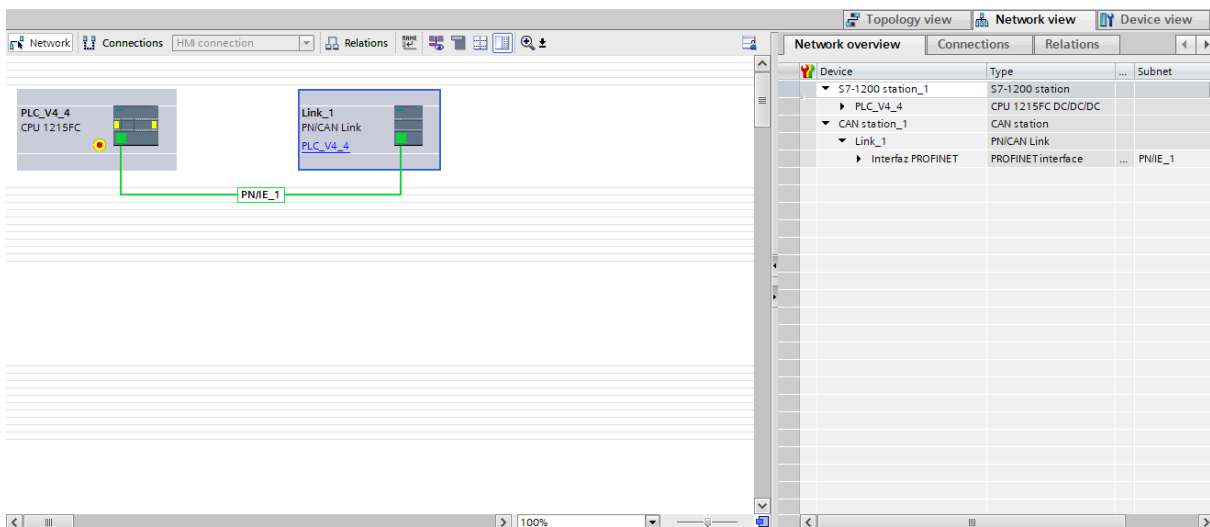


Figura 3.25: TIA Portal V17. Área de trabajo (II)

• Herramientas de Configuración.

En esta zona aparecen una gran variedad de funciones según el dispositivo con el que se esté trabajando. En el caso común de un PLC, aparecerán las instrucciones más empleadas en lenguaje *ladder*; si estamos trabajando con la pasarela, aparecerán diferentes tipos de configuración (*ver apartado 3.5 PN/CANopen Link*).

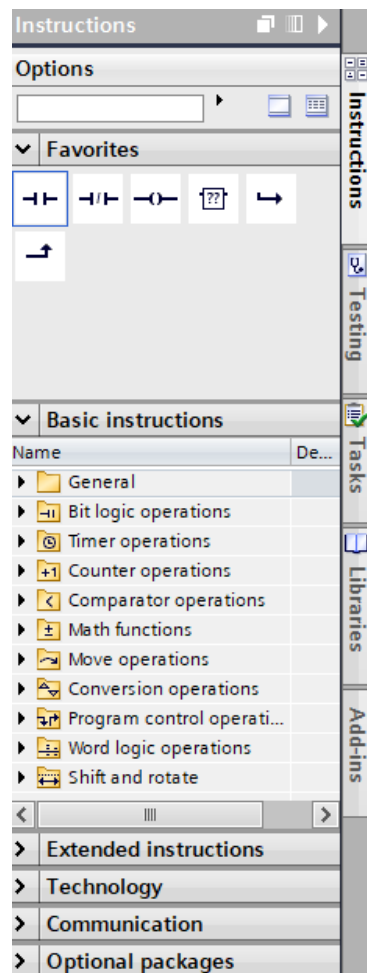


Figura 3.26: TIA Portal V17. Herramientas de configuración

• Ventana de Propiedades, Información y Diagnóstico.

- **Propiedades:** las propiedades que se muestran en la ventana de propiedades dependen del tipo de objeto que se ha seleccionado.
- **Información y diagnóstico:** son herramientas que se utilizan para obtener información sobre el estado y el rendimiento del proyecto y de los dispositivos y redes en él. Estas herramientas pueden incluir la visualización de mensajes de error, la visualización de información de diagnóstico de dispositivos, entre otras. Estas herramientas son útiles para solucionar problemas y mejorar el rendimiento del proyecto.

!	Path	Description	Go to	?	Errors	Warnings	Time
	Compiling finished (errors: 0; warnings: 0)						
	PLC_V4_4				0	0	9:31:58 AM
	Hardware configuration				0	0	9:31:58 AM
		Hardware was not compiled. The configuration is up-to-date.		?			9:32:04 AM
	Safety	Compile safety program 'Safety Administration'.					9:31:59 AM
	Program blocks				0	0	9:32:02 AM
		No block was compiled. All blocks are up-to-date.					9:32:02 AM
	Consistency check	Consistency check for safety program 'Safety Administration'.			0	0	9:32:02 AM
		The safety program is already consistent and for this reason is n					9:32:02 AM
	Compiling finished (errors: 0; warnings: 0)						

Figura 3.27: TIA Portal V17. Ventana de propiedades, información y diagnóstico

3.5 PN/CANopen Link

Este dispositivo de Siemens es una **pasarela de red PROFINET a redes CAN o CANopen**, y nos servirá para comunicar el *SIMATIC S7-1200* con el *driver* del motor C5-E. Incorpora dos conexiones PROFINET y una conexión CAN o CANopen por cable D-SUB9. Permite la configuración del mismo en 3 modos de funcionamiento diferentes: *CANopen transparent*, *CANopen slave* y *CANopen manager*.

3.5.1 CANopen Transparent

CANopen Transparent es un modo de operación en el que la pasarela *PN/CAN Link* simplemente transmite los datos entre el bus PROFIBUS y el bus CANopen sin realizar ninguna interpretación o procesamiento de los mismos. Esto permite que dispositivos CANopen se comuniquen directamente con dispositivos PROFIBUS sin la necesidad de un controlador o driver adicional. En este modo, la pasarela actúa como un simple puente entre ambos buses y no es necesaria ninguna configuración especial.

3.5.2 CANopen Master

En este modo de configuración, la pasarela actúa como el **controlador principal de la red CANopen**. Esto significa que es responsable de iniciar y finalizar las comunicaciones y de coordinar el intercambio de datos entre los diferentes dispositivos en la red.

El esclavo tiene un número de identificación único en la red, el número de nodo, que le permite al maestro identificarlo y comunicarse con él. El esclavo también tiene objetos de proceso (*Object Dictionary*) que contienen datos o información, y el maestro puede leer o escribir en estos objetos de proceso para controlar o monitorear al esclavo.

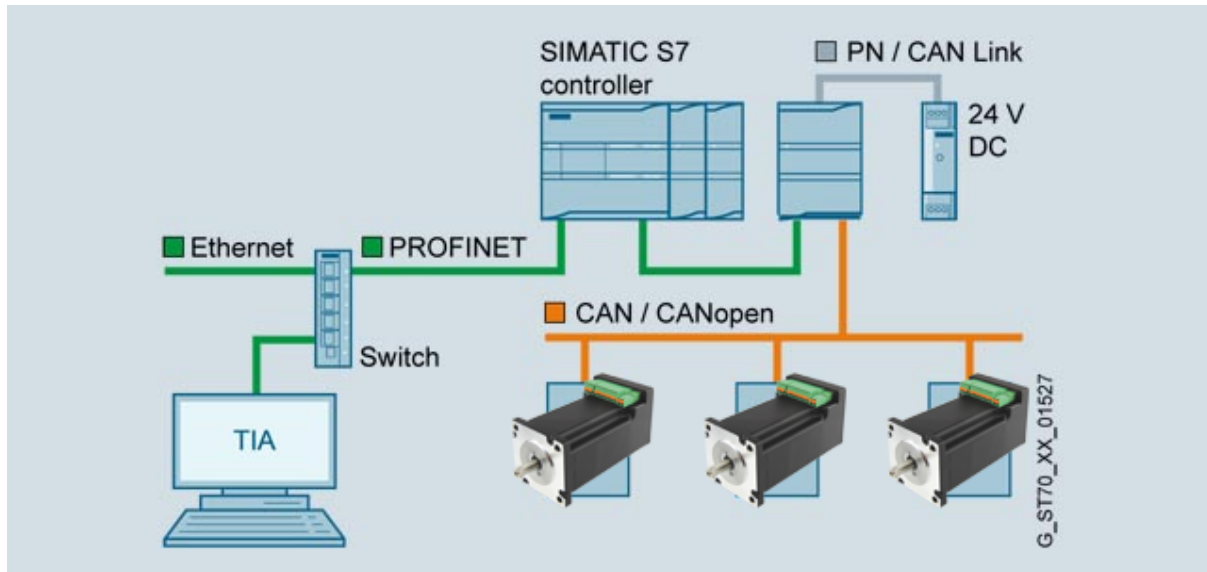


Figura 3.28: Ejemplo de una red CANopen similar a la desarrollada en el proyecto

3.5.3 CANopen Slave

En el modo de funcionamiento de **CANopen Slave**, el dispositivo se configura como esclavo y espera a que el maestro (*Manager*) inicie la comunicación. El esclavo solo puede responder a los mensajes enviados por el maestro, y no puede iniciar la comunicación por su cuenta.

En general, el modo de funcionamiento de *CANopen Slave* se utiliza en dispositivos periféricos que no tienen la capacidad de iniciar la comunicación, como sensores o actuadores, y que dependen del maestro para controlar su funcionamiento.

3.6 Schneider

Schneider Electric es una empresa multinacional francesa especializada en la fabricación de equipos y sistemas de automatización y control de energía. Fue fundada en 1836 y a lo largo de los años ha continuado expandiendo su presencia en el mercado eléctrico mediante la adquisición de empresas como *Telvent*, *MESA* o *Telemecanique*; y desarrollando nuevos productos y tecnologías.

En la actualidad es una de las empresas líderes en el mercado, ofreciendo soluciones para una amplia variedad de sectores industriales, comerciales y residenciales.

3.6.1 Elementos de potencia

Un **contactor** es un dispositivo electromecánico utilizado en sistemas eléctricos para controlar la conexión y desconexión de circuitos eléctricos de alta potencia, y es ampliamente utilizado en la industria para controlar motores, sistemas de iluminación y otros equipos eléctricos de alta potencia.

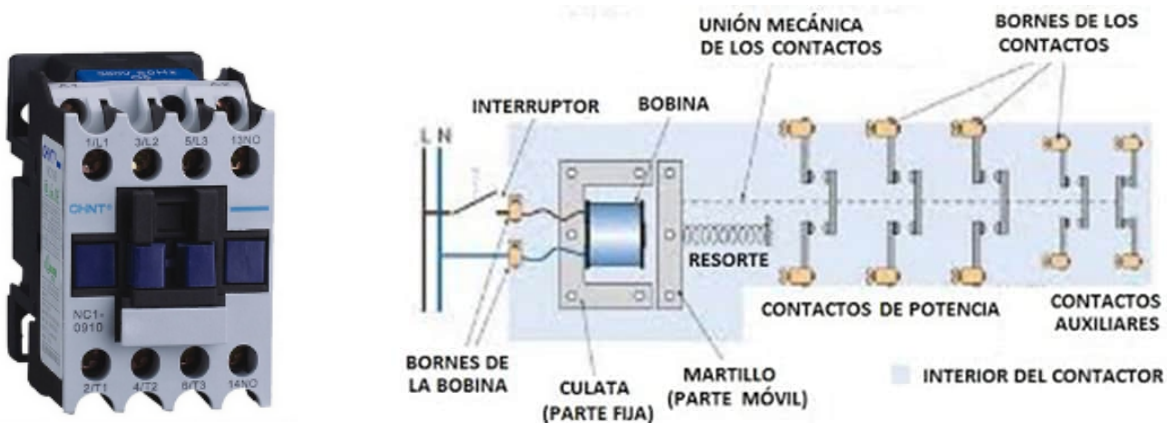


Figura 3.29: Despiece de un contactor

Los contactores funcionan mediante un electroimán que activa un juego de contactos eléctricos, que permiten la conexión o desconexión del circuito eléctrico. Estos dispositivos son capaces de manejar **altas corrientes y tensiones eléctricas**, lo que los hace ideales para su uso en sistemas de potencia industrial.

Están disponibles en una amplia variedad de tamaños y configuraciones para satisfacer diferentes necesidades y aplicaciones. Además, estos dispositivos se pueden integrar con otros componentes eléctricos y electrónicos, como relés de protección y temporizadores, para ofrecer una mayor funcionalidad y protección del sistema eléctrico.

3.6.2 Elementos de control

Los **pulsadores, selectores y setas de emergencia** son dispositivos de control utilizados en sistemas eléctricos y electrónicos para activar y desactivar circuitos eléctricos, seleccionar diferentes opciones y garantizar la seguridad del personal en caso de emergencia. Existen diferentes tipos de clasificación dentro de los elementos de control, pero los dos grandes grupos son: **normalmente abierto** (NA o NO en inglés) o de conexión pues al pulsar se conecta el circuito a la corriente eléctrica; y **normalmente cerrado** (NC) o de desconexión, por la razón contraria.

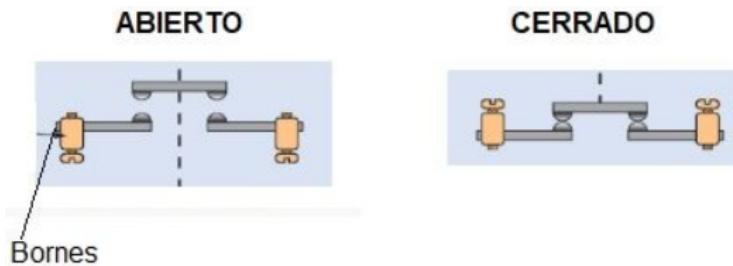


Figura 3.30: Contactos NA y NC

Los **pulsadores** son dispositivos que se activan temporalmente cuando se presiona un botón y se sueltan inmediatamente después de ser liberados. Se utilizan para encender o apagar dispositivos eléctricos o para enviar señales a sistemas de control. Los **selectores**, por otro lado, son dispositivos que se utilizan para seleccionar una opción entre varias posibilidades. Un selector puede tener varias posiciones y cada posición corresponde a una opción diferente.



Figura 3.31: Despiece de pulsadores (izquierda) y selectores (derecha)

La **seta de emergencia**, también conocida como botón de parada de emergencia, es un dispositivo que se utiliza para interrumpir inmediatamente el funcionamiento de un sistema en caso de emergencia. Se caracteriza por ser de color rojo y tener una forma diferente a la de los otros elementos de control para evitar confusiones en su uso.

Cuando se acciona la seta de emergencia, se detienen inmediatamente todos los dispositivos eléctricos que están siendo controlados por el sistema. Es muy común que este elemento de control incorpore dos conexiones diferentes: una NA y otra NC, para asegurarse de su correcto funcionamiento.

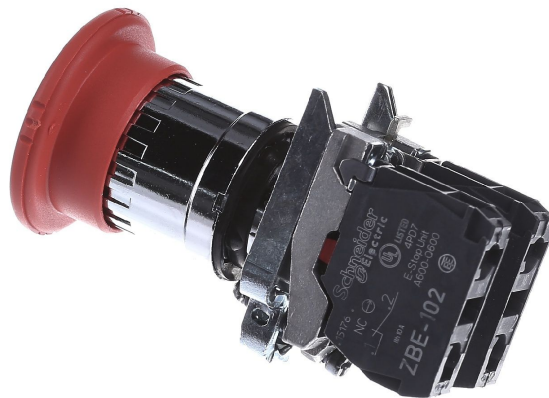


Figura 3.32: Despiece de seta de emergencia con doble conexión



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Capítulo 4. OBJETO DE ESTUDIO

4.1 Introducción

En este capítulo profundizaremos en el proyecto mecatrónico/robótico que se ha desarrollado. Por el lado de la mecánica tenemos una rueda de 140 mm de diámetro, en cuyo eje se incorpora un motor de corriente continua sin escobillas que la hace girar. El motor, a su vez, tiene incorporados un encoder que suministra retroalimentación al sistema de control de lazo cerrado, relacionado con el movimiento de rotación del eje; y unos frenos para el motor.

Por parte de la automática/robótica incorporamos un autómata programable (PLC) *SIMATIC S7-1200* que nos permitirá controlar el movimiento del motor a través de diferentes pulsadores y selectores, gracias al programa que se ha desarrollado y que será detallado más adelante en este documento.

Finalmente, y para establecer conexión entre ambos grupos, contamos con una pasarela *SIMATIC PN/CAN Link* que nos permite pasar del protocolo de comunicación PROFINET (PN) propio del PLC, al protocolo CANopen del *driver* del motor.

También incorporamos diferentes elementos de seguridad tales como dos contactores y una seta de emergencia, además de una fuente de alimentación de 24V para todo el circuito.

4.2 Componentes

4.2.1 Rueda motriz

El accionamiento de este proyecto es una rueda en cuyo eje se integra el motor que se hará rotar, y por consiguiente obligará a la rueda a girar. Algunas de sus características principales están recogidas en la *tabla 4.1*.

Referencia	WD14050-5616-23X
Diámetro de la rueda	140 mm
Ancho de la rueda	50 mm
Capacidad de carga	400 Kg

Torque medio de salida	24,6 Nm
Torque máximo de salida	39,4 Nm
Velocidad máxima de entrada	5968 rpm
Momento de inercia	32 Kg mm ²
Grado de protección IP	IP54
Material de la rueda	Vulkollan
Ratio de reducción	16
Eficiencia	94%
Peso	3,8 Kg
Rango de temperaturas de operación	desde -15 a +90°C

Tabla 4.1: Datos técnicos de la rueda WD14050-5616-23X de Nanotec

En la figura 4.1 se muestra un despiece de la misma:

1. Recubrimiento del segundo satélite.
2. Segundo satélite.
3. Primer satélite.
4. Anillo de rodamientos de bolas, y rueda.
5. Soporte de montura.

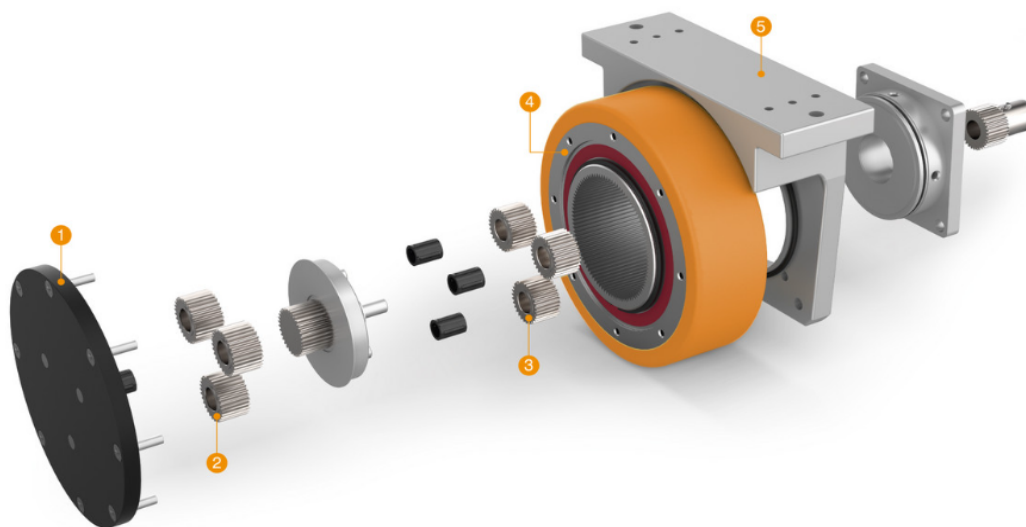


Figura 4.1: Despiece de la rueda WD14050-5616-23X de Nanotec



4.2.2 Motor DC. Frenos y encoder

El elemento del montaje encargado de hacer girar la rueda anterior es el motor *DB59L024035-A* de Nanotec. Se trata de un motor de corriente continua sin escobillas (*Brushless Direct Current - BLDC*) síncrono de corriente trifásica. Este tipo de motores se denominan así porque en muchas ocasiones reemplaza a los motores de corriente continua con escobillas (o motores de conmutador).

Funciona de la siguiente forma: el rotor sigue un campo de rotación magnético y el movimiento es síncrono con el voltaje de corriente alterna aplicado a los devanados.

Algunas de sus características principales son:

Referencia	DB59L024035-A
Tipo de motor	NEMA 23
Tamaño	56 mm
Potencia nominal	172 W
Voltaje nominal	24 V
Torque nominal	47 Ncm
Torque máximo	141 Ncm
Corriente nominal	9.4 A
Corriente máxima	28 A
Velocidad nominal	3500 rpm
Inercia del rotor	119 gcm ²
Resistencia línea a línea	0,22 Ohm
inductancia línea a línea	0,29 mH
Longitud	73,6 mm
Peso	0,72 Kg

Tabla 4.2: Datos técnicos del motor *DB59L024035-A* de Nanotec

En la *figura 4.2* aparecen dos motores: el modelo *DB59L024035-A* y el modelo *DB59L024035R-A*, cuya única diferencia es que este último incorpora una terminación en forma circular, mientras que el primero cuenta con ella en forma cuadrada.



Figura 4.2: Motores DB59L024035-A (derecha) y DB59L024035R-A (izquierda)

Además, el propio fabricante proporciona la opción de combinar sus productos, permitiendo la compra del motor con accesorios como el freno BWA (*Figura 4.3*) o un encoder (*Figura 4.4*).



Figura 4.3: Freno BWA de Nanotec

En el caso del freno, sus características están recogidas en la *tabla 4.3*.

Referencia	BWA 1,5-6,35
Voltaje de operación	24 V
Potencia nominal	11 W
Torque que retiene	150 Ncm
Diámetro de la asta	6,35 mm

Tamaño	51,5 mm
Momento de inercia	10 Kg mm ²
Centro	Taladro ø6,35 H7 con 2 tornillos prisioneros M3x5
Cierre	2 tornillos M4 (BWA 1,5-6,35)
Conexión	L = 300 mm
Tiempo de desconexión	30 ms
Tiempo de encendido	100 ms
Peso	0,3 Kg

Tabla 4.3: Datos técnicos del freno BWA de Nanotec

Por el lado del encoder, este nos permite convertir el movimiento del motor en una señal eléctrica que será leída por el dispositivo de control (lazo cerrado) que incorpora el propio motor. De esta manera, recogemos datos de la posición, velocidad, aceleración o dirección.

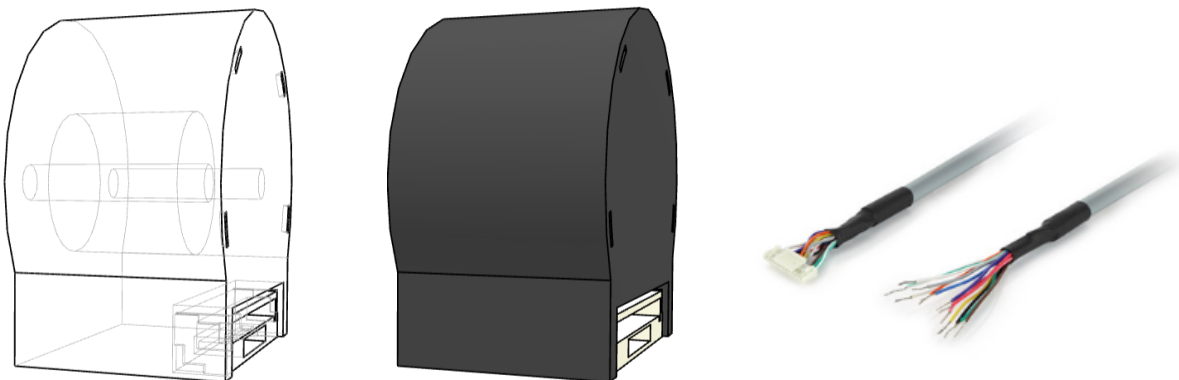


Figura 4.4: Encoder NME1 y conectores ZK-NME1-13-500-S de Nanotec

Para la conexión del encoder con el driver, se emplea el modelo de cable ZK-NME1-13-500-S. Para mayor información sobre la asignación de cables, ver el apartado 4.4 Conexión.



Las características técnicas del encoder son las siguientes:

Referencia	NME1-UVW-T14
Tipo de señal producida	Incremental
Resolución	1024 CPR
Diámetro de la asta	5 mm
Voltaje de operación	5.00 - 24.00 V
Frecuencia límite	500 KHz
Consumo de corriente	30 mA
Nivel de la señal	baja: <2.0 V (@I_load=20 mA) alta: VCC-0.2 V (@I_load=20 mA)
Velocidad límite	30000 RPM
Temperatura de operación	-20°C - 80°C
Corriente máxima de salida por canal	70 mA (@Vcc=5 V, Vout=3 V), 90 mA (@Vcc=24 V, Vout=18 V)
Humedad	max. 90 % (no condensation)
Temperatura de almacenamiento	-40 °C - 85 °C

Tabla 4.4: Datos técnicos del encoder NME1 de Nanotec

4.2.3 Driver

Se trata del elemento del montaje encargado de controlar el motor de corriente continua anterior, y en consecuencia, de hacer girar o no la rueda. Los driver en general son amplificadores de corriente cuya función es tomar una pequeña señal de control de baja corriente y convertirla en una señal de alta corriente que pueda alimentar el motor. La conexión con este último está detallada en el apartado 4.4 Conexionado de este documento.

En este caso, este driver está pensado para la comunicación a través de CANopen con un elemento de mando como puede ser un PLC (empleado en este proyecto), aunque existen modelos muy similares del mismo fabricante con protocolos de comunicación diferentes.

El driver además incorpora una conexión micro USB que permite conectarse al PC a través del *software* especializado de Nanotec comentado en el apartado 3.3.1 Plug and Drive Studio.



Figura 4.5: Driver C5-E de Nanotec

Las características técnicas del *driver* son las siguientes:

Referencia	C5-E-2-09
Protocolo de comunicaciones	CANopen
Interfaz	USB
Voltaje de operación	12 VDC - 48 VDC
Corriente nominal (RMS)	10 A
Corriente de pico (RMS)	30 A
Motores compatibles	Motores BLDC, motores paso a paso
Número de entradas digitales	5
Tipo de las entradas digitales	5/24 V conmutable
Número de entradas analógicas	2
Tipo de entradas analógicas	0 - 10 V, 0 - 20 mA / 0 - 10 V conmutable

Número de salidas digitales	3
Tipo de salidas digitales	open-drain (max. 24 V/100 mA)
Entrada para encoder	Sí
Tipo de señal del encoder	Incremental
Salida para frenos	Sí
Peso	0,27 Kg
Rango de temperatura	-10°C - 40°C

Tabla 4.5: Datos técnicos del driver C5-E-2-09 de Nanotec

4.2.4 Autómata programable (PLC)

Como se ha citado previamente, la definición de un **Controlador Lógico Programable** es un dispositivo electrónico utilizado para controlar procesos automatizados en una gran variedad de industrias. En el caso concreto de este proyecto se ha hecho uso de un *SIMATIC S7-1200*.

Los modelos *S7-1200* están muy extendidos actualmente. Las principales diferencias cuando cambiamos de la gama anterior (*S7-400*) a la gama *S7-1200* son una mayor capacidad de procesamiento, mayor velocidad de comunicación, mayor cantidad de entradas y salidas, mayor capacidad de memoria, y una interfaz de programación más intuitiva.



Figura 4.6: PLC SIMATIC S7-1200 con CPU 1215FC DC/DC/DC de Siemens

Además dentro de cada modelo se encuentran diferentes tipos de CPU, que cambian en el tipo de las señales y la alimentación. La CPU aquí usada es la *1215FC DC/DC/DC*, de referencia *6ES7215-1AF40-0XB0*, que es la de la *figura 4.6*. La versión de *firmware* del controlador es la V4.4, por lo que nuevas funcionalidades incorporadas a partir de la versión V4.5 no van a estar disponibles en el mismo.

Esta es alimentada por tensión continua entre 20,4-28,8V DC, al igual que las entradas (14) y las salidas digitales (10). Consta también de 2 entradas analógicas de 0-10V DC y de 2 salidas analógicas de 0-20mA DC. Además, tiene 2 puertos PROFINET.

No obstante, lo más destacable de esta CPU en concreto es la incorporación de funciones de seguridad para garantizar SIL3 que, como se ha mencionado en el *apartado 2.3.3.1 Funciones de seguridad PROFIsafe*, pueden impedir tanto que no se produzcan fallos en los elementos conectados, como minimizar los riesgos en caso de que estos ocurran.

4.2.5 Módulos adicionales de entradas y salidas

Para poder disponer de un mayor número de entradas y salidas analógicas se ha decidido ampliar la CPU con dos módulos adicionales:

- Se emplea el *SM 1226 F-DI*, con referencia *6ES7226-6BA32-0XB0* (*figura 4.7*), que dispone de 16 entradas digitales también a 24V DC.
- Se emplea el *SM 1226 F-DQ*, con referencia *6ES7226-6DA32-0XB0* (*figura 4.7*), que dispone de 4 salidas digitales también a 24V DC.



Figura 4.7: Módulos adicionales de entrada y salida de Siemens

Tener en cuenta que, estadísticamente, hay muchos más fallos en los sensores y actuadores que en el propio PLC. Es por ello que las entradas y salidas de estos módulos adicionales son también PROFIsafe e incorporan funciones de seguridad que garantizan nivel de protección SIL3.

4.2.6 Pasarela (PN/CAN Link)

Este dispositivo permite la transición de red PROFINET a redes CAN o CANopen, y nos servirá para comunicar el *SIMATIC S7-1200* con el driver del motor *C5-E*.

La alimentación de este dispositivo de referencia *6BK1620-0AA00-0AA0*, al igual que en los elementos anteriores, es a 24V DC. Incorpora dos conexiones PROFINET y una conexión CAN o CANopen por cable D-SUB9.

Su versión de *firmware* es v2.1 y permite la configuración del mismo en los 3 modos de funcionamiento comentados: *CANopen transparent*, *CANopen slave* y *CANopen manager*. En este caso, su configuración será en esta última (Ver apartado 4.5.2 Pasarela (PN/CAN Link). Protocolo CAN).



Figura 4.8: Pasarela PN/CAN Link de Siemens

4.2.7 Fuente de alimentación

Elemento encargado de dar el voltaje apropiado a toda la instalación. Esta unidad en concreto de Schneider y de referencia *ABL7 RE2405*.

Permite una tensión de alimentación entre 100 y 240V AC para conseguir una salida de 24V DC y 5A. Además, incorpora un potenciómetro para ajustar exactamente el voltaje de salida, en un rango de 28,8V a 24V.



Figura 4.9: Fuente de alimentación a 24V de Schneider

4.2.8 Contactores

Como hemos comentado en el [apartado 3.6 Schneider](#), un **contactor** es un dispositivo electromecánico utilizado en sistemas eléctricos para controlar la conexión y desconexión de circuitos eléctricos de alta potencia. El modelo del proyecto es el *LC1D18BL*, que permite una alimentación bien de 440V AC, o bien de 24V DC (en sus correspondientes pines) con un consumo de 2,4W. Su esquema eléctrico será detallado en el [apartado 4.4.4 Esquema eléctrico completo](#).

En nuestro caso lo empleamos para la desconexión del motor tras el accionamiento de la seta de emergencia, debido a su alto poder de corte. Esto garantiza que no continúe recibiendo corriente tras un fallo.

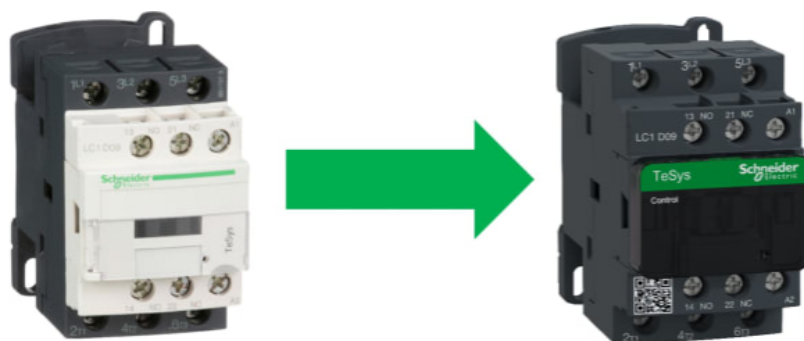


Figura 4.10: Contactor de Schneider

Pese a que los contactores permiten trabajar con altas tensiones e intensidades eléctricas, en este proyecto solo estamos alimentando con 24V DC, luego su uso no es imprescindible aunque dota al mismo de un perfil más industrial.

En la *figura 4.10* observamos el contactor empleado en el proyecto (izquierda), actualmente en desuso, y su relevo en el catálogo de productos (derecha).

4.2.9 Pulsadores. Selectores. Setas de emergencia

Son los **elementos de control** empleados en el proyecto con el objetivo de controlar el PLC y como consecuencia, controlar el movimiento del motor. Son del modelo *Harmony XB4* de Schneider, todos con anillo de fijación y diámetro de 22 mm.

El pulsador posee un contacto NA y otro NC, además de un LED incorporado alimentado a 24V DC. En nuestro programa emplearemos 4 pulsadores y el contacto NA para todos ellos. Por otro lado, el segundo elemento es un selector de dos posiciones NA.

Finalmente la seta de seguridad tiene contactos NA y NC, con reset por desenclavamiento y tipo de cabeza estándar.



Figura 4.11: Elementos de control de Schneider

4.2.10 Caja estanca

Envoltorio sobre las cuales se montan los elementos de control anteriores. Son del modelo *Harmony XAP* de Schneider, de dimensiones 80 x 80 x 77 mm y con un orificio de 22 mm. Su peso es de 0,605 Kg.



Figura 4.12: Caja estanca de Schneider

4.2.11 Cables RJ45 y D-SUB9

El **conector RJ45** tiene ocho pines en su interior que se utilizan para transmitir y recibir datos a través de la red. Utiliza un cable de par trenzado y se utiliza comúnmente para conectar dispositivos como computadoras, *routers*, *switches*, puntos de acceso inalámbrico y otros dispositivos de red a una red *Ethernet*.

Por un lado, el conector RJ45 nos permite conectar mediante PROFINET el PLC al PC, al igual que el PLC a la pasarela *PN/CAN Link*.

La pasarela también incluye un cable **D-SUB9** hembra en uno de sus extremos: la fila superior tiene cinco pines y la fila inferior tiene cuatro pines, lo que proporciona un total de nueve pines. Los pines se utilizan para transmitir y recibir señales de datos, como pueden ser los valores *CAN_H* y *CAN_L*, así como para la conexión a tierra y la alimentación.

Para mayor información sobre el conexionado, ver el [apartado 4.4 Conexionado](#).



Figura 4.13: Cables RJ45 y D-SUB9

4.2.12 Carril DIN

Un carril DIN o riel DIN es una barra de metal normalizada. Es muy usado para el montaje de elementos eléctricos de protección y mando, tanto en aplicaciones industriales como en viviendas [20]. En nuestro caso nos servirá como soporte de los diferentes componentes ya mencionados.



Figura 4.14: Ejemplo de carril DIN [20]

4.2.13 Borneros para carril DIN

Los borneros o bornas son componentes cuya función principal es garantizar la seguridad y asegurar las conexiones del conductor eléctrico y mecánico. Emplearemos tanto bornes simples como bornes terminales, ambos anclados al carril DIN anterior.



Figura 4.15: Bornero

4.3 Proceso de montaje

Para el **proceso de montaje** de los componentes mencionados en el apartado anterior, en primer lugar se ha adquirido una tabla de 60 x 86,5cm y de 1,5cm de espesor, en la cual se han emplazado los mismos. Los pasos seguidos para el montaje han sido los siguientes:

1. Anclaje del carril DIN. Se colocarán 3 carriles en total, de diferente longitud. Según la *figura 4.16*:
 - En el carril superior se sitúa la fuente de alimentación y bornes de tierra, bornes de alimentación positiva del circuito (24V) y bornes de alimentación negativa del circuito (0V).
 - En el carril del medio se sitúan los contactores y los elementos de Siemens (pasarela, PLC y módulos adicionales de E/S).
 - En el carril inferior se sitúa el driver y bornes para los cables salientes del encoder.
2. Montaje del motor de corriente continua en la rueda, y atornillado de esta a la maqueta. Colocación de los bornes y del driver en el carril DIN inferior. En este punto y a falta del conexionado, la parte de los componentes de Nanotec está completa.
3. Atornillado de las cajas estancas a la maqueta. Montaje de los pulsadores y selector.
4. Unión del PLC a los módulos adicionales de E/S mediante las patillas habilitadas para ello. Montaje del conjunto del PLC, la pasarela y los contactores en el carril DIN del medio.
5. Montaje de los bornes y la fuente de alimentación en el carril DIN superior. Punteado de los mismos de tal manera que todos los bornes de alimentación positiva están conectados entre sí, y lo mismo con los de alimentación negativa.
→ Con esto conseguimos que solo se necesite un único cable saliente de la fuente de alimentación para alimentar todo el circuito con los 24V necesarios (ver apartado 4.4 Conexionado).

En este momento todos los componentes están debidamente colocados y solo sería necesario su correcto conexionado. Así pues, una vez tenemos el montaje hecho podemos dividirlo en 4 zonas según el emplazamiento y la función de los componentes:

- **Zona I.** Fuente de alimentación.
- **Zona II.** Botonera (pulsadores, selector y contactores).
- **Zona III.** Elementos de Siemens. PLC.
- **Zona IV.** Elementos de Nanotec. *Driver*, motor y rueda.

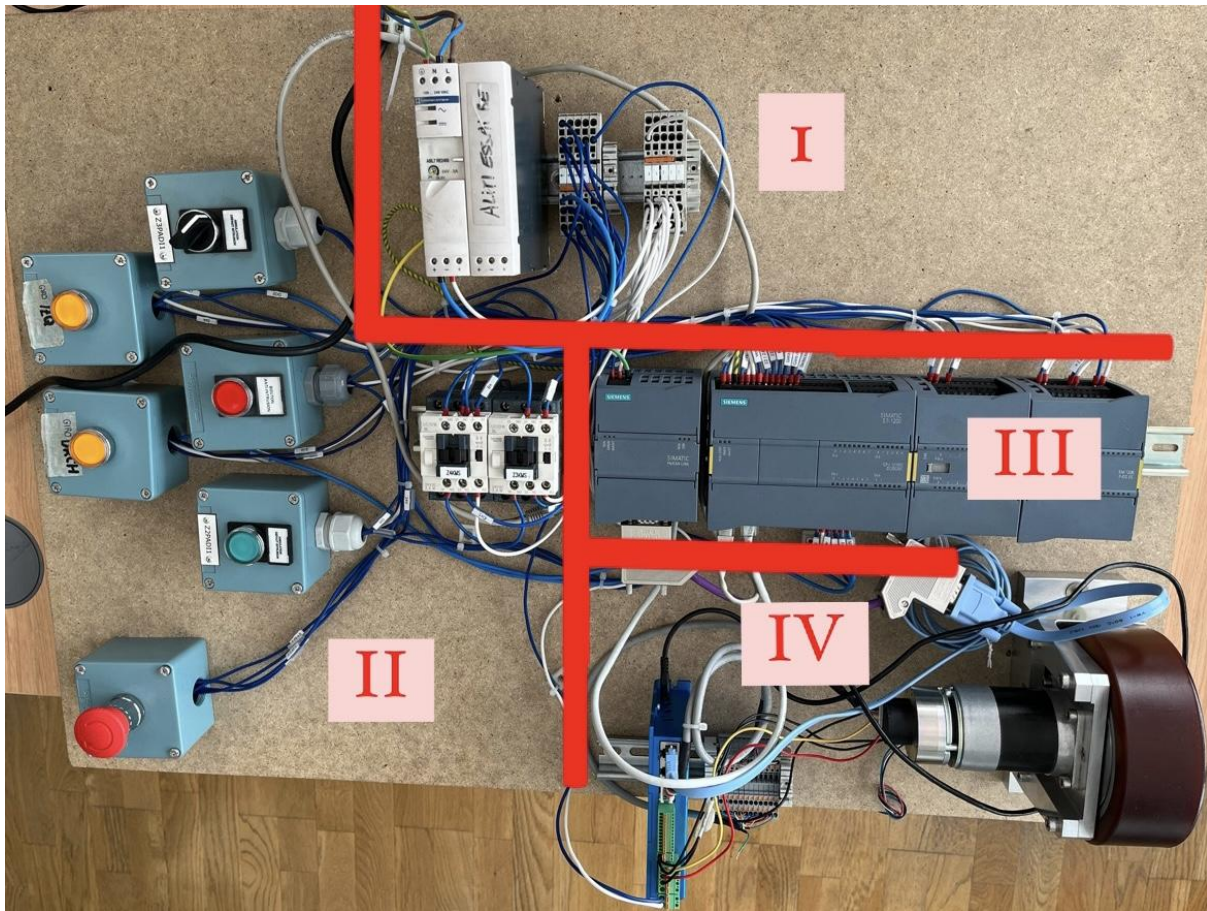


Figura 4.16: Montaje del proyecto, dividido por zonas

4.4 Conexionado

El **conexionado** empieza con el suministro de corriente alterna de 230V a la fuente de alimentación usada. Esta, a su vez, permite conseguir los 24V necesarios para toda la circuitería eléctrica y electrónica del montaje.

Debemos de alimentar a 24V:

- PLC S7-1200.
- Pasarela PN/CAN Link.
- Módulos adicionales de entradas y salidas.
- Driver del motor.
- Pulsadores, selector, conmutadores y seta de emergencia.



4.4.1 Componentes de Siemens

Además cabe destacar que en el caso del PLC, tendremos tres conexiones a 24V, en lugar de simplemente una. Esto es debido a la siguiente **asignación de pines**:

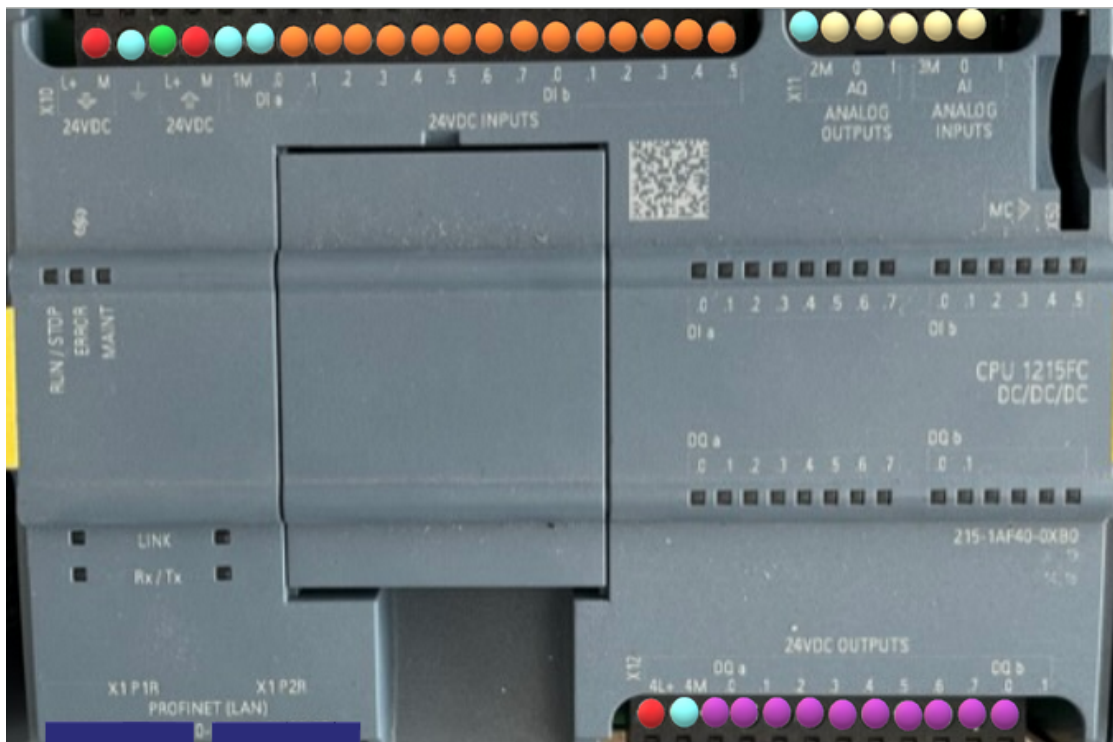


Figura 4.17: Asignación de pines del PLC

El color **rojo**, representado por L+, indica el bornero de la alimentación positiva de 24V. En este caso, por tratarse de una *CPU DC/DC/DC*, la primera designación de DC indica que la alimentación debe ser de corriente continua (las dos siguientes designaciones DC se corresponden con un tipo de entradas y salidas de corriente continua).

Este PLC cuenta con una fuente interna, por lo que una vez alimentemos por la primera borna de L+ podremos realizar el resto de alimentaciones de 24V a partir del mismo PLC.

También debemos alimentar a 24V la borna 4L+ ya que queremos usar las salidas digitales del autómatas.

Por otro lado, el color **azul claro**, que está representado por la letra xM (siendo x cualquier valor entero positivo), se corresponde con el borne de conexión negativa o 0V. Al igual que en el caso anterior, debemos conectar este borne siempre que queramos utilizar el conjunto de bornes a los que agrupa. Por ejemplo:

- El primer borne designado por la letra M es el de entrada de alimentación 0V del PLC.
- El segundo borne con la letra M es la salida de 0V, al igual que nos sucedía con la salida de 24V del PLC.
- El borne de 1M deberá de ser alimentado si queremos usar las entradas digitales. Lo mismo con los bornes 2M y 3M para las entradas y salidas analógicas (no usadas), y con el borne 4M para las salidas digitales.

El color **verde** representa la tierra o masa, y su designación es un símbolo de tierra. Solo contamos con un borne de tierra en el conjunto del PLC.

El color **naranja** y **morado** son las entradas y las salidas digitales respectivamente, y el color **blanco roto** las entradas y salidas analógicas. Como veremos en el apartado 4.4.4 Esquema eléctrico completo, solo hemos usado 7 de las 14 entradas digitales disponibles, y 4 de las 10 salidas del mismo tipo.

Y finalmente el color **azul oscuro** indica los puertos RJ45 de los que dispone el PLC, de los cuales emplearemos uno para conexión directa con el PC y otro para conectar el conjunto del PLC con la pasarela (y por consiguiente con el *driver*).

La misma leyenda sirve para el módulo de entradas y de salidas adicionales, incorporando el color **negro** que indica que esa borna no tiene conexión interna. Usaremos 2 entradas y 2 salidas de estos elementos.

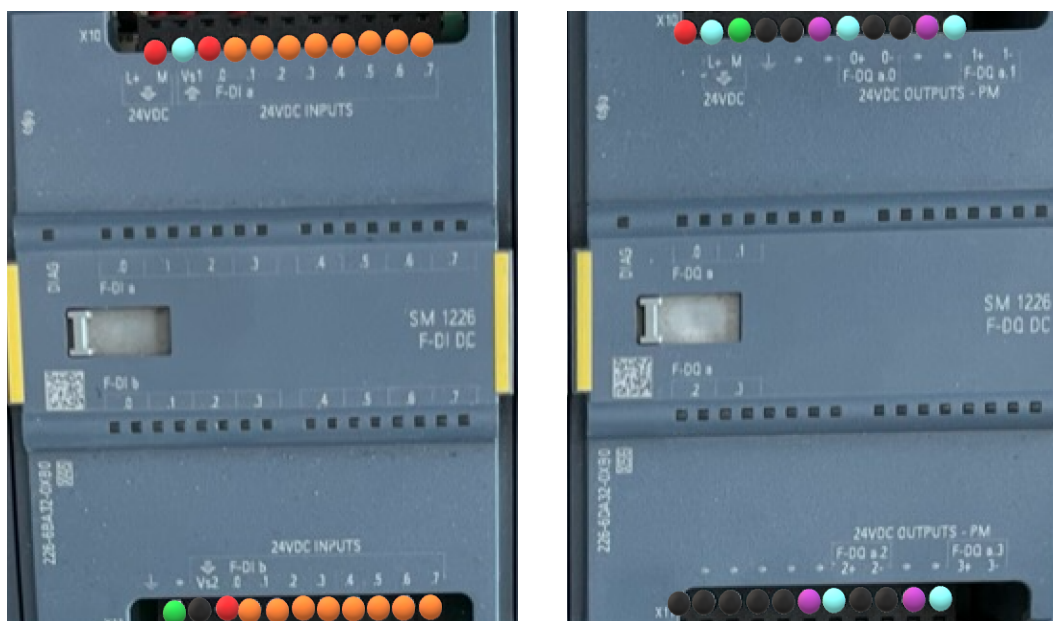


Figura 4.18: Asignación de pines del módulo de entrada (izquierda) y salida (derecha)

#Nota: la posición de los bornes en comparación con la numeración escrita de sus nombres está ligeramente desplazada hacia la izquierda.

En este punto se puede hacer una **apreciación de interés**, que es el por qué del uso de módulos adicionales si no estamos usando el 100% de las entradas y salidas del propio PLC, y por tanto no serían necesarias más. La explicación es puramente educativa y no tiene otro fin que aprender tanto a montar los correspondientes módulos en el PLC y en el carril DIN, como dotar al programa en TIA Portal de un aspecto más completo y similar al de una situación real industrial.

Por parte de la pasarela, el color **marrón clarito** representa la conexión mediante el cable D-SUB de 9 pines que permite establecer el protocolo CANopen entre el driver y la pasarela PN/CAN Link. En este caso solo emplearemos una de las dos conexiones RJ45 con el fin de conectarnos al PLC mediante *ProfiNet*.

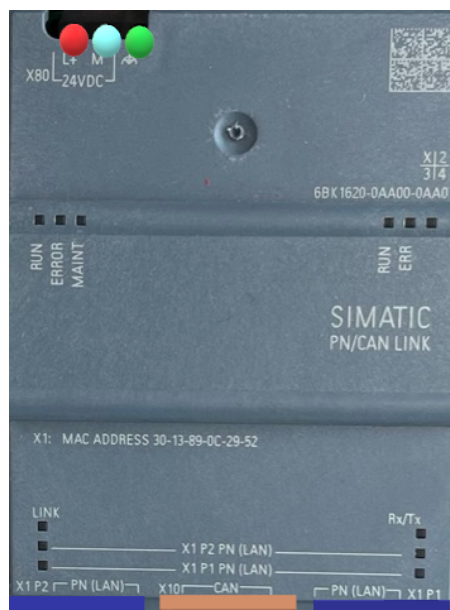


Figura 4.19: Asignación de pines de la pasarela

4.4.2 Componentes de Nanotec

Los elementos electrónicos de Nanotec incluidos en el montaje se encuentran conectados entre sí, y la única conexión con el resto del montaje que tienen, a excepción de la alimentación a 24V necesaria para el *driver*, es a través del cable RJ45 - DSUB9 que los permite comunicarse mediante el protocolo CANopen. No obstante, el driver también incorpora un cable USB - microUSB para la conexión directa con el ordenador en caso de no disponer de un PLC para su control.

Como ya hemos comentado en el *apartado 4.3 Montaje*, el encoder y los frenos se encuentran atornillados al motor y este, a su vez, integrado en la rueda que se pretende mover. Por parte del encoder empleamos el antes mencionado conector ZK-NME1-13-500-S, cuya asignación de pines (13, de ahí su nombre) es la de la *figura 4.20*, que vemos que evidentemente se corresponde con la asignación de pines del encoder.

Pin No.	Colour	NME1-UVW-xxx
1	GN/WH (Shielding)	GND
2	GN	A
3	BN	A\
4	GY	B\
5	WH	B
6	YE	I\
7	OG	I
8	BU	H1
9	PK	H3
10	VT	H2
11	BK	GND
12	RD	VCC (+5V to +24V)
13	GN/WH (Shielding)	GND

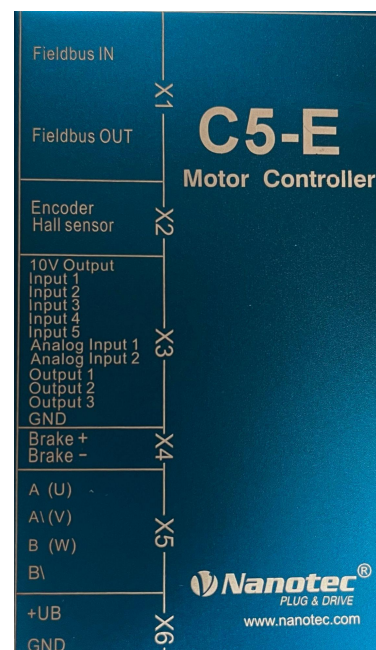
PIN assignment JST GH - 13pole		Connecting cable ZK-NME1-13-500-S
PIN No.	Function	Colour
1	GND	GN/WH (shielding)
2	A	GN
3	A\	BN
4	B\	GY
5	B	WH
6	I\	YE
7	I	OG
8	H1	BU
9	H3	PK
10	H2	VT
11	GND	BK
12	VCC (+5V to +24V)	RD
13	GND	GN/WH (shielding)

Figura 4.20: Asignación de pines del conector (izquierda) y del encoder (derecha)

Estos 13 pines del mismo cable deben ir conectados al pin designado como X2 del driver del motor, que recoge las conexiones del encoder y del *sensor Hall* (incluido en el anterior).

La alimentación de 24V del driver la realizaremos por los pines +UB y GND del mismo, y la conexión CANopen mediante el pin de *Fieldbus IN*. Esto último es debido a que la configuración que adoptaremos para nuestro proyecto es tal que el driver del motor es el esclavo, a las órdenes del PLC. La comunicación es bidireccional luego a través del mismo puerto será también capaz de responder las peticiones del maestro.

Figura 4.21: Asignación de pines del driver



Los cables del freno, también atornillado al propio motor, deben ir conectados al pin X4 del driver, a las conexiones de *Brake+* y *Brake-*.

Finalmente las conexiones del propio motor: los tres cables del círculo rojo (cables amarillo, rojo y negro de la *figura 4.22*) son los que deben ir conectados al pin X5, concretamente a las conexiones *U*, *V* y *W* respectivamente del *driver*. Por otro lado, los cables del círculo azul pueden no estar conectados pues estos pines ya están incluidos en las conexiones del *driver*.

WIRING DIAGRAM			
	COLOR	FUNCTION	LEAD GAUGE
MOTOR 6 POL.	YE	U	UL1332 AWG18
	RD	V	
	BK	W	
HALL 18 IMPL. REV.	RD	+5V	UL1430 AWG26
	BU	H1	
	GN	H2	
	WH	H3	
	BK	GND	

Figura 4.22: Asignación de pines del motor

4.4.3 Botonería

Los **pulsadores**, **selectores** y **seta de emergencia** nos permiten enviar señales a nuestro PLC (entradas), a partir de las cuales este realiza una serie de operaciones en base a la programación realizada. El símbolo de los pulsadores y del selector NA empleados en el proyecto son los de la *figura 4.23*.

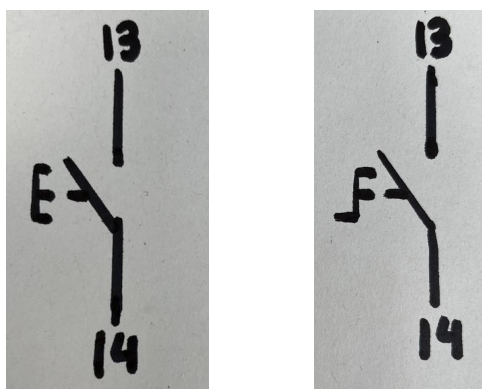


Figura 4.23: Símbolo eléctrico de pulsador NA (izquierda) y selector NA (derecha)

Los pulsadores empleados cuentan además con una luz LED que puede ser activada con las salidas del PLC.

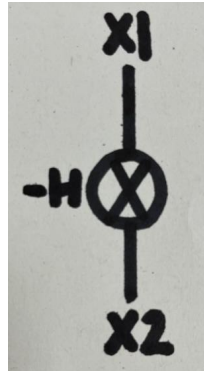


Figura 4.24: Símbolo eléctrico de una bombilla

También usamos una seta de emergencia, cuya activación provoca indirectamente que se activen los contactores y el motor se detenga. Por seguridad, algunos tipos de setas de emergencia cuentan con dos contactos, uno NA y otro NC, de tal manera que de cara a la programación tenemos una doble verificación del circuito de corriente para el funcionamiento del proyecto.

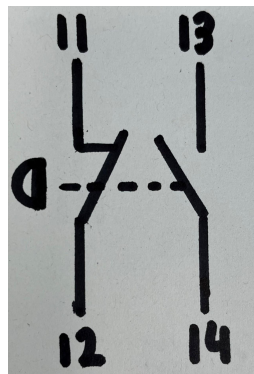


Figura 4.25: Símbolo eléctrico de seta de emergencia

Por parte de los contactores, cuentan con una bobina en su interior que al ser activada por un voltaje, hace conmutar sus conexiones internas. Es decir, el contacto NA (13-14 en la figura 4.25) pasaría a estar cerrado permitiendo el paso de corriente, mientras que el contacto NC (21-22) se abriría. La señal de activación de la bobina se denomina A1 y está conectada a una de las salidas del módulo adicional de salidas digitales del PLC. La salida A2 de la bobina cierra el circuito eléctrico (0V).

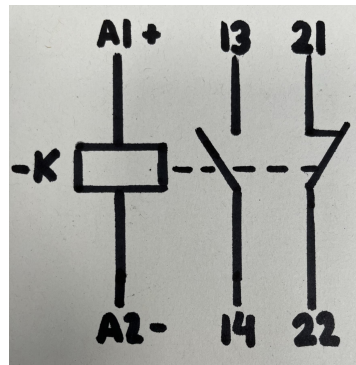


Figura 4.26: Símbolo eléctrico del contactor activo por relé

Debido a que la seta de emergencia empleada cuenta con dos conexiones (NA y NC), también usaremos en este caso dos contactores cuya bobina o relé estarán activos por dos salidas diferentes del módulo adicional, pero conectados en serie de tal manera que será necesaria la activación de ambas salidas para permitir el paso de corriente. En caso de producirse esto, se activan las dos entradas correspondientes en el PLC que son las que gobernarán el resto de la lógica interna.

Una vez más podemos hacer una **apreciación importante**: no es necesario enrevesar tanto el circuito eléctrico pues con una seta de emergencia de una única conexión (suelen ser un contacto NC) y un solo contactor tendríamos la misma función. Pero el circuito ha sido realizado de esta forma pues estamos trabajando con módulos de seguridad de Siemens, lo cual indica que este es un factor crítico para el usuario, y se ha considerado de importancia contar con **dobles verificaciones** para prevenir riesgos.

No obstante también podríamos conectar directamente la conexión de la seta de emergencia a una de las entradas del PLC y omitir el uso de contactores, pero como se ha comentado, su empleo es debido a que estos cuentan con una mayor **fuerza de corte** de corriente que es importante a la hora de trabajar con motores de más voltaje.

4.4.4 Esquema eléctrico completo

Acorde a las explicaciones hechas y con los símbolos eléctricos de las anteriores figuras, realizamos el **esquema eléctrico** del proyecto con todos los elementos incluidos en el mismo. Destacar que se han incluido fotografías para que su comprensión sea más fácil. *Ver apartado Anexo III. Esquema eléctrico.*

- Las conexiones a 24V están representadas en **rojo**.

- Las conexiones a 0V están representadas en **negro**.
- Las conexiones a tierra están representadas en **verde**.
- Las mismas en color **azul** indican conexiones RJ45 y D-SUB9.

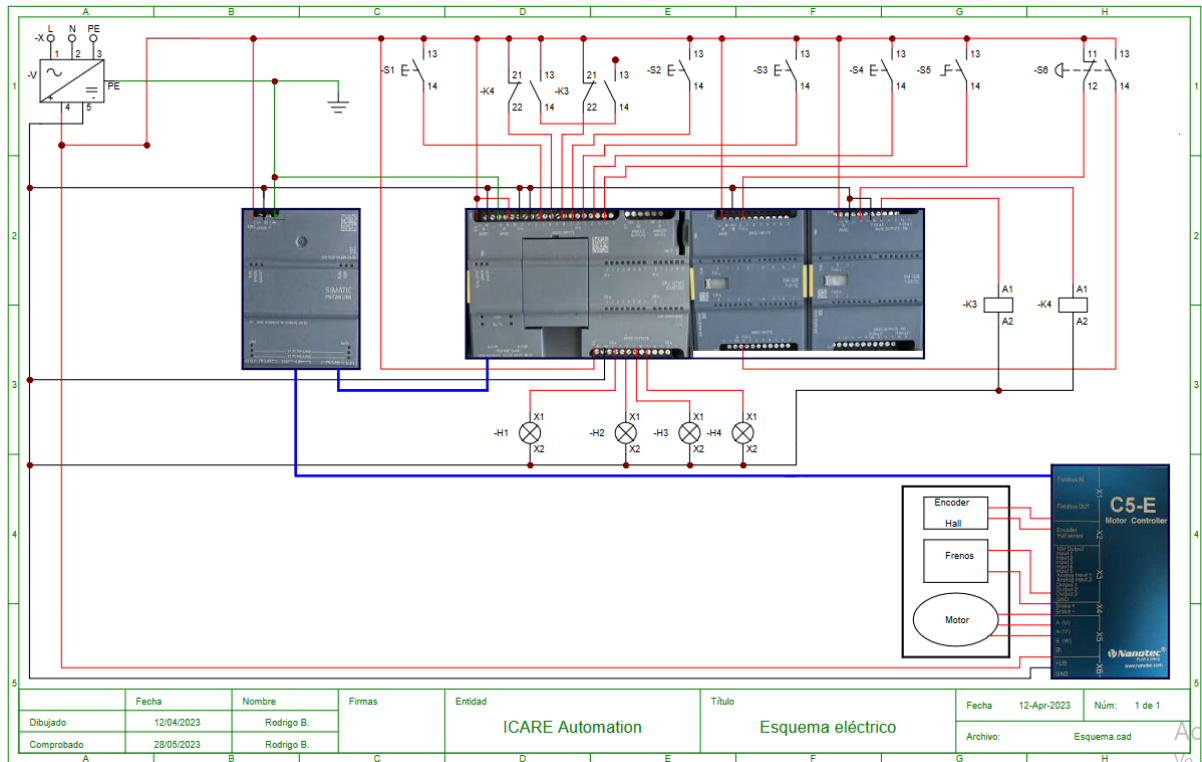


Figura 4.27: Esquema eléctrico del proyecto

4.5 Desarrollo en TIA Portal v17

4.5.1 Autómata programable (PLC)

Como ya se ha comentado, el PLC utilizado en el proyecto es un S7-1200 de Siemens con CPU 1215 FC y conexión Profinet. A mayores, se le han añadido un módulo adicional de 16 entradas digitales y un módulo adicional de 4 salidas, también digitales.

Existen tres lenguajes de programación de cara a la configuración de los autómatas en TIA Portal:

- **FUP: Diagrama de funciones.** Se basa en los símbolos lógicos de funciones usados en el álgebra booleana.
- **SCL: Lenguaje estructurado.** Lenguaje de programación basado en texto de alto nivel.
- **KOP: Esquema de contactos.** Este lenguaje se basa en el diagrama de escalera o 'ladder' que incluye segmentos compuestos por contactos y bobinas. En función

del valor de los contactos (abierto o cerrado) se activan o no las bobinas. De esta manera se puede implementar la lógica de control de un proceso. Es el método más habitual y el empleado en este proyecto.

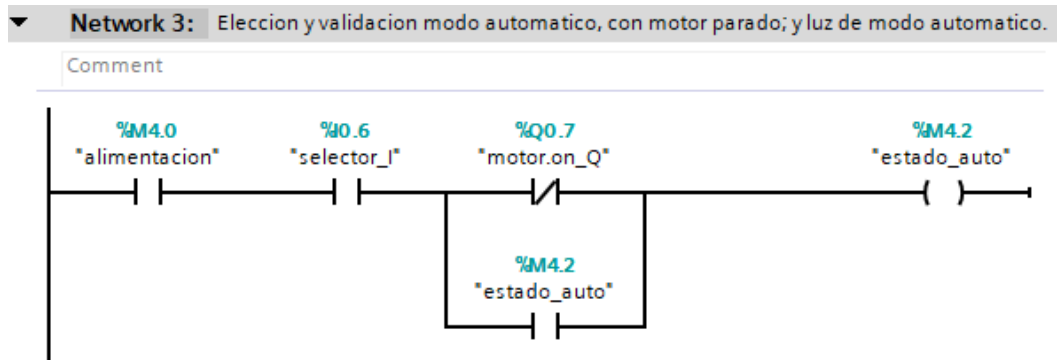


Figura 4.28: TIA Portal. Ejemplo de programación en ladder

Además de los contactos y bobinas mencionados, TIA Portal incluye otro tipo de elementos adicionales que sirven para generar los valores de las variables necesarios, como pueden ser: contadores, operaciones aritméticas, comparadores, operaciones lógicas, conversión del tipo de datos... Todos ellos se representan con el símbolo de la figura 4.29 hasta que se les da una función específica.



Figura 4.29: TIA Portal. Símbolo de contacto general

En TIA Portal existen diferentes tipos de variables. Nosotros usaremos las siguientes:

- **Variables de entrada (%I).**

Son las entradas físicas del controlador. Se emplean para recoger información del exterior. De cara a la programación, son contactos.

- **Marcas (%M).**

Son tipos de datos que sirven para almacenar información (datos y valores) en el controlador. No están relacionadas directamente con las entradas y salidas físicas. De cara a la programación, pueden ser bien contactos o bien bobinas. Es importante destacar que cada salida solo se puede incluir una única vez por cada ciclo de ejecución del programa, para evitar errores.



• **Salidas (%Q).**

Son las salidas físicas del controlador. Se emplean para generar la información de salida al exterior, a partir de los valores de las variables de entrada y de las marcas. De cara a la programación, son bobinas.

Nombre	Tipo	Dirección	Descripción	E/S del:
verde_I	Entrada	%I0.0	Interruptor verde	PLC
22.Z4_I	Entrada	%I0.1	Salida del contactor Z4KMS	PLC
22.Z3_I	Entrada	%I0.2	Salida del contactor Z3KMS	PLC
rojo_I	Entrada	%I0.3	Interruptor rojo	PLC
giro.izq_I	Entrada	%I0.4	Interruptor de giro antihorario (Giro izq)	PLC
giro.drch_I	Entrada	%I0.5	Interruptor de giro horario (Giro drch)	PLC
selector_I	Entrada	%I0.6	Selector de modo manual o automático	PLC
seta_a	Entrada	%I8.0	Seta de emergencia (NC)	Módulo adicional de entradas
seta_b	Entrada	%I9.0	Seta de emergencia (NA)	Módulo adicional de entradas
verde_Q	Salida	%Q0.0	Luz del pulsador verde	PLC
rojo_Q	Salida	%Q0.1	Luz del pulsador rojo	PLC

Nombre	Tipo	Dirección	Descripción	E/S del:
nar.izq_Q	Salida	%Q0.2	Luz del pulsador naranja (sentido antihorario)	PLC
nar.drch_Q	Salida	%Q0.3	Luz del pulsador naranja (sentido horario)	PLC
motor.on_M_Q	Salida	%Q0.6	Movimiento del motor en modo manual	PLC
motor.on_A_Q	Salida	%Q0.7	Movimiento del motor en modo automático	PLC
motor.drch_Q	Salida	%Q1.0	Movimiento horario	PLC
motor.izq_Q	Salida	%Q1.1	Movimiento antihorario	PLC
+_Z4	Salida	%Q17.0	Activación del contactor Z4KMS	Módulo adicional de salidas
+_Z3	Salida	%Q17.1	Activación del contactor Z3KMS	Módulo adicional de salidas

Tabla 4.6: Entradas y salidas del PLC y de los módulos adicionales

Nótese que las entradas y salidas de seguridad aparecen con un color amarillo de fondo. Todos estos elementos se agrupan en diferentes bloques según las necesidades de la programación. Existen varios tipos de bloques:

• **Bloques de organización (OB).**

Tipo de bloque que se ejecuta de manera cíclica o en respuesta a un evento específico. Los bloques de organización se emplean para ejecutar los programas de usuario.

• Bloques de función (FB).

Tipo de bloque que agrupa una o varias funciones específicas a los que el bloque de organización llama en función de ciertas condiciones que se den en el programa. Tienen asociado un bloque de datos (DB).

• Funciones (FC).

Tipo de bloque cuya única diferencia con un FB es que no tienen asociado un bloque de datos, es decir, el valor resultante de una operación solo estará disponible durante ese ciclo de ejecución. Los FB y las FC se suelen utilizar para simplificar el programa y hacerlo más fácil de mantener y depurar.

• Bloques de Datos (DB).

Almacenan datos del programa. Existen dos tipos de DBs:

- **DB de instancia:** Almacena los datos de un Bloque de función.
- **DB global:** Almacena datos globales del programa. Cualquier bloque lógico puede acceder a los datos de un DB global.

La programación del PLC del proyecto se ha realizado en base a un bloque de organización principal (*Main OB1*) en el cual se llama a un bloque de función (*FBmanual*). El programa está pensado para dos modos de funcionamiento para el control del motor desde el autómatas: manual y automático, pero este último dada su sencillez, se ha programado en el propio *Main*.

Además, dado que estamos trabajando con PLC y módulos adicionales de entradas y salidas del perfil PROFIsafe, debemos de incluir obligatoriamente un bloque de función adicional *Main_Safety_RTG1 [FB0]*, donde añadiremos las entradas y salidas de dichos módulos. En caso contrario, se producirá error al compilar. También se incluyen otros bloques de programa por defecto (*System blocks*) que no se empezaron en el proyecto.

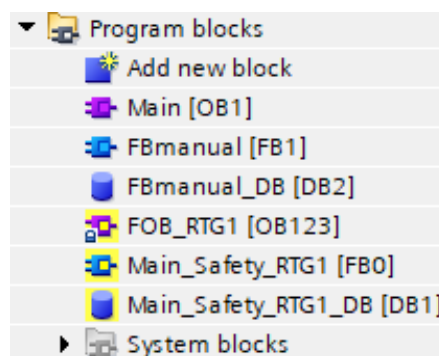


Figura 4.30: TIA Portal. Bloques de programación ladder usados en el proyecto

4.5.1.1 Main [OB1]

Es el **bloque principal del programa**. Se ejecuta de forma cíclica y en cada ciclo los segmentos del bloque son ejecutados secuencialmente. Contiene llamadas a bloques de función, lo cual hace el programa más organizado, que suelen tener alguna condición para poder realizar la llamada y “entrar” en el FB.

▶ Block title: MAIN OB1
▶ Network 1: Seta de emergencia (contactores).
▶ Network 2: En caso de emergencia, reset de motor. Parpadeo de señal roja rápido.
▶ Network 3: Eleccion y validacion modo automatico, con motor parado; y luz de modo automatico.
▶ Network 4: Maniobra manual.
▶ Network 5: Maniobra automatica.
▶ Network 6: RESET tras SOS (seta de emergencia).
▶ Network 7: Prueba del bus CAN.

Figura 4.31: TIA Portal. Segmentos del Main OB1

4.5.1.2 Main_Safety_RTG1 [FB0]

Como hemos comentado, necesitamos incluir este bloque de función para evitar errores de compilación. Básicamente, la principal diferencia entre los bloques de función normales y los bloques de función de seguridad es que los segundos están diseñados específicamente para programar funciones de seguridad en una aplicación. Además, estos bloques de función solo se pueden utilizar en conjunto con hardware de seguridad certificado, como son los módulos PROFIsafe.

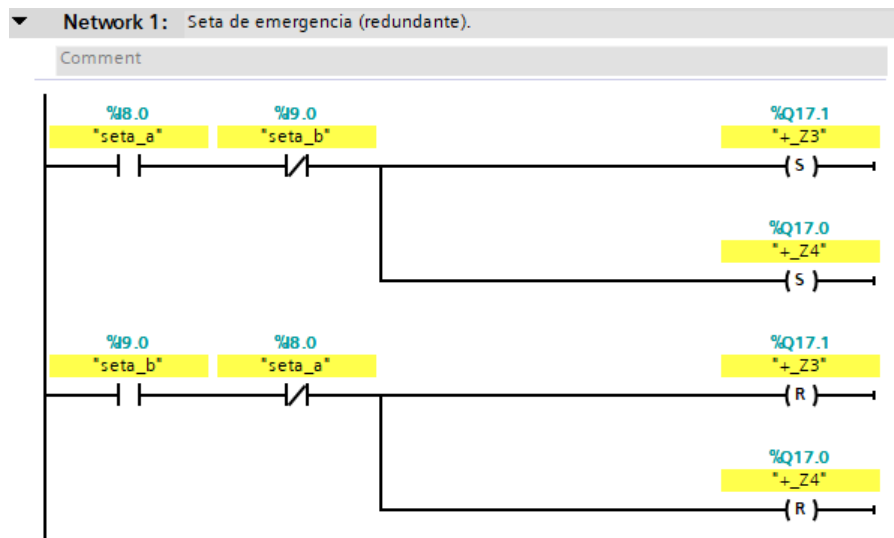


Figura 4.32: TIA Portal. Segmento del Main_Safety_RTG1

En este incluimos únicamente las entradas y salidas de seguridad, y las programaremos para ser las señales que activen y desactiven los contactores, y con ello el suministro eléctrico del resto del circuito.

4.5.1.3 FBmanual [FB1]

Incluye la programación necesaria para el correcto funcionamiento del modo manual.

▶ Block title: FBmanual (FB1)
▶ Network 1: RESET tras SOS (seta de emergencia).
▶ Network 2: Para luz verde y roja iniciales.
▶ Network 3: ON motor.
▶ Network 4: OFF motor.
▶ Network 5: Luces funcionamiento. Intermitente con motor sin arrancar. Fijo con motor arrancado.
▶ Network 6: Selección del sentido de giro. Sentido horario por defecto.
▶ Network 7: Luces del sentido del giro.

Figura 4.33: TIA Portal. Segmentos del FBmanual

→ El código completo, así como los Bloques de Datos, se adjuntan en el apartado de anexos: ANEXO II. Bloques de programación de TIA Portal.

4.5.2 Pasarela (PN/CAN Link). Protocolo CAN

A continuación se desarrollan todos los pasos que se han seguido para llegar al punto actual de la comunicación entre el PLC y el *driver* del motor.

Inicialmente añadimos a nuestro proyecto en TIA Portal el dispositivo con el que vamos a trabajar: la pasarela *PN/CAN Link*, que nos permite pasar del protocolo *ProfiNet* (en verde en la figura 4.34) al protocolo *CANopen*, usado por el *driver*. También deberemos darle un nombre y asignarle una dirección IP válida. La vista del proyecto quedará de la siguiente manera:

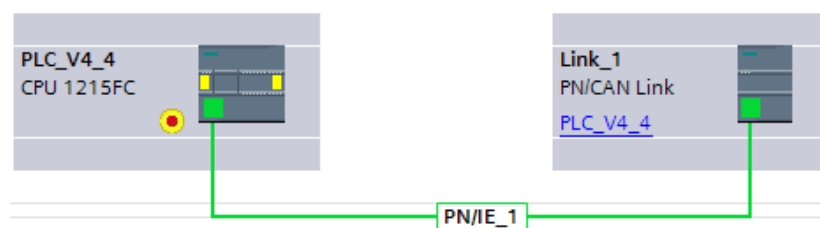


Figura 4.34: TIA Portal. Vista de red en TIA Portal al añadir la pasarela PN/CAN Link

Se debe decidir el modo en el que queremos configurar la pasarela. Como se ha comentado previamente, la opción elegida es *CANopen manager*.

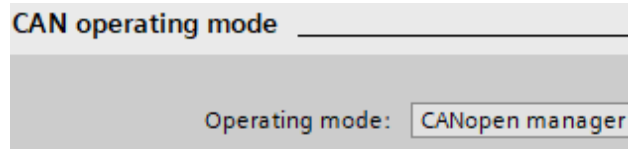


Figura 4.35: TIA Portal. Modo de configuración de la pasarela (*CANopen manager*)

A continuación añadimos un nodo universal a nuestra red CANopen, en el cual importamos el archivo *.eds* proporcionado por el fabricante. Este archivo contiene todos los datos relacionados, en este caso, con el *driver* como por ejemplo el modelo, su fabricante, la versión de *firmware* o las palabras que se van a intercambiar, recogidas en un diccionario de objetos general (*Object dictionary-OD*).

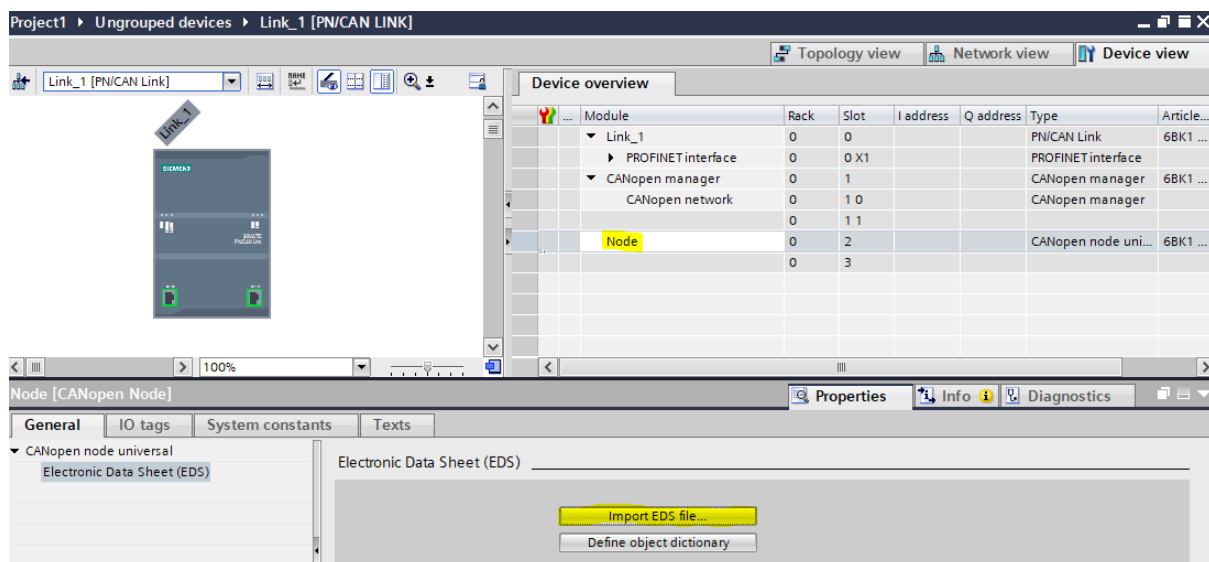


Figura 4.36: TIA Portal. Importar archivo *.eds* al nodo universal

Este proceso nos creará automáticamente en el nodo las PDOs que se van a intercambiar entre este (esclavo) y la pasarela (maestro), siendo únicamente necesario activarlas. Dichas PDOs pueden ser palabras que el nodo reciba o que el nodo envía, pues la comunicación es bidireccional mediante un único cable. Eso sí, dejando claro que siempre será **el maestro el que inicie la comunicación**.

El procedimiento que se debe seguir es crear en el manager las mismas PDOs, pero en el sentido contrario de la comunicación, es decir, si se trata de una palabra que el nodo envía (*“Send PDO”*), en el manager se debe de recibir dicha palabra (*“Receive PDO”*), y viceversa. Para poder crear las PDOs, antes se deben de crear

las palabras que contendrá cada una de ellas, denominadas como “*Transmit data object dictionary*” o “*Receive data object dictionary*”. Estas se pueden crear arrastrando desde las herramientas de configuración específicas de la pasarela.

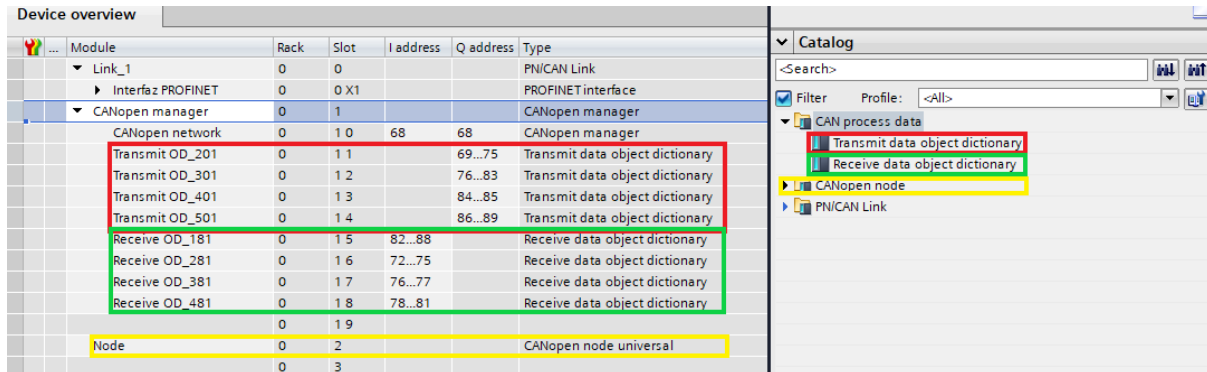


Figura 4.37: TIA Portal. Creando las PDOs en el lado del maestro

Para poder crear correctamente las palabras del OD, y por consiguiente las PDOs, es necesario que las características de las palabras en el lado del manager concuerden con las correspondientes en el lado del nodo:

• **COB-ID.**

Se puede decir que es la identificación de la PDO. Su formato siempre es el de un número (por ejemplo, 100) + el número de nodo. Hablaremos del número de nodo más adelante.

• **Formato.**

Por defecto de 11 bits.

• **Tipo de transmisión.**

Puede ser:

- Síncrono.
- Asíncrono.
- Cíclico síncrono.
- Acíclico síncrono.
- Evento específico definido por el fabricante (del nodo).
- Evento específico definido por la aplicación.

En nuestro caso concreto, las PDOs que se crean al importar el .eds son de este último tipo.



• **Tipo de dato.**

Tamaño. Las empleadas en el proyecto son siempre *Unsigned* o *Integer*, variando el número de bits. Se asignan automáticamente unas direcciones de entrada o de salida en función del tamaño de la PDO (y de los OD que engloba).

Transmit PDOs del nodo (luego necesario crear “Receive data OD” en el manager para agruparlos en las siguientes PDOs)

PDO-001	COB-ID: 181
---------	-------------

Índice OD	Nombre	Tipo de dato
6041.00	<i>Statusword</i>	Unsigned 16
6061.00	<i>Modes of operation display</i>	Integer 08

PDO-002	COB-ID: 281
---------	-------------

Índice OD	Nombre	Tipo de dato
6064.00	<i>Position actual value</i>	Integer 32

PDO-003	COB-ID: 381
---------	-------------

Índice OD	Nombre	Tipo de dato
6044.00	<i>VI velocity actual value</i>	Integer 16

PDO-004	COB-ID: 481
---------	-------------

Índice OD	Nombre	Tipo de dato
60FD.00	<i>Digital inputs</i>	Unsigned 32

Tabla 4.7: Transmit PDOs del nodo



Receive PDOs del nodo (luego necesario crear “Transmit data OD” en el manager para agruparlos en las siguientes PDOs)

PDO-001	COB-ID: 201
---------	-------------

Índice OD	Nombre	Tipo de dato
6040.00	<i>Controlword</i>	Unsigned 16
6060.00	<i>Modes of operation</i>	Integer 08
3202.00	<i>Motor drive submode select</i>	Unsigned 32

PDO-002	COB-ID: 301
---------	-------------

Índice OD	Nombre	Tipo de dato
607A.00	<i>Target position</i>	Integer 32
6081.00	<i>Profile velocity</i>	Unsigned 32

PDO-003	COB-ID: 401
---------	-------------

Índice OD	Nombre	Tipo de dato
6042.00	<i>vl target velocity</i>	Integer 16

PDO-004	COB-ID: 501
---------	-------------

Índice OD	Nombre	Tipo de dato
60FE.00	<i>Digital outputs #1</i>	Unsigned 32

Tabla 4.8: Receive PDOs del nodo

Si todas estas PDOs definidas concuerdan en ambos lados de la comunicación, el *software* nos asignará automáticamente el elemento del bus CAN con el que serán intercambiadas.

PDO	Active	COB ID	Format	Exchanged with	Transmission type	nth SYNC	Event timer	Inhibit time
PDO-001	<input checked="" type="checkbox"/>	181	Hex 11 bit	CANopen manager	Application-specific event control	0	ms 100	x10...
PDO-002	<input checked="" type="checkbox"/>	281	Hex 11 bit	CANopen manager	Application-specific event control	0	ms 100	x10...
PDO-003	<input checked="" type="checkbox"/>	381	Hex 11 bit	CANopen manager	Application-specific event control	0	ms 100	x10...
PDO-004	<input checked="" type="checkbox"/>	481	Hex 11 bit	CANopen manager	Application-specific event control	0	ms 100	x10...
PDO-005	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	ms 100	x10...
PDO-006	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	ms 100	x10...
PDO-007	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	ms 100	x10...
PDO-008	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	ms 100	x10...

Figura 4.38: TIA Portal. Vista general de las Transmit PDOs del nodo

Disable...	PDO	Acti...	COB ID	Format	Exchanged with	Transmission type	nth SYNC
<input type="checkbox"/>	PDO-001	<input checked="" type="checkbox"/>	181	Hex 11 bit	Node	Application-specific event control	
<input type="checkbox"/>	PDO-002	<input checked="" type="checkbox"/>	281	Hex 11 bit	Node	Application-specific event control	
<input type="checkbox"/>	PDO-003	<input checked="" type="checkbox"/>	381	Hex 11 bit	Node	Application-specific event control	
<input type="checkbox"/>	PDO-004	<input checked="" type="checkbox"/>	481	Hex 11 bit	Node	Application-specific event control	

Figura 4.39: TIA Portal. Vista general de las Receive PDOs del manager

Event timer	PDO	Acti...	COB ID	Format	Exchanged with	Transmission type	nth SYNC	Inhibit time
0 ms	PDO-001	<input checked="" type="checkbox"/>	201	Hex 11 bit	CANopen manager	Application-specific event control	0	x10...
0 ms	PDO-002	<input checked="" type="checkbox"/>	301	Hex 11 bit	CANopen manager	Application-specific event control	0	x10...
0 ms	PDO-003	<input checked="" type="checkbox"/>	401	Hex 11 bit	CANopen manager	Application-specific event control	0	x10...
0 ms	PDO-004	<input checked="" type="checkbox"/>	501	Hex 11 bit	CANopen manager	Application-specific event control	0	x10...
0 ms	PDO-005	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	x10...
0 ms	PDO-006	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	x10...
0 ms	PDO-007	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	x10...
0 ms	PDO-008	<input type="checkbox"/>	000	Hex 11 bit		Application-specific event control	0	x10...

Figura 4.40: TIA Portal. Vista general de las Receive PDOs del nodo

PDO	Active	COB ID	Format	Exchanged with	Transmission type	nth SYNC	Event timer	Inhibit time
PDO-001	<input checked="" type="checkbox"/>	201	Hex 11 bit	Node	Application-specific event control		0 ms 0	x10...
PDO-002	<input checked="" type="checkbox"/>	301	Hex 11 bit	Node	Application-specific event control		0 ms 0	x10...
PDO-003	<input checked="" type="checkbox"/>	401	Hex 11 bit	Node	Application-specific event control		0 ms 0	x10...
PDO-004	<input checked="" type="checkbox"/>	501	Hex 11 bit	Node	Application-specific event control		0 ms 0	x10...

Figura 4.41: TIA Portal. Vista general de las Transmit PDOs del manager

Tal y como se puede comprobar, las PDOs en ambos lados de la comunicación coinciden pues las características de todas ellas son las mismas. Cada PDO de las figuras anteriores contiene las palabras (OD) definidas en las tablas 4.7 y 4.8.

Para acabar la configuración, se debe asignar un número de nodo (*Node ID*) para la pasarela y para el *driver*, y un valor para la velocidad de transmisión. Estos valores también deben coincidir en ambos elementos; en caso contrario, tendríamos un error de comunicación.

Por ejemplo, para el caso del *Node ID*, asignaremos el valor de 1 al esclavo (al *driver*) y el valor de 2 al maestro (a la pasarela). Es por ello que las PDOs de las *tablas 4.7 y 4.8* siempre acaban en 1. Por otro lado, el valor de la velocidad de transmisión la ajustamos a 1000 kbps, valor por defecto del *driver* (aunque podría ser otro).

Para ajustar el valor del *Node ID* y de la velocidad de transmisión en el *driver*, debemos hacer uso de dos ruletas físicas especiales situadas en el lateral del mismo. Estas ruletas permiten asignar valores entre 0 y la letra F (en hexadecimal), de tal manera que la suma de ambos valores en hexadecimal dará un valor de *Node ID* y de velocidad de transmisión, según la siguiente figura.

Node-ID and baud rate

The following table shows the possibilities that arise for the rotary switches.

Number combinations of the rotary switches		Node-ID	Baud rate
dec	hex		
0	0	Object 2009 _h	1 MBd fixed
1-127	1-7F	Number of the rotary switches	1 MBd fixed
128	80	Object 2009 _h	Object 2005 _h
129-255	81-FF	(Number of the rotary switches)-128	Object 2005 _h

Figura 4.42: Tabla de los posibles valores de *Node ID* y velocidad de transmisión

Comprobamos los valores que hemos asignado mediante el *software* del *driver*, *Plug and Drive Studio*, en el apartado de *Object Dictionary*.

Description	Index	Sub-Index	Access	Type	Value	Decimal Value
can						
1 CANopen Baudrate	2005	00	read/write	UNSIGNED8	88	136
2 CANopen Config - Highest sub-index supported	2007	00	read only	UNSIGNED8	01	1
3 CANopen Config - BL Config	2007	01	read/write	UNSIGNED32	00000000	0
4 CANopen NodeID	2009	00	read/write	UNSIGNED8	01	1

Figura 4.43: Valores de *Node ID* y velocidad de transmisión desde *PnDS*

En *TIA Portal* debemos configurar los parámetros tanto del esclavo como de la pasarela. Nos aseguramos que estos valores sean los mismos, para que exista concordancia.

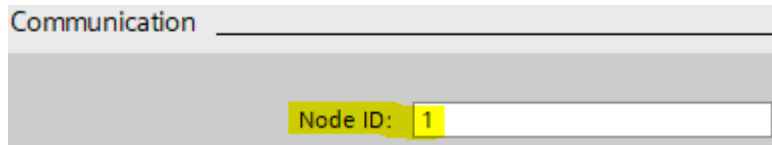


Figura 4.44: TIA Portal. Valor de Node ID del esclavo

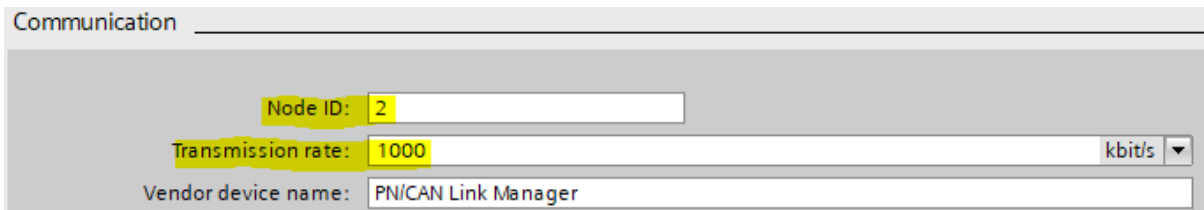


Figura 4.45: TIA Portal. Valor de Node ID del maestro

En este punto tenemos hecha la configuración para la comunicación CANopen en TIA Portal. Debemos activar los bits 0 y 2 del *byte* de control de la pasarela para ver si existen errores. Esto lo hacemos mediante el PLC, mediante una variable de memoria cualquiera (“*Forzado Bit Configuración*”). Por otro lado, el bit 1 de control lo usamos para *resetear* el bus en caso de existir errores en la comunicación.

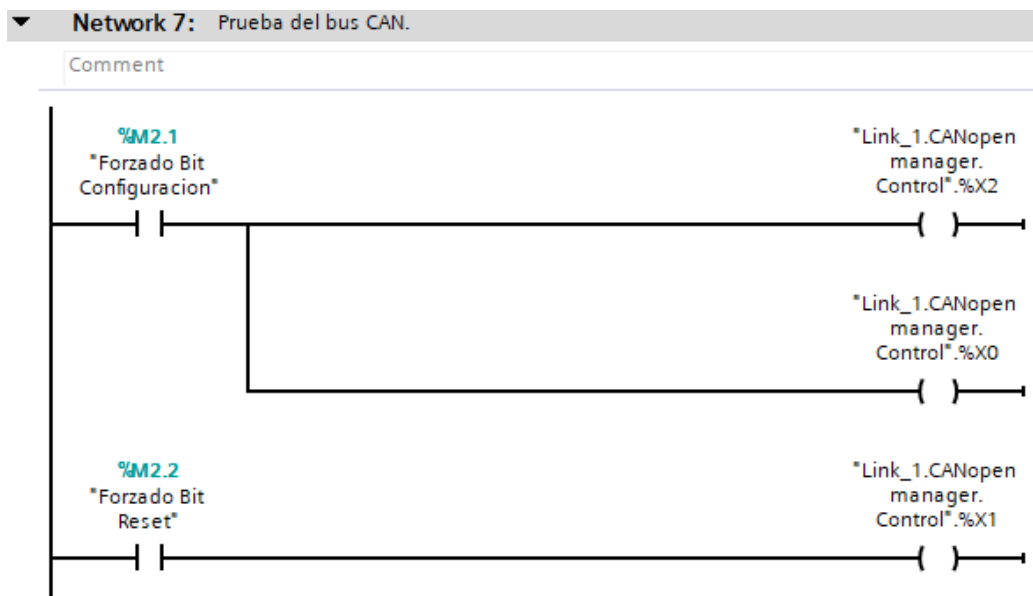


Figura 4.46: TIA Portal. Activar o desactivar bus de comunicación

→ Los fallos encontrados durante este proceso se comentarán en el apartado 6. Pruebas y resultados.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática

Capítulo 5. ESTUDIO ECONÓMICO

Uno de los aspectos más importantes de la ingeniería a la hora de desarrollar diferentes proyectos, con el objetivo de rentabilizar al máximo su fabricación, es realizar un **estudio económico** del mismo. Se deben tener en cuenta todos los costes involucrados en su proyección para conseguir el óptimo balance entre costes y gastos.

En lo sucesivo se distinguirá entre dos tipos de costes, **directos e indirectos**, para terminar con una estimación del total.

5.1 Costes directos

En este subapartado están incluidos todos los costes que guardan una **relación estrecha con el resultado final** del proyecto, así como costes del personal a cargo del mismo, costes del equipo y software necesarios y coste de materiales.

5.1.1 Costes de personal

Se realizan los cálculos teniendo en cuenta el hecho que el proyecto en su totalidad ha sido realizado por un Ingeniero Técnico, concretamente de la especialidad de Electrónica Industrial y Automática.

Se supone un sueldo bruto anual de 22.500€ para los recién egresados, como es nuestro caso, al que hay que sumar un porcentaje del 35% de cotización a la seguridad social. Por lo tanto, el coste anual del personal lo calculamos de la siguiente forma:

$$\text{Coste anual del personal} = 22.500 \cdot 1,35 = 30.375\text{€} \quad (5.1)$$

A continuación eliminamos del año laboral días festivos, vacaciones y fines de semana:

Días festivos (Castilla y León)	Días de fines de semana	Días de vacaciones	Total días no laborables	Total días laborables
14	105	30	149	216

Tabla 5.1: Días laborables en Castilla y León para el año 2023

Y teniendo en cuenta que la jornada laboral de un contrato general es de 8 horas diarias, calculamos el número de horas trabajadas en dicho año laboral:

$$\text{Horas trabajadas al año} = \frac{216 \text{ días}}{\text{año}} \cdot \frac{8 \text{ horas}}{\text{días}} = 1728 \frac{\text{horas}}{\text{año}} \quad (5.2)$$

Nuestro objetivo es saber el coste por horas de la mano de obra:

$$\text{Coste por hora} = \frac{30.375 \text{ €}}{1728 \text{ horas}} = 17,6 \frac{\text{€}}{\text{hora}} \quad (5.3)$$

Obtenemos de la siguiente tabla las horas dedicadas a la totalidad del proyecto, desglosadas (aproximadamente) en las mismas dedicadas a cada apartado.

Tarea	Tiempo empleado
Estudio inicial del problema	16 horas
Formación (búsqueda de documentación)	16 horas
Emplazamiento de los componentes y conexionado	40 horas
Movimiento del motor	24 horas
Programación del PLC	32 horas
Configuración de la pasarela (PN/CAN Link)	96 horas
Puesta a punto y corrección de errores	16 horas
Documentación	60 horas
TOTAL	300 horas

Tabla 5.2: Desglose de tiempo dedicado al proyecto

Concluimos entonces el coste del personal de la siguiente manera:

$$\text{Costes de personal} = 300 \text{ horas} \cdot \frac{17,6 \text{ €}}{\text{horas}} = 5280 \text{ €} \quad (5.4)$$

5.1.2 Costes de equipo y software

Se incluyen aquí tanto los costes del ordenador, como del software empleado.

Producto	Coste (€)
Ordenador	748
Licencia Step 7 v15.1 Basic	257
TOTAL	1.005

Tabla 5.3: Costes de equipo y software

5.1.3 Costes de material

La tabla que sigue incluye todos los elementos que incorpora la maqueta desarrollada, así como su referencia y nombre de la empresa fabricante para una identificación inequívoca. Destaca la comparación entre precios comprando una única unidad o bien grandes cantidades.

Artículo	Referencia	Fabricante	Ud.	Precio 1ud (€)	Precio al por mayor (€)	Página web
Rueda	WD14050-5 616-23X	Nanotec	1	479,8	407,83	Nanotec
Motor	DB59L0240 35-B	Nanotec	1	119	101,15	Nanotec
Frenos	BWA-1,5-6,3 5	Nanotec	1	62,6	56,34	Nanotec
Encoder	NME-UVW- T14	Nanotec	1	79,5	71,55	Nanotec
Driver	C5-E-2-09	Nanotec	1	274,8	247,32	Nanotec
PLC	6ES7215-1A F40-0XB0	Siemens	1	1.035	820	WiAutomati on ES
Módulo de entradas	6ES7226-6B A32-0XB0	Siemens	1	216,07	197,53	Siemens
Módulo de salidas	6ES7226-6D A32-0XB0	Siemens	1	227,36	199,98	Siemens
Pasarela PN/CAN	6BK1620-0A A00-0AA0	Siemens	1	531,54	513,34	Eléctrica Automation

Artículo	Referencia	Fabricante	Ud.	Precio 1ud (€)	Precio al por mayor (€)	Página web
Contactador	LC1D18BL	Schneider	2	152,03	143,17	Schneider
Fuente de alimentación	ABL7RE2405	Schneider	1	280	168,74	RGB Automatyka
Seta de emergencia	XB4BS8445	Schneider	1	43,3	40,66	Farnell
Pulsador luminoso	XB4BW33B5	Schneider	4	34,75	31,95	Farnell
Selector dos posiciones	XB4BD21	Schneider	1	23,9	22,47	Farnell
Caja estanca	XAPD1501	Schneider	6	68,92	60,41	RS
Cable	-	Balluff	20	0,44	0,44	Amazon
Bornas	3212934	Phoenix Contact	28	3,212	2,12	RS
Bridas	-	ERKO	15	2,5	1,99	Amazon
Cable RJ45	RJ45	RS	2	3,14	2,07	RS
CableD-Sub9	D-SUB9	RS	1	5,12	4,85	RS
Cable del encoder	ZK-NME-13-500-S	Nanotec	1	9,69	8,72	Nanotec

	Precio 1ud (€)	Precio al por mayor (€)
Total	4.386,78	3.739,23

Tabla 5.4: Costes de materiales



5.1.4 Costes totales

Concepto	Coste 1ud (€)	Coste al por mayor (€)
Costes del personal	5.280	
Costes de equipo y software	1.005	
Costes de materiales	4.386,78	3.739,23
Costes directos totales	10.671,78	10.024,23

Tabla 5.5: Costes directos totales

Destaca que la producción en gran escala, a largo plazo, puede suponer un ahorro en el coste de material (y por consiguiente de los costes directos totales) de 628,38€, pero supone una mayor inversión inicial.

5.2 Costes indirectos

Se han incluido como costes indirectos aquellos que no se pueden computar directamente al desarrollo del proyecto. Se incluyen en la *tabla 5.6*.

Concepto	Coste (€)
Consumo de electricidad	60
Internet	100
Costes indirectos totales	160

Tabla 5.6: Costes indirectos totales

5.3 Costes totales

En la *tabla 5.7* se indica el coste total, obtenido como suma de los costes directos e indirectos.

Concepto	Coste 1ud (€)	Coste al por mayor (€)
Costes directos totales	10.671,78	10.024,23



Costes indirectos totales	160	
Costes totales	10.831,78	10.184,23

Tabla 5.7: Costes totales



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática

Capítulo 6. PRUEBAS Y RESULTADOS

Al activar el bus de comunicación CAN entre la pasarela y el driver mediante los bits 0 y 2 del *byte* de control, y después de haber realizado la configuración del apartado 4.5.2 Pasarela (PN/CAN Link). Protocolo CAN), recibimos un error de TIA Portal que dice: “CAN bus not reachable (“Bus Off” state)”.

Esto nos indica que el bus CAN no está operativo por encontrarse en el estado “Bus Off”, es decir, no se pueden enviar o recibir mensajes por dicho bus. Las indicaciones para resolverlo que se nos indica desde el propio TIA Portal es comprobar los parámetros de comunicación (*Node ID* y velocidad de transmisión) y la instalación *hardware*.

Dando por sentado que los parámetros de comunicación concuerdan, se sigue la última indicación. El *driver* incorpora una **resistencia terminal de 120 ohms** que puede y debe ser activada externamente mediante un *switch* físico. Esto permitirá al bus CAN detectar que existe un dispositivo terminal.

La solución de este error en particular viene al añadir también una resistencia terminal en el otro extremo del bus, información que no se había tenido en cuenta por considerar que una sola resistencia era condición suficiente para establecer comunicación.

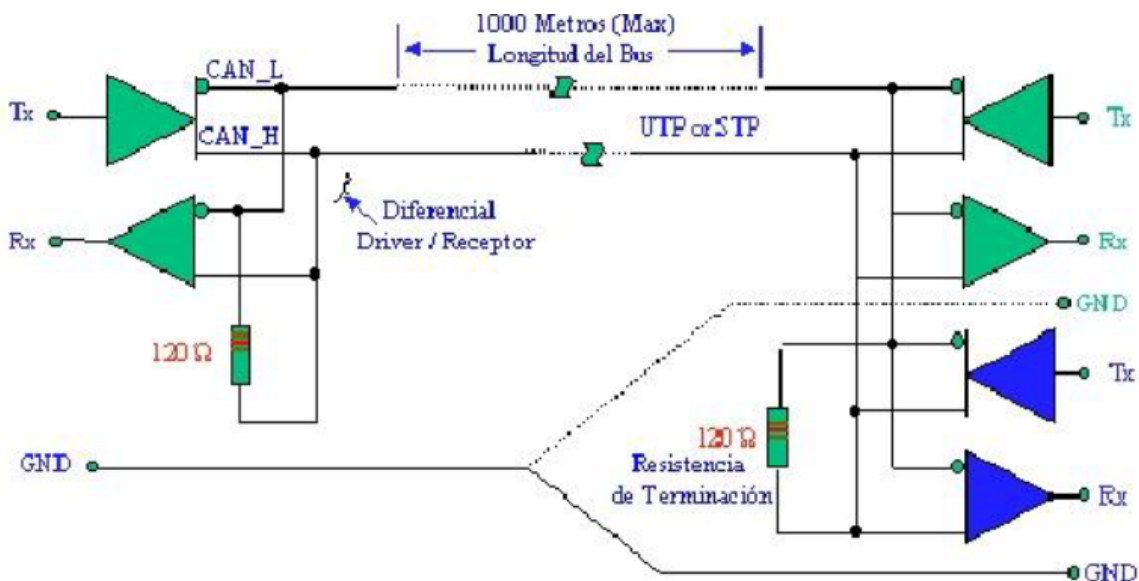
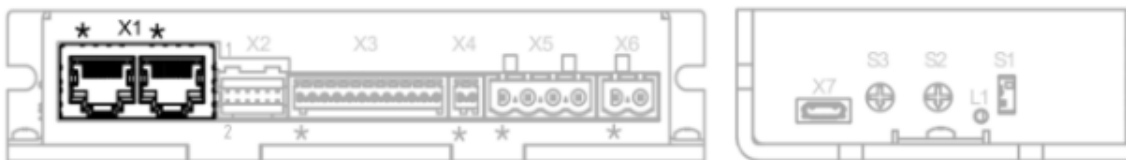


Figura 6.1: Modo de transmisión del protocolo CAN

No obstante, seguimos encontrándonos con el mismo error. La solución pasa por realizar una **correcta asignación de pines** en el cable de comunicación CAN, tanto en el lado del driver (terminación RJ45), como en el lado de la pasarela (terminación D-SUB9).



PIN	Function	Note
1	CAN+	
2	CAN-	
3	CAN GND	Internally connected with pin 7
4	n.c.	
5	n.c.	
6	CAN SHIELD	Connection for the shielding
7	GND	Internally connected with pin 3
8	+UB Logic	24 V DC input, external logic supply for communication, input voltage, current consumption approx. 36 mA

Figura 6.2: Asignación de pines en el lado del driver (RJ45)

PIN	Function	Note
1	--	Terminating resistor 120 Ω (install jumper from pin 1 to pin 2 for use)
2	CAN_L	CAN_L bus cable (dominant low)
3	CAN_GND	CAN Ground
4	--	Not connected
5	CAN_SHLD	CAN shield
6	CAN_GND	CAN Ground
7	CAN_H	CAN_H bus cable (dominant high)
8	--	Not connected
9	--	Not connected

Figura 6.3: Asignación de pines en el lado de la pasarela (D-SUB9)

Como se puede observar de las figuras anteriores, la asignación de pines no coincide en ambos lados del bus. Se procede a coger dos cabezales macho-hembra del cable D-SUB9 y ha realizar un conexionado manual, de tal forma que las señales de *CAN_H*, *CAN_L*, *CAN_SHIELD* y *GND* coincidan, para que se pueda transmitir la información.



Figura 6.4: Cabezales hembra-macho de la terminación D-SUB9



Figura 6.5: Conexión manual del cabezal hembra D-SUB9

Este es el mayor avance en cuanto a comunicación se refiere realizado en el proyecto; este último paso permite eliminar los errores de comunicación existentes al dar corriente al conjunto de los componentes. Es decir, **gracias a esta modificación la pasarela y el driver están en condiciones de intercambiarse información**, siempre y cuando se realice una correcta configuración de las palabras a intercambiar.

No obstante, nos seguimos encontrando con errores en TIA Portal una vez activamos el bus CAN mediante los bits 0 y 2 del *byte* de control. En este caso, el error está provocado por la *PDO 16#1020:16#1* (en hexadecimal).

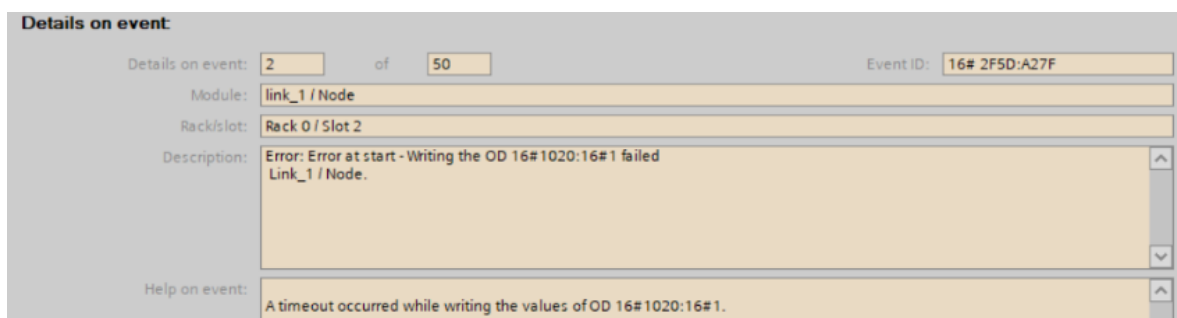


Figura 6.6: TIA Portal. Error con la PDO 16#1020:16#1



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



Capítulo 7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1 Conclusiones

Tras finalizar el presente trabajo se quiere destacar, en primer lugar, la **variedad de temas afines** con la carrera que se han tratado en el desarrollo del mismo.

Por la **parte mecánica**, se han llevado a cabo horas de taller para el correcto montaje de los componentes mediante los elementos de unión necesarios, desde la elección de la tabla de madera y su taladrado hasta el posterior emplazamiento de los dispositivos.

Por la **parte eléctrica**: diferenciación por colores del tipo de cable según el voltaje, etiquetado del mismo, conexionado, incorporación de diferentes elementos de control y potencia y desarrollo del esquema eléctrico completo.

Por la **parte de la automática** se ha trabajado con un PLC de Siemens visto en el grado como es el S7-1200, y se ha desarrollado su programación en el software TIA Portal, así que se han recordado conceptos de programación en lenguaje *ladder* adquiridos previamente.

Por la **parte de las comunicaciones industriales** se ha tratado principalmente con *ProfiNet*, muy relacionado con el *Ethernet* empleado en las industrias 4.0; y con CAN y CANopen que es el protocolo que usan numerosos dispositivos electrónicos como por ejemplo algunos tipos de AGV.

También se ha desarrollado un **análisis económico** del proyecto completo, por considerar de interés analizar el impacto económico de cualquier proyecto real. El desglose realizado permite diferenciar los costes directos e indirectos.

Por último y como se va a ver en el siguiente apartado, el prototipo actual puede servir como punto de partida para futuros proyectos sobre el que realizar cambios y mejoras para ampliar las funcionalidades del mismo. Estos pueden estar relacionados tanto con la parte de montaje y conexionado eléctrico, como con la parte de programación del PLC y de las comunicaciones con el driver.



7.2 Líneas futuras

La principal línea de investigación en este proyecto es obvia: **conseguir una comunicación completa y funcional entre el PLC y el driver del motor, a través de la pasarela**. Destacar que algunas de las opciones que se van a comentar conllevarían asumir un coste económico que habría que valorar.

Si se sigue con los estudios actuales, algunas posibles líneas de trabajo son:

1) CAN-Log.

Como se ha comentado previamente, se ha contactado en numerosas ocasiones con los fabricantes de los componentes utilizados. Las últimas conversaciones con Nanotec en busca de soporte técnico siempre han acabado con una recomendación de usar el dispositivo *USB-to-CAN V2* de la empresa IXXAT [21].

Este dispositivo se conecta al bus CAN y permite monitorizar los mensajes (PDOs y SDOs) que se transmiten por el mismo, con su tiempo de envío/recepción y realizando filtros de selección, **puediendo así diagnosticar los errores existentes en la comunicación**. El dispositivo incorpora además un software compatible con el *Plug and Drive Studio* de Nanotec.

La manera de añadir dicho dispositivo a nuestra red CAN actual es sencilla: únicamente deberemos conectar el pin *Fieldbus Out* del driver al adaptador, y este a nuestro PC. De esta manera tendremos acceso a la información transmitida a través del bus.

Aunque este componente ayudaría notablemente a conseguir nuestro cometido, no ha sido utilizado debido a su coste económico (+400€).

2) ET 200SP CM CAN.

Otra opción es la posibilidad de cambiar el dispositivo que nos conecte el PLC con el driver. En este documento se han detallado los pasos seguidos para la pasarela *Simatic PN/CAN Link*, sobre la cual se ha encontrado sorprendentemente poca bibliografía.

Por ello se contempla emplear el *Simatic ET 200SP CM CAN* [22], un dispositivo también de Siemens sobre el que parece existir más información relacionada con su funcionamiento. Algunas de las ventajas que proporciona este componente respecto del actual, es la incorporación de una biblioteca de



funciones ya desarrolladas para programar los diferentes Nodos que se conecten a la misma.

Por otro lado, esto implicaría un coste económico similar al del *CAN-Log*, además del previo desmontaje de la pasarela actual y el posterior montaje de la nueva, con su consiguiente configuración desde cero para nuestro proyecto.

3) Sin modificar componentes.

Evidentemente se puede conseguir una comunicación con los componentes actuales, sin la necesidad de contar con nuevos dispositivos. Las líneas de investigación en este contexto se deben centrar en **solucionar el problema relacionado con las PDOs** que provocan los errores actuales, así como una posible futura reprogramación del código.

4) AGV.

Como se ha comentado a lo largo de esta memoria, el objetivo final de este proyecto es un **vehículo de guiado automático**. Evidentemente para conseguirlo, lo primero que se debe de abordar es solucionar los errores y alcanzar la plena comunicación entre los componentes.

Una vez en este punto, se debe recolocar la rueda motriz actual en la parte inferior de la tabla (para que se pueda mover), y se pueden incluir mejoras como añadir ruedas “locas” a la maqueta, que permitan el movimiento del prototipo en varios ejes.

También se puede incluir un sistema para que la rueda, y por tanto el AGV, vayan siguiendo una banda magnética del suelo. Este sistema implicaría el uso de la **visión artificial**, y ya lo incorporan grandes factorías del sector automovilístico como por ejemplo Renault.



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



BIBLIOGRAFÍA/REFERENCIAS

- [1] A. Barrientos, L. Felipe Peñín, C. Balaguer y R. Aracil, FUNDAMENTOS DE ROBÓTICA, Madrid: McGraw-Hill, 1997. [Último acceso: 16 de mayo 2023].
- [2] Introducción a la robótica, <<Campus Virtual Universidad de Valladolid, Escuela de Ingenierías Industriales: Introducción a la robótica, Juan Carlos Fraile, Valladolid, Septiembre 2022>> [Último acceso: 21 de mayo 2023].
- [3] J. Ruiz-de-Garibay, «Robótica: Estado del arte» ACADEMIA. Accelerating the world research. [Último acceso: 16 de mayo 2023].
- [4] Tipos de robots manipuladores, <<Robots industriales: tecnología y aplicaciones>> Available: <https://www.sicma21.com/robots-industriales-tecnologia-y-aplicaciones/> [Último acceso: 16 de mayo 2023]
- [5] Gallo de Estrasburgo P. S. S. y R. H. Alberto Brunete, Introducción a la Automatización Industrial, Madrid: Bookdown, 2020.
- [6] D. G. Guzmán, «Historia de la automatización» [En línea] Available: <https://www.preceden.com/timelines/717427-historia-de-la-automatizaci-n-b-ryan-daniel-guti-rrez-guzm-n-1682755> [Último acceso: 16 de mayo 2023].
- [7] Estructura de un PLC, <<Controladores Lógico Programables>> [En línea] Available: <http://automatica.mex.tl/imagesnew/5/0/1/4/2/Presentaci%C3%B3n%20P.L.C..pdf> [Último acceso: 16 de mayo 2023].



- [8] CPU de los PLC, [En línea] Available: <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/ESTRUCTURA%20INTERNA/CPU/cpu.htm> [Último acceso: 16 de mayo 2023]
- [9] CIM, <<Comunicaciones Industriales>> [En línea] Available: <https://ningenia.com/comunicaciones-industriales-i/> [Último acceso: 16 de mayo 2023]
- [10] Buses de campo y Ethernet industrial, <<Campus Virtual Universidad de Valladolid, Escuela de Ingenierías Industriales: Tema 4A, Buses de campo y Ethernet industrial>> [En línea] Available: https://campusvirtual.uva.es/pluginfile.php/3446534/mod_resource/content/3/T4A_Buses%20de%20campo%20y%20Ethernet%20Industrial%202021-22.pdf [Último acceso: 21 de mayo 2023]
- [11] PROFIsafe, <<Seguridad Industrial: descripción general del perfil PROFIsafe>> [En línea] Available: <https://profinetuniversity.com/seguridad-funcional/seguridad-industrial-perfil-profisafe/> [Último acceso: 16 de mayo 2023]
- [12] CANopen, <<CANopen Product Guide>> [En línea]. Available: <https://can-newsletter.org/canopen/canopen> [Último acceso: 16 de mayo 2023]
- [13] José Delgado García y Alberto Herreros López, «UVaDOC,» Mayo 2018. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/41133> [Último acceso: 21 de abril 2023].
- [14] Jorge Villa González y Alberto Herreros López, «UVaDOC,» Enero 2023. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/58679> [Último acceso: 21 de abril 2023].



- [15] Roberto Gámez Pérez y Juan Carlos González Vítores, «Archivo UC3M» Septiembre 2012. [En línea]. Available: <https://e-archivo.uc3m.es/handle/10016/16873> [Último acceso: 21 de abril 2023].
- [16] Juan Moliner Malaxechevarría y Antonio Barrientos, «UPM» 2017. [En línea]. Available: https://oa.upm.es/49142/1/TFG_JUAN_MOLINER_MALAXECHEVARRIA.pdf [Último acceso: 21 de abril 2023].
- [17] Natalia Puente Carreño y Concepción Alicia Monje Micharet, «Archivo UC3M» 2018. [En línea]. Available: <https://e-archivo.uc3m.es/handle/10016/29115> [Último acceso: 21 de abril 2023].
- [18] Ricardo Martínez Martín y Juan Cuquejo Mira, «Repositorio Digital» Septiembre 2022. [En línea]. Available: <https://reunir.unir.net/handle/123456789/13318> [Último acceso: 21 de abril 2023].
- [19] Plug and Drive Studio, <<Nanotec Plug and Drive Studio Start Guide>> [En línea] Available: https://en.nanotec.com/fileadmin/files/Software/Plug_Drive_Studio/Quick_start_guide_1.1.1.pdf?1656012670 [Último acceso 21 de mayo 2023]
- [20] Carril DIN, <<Wikipedia>> [En línea] Available: https://es.wikipedia.org/wiki/Carril_DIN [Último acceso: 16 de mayo 2023]
- [21] IXXAt CAN-USB, <<Ixxat Products>> [En línea]. Available: https://www.ixxat.com/products/pc-interfaces-overview/details/usb-to-can-v2-professional?utm_source=GoogleAds&utm_medium=CPC&utm_campaign=Iberia-EC&gclid=CjwKCAjwxr2iBhBJEiwAdXECwyrms2KjQrvWofOpeMjH0a7LS0-ePmeDIEUdw89FnVtm68KYEeNZ_xoCYIYQAvD_BwE [Último acceso: 16 de mayo 2023]



[22]

ET200, <<Siemens Industry>> [En línea]. Available:
<https://support.industry.siemens.com/cs/document/109779453/simatic-et-200sp-cm-can?dti=0&lc=en-HN> [Último acceso: 16 de mayo 2023]



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

Grado en Ing. Electrónica Industrial y Automática



ANEXOS

ANEXO I. VARIABLES DE PROGRAMA DE TIA PORTAL

NOMBRE	TIPO I	TIPO II	DIRECCIÓN	TABLA QUE LA CONTIENE
Clock_2Hz	Bool	Memoria	%M0.3	Entradas_Salidas
Clock_1Hz	Bool	Memoria	%M0.5	Entradas_Salidas
Forzado Bits Control	Bool	Memoria	%M2.0	Entradas_Salidas
Forzado Bit Configuración	Bool	Memoria	%M2.1	Entradas_Salidas
Forzado Bit Reset	Bool	Memoria	%M2.2	Entradas_Salidas
verde_I	Bool	Entrada	%I0.0	Entradas_Salidas
22.Z4_I	Bool	Entrada	%I0.1	Entradas_Salidas
22.Z3_I	Bool	Entrada	%I0.2	Entradas_Salidas
rojo_I	Bool	Entrada	%I0.3	Entradas_Salidas
giro.izq_I	Bool	Entrada	%I0.4	Entradas_Salidas
giro.drch_I	Bool	Entrada	%I0.5	Entradas_Salidas
selector_I	Bool	Entrada	%I0.6	Entradas_Salidas
seta_a	Bool	Entrada	%I8.0	Entradas_Salidas
seta_b	Bool	Entrada	%I9.0	Entradas_Salidas
verde_Q	Bool	Salida	%Q0.0	Entradas_Salidas
rojo_Q	Bool	Salida	%Q0.1	Entradas_Salidas
nar.izq_Q	Bool	Salida	%Q0.2	Entradas_Salidas
nar.drch_Q	Bool	Salida	%Q0.3	Entradas_Salidas
motor.on_M_Q	Bool	Salida	%Q0.6	Entradas_Salidas
motor.on_A_Q	Bool	Salida	%Q0.7	Entradas_Salidas
motor.drch_Q	Bool	Salida	%Q1.0	Entradas_Salidas
motor.izq_Q	Bool	Salida	%Q1.1	Entradas_Salidas
Z4_+	Bool	Salida	%Q17.0	Entradas_Salidas
Z3_+	Bool	Salida	%Q17.1	Entradas_Salidas

alimentación	Bool	Memoria	%M4.0	Tabla de variables estándar
rst_mmanual	Bool	Memoria	%M4.1	Tabla de variables estándar
estado_auto	Bool	Memoria	%M4.2	Tabla de variables estándar
rst_mauto	Bool	Memoria	%M4.3	Tabla de variables estándar
ok_auto	Bool	Memoria	%M4.4	Tabla de variables estándar
Link_1.CANopen manager.Status	Byte		%IB68	Tabla de variables estándar
Link_1.CANopen manager.Control	Byte		%QB68	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_201.Transmit OD_201.01	UInt		%QW69	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_201.Transmit OD_201.02	SInt		%QB71	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_201.Transmit OD_201.03	UDInt		%QD72	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_301.Transmit OD_301.01	DInt		%QD76	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_301.Transmit OD_301.02	UDInt		%QD80	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_401.Transmit OD_401.01	Int		%QW84	Tabla de variables estándar
Link_1.CANopen manager.Transmit OD_501.Transmit OD_501.01	UDInt		%QD86	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_181.Receive OD_181.01	UInt		%IW82	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_181.Receive OD_181.02	SInt		%IB84	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_281.Receive OD_281.01	DInt		%ID72	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_381.Receive OD_381.01	Int		%IW76	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_481.Receive OD_481.01	UDInt		%ID78	Tabla de variables estándar
Link_1.CANopen manager.Receive OD_181.Receive OD_181.03	UDInt		%ID85	Tabla de variables estándar

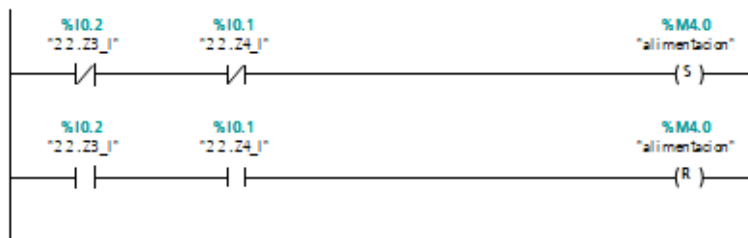
ANEXO II. BLOQUES DE PROGRAMACIÓN DE TIA PORTAL

Main (OB1)

- ▶ **Block title:** MAIN OB1
- ▶ **Network 1:** Seta de emergencia (contactores).
- ▶ **Network 2:** En caso de emergencia, reset de motor. Parpadeo de señal roja rápido.
- ▶ **Network 3:** Eleccion y validacion modo automatico, con motor parado; y luz de modo automatico.
- ▶ **Network 4:** Maniobra manual.
- ▶ **Network 5:** Maniobra automatica.
- ▶ **Network 6:** RESET tras SOS (seta de emergencia).
- ▶ **Network 7:** Prueba del bus CAN.

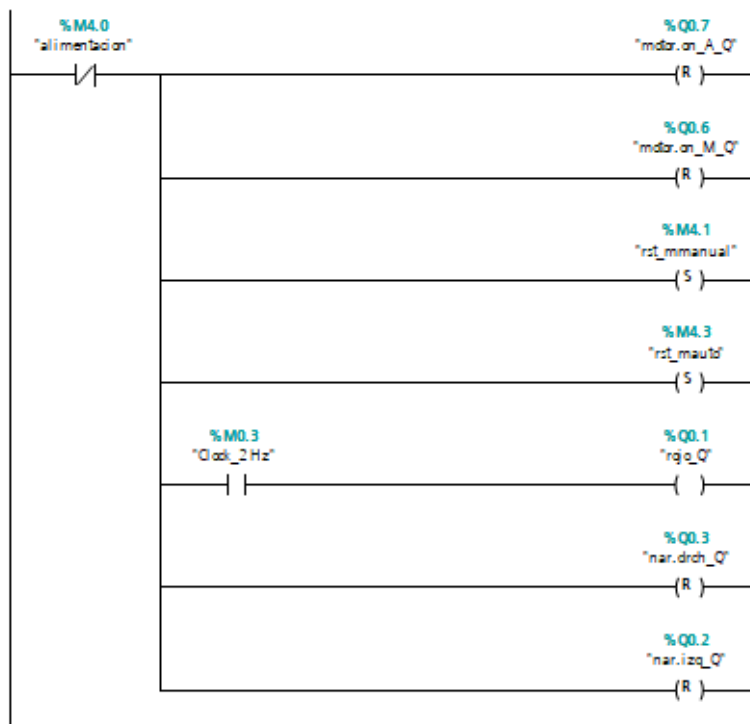
Network 1: Seta de emergencia (contactores).

Comment



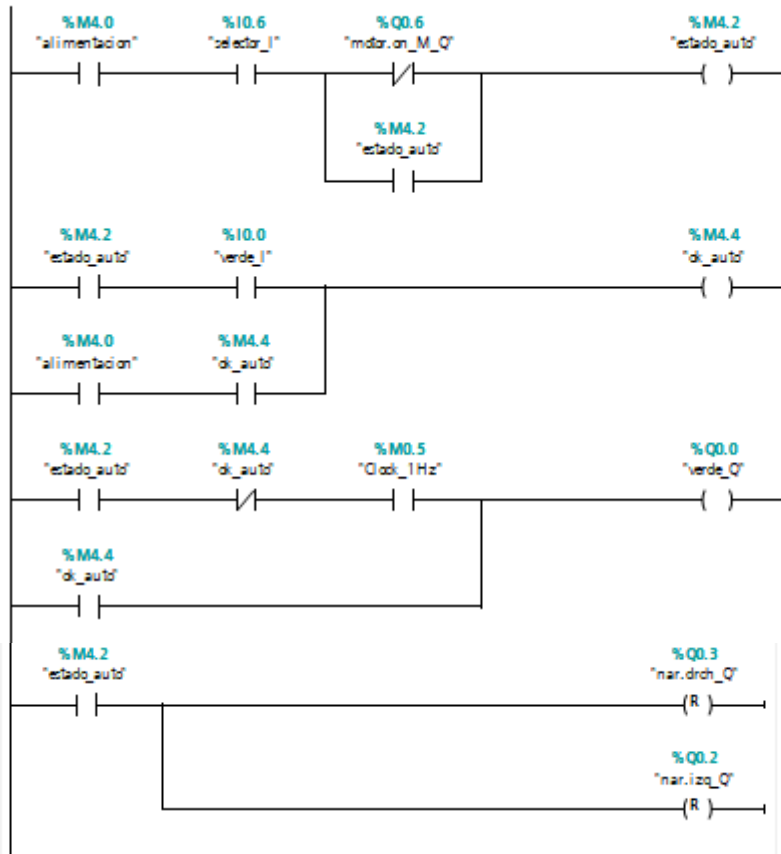
Network 2: En caso de emergencia, reset de motor. Parpadeo de señal roja rápido.

Comment



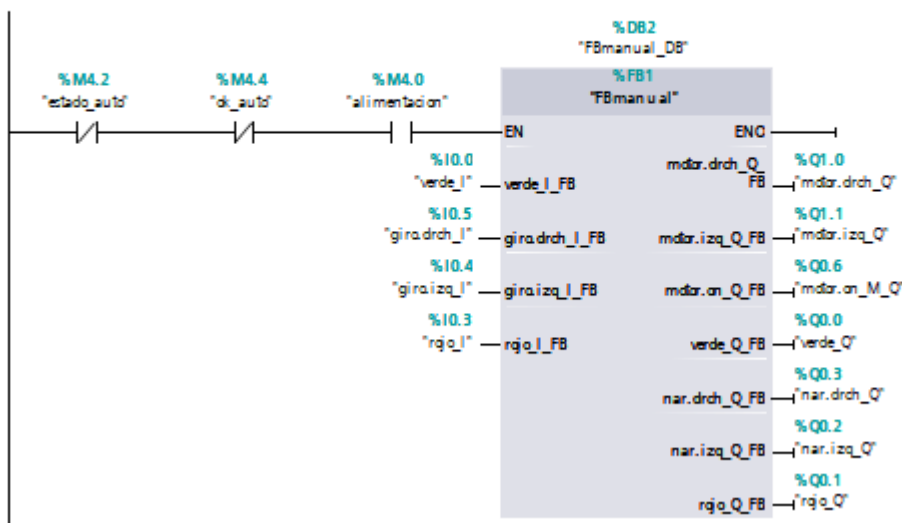
Network 3: Eleccion y validacion modo automatico, con motor parado; y luz de modo automatico.

Comment



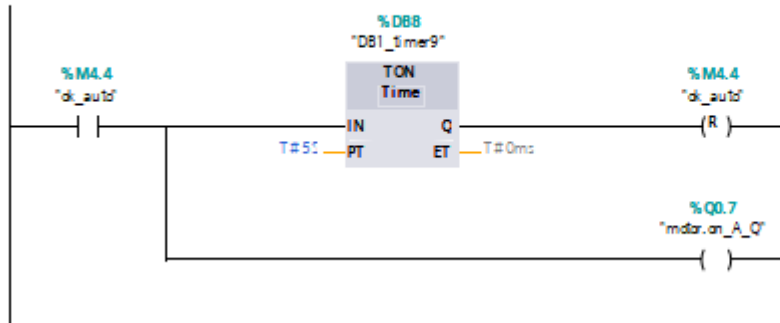
Network 4: Maniobra manual.

Comment



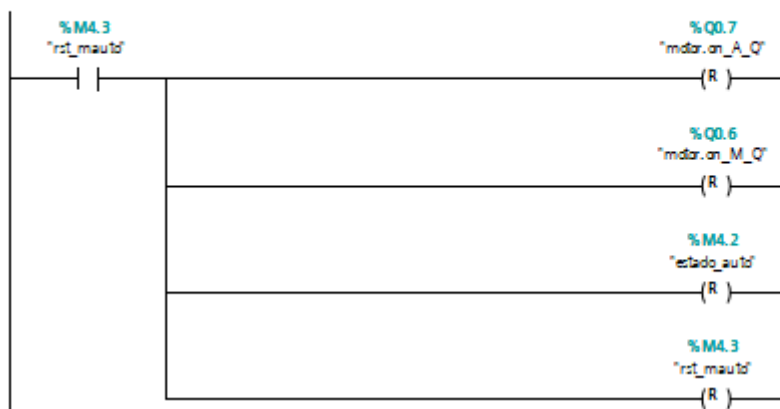
Network 5: Maniobra automática.

Comment



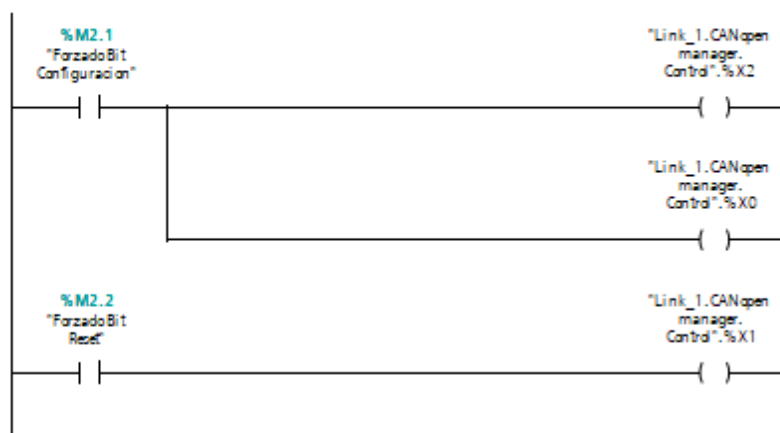
Network 6: RESET tras SOS (seta de emergencia).

Comment



Network 7: Prueba del bus CAN.

Comment

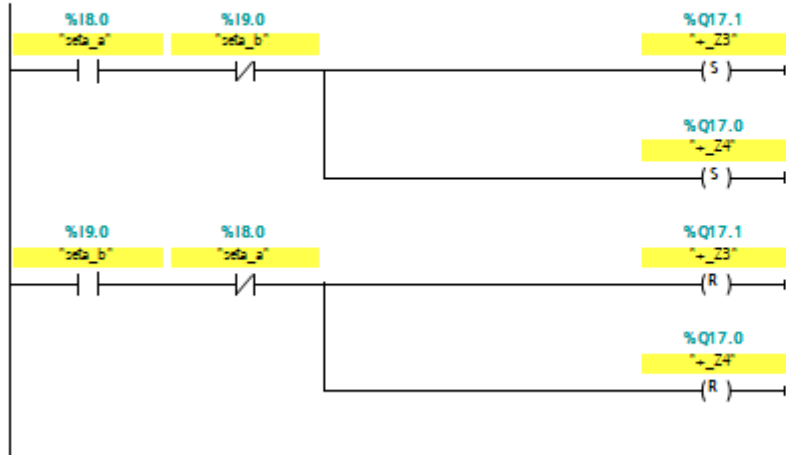




Main_Safety_RTG1 (FBo)

Network 1: Seta de emergencia (redundante).

Comment

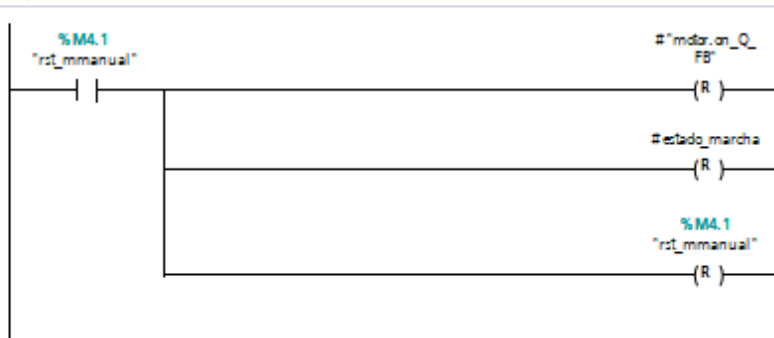


FBmanual (FB1)

- ▶ **Block title:** FBmanual (FB1)
- ▶ **Network 1:** RESET tras SOS (seta de emergencia).
- ▶ **Network 2:** Para luz verde y roja iniciales.
- ▶ **Network 3:** ON motor.
- ▶ **Network 4:** OFF motor.
- ▶ **Network 5:** Luces funcionamiento. Intermitente con motor sin arrancar. Fijo con motor arrancado.
- ▶ **Network 6:** Selecccion del sentido de giro. Sentido horario por defecto.
- ▶ **Network 7:** Luces del sentido del giro.

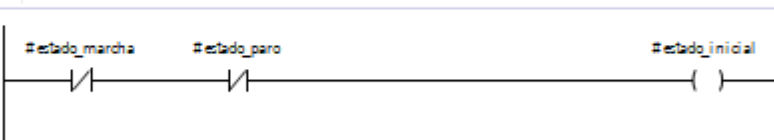
Network 1: RESET tras SOS (seta de emergencia).

Comment



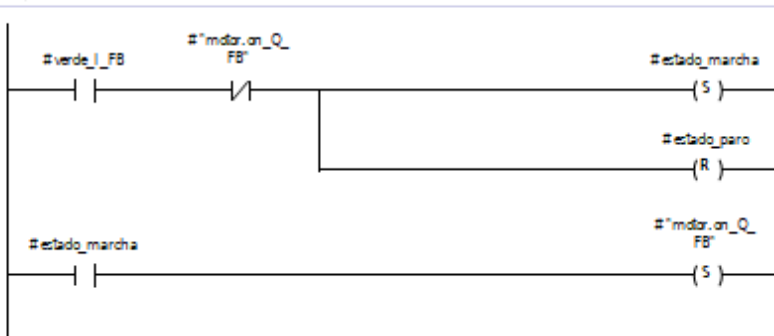
Network 2: Para luz verde y roja iniciales.

Comment



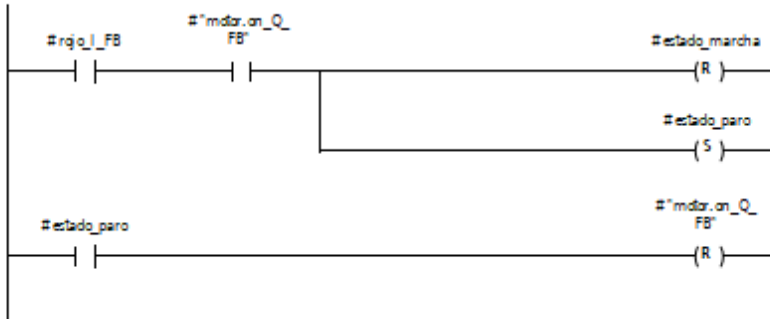
Network 3: ON motor.

Comment



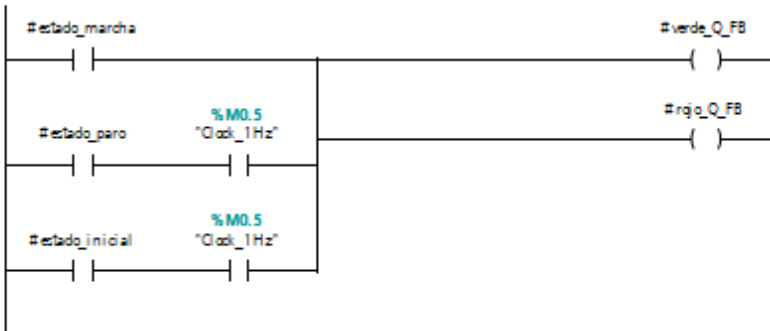
▼ **Network 4:** OFF motor.

Comment



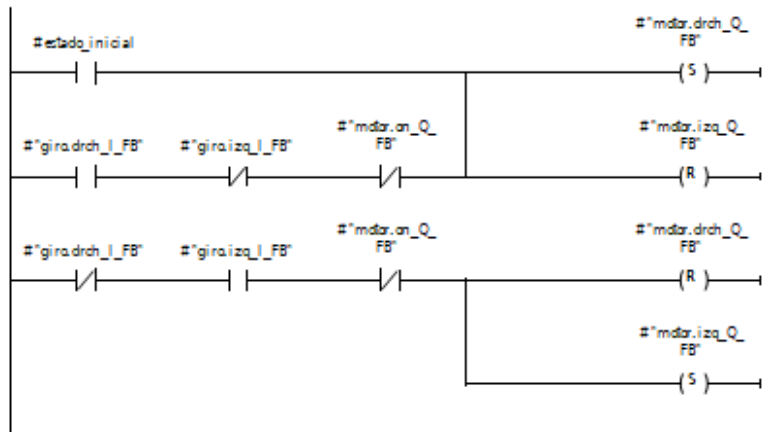
▼ **Network 5:** Luces funcionamiento. Intermitente con motor sin arrancar. Fijo con motor arrancado.

Comment



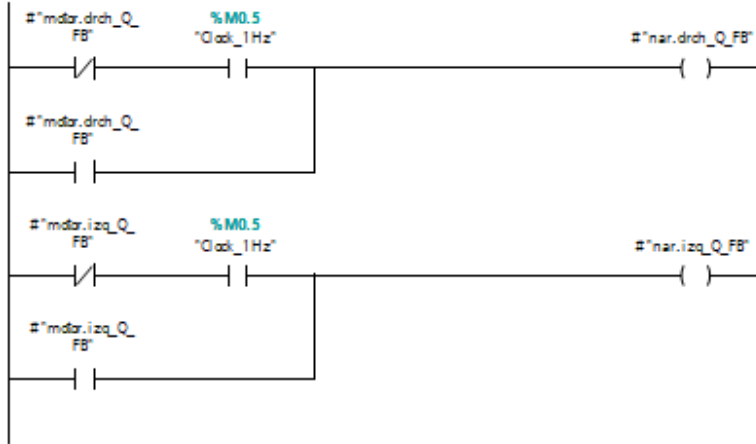
▼ **Network 6:** Selección del sentido de giro. Sentido horario por defecto.

Comment



Network 7: Luces del sentido del giro.

Comment

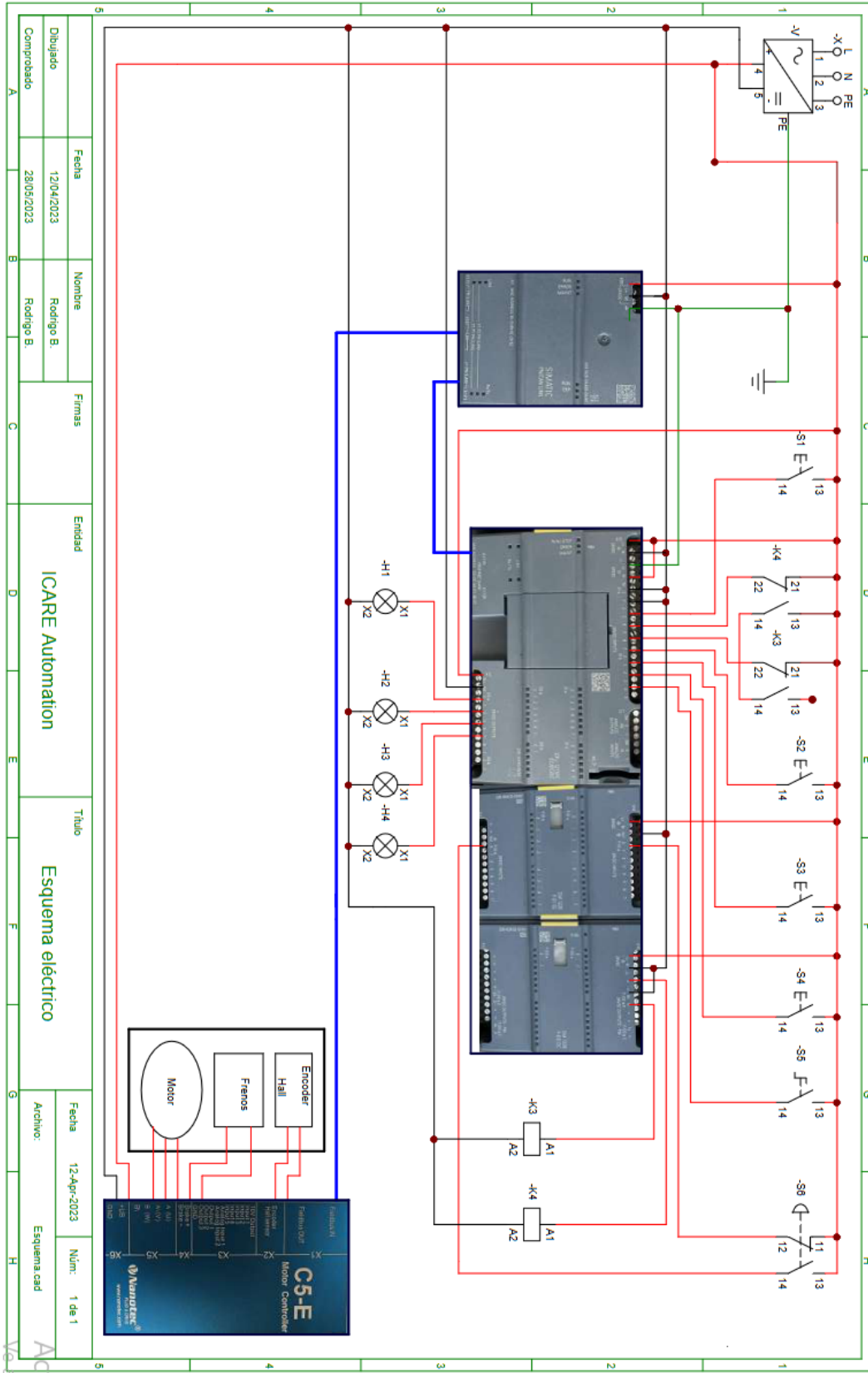




FBmanual_DB (DB2)

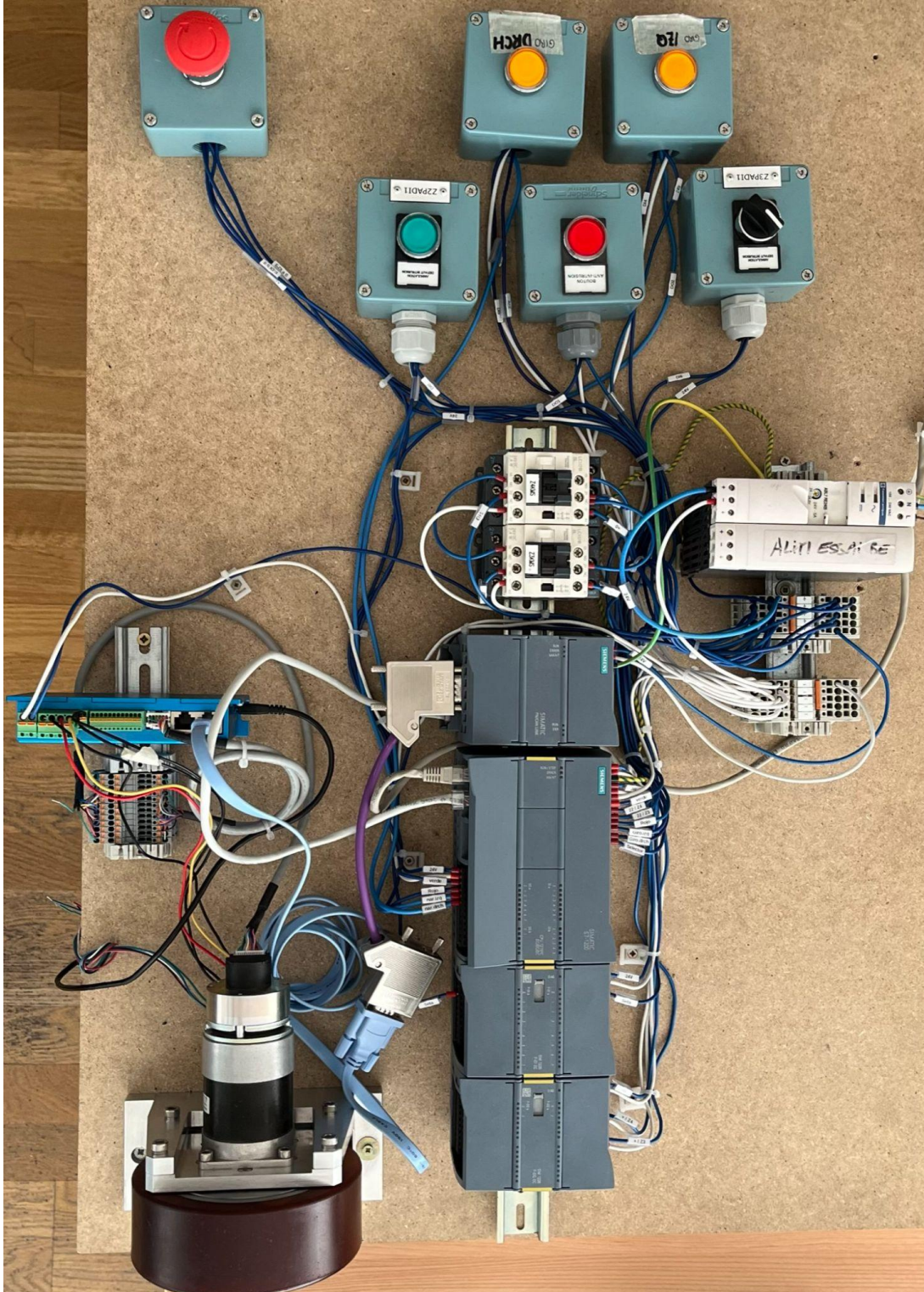
FBmanual_DB			
	Name	Data type	Start value
1	▼ Input		
2	verde_I_FB	Bool	false
3	giro.drch_I_FB	Bool	false
4	giro.izq_I_FB	Bool	false
5	rojo_I_FB	Bool	false
6	▼ Output		
7	motor.drch_Q_FB	Bool	false
8	motor.izq_Q_FB	Bool	false
9	motor.on_Q_FB	Bool	false
10	verde_Q_FB	Bool	false
11	nar.drch_Q_FB	Bool	false
12	nar.izq_Q_FB	Bool	false
13	rojo_Q_FB	Bool	false
14	InOut		
15	▼ Static		
16	estado_inicial	Bool	false
17	estado_marcha	Bool	false
18	estado_paro	Bool	false

ANEXO III. ESQUEMA ELÉCTRICO



Comunicación entre PLC S7-1200 y driver C5-E-2-09 mediante protocolo CANopen para el control de un motor de corriente continua. 133

ANEXO IV. FOTOGRAFÍAS



Comunicación entre PLC S7-1200 y driver C5-E-2-09 mediante protocolo CANopen para el control de un motor de corriente continua. 134



ANEXO V. HOJAS DE DATOS DE LOS COMPONENTES

Dentro de anexos, en la carpeta "Datasheets", se encuentran los archivos con información sobre los componentes utilizados en el proyecto.