



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial



Universidad
Politécnica
de Cartagena



EUROPEAN
UNIVERSITY OF
TECHNOLOGY
"Think human first"



UNIVERSIDAD POLITÉCNICA DE CARTAGENA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO FIN DE GRADO

CONTROL A DISTANCIA DE MÁQUINA DE
ENTRENAMIENTO CON EJE ELÉCTRICO Y
ENCODER.

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL.

ENRIQUE BAÑOS CAMPILLO.

*Directora:
NIEVES PAVÓN PULIDO.*

*Departamento:
Automática*

Cartagena, a 1 de marzo de 2023.



Universidad
Politécnica
de Cartagena



Universidad
Politécnica
de Cartagena



EUROPEAN
UNIVERSITY OF
TECHNOLOGY
"Think human first"



AGRADECIMIENTOS.

A mis padres, a mi hermana y mi tito Javi, a Luisa, a todas las personas que me han apoyado con mis estudios, a mis amigos, y a mi tutora del TFG.



RESUMEN

Este proyecto consiste en el diseño de un sistema de control capaz de modificar la tensión de una cuerda mediante el uso de un eje eléctrico. Dicho sistema se utilizará para una máquina de hacer ejercicio.

El sistema debe estar conectado con una aplicación móvil, donde los usuarios crearán sus entrenamientos a medida. El registro de los entrenamientos se almacenará en una base de datos para poder ver el historial cuando los usuarios lo deseen.

Necesitaremos seleccionar un canal de comunicaciones y definir qué variables se han de compartir entre la aplicación móvil y el controlador del eje eléctrico que regula la intensidad del ejercicio. Por último, seleccionaremos un sistema que sea capaz de medir la distancia recorrida por la cuerda y decidir la ubicación de dicho dispositivo dentro de la máquina.

En este documento se explicarán todas las conclusiones/soluciones obtenidas para la realización del proyecto.

ABSTRACT

This project consists of the design of a control system capable of modifying the tension of a rope through the use of an electric axis. Such a system will be used for an exercise machine.

The system must be connected with a mobile app, where users will create their bespoke workouts. The workout log will be stored in a database so that users can view the history whenever they want.

We will need to select a communication channel and define which variables have to be shared between the mobile application and the controller of the electric axis that regulates the intensity of the exercise. Finally, we will select a system that is capable of measuring the distance traveled by the rope and deciding the location of such a device within the machine.

This document will explain all the conclusions/solutions obtained for the realization of the project.



DOCUMENTO I: MEMORIA

ÍNDICE DE LA MEMORIA

1. Introducción.....	10
1.1. Introducción.....	10
1.2. Objetivos.....	11
1.3. Motivación.....	12
1.4. Fases del proyecto y planificación	12
1.4.1 DIAGRAMA DE GANTT	13
2. HARDWARE	14
2.1. Maqueta suministrada por el cliente.....	14
2.2. Controlador seleccionado.....	15
2.2.1 CARACTERÍSTICAS Y CONEXIONADO DEL CONTROLADOR.....	17
2.3. Fuente de alimentación	20
2.4. Eje eléctrico	20
2.4.1 CARACTERÍSTICAS Y CONEXIONADO DEL EJE ELÉCTRICO	21
2.5. RELÉS.....	23
2.6. Encoder.....	24
2.6.1 CARACTERÍSTICAS Y CONEXIONADO DEL ENCODER	24
2.7. Bornas.....	25
2.8. Router wifi Tp-Link CPE510.....	26
2.9. Conexionado del Hardware	28
3. SOFTWARE	30
3.1. Descripción de los elementos software utilizados.	30
3.1.1 TIA Portal	30
3.1.2 EPLAN 2.7.....	31
3.2. Descripción detallada del proceso con TIA Portal	32
3.2.1 Entradas y salidas del proceso.....	33
3.2.2 Escalado del eje eléctrico	37
3.2.3 Escalado del <i>encoder</i>	40
3.2.4 Control eje eléctrico.....	42
3.2.5 Control del <i>encoder</i>	43
3.2.6 Mecanismo de control.....	46
4. MODO DE OPERACIÓN DEL WIFI TP-LINK.....	53
5. COMUNICACIÓN APP - PLC (WEB SERVER).....	55



6. CONCLUSIONES.....	58
6.1 POSIBLES MEJORAS Y FUTURO DEL PROYECTO.....	58
7. BIBLIOGRAFÍA	59
8. ANEXOS.....	61
8.1 PROGRAMACIÓN PLC.....	62
8.1.1 Información general del proyecto	62
8.1.2 Tabla de variables estándar	63
8.1.3 Bloques de programa.....	64
8.2 PRESUPUESTO.....	82
8.3 CONFIGURACIÓN COMPLETA WIFI TP-LINK	83
8.4 CONFIGURACIÓN SERVIDOR WEB EN TIA PORTAL.....	86
8.5 PLANOS CON EPLAN 2.7.....	89
8.6 CONEXIONADO DEL HARDWARE EXTERNO	103

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Ejercicio que quiere automatizar el cliente.	10
Ilustración 2. Diagrama de Gantt del proyecto.	13
Ilustración 3. Maqueta entregada por el cliente.	14
Ilustración 4. Maqueta entregada por el cliente.	14
Ilustración 5. Raspberry.	15
Ilustración 6. Estructura de EtherCAT.	15
Ilustración 7. Periferia descentralizada de entradas y salidas Beckhoff.	16
Ilustración 8. PLC S7-1200 CPU1214C DC/DC/DC de SIEMENES.	16
Ilustración 9. Tarjeta entradas analógicas SM 1231.	17
Ilustración 10. Conexión interno CPU 1214C DC/DC/DC.	18
Ilustración 11. Conexión interna tarjeta entradas analógicas SM 1231.	19
Ilustración 12. Conexión de la fuente de alimentación.	20
Ilustración 13. Fuente de alimentación S8PS-05024CD.	20
Ilustración 14. Eje eléctrico de SMC.	20
Ilustración 15. Eje eléctrico Linak LA14.	21
Ilustración 16. Relé RSL1AB4BD.	23
Ilustración 17. Zócalo RSLZVA1.	23
Ilustración 18. Dimensiones zócalo RSLZVA1.	23
Ilustración 19. Encoder de Industrial Shields.	24
Ilustración 20. Ubicación encoder en el eje de la polea.	24
Ilustración 21. Conexión de los terminales del encoder.	24
Ilustración 22. Conexión del encoder con el exterior.	25
Ilustración 23. Resistencia Pull-Up de Industrial Shield para el encoder.	25
Ilustración 24. Borna de Phoenix Contact.	26
Ilustración 25. Wifi Tp-Link CPE510.	26
Ilustración 26. . Wifi Tp-Link CPE510.	27
Ilustración 27. Zonas del bornero.	28
Ilustración 28. Conexión de la fuente de alimentación.	28
Ilustración 29. Cambio de polaridad.	29
Ilustración 30. Activación de los relés.	29
Ilustración 31. Conector realizado para los encoders.	29
Ilustración 32. Ejemplo de diseño con EPLAN.	31
Ilustración 33. GRAFCET del proceso de variado de resistencia mediante el control de un eje eléctrico.	32
Ilustración 34. Valores normalizados.	37
Ilustración 35. Valores escalados.	38
Ilustración 36. Proceso de escalado de una señal analógica con TIA Portal.	38
Ilustración 37. Variables auxiliares del proceso de escalado.	39
Ilustración 38. . Function block resultante del proceso de escalado de una señal analógica.	39
Ilustración 39. Proceso de escalado de la señal enviada por los encoders.	40
Ilustración 40. Variables auxiliares del proceso de escalado del encoder.	40
Ilustración 41. Function block resultante del proceso de escalado del encoder.	41
Ilustración 42. Function block resultante del proceso de control del eje eléctrico.	42
Ilustración 43. Variables auxiliares del proceso.	42
Ilustración 44. . Elección del modo de funcionamiento de la instrucción HSC.	43
Ilustración 45. Function block resultante del proceso de control del encoder.	44
Ilustración 46. Controlador PID.	46



Ilustración 47. Diagrama de bloques PIDT1 con Anti-Windup	47
Ilustración 48. Ajustes básicos PIDT1 con Anti-Windup.....	48
Ilustración 49. Escalado de realimentación de posición.	48
Ilustración 50. Ajustes del valor real escalado.	49
Ilustración 51. DB resultante de la instrucción PID_3Step V2.....	49
Ilustración 52. Parámetros del PID.	51
Ilustración 53. Puesta en servicio PID_3STEP.....	51
Ilustración 54. Comportamiento del PID para la obtención de la salida deseada.	52
Ilustración 55. Conexión red TP-LINK_CFCA.....	53
Ilustración 56. Información del dispositivo router.	53
Ilustración 57. Modo de operación utilizado por el WIFI TP_LINK.....	54
Ilustración 58. . Escenario de configuración del protocolo OPC UA de un S7-1200.	55
Ilustración 59. Funcionamiento protocolo MQTT.	55
Ilustración 60. Instrucción del servidor web.	56

ÍNDICE DE TABLAS

Tabla 1. Datos técnicos CPU1214C DC/DC/DC.	17
Tabla 2. Asignación de pines de conectores para CPU1214C DC/DC/DC (6ES7 214-1AG31-0XB0)	18
Tabla 3. Datos técnicos tarjeta señales analógicas SM 1231.	19
Tabla 4. Especificaciones E/S s y conexionado del eje eléctrico Linak LA14.	22
Tabla 5. Datos técnicos del relé RSL1AB4BD.	23
Tabla 6. Variables entrada para la función de la comunicación PLC/app móvil.....	35
Tabla 7. Variables salida para la función de la comunicación PLC/app móvil.	37
Tabla 8. Magnitudes obtenidas durante el proceso de escalado del eje eléctrico.	40
Tabla 9. Magnitudes obtenidas durante el proceso de escalado del encoder.	41
Tabla 10. Tipo de datos de los parámetros utilizados en el control del encoder.	44
Tabla 11. . Códigos de error del parámetro STATUS.	45
Tabla 12. Códigos de error de la instrucción del servidor web.	57

1. Introducción

Al realizar las prácticas curriculares en la empresa de automatización industrial Binarii, se presentó la oportunidad de conocer a un cliente (LECY SPORT) interesado en crear una máquina de entrenamiento con unas características específicas. Tras el arduo desarrollo de la máquina de entrenamiento, el cual desarrollé completamente paso a paso y con los aciertos y errores cometidos, convertí el encargo de LECY SPORT en este Trabajo Final de Grado.

Posteriormente, se añadió un mecanismo de control para aumentar la precisión del proceso y se investigó sobre posibles protocolos de comunicación entre una aplicación móvil y un autómatas programable.

1.1. Introducción.

Se quiere automatizar un ejercicio de entrenamiento común que suelen hacer los futbolistas avanzando y retrocediendo enganchados a una cuerda elástica siguiendo una trayectoria definida (ver ilustración 1). Con estos movimientos, se conseguirá variar la intensidad a medida que el usuario avance o retroceda. Para automatizar este proceso, se requerirá una máquina que logre este propósito.



Ilustración 1. Ejercicio que quiere automatizar el cliente.

El cliente solicita el diseño de un sistema que sea capaz de modificar la resistencia de una cuerda y conseguir medir la distancia recorrida durante el entrenamiento. Para conseguir esto, él nos proporciona el prototipo mecánico de la máquina.

La idea es usar un eje eléctrico que consiga aumentar o disminuir la resistencia del ejercicio a medida que se aproxime a un imán ubicado en el prototipo obtenido. Para medir la distancia se utilizará un *encoder* situado en una de las poleas de la maqueta. Para realizar el control de los sensores y actuadores de la máquina de ejercicio hay que seleccionar un Controlador Lógico óptimo para el desarrollo del proyecto.

Se solicitó que el número de usuarios de la máquina simultáneamente fueran cuatro. Por lo tanto, se necesitarán cuatro *encoder* y cuatro ejes eléctricos para sus correspondientes prototipos.

Se necesitará seleccionar un entorno de desarrollo que nos permita realizar el proyecto adecuadamente. Se optó por utilizar el *software* TIA PORTAL, debido a los conocimientos adquiridos durante la formación universitaria. Además, incluye numerosas funciones de programación que nos facilitarán el trabajo más adelante, en especial, la función de control mediante un controlador PID.

Por último, se necesitará establecer comunicación con una aplicación que permita llevar el control de la máquina desde nuestro propio dispositivo móvil; para ello, se requerirá también una aplicación móvil donde se puedan crear entrenamientos con el nivel de intensidad determinada por el usuario.

1.2. Objetivos

En este apartado se describirán los objetivos que se han marcado para la realización del TFG:

- RECOGIDA DE DATOS Y PLANIFICACIÓN:

1. Definición de las especificaciones de funcionamiento de la máquina: el cliente especifica las características deseadas de la máquina para conocer y comprender lo más preciso posible el funcionamiento de la máquina, también establecerá un tiempo determinado para la realización del proyecto.

2. Definición del protocolo de comunicaciones: se necesita diseñar un protocolo de comunicaciones para recibir la información de la máquina en la aplicación móvil.

3. Selección del hardware del sistema: se tiene que escoger un *hardware* que haga posible el diseño requerido, después se tendrán que conectar todos los componentes.

- DESARROLLO:

4. Desarrollo del programa del PLC para realizar los procesos de escalado y de control del eje eléctrico y el *encoder*: se utilizará el *software* TIA PORTAL para realizar los procesos de control y escalado del eje eléctrico y del *encoder*.

5. Diseño y ajuste del controlador PID: Se deberá diseñar un controlador PID que optimice el funcionamiento de la máquina de entrenamiento. Este controlador PID regulará el movimiento del eje eléctrico.

6. Creación de tipo de datos para la comunicación: se comunicará todo el proyecto con una app móvil, para ello se tiene que crear un tipo de datos para la comunicación.

7. Conexión físico del PLC con el eje eléctrico y el *encoder*: se tendrá que alimentar y conectar todos los componentes externos, más tarde, algunos se quedarán dentro del cuadro eléctrico y otros en el exterior.

- PRUEBAS:

8. Verificación del control y funcionamiento del eje eléctrico y el *encoder*: se probará el correcto funcionamiento de todos los componentes que se integran en el proyecto, también se tendrá que configurar el control del eje eléctrico y del *encoder*.

9. Configuración de equipos para la comunicación con aplicación móvil a través del punto de acceso y adquisición de los datos correctamente: hay que establecer la comunicación con la aplicación móvil, para ello se escogerá un punto de acceso y después se comprobará que los datos se transmiten adecuadamente.

10. Prueba de la aplicación móvil: se harán una serie de pruebas finales desde la aplicación, donde se se podrán crear diferentes entrenamientos personalizados.



1.3. Motivación

La automatización industrial es una rama de la ingeniería que se encuentra en continuo avance y desarrollo. En muchas industrias, o en cualquier otra aplicación, podemos ver PLC (*Programmable Logic Controller*) utilizados para la automatización y el control de procesos. También nos permiten visualizar los procesos mediante pantallas HMI (*Human-Machine Interface*), para poder visualizar en cualquier momento nuestro proceso automatizado, será muy útil si se produce algún problema.

La motivación principal para la realización de este proyecto es adquirir aprendizaje a través de establecer la comunicación entre un proceso de automatización industrial y una aplicación móvil. Este aprendizaje me permitirá posteriormente desarrollar otros proyectos similares o ser objeto de punto de partida para otros diferentes. Otro estímulo para efectuar este trabajo es aplicar en un caso real los conocimientos adquiridos sobre regulación de sistemas automáticos mediante controlador PID, su diseño, configuración y puesta en servicio con el *software* TIA PORTAL.

1.4. Fases del proyecto y planificación

El proyecto constará de 16 fases para su completo desarrollo. Estas fases son las siguientes:

- PLANIFICACIÓN:
 1. Definir las especificaciones de funcionamiento de la máquina.
 2. Seleccionar un *hardware apropiado* para el sistema.
 3. Definir el protocolo de comunicaciones entre la aplicación móvil y el autómata programable.
 4. Adjuntar todos los datos recogidos en los apartados 1, 2, 3 y obtener el primer prototipo completo, el cual será orientativo para la realización del proyecto.

- DESARROLLO DEL PROYECTO:
 5. Conexionar todo el *hardware* del sistema.
 6. Desarrollar un programa específico en el *software* TIA PORTAL para realizar las funciones de escalado y control del eje eléctrico y *encoder*.
 7. Ajustar el controlador PID que regula el eje eléctrico.
 8. Crear un tipo de datos apropiado para la comunicación entre el PLC y la aplicación móvil.
 9. Desarrollar el plano eléctrico del proyecto mediante el programa EPLAN.

- PRUEBAS DEL FUNCIONAMIENTO:
 10. Probar el control del sistema sobre el *encoder* y el eje eléctrico.
 11. Gestionar errores y formas de mejora.
 12. Comprobar si el programa está realizando el escalado de la señal analógica correctamente.
 13. Configurar los equipos utilizados en el proceso para su comunicación con aplicación móvil a través del punto de acceso y adquirir los datos correctamente.
 14. Probar la aplicación móvil con los distintos tipos de entrenamiento disponibles.
 15. Gestionar errores y formas de mejora futuras.
 16. Documentar el proyecto.

1.4.1 DIAGRAMA DE GANTT

A continuación, se mostrará el diagrama de Gantt (ver ilustración 2) del proyecto, donde podemos visualizar todas las fases del proyecto con su duración determinada, mencionadas en el punto anterior:



2. HARDWARE

En este apartado se expondrá el Hardware utilizado para la realización del proyecto, se explicará la selección de cada componente y sus datos más relevantes.

2.1. Maqueta suministrada por el cliente

Al comienzo del proyecto, el cliente nos suministró una maqueta mecánica (ver ilustraciones 3 y 4), sin ningún componente eléctrico, ni ningún tipo de conexión. Este diseño está compuesto por cuatro poleas, una de ellas está recubierta por un imán. El objetivo de este diseño se basa en que una vez que instalemos el eje eléctrico, cuando lo activemos y se mueva, se aproxima a la polea que contiene el imán y produce una variación de la resistencia que ofrece la cuerda. Con este diseño se ajusta la resistencia de la cuerda mediante el movimiento controlado de un eje eléctrico.

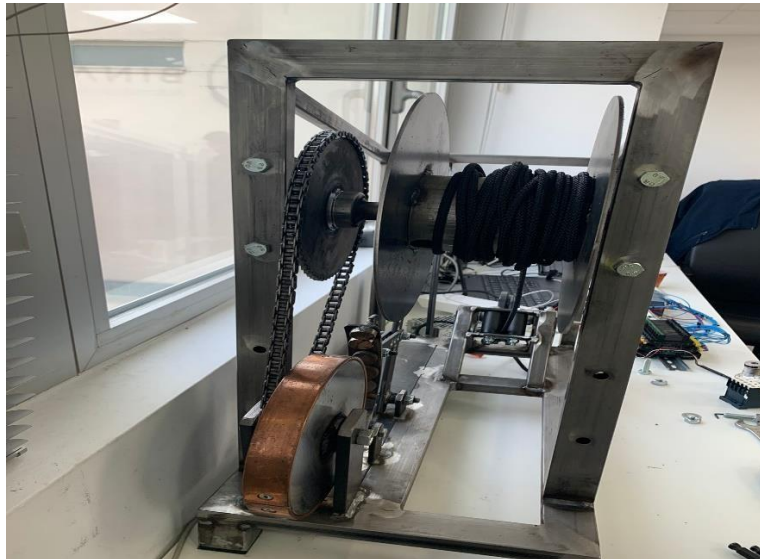


Ilustración 3. Maqueta entregada por el cliente.



Ilustración 4. Maqueta entregada por el cliente.

Para regular la fuerza que aplica el imán y, por consecuencia, la resistencia que tendrá la cuerda, se puede cambiar el tipo de imán, su tamaño y/o el diseño mecánico entregado por el cliente, entre otros.

En este proyecto el diseño mecánico entregado inicialmente no tendrá importancia, ya que la resistencia que haga la cuerda la modificará el cliente final, nuestro trabajo es el de conseguir esa variación de resistencia, sin importar el valor que varía la resistencia exactamente.

2.2. Controlador seleccionado

En un primer lugar se pensó en usar una *Single Board Computer* (tipo *Raspberry*) como controlador del eje eléctrico (ver ilustración 5). Al ser un miniordenador, con su sistema operativo, ofrece todas las ventajas propias de un PC y facilita el uso de multitud de lenguajes de programación, buses y protocolos de comunicación. Además de otras ventajas, como su pequeño tamaño, precio bajo y su sencilla utilización.



Ilustración 5. *Raspberry*.

En conjunto con la *Raspberry* se pretendía usar una cabecera distribuida de entradas y salidas basada en el bus decampo *EtherCAT* (ver ilustración 6) para controlar el eje eléctrico y el *encoder*.

EtherCAT es un protocolo de código abierto que utiliza una asignación *Ether-type* dentro del estándar *ethernet*. El uso de la red *ethernet* en buses de campo permite reducir los costes e interfaces, así como acceder a la tecnología de internet, pero con el contrapunto de no ser determinista.

Estructura de EtherCAT

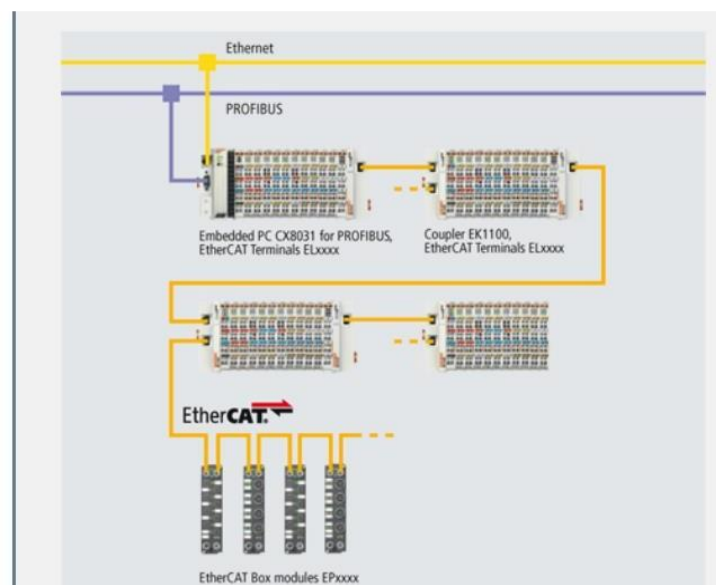


Ilustración 6. Estructura de *EtherCAT*.

La utilización de la Raspberry con el protocolo *EtherCAT* es posible, ya que esta cabecera está provista de entradas y salidas (ver ilustración 7), tanto digitales como analógicas.

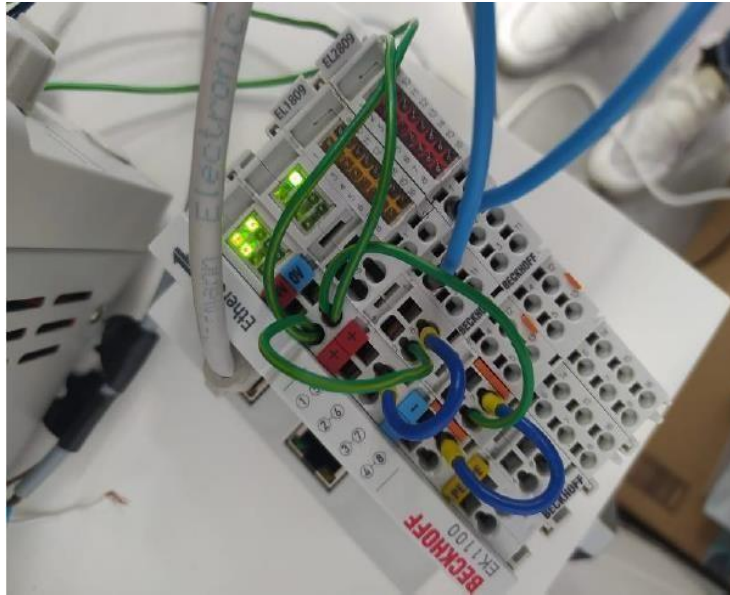


Ilustración 7. Periferia descentralizada de entradas y salidas Beckhoff.

Aunque el uso de la *Raspberry* presentaba muchas ventajas, se desechó su uso debido a la poca robustez que ofrece. En nuestro proyecto, la máquina de ejercicio se encuentra a la intemperie de un campo de fútbol, por lo tanto, las condiciones que experimentaría el controlador no serían las óptimas para su funcionamiento y podría estropearse.

Finalmente, el controlador seleccionado fue un S7-1200 con su CPU1214C DC/DC/DC de la marca Siemens (ver ilustración 8). Este autómatas ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. Además, tiene las entradas y salidas suficientes para controlar los sensores y actuadores de la máquina de ejercicio.

Los PLCs (*Programmable Logic Controller*) son los dispositivos más usados en la industria debido a su robustez y alta fiabilidad, están preparados para trabajar de forma ininterrumpida 24 horas al día, toda la semana. Son capaces de resistir vibraciones y electromagnetismos, estando testados por la TUV (*Technischer Überwachungs Verein*). Esta robustez es la característica principal por la que hemos decidido utilizar este controlador, debido a que la máquina de entrenamiento se situará en un campo de fútbol.



Ilustración 8. PLC S7-1200 CPU1214C DC/DC/DC de SIEMENES.

Este controlador tiene un inconveniente, solo posee entradas y salidas digitales. Para realizar el proyecto, se necesitan cuatro entradas analógicas para controlar el funcionamiento de sus respectivos ejes eléctricos utilizados en el proceso. Para solucionar este problema, se tuvo que añadir un módulo de señales analógicas SM 1231 AI4 x 13 bits (ver ilustración 9). Este módulo de señales analógicas mide el valor de tensión conectada a las entradas del módulo. El valor máximo del rango nominal serán 27648 decimales.



Ilustración 9. Tarjeta entradas analógicas SM 1231.

2.2.1 CARACTERÍSTICAS Y CONEXIONADO DEL CONTROLADOR

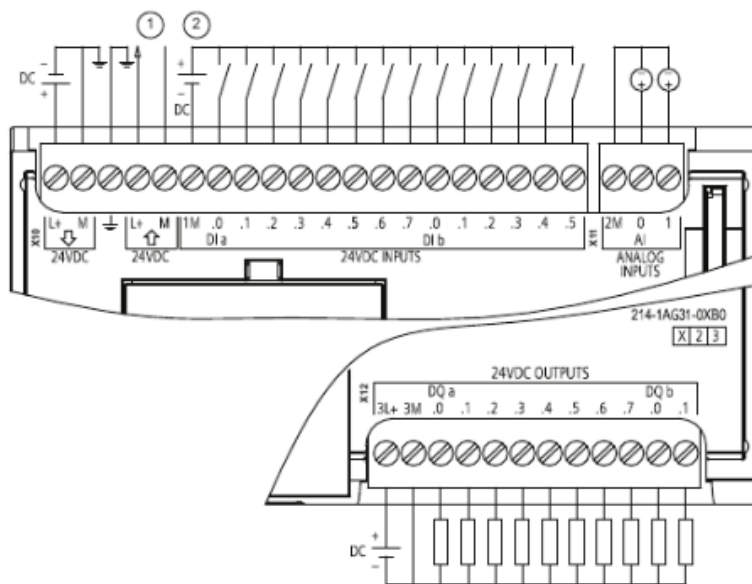
En este apartado, se presentan los datos técnicos (ver tabla 1) y el conexionado de la CPU/12140/DC/DC obtenidos del manual de sistema SIMATIC S7-1200 (SIEMENS, 2009). Las características más destacadas del controlador son las siguientes:

- CPU1214C DC/DC/DC.
- CPU compacta.
- E/S Integradas: 14 DI 24V DC; 10 DO 24V DC; 0-10V DC.
- Alimentación: DC 20,4-28,8 V DC.
- Memoria de programa datos de 100KB.

Datos técnicos	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Referencia	6ES7 214-1BG31-0XB0	6ES7 214-1HG31-0XB0	6ES7 214-1AG31-0XB0
Dimensiones A x A x P (mm)	110 x 100 x 75	110 x 100 x 75	110 x 100 x 75
Peso de envío	475 gramos	435 gramos	415 gramos
Disipación de potencia	14 W	12 W	12 W
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)	1600 mA máx. (5 V DC)	1600 mA máx. (5 V DC)
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)	400 mA máx. (alimentación de sensores)	400 mA máx. (alimentación de sensores)
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada	4 mA/entrada utilizada	4 mA/entrada utilizada

Tabla 1. Datos técnicos CPU1214C DC/DC/DC.

A continuación, se muestra el conexionado interno del controlador (ver ilustración 10) y la asignación de pines de conectores para esta CPU (ver tabla 2):



- ① Alimentación de sensores 24 V DC

Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
- ② Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

Nota: Los conectores X11 deben ser de oro. Consulte el anexo C, Piezas de repuesto, para ver la referencia.

Ilustración 10. Conexionado interno CPU 1214C DC/DC/DC.

Pin	X10	X11 (oro)	X12
1	L+/24 V DC	2 M	3L+
2	M/24 V DC	AI 0	3M
3	Tierra funcional	AI 1	DQ a.0
4	Salida sensor L+/24 V DC	--	DQ a.1
5	Salida sensor M/24 V DC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6

Tabla 2. Asignación de pines de conectores para CPU1214C DC/DC/DC (6ES7 214-1AG31-0XB0)

Por último, se muestran los datos técnicos más relevantes (ver tabla 3) y el conexionado interno (ver ilustración 11) del módulo de señales analógicas SM 1231 AI4 x 13 extraídos de la hoja de características del fabricante (AG, datasheet SM1231, 2024):

Información general	
Designación del tipo de producto	SM 1231, AI 4x13 bit
Tensión de alimentación	
Valor nominal (DC)	24 V
Intensidad de entrada	
Consumo, típ.	45 mA
de bus de fondo 5 V DC, típ.	80 mA
Pérdidas	
Pérdidas, típ.	1,5 W

Entradas analógicas	
Nº de entradas analógicas	4; Entradas diferenciales tipo corriente o tensión
Tensión de entrada admisible para entrada de tensión (límite de destrucción), máx.	35 V
Intensidad de entrada admisible para entrada de corriente (límite de destrucción), máx.	40 mA
Tiempo de ciclo (todos los canales), máx.	625 μ s
Rangos de entrada	
• Tensión	Sí; ± 10 V, ± 5 V, $\pm 2,5$ V
• Intensidad	Sí; 4 a 20 mA, 0 a 20 mA
• Termopar	No
• Termorresistencias	No
• Resistencia	No
Rangos de entrada (valores nominales), tensiones	
• -10 V a +10 V — Resistencia de entrada (-10 V a +10 V)	Sí ≥ 9 MOhm
• -2,5 V a +2,5 V — Resistencia de entrada (-2,5 V a +2,5 V)	Sí ≥ 9 MOhm
• -5 V a +5 V — Resistencia de entrada (-5 V a +5 V)	Sí ≥ 9 MOhm
Rangos de entrada (valores nominales), intensidades	
• 0 a 20 mA — Resistencia de entrada (0 a 20 mA)	Sí 280 Ω
• 4 mA a 20 mA — Resistencia de entrada (4 mA a 20 mA)	Sí 280 Ω
Formación de valor analógico para entradas	
Tiempo de integración y conversión/resolución por canal	
• Resolución con rango de rebase (bits incl. signo), máx.	12 bit; + signo
• Tiempo de integración parametrizable	Sí
• Supresión de perturbaciones de tensión para frecuencia perturbadora f1 en Hz	40 dB, DC a 60 V para frecuencia de perturbación 50/60 Hz
Dimensiones	
Ancho	45 mm
Altura	100 mm
Profundidad	75 mm

Tabla 3. Datos técnicos tarjeta señales analógicas SM 1231.

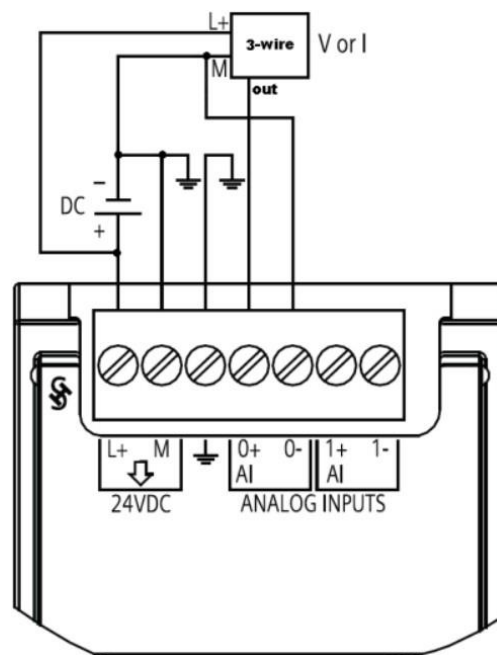


Ilustración 11. Conexión interna tarjeta entradas analógicas SM 1231.

2.3. Fuente de alimentación

La fuente de alimentación seleccionada es una S8PS-05024CD de OMRON (ver ilustración 13). Esta fuente es muy compacta y puede montarse sobre carril DIN, además cuenta con un sencillo conexionado (ver ilustración 12). Sus características más importantes son las siguientes (OMRON):

- Potencia de 50W.
- Configuración con carcasa con soporte de montaje en carril DIN.
- Entrada universal de c.a: 100 a 240 V c.a.
- Cumple las normas de CEM: EN50081-1, EN50082-2 y EN61000-3-2.
- 10 años de vida útil, como mínimo.

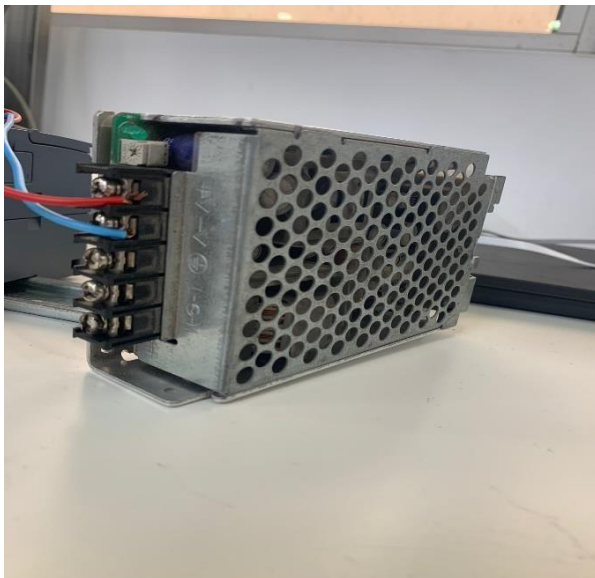


Ilustración 13. Fuente de alimentación S8PS-05024CD.

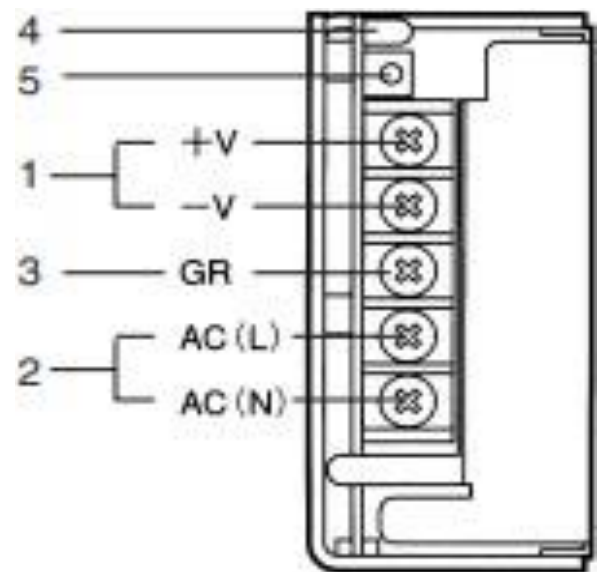


Ilustración 12. Conexionado de la fuente de alimentación.

2.4. Eje eléctrico

En primera instancia, se valoró usar un eje eléctrico de la marca SMC (ver ilustración 14), capaz de ser controlado mediante el bus de campo *EtherCAT*, mencionado anteriormente. Tiene un diseño compacto que permite ahorrar espacio de instalación, múltiples opciones de control de velocidad y posición, y rápida puesta en marcha. Al igual que la *Raspberry* presentaba numerosos puntos a favor, pero presentaba inconvenientes que lo convertían en una opción equívoca. El primer inconveniente es su elevado coste, y el más importante es la poca robustez que ofrece.



Ilustración 14. Eje eléctrico de SMC.

Finalmente, el eje eléctrico seleccionado fue de la marca Linak, el modelo LA14 (ver ilustración 15). Este eje puede ser controlado por señales digitales, en lugar de señales por comunicaciones a través de algún bus de campo. Es un actuador muy resistente con un alto grado de protección IP y alojamiento de aluminio que lo convierten en la unidad perfecta para entornos difíciles y exigentes, es decir, ideal para el entorno donde se alojará nuestro proyecto.

El LA14 garantiza un rendimiento fiable a temperaturas entre $-40\text{ }^{\circ}\text{C}$ y $+85\text{ }^{\circ}\text{C}$, y está bien preparado para trabajar en aplicaciones que requieren movimientos lineales cortos.



Ilustración 15. Eje eléctrico Linak LA14.

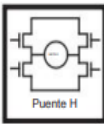
Este eje proporciona la posición en la que se encuentra mediante la señal analógica 0-10V. Con esta señal analógica podemos posicionar el eje en una posición determinada. Con ello conseguimos la posibilidad de indicar con precisión la resistencia de la cuerda que deseemos.

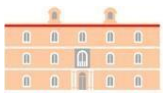
2.4.1 CARACTERÍSTICAS Y CONEXIONADO DEL EJE ELÉCTRICO

Para comenzar, se citan las características más relevantes del eje eléctrico Linak LA14:

- Motor magnético permanente 12 a 24 V DC.
- Tubo interior: Acero inoxidable AISI304/SS2333
- Rango de temperatura de funcionamiento: $[-40^{\circ}, 85]$ $^{\circ}\text{C}$

A continuación, se muestra una tabla con las especificaciones E/S y el conexionado del actuador Basic (LINAK) (ver tabla 4). Los terminales verde y amarillo del eje no deberán estar conectados. Se realiza la retroalimentación analógica (0-10) V mediante el terminal violeta. Esa señal analógica modifica la resistencia que ofrece la cuerda en la máquina de entrenamiento:

De entrada y salida	Especificación	Comentarios
Descripción	<p>Interfaz fácil de usar con electrónica de potencia integrada (puente H).</p> <p>El actuador también puede estar equipado con un circuito electrónico que proporcione una señal de realimentación absoluta o relativa.</p> <p>La versión con "opción IC" no puede funcionar con PWM (fuente de alimentación).</p>	 <p>Puente H</p>



Marrón	<p>12-24 VCC + (VCC)</p> <p>Conecte Brown a positivo</p> <p>12V ± 20%</p> <p>24V ± 10%</p> <p>Motor estándar motor rapido</p> <p>12V, límite de corriente 8A 12V, límite de corriente 8A</p> <p>24 V, límite de corriente 5 A. 24 V, límite de corriente 5 A.</p>	<p>Nota: ¡No cambie la polaridad de la fuente de alimentación en los cables marrón y azul!</p> <p>La fuente de alimentación GND (-) está conectada eléctricamente a la carcasa.</p> <p>Si la temperatura cae por debajo de -10°C, todos los límites actuales aumentarán automáticamente a 9A</p>
Azul	<p>12-24 VCC - (TIERRA)</p> <p>Conecte el azul al negativo</p> <p>12V ± 20%</p> <p>24V ± 10%</p> <p>Motor estándar motor rapido</p> <p>12V, límite de corriente 8A 12V, límite de corriente 8A</p> <p>24 V, límite de corriente 5 A 24 V, límite de corriente 5 A</p>	
Rojo	<p>Extiende el actuador</p>	<p>Tensiones de encendido/apagado:</p> <p>> 67% del VIN = ENCENDIDO</p> <p>< 33% del VIN = APAGADO</p> <p>Corriente de entrada: 10 mA</p>
Negro	<p>Retrae el actuador</p>	
Verde	<p>No estar conectado</p>	
Amarillo	<p>No estar conectado</p>	
Violeta	<p>Retroalimentación analógica</p> <p>0-10V (Opción A)</p>	<p>Consumo de energía en espera:</p> <p>12V, 60mA</p> <p>24 V, 45 mA</p> <p>Ondulación máx. 200mV</p> <p>Retraso de transacción 20 ms</p> <p>Retroalimentación lineal 0,5%</p> <p>Máx. salida de corriente: 1mA</p> <p>Es recomendable hacer que el actuador active sus finales de carrera periódicamente, para asegurar un posicionamiento más preciso.</p>
	<p>Salida de sala única (PNP)</p> <p>Movimiento por pulso Hall único:</p> <p>Actuador LA14020 = 0,2 mm por pulso</p> <p>Actuador LA14040 = 0,4 mm por pulso</p> <p>Frecuencia:</p> <p>La frecuencia es de 14-26 Hz en salida Single Hall dependiendo de la carga.</p> <p>Cada pulso está "ENCENDIDO" durante un mínimo de 3 ms.</p> <p>La sobretensión en el motor puede provocar pulsos más cortos.</p>	<p>Tensión de salida mín. Número de bastidor - 2V</p> <p>Máx. salida de corriente: 12mA</p> <p>Máx. 680nF</p>
Blanco	<p>Señal GND</p>	

Tabla 4. Especificaciones E/S s y conexionado del eje eléctrico Linak LA14.

2.5. RELÉS

Se utilizaron en nuestro prototipo 16 relés para activar los cuatro ejes eléctricos. Estos se utilizarán para que hagan de intermediario entre la CPU y los ejes eléctricos. Una vez que se activen los ejes desde la CPU, se activarán los relés y próximamente los ejes actuarán. Con esto, se protegen los ejes eléctricos ya que no estarán en contacto directo con la CPU. En nuestro proyecto se utilizaron unos relés de la marca Schneider Electric, modelo RSL1AB4BD (ver tabla 5 e ilustración 16) (Electric, 2014).

Tipo de relé	electromagnético
Configuración de contactos	SPDT
Tensión nominal de inductor	24V CC
Capacidad de carga de contactos CA@R (a carga resistiva)	6A / 250V CA
Capacidad de carga de contactos CC@R (a carga resistiva)	6A / 250V CC
Corriente máx. de contactos	6A
Tensión conectada	max. 277V CA, max. 277V CC
Versión de relé	miniatura
Montaje	PCB, soporte
Temperatura de trabajo	-40...70°C
Serie de relés	RSL1
Material de contactor	AgSnO ₂



Ilustración 16. Relé RSL1AB4BD.

Tabla 5. Datos técnicos del relé RSL1AB4BD.

Estos relés se sitúan en unos zócalos RSLZVA1 (ver ilustración 17) (electric, 2021), de la misma marca que los relés, que son compatibles con ellos debido a sus dimensiones (ver ilustración 18) . Los zócalos facilitan el cableado del relé y, por otro lado, proporcionan una interfaz física resistente con la instalación.



Ilustración 17. Zócalo RSLZVA1.

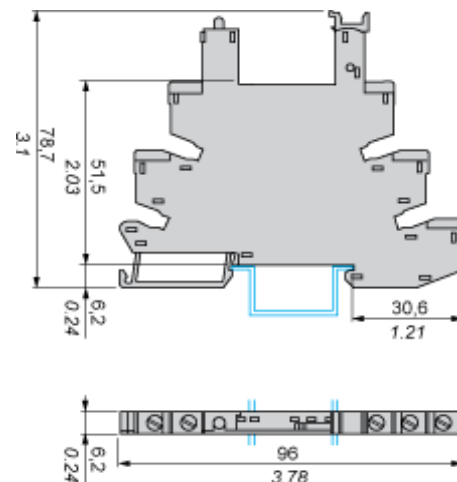


Ilustración 18. Dimensiones zócalo RSLZVA1.

2.6. Encoder.

Para medir la distancia recorrida se utilizó un *encoder* de la marca Industrial Shields, Codificador rotatorio incremental AB bifásico (ver ilustración 19) (shields), que cuenta los metros recorridos en cada entrenamiento. El *encoder* lo hemos situado en el eje de la polea exterior, es decir, la más cercana a la punta de la cuerda (ver ilustración 20). Para encajarlo en el eje de la polea hemos tenido que comprar un acoplamiento flexible de la marca R + W.

Este *encoder* convierte el movimiento angular de un eje rotatorio en un código analógico, emitiendo pulsos que representa ese movimiento o posición.



Ilustración 19. Encoder de Industrial Shields.

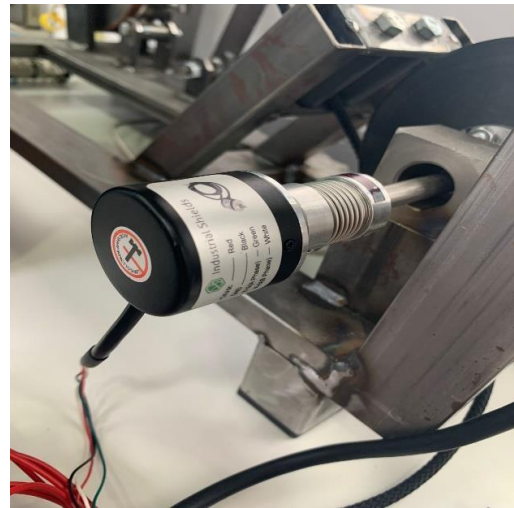


Ilustración 20. Ubicación encoder en el eje de la polea.

2.6.1 CARACTERÍSTICAS Y CONEXIONADO DEL ENCODER

A continuación, se muestran las características más relevantes del encoder de Industrial Shield:

- Fuente de alimentación de 9V-24V DC, fase constante bipolar del puente H.
- El máximo de 4.0A de los ocho tipos de corriente de salida.
- Función de bloqueo de corriente semiautomática de bajo consumo, protección de temperatura integrada y protección contra sobre corriente.
- Puede controlar el motor paso a paso de 2 fases hacia adelante y hacia atrás, con modo de excitación de fase1-2, W1-2, 2W1 -2, 4W1-2

Por último, se especifican las E/S del encoder (ver ilustración 21) y el conexionado final del dispositivo con el exterior (ver ilustración 22):

Encoder - Ethernet PLC Connections:

+24Vdc	-->	Red
Gnd	-->	Black
I0.5 (A Phase)	-->	Green
I0.6 (B Phase)	-->	Blanco

Ilustración 21. Conexionado de los terminales del encoder.

Conexión final:

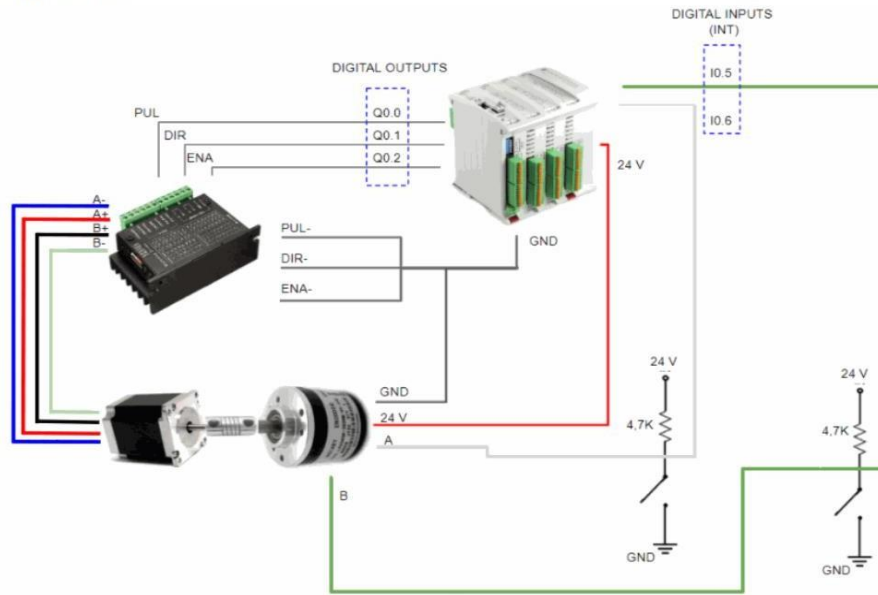


Ilustración 22. Conexión del encoder con el exterior.

Debido al voltaje tan pequeño que necesita el *encoder* para transmitir un pulso, a veces se producen fluctuaciones y no consigue llegar al voltaje interno necesario para enviar los pulsos. Al ser una entrada digital, el *encoder* no sabe si enviar 0 o 1 al PLC. Por todo esto, hemos utilizado bornas de Industrial Shields que contienen resistencias Pull-Up para estabilizar la tensión que llega al *encoder* (ver ilustración 23), de esta manera ya no se producirán estos inconvenientes.

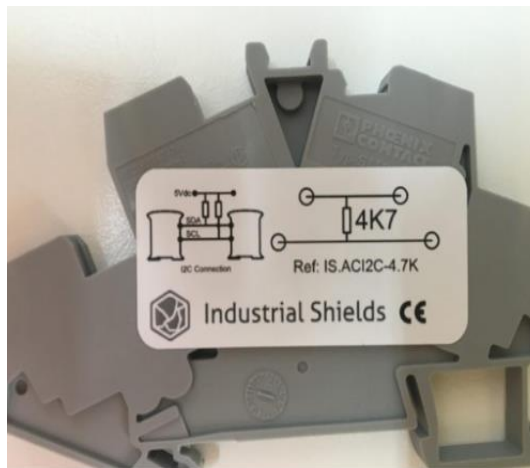


Ilustración 23. Resistencia Pull-Up de Industrial Shield para el encoder.

2.7. Bornas

Se utilizó en este prototipo 20 bornas de la marca Phoenix Contact (ver ilustración 24) más las 8 bornas de Industrial Shields (las bornas donde se conectan los *encoders*) comentadas anteriormente. Las bornas de conexión proporcionan seguridad a una instalación eléctrica y van a facilitar mucho el montaje final del prototipo. Da una visión más modular y estructurada del *hardware*.

Son bornas de doble piso, tensión nominal: 500V, corriente nominal: 24^a, corriente nominal: 24 A, tipo de conexión: conexión por tornillo y sección de dimensionamiento: 2,5 mm² (Contact, UTTB 2,5 - Borna de doble piso, 2024) .

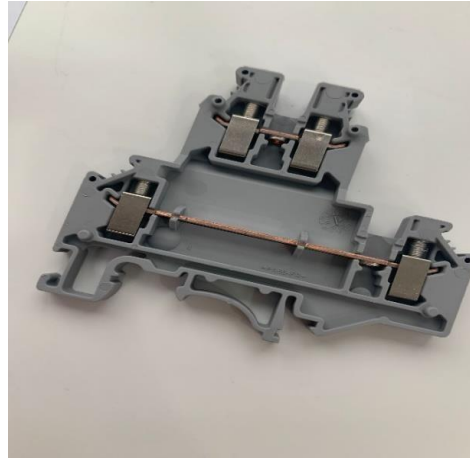


Ilustración 24. Borna de Phoenix Contact.

2.8. Router wifi Tp-Link CPE510

Para realizar la comunicación entre el PLC y la aplicación móvil, se optó por utilizar un *router* con comunicación vía wifi, para que cualquier dispositivo que tuviera instalada la aplicación móvil del ejercicio pudiera conectarse a la red de forma inalámbrica y recibir una dirección IP de forma automática mediante el protocolo de red DHCP.

Para realizar esta función, se escogió el *router* wifi Tp-Link CPE510 (ver ilustración 25) (TP-LINK). El CPE exterior de 5GHz 300Mbps 13 dBi de TP-Link está dedicado a soluciones rentables para aplicaciones de redes inalámbricas al aire libre. Puede transmitir datos inalámbricamente a más de 15 km. Se necesitaba un *router* con gran alcance debido a que tiene que llegar wifi a todo el campo de fútbol. También cuenta con la ventaja de que es bastante resistente, ya que está pensado para conectarse en exteriores.

A continuación, se mostrarán los datos técnicos más destacados:

- Polarización Dual Integrada de 13 dBi Antena Direccional.
- Modo de operación: Punto de Acceso / Cliente Router AP / Router AP Cliente (WISP).
- Protección contra Rayos de hasta 6000 V.
- Resistente a la Intemperie.
- Protección ESD de 15 Kv.
- Adaptador Poe pasivo



Ilustración 25. Wifi Tp-Link CPE510.

Este *router* necesita una alimentación de 24 V para funcionar. En su caja, se encontraba el *router* y un transformador cuya función es convertir los 230 V monofásicos a 24 V. No se ha utilizado dicho transformador porque, en el caso de que se rompiera, sería más complicado conseguirlo, la solución que se ha tomado ha sido la de utilizar un conector POE (muy barato), este conector permite alimentar el *router*, con 24 V, con el mismo cable de red (ver ilustración 26). De esta manera, si el cliente quisiera prescindir de la fuente de alimentación, podría alimentar todo este proyecto con unas baterías de 24 V, con esto no necesitaría ninguna toma de corriente.

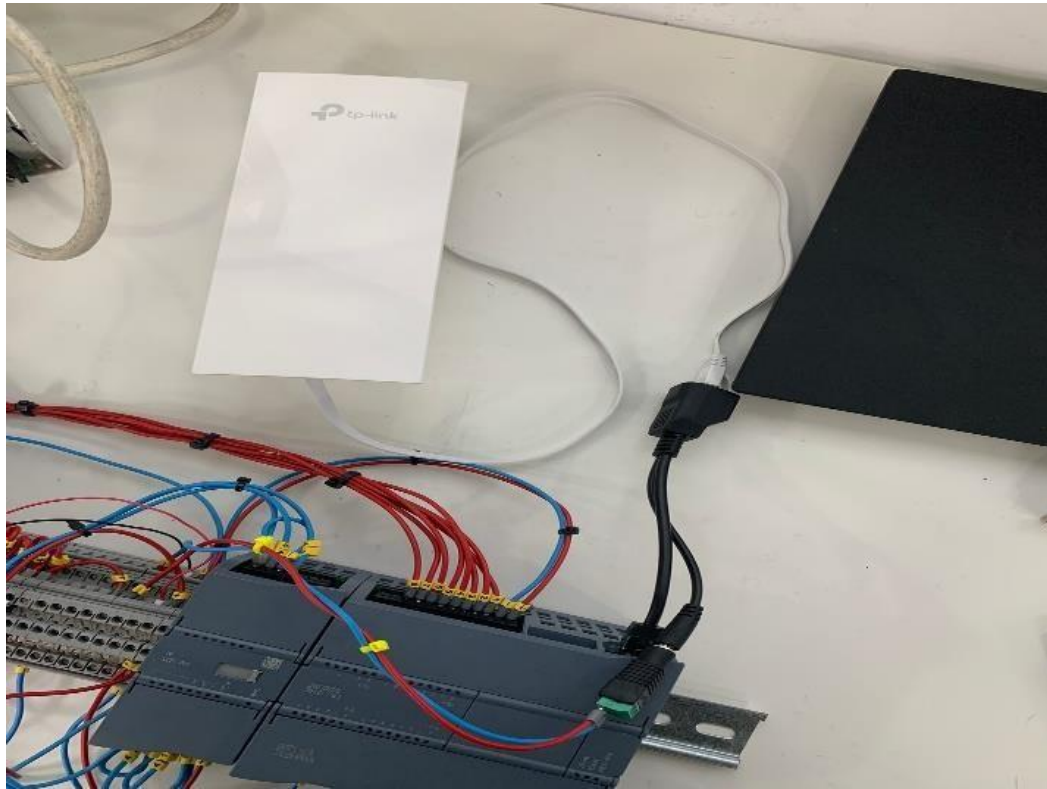


Ilustración 26. . Wifi Tp-Link CPE510.

2.9. Conexión del Hardware

Para comenzar con el conexionado, se ha creado un bornero con 28 bornas para que haga de interfaz entre los diferentes componentes mencionados en los apartados anteriores. Este bornero se situará en carril DIN, junto con el PLC y los relés. Todos estos componentes se encontrarán dentro del cuadro eléctrico.

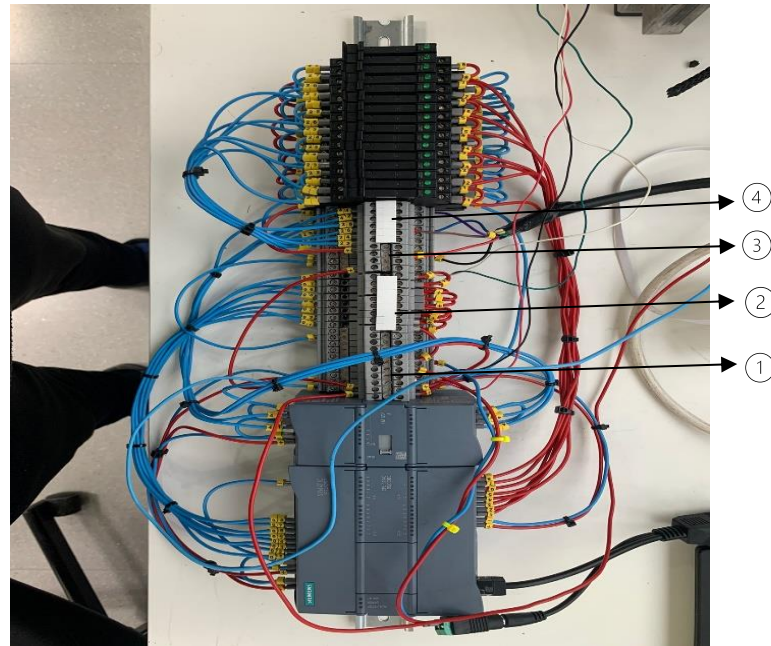


Ilustración 27. Zonas del bornero.

Como se puede observar, el bornero tiene 4 zonas bien diferenciadas (ver ilustración 27). En la segunda zona, se conectarán las fases A y B de los *encoders*, mientras que en la zona 4, se conectan los ejes eléctricos. De cada eje eléctrico se tendrán que conectar los terminales que provocan la función de avance y retroceso del eje, y la analógica. Por último, las zonas 1 y 3 están compuestas por positivos y negativos, que permitirán alimentar todos los dispositivos utilizados. Cabe destacar que las zonas 1 y 3 están puenteadas, es decir, solo hay positivos y negativos.

Con respecto a la fuente de alimentación, es el dispositivo que alimenta todos los componentes (ver ilustración 28). Ella misma es alimentada, en principio, con 230 V monofásicos.

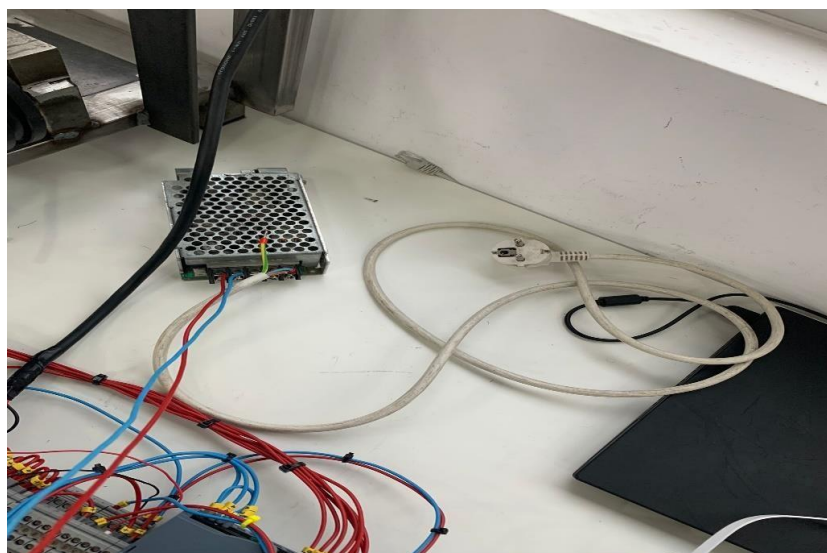


Ilustración 28. Conexión de la fuente de alimentación.

En los apartados anteriores, se ha explicado el conexionado del actuador y codificador rotatorio.

Es destacable el conexionado de los relés. Hubo problemas con este modelo de eje eléctrico, para cambiar de sentido necesitaba cambiar de polaridad. Por ello, tuvimos que utilizar cuatro relés para cada eje, aunque solo tenga 2 sentidos. Para conseguir cambiar la polaridad (ver ilustración 29), cada vez que activamos un eje se activarán 2 relés, los otros 2 no se activarán (ver ilustración 30). Físicamente, se ponen en paralelo un contacto que se cerrará (le llega un positivo) con uno que no se cerrará (le llega un negativo), una vez que pase esto el eje se mueve en un sentido o en otro. En el esquema eléctrico (ver Anexo 3) se visualiza el cambio de polaridad que realizamos para invertir el sentido de los ejes eléctricos Linak LA14. El montaje práctico es el siguiente:



Ilustración 30. Activación de los relés.

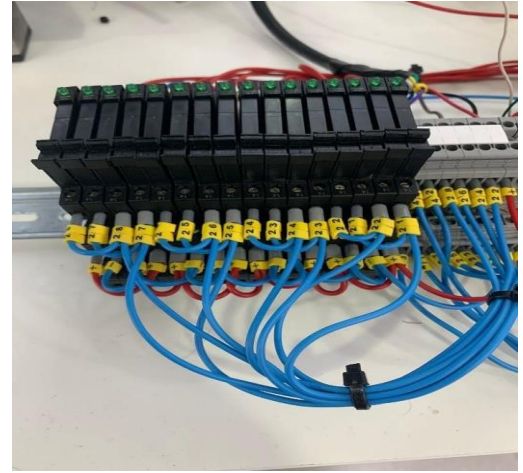


Ilustración 29. Cambio de polaridad.

Por último, como no se sabe la distancia que hay entre el cuadro eléctrico y los dispositivos que se sitúan fuera del cuadro (*encoders* y ejes eléctricos) se han hecho unos conectores. Estos, permiten darle seguridad al circuito, ya que los hilos del *encoder* son muy finos y también da la posibilidad de conectar los dispositivos a la distancia que se desee. Los conectores realizados en los ejes y los *encoders* se realizan de la misma manera, con la única diferencia de que los conectores de los *encoders* son de 4 pines y los de los conectores de los ejes eléctricos son de 5 pines. A continuación, se muestra el conector realizado (ver ilustración 31) para uno de los *encoders*:



Ilustración 31. Conector realizado para los encoders.

3. SOFTWARE

3.1. Descripción de los elementos software utilizados.

3.1.1 TIA Portal

Simatic TIA Portal es un marco de desarrollo de Siemens para programar y trabajar con PLC. El PLC, en resumen, es una interfaz de hardware que primero lee datos de las entradas digitalmente y de varios sensores, luego los procesa de acuerdo con el programa en la memoria y, finalmente, utiliza controladores de hardware y comandos de salida.

El PLC es en realidad una computadora inteligente que puede acceder a todos sus componentes con el programa Simatic TIA Portal y configurarlos y probarlos en diferentes modos en el proceso de simulación y programación. También puede solucionar problemas de programas escritos por otros para PLC de Siemens SIMATIC TIA.

Lo que hace este programa es planificar la lógica de estos PLC sobre cómo procesar los datos de entrada y, en última instancia, qué operaciones realizar en la salida. Este Software ofrece un entorno de desarrollo unificado para la realización de todas las tareas de control, visualización y accionamiento de diferentes tipos de actuadores .

El TIA Portal incorpora las nuevas versiones de software SIMATIC Step 7, WinCC y Stardrive para la programación, parametrización y diagnóstico de los controladores SIMATIC, pantallas de visualización y accionamientos, la nueva versión del sistema de ingeniería SIMATIC STEP 7 para la planificación, la programación y el diagnóstico de controladores SIMATIC.

Hay 3 tipos de programación en TIA Portal:

- KOP o Esquema de contactos.
- FUP o diagrama de funciones.
- SCL o Structured Control Language.

En este proyecto, la mayoría de la programación se ha realizado con el lenguaje KOP, aunque algunas partes se han realizado con lenguaje de control estructurado (SCL).

3.1.2 EPLAN 2.7

EPLAN 2.7 es una potente aplicación con un conjunto completo de herramientas (ver ilustración 32), viene con las mejores características de planificación y gestión para proyectos de automatización y diseño de diferentes sistemas eléctricos. Esta potente aplicación viene dada en un entorno integral con un conjunto directo de herramientas para informes detallados y documentación adicional con el fin de administrar los proyectos con el mínimo esfuerzo.

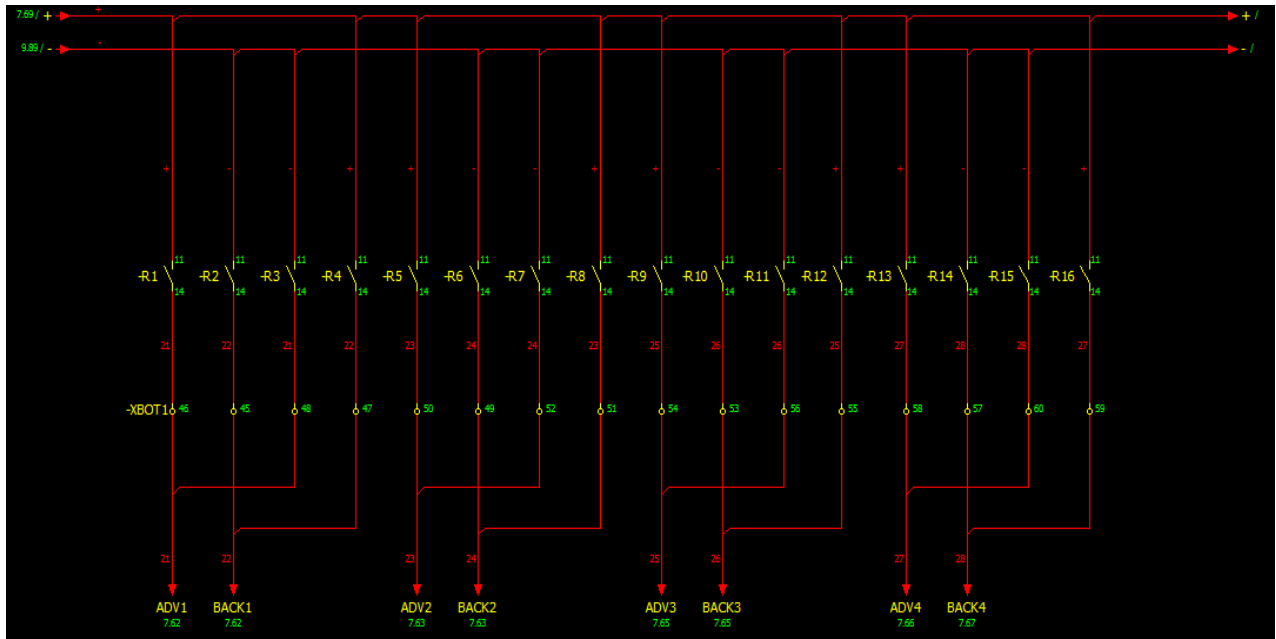


Ilustración 32. Ejemplo de diseño con EPLAN.

Con la licencia de EPLAN se puede contratar anualmente un mantenimiento que incluye un apartado llamado Data Portal muy útil, este nos acerca más a la Industria 4.0 ya que dispone de toda la información en la Nube, se puede trabajar directamente sin tener que buscar o ponerse en contacto con distribuidores o fabricantes, en él encontraremos una variedad de fabricantes que ofrecen las especificaciones técnicas de sus componentes para poder utilizarlos en el programa.

Hay varias formas de crear un proyecto con EPLAN, se puede hacer todo a mano sin ayuda de la inteligencia de este software, o bien se puede trabajar con el denominado proyecto macro, que es la opción más efectiva.

En EPLAN se pueden guardar determinados elementos en forma de macros de ventana o de símbolos. También se pueden guardar como macros de página una o varias páginas abiertas del editor gráfico. Todos estos elementos se guardaran como ficheros del programa.

EPLAN 2.7 está dividido en diferentes módulos, para realizar nuestro esquema eléctrico utilizaremos EPLAN Electric P8. EPLAN Electric P8 es un sistema de ingeniería utilizado para planificar y diseñar esquemas eléctricos para cualquier tipo de instalación/sistema. El software es compatible con una amplia variedad de procedimientos de ingeniería: desde la creación manual hasta una planificación basada en plantillas estandarizadas. Los datos del proyecto, una vez introducidos en el esquema, se convierten en la base de la documentación de la máquina y del sistema de planta.

3.2. Descripción detallada del proceso con TIA Portal

El proyecto consta de dos subprocesos: variar la resistencia que ofrece una cuerda mediante el control de un eje eléctrico y medir la distancia recorrida durante un entrenamiento mediante un *encoder*, este dispositivo convierte el movimiento angular de un eje rotatorio en un código analógico, emitiendo pulsos digitales, mediante un proceso de escalado y de control, representa ese movimiento o posición. En el primer proceso, se realiza el escalado de la señal analógica transmitida por el actuador a una señal establecida en un rango de [0,35]mm . Después, el usuario es el encargado de decidir ,desde la aplicación móvil, que modo de funcionamiento desea.

Con el modo manual puede tener el control del entrenamiento en todo momento, variando la intensidad en cualquier momento. Con el modo automático el usuario puede planificar un entrenamiento determinado antes de realizarlo. Este modo automático manda Set Point en periodos de tiempo definidos por el usuario .La salida deseada se consigue mediante un controlador PID que regula la salida del actuador.

A continuación se muestra el GRAFCET (Gráfico Funcional de Control de Etapas y Transiciones) de nivel 1 del proceso comentado anteriormente (ver ilustración 33):

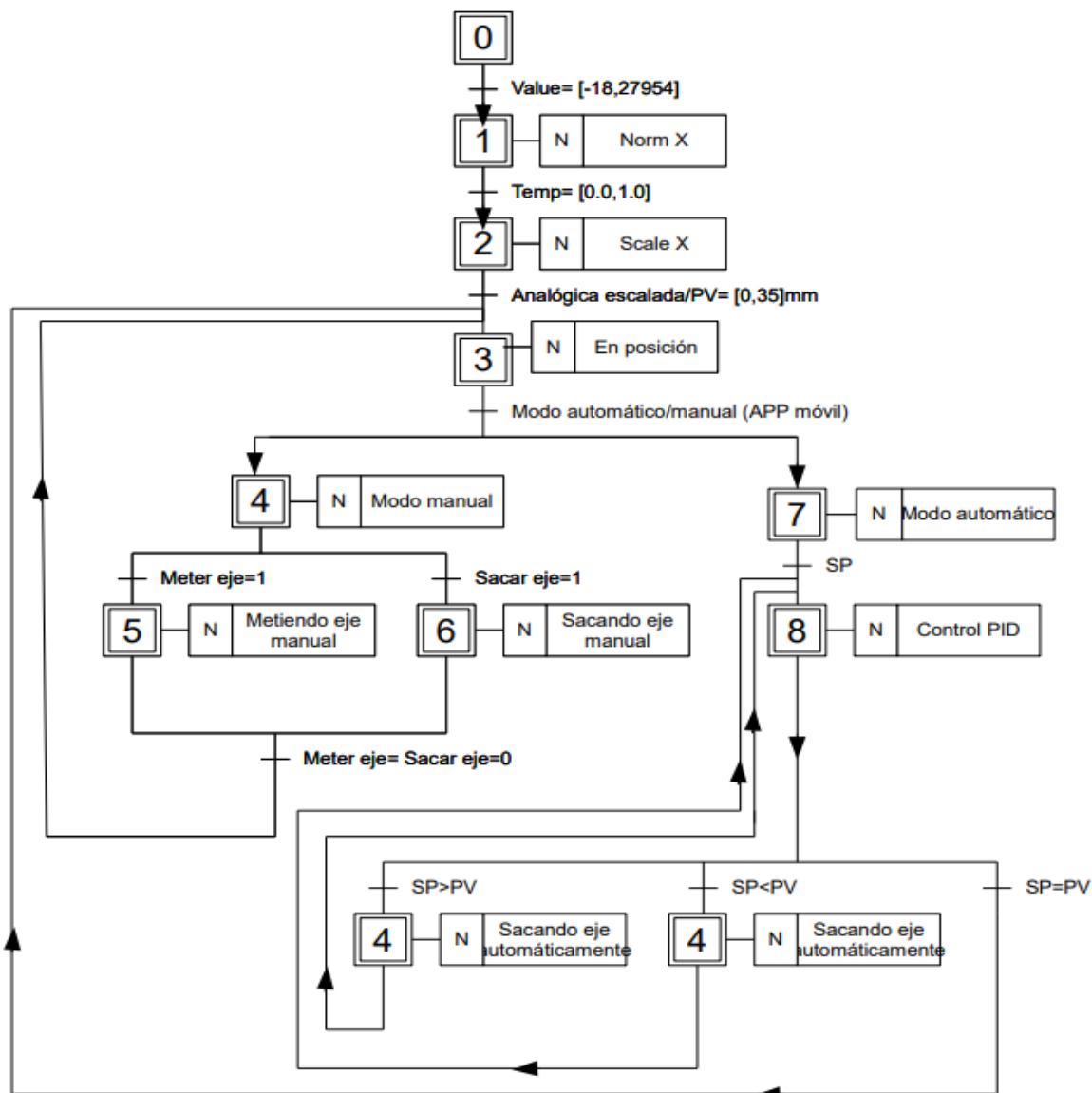


Ilustración 33. GRAFCET del proceso de variado de resistencia mediante el control de un eje eléctrico.

Un GRAFCET es un diagrama funcional, muy utilizado en el campo de la automatización industrial, que describe la secuencia de un proceso que se quiere automatizar. Representa el funcionamiento del proceso, facilitando su implementación posterior. Este diagrama controla el sentido del flujo mediante etapas, acciones asociadas y transiciones.

3.2.1 Entradas y salidas del proceso

En este apartado se explicaran las entradas y las salidas del proceso con su tipo de datos correspondiente, y las variables que hacen posible la comunicación entre la aplicación móvil y el PLC.

Las entradas utilizadas en el todo el proceso son la suma de todas las entradas de todos los subprocesos que se explican en apartados posteriores del proyecto. Hacen posible el escalado de los ejes eléctricos y *encoders*, con el respectivo control de ambos dispositivos. Al mismo tiempo, las entradas del proceso de TIA Portal se comunican con la aplicación móvil por medio de unas variables de comunicación debido a que el usuario desde la aplicación móvil va mandar las órdenes al PLC.

A continuación, se muestran las variables auxiliares en el proceso de comunicación entre la aplicación móvil y el PLC.

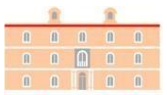
- **ENTRADAS:**

En el caso de las entradas, se convierten las variables de entrada que están en los DB (bloques de datos) de escalado y control del eje eléctrico y del *encoder* (variables que realizan los procesos de escalado y control que explicaremos después) a unas variables que serán las salidas de la aplicación móvil.

Se tiene que realizar la conversión del tipo de datos de algunas variables de entrada para que no exista ningún problema en la comunicación con la aplicación móvil. Esto se debe a que la aplicación solo puede leer datos de tipo entero, entonces para leer el valor con sus decimales se dividirán esos enteros entre 100 o entre 10 para obtener sus decimales en el programa.

Por último, se muestran las variables auxiliares de entrada (ver tabla 6) con su rango de valores y su tipo de datos asociado:

TAGS Entradas PLC	UNIDADES	TIPO
"APP". Estacion1.Esc_Eje.in.Min		INT
"APP". Estacion1.Esc_Eje.in.Max		INT
"APP". Estacion1.Esc_Eje.in.Min_Fisico	0-10 V	REAL
"APP". Estacion1.Esc_Eje.in.Max_Fisico	0-10 V	REAL
"APP". Estacion1.Esc_Eje.in.Min_Escalado	0-35 mm	REAL
"APP". Estacion1.Esc_Eje.in.Max_Escalado	0-35 mm	REAL
"APP". Estacion2.Esc_Eje.in.Min		INT
"APP". Estacion2.Esc_Eje.in.Max		INT
"APP". Estacion2.Esc_Eje.in.Min_Fisico	0-10 V	REAL
"APP". Estacion2.Esc_Eje.in.Max_Fisico	0-10 V	REAL
"APP". Estacion2.Esc_Eje.in.Min_Escalado	0-35 mm	REAL
"APP". Estacion2.Esc_Eje.in.Max_Escalado	0-35 mm	REAL
"APP". Estacion3.Esc_Eje.in.Min		INT
"APP". Estacion3.Esc_Eje.in.Max		INT
"APP". Estacion3.Esc_Eje.in.Min_Fisico	0-10 V	REAL
"APP". Estacion3.Esc_Eje.in.Max_Fisico	0-10 V	REAL
"APP". Estacion3.Esc_Eje.in.Min_Escalado	0-35 mm	REAL
"APP". Estacion3.Esc_Eje.in.Max_Escalado	0-35 mm	REAL



"APP". Estacion4.Esc_Eje.in.Min		INT
"APP". Estacion4.Esc_Eje.in.Max		INT
"APP". Estacion4.Esc_Eje.in.Min_Fisico	0-10 V	REAL
"APP". Estacion4.Esc_Eje.in.Max_Fisico	0-10 V	REAL
"APP". Estacion4.Esc_Eje.in.Min_Escalado	0-35 mm	REAL
"APP". Estacion4.Esc_Eje.in.Max_Escalado	0-35 mm	REAL
"APP". Estacion1.Eje.in.Habilitar		BOOL
"APP". Estacion1.Eje.in.SP	0-35 mm	REAL
"APP". Estacion1.Eje.in.Offset	0-35 mm	REAL
"APP". Estacion1.Eje.in.Sacar_Eje_Man		BOOL
"APP". Estacion1.Eje.in.Meter_Eje_Man		BOOL
"APP". Estacion2.Eje.in.Habilitar		BOOL
"APP". Estacion2.Eje.in.SP	0-35 mm	REAL
"APP". Estacion2.Eje.in.Offset	0-35 mm	REAL
"APP". Estacion2.Eje.in.Sacar_Eje_Man		BOOL
"APP". Estacion2.Eje.in.Meter_Eje_Man		BOOL
"APP". Estacion3.Eje.in.Habilitar		BOOL
"APP". Estacion3.Eje.in.SP	0-35 mm	REAL
"APP". Estacion3.Eje.in.Offset	0-35 mm	REAL
"APP". Estacion3.Eje.in.Sacar_Eje_Man		BOOL
"APP". Estacion3.Eje.in.Meter_Eje_Man		BOOL
"APP". Estacion4.Eje.in.Habilitar		BOOL
"APP". Estacion4.Eje.in.SP	0-35 mm	REAL
"APP". Estacion4.Eje.in.Offset	0-35 mm	REAL
"APP". Estacion4.Eje.in.Sacar_Eje_Man		BOOL
"APP". Estacion4.Eje.in.Meter_Eje_Man		BOOL
"APP". Estacion1.Esc_Encoder.in.Min		DINT
"APP". Estacion1.Esc_Encoder.in.Max		DINT
"APP". Estacion1.Esc_Encoder.in.Min_Escalado	mm	REAL
"APP". Estacion1.Esc_Encoder.in.Max_Escalado	mm	REAL
"APP". Estacion2.Esc_Encoder.in.Min		DINT
"APP". Estacion2.Esc_Encoder.in.Max		DINT
"APP". Estacion2.Esc_Encoder.in.Min_Escalado	mm	REAL
"APP". Estacion2.Esc_Encoder.in.Max_Escalado	mm	REAL
"APP". Estacion3.Esc_Encoder.in.Min		DINT
"APP". Estacion3.Esc_Encoder.in.Max		DINT
"APP". Estacion3.Esc_Encoder.in.Min_Escalado	mm	REAL
"APP". Estacion3.Esc_Encoder.in.Max_Escalado	mm	REAL
"APP". Estacion4.Esc_Encoder.in.Min		DINT
"APP". Estacion4.Esc_Encoder.in.Max		DINT
"APP". Estacion4.Esc_Encoder.in.Min_Escalado	mm	REAL
"APP". Estacion4.Esc_Encoder.in.Max_Escalado	mm	REAL
"APP". Estacion1.Encoder.in.DIR		BOOL
"APP". Estacion1.Encoder.in.CV		BOOL
"APP". Estacion1.Encoder.in.RV		BOOL
"APP". Estacion1.Encoder.in.PERIOD		BOOL
"APP". Estacion1.Encoder.in.NEW_DIR		INT
"APP". Estacion1.Encoder.in.NEW_CV	mm	DINT

"APP". Estacion1.Encoder.in.NEW_RV		DINT
"APP". Estacion1.Encoder.in.NEW_PERIOD		INT
"APP". Estacion2.Encoder.in.DIR		BOOL
"APP". Estacion2.Encoder.in.CV		BOOL
"APP". Estacion2.Encoder.in.RV		BOOL
"APP". Estacion2.Encoder.in.PERIOD		BOOL
"APP". Estacion2.Encoder.in.NEW_DIR		INT
"APP". Estacion2.Encoder.in.NEW_CV	mm	DINT
"APP". Estacion2.Encoder.in.NEW_RV		DINT
"APP". Estacion2.Encoder.in.NEW_PERIOD		INT
"APP". Estacion3.Encoder.in.DIR		BOOL
"APP". Estacion3.Encoder.in.CV		BOOL
"APP". Estacion3.Encoder.in.RV		BOOL
"APP". Estacion3.Encoder.in.PERIOD		BOOL
"APP". Estacion3.Encoder.in.NEW_DIR		INT
"APP". Estacion3.Encoder.in.NEW_CV	mm	DINT
"APP". Estacion3.Encoder.in.NEW_RV		DINT
"APP". Estacion3.Encoder.in.NEW_PERIOD		INT
"APP". Estacion4.Encoder.in.DIR		BOOL
"APP". Estacion4.Encoder.in.CV		BOOL
"APP". Estacion4.Encoder.in.RV		BOOL
"APP". Estacion4.Encoder.in.PERIOD		BOOL
"APP". Estacion4.Encoder.in.NEW_DIR		INT
"APP". Estacion4.Encoder.in.NEW_CV	mm	DINT
"APP". Estacion4.Encoder.in.NEW_RV		DINT
"APP". Estacion4.Encoder.in.NEW_PERIOD		INT

Tabla 6. Variables entrada para la función de la comunicación PLC/app móvil.

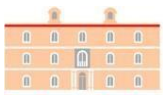
- **SALIDAS:**

Con respecto a las salidas del proceso, se encuentra el accionamiento del actuador en los dos modos de funcionamiento (modo manual y modo automático) y la señal analógica escalada del encoder, que nos permite medir la distancia recorrida durante el entrenamiento.

Para el correcto funcionamiento de ambos dispositivos se necesitan realizar los procesos de escalado y de control. En los siguientes apartados se pueden visualizar las E/S utilizadas para estos procesos.

En el caso de las salidas se tendrá que realizar el proceso opuesto, debido a que desde la aplicación se activarán las salidas (las salidas del PLC serán entradas de la aplicación móvil). Entonces se convierten las variables que llegan desde la aplicación a otras variables que serán las salidas de nuestros DB de escalado y control (las que permiten que el proceso se ejecute correctamente) por medio de estas variables de comunicación.

Al igual que en las entradas se tendrán que convertir el tipo de datos de algunas variables de salida para que no se produzca ningún problema que no permita la comunicación entre la aplicación móvil y el PLC. Por último, se muestran ,en una tabla, las variables auxiliares de salida (ver tabla 7) con su rango de valores y su tipo de datos asociado:



TAGS Salidas PLC	UNIDADES	TIPO
"APP". Estacion1.Escalado_Eje.out.Analogica		INT
"APP". Estacion1.Escalado_Eje.out.Analogica_Fisica	V	REAL
"APP". Estacion1.Escalado_Eje.out.Analogica_Escalada	mm	REAL
"APP". Estacion2.Escalado_Eje.out.Analogica		INT
"APP". Estacion2.Escalado_Eje.out.Analogica_Fisica	V	REAL
"APP". Estacion2.Escalado_Eje.out.Analogica_Escalada	mm	REAL
"APP". Estacion3.Escalado_Eje.out.Analogica		INT
"APP". Estacion3.Escalado_Eje.out.Analogica_Fisica	V	REAL
"APP". Estacion3.Escalado_Eje.out.Analogica_Escalada	mm	REAL
"APP". Estacion4.Escalado_Eje.out.Analogica		INT
"APP". Estacion4.Escalado_Eje.out.Analogica_Fisica	V	REAL
"APP". Estacion4.Escalado_Eje.out.Analogica_Escalada	mm	REAL
"APP". Estacion1.Eje.out.PV	mm	REAL
"APP". Estacion1.Eje.out.Sacar_Eje_Aut		BOOL
"APP". Estacion1.Eje.out.Meter_Eje_Aut		BOOL
"APP". Estacion1.Eje.out.En_Posicion		BOOL
"APP". Estacion1.Eje.out.SP_Normalizado	mm	REAL
"APP". Estacion1.Eje.out.Sacando_Eje		BOOL
"APP". Estacion1.Eje.out.Metiendo_Eje		BOOL
"APP". Estacion2.Eje.out.PV	mm	REAL
"APP". Estacion2.Eje.out.Sacar_Eje_Aut		BOOL
"APP". Estacion2.Eje.out.Meter_Eje_Aut		BOOL
"APP". Estacion2.Eje.out.En_Posicion		BOOL
"APP". Estacion2.Eje.out.SP_Normalizado	mm	REAL
"APP". Estacion2.Eje.out.Sacando_Eje		BOOL
"APP". Estacion2.Eje.out.Metiendo_Eje		BOOL
"APP". Estacion3.Eje.out.PV	mm	REAL
"APP". Estacion3.Eje.out.Sacar_Eje_Aut		BOOL
"APP". Estacion3.Eje.out.Meter_Eje_Aut		BOOL
"APP". Estacion3.Eje.out.En_Posicion		BOOL
"APP". Estacion3.Eje.out.SP_Normalizado	mm	REAL
"APP". Estacion3.Eje.out.Sacando_Eje		BOOL
"APP". Estacion3.Eje.out.Metiendo_Eje		BOOL
"APP". Estacion4.Eje.out.PV	mm	REAL
"APP". Estacion4.Eje.out.Sacar_Eje_Aut		BOOL
"APP". Estacion4.Eje.out.Meter_Eje_Aut		BOOL
"APP". Estacion4.Eje.out.En_Posicion		BOOL
"APP". Estacion4.Eje.out.SP_Normalizado	mm	REAL
"APP". Estacion4.Eje.out.Sacando_Eje		BOOL
"APP". Estacion4.Eje.out.Metiendo_Eje		BOOL
"APP". Estacion1.Escalado_Encoder.out.Entrada		DINT
"APP". Estacion1.Escalado_Encoder.out.Salida_Escalada	mm	REAL

"APP". Estacion2.Escalado_Encoder.out.Entrada		DINT
"APP". Estacion2.Escalado_Encoder.out.Salida_Escalada	mm	REAL
"APP". Estacion3.Escalado_Encoder.out.Entrada		DINT
"APP". Estacion3.Escalado_Encoder.out.Salida_Escalada	mm	REAL
"APP". Estacion4.Escalado_Encoder.out.Entrada		DINT
"APP". Estacion4.Escalado_Encoder.out.Salida_Escalada	mm	REAL
"APP". Estacion1.Encoder.out.BUSY		BOOL
"APP". Estacion1.Encoder.out.ESTATUS		INT
"APP". Estacion2.Encoder.out.BUSY		BOOL
"APP". Estacion2.Encoder.out.STATUS		INT
"APP". Estacion3.Encoder.out.BUSY		BOOL
"APP". Estacion3.Encoder.out.STATUS		INT
"APP". Estacion4.Encoder.out.BUSY		BOOL
"APP". Estacion4.Encoder.out.STATUS		INT

Tabla 7. Variables salida para la función de la comunicación PLC/app móvil.

3.2.2 Escalado del eje eléctrico

Al ser una CPU avanzada, los bloques de programa actuales facilitan mucho la función de escalado de una variable, por tanto, la programación es más sencilla que en una CPU inferior. Las funciones que facilitan el escalado son NORM X y SCALE X. Estos dos bloques están disponibles en CPUs superiores o iguales a la S71200.

- **NORM X: NORMALIZAR:** Esta instrucción normaliza el valor del parámetro de entrada VALUE representándolo en una escala lineal de valores comprendidos entre 0.0 y 1.0. Las variables MIN y MAX sirven para definir los límites del rango de valores de la entrada VALUE.

En función de la posición de la entrada VALUE, se calcula el resultado y se guarda como número en coma flotante en la salida OUT. Si el valor que se debe normalizar es igual o menor al valor de la entrada MIN, la salida devuelve el valor 0.0. Si el valor que se debe normalizar es igual o mayor al valor de la entrada MAX, la salida devuelve el valor 1.0 (ver ilustración 34).

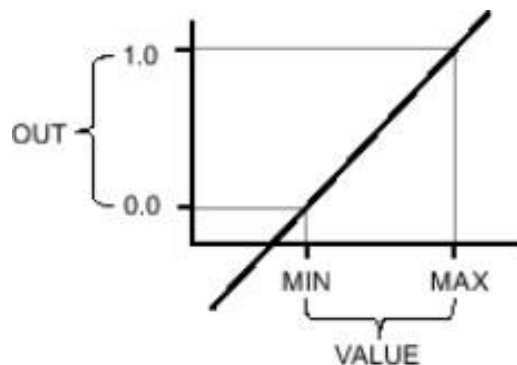


Ilustración 34. Valores normalizados.

- **Parámetros de la instrucción NORM X:**

EN: Entrada de habilitación.

MIN: Límite inferior de la escala, en nuestro caso es de -18.

VALUE: Valor que se quiere normalizar, en nuestro caso es la dirección de la entrada analógica.

MAX: Límite superior de la escala, en nuestro caso es de 27954.

ENO: Salida de habilitación.

OUT: Valor de salida normalizado, este parámetro siempre devuelve un valor entre 0.0 y 1.0, aquí se guarda la información en la variable #Temp1.

- SCALE |X|: ESCALAR:** Esta instrucción realiza el escalado del valor de la entrada VALUE mapeándolo en un determinado rango de valores establecidos por el usuario. Al ser ejecutada esta instrucción, el número en coma flotante de la entrada VALUE se escala al rango de valores definido por los parámetros MIN y MAX, el resultado de esta instrucción se guarda en la salida OUT (ver ilustración 35).

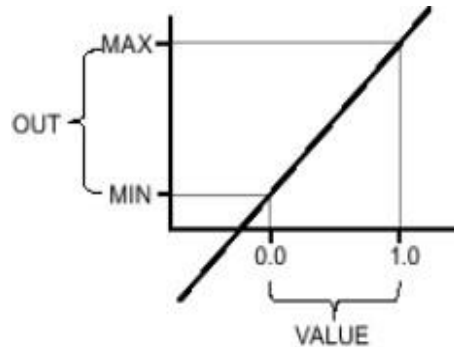


Ilustración 35. Valores escalados.

- Parámetros de la instrucción SCALE X:**

EN: Entrada de habilitación.

MIN: Límite inferior del rango de valores, en nuestro caso es 0, que se corresponde con los 0 mm de nuestro ejeeléctrico, cuando este devuelve 0V.

VALUE: Valor que se quiere escalar, en nuestro equipo utilizamos la misma variable que en el parámetro OUT de la función NORM_X, esta variable siempre debe tener un valor entre 0.0 y 1.0.

MAX: Límite superior del rango de valores, en nuestro caso es 35 mm, que se corresponde con los 35 mm de nuestro eje eléctrico, cuando este nos devuelve 10V.

ENO: Salida de habilitación.

OUT: Valor de salida escalado, este parámetro nos envía el valor de la entrada analógica escalada, en nuestro caso será un valor de entre 0 y 35 mm.

Con los controladores actuales, realizar el proceso de escalado de una señal analógica solo se hace en una línea de código (ver ilustración 36), mientras que en controladores inferiores se realiza en tres o cuatro líneas de código.

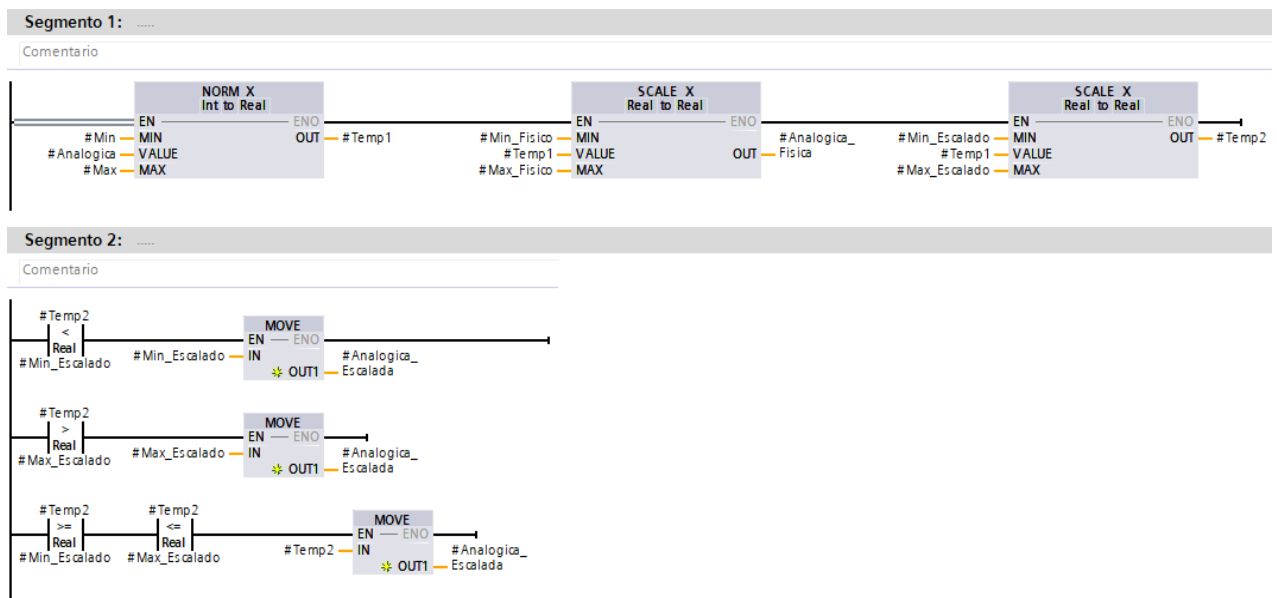


Ilustración 36. Proceso de escalado de una señal analógica con TIA Portal.

En este caso, se han utilizado en el programa dos instrucciones SCALE_X para saber, a parte de la distancia, el valor de voltaje que nos devuelve la señal analógica. Se podría haber obtenido únicamente la distancia y no hubiera habido ningún problema.

El segmento 2 del código anterior es básicamente de seguridad. Si la analógica escalada que llega es menor al mínimo escalado, el programa devuelve el mínimo escalado. Si la analógica escalada que llega es mayor al máximo escalado, el programa devuelve el máximo escalado. Sin embargo, si la analógica escalada que llega es menor que el máximo y mayor que el mínimo, devolverá el valor de analógica escalada verdadero. Con este segmento, evitamos, por ejemplo, que si la analógica escalada tiene un rango [0,35]mm, no pueda darnos valores fuera de dicho rango.

- Las variables auxiliares del proceso (ver ilustración 37) de escalado del eje eléctrico son las siguientes:

Escalado_Eje1_DB		
	Nombre	Tipo de datos
1	Input	
2	Analogica	Int
3	Min	Int
4	Max	Int
5	Min_Fisico	Real
6	Max_Fisico	Real
7	Min_Escalado	Real
8	Max_Escalado	Real
9	Output	
10	Analogica_Fisica	Real
11	Analogica_Escalada	Real
12	InOut	
13	Static	

Ilustración 37. Variables auxiliares del proceso de escalado.

Un *function block* es un bloque de programa que contiene un proceso específico dentro del código del programa, su uso facilita la implementación de la programación. El *function block* resultante (ver ilustración 38) de este proceso de escalado es el siguiente:

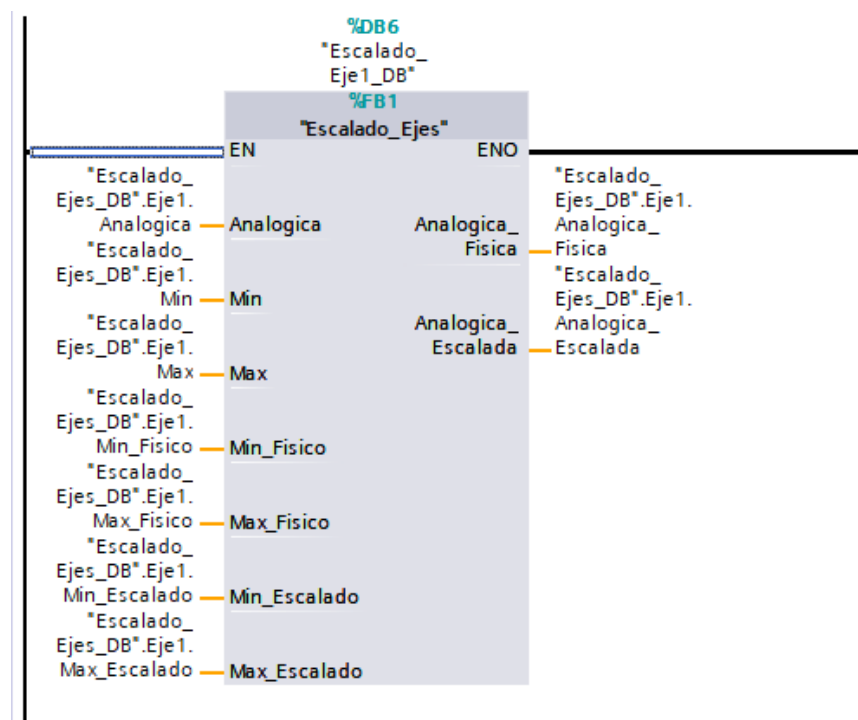


Ilustración 38. . Function block resultante del proceso de escalado de una señal analógica.

A continuación, se muestra una tabla con el rango de valores de las variables de escalado utilizadas en el proceso (ver tabla 8):

MAGNITUDES ESCALADO EJE ELÉCTRICO		
VARIABLES ESCALADO	RANGO	COMENTARIOS
ANALÓGICA	[-18, 27954]	Analógica sin unidades que llega al PLC desde el eje eléctrico
ANALÓGICA FÍSICA	[0-10] V	Conversión a [0,10]V mediante la ecuación de una recta escalada
ANALÓGICA ESCALADA	[0,35] mm	Conversión de ese voltaje a un rango de distancia [0,35]mm

Tabla 8. Magnitudes obtenidas durante el proceso de escalado del eje eléctrico.

3.2.3 Escalado del *encoder*

El escalado de la señal obtenida por los *encoders* se realiza de la misma manera que los ejes eléctricos, es decir, con las instrucciones NORM_X y SCALE_X. Los *encoders* utilizados en el proyecto transmiten 600 pulsos por vueltas. El radio exterior del *encoder* es de 3,5 cm. Con este radio podemos calcular la longitud de la circunferencia del *encoder*:

$$L_{\text{encóder}} = 2 * \pi * R_{\text{encóder}} = 2 * \pi * 3,5 \approx 22 \text{ cm};$$

De igual manera que los ejes eléctricos, el escalado se puede hacer con una línea de código (ver ilustración 39). La diferencia, en estecaso, es que no se han puesto las líneas de código de seguridad que se pusieron en los ejes. Esto se debe a que los ejes eléctricos tienen un rango limitado [0,35] mm; sin embargo, los *encoders* no tienen rango de medida ya que es un instrumento que no sabe la distancia que van a tener que medir porque no se saben las dimensiones del terreno donde se va a entrenar, ni la longitud que van a tener que recorrer los jugadores.

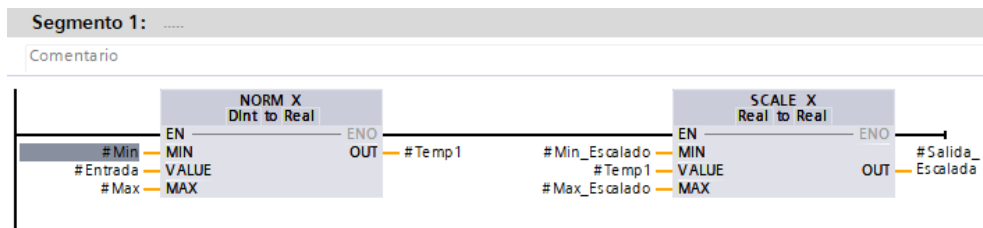


Ilustración 39. Proceso de escalado de la señal enviada por los *encoders*.

- Las variables auxiliares del proceso de escalado del *encoder* son las siguientes (ver ilustración 40):

Escalado_Encoder		
	Nombre	Tipo de datos
1	Input	
2	Entrada	DInt
3	Min	DInt
4	Max	DInt
5	Min_Escalado	Real
6	Max_Escalado	Real
7	Output	
8	Salida_Escalada	Real
9	InOut	
10	<Agregar>	
11	Static	
12	<Agregar>	
13	Temp	
14	Temp1	Real
15	Constant	

Ilustración 40. Variables auxiliares del proceso de escalado del *encoder*.

- El *function block* resultante de este proceso de escalado es el siguiente (ver ilustración 41):

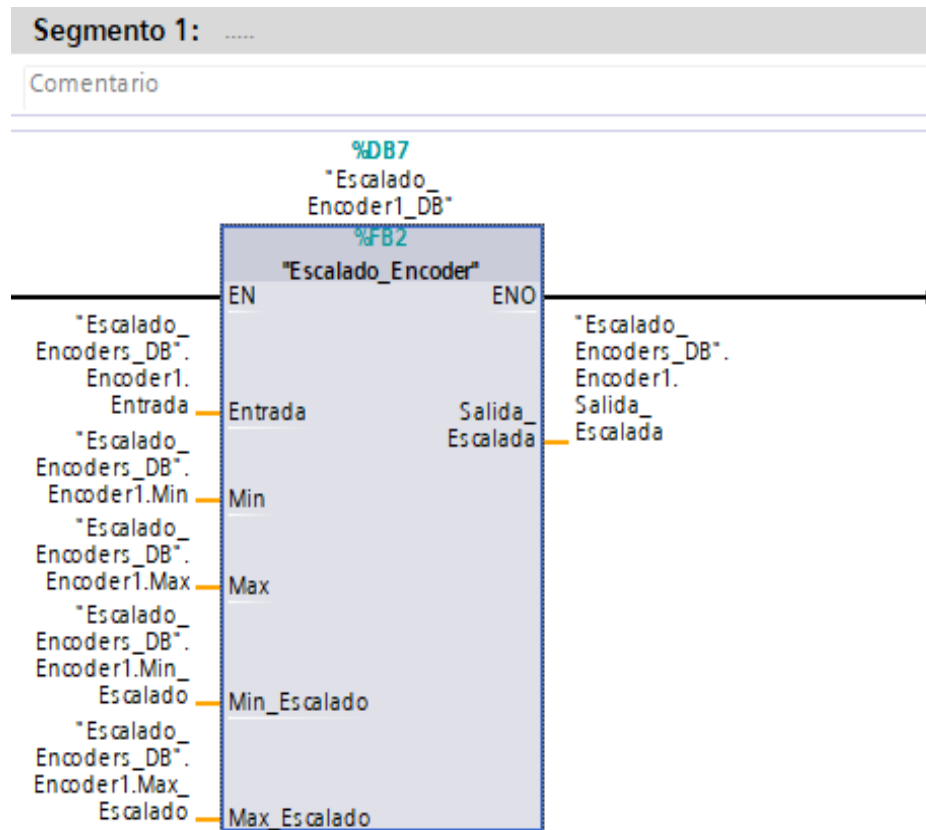


Ilustración 41. *Function block* resultante del proceso de escalado del *encoder*.

Por último, se muestra una tabla con el rango de valores de las variables de escalado utilizadas en el proceso de escalado del *encoder* (ver tabla 9):

MAGNITUDES ESCALADO ENCODER		
VARIABLES ESCALADO	RANGO	COMENTARIOS
PULSOS	[0, 600] pulsos	Son los pulsos que envía el <i>encoder</i> PLC mientras está girando. En una vuelta completa envía 600 pulsos.
DISTANCIA	[0-22] cm	Es la distancia que recorre el <i>encoder</i> durante una vuelta (22 cm). El objetivo de este escalado es medir la distancia recorrida convirtiendo estos pulsos.

Tabla 9. *Magnitudes* obtenidas durante el proceso de escalado del *encoder*.

3.2.4 Control eje eléctrico

Se necesita controlar la señal analógica del eje eléctrico para el correcto funcionamiento de la máquina. La intensidad del ejercicio es función de la variación de dicha señal analógica.

Para realizar el control de los ejes eléctricos se ha creado un *function block* llamado "Control_Eje1_DB" (ver ilustración 43). Este bloque permite que el eje eléctrico se mueva a la posición deseada por el usuario. También podemos observar las variables auxiliares utilizadas en el proceso (ver ilustración 42):

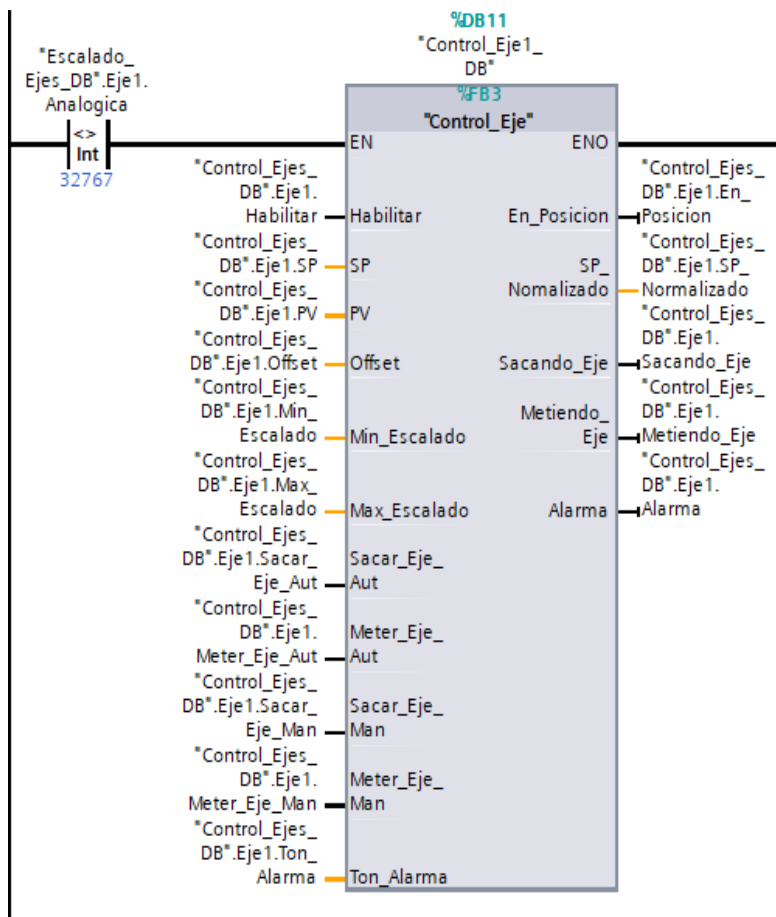


Ilustración 42. Function block resultante del proceso de control del eje eléctrico.

Control_Eje		
	Nombre	Tipo de dato
1	Input	
2	Habilitar	Bool
3	SP	Real
4	PV	Real
5	Offset	Real
6	Min_Escalado	Real
7	Max_Escalado	Real
8	Sacar_Eje_Aut	Bool
9	Meter_Eje_Aut	Bool
10	Sacar_Eje_Man	Bool
11	Meter_Eje_Man	Bool
12	Ton_Alarma	Time
13	Output	
14	En_Posicion	Bool
15	SP_Normalizado	Real
16	Sacando_Eje	Bool
17	Metiendo_Eje	Bool
18	Alarma	Bool
19	InOut	
20	<Agregar>	
21	Static	
22	SP_Sup	Real
23	SP_Inf	Real
24	Tiempo	TON_TIME
25	Temp	
26	<Agregar>	
27	Constant	

Ilustración 43. Variables auxiliares del proceso.

A continuación, se explica el significado de las variables auxiliares utilizadas durante el proceso de control del eje eléctrico que forman el *function block* (%FB3):

EN: entrada de habilitación.

Habilitar: habilita el movimiento del eje de forma automática. Si habilitar no está en "1" solo se puede mover el eje de forma manual.

SP: es el punto, dentro del rango del eje eléctrico, que se quiere conseguir. Es de tipo real. Una vez que se cree un entrenamiento desde la aplicación, cada vez que aumente o disminuya la intensidad del ejercicio, el *set point* cambiará.

PV: indica la posición del eje eléctrico actual.

Offset: este *offset* es muy importante. Cuando se marca un *set point*, el eje intenta llegar a ese punto, pero le cuesta mucho llegar al punto exacto y estará regulando infinitamente. Por ello, el *offset* le da al PID un pequeño margen (que será insignificante para nosotros), y una vez que llegue a ese margen (0,05 mm) el PID dejará de regular para optimizar el sistema.

Por ejemplo, si se pone el *set point* en 24 mm, una vez que llegue al rango [23,95-24,05) mm el PID dejará de regular y el eje eléctrico se parará.

Sacar_Eje_Aut: se activa cada vez que el eje avance hacia un *set point* (desde el programa o la aplicación móvil).

Meter_Eje_Aut: se activa cada vez que el eje retrocede hacia un *set point* (desde el programa o la aplicación móvil).

Sacar_Eje_Man: este parámetro permite sacar el eje de forma manual (desde el programa).

Meter_Eje_Man: este parámetro permite meter el eje de forma manual (desde el programa).

Ton_alarma: es el tiempo que estará la alarma sonando si salta. La alarma salta cuando esté en el modo automático y el eje eléctrico no esté en la situación que indique el *set point*.

ENO: salida de habilitación.

En_Posición: indica si el eje ha conseguido llegar al *set point*.

SP_Normalizado: Es una instrucción de seguridad. Si el *set point* es mayor que el valor máximo de escalado, el programa devolverá este *set point* normalizado con ese valor máximo. Si el *set point* es menor que el valor mínimo de escalado, el programa devolverá este *set point* normalizado con ese valor mínimo. Sin embargo, si el *set point* se encuentra entre el máximo y el mínimo escalado, el *set point* normalizado será igual al *set point*.

Sacando_Eje: salida que indica que el eje está saliendo.

Metiendo_Eje: salida que indica que el eje se está metiendo.

Alarma: la alarma saltará cuando la posición del eje eléctrico no coincida con el *set point*. Esta instrucción indica que se ha producido un problema en el eje eléctrico.

3.2.5 Control del *encoder*

A veces, se tiene la necesidad de realizar operaciones de contaje a una frecuencia mayor que la que puede tener un OB cualquiera, los *encoders* suelen tener frecuencias de trabajo mayores que las de un OB típico. Las aplicaciones para los *encoders* se deben realizar mediante las instrucciones adecuadas y, en este caso, las instrucciones que deben tenerse en cuenta son las de lectura rápida, más concretamente la "CTRL_HSC".

Por el contrario, si se tuviese una función o aplicación que la frecuencia a contar los pulsos fuese menor que la del OB a utilizar, sí se debería utilizar las instrucciones de contaje de uso común "CTU, CTD o CTUD".

Es una instrucción bastante versátil, ya que esta misma instrucción también puede servir para medir frecuencia de pulsos. Estos contadores también son utilizados para la instrucción para la generación de pulsos. Con lo que podemos concluir que puede utilizarse en tres modos distintos, pero sólo es posible que se elija un modo de funcionamiento (ver ilustración 44):

- **Contaje:** cuenta los pulsos cuadrados.
- **Frecuencia:** actúa como medidor de frecuencias de pulsos cuadrados.
- **Eje:** es la configuración para generar pulsos cuadrados.

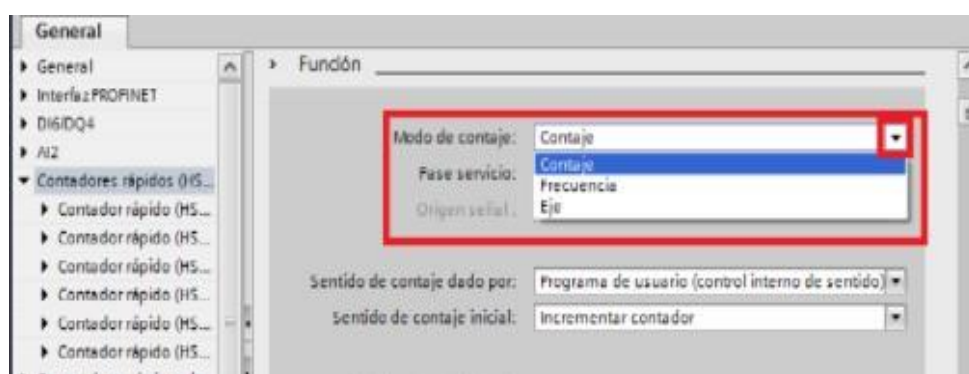
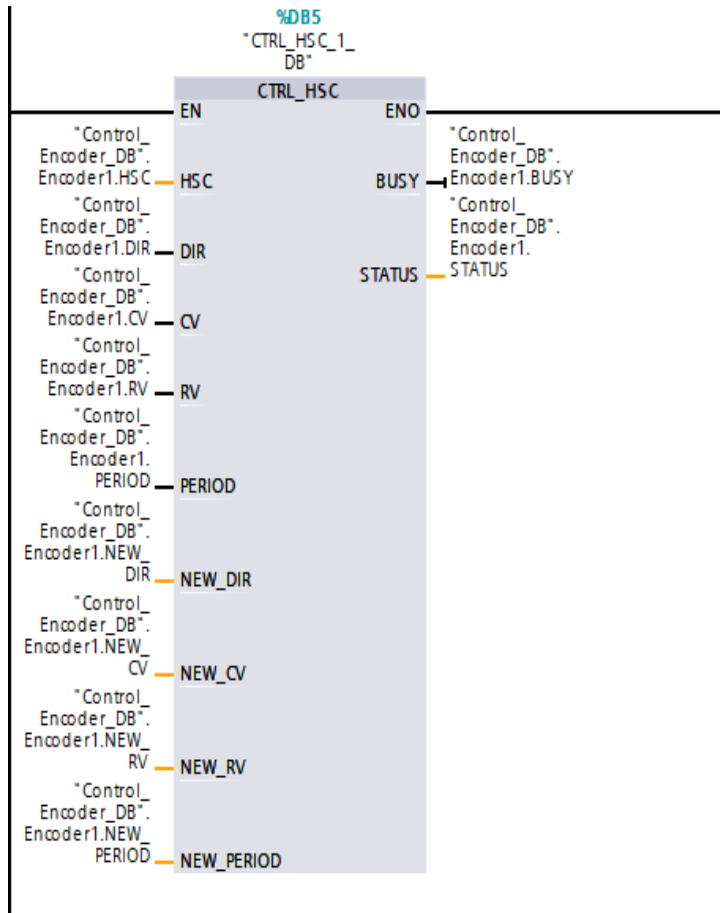


Ilustración 44. Elección del modo de funcionamiento de la instrucción HSC.

En este caso, se va a utilizar la instrucción "CTRL_HSC" en modo contaje. Este DB integrado en el S7-1200 (ver ilustración 45) permite medir la distancia recorrida por los jugadores durante el entrenamiento. Podríamos haber creado un *function block* para realizar el control de los *encoders*, pero se decidió utilizar la solución que nos ofrece SIEMENS con sus parámetros establecidos. A continuación, también se muestran las variables utilizadas en el proceso de control del *encoder* y los tipo de datos utilizados (ver tabla 10).



Parámetro	Declaración	Tipo de datos
EN	INPUT	BOOL
ENO	OUTPUT	BOOL
HSC	INPUT	HW_HSC
DIR	INPUT	BOOL
CV	INPUT	BOOL
RV	INPUT	BOOL
PERIOD	INPUT	BOOL
NEW_DIR	INPUT	INT
NEW_CV	INPUT	DINT
NEW_RV	INPUT	DINT
NEW_PERIOD	INPUT	INT
BUSY	OUTPUT	BOOL
STATUS	OUTPUT	WORD

Ilustración 45. Función block resultante del proceso de control del encoder.

Tabla 10. Tipo de datos de los parámetros utilizados en el control del encoder.

A continuación, se explican los parámetros más importantes de este DB:

EN: entrada de habilitación.

NEW_DIR: el sentido de contaje define si un contador rápido debe contar hacia delante o hacia atrás. Los siguientes valores de la entrada NEW_DIR definen el sentido de contaje 1 = adelante, -1 = atrás.

El sentido de contaje indicado en la entrada NEW_DIR se carga en un contador rápido si está activado el bit de la entrada DIR.

DIR: habilitación del nuevo sentido de contaje.

NEW_CV: el valor de contaje es el valor inicial con el que un contador rápido comienza a contar. El valor de contaje puede estar comprendido en un rango de -2147483648 a 2147483647.

El valor de contaje indicado en la entrada NEW_CV se carga en un contador rápido si está activado el bit de la entrada CV.

CV: habilitación del nuevo valor de contaje.

NEW_RV: el valor de referencia puede compararse con el valor de contaje actual para poder desencadenar una alarma. El valor de referencia, igual que el valor de contaje, puede estar comprendido en un rango de -2147483648 a 2147483647.

El valor de referencia indicado en la entrada NEW_RV se carga en un contador rápido si está activado el

bit de la entrada RV.

RV: habilitación del nuevo valor de referencia.

NEW_PERIOD: periodo de medición de frecuencia. Los siguientes valores de la entrada NEW_PERIOD definen el periodo de medición de frecuencia: 10 = 0,01s, 100 = 0,1s, 1000 = 1s.

El tiempo indicado en la entrada NEW_PERIOD se carga en un contador rápido si está activado el bit de la entrada PERIOD.

PERIOD: habilitación del nuevo periodo de medición de frecuencia.

HSC: muestra el valor actualizado de conteo de pulsos. Puede ser un valor negativo o positivo.

ENO: la salida de habilitación ENO se activa únicamente si el estado lógico de la entrada de habilitación EN es "1" y no ocurren errores al ejecutar la operación.

BUSY: indica el estado de procesamiento del proceso. Si hay un contador rápido en la CPU o en la *Signal Board*, el parámetro BUSY tiene siempre el valor 0.

STATUS: muestra el estado de la operación. En la salida STATUS es posible consultar si han ocurrido errores al ejecutar la instrucción "CTRL_HSC". En la tabla siguiente se explica el significado de los valores que se depositan en la salida STATUS (ver tabla 11):

Código de error (hexadecimal)	Descripción
0	Ningún error
80A1	Identificador de hardware no válido del contador rápido
80B1	Sentido de contaje (NEW_DIR) no válido
80B2	Valor de contaje (NEW_CV) no válido
80B3	Valor de referencia (NEW_RV) no válido
80B4	Periodo de medición de frecuencia (NEW_PERIOD) no válido
80C0	Acceso repetido al contador rápido
80D0	El contador rápido (HSC) no está activado en la configuración hardware de la CPU

Tabla 11. . Códigos de error del parámetro STATUS.

3.2.6 Mecanismo de control

Los sistemas de control son un conjunto de mecanismos utilizados para conducir el comportamiento de otros sistemas, con la finalidad de mantener constante una variable. En este caso, la variable que se quiere controlar es la entrada analógica del eje eléctrico, por lo tanto, se trata de un control de una única variable.

Existen diferentes maneras de diseñar un controlador óptimo para un proceso. Se optó por realizar el control de la señal analógica mediante un controlador PID.

La regulación PID es el método de control más utilizado en los procesos industriales que trabajan en lazo cerrado. Consta de tres ecuaciones o algoritmos al que corresponden cada una de las letras de su nombre Proporcional + Integral + Derivativo (ver ilustración 46). Estos reguladores permiten un mejor control del tiempo de respuesta, aumentando su precisión, y ayudan a mejorar errores propios de los dispositivos.

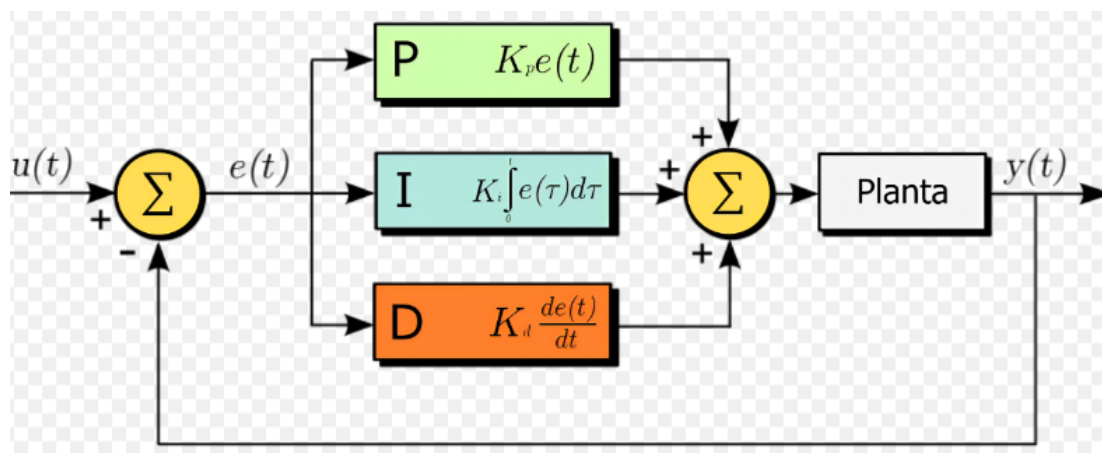


Ilustración 46. Controlador PID.

Mediante la regulación PID lo que se pretende es que la señal de salida de un proceso alcance la señal de referenciade entrada (*set point*) lo más rápido posible y con el error lo más pequeño posible. No solo es importante el error que se consigue al final del proceso, una vez las señales entren en régimen estacionario, sino que son importantes los valores que alcanza la señal de salida durante el proceso de ajuste. Aunque la señal al final alcance la señal de referencia no son convenientes oscilaciones por encima del 30% de la señal de referencia.

La configuración de un controlador PID empieza determinando las ganancias óptimas para tu sistema. Estas ganancias son las constantes proporcional, integral y derivativa:

- **Término Proporcional (P):** Esta constante es muy importante para lograr estabilidad y capacidad de respuesta. Aumentar el valor P hace que el controlador aumente el tiempo de respuesta, pero un valor alto de P provocaría inestabilidad.
- **Término derivativo (I):** Esta constante elimina cualquier error en estado estacionario y aplica una acción correctiva.
- **Término integral (D):** Esta constante es responsable de predecir cambios repentinos en el comportamiento del sistema amortiguando las oscilaciones, dando estabilidad al sistema.

Para regular el cambio de posición del eje eléctrico se ha optado por utilizar en TIA PORTAL la función PID_3Step V2 (SIEMENS, Controlador programable S7-1200).. Esta regulación permite al eje eléctrico alcanzar el *set point* lo más rápido posible y con el menor error posible. Esta instrucción reguladora ofrece las siguientes ventajas:

- Cambio entre el modo automático y manual.
- Medición del tiempo de transición de la válvula.
- Visualización del ajuste de la válvula.

- Permite cambiar la configuración en tiempo de ejecución.
- Monitoreo en línea del módulo controlador "PID_3Step V2".
- Especificación del comportamiento en caso de error y su simulación.
- Compensación de perturbaciones en operación simulada.
- Visualización del *set point*, la variable real y la variable manipulada.

A parte de todas estas ventajas, se optó por el uso del PID_3Step V2 por su funcionalidad de "Tuning" con los parámetros P, I y D, esto permite calcular estos parámetros automáticamente en función del sistema controlado (se calculan en el apartado posterior). Sin embargo, también se pueden especificar los parámetros de control manualmente.

PID_3Step V2 es un regulador PIDT1 con Anti-Windup, controla la sobrecarga del término integral con el fin de evitar los transitorios muy largos, (ver ilustración 47) y ponderación de la acción P y D. El algoritmo PID funciona de acuerdo con la siguiente fórmula:

$$\Delta y = K_p \cdot s \cdot [(b \cdot w \cdot x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot S}{a \cdot T_D \cdot S + 1} (c \cdot w - x)]$$

Δy	Valor de salida del algoritmo PID
K_p	Ganancia proporcional
s	Operador laplaciano
b	Ponderación de la acción P
w	Consigna
x	Valor real
T_i	Tiempo de integración
T_D	Tiempo derivativo
a	Coficiente para el retardo de la acción derivada (retardo de la acción derivada $T1 = a \times T_D$)
c	Ponderación de la acción D

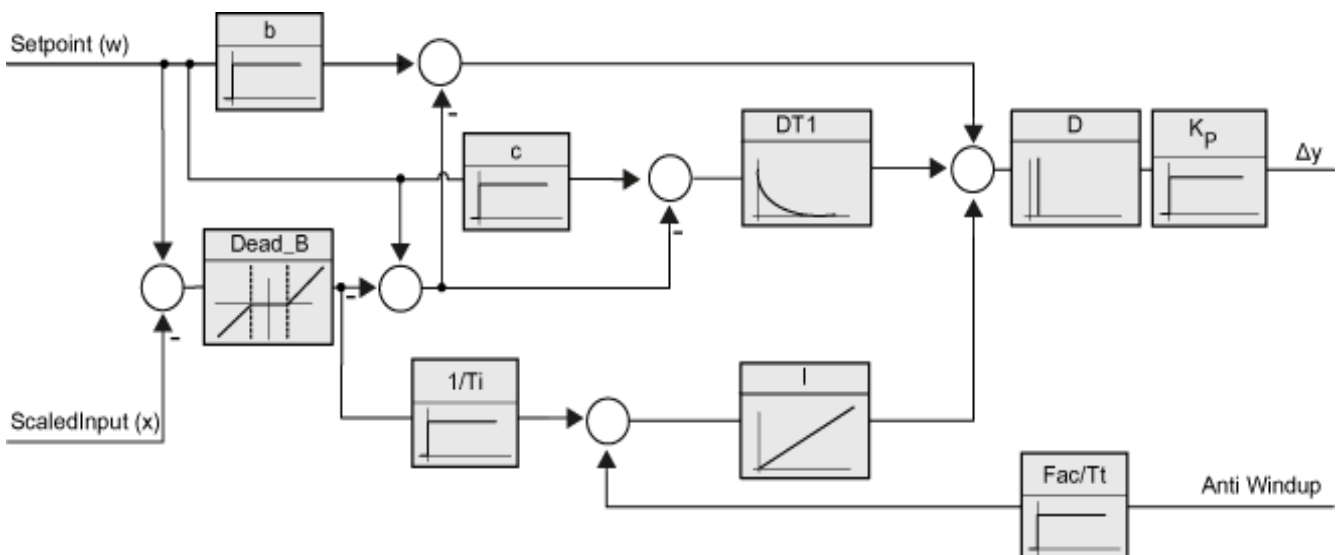


Ilustración 47. Diagrama de bloques PIDT1 con Anti-Windup.

Por último, se va a proceder a configurar los parámetros ,en TIA Portal, de este PID_3Step V2 para su correcto funcionamiento (ver ilustración 49, 50 y 51). Para comenzar seleccionamos el tipo de regulación, seleccionando la unidad de longitud (ver ilustración 48). Se selecciona sin realimentación el tipo de estado de dispositivo devuelto al lazo PID.

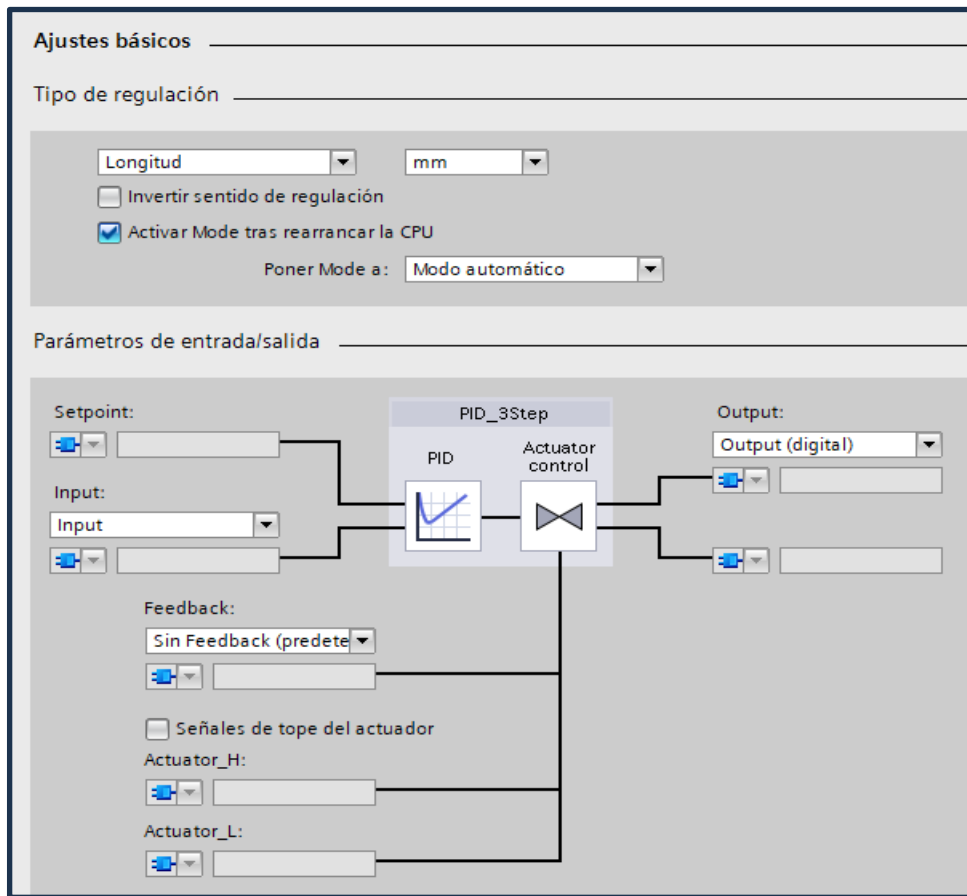


Ilustración 48. Ajustes básicos PID1 con Anti-Windup.

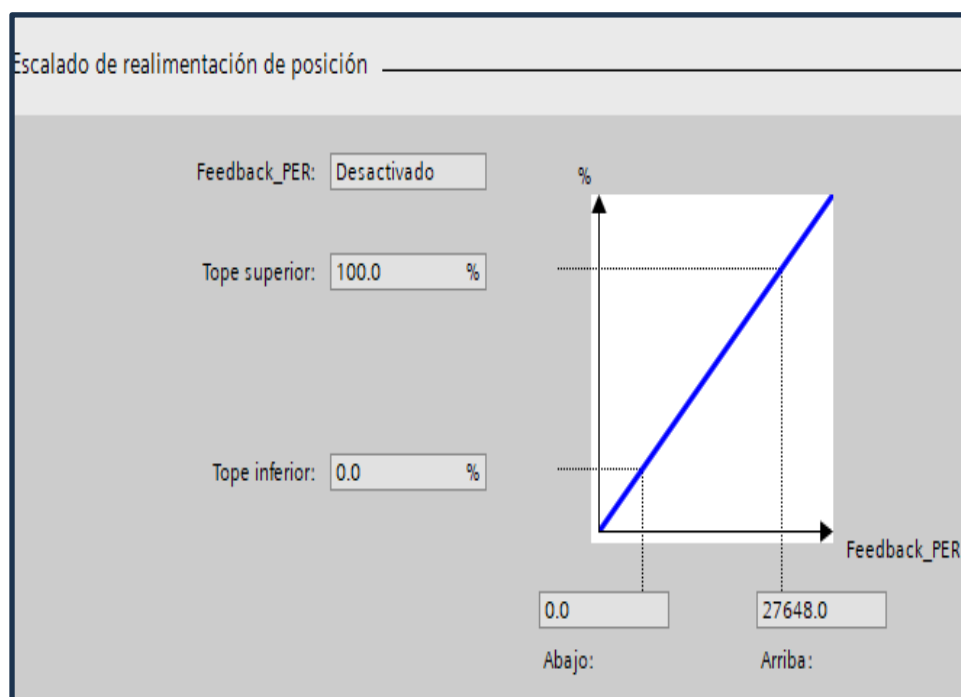


Ilustración 49. Escalado de realimentación de posición.

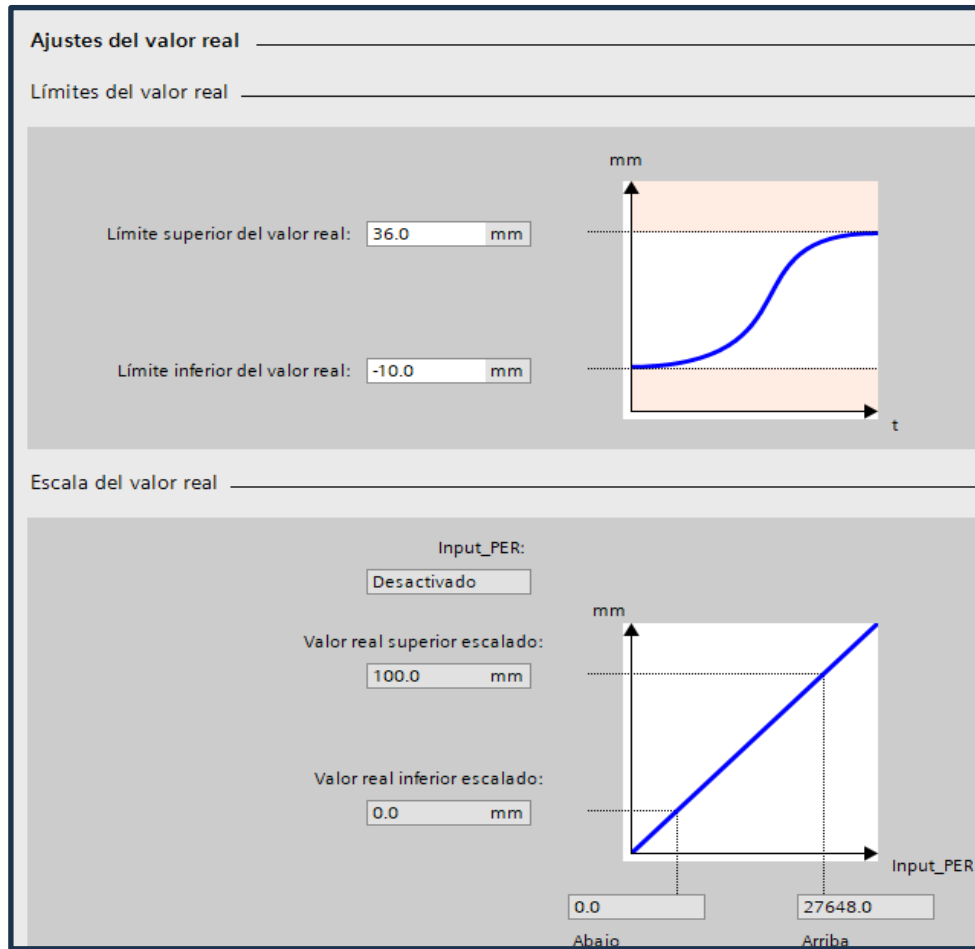


Ilustración 50. Ajustes del valor real escalado.

En nuestro caso, se utiliza la instrucción "PID_3Step V2" en modo automático. Este DB de SIEMENS del S7-1200 permite posicionar el eje eléctrico en el *set point* que se indique.

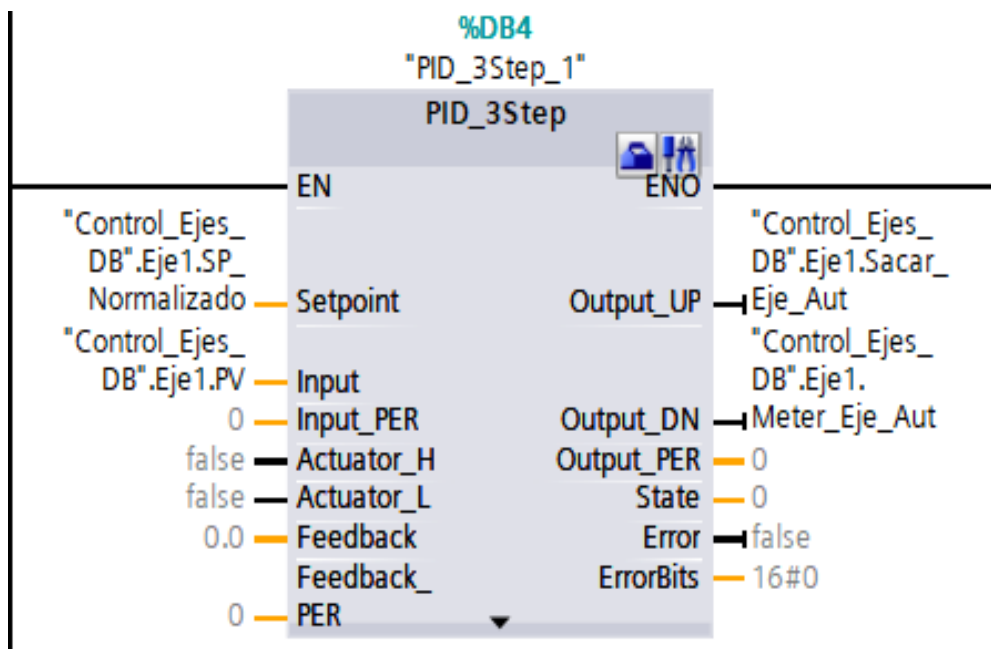


Ilustración 51. DB resultante de la instrucción PID_3Step V2.

Desde la aplicación móvil se realiza un entrenamiento variando la intensidad del recorrido. A medida que varíe la intensidad del recorrido, el eje eléctrico va modificando su posición continuamente y el PID estará regulando todo el tiempo para conseguir la posición más exacta posible en el menor tiempo posible (tiempo de respuesta menor y mayor estabilidad del proceso):

Para terminar, se explican los parámetros más importantes de este DB:

EN: entrada de habilitación.

ENO: la salida de habilitación ENO se activa únicamente si el estado lógico de la entrada de habilitación EN es "1" y no ocurren errores al ejecutar la operación.

Setpoint: valor de posición que se quiere conseguir en formato REAL. En este parámetro del PID se introduce la variable de *set point* normalizado, para que el PID no lea valores incorrectos, es decir, este *set point* solo acepta valores que se encuentran en el rango del recorrido del eje eléctrico [0,35] mm.

Input: valor actual de la posición del eje eléctrico en formato REAL.

Error: se activa si se produce un fallo en el proceso de regulación del PID.

Output_UP: valor de control para sacar el eje eléctrico. Es de tipo BOOL.

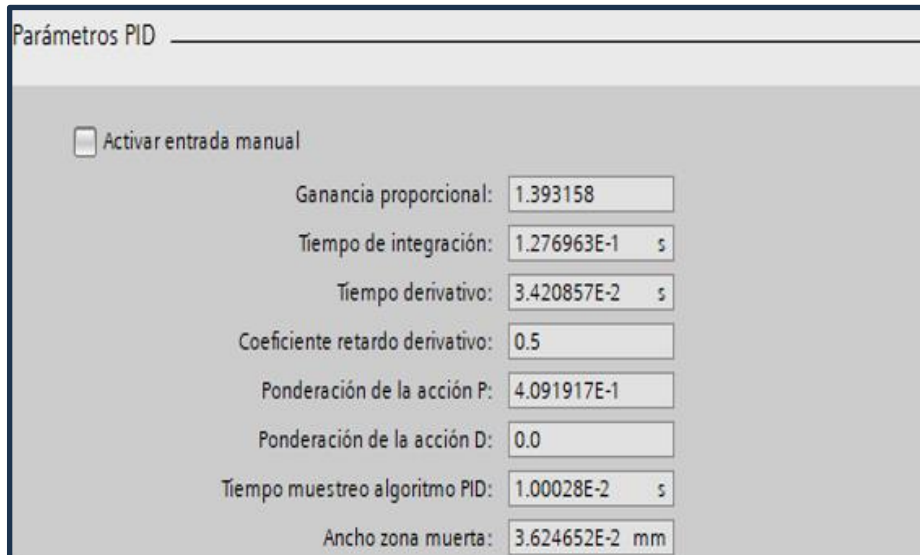
Output_DN: valor de control para meter el eje eléctrico. Es de tipo BOOL.

Input_PER: valor entero de corriente en la entrada periférica.

3.2.7 PUESTA EN SERVICIO PID_3STEP

La puesta en servicio del regulador PID permite configurar el controlador de modo que se autoajuste al arrancar y durante el funcionamiento.

Para optimizar el lazo PID se selecciona el modo de ajuste "Optimización inicial". Se ha utilizado un tiempo de muestreo de 0,3 s. Una vez seleccionado el tiempo de muestreo, se hace clic en el botón "Iniciar". El regulador empieza a pasar por diferentes fases ,para calcular la respuesta del sistema y los tiempos de actualización, obteniendo los parámetros de optimización adecuados. Una vez terminada la optimización, se guardan los parámetros del PID (ver ilustración 52).



Parámetros PID

Activar entrada manual

Ganancia proporcional: 1.393158

Tiempo de integración: 1.276963E-1 s

Tiempo derivativo: 3.420857E-2 s

Coefficiente retardo derivativo: 0.5

Ponderación de la acción P: 4.091917E-1

Ponderación de la acción D: 0.0

Tiempo muestreo algoritmo PID: 1.00028E-2 s

Ancho zona muerta: 3.624652E-2 mm

Ilustración 52. Parámetros del PID.

Se utilizan las señales de tope del actuador para la realimentación de posición. Por último, se muestra el comportamiento del regulador introduciendo un *set point* (ver ilustración 53). El regulador se activa y consigue la salida deseada con un tiempo de respuesta óptimo , debido al *offset* utilizado en el DB de control del eje eléctrico, sin inestabilidad existente durante el proceso (ver ilustración 54).

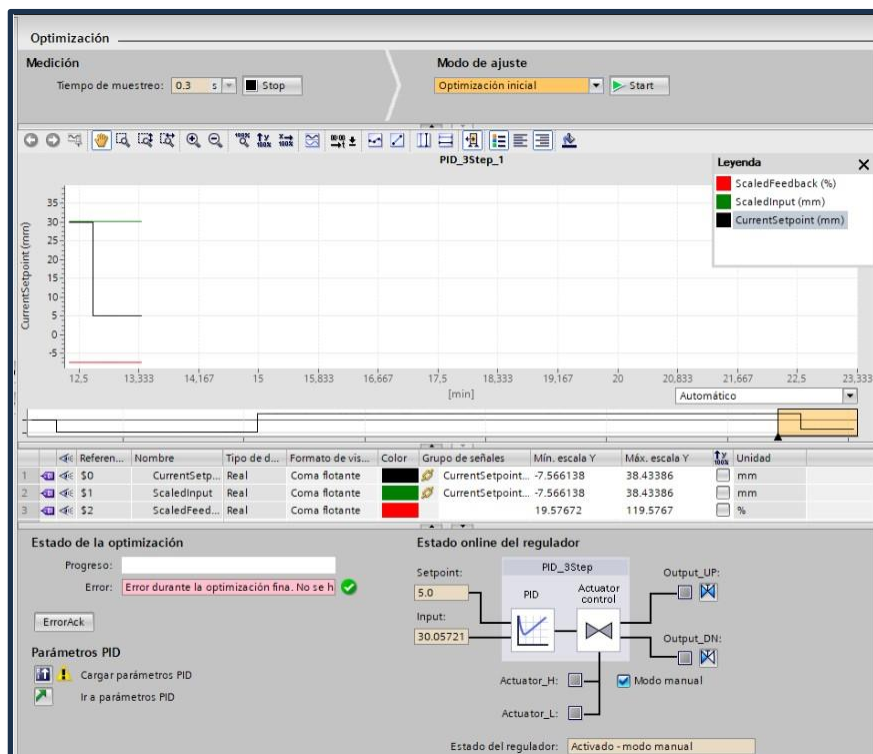


Ilustración 53. Puesta en servicio PID_3STEP.

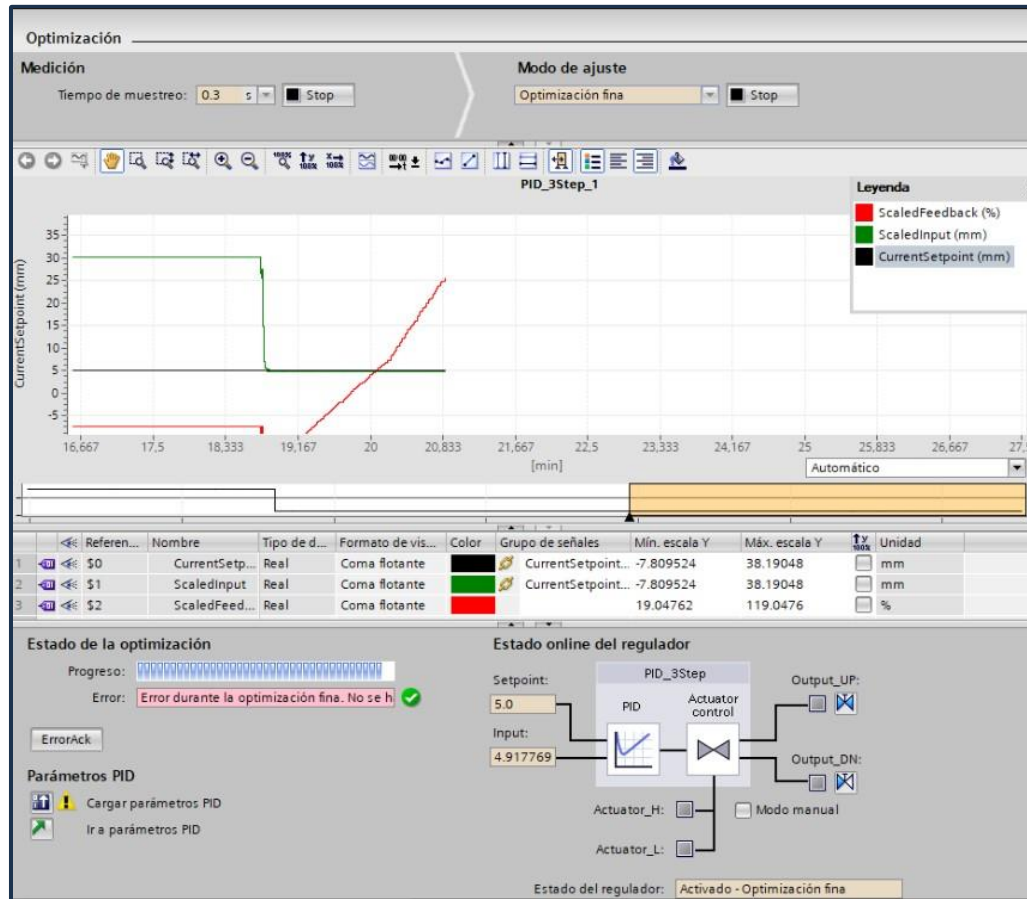


Ilustración 54. Comportamiento del PID para la obtención de la salida deseada.

4. MODO DE OPERACIÓN DEL WIFI TP-LINK

Para comenzar la configuración el WIFI Tp-Link CPE510 tiene que estar alimentado a 24 V. Una vez alimentado el dispositivo, desde el ordenador se conecta a la red "TP-LINK_CFCA". Una vez conectada, se ingresa en internet la dirección IP que tiene el dispositivo *router* (ver ilustración 55):

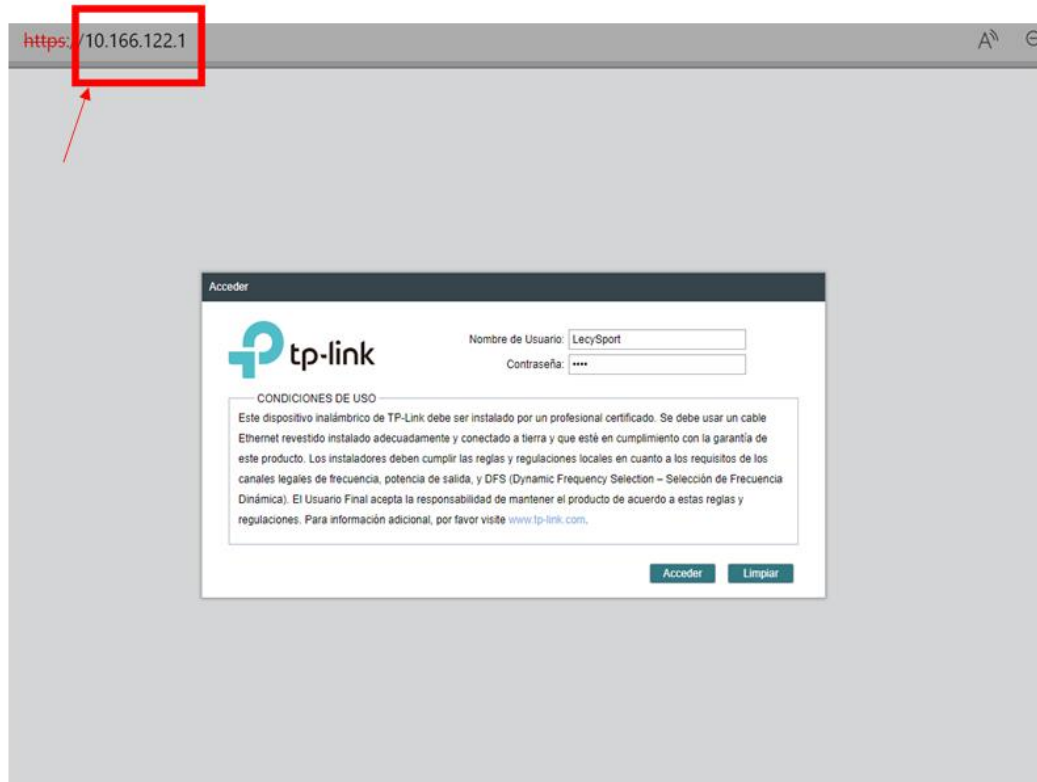


Ilustración 55. Conexión red TP-LINK_CFCA.

Cuando se introduce el nombre de usuario y la contraseña, se abre la configuración interna del TP-LINK CPE510. Antes de configurar el dispositivo, se puede visualizar la información del dispositivo destacada como la versión del *Firmware*, el modelo del Dispositivo, CPU y la capacidad de la memoria (ver ilustración 56).

Información del Dispositivo

Nombre del Dispositivo: CPE510

Modelo del Dispositivo: CPE510 v3.20

Versión de Firmware: 2.2.2 Build 20201111 Rel. 58267 (4555)

Hora del sistema: 2020-05-05 09:06:19

Tiempo de Activación: 0 días 01:07:13

CPU:  1%

Memoria:  69%

Ilustración 56. Información del dispositivo router.

Para la realización de este proyecto, el dispositivo *router* puede configurarse como Punto de Acceso y cliente WISP. A continuación, se justifica el modo de funcionamiento utilizado finalmente.

WISP son las siglas de *Wireless Internet Service Provider*. Esta configuración aporta conexión a Internet de forma inalámbrica aportando una serie de ventajas importantes. Muchos usuarios solo pueden optar por seleccionar esta configuración para conectarse a una red debido a la simplicidad que nos ofrece.

Hay que indicar que lo normal, en la actualidad, es contratar un servicio de Internet de fibra óptica para el hogar (punto de acceso). Ofrece velocidades simétricas de hasta 1 Gbps, buena estabilidad, baja latencia y, en definitiva, un servicio que se adapta normalmente a cualquier usuario.

Sin embargo, no siempre es posible llevar a cabo la instalación. En nuestro caso la infraestructura no es la mejor, ya que el proyecto se desea instalar en un campo de fútbol o en otro tipo de instalación seleccionada por el usuario.

Para evitar el problema derivado de la instalación, son importantes los WISP. Su misión es ofrecer conexión de forma inalámbrica a los hogares o cualquier lugar donde se necesite conexión. Todo esto sin necesidad de realizar una instalación con cables, como ocurre con la fibra óptica.

Esto es muy interesante para el proyecto, ya que no se sabe dónde se va a posicionar el *router* dentro del campo de entrenamiento. Esta tecnología se basa en antenas o estaciones. Además, en la vivienda del usuario se instala otra antena más pequeña que se conecta con la estación. De esta forma se puede tener conexión a Internet de forma inalámbrica, sin llevar a cabo una compleja instalación.

Generalmente, la velocidad que ofrecen los WISP va variando entre 6 y 50 Mbps. Suelen ofrecerlo de forma simétrica, por lo que tendremos lo mismo de bajada que de subida. El uso de la nube, por ejemplo, hace que sea necesario tener una buena velocidad de subida.

Se puede decir, por tanto, que intervienen dos elementos: una antena mayor (del proveedor), la que ofrece conexión a los clientes, y una antena menor que instala el usuario en su vivienda. Con esta segunda antena se puede repartir la señal a los diferentes dispositivos que estén dentro del alcance de nuestro dispositivo.

Por todo lo comentado en este punto, se decidió seleccionar el modo "Ap Router Cliente" o Cliente WISP (ver ilustración 57). En el anexo 3 se puede visualizar la configuración completa de este dispositivo.

CONFIGURACIÓN RÁPIDA	ESTADO	RED	INALÁMBRICO	ADMINISTRACIÓN	SISTEMA
Modo de Operación					
Por favor seleccione el Modo de Operación adecuado de acuerdo a sus necesidades:					
<input type="radio"/>	Punto de Acceso	En este modo, el AP actuará como un hub central para diferentes clientes de redes locales inalámbricas. El Multi-SSID también está disponible en este modo, el cual soporta hasta 4 diferentes SSIDs y contraseñas.			
<input type="radio"/>	Cliente	Con el modo cliente, el dispositivo puede conectarse a un dispositivo por cable y funciona como un adaptador inalámbrico para recibir la señal inalámbrica de su red inalámbrica.			
<input type="radio"/>	Repetidor	En este modo, el dispositivo puede copiar y reforzar la señal inalámbrica existente para extender la cobertura de la señal, especialmente para eliminar las esquinas sin señal.			
<input type="radio"/>	Puente	El modo puente pide prestado el Internet inalámbrico existente y lo retransmite usando un SSID y contraseñas diferentes. En este modo, configura un cliente inalámbrico para conectarlo con el AP raíz para la cobertura inalámbrica local.			
<input type="radio"/>	Router AP	En este modo el dispositivo activa múltiples usuarios para compartir Internet. Los puertos inalámbricos comparten el mismo IP para ISP a través del Puerto WAN Ethernet. El puerto Inalámbrico actúa igual que un Puerto LAN al mismo tiempo que en el modo AP Router.			
<input checked="" type="radio"/>	AP Router Cliente (Cliente WISP)	En este modo el dispositivo activa múltiples usuarios para compartir Internet desde WISP. Los dispositivos del puerto LAN comparten el mismo IP desde WISP a través del puerto Inalámbrico. Mientras se conecta a WISP, el puerto Inalámbrico funciona como un Puerto WAN en el modo de Cliente AP Router. El puerto Ethernet actúa como puerto LAN.			

Ilustración 57. Modo de operación utilizado por el WIFI TP_LINK.

5. COMUNICACIÓN APP - PLC (WEB SERVER)

Para realizar la comunicación entre la aplicación y el PLC, en primera instancia, se pensó en la tecnología de comunicación multiplataforma OPC UA (Arquitectura Unificada de Comunicaciones de Plataforma Abierta). Esta tecnología de comunicación es la evolución de la tecnología OPC Clásica orientada a servicios determinados.

Se trata de un protocolo de comunicaciones pensado para comunicar datos de equipos industriales, pero su principal diferencia es que, a diferencia del OPC Clásico, no se limita solo a comunicar datos entre aplicaciones SCADA y sensores, sino que su objetivo es ir más allá y que pueda comunicarse con todas las aplicaciones de la empresa y a través de todas las capas empresariales, es decir, permite llevar datos de máquinas donde te imagines, de forma segura y multiplataforma. Aplicaciones en el móvil, programas de gestión empresarial, MES, entre otros.

El inconveniente que se encuentra para utilizar este protocolo de comunicaciones se debe a la necesidad de crear un interfaz para utilizar el protocolo OPC UA en Tia Portal (ver ilustración 58) (SIEMENS, Modelado de la interfaz del servidor S7-1200 OPC UA en el TIA Portal, 2020). Con un S7-1500 se puede acceder directamente al dato ya que todas las variables tienen su propio nodo, sin necesidad de un interfaz, simplificando el proceso.

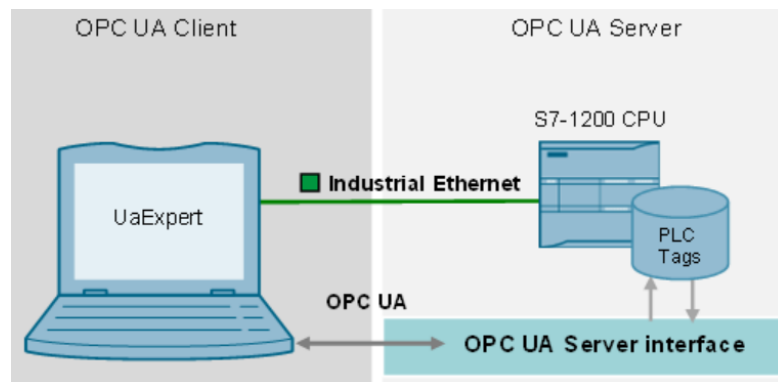


Ilustración 58. . Escenario de configuración del protocolo OPC UA de un S7-1200.

También se buscó información sobre el protocolo MQTT (MQ Telemetry Transport). Es un servicio de mensajería *push* con patrón publicador/suscriptor (*pub-sub*) donde los clientes se conectan con un servidor central denominado bróker (ver ilustración 59) (MQTT, 2019).

Para filtrar los mensajes que son enviados a cada cliente, los mensajes se encuentran en *topics* organizados jerárquicamente. Un cliente puede publicar un mensaje en un determinado *topic*. Otros clientes pueden suscribirse a este *topic*, y el *bróker* le hará llegar los mensajes suscritos.

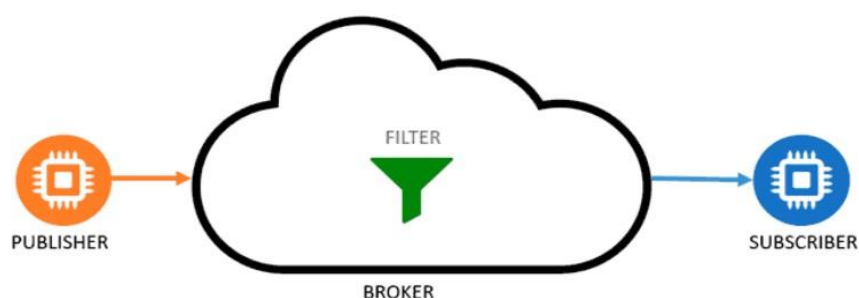


Ilustración 59. Funcionamiento protocolo MQTT.

Los clientes inician una conexión TCP/IP con el bróker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantendrá abierta hasta que el cliente finalice la comunicación.

Al igual que la OPC-UA, tiene el mismo problema, se necesita montar un servidor/interfaz (en este caso bróker) para la comunicación.

Después de pensar mucho el protocolo de comunicaciones que se iba a utilizar, se optó por utilizar un servidor web que ofrece el PLC de Siemens que permite utilizar el protocolo HTML para el intercambio de datos simplificando mucho la aplicación (SIEMENS, S7-1200 servidor web, 2012).

Este servidor web que ofrece el PLC permite que los usuarios autorizados monitoricen y administren la CPU a través de una red; esto permite llevar a cabo evaluaciones y diagnósticos salvando grandes distancias.

En el estado de suministro de la CPU el servidor web está desactivado. Tan solo después de cargar un proyecto en el que esté activado el servidor web es posible el acceso a través del navegador web. Para ello, necesitamos realizar la configuración del servidor web. La configuración del servidor web se encuentra en el Anexo 4.

Una vez terminada la configuración del servidor web, STEP 7 genera la instrucción WWW al pulsar "Generar bloques" (ver ilustración 60). La instrucción WWW inicializa el servidor web de la CPU o sincroniza las páginas de usuario con el programa de usuario en la CPU. El DB Web de Control es el parámetro de entrada para la instrucción WWW e indica el contenido de las páginas tal como se representan en los DB de fragmento, así como informaciones de estado y de control.

El programa de usuario debe ejecutar la instrucción WWW para que pueda accederse a las páginas de usuario en el servidor web. En este proyecto se encuentran en el Main [OB1]:

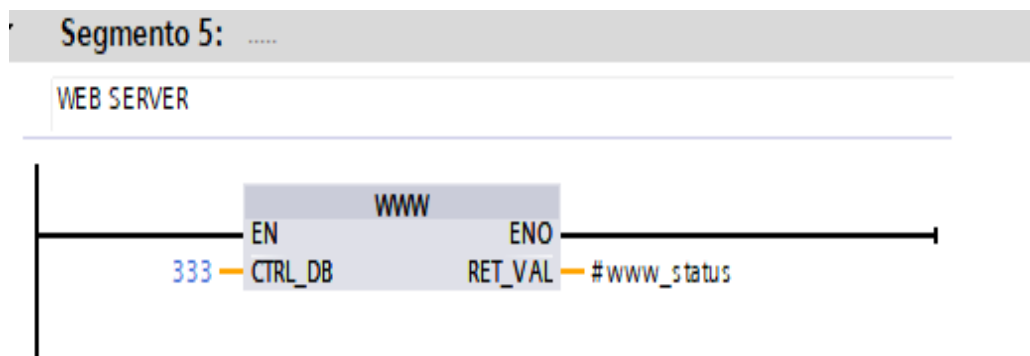


Ilustración 60. Instrucción del servidor web.

A continuación, se van a explicar los parámetros de esta instrucción:

- **CTRL_DB:** Entrada tipo DB_WWW que describe las páginas de usuario (DB Web Control).
- **RET_VAL:** Salida tipo entero que nos informará de los posibles errores que pueden cometerse. En la siguiente tabla se muestran los códigos de error que puede mostrar este parámetro con su explicación (ver tabla 12).



Código de error (W#16#...)	Explicación
0000	No se han producido errores. No existen solicitudes de página web que deban ser autorizadas por el programa de usuario.
00xy	x: muestra si durante la inicialización del DB Web Control (CTRL_DB) se ha producido un error: x=0: no se han producido errores. x=1: se han producido errores. El error está codificado en el byte "CTRL_DB.last_error" del DB Web Control. y: número de la petición pendiente. Son posibles varias peticiones (p. ej. si están pendientes las peticiones "0" y "1": y="3"). y="1": petición "0" y="2": petición "1" y="4": petición "2" y="8": petición "3"
803A	El DB Web Control indicado no está disponible en la CPU.
8081	Versión o formato erróneos del DB Web Control.
80C1	No hay recursos disponibles para inicializar la aplicación web.

Tabla 12. Códigos de error de la instrucción del servidor web.

En este proyecto, se han empleado tres archivos *HTML* para la comunicación entre el autómatas y la aplicación móvil:

- Consulta.html: En este archivo el PLC enviará la distancia (escalada) adaptada a JSON (*JavaScript Object Notation*), una representación textual sin patrones de datos estructurados basados en listas ordenadas, a la aplicación móvil. JSON se utiliza comúnmente, pero no exclusivamente, para intercambiar información entre clientes y servidores web.
- Entrada.html: En este archivo se mandan todas las variables desde la aplicación móvil al PLC. Concretamente los *Set Points*.
- Salida.html: Al igual que en consulta.html, en este archivo el PLC enviará las variables restantes a la aplicación móvil. En este caso, no estará adaptado a JSON.

Hay que remarcar que el desarrollo de la aplicación móvil y los archivos *HTML* los ha realizado una empresa de desarrollo de aplicaciones web. Para programar los archivos HTML, explicados anteriormente, han utilizado el lenguaje de programación Java de Android, un lenguaje que posee algunas librerías distintas al java de escritorio, pero son muy similares.

6. CONCLUSIONES

A lo largo del proyecto se ha tenido presente el principal motivo para la realización de este: la adquisición de aprendizaje a través de establecer la comunicación entre un proceso de automatización industrial y una aplicación móvil; este aprendizaje adquirido una vez terminado el proyecto sirve para futuros proyectos que se quieran realizar, ya que no solo se ha adquirido la parte positiva del proyecto, sino también los errores.

Tras la planificación y el seguimiento de las fases, una vez terminado el proyecto, la conclusión general a la que se llega es que se han cumplido los objetivos del proyecto con el correcto funcionamiento de este; tras el desarrollo del programa del PLC para realizar los procesos de escalado y de control de eje eléctrico y el *encoder*, el diseño y ajuste del controlador PID, la creación de tipo de datos para la comunicación, el conexionado físico del PLC con el eje eléctrico y el *encoder*, se pasó a la realización de pruebas. Se comprobó el correcto funcionamiento de los componentes integrados en el proyecto, también se realizó una prueba de la máquina real con sus diferentes modos de funcionamiento (manual y automático).

6.1 POSIBLES MEJORAS Y FUTURO DEL PROYECTO.

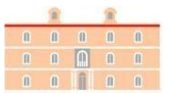
Desde la parte más técnica, se podría haber utilizado un controlador S7-1500 para utilizar el protocolo de comunicación OPC UA sin necesidad de crear un interfaz/servidor, simplificando así el proceso de comunicación entre el PLC y la aplicación móvil.

Si se pone el foco en modificar la resistencia o fuerza de una máquina de entrenamiento desde la aplicación móvil, podría estudiarse la posibilidad de que se pudiese adaptar la aplicación y el funcionamiento a cualquier máquina de un centro deportivo o, incluso, a más de una máquina a la vez. Se podría sustituir el peso físico de una máquina (diferentes pesas que constituyen una máquina convencional de gimnasio) por el sistema utilizado en esta máquina, adaptando está a las necesidades personales, por ejemplo, si una máquina de un centro de entrenamiento tuviese una pantalla HMI donde el usuario pudiese introducir automáticamente un peso (60kg), iniciar el ejercicio, terminar y volver a cambiar de peso (50kg), este usuario no tendría que realizar manualmente el cambio de peso sustituyendo un material físico por otro, así economizaría su tiempo, ahorrando entre ejercicio y ejercicio.

Yendo un poco más allá, desde la misma pantalla HDMI se podría barajar la posibilidad de monitorizar al usuario sus constantes vitales durante el desarrollo de cada entrenamiento, con el fin de prevenir cualquier tipo de accidente, mediante el establecimiento previo de los límites del usuario, por ejemplo, si sus pulsaciones haciendo ejercicio normalmente no suben de 120 lpm (latidos por minuto), establecería como límite 120 lpm, si la máquina detectase 123 lpm se activaría una alerta y la máquina se pararía.

7. BIBLIOGRAFÍA

- AG, S. (2024). *datasheet SM1231*. Obtenido de datasheet SM1231:
<https://media.distributordatasolutions.com/siemenscomplete/2021q2/documents/53a7c2b12e27dffbca90bef78066612dfe6ce899.pdf>
- Contact, P. (2024). *UTT B 2,5 - Borna de doble piso*. Obtenido de UTTB 2,5 - Borna de doble piso:
https://www.phoenixcontact.com/product/pdf/api/v1/MzA0NDYzNg?_realm=es&_locale=es-ES&blocks=technical-data%2Cdrawings
- Contact, P. (2024). *UTT B 2,5 - Borna de doble piso*. Obtenido de UTTB 2,5 - Borna de doble piso:
https://www.phoenixcontact.com/product/pdf/api/v1/MzA0NDYzNg?_realm=es&_locale=es-ES&blocks=technical-data%2Cdrawings
- Electric, S. (13 de Octubre de 2014). *RSL1AB4BD*. Obtenido de RSL1AB4BD:
<https://www.se.com/es/es/product/RSL1AB4BD/zelio-relay-rel%C3%A9-1nanc-6a-24-vdc/>
- electric, S. (12 de Enero de 2021). *RSLZVA1*. Obtenido de RSLZVA1:
<https://media.distributordatasolutions.com/schneider2/2021q1/documents/747a4f5132ac4e32e430e1d758ad193b02aca1ef.pdf>
- LINAK. (s.f.). *Actuador LA14*. Obtenido de Actuador LA14: <https://cdn.linak.com/-/media/files/data-sheet-source/en/linear-actuator-la14-data-sheet-eng.pdf>
- Llamas, L. (19 de Abril de 2019). *MQTT*. Obtenido de MQTT: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- OMRON. (s.f.). *Fuente de alimentacion conmutada S8PS*. Obtenido de Fuente de alimentacion conmutada S8PS:
<https://docs.rs-online.com/7455/0900766b805308b8.pdf>
- shields, I. (s.f.). *Two-phase AB incremental rotary encoder, 600P / R, 6 mm shaft (DC5-24Vdc)*. Obtenido de Two-phase AB incremental rotary encoder, 600P / R, 6 mm shaft (DC5-24Vdc):
https://www.industrialshields.com/es_ES/shop/210001000100-two-phase-ab-incremental-rotary-encoder-600p-r-6-mm-shaft-dc5-24vdc-934#attr=3672
- SIEMENS. (2009). *simatic_s71200*. Obtenido de simatic_s71200:
https://cache.industry.siemens.com/dl/files/875/39644875/att_76197/v1/s71200_getting_started_es-ES_es-ES.pdf
- SIEMENS. (Abril de 2012). *S7-1200 servidor web*. Obtenido de S7-1200 servidor web:
https://cache.industry.siemens.com/dl/files/622/91696622/att_42774/v1/s71200_system_manual_es-ES_es-ES.pdf
- SIEMENS. (24 de Noviembre de 2020). *Modelado de la interfaz del servidor S7-1200 OPC UA en el TIA Portal*. Obtenido de Modelado de la interfaz del servidor S7-1200 OPC UA en el TIA Portal:
<https://support.industry.siemens.com/cs/document/109781701/modelado-de-la-interfaz-del-servidor-s7-1200-opc-ua-en-el-tia-portal?dti=0&lc=es-PA>
- SIEMENS. (s.f.). *Controlador programable S7-1200*. Obtenido de
https://cache.industry.siemens.com/dl/files/622/91696622/att_42774/v1/s71200_system_manual_es-ES_es-ES.pdf
- TP-LINK. (s.f.). *CPE de Exterior de 13dBi en 5GHz a 300Mbps*. Obtenido de CPE de Exterior de 13dBi en 5GHz a 300Mbps: <https://www.tp-link.com/es/business-networking/outdoor-radio/cpe510/#overview>



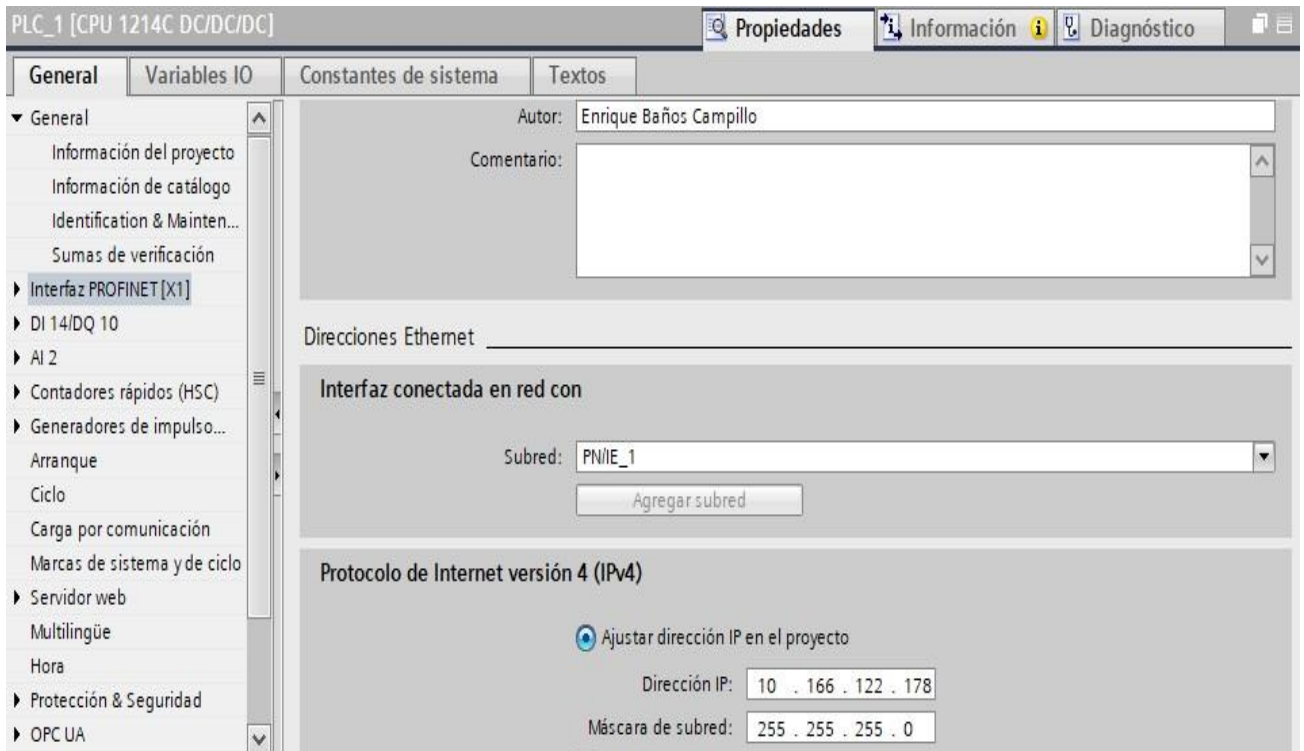
8. ANEXOS

En este apartado se mostrarán los distintos documentos que se han utilizado para la realización del proyecto, entre los que se encuentran:

- Programación PLC.
- Presupuesto.
- Configuración completa WIFI TP-LINK.
- Configuración un servidor web en TIA Portal.
- Planos con EPLAN 2.7.
- Conexión del hardware exterior.

8.1 PROGRAMACIÓN PLC

8.1.1 Información general del proyecto



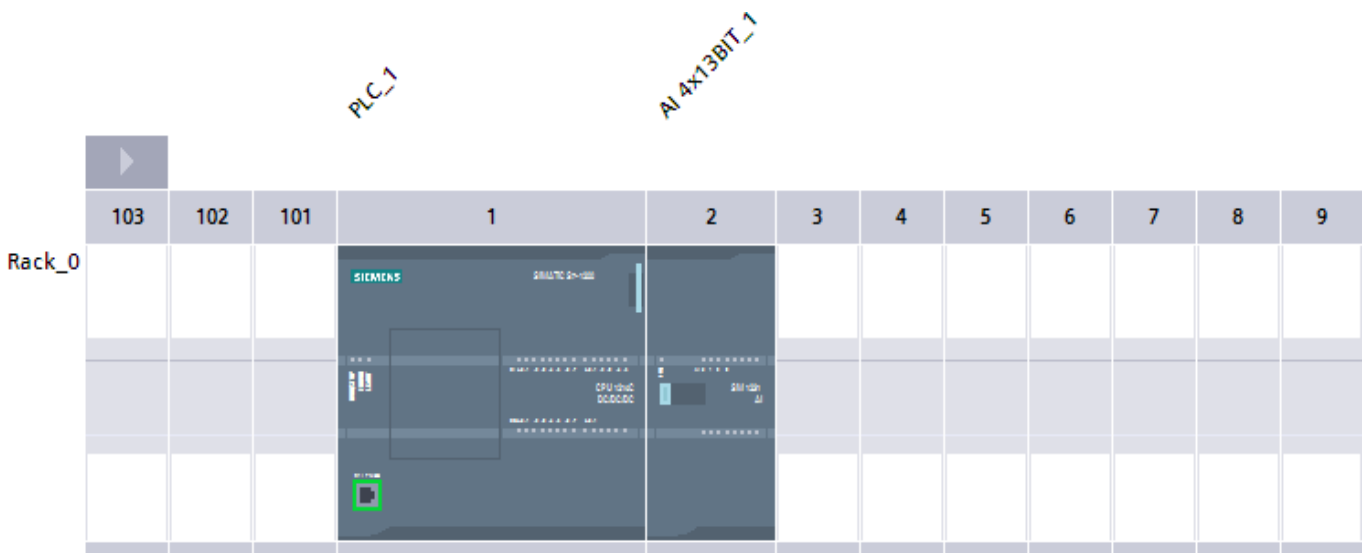
PLC_1 [CPU 1214C DC/DC/DC]

Autor: Enrique Baños Campillo
 Comentario:

Direcciones Ethernet

Interfaz conectada en red con
 Subred: PN/IE_1

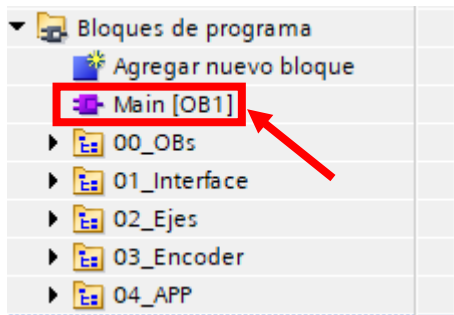
Protocolo de Internet versión 4 (IPv4)
 Ajustar dirección IP en el proyecto
 Dirección IP: 10 . 166 . 122 . 178
 Máscara de subred: 255 . 255 . 255 . 0



8.1.2 Tabla de variables estándar

Tabla de variables estándar				
		Nombre	Tipo de datos	Dirección
1		Out_Sacar_Eje	Bool	%Q0.1
2		Out_Meter_Eje	Bool	%Q0.0
3		I_Analogica 1_0-10v	Int	%IW64
4		System_Byte	Byte	%MB1
5		FirstScan	Bool	%M1.0
6		DiagStatusUpdate	Bool	%M1.1
7		Always TRUE	Bool	%M1.2
8		Always FALSE	Bool	%M1.3
9		Clock_Byte	Byte	%MB0
10		Clock_10Hz	Bool	%M0.0
11		Clock_5Hz	Bool	%M0.1
12		Clock_2.5Hz	Bool	%M0.2
13		Clock_2Hz	Bool	%M0.3
14		Clock_1.25Hz	Bool	%M0.4
15		Clock_1Hz	Bool	%M0.5
16		Clock_0.625Hz	Bool	%M0.6
17		Clock_0.5Hz	Bool	%M0.7
18		HSC1	DInt	%ID1000
19		I_Analogica2_0-10v	Int	%IW66
20		HSC2	DInt	%ID1004
21		HSC3	DInt	%ID1008
22		HSC4	DInt	%ID1012
23		Out_Meter_Eje_2	Bool	%Q0.2
24		Out_Sacar_Eje_2	Bool	%Q0.3
25		Analogica 1	Int	%IW96
26		Analogica 2	Int	%IW98
27		Analogica 3	Int	%IW100
28		Analogica 4	Int	%IW102
29		Out_Meter_Eje_3	Bool	%Q0.4
30		Out_Meter_Eje_4	Bool	%Q0.6
31		Out_Sacar_Eje_3	Bool	%Q0.5
32		Out Sacar Eje 4	Bool	%Q0.7

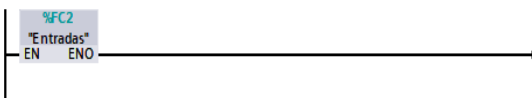
8.1.3 Bloques de programa



▼ Título del bloque: "Main Program Sweep (Cycle)"
MAIN [OB1]

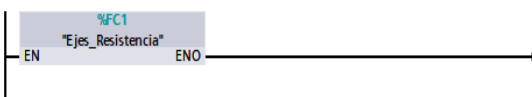
▼ Segmento 1:

ENTRADAS



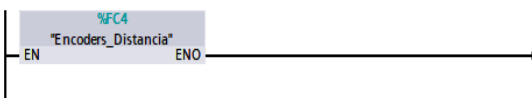
▼ Segmento 2:

EJES_RESISTENCIA



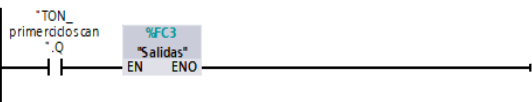
▼ Segmento 3:

ENCODERS_DISTANCIA



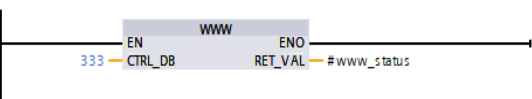
▼ Segmento 4:

SALIDAS



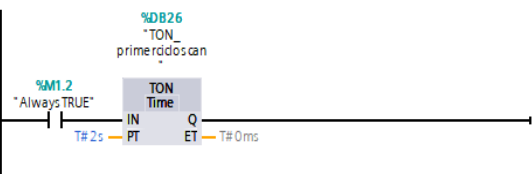
▼ Segmento 5:

FUNCTION BLOCK QUE NECESITAMOS PARA HABILITAR EL WEB SERVER



▼ Segmento 6:

TON_PRIMER_CICLO_SCAN



Bloques de programa
 Agregar nuevo bloque
 Main [OB1]
 00_OBs
Hardware interrupt [OB.]
 01_Interface
 02_Ejes
 03_Encoder
 04_APP

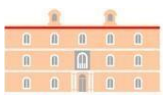
Título del bloque: FUNCIÓN CREADA POR EL TIA PORTAL
 Comentario

Segmento 1:
 Comentario

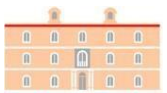
Bloques de programa
 Agregar nuevo bloque
 Main [OB1]
 00_OBs
 01_Interface
Entradas [FC2]
 Salidas [FC3]
 02_Ejes
 03_Encoder
 04_APP

```

1 //EJES ELECTRICOS
2
3 //ESCALADO EJE1
4 "Escalado_Ejes_DB".Eje1.Analogica := "Analogica1";
5 "Escalado_Ejes_DB".Eje1.Min := "APP".Estacion1.Esc_Eje.in.Min;
6 "Escalado_Ejes_DB".Eje1.Max := "APP".Estacion1.Esc_Eje.in.Max;
7 "Escalado_Ejes_DB".Eje1.Min_Fisico := INT_TO_REAL("APP".Estacion1.Esc_Eje.in.Min_Fisico)/100.00;
8 "Escalado_Ejes_DB".Eje1.Max_Fisico := INT_TO_REAL("APP".Estacion1.Esc_Eje.in.Max_Fisico)/100.00;
9 "Escalado_Ejes_DB".Eje1.Min_Escalado := INT_TO_REAL("APP".Estacion1.Esc_Eje.in.Min_Escalado)/100.00;
10 "Escalado_Ejes_DB".Eje1.Max_Escalado := INT_TO_REAL("APP".Estacion1.Esc_Eje.in.Max_Escalado)/100.00;
11
12 //ESCALADO EJE2
13 "Escalado_Ejes_DB".Eje2.Analogica := "Analogica2";
14 "Escalado_Ejes_DB".Eje2.Min := "APP".Estacion2.Esc_Eje.in.Min;
15 "Escalado_Ejes_DB".Eje2.Max := "APP".Estacion2.Esc_Eje.in.Max;
16 "Escalado_Ejes_DB".Eje2.Min_Fisico := INT_TO_REAL("APP".Estacion2.Esc_Eje.in.Min_Fisico)/100.00;
17 "Escalado_Ejes_DB".Eje2.Max_Fisico := INT_TO_REAL("APP".Estacion2.Esc_Eje.in.Max_Fisico)/100.00;
18 "Escalado_Ejes_DB".Eje2.Min_Escalado := INT_TO_REAL("APP".Estacion2.Esc_Eje.in.Min_Escalado)/100.00;
19 "Escalado_Ejes_DB".Eje2.Max_Escalado := INT_TO_REAL("APP".Estacion2.Esc_Eje.in.Max_Escalado)/100.00;
20
21 //ESCALADO EJE3
22 "Escalado_Ejes_DB".Eje3.Analogica := "Analogica3";
23 "Escalado_Ejes_DB".Eje3.Min := "APP".Estacion3.Esc_Eje.in.Min;
24 "Escalado_Ejes_DB".Eje3.Max := "APP".Estacion3.Esc_Eje.in.Max;
25 "Escalado_Ejes_DB".Eje3.Min_Fisico := INT_TO_REAL("APP".Estacion3.Esc_Eje.in.Min_Fisico)/100.00;
26 "Escalado_Ejes_DB".Eje3.Max_Fisico := INT_TO_REAL("APP".Estacion3.Esc_Eje.in.Max_Fisico)/100.00;
27 "Escalado_Ejes_DB".Eje3.Min_Escalado := INT_TO_REAL("APP".Estacion3.Esc_Eje.in.Min_Escalado)/100.00;
28 "Escalado_Ejes_DB".Eje3.Max_Escalado := INT_TO_REAL("APP".Estacion3.Esc_Eje.in.Max_Escalado)/100.00;
  
```



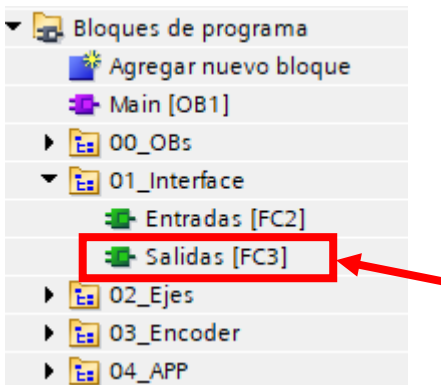
```
29
30 //ESCALADO EJE4
31 "Escalado_Ejes_DB".Eje4.Analogica := "Analogica4";
32 "Escalado_Ejes_DB".Eje4.Min := "APP".Estacion4.Esc_Eje.in.Min;
33 "Escalado_Ejes_DB".Eje4.Max := "APP".Estacion4.Esc_Eje.in.Max;
34 "Escalado_Ejes_DB".Eje4.Min_Fisico := INT_TO_REAL("APP".Estacion4.Esc_Eje.in.Min_Fisico)/100.00;
35 "Escalado_Ejes_DB".Eje4.Max_Fisico := INT_TO_REAL("APP".Estacion4.Esc_Eje.in.Max_Fisico)/100.00;
36 "Escalado_Ejes_DB".Eje4.Min_Escalado := INT_TO_REAL("APP".Estacion4.Esc_Eje.in.Min_Escalado)/100.00;
37 "Escalado_Ejes_DB".Eje4.Max_Escalado := INT_TO_REAL("APP".Estacion4.Esc_Eje.in.Max_Escalado)/100.00;
38
39 //CONTROL EJE1
40
41 "Control_Ejes_DB".Eje1.Habilitar := INT_TO_BOOL("APP".Estacion1.Eje.in.Habilitar);
42 "Control_Ejes_DB".Eje1.SP := INT_TO_REAL("APP".Estacion1.Eje.in.SP)/100.00;
43 "Control_Ejes_DB".Eje1.Offset := INT_TO_REAL("APP".Estacion1.Eje.in.Offset)/100.00;
44 "Control_Ejes_DB".Eje1.Sacar_Eje_Man := INT_TO_BOOL("APP".Estacion1.Eje.in.Sacar_Eje_Man);
45 "Control_Ejes_DB".Eje1.Meter_Eje_Man := INT_TO_BOOL("APP".Estacion1.Eje.in.Meter_Eje_Man);
46 "Control_Ejes_DB".Eje1.Ton_Alarma := INT_TO_TIME("APP".Estacion1.Eje.in.Ton_Alarma);
47
48 //CONTROL EJE2
49
50 "Control_Ejes_DB".Eje2.Habilitar := INT_TO_BOOL("APP".Estacion2.Eje.in.Habilitar);
51 "Control_Ejes_DB".Eje2.SP := INT_TO_REAL("APP".Estacion2.Eje.in.SP)/100.00;
52 "Control_Ejes_DB".Eje2.Offset := INT_TO_REAL("APP".Estacion2.Eje.in.Offset)/100.00;
53 "Control_Ejes_DB".Eje2.Sacar_Eje_Man := INT_TO_BOOL("APP".Estacion2.Eje.in.Sacar_Eje_Man);
54 "Control_Ejes_DB".Eje2.Meter_Eje_Man := INT_TO_BOOL("APP".Estacion2.Eje.in.Meter_Eje_Man);
55 "Control_Ejes_DB".Eje2.Ton_Alarma := INT_TO_TIME("APP".Estacion1.Eje.in.Ton_Alarma);
56
57 //CONTROL EJE3
58
59 "Control_Ejes_DB".Eje3.Habilitar := INT_TO_BOOL("APP".Estacion3.Eje.in.Habilitar);
60 "Control_Ejes_DB".Eje3.SP := INT_TO_REAL("APP".Estacion3.Eje.in.SP)/100.00;
61 "Control_Ejes_DB".Eje3.Offset := INT_TO_REAL("APP".Estacion3.Eje.in.Offset)/100.00;
62 "Control_Ejes_DB".Eje3.Sacar_Eje_Man := INT_TO_BOOL("APP".Estacion3.Eje.in.Sacar_Eje_Man);
63 "Control_Ejes_DB".Eje3.Meter_Eje_Man := INT_TO_BOOL("APP".Estacion3.Eje.in.Meter_Eje_Man);
64 "Control_Ejes_DB".Eje3.Ton_Alarma := INT_TO_TIME("APP".Estacion1.Eje.in.Ton_Alarma);
65
66 //CONTROL EJE4
67
68 "Control_Ejes_DB".Eje4.Habilitar := INT_TO_BOOL("APP".Estacion4.Eje.in.Habilitar);
69 "Control_Ejes_DB".Eje4.SP := INT_TO_REAL("APP".Estacion4.Eje.in.SP)/100.00;
70 "Control_Ejes_DB".Eje4.Offset := INT_TO_REAL("APP".Estacion4.Eje.in.Offset)/100.00;
71 "Control_Ejes_DB".Eje4.Sacar_Eje_Man := INT_TO_BOOL("APP".Estacion4.Eje.in.Sacar_Eje_Man);
72 "Control_Ejes_DB".Eje4.Meter_Eje_Man := INT_TO_BOOL("APP".Estacion4.Eje.in.Meter_Eje_Man);
73 "Control_Ejes_DB".Eje4.Ton_Alarma := INT_TO_TIME("APP".Estacion1.Eje.in.Ton_Alarma);
74
75 //ENCODERS DISTANCIA
76
77 //ESCALADO ENCODER1
78 "Escalado_Encoders_DB".Encoder1.Entrada := "HSC1";
79 "Escalado_Encoders_DB".Encoder1.Min := INT_TO_DINT("APP".Estacion1.Esc_Encoder.in.Min);
80 "Escalado_Encoders_DB".Encoder1.Max := INT_TO_DINT("APP".Estacion1.Esc_Encoder.in.Max);
81 "Escalado_Encoders_DB".Encoder1.Min_Escalado := INT_TO_REAL("APP".Estacion1.Esc_Encoder.in.Min_Escalado)/10.00;
82 "Escalado_Encoders_DB".Encoder1.Max_Escalado := INT_TO_REAL("APP".Estacion1.Esc_Encoder.in.Max_Escalado)/10.00;
83
84 //ESCALADO ENCODER2
85 "Escalado_Encoders_DB".Encoder2.Entrada := "HSC2";
86 "Escalado_Encoders_DB".Encoder2.Min := INT_TO_DINT("APP".Estacion2.Esc_Encoder.in.Min);
87 "Escalado_Encoders_DB".Encoder2.Max := INT_TO_DINT("APP".Estacion2.Esc_Encoder.in.Max);
88 "Escalado_Encoders_DB".Encoder2.Min_Escalado := INT_TO_REAL("APP".Estacion2.Esc_Encoder.in.Min_Escalado)/10.00;
89 "Escalado_Encoders_DB".Encoder2.Max_Escalado := INT_TO_REAL("APP".Estacion2.Esc_Encoder.in.Max_Escalado)/10.00;
90
91 //ESCALADO ENCODER3
92 "Escalado_Encoders_DB".Encoder3.Entrada := "HSC3";
93 "Escalado_Encoders_DB".Encoder3.Min := INT_TO_DINT("APP".Estacion3.Esc_Encoder.in.Min);
94 "Escalado_Encoders_DB".Encoder3.Max := INT_TO_DINT("APP".Estacion3.Esc_Encoder.in.Max);
95 "Escalado_Encoders_DB".Encoder3.Min_Escalado := INT_TO_REAL("APP".Estacion3.Esc_Encoder.in.Min_Escalado)/10.00;
96 "Escalado_Encoders_DB".Encoder3.Max_Escalado := INT_TO_REAL("APP".Estacion3.Esc_Encoder.in.Max_Escalado)/10.00;
97
98 //ESCALADO ENCODER4
99 "Escalado_Encoders_DB".Encoder4.Entrada := "HSC4";
100 "Escalado_Encoders_DB".Encoder4.Min := INT_TO_DINT("APP".Estacion4.Esc_Encoder.in.Min);
101 "Escalado_Encoders_DB".Encoder4.Max := INT_TO_DINT("APP".Estacion4.Esc_Encoder.in.Max);
102 "Escalado_Encoders_DB".Encoder4.Min_Escalado := INT_TO_REAL("APP".Estacion4.Esc_Encoder.in.Min_Escalado)/10.00;
103 "Escalado_Encoders_DB".Encoder4.Max_Escalado := INT_TO_REAL("APP".Estacion4.Esc_Encoder.in.Max_Escalado)/10.00;
104
105 //CONTROL ENCODER1
106 "Control_Encoder_DB".Encoder1.HSC := "Local-HSC_1";
107 "Control_Encoder_DB".Encoder1.DIR := INT_TO_BOOL("APP".Estacion1.Encoder.in.DIR);
108 "Control_Encoder_DB".Encoder1.CV := INT_TO_BOOL("APP".Estacion1.Encoder.in.CV);
109 "Control_Encoder_DB".Encoder1.RV := INT_TO_BOOL("APP".Estacion1.Encoder.in.RV);
110 "Control_Encoder_DB".Encoder1.PERIOD := INT_TO_BOOL("APP".Estacion1.Encoder.in.PERIOD);
111 "Control_Encoder_DB".Encoder1.NEW_DIR := "APP".Estacion1.Encoder.in.NEW_DIR;
112 "Control_Encoder_DB".Encoder1.NEW_CV := INT_TO_DINT("APP".Estacion1.Encoder.in.NEW_CV);
113 "Control_Encoder_DB".Encoder1.NEW_RV := INT_TO_DINT("APP".Estacion1.Encoder.in.NEW_RV);
114 "Control_Encoder_DB".Encoder1.NEW_PERIOD := "APP".Estacion1.Encoder.in.NEW_PERIOD;
115
```



```

116 //CONTROL ENCODER2
117 "Control_Encoder_DB".Encoder2.HSC := "Local~HSC_2";
118 "Control_Encoder_DB".Encoder2.DIR := INT_TO_BOOL("APP".Estacion2.Encoder.in.DIR);
119 "Control_Encoder_DB".Encoder2.CV := INT_TO_BOOL("APP".Estacion2.Encoder.in.CV);
120 "Control_Encoder_DB".Encoder2.RV := INT_TO_BOOL("APP".Estacion2.Encoder.in.RV);
121 "Control_Encoder_DB".Encoder2.PERIOD := INT_TO_BOOL("APP".Estacion2.Encoder.in.PERIOD);
122 "Control_Encoder_DB".Encoder2.NEW_DIR := "APP".Estacion2.Encoder.in.NEW_DIR;
123 "Control_Encoder_DB".Encoder2.NEW_CV := INT_TO_DINT("APP".Estacion2.Encoder.in.NEW_CV);
124 "Control_Encoder_DB".Encoder2.NEW_RV := INT_TO_DINT("APP".Estacion2.Encoder.in.NEW_RV);
125 "Control_Encoder_DB".Encoder2.NEW_PERIOD := "APP".Estacion2.Encoder.in.NEW_PERIOD;
126
127 //CONTROL ENCODER3
128 "Control_Encoder_DB".Encoder3.HSC := "Local~HSC_3";
129 "Control_Encoder_DB".Encoder3.DIR := INT_TO_BOOL("APP".Estacion3.Encoder.in.DIR);
130 "Control_Encoder_DB".Encoder3.CV := INT_TO_BOOL("APP".Estacion3.Encoder.in.CV);
131 "Control_Encoder_DB".Encoder3.RV := INT_TO_BOOL("APP".Estacion3.Encoder.in.RV);
132 "Control_Encoder_DB".Encoder3.PERIOD := INT_TO_BOOL("APP".Estacion3.Encoder.in.PERIOD);
133 "Control_Encoder_DB".Encoder3.NEW_DIR := "APP".Estacion3.Encoder.in.NEW_DIR;
134 "Control_Encoder_DB".Encoder3.NEW_CV := INT_TO_DINT("APP".Estacion3.Encoder.in.NEW_CV);
135 "Control_Encoder_DB".Encoder3.NEW_RV := INT_TO_DINT("APP".Estacion3.Encoder.in.NEW_RV);
136 "Control_Encoder_DB".Encoder3.NEW_PERIOD := "APP".Estacion3.Encoder.in.NEW_PERIOD;
137
138 //CONTROL ENCODER4
139 "Control_Encoder_DB".Encoder4.HSC := "Local~HSC_4";
140 "Control_Encoder_DB".Encoder4.DIR := INT_TO_BOOL("APP".Estacion4.Encoder.in.DIR);
141 "Control_Encoder_DB".Encoder4.CV := INT_TO_BOOL("APP".Estacion4.Encoder.in.CV);
142 "Control_Encoder_DB".Encoder4.RV := INT_TO_BOOL("APP".Estacion4.Encoder.in.RV);
143 "Control_Encoder_DB".Encoder4.PERIOD := INT_TO_BOOL("APP".Estacion4.Encoder.in.PERIOD);
144 "Control_Encoder_DB".Encoder4.NEW_DIR := "APP".Estacion4.Encoder.in.NEW_DIR;
145 "Control_Encoder_DB".Encoder4.NEW_CV := INT_TO_DINT("APP".Estacion4.Encoder.in.NEW_CV);
146 "Control_Encoder_DB".Encoder4.NEW_RV := INT_TO_DINT("APP".Estacion4.Encoder.in.NEW_RV);
147 "Control_Encoder_DB".Encoder4.NEW_PERIOD := "APP".Estacion4.Encoder.in.NEW_PERIOD;

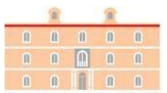
```



```

1 //SALIDAS
2 "Out_Meter_Eje" := "Control_Ejes_DB".Eje1.Metiendo_Eje;
3 "Out_Sacar_Eje" := "Control_Ejes_DB".Eje1.Sacando_Eje;
4 "Out_Meter_Eje_2" := "Control_Ejes_DB".Eje2.Metiendo_Eje;
5 "Out_Sacar_Eje_2" := "Control_Ejes_DB".Eje2.Sacando_Eje;
6 "Out_Meter_Eje_3" := "Control_Ejes_DB".Eje3.Metiendo_Eje;
7 "Out_Sacar_Eje_3" := "Control_Ejes_DB".Eje3.Sacando_Eje;
8 "Out_Meter_Eje_4" := "Control_Ejes_DB".Eje4.Metiendo_Eje;
9 "Out_Sacar_Eje_4" := "Control_Ejes_DB".Eje4.Sacando_Eje;
10
11 //EJES ELECTRICOS
12
13 //ESCALADO EJE1
14 "APP".Estacion1.Esc_Eje.out.Analogica := "Escalado_Ejes_DB".Eje1.Analogica;
15 "APP".Estacion1.Esc_Eje.out.Analogica_Fisica := REAL_TO_INT("Escalado_Ejes_DB".Eje1.Analogica_Fisica);
16 "APP".Estacion1.Esc_Eje.out.Analogica_Escalada := REAL_TO_INT("Escalado_Ejes_DB".Eje1.Analogica_Escalada);
17
18 //ESCALADO EJE2
19 "APP".Estacion2.Esc_Eje.out.Analogica := "Escalado_Ejes_DB".Eje2.Analogica;
20 "APP".Estacion2.Esc_Eje.out.Analogica_Fisica := REAL_TO_INT("Escalado_Ejes_DB".Eje2.Analogica_Fisica);
21 "APP".Estacion2.Esc_Eje.out.Analogica_Escalada := REAL_TO_INT("Escalado_Ejes_DB".Eje2.Analogica_Escalada);
22
23 //ESCALADO EJE3
24 "APP".Estacion3.Esc_Eje.out.Analogica := "Escalado_Ejes_DB".Eje3.Analogica;
25 "APP".Estacion3.Esc_Eje.out.Analogica_Fisica := REAL_TO_INT("Escalado_Ejes_DB".Eje3.Analogica_Fisica);
26 "APP".Estacion3.Esc_Eje.out.Analogica_Escalada := REAL_TO_INT("Escalado_Ejes_DB".Eje3.Analogica_Escalada);
27
28 //ESCALADO EJE4
29 "APP".Estacion4.Esc_Eje.out.Analogica := "Escalado_Ejes_DB".Eje4.Analogica;
30 "APP".Estacion4.Esc_Eje.out.Analogica_Fisica := REAL_TO_INT("Escalado_Ejes_DB".Eje4.Analogica_Fisica);
31 "APP".Estacion4.Esc_Eje.out.Analogica_Escalada := REAL_TO_INT("Escalado_Ejes_DB".Eje4.Analogica_Escalada);

```

```
32
33 //CONTROL EJE1
34 "APP".Estacion1.Eje.out.PV := REAL_TO_INT("Control_Ejes_DB".Eje1.PV);
35 "APP".Estacion1.Eje.out.Sacar_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje1.Sacar_Eje_Aut);
36 "APP".Estacion1.Eje.out.Meter_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje1.Meter_Eje_Aut);
37 "APP".Estacion1.Eje.out.En_Posicion := BOOL_TO_INT("Control_Ejes_DB".Eje1.En_Posicion);
38 "APP".Estacion1.Eje.out.SP_Normalizado := REAL_TO_INT("Control_Ejes_DB".Eje1.SP_Normalizado);
39 "APP".Estacion1.Eje.out.Sacando_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje1.Sacando_Eje);
40 "APP".Estacion1.Eje.out.Metiendo_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje1.Metiendo_Eje);
41 "APP".Estacion1.Eje.out.Alarma := BOOL_TO_INT("Control_Ejes_DB".Eje1.Alarma);
42
43 //CONTROL EJE2
44 "APP".Estacion2.Eje.out.PV := REAL_TO_INT("Control_Ejes_DB".Eje2.PV);
45 "APP".Estacion2.Eje.out.Sacar_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje2.Sacar_Eje_Aut);
46 "APP".Estacion2.Eje.out.Meter_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje2.Meter_Eje_Aut);
47 "APP".Estacion2.Eje.out.En_Posicion := BOOL_TO_INT("Control_Ejes_DB".Eje2.En_Posicion);
48 "APP".Estacion2.Eje.out.SP_Normalizado := REAL_TO_INT("Control_Ejes_DB".Eje2.SP_Normalizado);
49 "APP".Estacion2.Eje.out.Sacando_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje2.Sacando_Eje);
50 "APP".Estacion2.Eje.out.Metiendo_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje2.Metiendo_Eje);
51 "APP".Estacion2.Eje.out.Alarma := BOOL_TO_INT("Control_Ejes_DB".Eje1.Alarma);
52
53 //CONTROL EJE3
54 "APP".Estacion3.Eje.out.PV := REAL_TO_INT("Control_Ejes_DB".Eje3.PV);
55 "APP".Estacion3.Eje.out.Sacar_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje3.Sacar_Eje_Aut);
56 "APP".Estacion3.Eje.out.Meter_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje3.Meter_Eje_Aut);
57 "APP".Estacion3.Eje.out.En_Posicion := BOOL_TO_INT("Control_Ejes_DB".Eje3.En_Posicion);
58 "APP".Estacion3.Eje.out.SP_Normalizado := REAL_TO_INT("Control_Ejes_DB".Eje3.SP_Normalizado);
59 "APP".Estacion3.Eje.out.Sacando_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje3.Sacando_Eje);
60 "APP".Estacion3.Eje.out.Metiendo_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje3.Metiendo_Eje);
61 "APP".Estacion3.Eje.out.Alarma := BOOL_TO_INT("Control_Ejes_DB".Eje1.Alarma);
62
63 //CONTROL EJE4
64 "APP".Estacion4.Eje.out.PV := REAL_TO_INT("Control_Ejes_DB".Eje4.PV);
65 "APP".Estacion4.Eje.out.Sacar_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje4.Sacar_Eje_Aut);
66 "APP".Estacion4.Eje.out.Meter_Eje_Aut := BOOL_TO_INT("Control_Ejes_DB".Eje4.Meter_Eje_Aut);
67 "APP".Estacion4.Eje.out.En_Posicion := BOOL_TO_INT("Control_Ejes_DB".Eje4.En_Posicion);
68 "APP".Estacion4.Eje.out.SP_Normalizado := REAL_TO_INT("Control_Ejes_DB".Eje4.SP_Normalizado);
69 "APP".Estacion4.Eje.out.Sacando_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje4.Sacando_Eje);
70 "APP".Estacion4.Eje.out.Metiendo_Eje := BOOL_TO_INT("Control_Ejes_DB".Eje4.Metiendo_Eje);
71 "APP".Estacion4.Eje.out.Alarma := BOOL_TO_INT("Control_Ejes_DB".Eje1.Alarma);
72
73 //ENCODERS DISTANCIA
74
75 //ESCALADO ENCODER1
76 "APP".Estacion1.Esc_Encoder.out.Entrada := DINT_TO_INT("Escalado_Encoders_DB".Encoder1.Entrada);
77 "APP".Estacion1.Esc_Encoder.out.Salida_Escalada := REAL_TO_INT("Escalado_Encoders_DB".Encoder1.Salida_Escalada);
78
79 //ESCALADO ENCODER2
80 "APP".Estacion2.Esc_Encoder.out.Entrada := DINT_TO_INT("Escalado_Encoders_DB".Encoder2.Entrada);
81 "APP".Estacion2.Esc_Encoder.out.Salida_Escalada := REAL_TO_INT("Escalado_Encoders_DB".Encoder2.Salida_Escalada);
82
83 //ESCALADO ENCODER3
84 "APP".Estacion3.Esc_Encoder.out.Entrada := DINT_TO_INT("Escalado_Encoders_DB".Encoder3.Entrada);
85 "APP".Estacion3.Esc_Encoder.out.Salida_Escalada := REAL_TO_INT("Escalado_Encoders_DB".Encoder3.Salida_Escalada);
86
87 //ESCALADO ENCODER4
88 "APP".Estacion4.Esc_Encoder.out.Entrada := DINT_TO_INT("Escalado_Encoders_DB".Encoder4.Entrada);
89 "APP".Estacion4.Esc_Encoder.out.Salida_Escalada := REAL_TO_INT("Escalado_Encoders_DB".Encoder4.Salida_Escalada);
90
91 //CONTROL ENCODER1
92 "APP".Estacion1.Encoder.out.BUSY := BOOL_TO_INT("Control_Encoder_DB".Encoder1.BUSY);
93 "APP".Estacion1.Encoder.out.STATUS := "Control_Encoder_DB".Encoder1.STATUS;
94
95 //CONTROL ENCODER2
96 "APP".Estacion2.Encoder.out.BUSY := BOOL_TO_INT("Control_Encoder_DB".Encoder2.BUSY);
97 "APP".Estacion2.Encoder.out.STATUS := "Control_Encoder_DB".Encoder2.STATUS;
98
99 //CONTROL ENCODER3
100 "APP".Estacion3.Encoder.out.BUSY := BOOL_TO_INT("Control_Encoder_DB".Encoder3.BUSY);
101 "APP".Estacion3.Encoder.out.STATUS := "Control_Encoder_DB".Encoder3.STATUS;
102
103 //CONTROL ENCODER4
104 "APP".Estacion4.Encoder.out.BUSY := BOOL_TO_INT("Control_Encoder_DB".Encoder4.BUSY);
105 "APP".Estacion4.Encoder.out.STATUS := "Control_Encoder_DB".Encoder4.STATUS;
```

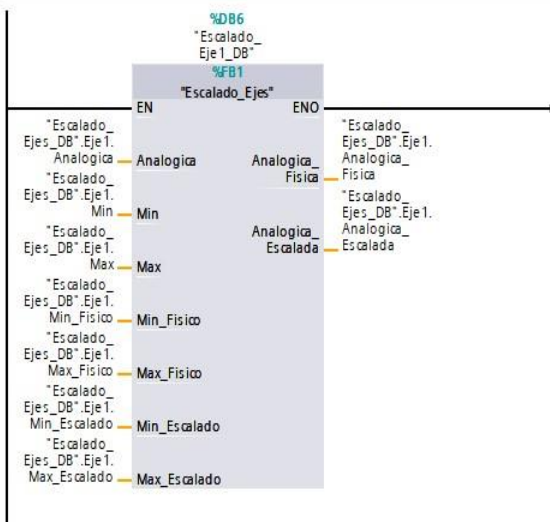

- ▼ Bloques de programa
 - ✦ Agregar nuevo bloque
 - ✦ Main [OB1]
 - ▶ 00_OBs
 - ▶ 01_Interface
 - ▼ 02_Ejes
 - ✦ Ejes_Resistencia [FC1]
 - ▶ Control
 - ▶ Escalado
 - ▶ Regulacion_PID
 - ▶ 03_Encoder
 - ▶ 04_APP

▼ Título del bloque:

EJES_RESISTENCIA

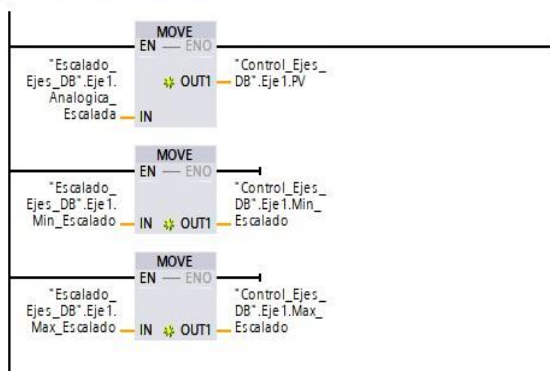
▼ Segmento 1:

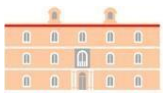
ESCALADA EJE 1



▼ Segmento 2:

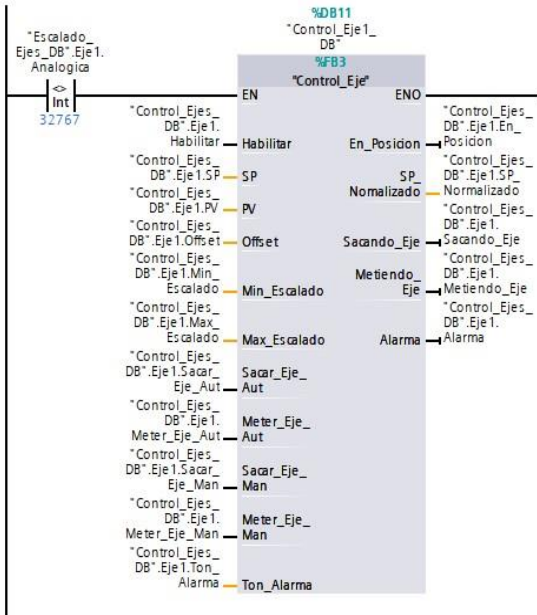
INTERFAZ VARIABLES ESCALADO Y CONTROL EJE 1





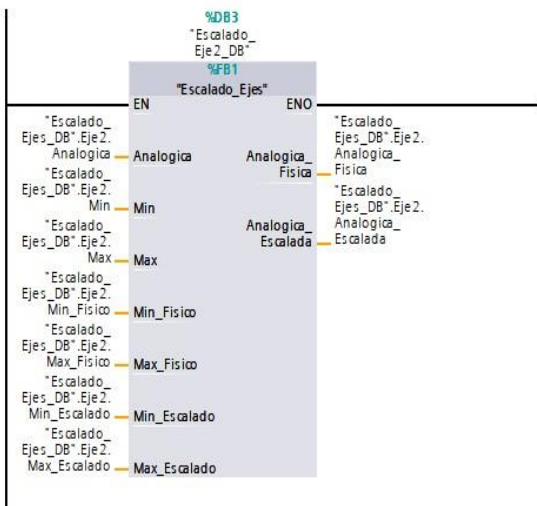
Segmento 3:

CONTROL EJE 1



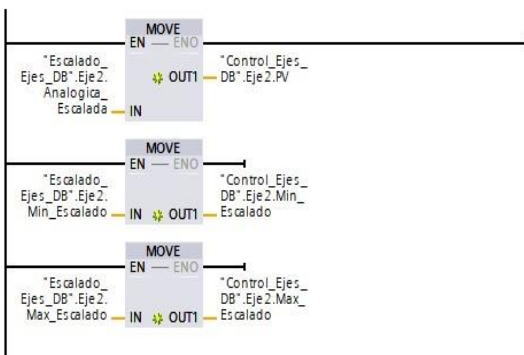
Segmento 4:

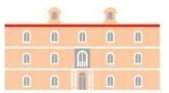
ESCALADO EJE 2



Segmento 5:

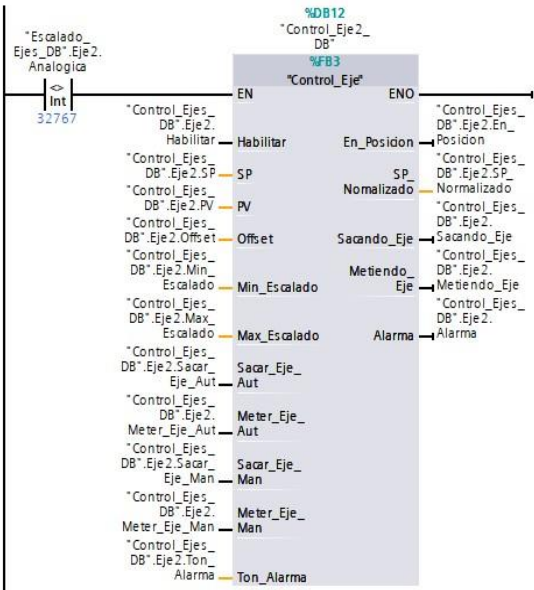
INTERFAZ VARIABLES ESCALADO Y CONTROL EJE 2





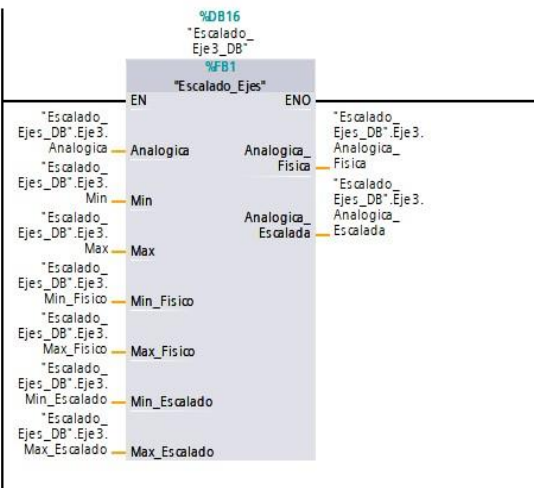
Segmento 6:

CONTROL EJE 2



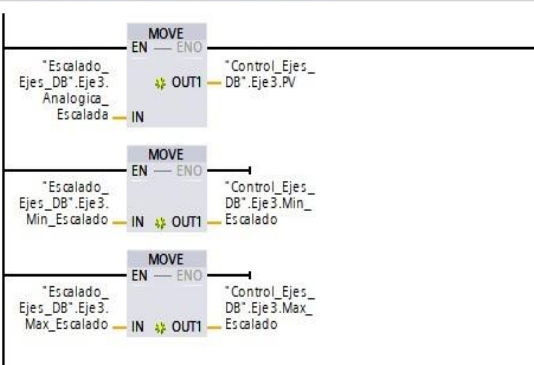
Segmento 7:

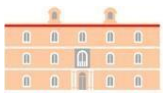
ESCALADO EJE 3



Segmento 8:

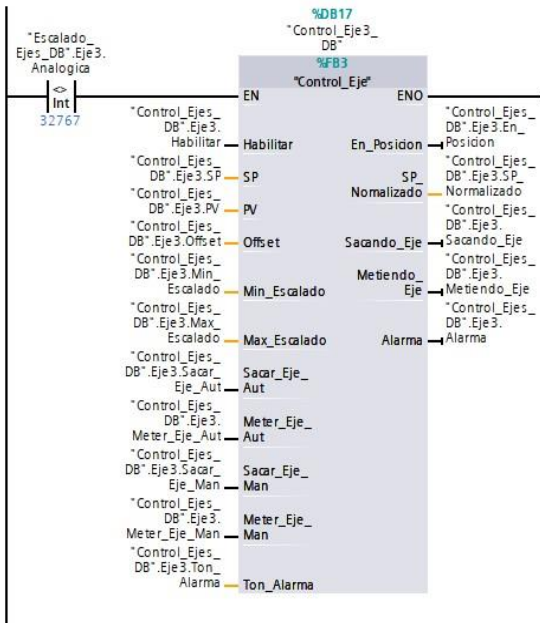
INTERFAZ VARIABLES ESCALADO Y CONTROL EJE 3





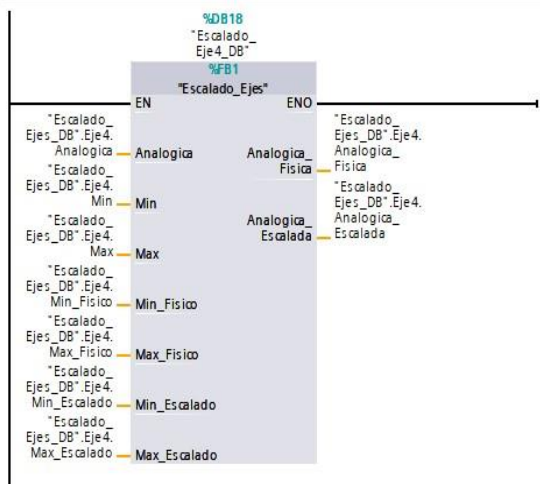
Segmento 9:

CONTROL EJE 3



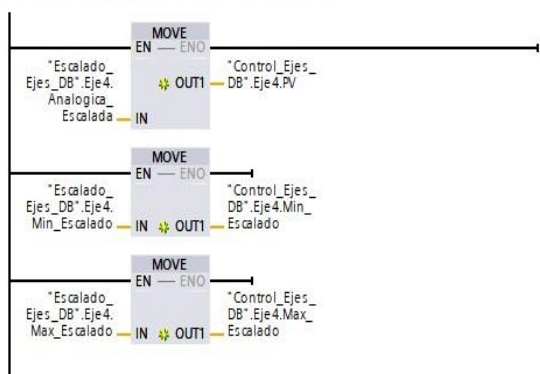
Segmento 10:

ESCALADO EJE 4



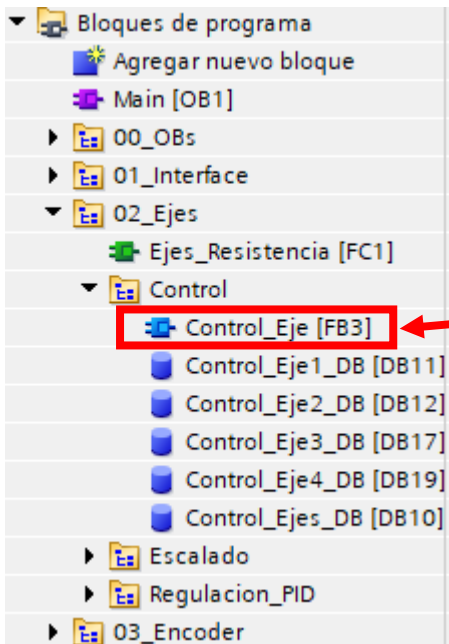
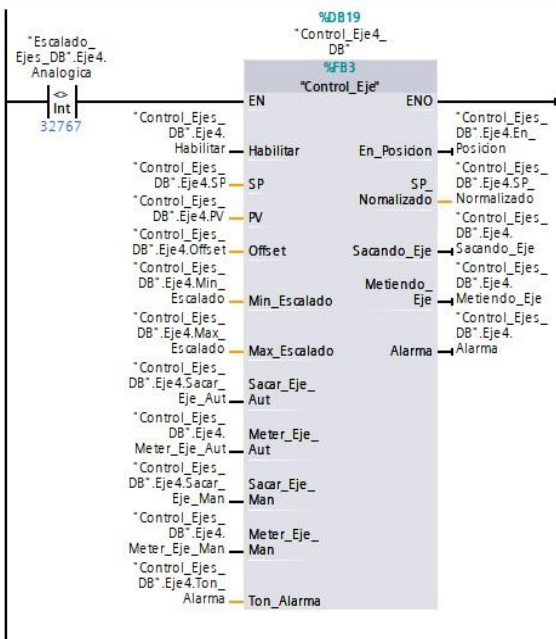
Segmento 11:

INTERFAZ VARIABLES ESCALADO Y CONTROL EJE 4



▼ Segmento 12:

CONTROL EJE 4

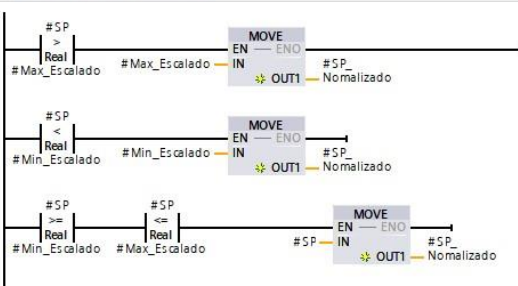


▼ Título del bloque:

CONTROL EJES

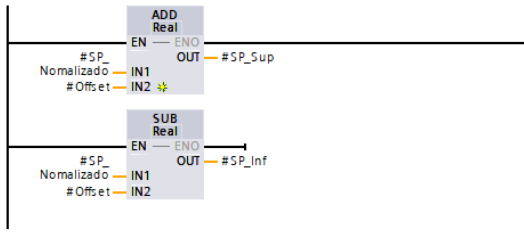
▼ Segmento 1:

▼ SEGMENTO DE SEGURIDAD PARA QUE EL SET-POINT ESTE DENTRO DEL RANGO DEL MÍNIMO Y EL MÁXIMO ESCALADO



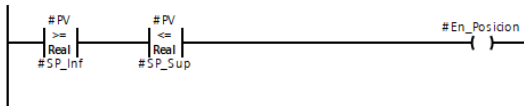
Segmento 2:

SUMAMOS Y RESTAMOS EL OFFSET AL SET-POINT NORMALIZADO PARA CREAR UN RANGO DE VALORES DONDE EL EJE DEJARÁ DE REGULAR (ESTARÁ EN POSICIÓN)



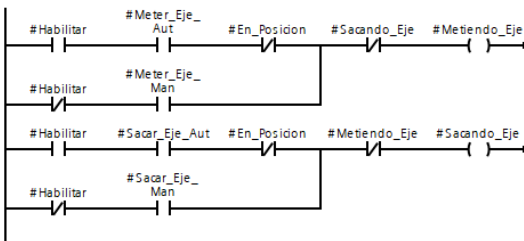
Segmento 3:

SI EL VALOR ACTUAL DE LA POSICIÓN DEL EJE ESTA DENTRO DEL RANGO QUE HEAMOS CREADO EN EL SEGMENTO 2, EL EJE ESTARÁ EN LA POSICIÓN INDICADA



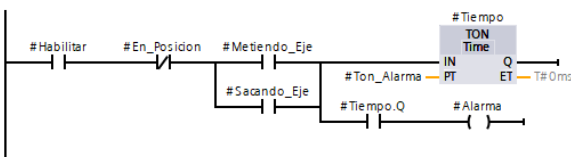
Segmento 4:

MODOS DE FUNCIONAMIENTO: MANUAL Y AUTOMÁTICO



Segmento 5:

ALARMA



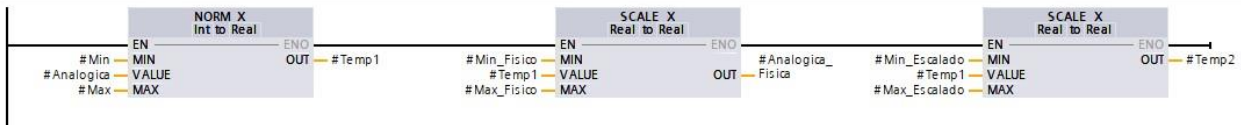
- Variables auxiliares utilizadas en el proceso de control de los ejes:

Control_Ejes_DB		
	Nombre	Tipo de datos
1	Static	
2	Eje1	*Eje_UDT
3	Habilitar	Bool
4	SP	Real
5	PV	Real
6	Offset	Real
7	Min_Escalado	Real
8	Max_Escalado	Real
9	Sacar_Eje_Aut	Bool
10	Meter_Eje_Aut	Bool
11	Sacar_Eje_Man	Bool
12	Meter_Eje_Man	Bool
13	Ton_Alarma	Time
14	En_Posicion	Bool
15	SP_Normalizado	Real
16	Sacando_Eje	Bool
17	Metiendo_Eje	Bool
18	Alarma	Bool
19	Eje2	*Eje_UDT
20	Eje3	*Eje_UDT
21	Eje4	*Eje_UDT

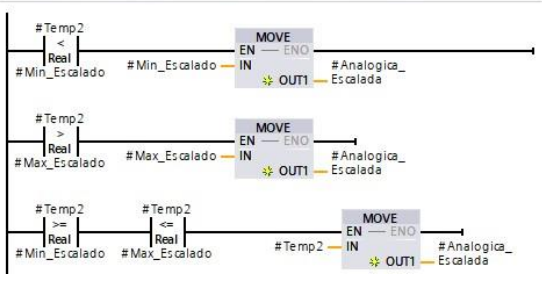
Bloques de programa
 Agregar nuevo bloque
 Main [OB1]
 00_OBs
 01_Interface
 02_Ejes
 Ejes_Resistencia [FC1]
 Control
 Escalado
Escalado_Ejes [FB1] ←
 Escalado_Eje1_DB..
 Escalado_Eje2_DB..
 Escalado_Eje3_DB..
 Escalado_Eje4_DB..
 Escalado_Ejes_DB..
 Regulacion_PID
 03_Encoder

Título del bloque: ESCALADO EJES

Segmento 1: FUNCIONES QUE NOS FACILITAN EL ESCALADO EN S71200



Segmento 2: SEGMENTO DE SEGURIDAD PARA QUE EL VALOR OBTENIDO DE ANALÓGICA ESCALADA EN mm ESTÉ DENTRO DEL MÍNIMO Y EL MÁXIMO ESCALADO



- Variables auxiliares utilizadas en el proceso de escalado de los ejes:

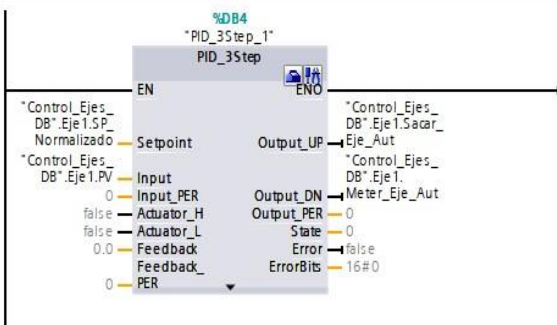
Escalado_Ejes_DB		Tipo de datos
	Nombre	
1	Static	
2	Eje 1	*Escalado_Eje_U..
3	Analogica	Int
4	Min	Int
5	Max	Int
6	Min_Fisico	Real
7	Max_Fisico	Real
8	Min_Escalado	Real
9	Max_Escalado	Real
10	Analogica_Fisica	Real
11	Analogica_Escalada	Real
12	Eje2	*Escalado_Eje_UDT
13	Eje3	*Escalado_Eje_UDT
14	Eje4	*Escalado_Eje_UDT

▼ Bloques de programa

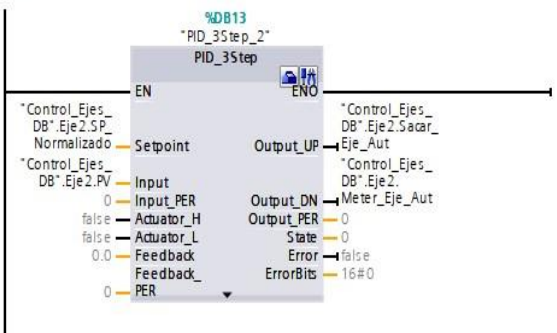
- Agregar nuevo bloque
- Main [OB1]
- ▶ 00_OBs
- ▶ 01_Interface
- ▼ 02_Ejes
 - Ejes_Resistencia [FC1]
 - ▶ Control
 - ▶ Escalado
 - ▼ Regulacion_PID
 - Regulación PID Ej...
- ▶ 03_Encoder
- ▶ 04_APP

▼ Título del bloque:
REGULACIÓN PID DE LOS EJES

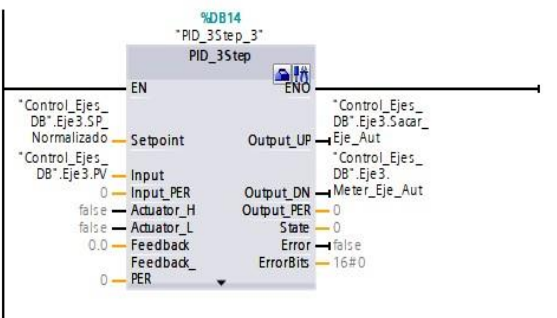
▼ Segmento 1:
PID_3STEP_1

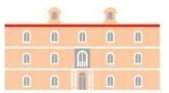


▼ Segmento 2:
PID_3STEP_2



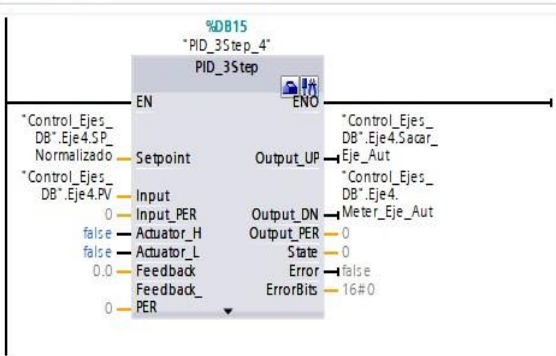
▼ Segmento 3:
PID_3STEP_3





Segmento 4:

PID_3STEP_4



Bloques de programa

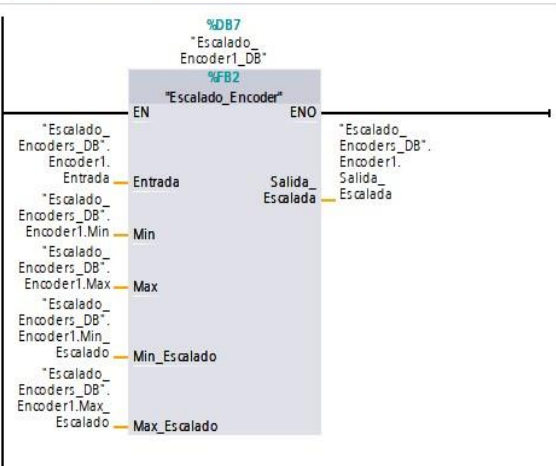
- Agregar nuevo bloque
- Main [OB1]
- 00_OBs
- 01_Interface
- 02_Ejes
- 03_Encoder
 - Encoders_Distancia [...]**
 - Control
 - Escalado
- 04_APP

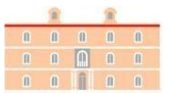
Título del bloque:

ENCODERS_DISTANCIA

Segmento 1:

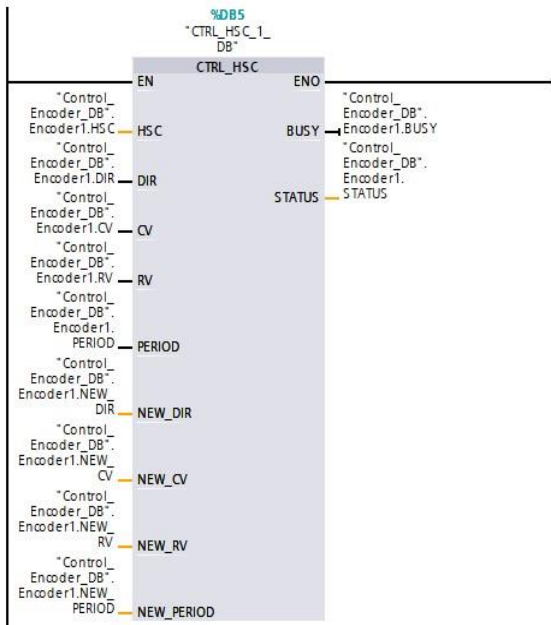
ESCALADO ENCODER 1





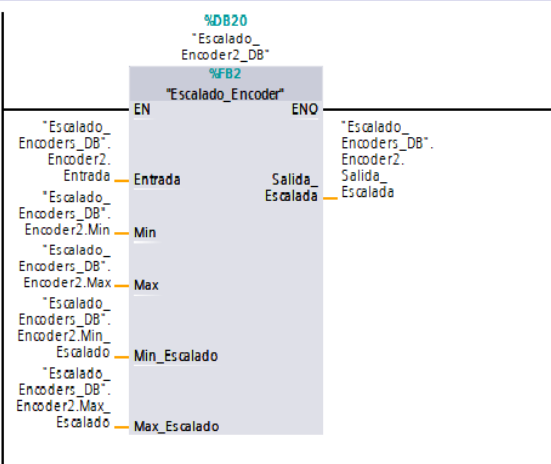
Segmento 2:

CONTROL ENCODER 1



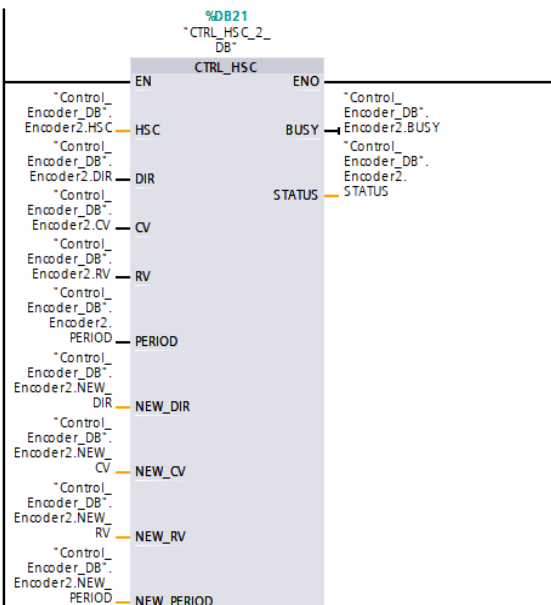
Segmento 3:

ESCALADO ENCODER 2



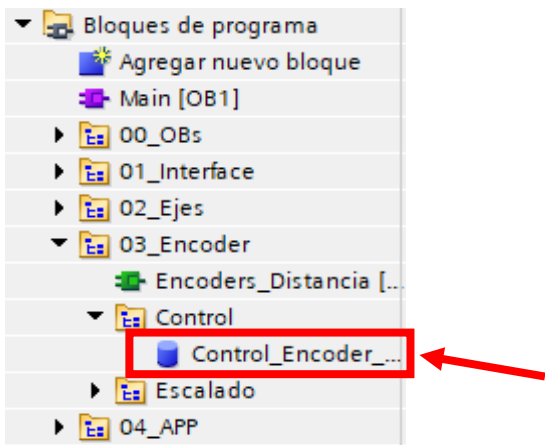
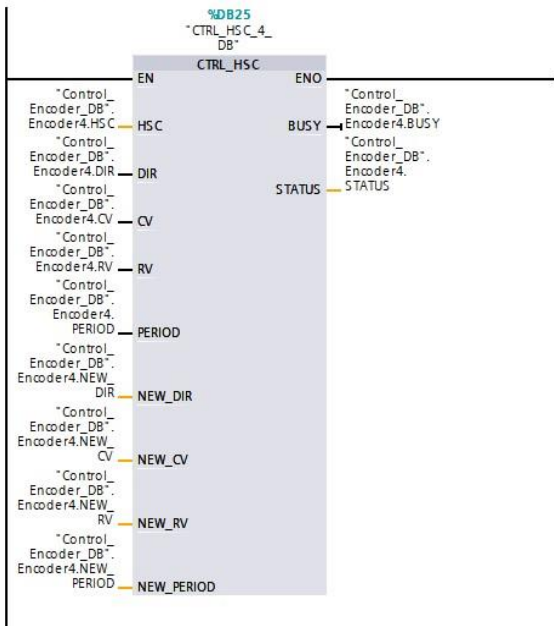
Segmento 4:

CONTROL ENCODER 2



Segmento 8:

CONTROL ENCODER 4



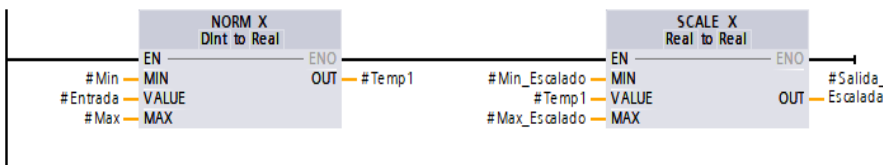
- Variables auxiliares del proceso de control de los encoders (función proporcionada por SIEMENS):

Control_Encoder_DB		
	Nombre	Tipo de datos
1	Static	
2	Encoder1	*Encoder_UDT
3	HSC	HW_HSC
4	DIR	Bool
5	CV	Bool
6	RV	Bool
7	PERIOD	Bool
8	NEW_DIR	Int
9	NEW_CV	DInt
10	NEW_RV	DInt
11	NEW_PERIOD	Int
12	BUSY	Bool
13	STATUS	Int
14	Encoder2	*Encoder_UDT
15	Encoder3	*Encoder_UDT
16	Encoder4	*Encoder_UDT

Bloques de programa
 Agregar nuevo bloque
 Main [OB1]
 00_OBs
 01_Interface
 02_Ejes
 03_Encoder
 Encoders_Distancia [FC4]
 Control
 Escalado
Escalado_Encoder [FB2]
 Escalado_Encoder1_DB [DB7]
 Escalado_Encoder2_DB [DB20]
 Escalado_Encoder3_DB [DB22]
 Escalado_Encoder4_DB [DB24]
 Escalado_Encoders_DB [DB9]
 04_APP

Título del bloque: ESCALADO ENCODERS

Segmento 1: FUNCIONES QUE NOS FACILITAN EL ESCALADO EN S71200



- Variables auxiliares del proceso de escalado de los encoders:

Escalado_Encoders_DB		
	Nombre	Tipo de datos
1	Static	
2	Encoder1	*Escalado_Enco...
3	Entrada	DInt
4	Min	DInt
5	Max	DInt
6	Min_Escalado	Real
7	Max_Escalado	Real
8	Salida_Escalada	Real
9	Encoder2	*Escalado_Encoder...
10	Encoder3	*Escalado_Encoder...
11	Encoder4	*Escalado_Encoder...

8.2 PRESUPUESTO

En este apartado se van a mostrar los costes del proyecto:

- Costes del hardware
- Costes de mano de obra: instalación de los nuevos equipos de control y programación de estos.

HARDWARE	Uds	€/Ud	Precio total
Maqueta suministrada por el cliente	0	0	0
PLC S7-1200 de Siemens	1	432,12	432,12
SM 1231 AI4 x 13 bits	1	251,60	251,60
Fuente S8PS-05024CD de OMRON	1	53,24	53,24
Eje eléctrico Linak LA14	4	370,80	1.483,20
Relé RSL1AB4BD	16	23,40	374,40
Zócalo RSLZVA1	16	8,93	142,88
Encoder Industrial de Industrial Shields	4	33,55	134,20
Bornas con resistencia Pull-Up de Shields	8	10,26	82,08
Bornas Phoenix Contact	20	8,23	164,60
Wifi Tp-Link CPE510	1	63,63	63,63
Conector POE	1	9,86	9,86
Conector 4 hilos para encoder	4	5,65	22,60
Conector 5 hilos para eje eléctrico	4	5,65	22,60
Material pequeño	1	80	80
TOTAL			3.317,01 €

Tabla - Costes del hardware

Mano de obra	Horas	Costes mano de obra
Programación TIA PORTAL	80	1.600
Realización plano eléctrico	20	400
Conexionado del hardware	10	200
Desarrollo aplicación móvil	40	800
Configuración punto de acceso	15	300
Pruebas de funcionamiento	20	400
Instalación de la maqueta en el campo de entrenamiento	10	200
TOTAL		3.900 €

Tabla - Costes de la mano de obra

Coste del hardware	3.317,01 €
Costes de la mano de obra	3.900 €
COSTE FINAL	7.217,01 €

Tabla – Coste final

8.3 CONFIGURACIÓN COMPLETA WIFI TP-LINK

Una vez seleccionado el modo de operación del dispositivo *router*, falta por configurar otros parámetros importantes, como el tipo de conexión WAN, atendiendo a su funcionamiento específico. Se utiliza un IP Dinámico.

Se desea que este ISP utilice un servicio DHCP para asignar a su *router* una dirección IP cuando se conecte a internet.

CONFIGURACIÓN RÁPIDA ESTADO RED INALÁMBRICO ADMINISTRACIÓN SISTEMA

Tipo de Conexión de WAN

Por favor seleccione el Tipo de Conexión del puerto WAN de acuerdo a sus necesidades:

- PPPoE - Para esta conexión, necesitará su nombre de cuenta y contraseña de su ISP.
- IP Dinámico - Su ISP usa un servicio DHCP para asignar a su Router una dirección IP cuando se conecte a Internet.
- IP Estático – Este tipo de conexión usa una dirección IP fija (estática que su ISP asignó).

Volver Siguiente

El protocolo de red DHCP se encarga de asignar de manera dinámica y automática una dirección IP, ya sea una dirección IP privada desde el router hacia los equipos de la red local, o también una IP pública por parte de un operador que utilice este tipo de protocolo para el establecimiento de la conexión.

El protocolo DHCP incluye varias formas de asignación de direcciones IP, dependiendo de la configuración que se realice y el escenario. En este caso se va a utilizar una asignación automática de direcciones IP.

Para la asignación de direcciones IP de forma automática, el DHCP se encarga de proporcionar una dirección IP al cliente que realiza la solicitud y está disponible para este cliente hasta que la libere. Existen *routers* que internamente están configurados para proporcionar direcciones IP privadas de forma secuencial, sin embargo, hay *firmwares* que están diseñados para proporcionar una dirección IP específica dentro del rango y que no es secuencial, en base a un algoritmo interno y la dirección MAC que se haya conectado.

Se ha fijado la dirección IP del dispositivo Wifi para que no varíe su dirección IP cada vez que se le dé corriente. De esta manera, en el proyecto se tendrán dos direcciones IP fijas: la del PLC y la del dispositivo Wifi.

En la siguiente imagen se muestra la configuración de Cliente Inalámbrico. El cliente, dentro del rango de acción de este dispositivo, cuando busque redes wifi encontrará una red llamada "TP-Link_CFCA" gracias al SSID que se le ha puesto. Los parámetros Modo, WDS, Ancho de canal y Seguridad, se dejan de fábrica. Sin embargo, con respecto al parámetro de la Distancia, se ha reducido un poco el alcance porque se quiere que la intensidad de WIFI siempre sea la adecuada y no haya problemas de conexión. Se podría reducir mucho más, ya que un campo de fútbol apenas tiene 1 km, pero al tener el máximo en 27,9 km, en el intervalo [0,20] km no se debería tener ningún problema para conectarnos.

Configuraciones de Cliente Inalámbrico

SSID de AP Remoto:

MAC de AP Remoto: Fijar al Punto de Acceso

Modo:

WDS:

Ancho del canal:

Seguridad:

Contraseña de PSK: Mostrar

No recomendamos usar la encriptación WEP, puede ir a la página de INALÁMBRICO para configurarlo.

Configuración de Distancia: (0-27.9)km

Volver

Siguiente

En el final de la configuración se puede observar los parámetros más importantes que se acaban de configurar. Para terminar la configuración, únicamente, se tiene que pulsar en Finalizar.

Finalizar

Modo de Operación: AP Router Cliente (Cliente WISP)

Tipo de Conexión de WAN: IP Dinámica

SSID de AP Remoto: TP-Link_CFCA

Modo: 802.11a/n

WDS: Auto

Ancho del canal: 20/40MHz

Seguridad: Ninguno

Configuración de Distancia: 20 km

Señal inalámbrica: Activar

SSID: TP-Link_CFCA

Seguridad: Ninguno

Volver

Finalizar

El último apartado trata sobre las redes WAN y LAN de este dispositivo.

CONFIGURACIÓN RÁPIDA **ESTADO** **RED** **INALÁMBRICO** **ADMINISTRACIÓN** **SISTEMA**

WAN ⌵

Tipo de Conexión:

IPv6: Activar ?

Configuraciones Avanzadas

Tamaño MTU:

Usar Estos Servidores DNS: Activar

Dirección de WAN MAC:

Su Dirección de MAC de PC:

LAN ⌵

Tipo de Conexión:

Dirección IP:

Máscara de Red:

Tamaño MTU:

Proxy IGMP: Activar

Servidor DHCP: Activar

Servidor DHCP

Iniciar Dirección IP: Dirección de IP Final:

Puerta de enlace predeterminada: Dominio predeterminado:

DNS Primario: DNS Secundario:

Tiempo de Concesión: minutos

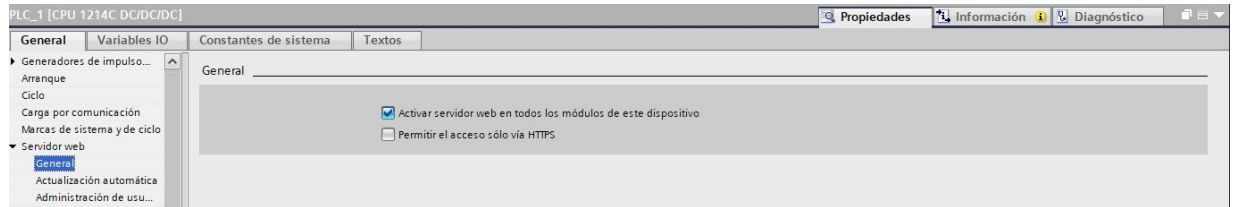
Nos centramos en la red de área local (LAN). Como se ha comentado anteriormente, se fija la dirección IP del wifi TP-Link en la dirección 10.166.122.1, al igual que la del PLC (dirección IP 10.166.122.178).

También se ha comentado el protocolo de red DHCP que se ha utilizado. Se ha establecido un rango de 10 posiciones IP aleatorias [10.166.122.190 – 10.166.122.199]. A cada dispositivo que se conecta a esta red se le asigna una IP aleatoria que está en ese rango. Si hubiera 10 dispositivos conectados, no podría conectarse ningún dispositivo más. Si el cliente lo especificara se podría ampliar este rango hasta en 241 direcciones IP, debido a que esta red permite tener 253 direcciones IP.

Es importante destacar que la dirección IP fija del dispositivo TP-LINK será la puerta de enlace predeterminada del protocolo DHCP

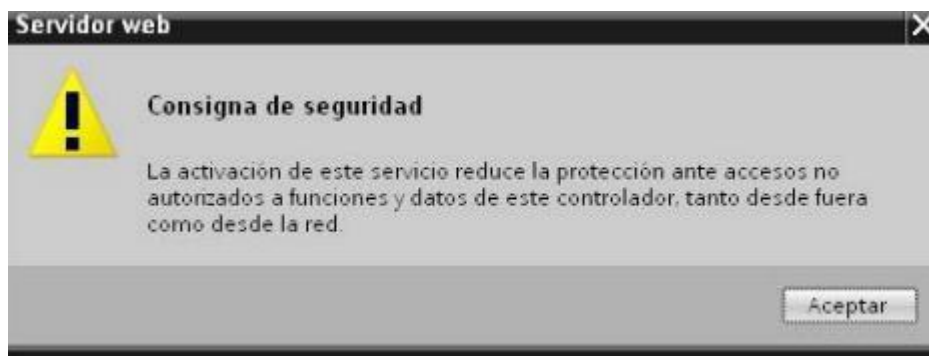
8.4 CONFIGURACIÓN SERVIDOR WEB EN TIA PORTAL

A continuación, se muestra la realización paso a paso de la configuración del servidor web en TIA Portal:



Primero, para disponer de toda la funcionalidad del servidor web, hay que activarlo. Para ello, en la ventana de inspección "Propiedades", después "General", se va a la sección "Servidor web". Se pulsa la casilla "Activar servidor web en todos los módulos de este dispositivo" (ver figura 55).

Al hacerlo, aparece el siguiente aviso:



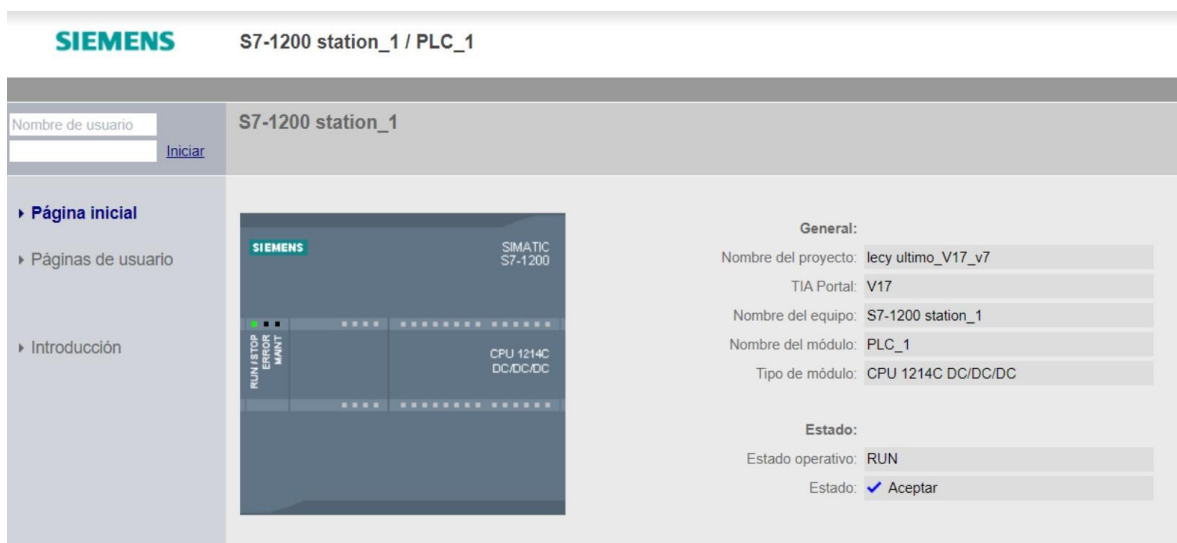
En el área "Vista general de las interfaces" existe la posibilidad de abrir o cerrar el acceso al servidor web:

Vista general de las interfaces		
Dispositivo	Interfaz	Activar acceso al ...
PLC_1	Interfaz PROFINET_1	<input checked="" type="checkbox"/>

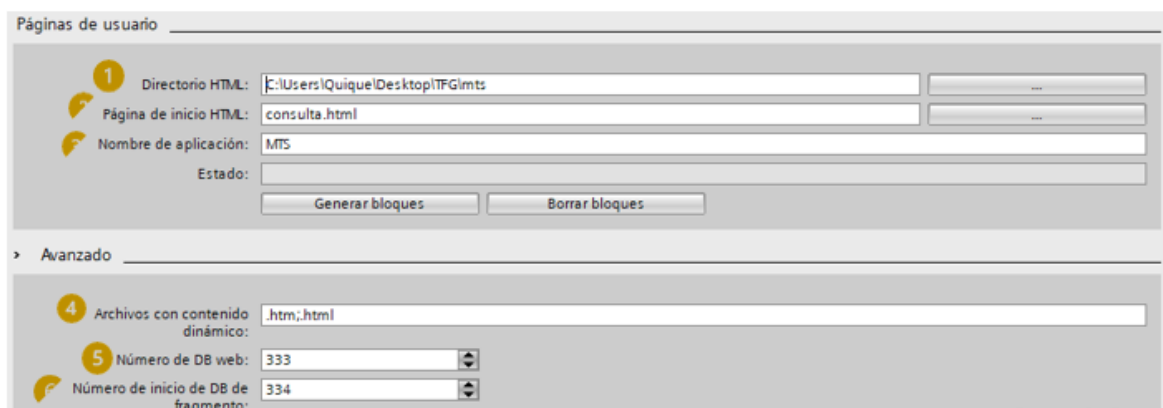
Para establecer una conexión con el servidor web se introduce la dirección IP de la CPU configurada en la barra de dirección del navegador web: <http://10.166.122.178>. Se establece la conexión y se abre la introducción del servidor web SIMATIC S7-1200: donde se muestran la información principal de nuestra CPU.



Al pulsar “INTRO” se abre la página inicial donde se muestra la información principal de nuestra CPU:



A continuación, se va a realizar la configuración de las páginas de usuario. En este apartado se cargan los archivos *HTML* que realizan la comunicación entre la aplicación y el autómata.



1. En el área “Páginas de usuario” se selecciona en ① “Directorio HTML” la carpeta en la que se ha guardado nuestro archivo HTML.
2. En ② “Página de inicio HTML” se introduce el nombre de la página HTML que se debe abrir al arrancar la aplicación.
3. En ③ se ha asignado el nombre de MTS a la aplicación web. Este nombre sirve para la posterior clasificación o agrupación de las páginas web.
4. Una vez introducidos los campos anteriores, el software permite hacer clic en “Generar bloques” para generar bloques de datos a partir de los archivos fuente. Los bloques de datos generados se guardan en el árbol del proyecto de STEP 7 en la carpeta “Bloques de sistema > Servidor web”. Estos bloques de datos se componen de un bloque de datos de control, que regula la visualización de las páginas web, y de uno o varios fragmentos de bloques de datos con las páginas web compiladas
5. En el área “Avanzado” se selecciona ④ “Archivos con contenido dinámico”. En este apartado se introducen las extensiones de los archivos que deben comprobarse en los comandos AWP. En este caso “.htm” y “.HTML”.
6. En el apartado ⑤ y ⑥ se generan automáticamente el número para el DB Web y el número de inicio del DB de fragmento respectivamente. Si se desea, se puede modificar estos números por otros que no estén ocupados.

8.5 PLANOS CON EPLAN 2.7

En este apartado se van a adjuntar los siguientes planos:

1. Esquema de fuerza: Acometida y distribución.
2. Configuración CPU.
3. Entradas digitales PLC.
4. Conexionado encoder.
5. Entradas módulo 1 (Analógicas).
6. Conexionado eje eléctrico.
7. Salidas PLC.
8. Activación de los ejes.
9. Plano de conexiones de bornes XBOT1.



BINARII

Cliente: Lecy Sport
Desig. instalación: Control a distancia de maquina de entrenamiento
N.º proyecto: 1

Destino: Lecy Sport
Lugar de instalación: Campo de entrenamiento

Tensión de alimentación: 230 VAC / 50 Hz
Tensión de mando: 24 VDC
Potencia / Intensidad: X kW / X A
Año de fabricación: 2022

Responsable de la instalación: Enrique Baños Campillo
Realizado: 17/02/2022 por GCC2
Comprobado: 08/04/2022 por GCC3

Cantidad de páginas: 13

2

Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



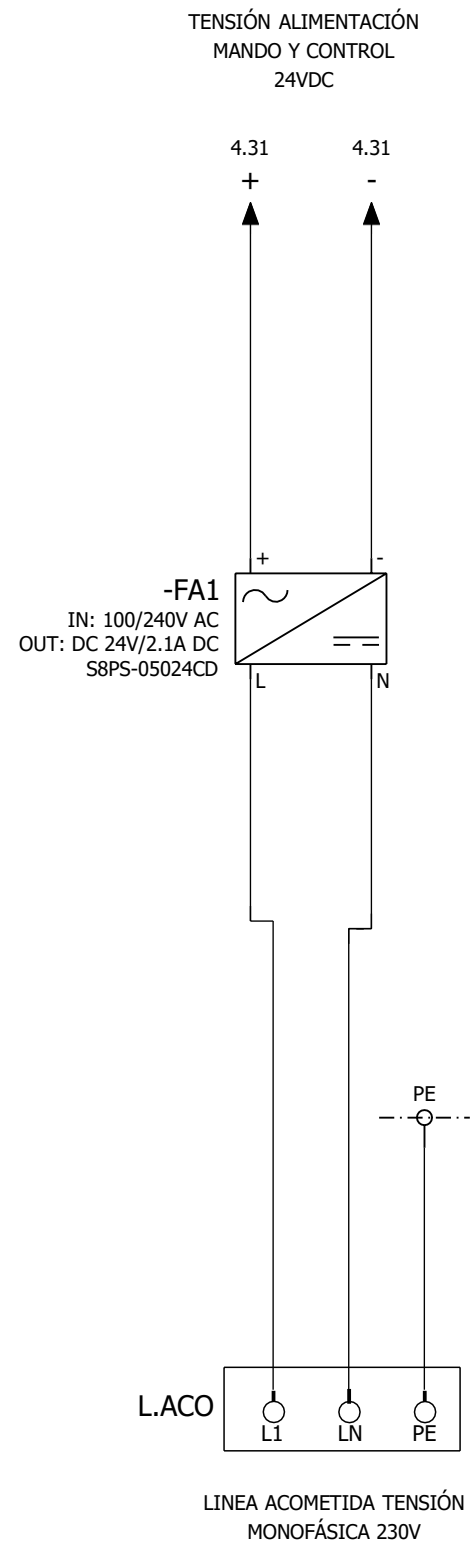
BINARII

PORTADA

Lecy Sport

Control a distancia de máquina de entrenamiento

Pág 1
De 13 Págs



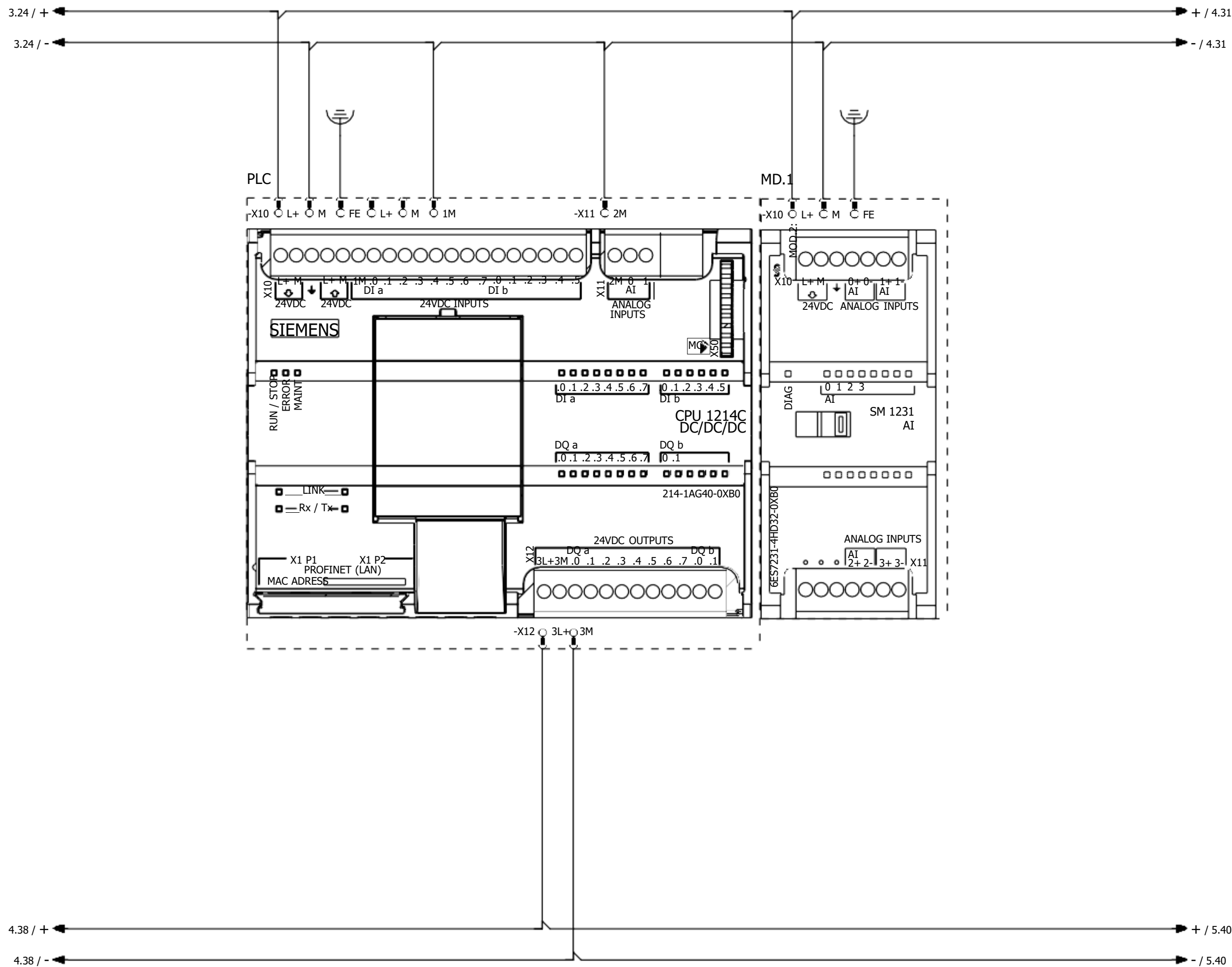
Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



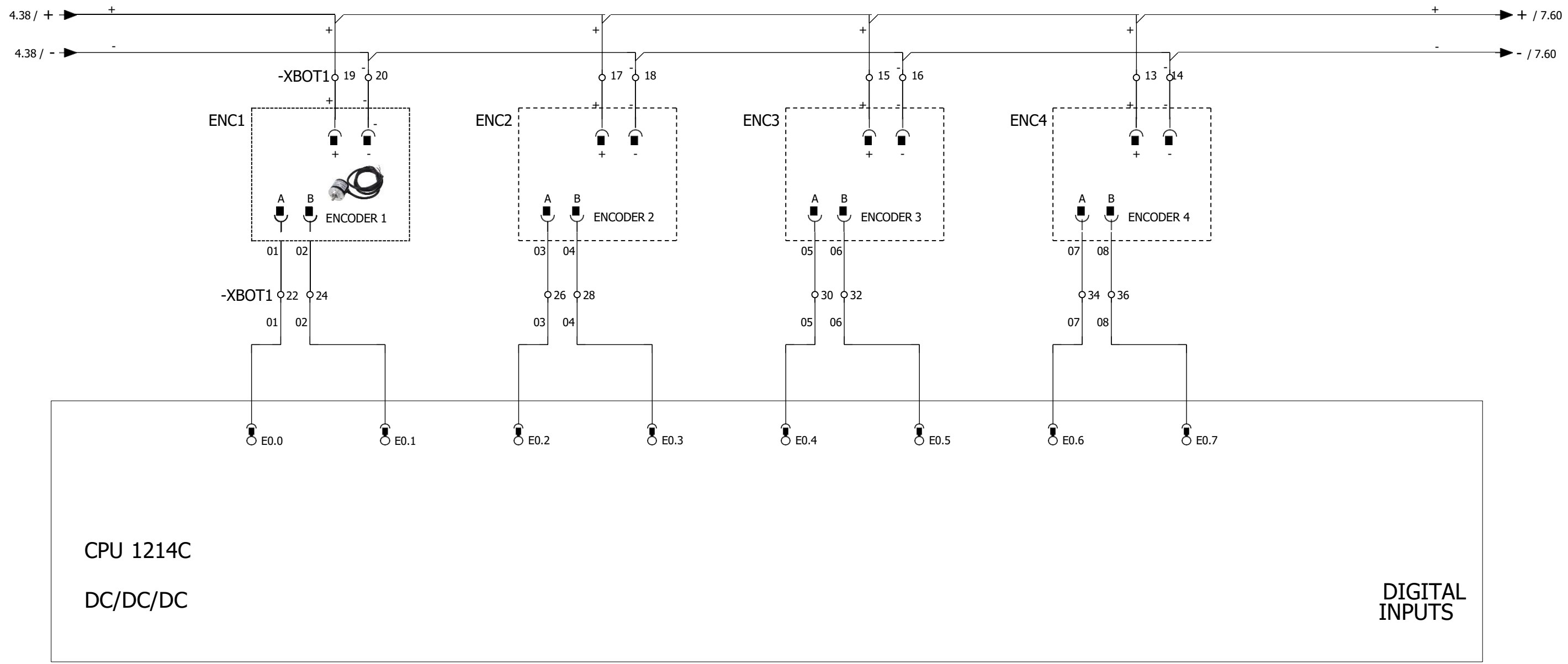
ESQUEMA DE FUERZA: ACOMETIDA Y DISTRIBUCIÓN

Lecy Sport

Control a distancia de máquina de entrenamiento



Date	24/02/2022
Responsible	Manuel David Santillana
Reviewed	Manuel David Santillana
Original	



A PHASE (ENCODER 1)

B PHASE (ENCODER 1)

A PHASE (ENCODER 2)

B PHASE (ENCODER 2)

A PHASE (ENCODER 3)

B PHASE (ENCODER 3)

A PHASE (ENCODER 4)

B PHASE (ENCODER 4)

CPU 1214C
DC/DC/DC

DIGITAL
INPUTS

Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



ENTRADAS DIGITALES PLC

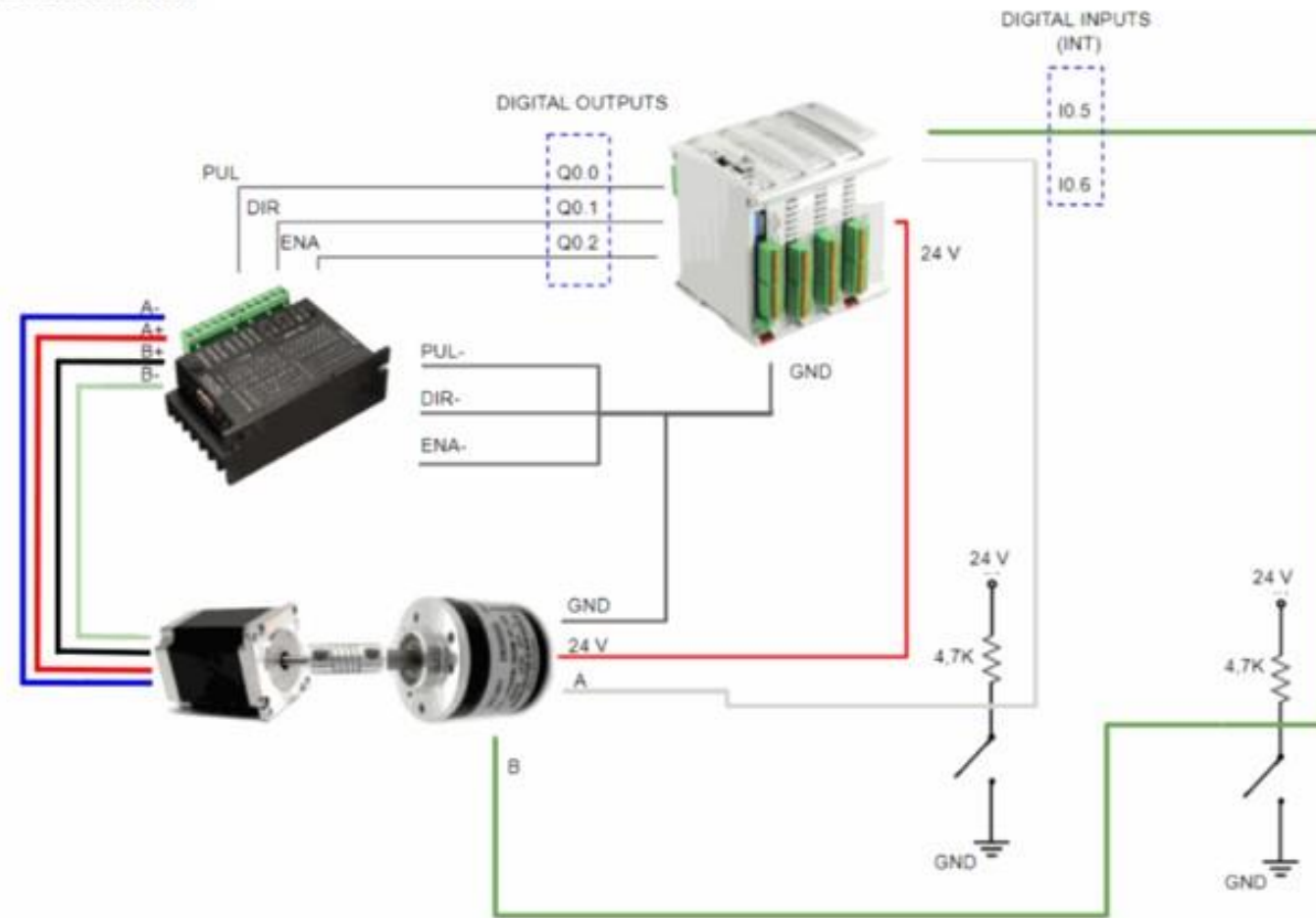
Lecy Sport	
Control a distancia de máquina de entrenamiento	Pág 5
	De 13 Págs

Encoder - Ethernet PLC Connections:

- +24Vdc --> Red
- Gnd --> Black
- I0.5 (A Phase) --> Green
- I0.6 (B Phase) --> Blanco

Figura 29. Conexión Encoder

Conexión final:



Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		

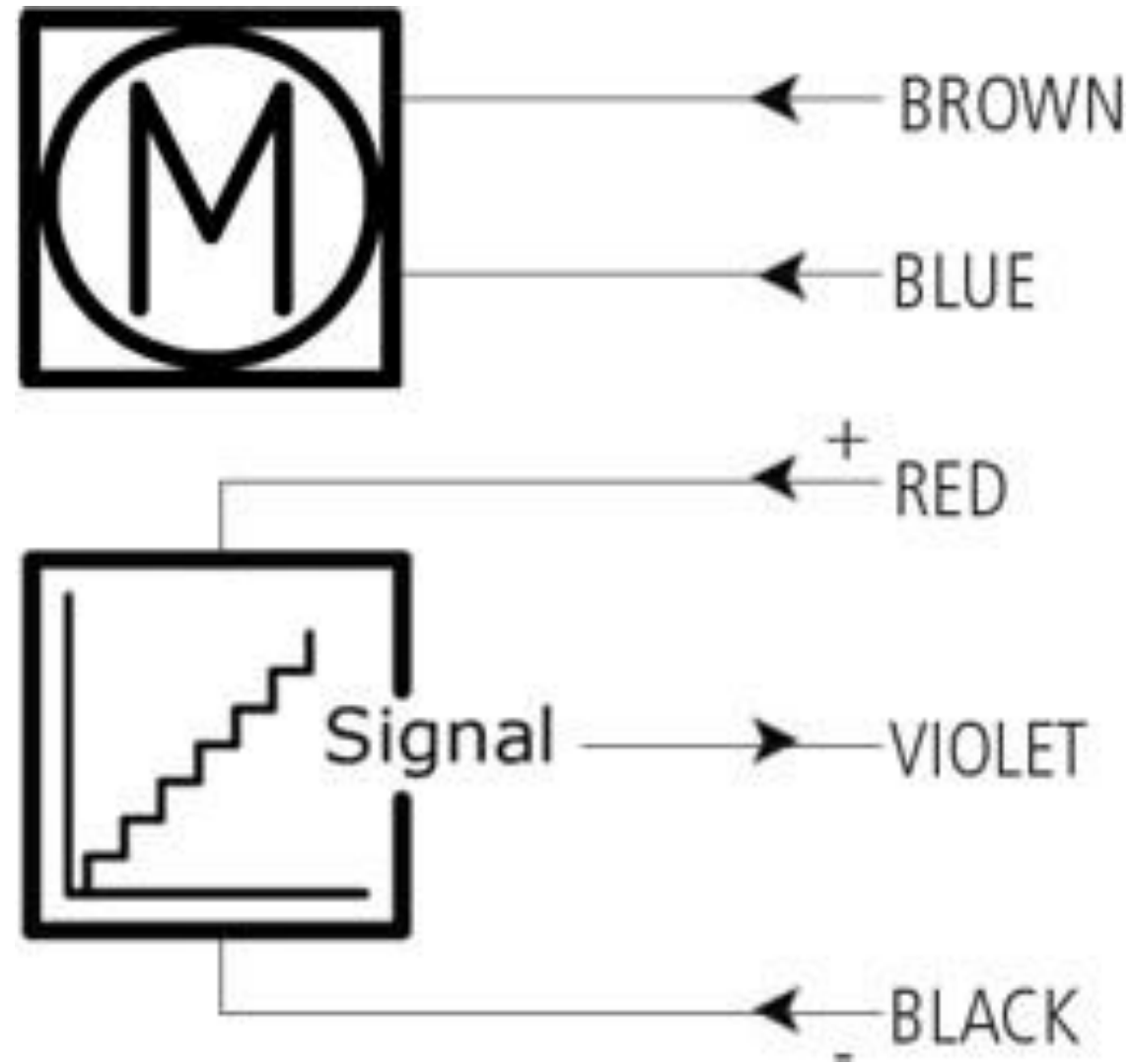


BINARII

CONEXIONADO ENCODER

Lecy Sport

Control a distancia de máquina de entrenamiento



Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



CONEXIONADO EJE ELÉCTRICO

Lecy Sport	
Control a distancia de máquina de entrenamiento	Pág 8 De 13 Págs

AVANZA EJE 1

RETROCEDE EJE 1

AVANZA EJE 2

RETROCEDE EJE 2

AVANZA EJE 3

RETROCEDE EJE 3

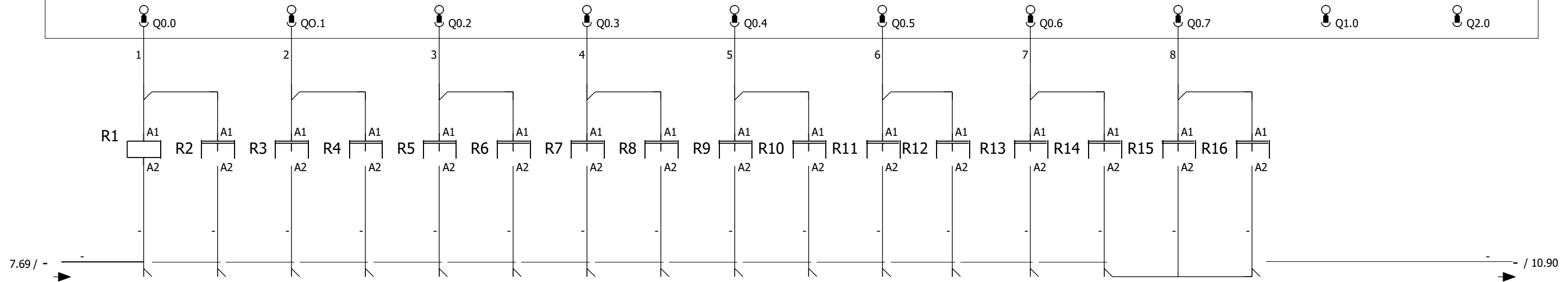
AVANZA EJE 4

RETROCEDE EJE 4

LIBRE

LIBRE

CPU 1214C
DC/DC/DC



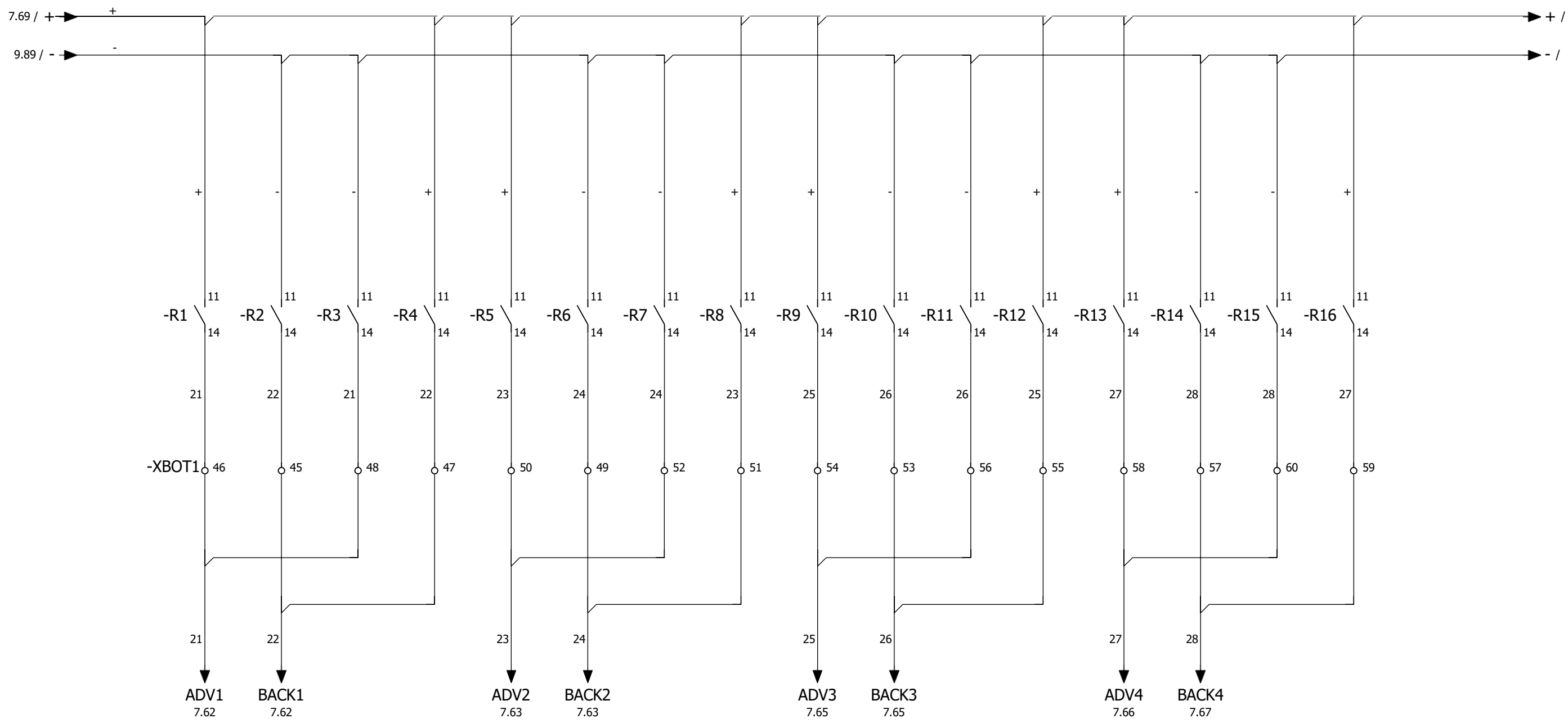
Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



SALIDAS PLC

Lecy Sport

Control a distancia de máquina de entrenamiento



Elaborador	GCC2	Cambios	Fecha
Realizado	17/02/2022	GCC3	08/04/2022
Comprobado	08/04/2022		
N.º proyecto	1		



BINARII

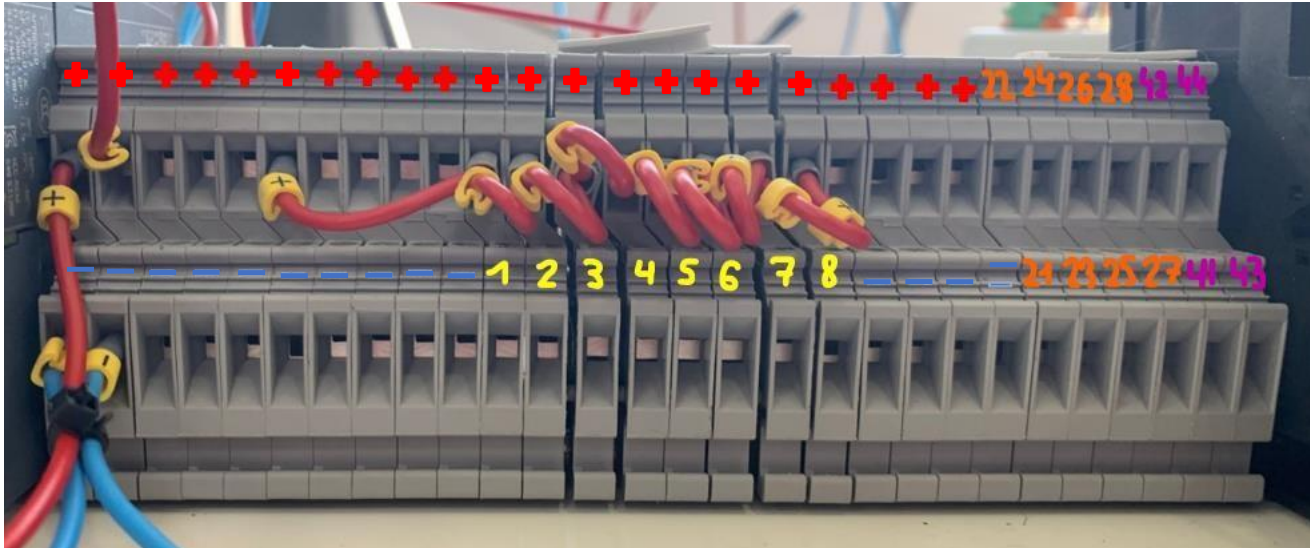
ACTIVACIONES DE LOS EJES

Lecy Sport	
Control a distancia de máquina de entrenamiento	Pág 10 De 13 Págs

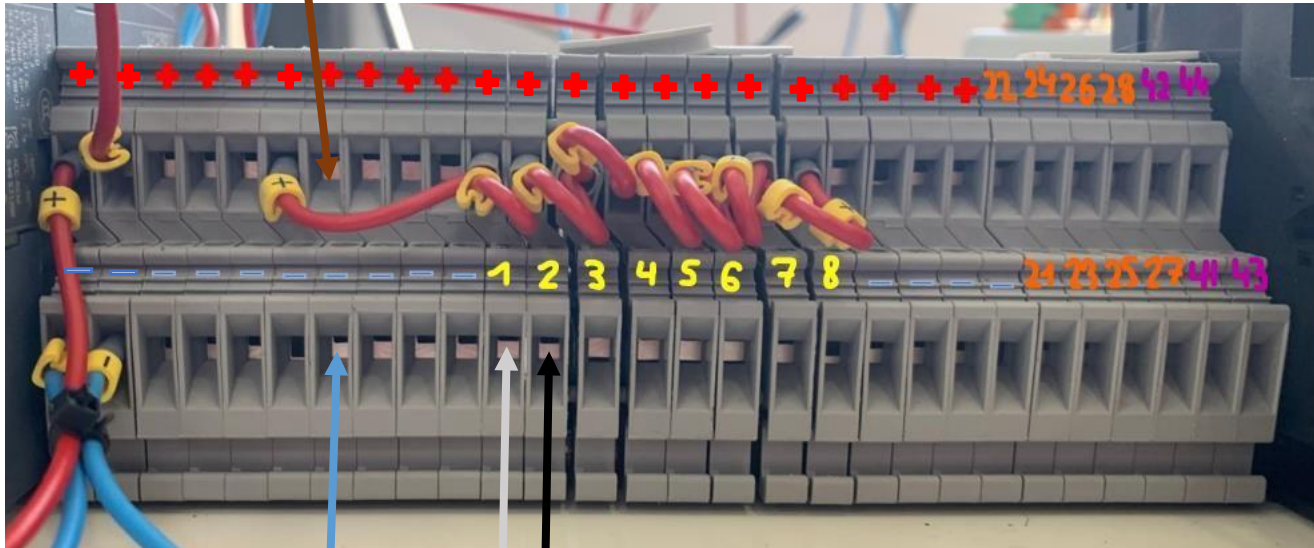
8.6 CONEXIONADO DEL HARDWARE EXTERNO

DISTRIBUCIÓN BORNERO	2
CONEXIONADO ENCODER 1	3
CONEXIONADO TODOS LOS ENCODERS.....	4
CONEXIONADO ENCODER 1 (FÍSICAMENTE).....	5
CONEXIONADO EJE ELÉCTRICO 1	6
CONEXIONADO TODOS LOS EJES ELÉCTRICOS (1).....	7
CONEXIONADO TODOS LOS EJES ELÉCTRICOS (2).....	8
CONEXIONADO EJE ELÉCTRICO 1 (FÍSICAMENTE)	9
CONEXIONADO WIFI TP-LINK	10
CONEXIONADO BATERÍA 24V	11

DISTRIBUCIÓN BORNERO



CONEXIONADO ENCODER 1

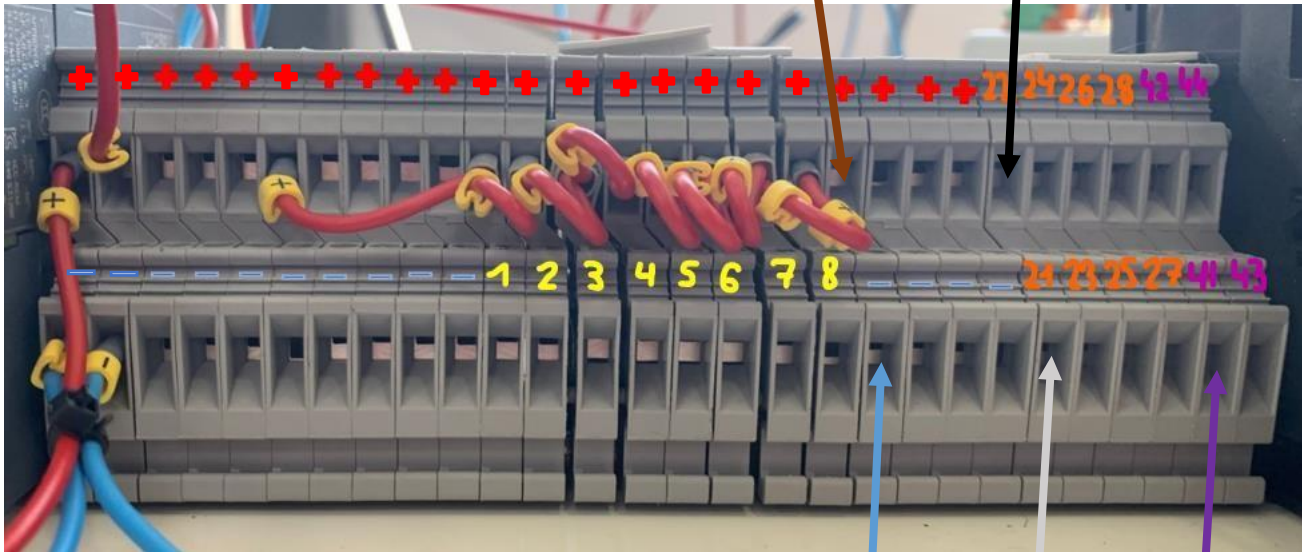
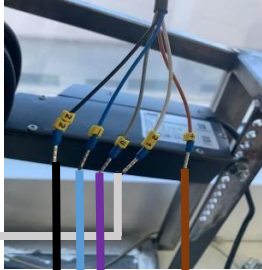


ENC 1

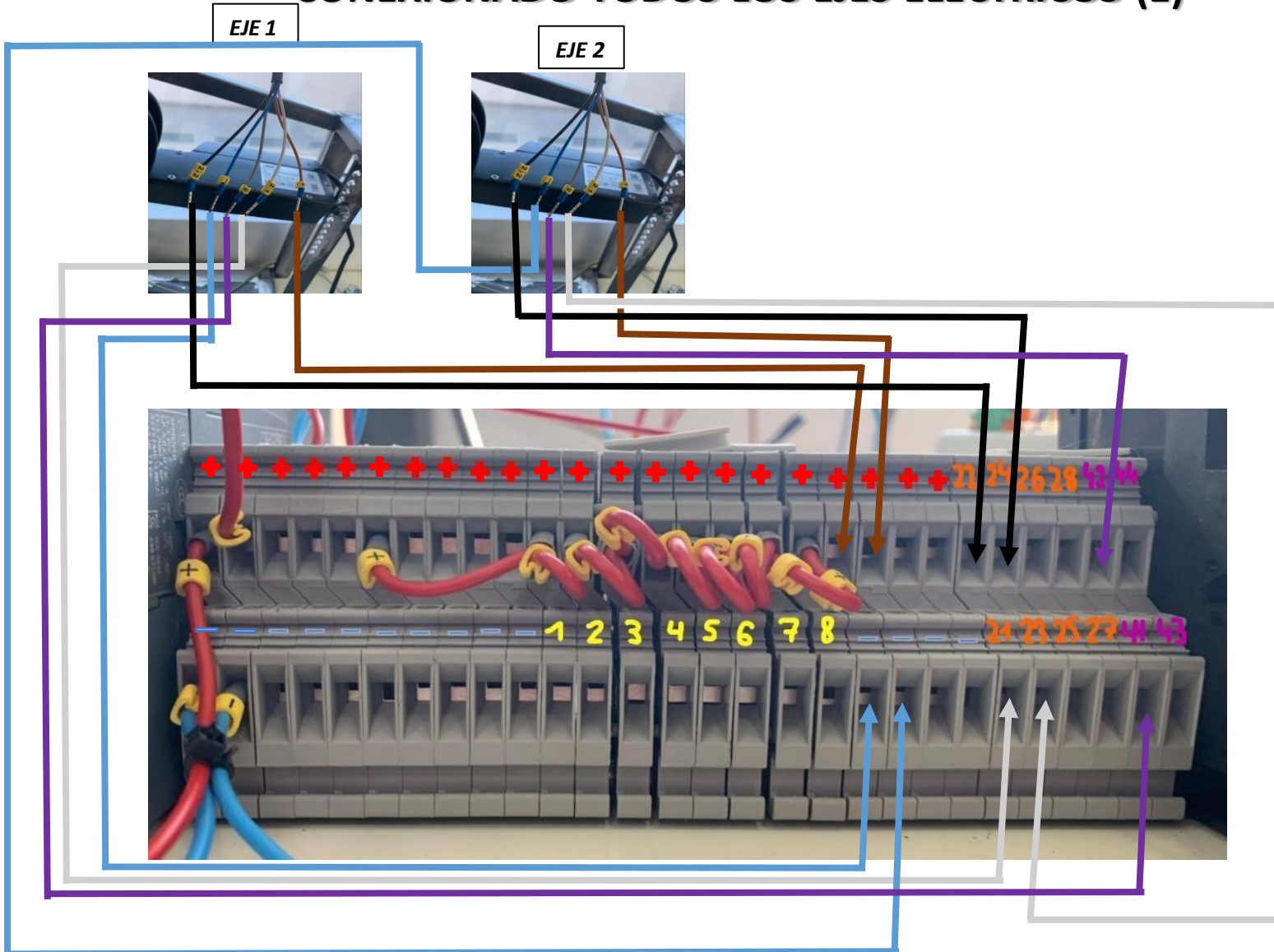


CONEXIONADO EJE ELÉCTRICO 1

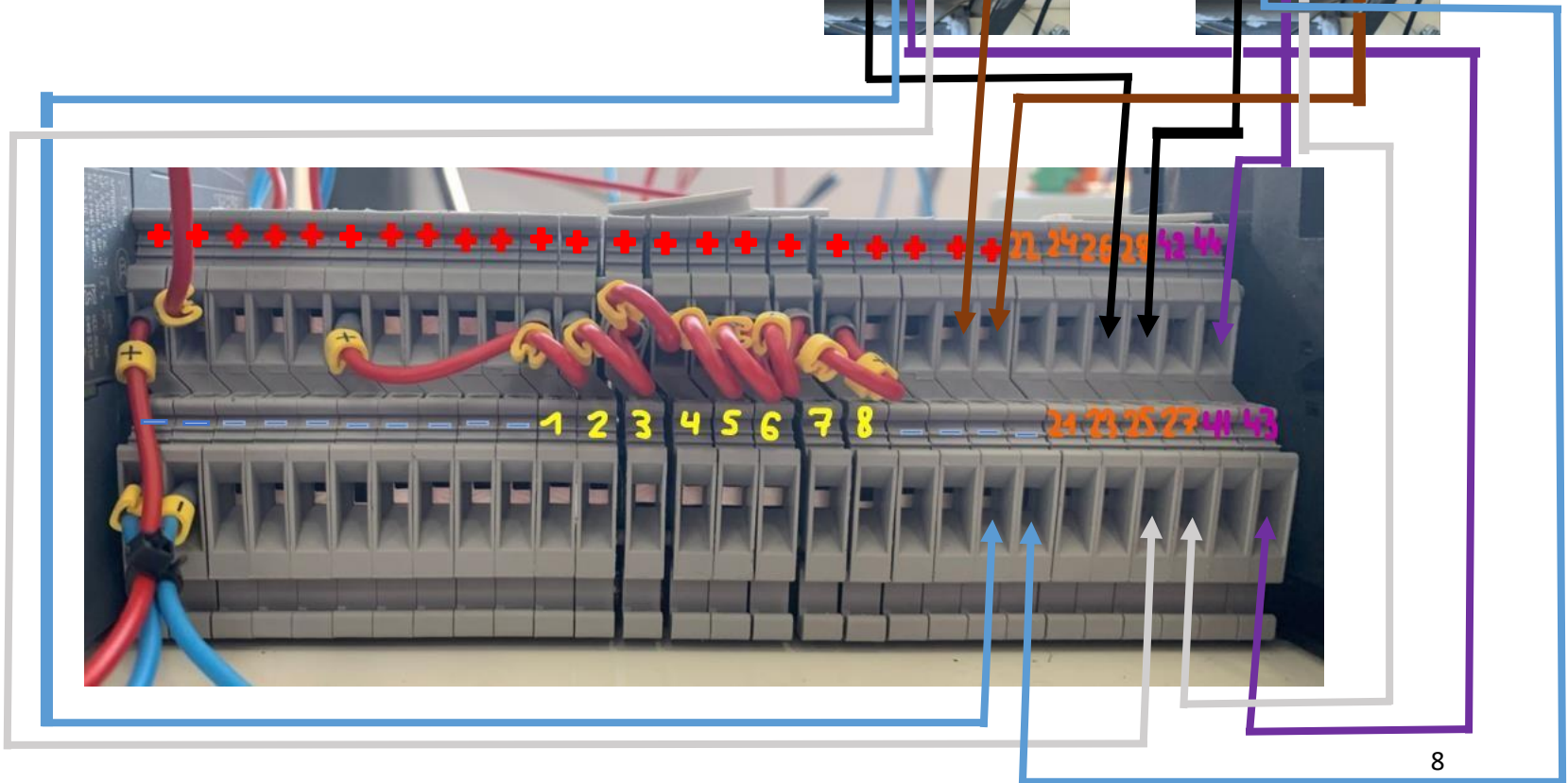
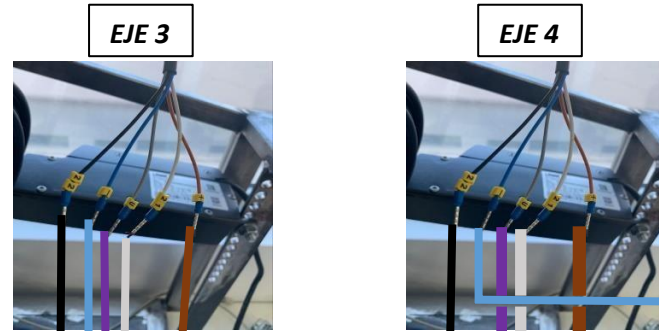
EJE 1



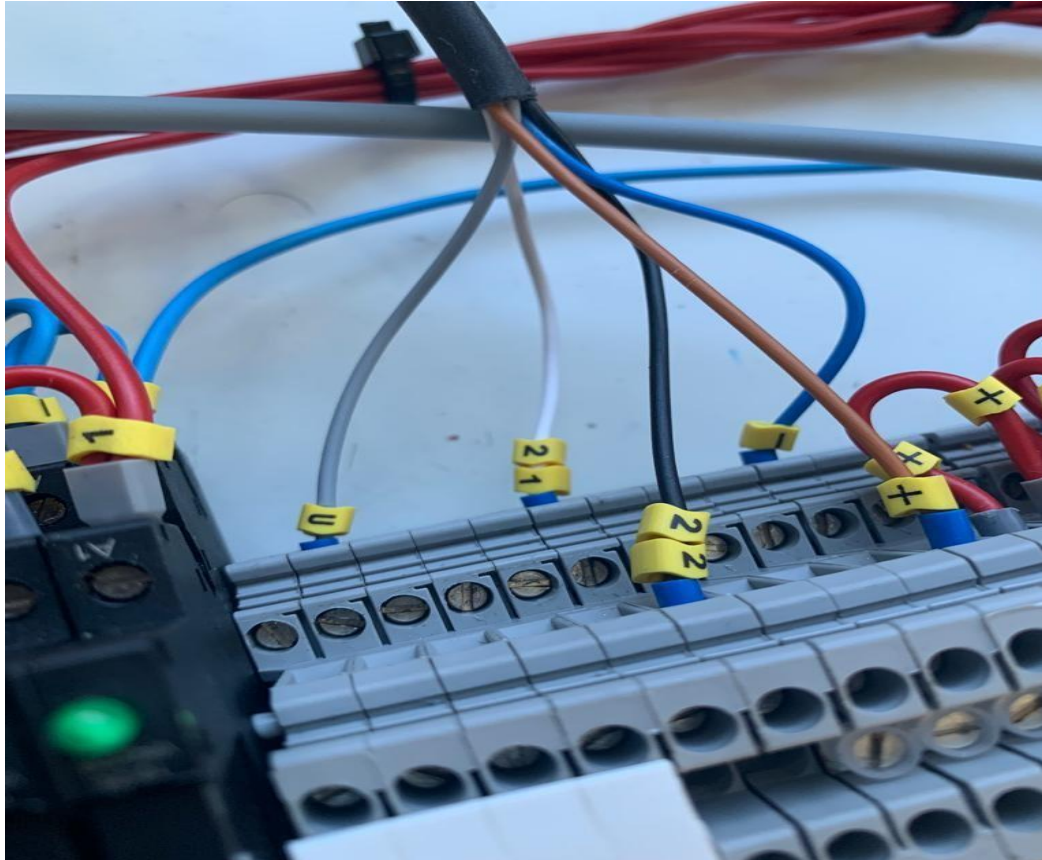
CONEXIONADO TODOS LOS EJES ELÉCTRICOS (1)



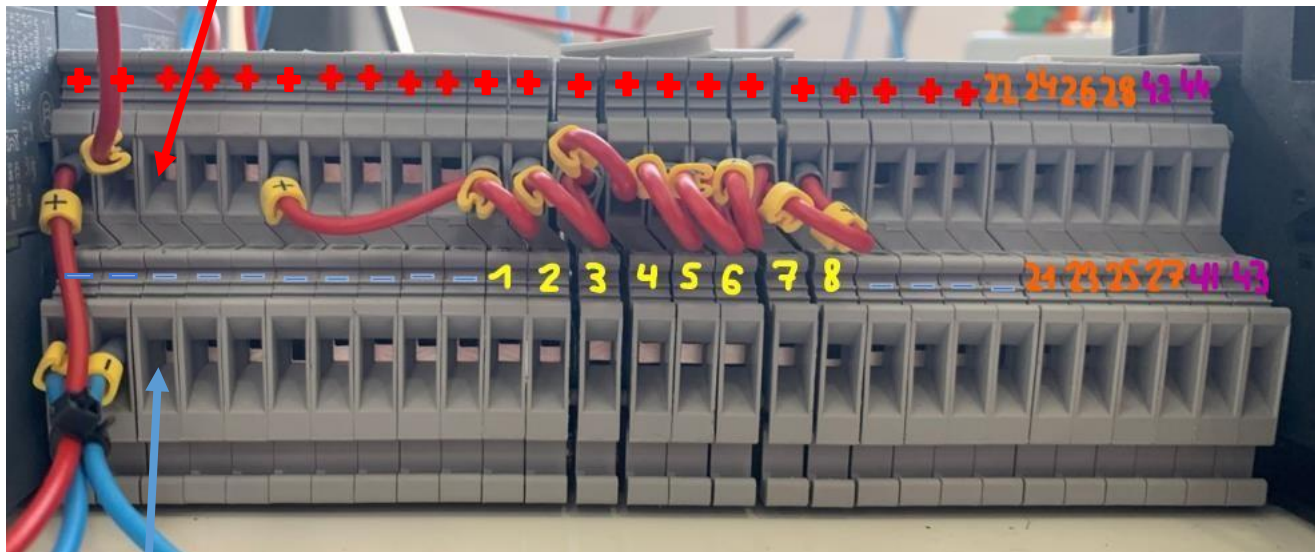
CONEXIONADO TODOS LOS EJES ELÉCTRICOS (2)



CONEXIONADO EJE ELÉCTRICO 1 (FÍSICAMENTE)



CONEXIONADO WIFI TP-LINK



CONEXIONADO BATERÍA 24V

